



普通高等教育
“十一五”规划教材

PRINCIPLES AND DEVELOPMENT OF EMBEDDED SYSTEMS

嵌入式系统原理与开发

范延滨 于忠清 郑立爱 编著



机械工业出版社
CHINA MACHINE PRESS

普通高等教育“十一五”规划教材

嵌入式系统 原理与开发

范延滨 于忠清 郑立爱 编著



机械工业出版社

本书基于“七个一”嵌入式系统学习方案：一个体系结构、一款微处理器、一款开发板、一种操作系统、一种驱动程序、一类开发环境、一类开发方法。优选了 ARM 体系结构、基于 ARM7TDMI 的 S3C44B0X 处理器、 μ C/OS-II 操作系统、RVDS 开发环境，由浅入深地对其逐一详细分析和研究，最终引导读者自己独立设计一款具有丰富外部接口的开发板、编写 BootLoader、移植 μ C/OS-II 和开发主要驱动程序，全面地完成一款开发板软硬件的研发。

本书强调“All in One”。一书贯通、速成高手！第 1 章对嵌入式系统的技术与应用进行了较全面地概述；第 2 章和第 3 章详细介绍了 ARM 处理器技术和 ARM 指令系统；第 4 章讨论了 ARM 汇编语言程序设计方法并给出了设计实例；第 5 章详细分析了 μ C/OS-II 嵌入式操作系统；第 6 章介绍了 S3C44B0X 微处理器并给出了寄存器的配置方法；第 7 章基于 S3C44B0X 设计了一款多功能开发板；第 8 章基于自己设计的开发板详细介绍了软件系统的设计技术。

本书适合于作高等学校计算机类、电子信息类、仪器仪表类本科生教材，也可供从事基于 ARM 的嵌入式系统开发的工程技术人员参考。

图书在版编目 (CIP) 数据

嵌入式系统原理与开发/范延滨，于忠清，郑立爱编著. —北京：机械工业出版社，2010.2

普通高等教育“十一五”规划教材

ISBN 978-7-111-29277-7

I. 嵌… II. ①范…②于…③郑… III. 微型计算机—系统开发—高等学校—教材 IV. TP360.21

中国版本图书馆 CIP 数据核字(2009)第 229561 号

机械工业出版社(北京市百万庄大街 22 号 邮政编码 100037)

策划编辑：王保家 责任编辑：王雅新 版式设计：霍永明

封面设计：王洪流 责任校对：张晓蓉 责任印制：洪汉军

三河市国英印务有限公司印刷

2010 年 3 月第 1 版第 1 次印刷

184mm×260mm·24.25 印张·599 千字

标准书号：ISBN 978-7-111-29277-7

定价：39.00 元

凡购本书，如有缺页、倒页、脱页，由本社发行部调换

电话服务

网络服务

社服务中心：(010)88361066

门户网：<http://www.cmpbook.com>

销售一部：(010)68326294

教材网：<http://www.cmpedu.com>

销售二部：(010)88379649

读者服务部：(010)68993821

封面无防伪标均为盗版

全国高等学校电气工程与自动化系列教材

编 审 委 员 会

主任委员 汪樾生 浙江大学

副主任委员 (按姓氏笔画排序)

王兆安 西安交通大学

王孝武 合肥工业大学

田作华 上海交通大学

刘 丁 西安理工大学

陈伯时 上海大学

郑大钟 清华大学

赵光宙 浙江大学

赵 曜 四川大学

韩雪清 机械工业出版社

委 员 (按姓氏笔画排序)

戈宝军 哈尔滨理工大学

王钦若 广东工业大学

吴 刚 中国科技大学

张纯江 燕山大学

张晓华 哈尔滨工业大学

邹积岩 大连理工大学

陈庆伟 南京理工大学

夏长亮 天津大学

萧蕴诗 同济大学

韩 力 重庆大学

熊 蕊 华中科技大学

方 敏 合肥工业大学

白保东 沈阳工业大学

张化光 东北大学

张 波 华南理工大学

杨 耕 清华大学

陈 冲 福州大学

范 瑜 北京交通大学

章 兢 湖南大学

程 明 东南大学

雷银照 北京航空航天大学

序

随着科学技术的不断进步，电气工程与自动化技术正以令人瞩目的发展速度，改变着我国工业的整体面貌。同时，对社会的生产方式、人们的生活方式和思想观念也产生了重大的影响，并在现代化建设中发挥着越来越重要的作用。随着与信息科学、计算机科学和能源科学等相关学科的交叉融合，它正在向智能化、网络化和集成化的方向发展。

教育是培养人才和增强民族创新能力的基础，高等学校作为国家培养人才的主要基地，肩负着教书育人的神圣使命。在实际教学中，根据社会需求，构建具有时代特征、反映最新科技成果的知识体系是每个教育工作者义不容辞的光荣任务。

教书育人，教材先行。机械工业出版社几十年来出版了大量的电气工程与自动化类教材，有些教材十几年、几十年长盛不衰，有着很好的基础。为了适应我国目前高等学校电气工程与自动化类专业人才培养的需要，配合各高等学校的教学改革进程，满足不同类型、不同层次的学校在课程设置上的需求，由中国机械工业教育协会电气工程及自动化学科教育委员会、中国电工技术学会高校工业自动化教育专业委员会、机械工业出版社共同发起成立了“全国高等学校电气工程与自动化系列教材编审委员会”，组织出版新的电气工程与自动化类系列教材。这类教材基于“**加强基础，削枝强干，循序渐进，力求创新**”的原则，通过对传统课程内容的整合、交融和改革，以不同的模块组合来满足各类学校特色办学的需要。并力求做到：

1. **适用性**：结合电气工程与自动化类专业的培养目标、专业定位，按技术基础课、专业基础课、专业课和教学实践等环节，进行选材组稿。对有的具有特色的教材采取一纲多本的方法。注重课程之间的交叉与衔接，在满足系统性的前提下，尽量减少内容上的重复。

2. **示范性**：力求教材中展现的教学理念、知识体系、知识点和实施方案在本领域中具有广泛的辐射性和示范性，代表并引导教学发展的趋势和方向。

3. **创新性**：在教材编写中强调与时俱进，对原有的知识体系进行实质性的改革和发展，鼓励教材涵盖新体系、新内容、新技术，注重教学理论创新和实践创新，以适应新形势下的教学规律。

4. **权威性**：本系列教材的编委由长期工作在教学第一线的知名教授和学者组成。他们知识渊博，经验丰富，组稿过程严谨细致，对书目确定、主编征集、资料申报和专家评审等都有明确的规范和要求，为确保教材的高质量提供了有

力保障。

此套教材的顺利出版，先后得到全国数十所高校相关领导的大力支持和广大骨干教师的积极参与，在此谨表示衷心的感谢，并欢迎广大师生提出宝贵的意见和建议。

此套教材的出版如能在转变教学思想、推动教学改革、更新专业知识体系、创造适应学生个性和多样化发展的学习环境、培养学生的创新能力等方面收到成效，我们将会感到莫大的欣慰。

全国高等学校电气工程与自动化系列教材编审委员会

汪植生 陈永时 郑大钟

前言

嵌入式技术是继网络技术之后,又一个新的技术发展方向。嵌入式系统是计算机软件与硬件的完美结合,广泛地应用于手持设备、信息家电、仪器仪表、汽车电子、医疗仪器、工业控制、航天航空等各个领域,并嵌入在各类设备之中,起着核心作用。

嵌入式系统已经无处不在,嵌入式人才主要分为两大类:一类是嵌入式硬件工程师,主要从事嵌入式系统硬件研发,包括硬件系统原理图的设计、PCB印制电路板的设计,开发与硬件相关的底层软件(如 BootLoader、嵌入式驱动程序等);另一类是嵌入式软件工程师,主要从事嵌入式操作系统和应用软件的开发,如果对嵌入式硬件原理和接口技术有较好的掌握,也可以从事嵌入式系统底层程序的开发。

在嵌入式系统的学习过程中,建议读者抓住“七个一”来学习:一个体系结构、一款微处理器、一款开发板、一种操作系统、一种驱动程序、一类开发环境、一类开发方法。

实际上,虽然说嵌入式系统纷杂繁多,但是都符合“七个一”的组织结构。因此,只要读者能够独立地、完整地开发出一款嵌入式系统产品,就能够掌握嵌入式系统开发的核心技术,那么在以后的实际工作中,不论遇到何种嵌入式系统,都能够独立开发了。

一个嵌入式系统必定是由嵌入式硬件和嵌入式软件组成;嵌入式硬件主要包括嵌入式处理器、外设控制器、接口电路等;嵌入式软件主要包括启动程序、嵌入式操作系统、应用软件等。嵌入式处理器是嵌入式系统的硬件核心;嵌入式操作系统是嵌入式软件的核心。

嵌入式处理器和嵌入式操作系统种类繁多,初学者在学习嵌入式系统时,都存在应该选择何种嵌入式处理器、何种嵌入式操作系统来学习等疑问。本书选择了 ARM 体系结构、基于 ARM7TDMI 的 S3C44B0X 处理器、 μ C/OS-II 操作系统、RVDS 开发环境,由浅入深地对其逐一详细分析和研究,最终引导读者自己独立设计一款开发板、编写启动代码、移植 μ C/OS-II 和开发部分驱动程序,直到开发板成功运行。之所以这样选择,原因如下:

在选择嵌入式处理器时,初学者应该更多地注重其适应性、易学性,不要选择冷、偏、难的处理器作为学习对象。在各种嵌入式系统中,ARM 体系结构占据 32 位市场的 80%;在 ARM 体系结构中,ARM7TDMI 是颇为成功的一款内核;在基于 ARM7TDMI 的嵌入式处理器中,S3C44B0X 是颇具代表性的一款芯片。而且,S3C44B0X 具有丰富的片内外设和很强的外围扩展功能,如果学会了用 S3C44B0X 构建嵌入式系统,那么基本上就掌握了嵌入式系统的硬件设计技术,包括微处理片内外设配置技术、高速电路设计技术、外扩控制器技术等。另外,更为重要的是基于 S3C44B0X 的系统,能够用双面印制电路板实现,便于设计和制作,极大地降低了学习成本和难度,但是技术水平却很高。

在选择嵌入式操作系统时,初学者应该更多地注重其开放性、易读性,并能更好地发挥出处理器的性能。在各种嵌入式操作系统中, μ C/OS-II 是一款实时嵌入式操作系统内核。

首先,其代码精炼、设计规范且源码开放,5000 余行代码十分适合于学习和研读;另外, $\mu\text{C}/\text{OS-II}$ 的功能强大,充分展现了实时嵌入式操作系统的各种技术;更重要的是 $\mu\text{C}/\text{OS-II}$ 不仅适合于在基于 S3C44B0X 的系统上运行,而且也适合于在 DSP 系统上运行。详细研读之,将终生受益。

本书所提供的开发板、代码都是调试通过的,可以向读者提供源文件。开发板以 S3C44B0X 为核心扩展了 2MB 的 Flash、8MB 的 SDRAM、IDE/CF 卡接口、LCD 接口,外扩了 RTL8019 控制器并设计了 10M 以太网接口,外扩了 USB D12 控制器并设计了 USB 接口,外扩了 ZLG7290 控制器并设计了 8 位 LED 和 16 键数字键盘。软件系统设计了 Boot-Loader 和串行口驱动程序、移植了 $\mu\text{C}/\text{OS-II}$ 。

本书适合于本科生教材。作为教材最关键的问题是要具有系统性、完整性、可读性和代表性,并能向学生开放教学资料。本书在“七个一”的指导下,每一部分都选择了最有代表性内容研究,全书从基本内容出发,以读者自行完成一款开发板为目标,展开系统、完整地学习,最终让读者经历一个具有丰富功能的开发板项目研发的全过程。

作者向读者提供下述资料:

免费提供本书 PPT 格式的电子课件;

免费提供实例源码、BootLoader 源码、 $\mu\text{C}/\text{OS-II}$ 移植源码、应用程序源码;

免费提供基于 S3C44B0X 实验板的 PDF 格式的原理图、PCB 版图;

提供基于 S3C44B0X 实验板的 Protel 格式的原理图(共 10 张图)、PCB 版图,需付费;

提供基于 S3C44B0X 核心板的 Protel 格式的原理图(共 4 张)、PCB 版图(2 层板和 4 层板),需付费;

提供基于 S3C44B0X 实验板的开发套件,需付费。

欢迎选用本书做教材的老师和读者通过以下电子邮箱联系索取。

电子邮箱:ataisimlm@163.com。

作者

参 考 文 献

- [1] 田泽. 嵌入式系统开发与应用 [M]. 北京: 北京航空航天大学出版社, 2005.
- [2] 任哲. 嵌入式实时操作系统 $\mu\text{C}/\text{OS-II}$ 原理及应用 [M]. 北京: 北京航空航天大学出版社, 2005.
- [3] 杜春雷. ARM 体系结构与编程 [M]. 北京: 清华大学出版社, 2003.
- [4] Labrosse Jeans J. 嵌入式实时操作系统 $\mu\text{C}/\text{OS-II}$ [M]. 2 版. 邵贝贝, 译. 北京: 北京航空航天大学出版社, 2005.
- [5] 马忠梅. ARM 嵌入式处理器结构与应用基础 [M]. 北京: 北京航空航天大学出版社, 2002.

目 录

序

前言

第1章 嵌入式系统概述	1	2.4.1 ARM 流水线概述	40
1.1 嵌入式系统的概念	1	2.4.2 ARM7 的3级流水线	40
1.1.1 嵌入式系统定义	1	2.5 ARM 处理器的存储器组织	41
1.1.2 嵌入式系统特点	2	2.5.1 ARM 存储器的层次	42
1.2 嵌入式系统分类	4	2.5.2 ARM 存储器的管理	44
1.3 嵌入式系统组成结构	5	2.5.3 ARM 非对齐的存储访问操作	45
1.4 嵌入式处理器简介	8	2.6 ARM 处理器的片上总线	46
1.4.1 嵌入式处理器分类	8	2.6.1 AMBA 简介	46
1.4.2 嵌入式微处理器	9	2.6.2 AHB 简介	47
1.4.3 嵌入式微控制器	9	2.6.3 ASB 简介	48
1.4.4 嵌入式数字信号处理器	10	2.6.4 APB 简介	48
1.4.5 嵌入式片上系统	10	2.7 ARM 处理器的指令体系结构	48
1.5 嵌入式操作系统简介	10	第3章 ARM 指令系统	50
1.5.1 嵌入式操作系统发展	11	3.1 ARM 指令集概述	50
1.5.2 嵌入式实时操作系统	11	3.2 ARM 寻址方式	54
1.5.3 嵌入式操作系统选型	12	3.3 ARM 数据处理指令	59
1.5.4 几种嵌入式操作系统比较	13	3.4 ARM Load/Store 指令	67
1.6 嵌入式系统设计流程	14	3.5 ARM 程序状态寄存器传送指令	71
1.6.1 嵌入式系统开发模式	15	3.6 ARM 转移指令	73
1.6.2 嵌入式系统开发流程	16	3.7 ARM 异常中断指令	75
1.7 嵌入式系统应用	16	3.8 ARM 协处理器指令	76
第2章 ARM 处理器	19	第4章 ARM 汇编语言程序设计	80
2.1 ARM 系列处理器概述	19	4.1 概述	80
2.1.1 ARM 系列处理器术语	19	4.1.1 开发工具选择	80
2.1.2 ARM 处理器体系结构	21	4.1.2 ADS 开发工具	81
2.1.3 ARM 处理器产品系列	22	4.2 ADS 环境下的伪操作和伪指令	82
2.1.4 ARM 处理器技术进展	23	4.2.1 符号定义类伪操作	83
2.1.5 ARM 处理器组成结构	27	4.2.2 数据定义类伪操作	85
2.2 ARM 处理器的寄存器文件	29	4.2.3 汇编控制类伪操作	89
2.2.1 ARM 处理器的工作状态	29	4.2.4 信息报告类伪操作	91
2.2.2 ARM 处理器的工作模式	30	4.2.5 其他伪操作	92
2.2.3 ARM 状态下寄存器组织	31	4.2.6 ARM 汇编语言伪指令详解	96
2.2.4 Thumb 状态下寄存器组织	34	4.2.7 Thumb 汇编语言伪指令详解	98
2.3 ARM 处理器的异常与中断	36	4.3 ARM 汇编语言程序设计	98
2.4 ARM 处理器的流水线结构	40	4.3.1 ARM 汇编语言程序设计概述	98
		4.3.2 ARM 汇编语言语句格式	99
		4.4 GNU ARM 汇编语言	102

4.5 ARM 汇编语言程序设计实例	106	6.2 S3C44B0X 存储控制器功能及 应用开发	181
4.5.1 ARM 汇编语言子程序格式	106	6.2.1 S3C44B0X 存储空间概述	181
4.5.2 ARM 汇编语言子程序	107	6.2.2 S3C44B0X 存储控制器的 SFR	184
4.5.3 ARM 汇编语言子程序与 C 语言	112	6.2.3 S3C44B0X 存储控制器应用编程	187
第 5 章 μC/OS-II 嵌入式操作系统	113	6.3 S3C44B0X 的 GPIO 端口功能及 应用开发	189
5.1 μ C/OS-II 嵌入式操作系统的概念	113	6.3.1 S3C44B0X 的 GPIO 概述	189
5.2 μ C/OS-II 中的任务	115	6.3.2 S3C44B0X 的 GPIO 端口的 SFR	191
5.2.1 任务定义	116	6.3.3 S3C44B0X 的 GPIO 端口的应用 编程	197
5.2.2 任务堆栈	119	6.4 S3C44B0X 时钟电源管理功能及 开发	198
5.2.3 任务控制块	120	6.4.1 S3C44B0X 的时钟电源管理部件的 简述	198
5.2.4 任务调度	123	6.4.2 S3C44B0X 的时钟电源管理部件 的 SFR	199
5.2.5 任务创建	130	6.5 S3C44B0X 的 INTC 功能及开发	201
5.2.6 任务的挂起和恢复	132	6.5.1 S3C44B0X 的 INTC 概述	201
5.2.7 其他任务管理	133	6.5.2 S3C44B0X 的 INTC 的 SFR	203
5.3 μ C/OS-II 中的中断	135	6.6 S3C44B0X 的 UART 功能及应用 开发	209
5.3.1 中断过程	135	6.6.1 S3C44B0X 的 UART 概述	209
5.3.2 中断服务程序	136	6.6.2 S3C44B0X 的 UART 的 SFR	211
5.3.3 中断级任务切换函数	137	6.7 S3C44B0X 的 WDT 定时器功能及 应用开发	216
5.3.4 临界段	137	6.7.1 S3C44B0X 的看门狗概述	216
5.4 μ C/OS-II 中的时钟	138	6.7.2 S3C44B0X 的 WDT 的 SFR	217
5.4.1 时钟节拍	138	6.8 S3C44B0X 的 IIC 定时器功能及应用 开发	217
5.4.2 时间管理	140	6.8.1 S3C44B0X 的 IIC 概述	218
5.5 μ C/OS-II 中任务的同步与通信	142	6.8.2 S3C44B0X 的 IIC 的 SFR	219
5.5.1 同步	142	6.9 S3C44B0X 的 LCD 功能及应用开发	221
5.5.2 事件	143	6.9.1 S3C44B0X 的 LCD 控制器概述	221
5.5.3 事件控制块	144	6.9.2 S3C44B0X 的 LCD 控制器的 SFR	224
5.5.4 事件方法	147	6.10 S3C44B0X 的 DMAC 功能及应用 开发	227
5.6 μ C/OS-II 中的信号量	150	6.10.1 S3C44B0X 的 DMAC 概述	227
5.7 μ C/OS-II 中的互斥信号量	153	6.10.2 S3C44B0X 的 DMAC 的 SFR	229
5.8 μ C/OS-II 中的消息邮箱	155	6.11 S3C44B0X 的 RTC 功能及应用 开发	234
5.9 μ C/OS-II 中的消息队列	157		
5.10 μ C/OS-II 中的信号量集	162		
5.10.1 信号量集的定义	163		
5.10.2 信号量集的操作	166		
5.11 μ C/OS-II 中的内存管理	168		
5.11.1 内存控制块	168		
5.11.2 动态内存的管理	170		
第 6 章 S3C44B0X 嵌入式微处理器	173		
6.1 S3C44B0X 简介	173		
6.1.1 S3C44B0X 的功能	173		
6.1.2 S3C44B0X 的内特性	174		
6.1.3 S3C44B0X 的外特性	177		

6.11.1 S3C44B0X 的 RTC 概述	235	7.10 CF 卡/ATA 接口的设计	306
6.11.2 S3C44B0X 的 RTC 的 SFR	236	7.11 PCB 版图	313
6.12 S3C44B0X 的 PWM 定时器功能及 应用开发	241	第 8 章 基于 S3C44B0X 的软件系统	
6.12.1 S3C44B0X 的 PWM 概述	241	开发	315
6.12.2 S3C44B0X 的 PWM 的 SFR	244	8.1 BootLoader 技术	315
6.13 S3C44B0X 的 IIS 功能及应用开发	247	8.1.1 BIOS 技术	315
6.13.1 S3C44B0X 的 IIS 概述	248	8.1.2 BootLoader 的基本概念	316
6.13.2 S3C44B0X 的 IIS 的 SFR	250	8.1.3 基于 Linux 的 BootLoader 的基本 原理	318
6.14 S3C44B0X 的 SIO 功能及应用开发	252	8.1.4 开发板的主要配置	324
6.14.1 S3C44B0X 的 SIO 概述	252	8.2 基于 S3C44B0X 的 BootLoader 的 设计	324
6.14.2 S3C44B0X 的 SIO 的 SFR	254	8.2.1 异常中断机制	324
6.15 S3C44B0X 的 ADC 功能及应用开发	256	8.2.2 Stage1 程序设计	329
6.15.1 S3C44B0X 的 ADC 概述	256	8.2.3 Stage2 程序设计	344
6.15.2 S3C44B0X 的 ADC 的 SFR	257	8.2.4 UART0 初始化	347
第 7 章 基于 S3C44B0X 硬件系统		8.3 基于 S3C44B0X 的 $\mu\text{C}/\text{OS-II}$ 移植	354
开发	259	8.3.1 $\mu\text{C}/\text{OS-II}$ 的移植技术	354
7.1 S3C44B0X 硬件开发概述	259	8.3.2 OS_CPU.H	357
7.1.1 S3C44B0X 开发板的结构	259	8.3.3 OS_CPU_A.ASM	358
7.1.2 硬件设计技术	260	8.3.4 OS_CPU_C.C	362
7.2 S3C44B0X 的硬件特性	264	8.4 基于 $\mu\text{C}/\text{OS-II}$ 的串行口驱动程序 开发	364
7.3 核心板的设计	267	8.4.1 无操作系统的串行口驱动程序 开发	364
7.3.1 核心板的概述	267	8.4.2 基于 $\mu\text{C}/\text{OS-II}$ 的串行口通信 程序开发	370
7.3.2 Flash 的扩展接口	267	8.5 基于 $\mu\text{C}/\text{OS-II}$ 的 A/D 驱动程序 开发	371
7.3.3 SDRAM 的扩展接口	272	8.6 基于 $\mu\text{C}/\text{OS-II}$ 的设备驱动程序统一 框架	373
7.3.4 时钟与复位电路设计	277	参考文献	376
7.3.5 JTAG 调试电路设计	279		
7.4 电源板的设计	280		
7.5 UART 接口的设计	281		
7.6 USB 接口的设计	283		
7.7 NET 接口的设计	287		
7.8 矩阵键盘和数码管显示接口的设计	295		
7.9 LCD 接口的设计	301		

第 1 章 嵌入式系统概述

嵌入式系统 (Embedded System) 是计算机技术、半导体技术、微电子技术、制造技术等多种先进技术的高度融合。嵌入式技术是继网络技术之后, 又一主流技术方向。嵌入式系统已经渗透到人类社会的每个角落, 如科学教育、工业生产、军事国防、航空航天、医疗卫生、汽车电子、网络设备、消费电子、信息家电等, 都在广泛地、深入地应用着嵌入式系统。例如, 小到微型机器人、手机、PDA 等, 大到飞机、导弹、航天器等, 其中都嵌入着一个乃至多个独立的“嵌入式系统”。

1.1 嵌入式系统的概念

计算机系统有两大分支: 通用计算机系统和嵌入式计算机系统 (简称嵌入式系统)。

通用计算机系统采用标准化设计: 例如 PC, 在硬件上采用标准架构, 包含通用中央处理器 (CPU)、标准输入输出 (I/O 设备)、标准存储设备、标准总线结构等; 在软件上也采用标准架构, 通用操作系统、标准 API 函数等。因此, PC 的软硬件都是标准的, 生产商也是相对集中的, 主要有 Intel、AMD、Microsoft 等。

嵌入式计算机系统是非标准化设计: 虽然嵌入式系统具有通用计算机系统的一般特点, 但它是服务于所嵌入的应用对象为目标的, 是一种专用计算机系统, 具有很多的特殊性。例如, 一款手机除了具有一般的通话功能外, 还可以具有 GPRS 功能、照相功能、MP3 功能等。因此, 嵌入式系统需要根据具体应用量身定制, 通过选择合适的嵌入式处理器、外围功能模块等来定制硬件系统, 通过选择合适的嵌入式操作系统、应用协议等来定制软件系统。

1.1.1 嵌入式系统定义

嵌入式系统的应用范围迅速扩大, 使得其外延更加宽泛。例如, 一台 PC104 微型工业控制计算机是典型的嵌入式系统; 一部手机也是典型的嵌入式系统。然而, 二者几乎又是完全不同的。这样就为嵌入式系统定义的标准化带来了困难, 一般有如下几种定义方式。

1. IEEE (国际电子电气工程师协会) 的定义

嵌入式系统是“控制、监视或者辅助装置、机器和设备运行的装置” (Devices used to control, monitor, or assist the operation of equipment, machinery or plants)。这个定义主要是从应用上加以定义的, 从中可以看出嵌入式系统是软件和硬件的融合体, 还可以涵盖机械等附属装置。

2. 国内的定义

嵌入式系统是“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”。

国内从事嵌入式系统的工作者一般认为 IEEE 的定义并没有充分体现出嵌入式系统的精

髓,而国内专家学者所给出的定义更为深刻、更为确切、更具有普遍意义。

- 嵌入式系统是面向用户、面向产品、面向应用的,它必须与具体应用相结合才会具有生命力、才更具有优势。因此,可以这样理解该定义的含义:即嵌入式系统是一个计算机系统,它与应用紧密结合且具有很强的专用性,它必须结合实际系统需求而进行合理的设计。

- 嵌入式系统是多种高新技术的融合体,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。因此,进入嵌入式系统行业者,必须要有一个正确的定位。例如, Palm 之所以能在该领域占有 70% 以上的 PDA 市场,就是因为其立足于消费电子,着重发展图形界面和多任务管理;而风河的 Vxworks 之所以在火星车上得以应用,则是因其高实时性和高可靠性。

- 嵌入式系统必须根据应用需求对软硬件进行裁剪,满足应用系统的功能、可靠性、成本、体积等要求。所以,如果能建立相对通用的软硬件基础,然后在其上开发出适应各种需要的系统,是一个比较好的发展模式。目前的嵌入式系统的核心往往是一个只有几千到几万字字节微内核,需要根据实际的使用进行功能扩展或者裁减,但是由于微内核的存在,使得这种扩展能够非常顺利地进行。

3. 行业的定义

AMD 认为“除桌面微型电脑与笔记微型电脑之外,其他的都叫做嵌入式产品”。显然,这个定义极为宽泛。在这种定义下, x86 系列的嵌入式产品市场变得巨大,因此对生产 x86 体系结构 CPU 的厂商更加实用。例如, Intel 的 Atom 系列已经成为嵌入式处理器的重要分支。

4. 本书的注解

实际上,嵌入式系统是一个外延极广的名词。本书认为凡是结合产品功能的、含有嵌入式处理器的、运行嵌入式操作系统的、设计专门化的计算系统都叫嵌入式系统。一般而言,嵌入式系统的构架可以分成四个部分:处理器、存储器、IO 控制器和软件(包括操作系统和应用软件)。这样的定义更便于读者明确对嵌入式系统学习的目标、方向、内容等。一个嵌入式系统开发者,一般需要具有如下知识结构:

- 以嵌入式处理器为核心的知识体系:包括计算机组成与结构、具体嵌入式微处理器(如 ARM、MIPS 等)、数字电子技术、汇编语言程序设计、硬件描述语言等。

- 以嵌入式操作系统为核心的知识体系:操作系统原理、具体嵌入式操作系统(如 $\mu\text{C}/\text{OS-II}$ 、 μCLinux 等)、C/C++ 语言程序设计、数据结构等。

- 以产品应用为核心的知识体系:模拟电子技术、传感器技术、与应用相关的专业知识(如机械原理、医疗技术等)等。

- 以设计技术为核心的知识体系:硬件设计技术、软件设计技术、软硬件协同设计方法学、编译技术等。

本书的注解不在于对嵌入式系统定义的标准化,而是注重定义的通俗性、功能性、层次性和知识性,便于初学者依照自己的知识体系来理解和学习。

1.1.2 嵌入式系统特点

在了解了嵌入式系统定义的基础上,再进一步讨论其特点,使读者明确在嵌入式系统的设计与开发过程中,需要充分注意的事项。

从整体上看,嵌入式系统包括嵌入式处理器、定时器、控制器、存储器、传感器等一系

列微电子芯片和嵌入在存储器中的嵌入式操作系统、控制应用软件组成,实现诸如实时控制、监视、管理、移动计算、数据处理等各种自动化处理任务。以应用为中心,以微电子技术、控制技术、计算机技术和通信技术为基础,强调硬件软件协同性与整合性,软件与硬件可剪裁,以满足系统对功能、成本、体积和功耗等要求。

1. 嵌入性

嵌入式系统是嵌入到“特定对象体系”中的实体。因此,嵌入式系统通常不是应用对象的主体、只是辅助性的装置,要求其体积小、功耗低、抗干扰性好、成本低等。

2. 系统性

嵌入式系统是一个计算系统。因此,嵌入式系统会具有软硬件两方面,软件包括嵌入式操作系统和应用软件;硬件包括嵌入式处理器子系统、外部设备控制器等多种接口电路。

3. 智能性

嵌入式系统的核心是嵌入式处理器。每一个嵌入式系统至少含有一个嵌入式处理器,系统在嵌入式处理器的控制之下,运行嵌入式操作系统和应用程序,智能地实现预定的功能。

4. 专用性

嵌入式系统是为“特定对象体系”专门设计的。主要体现在嵌入式处理器上,称为“专用处理器”。如,语音处理系统、视频处理系统、电机控制系统的嵌入式处理器的功能是不同的,因此要熟悉嵌入式处理器类型、性能等。

5. 实时性

嵌入式系统是“特定对象体系”的一个模块。常作为控制、监视或辅助操作机器和设备的装置,因此,要具有很好的实时性,满足对象系统的要求。

6. 恰适性

嵌入式系统要恰好满足“特定对象体系”的需要。因此,对于一个成功的嵌入式系统设计,对其量体定制、软硬件都无冗余是最好的。不同的嵌入式系统,其资源也是不同的。就整体而言,比PC的资源大为减少。

7. 系统设备电子化

PC中的资源有缓存和内存(电子的)、外存(磁性的、光介质)、打印机(机电一体的)、显示器(LCD、CRT等)等,种类繁多。嵌入式系统的内存、外存都是电子的(通常使用Flash等),其特点可以被嵌入式处理器直接操作。

8. 编程语言低级化

嵌入式系统的编程语言更趋于低级语言,主要为汇编语言、C语言、C++语言等。

9. 操作系统实时化

嵌入式系统的一个很重要的特性是实时性,一般要求操作系统应该为实时操作系统(RTOS)。

10. 系统性能可测化

嵌入式系统要求具有软硬件可测试,保证系统无错误。

11. 开发系统专用化

嵌入式产品分为集成开发环境(IDE, Integrated Developement Environment, 也有人称为 Integration Design Environment、Integration Debugging Environment)、在线仿真器(ICE, In-Circuit Emulator)、实时操作系统(RTOS, Real Time Operating System)。ARM

芯片上都包含专用的在线调试电路,如 ARM 的 Embedded ICE,ARM 微处理器基于 JTAG 接口等。

1.2 嵌入式系统分类

根据不同的分类标准,嵌入式系统可有许多不同的分类方法。例如,可以按照嵌入式系统的复杂度分类、按照嵌入式系统的组成分类、按照嵌入式处理器的位数分类、按照嵌入式系统的实时性分类或按照嵌入式系统的应用领域分类等。

1. 按照嵌入式系统的复杂度分类

嵌入式系统的应用领域广泛,简单的有用 4 位单片机实现的电动玩具,复杂的有基于 32 位处理器的飞机飞行控制系统。按照其复杂程度,嵌入式系统可以分为简单嵌入式系统和复杂嵌入式系统。在实际应用中,简单嵌入式系统在数量上占有主要的市场。

简单嵌入式系统的软硬件复杂度都比较低。例如,常用的有 8 位或 16 位单片机系统,不使用操作系统,只有一个监控程序和应用程序。

复杂嵌入式系统的软硬件复杂度都比较高。通常,在硬件上需要使用 32 位微处理器,在软件上含有嵌入式操作系统,这类嵌入式系统集成度高、接口丰富、功能强大。

2. 按照嵌入式系统的实时性分类

嵌入式系统的重要特性就是实时性。按照其实时性要求,嵌入式系统可分为硬实时嵌入式系统、软实时嵌入式系统和非实时嵌入式系统。

(1) 硬实时系统 硬实时系统是指系统要确保在最坏情况下的服务时间,即对于事件响应时间的截止期限必须得到满足,而且其响应时间是可计算的。比如汽车、舰船、飞机、导弹、卫星的控制等就是这样的系统;通常,工业控制系统也是实时系统。

(2) 软实时系统 软实时系统就是那些从统计的角度来说,一个任务能够得到确保的处理时间,到达系统的事件也能够截止期限前得到处理,但违反截止期限并不会带来致命的错误。如手机就是一种软实时系统;MP3 等媒体播放器也是软实时系统。

(3) 非实时系统 系统对实时性没有明确的要求,也就是说没有事件响应时间的截止期限,对事件的响应完全由系统自身工作状态所决定。如 PDA、PC 等。

系统的实时性需要由硬件和软件共同作用来实现,硬件上通过中断来保证事件能够被实时感知,软件上通过实时操作系统来保证事件能够被实时响应。

3. 按照嵌入式处理器的位数分类

嵌入式处理器是嵌入式系统的硬件核心。嵌入式处理器一般可分为 8 位、16 位、32 位嵌入式处理器。那么,嵌入式系统一般也可根据所采用的嵌入式处理器分为 8 位、16 位、32 位嵌入式系统。

8 位和 16 位的嵌入式系统通常是以 8 位和 16 位微控制器(单片机)为核心构成的简单嵌入式系统,广泛地应用于控制系统、家用电器、汽车电子等。

32 位嵌入式系统通常是以 32 位微处理器或微控制器为核心构成的复杂嵌入式系统,广泛地应用于手机、消费电子、医疗电子、网络设备等。

4. 按嵌入式系统的组成分类

嵌入式系统是嵌入式计算机系统。按照其组成,嵌入式系统可分为嵌入式系统硬件和嵌

入式系统软件两大构建。

(1) 嵌入式系统硬件 嵌入式系统硬件是以嵌入式处理器为核心, 以丰富的功能接口部件为外围扩展层构建的。也正是基于这些丰富的外设接口, 才带来嵌入式系统越来越丰富的应用。外设接口部件一般是以其接口控制器芯片(如 USB 控制器等)为核心搭建的, 通过标准的扩展总线与嵌入式处理器实现连接。在嵌入式系统硬件设计中, 通常只要把处理器和控制器进行物理连接就可以实现外设接口扩展了。结构为: **嵌入式处理器** ↔ **片外控制器**

嵌入式处理器又以处理器核(如 ARM 核)为核心, 片内集成适当的接口控制器(称为片内外设)构成的。随着嵌入式处理器高度集成化技术的发展, 使得片内可以实现的控制器的数量越来越多, 功能也越来越强。例如, 基于 ARM 核处理器 S3C44B0X 的内部就封装了 Cache、I2C、UART、LCD 控制器等十几种片内外设。在嵌入式系统设计中, 所需要的外围接口电路越来越少。结构为: **嵌入式处理器核** ↔ **片内控制器**

(2) 嵌入式系统软件 嵌入式系统软件一般来说是由嵌入式操作系统和应用软件两部分组成的。本书中的嵌入式系统软件可以分成启动代码(BootLoader)、操作系统内核与底层驱动、文件系统与应用程序等几部分。

BootLoader 是嵌入式系统的启动代码, 主要用来初始化处理器、必须使用的控制器、传递内核启动参数给嵌入式操作系统内核, 使得内核可以按照所设定的参数要求启动。另外, BootLoader 通常都具有从 Flash 搬运内核代码到 RAM 并跳转到内核代码地址运行的功能。操作系统内核则主要有 4 个任务: 进程管理、进程间通信与同步、内存管理及 I/O 资源管理。底层驱动程序主要提供给上层应用程序, 是处理器、外设接口控制器和外部设备进行通信的一个媒介。文件系统则可以让嵌入式软件工程师灵活方便地管理系统。应用程序才是真正针对需求的、可以是嵌入式软件工程师完全自主开发的。

总的来说, 嵌入式系统的硬件是系统的基石, 嵌入式系统软件则是在这个基石上面镌刻出来的工艺品。对于任何一个需求明确的嵌入式系统来说, 两者缺一不可。在对系统做了相对完整而细致的需求分析之后, 通常采用软件和硬件基本同步进行的方式来开发, 前期硬件系统的设计要比软件系统设计稍微提前, 到了后期软件系统的开发工作量会比硬件系统的开发工作量大一些。

1.3 嵌入式系统组成结构

嵌入式系统一般由硬件平台和软件平台两部分组成。其中, 硬件平台由嵌入式处理器、外设控制器芯片和硬件设备组成; 而软件平台由 BootLoader、嵌入式操作系统、驱动程序和应用软件组成。

随着芯片技术的不断发展, 嵌入式处理器的主频也越来越高, 多处理器、多核处理器平台也逐渐应用在嵌入式领域。嵌入式系统的组成也将由单核、低频设计进入多核、高频设计。

1. 典型嵌入式系统的组成结构

图 1.1 完整地描述了典型嵌入式系统的硬件和软件各部分的组成结构。从硬件上大致可分为: 处理器内核层、芯片级外设层、用户级外设层; 从软件上大致可分为: 启动层、操作系统层、应用层。对于一个具体的嵌入式系统, 图 1.1 中的各种资源并不一定全部使用、