

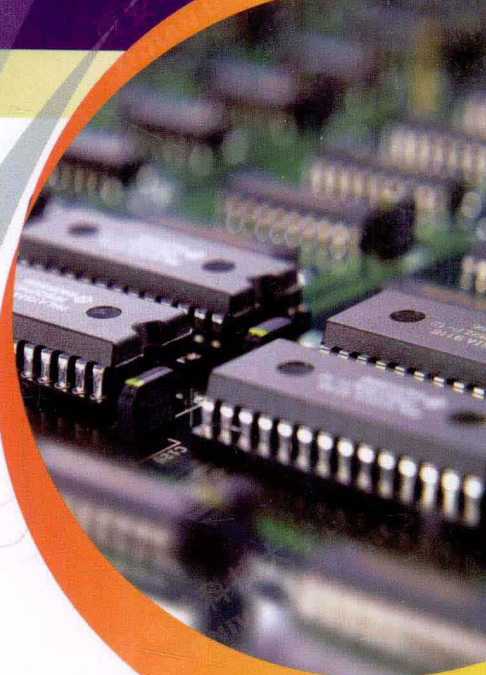


高等学校本科规划教材

嵌入式系统原理与开发

(第二版)

夏靖波 陈雅蓉 编著
胡健生 王 航



西安电子科技大学出版社
<http://www.xduph.com>

面向 21 世纪高等学校本科规划教材

嵌入式系统原理与开发

(第二版)

夏靖波 陈雅蓉 胡健生 王航 编著

西安电子科技大学出版社

内 容 简 介

本书根据嵌入式系统的结构体系, 从其所涉及的基本概念出发, 分层次介绍了嵌入式系统的设计原理, 并保持了各层次之间的相关性。嵌入式系统从结构上划分为嵌入式处理器、嵌入式外围硬件、设备驱动、实时操作系统、嵌入式应用程序这五个层次。本书在阐述每个层次原理的基础上, 还基于 ARM 处理器和 $\mu\text{C}/\text{OS-II}$ 操作系统平台, 介绍了一个完整的嵌入式系统开发所涉及的基本技术方法、开发工具、调试原理与方法。

本书可作为计算机网络工程、电子工程、自动化控制等相关专业本科生或研究生的教材, 也可作为从事嵌入式系统开发的工程技术人员的参考书。

图书在版编目 (CIP) 数据

嵌入式系统原理与开发 / 夏靖波等编著. — 2 版

—西安: 西安电子科技大学出版社, 2010.8

ISBN 978-7-5606-2433-4

I. ① 嵌… II. ① 夏… III. ① 微型计算机—系统开发 IV. ① TP360.21

中国版本图书馆 CIP 数据核字 (2010) 第 099833 号

策 划 云立实

责任编辑 阎 彬 云立实

出版发行 西安电子科技大学出版社 (西安市太白南路 2 号)

电 话 (029)88242885 88201467 邮 编 710071

网 址 www.xduph.com 电子邮箱 xdupfxb001@163.com

经 销 新华书店

印刷单位 西安文化彩印厂

版 次 2010 年 8 月第 2 版 2010 年 8 月第 4 次印刷

开 本 787 毫米 $\times$ 1092 毫米 1/16 印张 20.75

字 数 490 千字

印 数 10 001~13 000 册

定 价 30.00 元

ISBN 978-7-5606-2433-4 / TP·1214

XDUP 2725002-4

*** 如有印装问题可调换 ***

本社图书封面为激光防伪覆膜, 谨防盗版。

前 言

《嵌入式系统原理与开发(第一版)》自 2006 年由西安电子科技大学出版社出版以来,已被全国近 30 所高等院校的相关专业选为授课教材。与该教材配套开发的网络课程在 2008 年全军电教教材评比中获得网络课程类一等奖;在 2009 年全国第十三届多媒体教育软件大奖赛中获得网络课程组一等奖。

随着信息化、智能化、网络化的持续深入,嵌入式技术已成为信息产业中发展最快、应用最广的计算机技术之一。本版我们根据嵌入式技术的最新发展动态,结合近年来的开发实践与教学体会,修订了第一版中的一些错误,更新了部分数据;为了更好地辅助读者学习,在第 4 章增加了 4.3 节设备驱动控制实例;新增加了附录,内容包括嵌入式专业词汇与缩略语、ARM 指令集列表、Thumb 指令集列表、汇编预定义变量及伪指令、 $\mu\text{C}/\text{OS-II}$ 内核函数。

在本教材的再版修订过程中,得到了空军工程大学电讯工程学院网络工程系张锐讲师、李娜博士等很多同志的帮助和支持,在此表示感谢。

如在本教材的使用和学习方面存在问题,欢迎与作者联系。作者的电子邮箱为 jbxiaad@sina.com。

作 者

2010 年 5 月

第一版前言

嵌入式技术的飞速发展是从 20 世纪 80 年代开始的。推动其发展的主要原因是计算机的应用领域出现了进一步的拓展。尤其是在 20 世纪末和 21 世纪初,嵌入式技术与信息技术和网络技术相结合,拓展出许多新的应用领域,从而成为嵌入式技术发展的强大动力。目前,嵌入式系统被广泛应用于消费电子、通信网络、工业控制、汽车电子、医疗仪器、信息家电、航空航天、军事等众多领域。在全球工作的计算机中,有 95% 以上都是嵌入式计算机。在处理器方面,面向嵌入式计算的处理器的产量要远远大于面向通用计算的处理器的产量。嵌入式系统无处不在,例如,手机、MP3 播放器、微波炉、DVD 机、ADSL Modem、数码相机/摄像机、车载的 GPS 导航器等产品内部都包含有嵌入式系统。

回顾嵌入式技术的发展过程,早期的嵌入式系统被认为是人们熟知的单片机系统。通常,这种系统以 8 位或 16 位 MCU 为核心,配合一些简单的传感器、伺服控制、显示设备等,实现一定的测量、信息处理、自动控制和显示等功能。系统的组网通常利用 CAN、RS-232、RS-485 等总线的方式,通信协议比较单一,一般孤立于 Internet 之外。随着嵌入式技术与网络技术的广泛结合,手机、PDA、路由器、Modem 等复杂的高端嵌入式应用不断涌现。这使得人们需要设计更加复杂的嵌入式系统。这种需求推动了微电子技术的发展,使得 32 位处理器被广泛使用。基于 32 位的处理器,可以方便地运行嵌入式操作系统;而基于嵌入式操作系统,人们又可以开发复杂度更高的嵌入式软件。目前,我们所提到的嵌入式系统通常是基于 32 位嵌入式处理器,使用了嵌入式操作系统的平台。随着嵌入式应用的不断普及和深入,嵌入式技术的发展趋势包含了嵌入式行业标准化、SOC 成为应用主流、嵌入式开发工具更加强大、嵌入式软件的复杂度越来越高、面向网络的嵌入式系统成为必然、建立嵌入式计算的新型模型等若干方面。

在多年的教学实践中,我们认识到高校开设嵌入式教学课程的必要性和紧迫性。当前,我国掌握嵌入式技术的人才十分匮乏,这集中体现在高校课程设置的企事业单位的人才需求存在着突出的矛盾。由于和单片机教学相比,32 位嵌入式系统的软、硬件开发的复杂度急剧加大,因此学生们必须进行更高层次的学习。

现在有很多高校已经或正在开设这方面的课程,但就总体而言我们认为必须从理论和实践两个环节同时入手,既要培养学生的理论知识,又要培养学生的实践能力,还要注重培养学生的创新意识,才能造就合格的嵌入式人才。

本书的特点是将嵌入式系统的开发和原理的介绍相结合,难度适中。读者只要掌握微机原理、电路原理、C 语言程序设计、操作系统原理、网络协议等方面的一些基础知识就可以进行学习。

经讨论和实践证明,获得公认的最适合于嵌入式教学的架构是“ARM+ μ C/OS-II (Linux)”。本书在教学内容的安排上,也采用了这种架构。本书共分为 9 章,从内容上可

分为 5 个部分, 其中:

- 第一部分为嵌入式系统概述, 包括第 1 章, 主要介绍嵌入式系统的概念、应用领域和发展趋势。
- 第二部分为嵌入式系统的硬件开发技术, 包括第 2~4 章。第 2 章以 ARM 处理器为例, 介绍了嵌入式处理器的基础知识, 重点从基于处理器编程的角度学习如何掌握嵌入式处理器的使用。第 3 章主要介绍嵌入式系统的外围硬件, 包括总线系统、存储设备、I/O 设备、通信设备和附属电路等。第 4 章介绍基于硬件平台的启动程序和驱动程序设计, 通常这部分内容被划分在硬件设计范畴。
- 第三部分为嵌入式系统的软件程序设计, 包括第 5~7 章。第 5 章以 $\mu\text{C}/\text{OS-II}$ 实时操作系统为例, 介绍了嵌入式操作系统的基础知识。第 6 章讲述嵌入式应用程序的设计方法, 内容包括程序设计的基本方法、基本技术以及嵌入式程序的优化。第 7 章介绍基于网络的嵌入式系统设计, 内容包括分布式嵌入式系统的设计技术、几种常用的嵌入式网络的设计方法以及基于网络的设计实例分析。
- 第四部分为嵌入式系统的开发工具和调试方法, 包括第 8 章, 主要以 ARM 的开发和调试为例进行介绍。
- 第五部分为系统设计技术, 包括第 9 章, 主要从系统设计的高度介绍嵌入式系统的设计流程、设计过程中使用的技术以及需要注意的问题。

建议课时安排如下:

第一部分: 嵌入式系统的基础知识, 包括第 1 章。建议课时: 3 h。

第二部分: 嵌入式系统的硬件开发技术, 包括第 2~4 章。建议课时: 20 h(第 2 章 10 h, 第 3 章 6 h, 第 4 章 4 h)。

第三部分: 嵌入式系统的软件程序设计, 包括第 5~7 章。建议课时: 21 h(第 5 章 10 h, 第 6 章 3 h, 第 7 章 8 h)。

第四部分: 嵌入式系统的开发工具和调试方法, 包括第 8 章。建议课时: 6 h。

第五部分: 嵌入式系统的设计方法, 包括第 9 章。建议课时: 8 h。

以上的内容安排可以根据各学校的具体情况在教学时有所取舍。

为便于教学, 编者为一本书配备了多媒体教学课件, 并可提供与本书配套的嵌入式教学实验系统, 联系方式为: jbxia@sina.com。

本书在编写过程中得到了空军工程大学电讯工程学院和西安电子科技大学出版社的支持。李娜、赵锡臻、李德峰、武兆雪等研究生为本书的编写付出了辛勤劳动, 编者对此深表感谢。

由于编者知识所限, 书中存在的错误和不足之处在所难免, 在此真诚地欢迎读者提出宝贵意见和建议。

编者

2005 年 10 月

目 录

第 1 章 嵌入式系统概述..... 1	第 3 章 嵌入式硬件平台..... 87
1.1 嵌入式系统的产生和发展..... 1	3.1 引言..... 87
1.2 嵌入式系统的概念..... 3	3.2 嵌入式硬件平台概述..... 87
1.2.1 嵌入式系统的定义..... 3	3.3 总线..... 88
1.2.2 嵌入式系统的特点..... 3	3.3.1 总线协议..... 89
1.2.3 嵌入式系统的组成结构..... 6	3.3.2 DMA..... 91
1.2.4 嵌入式系统的分类..... 7	3.3.3 总线配置..... 92
1.3 嵌入式系统的应用领域..... 11	3.3.4 总线实例..... 93
1.4 嵌入式系统的发展趋势..... 12	3.4 存储设备..... 93
思考与练习题..... 15	3.4.1 嵌入式系统存储器子系统的结构..... 93
第 2 章 嵌入式处理器..... 16	3.4.2 RAM..... 95
2.1 引言..... 16	3.4.3 ROM..... 97
2.2 嵌入式处理器概述..... 16	3.4.4 Flash Memory..... 98
2.2.1 嵌入式处理器的分类..... 17	3.5 I/O 设备..... 99
2.2.2 典型的嵌入式处理器..... 18	3.5.1 定时器/计数器..... 99
2.3 ARM 处理器基础..... 22	3.5.2 模/数转换器和数/模转换器..... 100
2.3.1 ARM 简介..... 22	3.5.3 人机接口设备..... 101
2.3.2 ARM 处理器系列..... 23	3.6 通信设备..... 104
2.3.3 ARM 处理器体系结构..... 29	3.6.1 通用异步收发器..... 104
2.3.4 ARM 处理器应用选型..... 34	3.6.2 USB 设备..... 107
2.4 ARM 指令系统..... 35	3.6.3 Ethernet 设备..... 112
2.4.1 ARM 编程模型..... 35	3.7 其他..... 113
2.4.2 ARM 寻址方式..... 45	3.7.1 电源..... 113
2.4.3 ARM 指令集..... 49	3.7.2 时钟..... 114
2.4.4 Thumb 指令集..... 63	3.7.3 复位..... 115
2.5 ARM 程序设计基础..... 73	3.7.4 中断..... 116
2.5.1 ARM 汇编语句格式..... 74	思考与练习题..... 118
2.5.2 ARM 汇编程序格式..... 74	第 4 章 BootLoader 与设备驱动..... 119
2.5.3 汇编语言编程实例..... 75	4.1 引言..... 119
2.5.4 汇编语言与 C 语言的混合编程..... 77	4.2 BootLoader..... 120
思考与练习题..... 84	4.2.1 BootLoader 概述..... 120

4.2.2 BootLoader 的典型结构	121	5.6.2 $\mu\text{C}/\text{OS-II}$ 的移植步骤	177
4.2.3 实例分析	124	5.6.3 测试移植代码	182
4.3 设备驱动	135	5.7 基于 $\mu\text{C}/\text{OS-II}$ 构建的 TCP/IP/PPP	
4.3.1 设备驱动概述	135	协议栈	182
4.3.2 LCD 驱动控制实例	139	5.7.1 嵌入式协议栈概述	182
4.3.3 A/D 转换功能驱动实例	144	5.7.2 选择协议栈	183
思考与练习题	148	5.7.3 嵌入式 TCP/IP 协议栈	183
第 5 章 嵌入式操作系统	149	5.7.4 基于 $\mu\text{C}/\text{OS-II}$ 的 TCP/IP/PPP	
5.1 引言	149	协议栈	185
5.2 嵌入式操作系统概述	150	思考与练习题	187
5.2.1 嵌入式实时操作系统	150	第 6 章 嵌入式应用程序设计	188
5.2.2 典型的嵌入式操作系统	150	6.1 引言	188
5.3 操作系统的基本概念	155	6.2 程序设计方法	188
5.3.1 多进程和多线程	155	6.2.1 设计范型	188
5.3.2 任务	155	6.2.2 编程模型	191
5.3.3 任务切换	157	6.3 程序设计技术	192
5.3.4 内核	157	6.3.1 消息	192
5.3.5 任务调度	157	6.3.2 任务和任务间同步	193
5.3.6 任务间的通信与同步	159	6.3.3 绘图函数	194
5.3.7 操作系统的结构和功能	162	6.3.4 控件	195
5.4 $\mu\text{C}/\text{OS-II}$ 简介	162	6.4 嵌入式程序优化	196
5.4.1 $\mu\text{C}/\text{OS-II}$ 概述	162	6.4.1 编译过程	196
5.4.2 $\mu\text{C}/\text{OS-II}$ 的特点	163	6.4.2 执行时间优化	200
5.4.3 $\mu\text{C}/\text{OS-II}$ 的软件体系结构	164	6.4.3 能量优化	204
5.5 $\mu\text{C}/\text{OS-II}$ 内核结构	165	6.4.4 长度优化	205
5.5.1 临界段	165	思考与练习题	207
5.5.2 任务	166	第 7 章 基于网络的嵌入式系统设计	208
5.5.3 任务控制块 OS_TCB	167	7.1 引言	208
5.5.4 任务调度	168	7.2 分布式嵌入式系统	208
5.5.5 任务管理	168	7.2.1 系统概述	208
5.5.6 中断服务	169	7.2.2 OSI 模型	209
5.5.7 时钟节拍与时间管理	170	7.2.3 网络结构	210
5.5.8 任务间同步与通信的管理	171	7.2.4 网络编程	210
5.5.9 内存管理	174	7.2.5 设计技术	212
5.5.10 $\mu\text{C}/\text{OS-II}$ 的初始化	175	7.3 嵌入式系统网络	214
5.5.11 $\mu\text{C}/\text{OS-II}$ 的启动	176	7.3.1 I ² C 总线	214
5.6 $\mu\text{C}/\text{OS-II}$ 在 ARM 上的移植	176	7.3.2 CAN 总线	217
5.6.1 $\mu\text{C}/\text{OS-II}$ 的移植条件	176		

7.3.3 Ethernet.....	220	思考与练习题	266
7.3.4 GPRS	221		
7.3.5 蓝牙技术	224	第 9 章 系统设计技术	267
7.3.6 Internet.....	227	9.1 引言	267
7.4 基于网络的设计示例	229	9.2 嵌入式系统的开发过程和设计流程	267
7.4.1 远程温度检测系统	229	9.2.1 开发过程	267
7.4.2 基于 VoIP 和蓝牙的无线电话系统	232	9.2.2 设计流程	271
7.4.3 机顶盒	235	9.3 系统设计的形式化方法	274
思考与练习题	237	9.3.1 UML 简介	274
第 8 章 嵌入式系统开发	239	9.3.2 结构描述	277
8.1 引言	239	9.3.3 行为描述	282
8.2 嵌入式系统的开发工具	239	9.4 需求分析与规格说明	284
8.2.1 开发工具概述	239	9.4.1 需求分析	284
8.2.2 ARM ADS 简介	240	9.4.2 规格说明	284
8.2.3 使用 ADS 创建工程	244	9.5 系统分析与体系结构设计	287
8.2.4 使用 AXD 调试代码.....	251	9.6 质量保证	288
8.3 嵌入式系统的调试方法	254	思考与练习题	290
8.3.1 嵌入式系统调试简介	254		
8.3.2 ARM 调试系统概述	254	附录 A 嵌入式专业词汇与缩略语.....	291
8.3.3 基于 Angel 的调试系统	256	附录 B ARM 指令集列表.....	300
8.3.4 基于 JTAG 的调试系统.....	260	附录 C Thumb 指令集列表	303
8.4 其他实用工具	262	附录 D 汇编预定义变量及伪指令	306
8.4.1 Source Insight.....	262	附录 E μ C/OS-II 内核函数	311
8.4.2 SkyEye.....	263	参考文献	321

第 1 章

嵌入式系统概述

1.1 嵌入式系统的产生和发展

嵌入式系统起源于微型计算机时代。20 世纪 70 年代,微处理器的出现使得计算机发生了历史性的变化,以微处理器为核心的微型计算机走出机房,深入到千家万户。这一时期被人们称为 PC 时代。

由微型机不断强大的计算能力所表现出来的智能化水平,人们首先想到的就是将其用于自动控制领域中,例如将微机配置相应的外围接口电路后用于实现对电厂发电机的状态监测与工作控制。然而更多的场合要求将计算机嵌入到对象体系中,实现对象体系的智能化控制,例如飞机、舰船的自动驾驶,洗衣过程的自动化,汽车的自动点火、自动刹车等。在如此众多的应用背景下,这类计算机便逐渐失去了原来的形态和通用的计算功能,从而成为一种嵌入到对象体系中,实现对象体系智能化控制的计算机,我们称之为嵌入式计算机系统,简称嵌入式系统。

分析嵌入式计算机系统的产生背景,可以发现它与通用计算机系统有着完全不同的技术要求和技術发展方向。通用计算机系统要求的是高速、海量的数值运算,在技术发展方向上追求总线速度的不断提升、存储容量的不断扩大。而嵌入式计算机系统要求的是对象体系的智能化控制能力,在技术发展方向上追求针对特定对象系统的嵌入性、专用性和智能化。这种技术发展的分歧导致 20 世纪末计算机进入了通用计算机系统和嵌入式计算机系统两大分支并行发展的时期。这一时期被人们称为后 PC 时代。

嵌入式系统能够走上独立发展道路的一个重要的技术支撑来源于现代微电子技术的发展。在大多数应用背景下,微型机的体积、价格、可靠性都无法满足对象体系的嵌入式要求。在这种情况下,人们需要将计算机系统全部或部分地集成到一个芯片中,实现计算机的芯片化,这就是通常所说的单片机。世界上最早的单片机是 1976 年 Intel 公司开发的 8048,几年后 Intel 公司又在它的基础上开发出了著名的 8051,同时期的还有 Motorola 公司的 68HC05、Zilog 公司的 Z80 等。Philips 公司又进一步将 MCS-51 发展成为 MCU(Micro Controller Unit,微控制器)。单片机一般都集成了 8/16 位微处理器、RAM、ROM、串口、并口、定时器、ADC/DAC、看门狗、PWM 定时器、中断控制器等。用户在单片机的外围增加若干接口电路、设计相应的控制程序就可以实现多种应用。单片机被广泛用于消费电子、医疗电子、智能控制、通信、仪器仪表、工业控制、安全报警、交通管理等各领域。

根据应用的需求,单片机的发展方向主要有两个:一个是提升 CPU 的性能,如提高微处理器的运行速度、降低芯片的功耗等;另一个是扩充各种功能,把各种不同的外围设备继续向芯片内部集成,并与成本相结合,衍生出面向不同应用的各种型号。单片机发展的结果被称为 MCU,通常根据微处理器的字长划分成 8 位、16 位、32 位、64 位机。

在嵌入式系统应用领域不断发展的过程中,伴随着大量需要进行数字信号处理的应用,如数字音频处理、数字图像处理、信号变换、数字滤波、谱分析等。数字信号处理(Digital Signal Processing, DSP)在 20 世纪 60~70 年代主要处于理论研究阶段。进入 80 年代,随着微电子技术的发展,出现了 DSP 器件,这些器件的出现使得各种 DSP 的算法得以实现,使 DSP 从仅限于理论研究进而推广到实际应用。1983 年,美国 TI 公司推出 TMS320 系列的第一个产品,标志着实时数字信号处理领域的重大突破。DSP 技术的迅速发展,使其应用范围拓展到通信、控制、计算机等领域。

20 世纪 80 年代后,嵌入式系统的另一个发展来源得益于软件技术的进步,一方面体现在编程语言上,另一方面体现在实时操作系统的使用上。在微处理器出现的初期,为了保证嵌入式软件的空间和时间效率,只能使用汇编语言进行编程。随着微电子技术的进步,系统对软件时空效率的要求不再十分苛刻,从而使得嵌入式软件可以使用 C 语言等高级语言。高级编程语言的使用提高了软件的生产效率,保障了软件的可重用性,缩短了软件的开发周期。另外,嵌入式系统大多是实时系统,对于复杂的嵌入式系统而言,除了需要高级语言开发工具外,还需要嵌入式实时操作系统的支持。20 世纪 80 年代初,一些软件公司开始推出商业嵌入式实时操作系统和各种开发工具,如 Ready System 公司的 VRTX 和 XRAY, Integrated System Incorporation(ISI)的 pSOS 和 PRISM+, WindRiver 公司的 VxWorks 和 Tornado, QNX 公司的 QNX 等。这些嵌入式实时操作系统都具有强实时、可裁剪、可配置、可扩充、可移植的特点,支持主流的嵌入式微处理器。开发工具包括了各种面向软、硬件开发的工具,如硬件仿真器、源码级的交叉调试器等。嵌入式软件工程师开始使用操作系统来开发自己的软件。嵌入式操作系统的出现和推广带来的最大好处就是可以使嵌入式产业走向协同开发和规模化发展的道路,从而促使嵌入式应用拓展到更加广阔的领域。

嵌入式系统的全面发展是从 20 世纪 90 年代开始的,主要被分布式控制、数字化通信、信息家电、网络等应用需求所牵引。现在,人们可以随处发现嵌入式系统的应用,如手机、MP3 播放器、数码相机、VCD、机顶盒、路由器、交换机等。嵌入式系统在软、硬件技术方面迅速发展:首先,面向不同应用领域、功能更加强大、集成度更高、种类繁多、价格低廉、低功耗的 32 位微处理器逐渐占领统治地位,DSP 器件向高速、高精度、低功耗发展,而且可以和其他的嵌入式微处理器相集成;其次,随着微处理器性能的提高,嵌入式软件的规模也呈指数型增长,所体现出的嵌入式应用具备了更加复杂和高度智能的功能,软件在系统中体现出来的重要程度越来越大,嵌入式操作系统在嵌入式软件中的使用越来越多,所占的比例逐渐提高,同时,嵌入式操作系统的功能不断丰富,在内核基础上发展成为包括图形接口、文件、网络、嵌入式 Java、嵌入式 CORBA、分布式处理等完备功能的集合;最后,嵌入式开发工具更加丰富,已经覆盖了嵌入式系统开发过程的各个阶段,现在主要向着集成开发环境和友好人机界面等方向发展。

1.2 嵌入式系统的概念

1.2.1 嵌入式系统的定义

嵌入式计算系统简称为嵌入式系统。那么究竟什么是嵌入式系统呢？

在 Wayne Wolf 所著的一本有关嵌入式系统设计的教科书《嵌入式计算系统设计原理》里这样定义：“不严格地说：它是任意包含一个可编程计算机的设备，但是这个设备不是作为通用计算机而设计的。因此，一台个人电脑并不能称之为嵌入式计算系统，尽管个人电脑经常被用于搭建嵌入式计算系统。”

IEEE(国际电气和电子工程师协会)的定义是：“Device used to control, monitor, or assist the operation of equipment, machinery or plants”。

微机学会的定义是：“嵌入式系统是以嵌入式应用为目的的计算机系统”，并将其分为系统级、板级、片级。系统级包括各类工控器、PC104 模块等；板级包括各类带 CPU 的主板和 OEM 产品；片级包括各种以单片机、DSP、微处理器为核心的产品。

目前被大多数人接受的一般性定义是：“嵌入式系统是以应用为中心，以计算机技术为基础，软、硬件可裁剪，适应应用系统对功能、可靠性、成本、体积和功耗等严格要求的专用计算机系统。”

由于嵌入式系统的概念从外延上很难统一，其应用形式多种多样，因此定义嵌入式系统非常困难。不过，通过分析上述定义后不难发现，从嵌入式系统概念的内涵上讲，它的共性是一种软、硬件紧密结合的专用计算机系统。通常我们所说的嵌入式系统，硬件以嵌入式微处理器为核心，集成存储系统和各种专用输入/输出设备；软件包含系统启动程序、驱动程序、嵌入式操作系统、应用程序等，将这些软件有机结合，便可构成系统特定的一体化软件。这种专用计算机系统必然在可靠性、实时性、功耗、可裁剪等方面具有一系列特点。如果我们关注一下嵌入式系统的特性，也许能够对嵌入式系统的概念有更深入的理解。

1.2.2 嵌入式系统的特点

由于嵌入式系统面向的是专业领域中工作在特定环境下的应用系统，不同于通用计算机系统应用的多样性和普遍适应性，因此具有以下特点。

1. 专用的计算机系统

嵌入式系统通常面向特定任务，是专用的计算机系统。整个系统的设计必须满足具体的应用需求，一旦任务变更，整个系统将很可能需要重新设计。这种“量体裁衣”型的专用计算平台与通用计算平台有很大的不同，主要表现在如下几个方面：

(1) 形式多样。在共同的基本计算机系统架构上，针对不同的应用领域，嵌入式系统的构造不尽相同，处理器、硬件平台、操作系统、应用软件等种类繁多。不同的嵌入式微处理器的体系结构和类型的适应面不同。嵌入式系统工业成为不可垄断、高度分散的工业。

(2) 对运行环境的依赖性。在众多应用背景下，温度、湿度、震动、干扰、辐射等因素构成了嵌入式系统赖以生存的环境，因此在系统设计时就需要充分考虑其运行环境的各种因素。

(3) 综合考虑成本、资源、功耗、体积因素。这些原本在基于通用计算平台进行系统设计时无需考虑或无需过多考虑的因素，在基于专用计算平台的系统设计中都需要充分考虑。对于大量的消费类数字化产品，成本是影响产品竞争力的关键因素之一。为了节省成本，就必须精简使用和合理利用资源。进一步讲，在很多情况下，由于环境、功耗、体积等因素的存在，系统能够使用的资源可能会受到限制。例如，对于需要考虑抗震的系统，最好不要采用硬盘存储。考虑功耗的原因一方面是因为系统本身紧张的电能供给，如便携式设备和电池供电的设备；另一方面是系统散热的问题，如环境或体积的因素使得无法采用风扇散热。对于航空电子设备，在体积、重量等方面的因素往往成为一项决定性的系统约束。成本、资源、功耗、体积等因素往往相互关联，设计时必须权衡轻重，这也增加了系统设计的难度。

(4) 软、硬件紧密结合，高效设计。嵌入式微处理器与通用微处理器的最大区别在于每种嵌入式微处理器大多专用于某种或几种特定应用，工作在为特定用户群设计的系统中。它通常具有功耗低、体积小、集成度高等特点。把通用微处理器中许多由板卡完成的功能集成在芯片内部，有利于嵌入式系统设计小型化，增强移动能力，增强与网络的耦合度。嵌入式软件是应用程序和操作系统两种软件的一体化程序。对于嵌入式软件而言，系统软件和应用软件的界限并不明显，原因在于嵌入式环境下应用系统的配置差别较大，所需操作系统的裁剪配置不同，I/O 操作没有标准化，驱动程序通常需要自行设计。最终，这种不同配置的操作系统和应用程序被连接编译成统一运行的软件系统。这些过程都需要在系统总体设计目标指引下综合设计和实现，因此整个系统是高效设计的。

2. 代码固化

嵌入式系统的目标代码通常固化在非易失性存储器中，如 ROM、EPROM、EEPROM、Flash 等。这样做一方面是因为系统资源受限，而更主要的原因是为了提高系统的执行速度和系统的可靠性。

3. 实时性要求

嵌入式系统大多有实时性要求，根据系统对实时性要求的强度的不同，嵌入式系统可分为硬实时系统和软实时系统。

4. 可靠性要求

嵌入式系统一般要求具有出错处理和自动复位功能，特别是对于运行在极端环境下或重要场合的嵌入式系统而言，其可靠性设计尤其重要。在嵌入式系统设计中一般要使用一些硬件和软件机制来保证系统的可靠性，如硬件的看门狗定时器、软件的内存保护和重启机制等。

5. 操作系统的支持

嵌入式软件可以在没有操作系统支持的情况下进行设计，但是随着系统功能的复杂程度和性能要求的进一步提高，需要采用多任务结构设计软件。为了合理地进行任务调度、

利用系统资源以及各种函数接口,必须使用嵌入式操作系统平台开发软件。嵌入式系统在嵌入式操作系统的支持下设计,一方面可以保证程序执行的实时性和可靠性,另一方面可以有效地减少开发周期,保障软件质量。嵌入式操作系统是嵌入式行业走向标准化道路的基础。

嵌入式操作系统通常支持多种类型的处理器,并且与通用操作系统相比具有体积小、可裁剪、实时性好、可靠性高、可固化等特点。与嵌入式微处理器相同,嵌入式操作系统也具有专用性的特点,并且通过不同的裁剪和配置,可适应不同的应用背景。因此,嵌入式操作系统也是多种多样的。

6. 专门的开发工具、环境和方法

由于嵌入式系统是软/硬件紧密结合的系统,因此嵌入式系统的开发通常是软件与硬件并行设计、开发的过程,软、硬件协同设计的开发方法是最适合嵌入式系统开发的方法。其开发过程一般分为以下几个阶段:系统定义、软件与硬件的设计与实现、软/硬件集成、系统测试、可靠性评估等。嵌入式系统在可靠性方面的特点使得系统测试和可靠性评估非常重要,在这方面的方法研究已经发展成为一门学科。

由于系统资源有限,嵌入式系统一般不具备自主开发能力,设计完成后,用户也不能对其中的软件进行修改,即嵌入式系统的开发必须借助于一套专门的开发工具(开发环境),包括设计、编译、调试、测试等工具,采用交叉开发的方式进行。交叉开发环境由宿主机和目标机组成,如图 1-1 所示。宿主机一般采用通用计算机系统,利用通用计算机的丰富资源,来承担开发工具的大部分工作,构成主要的开发环境;目标机就是所要开发的嵌入式系统,构成最终的执行环境,配合宿主机完成开发工作。

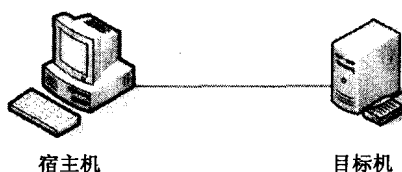


图 1-1 交叉开发环境

目前的嵌入式微处理器基本上都在片上集成了专用的调试电路,如 ARM 的 Embedded ICE。片上调试电路成为嵌入式开发的必要条件之一,为嵌入式系统的调试提供了方便的解决方案。综合而言,这种解决方案更有利于嵌入式系统的经济性,因此嵌入式微处理器包含专用调试电路已成为嵌入式开发的一个特点。

7. 知识集成系统

嵌入式系统是先进的计算机技术、半导体技术、电子技术、通信网络技术以及各个应用领域的专用技术相结合的产物。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。嵌入式系统的广泛应用和巨大的发展潜力已使它成为 21 世纪 IT 技术发展的热点之一。

从某种意义上讲,通用计算机行业的技术是垄断的。占整个计算机行业 90% 的 PC 产业,80% 采用 Intel 公司的 8x86 体系结构,芯片基本上出自 Intel、AMD、Cyrilx 等几家公司。在几乎每台计算机必备的操作系统和办公软件方面,Microsoft 公司的 Windows 及 Office 约

占 80%~90%，并且凭借操作系统还可以搭配其他应用程序。因此，当代的通用计算机行业的基础被认为是由 Wintel(Microsoft 和 Intel 20 世纪 90 年代初建立的联盟)垄断的行业。

嵌入式系统则不同，嵌入式行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。它是一个高度分散的行业，充满了竞争、机遇和创新，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，各不相同的应用领域也决定了不可能由少数公司、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是高度分散的，留给各个行业的中小规模高技术公司的创新余地很大。并且，社会上的各个应用领域是在不断向前发展的，要求其中的嵌入式微处理器核心也同步发展，这也构成了推动嵌入式行业发展的强大动力。

另外，通用计算机的开发人员一般是计算机科学或计算机工程方面的专业人士，而嵌入式系统则要和各种不同行业的应用相结合，要求更多的计算机以外的专业知识，其开发人员往往以各个应用领域的专家为主。当然，这也要求开发工具具有易学、易用、可靠、高效的特点。

1.2.3 嵌入式系统的组成结构

在不同的应用场合，嵌入式系统虽然呈现出不同的外观和形式，但是其核心的计算系统仍然可以抽象出一个典型的组成模型。嵌入式系统的组成结构一般可划分为硬件层、中间层、软件层和功能层，如图 1-2 所示。

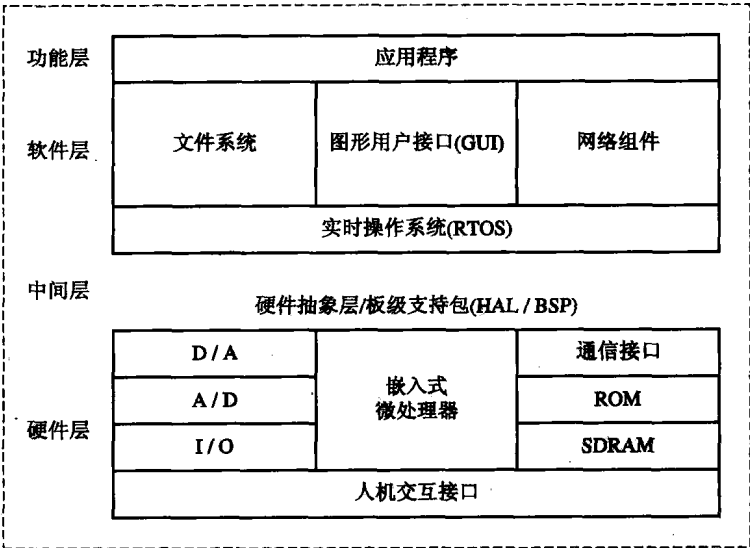


图 1-2 嵌入式系统的组成结构

1. 硬件层

硬件层由嵌入式微处理器、存储系统、通信接口、人机交互接口、其他 I/O 接口(A/D、D/A、通用 I/O 等)以及电源等组成。嵌入式系统的硬件层以嵌入式微处理器为核心，附带有电源电路、时钟电路和存储器电路(RAM 和 ROM 等)。它们共同构成了一个嵌入式核心控制模块，操作系统和应用程序都可以固化在 ROM 中。

2. 中间层

硬件层与软件层之间为中间层，它把系统软件与底层硬件部分隔离，使得系统的底层设备驱动程序与硬件无关。中间层一般包括硬件抽象层(Hardware Abstract Layer, HAL)和板级支持包(Board Support Package, BSP)。

硬件抽象层是位于操作系统内核与硬件电路之间的接口层，其目的是将硬件抽象化，即可以通过程序来控制所有硬件电路(如 CPU、I/O、存储器等)的操作，这就使得系统的驱动程序与硬件设备无关，从而大大提高了系统的可移植性。从软/硬件测试的角度看，软/硬件的测试工作可分别基于硬件抽象层完成，使得软/硬件的测试工作能够并行进行。在定义 HAL 时，需要规定统一的软/硬件接口标准，其设计工作需要基于系统需求来做，代码编写工作可由对硬件层比较熟悉的人员来完成。HAL 一般应包含相关硬件的初始化、数据的输入/输出操作、硬件设备的配置操作等功能。

板级支持包介于主板硬件和操作系统中的驱动层程序之间，一般认为它属于操作系统的一部分，主要实现对操作系统的支持，为上层的驱动程序提供访问硬件设备寄存器的函数包，使之能够在硬件主板上更好地运行。BSP 是相对操作系统而言的，不同的操作系统对应于不同定义形式的 BSP。因此，BSP 一定要按照该系统 BSP 的定义形式来写(BSP 的编程过程大多数是在某一个成型的 BSP 模板上进行修改)，这样才能与上层的操作系统保持正确的接口，良好地支持上层操作系统。BSP 一般实现以下两个方面的功能：系统启动时完成对硬件的初始化；为驱动程序提供访问硬件的手段，即为上层的驱动程序提供访问硬件设备寄存器的函数包。

3. 软件层

软件层由实时操作系统(Real Time Operating System, RTOS)、文件系统、图形用户接口(Graphical User Interfaces, GUI)、网络组件组成。RTOS 是嵌入式应用软件的基础和开发平台。大多数 RTOS 都是针对不同微处理器优化设计的高效实时多任务内核，可以在不同微处理器上运行而为用户提供相同的 API 接口。因此，基于 RTOS 开发的应用程序具有非常好的可移植性。

4. 功能层

功能层由基于 RTOS 开发的应用程序组成，用来完成实际所需的应用功能。功能层是面向被控对象和用户的，当用户操作时往往需要提供一个友好的人机界面。

1.2.4 嵌入式系统的分类

嵌入式系统可按照嵌入式微处理器的位数、实时性、软件结构以及应用领域等进行分类。

1. 按照嵌入式微处理器的位数分类

按照嵌入式微处理器字长的位数，嵌入式系统可分为 4 位、8 位、16 位、32 位和 64 位。其中，4 位、8 位、16 位嵌入式系统已经获得了大量应用，32 位嵌入式系统正成为主流，而一些高度复杂和要求高速处理的嵌入式系统已经开始使用 64 位嵌入式微处理器。

2. 按照实时性分类

实时系统是指系统执行的正确性不仅取决于计算的逻辑结果，还取决于结果产生的时间。根据嵌入式系统是否具有实时性，可将其分为嵌入式实时系统和嵌入式非实时系统。

大多数嵌入式系统都属于嵌入式实时系统。根据实时性的强弱，实时系统又可进一步分为硬实时系统和软实时系统。

硬实时系统是指系统对响应时间有严格要求，如果响应时间不能满足，就会引起系统崩溃或出现致命错误，如飞机的飞控系统。软实时系统是指系统对响应时间有一定要求，如果响应时间不能满足，不会导致系统崩溃或出现致命错误，如打印机、自动门。可以认为两者的区别在本质上属于客观要求和主观感受的区别。

3. 按照嵌入式软件结构分类

按照嵌入式软件的结构分类，嵌入式系统可分为循环轮询系统、前后台系统和多任务系统。

1) 循环轮询系统

循环轮询(polling loop)是最简单的软件结构，程序依次检查系统的每个输入条件，如果条件成立就执行相应处理。其流程图如图 1-3 所示。

示意代码如下：

```
initialize()  
while(true) {  
    if (condition_1) action_1()  
    if (condition_2) action_2()  
    :  
    if (condition_n) action_n()  
}
```

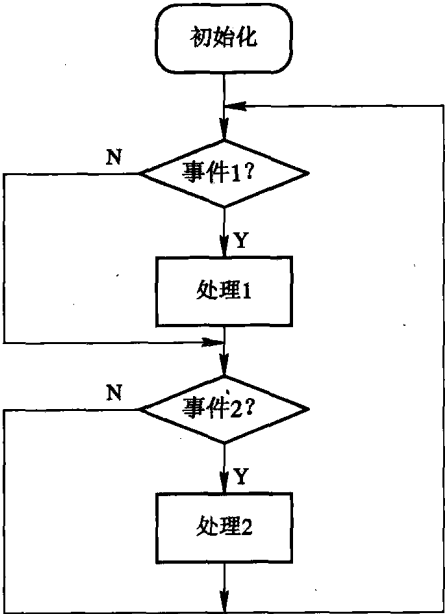


图 1-3 循环轮询流程图

2) 前后台系统

前后台(foreground/background)系统属于中断驱动机制。后台程序是一个无限循环，通过调用函数实现相应操作，又称任务级。前台程序是中断处理程序，用来处理异步事件，又称中断级。设计前后台的目的主要是通过中断服务来保证时间性很强的关键操作(critical operation)。通常情况下，中断只处理需要快速响应的事件，将输入/输出数据存放在内存的缓冲区里，再向后台发信号，由后台来处理这些数据，如运算、存储、显示、打印等。其流程图如图 1-4 所示。

在前后台系统中，主要考虑的问题包括中断的现场保护和恢复、中断的嵌套、中断与主程序共享资源等问题。系统性能由中断延迟时间、响应时间和恢复时间来描述。

一些不复杂的小系统比较适合采用前后台系统的结构来设计程序。甚至在某些系统中，为了省电，平时让处理器处于停机状态(halt)，所有工作都依靠中断服务来完成。