

高等学校电子信息类专业
“十三五”规划教材

ELECTRONIC
INFORMATION SPECIALTY

嵌入式系统原理 与开发 (第三版)

主 编 夏靖波 陈雅蓉
参 编 胡健生 王 航

 西安电子科技大学出版社
<http://www.xduph.com>



高等学校电子信息类专业“十三五”规划教材

嵌入式系统原理与开发

(第三版)

主编 夏靖波 陈雅蓉

参编 胡健生 王 航

西安电子科技大学出版社

内 容 简 介

本书全面介绍了嵌入式系统基本技术和开发方法,根据嵌入式系统的结构体系,从基本概念出发,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可划分为嵌入式处理器、嵌入式外围硬件、设备驱动、实时操作系统和嵌入式应用程序五个层次。本书在阐述每个层次原理的基础上,基于 ARM 处理器和 $\mu\text{C}/\text{OS-II}$ 操作系统平台,完整地分析了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,并给出了相应的应用实例。

本书结构合理,内容系统、全面,可作为高等院校计算机网络工程、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,陈雅蓉主编. —3版. —西安:西安电子科技大学出版社,2017.8
ISBN 978-7-5606-4517-9

I. ①嵌… II. ①夏… ②陈… III. ①微型计算机—系统设计 IV. ①TP360.21

中国版本图书馆 CIP 数据核字(2017)第 150645 号

策 划 云立实

责任编辑 买永莲

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 陕西利达印务有限责任公司

版 次 2017年8月第1版 2017年8月第1次印刷

开 本 787毫米×1092毫米 1/16 印 张 20.75

字 数 493千字

印 数 17 001~20 000册

定 价 42.00元

ISBN 978 - 7 - 5606 - 4517 - 9/TP

XDUP 4809003-7

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

《嵌入式系统原理与设计(第一版)》自 2006 年出版以来,已被国内 30 余所高等院校相关专业选为授课教材。与该书配套开发的网络课程在第十三届全国多媒体教育软件大奖赛中,获得高等教育组网络课程一等奖。2010 年,编者对第一版进行了修订,更新了部分内容,推出了该书的第二版,受到读者的广泛欢迎。

近年来随着微电子、通信和网络等技术的迅猛发展,嵌入式技术不断更新,各类产品功能日趋强大,应用范围更加广泛。因此,我们在第二版的基础上,结合长期工程实践和教学活动经验,增加了嵌入式技术的最新发展动态,更新了部分数据,扩展了部分技术原理及应用实例,同时删减了一些陈旧内容。具体如下:更新了 ARM 处理相关的统计数据,增加了对 ARM 处理器总线架构和 Cortex 系列产品的介绍;对硬件接口电路及应用实例进行了完善,增加了 SD 卡电路设计与实例、LCD 控制原理、IIS 控制原理与应用实例以及 Wi-Fi 技术等相关内容;添加了基于 ARM11 和 FPGA 的图像采集处理系统应用实例;对嵌入式系统设计流程和方法等内容进行了完善,删除了第一、二版中关于系统设计形式和方法等理论知识的介绍,增加了电子器件封装、硬件开发与软件开发常用工具的介绍。

通过上述几个方面的修改和完善,新版本紧跟当前嵌入式技术的最新发展趋势,内容上覆盖了嵌入式系统中常用接口技术和典型应用实例,更加贴近工程实践。

编 者

2017 年 4 月

目 录

第1章 嵌入式系统概述.....1	第3章 嵌入式硬件平台.....88
1.1 嵌入式系统的产生和发展.....1	3.1 引言.....88
1.2 嵌入式系统的概念.....3	3.2 嵌入式硬件平台概述.....88
1.2.1 嵌入式系统的定义.....3	3.3 总线.....89
1.2.2 嵌入式系统的特点.....3	3.3.1 总线协议.....90
1.2.3 嵌入式系统的组成结构.....6	3.3.2 DMA.....92
1.2.4 嵌入式系统的分类.....7	3.3.3 总线配置.....93
1.3 嵌入式系统的应用领域.....11	3.3.4 总线实例.....94
1.4 嵌入式系统的发展趋势.....12	3.4 存储设备.....94
思考与练习题.....15	3.4.1 嵌入式系统存储器子系统的结构.....94
第2章 嵌入式处理器.....16	3.4.2 RAM.....96
2.1 引言.....16	3.4.3 ROM.....98
2.2 嵌入式处理器概述.....16	3.4.4 Flash Memory.....99
2.2.1 嵌入式处理器的分类.....16	3.4.5 SD/TF 存储卡.....100
2.2.2 典型的嵌入式处理器.....18	3.5 I/O 设备.....102
2.3 ARM 处理器基础.....22	3.5.1 定时器/计数器.....102
2.3.1 ARM 简介.....22	3.5.2 模/数转换器和数/模转换器.....103
2.3.2 ARM 处理器系列.....23	3.5.3 人机接口设备.....103
2.3.3 ARM 处理器体系结构.....30	3.6 通信设备.....108
2.3.4 ARM 处理器应用选型.....36	3.6.1 通用异步收发器.....108
2.4 ARM 指令系统.....37	3.6.2 USB 设备.....111
2.4.1 ARM 编程模型.....37	3.6.3 Ethernet 设备.....116
2.4.2 ARM 寻址方式.....46	3.6.4 Wi-Fi.....117
2.4.3 ARM 指令集.....50	3.7 其他.....118
2.4.4 Thumb 指令集.....64	3.7.1 电源.....118
2.5 ARM 程序设计基础.....74	3.7.2 时钟.....118
2.5.1 ARM 汇编语句格式.....75	3.7.3 复位.....119
2.5.2 ARM 汇编程序格式.....75	3.7.4 中断.....121
2.5.3 汇编语言编程实例.....76	思考与练习题.....122
2.5.4 汇编语言与 C 语言的混合编程.....78	第4章 BootLoader 与设备驱动.....123
思考与练习题.....85	4.1 引言.....123
	4.2 BootLoader.....124

4.2.1 BootLoader 概述.....	124	5.6 $\mu\text{C}/\text{OS-II}$ 在 ARM 上的移植.....	185
4.2.2 BootLoader 的典型结构.....	125	5.6.1 $\mu\text{C}/\text{OS-II}$ 的移植条件.....	185
4.2.3 实例分析.....	128	5.6.2 $\mu\text{C}/\text{OS-II}$ 的移植步骤.....	186
4.3 设备驱动.....	139	5.6.3 测试移植代码.....	191
4.3.1 设备驱动概述.....	139	5.7 基于 $\mu\text{C}/\text{OS-II}$ 构建的 TCP/IP/PPP	
4.3.2 LCD 驱动控制实例.....	143	协议栈.....	191
4.3.3 A/D 转换功能驱动实例.....	148	5.7.1 嵌入式协议栈概述.....	191
4.3.4 IIS 音频接口驱动实例.....	151	5.7.2 选择协议栈.....	192
思考与练习题.....	157	5.7.3 嵌入式 TCP/IP 协议栈.....	192
第 5 章 嵌入式操作系统.....	158	5.7.4 基于 $\mu\text{C}/\text{OS-II}$ 的 TCP/IP/PPP	
5.1 引言.....	158	协议栈.....	194
5.2 嵌入式操作系统概述.....	159	思考与练习题.....	196
5.2.1 嵌入式实时操作系统.....	159	第 6 章 嵌入式应用程序设计.....	197
5.2.2 典型的嵌入式操作系统.....	160	6.1 引言.....	197
5.3 操作系统的基本概念.....	164	6.2 程序设计方法与技术.....	197
5.3.1 多进程和多线程.....	164	6.2.1 程序设计方法.....	197
5.3.2 任务.....	164	6.2.2 程序设计技术.....	201
5.3.3 任务切换.....	166	6.2.3 嵌入式程序优化.....	205
5.3.4 内核.....	166	6.3 基于网络的嵌入式系统设计.....	215
5.3.5 任务调度.....	166	6.3.1 分布式嵌入式系统.....	215
5.3.6 任务间的通信与同步.....	168	6.3.2 嵌入式系统网络.....	221
5.3.7 操作系统的结构和功能.....	171	6.3.3 基于网络的设计示例.....	235
5.4 $\mu\text{C}/\text{OS-II}$ 简介.....	171	6.4 基于 ARM11+FPGA 的图像采集	
5.4.1 $\mu\text{C}/\text{OS-II}$ 概述.....	171	处理系统设计.....	243
5.4.2 $\mu\text{C}/\text{OS-II}$ 的特点.....	172	6.4.1 系统总体结构.....	244
5.4.3 $\mu\text{C}/\text{OS-II}$ 的软件体系结构.....	173	6.4.2 多路视频信号的采集、缓存和	
5.5 $\mu\text{C}/\text{OS-II}$ 内核结构.....	174	时序同步.....	244
5.5.1 临界段.....	174	6.4.3 视频信号的压缩编码.....	246
5.5.2 任务.....	175	6.4.4 结论.....	246
5.5.3 任务控制块 OS_TCB.....	176	思考与练习题.....	247
5.5.4 任务调度.....	177	第 7 章 系统设计技术.....	248
5.5.5 任务管理.....	177	7.1 引言.....	248
5.5.6 中断服务.....	178	7.2 设计流程.....	248
5.5.7 时钟节拍与时间管理.....	179	7.2.1 开发过程.....	248
5.5.8 任务间同步与通信的管理.....	180	7.2.2 设计流程.....	252
5.5.9 内存管理.....	183	7.3 需求分析与规格说明.....	255
5.5.10 $\mu\text{C}/\text{OS-II}$ 的初始化.....	184	7.3.1 需求分析.....	255
5.5.11 $\mu\text{C}/\text{OS-II}$ 的启动.....	185		

7.3.2 规格说明.....	255	8.3.2 交互式开发调试方法.....	282
7.4 系统分析与体系结构设计.....	258	8.4 其他实用工具.....	290
思考与练习题.....	259	8.4.1 Source Insight.....	290
		8.4.2 SkyEye.....	291
第8章 嵌入式系统开发调试方法.....	260	思考与练习题.....	294
8.1 引言.....	260	附录A 嵌入式专业词汇与缩略语.....	295
8.2 嵌入式系统硬件开发基础及流程.....	260	附录B ARM 指令集列表.....	304
8.2.1 电子元器件的封装.....	260	附录C Thumb 指令集列表.....	306
8.2.2 PCB 基础知识.....	262	附录D 汇编预定义变量及伪指令.....	309
8.2.3 硬件设计调试中常用工具.....	266	附录E μ C/OS-II 内核函数.....	314
8.3 嵌入式系统交互式开发调试工具.....	267	参考文献.....	324
8.3.1 嵌入式系统集成开发环境.....	267		



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。

由微型机不断增强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,例如给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。在更多的场合下,人们还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则得益于软件技术的进步，这些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，为了保障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，相关的操作系统有 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，并且支持主流的嵌入式微处理器。开发工具则有各种面向软、硬件开发的工具，如硬件仿真器、源码级的交叉调试器等。在这些软件的支持下，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，如智能手机、MP4 播放器、数码相机、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，嵌入式软件的规模呈指数型增长，嵌入式应用具备了更加复杂和高度智能化的功能，软件在系统中的重要程度越来越高，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。

如今，嵌入式系统已经无处不在，在应用数量上已远超通用计算机，据相关机构统计，2012 年全球嵌入式软件的销售规模已经达到了 500 亿美元，而嵌入式体系产品的产值达到了 6000 亿美元。2004 年~2011 年，全球嵌入式系统市场的收入规模比 2011 年增长了 30.1%；2015 年，全球智能系统的设备量则达到了 150 亿之巨。



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》中这样定义：“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被称为嵌入式计算系统，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，便可构成特定系统的一体化软件。这种专用计算机系统必然在可靠性、实时性、功耗、可裁剪性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) 对运行环境依赖。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、体积等因素。这些等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、软件与硬件集成、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,这方面的方法的研究已经发展成一门学科。

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、Cyrix 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 于 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个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。

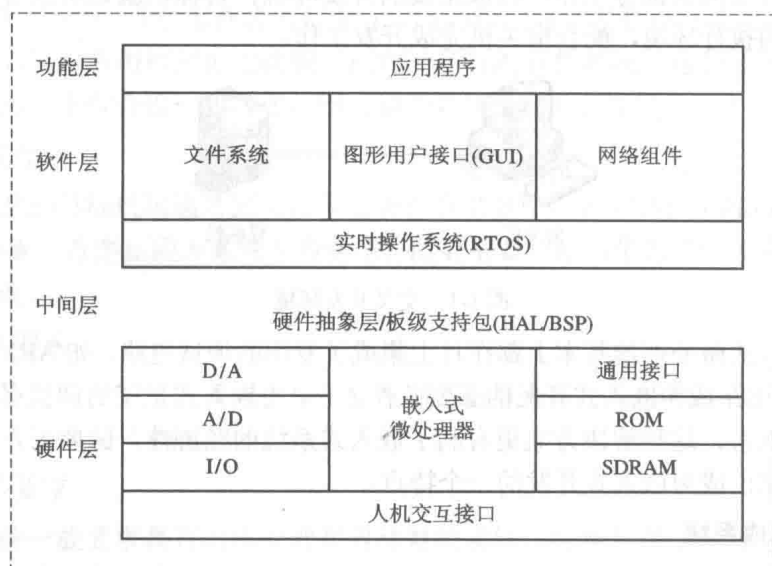


图 1-2 嵌入式系统的组成结构

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()  
}
```

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。

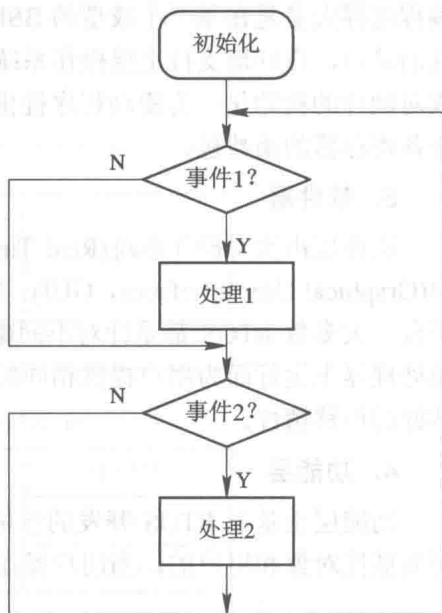


图 1-3 循环轮询系统流程图

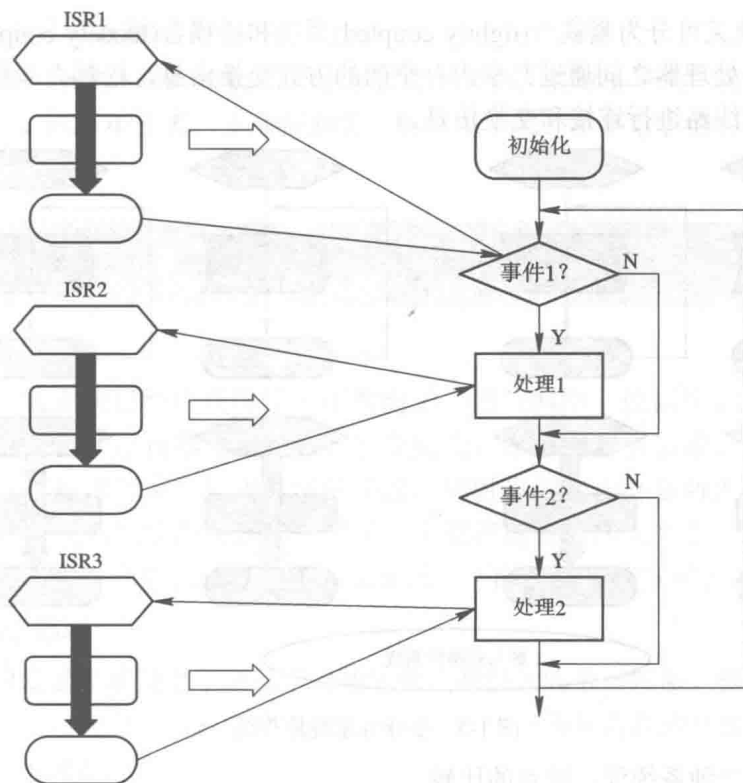


图 1-4 前后台系统流程图

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。

3) 多任务系统

对于较复杂的嵌入式系统而言，许多互不相关的过程有时需要计算机同时处理，在这种情况下就需要采用多任务(multitasking)系统。采用多任务结构设计软件有利于降低系统的复杂度，保证系统的实时性和可维护性。

多任务系统的软件由多个任务、多个中断服务程序以及嵌入式操作系统组成。任务是顺序执行的，其并行性通过操作系统完成。操作系统主要负责任务切换、任务调度、任务间以及任务与中断服务程序之间的通信、同步、互斥、实时时钟管理、中断管理等。其流程图如图 1-5 所示。

多任务系统的特点包括如下内容：

- (1) 每个任务都是一个无限循环的程序，等待特定的输入，从而执行相应的处理。
- (2) 这种程序模型将系统分成相对简单、相互合作的模块。
- (3) 不同的任务共享同一个 CPU 和其他硬件，嵌入式操作系统对这些共享资源进行管理。
- (4) 多个顺序执行的任务在宏观上看是并行执行的，每个任务都运行在自己独立的 CPU 上。

在单处理器系统中，任务在宏观上看是并发执行的，但在微观上看实际是顺序执行的。在多处理器系统中，可以让任务同时在不同的处理器上执行，因此在微观上看任务也是并发执行的。多处理器系统可分为单指令多数据流(SIMD)系统和多指令多数据流(MIMD)系



统。MIMD 系统又可分为紧耦合(tightly-coupled)系统和松耦合(loosely-coupled)系统。紧耦合系统是指多个处理器之间通过共享内存空间的方式交换信息，松耦合系统是指多个处理器之间通过通信线路进行连接和交换信息。

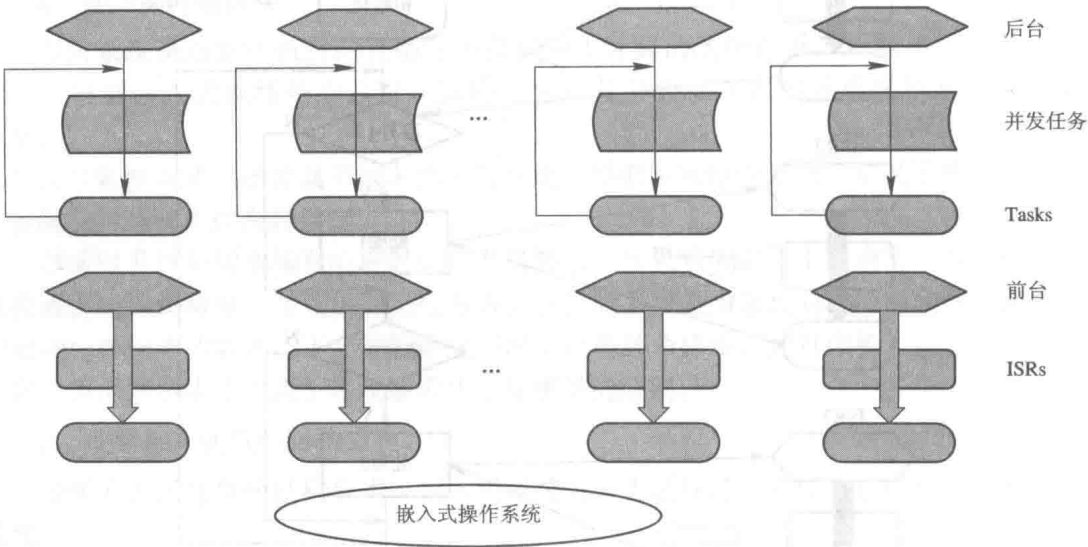


图 1-5 多任务系统流程图

表 1-1 是这三种系统优、缺点的比较。

表 1-1 三种系统的比较

系统分类	优 点	缺 点
循环轮询系统	编程简单； 没有中断，不会出现随机问题	应用领域有限； 不适合有大量输入/输出的服务； 程序规模增大后不便于调试
前后台系统	可并行处理不同的异步事件； 设计简单，无需学习操作系统的相关知识	对于复杂系统，其主程序设计复杂， 可靠性降低； 实时性只能通过中断来保证，一旦主 程序介入处理事件，其实时性难以保证； 中断服务程序与主程序之间共享、互 斥的问题需要解决
多任务系统	复杂的系统被分解成相对独立的多个任 务，降低了系统的复杂度； 可以保证系统的实时性； 系统模块化，可维护性强	需要引入新的软件设计方法； 需要对每个共享资源进行互斥； 任务间存在竞争； 嵌入式操作系统的使用将会增加系统 开销

从表 1-1 可以看出，循环轮询系统适合于实时性要求不高、非常简单的系统；前后台系统适合于小型、较简单的系统；多任务系统适合于大型、复杂、实时性要求较高的嵌入式系统。目前，多任务系统已经广泛应用于 32 位嵌入式系统。