



21st CENTURY
规划教材

高等院校信息与电子技术类规划教材

University Textbooks of Information Technology

嵌入式系统设计与开发

The Embedded System Designing and Developing

马文华 主编



21st CENTURY



科学出版社

www.sciencep.com

高等院校信息与电子技术类规划教材

嵌入式系统设计与开发

马文华 主编

邓耀华 汤秀春 胡 伟 副主编

科学出版社

北 京

内 容 简 介

本书以目前流行的 ARM 处理器和易于学习的 $\mu\text{C}/\text{OS-II}$ 操作系统为核心, 系统介绍了嵌入式系统的原理、设计方法和编程开发。全书共分 10 章, 分别为嵌入式系统概述、ARM 及其编程模型、ARM 指令集、嵌入式程序设计、 $\mu\text{C}/\text{OS-II}$ 实时操作系统、嵌入式最小系统设计、部件工作原理与开发实例、基于 $\mu\text{C}/\text{OS-II}$ 的程序设计、ARM 嵌入式开发工具以及 ARM9 处理器。读者可从中系统地学习嵌入式系统的有关知识, 并通过实例完成嵌入式系统设计的基础训练。

本书可以作为高等学校嵌入式系统教学的教材, 也可作为有关工程技术人员的参考书。

图书在版编目 (CIP) 数据

嵌入式系统设计与开发/马文华主编. —北京: 科学出版社, 2006

(高等院校信息与电子技术类规划教材)

ISBN 7-03-016965-4

I. 嵌… II. 马… III. 微型计算机—系充开发—高等学校—教材
IV. TP360.21

中国版本图书馆 CIP 数据核字 (2006) 第 014943 号

责任编辑: 李 伟/责任校对: 刘彦妮

责任印制: 吕春珉/封面设计: 飞天创意

科学出版社 出版

北京东黄城根北街 16 号

邮政编码: 100717

<http://www.sciencep.com>

德清彩色印装有限公司 印刷

科学出版社发行 各地新华书店经销

*

2006 年 3 月第 一 版 开本: 787×1092 1/16

2006 年 3 月第一次印刷 印张: 21

印数: 1—3 000 字数: 478 000

定价: 30.00 元

(如有印装质量问题, 我社负责调换〈环伟〉)

高等院校信息与电子技术类规划教材

编委会

主任 吴黎明（广东工业大学信息工程学院副院长、教授）

副主任 贺前华（华南理工大学电子与信息学院副院长、教授）

委员（按姓氏笔画排序）

马文华（广东外语外贸大学信息科学与技术学院副教授）

汤庸（中山大学信息科学与技术学院副院长、教授）

杨振野（广东大学师范学院电子系教授）

吴正光（广州大学实验中心副主任、高级工程师）

周美娟（广东海洋大学信息学院院长、教授）

洪添胜（华南农业大学信息学院院长、教授）

徐杜（广东工业大学信息工程学院陆军长、教授）

颜国正（上海交通大学电子信息与电气工程学院测控系主任、教授）

前 言

嵌入式计算机系统起源于微型机时代，其低端应用以 8 位 MCS-51 为代表，至今仍在大量使用。近几年，网络、通信、多媒体和计算控制技术的发展为嵌入式系统应用开辟了广阔的天地，使嵌入式系统成为继个人计算机、因特网后 IT 界新的技术热点，但在这些高端应用中，8 位处理器已不能满足应用的需求，取而代之的是 32 位微处理器。目前，32 位的 ARM 处理器以其优良的性能，占据了低功耗、低成本和高性能的嵌入式系统应用领域的领先地位。ARM7TDMI 是 ARM 公司的经典内核，三星公司以此内核为基础生产的 SC344B0X 芯片得到了广泛应用，拥有许多的用户，对于用户来说，将该芯片作为嵌入式系统的入门是个很好的选择。32 位微处理器面向嵌入式系统的高端应用，由于其速度快、资源丰富等特点，加上应用本身的复杂性、可靠性等要求，软件的开发一般选择操作系统平台支持。常见的嵌入式操作系统有很多，例如，VxWorks、Windows CE、嵌入式 Linux、QNX、Nucleus 和 μ C/OS-II 等，从中选择一个能体现操作系统机理、有实用价值、而门槛又较低的操作系统是一项很重要的工作。 μ C/OS-II 是一个源代码公开的实时多任务操作系统，拥有很多产品的成功应用案例并且得到美国有关方面的认证，用户可以很容易地通过阅读源代码领会操作系统的精髓，使操作系统的学习不再如“空中楼阁”。

本书以 SC344B0X 和 μ C/OS-II 为平台来介绍嵌入式系统的设计和开发技术。由于 ARM 体系结构的一致性、外围电路的通用性及操作系统机理的一致性，本书对设计其他 ARM 内核芯片的应用系统也具有很好的参考价值。其编写思路符合嵌入式系统注重实践的学习规律，力求通俗、易懂，适合作为电信类和计算机学科本科生及研究生教材，也可作为课程设计、毕业设计和从事嵌入式系统开发工程技术人员的参考资料。

嵌入式系统的应用几乎涉及现代生活的各个方面，如手机、电视机、DVD、照相机、游戏机、微波炉和汽车等，即使我们通用的个人计算机，其外围电路中也包含众多的嵌入式处理器。社会对嵌入式系统设计人才需求旺盛，目前，嵌入式系统方面高端人才严重缺乏。很多高校也在进行课程改革，并且增加嵌入式系统方面的课程，这将为社会培养专业人才、为我国在这一领域赶超世界先进水平打下坚实的人才基础。

本书各章节安排如下：

第 1 章介绍嵌入式系统的概念、发展历史、应用范围，嵌入式系统的硬件和软件组成及嵌入式系统开发的相关技术。

第 2 章介绍精简指令集计算机 RISC 和复杂指令集计算机 CISC，在此基础上，介绍了 ARM 体系结构版本及 ARM 的编程模型，为 ARM 体系结构的学习打下基础，对 ARM7TDMI 做了较详细的说明。

第 3 章详细介绍 ARM 体系的指令系统和寻址方式。Thumb 指令集是 ARM 指令集的子集，本章对 Thumb 指令集及其与 ARM 指令集的区别也做了简要介绍。

第 4 章介绍 ARM 的嵌入式程序设计，包括 ARM 汇编语言程序设计、嵌入式 C 语

言程序设计以及 C 语言和汇编混合编程。

第 5 章介绍嵌入式实时操作系统 $\mu\text{C}/\text{OS-II}$ 的结构、原理和特点。在此基础上,介绍 $\mu\text{C}/\text{OS-II}$ 在 ARM 上的移植。

第 6 章介绍基于 S3C44B0X 最小系统的设计步骤和方法以及 BootLoader 的设计。在嵌入式系统中, BootLoader 是操作系统内核运行之前运行的一小段程序,通过 BootLoader 可以完成对系统板上的主要部件进行初始化,从而将系统的软硬件环境带到一个合适的状态。

第 7 章详细介绍 S3C44B0X 主要部件的工作原理与开发实例,内容包括存储器、通用 I/O 接口、串行通信接口、中断控制器与键盘、定时器、LCD 控制器和 I²C 总线接口。通过本章的学习,可以建立起基于 S3C44B0X 开发嵌入式系统的基础。

第 8 章介绍基于 $\mu\text{C}/\text{OS-II}$ 的应用程序设计和实例以及 $\mu\text{C}/\text{OS-II}$ 的扩展。由于 $\mu\text{C}/\text{OS-II}$ 仅是一个实时多任务内核,移植 $\mu\text{C}/\text{OS-II}$ 到处理器平台后,和实际应用还有一段距离,因此要建立一个实用的实时操作系统必须对 $\mu\text{C}/\text{OS-II}$ 进行必要的扩展。

第 9 章介绍 ARM 开发工具的组成。其中详细地介绍了 ARM 芯片上调试系统、仿真器和上层 IDE 集成开发环境。对于集成开发环境 IDE 主要介绍目前国内常用的两种开发工具: ARM 公司的 ADS 和英蓓特公司的 Embest IDE。

第 10 章对 ARM7 和 ARM9 的处理器核 ARM7TDMI 和 ARM9TDMI 进行了对比性介绍。

全书由马文华任主编,邓耀华、汤秀春和胡伟任副主编,薛向东、伍冯洁、刘润予和甘达也参与了本书的编写。本书的部分文字输入工作和电路图绘制由朱高峰、陈丹青、李政广、林金萱和赖南辉完成。卢绮雯、张巨荣、陈景郁、洪卓辉和梁顺发对书中程序进行上机实验验证。

吴黎明教授对本书的编写提出许多建设性意见,并审阅了全书,在此表示衷心感谢。

在本书的编写过程中,得到深圳英蓓特信息技术公司、深圳旋极科技公司、ARM 公司上海办事处、上海祥佑数码科技公司的大力支持,在此谨向他们深表感谢。

由于作者水平有限及嵌入式系统在国内开展教学时间不长,书中难免存在不当之处,恳请读者批评指正。

目 录

第 1 章 概述	1
1.1 嵌入式系统.....	1
1.1.1 嵌入式系统的定义.....	1
1.1.2 嵌入式系统的发展历史.....	1
1.1.3 嵌入式计算机系统与通用计算机系统.....	2
1.2 嵌入式系统的组成.....	3
1.2.1 嵌入式系统的硬件组成.....	3
1.2.2 嵌入式系统的软件组成.....	4
1.3 嵌入式系统的主要应用领域.....	6
1.4 嵌入式处理器.....	7
1.4.1 嵌入式处理器分类.....	7
1.4.2 嵌入式处理器现状及发展趋势.....	9
1.5 嵌入式操作系统.....	10
1.5.1 嵌入式操作系统分类.....	11
1.5.2 实时操作系统.....	11
1.5.3 嵌入式操作系统的特点.....	13
1.5.4 目前市场上流行的嵌入式操作系统.....	14
1.5.5 嵌入式操作系统在我国的应用现状.....	17
1.6 嵌入式系统的开发.....	18
1.6.1 嵌入式系统设计流程.....	18
1.6.2 嵌入式系统开发考虑的一些问题.....	19
本章要点.....	22
习题.....	23
第 2 章 ARM 及其编程模型	24
2.1 ARM 简介.....	24
2.2 RISC 体系结构.....	24
2.3 ARM 体系结构.....	25
2.3.1 ARM 体系结构的版本.....	25
2.3.2 ARM 体系结构的变种及版本命名格式.....	27
2.3.3 ARM 处理器系列.....	29
2.4 ARM 微处理器的编程模型.....	32
2.4.1 ARM 处理器支持的数据类型.....	32
2.4.2 ARM 处理器的工作状态.....	33
2.4.3 ARM 处理器的工作模式.....	34

2.4.4	ARM 状态下的寄存器组织.....	34
2.5	ARM 的异常中断	38
2.5.1	ARM 体系结构所支持的异常类型.....	38
2.5.2	异常的响应及返回.....	39
2.5.3	各类异常及返回.....	40
2.6	ARM7TDMI 处理器内核简介	43
2.6.1	ARM7TDMI 介绍	44
2.6.2	ARM7TDMI 处理器核的硬件接口	46
	本章要点	50
	习题.....	50
第 3 章	ARM 指令集	52
3.1	ARM 指令集概述	52
3.1.1	ARM 指令的编码格式.....	52
3.1.2	ARM 指令的条件码域.....	53
3.1.3	指令流水线.....	53
3.2	ARM 寻址方式	54
3.3	ARM 指令分类介绍	57
3.3.1	数据处理指令.....	57
3.3.2	分支指令	62
3.3.3	存储器访问指令.....	64
3.3.4	程序状态寄存器与通用寄存器之间的传送指令	68
3.3.5	乘法指令	69
3.3.6	协处理器指令.....	70
3.3.7	异常中断产生指令.....	72
3.4	Thumb 指令集.....	73
3.4.1	Thumb 指令集概述	73
3.4.2	Thumb 寄存器和 ARM 寄存器之间的关系	74
3.4.3	Thumb 指令分类介绍	75
	本章要点	84
	习题.....	84
第 4 章	嵌入式程序设计	86
4.1	汇编语言程序结构	86
4.1.1	ARM 编译模式与开发环境.....	86
4.1.2	ARM 汇编语言的伪操作、伪指令与宏指令	89
4.1.3	CodeWarrior 编译器下的伪操作与伪指令	90
4.2	ARM 汇编语言程序设计	96
4.2.1	ARM 汇编中的文件格式.....	96
4.2.2	ARM 汇编语句格式.....	96
4.2.3	ARM 汇编程序设计.....	100

4.2.4	ARM 与 Thumb 之间的状态转换及函数的互相调用	106
4.3	C 语言与汇编语言混合编程	108
4.3.1	C 程序与汇编程序互相调用规则	109
4.3.2	内嵌汇编程序设计	111
4.3.3	C 语言函数和 ARM 汇编语言函数间互相调用	114
	本章要点	116
	习题	116
第 5 章	μC/OS-II 实时操作系统	118
5.1	μC/OS-II 实时操作系统概述	118
5.1.1	实时系统概念	118
5.1.2	μC/OS-II 的主要特点	119
5.2	μC/OS-II 的内核结构	121
5.2.1	μC/OS-II 内核保护机制	121
5.2.2	任务	123
5.2.3	任务状态	125
5.2.4	任务控制块	126
5.2.5	任务调度	129
5.3	同步	131
5.4	μC/OS-II 中的基本数据类型	133
5.5	μC/OS-II 在 ARM 上的移植	133
5.5.1	移植步骤	134
5.5.2	μC/OS-II 移植后的测试	139
	本章要点	141
	习题	141
第 6 章	嵌入式最小系统设计	142
6.1	系统设计概述	142
6.2	S3C44B0X 概述	143
6.2.1	S3C44B0X 引脚及信号描述	144
6.2.2	S3C44B0X 特性	147
6.2.3	ARM 存储器	150
6.2.4	S3C44B0X 存储控制器	152
6.3	系统硬件基本单元电路设计	159
6.3.1	电源、复位、时钟电路和 JTAG 接口	159
6.3.2	Flash 存储器接口电路	160
6.3.3	SDRAM 存储器接口电路	161
6.4	S3C44B0X 启动程序设计	161
6.4.1	BootLoader 介绍	161
6.4.2	启动程序设计实例	162

本章要点	168
习题	168
第7章 部件工作原理与开发实例	169
7.1 Flash 存储器	169
7.1.1 概述	169
7.1.2 存储器初始化	169
7.1.3 Flash 的调试和程序固化	170
7.1.4 Flash 器件 SST39VF160 的应用	171
7.2 SDRAM 存储器	175
7.2.1 概述	175
7.2.2 SRAM 器件的结构特点	175
7.2.3 DRAM 器件的结构特点	175
7.2.4 SDRAM 器件的构成原理和应用特点	177
7.2.5 SDRAM 器件 HY57V641620HG 简介	177
7.2.6 SDRAM 应用编程	178
7.3 通用 I/O 接口	179
7.3.1 S3C44B0X 的 I/O 端口	179
7.3.2 S3C44B0X 芯片与端口相关的寄存器	181
7.3.3 应用实例	183
7.4 串行通信接口	185
7.4.1 概述	185
7.4.2 S3C44B0X 的串行通信单元	185
7.4.3 UART 操作	186
7.4.4 UART 寄存器	189
7.4.5 串行通信的应用与编程	193
7.5 中断控制器与键盘	195
7.5.1 传统键盘介绍	195
7.5.2 中断控制器概述	196
7.5.3 键盘的软硬件实现原理	202
7.5.4 中断式键盘的应用编程	204
7.6 定时器	207
7.6.1 概述	207
7.6.2 S3C44B0X PWM 定时器	208
7.6.3 PWM 定时器工作原理	208
7.6.4 PWM 定时器操作	209
7.6.5 PWM 定时器的特殊功能寄存器	212
7.6.6 PWM 定时器的应用编程	215
7.7 LCD 控制器	216
7.7.1 概述	216

7.7.2	液晶显示屏原理.....	217
7.7.3	S3C44B0X LCD 控制器.....	218
7.7.4	开发实例设计.....	223
7.7.5	参考程序.....	226
7.8	I ² C 总线接口.....	227
7.8.1	S3C44B0X I ² C 总线概述.....	227
7.8.2	S3C44B0X I ² C 总线接口操作.....	228
7.8.3	S3C44B0X I ² C 接口特殊功能寄存器.....	231
7.8.4	S3C44B0X I ² C 应用编程.....	233
	本章要点.....	235
	习题.....	236
第 8 章	基于μC/OS-II 的程序设计	237
8.1	基于 μ C/OS-II 的应用程序设计方法.....	237
8.2	μ C/OS-II 操作系统扩展.....	241
8.3	μ C/OS-II 的文件系统.....	258
8.3.1	Flash 盘文件系统实现原理.....	258
8.3.2	Flash 盘文件系统的实现过程.....	259
8.4	基于 μ C/OS-II 的图形用户界面基础.....	260
	本章要点.....	263
	习题.....	264
第 9 章	ARM 嵌入式开发工具	265
9.1	ARM 下层开发工具.....	265
9.1.1	JTAG 接口.....	266
9.1.2	JTAG TAP.....	267
9.1.3	Embedded ICE 嵌入式在线仿真器.....	270
9.1.4	ETM 嵌入式跟踪宏.....	272
9.2	ARM 开发工具概述.....	273
9.2.1	JTAG 仿真器.....	273
9.2.2	在线仿真器 ICE.....	276
9.2.3	集成开发环境本身带的仿真软件.....	276
9.2.4	Multi 2000 集成开发环境.....	276
9.2.5	RVDS 集成开发环境.....	278
9.3	ARM 的 ADS.....	278
9.3.1	ADS 介绍.....	278
9.3.2	ADS 应用.....	279
9.4	Embest IDE 介绍.....	287
9.4.1	概述.....	287
9.4.2	工程示例.....	289

本章要点	293
习题	293
第 10 章 ARM9 处理器	295
10.1 ARM9TDMI 特点及其与 ARM7TDMI 的比较	295
10.2 ARM920T 处理器	297
10.2.1 ARM920T 功能模块	297
10.2.2 ARM920T 功能概述	297
10.3 S3C2410X 芯片	301
10.3.1 AMBA 总线	301
10.3.2 S3C2410X 框图	302
10.3.3 S3C2410X 功能概述	303
本章要点	308
习题	308
附录 A ARM 指令集和 Thumb 指令集速查表	309
附录 B 伪操作与伪指令	317
参考文献	322

第 1 章 概 述

1.1 嵌入式系统

1.1.1 嵌入式系统的定义

嵌入式系统是一个相对模糊的概念，国内一般定义为：以应用为中心，计算机技术为基础，软硬件可裁剪，以适应应用系统对功能、可靠性、成本、体积、功耗有严格要求的专用计算机系统。

由此可以看出嵌入式系统实际上是嵌入式计算机系统，不过它是嵌入到更大的、专用的应用系统中的计算机系统，是实际应用系统的一个部件。因此，也有人把嵌入式系统定义为：嵌入到对象体系中的专用计算机系统。对象系统是指嵌入式系统所嵌入的宿主系统。

1.1.2 嵌入式系统的发展历史

嵌入式计算机系统起源于微型机时代，近几年网络、通信、多媒体技术的发展为嵌入式系统应用开辟了广阔的天地，使嵌入式系统成为继 PC 和 Internet 之后，IT 界新的技术热点。

20 世纪 70 年代发展起来的微型计算机，由于体积小、功耗低、结构简单、可靠性高、使用方便、性能价格比高等一系列优点，得到了广泛的应用和迅速的普及。微型机表现出的智能化水平引起了控制专业人士的兴趣，要求将微型机嵌入到一个对象体系中，实现对象体系的智能化控制。例如，将微型计算机经电气加固和机械加固，并配置各种外围接口电路，安装到大型舰船中构成自动驾驶仪或轮机状态监测系统。这样一来，计算机便失去了原来的形态与通用的计算机功能。为了区别原有的通用计算机系统，把嵌入到对象体系中、实现对象体系智能化控制的计算机，称为嵌入式计算机系统。由此可见，嵌入式系统的嵌入性本质是将一个计算机嵌入到一个对象体系中去。

尽管嵌入式系统起源于微型机时代，但微型计算机的体积、价位和可靠性都无法满足广大对象系统的嵌入式应用要求，因此，嵌入式系统必须走独立发展的道路。这条道路就是单芯片化道路，即将计算机做在一个芯片上，从而开创了嵌入式系统独立发展的单片机时代。1976 年，Intel 公司推出了 MCS-48 单片机，这个只有 1KB ROM 和 64B RAM 的简单芯片成为世界上第一个单片机，同时也开创了将微处理机系统的各种 CPU 外的资源（如 ROM、RAM、定时器、并行口、串行口及其他各种功能模块）集成到 CPU 硅片上的时代。1980 年，Intel 公司对 MCS-48 单片机进行了全面完善，推出了 8 位 MCS-51 单片机，并获得巨大成功，奠定了嵌入式系统的单片机应用模式。至今，MCS-51 单片机仍在大量使用。1984 年，Intel 公司又推出了 16 位 8096 系列并将其称之为嵌入式微控制器，这可能是“嵌入式”一词第一次在微处理机领域出现。

此外，为了高速、实时地处理数字信号，1982年诞生了首枚数字信号处理芯片（DSP），DSP是模拟信号转换成数字信号以后进行高速实时处理的专业处理器，其处理速度比当时最快的CPU还快10~50倍。随着集成电路技术的发展，DSP芯片的性能不断提高，目前已广泛用于通信、控制、计算机等领域。

在近30年的历史中，各种改进的、面向具体应用的不同品牌单片机风起云涌，得到了广泛应用，但这些应用基本上是基于硬件底层的单线程程序。20世纪90年代后，伴随着网络时代的来临，网络、通信、多媒体技术得以发展，8/16位单片机在速度和内存容量上已经很难满足这些领域的应用需求。而由于集成电路技术的发展，32位微处理器价格不断下降，综合竞争能力已可以和8/16位单片机媲美。32位微处理器面向嵌入式系统的高端应用，由于速度快，资源丰富，加上应用本身的复杂性、可靠性要求等，软件的开发一般会需要操作系统平台支持。近几年，嵌入式设备（内部有嵌入式系统的产品）大量涌现，如手机、PDA、MP3、微波炉、数码相机、机顶盒、各种网络设备等。嵌入式系统开发应用需求越来越大，使嵌入式系统成为继PC和Internet之后IT技术的最热点，而构成嵌入式系统的主流趋势是32位嵌入式微处理器加实时多任务操作系统，目前的嵌入式系统往往指的是包含这种资源的系统。

1.1.3 嵌入式计算机系统与通用计算机系统

嵌入式计算机系统是相对于通用计算机系统而言的，通用计算机系统要求满足各种不同的应用需求，因而要求有丰富的硬件资源、完善的操作系统、高速的运算和海量的存储，技术发展方向是总线速度的无限提升，存储容量的无限扩大。而嵌入式计算机系统则是面向具体应用，要有针对具体应用的“量体裁衣”的软、硬件，操作系统一般采用实时操作系统，技术发展方向是与对象系统密切相关的嵌入性能、控制能力与控制的可靠性。嵌入式计算机系统同通用计算机系统相比具有以下特点：

- 嵌入式系统通常是面向特定应用的，应用的多样性决定了硬件平台的多样性。嵌入式CPU与通用型的CPU最大不同就是嵌入式CPU大多工作在为特定用户群设计的系统中，它通常都具有低功耗、体积小、集成度高等特点，能够把通用系统中许多由单独芯片或板卡完成的功能集成在芯片内部，从而有利于嵌入式系统设计趋于小型化、低功耗，移动能力大大增强，跟网络的耦合也越来越紧密。
- 大多嵌入式系统都有实时性要求，在高端应用中，为满足应用需求、增强可靠性和便于开发，往往要有实时多任务操作系统的支持。
- 嵌入式系统是将先进的计算机技术、半导体技术和电子技术与各个行业的具体应用相结合后的产物。这一点就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。
- 功耗、成本和可靠性对嵌入式系统有特别重要的意义。
- 嵌入式系统和具体应用有机地结合在一起，它的升级换代也是和具体产品同步进行的，因此嵌入式系统产品进入市场后具有较长的生命周期。
- 嵌入式系统本身不具备自主开发能力，即使设计完成以后，用户通常也不能对其中的程序功能进行修改，必须有一套交叉开发工具和环境才能进行开发。

- 嵌入式系统工业是不可垄断的高度分散的工业，充满了竞争、机遇与创新，是一个可以大有作为的行业。

1.2 嵌入式系统的组成

嵌入式系统由硬件和软件组成，两类不同的嵌入式系统结构模型见图 1.1。硬件是整个嵌入式操作系统和应用程序运行的平台，不同的应用通常有不同的硬件环境。嵌入式系统的硬件部分包括处理器/微处理器、存储器、I/O 接口及输入/输出设备。嵌入式系统的软件由嵌入式操作系统和应用程序组成。嵌入式操作系统完成嵌入式应用的任务调度和控制等核心功能，嵌入式应用程序运行于操作系统之上（对于一些简单的嵌入式应用系统，应用程序可以不需要操作系统的支持，直接运行在底层，见图 1.1（a）），利用操作系统提供的机制完成特定功能的嵌入式应用。

由于嵌入式系统的灵活性和多样性，图 1.1 中各个层次之间缺乏统一的标准，几乎每一个独立的系统都不一样，这样就给上层的软件设计人员开发应用程序带来了极大的困难。

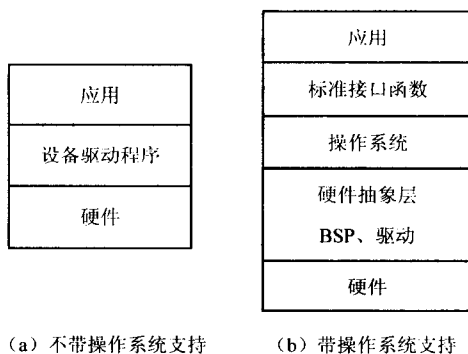


图 1.1 两类不同的嵌入式系统结构模型

1.2.1 嵌入式系统的硬件组成

嵌入式系统的硬件组成见图 1.2。

1. 嵌入式处理器

嵌入式系统的核心是各种类型的嵌入式处理器，嵌入式处理器的体系结构经历了从 CISC（复杂指令集）到 RISC（精简指令集）和 Compact RISC 的转变，位数则由 4 位、8 位、16 位、32 位逐步发展到 64 位。现在常用的嵌入式处理器可分为低端的嵌入式微控制器（microcontroller unit, MCU）、中高端的嵌入式微处理器（embedded microprocessor unit, EMPU）、嵌入式 DSP 处理器（embedded digital signal processor, EDSP）和高度集成的嵌入式片上系统（system on a chip, SoC）。目前几乎每个半导体制造商都生产嵌入式处理器，并且越来越多的公司开始拥有自主的处理器设计部门。据不完全统计，

全世界嵌入式处理器已经超过 1000 种，流行的体系结构有 30 多个系列，其中以 ARM、PowerPC、MC 68000、MIPS 等使用得最为广泛。

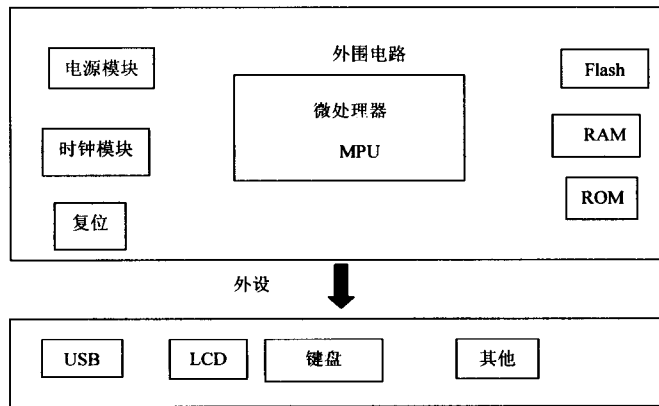


图 1.2 嵌入式系统的硬件组成

2. 存储器

嵌入式系统有别于一般的通用计算机系统，它不具备像硬盘那样大容量的存储介质，而用静态易失型存储器（RAM、SRAM）、动态存储器（DRAM）和非易失型存储器（ROM、EPROM、EEPROM、FLASH）作为存储介质，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。

3. I/O 接口

I/O 接口是处理器与 I/O 设备连接的桥梁，与通用 CPU 不同的是嵌入式处理器芯片将通用机中许多由单独芯片或板卡完成的接口功能集成到芯片内部，从而有利于嵌入式系统在设计时趋于小型化，同时还具有很高的效率和可靠性。

4. 输入/输出设备

为使嵌入式系统具有友好的界面、方便人机交互，嵌入式系统中需配制输入、输出设备，常用的输入/输出设备有液晶显示器（LCD）、触摸板、键盘等。

嵌入式开发的硬件平台选择主要是嵌入式处理器的选择。在具体应用中处理器的选择决定了其市场竞争力。在一个系统中使用什么样的嵌入式处理器主要取决于应用领域、用户的需求、成本、开发的难易程度等因素。在开发过程中，选择最适用的硬件平台是一项很复杂的工作，包括要考虑其他工程的影响以及缺乏完整或准确的信息等。

1.2.2 嵌入式系统的软件组成

1. 嵌入式操作系统

嵌入式操作系统是一种支持嵌入式系统应用的操作系统软件，它是嵌入式系统（包括硬、软件系统）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、图形界面、标准化浏览器等。嵌入式操作系统具有通用

操作系统的基本特点,如能够有效管理越来越复杂的系统资源,能够把硬件虚拟化、使得开发人员从繁忙的驱动程序移植和维护中解脱出来,能够提供库函数、驱动程序、工具集以及应用程序。与通用操作系统相比较,嵌入式操作系统在系统的实时高效性、硬件的相关依赖性、软件固态化以及应用的专用性等方面具有较为突出的特点。

2. 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,而且还要尽可能地优化,以减少对系统资源的消耗,降低硬件成本。

3. 硬件抽象层 HAL

硬件抽象层 HAL (hardware abstraction layer) 是位于操作系统内核与硬件电路之间的接口层,其目的在于将硬件抽象化。也就是说,可通过程序来控制所有硬件电路,如 CPU、I/O、存储器等的操作。这样就使得系统的设备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。

4. 板级支持包 BSP

板级支持包 BSP (board support package) 是介于主板硬件和操作系统中驱动层程序之间的一层,一般认为它属于操作系统的一部分,主要是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,使之能够更好地运行于硬件主板。

5. 设备驱动程序

计算机系统中安装设备后,只有在安装相应的设备驱动程序之后才能使用,驱动程序为上层软件提供设备的操作接口。上层软件只需调用驱动程序提供的接口,而不用理会设备的具体内部操作。

6. 操作系统的应用程序接口函数 API

API (application programming interface) 是一系列复杂的函数、消息和结构的集合体。嵌入式操作系统下的 API 和一般操作系统下的 API 在功能、含义及知识体系上完全一致。

嵌入式应用软件是实现嵌入式系统功能的关键,对嵌入式系统软件和应用软件的要求也和通用计算机有所不同,嵌入式软件主要有以下一些特点。

- 软件要求固化存储。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嵌入式微处理器本身中,而不是存储于磁盘等载体中。
- 软件代码要求高质量、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,存储空间仍然是宝贵的,还存在实时性的要求。为此,要求程序编写和编译工具的质量要高,以减小程序二进制代码长度、提高执行速度。