




冯新宇 霍滨焱 吴岩◎编著

ARM 9

嵌入式开发基础 与实例进阶

 赠配书光盘：包含了实例源文件及相关学习资料。

- 从零开始，轻松入门
- 图解案例，清晰直观
- 图文并茂，操作简单
- 实例引导，专业经典
- 学以致用，注重实践



清华大学出版社

ARM 9 嵌入式开发基础与实例进阶

冯新宇 霍滨焱 吴 岩 编著

清华大学出版社

北 京

内 容 简 介

本书主要包括嵌入式 Linux 系统管理与编程基础、Linux 应用程序设计、ARM 程序设计、内核开发与系统移植、驱动程序开发等 5 部分内容。

本书作者都是长期使用 Linux 系统进行教学、科研和实际生产工作的教师和工程师，具备丰富的教学和实践经验。本书在内容编排上，遵循读者学习的一般规律，结合大量实例讲解操作步骤，能够使读者快速、真正地掌握嵌入式 Linux 系统的基本原理和应用。

本书可作为大中专院校相关专业的教材，也可以作为嵌入式开发爱好者的自学教程和专业人员的参考手册。

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

ARM 9 嵌入式开发基础与实例进阶/冯新宇，霍滨焱，吴岩编著. —北京：清华大学出版社，2012.1

ISBN 978-7-302-26295-4

I. ①A… II. ①冯… ②霍… ③吴… III. ①微处理器，ARM 9-系统设计-教材 IV. ①TP332

中国版本图书馆 CIP 数据核字 (2011) 第 141802 号

责任编辑：钟志芳

封面设计：刘 超

版式设计：文森时代

责任校对：张兴旺

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 装 者：三河市金元印装有限公司

经 销：全国新华书店

开 本：185×260 印 张：23.5 字 数：543 千字

(附 DVD 光盘 1 张)

版 次：2012 年 1 月第 1 版

印 次：2012 年 1 月第 1 次印刷

印 数：1~4000

定 价：49.80 元

产品编号：040502-01

前 言

关于嵌入式系统

嵌入式系统无疑是当前最热门、最有发展前途的 IT 应用领域之一。随着计算机技术的飞速发展，嵌入式系统应用已深入到国民经济的各个方面，在工业控制、家用电器、智能仪器仪表、机电控制等领域，不断展现出其独特魅力。与人们日常生活息息相关的诸多设备，如手机、PDA、电子字典、可视电话、数码相机、数码摄像机、机顶盒、高清电视、游戏机、智能玩具、交换机、路由器、数控设备或仪表、汽车电子、家电控制系统、医疗仪器、航天航空设备等都是典型的嵌入式系统，可以说，它无处不在。

那么到底哪些专业的人员适合学习嵌入式？如何学习？

嵌入式系统是软硬结合的系统。从事嵌入式开发的人员主要分为两类：

一类是电子工程、通信工程等专业人员，他们主要从事硬件设计，有时需要开发一些与硬件关系最密切的最底层软件，如 **BootLoader**、**Board Support Package**、最初级的硬件驱动程序等。他们的优势是对硬件原理非常清楚，不足之处是对复杂软件系统的原理往往不熟悉（例如嵌入式操作系统原理和复杂应用软件原理等）。

另一类是计算机专业人员，他们主要从事嵌入式操作系统和应用软件的开发。

如果学习软件的人对硬件原理和接口有较好的掌握，那么完全可以开发 **BSP** 和硬件驱动程序。嵌入式硬件设计完成后，各种功能就全靠软件来实现了，嵌入式设备的增值很大程度上取决于嵌入式软件，开发嵌入式系统的最主要工作便在于此。越是智能设备，越是复杂系统，软件越起关键作用。

目前国内外这方面的人才非常稀缺。一方面，是因为这一领域入门门槛较高，不仅要懂底层软件（例如操作系统级、驱动程序级软件），对软件专业水平要求较高（嵌入式系统对软件设计的时间和空间效率要求较高），而且必须懂得硬件的工作原理，所以非专业 IT 人员很难进入这一领域；另一方面，是因为这一领域较新，发展太快，很多软硬件技术出现时间不长或正在出现（如 **ARM** 处理器、嵌入式操作系统、**MPEG** 技术、无线通信协议等），掌握这些新技术的人当然很难找。嵌入式人才稀缺，身价自然就高，而经验丰富的开发人员更是如此。其实嵌入式人才稀少，根本原因可能是大多数人无条件接触相关知识（这需要有相应的嵌入式开发板和软件；另外，还需要有经验的人进行指导）。

与企业计算等应用软件不同，嵌入式领域人才的工作强度通常低一些、收入却很高。对于从事企业应用软件开发的公司来说，一个用户的系统开发完了，又得去开发下一个用户的，而且每个用户的需求和完成时间都得随客户的要求而改变，往往疲于奔命，重复劳动。相比而言，研发嵌入式系统的公司都有自己的产品计划，按自己的节奏行事。它所开发的产品通常是通用的，不会因客户的不同而修改。一个产品型号开发完了，开发人员

ARM 9 嵌入式开发基础与实例进阶

往往有较长一段空闲时间，可以对软件进行一些小修补，或者进行“充电”和休整。另外，从事嵌入式软件研发的人员工作范围相对较窄，所涉及的专业技术范围有限（ARM、RTOS、MPEG、802.11 等），随着时间的推移，其设计经验会愈加丰富。

学习嵌入式，显然应偏重于嵌入式软件，特别是嵌入式操作系统方面。对于从事嵌入式软件开发的人员来说，最重要的显然是：

- ◇ 掌握主流嵌入式微处理器的结构与原理。
- ◇ 掌握一种嵌入式操作系统。
- ◇ 熟悉嵌入式软件开发流程并至少参与开发过一个嵌入式软件项目。

嵌入式系统开发最难的就是入门。正如上面所说的，嵌入式系统开发涉及软件、硬件等各方面的知识，初学者很难从纷杂的知识中快速上手。幸运的是，现在市面上用于嵌入式开发的学习板比比皆是，价格都比较低廉，读者可以购买一款相对通用的开发板，然后按照本书中的讲解一步一步进行实际操作，即可轻松进入嵌入式开发的大门。任何知识的学习都是由浅入深，由感性认识到理性认识的过程，只要坚持学习，相信一定能掌握嵌入式开发的要领。

基本内容

本书主要包括 5 部分内容：嵌入式 Linux 系统管理与编程基础、Linux 应用程序设计、ARM 程序设计、内核开发与系统移植、驱动程序开发等，书中全面、系统地介绍这几部分知识，对于初学者，特别是嵌入式系统应用开发入门者来说，一定会有所帮助。

主要特点

具体地讲，本书具有以下鲜明的特点：

- ◇ 从零开始，轻松入门
- ◇ 图解案例，清晰直观
- ◇ 图文并茂，操作简单
- ◇ 实例引导，专业经典
- ◇ 学以致用，注重实践

读者对象

本书面向嵌入式开发的初中级读者，主要包括：

- ◇ 学习嵌入式开发的初级读者。
- ◇ 具有一定嵌入式基础知识，希望进一步深入嵌入式开发的中级读者。
- ◇ 大中专院校电子信息及计算机专业的学生。
- ◇ 从事嵌入式开发的工程技术人员。

配套光盘简介

为了方便读者学习，本书提供了配套多媒体教学光盘。其中包含了本书实例源文件和数据，这些文件都被保存在与章节相对应的文件夹中。

注意：由于光盘上的文件都是“只读”的，因此不能直接修改。读者可以先将这些文件复制到硬盘上，去掉文件的“只读”属性，然后再使用。

本书第1~4章由冯新宇编写，第5~10章由李永新编写，第11~12章由吴岩编写，第13~16章由霍滨焱编写，书稿的最后统编由冯新宇负责。参与编写的人员还有边莉、刘宇莹、刘琳、史殿发、孟莹等。同时，感谢宋一兵、管殿柱、付本国、赵景伟、赵景波、王献红、张忠林、王臣业、谈世哲等老师的大力协助。

感谢广州碾展公司的技术支持！

感谢黑龙江省惠业人力资源有限责任公司的大力协助！

感谢您选择了本书，希望我们的努力对您的工作和学习有所帮助，也希望您把对本书的意见和建议告诉我们。

零点工作室网站地址：www.zerobook.net。

零点工作室联系邮箱：gdz_zero@126.com。

零点工作室
2011年12月

目 录

第 1 章 Linux 概述与系统管理	1
1.1 嵌入式系统概述	1
1.1.1 嵌入式系统发展历史	1
1.1.2 嵌入式系统的特点	2
1.1.3 嵌入式系统的体系结构	2
1.1.4 典型嵌入式系统介绍	3
1.2 嵌入式 Linux 基础	4
1.2.1 Linux 发行版本	4
1.2.2 Linux 定制安装	6
1.3 Linux 常用命令	11
1.3.1 用户系统相关命令	11
1.3.2 文件操作相关命令	12
1.3.3 压缩打包相关命令	17
1.3.4 网络相关命令	18
1.3.5 其他常用命令	21
1.4 服务器配置	25
1.4.1 Samba 服务器	25
1.4.2 实例进阶一：Samba 服务器配置	26
1.4.3 NFS 服务器	27
1.4.4 实例进阶二：NFS 服务器配置	28
1.4.5 TFTP 服务器	29
1.4.6 实例进阶三：TFTP 服务器配置	29
1.5 思考与练习	30
第 2 章 Linux 脚本编程	32
2.1 常用 Shell 命令	32
2.2 脚本编写基础	33
2.2.1 特殊字符	33
2.2.2 变量和参数	35
2.2.3 退出和退出状态	37
2.3 流程控制	39
2.3.1 条件测试	39
2.3.2 操作符	41

ARM 9 嵌入式开发基础与实例进阶

2.3.3	循环控制.....	43
2.3.4	测试与分支.....	45
2.4	实例进阶.....	47
2.4.1	实例进阶一：参数大小比较.....	47
2.4.2	实例进阶二：当前目录下的文件数目统计.....	48
2.5	思考与练习.....	48
第 3 章	Linux 编程基础.....	50
3.1	编辑器介绍.....	50
3.1.1	vi 介绍.....	50
3.1.2	vi 的各模式功能键.....	50
3.2	程序编译与调试.....	52
3.2.1	gcc 编译流程.....	52
3.2.2	gcc 编译选项分析.....	53
3.2.3	gdb 程序调试.....	54
3.3	实例进阶.....	55
3.3.1	实例进阶一：vi 使用演示.....	55
3.3.2	实例进阶二：gcc 使用演示.....	58
3.3.3	实例进阶三：gdb 使用演示.....	60
3.4	思考与练习.....	65
第 4 章	文件 I/O 编程.....	66
4.1	基于 Linux 系统的文件编程.....	66
4.1.1	文件打开.....	66
4.1.2	文件创建.....	68
4.1.3	文件关闭.....	68
4.1.4	文件读取.....	69
4.1.5	数据写入.....	69
4.1.6	移动文件的读写位置.....	69
4.1.7	文件描述词操作.....	70
4.1.8	锁定文件或解除锁定.....	71
4.2	基于 C 语言库函数的文件编程.....	71
4.2.1	文件打开.....	72
4.2.2	从文件流读取数据.....	73
4.2.3	将数据写至文件流.....	74
4.2.4	由文件中读取一个字符.....	75
4.2.5	由文件中读取一字符串.....	75
4.2.6	将一指定字符写入文件流中.....	76
4.2.7	将一指定的字符串写入文件内.....	77

4.2.8	格式化字符串输入	77
4.2.9	格式化输出数据至文件	78
4.2.10	移动文件流的读写位置	79
4.3	Linux 时间编程	80
4.3.1	取得目前的时间	80
4.3.2	取得目前时间和日期	81
4.3.3	取得当地目前时间和日期	82
4.3.4	将时间结构数据转换成经过的秒数	83
4.3.5	设置目前时间	84
4.3.6	取得当前时间	84
4.3.7	将时间和日期以 ASCII 格式表示	85
4.3.8	将时间和日期以字符串格式表示	86
4.4	实例进阶	87
4.4.1	实例进阶一：文件创建	87
4.4.2	实例进阶二：时间转换	88
4.5	思考与练习	88
第 5 章	进程控制	90
5.1	进程控制概述	90
5.1.1	进程的定义	90
5.1.2	进程控制块	90
5.1.3	分配进程描述符	91
5.1.4	进程描述符的存放	92
5.1.5	进程状态	93
5.1.6	进程调度	94
5.2	进程控制编程	94
5.2.1	获取 ID	94
5.2.2	进程创建	96
5.2.3	exec 函数族	99
5.2.4	system()	102
5.2.5	等待函数	103
5.2.6	退出	106
5.3	实例进阶	107
5.3.1	实例进阶一：进程创建	107
5.3.2	实例进阶二：exec 函数族使用	108
5.4	思考与练习	109
第 6 章	进程间通信	110
6.1	进程间通信概述	110

6.2	管道通信.....	112
6.2.1	创建管道.....	112
6.2.2	建立管道 I/O.....	113
6.2.3	关闭管道.....	113
6.2.4	建立命名管道.....	114
6.3	消息队列.....	115
6.3.1	键值.....	115
6.3.2	打开/创建消息队列.....	116
6.3.3	发送消息.....	117
6.3.4	接收消息.....	117
6.3.5	消息控制.....	118
6.4	信号.....	118
6.4.1	信号发送与捕捉.....	118
6.4.2	信号处理.....	121
6.4.3	信号集.....	124
6.5	信号量.....	126
6.5.1	信号量创建.....	126
6.5.2	信号量操作.....	127
6.6	内存共享.....	129
6.6.1	共享内存创建.....	129
6.6.2	共享内存的操作.....	131
6.6.3	共享内存段连接到本进程空间.....	132
6.6.4	共享内存解除.....	132
6.7	实例进阶.....	134
6.7.1	实例进阶一：通过管道输出字符.....	134
6.7.2	实例进阶二：消息队列综合应用.....	134
6.8	思考与练习.....	137
第 7 章	多线程技术.....	138
7.1	Linux 多线程概念.....	138
7.2	Linux 线程实现.....	139
7.2.1	线程创建.....	139
7.2.2	线程退出.....	139
7.2.3	线程等待.....	140
7.2.4	线程标识获取.....	140
7.2.5	线程清除.....	140
7.3	修改线程的属性.....	141
7.3.1	绑定属性.....	142

7.3.2	分离属性	143
7.3.3	优先级属性	144
7.4	实例进阶	145
7.4.1	实例进阶一：创建线程	145
7.4.2	实例进阶二：线程退出	146
7.4.3	实例进阶三：用 <code>pthread_join()</code> 实现线程等待	147
7.4.4	实例进阶四：获取线程 ID	148
7.4.5	实例进阶五：线程清除	149
7.5	思考与练习	151
第 8 章	网络编程	152
8.1	基本概念	152
8.1.1	OSI 模型	152
8.1.2	常用的命令	153
8.2	TCP/IP 协议	153
8.2.1	整体构架概述	153
8.2.2	IP 协议	155
8.2.3	ICMP 协议	155
8.2.4	UDP 协议	156
8.2.5	TCP 协议	156
8.2.6	TCP 连接的建立	157
8.3	基本网络函数介绍	157
8.3.1	建立一个 Socket 通信	158
8.3.2	对 Socket 定位	158
8.3.3	等待连接	159
8.3.4	建立 Socket 连接	160
8.4	服务器和客户机的信息函数	161
8.4.1	字节转换函数	161
8.4.2	IP 和域名的转换	163
8.4.3	字符串形式的 IP 和 32 位的 IP 转换	163
8.4.4	服务信息函数	164
8.5	完整的读写函数	164
8.5.1	<code>write()</code>	164
8.5.2	<code>read()</code>	165
8.5.3	数据的传递	165
8.6	用户数据报发送	166
8.6.1	<code>recvfrom()</code>	166
8.6.2	<code>sendto()</code>	166

ARM 9 嵌入式开发基础与实例进阶

8.7 高级套接字函数.....	167
8.7.1 recv().....	167
8.7.2 send().....	168
8.7.3 recvmsg().....	168
8.7.4 sendmsg().....	169
8.7.5 套接字的关闭.....	170
8.8 套接字选项.....	170
8.8.1 getsockopt().....	170
8.8.2 setsockopt().....	171
8.8.3 ioctl().....	172
8.9 服务器模型.....	173
8.9.1 循环服务器：UDP 服务器.....	173
8.9.2 循环服务器：TCP 服务器.....	173
8.9.3 并发服务器：TCP 服务器.....	174
8.9.4 并发服务器：多路复用 I/O.....	174
8.9.5 并发服务器：UDP 服务器.....	176
8.10 实例进阶.....	176
8.10.1 实例进阶一：IP 与域名转换.....	176
8.10.2 实例进阶二：编写 UDP 协议程序实现数据通信.....	178
8.11 思考与练习.....	181
第 9 章 ARM 基础知识.....	182
9.1 ARM 概述.....	182
9.1.1 ARM 处理器工作状态与工作模式.....	182
9.1.2 ARM 寄存器组织结构.....	183
9.1.3 ARM 异常处理机制.....	185
9.1.4 ARM 存储器结构.....	186
9.2 ARM 指令系统.....	187
9.2.1 ARM 指令格式.....	187
9.2.2 ARM 指令寻址方式.....	188
9.3 ARM 指令集.....	190
9.3.1 数据处理指令.....	191
9.3.2 分支指令.....	196
9.3.3 存储器访问（Load/Store）指令.....	197
9.3.4 状态寄存器访问指令.....	201
9.3.5 协处理器指令.....	202
9.3.6 SWI 软中断指令.....	204
9.4 思考与练习.....	205

第 10 章 ARM 程序设计	206
10.1 ADS 集成开发环境.....	206
10.1.1 ADS 工具包.....	206
10.1.2 命令行开发工具.....	206
10.1.3 CodeWarrior IDE (集成开发环境) 的使用.....	207
10.1.4 AXD 调试器的使用.....	214
10.2 实例进阶.....	216
10.2.1 实例进阶一: LED 程序设计.....	216
10.2.2 实例进阶二: 按键程序设计.....	218
10.2.3 实例进阶三: 串口程序设计.....	220
10.3 思考与练习.....	224
第 11 章 内核开发基础	225
11.1 嵌入式开发环境的搭建.....	225
11.1.1 交叉编译工具链.....	226
11.1.2 终端软件.....	227
11.2 Linux 内核简介.....	229
11.2.1 Linux 内核.....	229
11.2.2 Linux 内核源代码.....	231
11.3 Linux 内核配置与编译.....	232
11.3.1 Linux 内核配置.....	232
11.3.2 编译内核.....	234
11.4 Linux 内核模块.....	235
11.4.1 内核模块简介.....	235
11.4.2 内核模块编译与相关命令.....	236
11.5 文件系统.....	236
11.5.1 文件系统概述.....	236
11.5.2 根文件系统.....	238
11.5.3 Busybox.....	240
11.5.4 Ramdisk 文件系统.....	245
11.6 实例进阶.....	246
11.6.1 实例进阶一: 安装交叉编译工具链 arm-linux-gcc-4.3.3.....	246
11.6.2 实例进阶二: x86 平台内核配置、编译.....	247
11.7 思考与练习.....	248
第 12 章 BootLoader	250
12.1 BootLoader 介绍.....	250
12.1.1 BootLoader 的安装和启动过程.....	250

ARM 9 嵌入式开发基础与实例进阶

12.1.2	BootLoader 的操作模式	251
12.1.3	BootLoader 与主机之间的通信方式	251
12.1.4	常用 BootLoader 介绍	252
12.2	U-Boot 介绍	252
12.2.1	目录结构	253
12.2.2	U-Boot 主要功能	254
12.2.3	U-Boot 的工具	254
12.3	U-Boot 命令	254
12.3.1	获取帮助	255
12.3.2	环境变量	256
12.3.3	文件下载	258
12.3.4	内存操作	260
12.3.5	Flash 操作	261
12.3.6	执行程序	264
12.4	U-Boot 工作流程	264
12.4.1	stage1	264
12.4.2	stage2	266
12.5	U-Boot 编译	266
12.6	实例进阶: U-Boot 移植	267
12.7	思考与练习	270
第 13 章	Linux 驱动开发基础与调试	271
13.1	设备驱动简介	271
13.1.1	设备类型划分	271
13.1.2	内核空间和用户空间	272
13.1.3	驱动程序层次结构	273
13.1.4	驱动程序与外界的接口	274
13.2	打印调试	276
13.3	思考与练习	278
第 14 章	字符设备驱动	279
14.1	字符设备驱动程序基础	279
14.1.1	关键数据结构	279
14.1.2	设备驱动开发的基本函数	281
14.1.3	设备文件和设备号	286
14.1.4	加载和卸载驱动程序	287
14.2	实例进阶一: LED 设备驱动程序	288
14.2.1	LED 接口电路	289
14.2.2	LED 驱动程序	289

14.2.3	加载运行 LED 驱动程序	292
14.3	实例进阶二：按键设备驱动程序	293
14.3.1	按键模块硬件电路	294
14.3.2	按键驱动程序	295
14.3.3	加载、运行按键驱动程序	298
14.4	思考与练习	300
第 15 章	块设备驱动	301
15.1	块设备驱动程序开发基础	301
15.1.1	块设备的 I/O 操作特点	301
15.1.2	块设备主要数据结构	301
15.1.3	块设备的操作	310
15.2	实例进阶：IDE 硬盘设备驱动	316
15.2.1	IDE 硬盘设备原理	316
15.2.2	S3C2440 与 IDE 接口电路	318
15.2.3	block_device_operations 及成员函数	319
15.2.4	I/O 请求处理	321
15.2.5	在内核中增加对新系统 IDE 设备的支持	328
15.3	思考与练习	333
第 16 章	网络设备驱动	334
16.1	网络设备驱动简介	334
16.1.1	驱动程序体系结构	334
16.1.2	主要数据结构	335
16.1.3	基本函数	337
16.2	CS8900A 网卡设备	339
16.2.1	CS8900A 简介	340
16.2.2	CS8900A 网卡接口电路	341
16.3	实例进阶：CS8900A 设备驱动程序	341
16.3.1	初始化网络设备	342
16.3.2	打开网络设备	347
16.3.3	关闭网络设备	349
16.3.4	中断处理	349
16.3.5	发送数据	352
16.3.6	接收数据	354
16.4	思考与练习	355
参考文献	357

第 1 章 Linux 概述与系统管理

随着计算机技术的发展，嵌入式 Linux 系统在嵌入式处理器中的应用越来越广泛。这就意味着只有熟练掌握了 Linux 系统，才可能在嵌入式开发领域得心应手。本章将从嵌入式系统的基本概念入手，了解嵌入式系统的发展；在应用基础上，将详细阐述 Linux 操作系统的安装和使用方法，使读者可以对 Linux 常用命令和系统的管理方法有一个全面的认识。

1.1 嵌入式系统概述

嵌入式系统是以应用为中心、以计算机技术为基础并且软硬件可裁剪，适用于应用系统对功能、可靠性、成本、体积、功耗有严格要求的专用计算机系统。

嵌入式系统是把计算机直接嵌入到应用系统中，融合了计算机软/硬件技术、通信技术和微电子技术。随着微电子技术和半导体技术的高速发展，超大规模集成电路技术和深亚微米制造工艺已十分成熟，从而使高性能系统芯片的集成成为可能，并推动着嵌入式系统向最高级构建形式，即片上系统（System On a Chip, SOC）的方向发展，进而促使嵌入式系统得到更深入、更广泛的应用。嵌入式技术的快速发展不仅使其成为当今计算机技术和电子技术的一个重要分支，同时也使计算机的分类从以前的巨型机、大型机、小型机和微型机变为通用计算机和嵌入式计算机（即嵌入式系统）。

1.1.1 嵌入式系统发展历史

20 世纪 80 年代初，在嵌入式系统的研发中开始采用商业级“操作系统”编写嵌入式应用软件，从而大大降低了开发成本，缩短了开发周期。这些嵌入式操作系统均具有嵌入式技术的典型特点：采用占先式的调度，响应的时间很短，任务执行的时间可以确定；系统内核很小，且具有可剪裁性、可扩充性和可移植性，可以移植到各种型号的微处理器（单片机）上；具有较强的实时性和可靠性，适合于嵌入式应用。

20 世纪 90 年代以后，随着诸多应用领域对嵌入式系统实时性要求的提高，各种应用软件的规模不断扩大，又促使嵌入式系统的实时内核逐渐发展为实时多任务操作系统（RTOS），并作为一种软件平台逐步演变成为目前国际上流行的嵌入式操作系统。

在嵌入式操作系统迅速发展的同时，系统芯片的设计与制造技术也在不断进步。系统芯片就是把一个完整的最终产品的主要功能单元集成到一块或一组大规模集成电路芯片上，这是现代集成电路工艺技术——深亚微米技术迅速发展的必然结果。系统芯片制造技术的快速发展主要体现在硅圆片的尺寸逐渐增大，硅晶片的特征线宽逐步减小，同时芯片

ARM 9 嵌入式开发基础与实例进阶

的集成度不断提高。系统芯片技术的发展，使得嵌入式系统硬件进一步向微型化、高集成化发展，从而为嵌入式系统的应用开辟了更为广阔的天地。

1.1.2 嵌入式系统的特点

嵌入式系统既然是计算机系统，就不可避免地由三大部分构成——CPU、内存和输入/输出设备。此外，当然还得有将这三大部分连接起来的“总线”。这是所有计算机系统的共性，但与以 PC 机为代表的通用计算机系统相比，嵌入式系统有其鲜明的特殊性。其特点概括如下：

- ◇ 具有良好的可靠性与稳定性。
- ◇ 嵌入式系统的软硬件均是面向特定应用对象和任务设计的，具有很强的专用性。
- ◇ 有些嵌入式系统需要长期连续运行（如电话交换机）。
- ◇ 有些要求高可靠性的嵌入式系统还需要采用容错技术，即系统在损坏时能自动切换到其备份，或者对系统进行重构。
- ◇ 许多嵌入式系统都有实时要求，需要具备对外部事件迅速作出反应的能力。
- ◇ 在系统组成上，因为嵌入式系统常常用于控制目的，其外设接口自然就比较多样，并且数量也比较多。
- ◇ 与通用计算机相比，嵌入式系统一般都不带用于大容量存储目的的外部设备，也就是不带磁盘。
- ◇ 许多嵌入式系统的人机界面也有其特殊性。

1.1.3 嵌入式系统的体系结构

嵌入式系统早期主要应用于军事及航空航天等领域，以后逐步广泛应用于工业控制、仪器仪表、汽车电子、通信和家用消费电子类等领域。随着 Internet 的发展，新型的嵌入式系统正朝着信息家电和 3C 产品方向发展。嵌入式系统采用“量体裁衣”的方式把所需的功能嵌入至各种应用系统中。根据应用形式的不同，可分为 IP 级、芯片级和模块级 3 级不同的体系架构。其体系结构的主要特点概括如下：

- ◇ IP 级架构：也就是系统级芯片 SOC 的形式。把不同的 IP 单元，根据应用的要求集成在一块芯片上；各种嵌入式软件也可以以 IP 方式集成在芯片中。
- ◇ 芯片级架构：根据各种 IT 产品的要求，选用相应的处理器芯片、RAM、ROM（EPROM/EEPROM/FLASH）及 I/O 接口芯片等组成相应的嵌入式系统；相应的系统软件或应用软件也以固件形式固化在 ROM 中。这是目前嵌入式系统最常见的形式。
- ◇ 模块级架构：以 X86 处理器构成的计算机系统模块嵌入到应用系统中。这样可充分利用目前常用 PC 机的通用性和便利性。不过，此方式不但要缩小体积、增加可靠性，而且还要把操作系统 OS 改造为嵌入式操作系统 OS，把应用软件固化在固态盘中。这种嵌入式系统较多地出现在工业控制和仪器仪表中。