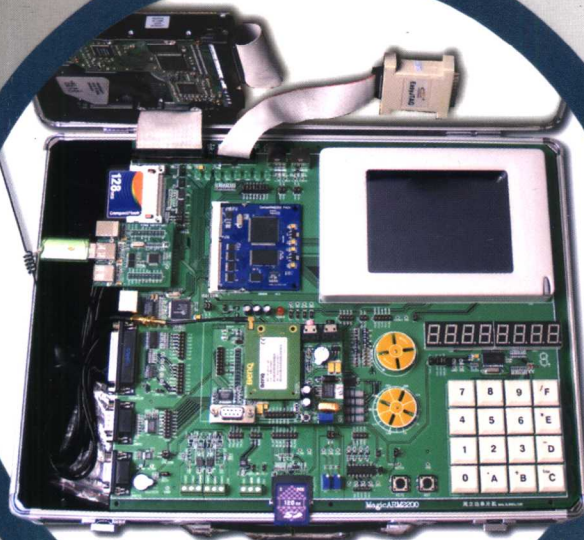


ARM



林晓飞 刘彬 张辉◎编著

基于 ARM

嵌入式Linux应用开发与实例教程

- ◆ 知晓理论 掌握方法 应用实践
- ◆ 典型实例 重点突出 实用性强
- ◆ 选材恰当 深入浅出 可读性强



清华大学出版社

基于 ARM 嵌入式 Linux 应用开发与 实例教程

林晓飞 刘 彬 张 辉 编著

清华大学出版社

北 京

内 容 简 介

本书从理论基础及实际应用的角度出发,不仅详细地讲述了嵌入式系统基础理论方面的知识,还结合具体的实验开发板讲述了嵌入式系统的详细开发方法和步骤。实际案例丰富,内容的可读性、可用性和实践性较强。

本书所讲述的理论基础知识包括嵌入式系统的硬件、软件方面的知识和相关的系统开发方法。硬件方面重点介绍了 ARM 微处理器的知识,软件方面主要介绍了 Linux 操作系统方面的知识。实验开发案例的开发环境基于 ARM9 微处理器和 Linux 操作系统。

本书不仅有详细的理论基础知识介绍,还有大量的开发案例以供参考,可读性和实用性强,可供从事嵌入式系统设计和开发的广大科技人员阅读,也可作为大专院校电子控制专业及其他相关专业的教材或参考资料。

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 13501256678 13801310933

图书在版编目(CIP)数据

基于 ARM 嵌入式 Linux 应用开发与实例教程/林晓飞,刘彬,张辉编著. —北京:清华大学出版社,2007.7
ISBN 978-7-302-15559-1

I. 基… II. ①林… ②刘… ③张… III. ①微处理器, ARM—教材 ②Linux 操作系统—教材 IV. TP332
TP316.89

中国版本图书馆 CIP 数据核字(2007)第 096982 号

责任编辑:邹杰 宋延清

装帧设计:杨玉兰

责任校对:周剑云

责任印制:李红英

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

c-service@tup.tsinghua.edu.cn

社 总 机:010-62770175 邮购热线:010-62786544

投稿咨询:010-62772015 客户服务:010-62776969

印 刷 者:北京市世界知识印刷厂

装 订 者:北京市密云县京文制本装订厂

经 销:全国新华书店

开 本:190×260 印 张:23.75 字 数:573 千字

版 次:2007 年 7 月第 1 版 印 次:2007 年 7 月第 1 次印刷

印 数:1~4000

定 价:36.00 元

本书如存在文字不清、漏印、缺页、倒页、脱页等印装质量问题,请与清华大学出版社出版部联系调换。联系电话:(010)62770177 转 3103 产品编号:022513-01

前 言

近些年来，随着以计算机技术、通信技术为主的信息技术的快速发展和 Internet 的广泛应用，嵌入式系统也受到了越来越多的重视。

嵌入式系统是指用于执行独立功能的专用计算机系统。它由微处理器、定时器、微控制器、存储器、传感器等一系列微电子芯片与器件，和嵌入在存储器中的微型操作系统、控制应用软件组成，以完成诸如实时控制、监视、管理、移动计算、数据处理等各种自动化处理任务。嵌入式系统以应用为中心，以微电子技术、控制技术、计算机技术和通信技术为基础，强调硬件和软件的协同性与整合性，软件及硬件可剪裁，以满足系统对功能、成本、体积和功耗等的要求。

由于嵌入式系统的开发及应用越来越受到人们的关注，因此也有许多介绍该方面的图书出版。已有的图书大多是专一地介绍硬件或软件方面的知识，很少将它们系统地结合在一起讲述。本书则从理论基础及实际应用的角度出发，不仅详细地讲述了嵌入式系统基础理论方面的知识，还结合具体的实验开发板讲述了嵌入式系统的详细开发方法和步骤。实际案例丰富，内容的可读性、可用性和实践性较强。

全书共分两大部分，第一部分主要介绍嵌入式系统的基础理论知识，第二部分主要介绍实验开发环境的建立，并结合具体的实验案例来系统地介绍嵌入式系统的开发方法。全书关于嵌入式系统的硬件介绍以 ARM 微处理器为主，操作系统以 Linux 为主。现分别介绍如下。

第一部分

这一部分包括绪论及前 8 章，主要介绍嵌入式系统硬件及软件方面的基础知识，详细内容如下。

- 绪论：简单介绍嵌入式系统的定义、嵌入式系统的硬件/软件特征和嵌入式操作系统方面的知识，以及学习嵌入式系统的意义。
- 第 1 章：嵌入式系统的硬件构成。主要介绍嵌入式系统硬件及其开发的相关技术。
- 第 2 章：嵌入式处理器介绍。在第 1 章的基础上，详细介绍嵌入式微处理器方面的知识，重点介绍 ARM 微处理器的特点及应用编程方法，最后简单介绍其他一些主流嵌入式微处理器的知识。
- 第 3 章：嵌入式系统的软件构成。在前面介绍完硬件知识后，这一章开始介绍软件方面的知识，主要包括操作系统的组成及各部分的详细知识，以及当今主流操作系统的情况。
- 第 4 章：嵌入式 Linux 介绍。考虑到 Linux 操作系统的特殊优势，本书重点介绍基于 Linux 的嵌入式系统的开发方法。这一章会有详细的介绍。
- 第 5 章：嵌入式系统中的 Bootloader。在嵌入式操作系统中，Bootloader 有着重要的作用，本章将详细介绍。
- 第 6 章：交叉编译。在嵌入式系统开发中，交叉编译是非常重要的开发方法，本章将详细介绍。



- 第7章：嵌入式 Linux 的开发环境。主要介绍嵌入式 Linux 软件开发环境方面的知识。
- 第8章：设备驱动程序。作为嵌入式系统的一项重要应用，设备驱动程序是非常重要的，本章将详细介绍设备驱动程序的知识。

第二部分

这一部分针对具体的开发环境并结合具体的实验案例，介绍嵌入式系统的开发知识，详细内容如下。

- 第9章：GX-ARM9-2410EP 说明。详细介绍 GX-ARM9-2410EP 实验开发板的硬件结构和电路说明。
- 第10章：开发环境的建立。详细介绍实验开发环境的建立步骤。
- 第11章：基础实验部分。详细介绍嵌入式系统的基础实验，包括操作系统的一些配置系数的设置和一些典型实验的实验方法。
- 第12章：高级实验部分。在第11章的基础上，详细介绍一些进阶实验的实验方法。

本书不仅详细介绍嵌入式系统的基础理论知识，还结合具体的实验开发环境详细介绍一些具体实用的开发案例。使读者能够更加容易地掌握嵌入式系统的知识，具备设计一个可实际应用的基于 ARM Linux 的嵌入式系统的能力。

本书可供从事嵌入式系统设计、开发的广大科技人员阅读，也可作为大专院校电子控制专业及其他相关专业的教材或参考资料。

由于作者水平有限，加之本教材的覆盖面广，书中不妥之处在所难免，敬请广大读者批评指正。

编者

目 录

第一部分

绪论	1	1.7.2 编程技术	26
第 1 章 嵌入式系统的硬件构成	3	1.7.3 JTAG 与 IEEE1149 协议 简介	27
1.1 嵌入式处理器	3	1.7.4 3.3V 和 5V 装置的互连	29
1.1.1 嵌入式处理器的品种划分	3	第 2 章 嵌入式处理器介绍	31
1.1.2 嵌入式处理器的选择	6	2.1 ARM 微处理器概述	31
1.2 存储器	7	2.2 ARM 微处理器的应用领域及特点	31
1.2.1 ROM	8	2.2.1 ARM 微处理器的应用领域	31
1.2.2 RAM	8	2.2.2 ARM 微处理器的特点	32
1.3 输入输出设备	8	2.3 ARM 微处理器系列	32
1.3.1 液晶显示	8	2.3.1 ARM7 微处理器系列	32
1.3.2 触摸屏	9	2.3.2 ARM9 微处理器系列	33
1.3.3 语音输入输出技术	11	2.3.3 ARM9E 微处理器系列	33
1.3.4 键盘	11	2.3.4 ARM10E 微处理器系列	34
1.4 电源转换与管理	14	2.3.5 SecurCore 微处理器系列	34
1.4.1 电源 IC 分类	14	2.3.6 StrongARM 微处理器系列	35
1.4.2 电源 IC 的特点	15	2.3.7 Xscale 处理器	35
1.4.3 电源 IC 选用指南	16	2.4 ARM 微处理器结构	35
1.5 接口技术	17	2.4.1 RISC 体系结构	35
1.5.1 并行接口	17	2.4.2 ARM 微处理器的寄存器 结构	36
1.5.2 串口	18	2.4.3 ARM 微处理器的指令结构	36
1.5.3 USB	19	2.5 ARM 微处理器的应用选型	37
1.5.4 PCMCIA 和 CF	19	2.5.1 ARM 微处理器内核的选择	37
1.5.5 红外线接口	21	2.5.2 系统的工作频率	37
1.6 总线	22	2.5.3 芯片内存存储器的容量	37
1.6.1 ISA	22	2.5.4 片内外围电路的选择	37
1.6.2 PCI	23	2.6 ARM 微处理器的编程模型	38
1.6.3 I ² C 总线	23	2.7 ARM 体系结构的存储器格式	38
1.6.4 SPI 总线	24	2.7.1 大端格式	38
1.6.5 PC104 总线	24	2.7.2 小端格式	39
1.6.6 CAN 总线	25	2.8 指令长度及数据类型	39
1.7 嵌入式系统开发中常用的硬件调试 和编程技术	26	2.9 处理器模式	39
1.7.1 微代码支持的串口调试	26		



2.10 寄存器组织	40	2.16 其他嵌入式处理器介绍	71
2.10.1 ARM 状态下的寄存器 组织	40	2.16.1 x86	71
2.10.2 Thumb 状态下的寄存器 组织	42	2.16.2 Power PC	71
2.10.3 程序状态寄存器	44	2.16.3 Motorola 68000	72
2.11 异常	46	2.16.4 MIPS	72
2.11.1 ARM 体系结构所支持的 异常类型	46	第3章 嵌入式系统的软件构成	74
2.11.2 对异常的响应	46	3.1 基础知识	74
2.11.3 从异常返回	47	3.2 操作系统发展史	75
2.11.4 各类异常的具体描述	47	3.2.1 串行处理系统	75
2.11.5 异常进入/退出	48	3.2.2 简单批处理系统	75
2.11.6 异常向量	49	3.2.3 多道程序设计批处理系统	75
2.11.7 异常优先级	49	3.2.4 分时系统	75
2.11.8 应用程序中的异常处理	50	3.2.5 现代操作系统	75
2.12 ARM 微处理器的指令系统	50	3.3 操作系统内核	76
2.12.1 ARM 微处理器的指令的 分类与格式	50	3.3.1 内存管理功能	76
2.12.2 指令的条件域	51	3.3.2 内存分割	76
2.13 ARM 指令的寻址方式	52	3.3.3 虚拟内存	77
2.13.1 立即寻址	52	3.4 进程与中断管理	79
2.13.2 寄存器寻址	53	3.4.1 进程描述与控制	80
2.13.3 寄存器间接寻址	53	3.4.2 并发控制: 互斥与同步	84
2.13.4 基址变址寻址	53	3.4.3 并发控制: 死锁处理	91
2.13.5 多寄存器寻址	53	3.4.4 中断及中断处理	94
2.13.6 相对寻址	54	3.5 调度机制	95
2.13.7 堆栈寻址	54	3.5.1 调度类型	95
2.14 ARM 指令集	54	3.5.2 单处理器调度	97
2.14.1 跳转指令	54	3.5.3 多处理器调度	99
2.14.2 数据处理指令	56	3.5.4 实时调度	101
2.14.3 乘法指令与乘加指令	60	3.6 I/O 设备	104
2.14.4 程序状态寄存器访问指令	62	3.6.1 I/O 设备描述参数	104
2.14.5 加载/存储指令	63	3.6.2 I/O 技术的演变	105
2.14.6 批量数据加载/存储指令	65	3.6.3 I/O 设备逻辑描述	105
2.14.7 数据交换指令	66	3.6.4 I/O 缓冲技术	107
2.14.8 移位指令(操作)	67	3.6.5 磁盘调度	109
2.14.9 协处理器指令	68	3.7 文件管理	111
2.14.10 异常产生指令	70	3.7.1 文件与文件系统	112
2.15 Thumb 指令及应用	70	3.7.2 文件组织与访问	113
		3.7.3 文件共享	115
		3.7.4 记录分块	115
		3.7.5 外围存储设备管理	116

3.8 用户界面	117	4.6.7 uClinux 内核运行方式	148
3.8.1 图形用户界面	118	4.6.8 uClinux 支持的文件系统	148
3.8.2 关键技术	118	4.7 嵌入式设备的文件系统	148
3.9 智能化用户界面	120	4.7.1 闪存介绍	148
3.9.1 Agent 技术	120	4.7.2 第二版扩展文件系统 Ext2fs	149
3.9.2 Agent 技术与用户界面的 结合	123	4.7.3 临时文件系统 tmpfs	149
3.10 主流嵌入式操作系统介绍	124	4.8 日志闪存文件系统版本 2 ——JFFS2	150
3.10.1 Windows CE	124	4.8.1 概述	150
3.10.2 VxWorks	125	4.8.2 JFFS 的设计原理	150
3.10.3 OS-9	126	4.9 嵌入式用户界面	152
3.10.4 QNX	126	4.9.1 XFree86 4.X(带帧缓冲区 支持的 X11R6.4)	152
3.10.5 μC/OS-II	126	4.9.2 Microwindows	153
3.10.6 Linux	126	4.9.3 FLTK	153
第 4 章 嵌入式 Linux 介绍	128	4.9.4 Qt/Embedded	154
4.1 Linux 的基础知识介绍	128	4.10 Microwindows 剖析	155
4.2 Linux 的进程与中断管理机制	131	4.10.1 分层设计	155
4.2.1 Linux 进程状态	131	4.10.2 设备驱动层	155
4.2.2 Linux 进程控制块	132	4.10.3 设备无关的图形引擎层	155
4.2.3 Linux 内核同步机制	133	4.10.4 API(基于 Win32)	156
4.2.4 Linux 进程间的通信	134	4.10.5 消息传递机制	156
4.2.5 Linux 中断与定时服务	137	4.10.6 窗口操作	156
4.3 Linux 的调度机制	137	4.10.7 客户区域和绝对坐标	157
4.3.1 Linux 的一般调度机制	138	4.10.8 设备上下文	157
4.3.2 RT-Linux 的实时调度机制	138	4.11 Microwindows 的移植和中文化	158
4.4 Linux 的文件系统管理	139	4.11.1 针对 uClinux 所作的修改	158
4.4.1 文件系统管理	140	4.11.2 中文化处理	158
4.4.2 虚拟文件系统	140	4.12 应用实例——电子文本阅读器	159
4.5 Linux 下的用户界面	141	4.12.1 功能要求以及方案选择	159
4.5.1 X Window 简介	141	4.12.2 具体实现	159
4.5.2 X 服务器	142	4.13 主流嵌入式 Linux	160
4.5.3 窗口管理器	143	4.13.1 RT-Linux	160
4.6 嵌入式 Linux 内核	144	4.13.2 uClinux	161
4.6.1 uClinux 概述	144	4.13.3 Embedix	161
4.6.2 uClinux 的内存管理	145	4.13.4 XLinux	161
4.6.3 uClinux 内核结构	146	4.13.5 PocketLinux	161
4.6.4 内存保护	146	4.13.6 MidoriLinux	161
4.6.5 编程接口的改变	147	4.13.7 红旗嵌入式 Linux	162
4.6.6 uClinux 的应用程序库	147		



第5章 嵌入式系统中的 Bootloader.....163

5.1 BootLoader 概述163

5.2 Bootloader 设计分析.....163

5.2.1 启动加载模式.....164

5.2.2 下载模式164

5.3 Bootloader 的启动及初始化.....164

5.3.1 Bootloader 的 stage1.....164

5.3.2 Bootloader 的 stage2.....165

5.4 Bootloader 难点分析.....166

5.5 命令控制台168

第6章 交叉编译169

6.1 编译原理169

6.1.1 编译的一般过程.....169

6.1.2 与编译器相关的程序.....170

6.1.3 编译器的移植.....171

6.2 词法分析171

6.2.1 词法的形式化描述.....171

6.2.2 词法分析程序的设计.....174

6.3 语法分析175

6.3.1 自顶向下的语法分析.....176

6.3.2 自底向上的语法分析.....178

6.4 中间代码180

6.5 代码优化182

6.6 交叉编译技术185

6.7 GCC 交叉编译器186

6.7.1 GCC 编译流程186

6.7.2 Linux 环境下的 GCC 交叉编译器189

6.8 一个交叉编译器的生成实例.....190

6.8.1 可执行文件格式.....190

6.8.2 交叉编译器191

6.8.3 相关问题193

第7章 嵌入式 Linux 的软件开发环境194

7.1 交叉编译环境194

7.1.1 安装交叉编译环境.....194

7.1.2 添加设备驱动和内核模块.....195

7.2 可执行文件195

7.2.1 COFF 文件格式195

7.2.2 ELF 文件格式195

7.2.3 flat 文件格式.....195

7.3 调试技术.....196

7.3.1 常见调试方法196

7.3.2 内存调试.....197

7.3.3 MEMWATCH.....197

7.3.4 YAMD.....198

7.4 系统调用跟踪.....200

7.5 程序调试.....200

7.5.1 gdb.....200

7.5.2 kgdb.....201

7.5.3 Oops203

7.6 系统引导和内核启动204

7.6.1 Bootloader 程序的设计与实现.....204

7.6.2 硬件平台的通信205

7.6.3 硬件平台初始化程序205

7.6.4 硬件平台监控程序205

7.7 uClinux 移植206

7.7.1 第一阶段.....206

7.7.2 第二阶段.....208

第8章 设备驱动程序210

8.1 概述.....210

8.1.1 设备类型.....211

8.1.2 设备号.....211

8.1.3 模块化编程212

8.2 设备文件接口213

8.2.1 用户访问接口213

8.2.2 open 入口点213

8.2.3 close 入口点.....213

8.2.4 read 入口点214

8.2.5 write 入口点.....214

8.2.6 ioctl 入口点.....214

8.3 文件操作.....214

8.3.1 file_operations 结构214

8.3.2 file 结构.....216

8.4 I/O 操作217

8.4.1 阻塞型 I/O.....217

8.4.2	select	218
8.4.3	异步触发	218
8.5	中断处理	219
8.5.1	注册中断处理程序	219
8.5.2	返回值	220
8.5.3	参数	220
8.5.4	实现中断处理程序	221
8.6	应用实例	222
8.6.1	字符型设备	223
8.6.2	按键	223

8.6.3	触摸屏	224
8.7	网络设备	232
8.7.1	网络驱动的框架	233
8.7.2	网卡驱动程序的加载方法	235
8.7.3	CS8900A 芯片特点	236
8.7.4	CS8900A 芯片驱动程序 的实现	237
8.7.5	网络设备驱动程序的编译	243
8.7.6	网络驱动程序的测试	243

第二部分

第 9 章 GX-ARM9-2410EP 说明 245

9.1	系统概述	245
9.2	电路说明	248
9.2.1	系统供电	249
9.2.2	系统配置	249
9.2.3	复位逻辑	250
9.2.4	启动分区(BOOT ROM BANK0)	250
9.2.5	LCD 和触摸屏接口	251
9.2.6	键盘和 SPI 接口	252
9.2.7	A/D、D/A 转换接口	253
9.2.8	Smart Media Card(NAND Flash Memory)卡电路	254
9.2.9	PCMCIA 接口	255
9.2.10	SD 卡主机(MMC)接口	255
9.2.11	IIC 接口	256
9.2.12	USB 接口	256
9.2.13	UART 接口和 irDA 接口	257

第 10 章 开发环境的建立 258

10.1	建立主机开发环境	258
------	----------	-----

10.2	配置 minicom	259
10.3	配置 TFTP	260
10.4	配置 NFS 服务	261

第 11 章 基本实验部分 262

实验 1	编译 Bootloader	262
实验 2	编译 Linux 内核	266
实验 3	制作文件系统及程序的烧写	283
实验 4	烧写各部分到目标板	286
实验 5	交叉编译并运行简单程序	291

第 12 章 高级实验部分 297

实验 6	简单驱动程序的编写	297
实验 7	LED 点阵驱动	302
实验 8	中断试验	306
实验 9	IIC LED 实验	320
实验 10	IIS 音频实验	330
实验 11	RS-485 实验	355

第一部分

绪论

近些年来，随着以计算机技术、通信技术为主的信息技术的快速发展和 Internet 的广泛应用，嵌入式系统也受到了越来越多的重视。

1. 嵌入式系统的定义

嵌入式系统是指用于执行独立功能的专用计算机系统。它由微处理器、定时器、微控制器、存储器、传感器等一系列微电子芯片及器件，和嵌入在存储器中的微型操作系统及控制应用软件组成，共同实现诸如实时控制、监视、管理、移动计算、数据处理等各种自动化处理任务。嵌入式系统以应用为中心，以微电子技术、控制技术、计算机技术和通信技术为基础，强调硬件和软件的协同性及整合性，软件及硬件可剪裁，以满足系统对功能、成本、体积和功耗等的要求。

最简单的嵌入式系统仅有执行单一功能的控制能力，在惟一的 ROM 中仅有实现单一功能的控制程序，无微型操作系统。复杂的嵌入式系统，例如个人数字助理(PDA)、手持电脑(HPC)等，几乎具有与 PC 一样的功能。实质上与 PC 的区别仅仅是将微型操作系统与应用软件嵌入在 ROM 或 Flash 存储器中，而不是存储于磁盘等载体中。很多复杂的嵌入式系统又是由若干个小型嵌入式系统组成的。

2. 嵌入式系统的硬件和软件特征

嵌入式系统的硬件必须根据具体的应用任务，以功耗、成本、体积、可靠性和处理能力等为指标来选择。嵌入式系统的核心是系统软件和应用软件，由于存储空间有限，因而要求软件代码紧凑、可靠，大多对实时性有严格要求。早期的嵌入式系统的设计方法，通常是采用“硬件优先”原则。即在粗略估计软件任务需求的情况下，首先进行硬件设计与实现。然后，在此硬件平台之上，再进行软件设计。因而很难充分利用硬件和软件资源，取得最佳性能效果。同时，一旦在测试时发现问题，需要对设计进行修改时，整个设计流程将重新进行，对成本和设计周期的影响很大。这种传统的设计方法只能改善硬件和软件各自的性能，在有限的设计空间不可能对系统做出较好的性能综合优化，在很大程度上依赖于设计者的经验和反复实验。

20 世纪 90 年代以来，随着电子系统功能的日益强大和微型化，系统设计所涉及的问题越来越多，难度也越来越大。同时硬件和软件也不再是截然分开的两个概念，而是紧密结合、相互影响的。因而出现了软硬件协同设计(Codesign)方法，即使用统一的方法和工具对软件和硬件进行描述、综合和验证。在系统目标要求的指导下，通过综合分析系统软硬件功能及现有资源，协同设计软硬件体系结构，以最大限度地挖掘系统的软硬件能力，避免由于独立设计软硬件体系结构而带来的种种弊病，得到高性能低代价的优化设计方案。



3. 嵌入式操作系统

目前流行的嵌入式操作系统可以分为两类：一类是从运行在个人电脑上的操作系统向下移植到嵌入式系统中，形成的嵌入式操作系统，如微软公司的 Windows CE 及其新版本，Sun 公司的 Java 操作系统，朗讯科技公司的 Inferno，嵌入式 Linux 等。这类系统经过个人电脑或高性能计算机等产品的长期运行考验，技术日趋成熟，其相关的标准和软件开发方式已被用户普遍接受，同时积累了丰富的开发工具和应用软件资源。另一类是实时操作系统，如 Wind River 公司的 VxWorks，ISI 公司的 pSOS，QNX 系统软件公司的 QNX，ATI 公司的 Nucleus，中国科学院凯思集团的 Hopen 嵌入式操作系统等，这类产品在操作系统的结构和实现上都针对所面向的应用领域，对实时性和高可靠性等进行了精巧的设计，而且提供了独立而完备的系统开发和测试工具，较多地应用在军用产品和工业控制等领域中。

Linux 是 20 世纪 90 年代以后逐渐成熟的一个开放源代码的操作系统。PC 机上的 Linux 版本在全球数以百万计爱好者的合力开发下，得到了非常迅速的发展。20 世纪 90 年代末，uClinux、RT-Linux 等相继推出，在嵌入式领域得到了广泛的关注，它们拥有大批的程序员和现成的应用程序，是我们研究开发工作的宝贵资源。

4. 学习嵌入式系统的意义

从控制意义上说，嵌入式系统涉及系统最底层的、芯片级的信息处理与控制。在某种意义上，对这些“微观”世界的了解与驾驭正是控制的真正目的。嵌入式系统与通常意义上的控制系统在设计思路和总体架构方面有许多不同之处，而这些不同之处恰恰是传统控制学科教学中较少教给学生的。在当今信息化社会中，嵌入式系统在人们的日常工作和生活中所占的份额，可能已超过传统意义的控制系统，这就是为什么我们的学生感到学的没有用，而有用的又没有学的原因。在嵌入式系统及开发环境方面，目前仍有许多问题尚在研究发展中，例如，嵌入式系统的软硬件协同设计方法；面向多目标、多任务的微内核嵌入式操作系统；分布嵌入式系统的实时性问题，分布式计算，分布式信息交互与综合处理；以及嵌入式系统的多目标交叉编译和交叉调试工具的研究等。

“嵌入式系统”作为自动化学科中一门理论与实际密切结合的、知识与技术含量较高的综合性专业课程，必将随着信息产业的发展而逐渐趋于成熟。

第 1 章 嵌入式系统的硬件构成

嵌入式系统以应用为中心，以计算机技术为基础，软硬件可裁剪，是对功能、可靠性、成本、体积、功耗等有严格要求的专用计算机系统。嵌入式系统的硬件一般包括处理器、存储器、外设器件和电源等。

1.1 嵌入式处理器

嵌入式系统的核心部件是各种类型的嵌入式处理器，据不完全统计，到 2006 年全世界嵌入式处理器的品种总量已经超过 1200 多种，流行体系结构有 30 多个系列，其中 8051 体系的占有多半。生产 8051 单片机的半导体厂家有 20 多个，共 350 多种衍生产品，仅 Philips 公司就有近 100 种。现在几乎每个半导体制造商都生产嵌入式处理器，越来越多的公司组建了自己的处理器设计部门。嵌入式处理器的寻址空间一般介于 64KB~16MB 之间，处理速度介于 0.1~2000MIPS 之间，常用封装形式从 8 个引脚到 144 个引脚不等。

1.1.1 嵌入式处理器的品种划分

根据目前现状，嵌入式处理器可以分成下面几类。

1. 嵌入式微处理器

嵌入式微处理器(Embedded Microprocessor Unit, EMPU)的基础是通用计算机中的 CPU。在应用中，将微处理器装配在专门设计的电路板上，只保留与嵌入式应用有关的功能，这样可以大幅度减小系统的体积和功耗。为了满足嵌入式应用的特殊要求，嵌入式微处理器虽然在功能上同标准微处理器基本一样，但在工作温度、抗电磁干扰、可靠性等方面一般都做了增强。与工业控制计算机相比，嵌入式微处理器具有体积小、重量轻、成本低和可靠性高的优点，但是在电路板上必须包括 ROM、RAM、总线接口和各种外设等器件。嵌入式微处理器及其存储器、总线和外设等安装在一块电路板上，称为单板计算机，如 STD-BUS、PC104 等。近年来，德国、日本的一些公司又开发出了类似“火柴盒”的或名片大小的嵌入式计算机系列 OEM 产品，台湾省研华公司也推出了类似的模组化系统 SOM(System On Module)。

嵌入式处理器目前主要有 Am186/88、386EX、SC-400、Power PC、68000、MIPS 和 ARM 系列等。

嵌入式微处理器根据指令体系的不同，又可分为 CISC 和 RISC 两类。大家熟悉的大多数台式 PC 机都是使用 CISC 微处理器，如 Intel 的 x86。RISC 结构体系有两大主流：Silicon Graphics 公司(硅谷图形公司)的 MIPS 技术，ARM 公司的 Advanced RISC Machines 技术。此外 Hitachi(日立公司)也有自己的一套 RISC 技术 SuperH。

RISC 和 CISC 是目前设计制造微处理器的两种典型技术，虽然它们都试图在体系结构、操作运行、软件硬件、编译时间和运行时间等诸多因素中做出某种平衡，以求达到高效的目的，但采用的方法不同，因此，在很多方面差异较大，主要表现在以下几个方面。



(1) 指令系统

RISC 设计者把主要精力放在那些经常使用的指令上, 尽量使它们具有简单高效的特色。对不常用的功能, 则通过组合指令来完成。因此, 在 RISC 机器上实现特殊功能时, 效率可能较低。但可以利用流水技术和超标量技术加以改进和弥补。而 CISC 计算机的指令系统比较丰富, 有专用指令来完成特定的功能。因此, 处理特殊任务时的效率较高。

(2) 存储器操作

RISC 对存储器操作有限制, 使控制简单化; 而 CISC 机器的存储器操作指令多, 操作直接。

(3) 程序

RISC 汇编语言程序一般需要较大的内存空间, 实现特殊功能的程序复杂, 不易设计; 而 CISC 汇编语言程序编程相对简单, 科学计算及复杂操作的程序设计相对容易, 效率较高。

(4) 中断

RISC 机器在一条指令执行的适当地方可以响应中断; 而 CISC 机器是在一条指令执行结束后才响应中断。

(5) CPU

RISC CPU 包含较少的单元电路, 因而面积小、功耗低; 而 CISC CPU 包含丰富的电路单元, 因而功能强、面积大、功耗大。

(6) 设计周期

RISC 微处理器结构简单, 布局紧凑, 设计周期短, 且易于采用最新技术; CISC 微处理器结构复杂, 设计周期长。

(7) 用户使用

RISC 微处理器结构简单, 指令规整, 性能容易把握, 易学易用; CISC 微处理器结构复杂, 功能强大, 容易实现特殊功能。

(8) 应用范围

由于 RISC 指令系统的确定与特定的应用领域有关, 故 RISC 机器更适合于专用机; 而 CISC 机器则更适合于通用机。

2. 嵌入式微控制器

嵌入式微控制器(Microcontroller Unit, MCU)又称单片机, 顾名思义, 就是将整个计算机系统集成到一块芯片中。嵌入式微控制器一般以某一种微处理器内核为核心, 芯片内部集成 ROM/EPROM、RAM、Flash、总线、总线逻辑、定时/计数器、WatchDog、I/O、串行口、脉宽调制输出、A/D、D/A 等各种必要的功能模块。为适应不同的应用需求, 一般一个系列的单片机具有多种衍生产品, 每种衍生产品的处理器内核都是一样的, 不同之处在于存储器和外设的配置及封装。这样可以使单片机最大限度地同应用需求相匹配, 从而减少功耗和成本。

与嵌入式微处理器相比, 微控制器的最大特点是单片化, 体积大大减小, 从而使功耗和成本下降、可靠性提高。微控制器是目前嵌入式系统工业的主流。微控制器的片上资源一般比较丰富, 适合于控制, 因此称为微控制器。

嵌入式微控制器目前的品种和数量最多, 比较有代表性的通用系列包括 8051、P51XA、MCS-251、MCS-96/196/296、C166/167、MC68HC05/11/12/16 和 68300 等。另外还有许多半

通用系列,如支持 USB 接口的 MCU 8XC930/931、C540 和 C541。目前 MCU 占嵌入式系统约 70% 的市场份额。

特别值得注意的是,近年来提供 x86 微处理器的著名厂商 AMD 公司将 AM186CC /CH/CU 等嵌入式处理器称之为 Microcontroller, Motorola 公司把以 Power PC 为基础的 PPC505 和 PPC555 亦列入单片机行列。TI 公司亦将其 TMS320C2XXX 系列 DSP 做为 MCU 进行推广。

3. 嵌入式 DSP 处理器

DSP 处理器对系统结构和指令进行了特殊设计,使其适合于执行 DSP 算法,编译效率较高,指令执行速度也较高。在数字滤波、FFT 和谱分析等方面,DSP 算法正在大量进入嵌入式领域,DSP 应用正从在通用单片机中以普通指令实现 DSP 功能过渡到采用嵌入式 DSP 处理器。嵌入式 DSP 处理器有两个发展来源,一是 DSP 处理器经过单片化、EMC 改造和增加片上外设成为嵌入式 DSP 处理器, TI 的 TMS320C2000/C5000 等属于此范畴;二是在通用单片机或片上系统(SOC)中增加 DSP 协处理器,例如 Intel 的 MCS-296。

推动嵌入式 DSP 处理器(Embedded Digital Signal Processor, EDSP)发展的一个重要因素是嵌入式系统的智能化,例如各种带有智能逻辑的消费类产品,生物信息识别终端,带有加解密算法的键盘,ADSL 接入,实时语音压解系统,虚拟现实显示等。这类智能化算法一般都是运算量较大,特别是向量运算、指针线性寻址等较多,而这些正是 DSP 处理器的长处所在。

嵌入式 DSP 处理器比较有代表性的产品是 Texas Instruments 的 TMS320 系列和 Motorola 的 DSP56000 系列。TMS320 系列处理器包括用于控制的 C2000 系列,移动通信的 C5000 系列,以及性能更高的 C6000 和 C8000 系列。DSP56000 目前已经发展成为 DSP56000、DSP56100、DSP56200 和 DSP56300 等几个不同系列的处理器。

DSP 的设计者们把重点放在了处理连续的数据流上。在嵌入式应用中,如果强调对连续的数据流的处理及高精度复杂运算,则应该选用 DSP 器件。

4. 嵌入式片上系统

随着 VLSI 设计的普及化及半导体工艺的迅速发展,可以在一块硅片上实现一个更为复杂的系统,这就是 System On Chip(SOC)。各种通用处理器内核将作为 SOC 设计公司的标准库,和许多其他嵌入式系统外设一样,成为 VLSI 设计中一种标准的器件,用标准的 VHDL 等语言描述,存储在器件库中。用户只需定义出整个应用系统,仿真通过后就可以将设计图交给半导体工厂制作样品。这样除个别无法集成的器件以外,整个嵌入式系统大部分均可集成到一块或几块芯片中去,应用系统电路板将变得很简洁,对于减小体积和功耗、提高可靠性非常有利。

SOC 可以分为通用和专用两类。通用系列包括 Motorola 的 M-Core,某些 ARM 系列器件,Echelon 和 Motorola 联合研制的 Neuron 芯片等。专用 SOC 一般专用于某个或某类系统中,不为一般用户所知。一个有代表性的产品是 Philips 的 SmartXA,它将 XA 单片机内核和支持超过 2048 位复杂 RSA 算法的 CCU 单元制作在一块硅片上,形成一个可加载 Java 或 C 语言的专用的 SOC,可用于公众互联网(如 Internet 安全)方面。



1.1.2 嵌入式处理器的选择

针对各种嵌入式设备的需求，各个半导体芯片厂商都投入了很大的精力研发和生产适用于这些设备的 CPU 及协处理器芯片。用于嵌入式设备的处理器必须高度紧凑、低功耗和低成本。针对每一类应用来说，开发者对处理器的选择都是多种多样的，掌上电脑就是一例，如表 1.1 所示。与全球 PC 市场不同的是，没有一种微处理器和微处理器公司可以主导嵌入式系统，仅以 32 位的 CPU 而言，就有 100 种以上的嵌入式微处理器。由于嵌入式系统设计的差异性极大，因此选择是多样化的。设计者在选择处理器时要考虑的主要因素有如下几个方面。

表 1.1 部分掌上电脑处理器一览

厂家/型号	处理器	速度
卡西欧 Cassiopeia E-100 系列	MIPS-based NEC VR4121	131MHz
康柏 Aero 2100 系列	MIPS-based NEC VR4111	70MHz
飞利浦 Nino 500 系列	MIPS-based Toshiba PR31700	75MHz
惠普 Jornada 400 系列	Hitachi SH-3 7709a	100MHz/133MHz
3Com PalmPilotTM 系列	Motorola DragonBall 68VZ328	33MHz
苹果 MessagePad 2000/2100 Intel	Intel StrongARM SA-110	160MHz
康柏 iPAQ H3650Intel	Intel StrongARM SA-1110	206MHz

(1) 调查市场上已有的 CPU 供应商

有些公司(如 Motorola、Intel、AMD)很有名气，而有一些小的公司(如 QED)虽然名气很小，但也能生产很优秀的微处理器。另外，有一些公司，如 ARM、MIPS 等，只设计但并不生产 CPU，他们把生产权授予世界各地的半导体制造商。ARM 是另外一种近年来在嵌入式系统中有影响力的微处理器制造商，ARM 的设计非常适合于小的电源供电系统。Apple 在 Newton 手持计算机中使用 ARM，另外有几款数字无线电话也在使用 ARM。

(2) 处理器的处理速度

一个处理器的性能取决于多个方面的因素：时钟频率，内部寄存器的大小，指令是否对等处理所有的寄存器等。对于许多需用处理器的嵌入式系统设计来说，目标不是在于挑选速度最快的处理器，而是在于选取能够完成作业的处理器和 I/O 子系统。如果你的设计是面向高性能的应用，那么建议你考虑某些新的处理器，其价格极为低廉，如 IBM 和 Motorola 的 Power PC。以前 Intel 的 i960 是销售极好的 RISC 高性能芯片，但是最近几年却遇到强劲的对手，让位于 MIPS、SH 以及后起之秀 ARM。

(3) 技术指标

当前，许多嵌入式处理器都集成了外围设备的功能，从而减少了芯片的数量，进而降低了整个系统的开发费用。开发人员首先考虑的是，系统所要求的一些硬件能否无需过多的胶合逻辑(Glue Logic)就可以连接到处理器上。其次是考虑该处理器的一些支持芯片，如 DMA 控制器、内存管理器、中断控制器、串行设备、时钟等的配套。

(4) 处理器的低功耗

嵌入式微处理器最大并且增长最快的市场是手持设备、电子记事本、PDA、手机、GPS 导航器、智能家电等消费类电子产品，这些产品中选购的微处理器典型的特点是要求高性能、低功耗。许多 CPU 生产厂家已经进入了这个领域。

(5) 处理器的软件支持

若仅有一个处理器，没有较好的软件开发工具的支持，也是不行的，因此选择合适的软件开发工具对系统的实现会起到很好的作用。

(6) 处理器是否内置调试工具

处理器如果内置调试工具可以大大地缩小调试周期，降低调试的难度。

(7) 处理器供应商是否提供评估板

许多处理器供应商可以提供评估板来验证你的理论是否正确，验证你的决策是否得当。

1.2 存 储 器

存储器的物理实质是一组或多组具备数据输入输出和数据存储功能的集成电路，用于充当设备缓存或保存固定的程序及数据。存储器根据存储信息的功能可以分为只读存储器 ROM(Read Only Memory)和随机存储器 RAM(Random Access Memory)，如表 1.2 所示。

表 1.2 常用存储器分类

	种 类	存储器单元构造	数据变更		数据保 持功率	相对位 成本
			变更方式	写入速度		
ROM	EEPROM	2 晶体管+隧道 区域	写动作(重写)	10ms	不要	10
	EPROM	1 晶体管	紫外线擦除+ 电气写入	(1~9)×100μs	不要	1.2
	Flash	1 晶体管	电擦除 +电气写入	10μs	不要	1
	OTP	1 晶体管	不可变更数据(仅可一次电气 写入)		不要	0.8
	MASK ROM	1 晶体管	不可变		不要	0.5
RAM	DRAM	1 晶体管+1 电 容器	写动作(重写)	(1~9)×10ns	(1~9)× 100μW	1
	SRAM	4 晶体管+2 负 载元件	写动作(重写)	(1~9)×10ns	(1~9)× 100μW	4
	NVRAM	SRAM 单元 + EEPROM 单元	存储器动作(片 批)(重写)	10ms	不要	1000