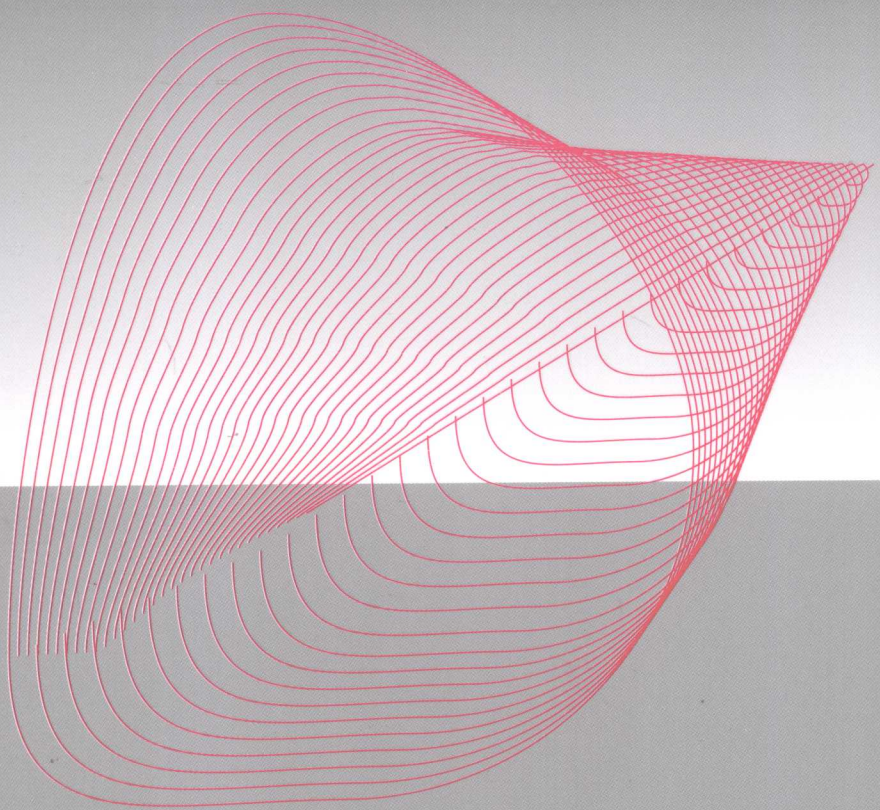
21 世纪高等学校计算机教育实用规划教材

ARM嵌入式系统结构与编程

邱铁 编著



清华大学出版社

21 世纪高等学校计算机教育实用规划教材

ARM嵌入式系统结构与编程

邱铁 编著

清华大学出版社
北京

内 容 简 介

本书是作者根据多年开发和教学实践经验并考察了当前嵌入式发展的最新动向编著而成。在内容设计上，本书采取了循序渐进的原则，对嵌入式底层硬件知识进行精心规划，以大量的实例说明技术难点，深入浅出，使嵌入式系统初学者能够以“ARM体系结构→指令系统→汇编程序设计→混合编程→硬件下编程”为主线，以阶梯式前进的方式，低起点、高效率地学习理论、深入实践，从而为嵌入式系统开发打下坚实的基础。

本书结构合理、实例丰富，具有很强的实践性和实用性，本书可作为高等学校计算机、电子信息类本科生、研究生进行嵌入式系统学习的教材或参考书，也适合嵌入式开发的工程技术人员和广大的嵌入式开发爱好者学习使用。

图书在版编目(CIP)数据

ARM嵌入式系统结构与编程/邱铁编著. —北京：清华大学出版社，2009.3
(21世纪高等学校计算机教育实用规划教材)
ISBN 978-7-302-19406-4

Ⅰ. A…　Ⅱ. 邱…　Ⅲ. 微处理器，ARM—系统设计—高等学校—教材　Ⅳ. TP332

中国版本图书馆CIP数据核字(2009)第012745号

责任编辑：梁　颖　李玮琪
责任校对：时翠兰
责任印制：王秀菊
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：北京鑫海金澳胶印有限公司
经　　销：全国新华书店
开　　本：185×260　**印　张**：24.25　**字　数**：601千字
版　　次：2009年3月第1版　　**印　　次**：2009年3月第1次印刷
印　　数：1～4000
定　　价：35.00元

前 言

嵌入式系统是软件和硬件的综合体，有人将其称为后 PC 时代和后网络时代的新秀。特别是近几年来，嵌入式产品以排山倒海之势占领了消费类电子产品市场，并开始在汽车电子、工业控制、航空航天、国防工业等领域进行全面应用。因此可以断言，面向嵌入式的信息时代已经到来。

本书作者在学生时代多次参加机器人大赛，工作后指导智能车控制大赛。最初设计机器人控制系统采用 8 位单片机，随着机器人控制功能的增强，原有 8 位芯片很难满足功能要求，另外软件设计也越来越烦琐。在 2002 年，ARM 技术产品开始大范围占领市场，基于 ARM 技术的嵌入式微控制器成为嵌入式开发的硬件支撑。世界上知名的半导体公司如 Intel、Samsung、Motorola、Philips 和 Atmel 相继推出了以 ARM 为核心的主流芯片，嵌入式开发成为信息领域研究与应用的热点。为了适应更加复杂的控制需求，作者当时选用 ARM 微控制器作为主控制单元，设计嵌入式机器人控制系统，在仅有有限的几本书可供参考的情况下，面向应用裁剪硬件，移植嵌入式操作系统，从此与嵌入式结下不解之缘。近年来，全国各大高校纷纷建立嵌入式方向，经过几年来的教学实践，已经成功地培养出一批具备嵌入式设计与开发技能的毕业生，走向嵌入式开发的各个领域。本书正是在立足于教学和实践的基础上进行编写的。

本书的编写力求将复杂问题简单化，为了说明一个问题，可能不惜篇幅，图表并用，并设有实例解析，力求使每一个嵌入式开发的初学者能快速上手，为嵌入底层开发打下坚实的基础。

本书的内容安排

- 嵌入式系统的发展历史，通过典型产品实例使读者体会嵌入式技术的研究方向和未来的发展趋势。
- ARM 处理器的内核调试结构，重点介绍了 ARM7TDMI-S、ARM9TDMI 两种结构，并分析了 ARM7 和 ARM9 的三级流水线运行机制和五级流水线运行机制。
- ARM 指令寻址方式、ARM 指令系统详细解析和 Thumb 指令系统详细解析。
- ARM 汇编语言伪指令，ARM 汇编语言程序设计中所用的伪操作，汇编语言程序设计规范，并用大量的实例说明汇编语言程序设计方法。
- 嵌入式 C 语言的编程规范，嵌入式开发中常用的位运算与控制位域以及在嵌入式 C 程序设计中要注意的问题，ARM 汇编语言与嵌入式 C 语言进行相互调用标准（AAPCS），并用大量的实例说明了相互调用应注意的问题。
- 三星公司两款流行的 ARM 处理器芯片：S3C44B0 是基于 ARM7TDMI 架构的，S3C2410 是基于 ARM920T 架构的。详细介绍了基于这两款微控制器的存储系统、

通用 IO、中断控制器、UART、I^2C 和 LCD 接口原理与应用开发。

致谢

本书编写过程中，研究生芦东泽、于玉龙翻译了相关的外文资料并做了很多开发与调试源程序的工作，感谢他们辛勤的工作；刘晓艳、巫黄旭、刘大伟、单世磊等协助校订和编辑文稿，在此一并表示感谢！

同时，关慧贞教授和郭禾教授提供了指导性意见，黄日新工程师给予了帮助，周宽久、吴国伟副教授和赖晓晨、侯刚也提供了一定的参考意见，向他们表示感谢。

另外，本书的编写参考和引用了国内外同行、专家、学者所撰写的大量文献以及网络技术论坛的精华资料。正是踩在巨人的肩上，才有本书的出现，感谢他们为本书所做的贡献。感谢清华大学出版社梁颖编辑为本书的出版所做的工作。

嵌入式系统发展非常迅速，新的技术成果不断更新。书中难免存在错误和不妥之处，恳请读者和同行批评指正。

邱　铁

2008 年 8 月

出版说明

随着我国高等教育规模的扩大以及产业结构调整的进一步完善，社会对高层次应用型人才的需求将更加迫切。各地高校紧密结合地方经济建设发展需要，科学运用市场调节机制，合理调整和配置教育资源，在改革和改造传统学科专业的基础上，加强工程型和应用型学科专业建设，积极设置主要面向地方支柱产业、高新技术产业、服务业的工程型和应用型学科专业，积极为地方经济建设输送各类应用型人才。各高校加大了使用信息科学等现代科学技术提升、改造传统学科专业的力度，从而实现传统学科专业向工程型和应用型学科专业的发展与转变。在发挥传统学科专业师资力量强、办学经验丰富、教学资源充裕等优势的同时，不断更新其教学内容、改革课程体系，使工程型和应用型学科专业教育与经济建设相适应。计算机课程教学在从传统学科向工程型和应用型学科转变中起着至关重要的作用，工程型和应用型学科专业中的计算机课程设置、内容体系和教学手段及方法等也具有不同于传统学科的鲜明特点。

为了配合高校工程型和应用型学科专业的建设和发展，急需出版一批内容新、体系新、方法新、手段新的高水平计算机课程教材。目前，工程型和应用型学科专业计算机课程教材的建设工作仍滞后于教学改革的实践，如现有的计算机教材中有不少内容陈旧（依然用传统专业计算机教材代替工程型和应用型学科专业教材），重理论、轻实践，不能满足新的教学计划、课程设置的需要；一些课程的教材可供选择的品种太少；一些基础课的教材虽然品种较多，但低水平重复严重；有些教材内容庞杂，书越编越厚；专业课教材、教学辅助教材及教学参考书短缺，等等，都不利于学生能力的提高和素质的培养。为此，在教育部相关教学指导委员会专家的指导和建议下，清华大学出版社组织出版本系列教材，以满足工程型和应用型学科专业计算机课程教学的需要。本系列教材在规划过程中体现了如下一些基本原则和特点。

（1）面向工程型与应用型学科专业，强调计算机在各专业中的应用。教材内容坚持基本理论适度，反映基本理论和原理的综合应用，强调实践和应用环节。

（2）反映教学需要，促进教学发展。教材规划以新的工程型和应用型专业目录为依据。教材要适应多样化的教学需要，正确把握教学内容和课程体系的改革方向，在选择教材内容和编写体系时注意体现素质教育、创新能力与实践能力的培养，为学生知识、能力、素质协调发展创造条件。

（3）实施精品战略，突出重点，保证质量。规划教材建设仍然把重点放在公共基础课和专业基础课的教材建设上；特别注意选择并安排一部分原来基础比较好的优秀教材或讲义修订再版，逐步形成精品教材；提倡并鼓励编写体现工程型和应用型专业教学内容和课程体系改革成果的教材。

(4) 主张一纲多本，合理配套。基础课和专业基础课教材要配套，同一门课程可以有多本具有不同内容特点的教材。处理好教材统一性与多样化，基本教材与辅助教材、教学参考书，文字教材与软件教材的关系，实现教材系列资源配套。

(5) 依靠专家，择优选用。在制订教材规划时要依靠各课程专家在调查研究本课程教材建设现状的基础上提出规划选题。在落实主编人选时，要引入竞争机制，通过申报、评审确定主编。书稿完成后要认真实行审稿程序，确保出书质量。

繁荣教材出版事业，提高教材质量的关键是教师。建立一支高水平的以老带新的教材编写队伍才能保证教材的编写质量和建设力度，希望有志于教材建设的教师能够加入到我们的编写队伍中来。

21 世纪高等学校计算机教育实用规划教材编委会

联系人：丁岭 dingl@tup.tsinghua.edu.cn

目 录

第 1 章 绪论 …… 1

1.1 嵌入式系统定义 …… 1
1.1.1 嵌入式系统发展历程 …… 1
1.1.2 嵌入式系统的定义与特点 …… 2
1.2 嵌入式操作系统 …… 3
1.2.1 嵌入式实时操作系统 …… 3
1.2.2 实时操作系统的典型应用 …… 5
1.3 嵌入式技术在工程领域的应用 …… 6
1.4 嵌入式技术的发展趋势 …… 8
思考与练习题 …… 9

第 2 章 ARM 技术与 ARM 体系结构 …… 10

2.1 ARM 体系结构版本与内核 …… 10
2.1.1 ARM 体系结构版本 …… 10
2.1.2 ARM 内核版本命名规则 …… 12
2.1.3 主流 ARM 处理器内核系列与应用 …… 12
2.2 ARM 内核模块 …… 15
2.3 ARM 处理器的工作模式 …… 18
2.4 内部寄存器 …… 19
2.4.1 通用寄存器及其分布 …… 19
2.4.2 程序状态寄存器 …… 20
2.5 ARM 异常处理 …… 22
2.6 存储方式与存储器映射机制 …… 25
2.7 ARM 流水线技术分析 …… 26
思考与练习题 …… 30

第 3 章 ARM 指令集寻址方式 …… 31

3.1 ARM 指令的编码格式 …… 31
3.2 数据处理指令寻址方式 …… 32
3.3 Load/Store 指令寻址 …… 35

3.3.1 地址计算方法 …… 35
3.3.2 字、无符号字节寻址 …… 36
3.3.3 半字、有符号字节寻址 …… 40
3.4 批量 Load/Store 指令寻址方式 …… 44
3.5 协处理器指令寻址方式 …… 46
思考与练习题 …… 49

第 4 章 ARM 指令集系统 …… 50

4.1 数据处理指令 …… 50
4.1.1 基本数据处理指令 …… 50
4.1.2 乘法指令 …… 57
4.1.3 杂类的数据处理指令 …… 61
4.2 ARM 分支指令 …… 61
4.3 加载/存储指令 …… 64
4.3.1 加载/存储字、无符号字节指令 …… 65
4.3.2 半字、有符号字节访问指令 …… 68
4.4 批量加载/存储指令 …… 69
4.4.1 基本批量字数据加载/存储指令 …… 70
4.4.2 用户模式下的批量字数据加载/存储指令 …… 71
4.4.3 带 PSR 操作的批量字数据加载指令 …… 72
4.5 交换指令 …… 73
4.6 程序状态寄存器 PSR 访问指令 …… 75
4.7 协处理器操作指令 …… 77
4.7.1 协处理器数据操作指令 …… 78
4.7.2 协处理器加载/存储指令 …… 78
4.7.3 ARM 寄存器与协处理器寄存器数据传输指令 …… 80
4.8 异常产生指令 …… 81
思考与练习题 …… 82

第 5 章 Thumb 指令 …… 84

5.1 Thumb 数据处理指令 …… 84
5.1.1 寄存器移位指令 …… 85
5.1.2 低位寄存器算术运算指令 …… 86
5.1.3 ALU 操作指令 …… 88
5.1.4 带高位寄存器操作的 Thumb 指令 …… 89
5.1.5 带 SP/PC 的算术运算指令 …… 90
5.2 Thumb 存储器操作指令 …… 91
5.2.1 字节、半字和字的加载/存储指令 …… 92
5.2.2 批量加载/存储指令 …… 96

5.3 Thumb 分支指令 …… 98
5.3.1 B 分支指令 …… 98
5.3.2 带链接的分支指令 …… 100
5.3.3 带状态切换的分支指令 …… 100
5.4 Thumb 软中断指令 …… 101
5.5 Thumb 指令功能码段分析 …… 102
5.5.1 Thumb 与 ARM 实现功能比较 …… 102
5.5.2 Thumb 与 ARM 性能比较 …… 103
思考与练习题 …… 103

第 6 章 ARM 汇编伪指令与伪操作 …… 105

6.1 汇编语言伪指令 …… 105
6.1.1 ARM 汇编语言伪指令 …… 105
6.1.2 Thumb 汇编语言伪指令 …… 108
6.2 ARM 汇编语言伪操作 …… 109
6.3 ARM 汇编伪操作 …… 110
6.3.1 符号定义伪操作 …… 110
6.3.2 数据定义伪操作 …… 115
6.3.3 汇编代码控制伪操作 …… 121
6.3.4 汇编信息报告控制伪操作 …… 124
6.3.5 指令集类型标识伪操作 …… 127
6.3.6 文件包含伪操作 …… 127
6.3.7 其他类型伪操作 …… 129
6.4 GNU ARM 汇编伪操作 …… 137
6.4.1 符号定义伪操作 …… 137
6.4.2 数据定义伪操作 …… 139
6.4.3 汇编与反汇编代码控制伪操作 …… 144
6.4.4 预定义控制伪操作 …… 147
思考与练习题 …… 150

第 7 章 汇编语言程序设计 …… 152

7.1 ARM 编译环境下汇编语句 …… 152
7.1.1 ARM 编译环境下汇编语句格式 …… 152
7.1.2 ARM 编译环境下汇编语句中符号规则 …… 152
7.2 GNU 环境下汇编语句与编译说明 …… 156
7.2.1 GNU 环境下 ARM 汇编语句格式 …… 156
7.2.2 GNU 环境下 ARM 汇编程序编译 …… 157
7.3 ARM 汇编语言程序设计规范 …… 158

7.4 ARM 汇编语言程序设计实例解析 …… 160
思考与练习题 …… 186

第 8 章 ARM 汇编语言与嵌入式 C 混合编程 …… 188

8.1 嵌入式 C 编程规范 …… 188
8.2 嵌入式 C 程序设计中的位运算 …… 190
8.3 嵌入式 C 程序设计中的几点说明 …… 193
8.3.1 volatile 限制符 …… 193
8.3.2 地址强制转换与多级指针 …… 194
8.3.3 预处理的使用 …… 196
8.4 嵌入式 C 程序设计格式 …… 200
8.5 过程调用标准 ATPCS 与 AAPCS …… 203
8.5.1 寄存器使用规则 …… 203
8.5.2 数据栈使用规则 …… 204
8.5.3 参数传递规则 …… 205
8.6 ARM 汇编语言与嵌入式 C 混合编程 …… 208
8.6.1 内嵌汇编 …… 208
8.6.2 ARM 汇编语言与嵌入式 C 程序相互调用 …… 211
思考与练习题 …… 214

第 9 章 S3C44B0/S3C2410 硬件结构与关键技术分析 …… 216

9.1 处理器简介 …… 216
9.2 S3C44B0/S3C2410 存储控制器 …… 219
9.2.1 S3C44B0 存储控制与地址空间 …… 219
9.2.2 S3C2410 存储控制与地址空间 …… 220
9.2.3 S3C44B0/S3C2410 存储位宽控制 …… 221
9.2.4 S3C44B0/S3C2410 存储器接口时序分析 …… 222
9.2.5 S3C44B0/S3C2410 存储控制寄存器 …… 225
9.2.6 SDRAM 接口电路设计 …… 232
9.2.7 S3C44B0 存储器初始化实例 …… 233
9.3 S3C2410 NAND Flash 控制器 …… 234
9.4 S3C44B0/S3C2410 时钟电源管理 …… 240
9.4.1 S3C44B0/S3C2410 时钟管理 …… 240
9.4.2 S3C44B0/S3C2410 电源管理 …… 243
9.4.3 S3C44B0/S3C2410 时钟与电源管理专用寄存器 …… 245
9.5 S3C44B0/S3C2410 通用 I/O 端口 …… 250
9.5.1 端口控制描述 …… 251
9.5.2 端口寄存器 …… 251
9.5.3 通用 I/O 接口设计实例 …… 276

9.6 S3C44B0/S3C2410 中断机制 …… 279
9.6.1 S3C44B0 中断控制器 …… 279
9.6.2 S3C2410 中断控制器 …… 281
9.6.3 S3C44B0/S3C2410 中断控制特殊功能寄存器 …… 284
9.6.4 S3C44B0/S3C2410 中断控制器设计实例 …… 297
思考与练习题 …… 299

第 10 章 S3C44B0/S3C2410 通信与 LCD 接口技术 …… 302

10.1 S3C44B0/S3C2410 UART …… 302
10.1.1 UART 原理 …… 302
10.1.2 S3C44B0/S3C2410 UART 模块 …… 303
10.1.3 S3C44B0/S3C2410 UART 操作 …… 305
10.1.4 UART 中断与波特率的计算 …… 306
10.1.5 S3C44B0/S3C2410 UART 专用功能寄存器 …… 308
10.1.6 S3C44B0/S3C2410 UART 设计实例 …… 313
10.2 S3C44B0/S3C2410 I^2C 总线接口 …… 315
10.2.1 I^2C 总线原理 …… 315
10.2.2 S3C44B0/S3C2410 I^2C 总线功能模块 …… 318
10.2.3 S3C44B0/S3C2410 I^2C 总线操作 …… 318
10.2.4 S3C44B0/S3C2410 I^2C 专用功能寄存器 …… 322
10.2.5 S3C44B0/S3C2410 I^2C 总线设计实例 …… 325
10.3 S3C44B0/S3C2410 LCD 控制器 …… 329
10.3.1 LCD 简介 …… 329
10.3.2 S3C44B0/S3C2410 LCD 控制器模块 …… 330
10.3.3 S3C44B0/S3C2410 LCD 控制器专用功能寄存器 …… 343
10.3.4 S3C44B0/S3C2410 LCD 控制器设计实例 …… 354
思考与练习题 …… 359

附录 A S3C44B0/S3C2410 封装与 I/O 复用信息 …… 361

附录 B 链接定位与系统引导程序 …… 368

参考文献 …… 371

第1章 绪 论

本章主要介绍嵌入式系统的发展历史和相关概念，当前嵌入式技术的主要应用以及市场上最流行的嵌入式产品，通过典型产品实例使读者了解当前嵌入式技术的应用状况和研究方向。最后介绍了嵌入式技术未来的发展趋势。

1.1 嵌入式系统定义

近年来，以集成电路为代表的微电子技术取得了重大突破，这使计算机技术、微控制器技术得到了迅速发展，再加上网络技术的应用与普及，加速了21世纪工业生产、军工国防、消费电子、商业活动、科学实验和家庭生活等领域的自动化和信息化进程，这些为嵌入式技术的大规模发展提供了强大的产业支撑。嵌入式技术正是在这些领域的产业需求下产生并一步步壮大的。

1.1.1 嵌入式系统发展历程

嵌入式系统从21世纪开始大规模发展起来，但这个概念在上世纪就已经出现。从20世纪70年代单片机的出现到目前各式各样的嵌入式微处理器，微控制器的大规模应用，嵌入式系统已经有了30多年的发展历史。

嵌入式系统的出现最初是基于单片机的。20世纪70年代单片机的出现，使得汽车、家电、工业机器人、通信装置以及成千上万种产品可以通过内嵌电子装置来获得更佳的使用性能：更容易使用、更快、更便宜。当时只是使用8位的芯片，执行一些简单的程序指令，不过这些装置已经初步具备了嵌入式的应用特点。

Intel公司于1971年开发出第一片具有4位总线结构的微处理器4004，当时主要用于电子玩具、家用电器，电子控制及简单的计算工具，可以说是嵌入式系统的萌芽阶段。1976年Intel公司推出功能相对较完备的单片机8048。Motorola同时推出了68HC05，Zilog公司推出了Z80系列。在80年代初，Intel又进一步完善了8048，在它的基础上研制成功了8051，这在单片机的历史上是值得纪念的一页。目前，51系列的单片机仍然在市场上占有很大的比例，在各种产品中有着非常广泛的应用。

在20世纪80年代早期，出现了商业级的“实时操作系统内核”，嵌入式系统开发的程序员开始在实时内核下编写嵌入式应用软件，从而使新产品的研制可以获取更短的开发周期、更低的开发资金和更高的开发效率。这个早期的操作系统实时内核包含了许多传统操作系统的特征，包括进程（或任务）管理、进程（或任务）调度与通信、中断机制、时间管理及内存管理等功能。其中比较著名的有Ready System公司的VRTX、Integrated System

Incorporation (ISI)的 PSOS 和 IMG 的 VxWorks、QNX 公司的 QNX 等。这些早期的嵌入式操作系统都具有嵌入式的典型特点：

(1) 采用抢占式的调度策略，任务的实时性好，并且执行时间是确定的；

(2) 具有可裁剪性(根据任务的需要与否进行添加或删除操作系统模块)和可移植性(移植到各种处理器上)；

(3) 具有较好的可靠性和可扩展性，适合嵌入式产品的应用开发。

在嵌入式操作系统出现以前，程序员直接在硬件平台上设计程序，这就要求程序员对硬件资源也要有所了解。这些嵌入式实时多任务操作系统的出现后，程序员可以根据任务需求和操作系统的接口定义进行编程，而硬件资料则由嵌入式操作系统来管理，因此对于任务的实现有了更高的灵活性。

进入 20 世纪 90 年代，随着任务复杂性的不断增加，软件规模也越来越大，实时核也随之逐渐发展并完善，并由此发展成为实时多任务操作系统(RTOS)，并作为一种可移植的软件平台成为当前国际嵌入式系统的应用软件支撑。更多组织和公司看到了嵌入式系统的广阔前景，开始大力发展自己的嵌入式操作系统。这一阶段在国际上相继出现了 Palm OS，WinCE，嵌入式 Linux，Nucleux 等嵌入式操作系统，为嵌入式软件应用开发铺平了道路。

步入 21 世纪以来，嵌入式系统得到了极大的发展。在硬件上，MCU 的性能得到了极大的提升，特别是 ARM 技术的出现与完善，为嵌入式操作系统提供了功能强大的硬件载体。当前几家知名半导公司如 Intel、Sansung、Motorola、Philips 和 Atmel 纷纷采用 ARM 技术，再加上其公司先进的外围接口技术与先进的制造技术，设计出功能完备的 MCU，应用到工业自动化、消费类电子、航空航天、军事工业等各个领域。至此，基于 ARM 技术的产品迅速占领了市场，将嵌入式系统推向一个崭新的阶段。

1.1.2 嵌入式系统的定义与特点

根据 IEEE(国际电机工程师协会)的定义：嵌入式系统是“用来控制或监视机器、装置或工厂等大规模系统的设备”(原文为 Devices used to control, monitor, or assist the operation of equipment, machinery or plants)。这主要是从应用上加以定义的，从中可以看出嵌入式系统是软件和硬件的综合体，还可以涵盖机械等附属装置。

国内嵌入式行业一个普遍被认同的定义是：以应用为中心、以计算机技术为基础、软件硬件可裁剪、适应应用系统对功能、可靠性、成本、体积、功耗严格要求的专用计算机系统。

从这个定义可以看出嵌入式系统是与应用紧密结合的，它具有很强的专用性，必须结合实际系统需求进行合理的裁剪利用。因此有人把嵌入式系统比作是一个针对特定的应用而“量身定做”的专用计算机系统。

IEEE 给嵌入式系统下的定义是从整体上对嵌入式系统的描述，而国内嵌入式行业所下的定义更侧重细节。从嵌入式系统的定义中，我们可以看出，与传统的计算机或基于计算机的数字化产品相比，一般的嵌入式技术产品有以下共同的特点：

(1) 嵌入式系统的底层硬件平台采用微控制器(MCU)作为主控单元，然后在此平台下移植嵌入式操作系统并进行相应的应用程序开发。这个系统所完成的功能是特定的、精简

的，相对来说功能比较单一。因此，无论从功耗、体积、重量、价格等哪一方面来讲都有一定的优势。例如，由使用嵌入式 Linux 的智能手机，由机内充电电池供电一般可以维持 3～6 天，但是一台笔记本电脑由机内充电电池供电，一般只能维持 2～4 个小时，从中我们可以看出嵌入式设备的低功耗。

(2) 嵌入式系统是微电子技术、计算机技术和特定的工程应用的综合体，是一门交叉学科。由于系统体积和内部硬件资源的限制，嵌入式系统的硬件和软件都必须协同式定制设计，要保证系统运行实现最优。为了提高代码运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(ROM 或 Flash ROM)或带有片内存储器的微控制器芯片中，一般不存储于磁盘、光盘等外部载体中。

当然，随着时代的发展，会不断赋予嵌入式系统新的内涵。因此我们对嵌入式系统定义的理解也会随着时代的不同而有所变化。例如，当前的嵌入式系统功能已经非常完善，其内部资源和处理速度远超过 90 年代的计算机。由于网络技术的发展，当前的嵌入式产品不仅支持各种通信协议，而且在互连网、浏览器、可视化等方面都取得很大的突破。

1.2 嵌入式操作系统

嵌入式操作系统(Embedded Operation System)产生于 20 世纪 80 年代，当时国际上一些 IT 公司开始进行商用嵌入式操作系统和专用操作系统的设计与开发。到目前为止，已经出现了很多嵌入式操作系统，在嵌入式产品开发中发挥着重要作用。

1.2.1 嵌入式实时操作系统

嵌入式实时操作系统是指在限定的时间内对输入进行快速处理并作出响应的嵌入式操作系统。实时操作系统具有实时性，必须有相应的硬件支持才能达到实时控制的目的。在嵌入式操作系统中，首先要保证实时性，需要调度一切可利用的硬件和软件资源来完成实时控制，其次要考虑提高计算机系统的使用效率，满足控制任务对时间的限制和要求。

实时操作系统(RTOS)在嵌入式领域应用广泛。嵌入式设备具有多样性，对于完成不同的任务，所构建的嵌入式平台一般都有所不同。所以 RTOS 必须具有很大的可剪裁性(Scalability)，以适应不同嵌入式硬件平台的需求。现在 RTOS 的研究与应用是国际上发展的热点，国内外很多的高校、组织和公司都组建了相关的研发团队。

当前，嵌入式实时操作系统大体可分为商用型和免费型(开源)两种。商用型的实时操作系统功能稳定、可靠，有完善的技术支持和售后服务，但价格一般较高。典型的商用嵌入式实时操作系统有 VxWorks，QNX，OSE，ECOS，PSOS，Windows CE 等等。美国 WindRiver 公司设计开发的嵌入式实时操作系统 VxWorks 被广泛地应用在通信、军事、航空、航天等高精尖技术领域中，例如 1997 年 4 月在火星探测器上也使用到了 VxWorks。OSE 主要是由 ENEA DataAB 下属的 ENEA OSE SystemsAB 负责开发和技术服务的，在实时操作系统以及分布式和容错性应用上取得了很大的成功。它通常应用于电信行业，比如爱立信等公司。商用嵌入式实时系统一般都是公司针对自己的特定硬件平台而定制的。免费型的实时操作系统在价格方面具有优势，主要有嵌入式 Linux 和 μC/OS-II，目前国内

外院校、组织大多数的研究一般都基于这两个操作系统。

1. 嵌入式 Linux

嵌入式 Linux 操作系统是针对嵌入式微控制器的特点而量身定做的一种 Linux 操作系统，包括常用的嵌入式通信协议和常用驱动，支持多种文件系统。使用嵌入式 Linux 进行产品开发具有以下好处：

(1) Linux 是源码开放的，每一个技术细节都是透明的，易于裁剪定制。全世界拥有众多 Linux 爱好者，当在开发中遇到问题时，可以通过网络向广大的 Linux 爱好者求助，有利于问题的快速解决。

(2) 目前嵌入式 Linux 已经在多种嵌入式处理器芯片移植成功，有大量且不断增加的开发工具，这些工具为嵌入式系统的开发提供了良好的开发环境。

(3) Linux 内核小、功能强大、运行稳定、效率高。经过众多 Linux 爱好者的不断努力与改进，Linux 系统的功能已经非常完善，现在 Linux 不仅支持网络协议，多种文件系统，而且支持很多应用软件，这对嵌入式开发者来说可以走很多捷径。

当前最为出色的 Linux 内核版本是 2.6，它在文件管理、多任务等很多方面都已非常完善，另外 Linux 正在向支持多核技术发展。Linux 是优秀的嵌入式系统开发软件平台，目前主要有 RT_Linux，μCLinux 和嵌入式 Linux，在嵌入式开发中应用广泛。

2. 嵌入式实时操作内核 μC/OS-II

μC/OS 是源代码公开的实时嵌入式系统，μC/OS-II 是 μC/OS 的升级版本。其主要特点如下：

1) 源代码公开

源代码全部公开，并且可以从有关书籍以及网络上找到详尽的源代码讲解和注释。这样系统变得透明，很容易把操作系统移植到各个不同的硬件平台上。

2) 可移植性

μC/OS-II 绝大部分源码是用 ANSIC C 写的，可移植性较强。而与微处理器硬件相关的部分是用汇编语言描述的，已经压到最低限度，使得 μC/OS-II 便于移植到其他微处理上。μC/OS-II 可以在绝大多数 8 位、16 位、32 位，甚至 64 位微处理器、微控制器、数字信号处理器(DSP)上运行。

3) 可固化

μC/OS-II 是为嵌入式应用而设计的，这就意味着，只要开发者有固化手段(C 编译、连接、下载和固化)，μC/OS-II 就可以嵌入到所开发的产品中。

4) 可裁剪

实际应用中，可以只使用 μC/OS-II 中应用程序需要的那些系统服务，也就是说有的产品可以只使用很少几个 μC/OS-II 调用，这样可以减少产品中的 μC/OS-II 所需的存储器空间(RAM 和 ROM)。这种可裁剪性是依靠条件编译实现的。

5) 占先式

μC/OS-II 完全是占先式的实时内核，这意味着 μC/OS-II 总是运行就绪条件下优先级最高的任务。

6）多任务

μC/OS-II 可以管理 64 个任务，不过目前系统保留 8 个，应用程序最多可以有 56 个任务，赋予每个任务的优先级必须是不相同的。

7）可确定性

全部的 μC/OS-II 的函数调用与服务执行时间是可知的。

8）系统服务

μC/OS-II 提供很多系统服务，例如邮箱、消息队列、信号量、块大小固定的内存的申请与释放、时间相关函数等。

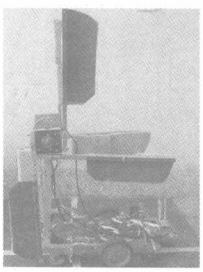
μC/OS-II 自 1992 年以来已经有很多成功的商业应用。另外，2000 年 7 月，μC/OS-II 在一个航空项目中得到了美国联邦航空管理局对商用飞机的、符合 RTCA DO-178B 标准的认证。这一结论表明，该操作系统的质量得到了认证，可以在任何应用中使用。

μC/OS-Ⅱ是一个实时操作系统内核，只包含了任务管理、任务调度、时间管理、内存管理和任务间的通信与同步等基本功能。没有提供文件系统、网络驱动及管理、图形界面等模块。但是由于 μC/OS-Ⅱ的可移植性和开源性，用户可以根据功能需求添加所需的各种服务（当然这些服务需要自己去定义）。

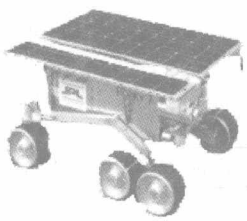
1.2.2 实时操作系统的典型应用

机器人的控制技术一直代表着信息技术、控制技术的前沿，因此我们关注一下嵌入式技术在机器人控制领域的应用情况。

目前机器人技术的发展，由于其应用的复杂性日益提高，对核心硬件的要求也越来越高，速度更快，端口更多，有的甚至采用多处理器和 DSP 进行分布式运算，而操作系统方面，除了支持底层硬件平台的复杂性外，实时性、可靠性和开放性，都成了机器人领域操作系统的必备特性。只有好的软硬件平台，才能使整个机器人系统有一个好的载体，机器人的整体性能才会提高。近年来，RTOS 在机器中的应用已经非常广泛，如图 1-1 所示，图 1-1(a)的自动运输车采用 μC/OS-Ⅱ作为实时内核，图 1-1(b)的火星探测器 rocky-7 使用 VxWorks 操作系统。



(a) 自动运输车



(b) 火星探测器rocky-7

图 1-1 带有操作系统的机器人

随着微控制器性能的不断提高，嵌入式系统的研究和应用发展，RTOS 将会在机器人上有着越来越广泛的应用。如 SONY 公司的 AIBO 系列机器人，就是一个 RTOS(Aperios)运行在 32 位的处理器硬件平台上。机器人系统既是一个典型的智能机器人系统，又为多智能