HZ BOOKS
华章教育

计 算 机 科 学 丛 书

PEARSON

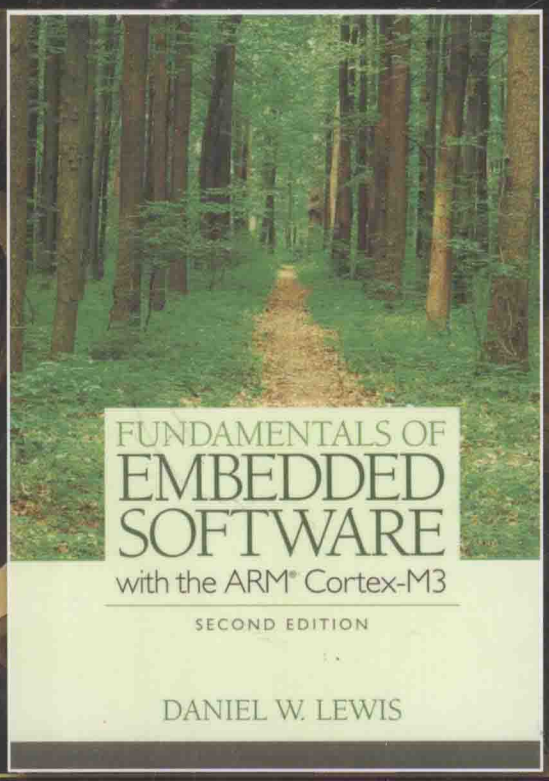
原书第2版

嵌入式软件设计基础

基于ARM Cortex-M3

（美）Daniel W. Lewis 著 陈文智 胡威 等译
圣克拉拉大学

Fundamentals of Embedded Software with the ARM Cortex-M3 Second Edition



机械工业出版社
China Machine Press

计 算 机 科 学 丛 书

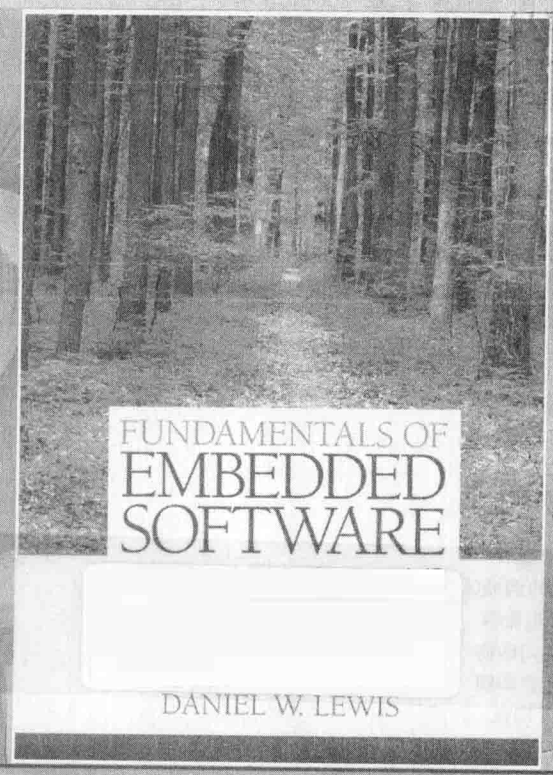
原书第2版

嵌入式软件设计基础

基于ARM Cortex-M3

(美) Daniel W. Lewis 著 陈文智 胡威 等译
圣克拉拉大学

Fundamentals of Embedded Software with the ARM Cortex-M3 Second Edition



机械工业出版社
China Machine Press

图书在版编目（CIP）数据

嵌入式软件设计基础：基于 ARM Cortex-M3（原书第 2 版）/（美）刘易斯（Lewis, D. W.）著；陈文智等译．—北京：机械工业出版社，2013.9
（计算机科学丛书）
书名原文：Fundamentals of Embedded Software with the ARM Cortex-M3, Second Edition

ISBN 978-7-111-44176-2

Ⅰ.嵌… Ⅱ.①刘… ②陈… Ⅲ.微处理器－系统设计 Ⅳ.TP332

中国版本图书馆 CIP 数据核字（2013）第 228809 号

本书以实践中最常运用的方式讲解汇编语言——实现小型、快速或特殊目的的例程，这些例程由主程序（高级语言编写，如 C）调用。通过运用嵌入式软件环境，本书介绍多线程程序设计、抢占式系统与非抢占式系统、共享资源和调度。

本书适用于高等院校工科各专业嵌入式计算机系统程序设计、C 语言程序设计及汇编语言程序设计类本科课程，也可供相关技术人员学习参考。

机械工业出版社（北京市西城区百万庄大街 22 号 邮政编码 100037）
责任编辑：姚 蕾 迟振春
藁城市京瑞印刷有限公司印刷
2014 年 1 月第 1 版第 1 次印刷
185mm × 260mm • 12.75 印张
标准书号：ISBN 978-7-111-44176-2
定 价：45.00 元

凡购本书，如有缺页、倒页、脱页，由本社发行部调换
客服热线：（010）88378991 88361066 投稿热线：（010）88379604
购书热线：（010）68326294 88379649 68995259 读者信箱：hzjsj@hzbook.com

文艺复兴以降，源远流长的科学精神和逐步形成的学术规范，使西方国家在自然科学的各个领域取得了垄断性的优势；也正是这样的传统，使美国在信息技术发展的六十多年间名家辈出、独领风骚。在商业化的进程中，美国的产业界与教育界越来越紧密地结合，计算机学科中的许多泰山北斗同时身处科研和教学的最前线，由此而产生的经典科学著作，不仅擘划了研究的范畴，还揭示了学术的源变，既遵循学术规范，又自有学者个性，其价值并不会因年月的流逝而减退。

近年，在全球信息化大潮的推动下，我国的计算机产业发展迅猛，对专业人才的需求日益迫切。这对计算机教育界和出版界都既是机遇，也是挑战；而专业教材的建设在教育战略上显得举足轻重。在我国信息技术发展时间较短的现状下，美国等发达国家在其计算机科学发展的几十年间积淀和发展的经典教材仍有许多值得借鉴之处。因此，引进一批国外优秀计算机教材将对我国计算机教育事业的发展起到积极的推动作用，也是与世界接轨、建设真正的世界一流大学的必由之路。

机械工业出版社华章公司较早意识到“出版要为教育服务”。自 1998 年开始，我们就将工作重点放在了遴选、移译国外优秀教材上。经过多年的不懈努力，我们与 Pearson，McGraw-Hill，Elsevier，MIT，John Wiley & Sons，Cengage 等世界著名出版公司建立了良好的合作关系，从他们现有的数百种教材中甄选出 Andrew S. Tanenbaum，Bjarne Stroustrup，Brain W. Kernighan，Dennis Ritchie，Jim Gray，Afred V. Aho，John E. Hopcroft，Jeffrey D. Ullman，Abraham Silberschatz，William Stallings，Donald E. Knuth，John L. Hennessy，Larry L. Peterson 等大师名家的一批经典作品，以“计算机科学丛书”为总称出版，供读者学习、研究及珍藏。大理石纹理的封面，也正体现了这套丛书的品位和格调。

“计算机科学丛书”的出版工作得到了国内外学者的鼎力襄助，国内的专家不仅提供了中肯的选题指导，还不辞劳苦地担任了翻译和审校的工作；而原书的作者也相当关注其作品在中国的传播，有的还专程为其书的中译本作序。迄今，“计算机科学丛书”已经出版了近两百个品种，这些书籍在读者中树立了良好的口碑，并被许多高校采用为正式教材和参考书籍。其影印版“经典原版书库”作为姊妹篇也被越来越多实施双语教学的学校所采用。

权威的作者、经典的教材、一流的译者、严格的审校、精细的编辑，这些因素使我们的图书有了质量的保证。随着计算机科学与技术专业学科建设的不断完善和教材改革的逐渐深化，教育界对国外计算机教材的需求和应用都将步入一个新的阶段，我们的目标是尽善尽美，而反馈的意见正是我们达到这一终极目标的重要帮助。华章公司欢迎老师和读者对我们的工作提出建议或给予指正，我们的联系方法如下：

华章网站：www.hzbook.com

电子邮件：hzjsj@hzbook.com

联系电话：（010）88379604

联系地址：北京市西城区百万庄南街 1 号

邮政编码：100037

华章教育
华章科技图书出版中心

译者序

Fundamentals of Embedded Software with the ARM Cortex-M3, 2E

嵌入式系统是面向应用而进行定制的计算系统，它将多领域技术与具体行业需求结合在一起。经过多年的发展，其应用领域已经非常广泛，包括交通、能源、工业控制等，目前在汽车、家电和移动终端等领域的嵌入式创新应用不断涌现。

在嵌入式系统发展迅速、应用范围不断扩大的背景下，我们迫切需要从嵌入式硬件与软件之间的关系出发，建立嵌入式软件设计与开发的系统性框架。而作为国内较早开设嵌入式系统课程的高校之一，浙江大学在教学过程中发现，嵌入式系统软硬件结合相对紧密，如果要学习设计和开发嵌入式软件，就必须对嵌入式硬件有一定程度的了解。因此，需要在嵌入式软件开发的课程中将具体的硬件知识与嵌入式软件知识结合起来，从而让学生能够真正深入了解嵌入式软件开发的本质。

在《Fundamentals of Embedded Software with the ARM Cortex-M3》一书的第 2 版出版后，我们应机械工业出版社的邀请，开始进行翻译工作。在阅读完本书后，发现书中内容正是学习和了解嵌入式软件开发所需要的。本书作者选择了典型的嵌入式处理器体系结构作为嵌入式软件开发的基础，以嵌入式软件开发为目标，将软硬件系统很好地融合起来，通过循序渐进的方式，将嵌入式软件开发的本质生动地展现在读者面前。

作者首先以实际生活中可能遇到的嵌入式系统来引入“嵌入式系统”的概念，并针对嵌入式系统做初步的介绍，以嵌入式系统中数的表示方法、算术运算的实现和 C 语言的基本特征等作为进一步探讨的基础；接下来分别介绍了嵌入式计算系统的构成、数据操作、控制结构和 I/O 程序设计，逐步深入到嵌入式软件设计与开发的主要内容中去；然后讨论了并发、调度、存储管理和共享存储，对嵌入式系统软件的核心进行剖析；最后探讨了嵌入式系统的软件初始化过程。同时，书中也提供了丰富的习题，可以让学生及时对所学知识进行全面把握和巩固。

本书由浙江大学的陈文智和施青松以及武汉科技大学的胡威共同翻译完成。胡威和施青松对翻译初稿进行了审阅和修订，最终由陈文智统稿、定稿。

在进行本书的翻译时，我们力求准确无误地表达出原文的意思，尽可能使文字流畅，易于理解。但是受译者水平和时间所限，不免有疏漏和不足之处，恳请读者不吝指正。

在此非常真诚地感谢机械工业出版社的王春华和其他编辑，是她们细致的工作使得本书的翻译和出版顺利完成。

译者

2013 年 8 月 12 日于求是园

我现在所坦诚的事情，是在很多年前就应该告诉大家的，不过我现在仍然对本科的学习记忆犹新。就像我自己的学生一样，由于老师正在为课程编写教材，我也曾不得不在某一门课程中通过幻灯片和讲义获得大量学习材料。因此，我要感谢所有那些不得不忍受与我同样经历的学生，那些在不知不觉中帮助我组织和理清这些素材，并巧妙指出我的错误，以及曾激发了我的灵感的学生。谨以此书献给你们和那些追随你们脚步的人们。

如果问“你家中有多少计算机”，多数人可能都会回答说有 2 台或 3 台。如果问“你家中有多少微处理器（microprocessor）”，回答之前得认真地想一想。提示：远比两三个要多得多！实际上甚至超过 10 个、20 个！今天微处理器几乎被嵌入到每一种你能想到的或者你想不到的电子产品当中。它们已经再普遍不过了，不仅是我们的家里，我们的工作场所、汽车、飞机、交通灯、超市、移动电话等中都有它们的身影，总之，几乎存在于我们生活的方方面面。

嵌入式系统为学生提供了一个令他们兴奋的机会来表现他们的创新能力。学生难道没有梦想设计下一个新的小工具，能够发挥他们的想象力吗？作为教育者，我们的挑战是利用这种兴奋，释放这些年轻人的能量，激发他们掌握这个主题的兴趣。

本版中的更新内容

实际上这是一个几乎完全重写的版本，具有以下几个显著的变化：

1）取代了之前介绍的 Intel IA32 处理器，而涵盖了 ARM Cortex-M3 v7 处理器的体系结构和指令集。之所以选择 32 位的 ARM 处理器，是因为：2004 年到 2010 年间设计的 75% 的嵌入式系统采用了 32 位处理器[⊖]；ARM 处理器的使用率快速增长，从 2007 年占所有嵌入式应用的 19% 增长到 2010 年的超过 35%[⊜]；ARM Cortex-M3 是为嵌入式实时应用专门设计的。

2）涵盖了 ARM Cortex-M3 处理器能够很好地适应于嵌入式应用的主要特征，包括能够避免排空流水线的条件执行，中断“尾链”，中断的“迟到处理”以及用于内存和 I/O 中单个位寻址的“位段”。

3）涵盖了 ARM 的过程调用标准以及针对函数参数、返回值和临时变量的寄存器的正确选择。

4）增加了新的章“实现算术运算”（第 3 章），对汇编语言编程和 I/O 进行了重新组织和扩展，从原来的三章增加为四章。

5）扩展了对二进制加法和减法的介绍，为以下知识点增加了新的内容：使用移位、加法和减法的方法作为常量乘法的替代方法，使用倒数乘法作为常量除法的替代方法，以及定点实数乘法的完整示例。

6）更新了 C 语言有关 C99 标准的内容。

7）大大扩充了第 1 ～ 12 章中每章结尾处的习题，并提供了部分习题的答案。

8）教学辅助网站上包括 14 个新的实验程序作业，这些作业使用德州仪器的 EKI-LM3S811 评估板（包括 LM3S811 评估板）、Stellaris 外设驱动程序库和 IAR 嵌入式测试程序集成开发环境的启动版本（Kickstart Edition）。LM3S811 评估板提供了可编程定时器、指

⊖ 来源：Jerry Krasner, Ph.D., Dolores A. Krasner, “Embedded Market Forecasters: 2010 Embedded Hardware/Software Design Preferences.”

⊜ 来源：VDC Research Group, “Embedded Software & Tools 2010 Market Intelligence Service, Track 2: Embedded System Engineering Survey Data, Volume 1: Operating Systems.”

轮驱动分压计、A/D 转换器、图形显示器、用户和复位按钮、用户 LED、并行端口、串行 UART 端口、USB 端口和 JTAG 连接器。

9）使用开源的 FreeRTOS 实时内核来开发多线程实验作业，以探索调度、处理器利用率、线程饥饿、非绑定的优先级反转、死锁、互斥、信号量和队列。

10）教学辅助网站上提供了完整的方案来组装简易的低端开发板，这个开发板用于最后的四个实验作业，从而开发基于反馈控制的小型直流电机的多线程程序。

目标

本书的最终目标是为理解嵌入式软件的多线程风格编程和高可靠性需求奠定良好的基础。为此，确定如下的具体目标：

1）理解在机器级别数据如何表示，并领会不同表示方法的效果及其限制。

2）掌握在嵌入式系统中最常用的语言特定的功能，比如位操作和变量访问。

3）学习程序员视角的处理器体系结构，以及为何有时在汇编层面进行编程是必要的或适当的。

4）学习 I/O 编程的不同风格，最终了解如何利用事件驱动的方法将数据处理分离成多个相互独立的计算线程。

5）学习抢占和非抢占的多线程编程、共享资源和临界区，以及调度是怎样管理系统响应时间的。

6）通过重新学习作用域（scope）、参数传递（parameter passing）、递归（recursion）和内存分配（memory allocation）来强化基本的编程技能。

7）学习与共享内存对象相关的问题，共享内存是如何受内存分配所影响的，什么样的编程实践能够用来减少共享内存的冲突。

目标读者

本书致力于为计算机科学、计算机工程或电子工程大学二年级课程打下基础。期望这样一门课程能够替代传统的计算机组成和汇编程序设计课程。

本书将描述实践中最常用的汇编形式——实现小而快或是特定目的的例程，以在类似 C 语言这样的高级语言编写的主程序中调用。本书因此将从“需要了解”的角度涵盖处理器组成和汇编语言的内容，而不是将其作为主要的目标。这种组织方式能帮助读者在学习本书和相关课程的过程中掌握嵌入式软件方面的汇编语言。因此，学生不仅需要学习仍然起到非常重要作用的汇编知识，也需要学习多线程编程、抢占和非抢占系统、共享资源和调度知识，这些知识有助于激发他们的兴趣，满足他们的好奇心并为学习后续的操作系统、实时系统、网络和基于微处理器的设计等课程打下基础。

在大多数学校，入门编程课程（CS1 和 CS2）已不再用 C 或者 Pascal 这样的过程编程语言教授，流行的方法是使用面向对象程序设计语言如 C++ 或者 Java 进行教学。尽管有这样一些变化，仍然不难发现在一门或多门高年级课程中仍然普遍使用过程化语言，并且在工业领域同样发现仍在使用这样的语言。在笔者的学校，我们通过围绕本书中的内容来重新组织传统的汇编语言课程以解决这个悖论，这种重新设计不仅能为在已经非常紧凑的课程中涵盖过程性方法并介绍嵌入式系统的流行主题预留出足够的时间，还能增强学生对 CS1 和 CS2 课程中引入的参数传递、作用域和内存分配模式等内容的理解。

本书假定学生已经知道如何使用C、C++或者Java编程，并且了解这些语言在底层语法上的相似性，这种相似性使他们能够相对容易地从C++或Java程序迁移到C。本书并不着重于介绍C语言中那些让人难以忍受的细节，而是强调C语言在嵌入式应用中最常用的特征，并通过例子和编程作业来介绍过程化风格，采用的例子和编程作业中包括大量预先写好的源代码。原则上唯一需要的绝对前置课程是介绍C、C++或Java的CS1课程。然而本书也强烈推荐通过讲解数据结构知识的CS2课程来提高编程熟练程度。

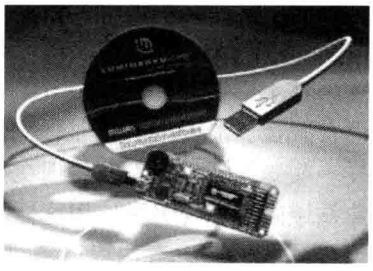
编程作业与教学辅助网站

本书在教学辅助网站（www.perasonhighered.com/lewis）上补充了大量程序设计作业。由于本书是面向大二学生的，作业将主要阐述书中的主题而不是扩展的程序设计项目。因此大部分作业的源代码已经提供，只要求学生关注与主题直接相关的部分。部分作业将解决在本书正文中没有涵盖的主题，比如A/D转换、反馈控制和处理器利用率。作业应该按照顺序完成，一些作业基于前面完成的某些作业内容。作业6~10是完整的程序，首先不加修改地运行来说明一个问题，然后修改后来解释一些具体的解决方案。编程作业11~14需要使用网站上所描述的特殊电机开发板。

编程作业	相关章节	备注
01 整数算术运算	2, 3	不需要特殊的硬件
02 工具介绍		实现俄罗斯方块游戏
03 时间测量		运行时间、程序延迟和PWM
04 模拟设备		探讨A/D转换器、PWM和中断（使用C语言）
05 定点实数	3, 5, 6, 7	使用汇编代码
06 处理器利用率		
07 饥饿	10	
08 互斥、信号量和队列	9, 10	这些作业是完整的程序，用于验证多线程程序中提出的问题。学生修改这些程序来学习或改进性能问题
09 优先级反转	10	
10 死锁	10	
11 电机速度控制Ⅰ		需要特殊电机开发板
12 电机速度控制Ⅱ	9, 10	添加简单多任务
13 电机速度控制Ⅲ	9, 10	添加互斥、信号量和队列
14 电机速度控制Ⅵ		添加处理器利用率监测

平台的选择

第5～8章中介绍的处理器体系结构是32位的ARM Cortex-M3。Stellaris LM2S811评估板使用了这款处理器并整合了大量适用的I/O设备，能够分别从德州仪器的EKI-LM3S811开发工具中的一部分来完成一个集成的软件开发环境和驱动库。



EKI-LM3S811评估工具
（经Luminary Micro公司许可使用）

致谢

如果没有很多人的共同努力，本书以及教学辅助网站上的软件不可能完成。在所有这些人中，我要特别感谢圣

克拉拉大学过去十年间 COEN 20 的学生和助教，他们见证了本书从第 1 版到几乎完全重写的第 2 版的过程，也要特别感谢那些帮助调整文字、修订问题和实验作业的人。

我也非常感谢我的很多同事，感谢 Qiang Li 和 Neil Quinn 耐心地使用我的材料进行教学——这可不是什么让人羡慕的事，他们也为最初的文本提出了大量的修改意见。感谢 Hal Brown 和 Vasu Alagar 帮助我开发了定点实数乘法的数学表示，从而能够在汇编语言中进行有效的直接实现，并且能够给出一个让学生相对容易理解的算法阐释。感谢 Darren Atkinson 在 C 语言方面宝贵的专业知识。同样感谢北卡罗来纳州立大学的 Dana Lasher，他提供了中断驱动的 I/O 一节（8.4.2 节）中关于电话呼叫的分析。

我非常感谢北俄亥俄大学的 Samuel Khorbotly 和其他审查了新版本草稿的同仁，他们发现了我在文稿中的错别字、拼写错误、语法错误、遗漏、疏忽和偶尔的彻底错误。我也要表达我对 Prentice-Hall 的工作人员和他们的生产人员的感谢，他们让本书持续一年多的完成过程更加轻松。最后，我要感谢圣克拉拉大学准许我休假，从而使我能够有充足的时间来完成此次修订工作。

目 录

Fundamentals of Embedded Software with the ARM Cortex-M3, 2E

导　论

1.1　什么是嵌入式系统

嵌入式系统是在装置中整合了微处理器的电子设备。使用微处理器的主要目的是简化系统设计，提供高度的灵活性。设备中具有一个微处理器意味着去除 bug、做出修改或者增加新的特征只需要重写控制设备的软件。然而，与 PC 不同，嵌入式系统可能没有磁盘，因此软件常存储在只读存储器（ROM）芯片中，这意味着修改软件需要替换 ROM 或者对其进行“重编程”。

如表 1-1 所示，嵌入式系统广泛应用于不同领域。起初，嵌入式系统只是用于昂贵的工业控制应用，但是随着技术发展降低了专用处理器的价格，嵌入式系统开始出现在中等昂贵的应用当中，例如汽车、通信和办公设备以及电视。今天的嵌入式系统价格相当低廉，它们几乎应用在我们生活中的每个电子产品之中。例如，在 NASA 2003 的火星探测车上（如图 1-1 所示），使用了嵌入式系统来作为整个探测车的一部分。在 Vitality 的 GlowCap（如图 1-2 所示）中使用的超小 ATMEL 8 位超低功耗 AVR 处理器，帮助人们记得按时服药。当药瓶被打开时，它能够感应到，并将信息无线传输给 Vitality 服务器，触发 LED 闪动并播放铃声。

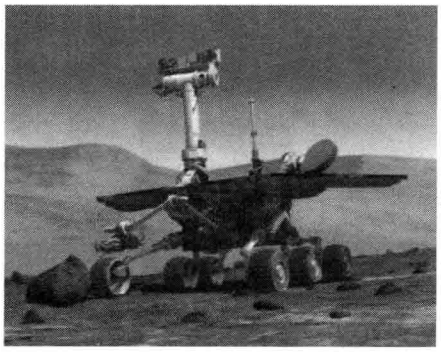


图 1-1　NASA 2003 火星探测车，使用了 BAE 系统，该系统基于 RAD6000 32 位 RISC 处理器和风河系统的 VxWorks 嵌入式实时操作系统

照片由 NASA/JPL/ 康奈尔大学（Cornell University）提供

虽然一个典型的家庭仅仅拥有 1 台或者 2 台个人计算机，但在其居所中、汽车上会发现大量的嵌入式处理器，个人物品中有更多的嵌入式处理器。在多数情况下，我们可能甚至没有意识到计算机的存在，或者没有认识到嵌入式系统是多么普遍。

最令人吃惊的恐怕是嵌入式处理器几乎占全世界微处理器数量的 100%。对应于用于桌面计算机的每个处理器，都有超过 100 个为嵌入式系统而生产的微处理器。如果考虑到 1999 年平均每个北美中产家庭中都能找到 40 ～ 50 个嵌入式微处理器，那么这就一点也不令人感到意外了⊖。可以在如下的产品中找到它们：钟控收音机、立体声音响、电视、VCR、DVD

⊖ Jim Turley, “Microprocessors for Consumer Electronics, PDAs, and Communications,” Embedded Systems Conference, San Jose, CA, September 26 – 30, 1999.

⊖ 此页码为英文原书页码，与索引中的页码一致。——编辑注

播放机、DVR、游戏机、远程控制、洗衣机、烘干机、洗碗机、微波炉、冰箱、咖啡机、温度调节器、草坪喷灌控制器、车库门遥控开关、相机、无线和移动电话、电话应答机、传真机、寻呼机、个人数字助理（PDA）、MP3 播放器、手表甚至牙刷（见图 1-3）。

表 1-1 嵌入式系统实例

类 型	描 述
航空	导航系统、自动着陆系统、飞行高度控制、引擎控制和空间探索
汽车	燃油喷射控制、乘客环境控制、防抱死系统、空气袋控制和 GPS 映射
游戏机	Nintendo DS、Nintendo Wii、微软 Xbox 360 和索尼 Playstation 3
通信	卫星，网络路由器、交换机和集线器
计算机外围设备	打印机、扫描仪、键盘、显示器、调制解调器、硬磁盘和 CD-ROM
家居	洗碗机、微波炉、DVD 播放器、电视、立体声音响、火警 / 安全警报系统、草坪喷灌控制器、温度调节器、相机、钟控收音机和电话应答机
工业	电梯控制、监测系统和机器人
仪器仪表	数据收集、示波器、信号发生器、信号分析器和电源
医疗	成像系统（比如 X 射线、MRI 和超声波）、监护仪和心脏起搏器
办公自动化	传真机、复印机、电话和点钞机
个人	个人数字助理、PMP（Portable Media Player）、便携式电子书阅读器、寻呼机、手机和智能手机

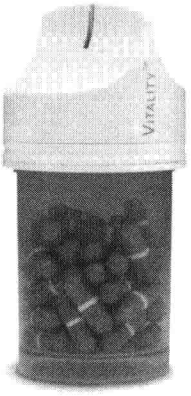


图 1-2 Vitality 的 GlowCap 使用了嵌入式 ATMEL 8 位超低功耗 AVR 处理器

照片由 Vitality 公司提供

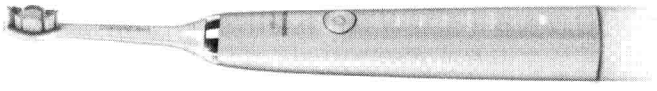


图 1-3 使用 8 位 PIC 微处理器的飞利浦 Sonicare DiamondClean 牙刷

照片由飞利浦优质生活提供

那么我们的汽车呢？2005 年，平均每辆豪车有 150 个左右的嵌入式微控制器和微处理器用于燃油喷射系统、防抱死系统、气囊控制、环境控制、GPS 定位、座椅和安全带、广播等系统与设备的微处理器，其数量是 2002 年的三倍[⊖]！

⊖ “Embedded Systems Development Trends: Asia,” EE-Time-Asia & Gartner Dataquest, February 2005.

1.2 嵌入式软件设计的目标有什么独特性

本书旨在帮助读者学习如何为嵌入式系统设计和实现软件。尽管你可能具有使用高级语言编写桌面应用程序的经验，但是编写嵌入式应用程序时，你会在可靠性（reliability）、性能（performance）和成本（cost）等方面面临挑战。

图 1-4 Vue40 自动售货机使用了 16 位 Hitachi H8/3007 CPU
照片由 SandenVendo Amertca 公司提供

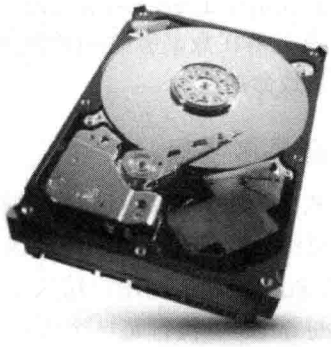


图 1-5 希捷 Barracuda XT 磁盘，使用了两个 ARM Cortex-R4 处理器：一个用于控制伺服系统，另外一个用于处理器命令和数据流
照片由希捷科技有限公司提供

可靠性预期会让程序员承担更大的责任来消除错误（bug）并设计软件来容忍错误和意外情况。许多嵌入式系统必须一年 365 天、每天 24 小时持续运行。出现错误时不能仅仅依靠“重启”！因此，好的编程实践和全面的测试在嵌入式处理器领域中的重要程度达到了新的高度。

性能目标会迫使我们学习和应用诸如多线程（multi-threading）和调度（scheduling）一类的新技巧。直接与传感器、驱动器、小键盘、显示器等设备进行通信的要求使得程序员需要更为深入地理解如何为改进输入 / 输出提供替代方法，以便在速度、复杂性和成本之间取得平衡。虽然我们经常会为了更高的效率而使用高级语言编程，使用这些替代方法则可能偶尔需要我们直接用汇编语言（assembly language）深入到计算机和程序中。

计算机使用固定精度和 2 的补码表示来存储和处理数值。这就引起了人们处理数字的细微差别，这通常是问题出现的不可预期的来源。我们的软件将不得不与有范围和精度限制的数字共存。然而作为程序员，我们需要确保我们全面了解超出这些限制的后果以及如何处理这种情况。

与嵌入在大型昂贵的系统中的处理器不同，消费类产品被设计为以最低的成本来进行大规模的生产（图 1-4 至图 1-6 都是这方面的例子）。为了具有竞争力，它们必须实现最快的上市时间，并且最理想的情况是一旦投入生产就不需要再修改。

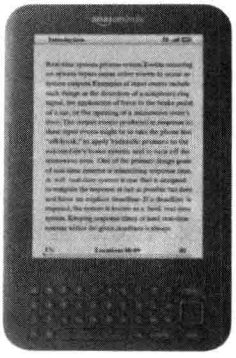


图 1-6 亚马逊 Kindle 2 使用了 32 位 ARM 处理器
照片由希捷科技有限公司提供

1.3 什么是实时系统

实时系统处理事件。系统输入中发生的事件会导致作为系统输出的其他事件的发生。输入事件的例子包括电话铃信号的检测、使用汽车强制制动踏板或者打开微波炉的门之类的事情。在对这些输入事件的响应中产生的输出事件可能会对电话进行“摘机”、应用汽车液压刹车系统或者关闭微波炉。

实时系统的主要设计目标之一是减少响应时间。软实时系统会尽快算出响应，但是并没有明确的最后期限。如果加上了最后期限，该系统就是一个硬实时系统。将硬实时系统的响应时间控制在给定的最后期限之内是非常重要的，否则整个系统可能无法正常运转。比如，防抱死系统必须在数毫秒之内检测并响应牵引力的丢失，1 秒或 2 秒的延迟是不能接受的，并且可能是致命的。

1.4 什么是多线程

嵌入式系统是具有若干输入和输出且必须响应多个独立事件的实时系统。例如，控制暖气设备的可编程温度调节器不断地执行以下三个任务：1）检测温度；2）监控当天的具体时刻；3）扫描小键盘上的用户输入。温度调节器中只有一个计算机，但可以通过单独的代码线程（code thread）执行每一个上述任务，如图 1-7 所示。尽管线程彼此通信，但多数时间下它们独立运行。

```
do forever {
  measure temp ;
  if (below setting)
     start furnace ;
  else if (above setting)
     stop furnace ;
}
```

```
do forever {
  measure temp ;
  if (time == 6am)
     temp = 72° ;
  else if (time == 11pm)
     temp = 60° ;
}
```

```
do forever {
  check keypad ;
  if (raise temp)
     setting++ ;
  else if (lower temp)
     setting-- ;
}
```

图 1-7 可编程温度调节器中的三个并发代码线程

2
~
5

由于分离代码线程允许程序员一次只关注一个任务，简化了软件的开发和维护。不过，这就需要处理器在线程之间不断地进行切换。从表面上看这种多线程的处理过程使多线程同时运行，但实际上在任何时刻仅有一个线程在运行。这些线程称为并发（concurrently）执行。为了清楚地进行说明，我们将任务（task）定义为一个将要完成的作业，线程（thread）定义为指令的序列。一个任务可能会被划分为一个或多个并发线程，但是一个线程可能会有助于一个或者多个任务。

1.5 嵌入式处理器到底有多强大

微控制器单元（MicroController Unit，MCU）在单一芯片上集成了包括处理器、内存、I/O 端口、定时器和多个外设在内的器件。过去典型的 MCU 使用 8 位或 16 位的处理器。现今智能手机、GPS 设备、PMP 和家用网络路由器等智能型设备日益增长的需求推动 MCU 采用 32 位处理器，以提供更高的性能和更多的功能。如图 1-8 所示，在新的嵌入式设计中 32 位处理器正在快速取代其前任的地位。