国外优秀信息科学与技术系列教学用书

# FUNDAMENTALS OF EMBEDDED SOFTWARE

## Where C and Assembly Meet

# 嵌入式软件基础

翻译版

## ——C语言与汇编的融合

[美] Daniel W. Lewis 著

陈宗斌 译

PEARSON Prentice Hall

高等教育出版社
Higher Education Press

国外优秀信息科学与技术系列教学用书

# 嵌入式软件基础

## ——C 语言与汇编的融合

（翻译版）

**FUNDAMENTALS OF EMBEDDED SOFTWARE**
**Where C and Assembly Meet**

[美] Daniel W. Lewis 著

陈宗斌 译

高等教育出版社

**图字：01-2004-4684 号**

嵌入式软件基础——C 语言与汇编的融合(翻译版)
**Fundamentals of Embedded Software: Where C and Assembly Meet**
[美] Daniel W. Lewis 著，陈宗斌 译

本书封面贴有 Pearson Education(培生教育出版集团)激光防伪标签。无标签者不得销售。



原版 ISBN: 0-13-061589-7

**图书在版编目(CIP)数据**

嵌入式软件基础——C 语言与汇编的融合：翻译版/[美]刘易斯（Lewis, D.W.）著；陈宗斌译. —北京：高等教育出版社，2005.5

书名原文：Fundamentals of Embedded Software: Where C and Assembly Meet

ISBN 7-04-016105-2

Ⅰ. 嵌… Ⅱ. ①刘… ②陈… Ⅲ. ①C 语言-程序设计-高等学校-教材 ②汇编语言-程序设计-高等学校-教材 Ⅳ. ①TP312 ②TP313

中国版本图书馆 CIP 数据核字（2005）第 025234 号

| | | | |
|---|---|---|---|
| **出版发行** | 高等教育出版社 | **购书热线** | 010-58581118 |
| **社　　址** | 北京市西城区德外大街 4 号 | **免费咨询** | 800-810-0598 |
| **邮政编码** | 100011 | **网　　址** | http://www.hep.edu.cn |
| **总　　机** | 010-58581000 | | http://www.hep.com.cn |
| | | **网上订购** | http://www.landraco.com |
| **经　　销** | 北京蓝色畅想图书发行有限公司 | | http://www.landraco.com.cn |
| **印　　刷** | 北京嘉实印刷有限公司 | | |
| **开　　本** | 787×1092　1/16 | **版　　次** | 2005 年 5 月第 1 版 |
| **印　　张** | 15.5 | **印　　次** | 2005 年 5 月第 1 次印刷 |
| **字　　数** | 340 000 | **定　　价** | 29.00 元（含光盘） |

本书如有缺页、倒页、脱页等质量问题，请到所购图书销售部门联系调换。



**物料号　16105-00**

# 出 版 说 明

20 世纪末，以计算机和通信技术为代表的信息科学和技术对世界经济、科技、军事、教育和文化等产生了深刻影响。信息科学技术的迅速普及和应用，带动了世界范围信息产业的蓬勃发展，为许多国家带来了丰厚的回报。

进入 21 世纪，尤其随着我国加入 WTO，信息产业的国际竞争将更加激烈。我国信息产业虽然在 20 世纪末取得了迅猛发展，但与发达国家相比，甚至与印度、爱尔兰等国家相比，还有很大差距。国家信息化的发展速度和信息产业的国际竞争能力，最终都将取决于信息科学技术人才的质量和数量。引进国外信息科学和技术优秀教材，在有条件的学校推动开展英语授课或双语教学，是教育部为加快培养大批高质量的信息技术人才采取的一项重要举措。

为此，教育部要求由高等教育出版社首先开展信息科学和技术教材的引进试点工作。同时提出了两点要求，一是要高水平，二是要低价格。在高等教育出版社和信息科学技术引进教材专家组的努力下，经过比较短的时间，第一批由教育部高等教育司推荐的 20 多种引进教材已经陆续出版。这套教材出版后受到了广泛的好评，其中有不少是世界信息科学技术领域著名专家、教授的经典之作和反映信息科学技术最新进展的优秀作品，代表了目前世界信息科学技术教育的一流水平，而且价格也是最优惠的，与国内同类自编教材相当。这套教材基本覆盖了计算机科学与技术专业的课程体系，体现了权威性、系统性、先进性和经济性等特点。

目前，教育部正在全国 35 所高校推动示范性软件学院的建设，这也是加快培养信息科学技术人才的重要举措之一。为配合软件学院的教学工作，结合各软件学院的教学计划和课程设置，高等教育出版社近期聘请有关专家和软件学院的教师遴选推荐了一批相应的原版教学用书，正陆续组织出版，以方便各软件学院开展双语教学。

我们希望这些教学用书的引进出版，对于提高我国高等学校信息科学技术的教学水平，缩小与国际先进水平的差距，加快培养一大批具有国际竞争力的高质量信息技术人才，起到积极的推动作用。同时我们也欢迎广大教师和专家们对我们的教材引进工作提出宝贵的意见和建议。联系方式：hep.cs@263.net。

高等教育出版社

二〇〇二年九月

# 序　言

你的家中有多少台计算机？大多数人可能会回答有两台或三台。你家中有多少个**微处理器**？在你回答前仔细考虑一下（提示：不止两三个！事实上，甚至会有10多个或20多个！）。今天，微处理器已嵌入到几乎所有你能想到的以及许多你可能想不到的电器中。它们已变得非常普及——不仅在我们家中，而且在我们的工作场所、我们的汽车、飞机、停车灯、超市、手机中——总之，在我们生活的几乎所有方面。

嵌入式系统为学生表达他们的创造力提供了一个令人兴奋的机会，他们梦想设计出下一代新的小器具，迎合大众的需要。作为教育工作者，我们的挑战是抓住这个激动人心的机会，并引导年轻人的精神活力，以激励他们掌握这个主题。

## 目标

本书的最终目标是打下一个基础，为学生掌握多线程编程风格和嵌入式软件的高可靠性要求提供支持。在本教材中，我们建立了以下目标：

1. 理解如何在机器级表示数据，认识到这些表示的后果和局限。
2. 掌握嵌入式系统最常用的语言特有的特性，如位操纵和变量访问。
3. 了解程序员如何看待处理器体系结构以及如何在汇编级编程，有时是必要的或合适的。
4. 了解多种不同风格的I/O编程，并且最终了解如何通过事件驱动方法把数据处理隔离到许多独立的计算线程中。
5. 了解非抢先式和抢先式多线程编程、共享资源和临界区，以及如何使用调度来管理系统响应时间。
6. 复习作用域、参数传递、递归和内存分配这类主题，以强化基本的编程技能。
7. 了解与共享内存对象有关的问题、内存分配如何影响共享内存以及可以使用哪些编程实践来最小化共享内存的发生。

## 目标读者

本书打算用作计算机科学、计算机工程或电子工程专业课程二年级教材，用来代替计算机组织和汇编语言编程的传统教材。

本书介绍了实践中最常用的汇编方法——以实现小型、快速或专用例程供主程序调用，这些主程序是用诸如C之类的高级语言编写的。因此，本书仅从“须知”的观点出发来介绍处理器组织和汇编语言，而不是将其作为一个主要目标来讲解。这种方法为用本书讲授嵌入式软件环境中的汇编语言提供了时间保证。因而，学生不仅学习了具有重要作用的汇编技能，而且他

们对多线程编程、抢先式和非抢先式系统、共享资源和调度的发现，还可以帮助维持他们的学习兴趣，满足他们的好奇心，同时进一步做好对操作系统、实时系统、网络和微处理器基础设计等后续课程的准备工作。

在大多数学院中，介绍性编程系列课程（CS1、CS2）不再用过程性编程语言（如 C 或 Pascal）来讲授；相反，当前流行的方法是使用一种面向对象的编程语言（如 C++或 Java）。尽管有了这种变化，人们还是能经常见到一种或多种后续分支课程仍然使用过程性语言，并且也能经常见到在工业中仍在使用这类语言。在作者所在的学院中，我们通过围绕本书中的材料重新设计我们传统的汇编语言课程，解决了这种矛盾；它不但为已有的课程拓展了空间，以介绍过程性方法和嵌入式系统流行的主题，而且还为加强学生对参数传递、作用域和内存分配模式的综合理解提供了一个机会，在 CS1 和 CS2 中首先会讲授这些主题。

本书假定学生已经知道了如何用 C、C++或 Java 编程，这些语言的低级语法之间的相似性使学生从 C++或 Java 转到 C 相对容易。本书没有繁琐地介绍 C，而是重点强调了嵌入式应用中频繁用到的 C 的特性，并通过例子和编程作业介绍了过程性风格，那些例子和作业中包含了大量预先编写好的源代码。因此，原则上绝对的先修课程仅是使用 C、C++或 Java 的 CS1 课程。但是，强烈建立你掌握额外的编程经验，如从 CS2 课程中学到的关于数据结构的编程经验。

## 编程作业和配书光盘

本书中的补充材料包括附录 D 中描述的一组编程作业。假定本书的目标读者是大学二年级学生，这些作业主要用于阐释本书中的一个主题，而不是作为扩展的编程项目。因而，配书光盘上为每一道作业题都提供了大部分源代码，并且要求学生仅关注那些直接与主题相关的部分。例如，最后三道作业题提供了完整的源代码，用于图形化地演示与共享资源、优先级倒置及死锁相关的问题，并且要求学生使用本书中介绍的策略修改代码的特定部分，来解决这些问题。

下表汇总了这些编程作业，并说明了根据本书内容何时应布置这些作业。

| 编程作业 | 相关章节 | 说　明 |
|---|---|---|
| 1 | — | 介绍 C 和 DJGPP 编译器；不需要了解本书中的内容，可以安排在课程的第一周。 |
| 2 | 2 | C 中固定精度的实数。 |
| 3 | 3 | C 中的宏和组合操作数。 |
| 4 | — | makefiles：学习第 4 章时通常会布置的这一道作业，但是与该章中的材料没有直接关系。 |
| 5 和 6 | 5 | 汇编语言编程。 |
| 7 | 6 | 汇编语言中中断驱动的 I/O。 |
| 8 | 7 | 具有非抢先式内核的多线程编程。 |
| 9 | 7 | 抢先式内核、共享资源和信号。 |
| 10 和 11 | 8 | 调度问题（优先级倒置和死锁）。 |

在配书光盘上，你还可以发现用于开发 Microsoft Windows 9x/2000/NT 下的嵌入式应用程序所需要的所有软件工具：GNU C 编译器和连接器的 DJGPP 端口、兼容的汇编器和运行时库。还提供了一个引导加载程序，用于把嵌入式应用程序从磁盘加载到内存中（并执行）。在可能的地方还包括了用于所有这些工具的源代码。在附录 D 的开始处可以找到使用所有这些工具的指导。

## 平台的选择

本书使用无所不在的 PC 作为学习处理器体系结构、汇编语言和嵌入式软件的平台。有两个主要因素促成我们做出了这种选择：(1) 由于 Intel 体系结构在 PC 市场上的统治地位，导致学生最终必将遇到这种体系结构；(2) 选择在 PC 上构建嵌入式应用程序具有附加的好处，它可以让教师提供关联的实验组件，而无需投资购买专门的单板计算机。

大多数针对 Intel 处理器的汇编语言编程教科书都涉及了原始的 8088 “实” 模式。但是，386 和后来的 Intel 处理器保护模式更能代表现代的体系结构，并且当配置成使用 “平坦” 内存模型时使编程实际上更容易。尽管本书中对实模式作了简短介绍，但重点还是放在保护模式上，并将其用在所有汇编语言例子中。

人们不应该把 PC 看作用于构建**实际的**嵌入式系统的最佳（甚至一个合适的）平台。大多数嵌入式应用程序不需要 PC 的许多标准外围设备（如显示器、键盘、磁盘），而 ROM BIOS 中的通电启动过程甚至不支持无盘应用程序。尝试编写初始化代码来代替 BIOS 是一个最大的难点，因为它需要与芯片组相关的特性知识，而这些特性随 PC 的不同而有所不同，并且它们通常是专有的。

## 致谢

没有许多人的努力，就不会产生本书和配书光盘中的软件。在所有这些人当中，我要特别感谢圣克拉拉大学 COEN 20 的学生和助教，他们通读了原稿的早期粗糙的版本，并帮助调试了 libepc 库例程。

我还要向我的许多同事致以诚挚的谢意：感谢 Qiang Li 和 Neil Quinn 忍受用我的材料来从事不值得羡慕的教学工作，并善意地指出了对本书的许多改进意见；感谢 Hal Brown 和 Vasu Alagar 帮助我开发了实数的定点乘法的数学表示，它导致了汇编中有效的直线式实现，以及第 2.3.3 节中的一个算法说明，学生们发现它相对容易理解；感谢 Darren Atkinson 把两天中最好的时光花在我的办公室中，来跟踪无伤大雅的小软件错误，而该错误仅在我升级到最新版本的编译器时才会出现；还要衷心感谢我的原稿的早期审阅者，他们的评论帮助提高了本书的最终质量，他们是：亚利桑那州立大学的 Walter Higgins 博士、奥斯汀得克萨斯大学的 Karram Mossaad 博士、北卡罗来纳州立大学的 Dana Lasher 博士。我要特别感谢 Lasher 博士，他贡献了一个电话呼叫模拟，它出现在第 6.4 节关于事件驱动型 I/O 的开头段落中。

我们应该倾心感谢那些无私的程序员，他们创建了具有专业质量的软件工具，并使之准备

好为每个人所用：感谢 DJ Delorie 的 GNU C 编译器的 DJGPP 端口；感谢 Simon Tatham 和 Julian Hall 的 NASM 汇编器；感谢 Jean Labrosse 的 μC/OS-II 抢先式实时内核；还要感谢 MIX Software 的 Dennis Saunders 的 Multi-C 非抢先式实时内核。

Daniel W. Lewis

Prentice-Hall 配套 Web 站点：
www.prenhall.com/divisions/ems/app/lewis

# 目　录

# 第1章 导 论

## 1.1 什么是嵌入式系统

嵌入式系统是指在其实现中纳入了微处理器的电子设备。微处理器的主要用途是简化系统设计和提供灵活性。设备中具有微处理器就意味着，消除错误、执行修改或添加新特性时，所需要做的只是改写用于控制设备的软件。但是，与 PC 不同的是，嵌入式系统可能没有磁盘驱动器，因此软件通常存储在只读存储器(ROM)芯片中；这意味着修改软件需要更换或“重新编程”ROM。

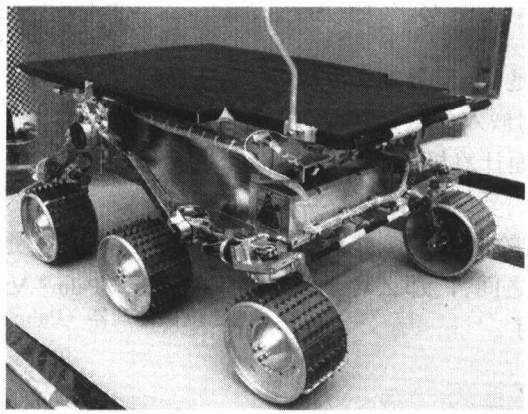

图 1-1 NASA 的火星探路者使用一个 Intel 80C85 8 位微处理器（NASA/JPL 惠许）

如表 1-1 所示，嵌入式系统具有广泛的应用领域。最初它们仅用于昂贵的工业控制应用，但是当技术降低了专用处理器的成本时，它们开始出现在适度昂贵的应用中，如汽车、通信和办公设备以及电视机。今天的嵌入式系统是如此廉价，以至于它们可以用在我们生活中的几乎所有的电子产品中。

**表 1-1 嵌入式系统的例子**

| 应 用 领 域 | 例 子 |
| --- | --- |
| 航空航天 | 导航系统、自动着陆系统、飞行高度控制、引擎控制、空间探测(例如，火星探路者) |
| 汽车 | 燃油喷射控制、乘客环境控制、防抱死制动系统、安全气囊控制、GPS 导航 |
| 儿童玩具 | Nintendo 的 Game Boy、Mattel 的 My Interactive Pooh、Tiger Electronic 的 Furby |
| 通信 | 卫星、网络路由器、交换机、集线器 |

续表

| 应用领域 | 例 子 |
| --- | --- |
| 计算机外围设备 | 打印机、扫描仪、键盘、显示器、调制解调器、硬盘驱动器、CD-ROM 驱动器 |
| 家庭 | 洗碗机、微波炉、录像机(VCR)、电视机、立体声系统、火灾/安全警报系统、草地洒水器控制、恒温箱、照相机、时钟收音机、应答机 |
| 工业 | 电梯控制、监视系统、机器人 |
| 仪器 | 数据采集、示波器、信号发生器、信号分析器、电源 |
| 医疗 | 成像系统(例如，X 射线、核磁共振成像、超声)、患者监视器、心脏调搏器 |
| 办公自动化 | 传真机、复印机、电话机、收银机 |
| 个人 | 个人数字助理(PDA)、寻呼机、手机、手表、电视游戏、便携式 MP3 播放机、GPS |

在许多情况下，我们甚至没有意识到计算机的存在，因而也就没有认识到它们已变得多么普及。例如，尽管一般的家庭可能拥有一两台个人计算机，但是在家中、汽车中以及人们的个人物品中，可以发现的嵌入式计算机的数量却是非常之多。

人们经常感到惊奇的是，全世界的微处理器生产实际上 100%都使用嵌入式微处理器！每有一块微处理器用于桌面计算机，就有 100 块以上的处理器用于嵌入式系统。考虑到 1999 年在北美中等家庭中发现的嵌入式微处理器数量平均估计在 40~50 块之间[①]，那么对于这种情况就不应该感到奇怪了。

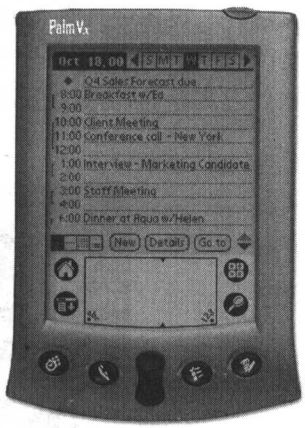


图 1-2 手持式 Palm™ Vx 使用 Motorola Dragonball EZ 32 位微处理器（Palm 是 Palm 公司的注册商标）

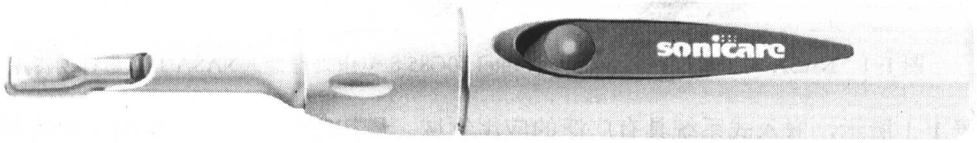


图 1-3 Sonicare Plus 牙刷使用了一个 Zilog Z8 8 位微处理器（飞利浦口腔保健事业部惠许）

## 1.2 嵌入式软件设计目标有何独特之处

本书的目标是帮助你了解如何设计和实现用于嵌入式系统的软件。尽管你已经具备了一些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还是会遇到一些关于**可靠性**、

① Jim Turley，“用于消费电子、PDA 和通信的微处理器”，嵌入式系统会议，加利福尼亚州圣何塞市，1999 年 9 月 26 日到 30 日。

**性能**和**成本**的新挑战。

人们对可靠性的期望赋予了程序员更大的责任来消除错误，并把软件设计成容许错误和不可预知的情况。许多嵌入式系统必须 1 年 365 天、1 周 7 天、1 天 24 小时持续不停地运行，当发生某些错误时，不能进行“重启动”。出于这种原因，良好的编码实践和彻底测试的重要性在嵌入式处理器领域达到了一个新的高度。

性能目标迫使我们学习和应用新技术，如**多任务处理**和**调度**。与传感器、调节器、键盘、显示器等直接通信的需要，要求程序员更好地理解那些用于输入和输出的替代方法如何为交易速度、复杂性和成本提供机会。尽管我们通常用高级语言编写程序，以获得更高的效率，但是使用这些替代方法偶尔也需要我们把计算机和程序直接降低到**汇编语言**级别。

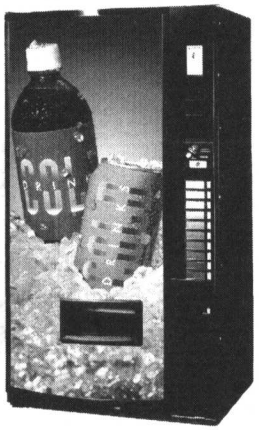

图 1-4 Vendo V-MAX 720 自动售货机使用 Motorola M68HC11 8 位微处理器（Vendo 公司惠许）

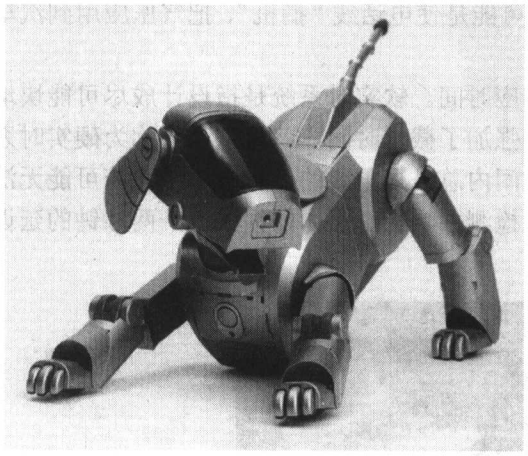

图 1-5 Sony 的 Aibo ERS-110 机器狗使用一个 MIRS 64 位 RISC 处理器（Sony 电子公司惠许）

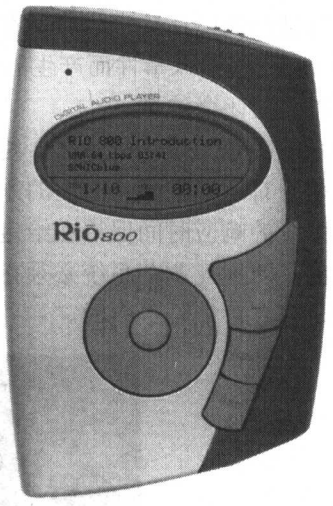


图 1-6 Rio 800 MP3 播放机使用一个 32 位 RISC 处理器（SONICBlue 公司惠许）

计算机使用**固定精度**和 **2 的补码**表示来存储和处理数值，这导致人们处理数字的方式有一些细微的差别，并且这通常是一个不可预知的问题源。我们的软件不得不与有关数字的范围和解析限制共存。但是作为程序员，我们必须确保全面理解了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的是，消费品设计成用最少的制造成本进行大批量生产。为了具有竞争力，它们必须快速上市，并且理想地要求在生产中不进行任何修改。

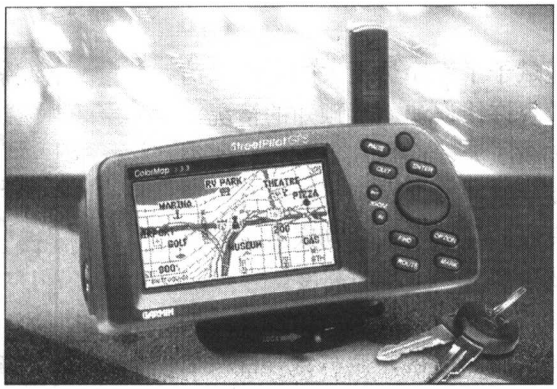


图 1-7 Garmin StreetPilot GPS 接收器使用一个 16 位处理器（Garmin International 公司惠许）

## 1.3 “实时”意味着什么

实时系统处理事件。发生在系统输入上的事件引发其他事件发生，作为系统输出。输入事件的例子包括检测电话铃声信号、在汽车的制动踏板施加作用力或者打开微波炉的门这样的事情。用于响应这些输入事件而产生的输出事件可能是使电话线“摘机”、把气压应用到汽车的制动系统上以及关闭微波炉。

实时系统的基本设计目标之一是最小化响应时间。**软**实时系统是指设计成尽可能快地计算响应，但是没有明确的截止时间的系统。如果强加了截止时间，就将该系统称为**硬**实时系统。把硬实时系统的响应时间保持在给定的截止时间内总是很重要的，否则整个系统可能无法正确工作。例如，防抱死制动系统必须在几毫秒内检测并响应牵引力的损失；一两秒钟的延迟是不可容忍的，并且可能是致命的。

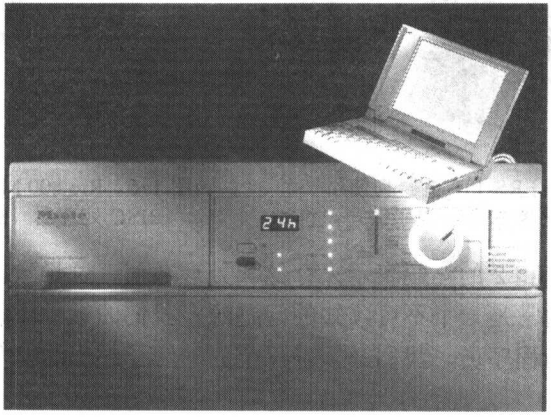


图 1-8 Miele 洗碗机使用一个 8 位 Motorola 68HC05 微处理器，并且可以用笔记本电脑重新编程，以调整温度和周期时间（Miele & Cie. GmbH & Co 公司惠许）

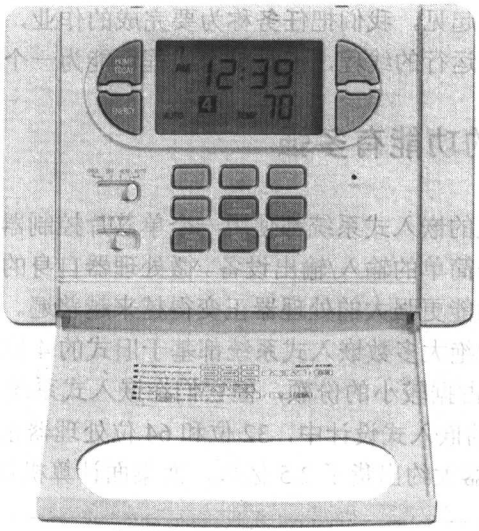

图 1-9 Hunter 44550 可编程恒温箱使用一个 4 位微处理器（Hunter Fan 公司惠许）

## 1.4 “多任务”意味着什么

嵌入式系统是具有多个输入和输出的系统，并且必须响应多个独立的事件。例如，控制炉子的可编程恒温器必须持续不断地执行 3 项**任务**：(1) 监测温度；(2) 监测一天中的时间；(3) 扫描键盘上的用户输入。恒温箱内仅有一台计算机，但它使用一个单独的代码**线程**来执行所有这些任务，如图 1-10 所示。尽管这些线程彼此之间会进行通信，但在很大程度上它们是独立运行的。

```
/* Monitor Temperature */
do forever {
    measure temp ;
    if (temp < setting)
        start furnace ;
    else if (temp > setting + delta)
        stop furnace ;
}
```

```
/* Monitor Time of Day */
do forever {
    measure time ;
    if (6:00am)
        setting = 72°F ;
    else if (11:00pm)
        setting = 60°F ;
}
```

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

图 1-10 可编程恒温箱内的三个并发代码线程

把代码分隔成多个线程，使程序员每次集中完成一项任务，从而简化了软件的开发和维护。但是，它也要求处理器以某种方式持续地在这些线程之间切换。这种**多任务处理**的过程给人一种这些线程在同时执行的**表象**，但是实际上在任何一个瞬间只有一个线程被执行。我们把这些