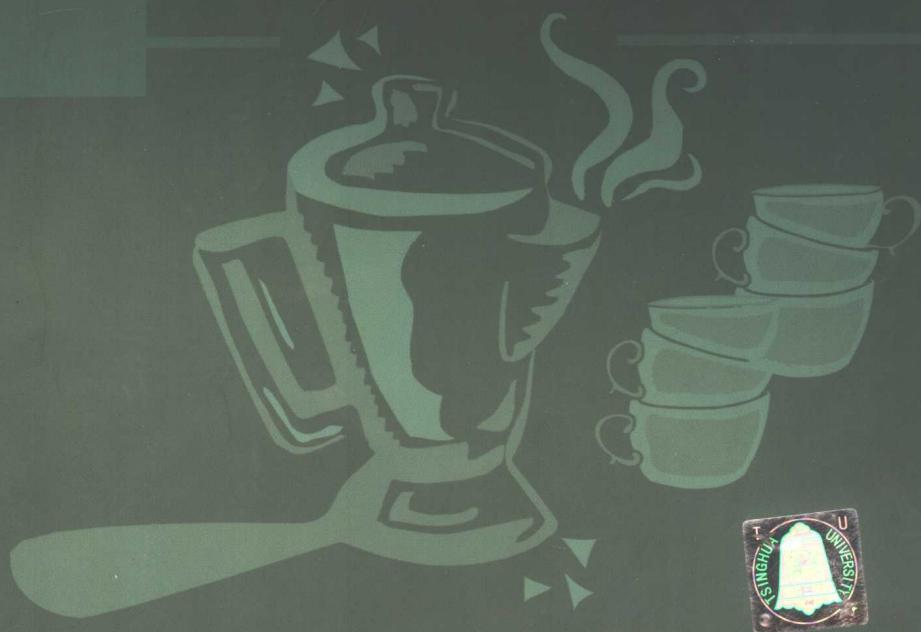


嵌入式实时多任务软件 开发基础

晨 风 编著



清华大学出版社

嵌入式实时多任务软件开发基础

晨 风 编著

清 华 大 学 出 版 社

北 京

内 容 简 介

随着嵌入式系统应用范围的日益广泛和深入,从事嵌入式系统开发工作的人员数量也在迅速增加。本书可以帮助从事嵌入式系统开发工作时间不长的新手们尽快掌握嵌入式系统开发的基本方法,理解嵌入式软件开发的基本概念,掌握编写稳定、高效的嵌入式系统的基本能力。

本书以一个具体的嵌入式系统——自动售茶机的开发过程作为参考范例,围绕这个简化的嵌入式系统模型,以5个章节的篇幅,对嵌入式系统的特点,商用嵌入式软件开发的流程和品质控制等进行了针对性介绍,展示了嵌入式系统及嵌入式软件开发的各个方面,提出了嵌入式多任务软件开发中的一些常见问题及基本的解决思路。

本文作者拥有对本书所含信息的所有权,请严格保密,不得复制本书的部分或全部内容。未经本书作者书面许可,不得用于产品设备的设计、生产及销售事宜。

版权所有,翻印必究。举报电话:010-62782989 13901104297 13801310933

本书封面贴有清华大学出版社激光防伪标签,无标签者不得销售。

图书在版编目(CIP)数据

嵌入式实时多任务软件开发基础/晨风编著. —北京:清华大学出版社,2004.9
ISBN 7-302-09188-9

I. 嵌… II. 晨… III. 微型计算机—系统设计 IV. TP360.21

中国版本图书馆CIP数据核字(2004)第081470号

出 版 者:清华大学出版社

<http://www.tup.com.cn>

社总机:010-62770175

地 址:北京清华大学学研大厦

邮 编:100084

客户服务:010-62776969

组稿编辑:丁 岭

文稿编辑:李 晔

封面设计:付剑飞

印 刷 者:世界知识印刷厂

装 订 者:北京鑫海金澳胶印有限公司

发 行 者:新华书店总店北京发行所

开 本:185×260 印张:12 字数:300千字

版 次:2004年9月第1版 2004年9月第1次印刷

书 号:ISBN 7-302-09188-9/TP·6466

印 数:1~4000

定 价:24.00元

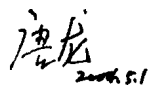
本书如存在文字不清、漏印以及缺页、倒页、脱页等印装质量问题,请与清华大学出版社出版部联

序（一）

当见到本书作者送来一摞书稿，信手翻阅了几页，读着读着，我的注意力很快就被完全吸引住了，显然，这是一本能够适应当前需求的书。读完全部书稿之后，我确信这是应该尽快摆在读者面前的好书。近两三年来，一些朋友和学生找我讨论的有关嵌入式技术的许多问题，在本书中大都能得到很好的启迪，甚至有明确的答案。

众所周知，在人类的科技发展史上，计算机科学与技术的发展无疑是最迅猛的，而且影响最广泛、最深刻。它不仅促进了其他尖端科学技术迅速发展，推动了现代经济腾飞，也造就了一批计算机科技精英和富豪。半个多世纪前问世的第一台计算机是摆放在大学实验室中的宠然大物，需要巨额投资，并且只能作为研发尖端军工产品所用的快速计算工具。如今，可以毫不夸张地说，以计算机软硬件为基本的信息科技已无处不在、无所不为。形成这种情况的关键因素之一就在于嵌入式系统开发和应用。进入 21 世纪，在这互联网时代，也是后 PC 时代，嵌入式系统开发和应用更成为人们关注的目标，每年以数千亿元计的巨额市场预测，更引发了投资者和科技开发者的热情。值得一提的是，就我国国情而言，从企事业单位的现有财力、科技水平和基础设施等客观条件来看，嵌入式系统的开发和应用，对我国信息产业乃至制造业提供了前所未有的难得机遇。目前，国内许多研究院（所）、高等院校、公司和厂商都看好这方面的发展前景，并正着手进行嵌入式 CPU 及其外围芯片等硬件研制、嵌入式系统和应用软件开发，以及嵌入式系统产品集成工作。

面对新机遇和新挑战，新入伍者急需学习相关的基础理论知识，借鉴先行者实践经验教训。本书作者以其名牌大学毕业生扎实的知识功底，加上十来年实践所积累的宝贵经验，奉献出了这部颇具特色的作品。全书结合剖析一个典型的嵌入式实时多任务系统实例——自动售茶机，用通俗易懂的语言，全面阐述了嵌入式实时多任务系统软件开发的基本问题。这是一本值得推荐的好书。希望本书的出版，能进一步推动我国嵌入式系统及其应用的研究和教学，使我国嵌入式系统研发者能把握住机遇，取得更大成果。



（清华大学计算机系教授）

序（二）

嵌入式系统比服务器和 PC 的应用更为广泛，甚至可以说是无处不在。实际上，与服务器和 PC 这种专门的计算机系统相比，运行嵌入式软件各类智能化电子装置对全球 GDP 的贡献要大得多。IDC 的研究报告指出，2003 由嵌入式软件驱动的电子产品对全球 GDP 的贡献在 1 万亿美元以上，而全球一年的 GDP 也不过是 40 多万亿美元。可见，嵌入式软件开发是一个非常值得重视的领域。

随着嵌入式应用的普及，越来越多的软件工程师进入了嵌入式软件开发的行列，而他们当中大部分只有桌面系统的软件开发经验。从开发方式上来讲，如果一个刚刚进入嵌入式软件开发领域的工程师能够注意以下几个方面，就会比较快速地掌握嵌入式软件开发的特点。

- 交叉开发环境。在桌面系统软件开发中，开发使用的计算机（称作“宿主机”）可能是 PC/Windows，开发出来的程序所运行的计算机（称作“目标机”）平台依然是 PC/Windows。也就是说，宿主机和目标机是同一台机器。这样的开发方式称作“本地”（native）开发方式。而在嵌入式软件开发中，宿主机和目标机往往是不同的，如宿主机是 PC 而目标机可能是基于 PowerPC 的单板计算机。宿主机与目标机之间通过以太网、串口、仿真器或其他方式连接起来。这种开发方式称作“交叉”（cross）开发方式。显然开发者所需要的开发环境也必须是交叉开发环境，如编译器必须是交叉编译器（编译器本身运行在宿主机上，而产生的代码却只能运行在目标机上），调试器必须是交叉调试器（调试器本身运行在宿主机上，而它调试的程序却运行在目标机上）。Wind River 的 Tornado 就是这样一个集成开发环境。
- 存储定位。在桌面系统软件开发中，开发者不用关心他所开发出来的可执行程序（如 EXE 文件）运行在内存的什么位置。这样的可执行程序是浮动地址程序。操作系统在将该程序装入执行时对程序进行最后的存储定位。而在嵌入式软件开发中，开发出来的程序是要直接运行在物理内存中的，因此程序员要负责指定代码、静态数据以及堆（heap）在物理内存中的位置（如果使用了 MMU，那么这里指定的是逻辑地址）。这一工作是通过连接器（linker）的命令行文件完成的。因此嵌入式程序员要清楚目标机上的存储器布局（memory layout）。
- 多任务。在桌面系统软件开发中通常采用的是面向过程（如在 UNIX 环境下）或面向事件（如在 Windows 环境下）的开发方式。但无论哪种方式，其程序往往都是单一执行行为的，或称作单任务的。而在嵌入式软件开发中，开发出来的程序都是多任务的。因此程序员必须有多任务的概念，并且知道如何协调与同步多任务的执行。多任务程序的调试要比单任务程序的调试复杂得多，因此需要更强有力的调试工具。
- 嵌入式实时操作系统。在桌面系统软件开发中，我们所接触到的操作系统都是一套完整的可执行程序。程序员往往不需要对操作系统进行裁剪和配置。而嵌入式实时操作系统都是可裁剪的。通常嵌入式实时操作系统是以目标文件（OBJ 文件）或库文件（LIB 文件）的方式提供给程序员的。程序员需要根据自己的需要选择所要的

操作系统功能。通常嵌入式实时操作系统都提供了这样的工具。另外对于选中的操作系统模块还要进行参数配置。由此也可看出典型的嵌入式可执行程序是将操作系统和应用程序连接在一起的。

如果了解了以上几点差别并且有桌面系统软件开发的经验(主要是 C 语言),那么学习嵌入式软件开发并不难。当然要成为一名优秀的嵌入式软件工程师还需要掌握很多技巧。比如,如何将一个应用划分成多个任务?如何分配任务的优先级?如何避免优先级逆转?如何避免死锁?如何避免临界区竞争?如何编写可重入子程序?如何安排数据的存储(是静态存储还是动态存储?如果是动态存储,是启动时预分配还是使用时再分配)?如何编写简洁高效的中断处理程序?如何编写 BSP 和驱动程序?如何利用编译器对代码进行优化?如何在 C 语言中访问物理存储(或 I/O)地址?如何编写 C 语言可调用的汇编子程序(或汇编程序可调用的 C 语言子程序)?如何保证对异步事件的响应时间要求?如何安排异常处理?如何计算程序对堆栈空间的要求?等等。相信阅读本书对你掌握这些技巧会很有帮助。

随着嵌入式应用的普及,嵌入式实时操作系统也成为越来越热门的话题。那么嵌入式实时操作系统与桌面操作系统的区别在哪里?为什么要使用嵌入式实时操作系统?如果某个应用不需要那么强的实时性,那么还需要嵌入式实时操作系统吗?这些问题看似简单,但它却涉及到了嵌入式实时操作系统的本质。我们必须了解嵌入式实时操作系统的设计理念或设计目标以便回答这些问题。嵌入式实时操作系统与桌面操作系统在设计之初对应用环境有 3 种截然不同的假设。

- 桌面操作系统假定硬件体系结构是标准的而且不变的。如 Windows 是专门针对 PC 设计的, Solaris 是专门为 Sun 工作站设计的。而嵌入式实时操作系统则假定每个应用在硬件上都可能是不同的。因此嵌入式实时操作系统具有天然的易移植性。我们知道,构成计算机的 3 个基本要素是 CPU、存储器和 I/O。嵌入式实时操作系统只依赖于 CPU,而在存储器和 I/O 上的差别是通过板级支持包(BSP)来封装的。嵌入式实时操作系统有一个非常小的可工作的硬件配置,如 VxWorks 操作系统核心只要在硬件上提供一个定时器和一个串口就可工作。桌面操作系统对硬件的要求要复杂得多,如必须有磁盘系统(可能是 Flash 盘),因此桌面操作系统很难做到在 ROM 中直接执行(ROM execution)。这也是我们经常看到桌面操作系统的启动时间(boot up time)通常较长的原因之一。
- 桌面操作系统假定应用程序是经常变化的(如 5 分钟之前在玩游戏而现在正在运行字处理软件),甚至几个各不相干的应用在同时运行(如在编写 C 程序的同时在听 MP3)。也就是说,桌面操作系统是一个通用操作系统,它以不变应万变。它把用户可能用到的所有功能统统包装在一起,不管应用程序怎么变它都能应付。嵌入式应用往往都是专用系统,如一台路由器只要完成路由功能就行了,它不用考虑明天是否可能会有人在上面玩游戏。所以嵌入式实时操作系统都是可裁剪的。在一个具体的应用中,只有那些在该应用中用到的操作系统功能才被放到设备中,而没有用到的功能则被裁剪下来。这就是为什么我们通常会发现嵌入式实时操作系统往往只占用较小的内存(small footprint),这在重视成本的应用中对控制硬件成本是非常有用的。另外,这对于提高设备的可靠性和稳定性也是有帮助的。可靠性的一个基本原则是“越简单越可靠”,把那些用不到的操作系统功能放到设备中,只会增加系统的

复杂度，从而降低可靠性。另外把那些用不到的操作系统功能裁剪出去，也会使得操作系统变得简洁明了，便于程序员把握，从而编写出可靠稳定的应用程序代码。

- 桌面操作系统假定应用程序是不可靠的。当桌面操作系统安装在计算机上售出时，它不知道谁会使用这台计算机，该计算机运行什么应用程序，这些应用程序可能是一个初级程序员刚刚花了一个小时编写的，里面还有很多错误。为了对付这些情况，桌面操作系统把计算机系统资源通过各种手段包得紧紧的，使得应用程序不能访问这些资源，或访问起来会受到各种各样的限制，以此来保证操作系统的健壮性。也就是说，桌面操作系统对应用程序员采取的“态度”是“不信任”。而在嵌入式应用的设备中，计算机系统是专用的，并且设备的功能都是通过应用程序来体现的，如路由器的路由功能完全是通过应用程序（路由协议软件）来体现的，如果应用程序垮掉了，那么只有操作系统在那里运行就没有任何意义。所以在嵌入式应用中，不能假定应用程序不可靠，应用程序和操作系统要同样可靠。因此，嵌入式实时操作系统对应用程序员采取的“态度”不是“防着点”或“留一手”，而是“充分信任”。也就是说，嵌入式实时操作系统对应用程序员是充分开放的。仅仅“开放”还是不够的，如果操作系统过于复杂（如由几千万行代码组成），即使有完整的源代码和文档，也依然无法把握整个系统。所以嵌入式实时操作系统，如 VxWorks，设计得非常简洁明了。这也就是为什么有些程序员说：“当用桌面操作系统控制一个设备时，总有一种‘隔靴搔痒’的感觉；而改用嵌入式实时操作系统后，却有一种‘指哪儿打哪儿’的感觉”。建立在这样一个能够充分把握的操作系统之上编写出来的应用程序一定更为可靠。这里的讨论实际上已经切入到实时操作系统的本质。有不少人认为实时操作系统不就是运行得快一点吗，我的应用不需要运行那么快，所以不需要实时操作系统。实际上学术界早已公认定实时操作系统的本质不是“快”，而是可预测性(predictability)。实时操作系统的行为是确定的(deterministic)。这种确定性是编写可靠软件的基础。你可能不需要一个运行得那么快的操作系统，但你一定需要一个便于写出高可靠性应用程序的操作系统。

在序中用这么长的篇幅来讨论桌面操作系统和嵌入式实时操作系统的区别，是希望那些对桌面操作系统比较了解的工程师，也能够对嵌入式实时操作系统感兴趣，并花些时间阅读本书。本书的作者们在嵌入式实时多任务商用产品的开发上有很丰富的经验，他们在本书中总结的实际开发经验有很高的参考价值，对刚刚从 PC 软件开发转到嵌入式实时多任务软件开发的工程师来说，很有帮助，建议仔细阅读。WindRiver 非常希望看到有更多的有经验的嵌入式软件工程师能够把他们的经验与大家分享。

WindRiver 愿意与更多的公司和个人一起提升中国的嵌入式系统开发水平。

美国 WindRiver 公司 中国首席代表

韩青

2004 年 2 月

前 言

嵌入式设备在日常生活中无处不在，诸如人们日常生活中的手机、MP3、数码相机、洗衣机、彩电、冰箱、电话机、CD 机、游戏机……里面可能都有嵌入式软件。由于嵌入式系统所特有的高效性、自动性、智能化、低能耗、高可靠性和高品质特点，所以除了在日常生活中的广泛应用外，在工业、农业和商业活动中，嵌入式系统同样也得到了越来越广泛的应用，嵌入式设备的应用数量是通用 PC 的 10 倍以上，嵌入式系统是后 PC 时代的计算机应用的重要特征。2002 年我国嵌入式软件市场销售额达到 118.56 亿元，随着移动通信、信息家电和工业控制等领域的不断成熟，我国嵌入式软件产业将迎来新一轮的快速增长，并将成为我国软件企业发展的重要领域。

随着嵌入式系统应用范围的日益广泛和深入，从事嵌入式系统研发的人员数量也在迅速增加。但他们中的许多人都是从事嵌入式软件开发工作不久，缺乏经验，因而如何帮助这些新手能够尽快地掌握嵌入式系统开发的基本方法，提高他们对错误的诊断能力，掌握编写一个稳定、高效、错误很少的嵌入式系统的基本能力，就成了一个急待解决的问题。早在 6、7 年前，本书作者在与公司的新进员工交谈时，常常听到他们谈及自己进入嵌入式实时多任务软件开发领域时，因为找不到一本能系统地介绍该类软件开发基本要点的参考书籍，走了不少弯路，吃了不少苦头。可是很少听到他们抱怨说，某个开发平台不会用，因为每个商用开发平台都有十分详细的文档和参考资料，同时经过试用期的一系列公司培训，开发平台的使用问题基本都解决了。所以尽管嵌入式设备的使用数量是通用 PC 的 10 倍以上，但嵌入式软件开发的参考书数量却不到通用 PC 软件开发参考书数量的 1/10，而且在这样有限的嵌入式软件开发参考书中，要么是着重于原理表述的理论书籍，要么是介绍特定的硬件平台、操作系统、编程语言、编译器或软件开发工具的应用指导。很少有深入探讨或介绍嵌入式实时多任务软件开发所面临的一些基本问题（比如如何在实际应用中编写稳定、高效、错误少和易维护的嵌入式多任务实时系统？或者如何在实际应用中避免编写一个不稳定、低效、错误多和难维护的嵌入式实时多任务系统？）的参考书籍。实时多任务所产生的问题在任何硬件平台、操作系统、编程语言和嵌入式实时多任务应用系统中都会遇到，是不可避免的重要问题，而且实时多任务产生的问题也是该类开发中工作最难解决的一类问题，也是影响嵌入式实时多任务应用系统的稳定性、性能和可维护性的根本问题。遗憾的是，6、7 年过去了，情况并没有多少改善。每一个新进从业人员仍然会反映同样的问题，这就是本书作者的著书起因。

本书的作者试图在本书中，对如何解决以上所描述的嵌入式实时多任务软件开发的基本问题，根据自己的实际工作经验，进行了一些基本的解答，同时对嵌入式系统的开发特点，商用嵌入式软件开发的流程和品质控制进行介绍。本文的作者有十年以上 32 位 RISC CPU 嵌入式软件开发经验（IBM PowerPC 系列、Intel I960 系列等），在美国 Wind River 公司的嵌入式实时操作系统 VxWorks 平台上有丰富的开发经验。所开发的拥有完全自主知识产权的嵌入式产品通过了世界一流大厂（RICOH、Konica Minolta、Sharp、Thoshiba 和

Panasonic 等) 的严格质量测试, 产品出口美国、日本和欧洲市场, 数量达几十万台套。本书的作者同时还有多项嵌入式软件发明专利, 并已申请美国专利保护。但由于作者的知识、经验有限, 所以本书提出的解决方法可能有缺陷, 有盲点, 不够深入。但无论如何, 我们坚信本书是有益的尝试, 希望起到抛砖引玉的作用。

本书主要是为那些刚开始(少于3年)从事嵌入式实时多任务软件产品开发的工程师, 或者是刚开始学习“嵌入式软件开发”课程的学生而编写, 期望能帮助他们在嵌入式实时多任务软件开发方面打下比较好的基础, 或者在实际工作中能马上应用, 提升所开发的嵌入式软件产品的品质。对于那些负责嵌入式软件测试或品质保证的工程师, 阅读本书可以了解嵌入式软件开发的原理及常见问题, 对编写测试范例也很有帮助。

本书尝试以一个具体的嵌入式系统(自动售茶机)开发作为参考范例, 围绕这个简化的嵌入式系统模型(自动售茶机), 以5个章节的篇幅, 对嵌入式系统的特点, 商用嵌入式软件开发的流程和品质控制进行了一些针对性介绍, 展示了嵌入式系统及嵌入式软件开发的各个方面, 提出了嵌入式多任务软件开发中的一些常见问题及基本的解决思路。本书的各个章节安排如下。

第1章 嵌入式系统的简介

本章利用标准的商用嵌入式软件开发流程和品质控制概念——可追踪性开发和管理思想, 对一个嵌入式实时多任务系统范例(自动售茶机)的产品规格进行了相对详细的描述, 并且有针对性地对嵌入式软件开发和通用PC软件开发的的不同, 嵌入式软件开发的常见错误等, 进行了介绍, 同时对嵌入式系统的应用范围和后PC时代的嵌入式系统的应用, 也进行了简单分析和预测。

第2章 嵌入式系统的架构

本章有针对性介绍了嵌入式系统的常见架构和开发环境, 并且基于可追踪性开发和管理的概念, 对自动售茶机这样一个实时多任务的嵌入式系统, 详细地规划和描述了它的系统架构和组成。

本章的前两节, 以一个具体的嵌入式系统(自动售茶机)开发设计作为参考范例, 系统地介绍了此类软件开发高层设计的思想和简单的文档模板, 软件开发的系统设计师和项目经理可以结合本公司的实际项目开发情况, 设计出一套合适的产品规格书和系统高层架构设计书。同时, 对本章所提出的一些常见错误和设计误区, 设计人员或管理者可以据此计划出软件开发的技术风险和预防措施。

第3章 嵌入式系统的概念详解

本章比较详细地介绍了嵌入式系统的一些重要概念, 诸如实时嵌入式操作系统(RTOS), 嵌入式系统初始化、任务、信号灯、消息队列、设备驱动程序和接口等。仔细阅读本章对后续的嵌入式软件开发非常有益。

第4章 嵌入式软件开发的编程风格

“编程风格”往往并不为国内大多数开发人员所注意, 而这一点却正是软件产品品质和

成本控制的关键之一。本章重点介绍了一些嵌入式软件的编程风格，诸如语言的选择、模块和文件组织、函数和变量的命名规则、程序的书写规则——注释、宏定义和缩排等，供读者参考。同时对可移植性的编程设计，也进行了简单讲解和实例介绍。

第5章 嵌入式软件实例开发详解

这是本书篇幅最长的一章，也是本书的重点。本章以大量的实例，详细地介绍了嵌入式多任务软件开发中会遇到的各种基本问题。例如：任务的划分、任务优先级的定义、任务的并行、任务之间的同步和互斥、中断程序设计、多任务环境下全局变量的保护、多任务环境下公共库函数的编写、设备驱动程序的编写和实时应用程序的编写等。希望读者仔细阅读和思考作者提出的经验法则和思考题。

本书特色

- 语言描述简单明了

没有高深的理论，没有难懂的专业术语，大量使用自然语言和日常生活中的例子来说明作者的观点，图表丰富，程序注释丰富。

- 覆盖嵌入式多任务软件开发中的基本问题

所有例子来源于实际开发中遇到的有一定普遍性的问题，对其进行抽象/简化。同时对所提出的解决方案也归纳了一些经验原则，力求有一定的深度，而且可以马上用到实际工作中。

- 重点（实时多任务）突出

本书以大半篇幅对实时（Real-Time）多任务（Multi-Tasking）的概念、设计和实现，做了非常详尽深入的阐述，这在现有的嵌入式软件开发的书籍中是十分少见的。

- 使用单一典型例子贯穿始终

本书从头到尾都以同一个典型的嵌入式多任务系统（自动售茶机）作为例子，使读者容易理解，让读者可以看到一个嵌入式多任务系统开发的整体画面（有机的整体），而不是一些片断。

- 丰富的思考题

每个章节之后，都列出了思考题（有一定难度和深度），以便读者做更进一步的思考和分析，同时列出了许多相关的参考书籍，以便读者进行更深一步的研究。

阅读本书，希望读者最好具有标准 C 语言基础，因为本书所有涉及嵌入式软件开发的程序范例，都是以类似于标准 C 语言方式书写的。即使读者不懂标准 C 语言，基于所有程序范例所附带的丰富注释和图表，也可基本了解本书的意思。

本书比较适合基于嵌入式实时多任务操作系统的嵌入式产品的开发，如果读者所开发的嵌入式产品，并不需要操作系统，只需一个主控程序再加上一些时钟中断和其他中断控制，那么本书的内容不太适合这类开发的需求。另一方面，如果读者关心的是非实时性的应用，对实时性的要求并不强烈或严格，而且开发中使用的嵌入式操作系统不是实时操作系统（比如，嵌入式非实时 Linux），那么本书的第 3 章所阐述的概念和第 5 章所介绍的实现方法仅供参考，因为其中有些描述可能不太合适，比如：任务的概念、优先级的概念、临界区保护的实现和消息队列/信号灯的具体使用等，这些描述可能需要修正。其次，本书

所叙述的多任务是指在单个 CPU 上运行的多任务，并不适用于运行在多个 CPU 上的多任务。

致谢

本书在编写过程中得到了清华大学计算机系张素琴教授、唐龙教授、陶品博士以及美国风河公司北京办事处首席代表韩青先生的大力支持和帮助，在此表示感谢！

技术支持

为了更好地解答有关该书所介绍的技术问题，或进行技术探讨，你可以填妥下表所列的资料，以 E-mail 的方式邮寄给我们，地址为：chenfeng@chief-win.com，我们将会在三周内给予答复，如需尽快答复，请标注或说明，我们将尽可能在一周内答复。

姓名：	性别：	年龄：
职业：	E-mail：	
请问您购买本书的原因(可复选)	1. 工作需要 2. 兴趣 3. 培训教材 4. 其他	
请问您从何处知道这本书	1. 书店 2. 网络 3. 培训班 4. 其他	
您觉得本书的内容如何	1. 很好 2. 好 3. 一般 4. 需加强	
读完本书后，您的感觉(可复选)	1. 值得推荐给别人 2. 物超所值 3. 美中不足 4. 没感觉 5. 不值得	
请问您对本书的技术意见		
您希望技术上有什么改善或补充		
您愿意得到付费的嵌入软件开发培训吗？ 1. 是 2. 否		
如果您愿意，您希望得到什么样的培训？ 1. 开发工具 2. RTOS 使用 3. 实时多任务 4. 应用系统 5. 其他		
如果您愿意，您希望支付多少培训费？ 1. 低于 1000 元 2. 1000~2000 元 3. 2000 元以上 4. 其他		

晨 风

2004 年 2 月于北京

目 录

第 1 章	嵌入式系统的简介	1
1.1	嵌入式软件开发实例的规格	1
1.1.1	自动售茶机功能描述	2
1.1.2	自动售茶机配置	3
1.1.3	自动售茶机操作流程	5
1.2	嵌入式软件开发和其他软件开发的区别	8
1.3	嵌入式软件开发的应用范围	12
1.4	嵌入式多任务软件开发常见错误归类及介绍	15
第 2 章	嵌入式系统的架构	18
2.1	嵌入式系统开发实例(自动售茶机)的系统架构	18
2.1.1	软件系统	18
2.1.2	硬件系统	19
2.1.3	微处理器 (IBM PowerPC 405EP)	20
2.2	嵌入式系统的常见架构	22
2.2.1	嵌入式系统的软件	22
2.2.2	嵌入式系统的硬件	25
2.3	嵌入式微处理器	25
2.3.1	PowerPC 系列	25
2.3.2	MIPS 系列	26
2.3.3	ARM 系列	28
2.3.4	DSP 简介	31
2.3.5	TI DSP 系列	31
2.4	嵌入式软件的开发环境简介	33
第 3 章	嵌入式系统的概念详解	37
3.1	嵌入式实时操作系统	37
3.1.1	引言	37
3.1.2	与通用操作系统的区别	37
3.1.3	VxWorks 简介	39
3.1.4	Nucleus 嵌入式操作系统简介	42
3.1.5	嵌入式 Linux 简介	44
3.1.6	嵌入式操作系统功能	48

3.1.7	嵌入式操作系统中重要多任务相关函数介绍	50
3.2	系统初始化	53
3.2.1	系统初始化的目的	53
3.2.2	系统初始化的范例	56
3.2.3	自动售茶机的初始化	57
3.3	任务	57
3.3.1	什么是任务	57
3.3.2	任务与函数的区别及任务的例子	58
3.3.3	一个 CPU 如何同时处理多个任务	59
3.3.4	多任务执行顺序优先级裁决	59
3.3.5	任务的状态	60
3.3.6	任务的划分	64
3.4	中断	64
3.4.1	什么是中断	64
3.4.2	中断向量表	65
3.4.3	中断处理过程	65
3.4.4	中断优先级	65
3.4.5	中断操作	67
3.4.6	中断和任务的区别	69
3.5	消息队列	69
3.5.1	什么是消息队列	69
3.5.2	消息队列的状态	70
3.5.3	消息队列的工作方式	71
3.6	信号灯	77
3.6.1	什么是信号灯	77
3.6.2	信号灯的用途	78
3.6.3	信号灯的基本操作实例	78
3.6.4	其他信号灯类型 (VxWorks)	79
3.6.5	使用信号灯需要注意的问题	80
3.6.6	信号灯的实现	80
3.7	高速缓存 (cache) 介绍	82
3.7.1	高速缓存的模式 (write-back 模式和 write-through 模式)	83
3.7.2	缓存的操作 (Flush 和 Invalidate)	84
第 4 章	嵌入式软件开发的编程风格	85
4.1	嵌入式系统编程语言	85
4.2	模块和文件的组织	87
4.3	C 源文件和 H 头文件的组织	88
4.4	命名规则	91

4.5	程序的书写格式	93
4.6	可移植性编程	98
第 5 章	嵌入式软件实例开发详解	102
5.1	嵌入式多任务系统实例——自动售茶机	102
5.1.1	嵌入式系统的设计	102
5.1.2	自动售茶机系统任务和中断的划分	102
5.1.3	自动售茶机系统任务和中断的优先级设定	103
5.1.4	自动售茶机系统任务和中断之间的信息交换设计	103
5.1.5	自动售茶机系统初始化代码	104
5.1.6	自动售茶机系统中断代码	106
5.1.7	自动售茶机系统任务代码	106
5.2	任务的优先级设计	109
5.2.1	任务优先级设定原则	109
5.2.2	如何设计高优先级的任务	113
5.3	多任务之间的同步/互斥	115
5.3.1	信号灯和任务间同步	115
5.3.2	消息队列和任务间同步	116
5.3.3	信号灯和任务间互斥	116
5.3.4	信号灯操作的异常处理	118
5.3.5	用消息队列实现任务间互斥	120
5.4	多任务间的信息传递	121
5.4.1	在中断与任务之间传递消息	121
5.4.2	任务与任务之间传递消息	125
5.4.3	用信号灯来传递信息	127
5.5	中断处理程序设计	129
5.5.1	中断处理程序的时效性	129
5.5.2	中断处理程序中调用其他函数的限制	130
5.5.3	中断处理程序中屏蔽/允许同类中断	132
5.6	处理速度的提升（任务并行）	133
5.7	多任务环境下全局变量的保护——原子操作	136
5.7.1	错误现象	137
5.7.2	错误原因	138
5.7.3	如何解决原子操作的问题	139
5.8	多任务环境下全局变量的保护（volatile）	144
5.9	多任务环境下公共库函数的编写	147
5.10	出错处理及恢复	152
5.11	设备控制模块	160
5.11.1	引言	160

5.11.2	外部设备寄存器的访问	160
5.11.3	外部设备的寄存器分类	161
5.11.4	CPU 与外部设备之间的通信	162
5.11.5	设备的错误处理	163
5.11.6	设备控制模块的分层设计	164
5.12	实时响应的实现	165
5.12.1	实时系统示例	165
5.12.2	实时时钟	167
5.12.3	实时时钟设备驱动程序的实现示例	167
5.12.4	实时应用系统的实现示例	173
参考文献		177

第 1 章 嵌入式系统的简介

1.1 嵌入式软件开发实例的规格

大多数软件开发人员往往会忽视被要求开发产品的规格 (specification)，他 (她) 们认为这是市场部或产品规划部门的工作，这是一个很严重的误解或者错误观点。事实上，市场部或产品规划部门仅仅提出一个相当简单扼要的产品规格，更详细具体的规格描述或定义应由工程开发部门确定和实现 (前提是得到市场部或产品规划部门的确认 (confirm) 和认可 (approve))。当然，在实际产品开发阶段，由于市场的变化，产品的规格往往需要更新或改变以提升产品的竞争力。所以，如果对开发产品的规格没有十分清晰的了解，产品开发工作 (development) 和产品的发布 (announcement) 将会受到相当严重的影响，同时也会影响到软件开发人员的信心和成长。从图 1.1 软件开发的基本流程来看，对开发产品规格的分析 and 了解，也是软件开发人员必须着重关注的一个起点，也就是常说的——首先是做对的事 (do right thing)，然后才是把事做对 (do thing right)。

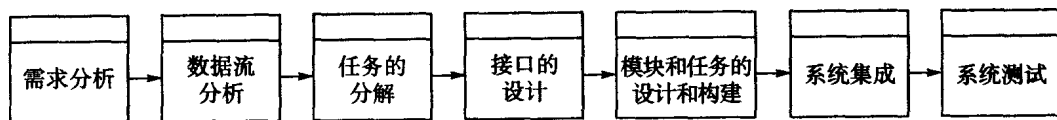


图 1.1 软件开发的基本流程

嵌入式软件产品特别强调“量身定制”的原则，也就是说基于某一种特殊用途的需求，因而特别针对该项要求而设计开发出与其他产品完全不同的系统，这也就是所谓的个性化 (personality) 和定制化 (customization)。特别对系统设计和开发部门而言，很难做到不经过重大改动而把其他系统的设计和套用到另外的嵌入式软件产品上，主要是嵌入式软件产品的硬件平台 (hardware platform) 非常多样化和个性化，同时基于成本 (cost & effort) 的考虑，应尽量简化不必要的软硬件的需求和设计，因此每一个嵌入式软件产品都有其特殊性，这也就更增加了嵌入式软件产品的设计和开发的困难和风险，特别是从一个软硬件平台移植 (porting) 到另一种软硬件平台。在本书作者十多年嵌入式软件产品的开发和管理实践中，类似的移植工作需耗时半年或一年以上，人力的投入十分巨大——平均 200 人月以上。如何有效防范和管理嵌入式软件产品设计和开发的风险 (risks)，保证高品质 (high-quality) 和快速地响应市场需求 (time-to-marketing)，不仅仅是管理人员的职责，工程部门的设计和开发人员同样承担着十分重大的义务和责任。

基于本书作者十多年商用嵌入式软件产品开发和管理的经验，认为“可追踪性开发和管理 (trace-ability) 方法”可以帮助软件开发商重复地 (repeatable) 利用人力资源和技术资源，从而达成高品质 (high-quality) 低成本 (low-cost) 快速响应市场需求 (time-to-marketing)

的目的,对工程部门而言,该方法也提供了对嵌入式软件开发计划(plan)、设计(design)、测试(testing)和品质保证(quality assurance)进行有效检查和认证的技术支持。有关“可追踪性开发和管理(trace-ability)方法”的介绍,我们会在后续的“嵌入式实时多任务软件开发——系统设计”、“嵌入式实时多任务软件开发——项目开发的技术管理”和“嵌入式实时多任务软件开发——工程测试和产品测试”中加以详细介绍,本书仅仅进行概念的引入。

自动售货机只是嵌入式系统应用中的一部分,在国外,可以说是遍布大街小巷、机场饭店,几乎出现在所有的公共场所,给使用者提供了极大的便利。本书引入的这个自动售茶机作为嵌入式软件开发的例子,同时基于教学和培训的目的,我们选用了32位CPU作为它的控制处理器(大多数嵌入式系统目前仍采用低端的8位CPU作为控制处理器,32位CPU仅仅应用于少量的高端嵌入式系统,诸如手机、PDA、高端的工业用数字控制系统CNC以及中高端的打印机和多功能机等),读者可以通过本书所叙述的实际开发设计要点,来了解什么是嵌入式软件、嵌入式软件的技术框架以及嵌入式软件的技术特点,并应用到实际的开发工作中。

1.1.1 自动售茶机功能描述

自动售茶机的功能和操作流程看起来十分简单——使用者可以通过设在机器前端的投币口投入硬币来购买热茶,当使用者投入硬币后,自动售茶机就开始自动运转,机器将自动从放纸杯的装置中拿纸杯,然后放入茶包,接着注入开水,最后通过机器的传输装置将泡好的热茶送到取物口。在整个服务过程中,服务进程始终显示于机器前端的显示屏上。

本书为了使读者能够更好、更容易地理解该系统,我们特别做了一些功能和操作流程的简化(删除了诸如多种茶型选择、多种钱币输入、水温选择、网络服务和管理等高级功能),只保留了该系统的核心模块。

自动售茶机产品规格(product specification)如表1.1所示。

表 1.1 自动售茶机产品规格

自动售茶机产品规格书 (product specification)		
项目 (No.)	分类 (category)	详细描述 (detail specification)
1	基本服务	该机器只接受一元新版硬币
2		客户可同时投入10个一元新版硬币
3		只有一种绿茶可供选择
4		只有一种200毫升容量的茶杯
5		每次服务仅提供一杯茶水(在该杯茶水被客户取出之前,不会提供下一杯茶水)
6		如果服务出错,客户投入的钱币将通过“退币口”返还
7		如果投入的钱币不符合要求,客户投入的物品将通过“退币口”返还