

Mentor Graphics公司资深嵌入式软件专家历时多年呕血之作。

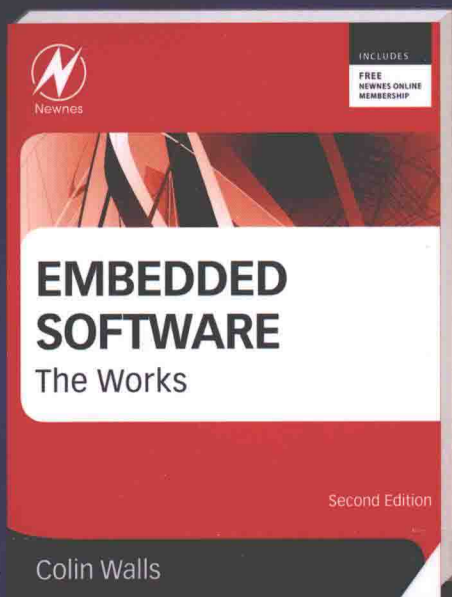
全景式展现了一幅嵌入式软件开发的路线图。

涉及嵌入式软件开发的所有环节，包括语言、方法、工具、RTOS、网络和USB等。

覆盖了USB3.0、多核设计、Linux和Android等嵌入式开发的前沿技术。



电子与嵌入式系统
设计译丛



嵌入式软件 开发精解

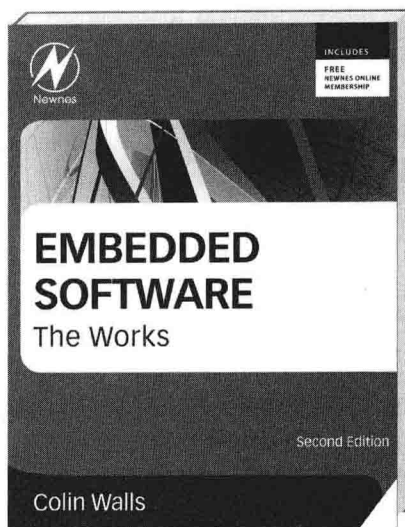
[美] Colin Walls 著 何小庆 张浩中 何灵渊 译



机械工业出版社
China Machine Press



电子与嵌入式系统
设计译丛



Embedded Software
The Works Second Edition

嵌入式软件 开发精解

[美] Colin Walls 著 何小庆 张浩中 何灵渊 译



机械工业出版社
China Machine Press

图书在版编目 (CIP) 数据

嵌入式软件开发精解 / (美) 瓦尔斯 (Walls, C.) 著; 何小庆等译. — 北京: 机械工业出版社, 2014.1
(电子与嵌入式系统设计译丛)

书名原文: Embedded Software: The Works, Second Edition

ISBN 978-7-111-44952-2

I. 嵌… II. ①瓦… ②何… III. 软件开发 IV. TP311.52

中国版本图书馆 CIP 数据核字 (2014) 第 008032 号

版权所有·侵权必究

封底无防伪标均为盗版

本书法律顾问: 北京大成律师事务所 韩光 / 邹晓东

本书版权登记号: 图字: 01-2013-4253

Embedded Software: The Works, Second Edition

Colin Walls

ISBN 978-0-12-415822-1

Copyright © 2012 by Mentor Graphics. Published by Elsevier Inc. All rights reserved.

Authorized Simplified Chinese translation edition published by the Proprietor.

Copyright © 2014 by Elsevier (Singapore) Pte Ltd. All rights reserved.

Printed in China by China Machine Press under special arrangement with Elsevier (Singapore) Pte Ltd. This edition is authorized for sale in China only, excluding Hong Kong SAR, Macau SAR and Taiwan. Unauthorized export of this edition is a violation of the Copyright Act. Violation of this Law is subject to Civil and Criminal Penalties.

本书简体中文版由 Elsevier (Singapore) Pte Ltd. 授权机械工业出版社在中国大陆境内独家出版和发行。本版仅限在中国境内 (不包括香港特别行政区、澳门特别行政区及台湾地区) 出版及标价销售。未经许可之出口, 视为违反著作权法, 将受法律之制裁。

本书封底贴有 Elsevier 防伪标签, 无标签者不得销售。

本书面向想要了解、使用嵌入式技术的读者, 全面介绍了嵌入式系统设计与开发技术, 秉承理论与实践相结合的指导思路, 帮助读者快速跨入嵌入式系统开发的门槛。本书围绕嵌入式系统中的软件开发技术展开讨论。从最基本的原理和概念入手, 介绍嵌入式系统、嵌入式操作系统的原理和概念, 并在此基础上, 介绍了嵌入式软件开发人员必须了解的知识, 包括基本硬件原理、嵌入式操作系统定制和移植以及嵌入式系统集成和软件调试。本着精简理论, 注重实践的原则, 本书的许多章节都附有详细的源代码及分析, 并且搭建了基于模拟器的实验环境, 有助于读者提高实践动手能力。

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: 秦 健

印刷: 北京市荣盛彩色印刷有限公司

2014 年 3 月第 1 版第 1 次印刷

186mm × 240mm • 21.5 印张

标准书号: ISBN 978-7-111-44952-2

定 价: 79.00 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: hzsj@hzbook.com

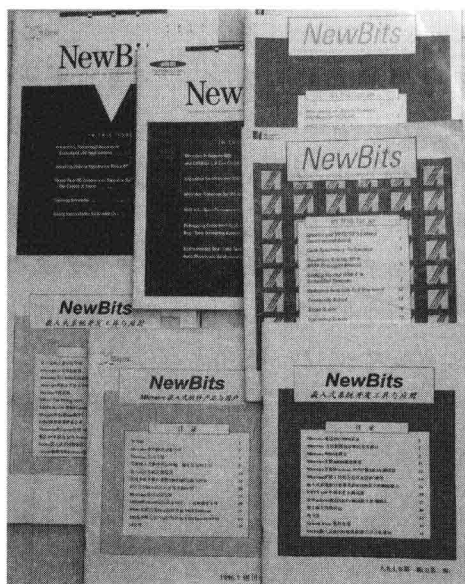
译者序

当我从编辑手里拿到这本书的时候，我一眼就看到作者 Colin Walls。虽然当时我的另外一本译作《解读云计算》还没有完成，但是我当即决定接下本书的翻译工作。

认识 Colin 可以追溯到 1994 年年底，那个时候我创办的公司开始与美国 Microtec Research 合作，于是我经常参加公司技术培训和全球的销售会议。来自英国分公司的 Colin 是资深 FAE，他是个个子高高，讲话很慢的人。他的讲演很清楚，在内容上总是能够谈及关于嵌入式软件工具的很深入的问题。更多地了解 Colin 是通过本书多次提到的《NewBits》杂志，在上面我经常能看到他写的文章，比如《使用背景调试模式》就对我们当时基于 Motorola 68360 系列芯片软件的开发工作非常有帮助。

Colin 在本书第 1 版前言中就已经提到，本书的许多内容来自《NewBits》。这本杂志原来是 Microtec Research 的一份季刊，从当初简单的公司新闻刊物变成了一份技术杂志。即使在公司被 Mentor Graphic 收购之后，该刊仍然坚持出版了一段时间。在 1996～1997 年期间，我们在中国出版了 4 期中文版（《嵌入式软件产品与应用》）。中文版除了翻译部分《NewBits》的技术文章外，还邀请北京航空航天大学计算机系金惠华教授和成都电子科技大学熊光泽教授，以及他们的学生为杂志撰文。我和我的同事江文瑞也为中文版撰写过技术文章。当然，Colin 的文章总是我们的首选，比如另外一篇《在嵌入式系统中使用 C++ 编程语言》在当时就很受关注。

正如硅谷嵌入式资深人士 Jack Ganssle 介绍的，Colin 是一个老手，本书给大家分享了他丰富经验和实际“如何做”的忠告。通过阅读本书，你会受益匪浅。本书的内容很广泛，覆



盖嵌入式软件设计方法、工具、语言、RTOS、网络和 USB。这样广泛的题材得益于 Colin 所工作过的 Microtec Research 和 Mentor Graphic 嵌入式事业部，这两家公司都具有嵌入式软件开发完整的产品线。比如 Microtec Research 的 Microtec 嵌入式 C/C++ 编译器、XRAY 调试器和 VRTX RTOS 都是在历史上极具影响力的技术和产品。

除了 Colin 本人的文章，书中引用了 Jim Ready 关于实时内核的文章，Jim 是著名的 VRTX RTOS 的作者，也是提供商业 RTOS 的第一人，曾经创办过 Ready System 和后来的嵌入式 Linux 公司——Montavista。书中还使用了 Neil Henderson 关于嵌入式网络协议的文章，他曾经是 Accelerated Technology 的 CEO，主持过著名的开源项目 Nucleus RTOS 的开发。这些重量级人物的文章为本书增色不少。

本书的第 1 版于 2006 年出版，第 2 版则于 2012 年出版。6 年时间嵌入式软件发生了很大的变化，作者也相应进行了更新，比如书中增加了 USB 3.0 的知识，代码示例也从以前基于 68K 芯片转为 PowerPC。更重要的是第 2 版新增加了多核设计和开源的嵌入式软件这两章，它们正好体现了嵌入式系统处理器技术和软件技术最新的发展，即在过去 6 年间，嵌入式硬件平台从使用单片的嵌入式处理器发展成多核的、SoC 的嵌入式处理器（包括 MCU），嵌入式软件平台也从基于 RTOS 开发转变到基于开源和面向应用的平台开发，比如基于 Linux 和 Android 平台。

本书的第 1、2、6、7 章是由张皓中翻译，第 3、4、5 章由何灵渊翻译，第 8、9、10 章由何小庆翻译，全书由何小庆统稿和审校。在翻译过程中，我们借鉴了由沈建华老师主持翻译的本书第 1 版——《嵌入式软件概论》（北京航空航天大学出版社），这本书对于我们更好地理解原文起了重要的作用。沈老师主持的翻译工作非常出色，不过我们也按照自己的理解对一些细节的翻译做了修正。我还要感谢本书的策划编辑张国强先生，他申请到的图片和电子稿，为我们工作在不同地区的翻译团队成员带来了极大的方便，他对工作的热情和执着也着实感动了我。

最后，希望本书的读者能从本书的阅读中受益。我们的翻译如有不足和纰漏之处，还请读者指出。

何小庆

序

你期待着什么？完美？

Jack Ganssle 的一些想法……

为什么许多固件项目都不能按时完成，而且充满着各种各样的 bug？我们可以从很多方面解释这个问题，比如软件太过复杂等。不过，我相信做不好并不是我们的错，因为真正做到完全不出错的一定是超人类，很显然我们都不是。

原始人并不追求每次捕猎都能得手，他们只需要填饱肚子；农民不会期待全部的种子都发芽，总有一些会浪费掉；商人都希望吸引消费者，但他们不可能让所有人都满意。

得到全 A 的孩子会让爸妈惊喜异常，但是 A 也不代表着完美，90% 的正确率就已经是 A 了。在生活中，我们并不需要完美，往往达到 A 的标准就已经足够了。

但是在软件行业，90% 的正确率已经是灾难了，做出的产品根本不能用。即使 99.9% 的正确率，产品也是垃圾一个，因为每十万行代码中就会有 100 个错误。很明显，这样依然不够。也就是说，虽然我们不是完人，但是我们写出的软件却需要无限趋近完美。

如果系统只有 100 行，每个人都可以达到完美。但是对于大规模的软件而言，达到完美需要极大的投入。就像牛仔需要不断地将掉队的牛赶回牛群一样，程序员也需要看护好程序中的每一个字节。但是随着牛群的壮大，牛仔会越来越忙，程序员也是如此。

那么解决方案是什么，究竟有解决方案吗？固件需要做到多完美呢？我们对于完美的执著真的是白费力气么？

复杂系统是这个世界的原生事物。早期的半导体收音机中最多有五六个元件，电视机中有 15 ~ 20 个真空管，两者使用的晶体管数目是差不多的。等到了 20 世纪 40 年代，由于 ENIAC 电脑中有 18 000 个真空管，工程师们不得不用购物车装满元件，不断进行维护工作。而一个有 25 年历史 Z80 芯片，就有 ENIAC 1/4 个晶体管数量，其管芯（Die）面积也只有 ENIAC 使用的单个电阻一样大。

如今，单独一个奔腾 IV 处理器之中有 4500 万个晶体管，一个比较大的内存芯片则有大

概 2.5 亿个晶体管。根据 Intel 的预测，在几年之内他们制造的处理器将能够搭载 10 亿个晶体管。即使是最简单的嵌入式系统，比如电子贺卡，其上也会有数千个活动的单元。

软件的发展更加快速，特别是对于嵌入式系统而言。1975 年，程序是在纸带和磁带上存储的，控制台则是电传打字机，所以 1 万行的汇编代码已经是非常庞大的工程了。而在今天，1 万行 C 代码（大约等同于四五万行汇编代码）已经是很小的程序了。另一个例子是我们的手机。手机的开发时间很短，尺寸小和功耗低，但其中却可能有超过 500 万行的 C 和 C++ 代码。

存储器占用是另外一个衡量软件规模的指标。1975 年 256 字节（是的，你没有看错）的 EPROM 可以存储 4k 大的程序空间，需要 16 片存储芯片，即使是一个很小的嵌入式系统的造价也不菲。今天呢？128k 闪存对于一个小型的应用而言也微不足道。处理器之所以会从 8 位发展到 16 位，再到 32 位，更多的是因为对地址空间需求的增加，而不是处理速度。

在 20 世纪 70 年代末期，Seagate 推出了首款 Winchester 硬盘，它的容量是 10Mb，重量有 4.5 公斤，单价则是达到了 1500 美元。在那时，5MB 就已经远远超过人们的需求了。如今，我们可以轻松地把 20GB 的硬盘买来装在口袋里，并在眨眼间就把它装满。[⊖]

所以说，我们的系统在规模和复杂性上都已经达到了空前的水平。不过，我们真的应付得来么？即使是很简单的系统，我们也难以做到完美，而大型的系统则永远会有缺陷。随着软件规模的增长，其中组件间的关系也会变得十分复杂。这样的复杂性更多的时候是工程的必需，而不是因为设计得不好。

硬件同样很难达到完美。即便是成熟的处理器，也会有一个错误列表。奔腾系列的一个鲜为人知的除法错误只是冰山一角。像奔腾 III 的错误列表（后来改名规格更新表）之中就记录了 83 个问题，Motorola（摩托罗拉）MPC555 则有接近 100 个问题。

现在的嵌入式系统有多可靠？没有人能够回答，也没有人针对这个问题进行研究。事实是，我们手头有不少数据，其中许多都不太乐观。

当年，火星探路者号探测器项目取得了空前的成功，除了在下降阶段出现了一个严重的软件错误。一个在地球上被认为是小瑕疵的优先级反转问题引发了更多的错误。万幸的是，因为看门狗程序设计得很好，探测器的任务最终没有失败。这个例子生动地告诉我们，额外的外部软硬件能帮助我们有效地应对预料之外的软件错误。

硬件和软件的规模和复杂性今非昔比。早先的微处理器中，探路者号上的优先级错误根本不可能出现，因为那时的应用根本就不需要使用 RTOS。今天，大多数嵌入式系统都运行着操作系统，这些设备都可能出现类似探路者号的问题。

⊖ 今天 8GB 的 U 盘和 500GB 的硬盘已经随处可见。——译者注

作为开发者，面对爆炸性增长的系统复杂性，我们应该怎么办？我们肯定需要不断学习新知识和巩固旧知识。从 C 语言的基础到 UML，我们需要不断为自己充电，像你手中的这本书就是非常好的学习资料。Colin 是位资深嵌入式专家，他会给你许多关于实际操作的建议。

我们确实可以从经验中学习，但是积累知识毕竟还是更为有效的学习方法。比学习知识更有效的方法则是从大师那里汲取智慧。花一点时间阅读本书，你就能从大师那里得到重要的启示。最终，有了独特的视角和强大的工具，你将能够解决更加复杂的问题。

在追求完美的过程中，我们也许会感到困惑，但是有智慧的开发者会孜孜不倦地寻找通向完美的道路。

前 言

自本书第 1 版出版之后，世界在前进，许多事情变了，也有许多事情没变。随着处理器和技术的普及，嵌入式软件领域也得到了发展。嵌入式软件工程师的事件也在变化，Elsevier 的工程师们建议我升级本书的第 1 版，而我也欣然同意。

本书第 2 版的许多章节中都新增了内容，而且增加了关于开源软件和多核的两个章节。除此之外，我还去掉了一些不太相关的内容。

我保留了第 1 版的前言，不过我需要在这里对下面的贡献者表示诚挚的谢意：Vlad Buzov、Dan Driscoll、Christopher Hallinan、Waqar Humayan、Geoff Kendall、Russel Klein、John Lehmann、Shabtay Matalon、Mark Mitchell。

然后是我个人最近的一些情况。尽管我病重的爱人接受了非常好的治疗，她还是没能坚持得更久，她于 2006 年 6 月离开了我们。积极治疗为她争取了两年的时间，也为我们带来了更多宝贵的回忆。现在，我和我的女儿们生活在一起。我依然在 Mentor Graphics 从事与嵌入式软件相关的工作。

我想再一次感谢我的上司，感谢他们同意将本书的获利捐赠给 LINC，我相信这笔钱会花在有用的地方。

Colin Walls

2012

第 1 版前言

本书的背景

我不知道这个世界有没有拟定生活计划并且严格执行的人，不过我相信是有的。但是对于我们之中的大多数人而言，即使我们有对未来的憧憬和想法，我们也不得不为身边种种的随机事件疲于奔命。我正是处于这样的状况。我为 **Accelerated Technology** 工作，它是 **Mentor Graphics** 的一个分支，专长是嵌入式软件设计、开发，工具研发和操作系统。多年来，公司几经易手，我也接触到了各式各样的工作。最近我在做一些市场营销，帮助欧洲和北美的人士了解公司的产品和背后的技术。我喜欢我的工作，也喜欢和我的同事们一起工作，我时常称呼自己是一位“专业的爱好者”。有时这项工作也是十分辛苦的，需要花很多的时间在路上（坐飞机和汽车）或者等待某一样东西。很多的时候我都不在家，不过我觉得这是得到这份好工作的代价。但是，后来发生的一件事情改变了一切……

2004 年的仲夏，我的爱人被诊断出患有急性髓细胞白血病，并被收治住院。我需要照顾我的两个女儿，并陪伴我的爱人，因此我的工作生活也就退到了次要的位置。从夏天到年末，我爱人一直在接受放疗。在这期间，我们经历了起起落落，不过效果也很明显。在我撰写本书的时候，她的情况有了很大的好转，每个月会验血检查疗效。

俗话说，患难见真交，此话不假。这段时间之中，我更加确定自己选对了工作的地方。我的上司和同事给予了我莫大的支持，我觉得十分亏欠他们（我会在后面列出他们的名字）。在一段时间内，我肯定不能继续我原来的工作了。我现在没有压力，不过我也在想，我能做些什么有用的，适合我新的生活方式的事情。在很长一段时间内，我一直酝酿着写一本书。我把这个想法告诉了公司管理层，他们认为这是个很好的计划。

本书是从哪里来的

我的上一本书是出版于 1986 年的《**Programming Dedicated Microprocessors**》(Macmillan

Education, 1986), 距现在已经接近 20 年了。那本书是讲述嵌入式软件的, 这个概念在那个时候还不是很常用。写一本书是一种个人实现, 而那书也是我十分得意的收藏之一。写一本书也十分费时。当我写那一本书的时候, 工作并不繁忙, 而且我也没有孩子, 所以抽出时间并不是很难。在那之后, 我时常有写书的新想法, 但却一直没有时间付诸实现。

尽管如此, 这几年我也一直在写作——大量的技术文章、会议论文、培训材料和演讲稿。后来我有一个新想法: 为什么不把这些内容整理成一本书呢? 为了简化这一过程, 我主要在 Accelerated Technology 内部收集资料, 这家公司拥有这些资料的版权。后来确实得到他人贡献的一些文章, 不过这是后话了。

在 Microtec Research 有一份名叫《NewBits》的季刊, 这份刊物从当初简单的新闻刊物变成了一个技术杂志。在 20 世纪 90 年代早期, 我是这本书的贡献者之一。在 Mentor Graphics 收购 Microtec 之后,《NewBits》继续出版了一段时间, 但最终还是停刊了。最近, 公司收购了 Accelerated Technology, 并重整了旗下的嵌入式软件团队。在这之后, 我们决定继续《NewBits》的出版, 而杂志也越办越好。我收集了《NewBits》的每一期, 把全部文章都放在手边。我很快确定了数篇适合收录的文章, 其中不少来自于我自己, 也有一些出自其他作者之手。其他的材料包括一些白皮书和另一份 Accelerated Technology 的刊物《Nucleus Reactor》。

我尽可能地减少对原文的修改。一些文章基本上完全不需要改动, 其他的一些要么需要更新, 要么需要去掉和产品相关的信息。每篇文章都通过一个简要的说明来介绍其来源和内容。

本书的内容

在选择本书的原材料时, 我的目标是确保每一篇文章都与现今的嵌入式软件开发流程和技术相关。当然了, 许多文章是从历史的角度阐述的, 不过这并不意味着它们就会被收录。有些文章涉及的技术没有流行起来, 或者已经被新的技术所替代, 这导致它们被排除在本书之外。这些过时的主题也许会在我撰写《A History of Embedded Software》一书时提到。我计划在嵌入式软件出现 50 年之际出版那本书(也就是 2020 年, Intel 4004 发布 50 周年纪念的时候)。

本书中包含的文章涵盖了嵌入式软件的方方面面: 设计、开发、管理、调试、授权和重用。

本书的目标读者

如果你对嵌入式软件感兴趣, 本书一定会有适合你的内容。本书中的文章涵盖了非常广

泛的内容，无论你是初学者还是老手，都能从中获益。如果你来自传统的软件领域，一些文章会帮助你从更加偏向硬件的角度看待问题；如果你来自硬件领域，你会有机会了解关于软件的一些内容。

如果你从事教学（学术或者商业培训），书中的许多文章会为你的学生提供很好的背景材料。光盘中的内容考虑到了这样的需要。详情请参考前言后面“光盘中的内容”一节。[⊖]

如何使用本书

本书并没有一个“正确”的使用方法。我已经将文章的出现次序进行了优化，适合从前往后读的顺序。同时我尽可能地将文章按章节分类，并添加了文章之间的引介，帮助你找到感兴趣的内容。

致谢

下面这段话看起来很像是奥斯卡获奖感言，但这也是没有办法的事情。

在提及具体人名之前，我想要先说一些笼统的感言。我和 **Accelerated Technology** 以及 **Mentor Graphic** 的许多同事都讨论过这本书，他们给予了我很大的鼓励。他们的热情感染了我，让我受益匪浅。

当我和 **Elsevier** 的人们一起工作的时候，**Carlos Lewis** 是我的编辑，也是本书的项目策划。有的时候，任务会被交给 **Tiffany Gasbarrini**，她一路都在支持着我。和他们的合作一直是非常愉快的。他们的专业精神令人钦佩，他们给我的引导也让我颇为受益。

我一直非常喜爱 **Jack Ganssle** 的作品，无论是他的书，还是他在《**Embedded System Programming**》杂志中的专栏。他给我们带来了非常实用的技术信息，还时常通过幽默的方法引导我们思考。当我策划本书的时候，我向他征询意见，他毫无保留。当他同意为本书撰写“序”的时候，我也感到十分高兴。谢谢他，**Jack**。

我需要感谢我的上司和同事，他们一直在我身边支持我，并在我遇到困难时出手相助。他们帮助我保持理智（也许），让这本书成为可能。**Neil Henderson**、**Robert Day**、**Michelle Hale**、**Gordon Cameron**、**Joakim Hedenstedt** 等人都帮助了我，当然还有其他更多的人。谢谢他们。

我在前面已经提及，本书中的许多材料来自 **Microtec Research** 的技术刊物《**NewBits**》，

⊖ 本书第2版没有附带光盘，而是开设了专门的网站用于提供补充材料。请参阅后面的“网站上的内容”。——译者注

所以我对于所有对此刊物有贡献的人都心存感激。Lucille Woo 是《NewBits》的责任编辑，他将一个新闻刊物发展成了技术杂志。杂志的图像编辑和设计师是 Gianfranco Paolozzi。在 Microtec，许多人都对杂志有着不同程度的贡献，比如 Eugene Castillo、Melanie Gill 和 Rob van Blommestein。《NewBits》在 Accelerated Technology 的“复活”是由 Charity Mason 领导的，他提出了一套行之有效的解决问题的方法。

商业领域和“法律”二字相关的内容都让我感到很不自在。虽然所有与此相关的流程都是必需的，但是我依然要感谢 Mentor Graphics 采购部的 Jodi Charter 等人，他们帮助我处理了合约的法律问题。

Xilinx 的 David Vornholt 和 Altera 的 Bob Garrett 为我提供了关于 FPGA 处理器的信息，并为本书建言献策，我感到十分高兴。

最后，感谢 Archie 的父母，Andrea 和 Barry Byford，他们允许我在后记中用到 Archie 的照片。

贡献者

这本书一半的内容都是我写的，剩下的则由其他作者贡献（也许有的人并不是那么情愿）。他们是：Zeeshan Altaf、Fakhir Ansari、Antonio Bigazzi、Sarah Bigazzi、Paul Carrol、Lily Chang、Robert Day、Michael Eager、Michael Fay、Jack Ganssle、Bob Garrett、Kevin George、Ken Greenberg、Donald Grimes、Larry Hardin、Neil Henderson、C. C. Hung、Meador Inge、Stephen Mellor、Glen Johnson、Pravat Lall、Tammy Leino、Nick Lethaby、Steven Lewis、Alasdar Mullarney、Stephen Olsen、Doug Phillips、Uriah Pollock、James Ready、John Schneider、Robin Smith、Dan Schiro、Richard Vlamynck、David Vornholt、Fu-Hwa Wang，以及 John Wolfe。

我要对他们表示我的谢意，感谢他们的贡献。如果我漏掉了谁，还请接受我的歉意。

封面和封底之间有许多字，这大概就是一本书了。不过，本书中许多文章都包含辅助理解的插图，就像大多数技术出版物中的文章一样。我很喜欢动笔写作，但我不是画家。因此，我要感谢 Chase Matthews 和 Dan Testen，他们为我制作了插图。

一项善举

当我的上司许可我将本书的获益捐赠给慈善机构的时候，我感到非常高兴。自然，我联想到了我的经历，并决定支持一家做实事的机构。我选择了 LINC 基金（白血病和强化化疗：www.lincfund.co.uk），他们的总部设于我爱人接受治疗的医院。

LINC 是一家帮助白血病和相关疾病患者的慈善机构。LINC 筹集到的资金被用作购买仪器和支持 Cheltenham 综合医院肿瘤科的研究。白血病一般事发突然，而且需要长期的治疗，这让很多不那么富裕的家庭陷入了经济困难之中。LINC 基金也会在这些家庭最需要的时候伸出援手。

在此，我要代表他们感谢你的贡献。他们一定会理智地对待捐献的每一分钱。

联系我们

如果你有关于本书的建议和问题，或者有任何关于嵌入式软件的内容想要分享，请联系我，我会十分高兴。电子邮件是联络我的最佳方法：

colin_walls@mentor.com

如果你想要联系其他的贡献者，请发送邮件给我，我会努力帮你联系。

想要获取本书最新的信息，包括更新、勘误和下载，请访问：

www.EmbeddedSoftwareWorks.com。

Colin Walls

2005

网站上的内容

为了增加本书的价值，我们创建了提供补充材料的一个网站：

<http://www.elsevierdirect.com/companions/9780124158221>

下面是关于网站内容的一些引导。

代码示例

许多章节中包含 C 或者 C++ 的代码。为了让你可以直接复制，我们将这些代码放到了网站上。所有代码都经过认真检查，应该都可以构建和执行。代码保存在对应章名的文本文件 (.txt 后缀) 中，总共有 23 个文件。

第 1 章 嵌入式软件

将软件移植到新处理器架构上

第 2 章 设计和开发

嵌入式软件和 UML

第 3 章 编程

为特殊存储器编程

嵌入式系统中的自检

命令行解释器

嵌入式软件应用：交通信号灯

第 4 章 C 语言

中断函数和 ANSI 关键字

针对 RISC 体系结构的优化

从位开始

浮点数在编程中的应用

从不同的角度看待 C 语言

精通结构布局

第 5 章 C++

为什么要从 C 转向 C++

扫清通向 C++ 的障碍

C++ 模板的优势与劣势

C++ 的异常处理

C++ 的代码大小和性能

C++ 中的只写端口

在 C++ 中使用非易失性 RAM

第 6 章 实时性

中断程序

第 7 章 实时操作系统

自己开发的 RTOS 调试解决方案

RTOS 驱动程序开发简介

第 8 章 网络

哪些人需要 Web 服务器

培训材料^①

本书中的许多文章都是面向培训课程和研讨会的，特别是其中很多我的文章。在可能的情况下，我为文章制作了一个幻灯片集（我显然不能为每篇文章都做出幻灯片）。

这些文件是成对出现的。其中一个 **Microsoft PowerPoint** 文件（.ppt 后缀）。想要运行这类文件，你需要拥有一份正版的 **PowerPoint**。在 **PowerPoint** 中你不仅可以查看文件，还可以对文件进行修改和定制。另一个文件是一个 **Adobe PDF** 文件。免费的 **Adobe Reader** 可以查看这些文件。如果打开 **PDF** 文件，它会立即全屏幕显示（不需要按下 **CTRL/L**），这样就可以演示了（按 **Esc** 键结束）。**Adobe Reader** 是很好的演示工具，但它不能修改 **PDF** 文件。

幻灯片集是按照连贯顺序排布的，和相关文章的内容一致。不过，一些幻灯片也可以在其他文章的范畴内起到阐释的目的。总共有 69 对文件，超过 400 页的幻灯片。

第 1 章 嵌入式软件

嵌入式应用的影响

嵌入式系统的存储器

存储器架构

软件是如何影响硬件设计的

将软件移植到新处理器架构上

汽车电子上的嵌入式软件

芯片设计时如何选择 CPU

USB 软件的介绍

走向 USB 3.0 时代

第 2 章 设计和开发

嵌入式系统软件开发的新兴技术

选择开发工具

Eclipse：集成嵌入式开发工具

跨越 RTOS 界限的开发系统

嵌入式软件和 UML

用户界面开发

软件和功耗

第 3 章 编程

为特殊存储器编程

嵌入式系统中的自检

命令行解释器

嵌入式软件应用：交通信号灯

第 4 章 C 语言

共同变量

C 函数的原型

中断函数和 ANSI 关键字

从位开始

浮点数在编程中的应用

从不同的角度看待 C 语言

减少函数调用的开销

精通结构布局

C 语言编程和存储器

C/C++ 中的指针和数组

C/C++ 中的动态存储

① 直到翻译完成的时候，我们查阅上述网站，目前已经有第 1 ~ 8 章的培训资料（一个压缩文件），作者应该还在补充，欢迎读者继续关注。——译者注

第 5 章 C++

从管理的角度看嵌入式系统中的 C++

为什么要从 C 转向 C++

扫清通向 C++ 的障碍

C++ 模板的优势与劣势

C++ 的异常处理

C++ 中的只写端口

第 6 章 实时性

实时系统

嵌入式系统的可视化程序模型

嵌入式系统的事件处理

中断程序

第 7 章 实时操作系统

RTOS 的调试技术

自己开发的 RTOS 调试解决方案

调试：堆栈溢出

何时考虑使用商业化 RTOS

移植 RTOS

RTOS 驱动程序开发简介

调度算法和优先级反转

时间与优先级调度比较

嵌入式文件系统

OSEK：一种 RTOS 标准

第 8 章 网络

Wi-Fi 简介

哪些人需要 Web 服务器

SNMP 介绍

下一代互联网协议：IPv6

DHCP 基础

NAT 详解

PPP 点对点协议：

SSL 介绍

DHCP 调试小技巧

IP 多播

第 9 章 开源软件、嵌入式 Linux 和 Android

嵌入式开发的 GNU 工具链：构建还是购买

嵌入式系统 Linux 简介

Android 架构和开发

垂直市场上的 Android、MeeGo 和嵌入式 Linux

第 10 章 多核操作系统

多核简介

多核：多个操作系统

选择多核系统多操作系统

CPU 与 CPU 之间的通信：MCAPI