

Bruce Perens 开源系列

嵌入式可配置实时操作系统 eCos 软件开发

Embedded Software Development
with eCos

[美] Anthony J. Massa 著
颜若麟 孙晓明 尤伟伟 林巧民 译



北京航空航天大学出版社

Bruce Perens 开源系列

TP316.2

23

嵌入式可配置实时操作系统 eCos 软件开发

Embedded Software Development with eCos

[美] Anthony J. Massa 著

颜若麟 孙晓明 尤伟伟 林巧民 译

北京航空航天大学出版社

Authorized translation from the English language edition, entitled EMBEDDED SOFTWARE DEVELOPMENT WITH ECOS 1st Edition, ISBN: 0130354732 by Massa, Anthony J., published by Pearson Education, Inc, publishing as Prentice Hall PTR, Copyright © 2003.

All rights reserved. No part of this book may be reproduced or transmitted in any form or by any means, electronic or mechanical, including photocopying, recording or by any information storage retrieval system, without permission from Pearson Education, Inc.

CHINESE SIMPLIFIED language edition published by BEIJING UNIVERSITY OF AERONAUTICS AND ASTRONAUTICS PRESS, Copyright © 2006.

本书中文简体字版由 Pearson Education, Inc 授权北京航空航天大学出版社在中华人民共和国境内(不包括香港、澳门、台湾地区)独家出版发行。版权所有。

北京市版权局著作权登记号:图字:01-2005-2949

图书在版编目(CIP)数据

嵌入式可配置实时操作系统 eCos 软件开发/(美)马萨(Massa, A. J.)著;颜若麟等译. —北京:北京航空航天大学出版社, 2006. 5

ISBN 7-81077-732-7

I. 嵌… II. ①马…②颜… III. 实时操作系统-软件开发 IV. TP316.2

中国版本图书馆 CIP 数据核字(2006)第 036365 号

嵌入式可配置实时操作系统 eCos 软件开发 Embedded Software Development with eCos

[美] Anthony J. Massa 著

颜若麟 孙晓明 尤伟伟 林巧民 译

责任编辑 王慕冰 朱胜军

*

北京航空航天大学出版社出版发行

北京市海淀区学院路 37 号(100083) 发行部电话:010-82317024 传真:010-82328026

<http://www.buaapress.com.cn> E-mail: bhp@263.net

涿州市新华印刷有限公司印装 各地书店经销

*

开本: 787 mm × 960 mm 1/16 印张: 22.25 字数: 498 千字

2006 年 5 月第 1 版 2006 年 5 月第 1 次印刷 印数: 5 000 册

ISBN 7-81077-732-7 定价: 39.00 元

译者序

本书由 Anthony J. Massa 先生撰写,该书的英文版自出版以来就得到了很好的评价。译者在一次非常偶然的的机会得到了该书,看了之后觉得写得非常好,就动了将其译为中文的念头;适逢北京航空航天大学出版社引进了该书的版权,进而与该出版社合作,从而得以实现此愿望。

随着国内计算机应用水平的不断提高,尤其是计算机嵌入式应用水平的提高,实时操作系统 RTOS(Real Time Operating System)越来越得到计算机嵌入式应用人员的重视,应用也越来越广泛。出现这种局面,是因为 RTOS 简化了应用系统软件的设计,保证了系统实时性(响应及时性)等。在苛刻的应用环境(如反应迅速的工业应用场合以及航空、航天应用场合等),RTOS 得到了非常广泛的应用,效果也非常好。例如,美国 NASA 自 20 世纪 80 年代末以来,就一直在航天器中采用实时操作系统,从最早的“火星登陆者”到“勇气号”都采用了 RTOS(VxWorks),事实也证明了这是非常成功的。尤其是后者,还对存储器进行了格式化。

译者一直认为,要想应用好 RTOS,必须深入研究 RTOS,而且要研究透,要不就谈不上高水平地应用 RTOS。对于不开放源代码的商业 RTOS 是如此,对于开放源代码的 RTOS 更是如此。要研究 RTOS,最好从开放源代码的 RTOS 开始,因为对于具备基本功能的 RTOS 而言,其原理差别不是很大。

嵌入式可配置操作系统 eCos (Embedded Configurable Operating

System)是开放全部源代码的操作系统,其支持的特性很多,包括了目前通用的各种硬件特性,如 PCI 总线、USB 总线、IPv4、IPv6、各种文件系统以及 GUI 接口等。它具备的这些特性对于一个有志于计算机嵌入式应用人员的“产、学、研”是非常有帮助的,这是其他开源 RTOS 所不具备的(一般的开源 RTOS 只是提供内核,要在实际工程中应用还需要做很多工作,如文件系统、GUI 接口、网络协议栈等,这些都需要自己解决)。

eCos 具有很强的扩展性,这与它采用的结构紧密相关。通过阅读本书后续章节就会明白,eCos 采用组件来扩展其功能,因此任何符合 eCos 组件规范的组件代码都可以对 eCos 进行某方面的扩充。经常登录 eCos 相关网站,可找到很多非常好用的第三方组件。这些组件都是 eCos 爱好者或应用开发人员辛勤劳动的结晶,从另一方面也说明了 eCos 的扩展性和大众对 eCos 的参与。

尤其重要的一点是,可根据需要对 eCos 的代码进行裁剪。这样便可以大大减少不必要的代码,而又不降低其性能,这对于计算机嵌入式应用是很重要的。在这方面,eCos 无疑是很出色的。通过配置工具,可对包含 eCos 内核代码在内的绝大部分代码进行裁剪,从而去掉无关代码,减小代码尺寸。这对于包括商业 RTOS 在内的 RTOS 来说都很难做到这一点。

eCos 的功能非常全面,商业 RTOS 能提供的功能,eCos 基本上都提供了。因此,可以说,任何商业 RTOS 能胜任的,eCos 都能胜任。

eCos 的性能无须质疑,全球众多的应用人员的口碑已经证实了这一点。而 eCos 在各种苛刻环境和条件下的正确运行,更说明了 eCos 非凡的品质。再者,eCos 的发展非常有活力,作为 eCos 东家的 Red Hat 公司实力是不容忽视的,何况还有全球这么多爱好者在不断地发展 eCos。因此,有理由相信,eCos 会越来越好!

译者相信,本书对于那些有志于嵌入式应用研究或正在研究 RTOS 的人员有非常高的参考价值。

在本书翻译过程中得到了北京航空航天大学出版社胡晓柏老师的大力支持。本书的出版与其努力是分不开的,在此表示感谢。本书的序、前言、第 4、5、8、9、10、11、12 章以及所有附录由颜若麟翻译,第 3、6、7 章由尤伟伟翻译,第 1、2、13 章由中科院软件所孙晓明和林巧民翻译。由于时间紧迫,错误在所难免,希望其不至于引起意义上的误解,欢迎批评指正。

译者

2006 年 3 月

作者简介

Anthony J. Massa 在 San Diego 大学取得电气工程 B. S 和 B. A 双学位。他的大部分时间是开发嵌入式软件、设备驱动程序和应用各种 8、16、32 位处理器以及各种各样的 RTOS 平台。Anthony 也有硬件设计的经验，同时还在各种 Windows 操作系统平台上进行驱动程序和应用程序的开发。

Anthony 具有开发各种成功产品的经历，如 PC 卫星接收卡、机顶盒、因特网无线调试解调器、网络广播设备以及电缆调试解调器等。Anthony 在包括 *Doctor Dobb's Journal*、*Embedded Systems Programming*、*Software Development* 和 *EDN* 等在内的顶级出版物上发表了大量的 eCos 方面的文章。

闲暇的时候，Anthony 喜欢与全家呆在家里，或者旅行（Anthony 发现在 San Francisco 的 Buena Vista 过得非常愉快），或者与祖父呆在后院或打高尔夫球。现在 Anthony 手头的项目已经完工了，Anthony 正在寻找他的下一本书的经历——下一本书或许是小说！

序

至 1997 年,有超过 100 个商业支持的嵌入式操作系统,但是没有哪个操作系统在整个嵌入式实时操作系统中占有较大的份额,更不用说那些为了特殊应用(移动电话、雷达、网络设备等)场合而开发的操作系统了,而这些操作系统除了在这些场合以外就没有应用开发人员。简而言之,嵌入式实时操作系统市场以前是相当不成体系的,而这种不成体系已经开始限制许多嵌入式软件项目的可见性和投资这些软件项目的 OEM 的生存。

很显然,主要的嵌入式软件公司需要从根本上改变他们的经营模式,公司都认为这样可以使他们的公司更具有竞争力,并且通过一种摩擦融合战略,而不是革新战略来巩固市场。在 Cygnus 方案中,我们不愿放弃其中 90% 的市场,而且,也不愿意在这已经进入死胡同的 90% 市场份额基础上折腾。因此,我们利用大量的嵌入式系统设计方案努力创造一个利用同样资源的嵌入式操作系统,并且其从简单的嵌入式操作系统到很复杂的嵌入式操作系统应用场合都能胜任。

在我们的市场研究中,发现人们自己编写实时操作系统的主要原因有两个:首先就是他们不想付单件产品使用费给第三方;其次就是他们不想承受那些间接代码的开销,即那些不是他们编写的、自己不好控制也不理解的代码消耗了系统的资源。事实上,编写和调试实时嵌入式操作系统的代价是很昂贵的(时间上和金钱上),而哪怕对实时嵌入式操作系统做很小的手工改动,大多数特定的实时操作系统要求对整个系统要有全面理解的事

实,常常使得系统的维护费用很高而功能方面的商业可替代性很弱,但这却是避免直接和间接单件产品开销的一种折衷。

eCos 的设计理念就是增强具有配置工具的开源嵌入式实时操作系统(意味着没有单件产品使用费)功能,使得开发人员可将操作系统从数百字节到上千字节进行配置而不需要手动改变一行代码。当然,如果一些代码需要重写以适应某些特殊要求,开源许可意味着可以这样做。但在绝大多数情况下,eCos 支持的 200 多个配置项使得可以很快地建立相应的系统(所有的艰辛活都已经编程为配置项),而且通过配置项生成的系统比手工方法生成的系统要小(因为自动规则的全盘能力比大多数嵌入式开发人员要强)。

自 eCos 在 1998 年发布以来,我们注意到一方面具有越来越多的用户,另一方面得益于那些天才的开发人员。随着本书的出版,eCos 会达到新的里程碑:有关 eCos 的完整技术资料 and 相当完整的代码。本书虽然主要面向嵌入式实时操作系统的工程人员,但技术主管和开发人员同样用得着,尽管他们实际上不维护嵌入式实时操作系统。

本书涵盖 eCos 的最新技术细节,包括用一章来介绍使用 eCos 的兼容层提供 POSIX、 μ Itron 服务和兼容 EL/IX 的嵌入式 Linux API。事实上,作为以嵌入式 Linux 为基础的有先进特点的嵌入式系统设计,eCos 之所以正在变得越来越重要有两个原因:首先它提供了转向 Linux API 的平台,而不需要完全运行基于 Linux 的系统;其次 eCos 是 RedBoot 的基础,RedBoot 是新的标准 ROM Monitor,Red Hat 利用它来支持嵌入式 Linux 移植。

Anthony 的这本书是最全面地讲述 eCos 系统开发的书籍,我相信它一定会成为 eCos 开发者案头的一本书。

Michael Tiemann
CTO, Red Hat, Inc.

前言

不论你是否正在做一个项目还是转向一个新的开发,也不论最后你决定采用何种嵌入式实时操作系统,都会产生很多问题,包括启动资金得花多少,它们是否包括与应用 RTOS 相关的技术使用费以及开发工具的质量如何和是否可以得到源代码,该嵌入式操作系统有些什么特点等。在大多数情况下,直接费用与技术使用费方面成本最低的方案就是最好的方案,要长期使用,排除技术使用费对于大批量的产品是非常重要的。这就要使用好每一分钱。

也要计算先前投入,同时估计目前方案的开发人员的学识和费用等,将一些现成的代码转到新的平台时可能会使人产生忧虑,这取决于项目的大小。将新的嵌入式实时操作系统移植到自己的新硬件会产生更多忧虑。在是否开发自己的嵌入式操作系统抑或使用已经商业化了的方案方面的决定,在某些方面是非常表面的,特别是当某些特定的硬件平台需要一些特殊功能的时候。在一些情况下,开发自己的嵌入式实时操作系统可能是唯一的解决方案。然而,可以通过已经应用过的软件提前进行开发,在各种平台上、不同情形下测试之;更为重要的是,以在即将交付使用的产品的正确运行来证明其可行性。这就排除了进行功能测试的必要。

本书着眼于这些方面的一个解决方案:嵌入式可配置操作系统(eCos)。eCos 的源代码开放和无技术使用费的特性使得其可以下载、安装和使用,而关键就是没有任何成本。当看完这本书时,你就有了一个完整的嵌入式软件开发环境——开发任何项目的所有必要的工具。

由于 eCos 是开源的,作为开发人员,处于对自己的嵌入式软件完全控制的地位,甚至在 eCos 开发系统中描述的工具也是开源的,因此允许完全自己作主,而且 eCos 开发社区可为你提供必要的帮助。

本书结构和内容浏览

现在来看本书的结构,并了解本书涵盖的内容及其所在的章节,这样就可以只看自己需要理解的 eCos 某一方面的内容。本书尽量在章节的开始部分提供内容信息。先从 eCos 的关键组件入手,然后是系统提供的附加功能,最后是试图使用 eCos 和开发环境。

对于新的 eCos 开发人员,或者对嵌入式软件也不熟悉的人员,从零开始讲述,对于其理解组成 eCos 系统的组件是很有帮助的。这给了他们一个理解 eCos 特点的基础,然后就可可在实际的系统中应用这些组件。

对于那些仅仅是测评 eCos 的有经验的开发人员,他们可以跳过接下来的一些章节,直接体验 eCos、开发平台的安装和一些可以快速安装的工具和立即得出结果的例子。这就回答了这些疑问:“eCos 能在我的项目中工作吗?”等。

当前 eCos 用户的疑问都可通过不同的视角来理解 eCos 的概念,从而在 eCos 的知识中找到答案。

第 1 章“eCos 导引”,从 eCos 的介绍开始,其中包括 eCos 开源项目的背景及其支持公司;同时也对 eCos 的术语进行了详细的描述,这些术语贯穿于本书的始终,并流行于 eCos 开发社区。本书的开始部分试图给那些不熟悉 eCos 的开发人员提供一种熟悉 eCos 开源项目的途径。

接下来,粗略地讨论了 eCos 系统中的关键组件。在关键组件的章节中提供了这些不同的软件组件是如何独立而又协同工作以提供系统所需要的功能的。

第 2 章“硬件抽象层”,集中讨论与硬件相关的软件,正是这些软件对高级别的软件模块屏蔽了硬件的低级别操作。

第 3 章“异常与中断”,详细讨论了异常与中断,并说明了它们是如何在 eCos 系统中建立和处理的。在第 4 章“虚拟向量”中讨论了虚拟向量,它提供了一种 ROM 和 RAM 应用程序共享服务的方法。eCos 嵌入式操作系统的核心——内核,是在第 5 章“内核”中讲述的。内核提供调度功能和软件的同步机制。第 6 章“线程和同步机制”,讨论了 eCos 中的执行基本单位——线程,并提供了 eCos 所支持的各种同步机制的详细情况。

第 7 章“其他 eCos 架构组件”,通过时间组件、声明和跟踪功能和 I/O 控制系统来继续讨论 eCos 的不同组件。

第 8 章“附加功能和第三方组件”，包括大量的 eCos 中可利用的一些额外特性。它们是由 eCos 维护人员和第三方人员开发的，这些包含网络支持、ROM Monitor、文件系统、PCI 总线支持、USB 支持和 GoAhead 网络服务器等。

第 9 章“RedBoot ROM Monitor”，主要讲述 RedBoot ROM Monitor。这个独立的程序设计用于嵌入式系统的调试和启动环境。RedBoot 是一个基于 eCos 的程序，是利用硬件抽象层作为其基础的。

第 10 章“宿主开发平台”，从开发即开发工具开始手把手地实践。讨论了 Cygwin 本地工具、GNU 交叉开发工具和 eCos 开发包，也讨论了 CVS 客户端的安装。该工具可以链接在线的 eCos 源代码库，这就给了用户充分利用补丁和 eCos 源代码扩展功能的机会。

在第 11 章“eCos 工具集”中，研究了 eCos 的开发工具集，详细讨论了这些工具是如何在 eCos 源代码上运行的，以及这些工具的布置，也讨论了一些开源工具和完全开源的嵌入式开发系统，为下一步作好准备：利用这些工具来建立应用程序。

第 12 章“eCos 应用实例”，使你学以致用。本章从建立 eCos 映像流程的综述开始，接下来就是建立 RedBoot ROM Monitor，然后就是建立 eCos 嵌入式实时操作系统，最后加入附加组件来集成系统以适应嵌入式软件的需求。第 13 章“eCos 的移植”，讨论如何将 eCos 移植到其他的硬件平台上。这是将自己的应用程序在新目标硬件平台上运行起来的关键一步，也是在嵌入式软件开发中的主要目标。

开发系统与实例

如前所述，在第 10 章我们讨论建立 eCos 开发环境的流程，该开发系统包括本地 Windows 版的 Cygwin 工具、GNU 交叉开发工具(binutils、compiler 和 debugger)、eCos 配置与管理工具、CVS 客户端、lint 程序。该系统可使用户配置和建立 eCos 库，然后把它们链接为运行 eCos 实时嵌入式操作系统的应用程序。紧接着就是配置一个完全开源的嵌入式开发环境。

在第 12 章给出了建立 RedBoot、eCos 库和应用程序的实例，虽然不要求特定的开发板来运行示例程序，但一台具有目标平台的 PC 机还是需要的。这是熟悉开发工具的较好方法，因为手头总是很容易找到空闲的 PC 机。虽然原版书的光盘*中包含有 Intel x86 系列和 PowerPC 系列处理器架构的 GNU 交叉开发环境工具的二进制文件，而为其他架构的处理器建

* 原版书的光盘内容可以在 <http://www.phptr.com/title/0130354732#> 中下载。

立 GNU 交叉开发环境的指导是包含在附录 D“建立 GNU 交叉开发工具”中的。

在 Windows 开发系统中安装嵌入式软件开发工具后,也建立了示例程序。但是,在 Linux 系统中建立嵌入式开发系统和一些必要文件也包含在光盘中。由于 eCos 配置工具可在 Linux 和 Windows 下运行,建立开发系统和运行示例程序的过程对两个操作系统都是适用的。

原版书赠送的光盘包含第 10 章所述的建立 eCos 嵌入式软件开发系统所需的文件,第 12 章和第 13 章的示例程序包含在原版书光盘的 examples 目录下。也可访问 <http://www.phptr.com/massa/> 网站下载源代码和升级代码。

如果发现需要更正的错误,随时与作者联系,我会将源代码进行相应的更正。

有关本书的说明

所有本书涉及的示例代码都是 C 语言或汇编语言的形式。eCos 也使用到了组件定义语言(CDL, Component Definition Language),CDL 是对已经存在的工具命令语言(TCL, Tool Command Language)进行了扩展的描述语言,将在第 11 章讲述该内容。

为了阅读方便,在需要显示 32 位十六进制值的文本文件中,高 16 位和低 16 位是用下划线“_”分开的(如 0xABCD_EF12)。也包含补充部分,它用来表明一些重要的、附加的信息。补充部分如下所示:

注意: 补充部分示例。

本书包含一些统一资源定位器(URL, Uniform Resource Locator)链接,用它来表示在 Internet 上何处可获得有关内容的更多信息。最新的链接包含在文本文件中,众所周知,链接不一定是永远存在的。

本书通篇列出的条目就是 eCos 的内核函数,这些列表包含一个叫做 context 的域,这个域说明了特定函数在何处可以调用的原因。这些包括初始化、ISR 和 DSR。any 表示在任何情况下都可以调用。

开放源代码运动发展概述

在开放源代码和不开放源代码的争论中,很多东西是不明朗的,有时会演变成是否有资金来购买软件工具和对所开发的软件进行支持。开放源

代码和不开放源代码各有优缺点。现在就来看一看开放源代码的开发人员和不开放源代码的开发人员相互在对方的角度所发现的问题,问题最终归结为产品能否工作、产品计划和经费预算是否一致。

开放源代码可能是一个令人困惑的术语。在 Daniel Benenstien* 撰写的一篇文章中,他认为开放源代码是出于知识自由交流意义的免费软件。更多人认为是由于个人开发者发现问题、解决问题,并将解决方案在开源社区中提供给其他人的过程。这个反复的过程就是创造更为稳定和没有缺陷代码的过程。

设想一下,因为能干的人是分布在全世界的,最好和最聪明的开发人员正努力使得产品更好。如果不开放源代码,开发、支持团队就忙于修正那些不是他们编写的代码的错误,这个团队整体可能很能干,然而,他们的能干与世界范围内的能干比较起来就非常渺小了。

经常是不开源或具有产权的软件和开源软件一起工作而又相互补充,许多具有产权的产品就是起源于开源项目。具有产权的软件必定有革新,而且不同于其起源的开源项目;不然,人们怎么会愿意付钱去买某个可以免费的东西呢?从这个视角来看待事物,开放的源代码将具有产权的软件推到可革新的极至。使用开源软件有几个优点和缺点,其中一个优点就是由于源代码可以得到,当从开源社区得不到软件支持时,便可以研究代码,看看代码是如何运行的,而没有必要从其他渠道来等待软件支持,这样就可大大减少调试时间。

开源软件的另一个优点就是该软件不会被局限于某一家公司。拿具有产权的软件来说,如果其归属公司改变了经营方向,或者跨了,则采用其方案进行应用开发的人员就被晾在一边了。开源软件就没有这个问题,因为开发人员对代码可以完全进行控制,可以按自己的思路选择自己的系统软件。

有时所有商销售的具有产权的软件并不总是适合所有使用其产品的客户。在不会产生大量数据的项目中,所有商对此问题的解决就不如需要的快。事实上,只有具有较高收益产品的公司才可能优先做到这一点。

拥有源代码,就给了实现按自己的意愿改变代码的权利,就可以按自己的应用需要来定制代码。另外,只有源代码开放,才可能有如我们在 eCos 中看到的那样源代码级的配置方法可用。虽然一些具有产权的软件生产商

* Benenstien, Daniel. Galleo Linux Multimedia Communicator[J]. *Embedded Linux Journal* (July/August 2001): 14 ~19.

也提供其源代码给开发人员,但一般都是需要付费的。

安全性通常被视为开放源代码的一个负面因素,因为每个人都可得到所有源代码,恶意的开发人员就能利用其安全漏洞。另一方面,支持该开放源代码工程的开发者社区会很快进行修正,因为他们也很关心安全问题。你不是处于仅仅是某一个人的提供安全问题修正的境地。

更多的开源运动的观点可以参考 *Open Sources: Voices from the Open Source Revolution*^{*}。

致 谢

没有其他人的艰苦工作和大量的努力,本书是不可能完成的。首先我得感谢技术评论家 Jonathan Larmour,我感谢他非常具有洞察力的评价和在本书写作过程中的技术支持(偶尔是实时的);其他一些与我分享他们敏锐的评价和非常有价值的观点的人,我应该感激他们,他们是 Grant Edwards、Bart Veer、Bill Gatliff、Larry Mittag 和 Paul Beskeen。我还得感谢为本书写序的 Michael Tiemann。

感谢编辑 Mark Taub,感谢他在整个写作过程中的信息反馈和支持。

感谢那些包含在光盘中软件背后的公司和开源项目人员,这些软件可是非常有用的东西。

前言最后的话

我得感谢我的 Nonno 和 Nonna 在我整个生活中的支持,他们在我无论需要什么的时候总是在我身边,而不论我是否给了他们奖励,也无论是到后院骑马或吃饭,或者是到 Bool Weevil 去旅行,我很想念 Nonna,我爱你们两个!

我得谢谢我的兄弟 Laurie 和妹妹 Catherine,谢谢他们的鼓励和理解,你们是每个人想得到的最好兄弟和妹妹。顺便说一下,恭喜我的妹妹通过了 CPA 考试。我爱你们!

谢谢我的妈妈和爸爸,你们在我职业生涯中给了我无尽的支持和鼓励,尤其是我一生都在做的事。无论我在干什么,你们总是在我的身边,有这样的父母,我非常感激,我全心全意地爱你们!

最后,我希望你们喜欢这本书!

Anthony J. Massa
amassa@san.rr.com

* DiBona, Chris; Ockman, Sam; Stone, Mark. *Open Sources: Voices from the Open Source Revolution*. (O'Reilly, 1999).

目 录

第 1 章 走近 eCos 的世界

1.1 一切从这里开始——Cygnus Solutions	1
1.2 eCos 的由来	2
1.2.1 简而言之——可配置性	3
1.2.2 eCos 的配置方法	3
1.2.3 eCos 核心组件	5
1.2.4 处理器和评估板的支持	6
1.2.5 eCos 的支持	6
1.3 eCos 的体系结构	7
1.3.1 组件框架	9
1.3.2 组件库	9
1.3.3 配置选项	12
1.3.4 组件和包	13
1.3.5 目标机	13
1.3.6 模 板	13
1.4 本章小结	14

第 2 章 硬件抽象层 HAL

2.1 概 述	15
2.1.1 HAL 的目录结构	17
2.1.2 HAL 的宏定义	20
2.1.3 HAL 的配置	22
2.1.4 HAL 的启动过程	24
2.2 本章小结	27

第 3 章 异常与中断

3.1 异 常	28
3.1.1 HAL 与内核异常处理	30

3.1.2 应用程序异常处理.....	35
3.2 中 断.....	36
3.2.1 eCos 中断模型	37
3.2.2 中断配置.....	38
3.2.3 中断处理.....	40
3.2.4 中断控制.....	46
3.3 本章小结.....	54
第 4 章 虚拟向量	
4.1 虚拟向量.....	55
4.1.1 虚拟向量配置.....	57
4.1.2 虚拟向量表的初始化.....	59
4.2 本章小结.....	66
第 5 章 内 核	
5.1 内 核.....	67
5.1.1 内核目录结构.....	68
5.1.2 内核启动.....	69
5.1.3 调度器.....	70
5.2 本章小结.....	76
第 6 章 线程与同步机制	
6.1 线 程.....	77
6.1.1 线程堆栈与堆栈大小.....	86
6.2 同步机制.....	87
6.2.1 互斥量.....	88
6.2.2 信号量.....	93
6.2.3 条件变量.....	98
6.2.4 标志量	103
6.2.5 消息邮箱	107
6.2.6 螺旋锁	112
6.3 本章小结	114

第 7 章 其他 eCos 架构组件

7.1 计数器、时钟、报警器和定时器	115
7.1.1 计数器	119
7.1.2 时 钟	123
7.1.3 报警器	125
7.1.4 定时器	128
7.2 声明和跟踪	128
7.3 ISO C 和数学库	133
7.4 I/O 控制系统	135
7.4.1 I/O 子系统	136
7.4.2 设备驱动程序	140
7.5 本章小结	142

第 8 章 附加组件和第三方组件

8.1 兼容层	144
8.1.1 POSIX	144
8.1.2 μ ITRON	145
8.2 ROM Monitor	146
8.2.1 CygMon	146
8.2.2 RedBoot	147
8.2.3 GDB Stub	147
8.3 文件系统	149
8.3.1 ROM 文件系统	150
8.3.2 RAM 文件系统	151
8.3.3 日志闪盘文件系统	153
8.4 PCI 总线支持	154
8.4.1 PCI 库 API	154
8.5 USB 总线支持	158
8.6 网络支持	159
8.6.1 OpenBSD	160
8.6.2 FreeBSD	161
8.6.3 LwIP	162
8.6.4 网络线程	162