



附赠：源代码、关键操作视频

(第2版)

VxWorks内核、设备驱动 与BSP开发详解

张杨 于银涛 编著

(第2版)

VxWorks内核、设备驱动 与BSP开发详解

张杨 于银涛 编著

人民邮电出版社
北京

图书在版编目 (C I P) 数据

VxWorks内核、设备驱动与BSP开发详解 / 张杨, 于银涛编著. -- 2版. -- 北京: 人民邮电出版社, 2011. 3
ISBN 978-7-115-24472-7

I. ①V… Ⅱ. ①张… ②于… III. ①实时操作系统, VxWorks IV. ①TP316. 2

中国版本图书馆CIP数据核字(2010)第232063号

内 容 提 要

在嵌入式行业飞速发展的今天, 嵌入式操作系统的应用越来越广泛, VxWorks 具有良好的实时性, 被大量应用于航空、通信、国防、工业控制、网络设备、医疗设备、消费电子等嵌入式实时应用领域。

本书共 24 章, 按照初学者学习的一般步骤, 详细介绍了 VxWorks 的开发环境、内核分析和应用程序开发、驱动程序开发与原理和板级支持包流程与移植, 每章均配以实例进行讲解。

本书语言通俗、实例丰富、代码分析详尽, 有较强的实用性和参考价值, 适合大专院校嵌入式相关专业学生学习参考, 也可供嵌入式开发人员和系统设计人员参考使用。

VxWorks 内核、设备驱动与 BSP 开发详解 (第 2 版)

- ◆ 编 著 张 杨 于银涛
责任编辑 蒋 佳
- ◆ 人民邮电出版社出版发行 北京市崇文区夕照寺街 14 号
邮编 100061 电子函件 315@ptpress.com.cn
网址 <http://www.ptpress.com.cn>
北京鑫正大印刷有限公司印刷
开本: 787×1092 1/16
印张: 21.25
字数: 507 千字 2011 年 3 月第 2 版
印数: 3 001-6 000 册 2011 年 3 月北京第 1 次印刷

ISBN 978-7-115-24472-7

定价: 49.00 元 (附光盘)

读者服务热线: (010) 67132692 印装质量热线: (010) 67129223
反盗版热线: (010) 67171154

第 2 版说明

嵌入式系统应用已经深入到我们生活中的各个领域，手机、机顶盒、MP4、交换机、路由器等都可以归结为嵌入式设备。在这些嵌入式设备中，有一部分设备需要功能强大、实时性强的操作系统，VxWorks 正是这类操作系统中的佼佼者。从 VxWorks 5.4 进入中国以来，已经过去了 10 多年，VxWorks 在国内的应用不但没有减少，反而呈现了面向架构多、应用范围广、民用化的特性，越来越多的嵌入式设备转而采用 VxWorks 实现。

本书的第一版《VxWorks 内核、设备驱动与 BSP 开发详解》于 2009 年 10 月上市，受到了广大读者的一致好评。很多读者都给予本书极高的评价，并提出了再版的建议和意见。针对技术的变化，为了完善递一本的内容，我们经过精心策划与改编后推出了本书的第 2 版。

关于本书

随着 VxWorks 应用逐渐广泛，相关书籍却不多。本书结合作者在 VxWorks 的多年工作经验，仔细描述了实际开发过程中可能遇到的一些问题，并深入分析 VxWorks 系统的工作机制，旨在希望能够给刚接触 VxWorks 的初学者及对 VxWorks 有一定了解的朋友提供一些帮助，为熟悉 VxWorks 的读者提供分析问题、设计软件的途径和思路。

本书内容结构

本书按照学习使用 VxWorks 的过程，由浅入深分为 4 个部分：VxWorks 的开发环境、内核分析和应用程序开发、驱动程序开发与原理、板级支持包流程与移植，共 24 章。

第 1 部分为 VxWorks 的开发环境，介绍了 Tornado 2.2 开发环境的使用方法、调试过程中的灵活运用、机理分析等内容，主要包括如下内容。

- 简单接触 Tornado 环境；
- 编译工程，分析 MakeFile；
- 加载操作系统，分析网络加载中错误的原因；
- 管理工程文件、组件和编译选项；
- 连接目标机，解决无法连接的问题；

- 交叉调试器，为程序设置多样化的断点；
- 浏览目标机资源，如何更好地利用 Browser；
- 合理运用 Shell、WindView 和 Triggering；
- 集成版和完整版的 VxSim。

第 2 部分为内核分析和应用程序开发，分析了内核的工作机理，为应用的设计和开发打下基础，同时还带有各种应用开发示例，其内容主要包括如下内容。

- 任务的状态和控制，内核调度任务的机理；
- 任务间的同步、通信，临界区域的保护；
- 系统计时和延时，各种计时方式的示例和比较；
- 文件系统的机理和访问方式，如何更好地利用文件系统；
- 通过 I/O 系统访问系统的外围设备；
- 访问网络，编写 TCP、UDP、zBuf 的客户端和服务端程序；
- VxWorks 程序设计中需要考虑的问题。

第 3 部分驱动程序开发与原理，描述了驱动程序层 VxWorks 的处理方式以及驱动程序的编写方法，给出了部分驱动程序示例，其内容主要包括如下内容。

- 利用并访问硬件资源，包括不同架构的内存、I/O 访问；
- 字符设备驱动程序的机理和编写方法；
- 块设备驱动程序的机理和编写方法；
- 网络设备驱动程序的机理和编写方法；
- USB 设备主机端驱动程序的机理和编写方法；
- USB 设备设备端驱动程序的机理和编写方法。

第 4 部分板级支持包流程与移植，描述了 VxWorks 中 BSP 的工作原理、用户修改 BSP 所需要的步骤以及如何同时让开发环境 Tornado 能够识别 BSP 的更改，其主要内容包括如下内容。

- BSP 工作流程以及移植过程中的修改；
- BSP 与开发环境结合的工具——CDF 格式。

最后，本书的附录介绍了新的 VxWorks 6，比较了它与 VxWorks 5.5 开发套件之间的差别。

本书特色

- 将开发中可能遇到的问题现象一一列举，给出其解决办法并分析其产生原因；

- 利用多实例详细描述开发过程，读者可以直接根据步骤操作；
- 分析各种机制实现机理，由浅入深剖析 VxWorks；
- 提供了多个可以直接应用的实例，利于读者直接分析、使用；
- 图例丰富，便于读者理解和操作。

联系作者

本书由张杨、于银涛执笔。

由于时间仓促，加之水平有限，书中的不足之处在所难免，敬请读者批评指正。

编者
2010年12月

目 录

第 1 部分 VxWorks 的开发环境

第 1 章	使用 Tornado	2
1.1	概述	2
1.2	Tornado 的安装目录	2
1.3	创建一个工程	4
1.3.1	可加载工程和可引导工程	4
1.3.2	实例: 创建可加载工程	5
1.3.3	实例: 创建可引导工程	7
1.4	更加有效率地工作	8
1.4.1	开发环境基本设置	8
1.4.2	实例: 设置一个更熟悉的编辑器	10
1.4.3	使用 Tcl 语言定义工具	11
第 2 章	编译工程——Builder	13
2.1	编译工程及 Boot Rom	13
2.1.1	编译工程	13
2.1.2	编译 Boot Rom	15
2.1.3	实例: 制作软盘引导盘	15
2.1.4	深入 1: 批处理文件 mkboot	17
2.1.5	深入 2: 制作 U 盘引导盘	18
2.1.6	深入 3: 制作硬盘引导盘	21
2.1.7	深入 4: 制作一个引导 ROM	21
2.1.8	实例: 多系统并存	23
2.2	深入编译环境	24
2.2.1	创建的过程	24
2.2.2	抛弃 IDE	25
2.2.3	使用 Diab	26
2.3	了解 MakeFile	26
2.3.1	MakeFile 的作用	26
2.3.2	MakeFile 格式	27
2.3.3	解决 MakeFile 所导致的错误	28

第 3 章 加载——FTP Server	29
3.1 FTP Server 的设置.....	29
3.1.1 用户和权限.....	29
3.1.2 其他设置.....	30
3.1.3 实例：下载一个工程.....	30
3.2 从信息看状态.....	32
3.2.1 正常下载过程.....	32
3.2.2 下载失败的原因.....	33
3.3 从其他设备加载操作系统.....	35
第 4 章 管理工程——Work Space	36
4.1 管理文件.....	36
4.1.1 文件的位置.....	36
4.1.2 文件依赖关系.....	36
4.2 管理组件.....	37
4.2.1 VxWorks 的各种组件.....	37
4.2.2 查找组件并修改属性.....	38
4.2.3 裁剪组件.....	39
4.3 编译选项.....	39
4.3.1 编译规则.....	40
4.3.2 宏定义.....	41
4.3.3 其他选项.....	42
第 5 章 连接目标机——Target Server	43
5.1 配置 Target Server.....	43
5.1.1 基本配置内容.....	43
5.1.2 重要配置属性.....	44
5.2 实例：通过网络连接目标机.....	47
5.3 实例：配置串口连接主机.....	48
5.4 连接错误的解决办法.....	48
5.4.1 目标文件不正确.....	48
5.4.2 网络错误.....	49

5.4.3	目标机无响应	49
5.4.4	Target Server 缓冲池不足	50
5.4.5	网络速度慢	50
第 6 章	交叉调试——Debugger	51
6.1	基本调试过程	51
6.1.1	运行 Debugger	51
6.1.2	发起任务	52
6.1.3	连接任务	52
6.2	断点	53
6.2.1	设置断点	53
6.2.2	管理断点	54
6.3	独特的查看功能	55
6.3.1	源代码与汇编混合查看	55
6.3.2	调试命令行	56
6.3.3	其他功能	57
6.4	系统级调试设置	57
第 7 章	浏览目标机资源——Browser	58
7.1	Browser 所提供的信息	58
7.1.1	Browser 的设置	59
7.1.2	目标机信息	59
7.2	实例：透过现象看调试	60
7.2.1	查看内存泄露	60
7.2.2	分配任务优先级	60
7.2.3	确定任务堆栈	61
7.2.4	查看中断挂接情况	61
7.2.5	查看任务状况	62
第 8 章	与目标机的交互——Shell 及其他	64
8.1	Shell	64
8.1.1	Shell 的常用特性	64

8.1.2	Shell 的快捷操作	64
8.1.3	Shell 的常用命令	67
8.1.4	Shell 的环境变量	69
8.1.5	执行脚本	69
8.2	WindView	70
8.2.1	设置 WindView	70
8.2.2	分析任务时序图	72
8.3	Triggering	75
8.3.1	Triggering 设置	75
8.3.2	Triggering 的使用	77
第 9 章	软件硬件同时开发——VxSim	78
9.1	集成的 VxSim	78
9.1.1	VxSim 适用范围	78
9.1.2	VxSim 的配置	79
9.1.3	VxSim 的中断系统	80
9.2	功能更强的 Full VxSim	81
9.2.1	Full VxSim 所需的网络配置	81
9.2.2	创建支持网络连接的 Full VxSim 映像	82
9.2.3	启动 Full VxSim	83
9.2.4	运行多个 Full VxSim	85
9.3	实例：在 VxSim 上运行程序	85

第 2 部分 内核分析和应用程序开发

第 10 章	管理任务——任务的状态	90
10.1	任务概况	90
10.1.1	任务状态	90
10.1.2	任务调度	92
10.1.3	任务优先级	93
10.2	生成和管理任务	94
10.2.1	任务控制 API	94

10.2.2	销毁任务	95
10.2.3	控制任务状态的切换	95
10.3	深入：任务切换的深层原因	96
10.3.1	时钟中断和时间片调度	96
10.3.2	优先级抢占调度机制	98
10.3.3	任务控制块	99
第 11 章	任务间的交互——同步与通信	103
11.1	任务之间的交流	103
11.1.1	共享内存	103
11.1.2	互斥	106
11.1.3	信号量	107
11.1.4	消息队列和管道	110
11.1.5	使用网络的任务间通信	114
11.1.6	信号	115
11.2	实例：任务间交流	116
11.2.1	通过信号量实现互斥	116
11.2.2	信号量实现同步	117
11.2.3	管道与消息队列	118
11.3	其他通信手段	119
第 12 章	控制时间——系统计时	121
12.1	系统时间	121
12.1.1	系统时间的来源	121
12.1.2	与现实时间相联系	121
12.2	系统计时控制	122
12.3	看门狗计时器	124
12.3.1	实例 1：看门狗计时	125
12.3.2	实例 2：看门狗定时	125
12.3.3	实例 3：看门狗延时	126
12.4	其他的计时方法	127
12.4.1	使用 CPU 时间戳计时	127

12.4.2	实例: 计算 CPU 频率	128
第 13 章	存储信息——访问文件系统	129
13.1	VxWorks 中的文件系统	129
13.1.1	操作文件的办法	129
13.1.2	实例: 文件操作	132
13.1.3	深入: 打开文件的过程	134
13.1.4	深入: 读取文件的过程	135
13.1.5	文件系统的作用	136
13.1.6	文件的特殊功能	138
13.1.7	文件的安装流程	138
13.2	文件系统分析	139
13.2.1	dosFs	139
13.2.2	rawFs	142
13.2.3	cdromFs	143
13.2.4	tapeFs	144
13.3	深入: 如何提高访问效率	144
13.4	实例: 支持多硬盘分区	145
第 14 章	与外界的交互——控制硬件	147
14.1	驱动程序管理	147
14.1.1	文件、设备、驱动程序	147
14.1.2	驱动管理 3 张表	148
14.1.3	驱动及设备的管理	152
14.2	访问硬件的手段	152
14.2.1	创建设备	152
14.2.2	操作设备	154
14.3	操作硬件	155
14.3.1	实例: 测试用程序	155
14.3.2	多设备异步通信——select	161
14.4	VxWorks 中的隐含设备	162

第 15 章 联通的世界——访问网络	164
15.1 VxWorks 中的网络组件	164
15.1.1 层次结构	164
15.1.2 基本组件	166
15.1.3 实例：配置网络	167
15.1.4 操作路由表	171
15.1.5 Zbuf	173
15.1.6 使用标准 I/O 访问函数	174
15.2 实例：网络编程	175
15.2.1 建立 TCP 连接	175
15.2.2 建立 UDP 连接	178
15.2.3 使用 Zbuf 访问 TCP	181
15.2.4 创建及使用 FTP	183
第 16 章 VxWorks 程序设计	186
16.1 资源	186
16.1.1 资源的有效利用	186
16.1.2 资源的保护	187
16.2 优先级安排	188
16.3 保证时间特性	189

第 3 部分 驱动开发与原理

第 17 章 硬件资源——操作硬件	192
17.1 硬件访问	192
17.1.1 内存访问	192
17.1.2 通用的内存映射程序	193
17.1.3 I/O 端口访问	196
17.2 中断	197
17.2.1 轮询方式	197
17.2.2 中断	198
17.2.3 中断号与中断向量	201

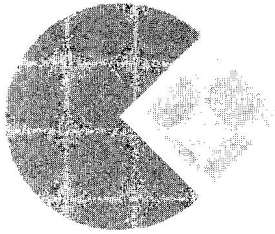
17.2.4	中断复用	201
17.2.5	中断处理中的问题	202
17.3	访问特定硬件的 API	203
17.3.1	访问 PCI	203
17.3.2	中断相关的 API	205
第 18 章	大道至简——字符设备	207
18.1	字符设备驱动程序	207
18.1.1	字符设备驱动程序的数据结构	207
18.1.2	字符设备驱动程序的接口	209
18.1.3	中断处理	215
18.1.4	安装驱动程序和创建设备	215
18.1.5	支持 Select 功能	217
18.2	实战字符设备开发	218
18.2.1	板卡介绍	219
18.2.2	程序实现	219
第 19 章	永久的记忆——块设备	232
19.1	块设备驱动程序的结构	232
19.2	块设备驱动程序的接口	234
19.3	安装块设备驱动程序	236
第 20 章	联通的基础——网络设备	237
20.1	网络设备的数据结构	237
20.2	网络数据传递	239
20.3	网络设备接口层	242
20.3.1	endLoad	243
20.3.2	endUnload	244
20.3.3	endSend	244
20.3.4	endMCastAddrAdd、endMCastAddrGet 和 endMCastAddrDel	245
20.3.5	endPollSend、endPollReceive	245

20.3.6	endStart	246
20.3.7	endStop	246
20.3.8	endAddressForm	246
20.3.9	endAddrGet	246
20.3.10	endPacketDataGet	247
20.3.11	endIoctl	247
20.4	中断模式下接收数据	248
20.5	网络设备驱动程序的安装	249
第 21 章	USB 设备——USB 1.1 支持	251
21.1	USB 工作过程	251
21.2	USB 外设端协议栈	252
21.2.1	USB 外设端协议栈结构	252
21.2.2	USB 外设端协议栈实现	253
21.2.3	USB 外设端协议驱动层的使用	255
21.2.4	USB 外设端控制器驱动程序	263
21.2.5	再看“回调函数”	267
21.3	USB 主机端协议栈	268
21.3.1	USB 主机端协议栈结构	268
21.3.2	USB 主机端协议栈使用	269
21.3.3	USB 主机端驱动程序	272
第 22 章	USB 设备——USB 2.0 支持	277
22.1	设备端协议栈	277
22.2	主机端协议栈	283
22.2.1	USB 与驱动程序	284
22.2.2	与已有程序的兼容	286

第 4 部分 板级支持包流程与移植

第 23 章	多样化的主板——板级支持包 BSP	288
23.1	引导过程	288

23.2	BSP 的流程	289
23.2.1	BSP 的作用和结构	289
23.2.2	BSP 初始化硬件	292
23.3	BSP 移植	293
23.3.1	实现 BSP 所需要的工作	293
23.3.2	BSP 中的系统接口	294
23.4	实例: BSP 包移植	295
23.4.1	开始移植 BSP 前的准备	296
23.4.2	提供中断控制器接口	297
23.4.3	提供时钟接口	297
23.4.4	提供硬件访问接口	299
23.4.5	提供设备支持——实时钟、GPIO 等	299
23.4.6	提供设备支持——串口	301
23.4.7	提供设备支持——网络	304
第 24 章	珠联璧合——BSP 与开发环境的结合	306
24.1	配置工具与 BSP 的关系	306
24.1.1	CDF 文件分析	306
24.1.2	Tornado 下通过 cdf 文件对 VxWorks 进行配置的过程	312
24.2	自制 BSP 配置	313
24.2.1	实例: 增加自定义文件夹结构	313
24.2.2	创建组件的一般步骤	316
附录	新的 VxWorks	318
FL.1	新开发环境	318
FL.1.1	新的界面	318
FL.1.2	新的开发工具	319
FL.1.3	开发环境的扩展	320
FL.1.4	与 Tornado 的对比	320
FL.2	新特性	321
FL.2.1	实时进程 RTP	321
FL.2.2	错误检测与报告	323



第1章 使用 Tornado

Tornado 是 WindRiver 公司为开发 VxWorks 提供的集成开发环境。在经历了最初的几个版本后，WindRiver 公司推出了 Tornado 2.2。相对于之前的版本，Tornado 2.2 的易用性和功能等得到了大幅地增强。由于目前 Tornado 2.2 的普及率非常高，因此本书以 Tornado 2.2 为基础介绍 Tornado 开发环境的使用。

1.1 概述

虽然 Tornado 2.2 相对于 Tornado 2.0 在版本号上只有很小的提升，却更加易用，功能上更为丰富，界面也更加友好。这种改进体现在很多细节上。

Tornado 2.2 提供了编译器的选择。Tornado 2.0 中使用的编译器是 GNU 的编译器，如果需要更换其他的编译器，必须使用命令行进行编译，也就无法使用集成开发环境所带来的便利。Tornado 2.2 在这方面有了很大的改进，第 2 章中会给出将默认的 GNU 编译器换成 Diab 编译器的例子。

更为人性化的编辑器。虽然 Tornado 2.2 所带有的编辑器依然很不方便，但是比起 Tornado 2.0 来说已经有了长足的进步。Tornado 2.2 支持使用“Ctrl+F”组合键弹出搜索框的方式，而在 Tornado 2.0 中，只能使用“Alt+F3”组合键开始搜索。更为重要的是，Tornado 2.2 支持自定义的编辑器，用户可以设定任意一种自己喜欢的编辑器来替代 Tornado 自带的编辑器，并从集成开发环境中调用它。

Tornado 2.2 的变化还有很多，这里没有必要一一列举。总而言之，在条件允许的情况下，请尽量放弃使用 Tornado 2.0 和 VxWorks 5.4，以减少不必要的麻烦。

在 VxWorks 6.0 诞生之际，WindRiver 公司提供了另一种集成开发环境“WorkBench”。这种开发环境基于开源工程“Eclipse”，习惯 Java 程序开发的读者对这个开发环境应该非常熟悉。WorkBench 的使用极为方便，然而由于软件升级等问题，国内还没有大量普及 VxWorks 6.0 和 WorkBench。

1.2 Tornado 的安装目录

Tornado 2.2 既拥有可以运行在 Windows 平台上的版本，也有可以运行在 UNIX 平台上的版本。