

VxWorks

驱动及分布式编程

孔祥营 张保山 俞烈彬 编著

先进性 ■ VxWorks版本不断提高,其功能越来越强,应用范围也越来越广,本书介绍的驱动开发和基于构件的分布式编程知识是目前研究的焦点。

实用性 ■ 全书图文并茂,理论分析和工程开发并重,结合作者多年丰富实践开发经验,给出了详尽的编程实例,具有很强的参考价值。

针对性 ■ 本书能够有针对性地提高读者的编程开发能力,是从事嵌入式开发技术人员的理想参考书。



中国电力出版社

www.infopower.com.cn

TP316.2VW
142
1=

VxWorks

驱动及分布式编程

孔祥营 张保山 俞烈彬 编著



中国电力出版社
www.infopower.com.cn

内容简介

本书在内容上分为两部分：驱动篇和分布式编程篇。驱动篇主要介绍了字符设备驱动、增强型网络设备驱动（END）以及 WindML 中文字库的设计和 MicroWindows 向 VxWorks 平台上的移植过程；分布式编程篇介绍了分布式构件对象模型（DCOM）和公共对象请求代理体系结构（CORBA）。

本书主要针对从事以 VxWorks 操作系统为基础内核的嵌入式系统开发人员，可以作为广大从事嵌入式技术相关工作的工程技术人员的参考书。

图书在版编目（CIP）数据

VxWorks 驱动及分布式编程 / 孔祥营，张保山，俞烈彬编著. —北京：中国电力出版社，2007.7

ISBN 978-7-5083-5645-7

（VxWorks 项目开发实践系列）

I. V… II. ①孔…②张…③俞… III. 实时操作系统，VxWorks—程序设计 IV.TP316.2

中国版本图书馆 CIP 数据核字（2007）第 068185 号

丛 书 名：VxWorks 项目开发实践系列

书 名：VxWorks 驱动及分布式编程

出版发行：中国电力出版社

地 址：北京市三里河路 6 号

邮政编码：100044

电 话：（010）68362602

传 真：（010）68316497，88383619

本书如有印装质量问题，我社负责退换

服务电话：（010）58383411（总机）

传 真：（010）58383267

E-mail: infopower@cepp.com.cn

印 刷：汇鑫印务有限公司

开本尺寸：185×260

印 张：23.75

字 数：580 千字

书 号：ISBN 978-7-5083-5645-7

版 次：2007 年 7 月北京第 1 版

印 次：2007 年 7 月第 1 次印刷

印 数：0001—4000 册

定 价：39.80 元

敬告读者

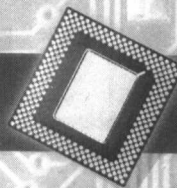
本书封面贴有防伪标签，加热后中心图案消失

本书如有印装质量问题，我社发行部负责退换

版 权 专 有 翻 印 必 究

知识改变命运，技术成就梦想！

嵌入式系统开发技术丛书



■ ARM系统开发从实践到提高

书号: ISBN 978-7-5083-5371-5 定价: 29.80元

■ DSP应用开发从实践到提高

书号: ISBN 978-7-5083-5365-4 定价: 22.00元

■ SoC系统开发从实践到提高

书号: ISBN 978-7-5083-5379-1 定价: 26.00元

■ ASIC芯片设计从实践到提高

书号: ISBN 978-7-5083-5378-4 定价: 26.00元

■ FPGA应用开发从实践到提高

书号: ISBN 978-7-5083-5377-7 定价: 27.00元

■ 嵌入式无线互连系统开发从实践到提高

书号: ISBN 978-7-5083-5642-6 定价: 35.00元

8051单片机技术应用系列丛书



■ 8051单片机彻底研究——入门篇

书号: ISBN 978-7-5083-5154-4 定价: 32.00元

■ 8051单片机彻底研究——实习篇

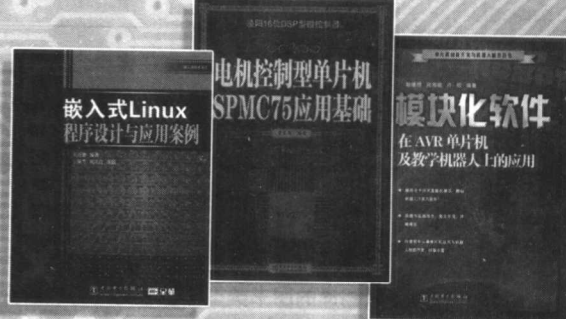
书号: ISBN 978-7-5083-4685-4 定价: 35.00元

■ 8051单片机彻底研究——基础篇

书号: ISBN 978-7-5083-4686-1 定价: 39.00元

■ 8051单片机彻底研究——经验篇

书号: ISBN 978-7-5083-5153-7 定价: 32.00元



■ 嵌入式Linux程序设计与应用案例

书号: ISBN 978-7-5083-5399-9 定价: 28.00元

■ 电机控制型单片机SPMC75应用基础

书号: ISBN 978-7-5083-4688-5 定价: 35.00元

■ 模块化软件在AVR单片机及教学机器人上的应用

书号: ISBN 978-7-5083-4666-1 定价: 19.80元 (含1CD)



中国电力出版社
www.infopower.com.cn

中国电力出版社计算机与艺术图书中心
地址: 北京市西城区三里河路6号 (100044)
电话: 010-58383411 传真: 010-58383267
E-mail: infopower@cepp.com.cn 网址: www.infopower.com.cn

中国电力出版社读者服务卡

非常感谢您选择中国电力出版社计算机与自动化类图书，您的支持是对我们工作最大的肯定！请对我们的图书提出宝贵的意见和建议，以帮助我们不断提升图书质量，继续推出更符合读者需求、更实用、品质更高的图书。

返回此服务卡后，您将成为我们的正式读者会员，能更快捷地了解最新的图书出版信息和优惠购书信息，谢谢！

姓名 _____ (必填) 性别 _____ 学历 _____

职业 _____ 职称 _____

年龄 ☐ 10~20 ☐ 20~30 ☐ 30~40 ☐ 40 以上

工作单位 _____

电子邮件 _____ (必填) 联系电话 _____ (必填)

通信地址 _____

邮政编码 _____

您经常阅读哪种类型的图书：

☐ 操作系统 ☐ 数据库 ☐ 网络 / 通信 ☐ Web 设计 ☐ 程序设计 / 软件开发 ☐ 图形图像与多媒体
☐ 单片机 / 嵌入式系统 ☐ 机电一体化 ☐ 自动化 ☐ 电子技术 ☐ 其他 _____

您对中国电力出版社计算机与自动化类图书印象最深的几本图书是：

您对本书的评价：

您认为此类图书的价格定位在多少合适？ _____

您最希望我们出版本方向哪些内容的图书？

您希望成为我们的作 / 译者吗？

您准备编写的图书名称： _____

您可以翻译的图书类型（从事的专业或研究方向） _____

您推荐引进出版的 _____

您的其他建议 _____

地址：北京市西城区三里河路6号中国电力出版社（100044）

电话：010-58383411 传真：010-58383267

E-mail: infopower@cepp.com.cn

敬请访问 www.infopower.com.cn

前 言

VxWorks 是美国风河 (WRS) 公司推出的面向嵌入式领域的强实时操作系统, 其集成开发环境为 Tornado, 随着发布版本的不断提高, 其功能越来越强大, 应用范围也越来越广, 目前已广泛应用于通信、银行、航空、航天、国防等领域, 已引起了众多嵌入式从业人员的关注。在基于 VxWorks 操作系统的应用与开发中, 硬件驱动程序开发和基于构件的分布式编程是目前研究的焦点。本书针对这两方面内容, 主要介绍了 VxWorks5.5 嵌入式实时操作系统下的驱动程序开发和分布式编程技术。通过将理论分析和工程开发的有机结合, 在说明基本原理的基础上, 给出了详尽的编程样例。

本书在内容上分为两大部分, 即硬件驱动篇和分布式编程篇。

硬件驱动篇主要介绍了字符设备驱动程序、增强型网络设备驱动程序(END)以及 WindML 中文字库的设计和 Microwindows 向 VxWorks 平台上的移植过程。字符设备驱动与 END 是两类最常见的设备驱动, 第 3 章和第 4 章首先介绍了两种类型驱动框架体系, 并分别以 i8250 和 ns83820 芯片为例, 给出了相应驱动的详细实现过程和代码; 第 5 章在分析 WindML BMF 字库结构的基础上, 给出了由 XWindows 的 BDF 字库转换得到 BMF 字库的代码实现; 图形是嵌入式操作系统的弱项, Microwindows 是嵌入式系统广泛使用的开源图形包, 第 6 章详细介绍了 Microwindows 向 VxWorks 平台的移植步骤, 并给出了几个底层驱动的实现。

分布式编程篇介绍了分布式构件对象模型 (DCOM) 和公共对象请求代理体系结构 (CORBA), 分别是由微软 (Microsoft) 和对象管理组织 (OMG) 推出的两种组件规范, 提供了 VxCOM/VxDCOM 的分布式组件对象模型, 同时第三方的软件开发商也提供了 CORBA, 用于解决分布式编程, 这其中包括宝兰 (Borland) 公司的 RT-VisiBorker、中创的 InforBuf 以及华盛顿大学开源的 TAO。

COM/DCOM、CORBA 是软件构件的两种不同规范, 它们是解决分布式系统编程的主要方法、嵌入式实时系统下的构件技术。随着结构化设计、面向对象技术和基于构件的软件开发应用的增多, 软件设计方法学的变化也改变着软件架构的设计, 促进软件构件在更高层次上的重用, 从源代码重用、二进制目标重用到架构重用, 构件的开发方便了软件的独立升级和并行开发。

本书第 1~5 章由孔祥营编写, 第 7~13 章由张保山编写, 俞烈彬编写了第 6 章并提供了关于 VxWorks 操作系统下 TAO 有关内容。希望本书的出版, 能够对有志于在嵌入式实时系统的领域进行探索的朋友有所借鉴和帮助。由于时间仓促和水平所限, 书中难免存在错误和疏漏, 敬请广大读者批评指正。

编 者
2007 年 3 月

目 录

前 言

第 1 章	建立开发环境	1
1.1	引导行详解	1
1.2	采用串口连接的开发调试环境	5
1.3	BootConfig 文件分析	8
1.4	VxWorks 可引导优盘制作	10
1.5	从文件中读引导行	15
1.6	VxWorks 与 Windows 系统文件互拷贝	17
第 2 章	驱动开发基础	21
2.1	硬件基础	21
2.2	VxWorks 设备驱动概述	32
2.3	VxWorks 与驱动相关的函数	34
第 3 章	字符驱动程序设计	40
3.1	i8250 芯片简介	40
3.2	非标串口驱动设计	45
3.3	字符设备 TTY 驱动设计	60
第 4 章	END 网络驱动设计	75
4.1	END 驱动概述	75
4.2	ns83820 芯片简介	81
4.3	ns83820 END 驱动实现	91
4.4	网络驱动程序调试	124
4.5	将新 END 驱动添加到 VxWorks 网络体系中	127
第 5 章	Zinc/WindML 本地化	130
5.1	WindML 字体字符显示原理	130
5.2	BMF 字体	132
5.3	汉字 BMF 字库构造	136
5.4	Zinc 中文字体支持	148
第 6 章	移植 Microwindows	149
6.1	Microwindows 介绍	149
6.2	建立一个基本的 Nano-X 程序	150
6.3	深入 Nano-X	158
6.4	Nano-X 在 VxWorks 下的实现	159
第 7 章	VxCOM/VxDCOM 软件开发	176
7.1	嵌入式实时系统软件开发的现状	176

7.2	VxDCOM 技术简介	180
7.3	Wind 对象模板库	181
7.4	创建 VxDCOM 应用程序	184
第 8 章	VxCOM/VxDCOM 客户程序设计	205
8.1	VxWorks COM/DCOM 库组成	205
8.2	COM 库组成	205
8.3	DCOM 库组成	209
8.4	客户端程序设计	211
第 9 章	VxCOM/VxDCOM 数据类型及接口	233
9.1	自动化数据类型	233
9.2	非自动化数据类型	244
9.3	HRESULT 说明	246
9.4	VxWorks 接口	248
9.5	虚函数表	253
第 10 章	VxDCOM 网络协议剖析	256
10.1	概述	256
10.2	RPC 网络通信报文分析	259
10.3	VxDCOM 网络协议分析实例	283
第 11 章	嵌入式实时系统 CORBA 技术	293
11.1	最小 CORBA (Minimum CORBA)	294
11.2	实时 CORBA (RT-CORBA)	311
11.3	VisiBroker-RT 编程说明	319
11.4	VxWorks 操作系统下的 TAO 技术	324
第 12 章	嵌入式系统 CORBA 编程技术	326
12.1	简介	326
12.2	命名服务例程	342
12.3	RT-CORBA 例程	349
第 13 章	GIOP 网络协议剖析	353
13.1	概述	353
13.2	GIOP 消息类型	354
13.3	基于 CORBA 规范的网络协议剖析	363
参考文献		372

第1章 建立开发环境

我们已经习惯了在 Windows、UNIX/Linux 等桌面操作系统下的软件开发，在这些环境下，有丰富的资源支持开发工作，CPU 足够快，存储器足够大，键盘鼠标配置齐全，各种软件开发工具足够友好，开发的程序同样运行在相同的平台（计算机、操作系统）之上。例如，使用 Microsoft Visual C++ 开发一个 Windows 程序，从编码、编译、连接、调试到发布，整个开发流程都可以在 Microsoft Visual Studio 中完成。

而嵌入式系统软件开发则不同，因为嵌入式系统大多数是应用于特定领域的定制系统，受系统资源限制，多数本身不具备提供运行开发工具的能力，因而需要采用主机目标机开发方式，即软件编辑和编译都连接在主机（通常是 Windows、Unix、Linux）上进行，由交叉编译工具生成的嵌入式软件目标码通过网口、串口等下载到目标机（待开发的嵌入式硬件系统）上。嵌入式软件运行在嵌入式硬件上，调试器运行在开发主机上，通过调试口控制嵌入式软件的运行，并获取相应信息，完成应用调试。这种开发方式称作主机目标机远程开发模式，如图 1-1 所示。

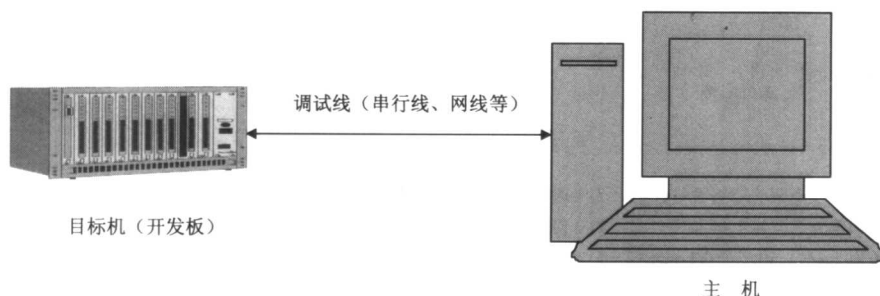


图 1-1 嵌入式系统软件开发方式示意图

VxWorks 是目前应用最多的嵌入式实时操作系统之一，广泛应用于工业控制、医疗器械、通信、航空航天以及国防电子等领域。VxWorks 软件开发也是采用主机目标机开发方式的。开发环境的建立是进行软件开发的第一步，本章介绍与 VxWorks 开发环境建立相关的问题，包括相关文件的介绍、串口下载调试方式、VxWorks 可引导优盘制作、VxWorks 与 Windows 文件互拷等内容。

1.1 引导行详解

VxWorks 支持多种引导下载方式，包括软盘、硬盘（ATA、SCSI）、DOC(DiskOnChip)、网络、串口等等，此外，通过修改相关文件还可以增加其他引导下载方式，如 NandFlash、优盘等等。这些引导下载方式配置定义的有关代码放在 BSP 目录下的 config.h 文件中，与网络之间的引导还涉及到 configNet.h 等文件，具体过程实现则在 target\all 目录下的 bootConfig.c 中。

config.h 中与引导有关的内容主要有：

- (1) 包含 configall.h 文件，configall.h 定义了 VxWorks 所有默认的设置。
- (2) 定义了引导行 (bootline)，引导行约定了引导设备、影像路径和网络 IP 地址等。
- (3) 文件系统及存储设备配置。
- (4) 定义了系统包含的网络设备 (END) 驱动程序。
- (5) 修订了 configall.h 文件中的一些设置，如硬件设备配置、WDB 调试方式等。

现以 X86 体系下 Pentium BSP 目录下的 config.h 为例，进行说明详细解释如图 1-2 所示。
该文件中，默认引导行为：

```
"fd=0,0(0,0)host:/fd0/vxWorks.st h=90.0.0.3 e=90.0.0.50 u=target"
```

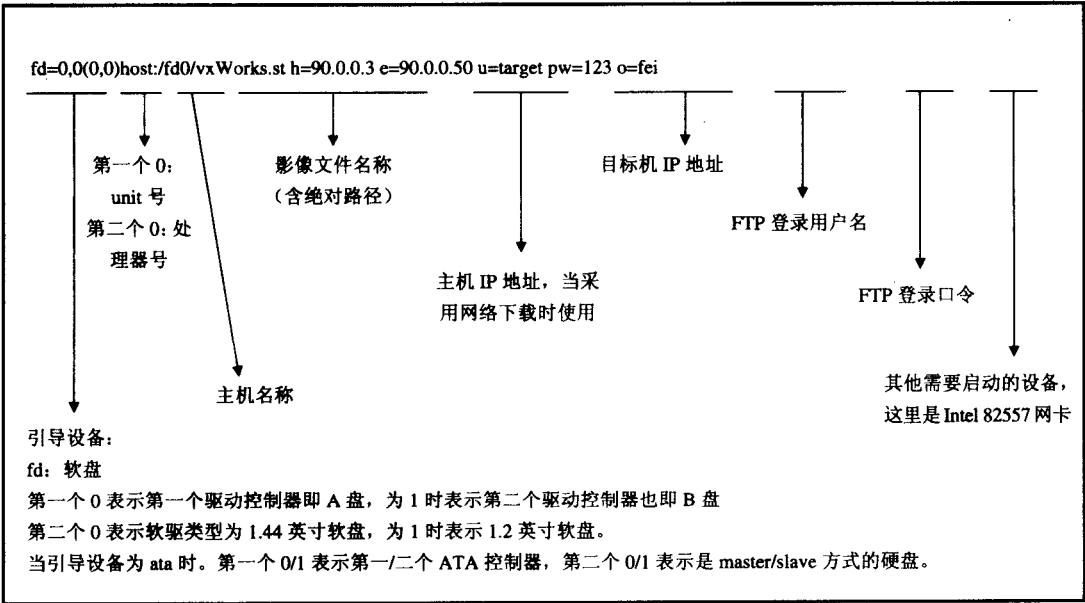


图 1-2 引导行详细解释

引导行中经常变化的是 host 之前的部分。当采用网络引导下载时，应为 nicName(0,0)，其中，nicName 为网卡设备的名称，常用网卡设备名称如表 1-1 所示，括号中的第一个"0"指明是系统中相同类型网卡的第 1 块（使用第二块应为"1"），第二个"0"是指处理器编号。采用网卡引导"时，影像文件名称绝对路径应为相应文件在主机 FTP 根目录下的相对路径，例如，如果主机 FTP 根目录为 D:\tornado\target\proj\project0\default\，VxWorks 影像即放在该目录下，那么，host:后应为 VxWorks。主机 IP 与目标机 IP 地址根据具体情况而定，二者必须在同一个子网段内。FTP 用户名与口令应该与 FTP server 分配的用户名和口令一致。

表 1-1 常用网卡设备名称

名称	设备宏定义	网卡芯片	驱动程序
fei	INCLUDE_FEI	Intel 82557/9 10/100 芯片	Fei82557end.c
elPci	INCLUDE_EL_3C90X_END	3COM 3c905B 芯片	El3c90xend.c

续表

名称	设备宏定义	网卡芯片	驱动程序
ln	INCLUDE_LN_97X_END	AMD 79C97x 芯片	Ln97xend.c
dc	INCLUDE_DEC21X40_END	Intel/DEC 21x4x 芯片	Dec21x40end.c
ene	INCLUDE_ENE_END	Ne2000 ISA 网卡	Ne2000end.c
elt	INCLUDE_ELT_3C509_END	3COM 3c509b ISA 网卡	elt3c509End.c

每种网卡驱动对应的引导设备名称可从其头文件中找到，这些头文件放在 `target\h\drv\end` 目录下，每个文件中通常有一个宏 `XXX_DRV_NAME` 来定义其设备名称，例如，在 `el3c90cend.h` 中，有如下定义：

```
#define EL3C90X_DEV_NAME    "elPci" /* device name */
```

当用相应设备引导时，必须在 `config.h` 中定义对应的宏，网卡如表 1-1 所示，其他设备如表 1-2 所示。

表 1-2 引导设备及宏定义

名称	设备宏定义	设备	影像文件名	备注
fd	INCLUDE_FD	软 盘	/fd0/vxWorks	还需要定义： INCLUDE_DOSFS
ata	INCLUDE_ATA	ATA 硬盘	/ide0/vxWorks	
ide	INCLUDE_IDE	IDE 硬盘	/ata0/vxWorks	
scsi	INCLUDE_SCSI	SCSI 硬盘	/sd0/vxWorks	
tffs	INCLUDE_TFFS	DOC (DiskOnChip)	/tffs0/vxWorks	
tsfs	INCLUDE_TSFS	串 口	/tgtsrv/vxWorks	见 1.2 节
pcmcia	INCLUDE_PCMCIA	PCMCIA 卡	/pcmcia0/vxWorks	

当系统中使用两块网卡时，如果是相同的网卡，则需要修改 `configNet.h` 文件，在数组 `endDevTbl` 中增加其定义。例如，当系统中有两块 `intel82557` 网卡时，需要增加以下内容：

```
#ifdef INCLUDE_FEI_END
    {0, FEI82557_LOAD_FUNC, FEI82557_LOAD_STRING, FEI82557_BUFF_LOAN,
     NULL, FALSE},
    {1, FEI82557_LOAD_FUNC, FEI82557_LOAD_STRING, FEI82557_BUFF_LOAN,
     NULL, FALSE},
#endif /* INCLUDE_FEI_END */
```

其中，斜体内容是针对第二块网卡而增加的，1 表示 unit 号，从 0 开始编号，1 即为第二块网卡。

如果系统中存在两块网卡，又需要通过其中一块进行调试（该网卡支持 WDB 协议），则必须把该块网卡的配置项移到 `endDevTbl` 数组的最前面，无论两块网卡是否相同。例如，在上面的例子中，如果采用第二块 `intel82557` 作为调试口，则需要改为如下形式：

```
#ifdef INCLUDE_FEI_END
    {1, FEI82557_LOAD_FUNC, FEI82557_LOAD_STRING, FEI82557_BUFF_LOAN,
    NULL, FALSE},
    {0, FEI82557_LOAD_FUNC, FEI82557_LOAD_STRING, FEI82557_BUFF_LOAN,
    NULL, FALSE},
#endif /* INCLUDE_FEI_END */
```

当采用不同网卡时，也要将调试网卡放在前面，例如当系统中同时存在 AMDLn97X 和 3COM3c905B 两种网卡时（各一块），configNet.h 中数组 endDevTbl 默认顺序为：

```
#ifdef INCLUDE_EL_3C90X_END
    {0, EL_3C90X_LOAD_FUNC, EL_3C90X_LOAD_STR, EL_3C90X_BUFF_LOAN,
    NULL, FALSE},
#endif /* INCLUDE_EL_3C90X_END */

#ifdef INCLUDE_LN_97X_END
    {0, LN_97X_LOAD_FUNC, LN_97X_LOAD_STR, LN_97X_BUFF_LOAN,
    NULL, FALSE},
#endif /* INCLUDE_LN_97X_END */
```

当使用 AMDLn97X 作为调试口，应改为：

```
#ifdef INCLUDE_LN_97X_END
    {0, LN_97X_LOAD_FUNC, LN_97X_LOAD_STR, LN_97X_BUFF_LOAN,
    NULL, FALSE},
#endif /* INCLUDE_LN_97X_END */

#ifdef INCLUDE_EL_3C90X_END
    {0, EL_3C90X_LOAD_FUNC, EL_3C90X_LOAD_STR, EL_3C90X_BUFF_LOAN,
    NULL, FALSE},
#endif /* INCLUDE_EL_3C90X_END */
```

引导行中，影像文件后的部分可根据情况删减，并且先后次序也可调整，每一项的意义由等号前面的字符确定，例如，h 表示等号后的是主机 IP 地址；e 则表示等号后的是目标机 IP 地址；u 表示 FTP 用户；pw 表示 FTP 口令；o 表示还需要启动的设备。除上述几个选项外，还有 tn、s 和 f 经常用到。

- tn=后可跟目标机名称，系统启动后可用 hostShow 看到。
- s=后通常是一个文本文件名(含路径)，该文件是一个 shell 可识别的文本文件。VxWorks 启动后，targetShell 将打开该文件，逐行解释并执行。下面是一个简单脚本文件的例子。

```
ifMaskSet("fei0", 0xffffffff00);
ifAddrSet("fei0", "192.0.1.12");
ld </ata0/app.out
sp appInit
```

该脚本文件首先设置 IP 掩码和 IP 地址，然后从硬盘中加载应用程序 app.out 文件，最后发起一个任务执行应用程序入口函数 appInit。在上面的例子中，采用脚本文件以及动态加载的方式，避免应用程序与 VxWorks 内核连接在一起，有利于应用程序单独升级，可大大提高系统配置的灵活性。

- f=后跟系统启动标示，是一个双字节十六进制整数，定义如表 1-3 所示。

表 1-3 引导标示定义

flag 位	为 1 时的意义
0	不作为系统控制器
1	加载本地系统符号
2	不自动启动
3	快速自动启动，不作数字递减
5	禁止登录安全
6	如果定义了 DHCP，则使用 dhcp 得到启动参数；如果定义了 BOOTP，则使用 bootp 得到启动参数
7	使用 tftp 得到引导影像
8	使用代理 ARP

1.2 采用串口连接的开发调试环境

很多驱动软件的开发是在不具备网络的环境下进行的，包括网络驱动的调试。在这种情形下，由于串口（驱动）相对简单，通常采用串口作为调试口。本节介绍如何建立以串行口作为调试口的过程。以 X86 平台为例。

准备工作：除了开发主机（已安装 Tornado）外，还要一台有软驱的 PC 机，至少有一个串口，串行电缆，空白软盘。如果没有 PC 机也没关系，就可采用 VMWARE 软件在开发主机上虚拟一台 PC 机，要求开发主机至少要有两个串口，一个分配给 VMWARE；一个作为 target Server 文件系统的输出口。

（1）修改引导行。

前面已经介绍过，采用串口作为引导下载设备时，引导行设备名称为“tsfs”，当主机 target Server 文件系统设置为 VxWorks 影像所在路径时，引导行如下：

```
"tsfs(0,0)host:vxWorks"
```

按照上面的引导行形式修改 BSP 目录下的 config.h 文件中的 DEFAULT_BOOT_LINE 定义：

```
#define DEFAULT_BOOT_LINE "tsfs(0,0)host:vxWorks"
```

（2）向 config.h 增加以下内容。

```
#define INCLUDE_SERIAL
#undef CONSOLE_TTY
#define CONSOLE_TTY NONE
#undef WDB_TTY_CHANNEL
#define WDB_TTY_CHANNEL 0 /* 使用第一个串口进行调试 */
#undef WDB_COMM_TYPE
#define WDB_COMM_TYPE WDB_COMM_SERIAL /* 使用串口作为调试口 */
```

```
#undef      WDB_TTY_BAUD
#define      WDB_TTY_BAUD      115200      /* 串口通信波特率, 建议采用 38400bps */
#define      INCLUDE_TSFS_BOOT      /* 使用第一个串口下载影像 */
```

(3) 修改 bootconfig.c 文件。

找到以下内容:

```
#define      INCLUDE_TSFS_BOOT_VIO_CONSOLE /* needed for Target Server Console */
```

改为:

```
#undef      INCLUDE_TSFS_BOOT_VIO_CONSOLE /* needed for Target Server Console */
```

(4) 重新编译该 BSP 的 bootrom 或 bootrom_uncmp。

(5) 用 mkboot.bat 制作引导盘。

(6) 用串行线物理连接主机与目标机串口。

(7) 配置一个可引导影像工程, 其中 VxWorks 配置如图 1-3 所示。

WDB agent connection

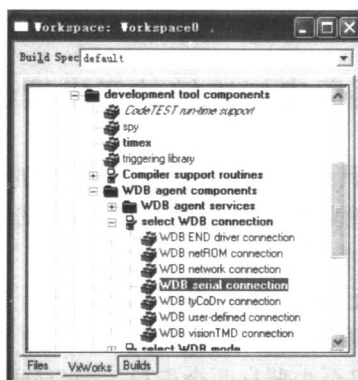


图 1-3 VxWorks 影像 WDB 配置

在 WDB serial connection 上单击鼠标右键, 修改其参数值, 保持与主机设置一致, 如图 1-4 所示。

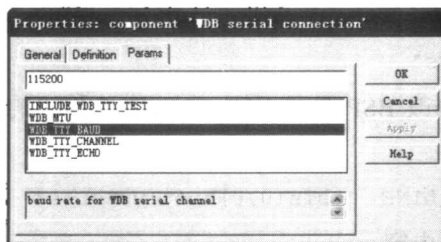


图 1-4 串口 WDB 参数

WDB_TTY_BAUD 为 115200bps

WDB_TTY_CHANNEL 为 0

(8) 设置主机 target Server。

在 Tornado 环境工具栏中, 依次点击 tools、target Server、configure, 新建一个连接配置, 如图 1-5 所示。

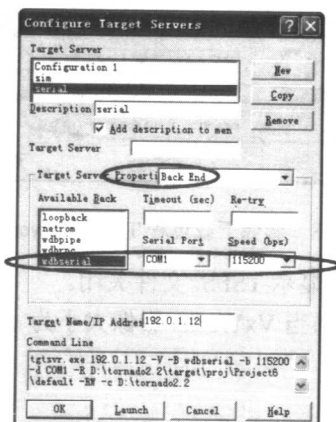


图 1-5 Target Server 串口连接配置图一

图中，Back End 方式选用 wdbserial，使用主机第一个串口（COM1），波特率为 115200bps（与目标机一致）。然后在 Target Server Project 下拉选框中，选中 Target Server File System，设置其根路径为 VxWorks 影像所在目录，并允许读写，如图 1-6 所示。最后点击 OK 保存。

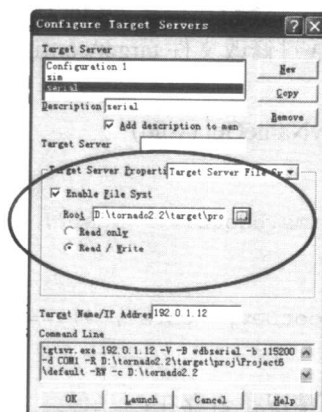


图 1-6 Target Server 串口连接配置图二

(9) 用制作的软盘启动目标机，并启动上面配置的 Target Server（在主机 Tornado 中）。目标机启动在自动计数结束后屏幕显示如图 1-7 所示。

```
Booting using TSFS...
Make sure that your Target Server is started with -R[oot] option.
Waiting for Target Server connection... Done

Loading /tgtsvr/vxWorks...
857696
```

图 1-7 串口引导目标机下载过程示意图

最后一行数字是 VxWorks 影像大小，不同配置的影像大小也不同。Target Server 窗口显示内容如图 1-8 所示。

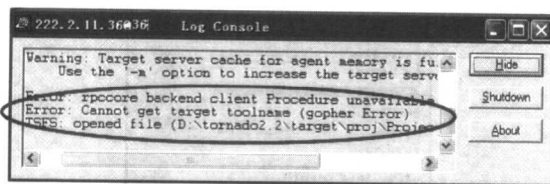


图 1-8 影像下载过程 Target Server 截图

当下载结束时，target server 显示 TSFS 文件关闭。

下载时间可以初步估算，例如当 VxWorks 影像大小为 size，串口波特率为 baud 时，那么下载传输时间 t （单位秒）约为：

$$t = \text{size} * 10 / \text{baud}$$

上述例子中，VxWorks 的大小为 2543328，波特率为 115200，传输时间约为：

$$857696 * 10 / 115200 = 75 \text{ 秒}$$

大约 2min，由于文件传输还有一些控制信息，因此，实际时间要大于计算时间。

上述配置中，PC CONSOLE 和网络都可以保留，便于信息浏览和调试网络驱动。当保留网络时，VxWorks 启动后可能会提示一些错误信息，是因为它错误地把引导设备 tsfs 作为 END 设备启动但不成功造成，这时可以通过修改文件 target\config\comps\src\net\usrNetBoot.c 来除掉这一错误。

(1) 找到函数：void usrNetDevNameGet (void)

(2) 在语句：

```
if ( (strcmp (sysBootParams.bootDev, "scsi", 4) == 0) ||
```

之后增加：

```
(strcmp (sysBootParams.bootDev, "tsfs", 4) == 0) ||
```

即可。

当连接不成功时，应考虑以下情况是否存在：

- (1) 主机方 target server 是否启动？bootrom 启动后，会等待主机 target server 启动。
- (2) 双方波特率是否一致？系统默认连接时使用 9600 bps 或 38400bps。
- (3) 波特率是否过高？降低波特率到 38400bps，波特率过高增加了传输出错概率。低波特率虽然慢些，但稳定可靠。115200b/s 传输虽然快但对线路质量要求较高，容易出错，可能会导致欲速则不达，建议使用 38400bps。
- (4) 串行线是否正确？可以用两台 PC 机超级终端进行线路通信验证。

1.3 BootConfig 文件分析

嵌入式系统各不相同，在有些情况下可能需要从特殊设备引导加载 VxWorks，例如希望能从优盘加载 VxWorks，或系统使用 NAND/NOR FLASH 作为存储器，这两种 FLASH 通常不使用 DOSFS，而 VxWorks 默认配置中不支持引导这些方式，就需要修改相应文件达到目的。

包含引导设备的选择与初始化代码的文件 bootconfig.c 是以源码方式提供的，放在

target\config\all 目录下,通过分析该文件,可以找到增加新的引导设备的方法。

Bootconfig.c 文件与引导设备相关的部分放在函数 bootLoad 中。该函数以引导行和影像入口点地址作为参数。bootLoad 函数根据引导设备名称,调用相应的加载函数,如配置串口,且引导设备名称为“tsfs”时,调用 tsfsLoad 完成影像加载。

以下以系统配置引导设备为软盘 fd 为例,分析 bootLoad 的流程,ata、ide 等设备的情况与此类似,如图 1-9 所示。

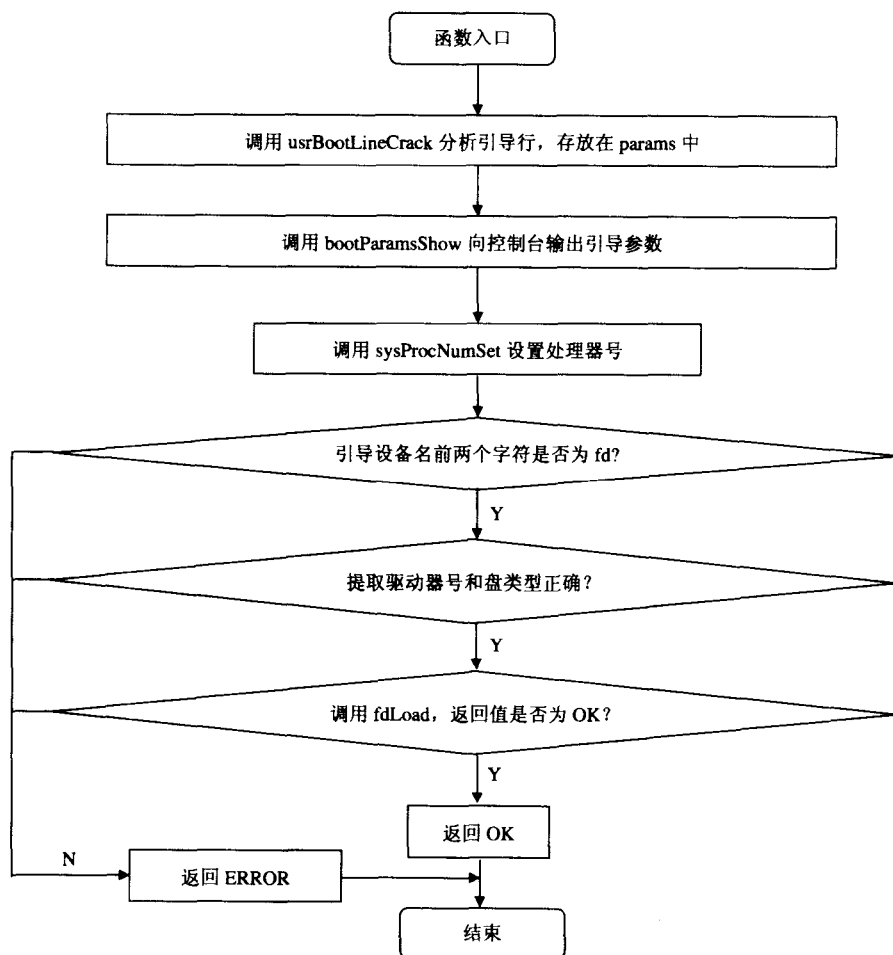


图 1-9 bootLoad 流程图

fdLoad 函数流程如图 1-10 所示。

从上述流程可知,从一个引导设备加载影像,实际上是根据引导设备名称判明是哪种类型设备,然后初始化相应的设备,如果是存储类,则用文件操作,调用 open 打开存放在相应设备上的影像文件,并返回文件句柄;如果是网络设备,则通过 FTP 或 TFTP 操作,同样也能获得文件句柄。将该文件句柄传递给 bootLoadModule 函数,bootLoadModule 负责将影像文件加载到系统内存(分析影像文件格式,将文件的 text、data、bss 等段放置到内存指定的位置)。

因此,当增加一种新的引导设备(非网络设备)时,可以仿照 fd 设备的引导流程,修改