

## VxWorks 实时操作系统及其在 PC104 下以太网编程的应用

李军峰

(中国电子科技集团公司第二十研究所 西安 710068)

**摘 要** 本文简单介绍了嵌入式操作系统 VxWorks 的特点, 以及 VxWorks 在 PC104 下基于以太网编程的思路。

**关键词** VxWorks; TCP/IP; UDP; PC104

### 1 前 言

VxWorks 操作系统是美国 WindRiver 公司于 1983 年设计开发的一种嵌入式实时操作系统 (RTOS)。它具有可裁减的微内核结构、高效的任务管理、灵活的任务间通讯及先进的网络功能, 其良好的可靠性和卓越的实时性被广泛地应用在通信、军事、航空、航天等高精尖技术及实时性要求极高的领域中。VxWorks 的网络结构与 OSI 参考模型相比是在 IP 层和链路层之间有一 Mux 层, Mux 层屏蔽了网络层和数据链路的直接交互, VxWorks 通过 END (Enhanced Network Drivers) 实现 IP 层与链路层之间的数据交互。END 向上通过 Mux 层与 IP 层相连, 向下与链路层直接相连, 从而在 IP 层与链路层之间建立一数据传输通道。

### 2 VxWorks 的特点

#### 2.1 可靠性

操作系统的用户希望在一个工作稳定, 可以信赖的环境中工作, 所以操作系统的可靠性是用户首先要考虑的问题。而稳定、可靠一直是 VxWorks 的一个突出优点。自从对中国的销售解禁以来, VxWorks 以其良好的可靠性在中国赢得了越来越多的用户。

#### 2.2 实时性

实时性是指能够在限定时间内执行完规定的功能并对外部的异步事件做出响应的能力。实时性的强弱是以完成规定功能和做出响应时间的长短来衡量的。

VxWorks 的实时性做得非常好, 其系统本身的开销很小, 进程调度、进程间通信、中断处理等系统公用程序精练而有效, 它们造成的延迟很短。VxWorks 提供的多任务机制中对任务的控制采用了优先级抢占 (Preemptive Priority Scheduling) 和轮转调度 (Round-Robin Scheduling) 机制, 也充分保证了可靠的实时性, 使同样的硬件配置能满足更强的实时性要

求，为应用的开发留下更大的余地。

## 2.3 可裁减性

用户在使用操作系统时，并不是操作系统中的每一个部件都要用到。例如图形显示、文件系统以及一些设备驱动在某些嵌入系统中往往并不使用。VxWorks 由一个体积很小的内核及一些可以根据需要进行定制的系统模块组成。VxWorks 内核最小为 3kB，即便加上其它必要模块，所占用的空间也很小，且不失其实时、多任务的系统特征。由于它的高度灵活性，用户可以很容易地对这一操作系统进行定制或作适当开发，来满足自己的实际应用需要。

## 3 VxWorks 在 PC104 下基于以太网通信的应用

此论述中目标机选用 lippert 公司生产的 PC104，网口芯片型号为 DP83815。主机与目标机之间采用以太网连接调试方式。在编写程序之前先需要根据硬件厂商提供的网口驱动程序包配置好网口驱动。

VxWorks 下的以太网通信采用 TCP/IP 协议，该协议基于网络互连的基本思想，它要求应用软件和通信软件分开。顾名思义，TCP/IP 得名于两个有名的协议，即 TCP (Transmission Control Protocol) 协议和 IP (Internet Protocol) 协议。TCP 是可靠的传输层协议，它以位于其下层的 IP 协议为基础，并利用 IP 提供的路由功能。传输层协议包含两个协议：

传输控制协议 (TCP)

数据报协议 (UDP)

以下是用户程序的设计方法，在程序设计前，首先确定系统的工作方式，再选择相应的通信协议。

### 3.1 TCP 协议方式

系统工作方式为客户机、服务器模式，点对点通信，面向连接，数据以字节流的形式传输。

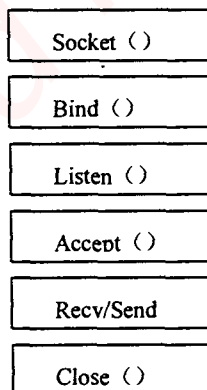


图1 服务器程序设计框图

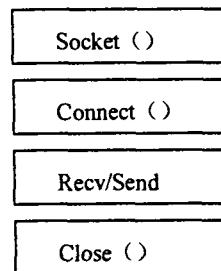


图2 客户机程序设计框图

服务器程序设计框图如图 1 所示，客户机程序框图如图 2 所示。

### 3.2 UDP 协议方式

UDP 协议方式是面向无连接的服务，应用程序传递的是数据块，适用于广播方式。

UDP 协议方式下服务器程序设计框图如图 3 所示，客户机程序框图如图 4 所示。

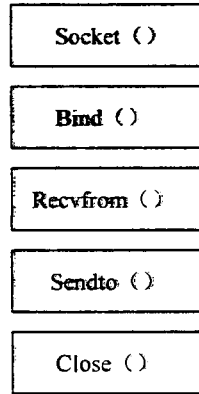


图 3 服务器程序设计框图

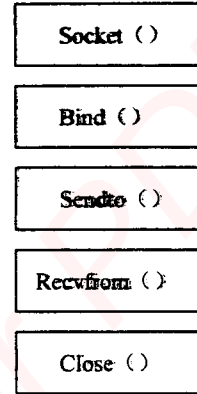


图 4 客户机程序设计框图

其中 socket 用来建立一个通信的端点并返回一个 socket 描述符。Bind 调用将一个名字绑定到一个 socket 上，为一个未命名的 socket 分配一个名字。Listen 调用在一个 socket 侦听连接并指定接受连接的队列长度限制。Accept 在指定的 socket 上接受一个连接请求。Send 和 sendto 用来从一个 socket 发送一个消息，其中 send 调用只用于处于连接状态的 socket，而 sendto 不需要建立连接。recv 和 recvfrom 用来从一个 socket 接收一个消息，同样 recv 调用只用于处于连接状态的 socket，而 recvfrom 不需要建立连接。

#### UDP 协议下广播方式通信的实现

UDP 广播模式的用途之一是假定服务器主机在本地子网上，但不知道它的单播 IP 地址时，对它进行定位，这就是资源发现 (resource discovery)；另一用途是当有多个客户和单个服务器通信时，减少局域网上数据流量，后一种应用较多。

广播地址的类型为子网广播地址，这类地址编排指定子网上的所有接口。例如，如果我们对地址 193.9 采用 8 位子网 ID，那么 193.9.207.255 将是 193.9.207 的子网上所有接口的子网广播地址。

服务器除设定广播地址外，还需要设置广播参数；客户端如下所示设置为广播方式，绑定服务器端口号就可以通信了。

```
sock.sin_family=AF_INET;
sock.sin_port=htons (base_port);           //服务器端口号
sock.sin_addr.s_addr=htonl (INADDR_ANY);   //广播地址
if (setsockopt (socket1,SOL_SOCKET,SO_BROADCAST, (char *) &on,sizeof (on))
== -1)
{
```

```
printf ("\n Error in initialize socket!");  
return (0);  
} //设置广播方式
```

以太网程序要执行还必须配置 IP 到链路层的接口，配置包括以下几步：

将 IP 绑定到 MUX；

设置接口的网络掩码；

给每一个接口分配一个 IP 地址；

分配广播地址。

引导接口默认将 IP 绑定到 MUX。如果目标机只有一个网络接口设备，可以在引导行设置接口网络掩码和地址（附加接口在引导行之后还要求进行明确的配置）。分配网络掩码还要设置广播地址：主机为全 1 的 IP 地址。

配置可以通过手动或自动来实现，手动配置即在目标机 shell 命令行下分别调用 ipAttach ()、ifMaskSet、ifAddrSet ()；若要在目标机上自动运行程序，则需在 usrAppInit.c 文件中加上以下语句：

```
ipAttach (0,"dp"); //dp for 83815*/  
ifMaskSet ("dp0",0xFFFFFFFF);  
ifAddrSet ("dp0","193.9.207.200");  
progStart ();
```

这样加电引导 VxWorks 之后就会自动加载以太网网口配置，并自动运行程序。

## 4 结束语

本论述简单介绍了 VxWorks 环境下以太网编程的思路，由于作者水平及文章篇幅的限制，VxWorks 许多内容并未涉及，在此只是为了抛砖引玉，希望大家能够一起学好 VxWorks 操作系统。

### 参考文献

- 1 王金刚等著. VxWorks 程序员指南. 北京：清华大学出版社. 2003.4
- 2 王金刚等著. VxWorks 网络程序员指南. 北京：清华大学出版社. 2003.4
- 3 孔祥营，柏桂枝著. 嵌入式实时操作系统 VxWorks 及其开发环境 Tornado. 中国电力出版社. 2002.3

# 嵌入式资源免费下载

## 总线协议:

1. [基于 PCIe 驱动程序的数据传输卡 DMA 传输](#)
2. [基于 PCIe 总线协议的设备驱动开发](#)
3. [CANopen 协议介绍](#)
4. [基于 PXI 总线 RS422 数据通信卡 WDM 驱动程序设计](#)
5. [FPGA 实现 PCIe 总线 DMA 设计](#)
6. [PCI Express 协议实现与验证](#)
7. [VPX 总线技术及其实现](#)
8. [基于 Xilinx FPGA 的 PCIE 接口实现](#)
9. [基于 PCI 总线的 GPS 授时卡设计](#)
10. [基于 CPCI 标准的 6U 信号处理平台的设计](#)
11. [USB30 电路保护](#)
12. [USB30 协议分析与框架设计](#)
13. [USB 30 中的 CRC 校验原理及实现](#)
14. [基于 CPLD 的 UART 设计](#)
15. [IPMI 在 VPX 系统中的应用与设计](#)
16. [基于 CPCI 总线的 PMC 载板设计](#)
17. [基于 VPX 总线的工件台运动控制系统研究与开发](#)
18. [PCI Express 流控机制的研究与实现](#)
19. [UART16C554 的设计](#)
20. [基于 VPX 的高性能计算机设计](#)
21. [基于 CAN 总线技术的嵌入式网关设计](#)
22. [Visual C 串行通讯控件使用方法与技巧的研究](#)
23. [IEEE1588 精密时钟同步关键技术研究](#)
24. [GPS 信号发生器射频模块的一种实现方案](#)
25. [基于 CPCI 接口的视频采集卡的设计](#)
26. [基于 VPX 的 3U 信号处理平台的设计](#)
27. [基于 PCI Express 总线 1394b 网络传输系统 WDM 驱动设计](#)
28. [AT89C52 单片机与 ARINC429 航空总线接口设计](#)
29. [基于 CPCI 总线多 DSP 系统的高速主机接口设计](#)
30. [总线协议中的 CRC 及其在 SATA 通信技术中的应用](#)
31. [基于 FPGA 的 SATA 硬盘加解密控制器设计](#)
32. [Modbus 协议在串口通讯中的研究及应用](#)
33. [高可用的磁盘阵列 Cache 的设计和实现](#)
34. [RAID 阵列中高速 Cache 管理的优化](#)

35. [一种新的基于 RAID 的 CACHE 技术研究与实现](#)
36. [基于 PCIE-104 总线的高速数据接口设计](#)
37. [基于 VPX 标准的 RapidIO 交换和 Flash 存储模块设计](#)
38. [北斗卫星系统在海洋工程中的应用](#)
39. [北斗卫星系统在远洋船舶上应用的研究](#)
40. [基于 CPCI 总线的红外实时信号处理系统](#)
41. [硬件实现 RAID 与软件实现 RAID 的比较](#)
42. [基于 PCI Express 总线系统的热插拔设计](#)
43. [基于 RAID5 的磁盘阵列 Cache 的研究与实现](#)
44. [基于 PCI 总线的 MPEG2 码流播放卡驱动程序开发](#)
45. [基于磁盘阵列引擎的 RAID5 小写性能优化](#)
46. [基于 IEEE1588 的时钟同步技术研究](#)
47. [基于 Davinci 平台的 SD 卡读写优化](#)
48. [基于 PCI 总线的图像处理及传输系统的设计](#)

## VxWorks:

1. [基于 VxWorks 的多任务程序设计](#)
2. [基于 VxWorks 的数据采集存储装置设计](#)
3. [Flash 文件系统分析及其在 VxWorks 中的实现](#)
4. [VxWorks 多任务编程中的异常研究](#)
5. [VxWorks 应用技巧两例](#)
6. [一种基于 VxWorks 的飞行仿真实时管理系统](#)
7. [在 VxWorks 系统中使用 TrueType 字库](#)
8. [基于 FreeType 的 VxWorks 中文显示方案](#)
9. [基于 Tilcon 的 VxWorks 简单动画开发](#)
10. [基于 Tilcon 的某武器显控系统界面设计](#)
11. [基于 Tilcon 的综合导航信息处理装置界面设计](#)
12. [VxWorks 的内存配置和管理](#)
13. [基于 VxWorks 系统的 PCI 配置与应用](#)
14. [基于 MPC8270 的 VxWorks BSP 的移植](#)
15. [Bootrom 功能改进经验谈](#)
16. [基于 VxWorks 嵌入式系统的中文平台研究与实现](#)
17. [VxBus 的 A429 接口驱动](#)
18. [基于 VxBus 和 MPC8569E 千兆网驱动开发和实现](#)
19. [一种基于 vxBus 的 PPC 与 FPGA 高速互联的驱动设计方法](#)
20. [基于 VxBus 的设备驱动开发](#)
21. [基于 VxBus 的驱动程序架构分析](#)
22. [基于 VxBus 的高速数据采集卡驱动程序开发](#)



23. [Vxworks 下的冗余 CAN 通讯模块设计](#)
24. [WindML 工业平台下开发 S1d13506 驱动及显示功能的实现](#)
25. [WindML 中 Mesa 的应用](#)
26. [VxWorks 下图形用户界面开发中双缓冲技术应用](#)
27. [VxWorks 上的一种 GUI 系统的设计与实现](#)
28. [VxWorks 环境下 socket 的实现](#)
29. [VxWorks 的 WindML 图形界面程序的框架分析](#)

## Linux:

1. [Linux 程序设计第三版及源代码](#)
2. [NAND FLASH 文件系统的设计与实现](#)
3. [多通道串行通信设备的 Linux 驱动程序实现](#)
4. [Zsh 开发指南-数组](#)
5. [常用 GDB 命令中文速览](#)
6. [嵌入式 C 进阶之道](#)
7. [Linux 串口编程实例](#)
8. [基于 Yocto Project 的嵌入式应用设计](#)
9. [Android 应用的反编译](#)
10. [基于 Android 行为的加密应用系统研究](#)
11. [嵌入式 Linux 系统移植步步通](#)
12. [嵌入式 C++ 语言精华文章集锦](#)
13. [基于 Linux 的高性能服务器端的设计与研究](#)
14. [S3C6410 移植 Android 内核](#)
15. [Android 开发指南中文版](#)
16. [图解 Linux 操作系统架构设计与实现原理（第二版）](#)
17. [如何在 Ubuntu 和 Linux Mint 下轻松升级 Linux 内核](#)
18. [Android 简单 mp3 播放器源码](#)
19. [嵌入式 Linux 系统实时性的研究](#)
20. [Android 嵌入式系统架构及内核浅析](#)
21. [基于嵌入式 Linux 操作系统内核实时性的改进方法研究](#)
22. [Linux TCP IP 协议详解](#)
23. [Linux 桌面环境下内存去重技术的研究与实现](#)
24. [掌握 Android 7.0 新增特性 Quick Settings](#)
25. [Android 应用逆向分析方法研究](#)
26. [Android 操作系统的课程教学](#)
27. [Android 智能手机操作系统的研究](#)
28. [Android 英文朗读功能的实现](#)
29. [基于 Yocto 订制嵌入式 Linux 发行版](#)

30. [基于嵌入式 Linux 的网络设备驱动设计与实现](#)
31. [如何高效学习嵌入式](#)
32. [基于 Android 平台的 GPS 定位系统的设计与实现](#)
33. [LINUX ARM 下的 USB 驱动开发](#)
34. [Linux 下基于 I2C 协议的 RTC 驱动开发](#)
35. [嵌入式下 Linux 系统设备驱动程序的开发](#)
36. [基于嵌入式 Linux 的 SD 卡驱动程序的设计与实现](#)

## Windows CE:

1. [Windows CE.NET 下 YAFFS 文件系统 NAND Flash 驱动程序设计](#)
2. [Windows CE 的 CAN 总线驱动程序设计](#)
3. [基于 Windows CE.NET 的 ADC 驱动程序实现与应用的研究](#)
4. [基于 Windows CE.NET 平台的串行通信实现](#)
5. [基于 Windows CE.NET 下的 GPRS 模块的研究与开发](#)
6. [win2k 下 NTFS 分区用 ntldr 加载进 dos 源代码](#)
7. [Windows 下的 USB 设备驱动程序开发](#)
8. [WinCE 的大容量程控数据传输解决方案设计](#)
9. [WinCE6.0 安装开发详解](#)
10. [DOS 下仿 Windows 的自带计算器程序 C 源码](#)
11. [G726 局域网语音通话程序和源代码](#)
12. [WinCE 主板加载第三方驱动程序的方法](#)
13. [WinCE 下的注册表编辑程序和源代码](#)
14. [WinCE 串口通信源代码](#)
15. [WINCE 的 SD 卡程序\[可实现读写的源码\]](#)
16. [基于 WinCE 的 BootLoader 研究](#)
17. [Windows CE 环境下无线网卡的自动安装](#)
18. [基于 Windows CE 的可视电话的研究与实现](#)
19. [基于 WinCE 的嵌入式图像采集系统设计](#)
20. [基于 ARM 与 WinCE 的掌纹鉴别系统](#)

## PowerPC:

1. [Freescale MPC8536 开发板原理图](#)
2. [基于 MPC8548E 的固件设计](#)
3. [基于 MPC8548E 的嵌入式数据处理系统设计](#)



4. [基于 PowerPC 嵌入式网络通信平台的实现](#)
5. [PowerPC 在车辆显控系统中的应用](#)
6. [基于 PowerPC 的单板计算机的设计](#)
7. [用 PowerPC860 实现 FPGA 配置](#)
8. [基于 MPC8247 嵌入式电力交换系统的设计与实现](#)
9. [基于设备树的 MPC8247 嵌入式 Linux 系统开发](#)
10. [基于 MPC8313E 嵌入式系统 UBoot 的移植](#)
11. [基于 PowerPC 处理器 SMP 系统的 UBoot 移植](#)
12. [基于 PowerPC 双核处理器嵌入式系统 UBoot 移植](#)
13. [基于 PowerPC 的雷达通用处理机设计](#)
14. [PowerPC 平台引导加载程序的移植](#)
15. [基于 PowerPC 嵌入式内核的多串口通信扩展设计](#)
16. [基于 PowerPC 的多网口系统抗干扰设计](#)
17. [基于 MPC860T 与 VxWorks 的图形界面设计](#)

## ARM:

1. [基于 DiskOnChip 2000 的驱动程序设计及应用](#)
2. [基于 ARM 体系的 PC-104 总线设计](#)
3. [基于 ARM 的嵌入式系统中断处理机制研究](#)
4. [设计 ARM 的中断处理](#)
5. [基于 ARM 的数据采集系统并行总线的驱动设计](#)
6. [S3C2410 下的 TFT LCD 驱动源码](#)
7. [STM32 SD 卡移植 FATFS 文件系统源码](#)
8. [STM32 ADC 多通道源码](#)
9. [ARM Linux 在 EP7312 上的移植](#)
10. [ARM 经典 300 问](#)
11. [基于 S5PV210 的频谱监测设备嵌入式系统设计与实现](#)
12. [Uboot 中 start.S 源码的指令级的详尽解析](#)
13. [基于 ARM9 的嵌入式 Zigbee 网关设计与实现](#)
14. [基于 S3C6410 处理器的嵌入式 Linux 系统移植](#)
15. [CortexA8 平台的  \$\mu\$ C-OS II 及 LwIP 协议栈的移植与实现](#)
16. [基于 ARM 的嵌入式 Linux 无线网卡设备驱动设计](#)
17. [ARM S3C2440 Linux ADC 驱动](#)
18. [ARM S3C2440 Linux 触摸屏驱动](#)
19. [Linux 和 Cortex-A8 的视频处理及数字微波传输系统设计](#)
20. [Nand Flash 启动模式下的 Uboot 移植](#)
21. [基于 ARM 处理器的 UART 设计](#)

22. [ARM CortexM3 处理器故障的分析与处理](#)
23. [ARM 微处理器启动和调试浅析](#)
24. [基于 ARM 系统下映像文件的执行与中断运行机制的实现](#)
25. [中断调用方式的 ARM 二次开发接口设计](#)

## Hardware:

1. [DSP 电源的典型设计](#)
2. [高频脉冲电源设计](#)
3. [电源的综合保护设计](#)
4. [任意波形电源的设计](#)
5. [高速 PCB 信号完整性分析及应用](#)
6. [DM642 高速图像采集系统的电磁干扰设计](#)
7. [使用 COMExpress Nano 工控板实现 IP 调度设备](#)
8. [基于 COM Express 架构的数据记录仪的设计与实现](#)
9. [基于 COM Express 的信号系统逻辑运算单元设计](#)
10. [基于 COM Express 的回波预处理模块设计](#)
11. [基于 X86 平台的简单多任务内核的分析与实现](#)
12. [基于 UEFI Shell 的 PreOS Application 的开发与研究](#)
13. [基于 UEFI 固件的恶意代码防范技术研究](#)
14. [MIPS 架构计算机平台的支持固件研究](#)
15. [基于 UEFI 固件的攻击验证技术研究](#)
16. [基于 UEFI 的 Application 和 Driver 的分析与开发](#)
17. [基于 UEFI 的可信 BIOS 研究与实现](#)
18. [基于 UEFI 的国产计算机平台 BIOS 研究](#)
19. [基于 UEFI 的安全模块设计分析](#)
20. [基于 FPGA Nios II 的等精度频率计设计](#)
21. [基于 FPGA 的 SOPC 设计](#)
22. [基于 SOPC 基本信号产生器的设计与实现](#)

## Programming:

1. [计算机软件基础数据结构 - 算法](#)
2. [高级数据结构对算法的优化](#)
3. [零基础学算法](#)
4. [Linux 环境下基于 TCP 的 Socket 编程浅析](#)

RT Embedded <http://www.kontronn.com>

5. [Linux 环境下基于 UDP 的 socket 编程浅析](#)
6. [基于 Socket 的网络编程技术及其实现](#)