

VxWorks概述

周扬帆

(珠海同方爱德科技有限公司 广东 珠海 519000)

【摘要】 本文主要论述业界极为出名的嵌入式操作系统 VxWorks 的基本特点和优势,还有它的基本结构和组成部件情况,以及它在计算机和通信领域中比较常用的网络系统。

【关键字】 VxWorks;Tornado;开发调试环境;网络系统

VxWorks是美国 Wind River 公司的一种嵌入式实时操作系统(RTOS),自上世纪八十年代初开始涉足商业化嵌入式实时操作系统市场以来,依据其良好的持续发展能力、高性能的内核及友好的用户开发环境,在嵌入式实时操作系统领域逐渐占据一席之地,它以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,如卫星通讯、军事演习、弹道制导、飞机导航等。在美国的 F-16、FA-18 战斗机、B-2 隐形轰炸机和爱国者导弹上,甚至连火星探测器上也使用到了 VxWorks。主要商业用户包括 Cisco systems, Bay Networks, 3Com, Fore systems, HP, Lucent, Qualcomm 等著名公司。

1 VxWorks 的基本特点及优势

VxWorks 在这一领域得以立足,得益于它的下列几个特点。

1.1 具有一个高性能的操作系统内核 Wind

VxWorks 的内核 Wind 是一个具有较高性能的、标准的嵌入式实时操作系统内核,主要特点包括快速多任务切换、可抢占式任务调度、任务间通信手段多样化等。该内核具任务间切换时间短、中断延迟小、网络流量大的特点,与其它嵌入式实时操作系统相比有一定的优势。

首先, VxWorks 的任务调度策略为以可抢占式调度 (Pre-emptive Scheduling) 为基础,辅以时间片轮转调度算法 (Round-Robin Scheduling)。这一调度算法使得 VxWorks 能够及时地响应高优先级的任务。而同级任务间则可选择时间片轮转法使多个同优先级的任务并发执行。

其次, VxWorks 采用中断处理与普通任务分别在不同的栈中处理的中断处理机制。这使得中断的产生只会引发一些关键寄存器的存储而不会导致任务的上下文切换,从而减小了中断延迟。同时, VxWorks 的中断处理程序只完成在最小时间内通告中断的发生,而将其他的处理尽量放入被引发的中断服务程序中来完成,进一步缩小了中断延迟。VxWorks 内核中普遍采用互斥信号量 (Sempahores) 而不是关中断来实现互斥访问的方法也对缩小中断延迟有一定的贡献。

VxWorks 还具有比较优秀的网络处理能力,与同类产品相比,其网络处理速度及流量 (Throughout) 目前均处于领先水平。

1.2 VxWorks 具友好的开发调试环境,便于操作、配置和应用程序的开发调试

嵌入系统的本质特点使其开发和调试过程较一般系统更为复杂。一个友好的开发环境对嵌入系统显得尤为重要。

VxWorks 的开发环境 Tornado 就是一个友好的开发环境,它可运行在多种主机上,包括 Sun, HP, IBM- rs6000, Dec, Mips 等,主机操作系统则支持 UNIX, WindowsNT, Windows2000/98/95 等。系统的各项配置由于使用较为流行的图形界面 (如对话框、列表、选项、按钮等) 而使使用者能较快地理解和掌握。

VxWorks 支持应用程序的动态链接和动态下载,使开发者省去了每次调试都将应用程序与操作系统核进行链接和下载的步骤,缩短了编辑-调试的周期。

VxWorks 提供功能强大的目标机仿真器 VxSim, 使开发者可独立于硬件环境而先行开发应用程序,从而节省了新产品的

研发时间和硬件方面的开销。

VxWorks 具可视化图形界面的调试工具很大程度地方便了开发者的调试工作。

1.3 VxWorks 具有较好的兼容性

VxWorks 是最早兼容 POSIX1003.1b 标准的嵌入式实时操作系统之一,同时是 POSIX 组织的主要会员。

VxWorks 的 TCP/IP 协议栈部分与 BSD4.4 版本的 TCP/IP 除了在实时性方面有较大差别外,其他方面基本兼容。这使得基于 BSD4.4 Unix Socket 的应用程序可以很方便的移植到 VxWorks 中。移植的主要工作也在于保证系统的实时性方面。

VxWorks 还是第一个通过 WindowsNT 测试的可在 WindowNT 平台进行开发和仿真的嵌入式实时操作系统。

VxWorks 同时支持 ANSI C 标准,并通过 ISO9001 的认证。

VxWorks 良好的兼容性,使其在不同运行环境间可以方便的移植,从而使用户在开发和培训方面所做的工作得到保护,减少了开发周期和经费。

1.4 VxWorks 支持多种开发和运行环境

VxWorks 的应用范围和领域比较广泛。VxWorks 开发环境支持的主机包括 Sun, HP, IBM-rs6000, Dec, Mips 等。系统运行环境支持 PowerPC, 68K, CPU32, SPARC, i960, x86, MIPS 等众多 CPU 及支持 RISC, DSP 等技术。支持多种硬件环境也是 VxWorks 得以流行的一个重要原因,同时,从中可以看出, VxWorks 同世界诸多硬件厂商有着紧密的关系。

1.5 在紧跟新技术持续发展方面有一定的能力和经验

在不断的发展过程中, VxWorks 完成了不少改进和革新。它是最早实现捆绑集成交叉开发环境的嵌入式实时操作系统之一,也是最早在其内核中加入 TCP/IP 网络协议的嵌入式实时操作系统,此后,又率先宣布支持网络文件系统 NFS。在系统集成最新网络协议方面一直保持良好的势头。VxWorks 还是最先支持 RISC 处理器的嵌入式实时操作系统。从这可以看出 VxWorks 具有较强的开发革新能力及较快的更新周期。

2 VxWorks 的基本结构及各组成部件情况

VxWorks 的开发环境包括两部分,运行在主机 (HOST) 上的集成开发环境 Tornado 和运行在目标机上的操作系统 VxWorks 本身。Tornado 完成应用程序的编辑、编译、调试、系统配置等, VxWorks 则是应用程序的最终执行环境,同时 VxWorks 支持动态下载、调试。Tornado 和 Target 通过以太网或串行接口连接。上述就构成了所谓的交叉开发环境 (CrossDevelopment), 如图 1 所示。

VxWorks 的基本构成组件包括 BSP (Board Support Package)、微内核 Wind、网络系统、文件系统及 I/O 系统。

VxWorks 系统具有较好的可裁剪的能力,可裁剪的组件超过 80 个。用户可根据自己系统的功能目标通过交叉开发环境方便地进行配置。

VxWorks 采用模块化设计方法,把依赖于硬件环境的函数和信息分离出来,放入称为 BSP (Board Support Package) 的组件中。而 BSP libraries 向上层软件提供一致的接口。VxWorks 在 target 上运行需要相应的 BSP 的支持。

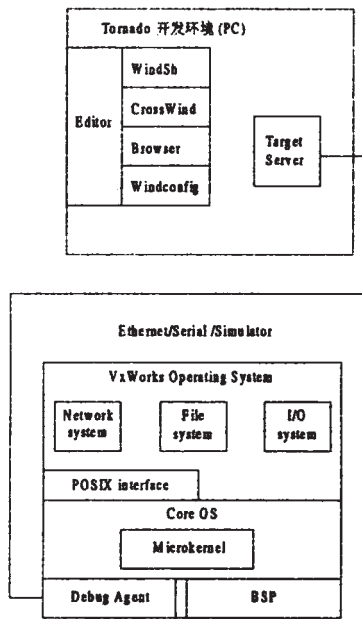


图 1 VxWorks 及其开发环境的构成

微内核 Wind 是 WindRiver 自行开发的一种嵌入式操作系统内核,经过多次的改进和修补,现趋于稳定,最近几个版本的 VxWorks 的 Wind 内核均没有变化。

VxWorks 的网络系统基本上是移植了 BSD4.4 UNIX 的 TCP/IP 协议栈,仅在实时性上进行了较大的修改,如用 Semaphores 代替原 TCP/IP 实现中的 interrupt lock 和修改原 Socket 代码中无 Timeout 的部分。这使得网络应用软件的开发和移植都较为方便。由于在实际的通信中,要经常用到网络系统,下面对此将作些简述。至于文件系统和 I/O 系统,限于篇幅,这里就不再讲述了。

3 VxWorks 的网络系统构成

VxWorks 的网络系统由标准的网络协议部分组成,即由物理层、数据链路层、IP 层、TCP 层、应用层。VxWorks 为了增强系统的应用性和可移植性,在新版的 VxWorks 中增加了一个称为 MUX 层的接口层。各部分关系如图 2 所示。

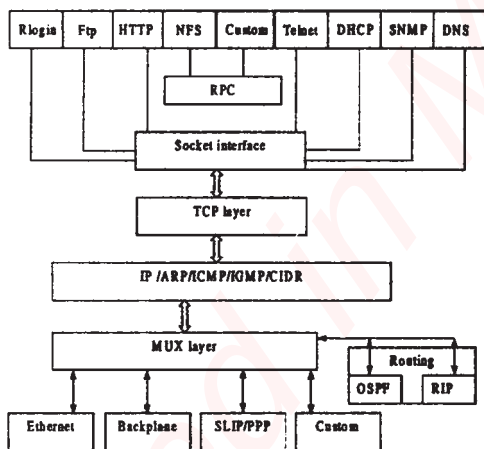


图 2 VxWorks 网络系统的结构

VxWorks 的网络系统从 MUX 层以上到 Sockets 层是一个紧耦合系统,其中涉及的协议如 TCP/IP,ICMP,IGMP,CIDR 及 Socket 之间的结合较为紧密,各协议间的接口关系复杂且 WindRiver 公司没有提供描述这些接口关系的文档和使用说明。同时,这部分协议虽然原理及大部分实现上与 BSD4.4 TCP/IP 一致,但其中涉及实时性的、与 VxWorks 结合的部分如何实现不可知,故修改或替换其中的协议将难以实现。目前,VxWorks 只提供功能有限的且仅供应用程序使用的各协议层间接口,其功能更强的网络协议开发工具包 (Network Protocol Toolkit) 试用版已经开发出来,有待进一步的测试。

修改或替换 VxWorks 部分网络协议虽然十分困难,但将整

个 TCP/IP 协议栈替换掉反而是一件相对容易的工作。因为其 Socket 层和 MUX 层的接口关系明确。

MUX 层是存在于网络协议层与数据链路层之间的接口层。MUX 的存在使网络协议层与数据链路层独立开来。网络协议层通过 MUX 层访问数据链路层,反之亦然。在新版的 VxWorks 中,无论是网络协议层还是数据链路层都不能直接访问对方。这样,一方的实现完全可以不考虑对方的实现细节而只需遵从标准的接口。这样,就使得用户完全可以在已有的链路层上添加自己开发的协议软件。

由于 socket 层遵从 BSD4.4 标准,故我们自己的 TCP/IP 协议层及 socket 层只需向上提供标准的 Socket 接口,遵循标准 Socket 接口的上层应用程序仍可正常工作在该网络协议之上。

目前,VxWorks 提供的协议层间接口主要有以下几个:

3.1 EtherLib

EtherLib 为应用程序提供了直接访问 Ethernet 层上传数据包的接口。该函数库中包含以下 7 个函数调用,分别为: EtherOutput(), etherInputHookAdd(), etherInputHookDelete(), etherOutputHookAdd(), etherOutputHookDelete(), etherAddrResolve(), etherTypeGet()。

应用程序通过调用函数 etherOutput() 就可以直接通过 Ethernet 层向目的地发送数据包。而对 Ethernet 层输入输出数据包的操作可以通过在 etherInputHookAdd() 和 etherOutputHookAdd() 中加入自己的包处理程序来实现。各函数的细节描述及具体调用形式请参见 VxWorks 的联机手册。

尽管目前的 VxWorks 版本中仍旧可以使用 EtherLib 来完成一些功能,但 WindRiver 公司并不推荐使用该函数库中的函数调用,并宣称在将来的 VxWorks 版本中将不再继续支持该函数库。

3.2 MUX 接口

MUX 应用接口包含 muxBind(),muxUnbind(),muxReceive(), muxSend(), muxIoctl() 等函数调用。需要从 Ethernet 层接收数据包的协议软件必须首先通过调用 muxBind() 同 MUX 建立连接。一旦某个协议软件与 MUX 建立起连接,MUX 就将接收到的数据包传给该协议处理。当协议软件需要发送数据包时,它调用 muxSend(),muxSend() 再调用 Ethernet 层的发包函数将包送出。如果 Ethernet 层的发包函数出错,将通过 muxSend() 向上层协议程序返回一个错误信息来暂停或取消上层协议的发包动作。当协议软件想退出使用该网络接口时,通过调用 muxUnbind() 来断开与 MUX 的连接。具体实现请参考 VxWorks 联机手册。

3.3 ipFilterLib

ipFilterLib 中包含三个函数: ipFilterLibInit(), ipFilterHookAdd(), ipFilterHookDelete()。该函数库为应用程序提供了直接访问 IP 包的接口。应用程序通过 ipFilterHookAdd() 实现对原始 IP 包的检查和处理。尽管目前的 VxWorks 版本中支持该函数库,VxWorks 建议不使用其中的函数来开发应用程序。所有该库的功能可通过 MUX 编程来实现。

3.4 SocketLib

该函数库中包含与 UNIX BSD 4.4 兼容的 Socket 函数调用,包括 socket(), bind(), select(), accept(), send() 等。该函数库为运行在不同或同一地点的应用程序提供连接通信的接口,其中包括网络连接及其他连接。库中各函数接口的编程方法同 UNIX BSD 4.4 完全一致。具体实现请参阅 VxWorks 联机手册和 UNIX 编程手册。

3.5 PppHookLib

该函数库提供了在建立和断开 PPP 连接时转入用户应用程序的手段。包括 pppHookAdd(), pppHookDelete()。应用程序可通过该函数库在 PPP 链路建立前和断开后的瞬间完成必要的功能。

上述几个接口就是目前 VxWorks 能够提供给我们进行网络软件开发的接口。WindRiver 公司计划在近期内推出其更具更强功能的网络协议开发工具包 (Network Protocol Toolkit)。该工具包将提供更完善的网络协议开发接口。

参考文献:

[1] VxWorks Programmer's Guide, WindRiver, Inc,2004

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](#)
49. [串口和以太网通信技术在油液在线监测系统中的应用](#)
50. [USB30 数据传输协议分析及实现](#)
51. [IEEE 1588 协议在工业以太网中的实现](#)
52. [基于 USB30 的设备自定义请求实现方法](#)
53. [IEEE1588 协议在网络测控系统中的应用](#)
54. [USB30 物理层中弹性缓冲的设计与实现](#)
55. [USB30 的高速信息传输瓶颈研究](#)
56. [基于 IPv6 的 UDP 通信的实现](#)
57. [一种基于 IPv6 的流媒体传送方案研究与实现](#)
58. [基于 IPv4-IPv6 双栈的 MODBUS-TCP 协议实现](#)
59. [RS485CAN 网关设计与实现](#)
60. [MVB 周期信息的实时调度](#)
61. [RS485 和 PROFINET 网关设计](#)
62. [基于 IPv6 的 Socket 通信的实现](#)
63. [MVB 网络重复器的设计](#)
64. [一种新型 MVB 通信板的探究](#)
65. [具有 MVB 接口的输入输出设备的分析](#)
66. [基于 STM32 的 GSM 模块综合应用](#)
67. [基于 ARM7 的 MVB CAN 网关设计](#)
68. [机车车辆的 MVB CAN 总线网关设计](#)
69. [智能变电站冗余网络中 IEEE1588 协议的应用](#)
70. [CAN 总线的浅析 CANopen 协议](#)
71. [基于 CANopen 协议实现多电机系统实时控制](#)
72. [以太网时钟同步协议的研究](#)
73. [基于 CANopen 的列车通信网络实现研究](#)
74. [基于 SJA1000 的 CAN 总线智能控制系统设计](#)
75. [基于 CANopen 的运动控制单元的设计](#)
76. [基于 STM32F107VC 的 IEEE 1588 精密时钟同步分析与实现](#)

77. [分布式控制系统精确时钟同步技术](#)
78. [基于 IEEE 1588 的时钟同步技术在分布式系统中应用](#)
79. [基于 SJA1000 的 CAN 总线通讯模块的实现](#)
80. [嵌入式设备的精确时钟同步技术的研究与实现](#)
81. [基于 SJA1000 的 CAN 网桥设计](#)
82. [基于 CAN 总线分布式温室监控系统的设计与实现](#)
83. [基于 DSP 的 CANopen 通讯协议的实现](#)
84. [基于 PCI9656 控制芯片的高速网卡 DMA 设计](#)
85. [基于以太网及串口的数据采集模块设计](#)

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](#)
30. [VxWorks 实时操作系统及其在 PC104 下以太网编程的应用](#)
31. [实时操作系统任务调度策略的研究与设计](#)
32. [军事指挥系统中 VxWorks 下汉字显示技术](#)
33. [基于 VxWorks 实时控制系统中文交互界面开发平台](#)
34. [基于 VxWorks 操作系统的 WindML 图形操控界面实现方法](#)
35. [基于 GPU FPGA 芯片原型的 VxWorks 下驱动软件开发](#)
36. [VxWorks 下的多串口卡设计](#)
37. [VxWorks 内存管理机制的研究](#)
38. [T9 输入法在 Tilcon 下的实现](#)
39. [基于 VxWorks 的 WindML 图形界面开发方法](#)
40. [基于 Tilcon 的 IO 控制板可视化测试软件的设计和实现](#)
41. [基于 VxWorks 的通信服务器实时多任务软件设计](#)
42. [基于 VXWORKS 的 RS485MVB 网关的设计与实现](#)
43. [实时操作系统 VxWorks 在微机保护中的应用](#)
44. [基于 VxWorks 的多任务程序设计及通信管理](#)
45. [基于 Tilcon 的 VxWorks 图形界面开发技术](#)
46. [嵌入式图形系统 Tilcon 及应用研究](#)
47. [基于 VxWorks 的数据采集与重演软件的图形界面的设计与实现](#)
48. [基于嵌入式的 Tilcon 用户图形界面设计与开发](#)
49. [基于 Tilcon 的交互式多页面的设计](#)
50. [基于 Tilcon 的嵌入式系统人机界面开发技术](#)
51. [基于 Tilcon 的指控系统多任务人机交互软件设计](#)
52. [基于 Tilcon 航海标绘台界面设计](#)
53. [基于 Tornado 和 Tilcon 的嵌入式 GIS 图形编辑软件的开发](#)
54. [VxWorks 环境下内存文件系统的应用](#)
55. [VxWorks 下的多重定时器设计](#)
56. [Freescale 的 MPC8641D 的 VxWorks BSP](#)
57. [VxWorks 实验五\[时间片轮转调度\]](#)
58. [解决 VmWare 下下载大型工程.out 出现 WTX Error 0x100de 的问题](#)
- 59.

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](#)
37. [Linux 系统中进程调度策略](#)
38. [嵌入式 Linux 实时性方法](#)
39. [基于实时 Linux 计算机联锁系统实时性分析与改进](#)
40. [基于嵌入式 Linux 下的 USB30 驱动程序开发方法研究](#)
41. [Android 手机应用开发之音乐资源播放器](#)
42. [Linux 下以太网的 IPv6 隧道技术的实现](#)
43. [Research and design of mobile learning platform based on Android](#)
44. [基于 linux 和 Qt 的串口通信调试器调的设计及应用](#)
45. [在 Linux 平台上基于 QT 的动态图像采集系统的设计](#)

46. [基于 Android 平台的医护查房系统的研究与设计](#)
47. [基于 Android 平台的软件自动化监控工具的设计开发](#)
48. [基于 Android 的视频软硬解码及渲染的对比研究与实现](#)
49. [基于 Android 移动设备的加速度传感器技术研究](#)
50. [基于 Android 系统振动测试仪研究](#)
51. [基于缓存竞争优化的 Linux 进程调度策略](#)
52. [Linux 基于 W83697 和 W83977 的 UART 串口驱动开发文档](#)

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](#)
21. [DCOM 协议在网络冗余环境下的应用](#)
22. [Windows XP Embedded 在变电站通信管理机中的应用](#)
23. [XPE 在多功能显控台上的开发与应用](#)
24. [基于 Windows XP Embedded 的 LKJ2000 仿真系统设计与实现](#)
25. [虚拟仪器的 Windows XP Embedded 操作系统开发](#)
26. [基于 EVC 的嵌入式导航电子地图设计](#)
27. [基于 XPEmbedded 的警务区 SMS 指挥平台的设计与实现](#)
28. [基于 XPE 的数字残币兑换工具开发](#)
29. [Windows CENET 下 ADC 驱动开发设计](#)

30. [Windows CE 下 USB 设备流驱动开发与设计](#)
31. [Windows 驱动程序设计](#)
32. [基于 Windows CE 的 GPS 应用](#)
33. [基于 Windows CE 下大像素图像分块显示算法的研究](#)
34. [基于 Windows CE 的数控软件开发与实现](#)

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](#)
18. [基于 MPC8260 处理器的 PPMC 系统](#)
19. [基于 PowerPC 的控制器研究与设计](#)
20. [基于 PowerPC 的模拟量输入接口扩展](#)
21. [基于 PowerPC 的车载通信系统设计](#)
22. [基于 PowerPC 的嵌入式系统中通用 IO 口的扩展方法](#)
23. [基于 PowerPC440GP 型微控制器的嵌入式系统设计与研究](#)
24. [基于双 PowerPC 7447A 处理器的嵌入式系统硬件设计](#)
25. [基于 PowerPC603e 通用处理模块的设计与实现](#)
26. [嵌入式微机 MPC555 驻留片内监控器的开发与实现](#)
27. [基于 PowerPC 和 DSP 的电能质量在线监测装置的研制](#)
28. [基于 PowerPC 架构多核处理器嵌入式系统硬件设计](#)
29. [基于 PowerPC 的多屏系统设计](#)
30. [基于 PowerPC 的嵌入式 SMP 系统设计](#)

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](#)
26. [ARM11 嵌入式系统 Linux 下 LCD 的驱动设计](#)
27. [Uboot 在 S3C2440 上的移植](#)
28. [基于 ARM11 的嵌入式无线视频终端的设计](#)
29. [基于 S3C6410 的 Uboot 分析与移植](#)
30. [基于 ARM 嵌入式系统的高保真无损音乐播放器设计](#)
31. [UBoot 在 Mini6410 上的移植](#)
32. [基于 ARM11 的嵌入式 Linux NAND FLASH 模拟 U 盘挂载分析与实现](#)
33. [基于 ARM11 的电源完整性分析](#)
34. [基于 ARM S3C6410 的 uboot 分析与移植](#)
35. [基于 S5PC100 移动视频监控终端的设计与实现](#)

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](#)
23. [基于龙芯平台的 PMON 研究与开发](#)
24. [基于 X86 平台的嵌入式 BIOS 可配置设计](#)
25. [基于龙芯 2F 架构的 PMON 分析与优化](#)
26. [CPU 与 GPU 之间接口电路的设计与实现](#)
27. [基于龙芯 1A 平台的 PMON 源码编译和启动分析](#)
28. [基于 PC104 工控机的嵌入式直流监控装置的设计](#)
29. [GPGPU 技术研究与发展](#)
30. [GPU 实现的高速 FIR 数字滤波算法](#)
31. [一种基于 CPUGPU 异构计算的混合编程模型](#)
32. [面向 OpenCL 模型的 GPU 性能优化](#)
33. [基于 GPU 的 FDTD 算法](#)
34. [基于 GPU 的瑕疵检测](#)
35. [基于 GPU 通用计算的分析与研究](#)
36. [面向 OpenCL 架构的 GPGPU 量化性能模型](#)
37. [基于 OpenCL 的图像积分图算法优化研究](#)

38. [基于 OpenCL 的均值平移算法在多个众核平台的性能优化研究](#)
39. [基于 OpenCL 的异构系统并行编程](#)
40. [嵌入式系统中热备份双机切换技术研究](#)

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](#)
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](#)
7. [数据结构考题 - 第 1 章 绪论](#)
8. [数据结构考题 - 第 2 章 线性表](#)
9. [数据结构考题 - 第 2 章 线性表 - 答案](#)
10. [基于小波变换与偏微分方程的图像分解及边缘检测](#)
11. [基于图像能量的布匹瑕疵检测方法](#)
12. [基于 OpenCL 的拉普拉斯图像增强算法优化研究](#)
13. [异构平台上基于 OpenCL 的 FFT 实现与优化](#)
14. [数据结构考题 - 第 4 章 串](#)
15. [数据结构考题 - 第 4 章 串答案](#)
- 16.

FPGA / CPLD:

1. [一种基于并行处理器的快速车道线检测系统及 FPGA 实现](#)
2. [基于 FPGA 和 DSP 的 DBF 实现](#)
3. [高速浮点运算单元的 FPGA 实现](#)
4. [DLMS 算法的脉动阵结构设计及 FPGA 实现](#)
5. [一种基于 FPGA 的 3DES 加密算法实现](#)
6. [可编程 FIR 滤波器的 FPGA 实现](#)
7. [基于 FPGA 的 AES 加密算法的高速实现](#)
8. [基于 FPGA 的精确时钟同步方法](#)
9. [应用分布式算法在 FPGA 平台实现 FIR 低通滤波器](#)
10. [流水线技术在用 FPGA 实现高速 DSP 运算中的应用](#)
11. [基于 FPGA 的 CAN 总线通信节点设计](#)

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

12. [基于 FPGA 的高速时钟数据恢复电路的实现](#)
13. [基于 FPGA 的高阶高速 FIR 滤波器设计与实现](#)
14. [基于 FPGA 高效实现 FIR 滤波器的研究](#)
- 15.