VxWorks 下的多串口卡设计

宋桂景，崔美英，班亚明
中国电子科技集团公司第五十四研究所，河北 石家庄 050081）

摘 要 针对目前通信系统中串口通信，论述了基于嵌入式操作系统 Vxworks 扩展多个串口以进行实时通信的设计方案，给出了硬件设计核心单元以及串口扩展单元实现原理、软件多任务交互以及通信设计流程，并对多串口卡的实时性处理、程序的模块化设计进行了详细分析。通信测试结果表明该设计提高了 CPU 利用率和软件效率，扩展了通信范围。

关键词 VxWorks 多任务 实时 串口
中图分类号 TP391 文献标识码 A 文章编号 1003－3114（2012）02－59－3

Design of Multi-serial Port Card Based on VxWorks

SONG Gui-jing, CUI Mei-ying, BAN Ya-ming
(The 54th Research Institute of CETC, Shijiazhuang Hebei 050081, China)

Abstract Aiming at multi-serial ports communication in communication system at present, the scheme of extending multi-serial ports for real-time communication based on embedded VxWorks system is dissertated. The core unit in hardware design and the principle of serial port extended unit are presented. The exchange among multi tasks and communication flow of software design are also introduced. The real-time processing of multi-serial ports communication and the modularization of software design are analyzed in detail. The test results show that this design scheme can improve CPU utilization rate and software efficiency, and extend communication range.

Key words VxWorks; multi-task; realtime; serial port

0 引言

近年来，随着嵌入式计算技术的不断发展，VxWorks 嵌入式实时操作系统以高度可裁减的微内核、高效的多任务调度和灵活的任务间通信等优点在目前的卫星通信领域中得到了广泛应用。VxWorks 作为一种嵌入式实时多任务操作系统，是指能在确定的时间内执行其功能，并对外部的异步事件作出实时响应的计算机系统。在通信系统内部，管理代理终端设备以及信道设备之间大多是以串口连接，通过串口进行一些控制指令的交互和数据的传输。串口在嵌入式系统中是一类重要的数据通信接口，其本质功能是作为 CPU 和串行设备间的编码转换器。应用程序在多串口间的通信方式，就是利用实时的多任务灵活调度机制来实现的数据通信的方式。

1 方案设计

在 VxWorks 的多串口卡设计中，应遵循以下 3 个原则：

① 尽量选用目前通用的嵌入式核心模块；

② 选用方便可靠的多串口扩展模块；

③ 程序设计框架要实用，方便串口信息的增加或删减。

1.1 硬件设计

在 VxWorks 的多串口卡硬件设计中，采用高集成度小型化的设计思想，选用具有独立功能的嵌入式 CPU 核心模块——PowerPC 8270 处理器模块作为核心单元，外接多串口扩展模块。按照功能类型划分，该种设计方式主要由 3 部分组成 核心单元、串口扩展单元以及串口转换单元，硬件原理框图如图 1 所示。

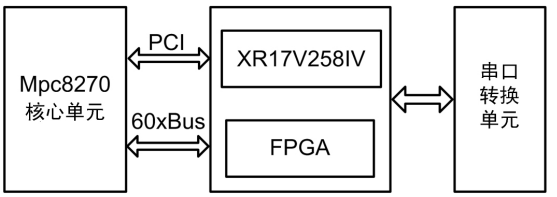

图 1 原理框图

收稿日期 2011－12－27
作者简介 宋桂景（1978— ），女，工程师。主要研究方向 地面接入通信。

1.1.1 核心单元

该 mpc8270 核心单元具有丰富的通信接口、大量数据传输能力、实时/多任务复杂逻辑和过程处理能力。在 8270 核心模块上运行嵌入式操作系统和应用软件，实现业务接入、监视控制等功能。该核心模块自带 2 个低速串口，可直接连接串口转换芯片，一般在系统应用中采用其中的 1 个串口作为打印串口以监测通信状态。其硬件框图如图 2 所示。

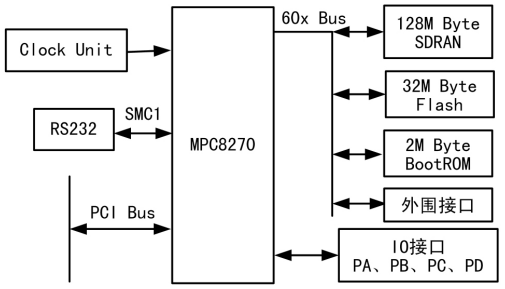

图 2 Mpc8270 核心模块原理框图

1.1.2 串口扩展单元

该设计实现串口扩展采用 2 种方式：采用基于 PCI 总线接口芯片 XR17V258IV 可扩展出 8 个串口，通过可编程逻辑芯片 FPGA 实现串口的进一步扩展。XR17V258IV 是基于 PCI 总线串口的单芯片解决方案，该串口扩展卡可以很容易地扩展为 8 个串口，而不需要进行硬件的更改和时钟时序调整；该设计还可以利用总线，通过 FPGA 进一步实现串口的扩展，该方法在可编程逻辑芯片内实现，对于串口的扩展以及管脚的分配更为方便，所以应用更为灵活。

1.1.3 串口转换单元

串口转换单元可以根据实际需要进行电平转换：TTL/RS232、TTL/RS485、TTL/RS422，以连接外部数据通信接口。

1.2 软件设计

VxWorks 实时内核 Wind 提供了基本的多任务环境。VxWorks 操作系统中任务调度算法有时间片轮转、优先级抢占和独占资源。时间片轮转算法依据时间片的分配来为任务分配 CPU 资源，优先级抢占算法指在任务生成之时都分配有不同或相同的优先级，独占资源式算法指任务独占 CPU 资源。采用优先级抢占式资源调度算法更贴近实际，在 VxWorks 中，需要尽快处理的设置为高优先级。而独占资源或时间片轮转的调度方式都是面向同优先级任务的。在该设计中采用优先级抢占调度和时间片轮转调度相结合的方式。

1.2.1 多任务交互

在 8270 核心模块上运行 VxWorks 嵌入式操作系统和应用软件，实现实时多串口通信的信息交互。多串口通信中多任务间的交互方式用到主要以下 3 种：

① 共享内存：将需要共享的数据声明为全局变量；

② 信号量：提供简单的同步和互斥机制；

③ 消息队列或管道：任务间传输带有数据的消息，为长度、数目可变的一组数据进行排队。

使用共享内存可以传递大量的数据，实现快速的数据交换，在串口数据状态的通信中使用 VxWorks 内核提供的共享内存工具环形缓冲：共享内存没有任务阻塞或互斥机制的保护，可能引起数据访问冲突，该设计中可以使用信号量将共享内存保护起来：任务间的数据通信采用消息队列机制，可以通过消息队列机制定制获取数据的时机、方式。

1.2.2 通信流程

在利用 VxWorks 操作系统进行多串口通信，必须在串口通信使用之前向操作系统提出资源申请要求（打开串口）并进行串口配置，通信完成后必须释放资源（关闭串口）。

针对 VxWorks 的上述通信机制，为每个串口创建 1 个串口守候接收任务，1 个发送任务，同时为每个串口建立 1 个二进制信号量。建立 1 个主处理任务的消息队列，同时还需启动看门狗的定时功能。通信数据流程如图 3 所示。

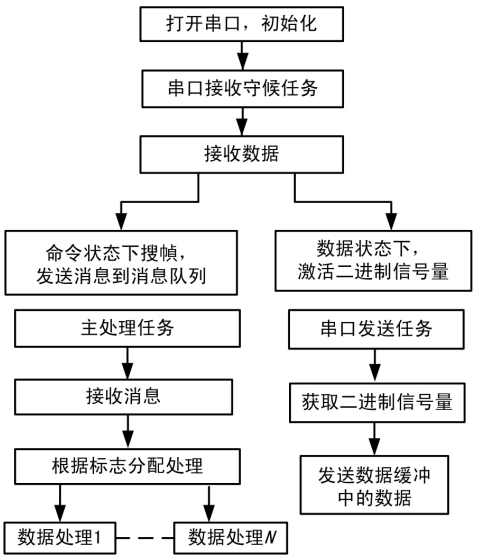

图 3 串口任务数据流程

串口通信在命令状态下对数据的实时性要求要

稍低一些，所以串口接收守候任务在命令状态下，接收到数据、搜帧，而后加上标志发送到主处理任务的消息队列，由主处理任务根据标志在分发到相应的处理函数进行处理。而在数据状态下，对数据的实时性要求要高，因此可以通过激活二进制信号量，启动数据状态下的数据接收，此时将数据内容放入环形缓存中，对应接收端有一个数据收指针。

串口数据发送任务获取二进制信号量，对应发送端有发送指针，通过发送指针从对应环形缓存中取得数据发送。

主处理任务可以接收多个串口接收任务和看门狗等其他任务发送过来的消息，根据消息的标志分配给不同的数据处理函数进行处理。

各数据处理函数采用状态机实现业务的处理，就是将一项工作分成若干阶段，明确指定在每阶段应该做什么，以及下一步可以做什么的转移，状态机是一个有向图，由一组节点和一组相应的转移函数组成，节点即工作流中的某个环节，而转移函数对应的是业务逻辑层的某一个对应类中的方法集合，转移函数返回“下一个”节点。此种处理方式可以方便地增加状态信息的处理。

2 关键技术

2.1 实时处理

在实时通信应用中，对时间的处理是至关重要的。需要将操作系统中的时间片与现实时间联系在一起，函数接口 sysClkRateGet 用于获取每秒产生的时间片数量，时间片函数 taskDelay 具有延时和任务调度的功能，任务在调用了 taskDelay 后，会被系统放置在等待队列中，此时 CPU 资源被让出，开始执行就绪队列中的下一个任务，当系统核心在调度时发现延时完成后，该任务会被放置在就绪队列末尾等待 CPU 资源，此种调度方式会导致定时不准确。看门狗计时可以提供比较精确的计时，定时时间一到，所安排的任务将以中断级别执行，会打断大部分的任务执行。看门狗计时方式比 taskDelay 消耗了更多的系统资源，但更精确，可以用于延时、计时和定时。

2.2 模块化设计

该设计中，采用模块化的思想，主要由 2 个方面来体现。

① 数据处理函数模块化 因为同一个物理串口可以根据实际需要连接不同的串口设备，所以在实际设计中可以按照实际应用分为不同的数据处理函数来处理 ；

② 函数状态模块化 采用状态转移来实现不同状态的数据通信。在不同的状态可以通过消息队列接收不同的消息来处理。此种数据处理方式扩展和分析问题都很方便，可以根据需要增加某个函数的处理，同时在数据状态处理中，既可以根据需要增加 1 条消息的处理，又可以根据需要增加 1 个状态的处理。

3 测试结果分析

在该电路设计中，核心模块通过 PCI 总线接口芯片 XR17V258IV 扩展出 8 个串口，通过可编程逻辑芯片 FPGA 扩展出 4 个串口，共 12 个串口，其中 6 个连接数据终端设备 可以连接计算机终端 ，6 个连接数据通信设备 可连接调制解调器 Modem 。

利用计算机自带的通讯工具—超级终端进行通信测试，打开超级终端，设置通信速率，通过信道 Modem 向对端发起呼叫，可测试呼叫成功率 测试命令状态下的通信 以及数据通信中报文发送成功率 测试数据状态下的通行，其中涉及到数据流控的处理 。测试结果如表 1 所示。经过多次呼叫和数据通信，测试结果满足实际要求。实际数据通信中需要考虑信道状态是否良好。

表 1 通信测试结果

串口号	串口号	呼叫		数据通信	
		次数/次	成功率/%	数据量/kbit	成功率/%
1	7	10	100	510	100
2	8	10	100	484	100
3	9	10	100	380	100
4	10	10	100	733	100
5	11	10	100	832	100
6	12	10	100	953	100

4 结束语

该串口卡设计综合考虑了系统中多任务程序的设计及任务间的通信机制，该设备已在某工程中应用，满足系统对各个任务实时性的要求。合理的通信机制可以优化整个系统的性能，提高了 CPU 的利用率和软件效率。在嵌入式实时操作系统中使用串口通信，不仅可扩展嵌入式设备通信能力，而且可扩大其应用范围。

下转第 72 页)

UMAC 同时将尽可能多的数据装入 MAC 块，以便充分利用无线资源。

MAC 块的组装采取在每个时隙的时间内对每个载波的 4 个时隙各轮流组装 1 次的策略。以 1 个载波为例描述 MAC 块的创建组装流程。将 1 个载波的 4 个时隙标志为 A、B、C、D。在 UMAC 将 A 时隙的 MAC 块发送给 LMAC 之后，并没有马上开始组装下一帧的 A 时隙数据，而是采用如下流程：

① 检查有无数据需要装入 B 时隙的 MAC 块，如果有则装入，由于 B 时隙数据紧跟即将发送，所以如果 B 的 MAC 块尚未装满，需要将该 MAC 块用空 PDU 填充，之后生成该时隙的 AACH 信息，并装入 MAC 块 ② 检查有无数据需要装入 C 时隙的 MAC 块，如果有则装入 ③ 检查有无数据需要装入 D 时隙的 MAC 块，如果有则装入 ④ 由于 A 时隙的数据已经发送，所以需要重新初始化 A 时隙的 MAC 块，然后检查有无数据需要装入 A 时隙的 MAC 块，如果有则装入。

这一流程，每个时隙的 MAC 块在一帧时间 56.67 ms 之内具有 4 次组装机会，有利于提高 MAC 块的利用效率，更重要的是频繁的检查可以将下行的数据尽快装入 MAC 块，有效缩短了数据在 UMAC 缓存的时间，提高了 UMAC 层发送数据的效率。

5 软件测试结果分析

对 UMAC 层软件的测试，分为 3 个步骤完成，分别是的软件白盒测试、软件黑盒测试及 BS 系统集成测试。软件白盒测试，对每个函数编写测试用例，要求达到所有函数语句覆盖、核心处理函数判定/条件覆盖 软件黑盒测试，编写程序模拟上层的 LLC 协议软件以及下层的 LMAC 协议软件，通过 Socket 发送数据驱动 UMAC 软件的运行，要求为协议栈的每一个功能设计测试用例 集成测试是将 BS 作为一个整体进行测试。但是需要指出的是，以下几个测试用例证明了上述几个 UMAC 软件的关键技术是正确的：

① MS 开机，通过随机接入向 MS 发送注册请求，MS 注册成功 ② 某一 BS 下的 MS 发起个呼，呼叫建立、通话及释放未出现异常 ③ 某一 BS 下的 MS 发起组呼并讲话，其他 BS 下的 MS 可以听到清晰的话音 ④ MS 连续发起 100 次呼叫，呼叫接通率 100%，呼叫建立时间小于 500 ms。

6 结束语

按照该设计方法实现的 UMAC 软件具有传输延时低、资源利用率高的优点，可以满足协议要求的功能和在基站部署的需要。在国外 TETRA 设备完全占领中国市场的今天，为国内 TETRA 技术的发展，起到了极大的推动作用。随着时间的推进，该软件必将跟随基站设备，广泛应用在铁路、民航和公安等多个领域，具有较强的实用性和先进性。

参 考 文 献

[1] ETSI EN 300 392-2 Terrestrial Trunked Radio (TETRA) Voice plus Data (V + D) Part 2: Air Interface (AI) [S].

[2] 郑祖辉，陆锦华，丁锐，等. 数字集群移动通信系统 [M]. 北京：电子工业出版社，2008：372 - 342.

[3] 罗明. 基于 IP 的 TETRA 系统的研究 [D]. 武汉：武汉大学，2004.

[4] 李延波. TETRA 空中接口协议栈软件设计与实现 [D]. 天津：天津大学，2007.

[5] 仲达帆. TETRA 数字集群系统上 MAC 层上行信道的研究与开发 [D]. 北京：北京交通大学，2008.

[6] 宋政育，孙昕. TETRA 数字集群系统上 MAC 层协议栈的开发 [J]. 移动通信，2009，1(2)：35 - 39.

[7] 孙昕，李海. TETRA 数字集群空中接口协议栈体系结构分析 [J]. 移动通信，2008，3(2)：34 - 37.

（上接第 61 页）

参 考 文 献

[1] 张扬，于银涛. VxWorks 内核、设备驱动与 BSP 开发详解 [M]. 北京：人民邮电出版社，2009.

[2] 孔祥营，柏桂枝. 嵌入式实时操作系统 VxWorks 及其开发环境 Tornado [M]. 北京：中国电力出版社，2002.

[3] 李洪亮，侯朝桢. VxWorks 下实时多任务程序的实现 [J]. 单片机与嵌入式系统应用，2008，2(4)：7 - 2.

[4] 樊争奇. VxWorks 操作系统下基于缓冲队列网络通信应用 [J]. 电脑开发与应用，2009，22(3)：67 - 68.

[5] 郭平. VxWorks 串口通信程序设计与实现 [J]. 科技信息，2010，1(2)：69 - 70.

[6] 蒲元远，孙大维. VxWorks 下多串口通信设计 [J]. 光电技术应用，2009，4(2)：48 - 50.

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. 基于磁盘异或引擎的 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

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

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 CC++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

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

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

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μC-OSⅡ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

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 NiosⅡ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

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