基于 PowerPC 的车载通信系统设计

刘常清，黄文君，詹　源

(浙江大学智能系统与控制研究所工业控制技术国家重点实验室，杭州 310027)

摘　要：提出一种城市车辆间通信系统。系统由 PowerPC 硬件平台与嵌入式 Linux 软件平台构成，采用全球定位系统技术实时获取车辆定位信息，通过专用短距离通信技术使车辆间形成高速双向交互式通信，利用 ARM9 搭载 Linux 系统实现触屏显示。测试结果表明，该系统能实现车辆间高速、可靠、低延时的通信。

关键词：PowerPC 硬件平台；车载通信；全球定位系统；专用短距离通信

Design of Vehicle Communication System Based on PowerPC

LIU Chang-qing, HUANG Wen-jun, ZHAN Yuan

(State Key Lab of Industrial Control Technology, Institute of Cyber-Systems and Control, Zhejiang University, Hangzhou 310027, China)

【Abstract】This paper presents a design strategy for system among city vehicles in communication. The system is composed of PowerPC hardware platform and software platform based on embedded Linux operating system. Real-time vehicle location information is obtained by Global Positioning System(GPS) technology. Dedicated Short Range Communication(DSRC) technology is employed to achieve high-speed and two-way interactive communication between city vehicles. ARM9 loaded with embedded Linux operating system plays a role of touch screen display. Test result shows that the system can achieve high-speed, reliable, low-latency communication between vehicles.

【Key words】PowerPC hardware platform; vehicle communication; Global Positioning System(GPS); Dedicated Short Range Communication (DSRC)

DOI: 10.3969/j.issn.1000-3428.2012.07.069

1　概述

随着城市车流量的日益增大，交通拥堵的加剧，智能交通系统已经得到广泛的关注。智能交通系统主要由车辆控制中心、传输通道与车载通信系统构成，而如何提高车载通信的速率以及可靠性、实时性成为智能交通系统设计的核心。车载通信技术主要有专用短距离通信(Dedicated Short Range Communication, DSRC)、Wireless Fidelity(WiFi)、蜂窝网络(Cellular)技术。WiFi 技术数据传输率高但通信距离短小于 100 m，链路建立延时长为秒级。蜂窝网络技术通信距离远，但链路建立延时长为秒级，且提供的带宽非常有限，覆盖成本比较昂贵。因此，上述 2 种技术的应用具有很大的局限性。美国、日本、欧洲等国家各自开发了适用于智能交通领域的 DSRC 通信协议[1]，其中以美国的 WAVE(Wireless Access in Vehicular Environment)/DSRC(也通称 DSRC)通信协议最为完善，通信速率最高可达 27 Mb/s，距离 1 km，链路建立时间短于 50 ms，此外还具备数据包发送延时短、支持运行速度在 500 km/h 以内的车载通信等优点[2]。

本文利用 WAVE /DSRC 技术设计了高速、高可靠、低延时的车载通信系统。由于系统实现 DSRC 协议的数据量比较大，车载通信要求速率高、延时短，且对网络通信要求高，因此本文系统主控制器硬件平台选用基于 PowerPC 架构的具有优越处理性能与网络通信性能的 MPC8377 处理器。通过移植嵌入式 Linux 操作系统搭建软件平台，由于 MPC8377 内部未集成 LCD 控制器，系统选用 Cirrus Logic 公司 ARM9 系列的 EP9315 作为从控制器实现 LCD 触摸屏显示功能。

2　系统框架

车载通信系统框架如图 1 中虚线框所示，由路侧单元(Road Said Unit, RSU)与车载单元(On Board Unit, OBU)组成。系统通信包括车与路通信和车与车通信，为了方便起见，统称为车载通信。车载单元 OBU 采集 GPS 定位信息、油量车门检测等状态信息以及司机在遇到故障、危险道路时按键输入的状态信息，并将这些信息由 DSRC 模块实时地发送给路侧单元 RSU，由 RSU 与控制中心进行数据通信。

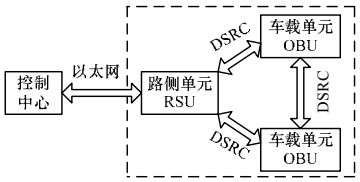

图 1　车载通信系统框架

此外，OBU 还将故障、危险道路等信息发送给同一 RSU 服务范围内的其他 OBU。RSU 将控制中心返回的电子地图、路况与调度等信息通过 DSRC 模块发送给各个 OBU，在 OBU 触摸屏上能实时地显示附近区域的电子地图、调度等信息。司机可以根据电子地图以及故障等路况、调度信息，选择最优路线，避开堵车或者危险区域，缓解交通拥堵，提高交通运输的安全性。

3　系统硬件设计

车载单元 OBU 硬件结构如图 2 所示。路侧单元 RSU 硬件结构只包含 DDR、NORFlash、DSRC 与以太网模块。

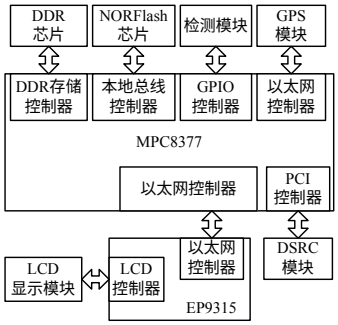

图 2　车载单元硬件结构

3.1　存储器模块

MPC8377 内部集成支持 64 位数据读写的 DDR SDRAM 控制器，外接 512 Mb/s DDR SDRAM 存储芯片，用于运行系统程序与用户程序，保证程序的高速运行与足够的内存空间。本地总线控制器控制 8 个存储块，每个存储块起始地址自由选择，在互不重叠情况下，各个存储块空间范围可在 32 KB~4 GB 间任意配置，且支持 NORFlash、SRAM、NandFlash 控制接口。第 1 个存储块配置成 NORFlash 接口，外接 8 Mb/s NORFlash 芯片，用于存放引导代码、操作系统与应用程序等数据，系统上电启动时从 NORFlash 中读取引导代码，完成基本的初始化工作后，把主程序拷贝至 DDR SDRAM 内存中运行。

3.2　LCD 显示模块

MPC8377 内部集成 2 个以太网控制器，均支持 10 Mb/s、100 Mb/s、1 000 Mb/s 3 种模式的通信速率，其中一个以太网接口负责与从控制器 EP9315 进行通信。主从控制器芯片内部仅实现了以太网的媒体介质访问控制(MAC)层协议，通过外接物理层 PHY 芯片 RTL8211 以 100 Mb/s 的速率进行通信。从控制器 EP9315 内部集成 LCD 控制器，外接 LCD 触摸屏显示模块用于显示接收到的数据信息，同时通过触摸屏按键产生中断的方式实时地监测司机按键事件的发生[3]。与主控制器类似，从控制器同样需扩展 8 Mb/s NOR Flash 用于存放引导代码、操作系统等固化程序。与 OBU 不同的是，RSU 并不包含从控制器显示模块，RSU 以太网接口用于与控制中心进行通信。

3.3　GPS 数据接收与处理模块

GPS 模块完成对 GPS 定位信号和时间同步数据的提取，对天线单元传来的 GPS 信号进行记录，并对信号进行解调和滤波处理，还原出 GPS 卫星发送的导航报文，最终获得导航定位的位置、方向、时间等数据。

GPS 模块选用目前市场上常用的 GP3SF2525W1 模块，该模块采用 RS232 通信协议接口，需要将 MPC8377 串口 UART 接口经物理层转换成 RS232 接口，系统通过 GP3SF 2525W1 模块实时获取 GPS 定位信息。

3.4　DSRC 数据收发模块

相对开放系统互连 OSI7 层模型来说，DSRC 协议具有如图 3 所示的 5 层结构：L1, L2, L3, L4 和 L7[4]。其中，数据链路层 L2 由逻辑链路控制子层(LLC)和 MAC 子层构成。物理层 L1 与 MAC 子层的下部分采用无线局域网协议标准中 IEEE802.11p 协议。该协议规定通信频率、速率、辐射功率、信号调制方式等。IEEE802.11p 以上的协议是由 IEEE1609 工作组针对车载环境下的无线接入(WAVE)所制订的。MAC 子层上部分为 IEEE1609.4 WAVE 多信道运行协议，它定义了一种 MAC 增强机制，提供多信道协作功能。LLC 子层及其以上的网络层 L3、传输层 L4 使用 IEEE1609.3 WAVE 网络服务协议，它定义了网络传输服务，利用了 IP 协议栈，以实现与现有网络的无缝融合，并引入了新的 WSMP(WAVE Short Message Protocol)协议以提高上层应用对底层设备的使用效率，支持车与车以及车与路通信的系列标准[4]。应用层 L7 由用户根据需要定制各种交通服务信息。

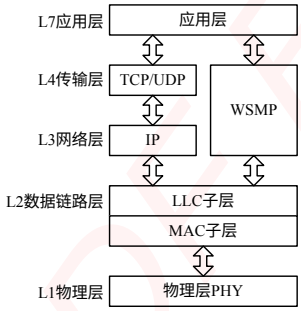

图 3　DSRC 协议层次

本文采用 Unex 公司的 DCMA-86P2 模块，该模块为 mini PCI 接口，需要将 MPC8377 PCI 接口转换成 miniPCI 接口与其进行通信。

此外，系统还设计了多路电源输出模块，满足硬件平台对不同电压与不同功率的电源需求；多路时钟输出模块，给各模块提供不同的精准时钟；GPIO 口检测模块，通过 GPIO 管脚连接各类传感器检测车内各种状态信息，如油量状态、车门状态等。

4　系统软件设计

4.1　Linux 系统移植

系统软件设计分为 MPC8377 主控制器软件设计与 EP9315 从控制器软件设计。由于主从控制器都需要同时完成多项并行的任务，为了解决多个任务运行带来的程序结构混乱以及减少从控制器软件开发的难度，均移植嵌入式 Linux 操作系统[5]。Linux 内核小、效率高，性能稳定，裁剪性好，支持图形界面，适应于多种 CPU 和多种硬件平台，内核结构在网络方面的支持非常完整。主控制移植基于 PowerPC 处理器的 Linux 2.6.32 内核，该内核集成了 WAVE 协议栈；从控制器移植针对 EP9315 的 Linux 2.6.32 内核以及图形界面。Linux 系统可以完成进程管理、内存管理、文件系统、设备控制、网络实现。

Linux 系统提供串口、PCI、GPIO、LCD 等设备驱动程序，降低了系统软件开发的难度，应用层开发人员通过调用标准接口对设备进行操作。由于串口、GPIO、LCD 驱动实现较为容易，本文不再介绍。对于 DSRC 协议而言，硬件 DSRC 模块仅实现了物理层与部分 MAC 层协议，驱动程序需要对 PCI 控制器进行设置，通过 PCI 接口配置 DSRC 模块，包括初始化、设置发送接收操作、中断处理等，上层协议由 Linux 内核实现，最终 Linux 给应用层用户提供 Socket API 标准接口，用户根据具体应用调用标准接口实现相应的操作。

4.2　控制器任务

按功能将主控制器系统软件分为 4 个主要的模块：读串口任务，DSRC 任务，GPIO 检测任务，以太网数据收发任务，如图 4 所示。

(1)读串口任务周期性地读取串口接收的 GPS 数据，并将数据放入固定的缓冲区以供 DSRC 任务读取。

(2)DSRC 任务一方面周期性地读取读串口任务获取的 GPS 定位数据以及由于 I/O 引发中断产生的异常数据，将其

打包成 DSRC 固定格式的数据包，即 WSMP 数据包，发送给 RSU；另一方面接收 RSU 发送的 WSMP 数据包，包括电子地图、路况、车况等数据。此外，当以太网接收到故障危险等数据包时，该任务还负责发送故障数据包给同一 RSU 范围内的其他 OBU 与 RSU。

(3)以太网数据收发任务将 DSRC 任务获取的电子地图、路况等数据用以太网数据包的形式，如 UDP，发送给从控制器，同时实时地接收从控制器在 LCD 触摸屏有按键时发送的故障危险等数据包。

(4)GPIO 检测任务周期性地检测 I/O 管脚电平状态，包括油量与车门检测。若发生异常，引发中断启动 DSRC 任务，发送相应的异常数据包通知 RSU，同时控制器也可由该任务改变 I/O 管脚的电平状态，起到控制的作用。

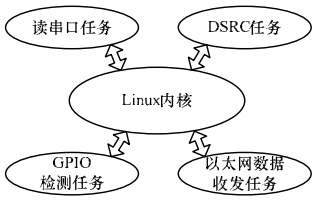

图 4　主控制器软件模块

从控制器功能较为简单，引入 Linux 操作系统主要执行 LCD 显示任务、按键处理任务与以太网数据收发任务。LCD 显示任务负责将以太网接收的数据在屏幕显示，当有按键发生时，引发中断发送以太网故障危险等数据包通知主控制器。路侧单元 RSU 仅实现 DSRC 任务的功能。

5　系统测试

系统原型机开发 5 套 OBU 与 1 套 RSU，将其安置于 1 km 范围内，OBU 的 2 个以太网接口分别用于连接从控制器与个人电脑，RSU 以太网接口连接电脑。系统启动后，6 台电脑以远程登录的方式分别进入 RSU 与 OBU 的 Linux 系统。在电脑端对测试所需的应用程序进行交叉编译，并将编译后的可执行文件下载至 RSU 与 OBU 端运行。设置 RSU 通过 DSRC 模块连续地发送 WSMP 数据包至 OBU，在不同的调制模式下测得通信速率如图 5 所示。从测试结果可知，系统的通信速率高达 8.8 Mb/s。

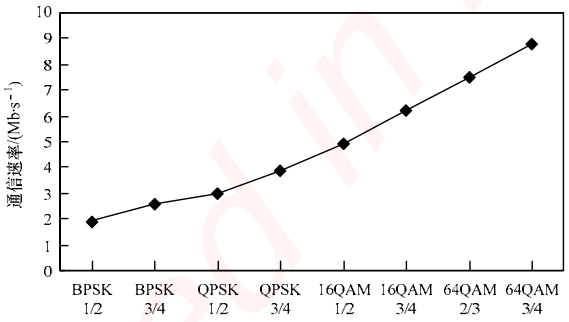

图 5　不同调制模式下 DSRC 通信速率

采用从 RSU 端发送 WSMP 数据包给各个 OBU，OBU 接收后立即回应相同大小的 WSMP 数据包的形式，进行数据包发送延时与丢包率测试。当数据包大小为 1 200 Byte 时，测得在不同的调制模式下发送延时时间如图 6 所示，可知选择合适的调制模式能使延时时间为 50 ms，系统具备低延时的特点。RSU 给每个 OBU 发送 1 000 个数据包，测得 RSU 端回应数据包个数为 5 000，未出现丢包。

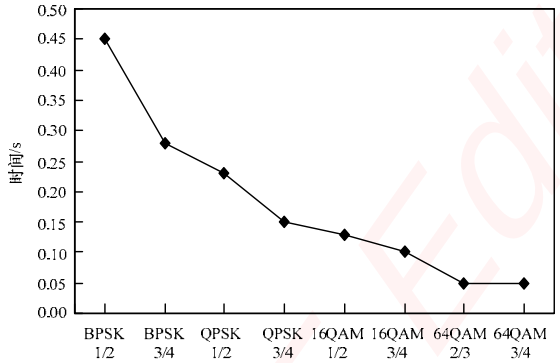

图 6　不同调制模式下数据包发送延时时间

由于原型机 OBU 过少，未出现丢包的现象，因此使用 NCTUns 6.0 网络仿真软件进行丢包率的测试，仿真结果如图 7 所示。可知在同一个 RSU 服务范围内车辆数达 30 辆时单辆车的丢包率低于百分之一，系统具有较高的可靠性。

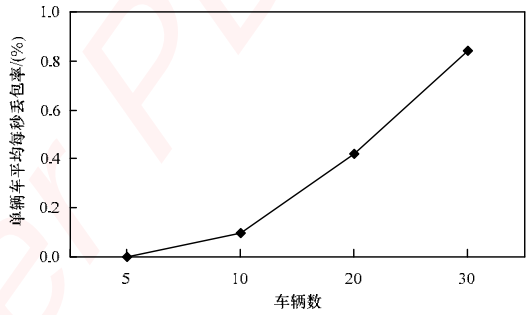

图 7　单辆车每秒平均丢包率

此外，还对 OBU 端串口接收 GPS 数据、触摸屏按键异常、I/O 管脚异常与 LCD 显示功能进行了测试，结果表明各模块均能实现相应的功能。

6　结束语

本文将 PowerPC 与嵌入式 Linux 操作系统结合，开发了车载通信系统主控模块，通过 ARM9 搭建 Linux 完成了显示从模块的设计。测试结果表明，系统不仅能够实现 GPS 数据采集、DSRC 数据收发等基本功能，而且具备高效、可靠、低延时的特点，具有较大的实用价值。下一步工作将会对多个 RSU 与多个 OBU 组网技术，以及 RSU 与控制中心的通信技术进行研究，以此构成一个完整的低成本易部署的智能交通系统。

参考文献

[1] 屠　宇, 徐建闽, 钟慧玲. DSRC 系统通信协议的开发[J]. 计算机工程, 2003, 29(21): 28-31.

[2] IEEE 1609.2-2006 IEEE Trial-Use Standard for Wireless Access in Vehicular Environments——Security Services for Applications and Management Messages[S]. 2006.

[3] 王文竹, 郭　华, 吴庆波. 基于 PowerPC 的高精度定时器设计与实现[J]. 计算机工程, 2010, 36(16): 267-269.

[4] 刘　学. 基于智能交通安全的车_车间无线通信系统传播模型研究[D]. 北京: 北京邮电大学, 2010.

[5] 崔再惠, 李铁男, 杨　峰. 基于嵌入式 Linux 的无线自动拍照终端[J]. 计算机工程, 2010, 36(13): 213-215.

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

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

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

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

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

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