

MVB 网络重复器的设计

曾争 蔡蕾

(同济大学 交通运输工程学院,上海 201804)

摘要: 随着我国交通运输的发展,TCN 通信网络得到了广泛的应用,特别是在城市轨道交通领域。MVB 网络重复器是 TCN 网络搭建中十分重要的设备,文章结合 MVB 的特点,提出了一种 MVB 网络重复器的设计方法。

关键词: TCN; 车辆总线; 重复器

中图分类号: TN919.2; U264.91

文献标识码: A

文章编号: 1672-1187(2009)02-0031-03

Design of repeater for MVB network

ZENG zheng, CAI Lei

(School of Transportation Engineering, Tongji University, Shanghai 201804, China)

Abstract: With the rapid development of subway system in China, the application of TCN becomes important, especially for the city transportation. MVB repeater is a key device in the TCN network. Based on the characteristics of MVB, a designing method is described in this paper.

Key words: TCN; MVB; repeater

1 概述

现代的列车车辆控制方式由过去的集中式控制逐渐转向了分布式控制,因此列车控制网络逐步替代了过去的简单连线而成为了如今列车设备通信的主要方式。网络具有信号传输的数量大、数据可靠性高、物理连线资源占用少等优点,这些优点在很大程度上促进了网络通信在列车通信中的应用。在经过了约 10 年的探索研究之后,1999 年国际电工委员会(IEC)结合 UIC 标准和各国现有的标准制式制定发布了针对于列车网络通信的 IEC61375 标准,即 TCN(Train Communication Network)标准。此标准为目前国际上大多数车辆和车辆器件制造商所采用。

针对于不同场合所需的不同功能,TCN 标准定义了两种总线,一种是用于连接可拆分的车辆编组单元的较接式列车总线 WTB(Wire Train Bus);另一种是在车辆中连接各种电气设备的多功能车辆总线 MVB。

在目前的城市轨道交通车辆编组中,车辆拆编重组的情况不多,因此更多时候采用 MVB 总线连接一个车辆单元,将一个单元中多个车辆里的设备连接到一个 MVB

网络中来。目前国际上著名公司如西门子公司等生产的多种车辆都使用了这种一个单元一个 MVB 子网的连接方式,在这样的应用情况下,MVB 要使用更长的电缆进行连接,使用时跨车辆连接在所难免,因此在实际应用中信号存在衰减。为了应对信号衰减,提高信号质量需要特定的设备来对长距离传输的信号进行信号增强中继,这种设备即 MVB 总线重复器。目前实际应用中跨车辆 MVB 网络需要使用重复器,这使得如何设计重复器成为了一个必要的应用问题。

2 MVB 总线通信方式

MVB 总线是连接车辆内电气设备的总线,通信编码方式采用曼彻斯特码,通信速率为固定的 1.5 Mbit/s。TCN 标准规定了 MVB 总线通信物理层的介质有如下三种:1)短距离电气介质 ESD 2)中距离电气介质 EMD 3)光纤介质。其中中距离电气介质和光纤介质是可选的,实际应用上主要采用的是 ESD。短距离电气介质 ESD 采用 RS-485 差分通信,通信距离为 20 m,过长的距离会使信号衰减难以识别,为了使得信号传得更远就需要使用重复器来对信号进行整形放大。

MVB 采用主从方式进行通信,对通信要求实时性很高。总线上有成为总线控制器的总线主设备,在固定的周期中在总线上广播数据帧,这一数据帧我们称为主帧。总线上其他设备为从设备,它们时刻监听总线上的主帧,当确认收到的主帧是让自己发送信息的时候,在最大允许延时前向总线上发出信息帧,这一信息帧为从帧。主帧和相对应的从帧组成一个报文。

MVB 主帧格式为 9 位的主帧起始符,之后是 4 位的功能码,然后为 12 位的地址或者参量,最后是 8 位校验码和终止符,为固定的 33 位。而 MVB 从帧长度包括 33 位、49 位、81 位、153 位或者 297 位 5 种可能,具体为 9 位的从帧起始符、若干位数据和校验符及终止分界符。从设备发送的帧的种类由主设备发给该从设备的主帧中的功能码来决定,不同的功能码对应着不同的从帧类型。功能码(F-code)和对应的从帧格式关系如表 1 所示。

表 1 F-code 和从帧对应关系

主帧			从帧			
F_code	地址	要求	源	长度	响应	目的
0				16		
1			预设	32	过程数据	预设
2	逻辑	过程数据	源的单	64	(由应用	的所有设
3			个设备	128	决定)	备
4				256		
5		保留		—		
6		保留		—		
7		保留		—		
8	设备	主权的移动	提议的主设备	6	主权的转移	主设备
9	所有设备	一般事件(参数)	一个或多个设备	16	事件的标识符	主设备
10	设备	保留	—	—		
11	设备	保留	—	—		
12	设备	消息数据	单个设备	256	消息数据	选择的设备
13	设备组	组事件	一个或多个设备	16	事件的标识符	主设备
14	设备	单个事件	单个设备	16	事件的标识符	主设备
15	设备	设备状态	单个设备	16	设备状态	主设备或监视设备

MVB 帧信号格式如图 1 所示,可以看出 MVB 的帧长度是不固定的。

3 重复器的设计

网络连接设备有很多种,它们分别工作在不同的网络层次中。例如网关工作在网络层,网桥工作在数据链路层,而重复器则是工作在物理层。重复器的作用是复制并放大传输信号,保证信号质量。

从功能上说重复器有两大特点:一是传输双向性,同

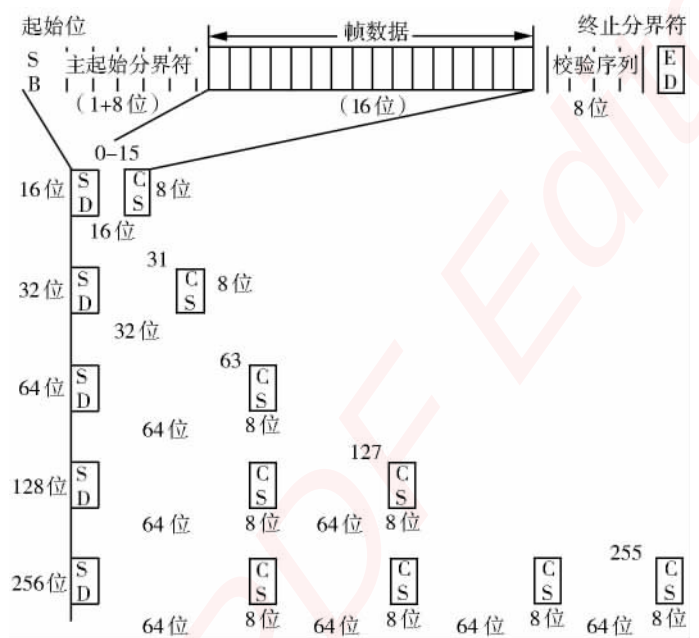


图 1 MVB 帧格式

一条线路在半双工的通信条件下有两种可能的信号流动方向,这就要求重复器有能力判断信号传输方向并及时适应。依据 TCN 协议,重复器要将接收到的第一个信号脉冲方向作为其初始方向并将信号反向发送,无需对信号译码,只需保证物理电平再生;另一特点是防止信号循环。当重复器将线路一端的信号转发到另一端时要防止另一端误以为这一转发信号是该端的输入信号从而将信号送回到输入端,造成信号循环回路,产生信号碰撞。

考虑到 MVB 总线在 ESD 介质中是采用基于双绞线和 RS-485 进行通信的,因此重复器要能识别 RS-485 标准的信号。总线是半双工方式通信的,同一时刻信号只有一个流动方向。鉴于上文所述重复器的特点,要求 MVB 重复器有能力判别 RS-485 信号传输方向,要同时监听总线两端。一旦发现一端是信号源就打开通路将信号送往另一端,同时切断另一端的监听状态并转为发送状态。发送完毕后两端又回到监听状态,从而防止信号循环碰撞。

一些帧长度固定的网络用 RS-485 进行通信时使用的重复器是需要事先计算帧传输时间的,在帧头第一位传到时打开通信通道并开始计时,可以由中断引入单片机或者由定时芯片完成定时功能,计时结束后关闭通信通道并重新监听总线两端。

而 MVB 的帧长度是不定的,可以是 33 位的主帧或者 33 位、49 位、81 位、153 位或者 297 位等 5 种长度的从帧,因而无法事先估计帧传递的时间,从而无法实用常规 RS-485 中继器来完成重复工作。而且根据 TCN 协议,从帧必须在最小 2 μs 最大 6 μs 的时间内响应主帧,而两个主帧之间的时间间隔为 1.3 ms,考虑到 MVB 总线的实际情况,我们按照如下方法设计 MVB 重复器。

这里用来识别转换 RS-485 信号的是普通的 RS-485 芯片 MAX485。芯片总共 8 个引脚, 除开电源和地线外, 有 6 个脚是功能脚: 其中 A、B 脚是 RS-485 的差分信号输入输出脚, RE 为输入使能脚, DE 是输出使能脚, RO 是逻辑电平的输入脚, DI 是逻辑电平的输出脚。

串行总线通信原则是没有信号的时候总线上保持逻辑“1”, 所有的帧信号以一个下降沿为起始位, 帧结束后总线回到“1”态。于是我们用上拉电阻和下拉电阻将输入的 AB 脚分别拉高、拉低, 形成 RS-485 的逻辑 1, 靠输入信号的“0”的变化来传递信号。芯片的 RO 脚在输入为“1”或者没有输入的时候输出高电平, 在输入为“0”的时候输出低电平。

在通信应用中, 起始总线状态是空的, 没有信号传输, 重复器需要实时监听两端的总线信号。一旦第一个信号从总线某一端传输过来, 重复器就要立即将信号传递到另一端。由于起初重复器是监听两端信号的, 当其中一端有信号发送过来后重复器就要转变为一端接收, 一端发送的状态, 直到信号结束。重复器状态的转换取决于对信号传输方向的鉴定, 整个系统的延时由信号采集延时和信号鉴相器的延时组成。由于信号传递速度要求我们必须要在很短的时间内判断出信号的传递方向。因此选用高速鉴相器利用硬件方法判断信号流向, 相较于软件方法来说大大缩短延时时间。系统结构如图 2 所示。

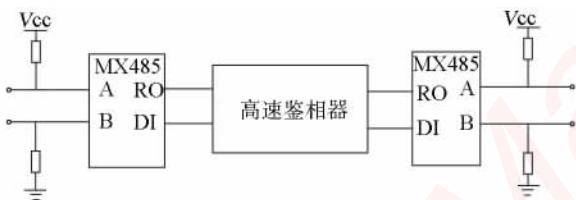


图 2 系统结构框图

信号逻辑传递过程为: 将芯片的转换输出脚通过高速鉴相器和另一头芯片的使能脚相连, 芯片的输出脚则接地, 这样当没有信号传递时两块芯片都是处在输入等待状态, 在高速鉴相器中被认为是两端监听状态, 不改变重复器状态。若有一端信号开始输入, 则该输入端信号发生变化, 信号被拉低, 变化被鉴相器检测到后鉴相器改变

重复器的状态, 将另一端芯片设为输出态, 通过对输出状态的设置此时可以输出 RS-485 的逻辑 0。考虑到可能存在的其它干扰以及对芯片的保护, 为了保证上电后初始状态为总线监听, 将输出通过上拉电阻拉高。即信号不是直接传递的, 而是通过是否拉低输出信号的逻辑来进行传递的, 这种方式符合关于串行通信的规定并且易于控制。

由于一旦检测到了信号, 重复器就会置位接收端和发送端, 发送端不再检测是否有信号输入, 直到发送结束为止。这样就会防止信号形成循环回路产生传输错误。

通过这种方式, 重复器的信号传递精确到了每一个位, 就避开了帧长度不定而带来的定时问题, 也解决了信号循环回路的问题。经过实际检验在 1.5 Mbit/s 的输入信号下, 输出信号稳定, 由于使用的是硬件通信方向算法, 因而系统延迟为硬件电路的固定延迟, 在允许范围内。

4 结论

在实验平台上, 我们输入了从 1 kHz 到超过 2 MHz 的不同频率信号, 并且随机调整信号输入信号频率, 经检测本设备都能够适应。因此, 这种重复器的设计原理适用于帧长度不定的 MVB 总线。本设备在实验平台上能够正常工作, 证明了这一设计方案的可行性。但是考虑到使用高速芯片和 1.5 Mbit/s 的通信速率, 同时 MVB 总线是应用在车辆上面的, 实际工作环境会不理想, 温度湿度和电磁环境都是变化的, 因而系统的稳定性和可靠性还有待长期适应应用的检验。

参考文献:

- [1] IEC 61375, Train Communication Network[S].
- [2] 程凯, 杨晓. 低功耗 RS485 中继器的研制[J]. 单片机与嵌入式系统应用, 2005(5).
- [3] 门雅彬, 韩家新. 一种智能 CAN 总线中继器的设计[J]. 电子产品世界, 2006(1).
- [4] 张雪田, 王建荣. 自制 RS-485 总线中继器[J]. 电子制作, 2004(12).
- [5] 陈可群. 基于单片机控制的 RS-485 中继器[J]. 现场总线与网络技术, 2005(8).

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](#)

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](#)

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](#)

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](#)

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](#)

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](#)

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 COM Express 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](#)

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](#)