RS485-CAN 网关设计与实现

唐祎禄[1]，陈　军[2]

（1. 69046 部队，乌鲁木齐 830001；2. 新疆武警总队，乌鲁木齐 830063）

摘　要：CAN 总线是目前使用较为广泛的现场总线，技术成熟、可靠性高、功能完善、成本合理；RS485 是一种平衡传输的串行通信方式，在集中控制领域大量使用，但是具有许多局限性；RS485－CAN 网关实现由 RS485 总线网络和 CAN 总线网络的互联，同时实现 RS485 总线网络向 CAN 总线网络过渡。

关键词：RS485；CAN；网关；自动控制

Design and implementation of the gateway between RS485 and CAN

TANG Yi-lu[1], CHEN Jun[2]

(1. 69046 Troops of PLA, Urumqi 830001, China; 2. Xinjiang Branch of Armed Police, Urumqi 830063, China)

Abstract: CAN bus is used widely more and more now. It is mature technology, high reliability, comprehensive function and reasonable cost. RS485 is a balanced transmission of serial communication. It is used large-scale in the field of centralized control, but it has many limitations. RS485-CAN gateway has the function of interconnection between the RS485 network and the CAN bus network, while enables transition from the RS485 bus network to the CAN bus network.

Key words: RS485; CAN; gateway; automatic control

0　引言

随着现代控制技术的迅速发展，RS485 局限性日益暴露出来，新的总线技术迅速推广使用，为满足控制系统对实时性、可靠性、灵活性以及成本的要求，原有控制系统向更新的技术和系统过渡，是控制领域不断发展进步的必然趋势。为满足原有 RS485 网络向 CAN 总线网络的顺利过渡，实现两种网络的双向互联，设计 RS485－CAN 总线网关具有现实意义[3-4]。

1　系统总体技术要求

该网关适用于工业控制领域新旧控制系统并存的场合，是一种过渡性设备，实现的功能有：一个 RS485 端口到一个 CAN 端口的转换；CAN 实现 CAN2.0B 协议；CAN 速率可达 500kb/s；RS485 速率可达 9.6kb/s；RS485 及 CAN 口传输距离不小于 1 公里。网关在新老混合的工业控制系统中具体应用示意如图 1 所示。

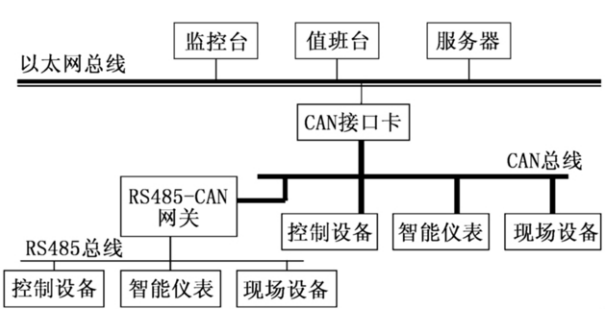

图 1　网关在工业控制系统中的使用

2　系统硬件设计

本装置以 MCS51 单片机为核心，结合 CAN 模块和 RS485 模块，编程实现 CAN 和 RS485 之间的通信[5]。总体硬件结构设计如图 2 所示。单片机选用 ATMEL89C52，其速度、存储容量等功能完全

符合设计要求。

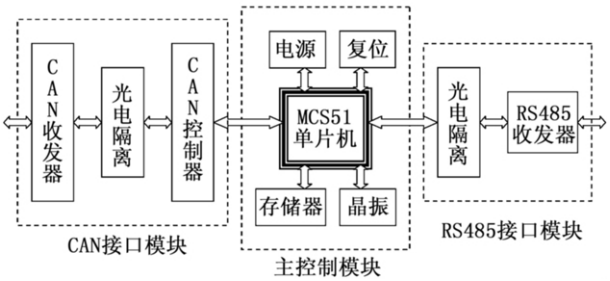

图2 系统总体硬件设计结构简图

CAN 接口模块 CAN 接口模块包括总线控制器和收发器两部分。CAN 总线控制器 SJA1000 是 Philips 公司完全符合 CAN 总线协议规定的 CAN 控制器,支持 PeliCAN (CANBUS 2.0B 模式 收发器 TJA1050 是 CAN 协议控制器和物理总线的接口,是高速 CAN 收发器。SJA1000 的 AD0 - AD7 与单片机的数据总线相连,P2.0 引脚作为 SJA1000 的片选使能端,由 ALE,WR,RD 控制 SJA1000 的数据发送和接收。

高速光耦 6N137 提高总线节点的抗干扰能力,CANH 和 CANL 与地之间并联两个 30PF 的小电容,滤除总线上的高频干扰。

RS485 接口模块 RS485 接口模块采用 MAX485 芯片作为总线收发器,驱动器输出是差分输出,满足 RS485 串行协议的要求,符合 RS485 串行协议的电气规范。同样在单片机与 RS485 之间增加光耦隔离,消除干扰和杂波。

3 系统软件设计

系统软件设计的任务是 实现单个模块的驱动程序和实现两种网络间不同协议的转换。CAN 协议参考模型中定义了物理层和数据链路层,RS485 协议参考模型中只定义了物理层[6]。

3.1 CAN 接口软件设计

CAN 总线数据链路层结构可分为逻辑链路控制 I(LC 子层和媒体访问控制 M(AC 子层。LLC 子层完成帧接收滤波,超载通告和恢复管理等功能,而 MAC 子层功能模块包括发送和接收两个部分。CAN 总线数据链路层的协议集成在 SJA1000 芯片中,具体应用时,只要对 SJA1000 芯片的控制寄存器进行相应的初始化即可进行通信,而数据的接收和发送等也是通过读写 SJA1000 内部寄存器来实现的。单片机对 SJA1000 的初始化程序部分如下所示[7]:

```
/* SJA1000 初始化程序* /
STATUS sja1000Init (CANPORTPTR pDev )
  {
  UCHAR value ;
  canBoardInit ( /* 进入复位模式* /
  canBoard_canOutByte (SJA1000_MOD,MOD_RM )/*;确定 CAN 控制器工作模式* /
  value = canBoard_canInByte (SJA1000_CDR ) ;
  if (pDev - >workmodel = = Peli )value| = PeliCAN ;
  else value & = ~PeliCAN ;* 设置模式寄存器 只在 PeliCAN 模式可访问 *) /
  canBoard_canOutByte (SJA1000_CDR,value |0x48 ) ;
  value = canBoard_canInByte (SJA1000_MOD ) ;
  if (pDev - >sleep = = SNORM )value & = ~ (MOD_SM )/*;睡眠模式* /
  else value| = MOD_SM ;* 正常模式* /
  if (pDev - >lisen = = LNORM )value& = ~ (MOD_LOM )/*;正常模式* /
  else value| = MOD_LOM ;* 单收模式* /
  if (pDev - >test = = TNORM )value& = ~ (MOD_STM )/*;正常模式* /
  else value| = MOD_STM ;* 自检测模式* /
  if (pDev - >filter = = SINGLE )value| = MOD_AFM ;* 单向验收滤波* /
  else value & = (~MOD_AFM )/*;双向验收滤波* /
  canBoard_canOutByte (SJA1000_MOD,value ) ;
  switch (pDev - >baud_rate )
    {
  case B20 :/*波特率为 20kpbs
    canBoard_canOutByte (SJA1000_BTR0, 0x07 ) ;
    canBoard_canOutByte (SJA1000_BTR1, 0x7F ) ;
  break ;
  case B80 :/*波特率为 80kpbs
    canBoard_canOutByte (SJA1000_BTR0, 0x01 ) ;
    canBoard_canOutByte (SJA1000_BTR1, 0x7F ) ;
  break ;
  case B250 :/*波特率为 250kpbs
    canBoard_canOutByte (SJA1000_BTR0,
```

```
0x00 ) ;
        canBoard _ canOutByte (JA1000 _ BTR1,
0x1C ) ;
        break ;
        case B500 ://波特率为 500kpbs
        canBoard _ canOutByte (JA1000 _ BTR0,
0x00 ) ;
        canBoard _ canOutByte (JA1000 _ BTR1,
0x14 ) ;
        break ;
        default ://默认波特率为 20kpbs
        canBoard _ canOutByte (JA1000 _ BTR0,
0x07 ) ;
        canBoard _ canOutByte (JA1000 _ BTR1,
0x7F ) ;
        }
        canBoard_canOutByte (JA1000_CMR,0 ) ;
/* 设置命令寄存器* /
        canBoard _ canOutByte (JA1000 _ EWLR,
0x60 ) /*;设置错误报警限额寄存器* /
        canBoard _ canOutByte (JA1000 _ IER,0 ) ;
/* 不使能所有中断* /
        canBoard _ canOutByte (JA1000 _ OCR,
0xaa ) /*;输出控制寄存器设置* /
        canBoard _ canOutByte (JA1000 _ TXERR,
0x0 ) /*;设置发送错误计数器* /
        canBoard _ canOutByte (JA1000 _ RXERR,
0x0 ) /*;设置接受错误计数器* /
        canBoard_canOutByte (JA1000_RMC,0 ) ;
/* 设置接收报文计数器* /
        canBoard _ canOutByte (JA1000 _ ACR0,
pDev - > code1 ) ;
        canBoard _ canOutByte (JA1000 _ ACR1,
pDev - > code2 ) /*;设置验收代码寄存器* /
        canBoard _ canOutByte (JA1000 _ ACR2,
pDev - > code3 ) ;
        canBoard _ canOutByte (JA1000 _ ACR3,
pDev - > code4 ) ;
        canBoard _ canOutByte (JA1000 _ AMR0,
pDev - > mask4 ) ;
        canBoard _ canOutByte (JA1000 _ AMR1,
pDev - > mask4 ) /*;设置验收屏蔽寄存器* /
        canBoard _ canOutByte (JA1000 _ AMR2,
pDev - > mask4 ) ;
        canBoard _ canOutByte (JA1000 _ AMR3,
pDev - > mask4 ) ;
        return OK ;
        }
```

3.2 RS485 接口软件设计

RS485 接口软件设计 RS485 接口软件设计主要包括串口的初始化，串口中断处理等函数的编写，技术总体比较成熟。其接收程序用中断实现，部分程序代码如下所示[8]：

```
/* 函数名 UART_isr (void )interrupt 1 using 2
功能 串口接收中断处理函数* /
UART_isr (void )interrupt 1 using 2
{CHAR Temp_Buff ;
if (RI )//处理接收中断
  {
  RI = 0 ;//清楚接收中断标志位
  Temp_Buff = SBUF ;
  if (start = = 0 )//判断是否为帧起始位
    {if (Temp_Buff = = 0x88 )//若起始位
正确，则往缓冲区存储数据
      Rbuffer [start + +] = Temp_Buff ;
    Else
      Start = 0 ;//若起始位不正确，则等待
起始位
    }
  else if (start = = 3 )//接收到 RS485 的数
据长度
  Rnum = Temp_buff ;//存储数据长度
  Rbuffer [start + +] = Temp_Buff ;
  else if (start < = 4 + Rnum ) //其他情况
将接收到的数据存储
  Rbuffer [start + +] = Temp_Buff ;
  else start = 0 ;//RS485 一帧数据接收完毕
  }
 if (TI )//处理发送中断
  TI = 0 ;
 TXover = 1 ;//置位发送接收标志
  }
}
```

4 结束语

CAN 总线是当前较为先进、应用较为广泛的现场总线之一，而 RS485 总线是最常用的串行总线，实现 RS485 - CAN 的转换具有现实意义[9]。在综合考虑可靠性、数据处理速度要求、成本控制的情况下，本设计采用基于单片机的设计，侧重于实现 RS485 - CAN 转换器数据的快速转发，

（下转第 67 页）

VSWR)一般要求是前者小于 -6dB,后者小于 3。比较以上两种方式的无源测试结果看以看出 馈点在主板内侧,短路点在主板外侧时,更接近对这两个参数的衡量标准,而且其带宽也很好,能够满足 GSM 带宽 80MHz 要求,所以本实验确定馈点位置为馈点在主板内侧、短路点在主板外侧的方式。

2.3 增益与效率测试结果

H 面测试结果如表 5 所示。

表 5 H 面测试结果

频率 MHz)	880	915	925	960	1710	1750	1840	1880
均值 dBi)	-3.48	-3.71	-5.23	-6.03	-10.29	-8.01	-6.63	-8.41
最大值 dBi)	-1.74	-2.03	-3.35	-4.29	-6.81	-4.67	-2.18	-3.27

E1 面测试结果如表 6 所示。

表 6 E1 面测试结果

频率 MHz)	880	915	935	960	1710	1750	1840	1880
均值 dBi)	-6.51	-6.60	-8.22	-9.18	-8.89	-6.70	-5.55	-7.05
最大值 dBi)	-2.63	-2.77	-4.45	-5.21	-3.35	-1.92	-1.10	-2.77

E2 面测试结果如表 7 所示。

表 7 E2 而测试结果

频率 MHz)	880	915	935	960	1710	1750	1840	1880
均值 dBi)	-5.61	-5.53	-7.1	-8.09	-7.89	-6.06	-5.18	-6.2
最大值 dBi)	-1.87	-2.71	-4.17	-4.94	-2.3	-1.14	-0.72	-2.02

天线效率测试结果如表 8 所示。

表 8 天线效率测试结果

频率 MHz)	880	915	935	960	1710	1750	1840	1880
效率	33%	33%	23%	19%	14%	23%	32%	26%

由表 5 -8 可以看出,本设计的天线增益和效率满足移动终端应用要求。

3 结束语

本文对超薄型 PIFA 的实验过程和测试参数进行了具体的研究。虽然在总体性能上,5mmPIFA 还没有完全达到 7mm 天线高度的指标,但是这个研究的过程为进一步的改善起指导作用。而且,在手机天线的设计中,天线性能与天线环境 机壳材质、金属装饰件、喇叭和马达等金属磁性器件的位置等)有重大关系。本文对一个天线的研发过程具有一定的指导意义。

责任编辑 张禹

上接第 64 页)

重点关注完备的协议转换,而流量控制和差错控制由传输层或应用层等高层实现,这样可以使设计更加趋于简化,便于单片机实现。

参 考 文 献 :

[1] 沈毅斌,陈善章,袁明东. 基于 PIC18 单片机的 RS -232/RS -485/CAN 互联适配器的设计 [J]. 科技资讯,2008 30):.

[2] 张海忠,邓先明,袁启东. 基于 PIC18 单片机的 RS485/CAN 智能转换器的设计 [J]. 现代电子技术,2007 4( )14 ÷16.

[3] 陈啸呐,贺前华,苏勇. 基于 LPC2119 的 RS485 -CAN 总线转换器设计 [J]. 微计算机信息,2006 14)20 :-22.

[4] 蔡子亮,赵忠彪,高荣. 一种新型远程抄表系统的集中抄表器设计及通信协议研究 [J]. 继电器,2007 7( )46 ÷48,53.

[5] 蔡莉,卢珞先. RS -485 通信与 CAN 总线的接口设计 [J]. 武汉理工大学学报,2002 1( )30 ÷33.

[6] 杨永超,佳灵. JP6C -T9 系列变频器 RS485 -CAN 转换器的设计 [J]. 湖北民族学院学报,2009 4( )412:-414.

[7] 蒋加伏,刘清华. RS485 -CAN 转换器的设计 [J]. 微计算机信息,2010 26)171:-173.

[8] 白延敏. 51 单片机典型系统开发实例精解 [M]. 电子工业出版社,2009 268 -292.

[9] RS485 -CAN 协议转换器的设计与应用 [J]. 微计算机信息,2010 20)81 ÷83.

责任编辑 么丽苹

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
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

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

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
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

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

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

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
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

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
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