

文章编号:1008-7842 (2010) 05-0004-03

具有 MVB 接口的输入输出设备的分析

张顺广, 王欣, 张翔

(中国铁道科学研究院 机车车辆研究所, 北京 100081)

摘要 描述了具有列车通信网络(TCN)总线的列车网络控制系统的组成,介绍用于司控台的具有车载多功能总线(MVB)接口的输入输出(I/O)设备,对I/O设备进行详细的需求分析和设计分析,总结了I/O设备的功能特点。

关键词 TCN; MVB; CCU; 列车网络控制; 接收时间

中图分类号: U264.91+1 **文献标志码**: A

列车网络控制系统由网络连接在一起的基于微处理器的分布式智能控制与诊断系统,执行对整车的控制、监测与诊断功能。全球具有列车网络控制系统的机车车辆大部分采用TCN(Train Communication Network,列车通信网络)总线标准。具有TCN总线的列车网络控制系统通常由CCU(Center Control Unit,中央控制单元),具有MVB接口的输入/输出(Input/Output,I/O)设备,网关,中继器及各个具有MVB(Multi-function Vehicle Bus,多功能车载总线)接口的子系统控制器组成。

具有MVB接口的I/O设备一般分为两类,一类是专门用于司控台中,另一类用于车辆的电气柜中。文中主要分析用于司控台中具有MVB接口的I/O设备,司控台上的元件动作结果需要反馈到CCU,供CCU对整车进行集中管理和调控使用;相反,CCU有一部分信号处理的结果,也需要反映到司控台上,用元件指示现在的状态。I/O设备用于连接CCU和司控台元件,应用在具有TCN网络的机车车辆的司控台中,属于智能的I/O设备。

1 I/O 设备的功能需求分析

司控台元件的动作在分布式微机控制系统的机车车辆中是非常重要的接口,信号传输的实时性要求高。司控台上的元件类型:(1)开关、按钮或扳钮,如司机钥匙、升/降弓按钮、主断按钮、方向扳钮等,对于CCU控制来说属于输入信号;(2)手柄,如牵引/制动手柄和速度手柄,对于CCU控制来说属于输入信号;(3)动作元件,如蜂鸣器、指示灯,小型的电流接触器等,对于CCU控制来说属于输出信号。

根据对司控台元件的分析,I/O设备具有以下功能:(1)符合TCN标准,具有MVB接口。(2)对司机

操纵台上的按钮、开关和速度手柄等司控台元件进行二进制数字信号采样,然后通过MVB接口传送到MVB网络上,CCU从MVB网络上获得司控台的信息,然后对整车进行集中管理和调控。(3)从MVB网络上接收CCU发出的控制指令,控制司控台的指示灯或蜂鸣器等元件动作。

2 I/O 设备的设计分析

2.1 输入设计分析

司控台上的元件二进制数字输入信号分为两种类型:二进制输入信号和高速二进制输入信号。

(1) 二进制输入分析

司控台上大部分元件的信号属于此类,例如钥匙位置、升弓按钮、主断路器按钮、方向扳钮等。该类输入信号的变化较为缓慢,用1位来表示一个开关的状态,各个位之间并无直接的逻辑关联。

图1是二进制输入的结构框图。I/O设备设有32个通道的二进制输入。根据单个通道采样电流的大小,把32个通道的电源供电分成2组,每组电源供电负责16个通道。以组为单位控制电源供电的电流。每组供电都有过流保护的功能,当组电流值超过设定的限值时,保护电路起作用,切断该组的电源供电,起安全保护作用。经过设定的时间后,电源监控尝试再次接通断开的电源,如果过流或短路已经消除,电路恢复正常工作。按组电流设定保护,使得硬件电路设计既简单,采样通道又有保护功能。同时I/O设备具有对电源供电的采样功能,能够读回每组的电源供电状况,供诊断使用。例如,CCU读到的输入数据中,相应的电源供电诊断位没有被置位,就忽略该组的采样数据,这样提高了采样数据的真实性。每个通道的输入经过电阻分压,光耦隔离,RC滤波和施密特非门进行整形,状态值储存

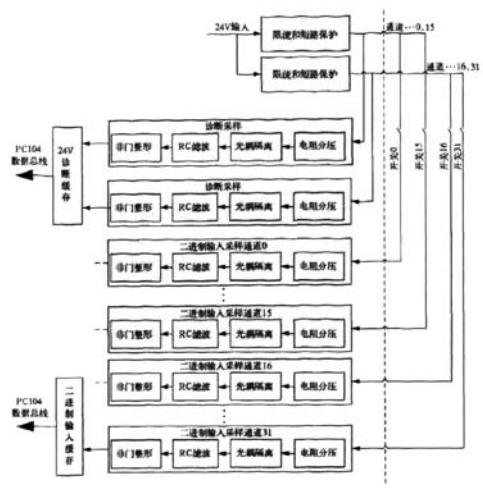


图1 二进制输入的结构框图

工作。电源的诊断采样与高速二进制输入采样的步骤相同。高速二进制输入由于信号变化较快,选用的采样电流比二进制输入采样电流小。每个通道的输入经过电阻分压,光耦隔离,RC滤波和施密特非门进行整形,状态值储存在缓存中。CPU通过扩展的PC104总线读入数据。

2.2 输出设计分析

二进制输出激活外部的指示灯/小型的电流接触器、驱动蜂鸣器。具有限流和短路保护的功能。二进制输出的数据输入从MVB接口卡上获得,具有故障导向安全。

图3是二进制输出的结构框图。每个I/O设备设有8个通道的二进制输出。从PC104总线送过来的数据首先用锁存器锁存,再通过光耦进行隔离,然后用MOSFET管驱动芯片驱动MOSFET管输出24V。该驱动芯片同时具有检流的功能。当某一通道的输出电流大于设定的限值时,停止对MOSFET管栅极的驱动,使得该通道24V输出停止,而不影响其他通道的输出。通过调整芯片外接电容的大小设定重试的时间,当过流已经消除,电路恢复正常工作。MOSFET管电压输出与地间接了一个快速恢复二极管,在感性负载断开时起到续流的作用。同时有对输出电压进行采样的功能,读回输出的数据,传输给CCU,供CCU对输出进行诊断。

到缓存中。CPU通过扩展的PC104总线读入数据。

(2) 高速二进制输入分析

牵引/制动手柄和速度手柄的输入信号属于此类型信号。手柄的位置输入具有特定的编码方式,一般采用格雷码形式,多个通道的输入信号并联在一起才能确定手柄的位置。该类输入的信号与普通的二进制输入信号相比变化较快。根据手柄的位置编码要求,用10位的编码表示手柄的位置。

图2是高速二进制输入的结构框图。每个I/O设备设有两个高速二进制输入,每个高速二进制输入具有10个通道。高速二进制输入电源供电同样也设有限流保护功能。通过电源诊断的采样可以读回电源供电状况。当电流值超过限定值或发生短路的时候,断开24V电压。经过设定的时间,电源监控尝试再次接通断开的电源,如果过流或短路已经消除,电路恢复正常

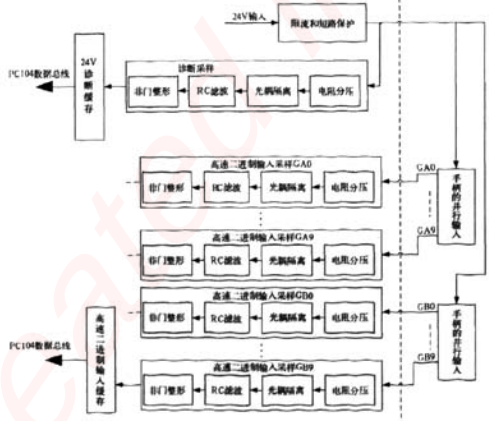


图2 高速二进制输入的结构框图

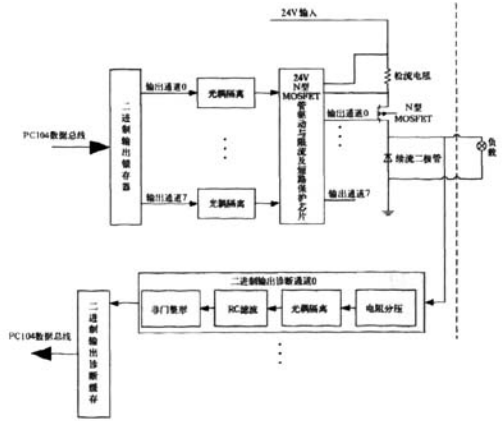


图3 二进制输出的结构框图

2.3 MVB功能分析

I/O设备通过集成有PC104接口的MVB网卡实现MVB接口。MVB网卡一般采用FPGA芯片技术,符合IEC 61375标准的串行通信板卡。MVB网卡的硬件结构如图4所示。图中FPGA芯片是整个MVB板

卡的核心,它包含以下功能:(1) 作为 MVB 通信的控制器,实现 MVB 通讯协议的各种功能;(2) 实现与 Traffic Memory 的硬件接口;(3) 实现与 MVB 总线驱动芯片的硬件接口;(4) 实现与上层 PC104 总线的硬件接口;(5) 实现与嵌入式 CPU 的硬件接口。

根据 TCN 标准对网络设备进行分类,I/O 设备属于 1 类设备,只有过程数据,没有消息数据的功能。I/O 设备设有一个输入端口,一个输出端口。根据列车网络控制系统对司控台的 I/O 设备要求实时性高的特点,输出端口的轮询周期被设置得更小一些。

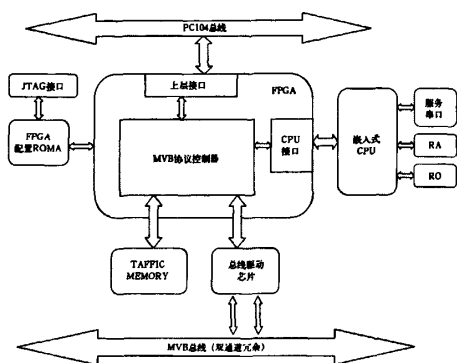


图 4 MVB 网卡的硬件结构图

2.4 系统安全性分析

从系统安全性考虑,I/O 设备设有复位电路和 Sink time(接收时间)监控功能。CPU 要在设定的时间内刷新看门狗的值,否则将触发复位电路,使得整个系统被复位。当 CPU 工作出现异常时,未能在预定的时间内刷新看门狗的值,看门狗定时溢出,复位电路起作用,CPU 被复位,MVB 网卡也被重新复位,二进制输出进入安全模式(一直为 0 V)。

接收时间监控功能,保证了从 MVB 接收的数据新鲜度。当 I/O 设备正常工作时,接收时间监控功能启用。当 MVB 网络通信发生故障时,在设定的时间内,网络通信不能恢复正常,将触发接收时间监控,使得二进制输出进入安全模式。复位所有通道二进制输出到 0 V,并通过指示灯显示触发了接收时间监控。重新接收到有效的二进制数据后,重新启动接收时间监控,二进制输出通道输出接收到的数据,接收时间监控误报警消除。

2.5 系统结构分析

根据对 I/O 设备的功能分析,I/O 设备整体功能结构框图如图 5 所示。I/O 设备的主处理器 CPU 通过

PC104 总线与 MVB 网卡相连。根据 MVB 网卡接口的数据交换位数是 8 位数据接口,所以 I/O 设备上的 CPU 采用 8 位 PC104 总线扩展。CPU 通过 8 位的 PC104 总线写或读 MVB 网卡上相关寄存器实现数据的发送或接收。CPU 通过读输入缓存获得二进制输入的数据和高速二进制输入数据,再通过 MVB 网卡发送到网络上。从 MVB 总线接收的二进制输出数据被 CPU 写到二进制输出的锁存器,然后锁存器的输出通过驱动电路在相应的通道上输出 24 V,驱动外部的设备。

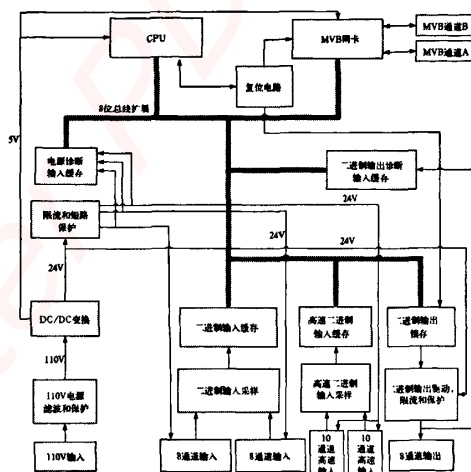


图 5 I/O 设备的结构框图

3 I/O 设备的功能特点

通过上述对 I/O 设备的设计分析,归纳出 I/O 设备具有以下一些特点:(1) 具有 MVB 接口,对端口设置较小的轮询周期,使得输出和输入实时性高;(2) 输入/输出通道数固定,并且所有的输入输出通道都进行了隔离保护;(3) 各输入/输出通道不需要额外的电源供电,且供电电源分组供电和管理,各组的电源都具有限流和过流保护,并具有过流断开和延时重试功能;(4) 各组电源的输出具有诊断输入,二进制输出通道也具有诊断输入,有利于 CCU 对 I/O 设备进行故障诊断;(5) 设有接收时间监控功能,使得二进制输出在网络故障时导向安全输出。(6) 设有复位电路,使得系统出现异常时,整个系统进行完全的复位,二进制输出导向安全输出。

参考文献

- [1] ICE 61375-1. Train Communication Network[S]. 2007.

文章编号:1008-7842 (2010) 05-0007-04

基于“边际轮重”的铁道机车轮重调整算法研究

杨勇军

(南车戚墅堰机车有限公司 产品设计部,江苏常州 213011)

摘要 针对铁道机车的称重调簧计算方法的研究,文章提出了“边际轮重”的概念,建立了铁道机车用于轮重分析的模型和通用方程组,演绎了轴箱弹簧加垫对机车各车轮轮重分布影响(即边际轮重)的计算过程,并以某三轴转向架为例给出了具体的边际轮重计算结果。文章还介绍了基于整车称重试验台实际测到的轮重称重数据和二系悬挂的结构特点,计算理想轮重和理想加垫量的方法,以及对理想加垫量计算结果进行规格化处理的原则。最后给出了所介绍的计算方法的适用范围。

关键词 轮重; 称重调簧; 边际轮重

中图分类号: U260.11⁺5 **文献标志码**: A

针对铁道机车和地铁相关的国家标准,GB/T 3314-2006《内燃机车通用技术条件》、GB/T 3317-2006《电力机车通用技术条件》和 GB/T 7928-2003《地铁车辆通用技术条件》都给出了动轴轴重偏差不超过2%、轮重偏差不超过4%的要求。为满足这一要求,相关生产厂家先后添置了各种称重调簧试验台。按功能和试验对象来划分,这些试验台大体可分为3类:整车称重试验台、转向架称重试验台和车体称重试验台^[1-4]。

轴箱弹簧加垫对各轮轮重的影响是十分复杂的^[5-7]。不仅影响对应车轮的轮重,还影响其他车轮的轮重,尤其是对相邻车轮轮重的影响,更不可忽略。因此,为满足国家标准关于轮重及轴重偏差的规定,根据称重数据,试图采用手工计算的方法计算轴箱弹簧所需加垫量,只能凭经验给出大概的数量,不能给出准确的需加垫量。如果机车质心的位置偏离机车几何中心较多,其对轴重及轮重的影响已占据标准允差的一半以上时,凭经验简单估算得到的结果(包括加垫量和调整后的轴重轮重偏差)十分不可靠。

以机车整车悬挂系统及通过整车称重试验得到的

各车轮轮重数据为分析对象,给出了各车轮理想轮重的计算方法;提出了“边际轮重”的概念,并在此基础上,给出了一种适用于计算机计算的算法,可根据实际称重数据,计算出各轴箱所需的“理想加垫量”。

1 轴箱弹簧加垫对机车各车轮轮重的影响

1.1 边际轮重

边际轮重是某位轴箱一系弹簧增加单位厚度的垫片对各车轮轮重的影响。边际轮重是一组数据,它表示某位轴箱一系弹簧加1 mm的垫片后,各车轮轮重的增量。

边际轮重的数值,可以通过在整车称重试验台上多次重复“加减调整垫——称重”试验过程的方法得到。但是,应该注意到,试验台测量系统的误差,以及试验对象的个体差异,对试验结果的准确程度会有显著的影响。因此,有效的试验方案,需要根据正确的理论分析来制定。下面给出用经典力学分析边际轮重的方法。

1.2 边际轮重的计算

(1) 通用方程组的建立

对两系悬挂的两转向架机车,视转向架构架和车体为刚体,并分别以前转向架、后转向架和车体作为分析对象进行受力分析。分析用坐标系参见图1。

杨勇军(1970-)男,江苏如皋人,教授级高工(收稿日期:2010-04-02)

Analysis of the Input/Output Device with the MVB Interface

ZHANG Shun-guang, WANG Xin, ZHANG Xiang

(Locomotive & Car Research Institute, China Academy of Railway Sciences, Beijing 100081, China)

Abstract: The paper introduces the composition of the train network control system with the train communication network bus, highlights the I/O device with the MVB interface for the driver's desk, and analyses the requirement and design of the I/O device in detail, lastly summarizes the functional characteristic of the I/O device.

Key words: TCN; MVB; CCU; train networks control; sink time

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](#)
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](#)

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](#)

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](#)

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 XPEEmbedded 的警务区 SMS 指挥平台的设计与实现](#)

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 I/O 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](#)

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](#)

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](#)

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

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](#)