

CAN总线的浅析 CANopen协议

Analysis on CANopen Protocol

摘要: 本文分析了 CAN总线的一些特点以及在国内的应用状况,提出了引入国际上通用的 CAN总线高层应用协议,以提高国内 CAN系统应用水平的建议,并简要介绍了一种 CAN的高层协议— CANopen协议。文章最后部分介绍了国际上一些较先进的 CAN产品和开发方法。

关键词: CANopen协议; CAN协议; 嵌入式软件中间件

在设计嵌入式系统,尤其是分布式嵌入式系统时,解决好系统各单元间可靠、有效的通信是系统设计成败的关键。对实时性和安全可靠性的要求,对网络而言就更是如此。解决这一问题有多种方案,如 RS-485 总线、CAN、APN、Profibus、FlexRay、Lora 等不同类型的现场总线,还有嵌入式以太网等。其中,尽管 RS-485 总线协议的性能不高,但由于其在硬件成本以及开发简便性上的巨大优势,目前仍然是国内广泛的总线应用。

随着嵌入式系统应用的发展,

RS-485 性能上的不足逐渐显露出来,已经不能满足设计一个高性能、高实时性系统的要求。尽管还需要实践的证明,但笔者经过多年的观察和实践,感觉到 CAN总线是其中最具有希

望成功的 RS-485 总线、CAN、APN、Profibus、FlexRay、Lora 等不同类型的现场总线,还有嵌入式以太网等。其中,尽管 RS-485 总线协议的性能不高,但由于其在硬件成本以及开发简便性上的巨大优势,目前仍然是国内广泛的总线应用。

选择 CAN总线实现通信的原因

选择 CAN总线作为最佳候选者,主要是基于以下几方面原因:

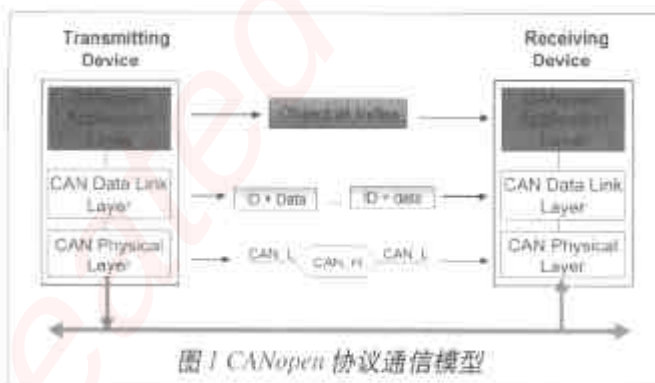
CAN 串行总线具有高性能; CAN 的传输距离可以达到 10 公里;通信速率最高可达 1 Mbit/s。

很高的实时性等等。性能上的优势保证了 CAN 可以应用在很多的领域,在汽车工业、船舶运输、机械控制、工厂自动化、楼宇自动化等都可以看到 CAN 的应用。

CAN 在硬件成本上很具优势。

除了性能外,和其它现场总线相比, CAN 总线在硬件成本上也有很大优势。从硬件芯片上来说,智能节点要收发信息需要一个 CAN 控制器和一个 CAN 收发器。经过多年的发展, CAN 已经获得了国际上各大半导体制造商的大力支持,据 CAN 最主要的推广组织 CAN 自动化 (CANopen) 统计,目前已经有 20 余种 CAN 控制器和收发器可供选择,片内集成 CAN 控制器的单片机更多达 100 余种。CAN 在开发成本上的优势也很明显。

目前,从广泛应用的 8 位 / 16 位单片机,到 32 位的 PowerPC A 等嵌入式处理器,均在芯片内部含有 CAN 总线硬件接口单元。因此,从硬件角度看, CAN 具备其它现



完善的错误检测机制;采用“多重访问冲突仲裁”机制的帧传输方式,可保证不丢失信息;每一帧中最多可以传输 8 字节数据,可提供



图2 CANopen 设备模型

场总线无法比拟的高集成化优势和广泛的市场支持基础。

C的开发平台也比较简单,用户如果选择普通单片机加上C控制器进行开发,则C的开发平台和普通单片机的开发平台完全相同;如果选择带有片内C控制器的单片机进行开发,则只要换用支持该单片机的仿真器就可以了,其他开发设备完全相同。开发C也需要相应的驱动程序,用户可以自行根据选择的C控制器开发驱动程序。

通过采用高层协议将C的应用推向深化

和其他的现场总线相比,C只定义了物理层和数据链路层的规范(遵循ISO标准),这种设计和C规范定义时的历史条件有关,也可以使C能够广泛地适应不同的应用条件,但必然给用户应用带来一些不便。用户在应用C协议时,必须自行定义高层协议。

如何将C协议的应用推向更深的层次,同时满足产品的兼容和互操作性?国际上通行的办法是发展基于C的高层应用协议,只用在应用层上,不同公司的产品才可能实现互操作。好的应用层协议更可以为用户带来系统性能的飞跃。

在C总线协议飞速发展的2年中,很多领域都制定了C在该

J等。这些协议和规范对C的推广起了很大的作用,但总体来说,

协议的模块化特性都不太好,一般只能应用于特定的领域。为了能够把

C推广到更多的领域,欧洲一些公司推出了C应用层(C协议N尽)

管C在理论上正确,并在工业上可以投入应用,但每个用户都必须设计

一个新的子协议,因为C是一个真正的应用层协议。C可以被看作一个应用

C方案的必要理论步骤,但在这一领域它不会被推广。从

1年起,由3B公司领导的一个

欧洲机构研究出一个协议原型,由此发展成为C A规范。p e n

C A是一个基于Cn的好L

协议,采用面向对象的思想设计,具有很好的模块化特性和很高的适应性,通过扩展可以适用于大量的应用

领域。在C A规范基本完成之后,B d将其移交

C组织,由其进行维护与发展。在1年,9C 5 I是一个C A设备的核心部分,对

发表了完整版的C A通信子协议;仅仅用了年的时间,它已成为全欧洲最重要的嵌入式网络标准。

C A不仅定义了应用层和通信子协议,而且为可编程系统、不同器件、接口、应用子协议定义了大量的行规,遵循这些行规开发出的

C A设备将能够实现不同公司

产品间的互操作。另外,C A协

议是免许可证的,任何组织和个人都

领域应用时所采用的高层协议规范。其中,比较著名的有美国汽车工程师协会

(制定的车)

内通信规范

可以开发支持C A协议的设备e n

而不用支付版权费,这也是C A得o p e r

到迅猛发展的重要原因之一。C A目前已在汽车工业控制系统、公共交通运输系统、医疗设备、海

运电子设备和建筑自动化系统中取得了广泛的应用,是将C应用推向深化的理想选择。

采用C A协议o p e n实现通信

C A协议中包含了标准的应用层规范和通信规范,其通信模型

如图所示。在C A的应用层,e n

设备间通过相互交换通信对象进行通信。良好的分层和面向对象的设计思想将带给用户一个清晰的通信模型。

C A设备模型e n

一个C A设备模块可以被分为部分,如图所示。

通信接口和协议软件提供在总线上收发通信对象的服务。不同

C A设备间的通信都是通过交换通信对象完成的。这一部分直接面向C控制器进行操作。

对象字典描述了设备使用的所有数据类型,通信对象和应用对象。

对象字典位于通信程序和应用程序之间,向应用程序提供接口,应用程序对对象字典进行操作就可以实现

C A通信。理解对象字典的概念是理解C A模型的关键e n

应用程序由用户编写,包括功能部分和通信部分。通信部分通过对对象字典进行操作实现C A通信,p e n

而功能部分由用户根据应用要求实现。

C A网络的通信和管理都

C A网络的通信和管理都

是通过不同的通信对象来完成的。为了能够实现通信、网络管理、紧急情况处理等功能，CAN规范定义了e四类标准的通信对象：

· 进程数据对象 (PDO)

第一类通信对象为进程数据对象。PDO被映射到单一的CAN帧中，使用所有的8字节的数据域来传输应用对象。每个PDO有一个独立的标识符并且可能只被一个节点发送，但它可以被多于一个节点接收。这种模式被称之为生产者-消费者通信模式。PDO可以通过多种模式传送，内部事件、外部时钟、远程帧请求以及从特定节点接收到同步报文都可以启动PDO发送。

· 服务数据对象 (SDO)

第二类通信对象为服务数据对象，该对象可以传输大于8字节的配置信息。也就是说，SDO协议允许传送任意长度的对象。接收者将确认收到的每个段信息，发送和接收者间将建立点对点的通信，称之为客户机-服务器模式。未来，CAN系统将允许快速传输SDO，不必对传送的每个段都进行确认，只要在整个对象传送完毕后进行确认即可。

· 网络管理对象 (NMT)

第三类通信对象是网络管理对象，包括节点警戒对象以及NMT对象。节点警戒对象是由N主节点远程请求发送的带有4字节数据的CAN帧，一个字节的数据中包含1个触发位以及3个用于表示节点状态的数据位。N主节点将周期性地发送节点警戒对象。发送周期、警戒时间的长度在对象字典中规定并且可以通过SDO进行配置。另外，系统还定义了生命警戒时间，N主节点

要在生命警戒时间过后向N从节点发送远程请求。这种机制保证了即使N主节点不在了，系统中的其他节点也可以通过用户定义的方式进行回应。

· 特殊功能对象

CAN还为同步、紧急状态表示以及时间标记传送定义了三个特定的对象。同步对象由同步制造者向网络进行周期性广播，该对象将提供基本的网络时钟。当设备发生严重的内部错误时，相关的一个紧急状态客户机将发送一个紧急状态对象。时间标记对象将为应用设备提供公共的时间帧参考。

要理解CAN规范核心是要理解CAN的设备模型和各种类型的通信对象。掌握了这两者后，通过利用各类标准的设备描述就可以开发出符合国际标准的CAN设备了。

展望

最近一段时期，国内开发、应用CAN系统的人员正在逐渐增多，对CAN协议的研究也在不断加深。在很

多领域，如研制电动汽车和混合动力汽车的8重伏课题，已经将CAN作为标准的车内通信协议确定下来。电力、航天等部门也在CAN方面取得了不小的应用成绩。

在CAN应用蓬勃发展的时候，我们也应当清醒地看到，尽管CAN协议在欧美已经发展了多年，应用层协议的发展也差不多有年时间，但目前国内大多数的应用系统仍然基于CAN规范开发，还不能在应用层的水平上进一步深入，这不能不说是很遗憾的事情。另外，国内研究、开发CAN协议，尤其是CAN高层协议的组织和人员还太少，这对CAN在中国的推广是十分不利的。笔者深切希望更多的有识之士能够加入这一行列。

参考文献

- 1 CAN Draft S (Version 4.02).
- 2 Prof. Dr. - Ing CAN-based High Profiles.

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](#)
65. [具有 MVB 接口的输入输出设备的分析](#)
66. [基于 STM32 的 GSM 模块综合应用](#)
67. [基于 ARM7 的 MVB CAN 网关设计](#)
68. [机车车辆的 MVB CAN 总线网关设计](#)
69. [智能变电站冗余网络中 IEEE1588 协议的应用](#)

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](#)
43. [实时操作系统 VxWorks 在微机保护中的应用](#)

44. [基于 VxWorks 的多任务程序设计及通信管理](#)
45. [基于 Tilcon 的 VxWorks 图形界面开发技术](#)
46. [嵌入式图形系统 Tilcon 及应用研究](#)
47. [基于 VxWorks 的数据采集与重演软件的图形界面的设计与实现](#)
48. [基于嵌入式的 Tilcon 用户图形界面设计与开发](#)
49. [基于 Tilcon 的交互式多页面的设计](#)
50. [基于 Tilcon 的嵌入式系统人机界面开发技术](#)
51. [基于 Tilcon 的指控系统多任务人机交互软件设计](#)
52. [基于 Tilcon 航海标绘台界面设计](#)
53. [基于 Tornado 和 Tilcon 的嵌入式 GIS 图形编辑软件的开发](#)

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](#)
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 XPEmbedded 的警务区 SMS 指挥平台的设计与实现](#)
28. [基于 XPE 的数字残币兑换工具开发](#)

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 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](#)
27. [基于 PowerPC 和 DSP 的电能质量在线监测装置的研制](#)
28. [基于 PowerPC 架构多核处理器嵌入式系统硬件设计](#)
29. [基于 PowerPC 的多屏系统设计](#)
30. [基于 PowerPC 的嵌入式 SMP 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 \$\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 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 CPU/GPU 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](#)
34. [基于 GPU 的瑕疵检测](#)
35. [基于 GPU 通用计算的分析与研究](#)
36. [面向 OpenCL 架构的 GPGPU 量化性能模型](#)
37. [基于 OpenCL 的图像积分图算法优化研究](#)
38. [基于 OpenCL 的均值平移算法在多个众核平台的性能优化研究](#)
39. [基于 OpenCL 的异构系统并行编程](#)
40. [嵌入式系统中热备份双机切换技术研究](#)

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](#)
10. [基于小波变换与偏微分方程的图像分解及边缘检测](#)
11. [基于图像能量的布匹瑕疵检测方法](#)
12. [基于 OpenCL 的拉普拉斯图像增强算法优化研究](#)
13. [异构平台上基于 OpenCL 的 FFT 实现与优化](#)

FPGA / CPLD:

1. [一种基于并行处理器的快速车道线检测系统及 FPGA 实现](#)

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

2.

WeChat ID: kontronn