

基于 CANopen 的列车通信网络实现研究

在分析原有列车和城轨车辆通信网络技术的基础上，将 CAN 总线技术应用到列车通信网络中，并对其高层协议 CANopen 进行了剖析，以车载空调系统为例，对基于 CANopen 协议的列车通信网络进行了研究与实现。最后通过 PLC 设备 CPU727、CPU725 以及应用软件对基本通信方式进行了验证。

王 奇 刘志刚 彭权威 符伟杰/西南交通大学电气工程学院

高速列车作为一种现代轨道交通工具，集高速化、自动化和舒适化等特点于一身，其通信网络是实现这些特点的关键技术之一，国外高速列车如德国的 ICE 与法国的 TGV 等都装有完备的通信网络，列车通信网络已成为高速列车的重要组成部分。

CAN 是一种有效支持分布式控制和实时控制的串行通信网络，它有着很高的实时性、可靠性、抗干扰能力和检错能力，并且具有比其他现场总线低的开发费用^[2]，适合用在分布式控制系统中。它在汽车以及城轨车辆的控制网络中已经得到了很好地应用，但它只对网络的物理层和数据链路层进行了规范，没有对应用层做相应的说明，CANopen 正是对这一点的补充，它对 CAN 的应用层做了具体规定，实现了不同厂商设备的互用，它被认为是在基于 CAN 的工业系统中占领导地位的标准^[3]。

本文对基于 CANopen 的列车通信网络进行了初步分析和设计，以车载空调系统为例对列车通信网络的实现进行了说明，通过施耐德电气的处理器设备及应用软件对简单通信进行了验证。

CANopen 协议介绍

CANopen 是在 CAN 基础上开发的，使用了 CAN 通信和服务协议子集，为分布式控制系统提供了实现方案。CANopen 协议集的两个主要部分是通信子集和设备子集。通信子集规定基本的通信机制及其特性，而设备子集定义不同类型的标准设备。

CANopen 的设备模型分为三个部分：通信接口、对象字典及应用程序。通信接口实现通信设备与总线的连接；对象字典描述所有的数据类型、通信对象以及应用对象，它提供了应用程序和通信设备的连接；应用程序提供了设备内部的控制机制。

CANopen 协议对总线上传输的数据进行了分类，即每一个传输数据都是某一特定类的对象，实现面向对象的程序设计。在 CANopen 中每一类传输对象都有其规范化的格式，从而实现了应用层结构上的统一。CANopen 协议定义了 4 类数据传输对象：管理对象（NMT）、服务数据对象（SDO）、过程数据对象（PDO）和特殊功能对象。

基于 CANopen 的列车通信网络实现

目前国产的 CRH1 动车组采用的是 TCN 标准，即应用 WTB 和 MVB 作为列车总线及车辆总线，采用 TCMS 系统实现整个列车的网络通信管理。CRH2 在 ARCnet 总线基础上，采用 20 mA 电流环的方式实现整个列车的网络，结合 TICS 系统实现全车的控制及通信。CRH5 动车组将 WTB 用作列车总线，车辆总线采用 MVB 与 CAN

关键词/Keywords

列车通信网络 ·
CAN 总线 ·
CANopen 协议 ·
动车组 ·
PLC 设备 ·

总线相结合的方式，同时也采用了 TCMS 系统管理网络。

本文考虑在全列车采用 CAN 总线完成网络的通信与控制，并应用其高层协议 CANopen 实现网络的功能。CAN 总线基于报文帧的模式，采用具有优先级的报文方式，任何时刻任何节点都可能获得总线的使用权。通过将得到的信息预定成具有不同优先级的报文，这样就可以保证车辆上诸如牵引控制等重要信息的传送。CANopen 协议能够方便地实现这样的功能，协议中提供了完整的网络管理和网络监护等功能，使得网络的启动、管理、监护及停止变得简单明了。

下面以八节动车编组上的空调控制单元为例，说明 CANopen 协议在列车通信网络中的应用。

1. 对象字典的建立

CANopen 协议规定每一个网络设备都要有一个标准的对象字典，来说明它所能完成的功能。对象字典的索引区 0000 ~ 0FFF 为设备的数据类型区，2000 ~ 5FFF 为生产商特定协议区，这些在标准中统一规定，不需要对其进行操作。索引 1000 ~ 1FFF 为通信协议区，要完成预定的通信功能，就要对其进行相应操作。

考虑到八节车厢中每节车厢装有一个控制单元，即八个从设备，一个主设备。主设备负责对八个从设备进行控制管理，从设备负责将空调控制单元的实时数据传送给主设备，即主节点要对八个从节点进行管理，所以主节点配置八个服务数据对象 SDO，分别对应完成从节点参数配置等功能。此外主节点在网络工作过程中要完成过程数据的接收和发送，所以配置一个过程数据对象 PDO。主节点对象字典如表 1 所示。网络管理对象以及紧急事件对象采用协议中的默认配置。

从节点配置相对于主节点有所不同，因为从节点在网络的终端设备处，要完成信息的采集以及信息的显示，所以要对其索引区 6000 ~ 9FFF 的标准设备协议区进行规定，在空调控制单元中，通常采用的是标准的 I/O 模块协议，完成数字信号的输入和输出。从节点对象字典如表 2 所示。

表 1 主节点对象字典

主索引	对象	类型	注释
1200	SDO1 (server)	22h	第 1 个 ServerSDO 参数
...
1207	SDO8 (server)	22h	第 8 个 ServerSDO 参数
1280	SDO1 (client)	22h	第 1 个 ClientSDO 参数
...
1287	SDO8 (client)	22h	第 8 个 ClientSDO 参数
1400	R_PDO1	20h	Rx_PDO 通信参数
1600	R_PDO1	21h	Rx_PDO 映射参数
1800	T_PDO1	20h	Tx_PDO 通信参数
1A00	T_PDO1	21h	Tx_PDO 映射参数

表 2 从节点对象字典

主索引	对象	类型	注释
1200	SDO1 (server)	22h	Server SDO 参数
1280	SDO1 (client)	22h	Client SDO 参数
1400	R_PDO1	20h	Rx_PDO 通信参数
1600	R_PDO1	21h	Rx_PDO 映射参数
1800	T_PDO1	20h	Tx_PDO 通信参数
1A00	T_PDO1	21h	Tx_PDO 映射参数
6000	I/O Module1	U32	CiA 标准协议区
6100	I/O Module2	U32	CiA 标准协议区

2. 网络设备模型

设备在对象字典基础上进行网络的通信，按照对象字典中相应索引中的定义进行数据的接收和发送。主设备接收其他所有设备发送来的报文，并对其进行处理，从设备将从 I/O 口接收到的数据对照对象字典中的设备协议区进行分类组装，再通过过程数据的映射参数以 Tx_PDO 的形式发送到网络上，这个对象在网络上以它的 COB-ID (CAN-ID) 表示自己的身份，不同的对象有不同的 COB-ID。当网络上存在着对象 Rx_PDO 时，对应节点地址 (Node-ID) 相同的设备就会以 Rx_PDO 形式将报文接收，并对照对象字典对数据进行处理。设备的收发报文模型如图 1 所示。

网络设备主要由处理器模块、存储模块、收发器以及控制器等组成，其中处理器模块是实现 CANopen 网络功能的核心，网络设备模型如图 2 所

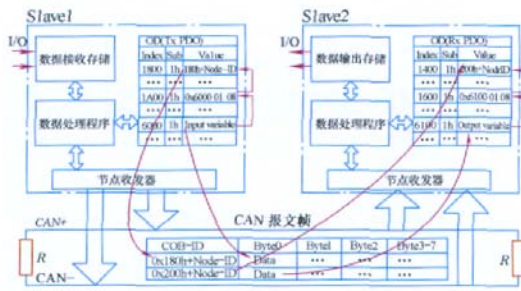


图1 收发报文模型

示。主设备处理器模块负责整个网络的配置、管理以及对网络所有数据的处理，CANopen 协议通过软件或相关程序在处理器模块中发挥作用，数据处理通过相应的处理程序实现。

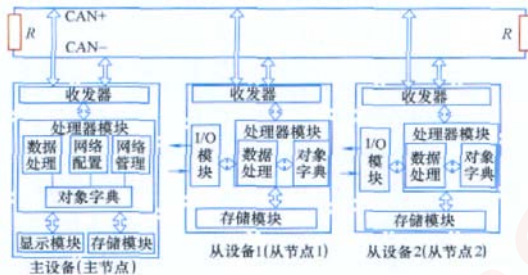


图2 网络设备模型

3. 网络通信管理

网络的启动过程是一个节点设备对自身进行配置，并通过主节点设备对从节点设备进行操作的一个过程，网络成功启动后节点会按照预先定义好的功能进行数据通信，在通信过程中，主节点负责对整个网络进行管理。

网络开启以后，所有从设备都要经历一个从初始化状态到操作状态的过程，主设备在此过程中充当管理者，主设备通过发送不同的网络管理对象（NMT）报文对从设备进行操作，控制从设备在启动过程中的动作，主设备节点与从设备节点启动流程如图3所示。

首先主设备发送模块控制报文：

COB-ID	Byte0	Byte1
0x000	CS	Node-ID

通过 Byte0 位的 CS 值的不同对从节点进行不同的操作，协议规定 CS 值为 1、2、128、129 及 130 分别代表开启远程节点、停止远程节点、进入预操作状态、节点复位及通信复位功能，Byte1 位为被操作节点的地址，为 0 时代表

所有节点。当从节点自身初始化完成后将发送启动成功报文 Boot-up：

COB-ID	Byte0
0x7000 + Node-ID	0

之后主设备通过 SDO 报文对从节点进行参数配置，完成后发送同步对象，使网络中所有节点同步。主节点开始对从节点进行节点监护，主节点先发送无数据

报文：

COB-ID
0x7000 + Node-ID

报文：

COB-ID	Byte0
0x7000 + Node-ID	bit7 + bit6 ~ bit0

其中 bit7 为交替出现的 0 和 1，bit0 ~ bit6 代表发送此报文的当前状态。

节点成功进入操作状态后便开始正常的通信，将采集到的数据经过数据处理程序处理，并组装成相应的 PDO 报文发送到网络中，完成数据的实时传输。

4. 空调控制单元与主设备的通信

对于车载空调控制单元具体功能这里不作详细说明，只关注其通信信号部分。空调控制单元涉及到的信号如图4所示，为便于说明，将信号进行了如图4中所示的编号，用二进制数表示这些信号的意义，1代表该信号有效，0为无效。

这里以点对点的通信模式为例进行说明，并设此控制单元节点地址为 1，其他节点通信与此相同。对于图6中的信号可以将它们映射到 CAN 报文中的数据部分，且只用一位或两位即可实现，其他位可以作为节点中其他模块的功能实现位。对于接收信息即网络中传送的 Rx_PDO 数据映射为

CAN 帧 1 为：

COB-ID	Byte0
0x201h	R7 R6 R5 R4 R3 R2 R1 R0

发送信息 Tx_PDO 映射为 CAN 帧 2 为：

COB-ID	Byte0	Byte1
0x281h	T7 T6 T5 T4 T3 T2 T1 T0	B0

其中 T7 为 空字节，B0 为由二进制数表示的秒数。

节点通过发送 CAN 帧 1，并对 Byte0 中相应的字节数进行操作实现对从节点的控制，从节点发送 CAN 帧 2 将当前数据传送给主节点，主节点据此再进行相应操作。通过 PDO 的传输便完成了主节点与从节点间的数据交换。

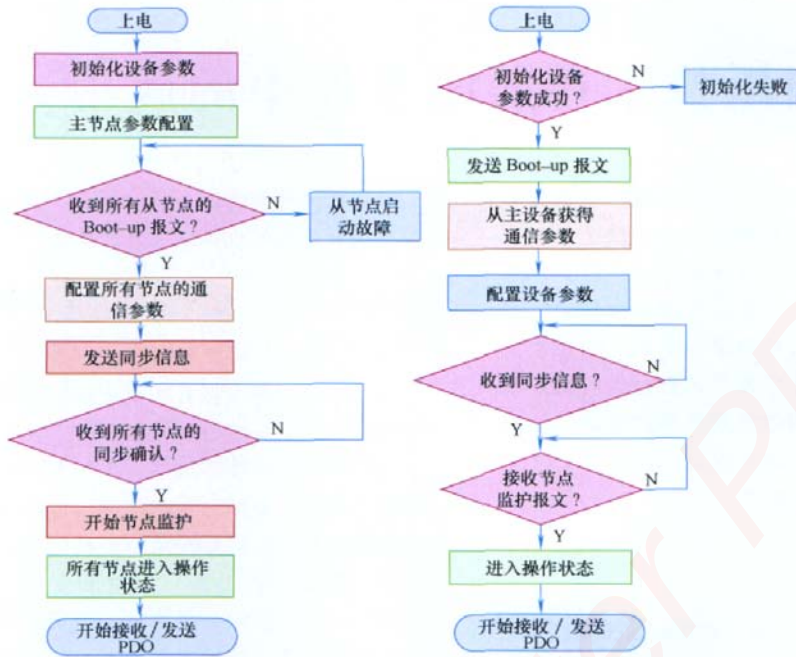


图3 主/从节点启动流程

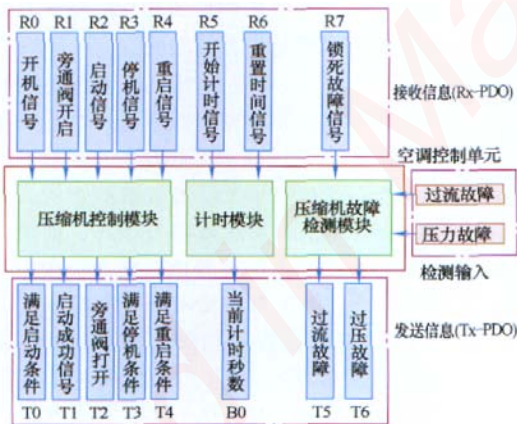


图4 空调控制单元通信框图

实验测试

测试采用施耐德电气 Selectron 公司生产的 PLC 设备 CPU727 作为主设备，从设备采用 CPU725，它们都带有八个输入端口和八个输出端口，方便我们输入模拟数据量以及数据的输出显

示。采用 Selectron 开发的基于 CANopen 并完全符合 IEC 61131 标准的 CAP1131 软件，完成网络的配置、管理以及在线监测等功能。

对空调控制单元传输信息进行编码，并定义其功能，通过编程实现单元功能。采用 PLC 设备的模拟输入输出开关对通信功能进行验证，得到比较满意的效果，实现了主设备与从设备的可靠通信，并达到了对相应功能控制的目的。

结束语

本文对动车组通信网络进行了分析比较，在此基础上提出将 CAN 总线用于全列车的通信网络，结合其高层协议 CANopen 说明了将它们应用于列车通信网络的优点。对 CANopen 协议进行了分析，并将其与列车通信网络相结合，对以 CAN 总线为基础的列车通信网络进行了初步实现，最后采用 Selectron 公司生产的 PLC 设备以及 CAP1131 软件对空调控制单元的通信进行了实验验证，得到比较满意的效果。

参考文献

- [1] 常振臣, 牛得田, 王立德, 等. 列车通信网络研究现状及展望 [J]. 机车电传动, 2005, 28 (3): 5-7.
- [2] 赵华华, 廖志明. 现场总线在列车通信网络中的应用 [J]. 工业控制计算机, 2007, 20 (7): 25-26.
- [3] 孔锋, 张衡, 宋雪桦, 等. 基于 CANopen 协议的汽车控制网络初探 [J]. 汽车工程, 2007, 29 (7): 594-596.
- [4] Hubert D Kirrmann, Ulrich Claessen. IEC 列车通信网络 [J]. 机车电传动, 1999 (3): 4-9.

(收稿日期: 2008-04-21) EA

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](#)
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](#)
70. [CAN 总线的浅析 CANopen 协议](#)
71. [基于 CANopen 协议实现多电机系统实时控制](#)
72. [以太网时钟同步协议的研究](#)

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](#)
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 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](#)
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 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](#)
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](#)
2. [基于 FPGA 和 DSP 的 DBF 实现](#)
3. [高速浮点运算单元的 FPGA 实现](#)
4. [DLMS 算法的脉动阵结构设计及 FPGA 实现](#)
5. [一种基于 FPGA 的 3DES 加密算法实现](#)
- 6.