

机车车辆的 MVB 总线网关设计

李 洋, 石 丽

(大连交通大学 研究生院, 大连 116028)

摘 要: 列车通信的网络组态是多种多样的, 其中 MVB 总线应用就是一种高性能的组网方式。基于对 CAN 和 MVB 报文结构的深入分析, 实现一种 MVB 的网关设计, 有效地完成了两种协议之间的各个网络层次的转换。

关键词: 列车通信网络; 控制局域网 (CAN); 多功能车辆总线 (MVB); 网关

中图分类号: TP393 文献标识码: A

Design of locomotive MVB-CAN network gateway

LI Yang, SHI Li

(Postgraduate College, Dalian Jiaotong University, Dalian 116028, China)

Abstract: The network configuration of train communication was various, among them the connection of MVB-bus and CAN-bus was one kind of high-performance networking mode. It was carried out one kind of MVB-bus and CAN-bus gateway design based on thorough analysis to CAN and MVB data message structure, completed the conversion of each network layer of two kinds of protocol efficiently.

Key words: train communication network; controller area network; multifunction vehicle bus; gateway

多功能车辆总线 MVB 以其高实时性、可靠性、可管理性等多方面的优势, 广泛应用于列车总线控制当中。但由于 MVB 是专门针对列车通信网络而开发的, 其适用范围、供货商、经济性均不如 CAN 总线。CAN 总线以其强有力的检错与抗干扰能力, 广泛应用于各种电气设备的通信网络。因此, 目前很多铁道机车车辆的列车通信网络系统都采用了以 CAN 总线和 MVB 共同组成的异构网络形式。而作为两种总线连接节点的 MVB 网关也越来越受到人们的关注。

本文基于对 CAN 和 MVB 报文结构的深入分析, 实现了 MVB 的网关设计, 有效地完成了两种协议之间的转换及总线间的数据传输。

1 CAN 的报文结构

控制器局域网 (CAN) 是国际上应用最广泛的现场总线之一。它为分布式控制系统实现各节点之间实时、可靠的数据通信提供强有力的技术支持。与其它现场总线相比, CAN 总线是具有通信速率

高、容易实现、性价比高等诸多特点的现场总线。

CAN 有两种不同的帧格式, 不同之处为标识符的长度不同: 具有 11 标识符的帧称为标准帧。而含有 29 标识符的帧称为扩展帧。本文主要考虑标准帧的情况。

报文由 4 种不同类型的帧表示和控制: 数据帧、远程帧、出错帧和超载帧。

数据帧由 7 个不同的位场组成: 帧起始、仲裁场、控制场、数据场、CRC 场、ACK 场和帧结束。数据帧组成如图 1。

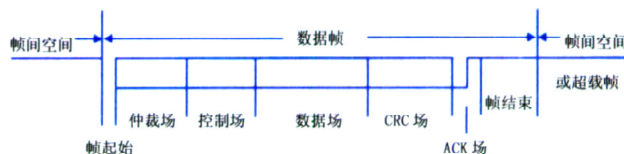


图 1 CAN 的数据帧组成

2 MVB 的报文结构

MVB 是一种主从方式的串行通讯总线, 是可以实现过程控制优化的总线。MVB 能提供最佳的响应速度, 适合用作车辆总线。MVB 层次结构也分为 3 层: 物理层、链路层和应用层。

在 M 中有两种帧格式，一种是只能由总线主设备发送的主设备帧，简称主帧，另一种是为响应主帧而由从设备发送的从设备帧，简称从帧。总线主设备在每一个特征周期里通过发主帧的方式对进程数据进行轮询，相应的从设备发送从帧进行真正的数据传输。报文由主帧以及为响应此主帧而送出的从帧组成。

一个主帧应以主起始分界符开始，其后为 16 帧数据，接着为 8 校验序列，如图 2。

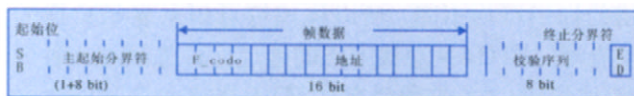


图 2 MVB 主帧

一个从帧应以从起始分界符开始；接着为 16 帧数据，接着为 8 校验序列，当帧数据只有 16 或 32 时则将 8 的校验序列附加其后，如图 3。

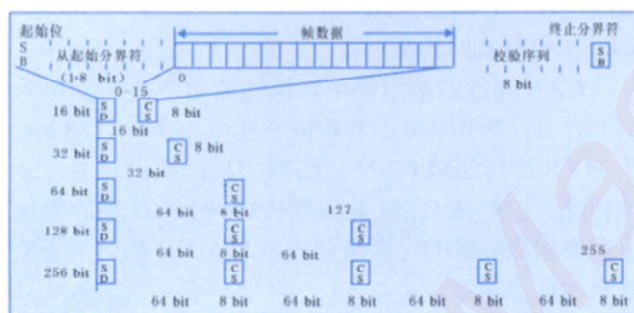


图 3 MVB 从帧

3 网关硬件设计

在本设计中，关键是实现 C 总线和 M 之间的数据交换，其转换装置是一个嵌入式硬、软件转换接口，它具有 C 总线检测和接入功能、M 检测和接入功能，通过嵌入式处理器控制数据帧的解析和重新封装，实现符合 C A N - 规范和 M 标准格式帧的相互转换，同时它还提供一个标准的 R S 的测试口用于应用层软件的调试和检测。

本设计采用嵌入式网关，不仅硬件成本低而且运行更可靠。嵌入式网关基本硬件由微处理器部分、C 总线接口、M 接口组成。

微处理器采用 E C U。它实现对 M 控制 B 芯片以及 C 器件的控制。

C 控制器采用 P H I L I P，主要完成 S J A 1 C 的通讯协议。它支持 11 位标准标识符格式和 29 位扩展标识符格式，具有扩展的 64 接收缓冲器，支持多种微控制器接口模式。

C 收发器采用 P H I L I P。主要提供对总线的差分发送能力和对 C 控制器的差分接收能力。微处理器对 C 控制器进行相应配置后，收发器自动完成相应的 C 总线动作。

M 通信控制器采用 M V B C 专用芯片，它采用 16 数据总线，提供了丰富的接口控制信号，简化了与各种宿主 C 以及通信存储器的接口设计，支持 M 协议中链路层及物理层的功能。

M 物理层接口采用电气短距离介质 E 接口，系统信号通道使用光耦实现主系统与外界的电隔离，从而提高了系统的可靠性。使用 R S 芯片作为收发器，并使用过压保护模块以防止瞬间过压对器件造成损坏。M 物理层接口电路如图 4。



图 4 MVB 物理层接口电路

通信控制器 M 独立于物理层和功能设备，它为总线上的各个设备提供通讯接口和通讯服务。

M 把来自于 M 的串行化信号转换为并行的数据字节，也把需发送的字节交由串行化电路发送到传输介质上。M 可实现数据链路层以及一部分传输层的数据处理，并通过通讯存储器来与上层软件交互。

如图 4，M 端口 C (C M A V B I n p u t V o l t a g e) 和 I (C A M V B I n p u t) 分别为 D M a V t B a 输入数据通道和输入数据通道，B 来自物理层收发器的 V M 信号由此端口送入 M V 中；C M 0 端口 C (C M V B O u t p u t) 是 M 数据输出端口，a c h 数据经此端口将发送至物理层收发器。M 端口 B S e n (S e n) 的输出信号可作为物理层的使能信号，该信号有效时表示一个报文正在通过 M 端口 B O u t (M V B O u t) 输出。D a t a C h a n g e 信号用于 M 端口 B 的数据线和地址线分时合用 P 00 的 8 9 完成对 S J 的读写。当 M V 总线网 C A N

如图 5，8 9 的数据线和地址线分时合用 P 00

口，在 /、C A S L / E 和 D / 几个信号的协助下，

8 9 完成对 S J 的读写。当 M V 总线网 C A N

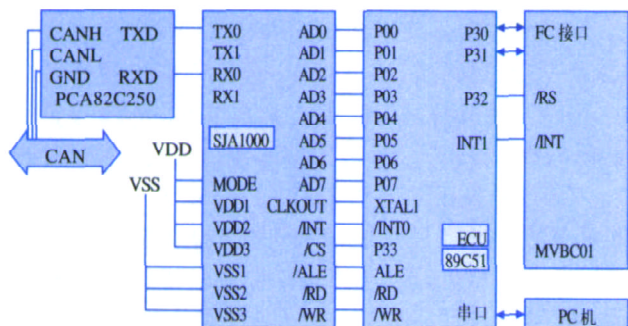


图5 网关系统硬件结构

关节点中 S J 接收到来自 O C 0 总线的数据时，S J 立即使用中断引脚 / 通知 8 9 的 C / 5 1 N 触发中断，由 8 9 的接收中断程序完成数据的读取。

M V 的引脚 0 / 1 除在 N M V 复位时作为 1 复位完成信号，还作为 M V 接收到网络数据时的通知信号，它的状态通过 8 9 的 C / 5 1 传递给 8 9 C 8 5 1 的 C / 5 1 检测 1 / 1，若检测到跳变发生，并且 M V 判断此信号为网络控制数据接收事件，就通过 I 接口读取 M V 的数据。O M 1 V 的复位是通过 8 9 的 C P 触发的，当 M V 复位完成后，通过 / 通知 8 9 C M 5 V 的控制口配置成串行方式的 I 接口，8 9 通过该接口对 M V 的寄存器进行初始化和其它控制操作。由于 8 9 没有 5 I 接口采用 8 9 的 C P 5 3 1 P 口来 1 模拟 I 接口的数据收发。在系统调试过程中，为检测 8 9 模拟 5 I 接口的正确性，将 8 9 写入 5 M V 寄存器的值读回到 8 9 中 C 再通过串口发送到 8 9 的 R S 上，在 3 8 6 的串口调试工具上显示，并与原发送数据比较。

4 软件实现

4 软件设计方案

4 . 网关的网络层次模型

M V 总线网关连接的是两个现场总线网络。对于 M V 总线网关来说，它的工作实际上是对信息重新封装以使它们能够被 C 总线或 M 所读取。为了完成这项任务，网关必须具备 M 和 V 总线所定义的网络层次。通过对 M 和 C 总线模型对比与分析，可以得到以 C A M 网关的基本模型结构，包含物理层、数据链路层、应用层。如图 6。

4 . 应用层的转换

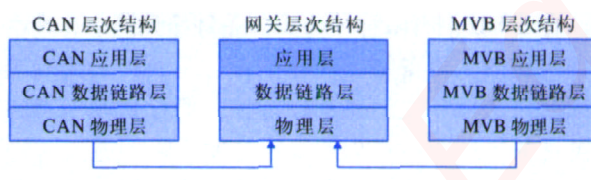


图6 网关的层次结构

为了将 M 总线当中表示过程数据的数据集与 C 的 A N 1 表示数据含义的标识符联系起来，实现透明的传输，每个 C 节点发送的信息帧所使用的标识符均不同，所以标识符可用来表示数据的信息源，即 C 总线节点的地址。首先建立一种规范协议的方式来预先定义 M 的各种变量的含义（应用层的协议），M 的端口地址，长度，以及在端口中数据集的偏移位置等，而后将这些信息按照一定的规则和 C 总线侧的帧的标识符等信息对应成一个表格，这样在 C 侧的数据接收就可以借助这个表格来完成。在此我们把表格定义为协议转换路由表（T A B L E）。E _ I N D E X

4 . 0 帧格式的转换

C 0 C 总线的数据帧中数据场的字节数是 8 b y t e 1 6 4，而 M 从帧中的数据长度为 2 4 ~ 2 b 8 i . 8 t b i 从 M 侧传送到 C 总线侧的数据就需要进行分帧处理，同样从 C 总线侧传送到 M 侧的数据也需要按照一定的规则组合成一个 M 帧。两者帧格式对比见表 1。

表1 MVB 和 CAN 帧格式对比表

帧间空间	数据帧						帧间空间
	帧起始	仲裁场	控制场	数据场 64 bit	CRC 场	ACK 场	帧结束
MSD	主帧数据	FD	SSD	从帧数据	ED		
9 bit	24 bit	2 bit	9 bit	24 bit ~ 288 bit	2 bit		

4 软件实现

软件模块由 C 控制模块、M 控制模块、总调度模块以及两个数据缓冲区 R 组成。M 其软件原理模型如图 7。

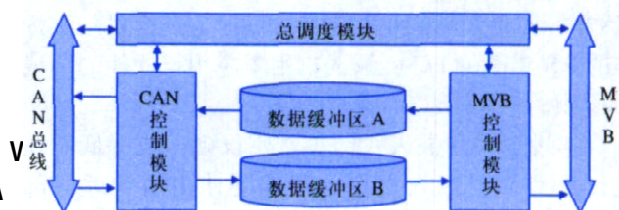


图7 网关原理模型

网关程序的整体设计思路为：当 C 应用层有

数据要发送到 M 网络时，首先，总调度模块检测到 C 数据传输以后，通知 C 控制模块接收数据，解码分析获取标识符，依据标识符查询 T A B L 找出对应 I M M 端口相关变量，然后将报文中的相关数据提取出来发送到数据缓冲区。B 总调度模块通知 M 控制模块从缓冲区中提取数据并进行完整的 M 报文封装，发送到 M 总线上，释放缓冲区。B

反过来，当 M 应用层有数据要发送到 C 节点时，首先，数据发送到 M 上，总调度模块检测 M 上是否有数据传输，通知 M 控制模块接受数据，并对信息解码分析，从中获取端口相关变量，依据端口相关变量查询 T A B L 找出对应 C 标识符，同时将数据发送到数据缓冲区。A 此时，总调度模块通知 C 控制模块从缓冲区中提取数据并进行完整的 C 报文封装，发送到 C 总线上，释放缓冲区。A

C 控制模块主要负责从 C 数据包中解析出完整 C 协议报文，存入数据缓冲区。B 同时，将数据缓冲区中的 C 数据封装成完整的 C 协议报文后发送到 C 总线上。C 控制模块的函数功能描述见表。2

表 2 C 控制模块的函数功能列表

R E C E	接受在 C 总线上传输的数据，并进行解码。	C 总
C H E C	读数据，对数据进行结构分析，看其是否符合 C 总线标准帧结构。若符合，则返回 C 标识符。	线向
T A B L	依据 C 标识符查找比对协议转换路由表，返回 M 端口相关变量。	M
S E N D	将数据场数据发送至数据缓冲区。B	C 控制
R E C E	当缓冲区中有来自 M 控制模块发送的数据时，R A C 控制模块从缓冲区中读入数据。	模块的
E N C	以 C 报文结构对数据封装编码。添加帧起始、仲裁场、控制场、C 场、A 场、帧结束等数据。	操作函
S E N D	将封装完毕的数据发送到 C 总线。	数

M 控制模块主要负责从 M 数据包中解析出完整 M 协议报文，存入数据缓冲区。A 同时，将数据缓冲区中的 M 数据封装成完整的 M 协议报文后发送到 M 上。B

总调度模块主要起到综合调度和监控作用，同时，它还用于整个传输过程中的中断响应。图 说明了总调度模块在 M 到 B 总线传输过程中的调度流程。

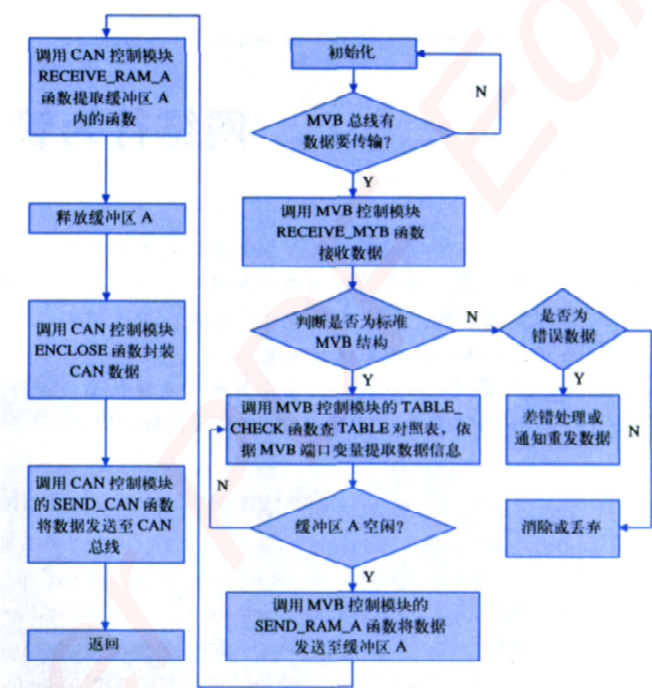


图 8 总调度模块的调度过程流程图 (MVB-CAN)

结束语

本文概述了 M 网络 C 总线网络的报文结构，描述了 M V 总线网关的硬件和软件的实现。通过试验验证，该网关实现了 M 与 V B 总线 N 之间的数据传递。

各种现场总线都有各自的应用特点及优势，所以，多类型总线异构组网的方式应用越来越广泛。M 及 V B 总线的异构组网方式具有广阔的市场应用前景。M V 网关的设计为列车通信网络的多元化发展提供了有力的支持，也为其他网络异构组网方式的网关设计提供了参考。

参考文献：

[邵如峰 李宝玲, 秦贵和, 等 车载 M O S 网络网关 C A N 设计 [汽车技术, 2 0) 0 . 7]

[刘海新 谢维达 . 网络接口单元的应用研究 [工业控制计算机, 2 () 0 . 2]

[刘文清, 王永翔, 王立德 基于 S 的列车通信网卡的设计 [电子产品世界, 2 0) 0 . 3]

[国际电工委员会 . I E C 株洲: 株洲电力机车研 [S] . 究所, 2 0 0 1 .

[李元熙 基于 A 的 MC 总线到以太网网关研究与实现 [南京: 南京理工大学, 2 0) 0 . 2 3 .

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](#)
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](#)

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](#)

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](#)
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](#)

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 技术研究与实现](#)
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](#)

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](#)