

基于 IPv4 / IPv6双栈的 MODBUS/TCP协议实现

王 钢, 巴特尔

(内蒙古工业大学网络中心, 内蒙古 呼和浩特 010051)

摘要: 基于 IPv6的应用不仅会扩展到家电控制, 更为现实的是走入广阔的工业控制领域. 现有国家标准 (GB / Z 19582 3 - 2004) ——《基于 Modbus协议的工业自动化网络规范 第 3部分: MODBUS协议在 TCP / IP上的实现指南》, 是基于 IPv4的实现方案. 本文通过对该实现指南的分析, 针对其中的配置层包和 TCP管理包, 提出了基于 IPv4 / IPv6双栈的 MODBUS/TCP协议实现解决方案, 并给出了基于微软 .net框架、使用 C#设计的控制主机实现关键代码.

关键词: IPv6; MODBUS; MODBUS/TCP; Socket

Modbus协议是一种在工业领域被广为应用的, 真正开放、标准的网络通讯协议, 控制器可以籍此经由网络和其它设备之间相互通信. 早期的 Modbus协议多数承载于 RS485 工业总线上, 由于以太网及 TCP / IP在工业控制领域的渗入, 后推出了基于 TCP / IP实现的 Modbus/TCP协议, IANA 赋予了 TCP端口号 502, 这是目前在自动化行业中唯一分配到的端口号. Modbus/TCP是目前工业以太网事实的标准, 同时也被采纳为我国国家标准, 即 (GB / Z 19582 3 - 2004) 《基于 Modbus协议的工业自动化网络规范》.

虽然 Modbus/TCP是基于 TCP协议的高层数据交换协议, 但目前最新的《MODBUS/TCP协议实现指南》(以下简称《实现指南》), 包括以此为基础的我国国家标准, 都还以 IPv4为基础, 其中仍然涉及了不少 IP层的内容. 而随着 IP技术的不断发展, IPv6提供的大地址空间等特性使人们可将其应用到家庭网络, 实际在工业控制领域中的也同样需要这样的地址体系结构. 因此, 研究基于 IPv4 / IPv6双栈实现的 Modbus/TCP协议, 具有广泛的应用前景.

1 现有 Modbus/TCP协议及实现

Modbus/TCP是 OS模型第 7层上的应用报文传输协议, 一般来说, 管理端作为主设备, 是 TCP客户端, 通过 502端口与服务端 (即从设备) 建立连接, 按照 Modbus规定的请求 / 应答协议规范, 提供 Modbus功能码规定的数据传输服务.

Modbus - DA 现有《实现指南》全面考虑了 Modbus/TCP实现中的各种情况, 面向大型的 Modbus管理端, 如图 1所示, 给出了详细的实现框架, 同时也给出了主要的类及结构体定义. 其中与 IPv4直接相关的是配置层和 TCP管理层, 这其中的核心数据结构为 TConfigObject和 TItemConnexion, 用于全局配置和管理 TCP连接, 其中的 IP地址以 32位长整型存储.

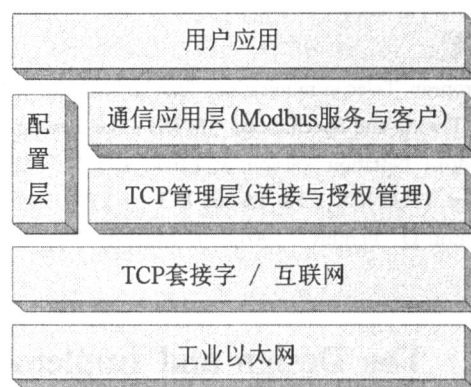


图 1 Modbus/TCP实现体系结构

```
struct TConfigObject { long MyIPAddress; long MyPortNumber; long ListAuthorized_IP[10]; long ListForbidden_IP[10]; .....; };
```

```
struct TItemConnexion { int SocketDescriptor; long IPSource; long PortSource; long PortDest; long IPDestination; char * BufferIn; char * BufferOut; }
```

此外, 《实现指南》中还有一些类的方法中传递了 TItemConnexion结构体类型的变量, 一些方法中也直接使用了 IPv4地址作为参数. 这些是导致该指南仅能适合于 IPv4的根本原因.

2 支持 IPv6 / IPv4 的 Modbus/TCP 实现指南

目前的《实现指南》虽然在标识连接方面、衔接 TCP Stack 方面,很多时候直接使用了 IPv4 地址,但分层还是其主导思想,所以应尽量保持原有体系结构,在能够容纳 IPv6 情况下作必要修改。

首先,对于 IP 地址的存储,由原有的 long 长整形,修改为 16 字节数组 byte[16],具体应用中也可利用结构体和联合体,来适合各种 CPU 的位数,即 struct in6_addr { union { uint8_t _S6_u8[16]; uint32_t _S6_u32[4]; uint64_t _S6_u64[2]; } _S6_un; }. 对于 IPv4 地址,采用映射的方法,前 96 位填充为固定的::FFFF,即::FFFF: <IPv4 地址>的方式存储. 在实现指南中所有直接用到 IP 地址的地方,都修改为此结构体,达到兼容存储两种地址目的。

其次,为表达如 fe80::1%ne0 的接口, IPv6 Socket 环境中必须使用作用域索引 scope_id, 在此采用 FreeBSD 内核的实现方法,强化为链路接口一一对应情况,将索引值嵌入到 16 字节中的第 3, 4 字节,达到统一存储格式的目的。

除上述修改外,为区分 IPv4 / IPv6 地址,以及链路-本地地址 scope_id 情况,分别增加 sa_family, AddressV4Mapped, AddressWithScopeId 标志,来反映实际情况. 修改完成的配置层 TConfigureObject 结构体为: struct TConfigureObject { char AddressV4Mapped; in6_addr MyIPv4Address; in6_addr MyIPv6Address; in6_addr ListAuthorized_IP[10]; in6_addr ListForbidden_IP[10];; }; 分别使用 MyIPv4Address、MyIPv6Address 两个本地配置地址,可以让双栈 Modbus 客户端利用 IPv4 和 IPv6 地址与不同地址家族的 Modbus 服务端设备连接. 修改完成的 TItemConnexion 结构体为: struct TItemConnexion { char sa_family; char AddressV4Mapped; in6_addr IPSource; long PortSource; long PortDest; in6_addr IPDestination; int SocketDescription; char * BufferIn; char * BufferOut; }.

通过上述修改,首先仍可以做到唯一标识连接,同时,在其他函数使用 IP 地址作为参数传递时,新的存储方式可以做到兼容两类地址;其次,在调用 socket API 时,两个结构体可提供 RFC3493 中 sockaddr_in6 结构体中所需全部信息,达到调用支持 IPv6 / IPv4 的 winsock 或 BSD Socket 之目的. 依据上述结构,可以方便实现支持 IPv4 / IPv6 双栈的 Modbus 客户端及服务端。

3 不同开发平台上的具体实现

在使用基于微软 .net 框架等高层开发平台时,它们提供的网络类库已经充分考虑了开发 IPv4 / IPv6 兼容应用程序的情况,因此除前面提到的必要参数外,可扩展高层类对象作为参数,方便调用类库. 下面给出使用 C# 在 .net 框架上实现的 TItemConnexion 类及简单连接代码。

```
class TItemConnexion { //篇幅有限,仅给出源指南中改动过的属性以及扩展属性
    public byte[] IPSource = new byte[16]; //以字节数组形式存放源地址信息
    public byte[] IPDestination = new byte[16]; //以字节数组形式存放目的地址信息
    public IPEndPoint IPSrcEndPoint; //IPEndPoint 类源端节点,包含地址和端口
    public IPEndPoint IPDesEndPoint; //IPEndPoint 类目的端节点,包含地址和端口
    public Socket SocketDesc; //Socket 类套接字,包含连接双方所有配置信息
    public NetworkStream BufferIn; //NetworkStream 类,代替 char * BbufferIn
    public NetworkStream BufferOut; //NetworkStream 类,代替 char * BbufferOut
}, 下面是客户端连接服务端的关键代码,
TItemConnexion aCon = new TItemConnexion();
aCon.sa_family = 6; aCon.PortDest = 502;
aCon.IPDestination = new byte[] { 0xfe, 0x80, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x01 }; //目的 IPv6 地址 fe80::1
aCon.IPDesEndPoint = new IPEndPoint(new IPAddress(aCon.IPDestination), aCon.PortDest);
aCon.SocketDesc = new Socket(aCon.IPDesEndPoint.AddressFamily, SocketType.Stream, ProtocolType.Tcp); aCon.SocketDesc.Connect(aCon.IPDesEndPoint); //建立 socket 并连接
```

在嵌入式下位机中,网络通信常使用 C 语言直接调用 socket API 实现. 根据需要按前述修改意见,通过对新结构体中的特殊信息进行判别,准备参数,即可完成对支持双栈的 socket()、bind() 等函数的调用. 篇幅所限,不再列出详细代码。

4 结 语

通过对 Modbus/TCP 实现指南的分析,比较了多

种支持双栈应用的解决方法,综合考虑兼容性需求及工业领域的易实现需求,给出了《实现指南》中与 IP 相关的关键结构体修改意见,可以适应 IPv6 走入工业控制领域的新需求。

参考文献:

[1] GB/Z 19582.3 - 2004 基于 Modbus 协议的工业自动化网络规范 第 3 部分: MODBUS 协议在 TCP/IP 上的实现指南 [S]. 2004

- [2] Modbus-DA. Modbus Messaging on TCP/IP Implementation Guide V1.0 [S]. 2006
- [3] Anthony Jones, Jim Ohlund Windows 网络编程 [M]. 2 版. 杨合庆,译. 北京: 清华大学出版社, 2002
- [4] FreeBSD 中文计划. FreeBSD 开发手册 [M]. 2006
- [5] RFC2553. Basic Socket Interface Extensions for IPv6 [S]. 1999
- [6] RFC3493. Basic Socket Interface Extensions for IPv6 [S]. 2003

Design and Implementation of Modbus/TCP Based on IPv4/IPv6

WANG Gang, BAATER

(Network Center, Inner Mongolia University of Technology, Hohhot 010051, China)

Abstract: IPv6 technology will be extended to not only Household Appliances but also Industry Control field practically, but the national standard of GB/Z 19582.3 - 2004 is designed based on IPv4 (Modbus Industrial Automation Network Specification—Part 3: Modbus Protocol Implementation Guide over TCP/IP). First, the packages of Configuration Layer and TCP Management in the implementation guide were mainly analyzed, then a new implementation guide of Modbus/TCP based on IPv4/IPv6 dual stack was put forward, at last the key codes for controlling host by C# were presented in detail.

Key words: IPv6; MODBUS; MODBUS/TCP; Socket

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](#)

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](#)

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](#)

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](#)

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](#)

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](#)