

## USB3.0 数据传输协议分析及实现

段俊红, 韩炼冰, 王松, 刘鸿博

(中国电子科技集团公司第三十研究所, 四川 成都 610041)

**摘要:** USB 接口以其简单通用的特点成为消费类连接协议的首选。但随着多媒体技术的快速发展, USB 2.0 已无法满足现在的高速应用需求。USB3.0 以其 5 Gb/s 的传输速率和向后兼容等特点, 成为下一代高速连接标准。根据 USB3.0 协议架构, 文中分别从物理层、链路层、协议层和架构层对 USB3.0 的数据传输协议进行分析, 并与 USB2.0 协议进行比较。最后介绍了基于 EZ-USB FX3 芯片的 USB3.0 加密 U 盘的实现方法。

**关键词:** USB3.0; 传输协议; 加密 U 盘; EZ-USB FX3

**中图分类号:** TN919.3    **文献标志码:** A    **文章编号:** 1009-8054(2013)12-0124-03

### Analysis and Realization of USB3.0 Data Transmission Protocol

DUAN Jun-hong, HANG Lian-bing, WANG Song, LIU Hong-bo

(NO. 30 Institute of CETC, Chengdu Sichuan 610041, China)

**Abstract:** For its simpleness and common-use, USB interface becomes the first choice of consume connect protocol. However, with the fast development of multimedia technique, the USB2.0 couldn't now meet the requirement in high-speed application. With its 5Gbps transmission speed and backward compatibility, USB3.0 becomes the standard of next-generation high-speed connection. In accordance with the framework of USB's protocol, the USB3.0's data transmission protocol is analyzed from physical layer, link layer, protocol layer and frame layer, and also compared with the USB2.0 protocol. Finally, the realization of USB3.0 encrypted USB disk based on EZ-USB FX3 is described.

**Key words:** USB3.0; transmission protocol; encrypted USB disk; EZ-USB FX3

## 0 引言

USB(Universal Serial Bus, 通用串行总线)技术的出现,使原来复杂多样的计算机外设接口变得标准、统一,即插即用、传输速度快等优点<sup>[1]</sup>使其成为消费类连接协议的首选。但随着多媒体技术的快速发展,单个文件的体积越

来越大,USB2.0 接口 480 Mb/s 的传输速率已无法满足日益增长的高速应用需求。2008 年 11 月 12 日,USB3.0 组织发布了 USB3.0 正式标准白皮书。作为下一代高速连接标准,USB3.0 的最大传输速率比 USB2.0 提高了将近 10 倍,可达到 5 Gb/s<sup>[2]</sup>。同时,USB3.0 在可靠性、兼容性和电源功耗管理等方面也具有很大的优势。

## 1 USB3.0 数据传输协议分析

虽然 USB3.0 能向后兼容 USB2.0,但二者的传输协议仍有许多不同。根据 USB3.0 协议架构,文中分别从 4 个层面对 USB3.0 的数据传输协议进行分析。

### 1.1 物理层

USB3.0 采用 9 针脚设计,新增的 4 条数据线可实现

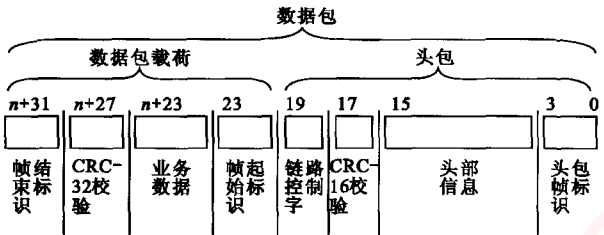
收稿日期:2013-09-23

**作者简介:**段俊红,1984 年生,男,工程师,研究方向为嵌入式系统和网络通信安全技术;韩炼冰,1984 年生,男,助理工程师,研究方向为网络通信安全技术;王松,1985 年生,男,助理工程师,研究方向为嵌入式系统和信息安全;刘鸿博,1982 年生,女,工程师,研究方向为网络通信安全技术。

全双工数据传输,改变了 USB2.0 带方向仲裁转换的单向数据流模式。采用 2 个信道把数据传输和确认过程分离,可以达到更高的传输速度。在数据编码方面,USB3.0 摒弃了原来的 NRZI 编码方式,采用安全性更高的 8b/10b 编码,这种编码方式可以保持电路中的 DC 平衡,提高传输速度,完成嵌入时钟的还原,提高传输的正确率<sup>[3]</sup>。

### 1.2 链路层

信息包是 USB 总线上数据传输的最小单位,包含进行 USB 通信所需的全部信息。一个信息包分为头包和包载荷两部分,头包由头包帧标识、包头和链路控制字 3 部分组成,其中包头由头部信息和 CRC 校验组成,包载荷由框架、业务数据和 CRC 校验 3 部分组成<sup>[4]</sup>。多数信息包只有头包部分而无包载荷部分,图 1 所示的数据包是一种特殊的信息包,它两者都包含。



注: 业务数据的长度为 0~1 024 字节, 所以 n 的取值范围是 0~1 024

图 1 数据包结构

### 1.3 协议层

在 USB2.0 协议中,信息包根据所实现的功能分为令牌包、数据包和握手包 3 类。但在 USB3.0 协议中,信息包已变为链路管理包、事务包、数据包和同步时间戳包这 4 种类型。新定义的信息包以 16 字节的头部开始,包含协议版本、类型、CRC-16 校验和链路控制字等信息,有的包头还含有路由信息和 1 个 3 参数的设备地址信息(设备地址、端点号和方向)。

USB2.0 协议基于信息包定义了 4 种传输类型:控制传输、块传输、中断传输和同步传输。USB3.0 协议沿用了同样的定义,但由于信息包结构发生变化,传输协议也随之做了改进。

USB3.0 控制传输的流程也分为建立、数据和状态 3 个阶段,但不同的是,输出过程中的令牌包被合并到数据包中,输入过程中的令牌包被握手包(ACK 事务包)代替。块传输在 USB3.0 协议中做了扩展,新增了 2 项功能“突

发”和“流”:①突发事务处理。USB3.0 协议允许主机连续发送数据到设备或从设备连续接收数据,中间没有确认包;②流模式。流是 USB2.0 块传输协议在 USB3.0 中的扩展。流模式提供标准块传输管道中对多路传输的多个独立逻辑数据流的协议级支持。中断传输使用标准的突发流程进行可靠的数据传输。同步传输与 USB2.0 一样不发起起始帧,但是时序信息要通过同步时间戳包发送给设备。

同步传输采用的是单向流管道方式,最大突发大小是 16,能在同一个服务周期中请求 3 次突发事务处理。

### 1.4 架构层

USB2.0 协议采用设备架构的概念来描述一个 USB 设备,它把 USB 设备看成是一个配置、接口和端点的集合,并采用标准 USB 描述符来说明它们,其中包括设备描述符、配置描述符、接口描述符、端点描述符和字符串描述符<sup>[5]</sup>。考虑到软件兼容性问题,USB3.0 沿用了以前版本的 USB 描述符定义,再根据实际需要进行了必要的扩展。

## 2 USB3.0 接口加密 U 盘实现

U 盘凭借其体积小、容量大、使用方便等诸多优点,得到了迅速的普及和发展,USB3.0 接口 5Gb/s 的超速模式传输速率更让 U 盘的存取速率有了近 10 倍的提升,甚至比 SATA 接口硬盘的最高传输速率还要高。但 U 盘在获得广泛应用的同时,也为用户带来了极大的安全隐患。由于 U 盘内存储的信息都是透明的,一旦丢失,存储的所有数据可以被任意读取,对有安全要求的用户而言危害很大。

目前,市面上的加密 U 盘大致可分为 3 类:一类是伪加密型,它只是对 U 盘设置了一个访问口令,并没有对 U 盘中存储的数据实施加密;一类是软加密型,它通过配套的主机端加密软件对数据加密后存储到 U 盘内;还有一类是硬加密型,在 U 盘内部完成数据的加密,加密过程对用户透明。

3 类加密 U 盘中硬加密型的安全性最高。但目前商用的硬加密型 U 盘还存在一些问题:加密算法普遍采用国外的公开算法,没有看到使用国密算法的;加密密钥通过将用户输入的认证口令进行变换后得到,这种方式最大的问题在于,一旦用户忘记认证口令将无法恢复 U 盘中存储

的数据;另外,为防止 U 盘扇区损坏导致关联扇区数据无法恢复,加密没有采用安全性更高的 CBC 模式,而是采用了 ECB 模式。

针对目前硬加密型 U 盘存在的问题,文中设计了一种新型的 USB3.0 接口的加密 U 盘,在实现数据快速存储的同时,实现加密 U 盘的口令认证和管理,以及用户数据加密存储的功能,有效地保护了用户信息的安全。

## 2.1 存储加密方案

采用国密对称算法 SM1 对存储数据进行加密,该算法分组长度和密钥长度都为 128 比特,算法安全保密强度及相关软硬件实现性能与 AES 算法相当。

采用安全性更高的 CBC 加密模式。为防止扇区损坏导致关联扇区数据无法恢复,对数据加密的方式进行一些特殊处理:以扇区为单位执行加密操作,以 U 盘扇区号为种子产生伪随机序列,将该序列作为每次加密操作的 IV 使用。

采用口令认证方式,只有通过口令认证才能对 U 盘执行操作。为防止用户忘记口令导致无法恢复 U 盘中存储数据的问题,用户数据加密的密钥使用固化在 U 盘内密钥存储器中的密钥,密钥存储器只有通过了口令认证才能被访问,从而保证了密钥在 U 盘中的安全性。用户输入的口令并不参与加解密运算,只是作为访问密钥存储器的认证口令。为防止暴力破解,当错误口令尝试次数达到限定值后锁定 U 盘,同时设置超级口令来解锁用户口令,超级口令不能修改,不对用户公开。数据加密密钥在 U 盘格式化的时候利用单向函数对其更新,使 U 盘无法恢复得到格式化以前的数据。

## 2.2 硬件平台

加密 U 盘由 USB 控制器、密钥存储器、Flash 存储器、电压转换电路、时钟电路等模块组成,共同完成与主机通信、用户口令认证和加密存储数据等功能。

USB 控制器采用赛普拉斯公司的 USB3.0 外设控制器芯片(CYUSB3014)。该芯片符合 USB3.0 规范,并兼容 USB2.0、USB1.1 和 USB1.0 协议,它集成了一个 USB PHY 和一个 32 位微控制器(ARM926EJ-S),片上 ARM9 CPU 内核可提供高达 200 MIPS 的计算能力,利用 ARM9 内核可完成 USB 设备的枚举、数据处理等功能。

密钥存储器采用的是国产安全芯片 Z32UF,它提供 32

K 字节 EEPROM 存储能力、智能卡专用内核、硬件加/解密和通讯等功能,作为 ARM9 CPU 内核的协处理器使用,通过智能卡专用内核可以实现身份认证、访问控制、密钥数据安全存储的功能。

## 2.3 软件实现

根据加密 U 盘的功能特点,软件划分为如下功能模块:固件框架模块、协议解析模块、访问控制模块、数据加解密模块、Flash 读写模块和用户界面模块。

为加快 USB 软件开发周期,赛普拉斯公司为用户提供了一个固件程序开发框架。该框架可以完成芯片初始化、标准 USB 设备请求处理等功能,用户只要根据实际情况修改少量代码,即可实现定制化的功能。

编译完成的程序文件可以通过外部(并行)FLASH/EEPROM 加载到 FX3 中运行,也可在设备连接到主机时,由主机自动将程序文件下载到 FX3 芯片内 RAM 中运行。通过主机下载方式,无需烧写 FLASH/EEPROM,可以很方便地修改芯片固件程序。

## 3 结语

伴随着信息技术的快速发展,采用 USB3.0 接口的高分辨率视频输入输出设备(显示器、摄像机、蓝光光驱等)、千兆以太网卡和外置硬盘等高速设备将会有更加广泛的应用。对 USB3.0 数据传输协议的分析 and 实现,有助于理解和掌握 USB3.0 接口技术,把握高速连接协议的技术发展方向,对嵌入式环境中 USB3.0 驱动程序的开发有一定的指导意义。

### 参考文献:

- [1] 张平,何向阳. 基于嵌入式指纹采集器 USB 驱动程序研究[J]. 通信技术,2009,42(5):210-212.
- [2] 高振江. USB3.0 通用串行接口技术[J]. 电子元器件应用,2009(7):77-80.
- [3] 李玉伟,潘明海. 8B/10B 编码对高速传输的影响分析[J]. 信息安全与通信保密,2011(3):41-43.
- [4] HP, Intel, Microsoft, etc. . Universal Serial Bus 3.0 Specification[EB/OL]. www.usb.org, July 30, 2008.
- [5] 李英伟,王成儒,练秋生,等. USB2.0 原理与工程开发(第2版)[M]. 北京:国防工业出版社,2007:101. ■

# 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](#)

## 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](#)

## 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](#)

## 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](#)

## 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](#)

## 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](#)

## 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](#)

## 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](#)