

基于 ARM11 的嵌入式视频采集系统设计

许 刚

(贵州师范大学 数学与计算机科学学院, 贵州 贵阳 550000)

摘要:随着多媒体技术的高速发展,视频采集有了更多的实现方式。介绍了一种基于 ARM11 处理器和嵌入式 Linux 的视频采集系统,通过搭建视频服务器,使用 USB 摄像头进行实时采集,通过无线网卡进行网络传输,在客户端可利用 Web 浏览器进行视频播放。该系统能独立完成视频的采集与传输工作,经测试表明,系统设计合理,运行稳定,并可扩展于远程监控系统、智能家居等领域。

关键词:嵌入式系统;USB 摄像头;视频采集;无线传输;V4L2

中图分类号:TN919.85 **文献标识码:**A **文章编号:**1000-8829(2013)12-0037-04

Design of an Embedded Video Capturing System Based on ARM11

XU Gang

(College of Mathematics and Computer Science, Guizhou Normal University, Guiyang 550000, China)

Abstract: With the rapid development of multimedia technology, video capturing has more implementation methods. A video capturing system based on ARM11 and embedded Linux is introduced. An embedded video server is built, and USB camera is used for capturing video real-time. The video stream is transmitted by USB WLAN card. Finally, video stream is played by Web browser in client. This system can independently complete video capturing and transmitting. Test results show that this system design is reasonable, and can run stably. It can be used in other fields such as remote monitoring system, smart home system.

Key words: embedded system; USB camera; video capturing; wireless transmission; V4L2

在当今多媒体技术、电子信息技术高速发展的背景下,视频图像的采集有了更多的实现方式,特别是嵌入式技术的出现使得视频采集系统呈现出小型化、智能化、低成本等特点^[1]。

近年来视频采集系统主要应用于个人与住宅监控、安防等领域。2006 年,AT&T 公司在美国推出家用视频监控设备,用户把手机或者 PC 机作为移动终端,就可随时接收家里的监控信息^[2];在 2008 年北京奥运会上,中国移动提供的无线视频监控服务、中国联通、电信分别推出的“智能家居”、“移动全球眼”业务等,这都极大地推动了视频监控技术的快速发展。

目前嵌入式视频采集系统所提出的方案主要是以 ARM9 为核心,将摄像头采集到的数据进行 MPEG-4/H.264 编码,再以 RTP/RTCP 协议或 TCP/UDP 协议

进行有线网络的传输,最终在 PC 机上进行视频回放及存储^[3]。

已有的研究中大多是以 ARM9 为核心处理器,进行 MPEG-4 或 H.264 编解码,传输大多也是采用有线网络。考虑到实际应用中 ARM9 核心处理器的性能已经不能满足实时多路监控要求,以及存在网络布线不便、成本高等问题。因此,提出一种以 ARM11 为高性能处理器,使用 V4L API 控制 USB 摄像头实现单帧图像和视频连续帧的采集,将采集到的数据进行 JPEG 编码,并在 802.11n 的无线局域网中进行传输,在终端进行视频实时播放的方案。

1 系统总体结构设计

系统主要由 USB 摄像头、嵌入式 Web 视频服务器、USB 无线网卡、客户端 4 部分组成。该系统实现的主要功能是先通过 USB 摄像头实现视频数据的采集工作,它将采集到的原始视频数据进行了 JPEG 压缩编码;编码后的视频数据经由内部总线传送给嵌入式 Web 视频服务器,根据 TCP 协议将打包好的视频数据通过无线局域网进行传输;最后在客户端接收视频数

收稿日期:2013-04-22

基金项目:国家自然科学基金资助项目(61262007);贵州省科技厅省校合作计划项目(黔科合[2013]7001)

作者简介:许刚(1988—),男,湖南常德人,硕士研究生,主要研究方向为嵌入式系统开发。

据,由于 Web 浏览器一般支持 bmp、jpg、gif 格式文件的显示,可以直接在 Web 浏览器里访问所采集到的视频数据。

2 系统的软硬件设计

2.1 硬件结构

嵌入式无线视频采集系统的硬件由主控制板、USB 摄像头和 USB 无线网卡组成,系统硬件结构如图 1 所示。

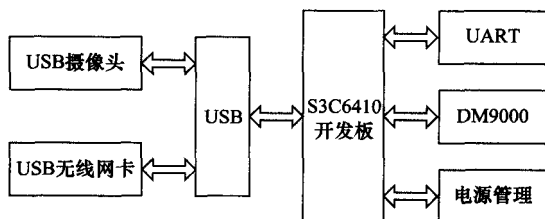


图 1 系统硬件结构图

主控制板包含核心板和扩展板,核心板主要是 S3C6410 微处理芯片,以及 NAND Flash、SDRAM,它们分别用来存储固化程序和系统运行的程序;在扩展板模块中,USB Hub 用来外接 USB 摄像头与 USB 无线网卡,串口 UART 用于程序的下载及调试。

2.1.1 核心处理器

S3C6410 是三星公司的一款 16/32 位 RISC 微处理器,核心时钟频率最高可达 667 MHz,以 ARM1176JZF-S 为内核,拥有 NAND Flash、SDRAM 控制器,4 通道 UART 口、USB2.0 OTG(最高 480 MB/S)等^[4]。S3C6410 对音视频处理有着很好的支持,提供了 MPEG-4/H. 263/H. 264 等编解码器,速率高达 30 f/s。

2.1.2 USB 摄像头

系统选用中星微 zc301 摄像头,采集速率为 30 f/s,内部集成 A/D 转换芯片和 DSP 芯片,输出视频格式为 JPEG 流,分辨率最大可达 640 像素 × 480 像素。摄像头将采集到的数据经内部 DSP 处理后,得到 JPEG 流格式的视频数据,使得数据量大为减少,提高了 USB 传输视频数据的速度。

2.1.3 USB 无线网卡

无线传输主要是通过 USB 无线网卡 TL-WDN3200 实现,该网卡支持 IEEE 802.11n/g/b/a,传输速率高达 300 Mbit/s,频率范围为双频(2.4 Hz, 5 GHz),采用 DBPSK、DQPSK 等调制方式,能满足局域网内视频的传输要求。

2.2 系统软件结构

系统的软件结构主要由嵌入式 Linux 操作系统、相应驱动模块两部分组成。

2.2.1 BootLoader、Linux 内核和根文件系统的移植

该系统引导程序 BootLoader 采用开发板商提供的 U-Boot-1.1.6 版本,通过 TFTP 将 BootLoader 从 PC Linux 上下载到开发板的 NAND Flash 中。Linux 内核选用 Linux-2.6.28,该版本能直接对采用 ZC301 芯片的 USB 摄像头进行支持,免去了移植 USB 摄像头驱动的步骤。在配置内核时,直接添加对 ZC301 系列摄像头驱动模块的支持:

Device Drivers - - - >

Multimedia devices - - - >

Video capture adapters - - - >

USB Video Class (UVC)

UVC input events device support

GSPCA based webcams - - - >

ZC3XXX USB Camera Driver

USB ZC0301 [P] Image Processor and Control Chip

support

设计中使用 NFS 根文件系统,通过使用 NFS 来启动存放在 NFS 服务器中的根文件系统^[5]。其中主要利用 BusyBox-1.20.1 产生 rootfs 目录下的 bin、sbin 和 usr 目录中的命令以及启动程序 Linux C。

2.2.2 无线网卡驱动的移植

对该无线网卡进行驱动移植时,首先需要添加对 USB 无线设备的支持:

Networking support - - - > Wireless 下添加对 802.11 协议的支持

Device Drivers - - - > USB support

USB Wireless Device Management support

- - - > Generic Driver Options 下添加对 mdev 和 fireware 的支持

再将编译的 Ralink RT5572 驱动模块动态加载进内核:

```
insmod rt5572sta.ko
```

2.2.3 Boa Web 服务器的移植

目前 Linux 下主要有 3 个 Web 服务器:Httpd、Thtpd 和 Boa,由于 Boa Web 服务器代码不足 70 KB,有系统资源占用少、单任务性、速度快、性能高等优势,所以 Boa 服务器是最佳的选择。

采用 Boa-0.94.13 版本,先修改 Makefile 文件、Boa 配置文件,交叉编译生成可执行文件并下载到开发板根文件系统,并在根文件系统中创建必要的目录,创建日志文件所在目录/var/log/boa,创建 HTML 文档的主目录/var/www,创建 CGI 脚本所在目录/var/www/cgi-bin^[7]。此外,还要将 Linux 主机下的/etc 目录下的 mimi.types 文件复制到文件系统/etc 目录下^[7]。至此,Boa Web 服务器环境搭建完成。

2.3 应用程序设计

在进行嵌入式软件开发前,需要在主机上搭建好交叉编译环境,使用交叉编译器 ARM-linux-gcc-4.4.3 来生成适合于目标板的可执行文件。应用层程序主要包括视频数据的采集与视频数据的传输。

2.3.1 基于 V4L2 的视频采集程序

V4L2 是 Linux 内核专为图像采集提供的一种系统接口,它是一些视频系统、音频软件的基础,当完成各音视频设备的驱动开发后,就可以调用 V4L 的 API 来控制音视频设备^[8]。视频采集程序的流程如图 2 所示。

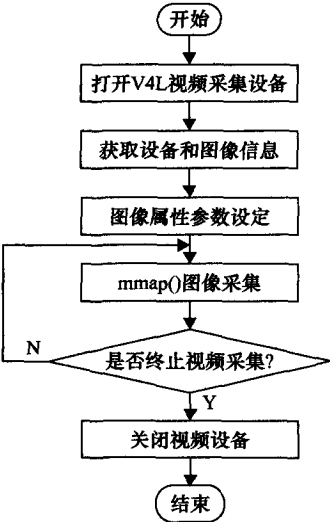


图 2 视频采集程序流程

- ① 打开视频设备,调用 open 函数打开设备文件:
`cam_fd = open( /dev/video2, O_RDWR )`
- ② 读取设备的基本信息,使用 ioctl 调用宏命令 VIDIOCGCAP:
`ioctl( cam_fd, VIDIOCGCAP, &cam_cap )`
- ③ 获取当前图像属性,使用 ioctl 调用宏命令 VIDIOCGPICT:
`ioctl( cam_fd, VIDIOCGPICT, &cam_pic )`
- ④ 改变 video_picture 中分量的值,先为分量赋新值,再调用 VIDIOCSPICT:
`cam_pic.colour = 65535;`
`ioctl( cam_fd, VIDIOCSPICT, &cam_pic )`
- ⑤ 更改设备当前设置,使用 ioctl 调用宏命令 VIDIOCSWIN:
`ioctl( cam_fd, VIDIOCSWIN, &win )`
- ⑥ 使用 ioctl 函数,调用宏命令 VIDIOCGMBUF,获取摄像头存储缓冲区的帧信息:
`ioctl( cam_fd, VIDIOCGMBUF, &cam_mbuf )`
- ⑦ 内存映射,进行视频采集通常有两种方法:read () 直接从设备读取和 mmap () 内存映射^[8]。当使用 read () 方式时,需要从内核缓冲区读取数据;而采用

mmap () 内存映射时,先将设备文件映射到内存中,进程对内存访问时可以像访问文件一样,不需要拷贝任何数据^[8]。mmap () 方式绕过了内核缓冲区,直接把设备文件映射到内存中,提高了系统的效率:

```
cam_data = ( char * ) mmap ( 0, cam_mbuf.size, PROT_READ|PROT_WRITE|MAP_SHARED, cam_fd, 0 )
```

⑧ 采集一帧图像,通过调用宏命令 VIDIOCMCAPTURE 后,按照 video_mmap 里所设置的图像大小、格式等截取一帧视频图像:

```
ioctl( cam_fd, VIDIOCMCAPTURE, &cam_mm )
```

⑨ 图像同步,通过调用 VIDIOCSYNC 等待一帧截取结束,并判断该图像是否截取完毕:

```
ioctl( cam_fd, VIDIOCSYNC, &frame );
```

⑩ 关闭视频设备:

```
close( cam_fd )
```

2.3.2 视频数据的无线传输

目前进行无线网络通信的标准主要有 3G、Wi-Fi、Bluetooth、ZigBee 等^[9],具体详见表 1。

表 1 常用无线通信网络标准的基础比较

类型	协议	最高传输速率/Mbit·s ⁻¹
3G	WCDMA、CDMA2000、TD-SCDMA	2
WiFi	IEEE 802.11 a/b/g/n	11 ~ 300
Bluetooth	IEEE 802.15.1	1
ZigBee	IEEE802.15.4	10

对几种通信标准进行比较可知,WiFi 较其他标准具有传输速率高、支持协议多等优点,所以本系统采用的网络标准为 WiFi。

2.3.3 无线网络的搭建

无线网卡通过在视频监控前端与接收端之间构建无线局域网,实现点对点的连接,采用的是 Ad Hoc 模式。通过 iwconfig 命令来进行无线网卡的配置,具体如下:

- ① 启动无线网卡,使用 ifconfig wlan0 up 命令;
- ② 搜索无线网络,使用 iwlist scanning 命令;
- ③ 设置 wlan0 的 ip 地址和子网掩码,使用如下命令:

```
ifconfig wlan0 192.168.1.30 netmask 255.255.255.0
```

④ 设置 ESSID,使用 iwconfig wlan0 essid xxx 命令区分不同的网络^[10]。

2.3.4 基于 TCP 协议的视频数据传输

本网络采用的是基于 TCP 协议的可靠传输。TCP 协议使用的是流式套接字,它提供了一个面向连接、可靠的数据传输服务,该种类型的套接字能保证数据无重复、无差错地发送,内设置流量控制,避免了双方不同步而造成的数据丢失^[11]。TCP 协议保证了数据传输的正确性和顺序性并且其无长度限制。

在本系统中,首先将视频数据封装成 TCP 数据包,再封装成 IP 数据包在网络中进行传输。然后,通过无线局域网将 IP 数据包传送到接收端。接收端将得到的 IP 数据按相反的顺序将 TCP 报头和视频数据流提取出来。该 TCP 通信流程图如图 3 所示。

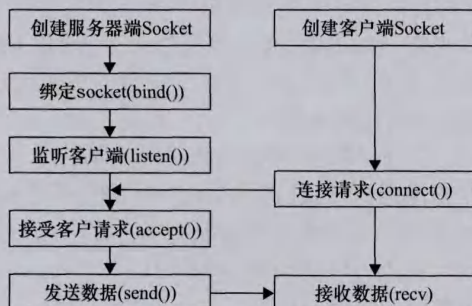


图3 TCP 通信流程图

- ① socket() 函数,用于创建 Socket 描述符;
- ② bind() 函数,使套接口与本地 IP 地址和端口号绑定,告知客户端本服务器的连接信息;
- ③ listen() 函数,监听客户端发送来的连接请求;
- ④ accept() 函数,返回一个新的 Socket 描述符,用来接收和发送数据;
- ⑤ 建立连接成功后,即可调用网络 I/O 进行读写操作:先发送帧头信息,再发送帧图像数据;
- ⑥ close() 函数,断开服务器端与客户端的 Socket 通道。

2.4 客户端通信

本系统在基于 TCP/IP 协议基础之上的应用层协议采用的是 HTTP 超文本传输协议,它实现浏览器与 Web 服务器之间的通信。HTTP 采用的是请求/响应模型,即客户端向服务器发送一个请求,服务器以一个状态行作为响应^[12]。HTTP 协议的工作原理如下:

- ① 建立连接。创建并打开 Socket 文件,让 Web 浏览器和服务器之间建立连接^[12]。
- ② 客户端请求。通过 Socket 通信管道,客户端浏览器向服务器发送请求^[12]。
- ③ 服务器端应答。请求提交后,服务器进行事务处理后,再将结果通过 HTTP 传给 Web 浏览器,并在浏览器上显示出来。
- ④ 关闭连接。当应答结束后,服务器和客户端断开连接,释放资源。

3 实验结果及性能分析

设计采用 Ad Hoc 传输模式,搭建的系统测试环境如图 4 所示。

系统测试是在 WiFi 信号覆盖的范围内完成,在开发板上使用 USB 摄像头采集视频,并通过 USB 无线网

卡将开发板接入无线局域网。由于客户端是在 Linux FireFox 浏览器中进行视频的播放,需要先安装 Java 插件 JDK。安装完成后,在浏览器地址栏中输入:<http://192.168.1.20:888/webcam/index-sample.html> 即可实现视频的实时播放。

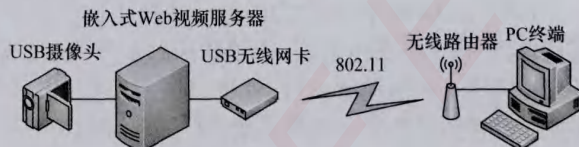


图4 系统测试环境

在该实验环境下,分别在服务器与终端相距 50 m 与 120 m 处进行测试,各测试 3 次,每次发送 1000 帧数据,实验数据如表 2 所示。

表2 实验数据

传输距离/m	接收帧数	丢帧率/%	效果描述
50	996,998,998	0.27	画质清晰
120	113,111,114	88.73	画面模糊

经测试表明,该系统能达到视频采集工程设计的要 求,测试结果如图 5 所示。

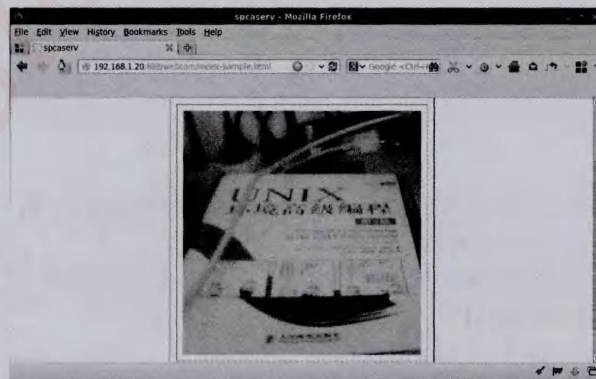


图5 视频传输测试结果

该系统性能分析如下:由于采用无线网络传输,在采集端与接收端之间相隔小于 50 m 时,丢帧率较少,视频画面清晰;在两端距离不断增大时,丢帧率逐渐上升,但在无线网络覆盖范围内,能满足视频监控要求。如需扩大监控范围,则需添加中继节点 AP 进行视频传输。

4 结束语

系统在 S3C6410 开发板、USB 摄像头、USB 无线网卡与嵌入式 Linux 操作系统基础上,实现视频信号的采集与传输。经测试整个系统具有高稳定性、易于安装、成本低等优点,可扩展于工业控制、小区视频监控系 统、智能家居系统等众多领域,具有广阔的应用前景。

(下转第 44 页)

3 系统测试与验证

按图 1 连接测试计算机、数字信号测试模块以及星上被测对象。从测试计算机控制界面读取到的星敏感器、陀螺 LTU 模拟器串行数据均为正确。

在电磁阀脉冲信号端输入电平周期为 5 s、正脉冲宽度为 1.2 s 的脉冲信号的方波信号,测试计算机的“电磁阀脉宽”中应采集到 1.2 s,实测结果为 1.19975 s。

在动量轮转速信号端输入频率为 500 Hz 方波,测试计算机“动量轮转速”的值应为 1250 r/min,实测结果为 1249.7 r/min;输入频率为 1000 Hz 方波,测试计算机“动量轮转速”的值应为 2500 r/min,实测结果为 2499.6 r/min。

在测试计算机设置红外敏感模拟器方波信号参数:“前沿”为 13°、“弦宽”为 30°,红外敏感模拟器方波信号输出结果如图 7 所示。其中,“1”为红外敏感模拟方波输出信号;“2”为 10800 Hz 时钟信号;“3”为 1 Hz 基准信号。示波器时间轴为 20 ms/格。根据式(1)和式(2),在当前姿态下,对应的弦宽值应为 83.33 ms,前沿值应为 5.55 ms。由于红外敏感器头部把角度转换成信号形式时,弦宽偏差 +0.66°,在信号上表示为 1.83 ms;前沿偏差 +0.74°,在信号上表示为 2.05 ms。测试计算机软件设计对这个误差进行了模拟。因此实际测得的弦宽值应比理论计算值大 1.83 ms,即 85.16 ms,示波器示数为 85.18 ms;实际测得的前沿值比理论计算值大 2.05 ms,即 7.60 ms,与示波器示数一致。

4 结束语

根据卫星 AOCS 测试的需要,设计了一个以 FPGA

为核心,适用于 APCI 总线系统的通用测试模块。在该模块上实现了部分卫星部件数字信号的采集与模拟。该模块通用性和移植性好,不仅可以用于其他卫星的测试,也适用于其他领域,具有广泛的应用价值。

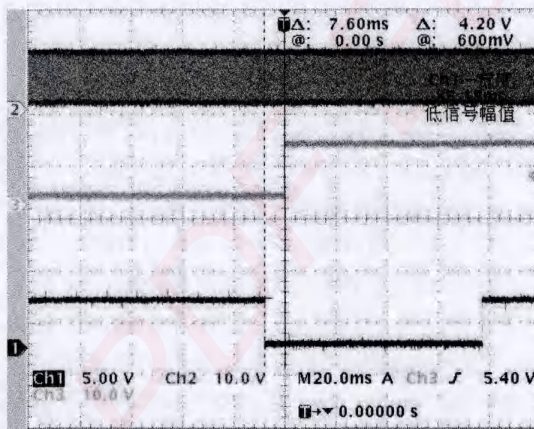


图 7 弦宽信息测量结果

参考文献:

- [1] 潘清,王庆成. 卫星地面测试总控软件设计与实现[J]. 计算机自动测量与控制,1997(2):1-6.
- [2] 黄艳华. 微小卫星地面综合测试系统的设计与实现[D]. 长春:吉林大学,2007.
- [3] 张兴国,王东盛. FPGA 在卫星地面测试设备中的应用[J]. 控制工程(北京),2005(4):33-38,59.
- [4] 杨强浩. 基于 EDK 的 FPGA 嵌入式系统开发[M]. 北京:机械工业出版社,2008.
- [5] 戴立江. 基于 EDA 技术的 FPGA 应用研究[D]. 天津:天津工业大学,2005.
- [6] 江雄杰,张哲,杨军,等. APCI 工控机在发变组微机监控系统的应用[J]. 电力自动化设备,2001,21(5):21-24.
- [7] 高巍,钟梅,徐松惠. 嵌入式 Linux 下基于 BOA 的远程控制[J]. 电脑知识与技术,2009,5(31):8703-8705.
- [8] 李文. 基于 ARM 和 Linux 的嵌入式网络视频监控系统研究[D]. 武汉:武汉科技大学,2011:32-35.
- [9] Chen M, Gonzalez S, Vasilakos A, et al. Body area networks: a survey[J]. Mobile Networks and Applications, 2011, 16(2):171-193.
- [10] Venkateswaran S. Essential Linux Device Drivers[M]. Prentice Hall, 2008.
- [11] 童永清. Linux C 编程实战[M]. 北京:人民邮电出版社,2008.
- [12] 张淮中. 实现基于 HTTP 的 servlet 与 applet 的通信[J]. 计算机工程与设计,2000,21(6):16-19.

(上接第 40 页)

参考文献:

- [1] 韦东山. 嵌入式 Linux 应用开发完全手册[M]. 北京:人民邮电出版社,2008.
- [2] Chen S Q, Wang H N, Zhang X D, et al. Segment-based proxy caching for Internet streaming media delivery[J]. IEEE MultiMedia, 2005, 12(3):59-67.
- [3] 郭巧云,许雪梅,李岸,等. 基于 ARM11 的无线视频监控系统的的设计[J]. 计算机测量与控制,2010,18(8).
- [4] Samsung. S3C6410X 66-YB40 User's Manuals[Z]. 2008.
- [5] 谢伟. 嵌入式 Linux 系统工程师标准培训教材[M]. 北京:清华大学出版社,2011.
- [6] 吴国燕. 嵌入式 Linux 无线视频监控系统的设计与实现[D]. 武汉:华中师范大学,2011:37-38.

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](#)

邀请注册码



关注论坛公众号

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](#)
73. [基于 CANopen 的列车通信网络实现研究](#)
74. [基于 SJA1000 的 CAN 总线智能控制系统设计](#)
75. [基于 CANopen 的运动控制单元的设计](#)
76. [基于 STM32F107VC 的 IEEE 1588 精密时钟同步分析与实现](#)

邀请注册码



关注论坛公众号

77. [分布式控制系统精确时钟同步技术](#)
78. [基于 IEEE 1588 的时钟同步技术在分布式系统中应用](#)
79. [基于 SJA1000 的 CAN 总线通讯模块的实现](#)
80. [嵌入式设备的精确时钟同步技术的研究与实现](#)
81. [基于 SJA1000 的 CAN 网桥设计](#)
82. [基于 CAN 总线分布式温室监控系统的设计与实现](#)
83. [基于 DSP 的 CANopen 通讯协议的实现](#)
84. [基于 PCI9656 控制芯片的高速网卡 DMA 设计](#)
85. [基于以太网及串口的数据采集模块设计](#)
86. [MVB1 类设备控制器的 FPGA 设计](#)
87. [MVB 接口彩色液晶显示诊断单元的显示应用软件设计](#)
88. [IPv6 新型套接字的网络编程剖析](#)
89. [基于规则的 IPv4 源程序到 IPv6 源程序的移植方法](#)
90. [MVB 网络接口单元的 SOC 解决方案](#)
91. [基于 IPSec 协议的 IPv6 安全研究](#)
92. [具有 VME 总线的车载安全计算机 MVB 通信板卡](#)
93. [SD 卡的传输协议和读写程序](#)
94. [基于 SCTP 的 TLS 应用](#)
95. [基于 IPv6 的静态路由实验设计](#)
96. [基于 MVB 的地铁列车司机显示系统研究](#)
97. [基于参数优化批处理的 TLS 协议](#)
98. [SSD 数据结构与算法综述](#)
99. [大容量 NAND Flash 文件系统中的地址映射算法研究](#)
100. [基于 MVB 总线的动车组门控系统的设计与仿真研究](#)
101. [多功能车辆总线 MVB 控制](#)

邀请注册码



关注论坛公众号

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](#)

邀请注册码



关注论坛公众号

53. [基于 Tornado 和 Tilcon 的嵌入式 GIS 图形编辑软件的开发](#)
54. [VxWorks 环境下内存文件系统的应用](#)
55. [VxWorks 下的多重定时器设计](#)
56. [Freescale 的 MPC8641D 的 VxWorks BSP](#)
57. [VxWorks 实验五\[时间片轮转调度\]](#)
58. [解决 VmWare 下下载大型工程.out 出现 WTX Error 0x100de 的问题](#)
59. [基于 VxWorks 系统的 MiniGUI 图形界面开发](#)
60. [VxWorks BSP 开发中的 PCI 配置方法](#)
61. [VxWorks 在 S3C2410 上的 BSP 设计](#)
62. [VxWorks 操作系统中 PCI 总线驱动程序的设计与实现](#)
63. [VxWorks 概述](#)
64. [基于 AT91RM9200 的 VxWorks END 网络驱动开发](#)
65. [基于 EBD9200 的 VxWorks BSP 设计和实现](#)
66. [基于 VxWorks 的 BSP 技术分析](#)
67. [ARM LPC2210 的 VxWorks BSP 源码](#)
68. [基于 LPC2210 的 VxWorks BSP 移植](#)
69. [基于 VxWorks 平台的 SCTP 协议软件设计实现](#)
70. [VxWorks 快速启动的实现方法\[上电到应用程序 1 秒\]](#)

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](#)

邀请注册码



关注论坛公众号

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](#)
52. [Linux 基于 W83697 和 W83977 的 UART 串口驱动开发文档](#)
53. [基于 AT91RM9200 的嵌入式 Linux 系统的移植与实现](#)
54. [路由信息协议在 Linux 平台上的实现](#)
55. [Linux 下 IPv6 高级路由器的实现](#)
56. [基于 Android 平台的嵌入式视频监控系统设计](#)

邀请注册码



关注论坛公众号

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](#)
29. [Windows CENET 下 ADC 驱动开发设计](#)
30. [Windows CE 下 USB 设备流驱动开发与设计](#)
31. [Windows 驱动程序设计](#)
32. [基于 Windows CE 的 GPS 应用](#)
33. [基于 Windows CE 下大像素图像分块显示算法的研究](#)
34. [基于 Windows CE 的数控软件开发与实现](#)
35. [NAND FLASH 在 WINCENET 系统中的应用设计](#)

邀请注册码



关注论坛公众号

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](#)
31. [基于 MPC850 的多功能通信管理器](#)
32. [基于 MPC8640D 处理系统的技术研究](#)
33. [基于双核 MPC8641D 处理器的计算机模块设计](#)
34. [基于 MPC8641D 处理器的对称多处理技术研究](#)

邀请注册码



关注论坛公众号

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](#)
36. [UBoot 在 AT91RM9200 上的移植简析](#)
37. [基于工控级 AT91RM9200 开发板的 UBoot 移植分析](#)
38. [基于 ARM11 和 Zigbee 的人员定位防丢器](#)

邀请注册码



关注论坛公众号

39. [基于 NAND FLASH 的嵌入式系统启动速度的研究](#)
40. [μCOS II 在 ARM7 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 技术研究与实现](#)
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](#)

邀请注册码



关注论坛公众号

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](#)
41. [EFI-Tiano 环境下的 AES 算法应用模型](#)
42. [EFI 及其安全性研究](#)
43. [基于 UEFI Shell 的 PreOS Application 的开发与研究](#)
44. [UEFI Bootkit 模型与分析](#)
45. [UEFI 计算机系统快速调试方法的实现](#)
46. [基于 EFI 系统的多文件系统解决方案](#)

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](#)
14. [数据结构考题 - 第 4 章 串](#)
15. [数据结构考题 - 第 4 章 串答案](#)
16. [用 IPv6 编程接口实现有连接通信的方法](#)
17. [一种战棋游戏的 AI 算法设计与实现浅析](#)
18. [基于 TLS 协议的 ECC 扩展研究](#)

邀请注册码



关注论坛公众号

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. [可编程 FIR 滤波器的 FPGA 实现](#)
7. [基于 FPGA 的 AES 加密算法的高速实现](#)
8. [基于 FPGA 的精确时钟同步方法](#)
9. [应用分布式算法在 FPGA 平台实现 FIR 低通滤波器](#)
10. [流水线技术在用 FPGA 实现高速 DSP 运算中的应用](#)
11. [基于 FPGA 的 CAN 总线通信节点设计](#)
12. [基于 FPGA 的高速时钟数据恢复电路的实现](#)
13. [基于 FPGA 的高阶高速 FIR 滤波器设计与实现](#)
14. [基于 FPGA 高效实现 FIR 滤波器的研究](#)
15. [FPGA 的 VHDL 设计策略](#)
16. [用 FPGA 实现串口通信的设计](#)
17. [GPIB 接口的 FPGA 实现](#)
18. [一种基于 FPGA 的 FFT 阵列处理器](#)
19. [基于 FPGA 的 FFT 信号处理器的硬件实现](#)

邀请注册码



关注论坛公众号