

基于 Android 平台的嵌入式视频监控系统设计

Design of the Embedded Video Monitoring System Based on Android Platform

郑海洲

(上海理工大学光电信息与计算机学院, 上海 200082)

摘要 对现有的视频监控设备进行了调查和研究, 提出一种基于 Android 平台的嵌入式视频监控系统。该系统主要由基于 S3C2440 芯片的开发板和 Android 智能手机两部分组成。目标板用于视频的采集和压缩编码, 并将生成 MPEG-4 视频流发送到 Helix Server 流媒体服务器。Android 智能手机作为监控端的视频监控界面, 以 3G 的方式进行登录, 并通过 Android 智能手机自带的浏览器播放视频数据。通过 Android 智能手机浏览器观看到的视频达到正常监控的标准。

关键词 嵌入式 Linux 视频监控 流媒体服务器 S3C2440 Android 智能手机

中图分类号 TP274

文献标志码 A

Abstract The existing video monitoring facilities are investigated and studied, and the design strategy of the embedded video monitoring based on Android platform is proposed. The system is composed of two parts, one is the S3C2440 development board, and the other is Android smart phone. The target board is used for video acquisition and encoding compression, the MPEG-4 video stream generated is sent to Helix streaming media server. Android smart phone is used as the video monitoring interface of the monitor that is log-in in 3G, and the video data are playing through built-in browser of Android smart phone. The video watched on the phone reaches normal monitoring standard.

Keywords Embedded Linux Video monitoring Streaming media server S3C2440 Android smart phone

0 引言

中国视频监控市场总体上经历了模拟监控、数字监控和网络监控三个主要阶段。作为第一代视频监控技术的模拟监控, 在图像质量、监控范围上都有很大的地域局限性。数字监控作为继模拟监控之后的第二代监控技术, 无论在图片质量、保存时间以及可靠性等性能上都有很大的提高, 为用户提供了高性价比的监控解决方案。随着 3G 网络的出现, 手机视频监控的应用领域得到拓展, 家庭监控和小型商铺监控的发展倍受看好。另外, 移动安防在公安、武警、人防、抗洪救灾、森林防火等公共部门的应用潜力非常巨大。本文提出一种以嵌入式 Linux 作为操作系统、S3C2440 作为中央处理硬件平台和 Android 智能 3G 手机作为监控客户端的视频监控系统方案^[1-3]。

1 系统整体架构

本系统选用韩国三星电子的 32 位处理器 S3C2440 作为系统控制的核心处理器。因为 USB 接口的摄像头成本低廉, 标准的接口安装灵活方便, 应用十分广泛,

所以本文采用了台湾嘉应的 USB 摄像头。整个系统硬件包括 USB 摄像头、嵌入式处理器 S3C2440、存储器模块和客户监控端。系统结构如图 1 所示。

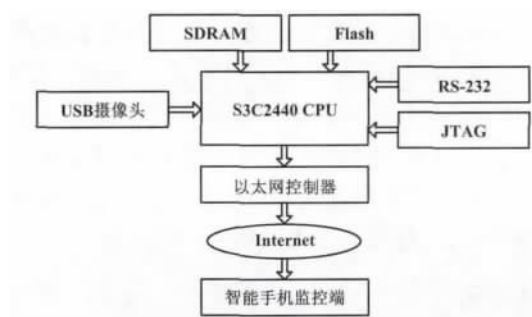


图 1 系统整体框图

Fig. 1 Overall structure of the system

1.1 系统硬件分析

S3C2440 是韩国三星公司开发的一款基于 ARM920T 内核和采用 CMOS 工艺的 32 位的 RISC 微处理器。其低成本、低功耗, 且全静态设计特别适合应用于对成本和功率比较敏感的场所。该处理器的主频达到 400 ~ 533 MHz, 接口多、资源丰富。存储单元包括 Flash 和 SDRAM。Flash 具有掉电保持数据的特性, 用于存储 Bootloader 启动程序、Linux 内核映像、文件系统以及用户应用程序等; SDRAM 数据存取速度快于 Flash 存储器, 用于为操作系统和应用程序提供运行空间。

修改稿收到日期: 2012-10-24。

第一作者郑海洲(1988-), 男, 现为上海理工大学检测技术与自动化装置专业在读硕士研究生; 主要从事嵌入式硬件方面的研究。

1.2 监控端分析

智能手机操作系统采用的是 Android 系统, Android 操作系统是基于 Linux^[4] 平台的开源手机操作系统。该平台由操作系统、中间件、用户界面和应用软件组成。Android 平台是一个先进的平台,而且 Android 操作系统是开源的项目,开发者可以利用现有资源进行自适应开发。Android 手机平台已集成了音视频编解码的功能,并且提供了相应的 API,因此可以根据系统的实际需要,利用 Android 开发工具编写基于网络的音视频播放器。Android 是基于 Linux 的自由及开放源代码的操作系统,用户可以对该系统进行裁剪和定制。这些特点为本系统的开发提供了十分便利的条件。

1.3 系统软件设计

系统软件是在 Linux 操作系统的基础上构建的,文件系统是针对 Flash 的无缓冲机制的日志式闪存文件系统 JFFS2 包括应用程序、模块、配置文件和库函数等。

视频采集系统程序包括选择合适的 Bootloader、编译嵌入式 Linux 内核、加载摄像头驱动、建立日志式闪存文件系统几个部分^[5]。

1.3.1 JFFS2 文件构建

JFFS2 文件构建步骤如下。

- ① 构建文件系统基本目录 `#mkdir bin dev home etc lib proc sbin sys var tmp root usr tools` ;
- ② 修改设备号 ;
- ③ 编写 MAPS 文件 ;
- ④ 配置内核 ;
- ⑤ 生成映像文件 `/mkfs.jffs2 -d jffs2 / -o jffs2.img`。

1.3.2 摄像头驱动的加载

本系统采用的是基于中芯微公司生产的 ZC301P 芯片的摄像头。该摄像头具有高质量的 VGA 感光器,传输接口为 USB1.1,最大分辨率为 640 × 480,支持动态与静态视频采集。

在 Linux 平台上加载 USB 摄像头驱动,首先把 USB 控制驱动模块静态编译到内核,使开发平台支持 USB 接口。当使用摄像头进行视频采集时,使用 `insmod` 命令动态加载器驱动模块,这样摄像头就可以工作了。

1.3.3 BootLoader 的移植

BootLoader 的启动过程可以分为单阶段和多阶段两种。通常多阶段的 BootLoader 能提供更为复杂的功能和更好的可移植性。从固态存储设备上启动的 BootLoader 大多采用多阶段启动。第一阶段使用汇编来实现,它完成一些依赖于 CPU 体系结构的初始化,

并调用第二阶段的代码。第二阶段则通常使用 C 语言来实现,这样可以实现更复杂的功能,而且代码会有更好的可读性和可移植性。

1.3.4 Linux 内核的移植

本系统采用的 Linux 内核为 Linux2.6 系列^[5],开发时无需做任何修改,只要根据硬件情况稍做改动,便可在 S3C2440 目标板上很好地运行。

Linux 内核移植的过程图如图 2 所示。



图 2 Linux 内核移植流程图

Fig. 2 Flowchart of Linux kernel transplantation

Linux 内核移植步骤如下。

- ① 下载 Linux-2.6.14.tar.gz,将压缩包移动到/usr/src 目录下,并解压缩。
- ② 编辑 Makefile,修改目标 CPU 体系结构和交叉编译工具的路径,修改 Makefile 中 ARCH 和 CROSS_COMPILE 对应内容。修改本机的交叉编译工具链的路径和前缀。
- ③ 修改 Nandflash 的分区,在 arch/arm/mach-s3c2440/devs.c 文件中加入分区信息。
- ④ 修改 fs/Kconfig 文件,使内核支持 devfs 以及在/sbin/init 运行之前能自动挂载/dev 为 devfs 文件系统。
- ⑤ 复制 make 的配置文件,对内核进行配置。其中 Video For Linux 配置必选。
- ⑥ 配置工作完成后,执行 `make zImage` 操作,把 arch/arm/boot 目录下的 zImage 下载到目标板上运行。

2 视频采集的应用程序

视频采集系统主要包括视频采集程序、视频编码程序和流媒体服务器 3 个应用层软件。

2.1 视频采集程序

Video4 Linux V4L 是 Linux 中关于视频设备的内核驱动,它为 Linux 的视频和音频提供了一整套的 API 接口,配合相应的视频采集设备和驱动程序,可以实现影像/图片采集等功能。

2.1.1 接口函数

drivers/media/video/videodev.c 文件提供了相关内核接口函数,具体介绍如下。

① `int video_register_device (struct video_device* , int type , int nr )`

该函数用于实现 Linux 视频设备在内核中的注册。其中,参数“`struct video_device*`”是想要注册的“`video_device`”结构,参数“`type`”是想要注册的设备类型,参数“`nr`”是想要注册的设备号。

② `struct video_device`

该函数用于定义 `video_device` 结构体。

③ `extern void video_unregister_device (struct video_device* ) ;`

该函数用于实现 Linux 视频设备在内核中的注销。其中,参数“`struct video_device*`”是想要注销的“`video_device`”结构。

2.1.2 视频采集的关键步骤

视频采集步骤包括打开视频、读取设备信息、读 `video_picture` 的信息、改变 `video_picture` 中分量的值、初始化 `channel`、关闭设备等,具体介绍如下。

① 打开视频。USB 摄像头在系统中对应的设备文件为 `/dev/video0`,采用系统调用函数 `open` (实现。

② 读取设备信息 采用 `int v4l_get_capability (v4l_device* vd)` 函数,可成功读取 `video_capability` 中的信息。

③ 读取 `video_picture` 中的信息 采用 `int v4l_get_picture (v4l_device* vd)` 函数,可成功读取图像的属性。

④ 改变 `video_picture` 中分量的值,先为分量赋新值,再调用 `VIDIOCSPICT`。

```
vd->picture.colour=65535 ;
If (ioctl (vd->fd,VIDIOCSPICT & vd->picture)<0)
{ perror ("VIDIOCSPICT" ) ;
Return -1 ; }
```

⑤ 初始化 `channel`。

视频采集有内存映射 `mmap` 和直接从设备读取 `read` 两种方式。`mmap` (系统的调用使得进程之间通过映射同一个普通文件实现共享内存。当普通文件被映射到进程地址空间后,进程可以像访问普通文件一样对文件进行访问,而不必调用 `read` (、`write` (等操作。其过程包括 6 个步骤,分别是 初始 `picture` 的属性 初始化 `video_mbuf`,以得到所映射的 `buffer` 的信息 修改 `video_mmap` 和帧状态的当前设置 将 `mmap` 与 `video_mbuf` 绑定 `mmap` 方式下的捕捉流程就是调用 `VIDIOCMAOTURE` 做视频截取 调用 `VIDIOCSYNC`,等待一帧截取结束。

⑥ 关闭设备。完成视频采集后,需要清理内存和

关闭视频设备。

2.2 视频编码程序

视频编码是数字图像处理中的一个重要研究领域,其目的就是在保证画面质量的前提下,用尽可能少的比特数表示视频信息。典型的视频编码技术有 JPEG、MPEG、H.261 和 H.264 等。

目前,基于 MPEG-4 标准的视频编码器比较多,主要有微软开发的 `msmpegv3`、`ffmpeg` 和 `Xvid`。`Xvid` 是应用较为广泛的 MPEG-4 视频压缩源代码,所以本文选择开源高效的 `Xvid` 视频编码器对采集的视频图像数据进行 MPEG-4 压缩编码。

`Xvid` 视频编码器选用 0.9.2 版本,在目标板的 Linux 操作系统中对 `Xvid0.9.2`^[6] 进行交叉编译的步骤如下。

① 解压缩 `Xvid` 源码 `tar-xzvf xvidcore-0.9.2.tar.gz`。

② 进入 `build/generic` 目录,创建编译配置文件。由于 `Xvid` 没有针对 ARM 的汇编优化,因此,编译时需将汇编关闭,还需指定程序所运行的平台为 `arm-linux`。

执行命令 :

```
./configure --disable-assembly --host=arm-linux
```

③ 编译 `make install`,安装程序。

④ 交叉编译完成后,会生成静态库 `libxvidcore.a` 和动态库 `libxvidcore.so.*`,用户程序可利用库文件所提供的编程接口,调用 `Xvid` 自带的函数。

摄像头采集的视频帧为 YUV420 格式,通过 `Xvid` 编码器的 `encoder_encode` 函数实现视频压缩。该函数定义如下 :

```
int encoder_encode (Encoder * pEnc , Xvid_ENC_FRAME * pFrame , Xvid_ENC_STATS * pResult ) ;
```

2.3 Real Server 流媒体服务器

目前,市场上主流的流媒体技术有 3 种,分别是 Real Networks 公司的 Real Media、Microsoft 的 Windows Media 和 Apple 公司的^[7-8] Quick Time。这 3 种技术都有自己的专利算法、专利文件格式和专利传输控制协议。本文以 `Helix Server` 为例,介绍如何在 Linux 下构建流媒体服务器。

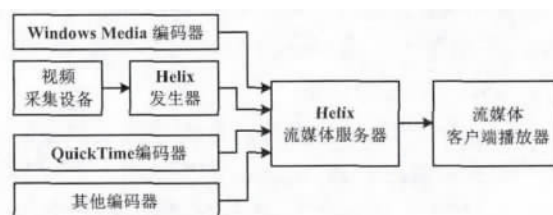


图 3 流媒体服务器架构

Fig. 3 Architecture of the streaming media server

2.3.1 Helix Server 的安装

Helix Server 的安装程序如下。

```
$ tar zxvf rs1113 - ga-linux-rhel4. tar. gz
//解压安装包文件
chmod +x helix-server-retail-11.01-rhel4-setup. bin
//改变下载的文件名称和组属性 ,使它成为可执行文件
./ helix-server-retail-11.01-rhel4-setup. bin
//运行安装程序
```

2.3.2 Helix Server 流媒体服务器的启动

使用命令在后台启动流媒体服务器 ,并在启动过程可以看到服务器版本和系统平台的信息如下。

① 启动服务器

```
Bin/rmsrver rmsrver. cfg &
//首先进入安装目录 ,使用命令在后台启动。
启动过程可以看到服务器版本和系统平台的信息 ,具体如下 :
/Bin/rmsrver rmsrver. cfg
Helix Server ( )1995-2005 RealNetworks , Inc. All
rights reserved.
Version Helix Server 11.0 (1.0.1.1 884 ) Build
74 277/5 688 )
Platform linux-rhel4-i686
.....
```

② 监控服务器的进程

```
#ps -efl grep rmsrver
//可以使用 ps 命令监控流媒体服务器的运行进程
root 3 765 3 655 0 15: 13 pts/3
00: 00: 00 ./Bin/rmsrver rmsrver. cfg
root 3 766 3 765 1 15: 13 ?
00: 00: 02 ./Bin/rmsrver rmsrver. cfg
root 3 767 3 765 0 15: 13 ?
00: 00: 00 ./Bin/rmsrver rmsrver. cfg
root 3 859 3 818 0 15: 15 pts/4
00: 00: 00 grep rmsrver
```

其中 ,PID 为 3 765 是主进程 ,PID 为 3 766 和 3 767 是子进程。

③ 监控端口

```
# netstat -ntulp | grep rmsrver
//可以使用命令监控流媒体服务器端口运行情况
tcp0 0: : : 9 090: : : * LISTEN 3 766/rmsrver
tcp0 0: : : 7 077: : : * LISTEN 3 766/rmsrver
.....
```

2.3.3 Helix 服务器的停止

由于 Helix 服务器没有提供服务器停止命令 ,所以只能通过使用传统的“Kill”来“停止服务器的进程” ,使用命令如下 :

```
#kill cat ./Logs/rmsrver. pid //停止使用 Helix 服务器
```

2.4 视频监控界面

本系统采用的是具有 3G 上网功能的 Android 智能手机 ,Android 手机有自带的浏览器。当在地址栏中输入开发板的 IP 地址时 ,手机就会以 3G 方式登入目标板的 Helix Server 流媒体服务器。

将编码器的码率分别设置为 256 kbit/s、512 kbit/s、768 kbit/s 图像格式设置为 CIF ,分别测试传统方法与本方案下的视频传输速度和视频质量 ,对比数据如表 1 所示。

表 1 对比数据
Tab.1 Comparative data

码率/ $\text{kb} \cdot \text{s}^{-1}$)	传输速率/ $\text{kb} \cdot \text{s}^{-1}$)	
	传统方法	本方案
256×10^3	$15 \sim 20 \times 10^4$	$20 \sim 30 \times 10^4$
514×10^3	$20 \sim 30 \times 10^4$	$40 \sim 60 \times 10^4$
750×10^3	$20 \sim 30 \times 10^4$	$40 \sim 60 \times 10^4$

由表 1 可知 ,本方案所提出的系统比传统方法具有更高的传输速率 ,实现了较为可靠的视频传输。

3 结束语

本文设计了一种采用 FFMPEG 与 Helix Server 流媒体服务器相结合的嵌入式网络视频监控系统。该系统以 S3C2440 处理器的开发板作为目标板和 Android 智能手机作为视频监控界面。在测试过程中 ,该系统实时性、稳定性、图象质量都比较理想。在不久的将来 ,视频监控的应用范围将扩展到交通、制造业、家庭等众多领域。

参考文献

[1] 杨丰盛. Android 应用开发揭秘[M]. 北京 机械工业出版社 2010.
[2] 梅玉平. 3G 的业务及管理[M]. 北京 人民邮电出版社 2007.
[3] 高宇 ,马福昌. 基于 S3C6410 及 Android 平台的视频监控系统的
设计[EB/OL]. [2011 - 12 - 30]. [http://www.paper.edu.cn/](http://www.paper.edu.cn/index.php/default/releasepaper/content/)
[index.php/default/releasepaper/content/](http://www.paper.edu.cn/index.php/default/releasepaper/content/).
[4] 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.
[5] 贺丹丹 张帆 ,刘峰. 嵌入式 Linux 系统开发教程[M]. 北京 清华大学出版社 2010.
[6] 牛玉冰 徐鑫. 浅谈流媒体服务器的安装与基本设置[J]. 中国
现代教育装备 2007 6(3) 7 ÷ 39.
[7] 何日辉. 嵌入式 Linux 视频监控系统的设计与实现[EB/OL].
[2010 - 04 - 19]. [http://www.paper.edu.cn/releasepaper/](http://www.paper.edu.cn/releasepaper/content/201004-660)
[content/201004-660](http://www.paper.edu.cn/releasepaper/content/201004-660).
[8] 曹江华. Linux 流媒体服务器安装配置图解[EB/OL]. [2007 - 07
- 25]. [http://www.ccw.com.cn/server/yyjq/htm2007/20070725_](http://www.ccw.com.cn/server/yyjq/htm2007/20070725_292311.shtml)
[292311.shtml](http://www.ccw.com.cn/server/yyjq/htm2007/20070725_292311.shtml).

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](#)

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](#)
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](#)

邀请注册码



关注论坛公众号

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](#)
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](#)

邀请注册码



关注论坛公众号

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](#)

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](#)

邀请注册码



关注论坛公众号

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](#)
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](#)

邀请注册码



关注论坛公众号

43. [基于 UEFI Shell 的 PreOS Application 的开发与研究](#)
44. [UEFI Bootkit 模型与分析](#)

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](#)

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](#)

邀请注册码



关注论坛公众号