

基于 S5PC100 移动视频监控终端的设计与实现

郝 昊, 许宝化, 唐 伦

(重庆邮电大学 通信与信息工程学院, 重庆 400065)

【摘要】为弥补传统固定监控架设成本高、监控有死角等不足,设计一种基于 S5PC100 处理器,以 WinCE6.0 为软件平台的移动视频监控终端。ZC0301 摄像头为采集设备,利用处理器内置编码器对原始视频进行 H.264 编码,通过支持 IEEE802.11b/g 协议的无线网卡或 WCDMA 模块将视频流发送至后台计算机,实现移动视频监控功能。实验表明该系统只占用有限的带宽,并能满足实际应用对高质量图像的需求。

【关键词】S5PC100 视频监控 无线网络 WinCE H.264

【中图分类号】TP277

【文献标志码】A

Design and Implementation of Mobile Video Monitoring Terminal Based on S5PC100 Embedded Processor

HAO Hao, XU Baohua, TANG Lun

School of Communication Engineering, Chongqing University of Posts and Telecommunications, Chongqing 400065, China

【Abstract】To overcome the drawback of the conventional installed video monitoring system, the design and implementation of a mobile video monitoring terminal is proposed based on S5PC100 embedded processor, in which WinCE6.0 acts as the software platform. The raw image sequences is captured by ZC0301 camera, and then encoded into H.264 baseline stream by the Multi-Format-Codec integrated in the processor, finally sent to background through a wireless network adaptor supporting IEEE802.11b/g protocol or WCDMA module. Application results show that this system only occupies a limited bandwidth and provides better monitoring quality.

【Key words】S5PC100 video monitoring wireless network WinCE H.264

移动视频监控是车联网中的一个重要组成部分。发展较为成熟的定点视频监控所提供的视频信息采集能力较为有限,在一些特殊场合难以发挥最大效用,并且成本消耗较大。而构建于车联网之上的移动视频监控可以很好地扫除固定监控的死角,弥补其缺陷,避免了繁杂的线路架设,安装和维护具有很大灵活性。

本文提出一种基于嵌入式处理器 S5PC100 的解决方案,充分发挥车联网的优势,结合 H.264 视频编码技术与无线网络传输技术,以 WinCE6.0 为软件系统平台^[1]搭建移动视频监控终端。该终端与后台监控中心结合可以组成数字化移动视频监控系统。

1 系统总体框架

本文实现一种嵌入式视频处理系统,以嵌入式处理器 S5PC100 为核心实现了对现场的实时监控,并通过无线网络把视频图像传输到后台计算机,以实现分析、存储和显示等功能。系统实现目标主要包括:原始视频采集、终端视频实时显示、H.264 视频压缩编码、无线网络发送。系统首先通过摄像头采集实时视频数据,将其显示在终端配备的 LCD 上,同时对其进行 H.264 压缩编码。最后通过 WiFi 无线网卡将压缩的视频数据传输到主机端,用户即可在后台主机通过媒体播放程序查看实时监控画面。

设计任务包括硬件设计与软件设计。设计指标有:
1 实时视频分辨率 VGA 640×480 或 CIF 320×240 可选;
2 帧频 20 帧/秒 f/s,最大可达到 30 f/s;
3 视频编码采用 H.264 标准,减小传输带宽;
4 终端最大可配备 4 Gbyte 的 SD 存储卡,保存本地视频监控记录;
5 采用支持 IEEE802.11b/g 协议的无线网卡或 WCDMA 模块进行无线视频传输。

1.1 系统硬件设计

视频监控终端选取 S5PC100 开发板作为硬件开发平台,该开发板采用核心板加底板结构,核心板搭载 Cortex-A8 内核 S5PC100 处理器。核心板与底板之间使用 4 个 2 mm 标准间距双排插针连接。底板上设有 4 路 UART 接口,4 路支持 USB1.1 的 USB HOST,1 路支持 USB 2.0 的 USB OTG 接口。显示输出方面,采用 FPC 接口连接显示屏,另外一个 50 针的 LCD 插针接口,方便连接其他型号 LCD。

设计以 S5PC100 为核心,协同各种外部设备实现视频的采集、编码以及网络发送等功能。外部设备包括 ZC0301 摄像头、4.3 in 1(in=2.54 cm) TFT 真彩液晶触摸屏以及无线网卡。摄像头与网卡通过 USB1.0 接口与开发板主体相连接。处理器芯片内部集成了 FIMV-MFC4.0 视频编码器,通过该模块进行原始视频流的 H.264 编码。另外

有视频处理引擎作为视频输出硬件 将当前采集到的画面输出到显示设备上 以减轻 CPU 的运行负荷。

硬件整体结构如图 1 所示。

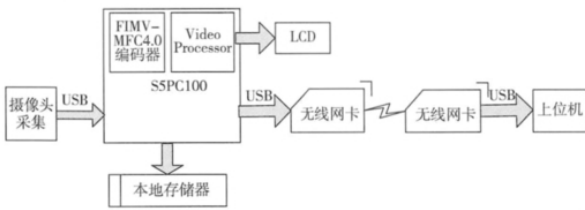


图 1 硬件整体结构

1.2 系统软件设计

本文软件平台选用 WinCE 嵌入式操作系统 在其基础上开发移动视频监控终端所需的应用软件。WinCE 是一个紧凑、高效的 32 位操作系统 适用于各种嵌入式系统和产品。系统层次结构清晰 利于开发人员程序编写及分工 专门面向资源有限的硬件系统 而且其模块化设计方式使得开发人员可根据具体应用场合定制系统功能。WinCE 系统非常适合诸如视频监控这种媒体处理能力要求高的应用场合^[2] 有利于各个功能模块的协同实现 同时不会耗费太多的系统资源。监控系统软件层次结构如图 2 所示。

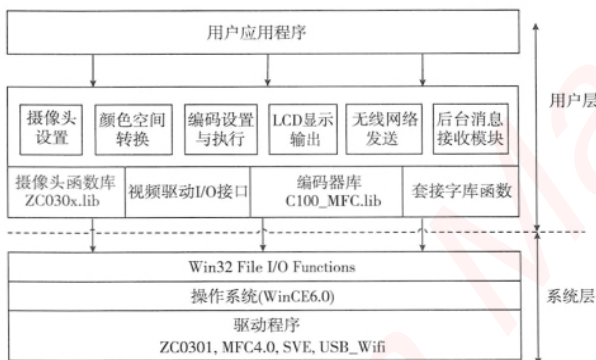


图 2 软件总体结构

整个嵌入式软件结构分为系统层和用户层两层。系统层包括系统软件本身、各硬件驱动程序以及 Win32 标准文件操作函数。在系统定制阶段完成功能定制与驱动程序添加 之后将系统烧写至 NAND Flash 操作系统即固化无法改变。

本文设计的移动视频监控终端用到了摄像头 ZC0301 驱动、编解码器驱动、视频引擎驱动以及 WiFi 网卡的 USB 驱动。驱动接口调用机制分两种 1: 应用程序直接调用设备驱动 I/O 函数 DeviceIoControl (没有封装好的 API 函数以供调用 视频显示驱动即以此方式调用 2; 应用程序调用设备 API。API 实际上是封装好的 I/O 函数, 便于应用程序调用。应用此机制的有编解码器和摄像头

ZC0301。开发人员利用 Windows 的 Socket 接口进行网络编程 不需要直接面对网卡驱动程序。

应用程序开发在可视化集成开发环境 VisualStudio2005 中完成 基于 MFC 编程用 C++ 语言实现。

2 关键技术实现

2.1 终端实时显示

WinCE 支持的标准绘图函数将绘图任务全部交由 CPU 执行 增加了 CPU 的运行负荷 降低了应用程序对于芯片内部专用硬件模块的依赖程度 限制了硬件性能的发挥。本设计开发的应用程序使用视频驱动 Video Driver 绘图。程序通过视频驱动程序 I/O 函数 直接控制处理器内部的视频处理器 又称视频引擎 SVE 来完成绘图操作。

2.1.1 原理介绍

视频处理器的主要功能是将内存中的图像数据以覆盖 Overlay 形式直接呈现在图形输出设备上 本设计使用 LCD。) 该功能的实现需要视频处理器与 Camera Interface 两部分协同实现。其中 CAMIF 模块作为视频处理器的前端预处理模块 图像经此模块转换为视频处理器所要求的格式。该部分结构如图 3 所示。

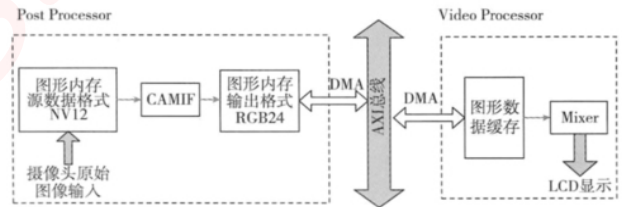


图 3 终端显示模块结构

应用程序编写相应的 API 对视频引擎进行控制 这些 API 函数将内存的物理地址作为参数进行传递。将图形内存的物理地址传递给视频驱动即可在显示设备上快速绘出预置的图像。

2.1.2 显示模块 API 编写

由于没有封装好的库函数 该部分程序需要直接使用驱动程序接口 即调用设备 I/O 函数 DeviceIoControl () 编写相关的 API 驱动视频引擎完成相关的显示操作。编写的 API 应执行以下 3 种操作 :

1 加载驱动程序。应用程序首先要获取驱动程序的句柄 调用 Win32 的标准库函数 CreateFile (来完成此操作 显示驱动程序名称为"VDEO"。

2 调用设备 I/O 函数完成显示操作。这一操作通过 DeviceIoControl (函数实现 操作分为请求/释放、功能执行、电源管理 具体操作类型由 IOCTL 代码决定。本文只

用前两种。具体步骤为：

- 1) 请求资源,代码 IOCTL_SVE_RSC_REQUEST_FIMD_WIN0 与 IOCTL_SVE_RSC_REQUEST_POST ；
- 2) 设置显示控制器为窗口 0 模式,代码 IOCTL_SVE_FIMD_SET_WINDOW_MODE ；
- 3) 调整前置处理单元的参数,代码 IOCTL_SVE_POST_SET_PROCESSING_PARAM ；
- 4) 传递源/目标图像地址参数,代码 IOCTL_SVE_POST_SET_SOURCE_BUFFER ；
- 5) 显示进程开始,代码 IOCTL_SVE_LOCALPATH_SET_WIN0_START。

3 关闭驱动 释放资源。该操作同样通过 DeviceIoControl (函数实现。对应 IOCTL 代码 IOCTL_SVE_RSC_RELEASE_FIMD_WIN0。执行完后调用 CloseHandle (函数) 数释放句柄。

2.1.3 主要函数

该部分程序主要包含 3 个 API 函数,具体说明如表 1 所示。

表 1 终端实时显示 API

函数名称	函数功能
GetDisplayPhyBuf (创建显示缓存 返回其物理地址
DisplayOpen ()	打开驱动并请求相应资源
DisplayStart ()	调用显示硬件驱动显示图像缓存中的内容

2.2 图像色彩空间转换

本文设计系统所选用的 ZC0301 摄像头输出格式为 YV12 格式,与编码器要求的 NV12 都属于 YUV420 格式,但存在色差信号排列方式的不同。两种色彩格式在内存表面的排列如图 4 所示。需要编写程序转换摄像头输出的原始图像颜色格式,成为编码器要求的交错类型 NV12 格式,否则编码器输出的视频帧颜色将出现混乱,大大影响监控效果。另外,显示引擎的前置处理器输入图像格式也要求为 NV12。

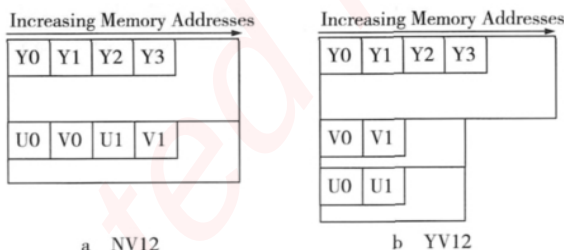


图 4 YV12 格式与 NV12 格式

程序利用指针重新排列内存中原始图像帧中的 U 和 V 分量,函数返回重排后图像缓存首地址。该地址作为参数用于接下来的编码函数输入。这部分程序仅当摄像头

输出 YUV420 无交错格式时有效,若改变输出格式或使用其他类型采集设备,需要重新考虑颜色空间问题。转换函数描述见表 2。

表 2 色彩空间转换函数

函数名称	函数功能	输入参数	返回值
ReconstructImgBuffer (重新排列图像帧缓存中的 U 和 V 分量,得到新的 NV12 格式	lpFrameBuffer 指针,原始图像帧缓存首地址	NewBuffer 指针,重新排列后得到的缓存首地址

色彩空间流程图如图 5 所示。

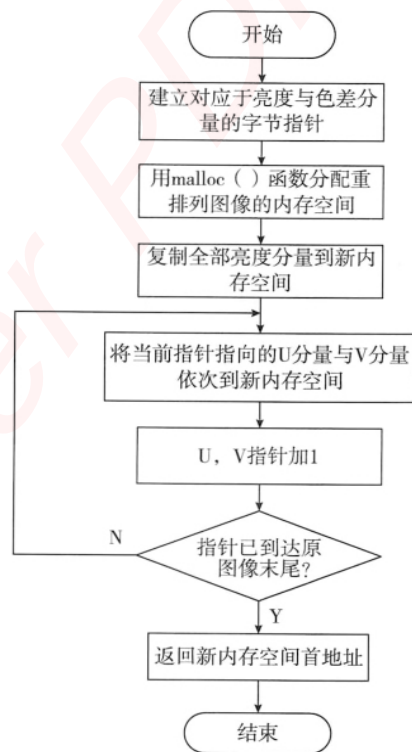


图 5 颜色空间转换流程

2.3 H.264 视频编码

S5PC100 拥有适合于多媒体处理应用的特性,其内部集成有一个多格式编解码器 MFC,支持多种标准的视频编解码^[3]。利用 S5PC100 的内置编码器 API 可以编写相应的多媒体处理程序。MFC 是独立于 CPU 的硬件编解码器,相对于传统的软件编解码,使用 MFC 可以大幅降低 CPU 占用率,这对于硬件资源有限的嵌入式系统具有十分重要的意义。

本设计选择 H.264 标准对原始视频数据进行编码。H.264 是目前监控领域中最先进的一种压缩标准,又称 AVC,是技术含量最高视频编码格式之一,可以在尽可能低的空间占用下获得较好的图像质量^[4]。与 MPEG-4 压缩方式相比,H.264 压缩比可提高近 30%,大大节约了

存储和网络传输带宽,同时也很好地抑制了噪声导致的图像失真,使图像质量更加清晰。

编码分 2 个步骤执行:

1 编码器初始化。建立编码操作实例,分配相应内存,并设置编码器相关参数^[5]。参数存放在 h264_param 结构体中。操作中可以根据实际需要对这些参数进行调整:

```
* hOpen = SsbSipMfcEncOpen H264_ENC //创建编码操作实例
virInbuf = SsbSipMfcEncGetInBuf *( hOpen, &phyInbuf, (width *
height * 3) > > 1 //获得编码器输入内存地址
重要编码参数设置:
h264_param -> profile = 1 //选择 H264 画质类型 0:为 baseline,1
为 main,2 为 high
h264_param -> level = 12 //H.264 的 Level 设置
h264_param -> gopNum = 150 // I-frame 之间的间隔
h264_param -> width = 320 //画面宽度
h264_param -> height = 240 //画面高度
```

2 执行编码操作。对原始输入图像帧进行 H.264 编码。执行该操作时,将 NV12 图像数据拷贝到分配好的编码器输入内存,执行相关的 API 进行编码操作。返回编码数据的内存地址,同时编码数据的相关信息存放在 output_info 结构体中。生成的 H.264 码流写入本地文件,作为车辆行驶过程中的记录保存,代码为:

```
Memcpy (virInbuf, (void*) lpFrameBuffer, (width * height * 3) > >
1) //将原始图像拷贝到编码器输入内存
ret = SsbSipMfcEncExe hOpen, output_info //执行一帧编码操作,
并保存相关信息到 output_info 结构体中
virOutbuf = SsbSipMfcEncGetOutBuf hOpen, &phyOutbuf //获得
编码器输出内存地址
fwrite (fwrite lpOutput_info, encodedSize, fp_OutStrm) //将编码后的数
据写入文件
```

2.4 无线网络发送

系统选用支持 IEEE802.11b/g 协议的无线网卡作为无线传输设备^[6]。另外,系统内部已经预置了 WCDMA 模块驱动程序,也可以通过 USB 接口连接 3G 网络模块,建立连接后通过 3G 网络向后台发送视频数据。监控终端通过访问点或点对点方式接入到组建好的无线网络中,向后台计算机发送经过编码的视频数据。网络发送程序模块直接调用 WinCE 的套接字接口^[7],通过 UDP 数据报的形式逐帧向指定的 IP 地址发送视频流。由于嵌入式硬件平台资源有限,因此在终端上采用简单发送 UDP 数据报的方式,其后的播放等工作交由后台计算机处理。

在 WinCE 中使用无线网络时首先要配置网卡的相关属性。测试网络发送时,在网卡属性中将网络设为点对点方式,并配置好本地 IP 地址。接收端的计算机也要做相

同的设置,之后就可以通过编写好的程序进行视频发送。

该部分网络通信程序通过调用 Sockets 的 API 函数来编写。使用套接字时要在代码中包含头文件 winsock2.h,并通过预处理命令 pragma 将库 ws2.lib 链接进程序中。

程序主要流程为:

- 1 建立一个异步套接字实例 MySock。
- 2 调用 WSASStartup 函数加载套接字库,并指定要加载的 WinSock 库版本。
- 3 指定本地套接字所使用的端口。
- 4 调用 Create 的方法创建 UDP 套接字。
- 5 向绑定好的 IP 地址发送编码后的视频帧。调用 SendTo 方法时,将编码输出信息 output_info 结构体成员 encodedSize 作为参数,确定要发送的数据报长度。

3 实际测试结果

编码输出数据如下:第一帧包括参数信息和编码数据,参数信息大小 HeaderSize 为 237 byte,实际编码大小 EncodedSize 为 1830 byte,其余帧只包含编码数据,每一帧的 EncodedSize 约为 2000 byte。

实际占用带宽测试 CIF 320×240,30 f/s 规格时约 50 kbit/s。

为了在后台计算机上实时查看车载终端传送回的画面,选用 VLC 播放器^[8]播放接收到的 H.264 视频流。VLC 是一款免费的开源媒体播放器,其优势在于可以直接播放原始的 H.264 码流文件,而不需要对码流进行视频容器的封装。同时由于其开源的特性,在实际应用中可以根据自己的需要对播放器进行具体的定制,以适应相应的场合。

实际测试效果如图 6 所示。

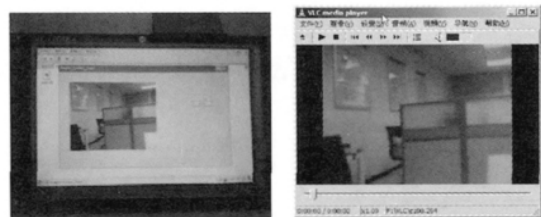


图 6 终端与 PC 接收端测试结果

具体实现时,以 S5PC100 核心板为基础,利用其丰富的功能扩展空间,可以根据实际需求灵活定制扩展板结构,有效控制成本避免浪费。开放的系统平台也易于其他开源应用程序的移植。同时,选用的 TFT 触摸屏使得用户操作界面简洁、高效,适用于车辆内部的狭小空间。模块化的硬件结构也使整个系统便于维护和管理。

件反应速度和问题解决效率,使发挥人的作用最大化,为要求较高的广电类建筑建立更安全的维护管理系统。因此有理由相信,无线技术在楼宇自控系统未来的发展进程中,必然会发挥其独特作用。无线楼控技术应用还很鲜见,本文论述方案深度亦有不足,难免疏漏,旨在抛砖引玉,希望各位学者予以指正。

参考文献 :

[1] 张浩军,祝跃飞,陈莉,等. WAPI 中 ASU 的研究与设计[J]. 计算机

应用研究 2005 6()199-201.

[2] 张金平,马旭东,王晓明,等. 智能楼宇系统中的软件化网络视频监控服务器[J]. 微计算机信息 2004 1()144-145.

[3] 王宏玉. 基于 Niagara 平台的智能建筑系统集成技术研究[D]. 长沙 湖南大学 2007.

[4] 童世华. 基于以太网的智能楼宇梯口机的设计与实现[J]. 电视技术 2012 36 15 50-53.

参考文献 :

[1] 张浩军,祝跃飞,陈莉,等. WAPI 中 ASU 的研究与设计[J]. 计算机

责任编辑 时 雯

收稿日期 2012-08-28

上接第 137 页)

益发展的今天,开放性和可编程性无线充电技术会发展的更好,相信有一天感应耦合电能传输会成为人们生活的好帮手。

参考文献 :

[1] 黄洁琳,章磊. 无线充电的设计[J]. 山西电子技术 2009 3()80. :

[2] 肖志坚,韩震宇,李绍卓. 关于便携式电子设备新型无线充电系统的研究[J]. 自动化技术与应用 2007 1()14-116.

[3] 池雪莲. 交直流自动切换无线传能充电器的设计[J]. 山东师范大

学学报 自然科学版 2010 25 5()35-37.

[4] 倪兰. 无线充电技术国际标准发布 市场风险致主流厂商态度未明[J]. 通信世界 2010 3()8. :

[5] 杨小牛,楼才义,徐建良. 无线充电原理与应用[M]. 北京 电子工业出版社 2011.

参考文献 :

[1] 黄洁琳,章磊. 无线充电的设计[J]. 山西电子技术 2009 3()80. :

[2] 肖志坚,韩震宇,李绍卓. 关于便携式电子设备新型无线充电系统的研究[J]. 自动化技术与应用 2007 1()14-116.

[3] 池雪莲. 交直流自动切换无线传能充电器的设计[J]. 山东师范大

作者简介 :

谢智娟 1970— 女,副教授,主研物理教学教育。

责任编辑 薛 京

收稿日期 2012-09-11

上接第 144 页)

4 总结

本文设计了一种基于嵌入式平台的移动视频监控终端,充分发挥了 S5PC100 在数字视频处理方面的优势,整个系统具有体积小、实时性强、性价比高、稳定性好等特点。作为新型的数字化视频监控设备,可以在各个领域中得到广泛应用。相比于传统的模拟监控,数字视频处理技术提高了图像的质量与监控效率,模块化结构的系统体积小、易于安装、使用和维护。嵌入式的数字化远程监控是基于现代通信技术的一种新应用,符合当今信息社会中数字化、网络化和智能化的发展趋势,拥有广阔的前景和优势。

视技术 2011 35 1()3-85.

[3] 游向东,刘福良,刘军,等. S3C6410 中 MFC 的研究与应用[J]. 电子产品世界 2011 1()36-40.

[4] 叶锐. 一种基于 PC 平台的 H. 264 快速视频采集系统[J]. 电视技术, 2011 35 9()25-127.

[5] 刘欢,夏良,史国川. 一种基于参数替换的 H. 264 标准水印嵌入算法研究[J]. 数字技术与应用 2012 2()125-126.

[6] 王艺,邓佳佳,林毅. 适用于车联网应用的移动通信网络[J]. 电信科学 2012 28 6()1-6.

[7] 杨桂霞. C++ builder 中基于 SOCKET 的网络编程[J]. 计算机光盘软件与应用 2012 6()169-170.

[8] 郑秀峰. 用 VLC 播放 DM500S 接收的卫视节目[J]. 卫星电视与宽带多媒体 2012 5()74. :

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](#)
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](#)

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](#)

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](#)

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 I/O 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](#)

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](#)

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](#)

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](#)

Created in Master PDF Editor