

Linux 和 Cortex - A8 的视频处理及数字微波传输系统设计[※]

苏宁,赵利

(桂林电子科技大学 信息与通信学院,桂林 541004)

摘要: 以基于 Linux 操作系统的 Cortex - A8 内核的 S5PV210 嵌入式平台为核心,完成音视频采集与压缩/解压处理的设计开发,通过 CP - FSK 高速调制/解调器完成数字音视频微波无线传输。采用 DDS 和 PLL 完成了 CP - FSK 调制解调器电路的设计,结合原有的微波射频电路实现了音视频的微波无线收发。实验结果表明,系统设计合理、运行稳定流畅,满足设计要求。

关键词: Linux; Cortex - A8; H. 264; USB 转 UART; 微波传输

中图分类号: TP37 **文献标识码:** A

Video Processing and Digital Microwave Transmission System Based on Linux and Cortex-A8[※]

Su Ning, Zhao Li

(School of Information and Communication, Guilin University of Electronic Technology, Guilin 541004, China)

Abstract: In order to achieve the development of audio&video acquisition and compression/decompression process and the microwave wireless transmission of digital audio&video by CP-FSK high speed modulator/demodulator, a system is designed based on Linux operating system, in which the Cortex-A8 Core S5PV210 embedded platform acts as the core. The CP-FSK modem circuit is finished by DDS and PLL. Combining with the original microwave radio frequency circuit, microwave wireless transmit-recvie of audio&video is achieved. The experimental results show that the system is reasonable, it works stably and smoothly and meets the design requirements.

Key words: Linux; Cortex-A8; H. 264; USB to UART; microwave transmission

引 言

数字视频通信系统较传统模拟视频通信系统而言,具有抗干扰、抗噪声能力强、便于计算机处理等优点。随着数字技术的发展,特别是高效的音视频压缩技术的发展,数字视频通信系统需要较大带宽和设备复杂的问题得到了很好的解决。由于计算机与大规模集成电路的快速发展,高性能的嵌入式系统不断涌现,从成本和技术上为原有模拟视频传输系统的改造提供了基础。针对采用模拟技术体制的某微波视频传输系统,存在占用带宽宽、邻道干扰严重、图像质量差等问题,需对该系统进行数字化改造。

在保留原系统射频模块和主要功能不变的基础上,以基于 Linux 的 Cortex - A8 嵌入式平台为核心,完成音视频数字化采集、压缩处理以及收发处理功能,实现了高帧率、高分辨率的音视频采集、压缩/解压及微波传输的数字视频无线通信系统。采用 mmap 视频采集、多线程、CP - FSK 调制解调和 USB 转 UART 等关键技术,系统具有硬件电路简单、成本低、易开发、高速传输、实时性好等优点,

具有重要的工程设计参考价值。

1 系统整体方案

1.1 系统参数与技术指标

① RF 通道:发送端中频频率为 60 MHz,接收端中频频率为 38 MHz,中频带宽为 8 MHz。

② 数字调制:调制方式为 CP - FSK,中心频率为中频 60 或 38 MHz,频偏为±2 MHz,调制速率为 4 Mbps。

③ 音视频:音频编码格式为 LPCM,视频编码格式为 H. 264,视频分辨率为 640×480,视频帧率为 25 fps。

1.2 发送端技术方案

在以 Linux 为操作系统的 Cortex - A8 嵌入式平台上,采用 USB 摄像头和麦克风分别实现视频和音频的数字化采集,同时进行本地视频回放。对采集所得的视频数据进行 H. 264 压缩处理,并与音频进行音视频组帧发送。经组帧的音视频数据通过 USB 转 UART 接口传送至 CP - FSK 调制模块。USB 转 UART 接口采用能实现高速异步通信的 XR21V1410 芯片实现,CP - FSK 调制模块采用

体积小、低功耗和控制灵活的 AD9959 DDS 芯片实现。微波系统的中频和射频电路保持不变,实现一个传输速率达到 4 Mbps 的数字音视频微波发送端,发送端的方案框图如图 1 所示。

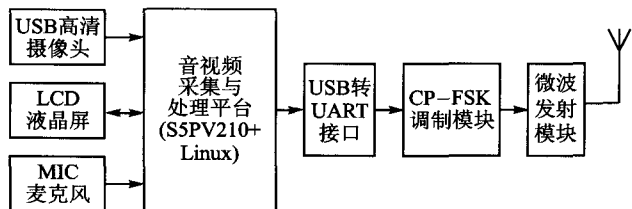


图 1 发送端方案框图

三星公司的 S5PV210 处理器具有高性能的图像处理器 SGX540,支持 USB2.0 接口,有丰富的接口、完善的 BSP、齐全的开发文档,最高可支持 1080p@30 fps 硬件解码视频流畅播放,格式可为 MPEG4、H. 263、H. 264 等^[1],因此高清视频的采集与压缩可以采用 S5PV210 处理器。

异步串行通信相比于同步串行通信,具有实现简单、使用方便和成本低等优点。单纯地使用开发平台上的 UART 接口无法满足系统对传输速率的设计要求,可以通过 USB 转 UART 方式来提高串口传输速率。采用单片 XR21V1410 实现 USB 转 UART,传输速率最高可达 4 Mbps,且该芯片体积小、硬件电路设计简单。

为了抑制带外辐射、避免邻道干扰,同时易于实现,选用频带利用率高、带外衰减快的 CP-FSK 调制方式。为了满足高达 4 Mbps 的高速 CP-FSK 的技术要求,采用 DDS 技术来实现 CP-FSK 全数字调制。ADI 公司的 AD9959 DDS 芯片具有功耗低、体积小、控制灵活、成本低等优点,可以执行高达 16 阶的频率、相位或幅度调制 (FSK、PSK、ASK),很容易实现信号全数字调制,成为系统实现 CP-FSK 全数字调制的一个选择。

1.3 接收端技术方案

原有系统中,中频及射频电路保持不变。采用由 SAW 和补偿放大器组成的邻道滤波器抑制邻道干扰;采用中频 CP-FSK 解调方式,中心频率为 38 MHz,频偏±2 MHz,实现速度为 4 Mbps 的 FSK 解调;解调电路采用集成 PPL 的单片 NE564 芯片实现;异步串行通信接收采用基于 XR21V1410 芯片的 USB 转 UART 接口实现;音视频的接收和解码采用以 S5PV210 芯片为主处理器的 Linux 嵌入式平台来实现,并完成音视频的本地回放。接收端的框图如图 2 所示。

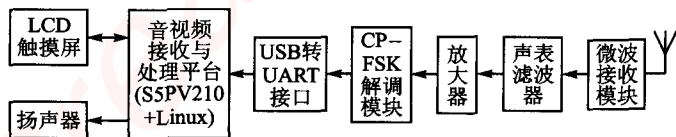


图 2 接收端框图

2 系统硬件设计要点

2.1 高速异步串口模块设计

XR21V1410 有自己的内部时钟,可精确产生任何波特率,提供大容量 FIFO,可实现速率高达 12 Mbps 的异步串行数据通信^[1]。图 3 为芯片内部结构框图。

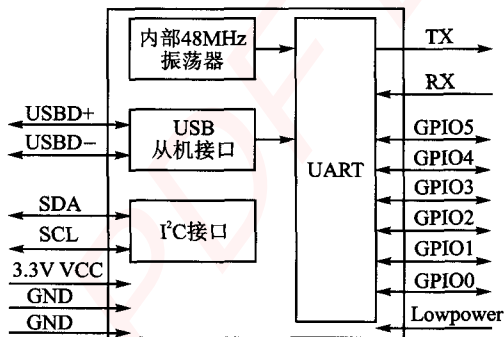


图 3 XR21V1410 内部结构框图

电路设计需注意:

- ① 芯片的工作电压为 2.97~3.63 V,所以引脚 VCC 接 3.3 V 电源,如果没有合适的电源,需多加一个电源电路;
- ② 引脚 USB+ 和 USB- 与开发板的 USB 接口连接;
- ③ RX 和 TX 为 UART 的异步串行收发引脚;
- ④ 可以加一个 ESD 保护电路,保护 USB+ 和 USB- 这两条高速数据免受静电放电和其他瞬变的影响;
- ⑤ Lowpower 引脚接一个外部上拉电阻,两个 GND 引脚接地,同时 SDA、SCL、GPIO0~GPIO5 引脚悬空。

2.2 CP-FSK 调制电路设计

CP-FSK 调制电路选用 ADI 公司的 AD9959 DDS 芯片,该芯片是一款 4 通道、内部时钟频率高达 500 MHz 的直接数字频率合成器芯片。每个通道均可提供独立的频率、相位和幅度控制,这种灵活性可用于校正信号之间由滤波、放大等模拟处理或 PCB 布局失配而引起的不平衡问题^[2-3]。CP-FSK 调制器采用微控制器与 AD9959 构成的硬件电路实现,其中 Atmel 公司的微控制器 ATmega32 用于 DDS 的配置,通过合理的设计,实现以 60 MHz 为中心、±2 MHz 的两个副载波频率之间的高速切换,并保持码元间相位的连续。此外,DDS 的输出是高速 D/A 转换后的信号,含有较多高次谐波分量,需外接低通滤波器 (LPF) 剔除信号中的杂波和噪声。

CP-FSK 调制器的硬件结构框图如图 4 所示。

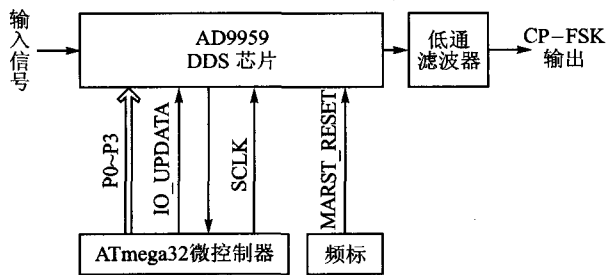


图4 CP-FSK调制器的硬件结构图

其中,P0~P3用于控制调制方式的选择,IO_UPDATA为寄存器刷新线,SCLK为串口同步信号,MARST_RESET为复位信号。

2.3 CP-FSK解调电路设计

CP-FSK信号的解调方式有很多种,内部有电压比较器、有与TTL电平相匹配的输入/输出端、集成了锁相环的NE564芯片是一种很好的选择方案。芯片NE564由输入限幅器、鉴相器、压控振荡器、放大器、直流恢复电路和施密特触发器6部分组成^[4]。图5为CP-FSK解调电路。从射频电路接收到的FSK信号经声表滤波器(SAW)和补偿放大电路(ERA)处理后,接入NE564芯片的6引脚进行解调处理。

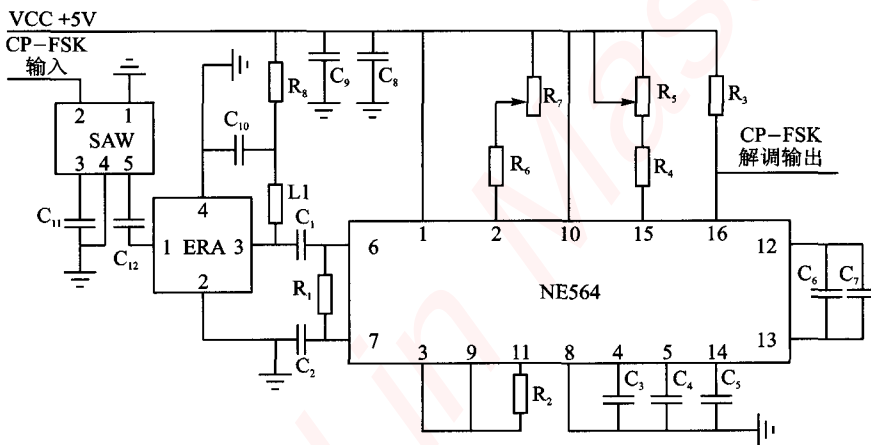


图5 CP-FSK解调电路

中心频率为38 MHz的CP-FSK解调电路的电路参数取值,关键要确定4、5引脚和12、13引脚处的电容值,其他外围器件大小的变化不大,影响比较小。计算方法如下:压控振荡器的固有频率与接在12、13端的定时电容 $C_T = C_6 + C_7$ 有关:

$$C_T = \frac{1}{2 \cdot 200 f_0}$$

中心频率 f_0 为38 MHz,可得 $C_T = 11.96$ pF,则可以选择6.8 pF和20 pF的可调电容来调节VCO的中心频率。通过可变电容调节VCO的频率到所需要的中心频率。

对于影响环路滤波器的接在4、5引脚上的 C_3 、 C_4 电容值的选择,可以根据芯片资料推导出计算方法如下:当规定环路的增益 $k=0.5$,芯片供电电压为5 V,且调节2引脚电位器使其对应电流 I_2 为200 μ A后,可计算电容来调节中心频率。此时,4、5引脚的电容大小,可用下式计算得到:

$$C_3 = C_4 = \frac{1}{2.25 \times 10^{10} \times f_0 k^2}$$

其计算值为4.67 pF。

3 系统软件设计

3.1 系统软件设计框架

软件系统包括:应用程序、库、操作系统和驱动程序,其层次关系略——编者注。系统在应用层可以实现视频图像的采集、压缩/解压和显示,声音采集和播放,视频格式转换和串口发送接收等应用。使用到的库有进行H.264视频编解码的编解码库等,系统采用的内核版本为Linux-3.0.8,驱动程序有USB摄像头驱动、音频驱动、7寸LCD彩色触屏驱动和USB转UART接口驱动等。

3.2 关键软件模块设计

3.2.1 视频采集模块设计

V4L2是Linux中关于视频设备的内核驱动。在Linux系统中,视频设备以设备文件的形式存在于/dev目录下,视频的采集可以通过对视频设备文件进行读写等操作来实现。根据V4L2接口编写视频图像采集程序,主要包括打开设备文件、取得设备功能信息、设置图像格式、向驱动申请帧缓冲、申请物理内存、采集图像数据、停止采集和关闭设备等^[5]。流程图略——编者注,具体操作如下:

① 打开视频设备文件,获取摄像头基本信息,并设置采集视频的格式等参数。

② 向驱动申请帧缓冲,并将申请到的帧缓冲映射到用户空间。

③ 将申请到的帧缓冲全部放入队列,并启动视频采集。

④ 开始视频采集,把采集到的视频数据帧缓冲从buf中出队列,获得视频数据,处理完毕后,将空的缓冲再次放入队列,循环往复采集连续的视频数据。

⑤ 停止视频的采集和关闭视频设备。

3.2.2 视频硬件编解码

S5PV210处理器采用了强大的信号处理扩展集,为H.264等媒体编解码提供加速。根据S5PV210处理器的用户手册,H.264编码的具体步骤、流程图略——编者注。

解码部分的步骤与编码大致相同,具体步骤、流程图略——编者注。

3.2.3 音频采集

脉冲编码调制 PCM 是对语音信号进行采样,并对每个采样值进行量化编码。在 Linux-3.0.8 中,已经存在音频驱动,可以很方便地对音频文件进行读写操作,从而进行音频应用程序的开发。使用系统的设备文件接口为“/dev/dsp”,声音采集的流程步骤大致如下:①打开设备文件;②配置参数,包括采样频率、量化位数和声道数目等;③采集声音;④停止采集并关闭设备。流程图如图 6 所示。

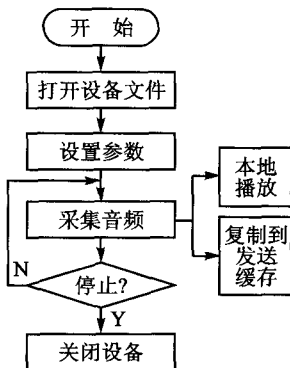


图 6 声音采集流程图

具体编程时,用 ioctl() 函数进行参数调用和配置生效,采集函数使用 read(), 播放函数使用 write()。

3.2.4 音视频数据发送与接收处理

(1) 数据的组帧与分段

音视频组帧方式采用声音+视频的方式,每个音视频帧的前端放声音,后面加入视频数据。为了便于进行音视频数据的串口收发和接收后对音视频数据的后续操作,传输前需对音视频帧数据分段处理。采用每次串口发送 512 字节的数据长度进行分段。每次串口发送的数据帧中有 3 个字节的数据长度进行分段。每次串口发送的数据帧中有 3 个字节的数据长度表示数据帧实际发送的数据长度(最多可为 504 字节),数据帧合计共 512 个字节。具体的示意图如图 7 所示。

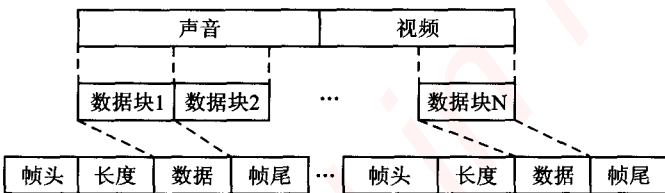


图 7 组帧与分段示意图

另外,在对音视频帧数据进行分段时需要考虑两种情况:一是数据能整除划分;二是数据不能整除划分。整除划分的可以直接发送,不能整除的音视频帧需要在音视频帧的最后一段数据块 N 补够相对应的数据,凑够 504 个字节数据后方可传输。

(2) UART 接口的收发

串口在嵌入式系统中有广泛的应用。串口属于字符设备,同其他设备一样,在 Linux 系统中以设备文件的形式存在,可以利用文件操作调用,包括常用的 open()、close()、read()、write() 等,比如以读写方式打开串口(int fd=open("/dev/ttyUSB0", O_RDWR);)。

最基本的设置串口包括波特率、校验位和停止位设置,主要是设置 structtermios 结构体的各成员值。read()、write() 用来实现串口的收发数据功能。实现串口收、发的基本流程图略——编者注。

3.3 USB 转 UART 驱动移植

基于 XR21V1410 芯片的 USB 转 UART 接口驱动属于 Linux 的字符设备。Linux-3.0.8 内核版本无 XR21V1410 芯片的驱动,需要从芯片厂商提供的源码进行驱动移植。驱动移植和制作新镜像可以进行如下操作:

- ① 把下载所得的适合该版本内核的驱动放到 /Linux-3.0.8/driver/usb/serial/下;
- ② 修改 /Linux-3.0.8/driver/usb/serial/下的 Kconfig 和 Makefile 文件;
- ③ 切换到 Linux-3.0.8 主目录下,在终端执行命令 cp mini210_linux_defconfig .config;
- ④ Linux-3.0.8 主目录下,在终端执行命令 make menuconfig,在图形界面中找到需要选中的名称进行配置;
- ⑤ 在终端执行命令 make zImage;
- ⑥ 编译完成后,进入到 /Linux-3.0.8/arch/arm/boot/下即可看到新生成的 zImage 文件,这就是刚编译好的镜像。

结 语

该数字微波视频传输系统选用 Linux 操作系统以及 Cortex-A8 内核的处理器 S5PV210,利用其内部集成的 H.264 硬件编/解码器,极大地提高视频编码效率。在分辨率为 640×480、帧率为 25 fps 的情况下,系统联调测试结果表明,音视频的无线微波传输可靠、图像清晰流畅、实时性良好,达到了系统设计的要求。

编者注:本文为期刊缩略版,全文见本刊网站 www.mesnet.com.cn。

参 考 文 献

- [1] EXAR company. XR21V1410 1-CH FULL-SPEED USB UART datasheet [EB/OL]. [2015-02]. <http://www.exar.com/connectivity/uart-and-bridging-solutions/usb-urats/xr21v1410/>.
- [2] 陈嘉佳,潘志浩,王天麟.一种基于 DDS 芯片 AD9959 的高精度信号发生器[J].自动化仪表,2007(4):24-28.
- [3] 孙文友,宋祖勋,段鹏.基于 DDS 技术的 2CPFSK 调制器的实现[J].通信技术,2006(S1).
- [4] 卢盛杰,宋刚.单片集成锁相环 NE564 在通信中的应用[J].黑龙江通信报,1995(3).
- [5] 王建新,张凯,王秀.基于 S3C2440 的视频采集及传输系统设计[J].电子器件,2013,36(3):367-370.

苏宁(研究生),研究方向为嵌入式系统开发;赵利(教授),研究方向为移动通信及网络。

(责任编辑:薛士然 收稿日期:2015-02-03)

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](#)

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](#)

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 定制嵌入式 Linux 发行版](#)

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](#)

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](#)

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](#)

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](#)

Programming:

1. [计算机软件基础数据结构 - 算法](#)