

基于 ARM 的图像采集与传输系统设计

杨成禹 庄晓奇

(长春理工大学 光电工程学院 长春 130022)

摘要: 针对功能设备的小型化、智能化趋势,提出了一种基于 ARM 微处理器 S3C2440 和嵌入式 Linux 操作系统的图像采集和传输的系统设计方案。本文介绍了系统软硬件平台的搭建,并且重点阐述了图像采集和传输的软件设计。该系统本质上是一个提供了监控功能的服务器。它实现了视频图像在 LCD 上的实时显示,此外用户还可以在 PC 机上通过网络查看所采集到的视频图像。系统测试表明:该系统性能稳定,成本低廉,功耗低,便携性好,可应用于工业控制、汽车电子等领域。

关键词: ARM; 嵌入式系统; V4L2; 图像采集; 网络传输

中图分类号: TP316 **文献标识码:** A **国家标准学科分类代码:** 520.4020

Design of image acquisition and transmission system based on ARM

Yang Chengyu Zhuang Xiaoqi

(Institute of Photoelectric Engineering, Changchun University of Science and Technology, Changchun 130022, China)

Abstract: For the trend of small and intelligent of the device, this article proposes design of image capture and transmission system based on ARM S3C2440 microprocessor and embedded Linux operating system. This article describes how to build the hardware and software platform of the system, and focuses on the software design of image capture and transfer. The system is essentially a server that provides a monitoring function. It enables real-time video images displayed on the LCD, in addition, the users can view the captured video images over the network on the PC. System tests show that the system is stable, low cost, low power and portable, it can be applied to industrial control, automotive electronics and other fields.

Keywords: ARM; embedded system; V4L2; image acquisition; network transmission

1 引言

随着嵌入式微处理器的推广和各种嵌入式操作系统的涌现,极大地促进了嵌入式系统技术的发展,更为图像采集与传输系统小型化、智能化的发展提供了无限的动力。目前具有图像采集与传输功能的嵌入式设备已广泛应用于各个领域,航天领域的探险小车,交通领域的智能视频交通控制系统,生活领域的小区安全监控,消费电子领域的手机等,都有它的身影。与传统 PC 相比,基于嵌入式系统的图像采集与传输系统具有体型小巧、携带方便的优点,解决了 PC 机体积大、布线复杂,具有使用场合限制的弊端。

将图像采集与传输和嵌入式技术相结合,实现图像采集与传输系统小型化就是本设计的主要目的。系统采用嵌入式 ARM 平台与嵌入式 Linux 操作系统相结合,利用内核驱动 V4L2 提供的编程接口操作 USB 摄像头采集图像,并显示在 LCD 上,同时在 PC 机上设计客户端,服务器与

客户端之间以 TCP/IP 协议进行网络通信,利用套接字编程实现视频图像数据的传输。

2 硬件系统设计

系统采用基于 ARM920T 内核的 S3C2440 作为核心处理器。它实现了 MMU(存储管理单元),可对内存进行有效管理。此外,它还采用了新的高级微控制器总线架构(AMBA)和哈佛高速缓存架构,工作频率可达 400 MHz,具有高速的处理计算能力,其内部还集成了外部存储器控制器、DMA 控制器、USB 主机控制器等丰富的片上功能,基本满足了系统采集图像、处理数据与传输程序流畅运行的设计要求。

外围电路配置有:暂存图像数据和临时用户数据的 64 M SDRAM;保存系统运行所必需的操作系统、应用程序、用户数据等各类数据的 256M NAND FLASH;

100 Mbps、低功耗、嵌入式专用的以太网网络芯片 DM9000,接口为标准 RJ45 插座;显示采集到的图像的 LCD 显示屏;用于与 PC 机进行通讯的 RS232 接口;用于往空白目标板烧写程序和调试的 JTAG 接口;用于与图像采集设备进行数据传输的 USB 接口。系统硬件结构如图 1。

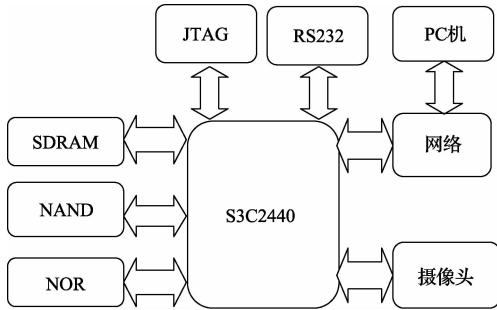


图 1 S3C2440 平台硬件结构框图

系统的外接图像采集设备选择了 USB 接口的 CMOS 摄像头,它采用台湾松翰电子公司制造的 SN9C255 处理芯片,遵循 UVC 协议。UVC,全称为 USB Video Class,是 Microsoft 与另外几家设备厂商联合推出的为 USB 视频捕获设备定义的协议标准。符合此协议的摄像头,即通常所说的“免驱动”摄像头,使摄像头实现真正意义上的即插即用。

3 系统的软件平台构建

嵌入式设备系统资源十分有限^[1],并不具备开发软件所需要的开发环境,因此需要在宿主机安装交叉编译工具链、工具库等开发工具,基于宿主机来进行软件开发,svglib 库的编译和后续部分需要用到的 libjpeg 库的移植。

系统的软件平台由 BootLoader、内核、根文件系统三部分组成^[2]。

3.1 Bootloader

Bootloader,即系统引导加载程序,是系统上电后执行的第一段程序^[3]。系统采用的 Bootloader 是通用性强、资源丰富、网络功能强大的 U-Boot,通过裁剪和配置 U-Boot 实现 tftp 下载功能,提高烧写程序的速度和便利性。

3.2 内核配置

系统采用 linux-3.4.2 版本作为系统的嵌入式操作系统。系统要求内核在配置^[4]时,需选中以下几项:

1)配置内核支持 USB 主控制器。

[*] USB support --->

{ * } Support for Host-side USB

< * > OHCI HCD support

2)将 V4L2 编译进内核,为图像采集程序提供编程接口函数。

-> Device Drivers

< * > Multimedia support --->

< * > Video For Linux

3)将 UVC 视频驱动编译进内核,这样就为 UVC 摄像头提供了驱动。

[*] Video capture adapters (NEW) --->

[*] V4L USB devices (NEW) --->

< * > USB Video Class (UVC)

4)配置网络支持

[*] Networking support

[*] Network device support --->

< * > DM9000 support

3.3 根文件系统

系统使用 busybox-1.20.0 来制作根文件系统,在完成对 busybox 的配置后,编译 busybox,再将编译生成的文件系统制作成 YAFFS2 格式,烧写进 NAND Flash 中,到此完成软件平台的构建。

4 图像的采集和传输程序设计

系统采用分层思想进行软件设计,将程序分为图像采集和网络传输两个模块,同时利用互斥量和条件变量,创建图像采集显示线程和网络传输线程,实现多线程同步工作^[5]。

4.1 图像采集模块

V4L2,全称 Video For Linux Version 2,是 V4L 的第二版本。它是 Linux 下开发视频设备程序的接口标准,使用分层方法对所有视频设备的驱动和应用编程提供一套完备的接口规范^[6-7]。UVC 驱动则是基于 V4L2 框架的摄像头驱动程序,它通过函数 video_register_device (struct video_device * vdev, int type, int nr)向 V4L2 的核心层注册 video_device 结构体,为应用程序提供操作接口函数和 I/O 控制接口函数,如 open(), read(), ioctl()等^[8]。

4.1.1 视频设备初始化

1)设置视频设备结构体

在图像采集部分中,定义一个用于保存图像数据和设备参数的数据结构体并对其进行初始化,初始化成员包括:

- videodevice: 视频设备文件节点
- width: 视频图像宽度
- height: 视频图像高度
- fps: 帧频
- formatIn: 输入图像格式

接下来根据参数的设置对视频设备进行初始化。

2)打开摄像头设备文件

将 USB 摄像头设备接到目标板上,系统会对设备进行识别,识别通过后就会自动创建设备节点,在系统中创建的节点为 /dev/video0,此后应用程序就是通过该节点完成与设备的信息交互。此节点在前面的设置中,已赋值给数据结构体的成员 videodevice,通过调用系统函数 vd->fd = open(vd->videodevice, O_RDWR)以读写的方式打开指定的设备文件,将返回的文件描述符保存在 vd->fd 中,后续程序通过此描述符来对设备进行操作。

3) 查询设备信息和设置采集属性

在 UVC 的驱动中,提供了各种带有不同命令参数的 ioctl 函数,这为开发者管理设备 I/O 提供了很大的便利。

调用函数 `ioctl(vd->fd, VIDIOC_QUERYCAP, &vd->cap)` 获取设备信息,并保存在用于存储设备信息的 `v4l2_capability` 结构体 `vd->cap` 中,然后逐项验证设备是否为设备是否为视频捕捉设备,该设备是否支持数据传输的方式等。

验证完毕后,可根据需求通过设置 `v4l2_format` 结构体变量 `vd->fmt` 以修改采集属性,然后调用 `ioctl(vd->fd, VIDIOC_S_FMT, &vd->fmt)` 将设置参数传递给视频采集设备。

4) 为视频数据分配缓存

为了保存采集到的图像数据,需要调用命令参数为 `VIDIOC_REQBUFS` 和 `VIDIOC_QUERYBUF` 的 `ioctl` 函数,以申请自定义数量的缓存并获取这些缓存的信息。但用户无法直接访问存放于内核空间的图像数据,利用 `read` 方式进行访问会由于其数据复制的步骤而降低效率,因此系统采用 `mmap` 方式将图像数据映射到用户空间^[9],加速 I/O 访问。之后调用 `ioctl(vd->fd, VIDIOC_QBUF, &vd->buf)` 将缓存放入输入等待队列中。

4.1.2 图像采集线程

在视频设备初始化完毕后,调用函数 `ioctl(vd->fd, VIDIOC_STREAMON, &type)` 使能摄像头设备,即向视频设备提交设置好的 `urb`,启动数据传输。当视频设备完成一帧图片的采集后,将其保存在等待队列的缓存 `vd->buf` 中,这时通过函数 `ioctl(vd->fd, VIDIOC_DQBUF, &vd->buf)` 将存有图像数据的缓存从队列中取出,利用 `memcpy` 函数将视频数据拷贝到临时缓存 `vd->tmpbuffer` 中,紧接着调用 `ioctl(vd->fd, VIDIOC_QBUF, &vd->buf)` 将使用完的缓存重新入列,以便接收图像数据使用。

至此完成一帧图像的采集,但由于系统使用的摄像头输出的数据格式为 YUV422,而 LCD 的显示数据格式为 RGB,因此需要进行格式转换。系统通过函数 `yuv422torgb565(unsigned char * input_ptr, unsigned char * output_ptr, unsigned int image_width, unsigned int image_height)` 将 YUV 图像数据转换为 RGB565 格式,并将其存放在 LCD 的显存 `vd->lcdbuffer`,完成图像在 LCD 上的显示,如图 2。

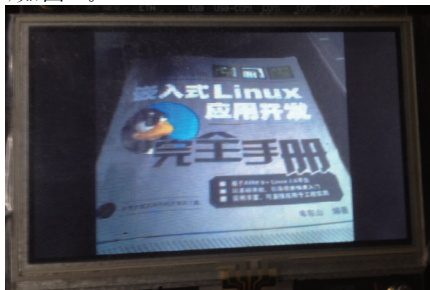


图 2 LCD 的图像显示

4.1.3 图像原始数据的处理

通过上面的采集程序得到的图像数据不便于网络传输和保存,为了提高网络传输效率,在网络传输之前,对原始图像数据进行压缩处理。

在前面已经将图像数据转换为 RGB 格式,那么接下来只需要利用开源的 `libjpeg` 库提供的函数对图像数据进行压缩,主要函数调用如下:

```
jpeg_create_compress (&cinfo)
dest_buffer (&cinfo, buffer, size, &written)
    /* 生成的 JPEG 数据存在 buffer 中 */
cinfo.image_width = vd->width;
cinfo.image_height = vd->height;
cinfo.input_components = 3;
cinfo.in_color_space = JCS_RGB;
jpeg_set_defaults (&cinfo);
jpeg_set_quality (&cinfo, quality, TRUE);
    /* 设置图像的宽、高、彩色位图、
    彩色图像和压缩质量 */
jpeg_start_compress (&cinfo, TRUE);
while (...)
{
    ...
    row_pointer[0] = line_buffer;
    jpeg_write_scanlines (&cinfo,
        row_pointer, 1);
    /* 将存放在 line_buffer 中的 RGB 数据
    逐行进行压缩 */
}

jpeg_finish_compress (&cinfo);
jpeg_destroy_compress (&cinfo);
```

4.2 网络传输模块

TCP^[10]是一种面向连接的、可靠的、基于字节流的传输层通信协议。它采用三次握手建立连接,并确定数据是否发送成功,若发送失败则重新发送数据,以避免数据丢失。鉴于此协议的可靠性,系统采用 TCP 协议传输数据。网络传输模块分为两个部分,服务器的数据传输部分和 PC 客户端的数据接收显示部分。

4.2.1 图像数据传输

系统通过套接字编程实现图像数据的传输^[11]。首先调用 `socket(..., SOCK_STREAM, ...)` 创建 TCP 数据报类型的套接字,然后配置服务器地址、端口号,分别调用 `bind()` 和 `listen()` 绑定端口,启动端口监测。在初始化结束后,在网络传输线程中通过一个死循环以阻塞方式等待客户端的连接,并在客户端连接上后,通过 `recv()` 函数来接收客户端发来的请求和确认信息,通过 `send()` 函数来发送报文和视频图像信息。

4.2.2 PC 客户端设计

PC 客户端要接收服务器发来的图像信息,首先需要与服务器建立连接,同样调用 `socket(..., SOCK_STREAM, ...)` 创建套接字,然后设置与服务器一样的端口号地址,之后通过 `connect()` 函数向服务器申请连接。连接上服务器后,也是通过 `send()` 函数向服务器发送请求和 `recv()` 函数接收服务器发送报文和视频图像信息。

接下来把接收到的视频图像数据显示到 PC 机上,实现在 PC 机上可以通过网络查询视频图像的功能。系统通过 `svglib` 库提供的函数将采集到的图像在 PC 机的控制台 `console` 上进行显示。`svglib` 是一套运行于 Linux 及 FreeBSD 下的开放源代码低阶绘图函数库,它允许程序设计人员在控制台上来回切换于文本界面和图像界面。实现步骤如下:

首先初始化显示设备,例如对显示图像的分辨率和像素点的 `bpp` 值进行设置。

```
vga_init();
vga_setmode(G640x480x64K);
gl_setcontextvga(G640x480x64K);
```

与在 LCD 上显示图像的步骤相似,也需要分配物理显存和缓存,对于视频显示设备,一般而言至少需要为其分配两块缓存用于存储图像数据,将两帧视频图像交替覆盖到物理显存上,可以更好地保证视频图像的流畅性。

最后利用 `svglib` 库提供的函数 `gl_setpixelrgb(int x, int y, int r, int g, int b)` 和 `gl_copyscreen(GraphicsContext * gc)` 在 PC 机的 VGA 显示屏上,将所采集到的图像数据在制定的位置逐行描画到像素点上,实现视频图像在 PC 机上的显示,效果如图 3。

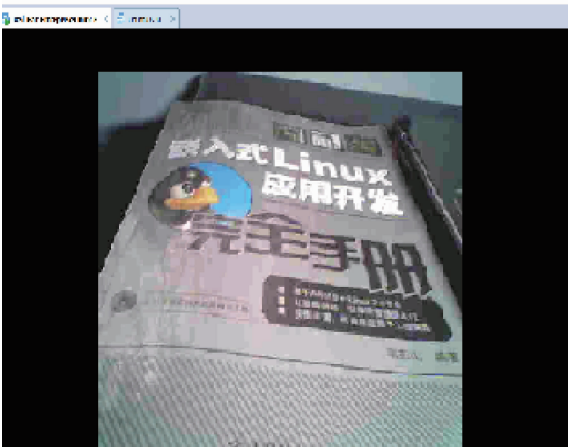


图3 PC 机客户端图像采集界面

5 结 论

本文在嵌入式 ARM 平台上结合嵌入式操作系统,采用外接 USB 摄像头和网络成功实现图像的采集和传输。其中的摄像头采用流行于市面上的基于 UVC 协议的 USB 摄像头,由于其“免驱动”的特性,省去了更换摄像头时需要更换驱动或者升级操作系统的麻烦,具有较强的通用性。此系统安装简便,成本低廉,可扩展应用于工业控制、智能家居、可视电话等诸多领域。

参考文献

- [1] 张杨,徐艳,朱家村,等. 嵌入式技术的应用研究[J]. 科技信息,2014(2):147.
- [2] 李传娣. 嵌入式系统关键技术分析与开发应用[J]. 黑龙江科学,2014,5(2):30-31.
- [3] 符秋丽. 基于 S3C2440 的 Bootloader 的分析与设计[J]. 微型电脑应用,2014,30(1):49-50.
- [4] 路青起,席丹丹. 嵌入式 Linux 系统移植[J]. 国外电子测量技术,2014,33(12):78-81.
- [5] 李树彪,韩敬伟. 基于多任务的智能测量仪器嵌入式软件设计[J]. 仪器仪表学报,2013,34(S1):1-7.
- [6] 唐人财,刘连浩. 基于嵌入式 Linux 远程图像监控系统的设计[J]. 计算机与现代化,2010(11):31-34.
- [7] 廖旭,张文普,彭烟,等. 基于嵌入式 Linux 的视频图像采集与传输[J]. 微计算机信息,2010(2):68-70.
- [8] 王宏志,宇皓林. 基于 ARM_Linux 远程视频监控系统设计的设计[J]. 吉林大学学报,2014,52(1):103-107.
- [9] 戚微苏,赵宇,潘勇. 基于 ARM-Linux 的车载监控终端设计及实现[J]. 电子测量技术,2012,35(2):85-88.
- [10] 车明玮,童金,王明宝,等. 基于网络的小车远程监控系统的设计[J]. 电子测量技术,2012,35(3):20-23.
- [11] 李彦,李镔. 基于嵌入式 Linux 系统的双网卡大数据传输[J]. 电子测量与仪器学报,2014,28(9):1027-1032.

作者简介

杨成禹,工学博士,讲师,主要研究方向为精密测控技术与仪器、现代测试计量技术与质量控制、智能结构系统与测量自动化等。

庄晓奇,硕士研究生,研究方向为现代测试计量技术与质量控制。

E-mail:zxq247@163.com