嵌入式 ARM 下的 USB 摄像头监控系统①

白立朋，李秋红，程　磊，王太宏

(湖南大学 微纳光电器件及应用教育部重点实验室，长沙 410082)

摘　要：介绍了一种应用于智能家居的视频监控系统实现方案。此设计以嵌入式 ARM 和 ARMS3C2440 为平台，利用 Video4Linux2 获得 USB 摄像头采集到的视频图像数据，并利用 SDL 在 PC 上实时显示了视频图像。

关键词：视频监控；ARM；USB 摄像头；V4L2；Socket；SDL

USB Camera Monitoring System Based on Embedded ARM

BAI Li-Peng, LI Qiu-Hong, CHENG Lei, WANG Tai-Hong

(Key Laboratory for Micro-Nano Optoelectronic Devices of Ministry of Education, Hunan University, Changsha 410082, China)

Abstract: A design of video monitoring system used in the field of smart home is introduced. It is based on the embedded ARM and S3C2440. Video4Linux2 is used to get the video image data captured by USB camera, and displaying the video image in real-time on PC with SDL.

Keywords: video surveillance; ARM; USB camera; V4L2; Socket; SDL

近年来，图像视频监控被广泛的应用在许多重要场合，对小区安防、智能家居、工厂的安全生产有着重要的意义。本文利用 ARMS3C2440 嵌入式设备和使用灵活方便的 USB 摄像头实现了视频图像数据的采集，并在客户端 PC 上实时显示采集到的视频图像，满足了视频监控功能的要求。

1　硬件系统介绍

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）、PC。其中USB摄像头和客户端PC分别采用USB数据线和双绞线与ARMS3C2440相联接。系统结构如图1所示。

首先在 S3C2440 上运行服务端程序，发送命令给 USB 摄像头准备开始采集视频数据，等待客户端程序的连接请求。在客户端 PC 上运行客户端程序后，初始化 SDL 显示程序，并向服务端发送连接请求，在服务端确认连接成功后，开始采集视频图像数据，向客户端传送视频数据。客户端在接收到视频数据后，通过 SDL 程序将视频图像显示在客户端的显示器上。

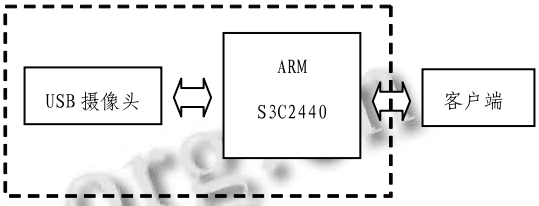


图 1　系统结构图

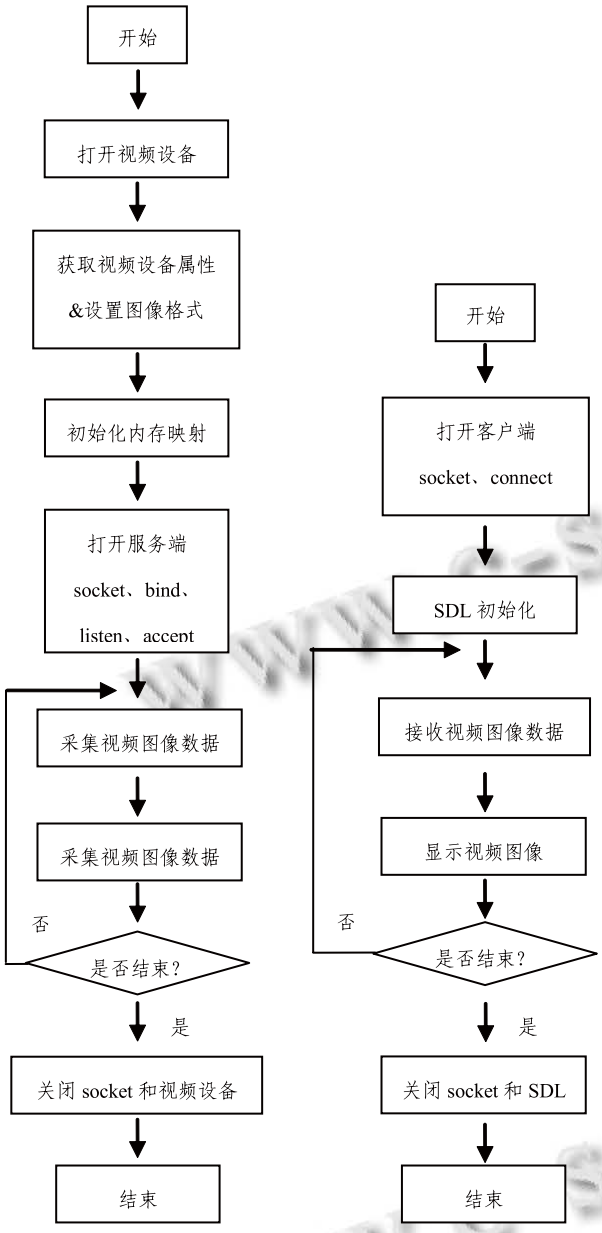
2　软件程序设计

软件部分首先采用 Video4Linux2 应用编程接口提供的数据结构和函数，实现视频图像数据的采集。然后将采集到的图像数据通过 Linux 下的 socket 编程传输到客户端 PC，客户端 PC 通过 Windows socket 编程接收到图像数据后，利用 SDL 开发包将视频图像显示出来。整个系统的流程图如图 2 所示：

2.1 Video4Linux2 程序设计

Video4Linux2 是 Video for Linux Two 的简称，是 Linux 内核中关于视频设备的子系统，使得应用程序可

① 基金项目:国家自然科学基金(21003041);国家 973 计划(2007CB310500);湖南省自然科学基金(10JJ1011)
收稿时间:2010-09-20;收到修改稿时间:2010-11-01



(a)服务端程序流程图 (b)客户端程序流程图

图 2 程序流程图

以使用统一的 API 函数操作视频设备，结合视频采集设备和相应的驱动程序实现图像采集，在远程会议、可视电话、视频监控系统中有着广泛的应用。

使用 Video4Linux2 应用编程接口提供的数据结构和函数，编写视频图像采集程序主要包括打开设备、获取设备属性信息、设置图像格式、分配缓冲区、采集图像数据、关闭设备等步骤[1,2]。具体操作如下：

①打开视频设备文件

```
int fd=open("/dev/video0",O_RDWR);
if(fd<0){perror("open camera failed!\n");exit(-1);}
```

成功打开视频设备后，会返回文件描述符 fd。

②获取视频设备属性信息

```
struct v4l2_capability cap;
if(-1==xioctl(fd,VIDIOC_QUERYCAP,&cap))
{fprintf(stderr,"vidioc_querycapwrong!\n");exit(-1);
}
if(!(cap.capabilitier&V4L2_CAP_VIDEO_CAPTURE))
{fprintf(stderr,"there is no video capture device\n");
exit(-1);}
if(!(cap.capabilities&V4L2_CAP_STREAMING))
{fprintf(stderr,"device dose not support streaming i/o\n");
exit(-1);}
```

③设置图片格式

将图片设置为大小 640×480，像素格式为 YUYV。

```
struct v4l2_format fmt;
fmt.type=V4L2_BUF_TYPE_VIDEO_CAPTURE;
fmt.fmt.pix.width=640;
fmt.fmt.pix.height=480;
fmt.fmt.pix.pixelformat=V4L2_PIX_FMT_YUYV;
if(-1==xioctl(fd,VIDIOC_S_FMT,&fmt))
{fprintf("FMT failed!\n");exit(-1);}
```

④分配缓冲区

本文采用 mmap 内存映射的方式获得图像数据。

```
struct v4l2_requestbuffers req;
req.count=4;
req.type=V4L2_BUF_TYPE_VIDEO_CAPTURE;
req.memory=V4L2_MEMORY_MMAP;
if(-1==xioctl(fd,VIDIOC_REQBUFS,&req))
{fprintf("VIDIOC_REQBUFS failed!\n");exit(-1);}
```

其后要分配用户的内存，在 for 循环中使用 mmap 函数在设备缓存和应用程序分配的内存间建立映射。然后将缓冲区放入视频采集队列。

⑤采集图像数据

```
type=V4L2_BUF_TYPE_VIDEO_CAPTURE;
ioctl(fd,VIDIOC_STREAMON,&type);
struct v4l2_buffer buf;
ioctl(fd,VIDOC_DQBUF,&buf);
```

采集到图像数据后，驱动会将一个缓冲区从采集

队列中取出。然后对得到的图像数据进行具体的处理，本设计中在此采用 socket 编程将得到的数据传送了到了客户端 PC。处理完数据后要将缓冲区重新放入采集队列中。

⑥关闭设备

先后使用 VIDIOC_STREAMOFF、munmap、close 函数完成设备的关闭。

2.2 socket 编程

本设计的服务端采用 linux socket 以流式 socket 编程实现对视频图像数据的传送。主要采用 socket、bind、listen、accept、send、close 几个函数完成相应的功能。

客户端采用 windows socket 编程接收服务端发送过来的视频图像数据。主要采用 socket、connect、recv、close 等几个函数完成相应的功能。

在数据传送过程中，服务端每次传送一帧图像的数据量，为确保客户端的接收函数 recv 能完全接收到一帧图像的数据，采用了如下方式：

```
char buf[w*h*2];
char *recvbuf=buf;
void socket_recv()
{
    recv_num=0;
    int sum=0;
    while(sum!=(w*h*2))
    {
    recv_num=recv(sClient,recvbuf+sum,w*h*2-sum,0);
    sum+=recv_num;
  }
  recvbuf=buf;
}
```

2.3 视频图像的显示

本设计是在 VC++6.0 环境下采用 SDL 编程实现的。SDL（Simple DirectMedia Layer）是一个自由的跨平台的多媒体开发包，简单易用，高性能和跨平台的优点，使它被广泛的应用于各种操作系统中。

在 VC++6.0 环境下使用 SDL 编程，先要搭建针对 SDL 的编程环境。

①解压 SDL-devel-1.2.14-VC6.zip。

②将解压后得到的 SDL.lib 和 SDLmain.lib 文件添加到 VC++6.0 的 lib 文件夹下。

③将解压后得到的 SDL.dll 文件添加到 WINDOWS /SYSTEM32 目录下。

④将解压后得到的 include 文件添加到 VC++6.0 的 include 文件夹下面。

⑤打开 VC++6.0,新建->project->win32 Application. 打开 project->setting->C/C++->Category-> Code Generation，Use run-time library->Multithread DLL。

⑥打开 project->setting->Link->Category->input，在 Object/library modules 中加入 SDL.lib 、SDLmain.lib。

编写视频显示程序主要包括 SDL 初始化、创建 display surface、创建 overlay、锁定 overlay、处理视频图像数据、解锁 overlay、显示图像、关闭退出[3,4]。具体过程如下：

① SDL_Init(SDL_INIT_VIDEO)

初始化 SDL 子系统函数，本设计中为初始化视频子系统，使用任何子系统前都要调用该函数来初始化

② screen=SDL_SetVideoMode(w, h, 32, SDL_HWSURFACE);

创建 display surface，同时制定创建的宽度、高度、像素的位值等。创建成功后函数会返回一个指向 SDL_Surface 的指针。

③ overlay=SDL_CreateYUVOverlay(w,h,SDL_YUY2_OVERLAY,screen);

创建 overlay。Overlay 用来将视频文件数据渲染到 surface 上面。创建成功后函数会返回一个指向 SDL_Overlay 的指针。

④ SDL_LockYUVOverlay(overlay);

锁定 SDL_Overlay。在访问 overlay 的像素数据前，先要锁定 SDL_Overlay，访问完毕后再解锁 overlay。

⑤ memcpy(overlay->pixels[0], pYUV, w*h*2);

处理视频图像数据。调用 memcpy 函数，将视频数据从缓冲中写入到 SDL_Overlay 结构体中。

⑥ SDL_UnlockYUVOverlay(overlay);

解锁 SDL_Overlay。

⑦ SDL_DisplayYUVOverlay(overlay, &drect);

显示图像。

⑧ SDL_FreeYUVOverlay(overlay);
SDL_FreeSurface(screen);
SDL_Quit();

关闭退出。先释放开始创建的 SDL_Surface 和

SDL_Overlay，再调用 SDL_Quit()函数关闭所有的 SDL 子系统。

3 运行结果

本设计客户端是在 PC（windows）上运行的，运行结果如图 3 所示：

图 3 图像显示结果

4 结语

本设计视频采集和显示系统分别基于 Linux 和 windows 系统平台，在分析了视频图像采集和播放技术的基础上，使用 Video4Linux2 应用编程接口和 SDL 开发包，实现了使用 USB 摄像头采集视频图像并在 PC 上实时显示的功能，在实际运行中显示了良好的效果。

参考文献

1 张聪敏,游向东.基于 V4L2 的远程图片采集系统.中国科技论文在线.2010.04.

2 Schimek MH. Video for Linux Two API Specification Revision0.24.[2010-08-20].http://v4l2spec.bytesex.org/v4l2spec/v4l2/pdf

3 Pazera E. Focus on SDL[2010-08-27].http://ishare.iask.sina.com.cn/f/9177718.html

4 江达秀,许建龙,孙树森.应用 SDL 及 GTK+实现视频多路回放.浙江理工大学学报,2009,26(6):897–900.

(上接第 139 页)

许，组播数据包就会被接受。原理如图 3 所示，软件部分要做的就是编一段程序来处理该设置组播地址寄存器中哪一位。

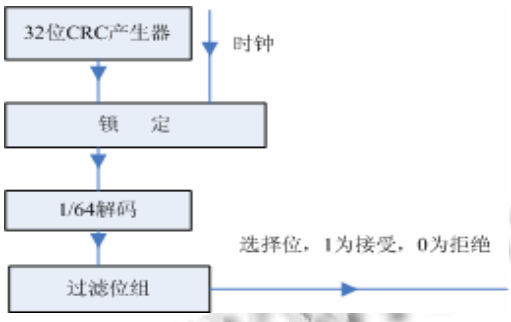


图 3 组播原理示意

5.5 ARP 协议

网络层的 IP 地址在数据传输的终点都要转化一个物理地址，所以网卡工作过程的另外一个环节就是实现地址解析，即 ARP 协议。除此之外，通讯过程中实际应用的数据信息是包含在传输层的 UDP 数据包中。所以对于帧传输协议格式，ARP 协议格式，IP 协议格式，用来探测并报告 IP 数据包传输中产生的各种错误的 ICMP 协议都要正确理解灵活应用。

当然，软件技术本身是一个方面，运用软件技术来实现实际需求是另外一个方面。这两方面都体现在软件的设计和调试过程中。

6 结论

当前以太网技术在中小型局域网领域应用十分普遍。这种嵌入式 TCP/IP 协议的单片机系统,具有成本低、硬件少、占用面积少、传输速度快、使用方便等优点，有着广泛的应用前景,特别是数据传输、数据采集领域。

本次设计的模块不仅成功应用于某型雷达，更是类似产品改造和更换像 RS232、RS422 以及 1553B 这样的通讯方式，使用 Ethernet 网络技术的实用参考。

参考文献

1 鲁士文.计算机网络协议和实现技术.北京:清华大学出版社,2000.

2 蔡宁果,何晓琼.用 8 位单片机实现串口-以太网转化器.电子技术应用,2002,(2):14–16.

3 葛永明,林继宝.嵌入式系统以太网接口的设计.电子技术应用,2002,(3):25–27.