基于 ARM11 的无线视频监控系统①

王　刚，毛剑飞，田　青，毛飞飞

(浙江工业大学 计算机科学与技术学院，杭州 310014)

摘　要：设计了一种基于 ARM11+Linux 系统组成的无线视频监控系统。为提高运算速度，系统采用 ARM11 微处理器 S3C6410 作为主处理器，采用 OV9650 摄像头作为图像采集设备，用硬编码方式对图像数据进行 H264 编码。接着通过 WI-FI 无线网络和 RTP 流媒体传输协议把已编码的数据传送到远端的服务器上显示。实现了运行稳定、速度快、成本低、体积小的无线视频监控平台，具有很大的实用价值。

关键词：视频监控；摄像头；H264；WI-FI；RTP

Wireless Video Surveillance System Based on ARM11

WANG Gang, MAO Jian-Fei, TIAN Qing, MAO Fei-Fei

(Department of Computer Science and Technology, Zhejiang University of Technology, Hangzhou 310014, China)

Abstract: This paper designs a wireless video surveillance system based on ARM11 and Linux operating system. To improve processing speed, the system adopts ARM11 as the main processor, and the video capturing equipment is OV9650 camera, also uses hard-coding to compress the video data with H.264. Then it can transfer encoded data to remote server and display that. It realizes a wireless video surveillance platform, which has stable running, quick speed, lower expenditure and small volume. It has great practical value.

Key words: video surveillance; camera; H264; WI-FI; RTP

1　引言

随着无线网络的普及，ARM 处理器运算的能力不断地增强以及计算机处理图像的技术不断地提高，基于 ARM 的视频监控正越来越广泛的应用于学校，社区，酒店，网吧，医疗等各种各样地领域。传统的视频监控系统布线复杂，设备庞大，智能化低，以及软硬件资源得不到充分的发挥。而 ARM 嵌入式系统的小型化、占用空间小、成本低廉、结构紧凑、支持无线网络等特点，使得利用 S3C6410 的 ARM11+linux 系统构成各种各样的无线网络数字监控系统具有广泛的应用价值。

2　系统整体设计

2.1 硬件总体设计

本系统采用韩国三星公司 ARM11 内核的 S3C6410 作为微处理器，该款处理器体积小，尺寸仅相当于一个 48mm*67mm 方块的大小，同时集成了丰富的接口，32 位数据总线和 32 位外部地址总线，SROM 控制器、 SRAM 控制器、NAND 闪存控制器、64 个中断源的中断控制器、五个三十二位定时器、四个 UART、四个 DMA 控制器、STN 与 TFT LCD 控制器、看门狗、IIS 音频接口、IIC-Bus 接口、两个 USB host 口、一个 USB device 口、两个串行外围接口电路、三个 SD 卡接口、camera_if 接口、TV_out 接口、MFC 接口、2 路 SPI、Touch Screen 接口，其主频可达 800MHz，扩展总线最大频率 133MHz。在此基本上，还进行了相关的扩展，引出了一个四线 RS-232 串口，该串口用于开发主机与 S3C6410 开发平台进行通信；配置了 1GB 的 NANDFLASH，用于存放嵌入式 linux 操作系统，应用程序和数据，128MB 的 DDR 内存，用于存放运行程序，摄像头捕获的数据；扩展了一个 WIFI 模块，用于开发平台与服务器传输视频数据，通

① 基金项目:浙江省自然科学基金(Y1090335)
收稿时间:2010-11-10;收到修改稿时间:2010-12-10

过无线网络实现视频远程监控。

2.2 软件总体设计

软件总体结构包括引导加载程序 Bootloader、操作系统内核，设备驱动程序和应用层程序，其软件结构如图 1 所示。

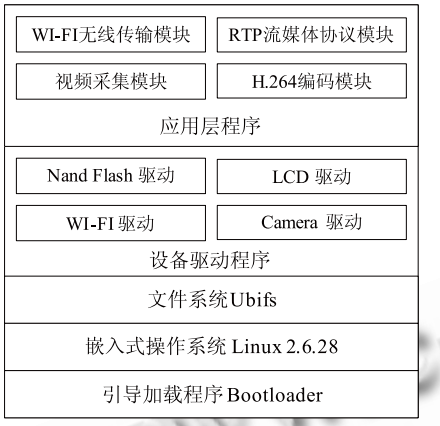

图 1 软件总体结构框图

该系统上电后，先运行引导加载程序 Bootloader，该程序的作用是初始化硬件设备、建立内存空间的映射表，引导和加载操作系统内核，然后启动嵌入式操作系统 linux，接着加载 Nand flash 驱动程序、LCD 驱动程序、WIFI 驱动程序等一些必要的驱动程序[1]。

3 视频数据采集和编码设计

3.1 基于 V4L2 视频数据采集设计

在 Linux 系统下，对视频设备的各种操作是通过 Video4Linux2 实现的，简称 V4L2[2]。应用程序通过 V4L2 提供的接口函数实现视频设备的操作。整个视频数据采集的过程如图 2 所示。

（1）打开视频设备，int open(const char * pathname, int flags)。调用该函数，若返回值为-1，表示打开失败，否则，表示所打开设备的文件描述符。

（2）取得设备信息。通过 ioctl(cam_fp, VIDIOC_ QUERYCAP, &cap)函数来取得设备文件的属性参数并存储于 cap 结构中，其中 cam_fp 指的是打开的视频设备的文件描述符。

（3）选择视频输入方式。通过 ioctl(cam_fp, VIDIOC_ S_INPUT, &chan)函数设置视频设备的输入方式，其中 chan 的数据结构类型是 v4l2_input，用来指定视频的输入方式。

（4）设置视频帧格式。通过 ioctl(cam_fp , VIDIOC_S _FMT, &fmt)函数设置视频的帧格式，其中 fmt 的数据结构类型是 v4l2_format，用来指定视频的宽度、高度、像素大小等。

（5）读取视频数据。通过 read(cam_fp, g_yuv, YUV_ SIZE)函数，把摄像头一帧的数据存放到 g_yuv 中，其中 YUV_ SIZE 指的是每帧数据的大小。

（6）关闭视频设备。通过 close(cam_fp)函数来实现视频设备的关闭。

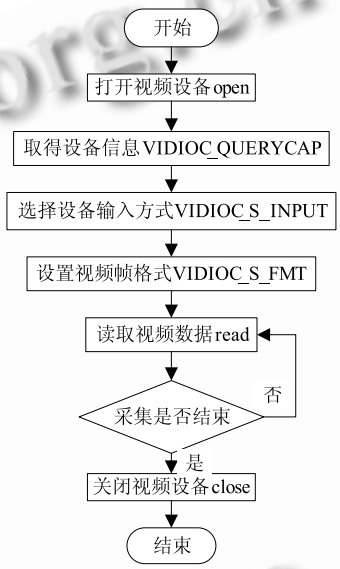

图 2 视频数据采集流程框图

3.2 视频数据的 H264 编码

为了提高视频数据编码速度，本系统采用的是 H264 硬编码方式，硬编码具有不占用 CPU 资源，运算速度快等优点，从而满足视频数据实时性的要求[3]。具体编码的过程如图 3 所示。

（1）创建 H264 编码结构。调用 SsbSipH264 EncodeInit (width, height, frame_rate, bitrate, gop_num)函数实现的，其中 width 表示图像的宽度，height 表示图像的高度，frame_rate 表示帧频，bitrate 表示比特率或码率，gop_num 表示两个相离关键帧之间最多包含多少个帧(B 或 P 帧)。

（2）初始化 H264 编码结构，调用 SsbSipH264 Encode Exe (handle)函数。

（3）获取视频输入地址，SsbSipH264EncodeGet InBuf (handle, 0)函数来实现，该函数返回视频输入的首地址，存放在 p_inbuf 中。

（4）输入视频数据，调用 memcpy(p_inbuf, yuv_buf, frame_size)函数实现，p_inbuf 存放需要编码的数据，yuv_buf 存放原始视频数据，frame_size 表示数据的大小。

（5）编码视频数据，对 p_inbuf 内容进行 H264 编码，调用 SsbSipH264EncodeExe(handle)函数实现。

（6）输出已编码的数据，SsbSipH264EncodeGet OutBuf (handle, size)，该函数返回已编码图像的首地址，size 表示已编码图像的大小。

（7）关闭硬编码设备，调用 SsbSipH264Encode DeInit (handle)函数实现的。

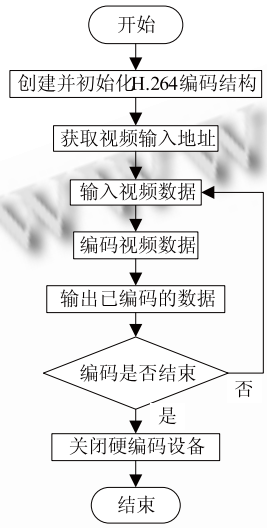

图 3　H264 编码流程框图

4　视频数据的传输和显示

4.1 视频数据传输模块设计

现代无线通信网络标准主要有 3G（第三代移动通信），WI-FI，Bluetooth，Zigbee（紫蜂）等[4]，具体详见表 1。

表 1　常用无线通信网络标准的基本比较

类型	3G	WI-FI	Bluetooth	Zigbee
协议	WCDMA CDMA2000 TD-SCDMA	IEEE 802.11 a/b/g	IEEE 802.15.1	IEEE 802.15.4
最高传输速率	2Mbps	11~54Mbps	1Mbps	10Mbps

由于 WI-FI 具有传输率高，支持的协议多，安装及设置简单，成本低等优点，所以本系统采用的无线网络标准是 WI-FI。

4.1.1 WI-FI 无线网络搭建过程

（1）加载 WI-FI 模块。通过 insmod 命令加载，这里需要加载 2 个文件 helper_sd.bin、sd8686.bin，这 2 个文件可以从 Marvel 官方网站下载。

（2）搜索 WI-FI 网络。先用 ifconfig eth1 up 命令把 WI-FI 网络接口卡打开，然后用 iwlist eth1 scanning 命令搜索 WIFI 网络。

（3）设置 eth1 的 ip 地址和子网掩码。

（4）设置 ESSID。通过 iwconfig eth1 essid 402 命令实现的，ESSID 用来区分不同的网络。

（5）设置密码。通过 iwconfig eth1 key s:your_key 命令实现的，其中 your_key 就是登陆密码。

4.1.2 基于 RTP 协议的视频数据传输

RTP 是实时传送协议（Real-time Transport Protocol）的缩写，代表一个网络传输的协议，为音频、视频上传中的常用协议[5]。RTCP 和 RTP 一起提供流量控制和拥塞控制服务，它们能以有效反馈和最小开销使传输效率最佳化，因而特别适合传送实时的数据，所以采用该协议传输视频数据。

本系统采用开源代码 Jrtplib 提供的 RTP 协议栈，由于 Jrtplib 对 RFC3550 的实现进行了封装，使得传输视频数据更加简单。由于本系统的网络最大有效载荷为 1500 字节，因此设置 RTP 包大小的上限为 1400 字节，如果发送的数据大于 1400 字节，则采用拆包的方法再发送，具体传输过程如图 4 和图 5 所示。

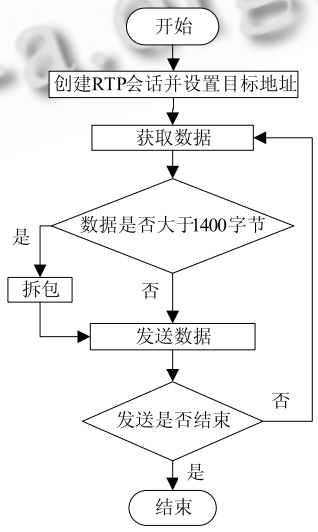

图 4　发送端流程框图

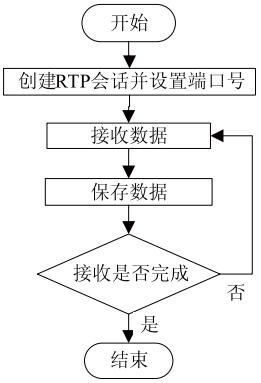

图 5　接收端流程框图

发送端主要过程如下：

（1）创建 RTP 会话并设置目标地址。调用 Create 方法得到 RTP 会话实例，然后调用 AddDestination 方法设置目标 IP 以及目标端口号。

（2）获得数据，调用 Get_Data()函数得到。

（3）发送数据，通过 SendPacket()方法实现。

接收端主要过程如下：

（1）创建 RTP 会话。调用 Create 方法来创建一个会话实例，并且在创建会话的同时设置端口号，要与发送端的端口号保持一致。

（2）接受 RTP 数据。调用 RTPSession 类的 PollData()方法接收数据。

（3）保存 RTP 数据报。通过创建了一个指针数组，里面存放的是 RTP 数据报的指针，只要将刚接收到 RTP 数据报的指针赋给这个指针数组即可，这样可以节省数据拷贝的时间。

（4）判断是否接收完成，如果没有，则跳转到第 b 步，否则接收端程序退出。

4.2 视频数据的解码和显示

由于接收到的数据是经 H264 编码的数据，因此，先要对该数据进行解码，然后才能显示。而在服务器端，对视频数据解码用到 FFmpeg。FFmpeg 是一个开源免费跨平台的视频和音频流方案，属于自由软件。解码时主要涉及 FFmpeg 下的 libavcodec 库、libswscale 库和 libavformat 库，其中第一个库是一个包含了所有 FFmpeg 音视频编解码器的库，第二个库是格式转化库，因为解码后的数据是 YUV420 格式，而要在计算机上显示该数据，则需要的是 RGB 格式的，该库功能就是把 YUV420 格式转化成 RGB 格式，第三个库是一个包含了所有的普通音视格式的解析器和产生器的库。

4.2.1 初始化解码线程。

(1) 注册全部的文件格式和编解码器，调用 av_register_all()函数完成注册。

(2) 设置 AVFormatContext 结构体。该结构体是 FFmpeg 格式转换过程中实现输入和输出功能，保存相关数据的主要结构，通过 av_open_input_file 函数设置该结构体。

(3) 检查视频流的信息，通过调用 av_find_stream_info(pFormatCtx)函数，pFormatCtx->streams 就填充了正确的视频流信息，pFormatCtx 类型是 AVFormatContext。

(4) 得到编解码器上下文，pCodecCtx= pFormatCtx -> streams[videoStream]->codec，pCodecCtx 指针指向了流中所使用的关于编解码器的所有信息。

(5) 打开解码器，先通过 avcodec_find_decoder 函数找到相应解码器，然后调用 avcodec_open 函数打开解码器。

(6) 申请内存用来存放解码数据，通过调用 avcodec_alloc_frame 函数实现，由于解码的数据是 YUV420 格式的，因此还需要将该数据转换成 RGB 格式，因此，再次调用 avcodec_alloc_frame 申请内存用来存放 RGB 格式数据。

(7) 申请内存用来存放原始数据，因为 H264 解码时，对于 P 帧需要参考前面一个关键帧或 P 帧，而 B 帧需要参考前后帧，因此需要存放原始数据，首先，用 avpicture_get_size 来获得需要的大小，然后调用 av_malloc 函数申请内存空间。

(8) 通过调用 avpicture_fill 函数将帧和新申请的内存结合起来。

(9) 创建格式转换上下文，通过 img_convert_ctx= sws _getContext(src_w, src_h，src_pix_fmt, dst_w, dst_h, PIX_FMT_RGB24, SWS_BICUBIC, NULL, NULL, NULL)方法实现。其中，src_w 表示源图像的宽度，src_h 表示源图像的高度，src_pix_fmt 表示源图像的格式，dst_w 表示目标图像的宽度，dst_h 表示目标图像的高度，PIX_FMT_RGB24 表示目标图像的格式。

4.2.2 对数据进行 H264 解码。

(1) 获得需要解码的一帧数据，由于前面接收端线程已经把接收到的数据存放在一个指针数组中，因此，解码线程只需要从指针数据中获取数据即可。

(2) 解码数据。调用解码函数 avcodec_ decode_video(pCodecCtx， pFrame， &finished， encoded Data，size)来解码视频文件。其中，参数 pCodecCtx 是前面得到视频流编码上下文的指针；参数 pFrame 存储解码后的图片的位置，参数 finished 用来记录已完成的帧数；参数 encodedData 是输入缓冲区指针，指向要解码的原始数据；参数 size 是输入缓冲区的大小。

(3) 将已解码的视频数据 YUV420 格式转换成 RGB 格式，通过调用 sws_scale()函数实现格式转换。

4.2.3 视频数据的显示

本系统使用 QT 下的 QImage 显示视频数据，由于

QImage 能够存取单个像素[6]，这样在显示前一帧图像的时候，将该图像保存下来，当显示后一帧图像的时候，如果该像素值与前一帧相同，则不必修改该值，从而节省了大量的时间，即哪里变修改哪里，显示过程的具体步骤如下：

(1) 取得已解码的视频数据，且该数据是 RGB 格式的。

(2) 循环取得视频数据的 R 分量、G 分量、B 分量。

(3) 判断该点的像素值是否与前一帧对应位置的像素值相同，若相同，跳转到第 2 步，否则，保存该像素值。

(4) 对取得的 RGB 各自分量，构造该像素点的颜色值，通过调用 qRGB（R,G,B）构造方法实现。

(5) 设置相应点的像素值，首先生成 QImage 类的对象，然后调用该类的 setPixel(x,y,rgb)。其中，x 是图像的 x 坐标值，y 是图像的 y 坐标值，rgb 是该点的颜色值。

(6) 显示图像，通过调用 update()方法，该方法会触发绘画事件，因此，在绘画事件里，写入显示图像代码，即可显示刚生成的 QImage 对象，通过调用 drawImage()方法绘制图像。

5 结论

本系统在视频图像采集时，为了降低数据量，采用 YUV420 的采样格式。视频数据编码采用 H264 硬编码方式，极大地提高了编码速度。而在无线网络传输时，考虑到丢包问题，将编码数据进行拆包然后发送，降低了丢包率。经测试，本系统采集一幅 OV9650 摄像头拍摄的且分辨率为 320X240 的图像，经 H264 硬编码，编码后的图像数据大致为 5KB 左右，降低了数据传输量，并且硬编码每秒可编码 25 帧图像数据，达到实时视频数据编码的要求。对于 WI-FI 无线网络的传输率一般在 11-54Mbps 左右，因此，该无线网络可以满足实时传输视频的需求。本系统构建了高实时性，低成本，低功耗的数字化无线视频监控平台，在该平台基础上，可以搭建各种各样的应用，比如，路况实时监控，人脸识别，仓库报警等应用，该系统具有一定的实用价值。

参考文献

1 王洪辉.嵌入式系统 Linux 内核开发实战指南.北京:电子工业出版社,2009.142–175.

2 弓雷.ARM 嵌入式 Linux 系统开发详解.北京:清华大学出版社,2010.324–336.

3 余兆明.图像编码标准 H.264 技术.北京:人民邮电出版社,2006.14–128.

4 螺丝起子研究室.共享 Wi-Fi 无线网络实务.北京:中国水利水电出版社,2005.4–39.

5 齐俊杰,胡洁,麻信洛.流媒体技术入门与提高.长沙:国防工业出版社,2009.71–79.

6 蔡志明,卢传富,李立夏.精通 Qt4 编程.北京:电子工业出版社,2008.175–183.

(上接第 17 页)

wireless body area networks: a survey and outlook. IEEE Communications Magazine, 2009,47(12):84–93.

2 Aghdasi HS, Abbaspour M, Moghadam ME, Samei Y. An energy-ffficient and high-quality video transmission architecture in wireless video-based sensor networks. Sensors, 2008,8:4529–4559.

3 Ng KT, Chan SC, Shum HY. Data compression and transmission aspects of panoramic videos. IEEE Trans. on Circuits and Systems for Video Technology, 2005,15(1):82–95.

4 Chen M, Gonzalez S, Vasilakos A, Cao H, Leung VCM, Body area networks: a survey. Mobile Networks and Applications, 2010, DOI: 10.1007/s11036-010-0260-8.

5 Ecma International. High rate ultra wideband PHY and MAC standard. ECMA-368, 3rd edition, 2008.

6 马力,张茂军,徐玮,熊志辉,王瑜.采用视频拼图方法构建高分辨率全景视频监控系统.中国图象图形学报,2008,13(12):2291–2296.

7 Joint Video Team of ITU-T VCEG and ISO/IEC MPEG. WD 1 reference software for MVC (JMVC) 1.0,JVT-AA212, Geneva, April 2008.