

Real Time Video Monitoring System Based On OMAP3730

M. Sunder Rao.
(M. Tech). Embedded Systems.
Aurora's Technology Research Institute.
Hyderabad.

S.Karunakar Reddy
Assoc. professor. ECE,
Aurora's Technology Research Institute.
Hyderabad.

Abstract:- This paper achieves the design of an embedded Web server, Linux as its operating system. This is because Linux can be reduced and transplanted. The method used to transplant Web server Boa on the embedded Linux platform is also discussed in detail, and through CGI technology functions of dynamic Web page is successfully realized. Real time video and Image monitoring system should have the characteristics of capturing, compressing and network transmission .Remote image monitoring system based on embedded Linux and OMAP3730 can achieve these functions well. It can be divided into two parts, server and browser. The first component is to capture video through the analog camera, compress the image into a jpeg format and video into mp4 format. Send the image and data compressed to the target pc through internet. The second component is to receive the image and data and display them on the browser. Experiments show that the video and images can be displayed clearly and smoothly.

Keywords- OMAP3730; Linux; video surveillance; embedded system.

I.INTRODUCTION:

With the development of the network technology, embedded technology and digital image processing technology, the embedded video monitoring system become a hot research. Compared with the previous system, it has many advantages, such as low power consumption, small size, and high integration and so on.

This paper designs an embedded remote video monitoring system. It is based on the Linux operating system. Using OMAP3730 as its center processor, using the analog camera to capture the images, it can capture the video, compress the image and transmit the data stably .when opening browser, java applet

will be downloaded; its role is to receive and display the video.

II.ARCHITECTURE OF EMBEDDED VIDEO MONITORING SYSTEM

The architecture of the system is shown in Figure 1. Analog camera is used to capture analog video signal this kind of signal is PAL format. In order to get digital signal, decoder circuit is needed. TI tvp5146 chip can convert PAL into UYVY; this part is put into server.

Server plays a most important role in this system. It is built on devkit8500 development board and Linux operating system, consists of web server and video server. when entering the server's IP address in the browser, web server can provide the test webpage for us, at the same time java applet embedded in this webpage would be downloaded into target pc and run, receive the image data and display. video server is in charge of controlling the camera, compressing the video images and transmitting the data.

Target pc should be installed JDK software package and internet explorer, providing java applet's operating environment.

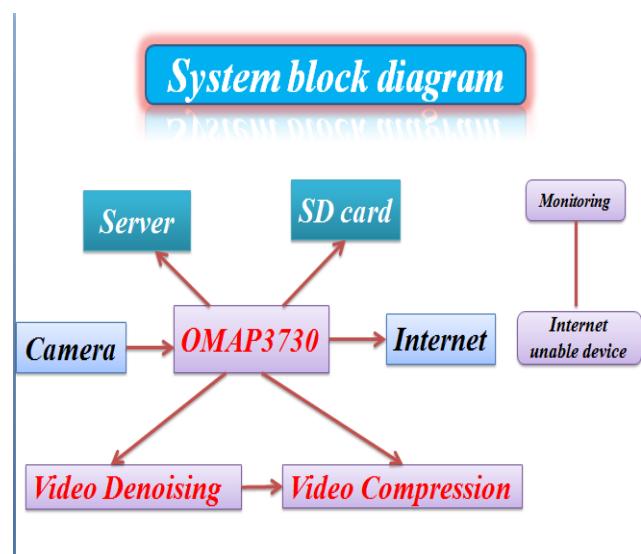


Figure 1.System block diagram.

III.TRANSPLANT WEB SERVER

There are three embedded web servers under the Linux operating system: httpd, thttpd and boa. Httpd is a static web server not supporting CGI, its function is weakest. thttpd supports CGI, but when running, it requires far greater system resources than boa. In view of this, boa is selected as the web server in this system. I had built the boa and vlc web server on Linux host system and tested the web servers.

Need to build the web server for target board. A step to build boa webserver is as follows.

Step 1: Go to <http://www.boa.org> and download the latest version.

Step 2: Go to the location where you downloaded the package.

cd <location of downloaded boa package>

Step 3: Unpack it by giving following command:

```
tar -xvzf boa-0.94.tar.gz
```

Step 4: Copy un-packed directory to your Preferred location. Let's assume it's '/home' for this tutorial. So path for you.

Step 5: cd /home/boa-0.94/src

```
gedit define.h
```

Step 6: Change the default SERVER_ROOT by setting the

```
#define at the top of defines .h
```

```
SERVER_ROOT = /home/boa-0.94/
```

Step 7: Go to parent directory (/home/boa-0.94/) and create

Following files and directory: access log error log -> files

Www

Step 8: Type. /configure and then make.

```
./configure
```

```
Make
```

Step 9: Copy boa.conf file from 'examples' to parent directory and open it using gedit.

```
cp /home/boa-0.94/examples/boa.conf /home/boa-0.94/
```

```
gedit boa.conf
```

Step 10: Go to 'src' directory and give following command.

```
sudo ./boa
```

Step 11: Open your internet browser and type following in the address bar:

<http://127.0.0.1> Enter

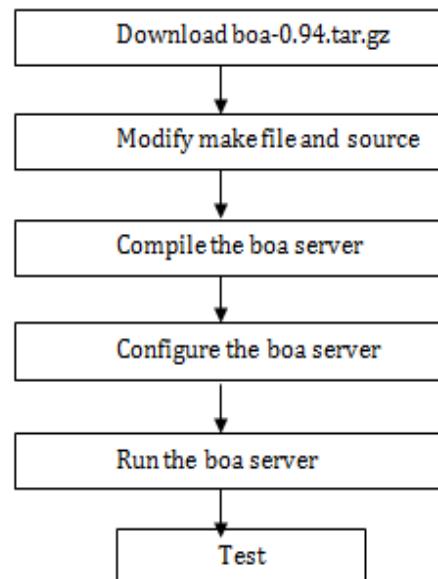


Figure 2. Process of transplanting boa.

IV.BUILD VIDEO SERVER

Video server mainly achieves three functions: control the camera to capture video, compress image and transmit data. It works as shown in Figure 3.

A. Video Capture on Video4Linux2

Video4Linux is a video capture application programming for Linux. Several USB webcams, TV tuners, and other devices are supported. Video4Linux is closely integrated with the Linux kernel. Video4Linux2 is the second version of Video4Linux, it fixed some design bugs and started appearing in the 2.5.X kernels. The driver of tvp5146 chip used in this system is written in accordance to V4L2 standard.

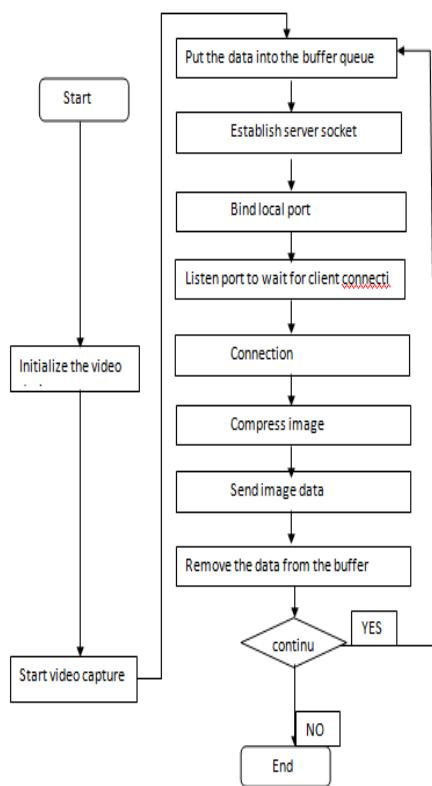


Figure 3. Video server workflow

1).Initialize the video device:

a) Open the video device:

capture_fd=open("/dev/video0",O_RDWR) capture_fd is the file descriptor, represents the capture device.

b) Get any active input and enumerate input to get the name of the input detected:

IOCtl (capture_fd, VIDIOC_G_INPUT, &index)

Ioctl (capture_fd, VIDIOC_ENUMINPUT, &input)

Ioctl (capture_fd, VIDIOC_QUERYSTD, &std)

c) Detect and get the standard in the input detected:

Ioctl (capture_fd, VIDIOC_QUERYSTD, &std_id)

Ioctl (capture_fd, VIDIOC_G_STD,&std_id)

d) Read and set the video capture format of the current

Driver: ioctl (capture_fd, VIDIOC_G_FMT,fmt)

Ioctl (capture_fd, VIDIOC_S_FMT,fmt)

e) Allocate memory and convert it into physical address:

Ioctl (capture_fd, VIDIOC_REQBUFS,&reqbuf)
ioctl(capture_fd,VIDIOC_QUERYBUF,&buf)

2) Start video capture:

Ioctl (capture_fd, VIDIOC_STREAMON,&a)

3) Put the data into the buffer queue:

Ioctl(capture_fd,
VIDIOC_DQBUF,&capture_buf)

4) Remove the data from the buffer queue:
Ioctl (capture_fd,VIDIOC_QBUF,
&capture_buf)

B. JPEG Compress

Making image files smaller is a plus for transmitting files across networks.JPEG can easily provide 20:1 compression of full-color data. When network transmission is involved, the time saving from transferring a shorter file is a must for real-time. At the same time, PEG is one of the formats that java applet support. That is why using JPEG.

1)In order to using the functions that jpeg-6b library provide, we should transplant it into host. When compiling the program, it can be linked.

2) JPEG cannot copress the image data in UYVY format directly. Converting the data into RGB24.

$$R = Y + 1.14V$$

$$G = Y - 0.39U - 0.58V$$

B = Y + 2.03U
int UYVY2BMP(unsigned char *pdst,unsigned char *psrc, int width, int height)

3) Compressing the image using the functions in the jpeg-6b library.

a) Jpeg_creat_compress (&cinfo): apply and initialize the object that JPEG is going to compress.

b) Dest_buffer (&cinfo,buffer,size,&written):jpeg_stdio_dest from libjpeg cannot store compressed pictures to memory instead of a file,Modifying the codes to meet this requirement.

c) Jpeg_set_defaults (&cinfo):set compression parameters

Include image width, image height and so on.

d) Jpeg_set_quality (&cinfo,70,TRUE):set the quality of the compression.

e) Jpeg_start_compress (&cinfo):star compression.

f) Jpeg_finish_compress (&cinfo):finish compression.

C. Image Data Transmit:

1)Sockfd=socket (AF_INET,
SOCK_STREAM,0):establish a socket communication endpoint.

2) bind (sockfd,&server_sockaddr,sizeof(struct sockaddr)):

Connect the socket communication endpoint with a particular Logical network.

- 3) Listen (sockfd, BACKLOG): set the socket into passive State.
- 4) accept (sockfd,&client_sockaddr,&sin_size):handle client connection.
- 5) send (client_fd,buffer,n,MSG_WAITALL):send image data to the client.

V. VIDEO IMAGE DISPLAY

Java applet is embedded in the web page .when users visit the web page with the applet, it will be automatically Executed, No need to install any special application onto the Client, just open the browser. In this system, java applet is Mainly used to receive the image data and display the image. It is embedded in the web page which is stored in the web server that we have transplanted. Its workflow is shown in the Figure 4.

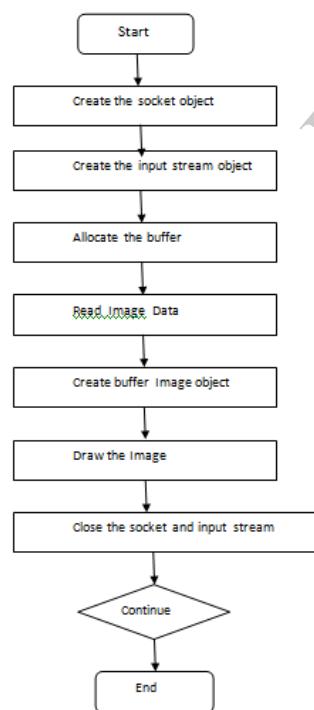


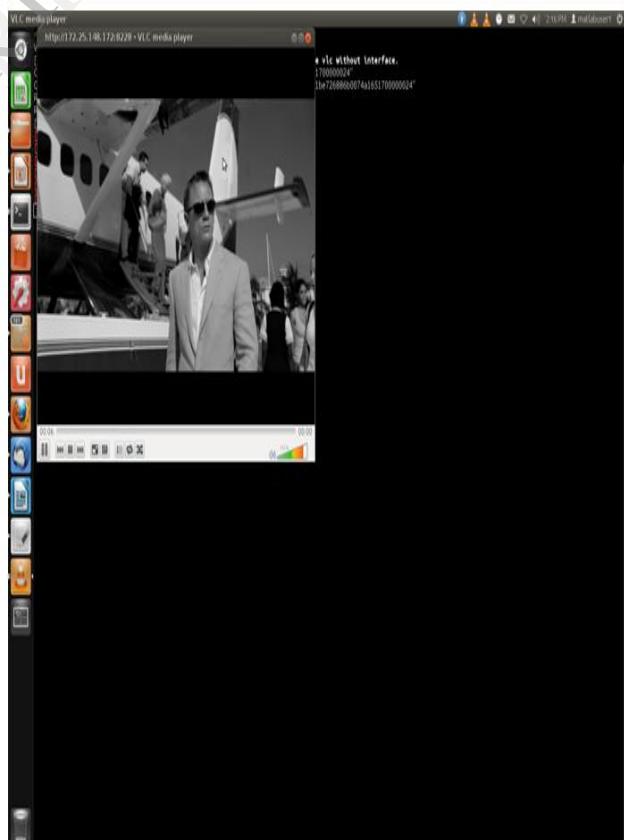
Figure 4. java applet workflow

- 1) Socket=newSocket("192.192.192.200", 7777):Create the socket object and found the connection with the server.

- 2) Input Stream in stream = socket. Get Input Stream ():Create the input stream object.
- 3) byte[] buffer = new byte[6000]:allocate the buffer to store the image data.
- 4) In stream. Read (buffer, 0, buffer.length): read the data from the input stream and store them into buffer.
- 5) Image=ImageIO.Read (newByteArryInputStream (buffer) :Create the buffered image object.
- 6) public idpaint(){g.drawImage(image,0,0,this);}: This function is used to draw the image.
- 7). Instream.close (), socket.close (): close the socket and the input stream.

VI.CONCLUSION:

Run boa server and video server, open the browser and enter



the URL "192.192.192.200/video.html", we can see the system test result that is shown in Figure 6. Experiment shows that the system runs stably.

Clear and coherent image can be display. The system can meet the requirements of the remote image monitoring system.

REFERENCES:

[1]Zhang Wei-wei, Zhao Ji-yin,Li Min, "High Definition Real-Time Monitoring System Based on Embedded Network Streaming Media,"2009 2nd International Congress on Image and signal Processing, pp.1-3, October 2009.

[2]Yakun Liu,Xiaodong Cheng, "Design and implementation of embedded Web server based on arm and Linux," 2010 2nd International Conference on Industrial Mechatronics and Automation,vol.2,pp.316-319,May 2010.

[3]Liu Yang, Linying Jiang, Heming Pang, "Design and Implementation of the Lab Remote Monitoring System Based on Embedded Web Technology,"2010 International Forum on Information Technology and Applications, vol.2, pp.172-175,July 2010.

[4]Yakun Liu,Xiaodong Cheng, "The study of video acquisition and compression codec system based on DaVinci," 2010 International Conference on Information Networking and Automation, vol. 2, pp.317- 321, October 2010.

[5]Yong-tao Zhou,Xu-ping Wang,Chun-jiang Yao, "Design of Equipment Remote Monitoring System Based on Embedded Web," 2008 International Conference on Embedded Software and System Symposia,pp.73-78, September 2008.



S.karunakar Reddy Received his B-tech and M-tech degree in Electronic &and Communication Engineering from JNTU. Hyderabad. Currently working as a Assoc. Professor in Aurora's Technological & Research Institute under JNTU. Hyderabad.



M. Sunder Rao received my B-tech degree in Electronics & Communication Engineering from JNTU. Hyderabad in 2011. Currently a M-tech student in the Department of Embedded System at JNTU. Hyderabad.