

The Design of Video Surveillance System Using S3C2440 and 3G Module

Golluri Venu, G.Anitha, G.Ramchander

Abstract—In this paper, A wireless video surveillance system based on 3G Module and ARM9 is designed. The embedded chip and the programming techniques are adopted. The central monitor which adopts S3C2440 chip as controller is the core of the whole system. First, USB camera video data are collected by the embedded Linux system, processed, compressed and transferred by the processing chip. Then, video data are sent to the monitor client by GRPS network. Tests show the presented wireless video surveillance system is reliable and stable. And it has a perfect application prospects with real-time monitor.

Keywords- Embedded Linux; S3C2440; Monitoring; GPRS

1. INTRODUCTION

Video surveillance systems play an increasingly important role to maintain social security. It has been widely used in many fields, such as finance, public security, banking, and home. Traditional video surveillance can generally achieve close distance monitoring, by using the PC as a monitor host, monitor host connected monitor camera with coaxial cable. With the development of IT technology, especially the rapid development in embedded CPU of the open Linux systems, video conferencing, remote monitoring, and mass data processing fields can apply the embedded technology. This embedded monitoring system to overcome the week points of the traditional video surveillance systems, such as complex structure, poor stability, and expensive cost. Meanwhile, it possess some advantages, for example, hardware and software can be cut, compact construct, portable, low power consumption, more suitable for long-distance transmission. In this system, we use the high-performance S3C2440 chip as the microprocessor. Video data is captured from a USB camera, compressed into JPEG format, transferred the 3G network under the control of the ARM9 chip; then, the monitor client will receive the compressed data frame to restructure, and recompose video images .Wireless video monitor system provide a practical solution for remote wireless monitoring with low cost[1]. The entire system is consisting of the video capture, send and receive of the wireless network, to image data recover. The

video capture is the core part of the system. The camera capture video data, the data compressed, and transferred into a Video stream format, this is pending transfer data. GPRS transceiver module is use for wireless network module[2][3]. Data can be transmitted into the frame format and sent to the been accepted, prepare it in two-column format, including 3G network; data arrival the receiving monitor through high-speed network transmission, this can also be a data center. Host to complete development system program development, debugging, etc.; the system structure is shown in Figure 1. The video signal monitor of the entire system is under the control of S3C2440

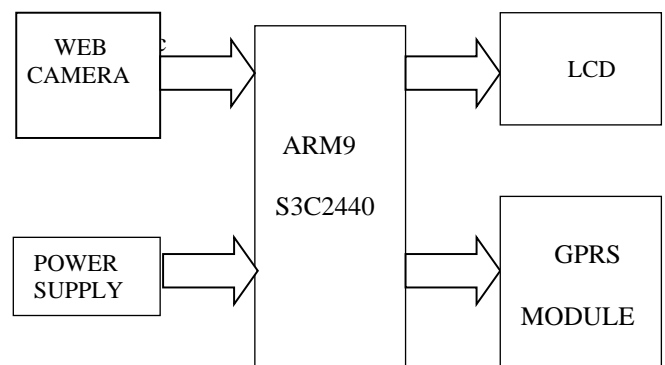


Fig 1. The overall composition of the system

II. THE SYSTEM DESIGN

A. The system hardware design

System hardware structure is shown in Figure 2

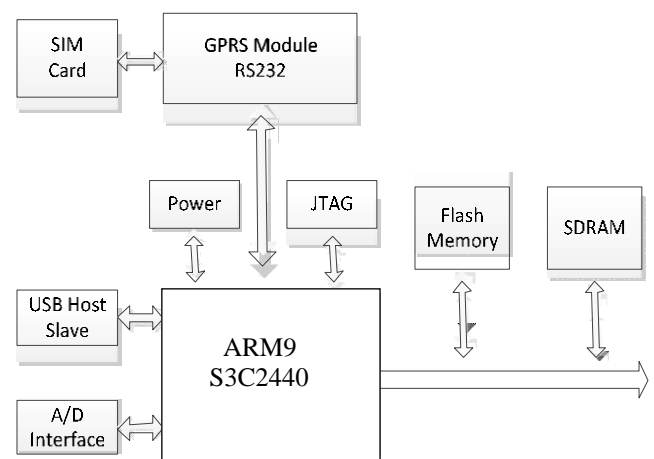


Fig 2. System hardware structure

Manuscript published on 30 October 2013.

* Correspondence Author (s)

Golluri Venu, Ece, Asst Prof.,Christhu Jyothi institute Of Technolgy And Science, Warangal, Andhra Pradesh, India.

G.Anitha, Ece,Asst Prof. Christhu Jyothi Institute Of Technology And Science, Warangal, Andhar Pradesh, India.

G. Ramchander, Ece, Asst Prof.,Chrithu Jyothi Institute Of Technology And Science, Warangal, Andhra Pradesh, India.

© The Authors. Published by Blue Eyes Intelligence Engineering and Sciences Publication (BEIESP). This is an [open access](http://creativecommons.org/licenses/by-nc-nd/4.0/) article under the CC-BY-NC-ND license <http://creativecommons.org/licenses/by-nc-nd/4.0/>

The main processor uses the S3C2440. The S3C2440 offers outstanding features with its CPU core, a16/32 bit ARM920T RISC processor designed by ARM, Ltd. The ARM920T implements MMU, AMBA BUS, and Harvard cache architecture with separate 16kB instruction and 16kB data caches, each with 8-word line length. It has 2-port USB Host and 1-port USB Device, support 64MB Flash and 64 MB SDRAM memory cell. GPRS module connected GPRS network through the RS232 expansion interface, the module with the address locator can also be extended, and system can take full advantage of the 3G network for data transmission. JTAG port is use for system debugging [4].

The design of GPRS Communication Module

In order to connect GPRS module with GPRS wireless communications network, the opened-GPRS-services external SIM card must be in use. SIM_VCC is the mains to which the G20 uses to provide power, SIM_Reset is used by G20 for the SIM card resetting control, SIM Clock is provided to the SIM card for clock, SIM_DIO is for the data transmission connection between the G20 and the SIM card

B. The system software design

1. The image capturing

The video capture is the core of the entire video capture module. The video capture is finished by the V4L (the video4linux) of Embedded Linux operating system and imaging device drivers. V4L is a set of API of the Linux imaging systems and the Linux kernel support embedded images. The OV511 chip with built-in drive is used in the system[5][6]. Driver loading process is as follows: making menu config in the kernel directory, then selecting the Video for Linux usb support directory, and support for usb and usb camera the ov511 support, finally save and exit. The program can be loaded successfully. and the system starts with the new kernel after running the ov511. O

The system functions are as follows:

```
int v4l_open(char *dev, v4l_device *vd){} //Open device
int v4l_get_capability(v4l_device *vd){} //Read information
int v4l_get_picture(v4l_device *vd){} //Capturing images
int v4l_close(v4l_device *vd){} //Close video device
The construction of data defined by the program
typedef struct v4l_struct
{
    int fd;

    struct video_capability capability;
    struct video_channel channel[4];
    struct video_picture picture; struct
    video_window window; struct
    video_capture capture; struct
    video_buffer buffer;

    struct video_mmap
    mmap; struct
    video_mbuf mbuf;
    unsigned char *map;

    int frame;

    int framestat[2];
```

```
}vd;
```

The external device is managed by the device file in the Linux operating system. So, the operations of external devices are transformed into the operation of the device files. The main process is shown in Figure 3

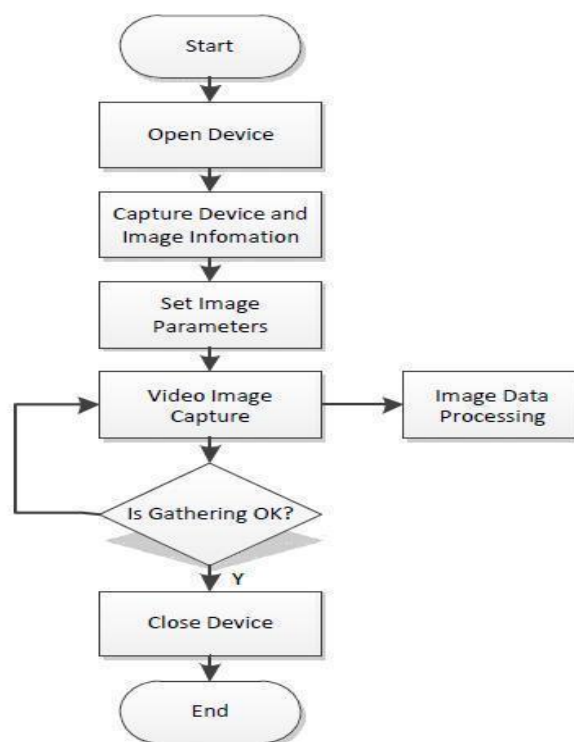


Fig 3. The image capturing process

After the above steps, the camera video data is acquisited into memory. First the video data of collection is compressed into JPEG. Then the data is packaged into the data packet transmission to the video application server for processing.

The capture function is as follows:
Loop :

```
{
    If (framestat[0] = 0)
        VIDIOCMCAPTURE(); //start
    if ( grab_fd,VIDIOCSYNC,&(vd->frame_current))
        framestat[1] = 1;
    if ( framestat[1] = 1 )
        VIDIOCMCAPTURE();
    if(grab_fd,VIDIOCSYNC,&(vd->frame_current))
        framestat[0] =0; //wait until the complete
}
```

2. Video Compression

The JPEG compression encoding scheme is used in the system based on the DCT (Discrete Cosine Transform) sequence. The original image is divided into 8 × 8 small video image after DCT transform.

The low frequency components are concentrated in the upper left corner, and high-frequency components are in the lower right corner (DCT transform is actually a low-pass filter of the spatial domain) [7].

The high spatial frequency for the human eye is far from low-frequency sensitive, so the treatment of visual loss is very small. The video image is clear and true by the method. The data is quantified after DCT. The quantization is a value divided by the quantization table corresponding value. Due to the smaller value of the upper-left corner of the quantization table, the value of the upper right corner on the upper left corner of the low frequency components can be maintained, high-frequency component can be fully inhibited. After Quantified at high spatial frequencies a large numbers of consecutive zero will appear. The coding process is used. The system uses Huffman variable length coding to encode the quantized DCT coefficients, so that the entropy is minimum. The JPEG encoding under Linux can use the library Libjpeg, that is used under linux standard library, and its function is the picture in accordance with a certain percentage of compressed into JPEG format pictures or JPEG format to decompress. Many software operating on JPEG are based on this library.

3. The program design

System programming is divided into client and server programming. The client program realizes the data collection and sending and server program completes the data reception and decoding, display. The socket programming is used.

The socket is the interface of the same host between the application layer and transport layer. TCP or UDP may be used to provide concurrent services for more than one application process while communicating between application layer and transport layer[8][9]. In order to distinguish between different application processes and connections, many computer operating systems provide for the application by TCP / IP protocol. The interact is called the socket. Its main parameters are three: the destination IP address, transport layer protocol and port number. These three parameters can be distinguished from a different application process or a network connection communication, concurrent services of data transmission. The system uses the stream socket design whose communication process is shown in Figure 4.

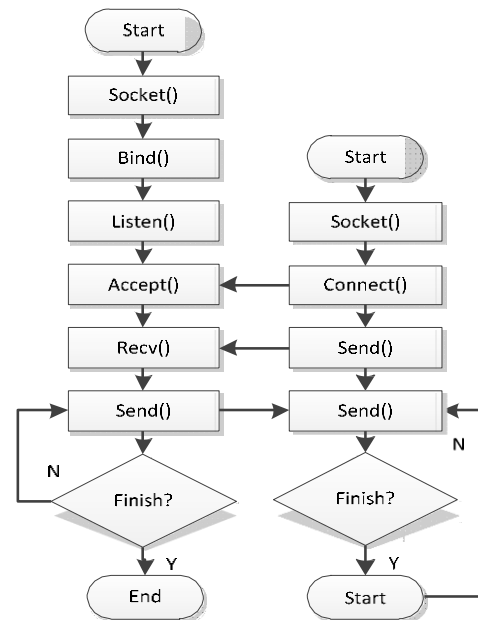


Fig 4. The TCP communication process between C/S

I. Server program design

- 1) Create a socket: sockkh (AF_INET, SOCK- STREAM, 0); // This function is used to create a new socket, AF_INET for specified Internet address; SOCK-STREAM specify a stream socket; 0 means the default protocol.
- 2) Bind function: the bind (sockkh, (struct the sockaddr*) & The server_addr, sizeof (struct the sockaddr)); // Associated the physical location of the socket returned by socket port on the network. Sockkh parameters are socket descriptor returned; local address; the length of the socket address structure.
- 3) Listen (sockkh, 20); // Socket port accepts the connection request is sent from the client. The maximum number of clients is 20.
- 4) Int accept (sockkh, struct sockaddr * addr, int * addrlen); // receiving connection requests from CLIENT. addr is used to store client information. Filling Address and port when connected to the client; addrlen length numbers of client address bytes.

II. Client program design

First its IP address is obtained by the domain name server, and then a socket is created. The data is received from the server after connection is successful by connect function; finally the socket is close after the communication.is finished.

```

sock_client=socket(AF_NET,SOCK_STREAM,IPPROTO
_TCP); //Open socket

```

```

initaddr(&servadr,address,port); // Initializat address

```

```

connet(client_handle,(struct sockaddr

```

```

*) &servadr,sizeof(struct sockaddr)); // Establish a connection

```

III.EXPERIMENTAL RESULTS AND ANALYSIS

The system running is shown in Figure 5.

The image size captured and compressed is 200 * 150 and an average of 20kB in the system. The image is clearer and the monitoring point is reflected. With the information image-level changes of collection, data acquisition frame rate is around 5. The running is stable. After the compression size is adjusted, frame rate can be improved, but not smooth.

```

root@localhost:~#
NET4: Unix domain sockets 1.0/SMP for Linux NET4.0.
NetWinder Floating Point Emulator V0.95 (c) 1998-1999 Rebel.com
RAMDISK: Couldn't find valid RAM disk image starting at 0.
Freeing initrd memory: 3072K
Looking up port of RPC 100003/2 on 192.168.0.39
hub.c: USB new device connect on bus1/1, assigned device number 2
usb.c: USB device 2 (vend/prod 0x5a9/0xa511) is not claimed by any active driver.
Looking up port of RPC 100005/1 on 192.168.0.39
VFS: Mounted root (nfs filesystem).
Mounted devfs on /dev
Freeing init memory: 80K
console=/dev/co?sole
init started: BusyBox v0.60.3 (2002.05.13-08:36+0000) multi-c?ll binary
Starting pid 11, console /dev/console: '/etc/init.d/rcS'
Bummer, could not run '/etc/init.d/rcS': No such file or directory
Waiting for enter to start '/bin/sh' (pid 13, terminal /dev/console)

Please press Enter to activate this console. ^[OM
Starting pid 13, console /dev/console: '/bin/sh'

BusyBox v0.60.3 (2002.05.13-08:36+0000) Built-in shell (ash)
Enter 'help' for a list of built-in commands.
    
```

Fig 5. The system running

The result is obtained from the experimental data.

1. The transmission rate is instability; with the changes of the network.
2. The image is not smooth due to USB driver. The part performance is raised by the improvement of USB.
3. The system is stable.

The monitoring window is shown in Figure 6

IV. CONCLUSIONS

The video surveillance system is designed based on ARM9 embedded processor and embedded Linux in the paper. The system adopts the higher standard treatment of JPEG compression. The test system is stable, reliable, clearer performance on video; to meet the monitoring requirements under the difficult, complex environment the research provides a novel application scheme with the wide application prospect for the monitor of the remote environment.



Fig 6. The monitoring

REFERENCES

- [1] Zhong Chongquan. Research and application of remote monitoring based on embedded web. Chinese Journal of Scientific Instrument, Vol.27, pp.575-577, June.2006
- [2] Jinxue Zhan, Ming Zhang. Research and design of embedded tank car monitoring system based on ARM9. 2009 International Symposium on Computational Intelligence and Design. ISCID 2009. 2009,(2):292-295
- [3] Peng Daogang, Zhang Hao, Zhang Kai, Li Hui. Research and development of the remote I/O data acquisition system based on embedded ARM platform. 2009 International Conference on Electronic Computer Technology ICECT 2009,341-344
- [4] Dong Xie.ARM-based embedded remote monitoring and control system design Gateway. Modern electronic technology, 2006, pp. 85-88
- [5] Chen Rongfang, Wang Haibin, Hu Zhenhua, Zheng Pin. Remote Monitor System of Power Network Based on Embedded Internet Technology. Microcomputer Information. vol. 24, no. 7-2, pp. 78-80,2008
- [6] Zhang YuJie, Huai Weihua, Tian Ze. Design of Industrial Temperature Monitoring Equipment Based on Embedded Linux Operating System. Computer Engineering and Design, vol. 30, no. 17, pp. 3940-3942, 2009.



First Author name, GOLLURI VENU, ASSISTANT PROFESSOR, ECE, CHRISTHU JYOTHI INSTITUTE OF TECHNOLOGY AND SCIENCE,WARANGAL,ANDHARPRADESH,INDIA CELLNO:+919908183968 ,venu413@gmail.com



Second Author name, G.ANITHA, ASSISTANT PROFESSOR,ECE, CHRISTHU JYOTHI INSTITUTE OF TECHNOLOGY AND SCIENCE, WARANGAL, ANDHARPRADESH, INDIA CELL NO.+917893222539 ,anithaguguloth@gmail.com



Third Author name, G.RAMCHANDER, ASSISTANT PROFESSOR, ECE, CHRITHU JYOTHI INSTITUTE OF TECHNOLOGY AND SCIENCE,WARANGAL,ANDHARPRADESH,INDIA CELLNO.+91-9640034572, ramchanderdsp@gmail.com