www.s2is.org
International Journal on Smart Sensing and Intelligent Systems
ISSN 1178-5608

HOME REAL-TIME REMOTE MONITORING SYSTEM BASED ON EMBEDDED LINUX

Huang Feijiang[1*,4], Li Zhaofeng[2], Lu Xiaochun[3,4], Liu Guangcan[1], Sun Liping[1], Wang Yingde[1], Chen Weibing[1], Xiao Xinle[1]

[1*] Department of Electronics and Communication Engineering

Changsha University, Changsha 410022, China

[2] College of Information Engineering

Henan Institute of Science and Technology, Xinxiang 453003, China

[3] National Time Service Center, Chinese Academy of Sciences, Xi'an 710600, China

[4] Key Laboratory of Precision Navigation and Timing Technology

Chinese Academy of Sciences, Xi'an 710600, China

Emails: ccsuhfj@163.com

Submitted: Aug.8,2012 ***Accepted: Sep.15,2012*** ***Published: Dec.1,2012***

Abstract- Remote video monitoring is an important way of residential security. Due to some deficiencies of traditional home monitoring system such as high cost, large power consumption and difficult maintenance, this paper designs an economical and practical home real-time remote monitoring system. This system includes hardware design and software design. The hardware analyzes and compares the processor of different architectures to configure the model of microprocessors and the composition of peripheral circuits, and then identifies the S3C2440 chip as the system's main

controller. For the software part, this paper first studies the construction of embedded Linux system containing bootstrapper, kernel compile, files preparation system and transplantation; then it focuses on a detailed analysis of video capture module under Linux to present the main process of video capture under Linux when the USB camera is selected and design the video capture program based on the Video4Linux2 Linux programming interface; finally, it designs the B/S architecture-based video sending program for the video output module and achieves the real-time browse on a PC through a browser, thereby realizing the remote monitoring for residence. This system relies on the mature embedded technology and achieves the embedded Linux-based real-time remote monitoring system on the Mini2440 development board in accordance with the above design. Compared with the traditional monitoring system, this design has a lower cost and higher cost performance, so it is more practical for the home security.

Index terms*: Family security facilities, remote monitoring, embedded Linux, ARM, video capture.*

I. INTRODUCTION

With the general increase in living standards, the stronger safety awareness for the personal property brings about the growing demand for family security facilities. The video image is a kind of intuitive and specific information expression for the objective things. With the rapid development of networking, communications and microelectronics technology, coupled with the increase in people's material life level, the video image obtains rapid development in information access and command and dispatch due to its advantages such as intuition, convenience and rich content, etc[1]. Nowadays, the video monitoring based on the image acquisition has become a major residential security facility. There are various types of video monitoring, for example, to acquire a single image, continuous pictures and video streams. Transmission modes include the Internet, GSM module, GPRS MMS and local area wireless network, etc.; sending modes includes the real-time transmission to the network, saving to the local disk storage, etc[2-5].

The traditional security products which exist in the market are always facing the public places, although the smart home residential security module has been done very well, its high cost still makes it fail to meet the need of majority of ordinary families. Therefore, it is necessary to develop a kind of residential security monitoring system for home use, which enables the ordinary family to be able to master the real-time situation of residence and can effectively reduce the property loss. Combined with the development trend of embedded technology[6,7], this paper designs an embedded Linux real-time remote monitoring system for home security. The system adopts the S3C2440 microprocessor and the B/S architecture to send the video information captured by the USB camera to the network, and then home users can view the residential situation using the browser on a PC which has been connected to the network. Thereby, the real-time remote monitoring of families can be achieved.

II. THE SCHEME DESIGN OF HOME REMOTE MONITORING SYSTEM

a. Demand Analysis

(1) Disadvantages of Traditional Monitoring Systems. The traditional monitoring system is composed by the front-end monitor, the relay transmission and the background control display terminal, as shown in Figure 1. Among the three parts, the front-end monitor is responsible for the acquisition of the video image, while the relay transmission part connects the front-end monitor and background control display terminal to achieve the information interaction by cable, fiber optic or radio waves. The background control display terminal is the key part of the system. It mainly processes the video information transmitted through the relay device, takes charge of the front-end monitor, displays the collected video image, and save the sound and video information to local disk.

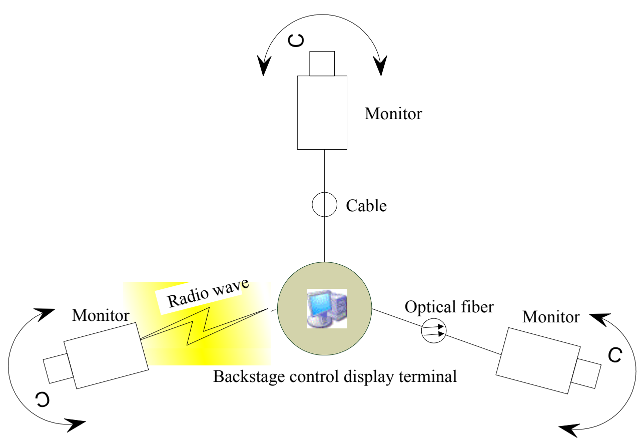

Figure 1. Structure of traditional monitoring system

Traditional monitoring system is mainly used in traffic monitoring, industrial monitoring and security surveillance in public places, and its main deficiency is mainly manifested in the following aspects: the large amount of video information, taking up too much transmission bandwidth, suitable for LAN but not conducive to long-distance network transmission; complex system structure, multi-device assembly, high cost, separation of the video capture module and the video processing module and not favorable for management; using high-performance processing device, large power consumption in operation and high maintenance costs; fixed monitoring terminal, which requires the management of special operations personnel.

(2) Demand of Home Users on Monitoring System. After the analysis of information on home security and video surveillance, monitoring system suitable for home security should have the following characteristics:

1) The location where the monitoring terminal is used is not fixed;

2) Customers have limited purchasing power of monitoring equipment and certain requirements for the operating power consumption of the monitoring equipment;

3) The system is with strong real-time work ability and high stability and can work steadily;

4) The equipment enjoys a high degree of intelligence and streamline operations;

5) The server has little redundancy and higher processing capacity.

Consequently, home security monitoring system should be designed to meet the requirements that users can view the residence in real-time and from anywhere, plus low equipment cost and operating cost; the system is able to ensure higher quality and clarity of the video image in lower transmission bandwidth; the operation is simple and easy to maintain; the system is compact with small size and easy to install and use; the system is with high stability and can work for a long time.

b. Overall Design Scheme of Monitoring System

Through the analysis of the disadvantages of traditional monitoring systems and the demand of home users on remote monitoring system, the embedded system become the first choice for its advantage of efficient processing power, ultra-low running power consumption and comprehensive technical support. As the embedded Linux is compatible with a variety of CPU and is characterized by stable performance, a kernel with compact structure, free source code on the opening up and development tools, plus it is inexpensive, powerful and easy to transplant, it is widely used by many businesses[8]. The system is divided into two modules-video capture and video output, using the most common USB camera to capture video. After the collected raw image data are processed, they are sent to the network by the B/S architecture. Customers can realize the surveillance of video through a browser. The overall design scheme of the system is as shown in Figure 2.

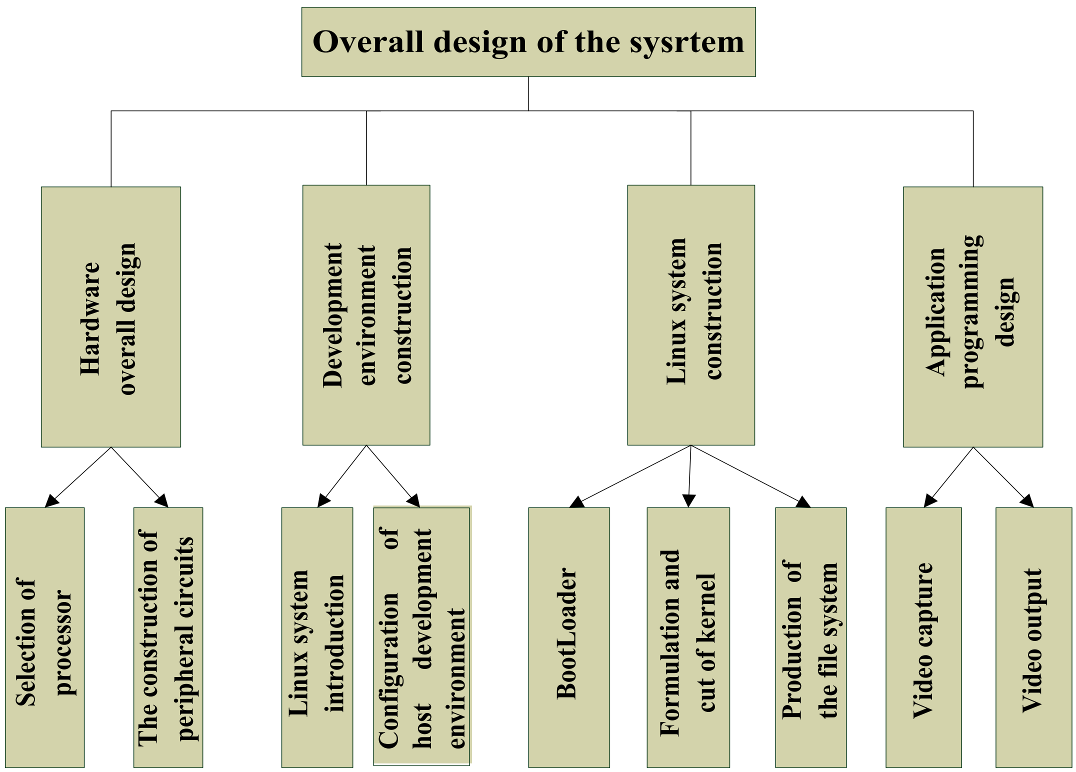

Figure 2. Overall design scheme of home remote real-time monitoring system

Figure 2 shows that the design of the system is divided into four parts: the overall design of the hardware, the establishment of the development environment, Linux system construction and application design. The overall design of the hardware includes the selection of processor and the construction of peripheral circuits, while the construction of Linux development environment involves in the selection of embedded Linux operating system and the configuration of host development environment. The embedded Linux system consists of the BootLoader, the formulation and cut of kernel, and the production and transplantation of the file system, while the design of the application is divided into two modules-video capture and video output.

III. OVERALL DESIGN OF THE SYSTEM HARDWARE BASED ON ARM PROCESSOR

After comparison and research, the S3C2440 microprocessor of Samsung is regarded as the main chip for video capture and processing. S3C2440 is a 16/32 bit RISC embedded microprocessor based on the ARM920T core introduced by Samsung and its maximum operating frequency is up to 400 MHz, which is mainly applied in the handheld equipment as well as the cost-effective and low-power field. The chip implements MMU, AMBA, BUS, and Harvard cache architecture[9]. The provided comprehensive and universal chip peripherals reduce the cost of the system to a minimum level without configuring additional components. The block diagram of the system hardware is as shown in Figure 3[10].

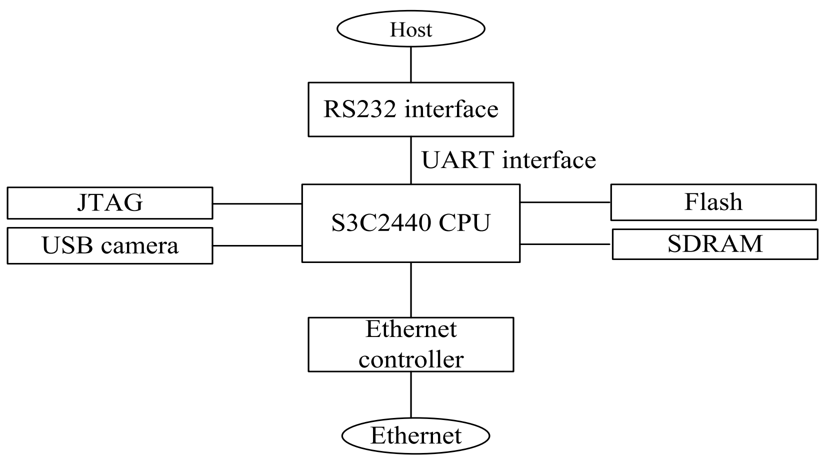

Figure 3. Block diagram of the monitoring system hardware

The system hardware is composed by seven major components- CPU, NandFlash NorFlash SDRAM memory, serial ports, USB camera, JTAG interface, and DM9000 NIC. Samsung S3C2440 microprocessor is used as the central controller and video image processing center, and the USB camera connected to the CPU is responsible for the capture of video images and sends the collected video image information to the central controller for format conversion and compression processing. 2M NorFlash saves a complete supervivi, and when the bootloader in NandFlash and system crash, we can start from NorFlash to enter download mode, copying the supervivi in NorFlash to NandFlash for reprograming the kernel and file system. The 256M NandFlash is divided into four partitions, respectively saved to BootLoader guide, partition information, and kernel and file system. The file system starts from the 0x0000000 address of

NandFlash, and after hardware initialization and setting, the kernel in Nandflash and file system image are loaded to SDRAM memory, and then the kernel is called to mount the file system. The DM9000 network controller is used to connect to the network and be responsible for the transceiver between communications and data. In the development stage, the serial port is used to communicate with the host, download information to the development board, and connect the host to establish a console for operating the target machine. JTAG interface is for buring system for bare computer, and when bootloader in NorFlash and NandFlash crashes, this method can be used to restore the system. The real object of the monitoring system is shown in Figure 4.

Figure 4. The real object of the monitoring system

IV. SYSTEM CONSTRUCTION BASED ON EMBEDDED LINUX

a. Development Environment Construction of Linux Software

The development environment for embedded Linux system is composed of two parts: the target and the host. At the early days of the development, developers usually conduct encoding, cross-compiling and debugging in the host. The host is generally universal personal PC with the universal X86 processor produced by Intel (or AMD); while the target (ie development board) adopts the S3C2440 microprocessor with ARM structural system. Due to the different processors, the code written by the compiler of the host cannot run properly on the target machine. Therefore, it is necessary to install the cross-compilation tools on the host to allocate the communication environment with the target machine. Embedded development requires the consistent operating system for the host and target machine, so it is necessary to install a Linux system on the host and build a good development environment for the embedded Linux software. First the network should be configured to ensure that both of them locate in the same network segment, thereby guaranteeing the proper communication; then the NFS and Samba services are added to ensure the file sharing between the virtual Linux host system and the host and save development time; finally, the cross compiler is installed. In this way, the target file could generate the executable code on the host by using arm-linux-cross-compiler, which is then downloaded to the target board to make the system running.

b. Construction of the Embedded Linux System

The embedded Linux system consists of three parts: the boot loader, kernel and file system.

(1) BootLoader Program Design. BootLoader is a small program executed before the running of the operating system kernel or user applications. This small program can initialize the hardware devices and establish the map of memory space in order to bring the system's hardware environment to a proper state, thereby providing right environment for the operating system kernel or user applications. BootLoader works properly in the bootstrap mode[11]. The designing process of bootloader includes two phases: Phase 1 and Phase 2[12]. Phase 1 includes the processing work related to programming using assembly language, CPU core as well as storage devices. In phase 2, C language is used to realize the clock frequency setting, entering the corresponding download mode or the system starting.

(2) Kernel Cut. The kernel of embedded Linux can be customized, for the kernel cut can remove unnecessary modules and make the system's architecture more concise and compact. First, unpack the kernel source code and then enter the kernel directory to modify the parameters of Linux sound code, including the makefile. Then input the clock and Nand flash partition information on the platform to enter the kernel home directory. Configure and compile the Linux kernel to generate kernel image file in kernel directory.

(3) File System Customization. Develop a minimal file system according to the requirements of the system, which can save storage space and reduce redundancy. First, build the root file system and then create a directory structure for the Linux root file system, as shown in Figure 5, where

/dev is the mount point of device file system, / bin /sbin /usr /bin /usr /sbin is the directory that executive commands exist, /lib /usr /lib is a dynamic link library file, /etc is used to place initialization script and configuration file, /proc mounts the virtual file system, and /sys mounts the sysfs file system. Second, develop the image file for the root file system. Finally, download the above kernel files and their image files to Nand Flash partition to complete the transplantation of kernel and file system.

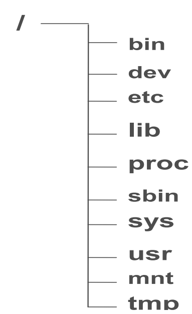

Figure 5. The structure of the root directory

V. APPLICATION DESIGN AND SYSTEM DEBUGGING

a. Overall Design of Application Program

The system application mainly constitutes two function modules-video capture and video output, as shown in Figure 6.

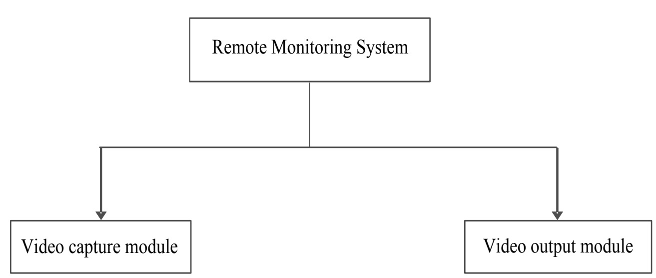

Figure 6. Functional block diagram of applications

Video capture module collects original image data using USB camera and configure parameters captured by USB camera such as the picture format, the size of the cache, and the number of frames through the Video4Linux function in the embedded Linux, and then frame by frame pictures are obtained by memory mapping, thus forming a continuous video. Video output module is based on B/S architecture, which sends the collected video to Internet through socket programming. Users can view the surveillance video with a browser of PC.

b. Design of Video Capture Module

Video4Linux is the kernel driver of Linux video equipment, which offers a range of interface functions for application programming of TV card, video capture card and USB video cameras and also provides data interface for radio communications and Teletext decoding and vertical blanking[13]. The work flow of USB camera based on Video4Linux is shown in Figure 7[14].

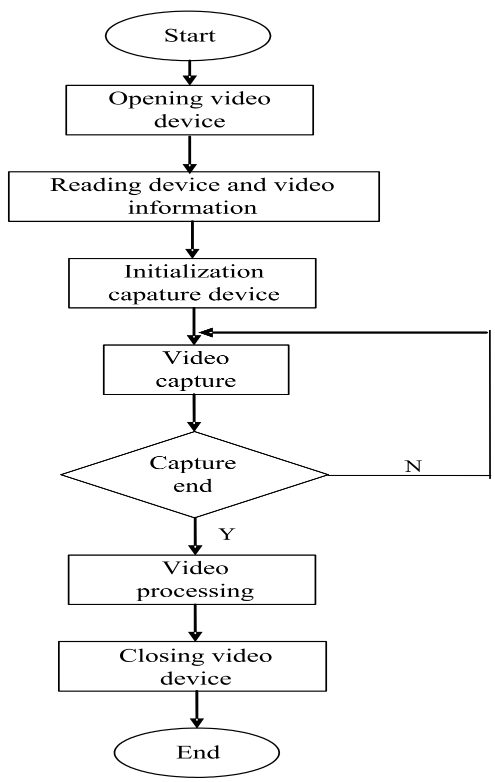

Figure 7. Work flow of video capture

When USB camera is connected with the development board, a file named video0 will be automatically generated in the /dev directory. First open USB camera and directly use open function to operate /dev/video files; read the current device information and video capture configuration information, such as the camera chip, sampling format and method, the acquisition frequency information; next, set parameters of samples; finally, initialize the set parameters of the camera to start the cycle of video capturing and determine whether to continue the capturing. If the judgment result is 0, then turn off the camera and release related resources to end the video capturing process.

The video capturing program relates to how the data is taken out from the camera. First, it is to encapsulate a structure to describe some information of the camera such as the width and height of the captured picture, image format, and so on. The structure is shown as follows:

```
struct vdIn {
        int fd;  // file handle
        char *videodevice;  //device type
        char *status;      //device status
        char *pictName;  // device name
        struct v4l2_capability cap;  // performance setting
        struct v4l2_format fmt;   // format setting
        struct v4l2_buffer buf;   // buffer
        struct v4l2_requestbuffers rb;   // application buffer
        void *mem[NB_BUFFER];    // memory pointers
        unsigned char *tmpbuffer;    // temporary buffer pointers
        unsigned char *framebuffer;    // structure buffer pointers
        int isstreaming;    // output stream
        int grabmethod;    // capture mode
        int width;      // output width
        int height;     // output height
        int fps;       // capture frequency
        int formatIn;    // input format
        int formatOut;   // output format
        int framesizeIn;   // size of capture frames
        int signalquit;    //interrupt quiting signal
        int getPict;
        int rawFrameCapture;
        unsigned int fileCounter;
        unsigned int rfsFramesWritten;
        unsigned int rfsBytesWritten
        FILE *captureFile;
        unsigned int framesWritten;
        unsigned int bytesWritten;
        int framecount;    // the number of the capture frames
```

```
        int recordstart;    //start time record
        int recordtime;    //capture time length
    };
```

Then this structure is written to the drive to initialize the camera. This operation is done via ioctl function, and the commands involved include VIDIOC_QUERYCAP, VIDIOC_S_FMT, VIDIOC_S_PARM, VIDIOC_REQBUFS, and VIDIOC_QUERYBUF, and then memory mapping is completed through mmap[14].

Finally, complete the reading of the picture through ioctl function, and the commands involved are VIDIOC_QBUF and VIDIOC_DQBUF. Then write the obtained data to the file to get the image, and the continuous images transmitted over the network become video.

c. Design of Video Output Module

The video output module adopts B/S architecture based on the requirements of home users to remotely view the video monitoring. The development board is regarded as a server; TCP sockets are used in the program, and each connection request is to create a thread alone to communicate with this request[15].

First socket a handle on server side to bind the IP address and the access port (set as 8080 port) of this computer, and then carry out online monitor to determine whether there is a request service (if there is no request, it has been in a blocked state). Then connect to the IP address and port number of the server on server side and then send the request. At this time, the server responses and begins to conduct data interaction with the client side, and then smooth video screen can be displayed on client side. When the end message is sent by the client side, the handle is closed at both ends and the communication terminates.

In order to respond to the requests of multiple clients simultaneously, the multi-threaded programming is used to establish a connection for each request, and each connection is a thread. Multi-thread programming can effectively improve the application responsiveness as well as the structure of the program. Multithreaded programming requires many functions, including pthread_create, pthread_join which are used for thread creation and waiting; pthread_cond_init, pthread_cond_destroy functions are used for thread command initialization and destruction; the initialization of mutex and the destroying of pthread_mutex_init and pthread_mutex_destroy is one of the key technologies in this program. When accessing specific resources, this resource is locked, so other programs cannot call on. It is unlocked after use, in this way the synchronization and integrity of resources can be protected.

Finally, after applications are programmed and debugged, run it and type the following command: http://192.168.1.230:8080 //determined by the specific IP address of the system. At the moment, the real-time surveillance video as shown in Figure 8 will appear in the browser window.

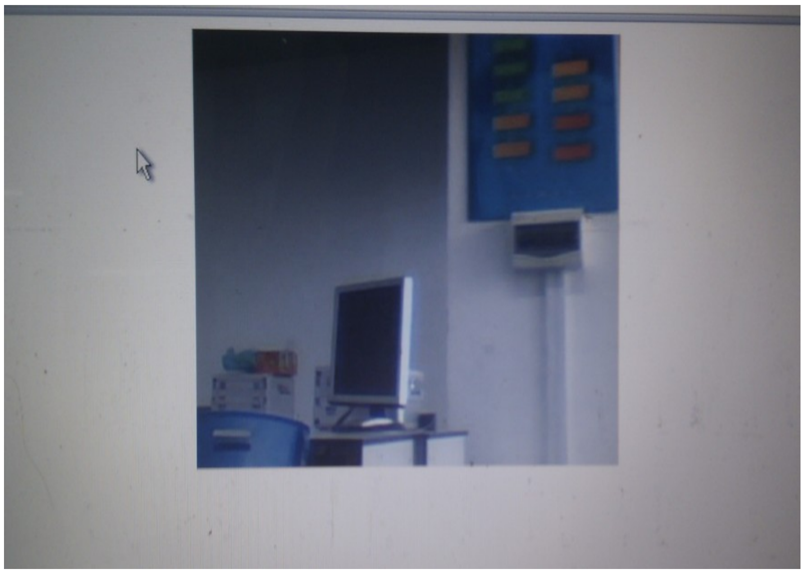

Figure 8. Real-time surveillance video appeared in the browser

VI. CONCLUSIONS

Targeting at the disadvantages of high system cost, great power consumption and difficulty in maintenance of the traditional home monitoring system, a remote monitoring system with the advantage of low cost and high performance price ratio is created from two aspects-the hardware configuration and software design, which is more suitable for family security. The S3C2440 microprocessor with ARM920T chip and Mini2440 development board are selected to achieve the real-time remote monitoring system based on the embedded Linux. Home users can clearly view the real-time residence via a browser. However, the system has not yet achieved the function to save the video. It is required to use the MPEG-4 standard based on JPEG encoding to encode the continuous motion frame image, and the file is saved on the client for video playback.

ACKNOWLEDGMENTS

This work was supported by National Natural Science Foundation of China(No.11073022, No.10673011), Hunan Provincial Natural Science Foundation of China(No. 11JJ3072), Science and Technology Planning Project of Changsha of China (No. K1008012-11), and Open Fund of Key Laboratory of Precision Navigation and Timing Technology, National Time Service Center, CAS(NO.2009PNTT07).

REFERENCES

[1] Huang Li, Qin Jiwu, “Study on Current Situation and Prospect of Embedded Linux”, Computer Knowledge and Technology, Vol. 21, No.15, 2005, pp. 16-18.

[2] Li Gang, Che XiLong, “The Remote Monitoring System Based on GPRS for the Health State of Civil Engineering”, JDCTA: International Journal of Digital Content Technology and its Applications, Vol. 6, No. 8, 2012, pp. 26-33.

[3] Guo Ming, Hou Yanhua, “Current Status and Future of Embedded Linux System”, Journal of Beijing Radio and Television University, No.3, 2003, pp. 27-30.

[4] Li Weimin, Liu Yuansheng, “Smart Controller for Household Appliance”, Electronic & Computer Design World, No.17, 2002, pp. 46-48.

[5] Qin Fangli, “The Research of Remote House Security Monitoring System Based on MMS”, Wuhan University of Technology (Master theses), China, 2010, pp. 5-13.

[6] M. Nadi, C. Margo, M. Kouider, J. Prado, and D. Kourtiche, “Embedded system design and implementation of standard auto-calibrated measurement chain”, International Journal on Smart Sensing and Intelligent Systems , Vol. 1, No. 1, 2008, pp. 21-33.

[7] Wael Benrejeb, Olfa Boubaker, “FPGA Modelling and Real-time Embedded Control Design via Labview Software: Application for Swinging-up a Pendulum”, International Journal on Smart Sensing and Intelligent Systems , Vol. 5, No. 3, 2012, pp. 576-591.

[8] Kang Hongyan, “Design and Realization of Internet of Things Based on Embedded System Used in Intelligent Campus”, IJACT: International Journal of Advancements in Computing Technology, Vol. 3, No. 11, 2011, pp. 291-298.

[9] Wu Wenzhong, Li Wanlei, "Smart Home System Based on ARM and ZigBee", Computer Engineering and Design, Vol. 32, No.6, 2011, pp. 17-20.

[10] Li Bing, Sun Jianping, Tan Yue, Zhang Qiqin, "Network Video Monitor Based on Embedded Linux and S3C2410", Journal of North China Electric Power University(Natural Science Edition), Vol. 33, No.4, 2006, pp. 9-11.

[11] Li Jun, "Transplantation of Bootloader Based on ARM9 Embedded System ", Science Technology and Engineering, Vol. 11, No.32, 2011, pp. 8061-8064.

[12] Zhang Lei, "Research and Design of Video Collection System Based on Embedded Linux", Xidian University (Master theses) , China, 2010, pp. 19-29.

[13] Yu Zhanran , Peng Chuwu , Li Fuhai, Wang Xin, "Wireless Image Monitoring System with Motion Detection", Application of Electronic Technique, Vol. 35, No.9, 2009, pp. 41-43.

[14] Pan Chunpeng, "Research and Design of Video Collection System Based on Embedded Linux", Chongqing University (Master theses), China, 2007, pp. 46-60.

[15] Wang Lei, Wang Zitao, "Performance Optimization of Socket Network Programming Based on Linux", Electronic Design Engineering, Vol. 17, No.9, 2009, pp. 101-103.