International Journal of Multimedia and Ubiquitous Engineering
Vol.11, No.11 (2016), pp.131-144
http://dx.doi.org/10.14257/ijmue.2016.11.11.11

Research of Embedded Remote Video Monitoring System Based on iMX6DL

Zhongfu Tian[1] and Yinglai Huang[2]

[1]*College of Mechanical and Electrical Engineering, Northeast Forestry University, Harbin, China*
[2]*College of Information and Computer Engineering, Northeast Forestry University,*
Tzf7802@163.com, 117968528@qq.com,

Abstract

With the development of information technology, the video surveillance has been widely used in different fields. The real-time video monitoring and controlling system adopts embedded technology, video decoding technology, communication and network technology and some other technologies. Therefore, there is much significance to research it. In this paper the Cortex-A9 is used as core hardware platform to develop the embedded remote video monitoring system, and its key technology is also researched. Based on the research of domestic video monitoring system, and combining with the practical application the overall hardware and software design of the embedded remote video monitoring system is presented. The hardware part of this system includes the microprocessor iMX6DL based on the Cortex-A9 framework which produced by FeiLing company and the USB camera based on the UT_USB_CAMERA module. The software pate includes the transplantation of the embedded operating system Linux, the development of streaming media server, the constructing of website and the writing the payer widget to obtaining a system which has integral real-time monitoring function. The test manifests that the system design is reasonable and stable, it also can be extended to Smart Home, Smart City, Internet of Vehicles and other applications fields.

Keywords*: video monitoring; embedded system; multithread communication; remote monitoring*

1. Introduction

In foreign, the video monitoring system and monochrome television appeared almost simultaneously. The earliest videophone who based on simulation technology is born in Bell Laboratory in the USA [1-2]. Subsequently, a multitude of video monitoring systems and specialized companies who provide solutions to clients also appeared, such as Mir company of Germany, Digital Semiconductor company of America, Picpo company of Canada, MediaCybernetics and so on. Those companies mainly produce monophonic image acquisition cards and compressing cards. Later, some manufacturers, such as LG company of Korea started to develop the multiway monitor system. Its representative product is LDVR2000/3000 series and the packed format is M-JEPG. Its system has 4-6 channels. At the same time, foreign countries have successfully embedded Web camera products, such as the SNC-100P of Korea's LG, the KX-HCM130 of Panasonic, the ANT-NWC10/50/100 of Antwerp and so on [3]. The performance of the above products is good, but their prices are too expensive to accept for internal users.

At present, Embedded video monitoring system has been rapid development under the promotion of some innovative and high technologies and it has broad market application prospect. Therefore, there is great practical significance to research and develop the video monitoring system.

ISSN: 1975-0080 IJMUE

2. The Requirement Analysis and Overall Design of System

2.1. The Requirement Analysis of System

2.1.1. The Functional Requirement

Users of the system are mainly divided into two categories: The administrators and common users [4]. The specific functional requirements of administrators include customer management function, video watching function and remote control function; The specific functional requirement of customer only include video watching function but they can't change the resolution of the video stream, in addition to this they have the same permissions as administrator's [5].

2.1.2. The Performance Requirement

Because this is a multiuser real-time monitoring system, so the requirements to the performance of the system are more stricter. Table 1 describes the performance requirements of the main focus of the system.

Table 1. The Performance Requirements Table of the System

The performance of related items	Description of performance requirements
Number of users what are allowed to login simultaneously	More than 10 users
Frame frequency (640x480)	More than 18frames/second
Setting the resolution time	Less than 1 second
Time of connecting streaming server	Less than 2 seconds
Time of disconnecting streaming server	Less than 1 second
Post-set time	Less than 2 seconds
Time of video button response	Less than 0.5 second
Time of capture button response	Less than 0.2second

2.2. The Overall Design of System

This paper presents an embedded video monitoring system based on embedded technology. The ARM+Linux is used as core construction platform of system and on this basis web server and video server are built. The system collects video image data by video front-end, after the JEPG compression they can be transmitted through the network to the server so the client can accomplish real-time monitoring. On the whole the system adopts B/S architecture. Client can receive monitor video through browser which has JAVA plug-in to realize the aim of monitoring. The general frame of the system is shown in Figure 1 [6].

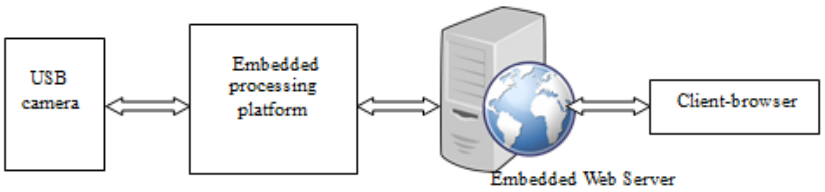


Figure 1. The General Frame Diagram of the System

The whole system is divided into the following four parts:

(1)Camera acquisition front: In the system, UT_USB_CAMERA is used as camera acquisition front, it also has hardware compression function of JPEG video image and mainly to complete video image data acquisition.

(2) Embedded system platform: In the system, embedded platform mainly includes ARM iMX6DL and embedded operating system Linux, the function that video data acquisition and transmission is realized by embedded application software.

(3)Embedded web server: In the system, Boa is selected as embedded web server, it need be configured accurately and transplanted into the whole system. As a compact and efficient web server, Boa supports CGI, is basis of client to monitoring real-time video by browser.

(4)Client: As long as in the same LAN, any authorized client (such as PC and mobile devices) can access and manage the system. The client can receive video data in real time and the data will be displayed on browser.

2.3. The Structure of System's Hardware Platform

The hardware design is consisted by main controller, the devices of video collection and transfer. The hardware structure is shown in Figure 2.

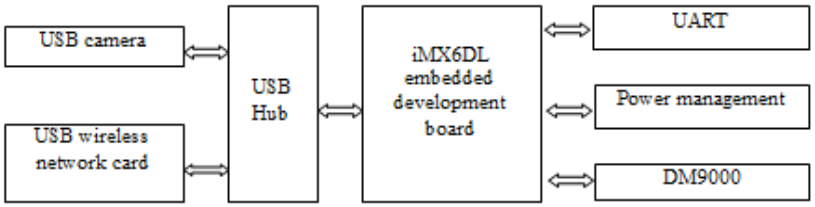


Figure 2. The Hardware Structure of System

The control panel includes core-board and expansion board [7]. The core-board mainly includes iMX6DL micro-processing chip, Nand flash and SDRAM; In expansion board USB Hub is used to circumscribe USB camera and USB wireless net card. Serial DART is used to download and debug procedures. The Feiling company's production iMX6DL is used as embedded development board and the real object of iMX6DL is shown in Figure 3.

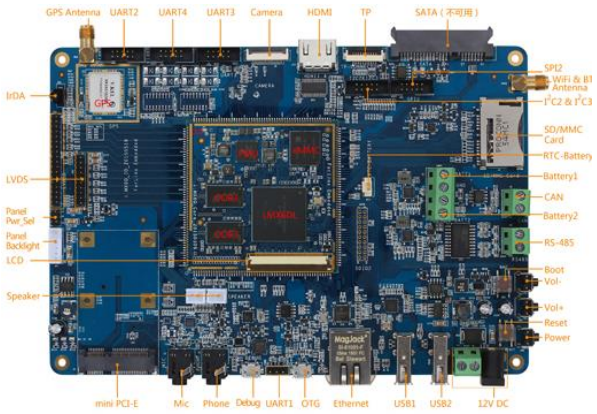


Figure 3. The Picture of Real Object of iMX6DL

2.4. The Software Design Scheme of System

The software design of system includes two aspects which are video capture server-side and video monitoring client. For the PC client, the design of software is mainly the development of upper layer application program and also includes the construction of underlying software development environment [8]. In the early stage of the development

of embedded system, embedded operating system is often not required. In the main function circulation is used to maintain programs' sustaining operation then through the interrupt to handle external request to realize much of functions of the system. But with the functions of embedded system are becoming more and more complex simple bare machine procedures hard to meet the function what the users need. Therefore, it seems very significant to use the embedded operating system to complete resource management, interrupt handling, event handling, task scheduling and other functions.

At present, there are many kinds of embedded operating system at the market and because Linux operating system is free, open source, stable and scalable it is used by most of embedded systems. According to the basic requirements of the system, Linux operating system which structured in embedded platform is selected to complete the development of software. The bottom-up software hierarchy of embedded Linux system is mainly consisted by following several parts: systematic guidance procedure Bootloader, kernel of Linux (include device driver), file system and application program.

2.5. The Construction of Embedded System Environment

Because Windows has a set of mature software and development tools, software structure of embedded side is paid more attention in the aspect of setting up system environment.

2.5.1. The Construction of Cross-compiling Environment

Embedded system needs special cross-development environment and development tools. Therefore, it is necessary to construct the cross-compiling environment in the PC, to generate code that running on embedded devices.

2.5.2. Bootloader

Bootloader is a small program that is executed before the running of operating system. After the initialization of system's hardware and construction of the memory space mapping table, Bootloader realizes the loading of operating [9]. The common kinds of Bootloader mainly include following parts: RedBoot, ARMboot, Vivi, U-Boot, Blob and so on. Because of openness, versatility, strong flexibility and supporting multi-platform of U-Boot, it has been widely used in a variety of platforms. Therefore, U-Boot is selected to boot the system to start. In actually the transplantation of U-Boot is adding related files and configuration options for the development board which is used by the system, then cross-compile is executed, in finally they are burnt to write into Nandflash of the target board.

2.5.3. The Transplantation of Embedded System Kernel and Related Drives

The transplantation of kernel is mainly modifying the codes which are related to hardware platform, according to their own needs users can clip and configurate the kernel, just need add the functions that they need to kernel. Because there are many options of kernel configuration, in general we only need find out the target board which is similar with our development board then using it as the masterplate to complete kernel configuration. In the system the kernel version of Linux is 2.6.28.

(1) The addition of camera driver module

The following are the main types of common USB camera driver:OVCam drivers(ov5xx)、Philips USB Webcam Drvier(pwc), QuickCam USB camera driver(qc-usb), Linux UVC driver(uvc) and so on. In among them UVC drive is suited to the camera devices which conform the specification of USB video category. It includes V4L2 the device drivers of kernel devices and relevant patches of userspace tools. Because the camera which is used in the system complies with UVC specification, it is only need to

compile the UVC drive into kernel. After entering the configuration interface of Linux kernel, the selection as following: Device Drivers->Multimediadevices->Video For Linux->Video capture adapters->V4L USB devices->USB Video Class(UVC) ->UVC input events device support, then compiling the USB camera drive into kernel.

(2) Addition of USB wireless network card drive

Wireless transmission is selected as communication mode in the system, it is realized by TL-WN321 G+ model USB wireless network card which is produced by TP-LINK company in hardware. Because it's the basic function of the system, so it will realize when the wireless network drive is compiled into the kernel directly, and The specific process is as follows: After entering the main configuration interface of Linux kernel selecting Networking support->Networkingoptions->Packet socket in turn, then returning to the Networking options page and selecting Wireless. After entering it[10], selecting the wireless related options. Backing to the upper menu step by step, backing to the home page of kernel configuration, then selecting Device Drivers->Network device support->Wireless LAN->Ralink driver support->Ralink rt2501/rt73 in turn. Selecting the related drive of the device that supports TL-WN321 G+.

2.5.4. The Root File System

The root file system is the first file system which is mounted when kernel is starting. The image files of kernel code are preserved in the root file system. After mounting the root file system, the system boot program will load some initialization scripts and services into internal storage to run. There are many of the file systems supported by embedded Linux system, YAFFS2 is embedded file system that is designed for NAND flash memory. At present there are two versions of file system, they are YAFFS and YAFFS2, one of their main differences is that YAFFS2 can better support NAND FLASH chip that has large capacity. According to the specific needs of embedded devices, YAFFS2 file system is selected by the system.

3. Implementation of System

3.1. The Design of Programs on Server-side

The main function of server-side is using the V4L2 video device programming interface of Linux system to realize the collection of video data [11]. And the video data can be returned to the local display, at the same time monitoring the client's monitoring connection request. If there is a connection request, firstly let the video data carry out the JPEG compression coding through the hardware decoder of iMX6DL processor, then using the JRTPLIB database which is based on RTP/ RTCP protocol to packet package for dataes and sending them to client via wireless network. The overall working process of the server side program is shown in Figure 4.

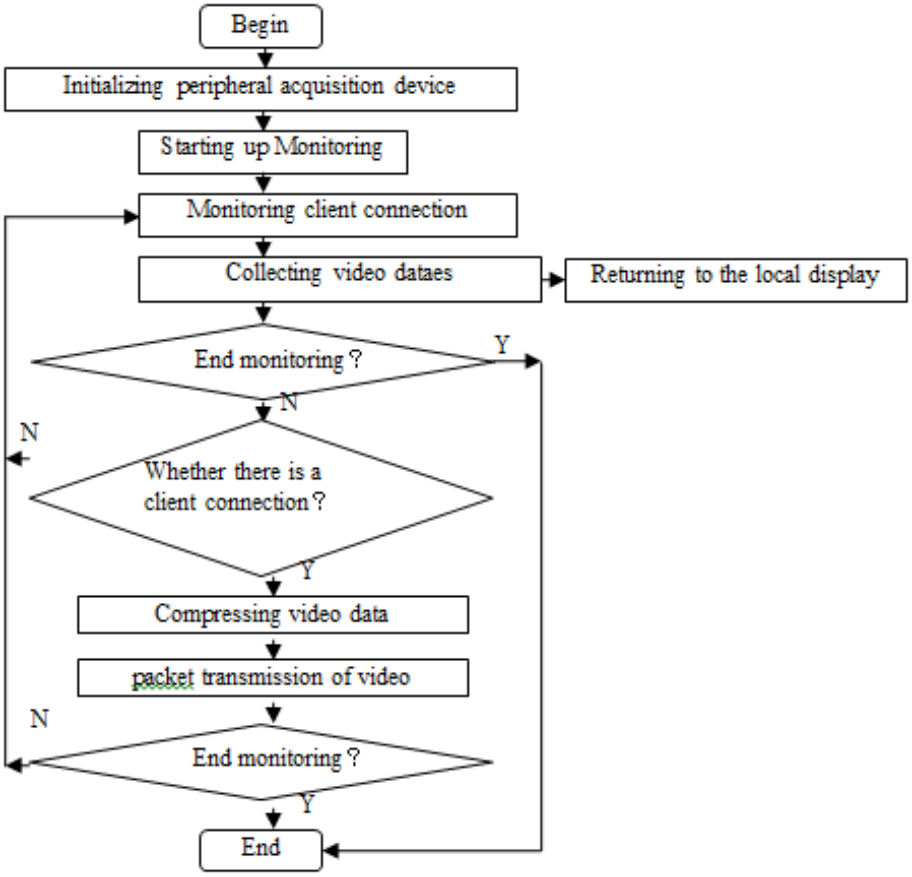


Figure 4. The Overall Working Process of the Server Side

3.1.1. The Multithread Programming of Server Side

The realization of server side is based on the embedded platform, because it is limited by the hardware resource, so it usually requires that the running of applications to use memory and resources as little as possible. But the video server has a number of tasks to be processed, if only one thread is used to handle all tasks serially and because the time and resources occupied by each task are not the same, it will inevitably lead to low efficiency of the system, even it will lead to the interface can't react and appearing the phenomenon of interface's stagnation. It obviously can't meet the requirements of real-time video capture and transmission. So the multithreading mode is used to deal with the work of each functional module to improve performance of overall system. Thus avoiding the phenomenon that the whole procedure is blocked due to the waiting of a link.

3.1.2 Implementation of Qt Multi-thread

According to the requirements of system, using the main thread of program to monitor the client's monitoring connection request and let video can be returned to the local display. Two sub threads are created in the main thread, they are used to respectively complete the compression and collection of video data and transmission of video data.

When using Qt to write multi-threaded program, only need subclass QThread and realize the virtual function run() of QThread class. In the implementation of the program, the subclasses of acquisition coding thread and data sending thread are created, and the two subclasses are instantiated in the main thread, then they call their start() member functions to start the sub-threads, after the starting of threads the codes of run() will be

executed, so the acquisition coding and the codes about video sending need be implemented in their run() functions.

3.2. Software Development of Acquisition Module

In video data acquisition module, Linux kernel calls camera driver program to control USB camera to carry on video data acquisition and compression coding, then the video data in JPEG format will be transmitted to next module.

The video acquisition of the system is implemented based on V4L2. In the embedded Linux operating system, the main process of video data acquisition based on API V4L2 is shown in Figure 5.

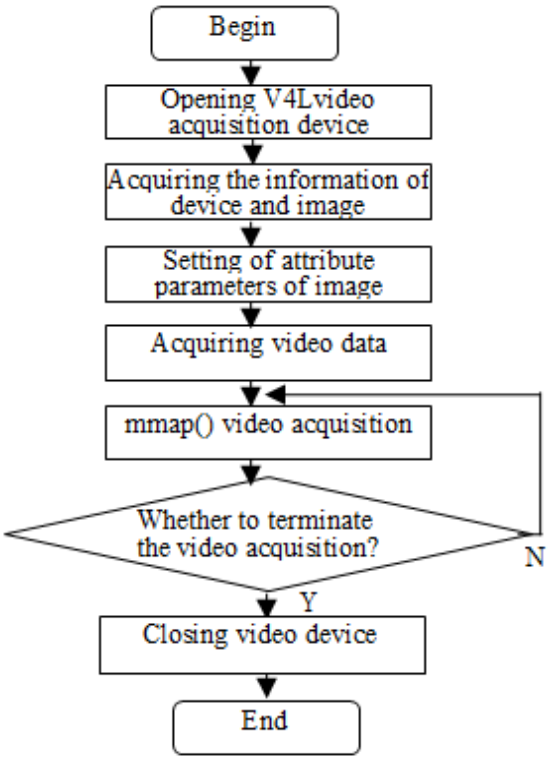


Figure 5. The Process of Video Data Acquisition Based on API V4L2

3.3. The Compression Coding of Video Data

Because of the demand for images and bandwidth of the real-time video monitoring is not high, and the implement of supervision terminal is simple so JPEG compression technique is selected by the system.

3.4. The Implementation of Video Transmission Module

Cross-compiling JRTPLIB, and transplanting it to the embedded platform to complete video transmission based on RTP/RTCP protocol.

(1) Creating the RTP session

Before using JRTPLIB to transmit data it needs to define the instance of the class and calls the Create () method to initialize it, it mainly to complete the setup of the sending port and the sending time interval. Then through the instance to call a member function of RTPSession class to implement addition, deletion of the target address, data transmission and some other operation.

(2) The addition and deletion of the address of data receiver

Before sending data it needs call int AddDestination(const RTPAddress &addr) that the function of RTPSession class to add the specified data receiver address to the list. The RTP protocol allows multiple destination addresses in a single session, when it doesn't need to send data to an address calling int DeleteDestination(const RTPAddress &addr) to delete the specified address, and the function void C1earDestinations() clearing all of addresses in the list.

(3) Sending data to destination address

After the address of the received data is specified, using int SendPacket(const void *data, size_t len, uint8_t pt, bool mark, uint32_t timestampinc) that the function of RTPSession class to send data packets to all of destination addresses.

(4) Closing the RTP session

After calling void BYEDestroy(const RTPTime &maxwaittime,const void *reason,size_treasonlength) function to send BYE group, then leaving session.

3.5. Program Design of Client-side

3.5.1. The Over Design Scheme of Client-side

The main functions of the system include video data receiving, decoding and display, after client program starting, firstly it needs establish a connection with server to inform server to get monitoring., then it receives video data through wireless network and JPEG decoding display them. The overall working process of the client program is shown in Figure 6.

3.5.2. Implementation of Video Data Reception

Client-side still uses the relevant interface functions which provided by JRTPLIB bank to implement data's reception. According to the setting which the transmitting terminal to data packet we can know that the same video frame of RTP packets have some timestamp, receiving end can accord the timestamp of data packet to judge whether it have received a frame of data. The receiving end receives a RTP packet and sets its time stamp to the current timestamp, after the next packet is received to determine whether the time stamp is consistent with the current time stamp if they are not same it indicates video data have been received, copy all the data which is received currently into the circle buffer and continue with the subsequent operation. The main process of the client receiving data is shown in Figure 7.

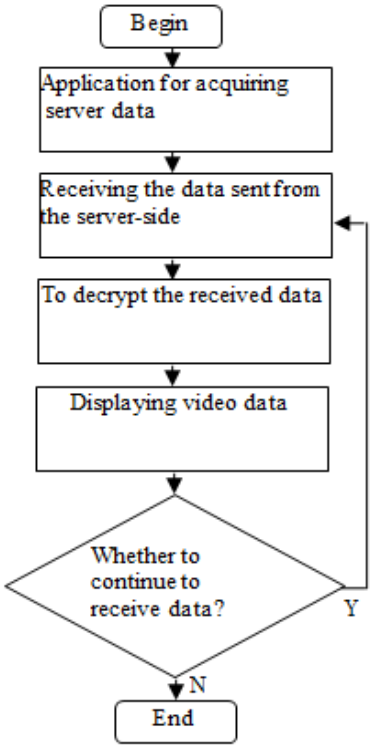


Figure 6. The Overall Working Process Receive Data

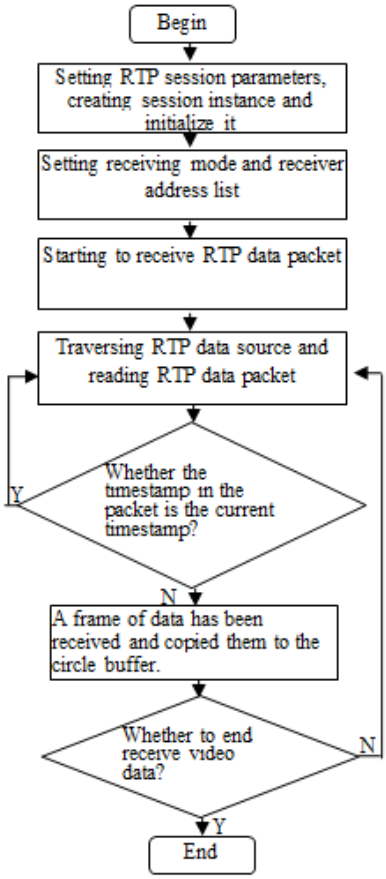


Figure 7. Main Processes for the Client to of Client Program

3.5.3. Display of Video Data

The client also uses QImage class of QT to complete the video display. After the decoding thread have decoded one frame data it will send display signal to main thread and call updata() function in corresponding slot function of main thread[12], then the system will call paintEvent() function to complete refresh display of screen. Because the decoded video data is YCbCr4:2:0 format, so it need be transferred to RGB format. The processing flow is shown in Figure 8.

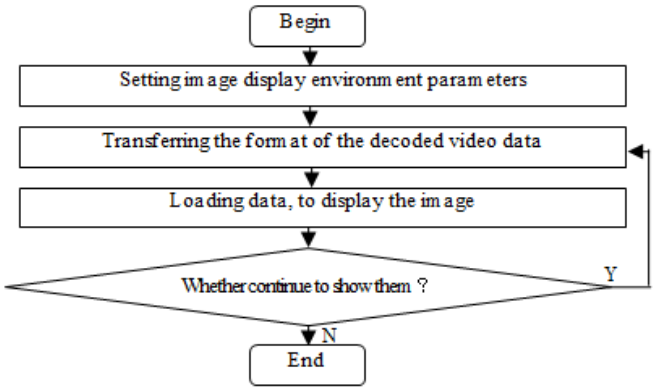


Figure 8. The Process of Client Video Data

4. System Testing

4.1. The Construction of Testing Environment

After the system have completed customization of actual software and hardware, that is, the customized tailoring and transplantation in the ARM platform is completed, through testing to verify the performance of the monitoring system.

The system testing is completed within the area of WIFI signal coverage, using the USB camera to capture video on development board and accessing the development board to the wireless network through USB wireless network card. Because client plays video in the Linux FireFox browser it needs to install the java plugin JDK. After installing it enter the http://192.168.1.20:888/webcam/index-sample.html in the browser, then it can realize realtime paly of video. The overall hardware in the system is shown in Figure 9.

4.2. The Test Results

At first, starting PCLinux host computer and running target board embedded Linux system, then entering the /etc/boa directory of the root file system of target board. Starting web boa server and running the video capture program servfox in /etc directory, the running results are shown in Figure 10.

After starting the video server, client web browser sends connecting request to set up network communication to transfer video data. The information of client connecting the server is shown as Figure 11.

Figure. 9 The Hardware in the System

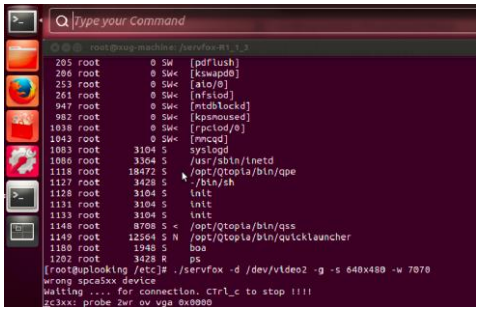


Figure 10. The Running Results of Video Capture Program

After capturing video data successfully, the information of images whose resolution ratio is 640*480 can be watch in the client web browser. The video monitor image is shown in Figure 12.

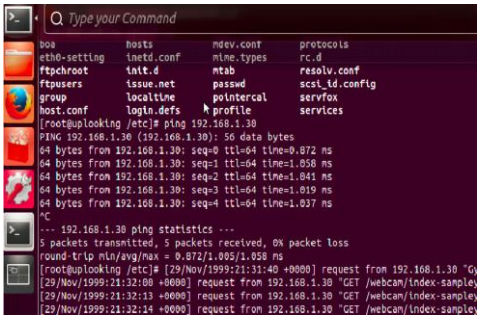


Figure 11. The Information of Connecting Video Server

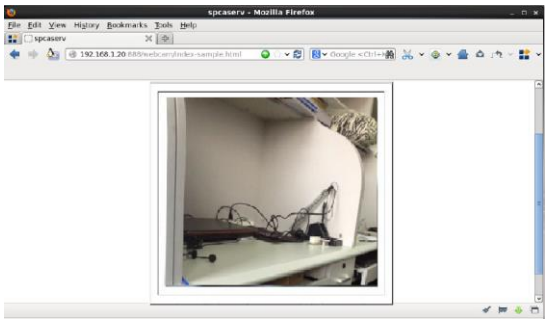


Figure 12. The Video Monitor Image

5. Conclusion

The test shows that the system can meet the requirements of video capture engineering design. But because of using wireless network transmission, distance between acquisition end and receiving end within 60 m the frame loss rate is less and video screen is clear. However, with the distance of each side gradually increased, frame loss rate increased gradually. But in the wireless network coverage, the requirements of video monitor can be met. If it needs to expand the scope of monitoring, it is necessary to add the relay node AP for video transmission.

If the resolution ratio of image capture of video monitoring system is set to 640*480, the decoding display of browser client only can reach 15fps and video display is not very smooth, therefore, the system needs to be further optimized and adjusted on the basis of this setting.

The system realizes video capture and video transmission on the basis of iMX6DL development board, USB camera, USB wireless network card and Linux operating system. The test shows that the whole system has the advantages of high stability, easy to install, low cost and so on. It can be extended for industrial control, residential video monitoring system and intelligent home systems and other fields, it has broad application prospects.

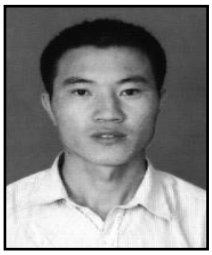
Acknowledgements

The paper is supported by the Fundamental Research Funds for the Central Universities (DL12BB01) and the natural science foundation of Heilongjiang Province (C201244).

References

[1] T. T. Li, J. Ren and X. C. Tang, "Secure wireless monitoring and control systems for intelligent grid and intelligent home", vol. 19, no. 3, **(2012)**, pp. 66-73.
[2] L. Y. Wang and Q. Liu, "Design and Realization of Smart Home Control System Based on S3C2440", vol. 219, no. 6, **(2011)**, pp. 1271-1275.
[3] K. Xiao, H. Li and X. L. Fu, "RTC Driver Development in Embedded Linux", vol. 171, no. 12, **(2011)**, pp. 791-794.
[4] Z. Zhanfeng, Z. Yuntao and Z. Zhiquan, "Design One Kind of Simplified Smart Home Monitor and Controller Based on ARM-Linux System and Information Network to the Public", vol. 26, no. 4, **(2013)**, pp. 825-827.
[5] J. Wang, J. Liu and H. Wang, "Design and Implementation of a Home Automation System Based on ZigBee and 3G Terminal", Applied Mechanics and Mechatronics Automation, vol. 19, no. 9, **(2012)**, pp. 1453-1457.
[6] J. Nightigale, W. Qi and C. Grecos, "Optimized transmission of H.264 scalable video streams over multiple paths in mobile networks", vol. 56, no. 4, **(2010)**, pp. 2161-2169.
[7] P. Zhihui and S. Juan, "Remote Wireless Monitoring Embedded System Design and Implementation in the Nature Reserve", vol. 39, no. 5, **(2012)**, pp. 1159-1162.
[8] K. Seajin, L. Byungjin and J. Jaewon, "Multi-object tracking coprocessor for multi-channel embedded DVR systems", vol. 58, no. 4, **(2012)**, pp. 1366-1374.
[9] O. Purdila, L. A. Grijincu and N. Tapus, "The Linux Kernel Library", vol. 16, no. 5, **(2010)**, pp. 316-319.
[10] B. B. Erdene, B. Lee and M. Y. Kim, "Extended smart meters-based remote detection method for illegal electricity usage", vol. 7, no. 11, **(2013)**, pp. 1332-1343.
[11] A. Sleman and R. Moeller, "SOA distributed operating system for managing embedded devices in home and building automation", vol. 57, no. 2, **(2011)**, pp. 945-952.
[12] O. Barnich and V. Droogenbroeck, "ViBe: A universal background subtraction algorithm for video sequences", vol. 20, no. 6, **(2011)**, pp. 1709-1724.

Authors



Zhongfu Tian, Male, Master of Engineering, Lecturer. He received the B.S. degree in computer application technology from Northeast Forestry University, Harbin, China, in 2003 and the M.S. degree in computer application technology from the same University, in 2006. He served as a teaching assistant at the School of Mechanical & Electrical Engineering of Northeast Forestry University (2006-2009). He joined the Department of Communication Engineering School of Mechanical & Electrical Engineering of Northeast Forestry University, worked on teaching and research in 2009. His general research interests include computer control, wood drying technology and artificial intelligence.

Yinglai Huang, He was born in October, 1978, who received his bachelor's degree in electronic information engineering (2003), master's degree in computer application technology (2006) and PhD in wood science and technology (2013) from Northeast Forestry University. Now he is mainly engaged in signal processing and computer intelligent processing direction as a full Associate Professor in Information and Computer Engineering College of Northeast Forestry University, he has published over 10 papers and hosted or participated in over 10 items of national and provincial projects.