

Embedded Network Video Monitoring System

Chen Long^{1,a}, Zhang Zhihui^{2,b}, Gao Peng^{3,c}

^{1, 2, 3}School of Electronic Information, Hangzhou Dianzi University

^{1, 2, 3} Hangzhou, Zhejiang Province, China

^aChenlong@hdu.edu.cn, ^bzzh326151@sina.com, ^cgaopengonly@163.com

Keywords: video surveillance, Embedded Linux, ARM, MPEG-4, V4L2

Abstract. With the continuous development of network technology and multimedia technology, video surveillance is toward IP, networking, high-definition, intelligence to develop. The applications of video surveillance are not just the connection of monitoring equipment, also including the market demand for video surveillance applications from simple equipment and facilities needs to start toward the overall solution to meet the individual needs of the industry. "Networking" has become the core business needs of video surveillance. Because the cost of traditional video surveillance system is high, and its transmission of large amount of video data cannot be in real-time and long distance, we have designed this embedded video monitoring system which is based on an ARM. The system uses embedded processor S3C2440A of Samsung's ARM family, and transplanted embedded Linux operating system which is after cutting and includes the USB camera-driven. It uses the programming interface (API) provided by V4L2 module for video data collection and selects MPEG-4 video compression standard for video compression, then upload the video data to the embedded boa server. Through the browser or client application, users can access boa server to get the compressed video data to realize video surveillance.

1 The basic components and working principle of the network video surveillance system

1.1 The basic components of the system

This system is mainly through collecting video data by the camera, and transmitting to the development board, then after processing and compression by the processor on the board, and ultimately sent to the client to realize the purpose of video surveillance. As shown in Figure 1, during the transmission of data, the system can be divided into three main parts which includes collection, processing and display part.

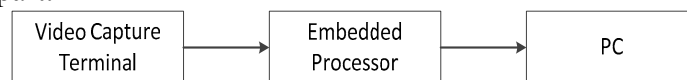


Figure 1. System composition diagram

Video capture terminal. Video capture is the process that turning analog signal of video source into a digital signal by processing, and output the digital information based on a certain interface standard. This part is mainly composed by the camera, and sent the video signal captured by the camera to the embedded processor. With the increasing maturity of photographic imaging technology and Internet technology, USB camera to get a very good development, its nature becomes higher price, applications become more widespread. Therefore this system selects general USB camera video instead the high cost video capture card as capture device.

Embedded processor. It is the core of the embedded video surveillance system, mainly composed by S3C2440A processor of ARM9 family and its associate peripheral equipment. It has good processing capabilities, and can operate and process the relevant data independently. It can also run the Linux operating system on the processor, then process compression and network transmission to the video signal captured by the camera, and as a small server.

PC. We access the boa server on the embedded processor mainly by via LAN and Internet to obtain the video signal processed by the processor. By client or browser to get the video data, then displayed again after decompression, and ultimately realize video surveillance.

1.2 The working principle of the system

The system designed by this scheme can be run on S3C2440 platform and support embedded server bo.a. It is a embedded Linux system. Linux system's powerful networking capabilities and multitasking support function ensure the normal operation of the system. Its main flow as follows: Upload the video images captured from USB camera to the embedded server after processing and compressing on ARM9 S3C2440 processor series. Then we can send it to the PC in real-time through the network. The signal decompressed by the PC-side can be displayed after calling client by running a video player. The transmission scheme of this program is mainly based on the connection, reliable TCP socket protocol. The sending program of ARM and receiving program of the PC are designed according to it. So when there is a loss of signal during the process of video transmission, it can be calibrated inspection and re-issued, providing a guarantee to the reliability of video transmission.

2 Hardware Design

The total hardware structure shown in Figure 2:

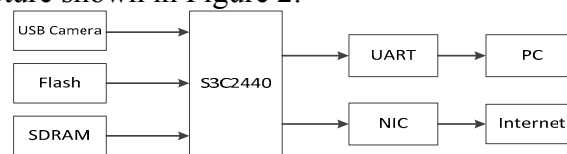


Figure 2. Total hardware configuration diagram

The S3C2440A of ARM9 series selected as the core processor in this system controls the video capture, processing and transmission. It integrates ARM920T core 16/32 bit embedded RISC microprocessor of ARM Company. It is designed for high-performance, low-power applications, with MMU, clocked at 400MHZ, up to 533MHZ. The platform of this system has an on-board 64MB SDRAM、64MB NAND Flash and 2MB NOR Flash. It also has a USB Host、one USB device、a 100M Ethernet RJ-45 port and serial port.

3 Software design

Shown in Figure 3, from a software point of view an embedded Linux system can often be divided into four levels, the boot loader, Linux kernel, file system and user applications.

Boot Loader. It is the paragraph of the program ran after power on the system for embedded systems. Its role is similar to the PC's BIOS. Boot loader should have the following functions: initialize RAM, initialize the serial port, detect the processor type, set the Linux boot parameters, and call the Linux Kernel image.

Linux kernel. embedded board-specific custom kernel and kernel boot parameters. It mainly exists in two forms in memory: one is compressed called z-Image, another non-compressed called Image. Image is performed without decompression and fast speed. However, due to limitations of embedded systems storage capacity, generally the kernel must be compressed.

File System. With the regularity, sociability and flexibility requirements improvement of system function of the embedded operating system, the embedded file system is different from the normal one. It is mainly custom-made for some special functions, including the root file system and the file system build on the Flash memory device.

User Applications Program. The applications are specific for users. The programs we usually run are in this layer. To provide a guarantee for the security of the system, it shields the bottom of all hardware interfaces. Sometimes between user applications and kernel layer may also include an embedded graphical user interface. The commonly used embedded GUI has Micro-Windows, Mini-GUI and so on. Shown in Figure 4 is the flow chart of the software system:

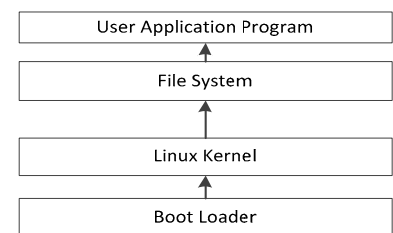


Figure 3. System software level diagram

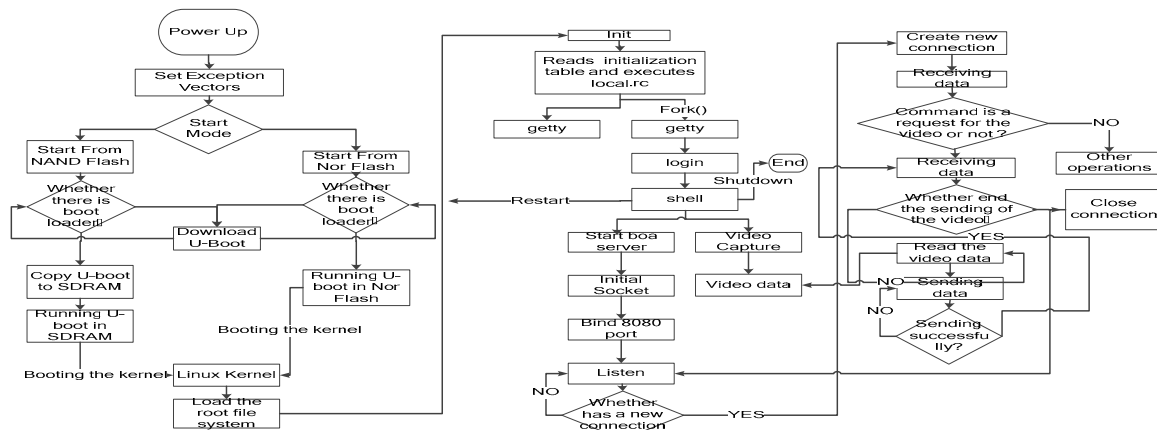


Figure 4. Flowchart of the total system

The system will start from different places according to the choice of different boot modes after power. When selecting boot from NAND Flash, NAND Flash is not supported due to perform in the film, so the U-boot will copy the code to SDRAM, and then execute code from SDRAM. However, the code can be executed directly in the NOR Flash. After loading the file system, it will initialize the environment variables, and create a new process to run the shell, and then we can open the boa server and video capture from entering shell commands on PC port. The server mainly used reliable TCP agreement based on protocol connections for video data transmission.

3.1 Boot loader, kernel transplant and file system produced

Boot Loader transplant. In the embedded operating systems, Boot Loader runs before the operating system kernel. It can initialize the hardware equipment, and establish space map of memory. The system hardware and software environment which will be brought to an appropriate state is for preparing a right environment to the final calling operating system kernel. Boot loader used in this paper is widely used and its source code directory, compiled form and Linux kernel are very similar to U-Boot. The appropriate version of the source code can be downloaded from the website: <ftp://ftp.de/pub/u-boot/>. Due to the current U-Boot does not provide support for the S3C2440, so you need to transplant manually. Fortunately their interface resources are very similar, just have some differences in the peripherals and running frequency of CPU, so we can refer S3C2440 configuration file to modify the configuration.

Kernel transplant. Although the Linux kernel source file has nearly more than 20,000, its design uses a modular design approach which allows us to cut out the module. Since S3C2440 belongs to ARM architecture chips, so during transplantation, the main job is to modify the kernel directory files under arch/arm. At the same time we need to amend the relevant drivers, such as USB, LAN, serial, etc. Appropriately modified to them will add support for USB camera, network, serial communication and YAFFS file system functions.

File system produced. File system is the operating system used to definite the data structures and method of the files in memory or partition. A basic file system should include: a busy-box which provides shell commands, configuration files of file system which are used to initialization and layout, equipment files, and necessary library file system. Common embedded file system has JFFS2, YAFFS, Cram-fs, Rom-fs, Ram-disk and Ram-fs. The first four are based on FLASH memory; the latter two are based on RAM. Porting process of the file system is as follows: transplant Busy-box, install glib-c library, set the Linux system boot process, create the Dev directory, create etc. directory, and create a home directory. Finally, build a good file system YAFFS authoring tools mkyaffs2image compilation.

3.2 Video capture subroutine design

V4L2 (Video for Linux two) is a uniform interface that the application program access the video drive provided by kernel. Under Linux, all peripherals are viewed as a special type of document; we call it “device file”. You can read and write like visiting ordinary file. In general, the camera equipment file drove by V4L2 is video0 in v4l directory under Dev directory. For GM, you can create a link to video0 in Dev directory. V4L2 supports two ways to capture images: memory mapping

(m-map) and direct read mode (read). The former one is generally used to capture continuous video data and the latter is commonly used in the static image data acquisition. In this passage, we use the former one. The flow chart of video capture is shown in figure 5.

When capturing the video, just need open the video0 file under Dev directory like normal file, and then set the parameters of the device, such as data flow types, data formats, the size of the image and so on. Then initialize memory mapping and set the size of the buffer. Start streaming IO to collect images, waiting for a complete acquisition to compress the image. At the same time give a judgment to whether it continues collecting. If not, then continue acquisition, otherwise release mapped memory and turn off the camera equipment.

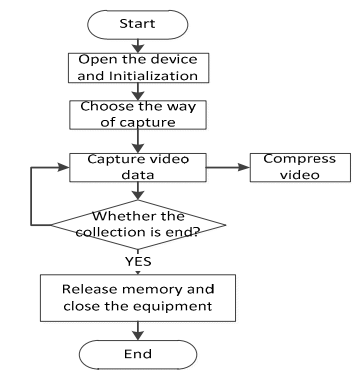


Figure 5. The flow chart of video capture

4 Conclusions

The architecture of embedded network monitoring system based on ARM has used the current popular embedded core Linux. It also has a file system which is custom-made for video monitoring. The system has the following characteristics:

(1) It has more versatility and wider range of use than the current video surveillance system based on PC.

(2) Low power consumption and low cost, can meet the requirements of video surveillance.

(3) Boa server supports multiple client connections, capable of serving multiple clients. However, due to limitations of system resources, in a multicultural connection, the video does not appear smooth phenomenon.

(4) The design of system client software is based on the current popular Qt technology, so that it has good portability.

This structure ported a stable performance boa server which provides a guarantee to video data request. The video coding, decoding, transmission and display are entirely based on software implementation. Through applying this system in different environments, it proves that the system can make full use of network bandwidth resources, achieving a good result of video monitoring.

Acknowledgments

This work was supported in part by the Zhejiang Provincial Natural Science Foundation of China under Grant No.LQ12F01001, and the Scientific Research Fund of Zhejiang Provincial Education Department (No.Y200803237).

References

- [1] Liang Duguo. Network Video Surveillance Technology and Intelligent Application. Press of People Post, 2013.
- [2] Qiu Tie. ARM Embedded System Architecture and Programming. Press of Tsinghua University, 2013.
- [3] Hou Dianyou. C Language Programming Based on ARM9 Microprocessor. Press of Tsinghua University, 2013.
- [4] Luo Yi. Embedded Linux Training Tutorials Based on ARM9. Press of Xi'an University of Electronic Science and Technology, 2013.
- [5] He Qingyun, Liu Jizhong, Zhang Hua. A Novel Embedded System of Video Capture and Transmission Applied in Remote Monitoring for Wheelchair-Bed Service Robots. Digital Manufacturing and Automation (ICDMA), 2010, 88-91