Hardware/Software Design of a Low-end Embedded ARM Microprocessor Based IP Video Surveillance and Control System

Vu Huu Tiep, Nguyen Ba Dung, Tran Duc Thien, Bui Quang Minh, Pham Ngoc Nam

School of Electronics and Telecommunications
Hanoi University of Science and Technology
vhuutiep@gmail.com, dungnb.fet10@gmail.com, thienthien.tran88@gmail.com, mbuiquang@gmail.com, pnnam-fet@mail.hut.edu.vn

ABSTRACT
This paper presents the hardware and software design of IP video surveillance and control system based on ARM architecture platform and Linux operating system. The system allows users to observe video images captured by a camera via any standard web browser and control remote devices. This design focus on building an ARM microprocessor processor based embedded computer and the design of operating system services for running mjpg-streaming server. With the use of Linux operating system running on low power microprocessor the system can be upgraded easily to work with various network interfaces, more client service and more control functionality without incurring high additional design cost.

Keywords
ARM, Linux, embedded system, IP camera.

1. INTRODUCTION
IP video surveillance and control systems have widely been used in many applications such as store/super market, airport terminal monitoring, school management, intelligent building management system (IBMS) or in remote controlling of base transceiver station (BTS) in wireless communication etc. In these applications, the user can use a PC or portable devices to login into the remote system via a web browser or a dedicated application and then views the images captured by cameras and may active controls such as turning on/off lights or fans, air-conditioners. In addition, the user may also verify sensors status like temperature, water flow and electrical parameters via a simple user interface on the web-browser (Figure 1).

One advantage of IP cameras is the ability to use an existing network wiring infrastructure to support surveillance and control system. Especially, this connection could be wired or wireless and they can be accessed from anywhere. In this paper, we propose a low-end embedded computer solution to perform both web server functionality and image compression using embedded design flow provided by Atmel [7]. The purpose of our work is to design a low cost low-end video surveillance and control system using an AT91SAM9260 chip, low cost USB-camera and an extended control board to control the various devices.

Figure 1. System overview

The rest of the paper is organized as follows: In Section 2, the system architecture is described. The hardware (HW) design of embedded server is given in Section 3. Section 4 presents software (SW) design and in Section 5, we discuss the results. Finally, the paper is concluded in Section 6.

2. SYSTEM ARCHITECTURE
The architecture of the ARM microprocessor based IP video surveillance is shown in Figure 2. The system architecture consists of an ARM9 embedded AT91SAM9260 chip connected to other components including memories, network transmitter, I/O devices via an external control board.

The control board is a custom designed board in charge of receiving control signal from ARM embedded computer board to control the I/O devices such as lights, fan, air-conditioner etc. The camera will communicate and send video signal to system’s center via USB standard. The system’s center here is the ARM 32 bit microprocessor which encodes the video stream and transmits the compressed video over Ethernet. All of software services are implemented on Linux custom built operating system.
Within the AT91Sam9260, the incoming video stream is processed by an open source C program. The ARM then packetizes the data using the RTP/RTSP over UDP/TCP protocol and sends the stream to the Ethernet PHY chip for transmission over the network. A web server running on the ARM within the AT91Sam9260 allows users on the network to control the system through a browser utility. Besides, the system setting could be changed by a PC over the UART interface or SSH network interface. In our system, we design the control board with variety of features for more control functionality such as controlling electrical devices with various voltage levels, multiple sensor inputs.

3. HARDWARE DESIGN

This section describes the design of the embedded server board and the control board in our system.

3.1 Embedded server board

The heart of hardware system is AT91sam9260 chip which is a low power SoP consisting of a 32 bits general purpose microprocessor ARM9260ej and integrated peripherals.

According to the functional requirements of the system, the system should be able to capture video from a Webcam, compress image by software and transmit compressed video over an IP network. Therefore, interface design plays a very important role in the hardware design.

3.1.1 Video Interface

In our system, interface between the Webcam which captures the video images and the MJPEG (Motion JPEG) encoder implemented in software running on AT91SAM9260 is done via an USB interface. By setting the UDP (USB device port) block of the microprocessor with the appropriate parameters, the USB 2.0 compliant interface with a speed of 12 Mbits/s has been obtained, which allows the system to use Webcams from any vendor.

3.1.2 System Memory Interface

Two types of the memory are used in the system including nonvolatile flash memory and SDRAM. To store boot-loader, environment variables and important system parameters, 4Mbit AT45DB041D DataFlash chip running at 66 MHz is used. The DataFlash is designed to interface with the processor via SPI. To store the image file of the OS, root file system and applications, we use a 256 Mbyte NAND flash memory. The NAND flash interfaces with the processor bus directly using SMC (Static Memory Controller) as shown in Figure 3.

![Figure 3. NAND flash interface](Image)

The main memory of the system is the SDRAM. The Micron 32 Mbytes MT48LC64M4A2 SDRAM is selected in the design because of its integrated on-chip SDRAM Controller.

3.1.3 Network Interface

The network interface of the system is shown in Figure 4.

![Figure 4. Network interface](Image)

The interface uses a Davicom chip which is working in physical layer of the OSI model. The chip is connected to the on-chip Ethernet MAC which is compatible with the IEEE 802.3 standard, with 10 or 100 Mbps data throughput capability.

3.2 Control board

The control board we designed consists of a Pic16F877 chip which is working in physical layer of the OSI model. The chip is connected to the on-chip Ethernet MAC which is compatible with the IEEE 802.3 standard, with 10 or 100 Mbps data throughput capability.

4. SOFTWARE DESIGN

The software design of the system consists of three main tasks, these are porting and customizing the Linux kernel on the hardware, writing device drivers and developing application software.

The basic software of system works based on an embedded Linux operating system. The OS is constructed using the CLFS method (Cross Linux From Scratch) for accommodating with hardware and software’s needs. All device drivers are developed to run on Linux kernel 2.6.30 [6]. A main advantage of CLFS over an automated system is that errors are usually easy to identify, since everything is done manually at the terminal and developer will probably watch the configure scripts and compiler output. If something doesn’t look right, it’s easy to find idea of where the problem is, if not what the problem is.

By developing our own Linux OS release the options of external hardware design are easy to implement because of the support of
open source community. Moreover when useful utilities are integrated on the system the development of software application is more favorable, like working on the computer.

In order to run Linux OS on embedded environment, besides the support of hardware, the OS should be built as small as possible but still has to meet the application’s needs. As a full operating system, Linux kernel is responsible for program management, scheduling, memory management and peripheral driving.

The build of system consists of cross tool-chain (arm-Linux-gcc, -GDB, etc.), the secondary boot-loader (U-boot), Linux kernel and root file system. All of these is referred available source code [6] and CLFS method [3]. A Linux kernel is not useful without a root file system containing applications and settings. In our design, the root file system is created in several formats: mountable over a network (NFS) for development and then JFFS2 (Journaling Flash File System version 2) for storing in NAND flash (flash file system).

For reliability and flexibility considerations, ARM based IP camera uses a layered software architecture as shown in Figure 5. The architecture is divided into four layers, namely: device driver layer, OS layer, middle layer (media database and network) and application layer.

The minimal configuration of kernel includes a set of device drivers (camera interface drivers, serial drivers, network interface drivers, USB interface drivers, etc.), network protocol processing, monitoring, receiving and forwarding control procedures.

![Figure 5. Software layers](image)

After optimizing and customizing the embedded Linux kernel version 2.6.30, the kernel image is less than 1.2 MB and the boot-loader is less than 150 KB. In the device drives, 802.3 Ethernet, SD memories, SDIO, USB-OTG, video input devices, general purpose I/O are activated.

For multimedia processing, the software supports USB camera using USB video device class (UVC) and MJPEG format (using V4L2 API). The USB Device Class Definition for video devices, or USB Video Class, defines video streaming functionality on the Universal Serial Bus, all of APIs are support by kernel 2.6.30 [6].

As for network protocols, the system supports TCP/IP, UDP, SMTP, HTTP, FTP, Telnet, DHCP, NTP, DNS, DDNS, PPPoE, and UPnP, etc. In addition, some network services are also built-in for system’s flexibility during operation and maintenance including IP setup by easy IP installer/ DHCP/ Static IP, secure login by Open-SSH. Our design can also support wireless model with IEEE 802.11b/g wireless network standard via USB port.

In the application layer, we design the main application of the system with the following functions:

- Capturing pictures from camera and compressing it to JPEG format.
- Initializing a web server service for user interactions, transferring MJPEG stream to the client.
- Decoding control signals from client and forwarding to external board via RS-485 network to control various devices.

**JPEG compression**

The JPEG compression block is in charge of receiving the images captured from the USB camera which is in RGB-24 raw data and compressing the raw data in JPEG format. The JPEG encoding is implemented in the number of steps as shown in Figure 6.

![Figure 6. JPEG encoding](image)

**MJPEG Video transmitting**

When receive a request from a client the application will transmit the MJPEG video stream to the client by launching HTTPD (hypertext transfer protocol daemon) process to handle http request from client. The web server service provides a webpage that includes window to display video in MJPEG format, buttons to control the devices as shown in Figure 10. The contents of the web page will be transmitted to the client by HTTP while MJPEG stream will be transmitted by RTSP (real time streaming protocol).

![Figure 7. A snapshot of the web interface](image)
Devices controlling

In fact, the electrical devices which need to be controlled are usually placed far away from the camera. Our design use RS-485 network to forward control signals from client to those devices via the control board with the firmware written in C programming language.

5. IMPLEMENTATION RESULTS

The whole system with a camera-server and an external control block has been implemented successfully on 2 separate hardware boards as shown in Figure 8. The camera-server board (which is on the left hand side in Figure 8) has the size of 13cm x10cm. The control board can accommodate up to two 250V/30A outputs, seven 125V/1A outputs, four 5V sensor inputs and can be placed up to hundreds of meters from camera-server board with the RS485 standard. The system can be automatically rebooted when system errors occur than to the use of hardware watchdog timer.

6. CONCLUSIONS

In this paper we described the design of an ARM architecture based IP video surveillance and control system in both hardware and software aspects. Both image compressing and streaming functionality are accelerated through ARM hardware implementations. In our future work, we plan to use a CMOS image sensor to replace USB camera. The sensor will be directly connected to the image sensor module which is a built-in block of the AT91sam9260 chip. The solution will help reduce hardware design cost and improve system speed.

7. REFERENCES