ISSN: 2277-3754
ISO 9001:2008 Certified
International Journal of Engineering and Innovative Technology (IJEIT)
Volume 3, Issue 11, May 2014

An Embedded System for Video Encoding and Decoding

Chin-Chih Chang and Kuo-Wei Chuang
Department of Computer Sciences and Information Engineering, Chung Hua University
Hsinchu City, Taiwan

Abstract—Surveillance systems have become widely-used equipment for security. Most of them are PC-based but the cost of a PC would discourage a customer to purchase them. Hence, it would be reasonable to develop a cost-effective embedded system for surveillance video recording. Though H.264 can provide higher image quality and compression rate than the widely-accepted Motion JPEG (MJPEG), it requires more computing processing power. In this paper, we present an embedded digital video recorder (DVR) designed to reduce the cost and increase video quality. The system is built on Linux. USB is used as a storage medium for video and audio. H.264 Codec (Coder-decoder) hardware is built in the system. While the video is stored in the H.264 format, the audio is stored in ADPCM (Adaptive Differential Pulse-code Modulation) format. The system is built under limited budget and validated by experiments. The experimental results shows the system is feasible.

***Index Terms*—DVR; Embedded System.; H.264; Linux; Video compression.**

I. INTRODUCTION

With improvement of the living standard and security consciousness people are more aware of home safety and living quality. It is now quite common to see surveillance systems in the public areas but these systems are not designed for a family in terms of the cost. Their cost are mainly from the PC on which the system are built. The better solution is to build an embedded system for video recording designed with hardware optimized for surveillance and image processing. In this way the cost can be reduced and the processing capability can keep in the same level.

The core part of a surveillance system is digital video recording. The traditional surveillance system can be enhanced from two directions: focusing on core functions of surveillance and image processing and modulizing other extended functions such as networking and storage. To meet the purpose of system modulization we need an operating system that is so flexible and extendable. Linux is such a system that is widely adopted in a variety of devices equipped with different sorts of processors such as ARM, PowerPC, x86, and MIPS [1].

The traditional standard for video recording is MJPEG [2]. Though the new standard H.264 has higher compression rate [3, 4], it requires higher computing power than MJPEG. The obvious benefit of H.264 is higher image quality with less storage space. Higher compression rate is also good for file transfer in networks [2].

In this paper we present the embedded system we have designed and implemented. The paper is organized as follows. Section 2 discusses the related work. Section 3 describes the system design. Section 4 illustrates the system implementation and experiments. Section 5 presents the conclusion and future work.

II. RELATED WORK

The components of an embedded system can be divided into three layers: hardware, operation system, and application layers [5, 6]. We describe related work and specific hardware for a video recording system from these three aspects.

A. Embedded Hardware Design

To build an embedded system, the first thing needs to do is to create a hardware, commonly a PCB (printed circuit board) with mounted electronic devices. Basically, this can be accomplished in four steps with the help of electronic design automation (EDA) [6, 7, 8]:

1. Schematic capture: In this step the goal is to create a circuit as simple as possible using the most common and economic electronic components following to the design specifications. During this session capture tools such as OrCAD Capture, KiCad Eeschema are adopted to help design and revise circuits. After continuous discussions and revision the electronic diagram will eventually reach the design goal.
2. Circuit simulations and verification: In this step the functionalities and feasibility and of electronic circuits designed in the capture step is verified through simulation tools such as OrCAD PSpice, KiCad Spice. If simulation results show that the circuits do not meet the specifications or do not perform steadily as expected, then go back to the schematic capture step to revise the circuits.
3. Circuit layout: In this step tools such as OrCAD Layout, KiCad Pcbnew are used to layout the circuits according to the electronic diagram in the schematic capture step.
4. Board production and surface mounting: In this step PCB is produced and then electronic components are mounted on the circuit board. Once a PCB is completed, further testing can be applied for possible next round of revision.

B. H.264 Video Compression Standard and IP Cameras

Video is a sequence of images closely related in temporal

ISSN: 2277-3754
ISO 9001:2008 Certified
International Journal of Engineering and Innovative Technology (IJEIT)
Volume 3, Issue 11, May 2014

and spatial spaces. To facilitate fast retrieval and storage, video is generally encoded in a format that is fast for compression and decompression. MJPEG (Motion JPEG) is a video format in which each video frame is encoded as a sequence of JPEG images. It is now used by video-capture devices such as digital camera, webcams, and IP cameras. MJPEG is not so inefficient because of JPEG.

With the advance and popularity of video devices and video sharing platforms, video capture has become quite common. Video files have been dramatically increasing in these days. A more efficient video format with higher quality needs to be developed. H.264 or MPEG-4 Part 10 is a recent standard which offers higher compression rate and image quality. The major improvements come from two reasons: more accurate motion representation and more flexibility in prediction filtering [3, 4]. H.264 is best known as being the video standard for Blu-ray discs. Though H.264 had been released in 2003, it is not quickly adopted in industry because it requires much higher computing capability than MJPEG. Nowadays, H.264 chips are becoming more available. Interested read can refer to [3] or [4] for more details about H.264.

IP (Internet Protocol) camera is a type of digital video camera equipped with networking capability. Since it was released in 1996, it is now commonly used for surveillance. IP cameras are further enhanced with different functions through different technologies of image processing. These enhancements can release a lot of human work and possible provide more accurate image analysis. For example, video content analytics is can help detect different events such as if any object enters or leaves a predefined perimeter, an object disappears or is stolen.

C. *Embedded Operation System Design*

An operation system (OS) is a program that are used to manage the system resources and is required for running user interfaces or drivers. Though there are quite a numbers of operating systems, nowadays, Linux has become a popular OS for an embedded system not only because it is an open source but also it is small. Linux can be customized to run in a very limited computing resource. For example, every Android phone has Linux in the base layer on top other software components can be built.

A user interface is the point of interaction which could be a software or a hardware. They can present in a variety of facets: a button, a control panel, a touch panel, a voice control, etc. Firmware is a software contained in an embedded system to provide control programs for devices as drivers [4, 5]. When we use a device, our command is executed through a user interface or a control panel and then delivered to the driver to activate the corresponding device. Firmware is commonly written in assembly languages or C language.

III. System Design and Implementation

The procedures to develop the proposed embedded system goes as follows:

1. Define system specification/architecture: digital video recorder (DVR) as shown in Figure 1.
2. Design hardware.
3. Verify if hardware correct. If yes, go to next step. Otherwise, go to step 2.
4. Design software/firmware.
5. Verify if firmware is correct. If yes, finished. Otherwise, go to step 4.

A. *System Architecture and Functions*

We first specified our targeted system which is composed of three layers as shown in Figure 2. Each layer is described as follows:

- Hardware layer: Five components are built on top of this layer including CPU, memory, video capture, video encoder and decoder, Ethernet network, and USB.
- OS layer: This layer contains necessary operating system and drivers. Linux is adopted as OS because it is open source and can be easily customized.
- Application layer: This layer holds the running applications for video recording, file transmission, and network storage.

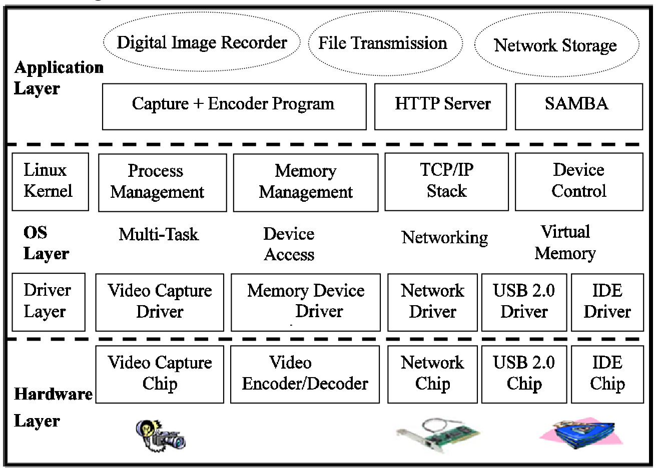

Figure 1. System architecture

Second, we employed OrCAD Capture to design the circuit as shown in Figure 2. After the circuit passed the verification of OrCAD PSpice, we applied OrCAD Layout to layout the circuit. Then, the PCB is produced and components are mounted. The finished hardware is shown in Figure 3.

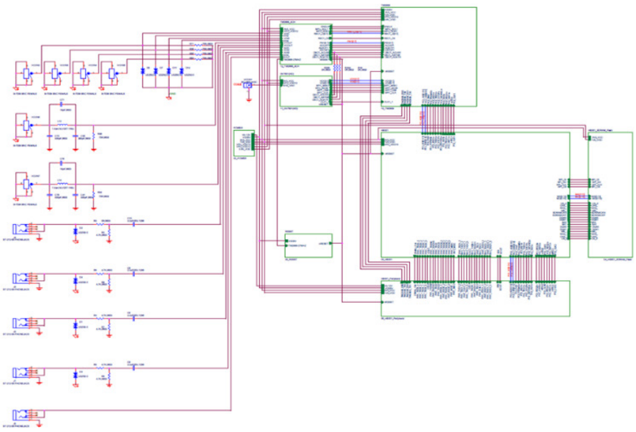

Figure 2. Circuit diagram

ISSN: 2277-3754
ISO 9001:2008 Certified
International Journal of Engineering and Innovative Technology (IJEIT)
Volume 3, Issue 11, May 2014

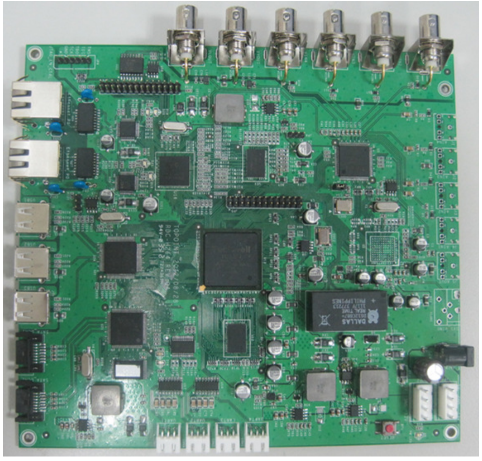

Figure 3. Finished hardware

The system functions are depicted in Figure 4. TW2866 is a 4-channel video decoder and audio Codec plus video encoder [9] and is used to convert input video and audio from analog to digital. TW2809 is a multichannel H.264 audio/video Codec and used to compress and decompress digitalized audio/video [10]. RDC's HB301 is a 32-bit x86 processor capable of supporting OS including MS-Windows, Linux, and DOS and is used to transfer those audio/video data through Ethernet and USB chips.

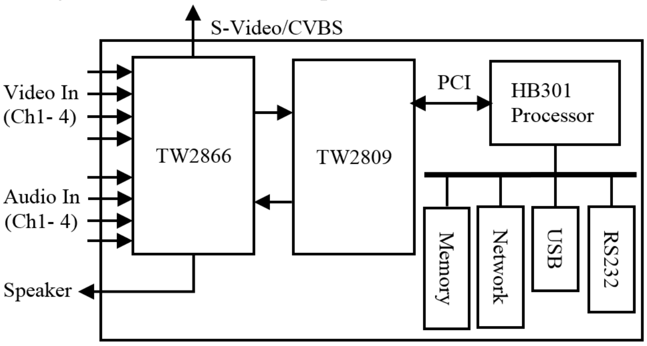

Figure 4. System function diagram

B. Operating System and Firmware

To make the embedded system work, we need install the operating system, firmware, and applications. This can be accomplished by a series of installations including development environment, cross compiler, boot loader, kernel, file system, driver, and applications. Usually, BSP (Board Support Package) is needed for the software layer installation including the following tasks:

1. Hardware initialization including readying CPU for other devices in the system.
2. Core OS Installation including boot loader, kernel, drivers, and interrupt services.
3. System software installation including multitasking initialization and basic system software.
4. OS initialization including readying OS for operations.

The processor of the system is HB301 whose BSP can be downloaded from the FTP site[1] of the manufacturer, RDC Semiconductor Co., Ltd. Installing BSP will take a couple of minutes until the necessary software is available to install into the embedded hardware as shown in Figure 5 and Figure 6.

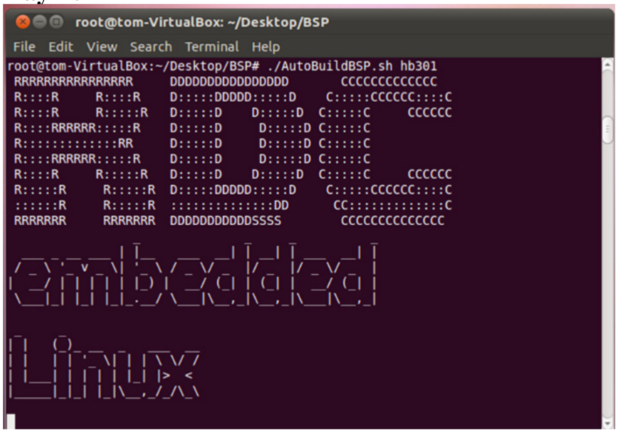

Figure 5. Executing BSP

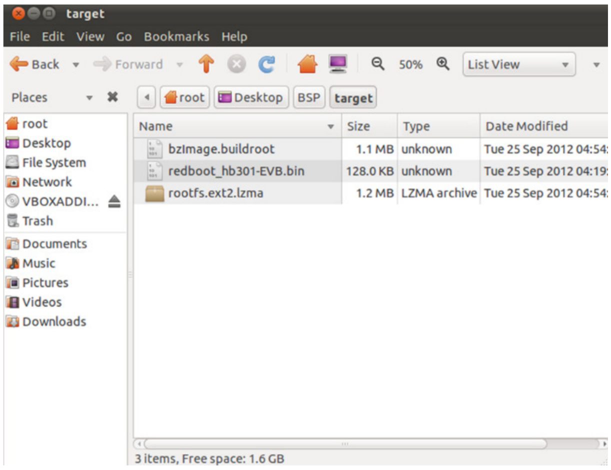

Figure 6. Boot loader, kernel, and file system

Next, a JTAG (Joint Test Action Group) tool provided by the manufacturer is connected to the USB port of the PC and the COM port of the embedded hardware. JTAG tool provides the necessary hardware and software to install the OS and software into the embedded hardware [5, 6]. We first select the boot loader and then use the JTAG software to load the boot loader into the system ROM as shown in Figure 7. Once the boot loader is successfully installed in the embedded hardware, we connect COM and RJ45 ports of the PC and the embedded hardware for the RS232 terminal and the network respectively. Then, we can use boot loader on RS232 terminal through the TFTP to load the kernel and file system into the embedded hardware as shown in Figure 8.

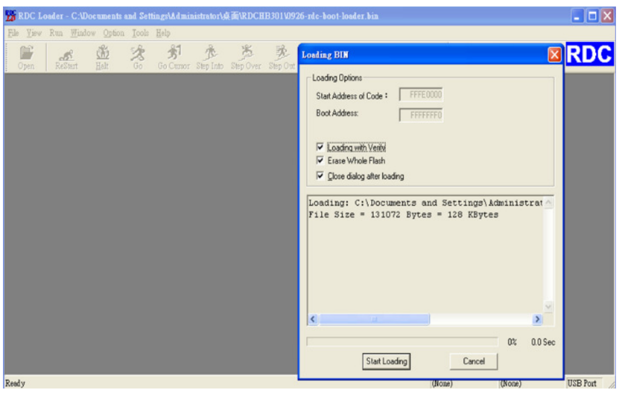

Figure 7. JTAG software

1 ftp://ftp.rdc.com.tw/pub/Linux/AutoBuildBSP.tar.bz2

ISSN: 2277-3754
ISO 9001:2008 Certified
International Journal of Engineering and Innovative Technology (IJEIT)
Volume 3, Issue 11, May 2014

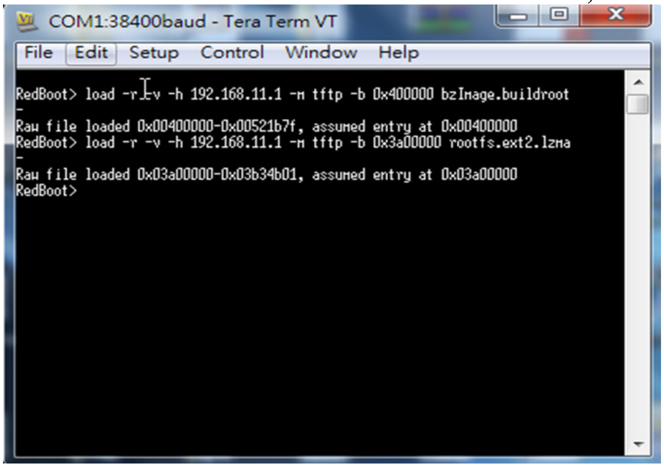

Figure 8. Loading kernel and file system

Because the original kernel does not have the drivers of TW2866 and TW2809, we need to recompile the kernel with source code of TW2866 and TW2809 drivers using a Makefile as shown in Figure 9 and Figure 10. After the compilation is completed, the new kernel is loaded into the ROM of the embedded hardware. Now, the embedded system contains the necessary system software and drivers.

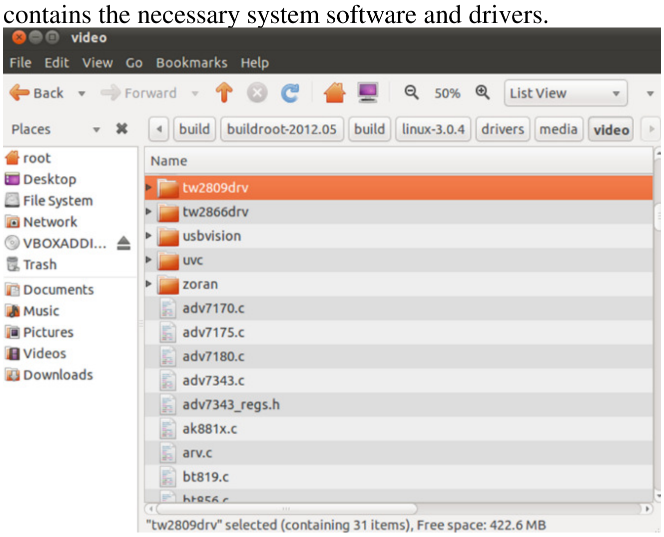

Figure 9. Source code for TW2866 and TW2809

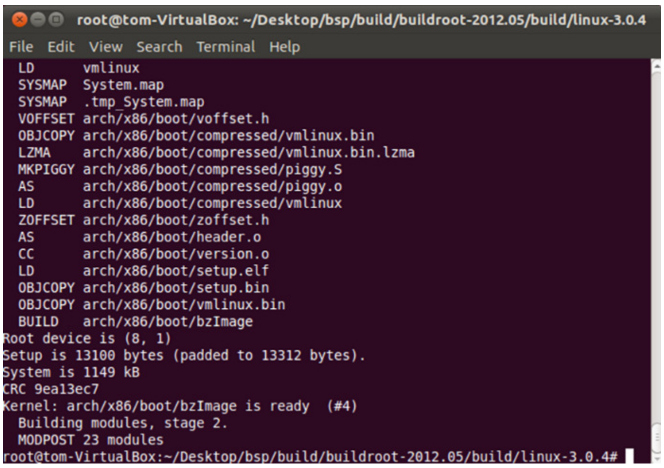

Figure 10. Making the new kernel

C. *Application Software*

The last step is to install the required applications which are video recording and playback in our case. We use the source code provided by the manufacturer. Video recording follows three steps: 1. ADC (Analog Digital Conversion): acquiring input video and converting them from analog signals to digital ones; 2. compressing them; 3. storing them into a USB flash drive. Video playback follows three steps reversely: 1. playing the video and audio on a USB flash drive; 2. decompress them; 3. DAC (Digital Analog Conversion): converting them from digital to analog signals and then output them. The processes are shown in Figure 11

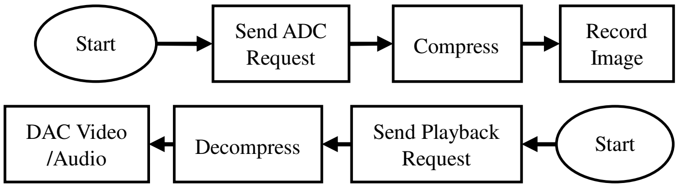

Figure 11. Video recording and playback processes

After the application programs are installed, the system is complete with hardware and software. To transfer the file between the host PC and embedded system, we can install NFS on the host PC and mount a directory of the host PC on the embedded system. Then we can use Linux commands to move around the video files as shown in Figure 12.

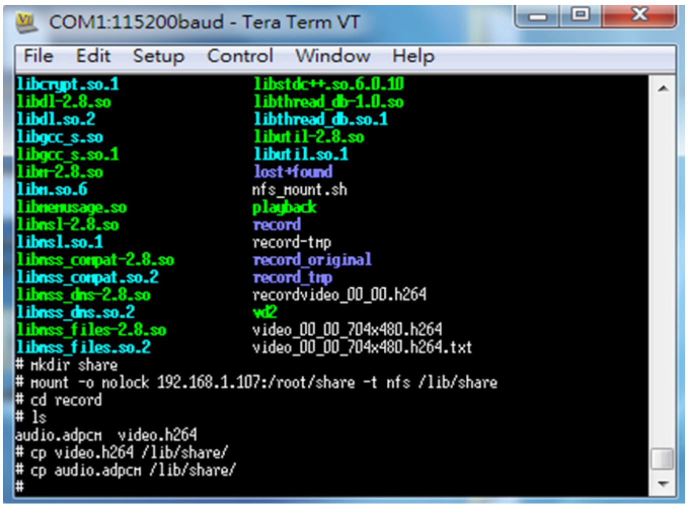

Figure 12. Mounting a directory of the host PC on the embedded system

IV. EXPERIMENTS AND ANALYSIS

A. *Experimental Results*

We have built a digital video recorder using much less cost but whether the low cost will sacrifice the performance was a question. Hence we undertook experiments to resolve our doubt. In our experiments there are two types of inputs: camera and DVD. We compare three systems: a low-cost PC (Pentium-based) with a PCI H.264 Codec card, a proposed embedded system with Nuvoton ARM9-NUC960 CPU [11], and the other one with RDC x86-HB301 CPU [12]. Then let them record two videos per channel of 1, 2, and 4 channels for

ISSN: 2277-3754
ISO 9001:2008 Certified
International Journal of Engineering and Innovative Technology (IJEIT)
Volume 3, Issue 11, May 2014

5 minutes and store them in an USB flash drive and then compare how much file size they can record. The experimental results for the PC and the proposed system are shown in Figure 13, Figure 14, and 15 respectively.

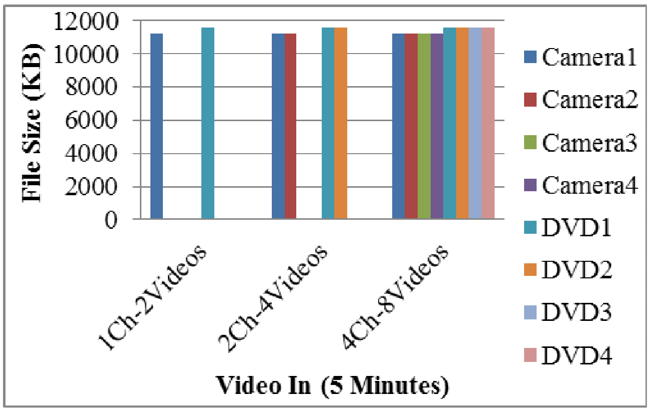

Figure 13. File size of 5 minute video recorded in USB flash drive of a Pentium PC with a PCI H.264 Codec card

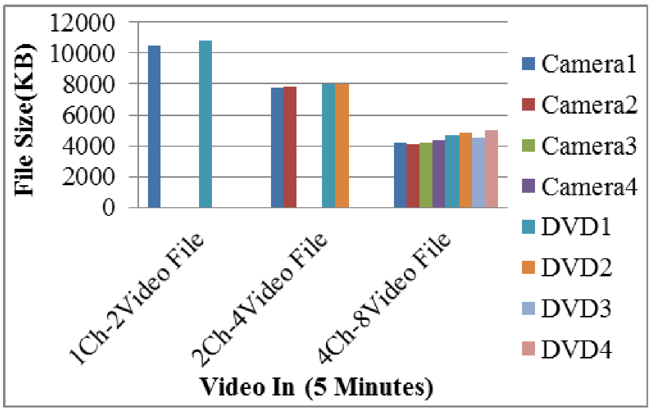

Figure 14. File size of 5 minute video recorded in USB flash drive of a Nuvoton ARM9-NUC960 embedded system

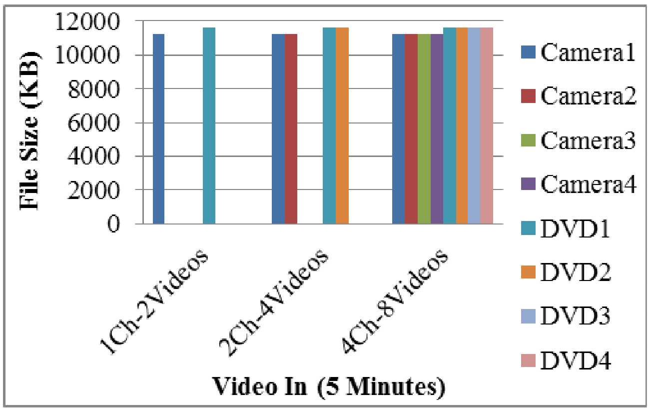

Figure 15. File size of 5 minute video recorded in USB flash drive of a RDC x86-HB301 embedded system

The experimental results show that PC system and the RDC x86-HB301 CPU system can record about the same size of videos during 5 minute session and more channel of videos do not affect the recording size too much. We then further verified the both videos by playing them. Both videos were playing smoothly without any noticeable loss. This indicates both systems can record videos in the required quality. But data loss happened in the system with Nuvoton ARM9-NUC960 CPU. The reason is that Nuvoton ARM9-NUC960 CPU without the fast back-to-back transactions [13] is not fast enough to process the data sent from H.264 chip and results in the data loss. Hence, RDC x86-HB301 is selected as the CPU in our design.

B. Analysis and Comparisons

From the hardware point of view experimental results show that the performance of DVR based on the proposed embedded system is equivalent to a PC but its cost is less than half of a PC as shown in Table 1. Sure, if we compare it with a high-end PC, its performance could be less but its cost is much less. Form the software point of view most current PC system are using Windows operating system while the proposed system are using Linux. Though installation on Linux could require a little bit more effort than on Windows, Linux is free. The comparisons of Windows and Linux are shown in Table 2. We can conclude that the proposed approach is very competitive.

Table 1. Comparisons between PC and Embedded system

	Personal Computer with a video capture card	Embedded system
CPU	Pentium/Core	Pentium/Strong ARM /PowerPC/x86 SOC
RAM	128M or above	8M~64M
Storage	Hard disk (large)	Flash drive (small)
Ethernet	NIC	Built-in
Console	VGA	Serial
Purpose	General	Specific
Stability	Fair	Good
Cost	High (~US $300)	Low (~US $120)

Table 2. Comparisons between operating systems

	Windows 7/8	Linux
Installation	Less effort	More effort
Driver Support	Good	Fair
Stability	Fair	Good
Cost	~US $50	Free

V. CONCLUSION AND FUTURE WORK

In this paper we present the design and implementation of an embedded system for surveillance video recording which is cost-effective without less performance. The system is validated by experiments and the results show it is feasible and the performance is equivalent to a PC. In addition, our design is flexible and extendable. Currently, we are modifying our design for an event data recorder in which HB301 processor plays the part of supporting fast back-to-back transactions.

In the future, we plan to add an IP camera module into the system. The other enhancement is adding a wireless network module that can facilitate wireless retrieval and storage. But before making it possible, we need apply some security

ISSN: 2277-3754
ISO 9001:2008 Certified
International Journal of Engineering and Innovative Technology (IJEIT)
Volume 3, Issue 11, May 2014

measure to make sure the data is safely transmitted. This would allow users freely to access the video through any mobile devices when they are in the nearby area.

ACKNOWLEDGMENT

This work is partly supported by the National Science Council of Taiwan under the grant numbers NSC101-2815-C-216-011-E and NSC100-2218-E-216-001.

REFERENCES

[1] Christopher Hallinan, Embedded Linux Primer: A Practical Real-World Approach, 2nd ed., Prentice Hall, 2010.
[2] Anthony C. Caputo, Digital Video Surveillance and Security, Butterworth-Heinemann, 2010.
[3] Thomas Wiegand, Gary J. Sullivan, Senior Member, Gisle Bjøntegaard, and Ajay Luthra, Overview of the H.264/AVC Video Coding Standard, IEEE Transactions On Circuits And Systems For Video Technology, vol. 13, no. 7, July 2003, pp. 560 – 576.
[4] Iain E. Richardson, The H.264 Advanced Video Compression Standard, Wiley; 2 ed., 2010.
[5] Tammy Noergaard, Embedded Systems Architecture: A Comprehensive Guide for Engineers and Programmers, 2nd ed., 2012.
[6] Peter Marwedel, Embedded System Design: Embedded Systems Foundations of Cyber-Physical Systems, Springer, 2nd ed., 2010.
[7] Kraig Mitzner, Complete PCB Design Using OrCad Capture and Layout, Newnes, 2007.
[8] Ronald Cohn and Jesse Russell, KiCAD, VSD, 2012.
[9] Intersil Americas Inc., Techwell TW2866 Data Sheet, Intersil Americas Inc., 2009.
[10] Intersil Americas Inc., TW2809 Data Sheet, Intersil Americas Inc., 2011.
[11] Nuvoton Inc., Microcontrollers Product and Tool Selection Guide, Nuvoton Inc., 2014.
[12] RDC Semiconductor Co., Ltd., HB301 Data Sheet, RDC Semiconductor Co., Ltd., 2007.
[13] Edward Solari and George Willse, PCI and PCI-X Hardware and Software: Architecture and Design, 6th ed., Research Tech Inc., 2004.

Chin-Chih Chang received the PhD degree in Computer Science from Oklahoma State University, USA. He is currently an assistant professor in the Department of Computer Sciences and Information Engineering, Chung Hua University, Hsinchu City, Taiwan. His research interests include service computing, intelligent systems, context-aware computing, and mobile computing.

Kuo-Wei Chuang received the Master degree in Computer Science and Information Engineering from Chung Hua University, Hsinchu City, Taiwan. He is currently an engineer in the Department of R&D, Fontal Technology Incorporation, Hsinchu City, Taiwan.