

리눅스 기반 4채널 임베디드 DVR 구현

이 흥 규*, 정 갑 천**, 최 종 현*, 박 성 모*
*전남대학교 컴퓨터공학과, **전남대학교 전자공학과
전화 : 062-530-0798 / 핸드폰 : 016-759-3577

Implementation of 4-channel Embedded DVR Based on Linux

Heung Kyu Lee*, Gab Cheon Jung**, Jong Hyun Choi*, Seong Mo Park*
*Dept. of Computer Eng., Chonnam National Univ.
**Dept. of Electronics Eng., Chonnam National Univ.
E-mail : hkleee@cep03w.chonnam.ac.kr

Abstract

This paper describes the implementation of a 4 channel embedded DVR system. It receives analog video from CCD cameras and converts to 640×480 CCIR-656 digital video by 30 frames/sec. These digital images are compressed to the wevelet transformed image using hardware codec which is capable of 350:1 real-time compression and decompression. The DVR is working on linux and it implemented on an embedded system which is based on StrongARM processor. For the interface between processor system module and image processing module, GPIO and memory control module are used, device drivers are developed. Linux kernel source is customized. This paper provides techniques of embedded system development and embedded linux porting.

I. 서론

최근 보안 시스템으로써 아날로그 CCTV를 대체하는

※ 본 논문은 한국과학재단 지정 전남대학교 고품질 전기전자부품 및 시스템연구센터의 연구비 지원에 의해 연구되었음.

DVR(Digital Video Recorder) 시스템이 보안 장비로 각광받고 있는데, 이는 DVR 시스템이 CCTV 방식의 아날로그 보안 시스템에 비해 저장 및 검색이 용이하고 장소제약이 적기 때문이다. 기존의 대부분의 DVR 시스템은 PC기반의 시스템에 영상캡처 카드를 장착한 후 윈도우즈 또는 리눅스 O/S를 탑재한 형태이다. 그러나 이러한 제품들의 가장 큰 문제점은 가격이 비싸다는 점이며, 특히 소프트웨어적으로 영상을 압축하여 속도가 느리다는 단점을 지니고 있다. 그러나 최근에는 프로세서 기술의 발달로 임베디드 프로세서에서도 일반 PC와 같은 처리가 가능해짐에 따라 DVR 시스템도 임베디드 형태로 변화하고 있는 실정이다.

임베디드 시스템의 구성에 있어 가장 중요한 요소 중 하나는 CPU 코어로서 현재 32비트 프로세서가 주류를 이루고 있으며 인텔, 모토로라, 썬마이크로시스템즈, ARM 등 많은 반도체 업체들이 경쟁하고 있다. 그 중 ARM 프로세서는 가격과 전력소모에 장점을 가지고 있어 많이 사용되어지고 있다[1].

임베디드 시스템을 구동하기 위한 운영체제는 WinCE, PalmOS, 리눅스 등 다양한 종류가 사용되고 있는데, 특히 리눅스는 사용료가 없고 신뢰성이나 기능성의 면에서도 뒤쳐지지 않아 저가형 시스템에 적합하다. 뿐만 아니라 개발에 있어서 리눅스는 소스가 공개

되어 개발자의 의도에 맞게 개발될 수 있고, 포지스(POSIX) 표준을 따라 일반적인 유닉스가 제공하는 API와 셸 기능을 대부분 지원하므로 응용 프로그램을 손쉽게 이식할 수 있다는 장점을 가지고 있다[2].

DVR 시스템은 크게 영상의 캡처 및 처리를 수행하는 영상처리 부분과 사용자 인터페이스와 제어를 담당하는 시스템 부분으로 나눌 수 있으며, 각 부분은 운영체제, 장치드라이버, 어플리케이션 등 소프트웨어에 의해 운영되어진다. 영상처리를 담당하는 부분은 카메라로부터 들어온 영상을 캡처 하여 압축 등의 처리를 한 후 시스템 모듈에 전달하는 역할을 하며 시스템 모듈은 전달받은 영상 데이터를 저장하고 사용자의 요구에 따라 재생하는 등의 역할을 한다.

본 논문은 저가, 소형화를 만족시키는 ARM 기반의 임베디드 DVR 구현에 대하여 기술하였다.

II. 하드웨어 구성

구현된 임베디드 DVR 시스템은 그림1과 같이 시스템 모듈과 영상처리 모듈로 구성되어지며 하드웨어 사양은 다음과 같다.

- StrongARM SA-1110 206MHz 32bit RISC
- 32MB SDRAM, 16MB 플래시메모리
- 이더넷 컨트롤러
- RS-232C Serial Port, JTAG Port
- IDE 하드디스크
- 4채널 비디오 입력, 1채널 비디오 출력
- 컬러 이미지 프로세서(영상분할)
- 웨이브렛 변환 압축 코덱

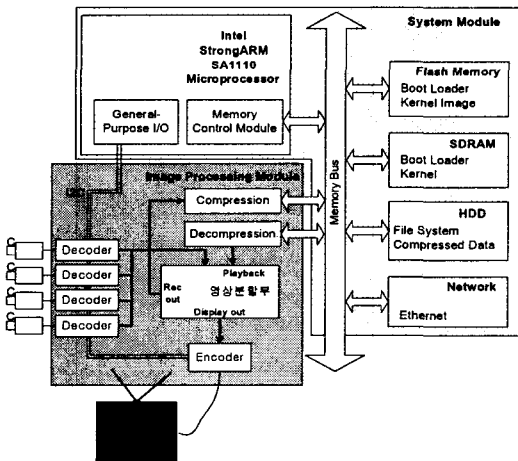


그림 1. 임베디드 DVR 시스템의 전체 블록도

1. 시스템 모듈

시스템 모듈은 마이크로 프로세서를 통해 DVR 시스템 전체를 제어하며 하드디스크와 네트워크 컨트롤러를 포함하고 있어 데이터의 저장과 네트워크를 통한 통신을 가능하게 한다. 시스템 모듈은 CPU 코어로 StrongARM SA-1110을 채택했으며 주 저장장치로 SDRAM과 부트로더, 커널 이미지를 저장하는 플래시 메모리, IDE 인터페이스를 사용하는 하드디스크, 그리고 네트워크 디바이스로서 구성되어진다[3][4]. 시스템 모듈의 보조기억장치는 플래시메모리와 하드디스크가 있으며 플래시메모리는 부트로더와 커널 이미지를 저장하는데 사용되어지고, 하드디스크는 파일시스템을 이용하여 필요한 라이브러리와 압축 데이터 저장을 위해 사용된다.

임베디드 시스템은 PC 호스트 시스템에 비해 사양이 떨어지며 개발 환경 또한 좋지 못하므로 소프트웨어 개발을 위해서는 교차개발환경이 필요하다. 따라서 임베디드 시스템은 교차개발환경을 통해 생성된 결과를 JTAG이나 RS-232C와 같은 통신 채널을 통해 전달받는다[5]. JTAG은 플래시메모리에 부트로더가 없을 때 플래시메모리에 부트로더를 업로드할 수 있는 통신 채널을 제공하고, RS-232C는 업로드된 부트로더의 수행으로 호스트 시스템과 임베디드 시스템간의 인터페이스 역할을 수행한다. 이를 통해 네트워크 설정 이전에 호스트 시스템을 이용해 임베디드 시스템을 제어하거나, 부트로더, 커널, 램디스크 이미지를 임베디드 시스템으로 전송하여 플래시메모리에 업로드할 수 있다. 부트로더 등 이미지를 더 빠르게 업로드하기 위해서 이더넷 컨트롤러를 이용한 tftp를 사용하기도 한다.

CPU 호스트는 시스템 모듈의 각 블록과 영상처리 모듈과의 인터페이스를 위해 GPIO (General Purpose I/O)와 메모리컨트롤모듈을 사용하도록 하였다. 영상분할부의 제어와 비디오 디코더/인코더 제어를 위한 I2C 통신에 GPIO를 사용하였고, 플래시메모리, SDRAM, 네트워크 디바이스, IDE 인터페이스, 압축/복원부 데이터의 전송은 메모리 버스클럭(CPU 클럭(206MHz) / 4 = 50MHz)에 동기되어 있는 메모리컨트롤모듈을 사용하였다.

2. 영상처리 모듈

구현된 영상처리 모듈은 영상분할부와 압축/복원부로 구성되어 있다. 영상분할부는 영상 캡처, 영상분할, OSD(On Screen Display), 모션 검출 기능 등 영상의 입출력을 담당한다. 반면 압축/복원부는 영상분할부로부터 들어온 영상을 웨이브렛 변환 압축방식의 하드웨어 코덱을 사용하여 인코딩 후 시스템 모듈에 전달하

고, 압축/저장된 데이터를 TV로 출력할 수 있도록 시스템 모듈에 저장된 압축 데이터를 디코딩하여 영상분할 모듈에 전달하는 역할을 수행한다.

영상분할부는 총 5채널의 디지털 비디오 입력과 2채널의 디지털 비디오 출력을 갖는다. 입력은 4대의 카메라로부터 실시간으로 들어오는 4채널 외에도 시스템 모듈로부터 저장되어 있는 영상을 복원하여 플레이 백할 수 있는 채널을 포함하고 있다. 출력은 디스플레이 채널과 레코드 채널로 나누어지는데, 디스플레이 채널의 경우 입력되는 영상을 한 채널, 4채널, 9 채널로 자유롭게 편집하여 디스플레이하고, OSD를 넣을 수 있다. 레코드 채널은 원하는 카메라 입력을 선택적으로 내보낼 수 있으며, 모션이 있는 채널만을 선택적으로 출력 할 수도 있다. 영상분할부는 Philips사의 비디오 디코더(SAA7113H), ALogics사의 Color Image Processor (AM204M), Analog Device사의 비디오 인코더(ADV7171)를 사용하여 구성되었다. 그림 2는 영상분할부를 통해서 영상이 4분할되어 TV에 출력되는 것을 보여 준다.

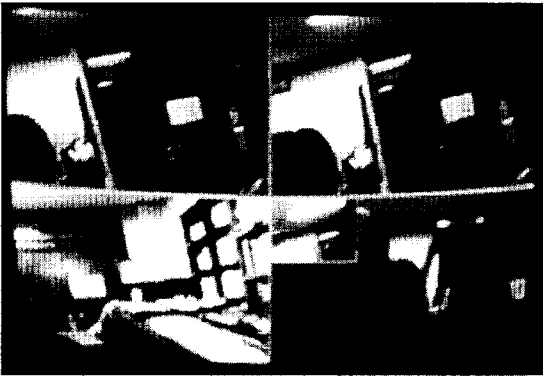


그림 2. 영상분할부를 통한 TV 출력 영상

압축/복원부는 웨이브렛 변환 압축 방식을 사용하며, 압축과 복원을 동시에 실시간으로 수행할 수 있도록 병렬 구성하였다. 인코드 모드에서는 영상분할부 레코드 채널의 디지털 영상을 압축하여 생성된 비트스트림을 시스템 모듈로 전송하며, 디코드 모드에서는 하드디스크에 저장되어 있던 압축된 디지털 영상을 시스템 모듈로부터 받아 복원하여 영상분할부에 보내준다. 웨이브렛 변환 코덱은 Analog Device사의 ADV601LC를 사용하여 구현되었다[6].

ADV601LC는 호스트 인터페이스로서 내부에 32비트 512 레벨의 FIFO를 가지고 있기 때문에 시스템 모듈은 ADV601LC의 FIFO 호환성을 고려하여 메모리 컨트롤 모듈의 인터페이스 중 Variable Latency I/O

를 사용하였다.

구현된 DVR 시스템은 그림 3과 같이 시스템+IDE보드, 영상분할보드, 압축/복원보드 등 3개의 보드로 제작되었으며, 각 보드들은 160(80×2)핀 Connector로 다른 보드들과 신호를 공유한다.

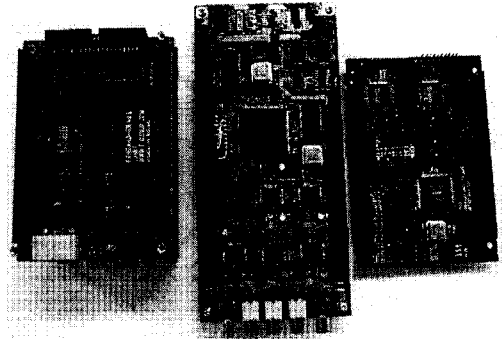


그림 3. 구현된 임베디드 DVR
(시스템+IDE보드, 영상분할보드, 압축보드)

III. 소프트웨어 구현

1. 리눅스 커널

임베디드 시스템은 PC나 워크스테이션과 같은 범용 시스템과 달리 기능에 따라 구조가 다르기 때문에 DVR 시스템의 운영체제로 리눅스를 사용하기 위해서는 커널 소스와 장치드라이버를 구조에 맞게 수정해야 한다. 사용되는 리눅스 커널은 (주)제이닷디엔티에서 제공한 커널을 기본으로 하여 DVR 시스템의 기능에 맞게 RS-232C 통신, 메모리의 구성, I/O 매핑, 네트워크 드라이버, 파일 시스템 등에 대한 수정과 설정을 맞춰 교차컴파일 과정을 거친 후 사용하였다[7].

교차 컴파일 과정을 거쳐 생성된 커널이미지 zImage는 부트로더와 함께 플래시메모리에 업로드된다. SA-1110은 메모리 주소 0x00000000을 부팅 주소로 지정하기 때문에 부트로더는 플래시메모리 0x00000000에, 그리고 커널이미지는 0x00020000에 저장하였다. 부트로더와 커널은 전원이 인가되면 0x00000000 메모리 번지에 위치한 부트로더의 제어의 의해 주기억 장치 SDRAM에 업로드되어 실행되게 된다. DVR 시스템의 메모리 맵은 그림 4와 같다.

2. 영상처리모듈 장치드라이버

리눅스는 장치를 파일로서 다루며 장치드라이버는 응용 프로그램에 open, read, write, ioctl, close의 여러

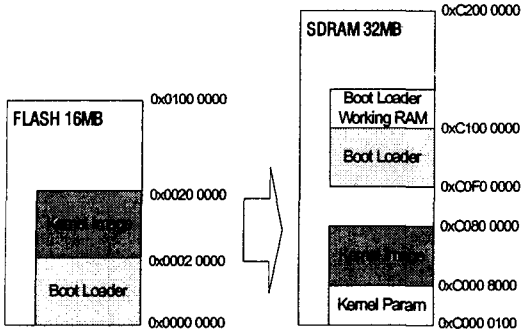


그림 4. DVR 시스템의 메모리 맵

파일입출력 함수를 제공함으로써 장치를 제어하거나 데이터를 획득할 수 있도록 한다[8]. DVR 시스템에서는 영상처리 모듈을 제어하기 위하여 각각 장치들에 제어를 위한 ioctl 함수와 데이터 전송을 위한 read/write 함수를 장치드라이버를 통해 구현하였다.

디바이스 드라이버는 크게 문자 장치드라이버, 블록 장치드라이버, 네트워크 장치드라이버로 나눌 수 있다. 문자 장치드라이버는 파일처럼 접근할 수 있는 것으로 데이터를 순차적으로 접근할 때 사용되며, 블록 장치드라이버는 블록 단위로 접근할 때 사용하며 버퍼를 통해 처리된다. 네트워크 장치드라이버는 네트워크 인터페이스를 제공하게 된다. 영상처리모듈은 스트림 기반의 장비이므로 문자 장치드라이버로서 구현하였다.

영상처리모듈의 영상분할부와 비디오 디코딩/인코딩을 제어하기 위한 GPIO는 입출력을 위해 핀 레벨 레지스터, 출력 셋/클리어 레지스터, 에지 상태 레지스터를 가지고 있고, 각 핀의 기능 설정을 위해 입출력 방향 설정 레지스터, 에지 검출 레지스터, 부가기능 설정 레지스터가 있다. GPIO를 이용한 장치드라이버는 기능 설정 레지스터를 용도에 맞게 초기화한 후 핀 레벨 레지스터와 출력 셋/클리어 레지스터의 읽고 쓰기를 반복함으로써 장치를 제어하도록 구현되었다.

압축/복원부는 SA-1110의 메모리컨트롤모듈을 이용해 인터페이스 하게 되는데 SA-1110은 버스 DMA가 없기 때문에 한번에 되도록 많은 데이터를 가져와야 한다. 이를 위해 부트로더에 압축/복원부가 사용하는 메모리 뱅크를 버스트 모드로 설정하였고, 압축/복원부의 FIFO 호환성을 고려하여 Variable Latency I/O 인터페이스를 사용하였다.

압축/복원부는 호스트 인터페이스로 32비트 데이터 버스, 2비트 주소, 그리고 인터럽트와 관련된 FIFO_SRQ, LCODE, nHIRQ핀을 제공한다. 다른 인터럽트 핀이 한번 호스트가 상태를 읽으면 클리어 되는 것과는 달리 FIFO의 준비 여부를 알리는 FIFO_SRQ는 호

스트가 상태정보를 읽은 후에도 클리어 되지 않고 FIFO_SRQ 상태를 유지한다. 따라서 FIFO로부터 데이터를 읽은 후 다시 인터럽트 대기상태도 들어가기 전에 FIFO_SRQ의 상태를 확인한 후 인터럽트 서비스 동안 채워진 FIFO 데이터를 곧바로 가져옴으로서, 보다 연속적인 데이터 전송을 할 수 있다. 인터럽트 핀 중 LCODE는 다음 데이터가 영상의 마지막임을 알리는 비트로서 영상 데이터 스트림에서 프레임 하나 하나를 구분 지을 수 있도록 하였다.

IV. 결론

본 논문은 저가, 소형화의 요구를 만족시키기 위한 4 채널 임베디드 DVR시스템 구현에 대하여 기술하였다.

DVR 시스템의 하드웨어는 StrongARM을 탑재한 시스템 모듈과 웨이브렛 기반의 압축 코덱으로 구성되어, DVR 시스템에 필요한 기능만을 가지도록 하였다. 그리고 소프트웨어는 교차개발환경이 갖춰진 PC 호스트를 통해 개발하였으며, 시스템의 저가, 소형화를 위해 사용료를 지불하지 않는 리눅스를 채택하였다.

본 시스템의 기본 기능은 카메라로부터 들어온 영상을 저장장치에 저장하는 것으로 응용 소프트웨어에 따라 DVR 뿐만 아니라 영상과 관련된 다양한 응용 분야에 적용될 수 있을 것으로 기대된다.

참고문헌

- [1] T. Litch and J. Slaton, "Strong ARMing Portable Communications", IEEE Micro., pp48~55, March/April 1998
- [2] 박재호, 임베디드 리눅스, 한빛미디어, 2002.11
- [3] ㄹ제이닷디엔티 홈페이지 <http://www.falinux.com>
- [4] Intel StrongARM SA-1110 Microprocessor Developer's Manual, Intel Corporations, 2001.
- [5] 이석, 문승빈, "리눅스 OS를 이용한 ARM CPU 기반 독립형 영상처리모듈 개발", 전자공학회논문지 제40권 CI편 제2호, pp.38~44, 2003. 03.
- [6] ADV601LC Ultralow Cost Video Codec Datasheet Analog Devices, Inc., 1999.
- [7] The Arm Linux Project, <http://www.arm.linux.org.uk/>
- [8] Alessandro rubini, Linux device drivers, 2nd ed., O'REILLY, 2001.06