

# CMOS 이미지 센서를 이용한 원격지 화상 감시 및 제어 시스템 구현

An Implementation of Remote Monitoring and Control System using CMOS Image sensor

최재우\*, 노방현\*, 이창근\*, 황희용\*

(Jae Woo Choi, Bang Hyun Ro, Chang Keun Lee, Hee Yeung Hwang)

\* 호서대학교 전기정보통신공학부 전자정보기술전공(전화:(041)548-5747, E-mail : cjwy@chol.net)

**Abstract :** We have designed embedded web server system and ported Linux operating system version 2.4.5 at our system. And then We implemented to control and monitor widely separated hardware and implemented to monitor widely separated image using CMOS image sensor HV7131B. Web server is the Boa web server with General Public License. We designed for this system using of Intel's SA1110 ARM core base processor and connecting input and output device at GPIO port of SA1110. Device driver of General purpose I/O for Embedded Linux OS is designed. And then the application program controlling driver is implemented to use of common gate interface C language. User is available to control and monitor at client PC. This method have benefit to reduce the Expenditure of hardware design and development time against PC base system and have various and capacious application against firmware base system.

**Keywords :** Embedded System, Embedded Web server, Embedded Web control, Embedded Web camera

## I. 서론

최근 유비쿼터스(Ubiquitous) 컴퓨팅환경의 인프라가 잘 갖추어진 국내의 인터넷자원을 활용하여 PC기반이 아닌 임베디드 시스템을 사용하여 생활의 편의를 제공해주는 서비스 요구에 대한 수요가 증가하고 있다. 예를 들어 핸드폰이나 PDA를 사용하여 영화를 감상하고 현재의 원하는 교통상황을 실시간적인 영상으로 확인하고 집안의 가전기기나 가스 밸브등의 상태를 확인하고 인터넷으로 TV를 집안의 VCR로 녹화하는 것 등이 그것이다[1]. 국가의 초고속 정보통신망 계획이 2005년까지 완료되면 유/무선 인터넷 통신의 QoS(Quality of Service)가 대폭 개선되어질 것이고 위와 같은 시스템에 대한 사용자의 요구는 더욱 늘어날 것으로 사료된다[2].

이러한 요구에 대한 시스템을 구현하는데 있어서 가장 고려 되어야할 점은 시스템제작시의 비용문제라 할 수 있다. 이에 임베디드 시스템에 고가의 상용화된 OS를 사용하지 않고 라이선스 비용이 필요없는 LINUX OS와 Boa Web server를 타겟보드에 포팅하여 원격지의 화상감시와 디바이스의 제어/감시가 가능하도록 하였다.

본 논문은 인텔사의 32비트 RISC 프로세서인 프로세서(SA1110)를 사용하여 웹 서버 시스템을 제작하고 하이닉스사의 CMOS 이미지센서 HV7131B를 보드에

인터페이스하여 사용자가 웹 브라우저만 있으면 원격지의 화상을 실시간으로 모니터링 할 수 있도록 하였다. 또한 LED와 스위치를 웹서버 보드에 연결하여 원격지의 디바이스 제어/감시를 검증하는데 사용하였다.

임베디드 웹 서버를 사용한 상용화된 제품들은 성능은 우수하더라도 운영체제 그리고 카메라 시스템과 영상 코덱 칩 등의 비용 때문에 가격이 고가일 수밖에 없다. 본 연구에서 구현된 영상감시 시스템은 영상의 품질과 초당 프레임 수가 많이 요구되지 않는 용도에 사용되는 것을 목적으로 영상획득 시 사용된 하드웨어는 이미지 센서만 사용하여 소프트웨어적으로 코덱 처리를 대신함으로써 비용절감에 비중을 두고 개발되었다. 또한 원격지의 디바이스 제어와 상태 감시를 위해 CGI 응용프로그램을 작성하여 디바이스드라이버에 접근하도록 하였다.

## II. 본론

### 1. 하드웨어 구성

인텔 SA1110 32비트 RISC 프로세서를 221Mhz에서 동작시켰으며 웹 서버의 기능을 위한 Ethernet 컨트롤러는 Cirrus Logic 사의 CS8900을 사용 하였고 16Mbyte의 인텔 스트라타 플래시와 32Mbyte의 SDRAM으로 구성하였다.

이미지센서와 원격지 제어/감시용 LED와 스위치는

GPIO 핀과 인터페이스 하였다. 표 1은 논문에서 사용한 GPIO 핀번호와 기능을 나타낸다.

기능	GPIO핀 번호	비고
Data[7:0]	GPIO[15:8]	픽셀 데이터 신호
MCLK	GPIO 7	(Master Clock)
ENB	GPIO 16	(Enable)
HSYNC	GPIO 2	수평동기신호
VSYNC	GPIO 3	프레임동기신호
LED1	GPIO 16	원격제어용
LED2	GPIO 17	원격제어용
Swich1	GPIO 18	원격감시용
Swich2	GPIO 19	원격감시용

표 1. 사용된 GPIO 핀과 사용한 기능.  
Table 1. Using GPIO pin and Function.

### 2. 소프트웨어 구성

Host PC의 작업환경은 윈도우XP OS에 cygwin이라는 리눅스 에뮬레이션 패키지를 설치하여 사용하였다. cygwin에서는 논문에 필요한 응용프로그램 즉, CGI응용프로그램, 이미지센서신호를 다른 형식으로 변환시키는 프로그램을 arm-linux-gcc로 크로스 컴파일하는 용도로 사용하였다[4].

타겟보드에 리눅스를 부팅시키고 리눅스의 컴파일된 이미지를 전송하기 위해 blob 부트로더를 포팅하여 사용하였다[3]. 부트로더는 JTAG를 사용하여 PC의 패러랠포트로 인텔스트라타 플래시 0번지에 다운로드 하였다. 리눅스 커널과 이미지화일은 부트로더에서 시리얼통신보다 속도가 개선된 tftp 프로토콜을 사용하여 Host의 tftp 서버는 SolarWinds사의 TFTPServer 공개버전을 사용하였다[5].

사용된 웹서버는 boa web 서버로서 ARM용 크로스 컴파일러로 컴파일하여 필요한 실행화일과 설정화일들을 호스트의 nfs 디렉토리를 마운트하여 타겟보드의 jffs2(journalling Flash file system2)로 되어있는 디렉토리에 카피하여 사용하였다. Jffs2 파일시스템은 Jffs와 같은형식의 파일 시스템이나 파일을 압축하여 저장한다는 점에서 좀더 저장 메모리를 효율적으로 사용할수 있다는 장점이 있다[6].

### 3. CMOS 이미지센서

사용된 이미지센서는 입력 클럭속도가 최대 15Mhz에서 동작가능하며 최대 640X480 (VGA)의 해상도를 가지며 센서픽셀신호 출력은 Bayer RGB 칼라 패턴이 D[7:0] 단자에서 나온다. 센서에서 나오는 신호중 HSYNC 단자는 수평동기신호로서 픽셀 데이터가 유효한 값일 경우에만 "High" 신호를 출력한다. 그리고 VSYNC 신호는 하나의 프레임이 모두 전송되고나면 "High"신호를 출력한다.

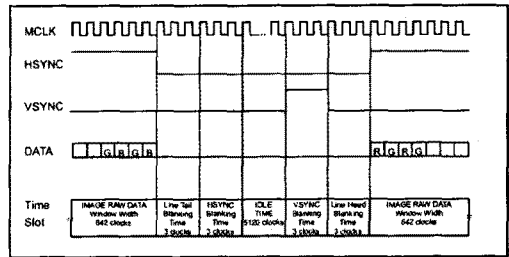


그림 1. HV7131B 이미지센서의 타이밍 차트.  
fig. 1. Timming chart of Image sensor HV7131B.

### 4. 이미지 센서의 픽셀신호 변환과정

이미지 센서의 픽셀 데이터는 MCLK(Master Clock)신호에 동기화되어 8비트단위로 출력 핀 D[7:0]에서 Bayer RGB 형식으로 출력된다. Bayer 영상형식은 칼라이미지를 디지털신호로 획득하는데 많이 사용되는 영상형식이다. 참고로 1970년대 Kodak사의 연구원 Dr. Bayer가 이미지센서에 의해 획득된 칼라영상이 모든 칼라로 재조립될수 있는 최적화된 RGB필터를 이미지센서에 처음으로 적용시켰다[7]. Bayer영상을 RAW RGB 영상형식으로 변환하고 다시 PPM(Portable PixMap) 영상형식으로 변환하여 최종적으로 HTML에서 지원하는 JPEG 영상형식으로 변환하였다.

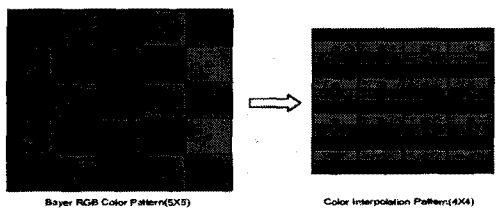


그림 2. Bayer RGB 칼라영상을 RAW RGB 칼라영상으로 변환되는 과정  
fig 2. Converting process of Bayer RGB color Image to Raw RGB color Image

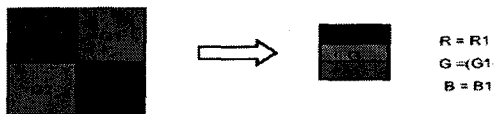


그림 3. Bayer RGB 영상과 RAW RGB영상신호 변환식  
fig 3. Converting Equation between Bayer color Image and Raw RGB color Image

Bayer RGB 형식을 RAW 형식으로 변환하는과정과 수식은 그림 2, 3과 같다.

변환된 RAW영상은 헤더정보를 가지고 있지 않기 때문에 최종적으로 JPEG영상으로 변환할 때 영상크기정보와 칼라의 개수 등의 정보를 수동으로 입력해줘야 하므로 다시 PPM 영상 형식으로 변환하였다. PPM 영상 형식의 헤더정보의 첫째 내용은 매직넘버를 의미하며 이것이 "P6"일 경우는 영상데이터가 바이너리형태로 저장되어 있음을 의미하고 "P3"일 경우에는 영상데이터가 ASCII Text 형태로 저장되어있음을 의미한다. 다음 정보는 영상의 크기를 의미하며 마지막 정보는 픽셀 당 밝기 값(intensity value)의 최대 값을 의미한다. 본 논문에서는 "P6 320X240 255"형태의 헤더정보를 사용하였다. 최종적으로 JPEG 영상으로의 변환은 cjpeg이라는 리눅스용 그래픽변환 프로그램을 arm-linux-gcc로 컴파일 하여 사용하였다. 참고로 cjpeg 은 RAW 이미지영상 형식은 지원하지 않는다.

### 5. 디바이스 드라이버

본 논문에서 사용된 디바이스 드라이버는 리눅스의 커널 디바이스 드라이버 작성법을 기초로 작성하였다[8]. 디바이스 드라이버는 순차방식이 아닌 이벤트방식으로 처리되어 나중에 커널에 삽입될 수 있는 모듈 프로그램이다. 모듈이 리눅스 커널에 적재/제거 되려면 insmod/rmmod 라는 리눅스 명령어를 사용하는데 이때 실행되는 init\_module()/cleanup\_module() 함수 안에서 register\_chrdev()/unregister\_chrdev() 함수에 의해 커널에 등록/제거된다. 디바이스 드라이버가 커널에 적재된 이후에 mkmod 명령을 사용하여 GPIO 디바이스 장치 파일을 /dev 디렉토리 밑에 생성시키면 사용자 메모리 공간상의 응용프로그램이 장치화일에 접근하여 읽고 쓰기가 가능해진다.

이미지센서를 구동하기위한 디바이스드라이버는 센서 출력 신호중 HSYNC와 VSYNC의 상태를 읽어내기 위한 is7131\_0.c 와 MCLK신호를 만들어 주고 픽셀데이터를 읽어내기위한 용도로 is7131\_1.c 파일을 구현하여 사용하였다. is7131\_0.c에서는 is7131\_0\_read()함수에서 이미지센서의 HSYNC, VSYNC 데이터를 읽어내는 동작을 기술해놓고 응용프로그램에서 HSNCA "high" 일때만 픽셀데이터를 is7131\_1.c의 is7131\_1\_read()함수를 사용하여 파일로 만들어 출력시키게 되면 image.bay(Bayer RGB 영상)이 만들어지게 된다. 이때 이미지데이터출력은 이미지센서의 MCLK에 의해 동기화되어 출력되며 MCLK 신호는 is7131\_1.c의 is7131\_1\_write()함수로 구현하였다. 또한 원격지의 LED와 switch on/off 상태를 제어/감시하기위해 gpio.c 에 gpio\_read()와 gpio\_write()함수를 구현하였다.

### 6. CGI 응용프로그램

CGI(Common Gateway Interface)란 웹서버와 외부프로그램간에 서로 데이터를 주고 받기 위한 표준규약으로서 이질적인 두 프로그램간에 데이터를 주고받기 위한 공통된 데이터 입출력 통로를 의미한다[9].

CGI에 사용되는 언어는 JAVA, perl, c, shell script 등 무엇이든 상관없지만 여기서는 디바이스 드라이버에 접근하는 코드가 포함되므로 저 수준 제어에 용이한 C언어를 사용하였다.

CGI 응용프로그램은 크게 디바이스 드라이버를 접근하는 부분과 HTML 코드를 printf()로 처리하여 웹브라우저에 정보를 표시하는 부분으로 구현되어있다. 즉, HTML 문서에서 전달한 LED점등데이터를 받아 GPIO 디바이스 드라이버를 통해 LED를 점등시키고 또한 스위치의 상태를 읽어내어 화면에 출력해주는 역할을 한다.

### III. 구현 및 실험

#### 1. 원격지 화상감시 실험

타겟보드에 이미지센서를 연결하여 드라이버 응용프로그램(im\_test.c)이 디바이스 드라이버(is7131\_0.c, is7131\_1.c)에 접근하여 센서의 data[7:0]에서 받은 Bayer영상을 파일(image.bay)로 저장한 후 이 bayer영상을 raw 영상으로 변환하는 Makeppm.c 프로그램이 수행되고 다시 ppm 영상으로 저장한다. 마지막으로 cjpeg 응용프로그램으로 최종 Image.jpg로 변환한다. 위의 모든과정을 반복해서 수행할수 있도록 bash shell script 로 만들어서 보드에서 수행시켜놓는다. 자바 스크립트를 사용하여 주기적으로 JPEG영상을 보여주는 기능을하는 cam.html에 사용자가 웹브라우저에서 접근하면 그림 5처럼 화상을 감시할수 있게 된다. 그림 4는 화상감시 실험을 위한 구성도이다.

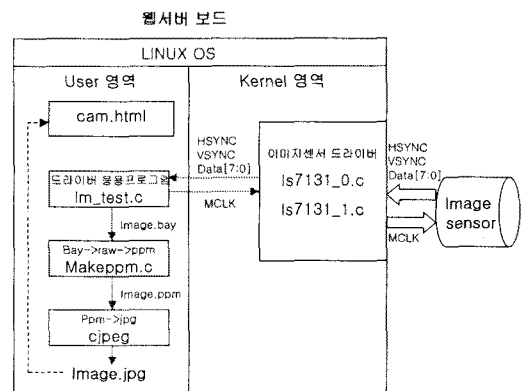


그림 4. 원격지 화상감시 실험 구성도

fig 4. block diagram of remote image monitoring experimentation

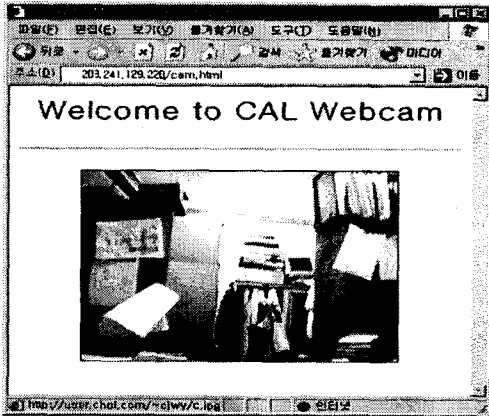


그림 5. 사용자 웹 브라우저상의 화상 감시화면  
fig 5. Image monitoring screen at User's Web browser

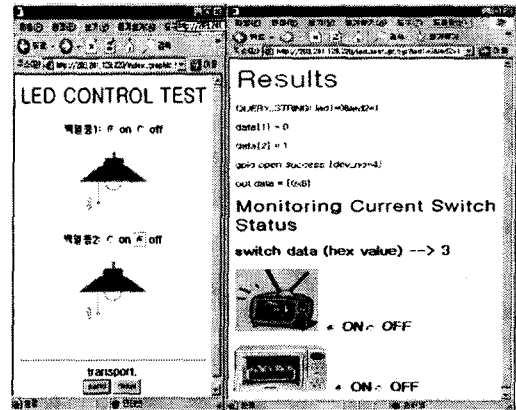


그림 7. 디바이스 제어/감시에 대한 화면  
fig 7. Device control and monitoring screen at User's Web browser

## 2. 디바이스 제어 및 감시실험

사용자가 그림 7과같이 웹브라우저에서 타겟보드의 주소를 입력하고 화면에 보이는 LED점등 데이터를 라디오버튼으로 선택해주고 "send"버튼을 누르면 CGI응용프로그램(led\_test\_gr.c)이 실행되어 get 방식으로 넘어간 인자가 디바이스에 출력된다. 또한 현 상태의 스위치상태를 미리구현한 디바이스드라이버를 통해 read()함수로 읽어내어 화면에 ON/OFF 표시를 그림8과 같이 라디오버튼으로 표시한다. 디바이스 제어와 감시를 위한 실험 구성도는 그림6과 같다.

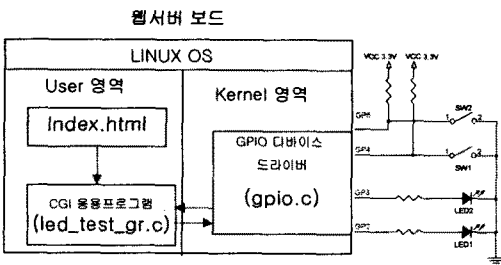


그림 6. 디바이스 제어/감시실험 구성도  
fig 6. block diagram of remote device monitoring and control experimentation

그림 7의 왼쪽 화면은 LED 디바이스의 점등값을 선택하는 html 문서에 의한 화면이고 "send" 버튼을 클릭하면 html에서 전달받은 인자를 사용하여 led에 신호를 출력하고 그때의 스위치 상태값을 읽어서 화면에 표시해 준다 (그림 7의 오른쪽 화면).

## IV. 결론

본 논문은 임베디드 웹서버 보드를 이용하여 원격지의 화상과 디바이스 제어/감시가 가능한 시스템을 구현 하였다. 하드웨어적으로 CMOS 이미지 센서만을 사용하여 영상을 획득하였으며 영상처리하는 모두 소프트웨어적으로 처리하였기 때문에 초당 프레임수가 많지 않은 응용에 사용가능하며 개발비용을 줄일 수 있다는 장점을 가지고 있다. 또한 CGI를 사용하여 웹브라우저에서 원격지에 있는 하드웨어 자원을 제어하고 상태를 감시할 수 있도록 하였다.

적은 비용으로 HA(Home Automation), FA(Factory Automation), ITS(Intelligent Transport System)에 분야 등에 널리 응용이 가능하며 펌웨어로만 개발하는 방법보다 리눅스가 가지고 있는 자원을 활용하면 개발시간을 줄일수 있다는 장점도 있다.

## 참고문헌

- [1] 홍성수, 조석재, "내장형 웹 서버 기술", 전기학회지 제 49권, 10호, 10, 2000.
- [2] 임기욱, "차세대 인터넷 서버 개발 계획", 정보처리학회지 제9권, 제1호, 1, 2002.
- [3] The LART Pages, <http://www.lart.tudelft.nl>
- [4] cygwin home page, <http://cygwin.com>
- [5] solarwinds home page, [www.solarwinds.net](http://www.solarwinds.net)
- [6] The Journalling Flash File System, version 2, <http://sources.redhat.com/jffs2/>
- [7] Color Correction for Image Sensors, Application note, Kodak, [www.kodak.com/go/imagers](http://www.kodak.com/go/imagers)
- [8] Linux device driver 2nd Edition, Alessandro rubini & Jonathan corbet, O'reilly, 2001.
- [9] <http://eekim.com/pubs/cgiwin/>