

CMOS 카메라 모듈 검사를 위한 네트워크 카메라

신원철*(선문대학교 대학원 제어계측공학과), 최병욱(선문대학교 제어계측공학과)

Network Camera for CMOS Camera Module Inspection

E. C. Shin(Control & Measurement Eng. , Sunmoon Univ.),
B. W. Choi(Control & Measurement. Eng. , Sunmoon Univ.)

ABSTRACT

In this paper, we developed a network camera for CMOS camera module inspection. The design, implementation details including embedded linux porting and CPLD logics, and performance of network camera are described. The network camera consists of SoC(S3C4530A), CPLD and CMOS image sensor. In order to image data of CMOS image sensor we designed capture logics on CPLD by using VHDL program. Embedded Linux such as uClinux is performed on the network camera to utilize development environment and TCP/IP protocol specification. The application is based on socket communication between GUI on PC and Embedded Linux based network camera. When JPEG compression is applied, the transmission speed was improved enough for this system to be used for an alternative of expensive CCTV or remote monitoring system in a power plant and uninhabited places.

Key Words : CMOS Camera module(CMOS 카메라 모듈), Embedded Linux(임베디드 리눅스), Network Camera(네트워크 카메라)

1. 서론

최근 SoC(System on a Chip) 기술과 디지털 영상 기술의 발전으로 인해 고성능, 저전력의 영상 디바이스가 출시되고 있으며, 모바일 폰과 같은 휴대용 장치에도 적용되고 있다. 이와 같은 영상 디바이스는 CMOS(Complementary Metal-Oxide Semiconductor) 이미지 센서가 주로 사용되고 있다. CMOS 이미지 센서는 기존의 많이 사용되던 CCD(Charge Coupled Device)와 달리 디지털 방식이기 때문에 영상 해독 속도가 빠르며, 크기가 작고, 소비전력도 적다. 과거에는 CMOS 이미지 센서가 CCD 이미지 센서에 비해 영상의 질이 매우 떨어졌기 때문에 많이 사용되지 않았지만, 최근 반도체 기술의 향상과 더불어 영상의 질이 획기적으로 향상되어 영상센서로서 각광 받고 있다.

본 논문에서는 위에서 언급한 CMOS 이미지 센서 검사를 위한 네트워크 카메라를 설계, 구현한다.[1] 네트워크 카메라는 MPU(Micro Processor Unit)와 CPLD(Complex Programmable Logic Device), CMOS 카메라 모듈로 구성된다. 또한 네트워크 카메

라를 효과적으로 제어하고 개발의 편의를 위해 운영 체제를 적용하였다. 본 연구에서는 운영체제로 RTOS (Real-Time Operating System) 또는 상용 임베디드 운영체제 대신 임베디드 리눅스를 사용하였다. 임베디드 리눅스는 일반 리눅스와 마찬가지로 오픈 소스이며, 강력한 네트워킹과 다양한 하드웨어 아키텍처를 지원한다. 또한 다양한 응용 프로그램 소스와 개발 툴, 라이브러리를 무료로 제공하기 때문에 개발비용을 절감하고 개발시간을 단축할 수 있는 이점이 있다.[2,3,4] CMOS 카메라 모듈로부터 출력되는 영상 데이터를 램에 저장하기 위해 CPLD를 사용하였으며, VHDL(Very high speed integrated circuit Hardware Description Language)를 이용하여 구현하였다. 램에 저장된 영상 데이터를 TCP/IP를 통해 PC로 전송하기 위해 리눅스 기반의 응용 프로그램을 개발하였다. 이 프로그램은 I²C 통신 프로그램과 GPIO(General Purpose Input Output) 핸들링 프로그램 그리고 네트워크 소켓 프로그램으로 구성된다. 그림 1은 시스템의 구성을 나타낸다.



Fig. 1 The system block diagram

2. CMOS 카메라 모듈 및 네트워크 카메라

2.1 CMOS 카메라 모듈

본 연구에서는 Pixel Plus의 PO1030KB CMOS 카메라 모듈을 사용하였다. 이 모듈은 이미지 센서의 크기가 1/4.5 인치이며, 컬러 필터와 마이크로 렌즈를 가지고 있다. 출력 포맷으로는 8bit YCbCr/YUV, 9bit Bayer RGB, 5:6:6 RGB, 12bit 8:8:8 RGB를 지원한다. 또한 연속으로 영상을 출력하는 비디오 모드와 기계적/전기적 shutter 기능을 지원하는 still image capture 모드를 지원한다. Frame Rate는 27MHz에서 최대 30fps까지 지원한다. PO1030 CMOS 카메라 모듈은 내부 레지스터의 값을 변경함으로써 이미지의 Gain, 밸런스, 출력 포맷 등을 변경할 수 있다. 내부 레지스터는 I²C를 통해서 변경된다.[5]

2.2 네트워크 카메라

네트워크 카메라는 CMOS 카메라 모듈로부터 받은 영상 데이터를 TCP/IP를 통해서 전송하기 위해 ARM7TDMI 코어 기반의 S3C4530A SoC를 사용하였다. 이 칩은 네트워크 기반의 시스템을 위해 설계된 MPU로써 이더넷 컨트롤러가 내장되어 있으며 I²C 인터페이스도 제공한다.[6]

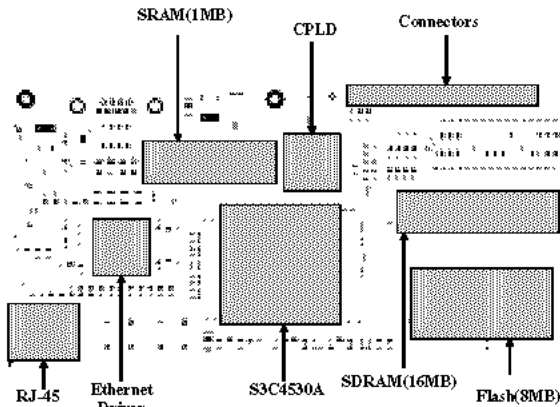


Fig. 2 The system layout of network camera

CMOS 카메라 모듈로부터 출력되는 영상 데이터를 램에 저장하기 위해 CPLD를 사용하였다. CPLD는 카운터와 버퍼, 디코더 등이 CMOS 카메라 모듈의 출력 포맷에 맞게 VHDL로 구현된다. 주변 장치로는 네트워크를 위한 RJ-45 포트와 이더넷 드라이버, 디버깅을 위한 시리얼 포트, 네트워크 카메라와 CMOS 카메라 모듈을 연결하기 위한 커넥터가 있다. 메모리는 커널과 사용자 응용 프로그램을 포함한 파

일 시스템이 저장되는 플래시 메모리와 리눅스가 동작하는 SDRAM 그리고 영상 데이터가 저장되는 SRAM이 포함되어 있다.

3. 네트워크 카메라 구현

3.1 CPLD 프로그래밍

CPLD는 CMOS 카메라 모듈로부터 출력되는 영상 데이터를 램에 저장하기 위해 사용된다. PO1030 CMOS 카메라 모듈의 출력 포맷을 그림 3과 같은 형태로 출력되도록 내부 레지스터를 설정하였다.

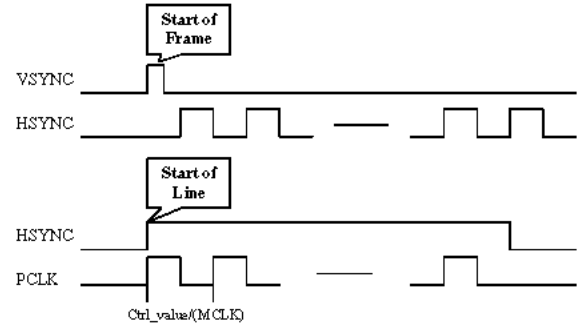


Fig. 3 The output format of CMOS camera module

VSYNC 신호는 영상 데이터에서 한 프레임의 시작을 나타낸다. HSYNC 신호는 프레임에 포함된 라인을 나타내는 신호이며, PCLK는 이미지의 한 픽셀을 나타내는 신호이다. PCLK 신호를 HSYNC 신호가 High일 때만 출력되도록 설정하였기 때문에 HSYNC 신호를 CPLD의 입력신호로 사용하지 않았다.

그림 4는 CPLD와 MPU 사이의 연결 관계를 나타낸다. CPLD는 MPU의 제어 신호에 따라 SRAM의 제어신호와 데이터 버스, 어드레스 버스를 스위칭한다. 즉 제어신호에 따라 SRAM의 입출력 버스를 MPU 또는 CMOS 카메라 모듈에 연결한다.

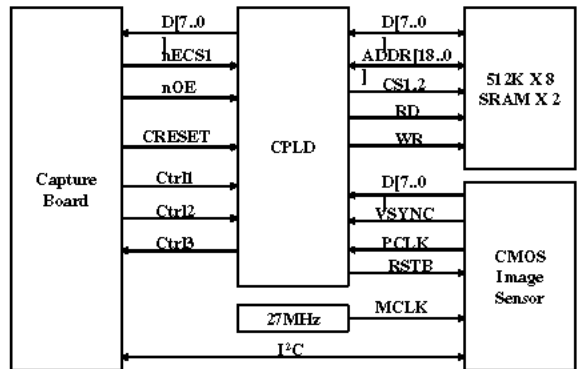


Fig. 4 Block diagram within CPLD

Ctrl1과 Ctrl2, CRESET는 MPU에서 출력되는 신호이고, Ctrl3는 MPU로 입력되는 신호이다. CRESET는 CMOS 카메라 모듈을 초기화하기 위한 리셋 신호로 RSTB와 연결된다. MPU(3.3V)와 CMOS 카메라 모듈(2.5V)의 동작 전압이 다르기 때문에 CPLD를 통

하여 연결하였다. Ctrl1은 SRAM에 입출력 버스를 MPU 또는 CMOS 카메라 모듈에 연결하기 위한 신호이다. High일 때는 MPU와 SRAM이 연결되며, Low일 때는 CMOS 카메라 모듈과 SRAM이 연결된다. 표 1은 SRAM과 MPU, CMOS 카메라 모듈간의 매핑 관계를 나타낸다. Ctrl2는 CPLD 내부의 구현된 카운터를 초기화하는 신호이다. Ctrl3는 한 프레임의 영상 데이터가 SRAM에 저장되었을 때 High에서 Low로 천이되는 신호이다.

Table 1 The mapping among SRAM, MPU and CMOS camera module

SRAM		MPU	CMOS
Data Bus		Data Bus	Data Bus
Addr Bus	Reset	Ctrl2	VSYNC
	Clock	nECS1	PCLK
RD		nOE	*
WR		*	PCLK
CS		nECS1	PCLK

CPLD는 버퍼와 디코더, 카운터로 구현되어 있다. 버퍼는 데이터 버스를 공유함으로써 일어나는 데이터의 충돌을 방지하기 위해 사용된다. 디코더는 MPU로부터 입력받은 제어신호를 해석하여 내부 제어신호를 만든다. 카운터는 SRAM의 어드레스 버스와 연결된다. 이는 영상 데이터를 순차적으로 저장하고 읽어가기 때문에 가능하다. 본 연구에서 사용한 CMOS 카메라 모듈의 출력은 YCbCr로 설정하였기 때문에 한 프레임의 영상 데이터 크기는 $640 * 480 * 2 = 600\text{KByte}$ 이다. 때문에 카운터는 20bit로 설계하였다. 카운터의 입력은 리셋과 클럭이고, 출력은 카운트된 값이다. Ctrl1의 신호에 따라 카운터의 리셋은 MPU의 Ctrl2 또는 CMOS 카메라 모듈의 VSYNC와 연결되며, 클럭은 MPU의 nECS1 또는 PCLK과 연결된다. 카운터의 출력은 SRAM의 CS와 어드레스 버스에 연결된다. 사용된 SRAM이 512KB이기 때문에 2개의 SRAM을 사용하였으며, 카운터 출력의 MSB를 CS로 사용하였다. 동작순서는 다음과 같다.

- 1) CRESET에 High->Low->High 순으로 신호를 출력한다. 즉 카메라 모듈을 리셋한다. 이것은 RST가 Low인 동안에 리셋이 된다.
- 2) Ctrl1신호를 Low로 만든다. SRAM의 제어 버스와 데이터 버스가 CMOS 카메라 모듈과 연결시킨다.
- 3) Ctrl1 신호가 Low 상태가 되며, 그림 5와 같이 VSYNC 신호가 Active Low 일 때부터 영상 데이터를 램에 저장한다.

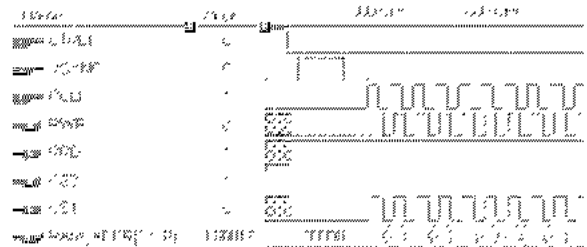


Fig. 5 The control signal when image data is stored into SRAM

4) 영상 데이터가 저장된 후 그림 6과 같이 VSYNC 신호가 다시 Active High가 되면 Ctrl3 신호가 High에서 Low로 천이된다.

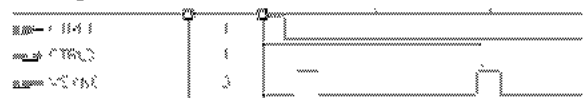


Fig. 6 The control signal when a frame of image data is stored into SRAM

5) MPU에서는 Ctrl1를 Low로 천이시킨 후 Ctrl3를 폴링으로 감시하다 Ctrl3가 Low가 되면 Ctrl1 신호를 High로 변경한다.

6) Ctrl2 신호를 그림 7과 같이 Low->High->Low로 만들어서 어드레스를 만드는 카운터를 리셋한다.

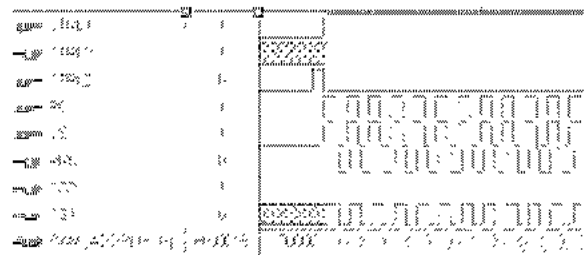


Fig. 7 The control signal when reading the image data stored in SRAM

- 7) SRAM에 저장된 영상 데이터를 읽는다.
- 8) Ctrl1 신호를 Low로 천이시켜 다시 영상 데이터를 SRAM에 저장한다.

3.2 임베디드 리눅스 포팅

리눅스는 기본적으로 32비트 프로세서인 i386에 처음으로 개발되어 현재는 메모리 관리 유닛(MMU)이 있는 여러 프로세서 아키텍처에 포팅되어 사용되고 있다. 그러나 본 논문에서 사용하는 S3C4530A 프로세서는 MMU가 없기 때문에 일반 리눅스 커널을 사용할 수 없다. 때문에 MMU가 없는 프로세서들을 위한 임베디드 리눅스인 uClinux를 네트워크 카메라의 운영체제로 사용하였다.[7]

리눅스 커널은 하드웨어 계층위에 존재하며 크게 프로세서와 관련된 부분과 관련 없는 부분으로 나뉘어진다. 일반적으로 포팅이라하면 프로세서와 관련된 부분을 수정하는 것을 말하며, 메모리 맵이나 디

바이스 드라이버 그리고 부트코드 등이 이에 해당된다.

본 논문에서는 개발된 시스템에 메모리 맵, 부트코드 등 프로세서 관련된 부분을 수정하여 포팅하였으며, 이더넷과 RS-232를 위한 디바이스 드라이버를 작성하였다.

3.3 영상 획득 프로그램

영상 획득 프로그램은 CMOS 카메라 모듈을 제어하여 영상 데이터를 SRAM에 저장하고, 저장된 영상 데이터를 TCP/IP를 통해 PC로 전송하는 프로그램이다. 이 프로그램은 uClinux에서 실행되며, 리눅스에서 크로스 컴파일러와 라이브러리를 이용하여 개발하였다.

영상 획득 프로그램은 TCP/IP 소켓 서버로서 CMOS 카메라 모듈을 제어하기 위해 I²C와 GPIO를 제어하는 기능이 포함되어 있다. 이 프로그램 리눅스가 부팅되면서 실행된다. 실행된 프로그램은 클라이언트의 접속을 기다리다, 클라이언트가 접속되면 CMOS 카메라 모듈을 초기화하고 클라이언트로부터 받은 명령에 따라 레지스터 값을 설정하고, "capture" 명령을 받으면 영상 데이터를 클라이언트로 보낸다.

3.4 영상 출력 프로그램

영상 출력 프로그램은 네트워크 카메라로부터 받은 영상 데이터를 출력하는 프로그램이다. 이 프로그램은 네트워크 소켓 클라이언트로서 CMOS 카메라 모듈을 제어하는 명령을 보내며, 네트워크로부터 받은 YCbCr 포맷의 영상 데이터를 RGB 포맷으로 변경한 후 비트맵으로 출력한다. 그림 8은 구현된 영상 획득 프로그램을 보여준다.

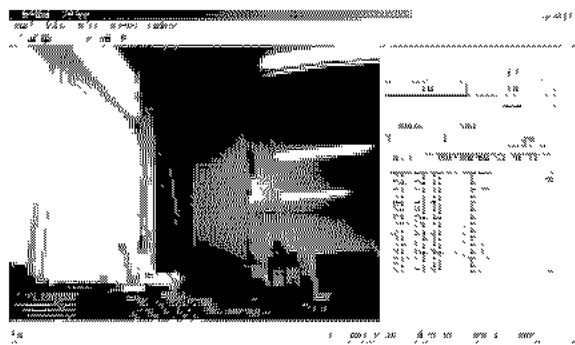


Fig. 8 The image display program

4. 네트워크 카메라 테스트

이상과 같이 개발한 네트워크 카메라를 2가지 조건에서 전송 속도를 테스트하였다. 첫 번째는 인터넷 망 속도에 따른 실험이며, 다른 하나는 저장된 이미지를 읽는 방법에 따른 실험이다. 인터넷 망에 따

른 실험은 100Mbps와 10Mbps에서 테스트하였다. 저장된 이미지를 읽는 방법은 저장된 이미지를 디바이스 드라이버(리눅스 버퍼 사용)를 이용하는 방법과 응용 프로그램에서 바로 읽어가는 방법으로 테스트하였다. 영상 데이터를 네트워크를 통해서 전송하기 전까지 걸리는 시간은 다음과 같으며, 실험 결과는 그림 9와 같다.

- 이미지 크기 = 해상도 * 픽셀당 바이트 수 = 640*480*2byte = 600Kbyte
- 픽셀 클럭 주기 = 메인 클럭 / 3 분주 = 27MHz / 3 = 9MHz
- 프레임/초(CMOS -> SRAM) = 600Kbyte/9MHz = 69ms
- 프레임/초(SRAM -> MPU) = 600Kbyte * 60ns = 55ms
- 네트워크 전송전 까지 걸린 시간은 = 69ms + 55ms = 124ms

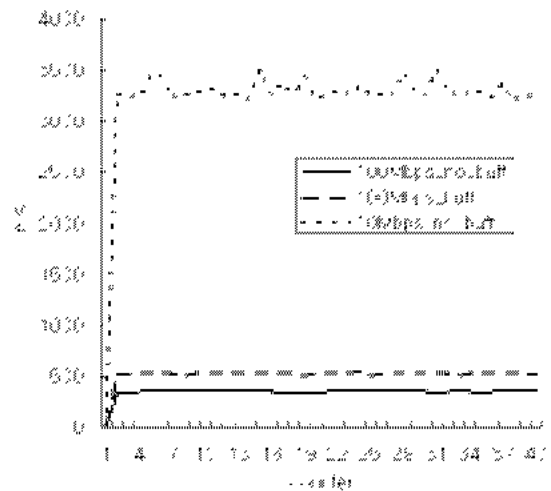


Fig. 9 The result of testing network camera

그림 9처럼 네트워크 망에 따라 전송 속도가 많이 차이가 발생하였으며, 리눅스 디바이스 드라이버를 이용 여부에 따라서도 전송속도 차이가 발생하였다.

5. 결론

본 논문에서는 CMOS 카메라 모듈 검사를 위한 네트워크 카메라의 설계와 구현에 대하여 기술하였다. 본 시스템은 네트워크를 통해서 영상 데이터를 전송하기 위해 네트워크 시스템 솔루션을 위해 설계된 ARM7TDMI 코어 기반의 S3C4530A SoC를 사용하였다. 또한 영상 데이터를 SRAM에 저장하기 위해 CPLD를 이용하였으며, 이를 VHDL를 이용하여 구현하였다. 개발된 네트워크 카메라의 운영체제로 임베디드 리눅스를 이용함으로써 개발 시간을 단축하고

이 운영체제가 제어 분야 적용 가능성을 보였다.

본 시스템은 CMOS 카메라 모듈 검사를 위해 개발된 시스템이기 때문에 영상 데이터의 용량이 커서 전송 속도가 약 3초/프레임 밖에 되지 않는다. 그러나 JPEG 압축 알고리즘을 적용할 경우 전송속도가 향상됨으로써 고가의 CCTV의 대안으로 활용될 수 있을 뿐만 아니라 원거리에 있는 발전소와 같은 시설의 원격 모니터 장비로도 이용될 수 있다.

후기

본 논문은 과학기술부/한국과학재단지정 선문대학교 공조기술 연구센터에서 지원하였습니다.

참고문헌

1. A.Filippov, "Reconfigurable High Resolution Network Camera", Proc, 11th Annual IEEE Symp, on Field-Programmable Custom computing Machines, 2003
2. A.Lennon, "Embedded Linux", IEE Review, pp33-37, May 2001
3. S.Hong, "Embedded Linux Outlook in the Post Industry", Proc. 6th Int. Symp. on Object-Oriented Real-Time Distributed Computing, 2003
4. A.Filippov, "Using Embedded Linux in a reconfigurable high-res network camera", LinuxDevices, www.linuxdevices.com, 2003
5. PO1030 Data Sheet, PixelPlus, 2003
6. S3C4530A User's guide, Samsung, 2001
7. uClinux Project, www.uclinux.org/description