

Dual CPU 기반 임베디드 웹 카메라 스트리밍 서버의 설계 및 구현

*홍진기, 문종려, 백승걸, 정선태
송실대학교 정보통신전자공학과
e-mail : uzze@syscon.ssu.ac.kr

Design and Implementation of A Dual CPU Based Embedded Web Camera Streaming Server

*Jin-Ki Hong, Jong-Ryeo Moon, Seung-Kul Paek, Sun-Tae Chung
Dept. of Electronic Eng., Soongsil University

Abstract

Most Embedded Web Camera Server products currently deployed on the market adopt JPEG for compression of video data continuously acquired from the cameras. However, JPEG does not efficiently compress the continuous video stream, and is not appropriate for the Internet where the transmission bandwidth is not guaranteed. In our previous work[1], we presented the result of designing and implementing an embedded web camera streaming server using MPEG4 codec. But, the server in our previous work did not show good performance since one CPU had to both compress and process the network transmission. In this paper, we present our efforts to improve our previous result by using dual CPUs, where DSP is employed for data compression and StrongARM is used for network processing. Better performance has been observed, but it is found that still more time is needed to optimize the performance.

I. 서론

'임베디드 웹 카메라 서버'는 카메라와 인터넷 접속을 갖는 임베디드 시스템으로, 접속된 클라이언트에게 카메라가 획득한 영상데이터를 전송하여, 클라이언트가 이 영상 데이터를 볼 수 있게 한다. 임베디드 웹 카메라 서버는 현재 시장에 많은 제품이 출시되어 있으며[9,10], 이에 대한 관련 연구도 많이 이루어지고 있다[7,8]. 클라이언트의 웹 접속은 HTTP를 이용하는 것이 보편적이기 때문에, 현재 판매되고 있는 대부분의 임베디드 웹 카메라 서버에는 HTTP 프로토콜을 처리하는 간단한 웹 서버 기능이 탑재되어 있다. 또한, 카메라에서 획득한 동영상은 프레임별로 JPEG 으로 압축되어 파일로 저장되고, 클라이언트가 HTTP URL로 계속 해당 JPEG 파일을 요청하면 웹서버가 이를 전송하며, 클라이언트는 웹브라우저로 이를 계속 재생하는 동작 구조로 구현되고 있다.

그런데, 현재의 임베디드 웹 카메라 서버는 실시간 동

영상을 JPEG 으로 압축하기 때문에 인코딩된 영상 데이터의 양이 많아 전송 대역폭이 보장되지 않는 현재의 인터넷에서의 전송에 불리하다. 따라서, 동영상을 MPEG, H.263 등의 동영상 코덱으로 인코딩하여 전송 대역폭을 줄이는 것이 필요하다. 본 저자들의 이전의 연구[1]에서는 한개의 CPU (StrongARM)을 이용한 임베디드 웹 카메라 서버의 설계, 구현 및 성능 평가 결과를 소개하였다. 그러나, 한 개의 CPU에서 동영상 압축 및 영상 전송까지 처리하여야 하는 데, 동영상 코덱을 이용한 영상 압축에 많은 시간이 소요되어 충분한 성능을 얻을 수 없었다.

본 논문에서는 영상 압축에는 DSP 칩을 별도로 사용한 이중 CPU 기반 임베디드 웹 카메라 스트리밍 서버를 설계하고 구현한 결과를 기술한다. 영상 압축 전용 프로세서 사용도 고려해 볼 수 있으나, 이 경우 특정 동영상 포맷으로 제한되어 차후 동영상 코덱의 개선이나, 추가적인 동영상 스트림 조작(동영상 전송 에러 제어 등을 위한 정보 추가 등)이 어려워지므로, 영상 압축에 DSP 칩을 사용하였다.

윈도우즈에서는 동영상의 스트리밍 전송을 지원하기 위해 ASF 포맷을 규정하고 있으며, ASF 로 포맷된 멀티미디어 데이터의 스트리밍 재생을 윈도우즈 미디어 플레이어에 지원하도록 제공하고 있다. 따라서, ASF 로 포맷된 영상 스트림은 윈도우즈 웹브라우저에서 스트리밍 재생이 가능해진다.

본 논문에서 설계·구현한 임베디드 웹카메라 스트리밍 서버는 카메라에서 획득한 동영상을 msmpeg4 로 인코딩하고 이를 ASF 로 변환하여, 접속 요청한 클라이언트에게 HTTP 로 스트리밍 전송하여, 클라이언트 웹 브라우저에서 실시간 스트리밍 재생이 가능하도록 지원하고 있다.

본 논문의 구성은 다음과 같다. 2절에서는 설계 및 구현 내용이 기술되며, 3절에서는 결론 및 향후 연구 방향이 주어진다.

II. 설계 및 구현

2.1 시스템의 하드웨어 설계 및 구현

1) 전체 시스템 H/W 구성

본 논문에서 구성한 임베디드 웹 카메라 서버의 하드웨어 구성은 크게 두 부분으로 나누어져 있다. 영상 처리 부분과 웹 스트리밍 서버의 부분으로 구성되어 있는데 전체 하드웨어 구성도는 그림 1과 같다.

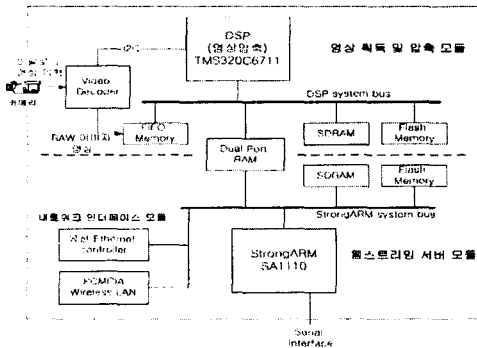


그림 1. 하드웨어 구성도

영상 처리부는 외부의 아날로그 CCD 카메라를 통해 들어오는 영상을 디지털로 변환해주는 video decoder와 영상을 MPEG4로 압축하기 위한 DSP, 그리고 video decoder와 DSP 사이에 영상데이터를 전달해주는 임시 버퍼인 Video FIFO 메모리로 구성된다.

웹 스트리밍 서버부는 유선 이더넷과 무선 랜의 네트워크 부분과 웹 스트리밍 서버 역할을 할 StrongARM과 32M의 SDRAM, 16M의 flash 메모리로 구성되어 있고[5], DSP와 StrongARM의 중간에 서로간의 데이

터를 공유할 수 있는 dual-ported RAM으로 구성된다.

2) 영상 처리부

영상을 압축하기 위해서는 강력한 임베디드 프로세서가 필요한데 보다 더 좋은 성능 구현을 위해서 TI DSP TMS320C6711을 사용하였고[11], DSP의 운용 및 영상 압축 코드가 저장될 1MB의 ROM, DSP 메인 메모리로 SDRAM 16MB를 사용하였다.

영상 입력은 외부의 아날로그 CCD 카메라를 통해 입력 받기 때문에 영상 압축을 위해서는 이를 디지털 영상으로 바꾸어 줄 video decoder가 필요하게 된다. video decoder로는 philips의 SAA7114H를 사용하였다 [4]. SAA7114H는 보편적으로 많이 쓰이는 SAA7111의 확장버전으로 내부에 scaler가 있어 원하는 크기의 영상을 가져 올 수 있는 이점이 있다.

SAA7114H의 제어를 위해서는 I2C 통신이 필요한데, 본 논문에서 구현한 DSP에는 I2C를 지원하지 않기 때문에 DSP의 I/O port를 이용하여 I2C를 에뮬레이션하여 제어하였다.

SAA7114H의 출력은 27MHz의 클럭에 동기화되어 초당 60field의 영상(NTSC의 경우)을 계속 출력하는 능동소자이기 때문에 같은 능동소자인 DSP와 데이터 교환을 하려면 중간에 버퍼 역할을 해줄 수 있는 메모리가 필요하다. 그리하여 FIFO 구조로 384KB의 용량을 갖는 AL422B 메모리를 사용하여 두 소자 간에 버퍼 역할을 하게 설계되었다[12].

그러나 이러한 구조의 부작용으로 나타날 수 있는 것이 video decoder에서 FIFO 메모리에 영상을 쓰는 도중 DSP에서 영상을 읽어 오게 되면 이전 영상과 나중 영상이 겹치게 되어 화면에 줄이 간 듯 보이게 되는 찢김 현상(tearing artifact)이 나타날 수 있게 된다. 이러한 현상을 제거하기 위해 데이터를 읽고 쓰는 것이 있어서 서로 간에 동기를 맞추어주어야 한다.

아래의 그림은 FIFO의 read, write 타이밍을 보여주고 있는데 SAA7114H는 영상신호와 관련하여 여러 가지 신호를 내어 주는데 이중 영상 데이터가 IDQ(output data qualifier)에 동기화되어 출력된다. SAA7114H의 IGP1(Image port general purpose signal 1) 출력에 field ID를 출력하게끔 레지스터를 설정하고 이를 IDQ와 OR 연산을 하여 FIFO에 write를 하여 아래 그림과 같이 한 필드에 대해서만 FIFO 메모리에 쓰고 또한 SAA7114H의 IGP1 출력을 DSP의 외부 Interrupt pin으로 연결해 even field의 하강 에지 이벤트 발생시 DSP의 메인 메모리로 영상을 DMA하게 끔 DSP의 EDMA(Enhanced DMA)를 설정해 주어 read, write 동기를 맞추었다.

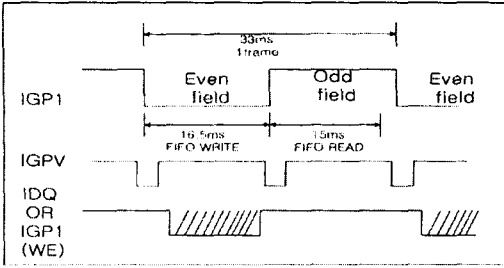


그림 2. FIFO read/write timing

DSP와 StrongARM의 중간에 서로간의 데이터를 공유할 수 있는 dual-ported RAM은 IDT사의 IDT70v27 2개를 사용하여 128KB의 32bit data width로 구성하였다[13]. Dual-ported RAM은 2개의 방향에 대해 서로 독립적인 어드레싱을 제공하는 장치인데 이를 이용해 DSP와 StrongARM에서 각각의 메모리에 맵핑을 하여 두 CPU의 공유메모리 역할을 해주게 된다. 이 장치는 하드웨어적인 세마포어와 인터럽트를 제공하는데 하드웨어적인 세마포어를 이용하여 서로간에 데이터의 일관성을 유지할 수 있기 편리하고, 또한 메모리의 최상위 번지에 데이터를 쓰는 경우 반대편에 인터럽트를 발생시키는 특징이 있어 데이터의 동기화에 용이하다는 장점이 있으나 값이 다른 메모리에 비해 상대적으로 비싸다는 단점도 있다.

3) 웹 스트리밍 서버부

HTTP 프로토콜 처리 및 스트리밍 전송 등의 네트워크 처리를 위해 인텔의 StrongARM을 사용하였다. 네트워크는 유·무선 랜을 모두 지원하는데 무선 랜은 PCMCIA 및 CF 타입의 wireless LAN 카드를 지원하도록 설계되었고, 유선 랜은 Cirrus Logic 의 CS8900A를 사용하여 10M ethernet을 구성하였다. StrongARM에서는 외부 I/O에 대해 DMA를 지원하지 않기 때문에 인터럽트를 이용한 I/O 모드로 설계되었다. 외부 터미널을 위한 com port 1개와 운영체제 및 프로그램 저장을 위해 16MB의 플래시 메모리와 메인메모리와 RAM 파일 시스템으로 쓰일 SDRAM 32MB로 구성되어 있다.

2.3 시스템 동작 흐름

그림3은 영상이 들어와서 네트워크를 통해 나가기까지의 시스템 동작 흐름을 보여주고 있다.

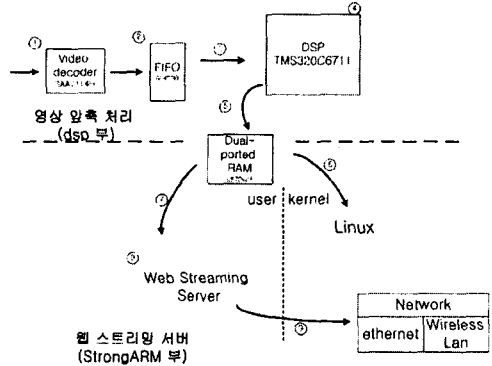


그림 3. 임베디드웹카메라서버의 시스템 동작흐름도

각 부분별로 설명을 하면

- ① 외부 카메라로부터 아날로그 영상 입력
- ② 320x240의 디지털 영상을 even field 만 FIFO 메모리로 전송
- ③ 영상을 FIFO 메모리로부터 EDMA를 이용하여 가져온다.
- ④ 영상을 MPEG4로 압축
- ⑤ 압축된 영상을 dual-ported RAM에 쓴다.
- ⑥ 인터럽트를 발생시켜 영상이 들어온 것을 Linux에 알림
- ⑦ dual-ported RAM에서 영상데이터를 mmap을 이용하여 램파일 시스템에 파일로 저장
- ⑧ 파일에서 스트림을 읽어 패키징한 후 전송
- ⑨ 네트워크를 통해 전달한다.

2.4 소프트웨어 설계 및 구현

1) 설계 고려 사항 및 임베디드 웹 카메라 스트리밍 서버의 S/W 구조

본 논문에서 구현하는 임베디드 웹 카메라 스트리밍 서버는 리소스 제약적인(작은 CPU 성능, 적은 메모리 용량, 없는 하드디스크 등) 임베디드 시스템 환경에서 영상 획득, 영상 압축, HTTP 프로토콜 처리, 스트리밍 전송, 클라이언트 접속 관리 등의 기능을 지원하여야 하는 시스템이다.

여러 기능들을 크게 두 부분으로 나누면 영상획득 및 영상압축 처리와 HTTP 프로토콜 처리 및 스트리밍 전송의 부분으로 나눌 수가 있다. 영상처리부분은 DSP에서, 스트리밍 전송은 StrongARM에서 각각 독립적으로 처리를 하게 하여 2개의 CPU에서 독립적인 일을 하는 방식으로 설계하였다.

StrongARM에는 네트워크 지원을 위해 임베디드 리눅스를 OS로 사용하였고, 또한 하드디스크가 없으므로 프로그램 및 영상 저장을 위해 램파일 시스템과 16M

의 플래쉬 메모리에 JFFS2 파일 시스템을 구축하였다.

2) 영상처리부의 소프트웨어 구조

DSP부의 소프트웨어 구조는 영상을 압축하는 메인 쓰레드와 2개의 인터럽트 루틴으로 구성되어있다. 인터럽트 루틴 중 하나는 영상의 field가 바뀔 때 발생하는 인터럽트로 이때 FIFO 메모리에서 메인 메모리로 RAW 영상을 DMA를 통해 가져오게 되고, 다른 인터럽트 루틴은 StrongARM에서 Dual-ported 메모리를 읽어 갈 때 발생하는 인터럽트로 인터럽트 발생시 DSP에서 압축된 영상을 메인 메모리에서 Dual-ported 메모리로 역시 DMA를 통해 복사를 하게 된다. RAW 영상을 가져올 때와 압축된 영상을 쓸 때, 메인 쓰레드에서는 영상 데이터에 동기화를 맞추기 위해 인터럽트 루틴에서 전송이 끝날 때까지 기다리게 된다.

3) 웹 스트리밍 서버부의 소프트웨어 구조

DSP부에서는 영상을 MPEG4로 압축하여 dual-ported RAM에 써 넣어 주는 역할을 하고, StrongARM부의 소프트웨어는 dual-ported RAM에서 영상을 가져와 공유파일에 쓰는 파일 쓰레드와 그 외의 기능을 하는 서버 쓰레드의 2개의 쓰레드 구조로 구성하였다. StrongARM 부에서의 소프트웨어 구조는 다음 그림 4와 같다.

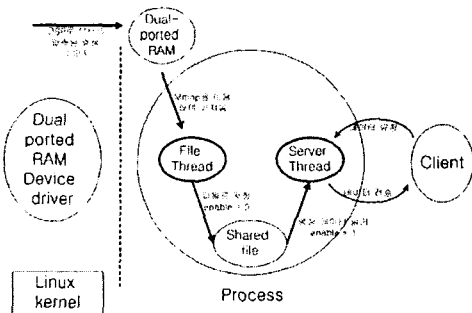


그림 4. 임베디드 웹 카메라 서버의 소프트웨어 구조

웹 스트리밍 서버부의 2개 쓰레드 사이의 데이터 교환이나 서버 쓰레드의 구조는 본 저자들의 이전의 연구[1]를 참고하기 바란다.

III. 결론 및 향후 연구 방향

본 논문에서는 DSP와 StrongARM 프로세서를 이용하여 Dual CPU 기반의 임베디드 웹 카메라 스트리밍 서버 시스템을 설계하였고 소프트웨어적인 영상 인코딩과 Client에 스트리밍 서비스를 하였다.

현재, 전체 시스템의 H/W 제작 및 초보적인 S/W 가 구현되어 있고, 동작은 잘 되고 있음을 확인하였다.

320x240의 영상을 15frame/s 정도로 DSP에서 인코딩을 하여 만족할만한 결과는 아니지만 이전[1]의 시스템보다 좋은 결과를 얻었다. 향후 H/W 설계의 개선과 DSP에 좀더 최적화된 영상 압축 코덱을 개발하여 인코딩 성능을 높이면 더욱 향상된 결과를 얻을 수 있을 것이다.

참고문헌

- [1] 이울삼, 백승걸, 정선태, "Design and Implementation of An Embedded Web Camera Streaming Server", 2002년도 대한전자공학회 추계학술대회 논문집, pp. 17-20
- [2] FFmpeg Homepag <http://www.ffmpeg.org>
- [3] A.Rubinei & J.Corbet, *LINUX DEVICE DRIVERS*, O'REILLY, pp 5-120 June 2001
- [4] SAA7114H datasheet, philips semi, 2000 <http://www.semiconductors.philips.com/>
- [5] SA1110 Users Guide, Intel, 1999. <http://developer.intel.com>
- [6] Daniel P. Bovet, *Understanding The Linux Kernel*, O'REILLY, pp 68-100, October 2000
- [7] Y.Kakimura, et al., "A Real-Time MPEG1 Ethernet Camera System", IEEE, 1999
- [8] Mi-Joung Choi, Hong-Taek Ju, Hyun-Jun Cha,Sook-Hyang Kim, Won-Ki Hong, "An Efficient Embedded Web Server for Web-based Network Element Management", IEEE, 2000
- [9] AXIS Homepage <http://www.axis.com>
- [10] WEBGATE Homepage <http://www.webgateinc.com>
- [11] TMS320C6711 datasheet, TI, 2001 <http://www.ti.com>
- [12] AL422B, datasheet, Averlogic, 2001 <http://www.averlogic.com>
- [13] IDT70v27, datasheet, IDT, 2001 <http://www.idt.com>