임베디드 리눅스 시스템 기반 실시간 원격 모니터링 기법 연구

A Study on the Real Time Remote Monitoring Technology based on Embedded Linux System

홍항설[1], 김성환[2], 박장현[3]

목포대학교 제어시스템공학과
E-mail: [1]hhseol@mokpo.ac.kr, [2]shkim@mokpo.ac.kr [3]jhpark72@mokpo.ac.kr

ABSTRACT

The Embedded Linux System has been developed as a system which can be used with a processor of low efficiency and small-sized memory. Unlike the usual Linux and Windows web server, it has some limitations in the install of application programs, compatibility and scalability when transferring data through web server in real-time.

In this paper, we present a real-time remote monitoring system which is very useful to the embedded linux system. The presented system use Java Script without the additional programs at the Embedded Linux System web server and confirm the efficiency of the system through the existing real-time remote monitoring techniques.

Key Words : Embedded System, Linux, Remote Monitoring, Web Server, Java Script

1. 서 론

최근 초고속 인터넷 기술의 확산으로 인터넷을 기반으로 하는 새로운 시장 창출이 다각화되고 있다. 여기에 전자 정보통신 기술 분야에서는 많은 제품이 임베디드 시스템으로 구현되고 임베디드 장치는 유선과 무선을 통해 네트워크로 연결되고 있다. 임베디드 시스템은 소형화, 경량화 되면서도 고성능의 다양한 기능을 제공하고 있으며, 많은 사람들이 웹을 통하여 데이터를 전송하고 이 데이터를 단말기에서 소리, 영상 또는 문자로 변환 하여 자신들이 원하는 정보를 얻고 있다.[4][5][6][9] 특히 원격지에서 웹 브라우저를 통한 실시간 영상 전송은 인적·물적 자원을 보호하기 위한 장비로서 원격 감시시스템이나 한 가정에 한 대씩의 로봇 보급이 국가적인 목표가 되고 있는 지금 로봇을 통한 영상 전송은 사용이 필수 요소가 되어가고 있다.[1]

기존의 웹 서버를 통한 캠의 실시간 영상 전송은 많이 보편화 되어있지만, 임베디드 시스템을 바탕으로 영상을 전송하는 방법은 시스템의 응용 프로그램 적재의 제한과 호환성 및 확장성 등의 제약사항이 발생한다. 기존에 제시된 웹 서버를 통한 영상 전달 방식들은 부가적인 프로그램을 필요로 하거나 여러 웹 브라우저 환경 변화에 대한 호환성의 문제점을 가지고 있다.[2][8]

본 논문에서는 임베디드 시스템의 제한된 자원을 수용하면서, 원격지의 웹 브라우저를 통해 영상을 실시간으로 전송하는데 있어서 기존에 연구된 실시간 영상 전달 방식을 고찰하고, 임베디드 리눅스 시스템 환경에서 가장 적합한 실시간 영상 전송을 통한 원격 모니터링 기법을 제안 한다.

2. 원격 모니터링 시스템 구성

2.1 시스템 개발환경

그림1은 임베디드 리눅스 시스템 상에서 CMOS Camera를 이용한 실시간 원격 모니터링을 위한 기본 구성도이다. Embedded System, Host PC(Linux), JTAG Cable, RS232 Serial Cable로 구성된다.

JTEG Cable과 RS232 Cable은 Host PC와 임베디드 시스템 사이에서 데이터를 주고받을 수 있는데 minicom이라는 통신터미널과 Jflash Xscale 프로그램을 통해 Flash Memory를

Host PC에서 제어 할 수 있도록 한다. 참고로 Flash Memory에는 bootloader, kernel, ramdisk, usr(jffs) 이미지가 저장된다.

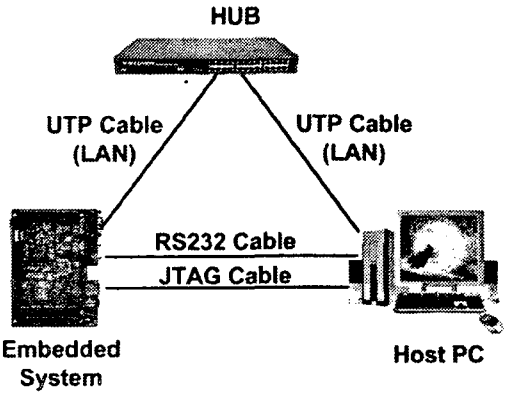

그림 1 개발 환경 구성도

2.2 임베디드 시스템의 구성

Embedded System은 메인 프로세서로 ARM계열의 인텔의 PXA255 프로세서를 사용하였다. PXA255는 XScale Core로서 400MHz로 동작하고, UARTs, USB, DMA 제어장치 등을 포함한다. 그림 2에서 보듯이 외부 Memory로는 Flash Memory(32M)와 SDRAM Memory(128M)이 사용되었다. 이미지 정보를 capture하기 위한 카메라로 CMOS image sensor와 CIS CAMERA 코어로 FPGA Cyclone가 사용되었다.[3]

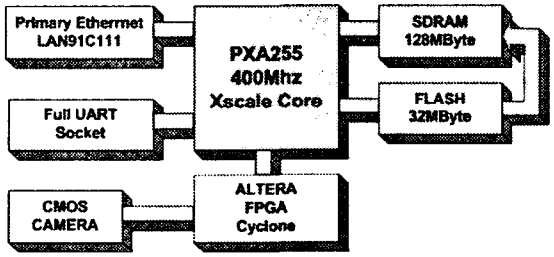

그림 2 임베디드 시스템 구성도

임베디드 시스템을 구동하기 위해서는 적어도 3가지 구성요소가 필요하다. 첫 번째로 Boot Loader는 하드웨어를 초기화 하고 Kernel image를 SDRAM에 올린 후 실행 권한을 Kernel에 넘기는 역할을 하는 프로그램이다. 두 번째로는 OS인데 Linux Kernel image는 OS의 핵심 프로그램이다. 세 번째 Ramdisk는 Kernel에서 사용할 File System이다. 그 밖에 Embedded System을 보다 효율적으로 사용하기 위해 User File System을 추가할 수 있다. User File System은 유틸리티나 data file 등을 위해 추가적으로 사용할 File System이다.

영상정보는 CMOS Image Sensor를 이용하여 획득하고, 획득한 영상은 FPGA Cyclone을 통하여 처리되고 저장된다. FPGA Cyclone에는 영상 데이터를 저장하기 위한 SDRAM이 구성되어 있다. 카메라로부터 영상을 읽어 SDRAM에 데이터를 저장하고 한 프레임이 다 읽었을 때 PXA255로 인터럽트를 보내는 동작을 반복한다. 디바이스 드라이버에서 인터럽트를 받으면 이 코어로 접근하여 안에 있는 SDRAM에 영상 데이터를 읽어와 디렉토리에 이미지를 일정 시간 간격으로 저장하게 된다. 저장된 영상은 웹 서버를 통하여 원격지의 Client에게 전송되며, 사용자는 웹 브라우저를 이용하여 영상을 통한 원격 모니터링을 수행하게 된다.[3][10]

많은 임베디드 웹 서버가 있지만, 본 연구에서는 웹 서버로 덩치가 작은 goahead를 사용하다. goahead는 임베디드 시스템용으로 디자인되어 있어 모듈이 작고 적은 양의 RAM으로도 운용이 가능한 특성을 갖는다. 실행 프로세스가 1KB 정도로 작은 웹 서버로 기본적인 자바스크립트는 지원하나 PHP 등 웹 프로그래밍 언어는 전혀 지원하지 않는 단점이 있다. 그래서 GNU 컴파일러에 적용되는 ANSI C 언어를 이용하여 CGI 인터페이스를 구현 하였고 이는 ANSI 표준을 따르기 때문에 이식능력이 뛰어나고 충분한 성능을 보장한다.[7]

3. 기존의 영상정보전달방법

영상전송 방식은 서버에서 제공하는 정보가 스트림(동영상)인가, 이미지인가에 따라 영상 스트림과 영상 이미지 전송으로 구분한다.

영상 스트림 전송 방식은 동영상을 전송하므로 끊김이 없는 부드러운 영상을 제공하지만, 동영상으로 인코딩하는 시간으로 빠른 응답시간을 위해 추가적인 하드웨어가 필요하다. 실시간 동영상을 사용자의 웹 브라우저에게 보여주기 위해서는 웹 브라우저에 윈도우즈 미디어 플레이어와 같은 외부프로그램이 설치되어야 한다.

영상 이미지 전송의 경우 서버에서 얻은 이미지를 계속 교체하여 마치 동영상과 같은 효과를 내는 방법으로, 영상 스트림 전송 방식과는 달리 추가적인 하드웨어나 외부프로그램 없이 이미지를 웹 브라우저에게 전달 할 수 있다. 영상 이미지 전송 방식은 크게 Client Pull 방식과 Server Push 방식으로 구분한다.

3.1 Client Pull 영상 전송

Client Pull 방식은 웹 브라우저를 사용하는 사용자가 영상정보를 얻고자 할 때, 영상정보의 요청을 주기적으로 웹 서버에게 요청하여

사용자에게 제공해주는 방식이다. Client Pull 방식을 구현하는 방법은 다음과 같이 3가지 형식으로 구분될 수 있다.

•Meta Tag를 통한 Refresh - 설정된 시간에 따라 주기적으로 HTML 페이지를 갱신하는 방법으로 메타 태그는 사용자에게 보여 지는 정보가 아니라 HTML 문서의 메타정보를 표현하기 위한 태그이다.
•Java Script를 통한 Refresh - 설정된 시간에 따라 주기적으로 영상 정보를 갱신하는 방법으로 메타 태그를 이용한 방식과 유사하나, 갱신할 이미지만을 갱신할 수 있어, 메타 태그보다 향상된 전송 속도와 영상을 제공할 수 있다.
•Java Applet를 통한 Refresh - Java Applet를 통해 영상 정보를 갱신하는 방법으로 화면 전체를 갱신 하지 않고 빠른 갱신 속도를 제공한다. 사용자의 웹 브라우저에서 Java Applet이 실행되어 웹 서버로부터 영상 정보를 획득한다.

3.2 Server Push 영상 전송

Client Pull 방식이 영상 정보를 갱신할 때마다 웹 서버와 연결을 만들어지는 것과 달리, 연결을 계속 유지한 상태에서 영상 정보를 전송하는 방식으로 다음과 같이 2가지 형식으로 구분될 수 있다.

•MIME type 사용 - 동적 문서 형식으로 단순한 CGI 프로그램을 통해 이미지를 웹 브라우저로 계속해서 전송하는 방식으로 Netscape에서는 가능하지만 Internet Explore에서는 지원하지 않는다.
•Java Push를 통한 Refresh - Java Push가 지원되는 특정 프로그램에서만 사용될 수 있으며, 고정 IP가 있어야 한다. 접속 사용자수가 많을 경우에는 시스템이 느려질 수 있다. 하지만 빠르고 깔끔한 이미지 전송이 가능하다.

3.3 영상 전송 방식 선정

영상 전송 방식의 선정은 다음의 기준에 따라 선정하였다.[2]

•목표지향성 - 영상정보를 사용자에게 제공하는 방식을 선정함에 있어, 영상정보가 사용되는 목적에 충분히 적합하여야 한다. 예를 들어 영상정보가 로봇의 움직임을 제어하기 위한 것이라면, 영상정보는 실시간 영상이어야 하고, 시간적인 버퍼가 최소이어야 한다.
•편의성 - 사용자는 웹 브라우저를 사용하여 임베디드 시스템이 제공하는 영상정보를 보는데 있어, 사용상의 편의성을 제공하여야 한다. 사용자는 추가적인 프로그램 설치와 같은 영상을 보기위한 추가적인 작업이 최소화되어야 한다.
•호환성 - 영상정보는 Internet Explorer와 Netscape로 크게 구분되는 웹 브라우저에서 정상적으로 실행되어야 한다. 특정 웹 브라우저에서만 실행되는 방식은 지양되어야 한다.
•성능 - 영상정보를 사용자의 웹 브라우저에 제공할 때에는 가장 효율적인 방식을 제공하여야 한다.

분 류	구현 방법	목표지향	편의성	호환성	성능
Client Pull	Meta Tag	◎	◎	◎	△
	Java Script	○	○	◎	◎
	Java Applet	◎	✕	○	○
Server Push	MIME	△	○	✕	○
	Java Applet	○	✕	✕	○

◎ 양호 △ 보통 ✕ 미흡

표 1. 영상 전송 방식 평가

위의 [표 1]에 나타나듯 MIME 방식은 Internet Explorer에서는 호환성을 제공하지 않는 단점이 있고, Client Pull 방식 중에는 Meta Tag 방식이 편의성이나 호환성에서 좋으나 영상이 깜박이는 현상이 있어 성능 부분에서 떨어진다. Java Applet는 JVM(Java Virtual Machine)을 설치해야하는 불편함이 있으며, 부가적인 Applet파일이 있어야 하므로 선정 기준상으로 미흡하다. 이에 비해 Java Script는 낮은 성능의 프로세서와 작은 크기의 메모리를 가진 내장형 임베디드 리눅스 시스템 상에서 이미지 전송을 하기 위해 부가적인 프로그램이나 실행 파일이 필요 없다. 또한 이식성이 높은 Java Script code만으로 웹 브라우저에 영상 정보를 계속해서 갱신하므로 사용자에게 마치 동영상을 보는 듯한 효과를 제공한다.

이에 따라 본 논문에서는 Java Script 방식의 영상 전송을 임베디드 시스템에서의 원격 모니터링을 위한 최적의 방법으로 제시한다.

4. 실험

임베디드 리눅스 시스템 기반의 영상 전송을 위해 선정된 Java Script 방식을 사용하여 실시간 원격 모니터링을 구현 하였다. 위에서도 언급하였지만 Java Script 방식을 사용하기 위해서는 부가적인 프로그램이나 파일이 필요 없으며, Java Script code만 모니터링 하기 위한 웹 페이지에 추가하면 된다.

4.1 Java Script code 구현

모니터링 하기 위한 웹 페이지의 HTML 코

드에 그림 4와 Java Script code만 추가 해주면 되며, code를 통해 Refresh 시간을 msec단위로 제어 할 수 있다.

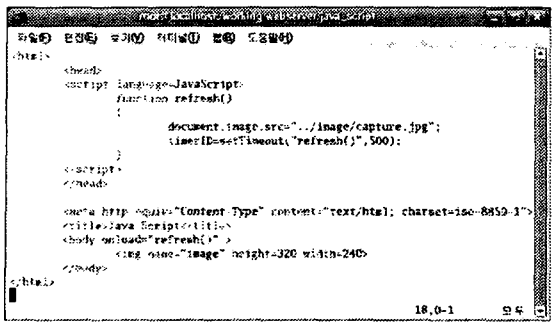

그림 3 Java Script code

4.2 모니터링 결과

Meta Tag 방식이나, Java Applet 방식은 웹 브라우저에서 서버의 이미지를 보여주는데, 이미지가 깜박거리는 현상이 있었으나, Java Script 방식을 통한 이미지 전송은 이미지만 Refresh하기 때문에 클라이언트는 사용자에게 슬라이드 방식으로 마치 동영상과 같은 효과로 영상정보를 제공한다.

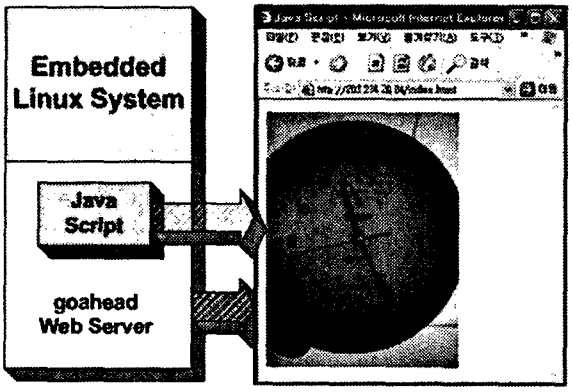

그림 4 모니터링 실행 화면

5. 결론

본 논문에서는 기존에 연구된 실시간 원격 모니터링을 위한 영상 정보 전달 방법에 대해 살펴보고 임베디드 리눅스 시스템에 가장 적합한 영상 정보 전달 방법을 제시하였다. 각각의 영상 정보 전달 방식은 선정 기준에 따라 장단점이 있지만, 이중에서 특히 Java Script 방식이 다른 영상 정보 전달 방식에 비해 좋은 성능을 가지고 있음을 임베디드 시스템 기반의 실험 결과를 통하여 확인하였다. 이에 제한된 사항을 가진 임베디드 리눅스 시스템 환경에서 추가적인 응용 프로그램 설치나 추가적인 실행 파일 없이 원격에서 실시간 모니터링을 하는데 Java Script 방식이 좋은 결과와 성능이 보임을 확인 할 수 있었다.

참 고 문 헌

[1] 남철, 김희식, 김영일, 오드게럴, 하규용, "임베디드 리눅스 상에서 USB Digital Camera를 이용한 실시간 동영상 방송시스템 구현" 산업기술연구소논문지, Vol. 12, No. 43, pp. 137-140, 2004

[2] 김주만, "URC 로봇 원격 모니터링 기술 개발", 한국콘텐츠학회논문지, Vol. 6, No. 8, pp. 9-19, 2006

[3] (주)한백전자 기술연구소, "모바일 장치의 실습과 응용" ITC.

[4] 박재호, "임베디드 리눅스", 한빛미디어.

[5] 김대욱, 이동욱, 심귀보, "Embedded Linux를 탑재한 Home Network Mobile Robot", Proceedings of KFIS Autumn Conference, Vol. 15, No. 2, pp 542-545, 2005

[6] 신은철, 최병욱, "실시간 임베디드 리눅스를 이용한 이동 로봇 플랫폼 구현", 제어·자동화·시스템공학 논문지, Vol. 12, No. 2, pp. 194-200, 2006

[7] 성하준, 김대진. "지능형 임베디드 홈 오토메이션 서버의 구현", 전자통신기술연구소 논문지, Vol. 8, No. 1, pp. 41-46, 2006

[8] 박태현, 강근택, 이원창, "인터넷을 통하나 비젼 기반의 자율이동 로봇 제어", 2000년도 대한 전기 학회 하계학술대회 논문집 2000.7.17-20, pp 2708-2710

[9] 최병욱, 김현기, 신은철, "임베디드 리눅스 기반의 다중 프로토콜 제어기 개발 및 빌딩자동화시스템과의 연동 적용", 제어·자동화·시스템공학 논문지, Vol. 10, No. 5, pp. 428-433, 2004

[10] 이연조, "임베디드 리눅스 프로그래밍", PC'BOOK, 서울, 2002