

USB 웹캠을 이용한 임베디드 감시시스템의 구현

강남호, 이석준, 이현용, 정두현, 홍영철, 권오준
동의대학교 컴퓨터소프트웨어공학과

An Implementation of Embedded Monitoring System with USB Webcam

Nam-Ho Kang, Seock-Jun Lee, Hyun-Yong Lee, Du-Hyun Jung, Young-chul Hong, and Oh-Jun Kwon
Dept. of Software Engineering, Dong-Eui University
E-mail : hyzzang84@dreamwiz.com

요 약

인터넷과 정보통신 기술의 발달로 네트워크의 활용 범위가 넓어지게 되었다. 이에 따라 감시용 모니터링 시스템의 응용분야가 다양해졌을 뿐만 아니라 시스템에 대한 요구사항도 많아지고 되었다. 특히 소규모의 매장에서의 응용을 비롯한 무인화 시스템의 확산은 일반인이 쉽게 다룰 수 있는 보급형의 저가 감시시스템의 개발을 요구하고 있다. 본 연구에서는 저렴하고 다양한 부가서비스가 가능하며 인터넷에 쉽게 접속할 수 있는 감시시스템을 구현하였다. 리눅스 기반의 임베디드 리눅스 시스템을 이용한 웹캠 서버를 구축하였다. 구현된 시스템은 웹서버를 통해 실시간 모니터링 기능이 잘 동작하는 것을 확인하였다

I. 서 론

정보통신 기술의 발달은 사회 전반을 유무선의 인터넷으로 연결하고 있다. 사람들은 이러한 환경 속에서 개인 휴대 장비를 통한 원격관리 및 감시로 편리함을 누릴 수 있다. 그러나 현재는 초기 단계로 CCTV 기반의 감시 시스템은 상당히 비싼 가격에 거래되고 있다. 하지만 기존의 이런 시스템은 특정시스템에 특화되어 있어 인터넷과 결합하기에는 무리가 있다. 인터넷과 정보통신 기술의 발달로 네트워크의 활용 범위도 넓어지게 되었다. 이에 따라 감시용 모니터링 시스템의 응용분야가 다양해졌을 뿐만 아니라 시스템에 대한 요구사항도 많아지고 되었다. 특히 소규모의 매장에서의 응용을 비롯한 무인화 시스템의 확산은 일반인이 쉽게 다룰 수 있는 보급형의 저가 감시시스템의 개발을 요구하고 있다. 결국 인터넷과 결합하기 쉬우며, 차세대 인터넷 및 네트워크에서 사용될 수 있는 감시 카메라 시스템이 필요로 한다. 본 연구는 임베디드

드 시스템과 USB(Universal Serial Bus) Webcam을 결합한 시스템에 영상을 캡처하여 웹서비스를 하는 소규모 웹서버가 탑재된 시스템을 제작한다. 시스템 특징은 USB Webcam을 이용하는 데 있다. USB WebCam의 장점은 PC에서 사용할 수 있는 화상 카메라로 CCTV(Closed Circuit Television)에서 쓰이는 전용 WebCam에 비하면 가격대 성능이 우수하여 경쟁력이 높다. 그리고 USB 인터페이스를 사용함에 따라 누구라도 손쉽게 연결을 설치할 수 있다.

II. USB 웹캠을 이용한 감시시스템의 개요

웹캠 서버 시스템의 가장 중요한 기능은 영상을 획득하는 일이다. 획득할 수 있는 영상의 종류는 동영상이나 정지영상이 있다. 이 중 정지영상은 일정한 시간 간격으로 얻어지므로 실시간 특성이 떨어진다. 그러나 정지영상을 캡처하는 시간을 줄인다면 이러한 단점을 줄일 수 있다. 이렇게 USB 웹캠으로부터 얻어진 영상은

웹캠 서버 시스템의 관리자 혹은 허락된 사용자가 볼 수 있어야 한다. 기존 시스템들은 주로 비디오테이프에 녹화하거나 실시간적으로 모니터에 보여주는 정도였다. 그러나 최근에는 인터넷이 널리 보급되어 어디서나 접속할 수 있다. 그래서 이러한 영상을 웹으로 제공한다면 휴대폰이나 PDA 등을 통해 언제 어디서든지 확인이 가능해 감시 관리의 편의성을 높여 줄 것이다. 또한 웹으로 시스템 온/오프와 같은 설정 또한 가능케 하면 설치가 어려운 장소에 한번 설치 후 모든 관리는 웹으로 이루어지므로 관리의 편의성을 한층 높여준다.

<표1> 기존 시스템과의 기능 비교

	기존 시스템	제안 시스템
영상 장치	고가의 전용 카메라	PC용 USB 웹캠
서비스 방법	비디오테이프 녹화	인터넷 웹 서비스
	실시간 모니터 출력	
부가 기능	줌 기능	줌기능
		다중 USB 웹캠 설치
		무선 인터넷
		다양한 센서 장착

USB웹캠은 영상을 캡처하는 카메라 장치이다. USB호스트 컨트롤러는 USB 웹캠을 제어하는 칩셋으로 전원 공급과 USB 장치의 제어 신호 및 데이터 흐름을 조절한다.

III. 제안 시스템의 구현

본 연구에 사용된 USB웹캠은 (주)로지텍에서 생산한 Logotech 1.3-MP Webcam C300 제품을 사용하였다.



(그림1) Logotech 1.3 -MP Webcam C300

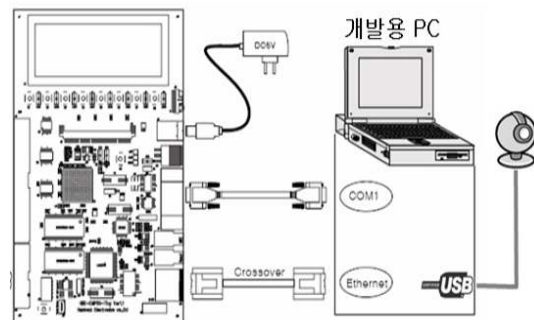
본 연구에서는 웹캠 서버 시스템을 구현하기

위해 임베디드 리눅스 시스템을 채택하였다. 임베디드 시스템 제품 중 X-Hyper320(이하 Hyper320)을 선정하여 여기에 임베디드 리눅스를 포팅하였다. 이 Hyper320보드는 Intel의 Xscale 프로세서를 장착하여 저전력, 고성능의 장비 개발을 위한 솔루션이며 각종 확장 포트를 갖추고 있어 다양한 실험이 가능하다.



(그림2) x-Hyper 320(TKU)

개발용 PC에 USB 웹캠을 장착하였다. 그리고 직렬통신과 이더넷 포트 간에 크로스케이블을 연결하여 네트워크 구성을 하였다. 개발용 PC에는 리눅스 커널을 사용하는 Ubuntu8.10 리눅스를 설치하였다. Ubuntu8.10 리눅스 커널에 기반한 완전한 오픈소스 운영체제이다.



(그림3) 개발환경 구성도

본 연구에 사용된 개발용PC의 사양은 리눅스 커널과 몇 개의 라이브러리를 컴파일하고 커널 Ubuntu8.10을 사용하기에는 조금 느렸다. 다음은 사용된 개발용 PC의 사양이다.

<표2> 개발에 사용된 PC사양

CPU	Pentium 4 CPU 2GHz
RAM	512 MB
NETWORK	Ethernet 10/100M

웹캠 서버 시스템에서 핵심 부분인 USB 웹캠에 의한 이미지 캡처 기능을 구현하는 방법에 대해 설명한다. 우선 USB 웹캠을 임베디드 리눅스 시스템에 인식시키고 동작제어를 위해 USB 웹캠과 호스트컨트롤러 드라이버 개발방법에 대해서 논한다.

리눅스에서 디바이스 드라이버는 하나의 파일로 처리된다. 즉 rm 파일을 통해 외부 주변장치를 액세스하거나 제어할 수 있다. 하나의 디바이스 드라이버는 주번호(Major number)와 부번호(Minor number)로 나타낸다. 주번호는 장치유형, 부번호는 그 유형의 단위기기를 나타낸다. 디바이스 드라이버는 크게 3가지로 나눌 수 있다. 문자(character) 디바이스 드라이버, 블록(block) 디바이스 드라이버, 네트워크(network) 디바이스 드라이버이다. 문자 디바이스 드라이버는 파일처럼 취급하여 액세스한다. 버퍼를 사용하지 않고 직접 읽기/쓰기를 하며, 데이터 형태는 스트림 방식으로 전송한다. USB 디바이스 드라이버는 USB에 대한 애플리케이션의 입출력 요구를 적절한 형태로 USB core 쪽으로 전송하는 역할을 하며, 또한 그 처리의 결과를 애플리케이션으로 되돌려 주는 역할을 한다. 여기서 말하는 애플리케이션은 사용자 수준의 프로그램이 될 수도 있을 것이며, 커널 내에 들어가는 또 다른 디바이스 드라이버도 될 수 있다. 따라서 상위의 애플리케이션은 하위의 USB 구현 구조에 상관없이 일관된 인터페이스를 제공해 주어야 한다. 또한 USB 디바이스 드라이버는 이렇게 주어진 요구에 맞춰서 적절히 USB core와 통신할 수 있어야 한다. 이것을 위해 도입된 것이 URB이다.

USB 장치에 대한 설정값을 configure하는 function이 필요하다. 이것은 USB 장치에 대한 control transfer의 형태로 주어진다. USB device structure는 usb.h 파일에 정의되어 있다.

다음은 USB device에 기본적인 function이

다. usb_get_descriptor는 단일의 USB descriptor를 디바이스로부터 읽어 들일 수 있는데, vendor specific descriptor를 파싱하는데 사용된다. 모든 USB와 관련된 function과 data structure는 “usb_”로 시작하는 규칙을 가진다.

<표3> USB Driver structure

```

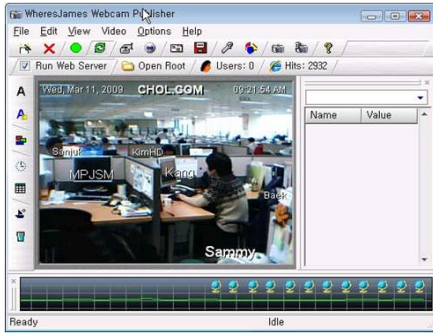
struct usb_driver{
    const char *name;
    void (*probe) (struct usb_device *,
        unsigned int);
    void (*disconnect) (struct usb_device
        *, void *);
    struct list_head driver_list;
    struct file_operations *fops;
    int minor;
}
    
```

USB 디바이스 드라이버는 USB에 대한 애플리케이션의 입출력 요구를 적절한 형태로 USBcore 쪽으로 전송하는 역할을 하며, 또한 그 처리의 결과를 애플리케이션으로 되돌려 주는 역할을 한다. 여기서 말하는 애플리케이션은 사용자 수준의 프로그램이 될 수도 있을 것이며, 커널 내에 들어가는 또 다른 디바이스 드라이버도 될 수 있다. 따라서 상위의 애플리케이션은 하위의 USB 구현 구조에 상관없이 일관된 인터페이스를 제공해 주어야 한다. 또한 USB 디바이스 드라이버는 이렇게 주어진 요구에 맞춰서 적절히 USB core와 통신할 수 있어야 한다. 이것을 위해 도입된 것이 URB이다.

<표4> USB Driver 접근에 관한 function

```

int usb_set_configuration(struct usb_de
    vice *dev, int configuration );
int usb_get_device_descriptor( struct
    usb_device *dev );
int usb_get_descriptor( struct usb_device
    *dev, unsigned char des
    
```



(그림4) 결과화면

IV. 결 론

본 연구에서는 임베디드 리눅스 시스템과 USB 웹캠을 이용하여 감시 카메라 시스템을 구현해 보았다. 리눅스 운영체제는 서버로서의 기능이 뛰어나고 다양한 응용프로그램 탑재 가능하다. 그리고 USB 웹캠은 가격에 비해 고화질의 영상을 캡처해 낸다. 또한 USB 인터페이스는 사용자들에게 친숙하다. 결국 리눅스와 USB 웹캠의 장점을 한데 모은 감시 카메라 시스템은 더욱더 많은 부가시스템을 추가하고도 사용자들이 쉽게 사용할 수 있는 장비가 될 것이다. 그리고 리눅스 운영체제와 USB 웹캠을 이용하여 구현함으로써 구현 비용을 많이 낮추었기 때문에 일반인이 경제적인 부담없이 보급형으로 사용할 수 있을 것으로 기대된다.

또한 본 연구결과는 가정에서 감시시스템, 대학 캠퍼스의 실습실 등에 다양한 용도로 이용될 수 있다. 단지 감시카메라와 X-hiper 보드가 합체화하여 단일한 간단한 시스템으로 구성되며, 무선랜 접속기능이 있는 경우에 더욱 다양한 감시 기능을 가질 수 있을 것이다.

앞으로의 연구 내용으로는 USB 접속을 통해 최대 접속 카메라 수와 이를 위한 성능 문제를 검토해야 한다. 그리고 시스템에서 접속 전 웹 카메라에서 전송된 캡처 이미지 정보를 저장하는 기능과 성능에 대한 분석과 신속한 동작 기능에 대한 연구가 요구된다.

참고문헌

- [1] Alessandro Rubini, Jonathan Corbet, Greg Kroah-Hartman, "Linux Device Drivers 3/E", O'REILLY, 2009.
- [2] Young-Chang Yu, "IT EXPERT Linux Device Drivers", Hanbit Media, 2008.
- [3] Daniel P. Bovet, "Understanding the Linux Kernel (2nd Edition)", ORBILLY, 2009.
- [4] Jae-Young Sung, "EMPOS-tiny Manual", HanbackJunJa, 2003.
- [5] Hyung-Hoon Kim, "USB GUIDE", OHM, 2008.
- [6] 오픈소스 USBWebCam Driver, <http://mxhard.free.fr>, 2007.
- [7] Korea Embedded System of Linux, <http://keal.org>, 2007.
- [8] Linux Xwindow 화상캡처프로그램, <http://camorama.flxedgear.org>, 2008.