

지능형 PVR의 원격제어를 위한 홈 서버 구현

The implementation of home-server for intelligent Personal Video Recorder

손강선*, 오영호**, 김대진***

Kang Sun Son, Young Ho Oh, Dae Jin Kim

Abstract - The intelligent PVR(Personal Video Recorder) is an enhanced PVR that provides viewers with some advanced features as well as pause, instant replay, search and skip forward found in conventional PVRs. By embedding a home server into a PVR, it is possible for an intellignet PVR to provide a powerful web-based management user interface constructed using HTML, graphics and other features common to web-browsers. When applied to other embedded systems, web technologies offer graphical user interfaces which are user-friendly, inexpensive, cross-platform and network-ready. It is the purpose of this paper to introduce implementation of intelligent PVR which is control by internet. We present the architecture of a home-server with a simple but powerful web-based network interconnection.

Key Words : embedded, LINUX, socket, home-server, PVR

1. 서론

최근 들어 디지털 기술이 눈부신 발전을 거듭하고 있다. 이를 바탕으로 다양한 전자제품이 디지털화되었고 그 대상이 점차 확대되고 있다. 이에 따라 가전, 통신 및 컴퓨터 간에 상호 영역을 뛰어넘는 디지털 컨버전스 추세가 두드러지고 각 기기간의 구별이 모호해지기 시작했다.

PVR은 하드디스크 저장/검색 기능과 EPG기능을 결합한 것으로 기존의 VCR에 비하여 편리하기는 하지만 인터페이스 기술을 접목하여 사용자에 대한 지원이 지능화되고 더욱 사용하기 쉬운 제품으로 개량될 필요가 요구된다.

PVR 역시 제약된 환경에서 동작하는 임베디드 시스템이고 앞으로 우리 주위에는 수많은 임베디드 시스템이 존재하게 될 것이다. 이러한 다양한 임베디드 시스템들을 서로 연결해서 네트워크를 구성하고, 이를 기반으로 다양한 형태의 서비스를 제공하기 위해서는 서로 간에 문제없이 통신하기 위한 공통의 규약이 필요하게 된다. 이 경우 RS-232C같은 단순한 통신 프로토콜에서부터 FTP나 HTTP 같은 다양한 종류의 프로토콜이 사용 가능하다. RS-232C처럼 단순한 연결을 통해서선 전송속도나 전송거리등의 문제가 있기 때문에

이를 기반으로 임베디드 시스템간의 네트워크를 구성하는데 많은 무리가 따른다.

이러한 문제의 해결책으로 임베디드 시스템에 인터넷 인터페이스를 구성하여, 임베디드 시스템에 적용 가능한 임베디드 리눅스 기반의 임베디드용 웹서버를 탑재시켜 HTTP를 사용해서 임베디드 시스템을 연결하여 네트워크를 구성할 경우 다음과 같은 몇 가지 장점이 있다.

첫째로 HTTP를 기반으로 연결되기 때문에 인터넷을 통해 사용자가 임베디드 시스템에 접근할 경우, 별도의 전용 프로그램이 필요하지 않게 된다. 사용자는 스마트 폰, PDA, PC등 웹 브라우저가 이용 가능한 환경에서 웹을 통해 쉽게 임베디드 시스템을 제어할 수 있게 된다.

둘째로 기존의 PC상에서 웹을 기반으로 이루어졌던 많은 서비스들이 약간의 수정을 통해 임베디드 시스템에서도 적용이 가능하게 되고, 이에 따라 많은 부분을 재사용할 수 있기 때문에 웹을 통한 서비스 개발의 비용감소 및 시간을 단축시킬 수 있다.

셋째로 임베디드 리눅스 기반으로 구현함으로써 임베디드 리눅스가 주는 많은 장점을 이용할 수 있다. 임베디드 리눅스는 오픈 소스 소프트웨어이기 때문에 상용 운영체제에 비하여 상당한 가격 경쟁력을 가지게 되고 개발에도 많은 도움을 준다. 무엇보다 임베디드 리눅스가 범용 리눅스와 같은 커널을 기반으로 만들어지기 때문에 사용된 코드를 다른 임베디드 시스템에 쉽게 포팅(Porting) 시킬 수 있다.

따라서 본 논문은 ARM 계열의 CPU가 탑재된 타겟보드로

저자 소개

* 孫 剛 宣 : 全南大學 電子情報通信工學科 碩士課程

** 吳 榮 浩 : 全南大學 電子情報通信工學 博士課程

***金 大 鎭 : 全南大學 電子컴퓨터情報通信工學 副教授 · 工博

구성된 PVR에서 동작 가능한 웹 서버를 설계하고 구현하여 웹브라우저를 통해 사용자가 쉽게 PVR의 원격제어가 가능하도록 전체 시스템을 구성하였다.

2. 개발 환경의 구성

그림 1은 개발 환경의 구성이다. 타겟보드로 구성된 PVR에는 CS8900A 이더넷 컨트롤러가 탑재되어 TCP/IP를 통해 네트워크에 연결이 가능하다. 호스트 PC에는 범용리눅스를 설치하고 타겟보드용 커널 및 응용 프로그램을 개발하기 위한 ARM용 크로스 컴파일 환경을 구축하였고 타겟보드와 RS-232C로 연결하여 터미널 에뮬레이터를 통해 타겟보드를 모니터링 한다. 사용자가 인터넷을 통하여 원격지에서 웹을 통해 타겟보드에 접속하여 타겟보드의 상황을 모니터링하고 제어할 수 있도록 허브를 통해 이더넷에 연결하였다.

개발의 편의를 위하여 타겟보드의 IP 주소를 고정시키지 않고 호스트 PC의 BOOTP서비스를 통하여 부팅시 IP주소를 할당받도록 구성하였고, TFTP를 통하여 커널 이미지 및 램 디스크를 타겟보드에 다운로드 하도록 구성하였다. 또한, 타겟보드에 컴파일된 실행 파일을 매번 부트로더로 다운로드 하는 불편함을 해결하기 위해 NFS를 이용하여 개발 호스트 상의 파일을 타겟보드에서 마운트 시켜 사용하였다. NFS를 이용하면 타겟보드에서 개발 호스트에 있는 파일이 접근 가능하고, 실행이 가능하기 때문에 램디스크에 올리기 큰 파일도 쉽게 처리가 가능하다.

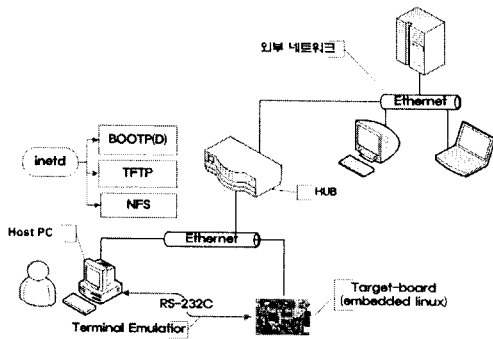


그림 1. 개발 환경 구성도

3. 웹 서버의 구조 및 구현

그림 2는 구현된 웹 서버의 수행과정을 나타내고 있다. 메인 프로세스가 시작되면 서버는 먼저 클라이언트(Client)와 TCP/IP 연결을 위한 서버의 소켓을 준비한다. 소켓이 준비된 후 클라이언트의 연결을 기다리게 된다. 클라이언트로부터 연결이 이루어지면 클라이언트에게 웹 서비스를 제공하는 새로운 프로세스(child process)를 생성하게 된다. 자식 프로세스는 연결된 클라이언트의 요청을 분석하고 응답을 하는 역할만 수행하고, 부모 프로세스(main process)는 새로운 클라이언트가 연결될 때마다 새로운 자식 프로세스를 생성해서 접속을 연결하는 역할과 종료된 자식 프로세스의 자원을 반환하는 역할을 수행하게 된다. 위와 같은 부모와 자식간의

프로세스의 구분은 설계 및 구현이 용이해질 뿐만 아니라 동시에 여러 사용자에게 웹을 통한 서비스를 제공할 수 있게 해준다.

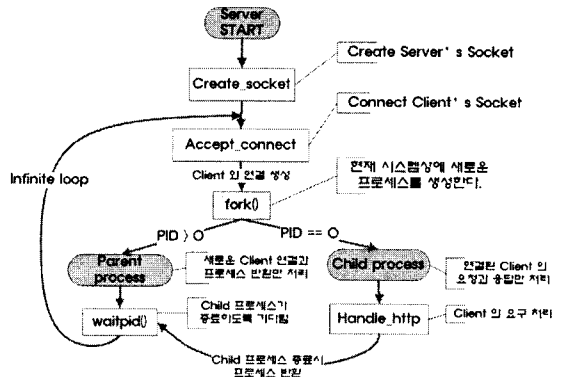


그림 2. 구현된 웹 서버의 수행과정

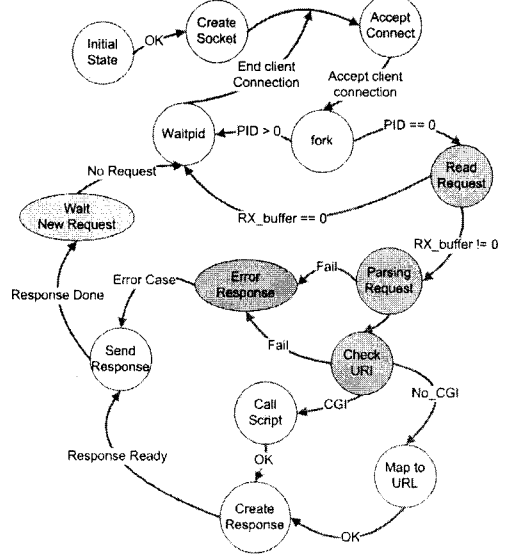


그림 3. 클라이언트 요청 처리 및 응답 과정

그림 3은 전체 프로세스에서 클라이언트의 요청이 처리되어 응답이 이루어지는 과정을 나타내고 있다. 먼저, 클라이언트와 연결이 되면 서버는 클라이언트의 요청을 읽어 RX_buffer에 저장한다. 클라이언트로부터 요청이 없는 경우(RX_buffer == 0)는 자식 프로세스가 종료되고 사용한 자원이 반환된다. 요청이 있는 경우(RX_buffer != 0) RX_buffer에 저장된 요청을 분석하여 MetaVariables를 설정하게 된다. 이렇게 설정된 환경변수를 바탕으로 URI(Uniform Resource Identifier)를 검사하여 동적인 데이터를 요구하는 CGI 스크립트(Script)인 경우는 새로운 프로세스로 스크립트를 호출하여 결과를 표준출력을 통하여 사용자에게 전달하고, 정적인 데이터를 요구하는 경우에는 요청된 URI를 전달하는 형태로 사용자의 요청에 대한 적절한 응답을 해주는 일련의 동작을 사용자의

요청이 없을 때까지 반복한다. 추가적으로 요구되는 기능들은 모듈형태로 URI확인(Check URI) 다음 단계에 병렬적으로 추가함으로써 CGI를 비롯한 다양한 서비스가 가능하게 된다.

4. 원격제어가 가능한 PVR 시스템

그림 4는 최종적으로 구현된 홈 서버가 탑재된 PVR 시스템을 나타내고 있다. 최종 구현된 홈 서버는 고정 IP 주소로 설정된 램디스크 이미지를 다운로드 하였지만, 네트워크 환경에 따라서 BOOTP나 DHCP를 이용해서 다른 서버로부터 부팅할때마다 IP주소를 할당받아 사용할 수도 있다.

그림 4에 나타난 PVR 시스템은 다양한 형태의 임베디드 시스템이 될 수 있다. 예를 들면 이더넷 인터페이스가 장착된 디지털 TV, 셋탑 박스등 앞으로 가정에서 흔히 볼 수 있는 가전제품일수도 있다. 구현된 홈 서버는 소켓프로그래밍을 통해서 소스레벨에서의 호환성을 최대한으로 고려하고 구성하였기 때문에 이더넷 인터페이스가 탑재되고 임베디드 리눅스가 동작하는 환경에서 별도의 수정 없이 크로스 컴파일러를 이용한 재컴파일만으로 동일하게 동작한다. 실제로 구현 과정에서는 호스트 PC에서 GCC를 이용하여 소스 컴파일하여 호스트 PC상에서 웹 서버의 동작을 확인한 다음 동일한 소스 크로스 컴파일러를 이용해서 재컴파일 한 후 타겟보드에 다운로드 시켜도 똑같이 동작함을 확인하였다.

또한, 구현된 홈 서버가 타겟보드에 대한 소스수정 없이 크로스 컴파일러에 의한 재컴파일 만으로 PXA255(XScale)기반의 타겟보드와 SA-1110(StrongARM)기반의 타겟보드에서 동작함을 확인하였다.

이렇게 구현된 홈 서버는 PVR 시스템 상에서 동작하면서 HTTP를 통해서 사용자가 PC나 다른 기기를 이용하여 웹 브라우저로 PVR 시스템에 접속이 가능하게 하고 있다. 사용자는 어떤 기기를 이용하든지 웹을 이용하여 웹 브라우저를 사용하는 공통된 환경을 통해 임베디드 시스템의 상태를 확인할 수 있고, 제어할 수 있게 된다. 임베디드 시스템의 제어를 위해서 구현된 홈 서버는 범용PC환경에서 사용되는 CGI도 지원하여 PVR 시스템의 제어를 위한 별도의 프로세스를 동작시키게 된다.

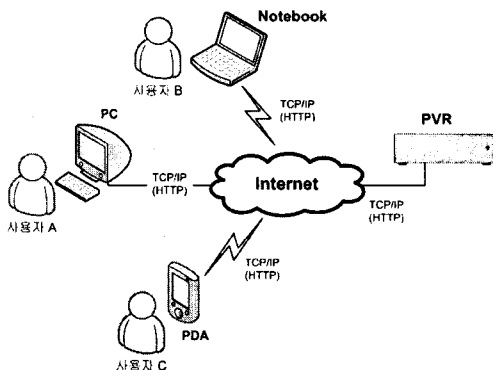


그림 4. 원격제어가 가능한 PVR 시스템

5. 결론

디지털 기술의 발전으로 다양한 제품의 디지털화가 급격히 가속되고 있으며, 이에 따라 가전, 통신 및 컴퓨터 등의 분야에서 서로 연결되고 융합되는 디지털 컨버전스가 활발히 진행되고 있다. 이러한 추세는 앞으로 더욱 가속되어 우리가 일상생활에서 흔히 접하게 되는 다양한 임베디드 시스템들까지 확장되고 네트워크화 되어 유비쿼터스 컴퓨팅 환경을 이루어 나갈 것이다.

본 논문에서는 임베디드 시스템에서 사용 가능한 임베디드 리눅스 기반의 홈 서버를 구현하였다. 구현된 홈 서버를 PVR에 탑재하여 웹 페이지 요청에 대한 응답 처리 같은 기본적인 동작에서부터 CGI를 통한 다양한 형태의 서비스 지원이 가능하도록 구현하였다. 이를 통하여 사용자는 별도의 전용 프로그램이 아닌, 웹 브라우저를 통해 PVR 시스템에 접속하여 원격으로 PVR 시스템을 제어할 수 있다. 또한, 이더넷 인터페이스가 연결되고 임베디드 리눅스가 동작하는 임베디드 시스템에서 동작하도록 소스 레벨에서의 호환성이 고려되었기 때문에 구현된 홈 서버는 다양한 임베디드 시스템에 탑재할 수 있어 이를 이용하여 가정에 있는 정보 기기들을 사용자가 원격에서 웹 브라우저를 통해 제어한다거나 서로 다른 기기 간에 정보 전달이 가능하고, 기존의 범용PC상에서 이루어지던 다양한 서비스를 간단히 수정하여 적용이 가능하기 때문에 개발 비용의 감소 및 소스코드의 재사용성의 증대 등의 장점이 있다.

*본 논문은 한국과학재단 지정 전남대학교 고품질 전기전자부품 및 시스템 연구 센터의 연구비 지원에 의해 연구되었습니다.

참 고 문 헌

- [1] Behrouz A. Forouzan, "TCP/IP Protocol Suite 2nd Ed.", McGraw-Hill, 2003
- [2] John Catsoulis, "Designing Embedded Hardware", O'Reilly, 2002
- [3] Michael J. Donahoo, Kenneth L. Calvert, "The Pocket Guide to TCP/IP Sockets: C version", Morgan Kaufmann Publishers, 2001
- [4] 박영환, "임베디드 시스템 임베디드 리눅스", 사이텍 미디어, 2002
- [5] 박재호, "임베디드 리눅스", 한빛 미디어, 2002
- [6] 윤성우, "TCP/IP 소켓 프로그래밍", 프리렉, 2003
- [7] <http://kdp.org>
- [8] RFC-2616, Hypertext Transfer Protocol-HTTP/1.1, 1999