

## 임베디드 리눅스 보드를 이용한 홈 네트워킹 시스템 구현에 관한 연구

A Study On The Development Of a Home Networking System  
Using An Embedded Linux Board

이 현 주\*, 이 종 수\*\*, 최 경 삼\*\*\*  
(Heon Joo Lee, Jong Su Lee and Kyung Sam Choi)

- \* 홍익대학교 전기정보제어공학과(전화:(02)325-7514, 팩스:(02)320-1110, E-mail : pulito@unitel.co.kr)
- \*\* 홍익대학교 전기정보제어공학과(전화:(02)320-1669, 팩스:(02)320-1110, E-mail : leejs@wow.hongik.ac.kr)
- \*\*\* 홍익대학교 전기정보제어공학과(전화:(02)320-1686, 팩스:(02)320-1110, E-mail : choiks@wow.hongik.ac.kr)

**Abstract** : In this paper, we have designed a Home Networking System using an embedded linux board. The system based on the World-Wide-Web is composed of three parts - a Server, a Client and a Simulator. The Home Networking Server is built in an embedded board using an embedded linux kernel. A web-server and Home Networking Server Service Demon programming with a Java-Language is included in the board. Clients can connect to the server board using a web-browser in the desktop computer, PDA or any other machines which include a web-browser. For this purpose, we made the client program using a Java-Applet. So, the clients who connect to the server for the control of the applications, download the class-file and execute the client-program in the web-browser. So, the clients don't need any other programs to control the applications from a remote place. The size of server board is very small (86.3x74mm), which makes it very useful not only for the Home-Networking-System but also in many other fields, e.g., embedded robot control system, etc. Using an embedded board instead of a desktop computer is good for a simple network environment and it occupies only a small space to make the system.

**Keywords** : Embedded Linux, Embedded JVM, PowerPC, Home Networking, Java Language, Java Applet

### 1. 서론

최근 홈 네트워킹 시스템을 구축하기 위하여 많은 연구가 이루어지고 있으며 다양한 분야에서 이를 응용한 제품이 출시되어 지고 있다. 여러 가지 기반에서의 홈 네트워킹을 위한 방법이 연구되어 지고 있으며 핸드폰, 일반전화 혹은 배포된 프로그램 등을 이용하여 서버에 접속을 하게 된다. 이를 위하여 집안에 홈 게이트웨이 등을 설치하게 되며, 가전기기 등과 같은 여러 가지 장치들을 원격지에서 제어하는게 가능하게 된다.

본 논문에서는 이러한 네트워크 시스템을 구축하기 위하여 사용자의 편의를 고려하는 방법을 채택하였다. 사용자는 별도의 프로그램 없이 웹 브라우저를 내장하고 있는 일반 컴퓨터나 단말기 등을 사용하여 가정에 설치되어 있는 네트워킹 서버에 접속하게 된다. 또한 이 서버는 웹서버의 역할도 가능함으로써 개인 홈페이지 서버로도 사용할 수 있게 되며, 사용자는 서버로부터 받은 프로그램을 웹 브라우저에서 직접 실행하여 별도의 프로그램 설치의 절차 없이 가정 내의 서버에 접속할 수 있게 된다.

웹 기반의 프로그램 개발을 위하여 자바 프로그램

언어를 사용하였다. 사용자는 서버로부터 자바로 작성된 바이트코드(class-file)를 다운 받은 후, 브라우저 내에 설치되어 있는 Java Virtual Machine(JVM)을 통하여 실행되게 되며, 그 실행 결과를 바로 웹 브라우저에서 확인할 수 있다는 장점을 가진다.[1] 이와 같이 사용자는 서버에 접속하여 어플리케이션(가전기기 등)의 제어가 가능하며, 현재 가전기기들의 상태를 모니터링할 수 있게 된다.

서버 시스템에 사용되는 컴퓨터는 웹서버를 내장하고 있어서 사용자가 웹을 통하여 프로그램을 받는 것이 가능도록 해야 한다. 또한 홈 네트워킹 시스템의 서버 역할을 하기 위하여, 데이터를 처리하고 저장할 수 있는 서버 프로그램이 실행되어야 하고, 여러 가지 다양한 가전기기에 데이터를 보내주기 위한 게이트웨이의 역할도 수행하여야 한다. 이를 위하여 하나의 컴퓨터와 운영체제를 탑재한 시스템이 필요하게 되며, 임베디드 보드는 이러한 역할을 대체할 수 있는 장치로서 사용되었다.

임베디드 보드에 리눅스를 포팅하고, 웹서버를 사용 가능하게 만들어 준 후, 자바 프로그램을 실행하기 위한 JVM을 포팅함으로써 웹서버로서의 역할과 홈 네트

워킹 서버로서의 역할을 동시에 수행하게 한다.[2][3]

## II. 홈 네트워킹 서버 시스템

### 1. Embedded Board - Linux Porting

임베디드 보드는 적은 자원을 가지나 많은 자원을 필요로 하지 않는 국한된 목적의 여러 시스템에 적용하기에 적합하다는 이유로 현재 여러 가지 분야에 적용되어 지고 있다. 홈 네트워킹과 같은 간단한 네트워크 프로그램을 실행하기 위해서는 충분한 환경을 제공해 준다. 본 논문에서 사용한 임베디드 보드의 세부 사항은 다음과 같다.[4]

- CPU : PowerQUICC 860P/T 50MHz
- SDRAM : 32Mbyte
- Flash Memory : 8Mbyte
- I/O Interface : 1 10/100 Base-Tx Ethernet Port  
1 BDM Pin Header  
1 RS-232C Serial Port
- BUS Interface : 32Bits Address Bus  
32Bits Data Bus I/O Interface

임베디드 보드에 리눅스를 포팅하기 위해서 사용되어진 파일은 다음과 같다.[5][6]

- i). ppcboot - powerpc embedded board 부팅
- ii). vmlinux.ppcboot - powerpc용 압축 리눅스 커널
- iii). ramdisk.ppcboot - linux Root File System(RFS)

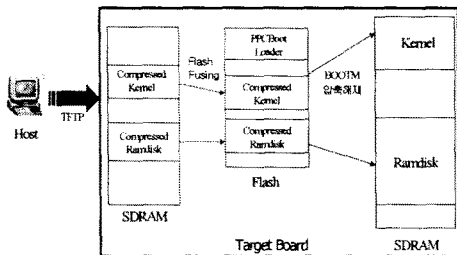


그림 1. Embedded Board 부팅 순서  
Fig 1. Booting Sequence of Embedded Board

보드 자체를 부팅 시키기 위한 ppcboot 파일은 BDM 포트를 이용하여 플래쉬 메모리에 다운로드 하게 된다. 리눅스를 부팅하기 위한 압축 커널 이미지와 RFS 이미지 파일은 개발 환경으로 hard-hat-linux를 사용하는 데스크탑 컴퓨터에서 tftp tool을 사용하여 ethernet port를 통하여 전송 받게 된다.[2][7] 이러한 다운로드된 파일들은 SDRAM에 저장된다. 실제 보드가 stand-alone-type 으로 동작하기 위해서는 보드에 내장되어 있는 Flash Memory에 저장하게 된다. 보드는 이러한 Flash Memory에 저장되어 있는 압축된 이미지 파일을 SDRAM으로 다운로드 한 후 압축을 풀게 된다. vmlinux.ppcboot 이미지 파일로 리눅스 커널을 생성 하며, ramdisk.ppcboot 파일로 부팅에 필요한 RFS를 구성하게 된다.[8] 또한 ramdisk파일은 보드를 홈 네트워킹 서버로 사용하기 위한 다양한 프로그램과

소스등도 포함되어져 있다. 보드를 모니터링 하기 위해서 RS-232 Serial Port를 사용하는 minicom(Linux), 하이퍼터미널(Window) 등의 프로그램을 이용하였다.

### 2. Embedded Board - Java Virtual Machine

현재 우리나라는 거의 모든 사람이 인터넷을 사용하고 있다. 예전 모뎀을 사용하던 방식에서 벗어나 케이블 망과 ADSL을 이용한 통신으로 바뀌었다. 또한 핸드폰 등을 통한 무선인터넷이 보편화되었으며 이러한 인터넷망의 확산에 따라 다양한 서비스를 할 수 있는 네트워크 프로그램의 개발 또한 필요하게 되었다. 네트워크망은 많은 사용자가 다양한 플랫폼을 사용하여 접속을 하고 있다. 자바는 이러한 분산 환경에 적합한 언어이다. 자바는 처음부터 인터넷을 기반으로 하는 네트워크 언어로 개발되었고, 다양한 시스템이 연결되어 있는 인터넷에서는 프로그램이 서로 다른 플랫폼에서 작동하기 위해서 이식성과 호환성이 중요하게 된다.[9] 자바는 이러한 조건을 만족시킬 수 있다. 즉 한번 코딩된 결과 파일을 가지고 서로 다른 CPU와 운영체제를 사용하는 플랫폼에서 별도의 컴파일 과정 없이 사용할 수 있으며, 이는 여러 가지 기기들이 서로 연결되어 있고, 다른 플랫폼을 사용할 수 있는 홈 네트워킹과 같은 시스템에 적합한 언어라고 할 수 있다.

이렇듯 자바가 플랫폼에 독립적인 언어로 사용할 수 있는 이유는 프로그램의 실행을 위해서 Java Virtual Machine을 사용하기 때문이다.

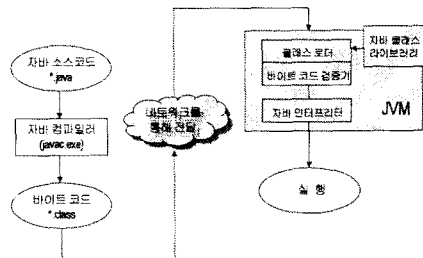


그림 2. 자바의 실행 원리[9]  
Fig 2. execution principal of Java Language

자바 컴파일러로 만들어진 바이트 코드는 네트워크를 통하여 어플리케이션으로 전송되며, 어플리케이션에 내장되어 있는 JVM은 바이트 코드를 해석하여 실행하게 된다. JVM은 각 플랫폼의 CPU, OS 등에 따라 각각 만들어져 있으므로, 각 플랫폼에 맞는 JVM을 설치함으로써 자바 코드는 해석되어 질 수 있다.

본 논문에서 네트워크 환경에 적합한 java 언어를 사용하였고 보드는 Powerpc 계열의 CPU를 사용하며, OS로는 Linux를 사용한다. 따라서 이에 맞는 JVM을 컴파일하여 설치하여야 java로 작성된 서버 데몬과 웹 서버 등을 실행 시킬 수 있게 된다.[3]

Powerpc 계열의 JVM으로는 JamVM(ver 1.0)과 GNU - ClassPath(ver0.04)을 컴파일 하여 사용하였

다.[10][11] 컴파일을 위해서는 hard-hat-linux에서 제공하는 툴을 사용하였다.[7] 이와 별도로 자바로 작성되어 컴파일된 바이트 코드를 직접 플랫폼에 맞는 native-machine-language로 변환하여 줄 수도 있으며, 이를 위해서는 gnu - gcc(ver3.0이상) 패키지 내에 있는 gcj를 사용할 수도 있다.[12][13] gcj역시 powerpc에 계열에 적합하게 컴파일을 하여 주코 보드에 포팅함으로써 사용할 수 있다.

### III. 홈 네트워크 클라이언트 시스템

사용자는 별도의 프로그램 설치 없이 쉽게 서버에 접속할 수 있다. 이는 자바 애플릿을 사용되어 구현되며 애플릿 또한 자바 언어로 만들어 진 것이므로 자바의 장점을 모두 가진다. 네트워크에 적합한 언어라는 장점을 살려 사용자가 어떠한 컴퓨터에서 작업을 하던 웹 브라우저를 통하여 플랫폼에 영향 없이 사용자가 프로그램을 실행할 수 있게 된다.

애플릿 코드는 서버에 저장되어 있으며 웹이 가능한 어느 곳에서든 사용자는 서버에 접속하여 어플리케이션을 동작시킬 수 있게 된다.

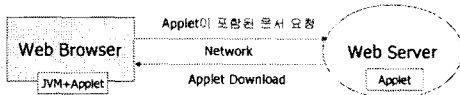


그림 3. 자바 애플릿의 실행 원리  
Fig 3. Execution principal of Java Applet

최신 버전의 웹브라우저는 애플릿을 실행하기 위한 JVM을 직접 내장하지 않은 경우도 있다. 이럴 경우 별도로 JVM을 설치해 주는 절차가 필요하다. 자바 플러그인(자바 애플릿을 지원하지 않은 웹브라우저에서 자동으로 섀마이크로스시스템즈사 홈페이지에 접속 JVM을 다운로드 설치하는 프로그램)등의 발표로 어느 정도 사용이 가능하다고는 할 수 있으나 다운로드 바로 설치하기에는 큰 용량과 느린 설치로 크게 호응을 얻지는 못하였다.[9] 그래서 이러한 문제를 근본적으로 해결하기 위하여 Java Web Start(JWS)라는 기술 또한 사용할 수 있다. JWS는 애플릿 뿐만 아니라 일반 자바 프로그램도 원격에서 실행할 수 있도록 지원한다.[1]

### IV. 홈 네트워크 전체 시스템 구성

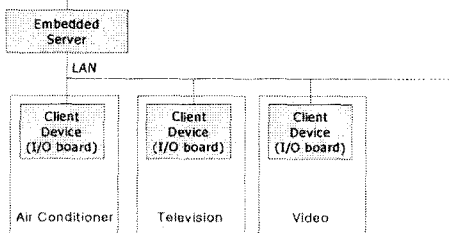


그림 4. 홈 네트워크 시스템 구조  
Fig 3. Architecture of Home Networking System

전체 시스템은 가정 내에 하나의 서버를 둔 방식을 사용한다. 네트워킹 서버는 전체 네트워크의 중심 역할을 하며, 자료를 저장하는 Data-Base의 역할도 하게 된다. 서버 내에서 동작하는 웹서버는 사용자가 웹을 통하여 클라이언트 프로그램을 다운로드 받을 것을 가능하게 만들어 준다. 또한 서버는 클라이언트 프로그램에서 현재 기기들의 상태를 모니터링하기 위하여 기기들의 현재 상태 데이터를 저장하게 되며, 클라이언트의 요청이 있을시 이를 보내 주는 방식을 사용한다.

실제로 동작 확인은 컴퓨터 시뮬레이션을 통해서 이루어진다. 서버측통신은 임베디드 보드가 맡게 되고, 클라이언트 측은 컴퓨터 시뮬레이션을 통하여 데이터의 정상적인 전송을 확인하게 된다.

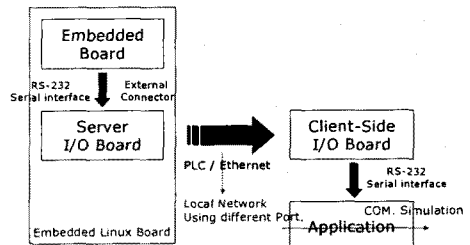


그림 5. 시뮬레이션 시스템 구조  
Fig 5. Architecture of Simulation model

데이터의 전송은 16비트의 데이터를 사용한다. 16비트의 데이터는 각각 기기의 종류와, ON/OFF상태, 세부동작상태 등의 데이터와 데이터의 정상적인 전송을 확인하는 1비트의 패러티 비트로 구성되어져 있다. 실제 네트워크를 통하여 전송되는 데이터는 8비트의 헤더를 포함한 24비트가 하나의 데이터를 이루게 된다.

각각의 시스템은 하나의 아이피로 통합되어 있으며, 시뮬레이션측 각각의 기기는 특정한 port번호로 각 기기 고유한 아이피를 갖게 되며 이를 가지고 각 기기를 구분하게 된다.

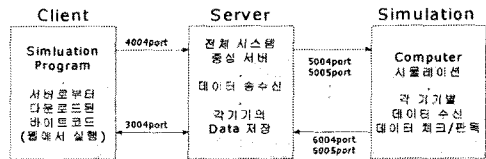


그림 6. 프로그래밍 구조  
Fig 6. Architecture of Programming

시뮬레이션 프로그램에서 기기를 작동시키게 되면, 변화한 데이터는 서버로 저장이 된다. 저장된 데이터는 각 기기별로 분리되며 클라이언트의 요청을 기다리게 된다. 사용자가 클라이언트 측에서 해당되는 기기를 선택하면 서버로부터 해당 데이터를 전송받게 되고 이 데이터는 클라이언트 측에서 분석하여 현재의 상태를 모니터링 할 수 있게 만들어 준다. 클라이언트 측

에서 사용자가 기기를 조작했을 경우 해당 데이터는 서버로 전송되며 기존에 가지고 있던 서버의 데이터는 지워지고 현재의 데이터 값이 저장된다. 이와 동시에 서버는 해당 데이터에 맞는 기기를 선택, 해당하는 포트에 데이터를 전송하게 된다.[14] 이러한 과정을 실시간으로 이루어지므로, 클라이언트의 변화는 서버를 거쳐서 시뮬레이터로 전송되어 데이터 값을 체크할 수 있게 된다.

### V. 실험 및 결과

서버측의 동작 상태 확인은 리눅스 컴퓨터에서 minicom 툴과 RS232-Serial Cable을 사용하여 이루어진다.

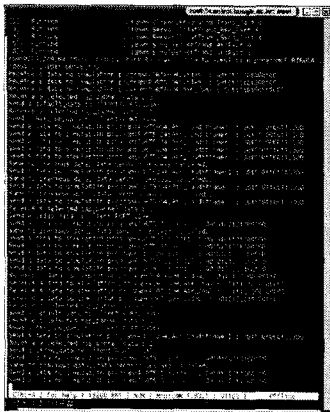


그림 7. minicom을 통한 서버의 결과 확인  
Fig 7. Result of server programming

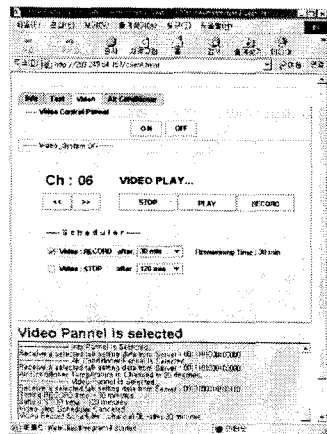


그림 8. 클라이언트 프로그램 결과 확인  
Fig 8. Result of Client Program

웹 브라우저에서 동작하는 클라이언트측 프로그램은 원격지에서 작동상태를 확인하게 된다.

시뮬레이션측은 다른 컴퓨터에서 동작되며 각각의 포트에 대한 전송되는 데이터를 확인/분석 하게 되며,

눈으로 확인하기 쉽게 모델링 하여 준다.

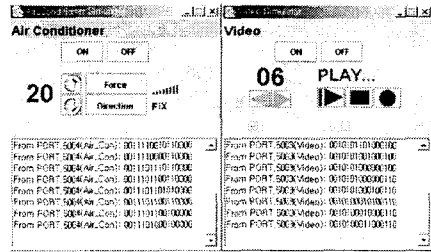


그림 9. 시뮬레이터 프로그램 결과 확인  
Fig 9. Result of Simulator

### VI. 결론

본 논문에서는 웹기반 홈네트워크 시스템의 기반 기술을 획득하는데 그 목적을 두었다.

그 결과 사용자의 편의를 고려한 웹 클라이언트 프로그램을 개발하였으며, 서버 시스템으로 별도의 데스크탑 컴퓨터 대신 임베디드 보드를 이용한 서버 시스템을 개발하였다. 현재 최종단에서의 데이터 전송 확인이 가능하며 실제로 하드웨어로 구성하여 실험해 볼 필요성이 있다.

또한 서버 보안을 위한 보안모델의 설계가 필요하며, 데이터 암호화를 위한 연구 또한 이루어져야 한다.

### 참고문헌

- [1] Sun Microsystems, "Java Technology", [www.sun.com/software/java](http://www.sun.com/software/java), 2003.
- [2] Craig Hollabaugh, Ph.D. "Embedded Linux - Hardware, Software, and Interfacing", Addison Wesley, 2001.
- [3] Merlin Hughes, Michael Shoffner, Derek Hammer, Umesh Bellur, "Java Networking Programming", Manning, 2001.
- [4] "Embedded Linux Board (SWAN-II) 제품정보" [www.itronixit.co.kr](http://www.itronixit.co.kr)
- [5] 리눅스를 이용한 독립장비 개발, [www.kesi.org](http://www.kesi.org)
- [6] Korea Embedded Linux Project, [www.kelp.or.kr](http://www.kelp.or.kr)
- [7] "HardHat Linux 2.0 Journeyman Edition - installation and Reference Guide", MontaVista, 2000.
- [8] 전우형, "Embedded Linux를 위한 Ramdisk image 만들기", [www.netmanias.com](http://www.netmanias.com), 2001.
- [9] 서보원, 최재규, "Java Programming", 사이버, 2003.
- [10] Robert Lougher, "JVM Reference Guide", [javvm.soerforge.net](http://javvm.soerforge.net), 2003.
- [11] Mark Wielaard, "GNU-ClassPath Reference Guide", [www.gnu.org](http://www.gnu.org), 2003.
- [12] Bill Gatliff, "The (very alpha) CrossGCC FAQ" [crossgcc.billgatliff.com](http://crossgcc.billgatliff.com), 2001.
- [13] Scott Howard, "CrossGCC Frequently Asked Questions", [www.objsw.com](http://www.objsw.com), 1999.
- [14] Elliotte Rusty Harold, "JAVAI/O", O'Reilly, 2000.