
임베디드 웹서버를 이용한 빌딩자동화시스템에 관한 연구

배용근* · 김형균** · 정일용*** · 김용호* · 이여진*

A Study on Building Automation System using Embedded Web Server

Yong-Guen Bae* · Hyeong-Gyun Kim** · Il-Yong Jung*** · Yong-Ho Kim* · Ye-Jin Lee*

이 논문은 2006년도 조선대학교 학술연구비의 지원을 받아 연구되었음

요 약

임베디드 웹서버는 여러 가지 형태의 입출력장치와 지능형 정보기기들을 인터넷에 손쉽게 접속할 수 있게 하는 초소형 웹 서버로서, 임베디드 웹서버의 적용 영역은 가정, 빌딩 및 공장 등의 전 영역에 걸쳐 있다. 지능형 정보기기를 인터넷에 접속하여 원격지에서 기기를 감시 및 제어를 하기 위해서는 웹 콘텐츠의 제공을 주목적으로 하는 일반 서버 컴퓨터와는 다른 임베디드 운영체제를 이용한 전용화 된 서버가 필요하다. 본 논문에서는 순수 소프트웨어 솔루션으로 구현되는 웹서버로서 전체적인 시스템의 데모를 구현하는 것을 목적으로 인터넷을 통해 P2P로 연결된 임베디드 센서들을 원격으로 데이터를 추출하고, 이를 제어하는 틀을 구성하는데 초점을 두었으며 ARM intel StrongARM SA-1110 프로세서를 기반으로 한 타겟보드에 임베디드 운영체제로서 리눅스를 포팅 하였고, 네트워킹 기능을 위하여 이더넷을 기반으로 한 기본적인 네트워크 프로그램을 동작 시켰다.

ABSTRACT

A micro embedded did the task to be simply the sequent at the past. The necessity of an embedded operating system did not exist in the reason. We satisfy the complication of the system and operating system have been desired for the officer of the process. The necessity of the process of real-time processing increased also with fast response time to users, The porting follows and have been composed of an real-time embedded operating system. We can control the control unit with the sensor to be established in the building, An embedded web server to use the resource of the at least is appearing a networking facility of interaction. In this paper, The web server to be implemented in a purity software. We draw data for the thing to implement overall demo to the remote through Internet and We put the point even though we organized the frame to control this. we have successfully ported linux on an embedded system, which is based on intel StrongARM SA-1110 processor, then written several network modules of ethernet-based network devices

키워드

Embedded, ARM, Linux

* 조선대학교 컴퓨터공학부

** 동강대학 컴퓨터인터넷과

*** 조선대학교 컴퓨터공학부 / BK핵심사업단 연구원

I. 서 론

첨단 정보화시대, 국제화시대의 도래와 함께 각종 건축물들이 지능화 단계에 이르러 인체의 두뇌에 해당되며 중앙처리할 수 있는 시스템을 요구하게 되었다. 이와같은 환경요구에 능동적으로 대처하고, 환경개선의 한 방법으로 빌딩자동제어 구축이 절실히 요구되고 있다[1][2].

임베디드 웹서버는 여러 가지 형태의 입출력장치와 지능형 정보기기들을 인터넷에 손쉽게 접속할 수 있게 하는 초소형 웹 서버로서, 임베디드 웹서버의 적용 영역은 가정, 빌딩 및 공장 등의 전 영역에 걸쳐 있다. 지능형 정보기기를 인터넷에 접속하여 원격지에서 기기를 감시 및 제어하기 위해서는 웹 컨텐츠의 제공을 주목적으로 하는 일반 서버 컴퓨터와는 다른 임베디드 운영체제를 이용한 전용화 된 서버가 필요하다[3].

본 연구에서는 소프트웨어 솔루션으로 구현되는 웹서버로서, 인터넷을 통해 P2P로 연결된 임베디드 센서를 이용해 원격으로 데이터를 추출하고, 이를 제어하는 틀을 구성하는데 초점을 두었다. 또한 실시간기능을 위하여 리눅스 기반위에서 동작하는 실시간 운영체제인 RTLinux를 포팅하여 실시간 운영체제상에서의 네트워킹 기능을 구현하여 최적화된 임베디드 웹서버와 Hybrid P2P를 이용하여 빌딩 자동화에 필요한 각각의 센서들의 정보를 모니터링과 제어 할 수 있도록 제안하고자 한다.

II. 임베디드 웹서버

2.1 임베디드 운영체제

임베디드 시스템이 가지는 특성 중 실시간 적인 요소를 충족하기 위한 것이 실시간 운영체제라 할 수 있는데, 임베디드 RTOS(Real Time OS)는 주어진 태스크를 달성하기 위한 목적뿐만 아니라 그 태스크의 종료시한을 만족시키느냐 못 시키느냐의 문제에 중점을 둔 시스템이다. 또한 각 태스크의 응답시간을 실시간으로 처리 해주는 것이다. 그렇지만 엄밀히 말한다면 멀티태스킹 환경에서 모든 작업을 즉각적인 실시간으로 처리할 수는 없는 것이다. 각 태스크를 분할 처리하여 모든 작업을 실시간으로 처리하는 것처럼 보이는 것이다. 가장 중요한 종료시한을 넘기지 않도록 하기 위해서는 각 태스크에 대한 우선순위를 부여해야 한다. 그래서 모든 태스크들의 우선순위에 따라

서 작업이 스케줄되고 실행되어야 한다. 또한 실행되는 중에 더 높은 우선순위의 작업이 스케줄될 수 있기 때문에 선점이 가능해야 하는 점도 고려되어야 한다. 그림 1에서는 리눅스를 이용한 개발 환경을 표현 해보았다.

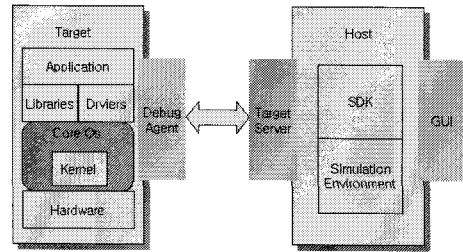


그림 1. 임베디드 개발 환경
Fig. 1. Embedded Development Environment

2.2 임베디드 웹서버

대표적인 임베디드 웹서버의 [2][3][4] 형태는 크게 네 가지 형태로 나뉜다.

첫째, Protocol Conversion Web-Server

I/O를 통하여 들어오는 정보를 중앙의 PC/Server에 저장한 상태에서 사용자가 접속시에 정보를 제공하는 방식으로 Software에 의한 Web-Server이다.

둘째, Distributed Web-Server

현재 대부분의 임베디드 웹서버들은 PC가 워크스테이션을 경유하여 네트워킹에 연결된다. 웹서버는 최소한으로 요구되는 정보를 RS-232 통신 포트를 통해 PC 웹서버의 정보를 수집한다. HTML 방식에서는 기기에 연산능력과 메모리 사이즈를 요구하지 않는다는 것이 장점이지만 PC를 사용함으로써 경제적이다[3]. 예를 들면 간단한 센서의 정보를 위해 PC를 웹서버로 사용한다면 비용도 비싸고 큰 공간을 차지하게 되므로 비효율적인 것이다. Web-Server는 적은 정보와 분산 형태의 제어를 필요로 하는 Home Net-working 부분이나 빌딩내의 사무기기 등의 감시 제어 분야에서 Machine Monitoring 부분이다.

셋째, Protocol-on-Chip

TCP, UDP, IP, DHCP, ARP, ICMP와 같은 TCP/IP 프로토콜 스택과 데이터 링크제어와 MAC 프로토콜과 같은 이더넷 프로토콜을 하나의 칩에 내장하는 형의 웹서버이다. TCP/IP 프로토콜 칩의 기간이 길어져 전체적인 개발 시간이 길어지지만, 소프트웨어로 구현하던 종래의 방법에 비해 사용자의 시스템에 이를 구현함으로써 인터넷을

빠르게 구현할 수 있게 해주어 소프트웨어 개발비용을 절감할 수 있다.

넷째, Web-Server With/Without real-time O/S

real-time O/S 마이크로 초 이내의 응답의 정확도는 물론 신뢰성과 더불어 원하는 시간 안에 필요한 작업을 수행하여야 하는 시스템, 즉 항공, 군수 산업같이 세밀한 제어를 보장해야 하는 시스템이나 서버에 이용하는 운영체제이다. 이러한 형태의 웹서버는 O/S 제공으로 개발이 용이하다. 하지만 정밀성과 실시간성에서 우수하지만 고가의 라이선스 비용 등 가격 면에서 소형의 저가형태의 제품생산에는 경쟁력이 약하다[4].

Ⅲ. 임베디드 웹서버를 이용한 빌딩자동화시스템

빌딩 자동화 시스템은 전력, 공조, 조명, 엘리베이터 시스템 등이 해당된다. 이들 각 시스템은 빌딩 내 주요 에너지원을 소비하는 시스템들로서 시스템간 상호운영성의 필요성이 우선적으로 요구된다. 과거에는 각 시스템의 독자적인 프로토콜에 기반한 인터페이스에 의해 통합이 수행되어, 개발에 필요한 비용과 기간이 길어지고, 기존 업체에 대한 종속 문제가 발생되었다. 따라서 최근에는 이러한 각 시스템을 통합해서 원격감시 및 상호 연동에 의한 최적 제어를 통하여 빌딩에 대한 효율적인 운전을 가능하게 하고 있다.

본 연구에서는 임베디드 시스템을 구현하기 위하여 Intel Strong ARM SA-1110프로세서 기반의 레퍼런스 보드에 리눅스를 포팅하고 실시간 운영체제로의 기능을 위하여 RTLinux를 포팅하여 빌딩자동화 시스템을 구성했다.

3.1 크로스 컴파일 환경

임베디드 시스템에서의 개발은 레퍼런스 보드상에서 이루어질 수 없다. 특히 운영체제를 포팅하는데 있어서 커널을 수정하고 재컴파일 하면서 보드에 맞는 환경을 구축하고 보드의 각종 디바이스 드라이버들을 적재하는데 작업을 할 수 있는 환경이 필요하다. 그러한 과정들을 보드상의 협소한 메모리만 가진 레퍼런스 보드 상에서 작업할 수 없다.

표 1. 네트워크 호스트 환경
Table 1. Host environment of Network

구분 \ 시스템	Window	Linux	호스트 PC	target board
용도	dhcp client	dhcp client	개발용 PC	dhcp server
운영 체제	Window xp	Redhat 9.0	Redhat 9.0	embedd linux
RAM용량	512mb	256mb	256mb	32mb
NIC card	리얼텍 8139	리얼텍 8139	리얼텍 8139	CS8900C3
네트워크 정보	N/A	N/A	IP: 210.182.203.201	IP: 210.182.203.202

따라서 레퍼런스 보드와 통신하면서 작업을 할 수 있는 호스트 PC가 필요하다. 호스트 PC로 사용되는 PC는 일반적으로 CPU의 환경이 Intel Pentium 계열이므로 그 환경에서 그대로 작업을 하면 ARM계열의 CPU에서는 동작을 하지 않는다. 운영체제나 응용프로그램의 개발에 있어서 보드의 환경에 맞춰서 호스트 PC의 환경을 구축해 주어야 한다.

본 연구에서는 ARM계열의 CPU를 가진 보드에서 개발을 해야 하므로 호스트 PC로 사용되는 컴퓨터에 ARM 환경을 위한 크로스 컴파일러 환경의 구축이 필요하다. 포팅하고자 하는 리눅스 운영체제를 임베디드 시스템의 목적에 맞게 커널을 재구성하여 크로스 컴파일러 환경에서 재컴파일 해야 한다.

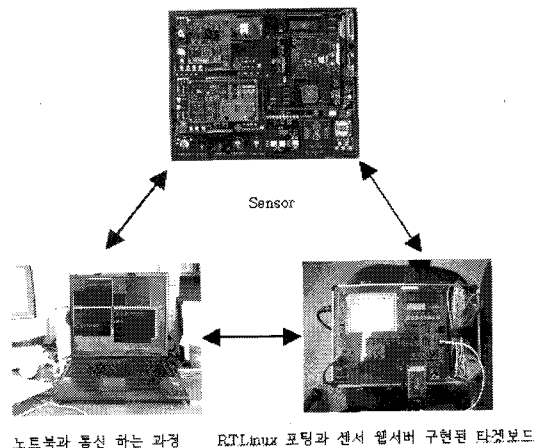


그림 2. 빌딩 자동화시스템의 통신과정
Fig. 2. Communication of Building Automation System

그림 2는 건물 실내 정보를 파악 할 수 있는 센서와 타켓 보드에 웹으로 접근하고 있는 과정을 나타낸 것이며 통신을 실시간으로 원활하게 작동 할 수 있도록 리눅스 기반위에서 동작하는 실시간 운영체제인 RTLinux를 포팅하여 실시간 운영체제상에서의 네트워킹 기능을 구현하여 최적화된 임베디드 웹서버를 제안 했다.

크로스 개발 환경이 갖춰진 호스트 PC에서 컴파일 되고 개발된 응용프로그램들을 시리얼 케이블을 통해 레퍼런스 보드로 다운로드 하고 보드에서 출력되는 내용을 노트북으로 모니터링한다. 서버에서는 인터넷을 통해 클라이언트와 빌딩모델간의 데이터 전송을 위한 중간 매개체로서 클라이언트에서 요청한 제어 신호를 빌딩 모델에 전달하고, 접속자의 수를 모니터링하며 또한 빌딩 모델의 상태정보를 체크하여 클라이언트에 전달하는 역할을 한다.

구축된 웹서버의 경우는 Distributed Web Server는 I/O를 통해 연결된 외부기기의 상태를 웹 페이지 형식으로 저장하고, 인터넷을 통해 접속된 관리자에게 그 정보를 제공하는 서버이다.

3.2 MMI 기반의 모니터링 시스템 구성

MMI(Man Machine Interface)란 사람과 기계의 대화를 도와주는 장치 또는 기능을 두고 MMI라 한다. 과거에는 기계를 전문 기술자만 다룰 수 있는 것이라고 생각해 왔는데 특히 기능이 복잡하고 조작버튼 등이 많으면 숙련된 기술자만 다룰 수 있었다. 하지만 지금은 컴퓨터 분야를 비롯한 과학기술이 하루가 다르게 발전하고 있고 이러한 첨단기술은 비전문가라도 기계를 다룰 수 있도록 도와주고 있는 도구이다. 본 연구의 모니터링을 위해 MMI라는 프로그램이 사용된다. 간단히 구조를 살펴보면 첫째로 시스템 구성의 입출력할 포인터에 대한 정보를 저장하는 DB(Data Base)가 있다. 그림 3의 a는 시스템에 사용되는 등록된 포인터를 나타내고 있다. 둘째로 MMIVIEW의 부분을 그려주는 윈도우즈 보조 프로그램의 그림판과 비슷한 그래픽 에디터가 있으며 그 에디터로 입출력할 포인터 및 버튼을 구성하게 된다. 그 구성에 대한 DB에 설정되어 있는 입출력 포인트 정보와 연결함으로써 DB의 내용이 업데이트 되면 해당 포인터가 변화하게 된다. 그림 3의 b는 그래픽 에디터로 그린 포인터에 대한 DB의 정보 포인터와 연결 시켜주는 것을 나타내고 있다.

셋째로 공유 메모리를 생성해주는 프로그램이 있으며

공유메모리의 순서와DB의 인덱스번호가 서로 연결되어 있다. 따라서 해당 공유메모리에 저장함으로써 DB에 설정된 해당 포인터의 내용이 업데이트된다. 이처럼 본 연구에 MMI 기반의 제어와 모니터링을 하기 위한 포인터의 DB설정과 MMI View화면 구성 그리고 통신을 하기 위한 MMI방식의 인터페이스가 추가된 TCP/IP 클라이언트 프로그램을 작성하였다.

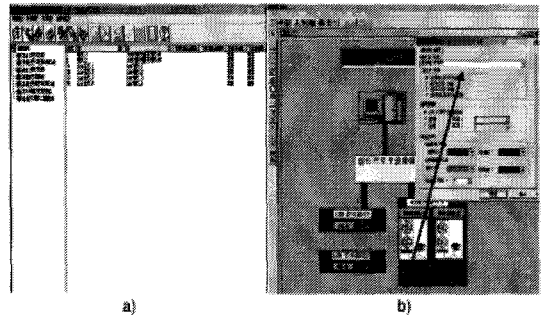


그림 3. MMI 환경 설정
Fig. 3. Environment setting for MMI

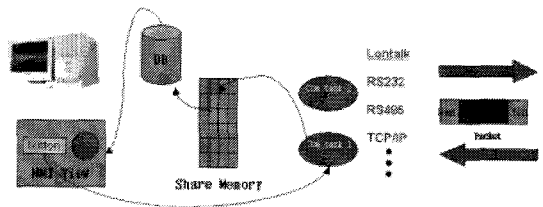


그림 4. MMI의 구성
Fig. 4. Construction of MMI

TCP/IP 방식의 클라이언트 프로그램을 작성할 때 MMI View와 연동 하기 위해서 다음과 같은 기능을 넣어야 한다. 받은 데이터를 해당 공유메모리에 쓰는 인터페이스 부분과 MMI의 제어 버튼을 누를 때 발생하는 해당 레지스터 윈도우즈 메시지 핸들러 함수를 작성하였다. 또한 주기적인 상태를 요청하는 부분이 필요하므로 이것은 타이머 기능을 이용하였다. 그림 5는 본 논문에서 구축된 시스템에서 소프트웨어적 동작 흐름을 나타낸다. 먼저 시스템의 파워를 온하면 DPRAM에 하부 제어장치의 초기 상태 값이 저장되게 된다. 다음으로 클라이언트는 다중 프로토콜 제어기의 서버 프로그램에 접속하여 타이머 기능에 의해서 설정된 시간 주기로 로컬 제어기의 상태 값이 저장되어 있는 값을 가져오게 된다. 그 값을 가져온

클라이언트는 데이터 베이스에 등록된 제어 포인트의 인덱스번호와 매핑 되어 있는 공유메모리에 상대 값을 라이트 함으로써 해당 제어기의 상태를 MMI View에 나타내게 된다.

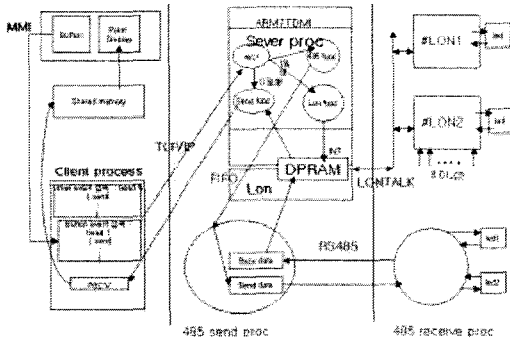


그림 5. 소프트웨어 흐름도
Fig. 5. Software flowchart

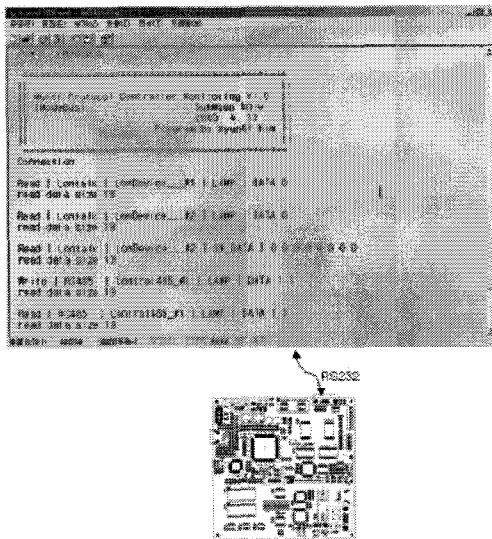


그림 6. 제어기의 서버 모니터
Fig. 6. Server monitor for controller

그림 6은 다중 프로토콜 제어기의 서버프로그램이 동작 상태를 보여 주는 화면이다.

IV. 결 론

본 논문에서는 인터넷을 기반으로 원격으로 기기의 상태에 대한 정보를 접속된 시스템 관리자에게 제공하고, 관리자의 지시를 받아들이기 위한 임베디드 웹서버를 구축하였다. 인터넷을 통한 빌딩이나 장비 등의 원격제어를 할 수 있는 Web-Server 개발의 기반이 되었으며 이를 이용해 가전제품, 사무기기, 건물 경비 등의 모니터링용으로 사용될 수 있다.

실시간기능을 위하여 리눅스 기반위에서 동작하는 실시간 운영체제인 RTLinux를 포팅하여 실시간 운영체제 상에서의 네트워킹 기능을 구현하여 최적화된 임베디드 웹서버와 Hybrid P2P를 이용하여 빌딩 자동화에 필요한 각각의 센서들의 정보를 모니터링과 제어 할 수 있도록 구현했다. 웹 감시 기능이 추가된 실시간 임베디드 리눅스 시스템을 구성하는 데에는 리눅스에 대한 전반적인 이해가 필요하고 필요한 RT 커널을 개발할 수 있는 고급 개발자가 필요하다는 단점도 있지만 작은 비용으로 효율적인 실시간 시스템을 구성할 수 있다는 점에서 무척 의미 있는 일이며 이로 인한 임베디드 시스템의 응용 분야는 모든 분야를 걸쳐 실로 방대하다고 할 수가 있다.

참고문헌

- [1] Bill Gallas, Vandana Verma, "Embedded Pentium Processor System Design for Windows CE", IEEE, 1998.
- [2] Bill Venners, Inside JAVA Virtual Machine, McGraw-Hill, 1998
- [3] David A Rusling, "The Linux Kernel", 1999.
- [4] Gary Nutt, "Kernel Project for Linux", Addison Wesley, pp.133-144
- [5] James Y. Wilsons, Aspi Havewala, "Building Powerful Platforms with Windows CE", Addison-Wesley Publishers, 2001
- [6] Moshe Bar, "Linux File System", McGraw-Hill, 2001
- [7] <http://www.embedded.com>, "Linux as an Embedded Operation System".
- [8] <http://www.emlinux.com>, "Embedded Linux in a Commercial Product".

저자소개



정 일 용(II-Yong Jung)

1983년 2월 한양대학교 공과대학
공학사
1991년 2월 City University of New
York 전산학박사

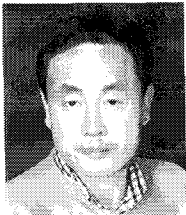
1994년 2월~1997년 11월 한국전자통신연구소 선임연구원
1999년~2000년 10월 조선대학교 정보전산원장
※관심분야: 정보보안, 암호학

김 용 호(Yong-Ho Kim)

1993년 8월 경남대학교 전자계산과 공학석사
2005년 2월 조선대학교 전자계산학과 이학박사
2006년 현재 중앙정보기술연구소 소장
※관심분야: 임베디드 리눅스, P2P, 데이터베이스

이 여 진(Yeo-Jin Lee)

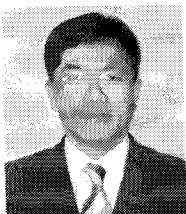
2003년 2월 조선대학교 교육대학원 교육학석사
2006년 2월 조선대학교 대학원 박사수료
※관심분야: 임베디드 리눅스, P2P



배 용 근(Yong-Guen Bae)

1984년 2월 조선대학교 컴퓨터
공학과 공학사
1987년 2월 조선대학교 대학원
공학석사

1993년 2월 원광대학교 대학원 공학박사
1997년~2006년 현재 조선대학교 컴퓨터공학부 교수
※관심분야: 마이크로프로세서, 프로그래밍언어



김 형 균(Hyeong-Gyun Kim)

1998년 2월 조선대학교 대학원
공학석사
2004년 2월 조선대학교 대학원
컴퓨터공학과 공학박사

2002년~2006년 현재 동강대학 컴퓨터인터넷과
초빙전임강사
※관심분야: 임베디드 리눅스, 모바일컨텐츠, P2P