

스마트 빌딩 시스템을 위한 임베디드 웹서버의 디자인

유종일, 신승우, 장경배, 심일주, 박귀태
고려대학교 전기공학과

Design of the Embedded Webserver for Smart Building System

Jong-Il Yu, Seung-Woo Shin, Kyung-Bae Chang, Il-Joo Shim, Gwi-Tae Park
Electric engineering, Korea University

Abstract - Recently, research about optimizing network of IBS(Intelligent Building System) is in progress activity. When we install existent building control network using general PC, system's efficiency drops. Because server that is charged with the task process is not and is expensive. Also, CapEx(Capital Expenditure) rises because of setting up the network equipment to integrate different protocol in building system. In this paper, we suggests the embedded web server that has various network structure.

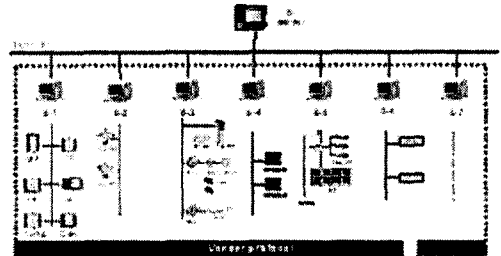
1. 서 론

우리가 현재 살고 있는 사회는 지식정보화 사회로 변하고 있다. 이에 정보화 사회의 기반이 되는 시설도 함께 정보화 되어 건물의 지능화 산업이 발전하였다.

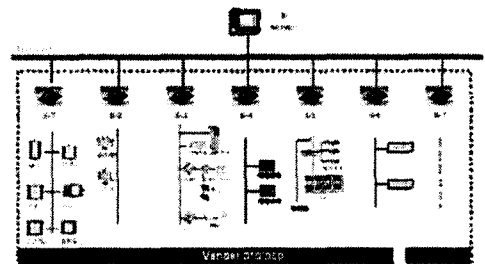
빌딩 내 네트워크 환경도 멀티미디어 시대에 따라 ATM이나 기가비트 네트워크가 보편화 되었다. 하지만 스마트 빌딩 시스템의 발전이 네트워크 환경과 함께 급속히 이루어짐에 따라 스마트 빌딩 시스템 구축시 네트워크 구조를 생각하지 않은 시스템 설치가 이루어져 왔다. 또한 한 빌딩 내에 설비별 프로토콜들도 다양화 되어 시스템 통합시 값이 비싼 router와 gateway등의 사용이 요구된다.

현재 빌딩 시스템은 오픈 프로토콜(LonWorks, BACnet, etc)을 사용하여 통합은 되었지만 각 설비의 서버에는 설비를 모니터링 하는 감독자가 있어야 하는 고용 부분에서 부정적 측면도 있다. 빌딩내에 설비들의 종류가 많을 경우는 PC 급 서버들이 설비 수만큼 설치되어 많은 초기투자비용이 요구되며 설치장소도 제어환경에 따라 한정되어 있으므로 범용으로 사용되는 통신 시스템으로서의 구축이 어렵다.

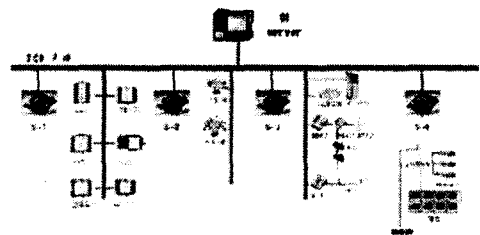
본 논문에서는 스마트 빌딩 시스템에서의 임베디드 웹서버를 제안한다. 이 임베디드 웹서버는 빌딩에서의 SI(System Integration) 서버가 아닌 각 설비의 서버를 의미하며 보다 상세하게는 빌딩 제어기의 보수 또는 점검을 위한 모니터링을 인터넷 통신망을 이용하여 모바일 기기로 원격수행을 할 수 있는 임베디드 웹서버 시스템이다. 임베디드 웹서버의 장점으로는 값이 싸고 task 처리를 전달하기에 수행능력이 일반 PC보다 우수하다. 그리고 각종 프로토콜을 내장할 수 있어 빌딩 시스템 통합에 기여할 것이다. 또한 필요에 따라 모니터링도 할 수 있어 고용적인 측면과 내장형의 웹서버를 구축하여 설치환경에 제한이 없이 설치할 수 있다는 장점이 있다. <Fig.1>은 종래의 빌딩 시스템을 보여주고 있으며 <Fig. 2>는 임베디드 웹서버를 적용한 시스템 구조이다. 더 나아가서 컨트롤러와 서버사이의 프로토콜이 Ethernet TCP/IP가 되는 시스템 연구가 진행되고 있다. 이 경우 종래에 각 시스템이 <Fig. 1> 과 같이 구성할 수밖에 없는 단점을 <Fig. 3> 와 같이 자유로운 네트워크 구조를 가질 수 있어 보다 많은 연동서비스와 기능을 가질 수 있다는 장점도 가지고 있다.



<Fig. 1> 현재의 빌딩 시스템 구성



<Fig. 2> 임베디드 웹서버를 적용한 빌딩 시스템



<Fig. 3> TCP/IP 기반의 임베디드 웹서버를 적용한 빌딩 시스템 구성

따라서 본 논문에서는 스마트 빌딩 시스템에서의 실시간 모니터링과 제어를 하는 임베디드 웹서버를 제안한다.

2. 본 론

2.1 연구의 범위

<Fig. 1>에서 보이는 바와 같이 현재 스마트 빌딩 시스템은 다양한 오픈 프로토콜로 이루어져 있다. <Fig. 3>의 빌딩 시스템이 가능한 것은 각 설비 서버의 하단에 위치하고 있는 컨트롤러(DDC)들이 각각 IP 를 가지고 Ethernet

TCP/IP 통신을 할 수 있다는 전제가 있기 때문이다. 그리하여 SI 서버와 컨트롤러(DCC or Embedded device)간, 컨트롤러와 각 설비 임베디드 서버간, 컨트롤러와 컨트롤러간의 통신이 가능하게 되는 것이다. 이러한 시스템을 이루기 위해서는 센서와 컨트롤러간이 아닌 컨트롤러상단에서부터 SI서버간이 Ethernet TCP/IP 프로토콜을 사용해야 한다.

스마트 빌딩시스템에서의 Ethernet TCP/IP 프로토콜을 이용한 시스템 통합은 많은 연구가 진행되고 있다[1]. Ethernet TCP/IP 프로토콜은 현재 전 세계에 퍼져 있으며, Ethernet TCP/IP 프로토콜의 사용으로 종전의 각 설비 시스템의 Ethernet 접속을 위한 gateway가 필요 없게 된다는 장점도 지닌다[2]. 이 밖에도 스마트 빌딩 시스템에서의 제어 네트워크로 TCP/IP의 사용이 문제를 없음을 보여준다[3,4,5]. 따라서 본 논문에서는 Ethernet TCP/IP 프로토콜을 이용한 빌딩 시스템에서의 임베디드 웹서버 제안으로 범위를 제한한다.

물론<Fig. 3>의 빌딩 시스템이 구성이 아닌 PC급을 웹서버로 사용한 <Fig. 1>의 빌딩 시스템에서 PC급 웹서버를 임베디드 웹서버로 바꿔서 사용하더라도 SI에는 영향을 주지 않는다. 즉 Ethernet TCP/IP 프로토콜이 아닌 다른 빌딩 오픈 프로토콜을 사용해도 임베디드 웹서버의 기능에는 문제가 없음을 알려준다

2.1.1 임베디드 웹서버 디자인

임베디드 웹서버는 시스템의 모니터링과 컨트롤을 위한 실행코드와 인터넷 소프트웨어를 포함한 마이크로 컨트롤러이다. 임베디드 웹서버의 기능을 가지기 위해서는 다음과 같은 것을 필요로 한다. 먼저 적은 메모리로 인해 메모리를 좀더 효율적으로 구동하기 위한 미리 분할된 메모리 블록 구조나 고정된 메모리의 사용이 필요하며, 두 번째 Ethernet TCP/IP 프로토콜을 사용하여 서브시스템 (DCC, 임베디드 디바이스, 센서)으로부터의 정보를 웹 브라우저를 통하여 보여주기 위한 동적인 페이지 구현이 필요하다. 그리하여 웹 브라우저에서의 정보가 실제 서브시스템의 정보가 되어야 한다. 마지막으로 웹서버는 저장하기 위한 disk driver를 가지지 않으므로 웹페이지, 데이터, 컨트롤 정보를 저장하기 위한 확장 disk drive나 메모리가 필요하다[6].

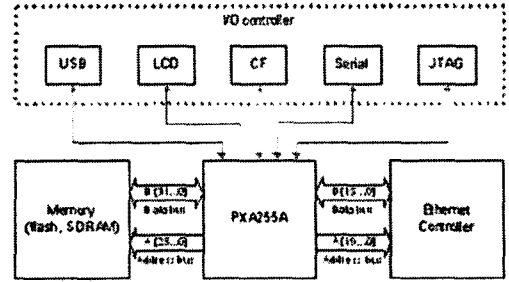
제한하고자 하는 임베디드 웹서버는 일반 웹서버의 기능을 유지하면서 서브시스템으로부터 컨트롤 및 데이터 정보를 필요에 따라 SI 서버로 보내주고, 다른 설비시스템의 웹서버 및 컨트롤러와 연동하는 시스템이다. 이렇게 함으로서 Ethernet TCP/IP통신에 의해 웹서버가 다운로드될 때에도 연동을 가능케 하는 시스템 디자인이 필요하다.

다음 부분에서는 제한하고자 하는 임베디드 웹서버 시스템의 하드웨어 및 소프트웨어, 시스템 디자인을 설명한다

2.1.1.1 Hardware concept

하드웨어는 크게 3가지로 구분된다. ARM 프로세서와 메모리, 인터페이스 확장 I/O 컨트롤러로 구성된다. ARM 프로세서는 32-bit 마이크로 프로세서로 400MHz의 속도를 지닌다. 시스템 부트 및 정보의 저장을 담당하는 메모리는 flash ROM 과 flash ROM과 fusing하게 되는 64MB의 SDRAM으로 구성되어 있다. I/O 컨트롤러에는 RS-232, USB 컨트롤러, Ethernet 컨트롤러, CF 컨트롤러, LCD 컨트롤러를 지닌다. CF 컨트롤러는 웹의 HTML과 Java를 저장하며 컨트롤 및 데이터 정보를 보관하는 확장 메모리를 컨트롤 하게 된다. USB 컨트롤러는 개발 환경에서 host와 target간의 데이터를 주고 받는 데 쓰이는 디버거용으로 사용되며 버전은 1.1이다. Ethernet은 10Mbps를 지원하고 Bootloader와 Kernel의 전송에 사용되어진다. LCD 컨트롤러는 시스템 개발에 이루어지는 디버거용으로 사용되어지게 된다.

아래 <fig. 4>은 하드웨어 구조를 보여준다.



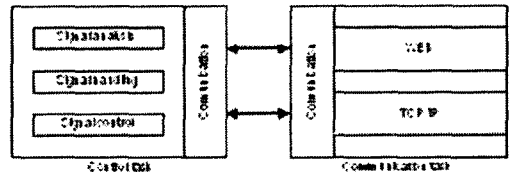
<Fig. 4> 임베디드 웹서버의 하드웨어 구조

2.1.1.2 Software concept

어플리케이션은 결합포용을 주기 위해 독립된 제어(task와 통신 task로 나누어 구성한다[7].

제어 task에는 크게 3부분인 신호 분석, 신호 핸들링, 신호 컨트롤부분으로 나뉜다. 신호분석에서는 시스템에서 웹서버 하단의 서브 제어기나 어느 설비의 컨트롤러, 혹은 어느 설비 서버들중 어느 컨트롤러로부터 어떠한 신호가 왔는지에 대한 신호 분석을 하게 된다. 신호 핸들링에서는 판별된 신호를 어떻게 분석할지에 결정하게 되며 마지막으로 신호 컨트롤에서 분석한 신호를 어느 컨트롤러에 보낼 것인지를 판별하여 웹상에 보낼 신호와 나누게 된다.

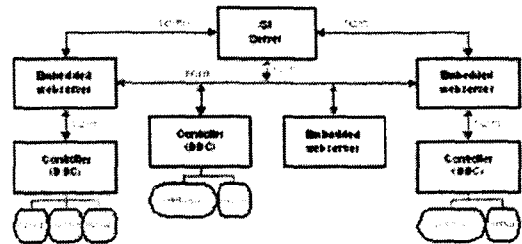
이렇게 받은 신호는 통신 task를 통하여 웹상과 Ethernet TCP/IP 신호로 변환하여 각 컨트롤러로 전송된다. <Fig 5>은 임베디드 웹서버의 소프트웨어 다이어그램을 보여준다.



<Fig. 5> 임베디드 웹서버의 소프트웨어 구조

2.1.1.3 System concept

위에서 임베디드 웹서버에 하드웨어와 소프트웨어를 설명하였다. 이러한 하드웨어와 소프트웨어를 내장한 임베디드 웹서버를 <fig. 3>에서의 구조와 같이 빌딩 시스템을 구성한다. 각 설비의 임베디드 웹서버는 다른 설비의 임베디드 웹서버와 Ethernet TCP/IP를 이용하여 컨트롤 정보 또는 데이터 정보를 주고받게 된다. 각 설비의 컨트롤러 또한 임베디드 웹서버와 컨트롤 및 데이터 정보를 주고 받게 된다. 컨트롤러는 임베디드 웹서버의 상태에 따라서 데이터 및 컨트롤 정보를 분류 및 샘플링을 하여 SI 서버에 업로드 한다. <Fig. 6>는 시스템 구성 블록 다이어그램을 나타낸다.



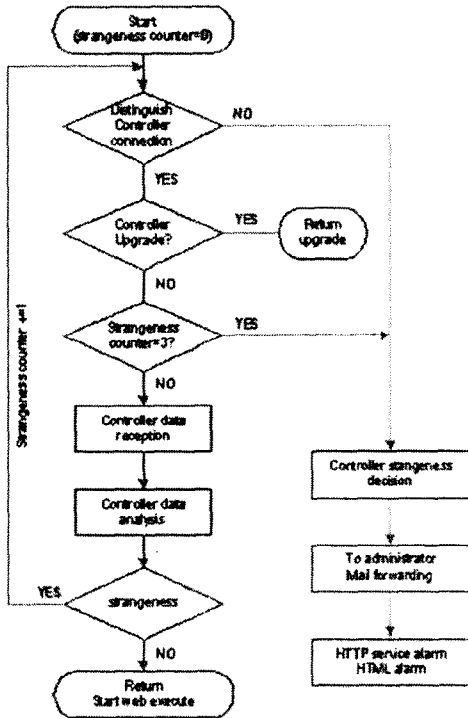
<Fig. 6> 임베디드 웹서버의 시스템 구성도

2.2 임베디드 웹서버의 세부기능

지금까지 임베디드 웹서버의 하드웨어와 소프트웨어 그

리고 시스템 디자인에 대해서 설명하였다. 임베디드 웹서버는 다음과 같은 flow 기능을 갖는다.

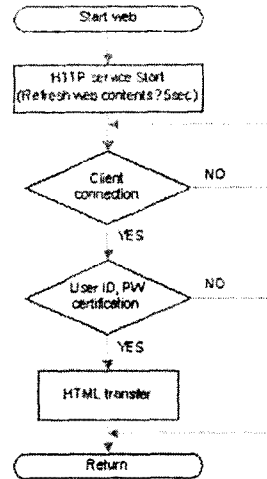
<Fig. 7>에서 프로세스가 시작이 되면 임베디드 웹서버간, 임베디드 웹서버와 컨트롤러간, SI 서버와 임베디드 웹서버간의 연결을 정의하게 된다. 연결이 이루어 지지 않았을 경우 메일서버를 통해 관리자에게 이상을 알리게 된다. 그러나 연결이 이루어지면 관리자의 환경 설정값 변경이 없는 경우, 제어기와 통신을 시작하게 된다. 제어기와의 통신에서 이상이 있을 경우, 계속되는 통신 에러에 대해 카운터를 해서 내번이상이 연속 에러로 판명되면 메일서버를 통해서 관리자에게 제어기 이상을 통보하게 된다. 또한 HTTP 서비스의 이상을 알림으로 웹에서의 확인이 가능하게 된다. 제어기의 이상이 구동이 되면 HTTP 소프트웨어를 실행하여 웹 서비스를 시작함과 동시에 다시 <Fig. 7> 루프를 반복 수행하게 된다. 이때, 관리자가 설정값을 바꾸게 되면 upgrade 루프인 <Fig. 9>을 실행하게 된다.



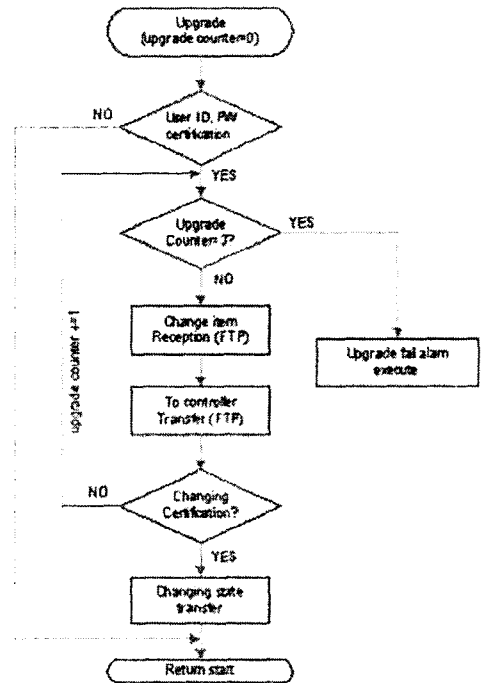
<Fig. 7> 제어기의 이상여부 판단 플로우 다이어그램

우선<Fig. 7>의 과정이 제어기의 이상 없이 이루어지면 <Fig. 8>의 HTTP 소프트웨어가 실행되어 웹기능을 갖게 된다. 웹 서비스가 시작되면 계속적인 웹 정보를 5초 간격으로 리프레쉬 시켜 웹 서비스 정보의 신뢰도를 높인다[8]. 클라이언트의 접속을 대기하다가 클라이언트가 접속을 시도하게 되면 클라이언트의 인증과정을 거쳐 웹 페이지를 화면상에 표시해주며 5초 간격으로 계속적인 리프레쉬가 이루어지게 된다. 만약 클라이언트가 로그아웃을 하게 되면 다시 처음 시작지점으로 돌아가게 되어 다음 클라이언트의 접속을 기다리게 된다. 하지만 클라이언트가 접속했다 하더라도 인증과정에서 실패하면 다음 클라이언트를 맞이하는 처음 상태로 돌아가게 된다.

<Fig. 7>에서 관리자가 컨트롤러의 환경을 바꾸어 주게 되면 upgrade 라는 루프<Fig. 9>로 점프하게 된다. 예를 들어 공조에서 관리자가 직접 혹은 인터넷을 통해 내부 온도를 설정온도에서 변경하게 되면 바로 이 루프가 실행되는 것이다.



<Fig. 8> HTTP 서비스의 시작 플로우 다이어그램



<Fig. 9> 컨트롤러의 환경 변경 설정 플로우 다이어그램

<Fig. 7>에서 관리자가 컨트롤러의 환경을 바꾸어 주게 되면 upgrade 라는 루프<Fig. 9>로 점프하게 된다. 예를 들어 공조에서 관리자가 직접 혹은 인터넷을 통해 내부 온도를 설정온도에서 변경하게 되면 바로 이 루프가 실행되는 것이다. 설정이 변경되면 관리자 인증을 거쳐서 FTP를 통해 제어기에 다운이 되며 다운받은 제어기는 설정값 변경의 여부를 다시 임베디드 웹서버에 알리게 된다. 이 과정에서 변경이 되지 않으면 연속 3번까지 계속 웹서버를 통해 변경사항을 다운받고 수정된 사항을 웹 서버에 알리는 작업을 하다가 4번의 실패가 이루어지게 되면 환경 설정에 실패했다는 알람을 울리게 된다. 만약 환경설정을 컨트롤러가 바꾸었다는 신호가 오게 되면 서버는 다시 <Fig. 7>

의 과정을 시작한다.

3. 결 론

이와 같이 현재 스마트 빌딩 시스템의 각 설비들의 서버를 맡고 있는 일반 PC 급에서의 수행을 저비용, 고성능의 내장형 임베디드 웹서버가 대체 가능하다. RS-232, USB, PCMCIA 등의 각 프로토콜을 통합하는데 용이하며, 설치가 유용하다. 또한 임베디드 웹서버는 저전력으로써 에너지 효율에도 큰 기여를 한다. 앞으로는 이러한 임베디드 시스템이 빌딩뿐 아니라 유비쿼터스 시대에 발맞추어 가는데 가전 및 제어 시장에 큰 기여를 할 것으로 기대된다. 향후 모바일로의 임베디드 웹서버의 접속으로 보다 통합된 스마트 빌딩 시스템을 구상해 보고 일반PC의 사용에서의 성능 차이를 시험해 본다.

[참 고 문 헌]

- [1].Sun Jun-ping, Sheng Wan-xing, Wang Sun an and WU Ke-gong, "Substation Automation High speed Network Communication Platform Based on MMS + TCP/IP + Ethernet", IEEE, 2002
- [2].John Petze, VP product Development, "Powering Smart Energy Networks Now", Article, 03 2000, [online] Available : <http://automatedBuilding.com>
- [3].Mike Donlon, Director of Reesearch and Development Computrols inc, "Using Standard Internet protocols in Building Automation", Networked controls white Papers, [online] Available : <http://automatedBuilding.com>
- [4].Dave VanGompel, Peter Cleaveland, "Moving popular industrial protocols to Ethernet TCP/IP", 06 2001, [online] Available : <http://controlsolutionmag.com>
- [5]. "Understanding and Evaluating Ethernet and TCP/IP Technologies for Industrial automation", 2000, [online] Available : <http://www.ManagetherealWorld.com>
- [6].Darko Stipanicev, Jadranka Marasovic, "Networked Embedded Greenhouse Monitoring and Control", IEEE, 2003
- [7].Christian Eckel, Georg Gaderer and Thilo Sauter, "Implementation Requirements for Web-enabled Appliances a Case Study", IEEE, 2003
- [8].Abhulash Ramakrishnan, "16-Bit Embedded Web Server", IEEE, 06 2004