

# 기기 제어를 위한 저가의 초소형 임베디드 웹 서버

오 민 정<sup>†</sup> · 임 성 략<sup>††</sup>

## 요 약

대부분의 기존 임베디드 웹 서버 시스템은 어떤 특정 기기를 감시하고 제어하기 위하여 고안되고 있다. 따라서 이들은 범용성 및 유연성이 없을 뿐만 아니라 비교적 가격이 저렴한 가정용 기기를 위해서는 너무 비싸다. 이러한 어려움을 극복하기 위하여 본 논문에서는 인터넷을 이용하여 가정용 혹은 산업용 기기를 감시 및 제어하기에 적합한 저가의 초소형 임베디드 웹 서버의 모델을 제시한다. 제시한 모델은 H/W 및 S/W의 비용을 최소화하기 위하여 ISP 기능과 플래시 롬을 내장한 원 칩 마이크로프로세서를 기본으로 한다. 또한 범용성 및 유연성을 제공하기 위하여 새로운 기능을 동적으로 추가할 수 있도록 한다. 마지막으로 제시한 모델의 활용 가능성을 평가하기 위하여 ATMega103 원 칩 마이크로프로세서를 기본으로 한 시험용 보드를 제작하고 제어 프로그램을 작성하여 MS 익스플로러 5.0 환경에서 실험하였다.

## A Lower-cost $\mu$ -Embedded Web Server for Controlling the Equipments

Min-Jung Oh<sup>†</sup> · Seong-Rak Rim<sup>††</sup>

### ABSTRACT

Most of the traditional embedded web server systems have been designed for monitoring and controlling some dedicated equipments. Hence, not only they have no generality and flexibility but also they are too expensive for the lower-cost domestic equipment. To cope with these difficulty, we suggest a lower-cost  $\mu$ -embedded web server model which is suitable for monitoring and controlling the industry or house equipments by using the internet. The suggested model is based on an one-chip  $\mu$ -processor in which the ISP (In-System Programming) function and flash ROM are embedded basically to minimize the cost of H/W and S/W. Also it allows to add an new function dynamically to provide the generality and flexibility. Finally, to evaluate the feasibility of the suggested model, we have manufactured a test-board based on the ATMega103  $\mu$ -processor and programmed the control program and tested it on the MS Explorer 5.0 environment.

**키워드 :** 임베디드 웹 서버(Embedded Web Server)

### 1. 서 론

정보 통신 기술의 발전과 인터넷 서비스의 다양화로 인하여 사회 전반에 걸쳐 많은 변화가 이루어지고 있다. 그 일환으로 고유의 기능을 독립적으로 수행하던 기존의 가정용 혹은 산업용 기기를 인터넷과 결합하여 그 기능을 확장하고 새로운 서비스를 창출하려는 방법이 요구된다 [1]. 예를 들어 일반 가정에서 사용되는 보안 기기, 냉난방 기기, 오디오/비디오 기기와 같은 다양한 종류의 가정용 기기를 인터넷과 연결하여 원격으로 감시 및 제어하고자 한다. 이러한 요구로 인하여 많은 종류의 기기가 인터넷에 연결되고 있다. 국내의 어떤 업체에서는 인터넷을 통하여 원격으로 감시 및 제어 할 수 있는 냉장고 혹은 세탁기를 상품화하고 있으며, 국외의 경우 인터넷을 통하여 자동 판매기의

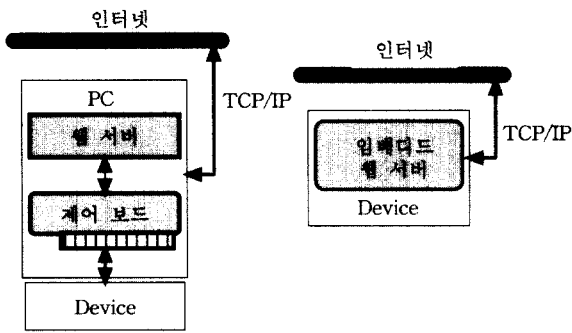
현재 상태 및 이상유무를 점검할 수 있다. 이처럼 가정용 혹은 산업용 기기를 인터넷에 연결하여 다양한 서비스를 제공하고자 하는 요구가 급증하고 있으나 이미 실생활에서 사용되어 오고 있는 대부분의 기기는 RS-232C나 적외선 리모컨 같이 매우 단순한 통신 프로토콜들을 내장하거나 통신 기능이 전혀 없다. 따라서 이러한 기기를 인터넷에 연결하기 위해서는 일종의 게이트웨이 기능을 할 수 있는 특별한 연결 장치가 요구된다 [2].

일반적으로 웹 서버를 이용하여 기기를 감시 및 제어 할 경우, (그림 1)과 같이 크게 두 가지 방법으로 접근할 수 있다.

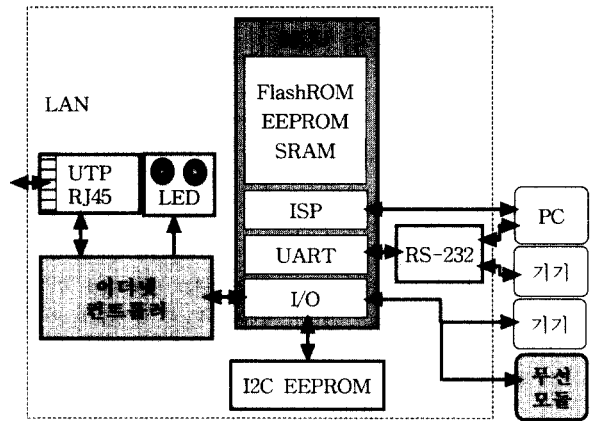
첫 번째 방법은 (그림 1.a)와 같이 웹서비스를 제공하기 위하여 별도의 PC를 이용하는 것이다. 이 방법은 별도의 PC는 물론 기기를 제어하기 위한 제어 보드를 내장해야하기 때문에 가격이 높아지고 전력 소모가 상당하기 때문에 유지비가 많이 소요된다. 또한 범용 PC의 운영체제는 신뢰도가 낮기 때문에 시스템 다운과 같은 여러 가지 문제가 발

<sup>†</sup> 정 회 원 : 호서대학교 벤처대학원 컴퓨터응용기술

<sup>††</sup> 종 신 회 원 : 호서대학교 컴퓨터학부 교수  
논문접수 : 2001년 10월 16일, 심사완료 : 2002년 2월 18일



(a) 별도의 PC를 이용 (b) 웹 서버를 기기에 내장  
(그림 1) 웹 서버를 이용한 기기의 제어 방법



(그림 2) 임베디드 웹 서버의 H/W 구성도

생할 수 있다. 두 번째 방법은 (그림 1.b)와 같이 웹서비스를 제공하기 위한 기능을 제어하려는 기기에 내장시키는 것이다. 이 방법은 웹 서버 기능의 구현과 시스템 운영을 위해 특별한 운영체제를 사용해야하고 원활한 기능 구현을 위한 강력한 마이크로프로세서와 고용량의 메모리를 필요로 하여 시스템의 가격을 높게 된다. 따라서 기존의 방법으로 비교적 가격이 저렴한 가정용 혹은 산업 기기를 인터넷으로 감시 혹은 제어하기에는 실용성이 떨어진다. 특히 제한된 네트워크 회선으로 다수의 기기를 제어 및 감시해야 하는 가정에서 사용하기에는 부적합하다.

일반적으로 임베디드 웹 서버는 범용 웹 서버와는 달리 비용과 크기에 대한 제약이 크다[3-5]. 따라서 범용 웹 서버에서 사용되고 있는 프로세서나 메모리 혹은 입출력 장치를 사용할 경우 많은 비용이 발생하게 됨으로 임베디드 시스템을 위한 마이크로프로세서의 선택과 임베디드 웹 서버의 기능들을 효과적으로 수행할 수 있도록 최적의 프로그램을 작성하는 것이 매우 중요하다.

본 논문에서는 비교적 가격이 저렴한 가정용 혹은 산업용 기기들을 인터넷을 통하여 원격으로 감시 및 제어할 수 있는 초소형 임베디드 웹 서버의 모델을 제시한다. 이를 위하여 저렴하고, 확장성을 제공할 수 있는 초소형 H/W 보드와 웹 서버 기능에 최적화된 S/W 모듈을 설계한다. 마지막으로 제시한 모델의 활용 가능성을 평가하기 위하여 기기의 감시 및 제어에 필요한 기본적인 기능들을 실험한다.

## 2. H/W 보드 설계

본 논문에서 저렴한 가격의 초소형 웹 서버를 구현하기 위하여 (그림 2)와 같이 플래쉬 롬이 내장된 ATMEL사의 ATmega103 [16] 원칩 마이크로프로세서를 사용하였다.

ATmega103은 8비트 RISC(Reduced Instruction Set Computer) 구조로 명령어가 간단하며 동작 속도가 빠르다. 제어를 위한 121개의 명령어를 내장하고 있고, 1MHz의 클럭당 약 1MIPS의 성능을 보이며 낮은 전압에서도 원활하게 동작한다. 프로그램의 수정 및 업그레이드가 용이하도록 마

이크로프로세서 내부에 프로그래밍 가능한 128Kbyte의 플래쉬 롬을 내장하고 있으며 고유한 프로그래밍 인터페이스를 통해 PC에 직접 연결하여 별도의 추가적인 장비 없이 프로그램을 마이크로프로세서 내부의 플래쉬 롬에 저장하고 실행할 수 있는 장점이 있다. 추가적으로 4Kbyte의 EEPROM을 내장하고 있어서 데이터 백업이 가능하고 다른 마이크로프로세서에 비해 비교적 큰 4Kbyte의 SRAM을 가지고 있어 이더넷 패킷 처리를 위해 사용하기에 충분하다. 또한 프로그래밍 가능한 UART, SPI(Serial Peripheral Interface), PWM(Pulse Width Modulation), 8비트 및 16비트 타이머, 10비트 ADC(Analog to Digital Converter)를 내장하고 있으며 48개의 많은 I/O 라인을 사용할 수 있어 다양한 기능을 추가 확장할 수 있다.

임베디드 웹 서버를 인터넷에 연결하기 위해 사용될 이더넷 컨트롤러는 시중에서 일반적으로 저가의 네트워크 카드에 많이 사용되어 성능과 안정성이 검증된 RealTek사의 RTL8019AS [17]를 선정하였다. 이더넷 컨트롤러의 I/O 라인과 ATmega103 I/O 라인을 직접 연결하여 제어 신호를 보내거나 요청되는 데이터를 송/수신하게 되는데 RTL8019AS는 PC의 ISA(Industry Standard Architecture) 버스 네트워크 카드에 주로 사용되는 컨트롤러로 PCI(Peripheral Component Interconnect) 관련 기능이나 PnP(Plug and Play) 기능을 필요로 하지 않는 임베디드 웹 서버 시스템에 사용하기에는 기능적인 측면이나 비용면에서 합리적인 선택이다.

임베디드 웹 서버에 내장된 커넥터는 직렬/병렬 포트, 무선 모듈 접속 포트, 디지털/아날로그 I/O 포트 등으로 각종 기기, 무선 모듈과 여러 가지 방법으로 연결되어 통신하게 되고, 10BaseT 펄스 트랜스에 연결된 RJ45 소켓에 UTP 랜 케이블을 연결하여 인터넷에 접속되는 구조를 가지고 있다.

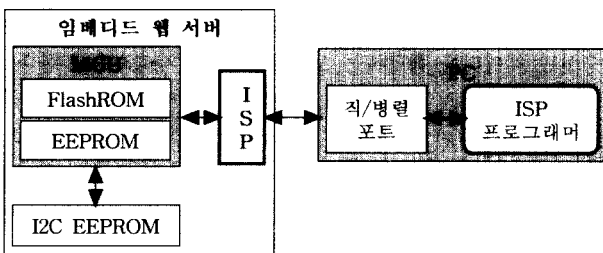
제시한 모델은 실생활에 쉽게 적용 가능한 임베디드 웹 서버로서 기존의 제품은 물론 새로 개발되는 제품에 대해

서도 쉽게 지원할 수 있도록 다양한 인터페이스를 고려하고, 하나의 임베디드 웹 서버로 다수의 기기를 유/무선으로 통합 제어 할 수 있도록 하였다. 이를 위하여 이더넷, 직/병렬, 다중 제어용 디지털 I/O, 무선 모듈 접속용 포트를 지원하고, PDA 혹은 휴대용 PC와 같은 정보 기기를 사용하여 웹 서버에 유/무선으로 접속하고, IrDA(InfraRed Data Association) 혹은 블루투스 [18]를 이용하여 무선으로 기기를 제어할 수 도 있도록 설계했다. 설계된 임베디드 웹 서버의 H/W는 핵심적인 역할을 하는 플래쉬 롬을 내장한 원칩 마이크로프로세서, 네트워크 기능을 위한 이더넷 컨트롤러, PC 또는 기기들과의 직렬 통신을 위한 RS-232C 레벨 컨트롤러, 10BaseT 신호 변환을 위한 펄스 트랜스, 상태 확인을 위한 LED, 추가적인 메모리 확장을 위한 직렬 I2C EEPROM, 각종 커넥터로 구성된다. 논문에서 제시한 임베디드 웹 서버의 상세 사양은 <표 1>과 같다.

<표 1> 임베디드 웹 서버 상세 사양

분류	내용
MPU	8Bit RISC 6MIPS
Memory	내부 128Kbyte 플래쉬 4Kbyte EEPROM 4Kbyte SRAM 외부 64Kbyte EEPROM
I/O port	UART, SPI, ISP, 무선 모듈 포트, UTP
Protocol	ARP, IP, TCP, UDP, ICMP, DHCP, HTTP
Max Speed	LAN 10Mbps, UART 115Kbps
Etc	PWM, 8/16 Timer, 8ch 10bit ADC, Analog Comparator, Power-down Mode
Voltages	4~5.5V
Size	가로 45mm, 세로 70mm
Price	약 20,000~25,000

임베디드 웹 서버에 내장할 프로그램 개발시에는 (그림 3)과 같이 임베디드 웹 서버에 내장된 ISP 포트를 통해 PC의 직렬/병렬 포트에 직접 또는 ISP 케이블을 이용해 연결하고, PC에서 프로그래머를 이용해 내장 플래쉬 롬과 EEPROM에 컴파일된 파일이나 데이터를 전송한다. 프로그램 개발에 필요한 별도의 장비는 없으며 임베디드 웹 서버와 PC만 있으면 프로그램 개발과 테스트가 가능하도록 설계하였다.

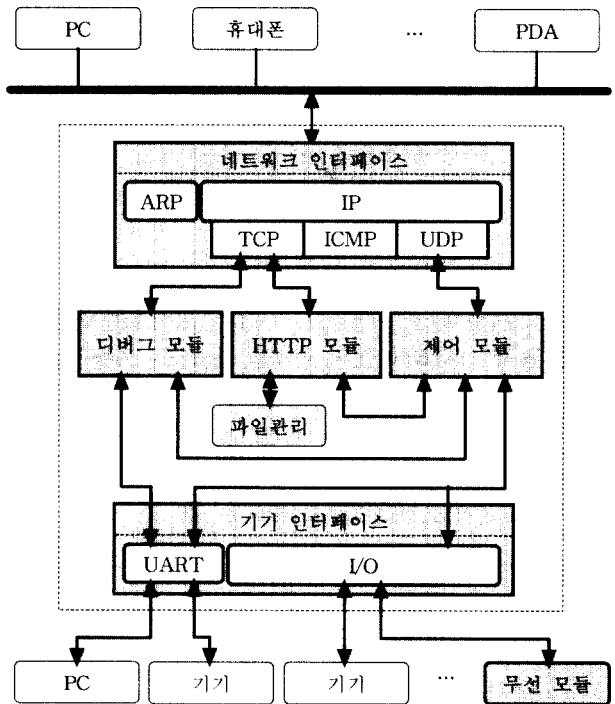


(그림 3) ISP를 이용한 프로그래밍

본 논문에서 제시한 임베디드 웹 서버는 H/W 설계의 최적화로 부품의 수를 줄이고 가격과 크기 문제를 해결하였고, 플래쉬 롬과 ISP 기능을 이용함으로써 확장성과 프로그램 개발의 용이성 등의 임베디드 웹 서버가 갖춰야할 기능과 요구 사항에 부합한다. 또한 임베디드 웹 서버의 H/W적인 범용성 및 유연성을 충족시키기 위해 무선 모듈 접속용 포트를 내장하여 IrDA, 블루투스 모듈과 같은 무선 모듈을 접속 가능하게 하였다. 이로써 다양한 기기를 유/무선으로 제어하기 위한 기능을 원활하게 수행할 수 있어 여러 분야에 응용할 수 있다.

### 3. S/W 모듈 설계

S/W 모듈은 웹 서버의 기능에 충실하면서 기기의 감시 및 제어를 안정적으로 수행할 수 있도록 불필요한 기능은 배제하고, 각각의 모듈을 최적화하여 최소의 코드 사이즈로 필요로 하는 기능을 수행 가능하도록 설계한다. 본 논문에서 제시한 웹 서버 기능에 최적화된 S/W 모듈의 기본적인 개념도는 (그림 4)와 같다.



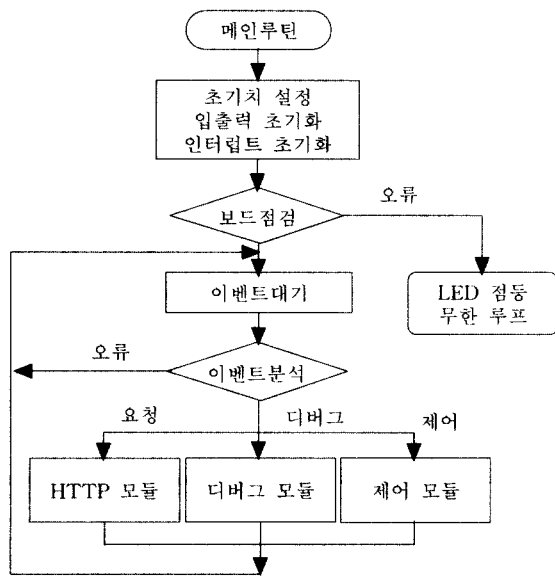
(그림 4) S/W 모듈의 개념도

(그림 4)에서 네트워크 인터페이스 모듈은 PC, 휴대폰 혹은 PDA와 같은 인터넷 매체로부터의 웹 서비스 요청 메시지를 받아들이고 이들 요청에 대한 응답을 전달하는데 필요한 네트워크 프로토콜들을 지원하도록 한다. 반면 기기 인터페이스 모듈은 인터넷을 통하여 기기의 동작 상태를 모니터링하고 이를 제어하기 위한 데이터 입출력 프로토콜

을 해결하는 기능을 담당한다. 이외에도 웹서비스를 지원하기 위한 HTTP 모듈, 웹 페이지 관리를 위한 파일 관리 모듈, 각종 기기를 제어하기 위한 제어 모듈들이 필요하다. 마지막으로 임베디드 웹 서버의 동작 상태를 모니터링하고 제어하기 위한 디버그 모듈을 포함한다.

3.1 제어의 흐름

임베디드 웹 서버 S/W 모듈의 기본적인 제어 흐름도는 (그림 5)와 같다.



(그림 5) 제어의 흐름도

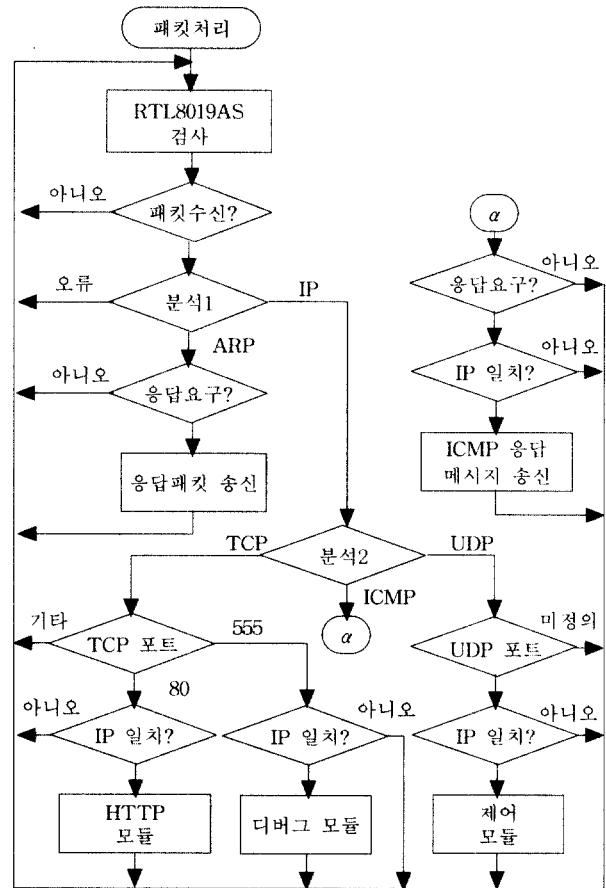
시스템이 시작되면 마이크로프로세서를 초기화하고 보드 점검을 시작한다. 보드 점검에서 오류가 발생하면 내장된 LED를 계속 점등하여 오류 상태를 표시하며 무한 루프로 들어간다. 아무 이상이 없으면 네트워크, UART, I/O등에서 발생하는 이벤트를 기다리고, 이벤트가 발생하면 그 이벤트의 종류를 분석하여 해당 프로그램에 전달한다. 각 이벤트에 대한 기본적인 처리 방법은 다음과 같다.

- ① **HTTP 요청일 경우** : HTTP 모듈에서 클라이언트 웹브라우저로부터의 요청에 해당하는 명령을 수행하거나 파일 관리 모듈을 통해 필요로 하는 웹 페이지 데이터를 임베디드 웹 서버의 기억공간에서 찾아서 네트워크 인터페이스 모듈을 통해 전송한다.
- ② **디버그 명령일 경우** : 디버그 모듈에서 해당하는 명령을 처리하여 결과를 기기 인터페이스 모듈의 UART 또는 네트워크 인터페이스 모듈을 통해 PC의 직렬 통신 프로그램이나 연결된 클라이언트의 웹브라우저에 전송한다.
- ③ **기기 제어 명령일 경우** : 기기 제어 명령에 해당하는 제어 모듈을 호출한다. 호출된 제어 모듈은 기기 인터

페이스 모듈의 UART, I/O 등을 제어하여 정의된 기능을 수행한다.

이벤트 분석 단계에서 정의되지 않은 이벤트나 잘못된 이벤트가 발생할 경우는 무시하고 무조건 이벤트 대기 단계로 돌아가게 된다. 각 이벤트의 처리 완료 후에도 이벤트 대기 단계로 돌아간다.

3.2 패킷 처리 알고리즘



(그림 6) 패킷 처리 알고리즘

본 논문에서 제시된 임베디드 웹 서버를 인터넷에 연결하기 위한 네트워크 인터페이스 모듈의 패킷 처리 알고리즘은 (그림 6)과 같다. ARP, ICMP 프로토콜은 단순 응답 요구만 처리하므로 각 프로토콜 내에서 모든 처리를 끝낸다. 다른 모듈과 연계되는 프로토콜은 TCP, UDP이며 각 프로토콜은 수신된 데이터를 해당하는 모듈에 보내서 처리하도록 한다.

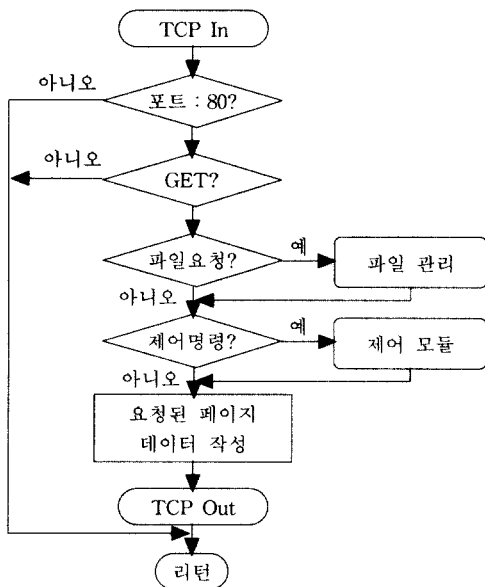
RTL8019AS의 RSR 레지스터를 검사하여 수신된 패킷이 있으면 정상적인 패킷인지 검사하여 불량 패킷은 버린다. 정상적으로 수신된 패킷은 타입 필드를 분석(분석 1)하여 ARP, IP를 구분하여 처리하고, 이외의 패킷은 불필요함으로 무시하고 패킷 수신 대기 상태로 돌아간다.

각 프로토콜의 기본적인 패킷 처리 방법은 다음과 같다.

- ① **ARP** : ARP 응답 요구이면서 IP가 일치하면 ARP 응답 패킷을 송신한다.
- ② **IP** : (분석1)에서 IP일 경우 IP 헤더를 분석(분석2)하여 상위층에 해당하는 프로토콜(TCP, UDP, ICMP)을 검사하고 각각의 프로토콜 처리 루틴으로 분기한다.
- ③ **TCP** : TCP 헤더의 포트 번호를 검사하여 웹서비스를 위한 HTTP 포트 80번과 온라인 디버그 모니터링을 위한 디버그 포트 555번에 해당하는 기능을 처리하고, 기타 다른 포트는 무시한다. 포트 80번의 경우는 IP가 일치하는지 검사하고, HTTP GET 요청에 의한 URL 쿼리를 분석하여 해당하는 데이터를 전송한다. 포트 555번의 경우는 요청된 디버그 명령을 처리하고, 결과를 HTML형식으로 작성하여 클라이언트의 웹브라우저에 전송한다.
- ④ **UDP** : UDP 헤더의 수신지 포트 번호가 서비스 가능한(정의된 포트) 것인지 검사하고, 서비스 가능한 포트 번호일 경우에는 IP가 일치하는지 검사한 후 해당 서비스 기능을 수행한다. 서비스 불가능한(미정의된 포트) 포트 번호이거나 IP가 일치하지 않는 경우에는 패킷을 무시하고 수신 대기 상태로 돌아간다.
- ⑤ **ICMP** : ICMP 헤더를 검사하여 ICMP echo, PING 응답 요구인지 확인하고, IP가 일치할 경우 에코 응답 헤더와 수신한 데이터를 송신한다.

3.3 HTTP 모듈

HTTP 모듈은 사용자의 웹 브라우저로부터 요청된 웹 페이지 데이터를 전송하는 기능 담당하는 모듈로써 개략적인 제어의 흐름도는 (그림 7)과 같다. 임베디드 웹 서버는 특성상 클라이언트로부터 전송되는 정보의 양이 비교적 작



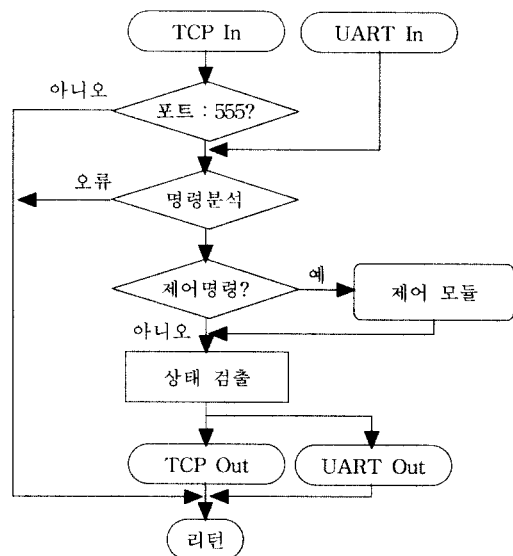
(그림 7) HTTP 모듈의 흐름도

아서 HTTP GET method를 사용하는 것이 메모리 관리와 데이터 처리에 효율적이다. 따라서 본 논문에서는 HTTP GET method만을 지원하도록 설계한다.

HTTP 모듈은 클라이언트 웹브라우저의 HTTP GET 요청에 대해 HTML 문서, 텍스트, 이미지, 자바 애플릿 등을 전송함으로써 실제 웹 서버와 같은 응답을 한다. 클라이언트의 직접적인 HTTP GET 요청에 대한 웹 페이지나 이미지, 자바 애플릿 등의 반환은 임베디드 웹 서버의 기억공간에 저장된 데이터를 이용하여 실제 파일시스템처럼 처리한다. 이러한 동작을 위해 파일관리 모듈을 이용한다. HTTP 모듈에서 URL 쿼리를 분석하여 필요로 하는 데이터를 요청하면 파일관리 모듈은 임베디드 웹 서버의 기억공간에 저장된 데이터를 검색하여 반환하게 된다. 웹 페이지 데이터에 기기 제어 명령이 포함되어 있을 경우에는 해당하는 기능을 처리하는 제어 모듈의 제어 루틴을 수행하고, 필요에 따라 수행 결과를 이용해 웹 페이지를 생성하여 클라이언트의 웹브라우저에 전송한다.

3.4 디버그 모듈

디버그 모듈은 임베디드 웹 서버의 마이크로프로세서, 기억공간, 각종 I/O 포트의 상태 등을 모니터링하기 위한 모듈로써 개략적인 제어의 흐름도는 (그림 8)과 같다. 네트워크 프로토콜의 지원으로 인터넷을 통해서 일반 웹브라우저로 모니터링 가능하게 하고, UART를 통해서 PC의 직렬 통신 포트와 직접 연결하여 직렬 통신 프로그램으로도 모니터링 할 수 있도록 설계한다.



(그림 8) 디버그 모듈의 흐름도

클라이언트의 웹브라우저를 이용한 모니터링의 경우 TCP 포트 555번을 이용하여 접속한다. 디버그 명령은 웹브라우저의 주소 창에 직접 입력하며 형식은 다음과 같다.

HTTP://IP 주소:555/명령 인수1 인수2 인수3 ...

직렬 통신 프로그램을 이용한 모니터링의 경우 임베디드 웹 서버의 초기 설정에 정의된 UART 통신 환경으로 PC와 연결되며 명령 입력 형식은 다음과 같다.

명령 인수1 인수2 인수3 ...

디버그 모듈은 크기를 줄이기 위해 텍스트 기반의 단순한 인터페이스를 가지며 명령의 형태는 <표 2>와 같이 5가지로 분류된다.

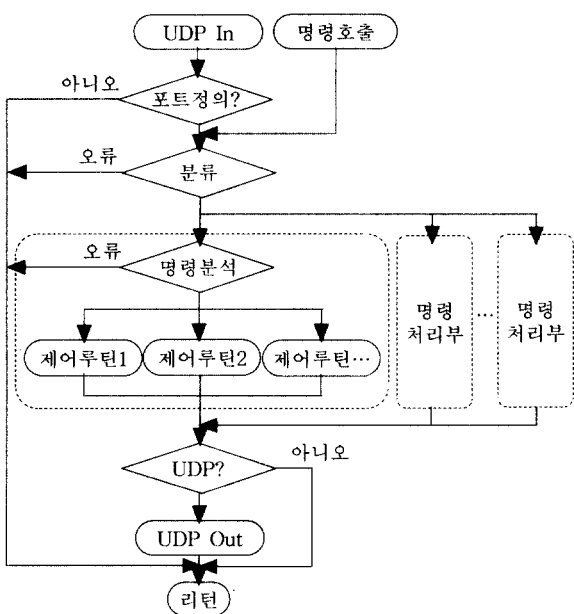
<표 2> 디버그 명령의 분류

분류	설명
I/O 읽기 요구	지정된 주소의 I/O 포트 읽기
I/O 쓰기 요구	지정된 주소의 I/O 포트에 지정 값 출력
메모리 읽기 요구	지정된 주소와 길이로 메모리 영역 읽기
메모리 쓰기 요구	지정된 주소의 메모리에 지정 값 출력
기능 호출 요구	지정된 기능의 루틴 수행

디버그 모듈에서 사용 가능하게 정의된 제어 명령의 경우, 해당하는 기능을 처리하는 제어 모듈의 제어 루틴을 수행하고, 필요에 따라 수행 결과를 UART를 통해 PC로 전송하거나 HTML 형식으로 변환하여 TCP/IP를 통해 접속된 클라이언트의 웹브라우저에 전송한다.

### 3.5 제어 모듈

제어 모듈은 다른 모듈에서 요청하는 제어 명령을 수행하여 기기를 제어하기 위한 모듈로서 개략적인 제어의 흐름도는 (그림 9)와 같다.



(그림 9) 제어 모듈의 기본 흐름도

제어 모듈 각각의 기능은 네트워크 인터페이스 모듈의 UDP 부분과 HTTP 모듈, 디버그 모듈에서 요청하는 기능 호출을 통해서 수행된다. 임베디드 웹 서버의 범용성 및 유연성을 제공하기 위하여 제어 모듈은 하나의 UDP 포트에 하나의 기기를 제어하기 위한 명령 처리부를 포함하며 각각의 명령 처리부는 UDP 포트 번호에 의해 분류된다. 각각의 UDP 포트 번호에 따라 기기를 할당함으로써 기기 증가에 따른 다양한 기능의 추가 및 수정이 용이하게 된다. HTTP 모듈, 디버그 모듈의 명령 호출의 경우 명령 호출 인수에 각각의 명령 처리부에 할당된 UDP 포트 번호를 포함하며 형식은 다음과 같다.

명령 분류번호(UDP포트번호) 인수1 인수2 인수3 ...

클라이언트 웹 브라우저에서 각각의 기기를 실시간으로 감시 및 제어하기 위해서 자바 애플릿을 이용하여 UDP로 통신한다. UDP 사용시 발생할 수 있는 패킷 손실이나 중복과 같은 에러에 대처하고 신뢰성을 확보하기 위해 명령을 전송한 다음 응답 메시지가 돌아오는 것을 확인한다. 응답 메시지가 수신되지 않는 경우는 패킷이 손실된 것이라 가정하고 명령을 재 전송하여 패킷의 손실에 대비한다.

## 4. 구현 및 실험

본 논문에서 제시한 임베디드 웹 서버 모델의 활용 가능성을 평가하기 위하여 다음과 같은 환경에서 S/W 모듈을 구현하여 기본적인 기능들을 실험하였다.

### 4.1 구현 환경

임베디드 웹 서버의 S/W 모듈은 ATMEL사에서 제공하는 AVR 어셈블러와 GNU 공개 컴파일러인 AVR-GCC를 이용하여 작성하고 컴파일하였다. 디버깅을 위한 직렬 통신 프로그램과 임베디드 웹 서버의 플래쉬 롬에 S/W 모듈을 적재하기 위해 사용되는 ISP 프로그램은 마이크로소프트사의 비주얼 스튜디오 6.0으로 작성하였다.

설계된 임베디드 웹 서버의 실험을 위한 환경으로는 제작된 H/W와 임베디드 웹 서버에 내장될 프로그램, PC에서 사용될 프로그램과 웹 페이지를 구성하기 위한 HTML 파일과 이미지 파일, 자바 애플릿 등이 필요하며 제시된 임베디드 웹 서버의 실험을 위해 필요한 기본적인 절차는 다음과 같다.

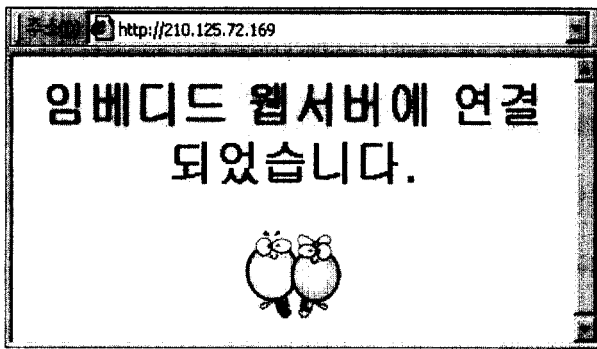
#### 실험을 위한 기본 절차

- ① 임베디드 웹 서버에 전원 인가
- ② PC와 ISP 포트를 통해 연결
- ③ 임베디드 웹 서버로 프로그램 전송
- ④ 임베디드 웹 서버의 IP등 환경 설정
- ⑤ 작성된 웹 페이지 파일 EEPROM에 저장.
- ⑥ 10BaseT UTP 랜 케이블 연결.
- ⑦ 인터넷을 통해 웹브라우저로 접속.

4.2 실험

본 논문에서 설계 구현된 임베디드 웹 서버의 활용 가능성을 검증하기 위해 기본적인 세 가지 실험을 한다. 첫 번째로 임베디드 웹 서버의 정상 동작을 확인하기 위한 기본 웹 페이지 출력 실험, 두 번째로 유선으로 연결된 다수의 기기를 제어하는 기능을 검증하기 위한 디스플레이 장치, 서보 모터 보드, 온도 센서 보드의 제어/감시 실험, 세 번째로 폭넓은 확장성과 범용성을 위한 무선 모듈 접속 기능을 검증하기 위해 임베디드 웹 서버에 적외선 무선 모듈을 연결하여 TV와 오디오를 인터넷폰으로 제어하는 실험을 하겠다.

[실험 1]

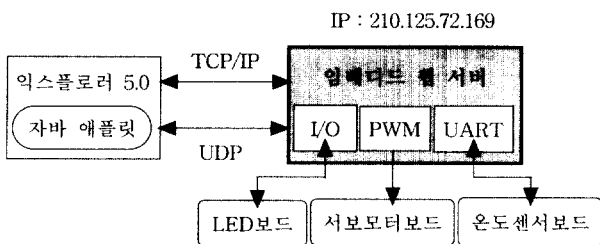


(그림 10) 실험 결과 화면 1

[실험 1]에서는 MS 익스플로러 5.0으로 임베디드 웹 서버에 설정한 IP로 접속시 I2C EEPROM에 저장된 기본 웹 페이지(index.htm과 test.jpg)가 정상적으로 전송되어 (그림 10)과 같이 웹브라우저에 출력되었다. 이것으로 임베디드 웹 서버가 정상적으로 동작하는 것을 확인하였으며 디버그 모니터 기능으로도 정상적인 동작 상태를 확인할 수 있었다.

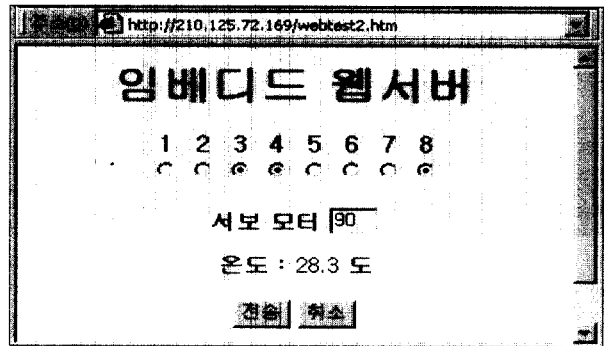
[실험 2]

[실험 2]에서는 (그림 11)의 실험 구성도와 같이 임베디드 웹 서버에 LED, 서보 모터 보드, 온도 센서 보드를 연결하고 클라이언트가 요청하는 작업을 내장된 제어용 프로그램으로 I/O 포트를 제어하여 처리한다. 제어용 프로그램은 기기의 제어 및 감시에 필요한 기본적인 기능을 내장하고 있으며 간단한 명령의 조합으로 제어에 필요한 여러 가지 기능을 구현할 수 있다.



(그림 11) 다수의 기기 유선 제어 실험 구성도

실험 2의 결과는 (그림 12)와 같으며 제어하고자하는 LED와 서보 모터의 각도를 설정하고 전송을 누르면 임베디드 웹 서버의 제어용 프로그램이 I/O 포트를 제어하여 해당하는 포트에 연결된 LED를 점등하고, PWM 신호로 서보 모터의 각도를 제어하게 된다.

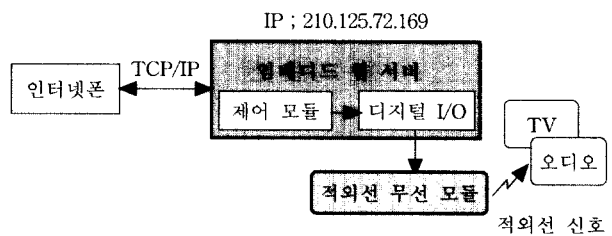


(그림 12) 실험 결과 화면 2

웹 서버에 연결된 온도 센서 보드로부터의 정보를 실시간으로 보여주기 위해 자바 애플릿을 사용하였으며 일정 시간마다 온도의 변화를 화면에 출력하게 했다. 온도 센서 보드와의 통신은 UART를 통한 직렬 통신으로 이루어졌고, 자바 애플릿과 임베디드 웹 서버간의 통신은 UDP 포트 1004번을 이용했다. 실험에서 연결된 각각의 기기들은 원활하게 동작하였으며 온도 센서의 정보도 정확히 출력되었다.

[실험 3]

[실험 3]은 (그림 13)과 같이 임베디드 웹 서버에 적외선 무선 모듈을 연결하고 인터넷폰으로 접속하여 TV와 오디오를 제어하는 실험이다.

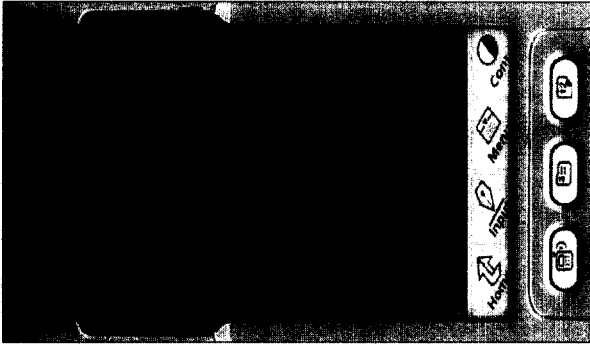


(그림 13) 무선 모듈 접속 기능 실험 구성도

기기 제어에 필요한 적외선 신호는 임베디드 웹 서버를 PC의 직렬 포트에 연결하고 통신 프로그램으로 접속한 후에 적외선 무선 모듈을 이용하여 각 기기의 리모컨 버튼을 누를 때 나오는 적외선 신호 데이터를 검출한 후 재사용하였다.

(그림 14)는 인터넷폰으로 임베디드 웹 서버에 접속한 화면으로써 터치 스크린에 나타난 버튼을 누르면 임베디드 웹 서버에 해당하는 기능을 요청하게 되고, 내장된 제어 프로그램에서 I/O 포트를 제어하여 적외선 무선 모듈에 제어

신호를 보낸다. 적외선 무선 모듈은 해당하는 기능의 리모컨 신호를 송출하게 되어 기기의 제어가 이루어진다.



(그림 14) 실험 결과 화면 3

## 5. 결 론

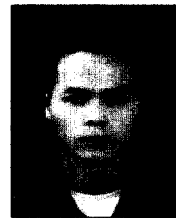
본 논문에서는 다양한 기기의 감시 및 제어를 위한 저가의 초소형 임베디드 웹 서버의 모델을 제시하고 ATMega103원 칩 마이크로프로세서를 기본으로 한 시험용 보드를 제작하여 활용 가능성을 검토하였다.

제시된 임베디드 웹 서버는 IrDA, 블루투스 와 같은 무선 모듈을 접속하면 여러 기기들과 양방향 무선 통신이 가능하여 향후 출시될 가전, 산업용 기기의 감시, 제어를 위한 게이트웨이 기능을 수행할 수 있으며 인터넷 연결 기능이 없는 각종 정보기기의 인터넷 연결 기기로도 사용할 수 있다. 또한 인터넷을 이용하여 유/무선으로 각종 기기를 통합 제어할 수 있는 장점과 시스템 설계의 최적화로 기존 제품보다 저렴한 비용으로 제작할 수 있어 고가의 임베디드 제품들을 대체할 수 있다.

## 참 고 문 헌

[1] M. Bathelt, "Using a Hypertext System as a Generic User Interface for Embedded Systems," Embedded Web Technologies-WWW Conference, Santa Clara, CA, 1997.  
 [2] Steve Wingard, "Embedded Web Server Technology," Fifth International Conference on the WWW, San Francisco, America, 1997.  
 [3] I. D. Agranat, "Engineering Web Technologies for Embedded Applications," IEEE Internet Computing, Vol.2-3, 1998.  
 [4] N. Witchey, "An Easy to Do Embedded Web Server," IEEE Internet Computing, Vol.2-3, 1998.

[5] 최성중, "임베디드 시스템으로의 PPP 소프트웨어 이식", 정보처리학회논문지 Vol.7-7, 2000.  
 [6] Bentham, Jeremy, "TCP/IP Lean Web Servers for Embedded Systems," CMP Books, 2000.  
 [7] Thomas, Stephen, "HTTP Essentials : Protocols for Secure, Scaleable Web Sites," Wiley, 2001.  
 [8] Jean J. Labrosse, "MicroC/OS-II : The Real-Time Kernel," R&D Technical Books, 1998.  
 [9] Jack G. Ganssle, "The Art of Designing Embedded Systems," Newnes, 1999.  
 [10] Michael Barr, "Programming Embedded Systems in C and C++," O'Reilly, 1999.  
 [11] Lipovski, G. J., "Embedded Microcontroller Interfacing H/C," Academic Press, 2000.  
 [12] Grehan, Rick/Moote, Robert/Cylix, Ingo, "Real-Time Programming," Addison-Wesley Pub Co (Sd), 1998.  
 [13] Uyless Black, "TCP/IP and Related Protocols," McGraw-Hill, 1997.  
 [14] Gene H. Miller, "Microcomputer Engineering, 2/E," Prentice Hall, 1999.  
 [15] Karanjit S. Siyan, 번역 이도희, "TCP/IP 완전 정복", 성안당, 1998.  
 [16] AVR 8-Bit RISC Data Sheets, "http://www.atmel.com/atmel/products/prod200.htm".  
 [17] RTL8019(AS) DataSheets, "http://www.realtek.com.tw/html/download/cgi/DLd1.cgi/h?class1=communication".  
 [18] Bluetooth Special Interest Group, "Bluetooth System Ver 1.1," 2000.



### 오 민 정

e-mail : dsdman@orgio.net

2000년 호서대학교 컴퓨터공학(학사)

2002년 호서대학교 벤처대학원 컴퓨터응

용기술(석사)

관심분야 : 임베디드 시스템, 자동제어, 보안 시스템



### 임 성 략

e-mail : srrim@office.hoseo.ac.kr

1979년 서강대학교 전자공학(학사)

1983년 서울대학교 컴퓨터공학(석사)

1992년 서울대학교 컴퓨터공학(박사)

1983년~1990년 금성 반도체(주) 연구소

1993년~현재 호서대학교 컴퓨터학부

부교수

관심분야 : 운영체제, 실시간처리 시스템, 분산처리 시스템