

# Web browser를 이용한 가정용 가스보일러 제어기술 개발

(Development of the Small Gas Boiler Controller Using Web Browser)

손수국\*

(Su-Goog Shon)

## 요 약

본 논문에서는 기존 보일러 제어기와 병행하여 사용될 수 있는 인터넷 웹 보일러제어 기술의 개발에 논한다. 웹 보일러제어기는 중요 요소로 RTL8019 네트워크제어기와 TS80C32 마이크로컨트롤러로 구성된다. 인터넷을 통한 통신을 위하여 IP, TCP, UDP, ICMP, 및 HTTP의 프로토콜과 이더넷 네트워크드라이버를 개발할 필요가 있다. 웹 보일러제어기 경우, RAM 사용량을 최소화하기 위한 공통전역버퍼 알고리즘을 제안한다. 끝으로 개발된 프로토콜 들의 올바름과 성능을 측정하기 위하여 CommView와 Dummysnet이 사용된다. 개발결과 수백 바이트 RAM에서도 호스트 간에 상호 호환성을 잃지 않고 임베디드 웹 제어기가 동작된다.

## Abstract

This paper describes the development of a web-based boiler controller which can be in parallel operated with an existing boiler controller. The web-based boiler controller mainly consists of RTL8019AS NIC and TS80C32 microcontroller. In order to communicate over the Internet, we need to develop network driver, IP, TCP, UDP, ICMP, and HTTP. For a specific application like web-boiler controller, we have proposed a common global data buffer algorithm to minimize the RAM memory usage. Finally, the correctness and performance of the protocols are tested and verified using CommView and Dummysnet. The development is satisfactorily operated only for few hundreds of bytes of RAM usage without sacrificing interoperability between hosts.

Key Words : Web controller, boiler, embedded system, TCP/IP

## 1. 서 론

국내 가정의 난방 구조가 20여 년 전부터 기름보일러에서 가스보일러로, 중앙 집중난방으로부터 개

별 난방으로 바뀌면서 가정용 가스보일러의 수요가 많아지고 소비자 선택폭이 넓어짐에 따라 가스보일러의 부가기능의 중요성이 증대되었고 날이 갈수록 제품간의 경쟁이 치열하면서 보일러 제어기를 중심으로 한 다양한 기술개발이 어느 때보다 절실하다.

특히 일반 가정이 핵가족화 되고 주부들의 사회진출이 늘어나면서 외부에서 원격으로 집을 감시하고 보일러 등을 원격 제어하는 필요성이 많아졌으며,

\* 주저자 : 수원대학교 정보통신공학과 교수  
Tel : 031-220-2532, Fax : 031-229-8281  
E-mail : sshon@suwon.ac.kr  
접수일자 : 2004년 9월 10일  
1차심사 : 2004년 9월 13일, 2차심사 : 2004년 10월 7일  
심사완료 : 2004년 10월 19일

한국통신에서도 IMT2000의 추진계획에 가전제품의 원격제어와 감시를 포함하고 있다. 또한 농촌까지도 가스배관이 이루어져 농사용에도 사용되는 만큼 온실 등의 무인화 난방 시설에 대한 원격지에서 수시로 가스보일러의 동작상태의 감시기능이 필요하다.

본 논문에서는 웹 브라우저로 인터넷에서 보일러의 운전/정지와 목표 실내 온도의 설정 등 보일러의 운전에 필요한 원격 제어 및 설정이 가능하도록 하며 보일러의 운전상태, 보일러 수의 온도, 실내온도 등 실시간 운전정보의 수집이 가능하도록 하는 보일러 제어기를 설계한다.

설계의 주요 고려사항 중 하나는 기존 보일러제어기가 주로 8비트 마이크로컨트롤러를 사용하고 있는데, 기존의 보일러제어기를 인터넷 보일러제어기와 결합하여 사용할 수 있도록 호환성을 달성하는 것이다. 이는 제품의 가격경쟁력 제고와 보일러제어의 편리함에 대한 중요한 고려사항이다.

인터넷 기능을 제공하는 프로토콜 TCP/IP를 소형 마이크로컨트롤러 시스템에 구현하려면 하드웨어의 한계와 개발 툴의 특성을 반영하여야 할 것이다. 어떻게 마이크로컨트롤러 시스템에 TCP/IP를 제한된 메모리에 적재하는 것이 중요하다. 물론 이렇게 구현된 프로토콜은 인터넷의 다른 호스트와 상호통신에 문제가 없어야 한다. 호환성의 지원을 위한 국제표준이 있는데, RFC 1122[1]는 호스트에서의 연결계층, 네트워크계층, 및 전달계층에 대한 기준을, 그리고 RFC1123는 응용계층에서의 요구조건을 규정하고 있다.

임베디드 시스템을 위한 인터넷 프로토콜 구현의 역사를 조사하면 버클리 BSD TCP/IP로부터 인터넷 프로토콜이 파생되어 많이 발전하여 왔다. 완벽한 TCP/IP 구현을 위하여 보통 수백 [Kbytes]의 프로그램메모리 ROM과 수백 [Kbytes]의 데이터메모리 RAM이 필요하다. 하지만 수십 [Kbytes]의 ROM과 수백 바이트 RAM을 갖는 마이크로컨트롤러에 TCP/IP의 전체기능을 구현하는 것은 불가능하다.

임베디드 시스템을 위하여 소형으로 TCP/IP를 구현한 몇 예가 있다. 8비트에서 32 비트 RISC 아키텍처까지 다양한 목표시스템이 있다. 8비트 마이크로컨트롤러를 기반으로 프로토콜을 구현한 예로서

uIP[2,3]이 있는데, Atmega128 AVR에서 5164 바이트의 프로그램 ROM 메모리를 사용하여 구현하였다. IPic[4]은 성능 골 크기의 칩에 서버를 구현하였으며, Jeremy Bentham의 PICmicro 스택[5]과 Atmel TCP/IP 스택[6]도 구현 예들이다. 이들의 일부는 간략화된 소형의 TCP/IP를 또 일부는 전체 기능의 TCP/IP를 구현하였다. 32비트 인텔 x386 CPU를 사용하여 인터넷 기반 RF 태그리더를 개발한 경험이 있는데, 본 논문에서는 8비트 마이크로컨트롤러를 사용하여 운영체제 없이 개발한다[7].

## 2. 인터넷 보일러제어기의 제안

가정용 가스보일러시스템은 그림1과 같이 보일러, 밸브, 냉/온수 배관, 센서, 보일러제어기, 및 전원 등으로 구성되어 있다. 이러한 다양한 시스템을 중앙에서 감시 및 제어의 운전을 하는 부분을 제어기라 한다. 사용자는 각종 온도 또는 가스누출 센서에서 측정되는 입력신호를 받아 보일러제어기를 통하여 냉온수의 밸브를 제어하는데 보일러의 안전사고에 크게 영향을 준다. 보일러제어기의 중요성 때문에 인터넷기능을 부여할 때는 그 설계 및 구현에 매우 주의 하여야 한다.

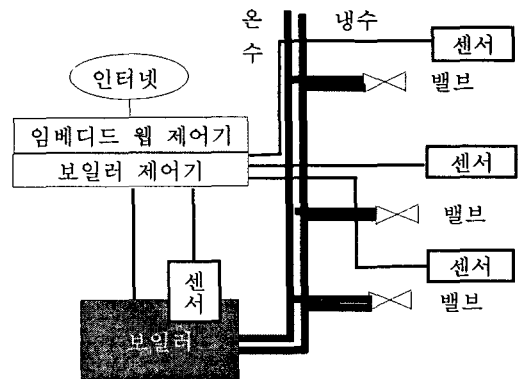


그림 1. 제안된 임베디드 웹 보일러제어 시스템  
Fig. 1. Proposed embedded web boiler system

기존 보일러제어기는 단일 동작시스템으로서 외부 접속은 RS-232를 통하여 가능하다. 기존의 보일러제어기를 단독으로 사용하는 것도 가능하지만, 여

기에 병행하여 임베디드 웹 제어기를 부가하여 인터넷 보일러제어도 가능한 개념으로 시스템을 제안한다.

그리고 임베디드 웹 제어기의 시스템 구성은 다음과 같다. 입출력신호를 중앙 처리하는 중앙처리부, 프로그램 및 데이터의 보관을 위한 저장부, 인터넷과 데이터통신을 수행하는 통신부, 시스템 전체에 필요한 에너지를 공급하는 전원부, 각종의 설정 값, 운전상태에 대한 표시부로 구성된다. 임베디드 웹 제어기에서 통신부는 인터넷 접속 및 웹 서버기능과 기존 보일러와의 RS-232 통신 기능을 갖는다. 그리고 웹 서버의 동작을 위하여 먼저 네트워크제어기(Network Interface Controller : NIC)의 제어를 위한 네트워크, TCP, UDP, ARP, IP 등의 프로토콜 개발이 필요하다. 또한 인터넷상의 웹 화면에서의 제어명령은 임베디드 웹 서버와 RS-232 통신을 통하여 보일러제어기에 양방향으로 전달되고, 보일러시스템의 제어 및 설정이 필요하다. 임베디드 웹 제어기 단독으로 제어 및 감시 기능이 가능하고, 제어대상과 감시대상의 설정(즉, 온도, 난방, 운전, 연소, 전원 스위치의 개폐 상태 감시 및 제어)이 필요하다. 또한 예러 검출 및 수정 기능이 필요하다.

본 연구에서는 컴퓨터를 기반으로 한 웹 기능의 지원 보다는 가격경쟁력이 있는 임베디드 웹 서버 설계방법을 채택한다. 임베디드 웹 서버 기술은 32비트 ARM을 기반에 리눅스 운영체제를 포팅하여 인터넷기능을 구현하는 방법이 일반적이나, 본 논문에서 적용할 때는 전력소모 및 가격경쟁력에 문제가 있다. 따라서 본 논문에서는 운영체제를 사용하지 않으면서 8비트 마이크로컨트롤러에 간략화된 TCP/IP 스택을 구현한다. 인터넷 네트워크제어기는 RealTek의 RTL8019AS를 사용한다.

### 3. 임베디드 TCP/IP 프로토콜 구현

#### 3.1 개발환경

마이크로컨트롤러와 관련된 소프트웨어 개발 도구는 하드웨어의 단순성으로부터 최대 능력을 발휘하도록 하는 것을 선택하는 것이 중요하다. 표준 소프트웨어 도구는 아니지만 마이크로컨트롤러에 적

합하고, 광범위하며 안정적으로 사용되는 것을 선택하였다. 본 연구에서 사용한 임베디드 웹 제어기의 개발환경 및 하드웨어 사양은 다음과 같다. TS80C32 8비트 마이크로컨트롤러로 256바이트 RAM과 64KB EPROM을 기본 사양이 있다. 프로그램 개발 환경은 컴파일러로 IAR ICC8051를 사용하며, 그 밖에 Xlink, Xlib이 사용된다.

#### 3.2 제안된 임베디드 웹 데이터구조

일반적인 TCP/IP 프로토콜 스택은 데이터 연결 계층에서 응용프로그램 계층까지 다수의 계층으로 구성된다[8]. 그리고 개별 계층은 주변계층과의 인터페이스를 위하여 입출력 데이터 버퍼를 갖는 것으로 모델링이 된다. 상하 계층은 그 버퍼에 데이터 읽기/쓰기를 수행한다.

본 연구에서는 프로토콜 구현 시 RAM 메모리 사용량을 절약하기 위하여 공통전역변수 기법을 제안한다. 이것은 데이터연결 계층, 네트워크, 전달, 및 응용프로그램 전 계층에서 공통되는 1개의 변수를 사용하는 것이다. 또한 입력 또는 출력용으로 구별하여 데이터 버퍼를 사용하지 않고 제안된 1개의 공통 전역변수 버퍼에 데이터의 읽기/쓰기를 수행한다. 물론 읽기 또는 쓰기가 동시 진행되지 않도록 하여야 한다. 이것은 프로그램의 알고리즘으로 처리하였다. 그림 2는 제안된 데이터의 구조이다. 제안된 메모리 버퍼는 패킷 데이터의 저장에 사용되는데 마이크로컨트롤러의 내부메모리 부족으로 외부 메모리에 할당된다. 그리고 데이터버퍼는 프로토콜의 모든 계층간의 입출력 인터페이스에 공통으로 사용된다.

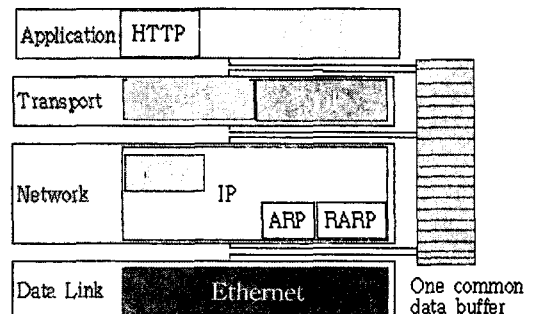


그림 2. 제안된 프로토콜 계층 및 데이터 모델  
Fig. 2. Proposed protocol layer and data model

### 3.3 임베디드 웹 프로토콜의 구현

주 메모리 용량의 제한에도 불구하고 임베디드 웹 제어기는 다른 인터넷의 호스트와 호환성을 달성하여야 한다. 전체 TCP/IP의 기능 중에서 어떤 것을 생략하여 목표를 달성할 것인가를 판단하여야 한다.

웹 프로토콜에 관한 국제표준 RFC 1122 또는 RFC 1123에 따르면 일부 기능의 생략으로도 호스트 간의 호환성이 이룩된다. 즉 임베디드 제어시스템 같은 특별한 상황에서 표준규격의 일부기능을 생략하는 것이 가능하다. 본 임베디드 웹 제어기에서는 몇 가지를 생략하였다. 예로서 긴급하게 데이터를 처리하는 기능, 혼잡제어, 및 데이터양이 적을 경우 IP 패킷의 조각 및 조립기능이 필요 없다[9].

#### 3.3.1 이더넷 드라이버

개발된 이더넷 네트워크 드라이버는 상위 네트워크계층으로부터의 패킷을 인터넷으로 패킷을 전송하는 역할을 하거나, 그 반대로 작동을 하게 한다.

네트워크제어기는 초기화 과정을 거친 후 원형수신 버퍼에 패킷이 도착하였는지를 검사한다. 만약에 데이터가 수신되면 수신버퍼에서 공통전역버퍼로 이동된다. 데이터처리 과정은 먼저 수신데이터에서 4바이트 헤더정보를 얻고, 연결계층에서 체크섬을 하여 수신데이터의 에러검사를 하고, 공통전역버퍼로 이동한다. 드라이버는 주기적으로 네트워크제어기의 인터럽트상태레지스터(Interrupt State Register: ISR)로부터의 패킷 도달여부를 검사한다. 네트워크제어기 내부에는 16[Kbytes]의 RAM이 있고 칩 내부에 제어 기능이 있어 복수의 패킷을 저장할 수도 있다. 만약 버퍼가 가득 차면 수신된 패킷을 제거하기도 한다. 드라이버에서 네트워크로의 전송도 같은 절차를 거치는데, 이때는 전송하는 데이터를 공통전역버퍼에서 네트워크제어기의 버퍼로 이동한다.

#### 3.3.2 네트워크 계층 프로토콜(IP)

개발된 IP 계층은 수신패킷의 IP 헤더정보에 대한 올바른과 ARP, ICMP, UDP, TCP 프로토콜 사이에 역 다중화(De-multiplex)를 한다.

ARP는 32비트 IP 주소와 48비트 MAC 주소사이에 변환작업을 한다. 클라이언트에서 ARP 요청에

대하여 개발된 ARP 프로토콜에서는 응답(Reply)을 하는 Echo-Reply 기능을 구현하였다. 그리고 ICMP 체크섬도 구현하였다[10]. 시간표시(Timestamp), 재전송(Redirect) 같은 에러처리 등의 다른 기능들은 생략하였다.

수신 IP 패킷은 목적지 IP 주소와 자신의 IP 주소의 부합여부를 검사하고, IP 헤더의 체크섬을 한다. 그러나 IP 패킷 조각내기 및 조립(Reassembly)은 구현하지 않았다. 이것은 임베디드 웹 제어기와 같은 경우에는 패킷 조각내기가 필요하지 않기 때문이다. 그리고 IP 옵션도 구현하지 않았다.

#### 3.3.3 전달계층 프로토콜(TCP/UDP)

전달계층은 네트워크상의 일대일 연결설정 및 양방향 데이터 전달기능을 수행한다. 두 지점간의 양방향 데이터 전송을 위하여, 두 지점은 지속적으로 데이터를 추적하여 데이터의 생략, 중복 등을 검출한다.

TCP는 유한상태기계(Finite State Machine)로서 작동하고, 어떤 이벤트가 발생할 때 까지 그 상태를 유지한다[8]. 프로토콜 구현에서는 상태변화를 잘 다루어야 한다. 상태기계를 임베디드 시스템에서 구현하는데 상태정보를 RAM에 저장하게 된다. 수신된 패킷은 체크섬을 확인 한 후에 출발지/목적지의 포트/IP주소가 검사된다.

흐름제어에 보통 이동윈도우(Sliding Window) 방법이 사용되는데 연구에서는 간단한 윈도우(Window) 방법을 사용하였다. 개발된 임베디드 제어기에서는 1개의 TCP 조각에 대하여만 응답하는데(Acknowledge), 그래도 호스트간의 호환성이 유지된다. 데이터 전송 중에 에러가 발생하면 재전송하여야 하는데 본 연구에서는 고려하지 않았다. 네트워크 성능향상을 위하여 매번 연결설정마다 데이터의 왕복여행시간(RTT)를 계산하는데 이러한 기능을 생략하였다. 혼잡제어는 동시 다수 접속에 대한 제어를 위하여 필요한데 연결마다 1개의 TCP 조각을 가정하여서 생략되었다.

## 4. 실험 및 결과

그림 3은 개발된 임베디드 웹 제어기(MiniWeb)

[11]에 TS80C32 마이크로컨트롤러와 RealTek의 RTL8019 네트워크제어기 등이 보인다.

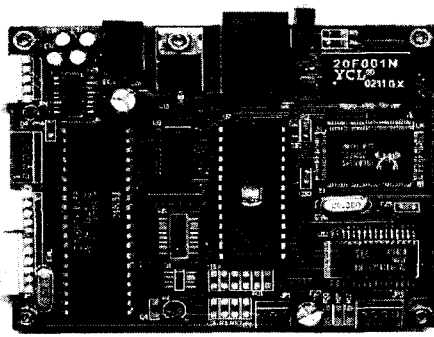


그림 3. 개발된 임베디드 웹 제어기  
Fig. 3. Developed embedded web controller

개발시스템에 네트워크 드라이버, ARP, IP, UDP, ICMP, TCP, 및 HTTP 프로토콜을 구현하여 컴파일 하였을 때, 동작되는 프로그램은 데이터의 임시저장을 위하여 내부 RAM 약 200바이트, 공통 전역변수 저장을 위하여 외부 RAM 1165바이트를 소모하며, 또한 프로그램 저장을 위하여 EPROM 5260 바이트를 소모하는 것으로 분석되었다. 물론 이러한 소모량은 목표 마이크로컨트롤러에서 사용되기에 문제가 되지 않는다.

보통 TCP/IP 프로토콜에서 데이터처리는 CRC (Cyclical Redundancy Check)계산, 버퍼간의 데이터 복사와 이동, 그리고 문맥전환(Context Switching)에서 주로 영향을 받는다 [12]. TCP의 QoS 향상을 위하여 네트워크 성능 분석이 활용된다[13]. TCP/IP의 성능은 주로 재전송 시간 초과(Retransmission Time Out : RTO)에 의하여 주로 영향을 받는다. 그래서 네트워크 성능 향상을 위하여 패킷의 왕복여행시간(Round Trip Time : RTT)을 감소하도록 하여야 한다. 개발된 프로토콜에 대한 RTT 테스트를 위하여 Dummynet을 사용하였다[14]. 실제 네트워크에서 TCP는 네트워크 트래픽을 관찰하여 RTT를 조정하여 성능을 향상하도록 노력하고 있다. 따라서 RTT의 측정은 프로토콜 성능예측에 중요하게 사용된다. 여기서 Dummynet은 실제 동작중인 네트워크에서 프로토콜을 분석할 수 있는 시뮬레이

터이다. 그림 4와 같이 테스트를 위하여 외부 네트워크에서 침입이 없는 고립시스템을 구성하였다. 466[MHz], Intel Celeron 128[MB] 사양의 컴퓨터에 Dummynet을 구동하고 MiniWeb 보드[11]와 10[Mbps] 이더넷 크로스케이블로 연결하였다.

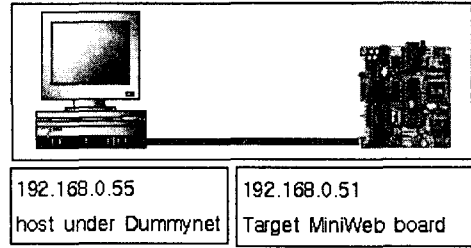


그림 4. 개발시스템에 대한 실험시스템  
Fig. 4. Experiment system for testing

IP주소가 192.168.0.55인 호스트 컴퓨터가 IP주소가 192.168.0.51인 MiniWeb 보드에 유틸리티 Ping 패킷을 전송한다. MiniWeb보드는 호스트로부터의 ICMP Ping 요청에 응답한다. Dummynet을 사용하여 RTT를 측정하였고, ICMP 패킷은 64바이트이다. 그리고 통계분석을 위하여 300회 이 과정을 반복하였다. Dummynet을 위하여 PicoBSD 버전을 사용하였다[14]. 결과가 표 1에 나타나 있고 1.8[GHz] Pentium 4 컴퓨터에는 윈도우XP가 동작 중이다. 결과는 상대적으로 느리지만 UDP, ICMP 프로토콜이 정상적으로 작동함을 알 수 있다. 이는 마이크로컨트롤러가 패킷 데이터 처리를 위하여 외부 RAM을 접근하기 때문이다.

표 1. RTT 측정 결과  
Table 1. RTT measurement results

RTT(ms)	TS80C32	Pentium 4
min/avg/ max/std	67.76/70.35/ 72.86/1.45	0.39/0.51/ 0.71/0.08

Ping은 TCP/IP가 올바르게 설정되었다는 것을 판단하는데 유용하게 사용되는데, 이것은 IP가 자신에게 패킷을 라우팅 할 수 있다는 것을 의미한다.

프로토콜의 성능 분석에 이어, 동작의 올바름 (Correctness)에 대한 측정을 위하여 CommView 소

Web browser를 이용한 가정용 가스보일러 제어기술 개발

프트웨어를 사용하였다[15]. 연결 설정 중에 이더넷, IP, 및 UDP 프로토콜의 동작상태가 그림 5에 보여진다. CommView 실행화면에는 TCP/IP 상태변수 데이터가 나타나있다. 이더넷 계층정보인 MAC 주소(52 : 46 : 54 : 41 : 47 : 31), 이더넷 상위계층 프로토콜 형식 정보(0x0800), 이더넷 프레임크기(63바이트), UDP 프로토콜 정보 등이 나타나 있다. 이는 개발한 임베디드 웹 서버 및 HTTP 관련 프로토콜이 잘 동작함을 의미한다.

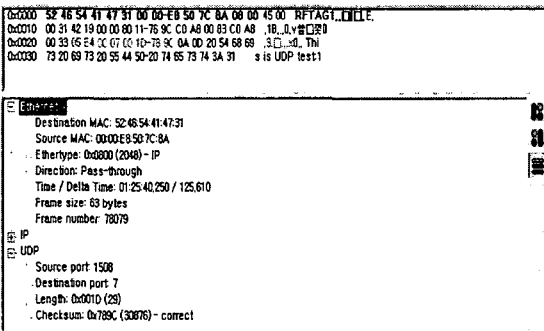


그림 5. CommView를 통한 프로토콜 분석  
Fig. 5. Protocol analysis using CommView

일반적인 아파치 웹 서버는 CGI(Common Gateway Interface)를 통하여 서버에 전송된 데이터를 얻지만, 마이크로컨트롤러를 기반으로 한 임베디드 웹 서버는 운영체제 및 파일 시스템이 없으므로 웹 문서의 관리 및 장치제어가 쉽지가 않다.

다음은 개발된 프로토콜 스택을 인터넷의 다른 컴퓨터 웹 브라우저를 통하여 실험하였다. 클라이언트의 웹 브라우저 URL에 임베디드 웹 제어기의 주소 http://192.168.0.51를 입력하였다. 인터넷을 통하여 웹 제어기에 접속하는 것이 가능하며, 웹 제어기는 클라이언트에게 그림 6과 같은 웹 문서를 제공하였다. 웹 브라우저에서 운전/정지 제어명령을 설정한 후에 송신명령(Send)을 보내면 클라이언트로부터의 명령이 웹 제어기의 RS-232를 통하여 보일러제어기의 RS-232에 전달된다. 웹 브라우저를 통하여 보일러의 운전/정지 제어 및 상태감시를 할 수 있다는 것을 보여준다. 결론적으로 임베디드 웹 제어기에 웹 서버가 설정되었으며 주변장비의 제어가 가능하게 된다. 제한된 메모리 및 운영체제의 부재에도 불구하고

하고, 웹 제어기로부터 웹 문서를 성공적으로 수신할 수 있었다.



그림 6. 웹 제어기의 접속에 대한 웹 브라우저 화면  
Fig. 6. Web browser's view for web controller connection

## 5. 결 론

본 연구에서 기존의 가정용 가스보일러의 운전/정지 및 운전 상태변수 설정과 간단한 제어항목을 갖는 경우에 적합한 제어기를 인터넷 보일러로 병행하게 운전하도록 하는 임베디드 웹 제어기를 개발하였다.

가격 면에서 경쟁력 있도록 하기 위하여 메모리 용량이 제한된 8비트 마이크로컨트롤러에서 TCP/IP 프로토콜 스택을 구현하고자 인터넷 계층에 공통으로 사용될 수 있는 데이터 모델을 제안하였고, 또한 구현된 결과가 인터넷상의 다른 호스트와 호환성을 잃지 않으면서 원하는 기능을 수행하였다. 구현에 사용된 RAM 사용량은 200바이트 내외로 TS80C32의 RAM에서 충분하였다.

그리고 Dummynet와 CommView라는 네트워크 성능 및 프로토콜 분석기를 통하여, 구현된 임베디드 웹 프로토콜이 성능은 떨어지지만 프로토콜의 올바른 바름에는 아무 문제가 없는 것으로 분석되었다.

끝으로 개발된 프로토콜이 TCP/IP의 전체기능을 만족하지는 않으므로 계속하여 추가적인 연구를 하는 것이 필요하다고 판단된다.

본 연구는 중소기업청 산학연컨소시엄 사업 지원에 의하여 수행되었음.

## References

- [1] R. Braden, Requirements for internet hosts - communication layers. RFC 1122, Internet Engineering Task Force, October 1989.
- [2] Adam Dunkels, "Full TCP/IP for 8-bit Architecture", In Proceedings of the First International Conference on Mobile Systems, Applications and Services (MobiSys), San Francisco, May 2003.
- [3] A. Dunkels. uIP - a TCP/IP stack for 8- and 16-bit microcontrollers. Available from dunkels.com/adam/uip/, 2004.
- [4] H. Shrikumar. "IPic - a match head sized web server", Available from www-ccs.cs.umass.edu/~shri/iPic.html, 2004.
- [5] J. Bentham, TCP/IP Lean: Web servers for embedded systems. CMP Books, October 2002.
- [6] Atmel Corporation, Embedded web server. AVR 460. Available from www.atmel.com, January 2001.
- [7] S. Shon, "KOSEF Research Final Report", KOSEF, 2003.
- [8] Behrouz A. Forouzan, "TCP/IP Protocol Suite", McGraw-Hill International, page 297-299, 2000.
- [9] David D. Clark, John Romkey, Van Jacobson, Howard Salwen, "An Analysis of TCP Processing Overhead", IEEE Communications Magazine, Pages 23 - 29, June 1989.
- [10] A. Rijsinghani, Computation of the internet checksum via incremental update. RFC 1624, Internet Engineering Task Force, May 1994.
- [11] S. Shon. MiniWeb ethernet kit. Embedded system Lab., University of Suwon, 2004.
- [12] J. Kay and J. Pasquale, The importance of non-data touching processing overheads in TCP/IP, In Proceedings of the ACM SIGCOMM '93 Symposium, pages 259-268, September 1993.
- [13] Craig Partridge, Jim Hughes, Jonathan Stone, "Performance of Checksums and CRCs over Real Data", In Proceedings of ACM SIGCOMM '95 (Jan '95). Pages 68-76, 1995.
- [14] L. Rizzo, Dummynet: a simple approach to the evaluation of network protocols. ACM Computer Communication Review, 27(1):31-41, 1997.
- [15] TamoSoft inc. CommView<sup>®</sup>, Available <http://www.tamos.com/products/commview/>, 2004.

## ◇ 저자소개 ◇

### 손수국 (孫守國)

1959년 7월 15일생. 1982년 서울대학교 전기공학과 졸업. 1984년 서울대학원 전기공학과 졸업(석사). 1996년 미국 University of Texas at Austin 공학박사. 현재 수원대학교 정보통신공학과 조교수.