

---

# 임베디드 네트워크 프로세서 개발

유문종\* · 최종운\*\*

Development of Embedded Network Processor

Moon-jong Yoo\* · Jong-woon Choi\*\*

---

이 연구는 2001년 산학연공동기술개발 컨소시엄과제의 지원을 받아 수행되었음

---

## 요 약

8비트 마이크로 프로세서인 TMP84C015과 RTL8019AS를 사용하여 이더넷 물리층을 구성한 뒤 Embedded HTTP서버를 구성하였다. 프로토콜의 사양을 최소한으로 선택한 뒤, 패킷 전송률을 최대화하기 위해서 어셈블리어로 코딩하였다. 서버 구현에 필요한 TCP, UDP, IP, ICMP, ARP 프로토콜이 내장되어 있으며, Client 측에서 각각의 프로토콜의 동작을 LabVIEW를 사용하여 확인하였다.

## ABSTRACT

This is an example of ABSTRACT format. We made a HTTP server using 8 bit microprocessor. It was TMP84c015 which applied a z80 core and RTL8019AS was installed for an ethernet physical layer. Assembly language was used to optimized a performance of the MPU, to overcome an restriction of memory size and to maximize the throughput of packet using TCP, UDP, IP, ICMP, ARP protocol. We used LabVIEW to verify the each protocol on the client side.

## 키워드

TCP/IP, HTTP, Embedded, 웹서버, 산학연공동

## 1. 서 론

생활의 일부가 된 인터넷의 응용 분야는 폭발적으로 증가하고 있다. 최근 국내의 유선통신 사업자들이 초고속통신이라고 부르는 ADSL망을 각 가정 및 소규모 사무실까지 경쟁적으로 서비스를 제공하면서 유선 인터넷의 전송속도는 기존의 전화망을 통한 속도에 비하여 가격대비 속도가 30배 이상 증가하였다. 이와

같은 통신망의 발달은 인터넷을 통한 원격계측이 가능하도록 하고 있다. 기존 네트워크에서 통신을 위한 최소한의 프로토콜을 사용하기 위해서는 OS에 실려있는 프로토콜 스택을 사용하여야만 하였다. 원격계측용 프로세서의 경우에는 주로 16비트 이상의 프로세서에 Linux, VxWorks, WinCE등의 OS를 사용하여 데이터를 전송하였다. 그러나 일반적으로 원격계측기에서는 대량의 데이터가 발생하지 않는 경우가 많이 있기 때

---

\*호남대학교 대학원 정보통신공학과

접수일자: 2001. 11. 13

\*\*호남대학교 정보통신공학부 전임강사

문에 단순히 짧은 데이터 전송을 위해서 16비트급 MPU와 OS를 사용하는 것은 경제적으로 낭비적인 요인이다.

본 논문에서는 8비트급 마이크로프로세서를 사용하여 HTTP 서버를 구현하였다. 사용한 프로세서는 Z80 코어를 채용한 TMP84C015이고, 이터넷의 물리층은 RTL8019AS를 사용하여 구현하였다. 8비트 프로세서라는 제약과 사용 가능한 메모리의 제한을 극복하기 위하여 프로토콜을 최대한 단순화하였고, 시간당 보낼 수 있는 패킷의 수를 최적화하기 위하여 어셈블리어언어를 사용하여 TCP, UDP, IP, ICMP, ARP 프로토콜을 구현하였다. 클라이언트 측에서는 LabVIEW를 이용하여 설계 제작한 임베디드 서버의 동작을 확인하였다.

## II. 프로토콜 구성

현재 사용되는 가장 일반적인 인터넷 통신망의 구성은 물리층을 UTP케이블<sup>1)</sup>과 Manchester Encodin<sup>2)</sup>을 사용하고, 데이터링크 계층은 802.3[3]을 사용하며 우리는 Ethernet Adapter를 보통 '랜카드'라고 부른다. 그리고 네트워크층을 IP로 전송계층은 TCP로 사용하고 그 위에 HTTP 프로토콜을 사용하는 웹브라우저를 사용하고 있다. (그림 1)

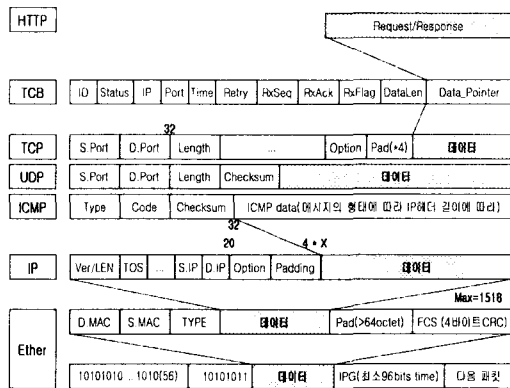


그림 1. 프로토콜 스택

물리층과 데이터링크층은 최근 가장 많이 사용되고

- 1) Unshielded Twisted Pair : 차폐층 없는 꼬인선
- 2) Biphasse 방식중 biphasse-L방식의 별명. 동기화가 가능하고 변조율이 2배, 오류복원이 쉬운 장점과, 대역폭이 2배로 필요한 단점이 있다.[3]

있고, 비교적 자료가 풍부한 RTL8019AS[10]를 사용하였다. 다음은 IP와 TCP인데 프로그램 작성에 큰 무리가 없도록 비교적 여유 있게 ROM, RAM을 할당하였다. 이는 완성된 이후에 쉽게 최소한의 사양으로 재구성이 가능하다.(그림 2) 8비트 Parallel 포트를 이용하여 하위 Nibble은 LED에 상위 Nibble은 스위치에 할당하여 서버와 클라이언트 사이의 데이터교환을 확인할 수 있도록 구성하였다.

## III. 프로그램 구조

프로토콜은 운영체제를 이용하지 않고 데이터교환을 위한 부분만을 선택적으로 프로그래밍 하였다. 프로그램의 전체적인 흐름을 그림 5의 순서도에 표시하였다. 초기에 전원이 투입되거나 RESET이 눌러지면 첫 번째 초기화가 진행된다. 초기화 과정에서는 IP 및 MAC주소의 일련번호 부분의 변경이 가능하다. 또한 EEPROM 전체의 내용을 초기 값으로 새로이 기록하는 것이 가능하도록 하였다. 이는 윈도우에서 하이퍼터미널을 이용하였으며, 물론 프로그램 내부에 Check Sum등의 계산을 새로이 하여 기록이 되도록 하였고, 128바이트 용량 중 일부만을 사용하고 있으므로 여기에 상당한 내용을 추가하여 기억시키는 것이 가능하다.

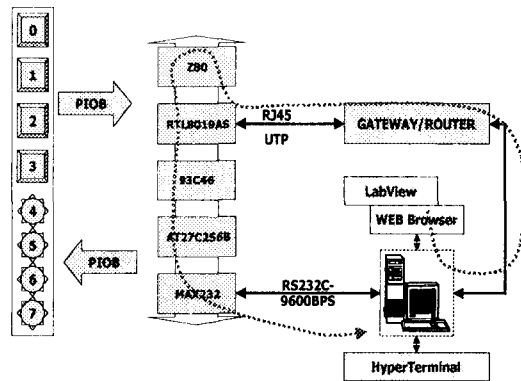


그림 2. 하드웨어 구성도

초기화 과정에서 특별한 사항이 없으면 시스템을 다시 초기화한 후 Ethernet 청취상태로 들어간다. 청취상태에서 패킷이 검출되면 ARP, IP만을 사용한다. ARP 패킷이면 ARP응답을 하게 되고, IP이면 ICMP/UDP/TCP 이외의 패킷은 버리게 되며, ICMP는

ICMP echo에만 응답하도록 되어있으나 필요하면 확장이 가능하다. TCP는 HTTP에만 응답하도록 하였으며, 포트를 제한하여 필요없는 패킷의 접근은 막았다. 가장 간단한 서버가 되도록, 패킷의 크기를 512바이트를 넘지 않도록 하였다. 또한 윈도우 버퍼 크기도 같은 크기로 제한하여 슬라이딩 윈도우, 패킷의 재조립 등의 문제가 발생하지 않도록 하였다. 이 때 문제는 패킷의 재전송인데 이는 TCB를 최대한 이용하도록 하고 있다. TCB의 최대 숫자는 8로 하였으나 남아있는 메모리를 이용하는 정도이고 각각 다른 Client에서 서로 다른 정보를 요구하는 경우가 아니라면 2개면 충분할 것으로 보며, 웬만한 응용에서는 HTTP를 상시 연결된 상태로 사용하지 않기 때문이다. 클라이언트로는 여러 가지 편리한 점 때문에 LabVIEW<sup>3)</sup> 프로그램을 이용하였다. LabVIEW는 데이터 수집이나 기초적인 분석에서 아주 뛰어난 편리한 인터페이스와 기능을 가진 프로그램이다. 그림 3에 LabVIEW와 보드가 연결된 상태를 보여준다. 참고로 LabVIEW화면에서 마우스로 Button을 Click하면 보드의 LED를 On/Off할 수 있고, 보드에서 Button 스위치를 누르면 LabVIEW상에 그 상태가 나타난다. 웹브라우저에서 이를 구현할 때는 CGI<sup>4)</sup>의 GET나 POST를 이용하여야 하고, 이를 웹서버에서 처리해 주는 부분이 있어야 한다. 물론 문자데이터에서 필요한 명령어를 검색하여 처리하는 방식이 된다. 따라서 쿠키(Cookie)<sup>5)</sup>를 이용하는 방안도 좋은 방안이 될 수 있으나, 프로그램 크기가 커지고 복잡해진다.

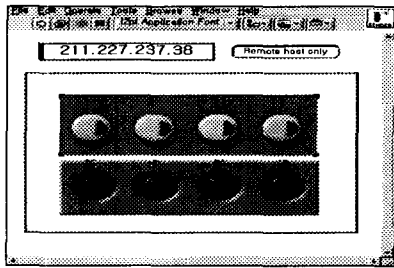


그림 3. LabVIEW를 사용한 화면

3) © Copyright 1999, 2000 Part Number 370110A-013) Labview National Instruments Corporation. All rights reserved. July 2000

4) Common Gateway Interface

5) 쿠키 : 웹서버가 클라이언트의 환경 또는 접속통계 등을 쉽게 알 수 있도록 Client측에 저장되는 문자열.

## N. 웹서버와 TCP

프로그램 작성 중 가장 난해한 부분은 TCP 부분이다. 1개의 패킷 전송은 UDP와 별반 다를 것이 없으나, 연결을 설정해놓고 모든 처리를 상대방과 협상하는 형식으로 진행되기 때문이다. TCP 프로그래밍을 하기 위해서 반드시 지나가야만 하는 과정으로는 첫째 하위의 IP 프로토콜과 상위의 HTTP프로토콜 및 Socket 개념이 필요하다. 둘째 TCP의 유한상태기계(Finite State Machine)에 대한 이해가 필요하다는 것이다. 이는 두 방향에서 진행되어야 하는데, 서버와 클라이언트 양쪽을 모두 고려하여야 하고, 상태 전이를 일으키는 요소와, 요소에 따른 TCB[1]의 변화 및 그 처리에 관하여 확실한 정리를 해둘 필요가 있다. 셋째로 TCP의 모든 기능을 필요로 하지 않기 때문에 시스템의 능력에 따라 적절한 삭감이 필요하다는 점이다. 대부분의 Embedded Processor는 필요한 성능을 만족하는 범위에서 소형, 저전력, 저가격이 필수조건이기 때문에 다량생산성을 만족하는 수준 또는 생산의 편의성 측면에서 최소 사양을 선택하여야만 한다. 이 시스템은 아직 최적화 단계를 지나지 않았기 때문에 미비한 점이 많이 있다. 이 논문에서는 TCP서버 작성에 필요한 부분을 집중적으로 논의하고자 한다.

### 가. State Change와 명령 전달(Flags)(1)(2)

그림 4에 TCP의 상태변화 중 서버의 상태변화를 굵은 선으로 나타냈다. 다만 RST 플래그를 가진 패킷이 입력되었을 경우에는 바로 Listen 상태로 이동하게 되고 혹시라도 이후에 도착되는 패킷은 무시되게 된다. 여기에 나타난 것은 정상적인 연결의 설정과 해제를 나타낼 뿐이고, 그 외의 ACK를 기다리다 설정된 시간이 되면 패킷을 재전송 하는 등의 예외적 처리가 있다.

### 나. 개략적인 순서도

TCP 모듈로 패킷이 전달되면, 먼저 HTTP인가를 검사하여 만약 HTTP이면 체크섬을 검사하게 된다. 체크섬에 이상이 없으면, FLAG를 조사하는데, URG는 검사하지 않고, 다른 FLAG에 대한 검사도 필요한 최소한의 조합만을 수행한다.

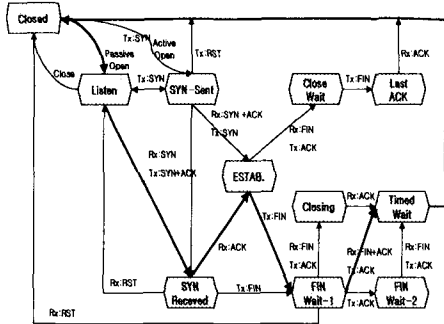


그림 4. TCP 프로토콜의 상태변화도

만약 SYN 패킷이 입력되면 이에 대한 응답으로 패킷의 최대 길이를 제한하는 Option을 SYN/ACK에 덧붙여 전송한다. FLAG 검사에서는 TCB 생성과 삭제에 관련된 내용과 데이터의 주고받음에 관한 내용만 검사한다. 한가지 유념해야 할 사항은 웹클라이언트로부터 데이터를 받아들여야 한다(Establish=설정 상태)는 것이다. 서버에서 전송되는 홈페이지에 CGI의 GET를 이용한 폼(FORM)을 전송하고, 사용자가 EVENT를 일으키도록 하여 서버가 응답하도록 구성하였다. 따라서 HTTP 모듈 내부에는 CGI TEXT를 처리하는 부분이 필요하고, 여기에서 데이터의 수집 및 반영도 처리하고 있다.

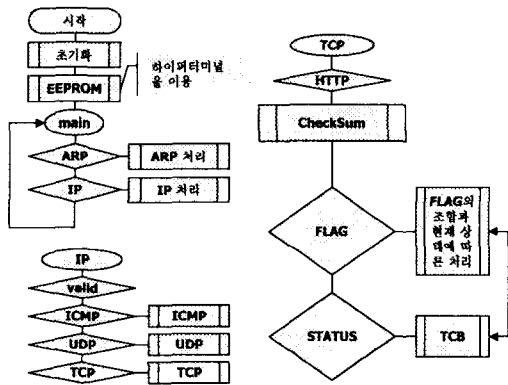


그림 5. 개략 순서도

또 한가지의 방법은 연속적인 화면의 갱신이 필요할 경우이다. 이 때는 HTTP 모듈이 타이밍 신호에 의하여 연속적으로 데이터를 전송하도록 하여야 하는데, 이 경우의 대처 방법은 시간제한성에 따라 프로그램이

다르거나 두 가지 경우 모두를 생각해야 한다. 예를 들어 CPU의 처리능력의 한계점에 가까워질 때는 TCP/HTTP가 계속 연결된 상태로 데이터 요청 패킷이 없어도 데이터를 계속 보내야 하는데 이러한 경우 CPU에 부담을 덜 주게되는 UDP를 시간 인터럽트 형태로 이용하는 편이 더 낫을 것으로 보인다. CPU의 처리능력이 충분할 때는 JavaScript의 Set/ClearTime Out()등을 이용하여 필요한 시간 간격으로 클라이언트가 데이터를 요청하고, 서버는 응답하는 시간을 제외한 데이터 요청이 없는 유휴시간을 다른 종류의 처리에 이용하는 것이 더 효율적이다.

### V. 하드웨어 구성

하드웨어(그림 2 참조)는 시험용이므로 포트를 많이 설치하지 않았고, 스위치나 LED를 기판에 직접 배치하였다. 외부 연결단자는 8비트 연결이 가능하도록 9핀 Connector를 배치하였다.

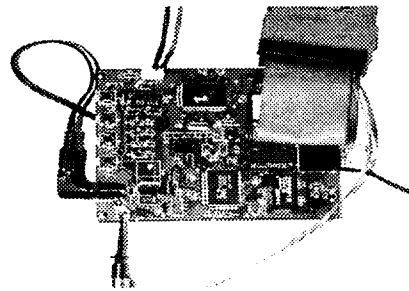


그림 6. 실험에 사용한 하드웨어

전원공급도 외부전원 5V를 공급하는 단자와 비안정화 6~12V 단자 2종을 설치하였다. 다만 8V 이상의 전압이 인가되면 정전압 IC에서 열 발생이 전압에 따라 증가하였다. EEPROM의 프로그래밍은 RTL8019 AS를 통하여 이루어지도록 하였으며 초기화된 EEPROM을 장착해야 하므로 롬라이터가 필요하다. 실제로는 2개의 EEPROM을 이용하여 읽기/쓰기를 분리하는 방식으로 처리하였다. QFP package의 LSI는 실험실에서 제작한 PCB에 부착하는 것이 쉽지 않았으나, 모두 실험실에서 수작업으로 부착하였다. 프로그램 작성 중에는 NLT TromICE-II+6) 에뮬레이터를 이용하였

6) NLT TromICE-II+ : NL Telecom Co.,Ltd. 의 롬 에뮬레이터

고, 완성된 보드에는 프로그램을 EPROM에 저장하여 완료하였다.

### V. 결 론

8비트급 마이크로프로세서를 사용하여 HTTP 서버를 구현하였다. 사용한 프로세서는 Z80 코어를 채용한 TMP84C015이고, 이더넷의 물리층은 RTL8019AS를 사용하여 구현하였다. 8비트 프로세서라는 제약과 사용 가능한 메모의 제한을 극복하기 위하여 프로토콜을 최대한 단순화하였고, 시간당 보낼 수 있는 패킷의 수를 최적화하기 위하여 어셈블리어를 사용하여 TCP, UDP, IP, ICMP, ARP 프로토콜을 구현하였다. 클라이언트 측에서는 LabVIEW를 이용하여 설계 제작한 임베디드 서버의 동작을 확인하였다. 일반적으로 LAN 카드를 이용한 전송이 훨씬 더 고속전송이 가능하나, CPU에 부담을 많이 주게 되므로 처리속도가 낮은 8bit 마이크로 controller에서 실제로는 직렬전송에 비하여 아주 빠른 속도를 낼 수는 없다. 다만 전송계(데이터링크층 이하)는 CPU가 거의 관여하지 않기 때문에 제어정보 수준의 데이터를 전송하는데 문제가 될 것은 없다.

### [감사의 글]

본 연구는 2001년 산학연공동기술개발 컨소시엄과 제의 지원을 받아 수행되었습니다. 관계자 여러분들에게 감사를 드립니다.

### 참고문헌

[1] Jeremy Bentham, TCP/IP Lean, CMP Books, 2000, chapters 6~7.  
 [2] Douglas E. Comer & David L. Stevens, Internetworking with TCP/IP Volume II 3rd Ed., Prentice-Hall Inc., 1999, chapters 11-14.  
 [3] 강창연, 정보통신이론, 북두출판사, 1995.7. p136.  
 [4] RFC 793 Transmission Control Protocol. J. Postel. IETF, Sep-01-1981.

[5] RFC 791 Internet Protocol. J. Postel, IETF, Sep-01-1981.  
 [6] 아세쓰 아끼히로의 공저, 인터페이스 시리즈8, Sep-25-1997, p42.  
 [7] 윤중호, TCP/IP 및 윈도우 네트워킹 프로토콜, (주)교학사, 1999.3.10., 14장-18장.  
 [8] 양홍석 외, 마이크로컴퓨터(Z80중심) 설계 및 응용, 동일출판사, 1998.  
 [9] James W. Coffron, Z80 Applications, Sybex, 1979.  
 [10] RTL DATA BOOK, REALTEK SEMI-CONDUCTOR CO., LTD.



유문중(Moon-jong Yoo)  
 전남대학교 자연대 물리학과 학사  
 1983~(주)화니백화점 전산실  
 1986~(주)동홍정밀 전자제어부  
 1989~무진컴퓨터학원  
 1994~한국전력 전남지사 정보화 전임강사  
 2000~호남대학교 정보통신공학 석사과정  
 ※ 관심분야 : Embedded 시스템, 센서, 광통신



최중운(Jong-woon Choi)  
 단국대학교 전자공학과 학사, 석사, 박사  
 전기통신기술사, 공업계측제어기술사  
 1981~1984: 육군기술장교  
 1986~1988: 해태전자 중앙연구소 연구원  
 1988~2000: 한국표준과학연구원 레이저그룹 책임연구원  
 2000~호남대학교 정보통신공학부 교수  
 ※ 관심분야 : 광통신, 임베디드 프로세서, 프로토콜

터, 홈페이지: <http://nltelecom.com/>