

# 임베디드 웹서버를 이용한 다관절 매니퓰레이터의 관절각제어

## Monitoring and Control of a Manipulator joint angle using Embedded Web Server

한주동\*, 이용현\*\*, 박종국\*\*\*, 최경진\*\*\*\*

\*경희대학교 전자공학과 (TEL : 031-201-2957 FAX : 031-204-4968 E-mail : [indinet@hanmir.com](mailto:indinet@hanmir.com))

\*\*강남대학교 지식정보공학부 (TEL : 031-280-3807 FAX : 031-280-3884 E-mail : [yhlee@kangnam.ac.kr](mailto:yhlee@kangnam.ac.kr))

\*\*\*경희대학교 전자공학과 (TEL : 031-201-2957 FAX : 031-204-4968 E-mail : [ckpark@khu.ac.kr](mailto:ckpark@khu.ac.kr))

\*\*\*\*경희대학교 전자공학과 (TEL : 031-201-2957 FAX : 031-204-4968 E-mail : [ckj6977@hotmail.com](mailto:ckj6977@hotmail.com))

**Abstract** : In this paper have embodied a TCP/IP based small embedded web server and proposed the technical method for monitoring the real-time system states and remote controlling system using HTTP about manipulator joint angle. This also uses TCP/IP, which is the general protocol, and it is made of software, not of the TCP/IP stack with hardware chip.

Then we attach the simulating results about the manipulator's input/output via web browser.

**Keyword** : Embedded Web-server, Embedded System, Remote Control and Monitoring, Manipulator.

### 1. 서 론

디지털화, 네트워크화가 급격히 가속화 되어감에 따라 네트워크는 의사 전달 방법만이 아닌, 인간의 살아가는 방식 또한 변화시키고 있다. 이에 발맞춰 최근 어떤 Device, Network, 어떤 환경이든 관계없이 인터넷을 통한 원격제어와 제어를 위해 건축물에 설치된 센서와 제어장치를 제어할 수 있으며, 기기 상호간의 네트워크 기능을 최소한의 자원만을 사용하는 임베디드 웹서버가 활발하게 나타나고 있다.

임베디드 시스템(Embedded System)이란 정해진 특정 기능의 수행을 위해 컴퓨터의 하드웨어와 소프트웨어를 조합한 전자 제어 시스템을 의미한다. 그 중 임베디드 웹 서버(Embedded Web server)는 여러 가지 형태의 입출력 장치와 지능형 정보 기기 들을 인터넷에 손쉽게 접속할 수 있게 하는 초소형 웹 서버로서, 임베디드 웹 서버의 적용 영역은 가정, 사무실, 빌딩 및 공장 등의 전 영역에 걸쳐 있다.

오늘날, 마이크로프로세서는 각종 전문화된 응용분야에 따라서 전용화 된 마이크로컨트롤러로 발전되어 수많은 지능형 정보 기기(Intelligent Information Devices)에 내장(Embedded)되어 있다. 이러한 지능형 정보 기기를 인터넷에 접속하여 원격지에서 기기를 감시 및 제어를 하기 위해서는 웹 콘텐츠(Web Contents)의 제공을 주목적으로 하는 일반 서버 컴퓨터(Server Computer)와는 다른 임베디드 운영체계를 이용한 전용화 된 서버가 필요하다.

SUN Microsoft의 JINI와 같은 분산 네트워크와 같은 기술도 이미 상용화되어 있는 상태이다. 하지만

JVM(Java Virtual 머신)을 내장한 시스템과의 통신이 가능하며, JVM을 내장하기 위한 고가의 프로세서와 많은 크기의 메모리를 가져야 하므로 System의 크기가 상승한다는 단점을 지적할 수 있다. 그리고 TCP/IP를 하드웨어 스택으로 구현한 Micro chip들도 상용화 되어 있다.

본 논문에서는 하드웨어 기반의 웹서버가 아닌, 8-bit Processor를 사용한 소형의 Embedded System을 구현할 것이며, 표준 프로토콜인 TCP/IP를 소프트웨어로만 구현할 것이다.

### 2. 본 문

#### 2.1 시스템의 구성

임베디드 웹서버를 구축하는 방법은 크게 두 가지로 분류가 된다. 첫 번째로, 많은 하드웨어(H/W) 회사들이 개발의 단축성을 위해 구현해 놓은 TCP/IP를 하나의 패키지로 설계한 경우이며, 두 번째로는 본 논문에서 언급하는 TCP/IP를 소프트웨어(S/W)로 구축하는 방법이다. H/W인 방법은 개발기간의 단축을 위해 유용하지만, 유지보수를 위한 비용이 적지 않게 소요된다. 이와는 달리 S/W적인 방법은 개발기간에 어려움이 따르지만, 유지보수가 효율적이다. 본 논문에서는 8-bit Processor를 가지고, S/W로 구현을 하였다.

8-bit Processor는 제한된 자원을 가진다. 특히 제한된 메모리의 크기는 임베디드 웹 서버의 구현에 있어 아주 취약한 부분으로 분류된다.

먼저 구현된 시스템의 하드웨어 블록도는 그림 1과 같다.

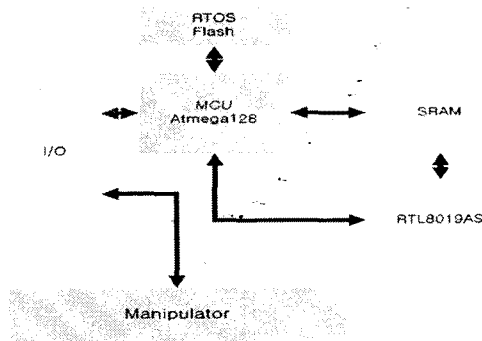


그림1. H/W 블록도  
Fig 1. H/W Block Diagram

소형 웹 서버의 사양을 보면 Atmega128 8bit MCU를 사용하며, 내부 128Kbyte의 flash를 가지는 프로그램 메모리를 가진다. MCU내부에 4Kbyte의 SRAM을 가지며, 40개의 입출력이 가능한 포트 가지고 여러 기능을 내장하고 있다. RTL8019AS는 송수신 되는 TCP/IP의 프레임 데이터의 버퍼역할을 수행하게 된다[1][2].

프로그램 메모리는 구현에 있어 충분한 저장 공간을 가지지만, TCP/IP를 통한 송수신되는 프레임을 저장하는 공간으로는 내부의 4K Byte공간으로는 매우 부족하다. 그래서 외부의 64K Byte의 External SRAM을 확장했다. 이러한 SRAM의 확장에도 불구하고 MUC의 I/O레지스터와 Work Space가 SRAM에 공유하기 때문에 송/수신하는 프레임을 제한하는 방법을 적용하여 네트워크상의 데이터 처리율이 떨어지지 않게 하여야 한다. 그래서 빠른 응답시간을 얻기 위하여 프레임을 수신하고 바로 Decode하고, 출력 프레임은 전송하기 바로 직전에 준비 된다[1][2].

구현된 Web Server는 그림 2와 같다.

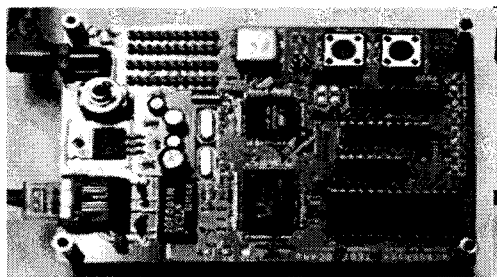


그림2. 임베디드 웹 서버  
Fig 2. Embedded Web-Server

## 2.2 S/W구성

본 논문의 Embedded Web Server에서는 두 가지로 구분되는 펌웨어로 구현되어있다.

첫 번째로 제한된 자원을 사용하는 임베디드 시스템의 자원을 효과적으로 사용하기 위한 OS부분과, 두 번째로 TCP/IP 스택을 구현한 부분으로 나누어진다.

임베디드 시스템은 PC나 UNIX 시스템처럼 부가적인 하드디스크를 가지지 못하기 때문에 그림 3에서 보여주는 것처럼 프로그램메모리에 최소한의 필요를 가지는 OS를 구현하였다[6][7].

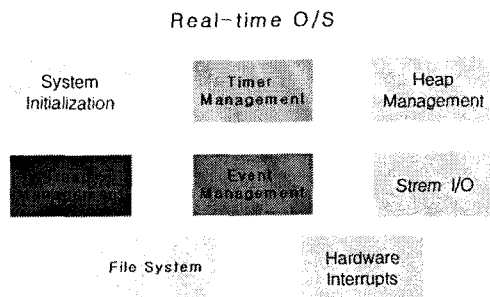


그림 3. 운영 체제 블록도  
Fig 3. Real-time O/S Block Diagram

특히 File System은 동적인 웹 콘텐츠를 위한 이미지 등을 처리하기 위해 Binary로 변환하여 프로그램 메모리에 적재하게 하는 역할을 하게 된다.

각 Task들은 62.5ms의 Tick Time을 가지고 Thread화되며, 웹 서버 각각의 입출력되는 데이터들은 네트워크 상에서 Virtual Device Stream형태로 데이터를 전송하고 처리하게 된다.

TCP/IP의 구현된 네트워크 계층의 프로토콜은 그림4에 나타내었다.

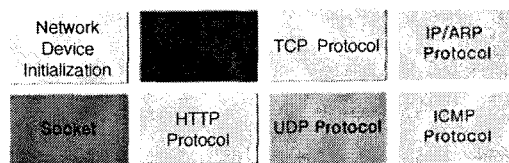


그림 4. 이더넷 프로토콜  
Fig 4. Ethernet Protocol

그림 4에 나타난 것처럼 웹 서버는 고정된 IP와 동적인 IP모두를 지원한다. 신뢰성을 가진 TCP와 비 신뢰성을 가진 UDP, 그리고 ICMP를 구현하여 Client에서 Ping과 같은 메시지에 응답하게 하였다[5][6][7].

## 3. 모의실험

HTTP(Hyper Text Transmitter Protocol)를 이용하여 원격지에서 Request 전달하고 서버에서 Response를 돌려보내게 된다. 비교적 지원되는 자원이 많은 PC나 UNIX System에서는 동적인 웹 콘텐츠를 위해

CGI(Common Gateway Interface)를 사용하지만 자원이 제한되어있는 Embedded Web Server에서는 제약점이 많이 따르기 때문에 EGI(Embedded Gateway Interface)를 구현하여 서버에 적재하여야 한다. 그림 5는 동적인 웹 콘텐츠를 위한 EGI블록도이며, 그림 6은 모의실험의 전체적인 블록도이다. 동적인 Web-Page는 프로그램 메모리에 모두 적재를 시키고(flash :128 Kbyte), 내부 램과 확장된 외부 램에서 네트워크에 관련된 프레임들을 적재/소멸시킬 수 있도록 하였다.

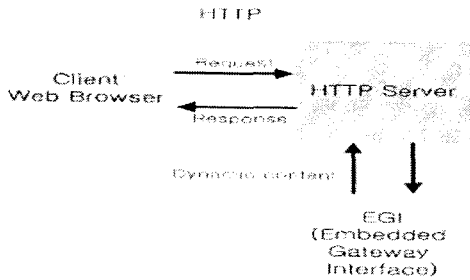


그림 5. 임베디드 게이트웨이 인터페이스 블록도  
Fig 5. EGI(Embedded Gateway Interface)

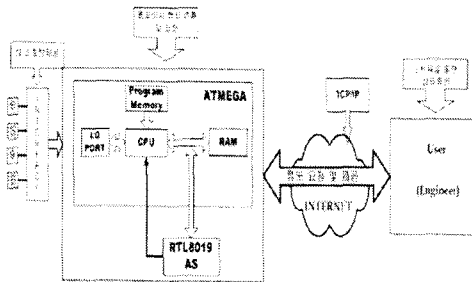


그림 6. 모의실험 구성도  
Fig 6 . Pseudo chart

아래에 보이는 그림 7은 Web Server의 초기 구성도이며, 그림 8은 Web Port컨트롤을 위한 암호화 페이지를 보여준다.

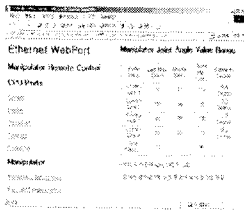


그림 7. 메인 페이지  
Fig 7. Main Page

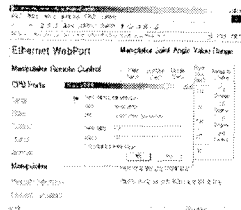


그림 8. 암호화 페이지  
Fig 8. Password Page

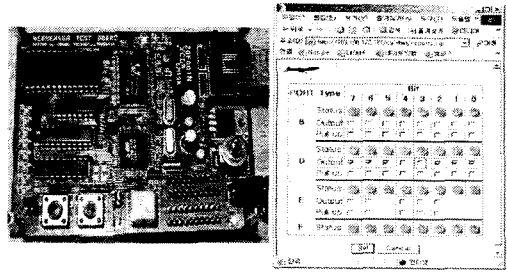


그림 9. 웹 포트 컨트롤 페이지  
Fig 9. Web Port Control Page

그림 9에서 Web Server에서 입출력할 수 있는 포트의 현재 상태를 보여 주고, 각 체크박스에서 해당 입출력 포트를 제어가능하게 하였다.

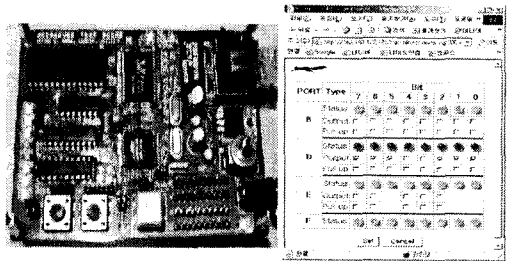


그림 10. 웹 포트 결과 페이지  
Fig 10. Web Port Result Page

그림 10에서는 웹 브라우저 상에서 Web server의 I/O port를 control하고, 그림 11에서 현재 웹 서버의 입출력상태를 매 1초마다 실시간으로 Reflash를 통해 모니터링 하여 보여 주고 있다.

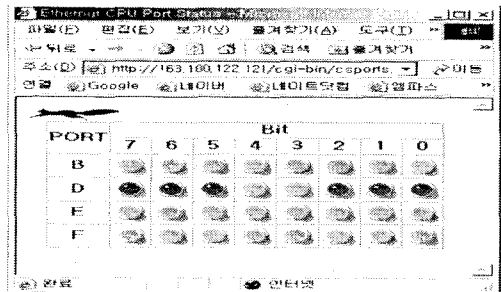


그림 11. 입/출력 모니터링 페이지  
Fig 11. I/O Monitoring page

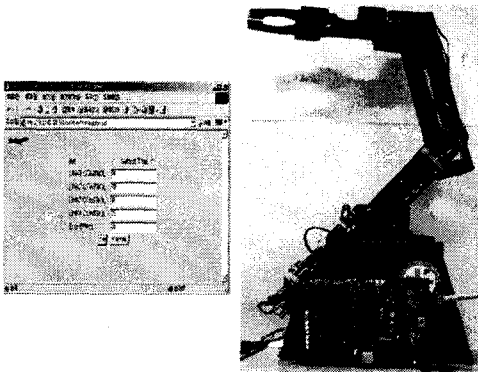


그림12. 매니플레이터 관절각 조절  
Fig12. Manipulator joint angle control page

그림12는 4자유도를 가지는 매니플레이터의 관절각을 각각의 관절에 대하여 제어 가능하게 하기 위한 페이지이다. Init와 Default의 radio button은 각각 매니플레이터의 초기위치치 가지며, Default버튼은 임의의 Task가 주어지지 않을 때 기본적으로 행하여지는 Task를 수행하게 된다.

#### 4. 결 론

이미 일반화 되어진 Network를 산업용이나 가정용 전자기에 접목하여 컨트롤하고 모니터링 하려는 기술은 널리 적용되어지고 있다. 대부분의 이러한 네트워크를 접목한 System들은 PC나 Unix System등 지원하는 자원이 풍부한 System에 적용 되고 있는 실정이다. Embedded System이 지향하는 것처럼 네트워크만을 구현하고 하나의 동작을 위해 최소한의 자원으로 소형의 System을 구성하는 것은 비용의 절감과 유지보수의 용이성을 증대 시켜 준다. I/O 레지스터만의 변화를 모니터링을 하고 있어 실제적인 시스템의 상태를 파악하지 못한다는 단점을 가지고 있다.

본 논문은 하나의 목적을 위한 소형의 웹 서버를 구현하고 실시간으로 모니터링과 제어가 가능함을 보이고 있다.

#### 5. 향후 과제

아래 그림 13에 나타난 것처럼 4개의 자유도를 가지는 Manipulator의 관절각을 정확하게 제어하는 알고리즘과 Web Server에서 Client에 독립적인 Task를 작성하고, Client에서 우선순위를 부여한 새로운 Task를 부여할 수 있게 해야 하며, Vision System을 통한 원격지 실시간 모니터링을 수행 할 수 있도록 해야 할 것이다.

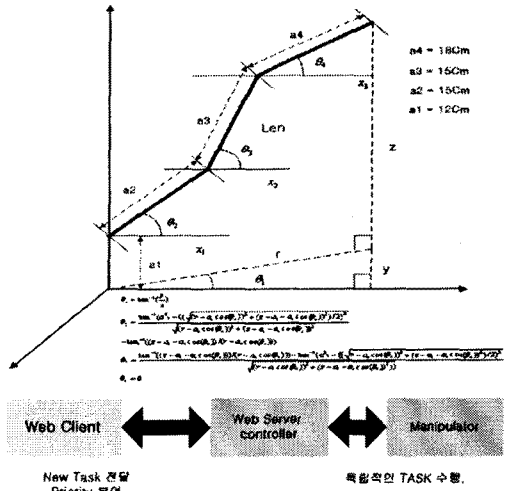


그림 13. 독립적인 태스크와 새로운 태스크  
Fig 13. Independent Task and New Task

#### 참 고 문 헌

- [1] 홍성수, 조석재, "내장형 웹서버 기술" 전기학회지 제 49권 10호, pp. 9-12, 2000년 10월.
- [2] P. Saucy, F. Mondada, "KhepOnTheWeb" : open access to a mobile robot on the Internet", IEEE Robotics & Automation Magazine, vol. 7 Issue:1, pp. 41-47, March 2000
- [3] A. Fryer, "Remote-control experiment using a networked robot", Rob. Mach. Perception, vol. 5, no. 1, p. 12, 1996
- [4] Rabbit Semiconductor, "Rabbit 2000 Microprocessor User's Manual", May 2000
- [5] M. F. Zakaria, S.H.M. Amin, R. Marmat. "Design and development of control system for Internet-based telerobotics". TENCON 2000. Proc., vol. 2. pp. 338-342, 2000
- [6] Peter Altenbernd: "Timing Analysis, Scheduling, and Allocation of Periodic hard Real-Time Tasks" Dissertation, Paderborn, 1996
- [7] W. Hardt, A. Rettberg, B. Kleinjohann. "The PARADISE design environment", 1st Embedded System Conference, Auckland(New Zealand), 1999