원격 제어를 위한 임베디드 통신 변환기 구현

이 병 권[†] · 전 영 숙[††] · 전 중 남[†††]

요 약

구형 산업용 계측 장비들은 대부분 직렬 통신 기능만 갖추고 있다. 이러한 장비들을 인터넷으로 연결하기 위한 임베디드 통신 변환기를 구현하였다. 이 장비는 입출력 장치로 한 개의 WAN 포트, 두 개의 LAN 포트, 두 개의 직렬 포트를 갖고 있으며, 소프트웨어에 의하여 직렬 통신과 네트워크 통신 간의 상호 변환 기능을 수행하고, 기존의 직렬-이더넷 변환기와 구별되는 웹 모니터링 기능을 지원한다.

ARM922T를 프로세서 코어를 사용하는 MICREL 사의 KS8695 네트워크 전용 칩셋을 사용하여 하드웨어를 구현하였고, 인터넷 기반 원격 제어를 위하여 웹 서버인 boa와 CGI 기능을 활용하였으며, 사설 IP 주소를 갖는 네트워크에서도 인터넷 접속이 가능하도록 IP 공유 기능을 추가하였으며, 통신 변환기의 역할을 수행하는 직렬-이더넷 변환 프로그램을 개발하였다. 그리고 개발된 통신 변환기를 태양열 에너지 전력 생산 시스템의 원격 감시 장치로 활용하는 예를 제시하였다.

키워드 : 임베디드 시스템, IP공유, 직렬이더넷 통신변환기, 원격제어

Implementation of An Embedded Communication Translator for Remote Control

Byung kwon Lee[†] · Young Suk Chon[††] · Joongnam Jeon[†††]

ABSTRACT

Almost of industrial measuring instruments usually are equipped only with serial communication devices. In order to connect these instruments to internet, we implement an embedded translator. This device has the hardware components composed of one WAN port, two LAN ports, and two UARTs, and functions as a communication translator between serial and internet communication. it also provides web-based monitoring function that is absent from existing serial-to-ethernet converter.

The hardware is implemented using the KS8695 network processor which has an ARM922T as processor core. We have installed the boa web server and utilized the CGI function for internet-based remote control, added the IP sharing function which allows the network with private IP addresses to access the internet, and developed a serial-to-ethernet translation program. Finally, we show an application example of the developed translator that remotely monitors the solar energy production system.

Key Words : Embedded System, IP Sharing, Serial-to-ethernet Translator, Remote Control

1. 서 론

컴퓨터 기술의 발전은 특수한 기능만 수행하는 임베디드 시스템에 커다란 변화를 가져왔다. 기존의 마이크로 제어기들은 지역적 감시 또는 제어 역할만 담당하였지만, 운영체제 탑재가 가능해짐으로써 인터넷 인프라를 이용한 통신이 가능해졌다[1]. 따라서 여러 지역에 산재되어 있는 산업용 제어 장비들을 인터넷을 통하여 중앙에서 원격 감시 또는 원격 제어하기 위한 환경은 이미 구축되어 있는 것이다[2]. 그렇지만 산업용 기기는 정보통신 기반 시설의 급속한 발전에 부응하지 못하고 있는 실정이다.

※ 이 논문은 2005학년도 충북대학교 학술연구지원사업의 연구비지원에 의하여 연구되었음(This work was supported by the research grant of the Chungbuk National University in 2005)
† 준 회 원 : 충북대학교 전자계산 대학원 박사과정 수료
†† 준 회 원 : 충북대학교 전자계산 대학원 이학박사
††† 정 회 원 : 충북대학교 전기전자컴퓨터공학부 교수
논문접수 : 2005년 8월 25일, 심사완료 : 2006년 5월 16일

인터넷을 통한 원격 제어를 위하여 산업용 제어 장비들을 인터넷 통신이 가능한 장비로 교체하여야 하지만 이것은 경제적인 부담이 크다. 구형 제어 장비들은 대부분 직렬 통신 기능을 갖추고 있으므로 직렬 통신 데이터를 인터넷을 통하여 전송하는 통신 변환기를 개발하여 활용하는 것이 더 경제적인 해결책이 될 수 있다.

본 논문에서는 구형 산업 장비를 원격으로 제어할 수 있도록 직렬 통신 장비를 인터넷 망으로 연결하는 임베디드 통신 변환기 개발에 관하여 설명한다. 여러 장비들이 하나의 변환기를 사용하여 인터넷 통신이 가능하도록 IP 공유 기능을 추가하였고, 기존의 장비는 단지 시리얼 데이터를 이더넷으로 변환 시켜주는 역할 밖에 하지 않는 단점을 보안하기위해 원격 제어 및 시스템 모니터링이 가능하도록 임베디드 웹 서버 및 통신 변환 프로그램을 작성하였다. 또한 통신망 장애에 대처하기 위하여 변환기에 비휘발성 SRAM

를 장착하여 데이터 손실을 방지하도록 설계하였다. 이 통신 중계기를 사용함으로써 적은 비용으로 구형 산업용 장비의 원격 제어 및 재활용이 가능해진다.

본 논문의 구성은 다음과 같다. 2장에서는 변환기 개발에 관련된 기술들을 설명하고, 3장에서 임베디드 통신 변환기의 구조 및 구현에 대하여 자세히 설명한다. 4장에서 구현 결과를 제시하고, 마지막 5장에서 결론을 맺는다.

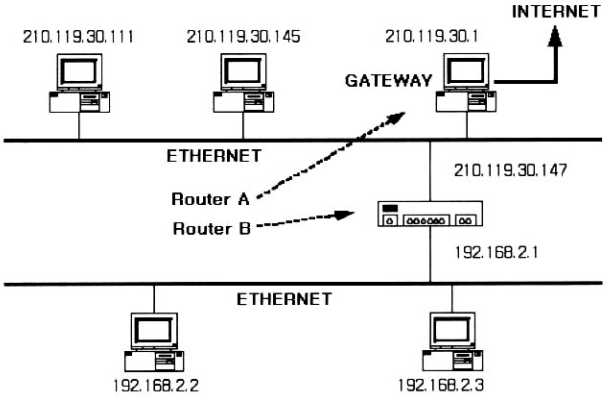
2. 관련 기술

여기에서는 임베디드 통신 변환기 구현과 관련된 IP 공유의 원리와 원격 제어에 필수적인 요소인 웹 서버와 CGI 표준 인터페이스에 대하여 기술한다.

2.1 IP 마스커레이드(IP masquerade)

IP 마스커레이드 기법은 네트워크 특별한 기능으로, 상용 방화벽(firewall)이나 네트워크 라우터에서 볼 수 있는 일대다(one-to-many) 방식의 NAT(Network Address Translation : 네트워크 주소 해석)를 이용한 IP 공유 방식이다[3]. 마스커레이드 기법을 사용하면 게이트웨이를 통해서 들어오는 WAN을 내부의 LAN으로 연결하여 여러 대의 컴퓨터가 IP를 공유하여 인터넷을 사용할 수 있다. 이러한 IP 마스커레이드 기법은 방화벽 원리로 이용되며 대단히 안전한 네트워크 환경을 제공한다.

(그림 1)는 IP를 공유하기 위한 마스커레이드 기법을 도식화한 것이다. 그림 1에서 210.119.30.1의 서버 컴퓨터에 있는 라우터 A에서 인터넷 정보를 IP 210.119.30.147을 통하여 라우터 B에 전달(forwarding)하면 라우터 B에 연결되어 있는 여러 대의 컴퓨터들은 IP 210.119.30.147을 공유해서 인터넷 사용이 가능해진다. 라우터 B에 연결되어 있는 컴퓨터들은 공인 IP가 아닌 사설 IP(10.0.0.1~10.255.255.255/172.16.0.0~172.31.255.255/192.168.0.0~192.168.255.255)가 할당되어져야 한다. 그림 1의 하단 컴퓨터들은 192.168.2.1를 게이트웨이로 사용하여 내부 네트워크가 구성된 것이다.

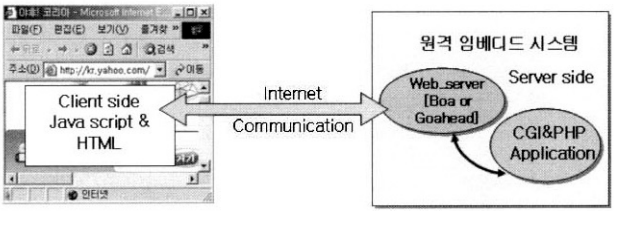


(그림 1) IP 마스커레이드

2.2 웹 서버와 CGI 인터페이스

웹 서버는 인터넷을 통하여 원격 사이트 또는 장비와 통신하는 기능을 제공한다. 임베디드 시스템에 웹 서버를 설치하여 서버로서 동작하게 하고, 원격 제어를 위한 클라이언트로 웹 브라우저를 사용한다. 서버/클라이언트 통신 프로토콜인 HTTP(HyperText Transport Protocol)를 사용하여 원격에서 웹 서버가 설치된 장비를 제어할 수 있다[4].

가장 많이 사용하는 임베디드 시스템의 웹 서버로 boa와 goahead가 있다. 특히 boa 서버는 C 언어로 작성되어 있기 때문에 소스 코드를 변경하여 시스템의 특성에 맞추어 변경하기 쉽다[5]. 웹 서버는 단순히 요청한 자료를 전달하는 역할 밖에 수행하지 못하는 단점이 있어 프로그램 제어나 반복 처리와 같은 다양한 처리가 가능하도록 표준 인터페이스 규약이 나오게 되었다. 표준 인터페이스로 서버측(server side) 스크립트와 클라이언트측(client side) 스크립트가 있다. 서버측 스크립트로 CGI(Common Gateway Interface), PHP 등이 있고 클라이언트측 스크립트는 JavaScript가 대표적이다. 서버측 스크립트 언어는 HTML 정보의 단방향적인 단점을 극복하여 웹 사이트를 이용하는 사용자로부터 정보를 받을 수 있도록 한 양방향적인 새로운 규약이다. (그림 2)는 임베디드 시스템에 설치되어 있는 웹 서버와 서버측 스크립트 간의 동작을 보여주고 있다. 원격의 웹 브라우저에서 HTML 태그와 클라이언트측 스크립트를 통하여 데이터를 웹 서버로 보내고 웹 서버는 서버측 스크립트를 통하여 데이터를 수신하고 서버에 설치되어 있는 어플리케이션으로 전달한다.



(그림 2) 웹 서버를 이용한 원격 제어

CGI는 사용자가 웹 서버에 접속하면 웹서버가 CGI 프로그램을 호출하고 CGI 프로그램은 그 실행 결과를 다시 웹 서버에 전달하여 사용자가 받아 보는 방식을 사용한다[6]. CGI는 특별히 어떠한 언어로 제작해야 한다는 규칙은 없으며 일정한 웹 형식에 맞추어 C, C++, Perl 등과 같은 언어로 제작할 수 있다. CGI의 장점은 서버의 자원을 모두 활용할 수 있는 점과 기존의 특정 언어를 알고 있다면 별도로 배울 필요가 없다는 것이다.

3. 임베디드 통신 변환기

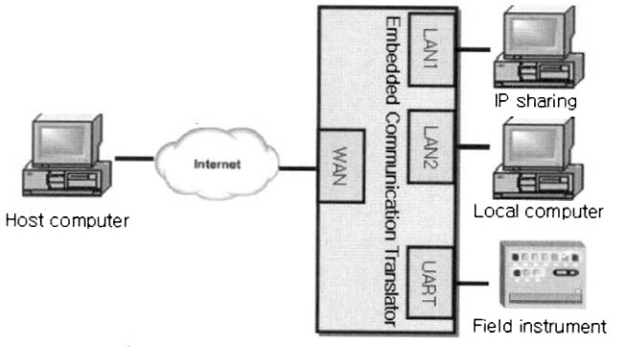
이 장에서는 원격 제어를 위한 임베디드 통신 변환기의 구조 및 구현 과정에 대하여 자세히 설명한다. 통신 변환기의 기능과 하드웨어 구조 및 소프트웨어 플랫폼을 설명하고,

직렬 통신을 인터넷 통신으로 중계하기 위한 중계 소프트웨어, IP 공유, 그리고 원격 제어 기능을 부여하기 위한 웹 서버 설치 및 CGI 프로그램에 대하여 설명한다.

3.1 임베디드 통신 변환기의 기능

임베디드 통신 변환기는 웹을 통하여 원격에서 산업용 장비를 제어하는 장치이다. 산업용 장비를 인터넷을 통하여 원격 제어하는 개념은 (그림 3)과 같다. 통신 변환기는 IP 공유, serial-to-ethernet 변환, 그리고 원격 제어 기능을 갖고 있다.

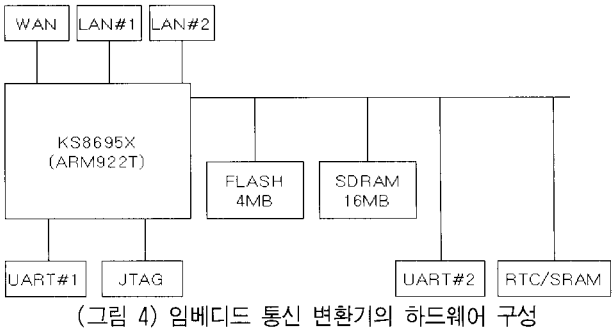
- serial-to-ethernet 변환 : 산업용 장비(field instrument)들 변환기의 직렬 포트에 연결하고, 변환기 내부의 변환 프로그램에 의하여 직렬로 수신한 데이터를 인터넷을 통하여 원격 컴퓨터로 전달하거나 인터넷에서 수신한 데이터를 직렬 포트로 전달한다. 이 부분은 3.3절에서 설명한다.
- IP 공유 : 리눅스에 포함되어 있는 IP 마스커레이드 기능을 사용하여 하나의 WAN 포트를 공유하여 두 개의 LAN 포트에 연결된 지역 컴퓨터들이 인터넷을 사용할 수 있다. 이 기능은 3.4절에서 설명한다.
- 원격 제어 : 웹 서버인 boa와 CGI를 사용하여 통신 변환기를 원격으로 제어한다. 이 부분은 3.5절, 3.6절에서 설명한다.



(그림 3) 임베디드 통신 변환기의 운영

3.2 하드웨어 구조 및 소프트웨어 플랫폼

(그림 4)는 임베디드 통신 변환기의 하드웨어 구조도이다. 주 프로세서인 MICREL사의 KS8695X는 ARM922T를 프로세서 코어로 사용하는 네트워크 전용 임베디드 칩셋이다[7]. 이 칩셋은 한 개의 WAN 포트, 네 개의 LAN 포트, 한 개의 UART, 그리고 JTAG 인터페이스를 제공한다. 구현에는 두 개의 LAN 포트만 사용하였으며, 외부에 4MB의 FLASH와 16MB의 SDRAM을 부착하였다. 칩셋이 제공하는 UART(UART#1)는 콘솔용으로 사용하고, 직렬 통신 장치를 연결하기 위하여 UART#2를 추가하였다. UART#2는 IBM PC/AT 및 PS/2와 호환인 TG16C550M[8]을 사용하였다. 그리고, 시간 추적을 위하여 RTC(Real-Time Clock)와 데이터 버퍼링을 위하여 비휘발성 SRAM을 추가하였다. 사용한 RTC는 M48T37[9]로 32KB의 낮은 전력으로 동작하는



(그림 4) 임베디드 통신 변환기의 하드웨어 구성

SRAM, Real-Time Clock, Power-Fail Circuit, 그리고 자체 건전지를 포함하고 있다. 또한 실시간으로 시스템의 동작 여부를 판단할 수 있는 WatchDog Timer를 포함하고 있다. RTC는 자체 건전지를 가지고 있어 원격에 있는 장비가 전원이 제거되더라도 데이터를 비휘발성 SRAM에 보관할 수 있다.

(그림 4)의 하드웨어에 임베디드 통신 변환기를 구현하기 위하여 임베디드 리눅스 기반 운영체제를 포팅하였다. 리눅스 커널은 버전 2.4.19를 사용하였고, 루트 파일시스템 내에 원격 제어를 위한 Web-CGI 모듈, IP 공유 모듈, 임베디드 통신 변환 모듈, RTC 제어 모듈을 추가하였다. 다음절에서 각각의 모듈에 대한 설명을 한다.

3.3 장치 포팅

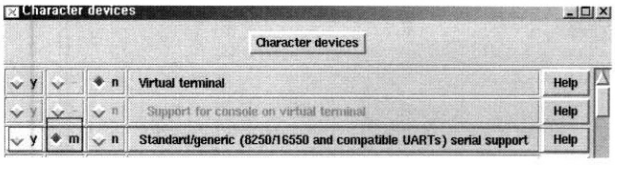
MICREL사는 KS8695X와 함께 임베디드 리눅스 BSP(Board Support Package)를 제공한다. 여기에는 KS8695에 내장된 장치들(WAN, LAN, UART#1)이 포팅되어 있으며 이 장치들에 대한 디바이스 드라이버들도 함께 제공한다. 그러므로, 별도로 부착한 UART#2와 RTC를 운영체제에 등록하고 디바이스 드라이버를 포팅할 필요가 있다.

장치를 등록하기 위하여 리눅스 커널 소스를 수정하여 UART#2와 RTC의 물리주소를 운영체제가 사용하는 가상 주소로 리맵(remap)하는 작업을 수행하여야 한다. 이를 위하여 (그림 5)와 같이 커널 소스 linux/arch/arm/mach-ks8695.c를 수정한다[10].

RTC로 사용한 M48T37은 주소 매핑 후 메모리 맵으로 연결되는 입출력 포트로 직접 액세스 된다. 데이터시트[11]

```
#define KS8695_RTC_BASE 0x01000000
#define KS8695_UART_BASE 0x03200000
static struct map_desc ks8695_io_desc[] __initdata = {
{IO_ADDRESS(KS8695_PCMCIA_IO_BASE),
  KS8695_PCMCIA_IO_BASE, 0x30000 * 4, DOMAIN_IO, 0, 1},
{ IO_ADDRESS(KS8695_IO_BASE), KS8695_IO_BASE,
  SZ_64K, DOMAIN_IO, 0, 1, 0, 0 },
{ IO_ADDRESS(KS8695_RTC_BASE),
  KS8695_RTC_BASE, 0x10000, DOMAIN_IO, 1, 1, 0, 0 },
{ IO_ADDRESS( KS8695_UART_BASE),
  KS8695_UART_BASE, 0x10000, DOMAIN_IO, 1, 1, 0, 0 },
LAST_DESC  };
```

(그림 5) 물리주소와 가상주소의 매핑



(그림 6) UART 커널 설정

을 참고하여 일자 및 시각을 설정하는 문자 장치 드라이버(character device driver)[12]를 개발하여 모듈로 등록하였다.

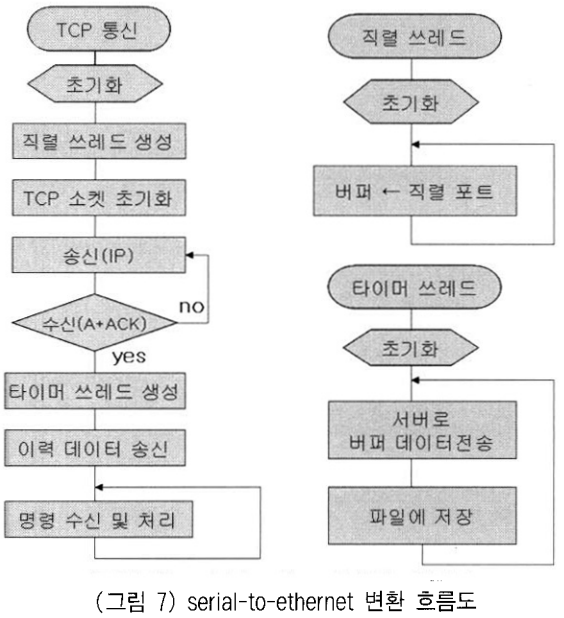
새로 추가된 UART#2를 구동하기 위하여 (그림 6)과 같이 먼저 커널 설정에서 UART 드라이버를 설정하여야 한다. 커널 옵션 Standard/generic serial support을 선택하면, /linux/kernel/driver/char/에 8250/16550용 직렬 통신 디바이스 드라이버 모듈 serial.o가 생성된다. 드라이버는 모듈 형태로 컴파일 되어 있어 리눅스가 부팅된 후 드라이버 추가 명령어인 insmod를 사용하여 커널에 등록한다.

마지막으로 RTC와 UART#2에 관련된 장치 노드를 만들어야 한다. 리눅스의 명령어 mknod를 사용하여 루트파일 시스템의 /dev/ttyS0와 /dev/rtc를 생성한다.

3.4 Serial-to-ethernet 변환 프로그램

기본적인 하드웨어와 운영체제 포팅을 완료하고 serial-to-ethernet 변환 프로그램을 개발하였다[13]. 이 프로그램은 직렬 통신으로 수신한 데이터를 이더넷 통신인 TCP/IP 패킷으로 변환하여 전달하고 반대 방향으로도 데이터를 전달하는 기능을 갖도록 설계하였다. 따라서, 인터넷을 통하여 직렬 통신 기능만 갖춘 산업용 장비에서 데이터를 수집하거나 산업용 장비를 제어할 수 있게 된다.

(그림 7)은 임베디드 통신 데이터 전달 과정을 흐름도이며, 메인 프로그램인 TCP 통신, 직렬 쓰레드, 그리고 타이머 쓰레드로 구성되어 있다. 메인 프로그램은 초기화 후 직렬 쓰레드를 생성하고, TCP 소켓을 통하여 서버의 연결 여부를 확인한 후 타이머 쓰레드를 생성한 후 비휘발성 SRAM에 저장되어 있을 수 있는 데이터를 송신한다. 그 다음에 서버에서 전달되는 명령을 처리한다. 직렬 쓰레드는 직렬 포트에서 입력되는 데이터를 비휘발성 SRAM에 저장하고, 타이머 쓰레드는 일정 시간 간격으로 SRAM에 저장되어 있는 데이터를 서버로 전송하고 동시에 파일에 저장한다.



(그림 7) serial-to-ethernet 변환 흐름도

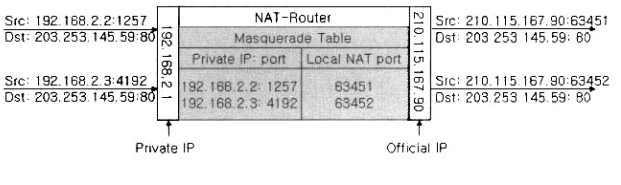
전송되는 데이터는 [기능코드 + 데이터]로 구성되어 있다. 기능 코드는 한 개의 문자이고 데이터 부분은 기능코드에 따라 의미가 결정된다. 다음과 같이 10 개의 기능코드를 정의하였다.

- Function "R" : 실시간(Realtime) 데이터 전송
- Function "H" : 이력(Historical) 데이터 전송
- Function "T" : 시스템 시각(Time) 동기화
- Function "A" : 원격 서버의 통신 상태(Alive) 점검
- Function "W" : 사이트 ID 확인(Who)
- Function "S" : 사이트 ID 설정(Setup)
- Function "I" : 데이터 전송 주기(Interval) 설정
- Function "F" : 데이터 저장 주기 (Flash data saving interval) 설정
- Function "M" : MAC Address 설정
- Function "C" : 데이터 이력 삭제(Clear)

예를 들어, 실시간 데이터 전송인 경우 임베디드 통신 변환기는 서버로 패킷구조 [R + 날짜 + 시간 + 사이트 ID(IP 주소) + DATA]를 전송한다. 서버는 이에 대하여 [R+ACK] 또는 [R+NAK]으로 응답한다.

3.5 IP 공유

IP 공유는 공인 IP(official IP)를 사설 IP(private IP)로 전달하여 사설 IP를 사용하는 컴퓨터를 인터넷에 연결하는 방법이다. IP 공유 기술은 리눅스 시스템이 제공하는 IP 마스커레이드 기법을 이용하고, 이와 같은 기술의 구현을 위해 두 개의 이더넷 장치가 필요하다. 첫 번째 이더넷 장치는 인터넷이 가능하도록 인터넷 패킷을 받아들이는 역할을 하고, 두 번째 이더넷 장치는 인터넷 패킷을 내부 네트워크(사설 IP로 구성된 망)로 전달한다. 리눅스 커널 버전 2.4 이후에는 기존의 공유 방법인 ipfwadm 명령을 사용하지 않고 iptables 명령을 이용하여 마스커레이드 테이블을 설정함으로써 IP를 공유를 구현한다. 마스커레이드 테이블은 사설 IP를 통신 포트(Local NAT port)로 매핑하는 기능을 수행한다.



(그림 8) 마스커레이드 테이블

(그림 8)은 두 개의 사설 IP 192.168.2.2와 192.168.2.3이 공인 IP 210.115.167.90을 공유하도록 마스커레이드 테이블을 구성한 예이다. 예를 들어, 192.168.2.2가 WAN에 연결되어 있는 외부의 203.253.145.59로 접속하려는 경우 매핑 테이블을 참고하여 210.115.167.90에서 203.253.145.59로 접속하는 형태로 IP 주소를 변환한다. (그림 8)과 같은 임베디드 리눅스 시스템에 IP 공유를 구현하기 위하여 마스커레이드 테이블을 생성하도록 커널을 설정하고 iptables 명령을 사용하여 마스커레이드 테이블 내부에 IP 주소를 포트로 매핑하는 과정을 거쳐야 한다. 이 과정은 다음과 같다.

3.5.1 커널 설정(Kernel Configuration)

IP 공유 기능을 구현하는 첫 번째 단계는 커널 옵션 설정이다[14]. 커널의 네트워킹 옵션(Networking options)에서 IP: Netfilter Configuration을 선택한 후 기본적으로 세팅되어 있는 항목 이외에 (그림 9)의 항목들을 YES로 설정한다.

(그림 9)와같이 네트워크 설정이 완료 되면 IP공유 옵션을 포함한 커널 이미지를 타겟 보드에 포팅한 후 네트워크 공유 작업을 수행한다.

- Prompt for development and/or incomplete code/drivers - YES
- Enable loadable module support - YES
- Networking support - YES
- Packet socket - YES
- Kernel/User netlink socket - YES
- Routing messages - YES
- Network Packet filtering (replaces ipchains) - YES
- TCP/IP Networking - YES
- IP: advanced router - YES
- IP: verbose route monitoring - YES
- IP: GRE tunnels over IP - YES
- IP: TCP Explicit Congestion Notification support - YES
- Connection tracking (required for masq/NAT) -YES
- IP tables support (required for filtering/masq/NAT) -YES
- Full NAT - YES
- MASQUERADE target support - YES
- REDIRECT target support - YES
- IP: TCP syncookie support - YES
- Network device support - YES
- Dummy net driver support - YES
- /proc filesystem support - YES

(그림 9) IP 공유를 위한 커널 설정

3.5.2 네트워크 공유 동작 설정

네트워크 공유 작업을 위해 두 개의 이더넷 장치(eth0와 eth1)가 필요하다. 명령(1)과 명령(2)를 사용하여 eth0에 공인 IP 210.115.167.90을 할당하고 eth1에 사설 IP 192.168.2.1을 할당한다. 이 사설 IP는 내부 네트워크에서 사용할 게이트웨이 역할을 담당한다.

```
#>ifconfig eth0 210.115.167.90 up          (1)
#>ifconfig eth1 192.168.2.1 up             (2)
```

다음에 iptables 명령을 사용하여 공인 IP을 마스커레이드 테이블에 등록하고(명령(3)), route 명령을 사용하여 공인 IP에 대한 게이트웨이(게이트웨이 주소를 210.115.167.1로 가정함)를 설정한다(명령(4)).

```
#>iptables -t nat -A POSTROUTING -o eth0
        -j SNAT --to 210.115.167.90        (3)
#>route add default gw 210.115.167.1 eth0   (4)
#>echo 1 > /proc/sys/net/ipv4/ip_forward    (5)
```

마지막으로 명령(5)로 커널에 IP 패킷을 외부 네트워크로 전달하라고 지시한다. 이후, 내부 네트워크에 연결된 사설 IP 장비가 외부 네트워크로 연결하려고 접속할 때 커널은 사설 IP들에 대한 마스커레이드 테이블의 매핑 값을 등록하여 IP 공유가 시작된다.

3.6 웹 서버 설치

원격에서 임베디드 시스템을 제어하기 위하여 임베디드 시스템에 웹 서버를 설치하고 CGI 프로그램을 통하여 외부의 제어를 임베디드 시스템으로 전달하여야 한다. 임베디드 시스템의 웹 서버로 사용한 boa는 PC 호환 리눅스에서 개발되었지만 어떤 플랫폼에도 사용 가능하다는 장점을 가지고 있다. 임베디드 시스템에 boa 서버를 포팅하는 과정은 다음과 같다.

- 1 단계 : boa-0.92.tar.gz를 다운로드받아 압축을 해제한다.
- 2 단계 : config.c와 lex.yy.c을 열어 사용하는 환경 파일이 설치될 디렉토리 경로를 수정한다.

```
yyin=fopen("/etc/boa/boa.conf","r");
        // in config.c                     (6)
yyin=fopen("/etc/boa/mime.types","r");
        // in lexyy.c                      (7)
```

- 3 단계 : 웹 서버 환경 설정 파일인 boa.conf 파일을 시스템 운영에 맞도록 (그림 10)과 같이 수정한다.
- 4 단계 : 환경 설정 작업이 완료되면 웹 서버 boa 데몬을 임베디드 시스템의 /usr/bin에 복사하고, boa.conf 파일은 타겟의 /etc/boa/에 복사하고, 임베디드 시스템

```
# Port: The port Boa runs on. The default port for http servers is 80.
# If it is less than 1024, the server must be started as root.
Port 80
# DocumentRoot: The root directory of the HTML documents.
DocumentRoot /etc/boa/web/docs
# Uncomment the next line if you want .cgi files to execute from anywhere
AddType application/x-httpd-cgi cgi
# ScriptAlias: Maps a virtual path to a directory for serving scripts
# Example: ScriptAlias /bin/ /wweb/bin/
ScriptAlias /cgi-bin/ /etc/boa/web/cgi/
```

(그림 10) boa 환경 설정

의 루트 파일시스템에 /web/cgi와 /web/docs를 생성한다. /web/cgi 디렉토리는 웹 서버에서 동작할 cgi 프로그램이 설치될 장소이고, /web/docs 디렉토리는 HTML 문서의 루트 디렉토리이다.

- 5 단계 : 마지막으로 임베디드 시스템에서 명령(8)에 의하여 boa 웹 서버를 백그라운드로 동작시킨다.

```
#>boa & > /dev/null 2>&1                    (8)
```

이와 같은 과정을 거쳐 임베디드 시스템에 boa 웹 서버의 설치가 완료된다. 다음 절에서는 웹 서버를 통하여 임베디드 시스템을 제어하는 방법에 대하여 설명한다.

3.7 CGI를 통한 임베디드 시스템 제어

개발된 임베디드 시스템을 원격에서 제어하기 위하여 웹 서버 기능을 사용한다. 서버측 스크립트인 CGI 프로그램을 사용하여 임베디드 시스템 자체의 환경 설정, 시스템 관리를 위한 리눅스 명령어 실행, 장치 제어 프로그램의 환경 설정 및 시작 및 종료 등의 기능을 구현할 수 있다[15].

제어를 위한 명령 또는 데이터를 임베디드 시스템 장치에 전달하기 위한 방법으로 HTML 문서를 이용하고, HTML 문서 내에 CGI 구동 링크를 작성한다. (그림 11)은 웹 화면에서 ID와 PW를 입력받아 CGI 모듈인 remote.cgi를 구동하는 간단한 HTML 문서의 예이다.

클라이언트측에서 서버측 프로그램으로 데이터전달 방법으로 GET과 POST 두 가지가 있다. GET방법은 정보를 환경 변수인 Query String을 통해 전달하는데 반해 POST 방법은 표준 입출력을 통해 전달한다. 환경 변수의 크기에 따라 크기가 제한되는 GET 방법과 달리 POST는 전달할 수 있는 정보의 크기에 제한을 받지 않는다. CGI 프로그램 remote.c(그림 12)의 라인 22, 23에서 POST 방법으로 전달된 ID와 PW를 추출한다. 이 프로그램을 구동하기 위하여 ARM 크로스 컴파일러인 arm-linux-gcc을 사용하여 명령(9)로 컴파일 한 후, 실행 이미지인 remote.cgi 파일을 임베디드 시스템의 /web/cgi-bin 디렉토리에 저장한다.

```
#>arm-linux-gcc -o remote.cgi remote.c util.c    (9)
```

지금까지 ID와 PW를 임베디드 시스템으로 전달하여 CGI 프로그램을 구동하는 간단한 예를 제시하였다. 동일한 방법으로 필요한 만큼의 파라미터들을 임베디드 시스템으로 전달할 수 있고, 전달된 파라미터들을 처리하도록 프로그램함으로써 다양한 방법으로 시스템을 제어할 수 있게 된다.

```
<html><head><title>CGI Test</title></head><body>
<form method=post action="/cgi-bin/remote.cgi">
<table><td>
<p align="center">CGI  Web </p>
<p> ID : <input type="text" name="ID"></p>
<p> PW: <input type="text" name="PW"></p>
<p><input type=submit value="저장"></p>
</td></tr></table></form> </body></html>
```

(그림 11) HTML 문서와 CGI.

```
1.  // 컨텐츠 타입이 URL 인코딩된 데이터임을 체크한다.
2.  if (strcmp(getenv("CONTENT_TYPE"),
        "application/x-www-form-urlencoded")) {
3.      printf("This script decode form results. \n");
4.      exit(1);  }
5.  // 데이터를 디코딩하여 변수명과 변수 값을 추출한다.
6.  cl = atoi(getenv("CONTENT_LENGTH"));
7.  for (x=0; cl && (!feof(stdin));x++) {
8.      m = x;
9.    entries[x].val = fmakeword(stdin, '&', &cl);
10.     plustospace(entries[x].val);
11.     unescape_url(entries[x].val);
12.     entries[x].name = makeword(entries[x].val, '=');
13.  }
14.  strcpy( ginfo.ID , "default" );
15.  strcpy( ginfo.PW , "default" );
16.  ginfo.TYPE = 4;
17.  // 입력된 내용중에 빈것은 없는지 체크하고 읽은 데이터
       를 ginfo 구조체에 저장한다.
18.  for (i=0; i<1; i++) {
19.    if (strcmp(entries[i].name, "ID") == 0) {
20.        ginfo.TYPE = 1;
21.        sprintf(ginfo.ID, "%s", entries[i].val);
22.        sprintf(ginfo.PW , "%s", entries[i+1].val);   } }
23.  // 파일 이름은 폼 문에서 넘긴 히든 타입의 ID에서 읽어
       들인다.
24.  sprintf(datadir, "/jffs2/view.in");
25.  fp = fopen(datadir, "w+");
26.  flock(fileno(fp), LOCK_EX);
```

(그림 12) remote.c 소스 프로그램

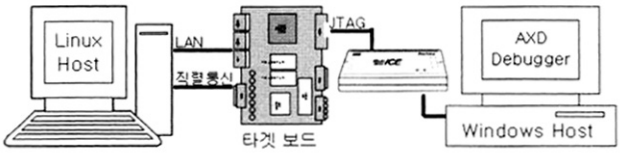
4. 구 현

임베디드 통신 변환기를 구현하기 위하여 참고용 보드(reference board)로 Micrel 사의 KS8695 Demo Board revision 1.0을 사용하였다. 데모용 보드인 KS8695는 임베디드 리눅스 기반의 라우터 기능을 제공하며, 이 보드를 기반으로 하드웨어 및 소프트웨어 기능을 추가하여 목표 시스템을 구현하였다. 이 절에서는 개발 환경, 구현 결과, 그리고 응용 사례에 대하여 기술한다.

4.1 개발 환경

KS8695 Demo Board revision 1.0는 다음과 같은 기능을 제공한다.

- KS8695를 사용한 하드웨어 : 입출력 장치로 한 개의 WAN 포트, 네 개의 LAN 포트, 한 개의 콘솔용 직렬 포트를 갖추고 있으며, 메모리는 SDRAM 16 MB와 플래시 메모리 4 MB가 장착되어 있다. 그리고 플래시 메모리에 이미지를 기록하기 위한 인터페이스로 JTAG 포트를 갖추고 있다.



(그림 13) 개발 환경

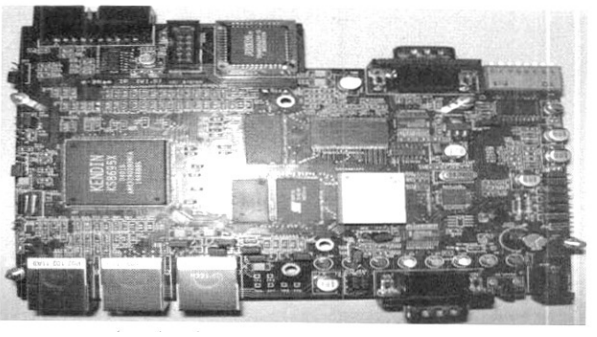
- KS8695 전용 부트로더(bootloader) : KS8695 칩셋에 포함되어 있는 레지스터들을 초기화 하고 커널을 적재하는 기능을 갖춘 전용 부트로더이다. 이 부트로더는 일반적인 ARM 프로세서용 부트로더인 blob이나 U-boot와는 다르게 부트로더 명령을 제공하지 않는다.
- 리눅스 커널 : 데모용 보드에 맞추어진 리눅스 소스를 함께 제공한다.
- 루트 파일시스템 : 하드웨어를 구동하기 위한 기본적인 드라이버 및 리눅스 명령을 포함하고 있다.

그리고 소프트웨어 디버그 및 JTAG을 통한 플래시 메모리 기록을 위하여 ARM 사에서 제공하는 Multi-ICE를 사용하였다. Multi-ICE는 EmbeddedICE 로직을 포함하고 있으며 ARM 프로세서 코어에서 실행되는 프로그램을 디버그하기 위한 하드웨어 장치이다. Multi-ICE를 구동하기 위한 소프트웨어로 Multi-ICE 서버와 ARM AXD(ARM eXtended Debugger)를 함께 제공한다. 이 소프트웨어는 Windows 환경에서 실행된다.

(그림 13)은 개발 환경을 보여준다. 타겟 보드는 리눅스 호스트와 직렬 통신 및 LAN을 통하여 연결된다. 직렬 통신은 타겟의 콘솔 연결을 위하여 사용하였고, LAN은 NFS(Network File System) 기능을 사용하기 위하여 리눅스 호스트와 연결하였다. Windows 호스트는 Multi-ICE를 통하여 타겟의 JTAG 포트로 연결된다.

4.2 구현 결과

(그림 14)은 개발된 임베디드 통신 변환기의 사진이고, <표 1>은 KS8695 Demo 보드와 하드웨어 사양을 비교한 것이다. 기존 보드에서 LAN 포트 두 개를 제거하고, UART 한 개를 추가하였다. 그리고 기존 보드에는 없는 RTC와 DIP 스위치를 추가하였다. DIP 스위치는 시스템 부팅 후 DIP 스위치 값에 따라 특정한 동작을 수행하도록 지시하는 용도로 사용할 수 있다. 그리고 가격을 고려하여 플래시 메모리를 용량이 2MB인 SST39VF160 두 개로 교환하였다.



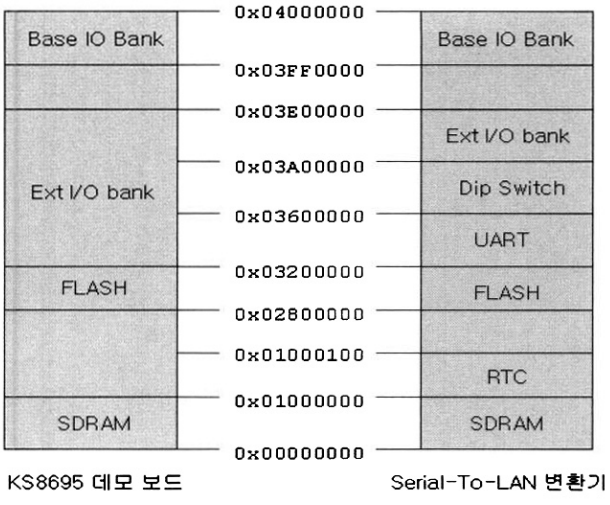
(그림 14) 임베디드 통신 변환기 보드

KS8695 Demo 보드에서 제공하는 부트로더와 리눅스 커널을 기반으로 다음과 같이 소프트웨어 기능을 추가하였다.

- 부트로더 : 새로 추가된 장치를 포팅하기 위하여 KS8695 매뉴얼[16]을 참고하여 레지스터들을 설정하였고, 커널 및 램 디스크의 크기가 증가함에 따라 리눅스 커널이 적재될 메모리 영역의 크기를 변경하였다. (그림 15)에 KS8695 Demo 보드와 구현된 임베디드 통신 변환기의 메모리 맵을 제시하였다.
- 커널 : 3.5절에서 설명한 바와 같이 IP 공유 기능을 제공하는 모듈을 추가하고, 직렬 포트에 대한 디바이스 드라이버 모듈을 활성화 하였다. 또한, 사용자 데이터 및 프로그램을 플래시 메모리에 저장하기 위하여 JFFS2 파일 시스템을 설치하였다.
- 루트 파일시스템 : KS8695 Demo 보드와 함께 제공되는 루트 파일시스템은 NFS 기능을 제공하지 않기 때문에, 임베디드 리눅스 용 명령어 툴 킷인 Busybox 0.60.3를 NFS 기능을 지원하도록 갱신하였다. 그리고, IP 공유를 위하여 iptables-1.2.8을 컴파일하여 실행 이미지와 관련

〈표 1〉 보드 사양 비교

구성 요소	KS8695 Demo 보드	임베디드 통신 변환기
CPU	ARM922T	ARM922T
WAN	1 port	1 port
LAN	4 port	2 port
SDRAM	MT48LC4M32B2: 16M(32bit)	MT48LC4M32B2: 16MB(32bit)
Flash ROM	AM29LV033C: 4M(8bit)	SST39VF160: 2M(16bit) 2개
UART	AMBA Serial	AMBA Serial
		TG16C550M(16550)
RTC/SRAM	없음	M48T37V (SRAM 256 KBit)
Dip Switch	없음	8 bit



(그림 15) 메모리 맵 비교

라이브러리를 루트 파일시스템에 추가하였고, 웹 서버로 동작하기 위한 boa 및 serial-to-ehternet 응용 프로그램도 루트 파일시스템에 추가하였다.

- 유저 파일시스템 : 루트 파일시스템은 휘발성 RAMDISK로 구현되어, RAMDISK의 내용은 리셋 후 수정된 내용이 소멸된다. 원격에서 제어하기 위한 시스템 환경 설정 파일은 상황에 따라 그 내용이 변경되기 때문에 비휘발성 메모리에 저장될 필요가 있다. 이를 위하여 기록 가능한 파일시스템인 JFFS2를 구현하여 유저 파일시스템으로 플래시 메모리 영역에 등록하였다. 이 영역은 환경 설정 정보와 응용 프로그램이 생성하는 데이터를 영구 보관하는 목적으로 사용된다.

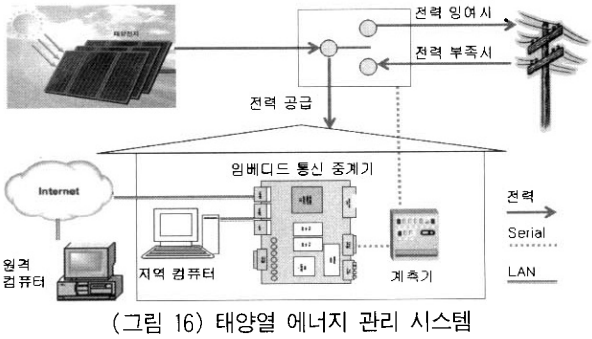
개발된 임베디드 통신 변환기와 기존의 공유 장치, serial-to-ethernet 전용 제품을 선정하여 비교하였다. <표 2>는 국내에서 판매되는 IP 공유 장비의 대표 모델로 Netmate사의 nmx-wb04g[17]와 MOXA Ltd. serial-to-ethernet 브릿지인 uc7410(moxa Ltd.)[18]를 비교한 결과이다. nmx-wb04g는 유무선 공유가 가능하지만 직렬포트가 존재하지 않아 serial-to-ethernet 변환이 불가능하다. uc7410은 serial-to-ethernet 변환기로 개발된 제품으로 단지 직렬 데이터를 받아 이더넷으로 포워딩하는 역할을 하도록 개발되었지만, IP 공유 기능이 없어 다른 장비들은 인터넷을 사용할 수 없고, 웹 서버 및 CGI를 포함하고 있지 않아 웹 모니터링이 불가능하다. 이에 반해 개발된 임베디드 통신변환기는 IP 공유 가 가능하고 web 서버와 CGI를 설치하여 원격에서 직렬 통신 장비를 감시 및 제어 가능할 수 있도록 개발되었다.

4.4 응용 사례

개발된 임베디드 통신 변환기를 태양열 에너지 관리 시스템에 적용하였다. 태양열 발전 시스템은 전기 에너지를 생성하여 전력이 건물에 공급하는데, 발전된 전력이 건물에서 사용하고 남을 때는 전력회사로 송전하고 부족할 때는 전력회사에서 에너지를 공급한다.

(그림 16)은 태양열 에너지 관리 시스템의 구성도이다. 계측기는 태양열 발전 시스템의 데이터를 수집하여 직렬 통신에 의하여 임베디드 통신 변환기로 데이터를 송신한다. 이 데이터는 변환 프로그램에 의하여 원격지에 있는 중앙 감시 컴퓨터로 전력생산 현황을 전송한다. 중앙의 컴퓨터는 웹 서버와 CGI 기능을 이용하여 임베디드 통신 변환기의 환경을 설정한다. 또한 IP 공유 기능을 활용하여 지역 컴퓨터도 인터넷을 사용할 수 있다. 이와 같이 인터넷 통신 기능이 없는 장비에게 인터넷 통신 기능을 부여함으로써 원격에서 감시 및 제어할 수 있다.

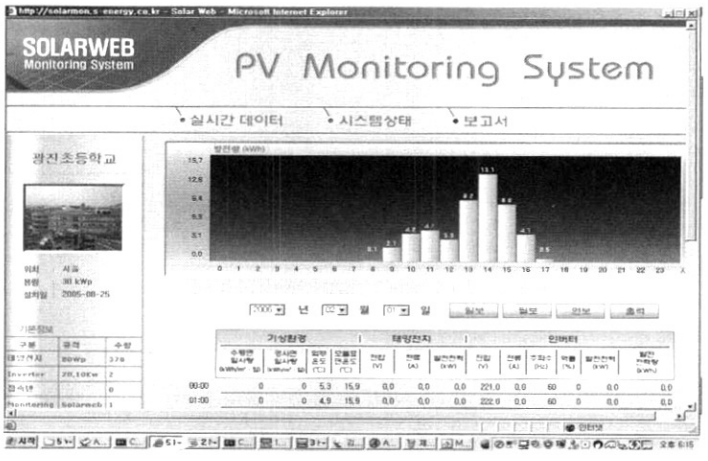
(그림 17)은 개발된 임베디드 통신 변환기를 통하여 수집된 정보를 변환기 차체에서 구동되는 웹서버(boa)를 통하여 전력생산량 대한 보고서이다. (그림 16)에 있는 계측기를 통하여 수집된 정보(기상현황, 전압, 전력.etc)는 웹 모니터링



(그림 16) 태양열 에너지 관리 시스템

〈표 2〉 장치 기능 비교

비교 항목	Netmate nmx-wb04g	MOXA uc7410	임베디드 통신 변환기
IP 공유	가능	불가능	가능
무선 통신	가능	불가능	불가능
직렬 포트	없음	있음	있음
직렬 통신	불가능	가능	가능
FTP, Telnet 지원	불가능	가능	가능
Web-sever와 CGI 지원	수정불가	없음	수정가능
Web-based Monitering 지원	불가능	불가능	가능



(그림 17) 웹 모니터링 시스템

시스템을 통하여 확인가능하다. 그러므로 이를 이용하여 (그림 16)의 지역 컴퓨터는 가정 또는 태양열 발전기를 관리하는 회사에서도 금일 혹은 월별, 년별 발전량을 알 수 있으며 인터넷을 통하여 임베디드 통신 변환기에 접속하여 태양열 발전기 혹은 계측기의 고장유무 등을 웹에서 상시 모니터링 할 수 있다.

5. 결 론

지금까지 직렬 통신을 인터넷 통신으로 중계하는 임베디드 통신 변환기의 개발 과정을 설명하였다. 임베디드 통신 변환기의 개발을 위하여 UART를 추가하고 WAN과 2개의 LAN을 구성하였다. 또한 원격 제어가 가능하도록 boa 서버를 설치하고, 사용자와 장치간의 인터페이스 위해 CGI 어플리케이션을 작성하였다. 또한 현장에 설치된 장비의 에러 상황에 대처하기 위한 방법으로 복구용 데이터를 임시로 저장 가능하도록 SRAM이 포함된 RTC를 추가하였다. 개발된 통신 변환기의 유효성 검증을 위하여 태양열 에너지 관리 시스템에 적용한 예를 제시하였다. 이 장비에 의하여 직렬 통신과 이더넷 통신이 상호 변환되는 기능은 물론 IP공유 기능이 가능해 졌다. 또한 기존의 장비와 구별되는 원격에서 웹 모니터링이 가능하도록 모니터 프로그램이 작성되었다.

컴퓨터 기술의 발전으로 산업용 장비들도 인터넷을 활용할 수 있는 인프라가 구축되어 있고, 인터넷의 급속한 발전으로 인하여 대부분의 최신 장비는 처음부터 네트워크 기능이 추가되어 출시되고 있다. 그러나 몇 년 전만해도 대부분의 산업용 장비들은 직렬 통신에 의한 정보 수집 기능만 보유하고 있었다. 본 연구에서 소개하는 임베디드 통신 변환기는 이러한 구형 장비들에게 최신 기술을 적용하는 역할을 할 것으로 기대한다. 또한, 임베디드 통신 변환기는 라이센스 제약을 받지 않는 리눅스 운영체제 환경에서 개발되었기 때문에 기업의 경제적인 부담을 크게 줄일 수 있다는 장점이 있다.

개발된 임베디드 통신 변환기는 여러 지역에 설치되어 있는 구형 장비들을 중앙에서 감시하는 용도로 다양한 응용분야에서 사용될 것으로 기대하고 있다. 현재는 장비에 연결되는 직렬 통신 포트가 하나이지만, 여러 개의 직렬 통신 포트들 갖도록 하드웨어를 개발하면 한 대의 임베디드 통신 변환기로 여러 개의 직렬 통신 장비를 제어하도록 확장할 수 있다.

참 고 문 헌

[1] 임채덕 외 6인, "임베디드 소프트웨어의 기술동향 및 산업 발전 전망," *Institute of Information Technology Assessment*, 2002.

[2] K. Goldberg, M. Mascha, S. Gentner, N. Rothenberg, "Desktop teleoperation via the World wide web", *IEEE int. conf. Robotics and automation*, 1995, pp.654-659.

[3] www.e-infomax.com/ipmasq/howto-trans/kr/IP-Masquerade-HOWTO-3.html.

[4] Bentham, Jeremy, *'TCP/IP LEAN Web Servers for Embedded Systems'*, CMP BOOKS, 2004.

[5] BOA, http://www.boa.org/documentation/.

[6] http://hoohoo.ncsa.uiuc.edu/cgi/intro.html.

[7] MICREL, http://www.micrel.com/product-info/archives.shtml.

[8] http://www.twistsemi.com/products/450_550_uart.html.

[9] http://www.alldatasheet.co.kr/datasheet-pdf/pdf_kor/STMICROELECTRONICS/M48T37.html.

[10] Karim Yaghmour, *'Building Embedded LINUX SYSTEMS'*, O'RELLY, 2003.

[11] http://www.premier-electric.com/files/STM/pdf/M48T37.pdf.

[12] Rubini, Corbet, *'Linux Device Drivers'*, O'RELLY, 2001.

[13] T. Sridhar, *'Designing Embedded Communications Software'*, CMP, 2003.7.

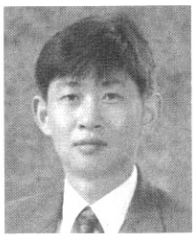
[14] Daniel P. Bovet, Marco Cesati, '*Understanding the Linux Kernel*', O'RELLY, 1999.
[15] Scott Guelich, Shishir Gundavaram, Gunther Birznieks, '*CGI Programming with Perl (2nd Edition/ Paperback)*', O'RELLY, 2000.
[16] http://www.micrel.com/_PDF/Ethernet/full_ds/ks8695px%20ds.pdf.
[17] http://www.terawork.co.kr/sub2_01.htm?Search=강원전자&part=4&cate=43&mode=1
[18] http://www.moxa.com/product/UC-7400.htm

이 병 권

e-mail : sonic474@msn.com
1999년 한밭대학교 전자계산(학사)
2002년 한남대학교 컴퓨터공학(석사)
2003년 충북대학교 전자계산대학원 박사과정
관심분야 : 임베디드시스템, 컴퓨터구조, 퍼지이론

전 영 숙

e-mail : yschon@hanmail.net
1996년 한남대학교 전자계산학과(학사)
1998년 한남대학교 컴퓨터공학과(석사)
2002년 충북대학교 컴퓨터과학과 박사 수료
관심분야 : 고성능 컴퓨터 구조, 병렬 처리, 메모리 시스템, 멀티미디어 시스템



전 중 남

e-mail : joongnam@cbu.ac.kr
1990년 연세대학교 전자공학과(박사)
1996년~1998년 미국 텍사스 A&M University 연구원
1990년~현재 충북대학교 전기전자컴퓨터공학부 교수
관심분야 : 컴퓨터 구조, 임베디드 시스템