

마이크로프로세서를 이용한 인터넷 원격감시제어 시스템에 관한 연구

서 인 호* · 유 영 호**

A Study on an Internet Remote Control and Monitoring System using a Microprocessor Embedded Controller

I-H Seo* · Y-H Yu**

Key words : Internet remote control and monitoring(인터넷원격감시제어장치), NE2000 compatible NIC(NE200호환NIC), Interfacing NIC(NIC인터페이싱)

Abstract

Serial communications such as RS-232C or RS-485 have been used to control and monitor the industrial plants for a long time in cooperating with a computer or microprocessor. In recent years a great deal of effort has been made to achieve these control and monitoring through Internet network. This paper proposes a microprocessor system to implement remote control and monitoring system through Internet network. The proposed system uses NE2000 compatible NIC for data link and physical layer to access Internet network. The microprocessor employed in this system plays a role of interfacing NE2000 compatible NIC, interpreting protocols above data link layer, controlling and monitoring industrial plants simultaneously. This paper also shows MMI and experimental results which control and monitor two power plants on the computer monitor with a mouse remotely to verify the proposed.

1. 서 론

최근 산업현장의 자동화는 제어기능별로 구분된 분산제어시스템을 네트워크로 연결시킴으로서 점점 거대화되어 가고 있다. 또한 제어시스템은 계층적 구조로 되어 있어 하층구조에서는 기계의 제

어와 감시의 기능을 수행하고 상층구조에서는 하층구조로부터 데이터를 수집하여 MMI에 표시하여 전반적인 시스템운용을 용이하게 하고 경향분석을 토대로 하여 시스템을 최적으로 운용하기 위한 명령을 내리거나 자재의 수급이나 주문을 자동으로 행하거나 관련 정보시스템과 연계하여 경영

* 한국과학기술원 인공위성연구소 연구원(원고접수일 : 2001년 3월)

** 한국해양대학교 기계·정보공학부

정보를 제공하는 등 주로 의사결정과 관련되는 전문적이며 지능적인 기능을 담당한다. 이러한 거대 자동화 시스템에 있어서 각 제어기기 사이의 원활한 데이터 통신이 이루어지도록하는 것은 매우 중요하며 대부분 하층구조에서는 RS232C/422A/485와 같은 물리계층을 이용하여 전용의 프로토콜에 의한 데이터통신을 구현하고 상층구조에서는 Ethernet 등의 망을 이용하여 구현한다.¹¹⁾ 그러나 이러한 경우에는 시스템이 거대화되어 감에 따라 하층구조의 복잡한 배선, 과도한 유지보수비, 데이터전송방식이 다른 기기간 통신의 어려움, 비동기 통신 방식의 전송 속도와 연결 시스템 수의 제한 등 많은 단점이 있을 수 있다.¹²⁾⁴¹⁾

따라서 본 논문에서는 이러한 단점을 보완하고 넓게 분포되어 있는 인터넷망을 활용한 원격 제어 감시시스템을 구성하기 위하여 마이크로프로세서 시스템이 제어대상을 제어하고 감시함과 동시에 직접 기존의 인터넷망에 접속하여 원격지에 있는 호스트 컴퓨터와 데이터통신을 행하게 함으로써 원격지에서 제어대상을 감시하고 제어할 수 있는 인터넷 원격감시 제어 시스템을 설계하고 구현하고자 한다. 이를 위하여 OSI 7층 구조의 프로토콜 중 물리층과 데이터링크층을 담당하고 있는 NIC(Network Interface Controller)는 현재 컴퓨터시스템에서 많이 사용하여 실용성과 신뢰성이 검증되어 있는 기존의 NE2000 호환 ISA 버스 NIC를 사용하고 이를 제어하기 위하여 16비트 마이크로프로세서 80C196KC를 사용한다. 또 OSI 7층구조 중 네트워크층 이상의 프로토콜 구현과 플랜트의 제어감시는 NIC를 제어하고 있는 마이크로프로세서가 담당하여 마이크로프로세서시스템이 직접 인터넷에 접속하여 데이터통신을 가능하게 한다.¹³⁾ 이렇게 함으로서 마이크로프로세서가 플랜트를 제어감시함과 동시에 원격지에 있는 컴퓨터시스템의 요구에 따라 인터넷망을 통하여 데이터를 주기적으로 전송하는 하층구조의 역할을 담당하고 윈도우즈 운영체제에서 동작하는 원격지 컴퓨터시스템은 전송된 플랜트의 제어감시 데이터를 MMI 응용프로그램과 연결하여 원격지에 있는 플랜트를 제어함과 동시에 감시된 데이터로부터 경향분석 등을 통하여 분산제어시스템의 상

층구조의 역할을 담당한다.

2. 프로토콜과 NIC

2.1 OSI 7층 프로토콜과 패킷의 구조

현재 통신 프로토콜은 국제표준화기구가 규정한 OSI(Open System Interconnection)의 기본 참조 모델이 가장 대표적이다. 이것은 기술적인 독립성, 기능적인 공통성이라는 관점에서 통신 기능을 7개로 분할하고 기능에 따라 물리층, 데이터링크층, 네트워크층, 트랜스포트층, 세션층, 프리젠테이션층, 어플리케이션층 등 7개 층으로 계층화되어 있다.¹³⁾⁵¹⁾ Fig. 1은 프로토콜의 OSI 기본참조 모델을 나타낸다.

Ethernet은 OSI 7층 프로토콜중 데이터링크층을 담당하고 있는 프로토콜로서 Ethernet Ver.2.0(DIX 사양)과 IEEE 802.3 이라는 2가지 표준이 존재하며 이 2개의 표준을 총칭하여 Ethernet이라 부른다. Ethernet은 CSMA/CD라는 전송제어 방식을 사용하며 NIC의 LAN 컨트롤러 칩이 담당하고 있으며 본 논문에서 사용한 NE2000 호환 NIC에서는 RTL8019AS 마이크로프로세서를 사용하고 있다.

Fig. 2는 Ethernet 프레임의 구조를 나타낸다. 그림에서 숫자는 필드의 크기를 바이트로 나타낸 것이다. 프리앰블 필드는 동기화를 맞추기 위한 것이고 CRC필드는 에러체크기능필드로서 LAN 컨트롤러칩이 담당하고 있으므로 고려치 않아도 된다. 수신 어드레스필드와 송신 어드레스필드를 MAC 어드레스라 하고 Ethernet 인터페이스는 고유한 MAC 어드레스를 가지고 있으며 이 MAC 어드레스를 통해 상대방을 인식한다.

타입필드는 데이터부의 프로토콜을 식별하는 값으로 IP는 0800H, ARP는 0806H이다. Ethernet 패킷은 최소 64바이트 최대 1518바이트까지이며 64바이트 미만인 경우에는 어떤 데이터(일반적으로 0)로 패딩 한다. MAC어드레스는 모두 48비트로 구성되며 8비트씩 나누어 표기한다. 또한 모든 비트가 1인 브로드캐스트 어드레스나 여러 Ethernet 인터페이스에게 동일 데이터를 한

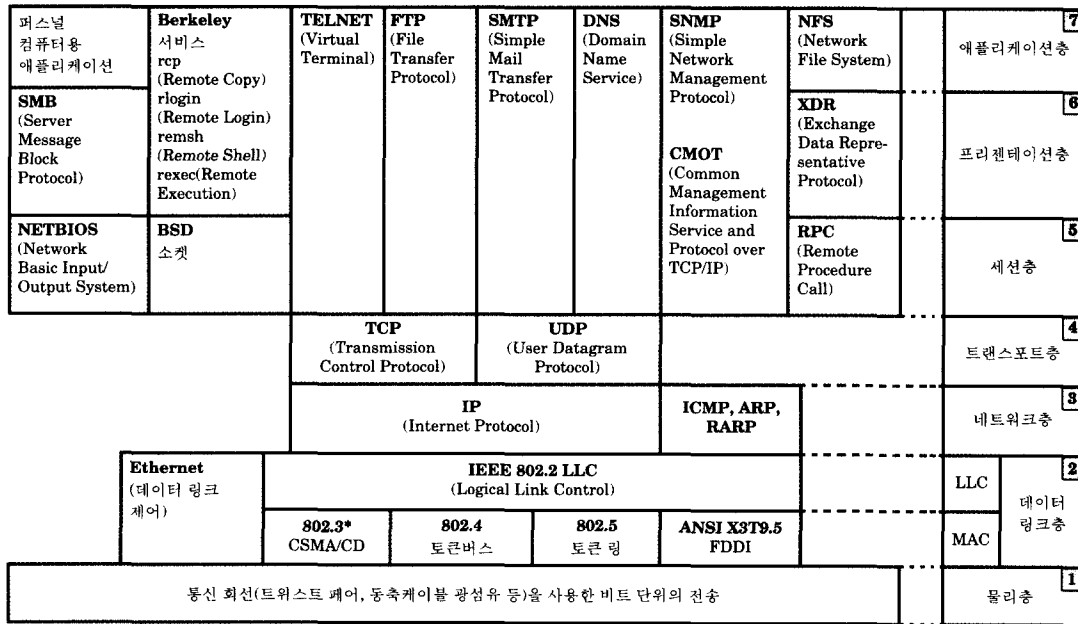


Fig. 1. OSI reference model

꺼번에 송신하기 위한 멀티캐스트 어드레스가 정의되어 있다.

IP프로토콜은 호스트간 또는 호스트와 게이트웨이 간에 통신을 실현하여 데이터그램을 목적하는 곳까지 전송하는 프로토콜이다. 이 프로토콜은 수신지 IP 어드레스로 경로를 선택하는 경로선택 기능과 데이터그램의 크기가 MTU(Maximum Transfer Unit)보다 클 경우 몇 개의 프래그먼트로 분할하여 송신하는 프래그먼트이션 기능이 있다. IP어드레스는 32비트로 구성되며 8비트씩 나누어 표기한다. 호스트는 상대방의 IP 어드레스만 알고 MAC 어드레스를 알지 못하기 때문에 MAC 어드레스를 알아내기 위한 프로토콜을 필요로 하는데 이 프로토콜을 ARP라하며 OSI 7층의 네트워크층에서 이 기능을 수행한다.

UDP는 IP의 기능이 어플리케이션에서 사용되도록 하는 프로토콜로서 어플리케이션에는 여러 종류가 있으므로 이를 구별하기 위해서 포트번호를 사용한다. 포트번호는 특정목적으로 웰노운(well known)이라는 예약되어 있는 번호가 있으므로 1000번 이상을 사용하도록 권하고 있다.

TCP는 과도한 오버헤드가 있어 마이크로프로

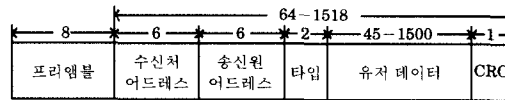


Fig. 2. Ethernet packet format

세서로 인터넷에 접속하고 실시간으로 제어감시하는 데에는 많은 부하를 주기 때문에 본 연구에서는 융통성과 자동 회복 그리고 간단한 인터페이스로써 구현이 가능한 장점이 있는 UDP를 사용하였다.^{[5][6][7]}

UDP/IP 프로토콜에 의한 패킷 송신 과정은 다음과 같다. 응용프로그램에서는 상대방의 IP 어드레스로 통신요구가 발생하므로 수신측에서 사용하고 있는 NIC의 MAC 어드레스를 알기 위하여 먼저 Ethernet 브로드캐스트로 ARP요구 패킷을 송신한다. 모든 노드에서 ARP 요구 패킷을 수신하고 해당 노드는 발신지 Ethernet 어드레스로 자신의 MAC 어드레스를 첨부하여 ARP 응답 패킷을 송신한다. 이 때 IP프로토콜에 의해 수신된 경로를 알고 있으므로 역경로를 통하여 IP패킷을 송신하면 송신측에서는 이를 수신하여 목적지의

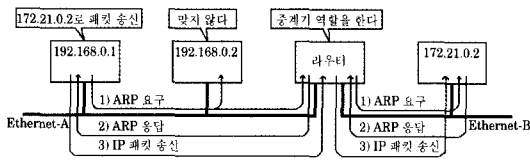


Fig. 3. Packet transmission using the UDP/IP

| | |
|--------------|--|
| 하드웨어 어드레스 타입 | 2...DIX-Ethernet=1 |
| 프로토콜 어드레스 타입 | 2...IP=0800H |
| 하드웨어 어드레스 길이 | 1...Ethernet=6 |
| 프로토콜 어드레스 길이 | 1...IP=4 |
| 코 드 | 2... ARP=1 ARP=2 |
| 소스 하드웨어 어드레스 | 6... ARP 응답이라면 Ethernet 어드레스가 온다 |
| 소스 프로토콜 어드레스 | 4 |
| 타겟 하드웨어 어드레스 | 6 |
| 타겟프로토콜 어드레스 | 4... ARP 요구이라면 얻고 싶은 어드레스 에 대응하는 IP 어드레스를 저장한다. |

Fig. 4. ARP packet format

MAC 어드레스를 알아낸 후 목적지 MAC 어드레스로 데이터패킷 전송을 시작한다. Fig. 3은 이 과정을 나타낸다.

Fig. 4는 ARP패킷의 구조를 나타낸다. 하드웨어 어드레스 타입 필드는 Ethernet의 종류를 나타내는 것으로 DIX Ethernet의 경우는 1이 된다. 프로토콜 어드레스 타입 필드는 상위 프로토콜을 나타내며 IP이므로 0800H이다. 하드웨어 어드레스 길이와 프로토콜 어드레스 필드는 각각 6과 4가 들어간다. 코드 필드는 패킷의 종류를 나타내는 것으로 ARP요구이면 1, ARP응답이면 2를 넣는다. ARP요구패킷의 경우에는 타겟 IP 어드레스 필드에 상대방 IP 어드레스를 저장하고 타겟 MAC 어드레스 필드는 모르므로 1로 채우며 소스 하드웨어 어드레스와 소스 IP 어드레스 필드는 자신의 어드레스를 저장하여 송신한다. ARP 응답패킷은 상대방의 MAC 어드레스와 IP 어드레스를 알고 있으므로 ARP요구 패킷과 같은 형태로 구성하여 송신한다. ARP의 경우 패킷의 길이가 64바이트 미만

| 버 전 | 헤더길이 | |
|-------------|------|---|
| | | 1...각각 4비트 [버전번호=4 헤더길이=5 |
| TOS | | 1...우선도를 표현하며 통상적으로 0 |
| 토탈 길이 | | 2...패킷의 길이(바이트 단위) |
| 시퀀스 번호 | | 2...패킷을 생성한 호스트의 순서번호 |
| 프래그먼트 정보 | | 2...프래그먼트하지 않은 경우는 0 |
| TTL | | 1...생존시간을 나타내며 통상적으로 255 |
| 프로토콜 번호 | | 1... 데이터부의 프로토콜을 나타내며 TCP 는 6이고 UDP는 17이다 |
| 헤더 체크섬 | | 2...헤더부의 체크섬으로 1의 보수로 합하고 -1한 값이 들어간다. |
| 발신지 IP 어드레스 | | 4...이 패킷을 생성한 호스트의 IP 어드레스 |
| 수신지 IP 어드레스 | | 4...이 패킷을 수용하는 호스트의 IP 어드레스 |
| 헤더 옵션 | | 임의의 길이...통상적으로는 사용하지 않는다 |
| 데이터 | | 임의의 길이... 전체 길이는 16바이트이므로 65535바이트까지 퍼드할 수 있다 |

Fig. 5. IP packet format

이므로 남은 부분은 0으로 패딩 한다.

Fig. 5는 IP 패킷의 구조이다. IP 패킷은 IP 헤더부와 데이터부로 나눈다. 버전 필드는 IP프로토콜의 버전을 나타내며 현재버전이 IPv4이므로 4가 된다. 헤더길이 필드는 IP헤더길이를 4로 나눈 값을 나타내며 헤더길이가 20이므로 5가 된다. TOS(Type of Service) 필드는 데이터그램을 어떻게 처리할 것인가를 정의하는 필드로서 일반적으로 0을 넣는다. 전체 길이 필드는 IP헤더와 데이터의 총 길이로서 16바이트이므로 65535 바이트가 최대가 된다. 하나의 데이터 크기가 MTU보다 클 경우에는 몇 개의 데이터그램으로 나누어 전송하고 이를 다시 재조립해야 하는데 이를 위해 필요한 것이 시퀀스번호와 프래그먼트 정보이다. 시퀀스번호는 데이터그램이 쪼개어질 때 어느 데이터그램으로부터 쪼개어진 것인지를 표시하고, 프래그먼트 정보는 현재의 데이터그램이 전체 데이터그램의 어느 위치에 있는 것인지를 표시한다. 프래그먼트가 없는 경우에 프래그먼트 정보는 0이 된다. TTL(Time to Live) 필드는 데이터그램이 통신망에서 얼마나 오랫동안 남아 있을 수 있는가를 나타

내는 필드로서 게이트웨이나 호스트는 소요된 시간만큼 감소시키다가 0이 되면 데이터그램을 통신망에서 제거하며 처음에는 255의 값을 가진다. 프로토콜번호필드는 상위 프로토콜을 나타내는 것으로 TCP는 6, UDP는 17이다. 헤더 체크섬 필드는 에러제어 기능을 담당하는 필드로서 헤더 체크섬 필드를 0으로 초기화한 후 전체 헤더를 16비트 단위로 1의 보수로 가산한 다음 합계의 비트를 반전하여 -1한 값을 체크섬의 값으로 한다. 헤더 옵션 필드는 일반적으로 사용하지 않는다.

네트워크에서 자료의 패킷 교환방식은 전달형태에 따라 연결형인 스트림 통신과 비연결형인 데이터그램 통신이 있다. 스트림 통신에서는 두 호스트간에 자료를 전송하기 전에 초기화작업을 통하여 두 개의 프로세스 사이에 실제로 존재하지 않는 가상의 통신로를 확립하고 이 통신로를 통하여 자료를 전송한다. 그러므로 가상통신로가 끊어지면 자료를 전송할 수 없게 되지만 신뢰성 높은 통신을 할 수 있다. 데이터그램 통신은 헤더와 데이터로 구성된 유일한 패킷을 서로 독립적으로 처리하여 보내는 방식으로 가상의 통신로가 존재하지 않으므로 같은 목적지의 주소를 가지는 패킷들이라도

패킷의 경로 배정이 다를 수도 있으며 트래픽에 따라 도착순서가 바뀔 수도 있다.¹⁷⁾ 반면에 가상의 통신로가 없으므로 연결설정 등 오버헤드가 작아 프로토콜이 간단하여 사용하기가 편하다.

이러한 데이터 통신의 장단점으로 어떤 방식을 제어모니터링 시스템에 사용할 것인가에 대하여서는 많은 논란이 있으나 본 연구에서는 데이터 통신과 제어의 기능을 동일한 마이크로프로세서가 담당하고 또 Fig. 6과 같이 여러 제어대상 기기로부터 하나의 호스트로 감시데이터를 독립적으로 보내고 또 하나의 호스트로부터 여러 제어대상 기기로 제어 명령을 주기적으로 보내는 구조의 제어 감시 시스템을 구성하는 것을 목적으로 하고 있으므로 연결설정에 오버헤드가 없는 비연결형 데이터그램 통신을 사용한다.

UDP는 IP의 기능을 어플리케이션에서 사용하도록 하는 프로토콜로서 여러 개의 어플리케이션이 있을 수 있으므로 구별하기 위하여 포트번호를 사용한다. 포트번호는 자주 사용되는 어플리케이션에 미리 예약되어 있는데 예를 들면 DNS는 53, DHCP는 67 등 0에서 511번까지는 예약되어 있어 사용자는 1000번 이상을 사용한다. Fig. 7은 UDP 패킷의 구조를 나타낸다. UDP는 헤더부와 데이터부로 구성되어 있으며 헤더부는 단순하여 용이하게 구현할 수 있다. 발신지 포트번호와 수신지 포트번호 필드는 이용자가 정한 번호를 기입하며 일

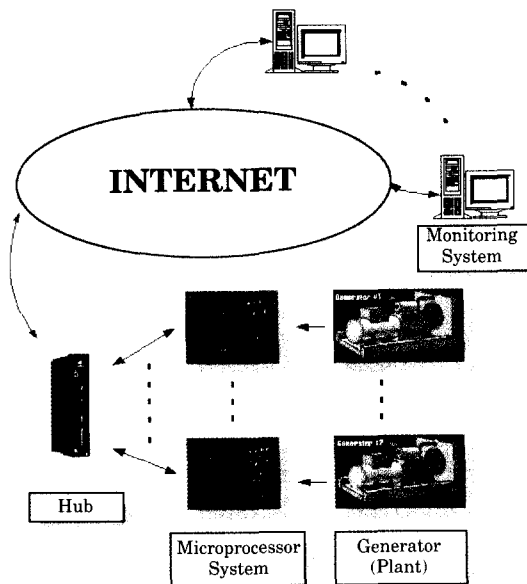


Fig. 6. Configuration of overall system

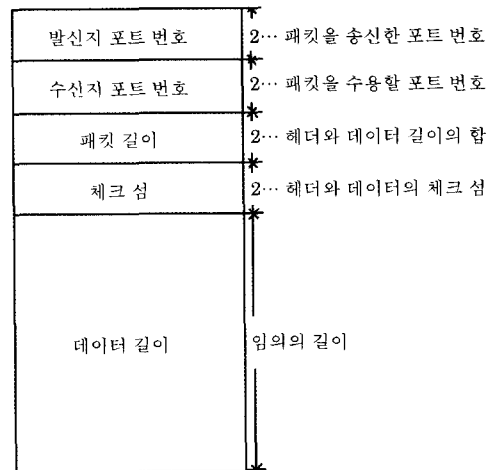


Fig. 7. UDP packet format

반적으로 서로 같게 설정된다. 패킷길이 필드는 UDP헤더부와 데이터부의 길이의 합이 들어가며 체크섬은 IP헤더와 같은 방법으로 1의 보수로 합계를 계산한다. 한가지 다른점은 체크섬이 0 또는 0FFFFH가 되었을 때 UDP는 0FFFFH를 저장하며 IP는 그대로 0 또는 0FFFFH를 저장한다. 이렇게 하는 이유는 UDP 체크섬은 옵션이므로 생략할 때 0을 기입하기 때문이며 생략하면 신뢰성이 떨어지게 된다. 데이터부는 실제로 전송하고자 하는 데이터로서 아스키 값이다.

2.2 NIC의 구조

NE2000 호환 NIC는 Ethernet 컨트롤러인 RTL8019AS와 송수신 버퍼 RAM, 시리얼 EEPROM, ISA 버스 인터페이스 회로, 미디어 인터페이스 회로로 구성되어 있다.

NIC의 MAC 어드레스, I/O 어드레스와 IRQ의 설정은 시리얼 EEPROM에 저장된다. NIC가 리셋되면 시리얼 EEPROM에서 설정값을 판독하여 초기화한다. 시리얼 EEPROM의 메모리 영역은 0000H에서 001FH까지이다. NIC에는 4000H에서 7FFFH까지 송수신 버퍼용 RAM이 매핑되어 있어 CPU에서 판독 및 기록이 가능하다. 이때 CPU에서 송수신 버퍼 RAM에 접근하기 위하여 RTL8019AS에 내장된 2개의 로컬 DMA와 리모트 DMA를 사용한다. 수신된 패킷은 로컬 DMA에 의

해 자동적으로 버퍼 RAM에 저장되며 송신패킷은 송신명령과 로컬 DMA에 의해 자동으로 송신된다. 또 리모트 DMA는 CPU와 버퍼 RAM 사이의 데이터 교환에 사용된다.

Fig. 8은 NIC의 I/O 공간을 나타내고 있다. ISA 버스 I/O 공간은 000H에서 03FH까지 이고 Fig. 8과 같이 NIC는 32바이트를 차지하고 있으며 개시 어드레스로는 240H, 280H, 300H, 340H, 360H등 16개가 있다. NIC의 32바이트 중 상위 16바이트는 RTL8019AS에 할당되어 있으며 하위 16바이트 가운데 상위 8바이트는 데이터 포트이고 하위 8바이트는 리셋 포트이다. 이중에서 데이터 포트는 16비트로 나머지는 8비트로 액세스된다. 또 인터럽트로써 8개의 IRQ 라인 중 1개의 IRQ만 사용 가능하며 본 연구에서는 개시 어드레스 300H와 IRQ 9번을 사용하였다. RTL8019AS에는 약 40개의 레지스터와 4개의 페이지가 있다. 통상적으로는 16개의 레지스터가 보이며 페이지 교체에 의하여 레지스터가 달라지게 된다. 페이지 1은 MAC 어드레스와 멀티캐스트 어드레스를 저장하기 위한 레지스터이고 페이지 3은 예약되어 있으므로 통상적으로 사용하지 않으며 일반적으로 페이지 0와 페이지 2의 레지스터를 사용한다. 또한 RTL8019AS는 수신버퍼를 페이지 단위(256 바이트)로 나누어 링버퍼를 만든다. 수신버퍼는 총 59 페이지로 되어 있으며 페이지 0에서부터 수신을 시작하여 58페이지까지 수신하고 다시 페이지 0으로 되돌아가 수신하여 유한한 버퍼 영역을 연속적으로 액세스하게 되어 수신된 데이터를 효과적으로 관리할 수 있게 된다. Fig. 9는 링버퍼의 구조를 나타낸다.

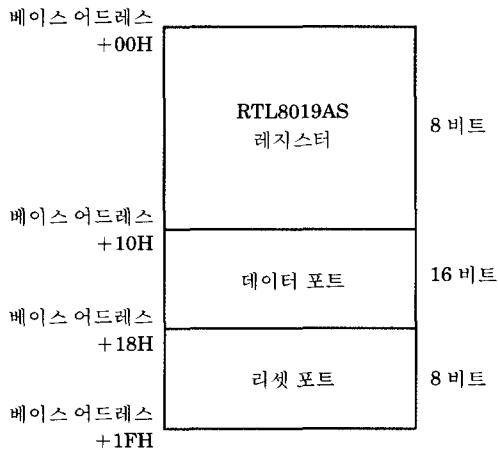


Fig. 8. NIC I/O map

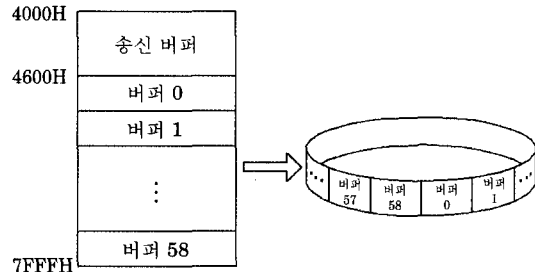


Fig. 9. Ring buffer structure and memory map

3. 원격감시제어 시스템 설계 및 구현

3.1 전체 시스템의 구성

전체 시스템은 수신된 데이터로 명령을 수행하고 플랜트의 온도나 압력 등 각종 센서 값을 인터넷을 통하여 PC로 전송하는 마이크로프로세서 시스템과 인터넷상에서의 마이크로프로세서 시스템과의 데이터통신을 통하여 플랜트의 상태를 원격으로 제어·감시하고 경향분석 등을 행하는 모니터링 시스템으로 구성되며 Fig. 6에 전체 시스템의 구성도를 나타내었다. Photo. 1은 개발된 시스템의 외형으로 인텔 80C196KC 마이크로 프로세서, NE2000 호환의 ISA 버스용 NIC, LCD(4×20), Key로 구성되어 있다.¹⁰⁾ 마이크로프로세서는 NIC에 의한 데이터 송수신을 통하여 OSI의 어플리케이션층, 트랜스포트층, 네트워킹층의 프로토콜 기능을 수행하고 NIC는 데이터링크층과 물리층의 기능을 수행한다. 또한 프로토콜의 순차적인 구현을 알아보기 위하여 LCD 표시장치와 Key를 사용하였다. Fig. 10은 개발된 시스템의 구성도를 나타

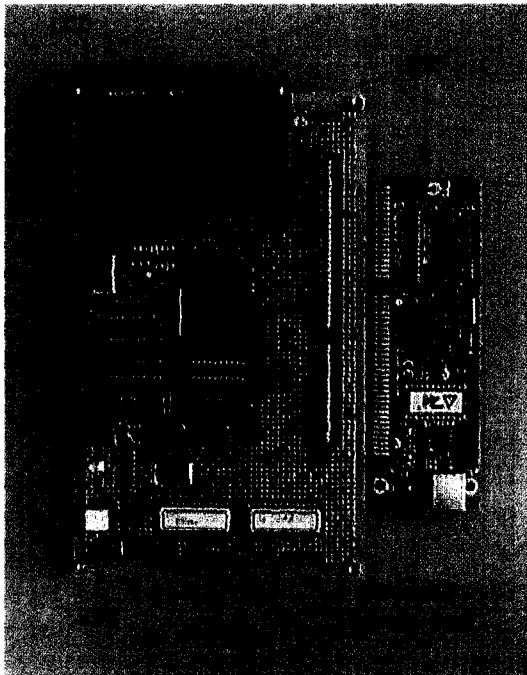


Photo. 1. Developed microprocessor system

내고 있다. NIC가 데이터 수신을 완료하여 인터럽트신호를 발생하면 80196KC프로세서는 HSI0 인터럽트로 받아 수신버퍼의 데이터를 읽어 RAM영역에 저장한 후 ARP, UDP/IP 프로토콜을 수행한다. 송신시에는 80196KC 마이크로프로세서는 ARP, UDP/IP 프로토콜을 수행하여 송신할 데이터에 해당하는 프로토콜의 헤더를 부쳐 NIC의 송신 버퍼에 데이터를 전송하고 NIC에 송신명령을 내려 NIC가 송신 동작을 수행하도록 한다.

3.2 소프트웨어의 구성도

Fig. 11에 전체 소프트웨어의 흐름도를 나타내었다. 프로그램은 크게 두 부분으로 나누어진다. 첫 번째는 ARP, UDP/IP 프로토콜을 처리하는 부분이고 두 번째는 LCD에 데이터를 보여 주는 부분이다. 실제로 제어시스템에 적용되었을 때는 대상시스템을 제어하는 프로그램이 추가되어야 하겠지만 대상시스템을 제어하는 프로그램은 본 연구의 범위를 벗어나는 것이므로 여기에서는 인터넷상의 호스트 컴퓨터로부터 제어신호 1점을 수신하여 제어신호를 출력하고 프로세서에서 랜덤 데이터를 발생하여 전송하고 인터넷상의 호스트 컴퓨터는 이 신호를 인터넷을 통하여 수신하여 MMI에 표시하는 기능과 경보기능, 경향그래프기

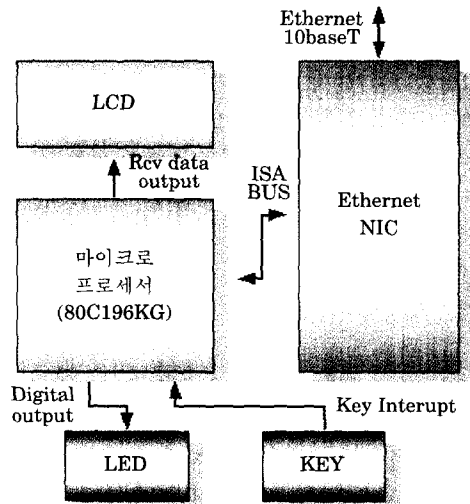


Fig. 10. Block diagram of the microprocessor system

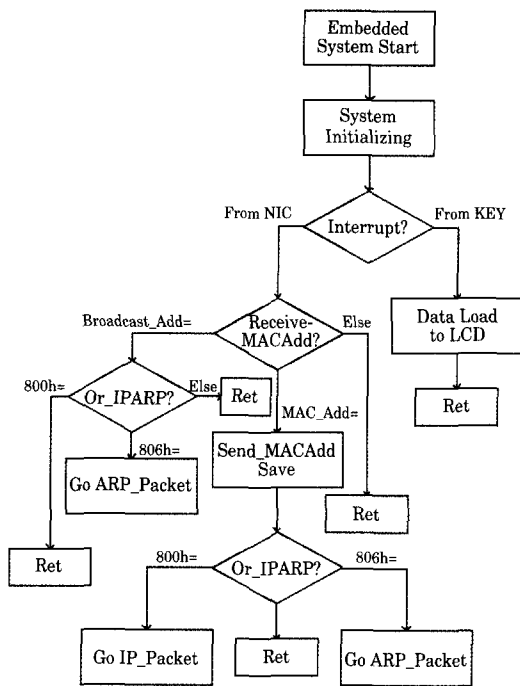


Fig. 11. Software flow chart

능 등 몇 가지 기능만 예시하였다. 또한 마이크로 프로세서 시스템의 메모리는 32Kbyte이므로 실제로 제어대상을 제어하는데 사용되는 제어프로그램의 활용영역을 고려하여 Ethernet의 1 패킷(1518 바이트)을 저장할 수 있는 최소한의 영역을 할당하였다.

Fig. 11에서 시스템이 시작되면 MAC어드레스, IP어드레스, NIC의 송수신 버퍼를 체크하여 초기화한 후 NIC로부터 패킷수신완료 인터럽트신호를 대기한다. NIC로부터 패킷수신을 완료하여 인터럽트신호가 발생되면 프로세서는 수신한 데이터를 RAM에 저장하고 수신 패킷의 헤더를 체크하여 해당 프로토콜의 루틴으로 점프하여 작업을 수행한다.

Fig. 12는 ARP 소프트웨어의 흐름도이다. 통신을 개시할 때 인터넷상의 원격 호스트는 플랜트를 제어하고 있는 마이크로프로세서 시스템의 MAC어드레스를 알지 못하므로 ARP요구 패킷을 송신한다. 수신측에서는 ARP요구 패킷을 수신하여 자신의 IP어드레스와 같으면 ARP응답 패킷을 만들

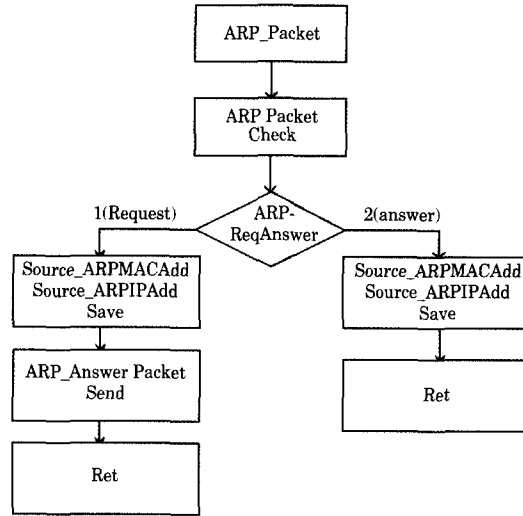


Fig. 12. S/W flow chart of ARP protocol

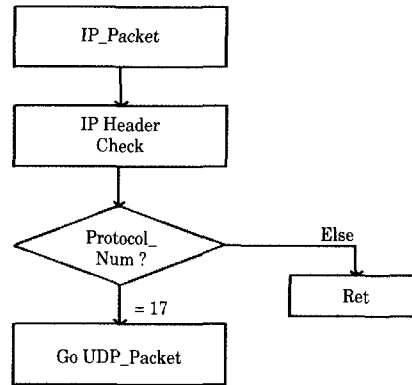


Fig. 13. S/W flow chart of IP protocol

어 전송하고 같지 않으면 수신된 ARP 요구패킷을 파기하고 다음 패킷수신을 대기한다. ARP응답 패킷으로 마이크로프로세서 시스템의 MAC어드레스를 확인하면 호스트는 IP 패킷을 송신한다. Fig. 13은 IP프로토콜 소프트웨어 흐름도이다. 그림에서 IP패킷을 수신한 후 헤더를 체크하여 자신의 IP 패킷이 맞으면 상위 프로토콜번호를 확인한 후 해당 루틴의 작업을 수행한다. 본 연구에서는 UDP를 사용하므로 상위 프로토콜 번호는 17이며 이것과 다른 패킷은 파기한다. Fig. 14는 UDP 프로토콜 소프트웨어 구성도이다. UDP 프로토콜로 넘어 온 패킷은 UDP헤더를 체크한 후 데이터를

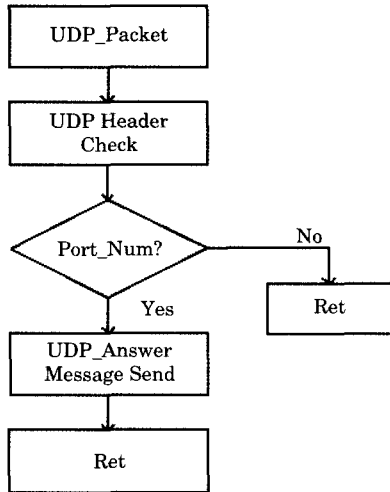


Fig. 14. S/W flow chart of UDP protocol

확인한다. 모든 헤더는 모두 이진수이나 데이터는 아스키코드에 해당하는 16진수이므로 이를 아스키코드로 변환한다. 본 연구에서는 인터넷상 원격호스트의 MMI로부터 플랜트를 제어하는 제어신호를 수신하면 제어신호를 표시하고 마이크로프로세서가 감시한 데이터를 호스트로 송신한다. 감시데이터를 수신한 원격호스트는 해당 플랜트의 데이터베이스에 데이터를 저장함과 동시에 MMI에 데이터를 표시하여 실시간으로 플랜트를 감시한다. 또한 감시자의 요구에 따라 데이터를 여러 가지 모양으로 가공하여 표시한다.

3.3 모니터링 시스템

모니터링 시스템의 MMI는 Visual C++ 6.0과 SQL 데이터베이스를 사용하여 개발하였다.^{[9][10]} Fig. 15는 본 연구의 한 적용 예로 개발된 모니터링시스템의 MMI로서 발전기 플랜트를 대상으로 하여 2기가 독립적으로 동작하도록 하였다. 각각 RPM, 온도, 압력 3개의 아날로그 데이터와 밸브 개폐용으로 1개의 디지털 데이터를 가지고 있다. 발전기의 제어용으로 밸브 등을 마우스로 더블클릭 하면 인터넷을 통하여 마이크로프로세서시스템에 전달되고 마이크로프로세서는 명령을 해석하여 제어신호를 발생한다. 제어명령신호와 감시된 신호가 일치하지 않을 때 밸브의 경우에는 노란

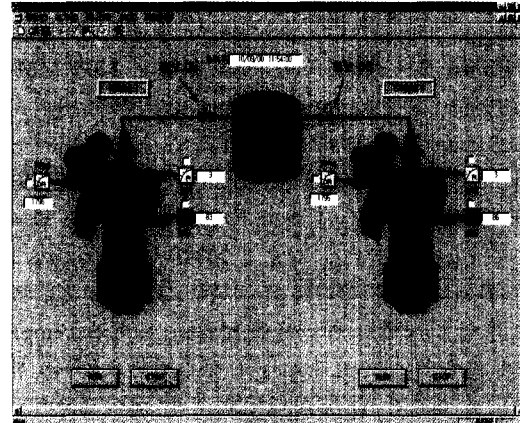


Fig. 15. MMI of remote control and monitoring system for generator plants

색으로 변하여 신호의 불일치를 알려주고 감시된 데이터가 명령신호와 일치되면 동작에 따라 밸브가 닫힌 경우이면 빨간색을, 열린 경우이면 파란색으로 변하여 색깔에 의해 동작상태를 알 수 있게 하였다. 마우스에 의한 제어동작이나 통신에 의해 감시된 데이터는 모두 데이터베이스에 저장되고 사용자의 요구에 따라 메뉴를 선택함으로써 날짜나 시간 또는 조건별로 가공되어 그래프로 볼 수 있어 MMI와 마우스에 의해 플랜트를 감시제어할 수 있다.

4. 실험 및 고찰

4.1 실험 장치의 구성

시스템의 유용성을 확인하기 위해 Fig. 6과 같이 부산지역에 허브를 이용하여 마이크로프로세서 시스템 2기를 설치하였고 부산과 서울에 각각 모니터링 시스템을 설치하여 실험을 행하여 보았다. 마이크로프로세서 시스템은 실제 설비 제약에 의해 발전기 플랜트에 직접 연결할 수 없어 온도, 압력, 회전수 등의 아날로그 데이터는 마이크로프로세서에 의해 가상의 랜덤데이터가 발생되도록 하였으며, 밸브 개폐용의 디지털 데이터는 LED를 On/Off 하여 원격지에서 제어 가능하게 하는지에 대한 실험을 행하였다.

4.2 동작 실험

Photo. 2는 밸브 개폐실험으로 Fig. 15의 MMI 화면에서 좌측 Engine은 Photo. 2의 좌측 마이크로프로세서시스템에 해당한다. 원격지 모니터링 시스템의 MMI화면의 밸브를 마우스로 더블클릭하여 Fig. 15와 같이 좌측 Engine의 밸브를 열고 우측 Engine 밸브를 닫으면 MMI 화면 Engine 1의 밸브가 열린 상태로 푸른색을, Engine 2의 밸브는 닫힌 상태로 빨간색을 나타낸다. 이때 Photo. 2의 좌측 마이크로프로세서 시스템의 LED는 점등으로 밸브는 열린 상태이고 우측 마이크로프로세서 시스템의 LED는 소등으로 밸브가 닫힌 상태를 표시하고 있어 원격제어가 원활히 이루어지고 있음을 알 수 있다.

Fig. 16은 경보실험을 행한 것이다. 마이크로프로세서 시스템에서 일정 범위 이상이나 이하의 데이터가 들어오면 경보를 울리고 원격지 MMI화면에서는 체크박스에 표시가 되므로 어떤 플랜트의 어떤 장치에 이상이 발생하였는지 여부를 쉽게 알 수 있게 된다. Fig. 16 (a)와 같이 MMI 화면상의 RPM에 체크 표시가 되어 있어 회전수에 경보가 발생하였다는 것을 알 수 있으며 Fig. 16(b)는 메뉴의 선택에 따라 데이터베이스에 저장된 데이터를 이용하여 회전수를 그래프로 그린 것으로 회전수가 일정 범위 이상으로 올라간 것을 확인할 수 있다.

패킷의 손실률을 알아보기 위해 마이크로프로

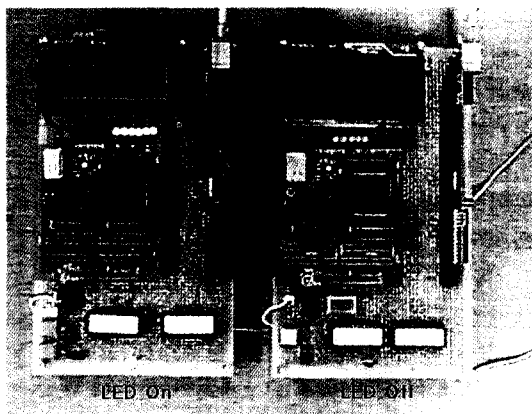
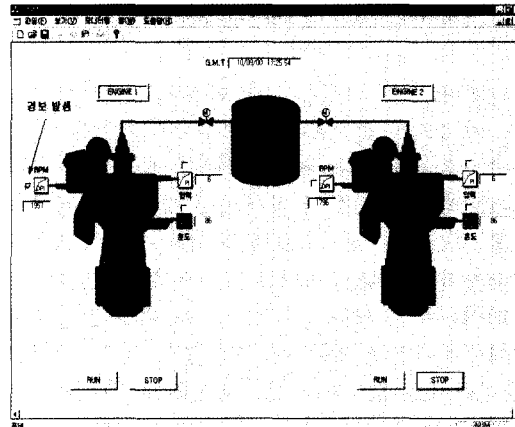


Photo. 2. Experiments for remote opening and closing of valve

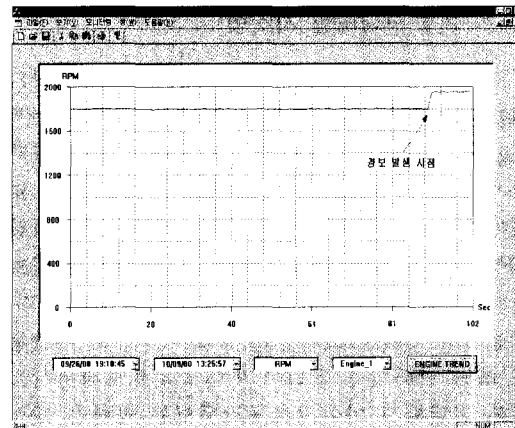
세서 시스템에서 0과 2의 데이터를 주기적으로 송신하여 모니터링시스템에서 수신한 데이터를 그래프로 그려본 결과 손실 없이 수신되는 것을 확인할 수 있었다. 또 서울과 부산에서 송수신할 경우 시간 지연을 알아보기 위해 실험한 결과 부산에서는 시간지연을 확인할 수 없었으며 서울에서는 트래픽의 사정에 따라 1초 이내의 시간지연이 있었음을 확인할 수 있었다.

5. 결 론

본 연구에서는 제어용으로 많이 사용되고 있는 인텔 80C196KC 마이크로프로세서를 이용하여



(a) Alarm on MMI



(b) Graph of rpm when an alarm is occurred

Fig. 16. Alarm test for rpm

제어대상을 제어함과 동시에 컴퓨터의 LAN카드 로 많이 사용하고 있는 NE2000호환 NIC을 인터 페이스함으로써 인터넷에 접속되어 있는 원격지 호스트 컴퓨터의 MMI 화면을 이용하여 제어대상 플랜트를 원격으로 감시 제어하는 시스템을 개발 하고자 하였다.

이를 위하여 NIC를 사용한 마이크로프로세서 시스템과 윈도우즈 운영체제에서 동작하는 모니터링 소프트웨어를 개발하고 인터넷을 통하여 원격지(서울)와 근거리(부산)에서의 동작실험을 행 함으로써 개발된 시스템이 효율적으로 동작하는 것을 확인할 수 있었다.

본 실험에서는 IP를 가진 마이크로프로세서 시 스템 2기를 사용하여 실험을 행하였으나 추후 많 은 마이크로프로세서 시스템과 연결한 경우에도 원활하게 동작하는지 확인하기 위하여 지속적인 실험을 행하여야 할 것이다.

이 논문은 2001년도 두뇌한국21사업에 의하여 지원되었음.

과학기술부의 한영과학기술협력기금사업에 의 하여 일부지원 되었음.

참고문헌

[1] J.R.Leigh, Applied Digital Control Theory, Design and Implementation 2nd edition, Prentice Hall 1992.
 [2] 이석원, 이영진, 강성귀, 남부희, "TCP/IP를 이용한 필드버스 monitoring 시스템 구현", KACC, pp.648-687, 1998
 [3] 손정호, "반도체 생산기기의 원격제어를 위한 인터넷 어댑터의 설계 및 구현", 부산대학교 대학원, 1999

[4] 박홍성, 진창기, 강영진, "Implementation of Easy API for MMS(Manufacturing Message Specification) over TCP/IP", KACC, pp.A41-A44, 1966
 [5] 진교홍, "고속실시간 통신을 위한 TCP/IP 프로토콜의 하드웨어 설계 및 구현", 부산대학교 대학원, 1997
 [6] 윤재식, 김재양, 정선태, "소형실시간 커널을 위한 TCP/IP 프로토콜 설계 및 구현", KACC, pp688-691, 1998
 [7] 하중성, User Datagram Protocol의 개발과 응용, 한국과학기술원, 1985.1.
 [8] 차영배, MICRO CONTROLLER 80196, 다다미디어, 1997.
 [9] 김용성, VISUAL C++ 6.0 완벽 가이드, 영진출판사, 2000. 1.
 [10] 이상엽, VISUAL C++ Programming Bible, 영진출판사, 1999.

저 자 소 개



서인호 (徐寅豪)
 1973년 3월3일생. 1998년 한국해양대학교 제어계측공학과졸. 2001년 한국해양대학교 대학원 제어계측공학과졸(공학석사). 당학 회회원. 2001년-현재 한국과학기술원 인 공위성연구센터 연구원.



유영호 (劉永昊)
 1951년 10월 10일생. 1974년 한국해양대 학교 기관학과졸업
 1990년 한국해양대학교 대학원(공학박사), 당 학회 재무이사. 1991년-현재 한국해양 대학교 기계·정보공학부 부교수