工學碩士 學位論文

마이크로 프로세서를 이용한 인터넷 원격감시제어 시스템에 관한 연구

A Study on the Internet Control and Monitoring System Using a Microprocessor Embedded Controller

指導敎授 劉 永 昊

2001년 2월

韓國海洋大學校 大學院

制御計測工學科

徐 寅 豪

목 차

A Study on the Internet control and monitoring system using a microprocessor embedded controller

ABSTRACT

The remote control system using the Internet is very useful for the area where it is difficult to approach due to the limitation of time, space and location. The previous conventional remote control equipments have a difficulty in applying to the LAN environment, especially Ethernet environment transmission scheme. because they use asynchronous communication like RS232/422A/485. Therefore such equipments need interconnecting device between the asynchronous method and CSMA/CD.

In this paper, a remote monitoring and controlling system accessible in the Internet is designed and implemented.

The developed system consists of monitoring system that operates under the Windows operating system and microprocessor system(80C196KC) mounting Ethernet NIC compatible with NE2000. The protocol used in this system is UDP/IP. A set of experiments is carried out to verify the effectiveness of the developed system.

제 1 장 서 론

1.1 연구 배경

그동안 산업현장에서는 생산력 증대와 품질향상을 위해서 급속히 발전하고 있는 컴퓨터 및 통신 기술 수용에 많은 노력을 기울여 공장 자동화를 이룩하였다[1]. 이러한 자동화 시스템에서는 기기들 사이에 원활한 데이터의 전달이 중요하다. 이를 지원하는 도구가 컴퓨터 네트워크이며 그림 1.1에 예를 나타내었다[16].

기존의 컴퓨터를 이용한 분산 제어시스템의 하위 레벨에서는 RS232C/422A/485 등 자체적인 네트워크를 형성, 제어 감시 데이터를 컴퓨터로 수집하고 상위 레벨에서는 Ethernet 등의 망을 이용하여 데이터 통신하는 시스템을 구성하고 있다. 그러나 최근에는 자동화 시스템이 고속, 복잡, 초정밀화 되어감에 따라 호스트 중심의 수직구조를 가진 하위 레벨 네트워크에서는 배선이 복잡해져 자동화 시스템을 유지/보수 하는데 많은 경비부담이 발생하고 있다[2],[4]. 또한 회사마다 전송방식이 다르므로 이기종의 자동화 장비들 간의 통신에 어려움이 있으며 RS232C/422A/485와 같은 비동기 통신 방식은 데이터 전송 속도와 시스템을 구성하는 스테이션의 수가 제한되어 있다. 그러므로 원거리 통신망에는 적합하지 않으며 원격제어가 불가능하다[1].

따라서 하위 레벨에서도 기존의 Ethernet 망과 표준 프로토콜을 이용하면 위의 문제점을 보완할 수 있게 된다.

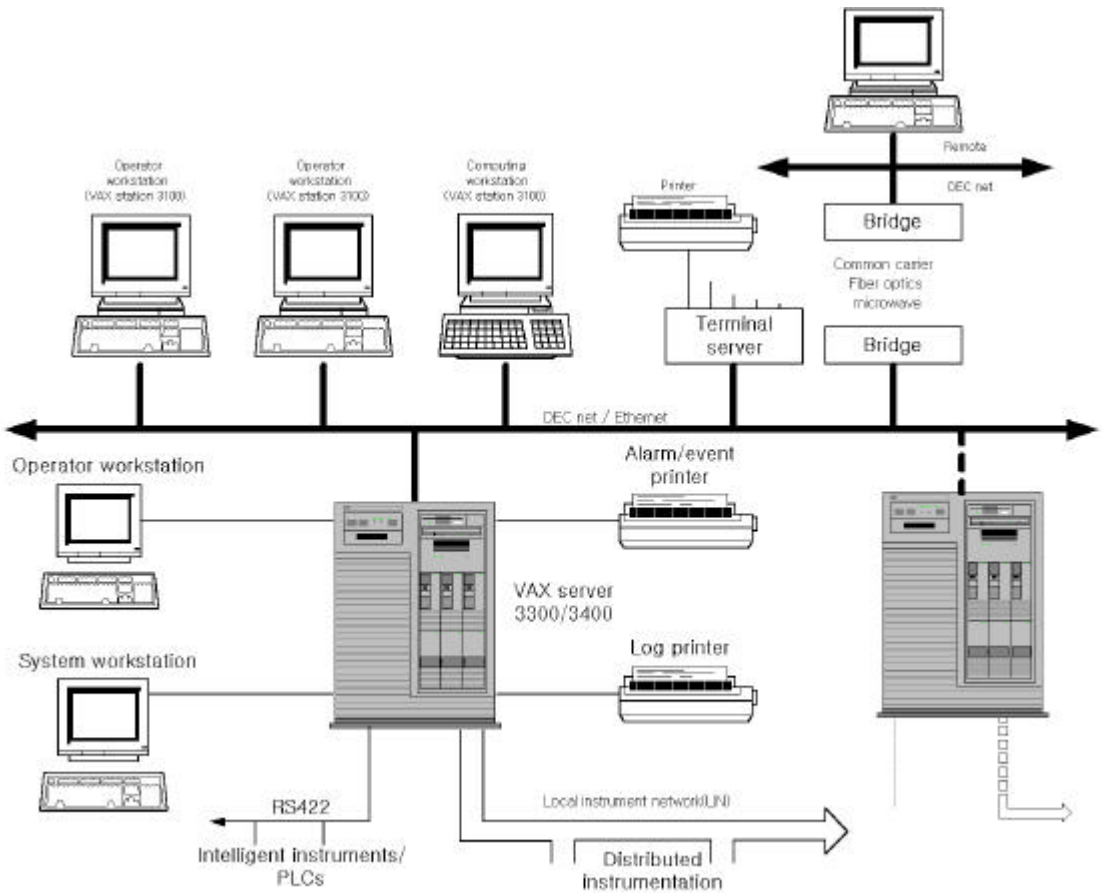

그림 1.1 컴퓨터 네트워크

Fig. 1.1 Computer network

1.2 선행 연구

생산 자동화 설비를 구축함에 있어서 기기들 간의 정보 교환을 위한 실시간 네트워크는 중요한 구성 요소 중의 하나이다. 1980년대에 미국의 보잉사와 G/E사를 중심으로 상호 호환성이 없는 이기종의 자동화 장비들 간의 통신을 위한 표준화된 네트워크 시스템으로 MAP(Manufacturing Automation Protocol)이 개발되었다. 그러나 자동화 네트웍 시스템의 연결 대상이 점차 공정 하부 기기로 내려감에 따라 OSI(Open System Interconnection) 표준모델에서 제시하는 7계층 구조를 모두 가지고 있는 MAP을 채용하기에는 실시간 통신에 여러 가지 문제점이 대두되었고 이에 물리층, 데이터링크층, 응용층의 3계층으로만 이루어진 mini-MAP이 개발되었다. 한편 유럽을 중심으로 제어기기들 간의 통신을 위한 비교적 저가이고 통신에 필요한 최소한도의 기능을 수행하는 단순한 구조를 가지면서도 빠른 응답시간으로 실시간 처리에 적용될 수 있는 네트워크 시스템으로서 필드버스 네트워크가 개발되었다. 필드버스 네트워크는 CIM (Computer Integrated Manufacturing)의 계층구조상 가장 하위 계층인 생산 필드의 분산 공정 제어에 있어서 센서, 공정제어기, 액츄에이터, 계측기기, 스위치, 분석기 등과 같은 제어기기들 간의 통신 네트워크 시스템으로서 프로토콜로는 SERCOS, CAN, WorldFTP, Profibus 등이 있으며 이들은 OSI의 7계층에서 물리계층, 데이터 링크 계층, 응용 계층의 3계층과 새롭게 설정된 사용자 계층(user layer)으로 구성되어 있다[7],[8].

최근에는 LAN이 근거리 통신망에서 가장 널리 사용되고 있으며 인터넷의 급속한 보급으로 인하여 TCP/IP에 대한 중요성이 크게 부

각되고 있다. 이러한 TCP/IP 프로토콜은 수년간에 걸쳐 여러 연구 기관과 대학에서 연구되고 개량되어 매우 안정적이며 수많은 응용 프로그램을 가지고 있어 자동화 시스템과 인터넷을 통한 감시제어 시스템에서 유용도가 급증되고 있는 형편이다[5],[6].

1.3 연구 내용

본 논문에서는 연구 배경에서 다룬 RS232C/422A/485를 이용한 통신 시스템의 문제점을 보완하기 위하여 마이크로프로세서 시스템에서 Ethernet 통신을 행하였고 기존의 인터넷 망을 이용하여 제어대상을 원격지에서 감시하고 제어할 수 있는 인터넷 원격감시제어 시스템을 설계하고 구현하였다.

마이크로프로세서 시스템에서는 Ethernet을 구현하기가 어려우므로 기존에 제품으로 나오는 NIC를 사용하여 마이크로프로세서 시스템과 인터페이스 시켰으며 Ethernet 인터페이스로 NE2000과 호환의 ISA 버스용 NIC를 사용하였다.

이 시스템은 인텔 80C196KC 프로세서와 NIC로 구성된 마이크로프로세서 시스템과 윈도우즈 운영체제에서 동작하는 모니터링 시스템으로 구성되어 있으며 통신 프로토콜로는 UDP/IP를 사용하였다.

본 논문의 내용은 다음과 같이 구성된다. 2장에서는 Ethernet 통신을 위한 이론적 배경이 되는 프로토콜의 구조와 각 프로토콜을 소개한다. 3장에서는 마이크로프로세서에서 Ethernet 통신을 행하기 위해서 사용한 NIC의 구조와 사용방법에 대하여 설명한다. 4장에서는 구현된 인터넷 원격감시제어 시스템의 하드웨어와 소프트웨어에 대하여 설명한다. 5장에서는 인터넷 상에서의 실시간 모니터링을 통

하여 성능을 평가한다. 6장에서는 앞서 얻은 결과를 바탕으로 결론을 내린다.

제 2 장 UDP/IP 프로토콜

2.1 프로토콜의 구조

현재 프로토콜의 표준화는 국제표준화기구(ISO)와 국제전기통신연합의 전기통신표준화 센터(ITU-T)를 중심으로 진행되고 있으며 LAN에 관해서는 미국전기전자기술자협회(IEEE)의 802위원회가 대표적인 기구이다.

통신 프로토콜은 알기 쉽게 계층화되어 있으며 국제표준화기구가 규정한 OSI의 기본 참조 모델이 가장 대표적인 것이다. 이것을 그림 2.1에 나타내었으며 기술적인 독립성, 기능적인 공통성이라는 관점에서 통신 기능을 7개로 분할하고 그것들이 동작하는 순서에 따라 계층화하고 있으며 크게는 물리층, 데이터 링크층, 네트워크층, 트랜스포트층을 합한 하위층과 세션층, 프리젠테이션층, 애플리케이션층을 합한 상위층으로 나눌 수 있다[10],[11],[12].

본 논문에서는 시스템 구현에 이용된 Ethernet, ARP, IP, UDP에 대해서 설명한다.

2.2 Ethernet

2.2.1 Ethernet과 IEEE802.3

Ethernet은 실제 통신에 있어서 최하위의 프로토콜로서 OSI 참조 모델의 물리층 및 데이터 링크층 하반부의 MAC층의 프로토콜이며 그 규격에는 Ethernet Ver.2.0(DIX 사양)과 IEEE802.3 이라는 2가지

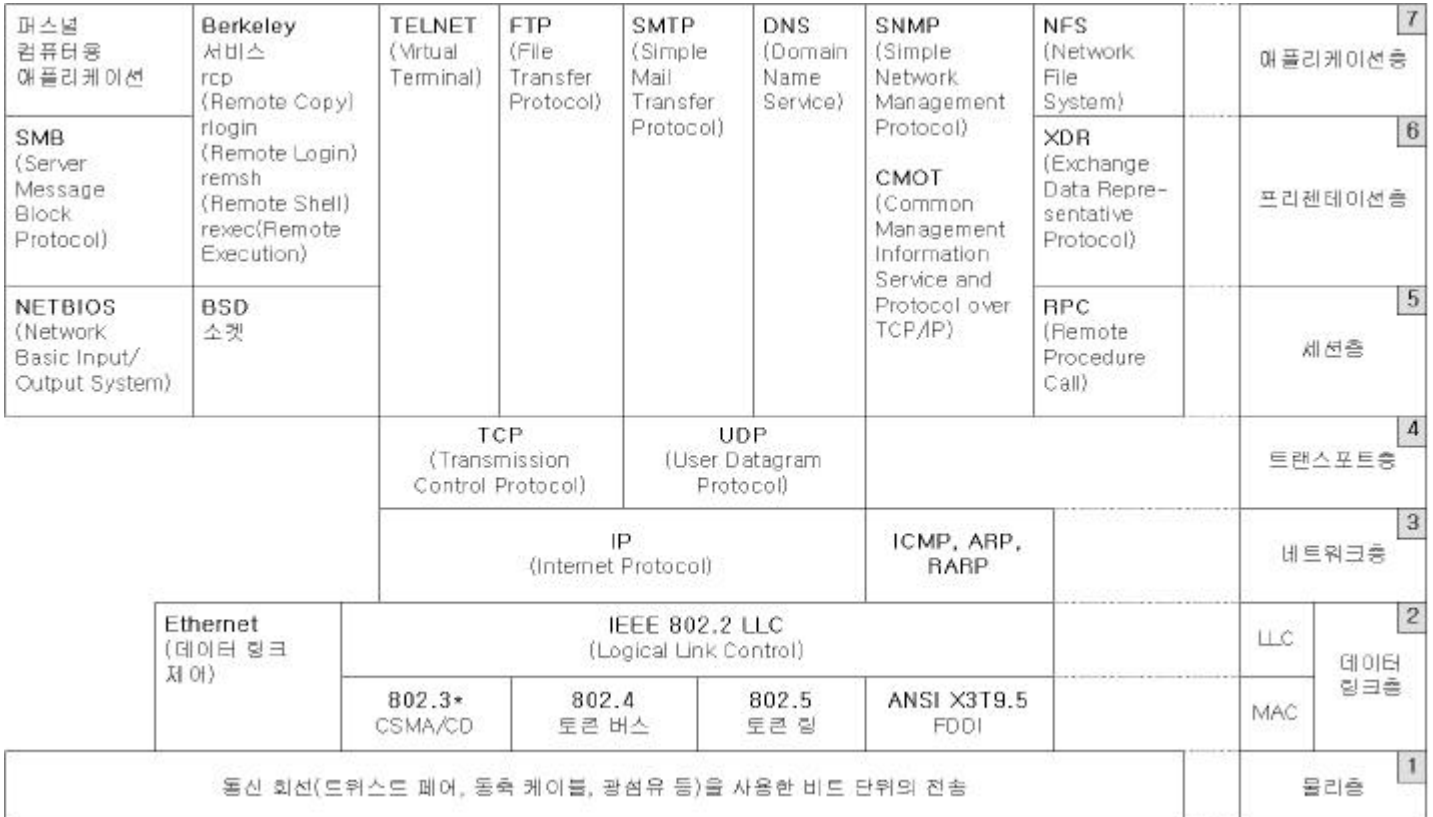

그림 2.1 OSI 기본 참조 모델

Fig. 2.1 OSI reference model

표준이 존재한다. 이 2개의 표준을 총칭하여 Ethernet이라 부르며 현재는 DIX사양이 주류를 이루고 있으므로 본 논문에서는 DIX-Ethernet에 대해서만 다루었다.

Ethernet은 CSMA/CD(Carrier Sense Multiple Access with Collision Detection)라는 전송제어 방식을 사용한다. 이것은 버스형 전송로를 복수의 노드가 공용하기 위한 방식으로써 그 특징을 보면 첫째는 LAN에 송신된 데이터 신호는 그 LAN에 접속되어 있는 모든 노드에 함께 도달한다는 것이고 두 번째는 하나의 LAN 내에서는 동시에 2개 이상의 노드가 데이터 신호의 송신을 실행할 수 없다는 것이다. 현재는 LAN 컨트롤러 칩이 CSMA/CD 부분을 담당하고 있다.

2.2.2 Ethernet 패킷의 구조

그림 2.2에 Ethernet 프레임을 나타내었으며 그림에서 숫자는 필드의 크기이고 단위는 바이트를 나타낸다. 프리앰블 필드는 동기를 맞추기 위한 것이고 CRC 필드는 에러체크로써 LAN 컨트롤러 칩이 담당하고 있으므로 고려하지 않아도 된다. 수신처 어드레스 필드와 송신원 어드레스 필드는 MAC 어드레스라고 부른다.

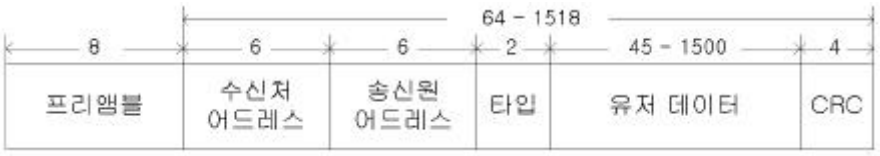

그림 2.2 Ethernet의 패킷의 구조
Fig. 2.2 Ethernet packet format

전세계 Ethernet 인터페이스가 고유한 MAC 어드레스를 가지고 있으며 현재는 미국전기전자기술자협회에서 관리하고 있다. 타입 필드는 데이터부의 프로토콜을 식별하는 값으로 IP는 0800H, ARP는 0806H이다.

Ethernet 패킷은 최소 64바이트, 최대 1518바이트까지이며 만약 64바이트 미만인 경우에는 어떤 데이터(일반적으로 0)를 추가하는데 이 조작을 패딩이라고 한다.

2.2.3 Ethernet 수신 처리

Ethernet 인터페이스는 다음과 같은 경우에만 수신처리를 해야한다.

- 자신의 MAC 어드레스와 일치되는 경우
- 브로드캐스트 어드레스와 일치되는 경우
- 자신이 수신하는 멀티캐스트 어드레스와 일치하는 경우

위의 경우 패킷을 수신하여 CRC가 맞지 않은 경우에는 패킷을 파기하며 이상이 없을 때는 패킷을 버퍼에 저장한다.

2.2.4 Ethernet 송신 처리

송신시에는 상위층의 정보를 조합한 후 Ethernet 패킷을 구성하여 미디어 상에 데이터가 흐르고 있지 않다는 것을 확인한 다음 송신한다. 다른 노드와 충돌이 검출되지 않으면 송신이 성공된 것으로 처리하고 만약 충돌이 검출된 경우에는 백오프 알고리즘에 따라 재송신을 시도하며 그래도 충돌이 일어났을 경우에는 송신 실패로 처

리하여 송신을 중단한다.

2.3 ARP(Address Resolution Protocol)

2.3.1 MAC 어드레스와 IP 어드레스

Ethernet 인터페이스는 고유의 MAC 어드레스를 가지고 있다. 모두 48비트로 구성되어 있으며 8비트씩 나누어 표기한다. 상위 24비트는 제조한 벤더에 할당된 벤더 코드이고 하위 24비트는 노드 번호로써 벤더가 임의적으로 할당할 수 있다. 또한 모든 비트가 1인 브로드캐스트 어드레스나 여러 Ethernet 인터페이스에게 동일 데이터를 한꺼번에 송신하기 위한 멀티캐스트 어드레스가 정의되고 있다.

그러나 일반적으로 네트워크에는 동일한 인터페이스 방식의 LAN만이 존재하지 않고 여러 인터페이스가 복합적으로 존재한다. 따라서 이러한 여러 혼재된 인터페이스를 총괄하는 계층이 필요하게 된다. 이 계층이 IP 층이고 여기서 일관되게 활용하는 어드레스 체계가 IP 어드레스이다. IP 어드레스는 32비트로 구성되어 있으며 8비트씩 나누어 표기한다. 네트워크 어드레스와 호스트 어드레스에 따라 5개의 클래스로 구분되어 있으며 클래스에 의해서 하나의 네트워크에 접속할 수 있는 호스트 수가 정해진다.

애플리케이션 계층에서는 IP 어드레스로 경로를 설정하므로 네트워크로 실제 데이터를 전송할 때는 IP 어드레스를 MAC 어드레스로 변화하는 과정이 필요하게 된다.

2.3.2 ARP의 역할

하나의 Ethernet 망에서는 MAC 어드레스로 상대방과 통신을 하게 된다. 그러나 응용에서는 상대방의 IP 어드레스로 통신요구가 발생되므로 거기에 대응하는 MAC 어드레스를 알아야 실제로 데이터의 송수신이 가능하게 된다. 이 일을 하는 프로토콜이 ARP이며 동작하는 순서는 다음과 같다.

- Ethernet 브로드캐스트로 ARP 요구 패킷 송신
- 모든 노드에서 ARP 패킷을 수신
- 해당 노드는 발신지 Ethernet 어드레스에 대해 ARP 응답 패킷을 송신
- ARP 응답 패킷을 수신하여 Ethernet 어드레스 취득

2.3.3 ARP 패킷의 구조

그림 2.3은 ARP 패킷의 구조를 나타낸다. 하드웨어 어드레스 타입 필드는 Ethernet의 종류를 나타내는 것으로 DIX-Ethernet의 경우는 1이 들어간다. 프로토콜 어드레스 타입 필드는 상위 프로토콜을 나타내며 IP의 경우는 0800H이다. 하드웨어 어드레스 길이와 프로토콜 어드레스 필드는 각각 6과 4가 들어간다. 코드 필드는 패킷의 종류를 나타내는 것으로써 ARP 요구인 경우는 1, ARP 응답인 경우는 2를 넣는다. ARP 요구 패킷의 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. 또한 ARP 응답 패킷의

경우에는 상대방의 MAC 어드레스와 IP 어드레스를 알고 있으므로 ARP 요구 패킷과 같은 형태로 구성하여 송신한다.

ARP의 경우 패킷의 길이가 64바이트 미만이므로 남는 부분은 0으로 패딩한다.

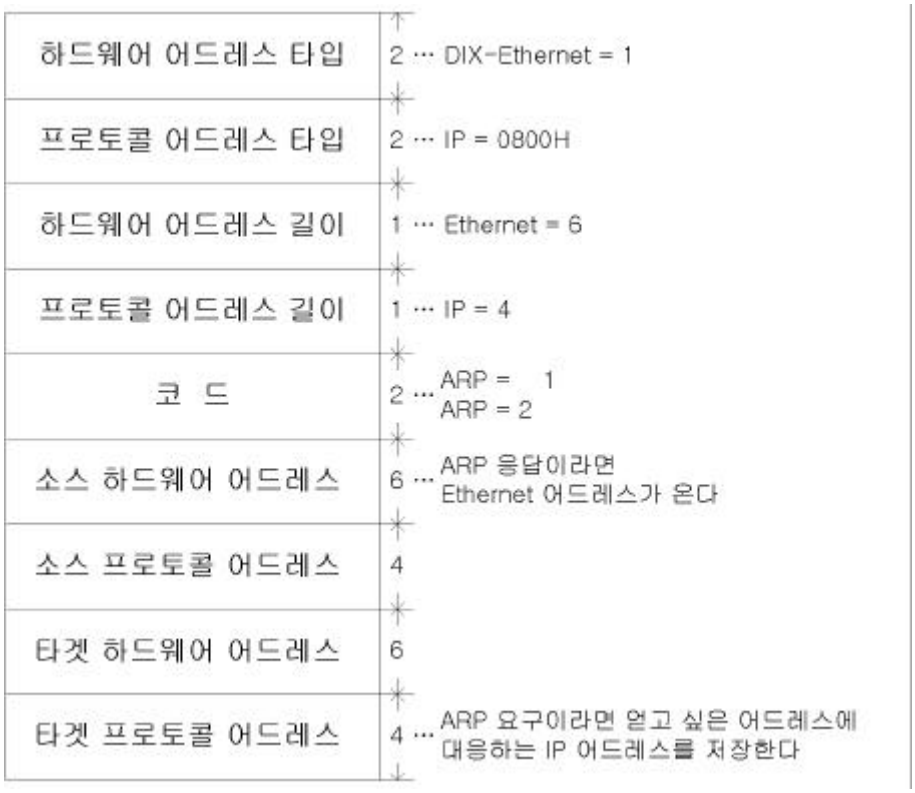

그림 2.3 ARP **패킷의 구조**

Fig. 2.3 ARP packet format

2.4 IP(Internet Protocol)

2.4.1 IP의 역할

IP는 호스트간 또는 호스트와 게이트웨이 간에 통신을 실현하여 데이터그램을 목적하는 곳까지 전송해주는 프로토콜이다. 이 프로토콜은 수신지 IP 어드레스로 경로를 선택하는 경로 선택 기능과 데이터그램의 크기가 MTU(Maximum Transfer Unit)보다 클 경우 몇 개의 프래그먼트로 분할하여 송신하는 프래그먼테이션 기능이 있다.

2.4.2 IP 패킷의 구조

그림 2.4는 IP 패킷의 구조이며 크게 IP 헤더부와 데이터부로 나눌 수 있다. 버전 필드는 IP 프로토콜의 버전(Version)을 나타낸다. 현재 버전이 IPv4 이므로 버전 필드의 내용은 4이다. 헤더길이 필드는 IP 헤더 길이를 4로 나눈 값을 나타내며 헤더 길이가 20 이므로 4로 나누면 5가 된다. TOS(Type Of Service) 필드는 데이터그램을 어떻게 처리할 것인가를 정의한 필드로써 일반적으로 0을 넣는다. 토탈길이 필드는 IP 헤더와 데이터의 총길이로써 16비트이므로 65535 바이트가 최대가 된다. 하나의 데이터 크기가 MTU보다 클 경우에는 몇 개의 데이터그램으로 나누어서 전송하고 이를 다시 재조립해야 하는데 이를 위해 필요한 것이 시퀀스 번호와 프래그먼트 정보이다. 시퀀스 번호는 데이터그램이 쪼개질 경우 어느 데이터그램으로부터 쪼개진 것인지를 표시하고 프래그먼트 정보는 현재의 데 이터그램이 전체 데이터그램의 어느 위치에 있는 것인지를 표시

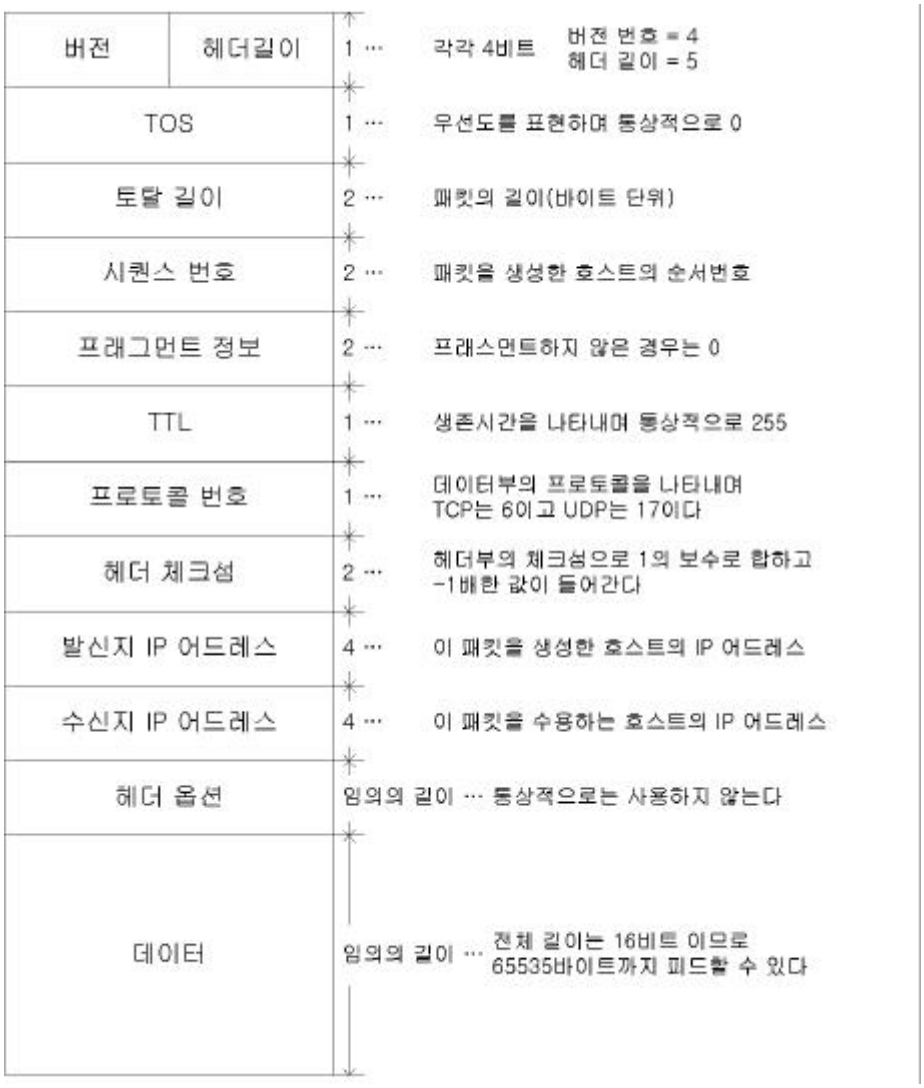

그림 2.4 IP 패킷의 구조

Fig. 2.4 IP packet format

하며 프래그먼트가 없을 경우에 프래그먼트 정보는 0이 된다.

TTL(Time To Live) 필드는 데이터그램이 통신망에서 얼마나 오랫동안 남아 있을 수 있는가를 나타내는 필드로써 게이트웨이나 호스트는 소요된 시간만큼 감소시키다가 0이 되면 데이터그램을 통신망에서 제거하며 처음에는 255의 값을 가진다. 프로토콜 번호 필드는 상위 프로토콜을 나타내는 것으로써 TCP는 6, UDP는 17이다.

헤더 체크섬 필드는 에러제어 기능을 담당하는 필드로써 헤더 체크섬 필드를 0으로 초기화한 후 전체 헤더를 16비트 단위로 1의 보수로 가산한 다음 합계의 비트를 반전하여 -1배 한 값을 체크섬의 값으로 한다. 헤더 옵션 필드는 일반적으로 사용하지 않는다.

2.4.3 IP 수신 처리

데이터링크층에서 IP 패킷을 받아서 헤더의 체크 섬을 계산한 후 에러 유무를 확인하여 체크 섬이 맞지 않을 경우에는 패킷을 파기한다. 발신지 IP 어드레스와 수신지 IP 어드레스를 체크하여 맞는 경우에는 IP 헤더의 프로토콜 번호를 확인하여 상위층으로 패스하며 프로토콜 번호가 맞지 않을 경우에는 패킷을 파기한다.

2.4.4 IP 송신 처리

상위층에서 데이터, 데이터 길이, 수신지 IP 어드레스, 상위층 프로토콜 번호가 패스되면 이 정보에 따라 헤더 체크 섬을 계산한 후 IP 헤더를 구성하여 하위층으로 패킷을 패스한다.

2.5 UDP(User Datagram Protocol)

2.5.1 스트림 통신과 데이터그램 통신

네트워크에서 자료의 패킷 교환 방식은 전달 형태에 따라 연결형인 스트림 통신과 비연결형인 데이터그램 통신이 있다.

스트림 통신에서는 두 호스트 간에 자료를 전송하기 전에 초기화 작업을 거쳐 두 개의 프로세스 사이에 실제로 존재하지 않는 가상의 통신로를 확립하고 이 통신로를 통하여 자료를 전송한다. 그러므로 가상의 통신로가 끊어지면 자료를 전송할 수가 없게 되지만 신뢰성 높은 통신을 하게 된다. 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 따라서 트래픽(traffic)에 따라 도착 순서가 바뀔 수도 있다. 반면에 가상의 통신로가 없으므로 연결 설정등 오버헤드가 작으며 프로토콜이 간단하여 사용하기 편하다. 이러한 스트림 통신과 데이터그램 통신 가운데 어떤 방식을 제어 및 모니터링 시스템에서 쓸 것인가에 대한 논쟁은 계속 되고 있으며 본 논문에서는 프로토콜이 시스템에 미치는 오버헤드를 고려하여 데이터그램 통신을 사용하였으며 다음에 그 장점을 나열하였다[9].

- 융통성 : 대부분의 분산체제에서와 같이 짧은 컨트롤 및 모니터링 결과가 트래픽(traffic)의 주류를 이루고 생성된 모니터링 데이터의 패킷이 손실되어도 시스템에 큰 영향을 미치지 않는 시

스템에서는 데이터그램 통신이 스트림 통신에 비해 적합하다.

- 자동 회복 : 네트워크상에서 어떤 노드가 작동이 되지 않더라도 그 노드를 거치던 데이터그램은 다른 경로를 통하여 전달될 수 있으므로 전달의 견고성이 이루어질 수 있다.
- 간단한 통신 프로토콜 : 데이터그램 통신은 스트림 통신에 비하여 복잡하지 않으므로 마이크로프로세서에서 구현하기 쉽다.

2.5.2 UDP의 역할

UDP는 IP의 기능이 애플리케이션에서 사용되도록 하는 프로토콜로써 애플리케이션에는 여러 종류가 있으므로 이를 구별하기 위해서 포트 번호를 사용한다. 따라서 개개의 패킷에 포트 번호를 부가하여 어느 패킷이 어느 애플리케이션인가를 구별하고 있다. 포트 번호를 사용할 때는 주의를 해야 하는데 지정된(well known) 포트로써 DNS는 53, DHCP는 67 등 0에서 511번까지는 예약되어 있으므로 이용자는 1000번 이상을 사용하도록 권하고 있다.

2.5.3 UDP 패킷의 구조

그림 2.5에 UDP 패킷의 구조를 나타내었다. UDP 헤더부와 데이터부로 구성되어 있으며 그림에서도 알 수 있듯이 UDP 헤더부는 단순한 구성이므로 구현하기 간단하다.

발신지 포트 번호와 수신지 포트 번호 필드는 이용자가 정한 번호가 들어가며 일반적으로 서로 같게 설정된다. 패킷 길이 필드는 UDP 헤더부와 데이터부의 길이의 합이 들어가며 체크 섬은 IP 헤

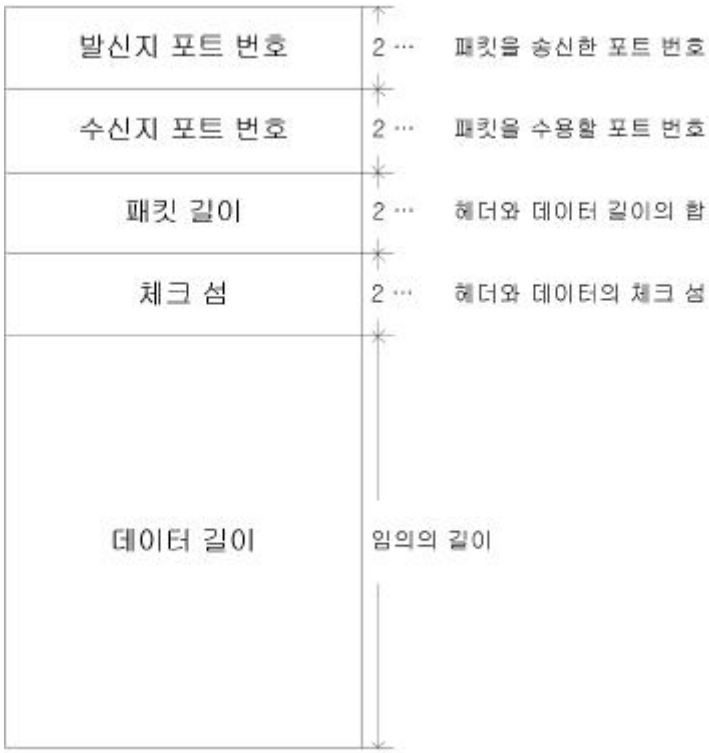

그림 2.5 UDP **패킷의 구조**

Fig 2.5 UDP packet format

더와 같은 방법으로 1의 보수로 합계를 계산한다. 한가지 다른 점은 체크섬이 0 또는 0FFFFH가 되었을 때 UDP는 0FFFFH를 저장하며 IP는 그대로 0 또는 0FFFFH로 저장한다. 이렇게 하는 이유는 UDP에서는 체크 섬이 옵션이므로 생략할 때는 0의 값이 들어가기 때문이며 생략하면 신뢰성은 떨어지게 된다. 데이터부는 실제로 전송하고자 하는 데이터로써 아스키 값으로 되어 있다.

2.5.4 UDP 수신 처리

체크 섬을 계산하여 패킷이 파괴되지 않았다는 것을 확인하며 체크 섬 필드가 0일 경우에는 생략되었으므로 고려하지 않는다. IP 어드레스와 포트 번호의 조를 소켓이라 하는데 UDP와 IP의 헤더에서 소켓을 구성하여 일치하는 애플리케이션이 있다면 그 애플리케이션을 호출하여 데이터를 처리한다.

2.5.5 UDP 송신 처리

애플리케이션에서 패스되는 소켓의 정보를 바탕으로 발신지와 수신지 포트 번호, 데이터 길이를 만들고 체크 섬을 계산한 후 수신지 소켓의 IP 어드레스를 수신지 IP 어드레스로 지정하고 IP 출력 루틴을 호출한다.

2.6 UDP/IP 프로토콜을 이용한 패킷 송신

그림 2.6은 UDP/IP 프로토콜을 이용한 패킷의 송신 과정을 그림으

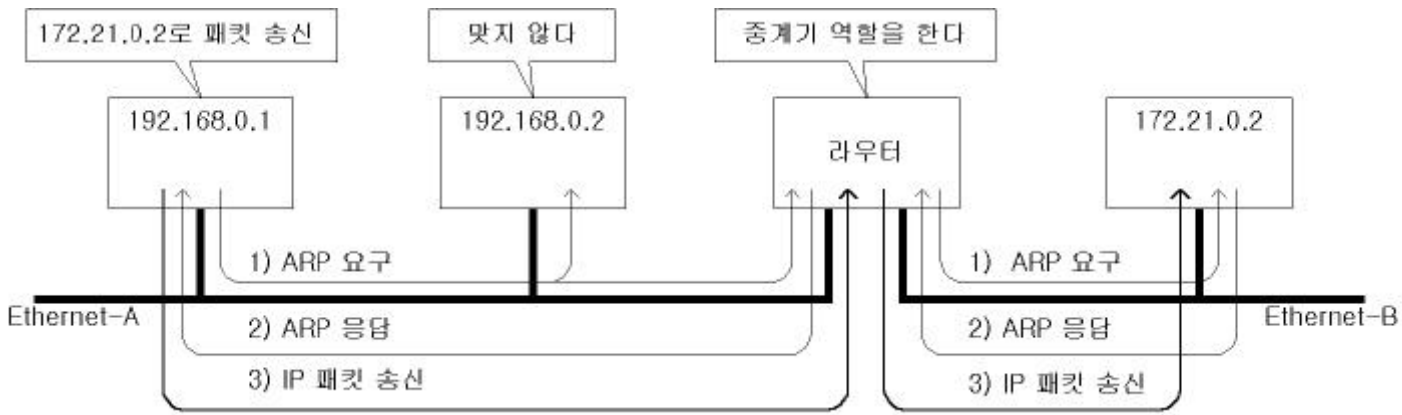

그림 2.6 UDP/IP**의 개념도**

Fig. 2.6 Packet transimision using the UDP/IP

로 나타내었다. 그림에서 보면 192.168.0.1의 IP를 가진 호스트가 172.21.0.2의 IP를 가진 호스트로 데이터를 전송하려고 한다. 그러나 MAC 어드레스를 모르므로 통신을 할 수가 없다. 또한 그림 상에서는 수신지가 동일 네트워크 상에 존재하지 않으므로 중계기 역할을 하는 라우터를 거치게 된다. 아래에 통신의 순서를 나타내었다.

- ARP 요구를 브로드캐스트
- ARP 응답 패킷을 수신
- IP 패킷을 송신
- 경로배정(Routing by Router)
- ARP에 의해 목적지 MAC 어드레스를 알아냄
- 목적지로 패킷 전송

제 3 장 NIC(Network Interface Controller)

3.1 NIC의 구조

본 연구에서는 Ethernet 인터페이스로써 16비트 마이크로프로세서에 장착할 수 있고 시중에서 쉽게 구할 수 있으며 저가인 NE2000 호환의 ISA 버스용 NIC를 사용하였다.

그림 3.1에 NIC의 블록도를 나타내었다. NIC는 크게 Ethernet 컨트롤러인 RTL8019AS와 송수신 버퍼 RAM, 시리얼 EEPROM, ISA 버스 인터페이스 회로, 미디어 인터페이스 회로로 구성되어 있다[25].

또한 마이크로프로세서와의 접속에 ISA 버스의 I/O 공간과 인터럽트를 사용한다.

3.2 시리얼 EEPROM

NIC의 MAC 어드레스, I/O 어드레스와 IRQ의 설정은 시리얼 EEPROM에 저장된다. NIC가 리셋되면 시리얼 EEPROM에서 설정값을 판독하여 초기화한다.

그림 3.2에 NIC의 메모리 맵이 나타나 있으며 시리얼 EEPROM의 메모리 영역은 0000H에서 001FH까지이다.

3.3 버퍼 RAM

그림 3.2에 NIC의 메모리 맵을 나타내었다. 4000H에서 7FFFH까지 송수신 버퍼용 RAM이 매핑되어 있어 CPU에서 판독 및 기록이 가

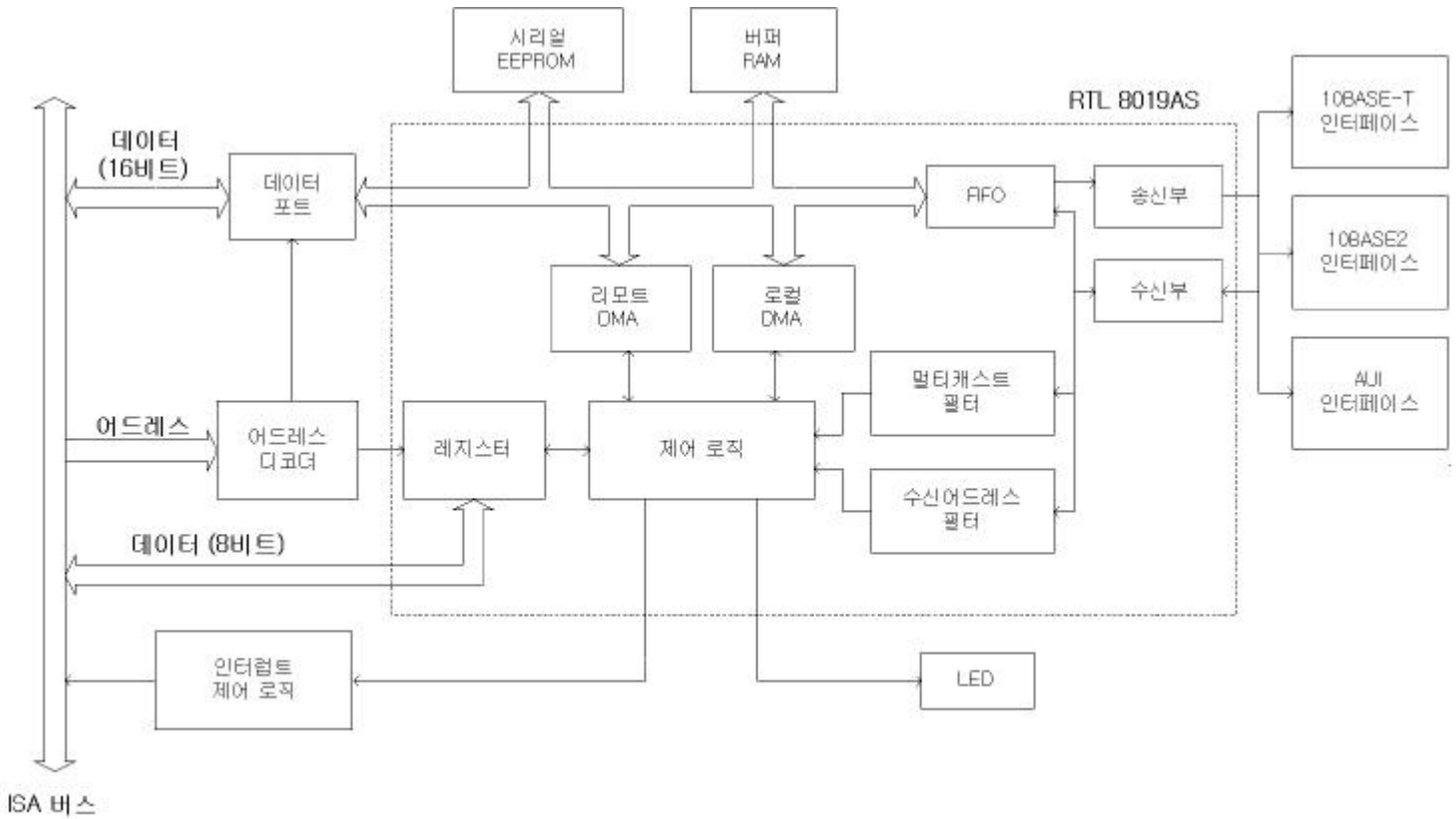

그림 3.1 NIC의 블록도

Fig. 3.1 NIC block diagram

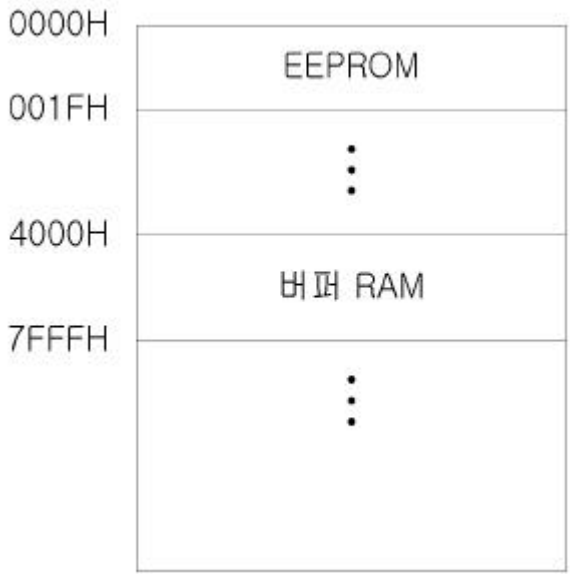

그림 3.2 **버퍼 메모리 맵**

Fig. 3.2 Buffer memory map

능하며 송수신 버퍼 RAM에 접근하기 위해서 RTL8019AS에 내장된 로컬 DMA와 리모트 DMA를 사용한다.

로컬 DMA는 미디어에 대한 송수신에 사용된다. 따라서 수신된 패킷은 로컬 DMA에 의해 자동적으로 버퍼 RMA에 저장되며 송신 패킷은 송신 명령에 의하여 로컬 DMA에 의하여 자동으로 송신처리가 된다.

리모트 DMA는 CPU와 버퍼 RAM 사이에 데이터를 교환할 때 사용하며 CPU가 데이터 포트에 데이터를 기록하면 리모트 DMA에 의해 버퍼 RAM에 기록된다. 이때 버퍼 RAM에 대한 기록 어드레스가 자동으로 증가되어 연속 영역에 대한 고속 액세스가 가능하게 된다. 버퍼 RAM의 내용을 CPU가 판독할 때는 반대로 동작하게 된다.

3.4 I/O 포트와 인터럽트

ISA 버스의 I/O 공간은 000H에서 03FH까지이다. NIC는 그 중에서 32바이트 분량을 찾이하고 있으며 개시 어드레스는 240H, 280H, 300H, 340H, 360H 등 16개가 있다. 인터럽트로써 8개의 IRQ 라인이 있으며 그 중 1개의 IRQ만 사용할 수 있다. 본 연구에서는 개시 어드레스가 300H이고 IRQ 9번을 사용하고 있다.

그림 3.3에 NIC의 I/O 공간을 나타내었다. 전반의 16바이트가 RTL8019AS에 할당되어 있으며 후반의 16바이트 가운데 전반 8바이트는 버퍼 RAM과의 데이터 교환을 위한 데이터 포트이고 후반의 8바이트는 리셋 포트로써 NIC의 소프트웨어 리셋에 사용된다. 이중 데이터 포트는 16비트로 나머지는 8비트로 액세스된다.

그림 3.3 NIC**의** I/O **맵**

Fig. 3.3 NIC I/O map

3.5 RTL8019AS

RTL8019AS는 RealTek사의 Ethernet 컨트롤러로써 데이터 링크층의 기능을 담당하고 있다.

3.5.1 레지스터

RTL8019AS에는 약 40개의 레지스터와 4개의 페이지가 있다. 통상적으로는 16개의 레지스터가 보이며 페이지의 교체에 의하여 레지스터가 달라지게 된다. 일반적으로 페이지 0와 페이지 2의 레지스터를 사용하며 페이지 1은 MAC 어드레스와 멀티케스트 어드레스를 저장하기 위한 레지스터이고 페이지 3은 예약되어 있으므로 통상적으로 사용하지 않는다.

3.5.2 링 버퍼

RTL8019AS는 수신버퍼를 페이지 단위(256 바이트)로 나누어 링버퍼를 만든다. 이렇게 되면 수신버퍼가 총 59페이지가 되어 페이지 0에서부터 수신을 시작하여 페이지 58까지 수신하고 나면 다시 페이지 0로 되돌아가서 수신하게 되므로 유한한 버퍼 영역을 연속적으로 액세스하게 되어 수신된 데이터를 효과적으로 관리할 수 있게 된다. 그림 3.4에 링버퍼의 구조를 나타내었다.

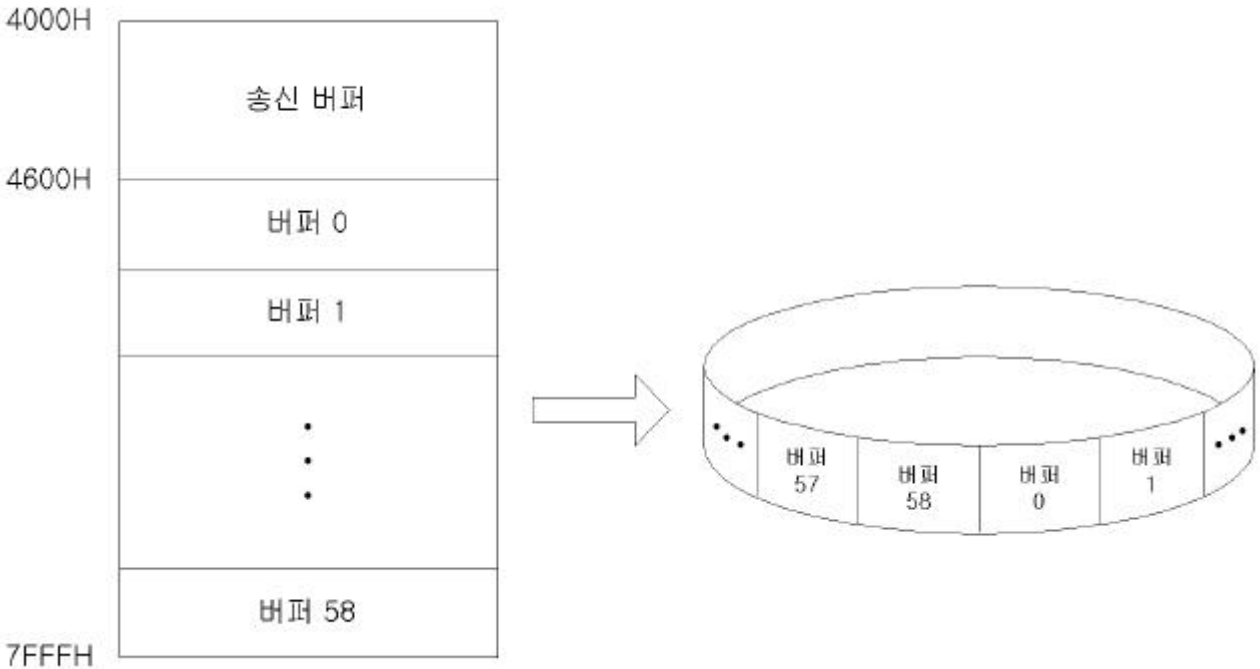

그림 3.4 링 버퍼의 구조

Fig. 3.4 Ring buffer structure

제 4 장 시스템 설계 및 구현

4.1 전체 시스템의 구성

전체 시스템은 마이크로프로세서 시스템과 모니터링 시스템으로 구성되어 있으며 그림 4.1에 전체 시스템의 구성도를 나타내었다.

마이크로프로세서 시스템은 모니터링 시스템으로부터 전송 받은 데이터로 명령을 수행하거나 플랜트의 온도나 압력 등 각종 센서 값들을 인터넷을 경유하여 모니터링 시스템으로 전송하는 일을 하며 모니터링 시스템은 인터넷 상에서의 실시간 모니터링을 통하여 플랜트의 상태를 원격으로 감시하고 제어하는 일을 한다.

4.2 마이크로프로세서 시스템

4.2.1 하드웨어의 설계

사진 4.1에 개발된 마이크로프로세서 시스템의 외형도를 나타내었으며 그림 4.2에 그 구성도를 나타내었다. 전체 시스템은 인텔 80C196KC 프로세서, NE2000 호환의 ISA 버스용 NIC, LCD(4×20), Key와 LED로 구성되어 있다.

프로세서는 NIC를 이용한 데이터 송수신을 통하여 ARP, UDP/IP 프로토콜 기능을 수행하며 ISA 버스용의 NIC와 연결하기 위하여 16비트 모드를 사용하였다. LED는 수신된 데이터에 의한 명령 수행을 알아보기 위한 디지털 접점에 해당한다.

NIC는 데이터 링크층의 기능을 수행하며 프로토콜의 순차적 구현

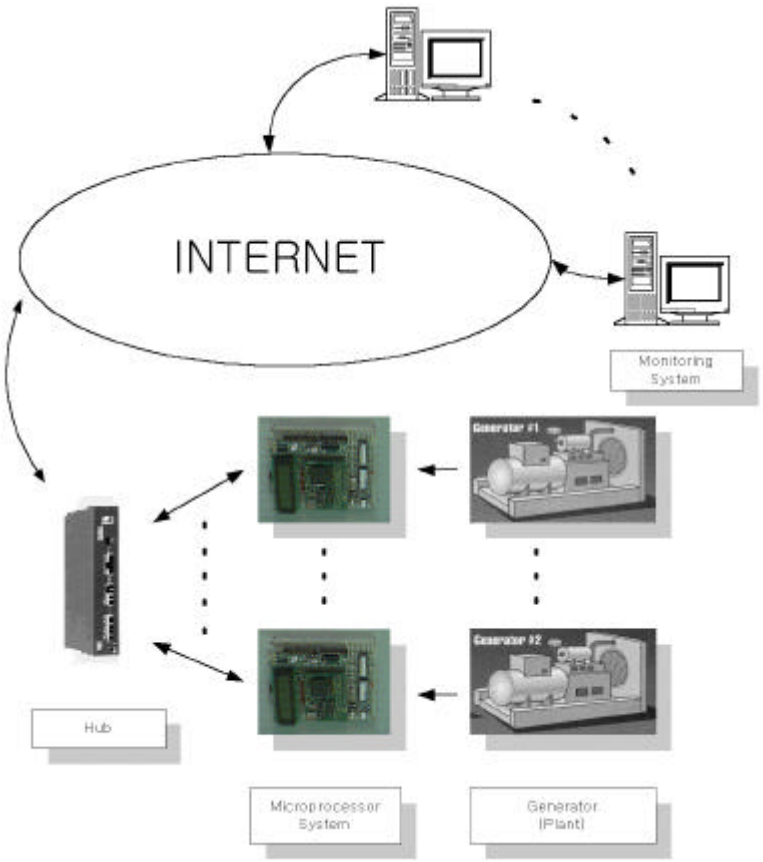

그림 4.1 전체 시스템의 구성도

Fig. 4.1 System diagram

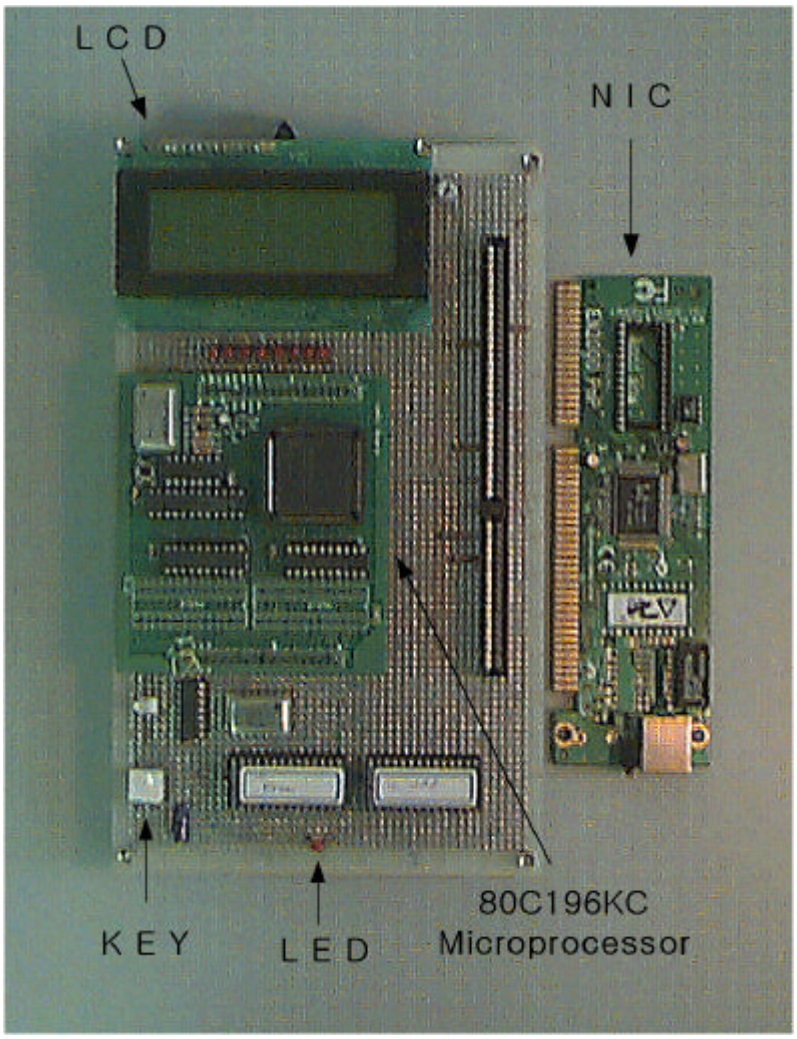

사진 4.1 개발된 마이크로프로세서 시스템의 외형도

Photo. 4.1 Developed microprocessor system

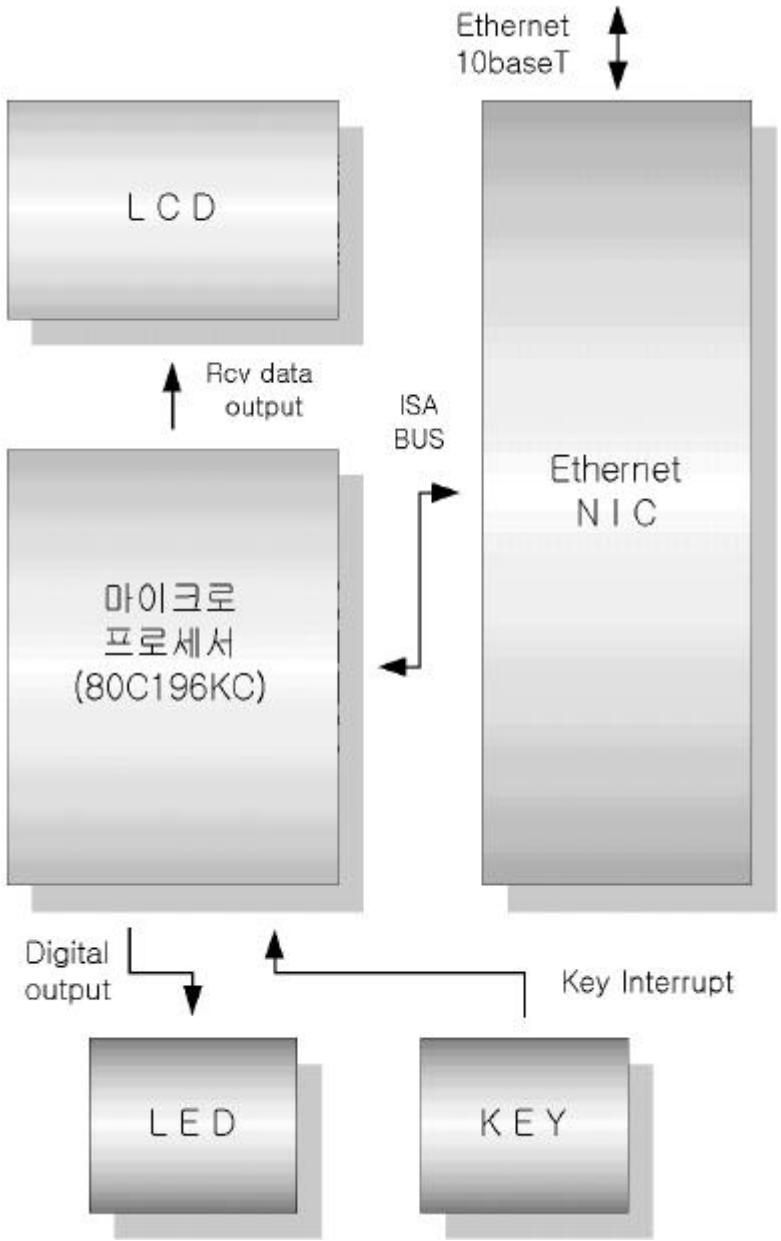

그림 4.2 마이크로프로세서 시스템의 구성도

Fig. 4.2 Block diagram of the microprocessor system

을 알아보기 위하여 LCD 표시장치를 사용하였다. 또한 Key에 의하여 수신된 데이터를 연속적으로 볼 수 있게 된다.

인터럽트는 2개를 사용하였다. NIC가 데이터 수신시 외부인터럽트가 발생하게 되며 Key 입력시 HSI(High Speed Interrupt)가 발생하게 된다.

시스템의 동작을 수신과 송신으로 나누어서 설명하면 다음과 같다.

NIC가 데이터 수신 완료시 프로세서에 인터럽트 신호를 보내면 프로세서는 인터럽트 신호를 체크하여 수신버퍼의 데이터를 읽어와 RAM 영역에 저장한 후 ARP, UDP/IP 프로토콜을 수행하게 되며 Key 입력에 의하여 LCD 표시장치에서 데이터를 확인할 수 있다.

송신시에는 프로세서에서 NIC의 송신버퍼에 데이터를 전송한 후 NIC에 송신명령을 내리면 NIC가 송신 동작을 수행하게 된다.

4.2.2 소프트웨어의 설계

프로그램은 크게 두 부분으로 나누어진다. 첫 번째는 ARP, UDP/IP 프로토콜을 처리하는 부분이고 두 번째는 LCD에 데이터를 보여주는 부분이다. 모두 인터럽트에 의해서 각각의 루틴을 수행하며 하드웨어와 연구의 특성상 다음과 같은 제약을 가진다.

- 마이크로프로세서 시스템에서는 RAM이 32KB 이므로 애플리케이션이 이용하는 영역을 생각하면 충분한 버퍼 영역이 잡히지 않는다. 그래서 Ethernet의 1패킷(1518 바이트)을 저장할 수 있는 최저한의 영역만 이용한다.
- 자발적으로 패킷을 송신하지 않으며 마이크로프로세서 시스템에

서 패킷을 수신했을 때만 패킷을 송신

- 실제 플랜트에 연결하지 않고 시뮬레이션 데이터를 만들어 전송

그림 4.3에 전체 소프트웨어의 구성도를 나타내었다.

프로그램이 시작하면 MAC 어드레서, IP 어드레스, NIC의 송수신 버퍼를 체크하여 초기화를 한다. 그 다음 계속 대기하면서 인터럽트 신호를 기다린다. NIC에서 패킷 수신을 완료하여 프로세서에 인터럽트 신호를 보내면 프로세서는 수신한 데이터를 RAM에 저장한다.

저장된 Ethernet 패킷의 헤더를 체크하여 해당 프로토콜의 루틴으로 가서 작업을 수행한다.

그림 4.4에 ARP의 구성도를 나타내었다. 통신을 개시할 때는 마이크로프로세서 시스템의 MAC 어드레스를 모르므로 호스트는 ARP 요구 패킷을 송신하게 된다. 수신측에서는 ARP 요구 패킷을 체크하여 자신의 IP 어드레스와 같은 경우에는 ARP 응답 패킷을 만들어서 전송하며 같지 않은 경우에는 패킷을 파기하고 다음 패킷 수신을 대기하게 된다. ARP에 의하여 마이크로프로세서 시스템의 MAC 어드레스를 확인한 후에 호스트는 IP 패킷을 송신하게 된다.

그림 4.5에 IP의 구성도를 나타내었다. IP 패킷을 수신한 후 헤더를 체크하여 자신의 IP 패킷이 맞으면 상위 프로토콜 번호를 확인한 후 그 루틴으로 가서 다음 작업을 수행하게 된다. 현재는 UDP를 사용하므로 상위 프로토콜 번호는 17이며 이것과 다른 프로토콜은 수행하지 않으므로 번호가 맞지 않으면 패킷을 파기한다.

그림 4.6에 UDP의 구성도를 나타내었다. UDP 헤더를 체크한 후 데이터를 확인하게 된다. 이때 주의할 점은 앞의 헤더는 모두 이진수이지만 UDP의 데이터는 아스키코드 값이므로 아스키 코드로 변

환하여 확인하여야 한다. 본 연구에서는 LED를 On/Off 하는 명령과 아날로그 데이터를 보내라는 요구의 데이터이므로 먼저 LED를 On/Off하고 아날로그 데이터를 전송하게 된다.

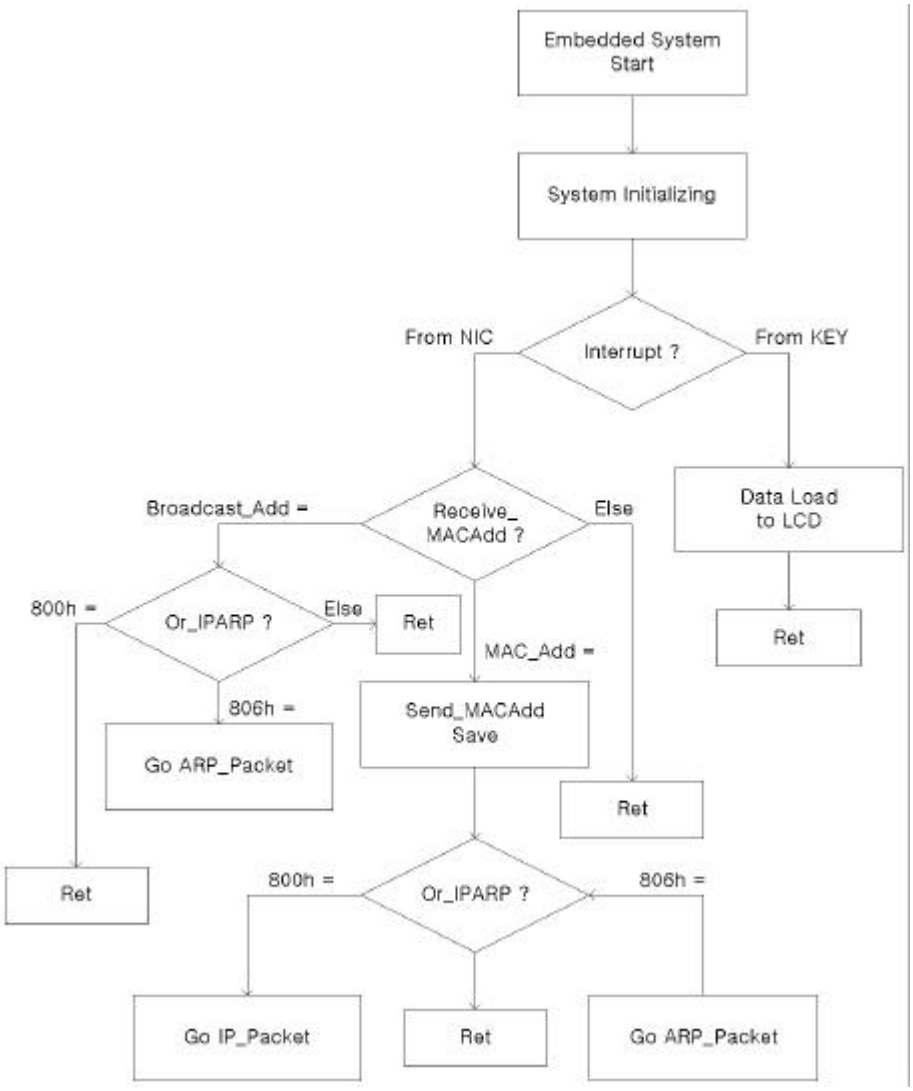

그림 4.3 **소프트웨어 흐름도**

Fig. 4.3 Software flow chart

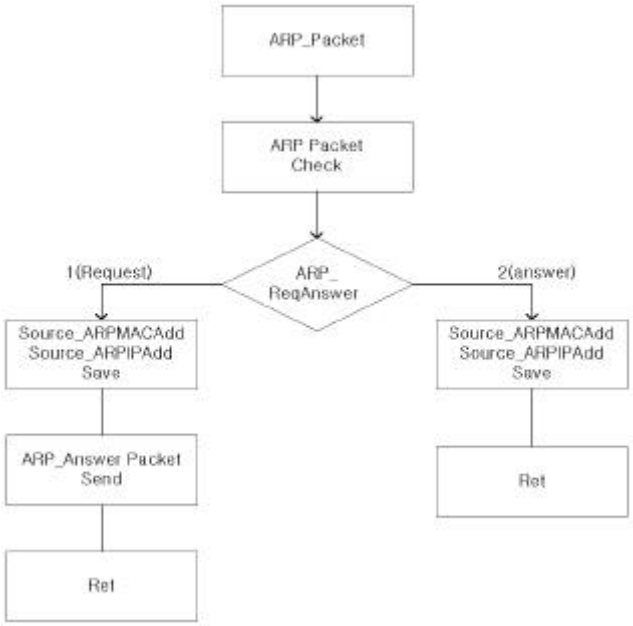

그림 4.4 ARP의 흐름도

Fig. 4.4 ARP flow chart

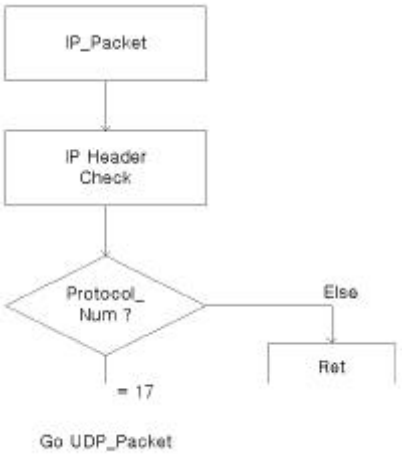

그림 4.5 IP의 흐름도

Fig. 4.5 IP flow chart

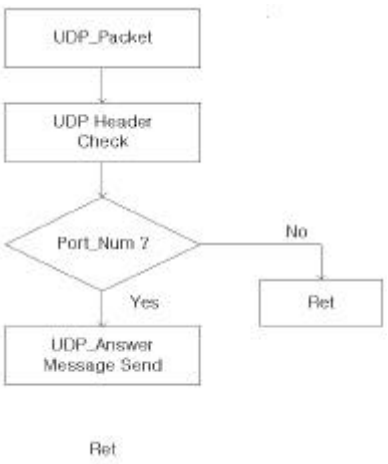

그림 4.6 UDP의 흐름도

Fig. 4.6 UDP flow chart

4.3 모니터링 시스템

개발된 마이크로프로세서 시스템을 원격으로 감시하고 제어하기 위한 모니터링 시스템은 Visual C++ 6.0에서 개발되었으며 데이터 관리를 위하여 MS SQL 데이터베이스를 사용하였다.

그림 4.7과 4.8에 개발된 모니터링 시스템을 나타내었다. 그림 4.7은 마이크로프로세서 시스템을 제어하기 위한 메인 윈도우이다. 발전기 플랜트를 대상으로 하여 2대가 독립적으로 동작하도록 하였으며 각각 회전수, 온도, 압력 3개의 아날로그 데이터와 밸브 On/Off 용으로 1개의 디지털 데이터를 가지고 있다.

모니터링 시스템은 타이머에 의해서 일정시간 간격으로 마이크로프로세서 시스템과 데이터를 주고 받는다. 모니터링 시스템에서 마이크로프로세서 시스템으로 밸브의 On/Off 데이터를 보내면 마이크로프로세서 시스템은 LED를 On/Off 하고 모니터링 시스템으로 3개의 아날로그 데이터와 LED On/Off 데이터를 보낸다. 모니터링 시스템은 이 데이터에서 3개의 아날로그 데이터는 에디터 창에 표시하며 LED On/Off 데이터는 처음에 보낸 데이터와 비교하여 맞으면 밸브를 파란색(On)/빨간색(Off)으로 표시하며 틀리면 노란색으로 표시하게 된다. 그 후에 데이터를 데이터베이스에 저장한다. 또한 아날로그 데이터가 일정 값의 범위 이상이나 이하로 들어오면 알람이 발생하여 이상 유무를 확인 할 수 있도록 하였다.

그림 4.8은 데이터베이스에 저장된 데이터를 그래프로 보여주는 트랜드 윈도우이다. 이 창의 밑에 있는 콤보박스에서 시간과 발전기 종류 그리고 아날로그 데이터의 종류를 선택하면 그래프를 볼 수 있으므로 발전기의 이상 유무를 알 수 있게 된다.

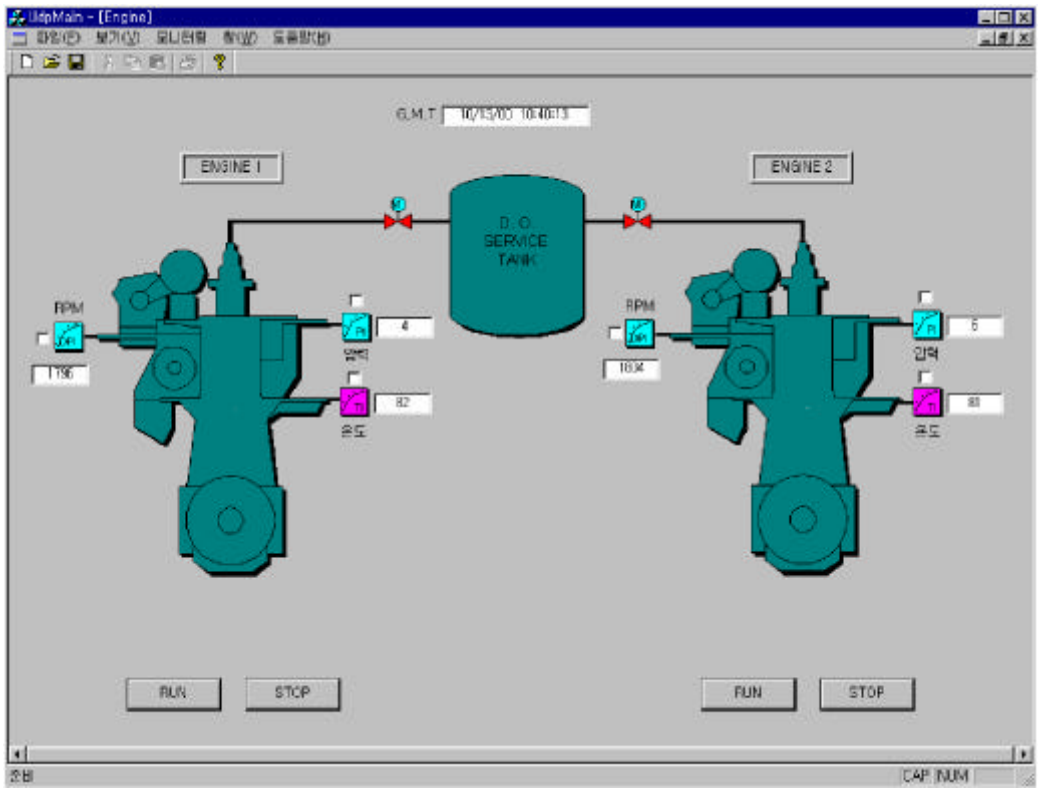

그림 4.7 개발된 모니터링 시스템의 메인 윈도우

Fig. 4.7 Main window of the developed monitoring system

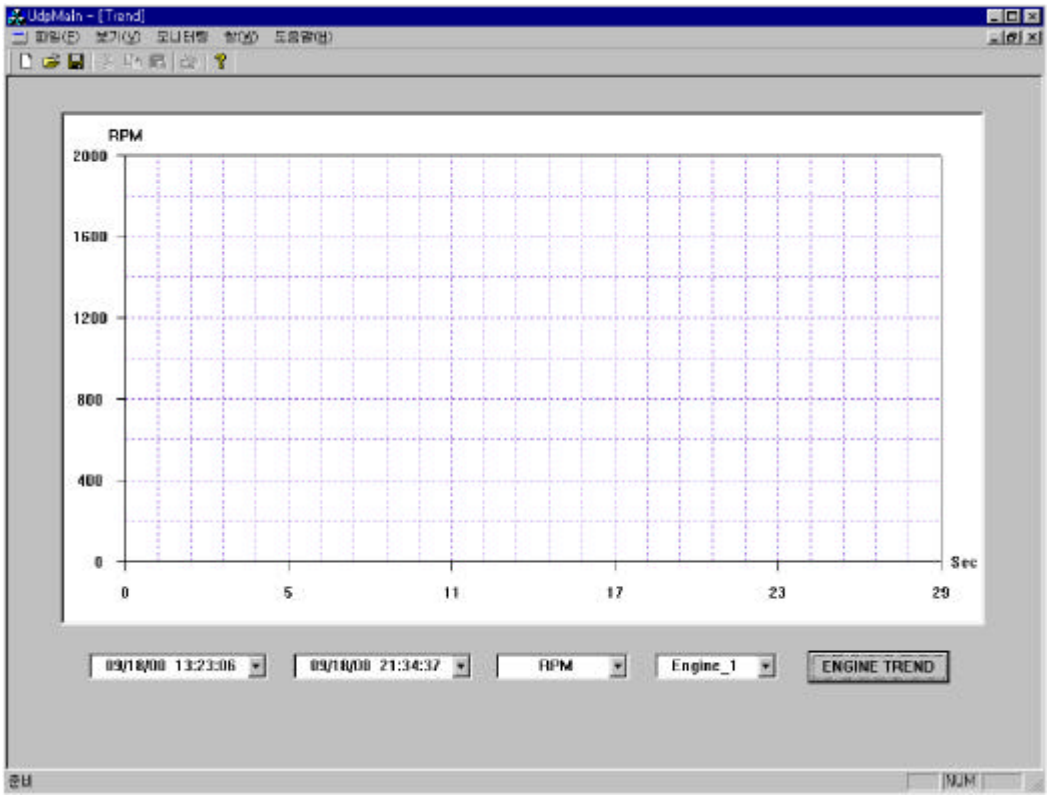

그림 4.8 개발된 모니터링 시스템의 트랜드 윈도우

Fig. 4.8 Trend window of the developed monitoring system

제 5 장 실험 및 고찰

5.1 실험장치의 구성

그림 5.1는 실험장치의 구성도로서 원격제어와 감시가 가능한지를 확인하기 위하여 다음과 같이 구성하였다.

- 원격제어를 확인하기 위하여 서울과 부산에 모니터링 시스템을 설치
- 다수의 플랜트 제어 확인을 위하여 마이크로프로세서 시스템 2대를 구성
- 마이크로프로세서 시스템을 연결하기 위해 Hub 사용
- 회전수, 온도, 압력 3개의 아날로그 데이터와 LED On/Off용의 한 개의 디지털 데이터를 사용

5.2 동작 실험

그림 5.2와 사진 5.1은 밸브 On/Off 실험이고 그림 5.3은 이 때의 회전수 데이터를 그래프로 그린 것이다.

그림 5.2에서 Engine 1에는 밸브가 파란색으로 되어 있으므로 On 명령을 나타내며 Engine 2는 빨간색으로 되어 있으므로 Off 명령을 나타낸다. 사진 2의 마이크로프로세서 시스템의 첫 번째 것은 LED가 On 되어 있고 두 번째 것은 LED가 Off 되어 있으므로 원격제어가 되고 있음을 알 수 있다.

그림 5.3은 데이터베이스에 저장된 기관 회전수 데이터를 그래프로

나타낸 것으로써 1800[RPM]을 계속 유지하고 있음을 알 수 있으며 콤보박스를 선택하면 다른 아날로그 데이터도 그래프로 볼 수 있다.

그림 5.4와 5.5는 경보시험을 행한 것이다. 마이크로프로세서 시스템에서 일정 범위 이상이나 이하의 데이터가 들어오면 알람이 울리면서 체크박스에 표시가 되므로 어느 것이 잘못 되었는지 금방 알 수 있게 된다. 그림 5.4에서 회전수에 체크 표시가 되어 있으므로 회전수에 이상이 있다는 것을 알 수 있으며 그림 5.5를 보면 회전수가 일정 범위 이상으로 올라간 것을 볼 수 있다.

그림 5.6은 패킷 손실률을 확인하기 위하여 마이크로프로세서 시스템에서 시뮬레이션 데이터로 0과 2를 주기적으로 전송하게 하여 실험을 행한 그래프이다. 그림을 보면 패킷이 손실됨 없이 통신이 되고 있음을 확인할 수 있다.

부산에서의 실험에서는 지연시간이 없었으며 서울과 부산간 실험에서는 1초 이내의 지연시간이 있었으므로 리얼타임으로 통신이 됨을 확인할 수 있었다.

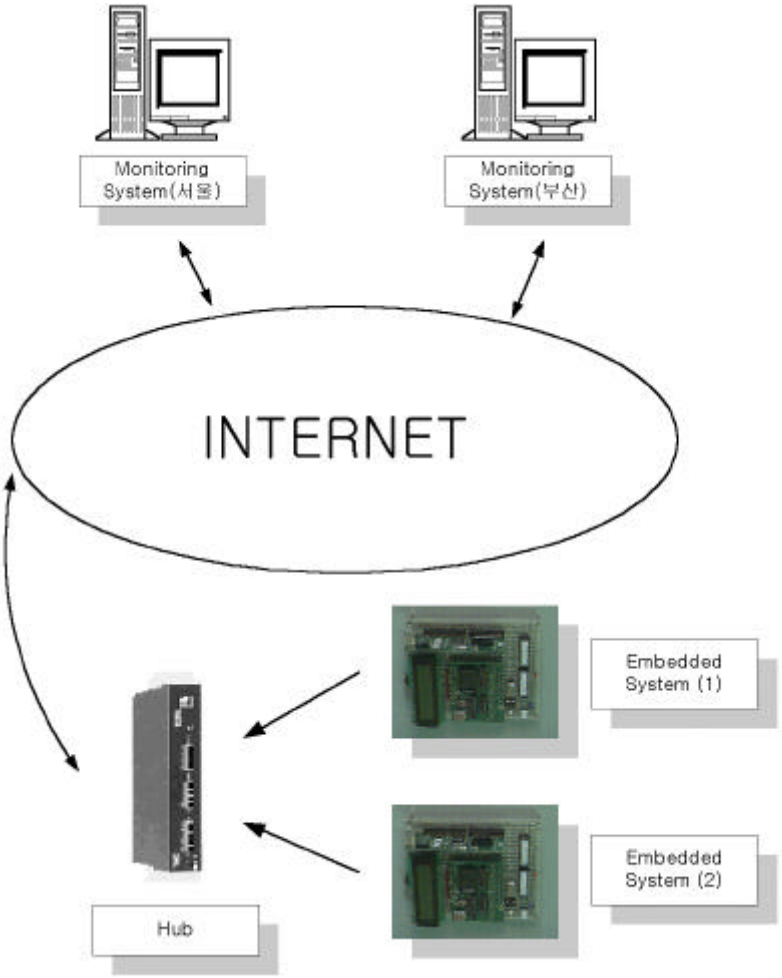

그림 5.1 실험장치의 구성도

Fig. 5.1 Experiment equipments diagram

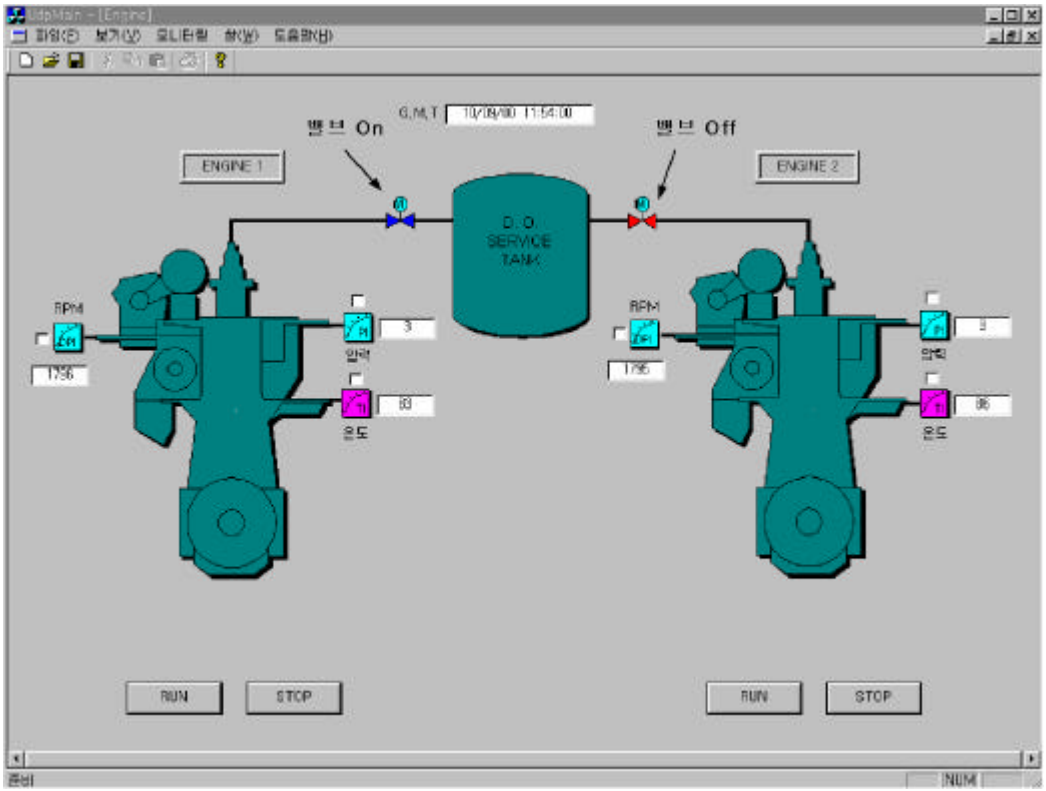

그림 5.2 **밸브** On/Off **실험**

Fig. 5.2 Valve On/Off experiment

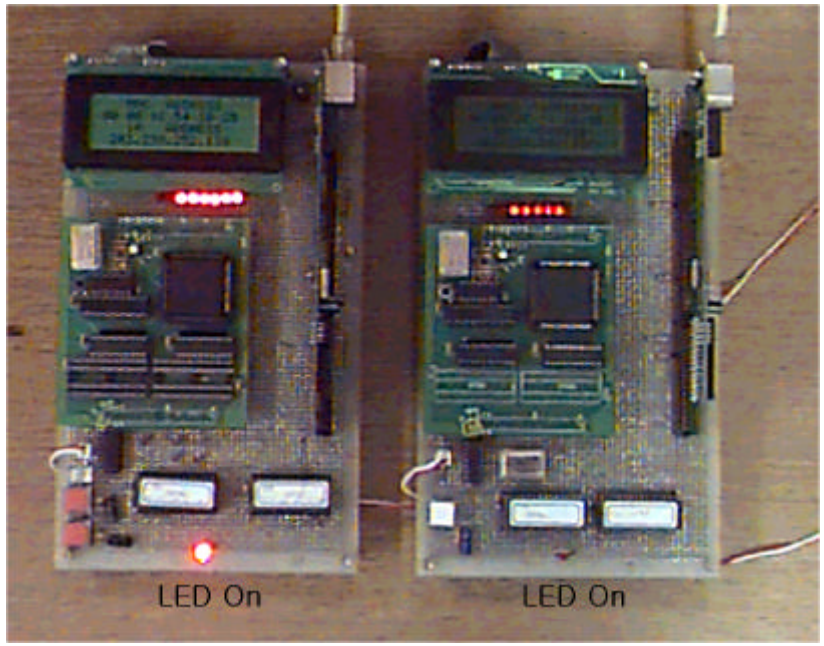

사진 5.1 LED On/Off **실험**

Photo. 5.1 LED On/Off experiment

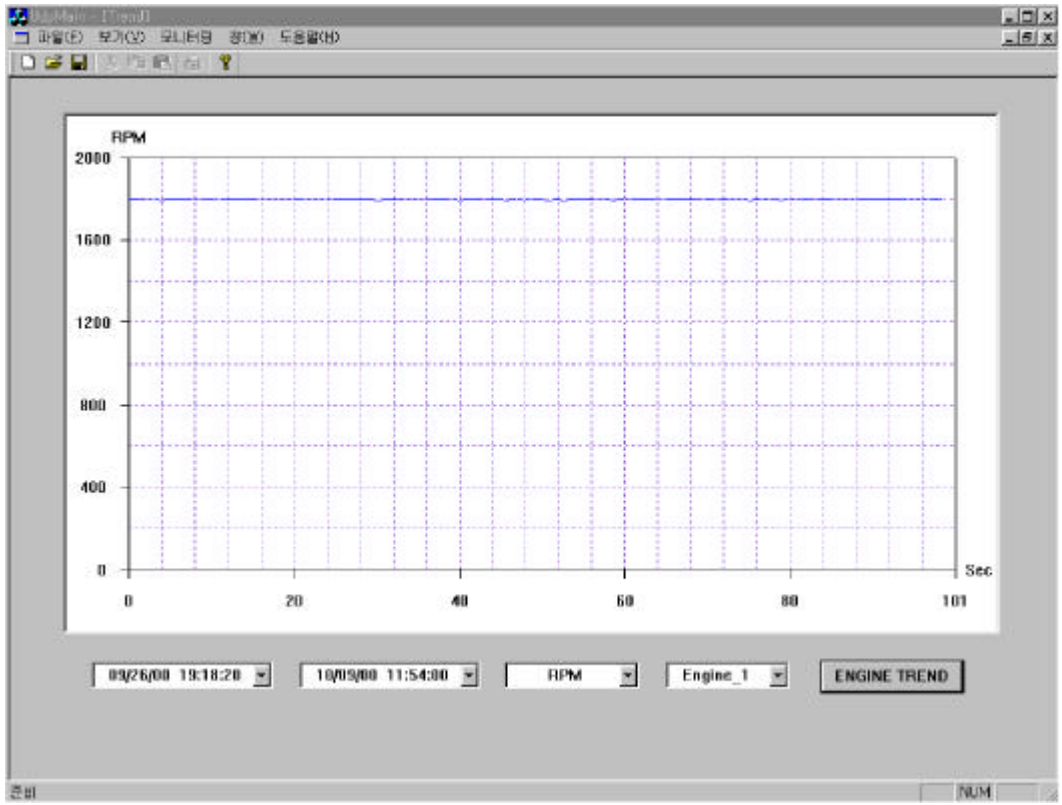

그림 5.3 기관 회전수 데이터 그래프

Fig. 5.3 Engine RPM graph

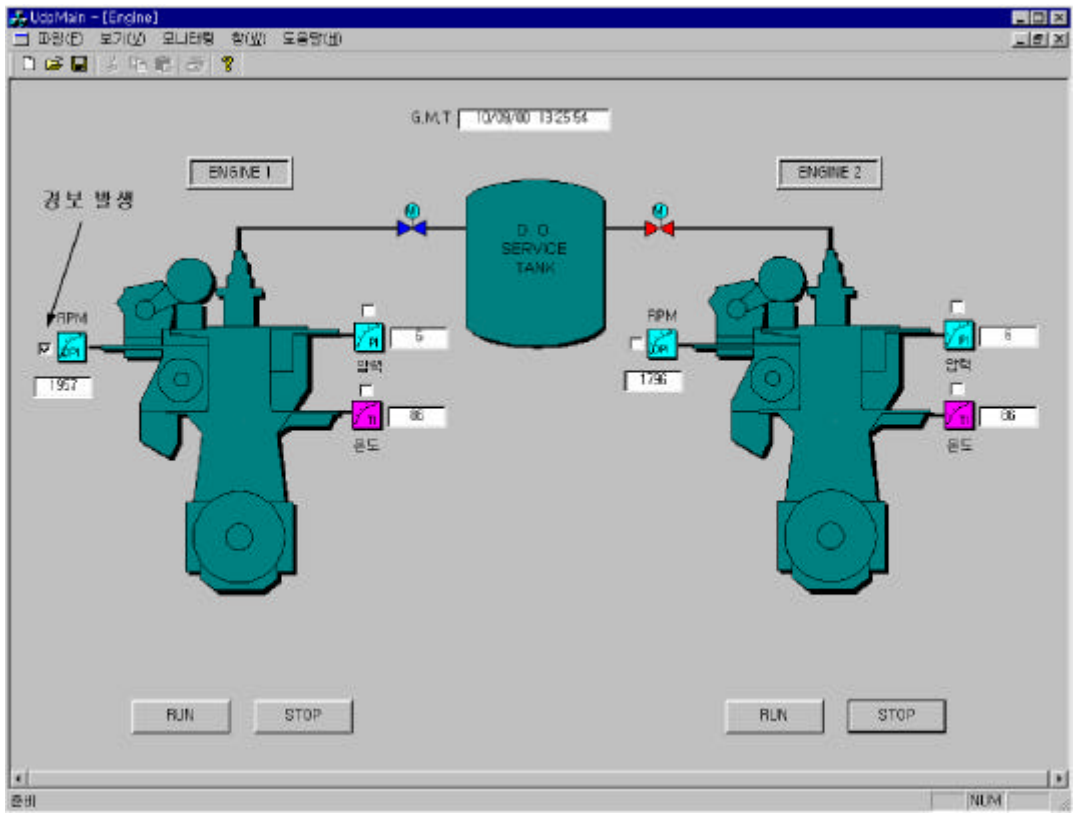

그림 5.4 경보 실험

Fig. 5.4 Alarm experiment

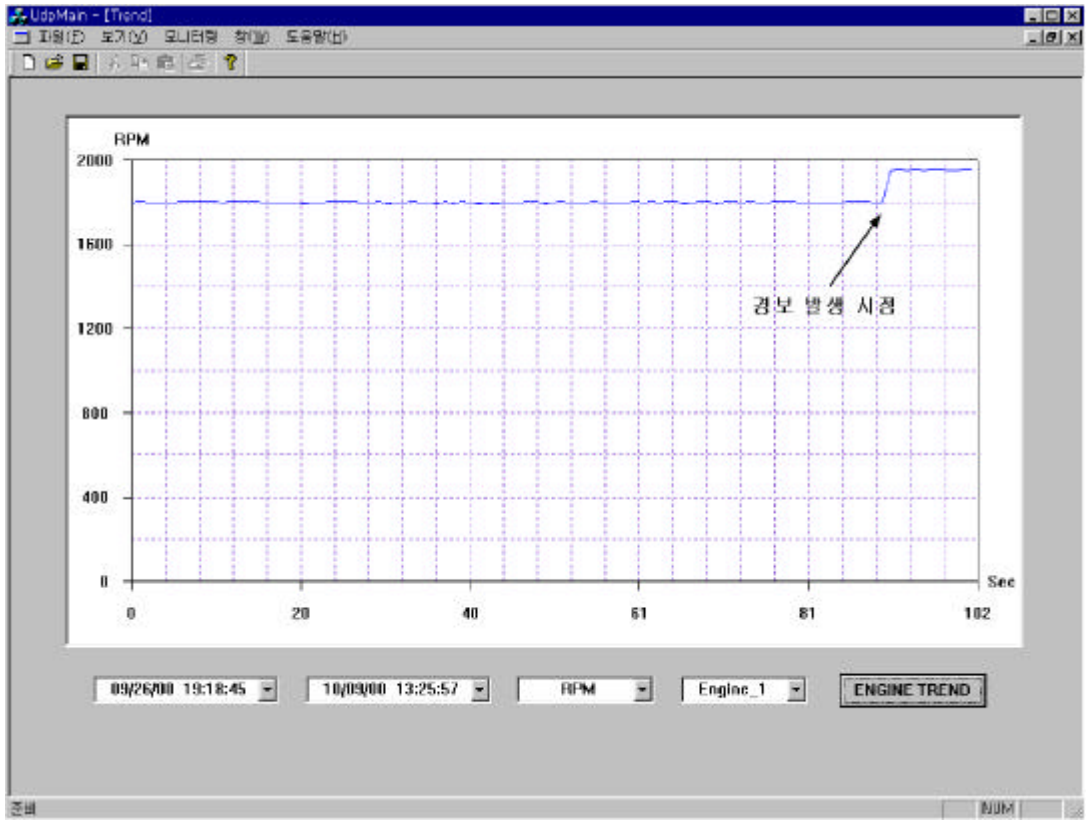

그림 5.5 경보 실험 그래프

Fig. 5.5 Alarm experiment graph

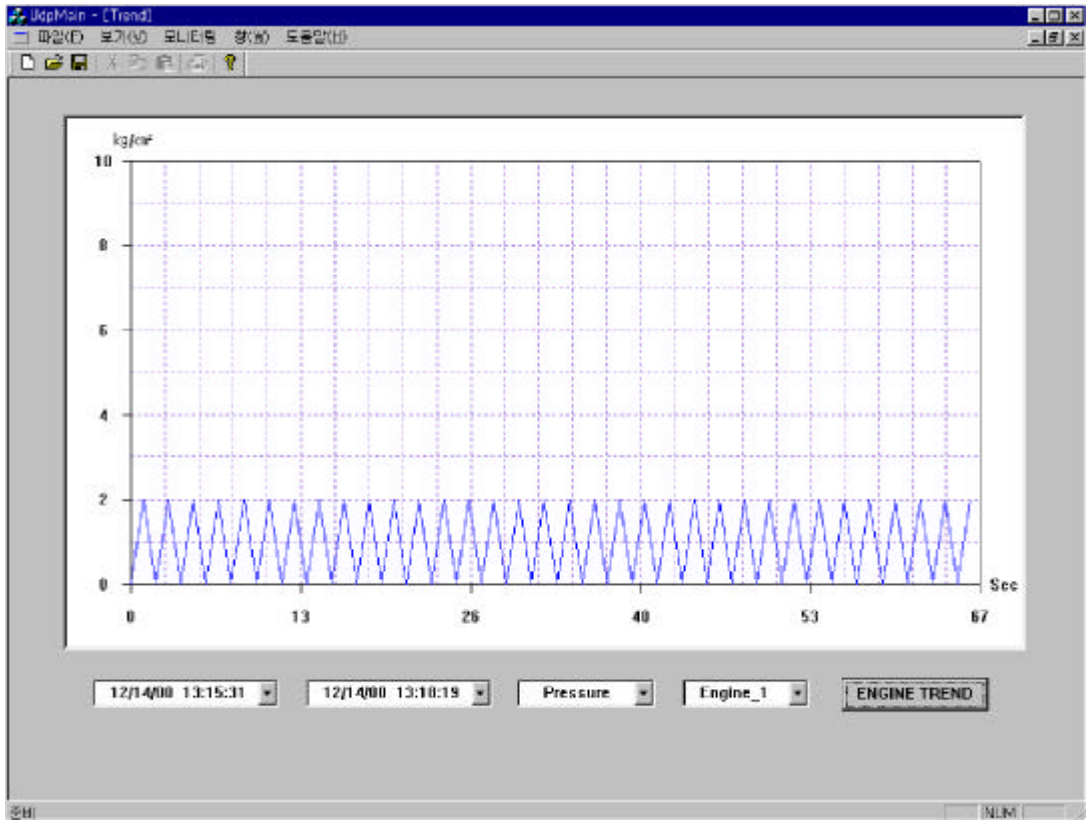

그림 5.6 통신 실험 그래프

Fig. 5.6 Communication experiment graph

제 6 장 결 론

본 연구에서는 인터넷 상에서 마이크로프로세서 시스템으로 제어 대상을 감시제어하기 위하여 마이크로프로세서를 이용한 인터넷 원격감시제어 시스템을 개발하였다. 이를 위하여 IEEE 802.3 표준 통신 프로토콜 기능을 수행하는 ISA 버스용 NE2000 호환의 Ethernet NIC를 사용하였다. NIC를 제어하고 ARP, UDP/IP 프로토콜 기능을 수행하기 위하여 인텔 80C196KC 프로세서를 사용하였으며 프로토콜의 순차적 수행상태를 알아보기 위하여 LCD 표시장치를 사용하였다.

PC상에서 윈도우즈 운영체제로 동작하는 감시제어시스템 응용 프로그램을 제작하고 IP인식형 마이크로프로세서 시스템과 UDP/IP 프로토콜을 이용하여 제어신호를 송신하고 그 동작 결과 감시신호를 수신하여 데이터베이스에 저장하였다. 응용 프로그램에서는 저장된 데이터를 Active-X를 이용하여 화면에 표시하였다.
원격지(서울)와 근거리에서의 실험을 통하여 개발된 시스템이 정상적으로 동작하는 것을 확인할 수 있었다.

본 실험에서는 2개의 IP 인식형 마이크로프로세서 시스템을 사용하였으나 추후 많은 IP인식형 마이크로프로세서 시스템과 연결된 경우에도 원활하게 동작하는 지를 확인하기 위하여 실험을 행해야 할 것이다.

참고문헌

[1] 이석원,이영진,강성귀,남부희, "TCP/IP를 이용한 필드버스 monitoring 시스템 구현", KACC, pp.684-687, 1998

[2] 차석근 "TCP/IP를 이용한 제어기기 인터페이스", KACC, pp.72-76, 1996

[3] 윤영찬,박재현, "공장 자동화를 위한 TCP/IP 네트웍 에서의 실시간 통신에 관한 연구", KACC, pp.1032-1035, 1996

[4] 박홍성, 진창기, 강영진 "Implementation of Easy API for MMS(Manufacturing Message Specification) over TCP/IP", KACC, pp.A41-A44, 1996

[5] 윤재식, 김재양, 정선태, "소형 실시간 커널을 위한 TCP/IP 프로토콜 설계 및 구현", KACC, pp.414-417, 1997

[6] 강성귀, 김동성, "MMS on TCP/IP를 이용한 시범 플랜트 구축", KACC, pp.688-691, 1998

[7] 홍승호, 김기암, 김지용, 고성준, "분산제어 및 자동화시스템과 필드버스", KACC, pp.19-29, 1996

[8] 홍승호, "필드버스 기술 동향", KACC, pp.13-18, 1997

[9] 하종성, "User Datagram Protocol의 개발과 응용", 한국과학기술원, 1985

[10] 진교홍, "고속실시간 통신을 위한 TCP/IP 프로토콜의 하드웨어 설계 및 구현", 부산대학교 대학원, 1997

[11] 손정호, "반도체 생산기기의 원격제어를 위한 인터넷 어댑터의 설계 및 구현", 부산대학교 대학원, 1999

[12] 최창식, "TCP/IP - Fast Ethernet 간 접속을 위한 하드웨어 인

터페이스 모듈의 설계 및 구현", 부산대학교 대학원, 1998

[13] 장용기, "TCP/IP를 이용한 無停電 電源供給裝置(U.P.S.)의 遠隔監視및 制御에 관한 硏究", 연세대학교 산업대학원, 1997

[14] User Datagram Protocol, RFC 768, 1980

[15] An Ethernet Address Resolution Protocol, RFC 826, 1982

[16] J.R Leigh, Applied Digital Control, Prentice-Hall, 1992

[17] Martin, James, TCP/IP Networking : Architecture, Administration and Programming, Prentice-Hall, 1994

[18] 유상렬, TCP/IP 인터네트, 성안당, 1995

[19] Michishita, Nobuhiro, 이기종접속과 TCP/IP, 東西交易出版部, 1994

[20] D.E. Comer, TCP/IP 인터네트워킹, 그린, 1998

[21] W.R. Stevens, TCP/IP 네트워크, 진영사, 1998

[22] 요시무라, 노비루, TCP/IP 동작원리와 트러블 대책, 동서네트연구회, 동서, 1999

[23] 차영배, Micro Controller 80196, 다다미디어, 1997

[24] 김용성, VISUAL C++ 6.0 완벽 가이드, (주)영진출판사, 1998

[25] 이상엽, VISUAL C++ Programming Bible, (주)영진출판사, 1998

[26] 월간 전자기술, (주)첨단, 2000. 2

감사의 글

본 논문이 완성되기까지 여러모로 부족한 제자를 헌신적인 지도와 격려로 이끌어 주신 유영호 지도 교수님께 진심으로 감사드립니다. 바쁘신 와중에도 자상한 관심과 열의로 세밀한 검토를 해주시고, 용기와 격려를 아끼지 않으신 진강규 교수님과 손주영 교수님께 감사드리며 항상 관심있게 지켜봐 주신 조석제 교수님, 하윤수 교수님, 황승욱 교수님, 류길수 교수님 학부의 모든 교수님께 감사 드립니다.

실험실에 있으면서 많은 도움을 주신 천행춘 이사님, 김영일 선배님과 같이 일했던 동료 김 경엽, 문 복산, 임 정훈, 오 현경, 안 순영에게 감사의 마음을 전합니다. 저의 논문을 도와준 위용욱과 각 연구실의 선후배 그리고 제어계측공학과 강군호, 정인 선배님께 감사의 마음을 전합니다.

옆에서 물심양면으로 도와주고 격려와 사랑을 아끼지 않은 형님 누나들과 친척 그리고 장인어른과 장모님, 처갓집 식구들에게 감사의 마음을 전합니다.

대학원 진학과 함께 저의 반려자가 되어 옆에서 도와준 된 소중한 아내 연화 그리고 얼마전 태어난 아들 민규와 이 기쁨을 나누고 싶습니다.

마지막으로 오늘의 제가 있기까지 언제나 변하지 않는 사랑과 인내하심으로 많은 어려움 속에서도 저의 모든 것을 뒷바라지 해주신 아버님, 어머님께 이 논문을 바칩니다.