

TCP/IP와 RS-232의 게이트웨이를 이용한 원격 모니터링 시스템 구현

오태근 김형배 남부희
강원대학교 BK21 전기전자정보통신공학부

Implementation of Remote Monitoring System Using Gateway for TCP/IP and RS-232

Tae Geun Oh, Hyung Bae Kim and Boo Hee Nam
BK21 Dept. of Electrical and Computer Engineering, KANGWON NATIONAL UNIVERSITY

Abstract - 현대사회에서 네트워크가 차지하는 비중은 점점 커져가고 있다. 따라서 네트워크를 활용할 수 있는 많은 노력들이 이루어지고 있다. 네트워크 프로토콜의 하나인 TCP/IP 프로토콜을 이용한 LAN은 다른 정보를 처리하는 장치들을 연결하고, 자원을 공유하며, 고속의 통신을 가능하게 하는 시스템이다.

본 논문은 TCP/IP 프로토콜과 장비와 장비 사이의 직렬 통신 방법인 RS232를 연계하는 게이트웨이를 통한 시스템 제어와 모니터링을 목적으로 하고자 한다. 제어 대상 시스템은 여러 가지의 적용이 가능하지만 본 논문을 위해 80C196KC CPU로 제어되는 스텝 모터를 사용하였다.

1. 서 론

최근, 폭발적인 인터넷의 발전은 메시지 전달이나, 간단한 정보의 이용을 넘어서 사용자가 네트워크상의 어떠한 시스템들에 쉽게 접근할 수 있는 인터넷의 장점을 이용한 원격 시스템 제어라는 분야로 그 범위가 확장되고 있다. 웹 브라우저를 이용한 로봇팔 제어, 원격 정원관리 시스템 그리고, 센서에 기초한 이동 로봇등이 그 예를 대표하고 있다.

TCP/IP 프로토콜은 완전히 다른 오퍼레이팅 시스템에서 운영되는 많은 서로 다른 컴퓨터 회사들의 모든 종류의 컴퓨터들이 서로 통신하는 것을 가능하게 해준다. 그것은, 대중적으로 별다른 문제 없이 실행되어지는 이유에서 open system이라고 할 수 있는 것이다. 결국, 백만 컴퓨터 이상의 wide area network(WAN) 인터넷의 기초를 형성하는 것이다.

서로 거리를 두고 있는 장비와 장비 사이에 데이터 통신이 필요한 경우 직렬 통신 또는 병렬 통신 방법을 생각할 수 있다. 병렬 통신은 다수의 신호선이 필요하므로 비용이나 설치 등이 불리하고 잡음에도 약하다. 이에 비해 직렬 통신은 신호선 수가 적고 비교적 먼 거리의 통신이 가능하므로 많이 사용한다. 비교적 가까운 거리(15M 이내)의 통신을 할 때에는 RS232C 방법을 사용하고, 좀 더 먼 거리의 통신을 원하는 경우는 RS485 방식(최대 1.2km)을 사용한다.

TCP/IP와 RS232C방식의 통신은 서로 다른 형태를 취하고 있다. 이러한 서로의 특징을 가지는 통신 프로토콜을 연계하기 위해서는 게이트웨이의 개념을 필요로 한다. 게이트웨이는 서로 다른 네트워크의 시스템들과 통신을 필요로 한다. 만약, 네트워크 위의 호스트가 다른 네트워크와의 통신을 필요로 한다면, 게이트웨이를 통한 라우터가 정의되어야 한다.

본 논문은 위의 각각의 특징을 가지는 프로토콜과 장치를 사용하여 네트워크에 연결된 PC에서 TCP/IP 프로토콜을 통해 서로 다른 프로토콜을 연계하는 게이트웨이에 접속하게 되고, PC의 제어 정보가 다시 게이트웨이에서 RS-232C통신을 통해 호스트가 제어하고자 하는 시스템을 제어하고, 모니터링하고자 하는 것이다. 이러한 원격 모니터링 시스템의 설계는 인터넷으로 전세계를

쉽게 연결 할 수 있고, 우리 생활의 대부분이 전자 제품의 사용을 일반화한다는 점을 감안한다면 원격지에서 전자 시스템을 제어하고 모니터링하는 것은 상당한 의미가 있는 것이다.

2. 본 론

2.1 TCP/IP

네트워크 프로토콜은 통신의 서로 다른 역할을 위해 각 층의 책임을 가지는 Layer들로 발전된다. TCP/IP와 같은 프로토콜 suite는 다양한 층들의 프로토콜의 조합으로 이루어진다.

2.1.1 Datalink & Physical layer

OSI의 하위 두 계층을 담당하는 Ethernet은 CSMA/CD 프로토콜을 이용하여 물리 매체를 통해 연결되어 있는 호스트와 호스트, 호스트와 노드, 노드와 노드 사이에 오류없는 데이터 전송으로 상위 계층에서

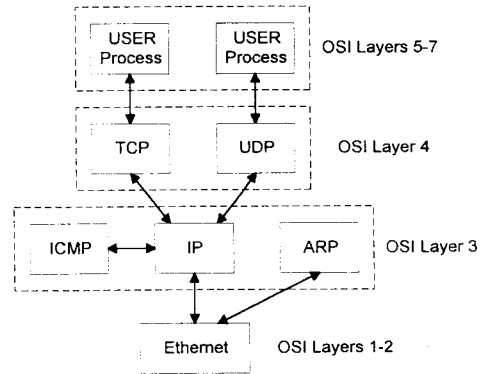


Fig.1. TCP/IP의 계층적 구조

비스를 제공해 주는 역할을 한다.

2.1.2 Network Layer

제 3계층인 네트워크 계층은 패킷에 주소를 지정하고 논리적인 주소와 이름을 물리적인 주소로 변환하는 일을 담당한다. 또한 이 계층은 송신측 컴퓨터에서 수신측 컴퓨터까지의 최적인 전송 경로를 결정한다. 전송 경로는 네트워크의 각종 조건(혼잡 상태 등)과 서비스의 우선 순위 및 다른 요소(비용 등)들에 의해 결정된다. 또한 패킷 스위칭, 라우팅, 혼잡 제어(Congestion Control)와 같은 네트워크상의 트래픽 문제를 관리한다.

2.1.3 Transport Layer

Internetwork를 통해 신뢰성 있는 전송을 어떻게 할 것인가 하는 문제는 Transport layer에서 고려된다. Service의 신뢰성을 보장하는 것과 더불어 Transport layer는 virtual circuit에 대한 생성, 유지 및 순서적

인 종료에 대한 방법을 제공하고, 이 밖에 전송 에러 검출 및 복구, Information Flow Control(한 System이 다른 System의 Data로 인해 Overrun되는 것을 방지해주는 기능)을 제공한다. Transport layer는 virtual circuit을 생성하고 안하고에 따라 크게 UDP와 TCP 두 개의 프로토콜이 있다.

2.1.3.1 UDP

UDP는 비연결형(connectionless) 서비스를 제공하는 프로토콜로 헤더 처리에 필요한 오버헤드가 적어 간단한 패킷을 주고받는 경우에 적합하지만 데이터그램의 손실 확인이나 순서유지를 보장해주지 않기 때문에 상위 계층에서 필요하면 이를 처리해야 한다.

2.1.3.2 TCP

연결형(Connection oriented) 서비스를 제공하는 프로토콜로 종단점간(end-to-end)의 연결 개설, 오류 발생 시 데이터 재전송, 패킷 전달순서 확인, 중복 패킷 제거, 데이터 흐름제어 등을 제공한다. 신뢰성 있는 통신을 제공하지만 전송할 데이터의 양이 적은 경우 데이터 전송에 관한 오버헤드가 상당히 크다.

2.1.4 Application Layer

어플리케이션 계층은 특별한 어플리케이션의 상세함을 취급한다.

2.2 소켓

Network 프로그램 작성에는 여러 가지의 레벨이 있을 수 있는데, 통신 장비를 직접 구동하는 디바이스 드라이버형 프로그램, TCP와 같은 트랜스포트 계층의 인터페이스를 이용하는 Socket 프로그램, 그리고 응용계층을 이용하는 응용 계층 프로그램등 크게 세가지로 구분할 수 있다.

또한 소켓은 TCP/IP의 소프트웨어적 기능을 담당한다. 즉 소프트웨어로 작성된 통신 접속점이라고 할 수 있으며, 응용 프로그램에서 통신망으로 데이터를 송수신해야 할 때의 통로가 되는 것이다. 소켓에 있어서 중요한 5가지의 요소를 생각할 수가 있는데, 프로토콜, 자신의 IP주소, 상대방의 IP주소, 자신의 포트번호, 상대방의 포트번호가 그것이다.

본 논문을 위해 게이트웨이인 HellowDevice 1300(IP address : 210.115.43.43)에 연결하기 위한 클라이언트 프로그램은 Visual C++을 사용하여 Fig.2와 같다. 이러한 클라이언트 프로그램은 게이트웨

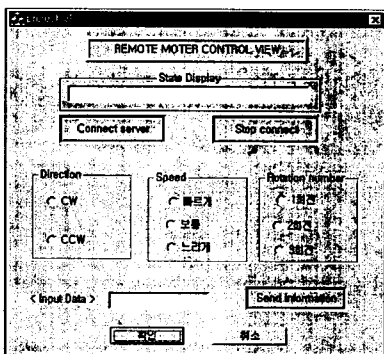


Fig.2. 클라이언트 moto control 이인 HellowDevice 1300에 접속을 하고, 스템 모터를 제어하기 위한 여러 가지 정보들을 80C196KC에 전달한다. 또한 클라이언트 프로그램은 소켓의 상태를 표시함으로써 네트워크의 상태를 확인할 수 있다. Fig.2의 클라이언트가 제어하는 스템 모터의 동작은 정확하게 모

를 회전하는 것이다. 종단의 스템 모터에 전달되는 데이터 패킷은 Fig.3과 같다. HellowDevice 1300은 보드내부에 512KB 플래쉬 메모리를 내장하고 있기 때문에 한번에 보내는 데이터의 양을 512KB로 제한하지만, 본 논문에서의 데이터는 방향, 스피드, 회전수의 데이터만 쓰였다.

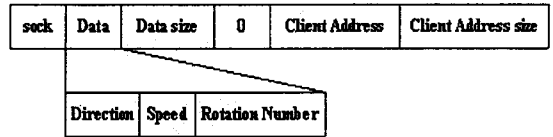


Fig.3. 데이터 패킷

게이트웨이 프로그램에 사용된 RS-232통신의 초기화는 80C196-KC의 RS-232통신과 호환을 위해 9600bps, 8비트 데이터를 사용하였다.

Fig.4는 시스템 알고리즘으로서 Fig.3의 데이터가 80C196KC보드에 전달되고, 각 단계에 알맞은 데이터들이 처리되어 스템 모터를 회전시키게 된다.

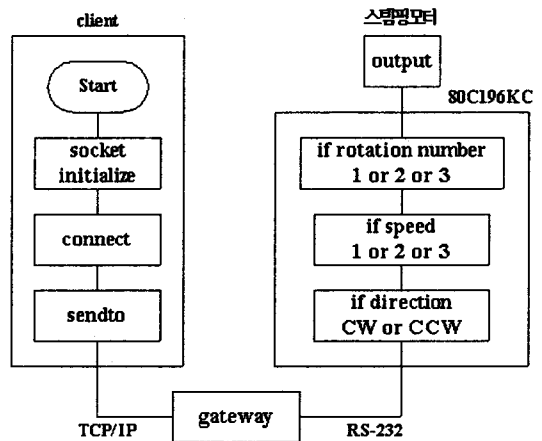


Fig.4. System Algorithm

2.3 RS-232 and 80C916KC

RS-232는 모델과 같이, 터미널과 현재의 데스크탑 컴퓨터 같은 종말 사용자의 데이터인 "data terminal equipment"(DTE)와 데이터의 송신과 같은 "data communication equipment"(DCE)사이의 통신을 위해 설계되었다. 결국 하나의 머신에서 전송된 데이터는 다른쪽 머신의 받는 데이터로 흐르게 되며, 반대도 성립한다.

장비와 장비간에 데이터 통신을 하는 경우 3가지(단방향, 반 이중, 전 이중)방식으로 분류해 볼 수 있는데 본 논문을 위해 전 이중 방식을 채택하였다. 전 이중 통신은 보내는 신호와 받는 신호가 별도로 분리되어 있어 동시에 데이터를 주고 받을 수 있다. 비동기식 통신의 데이터 포맷은 START 비트, DATA 비트, Parity 비트, STOP 비트로 나누어진다.

80C196KC의 고속 출력장치는 타이머1, 타이머2를 이용하여 정해진 시점에서 HSO.0 - HSO.5핀을 이용하여 외부로 펄스를 출력하거나, A/D컨버전 시작등의 내부 장치에 명령을 내리는 장치이다. 또, 고속 출력 장치는 대기중인 이벤트를 저장할 수도 있고, 지정된 시간을 CAM(Content Address Memory)화일에 저장할 수 있다. 본 논문에서는 이러한 80C196KC의 고속 출력장치를 이용하여 RS-232통신으로 인터넷을 통해 들

어오는 명령으로 80C196KC보드에 연결된 스템 모터를 돌린다.

2.4 게이트웨이 및 시스템 구성

본 논문을 위한 시스템의 구성은 TCP/IP통신을 사용하는 클라이언트 소켓 인터페이스 장치의 정보를 네트워크를 이용하여 게이트웨이의 역할을 하는 HellowDevice 1300에 전달한다. 이러한 정보는 다시 RS232통신을 통해 80C196KC보드에 전달되고, 정보에 맞는 스템핑 모터를 제어하게 된다. 본 논문을 위한 시스템 구성도는 Fig.3과 같으며, 실제적인 하드웨어는 Fig.4에 나타나 있다.

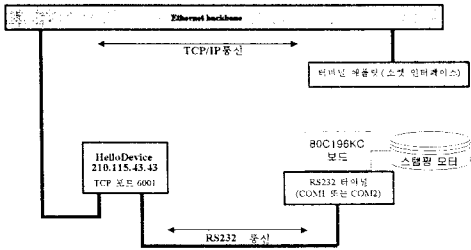


Fig.5. 시스템 구성도

본 논문을 위한 gateway system으로 상용된 하드웨어는 세나 테크놀로지의 HellowDevice 1300카드이다.

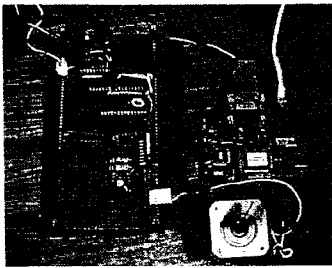


Fig.6. 시스템 구성

HellowDevice 1300카드는 Senix SX52BD 8비트 마이크로프로세서와 최적화된 웹 서버기능, 10Base-T 이더넷 접속장치, Rs-232/485 접속장치를 장착하고 있고, TCP/IP stack이 구성되어있어 게이트웨이 시스템에 적합하다. 다음 그림은 1300 board의 간단한 블록 다이어그램을 나타낸 것이다.

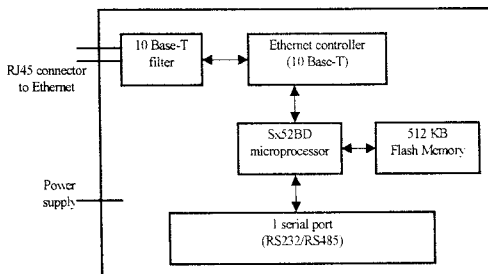


Fig.7. Block diagram of the 1300board

3. 결 론

본 논문은 게이트웨이인 HellowDevice 1300보드를 이용하여, 네트워크에 연결된 클라이언트의 정보를 이용하

여, 80C196KC보드의 스템 모터를 제어하고자하는 시스템의 이행을 보인 것이다. 이러한 시스템을 구현함으로써 네트워크에 연결된 PC에서 쉽게 원격 시스템을 제어 및 모니터링 할 수 있었다. 서로 다른 네트워크에서 각 네트워크의 특성을 침해하지 않으면서 데이터를 연계할 수 있는 방법을 제안하였다. 현재 게이트웨이로 사용되는 HellowDevice 1300은 자바 프로그램을 이용한 웹서버로의 역할도 할 수 있기 때문에, 추후에 이러한 기능을 사용하여 더욱 편리하게 장치를 제어하고 모니터링을 할 수도 있을 것이다. 또한 중단의 장치는 본 논문에서는 스템 모터를 사용하였지만, 다른 Device를 사용할 수 있을 것이다.

(참 고 문 헌)

- [1] 이영진, "Gateway for TCP/IP and Fieldbus", 석사논문
- [2] W.Richard Stevens, "TCP/IP Illustrated, Volume1", 1994
- [3] 차영배, "MICRO CONTROLLER 80196", 1999
- [4] 이상엽, "VisualC++ Bible ver.6.x", 1998
- [5] 이도희, "마스터링 TCP/IP", 1997
- [6] Rick Harrison, "MCSE Traing Kit Networking Essentials Plus Thurd Edition", 2000
- [7] 이봉주, 이형구, 이주호, 박기태, "New architecture for mobile robots in home network environment using Jini", IEEE ICRA, 471, 2001
- [8] Oliver C. Ibe, "Introduction to Local Area Networks for manufacturing and Office Systems," IEEE Contr. Syst. Mag., vol. 7, no.3, pp. 36-42.
- [9] W. Richard Stevens, TCP/IP Illustrated, volume 1, Massachusetts, Addison Wesley Publishing Company, 1994.
- [10] Gary R. Wright and W. Richard Stevens, TCP/IP Illustrated, volume 2, Massachusetts, Addison Wesley Publishing Company, 1994.[8] An Ethernet Address Resolution Protocol, RFC 826, 1982.
- [11] Computing the Internet Checksum, RFC 1071, 1988.
- [12] Requirements for Internet Host-Communication Layers, RFC 1122, 1989.
- [13] A TCP/IP Tutorial, RFC 1180, 1991.
- [14] P. Chun, Programming Winsock, Seoul, Inforbook, 1996.
- [15] 차영배, "MICRO CONTROLLER 80196", 1999
- [16] 이상엽, "VisualC++ Bible ver.6.x", 1998
- [17] 이도희, "마스터링 TCP/IP", 1997