인터넷 웹 기반 환경에서의 인버터 원격제어시스템 개발

최 주 엽*, 최 익**, 유 권 종***
광운대학교 전기공학과*, KIST 지능제어연구센터**,
KIER 태양광발전연구팀***

Development of Internet Web-Based Inverter Remote Control System

J. Y. Choi, I. Choy,
G. J. Yu
Dept. of Electrical Engineering, Kwangwoon Univ*,
Intelligent Control System Research Center, Korea Institute of Science and Technology**
Photovoltaic Generation Team, Korea Institute of Energy Research***

Abstract - This paper aims at developing remote control system to control and monitor distributed various devices such as UPS or photovoltaic inverter system through internet. TCP/IP(Transmission Control Protocol/Internet Protocol) and photovoltaic inverter system operated in a row are adopted for network management protocol and applied device, respectively. For controlling and monitoring distributed devices in real-time, java-environment software is constructed. Also, HelloDevice, general-use interface controller between network device and applied device is proposed. Finally, serial communication such as RS-232C is used between controller and applied device.

1. 서 론

요즘에는 여러 기기들이 대부분 지역적으로 멀리 떨어져 있는 경우가 많으나, 현재 산업 및 가정용기기를 관리하는 방식은 초기에 지나지 않고 있으며, 기존의 방식으로는, 전화나 모뎀을 통하여 접속하여 제어신호를 보내는 방식과 직렬통신을 이용하여 서로 데이터를 주고받는 방식이 있지만, 그러나 기존의 방식은, 직렬통신(RS-485, 422, 232)을 이용한 양방향 제어방식의 경우 거리상의 제한이 있고, 전화를 이용한 단방향 제어방식은 제어대상이 정확하게 제어되었는 지에 대한 확인이 불가능하다는 문제점을 지니고 있다. 또한 이러한 방식을 이용하면 하나의 모니터링 시스템이 여러 기기를 제어할 수 없으며, 마찬가지로 여러 모니터링 시스템이 하나의 기기를 제어할 수 없게 된다.

이 같은 문제점들을 해결하기 위해서는 전세계적으로 널리 사용되고 있는 인터넷을 사용해야 한다. 인터넷을 사용하게 되면 사용자는 거리와 시간의 제약을 받지 않고 인터넷을 통하여 기기에 접속하여 기기를 관리하고 제어할 수 있게 되지만, 관리하고자 하는 기기를 인터넷을 통하여 제어하려면 기기내부에 인터넷과 연결을 가능하게 하는 물리적인 Device(Network adapter)와 TCP/IP를 제어할 수 있는 Software(Embedded O/S)를 갖추고 인터넷상에서 서버로 동작해야 한다.

이것을 위해서 본 논문에서는 하드웨어 형태의 인터넷 접속장치인 HelloDevice(HD1300)를 채택하였고, HelloDevice는 기기를 제어하는 O/S로부터 인터넷과의 통신부분을 분리해냄으로서 효율적인 관리제어환경을 만들어 주고 기존의 자원을 효과적으로 이용하여 가용성을 높여줄 수 있다. 여기서는 그 적용 예로 태양광용 인버터를 사용하여 관리자가 인터넷을 통해서 HelloDevice에 접속, 인버터를 관리, 제어할 수 있도록 하였다.

2. TCP / IP Protocol

TCP/IP(Transmission Control Protocol/Internet Protocol) 통신 프로토콜은 원래 1970년대 초 DOD(Department of

Defense) Advanced Research Project Agency's Network (APRANET)을 위해 개발되었다. ARPANET은 원래 실험적으로 Packet-Switched Wide Area Network으로서 디자인되었었다. 실험은 매우 성공적이었고, 여러 세월동안 확장되고 다듬어진 프로토콜 스택들을 탄생시켰다. 이들 프로토콜들은 LAN상에서 유용하게 적용되었다. TCP/IP는 다양한 네트워킹 기술을 통해 여러 공급업체 장비사이의 연결을 제공한다. 이는 정의가 잘된 통신 프로토콜과 여러 개의 표준 응용 프로토콜로 구성되어 있다. TCP/IP가 포함된 응용 프로토콜들은 화일전송, 전자메일, 원격로그인, Name Service등을 지원한다.

네트워킹 표준으로서의 TCP/IP가 OSI를 능가하고 상업적으로 확고한 위치를 굳힌 데는 여러가지 이유가 있다.

① TCP/IP 프로토콜은 대체 프로토콜의 ISO 표 준화 이전에 이미 널리 사용되고 있었다.
② 인터넷이 TCP/IP 기반 위에 만들어 졌다. 인 터넷, 특히 World Wide Web(WWW)의 갑작 스런 성장은 TCP/IP의 승리를 더욱 공고히 만들었다.

TCP/IP가 가지는 강점은 다음과 같다.

① license 비용이 무료이고, 표준안 마련에 있어 서 개방적이다.
② 타 장비와의 상호호환성이 가능함으로, 단일 프로토콜이 네트워킹을 단순화시킨다.
③ 개별 계층은 다른 계층에 대해 간섭하지 않으므로 공통된 규약을 정하기가 쉽다.
④ 모든 규모의 장비(네트워크 서브)에 적용시킬 수 있다.
⑤ 동적 라우터기술을 지원한다.
⑥ 클라이언트/서버 컴퓨팅을 지원한다.
⑦ 복수 기술을 지원 (Ethernet, Token Ring, X.25 등)한다.
⑧ 논리적인 주소 할당을 한다. 규모가 큰 네트워크에서도 개별 호스트가 각각의 고유 주소를 가질 수 있다. 이러한 논리적인 주소는 고정된 주소가 아니므로 호스트의 이동 및 주소 재 할당, 네트워크 소속 변경 등이 용이하다.

TCP / IP Protocol은 OSI 모델을 직접 따르지 않는다. 비공식적인 TCP/IP모델이 존재하는데, 이것은 4개의 층을 이용하는 프로토콜을 특성화하는 데 이용할 수 있다.

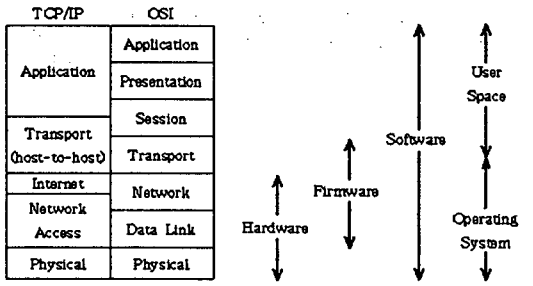

그림 1. TCP/IP 프로토콜 구조

2.1 Application Layer

이 계층은 네트웍을 실제로 사용하는 응용프로그램(FTP, Telnet, SMTP등)으로 이루어진다. OSI 모델에서 보면 애플리케이션 계층과 프리젠테이션 계층에 해당한다.

2.2 Transport Layer

이 계층의 역할은 도착하고자 하는 시스템까지 데이터를 전송하는 것이다. OSI 모델에서 보면 세션계층과 트랜스포트 계층에 해당한다. TCP/IP 소켓부분이 OSI모델의 세션계층에 해당한다. TCP/IP에서는 시스템의 주소(address)와 포트(port)를 가지고 각 Process를 연결해서 통신을 한다. OSI 모델의 트랜스포트에 해당하는 부분은 TCP/IP 프로토콜의 TCP 프로토콜에 해당한다. TCP는 패킷에 에러가 없고, 중복됨이 없이 보내진 순서대로 상대편이 받을 수 있도록 신뢰성이 있는 데이터의 전송을 보장하는 프로토콜이다. TCP는 신뢰성이 있는 만큼 헤더에 각종 에러 코드에 대응할 수 있는 각종 정보를 가지고 있다.

2.3 Internet Layer

이 계층의 역할은 데이터그램을 정의하고 데이터그램을 라우팅시키는 일을 담당한다. 데이터를 정확히 보내는 역할을 하는 것으로 데이터그램이란 IP프로토콜에서 다루는 패킷 데이터를 말한다. 데

이터그램이 가지고 있는 자료는 보낸 주소(source address), 받을 주소(destination address), 보내는 데이터, 그 외 몇 가지 제어 필드(control field)를 가지고 있다. OSI 모델에서는 네트웍계층과 데이터 링크 계층에 해당한다.

2.4 Network Layer

프레임을 물리적인 회선에 올리거나 내려 받는 역할을 한다. 즉, 보내는 쪽(A)는 상위계층에서 받은 패킷을 프레임으로 만들어서 물리적인 회선에 올리게 되고, 받는 쪽(B)는 필요한 데이터를 추출하여 조잡한 프레임을 패킷으로 변환하여 상위계층으로 전달한다.

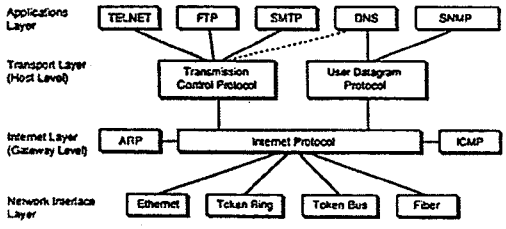

그림2. TCP/IP 프로토콜 Suit

네트워크 상에서 노드와 호스트 사이에서는 오류가 발생하기도 한다. UDP를 사용하는 SNMP(Simple Network Management Control)의 경우 비 연결성(connectionless)으로 신뢰성이 떨어진다. 따라서 사용자에게 정해진 시간이내에 반드시 전달되어야 하는 정보들을 다루는 시스템에서는 적합하지 않은 경우도 있다. TCP는 양방향 전송(전이중: full duplex)이 가능한 가상회선(virtual circuit)을 제공하고 사용자가 느끼는 데이터 전송은 블록단위의 전송이 아니라 데이터 스트림이다. 또, ACK 메시지와 타이머를 이용한 신뢰성 있는 데이터 전송을 보장하며, 슬라이딩 윈도우 방식의 흐름제어를 사용한다. TCP/IP를 사용하면 네트워크에서 일어날 수 있는 여러 가지 문제들을 사용자가 생각하지 않아도 되고 전송이 보장되므로 신뢰성이 필요한 네트워크 관리 시스템에 적합하다고 할 수 있다.

3. 시스템의 구성

3.1.1 인터넷 웹 기반 환경에서의 원격제어 시스템

그림 3은 본 연구에서 사용한 인터넷 웹기반 환경에서의 원격제어시스템의 구성도를 나타낸 것이다. 이 시스템은 관리되는 기기로 인버터를 사용하였으며, 인터넷에 연결하기 위해서 중간매체인 Hellodevice를 사용하였다. 기존에 SNMP를 사용하여 인터넷에 연결하기 위해서는 SNMP Adapter가 필요하고 Adapter와 LAN으로 연결하기 위해서 PC가 따로 필요하다. 하지만 본 연구에서는 Hellodevice가 TCP/IP를 처리함으로써 관리자가 인터넷이 연결된 어디에서든 관리 프로그램으로 Hellodevice에 직접 접속하여 기기를 관리할 수 있게 하였다.

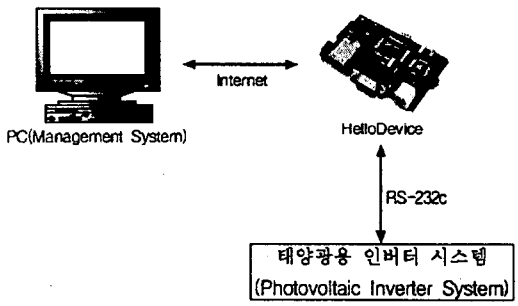

그림 3. Diagram of Remote Control System

3.1.2 관리되는 시스템 (Agent)

관리되는 시스템은 인버터를 예로 들었는데, 기존의 기기들은 직렬통신만을 사용하였다. 직렬통신을 사용하면 거리의 제한이 있고 주로 1:1 통신만이 허용되는 단점이 있었다. 여기서는 거리의 제약과 단일사용자라는 단점을 극복하기 위해서 TCP/IP를 사용하는데 HelloDevice를 이용하였다. HelloDevice는 태양광용 인버터와 RS-232c 통신을 하고 Serial Cable을 통해서 나오는 신호들을 TCP/IP packet으로 변환하여 UTP(Unshielded Serial Cable)선을 통해서 인터넷으로 연결되어 있는 관리시스템에 전송하게 된다. 관리되는 시스템은 관리객체에 대한 정보를 가지고 있으며, 명령에 대한 적절한 응답을 할 수 있도록 되어 있다.

3.1.3 관리 시스템 (Management System)

관리 시스템은 가정에서 사용하는 일반 PC로 하였다. 관리 시스템은 인터넷에 연결되어 있는 Window O/S를 사용하는 모든 컴퓨터에서 사용가능하며 Visual C++로 구현한 프로그램을 설치하여 작업할 수 있고 웹서버로도 동작할 수 있는 Hellodevice의 특징을 이용하여 JAVA를 사용, 별도의 프로그램 설치 없이 웹 브라우져를 통해서도 일반 홈페이지를 보듯이 사용할 수 있게 하였다.

4. 시스템의 구현

태양광용 인버터와 Hellodevice의 연결부분은 직렬 통신인 RS-232로 구현한다. 관리 시스템에 설치될 관리 소프트웨어는 Visual C++를 이용하여 구현한다. 관리자는 인터넷이 연결된 곳이면 어디서든지 시간과 장소에 상관없이 Hellodevice에 접속, 기기를 관리, 제어할 수 있게 된다. 또한 Windows 계열이 아닌 다른 OS(Operating System)에서도 기기를 관리제어할 수 있게 하기 위해서 Hellodevice가 가지는 Web Server기능을 이용하여 일반 홈페이지처럼 구현하였다. 사용자는 Web browser를 통해서도 전용 관리 프로그램을 쓰는 것과 동일하게 기기의 상태를 관리, 제어할 수 있다. 또한 Web browser를 통해서 접속할 경우에는 동시에 여러 사용자가 관리되는 기기의 상태를 확인하고 제어할 수 있게 해 준다. 이를 위해서 Java Applet을 이용하여 관리자 인증을 하고 기기 제어 명령을 Hellodevice를 통해서 관리되는 기기에 전달할 수 있다.

5. Monitoring program 설명

UPSMonitor 1.0은 Visual C++ 환경에서 짜여진 프로그램이다. 동작환경은 Window 98 , Window NT , Window 2000등 Window 계열을 사용하는 모든 컴퓨터에서 설치, 사용할 수 있다. 관리자는 인터넷을 통해서 Hellodevice에 접속, 프로그램을 다운로드하고 설치한 후 기기를 관리할 수 있다.

UPSMonitor를 통해서 알 수 있는 정보들은 다음과 같다.

- Model name
- System status
- Battery charge status
- Input/Output frequency
- Communication status
- Input/Output voltage
- Current
- Battery voltage
- Temperature

UPSMonitor는 관리 기기와 미리 정의된 형식의 데이터를 인터넷을 통해서 통신하고 기기로부터 수신한 데이터를 변환하여 관리자가 알아볼 수 있도록 보여준다. 관리기기로부터 수신되는 원본 데이터 또한 UPSMonitor를 통해서 볼 수 있다. 데이터 형식은 시작문자와 마감문자 필드를 포함하여 12개의 필드로 나누어진다. 시작문자와 종료문자는 1 bit의 character로 정의되고 나머지 필드는 각각 5 bit로 구성되어 매 데이터 수신 때마다 총 52 bit의 문자열을 수신하게 된다. UPSMonitor는 정의된 시작문자로 시작하지 않거나 종료문자로 종료되지 않는 모든 신호들을 무시한다. 또한 수신 문자열 길이가 52 bit가 아닌 모든 데이터도 무시한다. 관리자는 관리기기와의 통신시간 간격을 조절할 수 있고 상태창을 통해서 통신 시 발생되는 에러사항과 데이터의 변화 등을 파일로 저장할 수 있다.

5.1 동작 순서

관리자는 인터넷이 연결된 컴퓨터를 통해서 Hellodevice로 접속한다. 프로그램을 설치 후 Hellodevice의 IP와 port number를 입력하면 관리하고자 하는 기기로 접속할 수 있다. 따라서 관리하고자 하는 기기의 IP만 알고 있으면 인터넷이 연결된 어디서든 기기에 접속, 기기의 상태를 점검하고 모니터링 할 수 있다. 컴퓨터와 Hellodevice는 반드시 인터넷에 연결되어 있어야 하며 Hellodevice 없이 serial cable을 통해서 직접 관리 기기에 접속하는 것은 인터넷을 사용하는 것이 아니기 때문에 구현하지 않았다.

다음 그림은 관리 기기에 접속하기 전의 UPSMonitor 초기화면과 접속상태의 화면이다.

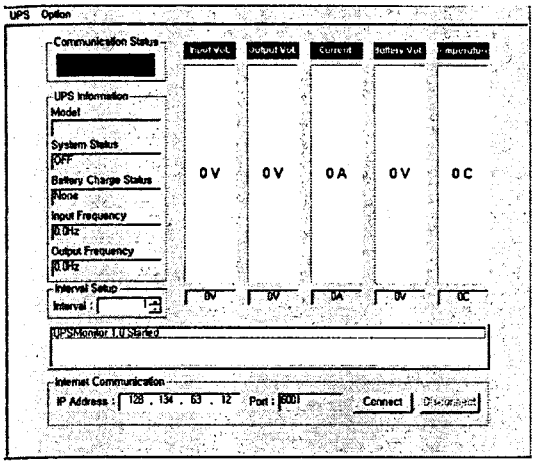

그림 4. 초기화면

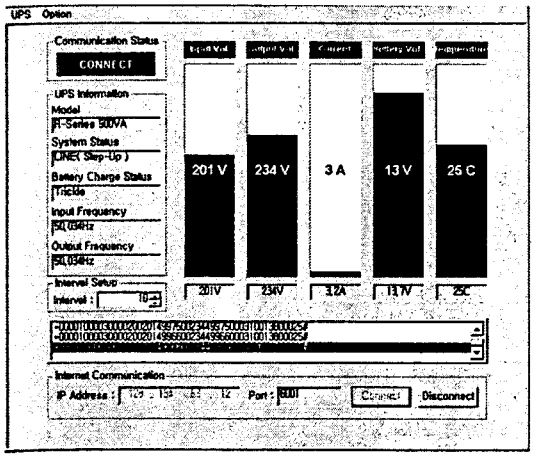

그림 5. 접속 화면 (부하 미인가)

접속이 완료되면 Communication status 창의 글씨가 disconnect에서 connect로 바뀌고 상태창에 socket 연결에 관련된 여러 가지 정보들이 나타난다. 연결이 성공적으로 이루어지면 UPSMonitor는 Interval에 설정되어 있는 시간 간격마다 Hellodevice에 요청신호를 송신하고 관리 기기로부터 수신된 데이터를 변환하여 화면에 표시하게 된다.

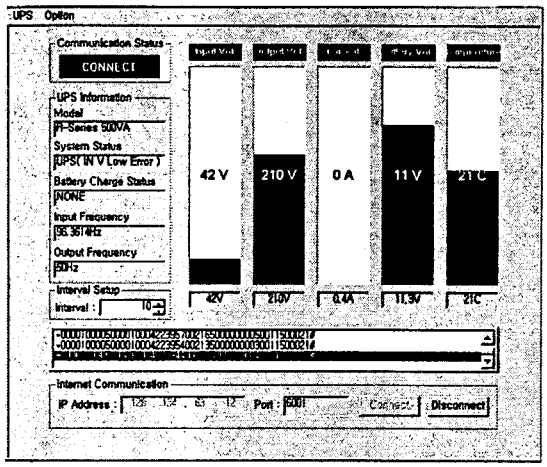

그림 6. 접속 화면 (부하 인가시)

관리자가 disconnect 버튼을 누르게 되면 UPSMonitor는 Hellodevice와의 연결을 끊고 시스템의 자원을 모두 해제 한 뒤 동작을 중지한다. 특성상 Hellodevice 는 한번에 하나의 socket 연결만을 허용한다. 따라서 동시에 여러 지점에서 Hellodevice에 접속할 수는 없으므로 멀티유저 사용으로 인한 데이터의 혼동문제는 일어나지 않는다.

5.2 동작원리

UPSMonitor에서 사용되는 internet통신은 TCP/IP를 사용하므로 연결기반 통신을 한다. 즉 UDP 기반의 SNMP와는 달리 송신 측과 수신 측이 서로 가상의 연결통로를 만들어 놓고 통신을 하는 것이다. 이것을 위해 사용되는 것이 IP와 port number를 결합시킨 구조체인 socket이다. UPSMonitor는 먼저 통신을 하는데 사용될 socket을 생성한다. 다음 connect() 함수를 사용하여 Hellodevice에 이미 대기하고 있는 socket과 연결을 하게 된다. UPSMonitor는 send() 함수를 사용, 연결된 회선을 통해서 요청 데이터를 보내고 receive() 함수를 호출한다. receive() 함수는 연결되어 있는 socket으로부터 날라오는 데이터를 수신하여 buffer에 저장하는 일을 한다. UPSMonitor는 buffer에 저장된 데이터의 무결성을 검사한 뒤 형식에 따라서 문자열을 가공, 화면에 출력한다.

5. 결 론

현대는 인터넷 시대라고 하여도 과언이 아닐 만큼 인터넷은 실생활에 큰 비중을 차지하고 있다. 네트워크를 통한 기기관리 및 제어방식도 인터넷의 출현으로 인해서 큰 변화를 맞이하고 있다. 기기에 이상이 생겼을 경우 관리자가 직접 관리기기가 있는 현장으로 가야만 했던 과거와는 달리 인터넷을 사용함으로서 거의 실시간으로 관리기기의 상태를 파악하고 문제를 해결, 보완함으로서 엄청난 시간과 비용을 절약 할 수 있다. TCP/IP는 이미 인터넷 표준으로서 자리를 확고히 하고 있고 기기를 인터넷에 연결시키기 위해서는 반드시 TCP/IP를 사용하고 자신만의 고유한 IP를 할당받아야 한다. IP 주소 체계가 기존의 IPv4에서 IPv6로 바뀜에 따라서 모든 기기들이 고유의 IP를 할당받고 인터넷을 통해서 제어될 수 있는 기술을 필요로 하고 있다. 전기기기 분야뿐 아니라 모든 분야에서 인터넷을 통해 기기를 직접 제어, 관리하는 기술은 이제 시간과 비용절감의 측면에서 필수적인 요소라고 할 수 있다.

6. 향후 과제

첫째, Hellodevice는 멀티유저를 지원하지 않는다. 따라서 동시에 여러 명이 접속을 할 수는 없다. 만일 어느 한 지점에서 관리자의 실수로 관리기기에 접속을 해놓은 상태로 그냥 방치시켜 놓은 경우, 다른 곳에서는 관리 기기에 접속할 방법이 없으므로 Hellodevice 연결실패가 관리 기기에서 오류를 일으킨 것인지, Hellodevice에 누군가가 접속해 있는 상태인지를 판단할 방법이 없다.

둘째, 관리 기기 자체로는 데이터 로깅 기능이 없으므로 UPSMonitor에서 시간의 흐름에 따른 데이터를 로깅해 주어야 한다. 하지만 관리자가 일정 기간 동안 로깅된 정보를 원할 경우, 특히 관리자가 여러 위치에서 기기를 모니터링하면 데이터를 저장할 수 없다.

[참 고 문 헌]

[1] Douglas Comer, "Internetworking with TCP/IP Vol. I: Principles, Protocols, and Architecture", Prentice Hall, February 22, 2000

[2] Michel Colin & Bernard Sales, "A Protocol Architec- ture For Integrated Management of Local and Large Networks", IEEE, 1993, PP. 1548~1552, 1993

[3] Bob Quinn & Dave Shute & David K. Shute, "Windows Sockets Network Programming" , Addison-Wesley Pub, 1995

[4] Stevens & W. Richard, "TCP/IP Illustrated, Volume I", Addison-Wesley Publishing Company, PP. 350~388, 1994

[5] Larry L. Peterson & Bruce S. Davie, "Computer Networks: A Systems Approach", Morgan Kaufmann Publishers, 1996

[6] Andrew S. Tanenbaum, "Computer Networks. 3rd Edition", Prentice-Hall International Inc., PP. 521~545, 1996

Acknowledgments

본 연구는 한국과학재단 목적기초연(R01-2001-00313)지원으로 수행되었음.