Journal of the Korean Institute of Illuminating and Electrical Installation Engineers Vol. 19, No.5, pp. 103~108 July 2005 논문 19-5-15

구내외 정보통신기기 제어를 위한 Linux System상에서의 UPnP프로토콜 구현

(Implementation of UPnP Protocol on the Linux System for Controlling Premises Equipment)

최동진*
(Dong-Jin Choi)

요 약

본 연구에서는 UPnP(Universal Plug and Play)기술을 이용하여 사용자가 구내외의 제어 가능한 모든 디바이스 즉 조명설비, 냉난방용 보일러, 방범장치와 같은 다양한 기기들을 쉽게 제어할 수 있고, 인터넷이나 휴대폰 등을 이용하여 구내외 어디에서든지 제어할 수 있도록 하였다. 이러한 UPnP프로토콜을 각 디바이스 제조사들은 자신의 기기에 탑재하기 위해 현재 PC시장의 대부분을 차지하는 윈도우즈뿐만 아니라 Linux와 같은 유연한 OS(Operating System)로의 포팅이 필요하다. 그러므로 모든 사용자들이 보다 표준화되고 안정적인 기본 네트워크 위에서 자신이 원하는 기능을 제공받아 사용하기 위하여 본 연구에서는 이러한 기능들을 Linux system에서 구현하였다.

Abstract

In this article, it has been shown that premises devices such as illumination facilities, heating/cooling systems and security equipment can be controlled even outside premises using UPnP (Universal Plug and Play) applicable to the Internet or cellular phone services. To load UPnP protocol into each device, current manufacturers will be required to port flexible OS (Operating System), that is, Windows or Linux to these premises devices. Furthermore, prospective users want to experience a variety of specific functions based on more standardized and stable network. This study aims to provide application examples by implementing these functions on the Linux system.

Key Words : UPnP, Premises device, Illumination facilities

1. 서 론

21세기는 디지털 TV 방송과 IMT-2000서비스의 개시로 유무선 구내외 네트워크와 인터넷을 연동하여 활용할 수 있는 인터넷 정보 가전기기에 많은 관심이 집중되고 있다.

* 주저자 : 경원대학교 전자 · 전기정보공학부 교수
Tel : 031-750-5345, Fax : 031-750-5345
E-mail : djchoi@kyungwon.ac.kr
접수일자 : 2005년 5월 19일
1차심사 : 2005년 5월 24일
심사완료 : 2005년 6월 9일

이러한 인터넷 정보 가전 분야는 구내외 기기들을 연결하기 위한 전화선, 무선 등 네트워크 통신방법에 대한 연구와 네트워크에 연결되는 기기들 간에 상호 운용성을 보장하는 미들웨어 기술 및 네트워크에 연결되는 단말 기술[1]이 주류를 이루고 있다. 구내에 홈 네트워크를 구축하여 다양한 가전기기들을 홈 네트워크에 연결하고, 외부에서 인터넷으로 이를 통제 할 수 있는 시스템을 많은 사용자는 바라고 있다. 예를 들어, 구외에서 오랫동안 구내를 비워 둔 상태에서 구내에 들어올 때 핸드폰을 통하여 인터넷으로 구내의 보일러나, 조명기기 등을 미리 켜고 끌 수 있도록 제어하거나 온수를 미리 준비할 수 도 있다. 이를 위해 보다 다양한 기기들이 구내외 네트워크에 접속 될 것이고 이를 감지하고 제어하기 위한 미들웨어 기술의 중요성은 나날이 더해지고 있다.

미들웨어 기술은 HomeRF[2]나 이더넷, IEEE1394와 같은 유선 기술뿐만 아니라, 블루투스나 IEEE802.11a, IEEE802.11b[3][4][5]와 같은 무선 기술을 통하여 서로 간의 정보교환을 보장하여 주는 기술이다. 미들웨어는 구내외의 가전기기, 정보기기 등을 모두 지원하는 기반 플랫폼 이므로 구내로 들어오는 ADSL이나 케이블모뎀 등을 통하여 인터넷에 연결되며, 미들웨어 하부의 네트워크 물리 매체나 운영체계에 무관한 응용서비스를 제공해야 하고, 구내서버를 중심으로 하여 네트워크에 연결된 다양한 업체에서 개발된 기기들 간의 상호 호환성을 보장하며, 네트워크 자원관리 및 각 기기들을 제어할 수 있는 기능을 제공하여야 한다.

이러한 미들웨어 기술 중 대표적인 기술로 UPnP(Universal Plug and Play)기술[6]과 VHN (Versatile Home Network), HAVi(Home Audio Video interoperability), JAVA를 기반으로 만든 Jini가 있다. 특히, UPnP는 기존에 널리 쓰이는 프로토콜인 TCP/IP와 HTTP, XML등을 이용하여 보다 광범위하고 쉽게 네트워크 구성에 참여할 수 있도록 하고 있다.

가까운 미래에 다양한 기능을 가진 수많은 구내외 정보통신기기들이 수많은 회사에서 나오게 될 것이다. 저자는 이중 UPnP가 기존의 인터넷 범용 통신 프로토콜들을 기반으로 하는 가장 뛰어난 미들웨어가 될 것이라고 생각한다.

이러한 미들웨어인 UPnP를 현재 가장 많이 쓰이는 마이크로소프트사의 윈도우즈뿐만 아니라 오픈 운영체제인 리눅스에서 구내외 정보통신기기들을 컨트롤하기 위한 환경을 구축하고 컨트롤을 실험하였다. 이 연구를 바탕으로 구내외 정보통신기기들의 다양한 컨트롤이 새로 개발될 수 있을 것이다.

2. UPnP 개요

마이크로소프트사의 운영체제인 윈도우즈에 Plug and Play 기능이 추가된 이후로 프린터나 각종 주변기기를 PC에 추가설치하고 설정하는 일이 훨씬 간편해졌다. UPnP(Universal Plug and Play)는 이런 편리함을 구내외 네트워크 전체로 확장시켜 게이트웨이, 프린터 및 각종 전기전자제품들을 쉽게 설치, 인식하고 제어할 수 있도록 설계되었다. 특히 UPnP 규격에 충족되면 윈도우즈 뿐만 아니라 Linux[7], Machintosh, BeOS 등 운영체제가 다르고, 다양한 제조업체에서 만든 이 기종 디바이스들을 모두 사용할 수 있다는 큰 장점을 가지고 있다.

UPnP를 지원하는 디바이스가 구내외 네트워크에 접속하면 IP주소를 얻어서 자신의 존재와 기능을 다른 디바이스 및 컨트롤 포인트에 알려주며 자신의 기능을 수행한다. 구내외 네트워크에서 빠져 나올 때에도 다른 기기에 영향을 주지 않고 쉽게 연결을 해지 할 수 있다. 이러한 모든 과정이 사용자의 작업 없이 자동으로 수행되기 때문에 각 디바이스는 서로 직접 통신할 수 있게 됨으로서 피어 투 피어 (Peer-to-Peer) 네트워킹이 가능하게 되었다.

이러한 구내외 네트워크에 연결되는 디바이스들은 매우 다양하고 지금도 계속 개발되고 있다. PC뿐만 아니라 디지털 TV, 오디오 등의 AV 시스템과 전자렌지, 보일러, 냉장고 등의 생활 전기전자기기 및 PDA, Smart Phone 등의 무선 인터넷 기기와 구내외 조명장치, 방범장치들을 제어하는 센서 네트워킹 기기들과 같이 다양한 많은 기기들을 그림 1과 같이 UPnP네트워크에 접속할 수 있다.

이 많은 기기들은 TCP/IP라는 인터넷 표준 프로토콜을 기반으로 UPnP에 접속되기 때문에 기존의

네트워크와도 쉽게 통합될 수 있고, 업계에서 오랫동안 이러한 프로토콜을 기반으로 하였기 때문에 기존에 제기되었든 문제점과 자료를 쉽게 활용할 수 있는 장점이 있다. UPnP는 사용하는 프로토콜에 의해 정의되는 표준 네트워크 아키텍처를 따르는 분산형태이기 때문에 운영체제, 프로그래밍언어 및 물리적 매체(IEEE1394 등)에 상관없이 독립적이다. UPnP의 프로토콜은 그림 2와 같다.

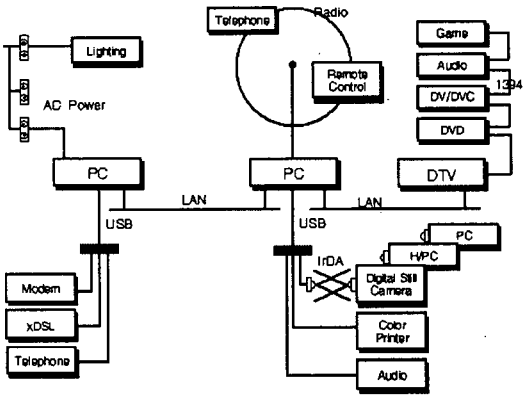

그림 1. 혼합된 미디어와 다양한 제조사로 구성된 네트워크
Fig. 1. A Mixed-Media Multiple-Vendor Network

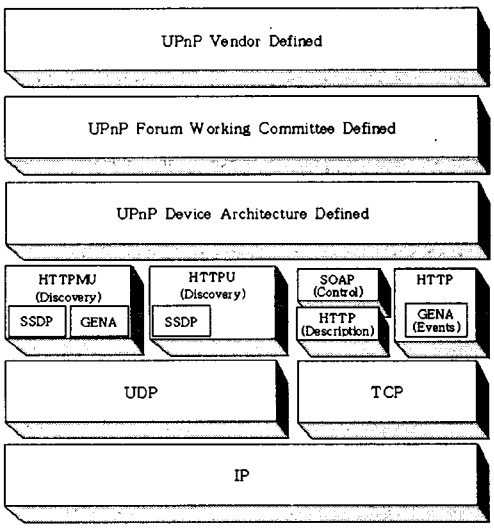

그림 2. UPnP 프로토콜 스택
Fig. 2. The UPnP Protocol Stack

이러한 스택들은 그림 3과 같은 UPnP 네트워킹 구현 단계에서 사용되며, TCP/IP 나 HTTP 등은 인터넷에서 널리 사용되는 프로토콜이다. SSDP (Simple Service Discovery Protocol)는 그림 3의 Step1에 사용되는 프로토콜이며, 그림 4와 같은 SSDP 매커니즘으로 UPnP에서는 그림 5와 같이 동작하여 다른 기기들을 찾게 된다.

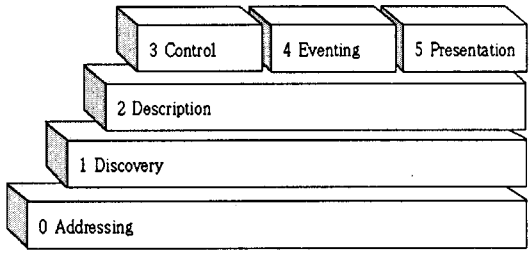

그림 3. UPnP네트워킹 구현 단계
Fig. 3. Steps to UPnP Networking

Step.0 - 컨트롤 포인트와 디바이스가 IP Address를 할당 받음
Step.1 - 컨트롤 포인트가 관심 있는 디바이스를 찾는 과정
Step.2 - 컨트롤 포인터가 디바이스의 기능을 탐지하는 과정
Step.3 - 컨트롤 포인트가 디바이스에 액션을 일으키는 과정
Step.4 - 컨트롤 포인트가 디바이스의 상태변화를 듣는 과정
Step.5 - 컨트롤 포인트가 디바이스와 디바이스의 상태를 HTML UI로 컨트롤하는 과정

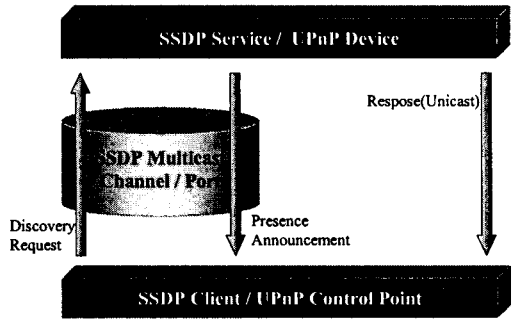

그림 4. SSDP 메커니즘
Fig. 4. A SSDP mechanism

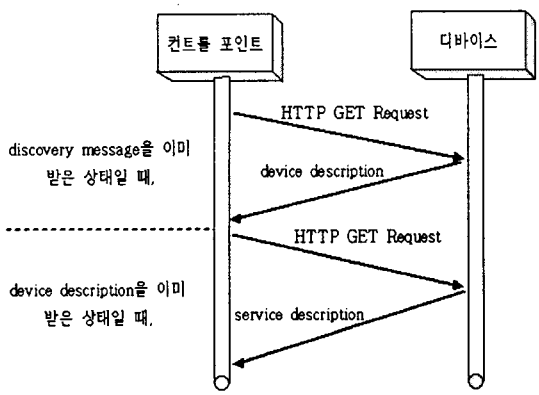

그림 5. 컨트롤 포인트와 디바이스간의 메시지 전달과정
Fig. 5. A message transfer process between control point and device

다음 단계인 그림 6과 같은 GENA포맷은 SSDP를 이용하여 전송될 존재 통보를 만드는 곳에 사용되며 또한, UPnP 이벤트에 대한 서비스 상태의 신호 변경 기능을 제공하기 위하여 사용된다.

이벤트 통보를 수신하는 컨트롤 포인트는 관심 있는 서비스와 이벤트를 보낼 위치, 그리고 이벤트 통보에 대한 구독 시간의 정보가 포함된 요청을 전송하여 이벤트 소스를 구독하게 된다. 이런 구독은 주기적으로 갱신되어 통보를 계속 수신할 수 있어야 하며 GENA를 이용하여 취소할 수도 있다.

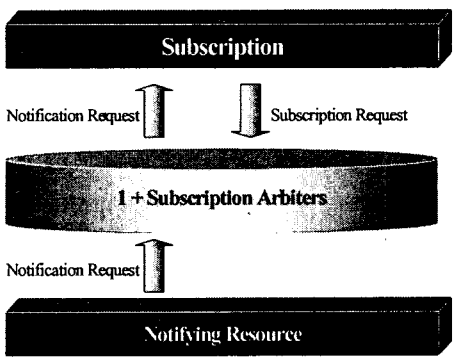

그림 6. GENA 메커니즘
Fig. 6. A GENA mechanism

SOAP은 원격 프로시저 호출을 수행하기 위한 XML 및 HTTP의 사용을 정의한다. 이것은 인터넷을 통한 RPC(Remote Procedure Call)[8] 기반 통신의 표준이 되고 있다. SOAP은 인터넷의 기존 인프라를 이용하기 때문에 방화벽과 프록시에서도 효과적으로 작동한다. 또한 SSL(Secure Sockets Layer)의 보안을 이용할 수 있고 HTTP의 연결 관리 기능을 사용할 수 있기 때문에, 인터넷을 통한 분산 통신 환경을 만드는 것이 웹페이지에 엑세스하는 것만큼이나 쉽다[9].

UPnP는 원격 프로시저 호출과 흡사하게, SOAP를 이용하여 컨트롤 메시지를 디바이스로 전송하며 결과나 에러를 컨트롤 포인트로 리턴한다. 각각의 UPnP컨트롤 요청은 여러 파라미터를 적용시켜 발생시키기 위한 액션이 들어 있는 SOAP메시지이다. 응답 역시 SOAP 메시지이며 상태 변수가 들어 있고 그 값과 리턴 파라미터들을 리턴한다.

3. 실 험

본 연구에서는 한쪽의 컴퓨터를 컨트롤 포인트로 설정하고, 나머지를 디바이스로 설정하였다. 단순히 화면에 보여 지는 측면에서 본 컨트롤 포인트의 기능은 Discovery 결과로 찾아낸 디바이스를 표시하고, Device Description을 파싱하여 디바이스가 제공하는 Service List를 표시하며, 필요한 Service를 선택했을 때, 해당 command를 표시하는 것이다. 컨트롤 포인트 프로그램 내부에서의 UPnP 스펙에 따른 실질적인 처리과정은 그림 7과 같다.

디바이스를 실행시킨 후 컨트롤 포인트 프로그램을 실행시키면 그림 8과 같은 모니터 화면을 볼 수 있다. 화면 첫줄에 네트워크에 존재하는 디바이스를 찾았다는 메시지와 그 아래에 찾은 디바이스 리스트를 보여주고 있다. 현재 네트워크에 2개의 디바이스가 연결되어 있으며 1번의 TV Emulator를 선택하면 그림 9와 같은 서비스 리스트를 볼 수 있다.

위의 과정을 위해 디바이스 리스트를 보여주고 글로벌 변수의 DEVICE STATE와 파라메터로 전달된 서비스 넘버로부터 서비스 구조체를 넘겨준다. Sevice List에서 1번 tvcontrol1을 선택하면 그림 10과 같이 Service되는 command를 선택할 수 있다. 이중 메뉴 1.의 Query를 선택하면 그림 11과 같이 변수 리스트를 볼 수 있고, 그림 12에서 처럼 선택한 변수의 상태를 알 수 있다.

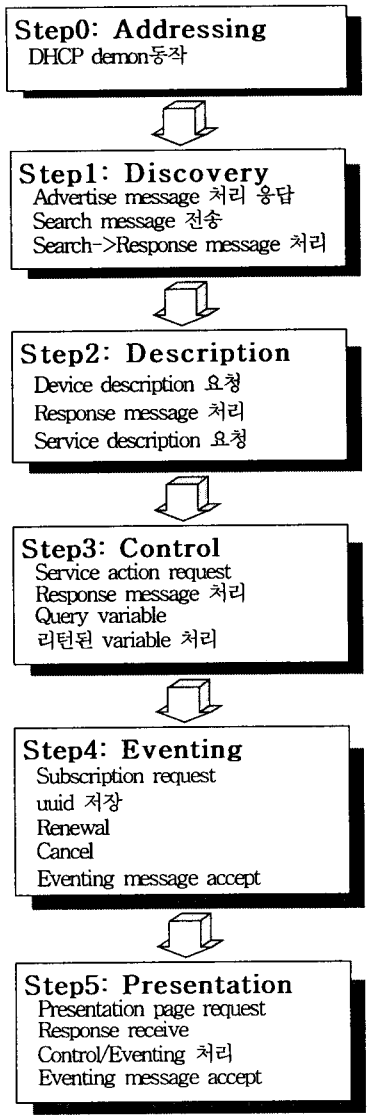

그림 7. 컨트롤포인트의 메카니즘
Fig. 7. Control Point mechanism

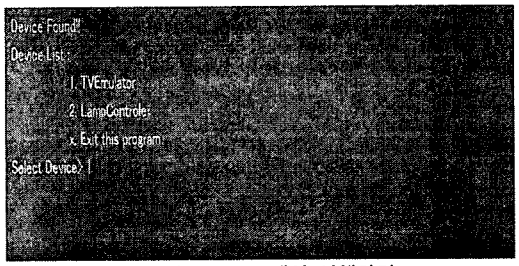

그림 8. 컨트롤포인트 프로그램의 실행화면
Fig. 8. Screen when Control Point Program is excuted

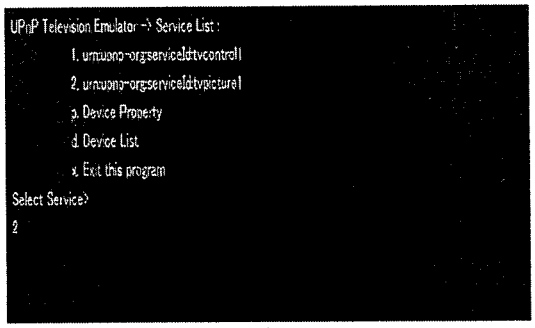

그림 9 디바이스의 서비스 리스트
Fig. 9 Service list of Device

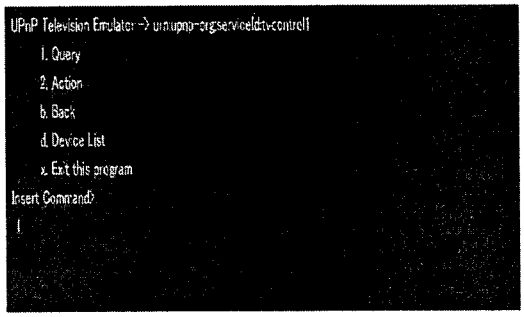

그림 10. 명령어 리스트
Fig. 10. Command List

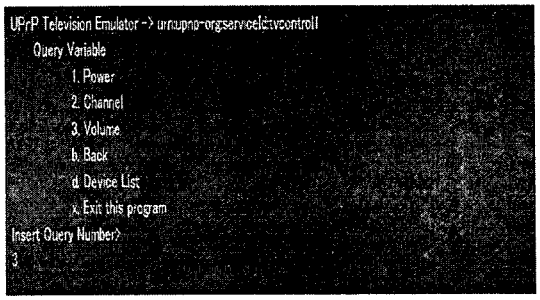

그림 11. 변수 리스트
Fig. 11. Variable List

그림 12. 볼륨 변수상태
Fig. 12. Volume Variable State

만약 그림 13과 같이 "2. Action"를 선택했다면 기기를 제어하기 위해 설정해 놓은 기능들이 실행된다.

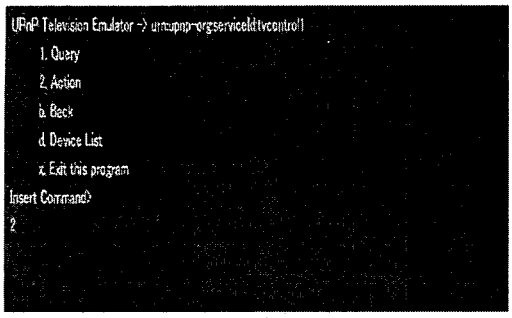

그림 13. Action Invoke 선택
Fig. 13. Action Invoke Selection

4. 결 론

다수의 디바이스와 컨트롤 포인트가 UPnP로 구성된 구내외 네트워크에 존재할 때, 디바이스 프로그램은 Mini Web Server 위에서 동작하며, 자신의 존재를 네트워크에 통보하거나 새로 들어오는 컨트롤 포인트를 인식하고, 자신의 서비스 및 이벤트를 제공하도록 설계되고 실행되었다. 한편 컨트롤 포인트 프로그램은 자신이 네트워크에 접속되었을 때 자신의 존재를 알리고, 관심있는 디바이스를 찾으며, 이 디바이스에 대한 표현을 XML로 받아 파싱(Parsing)하여 원하는 정보를 얻는다. 그리고 이러한 정보를 바탕으로 디바이스를 제어하거나 이벤트를 예약하여 원하는 정보를 계속 얻을 수 있다. 이러한 디바이스와 컨트롤 포인트간의 메시지는 HTTP헤더에 XML을 붙이거나 아니면 HTTP 메시지만을 이용하며, 텍스트(text)형태로 주고 받는다.

본 실험에서는 두 대의 디바이스와 한 대의 컨트롤 포인트만 이더넷으로 연결하여 각각의 기능을 실험했지만 여러 대의 기기를 연결해도 상관없이 정상적으로 동작한다. 이번 실험에서 디지털TV나 램프 컨트롤러와 같은 디바이스가 직접 구현된 것이 아니라 그 기능을 가진 PC로 에뮬레이팅 되긴 하였으나, 만약 실제 기기에서 구현할지라도 그 기기가 리눅스를 사용한다면 각 기능의 디바이스 드라이버 등을 사용하기 때문에 큰 문제없이 본 연구에서 작성된 프로그램을 수정하여 적용할 수 있다.

본 연구에서 작성된 프로그램을 이용하여 구내외 전체의 조명설비와 PDA같은 컨트롤 포인트를 본 연구에서 제시하는 구내외 정보통신 네트워크인 UPnP 네트워크로 구성하여 접속한다면 이 기기들은 리눅스가 아닌 다른 운영체제이거나 다른 유·무선통신방법을 사용할지라도 같은 UPnP기기로서 서로의 기능을 구내외에서 완벽하게 사용할 수 있다. 이것이 미들웨어의 기능이며, 모든 사용자들은 보다 표준화되고 안정적인 기본 네트워크 위에서 자신이 원하는 기능을 제공받아 사용할 수 있게 된다. 본 연구에서는 이러한 기능들을 Linux system에서 구현하였다.

References

[1] "IEEE Standard for a High Performance Serial Bus", Standard for a High Performance Serial Bus, IEEE Std. 1394, 1995.
[2] Caswell, W., "HomeRF Features, Devices, and Positioning," Wireless Home Networking Seminar, Oct. 12, 1999.
[3] ANSI/IEEE Standard 802.11, 1999 edition, IEEE, 1999.
[4] Universal Plug and Play White Paper, Microsoft, 2000.
[5] Jeong K.S., "Home Network as Direct Internet Technology Applications: Universal Plug and Play", Program World, pp.260-272, 2001.
[6] Universal Plug and Play Device Architecture V1.0, Microsoft / UPnP Forum, 2000.
[7] Linux Networking-HOWTO, TCP/IP Network Administration 2nd Ed., Hanbit pub., 1999.
[8] Control and Status Register (CSR) Architecture for Microcomputer Buses , ISO/IEC 13213 : 1994.
[9] Home Networking Technology Workshop, Open Computer Communications Research Society & KICS Data Communications Research Society, 2001.

◇ 저자소개 ◇

최동진 (崔東震)
1945년 8월 6일생. 1972년 한양대학교 전기공학과 졸업. 1980년 연세대학교(석사). 1985년 중앙대학교(박사). 1997년 동경공업대 전자물리과 해외 파견교수(학진). 2004년 Johns Hopkins대 전기공학과 연구교수. 현재 경원대학교 공대 전자·전기정보공학부 교수.