

리눅스 기반의 UPnP 지원 디바이스 드라이버 모듈 설계

고재진, 민수영, 김성진*

전자부품연구원 지능형정보시스템 연구센터

*한국의국어 대학교 컴퓨터 및 정보통신공학과

{[jaejini](mailto:jaejini@keti.re.kr), [minsy](mailto:minsy@keti.re.kr)}@keti.re.kr, *nomarkn1@hotmail.com

A Design of Device Module Supporting Linux-Based Universal Plug and Play

Jae Jin Ko, Soo Young Min, Sung Jin Kim*

*Dept of Computer Science and Engineering, Hankuk University
of Foreign Studies

Intelligent IT System Research Center, Korea Electronics
Technology Institute

요 약

본 논문은 홈 네트워크 기술의 제어 미들웨어인 UPnP (Universal Plug and Play) 프로토콜의 개념을 임베디드 시스템에 도입하여 네트워크 상의 임베디드 시스템으로 구성된 장치들에 프로토콜을 이식하여 네트워크 상에 연결된 전체 시스템으로 기능을 확장하기 위한 개념으로 '리눅스 기반의 UPnP 지원 디바이스 모듈 설계'를 주제로 기술하였다.

1. 서 론

인터넷의 급격한 성장과 대중화는 ISDN, ADSL과 같은 맥내 네트워크 장비의 고속화를 바탕으로 홈 네트워크 기술의 발전을 가져 왔고, 홈 네트워크 기술 중 제어 미들웨어는 네트워크에 연결된 기기들을 발견하고 관리하며 사용자가 이들을 제어할 수 있도록 해주는 것으로 UPnP, Jini, Havi 등이 있다.

리눅스 커널에서 시스템이 지원하는 하드웨어를 응용프로그램에서 사용할 수 있도록 하기 위한 시스템 호출 방식은 시스템이 사용하는 하드웨어가 늘어나고 시스템 호출을 이용하는 응용프로그램의 기능이 발전하면서 파일 형식의 디바이스 드라이버 방식이 추가되었다.

본 논문에서는 리눅스 커널의 디바이스 드라이버에 네트워크 상에 연결된 기기들을 제어 관리하도록 해주는 제어 미들웨어인 UPnP 기술의 개념을 도입하여 네트워크상의 리눅스 시스템에서 동작하는 응

용프로그램으로 하여금 사용할 수 있는 장치의 범위를 해당 시스템에서 네트워크에 연결된 전체 시스템으로 확장하고자 한다. 본 논문의 2장에서는 관련연구로 리눅스 기반의 UPnP 지원 디바이스 모듈에 관련된 국내외 연구에 대하여 기술하며, 3장에서는 리눅스 기반의 UPnP 지원 디바이스 드라이버 모듈 설계에 관하여 기술한다. 마지막으로 4장에서는 결론 및 향후 연구에 대하여 기술한다.

2. 관련연구

2. 1 UPnP

UPnP (Universal Plug and Play)는 정보가전, 무선통신장치, PC관련 장비 등 여러 장소에 분산되어 있는 장치와 서비스간의 쉽고 편리한 통신을 위해 개발되었다.

운영체제에 장치 Plug and Play 기능을 추가함으로써 주변 장치를 PC에 설치, 구성, 추가하는 일이 아주 쉬워졌는데, UPnP는 Plug and Play 기능을 전체 네트워크에 확장하여 네트워크 환경에서 사용자

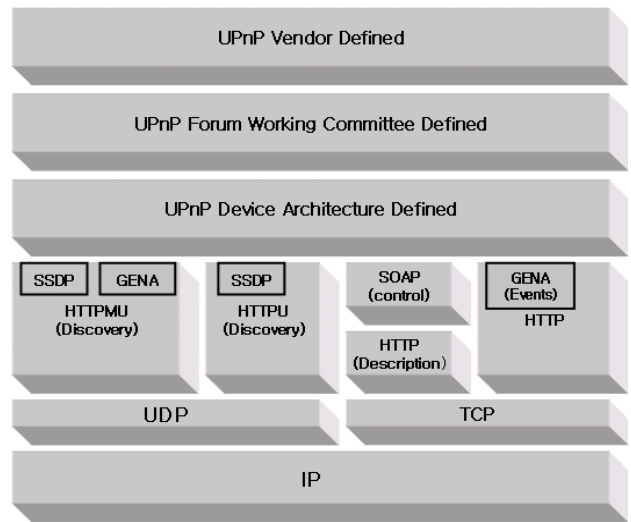
의 특별한 작업 없이 표준화된 방법으로 쉽게 장비 간 연결방안을 제공함으로써, 다양한 공급업체들이 제공하는 수많은 장치들을 자동으로 검색, 제어할 수 있도록 한다. UPnP를 통해 장치는 자동으로 IP 주소를 확보하여 동적으로 네트워크에 합류하며 다른 장치의 존재 여부 및 제공 기능을 확인할 수 있으므로 진정한 zero-configuration-networking을 실행한다. UPnP는 대내망의 여러 디바이스들을 단대단(peer to peer) 방식으로 연결시켜주는 미들웨어 구조로써 특정 운영체제나 프로그래밍 언어, 미디어에 독립적으로 네트워크상의 디바이스에 명령과 제어를 가능하게 한다. UPnP는 IP, TCP, UDP, HTTP, XML과 같은 기존의 프로토콜들을 사용한다. 기존의 프로토콜을 사용함으로써 다음의 장점들을 가질 수 있다.

- 1) 기존의 네트워크에 매끄러운 통합 가능
- 2) 다른 물리적 미디어로 확장용이.
- 3) 실제 복수 공급업체 간의 상호 운용성 보장
- 4) 인터넷과 많은 가정 및 사무실의 인트라넷과 시너지 효과를 기대할 수 있다.

[그림 1]은 UPnP를 구현하기 위해 사용하는 프로토콜들을 도식화한 것으로 장치들 간의 네트워크 연결성을 제공하는 TCP/UDP/IP 프로토콜, UPnP 네트워킹 단계의 일련의 동작들을 위해 사용되는 SSDP, GENA, SOAP 프로토콜, 모든 장치/서비스에 대한 장치 설명서 및 서비스 설명서를 작성하는데 필요한 스키마와 템플릿을 정의한 UPnP Device Architecture Layer, 정의된 스키마, 템플릿을 기반으로 다양한 장치 및 서비스 형태를 표준화하는 UPnP Forum Working Committee Defined Layer, 각각의 공급업체들이 고유한 장치 정보를 추가하는 UPnP Vendor Defined Layer를 계층적으로 표현한 것이다.

2. 2 리눅스 디바이스 드라이버

디바이스 드라이버는 시스템이 지원하는 하드웨어를 응용프로그램에서 사용할 수 있도록 커널에서 제공하는 라이브러리이다. 응용프로그램이 하드웨어를 제어하려면 커널에 자원사용을 요청하고, 커널은 이런 요청에 따라 시스템을 관리한다. 응용프로그램이 커널에게 자원 처리를 요청하는 방법은 크게 두 가지로 하나는 시스템 호출 방식이고 나머지 하나가 바로 파일 입출력 형식을 이용하여 디바이스 드라이버를 사용하는 방식이다. 처음 운영체제가 설계될 당시에는 시스템호출 방식만 존재했지만, 시스템에



[그림 1] UPnP Protocol Stack
 서 사용하는 하드웨어가 늘어나고 시스템 호출을 이용하는 응용프로그램의 기능이 발전하면서, 응용프로그램 개발자들은 하드웨어를 제어하기 위해 알아야 할 시스템 호출 관련 지식이 지속적으로 증가했고 커널 개발자들은 추가 하드웨어에 대한 부담이 커졌다. 그래서 도입된 것이 파일형식의 디바이스 드라이버다.

리눅스는 하드웨어를 다루기 위해 기존의 시스템 호출방식과 병행해서 파일입출력 함수로 하드웨어를 제어하는 유닉스 방식을 계승하고, 몇 가지 기능을 추가하였다. 디바이스 파일을 이용한 하드웨어 제어 방식은 응용프로그램을 개발하는 개발자가 디바이스에 접근하는 방식을 표준화 했고, 디바이스 드라이버를 제작하는 개발자도 일관된 인터페이스에 맞춰 서비스 루틴을 작성하게 하였다.

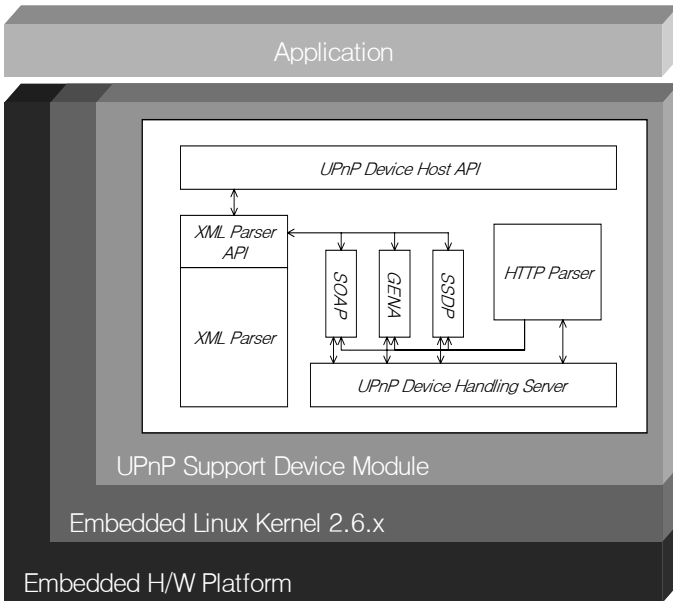
3. 리눅스 기반의 UPnP지원 디바이스모듈설계

리눅스 기반의 UPnP 지원 디바이스 모듈은 리눅스 시스템에서 응용프로그램이 하드웨어를 제어할 수 있도록 하기 위해 사용하는 디바이스 드라이버 모듈에 UPnP 기능을 추가하여 응용프로그램으로 하여금 네트워크 환경에서 쉽고 편리하게 네트워크상의 장치를 사용할 수 있게 한다.

네트워크상의 동작중인 리눅스 시스템에 UPnP 기능을 제공하는 모듈을 동적으로 추가/제거함으로써 응용프로그램이 사용할 수 있는 장치의 선택 범위를 해당 시스템에서 네트워크상의 전체 시스템으로 확대하여 장치 사용의 편의성과 효율성을 높일 수 있다. 다음은 UPnP 지원 디바이스 모듈을 탑재한 내장형 시스템과 디바이스와 애플리케이션 간의 통신 그리고 UPnP 지원 디바이스 모듈 구성에 대해

기술한다.

3.1 리눅스지원 UPnP디바이스모듈탑재시스템



[그림 2] UPnP 지원 디바이스 모듈 탑재 시스템 구성

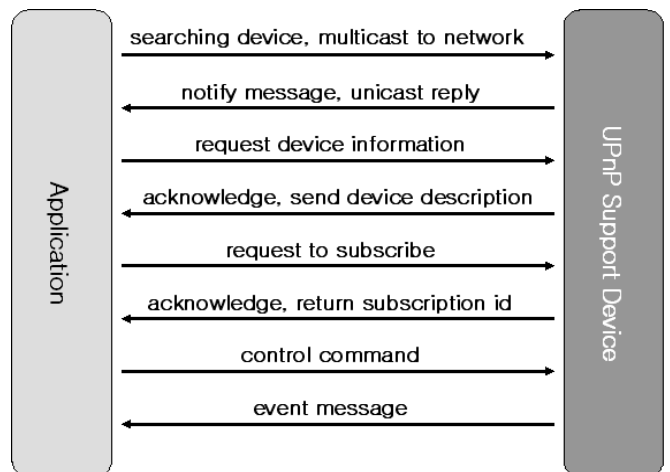
[그림 2]는 리눅스 기반의 UPnP 지원 디바이스 모듈을 탑재한 내장형 시스템의 구성을 도식화한 것이다. UPnP Support Device Module은 Embedded H/W Platform이 지원하는 Embedded Linux Kernel 위에서 동작하며 애플리케이션과의 통신을 위해 여러 가지 프로토콜을 사용한다. Embedded Linux Kernel을 위한 하드웨어 플랫폼은 메모리 관리 유닛 (Memory Management, Unit, MMU)의 유무에 따른 Linux Kernel의 분류에 따른다. 메모리 관리 유닛을 지원하는 일반 리눅스는 가상 메모리를 지원하며 Intel x86 계열, Motorola Power PC 계열, ARM (Xscale, SA-110/1100/1110, 720/940), MIPS, SH 등에 포팅 (Porting)이 가능하고 uClinux는 메모리 관리 유닛이 없는 ARM(7TDMI/9TDMI, 740/940), Motorola 68k계열, Intel i960, AXIS ETRAX 계열에 포팅이 가능하다. 최근 2.6.x 버전이 개발된 임베디드 리눅스는 기본적으로 POSIX를 지원하며, 막강한 네트워크 프로토콜 스택과 멀티태스킹, 사용자 인터페이스로 표준 X나 임베디드용 NanoX 등과 같은 GUI 환경을 지원하고 무료 공개 소스라는 점에서 상당한 장점을 갖는다. UPnP Support Device Module 설계를 위해서는 UPnP Protocol Stack을 비롯한 여러 가지 프로토콜들과 XML Parser, HTTP Parser, UPnP Device Handling Server 등이 필요하다. UPnP 지원 디바이스 모듈 설계를 위해 필요한 프로토콜들은 다음과 같다.

- TCP/IP: UPnP 장치들 간의 네트워크 연결성을 제공, 나머지 모든 UPnP 프로토콜의 기반 역할.
- HTTP, HTTPU, HTTPMU: TCP/UDP/IP기반 메시지 전달, HTTPMU는 멀티캐스트를 뜻하고 HTTPU는 유니 캐스트를 뜻함.
- SSDP: 네트워크 상의 네트워크 서비스를 검색하는 방법을 정의, 네트워크 상태에 관한 정보파악.
- SOAP : 네트워크 상에서 원격 프로시저 호출을 실행, 제어 메시지를 디바이스로 전송.
- GENA : 장비간의 이벤트 교환을 위해 사용, 디바이스의 상태가 변했음을 알림.
- XML : 웹상의 구조화 된 데이터를 위한 범용 포맷, 메시지 및 이벤트 생성에 사용.

위의 프로토콜들은 '3.2 UPnP 지원 디바이스 통신'에서 기술한 디바이스와 애플리케이션 사이의 일련의 통신 과정에서 필요에 따라 사용되어 디바이스와 애플리케이션 간에 통신을 가능하게 한다. XML Parser는 UPnP 지원 디바이스 모듈의 동작 과정에서 광범위하게 사용된다. 디바이스가 애플리케이션에 상세정보를 전송할 때, GENA 프로토콜이 디바이스의 상태변화를 알리는 이벤트 메시지를 생성할 때, SOAP 프로토콜이 디바이스에 제어 메시지를 전송할 때에 XML Parser를 통해 XML 형식의 메시지를 생성하는 역할을 한다.

UPnP Device Handling Server는 디바이스와 애플리케이션의 통신을 위한 모든 네트워킹을 총괄한다. 애플리케이션이 UPnP Device Host API를 통해 보내는 모든 요청과 제어 메시지들을 분석하여 디바이스가 상황에 맞게 동작하도록 하고 그 결과를 전송함으로써 지속적인 통신을 가능하게 한다.

3.2 리눅스 기반의UPnP지원디바이스모듈통신

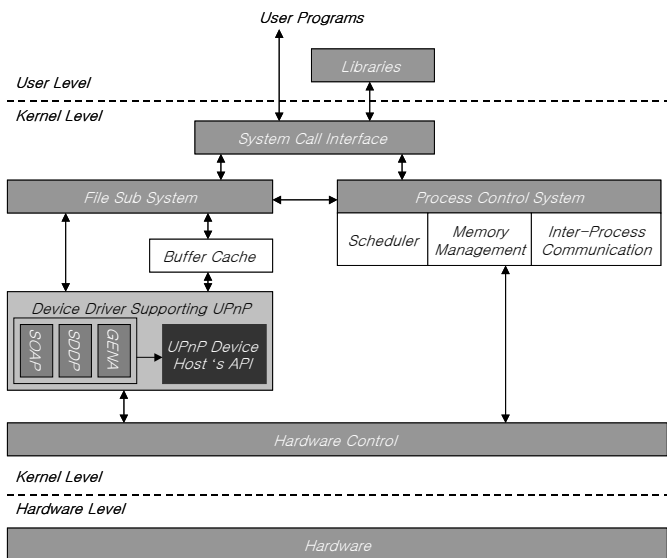


[그림 3] 애플리케이션과 UPnP 지원 디바이스의 통신

리눅스 기반의 UPnP 지원 디바이스와 애플리케이션의 통신을 위해서는 디바이스의 어드레싱 작업이 수행되어야 한다. TCP/IP 프로토콜을 기반으로 디바이스는 주소 지정을 위해 DHCP 서버를 통해 자동으로 IP주소를 할당받거나 SCP프로토콜을 사용한다. [그림 3]은 애플리케이션과 UPnP 지원 디바이스 간의 상호 통신을 도식화한 것이다. UPnP Protocol Stack을 바탕으로 애플리케이션과 디바이스 간 연결방안을 제공하며 상호 통신 과정은 다음과 같다.

- 1) 애플리케이션은 SSDP와 HTTPMU 프로토콜을 사용하여 관심 있는 디바이스를 검색한다.
- 2) HTTPU 프로토콜을 사용하여 일치하는 디바이스는 제어기로 응답하여 자신의 존재를 알린다.
- 3) 애플리케이션은 HTTP 프로토콜을 사용하여 통신을 위해 필요한 디바이스의 상세정보를 요청한다.
- 4) 디바이스는 애플리케이션으로 XML형식의 장비명세를 전송한다.
- 5) 애플리케이션은 디바이스의 상태 변화를 감지하기 위해 HTTP 프로토콜을 사용하여 디바이스에 서비스 등록을 요청한다.
- 6) 디바이스는 요청을 승인한다.
- 7) 애플리케이션은 SOAP와 HTTP 프로토콜을 사용하여 디바이스에 제어명령을 내린다.
- 8) 디바이스는 GENA와 HTTP 프로토콜을 사용하여 제어명령에 따른 이벤트 결과를 애플리케이션에 알린다.

3.3 리눅스 기반의 UPnP지원디바이스모듈구성



[그림 4] UPnP 지원 디바이스 모듈 구성

[그림 4]는 UPnP 기능을 지원하는 디바이스 드라이버 모듈의 구성을 도식화한 것이다.

4. 결론 및 향후과제

본 논문에서는 기존 리눅스 디바이스 드라이버 개발 시 제어 미들웨어인 UPnP 개념을 도입하여 응용프로그램의 장치사용 범위를 확장하고자 리눅스 기반 임베디드 시스템에서 UPnP 지원 디바이스 드라이버 모듈 계층과 통신에 대해 기술하였다. 현재 윈도우를 기반으로 한 UPnP 지원 장치들은 많이 개발되고 있으나 리눅스를 기반으로 한 임베디드 시스템에서의 UPnP 지원 디바이스는 많지 않은 실정이다. 임베디드 시스템에서의 UPnP 지원 디바이스의 의미는 우리가 일상생활에서 쓰는 거의 모든 임베디드 셋탑박스, 디지털 기기 등 폭 넓은 분야에서 네트워크에 연결된 장치를 제어하고 사용하는데 유용하게 쓰일 것이다.

향후 연구과제로는 임베디드 시스템 기반의 UPnP 지원 디바이스를 개발하고자 한다.

참고문헌

- [1] UPnP Forum "UPnP New Letter, Second Quarter, 2002"
- [2] UPnP Forum "UPnP New Letter, Third Quarter, 2002"
- [3] "Printer Device v1.0 and printer Basic Service v1.0" UPnP Standard
- [4] "Internet Gateway Device(IGD) v1.0" UPnP Standard
- [5] A.M Brent, N.Toby, T.Charlie and D.W Mark, "Home networking with Universal Plug and Play", IEEE Communication Magazine, Vol.39, No.2, Dec. 2001, pp.104-109
- [6] "MediaServer V.1.0 and MediaRenderer V 1.0," UPnP Standard
- [7] Linux Kernel API, <http://kernelnewbies.org/documents/kdoc/kernel-api/linuxkernelapi.html>
- [8] Kernel development tool, <http://www.jungo.com/products.html>
- [9] Linux Device Drivers 2nd Edition by Rubini & Corbet, O'Reilly Pub, ISBN 0-596-00008-1
- [10] 유명창, "Linux Device Driver", 한빛미디어, 2004