

LonWorks 을 이용한 디바이스 개발에 관한 연구

이호수*, 장경수**, 신동렬*
*성균관대학교 정보통신 공학부
**경인여자대학 컴퓨터정보기술학부
e-mail : winet@ece.skku.ac.kr

A Study on Development of Device Using LonWorks

Ho-Soo Lee*, Kyung-Soo Jang**, Dong-Ryeol Shin*

* School of Information and Communications SungKyunKwan University

** School of Computer & Information Technology, Kyungin Women's College

요 약

LonWorks 을 이용해 홈 네트워크를 구성할 수 있다. 기존에는 하나의 노드가 되는 조명, 덤머(dimmer), 스위치에 해당하는 디바이스를 중앙 집중식으로 제어할 수 있는 컨트롤러가 개발되어 왔다. 본 논문에서 제시하는 컨트롤러는 스위치와 조명사이에 데이터를 처리하는 중개자의 역할을 하는 중요한 오브젝트이다. 본 논문에서는 이 기술의 핵심인 LonTalk 프로토콜을 설명하고 네트워크에 구성되는 컨트롤러에 대해 주소 지정방식을 이용하여, 여러 개의 조명을 연결될 수 있도록 네트워크 변수 타입을 제시하였다. 뉴런(Neuron)C 로 펌웨어(Firmware)를 구성하여 결과값을 얻을 수 있었고, LonWorks 의 어플리케이션인 LNS(Lonworks Network Service)를 통해 플러그 인 부분까지 구현하여 해당 결과값을 확인할 수 있었다.

1. 서론

최근 홈 네트워크에 대한 부분에 관심이 고조되고 있으며, 그에 따른 네트워크 방식으로 전화선(Phone-line), 전력선(Powerline),무선네트워크(Wireless Network) 형태로 이루어져 있다. 이러한 것 중 전력선을 이용하는 LonWorks 은 제어 네트워크의 미들웨어로 자리잡고 있으며 이를 응용하여 홈 오토메이션 시스템의 구현과 개발이 되어 있고[1], UPnP(Universal Plug and Play)는 표준 인터넷 프로토콜을 근거로 하여 Plug-and-Plug(PnP)방식과 IEEE1394 기술을 응용하여 시스템이 구현되고 있다[2]. 또한, 홈 자바 플랫폼을 기반으로 하는 지니(Jini)가 홈 네트워크를 연결하는 게이트웨이 역할을 할 수 있도록 하는 미들웨어로써 대두되고 있다[3]. 현재, 기존에 LonWorks 을 이용한 홈 오토메이션의 구성을 제시하였지만, 프로토콜의 중요성과 노드를 연결하는 그룹 형태의 방식을 찾아볼 수 없었다. 따라서, 디바이스 컨트롤러의 개발의 목적은

실제 많은 부분에 적용될 조명제어에 대한 컨트롤러에 부합되는 특징을 설명하고, 이를 적용하는 네트워크 변수(Network Variable : NV) 타입을 결정하여 디바이스를 설계하는 방법을 제안하고 그룹 형태의 노드들의 상태변화를 확인하기 위함이다. 또한, 컨트롤러에 연결될 입력과 출력 부분과 각 릴레이(relay)와 조명에 연결할 부분을 제시하였다.

본 논문에서는 기존의 자동제어업체에서 제시한 중앙집중적 제어방식이 아닌 분산형 시스템 형태로 디바이스를 제어하기 위한 주소지정방식과 NV 타입을 보이고, 구현을 통해 상태변화 값을 얻음으로써 동작을 검증하고자 한다.

본 논문의 2 장은 LonWorks 시스템의 이해와 프로토콜의 주소지정 방식에 대해 다룰 것이며, 3 장은 디바이스 컨트롤러에 대한 론마크 오브젝트(LonMark - Object)타입을 기반으로 하여 컨트롤러의 설계를 구성해 볼 것이다. 4 장에서는 구현과 개발에 대한 결과를 다루며 마지막 5 장에서는 결론 및 향후 과제를 제시할 것이다.

2. LonWorks 시스템

LonWorks 시스템은 제어 네트워크이며 빌딩, 산업, 자동 제어분야를 위한 제어 솔루션을 제공하고 있다. LonWorks 네트워크는 기존 중앙 제어식의 마스터 / 슬레이브 방식이 아닌, 피어 투 피어(Peer-to-Peer) 방식으로 디바이스들간에 연결이 이루어져 같은 프로토콜을 사용하여 서로간에 통신이 이루어지고 있다. 그림 1에서 보는 것과 같이 노드와 노드가 연결되어 있고, 각 노드에 따라서 컨트롤러가 될 수 있어 단위 디바이스가 될 수 있는 것이다. 결국 디바이스간에 네트워크로만 연결되어 있다면 중속적인 아키텍처에서 벗어나 개방형 및 분산형 시스템으로 서로간에 제어가 이루어지게 된다.

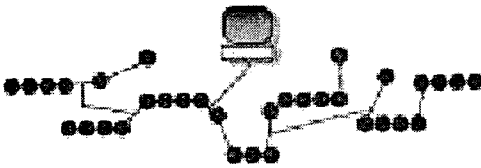


그림 1 Peer-to-Peer

이 방식을 수행할 수 있게 하는 프로토콜은 LonTalk이며, ANSI/EIA 709.1로 제어 네트워크의 표준 프로토콜로 채택 되어있다. 프로토콜의 설계방식은 OSI 방식의 7 계층을 따르고 있으며 두 가지의 중요한 특징을 가지고 있다. 양단에서의 정보를 신뢰성 있게 전달 하는 방식과 데이터의 안전성을 위해서는 16 비트 에러 체크를 하는 것이다. 이런 안정성을 얻기 위해서는 매체접근(Media Access) 기술이 적용되는데 그것이 바로 '부하 예측 알고리즘(Predictive p-persistent CSMA)'이다.

LonTalk 프로토콜은 모든 뉴론 칩(Neuron Chip)에 내장되어 있다. 각 뉴론 칩은 3 개의 8 비트 프로세서, RAM, ROM 그리고 EEPROM 과 같은 보드 메모리에 11 개의 I/O 포트 구성되어 있으며 프로세서 자체에 LonTalk 프로토콜이 내장되어 있기 때문에 실제 개발자는 프로토콜을 구현 할 필요 없이 디바이스 어플리케이션 부분만 작성해주면 된다. 또한, 매체와 통신을 담당할 부분인 LonWorks 트랜시버도 칩 형태로 되어 있어, OSI 7 계층에서 1 계층에 해당되는 물리계층에 해당된다. 주로 많이 사용하는 방식은 2선식을 이용하는 쌍 꼬임선(Twist Pair), 전력선(Power Line), 링크파워(Link Power)로 여러 매체를 지원하는 트랜시버가 있다. 트랜시버의 특징은 네트워크 구조(Topology)에 제한이 없기 때문에 버스, 스타, 링 구조등의 자유롭게 네트워크를 구축하여 배선을 연결 할 수 있다. 또한, 이기종간의 매체는 라우터와 연결하여 LonTalk 방식을 이더넷(Ethernet)을 이용하여 인터넷과 연결시킬 수 있다. 트랜시버의 종류와 매체의 종류, 토폴로지의 관계

는 표 1 과 같다.

표 1 트랜시버의 종류

Type	Transceiver	Bit Rate	Topology
Twisted pair	FTT-10A	78Kbps	Free-Bus, Star, Loop, Other combination
	TP/XF-1250	1.25Mbps	Bus
Link Power	LPT-10	78Kbps	Free-Bus, Star, Loop
Power Line	PLT-20	5Kbps	Free or bus

LonTalk 프로토콜의 주소지정(Addressing)방식에 의하면 주소타입은 물리적주소, 디바이스 주소, 그룹 주소, 브로드캐스트 주소의 형태로 되어있다. 특히, 디바이스 주소는 3 가지로 (도메인 ID, 서브 넷 ID, 노드 ID)로 나누어져 있으며, 그림 2 에서 서브 넷 ID 는 한 채널에 127 개의 노드로, 한 도메인(Domain)안에 255 개의 서브 넷을 구성하여 전체 한 도메인 안에는 32,385 개의 디바이스를 구성할 수 있는 것이다.

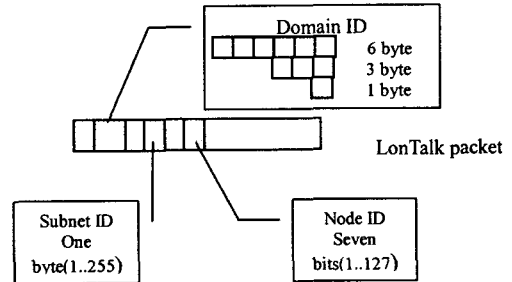


그림 2 LonTalk 어드레싱

그룹주소는 도메인 안에 논리적인 디바이스의 집합이라 할 수 있다. 그룹은 패킷에 따른 네트워크 대역폭(Bandwidth)의 최적화를 위한 효과적인 그룹의 형태는 한 도메인 안에 256 개이다. 시스템 안에서의 도메인 지정은 다음과 같다.

- 서브넷에서 노드-----127
- 도메인에서 서브넷-----255
- 한 도메인에서 노드-----32,385
- 네트워크 안에서 도메인-----2⁴⁸
- 시스템에서의 최대 노드-----32K x 2⁴⁸

이런 LonTalk 프로토콜을 수행하도록 하는 개념은 바로 네트워크 변수(Network Variables : NVs)로, 입출력을 선언할 수 있는 형태이다. 또한, 각 노드간의 정의된 네트워크 변수간의 바인딩(binding) 작업을 통해 네트워크 설계와 구성을 할 수 있다. 바인딩은 그림 3에서 보는 것과 같이 한 노드인 'Switch'의 출력 NV

를 다른 노드의 'Light'의 입력 NV 와 연결하여 동시에 갱신이 되도록 하는 과정이다.

해야 하는데 이중 그룹을 지정하여 노드를 설정하는 변수는 그림 5에 설명하는 SNVT_preset(94)타입이다.

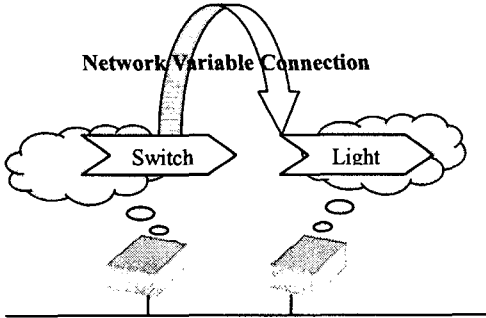


그림 3 네트워크 변수

```

Typedef struct {
    learn_mode_t    learn;
    unsigned long   selector;
    unsigned        value[4];
    unsigned long   day;
    unsigned short  hour;
    unsigned short  minute;
    unsigned short  second;
    unsigned long   millisecond;
} SNVT preset;
    
```

그림 5 SNVT_preset 구조

3. 디바이스 컨트롤러의 설계

하나의 디바이스를 구현하기 위해서는 하드웨어와 소프트웨어 구성을 생각해야 한다. 하드웨어는 뉴런 칩(Neuron Chip)과 트랜시버로 구성되어 있고, 소프트웨어는 뉴런 C로 펌웨어(Firmware)를 구성할 수 있다. 여기서는 하드웨어의 디바이스 부분은 생략하고 소프트웨어적인 LonMark Guildline 를 참고하여 론마크 오브젝트(LonMark Object) 타입으로 컨트롤러를 설계한 부분을 제안 하고자 한다.

이 타입은 센서(sensor)나 액추에이터(actuator)의 오브젝트에 대해 몇 개의 프로그램화 된 값(Value)들에 대해 제어가 가능하도록 되어있으며, 구체적인 내용으로 Selector 의 범위는 16 비트로 첫 번째 그룹지정을 255개로 지정할 수 있고, Value 의 범위는 32 비트로 릴레이(Relay)를 설정할 수 있다. 이 값은 16 진수로 표시해야 하므로 (0x80)은 (1000000) 나타내므로 첫번째 릴레이(Relay)를, (0xC0)은 (11000000)이므로 두 번째 릴레이(Relay)를 가리키므로 해당 값을 얻어올 때 설정된 부분을 확인 할 수가 있다. 따라서, 이 구조체 형태를 이용하여 그룹을 지정하는 주소를 이용해 여러 노드들을 구성할 수 있고, 이 변수를 가지고 실제 연결되어질 노드에 그룹과 패턴(pattern)방식을 적용하여 구성을 할 수 있다. 또한, 노드를 연결하여 설계하는 방식에 따라 해당 표준 네트워크 변수의 설정이 다르게 적용될 수 있다.

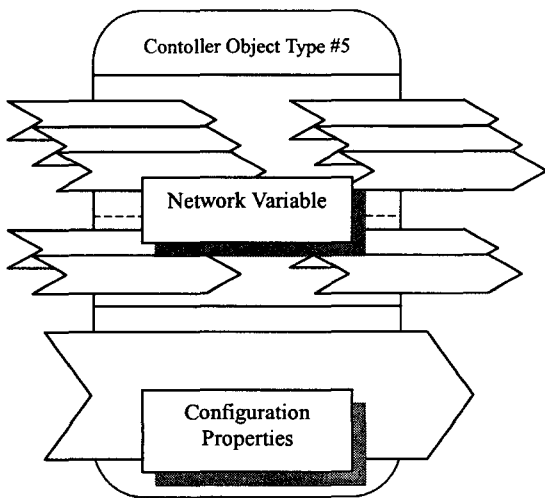


그림 4 컨트롤러 객체

그림 4는 컨트롤러 오브젝트 타입의 형태를 보여주고 있다. 컨트롤러가 어떤 목적으로 쓰일 것인지를 정하고, 목적에 따라서 론마크(LonMark)협회에서 제시하는 표준 네트워크 변수 타입(Standard Network Variable Type : SNVT)을 결정한다. 출력과 입력부분에 해당하는 변수들과 필요로 하는 특정변수를 설정하는 Configuration Properties 부분에 변수를 정한다. 해당 디바이스 컨트롤러에 적용될 표준 네트워크 변수를 정

4. 구현과 개발

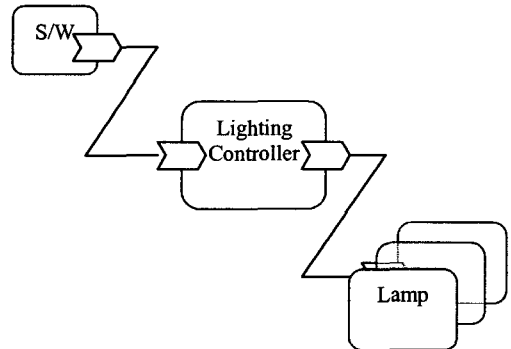


그림 6 컨트롤러 형태

4.1 구현 환경

환경은 그림 6에서 보는 것과 같이 디바이스 컨트롤러의 입력노드에 스위치가 있고 출력노드에는 발광장치(LED)를 구현하였으며 뉴런 칩(Neuron Chip)과 트랜시버를 사용하였다. 표준 네트워크 변수를 바탕으로

뉴런 C 를 이용하여 컨트롤러 디바이스에 들어갈 펌웨어를 작성 하였다. LonWorks 개발 키트(Kit)인 노드 빌더(Nodebuilder)를 이용하여 컴파일 과정을 거친 후, 적용하였던 변수들의 값을 브라우저(browser)로 확인할 수 있었고 XIF(eXternal Interface File) 이미지로 어플리케이션을 이용하여 디바이스와 연결하여 동작이 이루어질 수 있도록 구현 하였다.

4.2 시스템 인터페이스

어플리케이션으로 구현될 LonWorks 의 소프트웨어는 LNS(LonWorks Network Service)로 네트워크상에서 설치, 설정, 모니터 등을 구현 할 수 있도록 되어있다. 그리고, LonWorks 네트워크에 연결된 LNS 의 구조는 그림 7 에 제시하였다.

Configuring & Installing network	Monitor & Control		
LonMaker For Windows (visio embedded)	Into uch	HMI (Human Man Interface)	
	LNS DDE Server	LCA Object Server \$ API	
LNS(LonWorks Network Service)			

그림 7 LNS 요소

LNS 를 기반으로 Active-X 컨트롤 형태로 개발된 것을 플러그 인이라 하는데, LonMaker for Windows 는 PC 상에서 어댑터와 연결되어 LonWorks 네트워크를 이루고 있어 디바이스와 함수를 가져와 그림 8 에서와 같이 디바이스 플러그 인을 불러올 수 있었다.

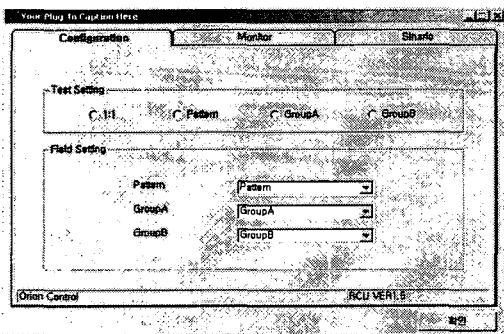


그림 8 디바이스 플러그 인

5. 결론 및 향후 연구과제

디바이스 컨트롤러 개발을 통해 LonWorks 시스템에서 LonTalk 프로토콜의 주소지정방식에 따라서 컨트롤러에 연결될 노드의 동작상태의 결과를 얻었다. 또

한, 네트워크 관리자 입장에서 상태를 파악할 수 있도록 디바이스 플러그 인을 통해서 해당 디바이스의 상태변화를 확인할 수 있어야 한다. 추후 여러 디바이스를 연결하여 각 노드들과 연동되는 부분에 대해 그룹과 패턴형식을 갖춘 컨트롤러를 확장하여 연구하고, 인터넷을 통해서도 제어가 가능한 에슬론(Echelon)에서 개발한 i.LON 1000 인터넷 서버로 웹을 통해서 가 능토록 할 계획이다.

본 연구는 오리온제어기술(주) 지원에 의한 연구 수행 결과의 일부임.

참고문헌

- [1] 손영성, 박준희, 이창은, 문경덕, 김채규, "LonWorks 네트워크를 이용한 원격 홈 오토메이션 시스템", 춘계정보처리학회 학술지, 제 8 권 제 1 호 P.833-836, 2001.
- [2] 김일중, 양수경, 이승학, 안태홍, 박종안, "UpnP 홈 네트워킹 기반 IEEE1394 디지털 가전 제어", 하계한국통신학회 Vol.23 No.1, P.963-966, 2001.
- [3] 구태연, 문경덕, 김채규, "Jini-LonWorks Gateway 에 관한 연구", 추계정보과학회지, 2001.
- [4] 이원열, 서은주, 안병욱, "Home Networking 기술 현황과 전망", 한국통신학회지 제 17 권 11 호 2000.
- [5] 백우진, "LonWorks 기초". <http://www.echelon.co.kr>
- [6] Echelon, <http://www.echelon.com>
- [7] LonMark Interoperability Association, <http://www.lonmark.org>
- [8] Gerard O'driscoll. "The Essential Guide to Home Networking Technologies", Prentice Hall, 2001.