

X10을 이용한 NON-IP 기반 홈 네트워크

안건국* 박상용 민필재 공기석 한익주

한국산업기술대학교 컴퓨터공학과

{naisr00t*, gnoygnas, acemin, kskong, ijhan}@kpu.ac.kr

NON-IP BASED HOME NETWORK USING X10

Gon-Guk Ahn* Sang-Yong Park Phil-Jae Min Ki-Sok Kong Ik-Joo Han

Department of Computer Engineering, Korea Polytechnic University

요 약

현존하는 홈 네트워크 기반의 건물들은 신축 시부터 홈 네트워크 구상 하에 만들어졌지만, 기존의 건물에서 홈 네트워킹을 실현하기 위해서는 리모델링을 해야 하는 어려움이 따른다. 또한 JINI, UPnP, Bluetooth 등의 미들웨어 기술은 IP를 기반으로 하는데, 현재의 IPv4 기반 하에서 IP의 고갈될 전망과 IPv6로 전환되어서도 각각의 가전 기기들이 개개의 IP를 갖는다는 것은 낭비를 초래한다. 이런 문제점의 대안으로 본 논문에서는 NON-IP 기반의 X10 디바이스와 Embedded Gateway를 이용하여 인터넷이 가능한 곳이면 어디에서든 가전기기를 쉽고, 빠르게 제어할 수 있는 환경을 제공한다.

1. 서 론

인터넷의 급속한 발달과 초고속망을 통한 인터넷 보급에 힘입어, 기업이나 공공기관의 중심으로 구축되던 네트워크 환경이택내의 디지털 전자기기로 확산되어가면서 홈 네트워크 산업과 관련기기 시장에 대한 관심이 높아지고 있다. 또한 기존의 유선에서만 가능했던 인터넷을 무선으로 탈바꿈한 무선 인터넷 시대가 도래 하고 있으며, 네트워크 인프라가 잘 구축되어 있는 우리나라에서는 이를 응용하는 기술도 붓물처럼 쏟아져 나오고 있다. 최근 들어 정보 통신기술, 컴퓨터 기술 그리고 가전 기술이 접목되면서 가전기기의 내장형 시스템이 지능화되어 가고 있다. 이른바 정보 가전의 출현이라고 할 수 있는 이러한 추세에 부응하여 정보 가전 기기들의 네트워크 형성 및 인터넷을 통한 제어를 위하여 JINI, UPnP, Bluetooth와 같은 미들웨어 기술들이 나타나게 되었다. 본 논문의 목적은 가전 기기들이 IP를 가지지 않고서도 인터넷을 통하여 제어할 수 있는 전문의 Home Automation 환경을 제공하는 것이다. 이를 위해 X10 전력선 디바이스에 연결된 가전 기기들에 대하여 URL 기반의 인터페이스를 제공하는 프로토콜과 이를 제어하는 Embedded Gateway Server를 구현하였다.

2. 관련 연구

홈 네트워킹 기술은 크게 유선과 무선으로 나눌 수 있으며, 유선기술로는 전화선, 전력선, 이더넷, IEEE1394등이 있고, 무선에는 IEEE802.11x 계열의 무선LAN, HomeRF, Bluetooth 등이 대표적인 기술이다. 홈 오토메이션을 위한 대표적인 표준 홈 오토메이션 네트워크에는 전력선 통신 방식으로 CEBus[1]와 LonWorks[2] 및 EIB[3]가 있다. CEBus(Consumer Electronics Bus)는 EIA(Electronic Industries Alliance)에서 가전제품을 비롯하여 홈 오토메이션 장비들 간에 통신을 위하여 개발한 EIA-600 표준 프로토콜 규격이다. LonWorks는 Echelon에서 공장, 빌딩, 주택, 철도 등의 다양한 분야에서 사용될 수 있도록 개발한 제어용 통신망으로 EIA-709 표준 규격으로 제정되었다. EIB(European Installation Bus)는 빌딩 및 주택에서 온

도, security, 조명, 에너지 시스템 등의 제어를 위하여 유럽에서 개발된 통신망이다. 본 논문과 관련된 연구로는택내의 기존 통신설비를 기반으로 하는 홈 네트워크가 있다. 그중 HomePNA[4]는 기존의 전화선을 이용하여택내에 존재하는 가전기기를 제어하는 것으로서 150m(최대 1.5km)거리의 동작영역을 가지고 있다. 이는 기존영역을 보다 효과적으로 활용하는 방안이어서 3Com, AT&T 에서 활발히 연구 중이다. 그렇지만 HomePNA는 각각의 가전기기들이 전화선과 연결되어야 하는 추가적인 어려움이 있다. 다음으로는 이더넷을 이용한 홈 네트워크가 존재하는데 주로 PC-PC간의 공유의 목적으로 많이 이용된다. 이더넷은 각각의 기기가 IP를 가지고 있어야 하며, IP 설정에 따른 불편함이 존재한다. 본 연구에서는 기존의 전력선을 이용하므로 가전기기 제어를 위한 추가 배선이 필요치 않아 통신 서비스를 위한 망 구축비용을 절감할 수 있다는 점이 가장 큰 특징이다. 또한 X10 디바이스의 HOUSE CODE와 UNIT CODE 만을 간단히 변경함으로써 각각의 가전기기가 독립적으로 제어될 수 있다.

3. 전력선을 이용한 NON-IP 기반의 X10 네트워크

최근 가장 각광을 받는 네트워크 구성 방식중의 하나로 전력선 통신이 거론되고 있으며 현재 활발한 연구가 진행 중이다. 전력선 통신의 가장 큰 특징은 음성 및 데이터 통신이 가능하다는 것인데, 이는택내 전력을 사용하는 모든 가전에 대한 통신 채널을 제공하는 기술로서 그 활용가치가 상대적으로 높다고 할 수 있다. 반면 통신을 위한 상대적인 기술적 어려움이 따르는데, 그 예로 잡음이 많으며 잡음에 대한 대처가 미약하다는 점이다[5][6].

X10 전력선 네트워크는 전력선 모듈이 작고 가벼우며, 제어하고자 하는 가전 기기와 콘센트 사이에 꼽기만 하면 되는 매우 편리한 설치 환경을 제공한다. 또한택내의 전력선을 이용하므로 별도의 통신 인프라 구축이 필요 없고 X10 디바이스는 URL기반으로 동작하므로 IP 사용에 따른 낭비를 줄일 수 있다. 일반적으로 사용자는 그림1과 같이 웹 브라우저를 사용해 Embedded Gateway를 통하여 X10 Network에 연결되어 있는택내의 가전기기를 제어할 수 있다[7].

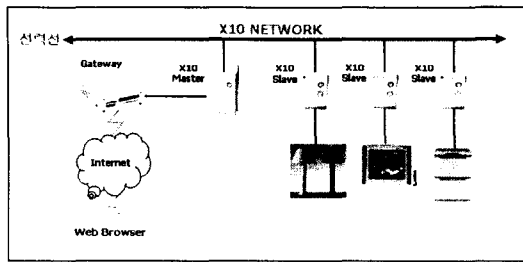


그림1. X10 NETWORK 개요

4. 설계와 구현

4.1 내부 구성도

그림 2에서는 본 고에서 구현한 사용자와 가전기기의 통신을 위한 X10 NETWORK의 내부 구성도이다.

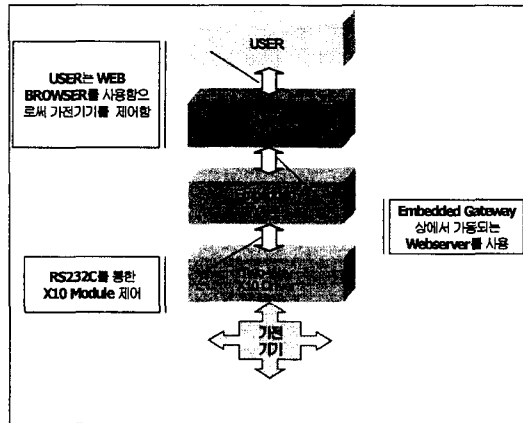


그림2. X10 NETWORK 내부 구성도

내부 구성도는 크게 4 부분으로 나누어지는데, 사용자는 웹 브라우저에서 정해진 명령을 지시하고 Embedded Gateway에서 웹 서버는 그 명령을 해석한 후, 다시 RS-232C를 통해 최종 X10 디바이스에게 명령을 전달한다. 마지막으로 X10 디바이스는 가전기기를 동작시킨다.

4.2 X10 디바이스 제어를 위한 인터페이스 구현

웹 브라우저를 통해 cml1a(X10 Computer Interface)를 제어함으로써 각각의 전력선 모듈에 대하여 독립적인 동작을 보장한다. 제어하기 위한 인터페이스는 실행 속도를 고려하여 HTML과 C 언어로 구현을 하고 내부적으로 System() Call 함수를 사용하여 X10 디바이스 드라이버를 호출하여 실행한다[8][9]. 또한 보안을 위한 인증과정을 통해서 ID를 소유한 사람의 접근을 통제하고, 동시에 여러 사람이 접근하여 하드웨어를 제어하지 못하도록 하였다. 또한 전등이 On/Off 되는 등의 변화가 있을시 마다 데이터를 파일로 기록하여 원격지의 상태를 쉽게 파악할 수 있도록 하였다. 그림3과 같이 Gateway IP를 입력함으로써 사용자 인터페이스의 첫 번째 웹페이지에 접근할 수 있다.

```
http://Gateway-IP/
```

그림3. URL 기반의 Gateway 접근

그림 4는 X10 제어를 위한 인터페이스를 구현한 소스의 일부분이다. QUERYX10 구조체는 URL로 넘어온 Query 값을 각각의 구조체 변수에 저장하여 가전 기기 제어에 이용하고, FILEX10 구조체는 deviceData.conf 파일로부터 읽어 들인 설정 내용을 구조체 변수에 임시 저장하여 이를 사용자 인터페이스에서 사용한다.

```
// QUERYSTRING에서 넘어온 값을 임시 저장하기 위한 구조체
// http://192.168.1.1/cgi-bin/x10.cgi?x10.mode=1&x10.zone=ma&x10.command=mpa&...
// 각각의 값들이 구조체 변수에 대입됨
struct QUERYX10 {
    char    mode[10];      // 현재 모드
    char    zone[10];     // HOUSE_ZONE
    char    command[10];  // DEVICE DRIVER COMMAND(mpa)
    char    action[10];   // TURN, REPEAT
    char    houseCode[10]; // HOUSE CODE : A.B.C.....O.P
    char    unitNumber[10]; // UNIT NUMBER : 1.2.3.....15.16
    char    cmd[10];      // ON / OFF / DIM / BRIGHT
    char    val[10];      // BRIGHT / DIM의 값
};

// deviceData.conf 파일로부터 읽어들이는 내용을 임시 저장하기 위한 구조체
struct FILEX10 {
    char    houseCode[64];
    char    roomName[256];
    char    unitNumber[64];
    char    unitStyle[256];
    char    unitName[256];
    char    unitStatus[64];
    char    unitBright[64];
};
```

그림4. 인터페이스 소스

4.3 EMBEDDED GATEWAY 구현

X10 디바이스는 60Baud의 작은 대역폭을 가지고 있으며 저렴한 설계를 고려한 나머지 컴퓨팅 능력과 저장 능력이 떨어진다는 단점을 가지고 있다. 따라서 X10 디바이스 자체에 웹 페이지를 저장하여 전송한다는 것은 바람직하다. 결론적으로 X10 네트워크와의 통신을 위한 Embedded Gateway를 구현하여 이런 문제를 해결 하였다. 전문의 Home Automation을 위한 Embedded Gateway에는 RedHat Linux 7.1 (Kernel 2.4.18)의 운영체제를 기반으로 gcc (version 2.96) 컴파일러를 사용하여 구현하였고, 어플리케이션으로는 저 용량의 Web Server인 BOA(version. 0.94.13)[10]를 설치하였다. Embedded Gateway와 연결된 cml1a(X10 Computer Interface)는 양방향의 2-Way Controller로서 Gateway에서 Controller로, 또 그 반대 방향으로 신호 전송이 가능하기 때문에 각각의 X10 모듈에 대해 Polling 기능을 수행한다.

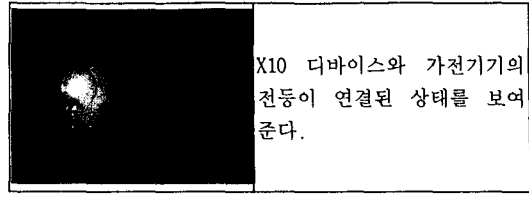
4.4 X10 응용 프로그램 구현

X10 Protocol Spec[11][12]을 기반으로 X10 디바이스와 Embedded Gateway 간의 Serial 통신을 위한 응용 프로그램을 구현하였다. 응용 프로그램은 하드웨어적 성능을 고려하여 C 언어로 프로그래밍 하였으며, gcc (version 2.96) 컴파일러를 사용하였으며 Embedded Gateway에 포팅한 후 Simple Daemon으로 이용하였다. 그림 5는 X10 디바이스를 제어하기 위한 응용 프로그램의 소스 일부이며 Command, House code, Unit code, Value 등의 argument를 사용한다.

```

if (0 == strcmp("on", *(argv+3))) {0 == strcmp("off", *(argv+3))} {
  ch:x_unit = get_unit_code(fd, *(argv+2));
  cm11a_ready(fd, chex_unit);

  chex_header = 0x06;
  chex_hfc=get_function_code(fd, *(argv+3), chex_unit);
  cm11a_function(fd, chex_header, chex_hfc);
}
    
```



X10 디바이스와 가전기기의 전등이 연결된 상태를 보여준다.

그림6. 수행 화면

```

.....
/*
 * 설명: 제어 대상인 cm11a(X10) 장치를 동작 시킨다 (2Way_Communication)
 */
/*
 * 매개변수: _head_code는 제어대상 장치의 코드값 이다.
 */
/*
 * 매개변수: _function_code는 제어대상 장치의 코드값 이다.
 */
.....
int cm11a_function(int _fd, char _header, char _function_code)
{
  char _query[2], _ark=0x00, _compare = 'W0', _checksum;

  _query[0] = _header;
  _query[1] = _function_code;
  write(_fd, &_query[0], 1);
  write(_fd, &_query[1], 1);
  read(_fd, &_checksum, 1);

  while (_compare != 'a') {
    .....
    /*
     * X10이 보낸 체크섬을 확인 후에 필요시 데이터를 재전송 한다.
     */
    /*
     * Note: 이걸 체크섬에서 &0x1f를 한 이유는 꼭 1바이트 크기 체크섬을 얻기 위해서 이다.
     */
    .....
    if ((checksum & 0x1f) == (((_query[0]*_query[1]) & 0x1f)) {
      write(_fd, &_ark, 1);

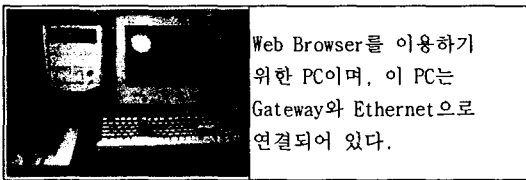
      //usleep(500000);
      read(_fd, &_ark, 1); /* ark 변수 잠시 사용하지 않음 */

      if (_ark == 0x55) {
        _compare = 'a';
      }
    } else {
      write(_fd, &_query[0], 1);
      write(_fd, &_query[1], 1);
      read(_fd, &_checksum, 1);
    }
  }
  return 0;
}
    
```

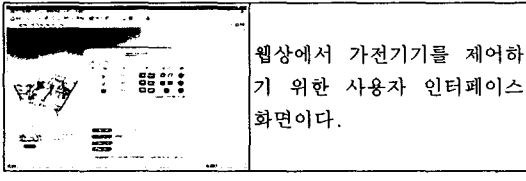
그림5. X10 디바이스를 위한 응용 프로그램 구현

4.5 X10 네트워크와 Embedded Gateway 구현 결과

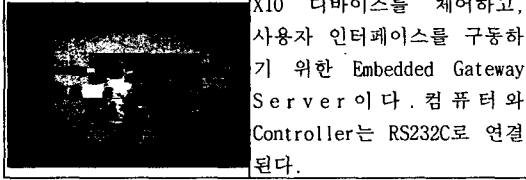
그림 4에서는 사용자가 웹 브라우저를 이용해 Embedded Gateway에 연결된 X10 디바이스를 제어하여 랙내의 전등을 On/Off하는 결과를 보여준다.



Web Browser를 이용하기 위한 PC이며, 이 PC는 Gateway와 Ethernet으로 연결되어 있다.



웹상에서 가전기기를 제어하기 위한 사용자 인터페이스 화면이다.



X10 디바이스를 제어하고, 사용자 인터페이스를 구동하기 위한 Embedded Gateway Server이다. 컴퓨터와 Controller는 RS232C로 연결된다.

6. 결론 및 향후 연구과제

X10 모듈과 이를 제어하는 Embedded Gateway를 구현함으로써 인터넷이 가능한 곳이라면 어디든 랙내의 가전기기를 쉽고 효율적으로 관리할 수 있다. 또한 현재 시공되고 있거나 이미 완공된 건물들은 홈 네트워크의 구상 하에 이루어지고 있지만, 구 건축물에서 홈 네트워크를 실현하고자 한다면 리모델링이라는 과정을 거쳐야 할 것이다. 하지만 본 고에서 언급하는 X10 디바이스를 이용하여 리모델링 과정 없이 제어하고자 하는 가전기기를 단지 X10 디바이스에 연결함으로써 그 목표를 쉽게 이룰 수 있다. 더욱이 각각의 가전기기가 IP를 소유한다는 것은 낭비를 초래할 수 있으므로 가정에서 사용할 수 있는 소형 Embedded Gateway(Home Server)를 이용하여 각각의 가전 기기를 제어 가능하도록 하였다.

본 연구에서는 사용자가 가전기기를 제어한 후 결과 값이 Embedded Gateway 에 파일로 기록이 된다. 하지만 Embedded Gateway의 전원이 Reset 되면, 데이터 파일 또한 초기화 되어 실제 가전기기의 상태와 다를 수 있는 문제점이 있다. 따라서 향후에는 이에 대한 보완과 가전 기기의 스케줄링을 지원하는 연구가 필요하다.

참고 문헌

- [1] CEBUS, "<http://www.cebus.org>".
- [2] LonWorks, "<http://www.lonworks.com>".
- [3] EIB, "<http://www.eiba.com>".
- [4] HomePNA, "<http://www.homepna.org>".
- [5] 이윤철, "최근의 홈 네트워크 기술 동향 및 시장 전망", 주간 기술 동향, 2003 6월.
- [6] 한국정보통신기술협회, "정보통신 및 표준화기술 동향", TTA 저널, 2000년 3월.
- [7] 김세화, "Providing URL-Based Interface for X10 Connected Non-IP Devices", 서울대학교 전기 컴퓨터 공학부, 2000년 9월, "<http://redwood.snu.ac.kr/~ksaehwa/00-fall-adv-internet/final-report.pdf>".
- [8] Korea Embedded Linux Project, "<http://www.kelp.or.kr>".
- [9] Korea Embedded System of Linux, "<http://www.kesl.org>".
- [10] BOA Web Server, "<http://www.boa.org>".
- [11] X10 Protocol Description, "<http://heyu.tanj.com/heyu/protocol.txt>".
- [12] X10 Community, "<http://www.x10.com>".