

---

# 레거시 기기 통합 제어를 위한 효율적 Zigbee Bridge 개발

이승민\* · 손성용\*\*

The Development of An Efficient Zigbee Bridge for Legacy Device Control

Seung-min Lee\* · Sung-yong Son\*\*

---

이 논문은 지식경제부 산업원천기술개발사업(10035359)에 의해 지원되었음

---

## 요 약

가정에서 사용중인 가전제품을 제어하는 방법은 적외선 리모컨을 이용한 방식이 대부분을 차지하고 있다. 이러한 레거시 기기들은 각각의 리모컨으로 제어되고 있고 표준화된 통신방식을 제공하지 않아 원격 제어에 어려움이 있다. 통합 리모컨방식의 원격 제어를 이용할 경우 여러 종류의 기기를 제어할 수 있지만 모든 명령을 지원하지는 않으며, 특히 새로운 명령이나 고유한 명령을 가진 제품이 출시될 경우 확장성에 한계가 있다. 본 논문에서는 레거시 기기의 효율적인 통합 제어를 위해 지그비로 전송되는 명령을 해당 기기의 적외선 제어 명령으로 변환하여 발신하는 Zigbee Bridge를 개발하고, 헤더 판별법을 이용한 학습 기능을 추가 하였다. 새로운 명령이 전송되는 경우 Bridge는 이 명령을 학습하여 저장하며, 기존 명령이 전송되는 경우 저장된 코드를 출력하는 방식을 사용하였다. 이 방식을 사용하는 경우 신규 명령에 대한 확장성과 유연성을 확보 할 수 있으며, 코드 전송의 효율을 향상하여 기존 대비 최대 58.6배 빠른 전송이 가능하다.

## ABSTRACT

IR remote control is the most common way to control home appliances nowadays. Each legacy device is controlled with their own remote controller, and it is not standardized yet. Although it is possible to handle multiple devices with a integrated remote controller, not all the devices and command sets are supported. There are also limitations in expandability especially when new or special command sets are required. In this paper, a efficient zigbee bridge which transforms a command set sent via zigbee to IR comment set is developed. The bridge also has a learning function based on a header comparison method. When a new command is sent, the bridge learn the new code. When a existing code is sent, the bridge dispatches applicable IR command to the appliance. With this approach, it is possible to obtain expandability and flexibility for new commands, and increase the communication efficiency up to 56.8 times.

## 키워드

지그비 브릿지, 레거시 기기, 통합 리모컨, 학습 리모컨

## Key word

Zigbee Bridge, Legacy Device, Integrated Remote Controller, Learning Remote Controller

---

\* 경원대학교

\*\* 경원대학교 (교신저자, xtra@kyungwon.ac.kr)

접수일자 : 2010. 06. 30

심사완료일자 : 2010. 07. 23

## I. 서 론

스마트홈 기술의 발달로 어디서나 자유롭게 맥내 기기들의 제어가 가능해지고 더 많은 기기의 제어 관리를 위해 지속적인 보급 확대가 이루어지고 있다. 스마트 홈을 기반으로 맥내 환경의 체감적 손실 없이 에너지와 자원의 지능형 관리를 통하여 CO2 발생, 에너지 / 자원 소비 및 사용자 비용 최적화를 지향하는 방향으로 기술이 개발되고 있다[1]. 하지만 여전히 TV, 에어컨, 공기청정기 등과 같이 적외선 방식의 리모컨으로 제어되는 기존의 레거시 기기들이 상존하고 있으며 앞으로도 정부 규격 또는 표준이 확정되어 지거나 레거시 기기에 Zigbee 수신 모듈을 장착하여 제어하는 등의 방법을 사용하지 않는다면 기존 레거시 기기는 제어할 수 없는 문제가 있다.

이러한 상황에서 원격의 레거시 기기를 통합 제어하고자 할 때 추가 비용을 들이지 않고 가장 손쉽게 접근할 수 있는 방법은 각 기기에 해당 하는 리모컨 신호를 동일하게 재생해 주는 방법이다.

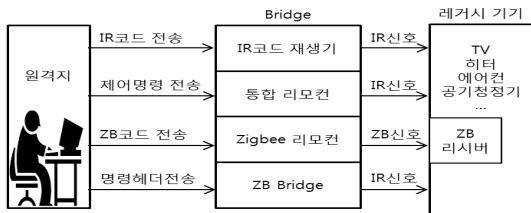


그림 1. 레거시 기기의 제어 방법  
Fig. 1 Control Method of Legacy device

기존의 리모컨 신호를 전송하기 위해 다양한 방법의 연구가 진행되어왔고 개별 리모컨을 통합하기 위한 대표적인 방법들은 그림 1과 같다. 첫 번째는 원격지에서 IR코드 재생기로 IR코드를 전송하면 전송받은 명령을 그대로 레거시 기기에 전달하여 제어하는 방법[2], 두 번째는 통합 리모컨으로 제어명령을 전송하면 리모컨에 저장되어 있는 명령을 검색하여 검색된 IR코드를 출력하는 방법[3], 세 번째는 레거시 기기가 Zigbee 수신기를 가지고 있을 때 사용 가능한 방법으로 원격지에서 Zigbee 코드를 전송하면 Zigbee 리모컨은 이 코드를 IR 신호로 변경하지 않고 Zigbee코드 그대로 기기에 전송하

는 방법이다[4].

본 논문에서 제안하는 방식인 마지막 방법은 위 방법들의 문제점인 긴 패킷 길이로 인한 패킷 손실 및 지연, 메모리 공간 부족과 수신부 Zigbee 리시버를 추가해야 하는 문제를 보완하였다. 매우 짧은 길이의 명령헤더를 Zigbee Bridge에 전송하고 만약 명령을 가지고 있으면 해당 신호를 출력하고 없을 경우 초기 1회 IR코드의 학습을 통해 메모리에 저장한 뒤 추후 같은 명령을 요청하면 헤더를 판단하여 저장된 명령을 출력하는 방법을 사용한다. 단 메모리가 부족할 경우 학습한 명령을 기존 저장 공간에 저장되 빈도순 정렬로 자주 사용하지 않는 기능을 삭제하여 그 공간에 저장한다. 제안하는 방법과 같이 운용할 경우 제어 신호의 손실 및 지연을 최소화하며 한정된 메모리 공간에서의 빠른 신호 출력과 최소한의 에너지 소모가 가능하다.

## II. 관련연구

### 2.1 통합리모컨

통합리모컨은 여러 가전기기 제조사의 제어 명령을 하나의 리모컨에서 제어할 수 있도록 만든 방식이다. 여러 제조사의 제품을 통합 제어하기 위한 제어신호를 리모컨 안에 미리 탑재하여 생산한다. 리모컨의 해당 제어 명령을 선택하면 펌웨어에 미리 저장되어 있던 명령을 검색하여 출력하는 방식으로 레거시 기기를 제어할 수 있다[5]. 가전 업체별로 각기 다른 프로토콜을 만들어 사용하므로 펌웨어에 저장되어 있지 않은 제품의 명령이나 새로 출시되는 제품 등 모든 가전기기를 제어할 수 없는 문제가 있다. 일반적으로 많이 사용하는 프로토콜로는 NEC Protocol, Sony Protocol 등이 있다. 이와 같은 소형 임베디드 기기는 한정된 물리적 공간과 배터리를 주전원으로 이용하므로 저전력 소모를 위한 설계가 요구된다.

### 2.2 학습리모컨

모든 제어 명령을 수용할 수 없는 구조인 통합리모컨의 문제를 해결한 방법으로 학습 가능한 방식의 리모컨이 있다[3].

이 방법은 프로토콜의 구조와 상관없이 원 신호를 그

대로 학습하여 출력하므로 모든 명령을 수용할 수 있다. 단 초기 1회 원하는 명령을 학습시키는 과정을 거쳐야 한다. 원 리모컨 신호가 시작하는 시점부터 신호가 끝나는 시점까지의 출력신호를 샘플링하여 저장한다. 각기 다른 프로토콜의 명령 신호를 샘플링 하여야 하므로 보다 정확한 결과를 얻기 위해서 샘플링 주기가 가장 짧은 프로토콜에 맞춰져야 한다. 샘플링주기와 신호 길이에 따라 학습된 명령의 크기는 커지고 저장 공간을 많이 차지하게 되어 많은 명령을 저장하기 위해서는 저장할 수 있는 메모리의 크기가 커져야 하는 단점이 있다. 또한 학습된 명령을 무선으로 전송할 때 명령의 크기에 따라 2~3회 나누어 전송해야 하는 문제가 발생하여 명령 출력 시간의 지연과 패킷 손실 등의 문제가 있다.

### 2.3 IR코드 재생기

원격지에서 제어하고자 하는 코드를 IR코드 재생기에 직접 전송하거나 홈서버에 저장되어 있는 명령을 전송 요청할 때 사용하는 것이 IR코드 재생기이다[2]. IR 코드 재생기는 리모컨에 별도의 명령을 저장하지 않고 원격에서 들어오는 명령을 받아 출력만 하는 기능을 한다. 명령을 호출할 때마다 명령코드 전체를 보내야하므로 무선으로 전송시 학습리모컨과 같이 패킷손실과 출력 지연 등의 문제가 있다.

### 2.4 Zigbee 리모컨

송수신부 모두 Zigbee통신을 이용하여 데이터를 주고 받는다[4]. 양방향일 가능하므로 가전기기의 상태와 메시지를 리모컨에서 받을 수 있어 차후 애플리케이션을 추가하거나 관리할 때 용이하다. 따로 IR신호를 출력할 수 있는 부분이 없기 때문에 일반적인 레거시 기기에서는 동작하지 않고 Zigbee 신호로만 통신할 수 있다. 레거시 기기의 수신부에 Zigbee 수신부 모듈이 있어야 데이터를 수신할 수 있으므로 가전기기가 출시될 때 수신부를 포함하여 출시하여야 한다. 가전제품 제조원가 상승의 문제가 있다.

## III. 테스트 환경 구축

제안된 방식의 테스트를 위하여 그림 2와 같이 이더넷 패킷을 Zigbee로 전달할 홈서버와 Zigbee 패킷 송수신

부인 코디네이터, Zigbee 패킷을 IR신호로 출력하여 레거시 기기를 제어할 Zigbee Bridge로 구성된 테스트 환경을 구축하였다.

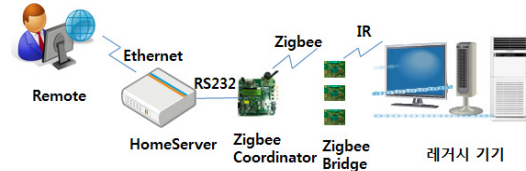


그림 2. 환경 구성도  
Fig. 2 Environment configuration

원격지에서 인터넷을 통해 명령을 전송하면 집 안의 홈서버가 이 명령을 받아 지정된 Packet 형태로 전송한다. Zigbee Coord (Coordinator)는 Zigbee 통신을 이용하여 Zigbee Bridge에게 이 명령을 전송한다. 이 때 Zigbee Bridge가 이 명령에 대한 명령 코드를 가지고 있으면 최종 목적지인 레거시 기기(사용자 가전제품)에게 IR(Infrared) Signal을 출력한다. 가전제품은 이 명령에 응답하여 해당 동작을 수행한다. 만약 명령 코드가 없으면 Zigbee Coord에게 코드 없음을 알리고 홈서버로부터 명령을 수신하여 출력한다. 홈서버가 Zigbee 모듈을 포함하고 있다면 Zigbee Coord는 따로 두지 않아도 된다. 본 논문에서는 Zigbee 통신 기능이 없는 홈서버를 사용하였으므로 Zigbee Coord를 추가로 배치하였다. Zigbee Coord는 원격에서 인터넷을 통해 홈서버로 들어온 명령을 시리얼로 받아 Zigbee통신을 이용, Zigbee Bridge에게 출력해 주는 기능만 한다. Zigbee Bridge는 제어 대상 디바이스의 IR 수신부에 부착하여 외란이나 감쇄로 인한 영향을 최소화한다.

## IV. 학습 가능한 리모컨 설계

### 4.1 IR 명령코드의 신호 구조

그림 3에 나타난 바와 같이 IR Remote Control 신호는 3개의 Layer로 구성되는 경향이 있다. Layer 3는 IR 신호의 Carrier 역할을 담당하는데 일반적으로 IR 영역인 36kHz ~ 40kHz의 IR Frequency를 사용한다. Layer 2는 실질적인 제어 신호를 담고 있는 Signal Envelope 영역으로 Pulse와 Gap의 Sequence로 구성된다.

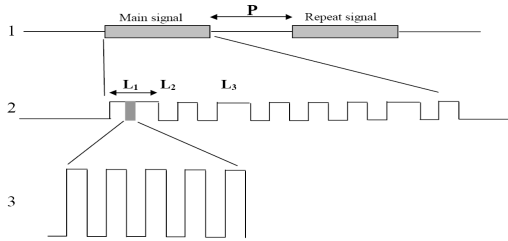


그림 3. IR Remote Control Signal의 Layer 구성  
Fig. 3 Layer structure of IR Remote Control Signal

각각의 Pulse와 Gap의 지속시간 (Duration)이 Signal의 특성을 결정짓게 된다. Layer 1은 Main & Repeat Signal 부로 송신의 안정성을 확보하기 위하여 사용하는 Layer이다. 일반적으로 Layer 2의 신호는 한번의 제어 명령에 대하여 1회 생성되어 전송되므로 Layer 1의 신호는 Layer 2와 동일하게 작동하게 된다. 그러나, 장비에 따라 Layer 2의 신호를 일정 시간의 휴지기를 가지며 2~3회 반복하는 경우도 있다. 이러한 경우 수신 기기의 작동 방식은 2가지로 나뉘게 되는데, Layer 1에서 Layer 2의 신호를 n회 반복하는 경우 n 회중 1회만 수신이 되어도 작동이 되는 경우와, n 회 모두 수신이 되어야 작동하는 경우가 그것이다.

#### 4.2 데이터 패킷 설계

본 논문에서는 사용자가 자주 사용하는 명령과 최근 사용한 명령들을 Zigbee Bridge에 아래와 같은 구조를 사용하여 저장하였다.

```
typedef struct {
    UINT8    deviceType;  UINT8    CMDType;
    UINT8    uTimeLen;   UINT16   uTime[8];
    UINT8    mSigLen;   UINT8    mSignal[250];
    UINT8    nRepeat;   UINT16   rGap;
    UINT8    rSigLen;   UINT8    *rSignal;
} DeviceCmd_t;
```

구조 중 deviceType과 CMDType는 Zigbee Bridge에서 저장하고 있는 명령들을 검색하고 관리하기 위해 제공되는 항목이다. 제어를 원하는 레거시 기기의 장치와 명령의 타입을 선택하면 해당 명령이 IR로 출력되어 제어할 수 있도록 설계 하였다. 레거시 기기 제조사마다 IR 신호를 출력할 때 사용하는 프로토콜이 다르므로 해당 신호의 출력을 위해 신호의 출력 시간을 인지할 필요가 있다. 해당 단위 시간동안 출력할 리모컨 신호의 길이를 uTime에 저장하고, 저장된 uTime 단위 길이의 종류가 몇 개인지 파악하기 위해 uTimeLen에 uTime의 개수를 저장한다. mSignal과 mSignalLen은 uTime의 조합으로 나타내어지는 Main Signal의 Sequence와 신호의 길이를 나타낸다. uTime의 단위 시간동안 실제 출력될 IR신호의 Data가 mSignal에 저장되어 있다. 이 신호의 길이는 기본적으로 uTime의 단위 시간 길이가 짧으면 더 늘어나고 단위 시간이 길어지면 줄어든다. 출력될 패킷의 길이는 최대 250Byte 이하이고 mSignal의 개수는 mSignalLen에 저장된다. mSignal의 신호가 반복되는 경우 nRepeat는 반복 횟수를 표시하며, Main Signal과 차이가 있을 경우를 나타내기 위하여 rSignal과 rSigLen가 사용된다. rGap은 Repeat Signal이 있는 경우 앞 신호의 종료후 Repeat Signal이 발생되기까지의 Delay 시간을 의미한다.

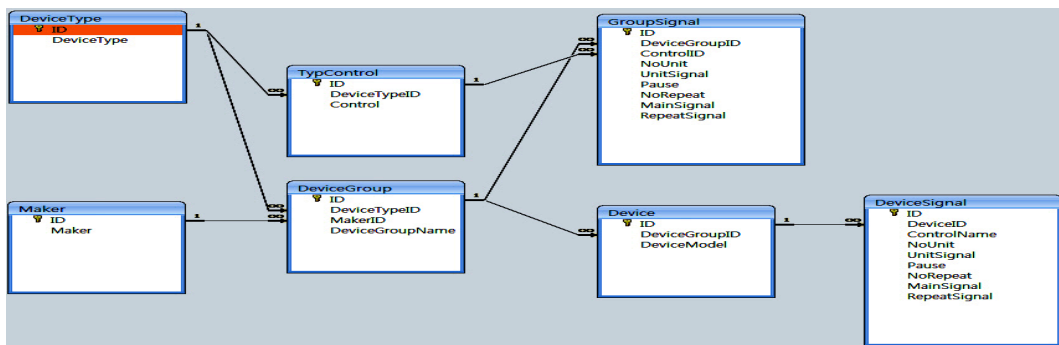


그림 4. DB구성도  
Fig. 4 DB Structure

### 4.3 효율적인 명령코드 저장 방법

홈서버에 구축되는 통합DB는 그림 4와 같이 설계하였다. 입력받는 장치의 타입(DeviceType) 과 제조사의 메이커(Maker)를 바탕으로 같은 그룹의 장치를 제어하고자 할 때 타입별로 저장할 명령(TypControl)을 구분하고 해당되는 그룹 (DeviceGroup) 신호의 공통 저장을 통해 그룹 제어신호를 공유할 수 있다. 같은 제조사 다른 모델의 제품인 경우에도 특별한 고유 명령이 아닌 제품의 기본이 되는 제어명령과 같은 경우는 동일한 경우가 많기 때문에 그룹 명령을 통해 그룹 제어 신호를 저장한다. 명령 공유를 통해 한정된 메모리의 저장공간을 효율적으로 사용할 수 있다. 장치별 개별 제어 신호는 디바이스 모델명(Device) 별로 디바이스 제어 신호를 저장하여 출력되도록 한다.



그림 5. IR신호 입력기  
Fig. 5 IR Signal Input screen

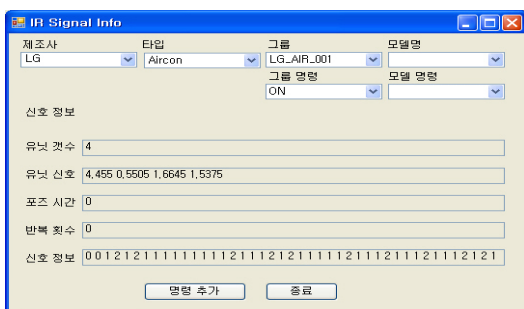


그림 6. IR신호 정보 확인  
Fig. 6 IR Signal information screen

명령 코드를 저장하기 위해 신호 입력기를 이용하여 우리가 원하는 형태의 신호를 그림 5와 같이 입력 받았

다. 입력 받는 항목은 제조사와 디바이스 타입, 그룹명과 그룹별 명령 혹은 모델명과 모델의 고유 명령을 입력받고 실제 명령 신호의 정보를 입력받는다. 이에 대한 간단한 구조는 표 1과 같다.

표 1. 명령코드 구조  
Table. 1 Command set structure

제조사	SAMSUNG, LG, DAEWOO ...
디바이스타입	Air Conditioner, Audio, TV ...
그룹	AA, BB, CC, DD, EE ...
모델명	AF R15, LN55A950D ...
모델명명	Power On, Power Off, Channel Up, Channel Down ...
신호 개수	250개 이하
신호재생시간	유닛신호의 정보를 바탕으로 재생한다.
출력될 신호	실제 신호가 저장될 공간
반복 횟수	반복신호를 요구할 경우에 대한 처리를 위해 저장

그룹 명령과 모델 명령을 구분하여 해당 기기에만 있는 고유 명령은 모델 명령에 추가하고 동일한 명령신호를 공유하는 제조사의 제품은 그룹 명령에 추가하여 저장 공간을 줄일 수 있도록 설계하였다.

그림 5에서 Length에는 4.455, 0.5505, 1.6645 등과 같이 실제 신호가 출력되는 시간의 길이를 저장해 놓았고, SigData에서 Length에 저장해 놓았던 시간 길이를 이용하여 명령 신호를 구성하는 것을 보여주고 있다. 이러한 방식으로 DB에 명령코드를 누적시킬 수 있고 그림 6과 같이 원하는 명령코드를 검색하거나 찾아볼 수 있다.

### 4.4 명령코드 학습 방법

사용자의 명령 요청이 있을 후 최단 시간 내에 출력을 완료하기 위해 임베디드 기기의 특성상 한정된 메모리 공간에서 문제를 해결해야 한다.

IR출력까지 최단시간 내에 최대한 빠른 출력을 위해 ZB Bridge 내에 버퍼를 마련하여 자주 사용하는 명령과 최근 사용한 명령을 저장해 두고 새로운 명령이 들어올 경우 해당 명령만 업데이트 하는 방법을 사용하기로 한다.

먼저 사용자가 원하는 명령을 홈서버가 수신하면 ZB Bridge에게 이 명령의 헤더를 전송한다. 헤더에는 명령을 수신 받을 리모컨의 ID와 모델명, 명령(TV ON/OFF, 온도 UP/DOWN 등)만 포함되며 그림 7에서

알 수 있듯이 매우 짧은 길이의 프로토콜 구조를 가지고 있다. ZB Bridge는 수신한 헤더를 확인하여 가지고 있는 명령인지 판단한다. 만약 가지고 있는 명령이면 자신의 메모리 DB에서 Load하여 IR신호로 바로 출력한다. 명령을 가지고 있지 않은 경우 홈서버에게 명령을 요청하고 홈서버는 자신이 가지고 있는 데이터베이스에서 명령을 검색하여 ZB Bridge에게 보내준다. 명령을 받은 ZB Bridge는 IR신호로 출력함과 동시에 자신의 DB에 새로 들어온 명령을 저장하게 된다. 이 때 저장 공간이 부족하다면 가장 오랫동안 사용되지 않았거나 사용 빈도수가 적은 명령을 검색하여 그 자리에 저장하고 DB를 업데이트 한다.

EndDevice. ID	DeviceModel	CMDType
---------------	-------------	---------

그림 7. 헤더 패킷 구조  
Fig. 7 Header packet structure

#### 4.5 학습한 명령코드 검색 방법

출력속도를 빠르게 하기 위해 ZB Device는 사용자가 어떤 명령을 내렸는지를 알 수 있는 헤더 패킷을 먼저 받아 자신의 메모리를 검색한다. 이를 위해 홈서버에서는 ZB Device에게 명령 코드를 전송하기 전에 먼저 명령이 담긴 헤더를 전송한다. ZB Device는 기존에 학습되었던 명령인지 판단하기 위해 자신의 메모리를 검색하고 메모리에 저장되어 있는 명령이면 코드를 검색하여 출력 배열에 삽입하여 전송하게 된다.

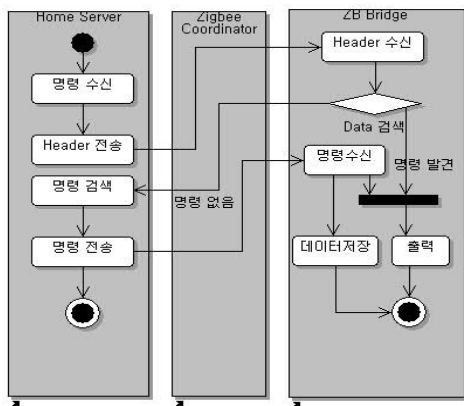


그림 8. 명령코드 학습방법  
Fig. 8 Learning method of command set

명령코드를 검색하기 위해 4.2절에서 설계한 데이터 타입 (DeviceCmd\_t)을 이용하여 다음과 같이 구현하였다.

헤더의 내용 중 장치의 모델명과 명령을 사용자가 보낸 명령과 비교하고 일치하면 출력 배열에 해당 데이터를 보내는 구조로 설계 되어 있다.

```

DeviceCmd_t *dc;

while(i<no_DeviceCmd)
{
    dc=&DeviceCmd[i];
    if((dc->deviceModel == rcvModel &&
        dc->CMDType==rcvCmdType)
        {
            for (int j=0;j<dc->mSigLen;j++)
            {
                wave[index++]=
                dc->uTime[dc->mSignal[j]];
            }
            ...
        }
        i++;
    }
}
    
```

#### 4.6 검색된 신호 출력 방법

위의 신호는 아래의 BinaryDecoding() 함수를 사용하여 wave[] array에 1/0의 Time Sequence로 저장된다. wave[0]와 wave[1]은 Carrier의 0/1의 시간을 각기 나타낸다. Wave[2] 이후는 신호의 sequence를 나타내는데, 짝수인 index는 1, 홀수인 index는 0을 각기 나타낸다.

```

void BinaryDecoding(uint16 *wave, byte *frame, uint16
*rx_index)
{
    DeviceCmd_t *dc;
    int i =0;
    wave[0] = 134;
    wave[1] = 134;
    while(i<no_DeviceCmd)
    {
        dc=&DeviceCmd[i];
        ...
        for (int j=0;j<dc->mSigLen;j++) {
            wave[index++] =
            dc->uTime[dc->mSignal[j]];
        }
        wave[index++]=100;
        RX_index = index;
    }
    i++;
}
    
```

IR 코드 생성은 ZigBee Chip으로 사용하는 CC2420이 내장하고 있는 Timer를 이용하여 구현된다. IR 코드를 생성하기 위해서는 Clock과 데이터 코드를 생성해야 한다. 따라서, 두개의 Timer가 이용되어야 한다. Clock을 만들어야 하는 Timer는 micro second 단위로 tick을 발생하는 clock (T3)를 이용하며 데이터 코드를 생성하는 Timer는 millisecond 단위로 tick을 발생하는 clock (T1)를 이용한다. 생성된 IR 신호는 IR Diode를 통하여 외부로 발송된다.

### V. 성능 분석

본 논문에서 제안한 방법을 통해 시스템을 설계하고 구현하였다. IR코드 재생기와 통합 리모컨 및 Zigbee 리모컨 방식 중 통합 리모컨은 새로운 명령을 추가할 수 없고, Zigbee 리모컨은 업체별로 각기 다른 프로토콜을 이용하므로 제안 방식과 비교 대상이 되지 않아 비교하지 않았다. 따라서 제안 방식과 비교할 수 있는 IR코드 재생 방식과 비교하여 성능을 분석하였다. 그림 9는 기존에 사용하던 IR코드 재생 방식을 이용하여 사용자가 명령을 요청하면 홈서버에서는 요청된 명령을 찾아 해당 명령의 데이터를 전송한다. 지그비 통신을 사용할 때 안정적으로 1회에 전송할 수 있는 크기를 150byte로 정하고 3회에 나누어 전송하였다.

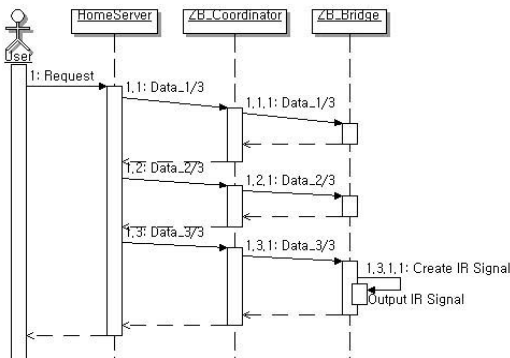


그림 9. 기존 데이터 전송 방식  
Fig. 9 Existing data transfer method

본 논문에서 제안하는 방식인 그림 10의 다이어그램과 같이 사용자가 명령을 요청하면 홈서버에서는 해당

명령의 헤더정보 (리모컨ID, 모델명, 명령)만을 먼저 전송하고 ZB\_Bridge에서는 해당 명령이 있는지 여부를 판단하여 IR을 출력하거나 다시 명령을 요청하게 된다.

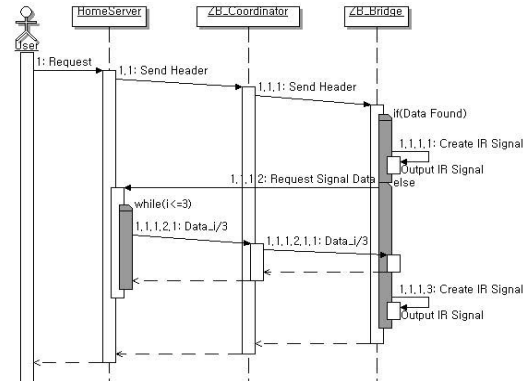


그림 10. 제안하는 전송 방식  
Fig. 10 The proposed transfer method

그림 11~13는 기존 방법과 제안하는 방법을 사용할 때의 전송 시간과 누적 시간, 소모 전력량에 대한 시뮬레이션 결과를 보여주고 있다. Zigbee 통신을 하기 위해 사용된 비트 당 전송시간은 4[us]이다[6][7].

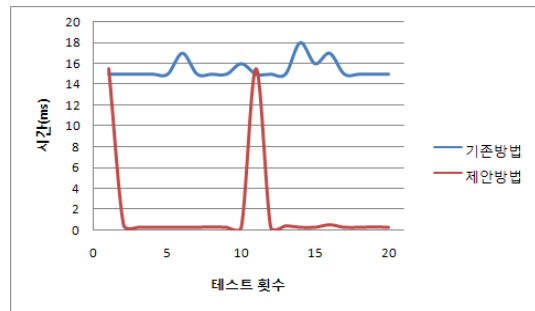


그림 11. 전송 시간 비교  
Fig. 11. Comparison of the transfer time

잡음 및 환경적 특성으로 스펙의 전송시간과 약간의 차이가 있음을 확인했고 우리는 실험을 통해 그림 11과 같은 결과를 얻을 수 있었다. 총 20번을 테스트 하였고 기존의 방법으로 할 경우 중간 중간 데이터를 수신하지 못하여 데이터를 다시 요청하고 받는 과정을 통해 약간의 시간적 차이가 있음을 확인할 수 있다. 제안하는 방법

의 경우 처음과 10회차에 약 15[ms] 정도의 시간이 소요 되었는데 명령 코드가 리모콘에 없어 학습하는 과정을 거치면서 그래프와 같은 결과가 발생하였다. 이 때의 소모 전류량은 그림 13의 제안방법 1과 같이 기존 방법에 비해 전류를 조금 더 소모한 것으로 나타났다. 그 외 학습한 데이터를 출력할 때에는 약 256[us] ~ 300[us] 정도의 시간이 걸렸음을 알 수 있다. 이는 기존 방법 대비 58.6배의 빠른 속도를 제공하는 것이다.

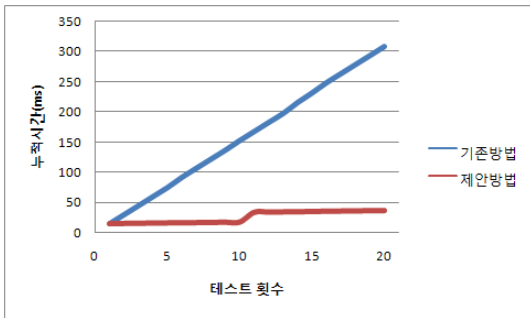


그림 12. 누적 시간 비교  
Fig. 12 Comparison of the cumulative time

누적 데이터의 경우 이러한 차이를 확연히 알 수 있는데 그림 11의 누적 시간 비교가 그것이다. 위에서 설명한 바와 같이 10회 차에 새로운 명령을 수신하는 과정으로 인해 약간의 누적시간이 증가했을 뿐 그 외에는 누적시간의 변동이 거의 없음을 알 수 있다.

데이터시트 스펙 상에 제시되어 있는 CC2420의 데이터 송신시 사용전류는 17.4[mA]이고 수신시의 사용전류는 19.7[mA]이다. 기타 거리 및 잡음 등에 의한 추가 전류 소모는 논외로 한다[7][8].

이 값을 통해 사용 전류를 계산하면 기존 방법은 데이터 요청과 수신을 포함한 1회 데이터 송수신시 사용되는 전류가 83.4048[nAH] 이고 제안하는 방법 중 리모콘에 명령코드가 존재하지 않을 경우 학습하는 동안 사용되는 전류는 기존방법보다 미세하게 큰 3.296[nAH]가 더 사용되지만 이미 학습한 명령 코드의 경우 명령의 헤더만 수신하면 되므로 1.7504[nAH]가 소모된다. 그림 13의 제안방법2의 차트에서 확인할 수 있다.

이는 기존 방법에 비해 약 47.6배 덜 소모하므로 그만큼의 배터리의 사용시간 연장이 가능함을 의미하여 매우 효율적이라 할 수 있다.

위에서 제시한 바와 같이 기존 방법 대비 전송시간은 58.6배 단축 되었고 배터리의 사용시간 또한 기존 방법 대비 47.6배 연장 되었음을 성능 분석을 통해 확인하였다.

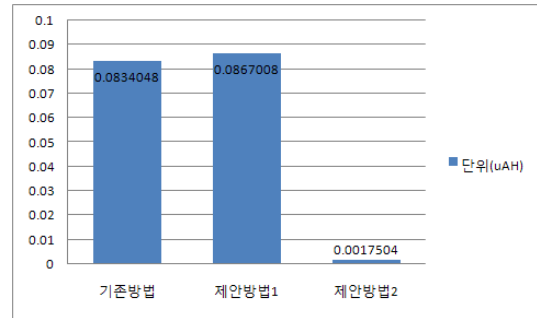


그림 13. 소모 전류량 비교  
Fig. 13 Comparison of current consumption

## VI. 결 론

본 논문에서는 원격지에서 각 가정의 기존 가전기기를 제어하고자 할 때 이전 방식인 데이터를 서버에 저장하거나 리모콘에 모두 저장하는 방식의 문제점들을 제거하고 이러한 문제점들을 해결하기 위해 필수적으로 요구되는 IR 신호 발생기의 명령 데이터를 효율적으로 관리하는 방법을 제안하였다.

공통으로 사용할 수 있는 표준화된 데이터 패킷과 통합DB를 설계하여 효율적인 명령 코드 저장방법을 제시하였고 명령 코드를 학습하고 저장할 수 있는 방법과 학습한 명령코드를 검색하는 방법, 검색된 명령코드의 IR 출력방법을 제안하였다.

제안한 방법을 사용할 경우 전력 소비를 최소화하여 장시간 배터리 교환 없이 유지할 수 있어 효율적이고 업데이트가 필요 없어 펌웨어 업데이트 비용이 전혀 들지 않으며 이로 인해 유지보수 비용이 감소함을 알 수 있다.

표준화된 저장 공간을 IR 재생기에 뒹으로써 빠른 검색 및 반응 속도를 확인할 수 있다.

추후 Zigbee 코디네이터를 홈서버에 포함시켜 좀 더 빠른 속도를 보장하고 무선 상에서의 패킷 손실로 인한 에러율을 낮추는 연구가 필요하다.



### 참고문헌

- [1] 김종원, 에너지-IT 기술 현황 및 전망, ETRI, pp13-14, 2010
- [2] Il-Kyu Hwang, "Home Network Configuring Scheme for all electric Appliances Using Zigbee-based Integrated Remote controller", IEEE Transactions on Consumer Electronics, Vol.55, No.3, pp 1300-1307, Aug 2009
- [3] Wan-Ki Park, "ZigBee based Dynamic Control Scheme for Multiple Legacy IR Controllable Digital Consumer Devices", IEEE Transactions on Consumer Electronics, Vol. 53, No. 1, pp172-177, Feb 2007
- [4] Khusvinder Gill, Shuang-Hua Yang, Fang Yao, and Xin Lu, "A Zigbee-Based Home Automation System", IEEE Transactions on Consumer Electronics, Vol.55, No.2, pp 422-430, May 2009
- [5] Dae-Yeong Im, Hyum-Rok Cha, Young-Jae Ryoo and Yong-Jun Lee, "Development of Remote Controller for Home Appliance Total control", Proceedings of KIIS Fall Conference. Vol.19, No.2, pp 120-122, Dec 2009
- [6] Zigbee Alliance, www.zigbee.org
- [7] Chipcon 2.4GHz 802.15.4 / Zigbee-ready RF transceiver CC2420 Datasheet
- [8] Heungsik Eom, "Study of Efficient Energy Management for Ubiquitous Sensor Networks with Optimization of the RF power", IEEK, Vol. 44, No. 3, pp37-42, May 2007

### 저자소개



**이승민(Seung-min Lee)**

2009년 한국교육개발원  
정보통신공학(학사)  
2010년~현재 경원대학교  
스마트그린홈연구센터

※관심분야: 스마트그리드, 임베디드시스템, 전력품질



**손성용(Sung-yong Son)**

2000년 Univ. of Michigan 박사  
1992~1995년 LG 소프트웨어  
2000~2004년 포디홈네트  
2004~2005년 아이크로스테크놀로지

2006년~현재 경원대학교 정보통신공학과 조교수

※관심분야: 스마트그리드, 스마트홈