

OpenWRT 기반 유무선 공유기와 Zigbee 통신을 이용한 홈 네트워크 서비스 구축

권기수 · 이정희*

Implementation of Home Network Services Using OpenWRT-based Wireless Access Point and Zigbee Communications

Kisu Kwon · Kyoung-Hee Lee*

Department of Computer Engineering, Pai Chai University, Daejeon 35345, Korea

요 약

최근 홈 CCTV, 가전기기의 원격제어, 가정용 보안 및 재난 예방 서비스와 같이 편리한 기능을 제공하는 스마트 홈 네트워크 서비스의 보급이 점차 증가됨에 따라, 기업체에서도 이를 위한 홈 게이트웨이, 스마트 스피커 등 제품들을 앞다투어 출시하고 있다. 이러한 서비스들은 제조사 별로 독자적인 플랫폼 상에서 개발된 제품을 통해 제공되고 있으므로, 타사 제품과 통신 방식 혹은 구조가 서로 달라 호환성이 낮은 단점이 있다. 이러한 문제점을 개선하기 위하여 본 논문에서는 오픈 소스 플랫폼을 활용하여 시스템 간 호환성을 보장하는 새로운 서비스 시스템을 제안한다. 본 논문에서는 OpenWRT기반의 유무선 공유기와 Zigbee 통신을 이용하여 전자기기들이 서로 통신하여 사물을 제어할 수 있는 홈 네트워크 서비스를 구축하였다. 유무선 공유기를 홈 네트워크의 게이트웨이로 이용하고, 전자기기들을 컨트롤하는 디바이스로 아두이노를 이용하여 기본적인 IoT 플랫폼과 이를 기반으로 몇 가지 시나리오의 스마트 홈 네트워크 서비스를 오픈 플랫폼 상에서 구축하였다.

ABSTRACT

As smart home network services such as home CCTV, outdoor control of home appliances, home security and disaster prevention services become popular, there appear various affiliated products including smart home gateway and smart speaker. Since those services are generally developed on the vendors' individual hardware and software platforms, it is not much expected for them to interwork well among different architecture and communication methods. In this paper, we propose a new home network service system running on an open source platform to address such issues. We implemented a home network system using OpenWRT-based wireless router(or access point) and Zigbee communication technology. In the proposed system, a wireless router replaces a commercial home gateway and small control units implemented with Arduino control electronic devices and sensors in home. Several service scenarios are also implemented to verify the operability of the proposed system.

키워드 : 홈 네트워크, 사물 인터넷, 유무선 공유기, OpenWRT, Zigbee 통신, 아두이노

Key word : Home Network, Internet of Things, Wireless Router, OpenWRT, Zigbee Communication, Arduino

Received 17 November 2017, Revised 22 November 2017, Accepted 28 December 2017

* Corresponding Author Kyoung-Hee Lee (E-mail:leekhe@pcu.ac.kr Tel:+82-42-520-5341)
Department of Computer Engineering, Pai Chai University, Daejeon 35345, Korea

Open Access <http://doi.org/10.6109/jkiice.2018.22.2.375>

pISSN:2234-4772

© This is an Open Access article distributed under the terms of the Creative Commons Attribution Non-Commercial License(<http://creativecommons.org/licenses/by-nc/3.0/>) which permits unrestricted non-commercial use, distribution, and reproduction in any medium, provided the original work is properly cited.
Copyright © The Korea Institute of Information and Communication Engineering.

I. 서론

스마트폰의 보급률에 비례하여 통신비 절약을 위해 무선 인터넷을 무료로 사용할 수 있는 유무선 공유기의 수요가 증가하였다. 국내 ISP(Internet Service Provider) 업체들은 인터넷 개통과 함께 유무선 공유기를 고객에게 보급하였고, 최상의 서비스를 제공하기 위해 가정과 같은 소규모의 네트워크에 전자기기들을 유선 혹은 무선으로 제어, 관리 및 모니터링 하는 기술을 개발하여 유료로 서비스를 상용화를 시작하였다.

유무선 공유기와 전자기기들이 서로 통신이 가능하면 인터넷으로 연결된 어디에서든지 원거리의 전자기기들을 다룰 수 있다. 공유기의 특성상 별도의 보안 설정이 되어있다는 전제하에 외부 네트워크에서 내부 네트워크로 접근이 가능하기 때문이다. 가정의 환경을 예로 들면, 집 밖 어디에서든지 집안의 에어컨, TV, 공기청정기 등을 편하게 다룰 수 있고, 집안의 실시간 상황과 애완동물을 CCTV로 확인할 수 있으며, 가스 누출 탐지기, 화재 경보기 등을 활용해 화재 예방을 할 수 있다. 특히, 스마트폰의 이동 통신망에서 위와 같은 기능을 구현하면 더 큰 편리함을 추구할 수 있다.

하지만 이러한 서비스들은 대부분 유료로 제공되고 있으며, 제조사별로 독자적인 플랫폼을 구축하여 제품을 개발하고 있으므로 타사 제품과 통신 방식 혹은 구조가 서로 달라 호환성이 낮은 단점이 있다. 예를 들어, A회사가 개발한 IoT 제품과 B회사가 개발한 IoT 제품은 겉으로는 둘 다 같은 기능을 할지라도, 독자적인 내부 동작 구조나 표현 방식을 이용하기 때문에 시스템 간의 호환성이 제공되기 어렵다.

시중에서 흔히 유통되는 유무선 공유기를 리눅스 기반 운영체제인 OpenWRT(Open Wireless Router)[1]로 적용하면 충분히 홈 네트워크 서비스의 게이트웨이로 활용이 가능하다. 오픈 소스 프로젝트를 지향하기 때문에 OpenWRT를 이용하는 많은 개발자들이 국내외 커뮤니티 등에서 활발하게 정보의 교환하고 피드백을 제공하고 있다. 특히 오픈 소스 프로젝트로 진행함에 따라 많은 유무선 공유기에 OpenWRT 적용이 가능해졌다. OpenWRT는 패키지 형태의 많은 소프트웨어들을 배포하고 있어 홈 네트워크를 구축하는데 필요한 소프트웨어들을 쉽게 얻을 수 있다.

본 논문에서는 OpenWRT를 기반으로 한 유무선 공

유기를 홈네트워크 게이트웨이로 사용하였고, 아두이노(Arduino)[2]에 온습도 센서를 장착하여 게이트웨이로 온도 값을 주기적으로 전송하도록 하였다. 또 다른 아두이노 보드에 USB 선풍기와 스위치 릴레이 모듈을 장착하여 온도 값을 받은 게이트웨이에서 사용자가 설정한 온도 값에 따라 스스로 전원을 켜고 끌 수 있는 시스템을 구축하였다. 유무선 공유기와 아두이노 보드 간 XBee 모듈을 이용하여 Zigbee 통신하도록 하였으며 사용자 정의 메시지를 구현하여 서로 의사소통 규칙을 정하였다.

II. 관련 연구

2.1. OpenWRT

OpenWRT는 2004년 가정용 유무선 공유기의 커스터마이징을 목적으로 개발된 리눅스 기반 오픈 소스 소프트웨어로, 다양한 임베디드 머신에 탑재가 가능하도록 발전해왔다[1]. ARM 아키텍처 타입의 CPU 뿐만 아니라 MIPS, x86 외 여러 하드웨어 칩셋과도 호환한다. OpenWRT SDK(Software Development Kit)를 무료로 제공받아 리눅스 운영체제에서 크로스 컴파일 하여 생성한 펌웨어를 유무선 공유기에 설치할 수 있다.

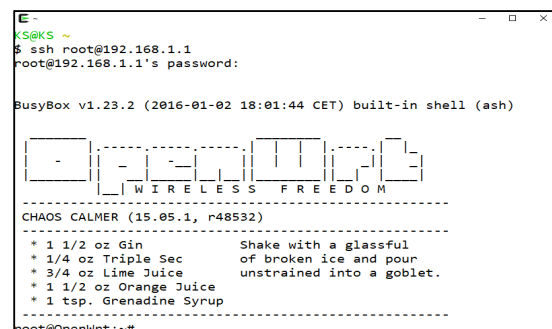


Fig. 1 OpenWRT connected with SSH

OpenWRT는 일반적인 유무선 공유기의 일반적인 기능 뿐만이 아닌 VPN(Virtual Private Network), NAS(Network Attached Storage) 등 다양한 서비스를 패키지 형태의 소프트웨어를 설치함으로써 이용할 수 있으며, 상황에 따라 소형 리눅스 운영체제 서버의 역할을 할 수 있다[1]. 그림 1은 원격 로그인 도구인

SSH(Secure Shell)를 이용하여 OpenWRT에 접속한 모습을 보여준다.

2.2. Zigbee

Zigbee는 IEEE 802.15.4 표준을 기반으로 한 저속도 무선 개인 통신망(Low Rate Wireless Personal Area Networks, LR-WPANs)이다[3-4]. 표 1과 같이 낮은 소비전력과 그에 비해 비교적 넓은 전송 거리를 제공하며, 구축할 때 저렴한 비용이 요구되므로 수 많은 센서들로 이루어진 RF(Radio Frequency) 네트워크 환경에 적합하다[5-6].

본 논문에서 사용된 Zigbee모듈은 IEEE 802.15.4 프로토콜에 네트워크 계층과 응용 계층이 추가된 XBee Series 2를 사용하였다[7]. XBee Series 2는 별도의 설정이 필요없는 XBee Series 1에 비해 통신 목적지 주소와 동작 모드 방식 등의 추가적인 설정을 필요로 한다. 또한, XBee Series 2는 P2P, 성형 구조 토폴로지만을 지원하는 Series 1에 망형 구조 토폴로지가 추가되어 XBee 모듈 간 1:1 통신이 가능하다.

Table. 1 Zigbee specifications

| | |
|---------------------|--------------------------------------|
| Standard | IEEE 802.15.4 |
| Data Rate | 250Kb/s |
| Range | 10-100m |
| Operating Frequency | 2.4GHz |
| Power Consumption | About 20% power consumption of Wi-Fi |

2.3. 아두이노

아두이노(Arduino)는 비교적 사용하기 쉬운 하드웨어와 소프트웨어를 기반으로 하는 오픈 소스 플랫폼 방식의 마이크로 컨트롤러 시스템이다[2]. 비전문가 수준에서도 충분히 아두이노 보드에 프로그램을 입력할 수 있고 ‘шил드(shield)’형태의 모듈을 장착하여 프로그래밍한 방법대로 모듈을 조작 및 제어할 수 있다[8]. 오픈 소스 플랫폼으로 아두이노 보드의 회로가 공개되어 있기 때문에 많은 하드웨어 제조사에서 아두이노 보드를 제조하여 저렴한 가격에 판매한다. 또한 아두이노와 호환되는 모듈은 대다수가 실드 형태를 하고 있기 때문에 용도와 목적에 맞는 모듈만을 사용함으로써 확장성과 연결성이 뛰어나다.

III. 설 계

3.1. 제안 시스템의 구성

그림 2는 본 논문에서 제안하는 홈 네트워크 시스템을 논리적인 구조를 보여준다. IoT 게이트웨이(IoT Gateway)와 디바이스 컨트롤러(Device Controller)가 서로 무선 통신으로 정보를 교환하며, IoT 게이트웨이가 디바이스 컨트롤러에게 제어 프로토콜을 이용하여 상황에 맞는 메시지로 제어 및 관리 할 수 있도록 하였다. 제어 프로토콜의 메시지들은 3.2절에서 자세히 다루도록 한다.

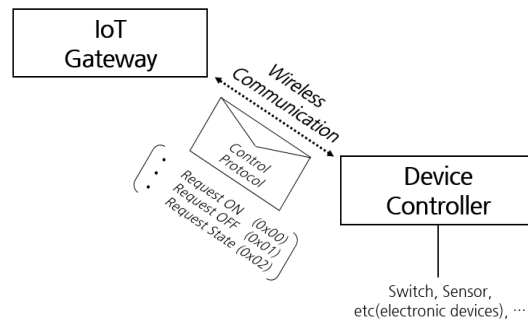


Fig. 2 Logical architecture of the proposed system

- **IoT Gateway** : 모듈(스위치, 센서,...)들의 상태 정보를 요청하거나 동작 제어를 하기 위한 명령을 디바이스 컨트롤러에게 전달한다.
- **Device Controller** : IoT 게이트웨이의 요청을 받아 각 모듈을 상태 정보를 얻어오거나 동작 제어를 하는 역할을 수행한다.

3.2. 제어 프로토콜 설계

각 모듈을 보다 효율적으로 제어하기 위하여 표 2와 같이 사용자 정의 메시지를 지정하였다. 표 2에서 언급된 메시지들은 그림 2에서 제어 프로토콜로 표현한 내용과 동일하다.

Table. 2 Control protocol messages

| Name | Hex No. | Content |
|---------------|---------|---------------------------------------|
| Request ON | 0x00 | Request to turn a Module On |
| Request OFF | 0x01 | Request to turn a Module Off |
| Request STATE | 0x02 | Request the Current State of a Module |

표 2의 *Request ON* 메시지는 특정 모듈의 전원을 켜도록 요청하는데 이용되며, *Request OFF*는 특정 모듈의 전원을 끄도록 요청하는 메시지이다. 마지막으로 *Request STATE* 메시지는 특정 모듈의 현재 상태를 요청할 때 전송된다. 표 2의 'HEX No.'는 각 메시지 내에 포함되는 16진수 코드로서, 메시지를 구분하기 위한 명령 타입 코드로 사용된다.

현재 구현된 시스템 상에서, *Request ON* 메시지와 *Request OFF* 메시지는 스위치 릴레이 모듈에게만 적용되며, *Request STATE* 메시지는 스위치 릴레이 모듈과 온습도 센서 둘 다 적용되는 메시지이다. *Request STATE*에 대한 응답으로 스위치 릴레이의 경우 현재 'On' 혹은 'Off' 상태를 반환하고, 온습도 센서의 경우 현재 온도와 현재 습도 값을 반환한다.

3.3. 제안 시스템 동작 절차

그림 3은 본 논문에서 제안하는 시스템이 동작하는 시나리오 중 하나이며, 그의 절차를 순서도로 나타낸 모습이다. 이 그림에서 보여주고 있는 예는 선풍기의 전원을 제어하는 게이트웨이와 디바이스 컨트롤러, 센서, 선풍기가 연동하는 전체적인 과정을 나타내고 있다.

프로세스가 시작된 후 (1)의 과정과 같이 스위치 릴레이 모듈의 상태(Relay State), 온도 센서 값(Sensor Value), 자동으로 선풍기를 켜고 끄도록 하는 오토 팬 컨트롤(Auto Fan Control), 사용자가 설정한 온도 값(Degree)들을 읽는다. 프로세스 논리 동작 반복 여부(Keep Going)의 초기 값은 true이며, 사용자 임의로 프로세스 종료 요청 시 false로 적용되어 (4)의 과정에서 프로세스가 종료된다. 릴레이 모듈의 상태정보와 온도 센서 값은 *State* 메시지를 통해 얻도록 하고, 오토 팬 컨트롤과 설정 온도 값의 정보는 LuCI(Lua Configuration Interface)에서 사용자가 설정한 값을 참조하여 얻는다. (2)에서 오토 팬 컨트롤의 값이 false일 때, 설정 온도 값에 따른 선풍기 동작 과정이 불필요하기 때문에 최초 분기로 되돌아온다.

(3)은 온도 센서 값이 설정 온도 값 보다 작고 스위치 릴레이 모듈의 상태가 true일 때 *OFF* 메시지를 스위치 릴레이 모듈에게 전송하여 선풍기를 끄도록 하고 그 외의 경우 *ON* 메시지로 선풍기를 켜도록 한다. 위와 같은 프로세스 동작 중 사용자의 프로세스 종료 명령이 있으면 (4)의 Keep Going이 false로 되어 프로세스를 종료

하지만 그렇지 않은 경우 최초 분기로 돌아와 프로세스의 처음 과정부터 시작하도록 한다.

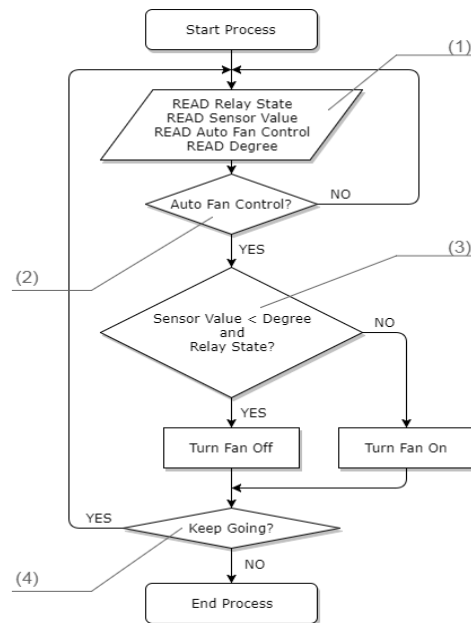


Fig. 3 Operation procedure of the USB-powered fan base on temperature

IV. 구현

4.1. 개발 환경

본 논문에서 구현한 IoT 게이트웨이의 홈 네트워크 서버 프로그램은 스크립트 언어 중 하나인 Lua 언어를 기반으로 작성하였다. 2.1절에서 언급한 OpenWRT는 Lua 언어를 기반으로 구현 되었는데, 그 이유는 가볍고 빠른 특징으로 라우터와 같은 임베디드 디바이스에 적합하기 때문이다. 디바이스 컨트롤러의 경우, XBee와 스위치, 온습도 센서를 다루기 위해 아두이노 라이브러리를 이용하였다. 아두이노에서 XBee를 사용하기 위한 라이브러리는 'xbee-arduino[9]'를 사용하였다. 온습도 센서의 경우, 본 논문에서 사용한 온습도 센서 모듈인 'DHT-sensor-library[10]'을 위해 제공되는 튜토리얼 라이브러리를 활용하였다.

4.2. 홈 네트워크 서버 구축

본 논문에서 사용한 유무선 공유기는 Buffalo사의 WZR-HP-AG300H를 이용하였다. 32MiB(메가 이진 바이트)의 여유로운 저장 공간과 USB 2.0 포트, OpenWRT 펌웨어를 지원하기 때문에 홈 네트워크 서버를 구축하기에 적합하다[11]. OpenWRT 운영체제는 내부적으로 대부분 가볍고 강력한 Lua 스크립트로 동작하며 유무선 공유기를 사용자 수준에서 쉽게 다루기 위한 웹 기반 사용자 인터페이스인 LuCI(Lua Configuration Interface)를 갖추고 있다. 홈 네트워크 서버는 Lua 스크립트로 구현하였으며 XBee 모듈을 쉽게 제어 및 관리할 수 있도록 LuCI 내에 사용자 인터페이스를 제공하였다. USB 2.0 포트에 XBee 모듈을 장착하여 타 모듈과 Zigbee 무선 통신이 가능하도록 하였다.

그림 4는 그림 2의 구조를 바탕으로 홈 네트워크를 구축하기 위해 필요한 장비들의 실질적인 구성도를 나타낸 모습이다.

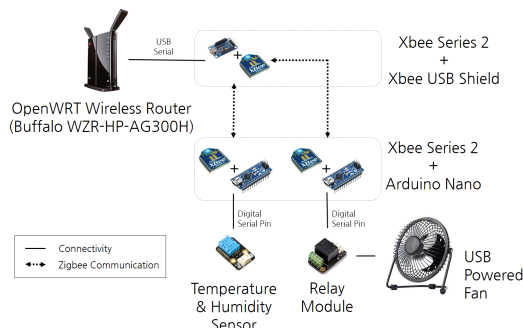


Fig. 4 Home Network Configuration

그림 4의 OpenWRT 무선 라우터와 아두이노 Nano는 앞서 보인 그림 2의 IoT 게이트웨이와 디바이스 컨트롤러에 해당된다. 무선 라우터와 아두이노 간에는 XBee Series 2 모듈을 이용하여 Zigbee 통신으로 메시지를 전달한다. 메시지를 전달받은 아두이노는 메시지 종류에 맞게 온습도 센서 혹은 릴레이 모듈(Relay Module)의 전원을 제어하거나 또는 상태 정보를 보내도록 요청한다.

그림 5와 같이 LuCI에서 모듈들의 실시간 상태정보를 모니터링 할 수 있으며 쉽게 제어가 가능하다.

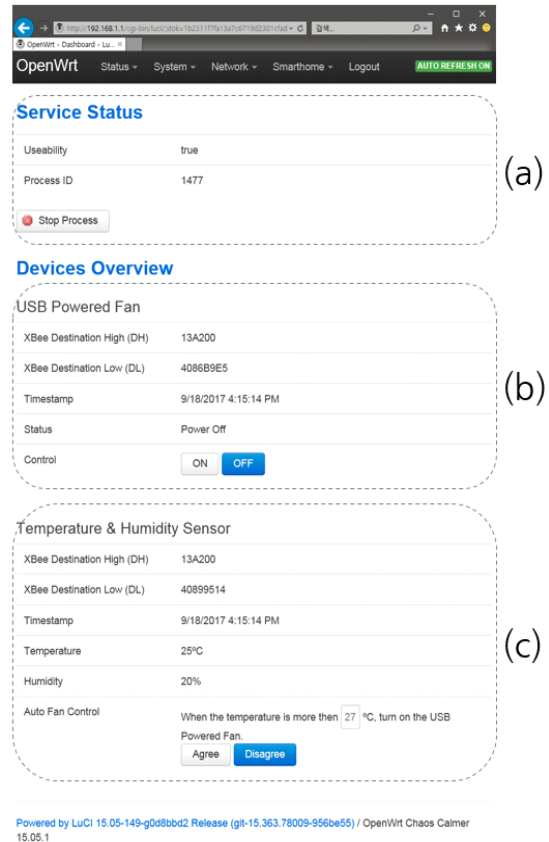


Fig. 5 User interface of the proposed system

그림 5의 (a)는 사용자가 IoT 게이트웨이 프로세스를 시작 혹은 정지하거나 프로세스의 ID를 확인할 수 있다. (b)에서는 XBee 모듈의 주소 정보들(DH, DL)과 최근 메시지를 수신한 시간을 나타내는 타임스탬프(Timestamp), 전원 상태(Status)를 확인할 수 있고, 제어(Control) 항목에서 수동으로 전원 상태를 조작할 수 있다. (c)는 (b)와 마찬가지로 온습도 센서의 주소 정보들과 타임스탬프를 확인할 수 있을 뿐 아니라 온도 값과 습도 값을 확인할 수 있다. ‘오토 팬 컨트롤(Auto Fan Control)’ 항목에서 온도 값을 입력한 후 ‘Agree’버튼을 클릭하면, 현재 온도가 입력한 온도 값보다 높을 경우 오토 팬 컨트롤이 작동한다.

그림 6은 그림 4를 토대로 실제 하드웨어를 구현한 모습이다.

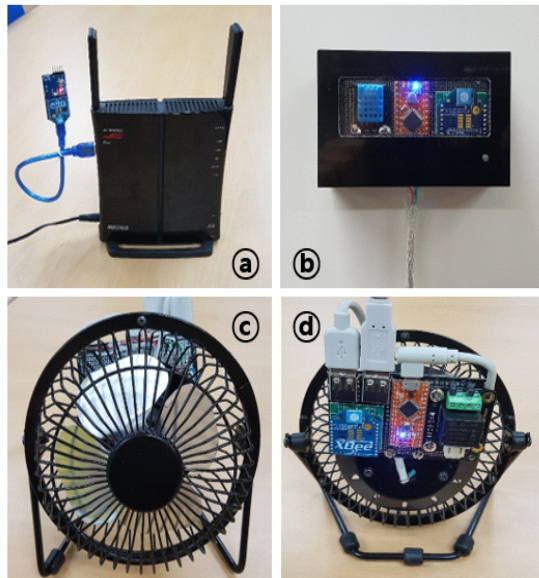


Fig. 6 System Implementation

그림 6의 ㉑는 OpenWRT 펌웨어를 적용한 ‘Buffalo WZR-HP-AG300H’ 라우터의 USB 2.0 포트에 XBee 모듈을 장착한 모습이다. ㉒는 온도 측정 단말을 구현한 모습이며 아두이노 보드에 XBee 모듈과 온습도 센서(DHT11)를 장착하였다. ㉓와 ㉔는 USB 전원 선풍기로, 아두이노 보드에 XBee 모듈과 스위치 릴레이 모듈을 장착하였다. 온도 측정 단말과 USB 전원 선풍기는 DC 3.3V의 낮은 전력을 필요로 하며, 상시 전원이 공급되어야 정상적인 서비스가 가능하다.

그림 7은 스마트폰에서 DDNS(Dynamic Domain Name System)가 설정된 라우터로 이동 통신망을 통하여 LuCI에 접속한 모습이다. 그림 5와 같이 단말들의 정보를 제어, 관리 및 모니터링을 할 수 있다. 그림 7에서 보이는 것과 같이, 스마트폰에서 이동 통신망을 통해 가정의 홈 네트워크를 제어할 수 있으므로 더욱 큰 편리함을 얻을 수 있다.

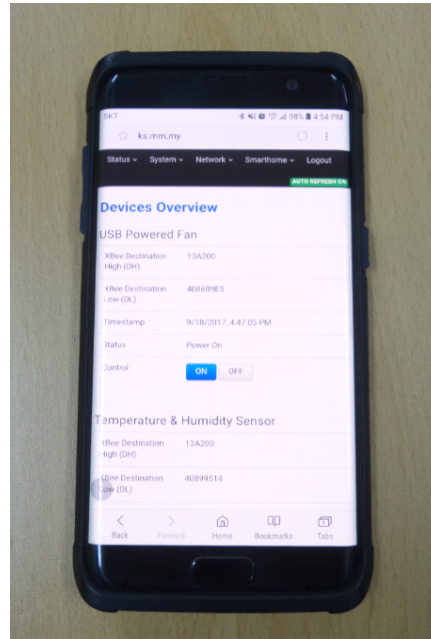


Fig. 7 User interface connected with smartphone

V. 결 론

본 논문에서 구현한 홈 네트워크 서비스는 오픈 플랫폼인 OpenWRT를 이용하여 상용 서비스에서나 제공되는 수준의 기능을 개발자들에게 익숙한 환경으로 제공한다. 또한, 유무선 공유기와 마이크로 컨트롤러 시스템 등 시중에서 쉽게 구할 수 있는 저렴한 비용의 하드웨어를 이용하여 구현할 수 있다. 오픈 소스 플랫폼을 사용하였기 때문에, Zigbee 통신 모듈이 장착된 아두이노 보드와 제어 프로토콜만 서로 맞춰준다면 다양한 기기들과 충분히 호환이 가능한 장점을 보여줄 수 있다.

특히 공유기와 전자기기들이 서로 Zigbee 통신을 하고 공유기를 제외한 단말들이 낮은 전력을 요구하기 때문에, 전력 소모에 부담이 없다. 그리고 IoT 게이트웨이 역할을 하는 공유기의 무선 인터페이스에 종속되지 않는 장점과 웹 기반 사용자 인터페이스를 통해 제어, 관리 및 모니터링 기능을 제공하는 장점이 있다. 웹 인터페이스를 사용하므로, PC와 스마트폰을 비롯하여 대다수 운영체제에서 호환이 가능하다.

REFERENCES

- [1] OpenWrt - Wireless Freedom, OpenWrt.org [Internet]. Available: <https://openwrt.org>.
- [2] Arduino forum [Internet]. Available: <https://forum.arduino.cc>
- [3] Zigbee Alliance Homepage [Internet]. Available: <https://www.zigbee.org>.
- [4] IEEE Std. 802.15.4, "Low-Rate Wireless Personal Area Networks (LR-WPANs)," *Institute of Electrical and Electronics Engineers*, 2011.
- [5] H. Kim, C. H. Kim, K. Y. Jin and J. M. Chung, "A Study on Efficient and Secure Zigbee Networking," *Proceedings of Symposium of the Korean Institute of communications and Information Sciences*, pp. 380-383, Nov. 2007.
- [6] J. Huh, S. Je and K. Seo, "Design and Simulation of Foundation Technology for Zigbee-based Smart Grid Home Network System using OPNET Simulation," *Asia-pacific Journal of Multimedia Services Convergent with Art, Humanities, and Sociology*, vol. 5, no. 4, pp. 81-89, Aug. 2015, <http://dx.doi.org/10.14257/AJMAHS.2015.08.65>.
- [7] Digi XBee Ecosystem - Everything you need to explore and create wireless connectivity - Digi International [Internet]. Available: <https://www.digi.com>.
- [8] S. Ahn, "Design and Implementation of Smart LED Bicycle Helmet using Arduino," *Journal of the Korea Institute of Information and Communication Engineering*, vol. 20, no. 6, pp. 1148-1153, June 2016.
- [9] xbee-arduino - Arduino library for communicating with XBee radios in API mode [Internet]. Available: <https://github.com/andrewrapp/xbee-arduino>.
- [10] DTH-sensor-library - Arduino library for DHT11, DHT22 and etc Temp & Humidity Sensors [Internet]. Available: <https://www.zigbee.org>.
- [11] OpenWrt Wiki. The Specification of "Buffalo WZR-HP-AG300H" hardware. [Internet]. Available: <https://wiki.openwrt.org/toh/buffalo/wzr-hp-ag300h>.



권기수(Kisu Kwon)

2016년 ~ 현재: 배재대학교 컴퓨터공학과 석사과정

2015년 ~ 현재: ㈜유비테크 소프트웨어 개발팀 엔지니어

※관심분야: Application Software, 무선 통신, 임베디드 시스템



이경희(Kyoung-Hee Lee)

2006년: 한국과학기술원(KAIST) 공학박사

2006년 ~ 2012년: 한국전자통신연구원(ETRI) 선임연구원

2013년 ~ 현재: 배재대학교 컴퓨터공학과 교수

※관심분야: 모바일/무선 인터넷, 5G 통신, 사물인터넷, 클라우드 컴퓨팅