

오픈소스 기반의 IoT 통합 컨트롤러 설계

이세훈*, 김주봉⁰, 고희창**

⁰*인하공업전문대학 컴퓨터시스템과

**인하대학교 산학협력단

e-mail:seihoon@inhac.ac.kr, violet0106.jb@gmail.com, heechang.koh@inha.ac.kr

Design of IoT Integrated Controller based on Open Source

Se-Hoon Lee*, Ju-Bong Kim⁰, Hee-Chang Koh**

⁰*Dept. of Computer Systems & Engineering, Inha Technical College

**Research and Business Foundation, Inha University

● 요약 ●

본 논문에서는 오픈 소스 하드웨어와 소프트웨어만을 이용해 IoT 컨트롤러를 구현한다. 컨트롤러는 무선 AP와 블루투스(Bluetooth), 지그비(ZigBee)등의 무선 통신을 이용하여 디바이스를 제어 및 처리하고 IPTV 셋탑 기능 등을 수행하는 컨트롤러로, 로컬 환경 전체에 한 대의 디바이스를 이용하여 센싱 및 정보 처리를 수행할 수 있는 시스템을 설계한다.

키워드: 사물인터넷(Internet of Things), 오픈소스 플랫폼(Open Source platform), 디바이스 제어기(Device Controller)

I. 서론

[1]에 따르면 셋톱박스의 대기전력이 대형 TV의 대기전력보다 260배까지 차이가 난다는 연구 결과가 있다. 42인치 대형 LED TV가 있는 가정집에서 전력 측정 장비로 실험 결과 TV는 대기전력 0.065W, 셋톱박스는 17.39W의 대기전력을 소모한다. 이는 한 가정의 전체 전기요금의 6%에 해당하는 큰 수치이다. 점차 다가올 IoT시대에 이는 큰 문제점으로 적용될 수 있다. [2]의 결과를 보면 IPTV의 셋톱박스 보급 예상 추이가 지속적으로 증가하는 것을 확인할 수 있다. 일반 사용자는 기본적으로 무선 AP공유기 1대, IPTV셋톱박스 1대, 인터넷 셋톱박스 1대를 평균적으로 사용한다고 본다면 이는 사용자의 편의성 및 환경적으로도 문제가 생길 수 있는 요자가 있다. 따라서 본 논문에서는 오픈 소스를 이용하여 AP 및 무선 통신기술을 이용하여 센싱값을 받아 처리해, 사용자에게 IoT 환경을 제공하여 줄 수 있는 오픈소스 기반의 IoT 통합 컨트롤러를 설계 한다.

II. 관련 연구

1. 통합형 IoT플랫폼 동향

IoT산업이 활발히 진행되며, 대기업 또한 발 빠르게 진행하고 있다. 대표적으로 LG CNS의 통합형 홈 IoT서비스와 삼성전자의 IoT플랫폼을 스마트TV에 탑재한 제품이 있다. [3]에 따르면 LG CNS의 통합형 홈 IOT서비스는 OSGi(Open Service Gateway

및, 제트 웨이브, 블루투스등을 이용하였다. 이로 인해 스마트 TV 및 무선 홈 IoT플랫폼이 셋톱박스 한가지로 해결이 가능하다.

[4]에 따르면 삼성전자의 스마트TV IoT플랫폼은 ‘스마트싱스’의 IoT플랫폼과 스마트TV를 결합한 제품이다. 스마트TV한대만으로 별도 외장형 IoT허브 없이 디바이스들과 연동되어 제어할 수 있는 이점이 있다.

2. OSGi

OSGi(Open Services Gateway Initiative)는 JAVA에서 동적 컴포넌트 시스템을 정의한 스펙이다. [5]에 따르면 컴포넌트 모듈을 만들고 이들을 추가하면 동적으로 반영하게 되고 상호 모듈끼리 통신하여 대화할 수 있도록 만들어 주는 것이다. 자바 인터페이스와 실제 구현 개체를 서비스로 만들고 서비스를 제공하기 위한 기능적 배포단위로 번들을 만든다. 번들의 생명주기 관리는 번들 컨테츠가 수행한다. OSGi 프레임워크는 번들을 관리하는 번들 컨테츠를 가지고 있어 서비스를 등록하여 번들끼리 상호 대화할 수 있도록 해준다. 또한 OSGi는 홈 네트워크 게이트 웨이기도 하다. [6]에 따르면 도이치 텔레콤, 프랑스텔레콤, NTT 등의 통신업체들, 삼성전자, Siemens, Toshiba 등의 가전업체들, HP, National Semiconductor, ProSyst, Sun Microsystems 등 컴퓨터 관련업체들, 인터넷 서비스 제공업체(ISP) 등 40여 개 이상의 업체들이 참여하고 있다. 따라서 본 논문에서는 홈 게이트웨이의 표준 중 OSGi를 지원하고자 한다.

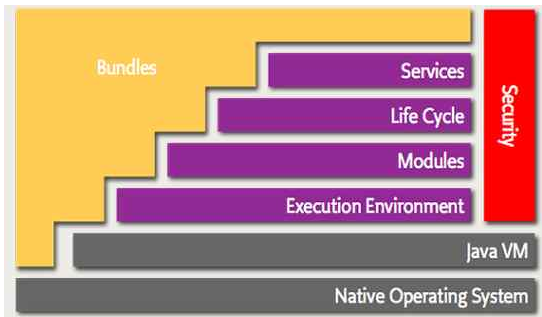


Fig 3. OSGi Layer

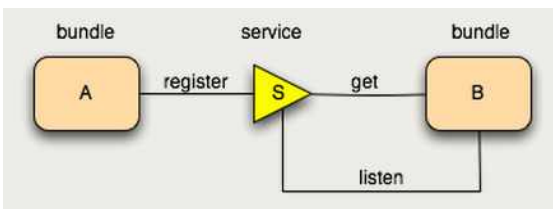


Fig 4. OSGi framework principles interactive

[7]에 따르면 그림5과 같이 OSGi서비스를 기반으로 하여 홈네트워크를 구성하였다. OSGi프레임워크를 중심으로 각 디바이스 단위별로 변들 및 서비스가 등록이 되며 이 서비스를 통하여 클라이언트가 접근해 디바이스 제어를 수행할 수 있다. [7]에서 제시하는 OSGi의 서비스 구조를 기획하였다.

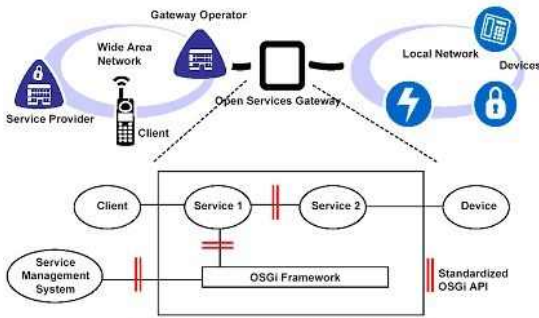


Fig 5. OSGi Home Network

3. 오픈 IoT 플랫폼

국내외적으로 IoT 플랫폼 표준화 및 개방화에 진전이 가속되고 있다. [8]에서는 세미나를 개최하고 IoT 플랫폼에 대한 다양한 논의가 진행되었다. 대표적인 국제적 표준으로 oneM2M은 글로벌 사물인터넷 서비스 플랫폼 표준 개발을 위해서 ETSI, TTA, ATIS 등 8개 표준화 단체가 공동 설립해 진행하고 있다. AllSeen Alliance는 Qualcomm, 시스코, MS, LG 등과 Linux 재단에서 표준화 진행하고 있다. OIC는 인텔, 삼성 등의 주도로 퀄컴 주도의 AllSeen에 대응하고

IoT 기기 연결성 확보를 노력하고 있으며, IoTivity를 통한 소스 배포하고 있다.

국내에서는 정부의 지원으로 오픈 IoT 플랫폼 개발과 보급이 진행되고 있다[8]. Modacom에서는 Omnipass라는 oneM2M 기반의 IoT 플랫폼과 SmartBrain 이라는 게이트웨이도 발표하였다. Cresprit에서는 ALOOH라는 모듈화된 IoT 플랫폼을 발표하였으며, 게이트웨이에서 자바스크립트를 이용해 응용할 수 있게 해 현장에 적용하기도 하였다. SK텔레콤에서는 Thing Plug라는 플랫폼을 발표하고, 오픈 서비스인 Sandbox 환경에 접속해 자유롭게 개발 및 테스트를 가능하도록 지원하고 있다.

III. IoT 통합 시스템 설계

1. 시스템 개요

시스템의 전체 구조도는 그림 6와 같다. Linux가 구동되는 오픈 소스 보드를 기반으로 AP서비스를 구동시키며 내부 버스를 통하여 각 데이터 태그별로 목적지에 데이터를 송수신한다. 사용자 가시적 출력은 오픈소스의 임베디드 서버를 사용하였으며 GCM, DB컨넥터, OSGi 번들로 이루어진 디바이스 컨트롤러 모듈로 구성한다. 각 모듈들은 서로 상호작용하며 동작하기 때문에 하나의 모듈이 런 타임(Run Time)에러 발생 시 모듈 매니저는 리셋과 동시에 각 모듈들을 실행시키며 정상가동을 체크한다.

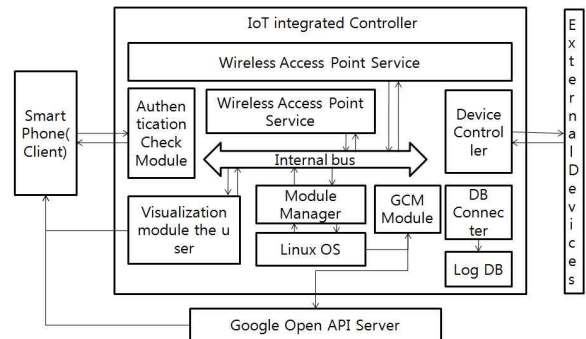


Fig 6. System Architecture

시스템의 기능과 흐름을 나타내는 시퀀스 다이어그램이 그림 6과 같다. 클라이언트는 서버로 접속을 시도하며 서버는 해당 클라이언트가 인증이 되었는지 확인이 된 후에 인증이 되어있으면 연결이 되었다는 푸싱 메시지를 보내주게 된다. 만약 허가되지 않은 사용자 접근 시에는 차단상태에 들어가게 된다. 해당 서버는 지속적으로 자신의 상태를 확인하며 이상 상황 발생 시 즉시 클라이언트로 해당 메시지를 전송하여 준다. 이 이상메시지는 화재 및 [6]에 따르는 오픈소스 하드웨어 기반의 스마트 도어락의 침입감지를 의미한다. 컨트롤러는 서버로 디바이스 제어 명령을 내리게 되면 서버는 내부 버스를 이용하여 해당 디바이스의 상태를 감지한 후에 이상 없을 시에 블루투스 및 WiFi 통신 기반으로 디바이스를 제어한다. 해당 작업 완료 후 리턴 메시지를 전송하여 준다. 만약 디바이스 제어 불능 상태 시

에러 메시지를 해당 클라이언트로 GCM(Google Cloud Message)를 이용하여 전송하여 준다. 모든 기록은 내부 버스 및 DB 컨넥터를 이용하여 로그를 데이터베이스에 저장한다.

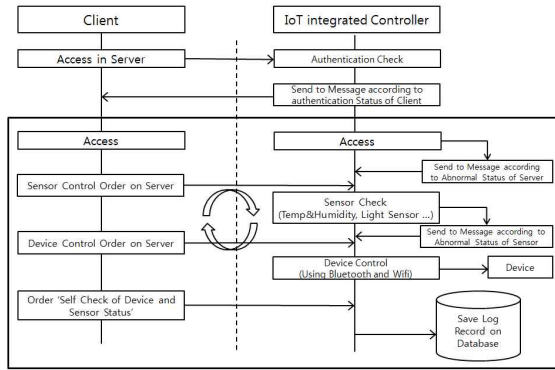


Fig 7. Sequence Diagram

2. Device Controller

다바이스 컨트롤러는 OSGi를 준수하여 구성한다. 또한 JAVA를 이용 OSGi 라이브러리를 이용한다면 손쉽게 구현할 수 있다. 본 논문에서는 보다 이해를 돕기 위하여 오픈소스 스마트 도어락을 예로들어 작성하였다. 해당 도어락의 잠금장치를 해제하는 빌더의 알고리즘은 아래의 표와 같다.

Algorithm 1 Device Control logic

```

Begin
    System.out.println("Devicie Service.id="+device,
        getProperty(IGPIOPinOutput.PIN_ID_PROP));
    IGPIOPinOutput pin = context.getService(
        reference);
    pin.setState(true);
    return pin;
End
    
```

다바이스 OSGi 라인에 해당 다바이스의 명령수행상태를 출력하게 되며 동시에 GPIO를 제어하여 개폐장치를 해제하게 된다. OSGi 커맨드 라인에 출력된 스트링 형식의 데이터를 리턴 받아 해당 다바이스의 명령수행상태를 로그에 저장하게 된다. IoT 통합 컨트롤러에서 해당 도어락의 잠금장치 해제 요청을 하는 알고리즘은 다음과 같다. 비동기 원격 서비스가 등록이 되어 있어야 하며 라이브러리의 지원으로 복잡한 스테딩 코드 없이 서비스 등록만으로 대기 상태에 돌입할 수 있다. 아래의 표는 비동기 원격 서비스를 등록한 간단한 알고리즘이다.

Algorithm 2 Asynchronous remote Services faced logic

```

Begin
    interface IGPIOPinOutputAsync{
        CompletableFuture <Boolean> getState
        Async ();
        CompletableFuture <Void> setState
        Async(boolean value);
    }
End
    
```

해당 알고리즘을 서비스 등록한 뒤에 클라이언트로 제어 명령 요청을 입력받게 되면 다음과 같이 지그비(ZigBee)제어 라이브러리를 이용, 해당 다바이스로 무선통신을 송신하게 된다.

Algorithm 3 ZigBee transmission logic

```

Begin
    CommPortIdentifier ports = Comm
    PortIdentifier.getPortIdentifier("COM4");
    SerialPort port = (SerialPoart) poarts
    .open("RS232C",1000);
    port.setSerialPortParams(9600, Seri
    alPort.DATABITS_8,SerialPoart.STOPBITS_1,
    SerialPort.FLOWCONTROL_NONE);
    bos = new BufferedWriter(new Output
    StreamWriter(port.getOutputStream()));
    bos.write(0x01);
    bos.finish();
End
    
```

3. Hardware

하드웨어의 구성은 OSHW(Open Source Hware Ware)인 라즈베리파이(Raspberry Pi)개발보드 상에 Wifi 및 블루투스 등의 무선통신 모듈을 부착하여 구성하였다. 구성한 모습은 아래의 그림 8과 같다. 이더넷 하드웨어 칩셋은 RTL8192CU 칩셋을 선택하였으며 블루투스의 하드웨어 칩셋은 BU-2086 H-20을 사용하였다.



Fig 8. IoT Integrated Controller

4. Software

소프트웨어의 스택은 다음과 같다. 통합 컨트롤러 ARM프로세서에 리눅스 기반의 OS가 존재하며 각 단위의 모듈을 관리하는 모듈 매니저가 구동된다. 각 모듈들은 내부 버스에 의해 서로의 데이터를 주고받을 수 있으며, GCM모듈을 통하여 사용자의 클라이언트 디바이스, 즉 스마트폰으로 푸싱 메시지를 줄 수 있다.

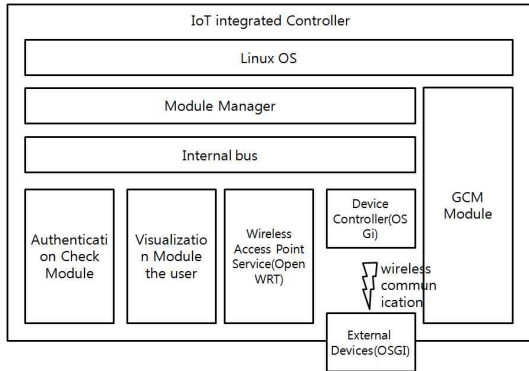


Fig 10. IoT Integrated Controller

IV. 결과

본 논문에서는 로컬에서 여러 기능을 통합한 IoT 컨트롤러를 설계하였다. 특히, 기능별 기기로 인한 전력 소모등의 문제를 해결하기 위해 통합 플랫폼으로 OSGi를 기반으로 해 확장성과 가용성을 극대화하였다. 셋탑박스의 대기전력은 현 가전제품 중 1순위에 손꼽힌다. 또한 [1]에 따르면 가전제품의 핵심 전략은 대기전력의 저전력화로 이어질 것이라고 예측하고 있다. IoT플랫폼의 추가 설치로 인해 늘어나는 셋탑박스의 추가 전력소모와 시각화, 사용자의 편의성 또한 떨어질 것이다. 이러한 셋탑박스가 모두 결합되어 하나의 셋탑박스만 존재한다면 환경적 및 사용자 측면에서 좋은 효과를 볼수 있을 것이다.

따라서 오픈소스 기반의 IoT통합 컨트롤러는 하나의 디바이스에 무선 AP 및 IoT플랫폼을 결합하여 센싱 데이터 출력 및 가전제품 제어, 무선 공유기의 기능을 모두 수행하여 3개의 셋탑박스가 아닌 단 하나의 셋탑박스를 이용해 소비전력 및 시각화를 높일 수 있을 것이다. 그러므로 해당 시스템으로 하여금 환경적 개선과 사용자의 편의성이 개선될 수 있을 것이라고 판단된다.

Reference

- [1] Jong-Gen Lee, "Electronics and IT market It was burned, mileage competiion", LGERI Report, pp.6, Nov 2014
- [2] Gwang Ok Choi, "Monthly Issue: Smart Complements of TV, Set-Top Box", National IT Industry Promotion Agency, pp 40, May 2013
- [3] About LG CNS, "LG CNS, Korea's first integrated home-based set-top boxes IoT solutions", March 2015, http://www.lgcns.co.kr/views/News/NewsDetail?SERIAL_NO=1534
- [4] Samsung Smart Things <https://www.smarthings.com/>
- [5] OSGI WebSite <http://www.osgi.org/Technology/WhatIsOSGi>
- [6] Se Hoon Lee, "Open Source Hardware based Conetext-Aware Smart Digital DoorLocks", Korea Society of Computer Information, July 2014
- [7] Chon-Gal Park, "Desin of OSGi based Intelligent Home Network Service Management Platform", KT Network Technology Research Institute, KNOM Review. Vol.9, June 2006
- [8] IoT 오픈플랫폼 기반 제품·서비스 개발 세미나, <http://www.kiot.or.kr/>, 2015