



Smart Home Hub

Team Bluethings

Team Partner

201332028 장승훈

201332001 강인성

사물인터넷 캡스톤디자인 프로젝트 최종보고서

성공회대학교 소프트웨어공학과 2017

프로젝트명	스마트 홈 허브
팀명 및 구성원	장승훈, 강인성
지도교수	유상신 교수님
프로젝트 기간	2017년 9월 - 2017년 12월

- 목 차 -

1. 차세대를 위한 Smart Home Hub

- 1.1 작품의 배경 및 개요
- 1.2 작품 목표
- 1.3 작품 주요 내용
- 1.4 조원 별 역할 분담

2. 이용 기술 소개

- 2.1 스마트 스위치, 스마트 콘센서
- 2.2 모바일 어플리케이션
- 2.3 라즈베리파이 모니터 어플리케이션
 - 2.3.1 Web UI
 - 2.3.2 Connector UI

3. Project 소개

- 3.1 스마트 스위치
- 3.2 스마트 콘센서
- 3.3 모바일 어플리케이션
- 3.4 라즈베리파이 모니터 어플리케이션
 - 3.4.1 Web UI
 - 3.4.2 Connector UI

4. 프로젝트 진행 최종 결과와 활용도

5. 작품 내에서 미흡했던 부분과 개선사항

5.1 지원비 부족

5.2 부품 지연 및 고장

5.3 서버 적용 문제

6. 참고자료 출처

- 참고자료 출처에 대한 목록과 소개

*. 부록

- 프로그램 소스 리스트

- 스마트 스위치 Arduino 스케치 소스(소스코드 - 스마트스위치_최종소스)
- 스마트 콘센서 Arduino 스케치 소스(소스코드 - 스마트콘센서_최종소스)
- Web UI 최종 소스(소스코드 - WebUI_최종소스)
- 모바일 어플리케이션 최종 소스(소스코드 - 모바일어플리케이션_최종소스)
- Connector UI 최종 소스(소스코드 - ConnectorUI_최종소스)

- Smart Home Hub 작동 Manual

1. 차세대를 위한 Smart Home Hub

1.1 작품의 배경 및 개요

스마트 홈 관련 제품들이 많이 출시되고 있지만 부담스러운 가격으로 사용하기가 꺼려집니다. 그래서 직접 부품들을 만들고 저희가 사용하기에 편한 환경으로 만들어보자. 라는 마음으로 작품을 만들게 되었습니다. 가격적인 부담을 줄이기 위해서 간단한 원리의 부품들만 사용하려고 계획했습니다. 그래서 스마트 홈 상품 중에서 스마트 콘센트를 제작했고 책상에서 많이 공부 및 작업을 하는 사용자를 대상으로 사용하기 편리한 스마트 데스크 환경을 제작하려고 계획했습니다. 그래서 라즈베리파이를 이용해 화면에 사용자에 대한 편의 정보와 제어 화면을 보여줄 수 있게 제작을 했습니다. 그리고 블루투스 제어만 활용해 가격적으로 비싸지 않게 제작했습니다. 그렇지만 블루투스 제어로도 WI-FI통신과 같이 원격통신 및 다중 제어가 가능하게 만들려고 계획했습니다.

1.2 작품 목표

작품의 목표는 스마트 홈 제품을 개발하는 것입니다. 하지만 기존의 스마트 홈 제품들과는 차별적으로 블루투스 통신만 이용해서 개발을 할 것이고 , 책상에 앉아서 제어를 할 수 있는 환경을 개발할 것입니다. 그래서 라즈베리파이를 활용해서 블루투스 제어 및 버스 정보를 보여줄 수 있는 화면을 목표로 했습니다. 대부분 외출 전에 대중교통을 이용하는 사람들은 버스 정보를 보고 나갑니다. 특히 날씨가 덥거나 추운 날에는 더욱 더 버스정보를 많이 찾습니다. 그래서 사용자 편의 제공에 있어서 버스 정보를 제공하는 것이 제일 적절하다 생각했습니다. 그리고 하드웨어적으로 간단한 원리로 만들어서 제품 비용을 간소화 시킬 것입니다. 이미 출시된 제품들은 멀리 떨어져 있어도 작동할 수 있도록 만들었기엔 가격적으로 비싸서 사용이 꺼려집니다. 하지만 블루투스 로만 사용이 가능하게끔 만들면 가격적으로 친숙하게 접근이 가능합니다. 그래서 블루투스 통신을 통해 제어 할 수 있는 전등과 콘센트를 만들 계획이고 시간의 여유가 있다면, 전등을 각도에 따라 조절이 가능하게끔 만들 수 있는 방안을 개발하려고 합니다. 그리고 라즈베리파이를 이용해 모든 스마트 기기들의 조작이 가능하게끔 만들고 라즈베리파이의 모니터를 달아서 그 모니터를 통해 생활에 필요한 정보들 (지도 , 날씨 , E-Mail 등등) 서버를 구축하여 사용자가 책상에서도 편리하게 사용할 수 있도록 할 예정입니다.

1.3 작품 주요 내용

모바일로 블루투스 통신을 할 경우 1 대 1 관계가 형성되면서 한 제품을 사용하다 다른 제품을 사용할 경우 블루투스 통신을 바꿔줘야 사용이 가능합니다. 하지만 라즈베리파이 같은 경우는 블루투스 통신이 1 대 다 관계가 형성되기에 다른 제품을 사용하더라도 블루투스 설정만 되어있으면 블루투스 통신을 바꿔주지 않아도 됩니다. 그래서 블루투스 통신이 가능한 스마트 홈 제품 (ex 스마트 스위치 , 스마트 콘센트)를 제작을 해서 모바일로 1 대 1 관계의 블루투스 통신을 하는 것을 보여드리고 , 라즈베리파이에서 1 대 다 관계의 블루투스 통신을 하는 것을 보여드리려고 합니다. 더 나아가서는 모바일로 라즈베리파이를 제어하여 모바일로도 블루투스 관계를 1 대 다 관계 형성을 할 수 있다는 것을 보여드리려고 합니다.

1.4 조원 별 역할 분담

BlueThings의 조원은 13학번 강인성 13학번 장승훈입니다.

첫 번째로 강인성 학우는 스마트 콘센트 제작에 대한 전반적인 아이디어 및 하드웨어 부분을 담당했습니다. 그리고 소프트웨어 부분에서 버스 정보를 얻어오는 방식에 대해서 공부 및 구현을 했습니다. 블루투스 통신을 통해서 모바일로 스마트 홈 제품을 제어하는 모바일 어플리케이션 제작도 했습니다. 그리고 라즈베리파이 부분에서 파이썬 2.7 GUI를 통해서 블루투스 제어 화면을 만들었습니다. 그 제어 화면을 통해서 저희가 만들고자 했던 블루투스 1 대 다 관계를 구현했습니다.

다음으로는 장승훈 학우는 스마트 스위치 제작에 대한 전반적인 아이디어 및 하드웨어 부분을 담당했습니다. 그리고 소프트웨어의 WEB-VIEW 부분을 담당해서 만들었습니다. 프로젝트 진행 및 관리를 담당했습니다. 하드웨어 디자인 부분에서 스케치업과 캐드로 도면을 만들기도 했습니다. 그리고 스마트 홈 제품 소스 코딩을 했습니다. 블루투스 통신과 토글 스위치에 대한 제어 소스를 코딩했습니다.

2. 이용 기술 소개

2.1 스마트 스위치, 스마트 콘센서



Arduino IDE



C++

스마트 스위치와 콘센서는 블루투스 통신을 기반으로 하여 Arduino IDE에서 하드웨어에서 얻어 오는 신호를 Byte 배열로 전송해서 Arduino UNO에 연결된 블루투스 모듈을 통해 시리얼 통신을 얻어 와서 스마트 스위치와 콘센서의 작동을 제어할 수 있도록 하였습니다. Byte 배열로 보내는 데이터는 "1\n", "2\n"으로 나뉘어서 보내게 되는데 전자를 이용해서 시리얼 통신에 응답을 하게 되면 기기들은 작동을 하게끔 구성을 하였고, 후자를 이용해서 시리얼 통신에 응답을 하게 되면 기기들은 종료를 하게끔 구성을 하였습니다.

2.2 모바일 어플리케이션



Oracle Java



Android Studio

최근 모바일 어플리케이션을 제작하는데 있어서는 Kotlin 언어가 대세로 오르고 있는데 현재 모바일 어플리케이션을 개발하는데 있어서 기본적인 작동을 구상을 하는 점에서는 Java를 이용해서 공부를 하는 것이 좋은 방안이라서 이를 택하게 되었습니다. 또한 모바일 어플리케이션을 제작하는데 있어서는 android 패키지를 이용을 해야 하는데 내부에 블루투스 통신과 관련된 패키지인 bluetooth를 제공을 해서 구현을 하는데 있어서 용이하게 이용을 할 수 있어서 큰 이익이 되었습니다. 또한 Android Studio 내부에서 Main Activity(메인 화면)에 대해서 xml 파일을 이용해서 작성을 해야 하는 점이 어렵지만 화면 내부에서 View 객체들을 직접 선택을 해서 가져와서 이용을 할 수 있도록 하는 점이 있어서 처음 모바일 어플리케이션을 만들어 보는데 있어서 크게 어렵지 않게 접근을 할 수 있고, 나중에 모바일 어플리케이션 방향으로 나아갈 생각이 있으면 xml 파일을 이용해서 직접 설정을 해 보는 연습을 하는 것이 좋을 것 같습니다.

2.3 라즈베리파이 모니터 어플리케이션

라즈베리파이 모니터 내부에 있는 어플리케이션은 크게 2가지로 나눌 수 있어서 다음 Project 소개에서는 이렇게 2가지로 나뉘어서 설명을 하는 것이 용이해서 각 어플리케이션들에 대해서 나누어서 설명을 하겠습니다.

2.3.1 Web UI



Oracle Java



Spring
Web MVC

과거에는 Java 내에서 데이터베이스와 연동을 하기 위해서는 JDBC를 이용해서 데이터를 얻어오는 방법이 있었지만 과거와는 달리 Spring Framework 내부에서도 JDBC와 MyBatis, JPA 등을 이용해서 데이터베이스를 받아오는 방안이 생겨서 데이터베이스와 Spring Web MVC를 이용해서 데이터베이스 내부에 있는 데이터들을 받아오는데 있어서 용이해졌습니다. MVC의 개념으로서는 Model, View, Controller 3가지로 나눌 수 있는데 Model은 실제 Web 어플리케이션 내부에 쓰이는 데이터들을 담아주는 Attribute와 같은 역할을 하게 되는데 저희가 공공데이터 API에서 얻어온 데이터들을 JSON으로 변환을 해서 이 Model에 주입을 시킵니다. 그 다음에 View는 Web 어플리케이션 내부에서 어느 Web 페이지(JSP, Html 등등)에 보내지는가에 대하여 주소를 남기는 개념으로 생각을 하시면 됩니다. 그럼 최종적으로 Web 어플리케이션을 작동을 하면 요청 주소를 알맞게 매칭을 시킨 페이지로 이동을 하게끔 해주는 역할을 하는 것이 바로 Controller입니다. Controller 내부에는 Service 객체를 이용해서 접근을 하는 것이 올바른 방법이지만, Spring Web MVC 내부에서 Bluetooth와 접속을 하는 점에 대해서 많이 알아봤지만 과거 2008년 쯤 Java 언어에서도 Bluetooth 디바이스 연동을 하는 프로젝트가 존재를 해서(Bluecove) 이를 이용해서 Service 객체에 주입을 하는 것을 시도해 봤지만 이 API가 제공하는 Bluetooth 체계에 대한 구조와 현재 JDK 버전과의 호환성이 안 맞게 되어서 컴파일 오류가 나오게 되었습니다. 또한 Java에서 블루투스 통신을 하는 추세가 줄어들어서 현재 시점으로는 모바일 어플리케이션 내부에서 블루투스 통신을 하는 축으로 방향을 두게 되어서 아쉽게도 블루투스 통신에 대해서는 JDK 업데이트가 되는 호환성이 안 맞게 되는 점을 고려하여 Web 페이지 상에 블루투스 통신을 하는 기능을 두는 것이 아닌 후술할 Connector UI를 만들어서 Bluetooth 통신을 하는데 있어서 차질이 안 생기게끔 해서 프로젝트에 대한 유지보수를 줄이도록 구상하였습니다.

데이터를 검색해보세요!



1. 지도

여러분이 원하는 데이터,

데이터 1번가



로 신청해주세요!

[그림] 공공데이터 API 홈페이지(<http://www.data.go.kr>)

2.3.2 Connector UI

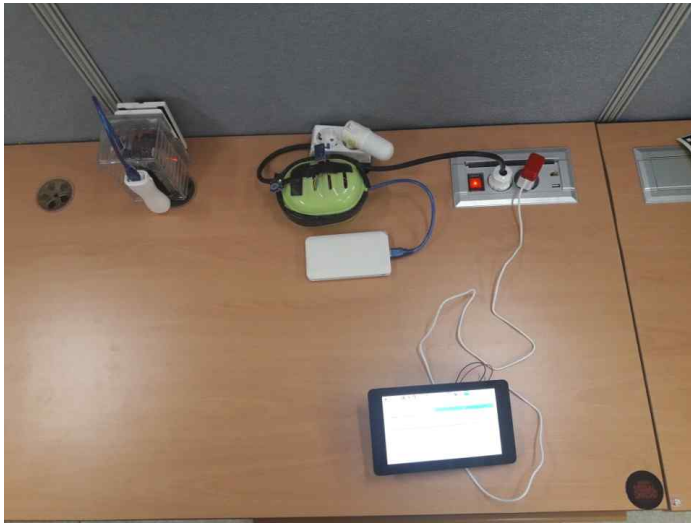


Python 2.7

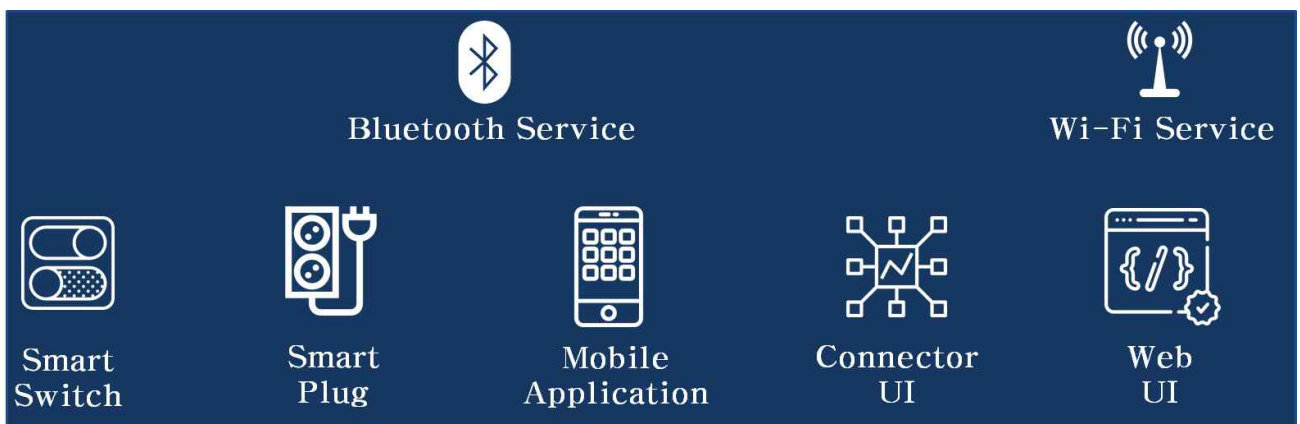
Python 언어는 Java와 마찬가지로 객체 지향 언어 체계로 되어 있어서 최근에 코딩 교육 관련되어 많은 화제가 되고 있는 언어들 중에 하나로 볼 수 있습니다. Python에서도 GUI(Graphic User Interface)를 용이하게 해서 운영 체제와 상관없이 화면을 사용자가 직접 GUI를 구성해서 화면을 제공할 하는 Tkinter 패키지를 이용해서 Button, Label, Graphic 등의 객체를 직접 구성하고 만들어서 pack 방식을 이용해 배치를 해서 이용을 하는 것이 원칙입니다. 그래서 저희 조는 Python을 이용해서 화면을 직접 제작해서 Button과 Label을 이용해서 사용자가 스마트 콘센트, 스위치 작동 여부에 대한 정보를 보여주는 방법을 이용해서 구성을 하였습니다. 또한 Python 패키지 중에서 Serial이란 패키지가 있는데 이를 이용해서 블루투스 통신을 하는데 있어서 여러 디바이스들에 대한 제어를 하는데 큰 기여를 하게 되어서 이를 기반으로 Connector UI를 완성을 했습니다. 또한 Raspberry PI를 이용하여 타 기종 Bluetooth Device에 대한 접속을 Android 모바일과 다르게 여러 통신을 제공할 하는데 저희가 구상했던 Smart Desk처럼 이용을 하는데 있어서 어렵지 않게 구현을 할 수 있게 되었으며, Tkinter 패키지가 타 OS로 작동을 할지라도 Python 만의 호환성을 그대로 전달을 받을 수 있어서 유지보수를 하는 방면에 있어서는 효율적으로 도움이 됩니다.

3. Project 소개

저희가 진행한 프로젝트를 통해서 최종 결과물에 대하여 5가지로 나누어서 볼 수 있는데 스마트 스위치, 스마트 콘센서, 모바일 어플리케이션, Web UI, Connector UI로 분류가 됩니다. 프로젝트 진행 언어에 대해서는 앞에서 언급을 하였기 때문에 이 본문에서는 각 프로젝트에 대한 구조 설명과 어떠한 원리를 이용해서 작동을 하는가에 대해서 설명을 하겠습니다.



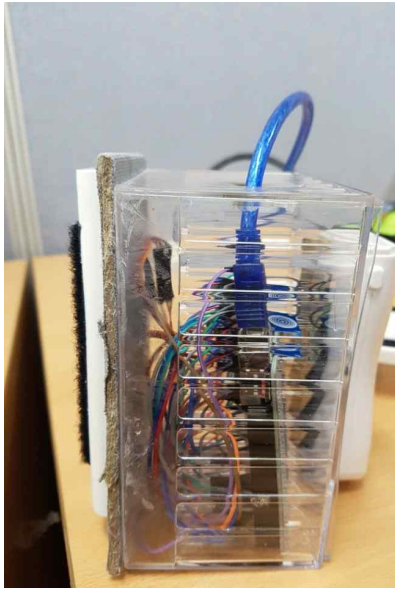
[그림]
프로젝트 최종 완성본



[그림] Smart Home Hub 구성 요소도

저희 측에서는 무선 통신들 중에서 대표적으로 선택할 수 있는 Bluetooth 통신과 Wi-Fi통신 두 가지로 나누어서 Smart Home Hub를 제공을 하였습니다. Bluetooth Service 버전은 4.0을 이용을 하였으며, Wi-Fi Service는 비록 학교에서 진행을 하는 경우가 대다수라서 Web UI를 띄우는데 대기 시간이 걸리는 제약 조건이 있었지만, 사용자가 실시간으로 필요한 데이터들에 대해서 빠른 시간 내에 얻어올 수 있도록 구성을 했습니다.

3.1 스마트 스위치

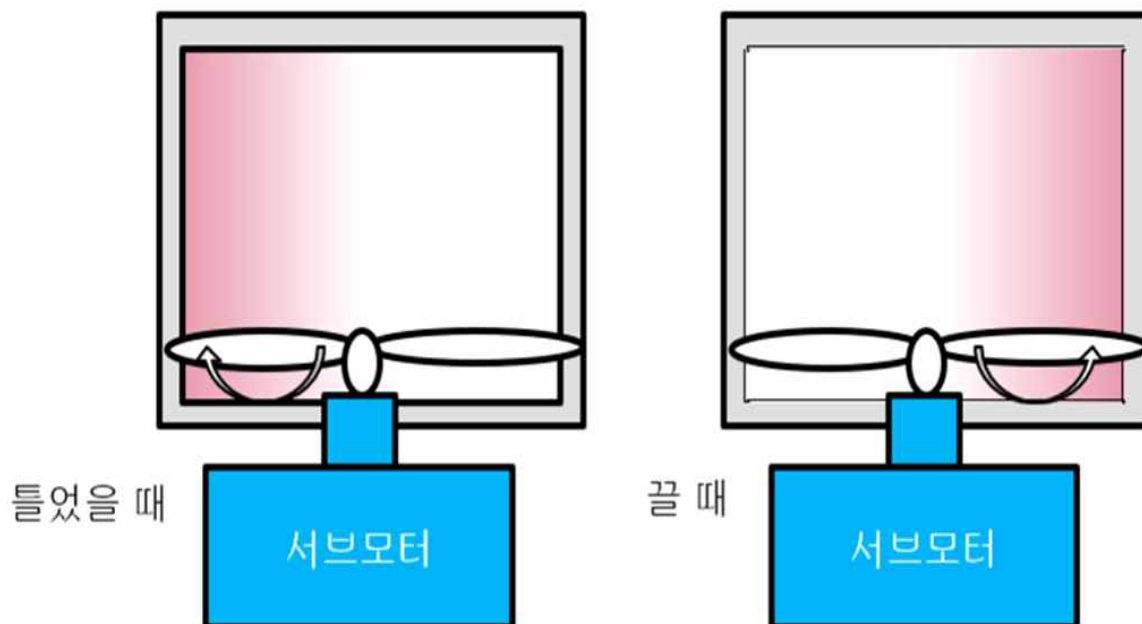


[그림]

(左) 스마트 스위치 옆모습

(右) 스마트 스위치 앞모습

스마트 스위치는 서보모터의 각도를 제어하면서 그 제어된 각도에 따라 부착된 스위치를 제어하도록 만들었습니다.



[그림] 스마트 스위치 서보모터를 이용한 제어 구상도

일단 구상도를 보시면 저렇게 불을 켜는 블루투스 통신을 했을 시 서보모터의 각도를 제어하면서 서보모터가 스위치를 켜고 불을 끄는 블루투스 통신을 했을 시 서보모터의 각도를 제어하면서 서보모터가 스위치를 끄는 방식으로 제작을 했습니다. 그래서 제작 부품은 아두이노 Uno 보드 , 블루투스 모듈 , 터치 스위치 , 서보 모터 , 브레드 보드 이렇게 부품을 최대한 간소화 시켰습니다. 그리고 전원 공급은 원래 건전지로 하려고 했으나 전압이 일정하지 않아 작동 중 오류가 많이 일어나서 전압이 일정한 보조 배터리로 제작을 했습니다. 사용자 입장에서든 배터리 충전도 용이하고 굳이 건전지를 교체할 필요 할 필요가 없어서 더 사용이 용이하다고 판단했습니다.



스마트 스위치 뒷면 보조배터리 부착 부분

보조 배터리 부착은 라벨로 붙였습니다. 때낼 수 있게 제작을 했습니다. 그리고 스위치 부분에 부착하는 부분은 모터 부분보다 앞에 있어야 합니다. 그래서 일반적인 스위치 앞 커버를 따서 모터 앞부분에 붙였습니다. 그리고 보조 배터리를 연결하기 위해서는 usb 선이 나와야하는데 그 구멍을 만들기 위해서 라이터로 가열 후 가위로 구멍을 내는 방식으로 플라스틱 부분에 구멍을 내서 케이스 제작을 했습니다. 이 방법으로 하실 경우 연기가 많이 생기니 창문을 열고 하시기 바랍니다. 그래서 사용하고 계시던 스위치 부분에 암 라벨을 붙이시면 스마트 스위치가 부착이 됩니다. 그러면 모바일 제어나 라즈베리 파이 제어를 하시면 스마트 스위치 사용을 하실 수 있습니다. 침대에 누워서 스마트 스위치를 사용해서 형광등을 끌 수도 있고 책상에 앉아서 스마트 스위치를 사용해서 형광등을 끄고 스탠드를 켤 수도 있고 집을 나가기 전 스마트 스위치를 사용해서 간편하게 형광등을 끄시고 나갈 수 있습니다. 물론 스마트 스위치를 사용해서 형광등을 키실 수도 있습니다.

3.2 스마트 콘센서



[그림]

(左) 스마트 콘센서 작동 이전

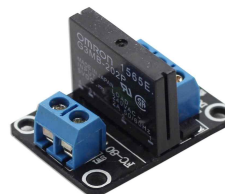
(右) 스마트 콘센서 작동 이후

Bluetooth 통신을 이용한 Serial 통신을 활용해서 스마트 콘센서를 제작을 하게 되었는데 시중에 판매하고 있는 스마트 콘센서는 Wi-Fi를 기반으로 한 제품들이 많이 나오고 있어서 최소 통신 거리가 10m인 방 안에서 이용하기 좋은 통신이 Bluetooth로 판단을 해서 이를 활용한 스마트 콘센서를 제작을 하게 되었습니다. 본론에 들어가기 앞서 스마트 콘센서가 작동하게 되는 원리를 설명하면, 릴레이(Relay)를 활용을 하게 되는데 여기서 릴레이의 종류는 크게 2가지로 나뉘어 볼 수 있습니다. 전자식 릴레이, 무접점 반도체 릴레이(Solid State Relay)로 볼 수 있는데 전자식 릴레이는 전류를 흐르게 하는 전자의 원리를 이용해서 접점을 물리적으로 움직이게 하여 접점을 개폐하게끔 만드는데 있어서 쓸 수 있는 릴레이인데, 이 릴레이는 구동하는 소비 전력이 크고 물리적인 동작을 동반하는데 있어서 응답 시간이 느린 점에다가 릴레이 내부에 기계적인 요소가 있어서 소음에 예민한 사람들에게 있어서 부적합합니다. 그렇지만 무접점 반도체 릴레이는 작은 압력 전력으로 큰 출력 전압을 개폐하는데 있어서 매우 효과적인 릴레이로 볼 수 있으며, 전자식 릴레이와 달리 접점의 수명, 소음 문제, 전류의 누전 등을 방지할 수 있도록 하는데 있어서 매우 큰 효과를 볼 수 있기 때문에 저희 측에서는 무접점 반도체 릴레이를 이용해서 스마트 콘센서를 제작하게 되었습니다.



[그림]

전자식 릴레이

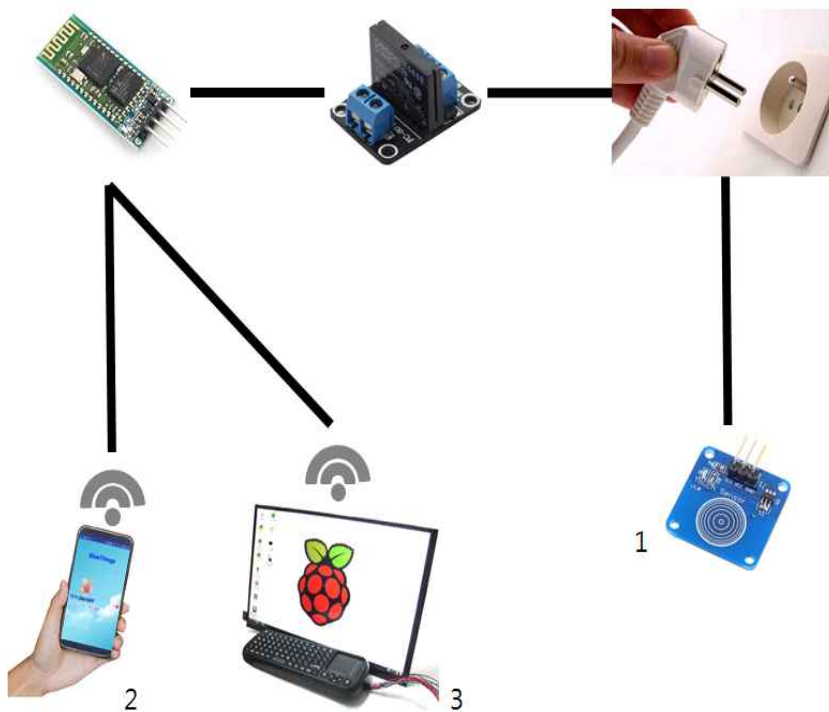


[그림]

전자식 무접점 릴레이
(Solid State Relay)

무접점 반도체 릴레이와 Bluetooth 통신을 이용해서 스마트 콘센서를 만든 결과물의 구조들을

살펴보면 오로지 무접점 반도체 릴레이와 콘센터 사이의 관계만 두면 효율성 없는 콘센서가 되어서 저희 측은 터치 센서를 추가를 해서 Bluetooth 통신이 불가능한 경우를 대비를 해서 사용자가 지속적으로 스마트 콘센터를 이용을 하는데 있어서 오작동이 되지 않도록 사전에 방지를 하였습니다.



[그림]

Smart Plug 작동 원리

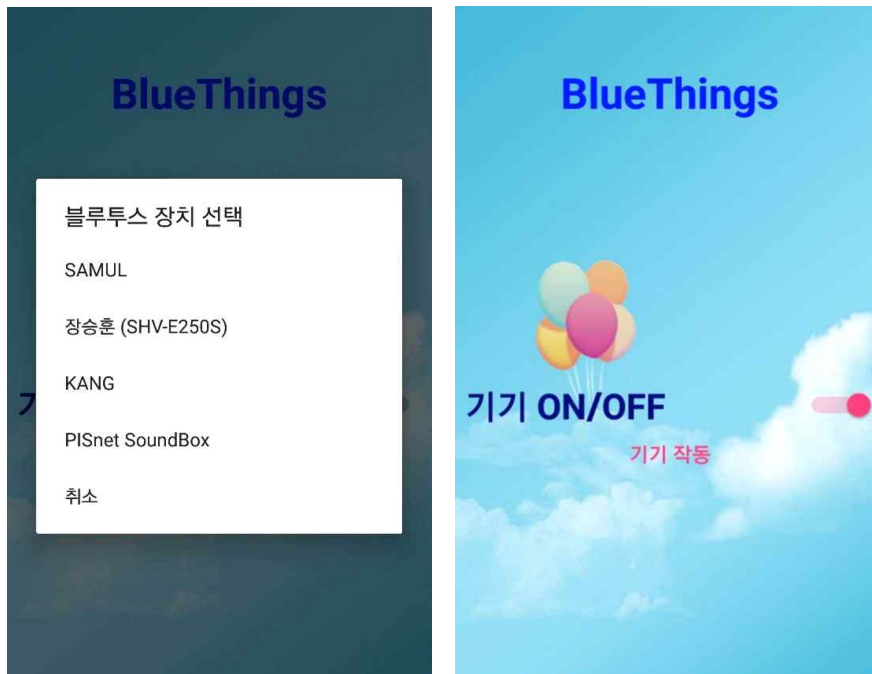
1 : 터치 센서를 이용한 제어

2 : 스마트폰 Bluetooth 통신을 이용한 제어

3 : 라즈베리 파이 모니터 통신을 이용한 제어

우선 저희 측은 터치 센서를 이용한 제어를 채택한 이유가 콘센서가 대부분 토글스위치를 이용하는 점에 대해서 아동들이 젓가락 등을 이용해서 남용을 하지 못하게끔 사고를 방지하기 위해 오로지 원 터치를 이용해서 작동을 하게끔 하여 안전성을 기여하였습니다. 그리고 집에서 모바일 어플리케이션이나 Raspberry PI 모니터 어플리케이션으로 제어를 하게 되면 Serial 통신을 이용해서 무접점 반도체 릴레이 작동과 종료를 구분함으로써 기계를 잘 못 다루는 사람들에게 도움을 줄 수 있도록 하는 점을 목표로 두고 제작을 했습니다. 그리고 2구 콘센터를 이용해서 여러 플러그들을 꽂게 만드는 컴퓨터나 계절성 전자제품인 에어컨, 전기장판 등을 한꺼번에 제어를 해주게끔 하여 전기 비용을 감축시키게끔 하는 점도 목표로 두고 스마트 콘센터를 제작하였습니다.

3.3 모바일 어플리케이션



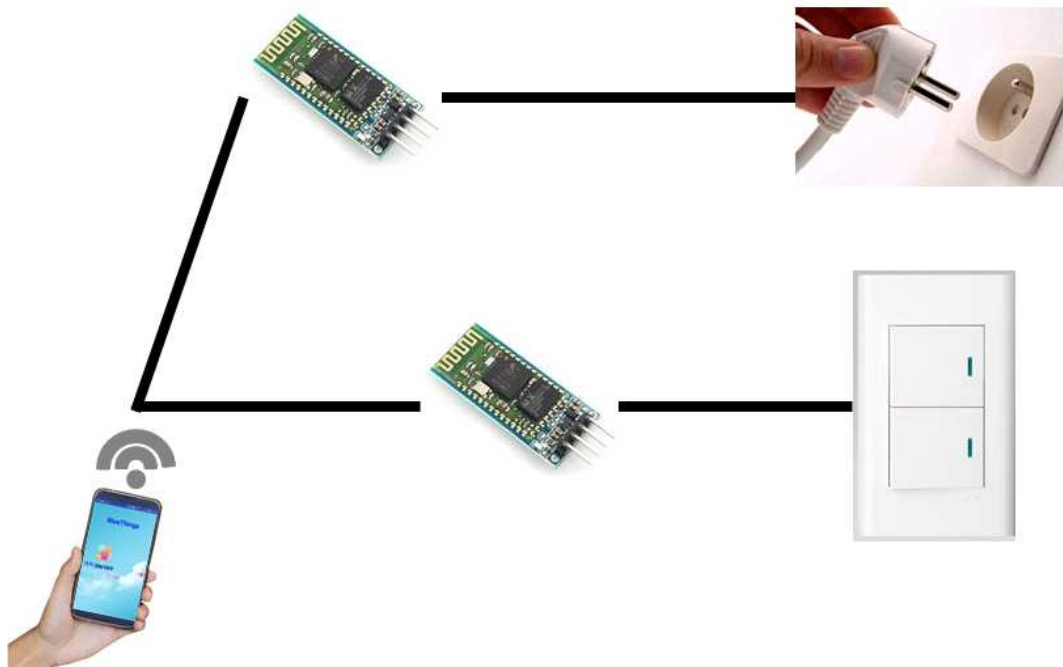
[그림]

(左) 모바일 어플리케이션 실행 작동 이전(Device 선택)

(右) 모바일 어플리케이션 실행 작동 이후
(기기 작동, 종료 제어)

모바일 어플리케이션은 Android 운영체제를 기반으로 제작을 하였습니다. Java Android에서는 Bluetooth 통신을 할 수 있는 기능들에 대해서 제공을 하고 있습니다. Android System에서는 Bluetooth 통신을 하기 위해서는 하나의 스마트폰으로 하나의 Smart Device를 연동을 하게끔 하도록 선택을 하였습니다. 이는 저희 측에서 Smart Device가 새로운 기기를 나오게 된다면 모바일 어플리케이션 인터페이스를 바로 적합하게 이용을 할 수 있도록 하기 위한 점을 염두를 하고 제작을 하게 되었습니다. 모바일 어플리케이션의 구조는 첫 화면에 사용자가 작동하기 원하는 Device들을 선택을 하게끔 만들었으며, 모바일 Device와 Smart Device가 서로 통신을 주고받기 가능한 상태이면 어플리케이션을 종료하기 이전까지 지속적으로 통신을 하게끔 구성을 하였습니다. 모바일 어플리케이션의 구도는 간단하게 구성이 되어 있지만, 모바일 어플리케이션을 실행하고 난 이후에 Bluetooth 작동을 하지 않은 경우에는 이를 작동을 하게끔 접근성 권한을 추가로 관리하게끔 어플리케이션을 제작하였으며, Bluetooth 장치 선택을 통해서 자신이 연결을 하고 싶은 Device를 선택을 해서 모바일과 Smart Device(스마트 스위치, 스마트 콘센서)와의 데이터를 통신을 하도록 유도를 하였습니다. 그리고 스위치 버튼을 통해서 스마트폰을 다루기 어려운 어르신이나 어린이들도 한 버튼을 통해서 굳이 전등 스위치를 누르지 않더라도, 아니면 콘센서를 빼지 않더라도 간편하게 전기를 제어를 할 수 있도록 이러한 기능들을 다루기 쉽게 제공을 하는데 기여를 했습니다. 스위치의 작동 여부에 따라서 작동을 하기 위한 문자열을 byte 배열로 변환을 해서 Serial 통신을 하도록 유도를 하는데 있어서 고려를 많이 하였는데 Java 내에서 문자열을

byte 배열로 변환을 해서 Smart Device와 연동을 하는데 있어서 각 Smart Device 내부 시리얼 코드 인식을 하는 부분을 한 번 더 확인해서 구현을 해둬으로서 모바일과 Smart Device 사이의 연동을 확인하게끔 만들었습니다. 하지만 모바일 어플리케이션에서 한 번 더 확인을 해야 하는 사실이 있는데 이에 대해서는 보안 사항에서 후술을 해두겠습니다.



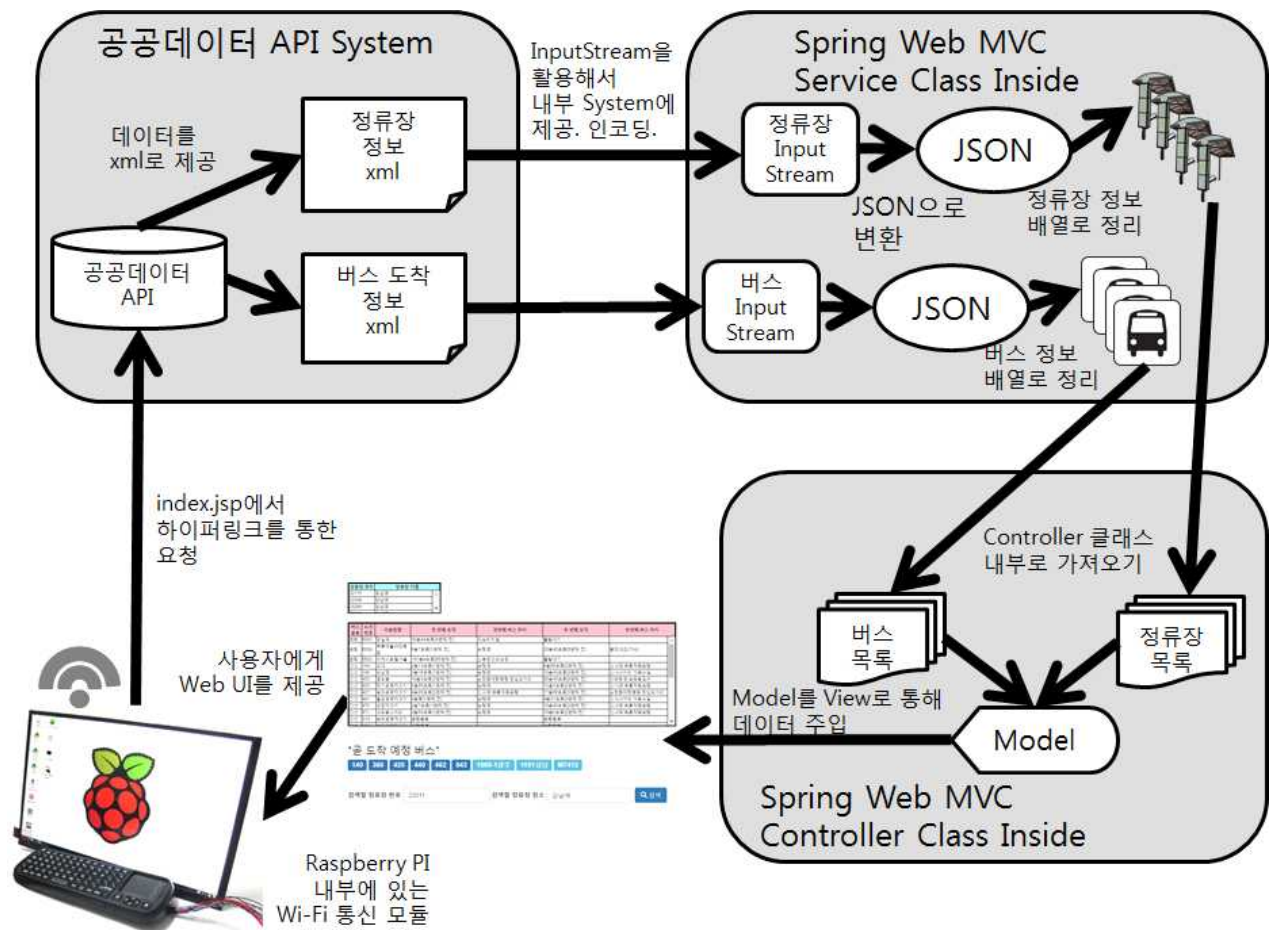
3.4 라즈베리파이 모니터 어플리케이션

라즈베리파이 모니터에는 어플리케이션을 2개로 구성을 해서 사용자가 실시간으로 필요한 공공 데이터들의 정보를 통해서 실시간으로 변화를 보여주게끔 구상을 한 Web UI 어플리케이션과 사용자가 각 Smart Device들을 모바일 어플리케이션 내부에서 1:1로 신호를 줄 수밖에 없었던 점을 보완하기 위해서 라즈베리파이 모니터 내부에서 각 Smart Device 별로 한꺼번에 관리를 할 수 있도록 기기와 모니터 사이의 Bluetooth 통신을 가능하게 하는 Connector UI 2가지로 나뉘어서 구체적으로 어떠한 기능을 이용해서 구현을 했는가에 대해서 언급을 하겠습니다.

3.4.1 Web UI

Web UI는 Java 프로그래밍 언어를 기반으로 작성을 할 수 있는 Spring Web MVC Framework를 이용하여 사용자들에게 실시간으로 제공하는 공공데이터들에 대해서 제공을 할 수 있도록 구현을 했습니다. 일단 Web UI에서 버스 정보들이 테이블에 완성되기까지의 과정을 기술적인 요소가 워낙 복잡하기 때문에 그림을 참고하면서 설명을 하는 방식으로 작성을

하겠습니다.



[그림] 라즈베리 파이 모니터 내부에서 제공하는 Web UI 매커니즘

본래 Web MVC 개념에 대해서는 Model, View, Controller 3가지로 나뉘는데 Model의 역할은 Web 하이퍼링크(Hyperlink)를 통한 요청을 토대로 넣을 데이터들에 대해서 묶어두는 Request Attribute의 역할을 하게 되고, 이를 토대로 View를 통해 사용자에게 데이터를 보여주는 역할을 하게 됩니다. 그 다음 View에서는 프론트 엔드를 통한 웹 페이지를 구축을 한 구조들에 대해서 각 데이터들이 어디에 보일지에 대해 선언을 함으로서 그 웹 페이지를 Web Application에서 보여주는 역할을 합니다. 마지막으로 Controller에서는 Model을 정리함과 동시에 View를 보여줄 주소를 정리를 하게 되면 최종적으로 Hyperlink에서 요청한 Web Application을 작동을 하게끔 하는 원리로 이용을 한다고 보시면 됩니다. 저희가 공공데이터 API(<http://www.data.go.kr>)를 참고를 해서 대부분의 오픈 API는 xml 파일로 제공을 하는 경우가 대다수인데 xml 파일들에 대해서 InputStream을 통해서 Java 어플리케이션 내부에서 읽어 들이게 됩니다. 공공 데이터 API에서 데이터를 얻어오기 위해서 필요한 정보는 크게 2가지

로 볼 수 있는데 하나는 서비스 키, 하나는 각 오픈 API 별 매개 변수들로 볼 수 있습니다. 서비스 키는 오픈 API 내부에서 쓸 수 있는 데이터들에 대하여 접근 가능 여부를 판단하는 비밀 번호와 같은 역할을 합니다. 그래서 이 서비스 키에 대해서는 철저한 보안을 위하여 properties 파일을 통해 묶어 두는 방법이 좋은 방안입니다. 두 번째로 API 별 매개 변수로 볼 수 있는데 저희가 이용한 Open API 요청 주소에 대해서는 아래와 같이 나뉘어서 볼 수 있습니다.

NO	상세기능	설명	활용제한 여부	일일 트래픽	심의결과	미리보기 다운로드						
1	getStationByNameList	정류소 명칭 검색	-	1000	승인	<div>실행</div>						
<div>요청변수(Request Parameter)</div> <div><div>초기화</div><div>닫기</div></div> <table><thead><tr><th>항목명</th><th>샘플데이터</th><th>설명</th></tr></thead><tbody><tr><td>stSrch</td><td><div>가곡초교</div></td><td>정류소명 검색어</td></tr></tbody></table>							항목명	샘플데이터	설명	stSrch	<div>가곡초교</div>	정류소명 검색어
항목명	샘플데이터	설명										
stSrch	<div>가곡초교</div>	정류소명 검색어										
2	getStationByUidItem	정류소고유번호를 입력받아 버스도착정보 목록을 조회한다.	-	1000	승인	<div>실행</div>						
<div>요청변수(Request Parameter)</div> <div><div>초기화</div><div>닫기</div></div> <table><thead><tr><th>항목명</th><th>샘플데이터</th><th>설명</th></tr></thead><tbody><tr><td>arsId</td><td><div></div></td><td>정류소고유번호</td></tr></tbody></table>							항목명	샘플데이터	설명	arsId	<div></div>	정류소고유번호
항목명	샘플데이터	설명										
arsId	<div></div>	정류소고유번호										

[그림] 공공데이터 API 요청 변수들 사례

이처럼 서비스 키, 요청 변수들에 대한 이용을 하게 되면 사용자가 이를 통한 검색을 하게 되면 각 데이터 별 결과를 출력을 할 수 있도록 하는 목표를 두고 구현을 완료했습니다. 또한 공공 데이터 API 뿐만이 아니라 대기업에서 제공하는 공공 데이터에 대한 이용 신청을 하게 되면 타 API 내부에서 필요한 요소들에 대해서 맞춰서 작성을 하게 되어도 대부분 JSON을 이용해서 제공을 하게 됩니다. 또한 저희 조도 xml 파일에 대하여 그대로 쓰는 것보다는 차후에 JavaScript를 이용하여 JSON을 기반으로 데이터를 받아서 쓸 수 있도록 하기 위해서 JSON으로 변환을 해주는 작업을 하였는데 JSON의 장점으로 객체 지향성을 이용하는 프로그래밍 내부에서 객체를 이용해 데이터를 주입하는데 문제없는 점과 다양한 데이터들에 대해서 배열이나 객체 형으로 제공을 하는 점이 최근에 데이터베이스 기술 중에 NoSQL에서 접목하기 좋은 점인

것 같습니다. 허나 단점으로서는 숫자 형 데이터들에 대해서 파싱(Parsing)을 하는 면에서는 아직 생각을 해야 하는 점이 많은 것 같습니다. 본래 JavaScript나 Python 내부에서 숫자 형 데이터는 정수와 유리수에 대한 구분이 따로 없어서 제한 없이 이용이 가능하지만, Java나 C++ 내부에서는 정수와 유리수를 int와 double 형 등등의 형태로 변환을 개별적으로 해줘야 하는 점에서는 단점으로 볼 수 있습니다.

정류장 코드	정류장 이름
22173	강남역
22339	강남역
23285	강남역

버스 종류	노선 번호	이동방향	첫 번째 도착	첫 번째 버스 위치	두 번째 도착	두 번째 버스 위치
공통	8000	잠실역	38분44초후[4번째 전]	고속리미트	출발대기	
공통	8009	북촌마을아파트 앞	5분7초후[1번째 전]	북현역	23분40초후[5번째 전]	동작대교(가상)
공통	8500	더케이호텔서울	147분9초후[55번째 전]	신촌역	출발대기	
간선	144	교대	2분13초후[1번째 전]	북현역	8분59초후[2번째 전]	신사역
간선	145	강남역	3분15초후[1번째 전]	북현역	10분43초후[3번째 전]	신사역
간선	400	영곡동	18분3초후[3번째 전]	신촌역	26분48초후[8번째 전]	이태원역
간선	402	장지공원차고지	3분55초후[1번째 전]	북현역	13분3초후[2번째 전]	신사역
간선	407	송파공원차고지	9분25초후[2번째 전]	신사역	17분55초후[3번째 전]	신촌역
간선	441	월암공원차고지	3분후[1번째 전]	북현역	9분21초후[3번째 전]	신사역
간선	470	상암차고지	2분7초후[1번째 전]	북현역	10분40초후[2번째 전]	신사역
간선	471	세곡동사거리	3분53초후[1번째 전]	북현역	15분2초후[2번째 전]	신사역
간선	N13	송파공원차고지	출발대기	신촌역	출발대기	신사역

곧 도착 예정 버스

140 360 420 440 462 542 1005-1광주 1151성남 M7412

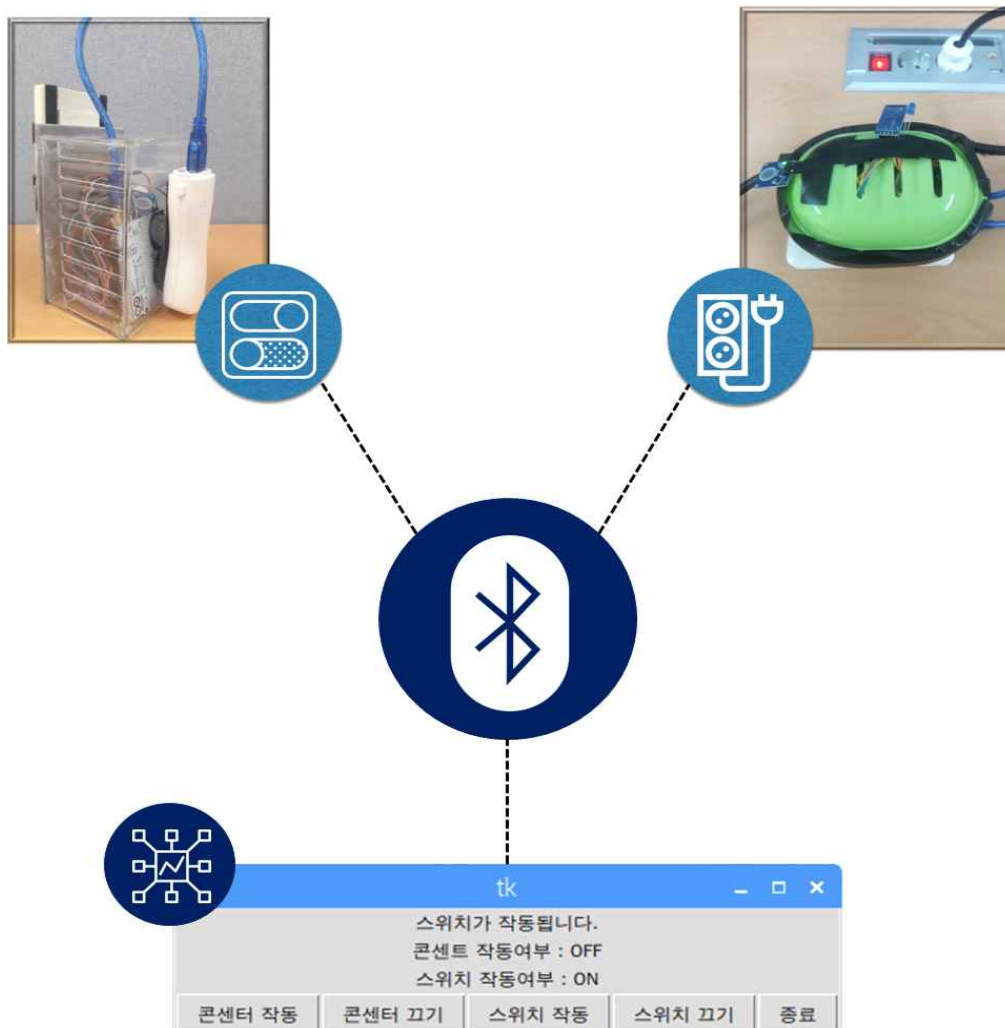
검색할 정류장 번호 검색할 정류장 장소

[그림] Web UI 최종 제작 Screenshot

3.4.2 Connector UI

Python 2.7 버전을 기반으로 제작한 Connector UI는 Python에서 기본적으로 제공하는 package에서 Tkinter 가 있는데 이는 Python 프로그래밍 언어를 어떠한 운영체제를 이용한다 할지라도 윈도우 창속에 있는 버튼, Label 등등의 요소들을 코딩을 한 결과에 따라서 그대로 전달을 하게끔 하는 GUI를 제공하는데 있어서 많은 도움이 되었습니다. 또한 Python 패키지 내부에도 Bluetooth 통신을 할 수 있도록 Serial 패키지를 이용을 해서 문자열을 Byte 배열로 변환을 해서 전달을 하는데 있어서 좋은 결과가 나오게 되었습니다. Connector라는 뜻은 1960

년대 과거에 전화를 교환을 해서 받아서 이용을 했던 시절에 발신자에게서 수신자에게로의 회로를 구성하는 점점으로 볼 수 있듯이, 실제로 저희가 제작한 Smart Device와 Raspberry PI Monitor와의 Bluetooth 통신을 원활하게 이뤄질 수 있도록 하는 중계자의 역할을 함으로서 Connector라는 명칭을 붙였습니다. 저희가 생각했던 원리는 웹 페이지 내부에서 Bluetooth 통신을 해보려고 노트북에서 Tera Term(Bluetooth Device와 노트북 사이의 중계자 역할)을 활용해서 실제로 Java 프로그래밍 속에 Swing을 활용한 Robot 클래스를 이용해서 웹 사이트 내에서도 Bluetooth 연동을 시도를 했으나 Raspberry PI에서는 exe 파일을 실행을 하는 방법에 대해서는 MS-DOS에서 실행이 가능하기 때문에 실제로 Tera Term을 이용을 하는 것은 PC 개발 환경에서 가능하기 때문에 많은 난해감이 있었습니다. 그래서 직접 Python에서 제공을 하는 Tkinter 패키지를 활용해서 아래와 같은 창을 연동시켜서 완성을 했습니다. Connector UI의 구성은 아래와 같습니다.



[그림] Connector UI 화면 구조와 작동 구도

Raspberry PI Monitor에서 Touch 기능을 이용을 하게 된다면 남녀노소 부담 없이 이용을 할 수 있도록 구성을 하였습니다. 일단 저희가 작동을 하는 Smart Device들에 대해서 구현을 했는데 앞으로도 새롭게 만들 Smart Device들과 연동을 하게 되면 Raspberry PI에서 Bluetooth 통신을 1대 多로 제공을 함으로서 원 터치를 통하여 전자 기기들에 대한 편의성을 제공을 하는데 도움이 되도록 구성을 하면 도움이 될 듯합니다. 각 기능 별로 콘센서, 스위치의 버튼을 각각 누른다면 Python 코드 내에서 돌아가는 클래스 내부에서 Serial 통신을 주고받음으로서 스위치나 콘센서들에 대한 작동을 제어할 수 있도록 구성을 하였습니다.

4. 프로젝트 진행 최종 결과와 활용도



[그림] 스마트 스위치 일부들 중에서 스마트 폰을 이용해서 각종 전자 제품들의 작동뿐만 아니라 기기 내부의 강약까지 조절할 수 있는 YOSWIT 제품

저희 조 진행 최종 결과를 보시면 스마트 Device와 Raspberry PI, Mobile Application 간 연동을 하는데 있어서 저희가 바라던 결과에 대해서 충족을 시켰습니다. 하지만 뒤에서 다시 언급하겠지만 실제로 저희가 아쉬웠던 점은 Mobile Device에서 각 Smart Device끼리 연동을 하는 관계성에서 좀 미흡했습니다. Android에서 Bluetooth 디바이스를 연동하는데 있어서 1:1 관계가 나오는 점인데 실제로 경험 상 태블릿 PC를 이용할 때 Bluetooth 키보드와 마우스를 서로 같이 이용을 하는데 둘 중 하나만 작동이 되기 때문에 결국에는 마우스는 OTG를 이용해서 연결을 해서 이용을 하고 있습니다. 실제로 집에서 거리가 먼 학생들 중에서 자취를 하는 학생들이 제일 걱정을 하는 것

이 바로 전기세입니다. 또한 전자 제품을 잘 활용을 하지 못하는 노년층, 아동들이 전자 제품을 잘 못 이용하게 되면 사고를 유발을 할 수 있기 때문에 최근에 스마트폰 이용률을 반영해서 많은 사람들이 스마트폰 하나를 이용을 해서 집 근처에 있는 모든 전자 제품들을 관리할 수 있는 방향으로 고려를 했는데 저희 측에서는 Blueinno를 다뤘던 개념들 중에서 릴레이, 서브 모터 2가지를 이용을 해서 Smart Device들을 만들어보면서 현재 쓰이고 있는 시제품들과 비교를 하면서 제작을 했지만, Bluetooth 통신에 대해서 집 근처 거리에서 작동이 되는 점을 고려를 한다면 기기 이용 시각까지 작동을 하는 점에 대해서까지 고려를 하려면 Wi-Fi 기술을 이용을 해야 하지만 학교 주변이 Wi-Fi 환경과 저희 측에서 서버와 네트워크 관련되어 공부를 해 본 경험이 부족해서 그 부분에 대한 구현을 못 해서 아쉽습니다. 여기서 Bluetooth 통신을 이용하기 좋은 장소가 원룸, 오피스텔 등 한 방에 있는 모든 전자제품들을 다루는데 있어서 원활한 장소로 볼 수 있고, 또한 호텔이나 민박 집 안에서 과거에 카드를 꽂아서 모든 전자제품들의 작동을 한꺼번에 하는 것이 아닌 일부 제품들만 작동을 하도록 하는 기능을 첨가하면 호텔 운영을 하는데 있어서 큰 도움이 됩니다. 또한 저희가 구상한 프로젝트에 지문 인식 기능을 첨가해서 투숙객이 확인 된다면 저희가 구상한 제품들이 작동을 하면서 콘센터, 스위치뿐만 아니라 에어컨, 냉장고, 블라인드 등에 활용을 하면 원 터치를 효율적으로 이용을 해서 Smart Hotel을 제공할 수 있어서 이러한 기능들에 대해서도 차후에 유지보수를 하면서 기능들을 추가하는 방법에 대해 공부를 하게 된다면 좋은 결과가 나올 것으로 판단이 됩니다.

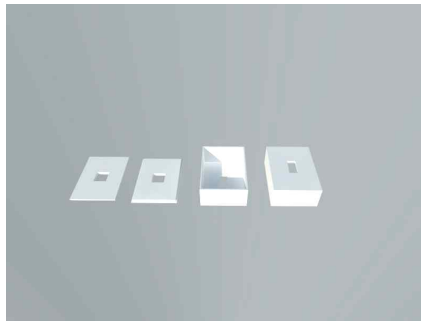
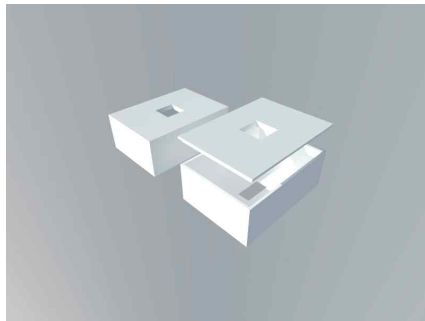


[그림] 실제로 Smart Home Hub 기술을 활용하는 사례[스마트 호텔]

5. 작품 내에서 미흡했던 부분과 개선사항

5.1 지원비 부족

스마트 제품 케이스가 원래는 아크릴 케이스에 락커를 칠해서 제작을 하려고 계획했습니다. 그래서 스케치 업과 캐드를 통해서 도면을 제작했습니다. 그리고 제작 의뢰를 했지만 생각보다 비싼 가격으로 인해서 조금 더 싼 가격의 시중에 파는 다용도 케이스를 가공해 케이스를 제작했습니다. 만약 아크릴 케이스로 제작을 했다면 깔끔한 디자인으로 나올 수 있었지만 아크릴 케이스 가격대가 생각했던 것 보다 비싸서 제작을 하지 못했습니다. Arduino 관련 제품을 제작하는데 있어서 3d프린터도 많이 쓰지만 조금 규모가 있는 큰 제품들은 아크릴 케이스를 많이 사용합니다. 다음 프로젝트에서 제품 케이스가 사이즈가 클 경우 아크릴 케이스를 사용하는 것을 추천 드립니다.

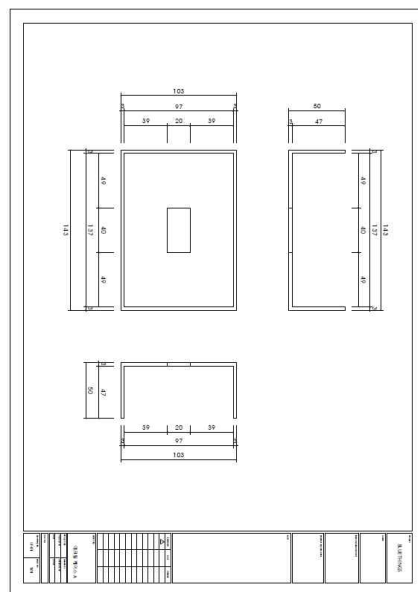
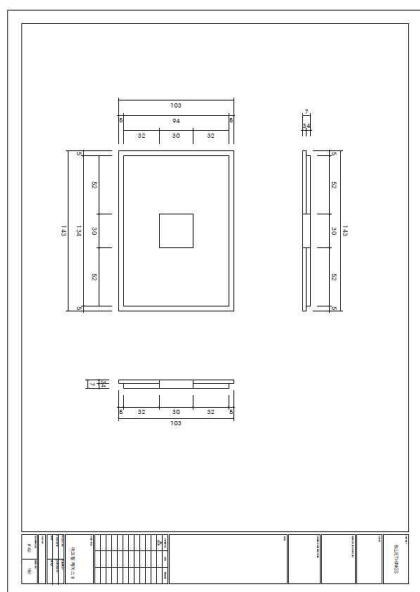


[그림]

(左) 스케치 업 단계

(右) 스케치 업 단계

스케치 업 단계에서 제품에 대한 구상도를 그립니다. 여기서 앞면 뒷면 측면 부분을 다 그려 놓는 것이 좋습니다.



[그림]

(左) 캐드 도면 제작 단계

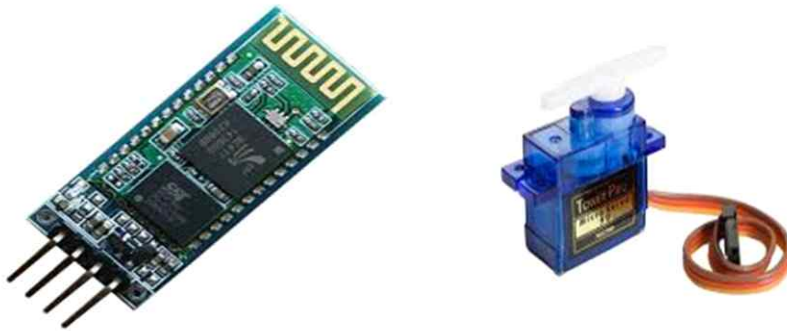
(右) 캐드 도면 제작 단계

캐드에서는 치수가 필요합니다. 제품에 대한 치수를 정확히 재서서 조금 더 넉넉한 치수로 제작

하시기 바랍니다. 생각보다 도면을 그리면 생각해야 되는 부분이 많아지므로 수정사항이 생기더라도 안고 갈 수 있도록 넉넉한 치수를 재서 도면을 제작하셔야 합니다.

굉장히 디테일한 부분은 아크릴로는 만들 수 없으니 사전에 미리 물어보고 제작하는 것을 추천드립니다.

5.2 부품 지연 및 고장



[그림]
(左) 블루투스 모듈
(右) 서보 모터

스마트 스위치 제작 중에 서보 모터의 선이 뒤로 꼬이면서 단선이 된 경우가 있었습니다. 그래서 여분의 부품을 주문하지 못했던 저희는 많이 당황했습니다. 그리고 블루투스 모듈도 고장이 났었는데 고장 원인을 찾지 못했습니다. 부품들이 품질이 좋은 부품들은 아니다 보니 많이 사용할 경우 고장이 잦아지는 상황이 벌어지고는 합니다. 다음에 있을 프로젝트에서는 중요한 부품은 꼭 필요한 수량보다 많이 주문해서 제작해야 된다고 알려드리고 싶습니다.



[그림]
라즈베리파이 모니터

저희 조는 라즈베리파이 모니터를 활용해서 사용자에게 모바일뿐만 아니라 라즈베리파이로도 제어 가능하도록 제작하려 했기에 라즈베리파이가 프로젝트의 핵심이었습니다. 하지만 라즈베리

파이 모니터가 해외 배송인 관계로 늦어진다고는 들었으나 , 12월 초 프로젝트 마무리하기 2주전에 도착했습니다. 그리고 받아보니 라즈베리파이 모니터는 고장이 나 있었습니다. 모니터가 오면 소프트웨어 호환 작업을 하고 제어를 만들자 생각했던 저희의 계획은 많이 틀어졌습니다. 다행히 유상신 교수님께서 모니터를 빌려주셔서 프로젝트를 완성 할 수 있었습니다. 다음에 있을 프로젝트에서는 해외배송은 자제해서 시켰으면 합니다. 해외 배송은 거의 프로젝트 마무리하기 바로 전에 도착하기 때문에 부품 선정에 있어서 참고해서 시켜야 할 것 같습니다.

5.3 서버 적용 문제



원래 저희 계획은 라즈베리파이 제어 부분에 서버를 적용해서 블루투스 통신의 단점을 보완시키려는 생각을 하고 있었지만 , 하드웨어 적인 부분에서도 시간이 오래 걸렸고 부품 문제도 있으면서 계획에 차질이 생기면서 서버는 적용하지 못했습니다. 만약에 서버를 적용했다면 위의 그림처럼 예약기능을 쓰실 수 있었습니다. 퇴근하기 몇 분전에 집 전등을 켜는 스마트 스위치, 출근하는 시간에 맞춰서 꺼지는 스마트 콘센트 등 예약 기능을 구현 할 수 있었습니다. 두 번째로 원격 통신 블루투스는 단거리 통신이지만 서버를 활용한다면, 와이파이통신처럼 원격으로 어디서든 인터넷만 된다면 작동이 될 수 있게 구현할 수도 있었을 겁니다. 라즈베리파이는 블루투스 연결이 되어 있는 상황에서 서버를 통한 라즈베리파이 제어를 통해서 원격으로 스마트 제품들을 제어할 수 있었을 겁니다. 세 번째로 간편한 사용 서버로 통신하게 되면 지금 모바일로 제어하는 사용보다 더 간편하게 사용할 수 있습니다. 모바일로는 블루투스 통신을 할 경우 1:1 관계라 여러 제품을 동시에 사용할 수 없습니다. 하지만 모바일로도 서버제어를 통해서 라즈베리파이의 동시 제어를 할 수 있게 만들면 다른 제품을 사용할 때 마다 블루투스 통신을 다시 할 필요가 없어집니다. 마지막으로 동시 제어 블루투스 통신은 한 기기에 하나씩 연결되기에 한 제품은 한 사람밖에 제어를 하지 못합니다. 그렇지만 서버를 이용하면 여러 핸드폰으로 동시에 한 제품을 제어 할 수 있습니다. 이렇게 되면 훨씬 더 사용하기 좋은 스마트 제품이 될 수 있지 않을까 생각합니다.

6. 참고자료 출처

- Smart Device(switch, plug)

> Smart Switch

http://blog.naver.com/cube_h2/220599575001 - 스마트 스위치 관련 자료

<https://www.youtube.com/watch?v=PHPU-xoFYgA> - 스마트 스위치 간단한 동영상

> Smart Plug

https://ko.wikipedia.org/wiki/%EB%B0%98%EB%8F%84%EC%B2%B4_%EB%A6%B4%EB%A0%88%EC%9D%B4 - 반도체 무접점 릴레이 개념

<http://deneb21.tistory.com/353>

<http://herbram.tistory.com/39> - 스마트 콘센서 구조 관련 자료

- Mobile Application

<https://onedrive.live.com/?authkey=%21ABTEmxiOSgU9qh0&id=C704E2ECDA408483%21243558&cid=C704E2ECDA408483> - 이승진 교수님 Android 강의 자료

<http://blog.naver.com/PostView.nhn?blogId=2hyoin&logNo=220391696574>

<http://dsnigh.tistory.com/44?category=48524>

모바일 어플리케이션을 통하여 Bluetooth 통신을 할 수 있는 어플리케이션 제작

- Raspberry PI 기초 공부

http://makeshare.org/bbs/board.php?bo_table=raspberrypi&wr_id=9

(위 사이트 내부에 있는 Raspberry PI 관련 사례 사이트를 통하여 기초 지식을 습득하였음.)

- Raspberry PI Web UI

<http://www.data.go.kr> - 공공데이터 API 제공 홈페이지

<http://api.bus.go.kr> - 서울특별시 교통정보과(실시간 버스 공공데이터 API 제공)

<http://blog.drunkhacker.me/?p=310> - 공공데이터 추출 사례

- Raspberry PI Connector UI

<http://pinkwink.kr/566>

python 2.7를 활용하여 Bluetooth 데이터 통신하기

[어서와 파이썬은 처음이지?] 천인국 저 Infinite Books 출판 - Tkinter를 활용하여 GUI 기능 구현

할 때 참고하였음.