

# Operation Manual

(Team: BlueThings)

팀원: 장승훈

팀원: 강인성

## - 목 차 -

### 1. BlueThings 제품 설명

1.1 Smart Switch 사용법

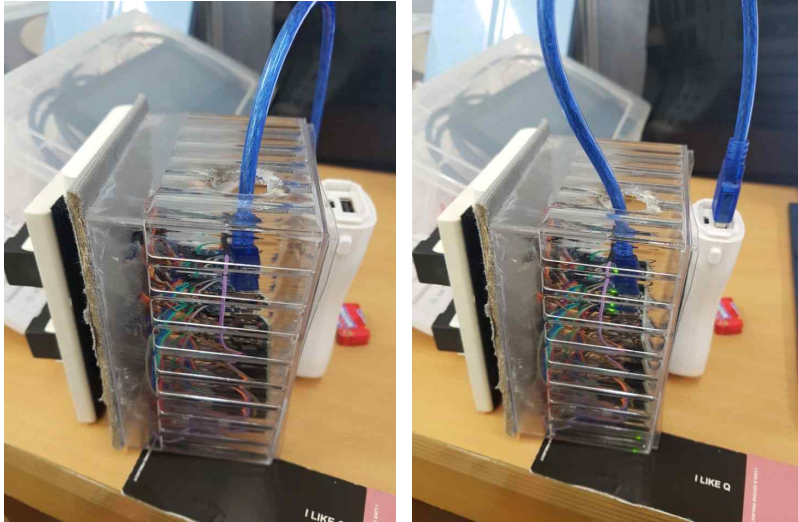
1.2 Smart Plug 사용법

1.3 Android Mobile App 사용법

1.4 Raspberry PI Web Application 사용법

1.5 Raspberry PI Connector 사용법

## Smart Switch 사용법



[그림]

(左) 스마트 스위치 작동 이전

(右) 스마트 스위치 작동 이후

사용법: 파란색 usb 케이블을 뒤에 부착된 보조배터리에 연결하시고 모바일 앱과 라즈베리파이 모니터를 활용해 사용하시면 됩니다. 자세한 방법은 밑에 제어방법을 참고하시기 바랍니다.

(작동이 되면 아두이노에 불이 들어옵니다.)

제어 방법: 스마트 스위치를 사용하려면 Android Mobile App BlueThings를 다운로드 한 후 Bluetooth 연결을 통한 제어 그리고 Raspberry Pi Monitor를 통해 제어가 가능합니다.

(스마트 스위치 Bluetooth 이름은 SAMUL입니다.)

(참고) 보조 배터리는 프로젝트에 포함이 되어 있지 않습니다. 보조 배터리는 별도로 준비하셔서 작동하시길 바랍니다.

## Smart Plug 사용법



[그림]

(左) 스마트 콘센서 작동 이전  
(右) 스마트 콘센서 작동 이후

사용법: 파란색 usb 케이블을 보조배터리에 연결하시고 모바일 앱과 라즈베리파이 모니터를 활용해 사용하시면 됩니다. 자세한 방법은 밑에 제어방법을 참고하시기 바랍니다.

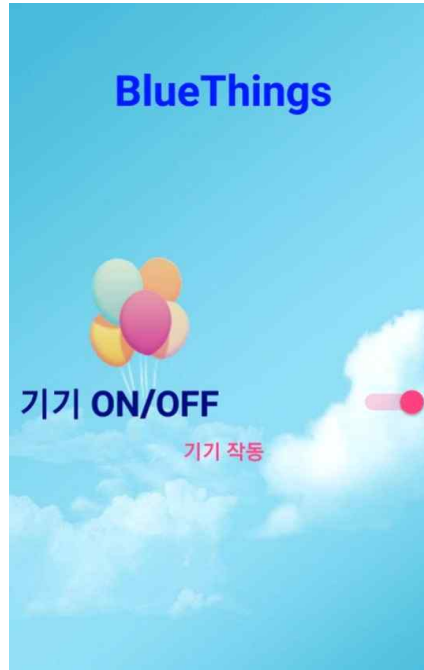
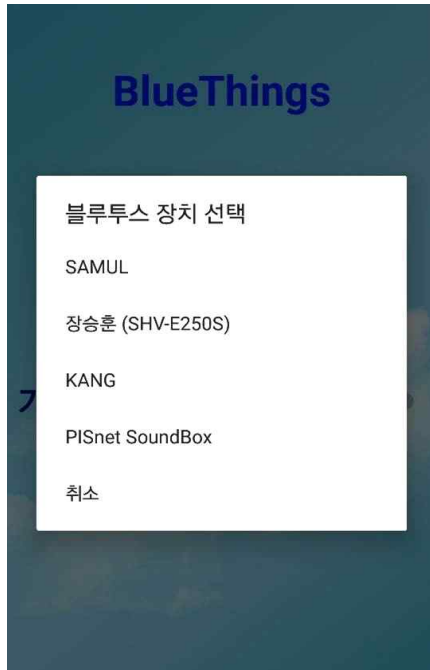
(작동이 되면 아두이노와 파란 led에 불이 들어옵니다.)

제어 방법: 스마트 콘센서는 2구 콘센서에 대해서 통합 관리를 할 수 있으며 옆에 부착되어 있는 터치 센서로도 사용이 가능하며, Bluethings Mobile Application으로 조작이 가능하며, Raspberry Pi Monitor에서도 제어가 가능합니다.

(스마트 콘센트 BlueTooth 이름은 HC-06입니다.)

(참고) 보조 배터리는 프로젝트에 포함이 되어 있지 않습니다. 보조 배터리는 별도로 준비하셔서 작동하시길 바랍니다.

## Android Mobile App 사용법



[그림]

(左) 모바일 어플리케이션 실행  
작동 이전(Device 선택)

(右) 모바일 어플리케이션 실행  
작동 이후

(기기 작동, 종료 제어)

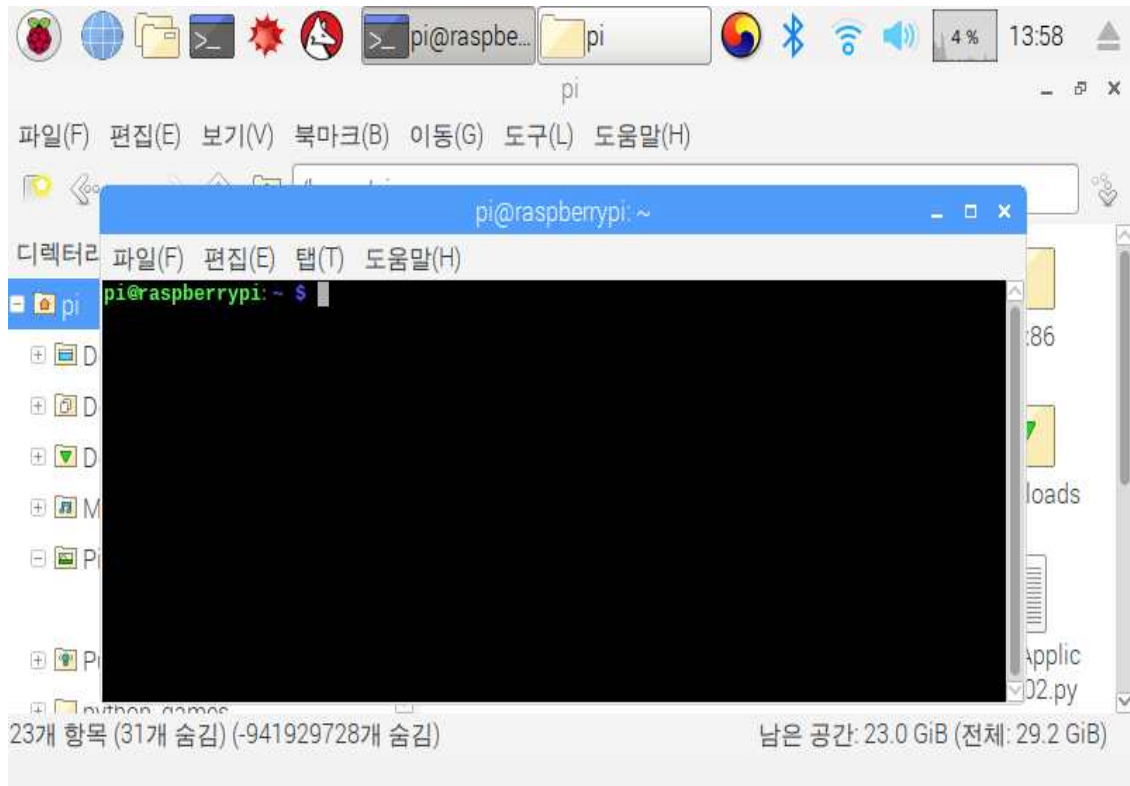
사용법: 안드로이드 apk를 다운로드 후 모바일에서 apk를 설치합니다. 설치 후 myBlueThings라는 app을 실행합니다. 실행 후 Bluetooth Device 목록에서 사용하고 싶은 제품을 선택하시면, 제어 화면이 실행됩니다. 제어화면에서는 스위치 부분으로 기기 on/off를 하실 수 있고 밑에 빨간 글씨로 기기 작동 여부가 확인됩니다.

(스마트 스위치 Bluetooth 이름은 SAMUL입니다.)

(스마트 콘센트 Bluetooth 이름은 HC-06입니다.)

# Raspberry PI Web Application 사용법

-> Apache Tomcat ver 8.0 연동하기



초기에 라즈베리 파이를 구동을 할 때 tomcat8 라이브러리를 설치할 의무가 있습니다.

그래서 위와 같이 콘솔 창을 통해 아래와 같이 설정을 해두고 시작을 합니다.

- \$ sudo service tomcat8 start

여기서 tomcat8 라이브러리가 설치가 안 된 경우에는 아래와 같은 명령어를 통하여 적당한 Wi-Fi 환경 내부에서 설치를 해줘야 한다.(개인용 Pocket Wi-Fi로 하면 데이터가 많이 나오니 웬만하면 집에 설치된 Wi-Fi 환경 내에서 설정을 하는 것이 좋습니다.)

- \$ sudo apt-get update

- \$ sudo apt-get install tomcat8

tomcat8을 설치 완료하였으면

/var/lib/tomcat8/webapps 파일 경로에 tomcat과 관련된 Web Application 파일들이 모여 있는데 BlueThings.war 파일을 그 쪽에 넣고 난 후에 sudo service tomcat8 start 명령어를 통해서 풀고 난 후에 모바일 웹 사이트를 통하여 <http://localhost:8080/BlueThings/> 사이트로 접속을 하면 됩니다.

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
일반	8000	강남역	38분44초[4번째 전]	고속터미널	출발대기	
일반	8009	북촌마을아파트	5분7초[1번째 전]	북영역	23분40초[5번째 전]	북장대교(가상)
일반	8500	더케이호텔서울	147분9초[55번째 전]	신사역신영역	출발대기	
간선	144	교대	2분13초[1번째 전]	북영역	8분56초[2번째 전]	신사역북영역
간선	145	강남역	3분15초[1번째 전]	북영역	10분43초[3번째 전]	신사역거리.가북수립
간선	400	영등포	18분3초[3번째 전]	신정동대명역(한빛오거리)	28분48초[8번째 전]	이태원로.보광동입구
간선	402	장지동영차고지	3분55초[1번째 전]	북영역	13분3초[2번째 전]	신사역북영역
간선	407	북파동영차고지	9분25초[2번째 전]	신사역북영역	17분55초[3번째 전]	신정동대명역(한빛오거리)
간선	441	북영동영차고지	3분[1번째 전]	북영역	9분21초[3번째 전]	신사역거리.가북수립
간선	470	상암차고지	2분7초[1번째 전]	북영역	10분40초[2번째 전]	신사역북영역
간선	471	세곡동사거리	3분53초[1번째 전]	북영역	15분2초[2번째 전]	신사역북영역
간선	N13	북파동영차고지	출발대기		출발대기	
간선	N32	신정동대명역(한빛오거리)	출발대기		출발대기	

### \*곧 도착 예정 버스\*

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

검색할 정류장 번호 22011 검색할 정류장 장소 강남역

하이퍼링크를 통해 들어가면 위와 같은 페이지가 작동을 합니다. 여기서 기본 값으로 정류장 번호는 22011, 장소 이름은 강남역으로 설정을 해 뒀는데 이 공공데이터가 제공을 하는 지역 범위가 서울 지역으로 서울 지역 이외의 결과는 안 나옵니다. 차후에 타 지역 버스 정보 시스템을 받아오면 이용을 할 수 있도록 마련을 따로 하는 것이 좋습니다.

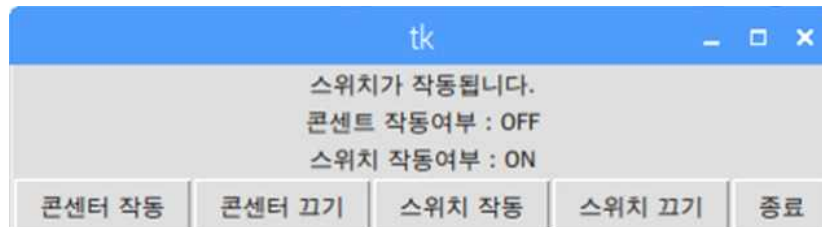
## Raspberry PI Connector 사용법

Connector를 사용하기 이전에 앞서 콘솔 창에 저희가 만들었을 때 쓴 맥 번호를 통하여 페어링 작업을 해주고 이용을 하셔야 됩니다. 페어링은 방금 콘솔 창에 이와 같이 코드를 작성을 하셔서 시작을 해주시길 바랍니다.

```
- $ sudo rfcomm bind rfcomm1 98:D3:33:80:B5:6B
```

```
- $ sudo rfcomm bind rfcomm2 98:D3:31:FB:88:82
```

콘솔 창에서 위와 같은 창을 통하여 rfcomm1, rfcomm2에 대한 블루투스 통신을 적용시켜야 이용이 가능합니다. 이 작업은 Python에서 어떤 Device 내부에 적용을 할지에 대한 설정 파일을 불러오기 위해 생성하는 것으로 보면 되겠습니다. 이를 적용하고 Python Shell를 통해 Module를 실행하면 아래와 같은 화면이 나오게 됩니다. 여기서 실행해야 하는 Python 파일은 MainApplication03.py 파일입니다.[단축키 F5를 통해 실행하면 됩니다.]



여기서 콘센터 작동, 끄기, 스위치 작동, 끄기 4가지를 통해서 각 Smart Device들에 대해서 제어를 할 수 있는데 여기서 라즈베리 파이 모니터는 터치 기능을 제공을 하기 때문에 각 버튼을 직접 클릭해서 이용을 하면 다 관계의 Smart Device를 작동할 수 있습니다. 단, 스마트 콘센서, 스마트 스위치 두 Device의 작동을 보조 배터리를 통해서 연결을 해주고 난 후에 실행을 해야 제어가 가능합니다. 하나만 연결 되어 있는 경우에는 Python 콘솔 내부에서 에러가 나오게 되니 참고하면 되겠습니다.