

# 인공지능 스피커 잘 활용하기

홈 오토메이션 강좌 I

황건순  
2017.08.26

# 홈오토메이션 강좌I

## 개요

인공지능 스피커 대중화

주요 기술의 배경

2017년 하반기 다수의  
유사제품의 출현 예정

소비자로서의 선택

## 연동 방식

3rd 파티 연동의 구조

아마존 에코의 기본 동작

Skill 연동 방식과 원격  
네트워크의 연동 동작

로컬 네트워크에서의

WeMo 연동 방식

보안에 따른 인지와 선택

## WeMo 프로토콜

경제적이면서 효과적인

홈 오토메이션 구현

WeMo 프로토콜을 어떻게  
어디에 적용할 것인가

Unix 계열을 이용한 방법

ESP8266 (NodeMCU)를  
이용한 방법

# 인공지능 스피커

정의와 기술

인공지능 스피커는

- AI Speaker
- Smart Speaker
- Voice Assistant

등의 이름으로 불리우고 있으며,

기본적으로

- 음성인식
- 음성합성
- 챗봇

등의 요소가 포함된 것으로

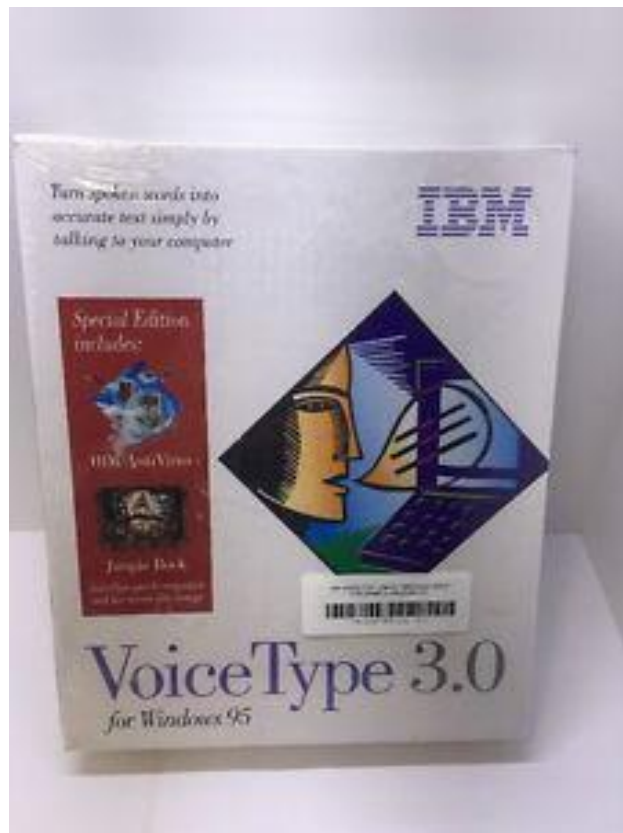
최근 비약적으로 발전한 인공지능  
기술의 산물이라고 할 수 있다.

# 과거의 음성인식 제품 중에서 IBM VoiceType 3.0

---

음성인식은 꽤 오래전부터 다양한 시도가 있었고 화자종속, 화자독립이란 분류로 기술을 크게 나누기도 했었다.

1996년 IBM이 VoiceType이란 제품으로 윈도우와 OS/2에서 일반사용자가 연속적으로 하는 말을 알아들을 수 있는 제품을 출시했다. 문서작성이나 넷스케이프 브라우저등을 기본적으로 지원할 수 있었고 유명전시회등에서 자주 볼 수 있었는데 발음을 무척 잘해야했고 신기하기는 했으나 현실에서는 여러가지로 쓰기는 어려웠다.



# 과거의 음성합성 제품 중에서 NEC PC-6001 mkII

— — —

음성합성 또한 다양한 학문적인 연구가 있었고 상용제품 중에 인상적이었던 것은 1983년에 일본에서 출시된 NEC의 PC6001mkII였다. 자주 쓰는 기본적인 녹음된 음성외에도 코드를 이용해서 음성을 합성할 수 있었다. 품질이 뛰어나지는 않았지만 일반인이 쉽게 접근해볼 수 있는 음성합성 기술이었다.

TALK "F64" [음성](#)

TALK "M2 KOXNICIWA." [음성](#)



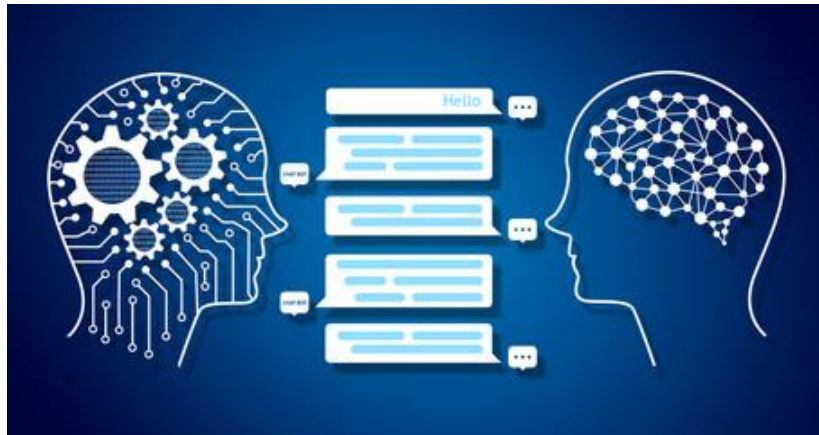
# 현재 진행형인 Chatbot

-----

여러가지 인공지능 기술을 기반으로 한 것 중에 대중화가 빨리 진행되고 있는 챗봇은 아직까지는 일반인이 생각하는 자유로운 대화 수준까지 만들어지고 있지는 않지만, 제한적인 환경안에서 잘 응용하고 있는 제품이 아마존 에코라고 할 수 있다.

아마존 에코의 Skill을 작성할 때 보면 대화에서 나올 수 있는 문장 형태를 미리 정하고 대화가 진행되는 것을 볼 수 있다.

최근 관련 기술이 비약적으로 발전하고 있어서 자유도가 높은 챗봇이 멀지 않은 시간이 나올 것으로 예상된다.



# 인공지능 스피커 시장을 연 아마존 에코 시리즈

아마존 에코 제품군은 2014년 후반기에 출시되어 현재까지 천만대 이상 팔린 제품으로 하드웨어가 많은 대중에 보급된 것은 물론 다양한 부가적인 기능을 확장할 수 있는 Skill 이 만개이상이 등록될 정도로 인공지능 스피커 분야에서 큰 위치를 자리하며 꾸준히 성장하고 있다.

서비스의 중심이 되는 Alexa Voice Service를 API 형태로 오픈되어 최근 3사 하드웨어들과 서비스를 포함시켜 세력을 더 크게 확장하고 있는 중이다.



# 2017년 출시예정인 인공지능 스피커까지

----

2017년 하반기에는 여러나라에서 다양한 인공지능 스피커의 출시가 예정되어 있어 기존 출시 제품들과의 상당한 경쟁이 예고되고 있다.

- Amazon Echo Series
- Google Home
- Apple HomePod
- Microsoft Invoke
- SK NUGU
- KT GiGA Genie
- Line Wave, Champ, Face
- Kakao Kakaomini

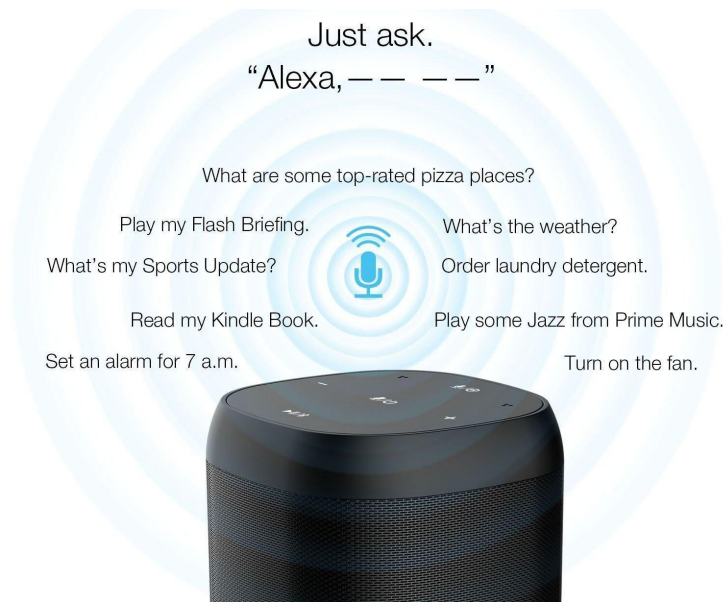




# 할 수 있는 것, 기대되는 것 그리고, 홈오토메이션

우리나라 제품 광고에서 나오는 것은 날씨, 음악,  
놀이줘?

인공지능 스피커만으로 처음에 쉽게 할 수 있는  
것이 날씨와 음악이고 몇가지 더 하면 알람, 타이머  
정도가 일반적으로 시도하는 것들이다. 익숙해지고  
또 미국같은 환경이라면 교통정보나 다양한 것을  
기대할 수 있지만 일반적인 소비자로 현실에서 가장  
유용한 것은 홈 오토메이션이 아닌가 생각된다.  
움직일 필요도 없고 리모콘이나 스마트폰을 찾을  
필요도 없이 말 하는것만으로 가전제품을 제어할 수  
있는 것은 편의성, 연속성면에서 인공지능 스피커가  
주는 새로운 기술적인 혜택이라고 말할 수 있다.



# 지원되는 제품들

## 보안과 경제성의 문제

아마존 에코의 경우 연동되는 제품이 [Alexa Smart Home](#)이라는 별도의 페이지가 존재할 정도로 많이 있다. 하지만, 문제는 대부분 생각보다 가격이 높은 것들이 많아 구매하기가 쉽지 않은 편이다. (유통 마진이 높은 한국과 일본의 경우는 중국에서 \$12에 구매하는 것이 10만원선, 6천엔을 넘기도 한다.)

게다가 구매하더라도 어떻게 동작하는지 보안에서 고려할 것이 무엇인지 잘 모르기 때문에 막연한 불안감이 생기는 것 또한 현실이다.

보안에서 내 선택이 무엇인지 이해하고 또 경제적 장애를 넘을 대안도 마련해서 더 많은 사람이 신기술의 혜택을 누릴 수 있는 게 바람직하다고 생각한다.



# 아마존 에코의 동작과 연동

네트워크 구성

아마존 에코는 호출시 사용하는 **Wake word**만 장치 자체에서 인식하고 이 후 모든 음성은 네트워크로 전송하고 처리 후 결과음성 또한 네트워크로 전송받는다.

외부 장치와 연동 될때는 별도의 외부 서비스 장비와 아마존 서비스가 연동되는 외부연동 형태와 아마존 에코와 장치가 직접 연동되는 내부 연동으로 나눌 수 있다.

# 아마존 에코는 간단하게 무엇인가

아마존 에코는  
**AVS**에 마이크와 스피커를  
인터넷을 통해 연결해주는 것이라고  
간단하게 설명할 수 있다.



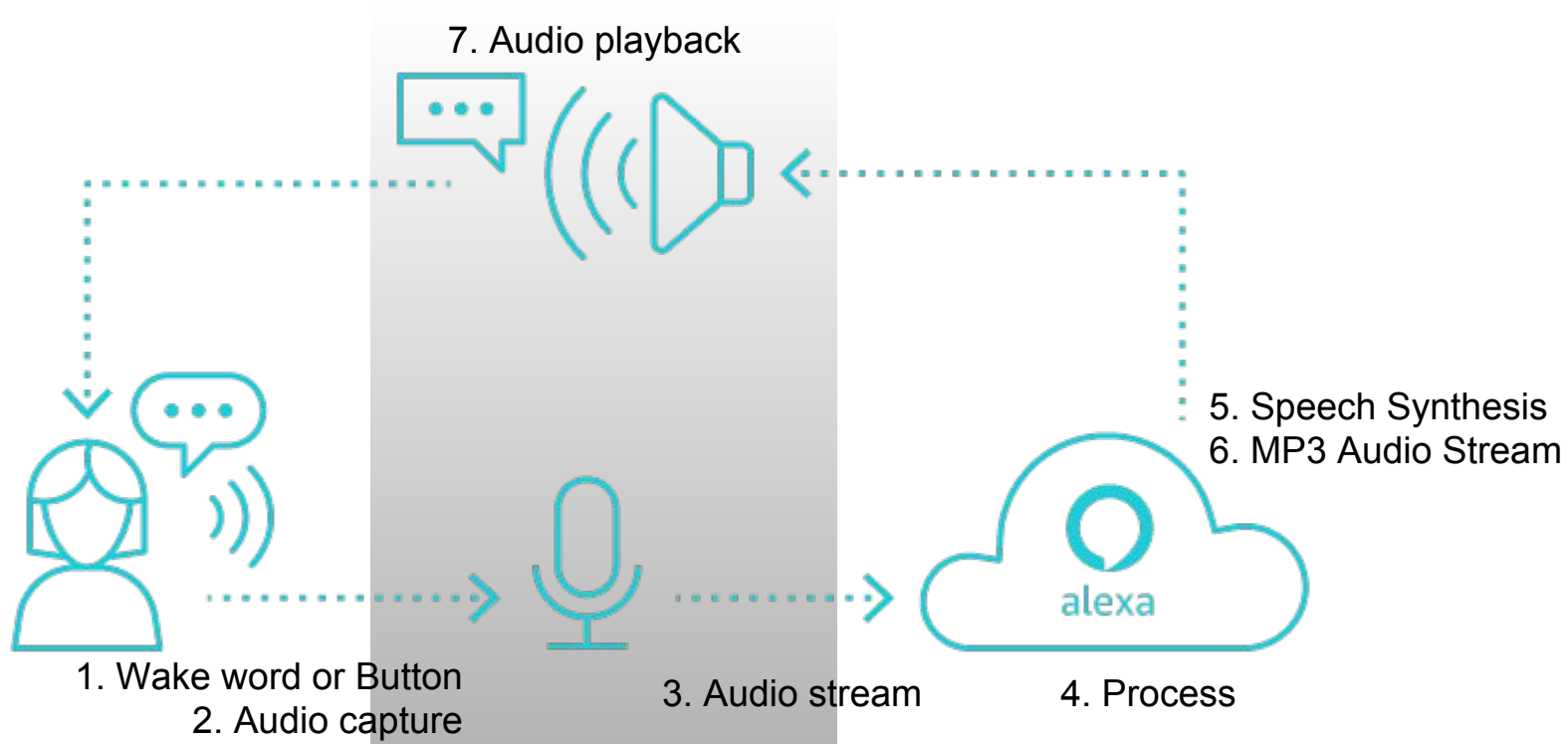
Alexa Voice Service (AVS)

내부 네트워크

외부 네트워크

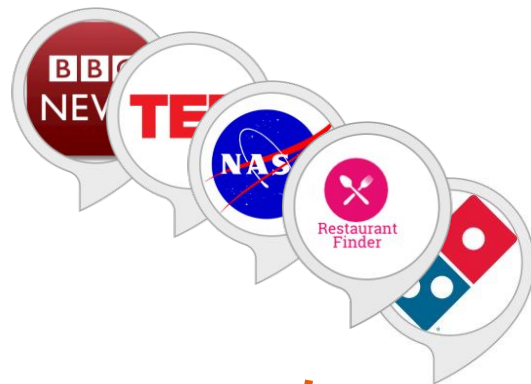


# 아마존 에코의 기본 동작 흐름



# Skill의 기본 동작 흐름

Skill은 스마트폰의 앱처럼 아마존 에코에 다양한 부가기능을 더 해주고 있고  
아마존 서비스와 Skill서비스는 https를 통해 연동된다.

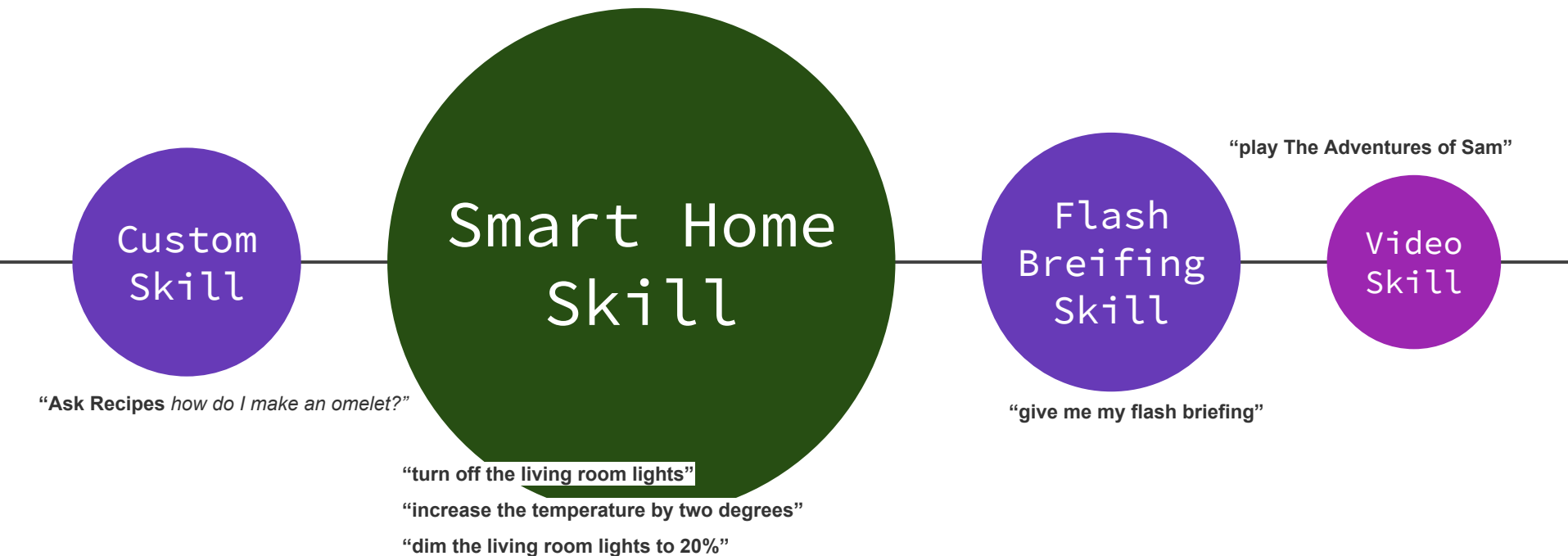


amazon alexa

Alexa Voice Service (AVS)

참고) [Alexa Skills](#)

# Skill의 종류



# Skill과 홈오토메이션

홈오토메이션에서 주로 사용되는 Skill은 Custom Skill과 Smart Home Skill이다.

Custom Skill은 대화의 진행과정과 동작을 스스로 프로그래밍해서 사용할 수 있고 목적하는 바를 정확하게 구성할 수 있다. 하지만, 여러가지 환경을 다 준비하기는 어려움이 있어 제3의 솔루션과 연동하는 것이 편리하다.

Smart Home Skill은 아마존 제품군중 Lambda를 기반으로 한 분산처리 IoT 인프라를 사용해서 운용된다. 도식으로 볼 때는 간단하지만 Amazon Lambda, AWS IoT 등 아마존 서비스에 익숙해야 하고 MQTT 등 부가적인 지식도 필요하다.



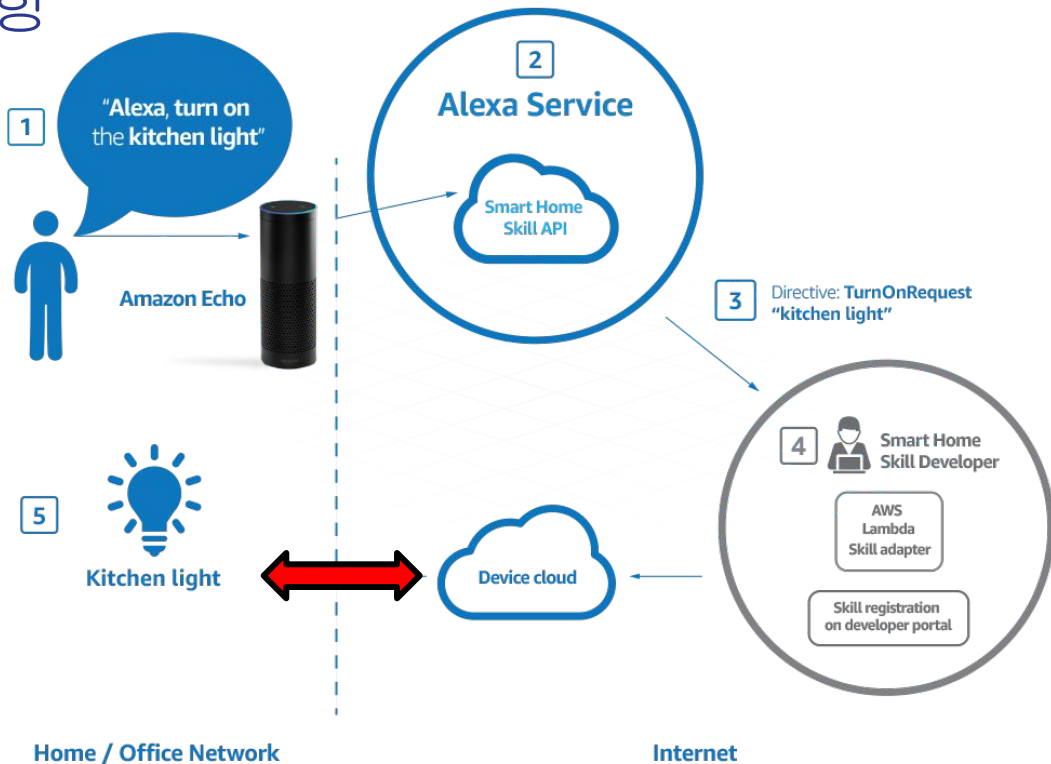


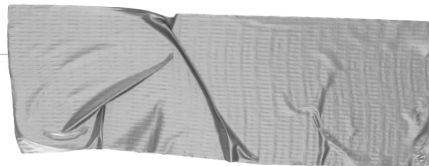
# 아마존이 설명하는 Smart Home Skill과 고려사항

아마존이 설명하는 Smart Home Skill의 도식에서 보면 1-5 순서에 따라 부엌 전등을 켜는것을 설명하고 있다. 여기에 빠져있는 동작이 하나 있는데 Kitchen light가 Device cloud에 접속되는 과정이다.

다시 말해서 집안의 장치가 외부에 연결되어 제어 명령을 받을 수 있는 준비가 되는 상태가 가장 먼저 일어나는 것으로 이 부분에서 우리는 보안적인 측면에서 의문을 가질 필요가 있다고 생각한다.

이 부분은 다른 IoT 서비스들도 가지고 있는 것으로 오동작,해킹,악의적 목적등 최악의 사태에 대한 상황을 생각해볼 필요가 있다.





## Skill 기반을 사용한다면

Skill 기반으로 홈오토메이션을 구성한다면  
다양한 지식과 환경에 대한 이해와 준비가  
되어야 한다.

- **다양한 환경에 대한 지식과 준비**  
Amazon Services, AVS, Certificate, Smart  
Hub, Smart devices, ...
- **원격지에서의 연동**  
아마존 에코를 위한 연동 외에  
부가적인 서비스가 외부에서  
연동되어 제어를 하게 되므로  
보안적인 이해와 선택이 필요하다.

## Belkin, WeMo 내부 네트워크 연동

Belkin이 만든 홈오토메이션 제품군의 이름이 **WeMo**고 제품들을 제어하기 위해서 사용하는 프로토콜은 장치가 속해있는 네트워크 안에서만 동작하는  
즉, 내부 네트워크 연동을 위해 사용된다.

아마존 에코는 2015년 업데이트를 통해 **Belkin Wemo**와 **Philips Hue**를 지원하기 시작했다.  
**Skill**과 달리 로컬네트워크에서 운영되기 때문에 장치를 찾는 동작이 있고 장치가 찾아지면 해당 장치의 정보를 저장해두고 해당하는 이름으로 동작명령이 되면 아마존 에코에서 직접 장치에 접속해 제어를 하게 된다.

(참고로 **Philips Hue**도 유사하나 조금 더 복잡한 구조를 가지고 있다.)



UPnP  
SOAP



## Belkin, Philips 제품들

벨킨 WeMo와 필립스 Hue는 아마존 에코뿐 아니라 Google Home도 같이 지원할 정도로 미국의 업계에서 자리매김하고 있다.

WeMo는 주로 스위치 기능의 제품군, 필립스는 조명 계통의 제품군이 대표적이다.

세일을 하는 특별한 상황을 제외하고는 최소 \$40정도는 지출을 해야 하고 국내에서는 미국의 최소 2배정도 가격을 예상해야 한다. (브리지가 포함된 A19 스타터 키가 국내에서는 11만원정도) 아마존 에코닷2의 가격이 \$50이 안되는 상황이니 가격은 아직까지 만족스럽지 않은 상황이다.

### WeMo - Home Automation

### Philips Hue - Products





## WeMo 기반을 사용한다면

WeMo 기반으로 홈오토메이션을 구성한다면  
지원 제품만 구매하는 것으로 바로 사용이  
가능하다.

### → 간단한 연동

지원 제품을 구매하거나 만드는 것으로  
간단하게 사용 가능

### → 단순 제어의 한계

대부분 켜고 끄는 단순 기능만을  
지원하고 있다. 필립스 제품은 조금  
더 세밀한 제어가 되지만 정의된  
제어는 그다지 많지 않다.

# 왜 무엇을 어떻게 하고 싶은가?



## 기존 IoT는?

IoT 붐으로 스마트폰에  
연동되는 수많은 솔루션이  
나왔다.

하지만 집에서 간단한 걸 하려고  
스마트폰을 켜고 앱을  
실행하는것은 그다지 하고 싶지  
않다.



## 정말 쓸만한가?

우리는 이미 누구 이름을 불러서  
원하는 것을 얘기하는데  
익숙하다.

리모콘 어디있지라고 얘기할  
필요도 생각할 필요도 없다.  
그냥 하던거 하면서 말만 하면  
끝



## 사서 연결하면 끝?

지원되는 제품은 구매하면  
간단히 연결해서 쓸 수 있다.  
그런데 기능대비 너무 비싼  
경우가 대부분이다.



## 그럼 어떻게?

조금 관심을 기울이고 수고를  
들인다면 경제적으로 우리  
생활이 새로운 기술로 좀 더  
편리해질 수 있고 나아가  
여러사람이 좀 더 혜택을 누릴  
수 있다.

# WeMo 프로토콜

구현과 활용

Belkin의 WeMo 프로토콜은 최소의 컴퓨팅 환경에서도 구현이 가능하여 IoT에 특화된 ESP8266과 같은 칩에서 구현 및 활용이 가능하다.

지원 제품을 구매해서 쓸 수도 있지만 관심만 있다면 직접 원하는 장치를 만들어 경제적으로 홈오토메이션을 구축할 수 있다.

# WeMo와 아마존 에코의 연동 흐름

WeMo와의 연동 동작은 크게 두가지로 나눌 수 있다.

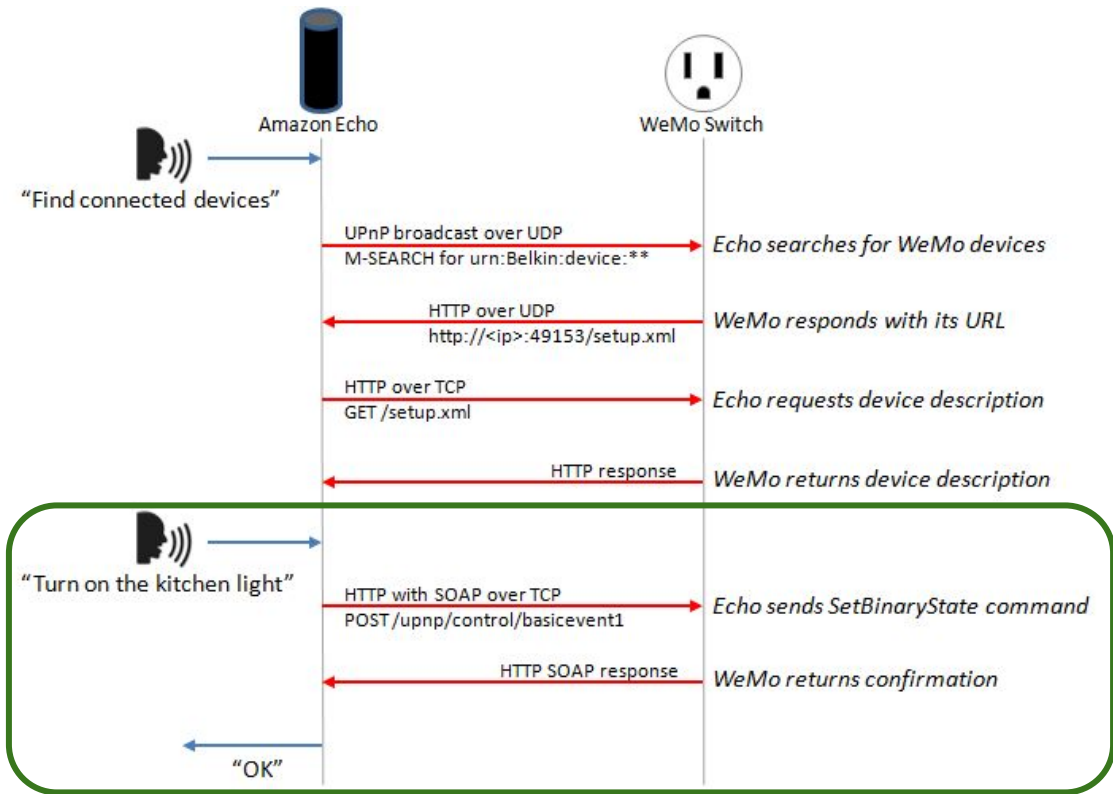
## - 장치 찾기

아마존 에코에 “Discover devices” 명령을 내리면 아마존 에코는 로컬네트워크에 브로드캐스트로 UPnP 검색 요청을 보내고 WeMo 장비는 그에 대한 응답으로 장치의 정보를 담고 있는 URL을 전송한다. 아마존 에코는 해당 정보를 해석해서 장비 정보를 저장한다.

## - 장치 사용

장치를 사용하는 명령어가 탐지되면 http 프로토콜로 제어명령을 장치에 보낸다.

(현재 켜고 끄는 동작만 되는 basicevent1만 지원된다.)





# 구현 환경에 따른 특성

## ESP 계열

ESP8266과 ESP32가 있고 ESP8266을 사용하기 쉽게 만든 NodeMCU가 많이 알려져있다. 최근 중국에서 구매를 하면 \$3내외로 저렴하다. WiFi내장에 I/O도 충분하다. Arduino, NodeJS, MOS 등 다양한 환경을 지원한다.

## Raspberry PI 계열

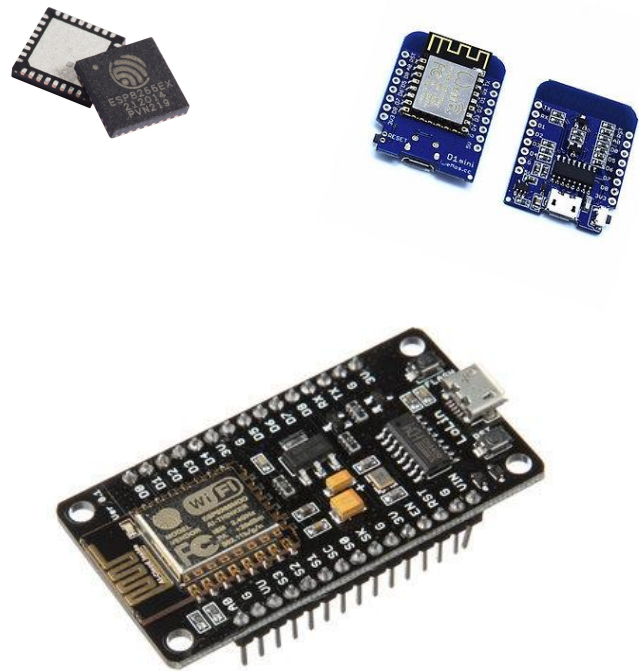
Linux 기반이라 다양하고 복합적인 구성이 가능하다. Python 등 고급언어에서 쉽게 연동이 가능하고 멀티태스킹 환경에서 다기능으로 사용할 수 있다. 직접 단일 장치를 붙이기에는 가격 부담과 부피가 있다.

## 프로토콜 브리지

아마존 에코와 제어하고자 하는 장치의 중간에서 브리지 (또는 허브) 역할로 구성한다. 타사 IoT 장치라던지 커스텀 장치 등 프로토콜이나 제어방식이 틀린 것을 중계해주는 역할을 한다. 저렴한 상용 IoT 장치 활용이 가능하다.

# ESP8266 그리고 NodeMCU

오픈소스 하드웨어 플랫폼으로 유명한 **Arduino** 시리즈가 있었지만 IoT에 대한 열풍을 배경으로 2014년 WiFi가 탑재되어 있고 TCP/IP가 가능한 칩으로 **ESP8266**이 발표되었다. 기존 아두이노보다 월등한 성능과 자원을 포함하고 WiFi를 통한 기기종간의 연결이 독보적이였다. 하지만, 전문가가 아니고는 사용하기 번거로워서 저렴한 IoT 기기를 제작하는 업체 위주로 많이 사용되었다. 2015년에 전문가가 아닌 사람들도 편하게 쓸 수 있는 오픈소스 IoT 플랫폼으로 **NodeMCU**가 보급되기 시작했다. USB케이블만 연결하면 바로 쓸 수 있는 편의성에 가격도 저렴하고 많은 사용자를 확보하고 있는 **Arduino IDE**도 지원되는 상황에서 인기를 얻게 되었고 이후 조금 더 적게 만든 **NodeMCU Mini** 제품까지 판매되고 있다.



# NodeMCU와 Arduino IDE

NodeMCU 에서 Arduino IDE를 사용하기 위해서는 기본적으로 2가지 필요하다.

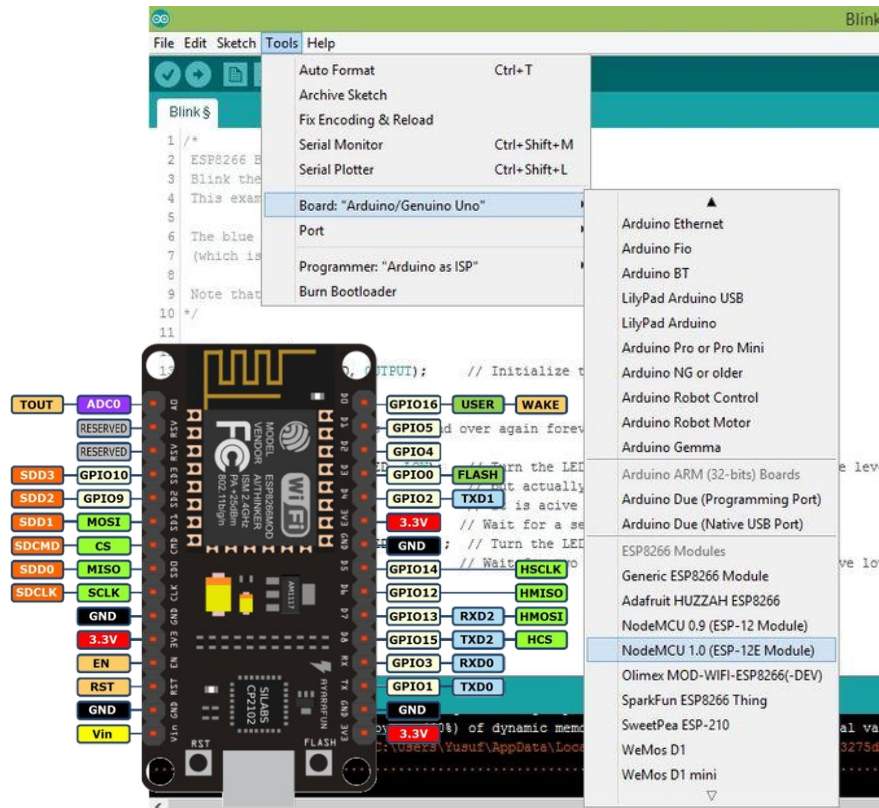
- 시리얼 드라이버

NodeMCU는 USB에 연결하면 시리얼통신이 되도록 만들어져 있다. NodeMCU를 제조하는 회사에 따라 시리얼통신을 지원하는 칩이 달라서 사용하는 OS에 맞는 드라이버를 설치해야 한다.(대부분 중국제품은 CH340)

- 보드 매니저를 통한 설치

Arduino IDE 설정에서 추가 보드 매니저 URL에 [http://arduino.esp8266.com/stable/package\\_esp8266com\\_index.json](http://arduino.esp8266.com/stable/package_esp8266com_index.json)

추가하고 보드 매니저를 메뉴에서 esp8266을 찾아서 설치한다.



## 연관 링크

- [Arduino IDE](#)
- CH340 Driver : [제조사](#) , [nodemcu-devkit](#)
- Esp8266 board URL : [http://arduino.esp8266.com/stable/package\\_esp8266com\\_index.json](http://arduino.esp8266.com/stable/package_esp8266com_index.json)
- [각종 NodeMCU 소개와 비교](#)
- [ESP8266 Arduino IDE](#)
- [Quick Start Nodemcu \(ESP8266\) on Arduino IDE](#)

# Arduino IDE 환경에서 WeMo 프로토콜 사용 준비

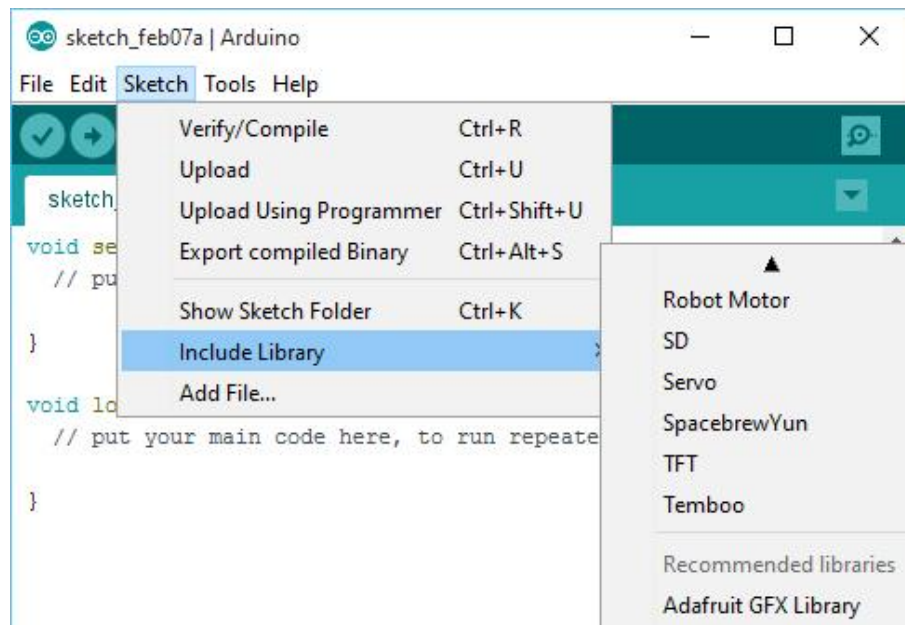
Arduino IDE 에서 WeMo 프로토콜을 사용하려면  
관련한 라이브러리를 설치해야 한다.  
각 라이브러리는 **Arduino Sketch** 디렉토리에 있는  
**libraries** 디렉토리 안에 라이브러리 별로 별도의  
디렉토리로 넣어두면 된다.

[fauxmoESP](#) - WeMo 프로토콜 지원

[ESPAsyncTCP](#)

[ESPAsyncWebServer](#)

[IRRemoteESP](#) - 적외선 리모콘 기능 지원



# fauxmoESP 샘플로 연동 테스트 준비

대부분의 라이브러리들은 샘플코드를 포함하고 있고 **fauxmo**도 마찬가지로 샘플코드가 하나 포함되어 있다.

**credentials.h** 파일에 WiFi 접속을 위한 SSID와 비밀번호를 지정해주면 빌드가 가능하다. 이 예제는 **light one**부터 **light four** 까지 4개의 장치가 있는 것으로 정의가 되어 있고 동작 요청이 올 때 내장되어 있는 **LED 2**로 정의되어 있는

```
// Fauxmo
fauxmo.addDevice("light one");
fauxmo.addDevice("light two");
fauxmo.addDevice("light three");
fauxmo.addDevice("light four");

// fauxmoESP 2.0.0 has changed the callback signature to add the device_id, this WARRANTY
// it's easier to match devices to action without having to compare strings.
fauxmo.onMessage([](unsigned char device_id, const char * device_name, bool state) {
    Serial.printf("[MAIN] Device #%d (%s) state: %s\n", device_id, device_name, state ? "ON" : "OFF");
    digitalWrite(LED, !state);
});
```

fauxmoESP\_Basic | 아두이노 1.6.6

파일 편집 스케치 툴 도움말

새 파일 Ctrl+N  
열기... Ctrl+O  
최근 파일 열기  
스케치북  
예제  
닫기 Ctrl+W  
저장 Ctrl+S  
다른 이름으로 저장... Ctrl+Shift+S  
페이지 설정 Ctrl+Shift+P  
인쇄 Ctrl+P  
환경설정 Ctrl+Comma  
종료 Ctrl+Q

//

void wifiSetup() {

// Set WiFi module to STA mode

WiFi.mode(WIFI\_STA);

// Connect

Firmata  
GSM  
LiquidCrystal  
Robot Control  
Robot Motor  
SD  
Servo  
Stepper  
Temboo  
RETIRED

사용자 지정 라이브러리의 예제

ArduinoOTA  
DNSServer  
EEPROM  
ESP Async WebServer  
ESP AsyncTCP  
ESP8266  
ESP8266AVRISP  
ESP8266HTTPClient  
ESP8266httpUpdate  
ESP8266HTTPUpdateServer  
ESP8266mDNS  
ESP8266SSDP  
ESP8266WebServer  
ESP8266WiFi  
ESP8266WiFiMesh  
Ethernet(esp8266)

FauxmoESP  
Hash  
IRremoteESP8266

SSID().c\_str(), WiFi.localIP().

fauxmoESP\_Basic

# fauxmoESP 샘플로 테스트

“Alexa, discover my devices”

“Starting to discovery. Device discovery can take up to 20 seconds” .....

“Device discovery is complete. I found 4 connected home devices ...”

“Alexa, turn on **light one**”

“Ok”

“Alexa, turn off **light four**”

“Ok”

시리얼 콘솔에서 동작때마다 찍히는 메시지를 보면 `device_id` (장치번호)와 요구동작 (`state`)가 출력된다.

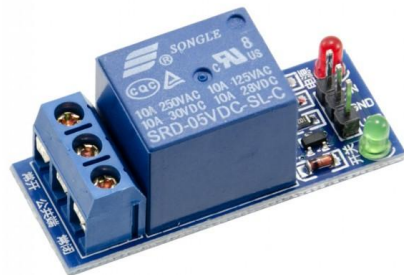


## 릴레이를 붙인다면

예제에서 사용하는 D4에 릴레이를 연결하고  
다시 릴레이에 콘센트를 연결한다면 바로  
가전제품을 켜다 켜다 할 수 있다.

저렴하게 부품을 구한다면 5천원정도와 선  
몇가닥을 연결하는 것으로 가전제품을 음성으로  
제어 할 수 있게 된다.

구형선풍기 하나라도 아마존 에코에 연동하면  
더운 여름에 정말 편리함을 체험 할 수 있다.  
작업을 하면서 덥다고 느껴지면 타이핑 중에  
아마존 에코에 선풍기 켜달라고 하면 바로  
켜지고 또 끄고 싶으면 끄라고 하면 바로 꺼지고  
^^





## IR 송신 장치를 붙인다면

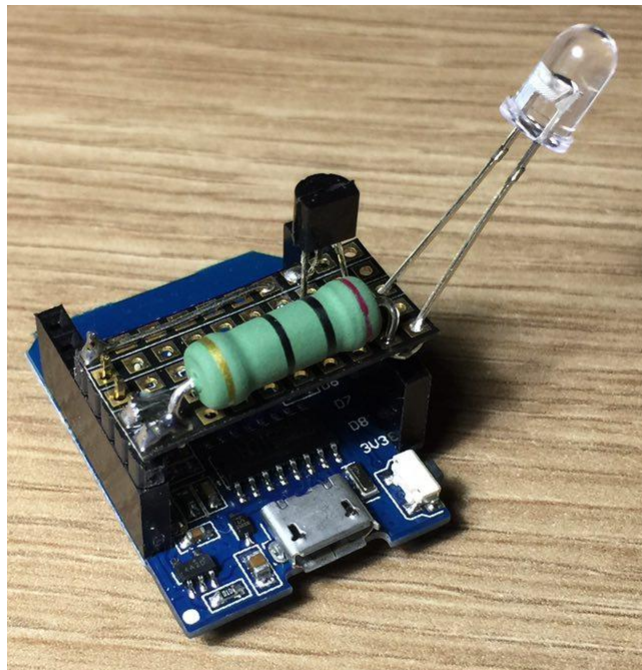
요즘은 리모컨을 사용하는 가전제품이 많다.  
그렇기에 홈오토메이션을 구성하는 과정에서  
리모컨 제어 기능은 꼭 필요한 것 중 하나라고  
말할 수 있다.

IR LED를 단순히 연결하는 것은 어렵지  
않지만 일정 거리이상에서 제어를 할 수  
있으려면 구동전류를 높여주는 약간의 회로  
구성이 필요하다.

또한 제품마다 리모컨 신호가 달라서 최초에는  
신호를 해석할 수 있는 간단한 장치를  
부착하거나 리모컨 신호방식과 코드를 알아야  
한다.

소프트웨어의 경우 **IRRemoteESP**

라이브러리를 사용하면 **2-4줄** 정도 추가 하는  
것만으로 기능을 완성할 수 있다. 가장  
대표적으로는 리모컨 방식 조명,에어컨등을



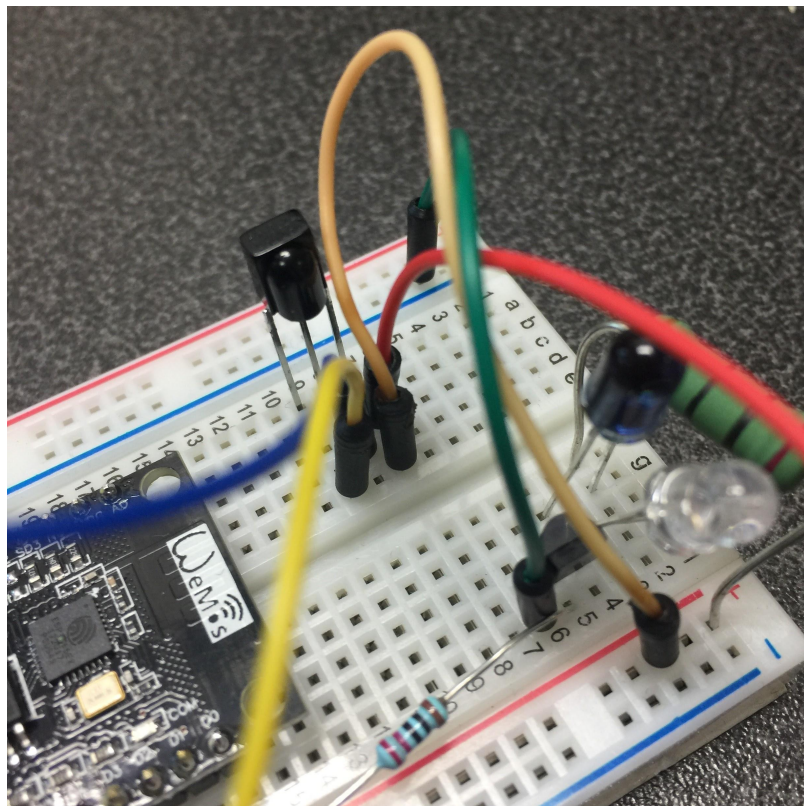
# IR 리모컨 신호 분석

리모컨 신호 분석은 IR 신호를 수신할 수 있는 부품하나만 연결하고 샘플 프로그램 중 Dump 프로그램을 사용하면 아주 쉽게 분석할 수 있다.

LG에어컨의 리모컨으로 끄는 신호를 분석했다면 방식과 코드가 나오고 실제로 프로그램에 다음과 같이 쓰면 같은 신호를 송신하게 된다.

```
irsend.sendLG(0x88C0051,28) ;
```

```
Encoding : LG
Code : 88C0051 (28 bits)
Timing[59]:
+8550, -3900    + 550, -1500    + 500, - 500    + 500, - 500
+ 500, - 500    + 500, -1500    + 500, - 500    + 500, - 500
+ 500, - 500    + 500, -1500    + 500, -1500    + 500, - 500
+ 500, - 500    + 500, - 500    + 500, - 500    + 500, - 500
+ 500, - 500    + 500, - 500    + 500, - 550    + 500, - 500
+ 500, - 500    + 500, - 500    + 500, -1500    + 500, - 500
+ 500, -1500    + 500, - 500    + 500, - 500    + 500, - 500
+ 500, -1500    + 500
uint16_t rawData[59] = {8550,3900, 550,1500, 500,500, 500,500, 500,500
```



## 리모컨 스위치 Case Study

벽에 붙은 전등 스위치가 수동인 경우 전등을 자동화 하는 방법은 크게 2가지로 나뉘볼 수 있다.

먼저, IoT 기능이 포함되어 있는 예를 들어 **WeMo Light Switch** 같은 것이다. 하지만 이 경우 전원 공급을 위해 중성선을 연결해야 하는데 우리나라 전등스위치에는 대부분 중성선이 내려와 있지 않아서 요비선등을 이용해서 선을 따와야 하는 어려움이 있다.

(아마존 \$39.99)

국내 환경에 맞는 것은 리모컨 스위치가 있다. 2만원대부터 제품이 있고 기존 전등 스위치에 연결된 선을 그대로 옮겨 꼽고 전등쪽에 역률개선 콘덴서를 상황에 따라 연결하기만 하면 된다.이후 **NodeMCU**에 **IR**송신기를 연결해서 연동한다. 온라인 구매 후 다음날



# Raspberry Pi 계열

2012년부터 판매되어 초기에는 저렴한 교육용 컴퓨터로 대중에게 알려졌고 이 후 일반인이 접근하기 쉬운 임베디드 리눅스의 대명사로 자리잡았다. 초기 제품들은 불안정하기도 했지만 **Pi 2** 이후 안정화되기 시작해서 현재 **Pi 3, Zero W**등에는 **WiFi**와 **BLE**가 탑재되어 나오고 있다. 리눅스 기반에 스펙도 좋은 편이라 다양한 구성이 가능하다.

**Python**으로 많은 관련 라이브러리들이 있어 간단하게 연동이 가능하고 **Custom Skill**을 만들기에도 크게 어렵지 않다고 할 수 있다.

크기와 비용의 부담을 감안한다면 다기능으로 사용할 수 있다.

유사한 스펙과 가격대에 **ARM**기반의 중국산 제품들도 있다.



## 연관 링크

- [Raspberry Pi Download](#)
- [WiringPi](#)
- [Python Fauxmo](#)
- [Python Flask extension for Alexa skill](#)
- [Emulate Philips Hue Bridge](#)
- [Ngrok Secure tunnel](#)
- [Cheap SSL Certificate](#)

# Python 샘플로 연동 테스트 준비

Python Fauxmo 지원 파일에 있는 `example-minimal.py` 파일을 실행하면 포트 52000에 `device`란 장치명으로 연동이 가능하도록 되어 있다.

초기에 UPnP의 라우팅을 위해서 다음 명령으로 네트워크 라우팅을 준비해야 한다.

```
route add -net 239.0.0.0 netmask 255.0.0.0  
eth0
```

Python 소스 코드에 있는 TRIGGERS에 장치를 더 추가하거나 콜백펑션인 `act`에 원하는 동작을 추가하면 바로 사용이 가능하다.

```
TRIGGERS = {"device": 52000}
```

```
def act(self, client_address, state, name):  
    print "State", state, "on ", name, "from client @", client_address  
    return True
```

```
TRIGGERS = {"aircon": 52010, "light": 52011, "tv": 52012, "set1": 52013}
```

```
def act(self, client_address, state, name):  
    print "State", state, "on ", name, "from client @", client_address  
    if name=="aircon":  
        if state:
```



# 프로토콜 브리지

직접 만들어 구성하는 경우 무척 저렴하지만 케이스 문제같은 외형적인 문제나 완성도면에서의 다른 대안은 필요하다면 **WeMo**가 아닌 다른 **IoT**제품을 선택해서 아마존 에코와 중간에 위치해서 프로토콜을 변환해서 중계해주는 브리지 구성을 생각해볼 수 있다.

중국 제품 중에는 **BroadLink**, **MiPOW**, **Orvibo** 등 세계적으로 많이 알려진 저렴하면서도 우수한 품질의 **IoT**제품들이 많이 있다. (**Aliexpress**로 구매시 가격은 \$10내외부터 구매가 가능하다.) 이런 제품들은 자체적인 스마트폰 앱을 제공하고 있고 프로토콜 또한 고유 방식을 택하고 있다. 그러나, 일부 사용자들이 리버스 엔지니어링을 통해 방식을 분석을 해주거나 라이브러리까지 만든 경우가 있어서 **NodeMCU**나 **Raspberry PI**를 통해서 중계 역할을 하는 브리지를 만들어 연동이 가능하다.



# 프로토콜 브리지 Case Study

Raspberry PI 등 리눅스 환경에서는 BroadLink, Orvibo 등이 Python을 통해 제어가 가능해서 어렵지 않게 연동할 수 있다.

NodeMCU 경우 최근에 만들어서 github에 공개하기 시작했고 지원 장치를 늘여가고 있다.

MiPOW 같은 BLE로 연동되어야 하는 것은 리눅스 계열이 가능하고 NodeMCU 상위 제품인 ESP32에서는 향후 실험해 볼 예정이다.  
대부분 경우 무선으로 연결하게 된다.

BroadLink MP1 - 개별 포트 제어 멀티탭

BroadLink RM Mini 3 - IR 리모컨 송신에 최적

Orvibo S20 - 스마트 플러그

MiPOW SmartBulb - 색상 및 효과가 지원되는 스마트 전구





## 제가 연동하고 있는 것들

Pump1 - 화분 10개의 수분 공급 펌프, 아두이노+모터 드라이버+블루투스, Linux Python Fauxmo + C code

Pump2 - 화분 5개 수분 공급 펌프, 아두이노+모터 드라이버+블루투스, Linux Python Fauxmo + C code

PC - Linux Python Fauxmo + etherwake , WOL으로 PC 켜기

Fan - 구형 선풍기, Orvibo S20와 Linux Python Fauxmo + Orvibo

Aircon - BroadLink RM Mini 3와 Linux Python Fauxmo

TV - BroadLink RM Mini3와 Linux Python Fauxmo

Room light - BroadLink RM Mini3와 Linux Python Fauxmo

Living Room light - NodeMCU Mini+ IR LED

Set1 - Fan+Aircon+TV+PC 동시켜기 Linux Python Fauxmo + Custom Code

Temperature - TI KeyFob, BLE GATT Linux,ngrok,Custom Skill

**Thanks!**