

# NodeMCU로 시작하는 사물인터넷 DIY

2019. 10.

김 학 용

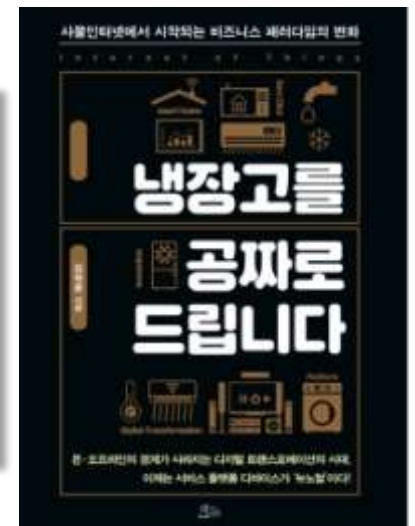
# Speaker : 김학용 교수/공학박사/작가/칼럼니스트

- 現) 순천향대학교 IoT보안연구센터 교수
- 現) 서울특별시 혁신성장위원회 위원 (IoT)
- 現) 삼성물산 Creative Director
- 現) IoT전략연구소 대표
- 前) LG유플러스 M2M사업담당 부장
- 前) 삼성SDS 신사업추진센터 차장

이메일 : [IoTStLabs@gmail.com](mailto:IoTStLabs@gmail.com)

[honest72@sch.ac.kr](mailto:honest72@sch.ac.kr)

전 화 : 010-4711-1434



# Open Source Hardware Platform

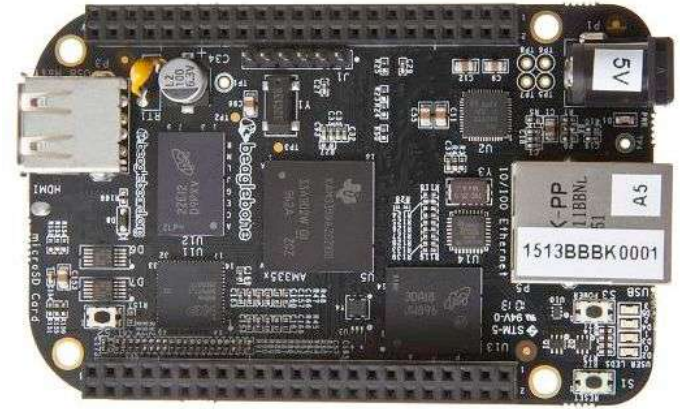
◆ 공개 코드를 이용하여 원하는 IT 기기를 만들어 볼 수 있는 개발 보드



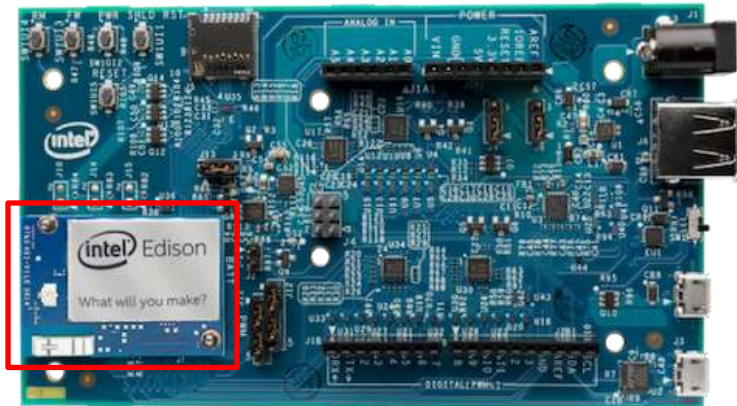
Arduino Uno



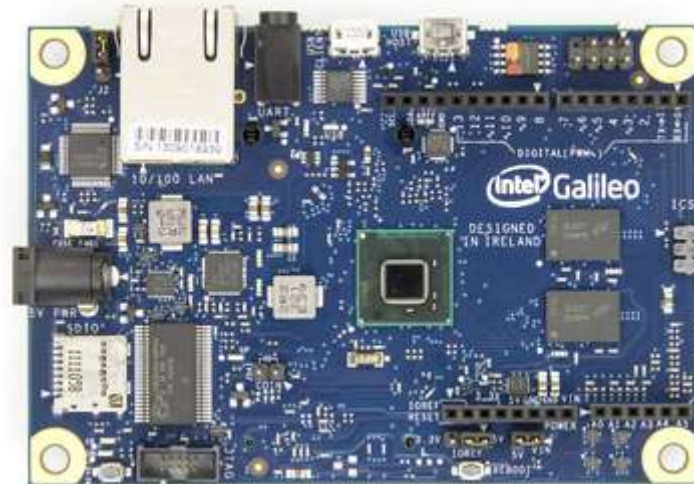
Raspberry Pi 3 B+



Beaglebone Black



Intel Edison



Intel Galileo



NodeMCU



# 아두이노(Arduino)

- ◆ Atmel AVR 마이크로 컨트롤러 기반의 원보드 마이크로컴퓨터
- ◆ 전문적인 지식이나 기술이 없는 초보자도 손쉽게 자신의 아이디어를 구현해 보는 것이 가능
- ◆ 아두이노 하드웨어 디자인은 모두 공개된 오픈소스임



← Arduino Mega



← Arduino UNO R3



← Arduino Nano 3



← Arduino Mini



← Arduino Gemma

# 아두이노(Arduino)란?

|                    |                      |                        |                    |                       |                    |              |          |                 |
|--------------------|----------------------|------------------------|--------------------|-----------------------|--------------------|--------------|----------|-----------------|
| ENTRY LEVEL        | UNO                  | LEONARDO               | 101                | ESPLORA               | MICRO              | NANO         | MINI     | MKR2UNO ADAPTER |
|                    | STARTER KIT          | LCD SCREEN             |                    |                       |                    |              |          |                 |
| ENHANCED FEATURES  | MEGA                 | ZERO                   | DUE                | MEGA ADK              | MO                 | MO PRO       | MKR ZERO | MOTOR SHIELD    |
|                    | USB HOST SHIELD      | PROTO SHIELD           | MKR PROTO SHIELD   | 4 RELAYS SHIELD       |                    |              |          |                 |
|                    | MEGA PROTO SHIELD    | MKR RELAY PROTO SHIELD | ISP                | USB2SERIAL MICRO      |                    |              |          |                 |
|                    | USB2SERIAL CONVERTER |                        |                    |                       |                    |              |          |                 |
| INTERNET OF THINGS | YÚN                  | ETHERNET               | TIAN               | INDUSTRIAL 101        | LEONARDO ETH       | MKR FOX 1200 | MKR1000  |                 |
|                    | YUN MINI             | YÚN SHIELD             | WIRELESS SD SHIELD | WIRELESS PROTO SHIELD | ETHERNET SHIELD V2 |              |          |                 |
|                    | GSM SHIELD V2        | MKR IoT BUNDLE         |                    |                       |                    |              |          |                 |

# 아두이노(Arduino)란?

| Description                 | Arduino UNO R3                   | Arduino Nano 3                   | Arduino Pro Mini                 | Arduino Mega 2560               |
|-----------------------------|----------------------------------|----------------------------------|----------------------------------|---------------------------------|
| Microcontroller             | ATmega328P                       | ATmega328P                       | ATmega328P                       | ATmega2560                      |
| Operating Voltage           | 5V                               | 5V                               | 5V or 3.3V                       |                                 |
| Input Voltage (Recommended) | 7V - 12V                         | -                                | -                                | 7V - 12V                        |
| Input Voltage (limits)      | 6V - 20V                         | -                                | -                                | 6V - 20V                        |
| Digital I/O pin             | 14 개,<br>그 중 6개는 PWM             | 14 개,<br>그 중 6개는 PWM             | 14 개,<br>그 중 6개는 PWM             | 54 개,<br>그 중 6개는 PWM            |
| Analog Input pin            | 6 개                              | 6 개                              | 6 개                              | 16 개                            |
| DC current per I/O pin      | 40 mA                            | 20 mA                            | 40 mA                            | 20 mA                           |
| DC current for 3.3V pin     | 50 mA                            | -                                | -                                | 50 mA                           |
| Flash Memory                | 32 KB, 0.5 KB used by bootloader | 32 KB, 0.5 KB used by bootloader | 32 KB, 0.5 KB used by bootloader | 256 KB, 8 KB used by bootloader |
| SRAM                        | 2 KB                             | 2 KB                             | 2 KB                             | 8 KB                            |
| EEPROM                      | 1 KB                             | 1 KB                             | 1 KB                             | 4 KB                            |
| Clock speed                 | 16 Mhz                           | 16 Mhz                           | 16 Mhz                           | 16 Mhz                          |
| Dimension                   | 68.6 mm x 53.4 mm                | 45 mm x 18 mm                    | 33 mm x 18 mm                    | 101.5 mm x 53.4 mm              |
| Weight                      | 25 g                             | 5 g                              | 5 g                              | 37 g                            |

※ PWM : Pulse Width Modulation

# 아두이노에서 와이파이를 이용하는 방법

- ◆ Wi-Fi Shield 같은 확장 쉴드 이용 → 추가 비용
- ◆ 통신 모듈을 내장한 아두이노 제품 이용 → 고가(高價)



아두이노 와이파이 쉴드 (Wifi Shield (Fi250) V1.1)

66,000원

디지털/가전 > PC액세서리 > 기타PC액세서리

등록일 2018.02. · ♥ 찜하기 0 · 📄 신고하기



아두이노 윤 (Arduino Yun)

110,000원

디지털/가전 > PC액세서리 > 기타PC액세서리

네이버페이 구매평 5 · 등록일 2014.08. ·

- ◆ ESP8266이라는 저가의 와이파이 모듈 이용
  - 2열로 핀이 배치되어 있어 와이어링이 불편



아두이노 우노



와이파이 모듈  
(ESP8266)

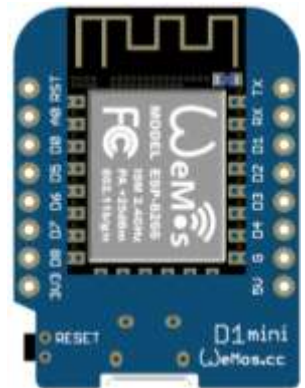
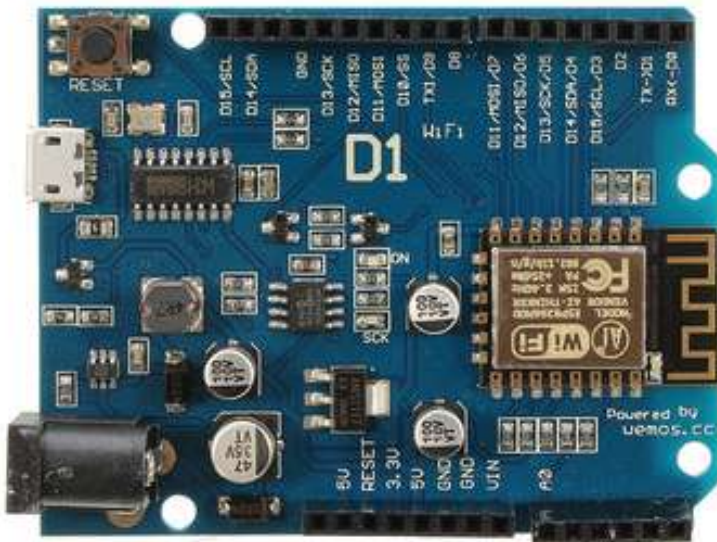


NodeMCU

# WeMos D1/D1R2

## ◆ Arduino Uno + ESP8266-12E

- 와이파이 모듈이 탑재된 변종 아두이노 우노 → 1만원 내외의 저렴한 가격
- 동작 전압 : 3.3V
- Digital I/O pins : 11개
- Analog I/O pins : 1개
- Clock Speed : 80MHz/160MHz





# Raspberry Pi 3 B+

## ◆ 기존 제품 대비 프로세서와 네트워크 기능을 대폭 강화

- 브로드컴 BCM2837(1.2GHz) → BCM2837B0(1.4GHz Quad Core) 사용
- 와이파이 : 2.4GHz 및 5GHz 모두 지원
- 블루투스 : 4.2 및 BLE 지원
- USB 2.0 : 최대 300Mbps로 데이터 전송
- 기가비트 이더넷 및 PoE 지원



# NodeMCU or ESP8266-12E

- ◆ NodeMCU (ESP8266-12E) : 아두이노에 무선 모듈(ESP) 추가한 것
  - 엄밀하게는 ESP8266에 32-bit RISC 프로세서를 추가한 오픈소스 하드웨어 플랫폼
  - 아두이노(5V)와 ESP8266(3.3V)의 서로 다른 동작 전압을 3.3V로 통일
  - 아두이노의 개발 환경을 그대로 승계



아두이노 우노



와이파이 모듈  
(ESP8266)

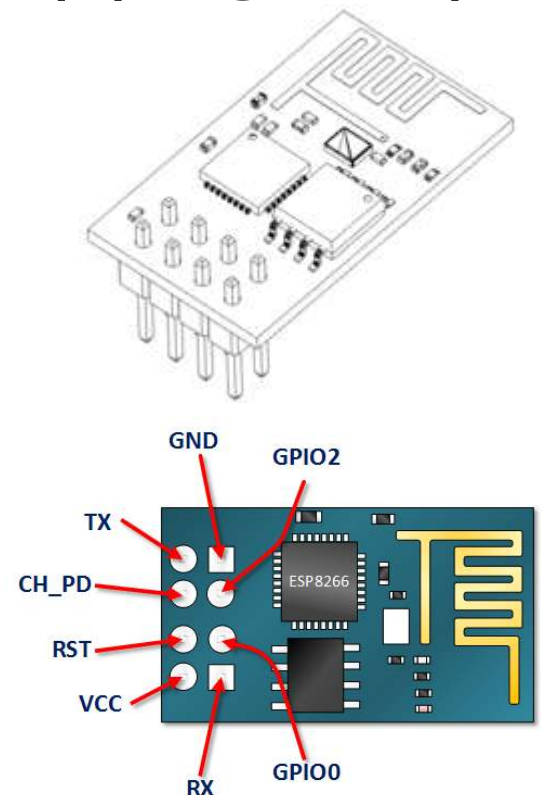
NodeMCU

# ESP8266

- ◆ 중국 상하이에 있는 ESPressif Systems에서 개발한 와이파이 모듈 (2013년 12월 30일 발표)
- ◆ TCP/IP Full Stack과 Micro-Controller를 탑재
- ◆ 2014년 8월, AI-Thinker라는 제조사에서 ESP-01이라는 상품 출시

## ◆ ESP8266의 주요 특징

- Tensilica Xtensa의 L106 32-bit microprocessor core
- 802.11 b/g/n 지원 (2.4GHz 대역)
- WEP, WPA/WPA2 인증 방식 지원
- 11개의 GPIO, 1개의 아날로그 입력 (10-bit ADC)  
→ 아두이노와는 TX/RX 핀을 통한 시리얼 통신
- SPI, I2C, I2S, UART 등 다양한 인터페이스 지원
- ESP8266 SDK, [Arduino IDE](#), Lua, ESPRUIINO, MicroPython, Sming 등 다양한 개발 환경 지원



# NodeMCU or ESP8266-12E

## ◆ ESP8266의 진화



ESP-01



ESP-02



ESP-03



ESP-04



ESP-05



ESP-06



ESP-07



ESP-08



ESP-09



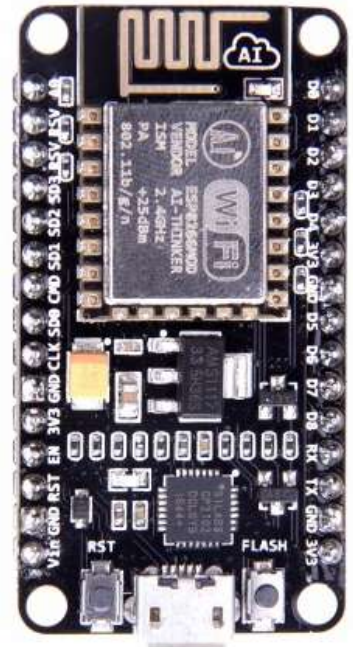
ESP-10



ESP-11



ESP-12E



NodeMCU



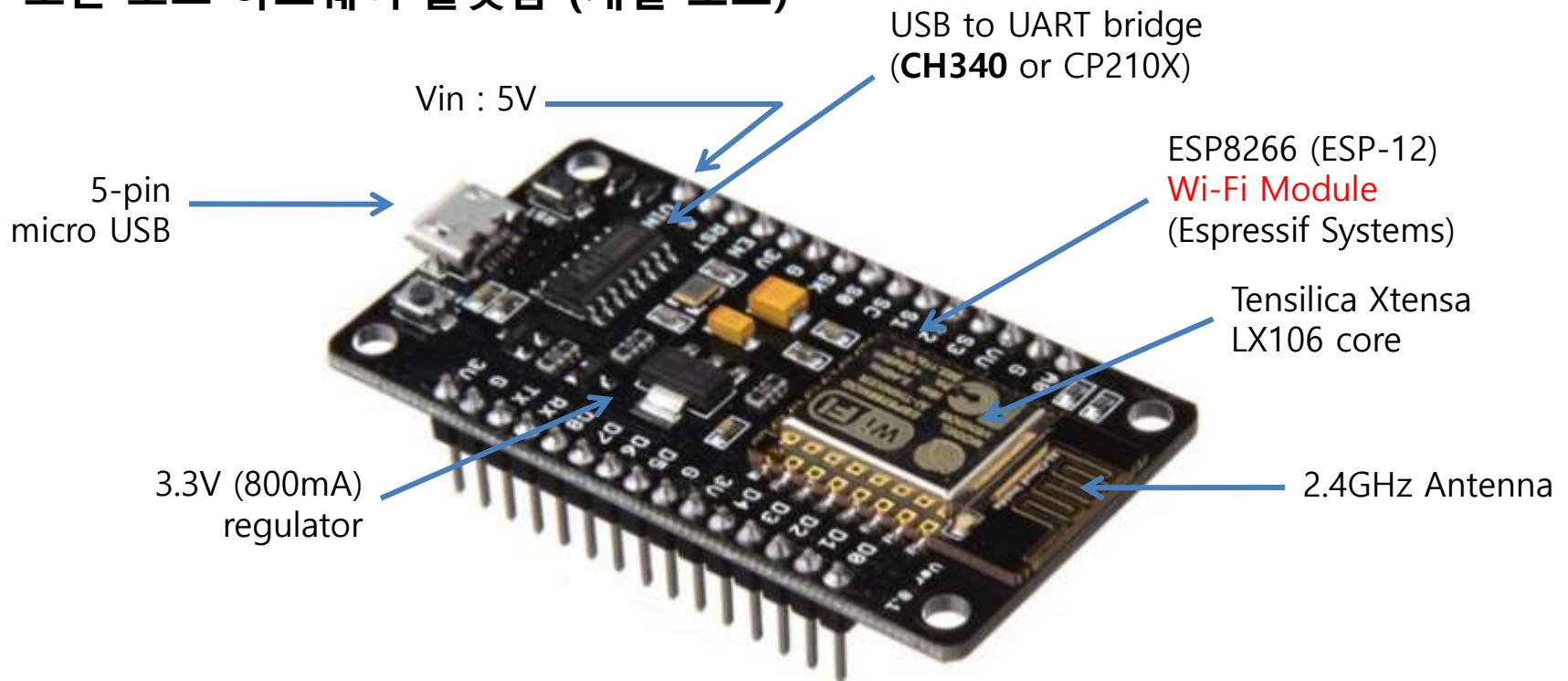
# NodeMCU ESP-12E

## ◆ NodeMCU의 주요 사양

- **Wi-Fi (IEEE 802.11b/g/n) 지원**
  - 11b 모드에서 최대 19.5dBm (89mW) 출력 파워
  - Power down leakage current of  $< 10\mu\text{A}$
- **Wi-Fi Direct (P2P) 및 Soft-AP 지원**
- **Integrated TCP/IP protocol stack**
- **32-bit RISC CPU 80MHz (OC 160MHz)**
- **64KB instruction RAM, 96KB data RAM, 4MB flash memory**
- **13 GPIO pins (11 GPIO + TX/RX) – Not 5v tolerant**
- **SPI, I<sup>2</sup>C**
- **1 10-bit AD (1 analog pin)**
- **동작 온도 범위 : -40°C~125°C**
- **Can be programmed with Lua, Arduino IDE, C/C++, Python, Basic, JavaScript**

# NodeMCU

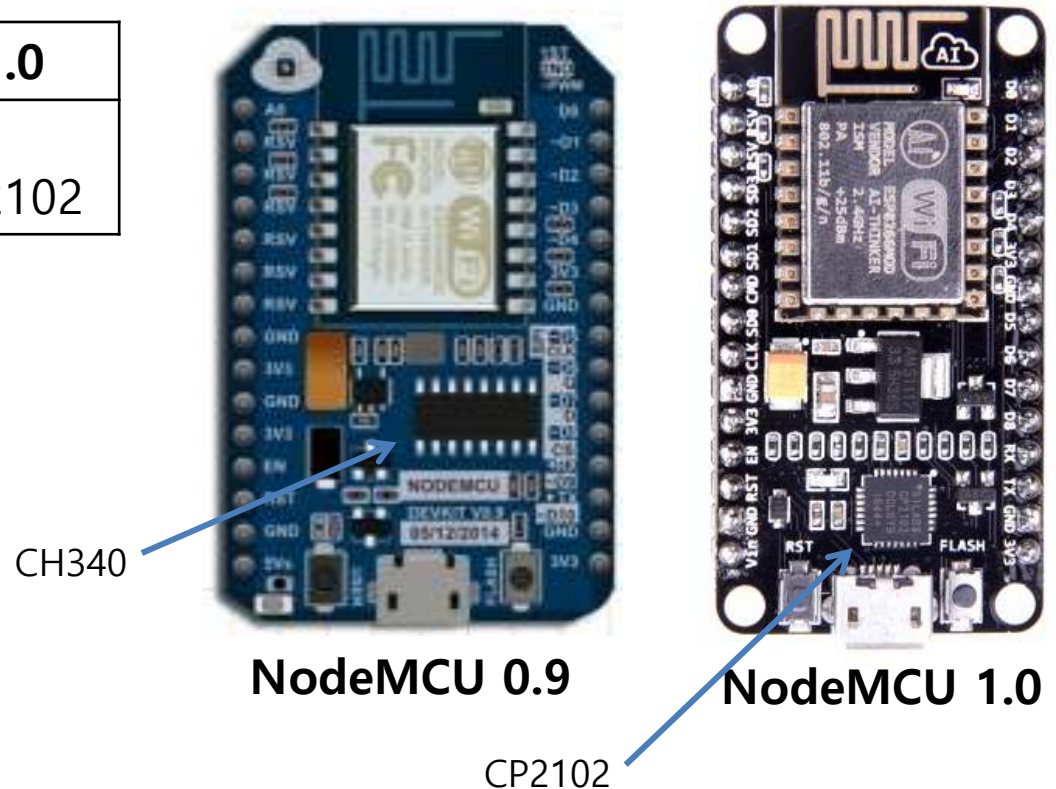
- ◆ An open-source firmware and development kit that helps you to prototype your IOT product within a few Lua script lines
  - Lua 스크립트 랭귀지를 이용해서 아주 쉽게 → Arduino IDE 환경에서 Sketch 사용
  - IoT 제품의 프로토타입을 만들어 볼 수 있게 해주는
  - 오픈 소스 하드웨어 플랫폼 (개발 보드)



# NodeMCU or ESP8266-12E

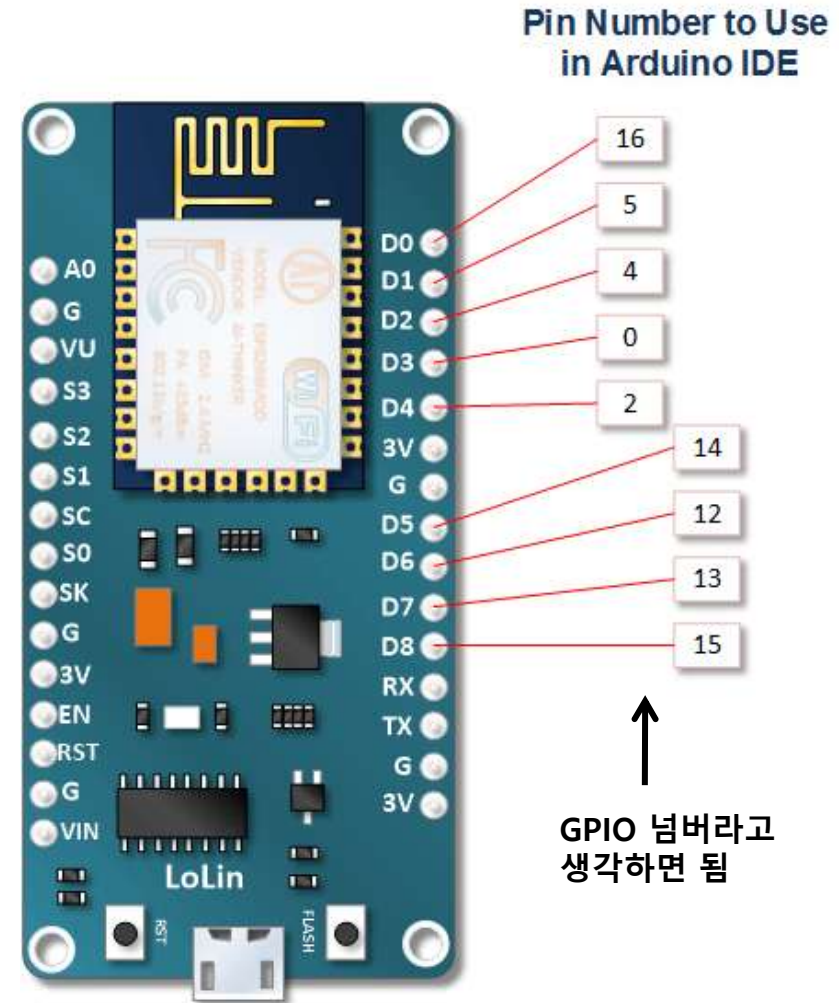
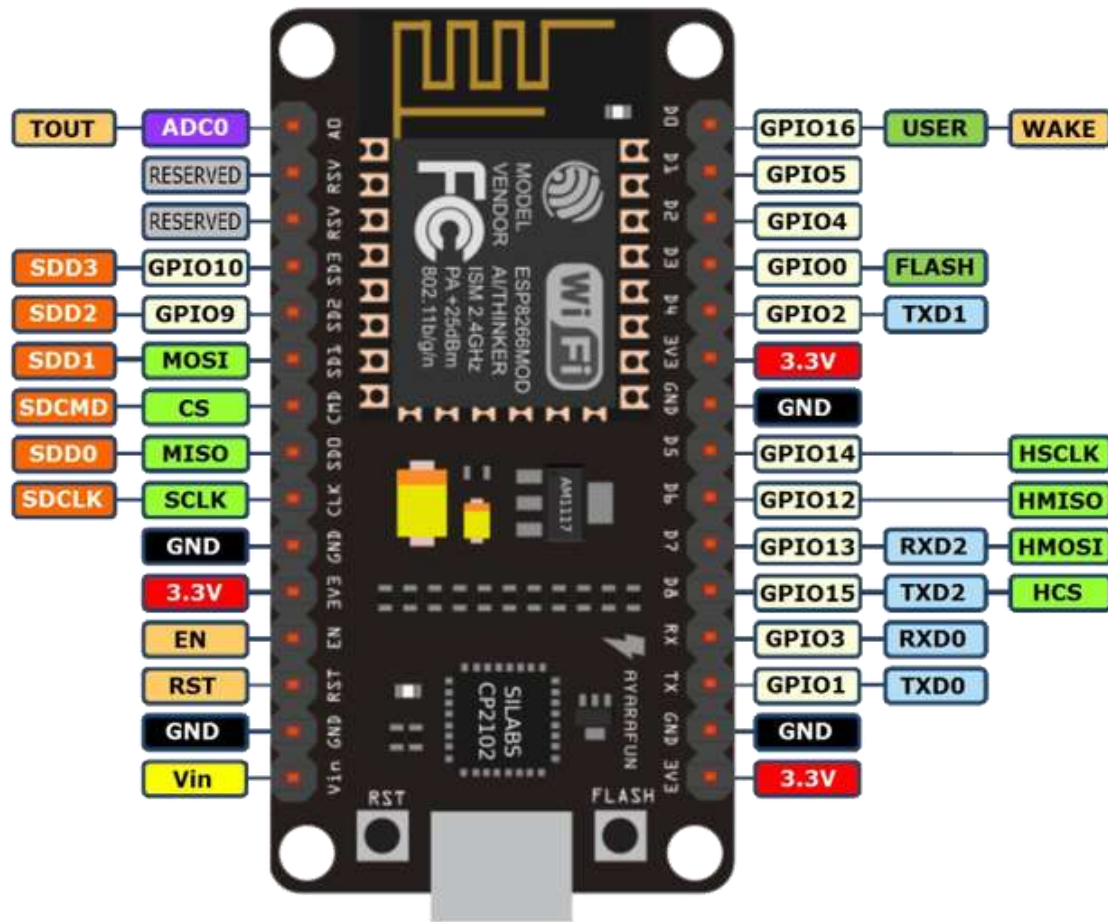
## ◆ NodeMCU의 종류

| NodeMCU 0.9  | NodeMCU 1.0   |
|--|---|
| <ul style="list-style-type: none"><li>• ESP8266-12</li><li>• CH340</li></ul> | <ul style="list-style-type: none"><li>• ESP8266-12E</li><li>• CH340 or CP2102</li></ul> |



※ CH340을 이용한 NodeMCU는 빵판에 이용하기 불편하므로 CP2102를 이용한 제품을 구매할 것!!

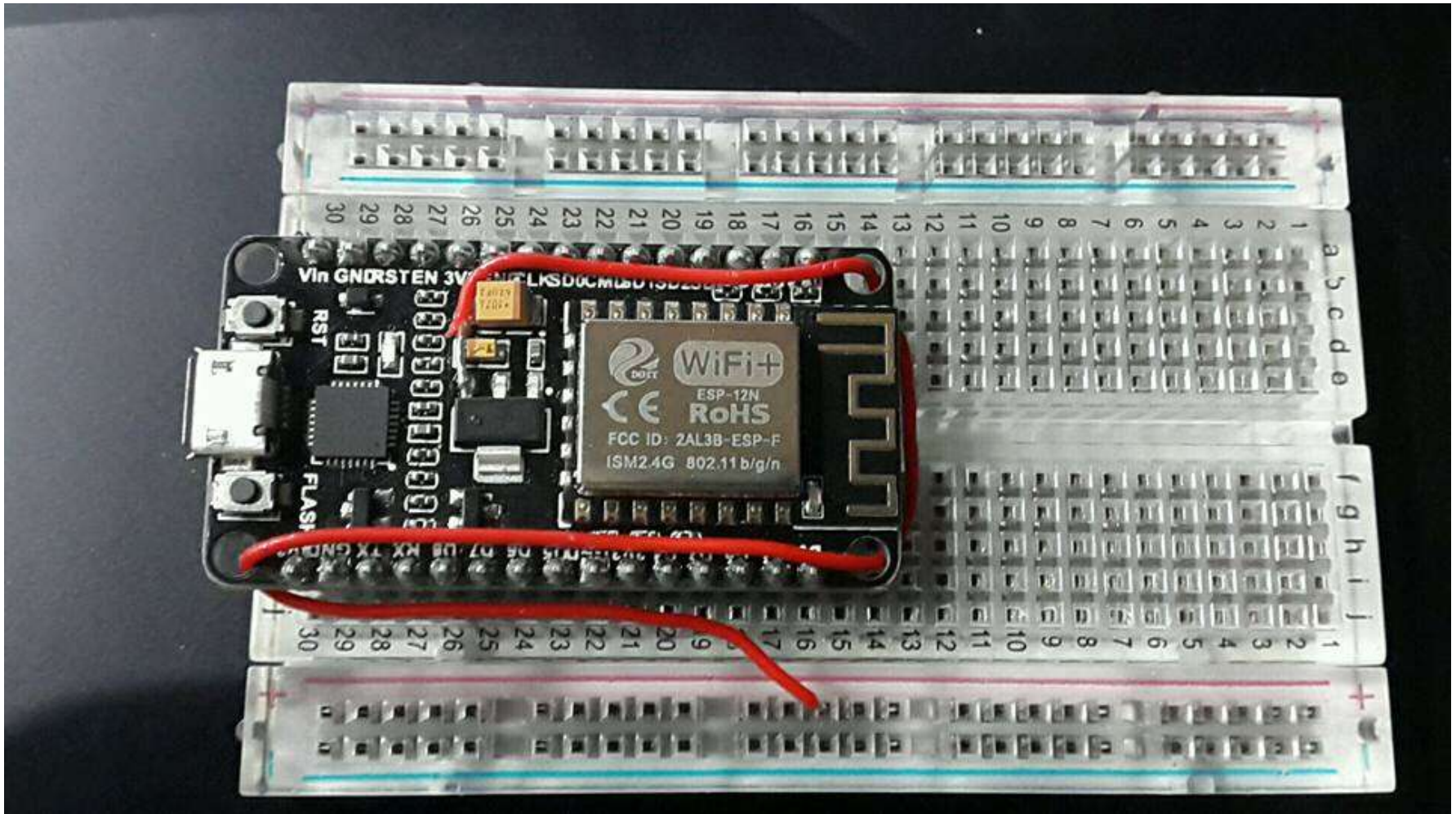
# NodeMCU ESP-12E Pinouts and Pin Number





# NodeMCU에서 5V 사용하기

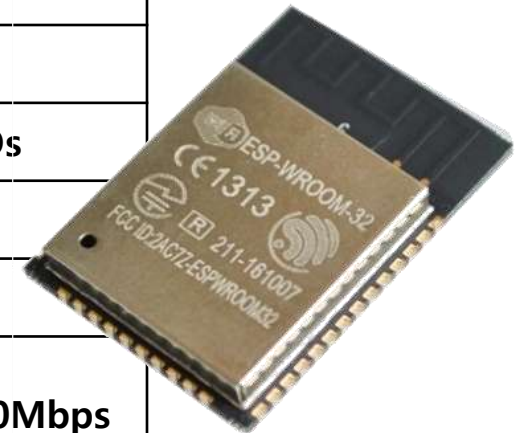
- ◆ Vin 을 사용하거나 아래 사진처럼 이용할 수 있음



# ESP-32S의 주요 특성

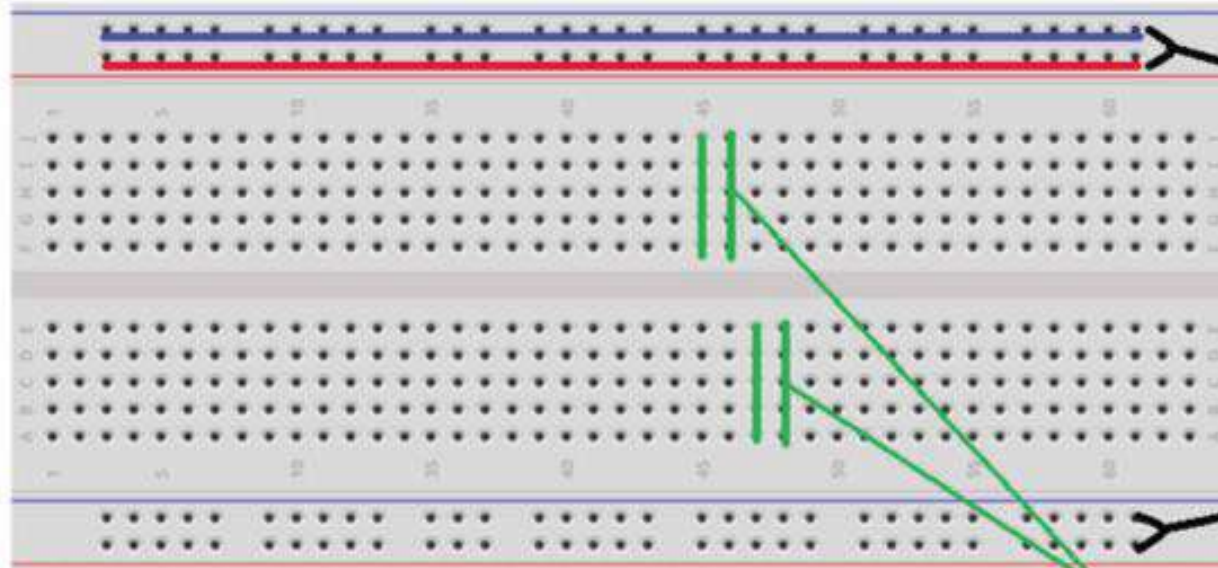
- ◆ Xtensa의 single-/dual-core 32-bit LX6 microprocessor 사용
- ◆ Wi-Fi 뿐만 아니라 Bluetooth도 동시에 지원

|                    | ESP-8266              | ESP-32S   |
|--------------------|-----------------------|---|
| Microprocessor     | Xtensa LX106 (32-bit) | Xtensa LX6 (dual-core 32-bit)   |
| Memory             | 128KB                 | 448KB ROM, 520KB SRAM,<br>16KB SRAM in RTC  |
| Storage            | 4MB Flash RAM         | 4MB Flash RAM   |
| GPIO               | 13 GPIOs              | 34 programmable GPIOs   |
| Analog support     | 1 10-bit AD Converter | 2 12-bit SAR ADC  |
| Clock Speed        | 80MHz (160MHz)        | 80/160/240MHz   |
| Communications     | 802.11 b/g/n (2.4GHz) | 802.11 b/g/n/e/i<br>802.11 n (2.4GHz), up to 150Mbps<br>WPA/WPA2/WPA2-Enterprise/WPS<br>Bluetooth v4.2 BR/EDR & BLE |
| Price (AliExpress) | 3\$ 내외                | 6\$ 내외  |



# 부품 설명

## ◆ Bread Board (빵판)

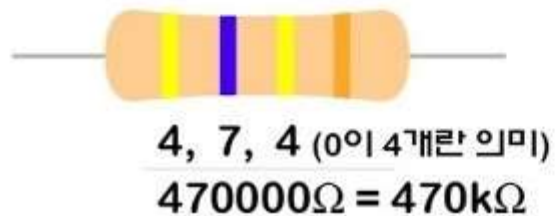
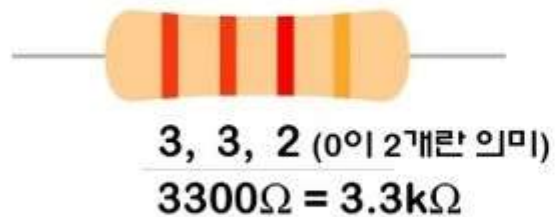
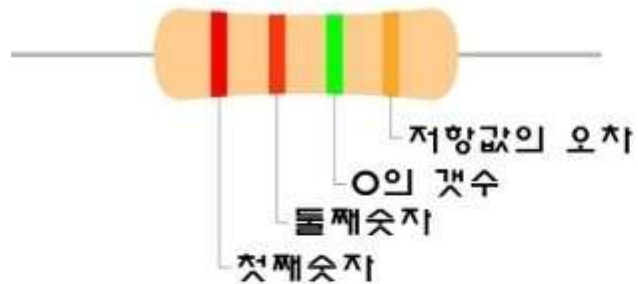


이부분은 가로 방향으로 연결되어 있으며 전원을 공급하는 부분입니다.

이부분은 세로 방향으로 연결되어 있으며 전자 부품을 꽂는 부분입니다.

# 부품 설명

## ◆ Resister (저항)



| 색  | 값          |
|--|------------|
|  검정색   | 0          |
|  갈 색   | 1          |
|  빨강색   | 2          |
|  주황색   | 3          |
|  노란색   | 4          |
|  초록색   | 5          |
|  파란색   | 6          |
|  보라색   | 7          |
|  회 색  | 8          |
|  하얀색 | 9          |
|  은 색 | $\pm 10\%$ |
|  금 색 | $\pm 5\%$  |



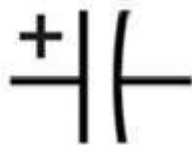
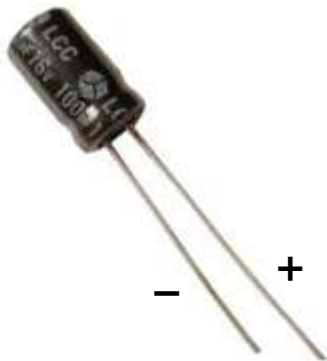
# 부품 설명

## ◆ Capacitor or Condenser (콘덴서)

- 직류를 저장(condense)하고 교류는 통과
  - 콘덴서의 용량만큼 저장된 후에는 전류가 흐르지 않음
- 불안정한 전원을 잡아주거나 노이즈를 차단
- 종류: 전해 콘덴서, 세라믹 콘덴서, 탄탈 콘덴서



세라믹 콘덴서

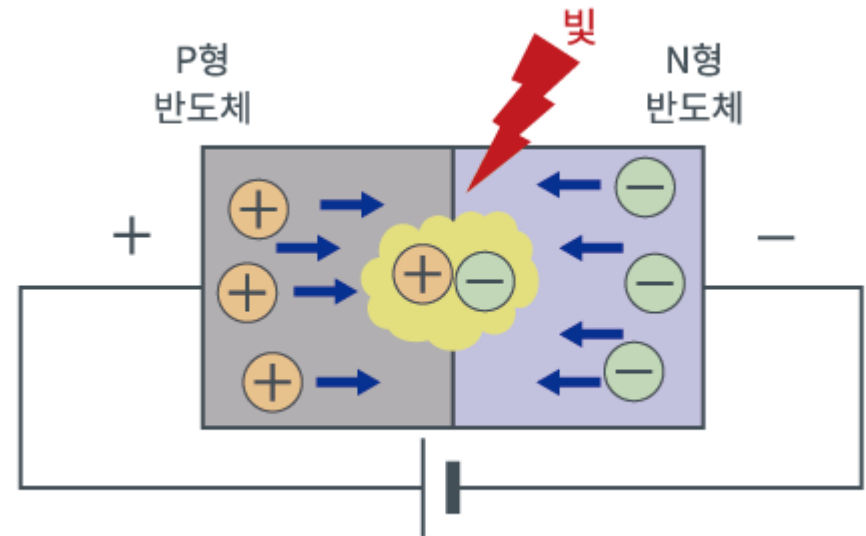
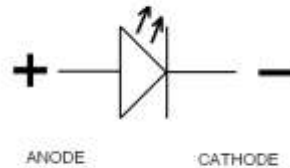
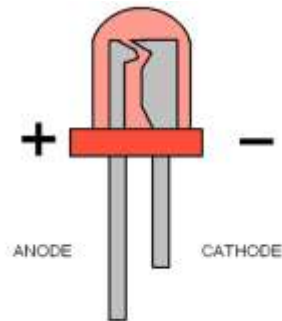


전해 콘덴서

# 부품 설명

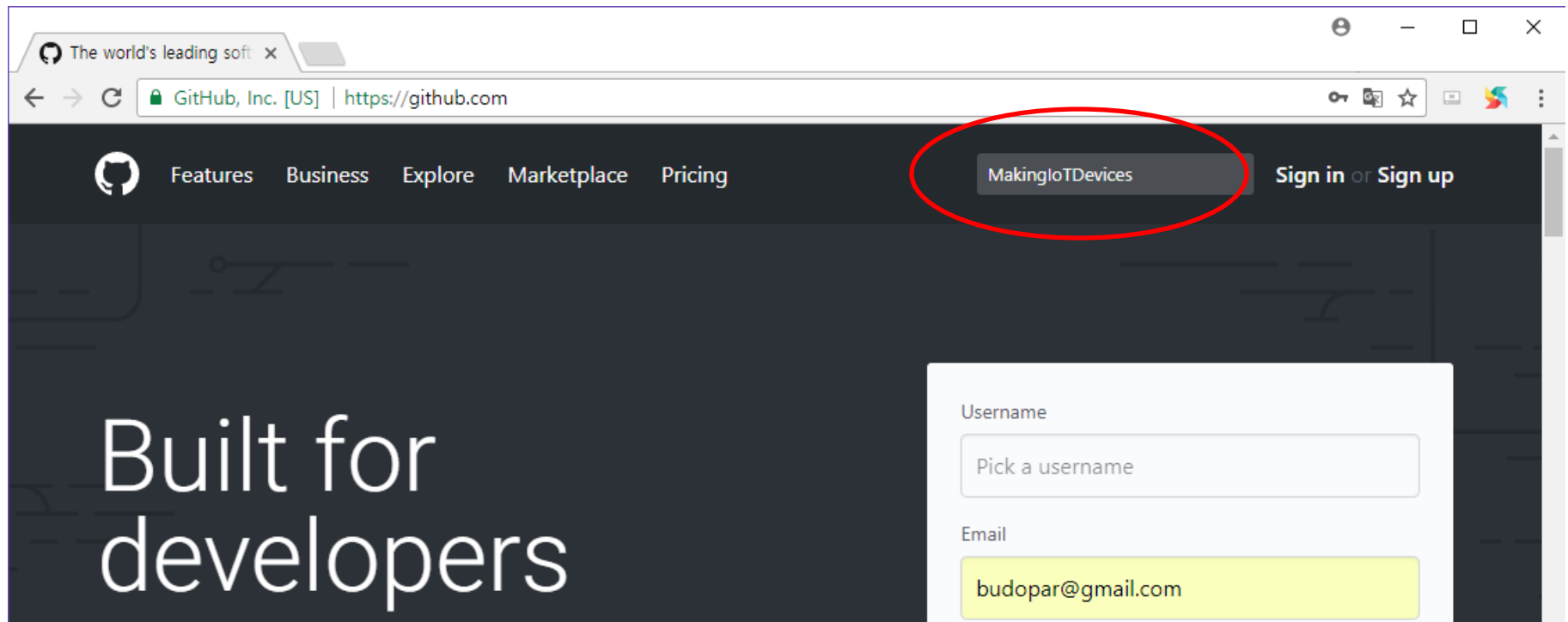
## ◆ LED (Light Emitting Diode)

- Diode : 전류를 한쪽으로 흐르게 하는 소자



# 예제 파일 다운로드

- ◆ Github.com 접속 → MakingIoTDevices 검색 → DonggukUniv 클릭
- ◆ 그냥 이 주소로 접속 : <https://github.com/iotstlabs/DonggukUniv>



# 예제 파일 다운로드

This screenshot shows the GitHub search results for the repository 'MakingIoTDevices'. The browser address bar shows the search URL. The repository is listed with its name, a description, and the language used (Arduino). A sidebar on the right shows the language distribution.

Search · MakingIoTDevices x

GitHub, Inc. [US] | <https://github.com/search?utf8=✓&q=MakingIoTDevices&type=>

Features Business Explore Marketplace Pricing MakingIoTDevices Sign in or Sign up

Repositories 1 Code Commits Issues Wikis Users Advanced search

**iotstlabs/MakingIoTDevices** ● Arduino

Source codes for the book, Making IoT Devices using NodeMCU (ESP8266)

Languages

|         |   |
|---------|---|
| Arduino | 1 |
|---------|---|

This screenshot shows the repository page for 'MakingIoTDevices'. It includes the repository name, a description, and a list of files. The files are listed with their names, the commit they belong to, and the time since the commit.

GitHub - iotstlabs/MakingIoTDevices

GitHub, Inc. [US] | <https://github.com/iotstlabs/MakingIoTDevices>

Branch: master New pull request Find file Clone or download

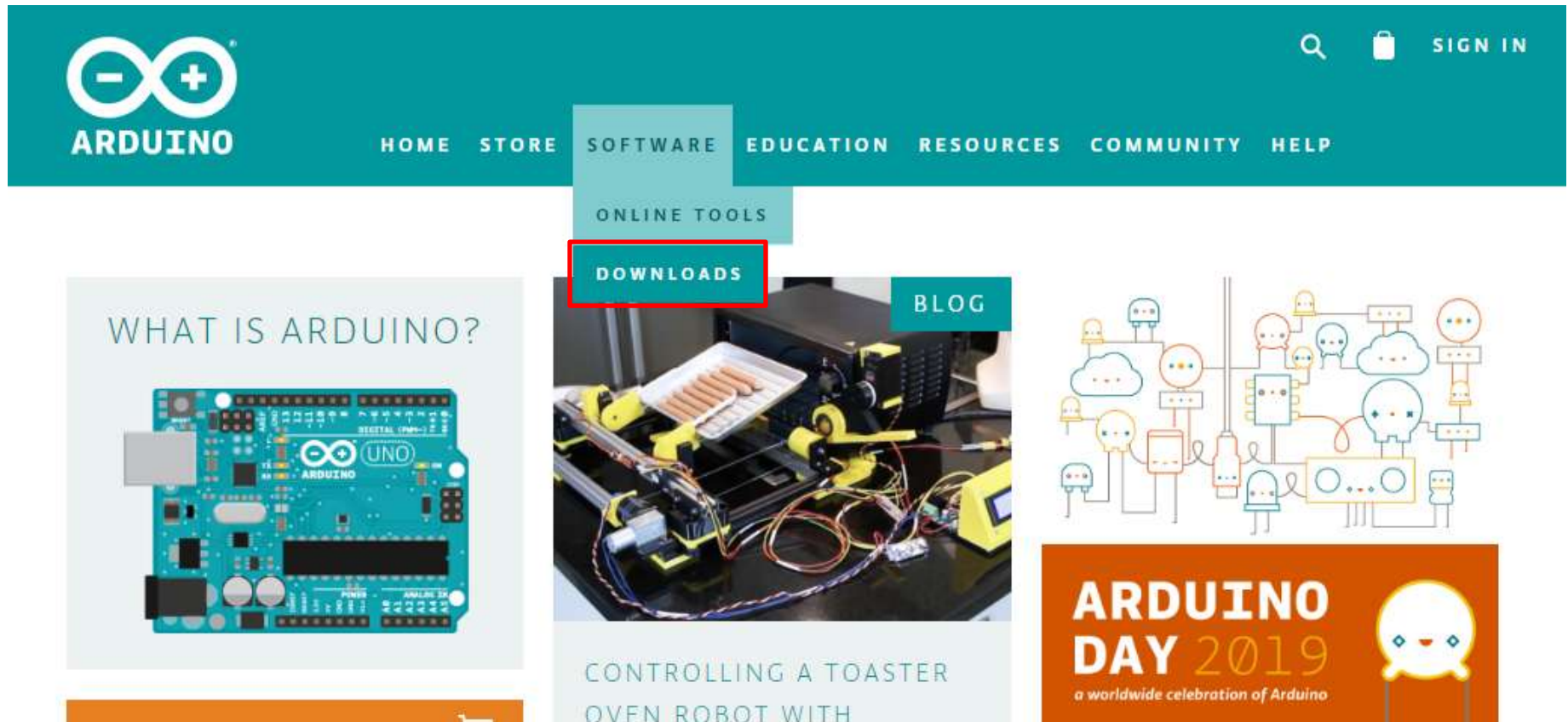
iotstlabs 2nd commit ... Latest commit 11b97c9 2 days ago

|                      |            |            |
|----------------------|------------|------------|
| IoTStLabs.zip        | 2nd commit | 2 days ago |
| NTP_SSD1306.ino      | 2nd commit | 2 days ago |
| Push_Button_Stay.ino | 2nd commit | 2 days ago |



# 아두이노 개발 환경 (IDE)

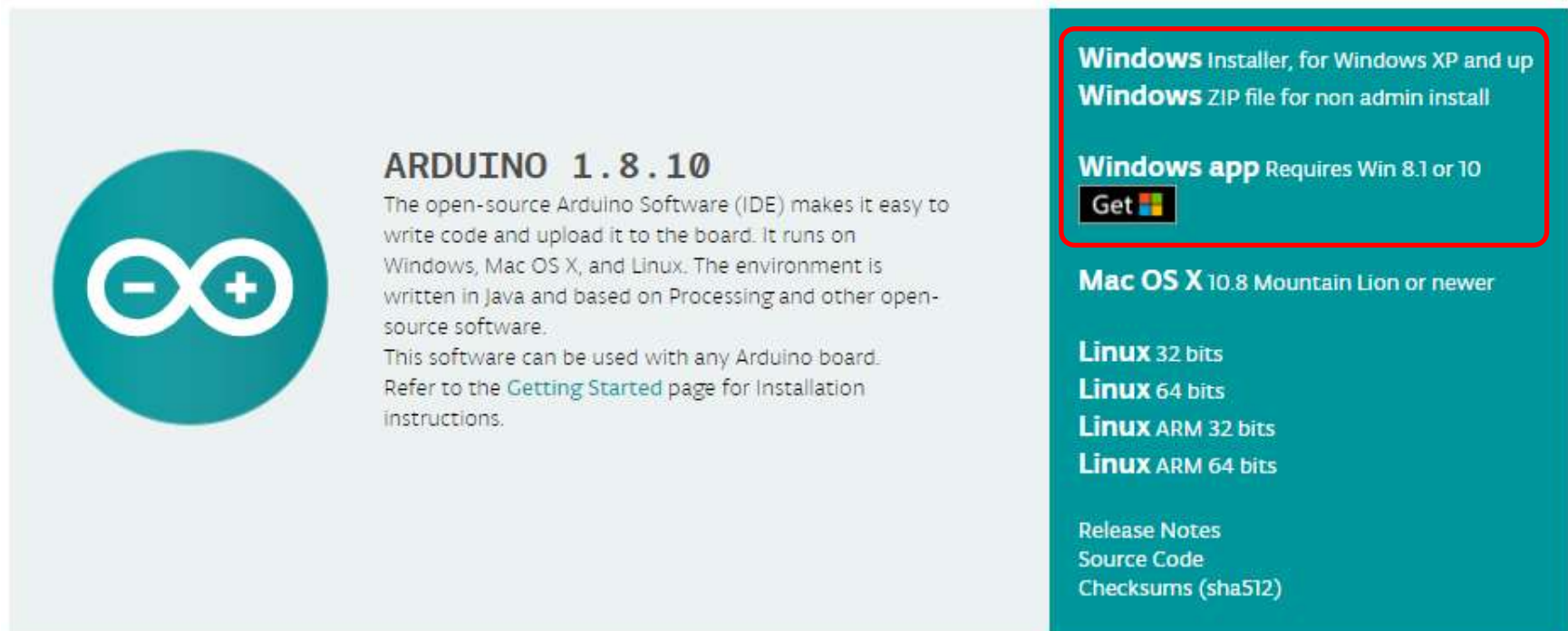
- ◆ 개발자를 위해 오픈소스 기반 개발용 프로그램을 무료로 제공
- ◆ <http://www.arduino.cc> 에서 다운받아 설치 후 프로그래밍
- ◆ PC의 문서 폴더에 설치 ➔ 문서>arduino



# 아두이노 개발 환경 (IDE)

## ◆ 자신의 PC 환경에 맞는 아두이노 다운로드 및 설치

- 아래 화면이 나올 때까지 스크롤 다운
- PC 주인은 Windows Installer 혹은 Windows app 클릭
- 공용 PC의 경우에는 Windows ZIP file for non admin install 클릭




**ARDUINO 1.8.10**

The open-source Arduino Software (IDE) makes it easy to write code and upload it to the board. It runs on Windows, Mac OS X, and Linux. The environment is written in Java and based on Processing and other open-source software.

This software can be used with any Arduino board. Refer to the [Getting Started](#) page for Installation instructions.

**Windows** Installer, for Windows XP and up  
**Windows** ZIP file for non admin install

**Windows app** Requires Win 8.1 or 10  
[Get](#) 

**Mac OS X** 10.8 Mountain Lion or newer

**Linux** 32 bits  
**Linux** 64 bits  
**Linux** ARM 32 bits  
**Linux** ARM 64 bits

[Release Notes](#)  
[Source Code](#)  
[Checksums \(sha512\)](#)

# 아두이노 개발 환경 (IDE)

## ◆ 기부(contribution) 여부 선택 – 일반적으로 Just Download 누름



### Contribute to the Arduino Software

Consider supporting the Arduino Software by contributing to its development. (US tax payers, please note this contribution is not tax deductible). [Learn more](#) on how your contribution will be used.

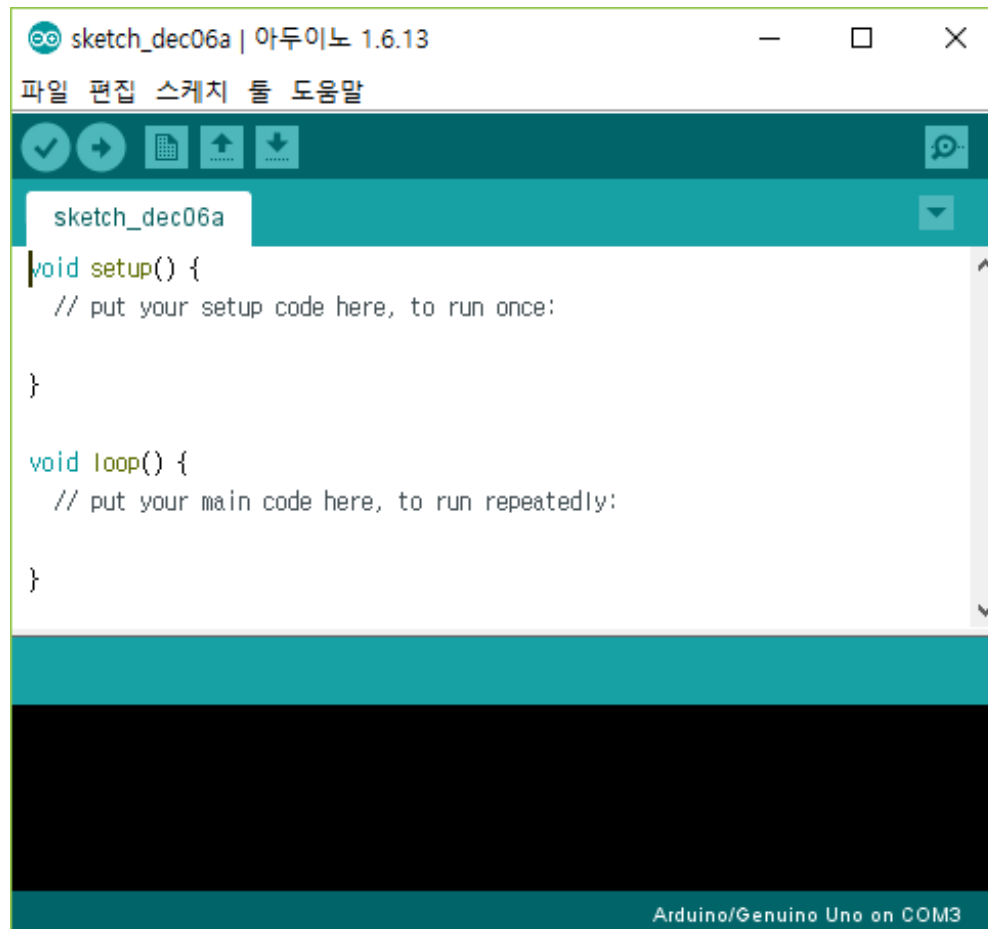


JUST DOWNLOAD

CONTRIBUTE & DOWNLOAD

# 아두이노 개발 환경 (IDE)

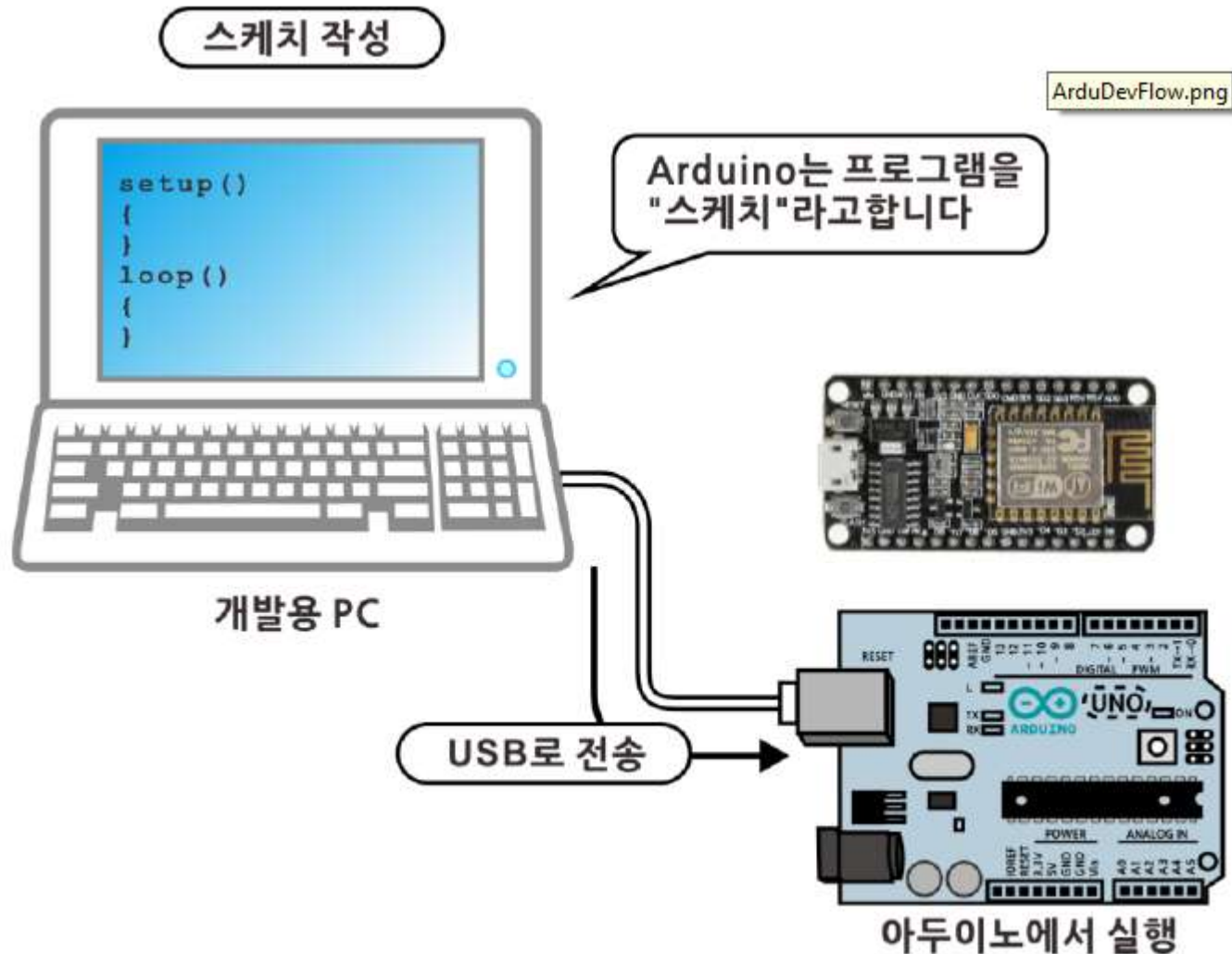
- ◆ 다운받은 프로그램(아두이노 스케치)을 설치 후 실행
- ◆ NodeMCU에서 수행할 기능을 작성(sketch)하여 보드에 업로드





# 아두이노 개발 환경 (IDE)

※ IDE : Integrated Development Environment



※ 아두이노와 NodeMCU가 사용하는 USB 케이블은 유형이 다릅니다. NodeMCU는 Micro USB 케이블 사용.

# 아두이노 개발 환경 설정

## ◆ 아두이노 IDE 실행 후 파일 - 환경설정 클릭 - 아래값 입력

- [http://arduino.esp8266.com/stable/package\\_esp8266com\\_index.json](http://arduino.esp8266.com/stable/package_esp8266com_index.json)

The screenshot shows the Arduino IDE interface. On the left, the '파일' (File) menu is open, and '환경설정' (Environment Settings) is highlighted. The main window displays the '환경설정' (Environment Settings) dialog box, specifically the '네트워크' (Network) tab. The '스케치북 위치' (Sketchbook location) is set to 'C:\Users\Whones\Documents\Arduino'. The '에디터 언어' (Editor language) is set to '시스템 기본설정' (System default settings). The '에디터 글꼴 크기' (Editor font size) is set to 12. The 'Interface scale' is set to '자동' (Automatic) at 100%. The '다음 동작중 자세한 출력 보이기' (Show detailed output during next operation) is set to '컴파일' (Compile) and '업로드' (Upload). The '컴파일러 경고' (Compiler warnings) are set to 'None'. The '출 번호 표시' (Show line numbers) is checked. The '코드 폴딩 사용하기' (Use code folding) is unchecked. The '업로드 후 코드 확인하기' (Check code after upload) is checked. The '외부 에디터 사용' (Use external editor) is unchecked. The '시작시 업데이트 확인' (Check for updates at startup) is checked. The '스케치 파일을 저장할때 새로운 확장자(.pde -> .ino)로 업데이트' (Update sketch files to new extension (.pde -> .ino) when saving) is checked. The '검증 또는 업로드 할 때 저장하기' (Save when verifying or uploading) is checked. The '추가적인 보드 매니저 URLs' (Additional board manager URLs) field contains the URL 'http://arduino.esp8266.com/stable/package\_esp8266com\_index.json'. A red arrow points to this field. The bottom of the dialog box shows the path 'C:\Users\Whones\AppData\Local\Arduino15\preferences.txt'.

환경설정

설정 네트워크

스케치북 위치:  
C:\Users\Whones\Documents\Arduino

에디터 언어: 시스템 기본설정 (아두이노를 재시작해야 함)

에디터 글꼴 크기: 12

Interface scale: ☒ 자동 100% (아두이노를 재시작해야 함)

다음 동작중 자세한 출력 보이기: ☐ 컴파일 ☐ 업로드

컴파일러 경고: None

☒ 줄 번호 표시

☐ 코드 폴딩 사용하기

☒ 업로드 후 코드 확인하기

☐ 외부 에디터 사용

☒ 시작시 업데이트 확인

☒ 스케치 파일을 저장할때 새로운 확장자(.pde -> .ino)로 업데이트

☒ 검증 또는 업로드 할 때 저장하기

추가적인 보드 매니저 URLs [http://arduino.esp8266.com/stable/package\\_esp8266com\\_index.json](http://arduino.esp8266.com/stable/package_esp8266com_index.json)

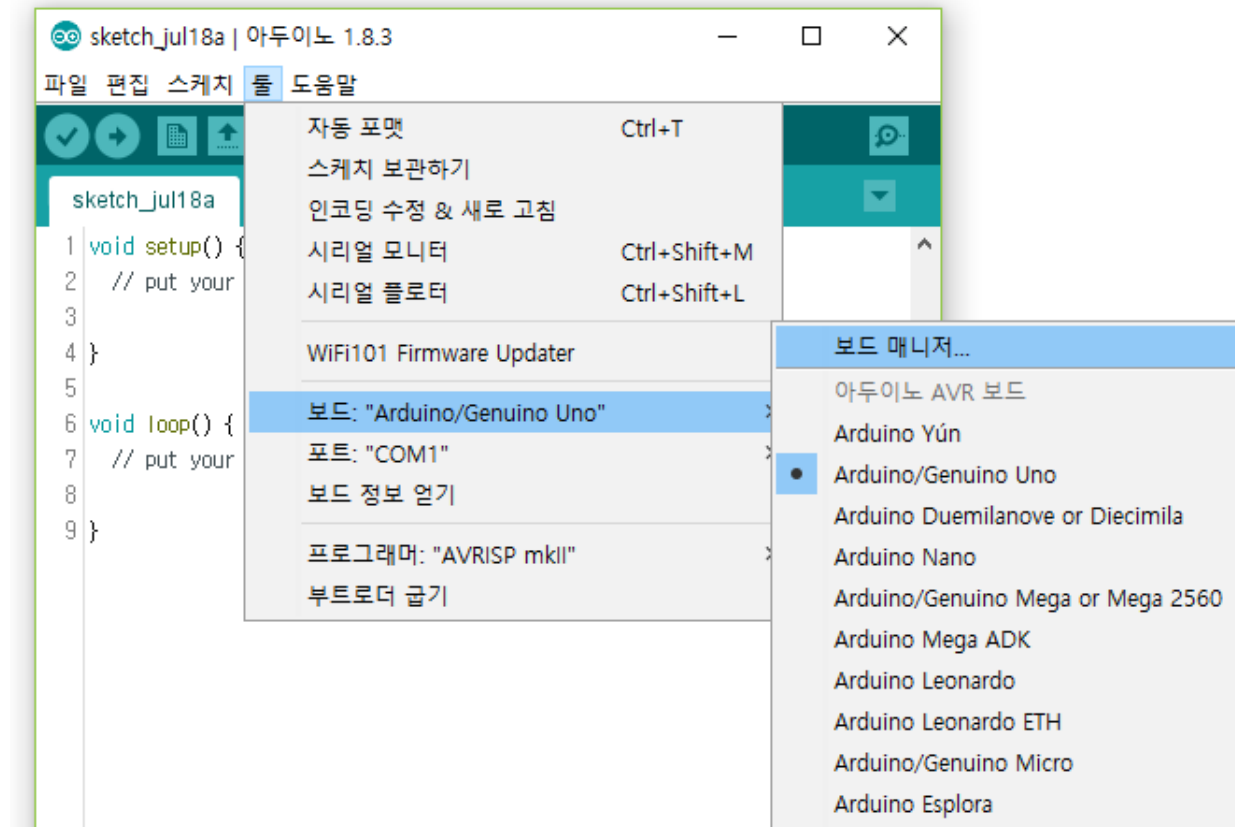
추가적인 환경 설정은 파일에서 직접 편집할 수 있습니다  
C:\Users\Whones\AppData\Local\Arduino15\preferences.txt

참고: <http://www.instructables.com/id/Quick-Start-to-Nodemcu-ESP8266-on-Arduino-IDE/>

# 아두이노 개발 환경 설정

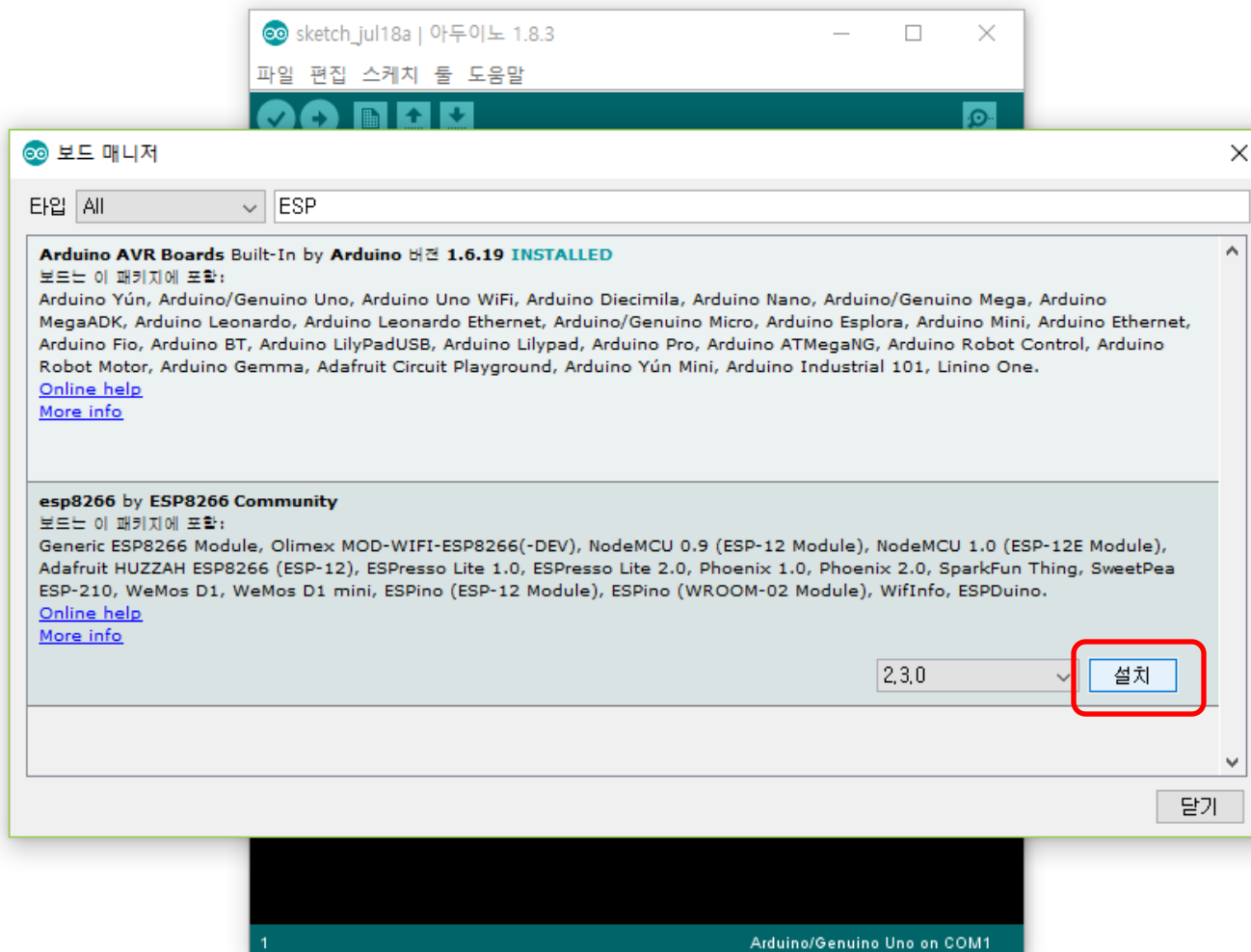
## ◆ 개발 보드 추가

- 도구>보드>보드 매니저 순으로 선택 후 검색창에서 ESP로 검색
  - esp8266 by ESP8266 Community 항목을 클릭 후 설치
  - 가능하면 가장 최신 버전(숫자 높은 것)을 선택해서 설치 ➔ 10분 가량 걸림



# 아두이노 개발 환경 설정

## ◆ 개발 보드 추가

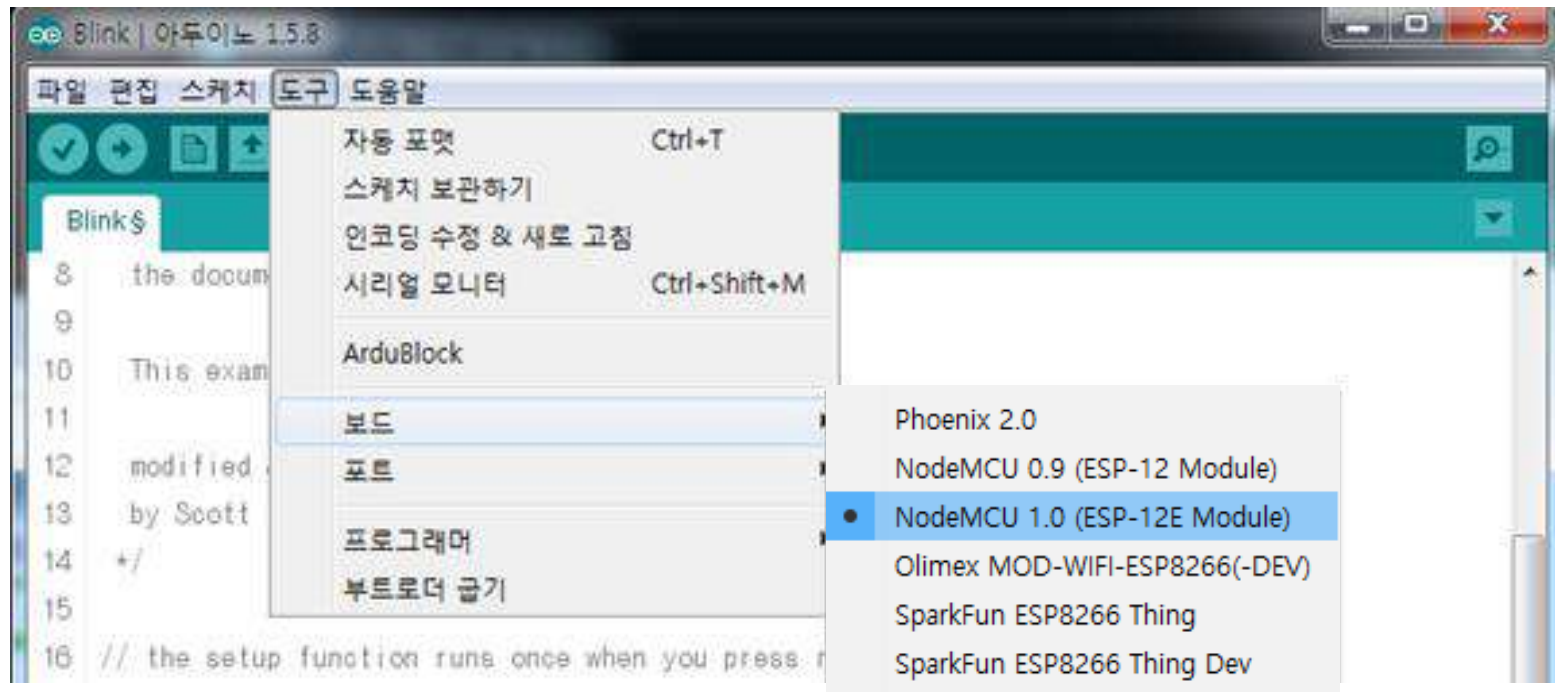




# 아두이노 개발 환경 설정

## ◆ 개발 보드 지정

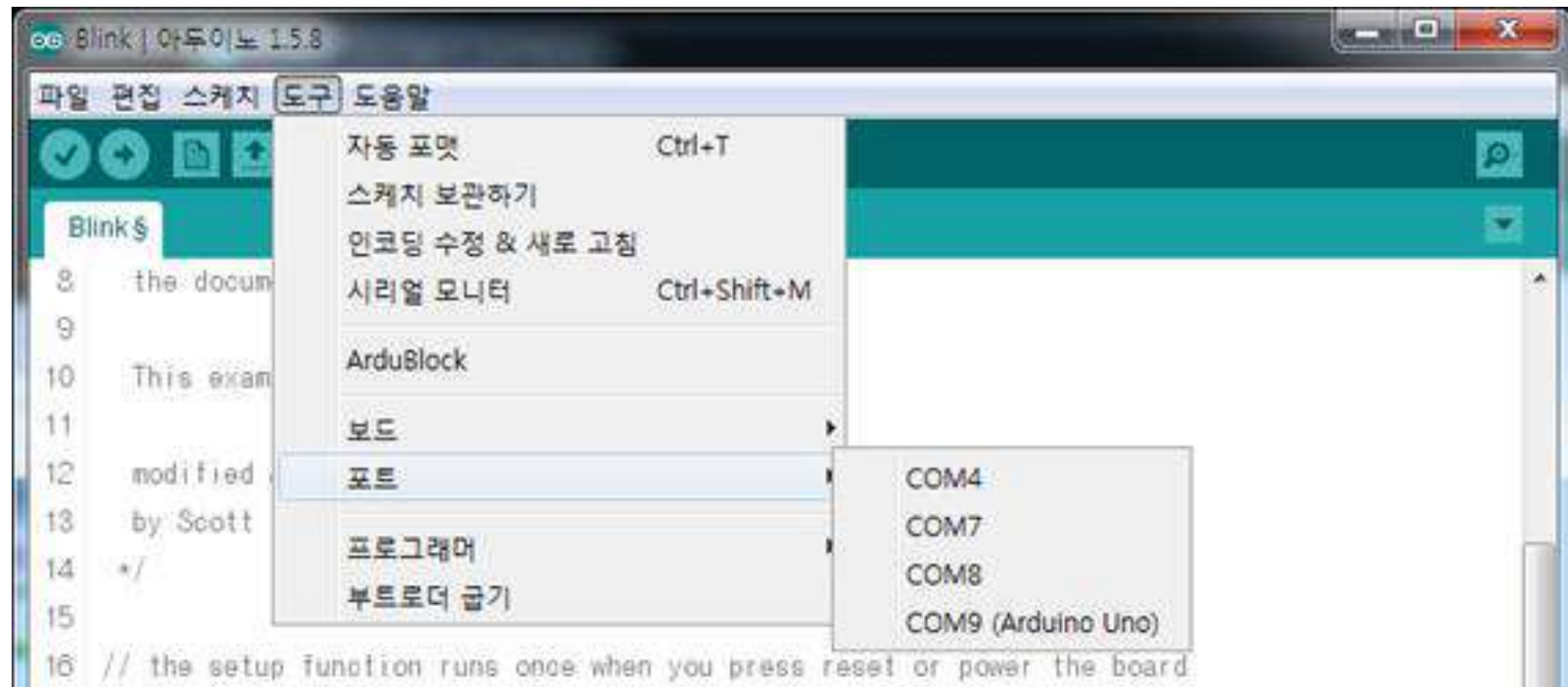
- 아두이노 IDE에서 PC에 연결된 개발 보드의 종류와 포트를 설정
- 도구>보드>**NodeMCU 1.0 (ESP-12E Module)** 선택



# 아두이노 개발 환경 설정

## ◆ 개발 보드의 시리얼 포트 지정

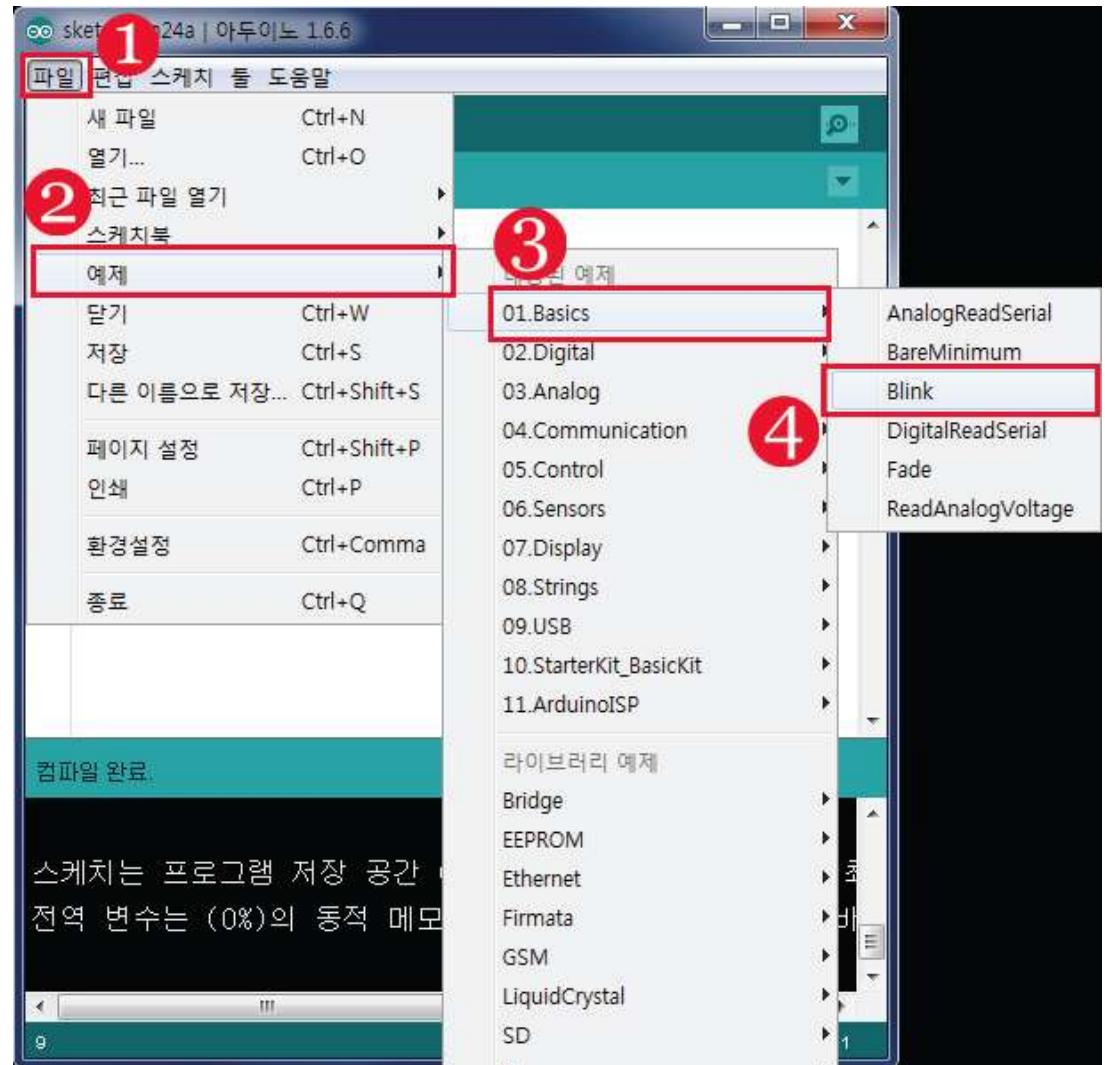
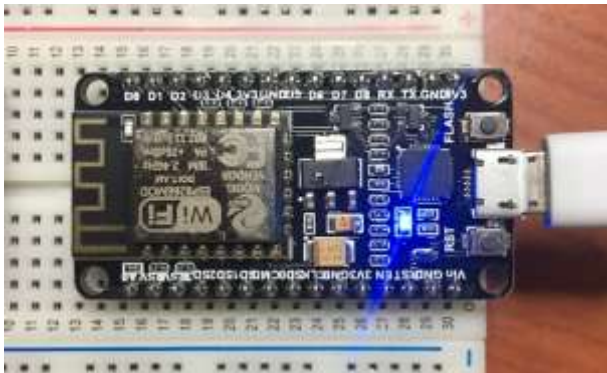
- 도구>포트 순으로 선택 후 개발 보드가 연결된 시리얼 포트를 지정
  - 윈도우의 장치관리자를 실행시킨 후 아래쪽의 포트(COM & LPT)를 선택 후 확인
- 개발 보드와 컴퓨터는 USB 케이블을 이용하여 필요한 데이터를 교환



# 아두이노 개발 환경 (IDE)

## ◆ 정상 동작 확인

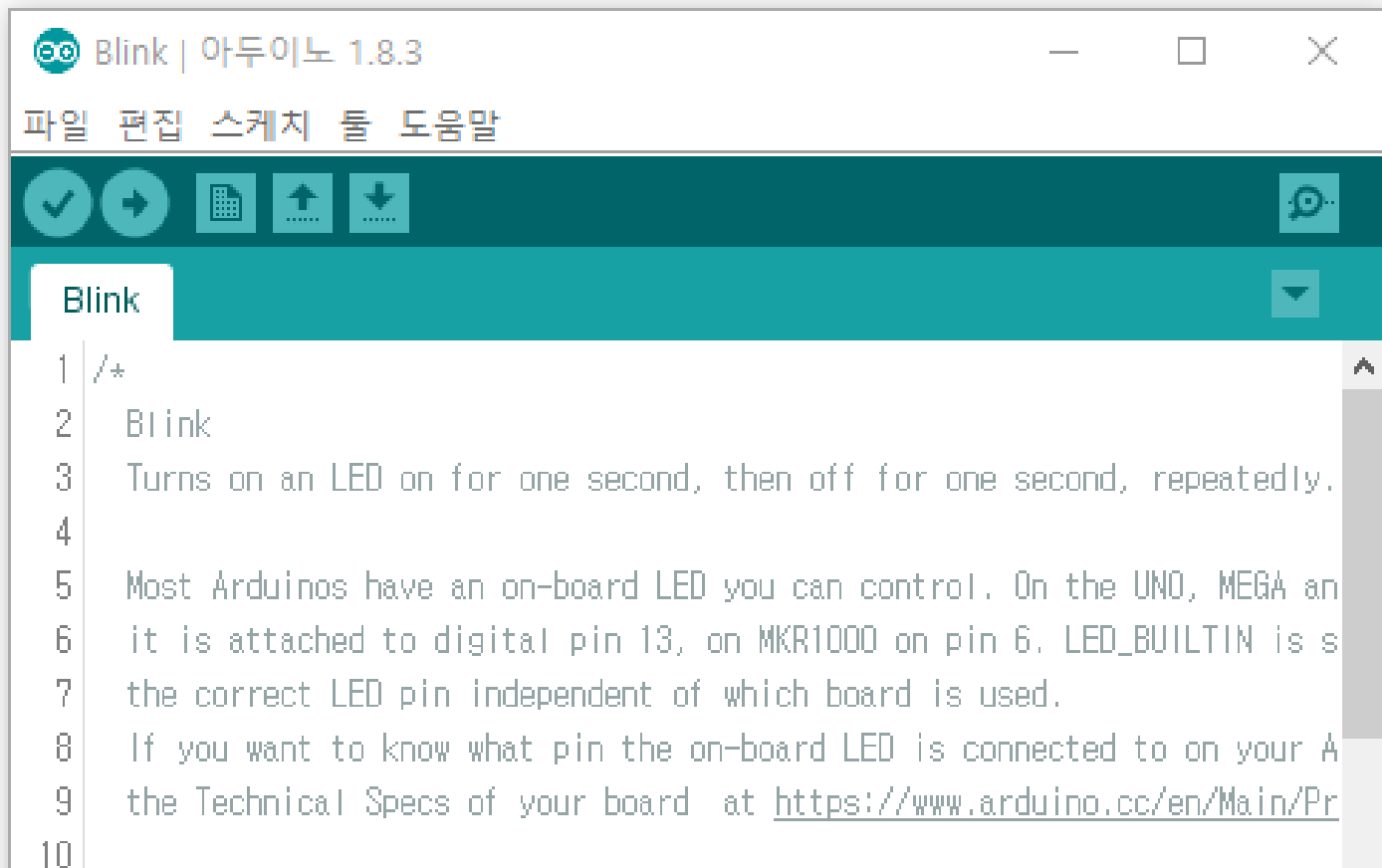
- USB 케이블을 이용하여 아두이노를 PC에 연결
- 오른쪽 그림처럼 예제 선택
- V 버튼으로 컴파일
- → 버튼으로 업로드
- NodeMCU 보드의 LED 점멸



# 아두이노 개발 환경 (IDE)

## ◆ 프로그램 컴파일 및 업로드

- V : 컴파일 버튼
- ➔ : 컴파일 후 바이너리 파일을 NodeMCU로 업로드





# 아두이노 개발 환경 (IDE)

## ◆ Blink 예제 파일 설명

// the setup function runs once when you press reset or power the board

```
void setup() {
```

```
  // initialize digital pin LED_BUILTIN as an output.
```

```
  pinMode(LED_BUILTIN, OUTPUT);
```

```
}
```

// the loop function runs over and over again forever

```
void loop() {
```

```
  digitalWrite(LED_BUILTIN, HIGH); // turn the LED on (HIGH is the voltage level)
```

```
  delay(1000);                      // wait for a second
```

```
  digitalWrite(LED_BUILTIN, LOW);  // turn the LED off by making the voltage LOW
```

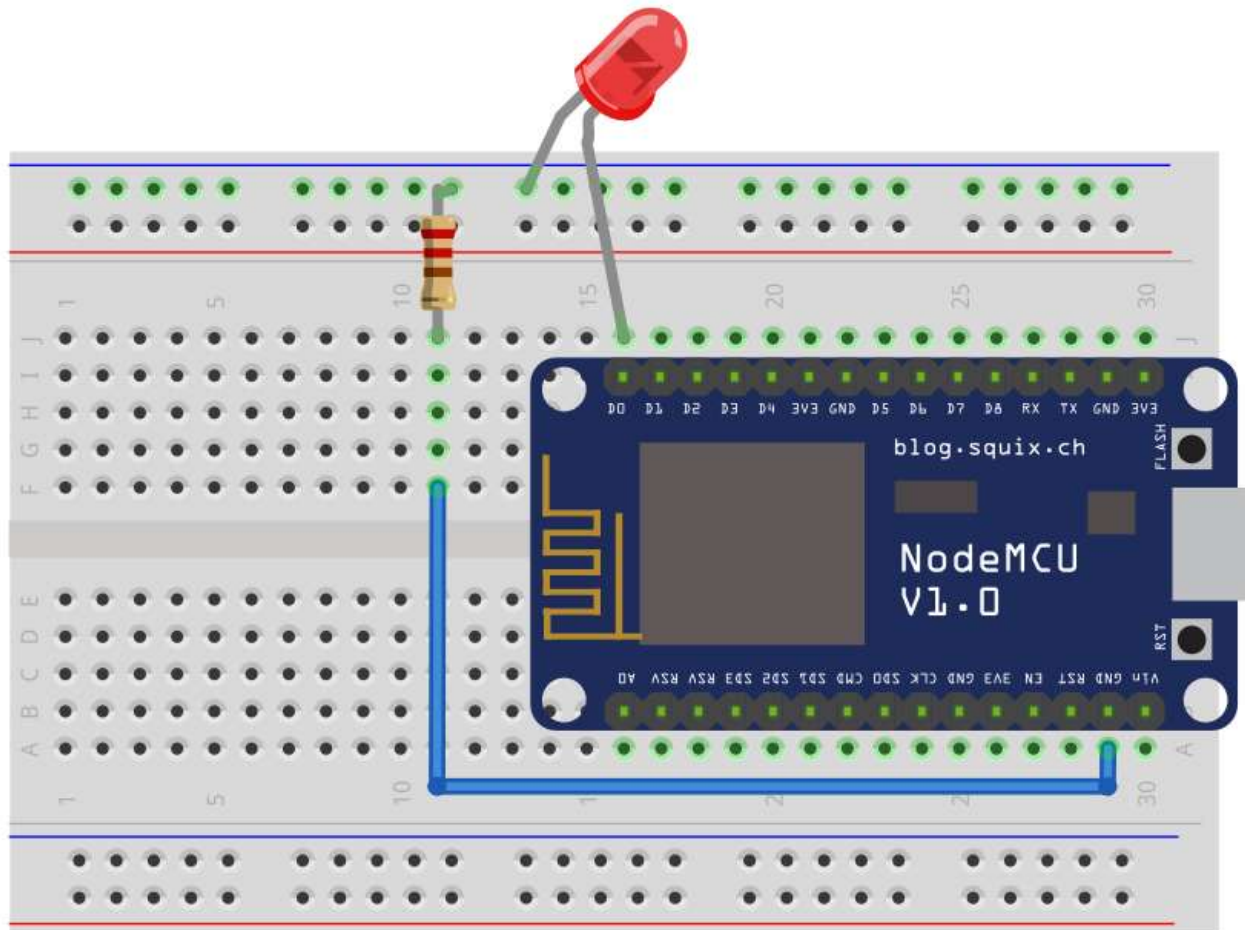
```
  delay(1000);                      // wait for a second
```

```
}
```

# LED 제어하기

## ◆ 그림과 같이 회로를 구성

- LED는 다리가 긴 쪽이 + 극이며, 이를 D0 혹은 다른 GPIO 핀에 연결
- NodeMCU를 연결한 후, 예제 코드(blink)를 다시 컴파일 및 업로드



# LED 제어하기

## ◆ 예제 파일(blink)을 통한 스케치 문법 이해

```
// the setup function runs once when you press reset or power the board
void setup() {
  // initialize digital pin LED_BUILTIN as an output.
  pinMode(LED_BUILTIN, OUTPUT);
}

// the loop function runs over and over again forever
void loop() {
  digitalWrite(LED_BUILTIN, HIGH); // turn the LED on (HIGH is the voltage level)
  delay(1000);                       // wait for a second
  digitalWrite(LED_BUILTIN, LOW);  // turn the LED off by making the voltage LOW
  delay(1000);                       // wait for a second
}
```

# pinMode(pin, mode)

## ◆ pinMode(16, OUTPUT);

- GPIO16 핀, 즉 D0 핀을 출력용으로 사용

*The GPIO Pin Number*

**pinMode**( pinNum, nMode);

*The Mode*

INPUT  
OUTPUT  
INPUT\_PULLUP  
INPUT\_PULLDOWN

## ◆ pinMode(0, INPUT\_PULLUP);

- GPIO0 핀, 즉 D3 핀을 PULLUP Input으로 사용
- PULLUP Input이라는 것은 지정된 핀에 아무 것도 연결되어 있지 않지만, 이를 HIGH 상태로 보겠다는 의미

# digitalRead(pin), digitalWrite(pin, value)

## ◆ digitalRead(switchPin);

- 변수 switchPin 값에 해당하는 핀에서 디지털 신호값을 읽어들이

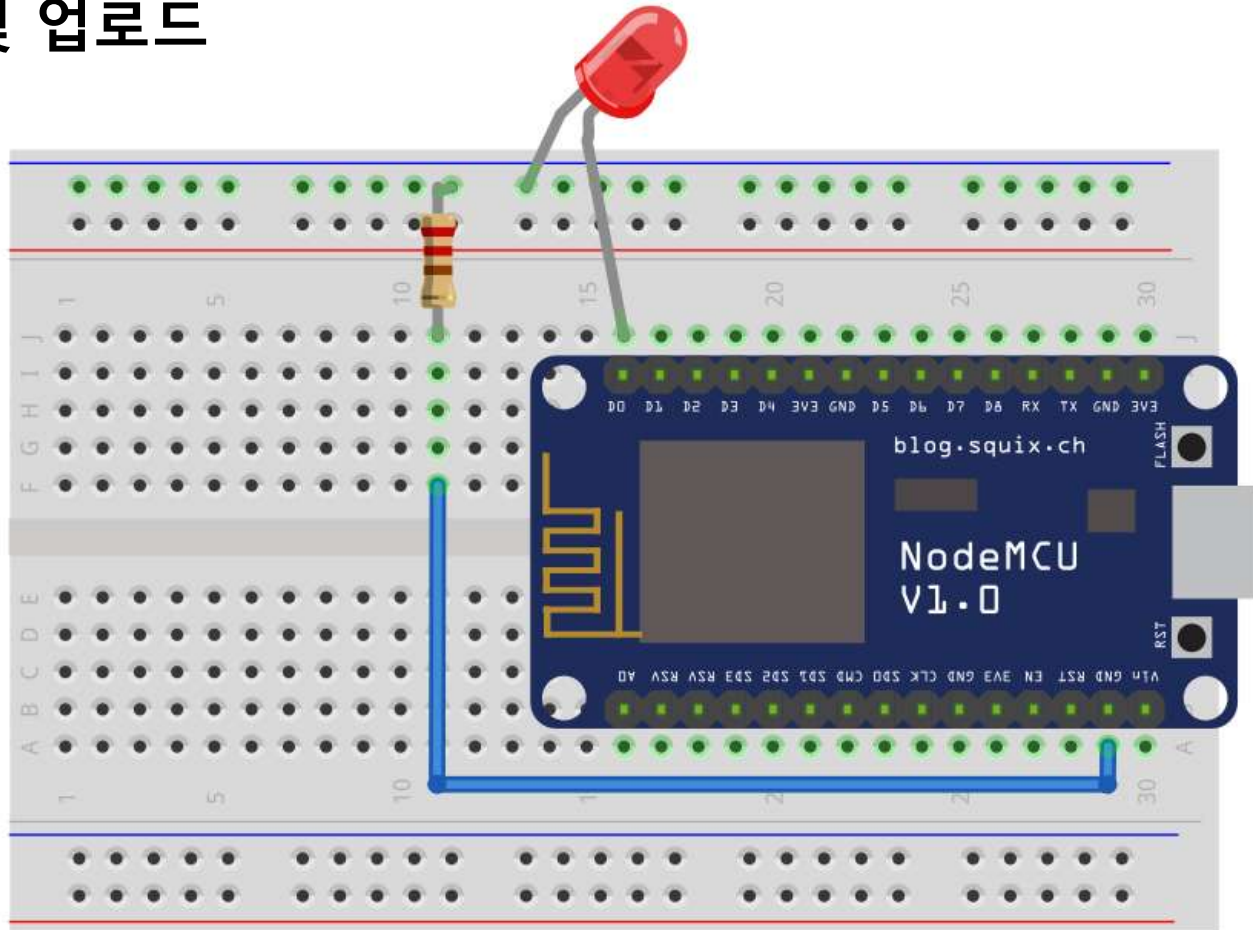
## ◆ digitalWrite(LedPin, switchValue);

- 변수 LedPin 값에 해당하는 핀에 변수 switchValue의 디지털 값을 내보냄
- 만약 switchValue의 반대 값을 내보내려면 앞에 !를 붙여서 !switchValue를 이용



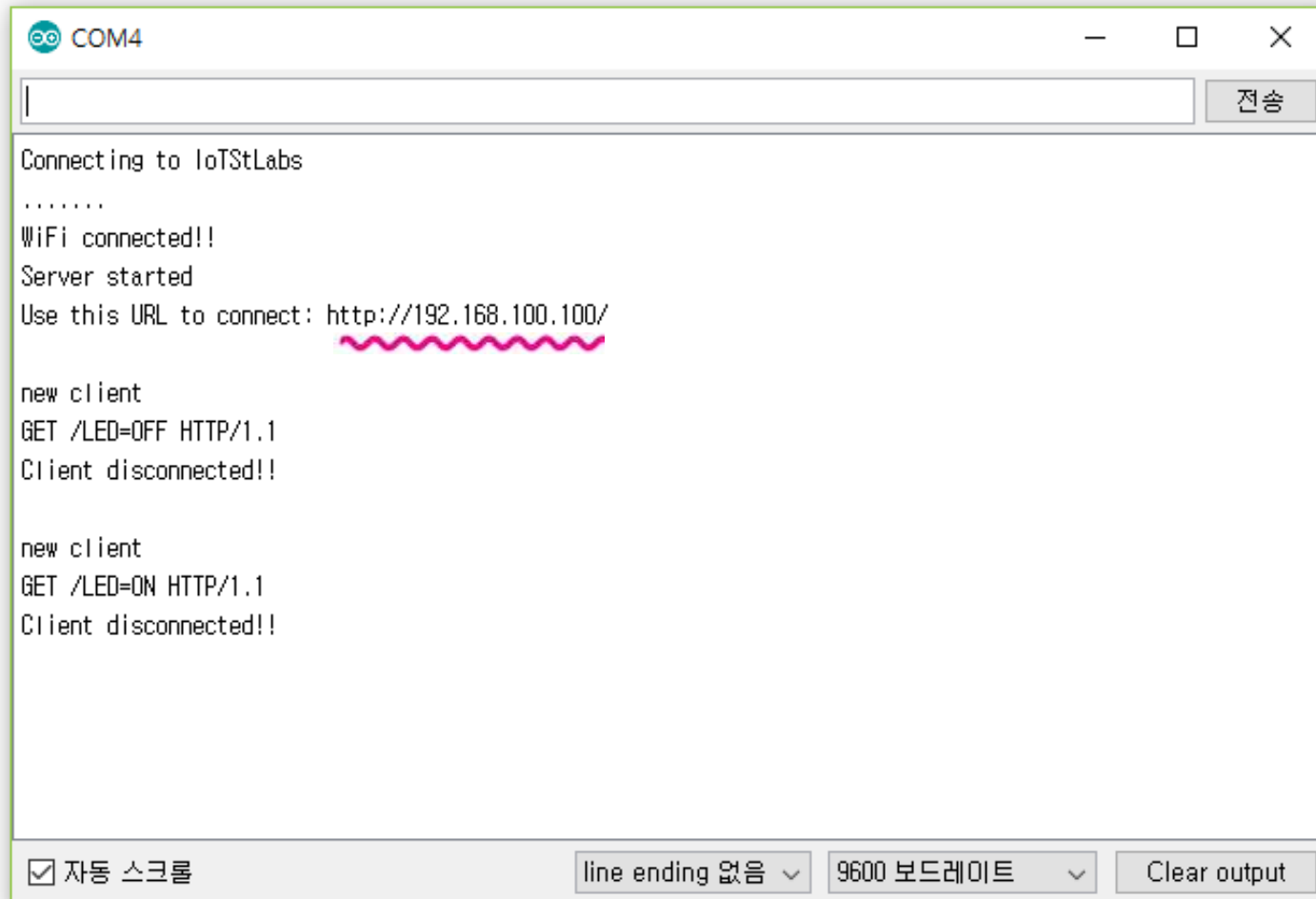
# 원격에서 LED 제어하기

- ◆ 그림과 같이 회로를 구성 (이전 회로와 동일함)
- ◆ 아두이노 IDE에서 다운로드 받은 WebServer1LED.ino 열기
- ◆ 컴파일 및 업로드



# 원격에서 LED 제어하기

## ◆ 아두이노 IDE의 도구>시리얼 모니터 선택



# 원격에서 LED 제어하기

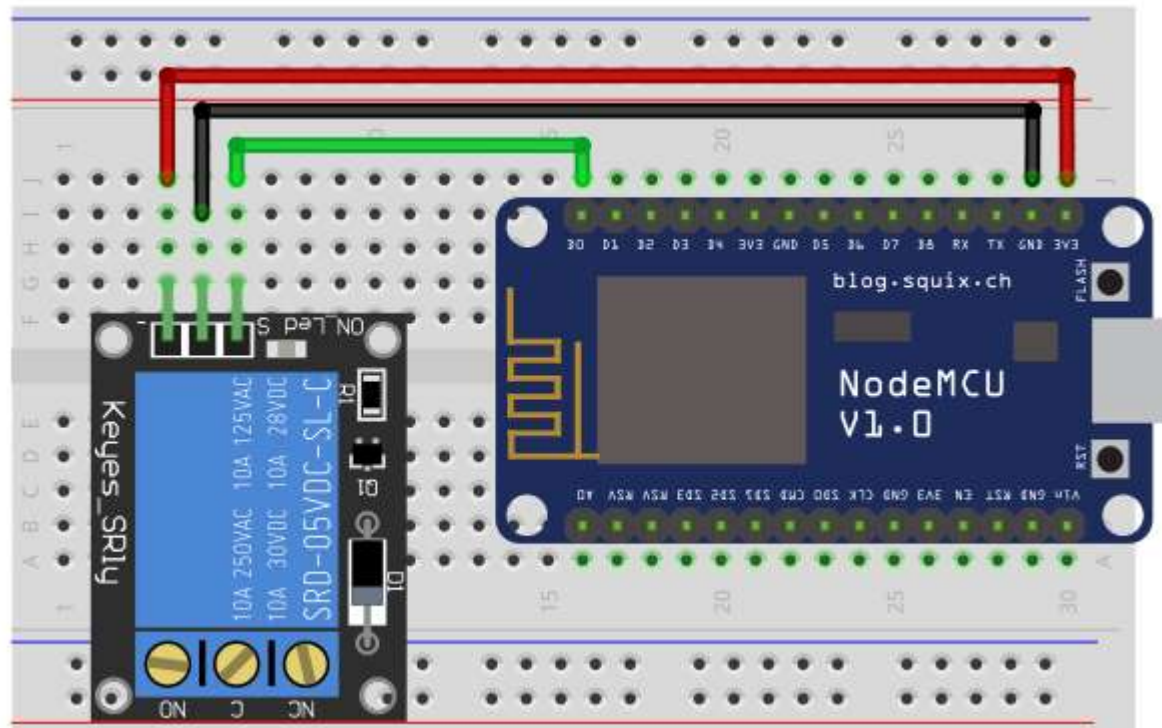
- ◆ 스마트폰 웹 브라우저를 이용해서 192.168.x.x 에 접속
- ◆ 정상적으로 화면이 뜨면, Turn On 혹은 Turn Off 버튼 클릭
  - NodeMCU에 연결된 LED의 상태 변화 확인



# 스마트 플러그 만들기

## ◆ 이전 코드를 그대로 유지한 상태에서 그림과 같이 회로를 구성

- 빨간 선으로 릴레이의 Vcc와 NodeMCU의 3V3을 연결
- 검정 선으로 릴레이의 GND와 NodeMCU의 GND와 연결
- 녹색 선으로 릴레이의 In1과 NodeMCU의 D0 연결



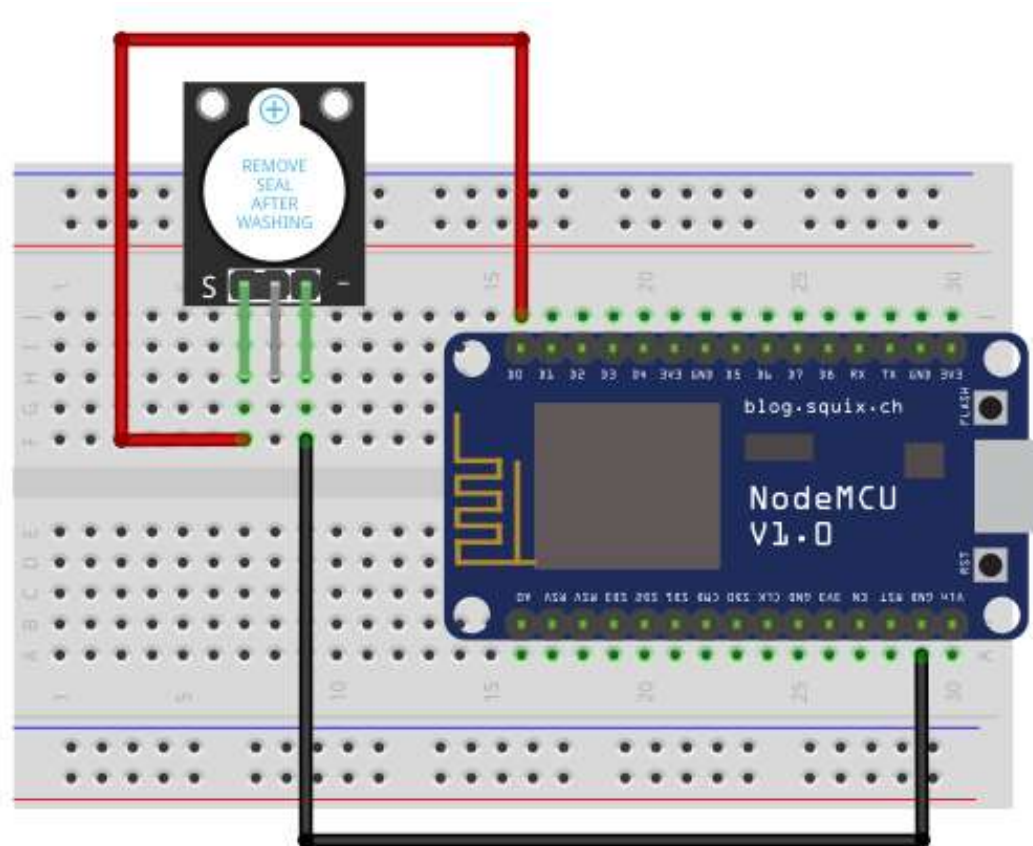
# 스마트 플러그 만들기

- ◆ 릴레이에 LED와  $220\Omega$  저항을 이용하여 스탠드를 구성
  - 릴레이의 딸깍거리는 소리로 동작 확인



# Active Buzzer 이용해 보기

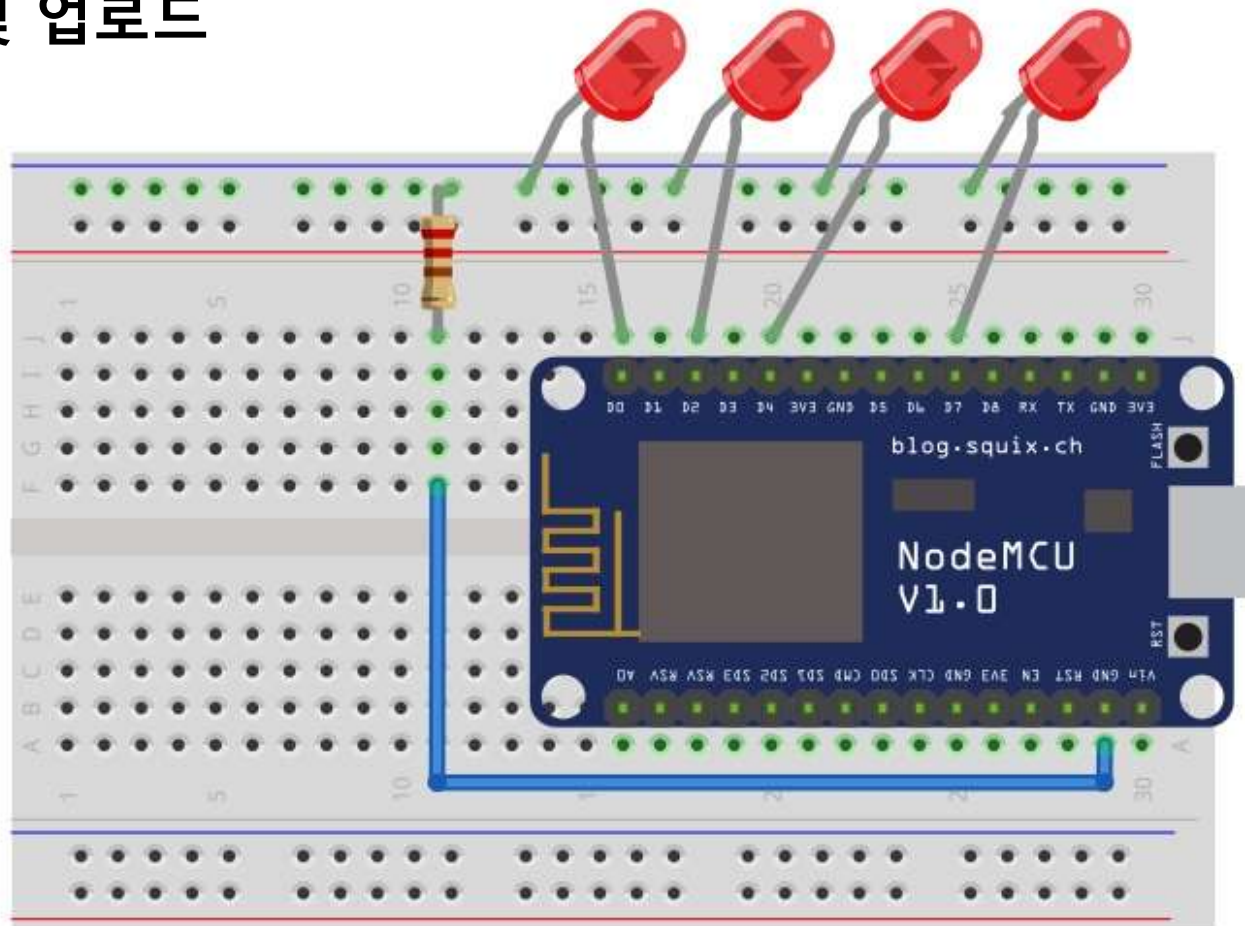
- ◆ 이전 코드를 그대로 유지한 상태에서 그림과 같이 회로를 구성
  - 부저의 긴 다리가 (+)이며 D0에 연결
  - 부저의 짧은 다리(-)는 GND 아무 곳이나 연결





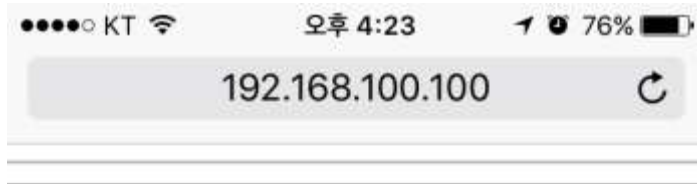
# 원격에서 여러 개의 LED 제어하기

- ◆ 그림과 같이 회로를 구성
- ◆ 아두이노 IDE에서 다운로드 받은 WebServer4LEDs.ino 열기
- ◆ 컴파일 및 업로드

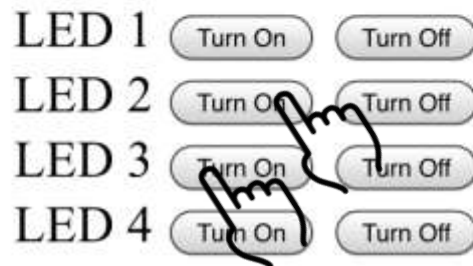


# 원격에서 여러 개의 LED 제어하기

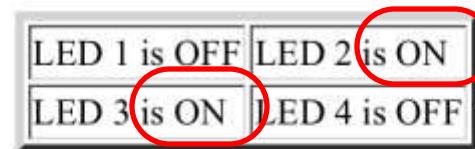
- ◆ 아두이노 IDE의 도구>시리얼 모니터 선택
- ◆ 스마트폰 웹 브라우저를 이용해서 192.168.x.x 에 접속
- ◆ 정상적으로 화면이 뜨면, LED 1~4의 Turn On 혹은 Turn Off 클릭



## Web Control of 4 LEDs



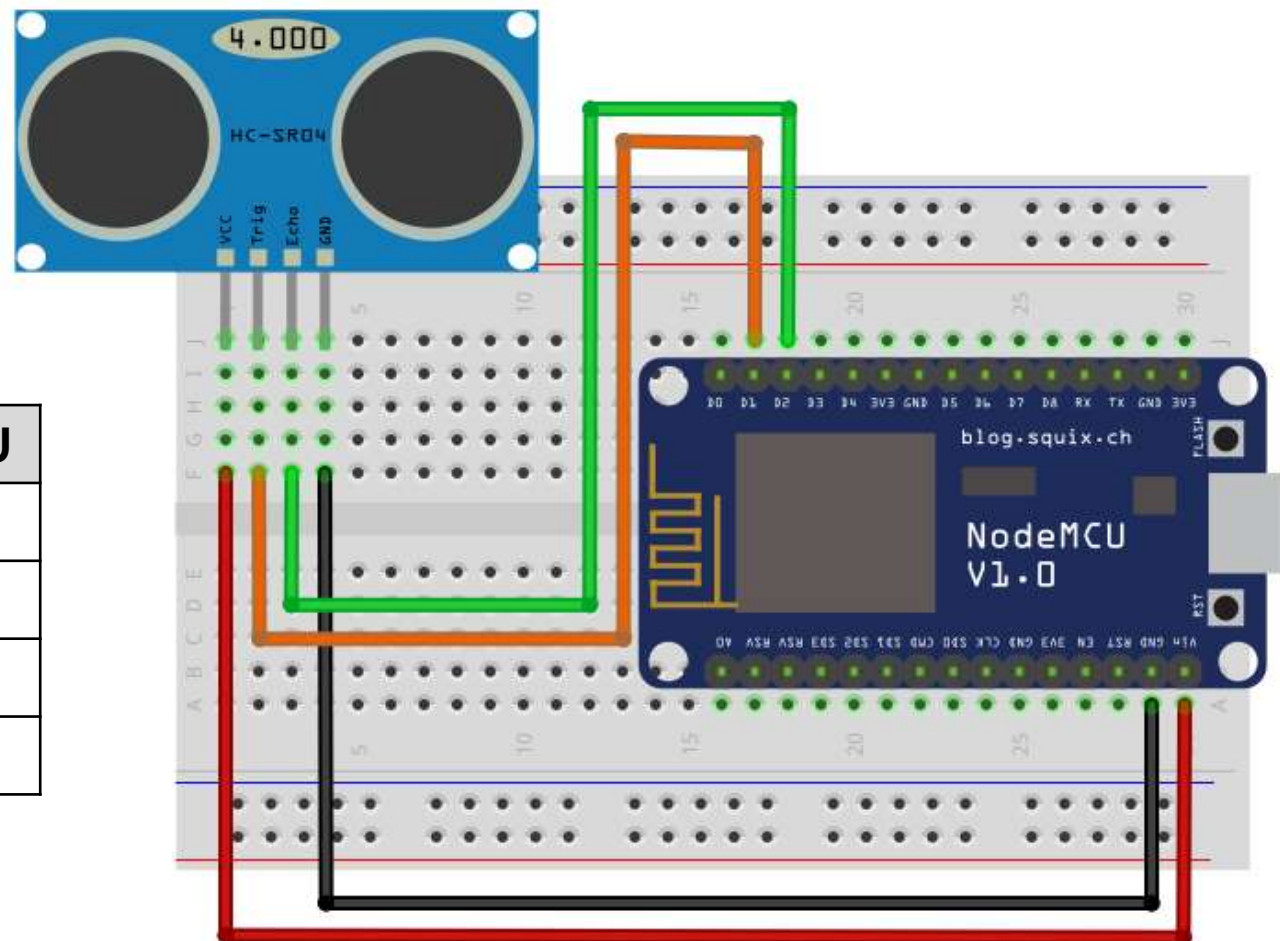
## Web Control of 4 LEDs



# 초음파 거리 센서(SR04) 이용하기

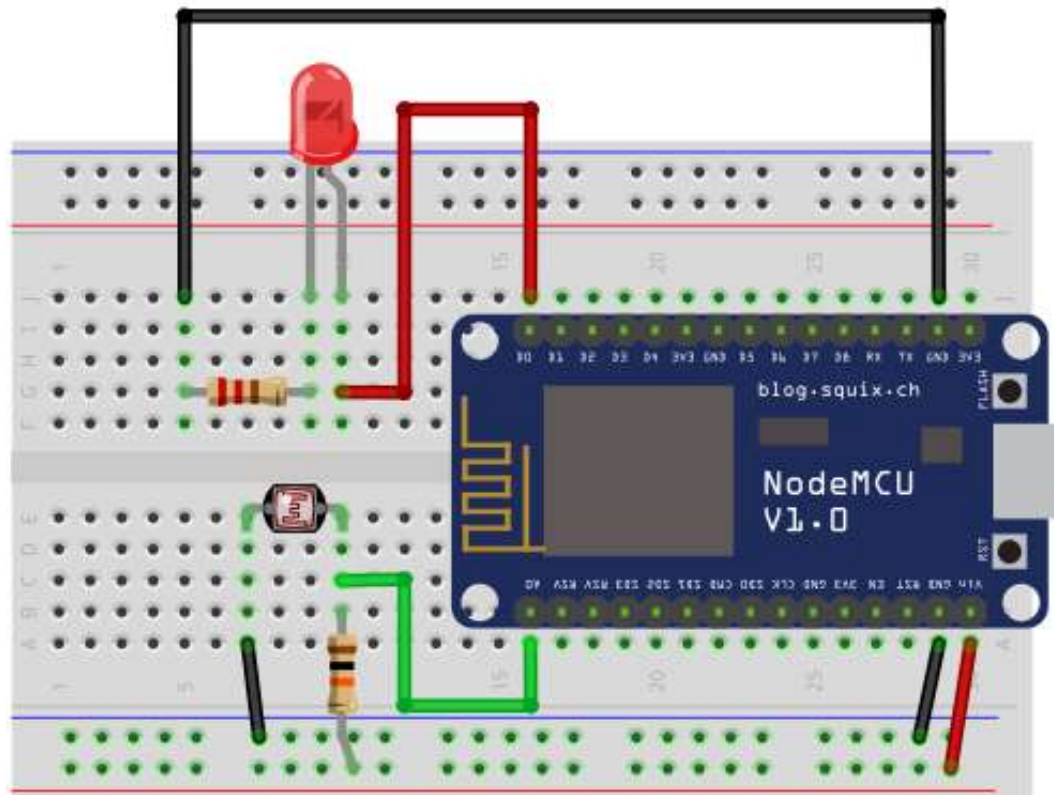
- ◆ 아두이노 IDE에서 SR04\_distance.ino 열기 → 디지털 센서
- ◆ 그림과 같이 회로를 구성 & 아두이노 IDE에서 시리얼 모니터 열기

| SR-04 | NodeMCU |
|-------|---------|
| Vcc   | 3V3     |
| Trig  | D1      |
| Echo  | D2      |
| GND   | CND     |



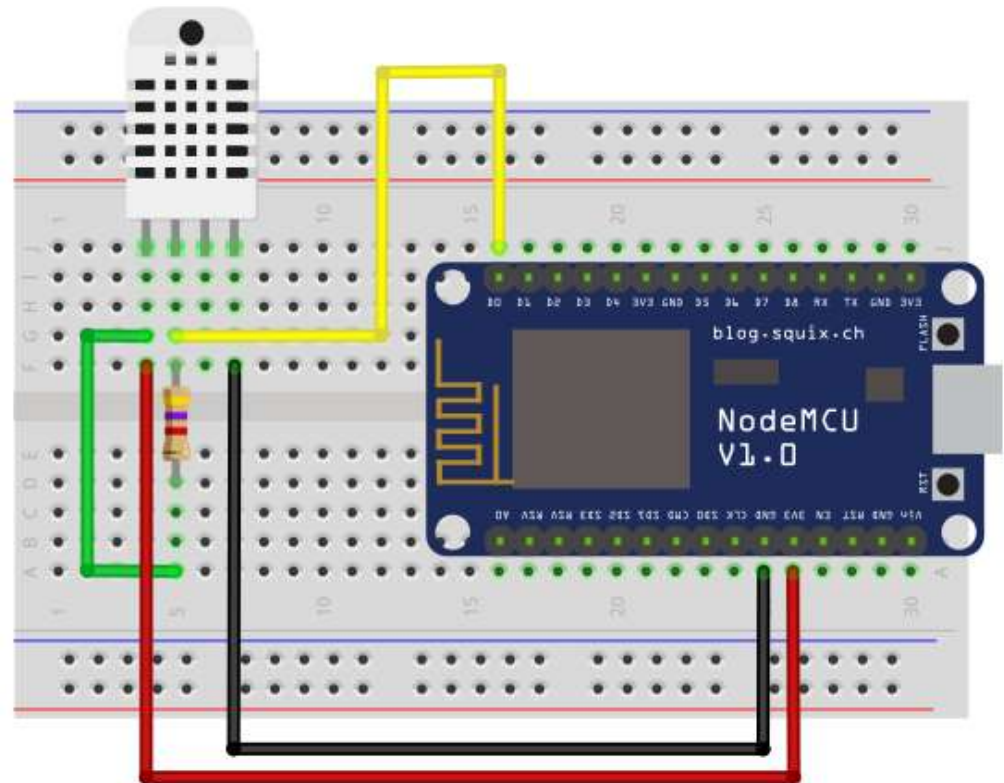
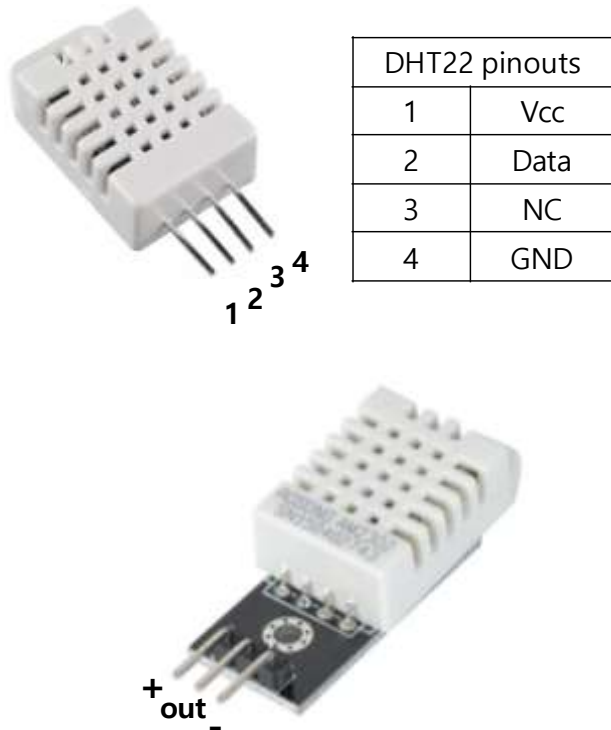
# 조도 센서(CDS)를 이용해서 스마트 가로등 만들기

- ◆ 아두이노 IDE에서 다운로드 받은 CDS.ino 열기 → 아날로그 센서
- ◆ 그림과 같이 회로를 구성 & 아두이노 IDE에서 시리얼 모니터 열기
- ◆ 손으로 CDS를 가리면서 숫자의 변화를 확인하면서 LED를 켜보세요.



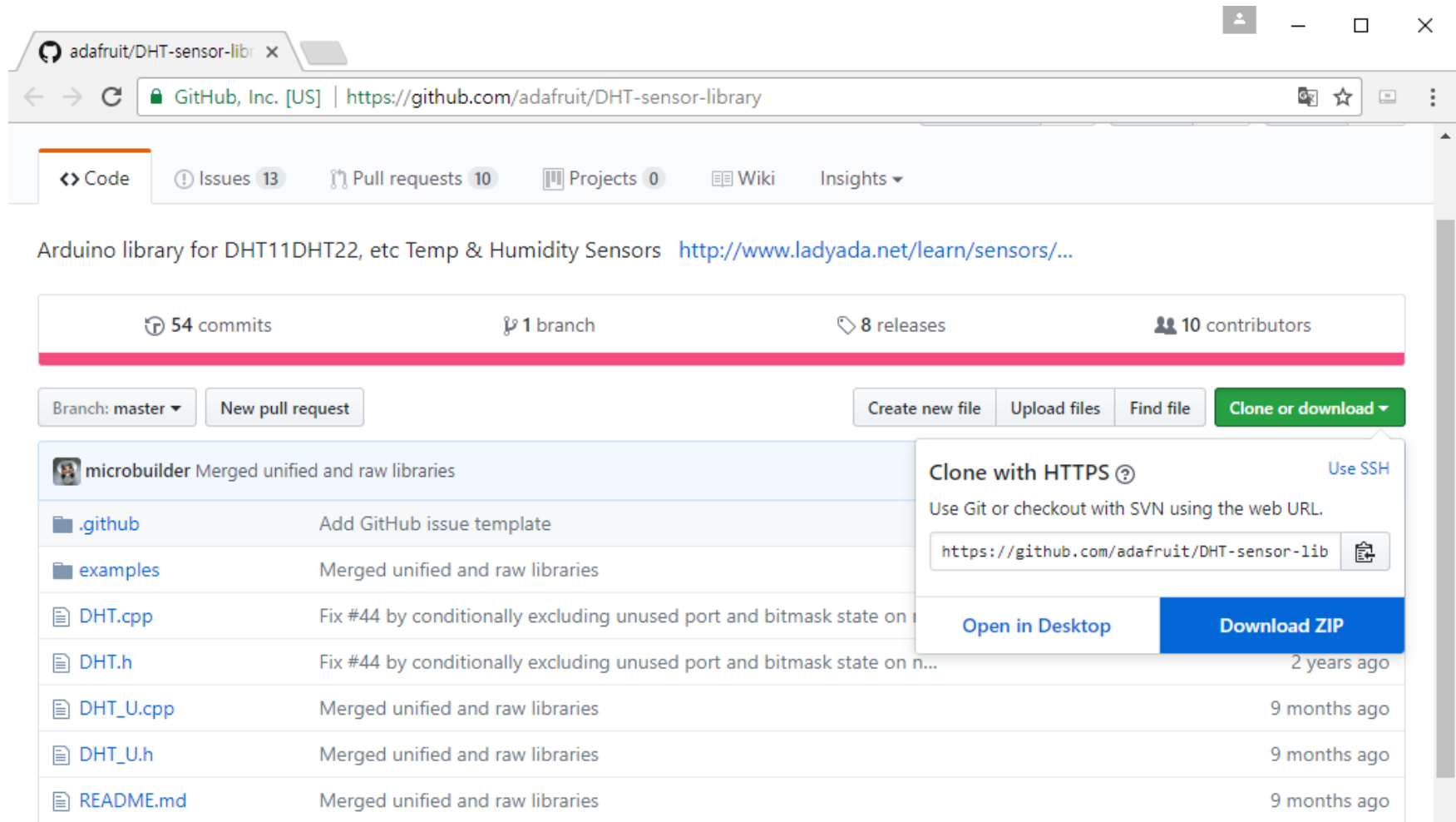
# 스마트 온습도 측정기 (1)

- ◆ 그림과 같이 회로를 구성
- ◆ 아두이노 IDE에서 다운로드 받은 Temp\_Humid\_Basic.ino 열기
- ◆ DHT22 라이브러리 다운로드 후 컴파일 및 업로드



# Library 설치 방법

- ◆ 구글에서 필요한 라이브러리 이름으로 검색 후 다운로드
- ◆ 가능하면 Github 등에서 ZIP 형태로 다운로드



adafruit/DHT-sensor-lib x

GitHub, Inc. [US] | <https://github.com/adafruit/DHT-sensor-library>

<> Code Issues 13 Pull requests 10 Projects 0 Wiki Insights

Arduino library for DHT11DHT22, etc Temp & Humidity Sensors <http://www.ladyada.net/learn/sensors/...>

54 commits 1 branch 8 releases 10 contributors

Branch: master New pull request Create new file Upload files Find file Clone or download

microbuilder Merged unified and raw libraries

|           |  |              |
|-----------|--|--------------|
| .github   | Add GitHub issue template  |              |
| examples  | Merged unified and raw libraries   |              |
| DHT.cpp   | Fix #44 by conditionally excluding unused port and bitmask state on      |              |
| DHT.h     | Fix #44 by conditionally excluding unused port and bitmask state on n... | 2 years ago  |
| DHT_U.cpp | Merged unified and raw libraries   | 9 months ago |
| DHT_U.h   | Merged unified and raw libraries   | 9 months ago |
| README.md | Merged unified and raw libraries   | 9 months ago |

Clone with HTTPS ? Use SSH

Use Git or checkout with SVN using the web URL.

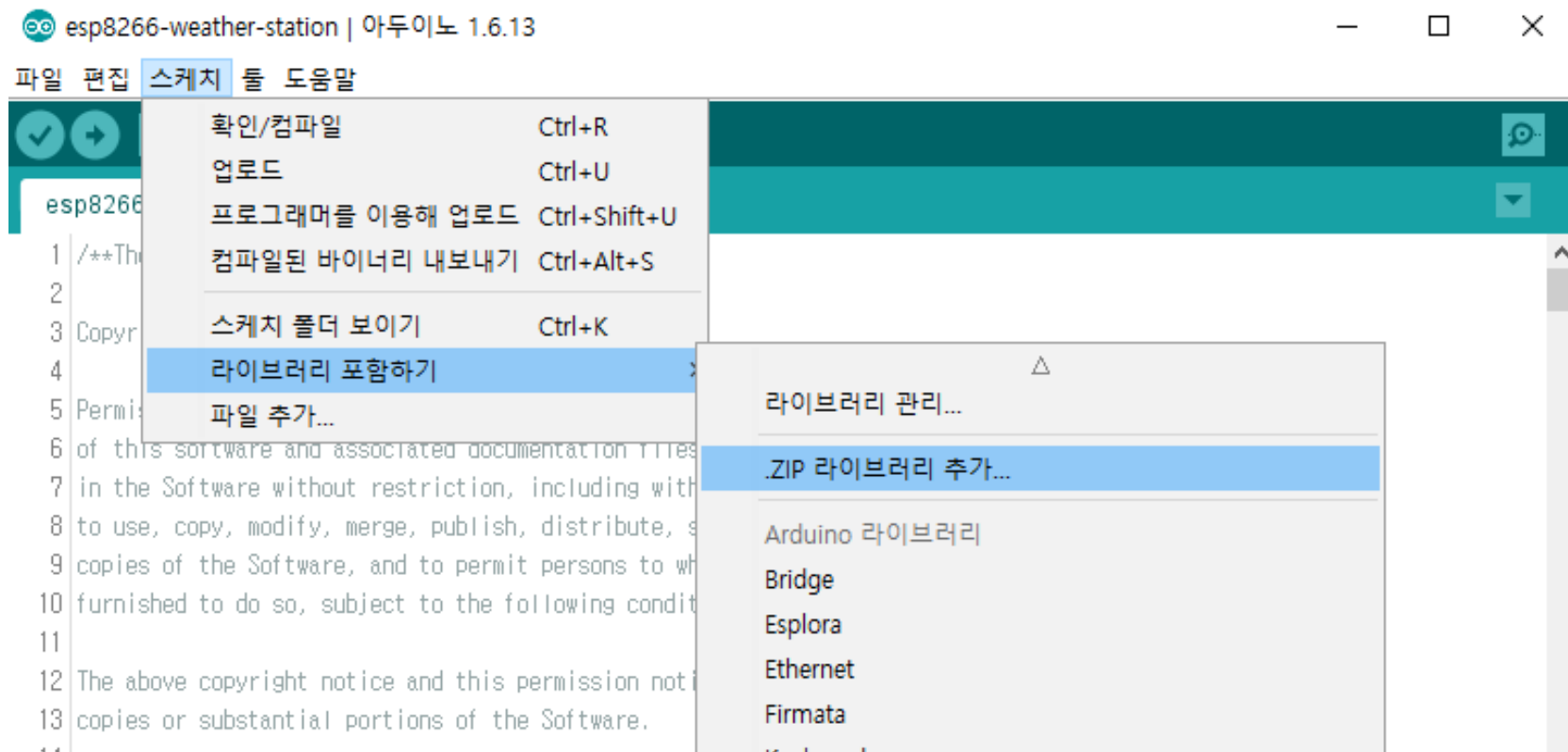
<https://github.com/adafruit/DHT-sensor-lib>

Open in Desktop Download ZIP



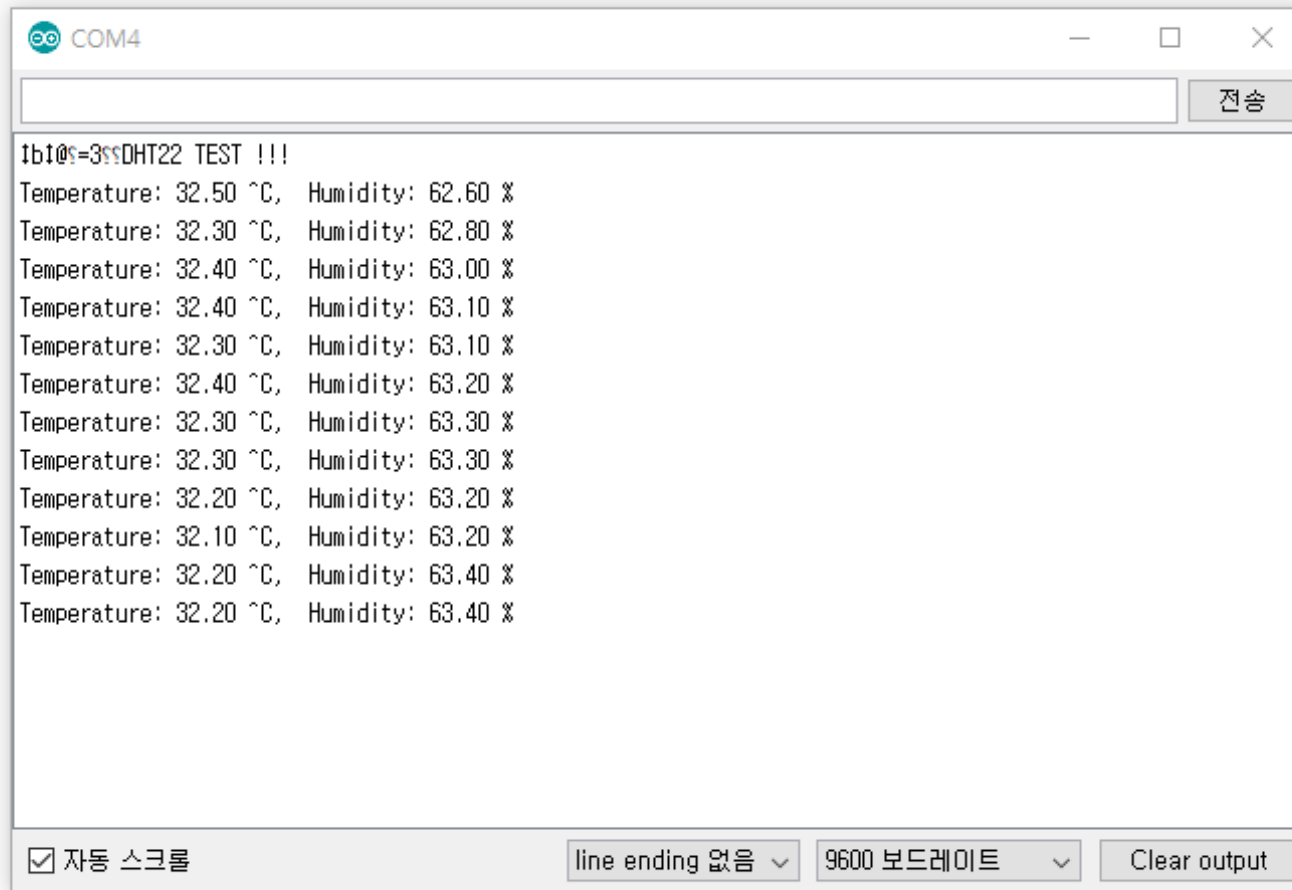
# Library 설치 방법

- ◆ Arduino IDE의 스케치 – 라이브러리 포함하기 - .ZIP 라이브러리 추가
  - 파일 탐색기에서 다운로드 받은 .ZIP 파일을 선택 (DHT-sensor-library-master.zip)
  - Adafruit\_Sensor-master.zip 파일 검색, 다운로드, 설치



# 스마트 온습도 측정기 (1)

## ◆ 시리얼 터미널을 열어서 온습도의 변화 확인

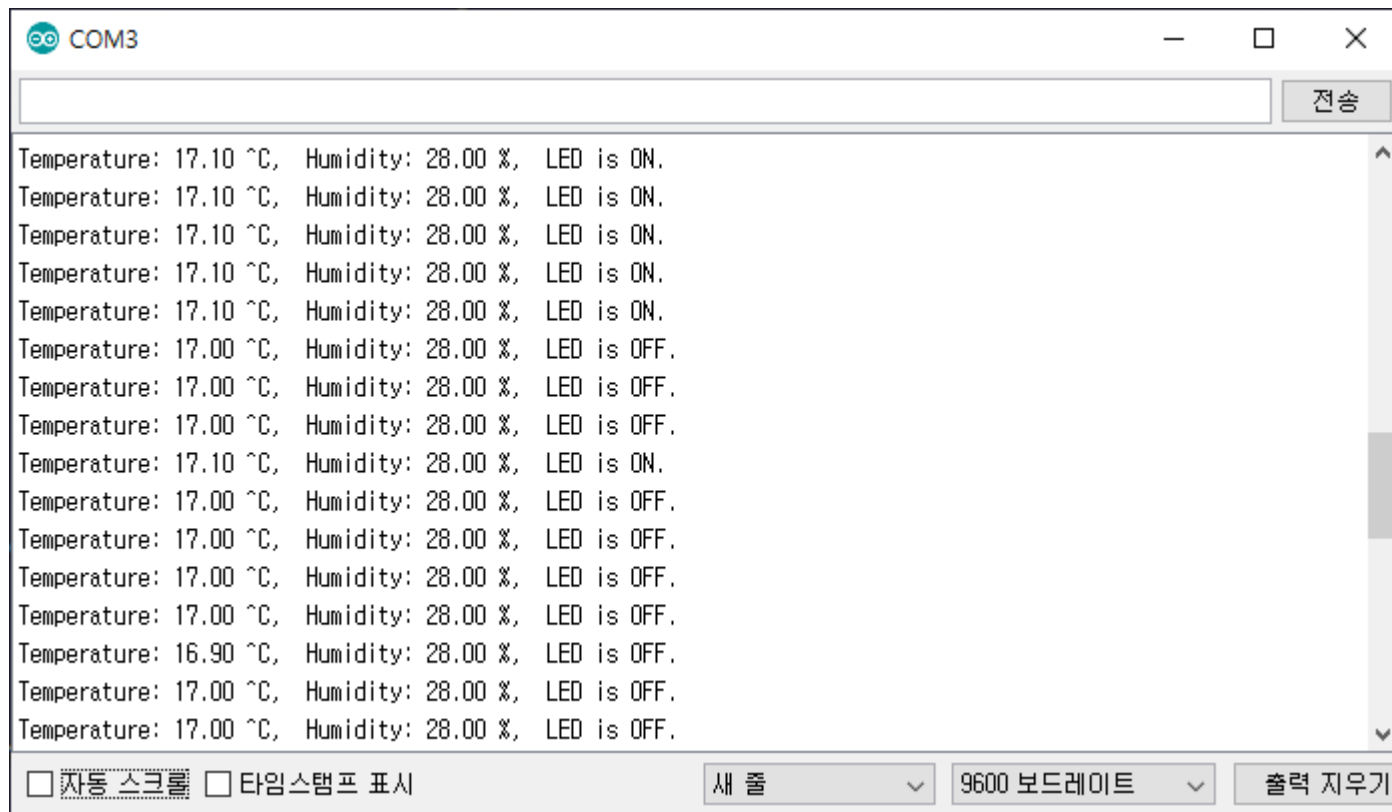


## 스마트 온습도 측정기 (2)

- ◆ 기존 코드를 변형하여 일정한 온도값 이상이면 LED가 켜지고  
이하면 LED가 꺼지는 코드를 작성하시오.
- ◆ DHT22 OUT은 D1에, LED는 D0에 연결하시오.

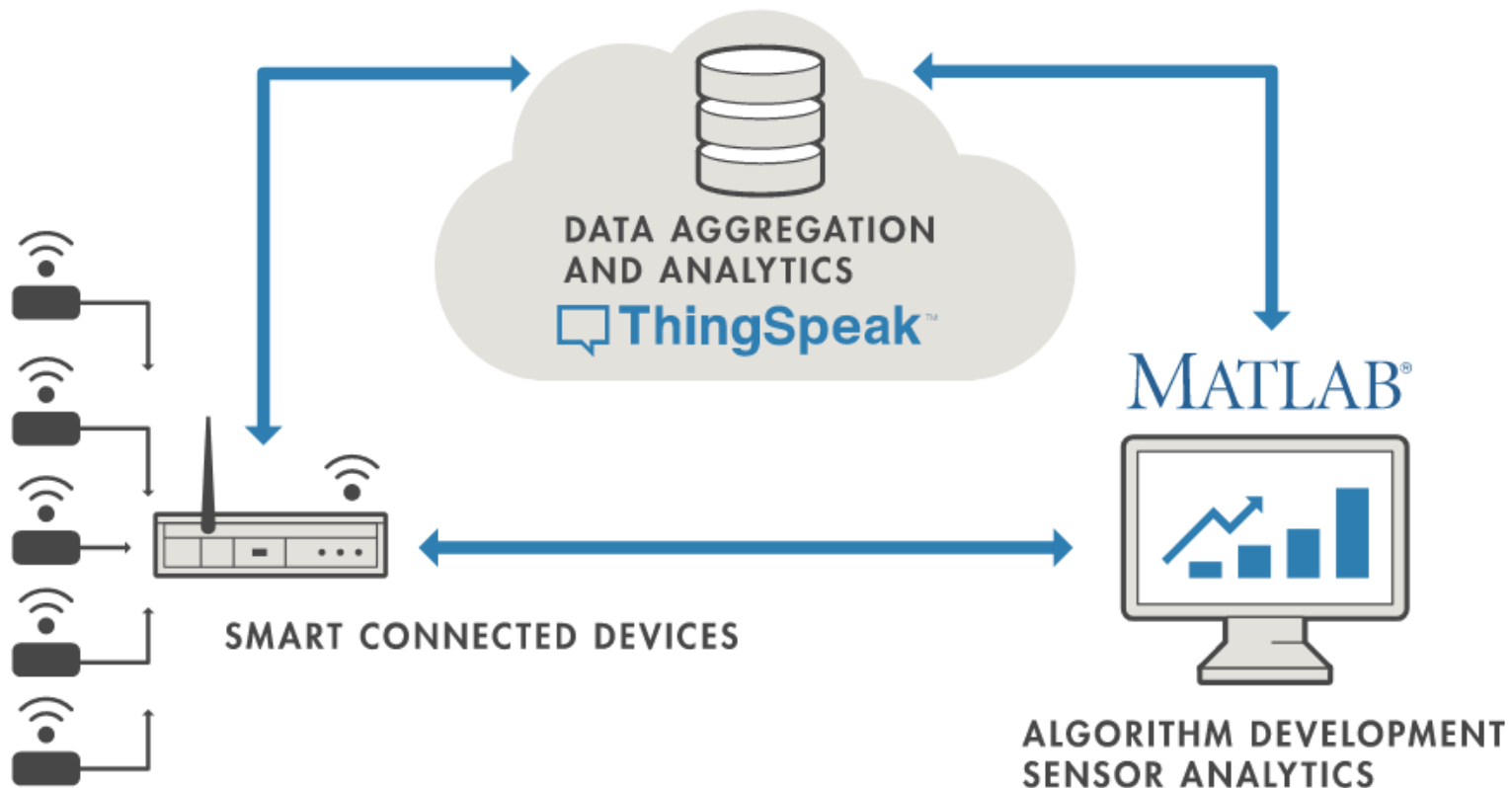
# 스마트 온습도 측정기 (2)

- ◆ 시리얼 터미널을 열어서 온습도의 변화 및 LED 상태 확인
- ◆ 이 경우에는 17도 초과인 경우에만 LED가 ON이 됨



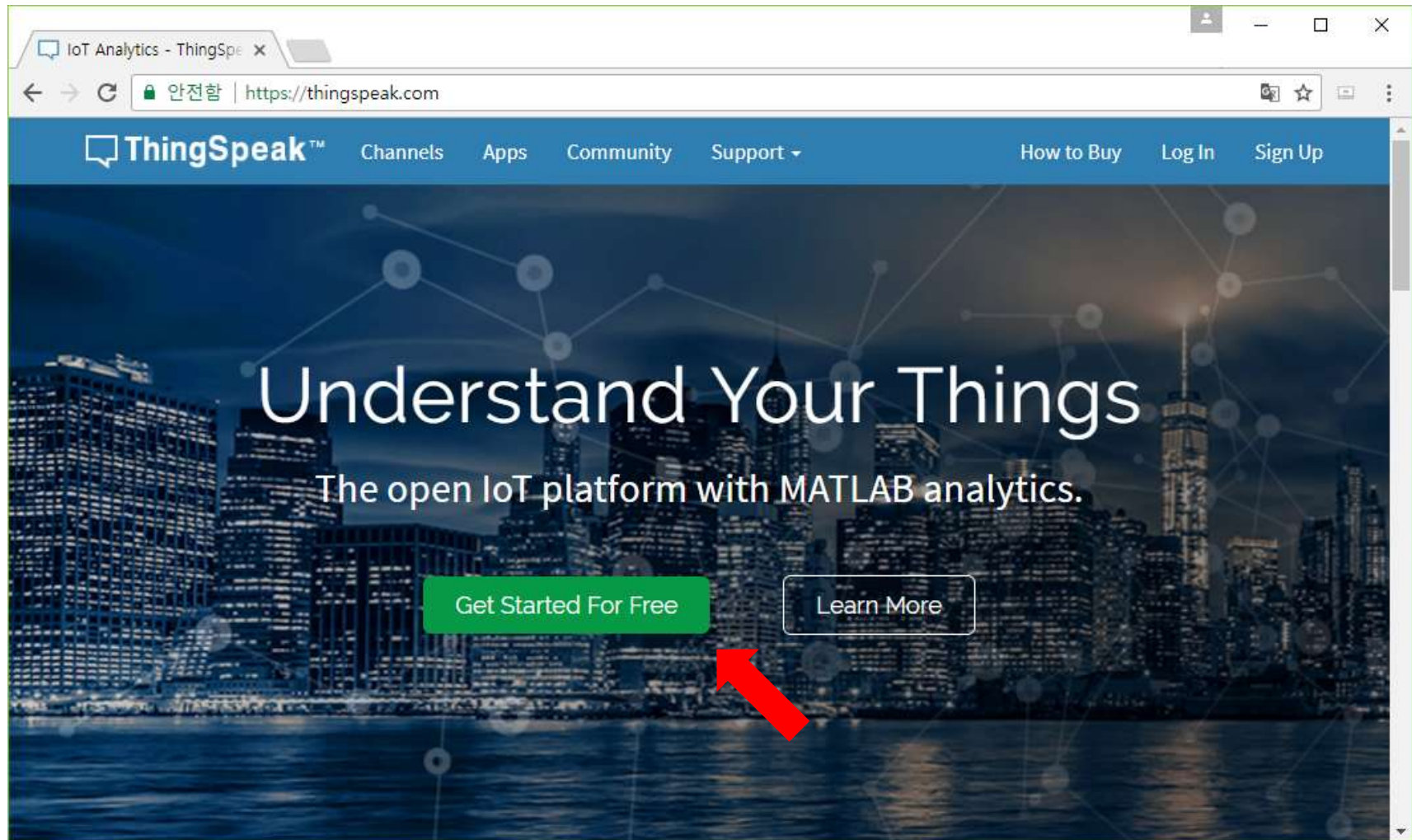
# Open IoT Platform

- ◆ 사물들이 생성한 데이터를 바탕으로 서비스를 생성
- ◆ 사용자 명령이나 사전에 정의된 룰셋에 따라 사물들을 제어
- ◆ 사물들의 연결을 통한 서비스 정의 등



# IoT Platform 가입 및 API Key 받기

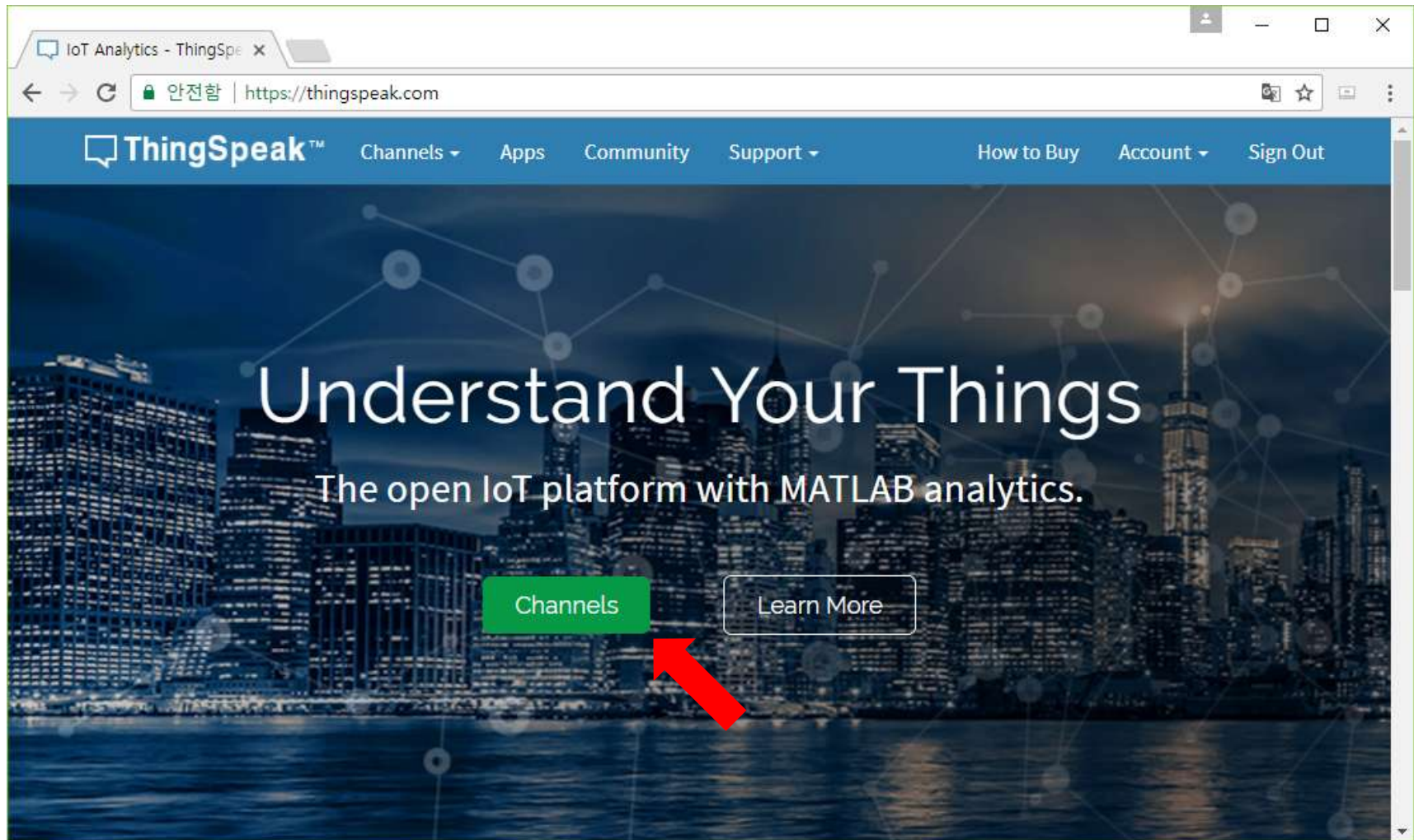
## ◆ Thingspeak.com 접속 후 회원 가입(Get Started for Free or Sign Up)





# IoT Platform 가입 및 API Key 받기

## ◆ 로그인 후 새로운 채널 만들기



# IoT Platform 가입 및 API Key 받기

## ◆ 새로운 채널 생성 (2개의 필드 생성)

Channels - ThingSpeak | x

← → ↻ | <https://thingspeak.com/channels/new> | ☆

ThingSpeak™ Channels Apps Community Support How to Buy Account Sign Out

### New Channel

Name: Temp & Humid

Description: 온습도 데이터 생성 및 플랫폼 전송 실습

Field 1: Temperature ☒

Field 2: Humidity ☒

Field 3:  ☐

Field 4:  ☐

Field 5:  ☐

Field 6:  ☐

### Help

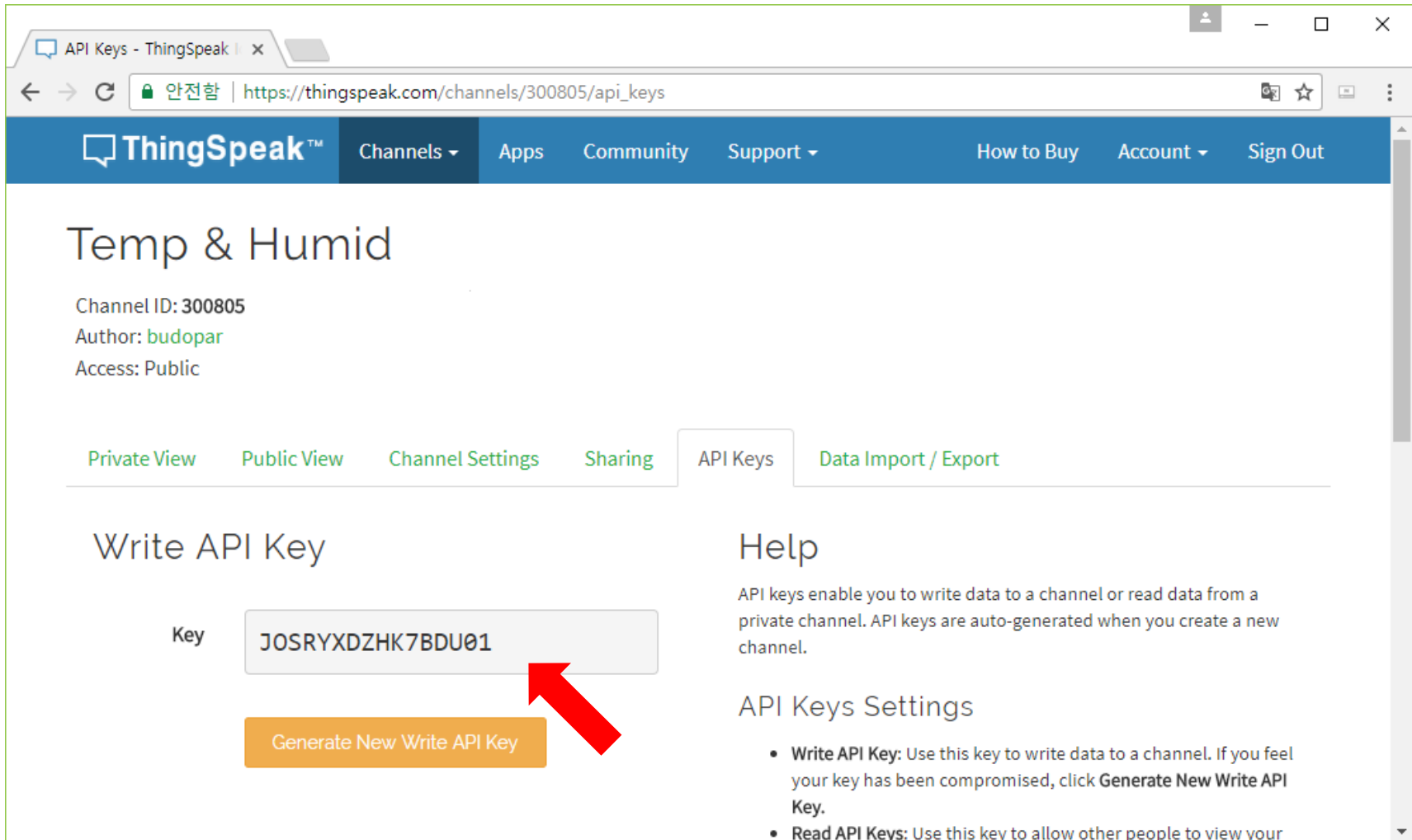
Channels store all the data that a ThingSpeak application collects. Each channel includes eight fields that can hold any type of data, plus three fields for location data and one for status data. Once you collect data in a channel, you can use ThingSpeak apps to analyze and visualize it.

#### Channel Settings

- **Channel Name:** Enter a unique name for the ThingSpeak channel.
- **Description:** Enter a description of the ThingSpeak channel.
- **Field#:** Check the box to enable the field, and enter a field name. Each ThingSpeak channel can have up to 8 fields.
- **Metadata:** Enter information about channel data, including JSON, XML, or CSV data.
- **Tags:** Enter keywords that identify the channel. Separate tags with commas.
- **Latitude:** Specify the position of the sensor or thing that collects data in decimal degrees. For example, the latitude of the city of London is 51.5072.

# IoT Platform 가입 및 API Key 받기

## ◆ 채널 메뉴의 API Keys 탭에서 Write API Key 값 복사



The screenshot shows the ThingSpeak website interface for a specific channel. The browser address bar indicates the URL is [https://thingspeak.com/channels/300805/api\\_keys](https://thingspeak.com/channels/300805/api_keys). The page title is "Temp & Humid" with Channel ID 300805, Author budopar, and Access: Public. The navigation bar includes links for Channels, Apps, Community, Support, How to Buy, Account, and Sign Out. The main content area has tabs for Private View, Public View, Channel Settings, Sharing, API Keys (selected), and Data Import / Export. Under the "Write API Key" section, the key "JOSRYXDZHK7BDU01" is displayed in a text box. A red arrow points to this key. Below the key is a button labeled "Generate New Write API Key". To the right, there is a "Help" section explaining API keys and an "API Keys Settings" section with instructions on how to use the keys.

API Keys - ThingSpeak

← → ↻ | [https://thingspeak.com/channels/300805/api\\_keys](https://thingspeak.com/channels/300805/api_keys)

ThingSpeak™ Channels Apps Community Support How to Buy Account Sign Out

Temp & Humid

Channel ID: 300805  
Author: budopar  
Access: Public

Private View Public View Channel Settings Sharing API Keys Data Import / Export

Write API Key

Key JOSRYXDZHK7BDU01

Generate New Write API Key

Help

API keys enable you to write data to a channel or read data from a private channel. API keys are auto-generated when you create a new channel.

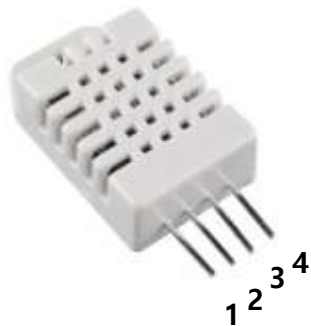
API Keys Settings

- **Write API Key:** Use this key to write data to a channel. If you feel your key has been compromised, click **Generate New Write API Key**.
- **Read API Keys:** Use this key to allow other people to view your

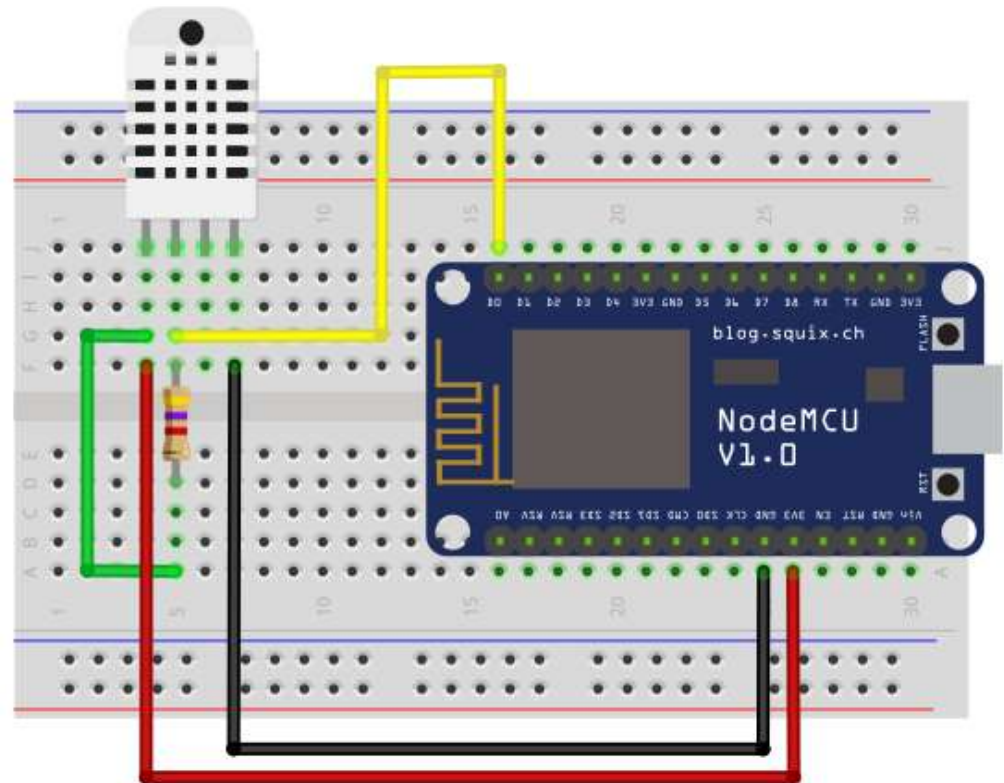
# 스마트 온습도 측정기 (3)

- ◆ 그림과 같이 회로를 구성 (이전 회로와 동일)
- ◆ 아두이노 IDE에서 다운로드 받은 Temp\_Humid\_ThinkSpeak.ino 열기
- ◆ 컴파일 및 업로드

DHT22로 바꾸고.. 핀 번호도 D01으로 변경



| DHT22 pinouts |      |
|---------------|------|
| 1             | Vcc  |
| 2             | Data |
| 3             | NC   |
| 4             | GND  |



# 코드 수정

## ◆ 제공된 코드에서 아래 부분의 값들을 변경

### ■ 접속할 Wi-Fi AP의 SSID와 P/W값 입력

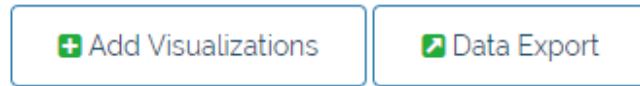
```
14  
15 const char* ssid    = "IoTStLabs";  
16 const char* password = "fold1108";  
17
```

### ■ API Key값 변경

```
6  
7 const char* ssid = "IoTStLabs";  
8 const char* password = "fold1108";  
9  
10 const char* server = "api.thingspeak.com";  
11 String apiKey = "WL5S2DCJMV021WWG";  
12
```

# 플랫폼을 통해 동작 확인

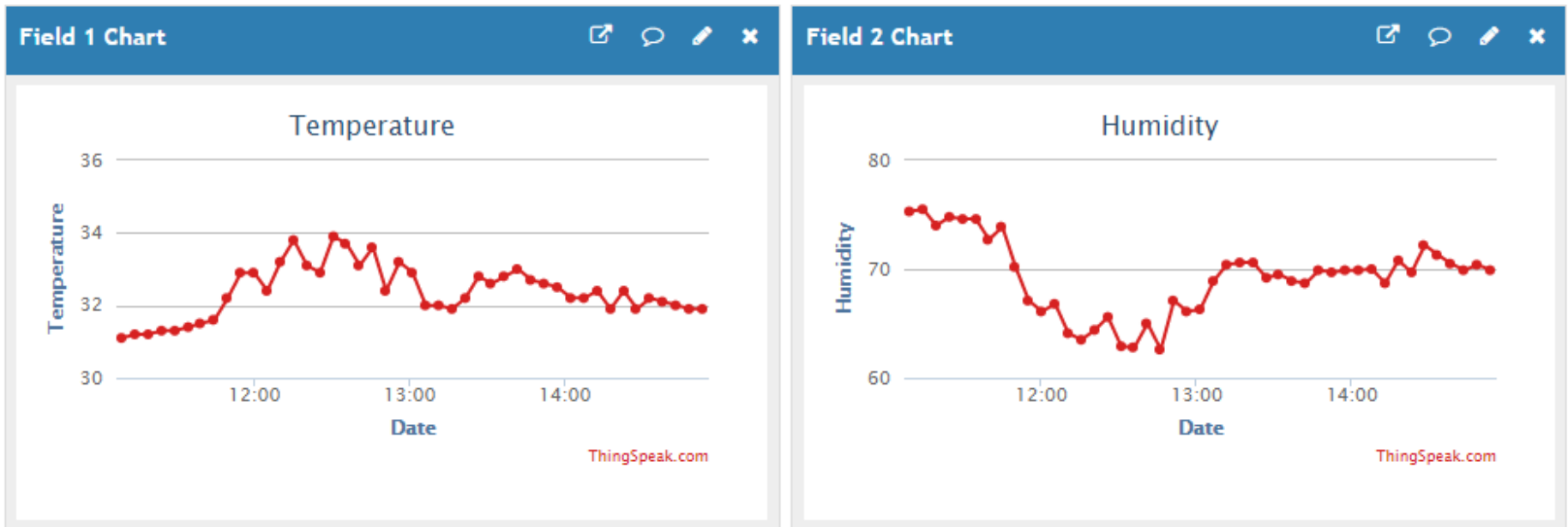
## ◆ 데이터를 다운로드 받을 수도 있음



### Channel Stats

Created: [about 21 hours ago](#)  
Updated: [4 minutes ago](#)  
Last entry: [4 minutes ago](#)  
Entries: 424

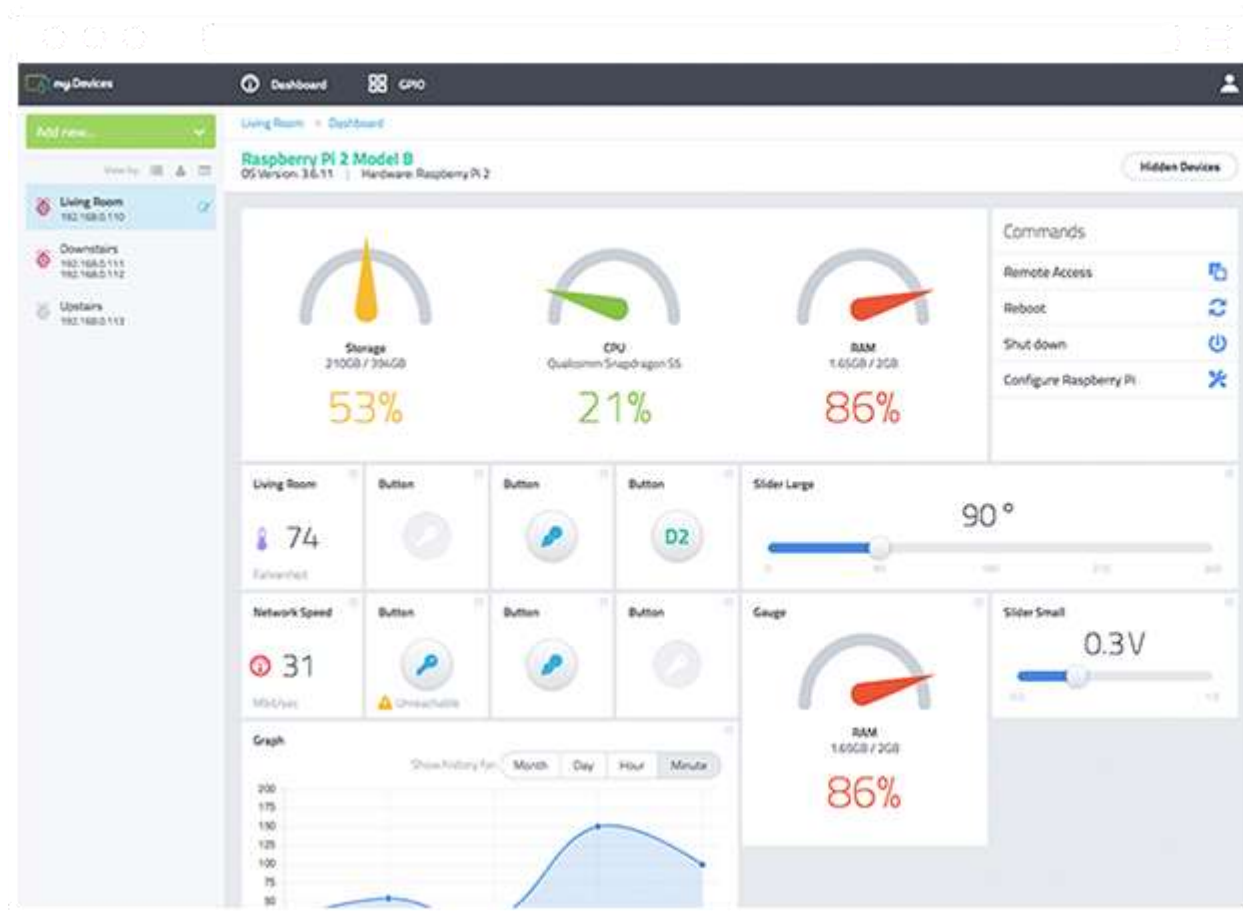
- 무료라서 100개 데이터로 제한됨





# 오픈 소스 IoT 플랫폼

## ◆ Cayenne (<https://mydevices.com/cayenne/features/>)



# 오픈 소스 IoT 플랫폼 / 모바일 앱

## ◆ Blynk



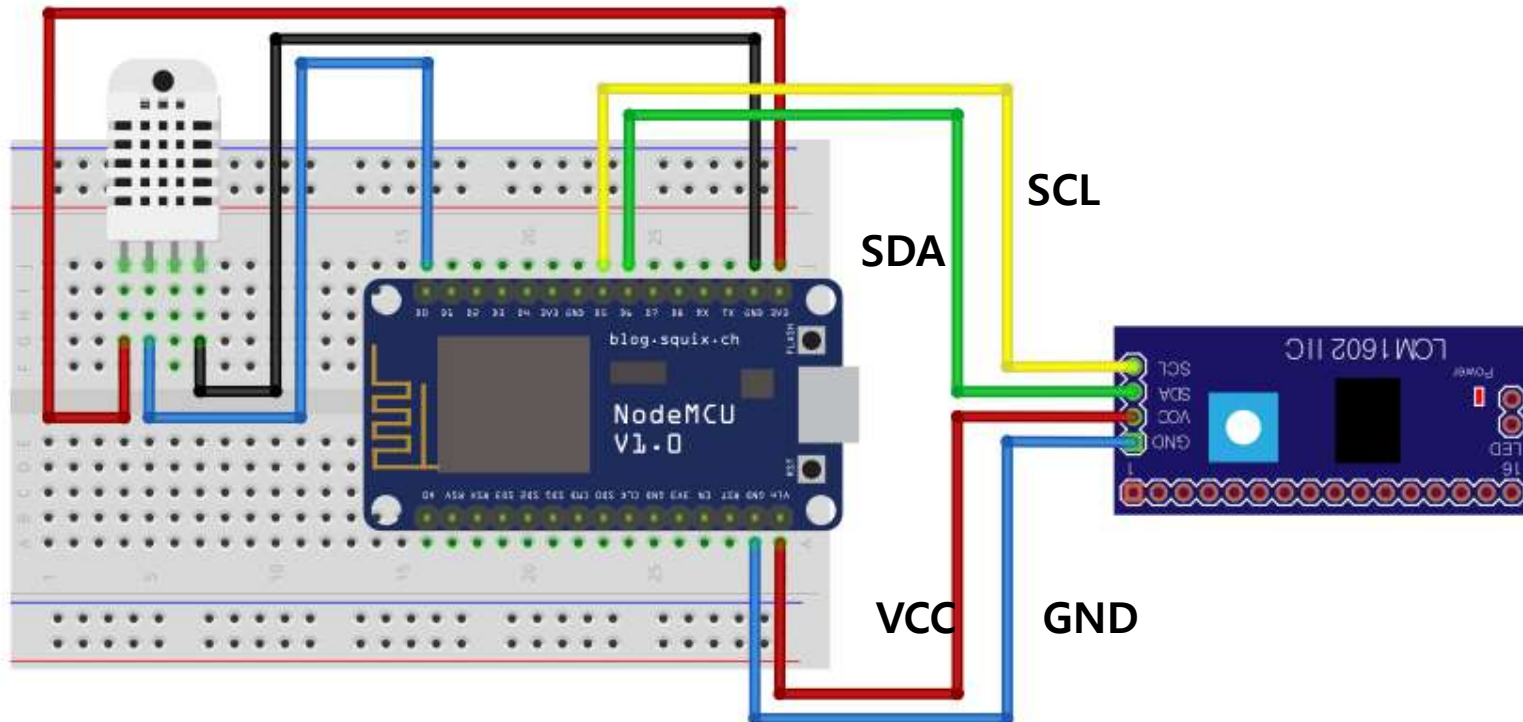
# 오픈 소스 분석 툴

## ◆ Grafana (https://grafana.com/)



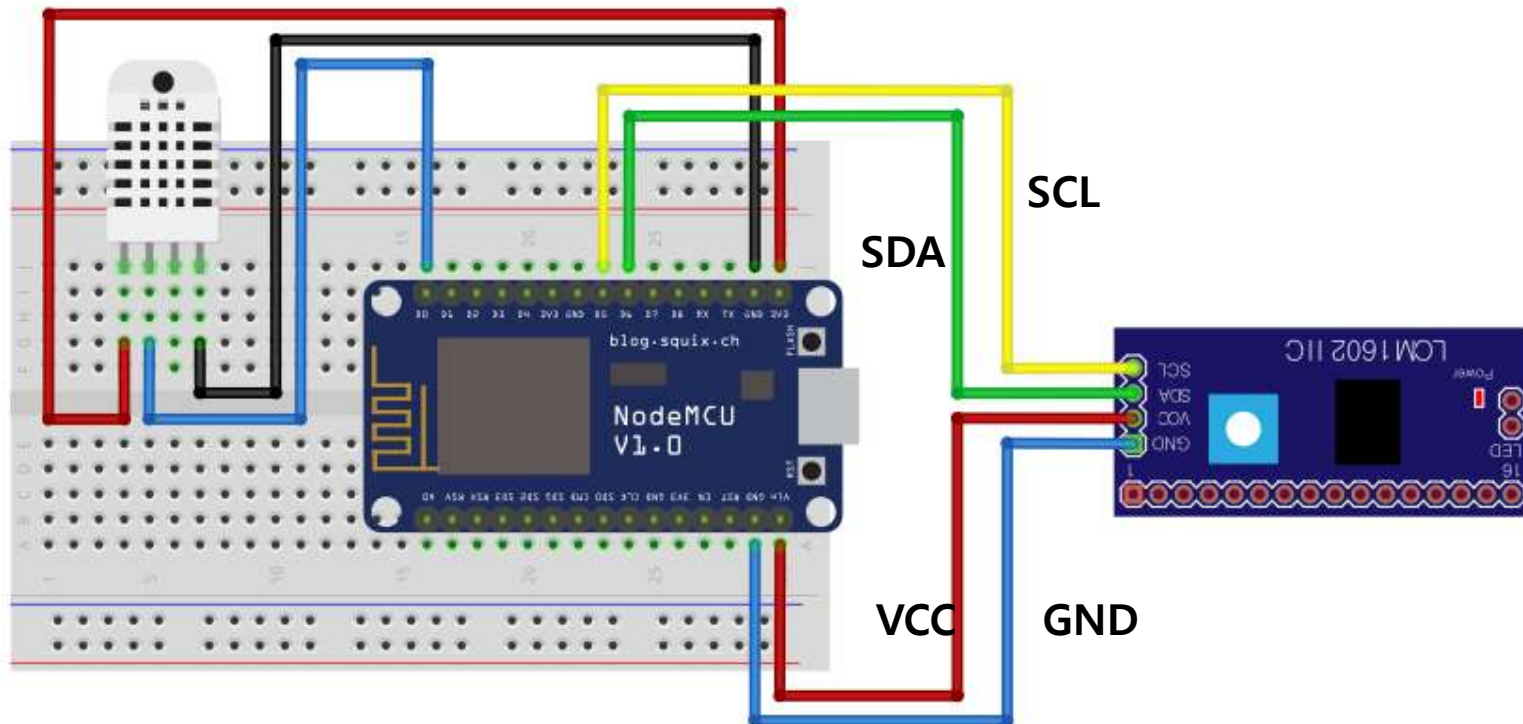
# 온습도 데이터를 I2C LCD에 표시하기

- ◆ 그림과 같이 회로를 구성 (DHT22의 Out 핀을 D0에 연결)
- ◆ 메일로 공유된 Temp\_Humid\_I2C\_LCD.ino 열기
  - Github.com에서 LiquidCrystal\_I2C.h 를 zip으로 다운받아 라이브러리 추가하기
  - LiquidCrystal\_I2C lcd(**0x3F**, 16, 2); or LiquidCrystal\_I2C lcd(**0x27**, 16, 2);



# 네트워크 시계 만들기

- ◆ 이전 회로 그대로 유지
- ◆ 메일로 전달받은 NTP\_I2C\_LCD.ino 열기
  - TimeLib.h esp8266으로 검색해서 TimeLib.h 라이브러리 추가
  - SSID와 password는 강의실 것으로 바꿔 주세요
  - 정상적으로 시간이 표시되면, 위쪽에는 시간 표시, 아래는 온습도 표시하도록 수정



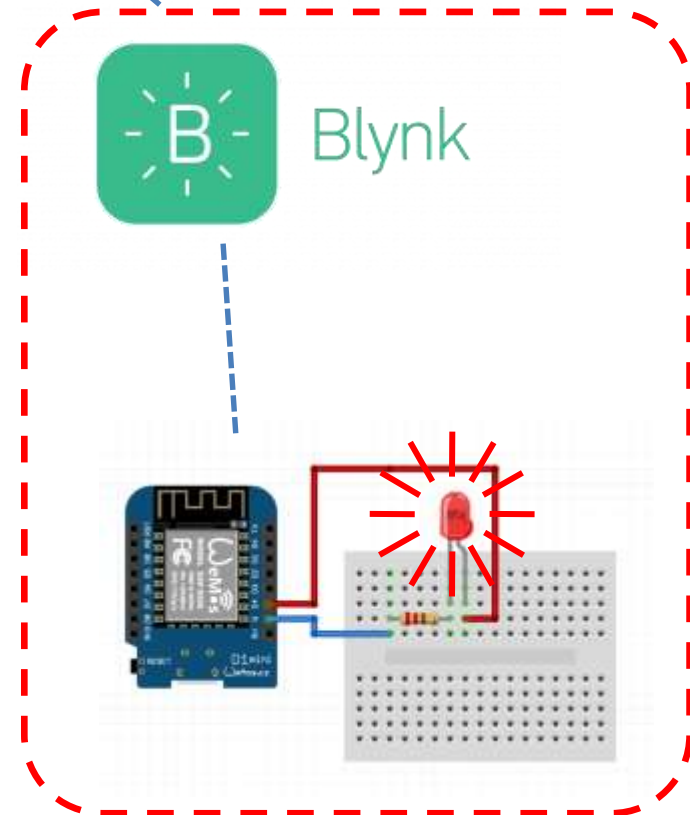


# 음성명령으로 LED 제어하기

## IFTTT



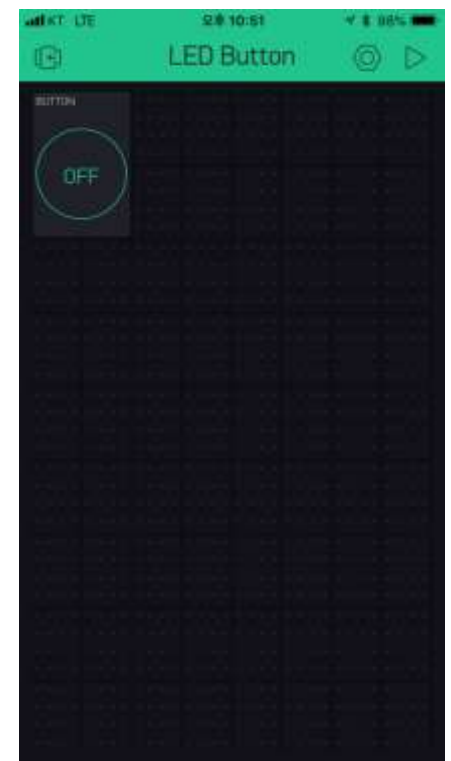
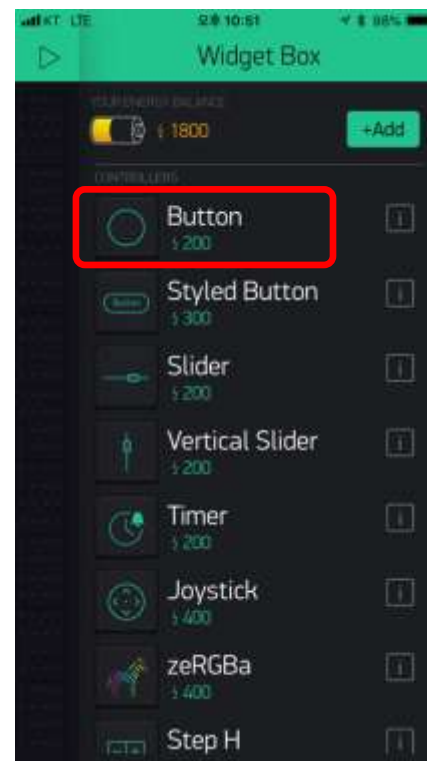
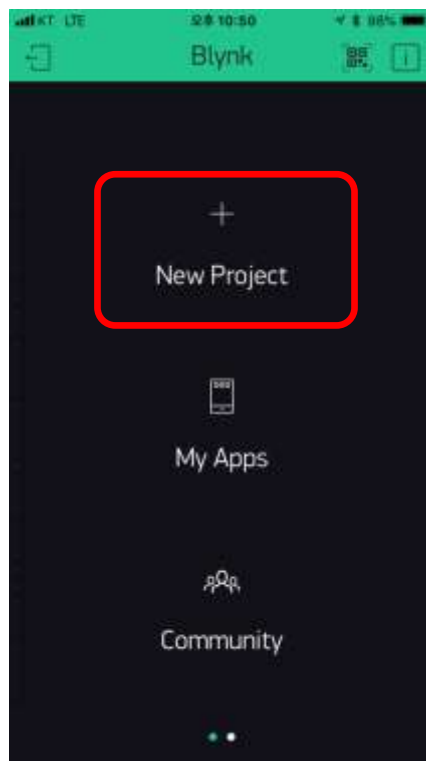
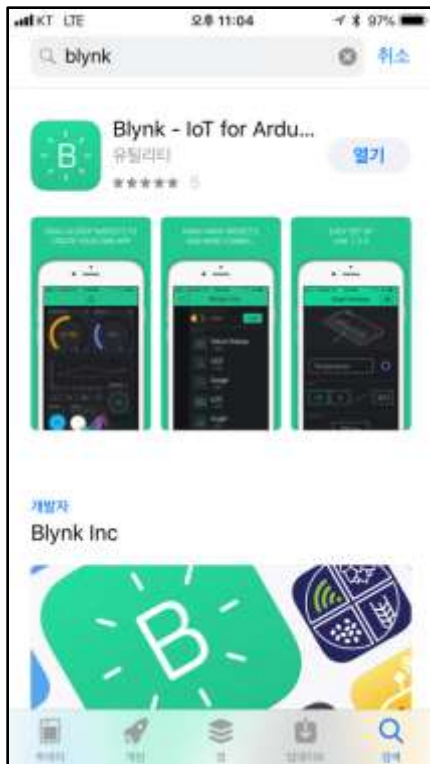
Ok Goolge, turn OFF yellow





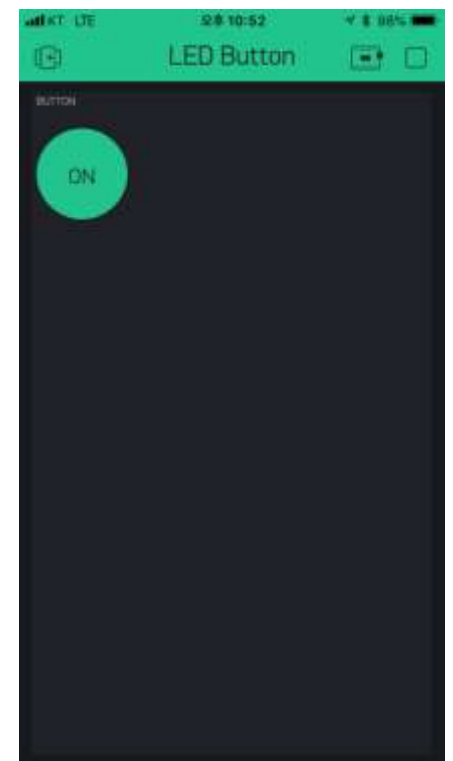
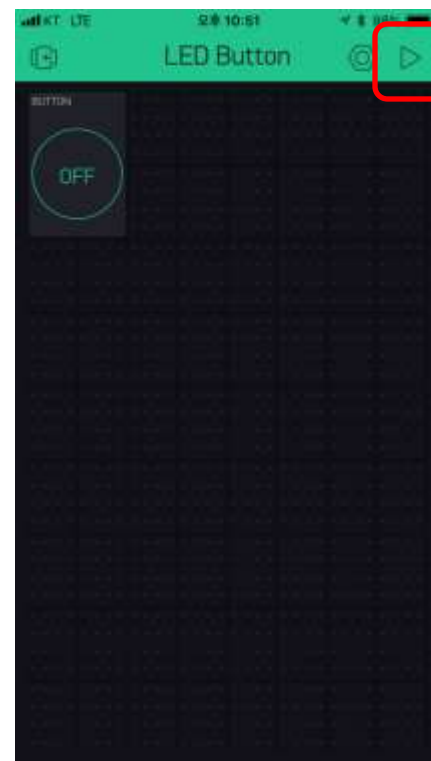
# Blynk 앱을 이용해서 LED 제어하기

- ◆ 스마트폰 앱스토어에서 Blynk 앱을 다운로드 해서 설치
- ◆ 회원 가입 후 +New Project 클릭, 프로젝트명 LED Button으로 설정
- ◆ 화면을 왼쪽으로 밀어서 Button 추가



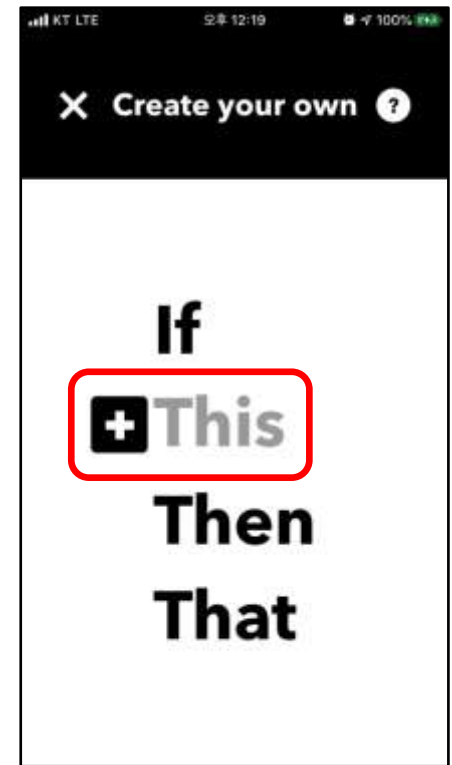
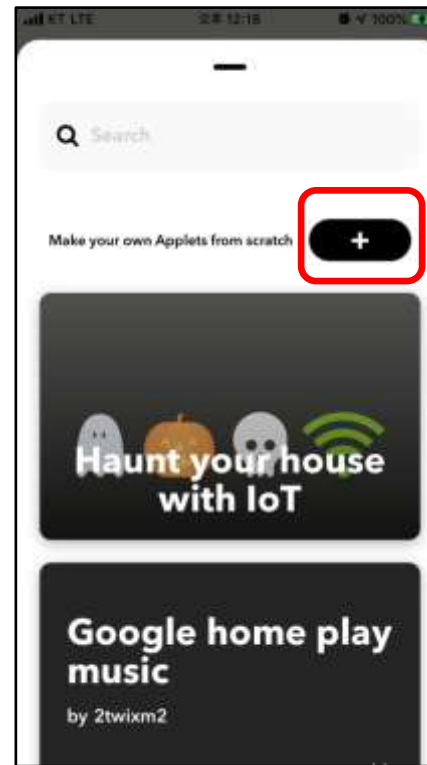
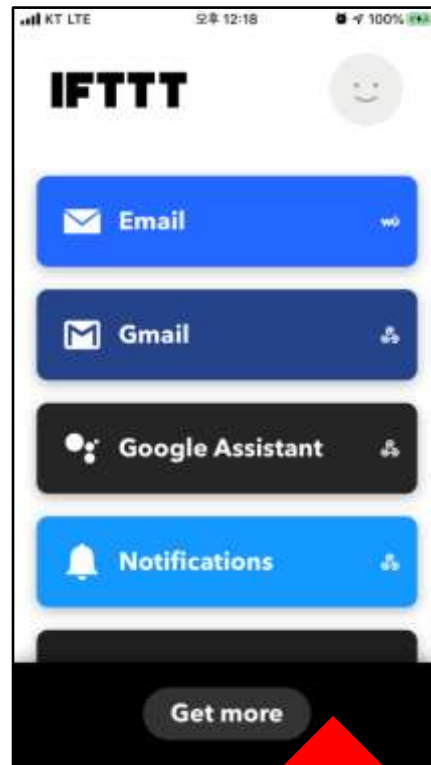
# Blynk 앱을 이용해서 LED 제어하기

- ◆ 버튼 모양 선택 후 설정, GP16은 D0에 연결된 것을 의미
- ◆ 스위치 상태는 SWITCH에 놓고 OK 클릭
- ◆ 화면 상단 오른쪽의 삼각형 클릭 후 버튼을 눌러 봄, OFF 버튼 클릭



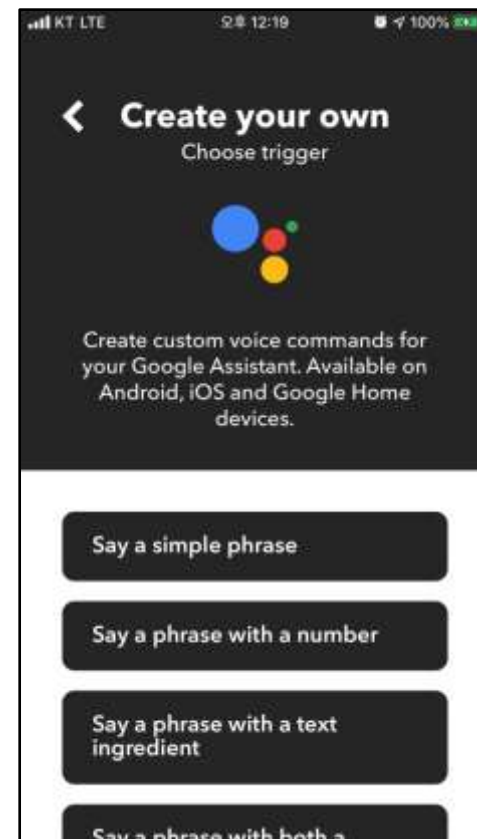
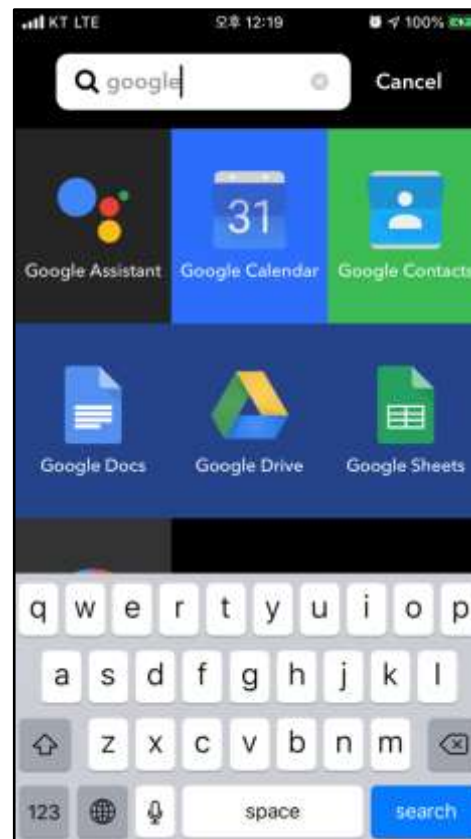
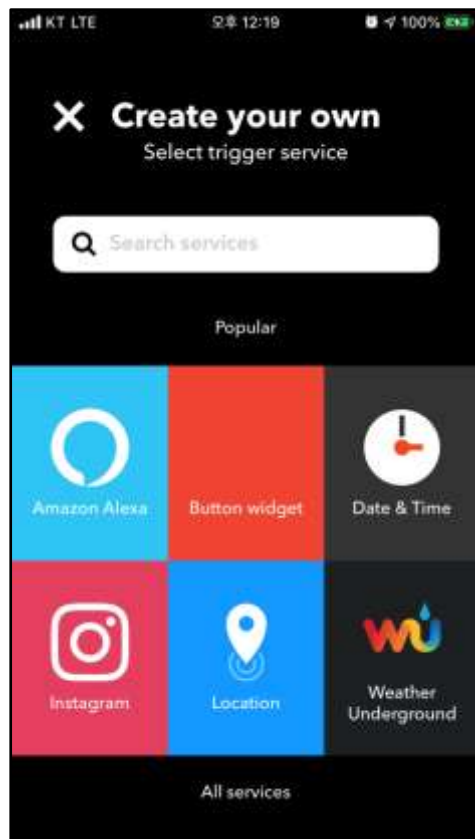
# IFTTT (IF This Then That) 설치

- ◆ 스마트폰 앱스토어에서 IFTTT 설치 및 실행
- ◆ Get More를 위로 올린 후 +를 클릭해서 새로운 애플릿 생성
- ◆ this 누르고 검색창에서 Google Assistant 입력 후 선택



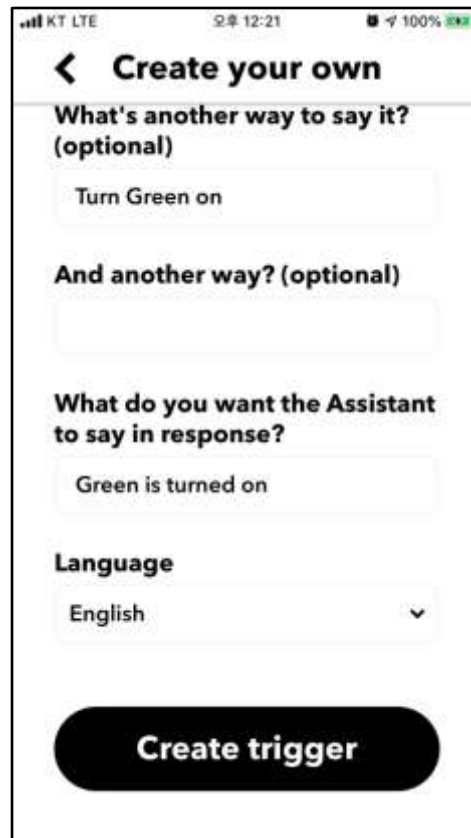
# IFTTT (IF This Then That) 설치

◆ Google Assistant가 선택되면 Say a simple phrase 선택



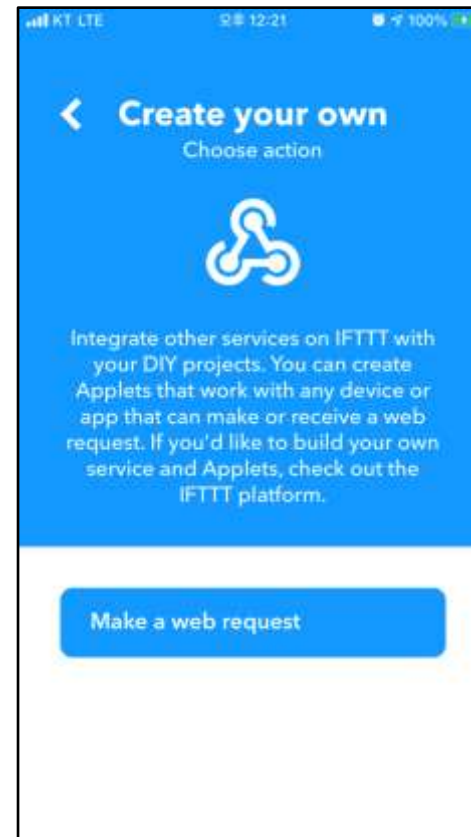
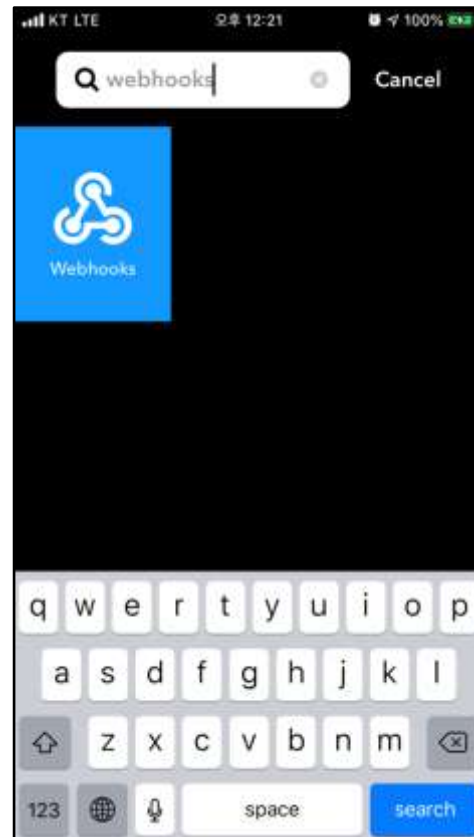
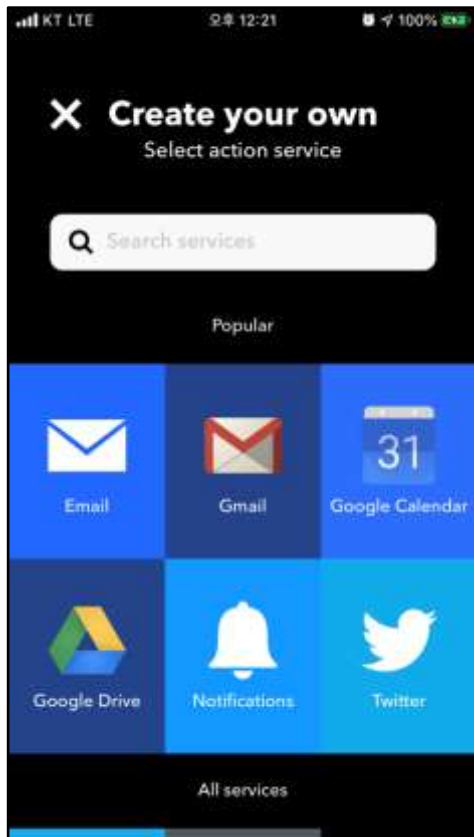
# IFTTT (IF This Then That) 설치

- ◆ 첫번째 두번째 보이는 이미지처럼 텍스트 입력 후 Create trigger 클릭
- ◆ 새 화면에서 +That 클릭



# IFTTT (IF This Then That) 설치

- ◆ 검색창에서 webhook을 검색해서 선택
- ◆ 그리고 Make a web request를 선택 후 다음 페이지처럼 입력





# IFTTT (IF This Then That) 설치

- URL에는 `http://188.166.206.43/blynk_auth_key/update/digital_pin_used_in_blynk`  
→ 이런 형태가 되어 함 `http://188.166.206.43/456fec7379~~~~66567b/update/D16`
- `blynk_auth_key`는 blynk에서 프로젝트 만들 때 이메일로 받은 키로  
Blynk 앱의 프로젝트 설정( )을 누른 후 확인 가능
- `Digital_pin_used_in_Blynk`는 D16 입력 (디지털의 D와 D0 핀의 GPIO 값)

Method: PUT

The method of the request e.g. GET, POST, DELETE

Content Type: application/json

Optional

Body: ["1"]

Surround any text with "<<>>" to escape the content

Create action

Make a web request

This action will make a web request to a publicly accessible URL. NOTE: Requests may be rate limited.

URL: `http://188.166.206.43/456fec...`

Surround any text with "<<>>" to escape the content

Method: PUT

The method of the request e.g. GET, POST, DELETE

Content Type

If You say "Turn on Green", then Make a web request

Edit title

Receive notifications when this connection is active

Finish

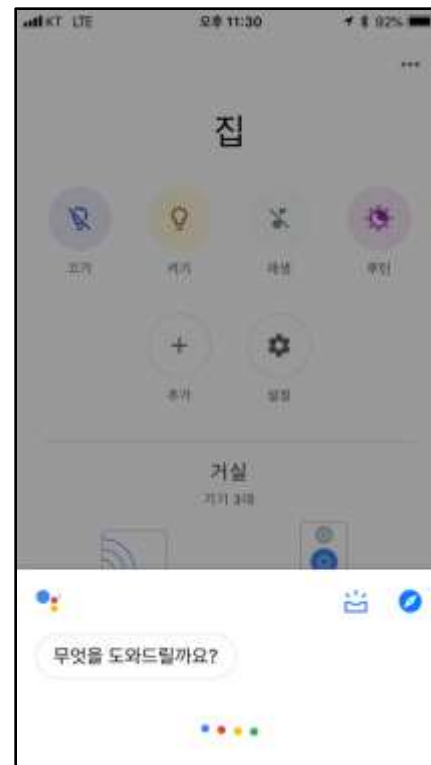
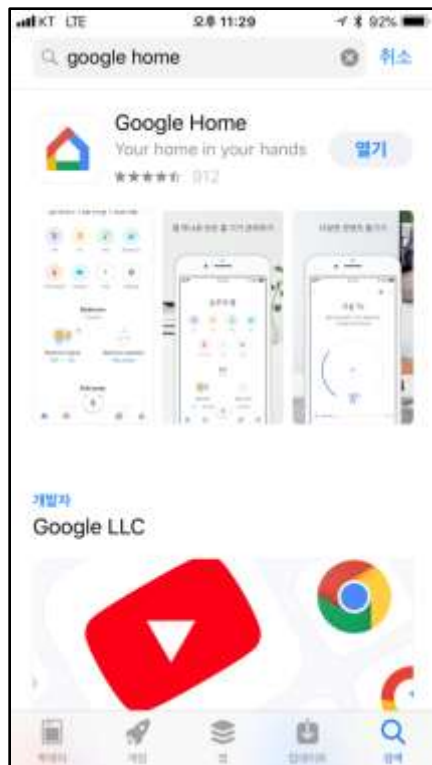
# IFTTT (IF This Then That) 설치

- ◆ 동일한 방식으로 새로운 애플릿 생성
  - Turn on yellow 대신 Turn off yellow 사용
  - Yellow is turned on 대신 Yellow is turned off 사용
  - Webhook 설정 시 Body 부분에 ["1"] 대신에 ["0"] 사용
- ◆ 동일한 방식으로 새로운 애플릿 생성
- ◆ Blynk\_ESP8266.ino가 실행 중인 NodeMCU에 전원 인가
  - Blynk 앱의 버튼을 클릭하는 식으로 동작 확인



# 구글홈 (Google Assistant) 설치

- ◆ 스마트폰의 언어를 영어로 설정
- ◆ 스마트폰 앱스토어에서 구글 어시스턴트 다운로드, 설치, 실행
- ◆ 마이크 버튼을 누르고 Turn on/off yellow 음성 명령 실행



# Thank You!!

**For more information, please visit**

- IoT Strategy Labs Homepage <http://weshare.kr>
- 사물인터넷 카페 : <http://cafe.naver.com/iotioe>
- 김학용 블로그 : <http://blog.naver.com/honest72>
- <https://www.facebook.com/hakyong.kim.12139>

**or contact me**

- phone : 010-4711-1434
- e-mail : [iotstlabs@gmail.com](mailto:iotstlabs@gmail.com)