

[ECE20008] 실전프로젝트1 IoT실습 - NodeMCU 활용

- 사물인터넷 (IoT) 이란 ?
- ESP8266 과 NodeMCU 소개
- NodeMCU 개발환경 구축 (ESP8266 Arduino IDE)
 - 아두이노 IDE 프로그램 설치
 - 아두이노 IDE 에 ESP866 개발환경 추가
 - 아두이노 IDE 기초
 - NodeMCU(ESP8266) 기초
- NodeMCU 를 이용한 IoT 실습 (실습 1 ~ 실습 4)

사물인터넷이란 ?



- 사물에 센서를 부착해 실시간으로 데이터를 인터넷으로 주고 받는 기술이나 환경
 - 각종 사물에 컴퓨터 칩과 통신 기능을 내장하여 인터넷에 연결하는 기술
- 인터넷에 연결된 기기 스스로 사람의 도움 없이 정보를 교환하고 동작하는 환경
- 가상세계 (웹 페이지 , 모바일 어플) 와 현실세계를 연결
 - 실세계와 가상세계에 존재하는 사람 , 사물 , 데이터 , 프로세스 등 모든 것이 인터넷으로 상호 연결되어 서로 소통하고 작용하는 지능형 서비스 인프라
- 블루투스나 근거리무선통신 (NFC), 센서데이터 , 네트워크들의 자율적인 소통을 돕는 기술

사물인터넷 종단 모듈 구성요소

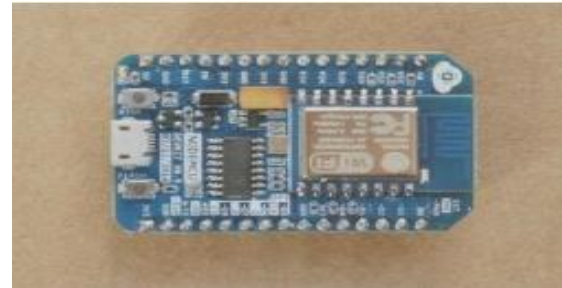
- 마이크로컨트롤러 장치 (MCU, Microcontroller Unit)
 - CPU 의 기능을 하는 핵심 장치와 주변 장치들을 포함하는 칩셋
 - 소프트웨어를 다운로드하면 coding 목적대로 동작하며 , 주변장치들을 제어함
 - (예) 아두이노 , **ESP8266** , NodeMCU (ESP8266 개발키트)
- 센서 (Sensor)
 - 디바이스 주변의 물리적 , 환경적 변화를 전기 신호의 변화로 바꿔주는 장치
- 액츄에이터 (Actuator)
 - 입력된 신호에 대응하여 작동을 수행하는 장치 또는 명령신호에 의하여 작동하는 장치
 - 전기 , 유압 , 압축 공기 등을 동력원으로 사용

마이크로컨트롤러 장치 예시

■ MCU(MicroController Unit)



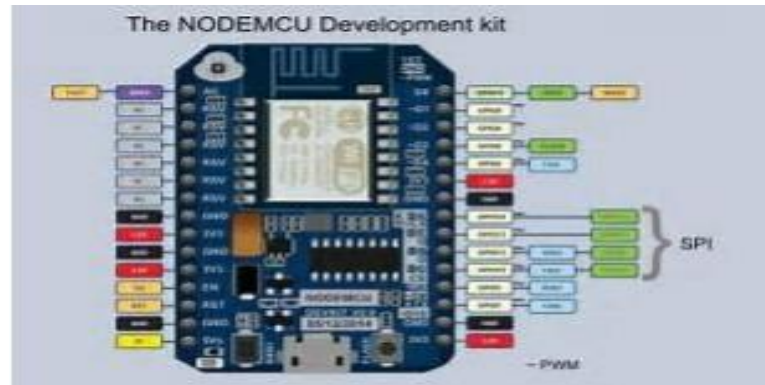
아두이노 우노



ESP8266



CC2541



NodeMCU

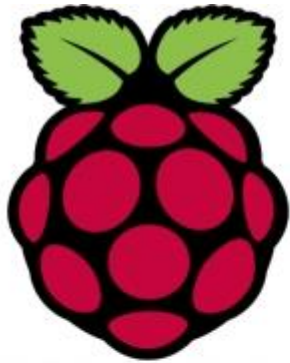
사진 출처 : [아두이노] <https://store.arduino.cc>

[nodemcu] <http://nodemcu.com>

[cc2541] <http://www.instructables.com/id/Turn-your-CC2541-based-HM-10-Bluetooth-Smart-Modul>

사물인터넷의 구조 및 구현 기술

■ MPU(MicroProcessor Unit)



라즈베리파이3



Samsung
ARTIK™



삼성ARTIK

intel Edison



Intel Edison

사진 출처 : [라즈베리파이 3] <https://www.raspberrypi.org>

[삼성 ARTIK] <https://www.artik.io>

[Intel Edison] <https://software.intel.com/en-us/iot/hardware/edison>

센서 예시 - 1

■ 센서 (Sensor)

초음파 센서



가스 센서



움직임 감지 센서



온습도 센서



소리 감지 센서



시간 센서



레이저 센서



불꽃 감지 센서

사진 출처 : <https://ko.aliexpress.com/item/Official-smarian-Smart-Electronics-16-kinds-of-sensor-Raspberry-Pi-3-Raspberry-Pi-2-Model/32796381356.html?spm=a2g12.search0305.4.95.uXCfKz>

센서 예시 - 2

■ 센서 (Sensor)

RF 송수신기 모듈

광 센서

진동 감지 센서

장애물 감지 센서



기울기 센서

적외선 광추적 센서

비 감지 센서

토양 수분 센서

사진 출처 : <https://ko.aliexpress.com/item/Official-smarian-Smart-Electronics-16-kinds-of-sensor-Raspberry-Pi-3-Raspberry-Pi-2-Model/32796381356.html?spm=a2g12.search0305.4.95.uXCfKz>

액츄에이터 (Actuator) 예시

■ 액츄에이터 (Actuator)

LED



7 세그먼트



LCD



팬 모터



DC 모터



서보 모터



진동 모터



부저



ESP8266 과 NodeMCU



■ ESP8266

- Espressif System사에서 생산한 TCP/IP 풀스택 및 MCU 기능을 갖춘 **저비용 Wi-Fi 칩**
- 모듈에 올라갈 펌웨어를 직접 개발할 수 있도록 Espressif사에서 SDK(Software Development Kit) 와 소스코드를 오픈소스로 제공함
- 서드파티 업체에서 제조한 ESP-01 부터 ESP-12 까지 다양한 모듈이 있음
- 3.3V 동작 전압을 가짐

■ NodeMCU

- ESP8266 을 이용하여 손쉽게 개발할 수 있도록 **모듈을 확장하여** 만든 개발보드임
- 컴퓨터와 USB 로 연결을 하면 ESP8266 프로그래밍을 시작할수 있음

ESP8266이란?

- 다양한 종류의 모듈이 있음



ESP8266 모듈의 종류

■ ESP-01

- 최소한의 회로만 갖춘 소형 모듈로 저렴
- 브레드보드에 붙여 사용하기 힘들고 부가적인 회로를 구성해줘야 하므로 통신 목적으로만 사용시 유용



■ ESP-12, ESP-12E(개선판)

- 전파 특성과 플래시메모리 등 성능이 향상된 버전으로 **GPIO**핀 또한 넉넉히 제공



ESP8266 모듈의 종류

- **NodeMCU Lua Dev board**
 - ESP-12 모듈을 기반으로 만든 custom 보드로 FTDI칩을 탑재하여 USB 연결만으로 개발, 테스트가 가능함



NodeMCU 개발 환경 구축

■ 개발 환경

- 오픈소스인 아두이노 개발환경(ArduinoIDE)의 경우 버전 1.6부터 32-bit 프로세서를 탑재한 보드들을 지원함
- ESP8266 ArduinoIDE 개발환경에서는 ESP8266을 아두이노보드처럼 인식하고 기존의 아두이노 스케치코드를 그대로 사용할 수 있음

■ NodeMCU 실습을 위한 툴 설치

- 1.아두이노(Arduino) IDE 프로그램 설치
 - 아두이노 공식사이트(Arduino.cc)에서 프로그램 설치
 - 사이트에서 **software** 메뉴를 선택한 후 아래 툴 설치
- 2.아두이노 프로그램에서 NodeMCU 보드를 사용할 수 있도록 ESP8266 개발환경 추가하기

1. 아두이노(Arduino) IDE 프로그램 설치

- 아두이노(Arduino)스케치설치
 - 아두이노 공식사이트(**arduino.cc**)에서 프로그램 설치
 - ▣ 사이트에서 **software** 메뉴를 선택한 후 아래 툴 설치

Download the Arduino IDE



ARDUINO 1.8.4

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

Windows ZIP file for non admin install

Windows app 

Mac OS X 10.7 Lion or newer

Linux 32 bits

Linux 64 bits

Linux ARM

Release Notes

Source Code

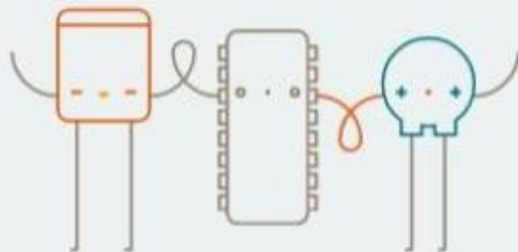
Checksums (sha512)

1. 아두이노(Arduino) IDE 프로그램 설치

■ Just Download 클릭

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.



SINCE MARCH 2015, THE ARDUINO IDE HAS BEEN DOWNLOADED **18,090,355** TIMES. (IMPRESSIVE!) NO LONGER JUST FOR ARDUINO AND GENUINO BOARDS, HUNDREDS OF COMPANIES AROUND THE WORLD ARE USING THE IDE TO PROGRAM THEIR DEVICES, INCLUDING COMPATIBLES, CLONES, AND EVEN COUNTERFEITS. HELP ACCELERATE ITS DEVELOPMENT WITH A SMALL CONTRIBUTION! REMEMBER: OPEN SOURCE IS LOVE!

\$3

\$5

\$10

\$25

\$50

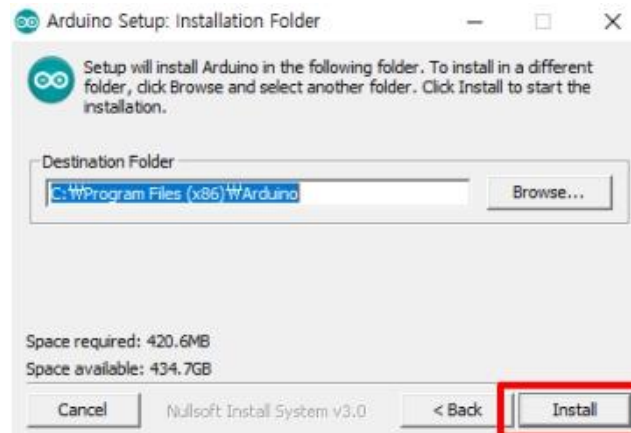
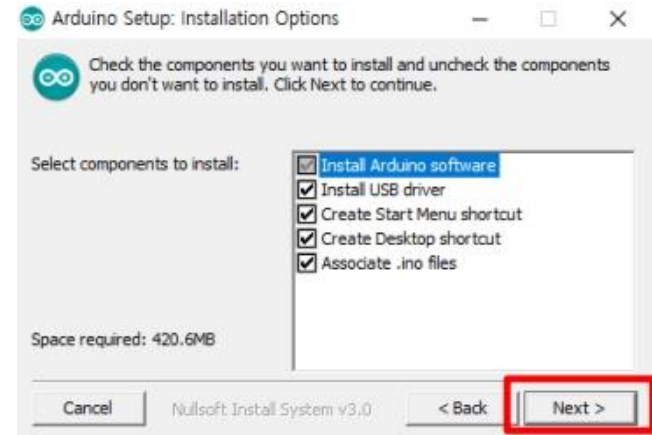
OTHER

JUST DOWNLOAD

CONTRIBUTE & DOWNLOAD

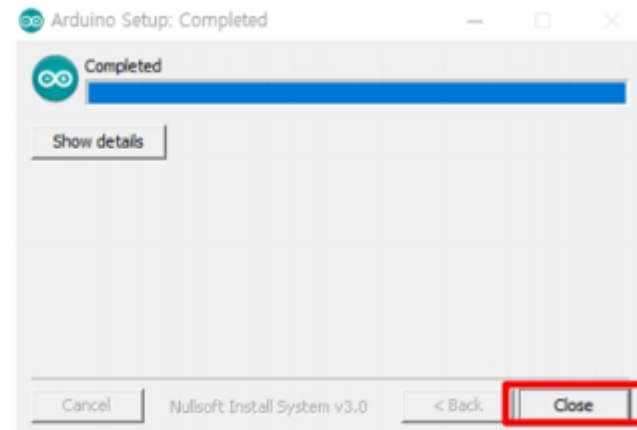
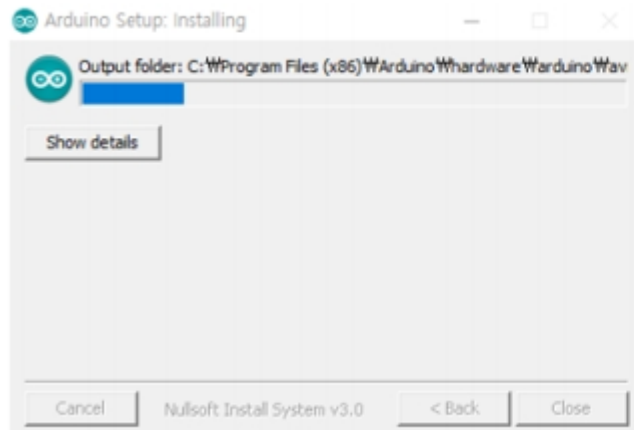
1. 아두이노(Arduino) IDE 프로그램 설치

■ I Agree->Next->Install



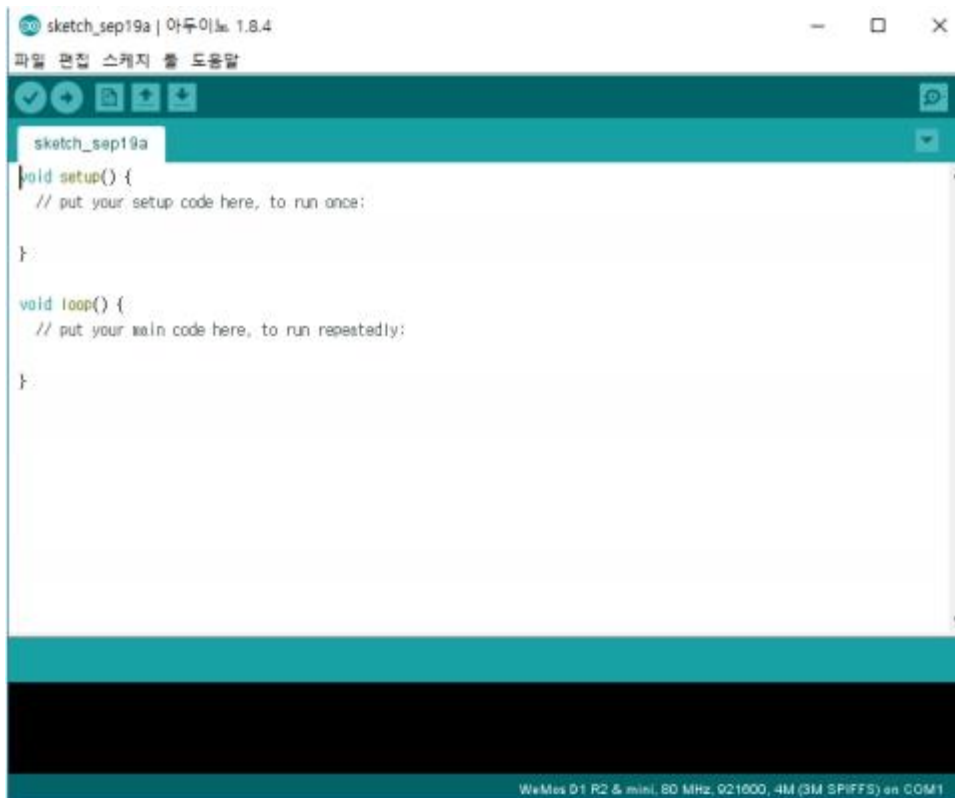
1. 아두이노(Arduino) IDE 프로그램 설치

■ Just Download 클릭



1. 아두이노(Arduino) IDE 프로그램 설치

- 설치 완료시 바탕화면에 아이콘 생성
- 더블 클릭시 아두이노 스케치 실행



2. 아두이노 IDE 에 ESP866 개발환경 추가

■ 상단의 파일 → 환경설정



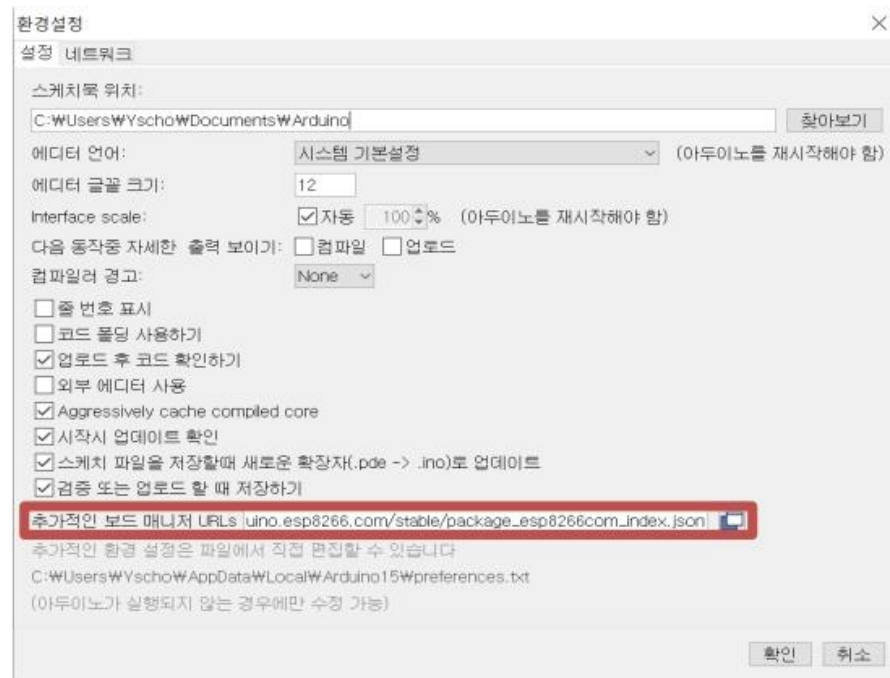
2. 아두이노 IDE 에 ESP866 개발환경 추가

■ 아두이노 IDE 환경설정에서

ESP8266 보드 매니저 URL 추가

– 아두이노 IDE → 파일 → 환경설정 → 추가적인 보드매니저 URL에
http://arduino.esp8266.com/stable/package_esp8266com_index.json

추가



2. 아두이노 IDE 에 ESP866 개발환경 추가

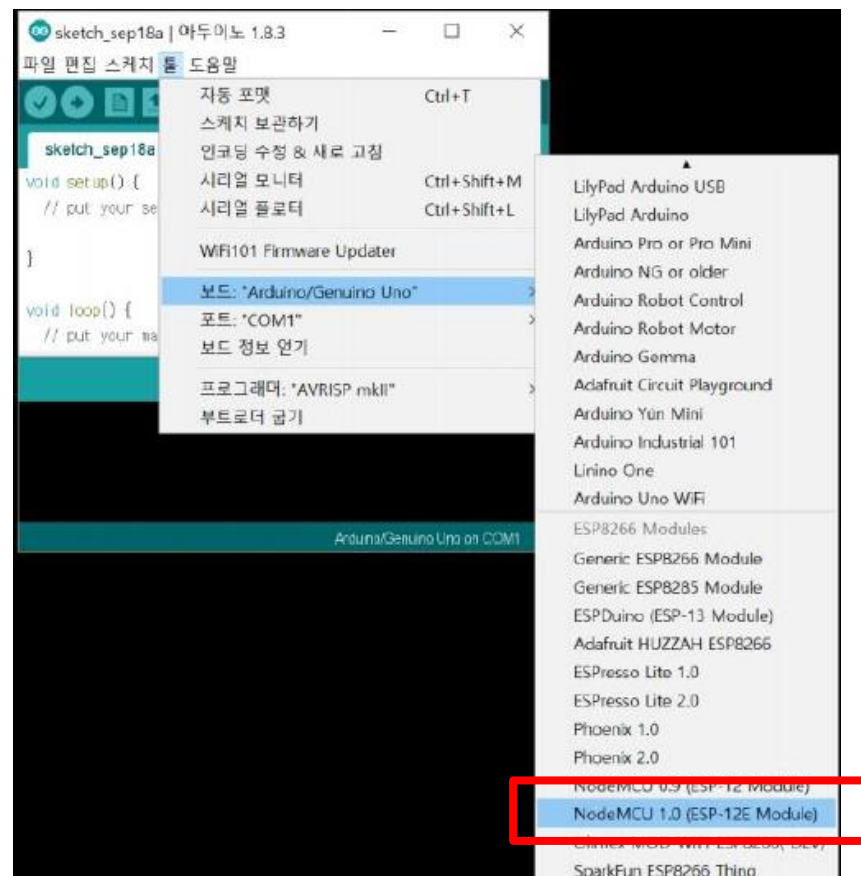
- 아두이노 IDE 에 ESP8266 관련 파일 설치
 - 툴→보드: "Arduino/ Genuino Uno" → 보드매니저 → esp검색
→ esp8266 by ESP8266 Community 설치



2. 아두이노 IDE 에 ESP866 개발환경 추가

■ nodeMCU 보드 선택하기

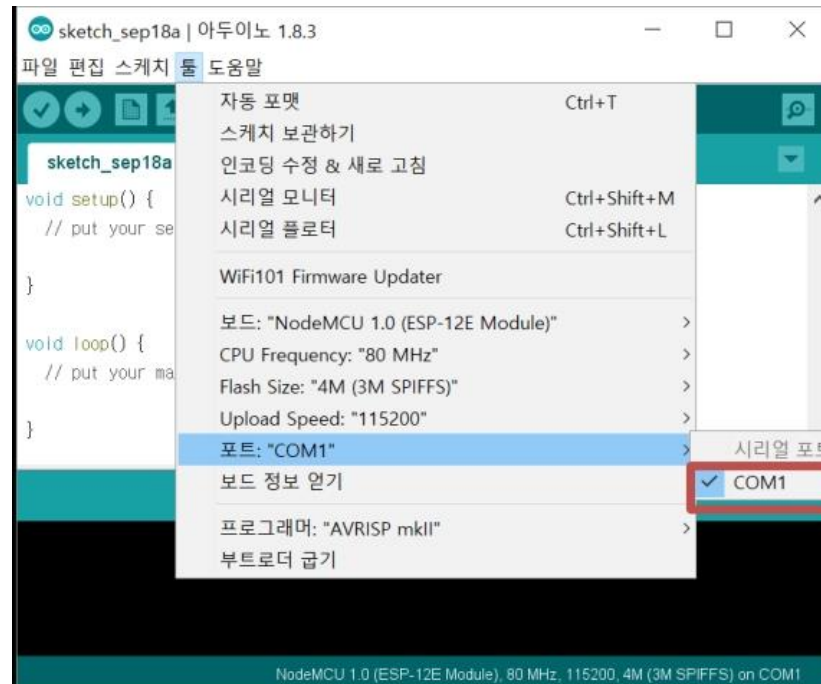
- 툴→보드→**NodeMCU1.0 (ESP-12E Module)** 선택



2. 아두이노 IDE 에 ESP866 개발환경 추가

■ 통신용 포트 선택

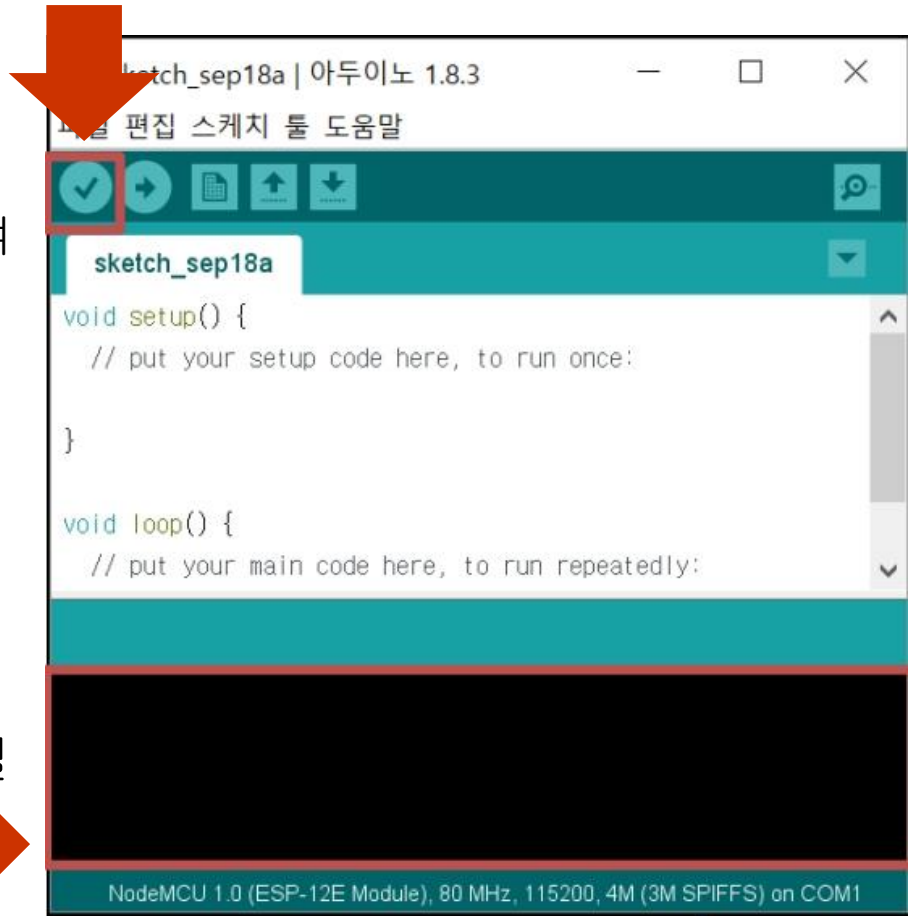
- 툴 → 포트 → 사용할 포트 선택
- 포트의 번호는 컴퓨터마다 다름



3. 아두이노 IDE 기초

■ Arduino IDE 화면 구성

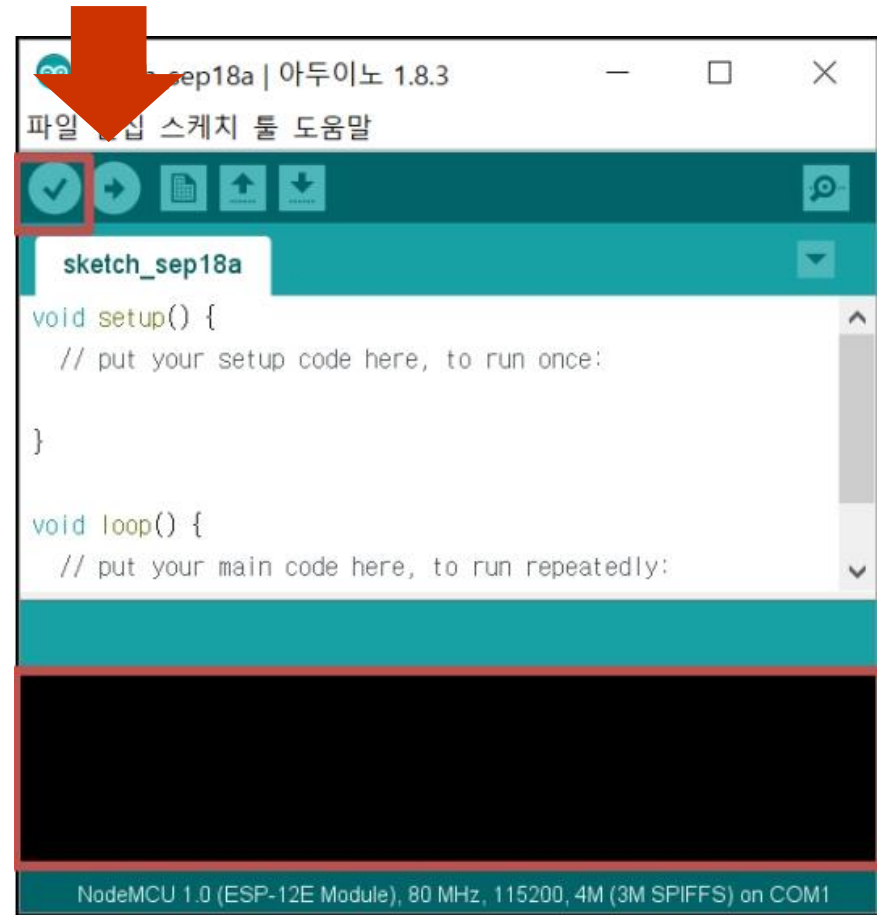
- 제일 왼쪽 위의 체크 형태의 아이콘은 컴파일 버튼
 - 컴파일:소스코드를 번역하여 MCU가 이해하고 실행할 수 있도록 기계어로 변환하는 과정을 의미
- 컴파일 결과는 하단의 터미널 창에 표시됨



3. 아두이노 IDE 기초

■ Arduino IDE 기본 구조

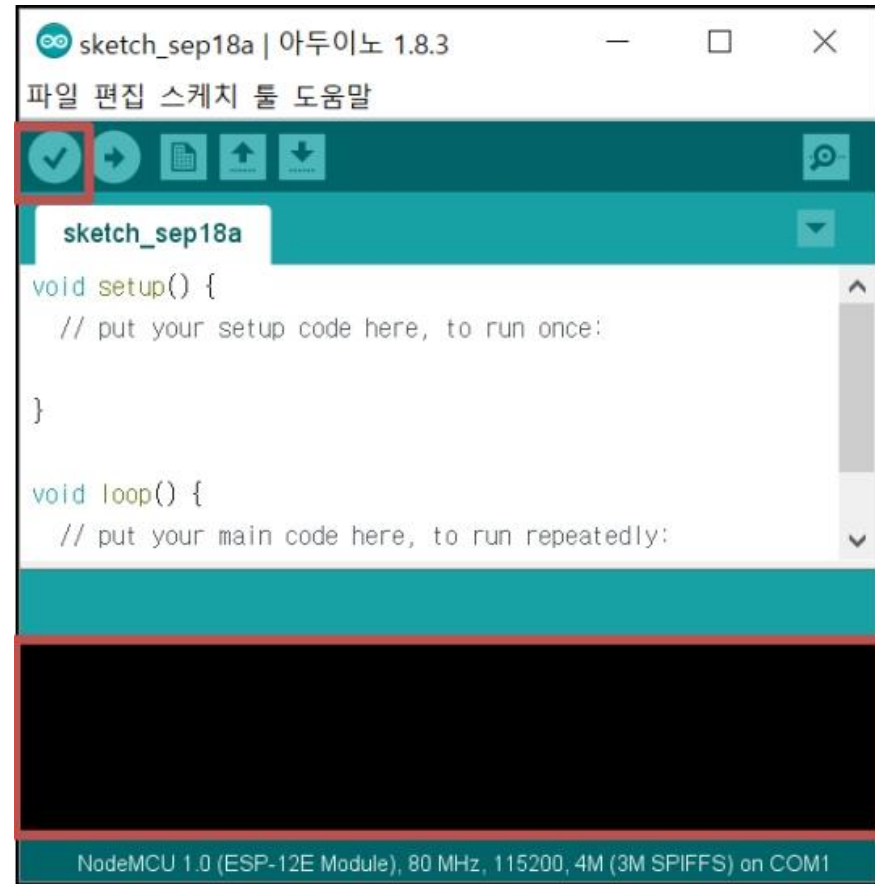
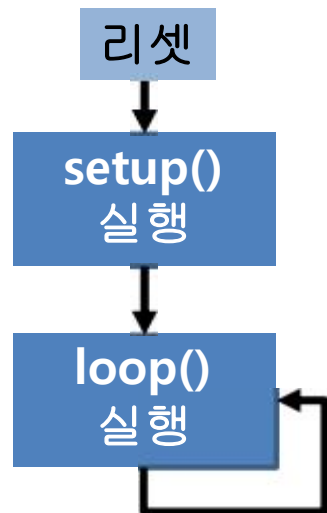
- 컴파일 버튼 옆의 오른쪽 방향 화살표 키 아이콘은 컴파일 및 업로드 버튼
 - 업로드: 코드를 컴파일한 결과를 MCU에 올리는 과정



3. 아두이노 IDE 기초

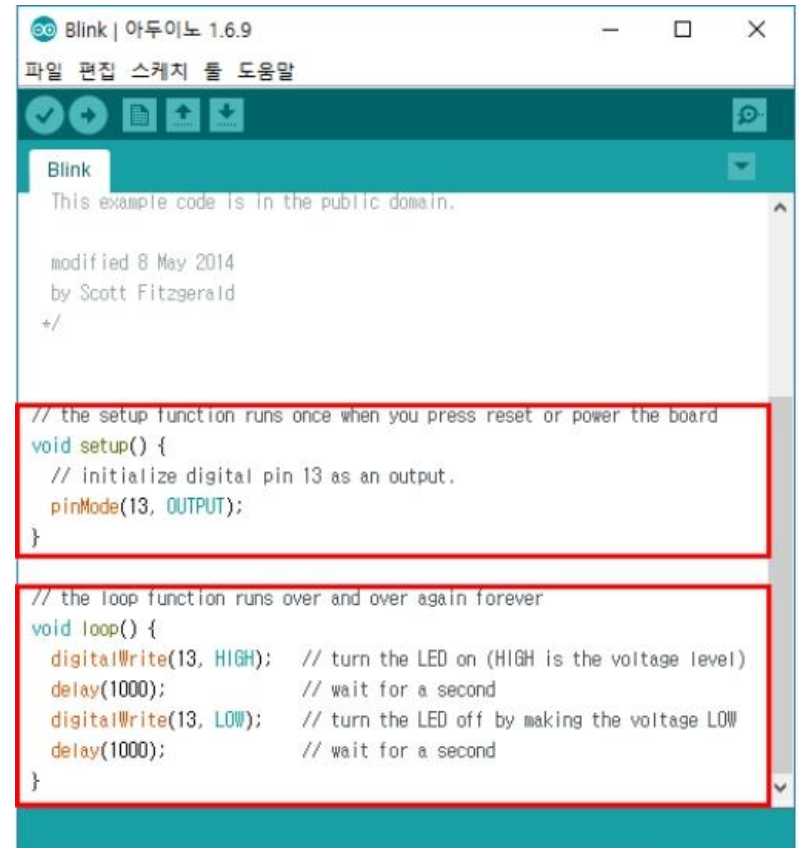
■ ArduinoIDE에서의 코드 (code)구조

- 아두이노 코딩의 기본 구조는 하나의 **setup()**함수와 하나의 **loop()** 함수로 구성됨
- **setup()**→처음 단1회 수행됨
- **loop()**→setup 이후 반복 수행



3. 아두이노 IDE 기초

- 프로그램 기본 구조
 - 주석: 코드의 역할을 설명하는 부분으로 한 줄은 `//` 여러 줄은 `/* */`을 이용
 - **pinMode(num,dir);**
 - Pinnum의 입출력 방향을 정의함 (dir은 INPUT or OUTPUT)
 - **digitalWrite(num,val);**
 - Pin num에 val의 값을 인가함 (val은 HIGH or LOW)
 - **delay(msec);**
 - Millisecond만큼의 시간 지연 후에 다음 실행문 실행



```
Blink | 아두이노 1.6.9
파일 편집 스케치 툴 도움말

Blink
This example code is in the public domain.

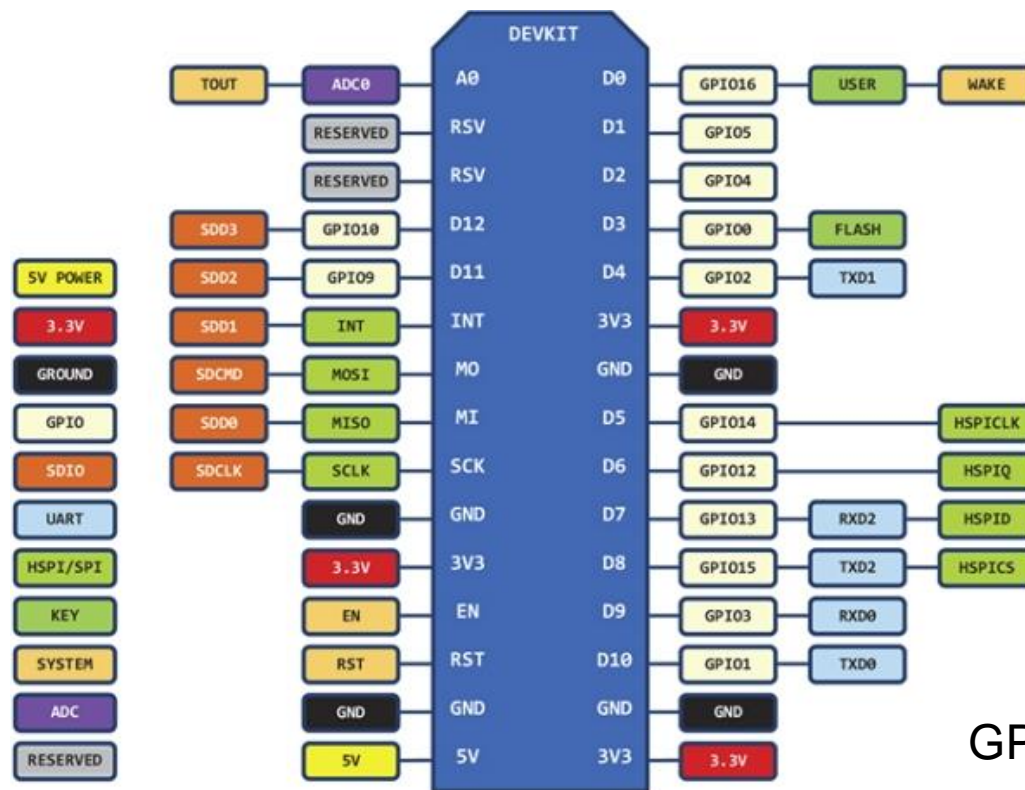
modified 8 May 2014
by Scott Fitzgerald
*/

// the setup function runs once when you press reset or power the board
void setup() {
  // initialize digital pin 13 as an output.
  pinMode(13, OUTPUT);
}

// the loop function runs over and over again forever
void loop() {
  digitalWrite(13, HIGH); // turn the LED on (HIGH is the voltage level)
  delay(1000);            // wait for a second
  digitalWrite(13, LOW);  // turn the LED off by making the voltage LOW
  delay(1000);            // wait for a second
}
```

4. NodeMCU(ESP8266) 기초

- 아두이노 IDE에서는 ESP8266 핀번호가 아닌 그림의 GPIO번호를 사용해야 함 (2nd Generation/V1.0/v2)



ESP8266 핀맵

- 아두이노 **IDE**의 핀 넘버에는 **GPIO** 넘버가 들어가야함
예) D1 → 5
D5 → 14

GPIO (General-purpose I/O)

사진 출처 : <http://www.c-sharpcorner.com/article/blinking-led-by-esp-12e-nodemcu-v3-module-using-arduinoide/>

Pin layout of LoLin NodeMCU V3

■ LoLin NodeMCU development board V3

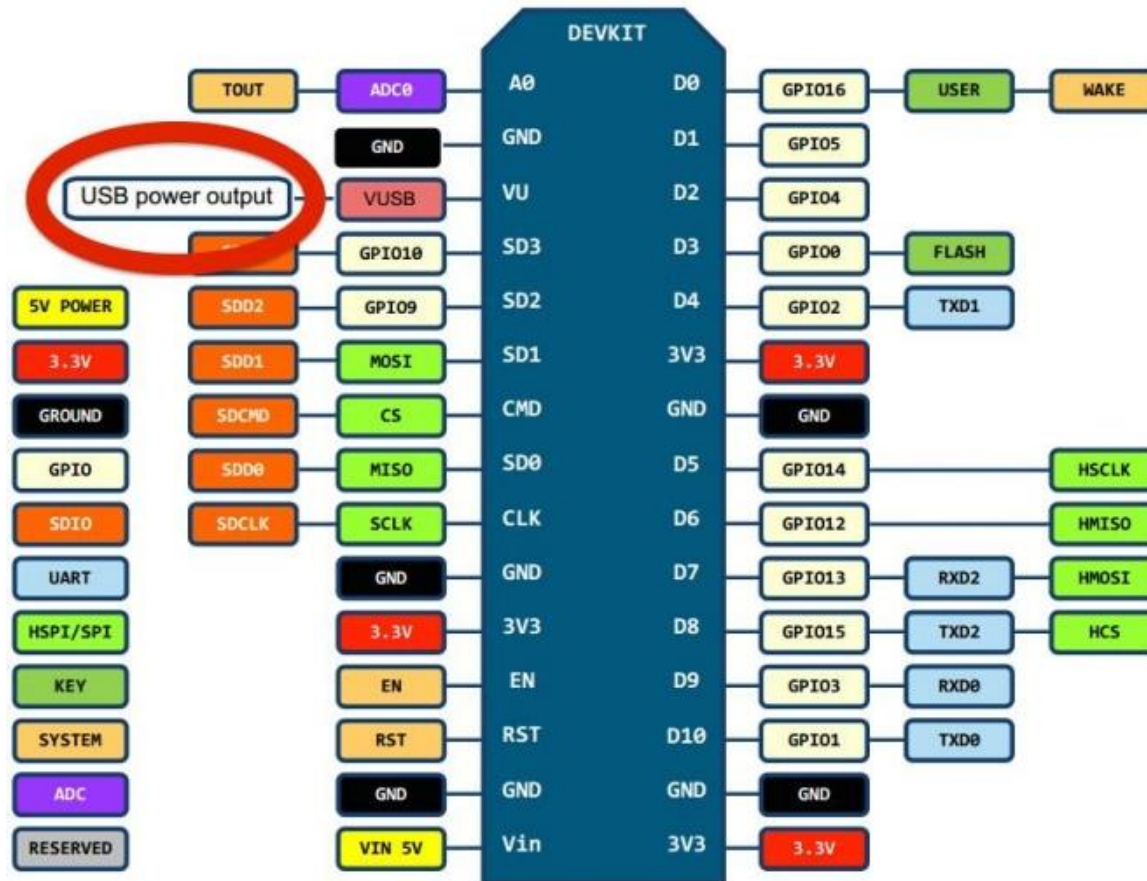


사진 출처 : <https://frightanic.com/blog/wp-content/uploads/2015/09/esp8266-nodemcu-dev-kit-v3-pins.jpg>

Arduino IDE 에서 ESP8266 핀맵 설정

- ESP8266 의 보드 핀번호 (예 :D0, D1) 을 아두이노 IDE에서 그대로 사용하려면 , 코드 작성시 아래의 코드를 프로그램에 삽입

```
#define D0 16
#define D1 5
#define D2 4
#define D3 0
#define D4 2
#define D5 14
#define D6 12
#define D7 13
#define D8 15
```

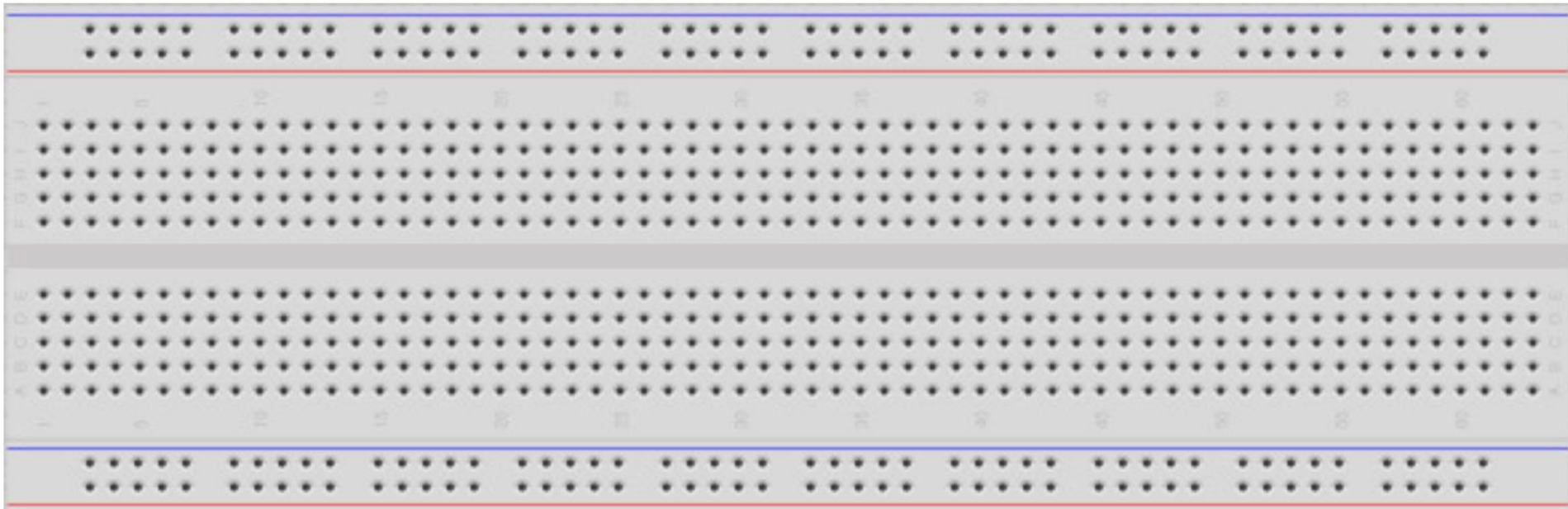


```
sketch_sep19a §
#define D0 16
#define D1 5
#define D2 4
#define D3 0
#define D4 2
#define D5 14
#define D6 12
#define D7 13
#define D8 15
|
void setup() {
  // put your setup code here, to run once:
}
```

NodeMCU 를 이용한 IoT 실습

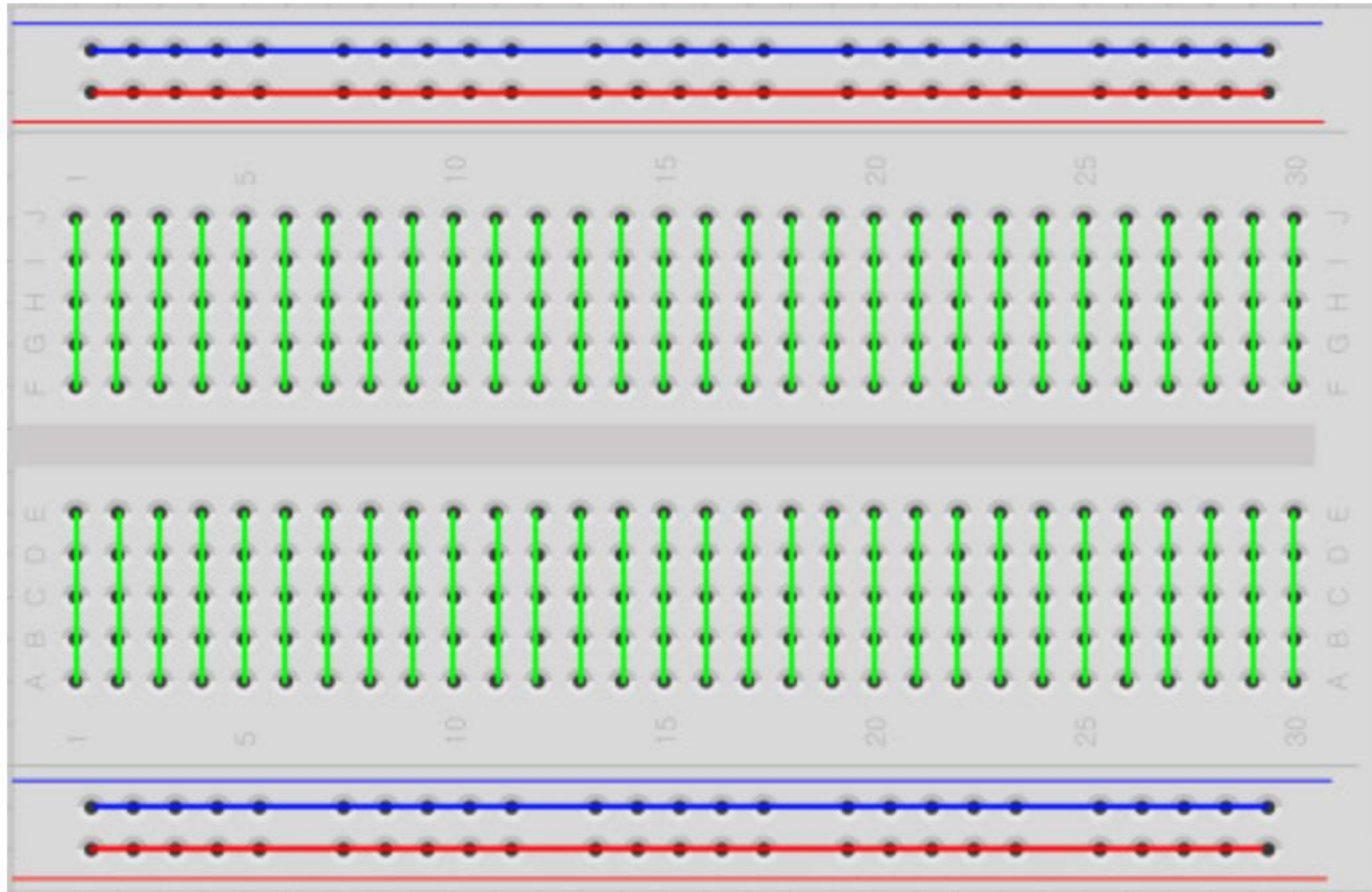
- [실습 1] GPIO 디지털 핀을 사용한 LED 제어
- [실습 2] GPIO 아날로그 핀을 사용한 조도센서 제어
- [실습 3] Serial 통신을 이용한 온습도 데이터 갖고 오기
- [실습 4] 릴레이 사용하여 장치제어하기

Bread Board



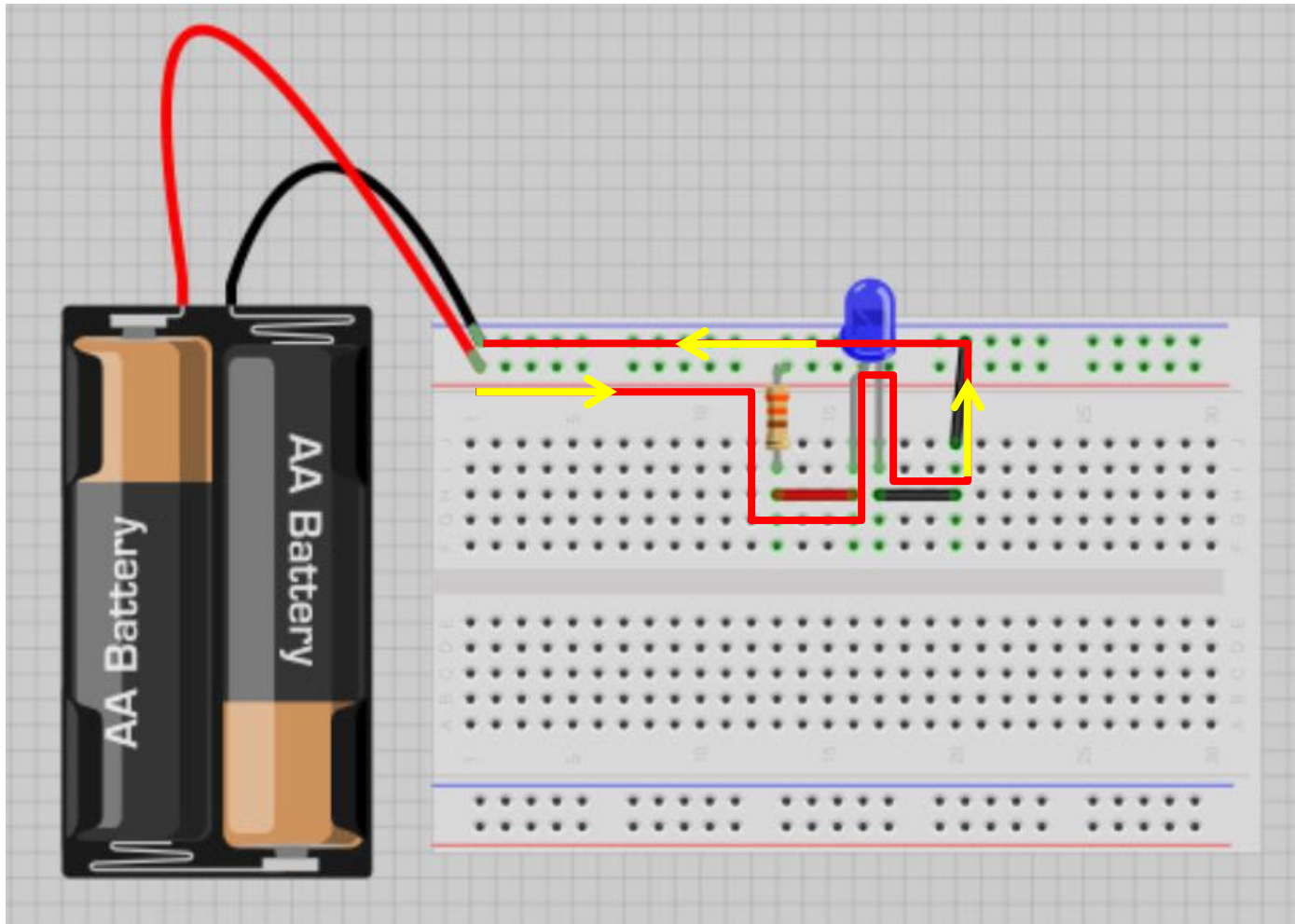
소스: <http://binworld.kr/13>

Bread Board



소스: <http://binworld.kr/13>

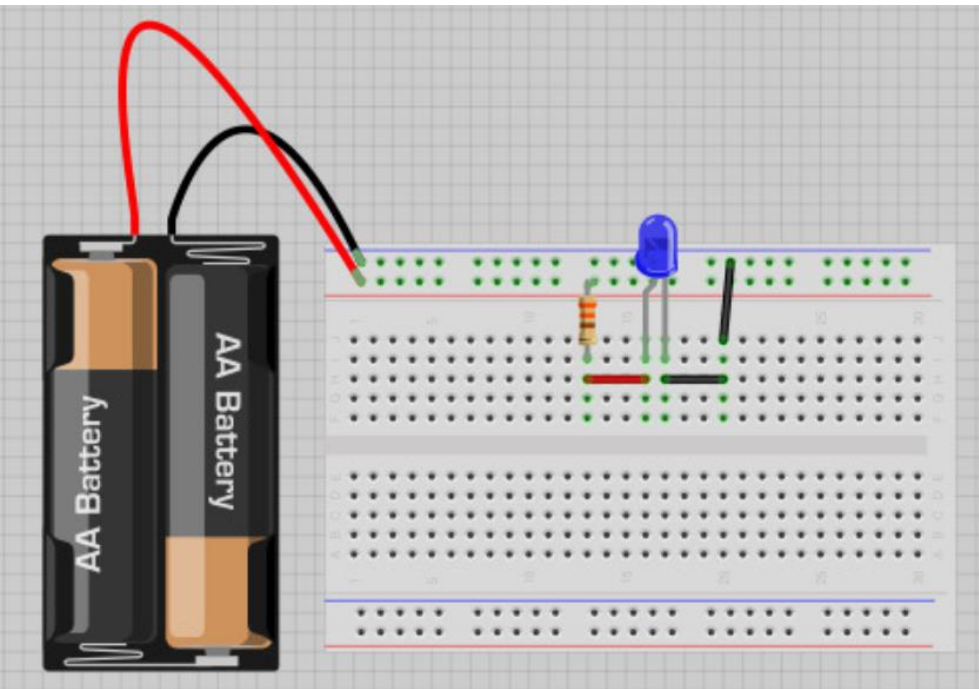
Bread Board



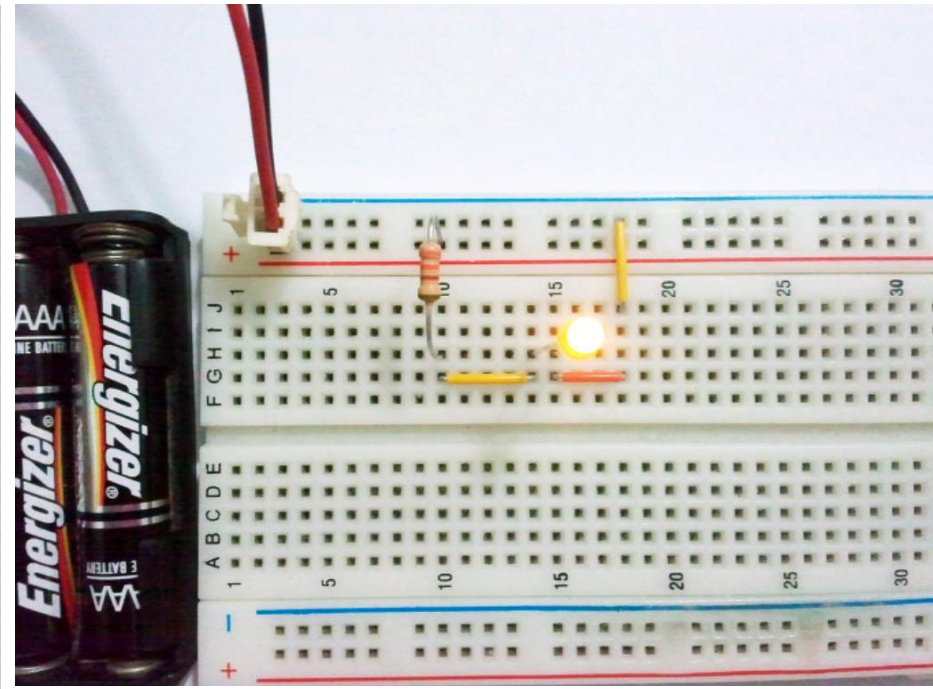
소스: <http://binworld.kr/13>

Bread Board

설계 예



구현 예



소스: <http://binworld.kr/13>

[실습 1] GPIO 디지털 핀을 사용한 LED 제어

■ 실습 목적

- 1. MCU GPIO의 임의의 핀을 출력핀으로 사용할 수 있다
- 2. 출력으로 설정된 핀에 HIGH/LOW의 신호를 가하여 소자(device)를 구동할 수 있다

■ 실습 시나리오

- LED를 1초마다 켜다 끄다 반복하기

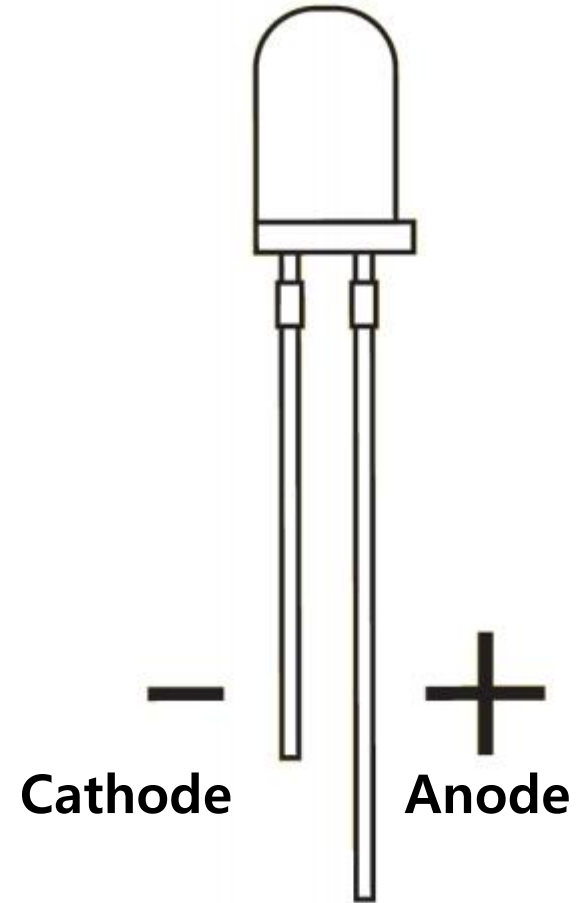
■ 실습 과정

- 1. NodeMCU D0(16)번 핀에 LED와 저항 연결
- 2. 아두이노 IDE에서 D0핀을 출력단자로 설정 → setup()
- 3. 13번 핀에 HIGH를 출력하고, 1초 대기, 그 다음 LOW출력하기, 1초 대기, 이러한 과정을 반복 → loop()

LED

■ LED소자

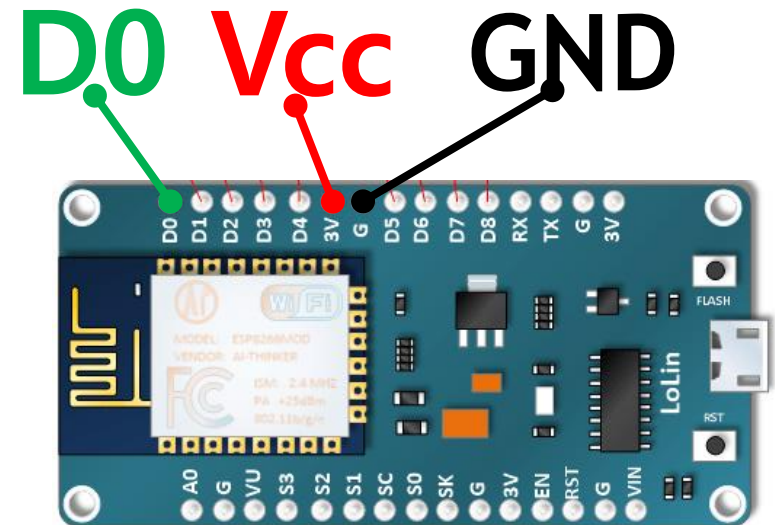
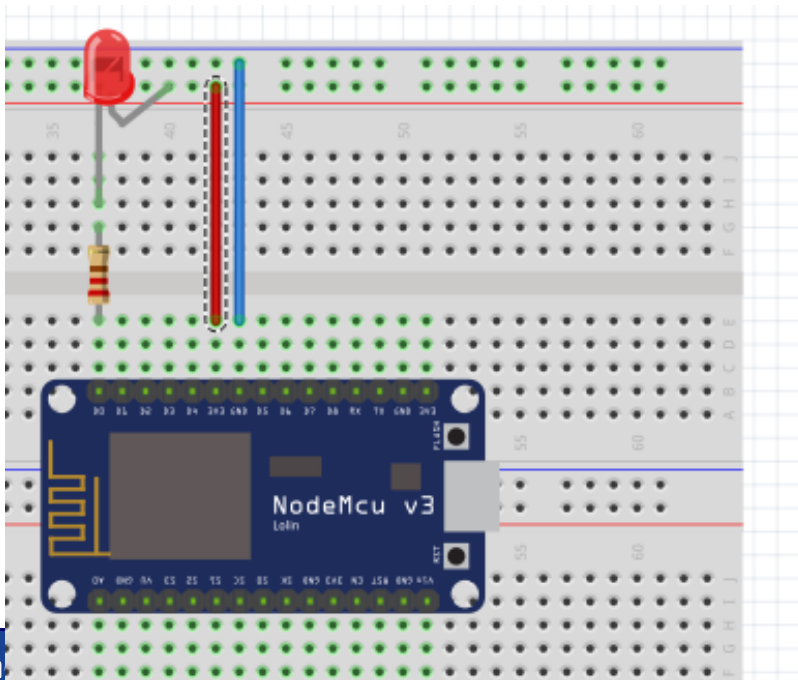
- 긴 단자가 Anode(+)
- 짧은 단자가 Cathode(-)



회로 구성 (실습 1)

■ LED 제어를 위한 회로 구성

- pin num(**D0**) 에 전선을 연결 후 저항 (220 옴) 과 LED 연결
- 이때 LED 의 anode 와 cathode 확인
- LED 의 cathode GND 와 연결



소스 코드(실습 1)

■ [Lab1_esp8266_led]

```
// GPIO Pin mapping 부분
// nodemcu 와 아두이노간 pin 이 달라서 초기 설정 필요
#define D0 16

void setup() { // setup 문은 제일 처음 한번만 실행됨
    // initialize digital pin D0 as an output.
    pinMode(D0, OUTPUT);
}

// the loop function runs over and over again forever
void loop() {
    digitalWrite(D0, HIGH); // turn the LED on (HIGH is the voltage level)
    delay(1000);             // wait for a second
    digitalWrite(D0, LOW);  // turn the LED off by making the voltage LOW
    delay(1000);             // wait for a second
}
```


소스코드 설명

■ 코드 설명

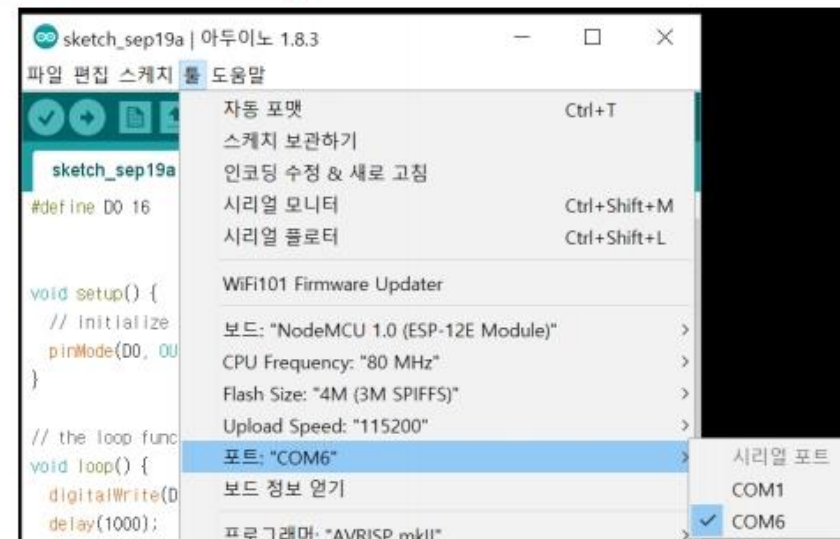
- 주석: 코드의 역할을 설명하는 부분, 한 줄은 `//` 여러 줄은 `/* */`을 이용
- **pinMode (num, dir)**: pin num의 입출력 방향을 정의함
(dir은 INPUT or OUTPUT)
예: **pinMode (D0, OUTPUT);** // D0 핀을 출력핀으로 설정한다.
- **digitalWrite(num, val)**: pinnum에 val의 값을 인가함
(val은 HIGH or LOW)
예: **digitalWrite (D0, HIGH);** // D0핀에 HIGH 전압을 인가한다.
- **Delay(msec)**: millisecond만큼의 시간 지연 후에 다음 실행문 실행
예: **delay(1000);** // 1초를 기다린다.

PC와 NodeMCU 간의 직렬포트 설정

- 컴퓨터에서 USB-to-Serial 케이블이 몇 번 COM 포트에 연결되었는지 포트 번호 확인
 - 장치관리자 열기
(윈10의 경우 윈도우키 +x 누른 후 장치관리자 선택)
 - 보이는 화면에서 직렬 포트 번호 확인 (이 경우는 COM6)

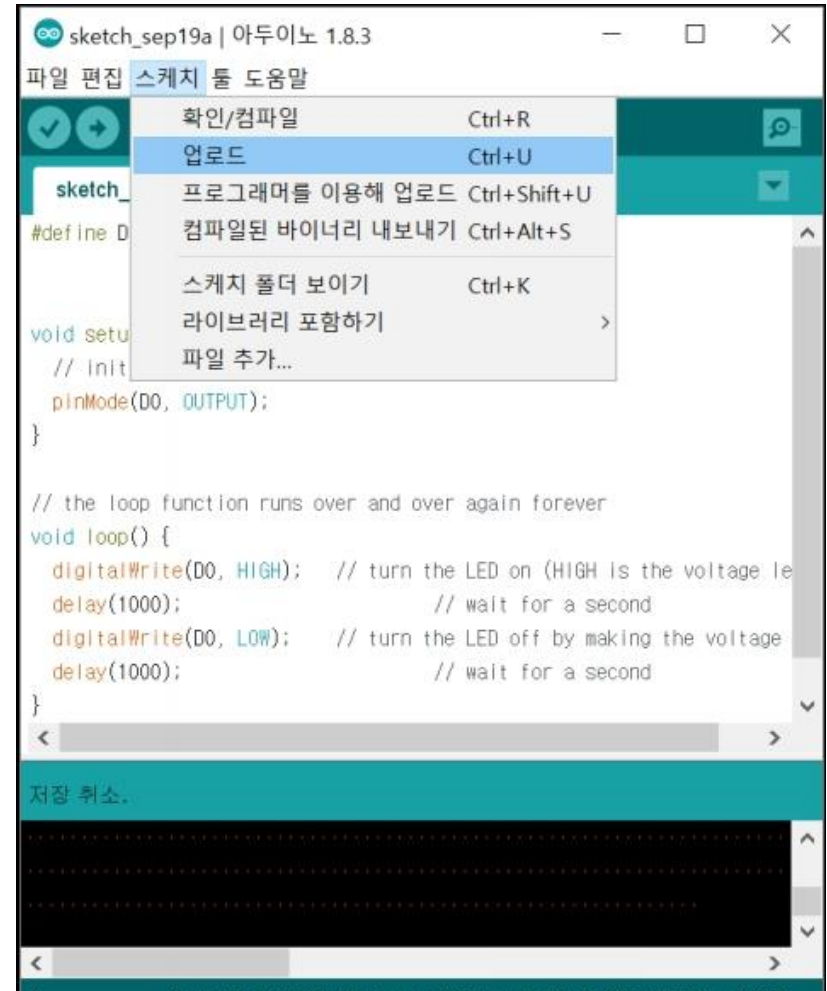


- 아두이노IDE에서
 - 툴 → 포트에서 위에서 찾은 COM 번호 선택
(본 화면의 경우 COM6)



컴파일 및 업로드

- “ 스케치 – 업로드 ”를 선택하여 코드를 컴파일하고 보드로 업로드
- 보드에서 **LED** 가 **1** 초마다 깜박거리는지 확인



[실습 2] 온습도 데이터 읽어오기

■ 목적

- 1.MCU GPIO 핀을 사용하여 온습도 데이터를 읽어올 수 있다
- 2.외부 라이브러리를 사용할 수 있다

■ 실습 시나리오

- 온습도 센서(DHT11)를 MCU의 GPIO에 연결하여 온습도 센서의 데이터를 읽어오기

■ 실습 과정

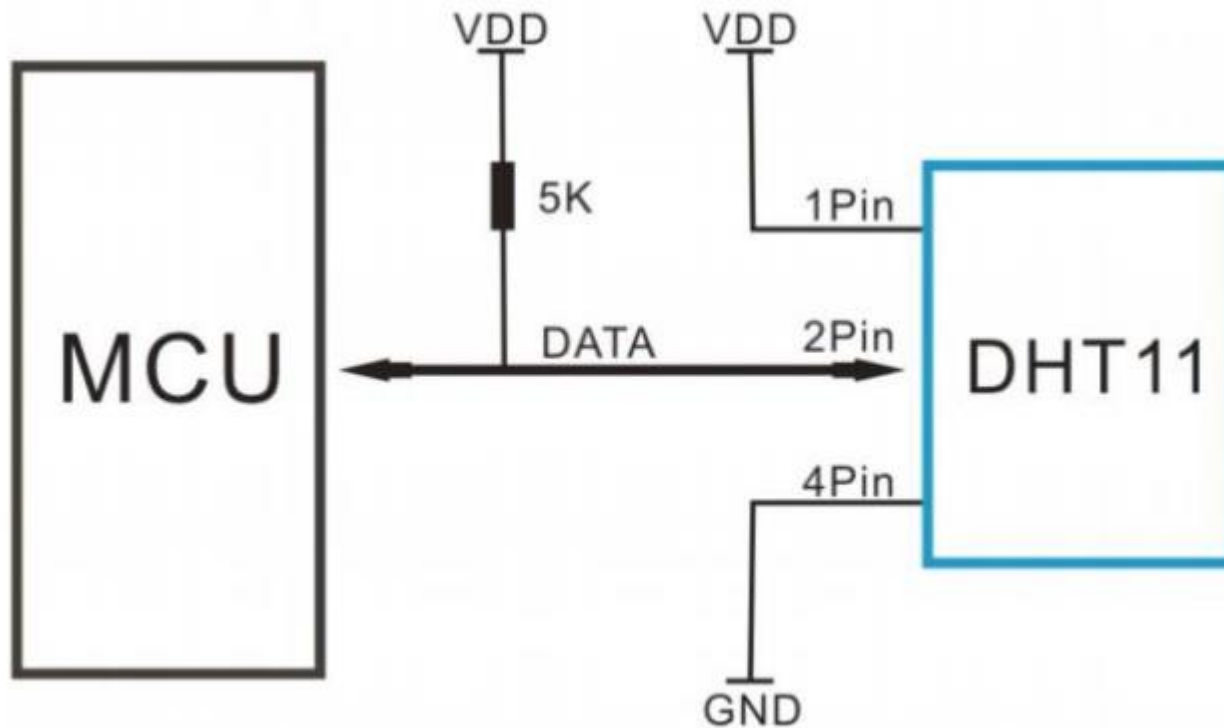
- 1. 온습도 센서는 3개의 핀으로 구성되어 있음:
VCC, GND, 데이터 핀
- 2. 전원은 각각 VCC와 GND에 연결하고 데이터 핀은 NodeMCU **D2** 핀에 연결
- 3. DHT11 용 라이브러리를 아두이노 **IDE**에 추가
- 4. DHT11 라이브러리를 활용하여 온습도 데이터 읽어오기

온습도 센서(DHT11)

- 매우 다양한 종류의 온습도 센서들이 있음. 여기에서는 실습용으로 간단히 사용할 수 있는 **DHT11** 소자 사용
- **DHT11** 소자의 특징
 - 하나의 데이터 핀으로 **MCU**와 데이터를 주고 받음
 - **DHT11** 소자를 사용하려면 **DHT11**의 통신 방식(프로토콜)을 분석한 후 일일이 그 기능을 프로그래밍하여야 함
- **DHT11 라이브러리**
 - **DHT11** 라이브러리에서 제공하는 함수를 사용하면 **DHT11** 소자로부터 온도 값과 습도 값을 손쉽게 읽어올 수 있음
 - 예: `dht.readHumidity()`, `dht.readTemperature()`

Typical Application of DHT11

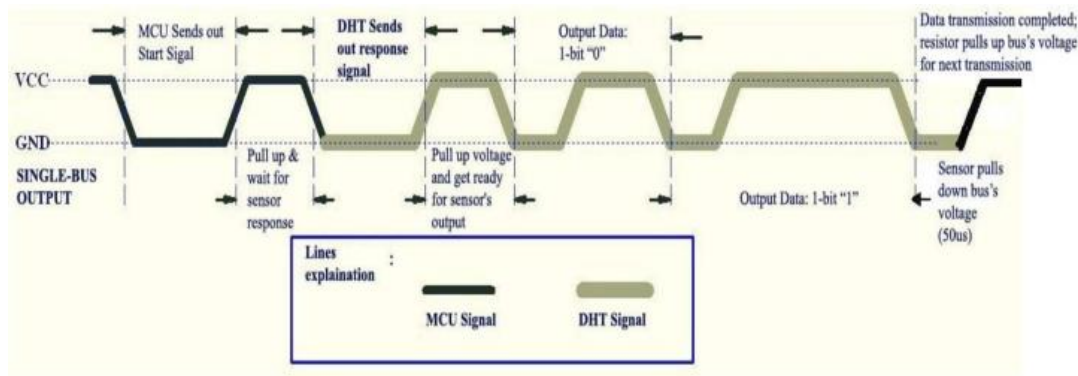
- MCU와 DHT11 (온습도 센서)와의 일반적 연결



Features of DHT11

■ Communication Process

- Single bus data format을 이용한 양방향 통신

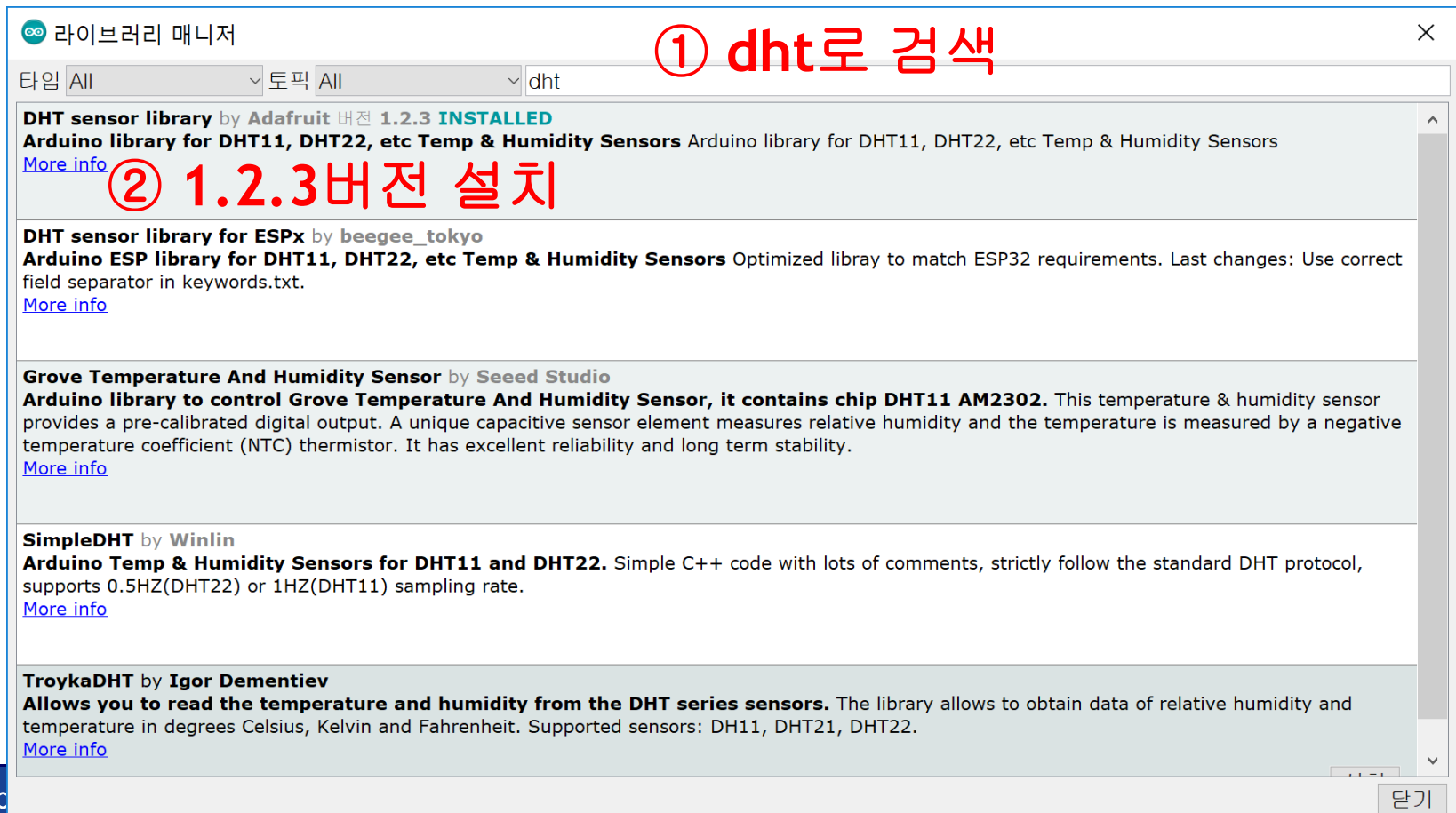


■ Electrical Characteristics

	Conditions	Minimum	Typical	Maximum
Power Supply	DC	3V	5V	5.5V
Current Supply	Measuring	0.5mA		2.5mA
	Average	0.2mA		1mA
	Standby	100uA		150uA
Sampling period	Second	1		

DHT11용 라이브러리 추가

- 메뉴-스케치-라이브러리포함하기-라이브러리 관리
- 라이브러리 매니저에서 dht 로 검색 후 설치



DHT11용 라이브러리

- 또는 라이브러리 zip파일이 있을시
 - “스케치” – “라이브러리 포함하기” – “.ZIP 라이브러리 추가” 선택
 - “DHT.zip” 선택 후 “열기”를 실행하면 DHT 라이브러리가 추가됨
- 프로그램 코드 내에서 DHT 라이브러리 사용하기

```
#include <DHT.h>
```

```
DHT dht(DHTPIN, DHTTYPE); // DHT 객체 선언
```

```
float humidity = dht.readHumidity(); // 습도값 읽어오기
```

```
float temperature = dht.readTemperature(); // 온도값 읽어오기 ( 섭씨 )
```

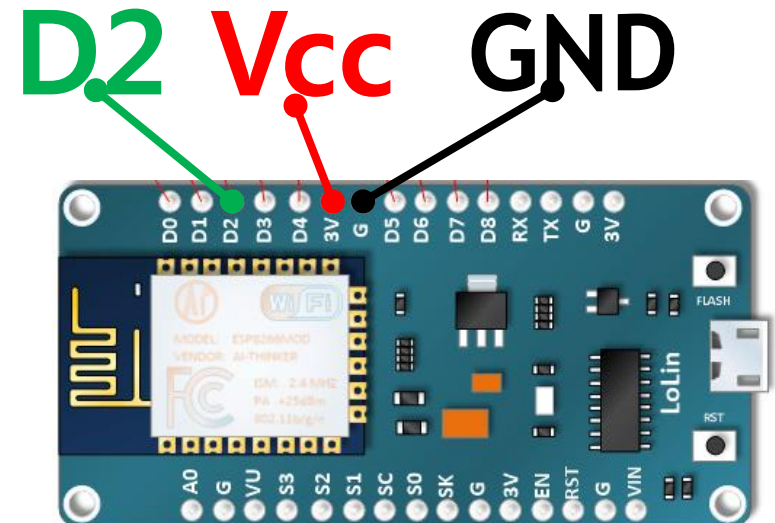
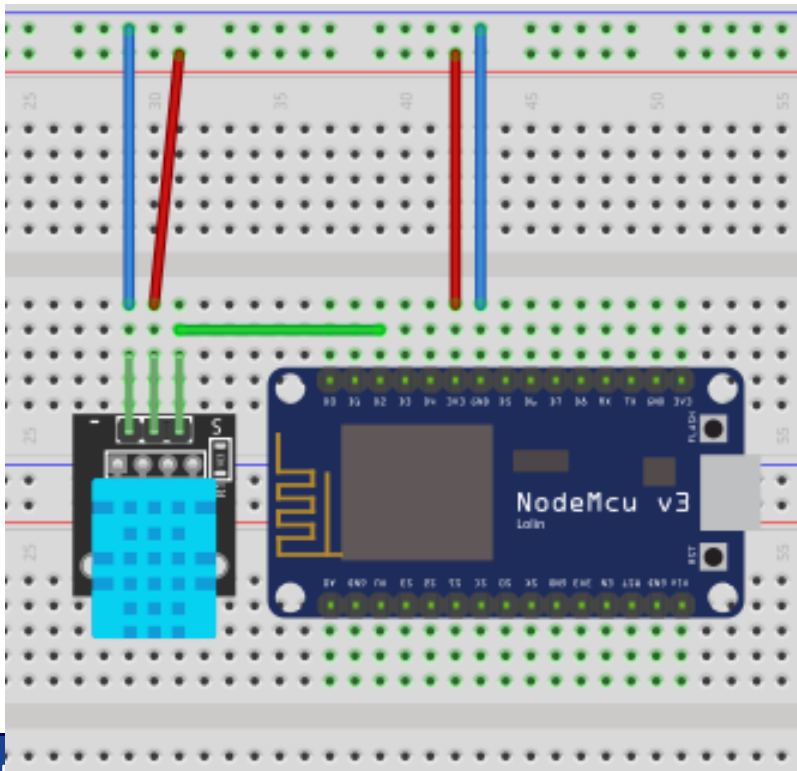
```
float fahrenheit = dht.readTemperature(true); // 온도값 읽어오기 ( 화씨 )
```

```
위의 세 함수 모두 read 실패시 nan(Not A Number) 반환
```

회로 구성 (실습 2)

■ 온습도 센서와 NodeMCU 연결

- DHT11 의 VCC, GND 단자를 전원 VCC, GND 단자에 연결
- DHT11 의 데이터 핀을 NodeMCU **D2** 에 연결



소스코드 (실습 2) - 1/2

- 프로젝트 이름: “esp8266_led.ino”
 - DHT 객체 선언 후 read Humidity 와 read Temperature 를 이용하여 온습도를 읽음
 - Serial을 이용하여 시리얼 모니터에 출력
- [Lab2_esp8266_DHT]

```
#include "DHT.h"
#define D2 4
#define DHTPIN D2 //digital pin
#define DHTTYPE DHT11 //dht module
DHT dht(DHTPIN, DHTTYPE);
float humidity, temperature;
void setup() {
  Serial.begin(9600);
  dht.begin();
}
```

소스코드 (실습 2) - 2/2

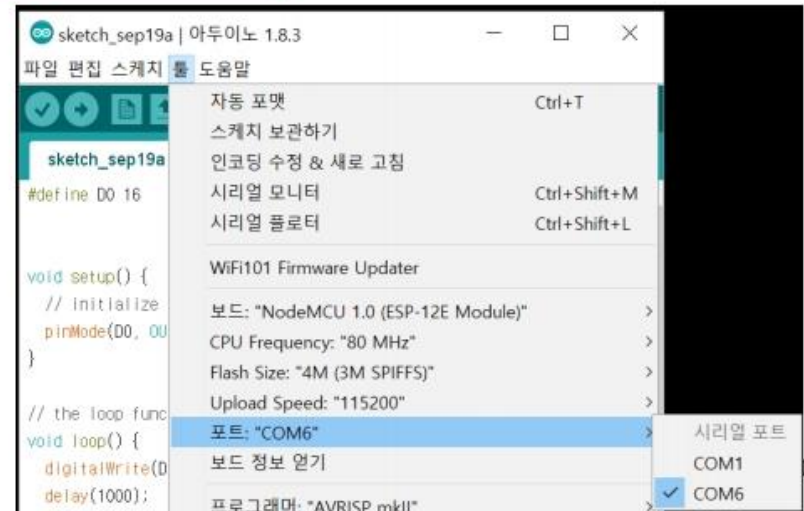
```
void loop() {  
    delay(2000); // Wait for a few seconds between measurements  
  
    humidity = dht.readHumidity();  
    temperature = dht.readTemperature();  
  
    if(!isnan(humidity) && !isnan(temperature)){  
        Serial.print ("온도 = ");  
        Serial.print(temperature);  
        Serial.print (" 습도 = ");  
        Serial.println(humidity);  
    } else{  
        Serial.println("Failed to read from DHT sensor");  
        return;  
    }  
}
```

PC 와 NodeMCU 간의 직렬포트 설정

- 컴퓨터에서 USB-to-Serial 케이블이 몇 번 COM 포트에 연결되었는지 포트 번호 확인
 - 장치관리자 열기(윈10의 경우 윈도우키+x 누른 후 장치관리자 선택)
 - 보이는 화면에서 직렬포트 번호 확인(이 경우는 COM6)

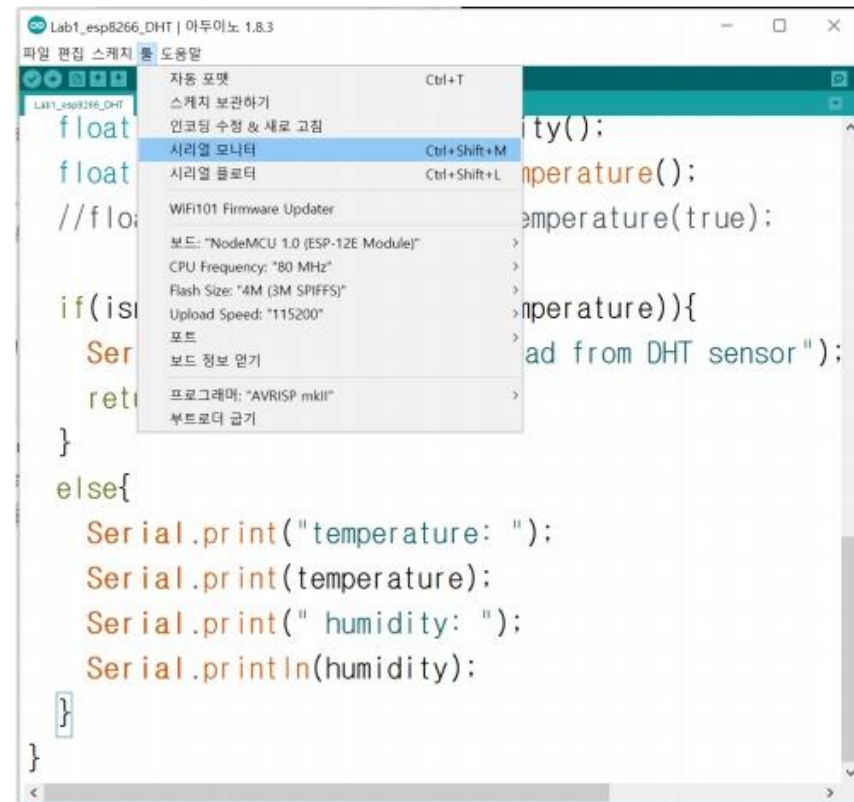


- 아두이노IDE에서
 - 툴→포트에서 위에서 찾은 COM 번호 선택(본 화면의 경우 COM6)



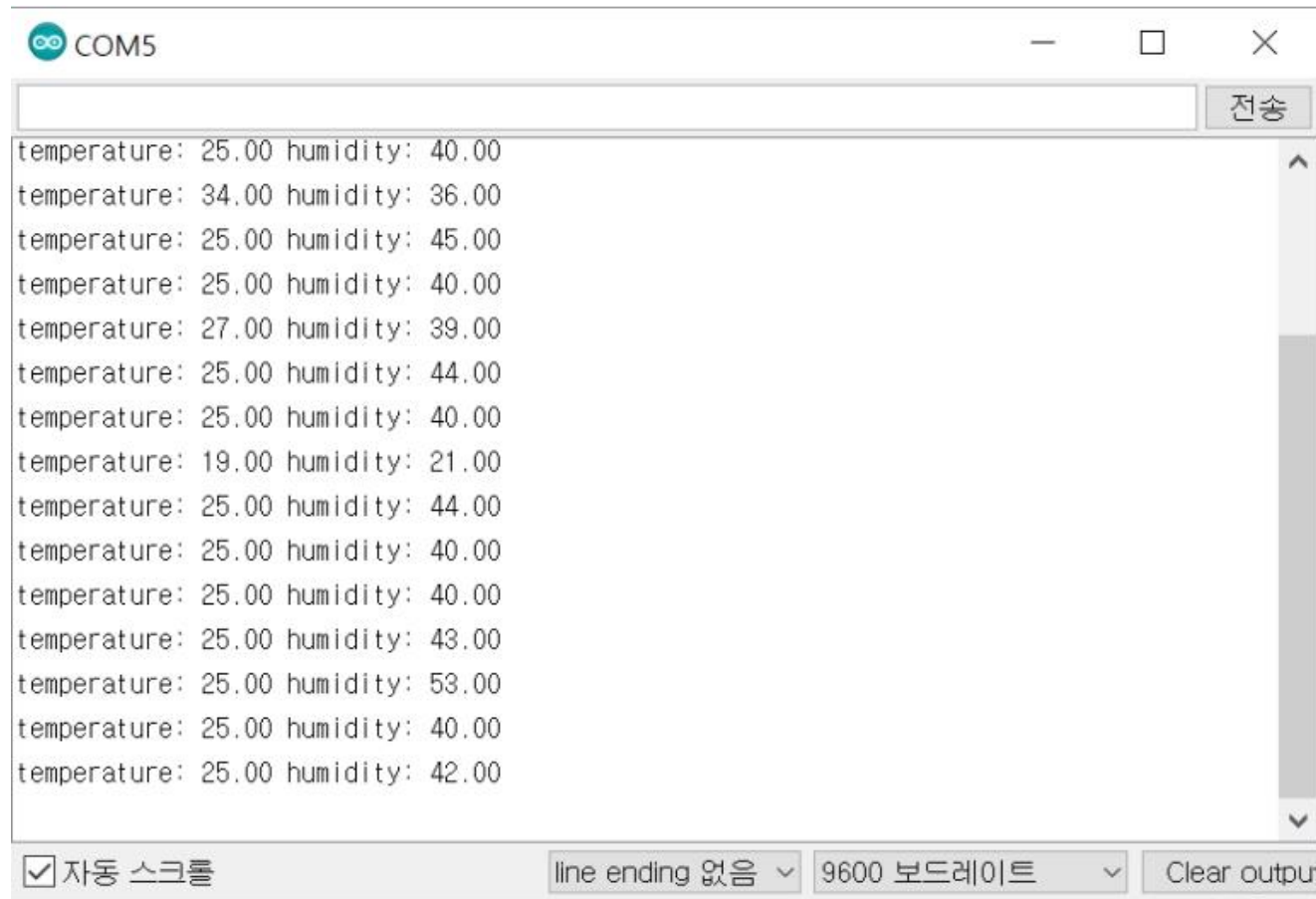
컴파일 및 업로드 (실습 2)

- “ 스케치 – 업로드 ”를 선택하여 코드를 컴파일하고 보드로 업로드
- 성공적으로 업로드 되었다면, “툴 - 시리얼 모니터” 선택



실행결과 (실습 2)

■ 아두이노 IDE 에서 “툴 - 시리얼모니터” 선택



[실습 3] 조도 센서값 읽어오기

■ 목적

- 1. MCU GPIO의 아날로그핀을 사용하여 조도 센서 값을 읽어올 수 있다
- 2. 디지털 값과 아날로그 값의 차이를 이해한다

■ 실습시나리오

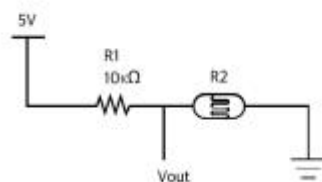
- 조도 센서를 MCU의 아날로그핀에 연결하여 빛의 밝기 값을 읽어오기

■ 실습과정

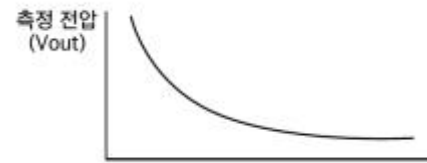
- 1. 조도 센서는 2개의 핀으로 구성되어 있고 극성(+,-)이 없음
- 2. 조도 센서의 한 핀을 NodeMCU A0에 연결하고, 다른 한 핀은 GND에 연결하기
- 3. 저항(4.7K)의 한쪽 단자를 NodeMCU A0에 연결하고, 한 단자는 VCC에 연결하기
- 4. 시리얼 모니터를 통해 읽어오는 빛의 밝기 값 확인하기

[실습 3] 조도 센서값 읽어오기

- 조도센서 (아날로그 입력)
 - 광에너지 (빛) 를 받으면 내부의 전자가 활성화 되어 저항 값이 변한다 .
 - 광에너지가 높을수록 저항이 작아지고 어두울수록 저항이 올라간다 .
 - 풀업 저항을 사용하여 회로를 구성한다 .



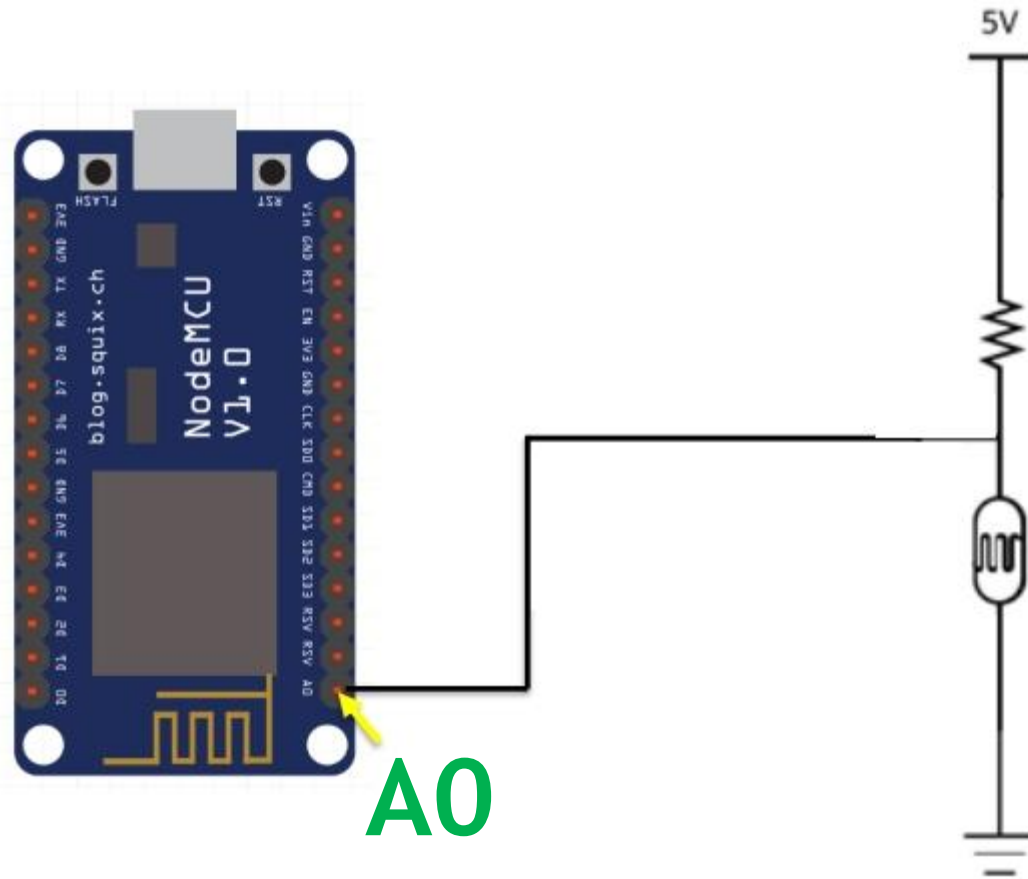
풀업 저항 사용



풀업 저항 사용시 밝기에 대한 측정 전압

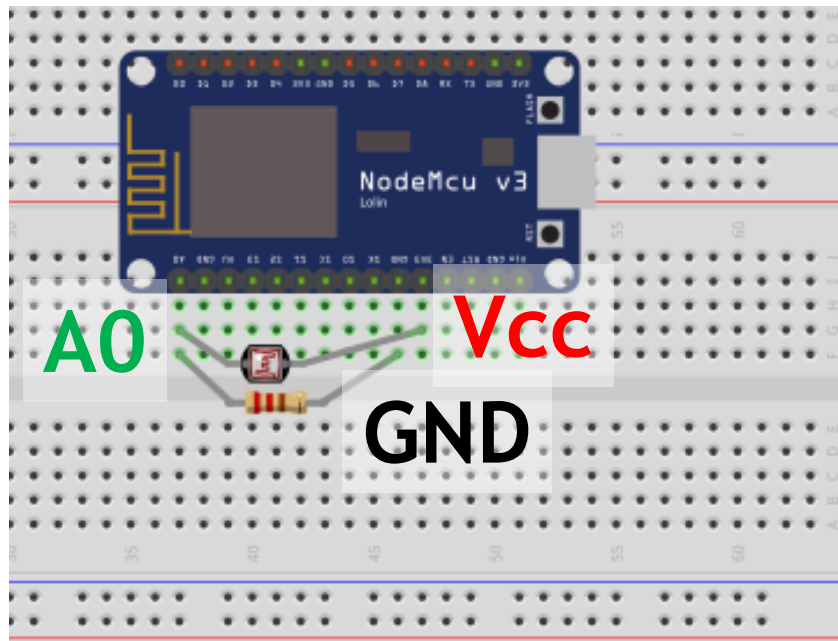
- 아날로그 입력
 - `analogRead(pinNum)` 함수를 이용하여 0~1023 사이의 값을 A0 핀에서 받는다 .

조도센서 연결 회로도



회로 구성 (실습 3)

- 조도 (CDS) 센서와 NodeMCU 연결
 - 조도센서의 한쪽은 NodeMCU **A0**에 , 한쪽은 **VCC** 에 연결
 - 저항 (4.7K) 의 한쪽은 NodeMCU **A0**에 , 다른 한쪽은 **GND** 에 연결



소스코드 (실습 3)

■ [Lab3_esp8266_light_sensor]

- `analogRead(pinNum)` 을 이용하여 값을 받은 뒤
- `Serial.print(val)` 을 이용하여 시리얼 모니터에 출력

```
// CDS 조도센서를 이용하여 빛의 밝기 읽어오기
#define LIGHT_PIN A0
int light_val;

void setup() {
  pinMode(LIGHT_PIN, INPUT);
  Serial.begin(9600);
}

void loop() {
  light_val = analogRead(LIGHT_PIN);
  Serial.print("Light intensity = ");
  Serial.println(light_val);
  delay(1000);
}
```

컴파일 및 업로드 (실습 3)

- " 스케치 - 업로드 "를 선택하여 코드를 컴파일하고 보드로 업로드
- 성공적으로 업로드되었다면, " 툴 - 시리얼 모니터 " 선택
- 시리얼 모니터 화면에 오른 쪽 그림의 결과 확인
 - 빛이 어두어지면 값이 커지게 됨



The screenshot shows a serial monitor window titled 'COMS'. It displays a list of light intensity values, each preceded by the word 'Light :'. The values are: 437, 439, 438, 436, 432, 429, 426, 430, 432, 431, 429, 498, 807, and 825. The window has a search bar at the top right with the text '전송' (Transmit) and a scroll bar on the right. At the bottom, there are checkboxes for '자동 스크롤' (Auto scroll) and 'line ending 없음' (No line ending), and a dropdown menu for '9600 보드레이트' (9600 baud rate). A 'Clear output' button is also visible.

```
COMS
Light : 437
Light : 439
Light : 438
Light : 436
Light : 432
Light : 429
Light : 426
Light : 430
Light : 432
Light : 431
Light : 429
Light : 498
Light : 807
Light : 825
```

[실습 4] 조도센서 값에 따라 LED 램프 점등

■ 실습 목적

- 1. 릴레이가 무엇인지 이해하고 , 릴레이를 사용하여 일반 가전기기를 제어할 수 있다 .
- 2. 몇 개의 소자를 연동하여 지능적인 동작을 구현할 수 있다

■ 실습 시나리오

- 릴레이에 USB 램프를 연결하고 , 조도센서와 연동하여 어두워지면 자동으로 LED 램프가 켜지고 , 밝으면 꺼지도록 한다

■ 실습 과정

- 1. 조도센서는 이전실험과 동일하게 A0 에 연결
- 2. 릴레이 신호 단자는 NodeMCU D1 에 연결
- 3. 릴레이 출력 부분에는 USB LED 램프를 연결
- 4. 조도값이 일정 값으로 작아지면 (어두워진 경우), 릴레이 신호단자에 HIGH 신호를 가해 LED 램프 불켜기

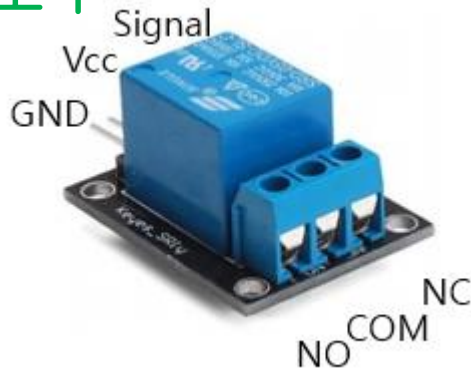
릴레이

■ 릴레이 모듈 (디지털 출력)

- 릴레이 모듈은 3 개의 전선 연결부분 (전원 및 기기연결) 과 신호 부분 (esp8266 연결) 으로 나뉨

- 3 개의 전선 COM(공통단자), NO(Normal Open), NC(Normal Close) 로 구성됨
- 3 개의 신호 부분은 Vcc, GND, Signal 로 나뉨

신호부



전선부

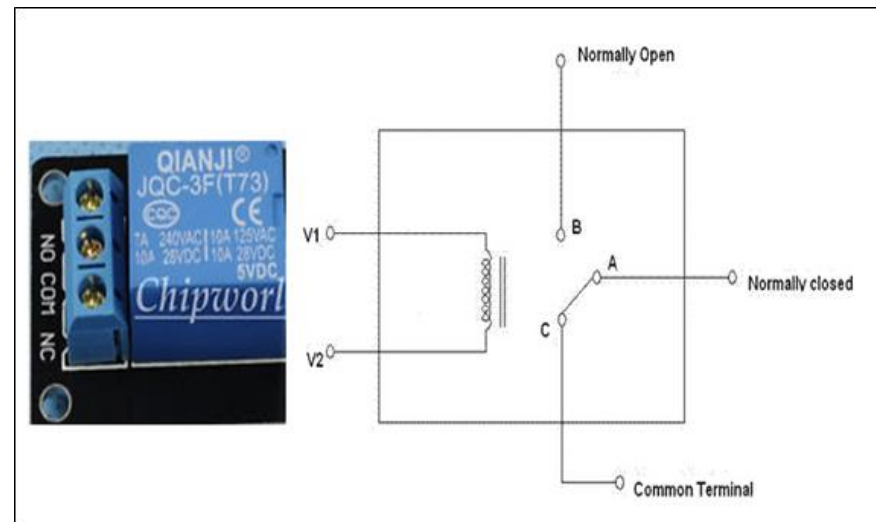
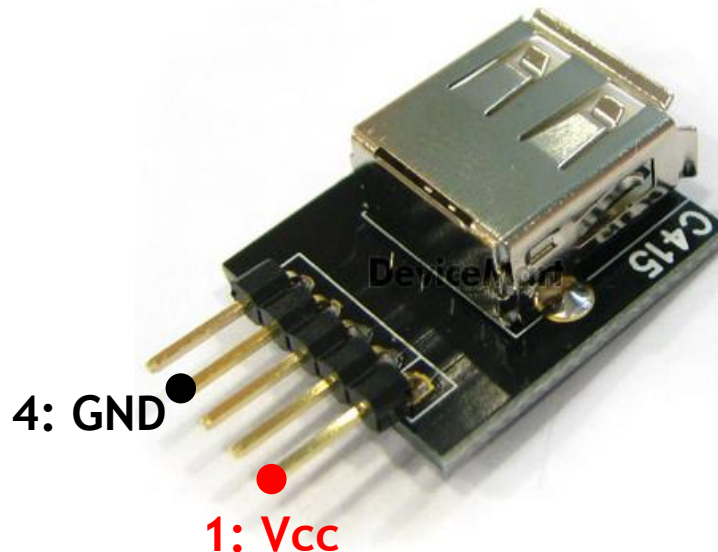


사진 출처 : <https://www.kocoafab.cc/make/view/184>

USB 커넥터

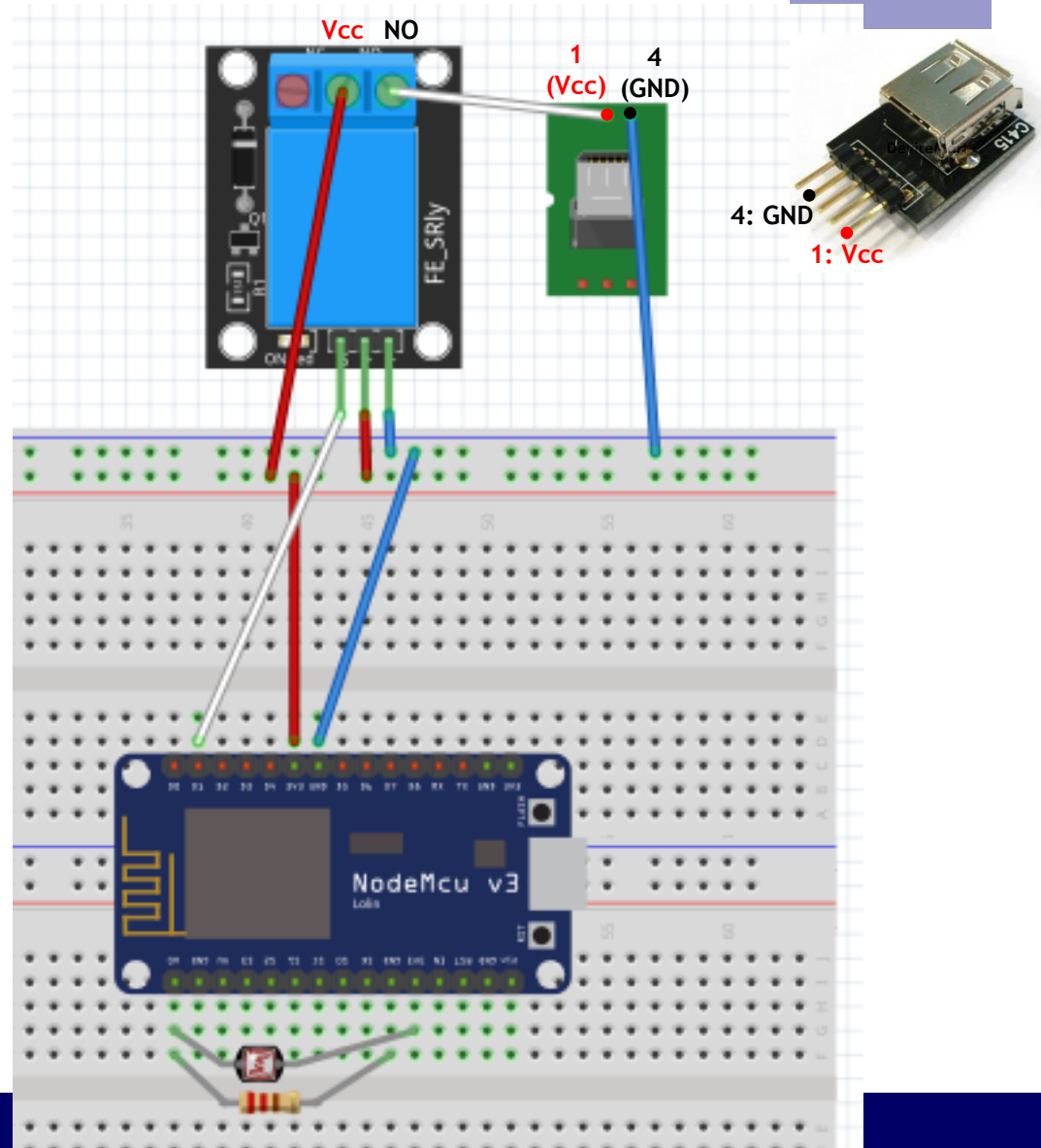
■ USB 커넥터

- USB LED 램프를 연결하기 위해 사용
- 핀맵은 아래와 같고 , VCC 는 NodeMCU VCC 에 , GND 는 릴레이 모듈의 NO 단자에 연결



회로 구성 (실습 4)

- 조도센서 + 릴레이
+ **USB LED 램프**
 - 조도센서를 이용하여
현재 밝기를
확인한다 .
 - 확인한 값에 따라
Relay 모듈의 전원을
ON/OFF 한다 .



소스코드 (실습 4) - 1/2

■ [Lab4_esp8266_Relay_USBLED]

```
#define D1 5
#define LIGHT_PIN A0
#define RELAY1_PIN D1
#define RELAY_OFF LOW
#define RELAY_ON HIGH

int lightValue;
int relay_state = RELAY_OFF; // 릴레이 OFF
void setup() {
    Serial.begin(9600);
    delay(10);
    pinMode(LIGHT_PIN, INPUT); // 조도센서 입력단자 설정
    pinMode(RELAY1_PIN, OUTPUT); // 릴레이 SIGNAL 단자 설정
    digitalWrite(RELAY1_PIN, RELAY_OFF); // LED OFF
}
```

소스코드 (실습 4) - 2/2

■ [Lab4_esp8266_Relay_USBLEDD]

```
void loop() {  
    lightValue = analogRead(LIGHT_PIN);  
    if(lightValue < 500){ // 조도 값이 낮으면 Relay ON  
        relay_state = RELAY_ON;  
    }  
    else if(lightValue > 500){ // 조도 값이 높으면 Relay OFF  
        relay_state = RELAY_OFF;  
    }  
    digitalWrite(RELAY1_PIN, relay_state);  
    Serial.println(lightValue);  
    delay(1000);  
}
```

컴파일 및 업로드 (실습 4)

- “ 스케치 - 업로드 ”를 선택하여 코드를 컴파일하고 보드로 업로드
- 성공적으로 업로드 되었다면, “ 툴 - 시리얼 모니터 ” 선택하여 조도 값 모니터링
- 조도센서에 빛을 가리면 조도 값이 작아지면서, **Relay** 가 **ON**되고, **LED** 램프에 불이들어옴



The screenshot shows a serial monitor window titled 'COM5'. The window displays a list of light intensity readings, each preceded by the word 'Light :'. The values are: 437, 439, 438, 436, 432, 429, 426, 430, 432, 431, 429, 498, 807, and 825. The window has a scrollbar on the right and a status bar at the bottom. The status bar includes a '자동 스크롤' (Auto Scroll) checkbox which is checked, a 'line ending 없음' (No line ending) dropdown menu, a '9600 보드레이트' (9600 baud rate) dropdown menu, and a 'Clear output' button.

```
COM5
Light : 437
Light : 439
Light : 438
Light : 436
Light : 432
Light : 429
Light : 426
Light : 430
Light : 432
Light : 431
Light : 429
Light : 498
Light : 807
Light : 825
```

< >
☒ 자동 스크롤 line ending 없음 9600 보드레이트 Clear output