

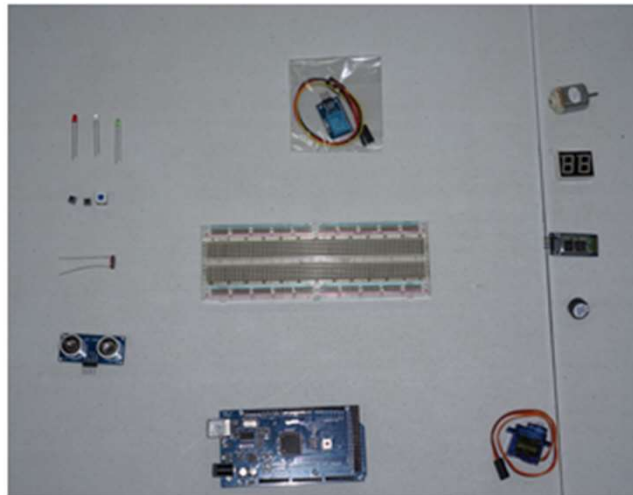
# 창의 융합 설계

## - Lesson 01

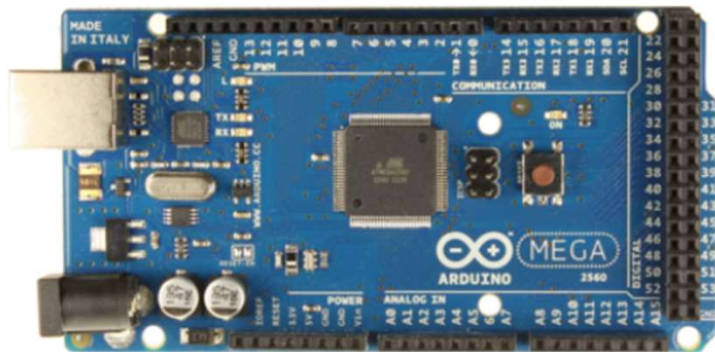
정중화

godopu16@gmail.com

# Arduino 부품 알아보기



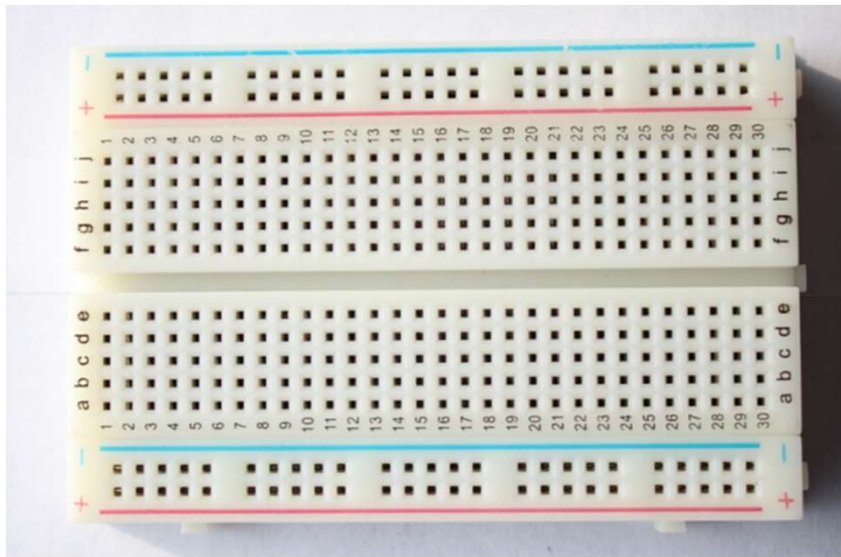
# Arduino 부품 알아보기



## ● Arduino MEGA 2560

Microcontroller	<a href="#">ATmega2560</a>
Operating Voltage	5V
Input Voltage (recommended)	7-12V
Input Voltage (limit)	6-20V
Digital I/O Pins	54 (of which 15 provide PWM output)
Analog Input Pins	16
DC Current per I/O Pin	20 mA
DC Current for 3.3V Pin	50 mA
Flash Memory	256 KB of which 8 KB used by bootloader
SRAM	8 KB
EEPROM	4 KB
Clock Speed	16 MHz
LED_BUILTIN	13
Length	101.52 mm
Width	53.3 mm
Weight	37 g

# Arduino 부품 알아보기



- Breadboard

- 전자 회로의 시제품을 만드는 데 사용하고 재사용할 수 있는 무땀납 장치

# Arduino 부품 알아보기



- LED(Light emitting diode)
  - 발광 다이오드로 불빛을 발하는 장치
  - +, - 극성 있음
    - 잘못 끼을 시 제품 고장

# Arduino 부품 알아보기



- SWITCH

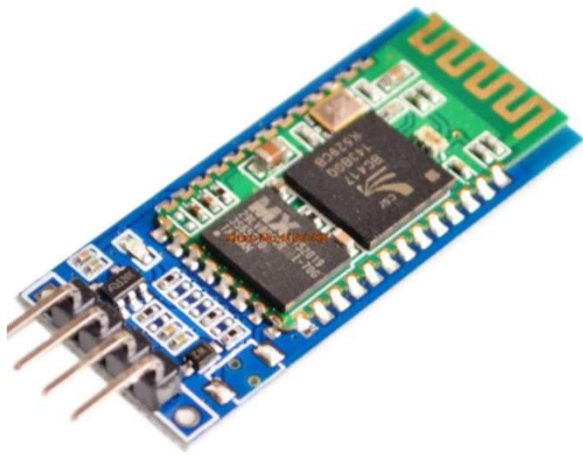
- 전기회로의 개폐나 접속상태를 변경하기 위해서 사용하는 기구

# Arduino 부품 알아보기



- 조도 센서 (illuminance sensor)
  - 빛의 양을 측정하는 센서
    - 일몰 때 가로등이 자동으로 켜지는 것이 이 센서를 이용

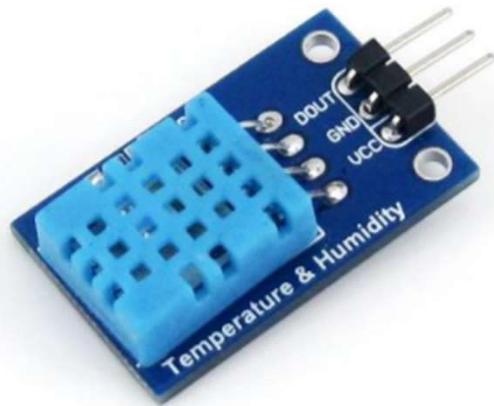
# Arduino 부품 알아보기



- 블루투스 모듈 (Bluetooth module)
  - 블루투스 통신을 하기 위해 사용하는 모듈



# Arduino 부품 알아보기



- 온습도 센서 (DHT sensor)
  - 온도와 습도를 측정하는 센서

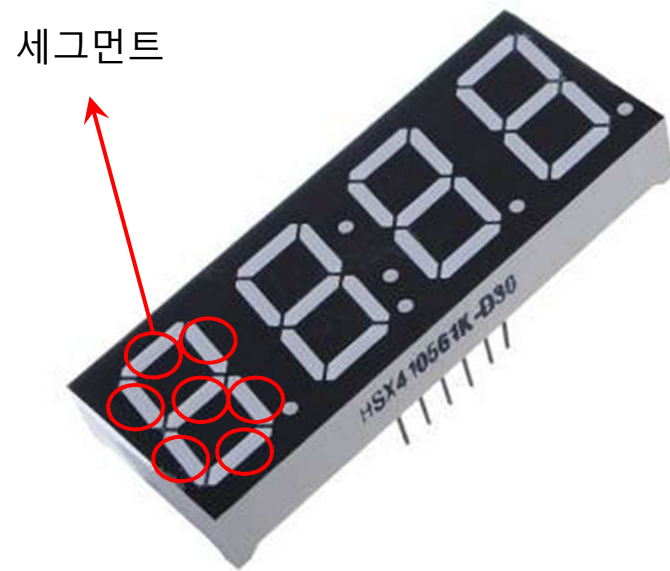
# Arduino 부품 알아보기



- 모터 (Motor)

- 전력을 받아서 회전하는 장치

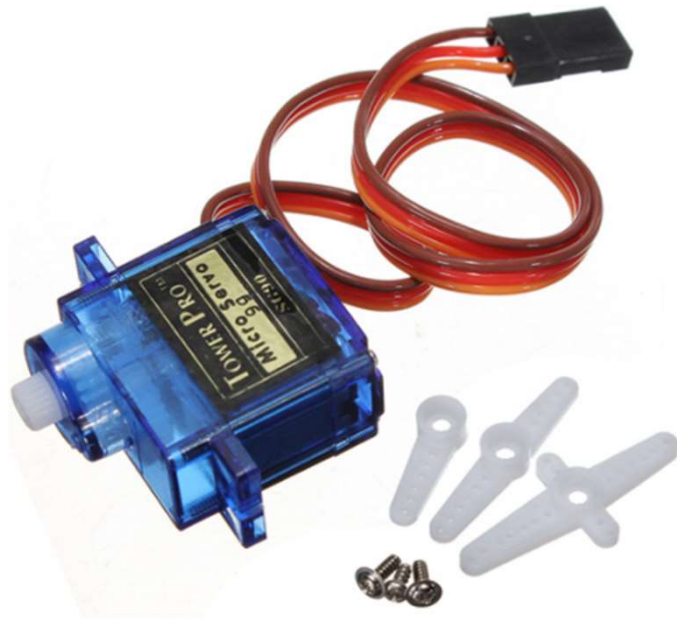
# Arduino 부품 알아보기



- 7세그먼트 (FND)

- 최대 7개의 세그먼트로 숫자를 표시하는 소자

# Arduino 부품 알아보기



- 서보모터 (Servo motor)

- 전압 입력을 회전각으로 바꾸기 위해 사용되는 전동기

# Arduino 부품 알아보기



- 피에조 부저

- 전압 입력을 받아 소리를 내는 장치

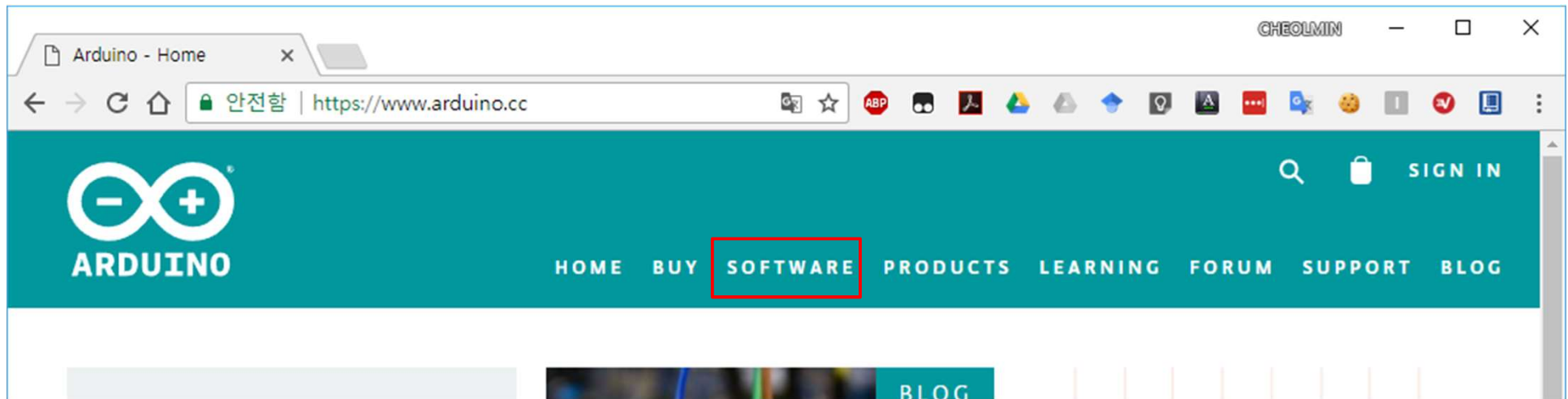
- 일종의 스피커라 생각할 수 있음

# Arduino

- 오픈 소스 전자 공학 플랫폼
  - 하드웨어 (전기, 전자 부품)와 소프트웨어를 쉽게 사용하게 해 줌
  - 아두이노 보드 (Uno, Mega, ...) 는 센서, 버튼, Twitter 메시지 등을 입력받아 (input) 모터나 LED혹은 인터넷을 이용하여 출력 (output) 할 수 있음
  - 취미, IoT, 교육, 웨어러블, 3D 프린팅 등에 사용 가능
- 자세한 것은 <https://www.arduino.cc> 에서 찾아볼 수 있음

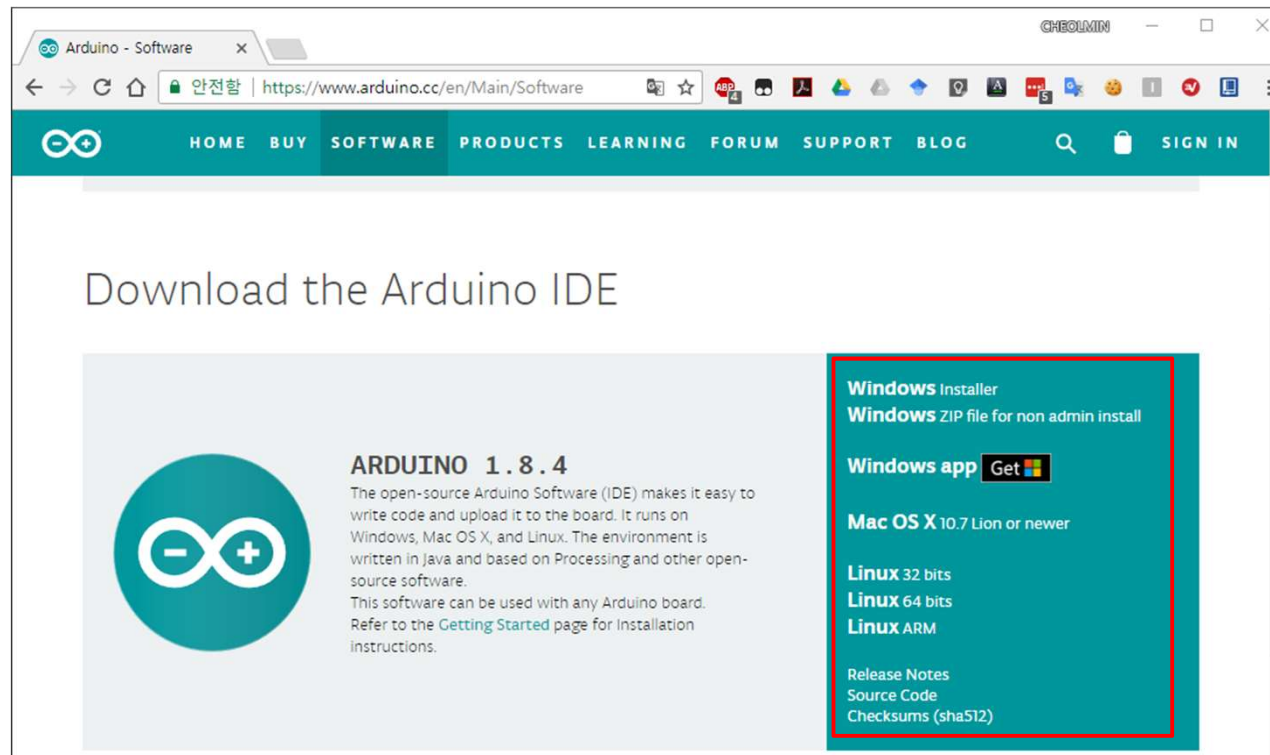
# Arduino IDE 설치

- <https://www.arduino.cc> 접속
  - Software 클릭



# Arduino IDE 설치

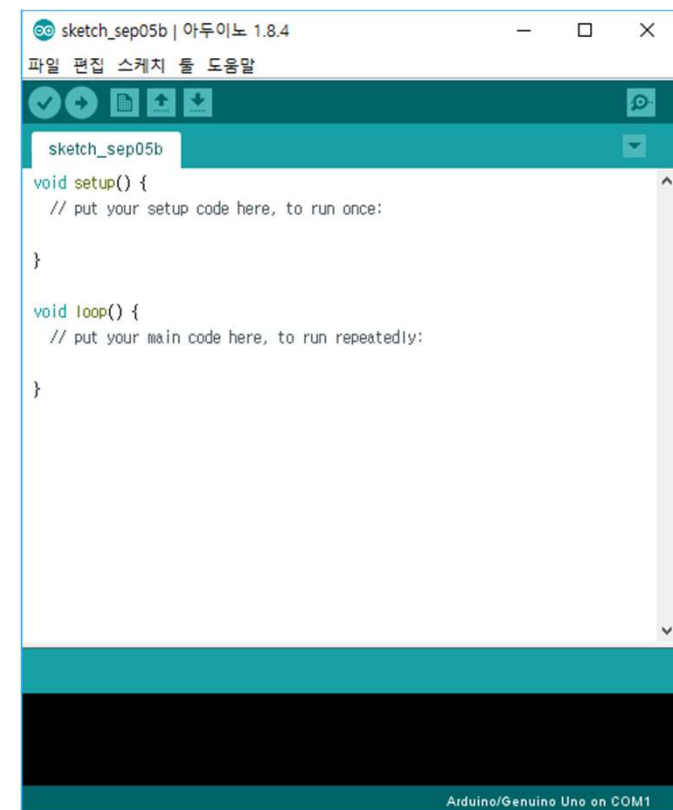
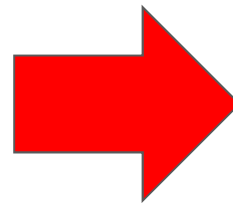
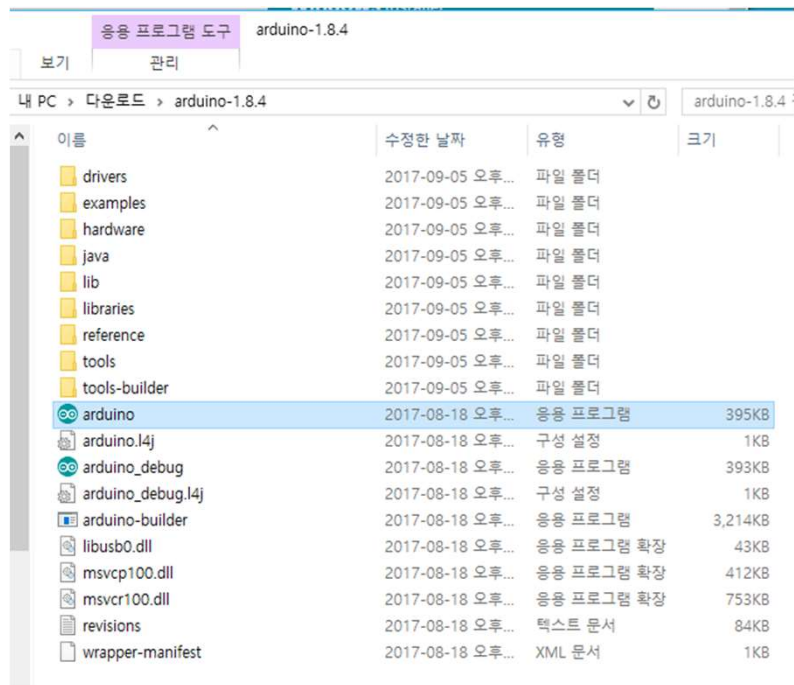
- Download the Arduino IDE에서 자신의 OS 환경에 맞는 버전 클릭





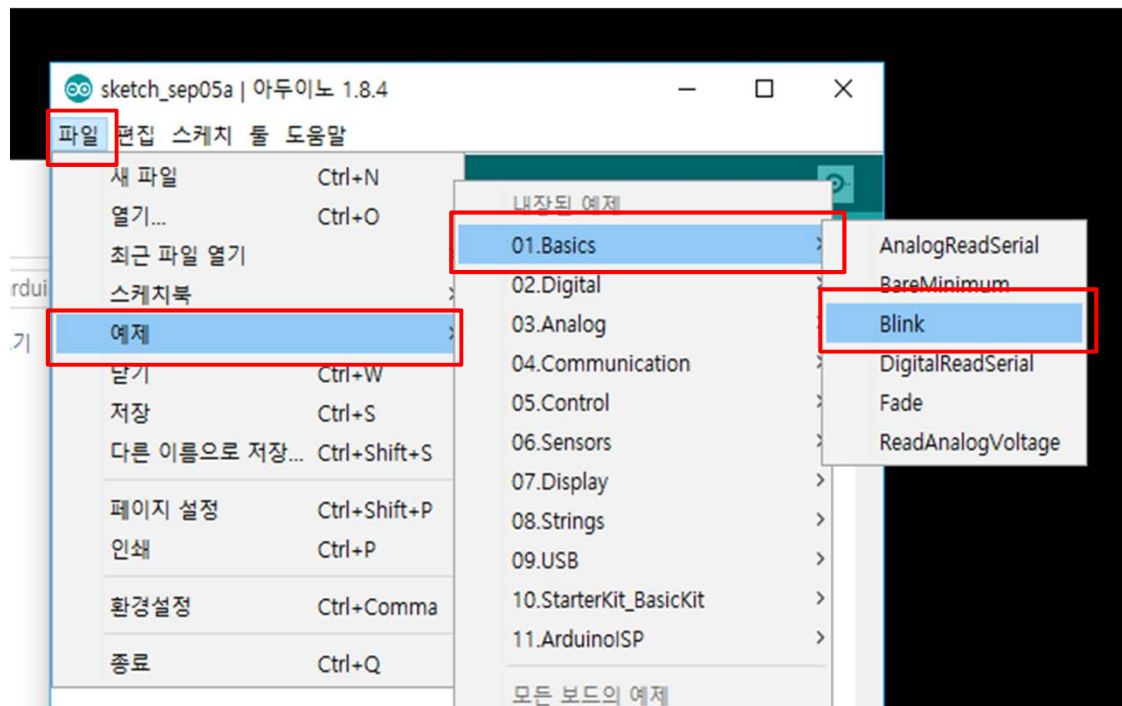
# Arduino IDE 설치

## • 다운로드 한 파일을 실행

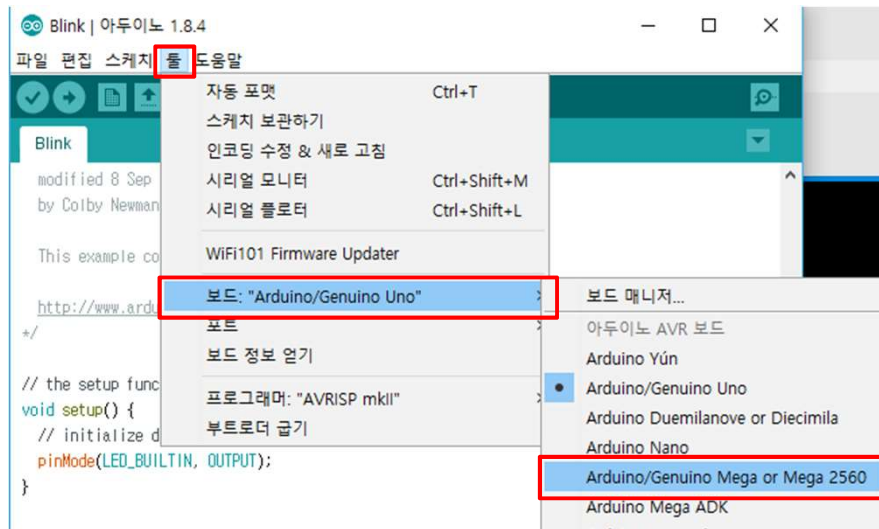


# Blink 예제 적용해보기

- 파일 -> 예제 -> 01. Basics  
-> Blink 선택

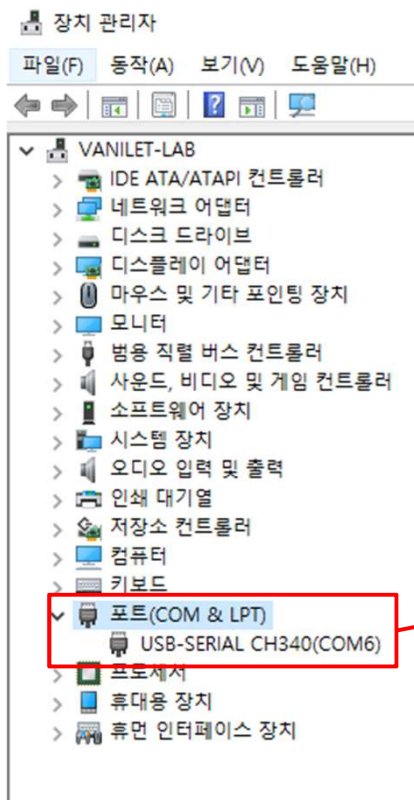


# Blink 예제 적용해보기



- 툴 -> 보드: “~~” -> Arduino/Genuino Mega or Mega 2560 선택

# Blink 예제 적용해보기



## ● 아두이노가 연결된 직렬 포트 확인

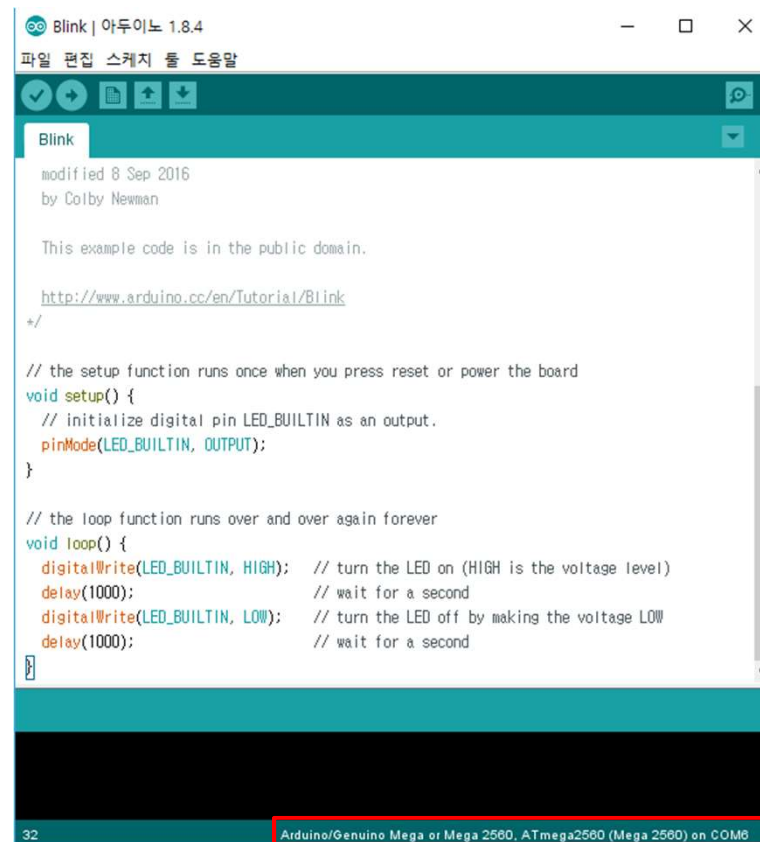
- 예제에서는 COM6

# Blink 예제 적용해보기

- 툴 -> 포트 에서 장치관리자에서 확인한 COM 포트 선택



# Blink 예제 적용해보기



The screenshot shows the Arduino IDE interface with the 'Blink' example code loaded. The window title is 'Blink | 아두이노 1.8.4'. The menu bar includes '파일', '편집', '스케치', '툴', and '도움말'. The toolbar has icons for opening, saving, and running. The code editor shows the following text:

```
Blink
modified 8 Sep 2016
by Colby Newman

This example code is in the public domain.

http://www.arduino.cc/en/Tutorial/Blink
*/

// 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
}
```

At the bottom of the IDE, the status bar shows '32' and 'Arduino/Genuino Mega or Mega 2560, ATmega2560 (Mega 2560) on COM8', which is highlighted with a red box.

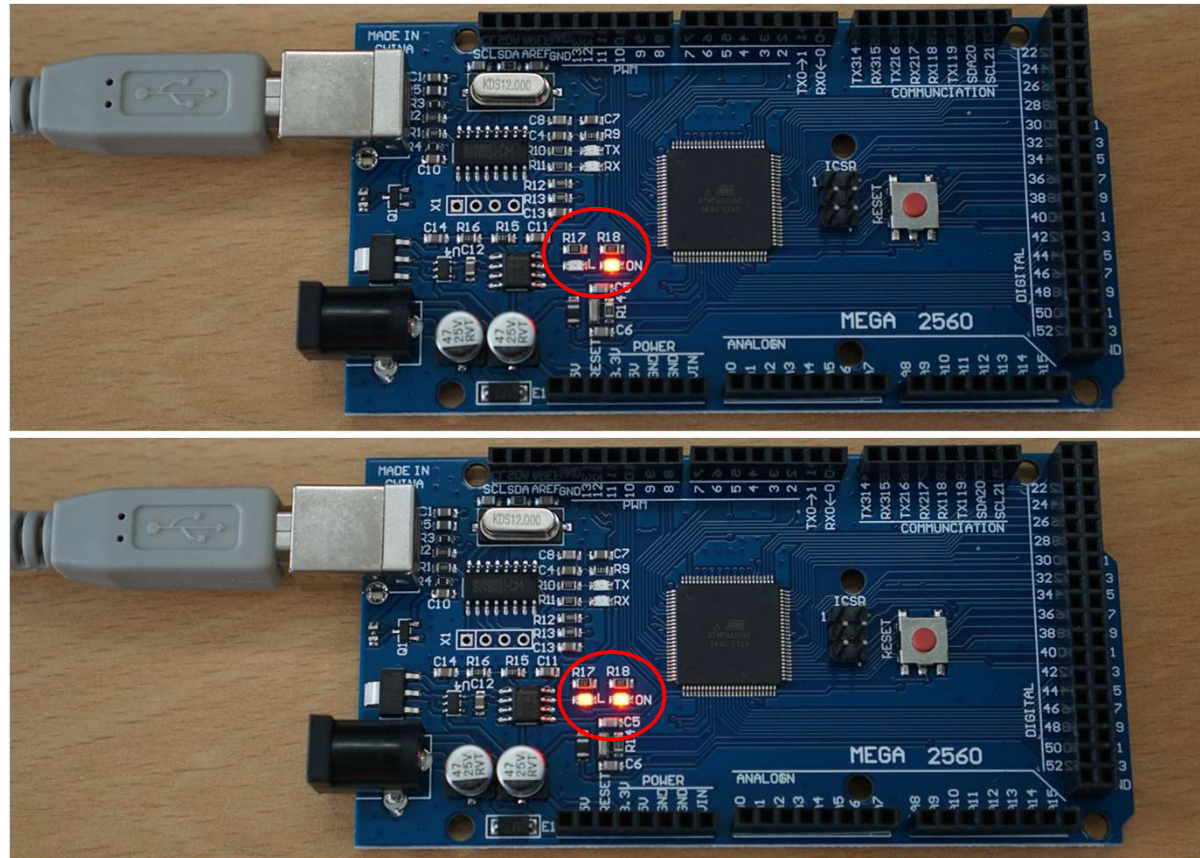
# Blink 예제 적용해보기

- 업로드 버튼 클릭

- 코딩한 프로그램이 아두이노에 설치되



## Blink 예제 적용해보기





# Blink 예제 적용해보기

## Blink\$

// 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  
}
```

- void setup() {
  - 아두이노의 전원이 켜졌을 때 가장 먼저 수행되는 함수
  - 전원이 켜질 때 한 번만 수행됨

- void loop() {
  - 아두이노의 전원이 꺼질 때 까지 계속 수행되는 함수
  - setup() 함수 이후 실행 됨
  - while(1) loop();

# Blink 예제 적용해보기

```
Blink$
// the setup function runs once when you press reset or
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 (LOW is the voltage level)
    delay(1000); // wait for a second
  }
}
```

- `delay(ms)`
  - `ms` (Millisecond) 동안 아두이노를 일시정지 시킴

- `pinMode(pin, mode)`
  - 해당 `pin`의 `mode`를 설정
  - `pin`: 아두이노에 있는 핀 번호 (0 ~ 13, A0 ~ A15, ...)
  - `mode`
    - INPUT: 해당 pin은 입력
    - OUTPUT: 해당 pin은 출력
- `digitalWrite(pin, value)`
  - 해당 `pin`에 `value`를 전달
  - `pin`: 아두이노에 있는 핀 번호
  - `value`
    - HIGH: 1 (True) 값
    - LOW: 0 (False) 값

# Blink 예제 적용해보기

- 소프트웨어에 관한 내용은 <https://www.arduino.cc/en/Reference/HomePage>를 참고하면 됨

Reference | Language | Libraries | Comparison | Changes

## Language Reference

Arduino programs can be divided in three main parts: *structure*, *values* (variables and constants), and *functions*.

### Structure

- `setup()`
- `loop()`

#### Control Structures

- `if`
- `if...else`
- `for`
- `switch case`
- `while`
- `do... while`
- `break`
- `continue`
- `return`
- `goto`

Further Syntax

### Variables

#### Constants

- `HIGH` | `LOW`
- `INPUT` | `OUTPUT` | `INPUT_PULLUP`
- `LED_BUILTIN`
- `true` | `false`
- integer constants
- floating point constants

#### Data Types

- `void`
- `boolean`
- `char`
- `unsigned char`
- `byte`
- `int`

### Functions

#### Digital I/O

- `pinMode()`
- `digitalWrite()`
- `digitalRead()`

#### Analog I/O

- `analogReference()`
- `analogRead()`
- `analogWrite()` - *PWM*

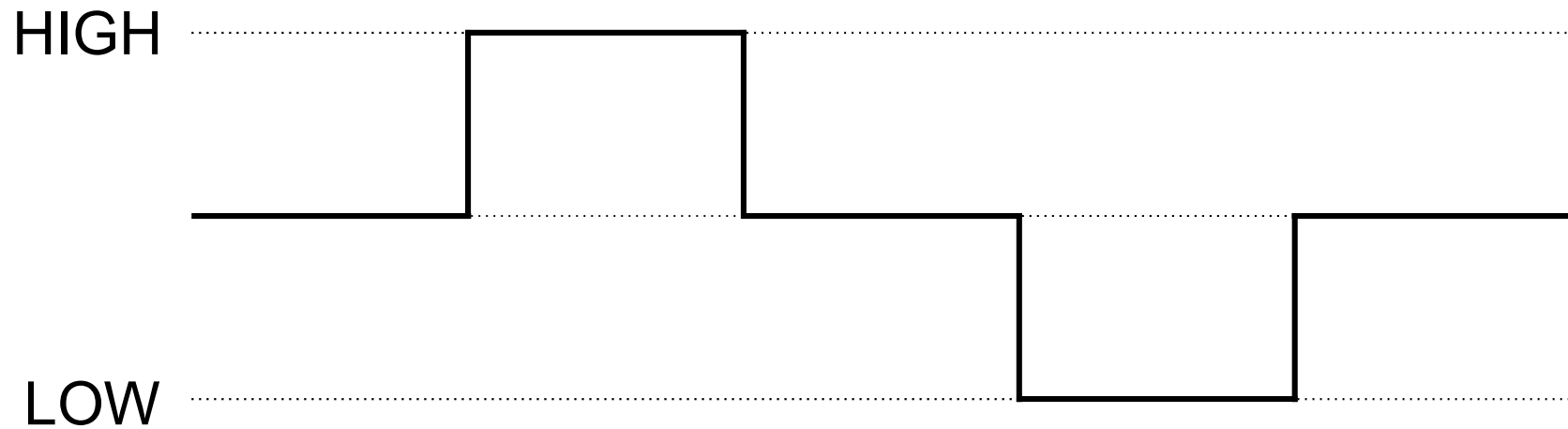
#### Due & Zero only

- `analogReadResolution()`
- `analogWriteResolution()`

#### Advanced I/O

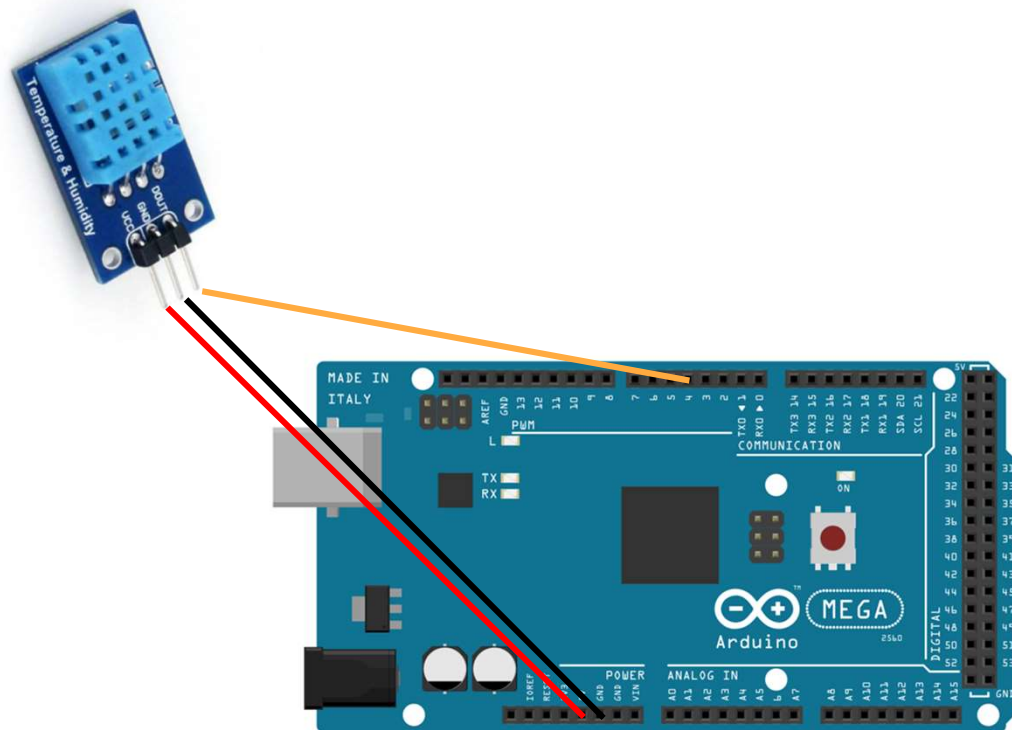
## Blink 예제 적용해보기

- 디지털 신호 입/출력



# 온습도 센서 사용해보기

- 회로 구성

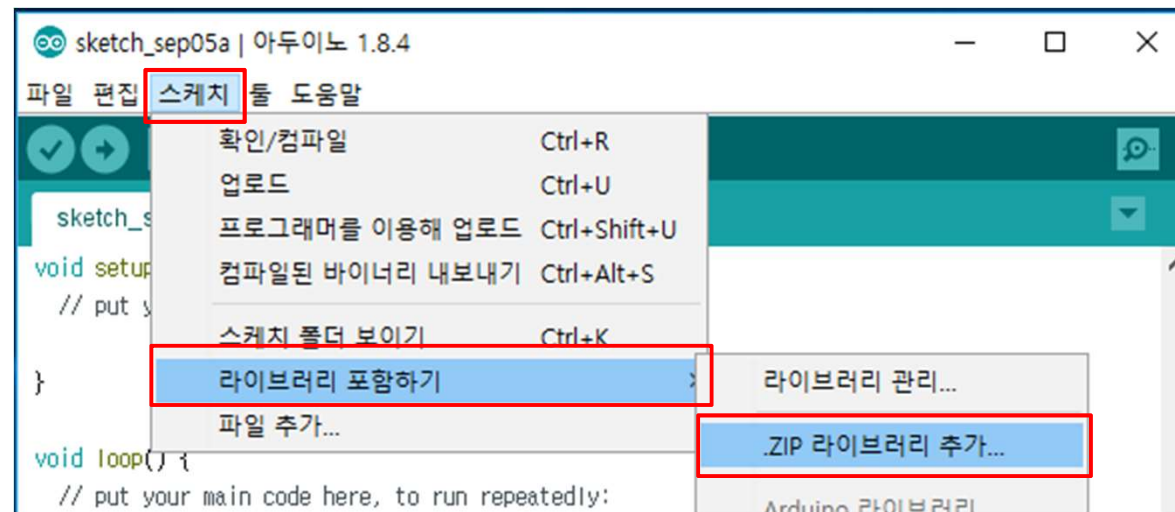


# 온습도 센서 사용해보기

- 온습도 센서 라이브러리 다운
  - [다운로드](#) <- 클릭

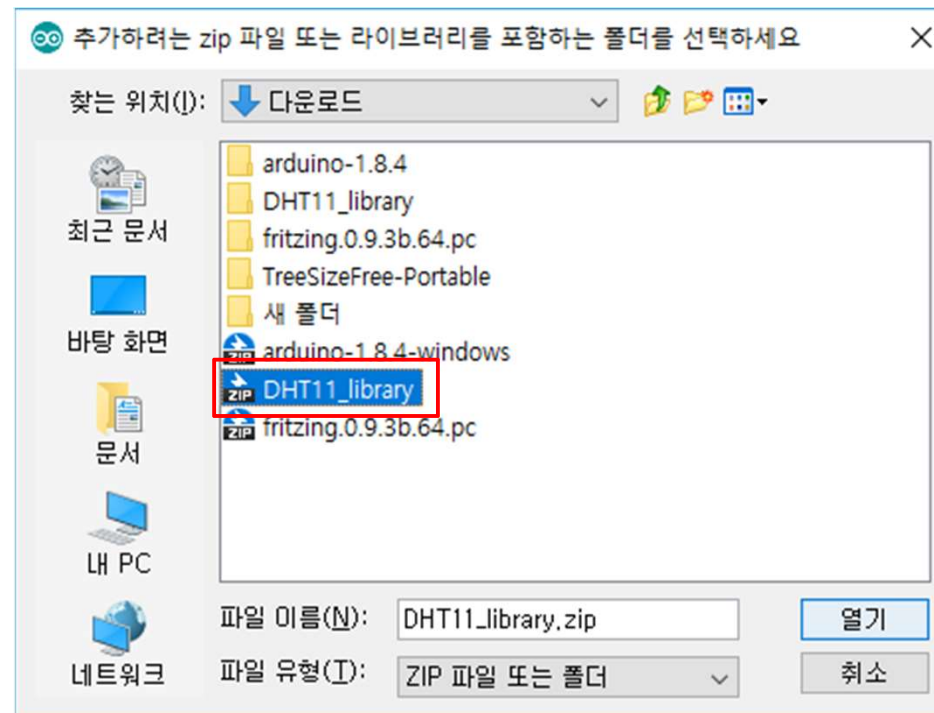
# 온습도 센서 사용해보기

- 스케치 -> 라이브러리 포함하기 -> .ZIP 라이브러리 추가 클릭



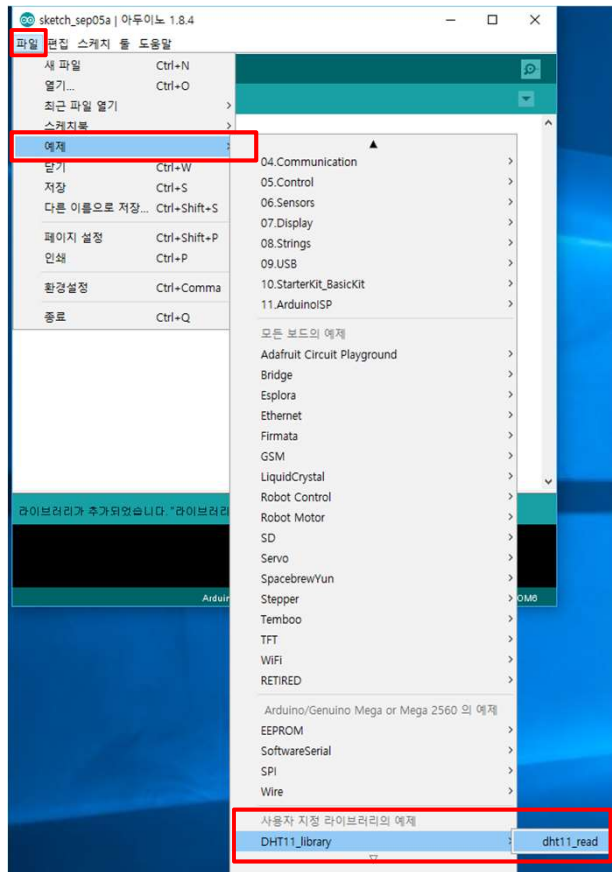
# 온습도 센서 사용해보기

- 다운로드 한 라이브러리 파일 선택





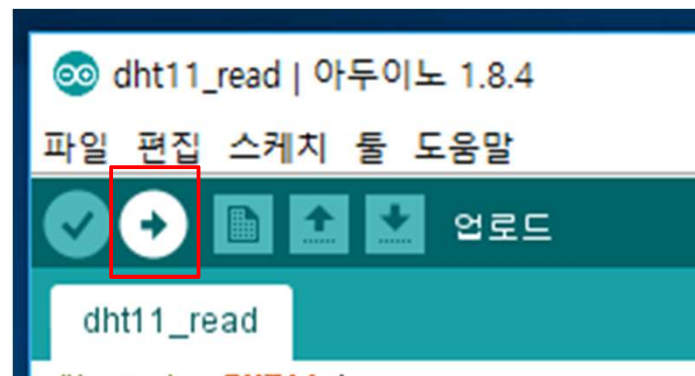
# 온습도 센서 사용해보기



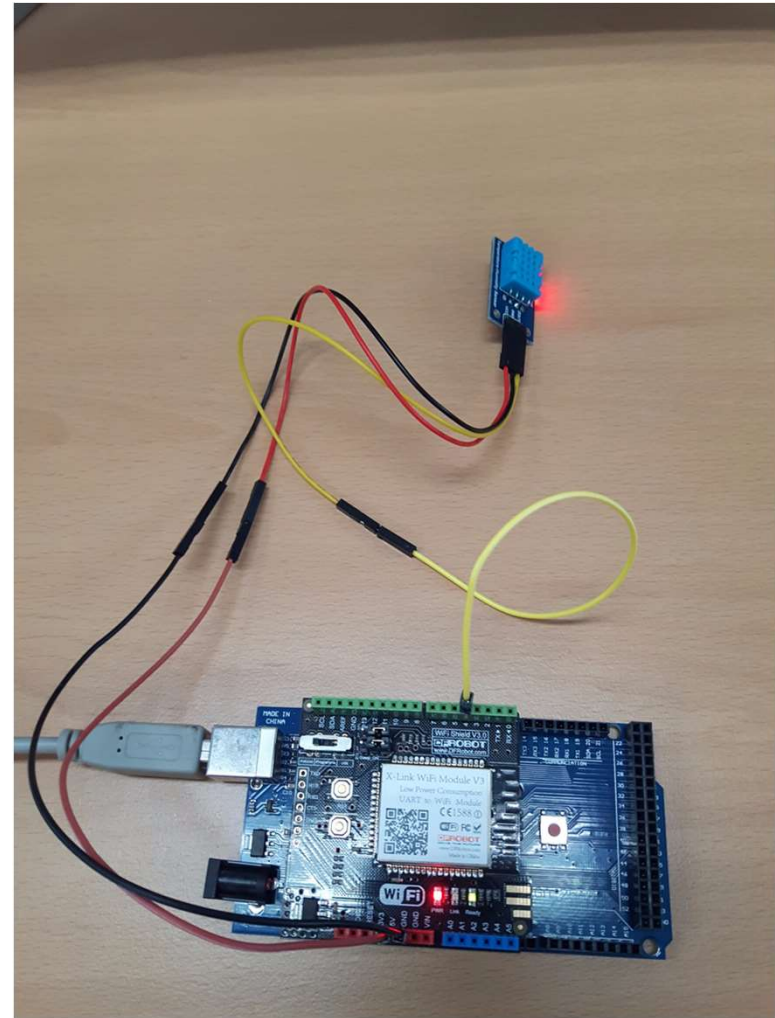
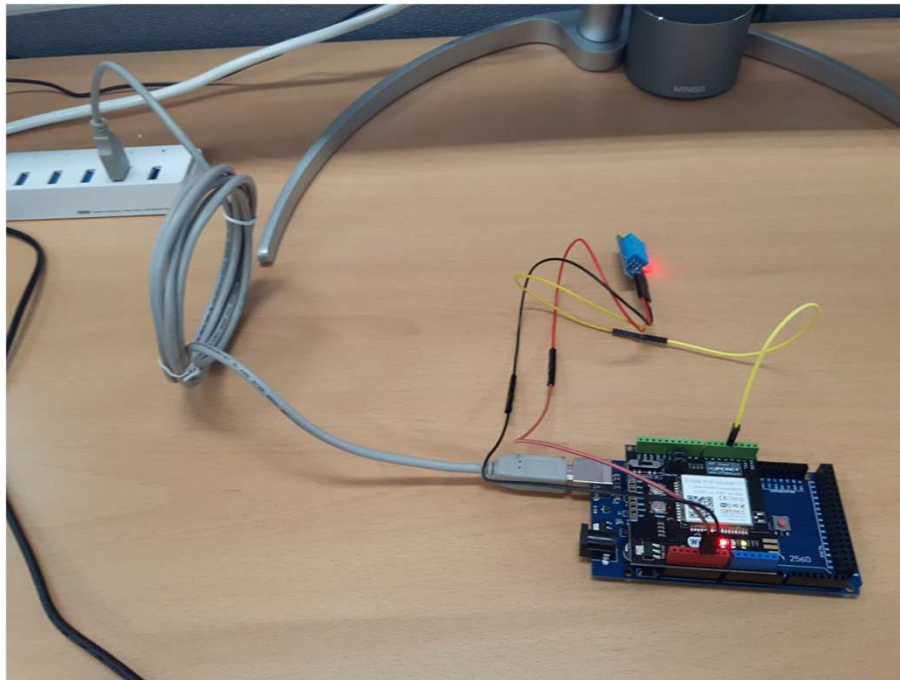
- 파일 -> 예제 -> DHT11\_library -> dht11\_read 선택

## 온습도 센서 사용해보기

- 업로드 버튼 클릭



## 온습도 센서 사용하기

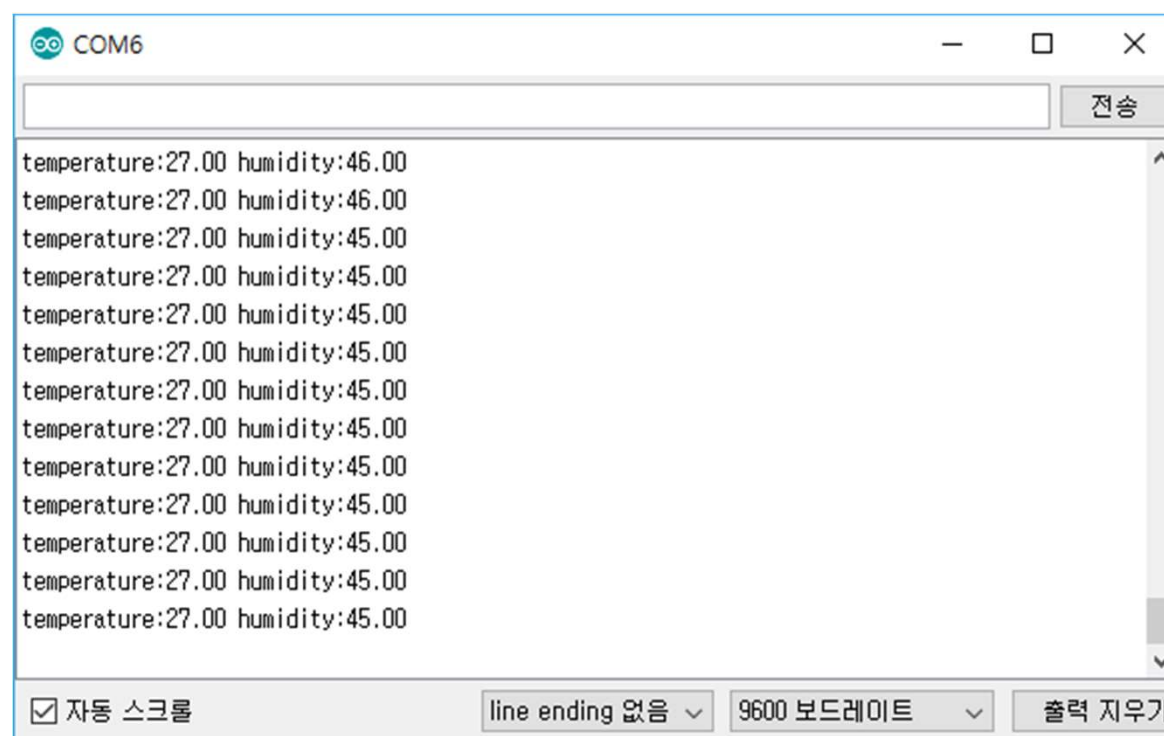


# 온습도 센서 사용해보기

- 옵션 1: 툴 -> 시리얼 모니터 선택하기
- 옵션 2: 돋보기 아이콘 클릭



## 온습도 센서 사용해보기



# 온습도 센서 사용해보기

```
dht11_read $
#include <DHT11.h>

int pin=4;
DHT11 dht11(pin);

void setup()
{
  Serial.begin(9600);
  while (!Serial) {
    ; // wait for serial port to connect. Needed for Leonardo only
  }
}
```

- Serial.begin(*speed*)
  - 컴퓨터와 아두이노 간 직렬 통신 (Serial 통신) 생성
  - *speed*: bits per second (baud)

# 온습도 센서 사용해보기

```
dht11_read $  
  
void loop()  
{  
  int err;  
  float temp, humi;  
  if((err=dht11.read(humi, temp))==0)  
  {  
    Serial.print("temperature:");  
    Serial.print(temp);  
    Serial.print(" humidity:");  
    Serial.print(humi);  
    Serial.println();  
  }  
  else  
  {  
    Serial.println();  
    Serial.print("Error No :");  
    Serial.print(err);  
    Serial.println();  
  }  
  delay(DHT11_RETRY_DELAY); //delay for reread  
}
```

- Serial.print(*val*, *format*)
  - 생성한 Serial 통신 포트에 값을 출력
  - 시리얼 모니터에서 확인 가능
  - *val*: 출력할 값 (문자열)
  - *format*: 진수 설정 (*val*이 정수일 경우), 소숫점 위 자리수 설정 (*val*이 실수인 경우)
    - DEC (10진수), BIN (2진수), HEX (16진수), OCT (8진수) 로 출력
- Serial.println(*val*, *format*)
  - Serial.print()의 기능 + new line 삽입