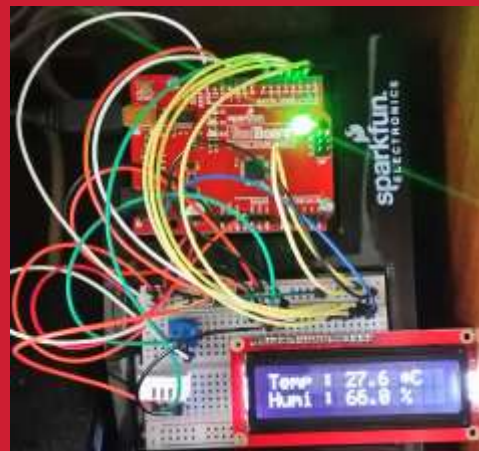




# Arduino-IOT

[wk04]

## Arduino circuits



Visualization of Signals using Arduino,  
Node.js & storing signals in MongoDB

Comsi, INJE University

2<sup>nd</sup> semester, 2018

Email : [chaos21c@gmail.com](mailto:chaos21c@gmail.com)





# My ID

진영빈	AA01
김태은	AA02
도한솔	AA03
박지수	AA04
신성	AA05
박현승	AA06
이석주	AA07
전규은	AA08
정영관	AA09
정의석	AA10

이근재

**AA11**



# [Review]

## ◆ [wk03]

- Express App (7.2 Express project)
- Add a new route in **routes/index.js**
- Add a new view: **views/aann.jade**
- Upload file name : **AAnn\_Rpt02.zip**



# [Practice-1] Modify routes/index.js

**Add a new route `/aann` in routes/index.js .**

## FOLDERS

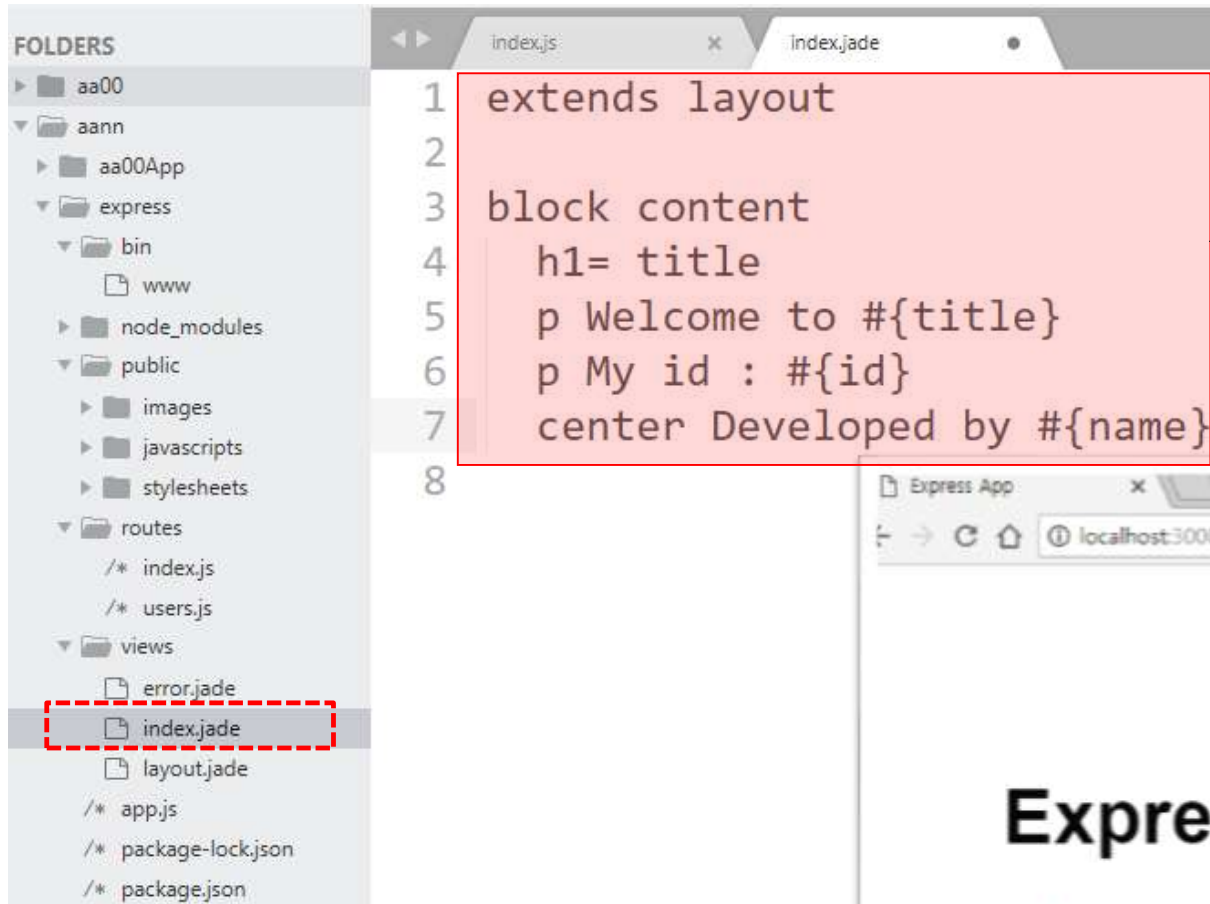
- ▶ aa00
  - ▼ aann
    - ▶ aa00App
    - ▼ express
      - ▼ bin
        - www
      - node\_modules
      - ▼ public
        - images
        - javascripts
        - stylesheets
      - ▼ routes
        - /\* index.js**
        - /\* users.js
      - ▼ views
        - error.jade
        - index.jade
        - layout.jade
      - /\* app.js
      - /\* package-lock.json
      - /\* package.json
    - ▶ expressTest
    - ▼ server
      - File

```
index.js x
1 var express = require('express');
2 var router = express.Router();
3
4 /* GET home page. */
5 router.get('/', function(req, res, next) {
6   res.render('index', { title: 'Express' });
7 });
8
9 /* GET my page by /aann. -> multi-routing*/
10 router.get('/aann', function(req, res, next) {
11   res.render('aann', { title: 'Express App',
12                      id: 'AA00',
13                      name: 'Redwoods' });
14   // views/aann.jade
15 });
16
17 module.exports = router;
```



# [Practice-2] Add aann.jade in views folder

## Add views/aann.jade in views folder



```
1 extends layout
2
3 block content
4   h1= title
5   p Welcome to #{title}
6   p My id : #{id}
7   center Developed by #{name}
8
```

Save as  
aann.jade

Saves as  
**AAnn\_Jade.png**

## Express App

Welcome to Express App

My id : AA00

Developed by Redwoods

## ◆ [Target of this week]

- Complete your works
- Save your outcomes and compress 4 figures

제출파일명 : **AAnn\_Rpt02.zip**

- 압축할 파일들

- ① **AAnn\_Express.png**
- ② **AAnn\_naver.png**
- ③ **AAnn\_Express\_App.png**
- ④ **AAnn\_Jade.png**

**Email : [chaos21c@gmail.com](mailto:chaos21c@gmail.com)**

**[ 제목 : id, 이름 (수정) ]**



# 1.0 What is node.js?





## 2.2 Install node.js portable



<https://gareth.flowers/nodejs-portable/>





# Node.js Server

**1. http, tcp, file**

**2. Express**



## 7. Node Server

# Node Server II.

- 1. Express server**
- 2. Full Express App**
- 3. My Express App**



## 7.1.1 Express server test

**Step 1 : npm init**

**Step 2 : npm install --save express**

**Step 3 : Write Express code**

**Step 4 : Run app.js**

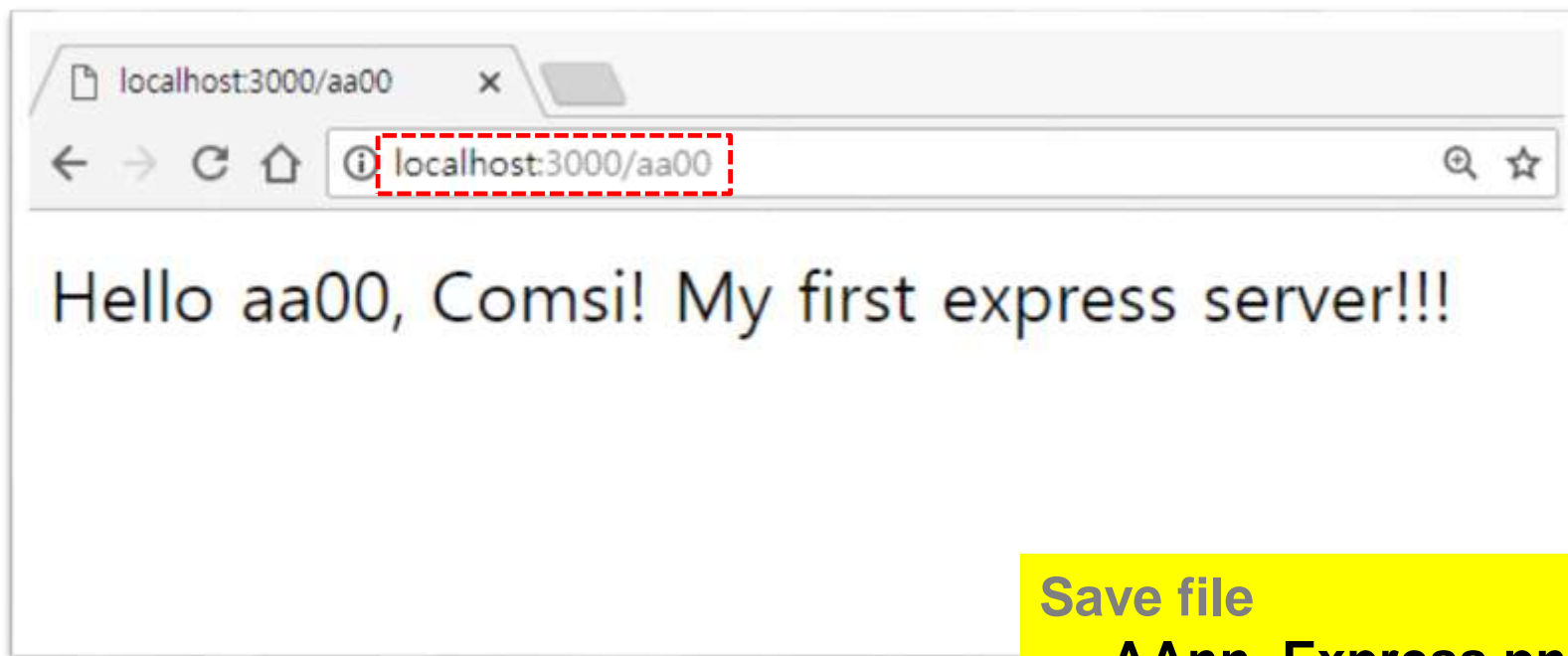
**Step 5 : http://localhost:3000**



## 7.1.9 Express server test: run app.js – DIY

### [DIY] My ID routing → localhost:3000/aann

```
app.get('/aa00', function(req, res) {  
  res.send('Hello aa00, Comsi! My first express server!!!');  
});
```



Save file  
AAnn\_Express.png

**Routing:** 라우팅은 애플리케이션 엔드 포인트(URI)의 정의, 그리고 URI가 클라이언트 요청에 응답하는 방식



# Full Express App using generator



## 7.2.1 Full Express App using generator

**Step 1 : make folder 'express'**

**Step 2 : go to the folder**

**Step 3 : `npm install -g express-generator`**

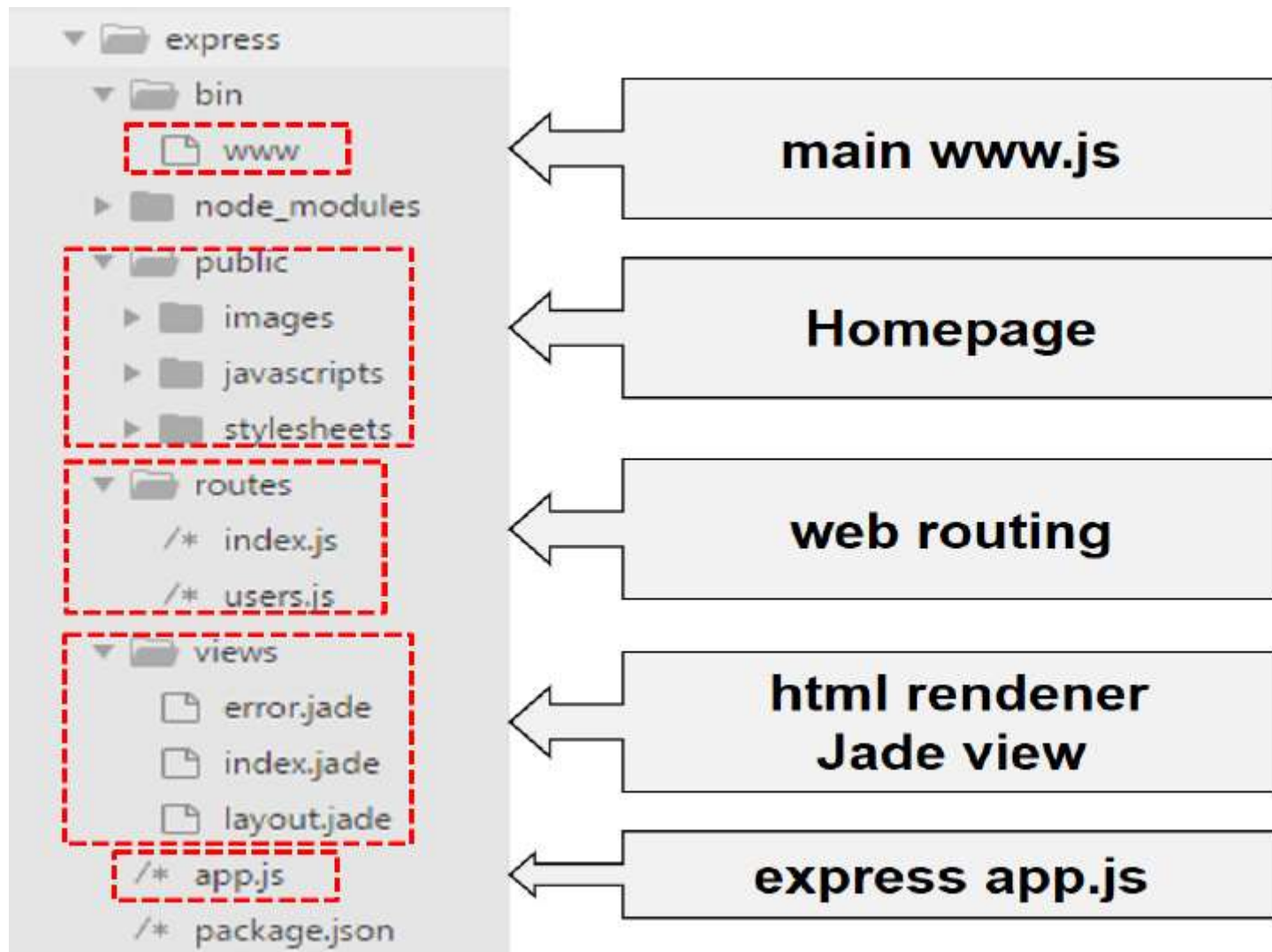
**Step 4 : `express`**

**Step 5 : `npm install`**

**Step 6 : `node ./bin/www`**

**Step 6 : <http://localhost:3000>**

## 7.2.6 Inner structure of express server





# My Express App using generator





# Express options

- e, --ejs** add ejs engine support (defaults to **jade**)
- hbs** add handlebars engine support
- H, --hogan** add hogan.js engine support
- c, --css <engine>** add stylesheet <engine> support (less|stylus|compass)
- f, --force** force on non-empty directory<engine></engine>

## 7.3.2 Templating using Express-generator

**express aa00App -e -c less**



The screenshot shows a code editor with a file explorer on the left and a code editor on the right. The file explorer shows a project structure with folders like 'aa00', 'aann', 'aa00App', 'bin', 'public', 'images', 'javascripts', 'stylesheets', 'routes', 'views', 'express', 'expressTest', 'server', 'File', 'HTTP', 'TCP', and 'start'. The 'aa00App' folder is expanded, showing subfolders 'bin' and 'public', and files 'index.js', 'users.js', 'error.ejs', 'index.ejs', 'app.js', and 'package.json'. The 'package.json' file is selected and its content is displayed in the code editor. The content of 'package.json' is as follows:

```

1 {
2   "name": "aa00app",
3   "version": "0.0.0",
4   "private": true,
5   "scripts": {
6     "start": "node ./bin/www"
7   },
8   "dependencies": {
9     "cookie-parser": "~1.4.3",
10    "debug": "~2.6.9",
11    "ejs": "~2.5.7",
12    "express": "~4.16.0",
13    "http-errors": "~1.6.2",
14    "less-middleware": "~2.2.1",
15    "morgan": "~1.9.0"
16  }
17 }
18

```

The dependencies section of the 'package.json' file is highlighted with a red dashed box, showing the following dependencies:

- "cookie-parser": "~1.4.3"
- "debug": "~2.6.9"
- "ejs": "~2.5.7"
- "express": "~4.16.0"
- "http-errors": "~1.6.2"
- "less-middleware": "~2.2.1"
- "morgan": "~1.9.0"



# Arduino



Home

Buy

Download

Products ▼

Learning ▼

Forum

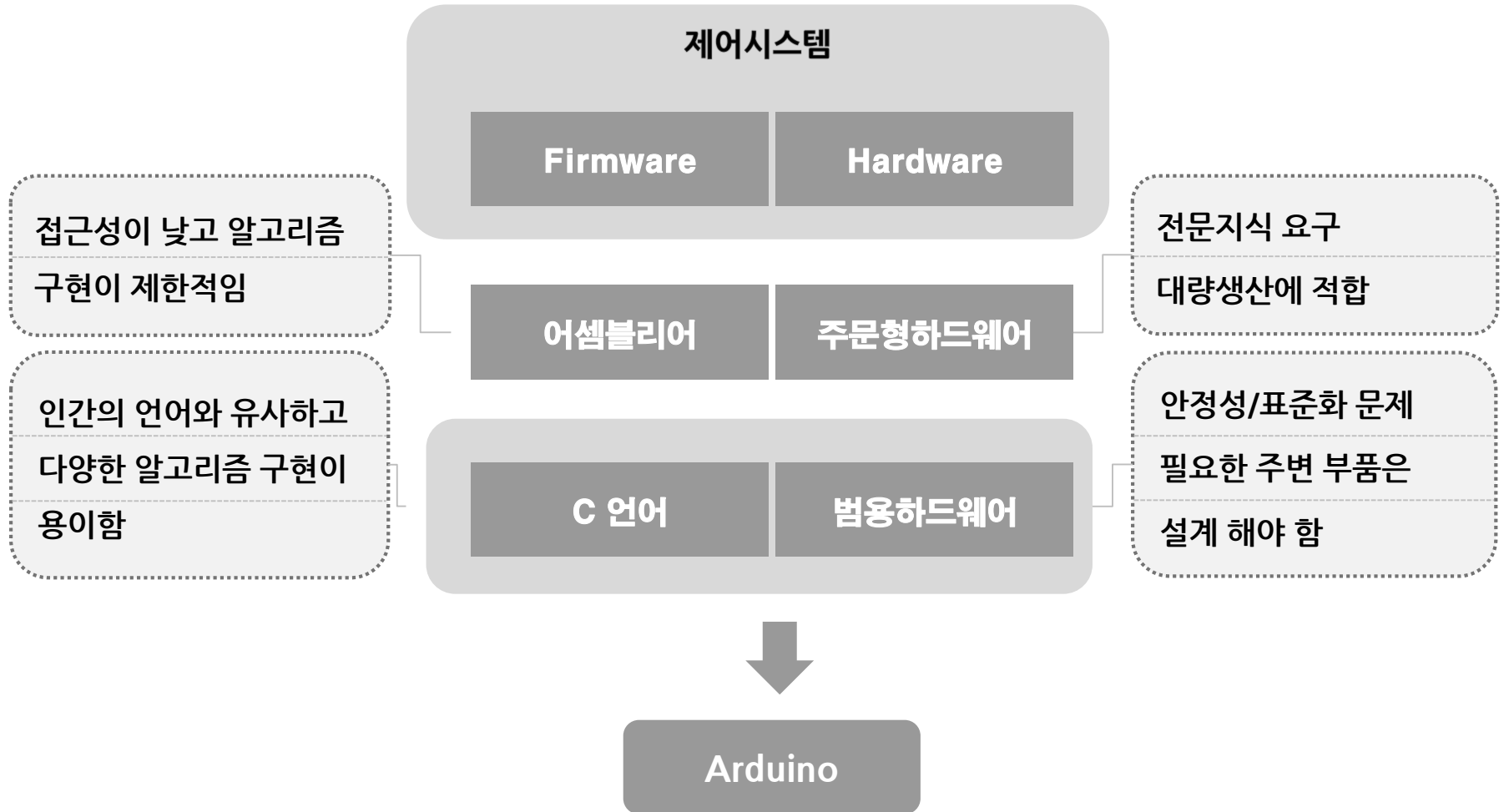
Support ▼

Blog

<https://www.arduino.cc/>



# 0.1 Arduino 란?



# 0.1 Arduino 란?

2005년 Italy의 Massimo Banzi & David Cuatielles에 의해 개발

예술가  
취미생활  
학생

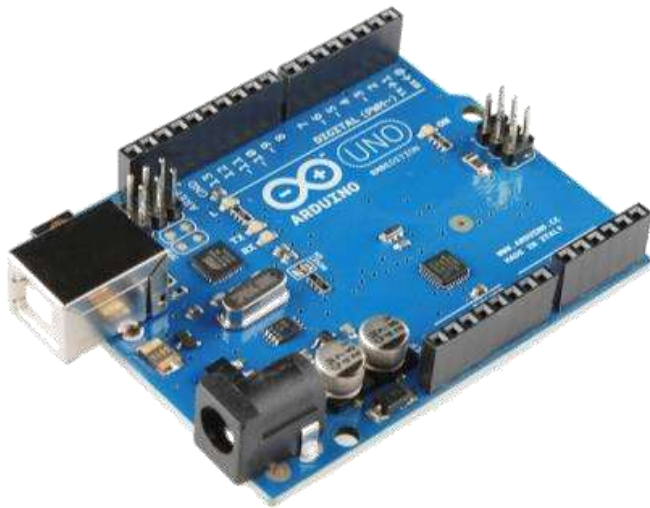
전자공학  
교육

누구나 쉽게  
사용 가능한  
제어장치

오픈소스  
하드웨어

GSM Wifi  
Ethernet  
Motor drive  
등의 쉴드 제공

다양한  
라이브러리



LabView

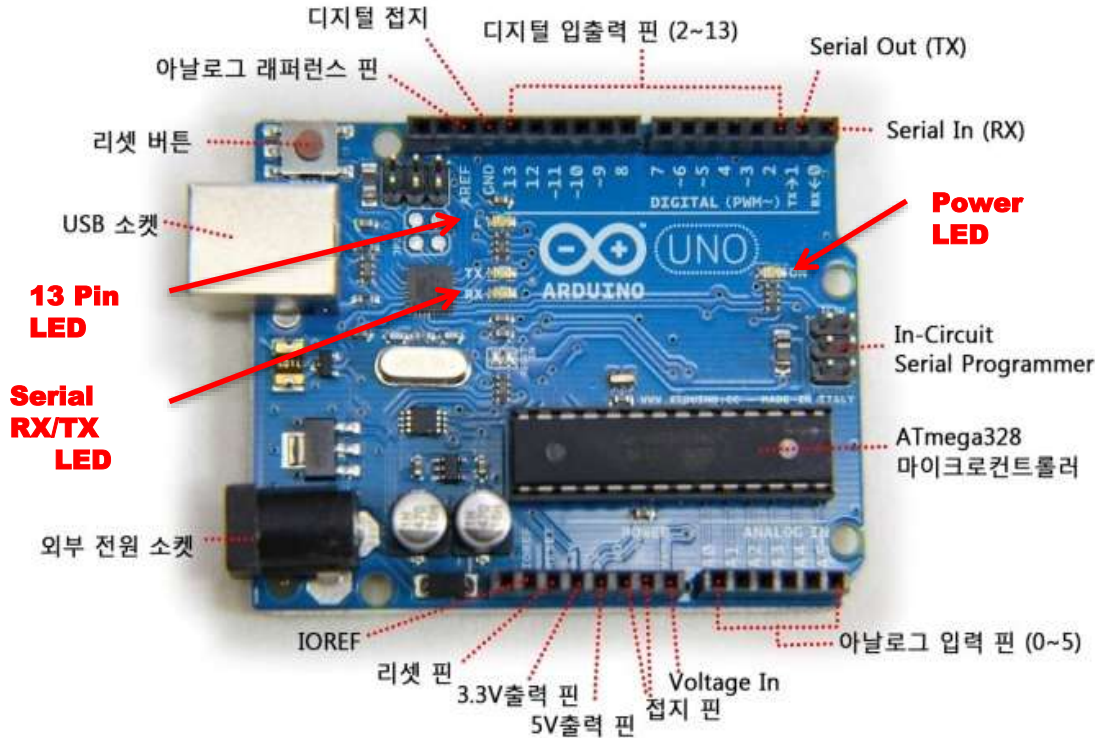
MATLAB

Node.js  
Plot.ly

Mongo DB

범용  
하드웨어  
IOT 의  
표준

# 0.2 Arduino hardware



## ✓ Arduino UNO R3

- ATmega328 microcontroller
- Input voltage: 7~12V
- 14 Digital I/O Pins (6 PWM outputs)
- 6 Analog Inputs
- 32KB Flash Memory
- 16Mhz Clock Speed

## 0.2.1 Arduino hardware



### ✓ Arduino Pro NANO

- ATmega168/328 microcontroller
- Input voltage: 7~12V
- 14 Digital I/O Pins (6 PWM outputs)
- 8 Analog Inputs
- 16KB Flash Memory
- 16Mhz Clock Speed

## 0.2.2 Arduino hardware

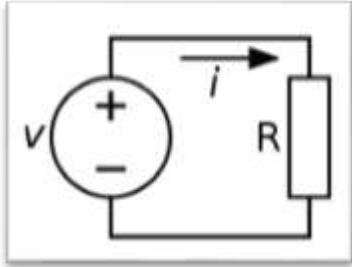


### ✓ Arduino Uno WiFi ESP8266 Wi-Fi Module

- ATmega328p microcontroller
- Input voltage: 7~12V
- 14 Digital I/O Pins (6 PWM outputs)
- 8 Analog Inputs
- ESP8266 Wi-Fi
- 16Mhz Clock Speed



## 0.3 전압, 전류, 저항



전압  
[V]

- ✓ 전위가 높은 쪽과 낮은 쪽의 차이
- ✓ 1쿨롱(coulomb: 전하의 단위)의 전하가 갖고 있는 에너지
- ✓ Arduino에서는 직류 3.3[V]와 5[V]를 지원

전류  
[A]

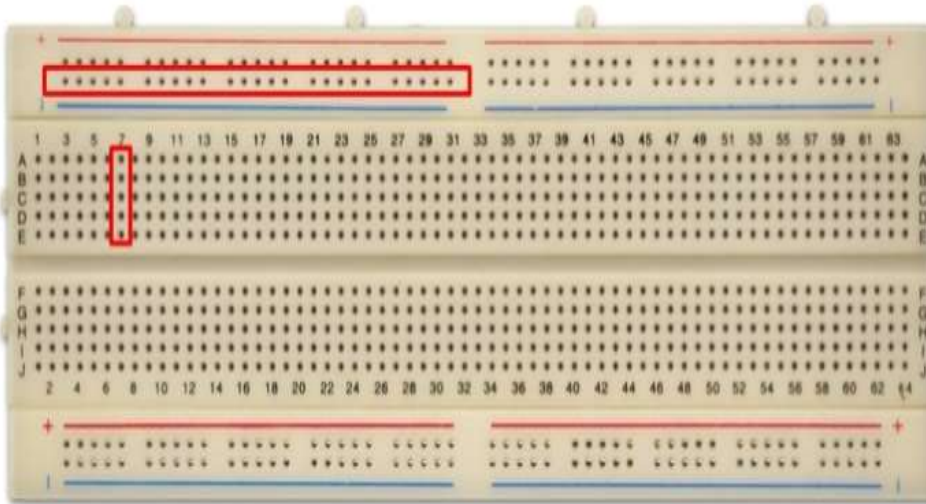
- ✓ 1초당 1쿨롱의 전하가 단위 면적을 통과했을 때를 1[A]로 정의
- ✓ Arduino에서는 1/1000[A] 단위인 [mA]를 사용

저항  
[Ω]

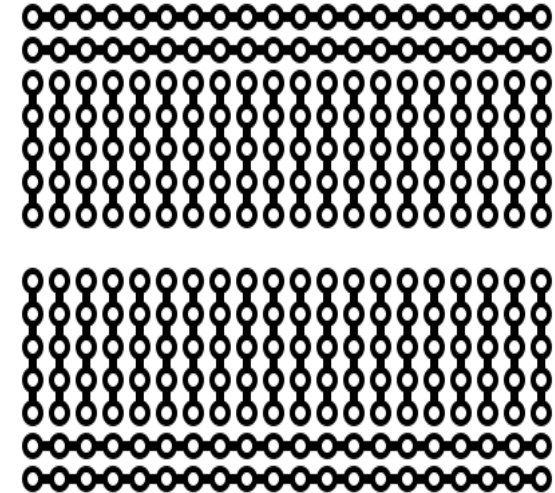
- ✓ 전류의 흐름을 방해하는 정도를 나타냄
- ✓ 색 띠나 숫자로 값을 표시
- ✓ Arduino에서는 칩 (chip) 형태의 저항이 사용

# 0.4 브레드 보드 (Bread board)

시제품 제작이나 실험용 와이어를 보드에 꽂아 사용



빨간색 묶음 홀끼리 내부회로가 연결되어 있음



내부 결선



# Arduino SW

<http://fritzing.org/home/>

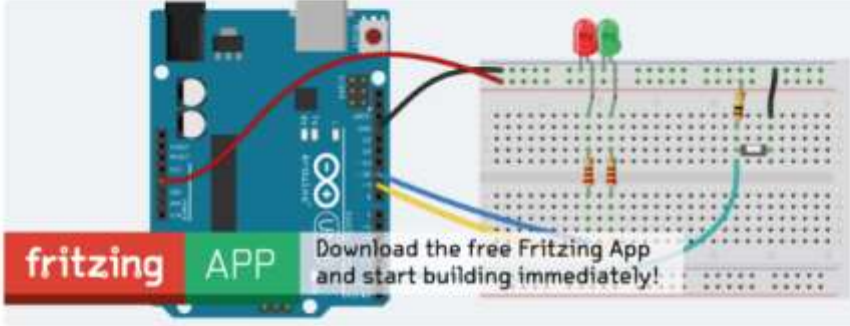
fritzing.org · Fritzing Fritzing

## fritzing

electronics made easy

Projects · Parts · Download · Learning · Services · Contribute

FORUM FAB



**fritzing APP** Download the free Fritzing App and start building immediately!

Fritzing is an open-source hardware initiative that makes electronics accessible as a creative material for anyone. We offer a software tool, a community website and services in the spirit of Processing and Arduino, fostering a creative ecosystem that allows users to document their prototypes, share them with others, teach electronics in a classroom, and layout and manufacture professional pcbs.

### Download and Start

Download our latest version 0.9.3b released on June 2, 2016 and start right away.

### Produce your own board

With Fritzing Fab you can easily and inexpensively turn your circuit into a real, custom-made PCB. Try it out now!

### Participate

Fritzing can only act as a creative platform if many



# Fritzing configuration

**fritzing** electronics  
made easy

Projects Parts **Download** Learning Services Contribute

FORUM

FAB

Fritzing is open source, free software. Be aware that the development of it depends on the active support of the community.

Select the download for your platform below.

Version **0.9.3b** was released on June 2, 2016.

Windows 32 bit

Windows 64 bit

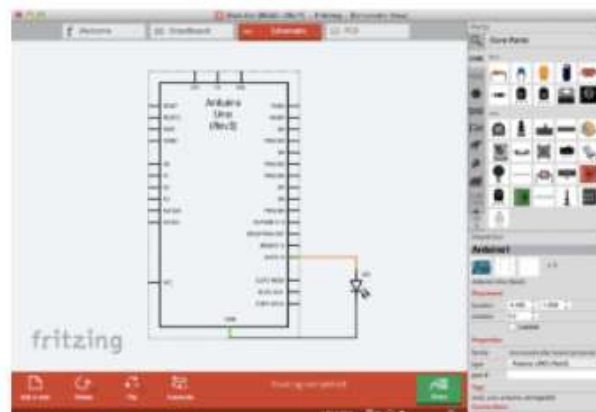
Mac OS X 10.7 and up

Linux 32 bit

Linux 64 bit

Source Github

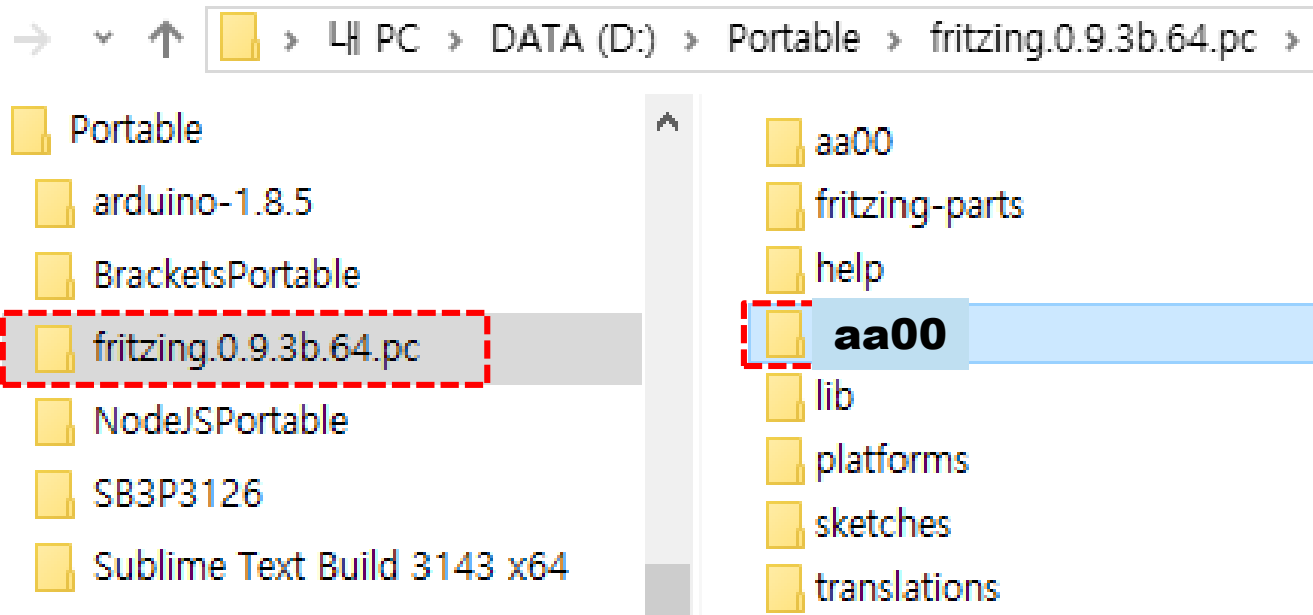
Downloaded 2578877 times.





# Fritzing configuration: working folder

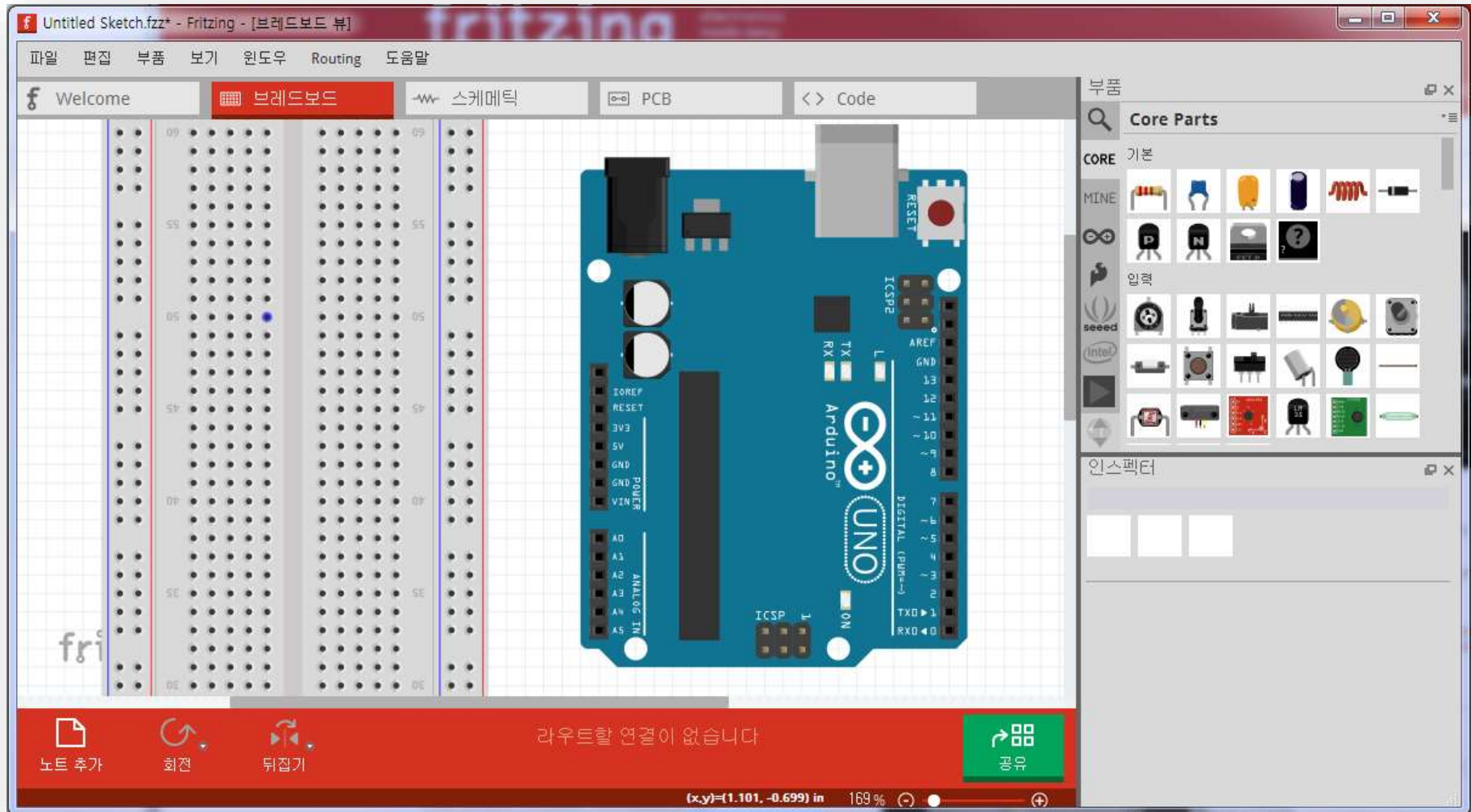
## [Fritzing] configuration





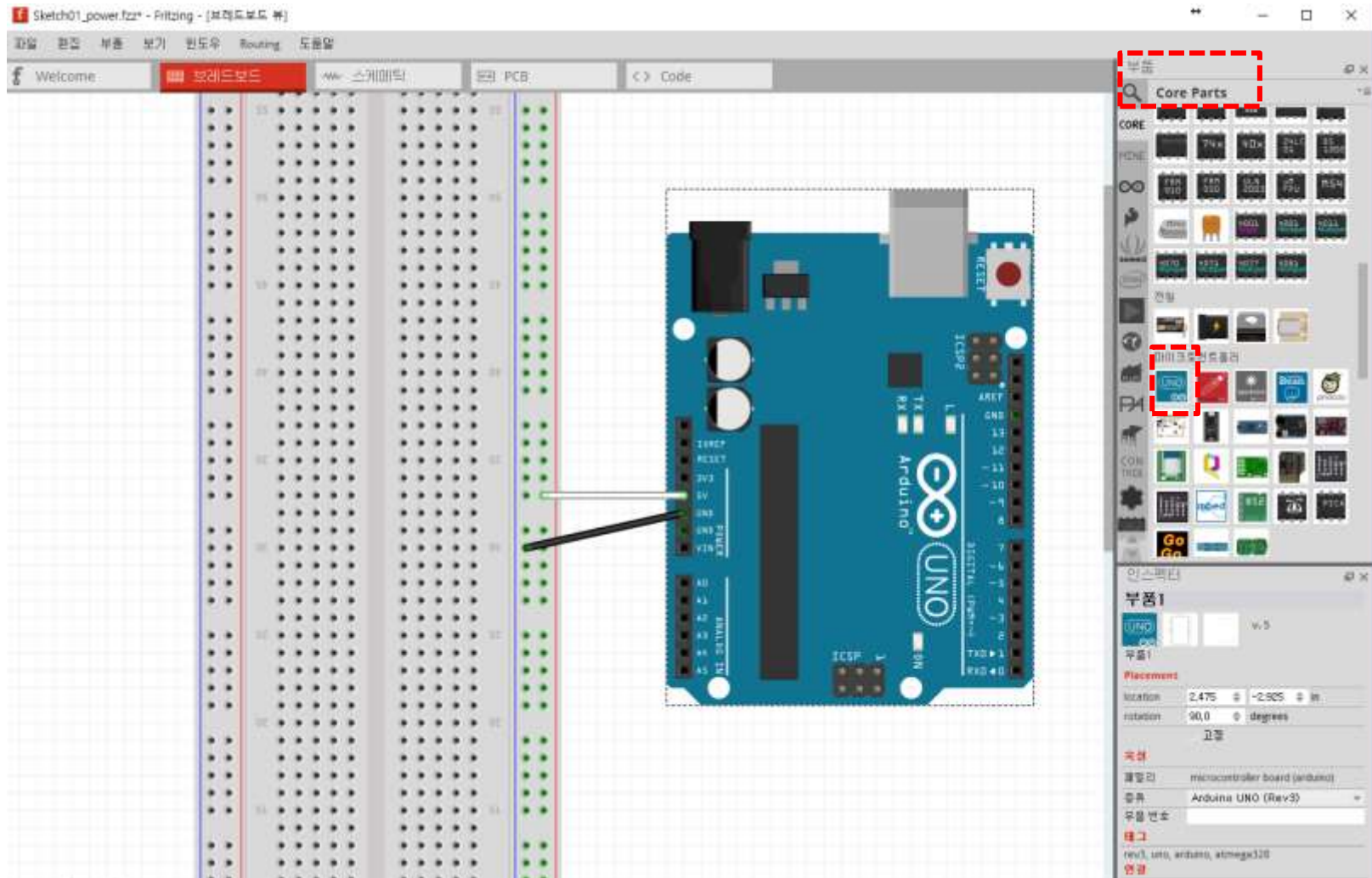
# Fritzing configuration

## [Fritzing] configuration





# Fritzing configuration – power

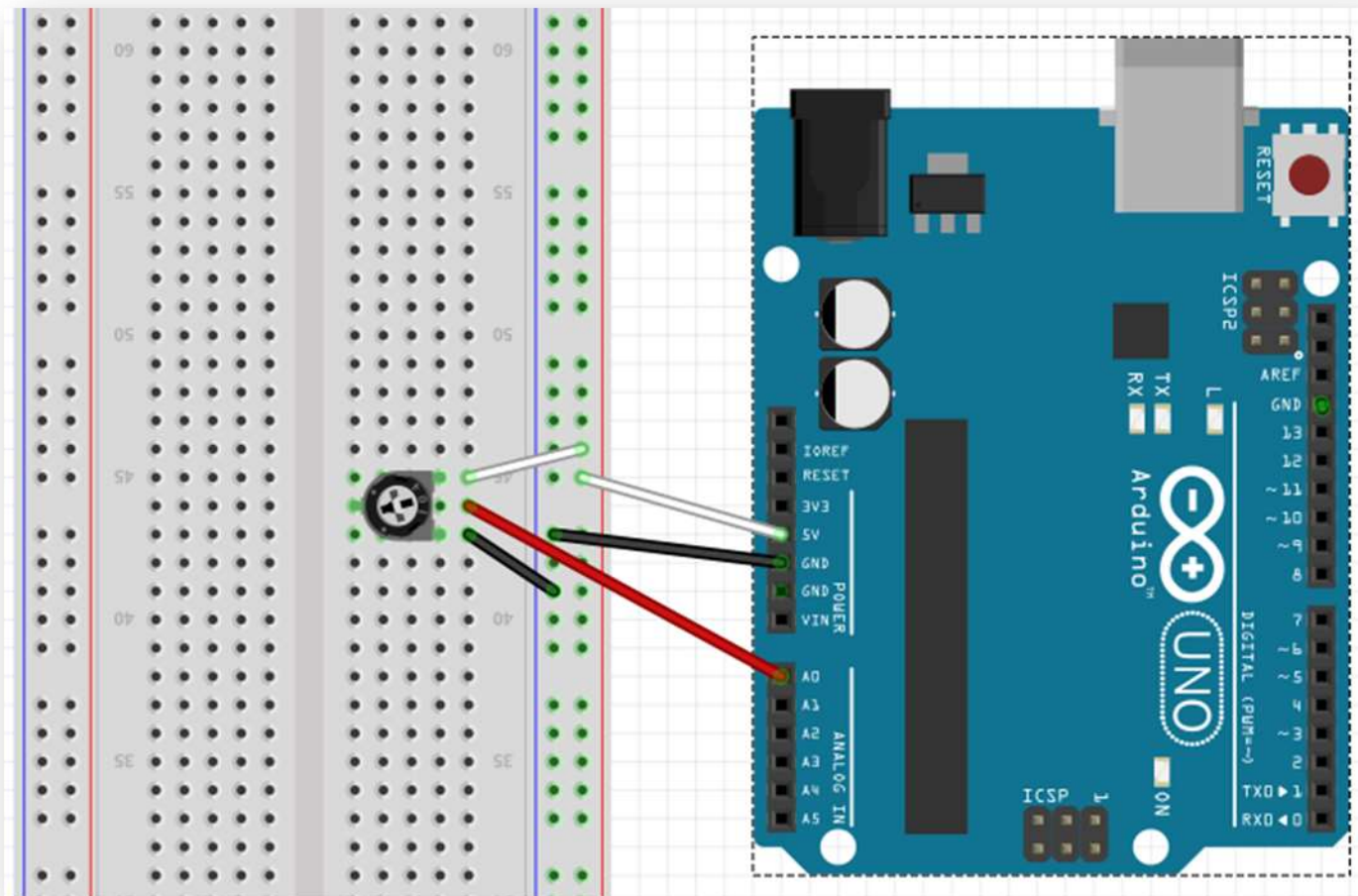




# Arduino circuits

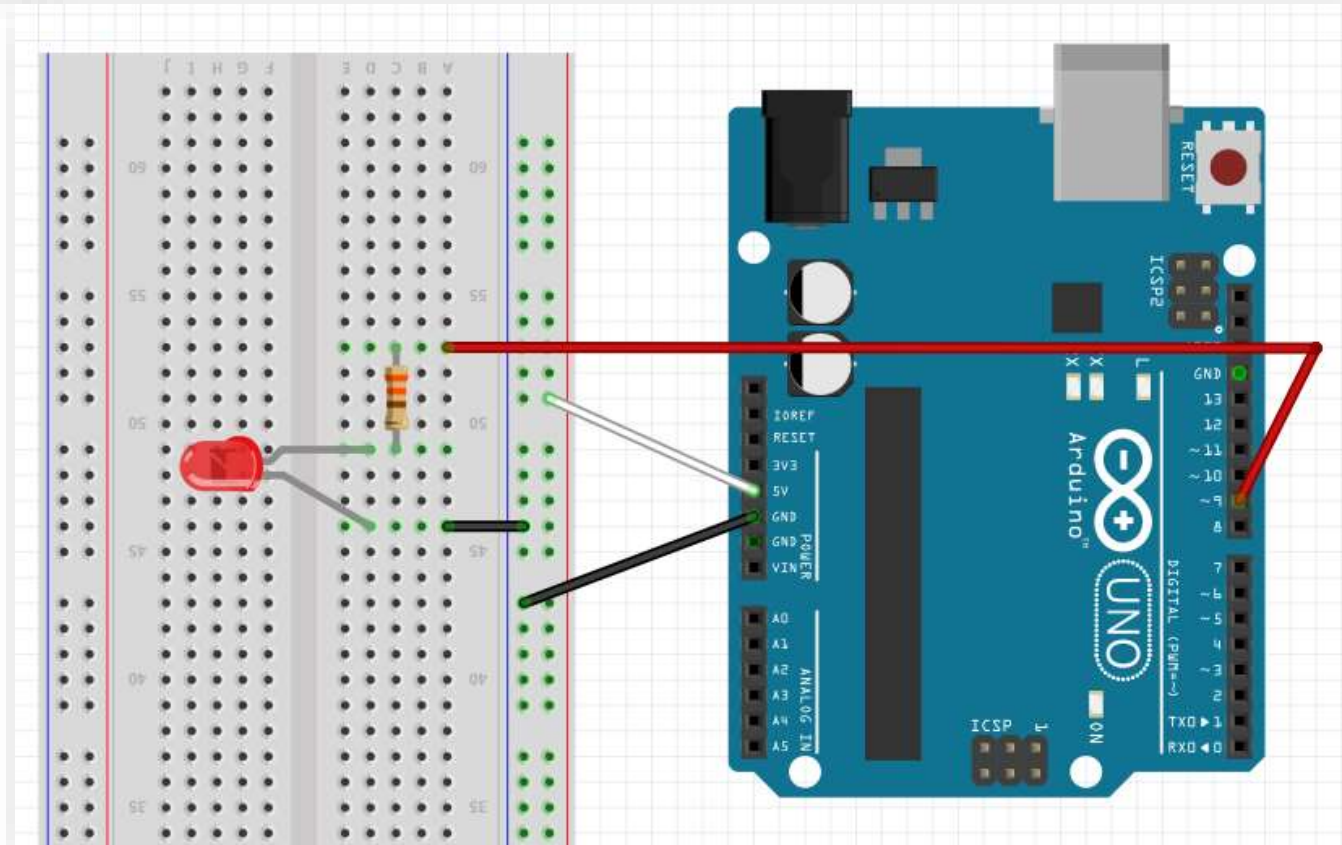


# 0.A1 Potentiometer (가변 저항기)

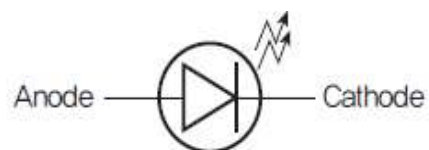


**Parts : 가변저항기**

# 0.A2 single LED



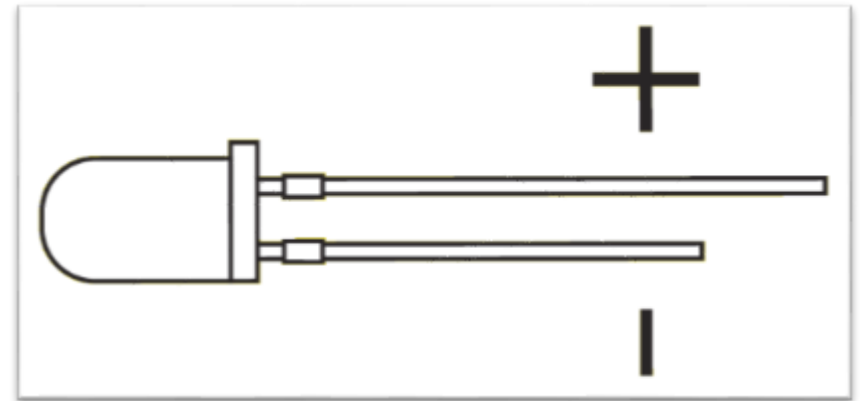
**Parts : LED (1), R (330  $\Omega$  X 1)**



## Polarity of Diode and LED



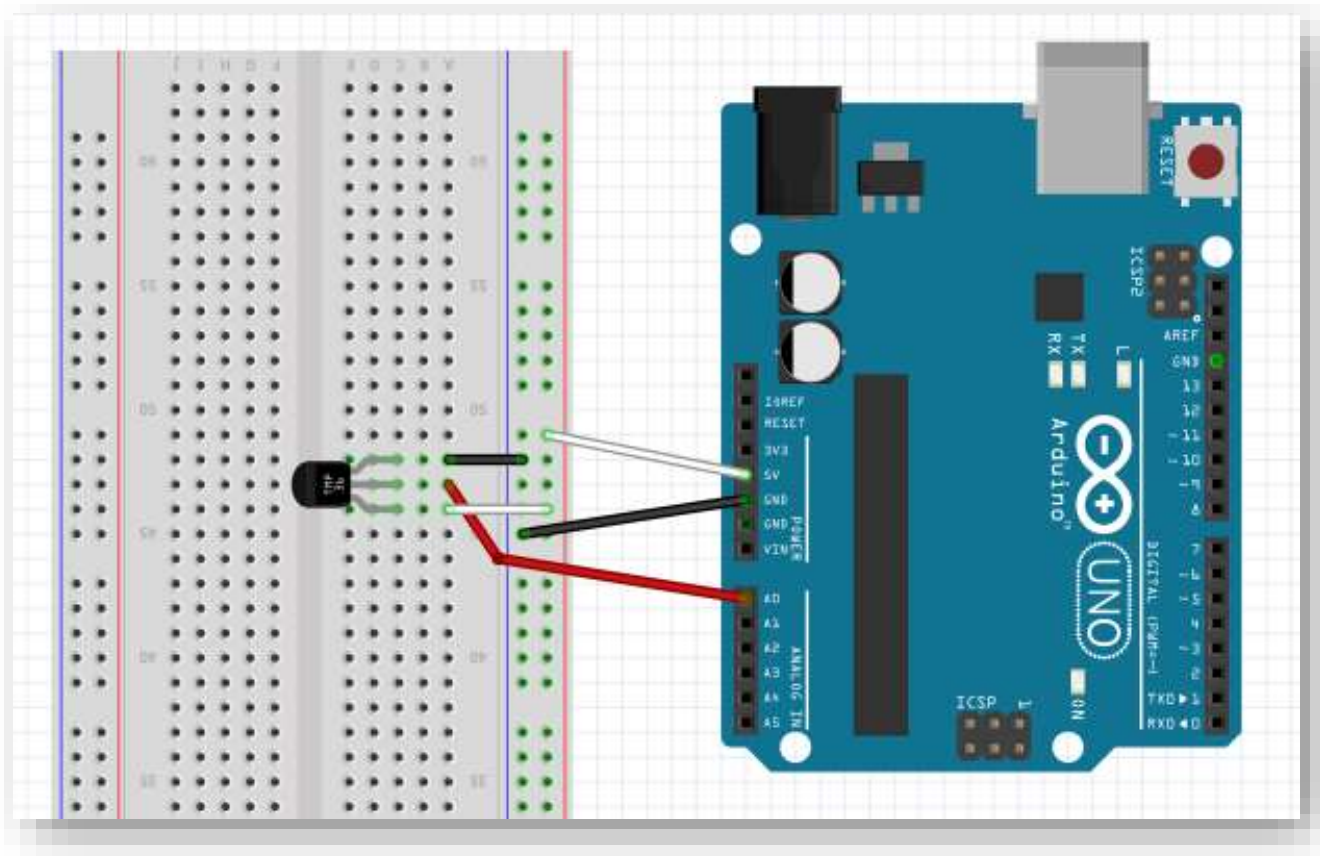
*The diode circuit symbol, with the anode and cathode marked.*



**Find the longer leg, which should indicate the positive, anode pin.**

<https://learn.sparkfun.com/tutorials/polarity/diode-and-led-polarity>

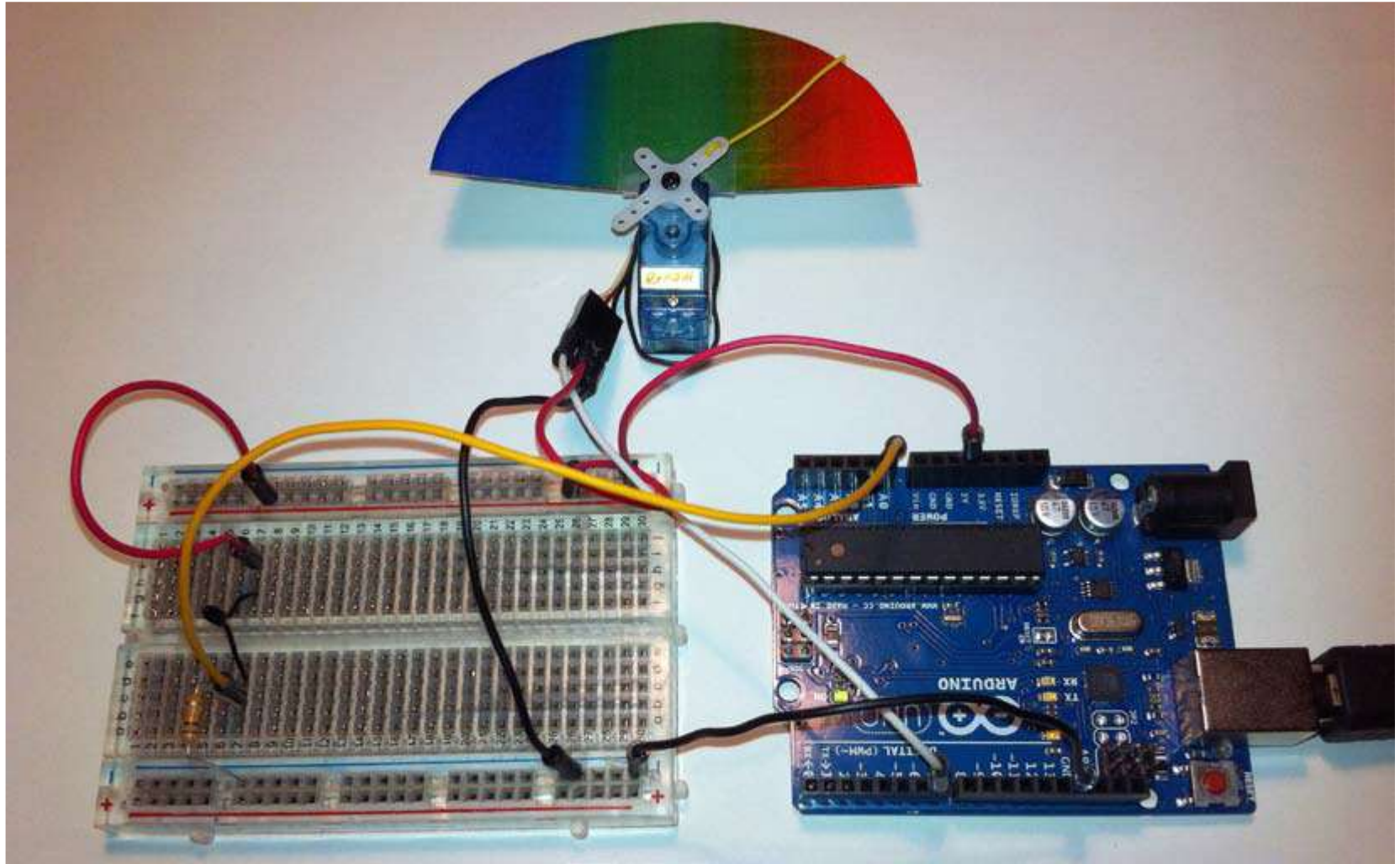
## 0.A3 Temperature sensor (TMP36)



**Parts : Temperature sensor (TMP36)**  
**A0 : analog signal input**

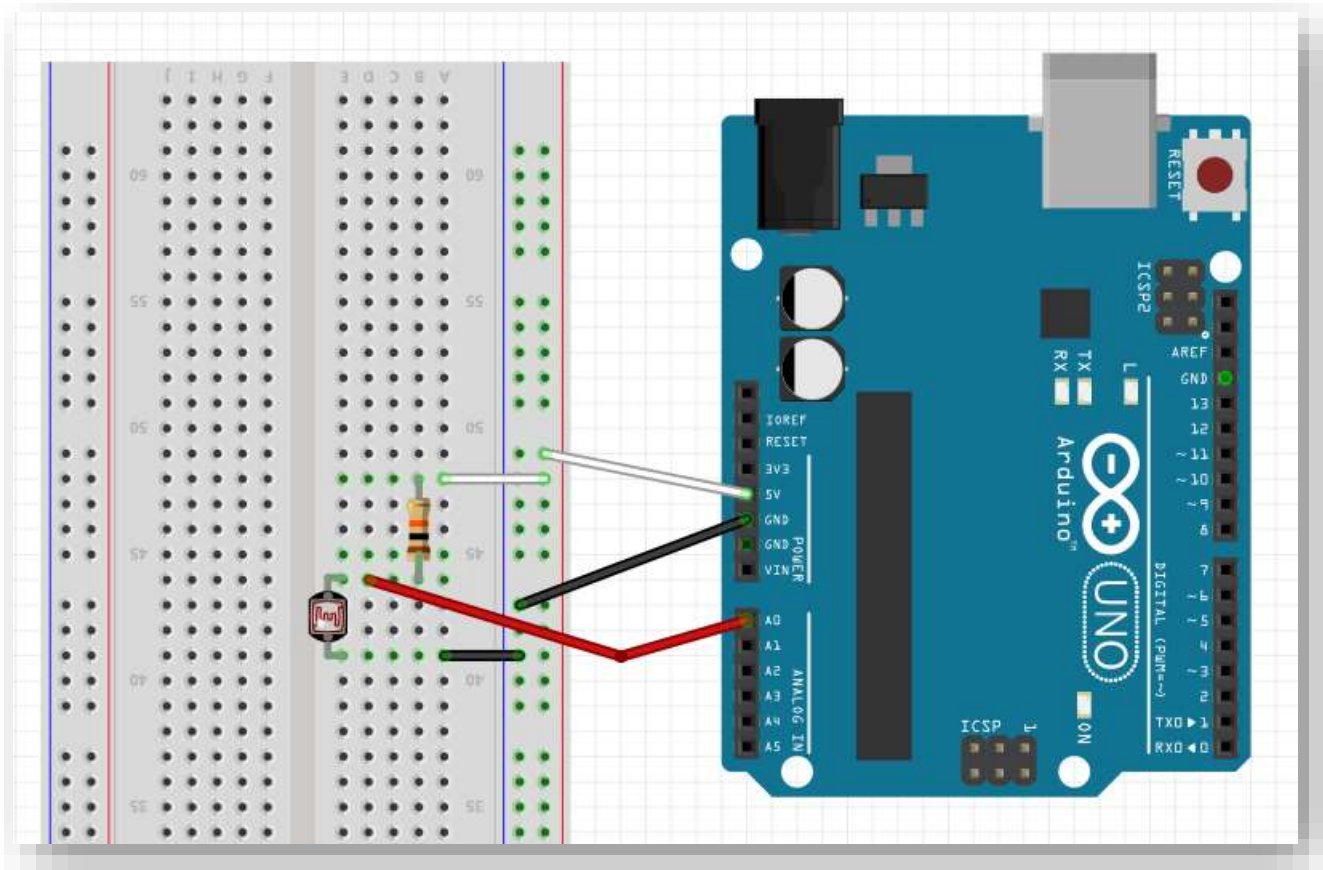


## 0.A3. DIY3 Servo



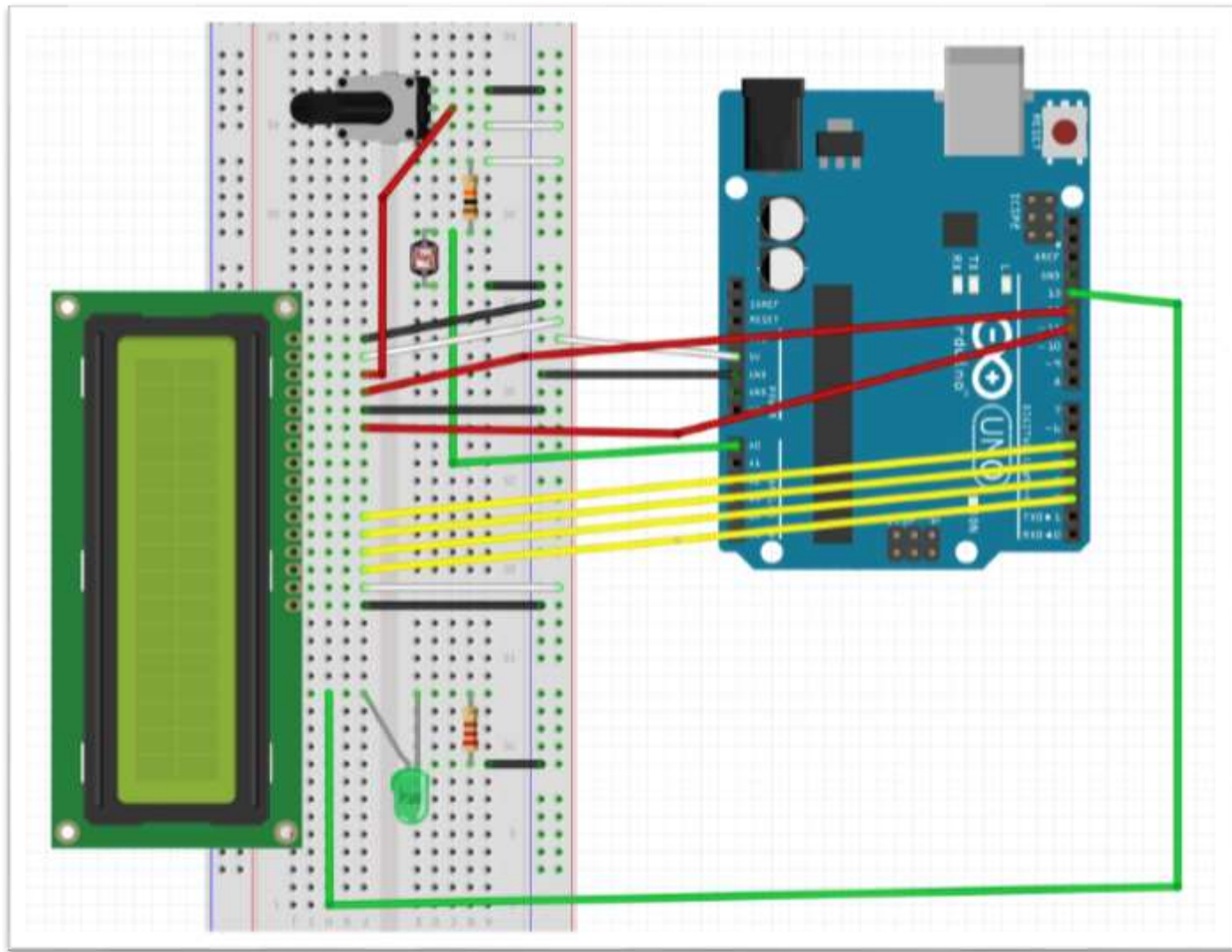


## 0.A4 Luminosity sensor : photo cell LDR



**Parts : 20 mm photocell LDR, R (10 k $\Omega$  X 1)**

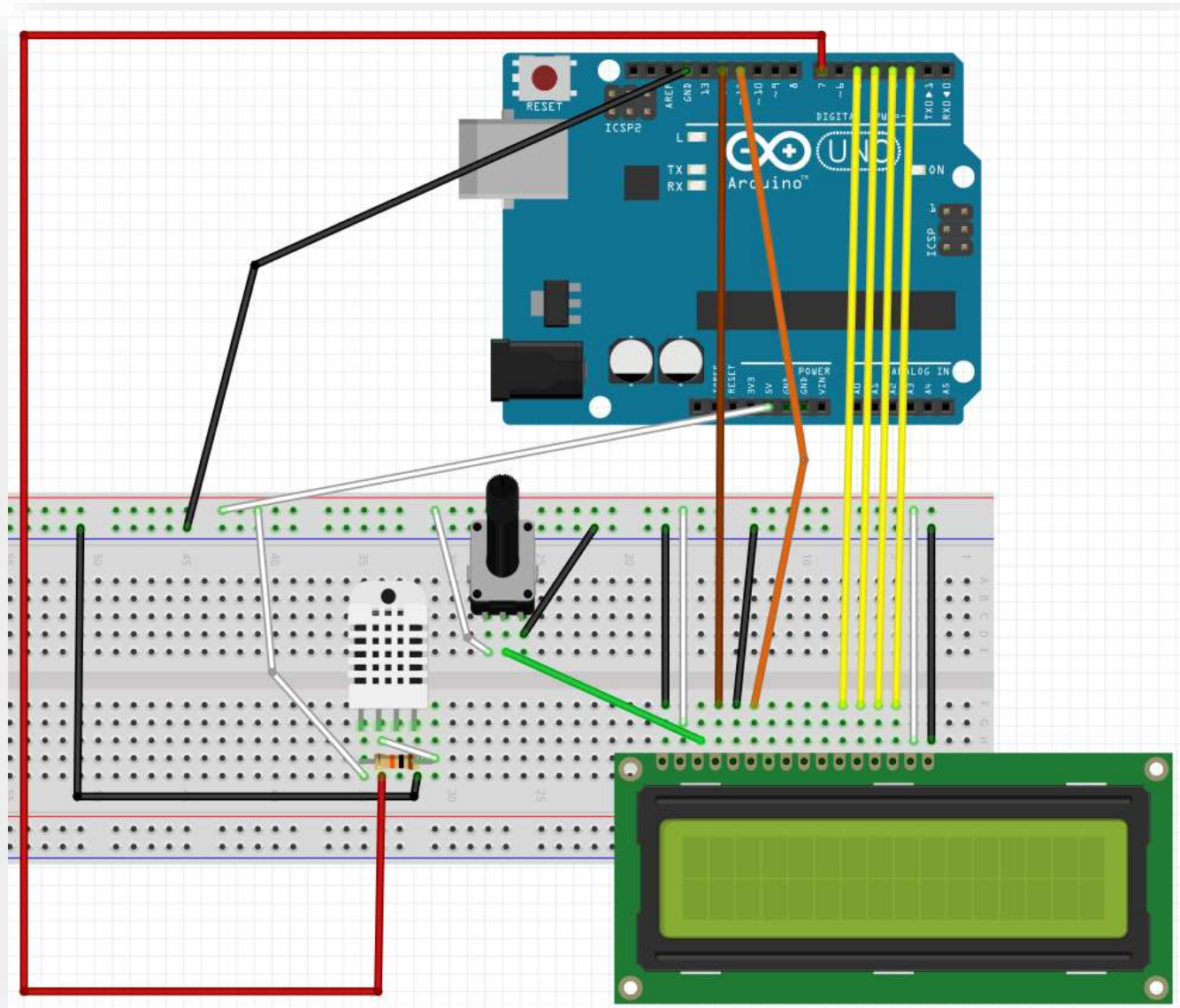
## 0.A5 Display of luminosity







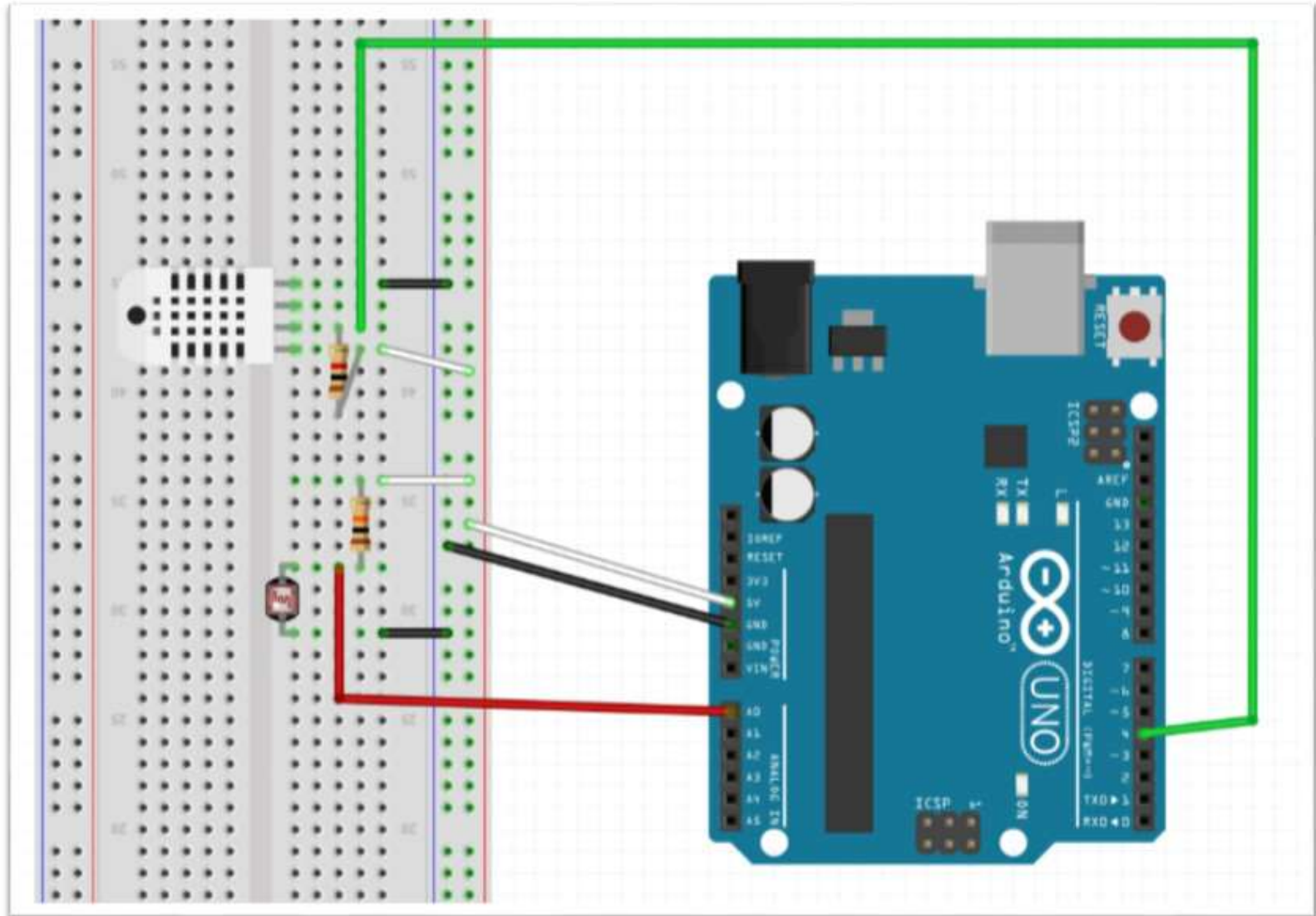
# 0.A6 Display of Temperature & Humidity







## 0.A7 DHT22 & CdS





# 1. Arduino SW: IDE

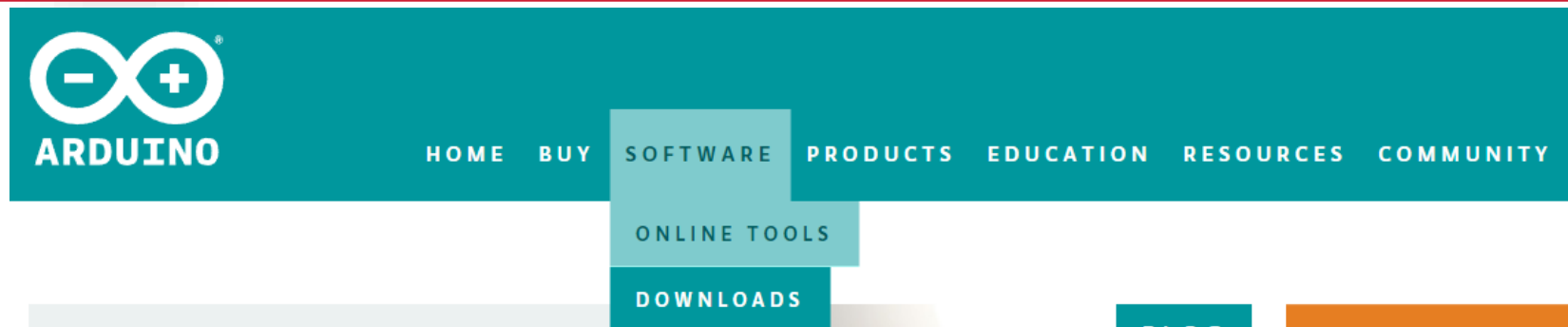


HOME BUY SOFTWARE PRODUCTS LEARNING FORUM SUPPORT BLOG

<https://www.arduino.cc/>



# A1.1 Arduino IDE – portable ver.



## Download the Arduino IDE



### ARDUINO 1.8.5

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** 10.7 Lion or newer

**Linux** 32 bits

**Linux** 64 bits

**Linux** ARM

[Release Notes](#)

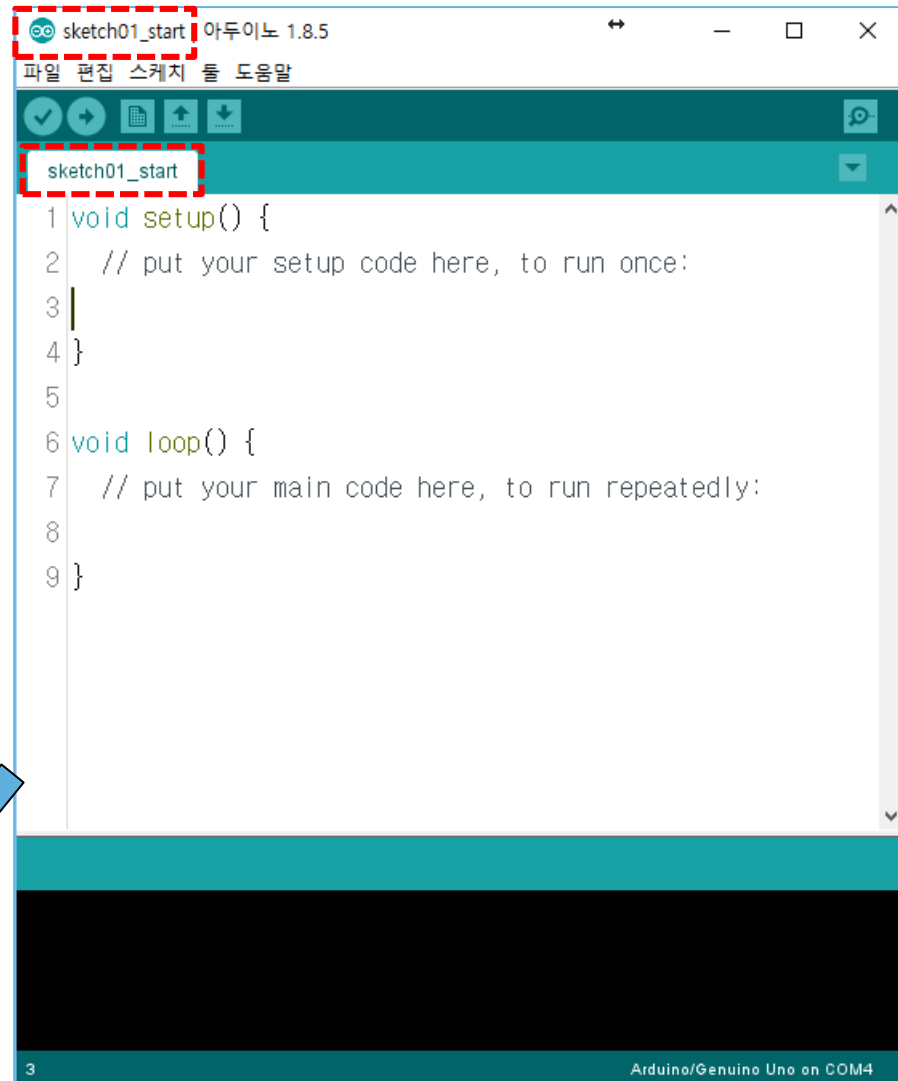
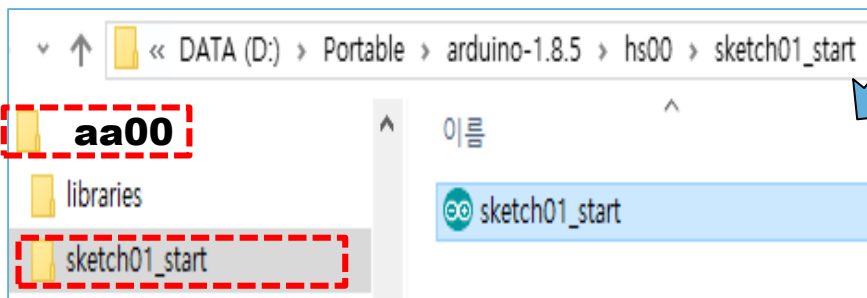
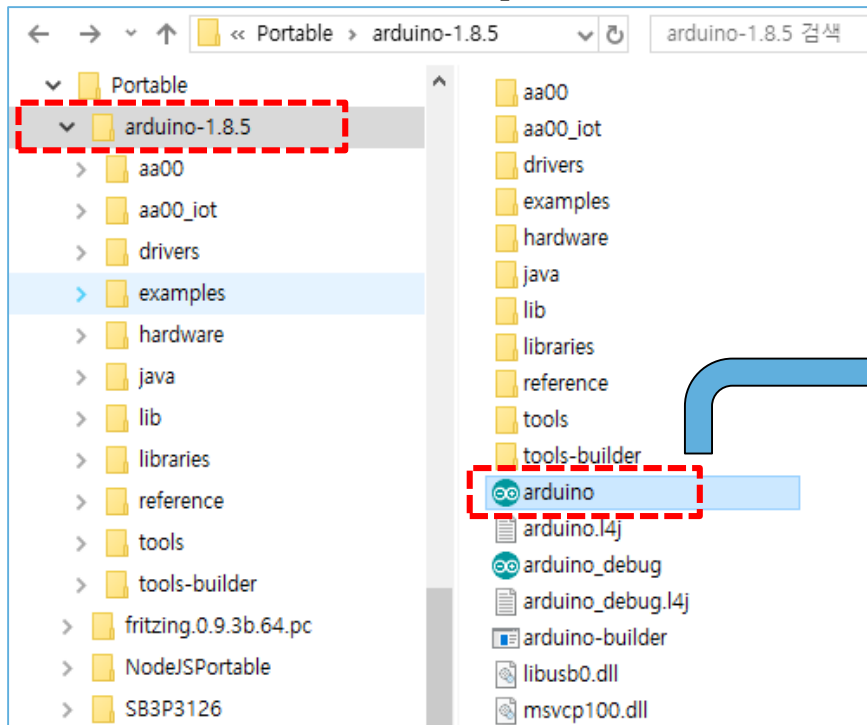
[Source Code](#)

[Checksums \(sha512\)](#)



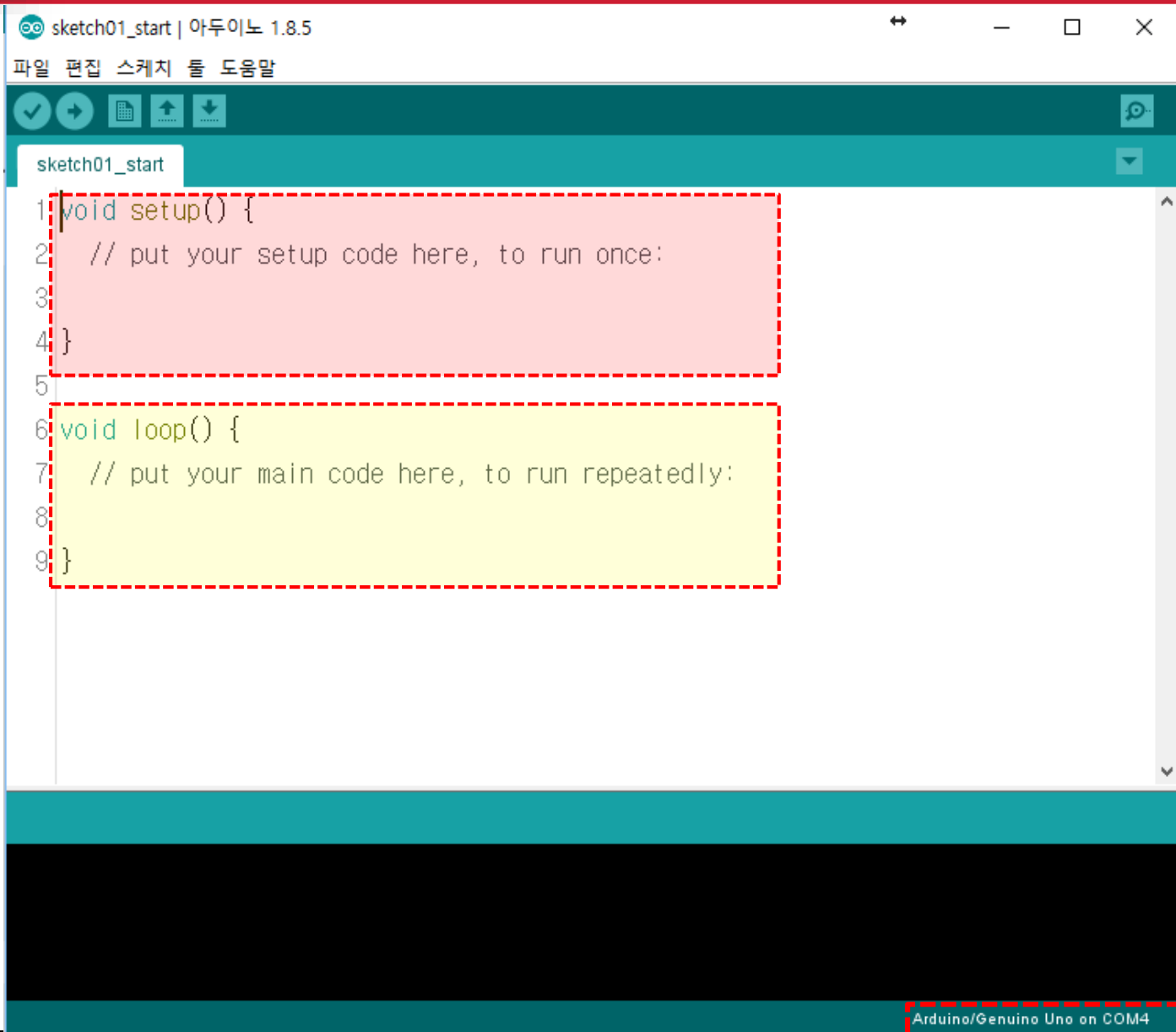
# A1.2 Arduino Portable (V1.8.6)

## Make folder **hsnn** in portable Arduino folder





# A1.3 Arduino Portable IDE





# A1.4 Arduino Portable IDE

sketch01\_start | 아두이노 1.8.5

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

환경설정

설정 네트워크

스케치북 위치:  
D:\Portable\arduino-1.8.5\aa00 찾아보기

에디터 언어: 시스템 기본설정 (아두이노를 재시작해야 함)

에디터 글꼴 크기: 18

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

☒ Aggressively cache compiled core

☒ 시작시 업데이트 확인

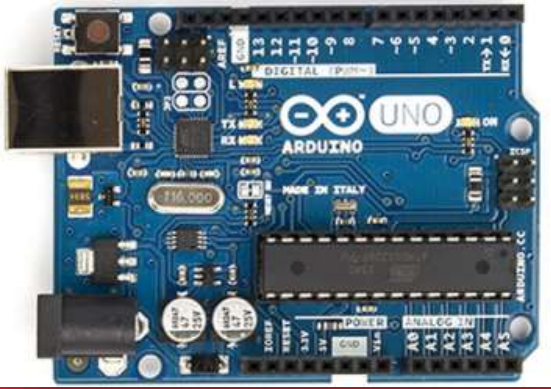
☒ 스케치 파일을 저장할때 새로운 확장자(.pde -> .ino)로 업데이트

☒ 검증 또는 업로드 할 때 저장하기

추가적인 보드 매니저 URLs

추가적인 환경 설정은 파일에서 직접 편집할 수 있습니다  
C:\Users\wish-HCIT\AppData\Local\Arduino15\preferences.txt  
(아두이노가 실행되지 않는 경우에만 수정 가능)

확인 취소



# LED

## LED (Light Emitting Diode)

- ✓ 전기 신호를 빛으로 출력하는 반도체 소자
- ✓ 고효율, 반영구적 수명
- ✓ 가정용 실내등, 산업용 특수등, 자동차용 전조등 및 실내등에 사용







## A2.1.1 Blink [ digitalWrite() ]

sketch01\_start | 아두이노 1.8.5

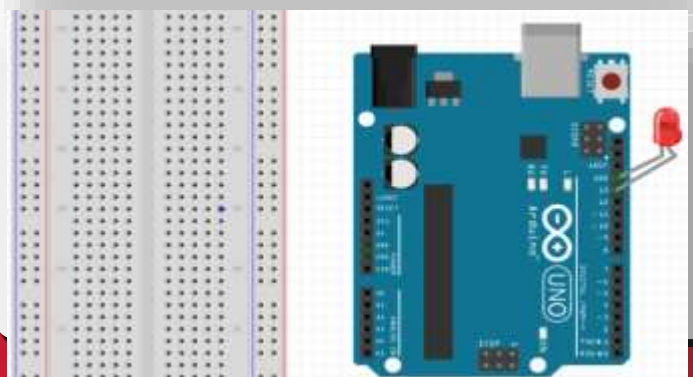
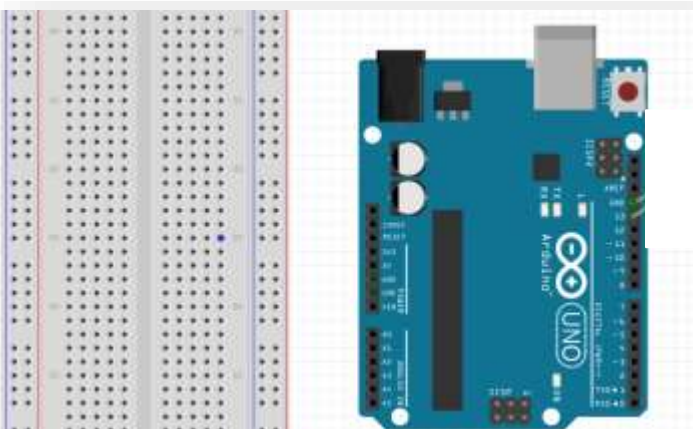
파일 편집 스케치 툴 도움말

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

내장된 예제

01. Basics  
02. Digital  
03. Analog  
04. Communication  
05. Control  
06. Sensors

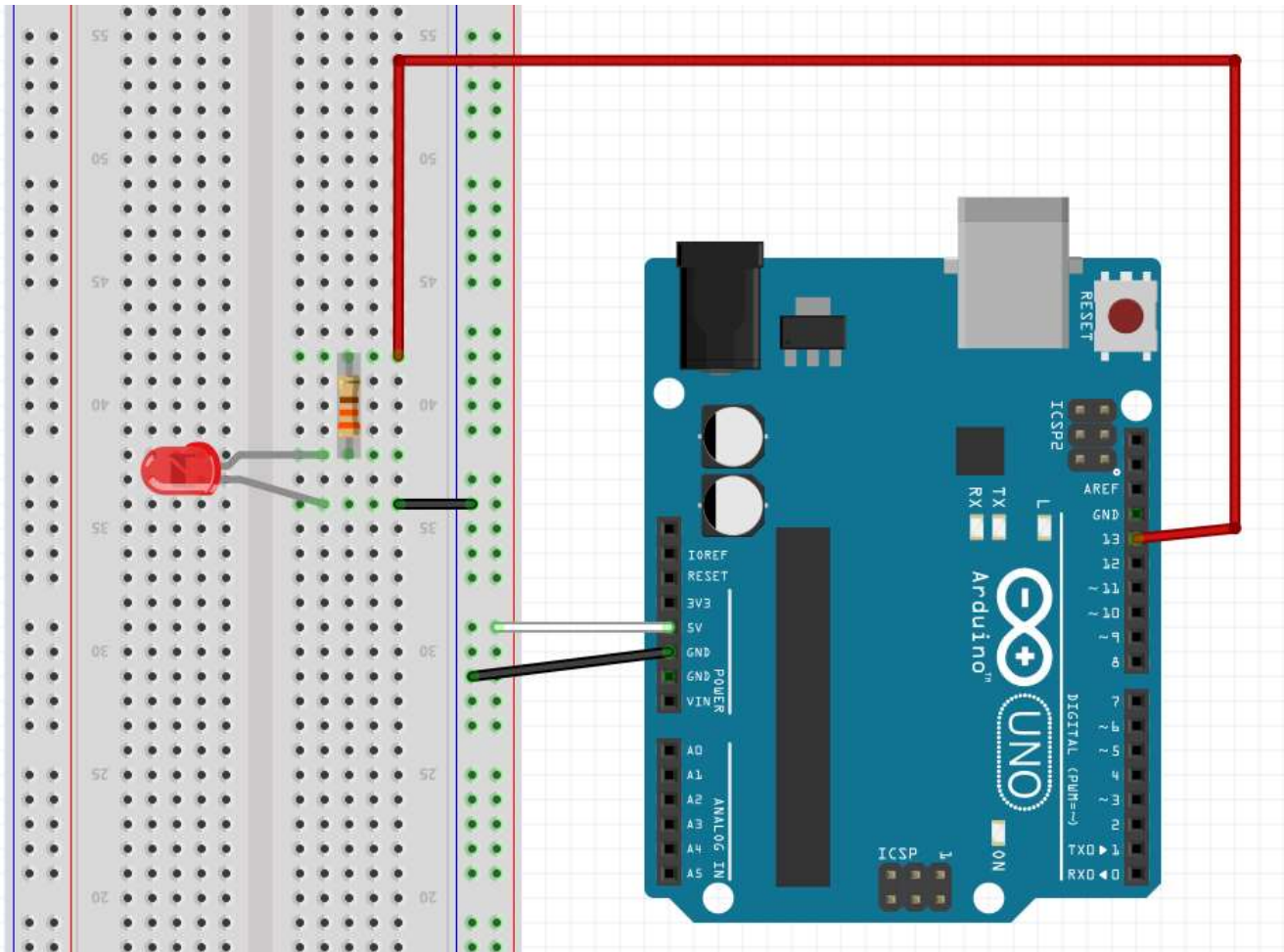
AnalogReadSerial  
BareMinimum  
Blink  
DigitalReadSerial  
Fade  
ReadAnalogVoltage



Blink 5

```
1 /*  
2  Blink  
3  Turns an LED on for one second, then off for one second, repeatedly.  
4 */  
5  
6 // the setup function runs once when you press reset or power the board  
7 void setup() {  
8   // initialize digital pin LED_BUILTIN as an output.  
9   pinMode(LED_BUILTIN, OUTPUT);  
10 }  
11  
12 // the loop function runs over and over again forever  
13 void loop() {  
14   digitalWrite(LED_BUILTIN, HIGH); // turn the LED on (HIGH is the voltage level)  
15   delay(1000); // wait for a second  
16   digitalWrite(LED_BUILTIN, LOW); // turn the LED off by making the voltage LOW  
17   delay(1000); // wait for a second  
18 }
```

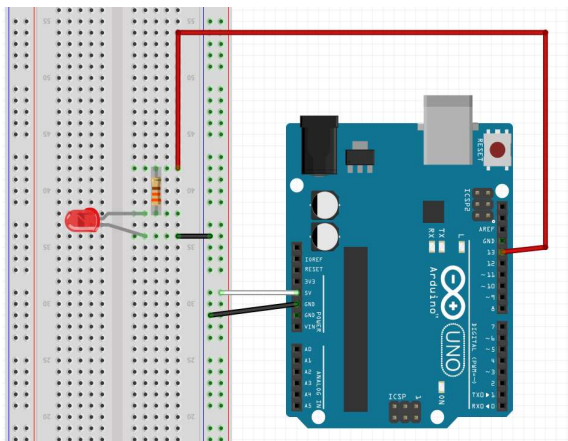
## A2.1.2 blink circuit



**Connect LED to D13 & GND  
with 330  $\Omega$**



## A2.1.3 blink [ modified your code, save it]



**Connect LED to  
D13 & GND  
with 330  $\Omega$**

aa00  
libraries  
sketch01\_blink  
sketch01\_start

이름

sketch01\_blink

sketch01\_blink\$

```
1 /*
2  Blink by AA00
3  Turns an LED on for one second, then off for one second, repeatedly.
4  */
5  int pinNum = 13; // D13
6
7  // the setup function runs once when you press reset or power the board
8  void setup() {
9    // initialize digital pin 13 as an output.
10    pinMode(pinNum, OUTPUT);
11  }
12
13 // the loop function runs over and over again forever
14 void loop() {
15    digitalWrite(pinNum, HIGH); // turn the LED on (HIGH is the voltage level)
16    delay(1000);                // wait for a second
17    digitalWrite(pinNum, LOW);  // turn the LED off by making the voltage LOW
18    delay(1000);                // wait for a second
19 }
```



## A2.2.1 LED control – 밝기 조절

### 밝기 조절 : 디밍 (Dimming)

- ✓ LED에 입력되는 전력은 **PWM (Pulse Width Modulation)**을 이용하여 조절.
- ✓ PWM : 고속의 스위칭으로 High와 Low 신호의 비율을 조절하여  
**LED의 밝기, 모터의 회전 등을 조절하는 방법**
- ✓ Arduino에서는 **analogWrite( )** 명령어로 구현
- ✓ Arduino UNO의 경우 **3, 5, 6, 9, 10, 11 번 핀이 PWM을 지원**한다.



## A2.2.2 LED control – 밝기 조절: PWM

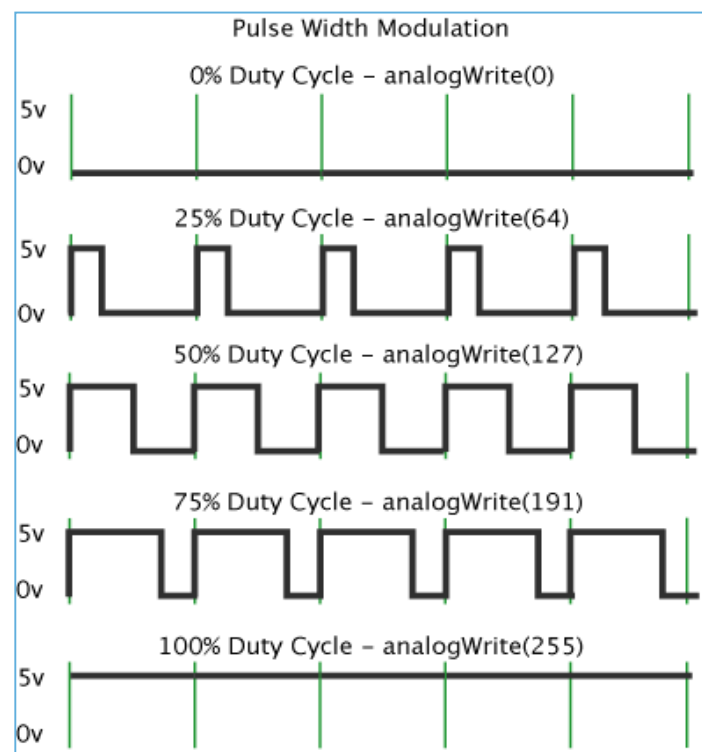
### PWM (Pulse Width Modulation)

Using [analogWrite\(pin, pwm\\_value\)](#) function in fading an LED off and on. AnalogWrite uses [pulse width modulation \(PWM\)](#), turning a digital pin on and off very quickly with different ratio between on and off, to create a fading effect.

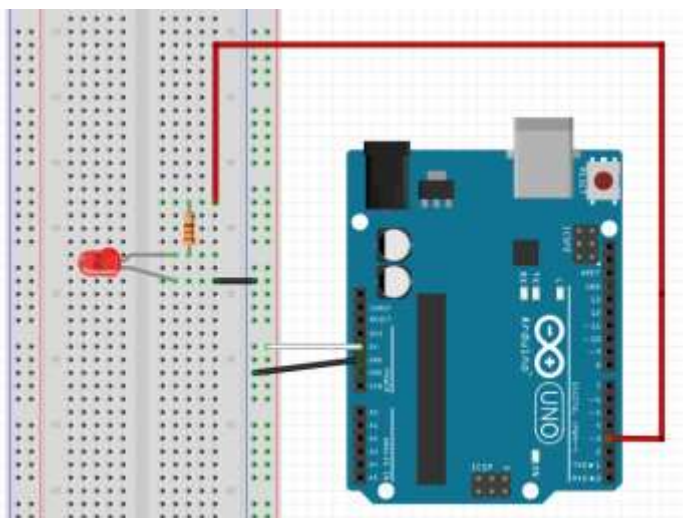
A call to [analogWrite\(\)](#) is on a scale of **0 - 255**, such that `analogWrite(255)` requests a 100% duty cycle (always on), and `analogWrite(127)` is a 50% duty cycle (on half the time)

PWM frequency = 500 Hz

<https://www.arduino.cc/en/Tutorial/PWM>



## A2.2.3 LED control – 밝기 조절: PWM



### ▶ 스케치 구성

1. LED의 핀 번호를 pwm 핀으로 설정한다. **D3**
2. 아날로그 출력에는 `setup()`에서의 핀 설정이 필요 없다.
3. `loop()`에서 마구잡이 수를 하나 발생시켜서 `analogWrite()` 함수로 LED의 밝기를 0.01초 간격으로 반복해서 변화시킨다.





## A2.2.4 LED control – 밝기 조절: PWM

### ▶ 사용 함수

- **analogWrite**(핀번호, 값)

정해진 핀에 아날로그 출력을 한다. '값'에는 0~255의 값을 넣는다.

- **random**(시작값, 종료값)

시작 값과 종료 값 사이의 정수를 마구잡이로 하나 만들어 반환한다.

- **pwmLed**(핀번호, 값)

정해진 PWM 출력 핀에 0~255의 pwm 값으로 아날로그 출력을 하는 사용자 정의 함수이다.



## A2.2.5 LED control – 밝기 조절: code

▶ 아두이노 코드 : sketch02\_pwm\_led.ino

```
int pwm = 0;
int led = 3; // D3

void setup() {
  // 아날로그 출력에서 핀 모드 설정이 필요 없다.
}

void loop() {

  pwm = random(0,255);
  pwmLed(led , pwm);
}

void pwmLed(int led, int pwmValue) {

  analogWrite(led, pwmValue);
  delay(10);
}
```

실습 결과

**LED**의 밝기가

**0.01초** 간격으로 마구  
잡이로 변하는 것을 확  
인





# 신호 발생 및 모니터링

## Serial monitor & plotter

## 시리얼 통신

### UART (Universal Asynchronous Receiver/Transmitter)

RS-232

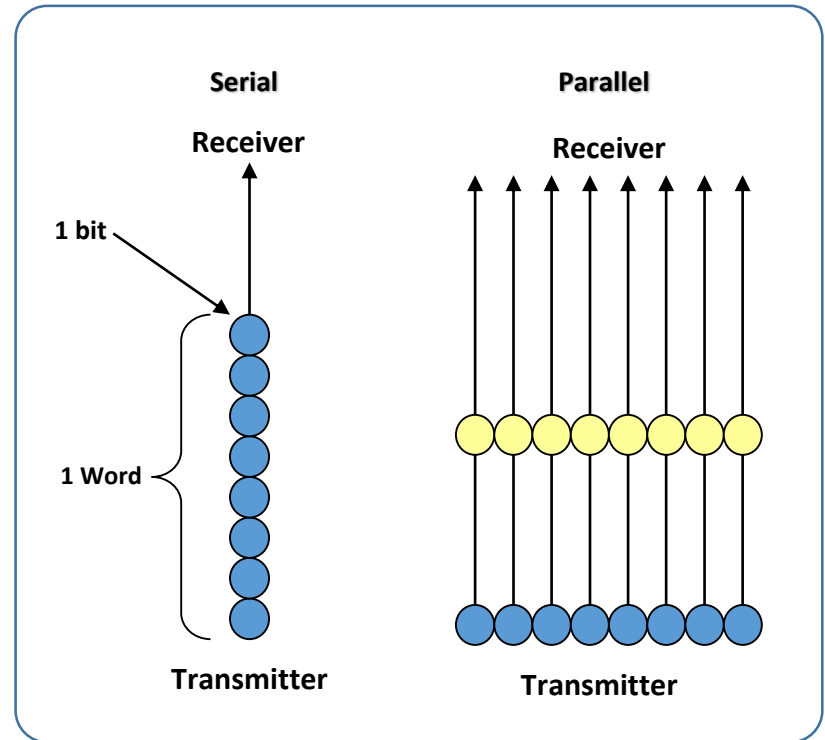
RS-422

RS-485

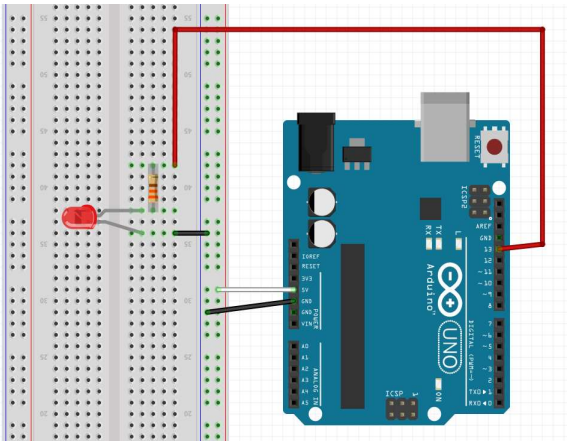
Arduino에서는 다음과 같은 목적으로 사용

Debugging : 프로그램의 오류를 수정하는 작업

데이터 통신 : Arduino와 컴퓨터 혹은 다른 장치와의 통신



# A2.3.1 LED 밝기 조정 및 모니터링 - 스케치



## ▶ 스케치 구성

1. LED의 핀 번호를 pwm 핀으로 설정한다.
2. setup()에서 직렬 통신 속도를 9600 bps 로 설정하고 컴퓨터와 연결한다.
3. loop()에서 마구잡이 수를 하나 발생시켜서 analogWrite() 함수로 LED의 밝기를 반복해서 변화시키면서 직렬 통신으로 pwm 값을 전송한다.



## A2.3.2 LED 밝기 조정 및 모니터링 - 함수

### ▶ 사용 함수

- **Serial.begin(전송속도)**

직렬 통신 포트를 컴퓨터와 연결한다. 전송속도는 bps (bits per sec)로 일반적으로 9600으로 설정한다. 19200, 57600, 115200 등의 값을 설정할 수 있다.

- **Serial.print(전송내용)**

괄호 안의 내용을 직렬 통신으로 전송한다. 따옴표로 구분된 부분은 텍스트를 직접 전송하고 따옴표 없이 변수를 써주면 변수의 값이 전송된다.

- **Serial.println(전송내용)**

‘Serial.print’와 같으나 전송 뒤 줄 바꿈을 한다.



## A2.3.3 LED 밝기 조정 및 모니터링 – code

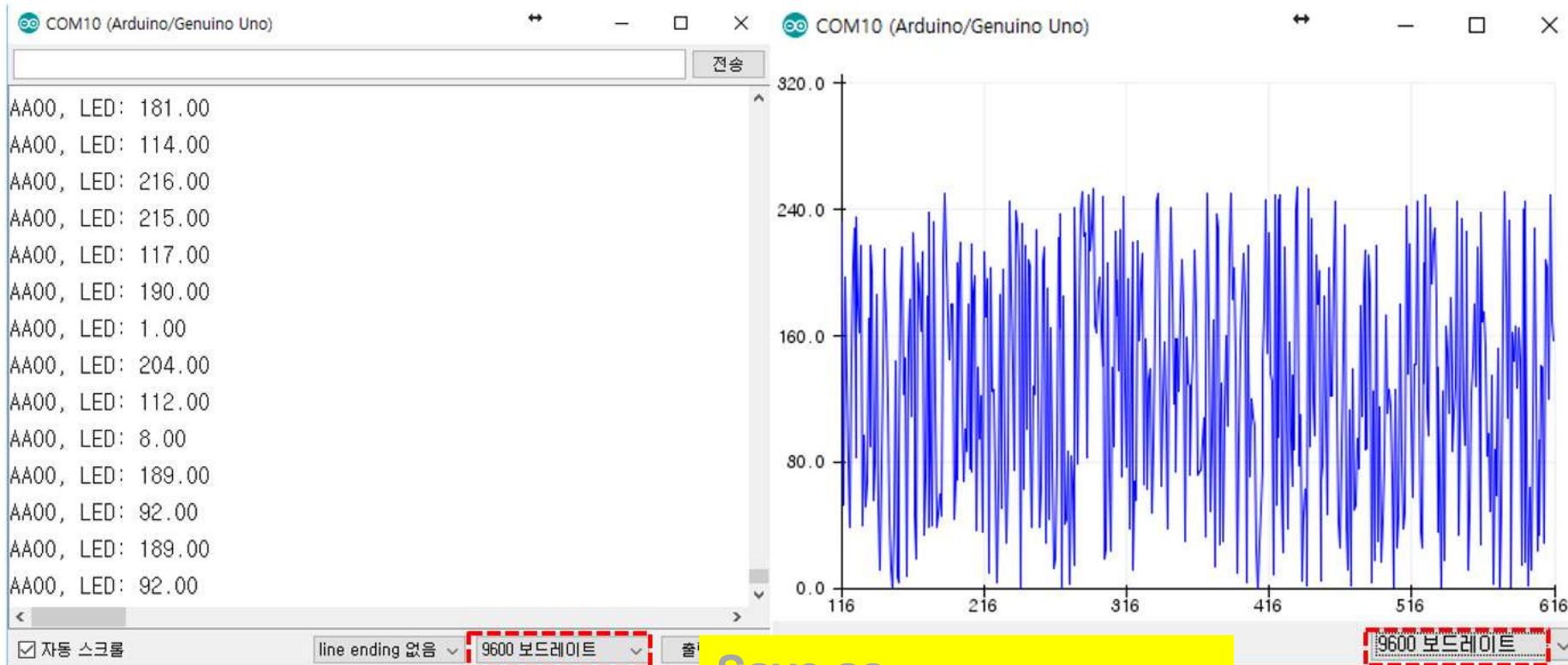
```
sketch03_pwm_led_serial
1 // sketch03_pwm_led_serial.ino
2 int pwm = 0;
3 int led = 3;
4
5 void setup() {
6   Serial.begin(9600);
7 }
8
9 void loop() {
10  // put your main code here:
11  pwm = random(0,255);
12  pwmLed(led , pwm);
13
14  Serial.print("AA00, LED: ");
15  Serial.println(pwm);
16  delay(10);
17 }
18
19 void pwmLed(int led, int pwmValue) {
20   analogWrite(led, pwmValue);
21   delay(10);
22 }
```



## A2.3.4 LED 밝기 조정 및 모니터링 – 결과

### 실습 결과

**LED**의 밝기가 **0**에서 **255** 단계로 마구잡이로 변하는 것을 확인할 수 있으며  
직렬모니터와 직렬플로터로 **pwm**의 값의 변화를 모니터링 할 수 있다.

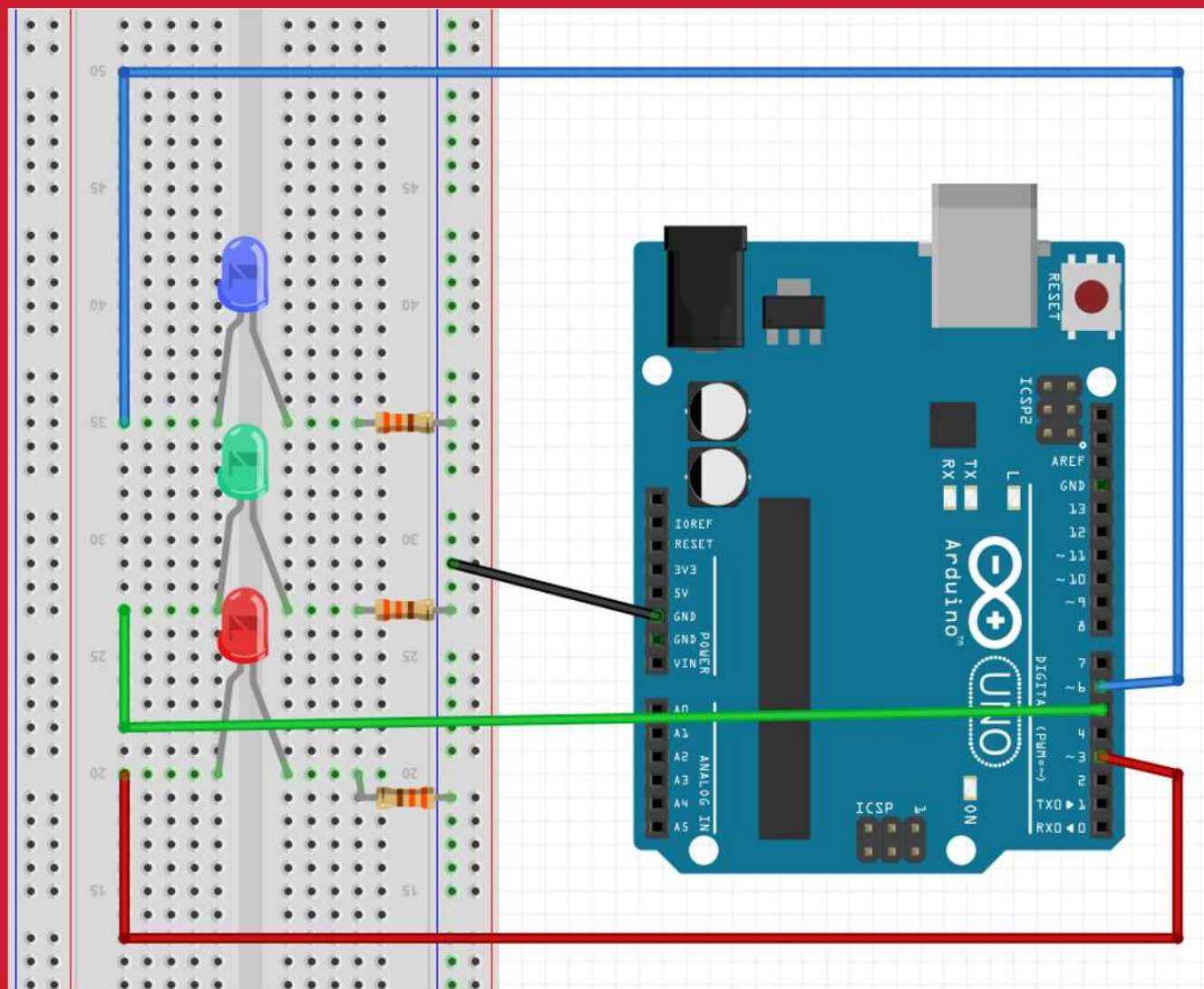


Save as

AAnn\_Monitoring.png

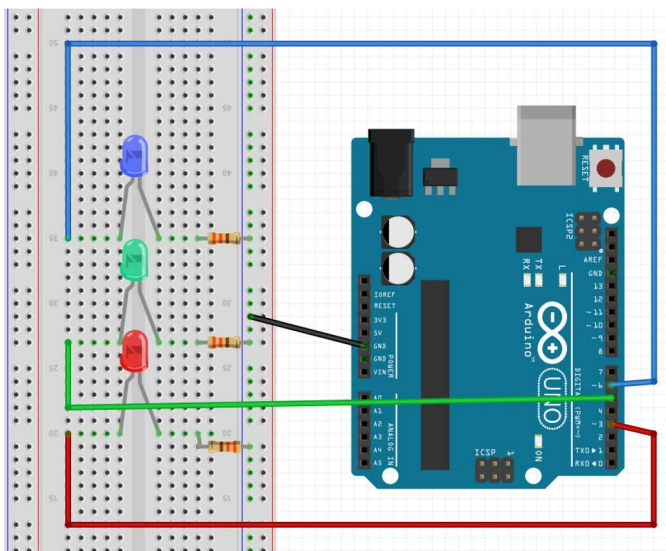


# 3 LED 모니터링





# A2.4.1 3개의 LED 밝기 조정 및 모니터링 - 스케치



## ▶ 스케치 구성

1. 3 개의 LED의 핀 번호를 각각 다른 pwm 핀 (3, 5, 6)으로 설정한다.
2. setup()에서 직렬 통신 속도를 9600 bps 로 설정하고 컴퓨터와 연결한다.
3. loop()에서 마구잡이 수를 세 개 발생시켜서 analogWrite() 함수로 세 개의 LED의 밝기를 각각 반복해서 변화시킨다.
4. 직렬 통신으로 3 개의 pwm 값을 한 줄로 컴퓨터로 전송한다.



## A2.4.2 3개의 LED 밝기 조정 및 모니터링 – code

```
sketch04_pwm_3_leds
1 // pwm_3_leds.ino
2 int pwm1 = 0;
3 int pwm2 = 0;
4 int pwm3 = 0;
5
6 int ledR = 3;
7 int ledG = 5;
8 int ledB = 6;
9
10 void setup() {
11
12     Serial.begin(9600);
13 }
```

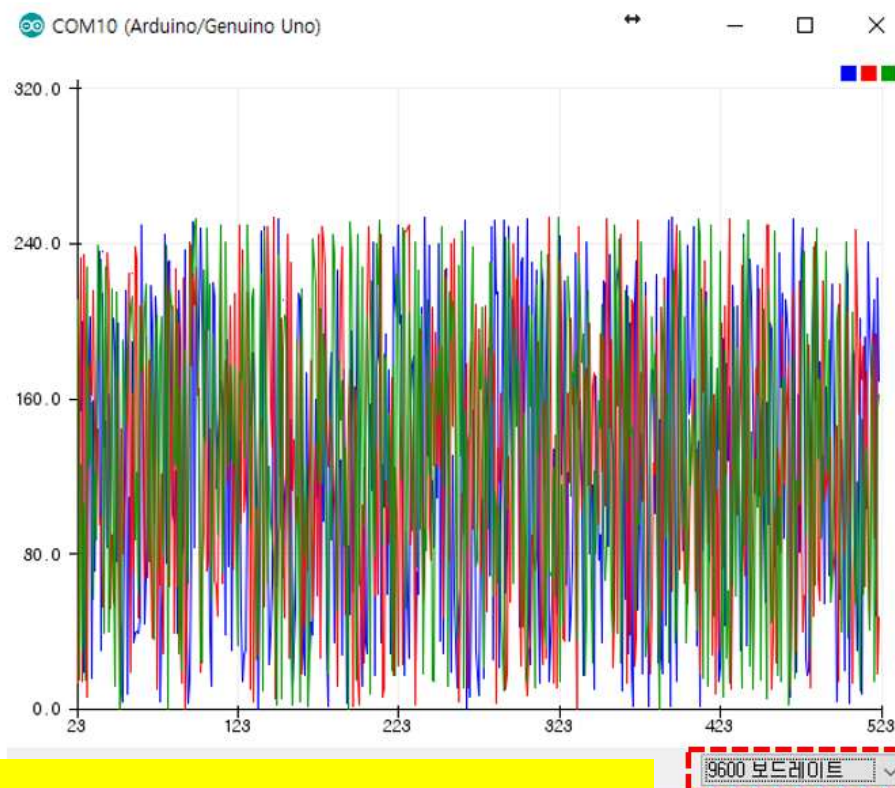
```
15 void loop() {
16
17     pwm1 = random(0,255);
18     pwm2 = random(0,255);
19     pwm3 = random(0,255);
20     pwmLed(ledR, pwm1);
21     pwmLed(ledG, pwm2);
22     pwmLed(ledB, pwm3);
23
24     Serial.print("AA00, LED_R: ");
25     Serial.print(pwm1);
26     Serial.print(" , LED_G: ");
27     Serial.print(pwm2);
28     Serial.print(" , LED_B: ");
29     Serial.println(pwm3);
30     delay(10);
31 }
32
33 void pwmLed(int led, int pwmValue) {
34     analogWrite(led, pwmValue);
35     delay(10);
36 }
```

# A2.4.3 3개의 LED 밝기 조정 및 모니터링 – 결과

## 실습 결과

세 개의 **LED**의 밝기가 각각 **0**에서 **255** 단계로 마구잡이로 변하는 것을 확인할 수 있다. 직렬모니터와 직렬플로터로 세 개의 **pwm**의 값의 변화를 모니터링 한다.

```
COM10 (Arduino/Genuino Uno)
전송
AA00, LED_R: 46 , LED_G: 190 , LED_B: 208
AA00, LED_R: 250 , LED_G: 209 , LED_B: 173
AA00, LED_R: 145 , LED_G: 180 , LED_B: 180
AA00, LED_R: 44 , LED_G: 67 , LED_B: 206
AA00, LED_R: 69 , LED_G: 192 , LED_B: 219
AA00, LED_R: 115 , LED_G: 68 , LED_B: 101
AA00, LED_R: 87 , LED_G: 180 , LED_B: 76
AA00, LED_R: 218 , LED_G: 74 , LED_B: 187
AA00, LED_R: 198 , LED_G: 37 , LED_B: 140
AA00, LED_R: 95 , LED_G: 122 , LED_B: 36
AA00, LED_R: 213 , LED_G: 172 , LED_B: 196
AA00, LED_R: 195 , LED_G: 10 , LED_B: 203
AA00, LED_R: 125 , LED_G: 126 , LED_B: 72
AA00, LED_R: 4 , LED_G: 64 , LED_B: 186
AA00, LED_R: 51 , LED_G: 121 , LED_B:
[자동 스크롤] line ending 없음 9600 보드레이트
```



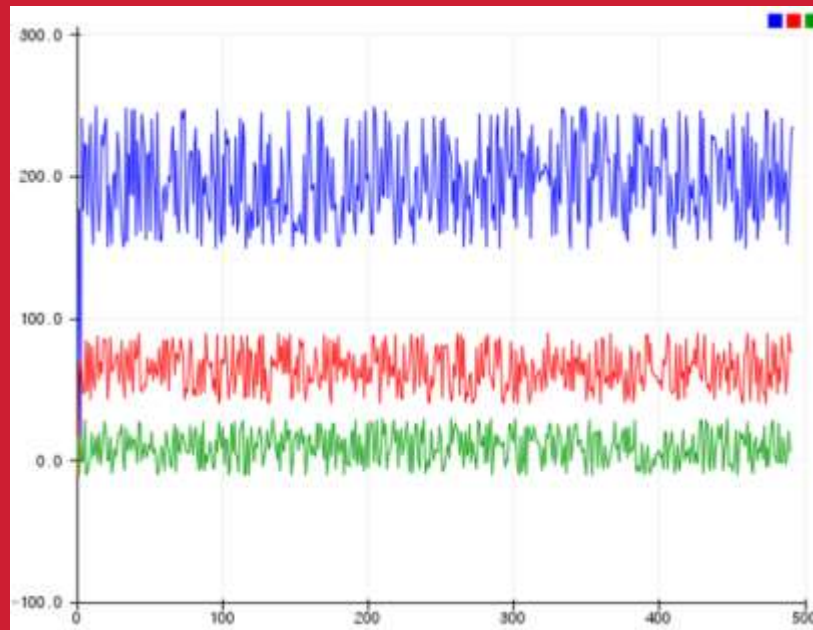
Save as

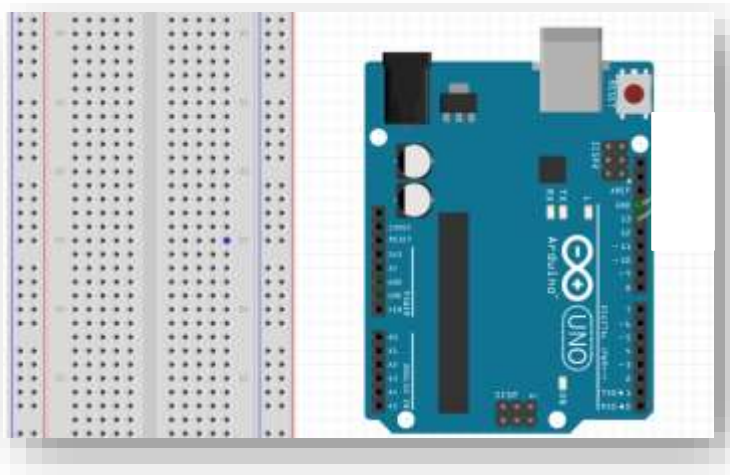
AAnn\_multi\_Monitoring.png



# [DIY] Multi-signals

## 다중신호 시뮬레이션 및 모니터링





아두이노에서 **LED**와 저항을 모두 제거하고 **USB**만 컴퓨터와 연결한다.

전자 소자 연결 없이 마구잡이 수 생성 함수를 이용해서 조도, 습도, 온도에 해당하는 **3**개의 신호를 만든다.

온도는 값의 범위를 **-10 ~ 30**, 습도는 **40 ~ 90**, 그리고 조도는 **150 ~ 250** 으로 가상적 으로 설정한다.

직렬통신 모니터링을 이용해서 세 개의 신호의 변화를 모니터링 하는 코드를 만들어 결과를 확인한다.

## ▶ 스케치 구성

1. 3 개의 신호를 담은 변수를 초기화한다.
2. setup()에서 직렬 통신 속도를 9600 bps 로 설정하고 컴퓨터와 연결한다.
3. loop()에서 마구잡이 수를 세 개 발생시켜서 직렬 통신으로 3 개의 pwm 값을 각각 컴퓨터로 전송한다.



# DIY - code

sketch05\_multi\_signals

```
1 /*
2   Multi Signals
3   Simulation of multiple random signals
4 */
5 // signals
6 int humi=0;
7 int temp=0;
8 int lux=0;
9
```

```
10 // the setup routine runs once when you press reset:
11 void setup() {
12   // initialize serial communication at 9600 bits per second:
13   Serial.begin(9600);
14 }
15
16 // the loop routine runs over and over again forever:
17 void loop() {
18   // Multi signals
19   humi = random(40,90);
20   temp = random(-10, 30);
21   lux = random(150,250);
22   Serial.print("AA00, Ambient lux: ");
23   Serial.print(lux);
24   Serial.print(" , Humidity: ");
25   Serial.print(humi);
26   Serial.print(" , Temperature: ");
27   Serial.println(temp);
28   delay(500);      // delay in between reads for stability
29 }
```





# DIY - result

## DIY 결과

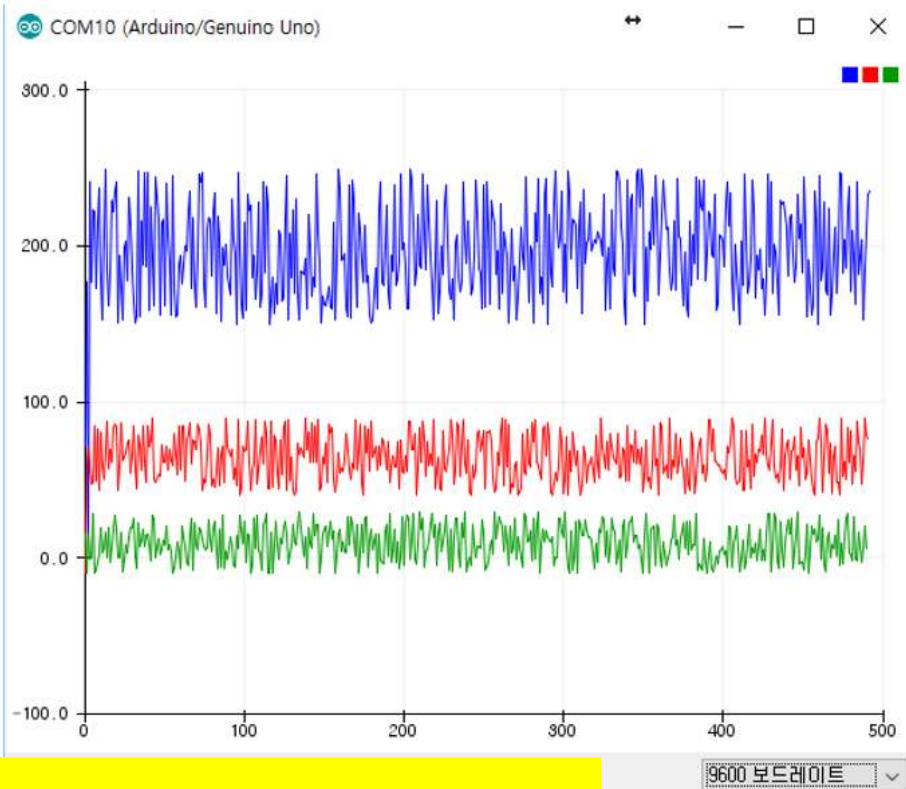
가상적인 세 개의 센서 신호 시뮬레이션: 조도(위), 습도(중간), 온도(아래).

COM10 (Arduino/Genuino Uno)

전송

```
AA00, Ambient lux: 186 , Humidity: 54 , Temperature: 13
AA00, Ambient lux: 165 , Humidity: 65 , Temperature: 19
AA00, Ambient lux: 151 , Humidity: 84 , Temperature: 19
AA00, Ambient lux: 155 , Humidity: 57 , Temperature: 25
AA00, Ambient lux: 248 , Humidity: 44 , Temperature: 1
AA00, Ambient lux: 155 , Humidity: 78 , Temperature: -7
AA00, Ambient lux: 216 , Humidity: 72 , Temperature: 22
AA00, Ambient lux: 188 , Humidity: 56 , Temperature: 7
AA00, Ambient lux: 247 , Humidity: 84 , Temperature: 11
AA00, Ambient lux: 187 , Humidity: 61 , Temperature: 18
AA00, Ambient lux: 247 , Humidity: 48 , Temperature: 7
AA00, Ambient lux: 159 , Humidity: 84 , Temperature: 14
AA00, Ambient lux: 225 , Humidity: 71 , Temperature: 15
AA00, Ambient lux: 192 , Humidity: 75 , Tempera
```

☒ 자동 스크롤 line ending 없음 9600 보드레이트

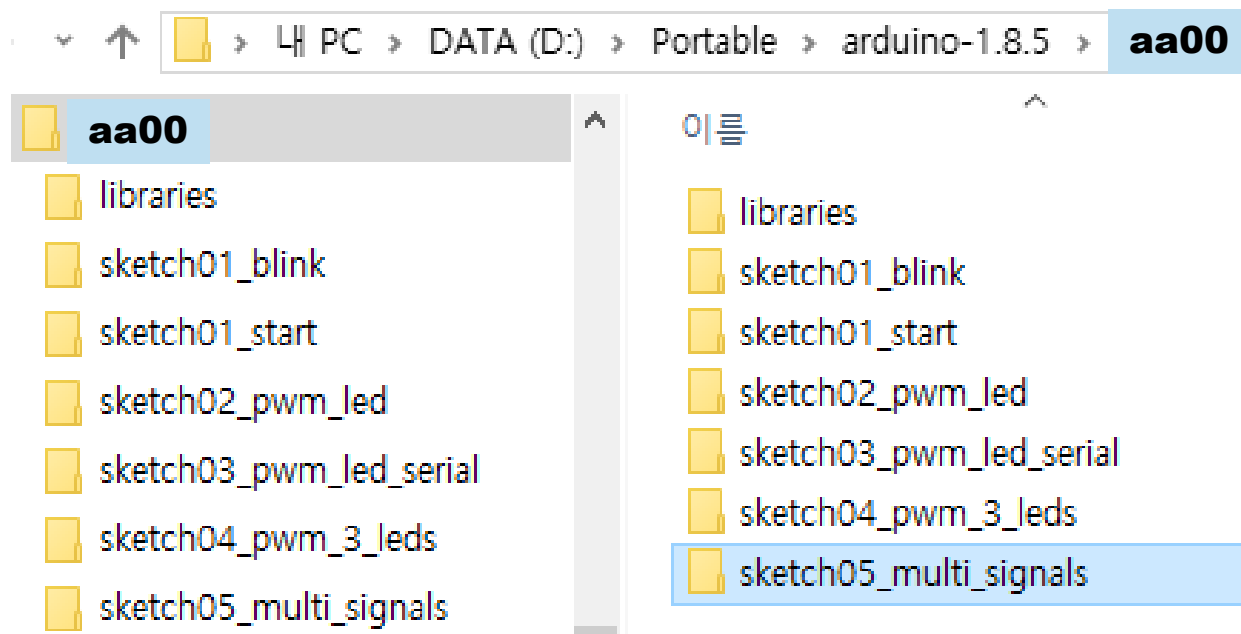


Save as

AAnn\_multi\_Signals.png



# [My working folder]



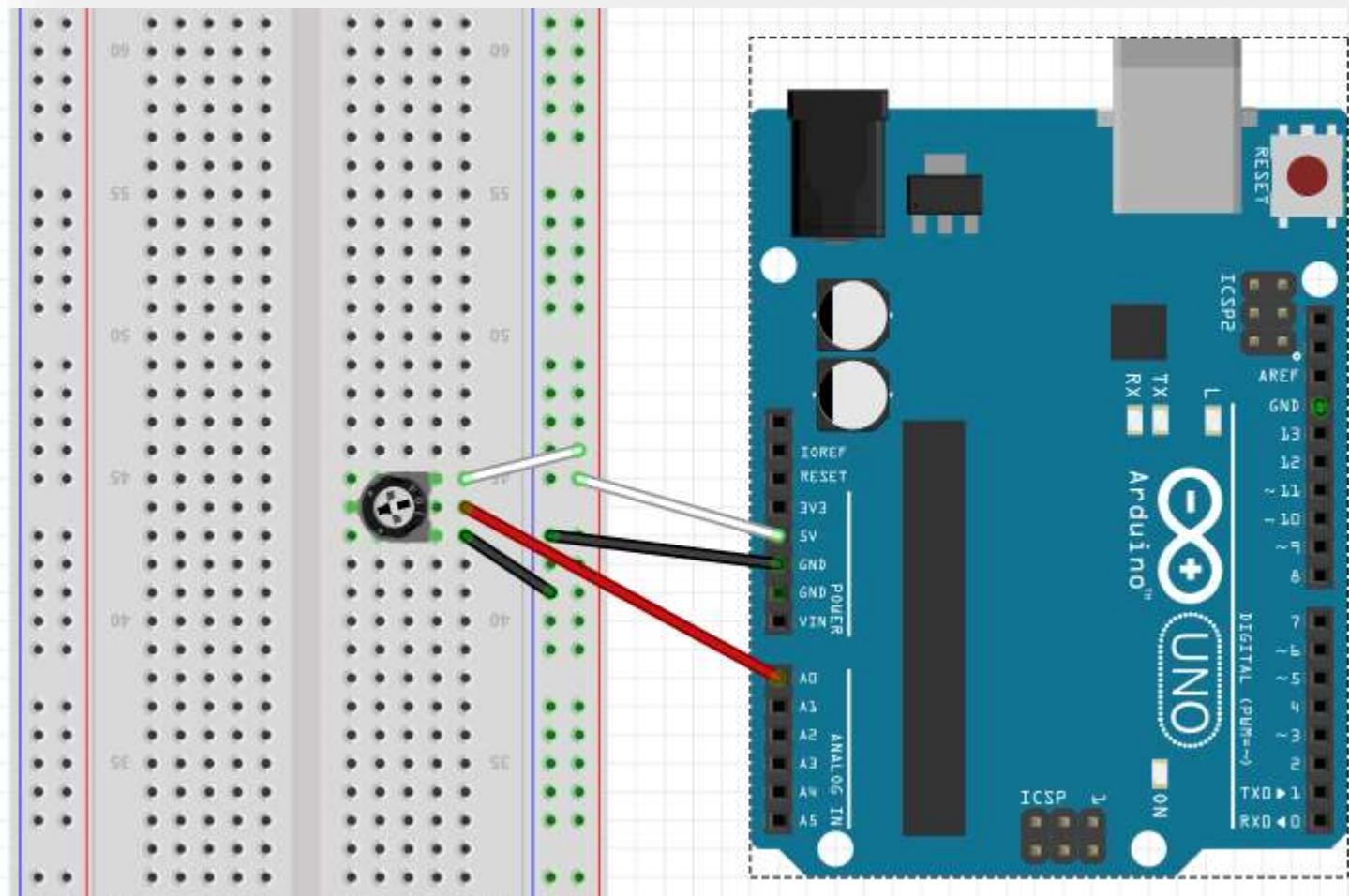




# Analog Signal

## A2.5.1 AnalogReadSerial (circuit)

### Standard potentiometer (가변 저항기)





# [Practice]

## ◆ [wk04]

- **Arduino basic circuits**
- **Complete your project**
- **Submit file : AAnn\_Rpt03.zip**

## ◆ [Target of this week]

- Complete your works
- Save your outcomes and compress 3 figures

제출파일명 : **AAnn\_Rpt03.zip**

- 압축할 파일들

- ① **AAnn\_Monitoring.png**
- ② **AAnn\_multi\_Monitoring.png**
- ③ **AAnn\_multi\_Signals.png**

**Email : [chaos21c@gmail.com](mailto:chaos21c@gmail.com) [ 제목 :  
id, 이름 (수정) ]**

## ● References & good sites

- ✓ <http://www.arduino.cc> Arduino Homepage
- ✓ <http://www.nodejs.org/ko> Node.js
- ✓ <https://plot.ly/> plotly
- ✓ <https://www.mongodb.com/> MongoDB
- ✓ <http://www.w3schools.com> By w3schools
- ✓ <http://www.github.com> GitHub



# 주교재 및 참고도서

아두이노와 Node.js에 기반한 IOT 신호 시각화

| 저자 이 상 훈 |

인제대학교 출판부

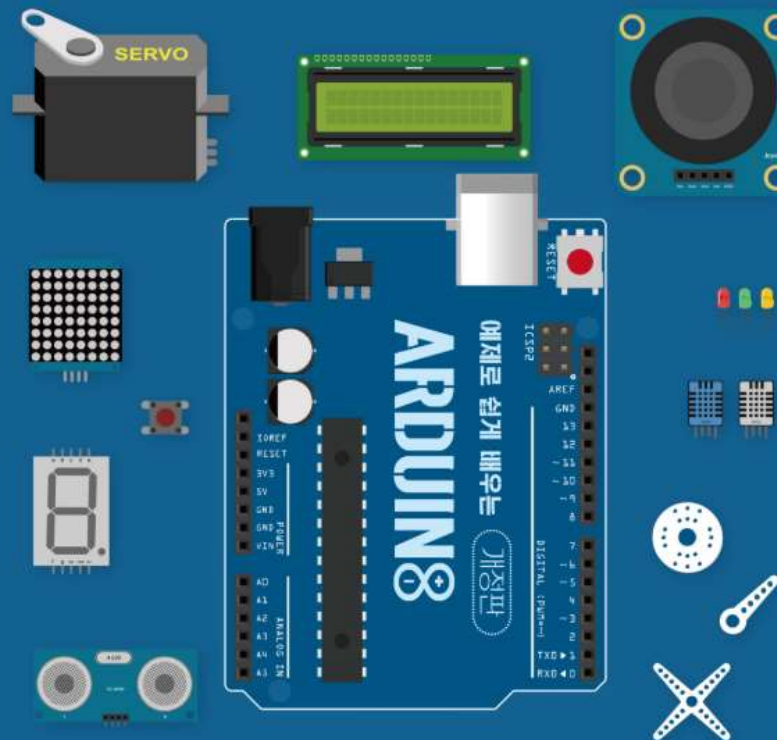
아두이노와 Node.js에 기반한

## IOT 신호 시각화

| 저자 이 상 훈 |



인제대학교 출판부



예제로 쉽게 배우는

## 아두이노

개정판

장성용 · 김진환 지음

새로운 출판

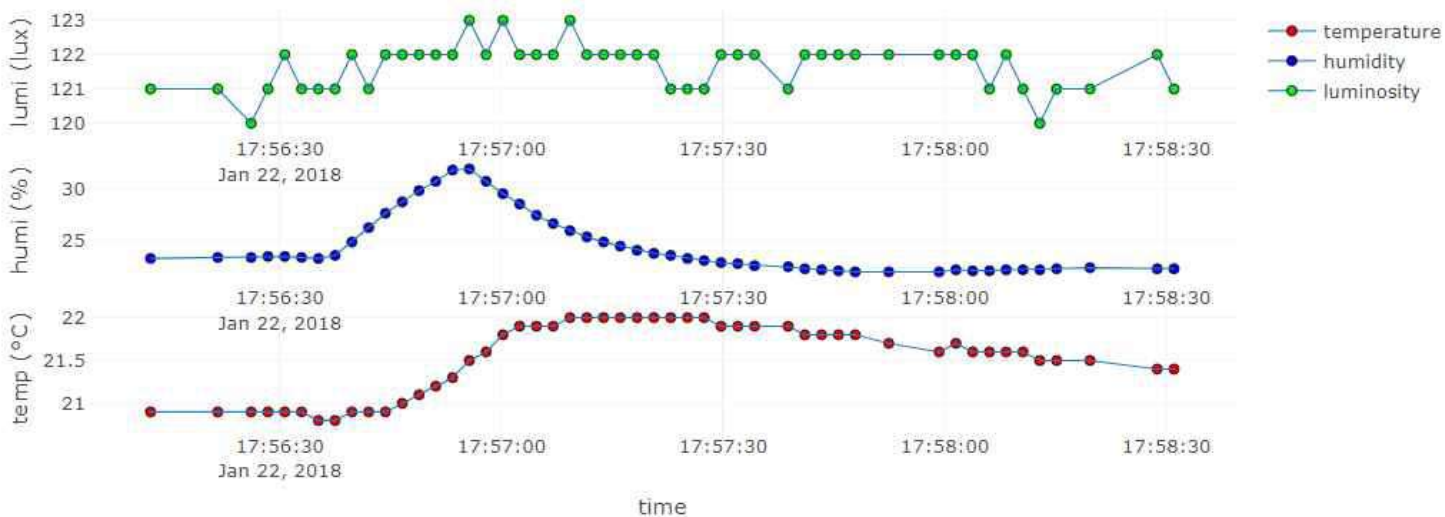


# Target of this class

## Real-time Weather Station from sensors



on Time: 2018-01-22 17:58:31.012





# Another target of this class

PPG with rangeslider

