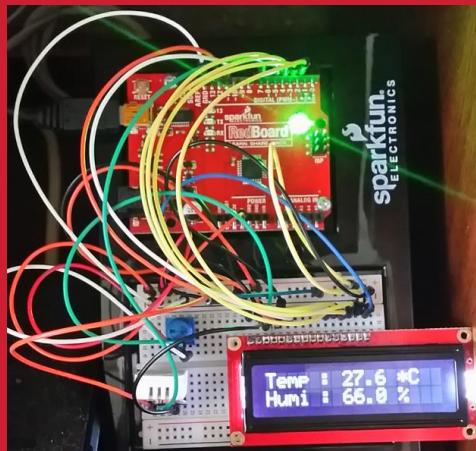


Healthcare-IOT

[wk04]

Arduino Circuit

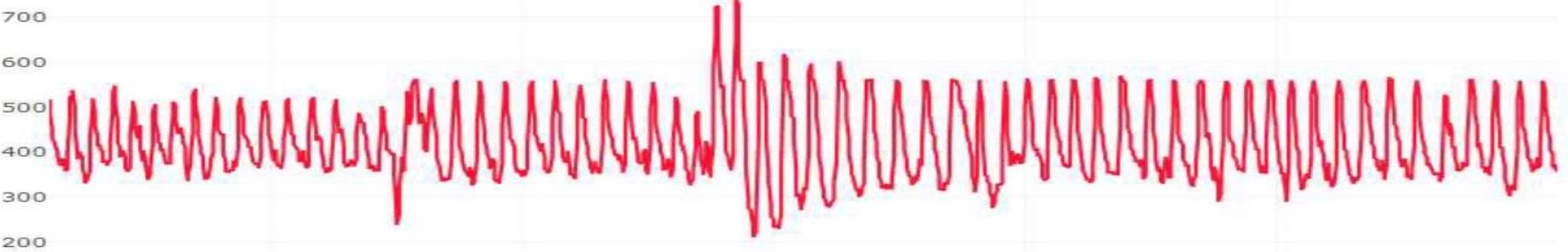


Visualization of Healthcare Signals using
Arduino & Node.js

HCit, INJE University

1st semester, 2018

Email : chaos21c@gmail.com





My ID

오전

성명	ID
김민선	HS01
김영걸	HS02
김주란	HS03
김주현	HS04
김태민	HS05
여준하	HS06
이수민	HS07
정민지	HS08
정유현	HS09
정재은	HS10
주하영	HS11
한준영	HS12

오후

성명	ID
신영주	HS21
오가영	HS22
윤민수	HS23
윤진아	HS24
이진영	HS25
임상은	HS26
임재형	HS27
최민영	HS28
황유빈	HS29



주교재

아두이노와 Node.js에 기반한

IOT 신호 시각화

| 저자 이상훈 |

인제대학교 출판부

아두이노와 Node.js에 기반한 IOT 신호 시각화

| 저자 이상훈 |



인제대학교 출판부



주간계획서

주간계획서			
주차	수업방법	수업내용	과제물
1	강의/실습	수업 및 실습 안내 - 포터블 소프트웨어 설치	
2	강의/실습	Node.js I - Node.js 코드의 기본 구조 - 기초 Node 서버 및 클라이언트	실습확인
3	강의/실습	Node.js II - Node.js Express 서버	실습확인
4	강의/실습/발표	Arduino I - 아날로그 신호 회로 - LCD를 이용한 센서 신호 모니터링	실습확인
5	강의/실습	Arduino II - 단일 센서 회로와 Node.js 연결 - 다중 센서 회로와 Node.js 연결	실습확인
6	강의/실습	프로젝트1 - 생체 센서 회로와 Node.js 연결 - 생체 신호 소개	프로젝트1
7	강의/실습/발표	IOT 데이터 시각화 I (Plotly.js) - 데이터 및 시계열 차트 - 데이터 스트리밍	실습확인
8	시험	중간고사	
9	강의/실습	IOT 데이터 시각화 II (Plotly.js) - 다중 센서 데이터 시각화 - 다중 센서 데이터 스트리밍	실습확인
10	강의/실습/발표	프로젝트II - 생체 센서 데이터 시각화 - 생체 센서 데이터 스트리밍	프로젝트II
11	강의/실습	IOT 데이터 저장과 처리 - MongoDB 설치 및 Mongo shell - MongoDB와 Node.js 연결 및 데이터 저장	실습확인
12	강의/실습	프로젝트III - MongoDB에 IOT 데이터 저장 및 모니터링 - 생체 센서 데이터 저장 및 시각화	프로젝트III
13	강의/실습	IOT 데이터 마이닝 - 아두이노에서 발생된 데이터 관리 - 데이터마이닝 소개	실습확인
14	강의/실습/발표	프로젝트IV - 생체 센서 데이터 관리 - 생체 센서 데이터 마이닝	프로젝트IV
15	시험	기말고사	



Purpose of HS

주요 수업 목표는 다음과 같다.

1. Node.js를 이용한 아두이노 센서 신호 처리
2. Plotly.js를 이용한 아두이노 센서 신호 시각화
3. MongoDB에 아두이노 센서 데이터 저장 및 처리
4. 생체 센서 발생 신호 처리, 시각화 및 저장
5. 생체 센서 발생 신호 저장 및 분석
6. 생체 신호 장비 활용 능력





[Review]

◆ [wk03]

- Express App (7.2 Express project)
- Add a new route in `routes/index.js`
- Add a new view: `views/hsnn.jade`
- Upload file name : `HSnn_Rpt02.zip`



[Practice-1] Modify routes/index.js

Add a new route '/hsnn' in routes/index.js .

FOLDERS

- ▶ hs00
- ▼ hs00
 - ▶ express
 - ▶ bin
 - ▶ node_modules
 - ▶ public
 - ▼ routes
 - /* index.js
 - /* users.js
- ▶ views
- /* app.js
- /* package.json
- ▶ expressTest
- ▶ hs00App
- ▶ myApp
- ▼ server
 - ▶ File
 - ▶ HTTP
 - ▶ TCP
- ▶ start

```
index.js
```

```
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 /hsnn. -> multi-routing */
10 router.get('/hsnn', function(req, res, next) {
11   res.render('hsnn', { title: 'Express App',
12                      id: 'HS00',
13                      name: 'Redwoods' });
14   // views/hsnn.jade
15 });
16
17 module.exports = router;
```



[Practice-2] Add hsnn.jade in views folder

Add views/hsnn.jade in views folder

The screenshot shows a code editor interface with a sidebar containing a file tree. The file tree includes folders like 'hs00', 'express', 'views', and files like 'index.jade', 'layout.jade', 'error.jade', and 'hs00.jade'. The 'hs00.jade' file is open in the main editor area, displaying the following Jade template:

```
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}
```

Below the editor is a browser window titled 'Express App' with the URL 'localhost:3000/hs00'. The browser displays the rendered output of the Jade template, which includes:

Sangs Heon

Welcome to Express App

My id : HS00

Developed by Redwoods

A yellow box highlights the rendered output in the browser, and the text 'Savs as HSnn_Jade.png' is overlaid on the right side of the browser window.

wk03 : Practice-02 : HSnn_Rpt02.zip

◆ [Target of this week]

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

제출파일명 : **HSnn_Rpt02.zip**

- 압축할 파일들

- ① **HSnn_Express.png**
- ② **HSnn_naver.png**
- ③ **HSnn_Express_App.png**
- ④ **HSnn_Jade.png**

Email : **chaos21c@gmail.com**

【 제목 : id, 이름 (수정) 】



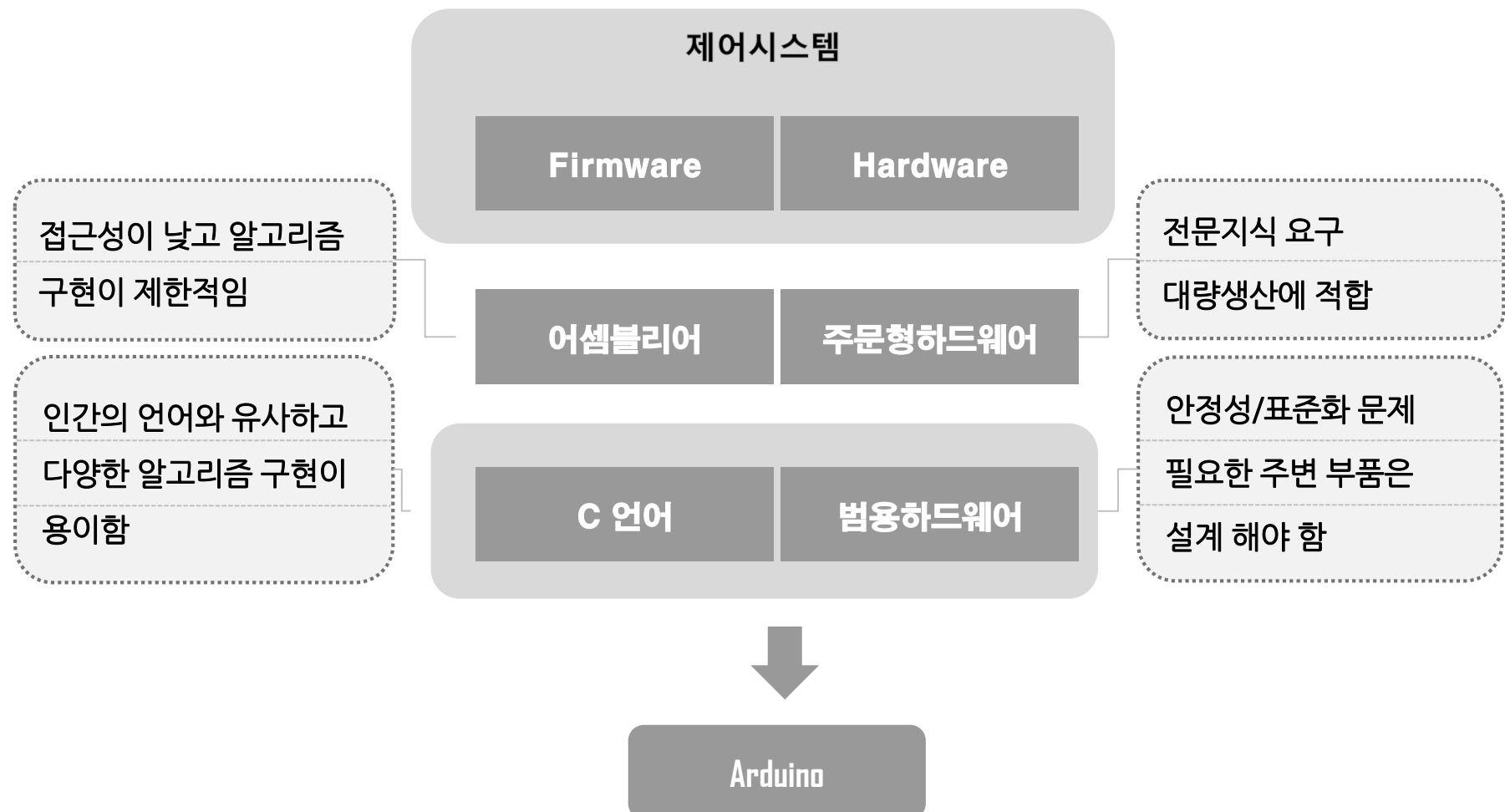
Arduino

[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 Cuartielles에 의해 개발

예술가
취미생활
학생

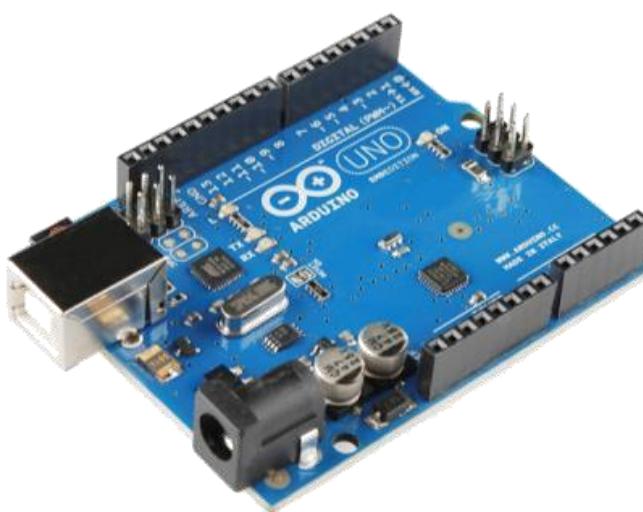
전자공학
교육

누구나
쉽게
사용
가능한
제어장치

오픈소스
하드웨어

GSM Wifi Ethernet
Motor drive
등의 월드 제공

다양한
라이브러리



LabView

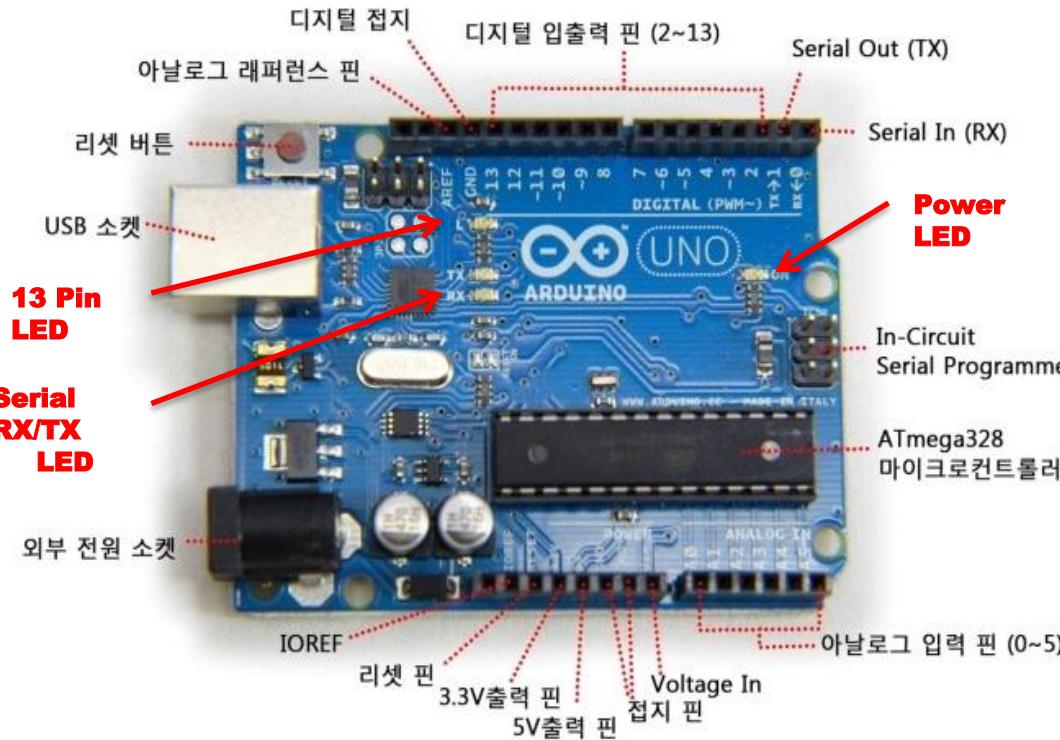
MATLAB

Node.js
Plot.ly

Mongo DB

범용
하드웨어
IoT 의
표준

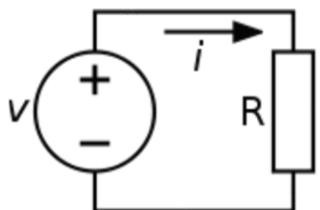
0.2 Arduino hardware



✓ 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

0.3 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

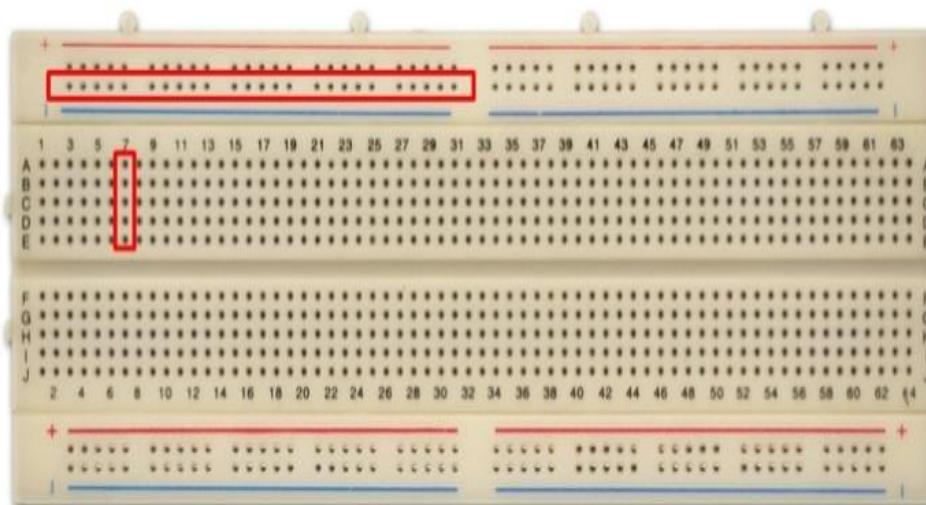
저항
[Ω]

- ✓ 전류의 흐름을 방해하는 정도를 나타냄

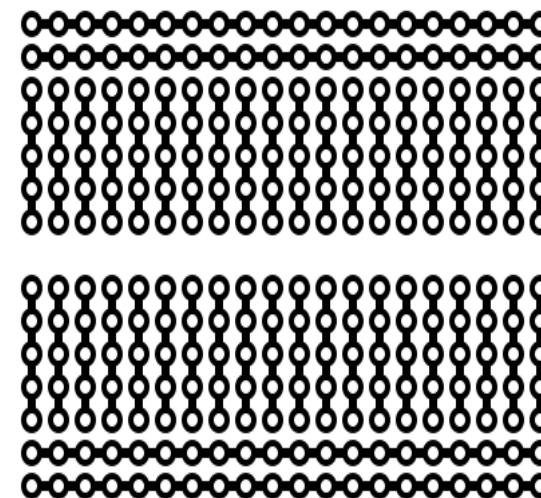
- ✓ 색 띠나 숫자로 값을 표시
- ✓ Arduino에서는 칩 (chip) 형태의 저항이 사용

0.4 브레드 보드 (Bread board)

시제품 제작이나 실험용 와이어를 보드에 꽂아 사용



빨간색 묶음 흘끼리 내부회로가 연결되어 있음



내부 결선



Arduino SW

[fritzing.org Fritzing](http://fritzing.org/home/)

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

The Fritzing website homepage features a large "fritzing" logo with the tagline "electronics made easy". Below the logo is a navigation bar with links: Projects, Parts, Download, Learning, Services, Contribute, FORUM, and FAB. A red banner below the navigation bar contains text about the software being open source and community-supported.

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](#)

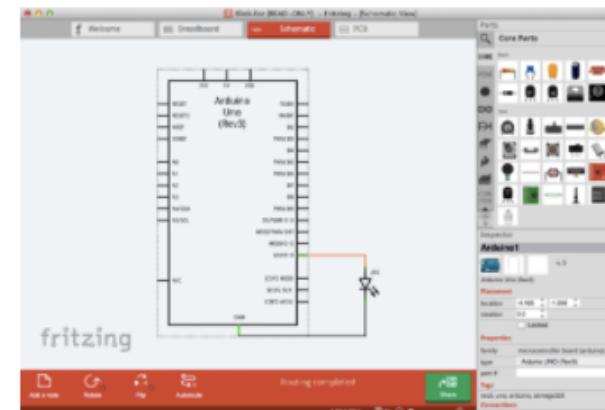
[Mac OS X 10.7 and up](#)

[Linux 32 bit](#)

[Linux 64 bit](#)

[Source Github](#)

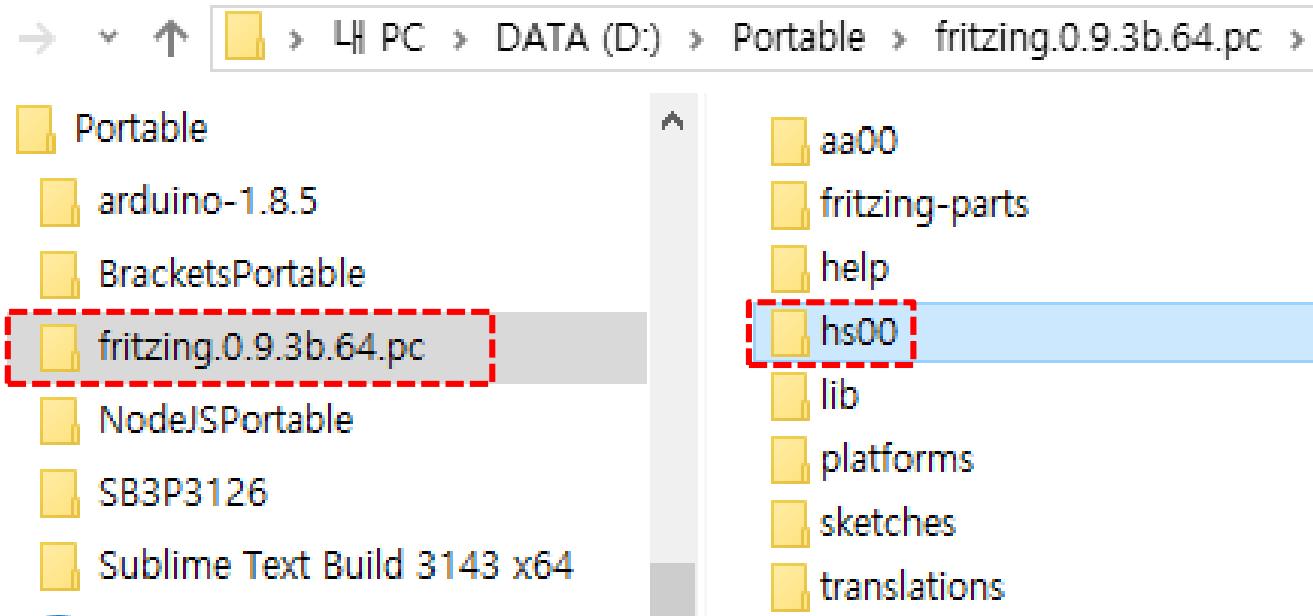
Downloaded 2578877 times.





Fritzing configuration: working folder

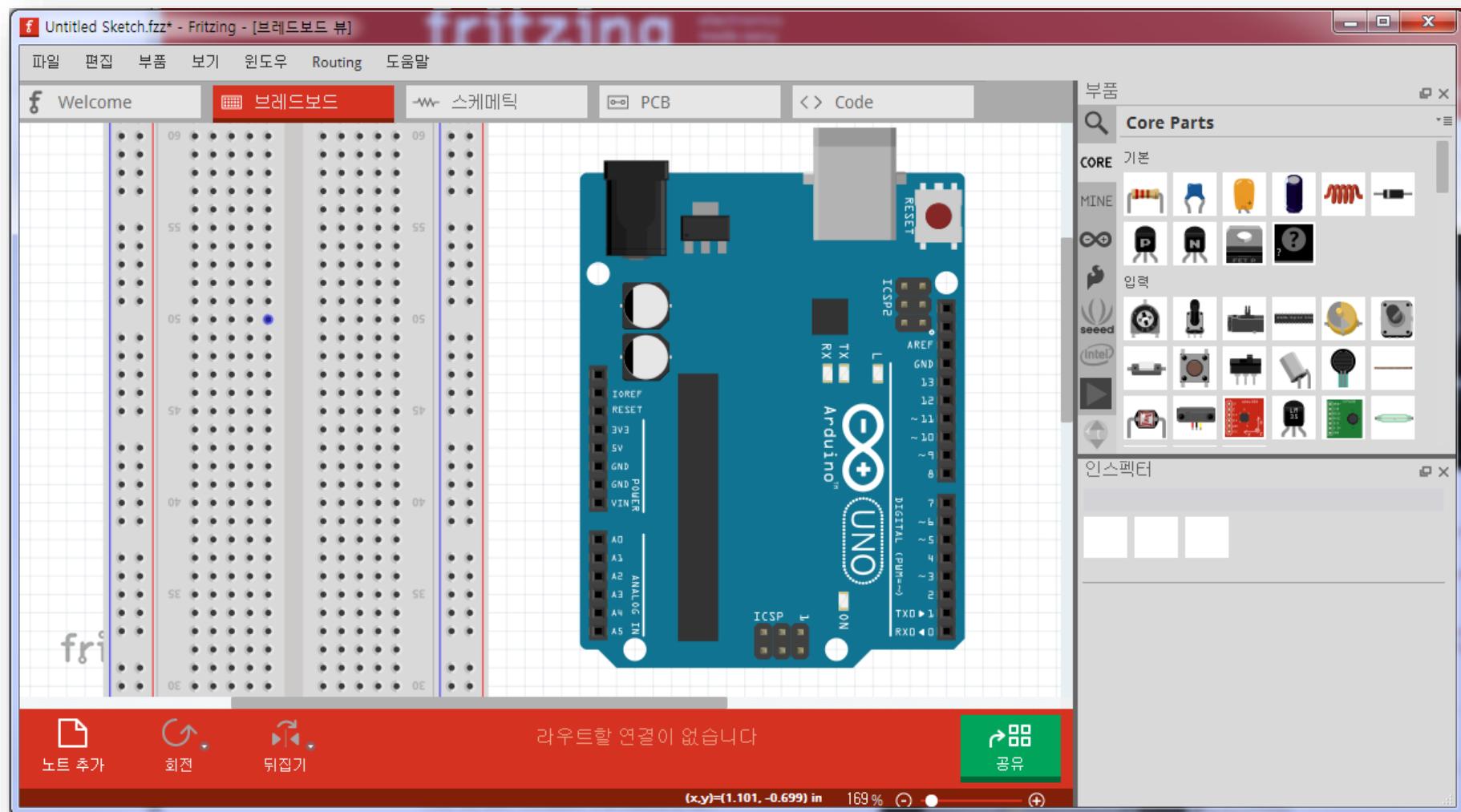
[Fritzing] configuration





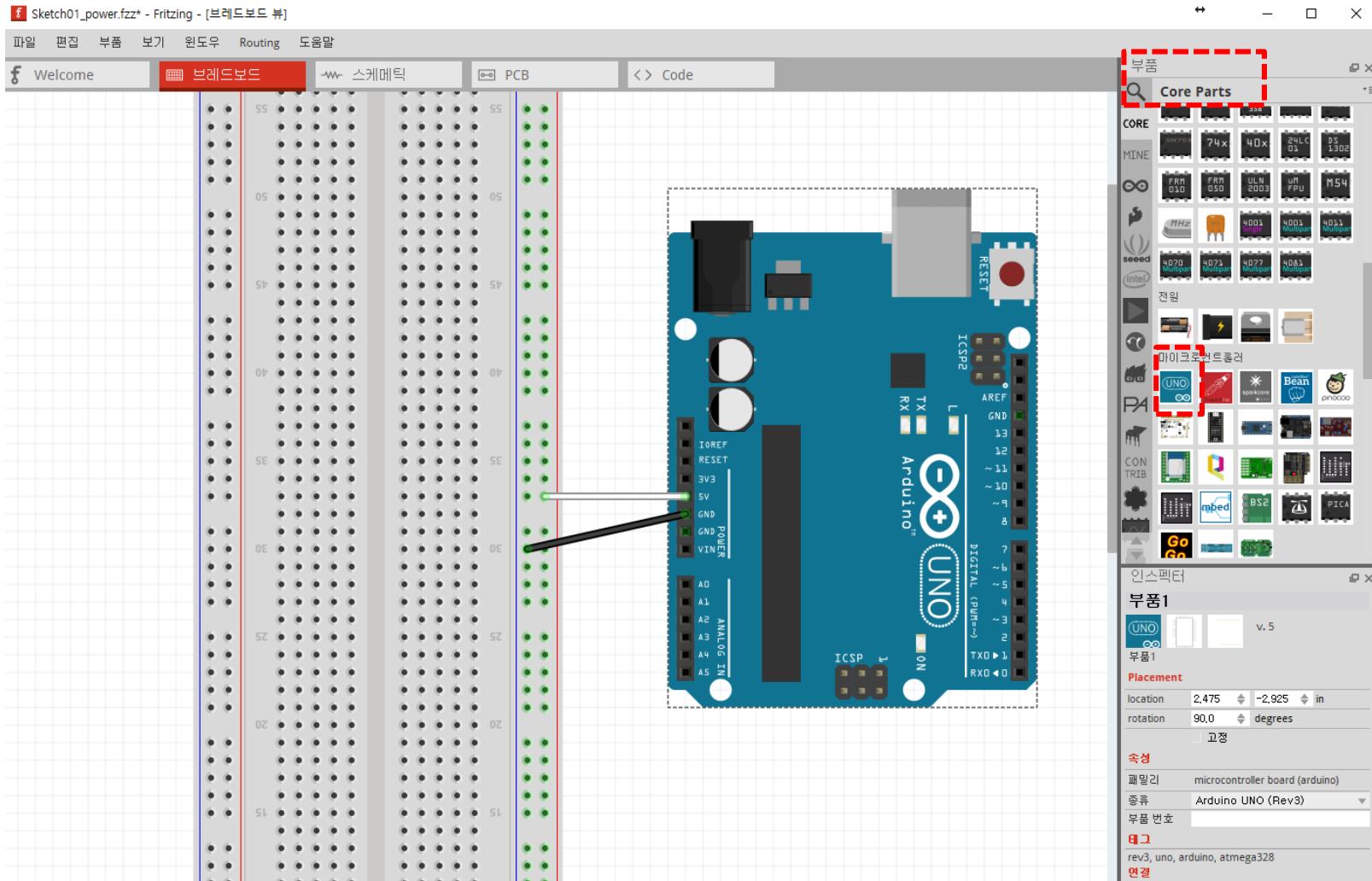
Fritzing configuration

[Fritzing] configuration





Fritzing configuration - power

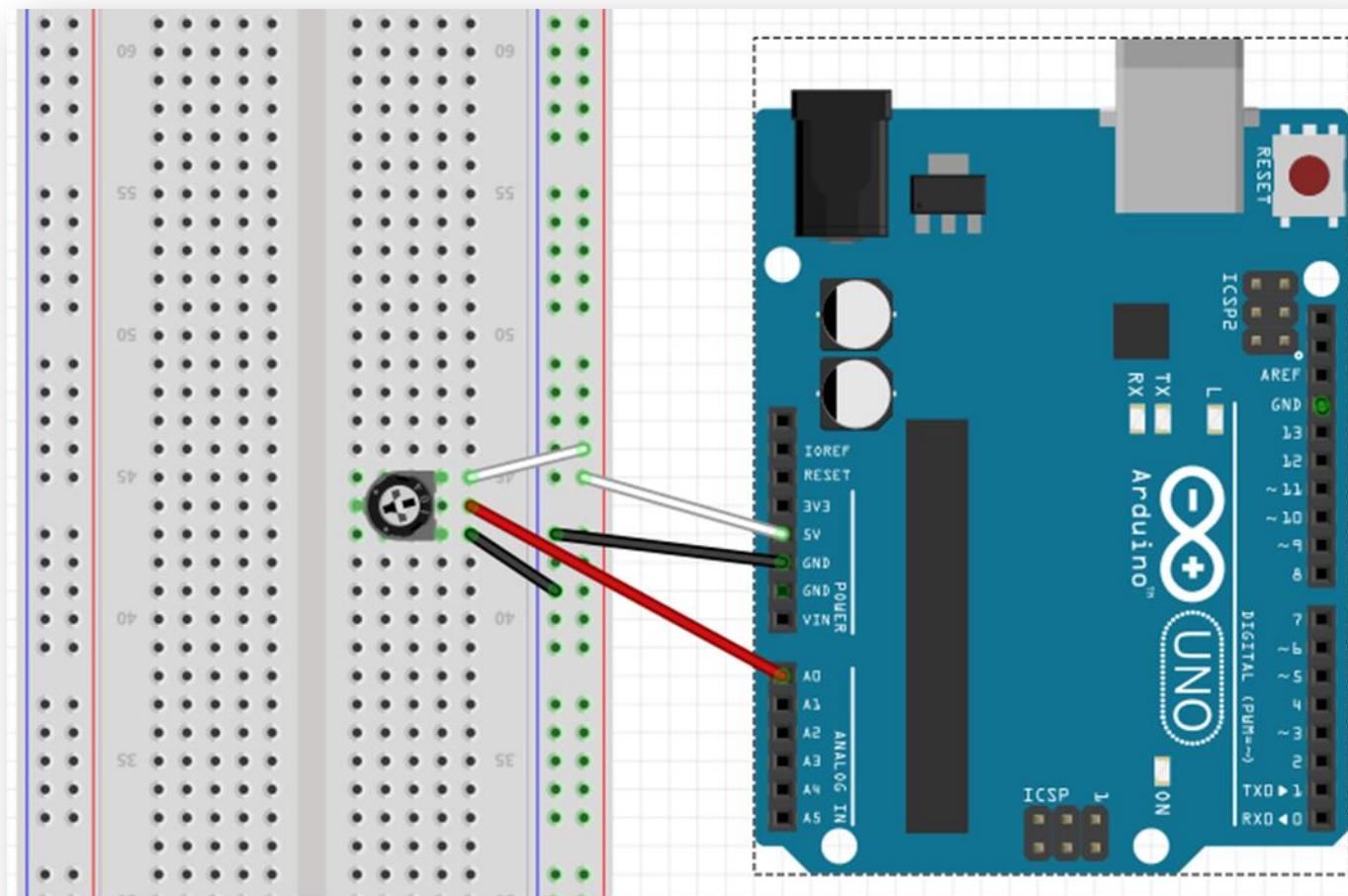




Arduino circuits



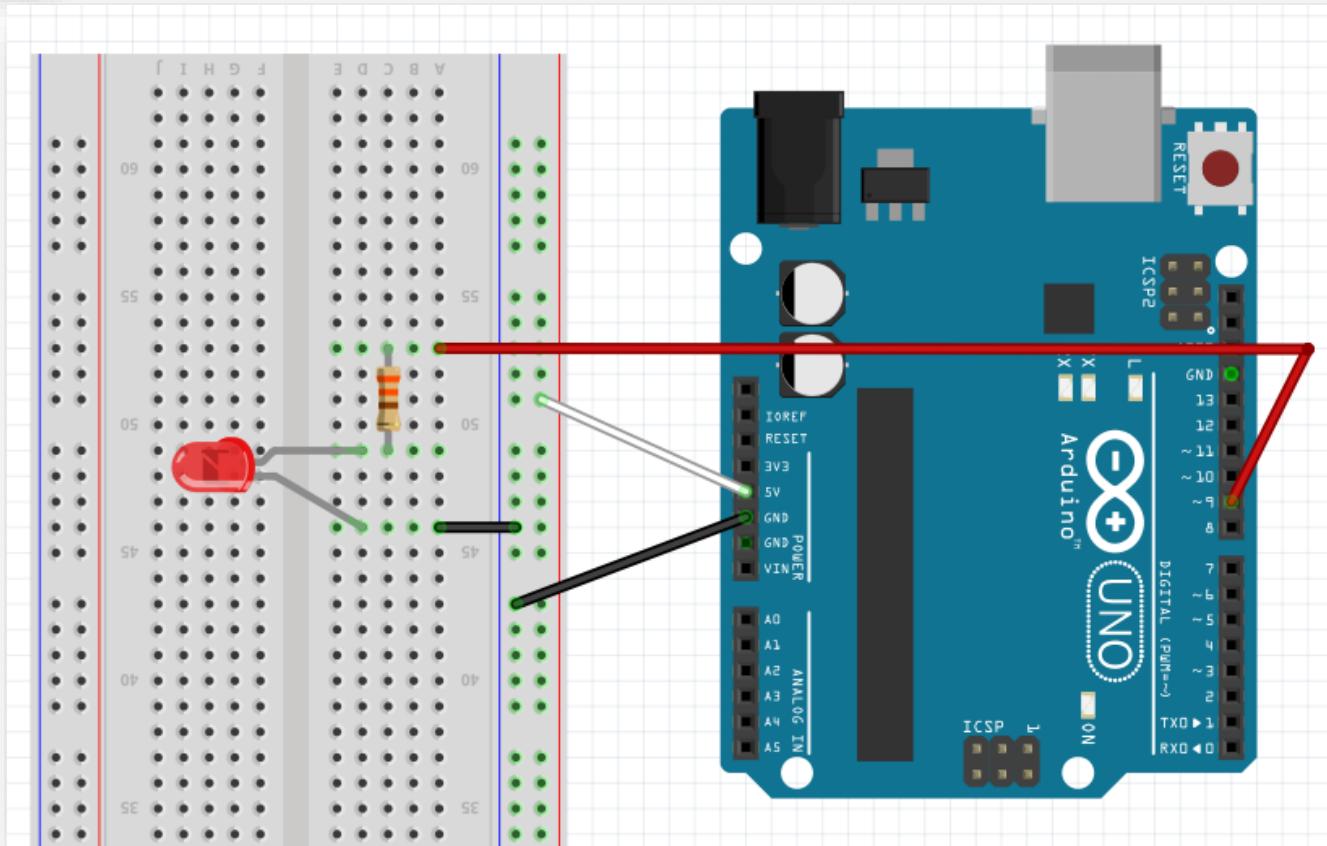
0.A1 Potentiometer (가변 저항기)



Parts : 가변저항기



0.A2 single LED



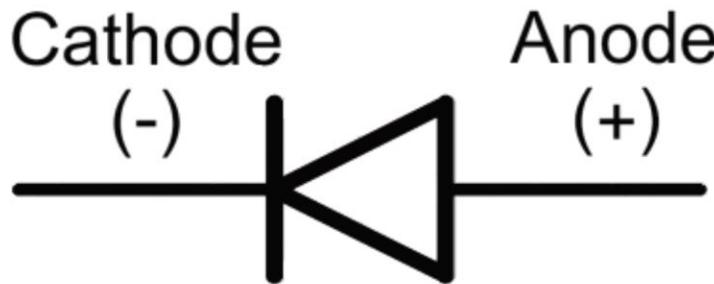
Parts : LED (1), R ($330 \Omega \times 1$)



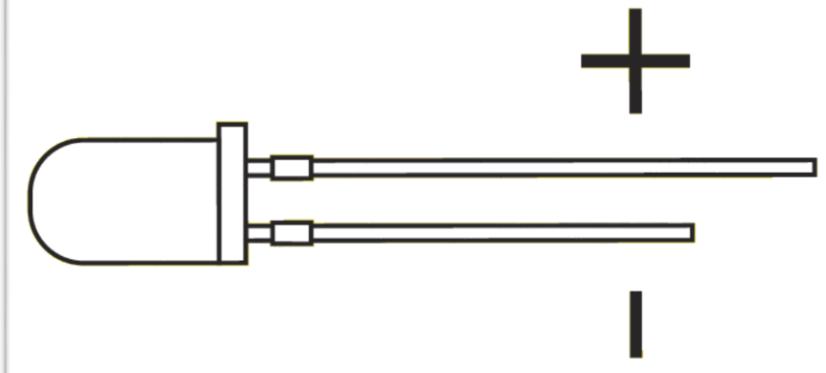


0.A2 single LED

Polarity of Diode and LED



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

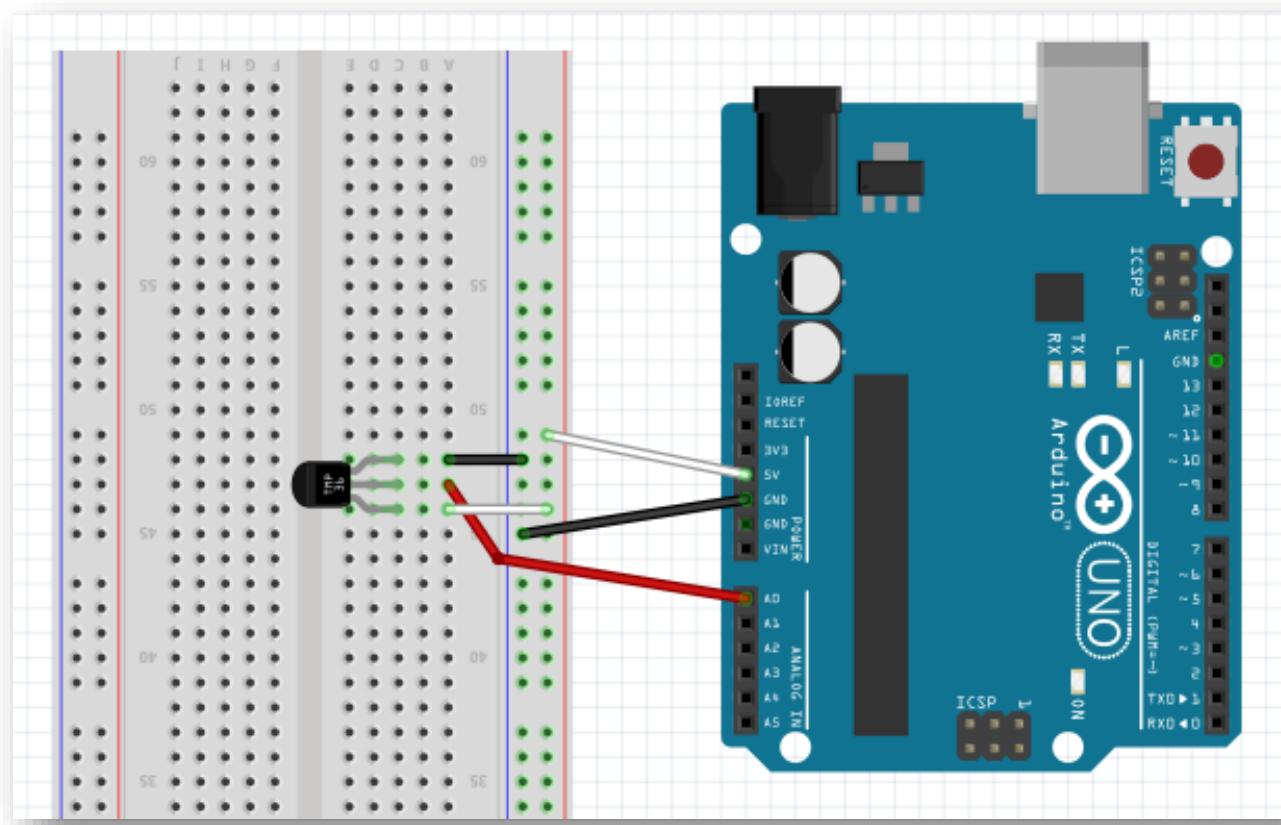


Find the longer leg, which should indicate the positive, anode pin.

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



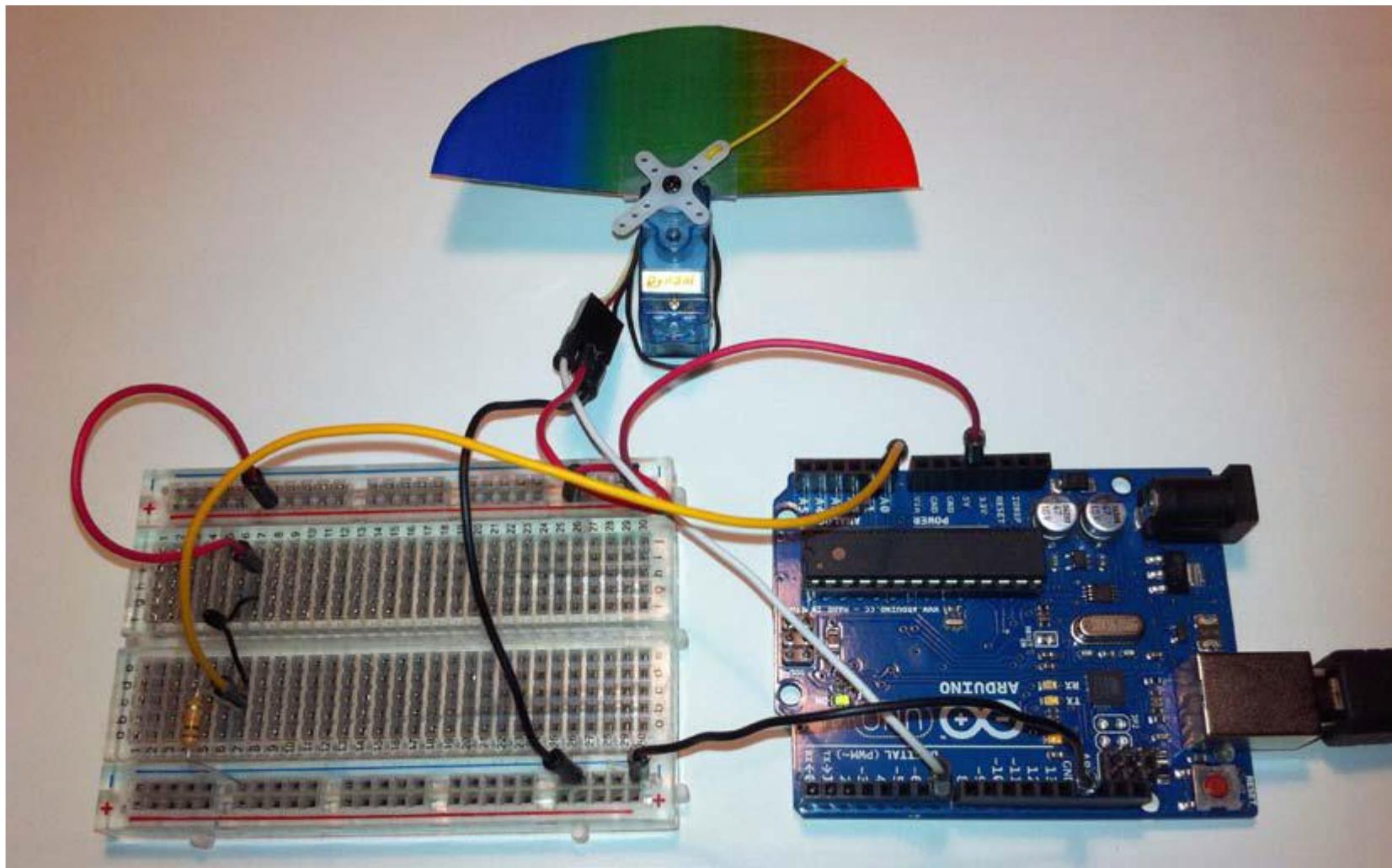
0.A3 Temperature sensor (TMP36)



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

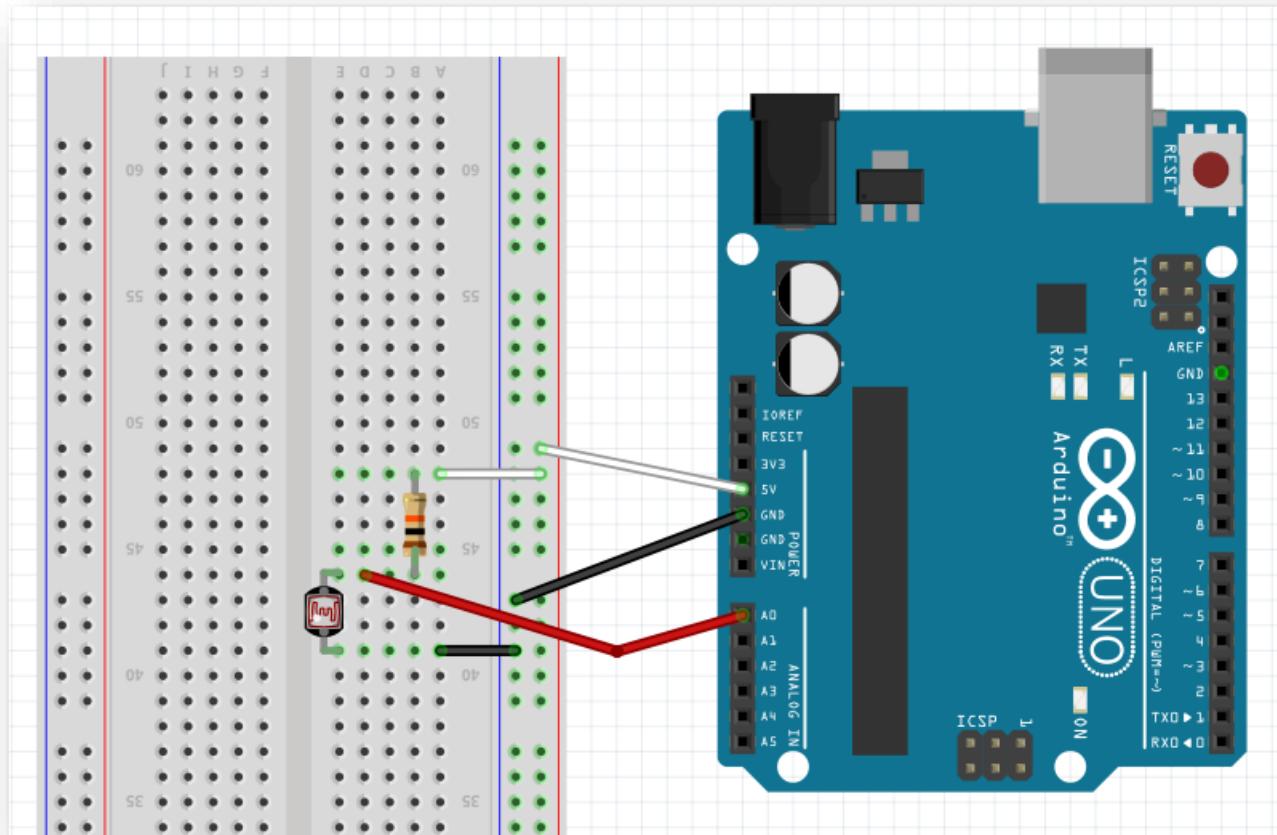


0.A3. DIY3 Servo





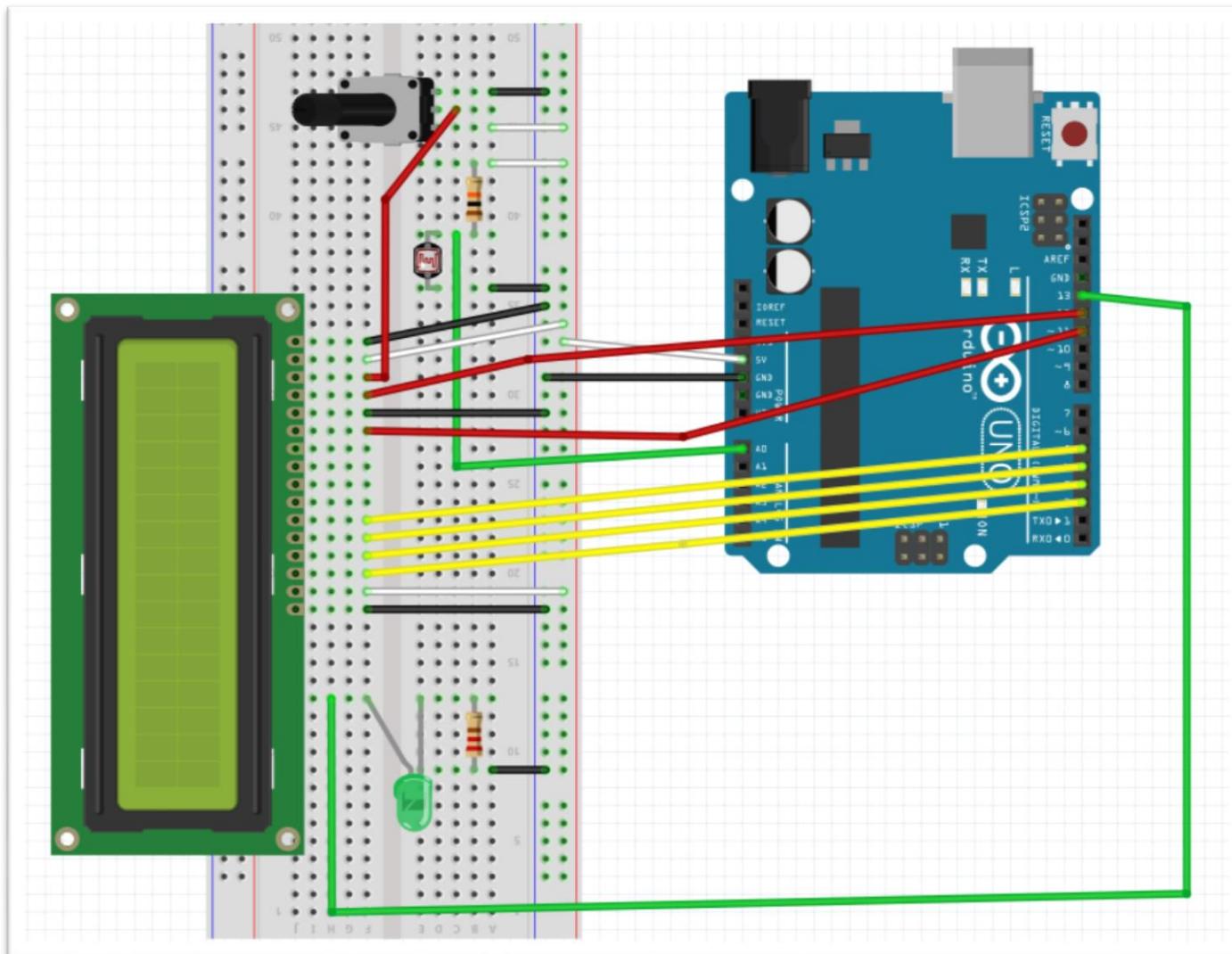
0.A4 Luminosity sensor : photo cell LDR



Parts : 20 mm photocell LDR, R (10 kΩ X 1)

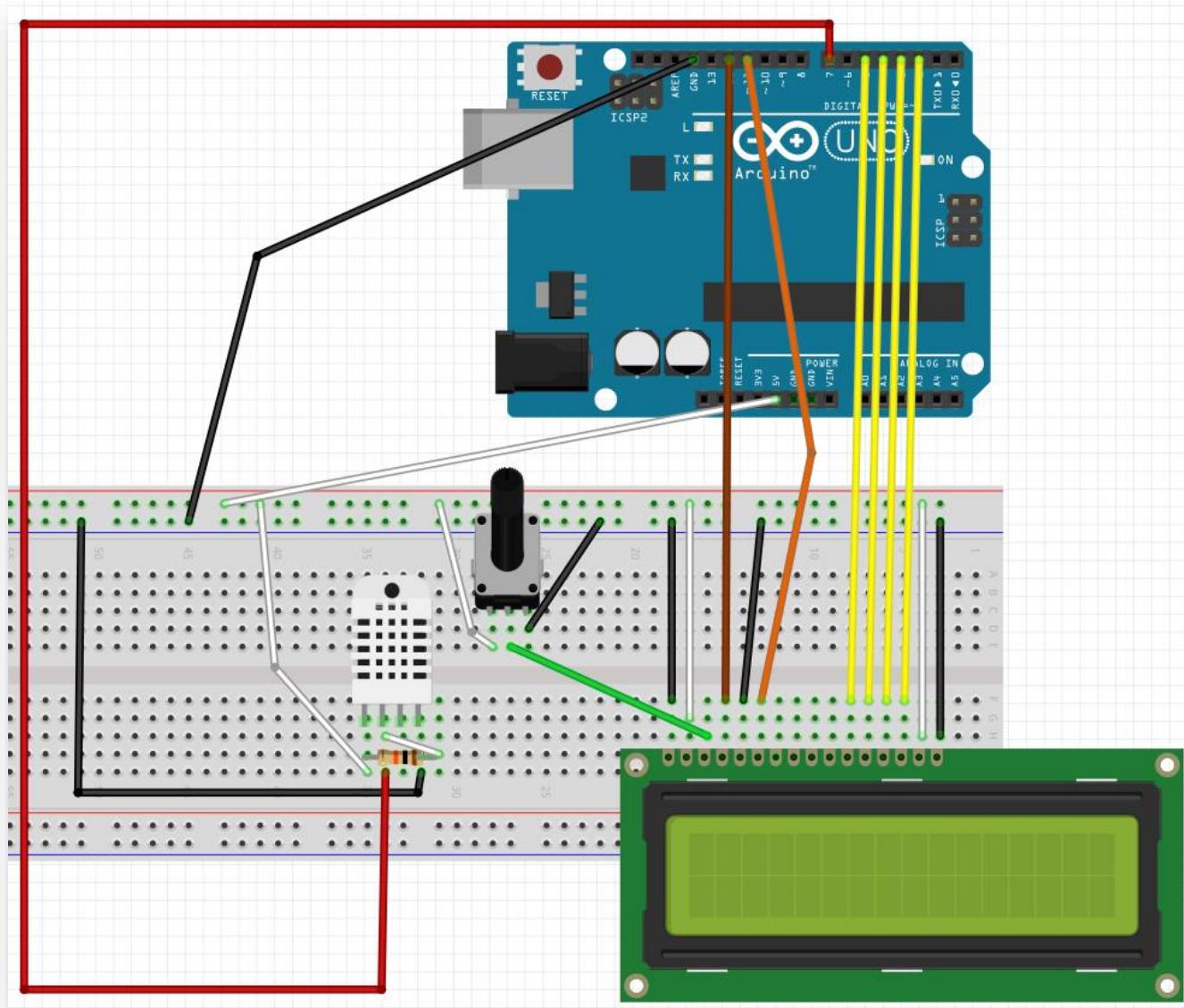


0.A5 Display of luminosity



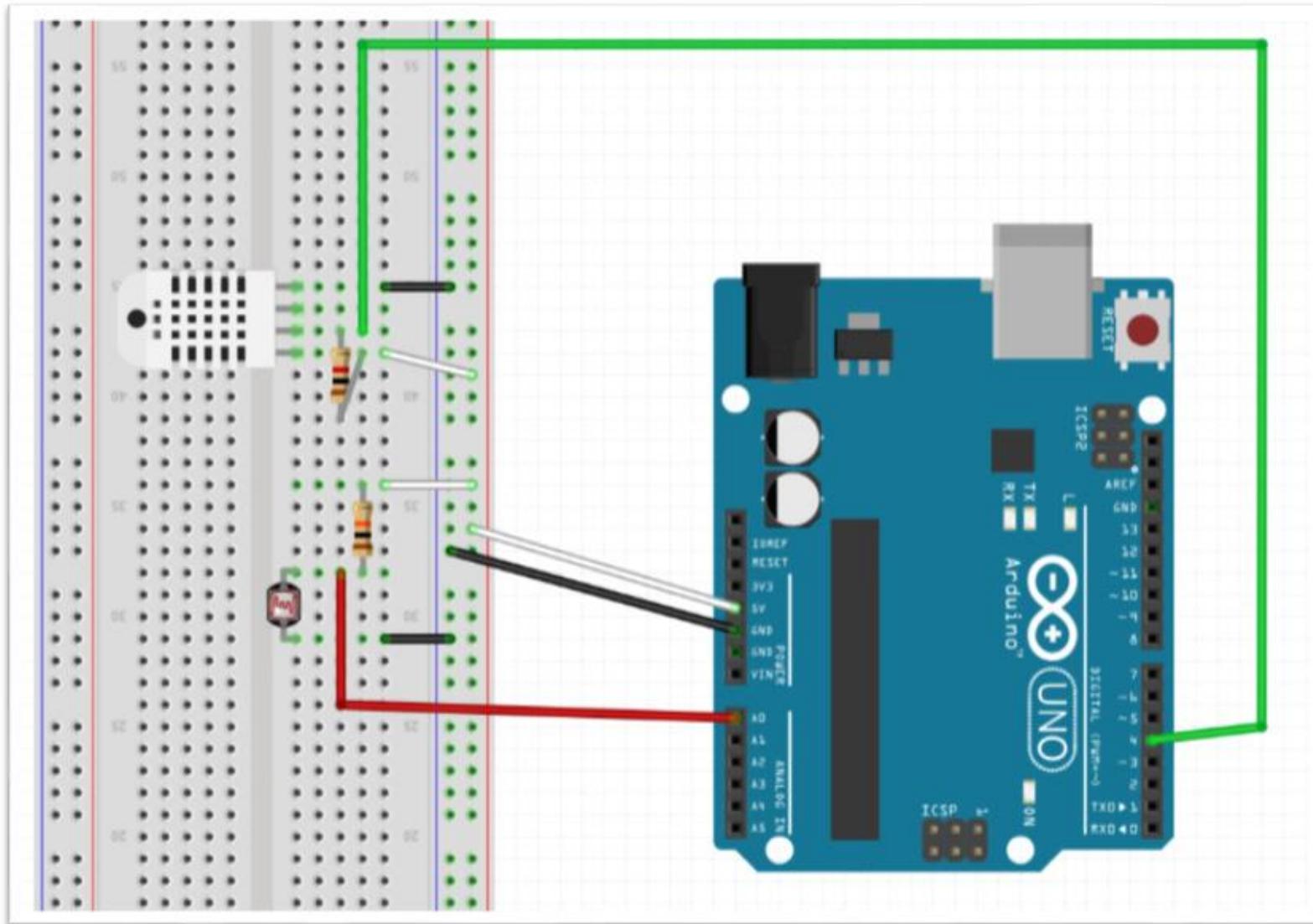


0.A6 Display of Temperature & Humidity



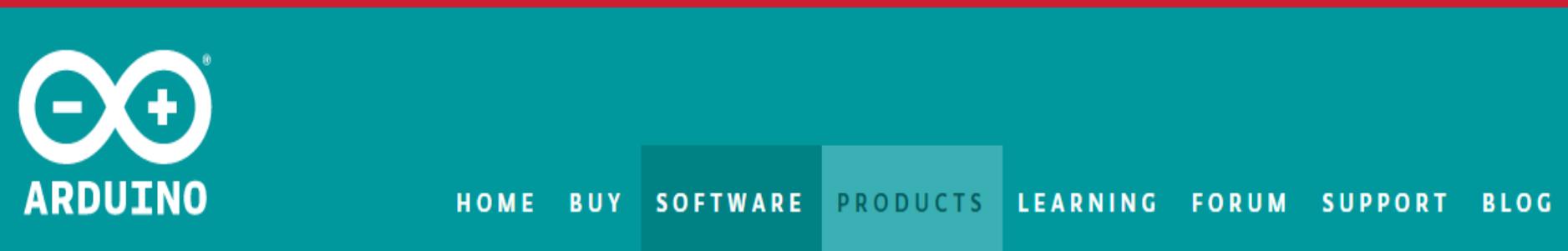


0.A7 DHT22 & CdS





1. Arduino SW: IDE



The screenshot shows the top navigation bar of the Arduino website. On the left is the Arduino logo. To its right are several menu items: HOME, BUY, SOFTWARE (which is highlighted in a darker shade), PRODUCTS, LEARNING, FORUM, SUPPORT, and BLOG. The background of the bar is a gradient of teal colors.

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



A1.1 Arduino IDE – portable ver.

The screenshot shows the Arduino website's header. On the left is the Arduino logo. To its right are links for HOME, BUY, SOFTWARE (which is highlighted in a blue box), PRODUCTS, EDUCATION, RESOURCES, and COMMUNITY. Below the main menu is a secondary navigation bar with links for ONLINE TOOLS and DOWNLOADS. The DOWNLOADS link is also highlighted in a blue box. To the right of the menu are two colored bars: a teal one labeled 'IDE' and an orange one labeled 'PORTABLE'.

Download the Arduino IDE

The screenshot shows the Arduino IDE download page. On the left, there is a large teal circle containing the Arduino logo. To its right, the text "ARDUINO 1.8.5" is displayed in bold. Below this, a paragraph explains what the Arduino Software (IDE) is and the platforms it supports. It also mentions that the software can be used with any Arduino board and refers to the "Getting Started" page for installation instructions. On the right side of the page, there are download links for Windows (Installer and ZIP file), Mac OS X, Linux 32 bits, Linux 64 bits, Linux ARM, and links for Release Notes, Source Code, and Checksums (sha512). The Windows download links are enclosed in a red dashed box.

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

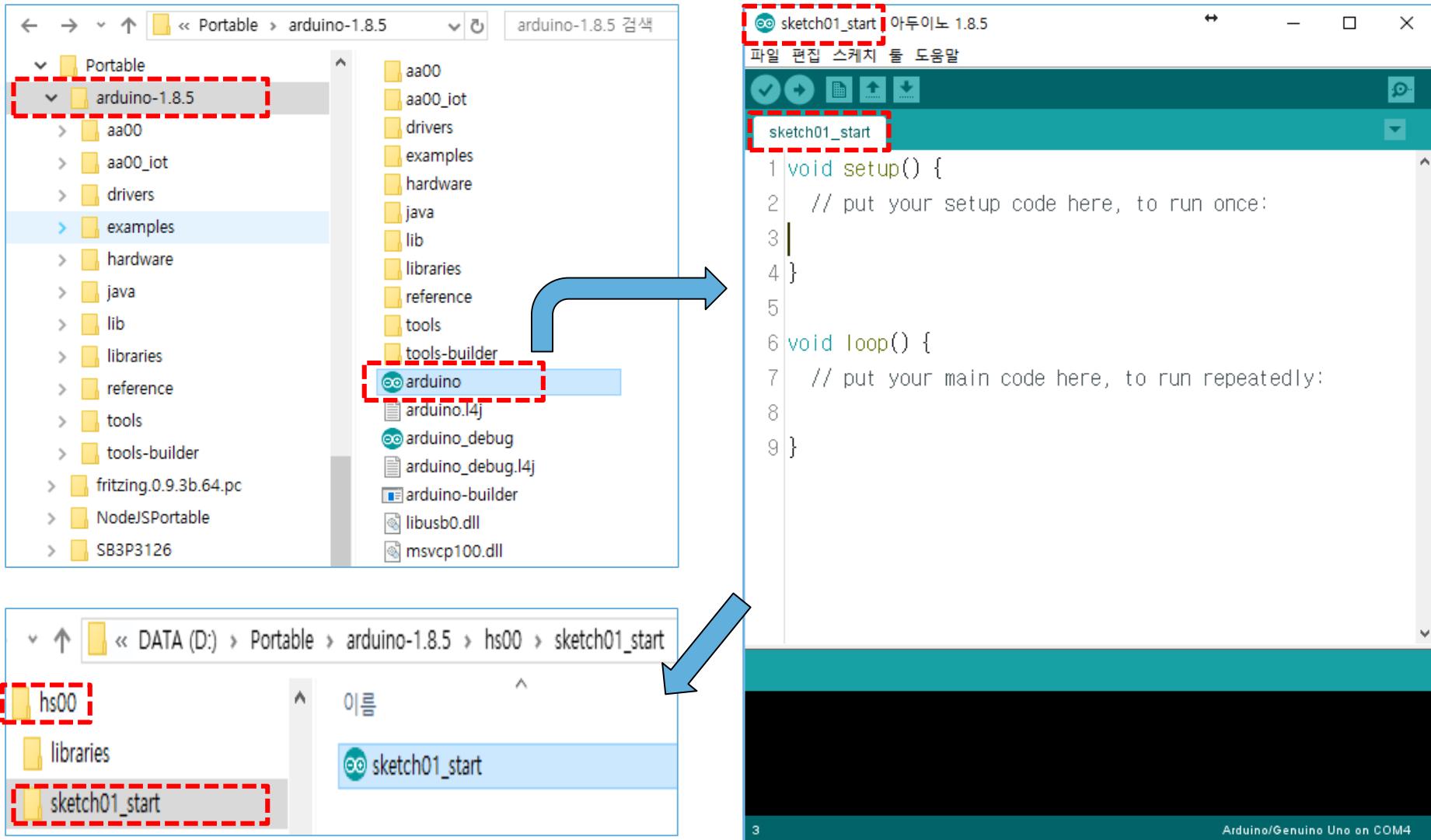
Linux 32 bits
Linux 64 bits
Linux ARM

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



A1.2 Arduino Portable (V1.8.5)

Make folder hsnn in portable Arduino folder





A1.3 Arduino Portable IDE

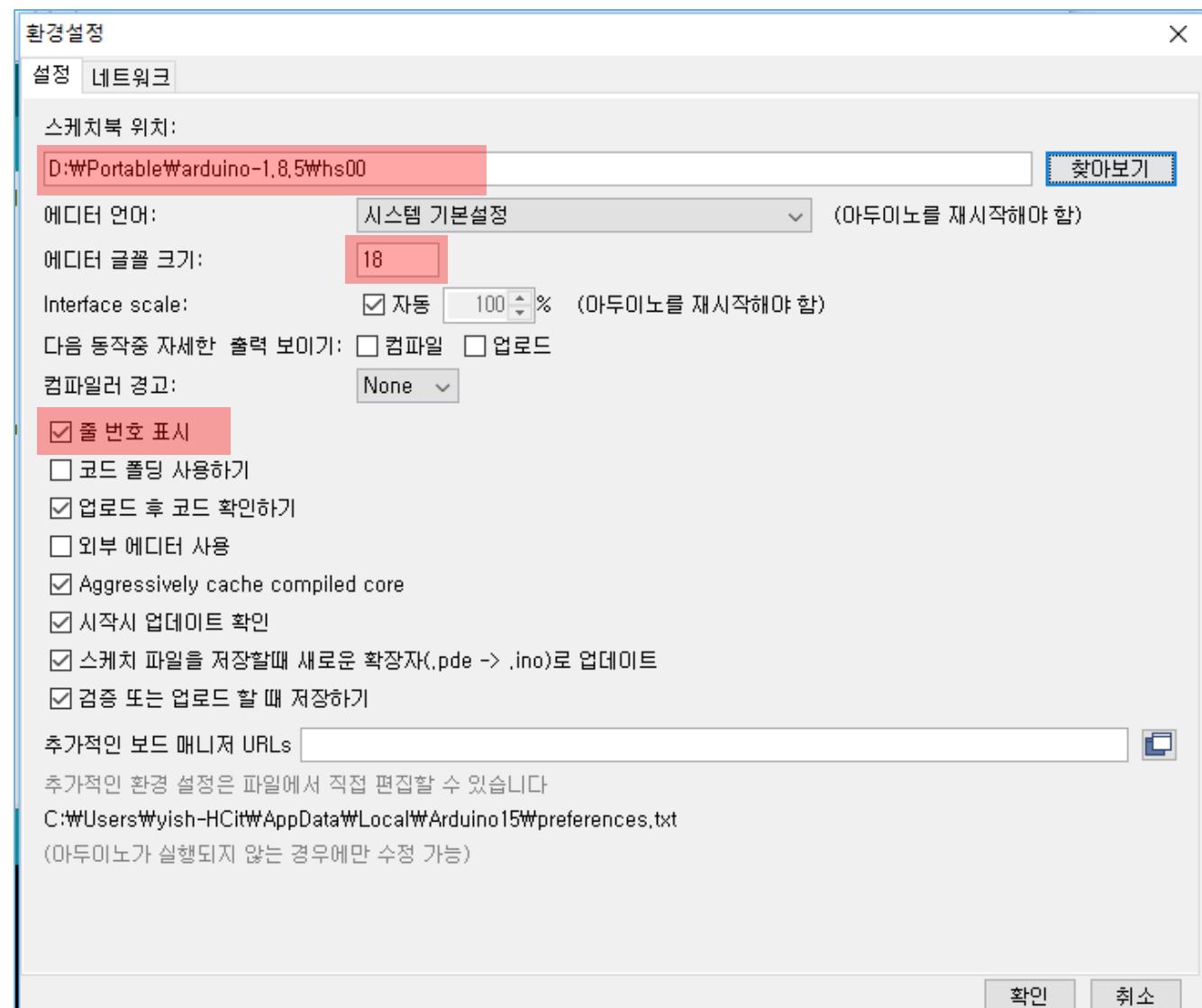
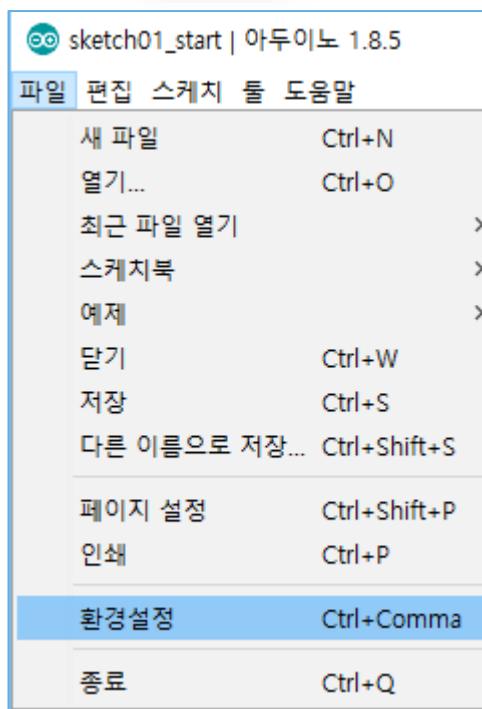
The screenshot shows the Arduino Portable IDE interface. The title bar reads "sketch01_start | 아두이노 1.8.5". The menu bar includes "파일 편집 스케치 들 도움말". The toolbar has icons for file operations like Open, Save, and Upload. The code editor window contains the following sketch:

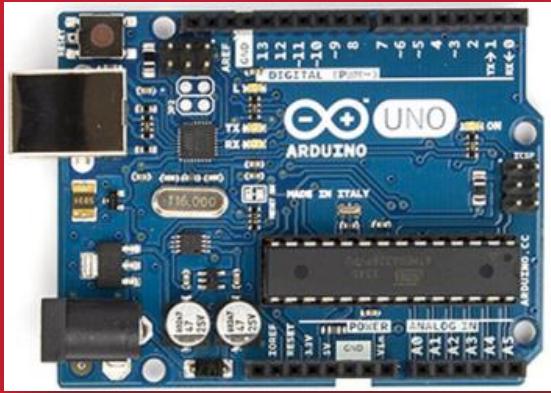
```
1 void setup() {  
2     // put your setup code here, to run once:  
3  
4 }  
5  
6 void loop() {  
7     // put your main code here, to run repeatedly:  
8  
9 }
```

The first code block (setup) is highlighted with a red dashed border and a pink background. The second code block (loop) is also highlighted with a red dashed border and a yellow background. The status bar at the bottom right shows "Arduino/Genuino Uno on COM4".



A1.4 Arduino Portable IDE





LED



A2.0 LED control

LED (Light Emitting Diode)

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

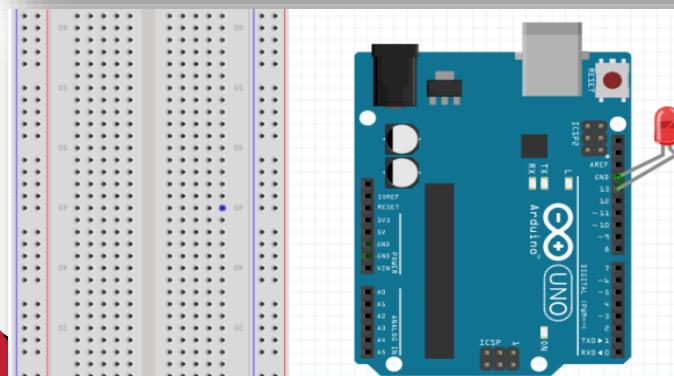




A2.1.1 Blink [digitalWrite()]

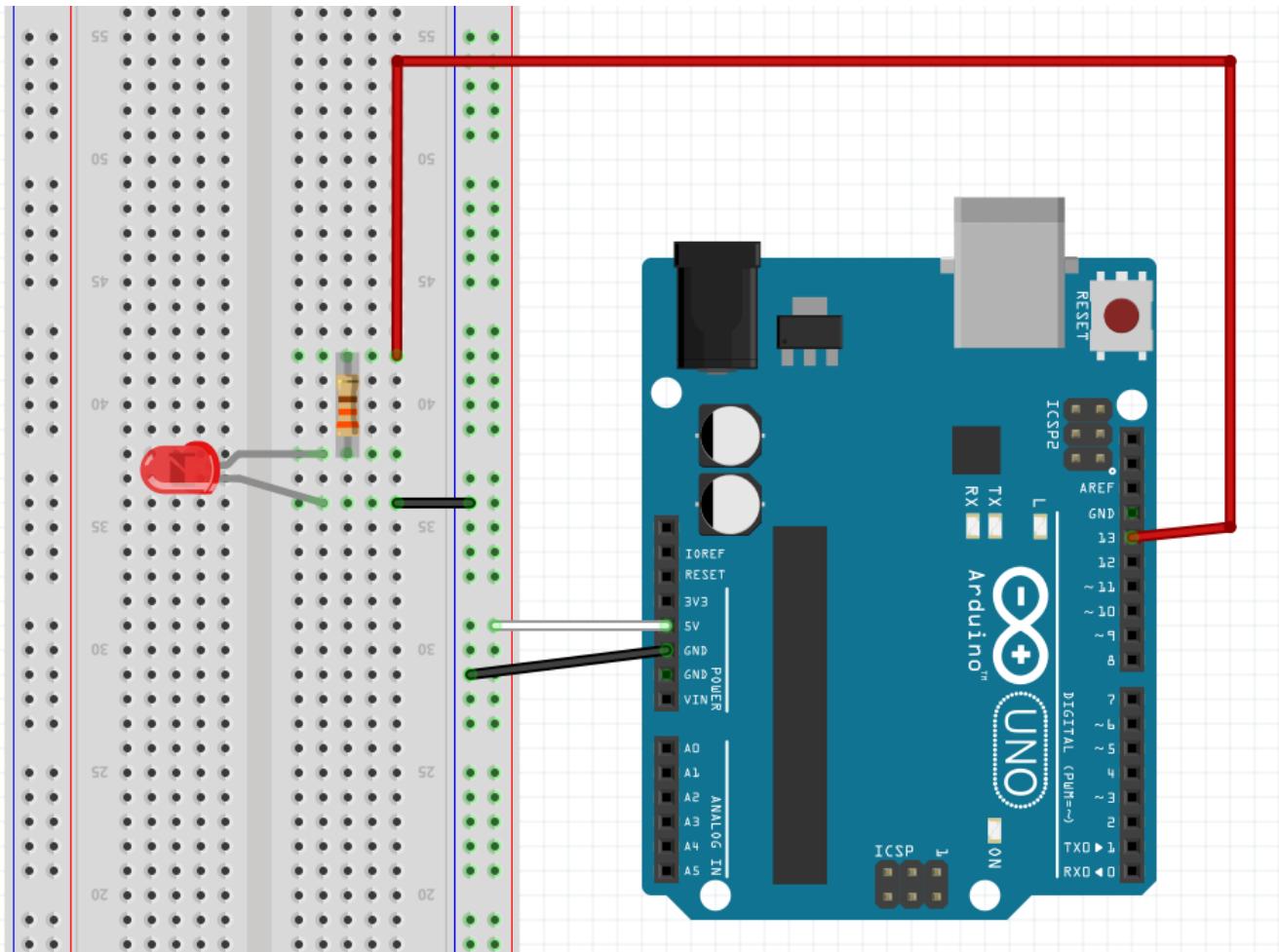
```
Blink §

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 }
```





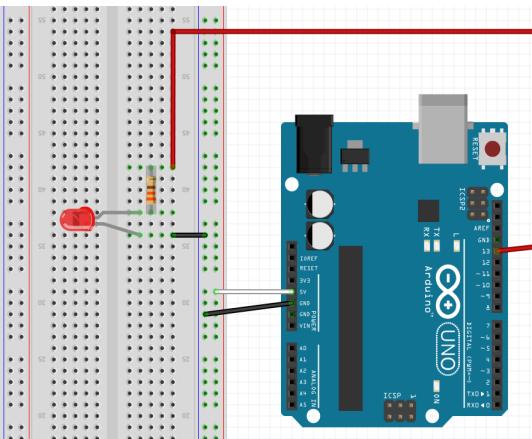
A2.1.2 blink circuit



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



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



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

hs00
libraries
sketch01_blink
sketch01_start

이름
sketch01_blink

```
sketch01_blink
1 /*
2  * Blink by HS00
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 }
```

Save as
HSnn_Blink.png



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 지원한다.

A2.2.2 LED control – 밝기 조절: PWM

PWM (Pulse Width Modulation)

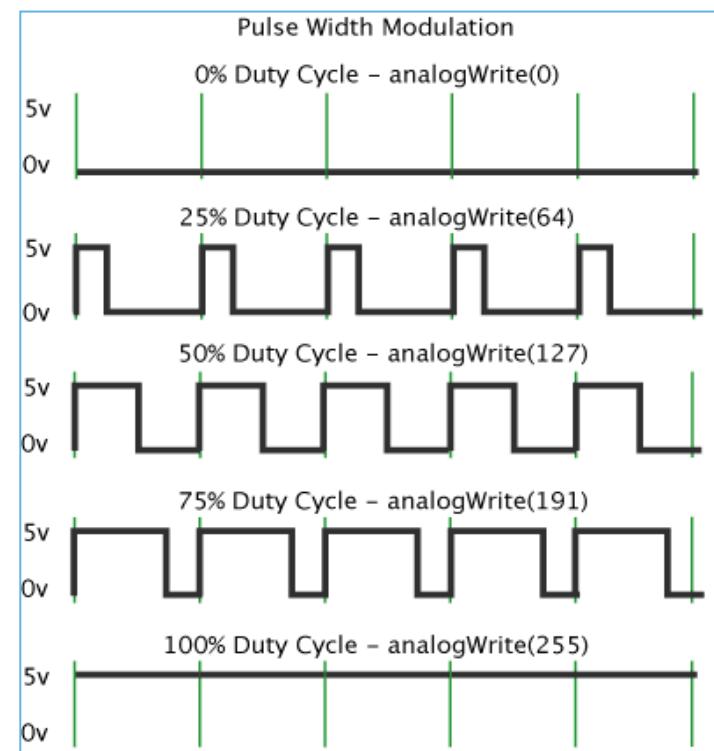
Using [analogWrite\(pin, pwm_value\)](#) function in fading an LED off and on.

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.

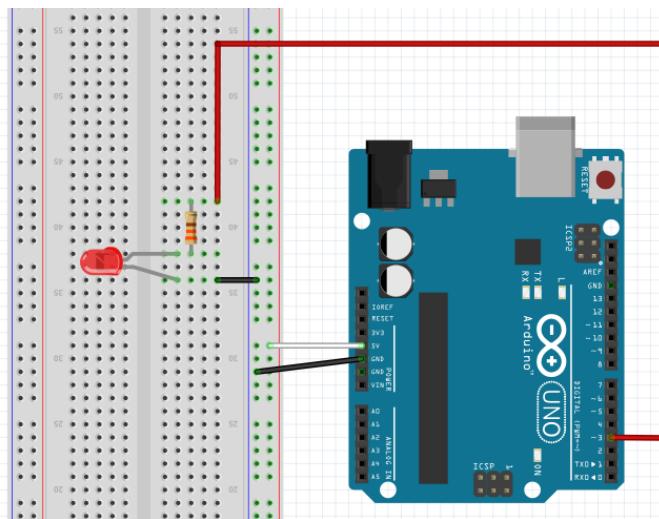
A call to [analogWrite\(\)](#) is on a scale of **0 - 255**, such that analogWrite(255) requests a 100% duty cycle (always on), and analogWrite(127) is a 50% duty cycle (on half the time)

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

LED의 밝기가
0.01초 간격으로 마구
잡이로 변하는 것을 확인



신호 발생 및 모니터링

Serial monitor
&
plotter

A2.3 시리얼 통신 (serial comm.)

시리얼 통신

UART (Universal Asynchronous
Receiver/Transmitter)

RS-232

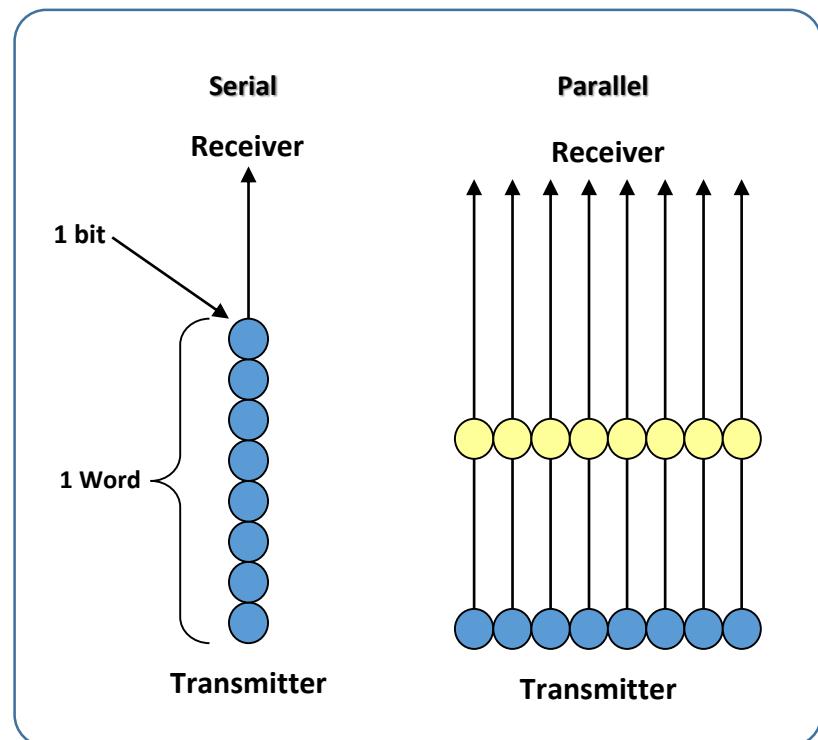
RS-422

RS-485

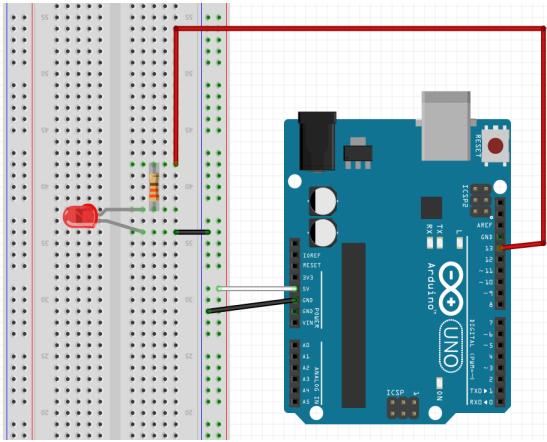
Arduino에서는 다음과 같은 목적으로 사용

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 bps로 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

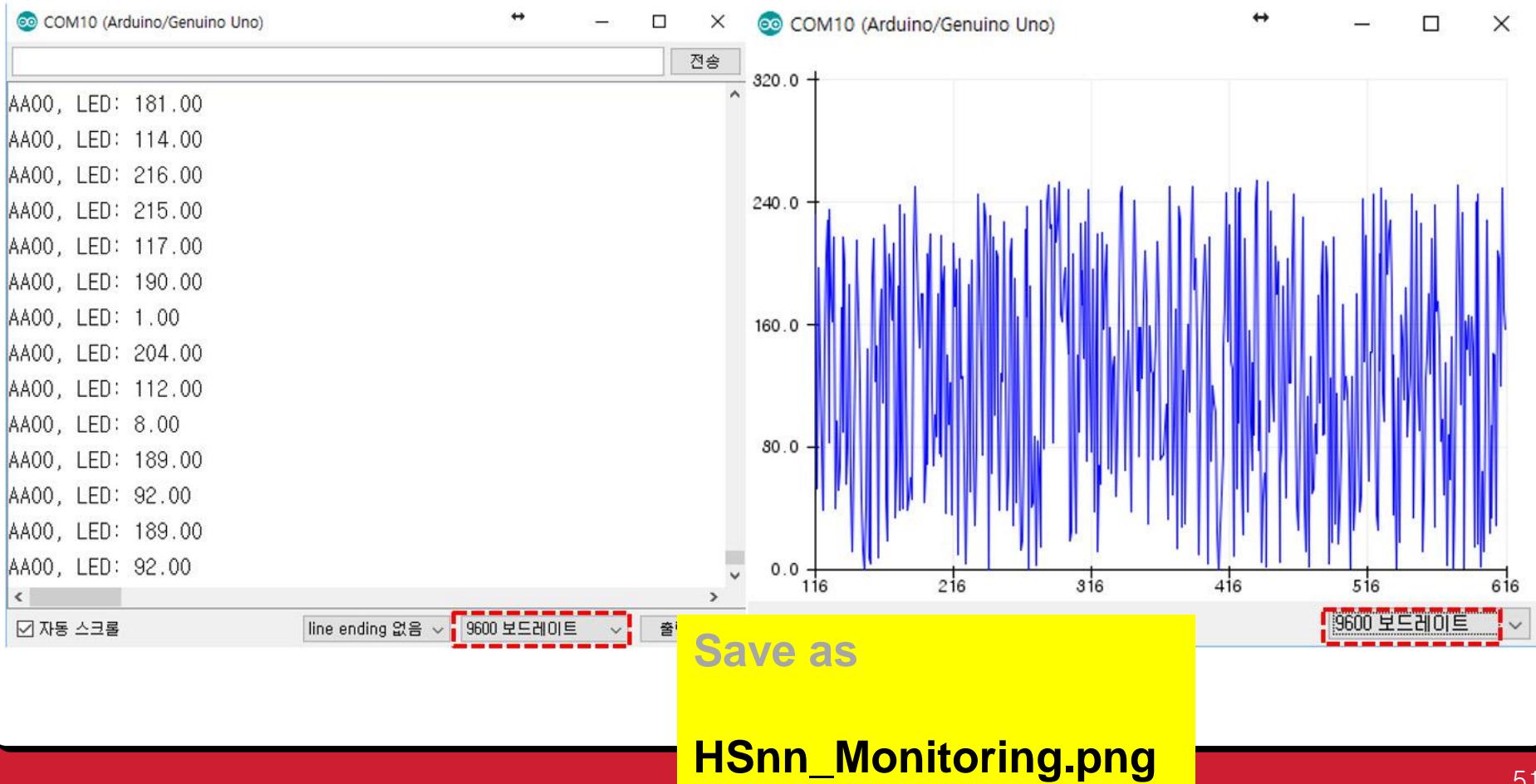
```
sketch03_pwm_led_serial $  
1 // sketch03_pwm_led_serial.ino  
2 int pwm = 0;  
3 int led = 3;  
4  
5 void setup() {  
6     Serial.begin(9600);  
7 }  
8  
9 void loop() {  
10    // put your main code here:  
11    pwm = random(0,255);  
12    pwmLed(led, pwm);  
13  
14    Serial.print("HS00, 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

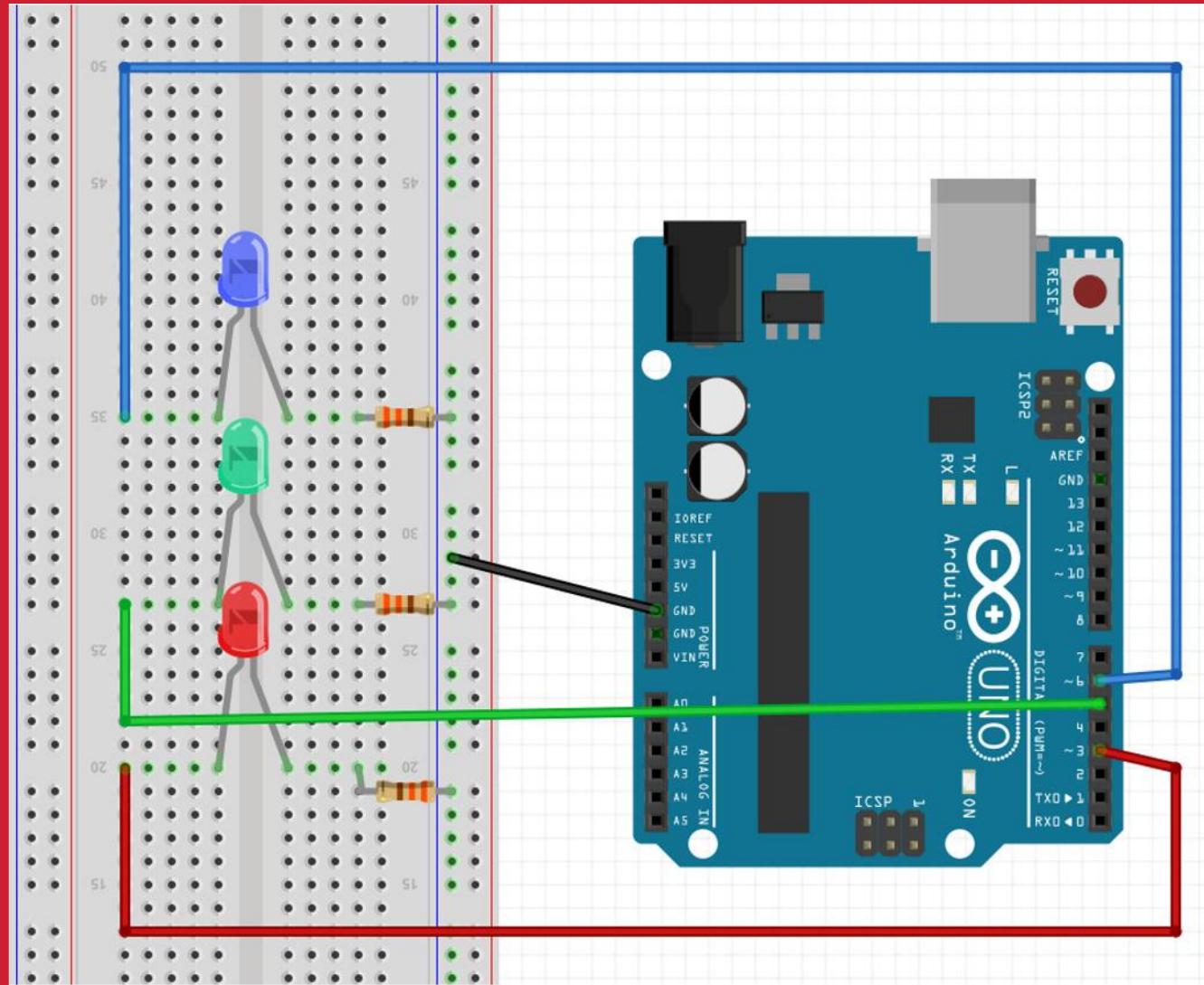
실습 결과

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

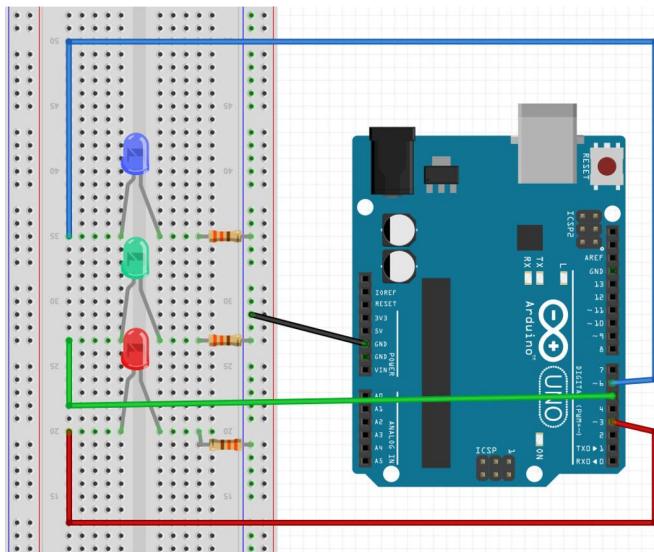




3 LED 끌어다



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("HS00, 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

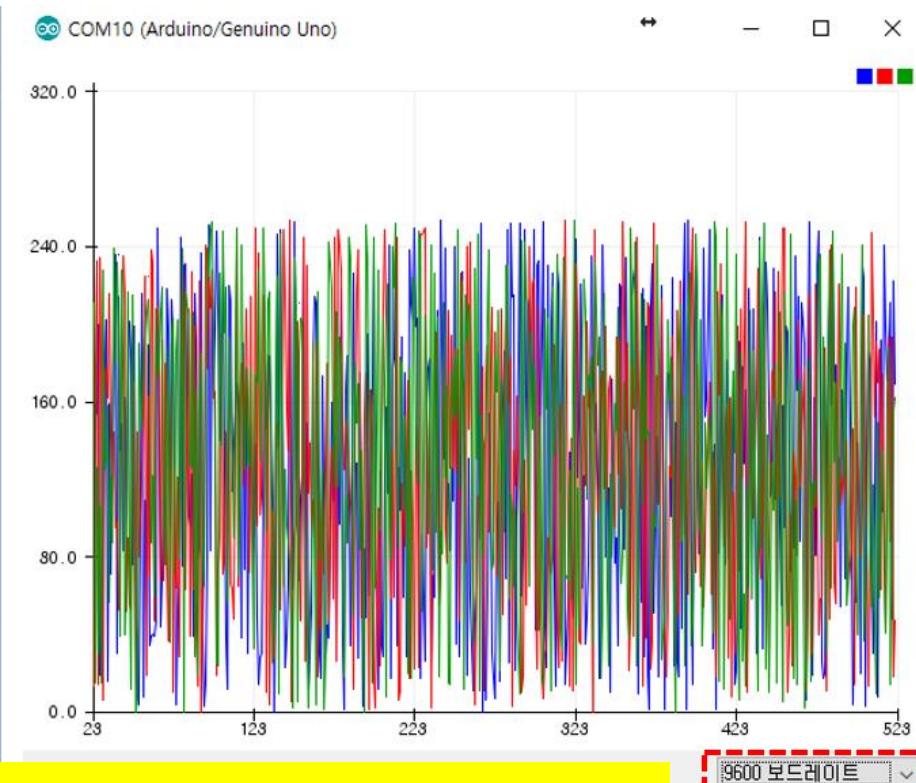
실습 결과

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

```
COM10 (Arduino/Genuino Uno)
```

```
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:
```

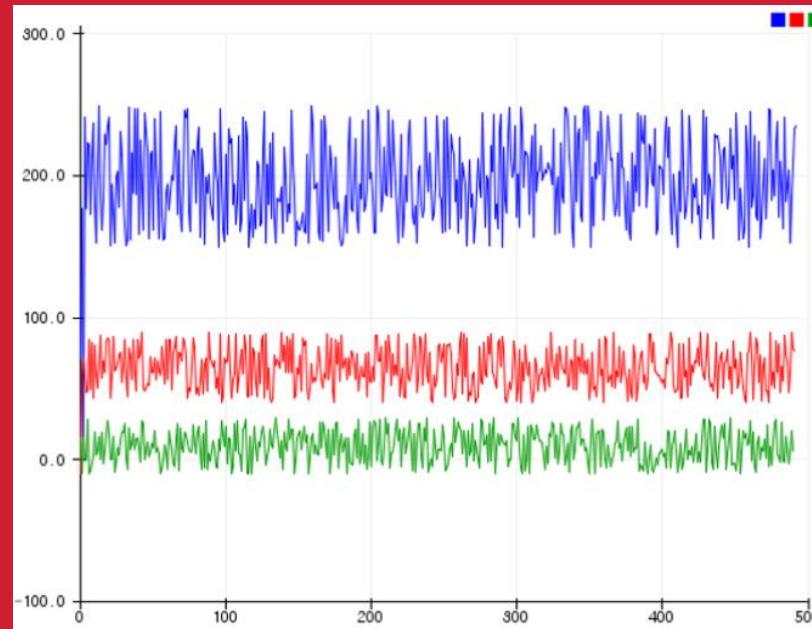
자동 스크롤 line ending 없음 9600 보드레이트





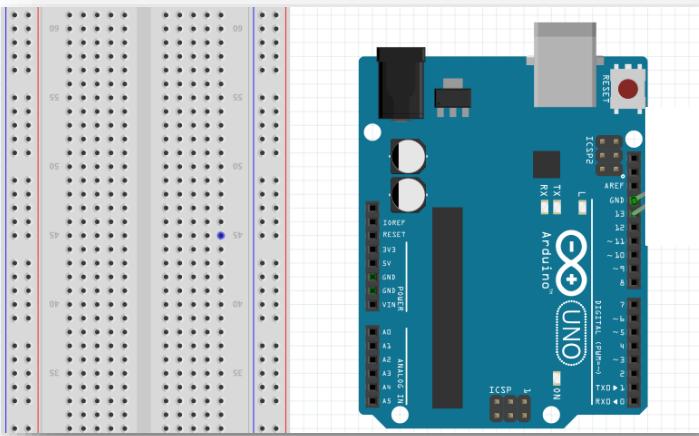
[DIY] Multi-signals

다중신호 시뮬레이션
및 모니터링





DIY - 스케치



아두이노에서 **LED와 저항을 모두 제거하고 USB만 컴퓨터와 연결한다.**

전자 소자 연결 없이 마구잡이 수 생성 암수를 이용해서
조도, 습도, 온도에 해당되는 **3개의 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 담을 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("HS00, Ambient Lux: ");
23   Serial.print(lux);
24   Serial.print(" , Humidity: ");
25   Serial.print(humi);
26   Serial.print(" , Temperature: ");
27   Serial.println(temp);
28   delay(100);      // delay in between reads for stability
29 }
```

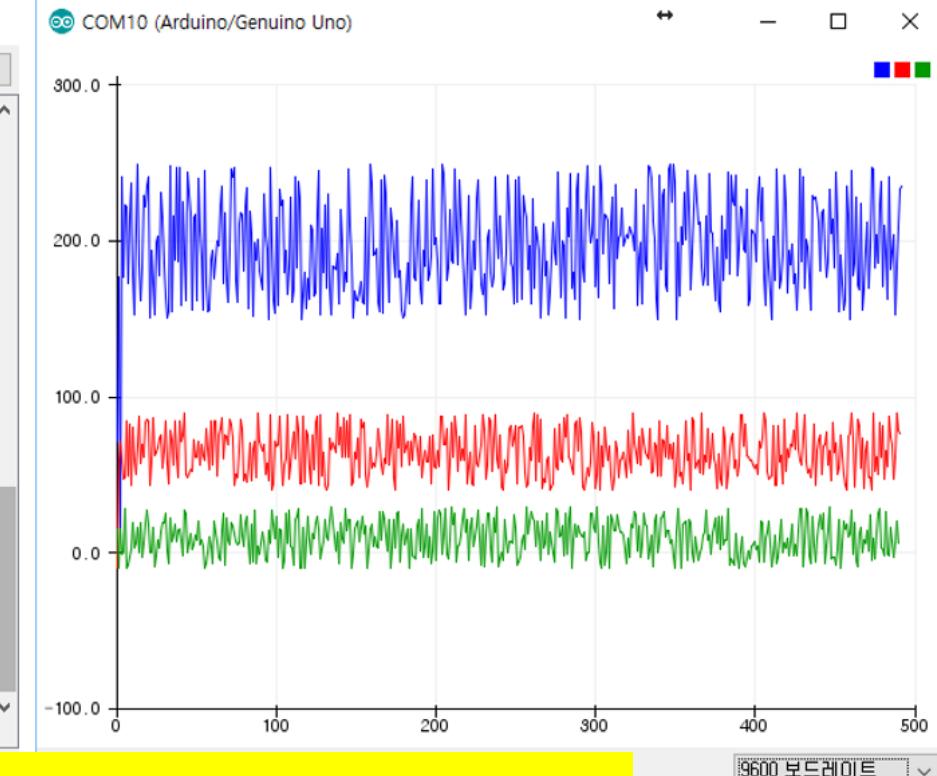


DIY – result

DIY 결과

가상적인 세 개의 센서신호 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
```

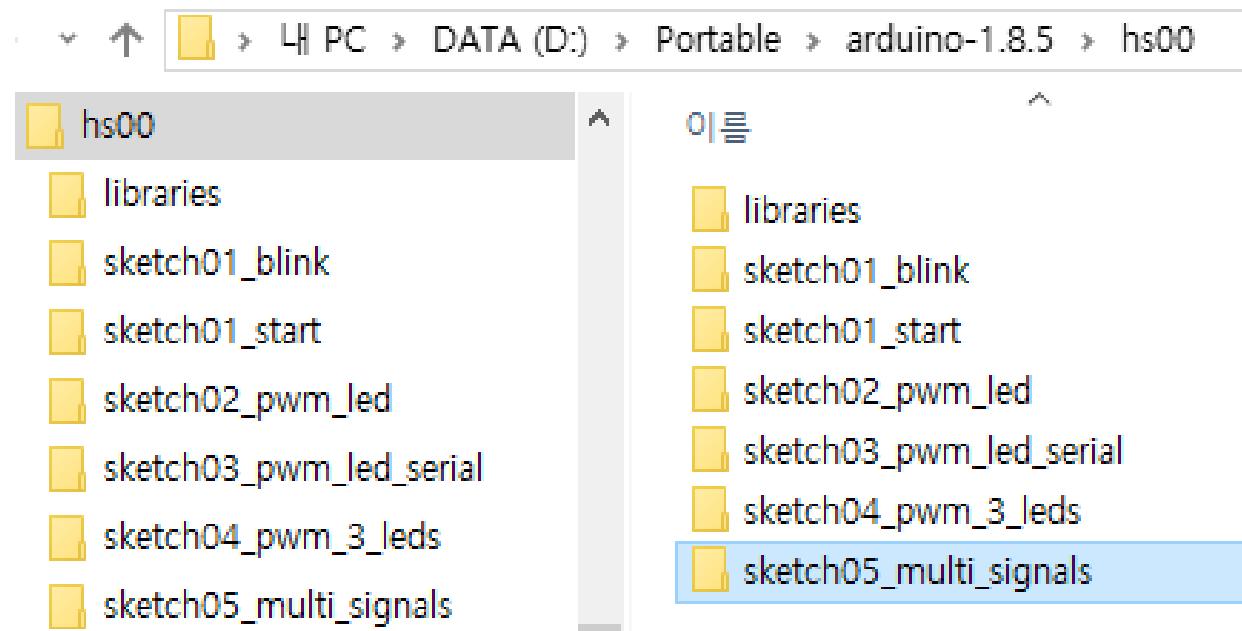


Save as

HSnn_multi_Signals.png



[My working folder]



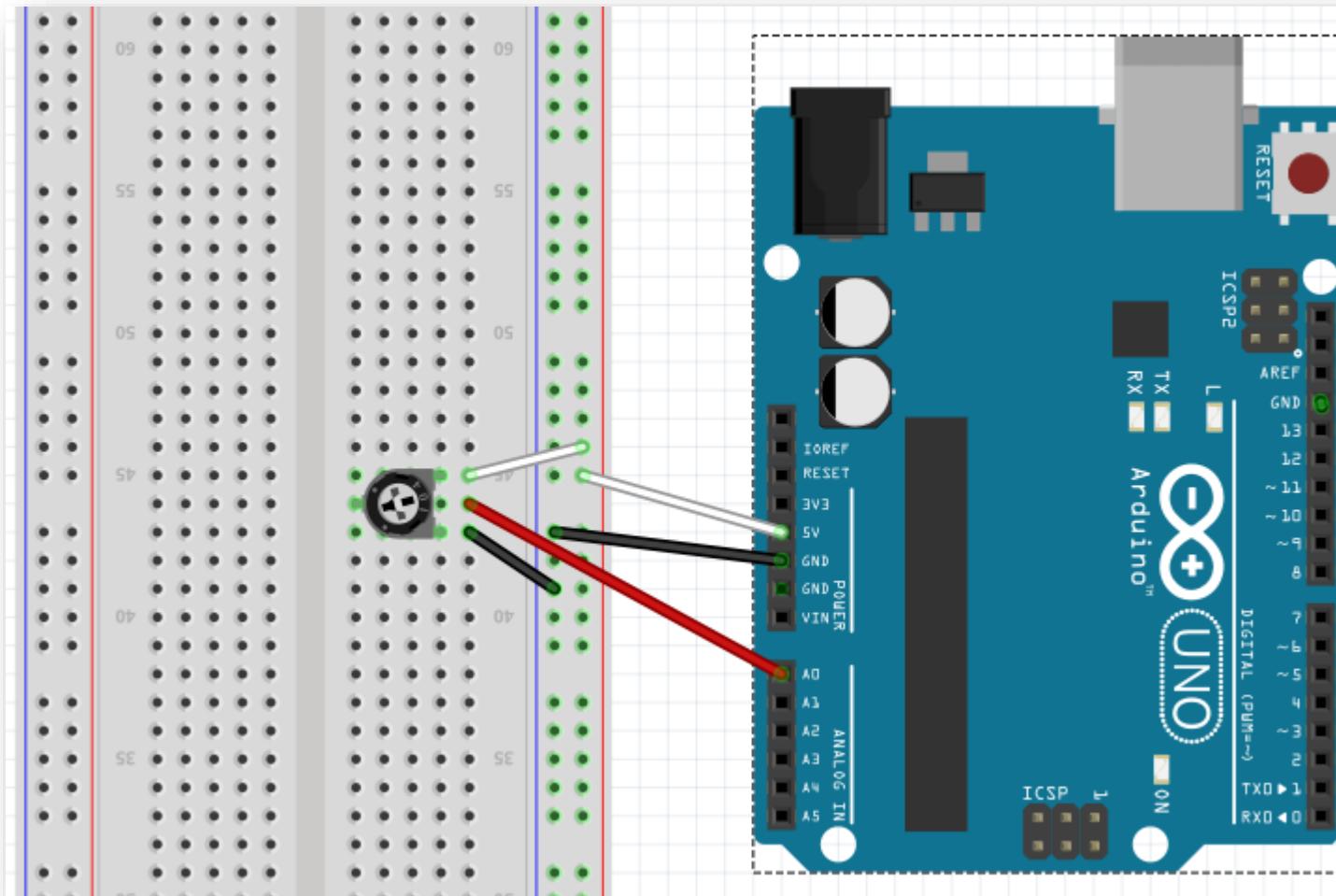


Analog Signal



A2.5.1 AnalogReadSerial (circuit)

Standard potentiometer (가변 저항기)





[Practice]

- ◆ [wk04]
 - Arduino basic circuits
 - Complete your project
 - Submit file : HSnn_Rpt03.zip

wk04 : Practice-03 : HSnn_Rpt03.zip

◆ [Target of this week]

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

제출파일명 : **HSnn_Rpt03.zip**

- 압축할 파일들

- ① **HSnn_Blink.png**
- ② **HSnn_Monitoring.png**
- ③ **HSnn_multi_Monitoring.png**
- ④ **HSnn_multi_Signals.png**

Email : **chaos21c@gmail.com**

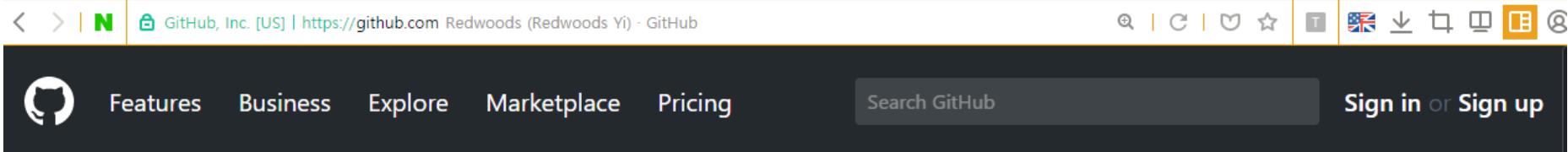
【 제목 : id, 이름 (수정) 】

Lecture materials

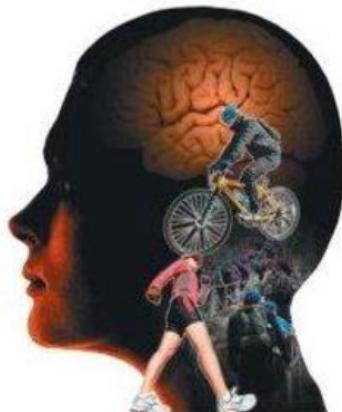


● References & good sites

- ✓ <http://www.nodejs.org/ko> Node.js
- ✓ <http://www.arduino.cc> Arduino Homepage
- ✓ <http://www.w3schools.com> By w3schools
- ✓ <http://www.github.com> GitHub
- ✓ <http://www.google.com> Googling



A screenshot of a GitHub user profile page. At the top, there's a navigation bar with icons for back, forward, and search, followed by the URL "GitHub, Inc. [US] | https://github.com Redwoods (Redwoods Yi) - GitHub". To the right are icons for search, refresh, notifications, and account management. Below the bar, the GitHub logo is on the left, followed by links for "Features", "Business", "Explore", "Marketplace", and "Pricing". A search bar says "Search GitHub" and there are "Sign in or Sign up" buttons.



Redwoods Yi

Redwoods

[Block or report user](#)

 GimHae, Republic of Korea

Overview

Repositories 5

Stars 2

Followers 0

Following 0

Pinned repositories

dht22-iot-project

Iot project to monitor data streaming from DHT22 wired at Arduino.

HTML

Lec

All lectures by Redwoods in Inje University

arduino-nodejs-plotly-streaming

This repo introduces a simple and efficient way to plot the streaming data from Arduino with Easy Pulse ppg sensor or DHT11 sensor.

HTML

hw-coding

Resource for lecture of Hardware Programming (2017, Inje university)

Arduino

Redwoods / Lec

Unwatch ▾

1

Code

Issues 0

Pull requests 0

Projects 0

Wiki

Insights

Settings

All lectures by Redwoods in Inje University

Add topics

81 commits

1 branch

0 releases

Branch: master ▾

New pull request

Create new file

Upload files

 Redwoods 2018 wk01 upload	Lat
 advanced-Arduino-iot	wk16 exam upload
 ev3	wk16 final exam. answers
 healthcare-signal-iot	2018 wk01 upload
 html5-basic	2018 wk01 upload
 html5-mobile-simulation	wk15 lec upload
 Lec.Rproj	2018 wk01 upload
 README.md	wk03 upload and fix links

This repository Search Pull requests Issues Marketplace Explore

Unwatch 1

Code Issues 0 Pull requests 0 Projects 0 Wiki Insights Settings

Branch: master Lec / healthcare-signal-iot / Create new file Upload

Redwoods 2018 wk03 upload Latest

..

src	2018 wk03 upload
README.md	2018 wk01 upload
wk01_hs_Intro.pdf	2018 wk01 upload-2
wk01_hs_Intro.pptx	2018 wk01 upload-2
wk02_hs_nodejs.pdf	2018 wk02 upload
wk03_hs_node_express.pdf	2018 wk03 upload

README.md

Lec : Introduction to Healthcare Signal Visualization

All lectures by Redwoods in Inje University from 2018 and 2017.



1.0 What is node.js?

← → ⌂ ⌂ 🔒 안전함 | https://www.w3schools.com/nodejs/nodejs_intro.asp

HOME CSS JAVASCRIPT SQL PHP BOOTSTRAP HOW TO JQUERY MORE ▾

Node.js Tutorial
Node.js HOME
Node.js Intro
Node.js Get Started
Node.js Modules
Node.js HTTP Module
Node.js File System
Node.js URL Module
Node.js NPM
Node.js Events
Node.js Upload Files
Node.js Email

Node.js MySQL
MySQL Get Started
MySQL Create Database
MySQL Create Table

Node.js Introduction

< Previous

What is Node.js?

- Node.js is an open source server framework
- Node.js is free
- Node.js runs on various platforms (Windows, Linux, Unix, Mac OS X, etc.)
- Node.js uses JavaScript on the server

Why Node.js?

Node.js uses asynchronous programming!

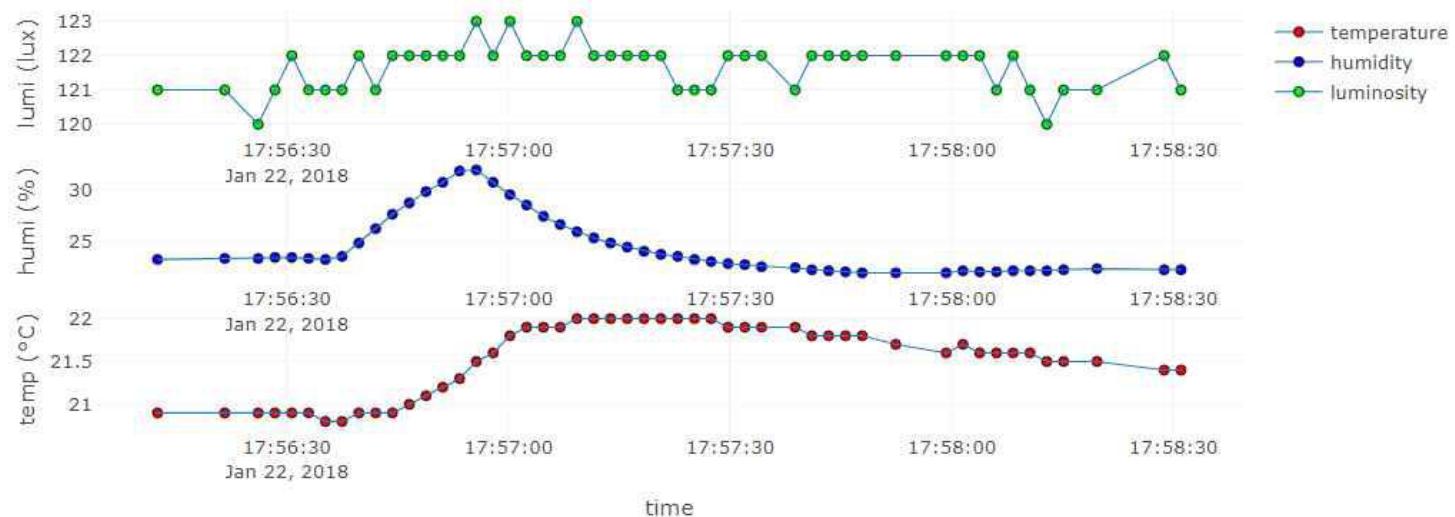
https://www.w3schools.com/nodejs/nodejs_intro.asp

Target of this class

Real-time Weather Station from sensors



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



Project of this class

