

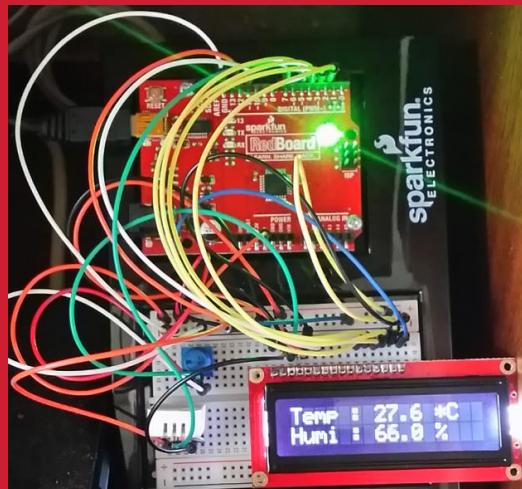


HW-SW-Connectivity

[wk05]

Arduino Circuit I.

on Time: 2015-09-02 12:48:14.192

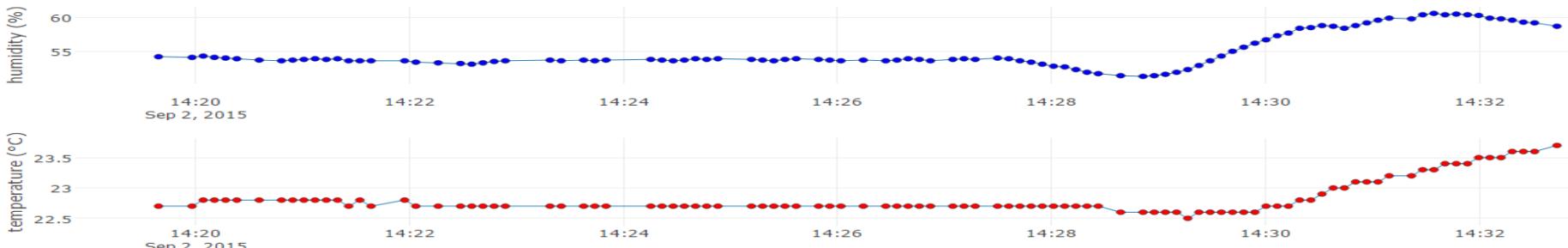


Basic HW and SW Integration using
Arduino & Javascript

COMSI, INJE University

2nd semester, 2017

Email : yish@inje.ac.kr





주간계획서

2017학년도 2학기

주간계획서		수업내용	과제물
주차	수업방법		
1	이론	교육과정 안내	가상강의 등록
2	이론/실습	모바일 서버 기초 : node.js 설치	
3	이론/실습	모바일 서버 프로그래밍 : node.js 응용	점검과제
4	이론/실습	기초 하드웨어: 아두이노 소개	점검과제
5	이론/실습	기초 하드웨어 프로그래밍: 아두이노 센서	프로젝트 1
6	이론/실습	시각화 프로그래밍: plotly.js	점검과제
7	이론/실습	아두이노 센서 신호 처리 및 시각화	프로젝트 1
8	시험 또는 실습과제	중간고사	
9	이론/실습	아두이노 센서 신호 시각화 고급 프로그래밍	점검과제
10	이론/실습	모바일 데이터베이스 I: mongoDB 설치	점검과제
11	이론/실습	모바일 데이터베이스 II: mongoDB 응용 프로그래밍, Compass 활용	프로젝트 1
12	이론/실습	모바일 클라인언트 프로그래밍 I: Angular.js 기초	점검과제
13	이론/실습	모바일 클라인언트 프로그래밍 II: Angular.js 응용 프로그래밍	점검과제
14	이론/실습	하드웨어와 소프트웨어 융합 IOT 프로젝트	프로젝트 IV
15	이론/실습	기말고사	

Weekly schedule of HSC– 2nd semester, 2017



- **wk01 : Introduction to class and enrollment in cyber class (Sublime text 3 install)**
- **wk02 : Basic mobile server : node.js install and test**
- **wk03 : Mobile server programming : node.js App**
- **wk04 : Basic HW : Arduino I. – circuit & programming, Arduino SW install**
- **wk05 : Basic HW : Arduino II. – sensor circuit & programming**
- **wk06 : Visualization using Javascript – ployly.js and gauge.js**
- **wk07 : Project-1 : Handling and visualization of signals from various sensors**
- **wk08 : Mid-term exam.**
- **wk09 : Advanced programming to visualize signals from sensors**
- **wk10 : Mobile database I : Mongo DB install**
- **wk11 : Mobile database II : Mongo DB App.**
- **wk12 : Mobile client programming I : Angular.js install**
- **wk13 : Mobile client programming II : Angular.js App.**
- **wk14 : Project-2 : IOT project fusing HW & SW**
- **wk15 : Final exam.**





[Practice]

◆ [wk04]

- My Express App: multi-routing
- Add a new route in index.js
- Upload file name : AAnn_Rpt03.zip



7.4.8 myApp : run myApp

1. npm install

```
cmd: npm
myapp@0.0.0 D:\Portable\NodeJSPortable\Data\aa00\myApp
+-- body-parser@1.17.2
| +-- bytes@2.4.0
| +-- content-type@1.0.4
| +-- debug@2.6.7
| +-- depd@1.1.1
| +-- http-errors@1.6.2
| | +-- inherits@2.0.3
| | +-- ionic-lite@0.4.15
| | +-- on-finished@2.3.0
| | | +-- ee-first@1.1.1
| | +-- qs@6.4.0
| | +-- raw-body@2.2.0
| | | +-- unpipe@1.0.0
| | | +-- type-is@1.6.15
| | | | +-- media-types@0.3.0
| | | | +-- mime-db@1.30.0
| +-- cookie-parser@1.4.3
| | +-- cookie@0.3.1
| | +-- cookie-signature@1.0.6
+-- debug@2.6.8
+-- ms@2.0.0
+-- express@4.15.4
| +-- accepts@1.3.4
| | +-- negotiator@0.6.1
+-- array-flatten@1.1.1
+-- content-disposition@0.5.2
+-- encodeurl@1.0.1
+-- escape-html@1.0.3
+-- etag@1.8.1
+-- finalhandler@1.0.4
+-- fresh@0.5.0
+-- merge-descriptors@1.0.1
+-- methods@1.1.2
+-- parseurl@1.3.2
+-- path-to-regexp@0.1.7
+-- proxy-addr@1.1.5
| +-- forwarded@0.1.2
+-- ipaddr.js@1.4.0
```

2. Run myApp !

- **^B on www (in SB3)**

```
Server running on 3000.  
Launch http://localhost:3000
```

- **node bin/www (in cmd)**

```
cmd: NodeJS - node ./bin/www
D:\Portable\NodeJSPortable\Data\aa00\myApp>node ./bin/www
Server running on 3000.
Launch http://localhost:3000
```

- **npm start (in cmd)**

```
cmd: npm
D:\Portable\NodeJSPortable\Data\aa00\myApp>npm start
> myapp@0.0.0 start D:\Portable\NodeJSPortable\Data\aa00\myApp
> node ./bin/www

Server running on 3000.
Launch http://localhost:3000
```



7.4.9 myApp : run myApp

D:\Portable\NodeJSPortable\Data\aa00\myApp\routes#index.js (aa00) - Sublime Text (UNREGISTERED)

File Edit Selection Find View Goto Tools Project Preferences Help

FOLDERS

```
▼ aa00
  ▼ myApp
    ▼ bin
      www
    ▷ node_modules
    ▼ public
      images
      javascripts
      stylesheets
    ▼ routes
      index.js
      users.js
    ▼ views
      error.jade
      index.jade
      layout.jade
    app.js
    package.json
```

```
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 by AA00' });
7 });
8
9 module.exports = router;
```



Server running on 3000.

Launch <http://localhost:3000>

```
ESC [0mGET / ESC [32m200 ESC [0m659.076 ms - 19
ESC [0mGET /stylesheets/style.css ESC [32m200
ESC [0mGET / ESC [36m304 ESC [0m13.718 ms - -ES
ESC [0mGET /stylesheets/style.css ESC [36m304
```

Express by AA00

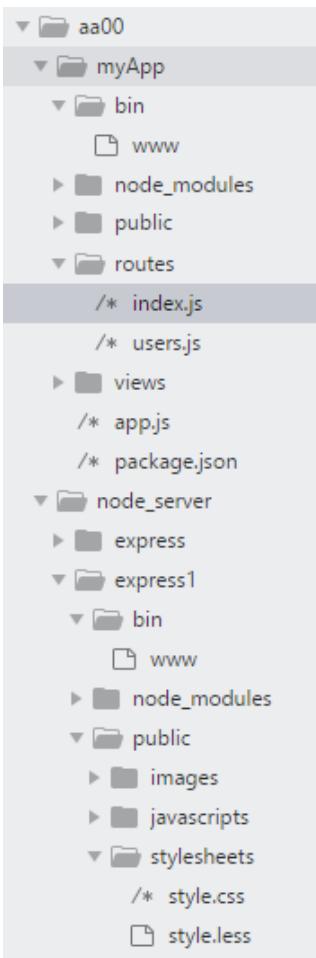
Welcome to Express by AA00

Save as
AAnn_App.png



[Practice-1] Modify routes/index.js

Add a new route '**/aann**' in index.js, multi-routing.



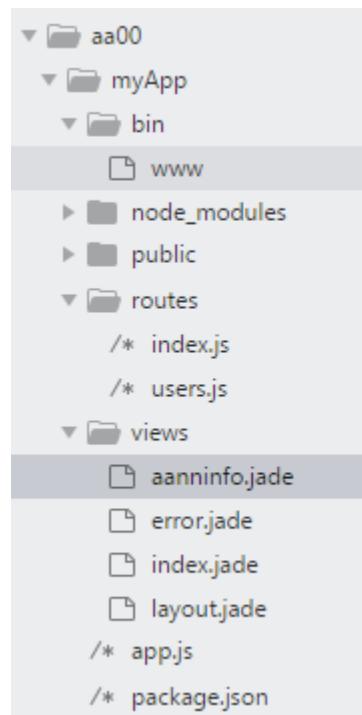
```
1 var express = require('express');
2 var router = express.Router();
3
4 /* GET home page. */
5 router.get('/', function(req, res, next) {
6   res.render('index', { title: 'Express by AA00' });
7 });
8
9 /* GET myInfo page, by /aann -> multi-routing */
10 router.get('/aann', function(req, res, next) {
11   res.render('aanninfo', { title: 'Express App by AA00',
12                         id: 'AAnn',
13                         name: 'COMSI' });
14   // views/aanninfo.jade
15 });
16
17
18 module.exports = router;
19
```

Use renderer aanninfo.jade in views folder



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

Add aanninfo.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_MyInfo_jade.png



[Practice-3] Result: your Info

Rerun node ./bin/www

localhost:3000/aann

The screenshot shows a web browser window titled "Express App by AA00". The address bar displays "localhost:3000/aann". The main content area of the browser shows the following text:

Express App by AA00

Welcome to Express App by AA00

My ID : AAnn

Developed by COMSI

Save as

AAnn_MyInfo.png

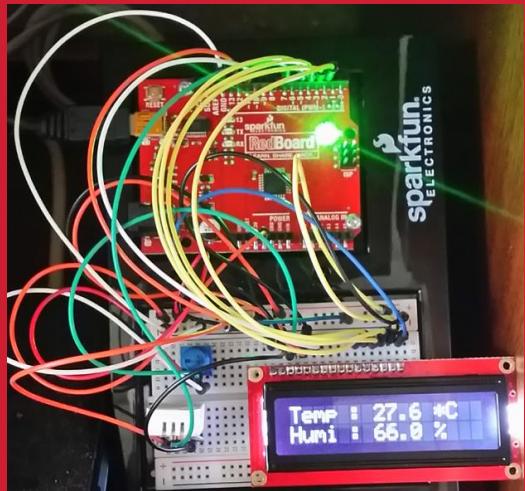


HW-SW-Connectivity

[wk04]

Arduino intro

on Time: 2015-09-02 12:48:14.192

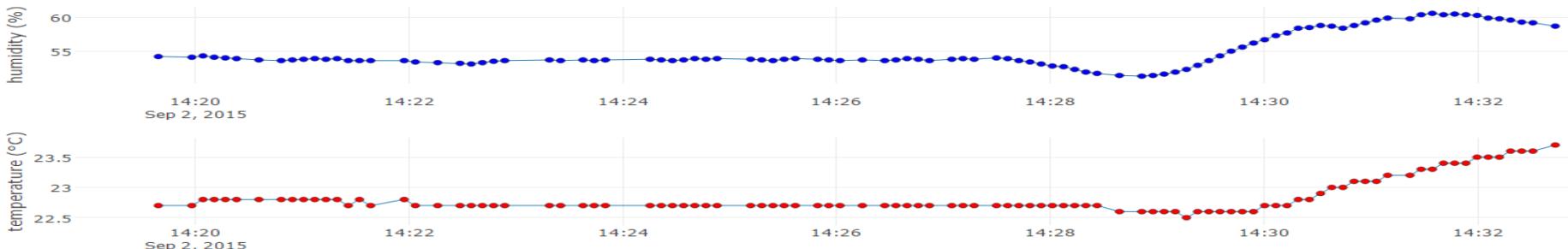


Basic HW and SW Integration using
Arduino & Javascript

COMSI, INJE University

2nd semester, 2017

Email : yish@inje.ac.kr





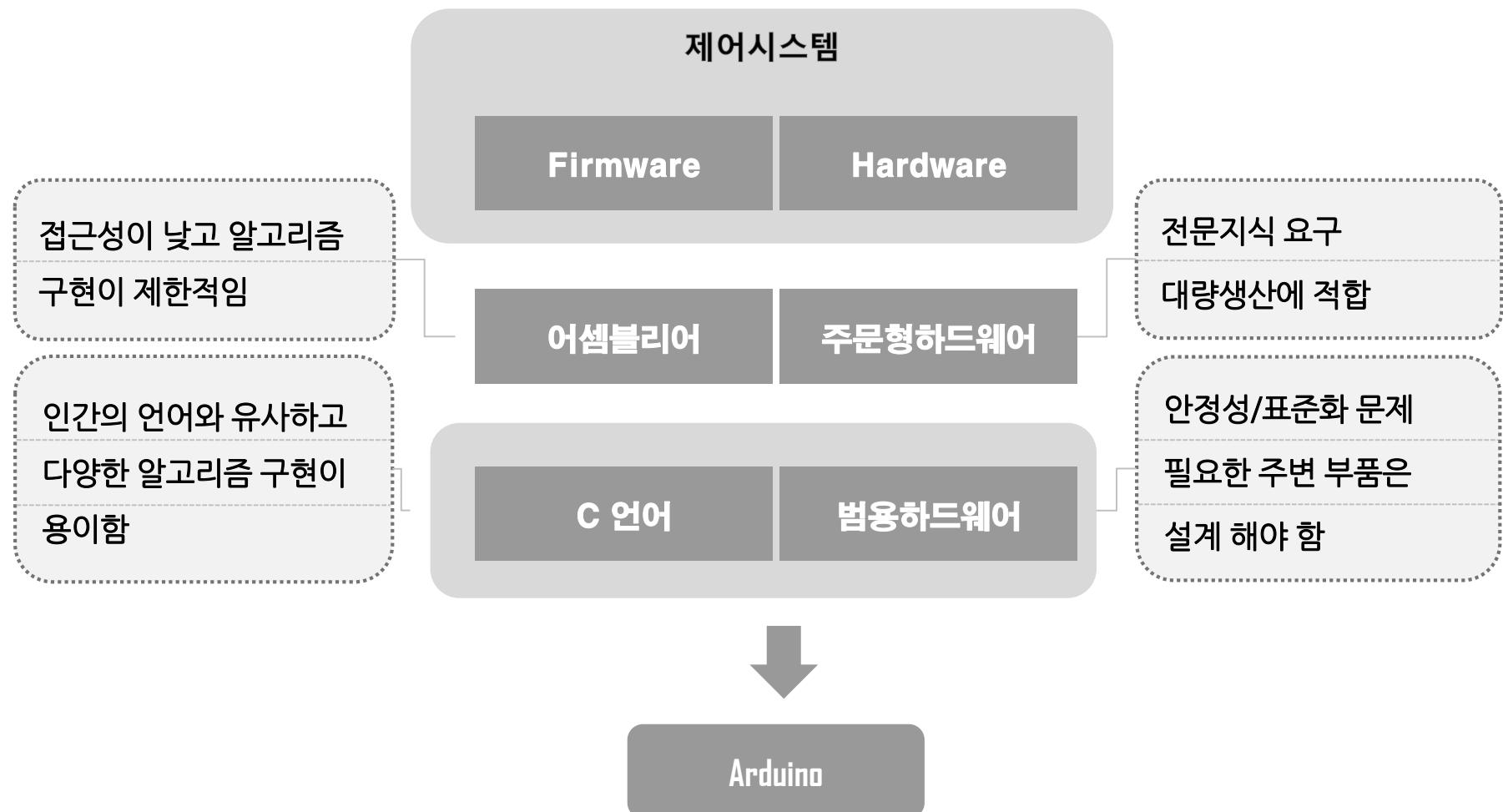
Arduino

[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

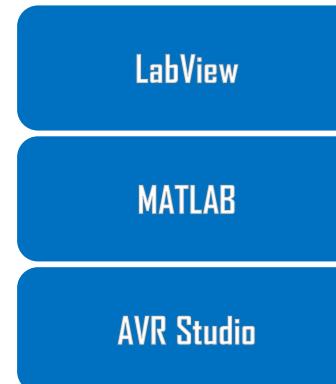
전자공학
교육

누구나
쉽게
사용
가능한
제어장치

오픈소스
하드웨어

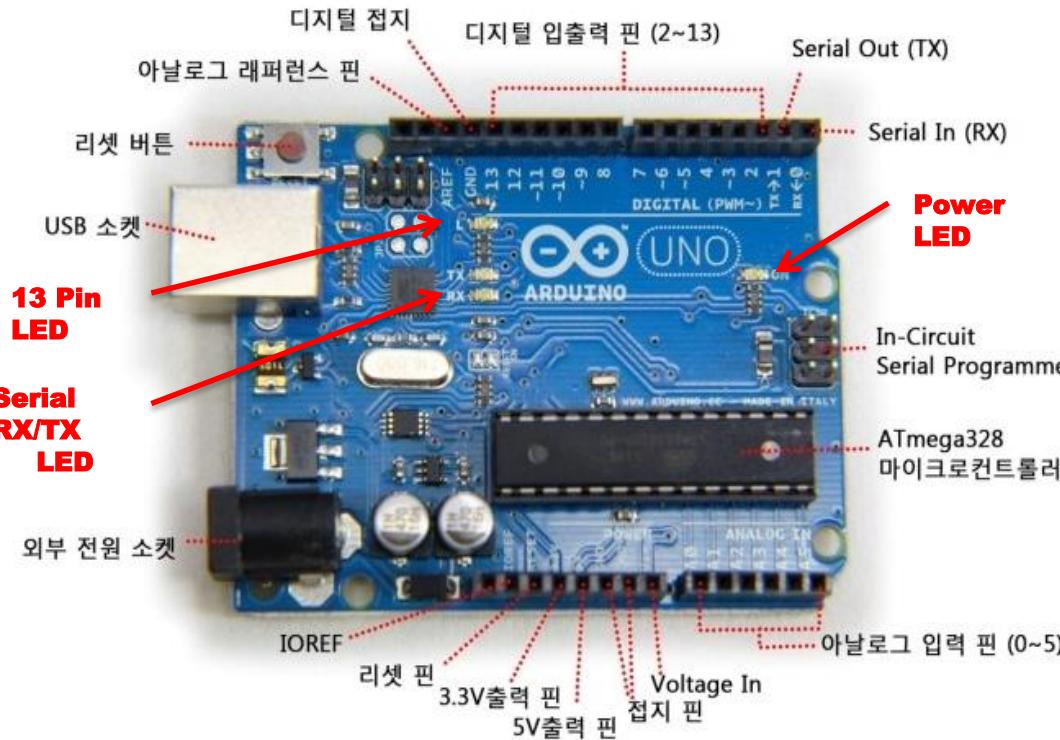
GSM Wifi Ethernet
Motor drive
등의 월드 제공

다양한
라이브러리



범용
하드웨어
의
표준

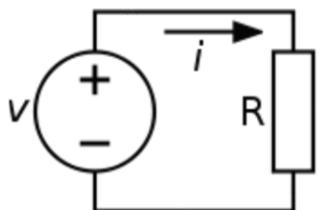
0.2 Arduino hardware



✓ 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

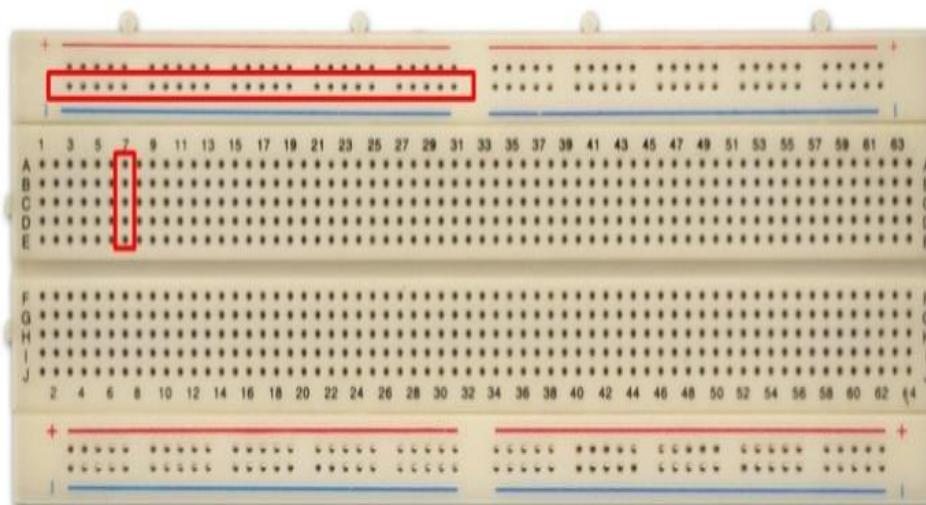
저항
[Ω]

- ✓ 전류의 흐름을 방해하는 정도를 나타냄

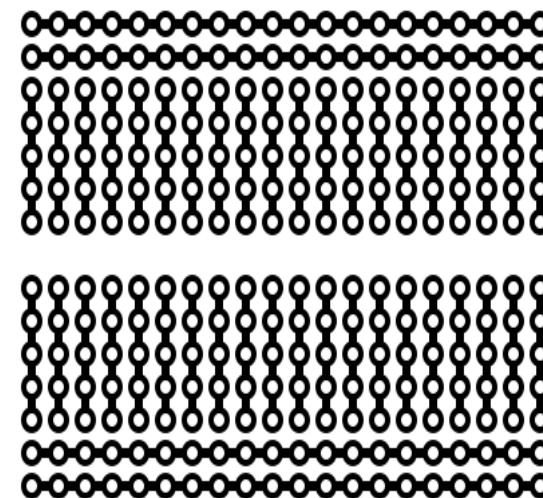
- ✓ 색 띠나 숫자로 값을 표시
- ✓ Arduino에서는 칩 (chip) 형태의 저항이 사용

0.4 브레드 보드 (Bread board)

시제품 제작이나 실험용 와이어를 보드에 꽂아 사용



빨간색 묶음 흘끼리 내부회로가 연결되어 있음



내부 결선

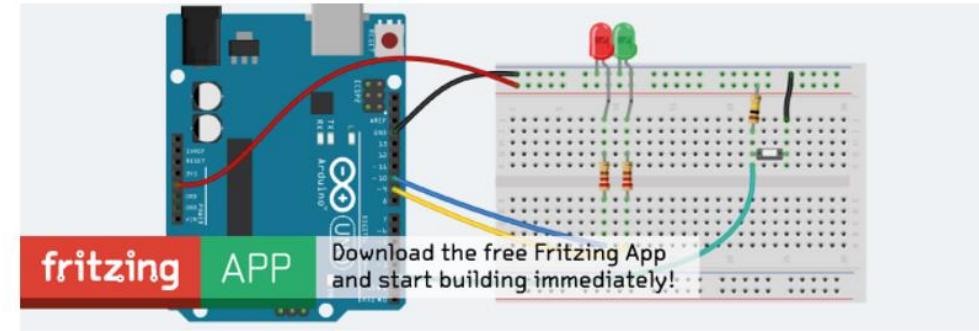


Arduino SW

[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.

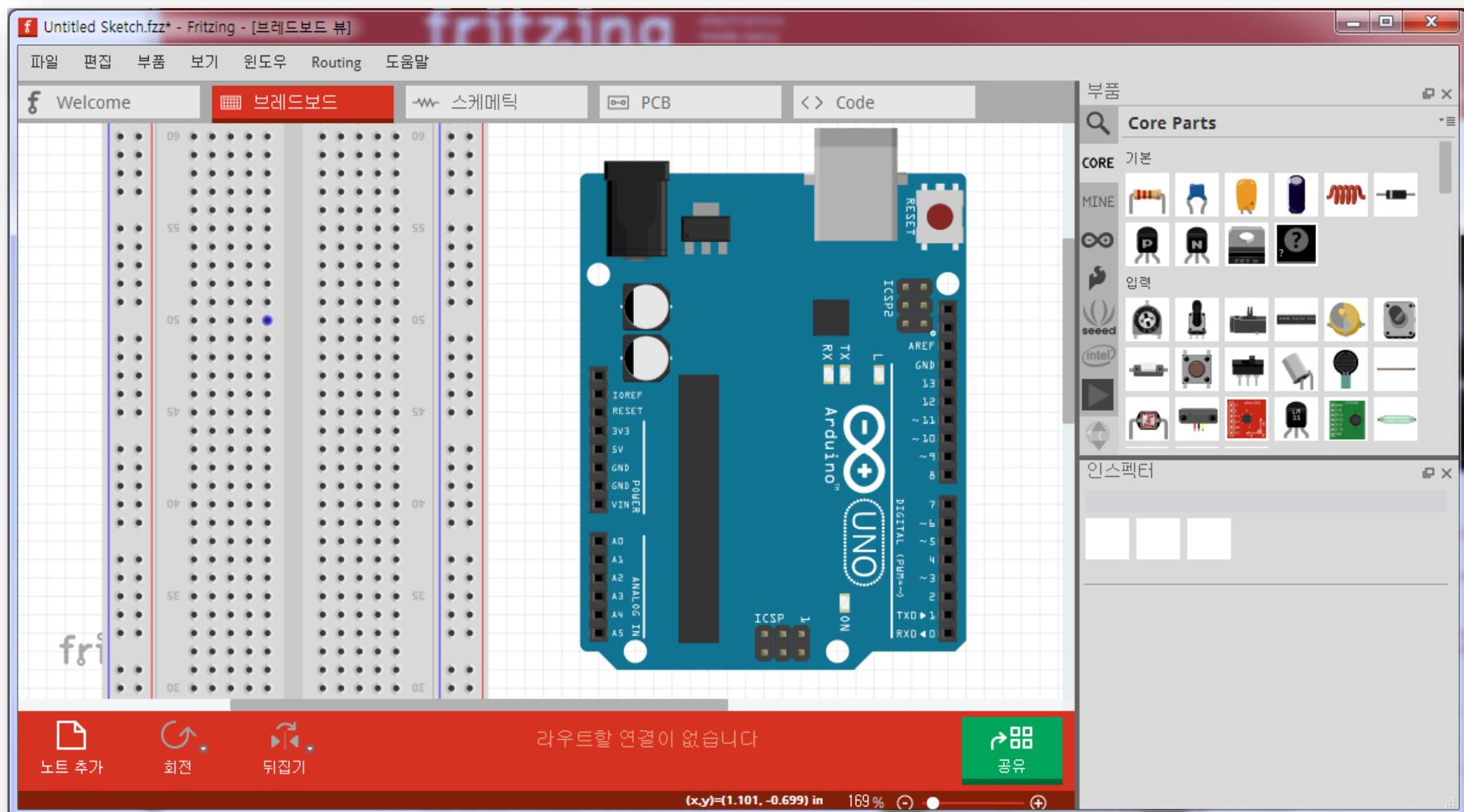
Produce your own board
With Fritzing Fab you can easily and inexpensively turn your circuit into a real, custom-made PCB. Try it out now!

Participate
Fritzing can only act as a creative platform if many



Fritzing configuration

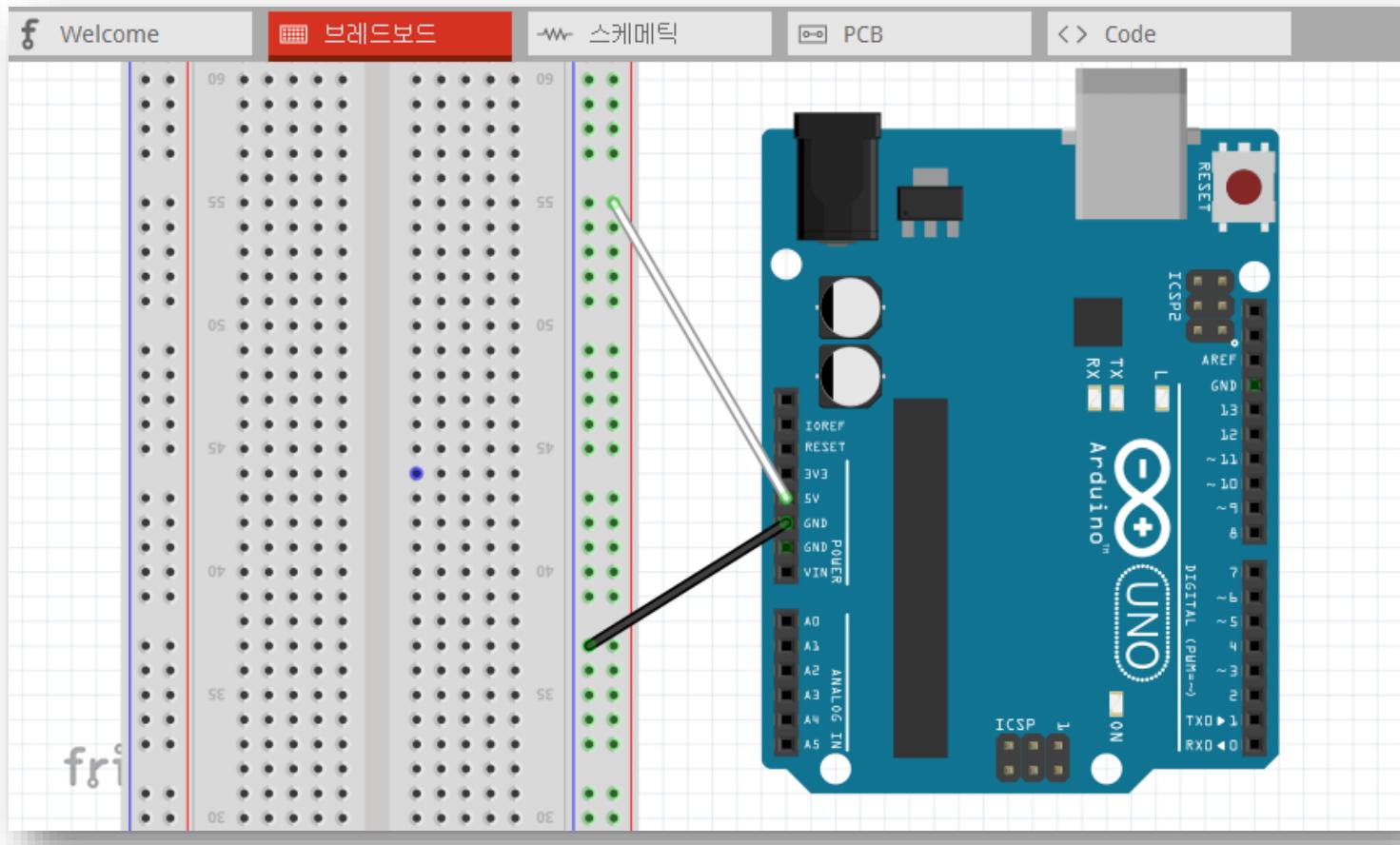
[Fritzing] configuration





Fritzing configuration

[Fritzing] power

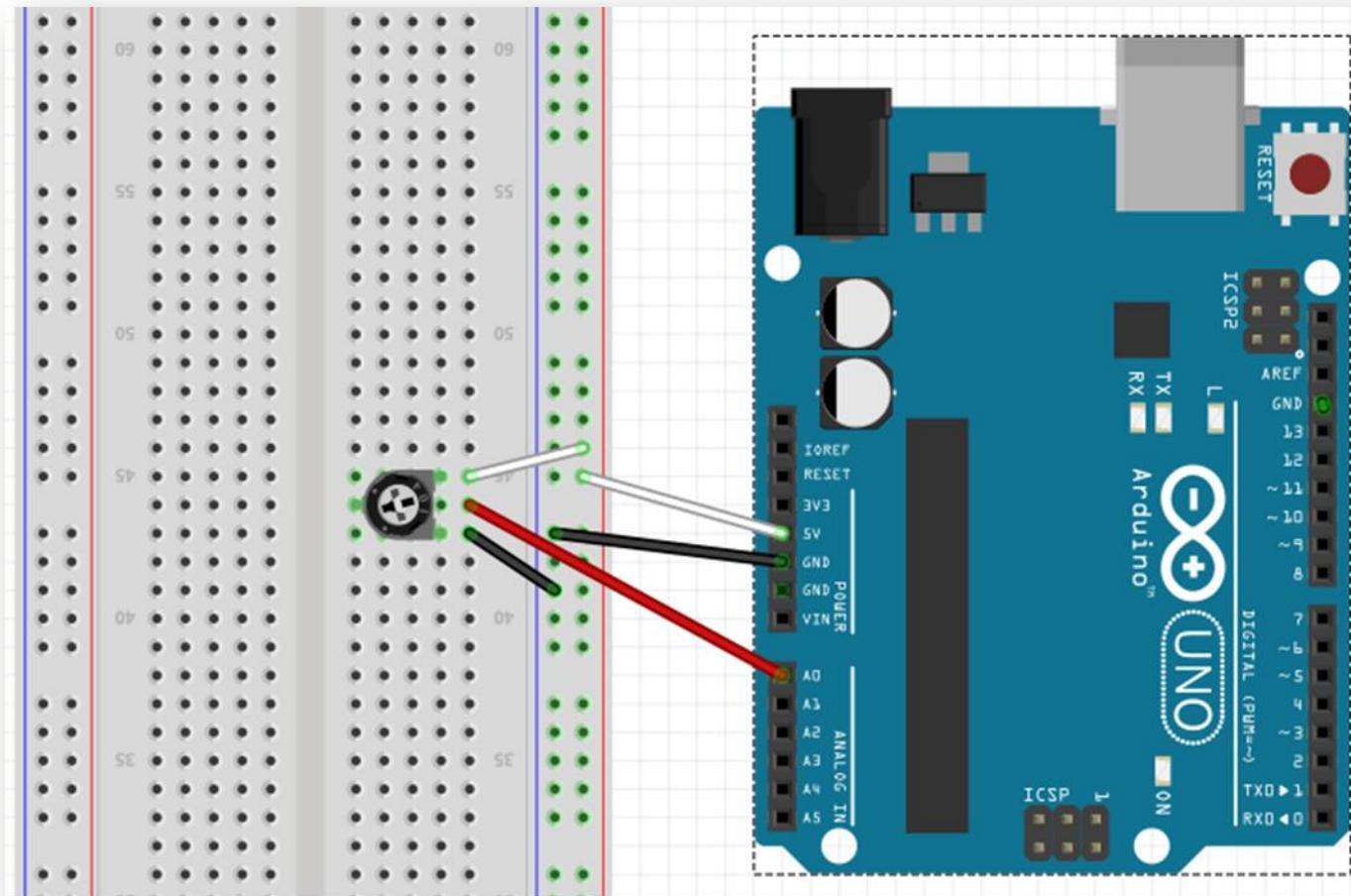




Arduino circuits



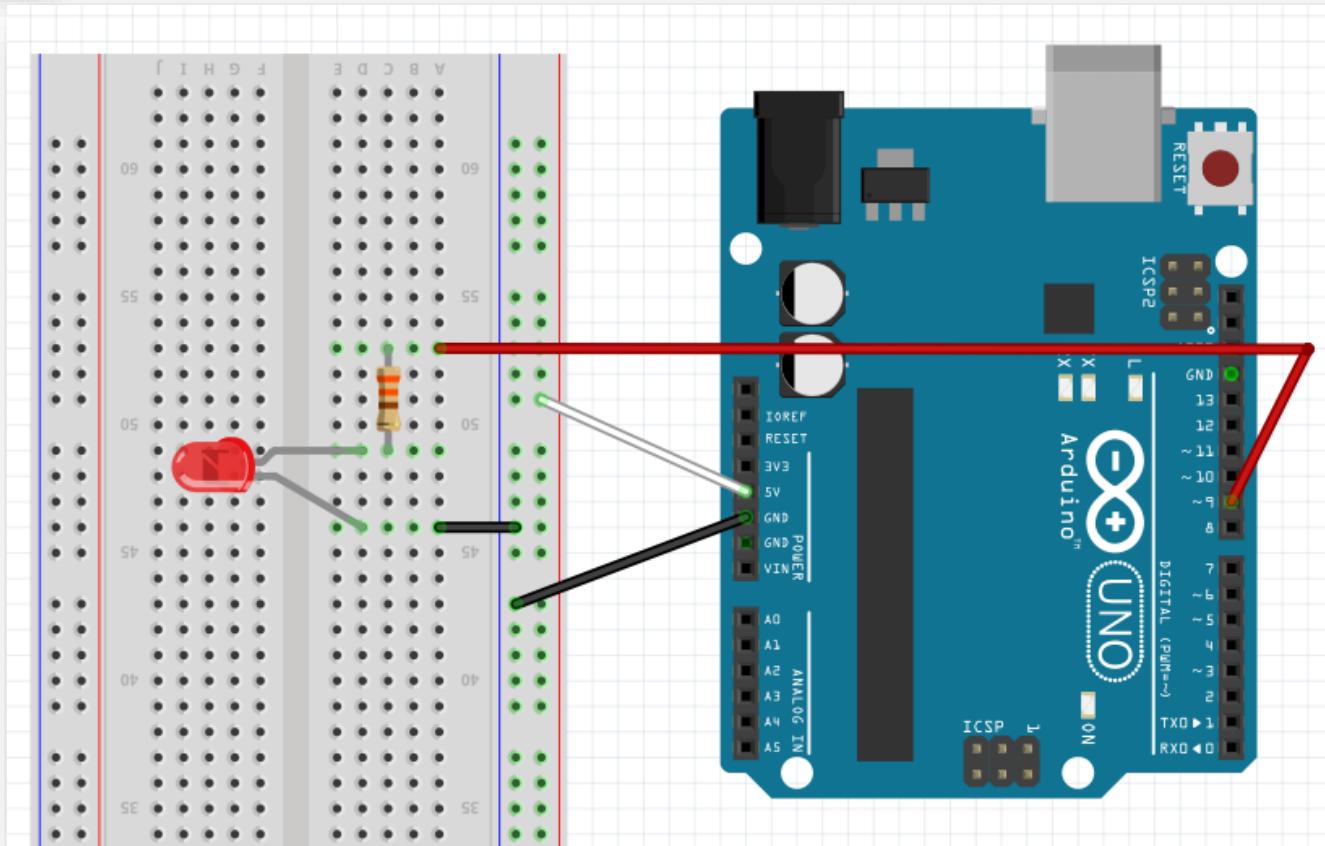
0.6.A1 Potentiometer (가변 저항기)



Parts : 가변저항기



0.6.A2 single LED



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



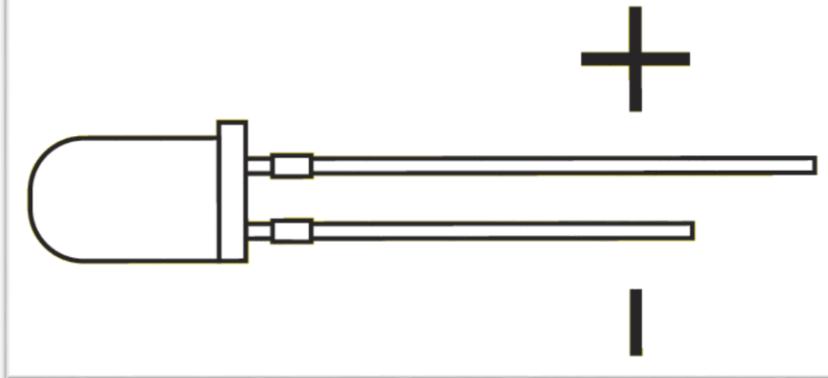


0.6.A2 single LED

Polarity of Diode and LED



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

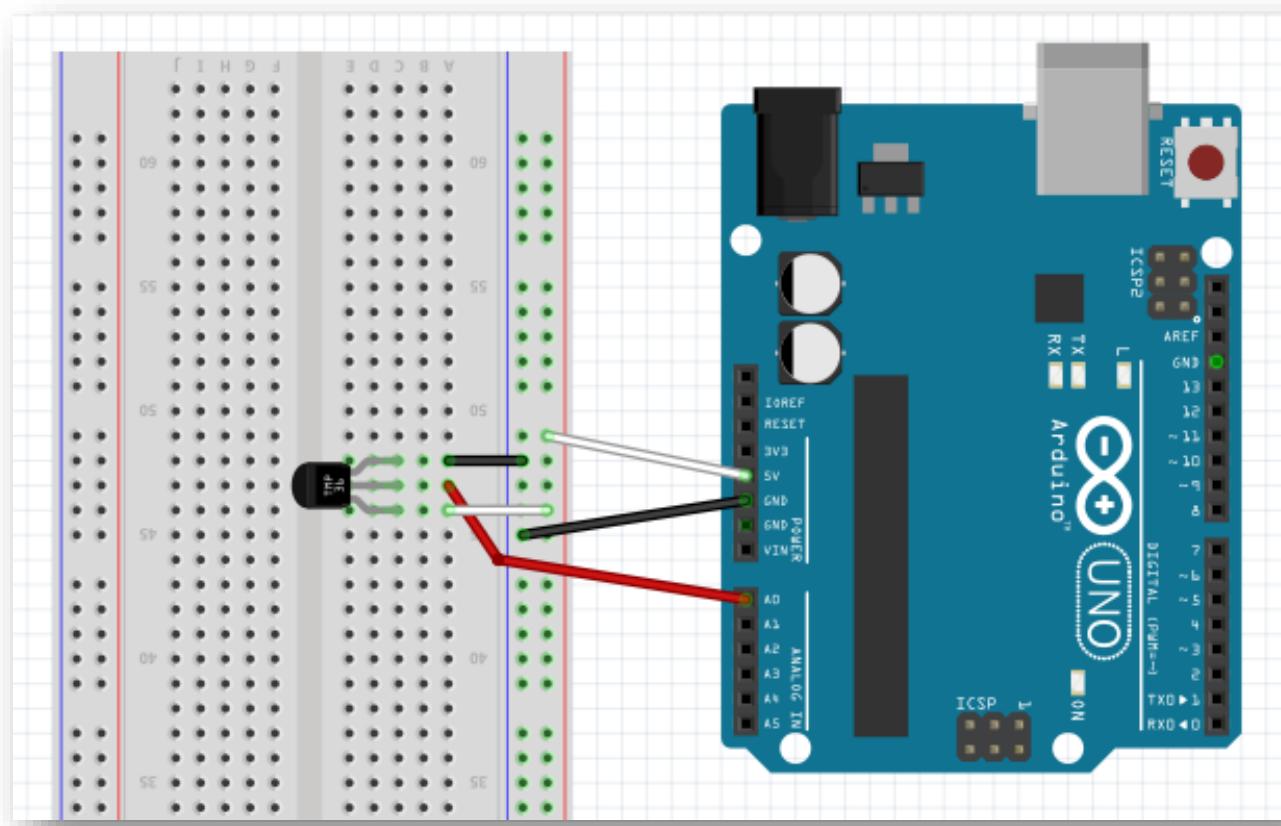


Find the longer leg, which should indicate the positive, anode pin.

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



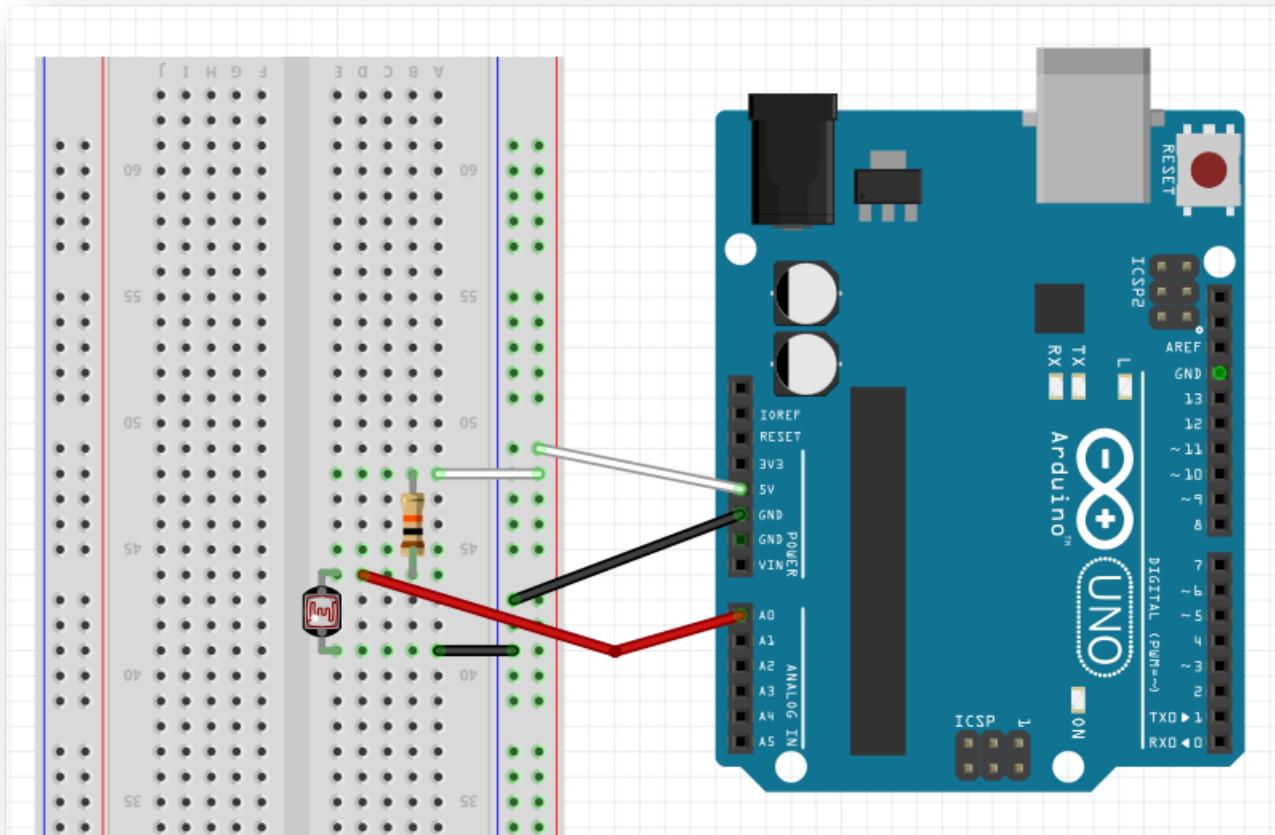
0.6.A4 Temperature sensor (TMP36)



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



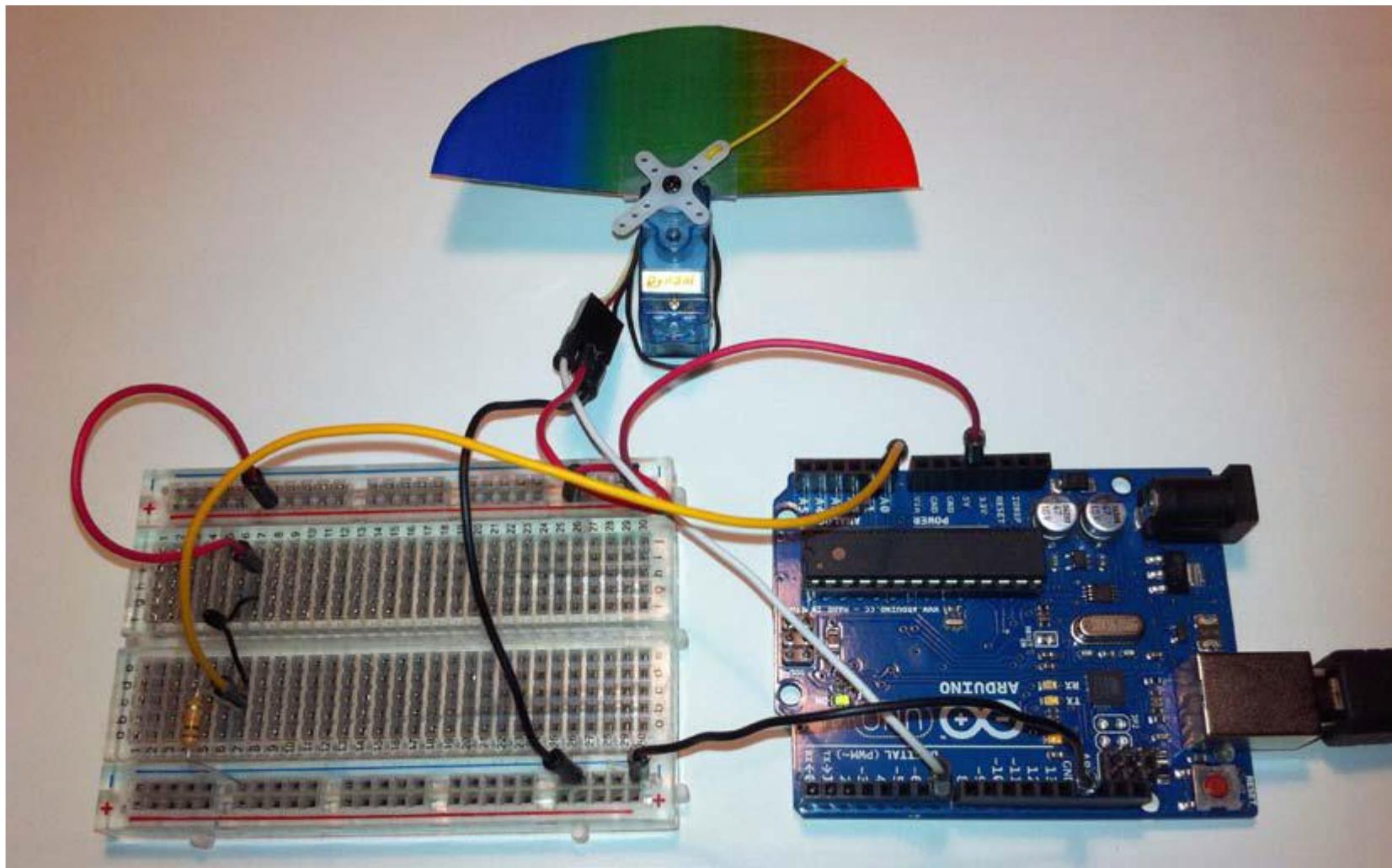
0.6.A5 Luminosity sensor : photo cell LDR



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

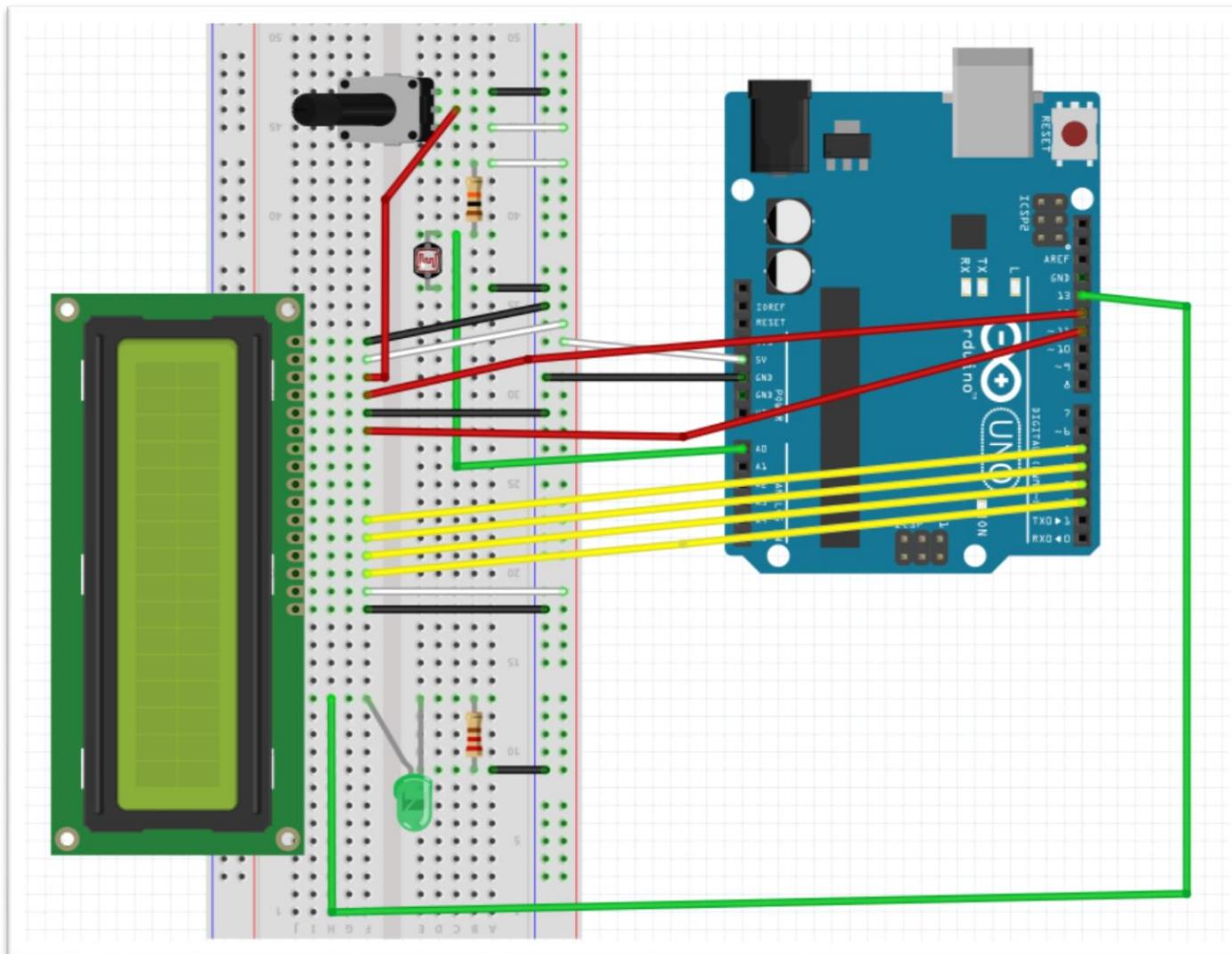


0.6.DIY3 Servo



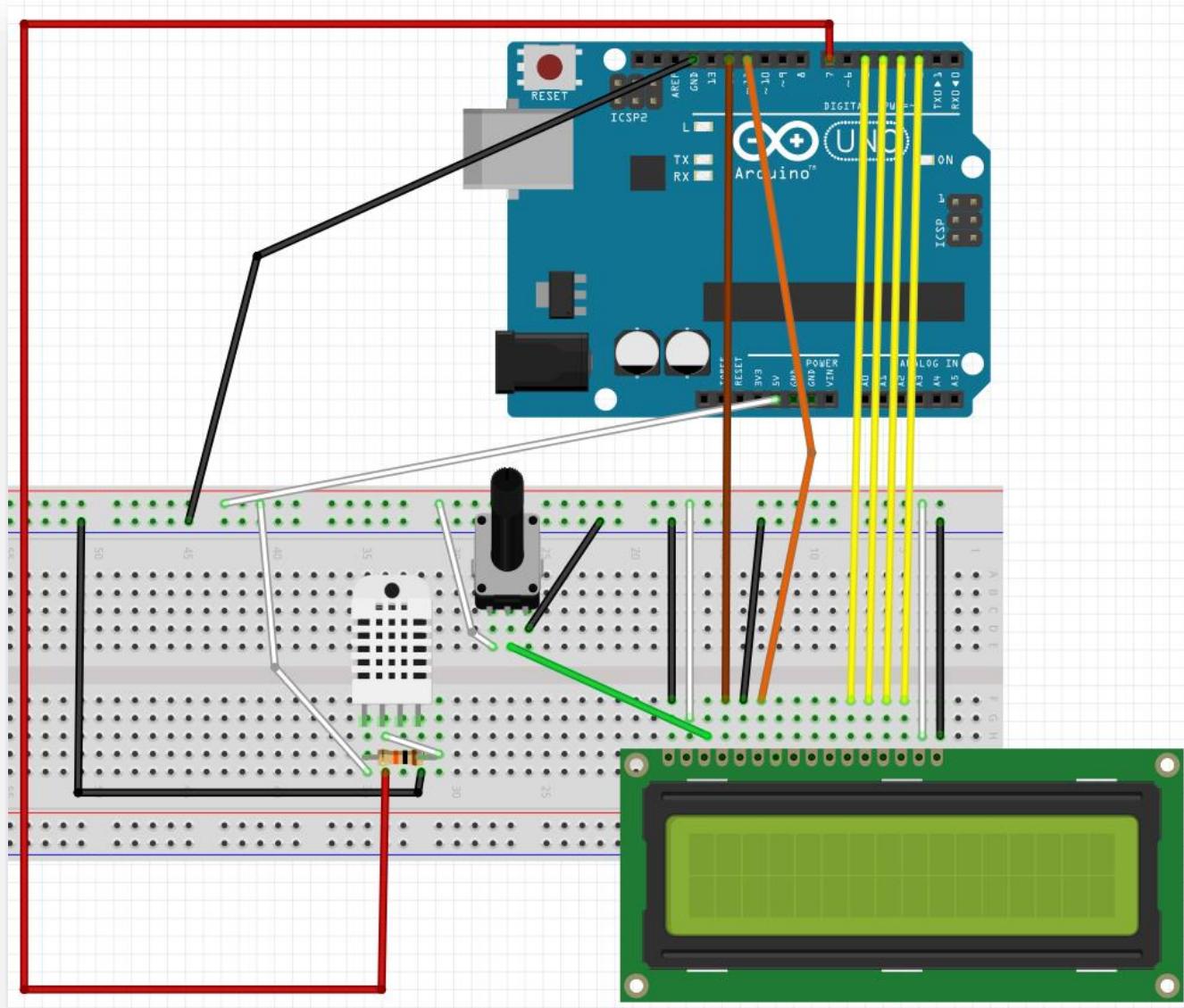


0.7.DIY4 Display of luminosity





0.8.DIY5 Display of Temperature & Humidity





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

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 [Get](#)

Mac OS X 10.7 Lion or newer

Linux 32 bits

Linux 64 bits

Linux ARM

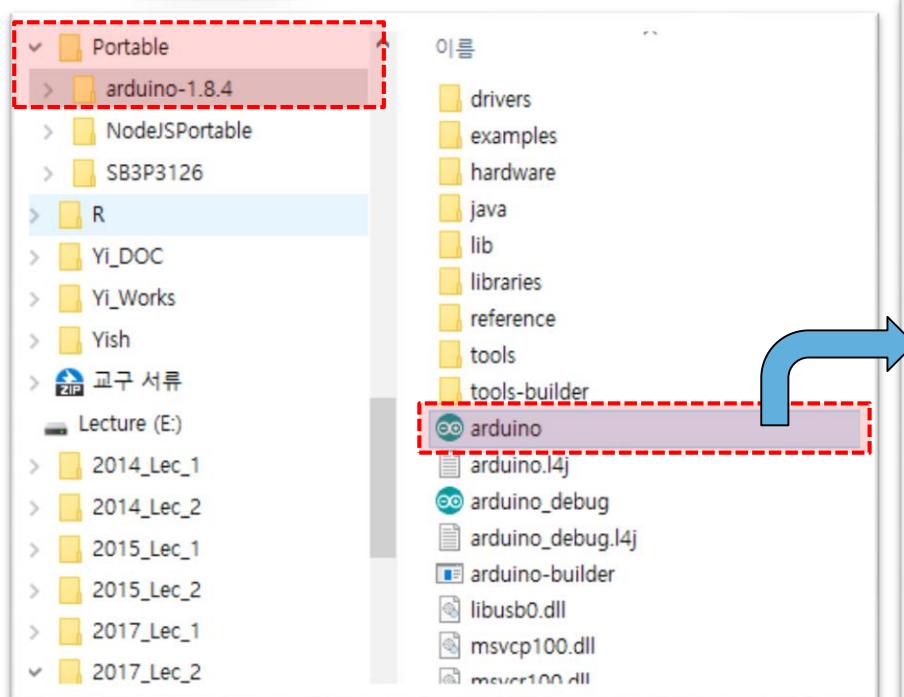
[Release Notes](#)

[Source Code](#)

[Checksums \(sha512\)](#)



A1.2 Arduino Portable (V1.8.4)



The screenshot shows the Arduino IDE interface. The title bar says 'sketch_sep27a | 아두이노 1.8.4'. The toolbar includes icons for save, upload, and upload selected. The code editor displays the following sketch:

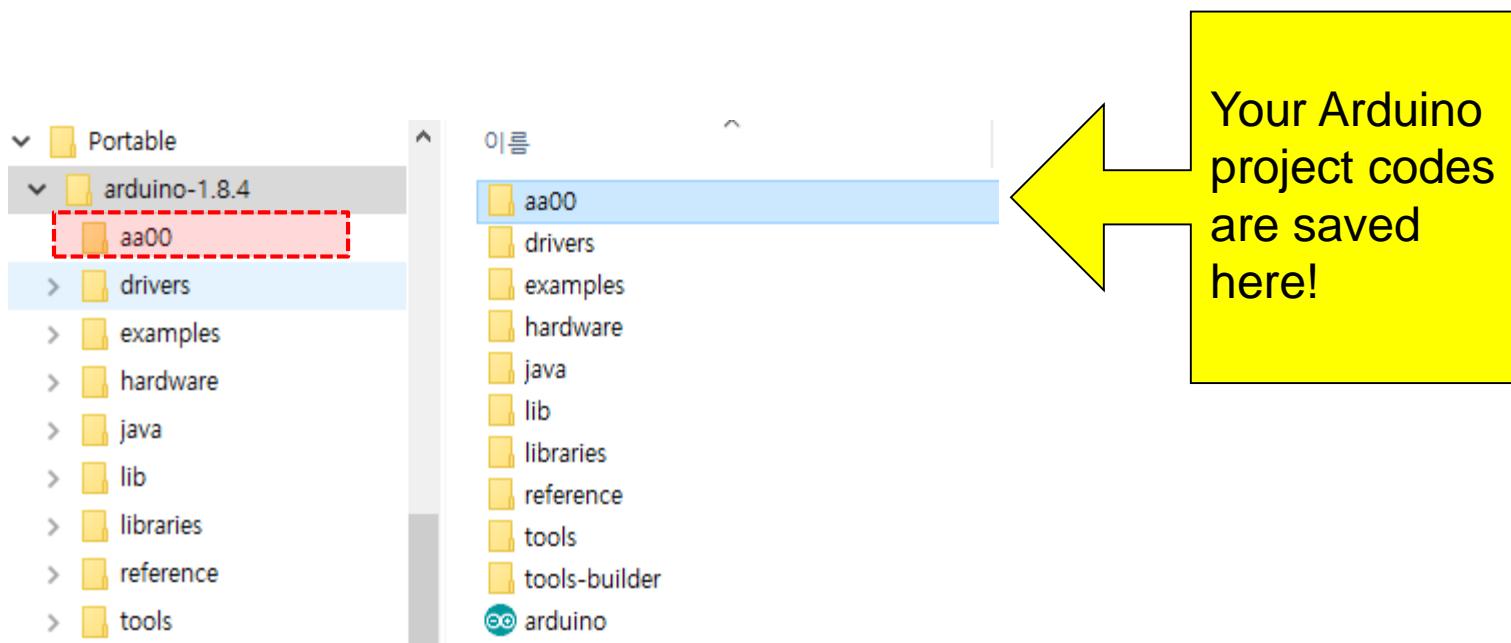
```
void setup() {  
  // put your setup code here, to run once:  
}  
  
void loop() {  
  // put your main code here, to run repeatedly:  
}
```

The status bar at the bottom right indicates 'Arduino/Genuine Uno on COM14'.

**Make folder `aann`
in Arduino folder**



A1.3 Arduino Portable (V1.8.4)



**Make folder `aann`
in Arduino folder**

Your Arduino
project codes
are saved
here!



A1.4 Arduino Portable IDE

The screenshot shows the Arduino Portable IDE interface with a sketch named "sketch_sep27a". The code editor displays the following code:

```
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 }  
10  
11
```

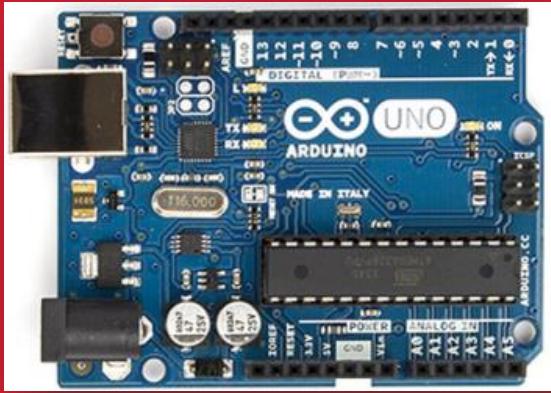
The first two lines of the `setup()` function and the entire `loop()` function are highlighted with red dashed boxes. The status bar at the bottom right indicates "Arduino/Genuino Uno on COM14".

A1.5 Arduino Portable IDE

The screenshot shows the Arduino Portable IDE interface. On the left, the main menu bar includes '파일' (File), '편집' (Edit), '스케치' (Sketch), '툴' (Tools), and '도움말' (Help). The '파일' tab is selected, showing options like '새 파일' (New), '열기...' (Open), '최근 파일 열기' (Recent Files), '스케치북' (Sketchbook), '예제' (Examples), '닫기' (Close), '저장' (Save), '다른 이름으로 저장...' (Save As), '페이지 설정' (Page Setup), '인쇄' (Print), '환경설정' (Preferences), and '종료' (Exit). The '환경설정' option is highlighted.

The central part of the screen displays the '환경설정' (Preferences) dialog box. The '환경설정' tab is selected, showing settings for the sketchbook location (D:\Portable\Arduino-1.8.4\aa00), editor font size (18), interface scale (100%), and various checkboxes for code compilation and upload. A note at the bottom states that changes take effect after restarting the Arduino IDE.

At the bottom right of the dialog box are '확인' (OK) and '취소' (Cancel) buttons.



LED



A2.0 LED control

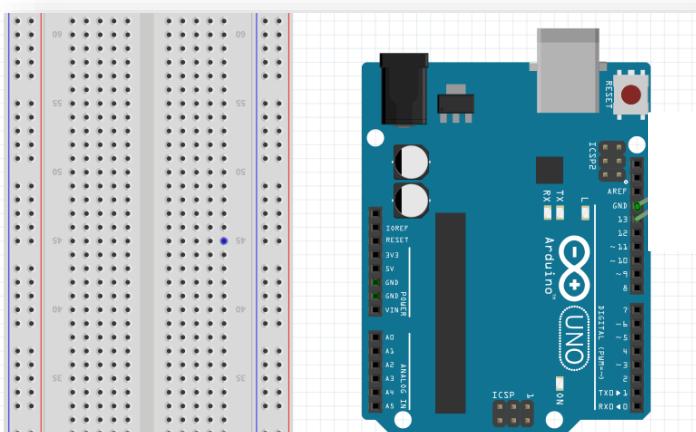
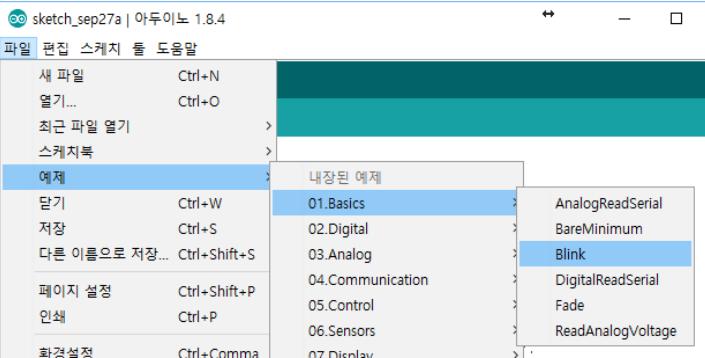
LED (Light Emitting Diode)

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





A2.1.1 Blink [digitalWrite()]

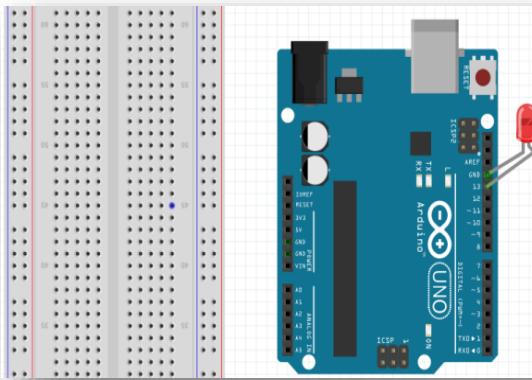


```
Blink $
```

```
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 [modified your code, save it]



**Connect LED to
D13 & GND**



AA00_Blink

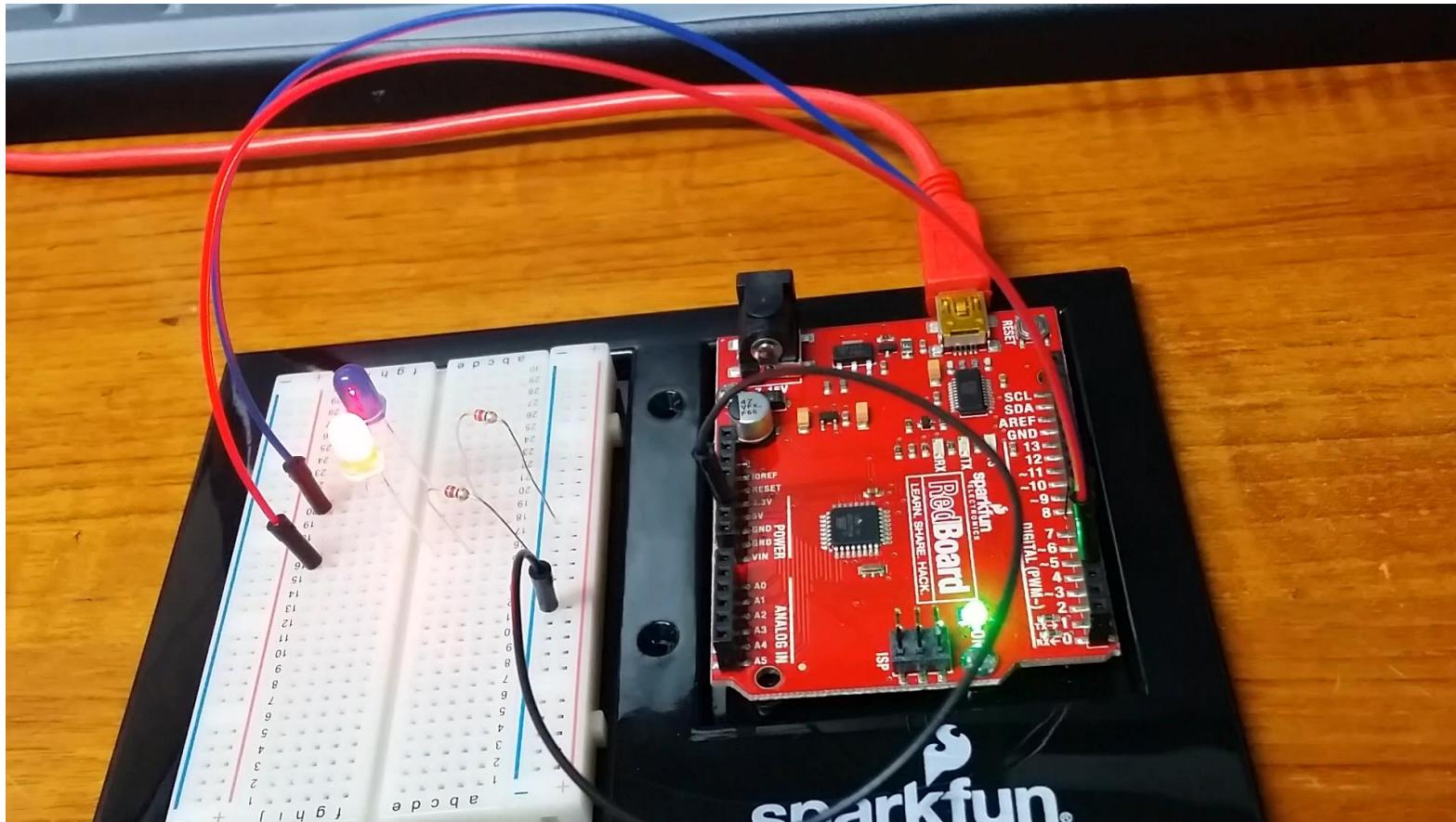
```
1 /*
2  * Blink by AAnn
3  *
4  * Turns an LED on for one second, then off for one second, repeatedly.
5 */
6
7 // the setup function runs once when you press reset or power the board
8 void setup() {
9     // initialize digital pin LED_BUILTIN as an output.
10    pinMode(pinNum, OUTPUT);
11 }
12
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
AAnn_Blink.png



A2.2 LED control

2.1 LED 교차 점멸



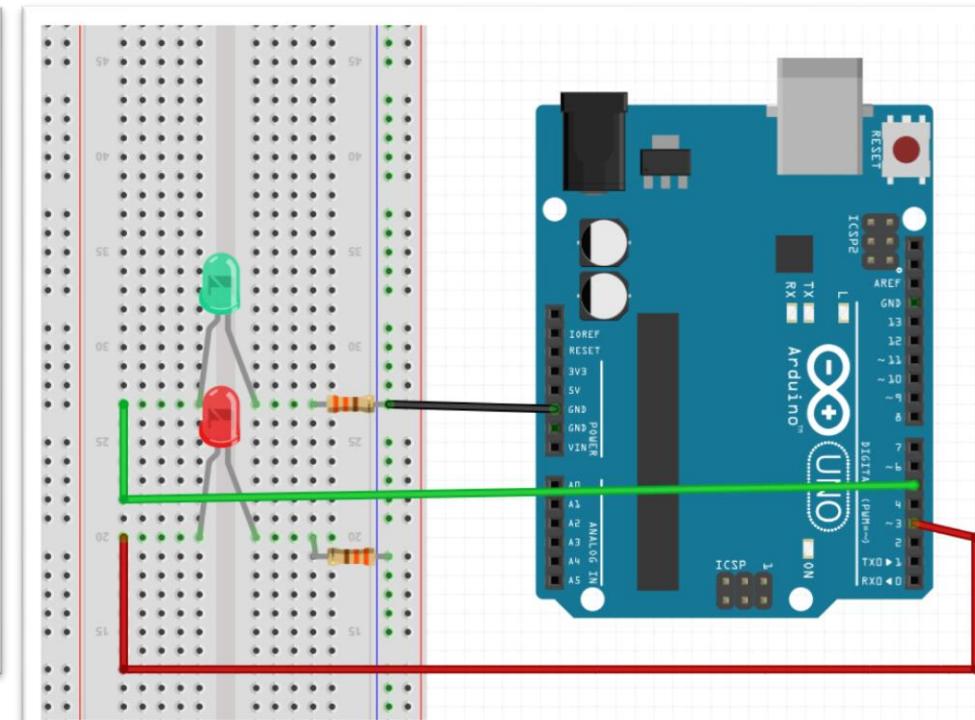
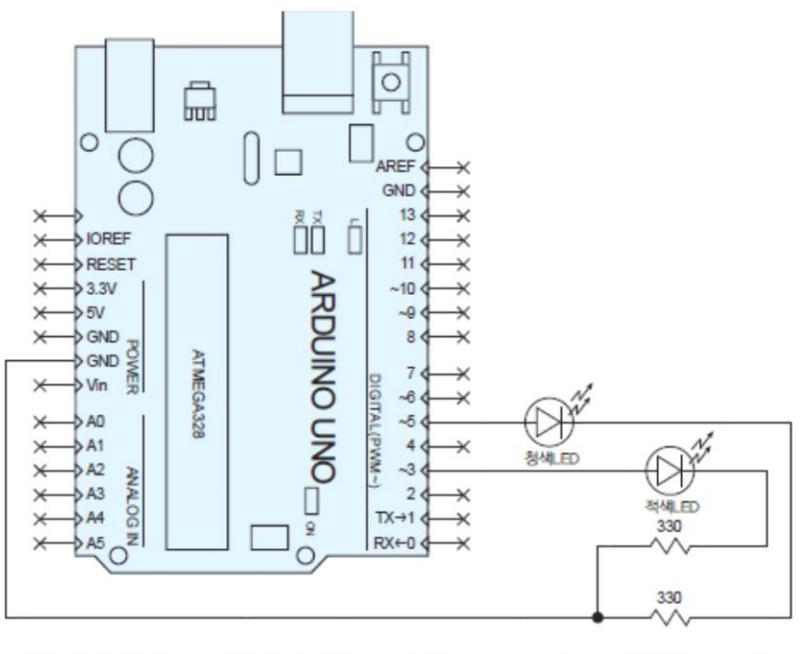


A2.2.1 LED control – 교차 점멸

LED 교차 점멸 (1/3)

실습목표 두 개의 LED를 0.1초 간격으로 교차하여 점멸시키자.

Hardware





A2.2.2 LED control – 교차 점멸

LED 교차 점멸 (2/3)

Commands

- `pinMode(핀번호, 설정)`

핀의 입출력 모드를 설정한다. ‘핀번호’에는 설정하고자 하는 핀의 번호와 ‘설정’에는 입력으로 사용하기 위해선 ‘INPUT’, 출력으로 사용하기 위해선 ‘OUTPUT’, 입력이며 풀업 사용시 ‘INPUT_PULLUP’을 설정한다.

- `digitalWrite(핀번호, 값)`

핀에 디지털 출력 (High or Low) 을 한다. ‘핀번호’에는 출력하고자 하는 핀의 번호를, ‘값’에는 ‘HIGH’ 혹은 ‘LOW’ 를 설정하여 High 혹은 Low 출력을 한다.

Sketch 구성

1. LED의 핀 번호를 설정한다.
2. `setup()`에서는 LED 출력으로 사용할 핀을 출력핀으로 설정한다.
3. `loop()`에서는 하나의 LED를 켜 후 일정시간이 지난 후에 소등하고, 다른 LED를 켠다.



A2.2.3 LED control – 교차 점멸

LED 교차 점멸 (3/3) - code

실습 결과 LED A와 B가 0.1초 단위로 교차하여 점멸한다.

AAnn_2Leds

```
1 /*  
2  LED 교차 점멸  
3 */  
4  
5 const int ledA    = 3;  
6 const int ledB    = 5;  
7  
8 void setup()  
9 {  
10  pinMode(ledA, OUTPUT);  
11  pinMode(ledB, OUTPUT);  
12 }
```

```
14 void loop()  
15 {  
16  digitalWrite(ledA,HIGH);  
17  digitalWrite(ledB,LOW);  
18  delay(100);  
19  digitalWrite(ledA,LOW);  
20  digitalWrite(ledB,HIGH);  
21  delay(100);  
22 }
```



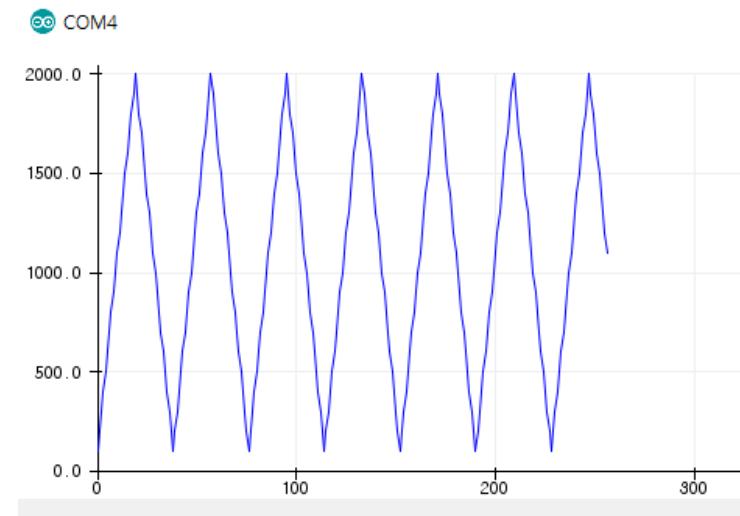
DIY: LED control – 교차 점멸

LED 교차 점멸 (도전 DIY)

응용 문제 점멸 주기가 0.1초부터 2초로 0.1초 단위로 증가하였다가 다시 반대로 2초부터 0.1초까지 감소하는 동작을 반복하는 스케치를 작성해 보자.
(hint: delay 명령어의 괄호 안의 숫자를 증감시킨다.)

delay = 1600 msec
delay = 1700 msec
delay = 1800 msec
delay = 1900 msec
delay = 2000 msec
delay = 1900 msec
delay = 1800 msec
delay = 1700 msec
delay = 1600 msec

delay = 500 msec
delay = 400 msec
delay = 300 msec
delay = 200 msec
delay = 100 msec
delay = 200 msec
delay = 300 msec
delay = 400 msec
delay = 500 msec

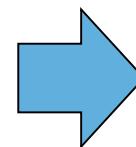




DIY: LED control – 교차 점멸

AAnn_2Leds \$

```
1 /*
2 LED 교차 점멸
3 */
4
5 const int ledA = 3;
6 const int ledB = 5;
7
8 void setup()
9 {
10 pinMode(ledA, OUTPUT);
11 pinMode(ledB, OUTPUT);
12 }
13
14 void loop()
15 {
16 digitalWrite(ledA,HIGH);
17 digitalWrite(ledB,LOW);
18 delay(100);
19 digitalWrite(ledA,LOW);
20 digitalWrite(ledB,HIGH);
21 delay(100);
```



```
6 const int ledA = 3;
7 const int ledB = 5;
8
9 int number = 1;
10 boolean flag = true;
11
12 void setup()
13 {
14 Serial.begin(9600);
15 pinMode(ledA, OUTPUT);
16 pinMode(ledB, OUTPUT);
17 }
18
19 void loop()
20 {
21 digitalWrite(ledA, HIGH);
22 digitalWrite(ledB, LOW);
23 delay(100 * number);
24 digitalWrite(ledA, LOW);
25 digitalWrite(ledB, HIGH);
26 Serial.print("delay = ");
27 Serial.print(100 * number);
28 Serial.println(" msec");
29 delay(100 * number);
30
31 if (flag) {
32 number++;
33 } else {
34 number--;
35 }
36
37 if (number >= 20) {
38 flag = false;
39 }
40 else if (number == 1) {
41 flag = true;
42 }
```

완성된 스케치 **code**를
AAnn_2Leds.ino
로 저장해서 제출.



A2.3.1 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.

A2.3.2 LED control – 밝기 조절: PWM

PWM (Pulse Width Modulation)

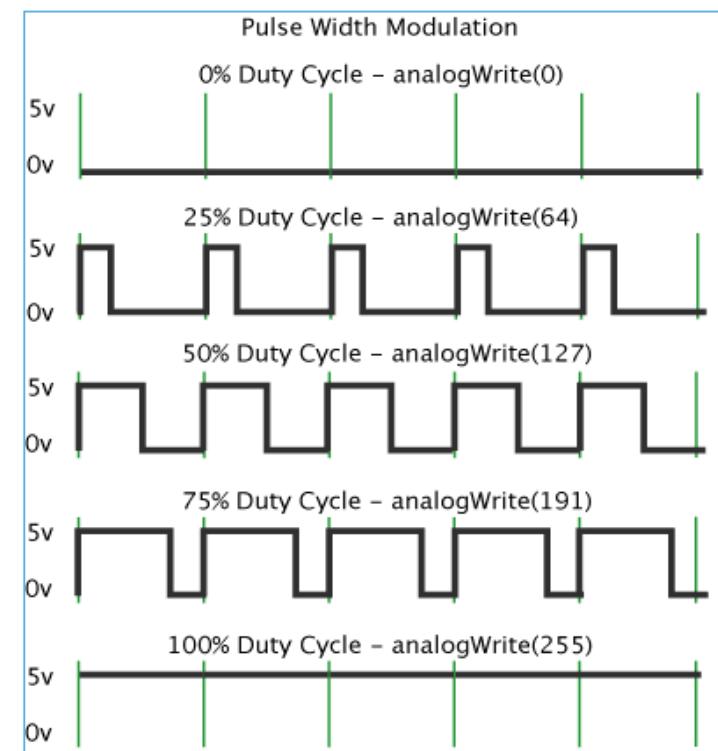
Using [analogWrite\(pin, pwm_value\)](#) function in fading an LED off and on.

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>

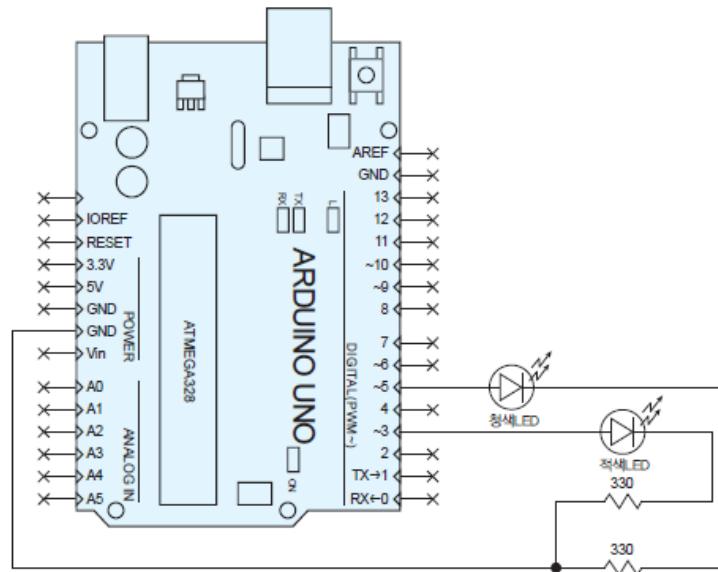


A2.3.3 LED control – 밝기 조절: PWM

LED 밝기 조절 (1/2)

- 실습목표**
1. 두 개의 LED의 밝기를 조절하자.
 2. 각각의 LED가 교차하여 밝아졌다 어두워졌다를 반복하도록 하자.

- Hardware**
1. 청색과 적색 LED의 Anode핀을 Arduino의 3번 5번 핀에 연결한다.
 2. Cathode핀에 330Ω저항을 연결하여 저항의 반대쪽은 Arduino의 GND에 연결한다.
 3. LED가 연결된 핀에 HIGH신호가 출력될 때 LED가 점등된다.





A2.3.4 LED control – 밝기 조절: PWM

LED 밝기 조절 (2/2)

Commands • `analogWrite(핀번호, 값)`

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

Sketch 구성 1. LED의 핀 번호를 설정한다.

2. `setup()`에서는 LED 출력으로 사용할 핀을 출력핀으로 설정한다.

3. 밝기를 저장할 변수를 설정한다.

4. 하나의 LED가 밝아질 때 다른 LED는 어두워져야 하므로 이를 조절할 변수를 설정한다.

5. `loop()`에서는 밝기와 밝기 변수 증감을 위한 변수를 조절하여 두 개의 LED를 교차 점멸시키는 동작을 반복한다.

실습 결과 LED A와 B가 밝기가 변화하며 점멸한다.

응용 문제 1. 네개의 다른 색깔의 LED를 Arduino에 연결한다.

2. 네개의 LED가 순서대로 디밍하는 스케치를 작성해보자.



A2.3.5 LED control – 밝기 조절: PWM

```
AAnn_Dimming
1 /*
2 LED 밝기 조절 (Dimming by PWM)
3 */
4
5 const int ledA = 3; //LED A를 3번핀에 연결
6 const int ledB = 5; //LED B를 5번핀에 연결
7 int brightness = 0; //밝기를 조절하기 위한 변수
8 int increment = 1; //밝기 변수 증감을 위한 변수
```

```
9
10 void setup()
11 {
12 // analogWrite 핀에는 별도의 설정이 불필요하다.
13 }
```

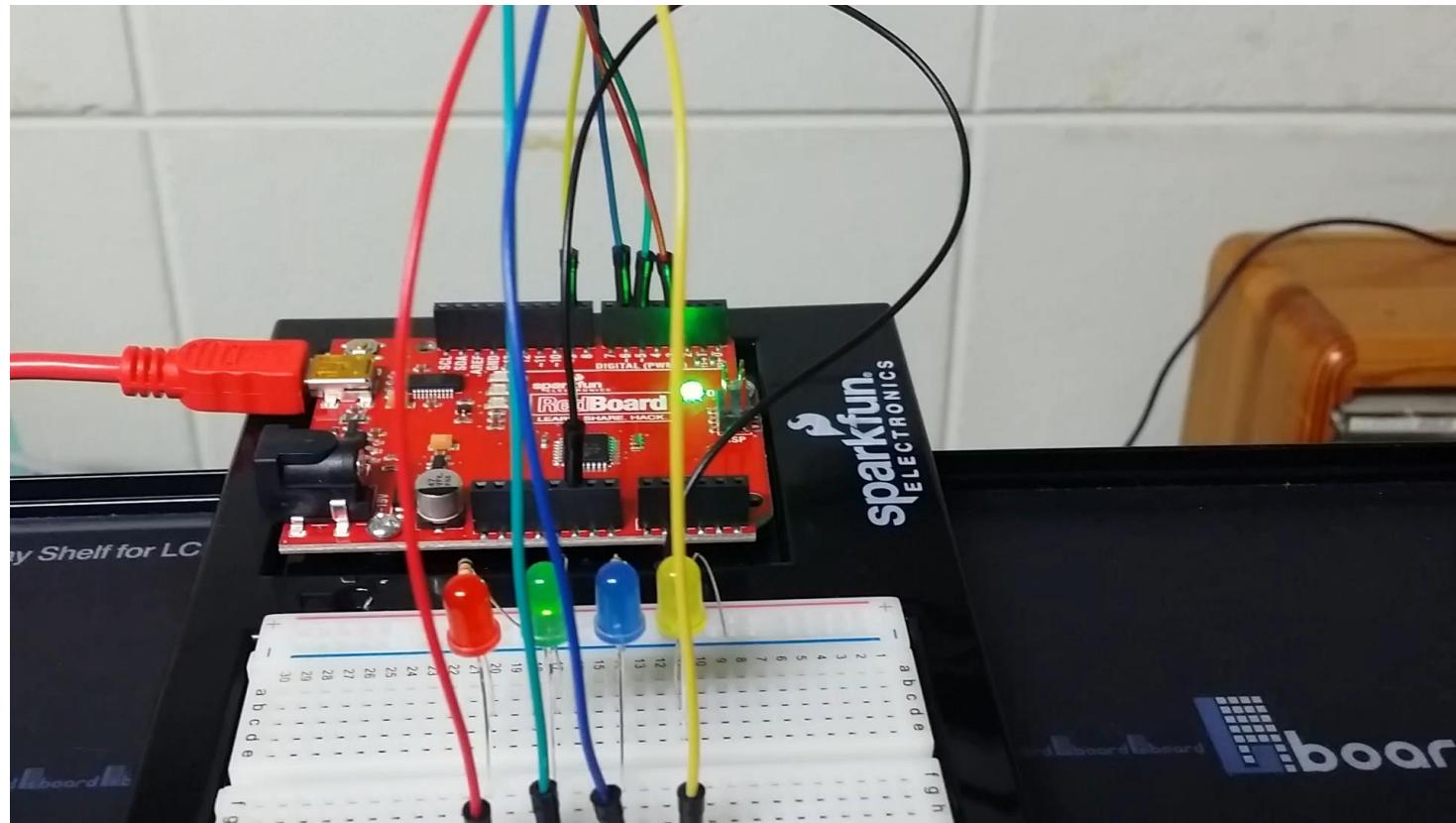
```
14
15 void loop()
16 {
17 analogWrite(ledA,brightness); // LED A 밝기 조절
18 analogWrite(ledB,255-brightness); // LED B 밝기 조절
19
20 brightness = brightness + increment; // 밝기 조절
21 if((brightness >= 255) || (brightness <= 0)) increment = -increment; // 밝기 변수 증감 방향 변경
22 delay(10); // 0.01 초간 지연
23 }
```

실습 결과

LED A와 B가 밝기가 변화하며 점멸한다.

A2.3.6 LED control – 밝기 조절: PWM

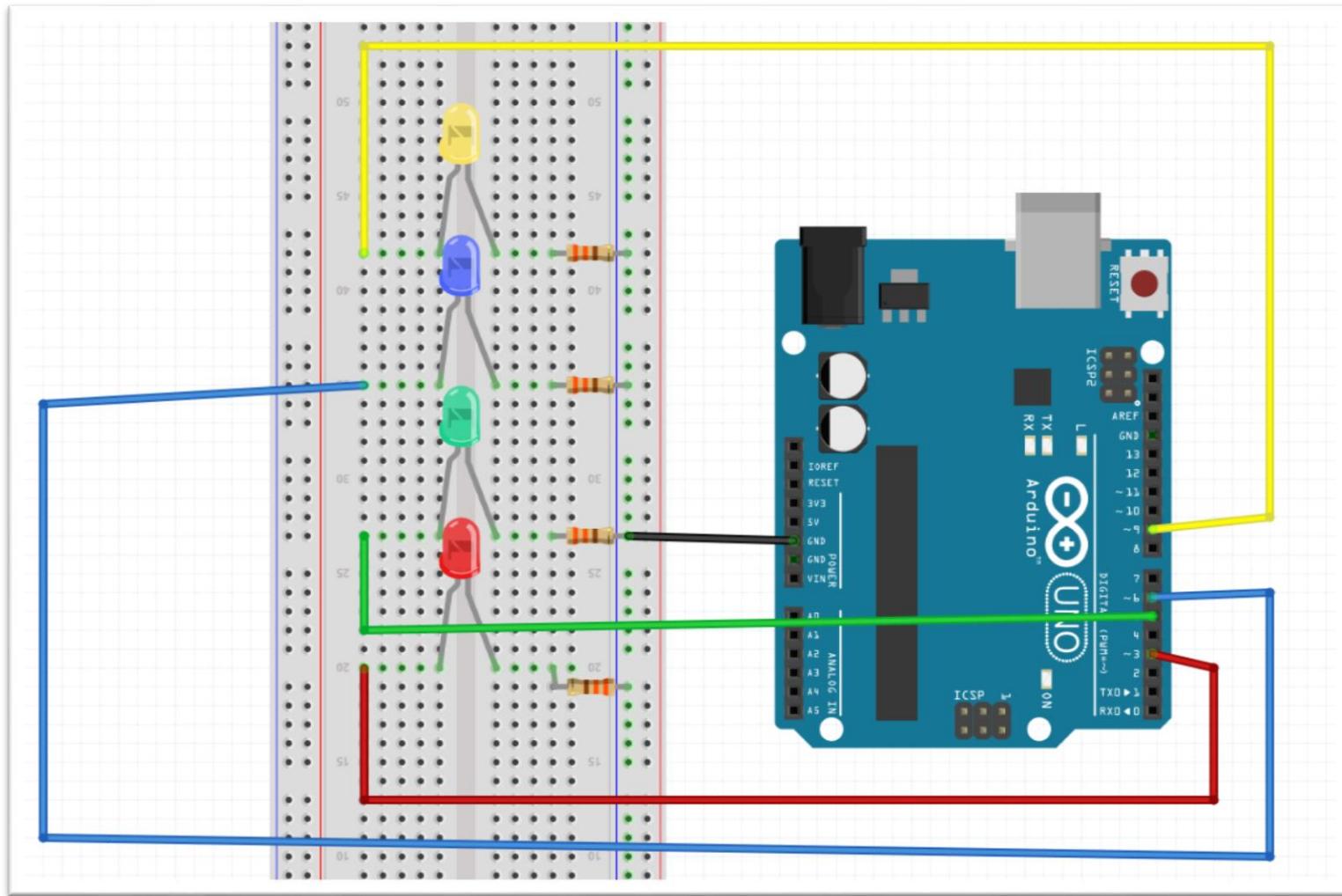
- DIY.
1. 네개의 다른 색깔의 LED를 Arduino에 연결한다.
 2. 네개의 LED가 순서대로 디밍하는 스케치를 작성해보자.





DIY – 밝기 조절: PWM

DIY. 1. 네 개의 다른 색깔의 LED를 Arduino에 연결한다. (pwm pin: 3,5,6,9)





DIY – 밝기 조절: PWM → code-1

```
1 /*  
2 Dimming 4 leds  
3 */  
4  
5 int ledR = 3; // LED connected to digital pin 3  
6 int ledG = 5;  
7 int ledB = 6;  
8 int ledY = 9;  
9  
10 int dimTime = 20;  
11  
12 void setup() {  
13 // nothing happens in setup  
14 }
```

```
16 void loop() {  
17 // fade in from min to max in increments of 5 points:  
18 for(int fadeValue = 0 ; fadeValue <= 255; fadeValue +=5) {  
19 // sets the value (range from 0 to 255):  
20 analogWrite(ledR, fadeValue);  
21 // wait for 30 milliseconds to see the dimming effect  
22 delay(dimTime);  
23 }  
24  
25 // fade out from max to min in increments of 5 points:  
26 for(int fadeValue = 255 ; fadeValue >= 0; fadeValue -=5) {  
27 // sets the value (range from 0 to 255):  
28 analogWrite(ledR, fadeValue);  
29 // wait for 30 milliseconds to see the dimming effect  
30 delay(dimTime);  
31 }  
32 // 각 led에 동일한 dimming code 적용  
33 for(int fadeValue = 0 ; fadeValue <= 255; fadeValue +=5) {  
34 // sets the value (range from 0 to 255):  
35 analogWrite(ledG, fadeValue);  
36 // wait for 30 milliseconds to see the dimming effect  
37 delay(dimTime);  
38 }
```



DIY – 밝기 조절: PWM → code-2

```
1 /*  
2 Dimming 4 leds  
3 */  
4  
5 int ledR = 3; // LED connected to digital pin 3  
6 int ledG = 5;  
7 int ledB = 6;  
8 int ledY = 9;  
9  
10 int dimTime = 20;  
11  
12 void setup() {  
13 // nothing happens in setup  
14 }
```

완성된 스케치 code를
[AAnn_4Leds.ino](#)
로 저장해서 제출.

```
16 void loop() {  
17 // fade ledR  
18 dimLed(ledR);  
19 // fade ledG  
20 dimLed(ledG);  
21 // fade ledB  
22 dimLed(ledB);  
23 // fade ledY  
24 dimLed(ledY);  
25 }  
26 void dimLed(int led) {  
27 // fade in from min to max in increments of 5 points:  
28 for(int fadeValue = 0 ; fadeValue <= 255; fadeValue +=5) {  
29 // sets the value (range from 0 to 255):  
30 analogWrite(led, fadeValue);  
31 // wait for 20 milliseconds to see the dimming effect  
32 delay(dimTime);  
33 }  
34 // fade out from max to min in increments of 5 points:  
35 for(int fadeValue = 255 ; fadeValue >= 0; fadeValue -=5) {  
36 // sets the value (range from 0 to 255):  
37 analogWrite(led, fadeValue);  
38 // wait for 20 milliseconds to see the dimming effect  
39 delay(dimTime);  
40 }  
41 }
```

각 led에 동일한 dimming code 적용

dimLed(int led) 함수 반복 사용

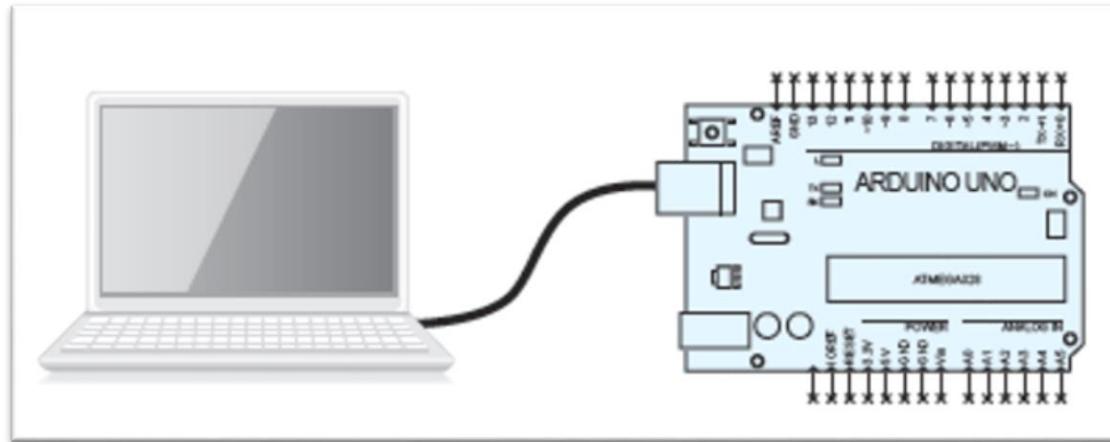


시리얼 통신

Serial
monitor &
plotter

A2.4 시리얼 통신

1. Arduino에서 컴퓨터로 데이터 전송하기
2. 변수 유형별로 컴퓨터에 전송하기





A2.4.1 Serial Monitor & Plotter

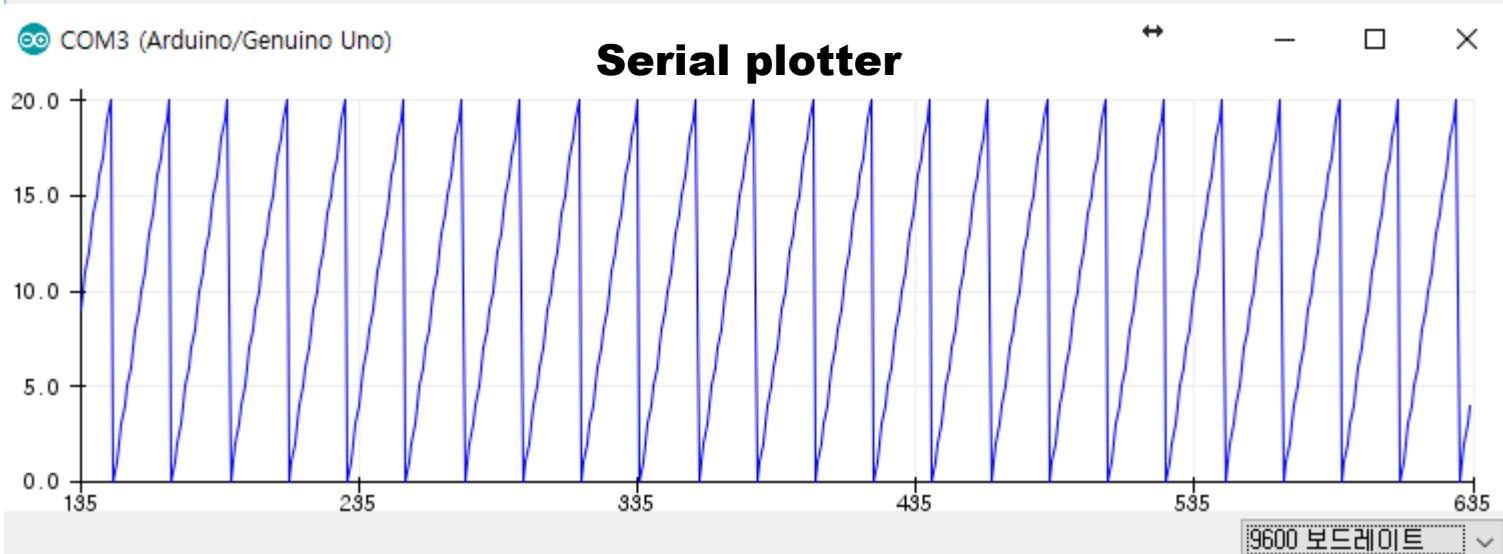
COM3 (Arduino/Genuino Uno)

14 sec
15 sec
16 sec
17 sec
18 sec
19 sec
20 sec
0 sec
1 sec
2 sec
3 sec
4 sec
5 sec
6 sec

자동 스크롤

line ending 없음 9600 보드레이트

Serial moniter
전송된 숫자와 문자를 모니터링



A2.4.2 시리얼 통신 (serial comm.)

시리얼 통신

UART (Universal Asynchronous
Receiver/Transmitter)

RS-232

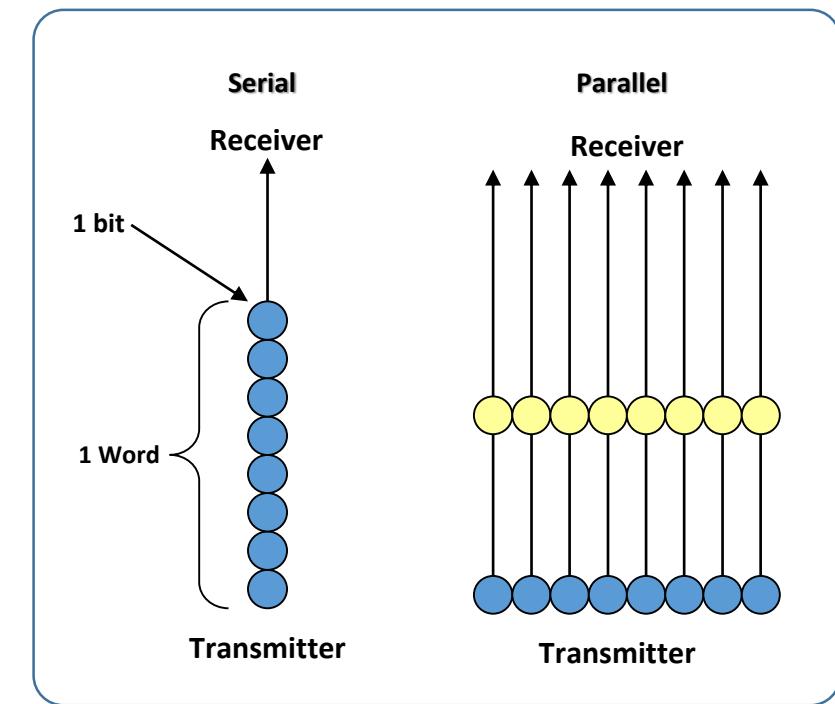
RS-422

RS-485

Arduino에서는 다음과 같은 목적으로 사용

Debugging : 프로그램의 오류를 수정하는 작업

데이터 통신 : Arduino와 컴퓨터 혹은 다른 장치와의 통신



A2.4.3 Arduino에서 컴퓨터로 데이터 전송하기

Arduino에서 컴퓨터로 변수와 문자열 전송하기 (1/3)

- 실습목표**
1. Arduino에서 문자열과 데이터를 시리얼 통신을 이용하여 컴퓨터로 전송한다.
 2. 전송할 데이터는 0부터 1초 간격으로 1씩 증가하는 숫자와 'sec'라는 문자열이다.
 3. Arduino IDE의 시리얼 모니터에서 이를 확인해 본다.

Hardware Arduino와 PC를 USB 케이블로 연결한다.

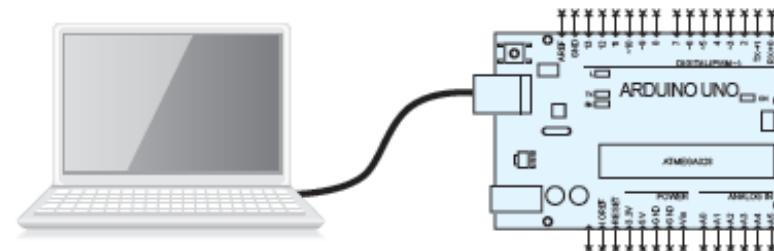


그림 2.1 Arduino와 PC와의 연결



A2.4.4 Arduino에서 컴퓨터로 데이터 전송하기

Arduino에서 컴퓨터로 변수와 문자열 전송하기 (2/3)

Commands

- `Serial.begin`(전송속도)

시리얼 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 시리얼 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.

- `delay`(지연시간 in ms)

지연시간에는 잠시 동작을 지연시키기 위한 값을 넣는다. 1/1000초 단위로 넣는다. 즉 1초를 지연시키기 위해선 1000의 값을 입력시킨다.



A2.4.5 Arduino에서 컴퓨터로 데이터 전송하기

Arduino에서 컴퓨터로 변수와 문자열 전송하기 (3/3)

Sketch 구성 1. 시리얼통신을 시작한다. ‘Serial.begin()’ 명령어로 할 수 있다.

2. 변수를 전송하기 위해 ‘Serial.print()’ 명령어를 사용하고,

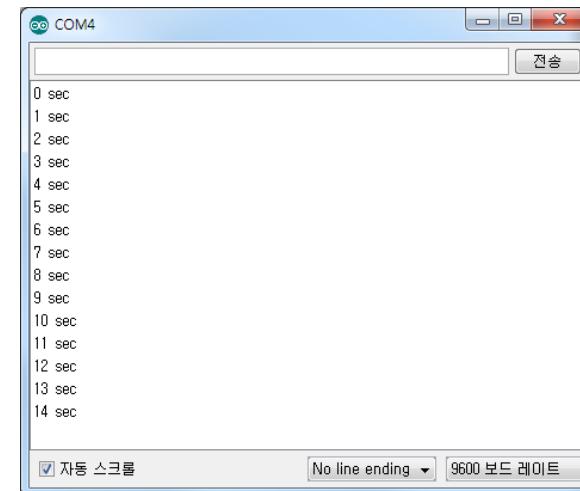
문자열 전송 후 줄 바꿈을 하기 위해서 ‘Serial.println()’ 명령어를 사용한다.

3. 루프를 1초마다 실행하기 위해서 ‘delay()’명령어로 시간지연을 시켜준다.

실습 결과 IDE의 시리얼 모니터를 실행시켜 Arduino에서 전송되는 메시지를 확인할 수 있다.

응용 문제 1. 2초, 5초 단위로 시간을 변경해 보자.

2. 자신만의 메시지를 출력해보자.





A2.4.6 Arduino에서 컴퓨터로 데이터 전송하기

The screenshot shows the Arduino IDE interface. On the left, the code for 'AAnn_Serial_Comm' is displayed:

```
1 /*  
2 Arduino에서 컴퓨터로 변수와 문자열 전송하기  
3 */  
4  
5 int number = 0; // -32768~32767 범위의 변수 number 설정, 초기값은 0  
6  
7 void setup() {  
8   Serial.begin(9600); // 9600bps로 시리얼 통신 설정  
9 }  
10  
11 void loop() {  
12   Serial.print(number); // number 변수값 출력  
13   Serial.println(" sec"); // " sec"를 출력 후 줄 바꿈  
14   delay(1000); // 1초동안 지연시킨다.  
15   number++; // number 변수값을 하나 증가시킨다.  
16 }  
17
```

The lines from 12 to 13 are highlighted with a red dashed box. To the right of this box, a green box contains the text '숫자와 문자열을 전송' (Transmitting numbers and strings).

On the right side of the interface, a terminal window titled 'COM4' shows the output of the serial communication:

```
24 sec  
25 sec  
26 sec  
27 sec  
28 sec  
29 sec  
30 sec  
31 sec  
32 sec  
33 sec  
34 sec  
35 sec  
36 sec  
37 sec
```

At the bottom of the terminal window, there is a message about memory usage:

스케치는 프로그램 저장 공간 1902 바이트(5%)를 사용. 최대 32256 바이트.
전역 변수는 동적 메모리 194바이트(9%)를 사용, 1854바이트의 지역변수가 남음. 최대는 2048 바이트.

At the very bottom of the interface, it says 'Arduino/Genuine Uno on COM4'.



DIY: Sawtooth wave

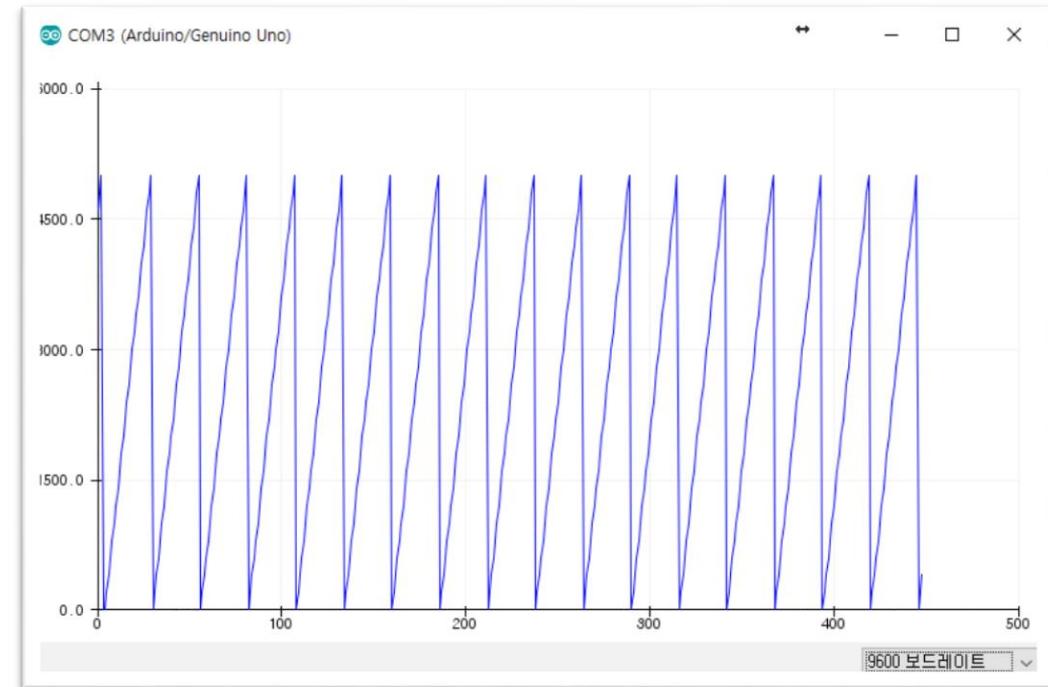
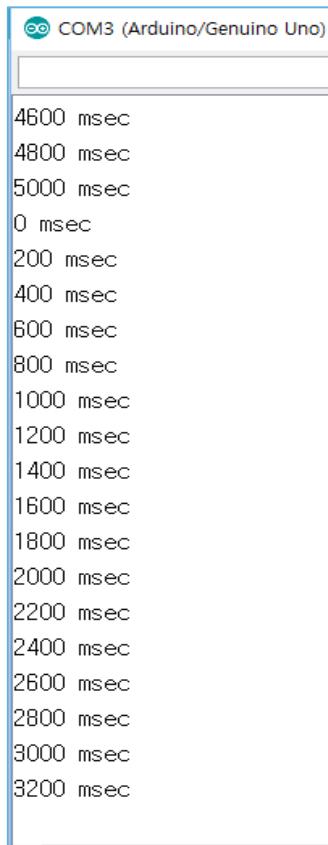
[DIY]

delay를 0.2초로 설정후

5초마다 number를 초기화하여

시리얼플로터로 톱니파를 발생.

시간은 ms로 계산해서 출력



Save as
AAnn_Sawtooth.png



[DIY] sawtooth signal : Code-1

```
AAnn_Sawtooth
4 */
5
6 int number = 0;           // -32768~32767 범위의 변수 number 설정, 초기값은 0
7
8 void setup() {
9     Serial.begin(9600);    // 9600bps로 시리얼 통신 설정
10 }
11
12 void loop() {
13     Serial.print(number*200); // number 변수값 출력
14     Serial.println(" msec"); // " msec"를 출력 후 줄 바꿈
15     delay(200);           // 0.2초동안 지연시킨다.
16     number++;             // number 변수값을 하나 증가시킨다.
17
18     if (number > 25) {
19         number = 0;
20     }
}
```

업로드 완료.

스케치는 프로그램 저장 공간 1932 바이트(5%)를 사용. 최대 32256 바이트.

전역 변수는 동적 메모리 196바이트(9%)를 사용, 1852바이트의 지역변수가 남음. 최대는 2048 바이트.

COM4

5000 msec

0 msec

200 msec

400 msec

600 msec

800 msec

1000 msec

1200 msec

1400 msec

1600 msec

1800 msec

2000 msec

2200 msec

2400 msec

자동 스크롤

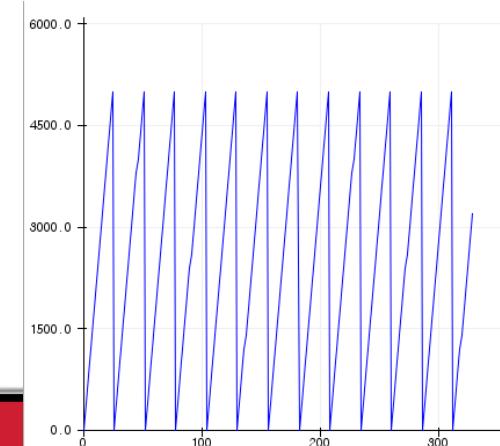


[DIY] sawtooth signal : Code-2

```
AA00_Sawtooth
1 /*
2 Sawtooth wave
3 Arduino에서 컴퓨터로 변수와 문자열 전송하기
4 */
5
6 int number = 0;           // -32768~32767 범위의 변수 number 설정, 초기값은 0
7
8 void setup() {
9     Serial.begin(9600);    // 9600bps로 시리얼 통신 설정
10 }
11
12 void loop() {
13     Serial.print(number); // number 변수값 출력
14     Serial.println(" msec"); // " msec"를 출력 후 줄 바꿈
15     delay(200);           // 0.2초동안 지연시킨다.
16     //number++;           // number 변수값을 하나 증가시킨다.
17
18     if (number < 5000) {
19         number += 200;
20     } else {
21         number = 0;
22     }
23 }
```

COM4

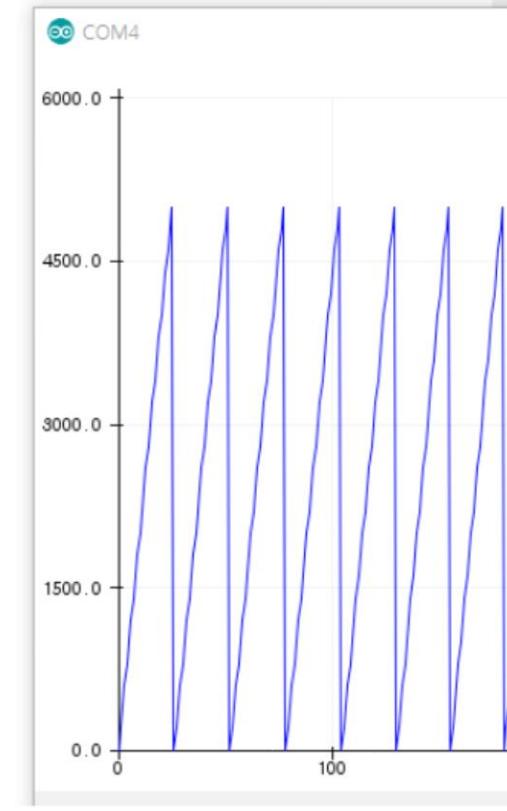
3800 msec
4000 msec
4200 msec
4400 msec
4600 msec
4800 msec
5000 msec
0 msec
200 msec
400 msec
600 msec
800 msec
1000 msec
1200 msec
1400 msec
1600 msec
1800 msec
2000 msec
2200 msec
2400 msec
2600 msec
2800 msec
3000 msec
3200 msec





[DIY] sawtooth signal : Code-3

```
AA00_Sawtooth_2
1 /*
2 Sawtooth wave
3 Arduino에서 컴퓨터로 변수와 문자열 전송하기
4 */
5
6 int number = 0;           // -32768~32767 범위의 변수 number 설정, 초기값은 0
7
8 void setup() {
9     Serial.begin(9600);    // 9600bps로 시리얼 통신 설정
10 }
11
12 void loop() {
13     Serial.print(number); // number 변수값 출력
14     Serial.println(" msec"); // " msec"를 출력 후 줄 바꿈
15     delay(200);           // 0.2초동안 지연시킨다.
16     number += 200;         // number 변수값을 200 증가시킨다.
17
18     if (number > 5000) {
19         number = 0;
20     }
21 }
```



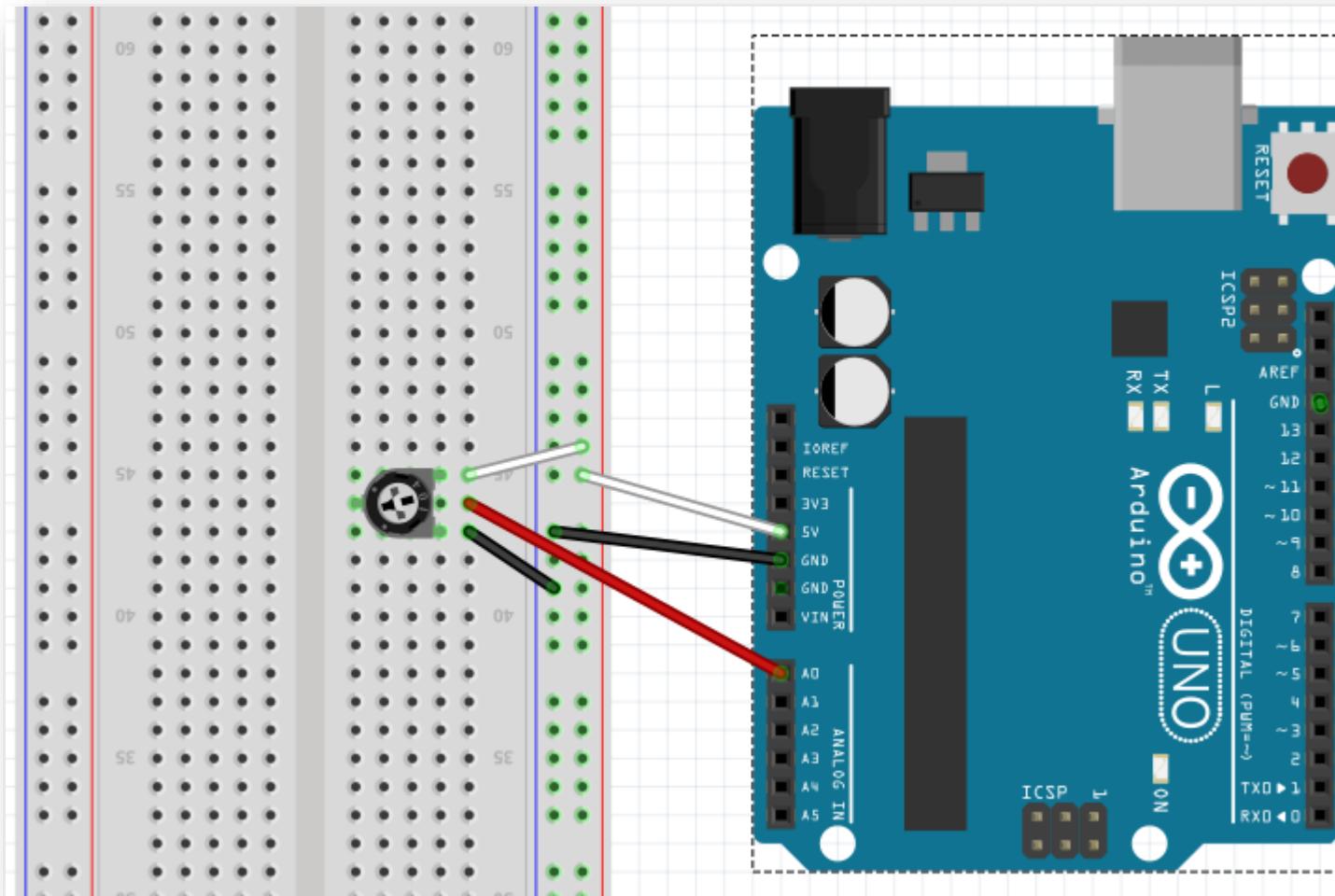


Analog Signal



A2.5.1 AnalogReadSerial (circuit)

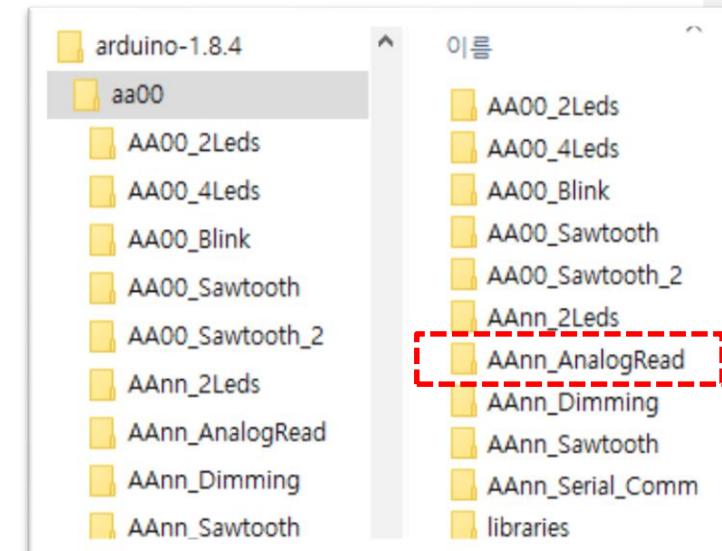
Standard potentiometer (가변 저항기)





A2.5.2 AnalogReadSerial (code)

```
AAnn_AnalogRead
1 /*
2  * AnalogReadSerial
3 *
4  * Reads an analog input on pin 0, prints the result to the Serial Monitor.
5  * Attach the center pin of a potentiometer to pin A0,
6  * and the outside pins to +5V and ground.
7 */
8
9 // the setup routine runs once when you press reset:
10 void setup() {
11     // initialize serial communication at 9600 bits per second:
12     Serial.begin(9600);
13 }
14
15 // the loop routine runs over and over again forever:
16 void loop() {
17     // read the input on analog pin 0:
18     int sensorValue = analogRead(A0);
19     // print out the value you read:
20     Serial.println(sensorValue);
21     delay(1);          // delay in between reads for stability
22 }
```



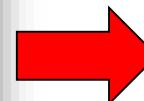


A2.5.3 AnalogReadSerial

Serial monitor

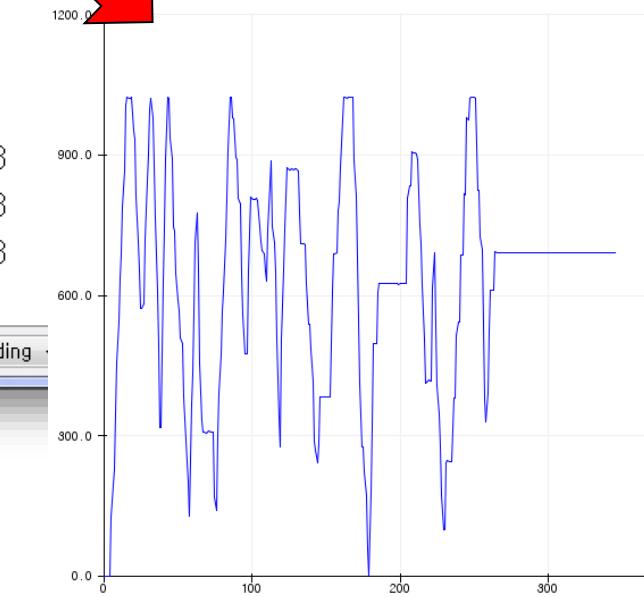
COM7 (Arduino/Genuino Uno)

0
0
121
584
971
1023
1023
1023
1023
896
388
192
0
0



Present value (0 ~ 1023) : 0
Present value (0 ~ 1023) : 0
Present value (0 ~ 1023) : 6
Present value (0 ~ 1023) : 14
Present value (0 ~ 1023) : 82
Present value (0 ~ 1023) : 199
Present value (0 ~ 1023) : 300
Present value (0 ~ 1023) : 429
Present value (0 ~ 1023) : 525
Present value (0 ~ 1023) : 634
Present value (0 ~ 1023) : 819
Present value (0 ~ 1023) : 989
Present value (0 ~ 1023) : 1023
Present value (0 ~ 1023) : 1023
Present value (0 ~ 1023) : 1023

자동 스크롤 No line ending





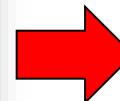
A2.5.4 ReadAnalogVoltage

Serial monitor

COM7 (Arduino/Genuino Uno)

```
Present value (0 ~ 1023) : 0
Present value (0 ~ 1023) : 0
Present value (0 ~ 1023) : 6
Present value (0 ~ 1023) : 14
Present value (0 ~ 1023) : 82
Present value (0 ~ 1023) : 199
Present value (0 ~ 1023) : 300
Present value (0 ~ 1023) : 429
Present value (0 ~ 1023) : 525
Present value (0 ~ 1023) : 634
Present value (0 ~ 1023) : 819
Present value (0 ~ 1023) : 989
Present value (0 ~ 1023) : 1023
Present value (0 ~ 1023) : 1023
Present value (0 ~ 1023) : 1023
```

자동 스크롤 No line ending 9600 보드 레이트



COM4

```
AA00, Present voltage (0.0 ~ 5.0) : 5.00
AA00, Present voltage (0.0 ~ 5.0) : 3.68
AA00, Present voltage (0.0 ~ 5.0) : 2.42
AA00, Present voltage (0.0 ~ 5.0) : 1.37
AA00, Present voltage (0.0 ~ 5.0) : 0.00
AA00, Present voltage (0.0 ~ 5.0) : 0.00
AA00, Present voltage (0.0 ~ 5.0) : 0.00
AA00, Present voltage (0.0 ~ 5.0) : 0.88
AA00, Present voltage (0.0 ~ 5.0) : 1.47
AA00, Present voltage (0.0 ~ 5.0) : 2.11
AA00, Present voltage (0.0 ~ 5.0) : 2.79
AA00, Present voltage (0.0 ~ 5.0) : 3.38
AA00, Present voltage (0.0 ~ 5.0) : 3.99
AA00, Present voltage (0.0 ~ 5.0) : 4.91
AA00, Present voltage (0.0 ~ 5.0) : 5.00
AA00, Present voltage (0.0 ~ 5.0) : 5.00
AA00, Present voltage (0.0 ~ 5.0) : 4.68
AA00, Present voltage (0.0 ~ 5.0) : 3.88
AA00, Present voltage (0.0 ~ 5.0) : 3.35
```



A2.5.5 ReadAnalogVoltage

Hint code

```
/*
ReadAnalogVoltage
Reads an analog input on pin 0, converts it to voltage, and prints the result to the serial monitor.
Graphical representation is available using serial plotter (Tools > Serial Plotter menu)
Attach the center pin of a potentiometer to pin A0, and the outside pins to +5V and ground.

This example code is in the public domain.
*/

// the setup routine runs once when you press reset:
void setup() {
  // initialize serial communication at 9600 bits per second:
  Serial.begin(9600);
}

// the loop routine runs over and over again forever:
void loop() {
  // read the input on analog pin 0:
  int sensorValue = analogRead(A0);
  // Convert the analog reading (which goes from 0 - 1023) to a voltage (0 - 5V):
  float voltage = sensorValue * (5.0 / 1023.0);
  // print out the value you read:
  Serial.println(voltage);
}
```



A2.5.6 ReadAnalogVoltage

Hint code

AA00_AnalogRead

```
1 /*  
2  AnalogReadSerial  
3  
4  Reads an analog input on pin 0, prints the result to the Serial Monitor.  
5  Attach the center pin of a potentiometer to pin A0,  
6  and the outside pins to +5V and ground.  
7 */  
8  
9 // the setup routine runs once when you press reset:  
10 void setup() {  
11   // initialize serial communication at 9600 bits per second:  
12   Serial.begin(9600);  
13 }  
14  
15 // the loop routine runs over and over again forever:  
16 void loop() {  
17   // read the input on analog pin 0:  
18   int sensorValue = analogRead(A0);  
19   //float voltage = map(sensorValue, 0, 1023, 0.0, 5.0); // map 0~1023 to 0~5  
20   float voltage = sensorValue*(5.0/1023.0);  
21   // print out the value you read:  
22   Serial.print("AA00, Present voltage (0.0 ~ 5.0) : ");  
23   Serial.println(voltage);  
24   delay(500);      // delay in between reads for stability  
25 }
```

COM4

```
AA00, Present voltage (0.0 ~ 5.0) : 5.00  
AA00, Present voltage (0.0 ~ 5.0) : 5.00  
AA00, Present voltage (0.0 ~ 5.0) : 3.91  
AA00, Present voltage (0.0 ~ 5.0) : 2.76  
AA00, Present voltage (0.0 ~ 5.0) : 1.59  
AA00, Present voltage (0.0 ~ 5.0) : 0.00  
AA00, Present voltage (0.0 ~ 5.0) : 0.00  
AA00, Present voltage (0.0 ~ 5.0) : 0.00  
AA00, Present voltage (0.0 ~ 5.0) : 0.81  
AA00, Present voltage (0.0 ~ 5.0) : 1.89  
AA00, Present voltage (0.0 ~ 5.0) : 2.92  
AA00, Present voltage (0.0 ~ 5.0) : 3.57  
AA00, Present voltage (0.0 ~ 5.0) : 4.29  
AA00, Present voltage (0.0 ~ 5.0) : 5.00  
AA00, Present voltage (0.0 ~ 5.0) : 4.99  
AA00, Present voltage (0.0 ~ 5.0) : 4.62  
AA00, Present voltage (0.0 ~ 5.0) : 3.21  
AA00, Present voltage (0.0 ~ 5.0) : 1.82  
AA00, Present voltage (0.0 ~ 5.0) : 1.80  
AA00, Present voltage (0.0 ~ 5.0) : 1.80  
AA00, Present voltage (0.0 ~ 5.0) : 1.80
```

자동 스크롤



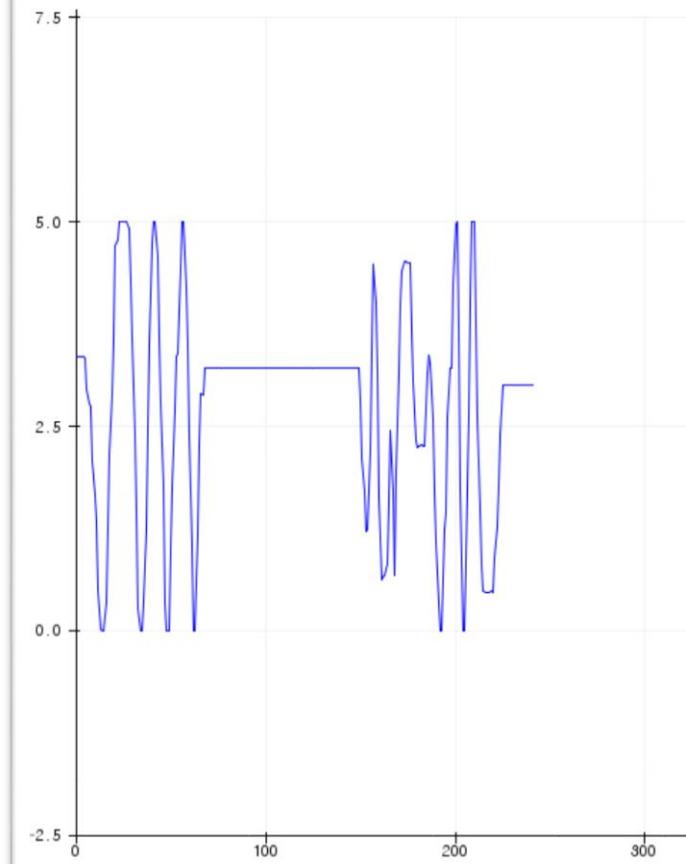
A2.5.7 ReadAnalogVoltage

Result

COM4

```
%A00, Present voltage (0.0 ~ 5.0) : 5.00  
%A00, Present voltage (0.0 ~ 5.0) : 3.68  
%A00, Present voltage (0.0 ~ 5.0) : 2.42  
%A00, Present voltage (0.0 ~ 5.0) : 1.37  
%A00, Present voltage (0.0 ~ 5.0) : 0.00  
%A00, Present voltage (0.0 ~ 5.0) : 0.00  
%A00, Present voltage (0.0 ~ 5.0) : 0.00  
%A00, Present voltage (0.0 ~ 5.0) : 0.88  
%A00, Present voltage (0.0 ~ 5.0) : 1.47  
%A00, Present voltage (0.0 ~ 5.0) : 2.11  
%A00, Present voltage (0.0 ~ 5.0) : 2.79  
%A00, Present voltage (0.0 ~ 5.0) : 3.38  
%A00, Present voltage (0.0 ~ 5.0) : 3.99  
%A00, Present voltage (0.0 ~ 5.0) : 4.91  
%A00, Present voltage (0.0 ~ 5.0) : 5.00  
%A00, Present voltage (0.0 ~ 5.0) : 5.00  
%A00, Present voltage (0.0 ~ 5.0) : 4.68  
%A00, Present voltage (0.0 ~ 5.0) : 3.88  
%A00, Present voltage (0.0 ~ 5.0) : 3.35
```

COM4



Save as

AAnn_AnalogVoltage.png

Save as

AAnn_AnalogVoltage_Plot.png



[Practice]

◆ [wk05]

- Arduino basic circuits
- Complete your project
- Upload file name : AAnn_Rpt04.zip

wk05 : Practice-04 : AAnn_Rpt04.zip

◆ [Target of this week]

- Complete your projects
- Save your outcomes and compress 4 figures and 2 codes.

제출파일명 : **AAnn_Rpt04.zip**

- 압축할 파일들

- ① **AAnn_Blink.png**
- ② **AAnn_2Leds.ino**
- ③ **AAnn_4Leds.ino**
- ④ **AAnn_Sawtooth.png**
- ⑤ **AAnn_AnalogVoltage.png**
- ⑥ **AAnn_AnalogVoltage_Plot.png**

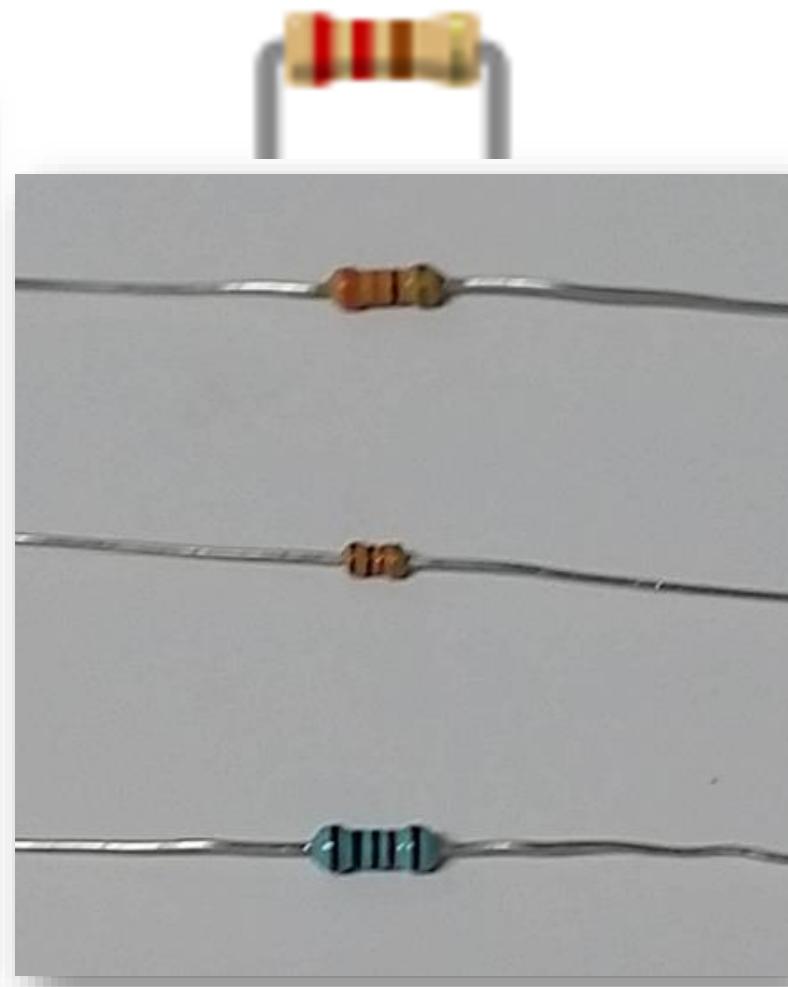


[참고 : 저항 값 읽기]

Carbonfilm resistor

sm2k (c) 2006

Color	First	Second	Third	Multiplier	Tolerance
Black	0	0	0	x1	
Brown	1	1	1	x10	1%
Red	2	2	2	x100	2%
Orange	3	3	3	x1000	
Yellow	4	4	4	x10 000	
Green	5	5	5	x100 000	0,50%
Blue	6	6	6	x1 000 000	0,25%
Violette	7	7	7	x10 000 000	0,10%
Gray	8	8	8		
White	9	9	9		
Silver				x0,01	10%
Gold				x0,1	5%

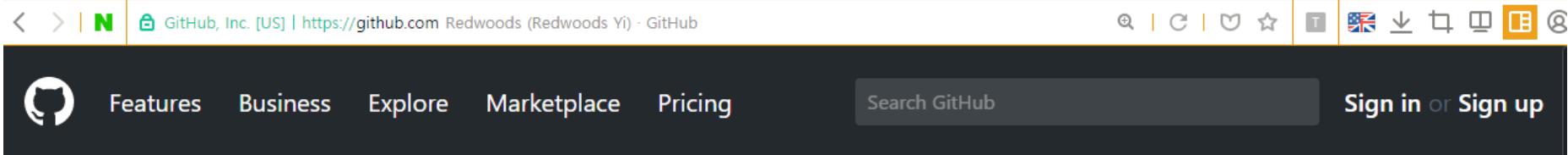


Lecture materials

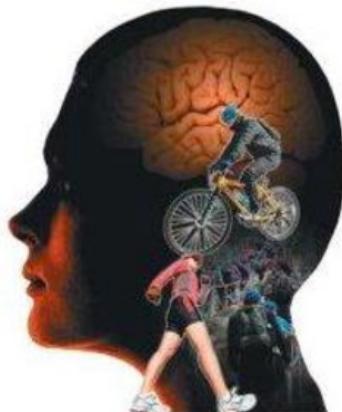


● References & good sites

- ✓ <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 of the URL are icons for search, refresh, notifications, and user profile. Below the bar, the GitHub logo is on the left, followed by links for "Features", "Business", "Explore", "Marketplace", and "Pricing". On the far right, there are "Sign in or Sign up" buttons. A search bar labeled "Search GitHub" is positioned between the navigation bar and the main content area.



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

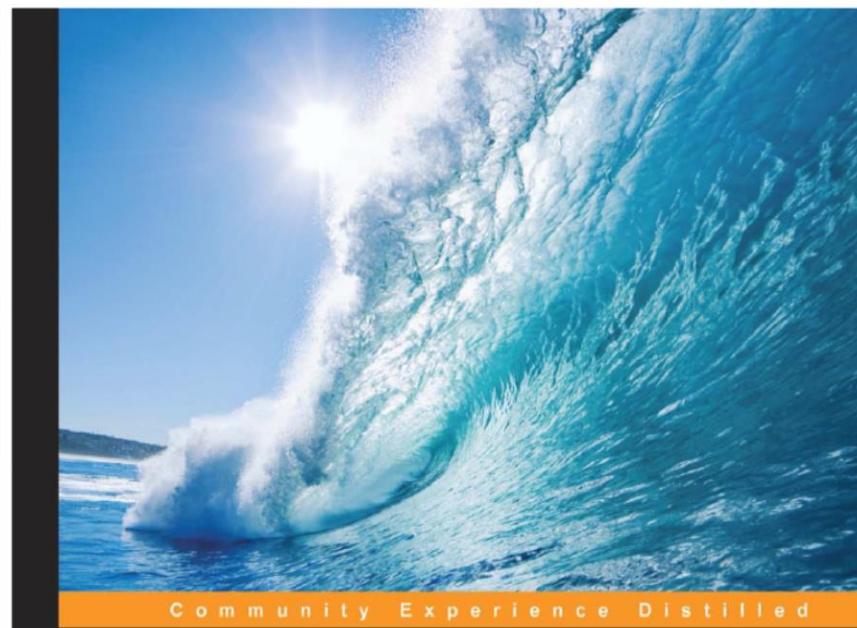
[hw-coding](#)

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

 Arduino



References

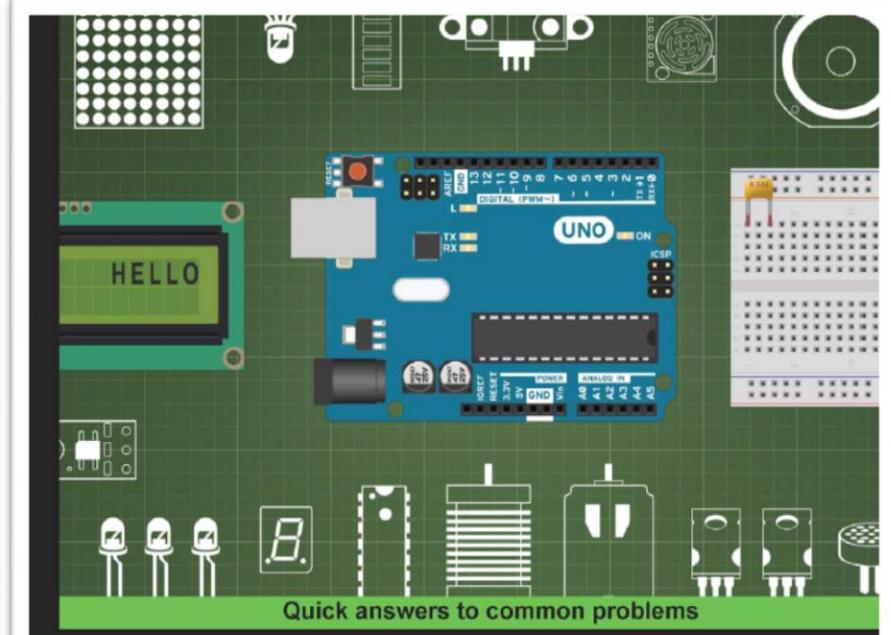


Arduino Essentials

Enter the world of Arduino and its peripherals and start creating interesting projects

Francis Perea

[PACKT]
PUBLISHING



Arduino Development Cookbook

Over 50 hands-on recipes to quickly build and understand Arduino projects, from the simplest to the most extraordinary

Cornel Amariei

[PACKT] open source★
PUBLISHING