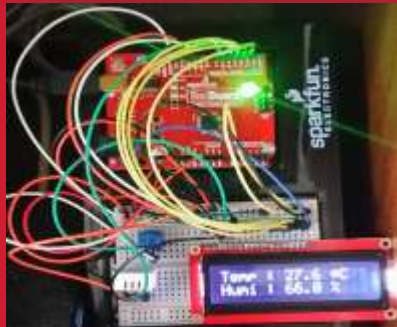




# Arduino-basic

## [wk04]

# LED-I.



Learn how to code Arduino from scratch

Comsi, INJE University

2<sup>nd</sup> semester, 2019

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



# My ID (ARnn)

<b>AR01</b>	염현제
<b>AR02</b>	강민수
<b>AR03</b>	구병준
<b>AR04</b>	김종민
<b>AR05</b>	박성철
<b>AR06</b>	이승현
<b>AR07</b>	이창호
<b>AR08</b>	변성현
<b>AR09</b>	손성빈
<b>AR10</b>	안예찬
<b>AR11</b>	유종인
<b>AR12</b>	이석민
<b>AR13</b>	이주원
<b>AR14</b>	정재영
<b>AR15</b>	차요신

<b>AR16</b>	하태성
<b>AR17</b>	강현이
<b>AR18</b>	신종원
<b>AR19</b>	최진솔
<b>AR20</b>	김경미
<b>AR21</b>	김경영
<b>AR22</b>	김규년
<b>AR23</b>	김민재
<b>AR24</b>	김영록
<b>AR25</b>	송다은
<b>AR26</b>	정지환
<b>AR27</b>	김종건



# [Review]

## ◆ [wk03]

- **Arduino LCD & LED-I.**
- **Complete your project**
- **Submit folder : ARnn\_Rpt02**

# wk03 : Practice-02 : ARnn\_Rpt02

## ◆ [Target of this week]

- Complete your works
- Save your outcomes
- Upload all in github.

제출폴더명 : **ARnn\_Rpt02**

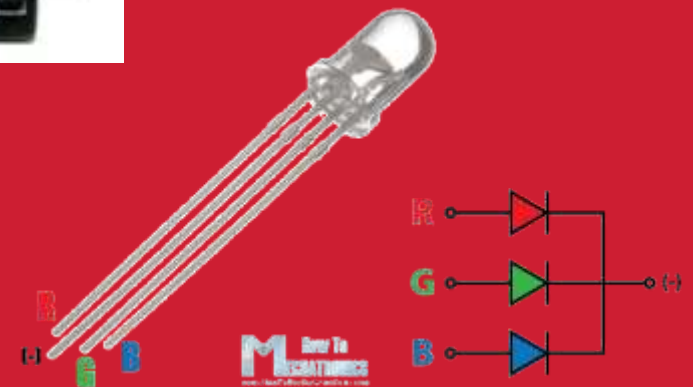
■ 압축할 파일들

- ① **ARnn\_period.ino**
- ② **ARnn\_number.ino**
- ③ **ARnn\_LCD\_hello.png**
- ④ **ARnn\_LCD.ino**



# 4. LED I.

## Light Emitting Diode

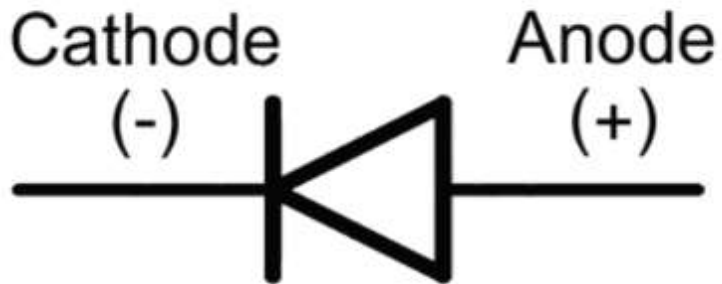


## LED (Light Emitting Diode)

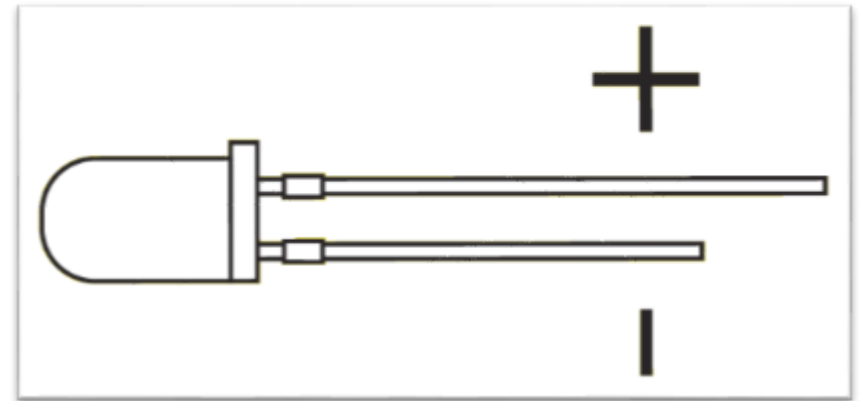
- ✓ 전기 신호를 빛으로 출력하는 반도체 소자
- ✓ 고효율, 반영구적 수명
- ✓ 가정용 실내등, 산업용 특수등, 자동차용 전조등 및 실내등에 사용



## Polarity of Diode and LED



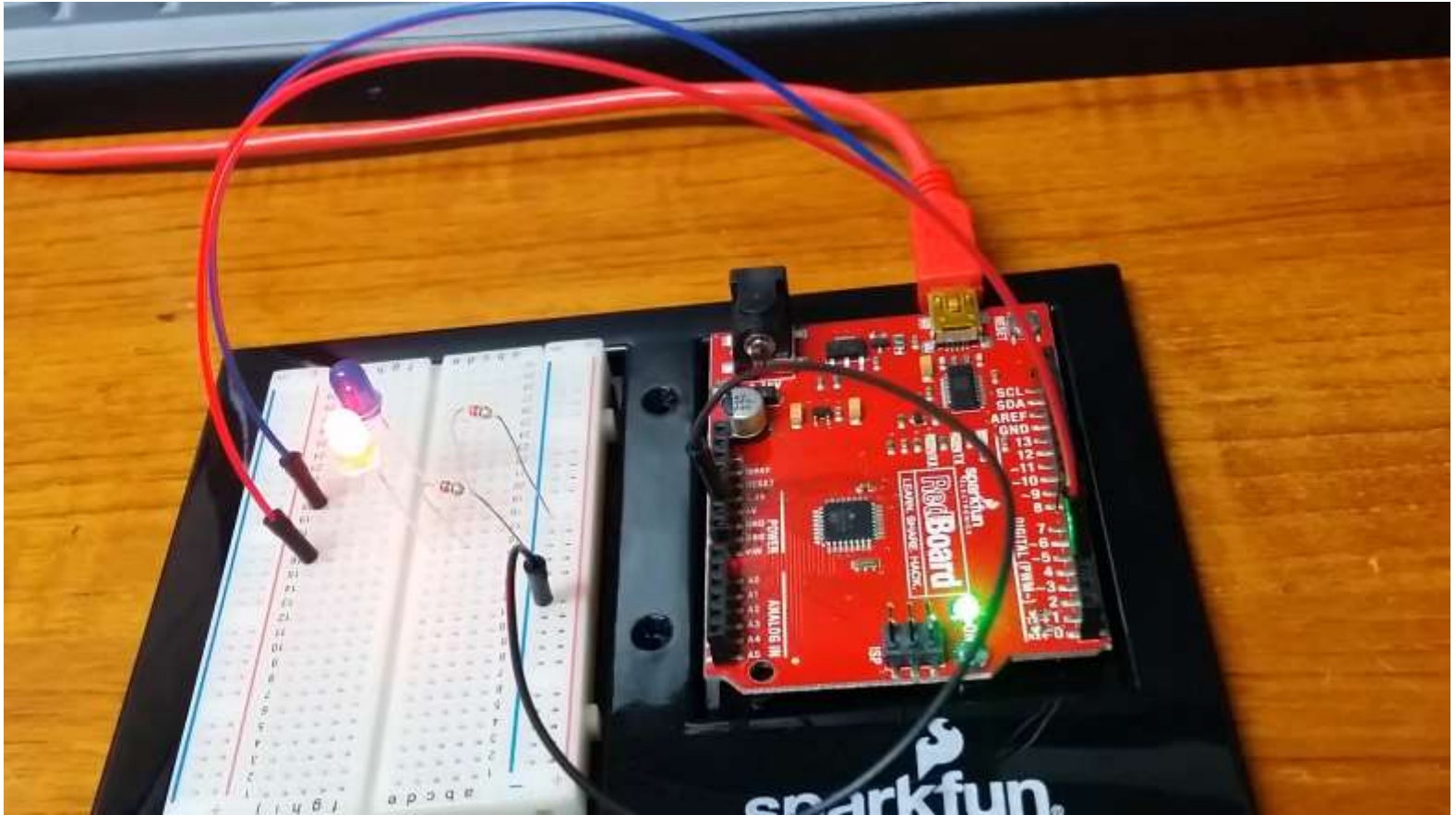
*The diode circuit symbol, with the anode and cathode marked.*



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

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

## 4.1 LED 교차 점멸



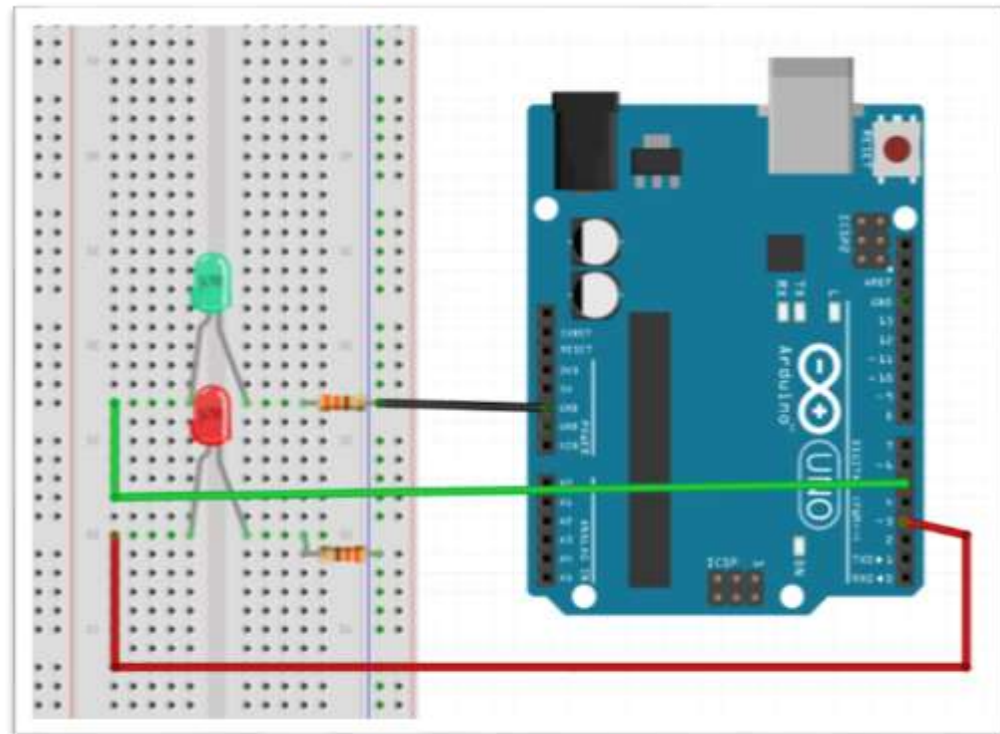
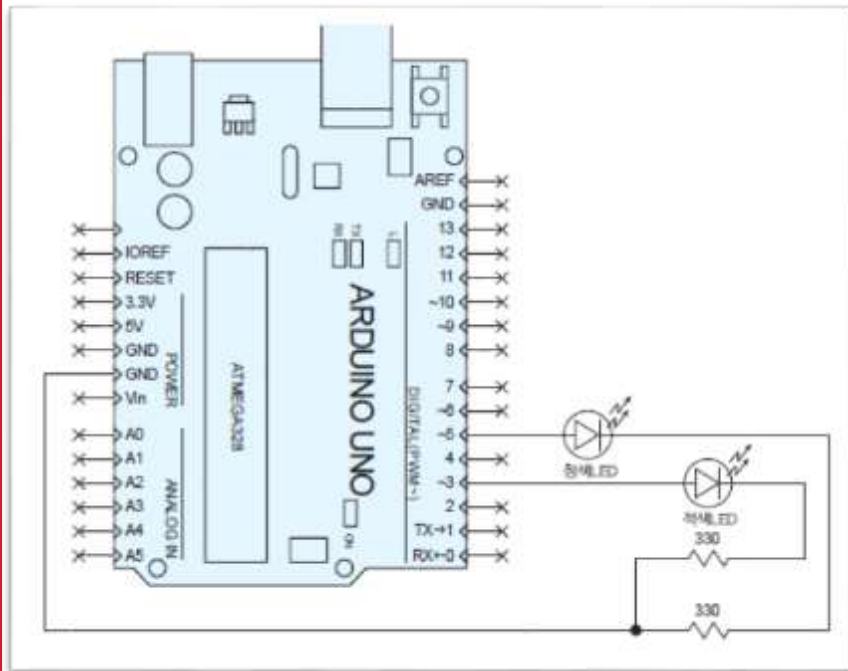


# 4.1.1 LED control - 교차 점멸

## EX 4.1 LED 교차 점멸 (1/3)

**실습목표** 두 개의 LED를 0.1초 간격으로 교차하여 점멸시키자.

**Hardware**



**Save ARnn\_2LED.fzz**

## 4.1.2 LED control - 교차 점멸

### EX 4.1

### 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 켜다.

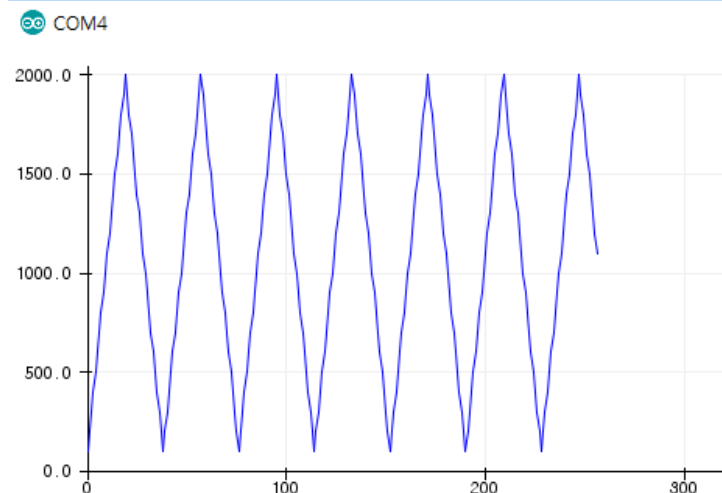
# 4.1.3 LED control - 교차 점멸

## EX 4.1 LED 교차 점멸 (3/3)

**실습 결과** LED A와 B가 0.1초 단위로 교차하며 점멸한다.

**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.)

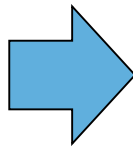
delay = 1600 msec	delay = 500 msec
delay = 1700 msec	delay = 400 msec
delay = 1800 msec	delay = 300 msec
delay = 1900 msec	delay = 200 msec
delay = 2000 msec	delay = 100 msec
delay = 1900 msec	delay = 200 msec
delay = 1800 msec	delay = 300 msec
delay = 1700 msec	delay = 400 msec
delay = 1600 msec	delay = 500 msec



## 4.1.4 LED control - 교차 점멸 (code)

ex\_4\_1

```
1 /*
2  예제 4.1
3  LED 점멸
4  */
5
6 const int ledA = 3;
7 const int ledB = 5;
8
9 void setup()
10 {
11   pinMode(ledA, OUTPUT);
12   pinMode(ledB, OUTPUT);
13 }
14
15 void loop()
16 {
17   digitalWrite(ledA, HIGH);
18   digitalWrite(ledB, LOW);
19   delay(100);
20   digitalWrite(ledA, LOW);
21   digitalWrite(ledB, HIGH);
22   delay(100);
23 }
```



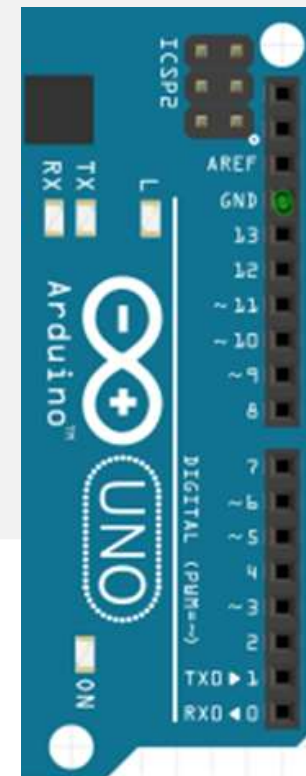
```
6 const int ledA = 3;
7 const int ledB = 5;
8
9 int number = 1;
10 boolean flag = true;
11
12 void setup()
13 {
14   Serial.begin(9600);
15   pinMode(ledA, OUTPUT);
16   pinMode(ledB, OUTPUT);
17 }
```

완성된 스케치 code를  
**ARnn\_2led.ino**  
로 저장해서 제출.

```
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     Fill in your code!
39   }
40   else if (number == 1) {
41     flag = true;
42   }
43 }
```

## 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.



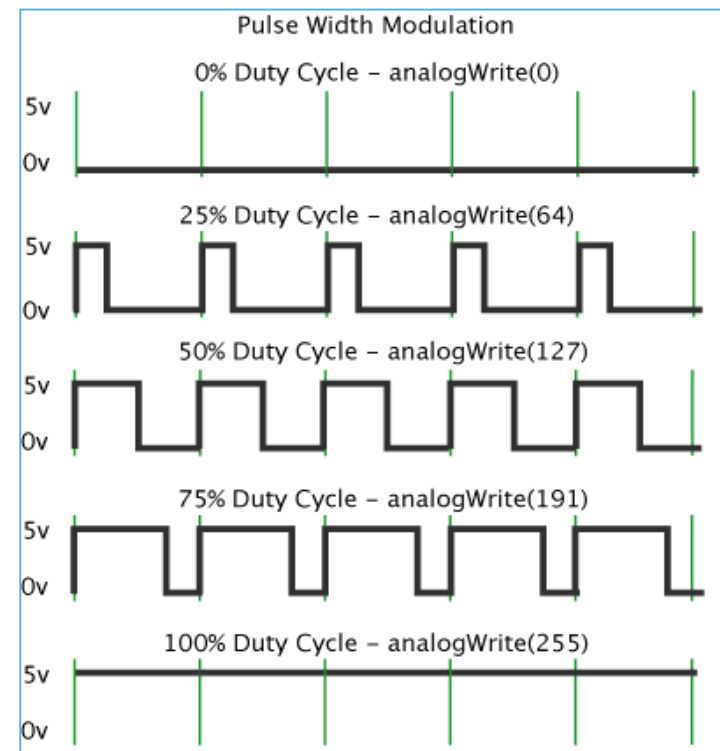
## 4.2.1 LED control - 밝기 조절

### 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



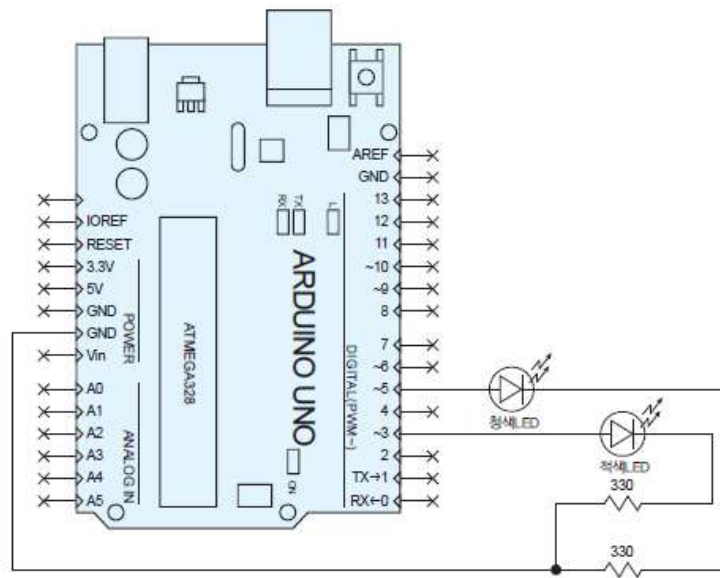
<https://www.arduino.cc/en/Tutorial/PWM>

## EX 4.2

## LED 밝기 조절 (1/2)

- 실습목표**
1. 두 개의 LED의 밝기를 조절하자.
  2. 각각의 LED가 교차하며 밝아졌다 어두워 졌다를 반복하도록 하자.

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



### EX 4.2

### 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.

**실행 결과**    LED A와 B가 밝기가 변화하며 점멸한다.

**응용 문제**

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



## 4.2.4 LED control - 밝기 조절 (code)

ex\_4\_2\_start

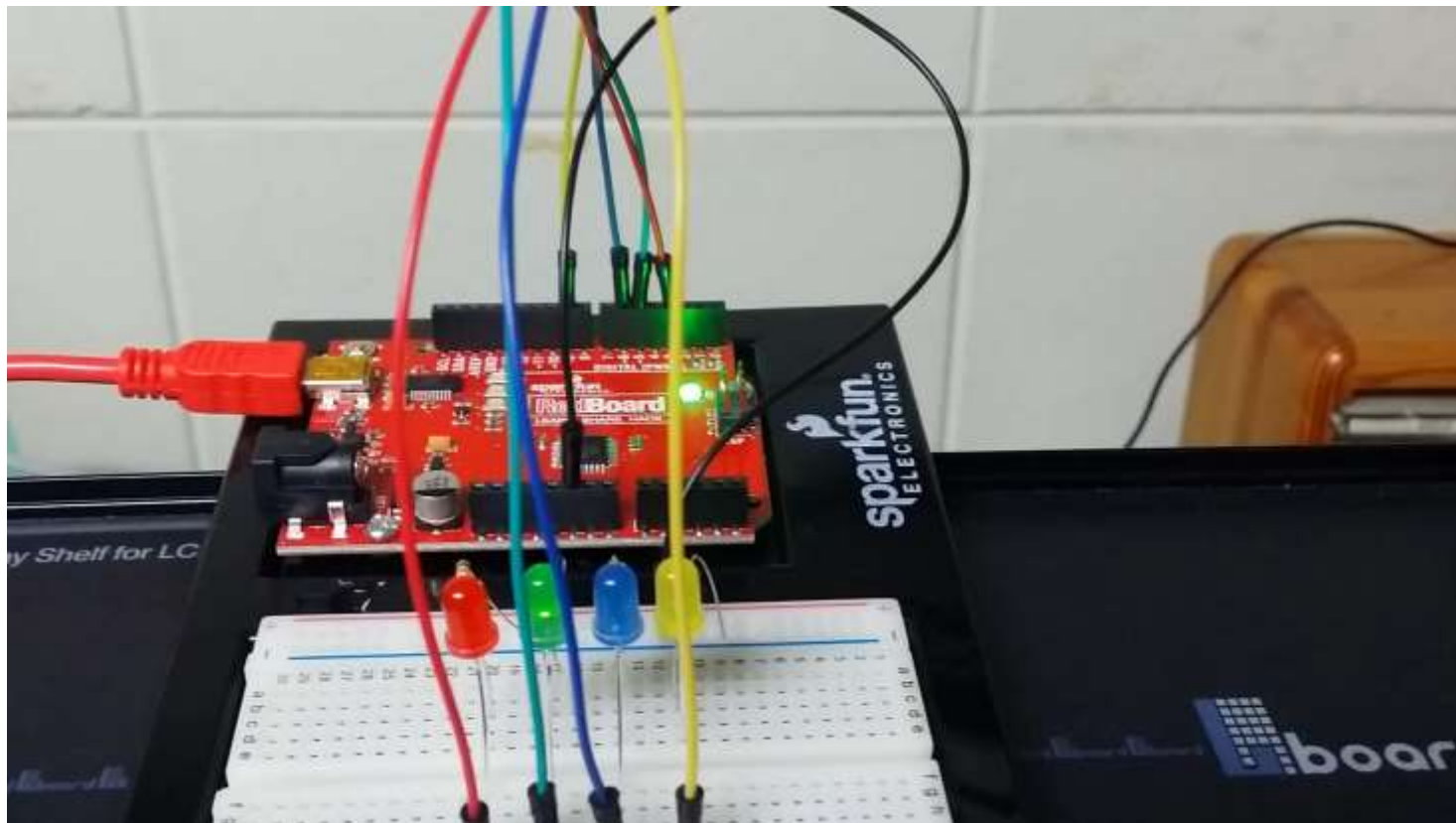
```

1  /*
2  예제 4.2
3  LED 밝기 조절
4  */
5
6  const int ledA   = 3;    //LED A를 3번핀에 연결
7  const int ledB   = 5;    //LED B를 5번핀에 연결
8  int brightness = 0;      //밝기를 조절하기 위한 변수
9  int increment = 1;       //밝기 변수 증감을 위한 변수
10
11 void setup()
12 {
13  // analogWrite 핀에는 별도의 설정이 불필요하다.
14 }
15
16 void loop()
17 {
18  analogWrite(ledA,brightness); // LED A 밝기 조절
19  analogWrite(ledB,255-brightness); // LED B 밝기 조절
20
21  brightness = brightness + increment; // 밝기 조절
22  if((brightness >= 255)|| (brightness <= 0)) increment = -increment; // 밝기 변수 증감 방향 변경
23  delay(10);    // 0.01 초간 지연
24 }

```

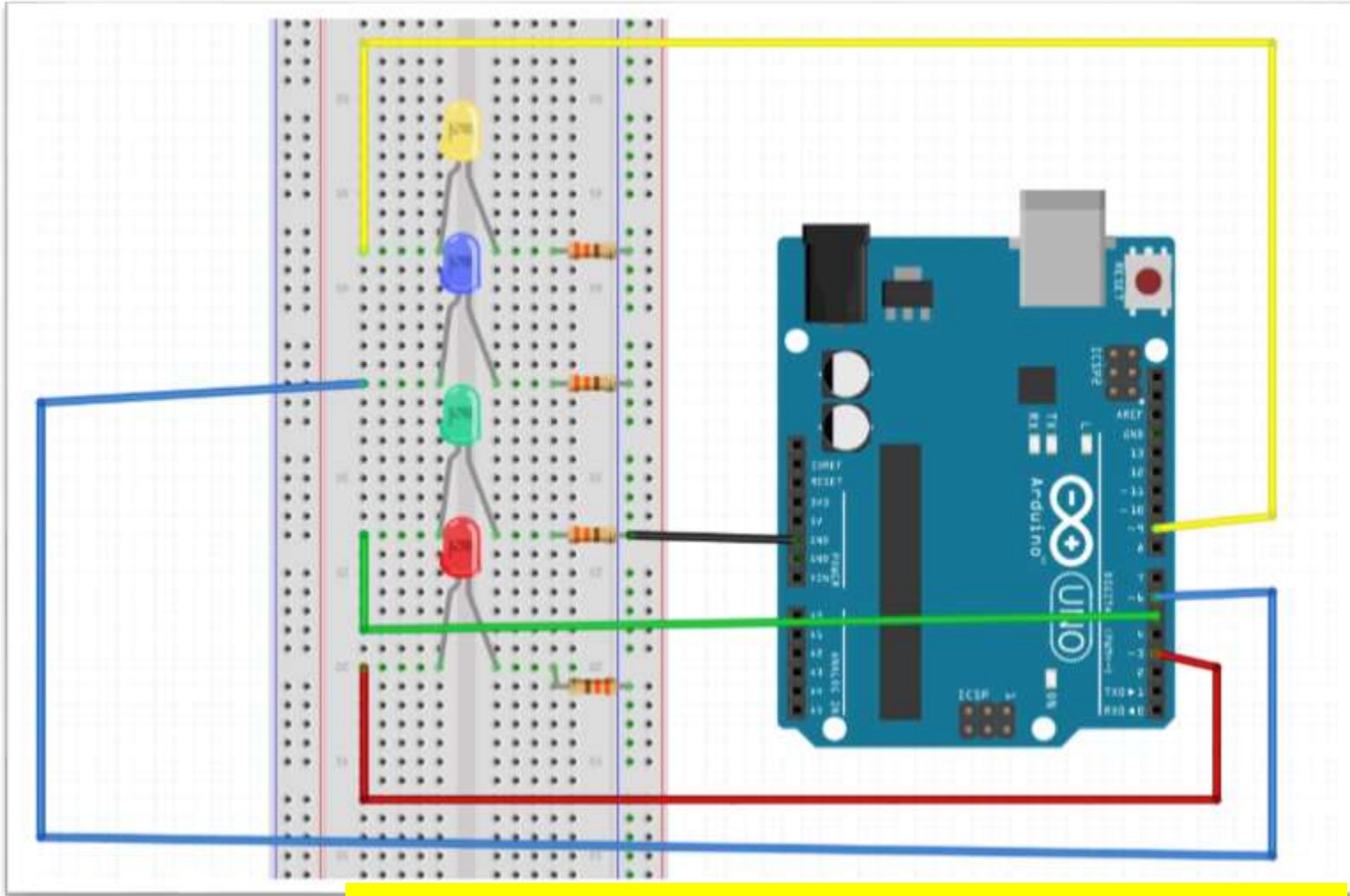
## 4.2.5 LED control - DIY

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



## 4.2.5 LED control - DIY

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



완성된 회로를 [ARnn\\_4led.fzz](#)  
로 저장해서 제출.

## 4.2.5 LED control - DIY: code-1

### ARnn\_4\_led\_start.ino

```

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 }

```

## 4.2.5 LED control - DIY: 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

**ARnn\_4led.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) 반복 사용**

# 4.3 RGB LED control - 색상 조절

## RGB LED

- ✓ 빛의 삼원색인 빨강(Red), 초록(Green), 파랑(Blue)빛을 조절하여 다양한 색을 표현하는 LED.
- ✓ 각각의 색이 0~255단계로 조절됨.
- ✓ 간판, 조명기구 등에 사용
- ✓ 모든 색이 출력될 때 백색 빛을 출력

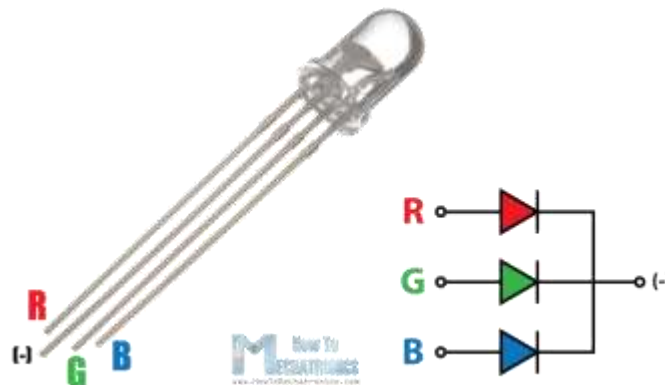
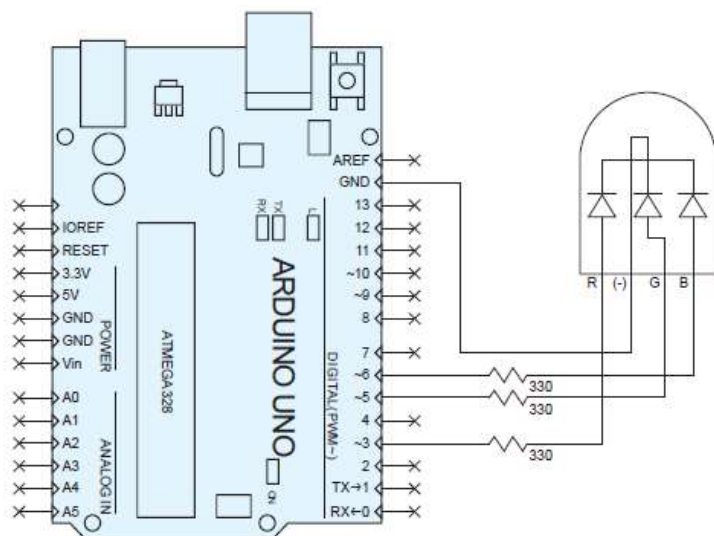


# 4.3.1 RGB LED control - 색상 조절

## EX 4.3 RGB LED로 색상 표현하기 (1/2)

**실습목표** RGB LED를 이용하여 다양한 색을 표현해 보자.

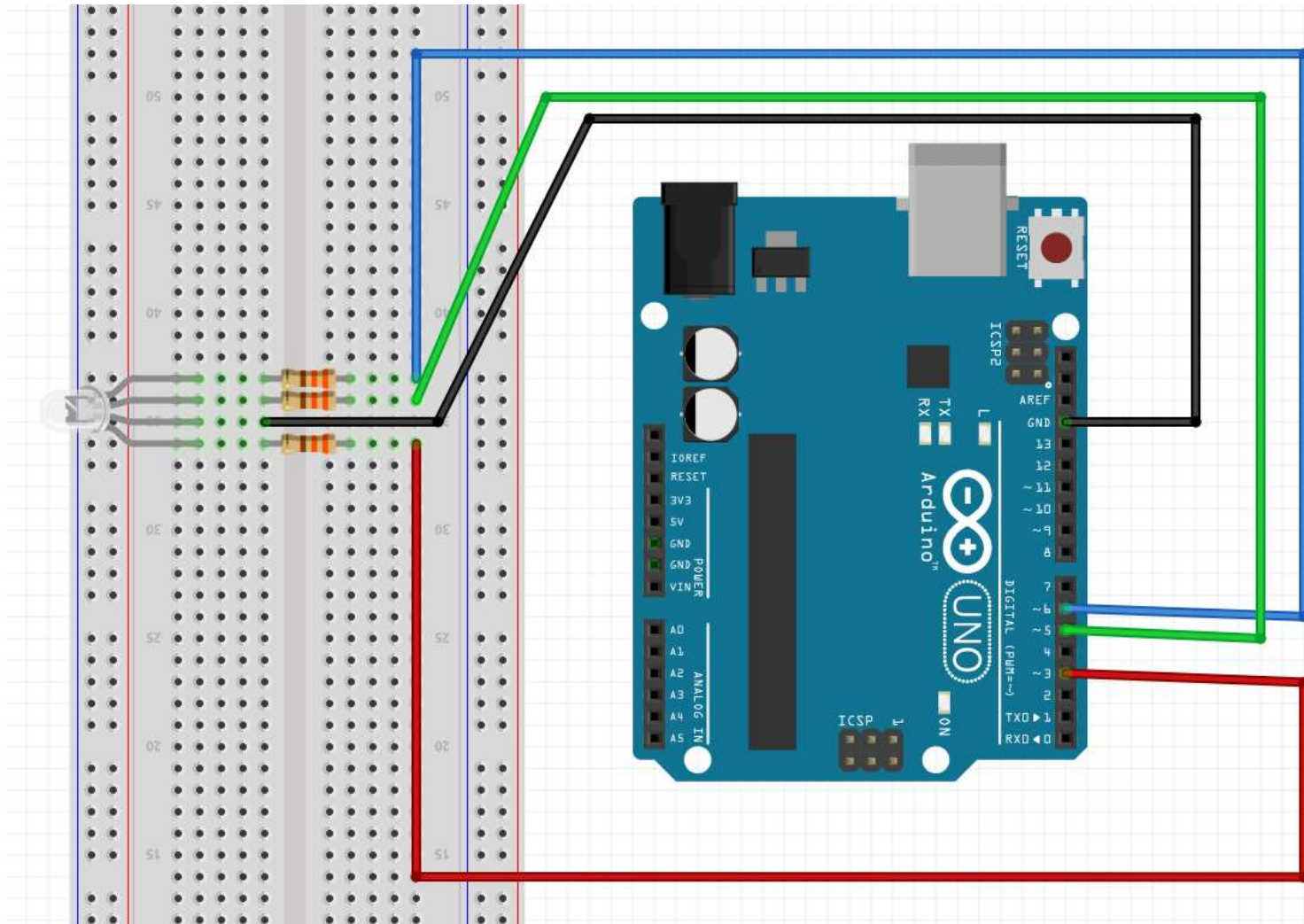
- Hardware**
1. RGB LED는 Red, Green, Blue의 세 개의 Anode 핀과 공통으로 연결된 캐소드핀으로 구성되어 있다.
  2. RGB LED 단독으로 연결하려면 **각 Anode 핀에 330Ω의 저항을 연결**해야 한다.
  3. **저항이 내장된 RGB LED 모듈을 사용한다면 별도의 저항이 필요 없다.**
  4. Red, Green, Blue의 세 개의 Anode 핀을 Arduino의 3, 5, 6 번핀에 연결한다.



<http://howtomechatronics.com/tutorials/arduino/how-to-use-a-rgb-led-with-arduino/>



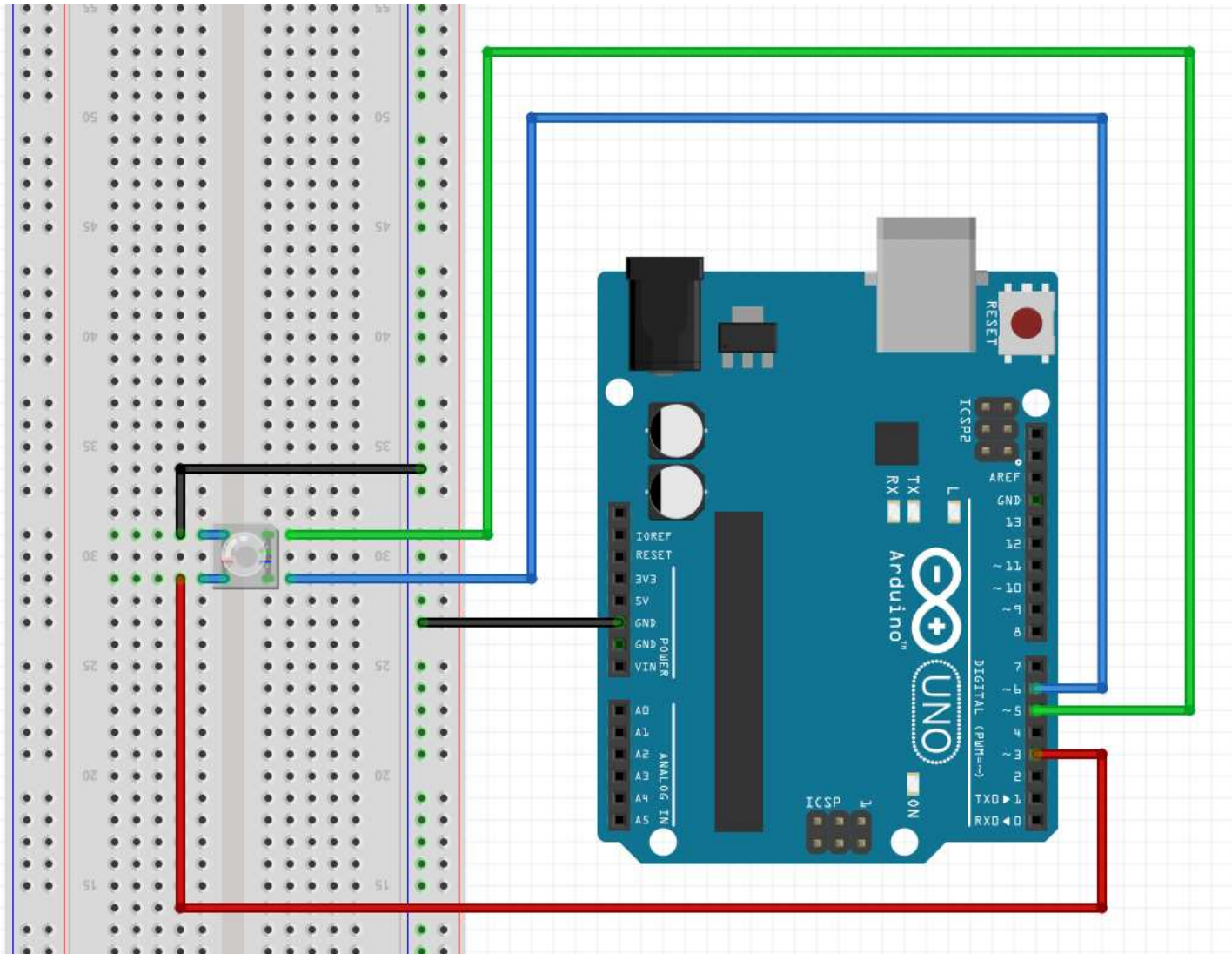
## 4.3.2 RGB LED control - 색상 조절



완성된 회로를 [ARnn\\_RGB.fzz](#)  
로 저장해서 제출.



## 4.3.2 RGB module control - 색상 조절



# 4.3.3 RGB LED control - 색상 조절

## EX 4.3 RGB LED로 색상 표현하기 (2/2)

**Commands**

- analogWrite(핀번호, 값)  
정해진 핀에 아날로그 출력을 한다. '값' 에는 0~255의 값을 넣는다.
- delay(지연시간)  
지연시간에는 잠시 동작을 지연시키기 위한 값을 넣는다. 1/1000초 단위로 넣는다.  
즉 1초를 지연시키기 위해선 1000의 값을 입력시킨다.
- for(변수=시작 값 ; 조건 ; 변수의 증분){}  
변수의 시작 값부터 조건이 만족하는 경우 '{ }' 내의 명령을 수행한다. '변수의 증분'에서는 1회 명령이 수행될 때마다 변수를 증가 혹은 감소시킨다.

**Sketch 구성**

1. LED의 핀번호를 설정한다.
2. setup()에서는 LED 출력으로 사용할 핀을 출력 핀으로 설정한다.
3. `ledOutput(int Red, int Green, int Blue)`라는 함수를 만든다. 적색, 녹색, 청색 LED의 빛의 세기를 조합하여 원하는 색을 출력하는 함수이다.
4. 적색, 녹색, 청색 LED의 세기를 조절하면서 LED에 빛을 출력시킨다.

**실습 결과**  
**DIY**

LED의 색상이 변화를 조사한다.

[http://www.rapidtables.com/web/color/RGB\\_Color.htm](http://www.rapidtables.com/web/color/RGB_Color.htm)



## 4.3.4 RGB LED control - code

```

ex_4_3_start
1 /*
2  예제 4.3
3  RGB-LED 밝기 조절
4 */
5
6 const int RedLed    = 3:    //를 3번핀에 연결
7 const int GreenLed  = 5:    //LED B를 5번핀에 연결
8 const int BlueLed   = 6:    //LED B를 5번핀에 연결
9
10 void setup()
11 {
12   ledOutput(255, 0, 0);
13   delay(1000);
14   ledOutput(0, 255, 0);
15   delay(1000);
16   ledOutput(0, 0, 255);
17   delay(1000);
18 }

```

완성된 스케치 code를  
**ARnn\_RGB.ino**  
로 저장해서 제출.

```

48 void ledOutput(int Red, int Green, int Blue){
49   analogWrite(RedLed, Red);
50   analogWrite(GreenLed, Green);
51   analogWrite(BlueLed, Blue);
52 }

```

```

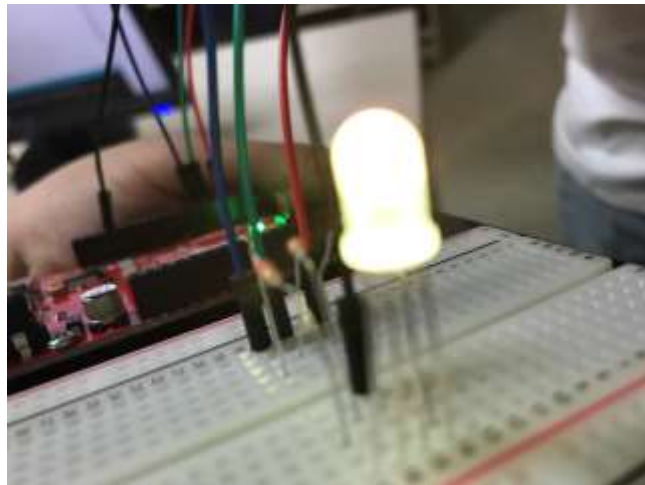
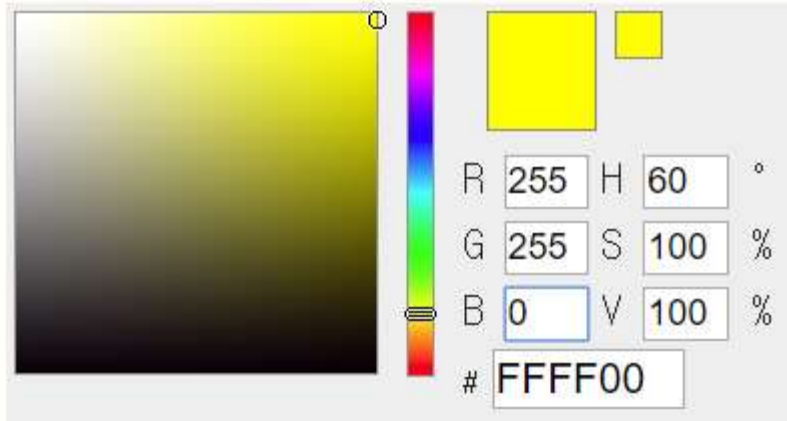
20 void loop()
21 {
22   for(int i=0; i<=255; ++i){
23     ledOutput(255, i, 0);
24     delay(10);
25   }
26   for(int i=0; i<=255; ++i){
27     ledOutput(0, 255, i);
28     delay(10);
29   }
30   for(int i=0; i<=255; ++i){
31     ledOutput(i, 0, 255);
32     delay(10);
33   }
34   for(int i=0; i<=255; ++i){
35     ledOutput(i, 255, 255);
36     delay(10);
37   }
38   for(int i=0; i<=255; ++i){
39     ledOutput(255, i, 255);
40     delay(10);
41   }
42   for(int i=0; i<=255; ++i){
43     ledOutput(255, 255, i);
44     delay(10);
45   }
46 }

```

# 4.4 RGB LED control - 색상 조절 [DIY]

DIY. RGB LED의 색이 노란색일 때 사진을 촬영하시오.

RGB color picker



ARnn\_RGB\_Y.png 로 저장



# 4. LED II.

## FND

4 1EA



7세그먼트 1채널

---

공통 음극 7세그먼트  
시계나 점수 등의 숫자를  
표현 할 때 많이 사용됩니다.

5 1EA



74HC595N

---

기본 메인보드입니다.  
74HC595N LED,  
도트매트릭스, NFD 제어 IC 입니다.

3 1EA



7세그먼트 4채널

---

7세그먼트가 4개 연결된 형태의  
부품입니다.  
총 12개의 핀을 사용합니다.

23 1EA



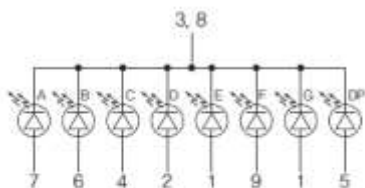
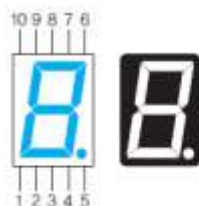
8x8 도트매트릭스 모듈

---

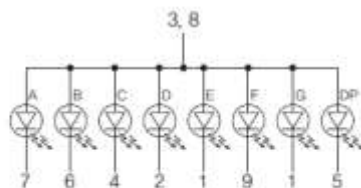
LED로 다양한 연출을  
할 수 있습니다.

## FND (Flexible Numeric Display)

- ✓ LED의 조합으로 숫자를 표시하는 장치
- ✓ 7개의 LED를 사용하기 때문에 7-segment 라고도 함.
- ✓ 숫자뿐만 아니라 간단한 기호나 16진수 까지 표현 가능



(a)



(b)

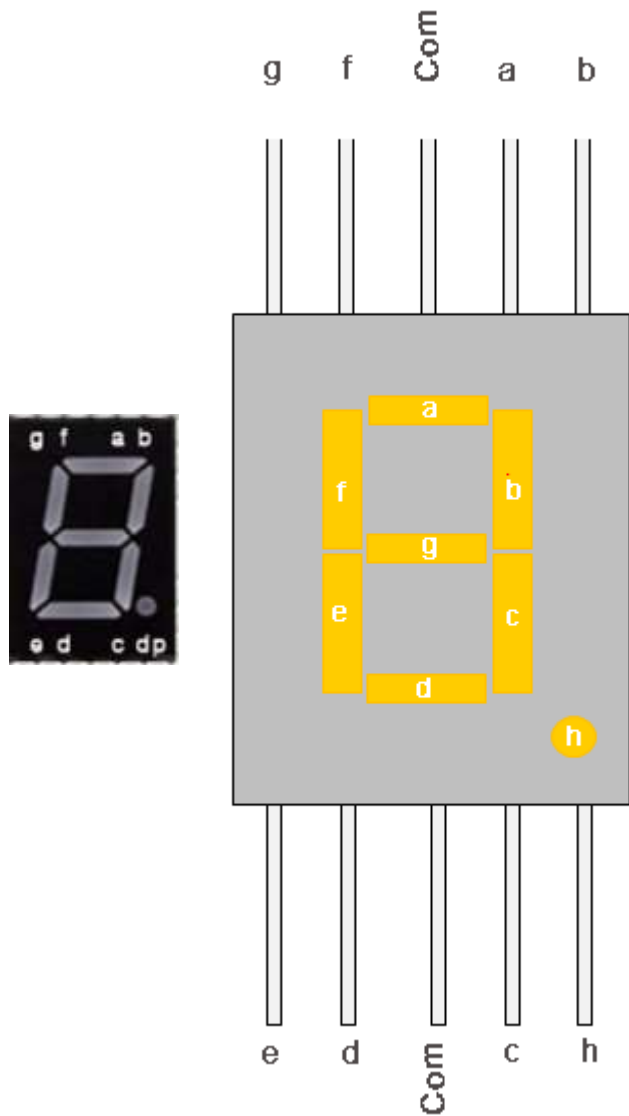
그림 4.4 Common Cathode 형(a)와 Common Anode 형(b)

표 4.1 Common Cathode FND 표시

캐소드 공통 7-세그먼트 한 자리 제어 방법										7-Seg. 출력 내용
I/O 포트 출력 내용										16진수
Q0	DP	G	F	E	D	C	B	A	16진수	7-Seg. 출력 내용
1	X	X	X	X	X	X	X	X	X	8 (소등)
0	0	0	1	1	1	1	1	1	0x3f	8 (0)
0	0	0	0	0	0	1	1	0	0x06	8 (1)
0	0	1	0	1	1	0	1	1	0x5b	8 (2)
0	0	1	0	0	1	1	1	1	0x4f	8 (3)
0	0	1	1	0	0	1	1	0	0x66	8 (4)
0	0	1	1	0	1	1	0	1	0x6d	8 (5)
0	0	1	1	1	1	1	0	1	0x7d	8 (6)
0	0	0	0	0	0	1	1	1	0x27	8 (7)
0	0	1	1	1	1	1	1	1	0x7f	8 (8)
0	0	1	1	0	1	1	1	1	0x6f	8 (9)
0	0	1	1	1	0	1	1	1	0x77	8 (A)
0	0	1	1	1	1	1	0	0	0x7c	8 (b)
0	0	0	1	1	1	0	0	1	0x39	8 (C)
0	0	1	0	1	1	1	1	0	0x5e	8 (d)
0	0	1	1	1	1	0	0	1	0x79	8 (E)
0	0	1	1	1	0	0	0	1	0x71	8 (F)
0	1	0	0	0	0	0	0	0	0x80	8 (.)



## 4.5 FND 제어



	h	g	f	e	d	c	b	a	hex value
0	0	0	1	1	1	1	1	1	3F
1	0	0	0	0	0	1	1	0	06
2	0	1	0	1	1	0	1	1	5B
3	0	1	0	0	1	1	1	1	4F
4	0	1	1	0	0	1	1	0	66
5	0	1	1	0	1	1	0	1	6D
6	0	1	1	1	1	1	0	1	7D
7	0	0	0	0	0	1	1	1	07
8	0	1	1	1	1	1	1	1	7F
9	0	1	1	0	1	1	1	1	6F
A	0	1	1	1	0	1	1	1	77
b	0	1	1	1	1	1	0	0	7C
C	0	0	1	1	1	0	0	1	39
d	0	1	0	1	1	1	1	0	5E
E	0	1	1	1	1	0	0	1	79
F	0	1	1	1	0	0	0	1	71



# [Practice]

## ◆ [wk04]

- **Arduino LED-I.**
- **Complete your project**
- **Submit folder : ARnn\_Rpt03**



# wk04 : Practice-03 : ARnn\_Rpt03

## ◆ [Target of this week]

- Complete your works
- Save your outcomes
- Upload all in github.

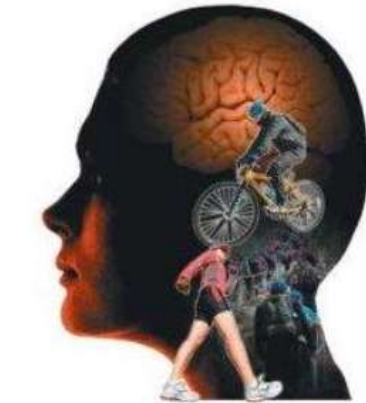
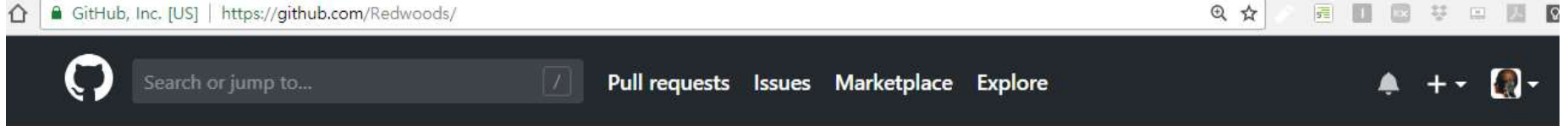
제출폴더명 : **ARnn\_Rpt03**

### 제출할 파일들

- ① **ARnn\_2led.ino**
- ② **ARnn\_4led.fzz**
- ③ **ARnn\_4led.ino**
- ④ **ARnn\_RGB.fzz**
- ⑤ **ARnn\_RGB.ino**
- ⑥ **ARnn\_RGB\_Y.png**

## ● References & good sites

- ✓ <http://www.arduino.cc> Arduino Homepage
- ✓ <http://www.github.com> GitHub
- ✓ <http://www.google.com> Googling
- ✓ <https://www.youtube.com> Youtube



**Redwoods Yi**

Redwoods

Add a bio

GimHae, Republic of Korea

chaos21c@gmail.com

Overview

Repositories 7

Stars 2

Followers 1

Following 0

## Pinned repositories

Customize your pinned repositories

Py

Lectures on coding python from scratch to the advanced level.

Jupyter Notebook

Arduino

Lectures on learning Arduino from scratch to the advanced level in iot environment.

Lec

All lectures by Redwoods in Inje University

Jupyter Notebook

hw-coding

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


Arduino

171 contributions in the last year




Contribution settings

Redwoods/Arduino: Lect

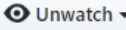
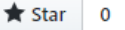

GitHub, Inc. [US] | https://github.com/Redwoods/Arduino

 Search or jump to...


[Pull requests](#) [Issues](#) [Marketplace](#) [Explore](#)


  


Redwoods / Arduino


 1  0  0


<> Code


 Issues 0

 Pull requests 0

 Projects 0

 Wiki


 Insights


 Settings


Lectures on learning Arduino from scratch to the advanced level in iot environment.


Edit

Add topics

 2 commits

 1 branch

 0 releases

 1 contributor

Branch: master


New pull request

Create new file


Upload files

Find file

Clone or download


 Redwoods 2018 start

Latest commit 38ca9e0 28 minutes ago

 ar-basic


2018 start

28 minutes ago

 ar-iot


2018 start

28 minutes ago

 README.md

Initial commit

43 minutes ago

 README.md

## Arduino

---

Lectures on learning Arduino from scratch to the advanced level in iot environment.





[http://arduinostory.com/goods/goods\\_view.php?goodsNo=1000000306](http://arduinostory.com/goods/goods_view.php?goodsNo=1000000306)



## 상급키트 구성품

<b>1</b> 1EA  <b>아두이노 우노 R3 DIP</b> 아두이노 우노 R3 (DIP) 호환보드 기본 메인보드입니다.	<b>2</b> 1EA  <b>9V 배터리 홀더</b> 9V 배터리를 연결하여 아두이노에 외부전원을 공급할 수 있습니다.	<b>3</b> 1EA  <b>7세그먼트 4채널</b> 7세그먼트가 4개 연결된 형태의 부품입니다. 총 12개의 핀을 사용합니다.	<b>4</b> 1EA  <b>7세그먼트 1채널</b> 공통 음극 7세그먼트 시계나 점수 등의 숫자를 표현 할 때 많이 사용됩니다.
<b>5</b> 1EA  <b>74HC595N</b> 기본 메인보드입니다. 74HC595N LED, 드레드텍스, NFD 제어 IC 입니다.	<b>6</b> 1EA  <b>65핀 점퍼 와이어</b> 브레드보드에 연결할 때 사용하는 65핀 점퍼와이어 입니다.	<b>7</b> 1EA  <b>무지개 점퍼선 F-M 20cm</b> M타입과 F타입이 양쪽으로 달린 무지개 점퍼선입니다.	<b>8</b> 1EA  <b>투명 부품 케이스 대,소</b> 키트 구성품을 담을 수 있는 투명 부품 케이스입니다.
<b>9</b> 1EA  <b>가변저항10K</b> 물리변 저항값이 바뀝니다. (0~10KΩ)	<b>10</b> 1EA  <b>1602 I2C LCD</b> 아두이노 16x2 I2C LCD 모듈입니다. LCD입니다.	<b>11</b> 1EA  <b>저항</b> 100, 220, 330, 1K, 2K, 4.7K, 10K, 47K, 100K	<b>12</b> 1EA  <b>브레드 보드 830홀</b> 브레드 보드 830홀(봉무형) 센서 테스트나, 회로 프로토타입을 작성할 때 사용됩니다.

<b>13</b> 1EA  <b>수동부저</b> 아두이노의 tone함수를 통해 소리를 내는 부저입니다.	<b>14</b> 6EA  <b>택트스위치 (12x12x7)</b> 스위치를 누르고 있을 경우만 ON됩니다.	<b>15</b> 1EA  <b>택트스위치 컵 (피랑,노랑,초록,빨강,하양)</b> 택트스위치를 사용할 때 스위치간의 구분을 할 수 있습니다.	<b>16</b> 3EA  <b>조도센서</b> 빛을 감지하거나 빛의 밝기를 아날로그로 출력해주는 CDS 센서입니다.
<b>17</b> 1EA  <b>LED 5mm (빨강,노랑,초록,하양,파랑)</b> 기본으로 사용되는 LED입니다. 동작전압 : 2.2~2.4V 사용전류 : 20mA 미만	<b>18</b> 1EA  <b>헤더핀 1x40/2.54mm</b> 핀 간격은 2.54mm이며 헤더핀의 길이는 약 1.15cm입니다.	<b>19</b> 1EA  <b>USB케이블 50cm</b> PC와 아두이노 우노 보드를 연결하여 프로그램을 다운로드 할 때 사용합니다.	<b>20</b> 1EA  <b>저항값 카드</b> 저항값을 쉽게 확인 할 수 있는 카드입니다. 사이즈 : 60mm x 50mm
<b>21</b> 1EA  <b>능동부저</b> Signal 단자가 HIGH 일 때 약 2.5kHz의 음이 발생됩니다.	<b>22</b> 1EA  <b>5V 1채널 릴레이 모듈</b> 아두이노의 디지털 핀과 모듈 하단의 IN 핀들을 연결해 릴레이를 제어할 수 있는 모듈입니다.	<b>23</b> 1EA  <b>8x8 도트 매트릭스 모듈</b> LED로 다양한 연출을 할 수 있습니다.	<b>24</b> 1EA  <b>4x4 16 키패드 모듈</b> 16개의 버튼을 사용할 수 있습니다.

# 아두이노 키트(Kit) : Part-2

<div>25</div> <div>1EA</div> <div></div> <div>무선 리모콘 키트</div> <div>핵파선을 사용해서 리모콘 기능을 구현할 수 있습니다.</div>	<div>26</div> <div>2EA</div> <div></div> <div>가열기 센서 스위치</div> <div>센서의 가열기에 따라 스위치 역할을 합니다.</div>	<div>27</div> <div>1EA</div> <div></div> <div>or</div> <div>사운드 센서 모듈</div> <div>아두이노와 호환되는 사운드센서 모듈입니다.</div>	<div>28</div> <div>1EA</div> <div></div> <div>불꽃 센서</div> <div>근거리 화재, 불꽃을 감지하는 센서입니다.</div>	<div>37</div> <div>1EA</div> <div></div> <div>DC 5V 스텝 모터</div> <div>28BYJ 28BYJ48 스텝 모터 중 저렴한 편에 속하는 모델입니다. 5개의 핀을 사용합니다.</div>	<div>38</div> <div>1EA</div> <div></div> <div>DS1302 RTC 모듈</div> <div>아두이노 등 마이크로컨트롤러에서 사용이 가능합니다.</div>	<div>39</div> <div>1EA</div> <div></div> <div>아두이노 우노 프로토 쉼드</div> <div>UNO 보드에서 회로를 간단히 짜기 위해 보드 위에 얹어 사용하는 쉼드입니다.</div>	<div>40</div> <div>1EA</div> <div></div> <div>3축 가속도 센서 모듈</div> <div>가속도를 측정할 수 있는 센서입니다.</div>
<div>29</div> <div>1EA</div> <div></div> <div>모터 드라이버 모듈</div> <div>ULN2003 스텝 모터 드라이버 모듈 5V ~ 12V를 사용할 수 있습니다.</div>	<div>30</div> <div>1EA</div> <div></div> <div>LM35 온도 센서</div> <div>온도를 아날로그 값으로 출력합니다.</div>	<div>31</div> <div>1EA</div> <div></div> <div>수위 센서 모듈</div> <div>센서가 액체에 잠긴 정도를 아날로그 값으로 출력합니다.</div>	<div>32</div> <div>1EA</div> <div></div> <div>SG90 서보모터</div> <div>Vcc, GND, 신호선, 총 3개의 핀이 있습니다. 로봇팔이나 자동차, 비행기 조종에 사용됩니다.</div>	<div>41</div> <div>1EA</div> <div></div> <div>5V DC모터</div> <div>5V DC모터</div>	<div>42</div> <div>1EA</div> <div></div> <div>인체 감지 센서 모듈</div> <div>핵파선을 이용해 움직임 감지하는 센서입니다. 오선이 감지되면 HIGH 신호를 출력합니다.</div>	<div>43</div> <div>5EA</div> <div></div> <div>다이오드 1N4001</div> <div>다이오드 1N4001</div>	<div>44</div> <div>5EA</div> <div></div> <div>세라믹 캐패시터 (22pF)</div> <div>세라믹 캐패시터 (22pF)</div>
<div>33</div> <div>1EA</div> <div></div> <div>초음파 거리 센서 모듈</div> <div>5V를 사용하여 인식 거리는 2cm에서 500cm입니다.</div>	<div>34</div> <div>1EA</div> <div></div> <div>조이스틱 모듈</div> <div>기본적으로 조이스틱 모듈은 두개의 가변저항이 서로 수직으로 회전하는 형태로 되어 있습니다.</div>	<div>35</div> <div>1EA</div> <div></div> <div>온습도 센서 모듈</div> <div>아두이노 온습도 센서중 가장 대중적으로 사용되는 DHT11 디지털 센서입니다.</div>	<div>36</div> <div>1EA</div> <div></div> <div>RGB LED 모듈</div> <div>RGB LED 모듈로 RGB LED 세개를 하나로 묶은 상품입니다.</div>	<div>45</div> <div>5EA</div> <div></div> <div>세라믹 캐패시터 (1uF)</div> <div>세라믹 캐패시터 (1uF)</div>	<div>46</div> <div>5EA</div> <div></div> <div>트랜지스터 2N2222</div> <div>트랜지스터 2N2222</div>	<div>47</div> <div>5EA</div> <div></div> <div>트랜지스터 BC547</div> <div>트랜지스터 BC547</div>	<div>48</div> <div>5EA</div> <div></div> <div>트랜지스터 BC557</div> <div>트랜지스터 BC557</div>
<div>49</div> <div>2EA</div> <div></div> <div>전해 캐패시터 (50V 10uF)</div> <div>전해 캐패시터 (50V 10uF)</div>	<div>50</div> <div>2EA</div> <div></div> <div>전해 캐패시터 (50V 100uF)</div> <div>전해 캐패시터 (50V 100uF)</div>	<div></div>					