



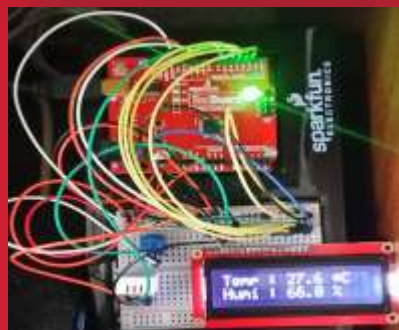
Arduino-basic

[wk04]



LED - 1

R,G,B,Y & RGB



Learn how to code Arduino from scratch

Comsi, INJE University

2nd semester, 2018

Email : chaos21c@gmail.com



My ID (ARnn)

성명	ID
백동진	AR01
김도훈	AR02
김희찬	AR03
류재현	AR04
문민규	AR05
박진석	AR06
이승현	AR07
이승협	AR08
이후정	AR09
최민구	AR10

김다영	AR11
공진영	AR12
김해인	AR13
류성현	AR14
류재환	AR15
박상현	AR16
박해주	AR17
백지혜	AR18
송원식	AR19
신송주	AR20
윤지훈	AR21
정은성	AR22



[Review]

◆ [wk03]

- **Arduino LCD**
- **Complete your project**
- **Submit file : ARnn_Rpt02.zip**

wk03 : Practice-01 : ARnn_Rpt02.zip

◆ [Target of this week]

- Complete your works
- Save your outcomes and compress all.

제출파일명 : **ARnn_Rpt02.zip**

- 압축할 파일들

- ① **ARnn_period.ino**
- ② **ARnn_number.ino**
- ③ **ARnn_LCD.fzz**
- ④ **ARnn_LCD_hello.png**
- ⑤ **Arnn_LCD.ino**

Email : chaos21c@gmail.com

[제목 : id, 이름 (수정)]



2. Serial comm. monitor & plotter

2. Serial comm.

시리얼 통신

- 2.1 Arduino에서 컴퓨터로 데이터 전송하기
- 2.2 변수 유형별로 컴퓨터에 전송하기
- 2.3 Arduino에서 시리얼 통신을 이용하여
 데이터 수신하기



3. LCD

Liquid Crystal Display



핀에 직접 연결
7 핀



I²C 통신
2 핀

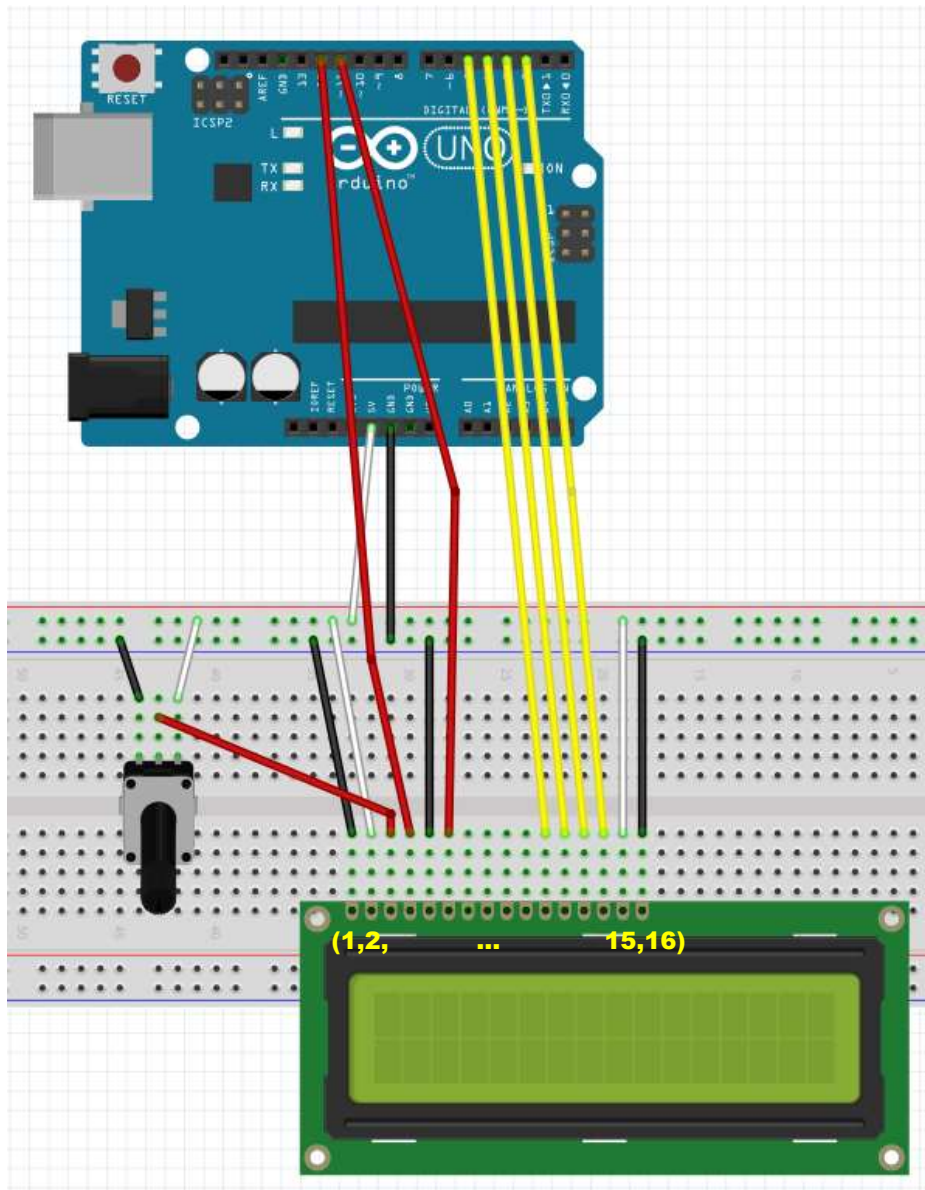


얇은 액정판 아래 조명을 비추는 장치로서 액정판의 전류 흐름을 제어하여 문자나 그림을 표시

Liquid crystal display

- 3.1 입출력 핀을 이용하여 LCD 모듈에 표시하기
- 3.2 I²C를 이용한 LCD 출력

3.1.5 데이터 입력 초기화 (pin-4, 6, 11,12,13,14)



Pin 1 to Arduino GND

Pin 2 to Arduino 5V

Pin 3 to wiper

Pin 4 to Arduino pin D12

Pin 5 to Arduino GND

Pin 6 to Arduino pin D11

Pin 11 to Arduino pin D5

Pin 12 to Arduino pin D4

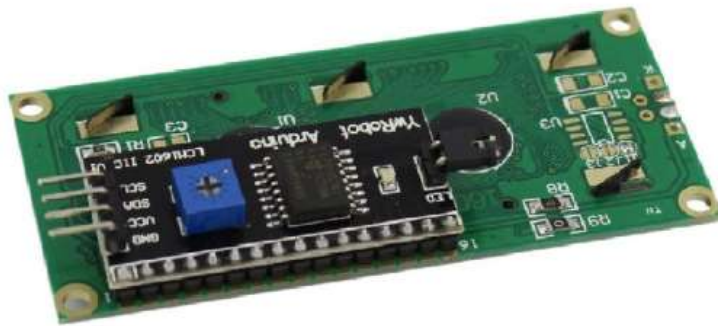
Pin 13 to Arduino pin D3

Pin 14 to Arduino pin D2

Pin 15 to +5V

Pin 16 to GND

3.2 I²C를 이용한 LCD 출력



I²C(아이스퀘어드시, **Inter-Integrated Circuit**)는 필립스에서 개발한 직렬 버스이다. 마더보드, 임베디드 시스템, 휴대 전화 등에 저속의 주변 기기를 연결하기 위해 사용된다.

I²C는 풀업 저항이 연결된 직렬 데이터(**SDA**)와 직렬 클럭(**SCL**)이라는 두 개의 양 방향 오픈 컬렉터 라인을 사용한다. 최대 전압은 **+5 V**이며, 일반적으로 **+3.3 V** 시스템이 사용되지만 다른 전압도 가능하다.

<https://ko.wikipedia.org/wiki/I%C2%B2C>

<http://www.ifuturetech.org/product/16x2-lcd-i2c-lcd/>



EX 3.2 I²C를 이용한 LCD 출력 (3/3)

- 실행 결과
1. Arduino LCD 표시 후 백라이트가 2회 점멸한다.
 2. 시리얼 모니터를 실행 시킨 후 메시지를 입력하여 보자. → “Hello ARnn”
 3. 메시지가 LCD에 출력되는지를 확인해 보자.

Take a photo of LCD screen.

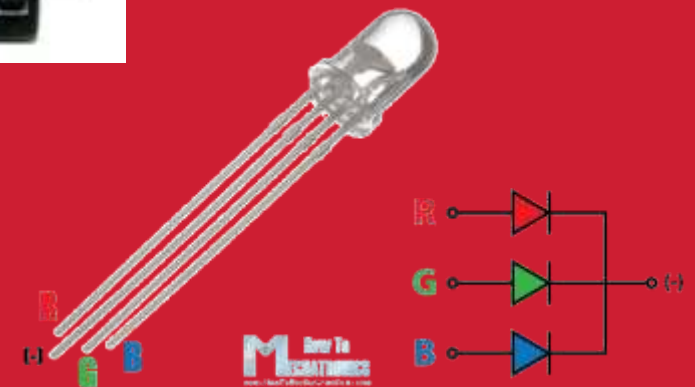
**Save photo as
ARnn_LCD_hello.png**





4. LED

Light Emitting Diode

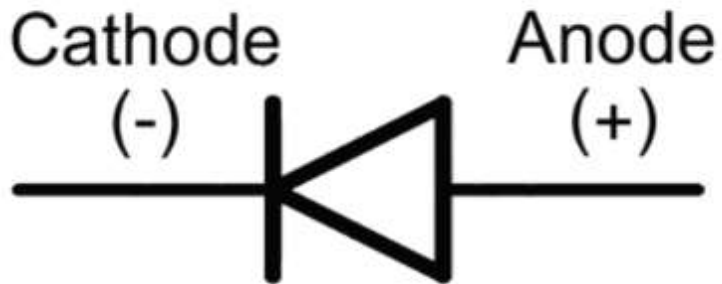


LED (Light Emitting Diode)

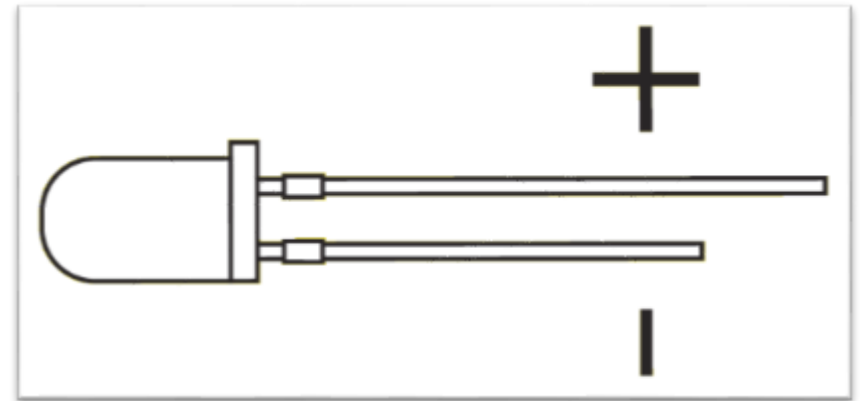
- ✓ 전기 신호를 빛으로 출력하는 반도체 소자
- ✓ 고효율, 반영구적 수명
- ✓ 가정용 실내등, 산업용 특수등, 자동차용 전조등 및 실내등에 사용



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>

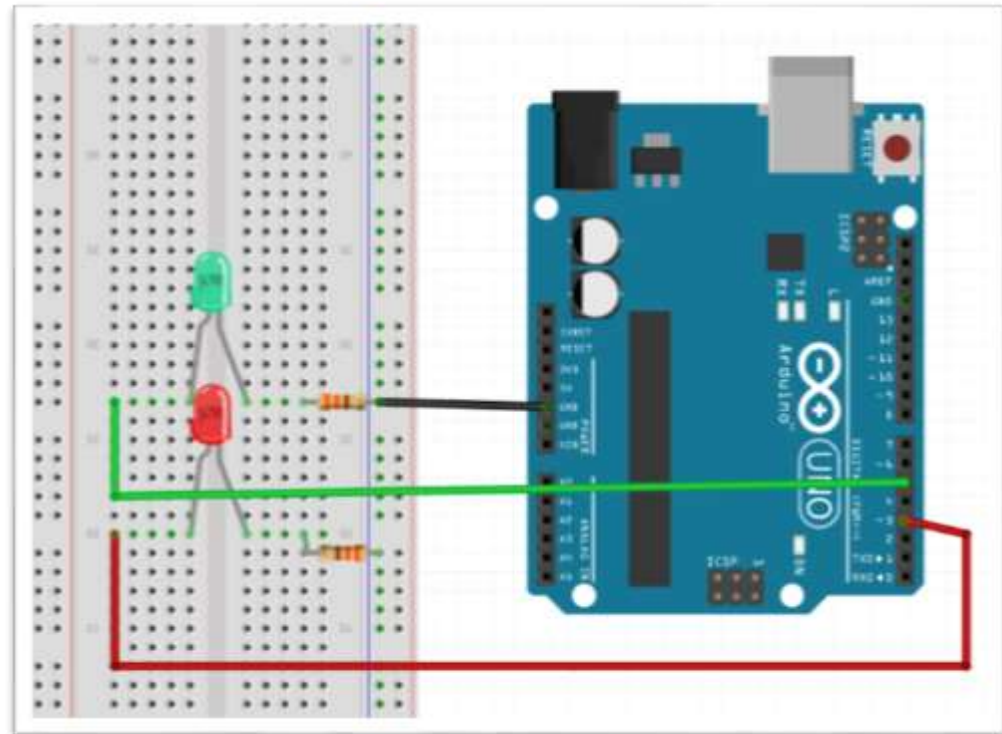
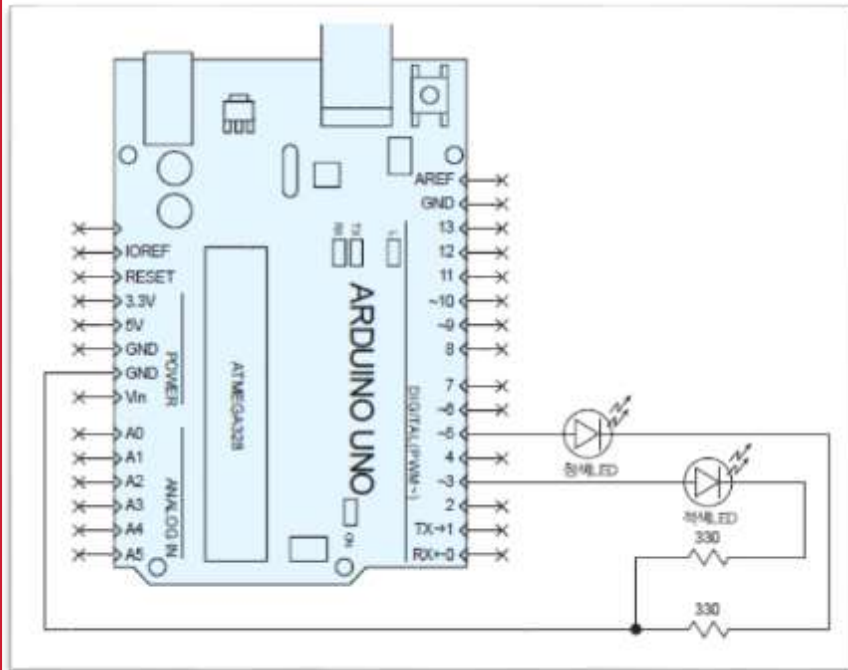
4.1 LED 교차 점멸



EX 4.1 LED 교차 점멸 (1/3)

실습목표 두 개의 LED를 0.1초 간격으로 교차하여 점멸시키자.

Hardware



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.

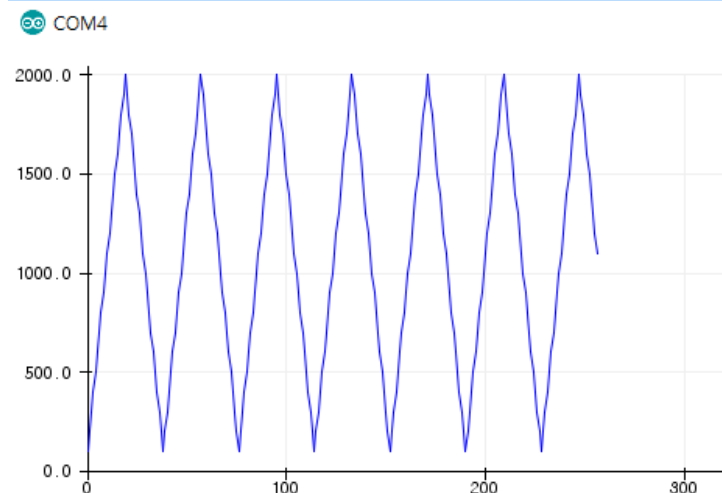
4.1.3 LED control - 교차 점멸

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



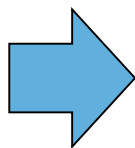
4.1.4 LED control - 교차 점멸 (code)

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.

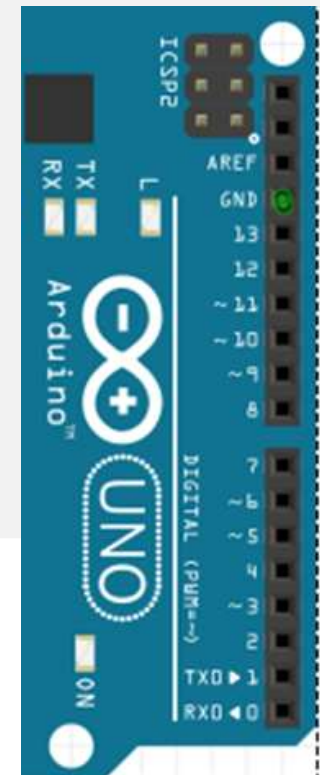
```

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 핀**이 PWM을 지원한다.



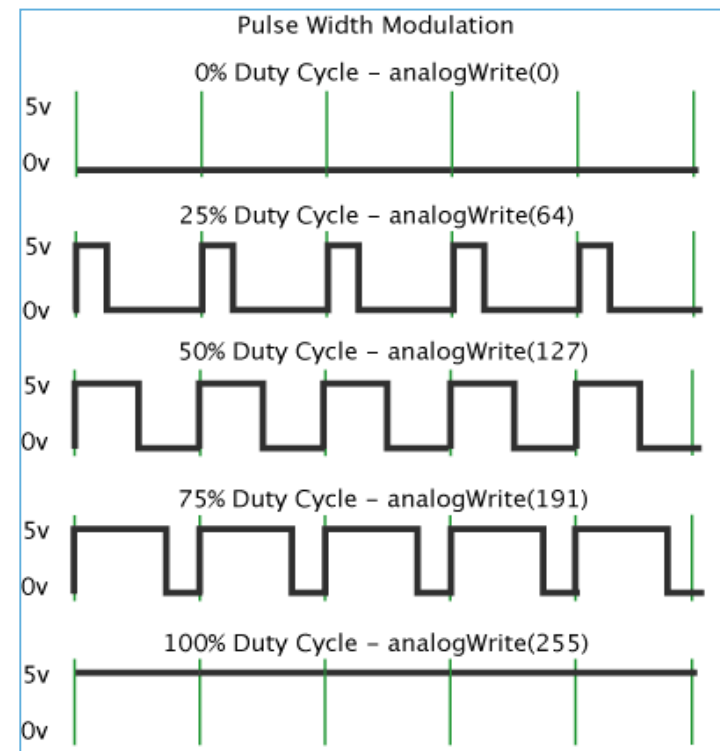
4.2.1 LED control - 밝기 조절

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

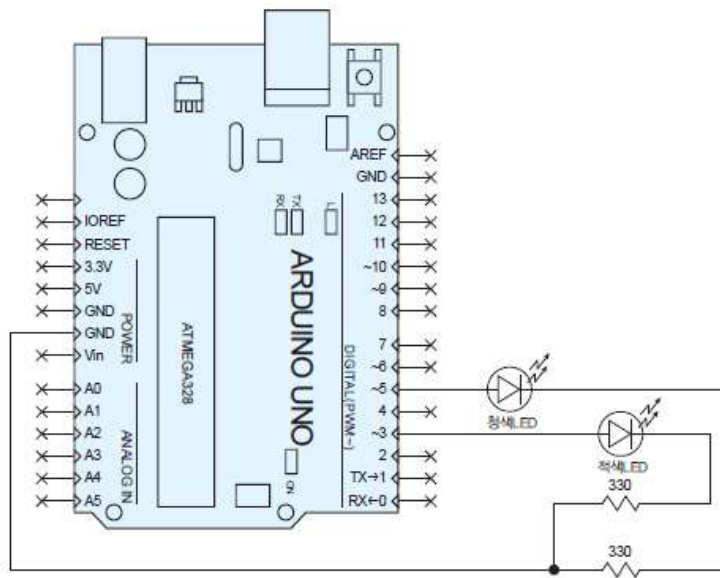


<https://www.arduino.cc/en/Tutorial/PWM>

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 Ω 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.

실험 결과 LED A와 B가 밝기가 변화하며 점멸한다.

응용 문제

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

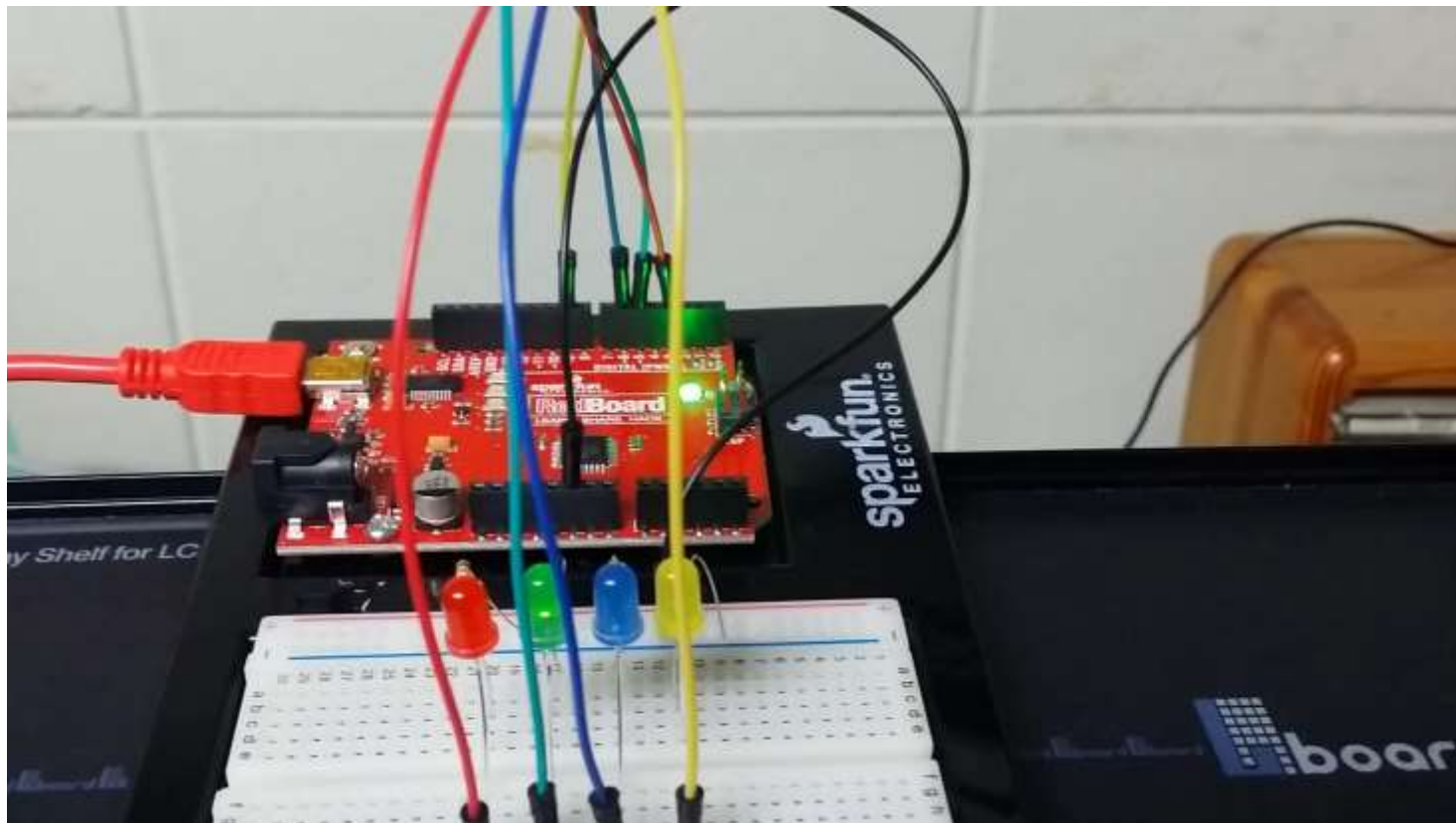
4.2.4 LED control - 밝기 조절 (code)

```

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 }
  
```

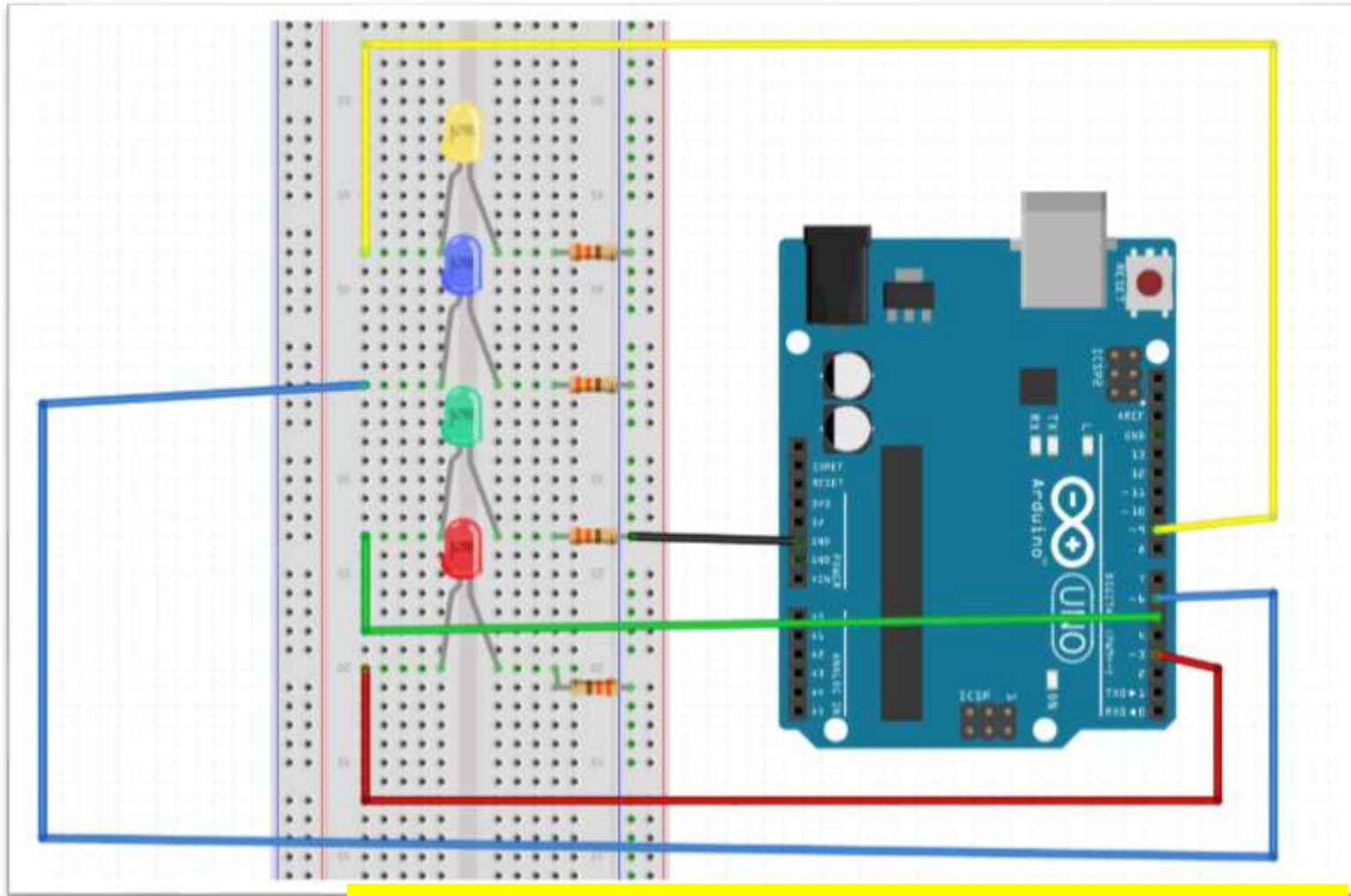

4.2.5 LED control - DIY

- 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   }

```

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

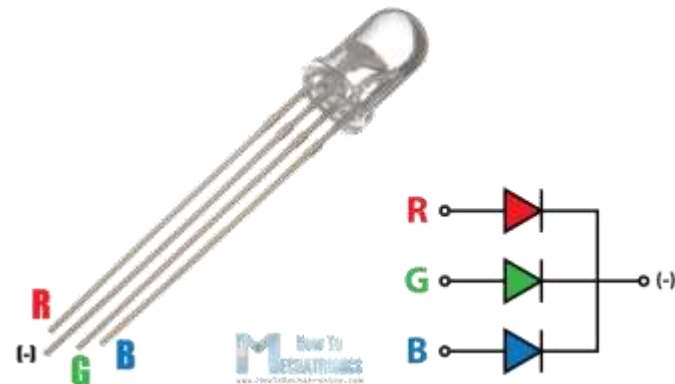
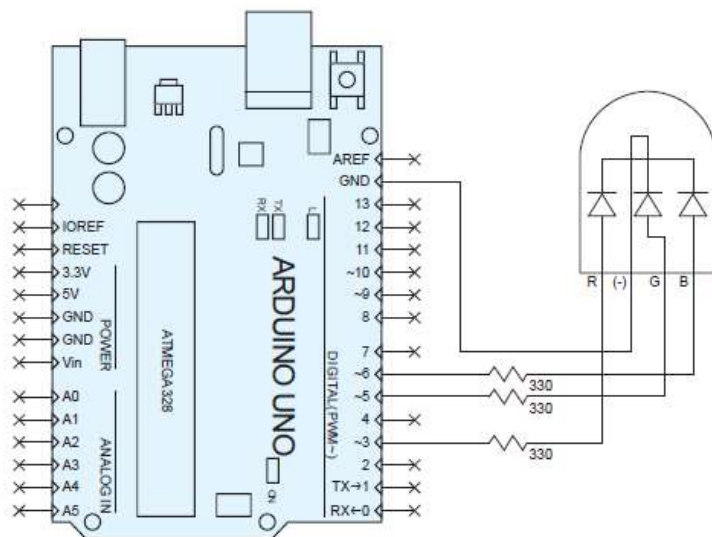


4.3.1 RGB LED control - 색상 조절

EX 4.3 RGB LED로 색상 표현하기 (1/2)

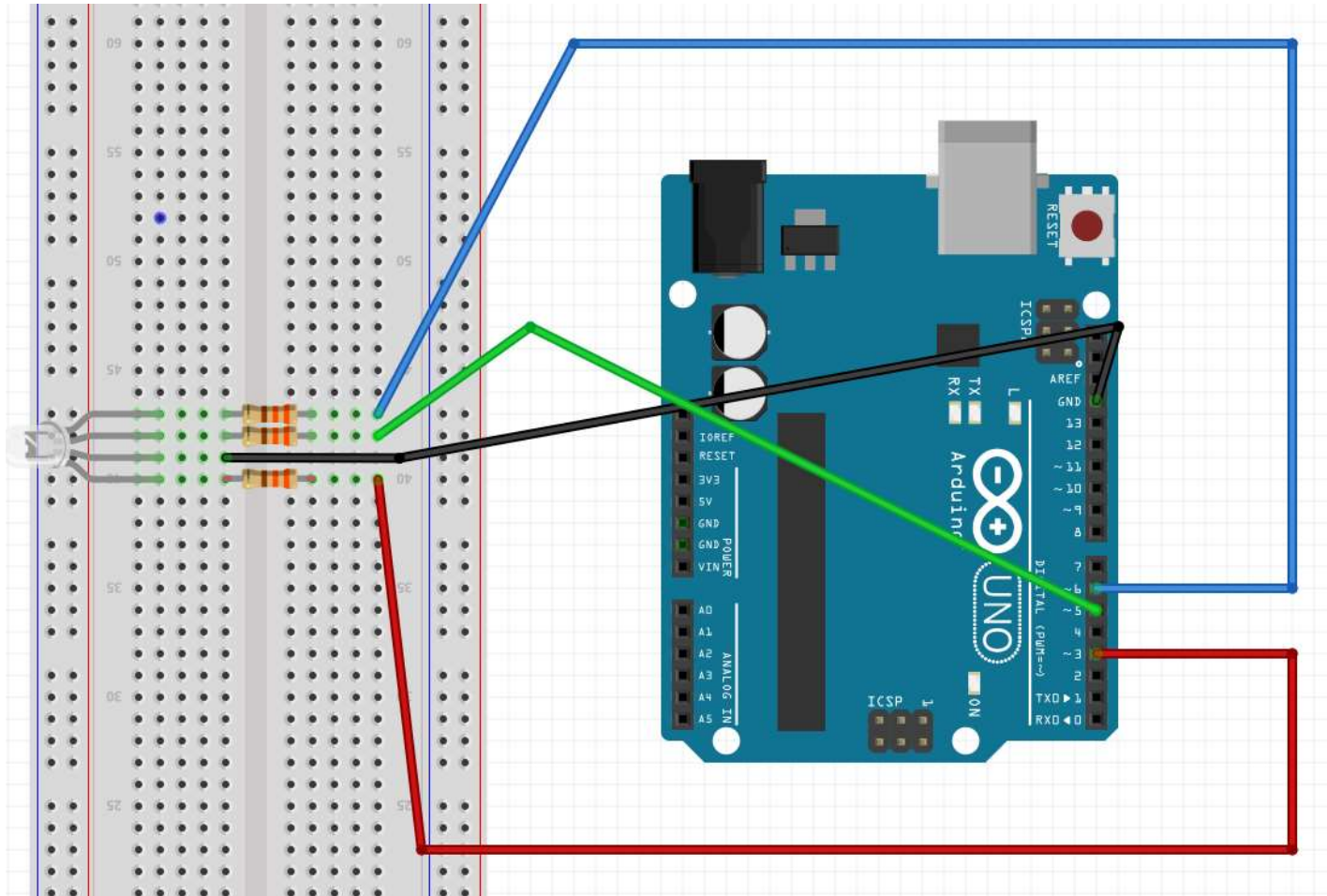
실습목표 RGB LED를 이용하여 다양한 색을 표현해 보자.

- 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.

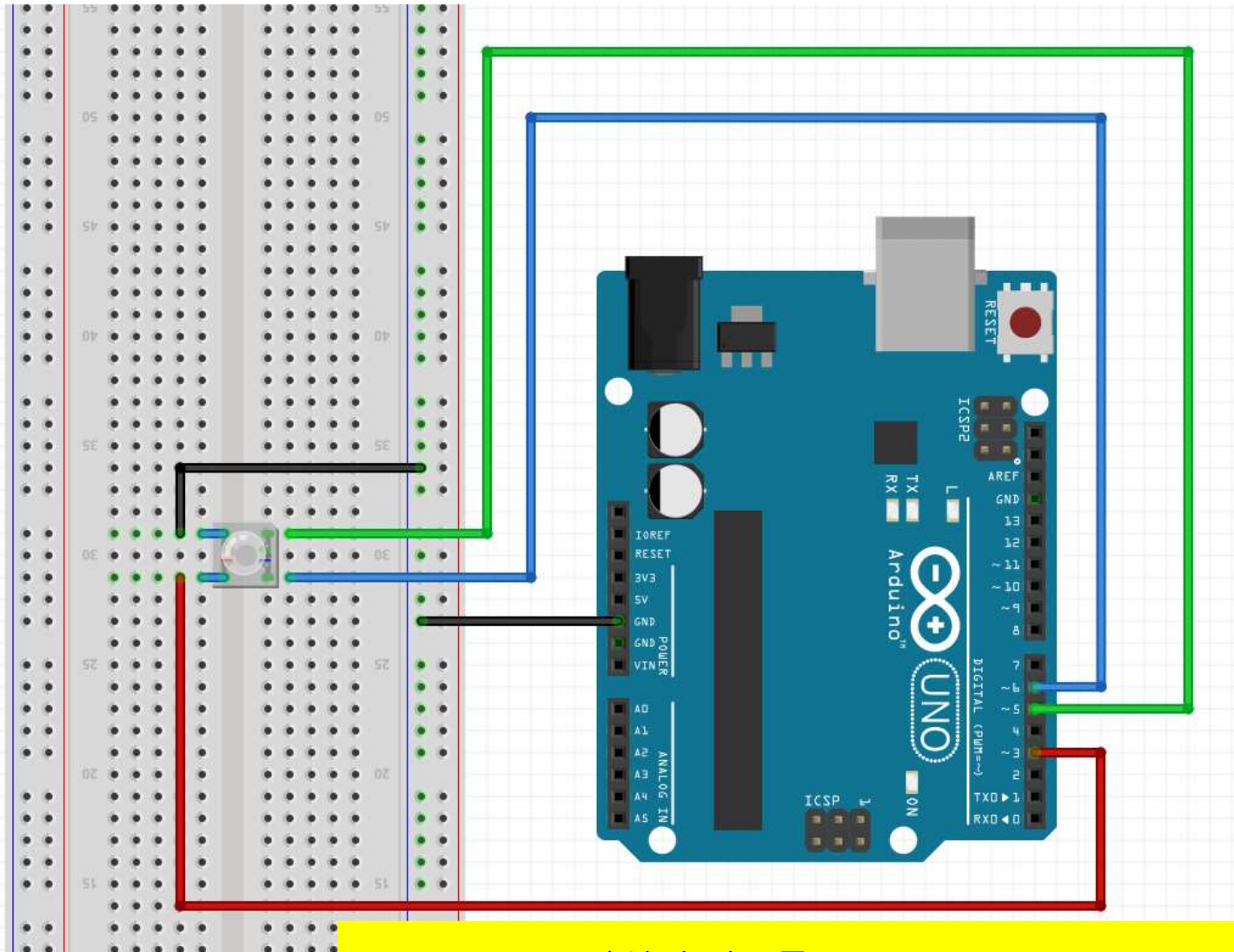


<http://howtomechatronics.com/tutorials/arduino/how-to-use-a-rgb-led-with-arduino/>

4.3.2 RGB LED control - 색상 조절



4.3.2 RGB module control - 색상 조절



완성된 회로를 **ARnn_RGB.fzz**

로 저장해서 제출.

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



4.3.4 RGB LED control - code

```
ex_4_3_start
2  예제 4.2
3  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 }
```

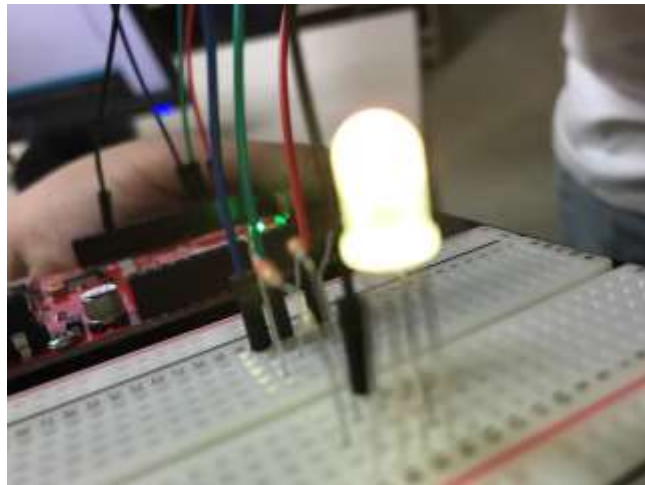
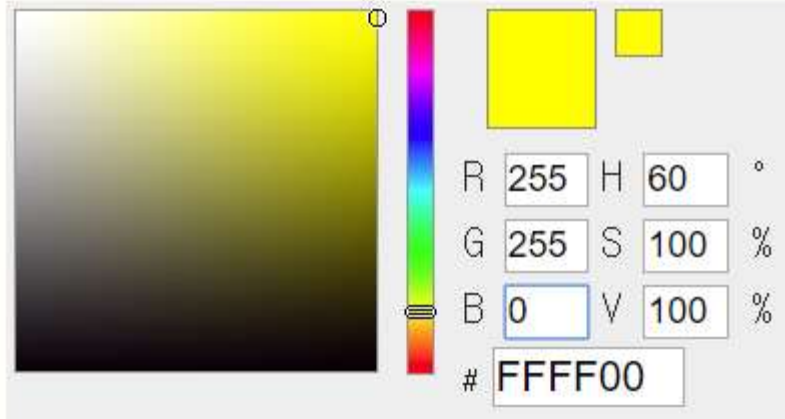
```
48 void ledOutput(int Red, int Green, int Blue){
49     analogWrite(RedLed, Red);
50     analogWrite(GreenLed, Green);
51     analogWrite(BlueLed, Blue);
52 }
```

```
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



[Practice]

◆ [wk04]

- **Arduino LED - I**
- **Complete your project**
- **Submit file : ARnn_Rpt03.zip**

wk04 : Practice-03 : ARnn_Rpt03.zip

◆ [Target of this week]

- Complete your works
- Save your outcomes and compress all.

제출파일명 : ARnn_Rpt03.zip

- 압축할 파일들

① **ARnn_2led.ino**

② **ARnn_4led.fzz**

③ **ARnn_4led.ino**

④ **ARnn_RGB.fzz**

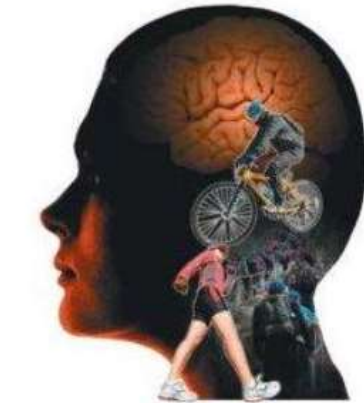
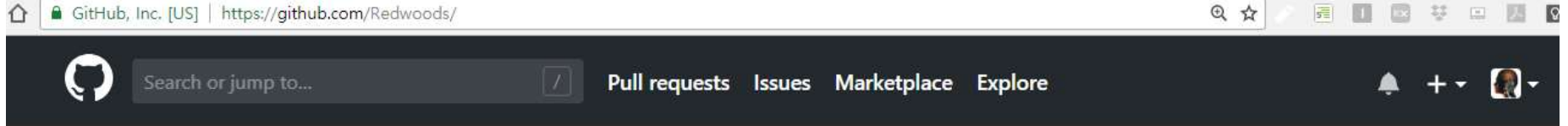
⑤ **Arnn_RGB_Y.png**

Email : chaos21c@gmail.com

[제목 : id, 이름 (수정)]

● 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

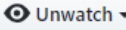
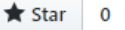

GitHub, Inc. [US] | https://github.com/Redwoods/Arduino

 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 

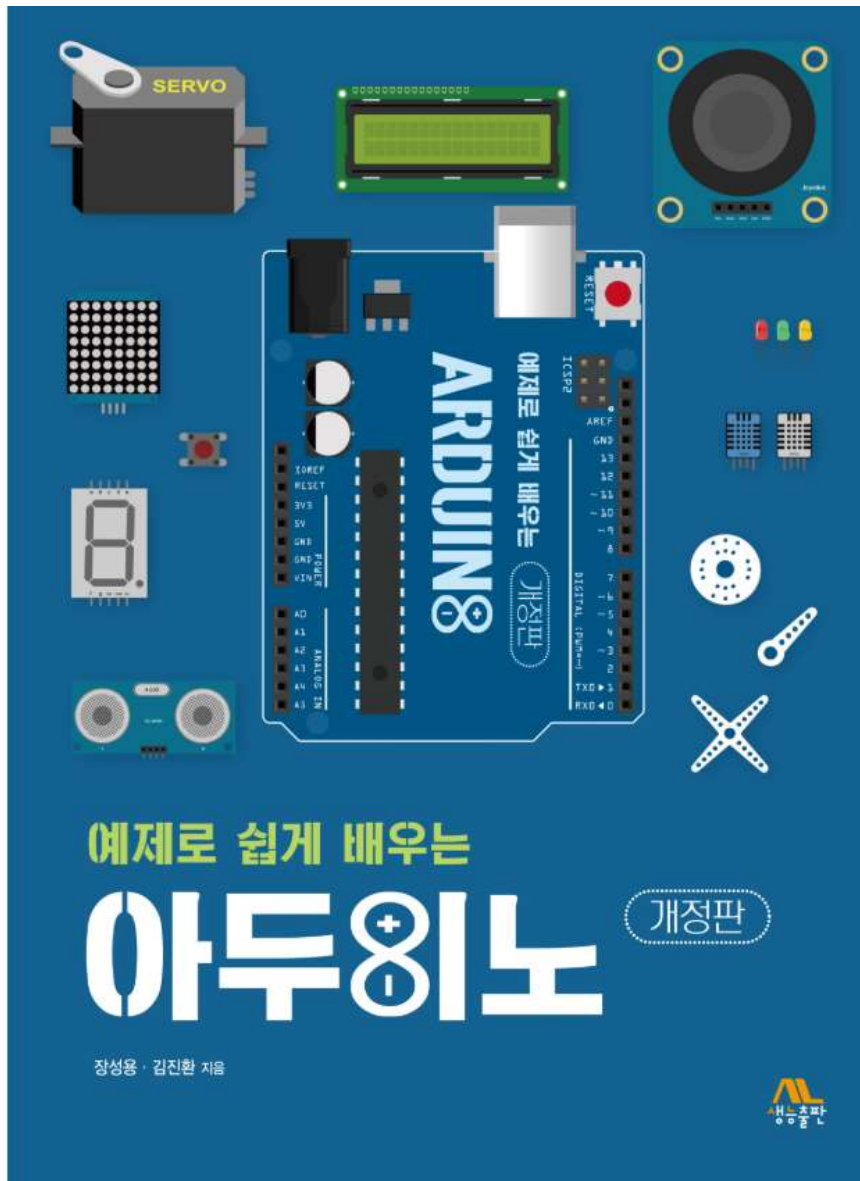
 Redwoods 2018 start Latest commit 38ca9e0 28 minutes ago

 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

상급키트 구성품

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

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

아두이노 키트(Kit) : Part-2

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

[참고 : 저항 값 읽기]



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%

