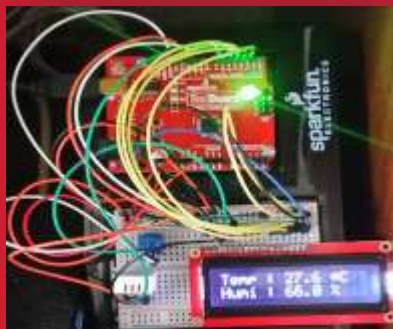


Arduino-basic

[wk04]

LED - I



Learn how to code Arduino from scratch

Comsi, INJE University

1st semester, 2023

Email : chaos21c@gmail.com



My ID (ARnn, github repo)

AR01	강동하
AR02	정재윤
AR03	유석진
AR04	정창민
AR05	정희서
AR06	유동기
AR07	장세진
AR08	정호기

위의 **id**를 이용해서 **github**에 **repo**를 만드시오.

wk03 : Practice-02 : ARnn_Rpt02

◆ [Target of this week]

- Complete your works
- Save your outcomes
- Upload figures & sources

Upload 폴더 명 : ARnn_Rpt02

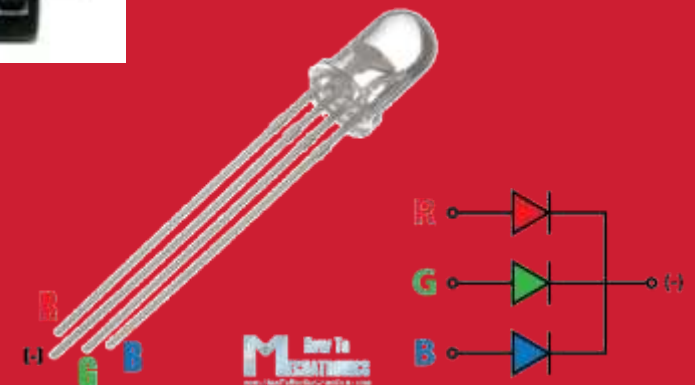
제출할 파일들

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



4. LED I.

Light Emitting Diode

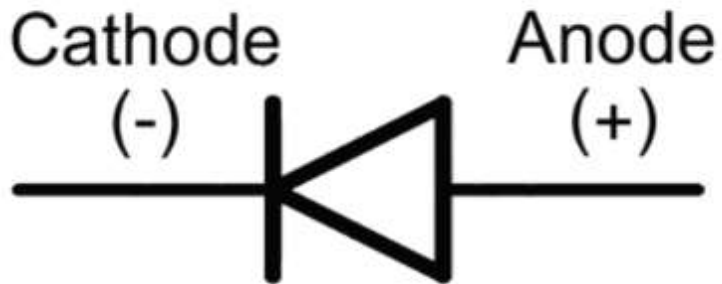


LED (Light Emitting Diode)

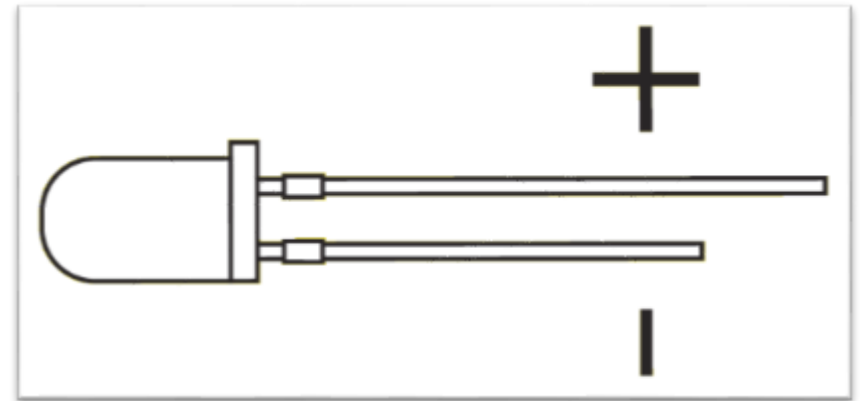
- ✓ 전기 신호를 빛으로 출력하는 반도체 소자
- ✓ 고효율, 반영구적 수명
- ✓ 가정용 실내등, 산업용 특수등, 자동차용 전조등 및 실내등에 사용



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

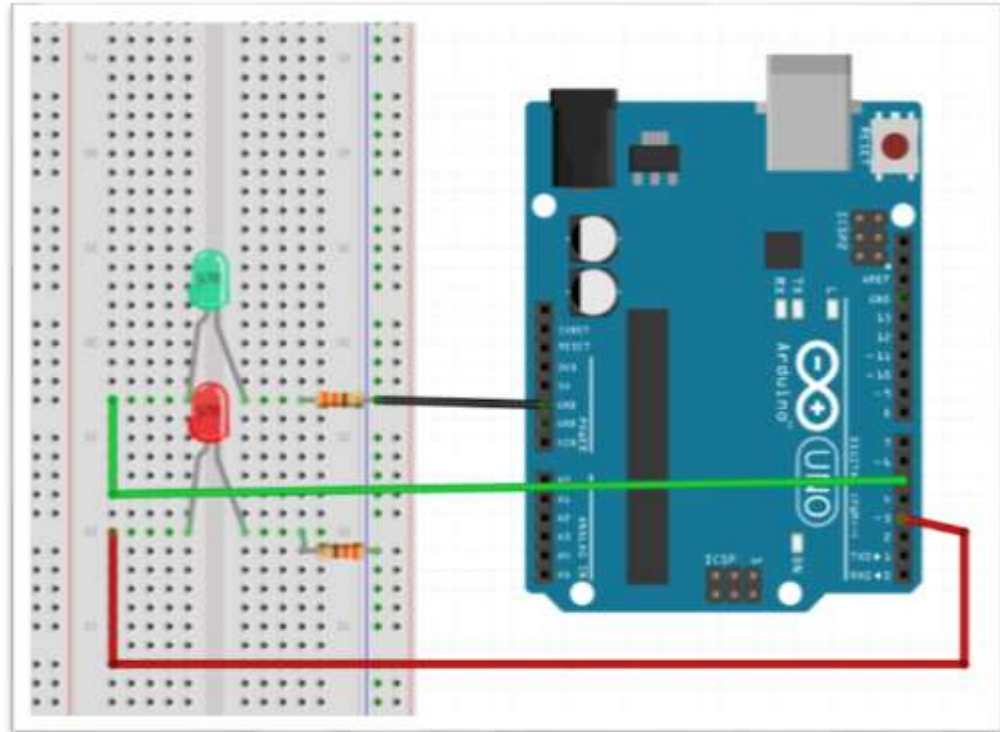
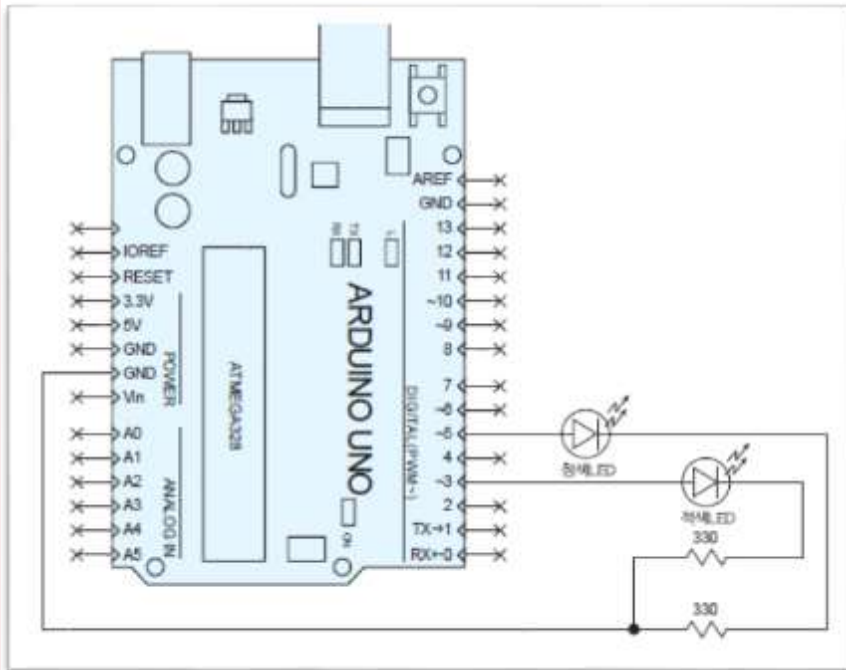


4.1.1 LED control – 교차 점멸

EX 4.1 LED 교차 점멸 (1/3)

실습목표 두 개의 LED를 0.1초 간격으로 교차하여 점멸시키자.

Hardware



Save ARnn_2LED.fzz

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.

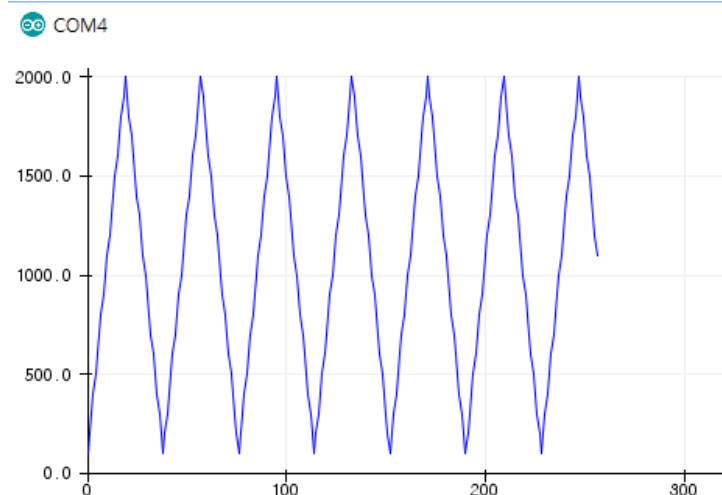
EX 4.1

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.)

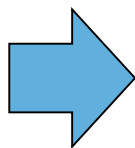
delay = 1600 msec	delay = 500 msec
delay = 1700 msec	delay = 400 msec
delay = 1800 msec	delay = 300 msec
delay = 1900 msec	delay = 200 msec
delay = 2000 msec	delay = 100 msec
delay = 1900 msec	delay = 200 msec
delay = 1800 msec	delay = 300 msec
delay = 1700 msec	delay = 400 msec
delay = 1600 msec	delay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



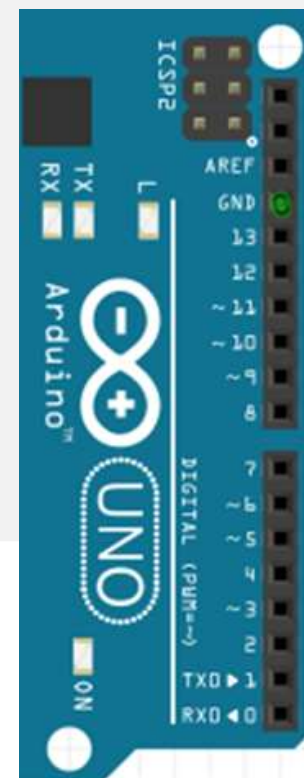
```
6 const int ledA = 3;
7 const int ledB = 5;
8
9 int number = 1;
10 boolean flag = true;
11
12 void setup()
13 {
14   Serial.begin(9600);
15   pinMode(ledA, OUTPUT);
16   pinMode(ledB, OUTPUT);
17 }
```

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.



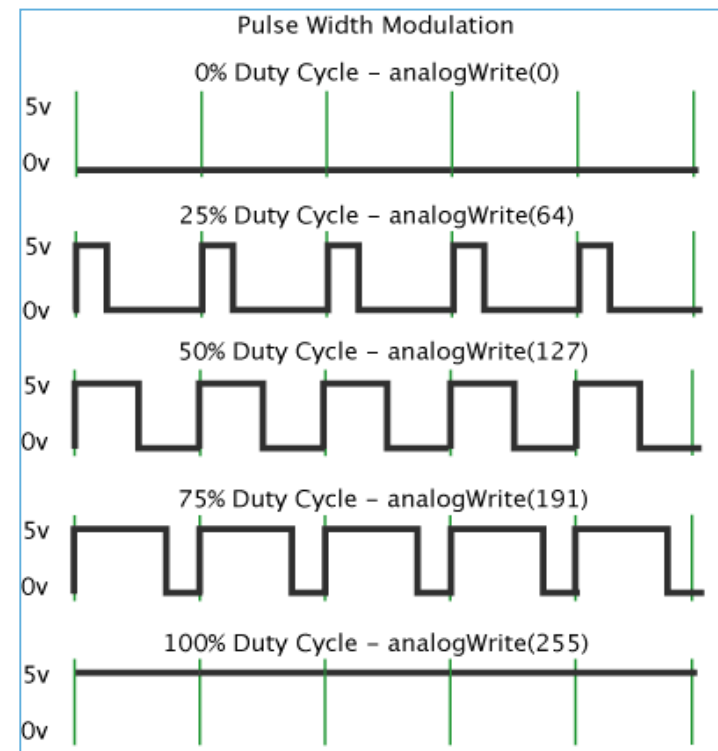
4.2.1 LED control – 밝기 조절

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

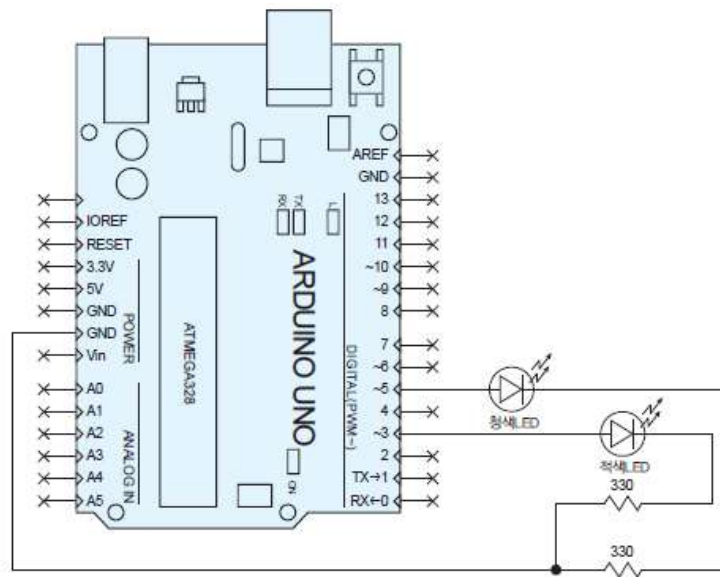


<https://www.arduino.cc/en/Tutorial/PWM>

EX 4.2 LED 밝기 조절 (1/2)

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

- Hardware**
1. 청색과 적색 LED의 Anode핀을 Arduino의 3번 5번 핀에 연결한다.
 2. Cathode핀에 330Ω(또는 220Ω)저항을 연결하여 저항의 반대쪽은 Arduino의 GND에 연결한다.
 3. LED가 연결된 핀에 HIGH신호가 출력될 때 LED가 점등된다.



4.2.3 LED control – 밝기 조절

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

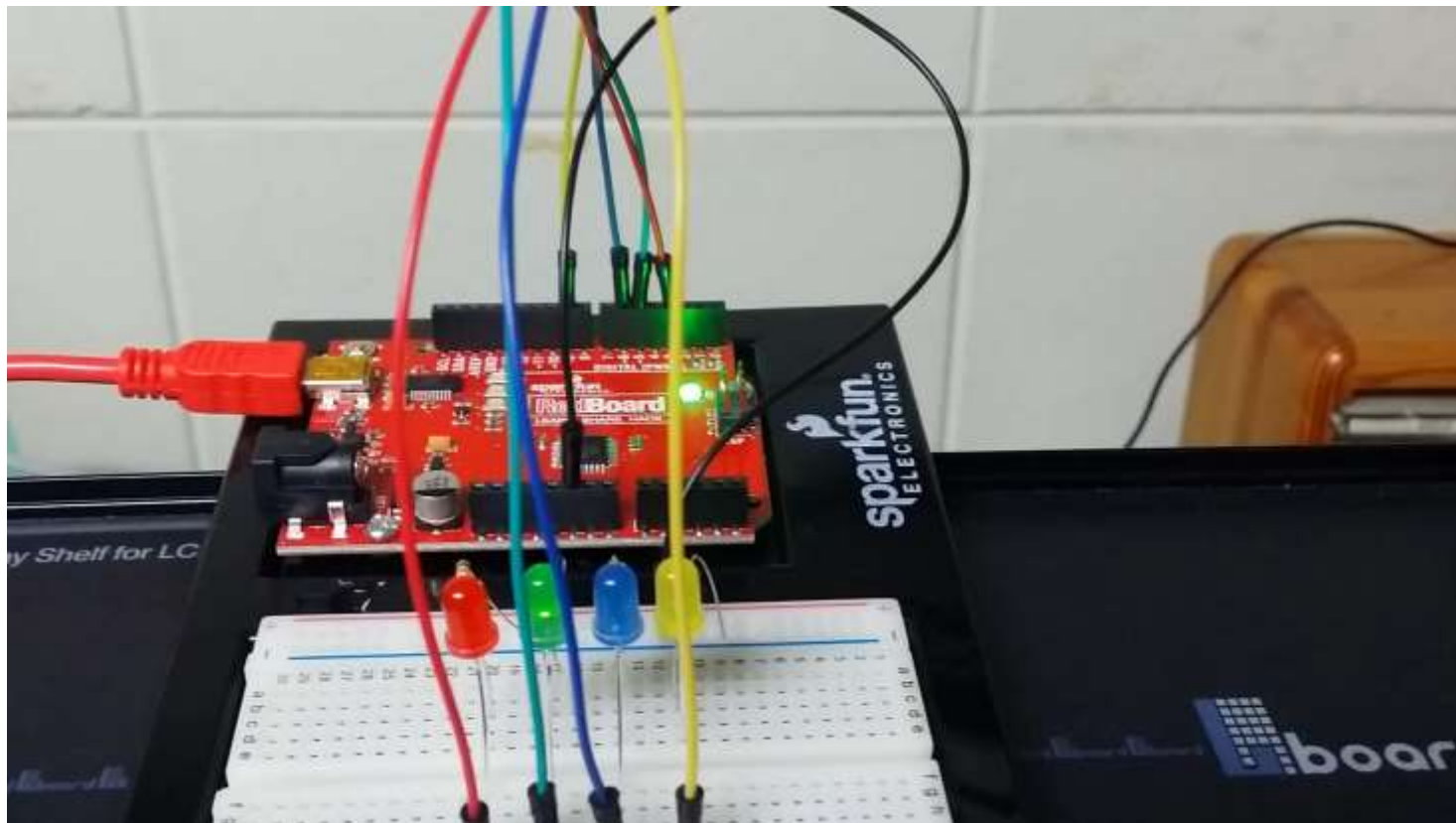
```

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 }

```

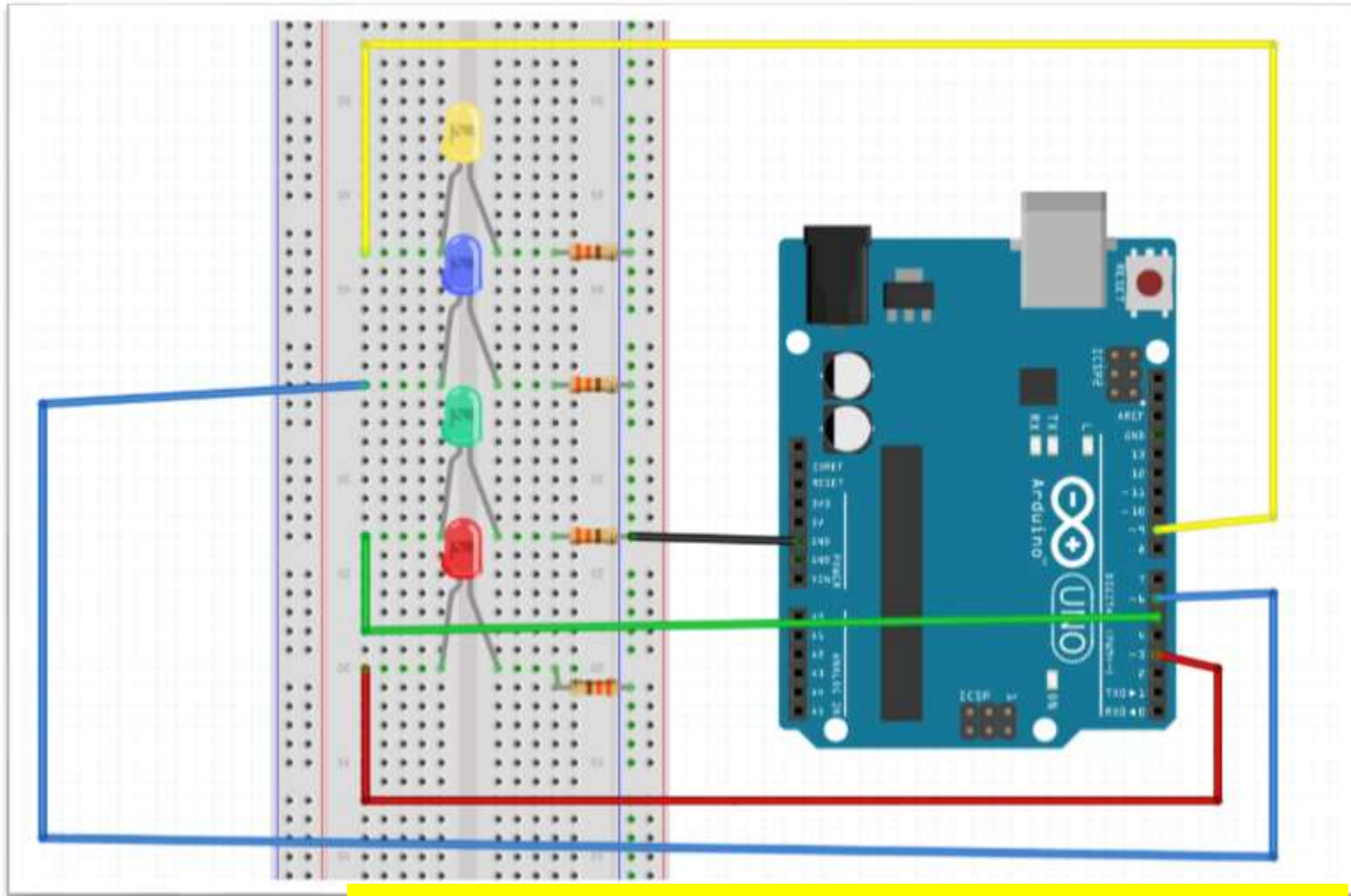
4.2.5 LED control – DIY

- 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   }

```

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

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



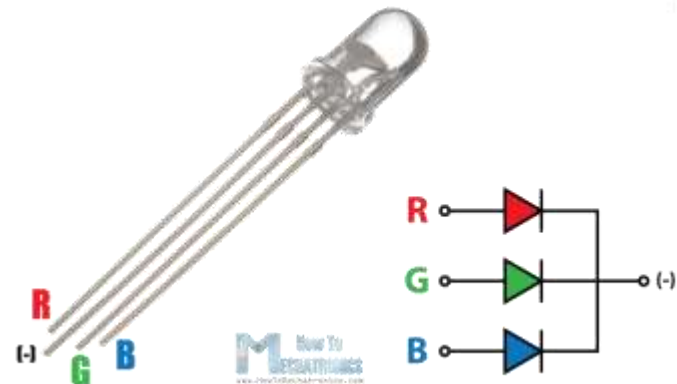
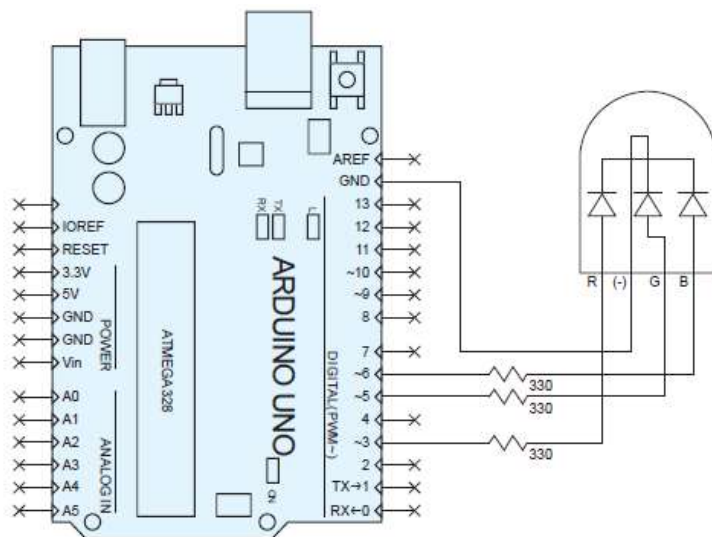
4.3.1 RGB LED control – 색상 조절

EX 4.3

RGB LED로 색상 표현하기 (1/2)

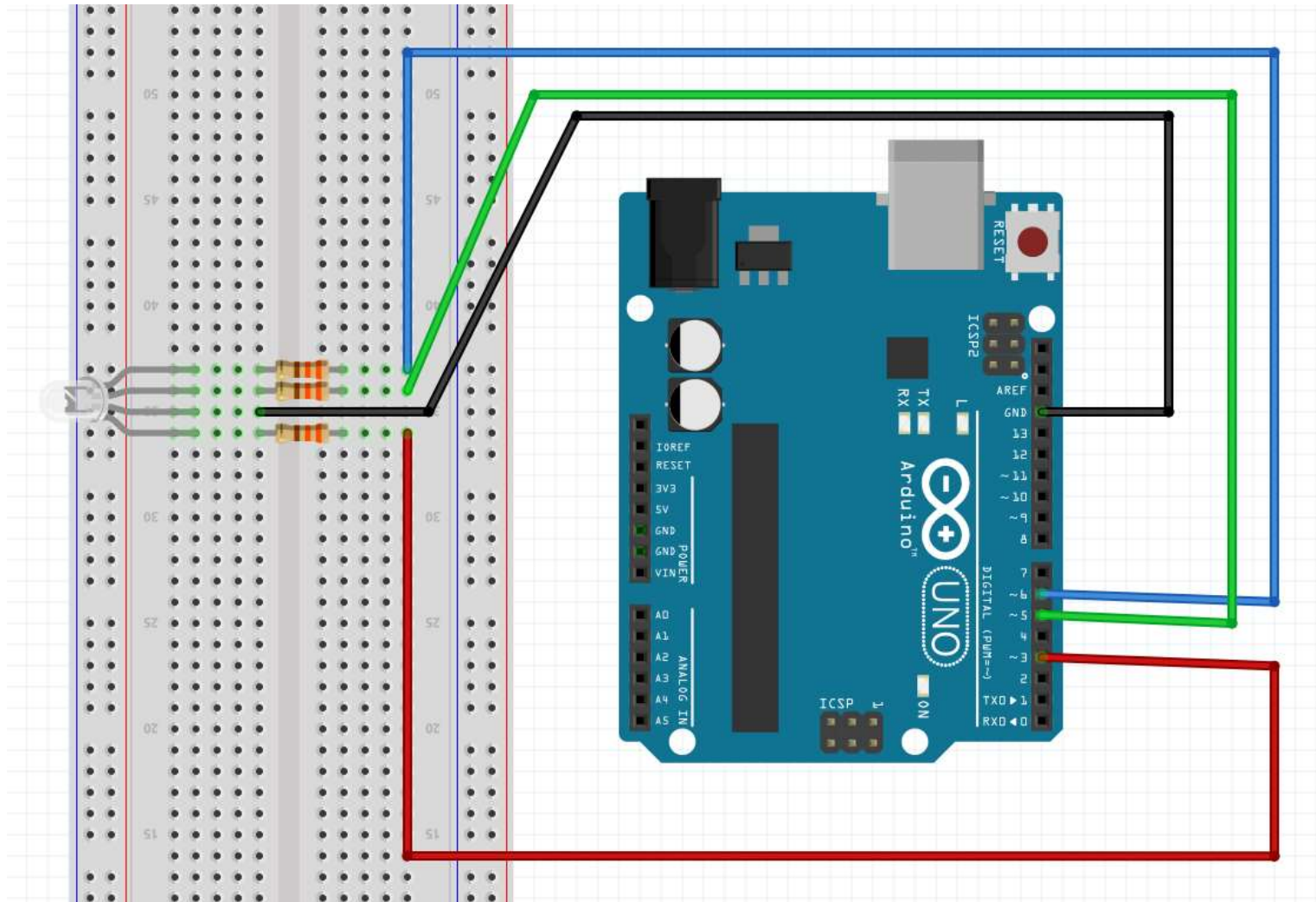
실습목표 RGB LED를 이용하여 다양한 색을 표현해 보자.

- Hardware**
1. RGB LED는 Red, Green, Blue의 세 개의 Anode 핀과 공통으로 연결된 캐소드핀으로 구성되어 있다.
 2. RGB LED 단독으로 연결하려면 **각 Anode 핀에 330Ω(또는 220Ω)의 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



ARnn_RGB.fzz

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
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 **code**를
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 }

```

```

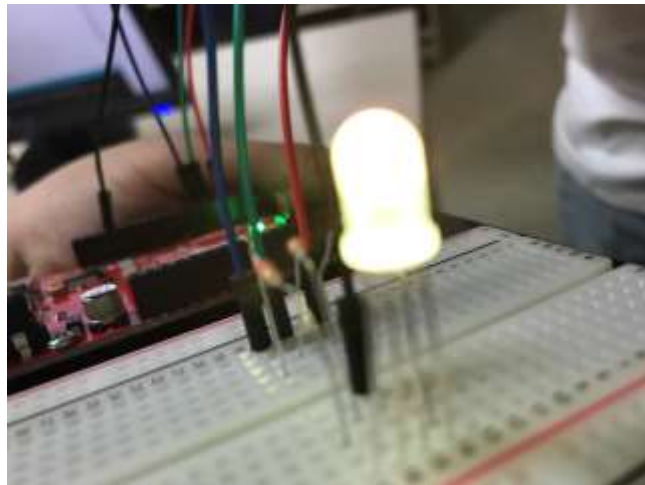
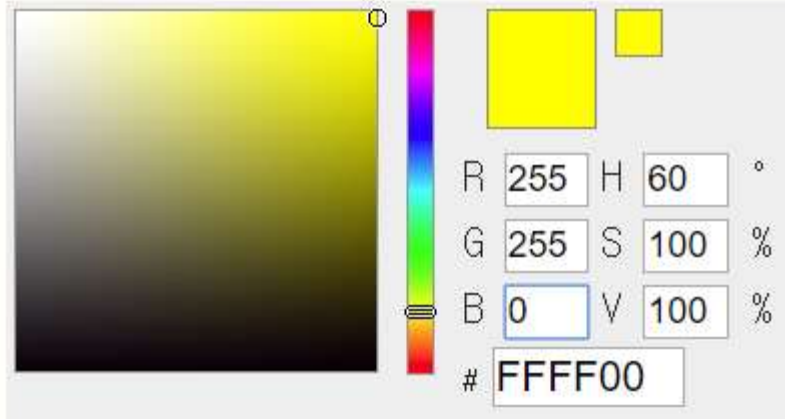
20 void loop()
21 {
22   for(int i=0; i<=255; ++i){
23     ledOutput(255, i, 0);
24     delay(10);
25   }
26   for(int i=0; i<=255; ++i){
27     ledOutput(0, 255, i);
28     delay(10);
29   }
30   for(int i=0; i<=255; ++i){
31     ledOutput(i, 0, 255);
32     delay(10);
33   }
34   for(int i=0; i<=255; ++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

4.4 RGB LED control – 색상 조절 [DIY]

DIY. RGB LED의 색이 노란색일 때 사진을 촬영하시오.



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

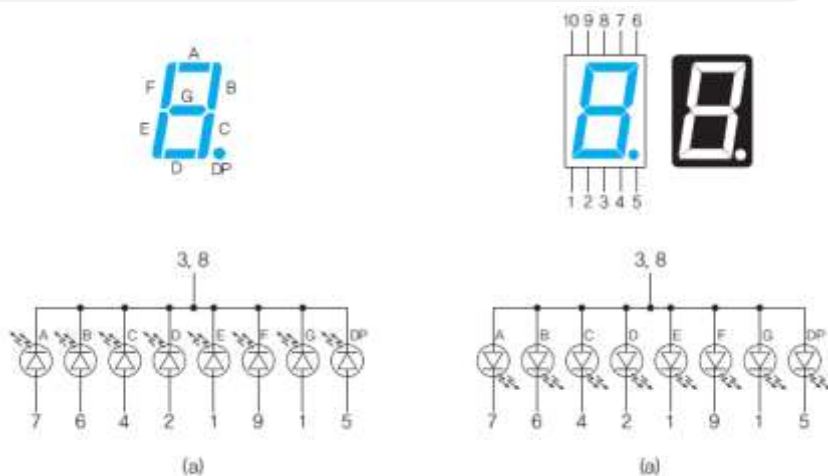
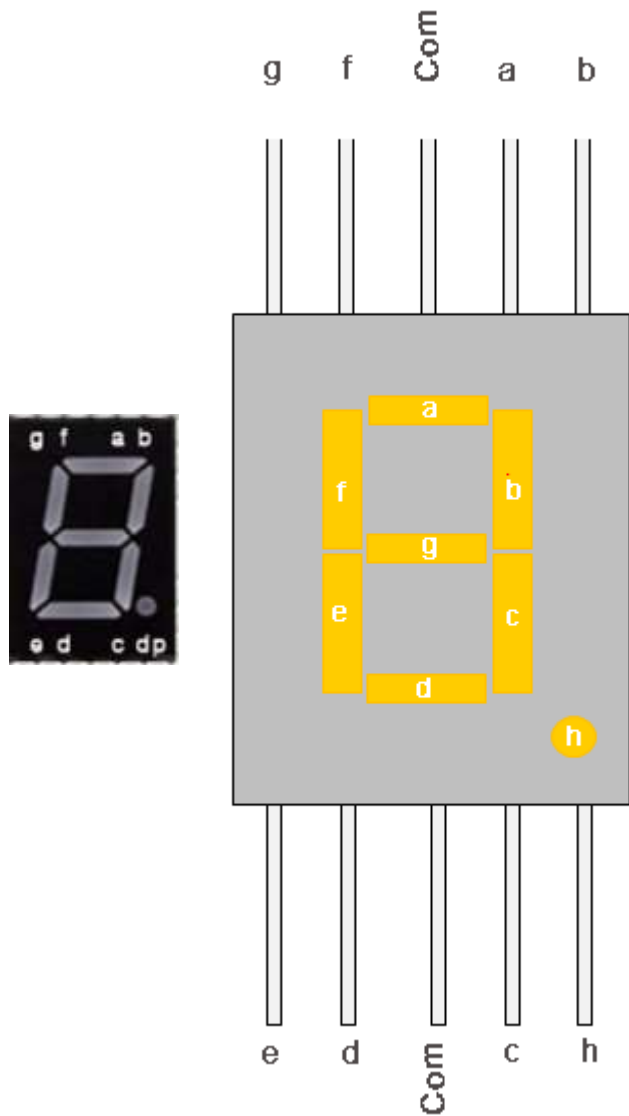


그림 4.4 Common Cathode 형(a)와 Common Anode 형(b)

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		
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**
- **Upload folder : ARnn_Rpt03**

wk04 : Practice-03 : ARnn_Rpt03

◆ [Target of this week]

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

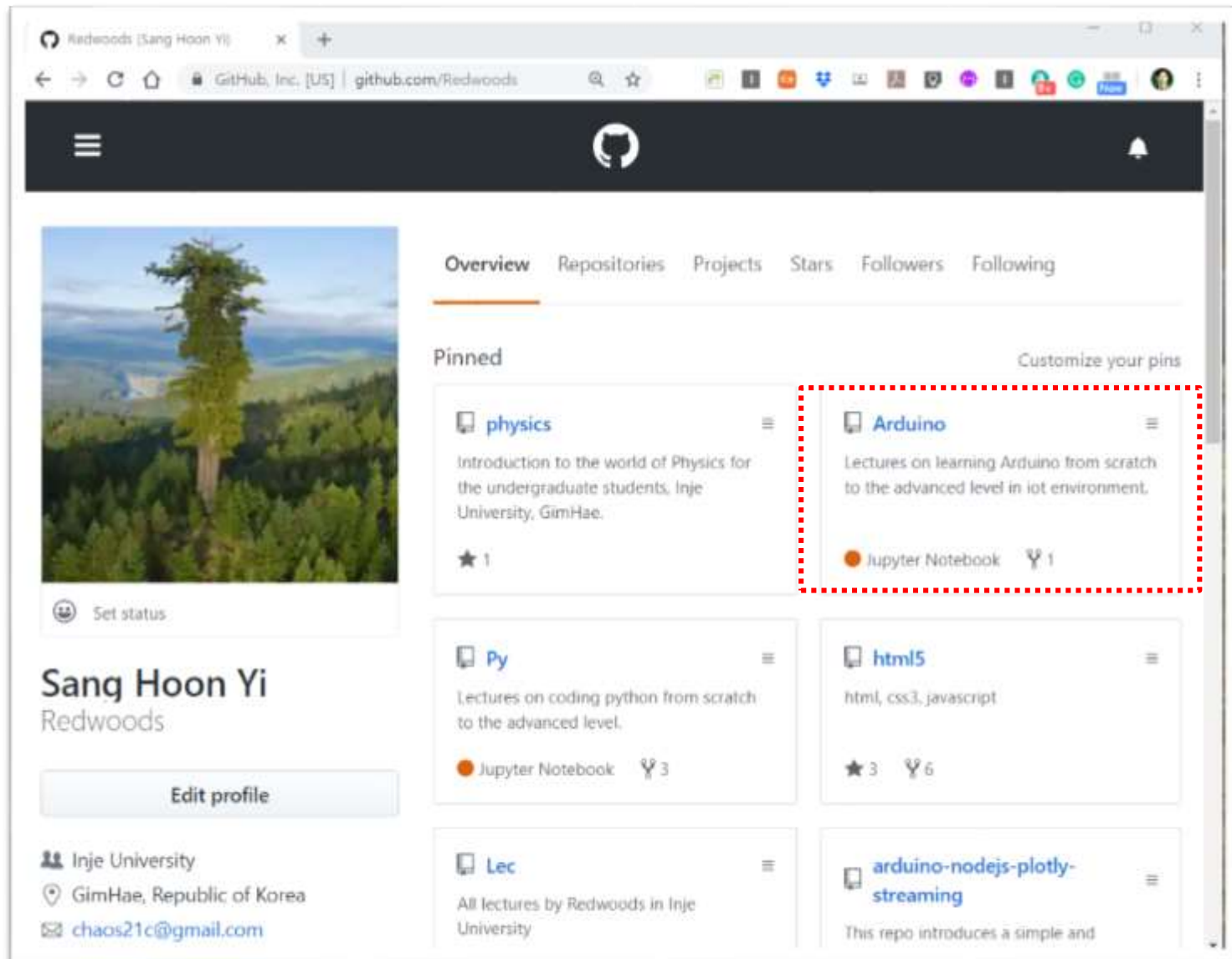
Upload 폴더 명 : ARnn_Rpt03

제출할 파일들

- ① **ARnn_2led.ino**
- ② **ARnn_4led.fzz**
- ③ **ARnn_4led.ino**
- ④ **ARnn_RGB.ino**
- ⑤ **ARnn_RGB_Y.png**

● 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



Redwoods (Sang Hoon Yi)

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

Overview Repositories Projects Stars Followers Following

Pinned Customize your pins

physics
Introduction to the world of Physics for the undergraduate students, Inje University, GimHae.
★ 1

Arduino
Lectures on learning Arduino from scratch to the advanced level in iot environment.
Jupyter Notebook 🍴 1

Py
Lectures on coding python from scratch to the advanced level.
Jupyter Notebook 🍴 3

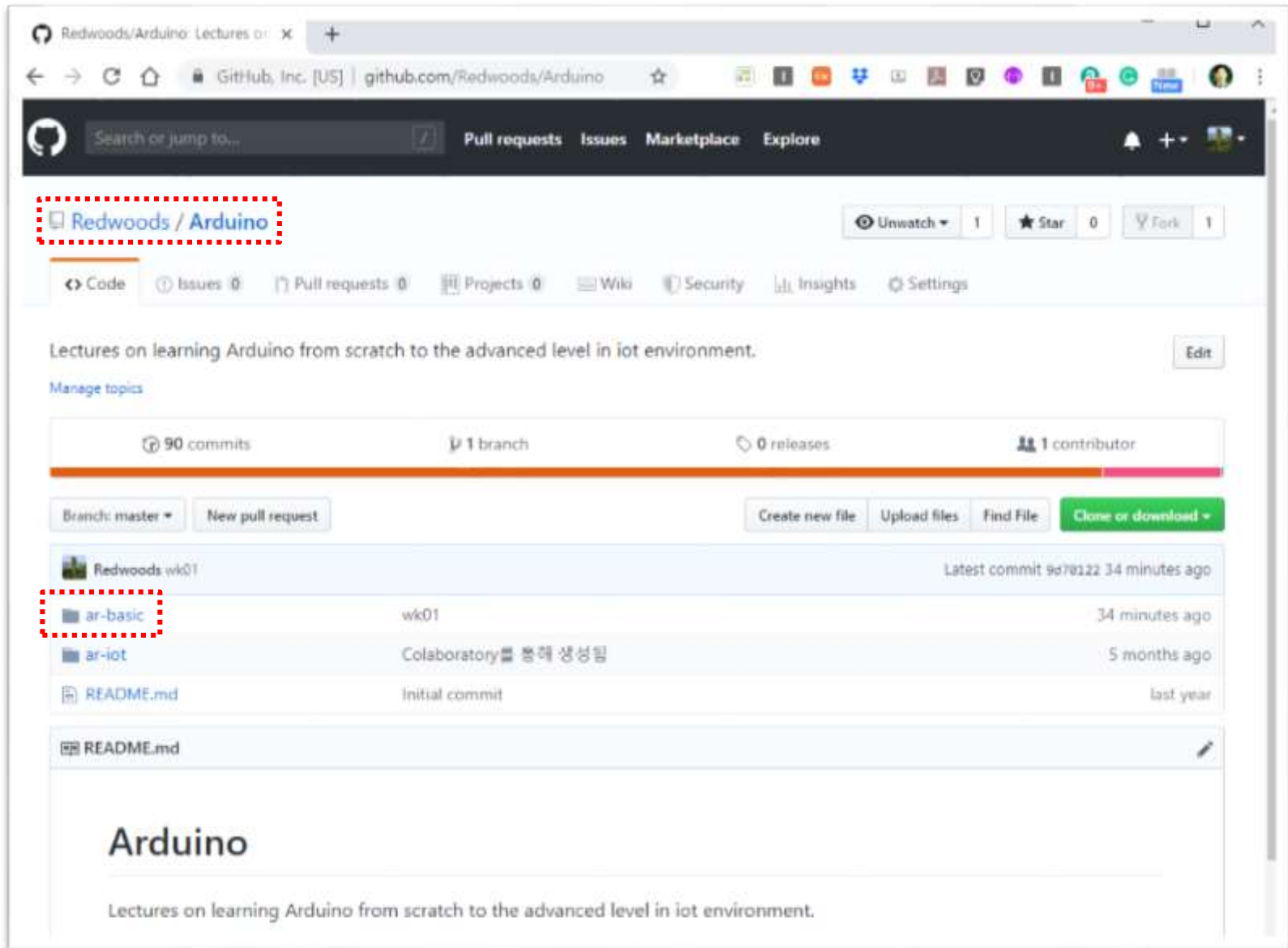
html5
html, css3, javascript
★ 3 🍴 6

Lec
All lectures by Redwoods in Inje University

arduino-nodejs-plotly-streaming
This repo introduces a simple and

Sang Hoon Yi
Redwoods
Edit profile

Inje University
GimHae, Republic of Korea
chaos21c@gmail.com



The screenshot shows the GitHub repository page for **Redwoods/Arduino**. The repository description is "Lectures on learning Arduino from scratch to the advanced level in iot environment." The repository has 90 commits, 1 branch, 0 releases, and 1 contributor. The file list includes **ar-basic** (34 minutes ago), **ar-iot** (5 months ago), and **README.md** (last year). The **ar-basic** file is highlighted with a red dashed box. The repository is currently on the **master** branch.

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

Manage topics

90 commits 1 branch 0 releases 1 contributor

Branch: master New pull request Create new file Upload files Find File Clone or download

File	Commit	Time
Redwoods (wk01)	Latest commit 9d70122	34 minutes ago
ar-basic	wk01	34 minutes ago
ar-iot	Colaboratory를 통해 생성됨	5 months ago
README.md	Initial commit	last year

Arduino

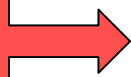
Lectures on learning Arduino from scratch to the advanced level in iot environment.



아두이노 키트(Kit)



web



<https://www.devicemart.co.kr/goods/view?no=12170416>

아두이노 레벨업키트(골드) 구성품

						
아두이노 UNO	USB 케이블	830핀 브레드보드	미니 브레드보드	점퍼와이어 세트	듀폰케이블 M/F	듀폰케이블 M/M
						
저항 220Ω	저항 1KΩ	저항 10KΩ	가변저항 10KΩ	빨강 LED	녹색 LED	파랑 LED
						
노랑 LED	RGB LED (CA)	RGB LED 모듈	1digit FND (CA)	4digit FND (CA)	8x8 도트 매트릭스	택트 스위치
						
택트 스위치 캡	볼 스위치	리드 스위치 센서	4x4키 매트릭스	5V 릴레이 모듈	조이스틱 모듈	수위 센서
						
온도센서 LM35	저미스터	습도센서	CDS 조도센서	불꽃감지센서	적외선 수신기	IR 리모콘

아두이노 키트(Kit) : Part-2

						
TCRT5000 적외선 센서	인체감지센서 모듈	사운드센서	능동부저	수동부저	초음파센서	I2C 1602 LCD 모듈
						
서보모터	스텝모터	스텝모터드라이버	RFID 수신 모듈	RFID 카드	RFID 태그	DS1302 RTC 모듈
						
1N4001 다이오드	2N2222 트랜지스터	74HC595	1x40 핀헤더	9V 배터리 스냅	아크릴 고정판	

■ 아두이노 UNO × 1	■ USB 케이블 × 1	■ 830핀브레드보드 × 1	■ 미니 브레드보드 × 1	■ 점퍼와이어세트 × 1
■ 듀폰케이블 × 80 (M/F,M/M)	■ 저항 × 30	■ 가변저항 × 1	■ LED × 20	■ RGB LED × 1
■ RGB LED 모듈 × 1	■ 1digit FND(CA) × 1	■ 4digit FND(CA) × 1	■ 8×8도트 매트릭스 × 1	■ 탭스위치 × 5
■ 탭스위치 캡 × 5	■ 볼스위치 × 1	■ 리드 스위치 센서 × 1	■ 4×4 키 매트릭스 × 1	■ 5V 릴레이 모듈 × 1
■ 조이스틱 모듈 × 1	■ 수위 센서 × 1	■ 온도센서 LM35 × 1	■ 써미스터 × 1	■ 온습도센서 × 1
■ CdS 조도센서 × 1	■ 불꽃감지센서 × 1	■ 적외선 수신기 × 1	■ IR 리모컨 × 1	■ TCRT5000 적외선 센서 × 1
■ 인체감지센서 모듈 × 1	■ 사운드센서 × 1	■ 능동부저 × 1	■ 수동부저 × 1	■ 초음파센서 × 1
■ I2C 1602 LCD 모듈 × 1	■ 서보모터 × 1	■ 스텝모터 × 1	■ 스텝모터드라이버 × 1	■ RFID 수신 모듈 × 1
■ RFID 카드 × 1	■ RFID 태그 × 1	■ DS1302 RTC 모듈 × 1	■ 1N4001 다이오드 × 1	■ 2N2222 트랜지스터 × 1
■ 74HC595 × 1	■ 1x40 핀헤더 × 1	■ 9V 배터리 스냅 × 1	■ 아크릴 고정판 × 1	