

03주차

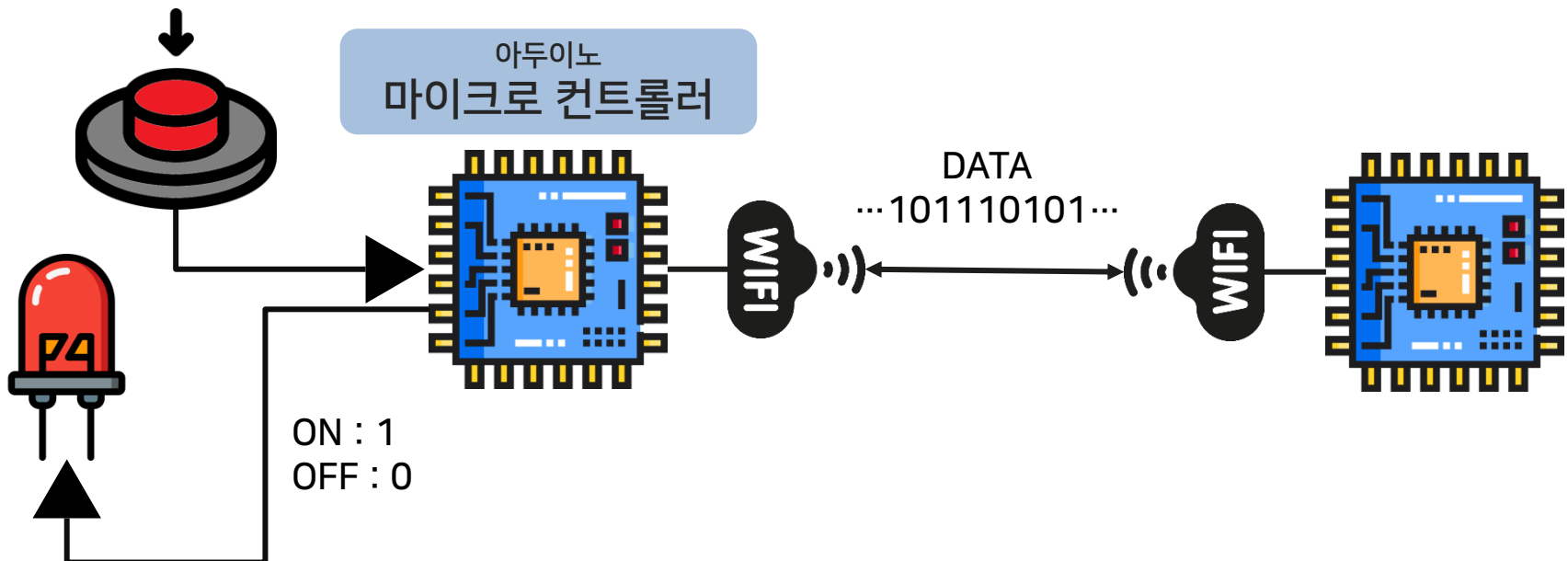
# 아두이노 출력모듈:LED와 RGB

# 목차

- 0. 모듈 종류
- 1. LED 모듈 연결
- 2. LED 다중 연결
- 3. RGB 모듈 연결

# 모듈 종류

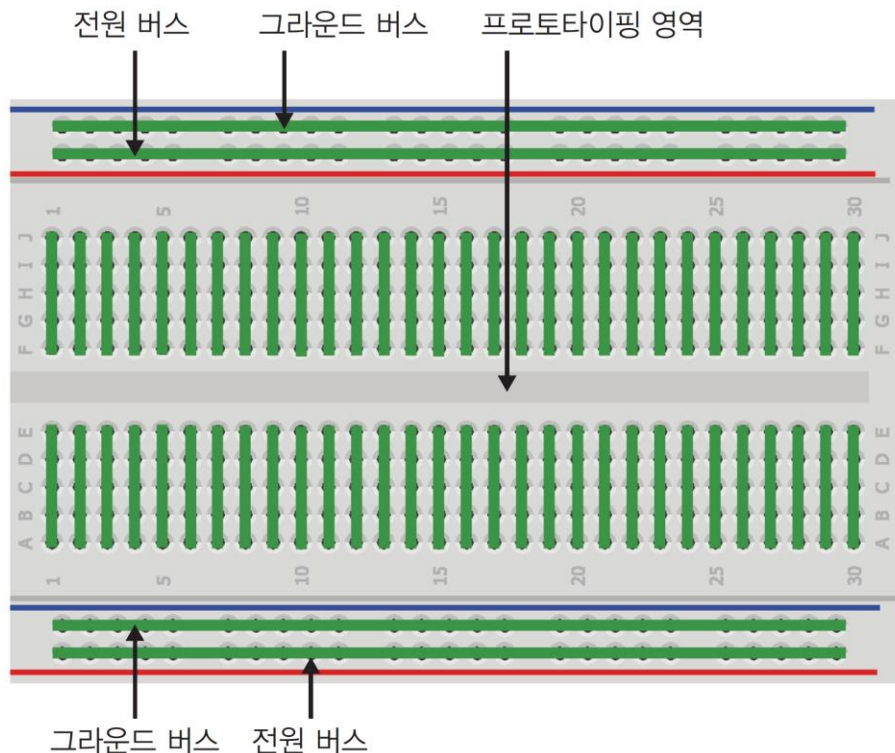
- 출력 모듈: 컴퓨터가 생성한 정보를 컴퓨터의 외부로 전송하는 데에 사용되는 장치
  - LED, 스피커 등
- 입력모듈: 신호 등을 발생시켜 컴퓨터로 전송하는데 사용되는 설비나 장치
  - 스위치, 온도, 기울기 센서 등
- 통신모듈: 외부 컴퓨터와 데이터를 송수신하는데 사용되는 설비나 장치
  - Ethernet, wifi, 블루투스 등



# 01 LED 모듈 연결

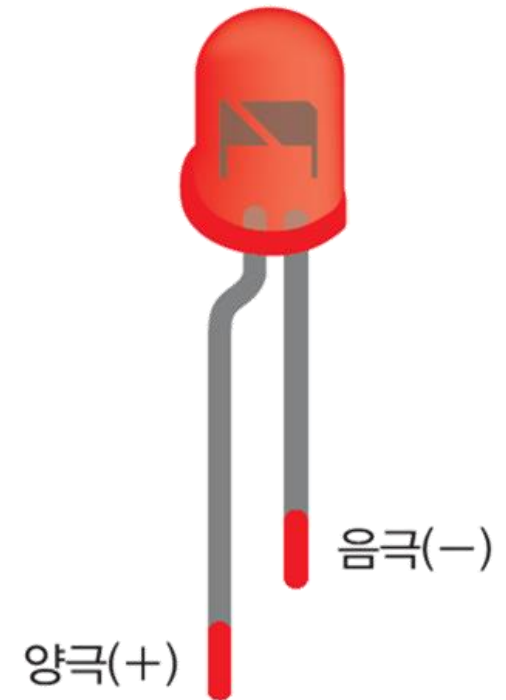
# 필요한 도구 : 브레드 보드

- 부품을 납땜하지 않고도 회로 구성을 가능하게 하는 프로토타이핑 도구
- 양쪽에 파란색(그라운드)과 빨간색(전원 버스) 홀은 연결된 상태
- 수직 홀은 5개 그룹으로 연결된 상태
- 각 핀이 내부적으로 연결되어 있어 각 전기/전자 부품을 연결하기 편리함



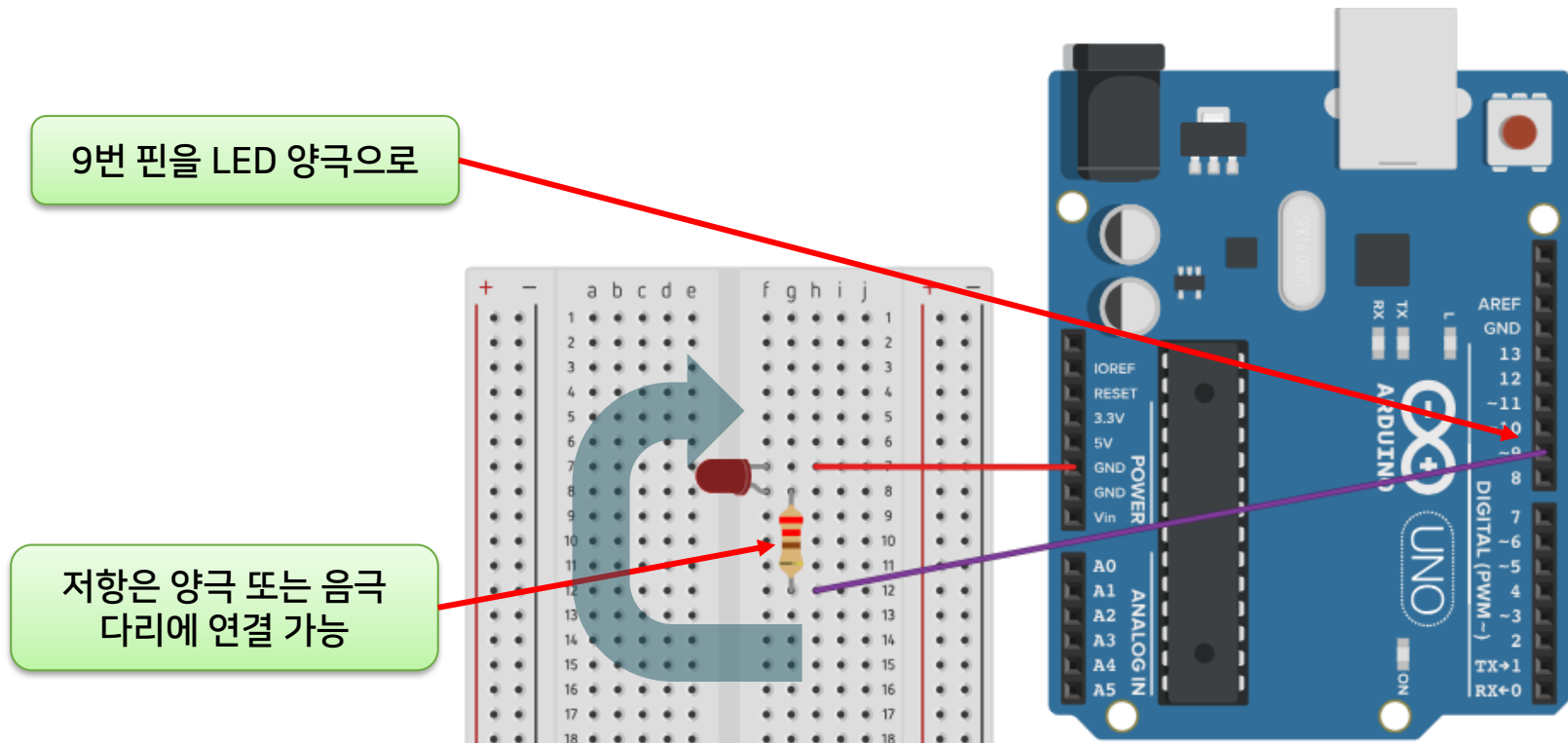
# 필요한 도구 : LED(Light Emitting Diode, 발광 다이오드)

- 양극(+, 애노드)에서 음극(-, 캐소드)방향으로만 전류가 흐름
  - 따라서 +, - 방향을 고려하여 연결해야 한다.
- 양극과 음극의 2개 다리를 가짐
  - 다리 길이가 긴 쪽이 양극
- 과도한 전류가 흐르지 않도록 전류 제한 필요
  - $200\Omega$  전후 저항이 일반적으로 사용됨



# 아두이노에 LED 연결하기

- 아래 그림과 같이 LED를 아두이노에 연결한다.
  - 일반적인 LED는 2V의 전압과 2mA의 전류를 소모하기 때문에 5V의 아두이노 전압을 낮추기 위해 저항을 사용해야함
  - 220Ω 저항을 LED에 직렬로 연결
- 저항을 사용하지 않고 LED를 직접 연결하면 LED가 타버리거나 손상될 수 있음



# LED 예제 함수

## `void pinMode(uint8_t pin, uint8_t mode)`

- 매개변수
  - pin : 설정하고자 하는 핀 번호
  - mode : INPUT, OUTPUT, INPUT\_PULLUP 중 하나
- 반환값 : 없음

## `void digitalWrite(uint8_t pin, uint8_t value)`

- 매개변수
  - pin : 핀 번호
  - value : HIGH(1) 또는 LOW(0)    => 디지털 출력: 1 또는 0으로만 표현
- 반환값 : 없음

## `void delay(unsigned long ms)`

- 매개변수
  - ms : 밀리초 단위의 지연 시간
- 반환값 : 없음



# uno\_ex3-1: LED 모듈 예제

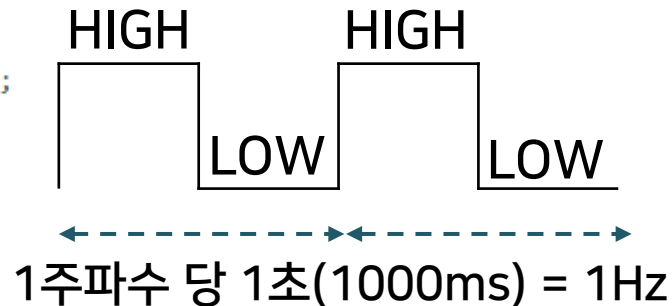
- 아두이노의 9번 핀에 연결된 LED를 켜기 위해, 9번 핀을 출력 모드로 설정

```
1  const int LED=9; //9번 핀을 사용하는 LED 상수 정의
2
3  void setup()
4  {
5      pinMode(LED, OUTPUT); //LED(9번 핀)를 출력으로 지정
6      digitalWrite(LED, HIGH); //LED를 HIGH로 지정 → 9번 핀을 5V로 설정 → LED 켜짐
7  }
8
9  void loop()
10 {
11     //이 실습에서는 loop문을 비움
12 }
```

디지털 9번 핀 (5V) → 저항 (220Ω) → LED (+극 → -극) → GND → 아두이노 내부 회로 → 다시 전원 공급 회로로 반환

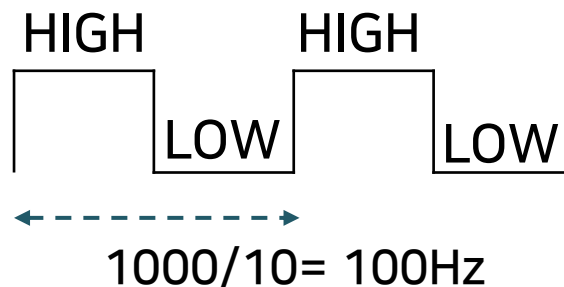
## uno\_ex3-2: LED 점멸

```
1  const int LED=9; //9번 핀을 사용하는 LED 상수 정의
2
3  void setup()
4  {
5      pinMode(13, OUTPUT); //LED(13번 핀)를 출력으로 지정
6      digitalWrite(13, LOW); //13핀 OFF
7
8      pinMode(LED, OUTPUT); //LED(9번 핀)를 출력으로 지정
9  }
10
11 void loop()
12 {
13     digitalWrite(LED, HIGH);
14     delay(500);
15     digitalWrite(LED, LOW);
16     delay(500);
17 }
18
```



### loop함수 변경

```
digitalWrite(LED, HIGH);
delay(5);
digitalWrite(LED, LOW);
delay(5);
```



파형이 초당 50개 이상이 되면, 즉 50Hz이상의 주파수로 LED 점멸을 반복하면 인간의 눈으로는 깜빡거림을 감지할 수 없음

## uno\_ex3-3: 디지털 출력으로 밝기 조절

### [ 어둡게 출력 ]

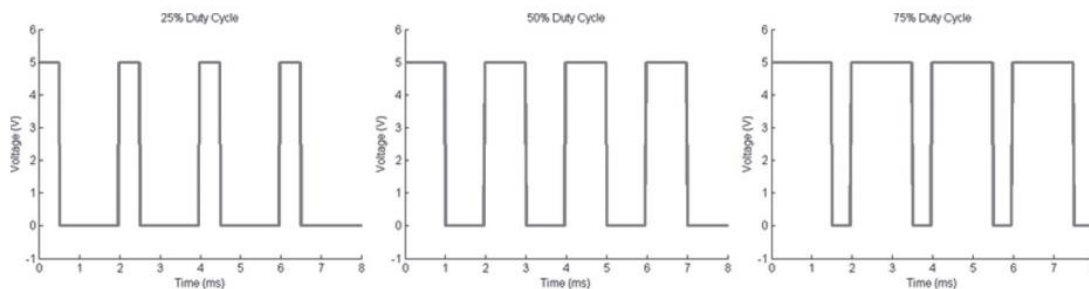
```
1  const int LED=9; //9번 핀을 사용하는 LED 상수 정의
2
3  void setup()
4  {
5      pinMode(LED, OUTPUT); //LED(9번 핀)를 출력으로 지정
6  }
7
8  void loop()
9  {
10     digitalWrite(LED, HIGH); //LED를 HIGH로 지정
11     delay(1);
12     digitalWrite(LED, LOW); //LED를 LOW로 지정
13     delay(9);
14 }
```

### [ 밝게 출력 ]

```
1  const int LED=9; //9번 핀을 사용하는 LED 상수 정의
2
3  void setup()
4  {
5      pinMode(LED, OUTPUT); //LED(9번 핀)를 출력으로 지정
6  }
7
8  void loop()
9  {
10     digitalWrite(LED, HIGH); //LED를 HIGH로 지정
11     delay(9);
12     digitalWrite(LED, LOW); //LED를 LOW로 지정
13     delay(1);
14 }
```

# 아날로그 출력

- 아날로그 출력 : ~~0~5V(3.3V 보드는 0~3.3V)~~ 사이의 전압 0~255의 값을 자유롭게 내보내는 기능
- PWM(Pulse Width Modulation, 펄스폭변조)
  - 순수 아날로그 제어를 모방한 것으로 다양한 환경에서 사용
  - PWM을 사용하면 디지털 신호를 아날로그 신호와 비슷하게 생성
  - PWM은 신호를 켜거나 끄는 방식으로 네모파의 듀티비(Duty Cycle)를 변조하여 사용
    - 듀티비 : 전체 주기에서 HIGH상태가 차지하는 비율
  - 아두이노 Uno는 3, 5, 6, 9, 10, 11 번 핀에서 PWM 출력이 가능(반드시 ~표기 확인)



문제: LED를 통과하는  
전류의 전압은 낮추지 않은 채  
LED의 밝기를 어둡게 하려면 어떻게 해야 할까?

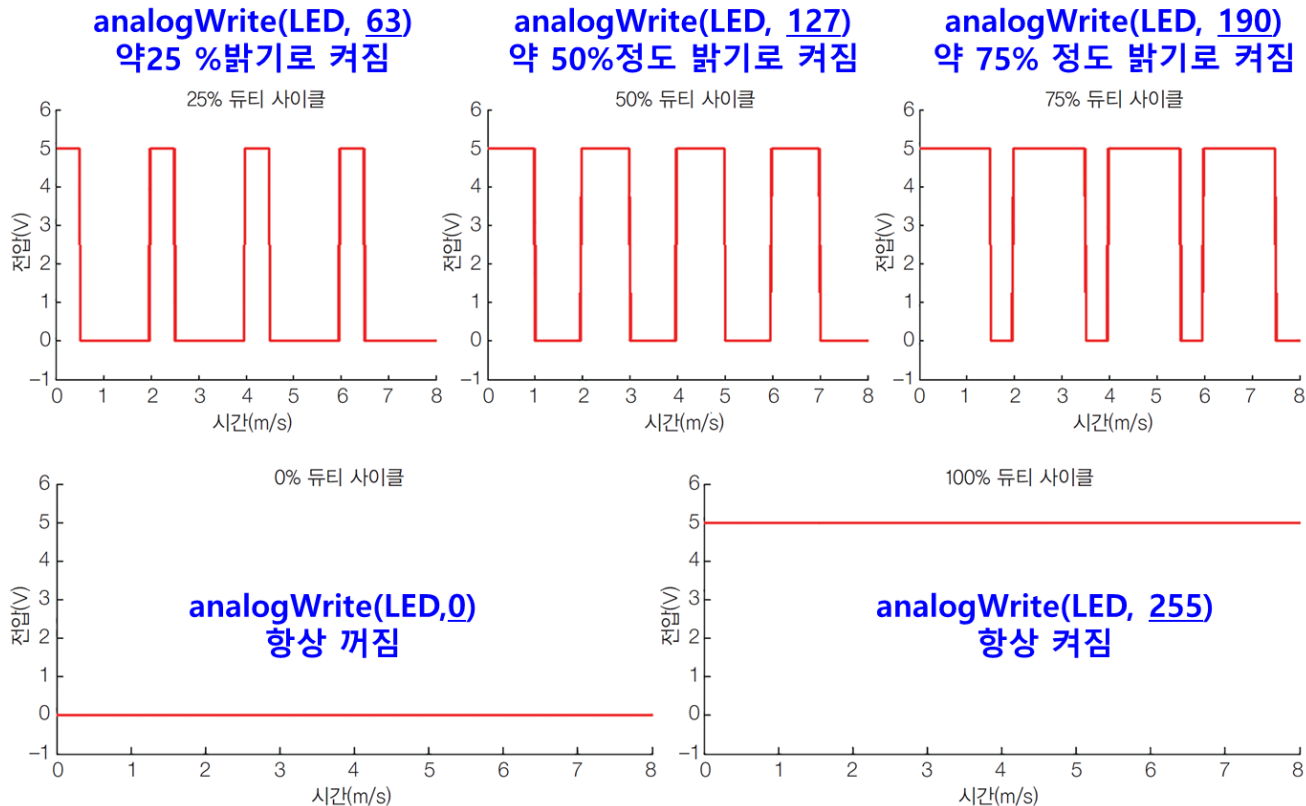
### ■ 해결 방법

- PWM으로 LED를 빠른 속도로 켜고 끌 수 있다면(듀티비 50%) LED는 중간 정도의 밝기로 켜져 있는 것처럼 보일 것이다. 우리 눈이 지각할 수 있는 것보다 LED가 더 빠르게 깜빡이기 때문이다.

# analogWrite 함수

## ■ analogWrite() 함수는 두 번째 인자 값에 따라 네모파의 듀티비를 지정

- 0을 지정하면 네모파의 듀티비는 0%(항상 LOW)
- 255를 지정하면 네모파의 듀티비는 100%(항상 HIGH)
- 127을 지정하면 네모파의 듀티비는 50%(반은 HIGH, 반은 LOW)



## uno\_ex3-4: 아날로그 출력으로 밝기 조절

```
1  const int LED=9; //9번 핀을 사용하는 LED 상수 정의
2
3  void setup()
4  {
5      pinMode(LED, OUTPUT); //LED(9번 핀)를 출력으로 지정
6  }
7
8  void loop()
9  {
10     for(int i=0; i<256; i++)
11     {
12         analogWrite(LED, i);
13         delay(10);
14     }
15     for(int i=255; i>=0; i--)
16     {
17         analogWrite(LED, i);
18         delay(10);
19     }
20 }
```

## uno\_ex3-5: 사용자 입력으로 아날로그 출력

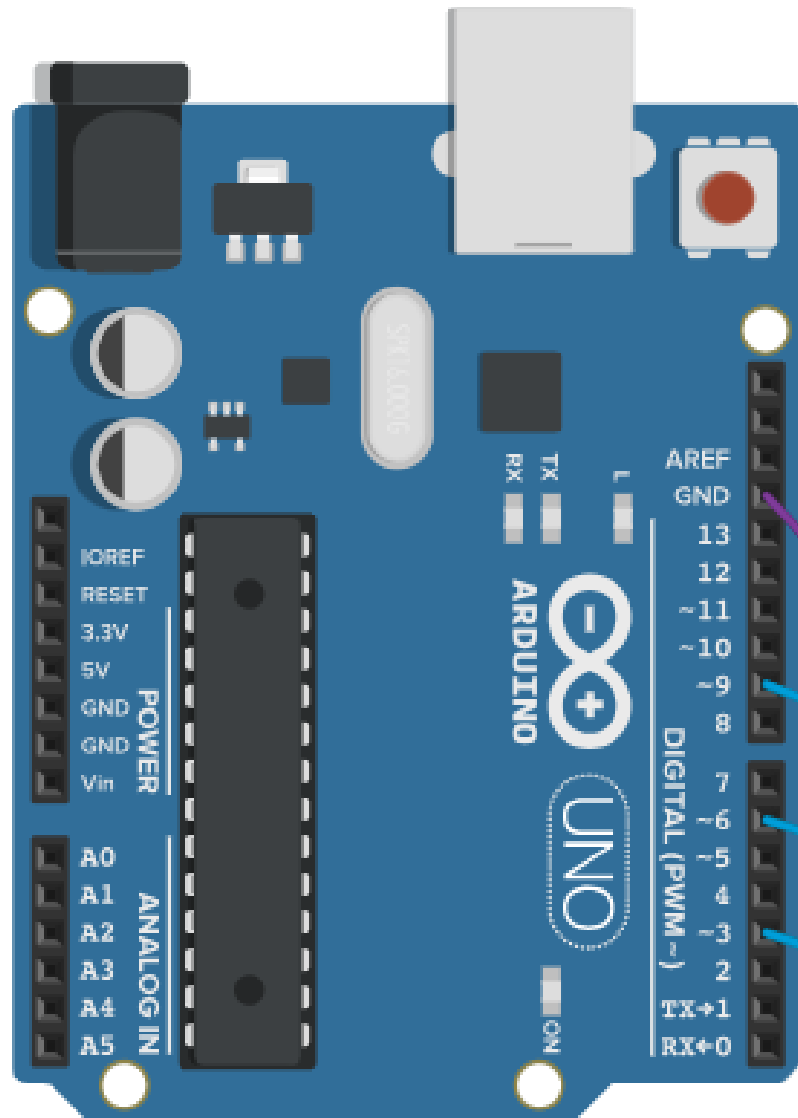
시리얼 모니터에서 숫자를 입력하면 LED 밝기를 PWM으로 조절하는 프로그램

```
1  const int LED=9;
2
3  byte data = 0; //시리얼 입력값
4  String input=""; //최종 문자열
5  int adapt = -1; //최종 문자열 형변환
6
7  void setup()
8  {
9      Serial.begin(9600);
10     pinMode(LED, OUTPUT);
11 }
12
13 void loop()
14 {
15     if(Serial.available() > 0) { //수신된 데이터 존재 여부 확인
16
17         data = Serial.read(); //시리얼 입력값, 한 글자씩 data에 저장
18
19         if(data == 10) { //엔터입력시 실행
20
21             adapt = input.toInt(); //입력받은 숫자 형변환
22
23             if(0 <= adapt && adapt <= 255) { //입력받은 숫자가 아날로그 값 범위 내일 경우 실행
24
25                 Serial.println("Brightness:" + input); //input값 확인
26
27                 while(Serial.available() <= 0) { //수신된 데이터 존재 여부 없으면 계속실행
28                     analogWrite(LED, input.toInt());
29                     delay(10);
30                 }
31
32             } else Serial.println("wrong input");
33
34             input=""; //초기화
35         }
36         else input += (char)data;
37     }
38 }
39 }
```

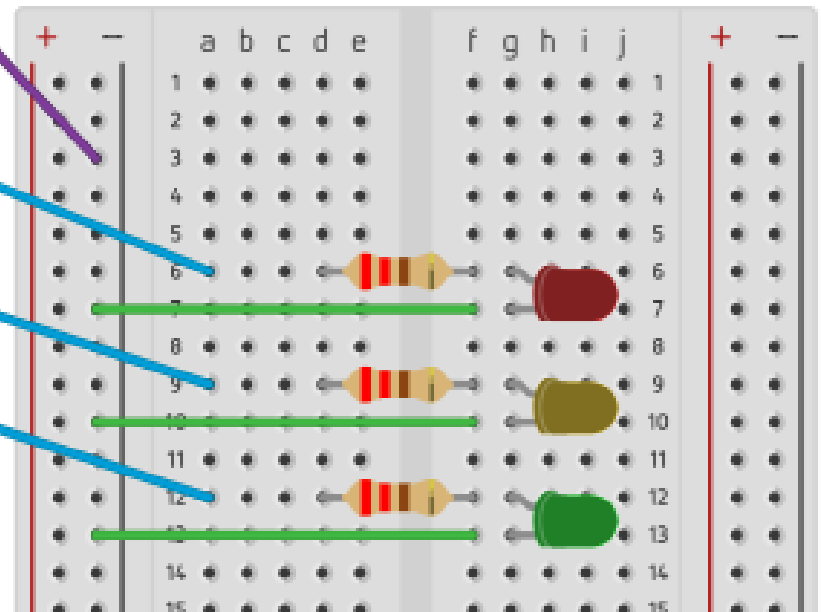


## 02 LED 다중 연결

# LED 다중 연결



음극은 모아서 처리!



## uno\_ex3-6 : 신호등

```
1  int pin[3] = {9, 6, 3};
2  int i=0;
3
4  void setup() {
5
6      for(int i=0; i<sizeof(pin); i++) pinMode(pin[i], OUTPUT);
7  }
8
```

```
9  void loop() {
10
11  switch(i){
12  case 0: //빨간불일 경우
13      digitalWrite(pin[i], HIGH);
14      delay(3500);
15      digitalWrite(pin[i], LOW);
16      break;
17
18  case 1: //노랑불일 경우
19      digitalWrite(pin[i], HIGH);
20      delay(1000);
21      digitalWrite(pin[i], LOW);
22      break;
23
24  case 2: //파란불일 경우
25      digitalWrite(pin[i], HIGH);
26      delay(2500);
27      digitalWrite(pin[i], LOW);
28      break;
29  }
30
31  //pin 번호 조정
32  if(i<2) i++;
33  else i=0;
34  }
```

## uno\_ex3-7: 사용자입력으로 LED 디지털 출력

```
1  int pin[3] = {9, 6, 3};
2  int input=0;
3  byte data=0;
4
5  void setup() {
6
7      Serial.begin(9600);
8      for(int i=0; i<sizeof(pin); i++) pinMode(pin[i], OUTPUT);
9  }
10
```

```
11 void loop() {
12
13     if(Serial.available()>0){
14
15         data = Serial.read();//사용자 입력
16
17         if(data == 10){
18             switch(input){
19                 case 0:
20                 case 1:
21                 case 2:
22                     Serial.println((String)input+" ON...");
23
24                     digitalWrite(pin[input], 1);
25                     delay(1000);//1초만 켜지고 off
26                     digitalWrite(pin[input], 0);
27
28                     break;
29
30                 default: Serial.println("wrong input");
31             }
32         }else input = data-48;//아스키코드->숫자변환
33     }
34
35 }
```

## uno\_ex3-8: 사용자입력으로 LED 아날로그 출력

```
1  int pin[3] = {9, 6, 3};
2
3  String data=""; //사용자 입력
4  int no=0; //핀 번호 인덱스 0, 1, 2
5  int adapt=0; //밝기 0~255
6
7  void setup() {
8
9      Serial.begin(9600); //시리얼 통신을 위한 주파수
10
11     for(int i=0; i<3; i++) pinMode(pin[i], OUTPUT);
12
13 }
```

## uno\_ex3-8: 사용자입력으로 LED 아날로그 출력

```
15 void loop() {  
16  
17   if(Serial.available()>0){  
18  
19     data = Serial.readString(); //사용자 입력  
20  
21     no = data.charAt(0)-48; 첫 번째 문자 숫자로 변환  
22     adapt = data.substring(2, data.length()-1).toInt(); //띄어쓰기 이후 문자 숫자로 변환  
23     6 in case  
24     Serial.println("no:"+(String)no+ " / adapt:"+(String)adapt);  
25  
26     while(Serial.available()<=0){ //다음 입력을 받을 때까지 계속 출력  
27       analogWrite(pin[no], adapt);  
28       delay(10);  
29       analogWrite(pin[no], 0);  
30       delay(10);  
31     }  
32   }  
33 }  
34 }
```

출력 시리얼 모니터 x

2 215

20:55:06.091 -> no:2 / adapt:215

char charAt(index)  
String substring(from, to)

index	0	1	2	3	4	5
value	2		2	1	5	Wn

charAt(0)

substring(2,5)

## 03 RGB 모듈 연결

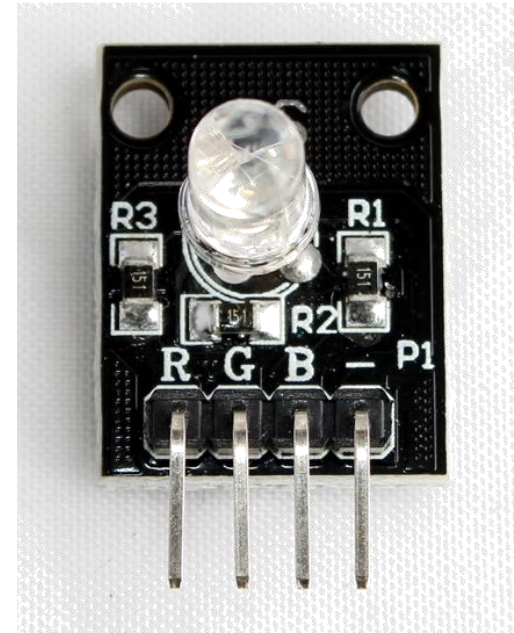
# RGB LED

## ■ RGB LED : 내부에 빨간색(R), 초록색(G), 파란색(B)의 광원이 내장되어 컬러를 표현할 수 있는 모듈

- 3개 색상의 LED가 4개의 다리를 가지도록 구성
- 각 색상별 양극(+)핀 3개와 음극(-)핀 1개

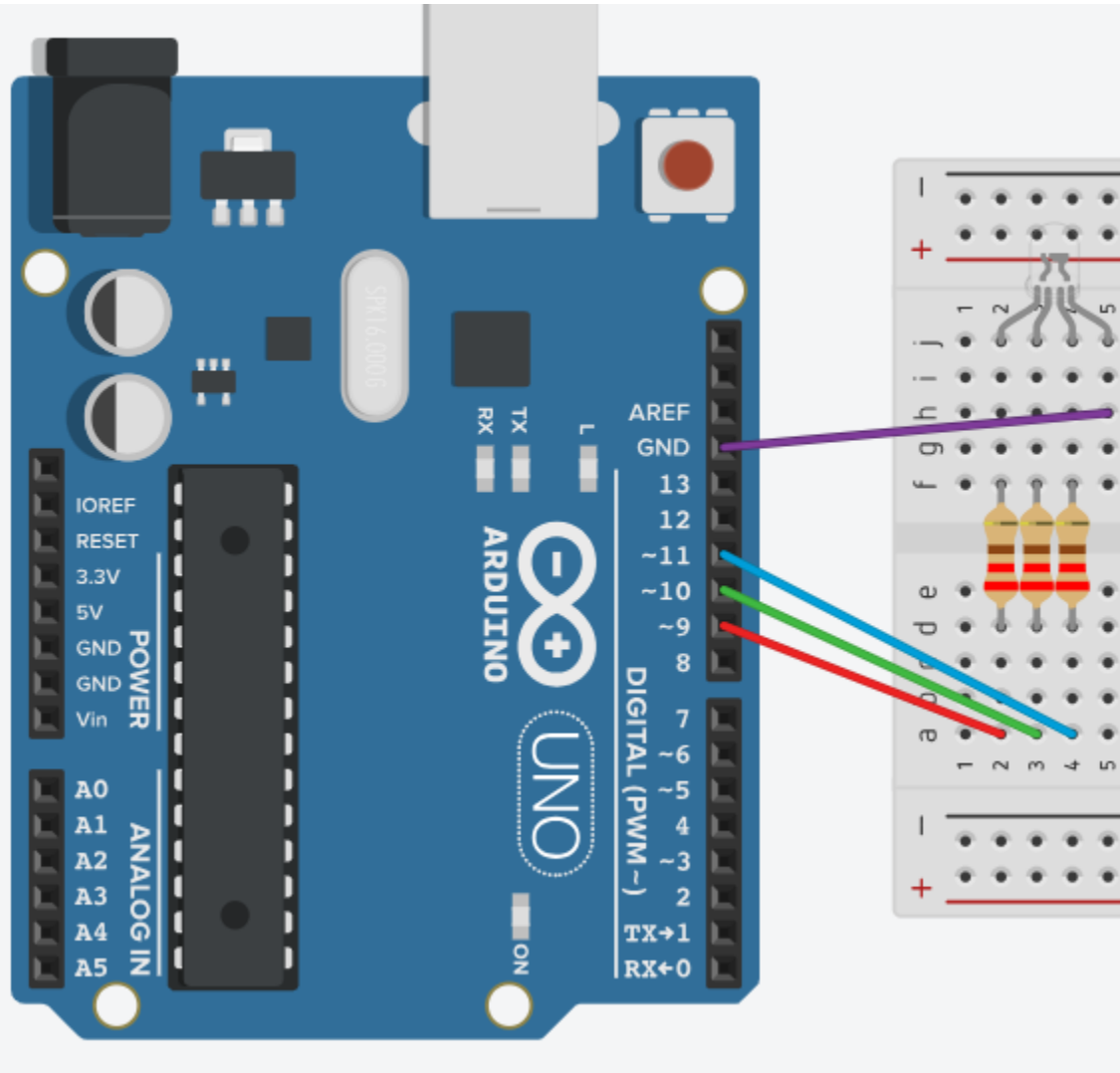
## ■ RGB LED 제어

- 디지털 혹은 PWM 신호로 각 LED 색상 밝기 제어
  - 각 LED의 밝기 정도를 조합하여 다양한 색 표현 가능





# RGB LED 조명등 회로



RGB LED의 각 LED에  
전류 제한을 위한 저항  
연결이 필요함

R: 9번 핀  
G: 10번 핀  
B: 11번 핀

## uno\_ex3-09:RGB LED 조명등을 위한 디지털 출력

```
1  const int RLED=9;
2  const int GLED=10;
3  const int BLED=11;
4
5  int i=1;
6
7  void setup()
8  {
9      pinMode(RLED, OUTPUT);
10     pinMode(GLED, OUTPUT);
11     pinMode(BLED, OUTPUT);
12 }
13
14
15 void allOff(){
16     digitalWrite(RLED, LOW);
17     digitalWrite(GLED, LOW);
18     digitalWrite(BLED, LOW);
19 }
20
21
22 void setMode(int mode)
23 {
24     switch(mode){
25     case 1:
26
27         digitalWrite(RLED, LOW);
28         digitalWrite(GLED, HIGH);
29         digitalWrite(BLED, HIGH);
30         delay(500);
31
32         allOff();
33         delay(500);
34         break;
35     case 2:
36         digitalWrite(RLED, HIGH);
37         digitalWrite(GLED, LOW);
38         digitalWrite(BLED, HIGH);
39         delay(500);
40
41         allOff();
42         delay(500);
43         break;
44     case 3:
45         digitalWrite(RLED, HIGH);
46         digitalWrite(GLED, HIGH);
47         digitalWrite(BLED, LOW);
48         delay(500);
49
50         allOff();
51         delay(500);
52         break;
53     }
54 }
55
56 void loop() {
57
58     setMode(i);
59
60     if(i<3) i++;
61     else i=1;
62 }
```

## uno\_ex3-10:RGB LED 조명등을 위한 아날로그 출력

```
1  const int RGB_PIN[]={9,10,11};
2  const int SIZE=3;
3  int randValue[SIZE]={0,0,0};
4
5  void setup()
6  {
7      Serial.begin(9600); //시리얼 통신을 위한 주파수
8
9      //핀 번호 설정
10     for(int i=0; i<SIZE; i++) pinMode(RGB_PIN[i], OUTPUT);
11
12     //랜덤시드값 설정
13     randomSeed(analogRead(0));
14 }
15
16 void alloff(){
17     for(int i=0; i<SIZE; i++) analogWrite(RGB_PIN[i], LOW);
18 }
19
20 void loop() {
21
22     for(int i=0; i<SIZE; i++) {
23         randValue[i] = (int)random(256);
24         analogWrite(RGB_PIN[i], randValue[i]);
25     }
26
27     Serial.println("R:"+(String)randValue[0]+"/G:"+(String)randValue[1]+"/B:"+(String)randValue[2]);
28
29     delay(500);
30
31     alloff();
32     delay(500);
33
34 }
```

analogRead(0): 아날로그 0번 핀의 신호를 받아서 난수 발생 시작값으로 사용  
연결되지 않은 analog포트를 읽으면 0~1023중 하나의 값 반환

# 난수 생성 함수

## `void randomSeed(unsigned int seed)`

- 매개변수
  - seed : 의사 난수의 시작 위치 결정을 위한 값
- 반환값 : 없음

## `long random(long max)` `long random(long min, long max)`

- 매개변수
  - min : 생성될 난수의 최솟값
  - max : max - 1이 생성될 난수의 최댓값
- 반환값 : [min, max - 1] 범위의 난수값

# 요약

- LED : 출력 신호 모듈
- LED 모듈을 아두이노와 연결시 반드시 **저항**필요!
- 디지털 출력 vs 아날로그 출력
  - 디지털 출력 : 0 or 1
  - 아날로그 출력: 0~255
  - 핀 번호 고려하여 연결
- LED 모듈 다중 연결 시 음극은 GND에 한 번에 처리
- RGB모듈 연결: 핀 번호와 R,G,B LED 연결 선 확인

**Thank You**