

# LED, RGB LED, 가변저항 제어

RGB LED 부품 개요

가변 저항 부품 개요

회로구성

LED 여러 개

LED Analog 출력

LED 출력 가변저항 입력

RGB LED

LED 제어 프로그래밍

LED 여러 개

LED Analog 출력

LED 출력 가변저항 입력

RGB LED

# LED 여러 개

```
int LED1 = 2
```

```
int LED2 = 4
```

```
int LED3 = 6;
```

```
void setup(){
```

```
    pinMode(LED1, OUTPUT);
```

```
    pinMode(LED2, OUTPUT);
```

```
    pinMode(LED3, OUTPUT);
```

```
}
```

```
void loop(){
```

```
    digitalWrite(LED1, HIGH);
```

```
    digitalWrite(LED2, HIGH);
```

```
    digitalWrite(LED3, HIGH);
```

```
    delay(1000);
```

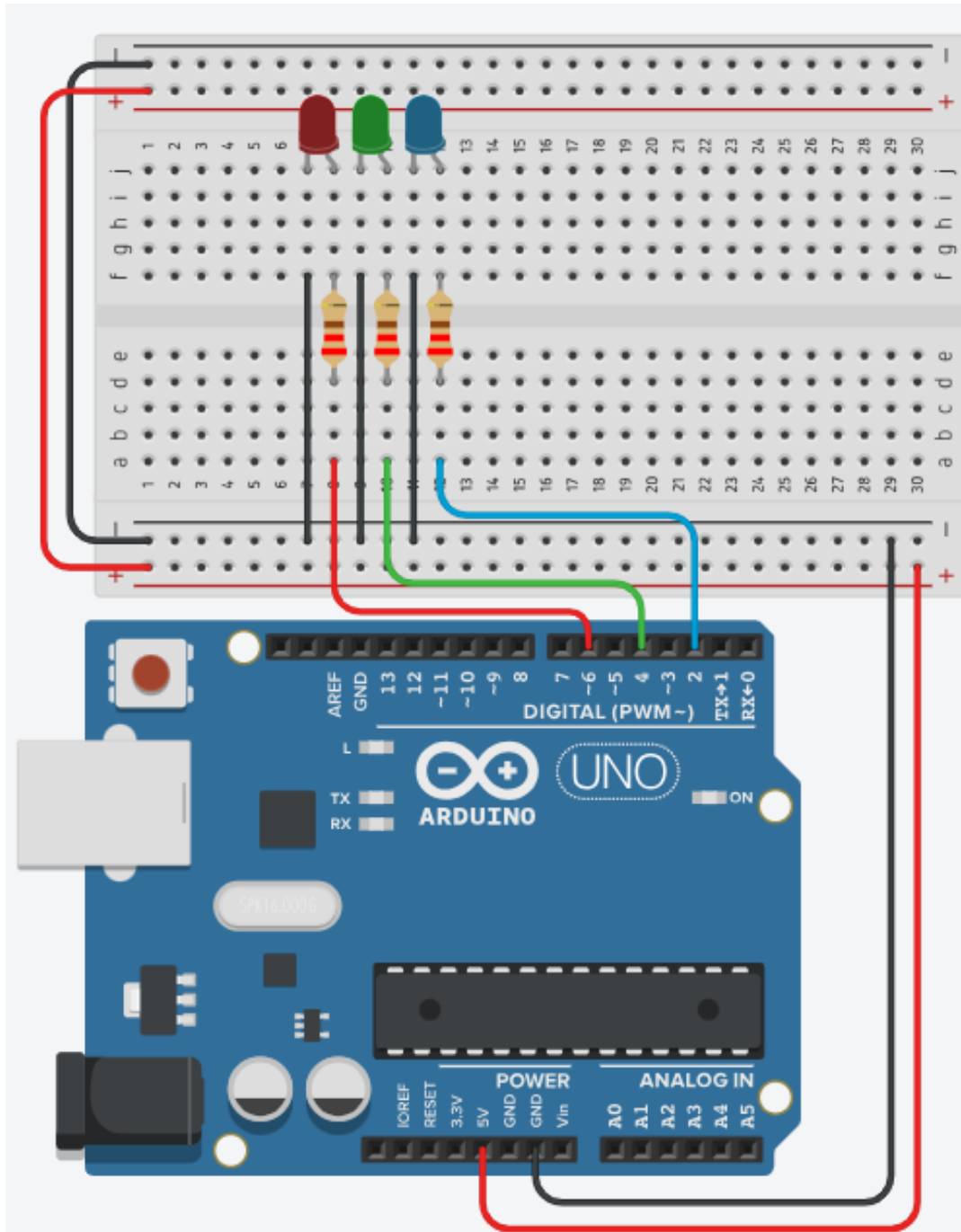
```
    digitalWrite(LED1, LOW);
```

```
    digitalWrite(LED2, LOW);
```

```
    digitalWrite(LED3, LOW);
```

```
    delay(1000);
```

```
}
```

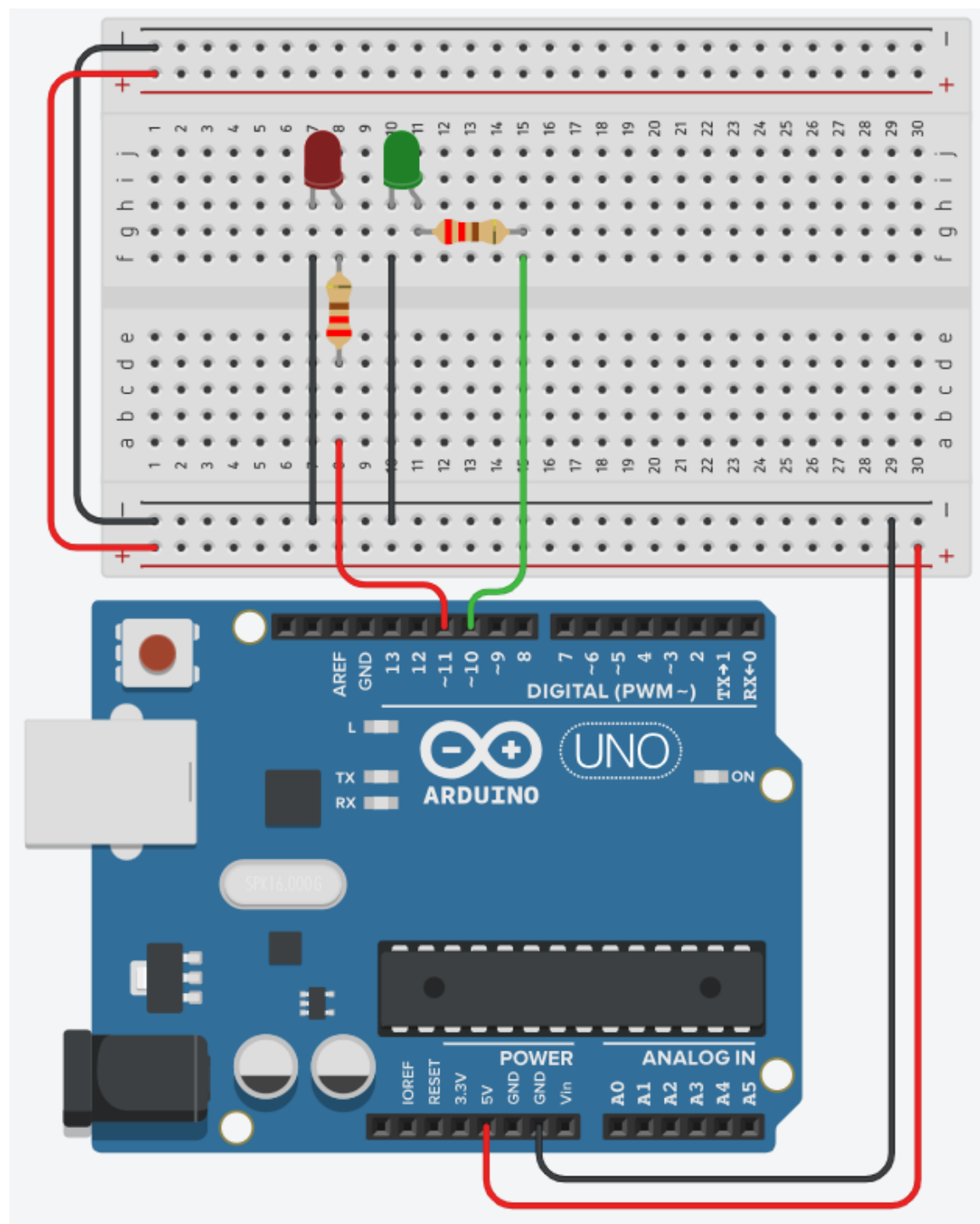


# LED 아날로그 출력

```
#define LED_Red      11
#define LED_Green   10

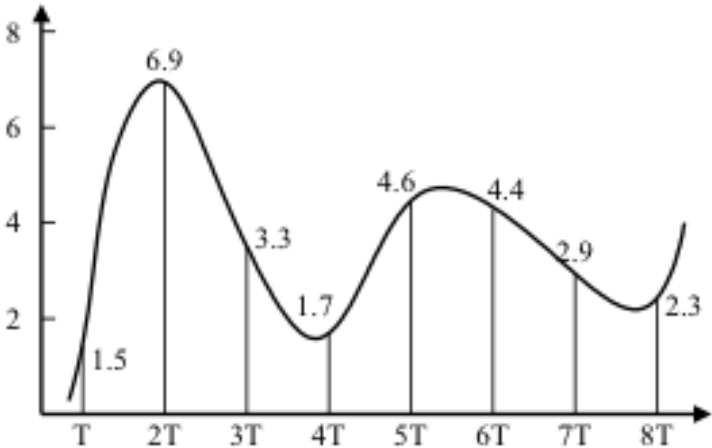
void setup(){
  Serial.begin(9600);
  pinMode(LED_Red, OUTPUT);
  pinMode(LED_Green, OUTPUT);
}

void loop(){
  int valBright;
  for(valBright=0; valBright<=255; valBright++){
    analogWrite(LED_Red, valBright);
    analogWrite(LED_Green, 255-valBright);
    Serial.print("Red : ");
    Serial.print(valBright);
    Serial.print(", Green : ");
    Serial.println(255-valBright);
    delay(10);
  }
}
```

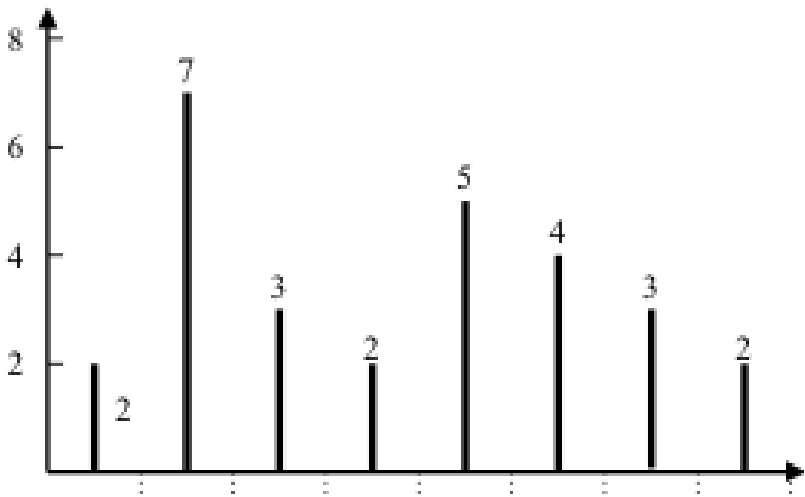


# ADC(Analog-Digital Convertor)

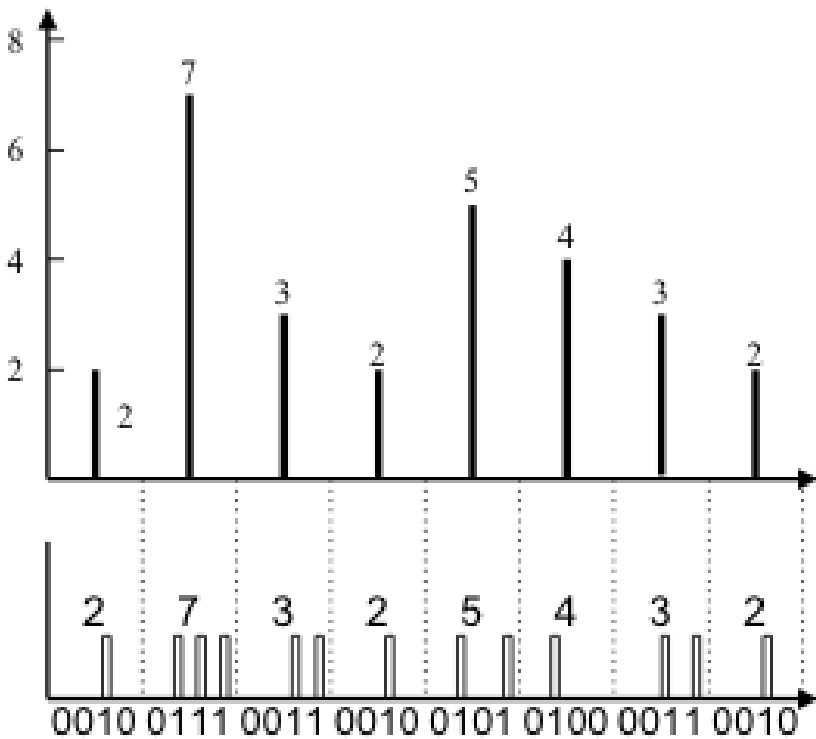
표본화 : 연속적인 아날로그 파형을 이산적인 디지털 데이터로 변환하려고 표본을 얻는 과정



양자화 : 표본화된 실수값을 표현 가능한 이진수로 표현하기 위하여 어느 정도의 정밀도로 표현할 것인지 결정하는 과정



부호화 : 샘플링되거나 양자화된 데이터를 2진 데이터로 변환하는 과정



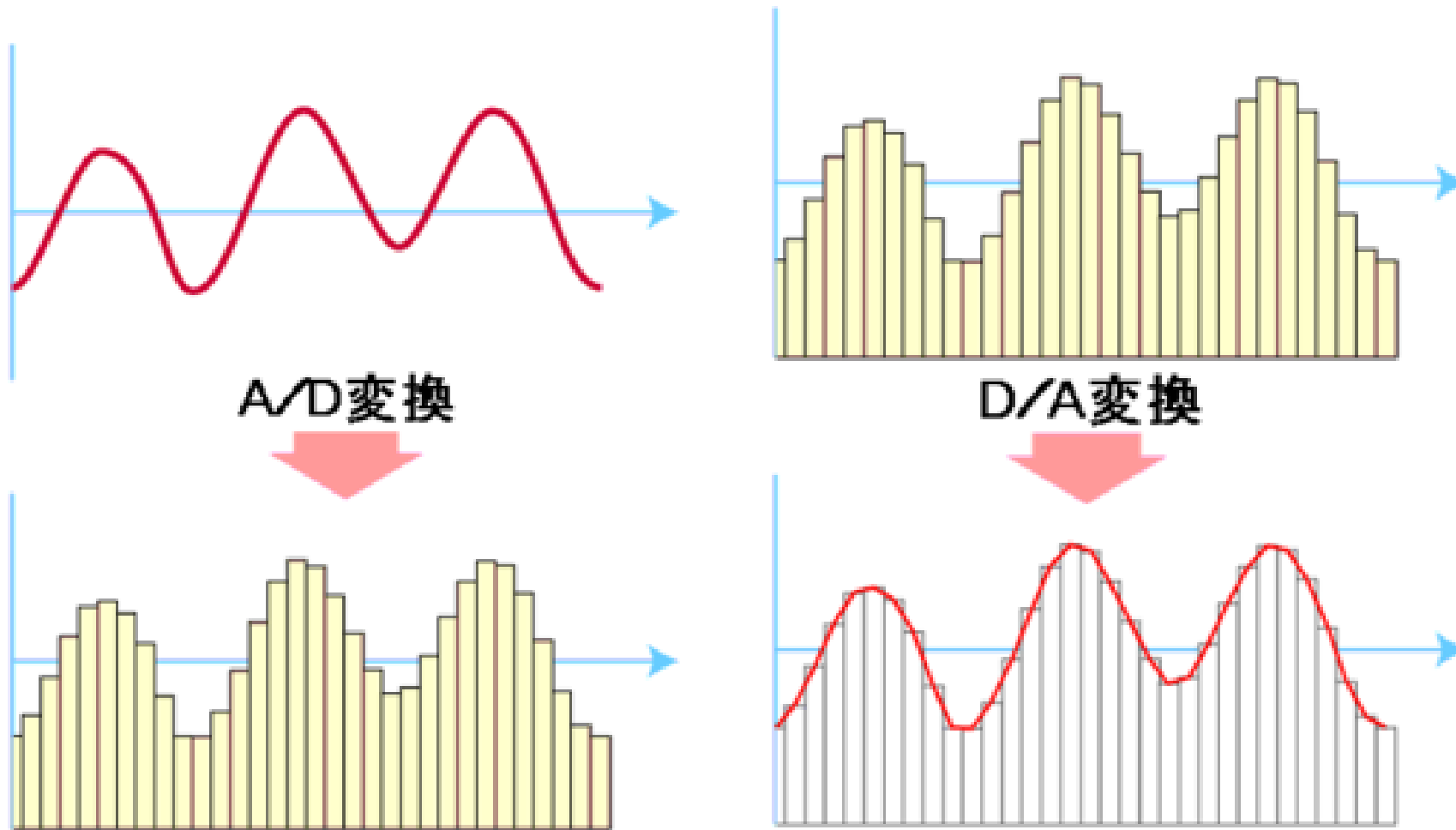
# ADC(Analog-Digital Convertor)

PWM : 펄스의 폭으로 디지털 동작을 아날로그 동작처럼 보이도록 변환하는 과정

ADC : 아날로그 값을 손실을 감수하고 디지털로 변환하는 과정

DAC : 디지털 값을 왜곡을 감수하고 아날로그로 변환하는 과정

해상도 : ADC와 DAC는 비트에 따라서 단계와 범위가 다름(8 bits, 9 bits, 10 bits 등)



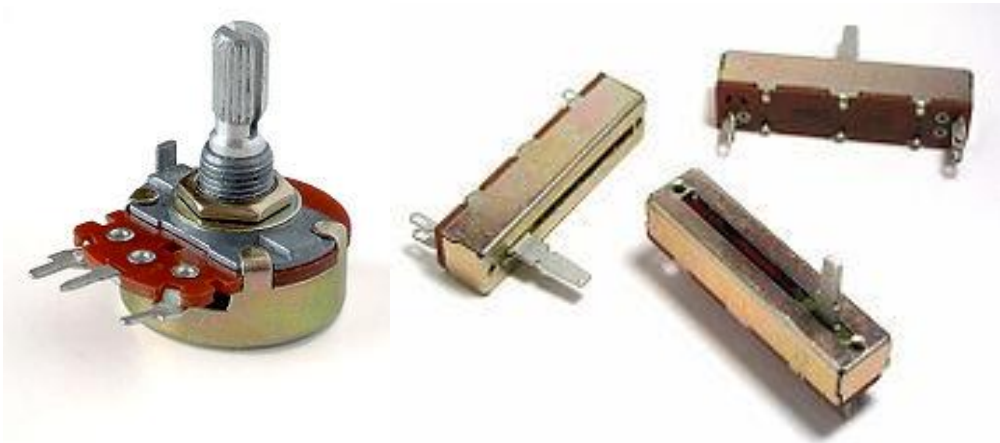
# 가변저항(Potentionmeter)



→  
리얼 기호 중간고사

가변저항 : 전자회로에서 저항값을 정해진 범위 안에서 바꿀 수 있는 저항

용도 : 볼륨, 온도, 밝기, 주파수 등을 조절

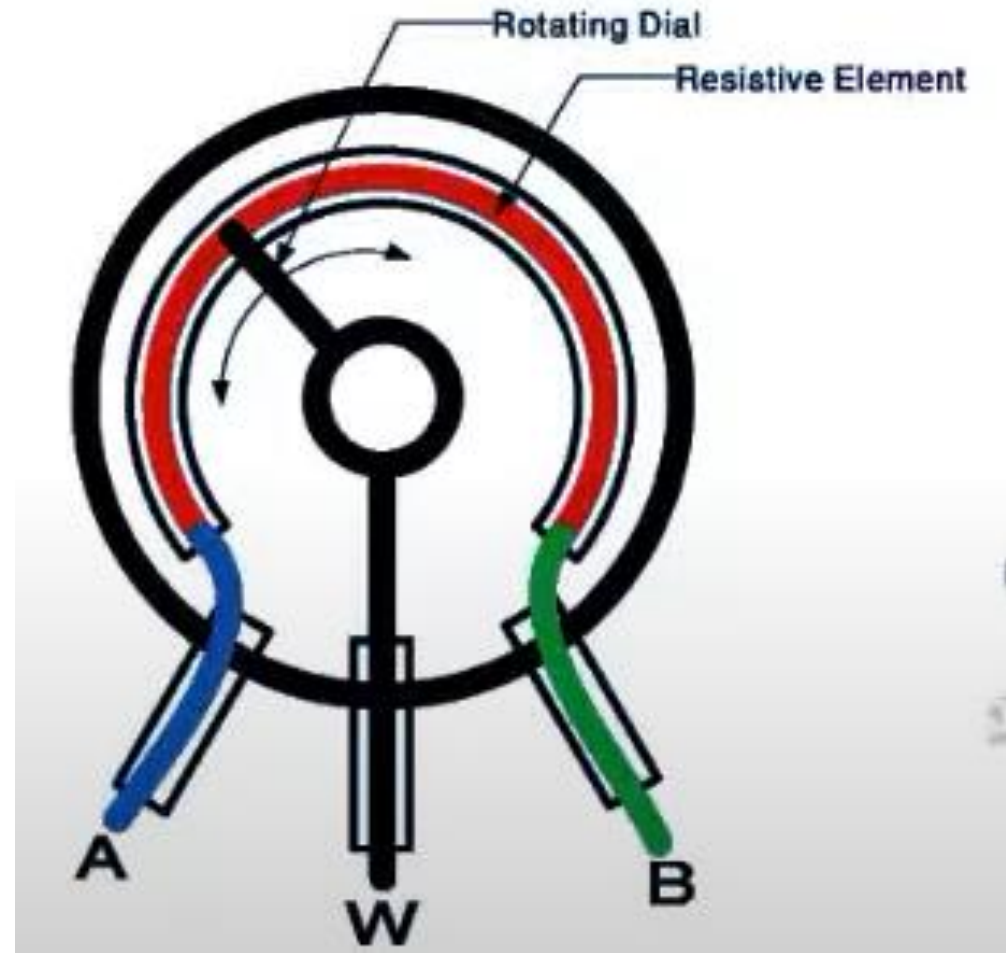
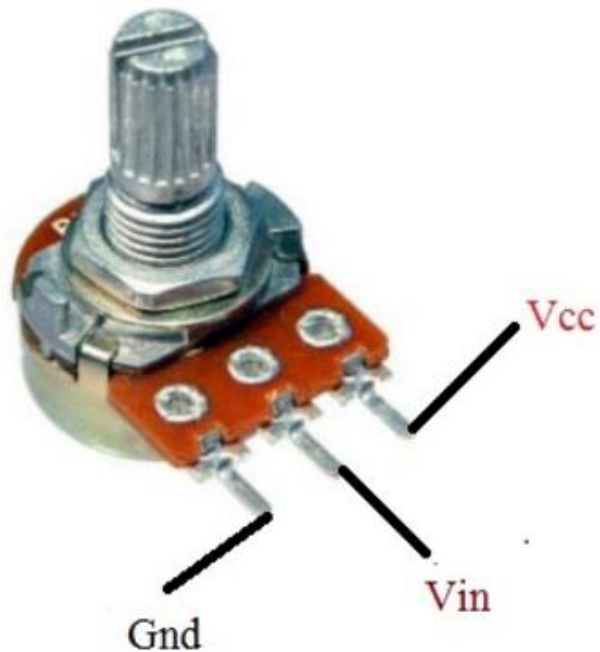




# 가변저항(Potentionmeter)

## 다이얼타입 가변저항 🗨️

- Vin : 저항값 전달
- VCC와 GND는 서로 바꿔서 끼울 수 있는 대신, 최소-최대 방향이 달라짐
- Wiper(Rotating Dial) : 막대를 돌리는 도구
- Resistive Material(Resistive Element, 저항 물질) : 저항성을 가진 물질, 이 길이가 조절되면서 값이 조절

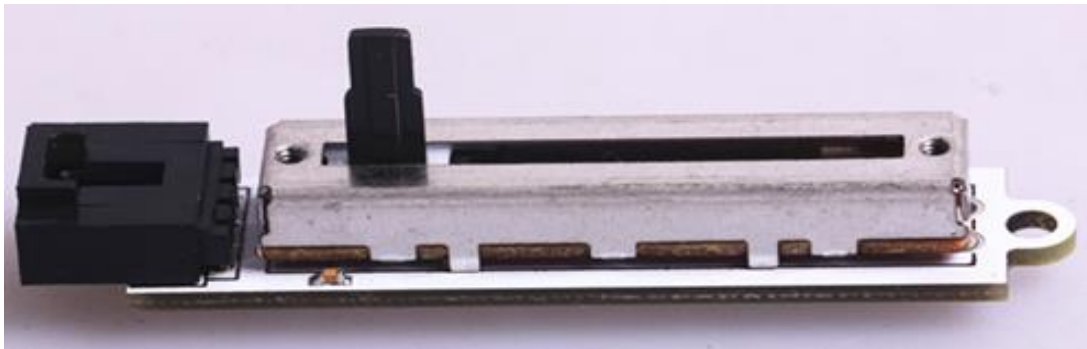


# 가변저항(Potentionmeter)

## 슬라이딩타입 가변저항



- OUT : 저항값 전달
- VCC와 GND는 서로 바뀌서 끼울 수 없음
- Wiper(Rotating Dial) : 막대를 돌리는 도구
- Resistive Material(Resistive Element, 저항 물질) : 저항성을 가진 물질, 이 길이가 조절되면서 값이 조절
- LED는 옵션

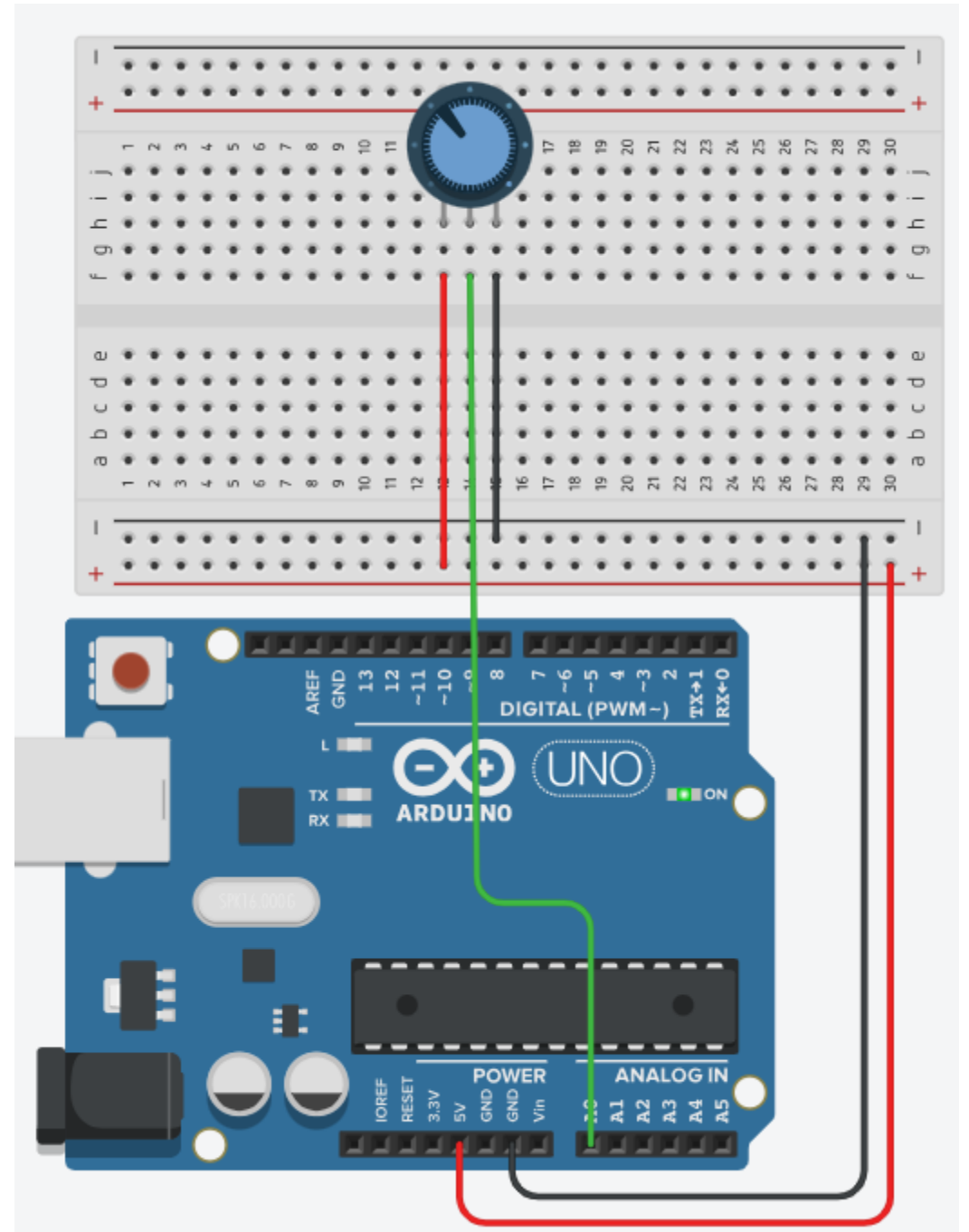


Sliding Type Potentiometer	Arduino UNO
GND	GND
VCC	+5V
OUT	A4
LED	A5



# 가변저항 값을 시리얼 모니터에 출력

```
int pin_Poten = A0;  
void setup() {  
    Serial.begin(9600);  
    pinMode(pin_Poten, INPUT);  
}  
  
void loop() {  
    int readValue = analogRead(pin_Poten);  
    Serial.println(readValue);  
}
```

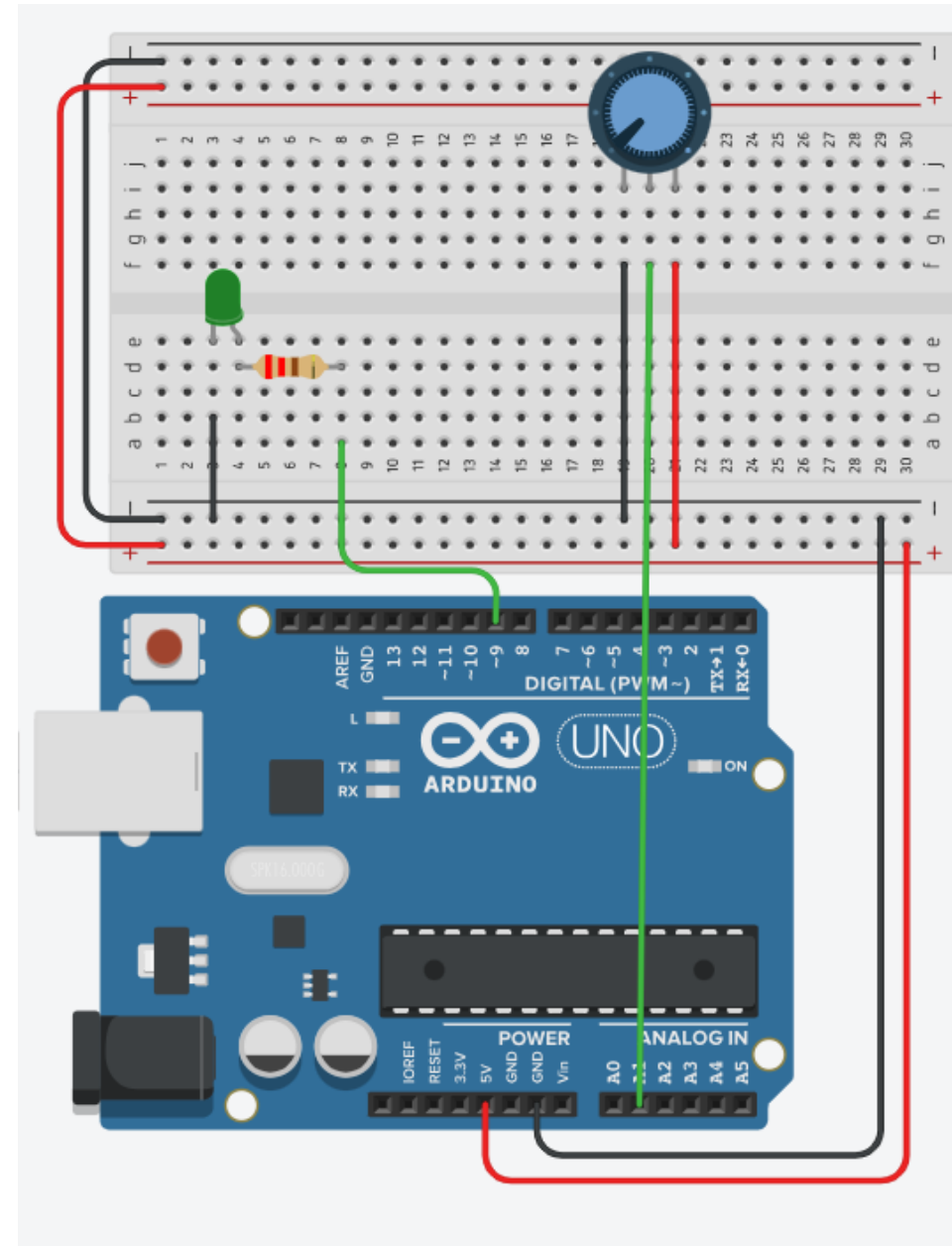


# 가변저항 값과 LED 밝기 : 시리얼 모니터에 출력

```
int LED_Green = 9;  
int Pot_Green = A1;  
int val_Sensor_Green = 0;  
int val_Output_Green = 0;
```

```
void setup() {  
  pinMode(LED_Green, OUTPUT);  
  pinMode(Pot_Green, INPUT);  
  Serial.begin(9600);  
}
```

```
void loop() {  
  val_Sensor_Green = analogRead(Pot_Green);  
  val_Output_Green = map(val_Sensor_Green, 0, 1023, 0, 255);  
  Serial.print("Potentionmeter Value = ");  
  Serial.print(val_Sensor_Green);  
  Serial.print(", LED's Bright = ");  
  Serial.println(val_Output_Green);  
  analogWrite(LED_Green, val_Output_Green);  
  delay(10);  
}
```



# RGB LED

## RGB Color와 CMYK Color

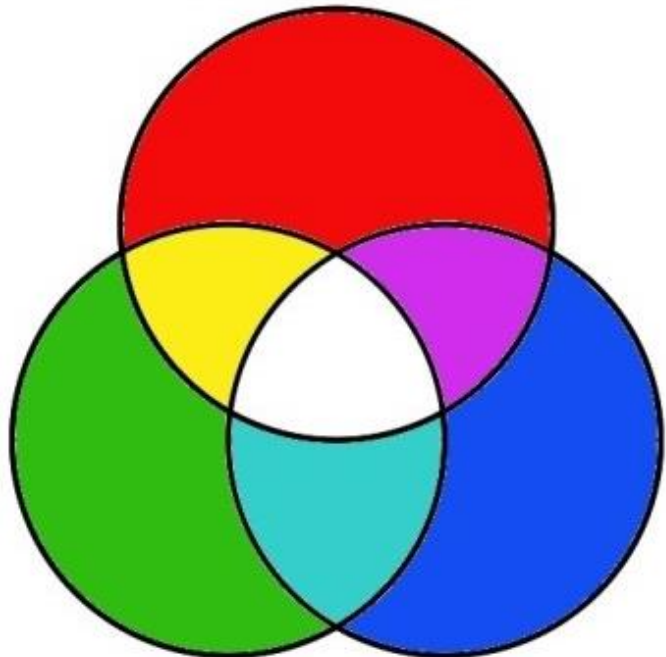
- RGB Color : Red, Green, Blue
- CMYK Color : Cyan, Magenta, Yellow, Key(black)

## 가산혼합과 감산혼합

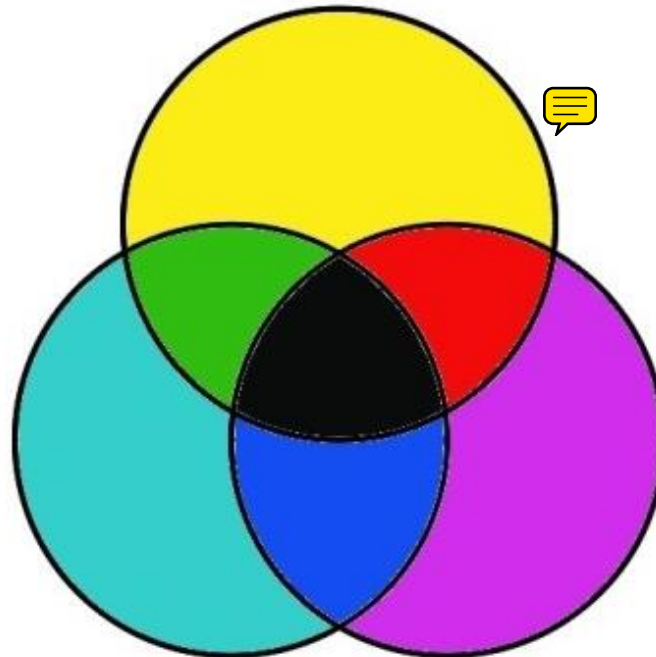
가산혼합 : 섞을수록 명도가 높아진다.

감산혼합 : 섞을수록 명도가 낮아진다.

ADDITIVE



SUBTRACTIVE



```
index.html - Windows 메모장
파일(F) 편집(E) 서식(O) 보기(V) 도움말(H)

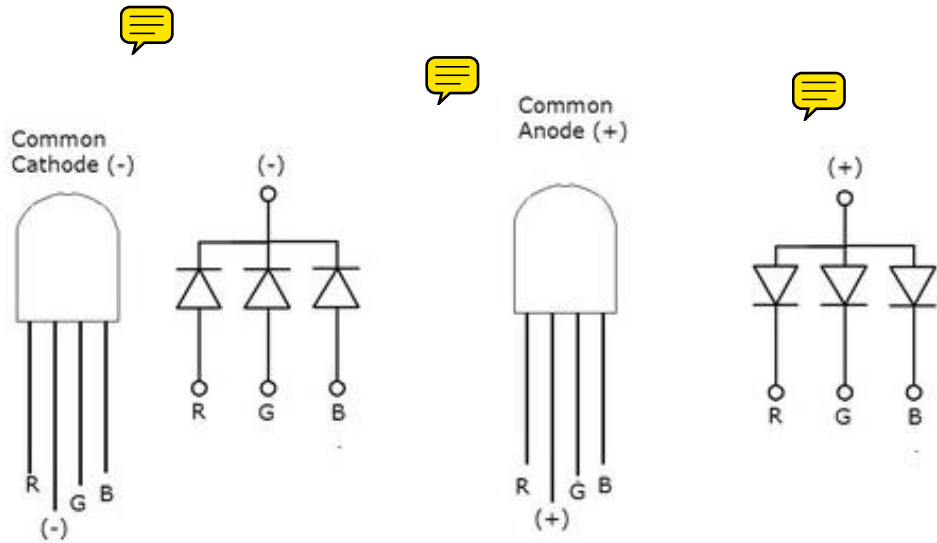
<html>
    <head>
        <title>RGB Color</title>
    </head>
    <body bgcolor = "#00FF00">
        <!-- <body bgcolor = "#FFFF00"> -->
    </body>
</html>
```

Ln 6, Col 36    100%    Windows (CRLF)    UTF-8

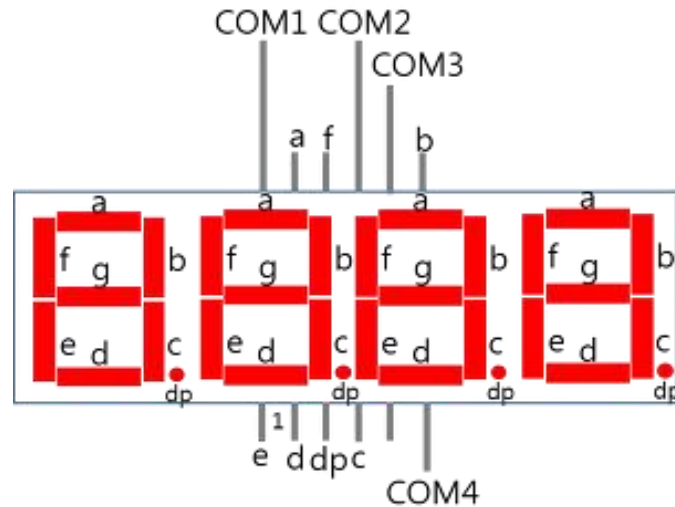
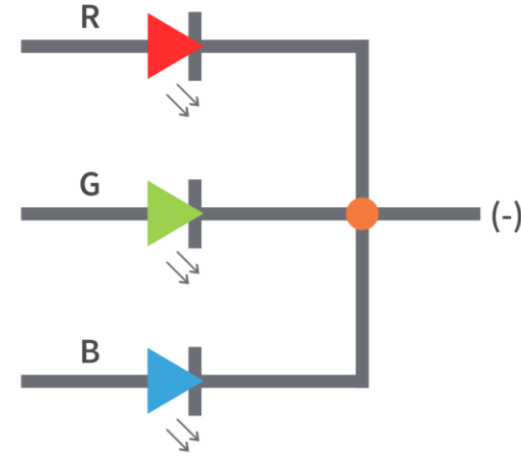
# RGB LED

## Common Cathode와 Common Anode

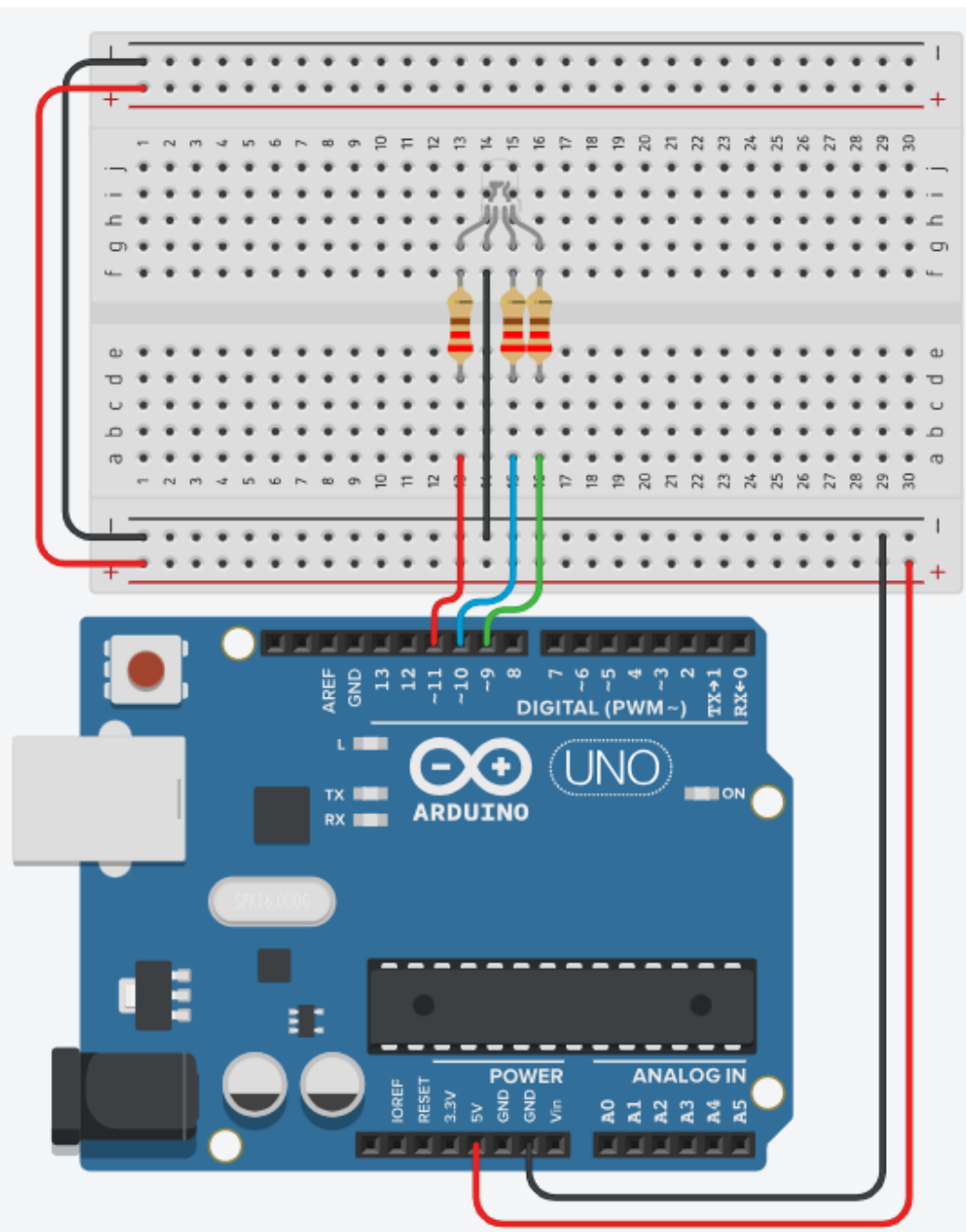
- Cathode
- Anode



## Common Cathode와 Common Anode



# RGB LED



```
int LED_Red_Pin = 11;  
int LED_Green_Pin = 9;  
int LED_Blue_Pin = 10;
```

```
void setup()
```

```
{
```

```
    pinMode(LED_Red_Pin, OUTPUT);  
    pinMode(LED_Green_Pin, OUTPUT);  
    pinMode(LED_Blue_Pin, OUTPUT);
```

```
}
```

```
void loop()
```

```
{
```

```
    RGB_Color(255, 0, 0);  
    delay(1000);  
    RGB_Color(0, 0, 0);  
    delay(1000);  
    RGB_Color(0, 255, 0);  
    delay(1000);  
    RGB_Color(0, 0, 0);  
    delay(1000);  
    RGB_Color(255, 255, 0);  
    delay(1000);  
    RGB_Color(0, 0, 0);  
    delay(1000);
```

```
}
```

```
void RGB_Color(int LED_Red_Val, int LED_Green_Val, int LED_Blue_Val)
```

```
{
```

```
    analogWrite(LED_Red_Pin, LED_Red_Val);  
    analogWrite(LED_Green_Pin, LED_Green_Val);  
    analogWrite(LED_Blue_Pin, LED_Blue_Val);
```

```
}
```