

---

# LED 제어하기

# LED 제어하기

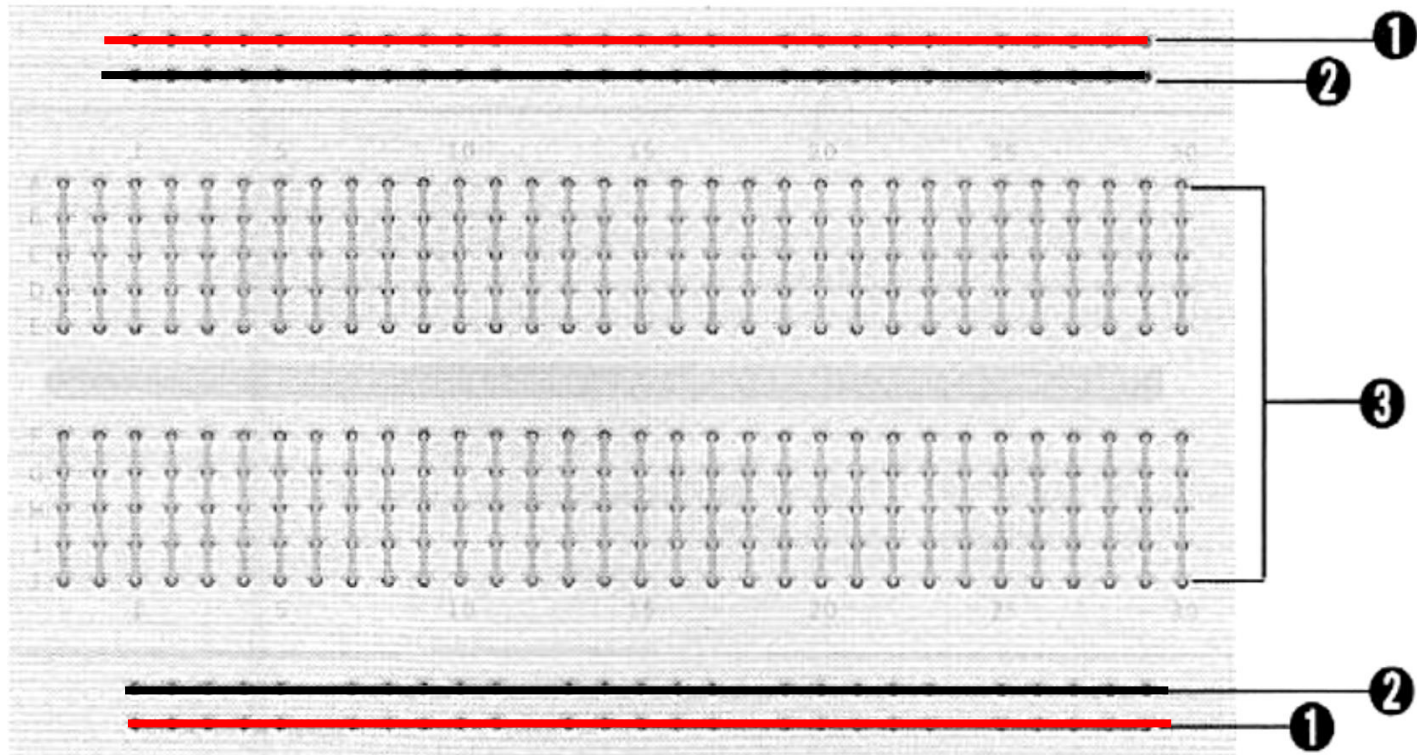
## ❖ 브레드 보드(Bread board, 빵판)

○ 남뚝이 필요없는 프로토타입용 보드

① VCC (빨간색)

② GND (검은색)

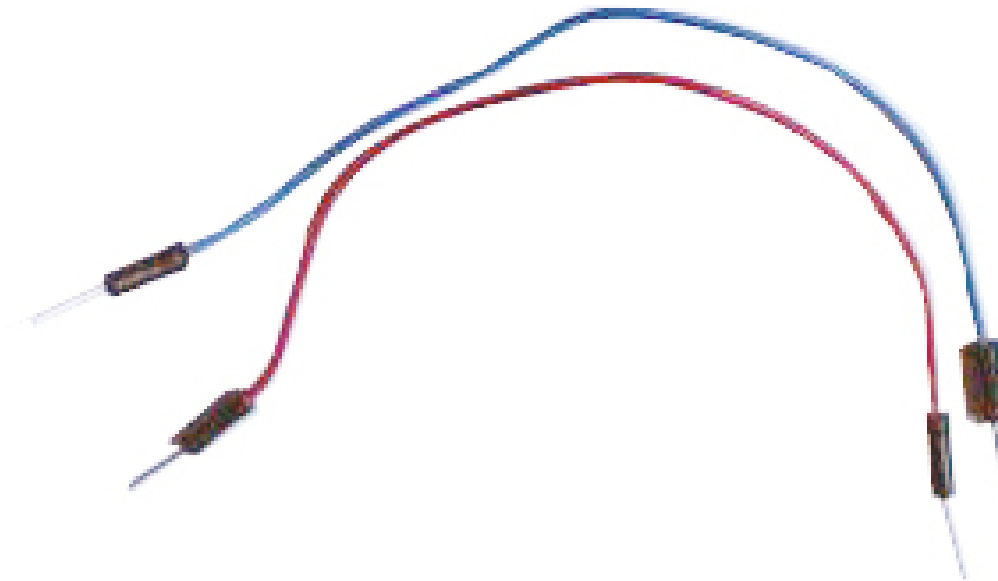
③ 부품 연결선



---

## ❖ 점퍼선

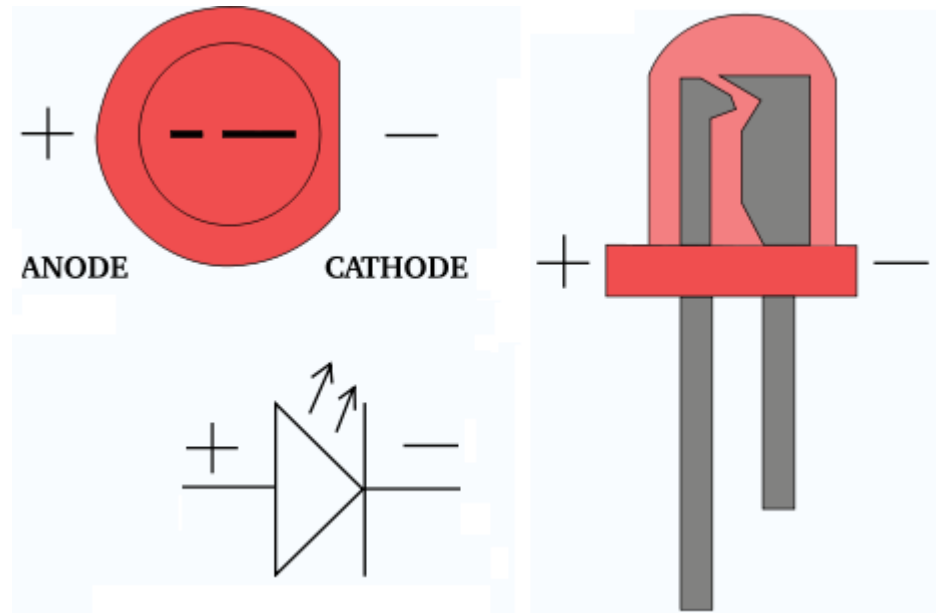
- 브레드 보드에 부품들을 서로 연결할 때 이용



# LED 제어하기

## ❖ LED 발광 다이오드

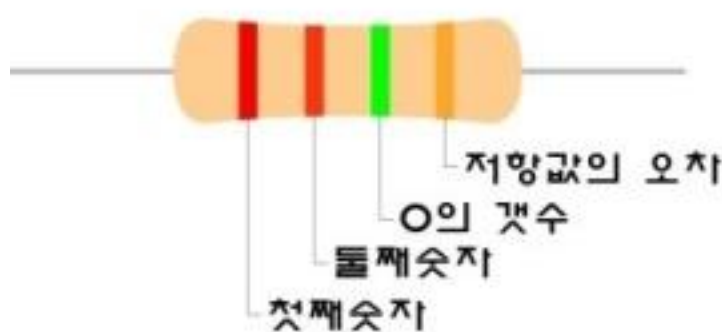

- 전류가 흐르면서 빛을 발하는 반도체
- 다이오드 : 전류를 한쪽 방향으로만 흐르게 함
- 과다 전류가 흐를 경우 파손
  - LED 앞에 저항을 배치



# LED 제어하기

## ❖ 저항

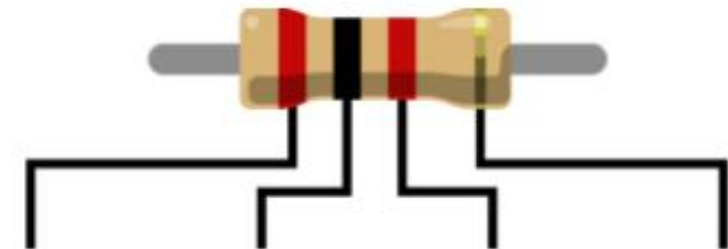
- 전기의 흐름을 제한

	색	값
	검정색	0
	갈색	1
	빨강색	2
	주황색	3
	노란색	4
	초록색	5
	파란색	6
	보라색	7
	회색	8
	하얀색	9
	은색	$\pm 10\%$
 <p data-bbox="518 1242 994 1356">4, 7, 4 (0이 4개란 의미)  <math>470000\Omega = 470k\Omega</math></p>	금색	$\pm 5\%$

# LED 제어하기

## ❖ 저항

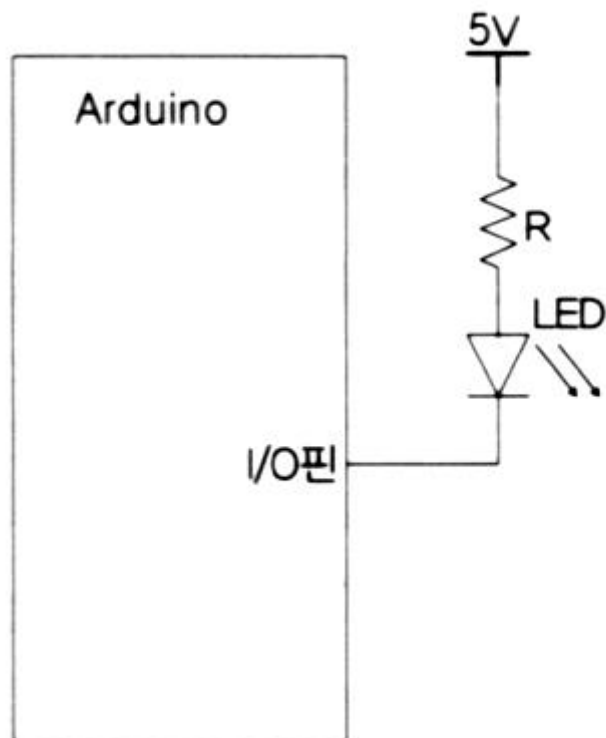
- 전기의 흐름을 제한



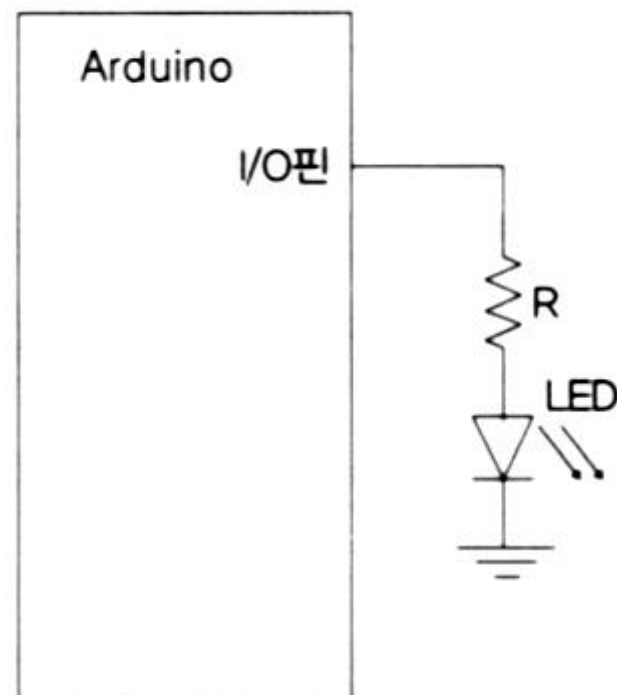
1st digit	2nd digit	Multiplier	Tolerance
0	0	x 1	
1	1	x 10	±1%
2	2	x 100	±2%
3	3	x 1K	
4	4	x 10K	
5	5	x 100K	
6	6	x 1M	
7	7		
8	8	x 0.1	±5%
9	9	x 0.01	±10%

# LED 제어하기

## ❖ LED 연결 방식



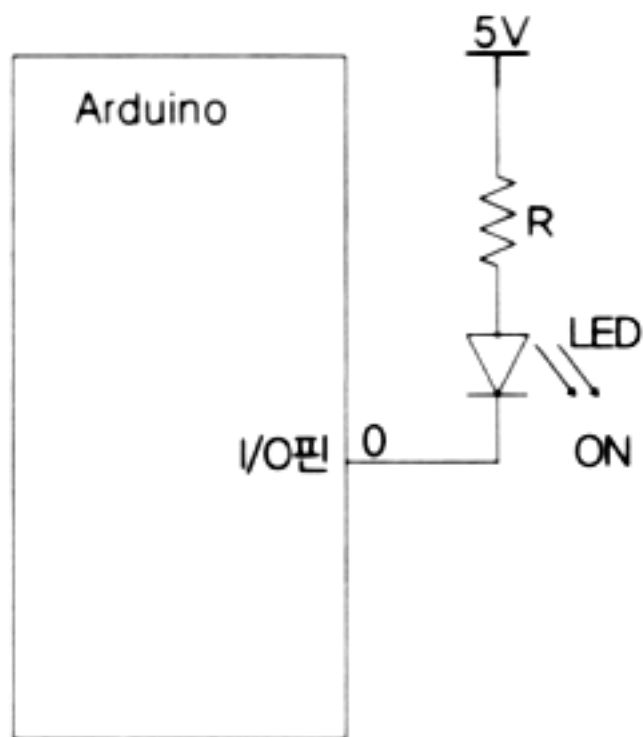
(a) 풀업 방식



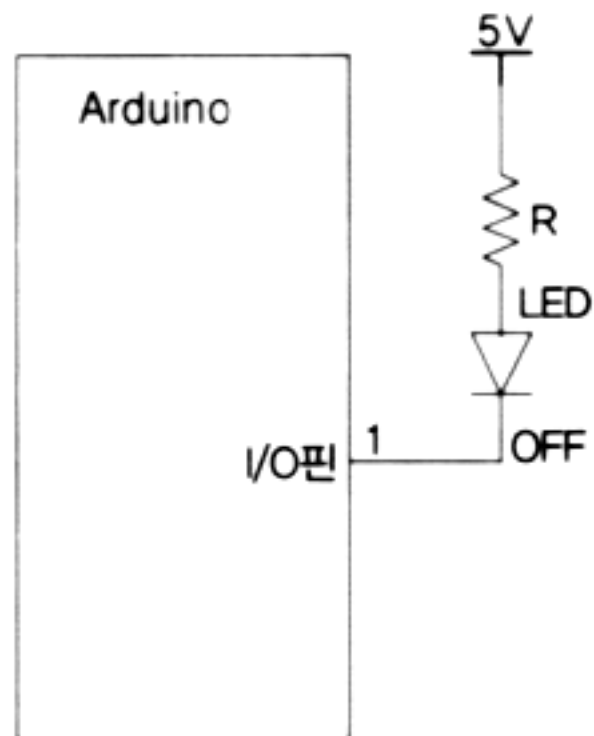
(b) 풀다운 방식

# LED 제어하기

## ❖ 풀업 방식 LED 제어



(a) 0 출력

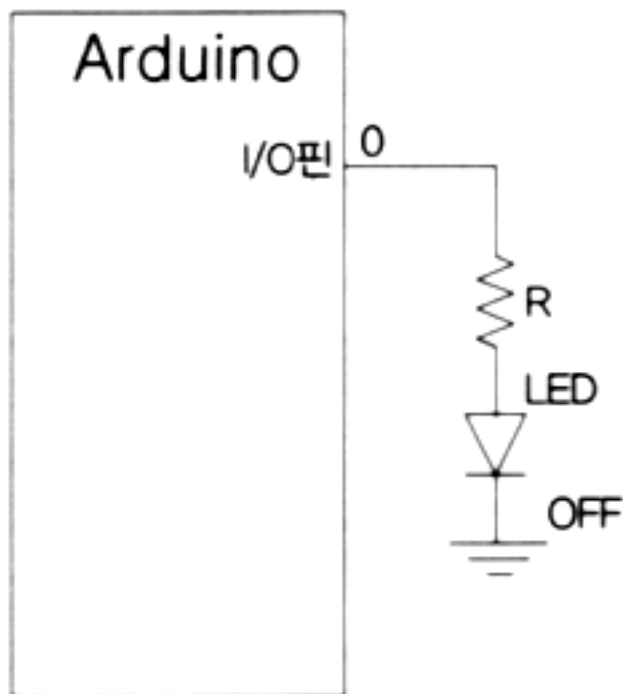


(b) 1 출력

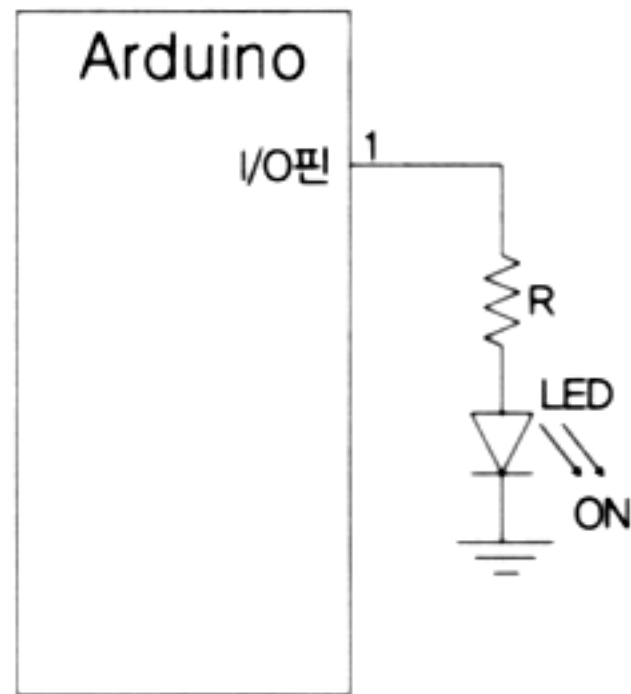


# LED 제어하기

## ❖ 풀다운 방식 LED 제어



(a) 0 출력



(b) 1출력

# LED 제어하기

---

## ❖ LED에 연결 저항값 계산

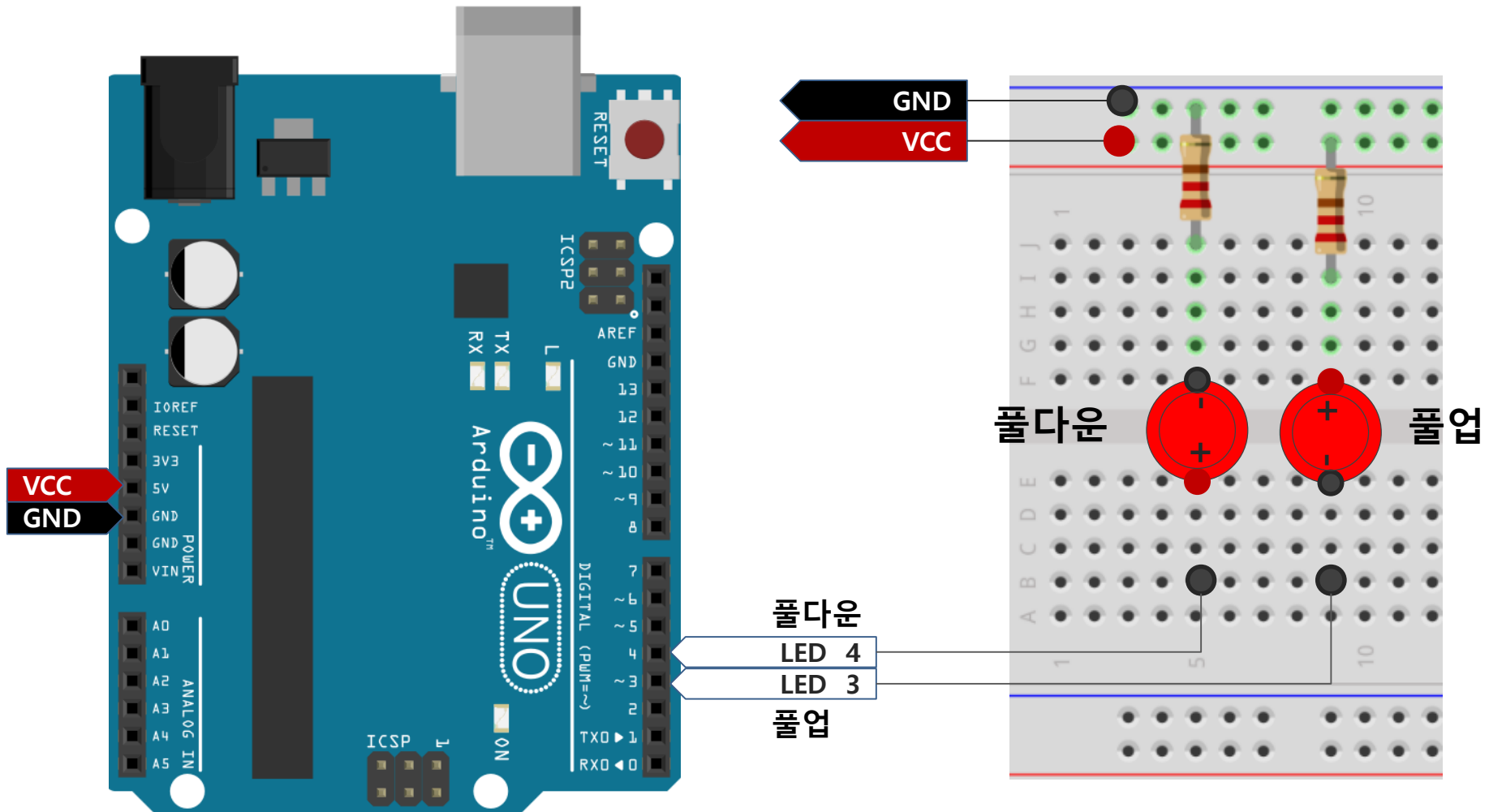
○  $V = IR$

$$5V = Ri + V_{LED} \rightarrow i = \frac{5V - V_{LED}}{R}$$

$$\frac{5 - 2.0}{R} = 10mA \rightarrow R = 300\Omega$$

# 실습1: LED 풀업/풀다운 연결핀 HIGH 출력 확인

## ❖ 회로 구성



# 실습1: LED 풀업/풀다운 연결핀 HIGH 출력 확인

## ❖ led1.ino

```
// LED 풀업/풀다운 연결핀 HIGH 출력 확인
const int pu_led_pin = 3;      // 풀업 연결핀
const int pd_led_pin = 4;      // 풀다운 연결핀

void setup() {
    pinMode(pu_led_pin, OUTPUT);    // 풀업 LED 연결핀 출력 설정
    pinMode(pd_led_pin, OUTPUT);    // 풀다운 LED 연결핀 출력 설정
}

void loop() {
    digitalWrite(pd_led_pin, HIGH); // 풀다운 LED 연결핀 HIGH 출력
    digitalWrite(pu_led_pin, HIGH); // 풀업 LED 연결핀 HIGH 출력

    // digitalWrite(pd_led_pin, LOW); // 풀다운 LED 연결핀 LOW 출력
    // digitalWrite(pu_led_pin, LOW); // 풀업 LED 연결핀 LOW 출력
}
```

## 실습2: LED 풀업/풀다운 연결핀 LOW 출력 확인

### ❖ led2.ino

```
const int PULLUP_LED = 3;
const int PULLEDOWN_LED = 4;

int state = 0;

void setup() {
    Serial.begin(115200);
    pinMode(PULLEDOWN_LED, OUTPUT);
    pinMode(PULLUP_LED, OUTPUT);
}

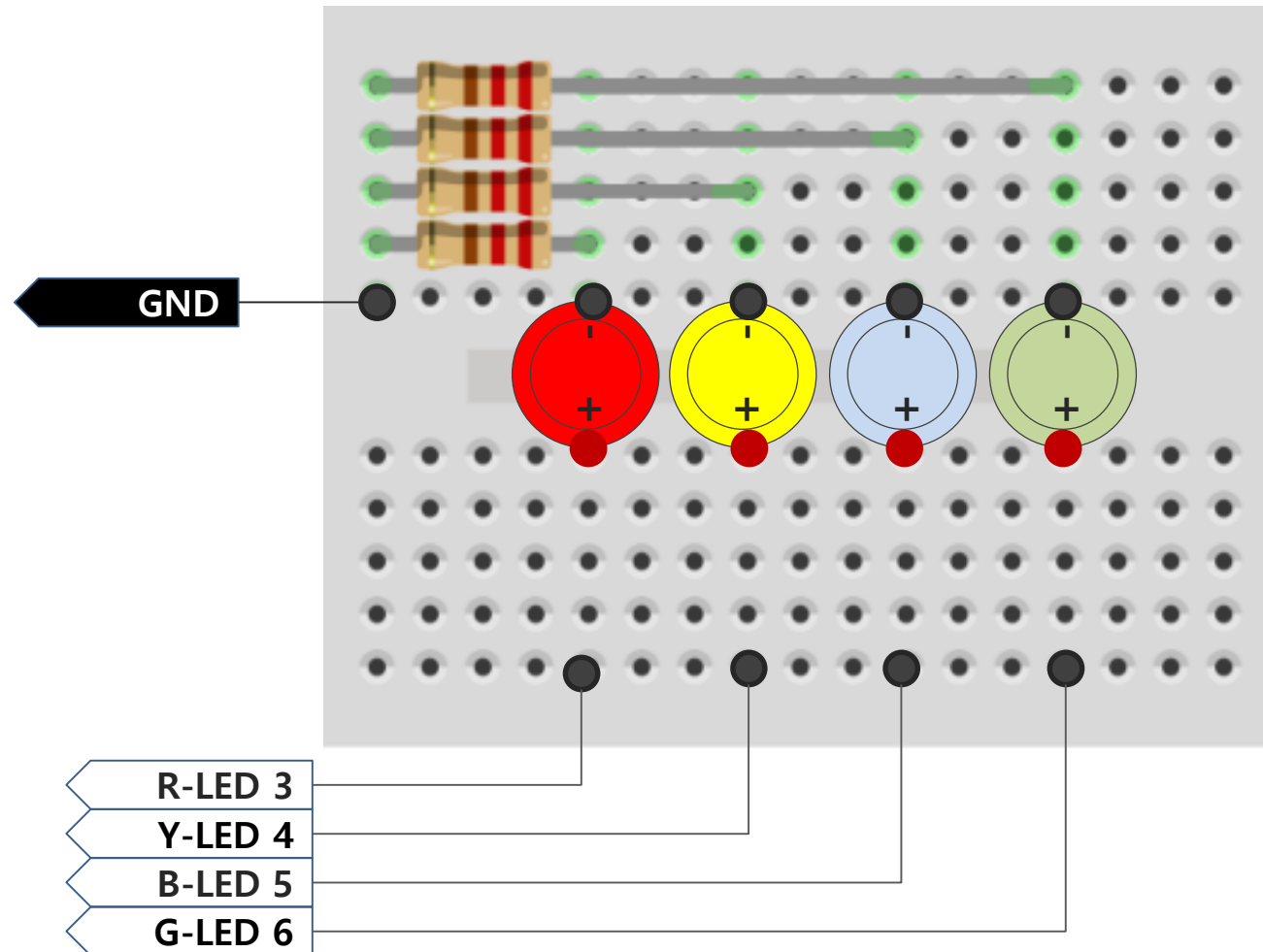
void loop() {
    Serial.print("state value : ");
    Serial.println(state);

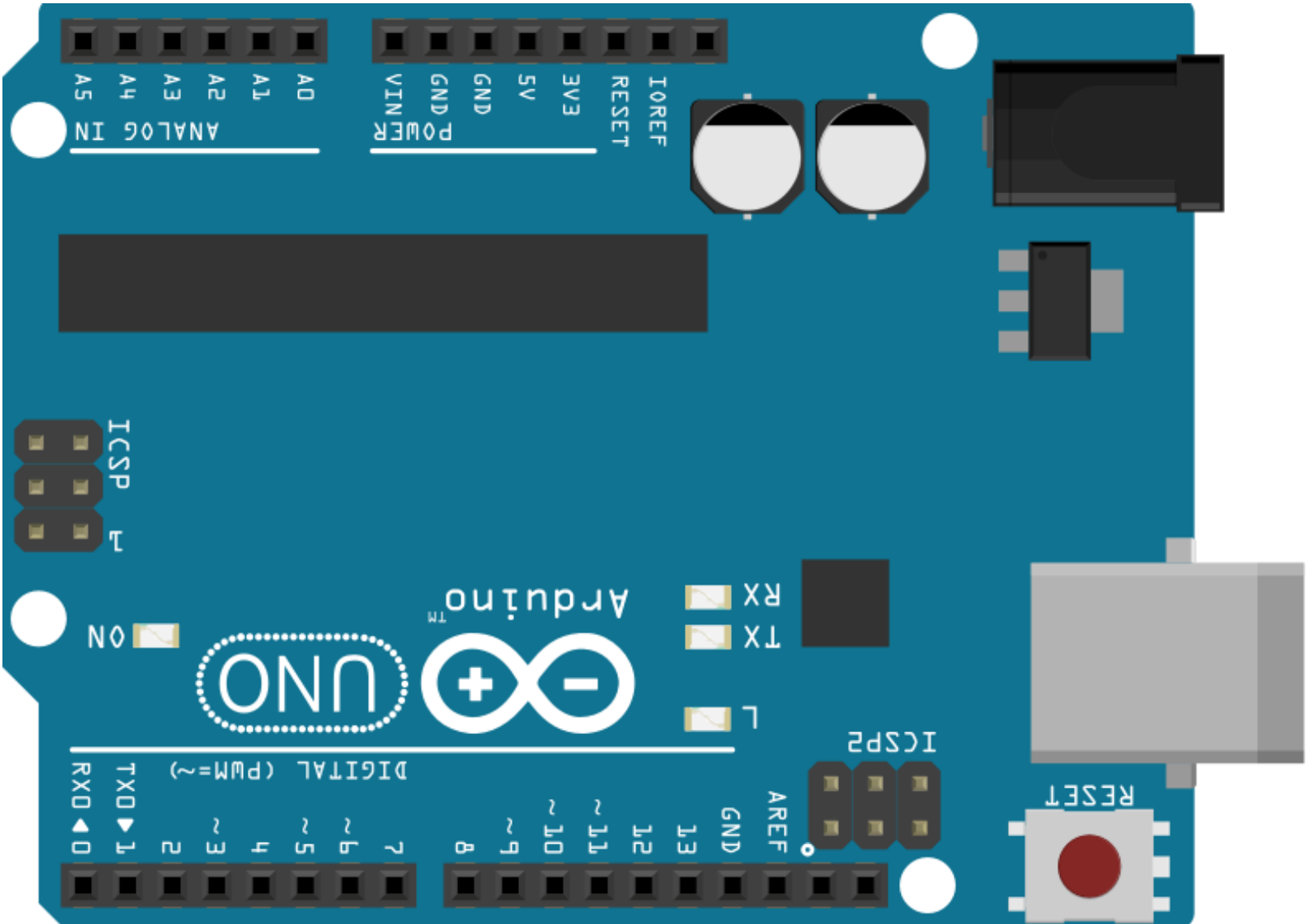
    digitalWrite(PULLEDOWN_LED, state);
    digitalWrite(PULLUP_LED, state);

    state = !state;
    delay(1000);
}
```

## 실습3: LED 4개 제어하기

### ❖ 회로 구성





## 실습3: 4개의 순차점멸(LED 풀다운 연결)

### ❖ ex03/app.ino

```
// 4개의 순차점멸(LED 풀다운)
const int led1_pin = 3;
const int led2_pin = 4;
const int led3_pin = 5;
const int led4_pin = 6;

void setup()
{
    pinMode(led1_pin, OUTPUT);    // 3번 핀 출력 설정
    pinMode(led2_pin, OUTPUT);    // 4번 핀 출력 설정
    pinMode(led3_pin, OUTPUT);    // 5번 핀 출력 설정
    pinMode(led4_pin, OUTPUT);    // 6번 핀 출력 설정
}

void loop()
{
    // LED1만 ON
    digitalWrite(led1_pin, HIGH); // 3번 핀 HIGH 출력(on)
    digitalWrite(led2_pin, LOW);  // 4번 핀 LOW 출력(off)
    digitalWrite(led3_pin, LOW);  // 5번 핀 LOW 출력(off)
    digitalWrite(led4_pin, LOW);  // 6번 핀 LOW 출력(off)
    delay(1000);
}
```



## 실습3: 4개의 순차점멸(LED 풀다운 연결)

### ❖ ex03/app.ino

```
// LED2만 ON
digitalWrite(led1_pin, LOW); // 3번 핀 LOW 출력(off)
digitalWrite(led2_pin, HIGH); // 4번 핀 HIGH 출력(on)
digitalWrite(led3_pin, LOW); // 5번 핀 LOW 출력(off)
digitalWrite(led4_pin, LOW); // 6번 핀 LOW 출력(off)
delay(1000);

// LED3만 ON
digitalWrite(led1_pin, LOW); // 3번 핀 LOW 출력(off)
digitalWrite(led2_pin, LOW); // 4번 핀 LOW 출력(off)
digitalWrite(led3_pin, HIGH); // 5번 핀 HIGH 출력(on)
digitalWrite(led4_pin, LOW); // 6번 핀 LOW 출력(off)
delay(1000);

// LED4만 ON
digitalWrite(led1_pin, LOW); // 3번 핀 LOW 출력(off)
digitalWrite(led2_pin, LOW); // 4번 핀 LOW 출력(off)
digitalWrite(led3_pin, LOW); // 5번 핀 LOW 출력(off)
digitalWrite(led4_pin, HIGH); // 6번 핀 HIGH 출력(on)
delay(1000);
}
```

## 실습4: 4개의 순차점멸(LED 풀다운 연결)

### ❖ ex04/app.ino

```
const int led1_pin = 3;
const int led2_pin = 4;
const int led3_pin = 5;
const int led4_pin = 6;

void setup()
{
  pinMode(led1_pin, OUTPUT); // 3번 핀 출력 설정
  pinMode(led2_pin, OUTPUT); // 4번 핀 출력 설정
  pinMode(led3_pin, OUTPUT); // 5번 핀 출력 설정
  pinMode(led4_pin, OUTPUT); // 6번 핀 출력 설정

  // LED 모두 off
  digitalWrite(led1_pin, LOW); // 3번 핀 LOW 출력(off)
  digitalWrite(led2_pin, LOW); // 4번 핀 LOW 출력(off)
  digitalWrite(led3_pin, LOW); // 5번 핀 LOW 출력(off)
  digitalWrite(led4_pin, LOW); // 6번 핀 LOW 출력(off)
}
```

## 실습4: 4개의 순차점멸(LED 풀다운 연결)

### ❖ ex04/app.ino

```
void loop()
{
    // LED1만 ON
    digitalWrite(led4_pin, LOW); // LED4 off
    digitalWrite(led1_pin, HIGH); // LED1 on
    delay(1000);

    // LED2만 ON
    digitalWrite(led1_pin, LOW); // LED1 off
    digitalWrite(led2_pin, HIGH); // LED2 on
    delay(1000);

    // LED3만 ON
    digitalWrite(led2_pin, LOW); // LED2 off
    digitalWrite(led3_pin, HIGH); // LED3 on
    delay(1000);

    // LED4만 ON
    digitalWrite(led3_pin, LOW); // LED3 off
    digitalWrite(led4_pin, HIGH); // LED4 on
    delay(1000);
}
```

## 실습5: 4개의 순차점멸(LED 풀다운 연결) 배열 이용

### ❖ ex05/app.ino

```
// 4개의 순차점멸(LED 풀업 연결) 배열 이용
const int led_pin[4] = {3, 4, 5, 6}; // LED 연결 핀 번호

// LED 연결핀에 차례로 출력할 출력 값
const int led_out[4][4] = {
    {1, 0, 0, 0},
    {0, 1, 0, 0},
    {0, 0, 1, 0},
    {0, 0, 0, 1}
};

int out_no = 0; // 출력 값 출력 순서 번호(0-3)
void setup() {
    int n;

    for(n = 0; n < 4; n++){
        pinMode(led_pin[n], OUTPUT); // led_pin[n]번핀 출력 설정
        digitalWrite(led_pin[n], LOW); // led_pin[n]번핀 LOW 출력
    }
}
```

## 실습5: 4개의 순차점멸(LED 풀다운 연결) 배열 이용

### ❖ ex05/app.ino

```
void loop() {  
    int n;  
  
    for(n = 0;n < 4;n++){  
        // led_pin[n]번핀에 out_no 패턴 출력  
        digitalWrite(led_pin[n], led_out[out_no][n]);  
    }  
  
    out_no++;    // 다음 출력 패턴 번호 설정  
    if(out_no == 4) out_no = 0;    // 마지막 다음은 처음 패턴으로  
  
    delay(1000);  
}
```

# Led 클래스

---

## ❖ Led 클래스

- Led 제어 코드 재사용을 위해 클래스화
- 개발 완료 후 라이브러리 코드로 등록
  - 문서\Arduino\libraries 폴더에 배치

# Led 클래스

---

## ❖ ex06/Led.h

```
#pragma once

#include <Arduino.h>

class Led {
protected:
    int pin;    // 연결할 핀 번호

public:
    Led(int pin);

    void on();
    void off();
    void setValue(int value);
    int toggle();
};
```

# Led 클래스

---

## ❖ ex06/Led.cpp

```
#include "Led.h"

Led::Led(int pin): pin(pin) {
    // this->pin = pin;
    pinMode(pin, OUTPUT);
}

void Led::on() {
    digitalWrite(pin, HIGH);
}

void Led::off() {
    digitalWrite(pin, LOW);
}

void Led::setValue(int value) {
    digitalWrite(pin, value);
}

int Led::toggle() {
    int v = !digitalRead(pin);
    digitalWrite(pin, v);
    return v;
}
```



# Led 클래스

---

## ❖ ex06/app.ino

```
#include "Led.h"

Led leds[4] = { Led(3), Led(4), Led(5), Led(6) };

int out_no = 0;  // 출력 값 출력 순서 번호(0-3)

void setup() {
}

void loop() {
    int n;

    for(n = 0;n < 4;n++){
        if(n == out_no) {
            leds[n].on();
        } else {
            leds[n].off();
        }
    }
    out_no = (++out_no) % 4;
    delay(1000);
}
```