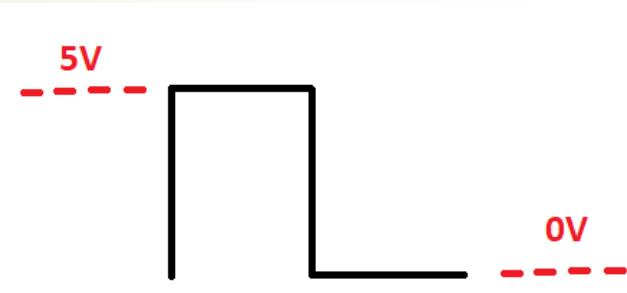


제 4장 디지털 입출력



4.1 디지털 신호

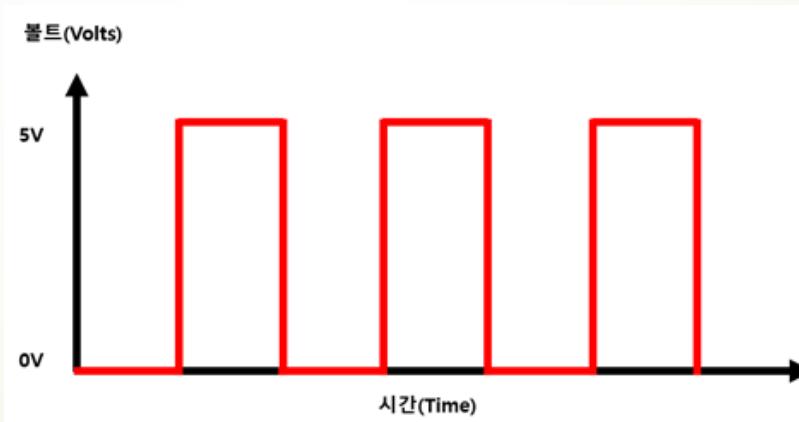
4.2 디지털 출력하기

4.3 디지털 입력받기

4.1 디지털 신호

▶ 디지털 신호

- ▶ 0과 1로 표현할 수 있는 신호
- ▶ 시간에 따라 디지털 신호를 표현하며 1을 전압 값으로 표현하면 5V, 0은 0V를 의미
- ▶ 아두이노에서 5V는 HIGH, 0V는 LOW 값으로 표현



4.2 디지털 출력하기

- ▶ 아두이노 입출력 핀은 입력 모드가 기본 값으로 설정되어 있다.
- ▶ digitalWrite()의 사용 형식

digitalWrite(pin, value);	
매개변수	pin : 아두이노 디지털 핀 번호 value : HIGH 또는 LOW
반환값	없음

- ▶ 교차로 신호등 제어하기

- ▶ 실습부품

- ▶ 아두이노 메가2560 보드, 브레드보드, 빨간색 LED 2개, 노란색 LED 2개, 녹색 LED 2개, 그리고 220 Ω 저항 6개와 점프선 13개가 필요

```
// 첫 번째 신호등  
int red1 = 10;  
int yellow1 = 9;  
int green1 = 8;  
  
// 두 번째 신호등  
int red2 = 13;  
int yellow2 = 12;  
int green2 = 11;  
  
void setup() {  
    // 첫 번째 신호등  
    pinMode(red1, OUTPUT);  
    pinMode(yellow1, OUTPUT);  
    pinMode(green1, OUTPUT);  
  
    // 두 번째 신호등  
    pinMode(red2, OUTPUT);  
    pinMode(yellow2, OUTPUT);  
    pinMode(green2, OUTPUT);  
}  
  
void loop() {  
    changeLights();  
    delay(15000);
```

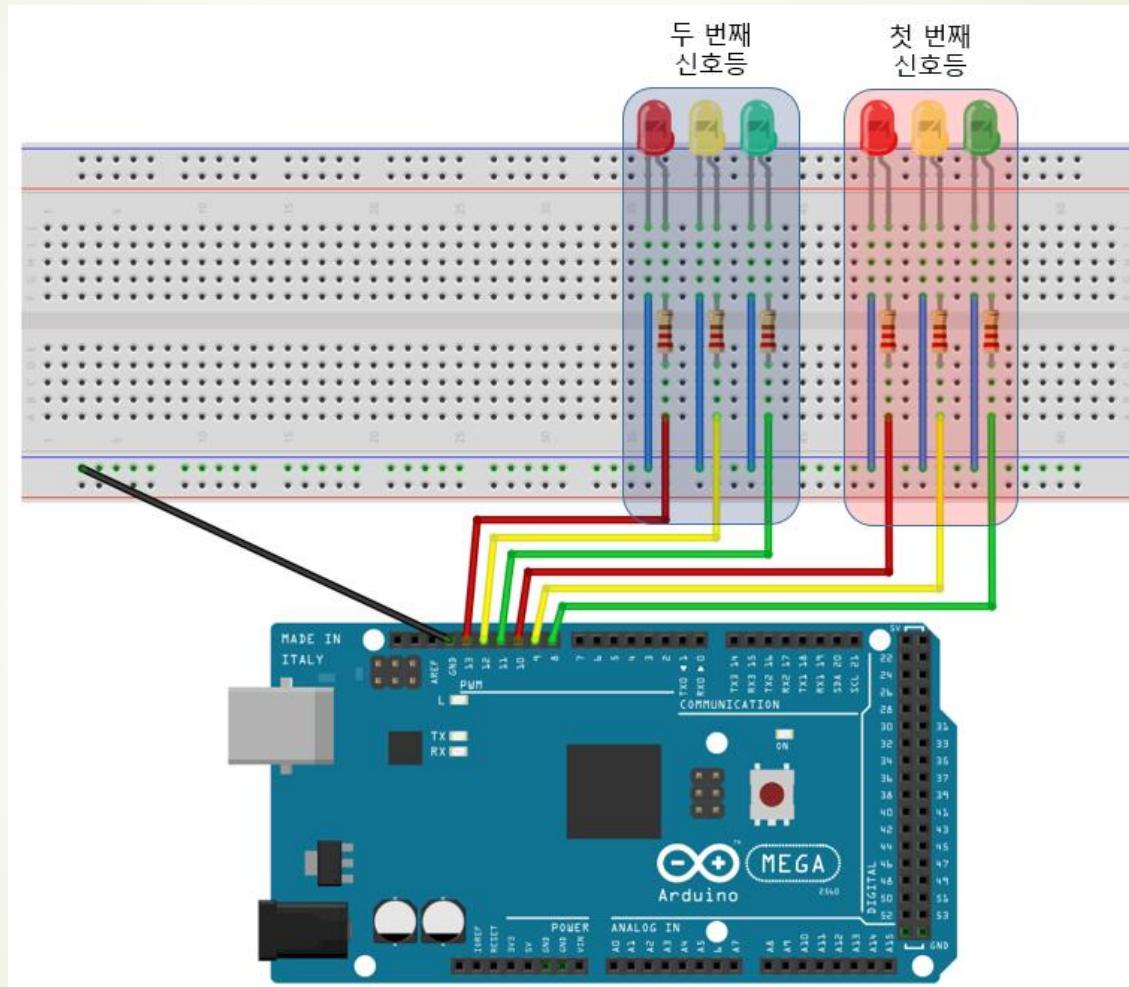
4. 디지털 입출력

```
}  
  
void changeLights() {  
    // 노란 신호를 둘 다 켠다.  
    digitalWrite(green1, LOW);  
    digitalWrite(yellow1, HIGH);  
    digitalWrite(yellow2, HIGH);  
    delay(5000);  
  
    // 노란 신호등 둘 다 끄고, 빨간 신호등과 건너 편에  
    // 녹색 신호등을 켠다.  
    digitalWrite(yellow1, LOW);  
    digitalWrite(red1, HIGH);  
    digitalWrite(yellow2, LOW);  
    digitalWrite(red2, LOW);  
    digitalWrite(green2, HIGH);  
    delay(5000);  
  
    // 노란 신호등 둘 다 다시 켠다.  
    digitalWrite(yellow1, HIGH);  
    digitalWrite(yellow2, HIGH);  
    digitalWrite(green2, LOW);  
    delay(3000);  
  
    // 노란 신호등 둘 다 끄고, 빨간색과 건너 편에 녹색 신호등을 켠다.  
    digitalWrite(green1, HIGH);  
    digitalWrite(yellow1, LOW);  
    digitalWrite(red1, LOW);  
    digitalWrite(yellow2, LOW);  
    digitalWrite(red2, HIGH);  
    delay(5000);  
}
```

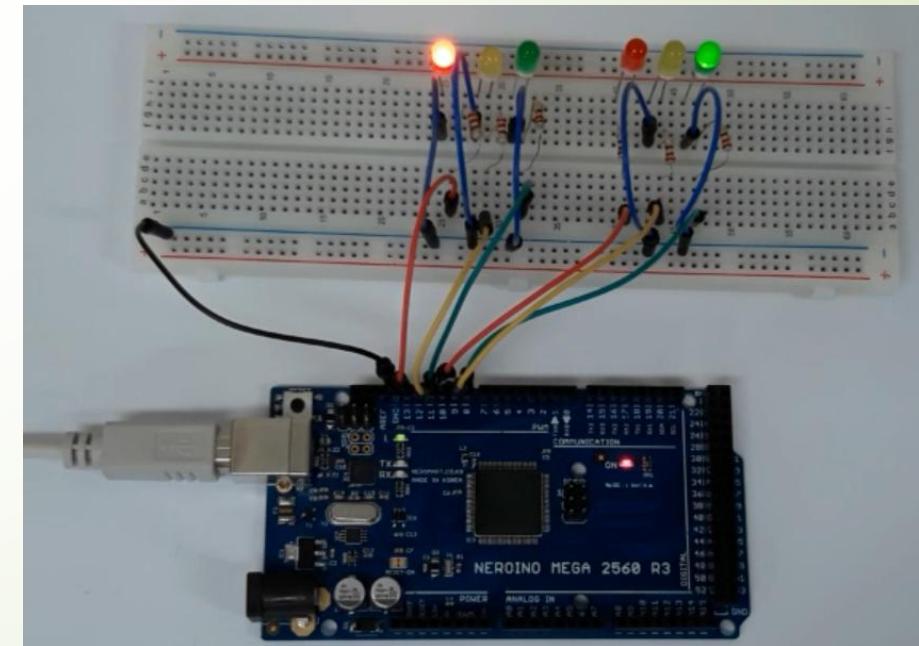
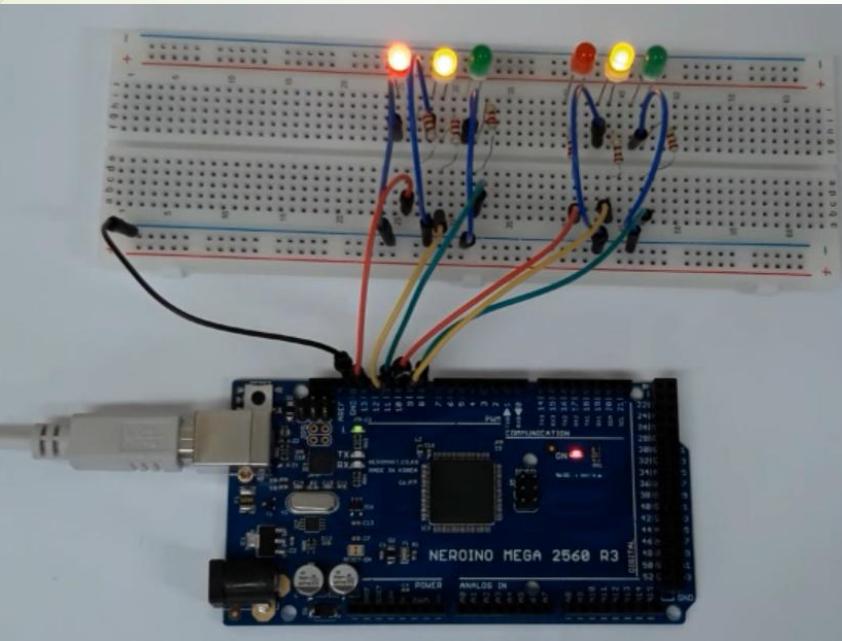
[스케치] 4-1] 교차로 신호등 제어하기를 위한 스케치

5

▶ 교차로 신호등 제어하기를 위한 아두이노 연결회로



- ▶ 교차로 신호등 제어하기를 위한 스케치, 교재 pp. 52~53, 참고
- ▶ 스케치 실행



4.3 디지털 입력받기

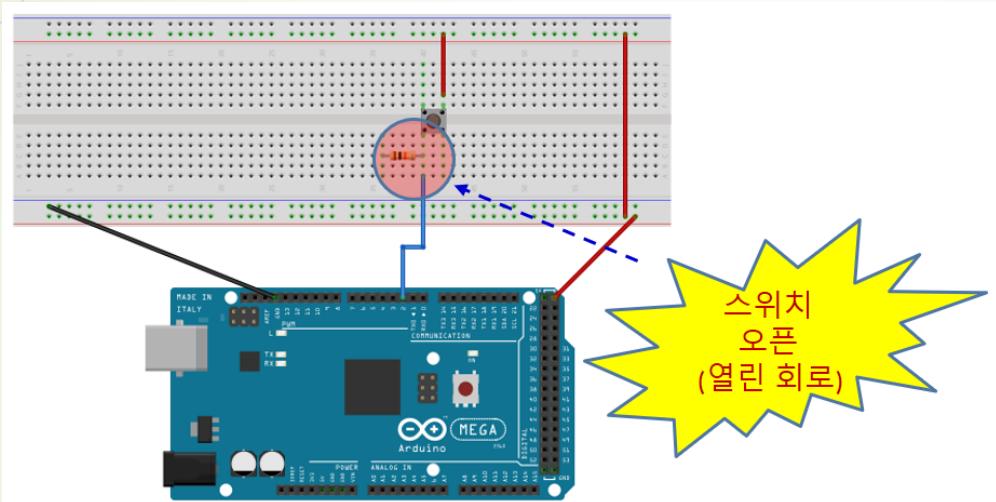
- ▶ 아두이노의 디지털 핀으로부터 값 (HIGH 또는 LOW)을 읽어오기 위하여 `digitalRead()`를 사용
- ▶ 아두이노의 디지털 핀은 기본적으로 ‘INPUT’ 모드 설정되어 있다.
- ▶ `digitalRead()`의 사용 형식

digitalRead(pin)	
매개변수	pin : 아두이노 디지털 핀 번호
반환값	HIGH 또는 LOW

- ▶ 플로팅 현상
 - ▶ 스위치가 연결되지 않은 상태에서 전류가 흐르는지 안 흐르는지 알 수 없는 상태
 - ▶ 아두이노 내의 주변회로의 영향으로 또는 정전기 등에 의해 디지털 입력 핀에 임의의 값이 입력되어 플로팅이 발생
 - ▶ 회로에서 스위치가 열려 있어 회로의 상태가 불연속적일 때의 회로의 상태를 열린회로라고 한다.

▶ 열린회로의 예와 플로팅 현상

▶ 열린 회로



▶ 플로팅 현상

floating

```

/*
DigitalReadSerial
핀 2의 디지털 입력을 읽고, 그 결과를 시리얼 모니터에 출력한다.
*/
int pushButton = 2;

void setup() {
  Serial.begin(9600);
  pinMode(pushButton, INPUT);
}

void loop() {
  // 입력 핀을 읽는다:
  int buttonState = digitalRead(pushButton);
  // 버튼의 상태를 출력한다.
  Serial.println(buttonState);
  delay(10);
}

```

COM7

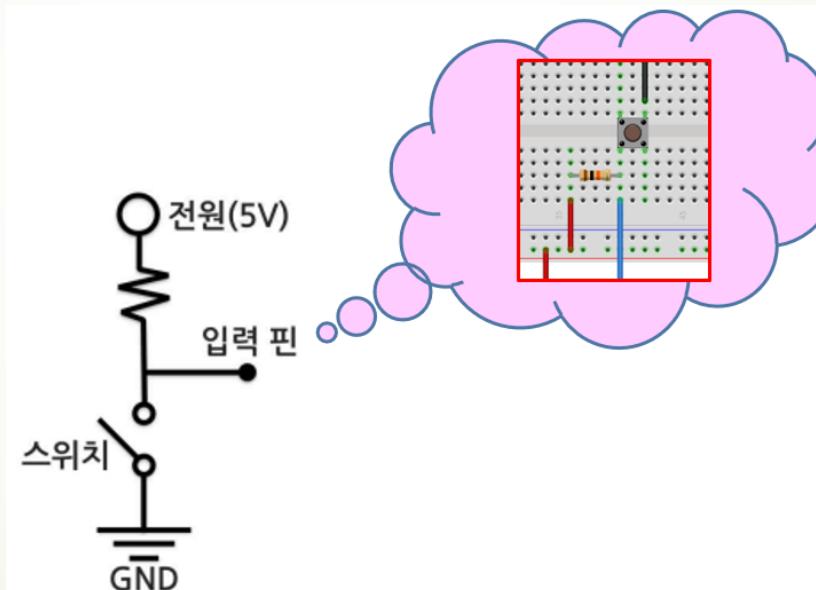
1
1
0
1
0
0
1
0
1
0
0
1
0
1
0
0
1
0
1
0
0
1

플로팅

자동 스크롤

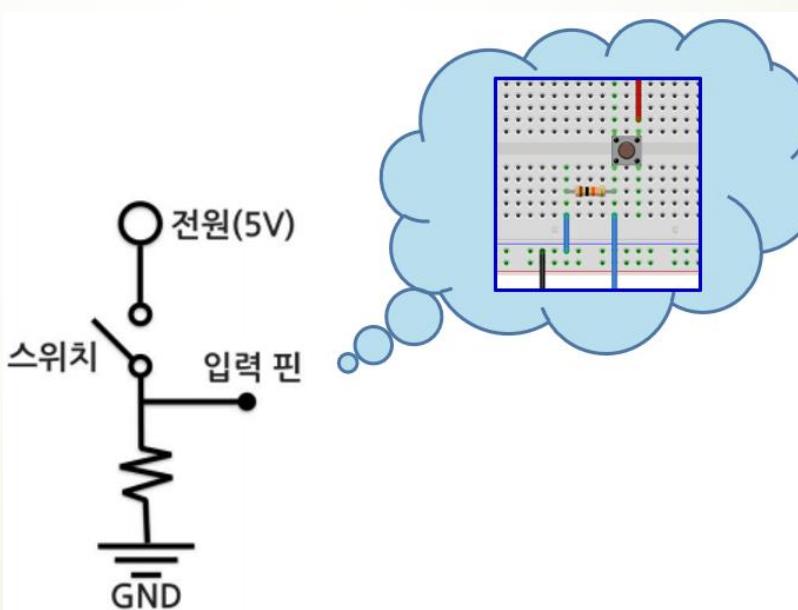
▶ 풀업 저항

- ▶ 풀업(PULL-UP)이라는 말의 의미는 플로팅 상태일 때의 값을 끌어올린다는 의미
- ▶ 플로팅 상태의 값을 올리기 때문에 스위치가 열려있을 때의 상태는 1이 된다.
- ▶ 저항을 VCC 단자에 연결시켜 스위치가 열려진 상태인 경우 입력 핀이 전원과 연결이 되어 있으므로 전압이 5V(HIGH, 1)가 되고, 스위치를 누르면 입력 핀의 전압은 접지와 동일한 0V가 된다.



▶ 풀다운 저항

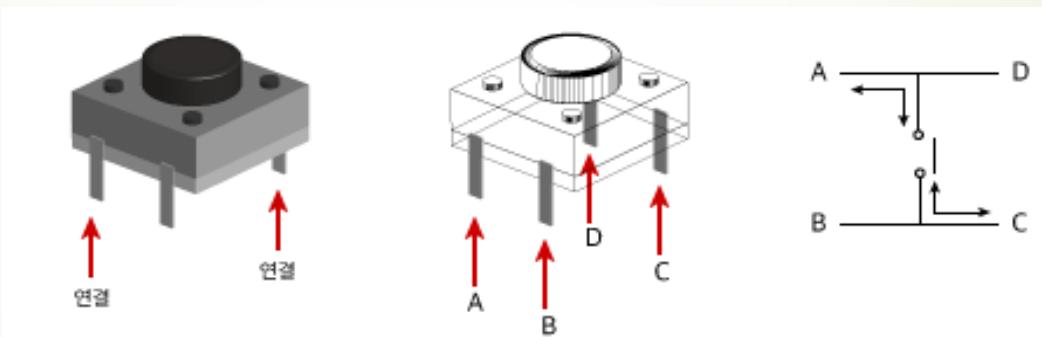
- ▶ 풀다운(PULL-DOWN)이라는 말의 의미는 플로팅 상태 상태일 때의 값을 끌어내린다는 의미
- ▶ 이것은 플로팅 상태의 값을 내리기 때문에 스위치가 열려있을 때의 상태는 0이 된다.
- ▶ 풀다운 저항은 저항을 GND 단자에 연결시켜 스위치가 열려진 상태인 경우 입력 핀이 접지와 연결되어 있으므로 0V 전압이 걸리게 된다.
- ▶ 스위치를 누르면 입력 핀의 전압은 전원과 동일한 5V가 된다.



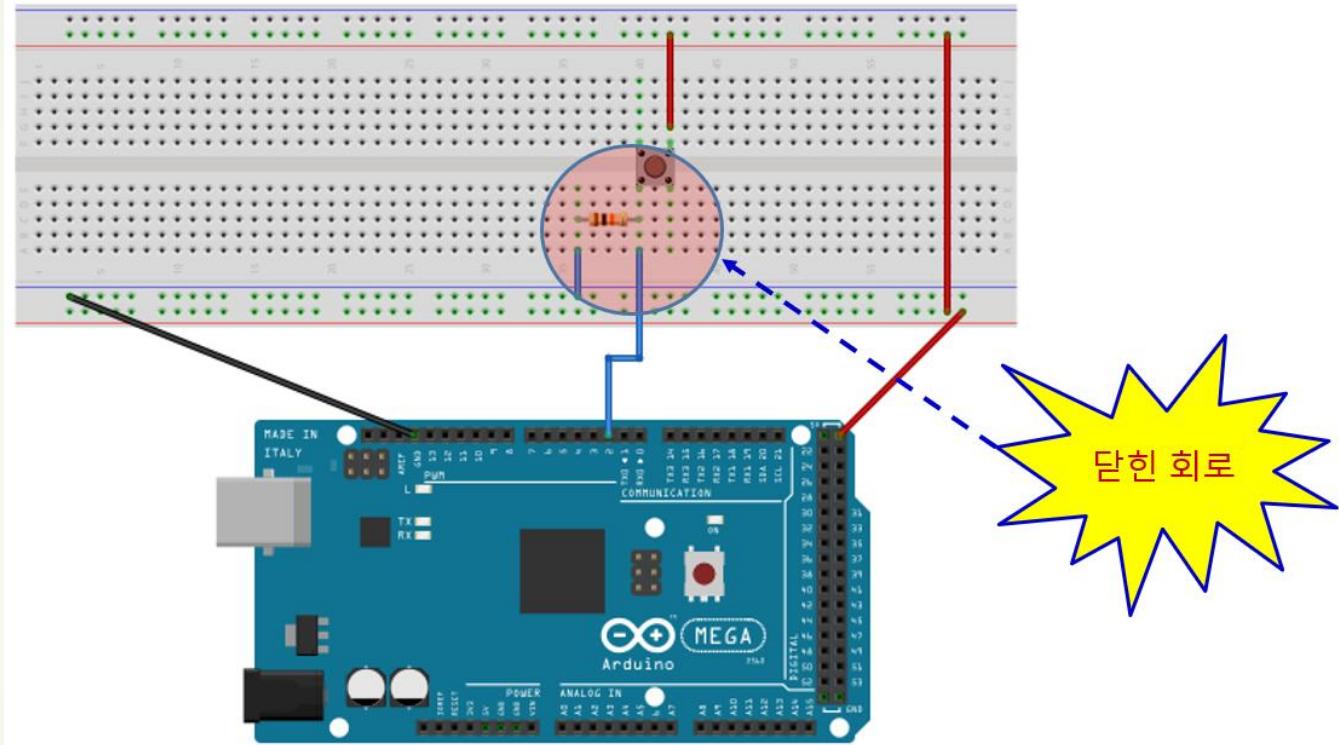
▶ 폴다운 저항 테스트

▶ 실습 부품

- ▶ 실습 부품으로 아두이노 메가2560 보드, 브레드보드, 푸시 버튼 스위치 1개, 10 k Ω 저항 1개, 점프선 6개가 필요
- ▶ 푸시 버튼의 구조와 회로 구성 방법



▶ 아두이노 연결 회로: 풀다운 저항을 갖는 닫힌 회로



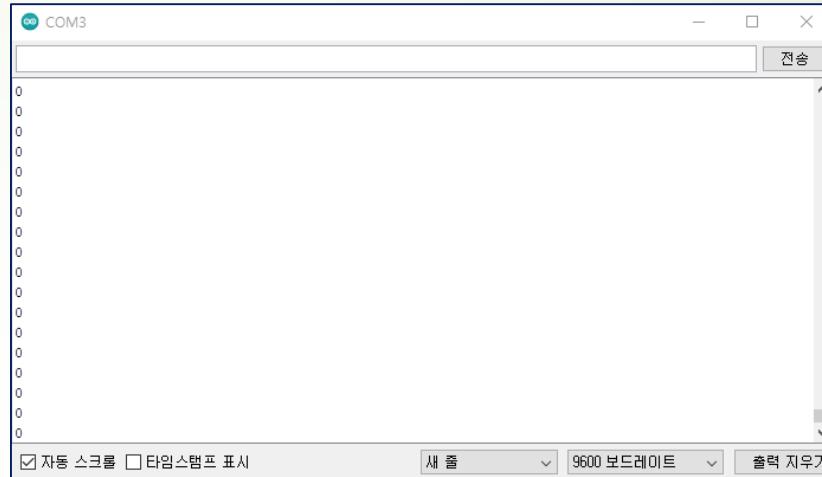
▶ 스케치 작성, 교재 p. 60 참고

```
// DigitalReadSerial  
// 핀 2의 디지털 입력을 읽고, 그 결과를 시리얼 모니터에 출력한다.  
  
int pushButton = 2;  
  
void setup() {  
    Serial.begin(9600);  
    pinMode(pushButton, INPUT);  
}  
  
void loop() {  
    // 입력 핀을 읽는다:  
    int buttonState = digitalRead(pushButton);  
    // 버튼의 상태를 출력한다.  
    Serial.println(buttonState);  
    delay(10);  
}
```

[스케치] 4-2] 디지털 입력을 시리얼 모니터로 출력

▶ 스케치 실행

▶ 버튼을 누르지 않은 경우



▶ 버튼을 누른 경우

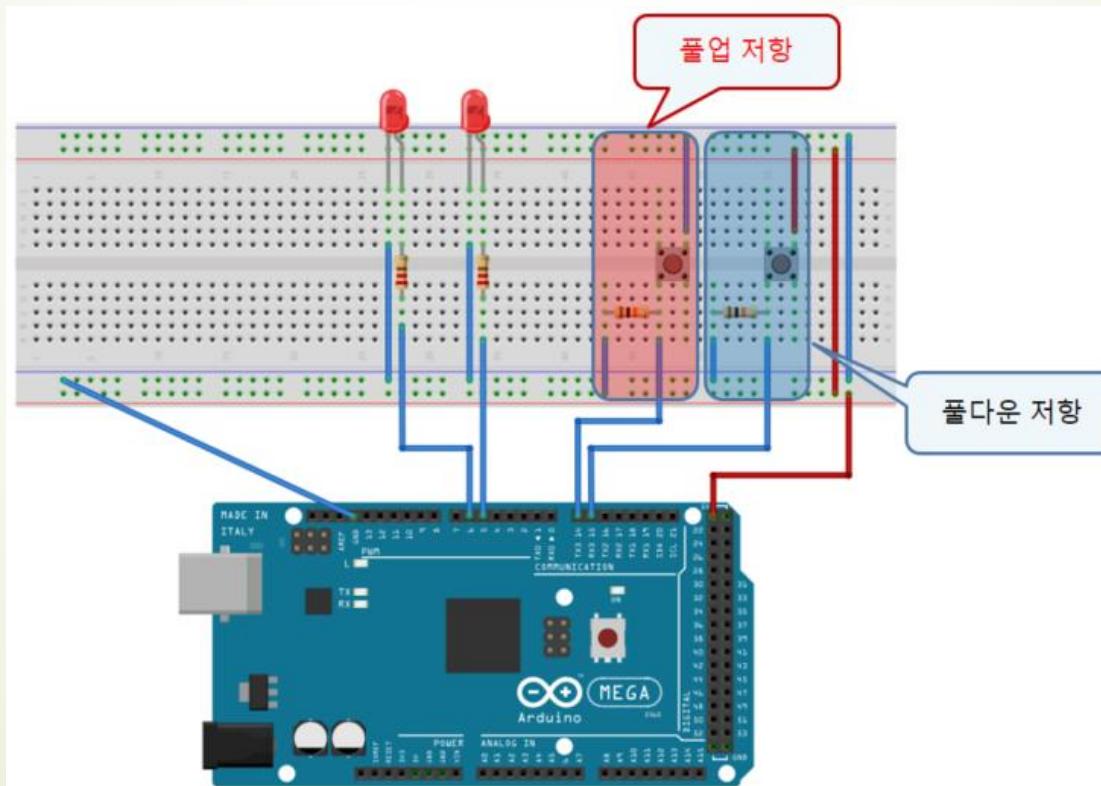


▶ 버튼으로 LED 제어하기

▶ 실습 부품

- ▶ 실습 부품으로 아두이노 메가2560 보드, 브래드보드, $220\ \Omega$ 저항 2개, $10\ k\Omega$ 저항 2개, 푸시버튼 스위치 2개, 빨간색 LED 2개, 점프선 13개가 필요

▶ 아두이노 연결 회로: 버튼으로 LED 제어하기 아두이노 연결회로



```
int pins_LED[] = {5, 6}; // LED 연결 핀
int pins_button[] = {14, 15}; // 버튼 연결 핀 (14 : 풀업 저항, 15 : 풀다운 저항)

void setup() {
    Serial.begin(9600); // 시리얼 통신 초기화
    for (int i = 0; i < 2; i++) {
        pinMode(pins_button[i], INPUT); // 버튼 연결 핀을 입력으로 설정
        pinMode(pins_LED[i], OUTPUT); // LED 연결 핀을 출력으로 설정
    }
}

void loop() {
    for (int i = 0; i < 2; i++) {
        boolean state = digitalRead(pins_button[i]); // 버튼 상태 읽기
        digitalWrite(pins_LED[i], state); // LED 출력
        Serial.print(state);
        Serial.print(" ");
    }
    Serial.println();
    delay(1000);
}
```

[스케치] 4-3] 버튼으로 LED 제어하기

- ▶ 스케치 작성: 버튼으로 LED 제어하기, 교재 p. 63 참고
- ▶ 스케치 실행
 - ▶ 풀업 저항 버튼을 누른 경우

