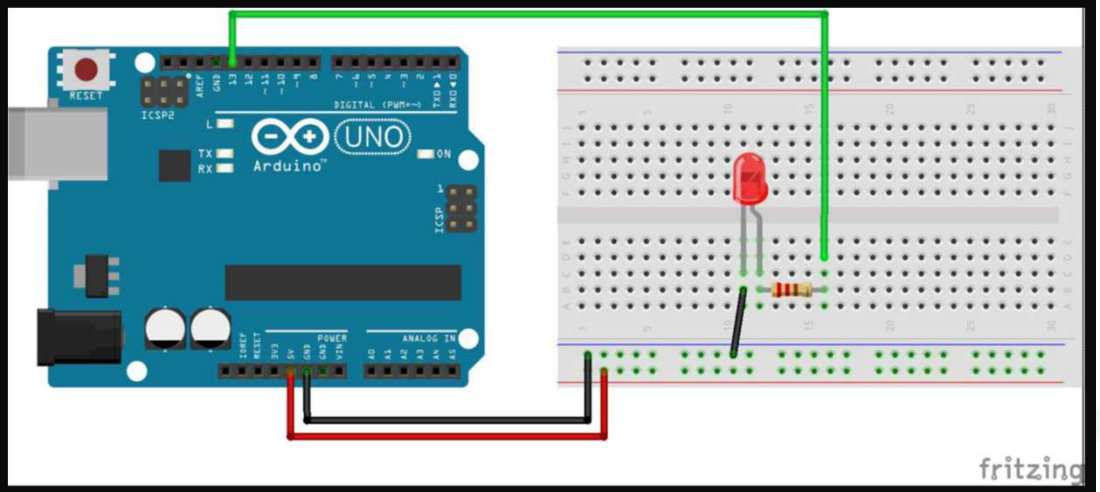
아두이노 시리얼 통신 제어



Step1 핀모드와 시리얼 통신 설정

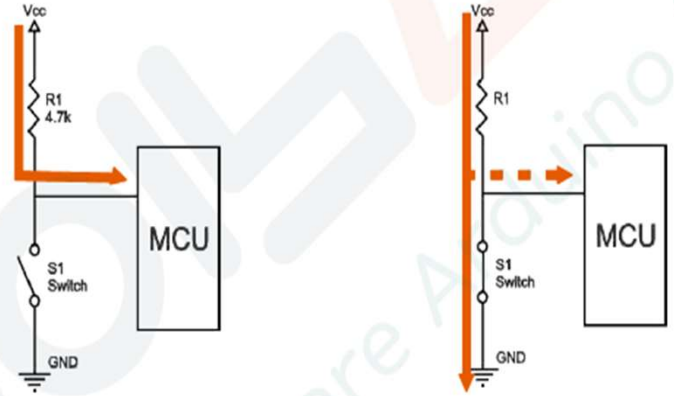
```
1  const int ledPin =13;          //  LED pin을 13번으로 설정
2  int incomingByte;              //  시리얼 데이터를 저장하기 위한 정수형 변수 설정
3
4  void setup() {
5
6    Serial.begin(9600);          //  9600 속도로 시리얼 통신 시작
7
8    pinMode(ledPin,OUTPUT);      // LED pin을 출력으로 설정
9  }
10
```

Step2 반복실행문

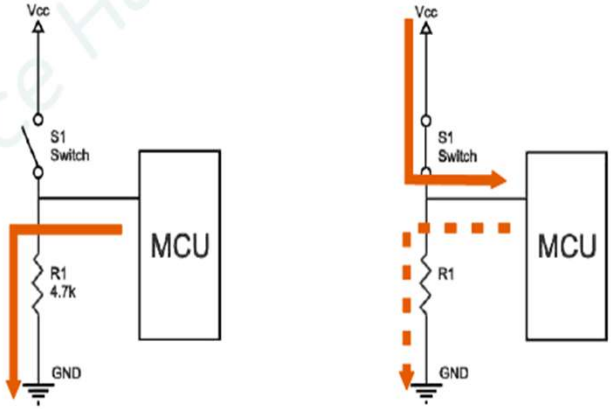
```
10
11 void loop() {
12
13   if(Serial.available() >0) {          // 시리얼 데이터가 들어오면
14
15     incomingByte =Serial.read();       // 데이터를 읽어서 incomingByte 변수에 넣고
16
17     if(incomingByte =='H') {                 // incomingByte 변수에 들어오 값이 H면
18       digitalWrite(ledPin,HIGH);             // LED를 켜라.
19     }
20
21     if(incomingByte =='L') {                 // incomingByte 변수에 들어오 값이 L면
22       digitalWrite(ledPin,LOW);              // LED를 꺼라.
23     }
24   }
25 }
```

디지털 입력

- Pull-up : 입력 핀의 평소 상태를 HIGH로 연결한다. 저항이 전원 5V에 연결된 상태
  digitalRead에서 HIGH는 0, LOW는 1 값을 리턴한다.

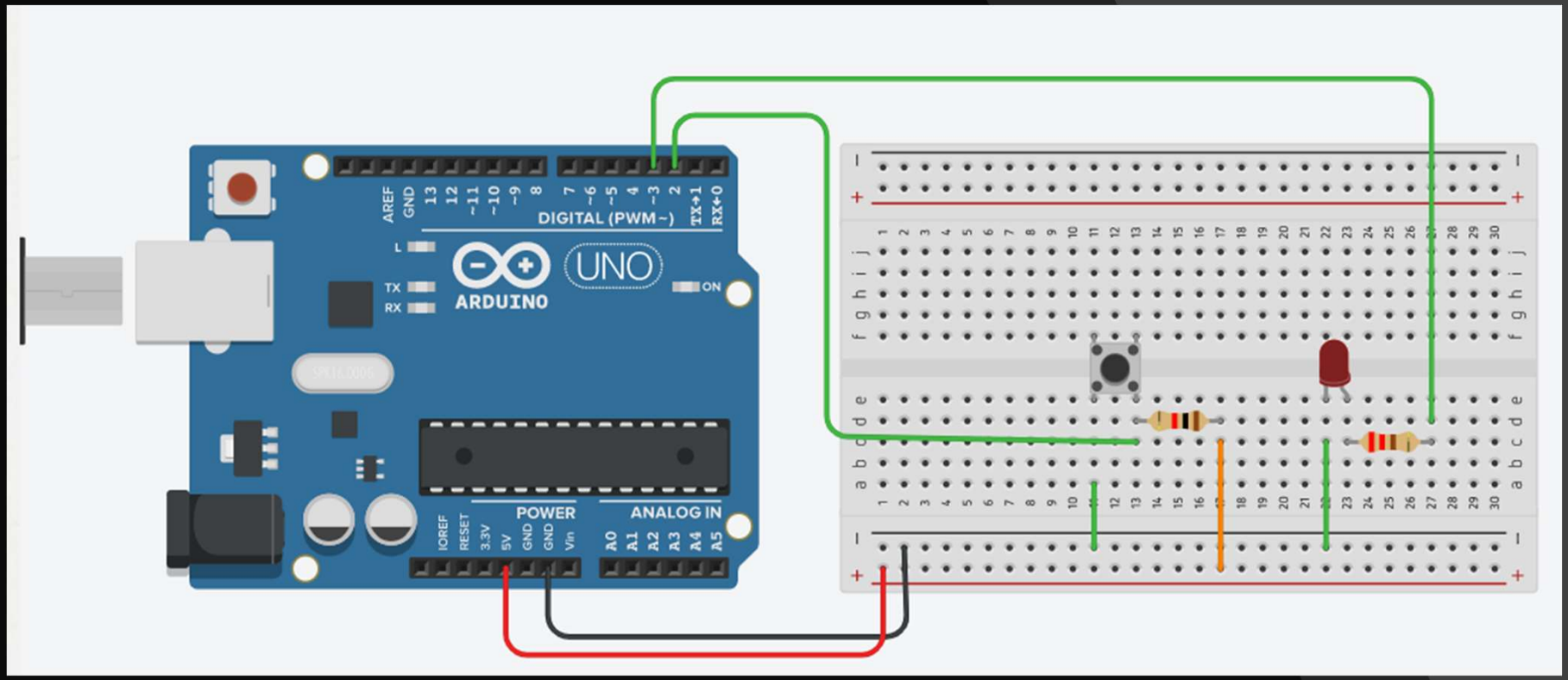


- Pull-down : 입력 핀의 평소 상태를 LOW로 연결한다. 저항이 GND (0V)에 연결된 상태
  digitalRead에서 HIGH는 1, LOW는 0 값을 리턴한다.



pinMode(핀 번호, INPUT_PULLUP) 을 이용하면 보드에 내장된 20KΩ 풀업 저항을 사용할 수 있다.
INPUT_PULLUP 으로 구성된 핀에 센서를 연결할 때는 다른 쪽 끝을 GND (0V) 에 연결해야 한다.

디지털 입력


AREF
GND
13
12
~11
~10
~9
8
7
~6
~5
4
~3
2
TX→1
RX←0
DIGITAL (PWM~)
L
TX
RX
ARDUINO
UNO
ON
POWER
ANALOG IN
IOREF
RESET
3.3V
5V
GND
GND
Vin
A0
A1
A2
A3
A4
A5

디지털 입력

Step1 시리얼 통신시작과 핀 모드 설정

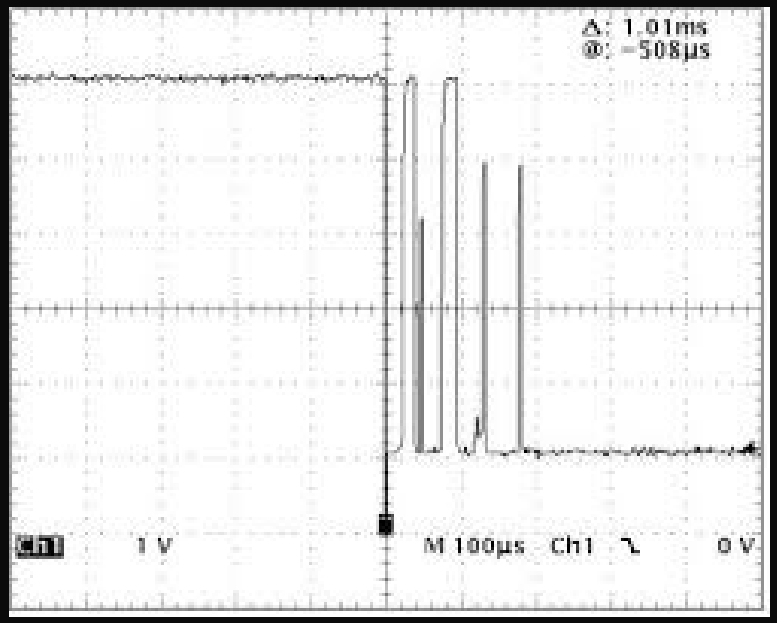
```
1  void setup() {
2
3    Serial.begin(9600);
4
5    pinMode(2, INPUT_PULLUP);          // 2 번 핀을 풀업 저항으로 초기화
6    pinMode(13, OUTPUT);               // 내장 LED 핀을 출력으로 설정
7
8  }
9
```

Step2 반복 실행문

```
10 void loop() {
11
12   int sensorVal = digitalRead(2);  // sensorVal 변수에 2 번 핀의 버튼 입력 값을 받는다.
13
14   Serial.println(sensorVal);       // 버튼의 값을 시리얼 창에 한 줄씩 나타내라.
15
16                // pull-up 은 버튼이 열려있을 때 HIGH 값을, 눌렀을 때 LOW 값을 준다.
17
18   if(sensorVal == HIGH) {          // 버튼이 열려있으면
19     digitalWrite(13, LOW);         // LED 끄고
20   } else{                          // 버튼이 눌려있으면
21     digitalWrite(13, HIGH);        // LED 를 켜라.
22   }
23 }
```

디지털 입력 – Debounce

버튼 스위치가 열리고 닫힐 때 종종 기계적 및 물리적 문제로 잘못된 신호를 발생한다. 이러한 상황은 프로그램을 속일 수 있는 매우 짧은 시간에 여러번 눌러 읽는 방법으로 방지할 수 있는데 이 과정을 디바운싱이라 한다.



디지털 입력 – Debounce

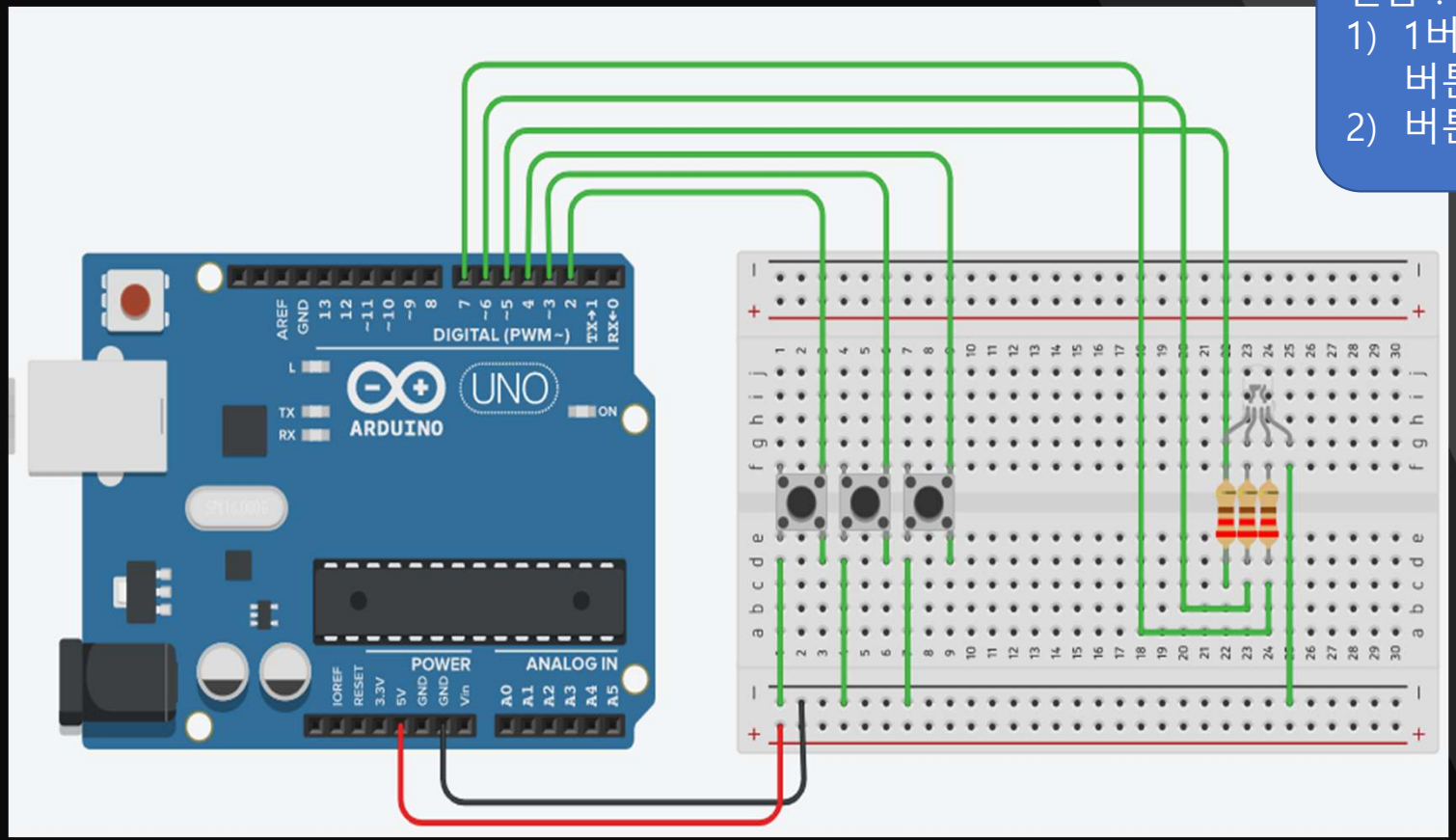
Step1 변수와 핀 모드 설정

```
1  const int buttonPin =2;     // 버튼 스위치 핀을 2 번으로
2  const int ledPin = 13;      // LED 핀을 13 번으로 설정
3
4  int ledState =HIGH;          // 출력핀을 HIGH 로 설정
5  int buttonState;             // 현재 버튼 스위치 상태를
6  int lastButtonState =LOW;    // 이전 버튼 스위치 상태를
7
5  unsigned long lastDebounceTime =0;    // 출력핀이 마지
8  unsigned long debounceDelay =50;      // 안정된 상태가
9
10 void setup() {
11   pinMode(ledPin, OUTPUT);            // LED 핀을 출력
12   pinMode(buttonPin, INPUT);          // 버튼스위치 핀
13
14   digitalWrite(ledPin,ledState);      // ledState에 따
15 }
16
```

Step2 반복 실행문

```
16
17 void loop() {
18
19   int reading = digitalRead(buttonPin);   // 버튼 상태를 읽어 reading 변수에 저장
20
21   if(reading != lastButtonState) {     // 노이즈 또는 누르기로 버튼 상태가 변경되면
22
23     lastDebounceTime =millis();          // debouncing timer 를 재설정하고
24   }
25
26   // 읽은 값이 무엇이든 debounce 지연보다 오래 있었으면
27   if((millis() -lastDebounceTime) > debounceDelay) {
28     if(reading != buttonState) {             // 버튼 상태가 바뀌면
29       buttonState =reading;                  // 버튼 상태를 저장하고
30
31       if(buttonState ==HIGH) {                    // 새로운 버튼 상태가 HIGH 면
32         ledState =!ledState;                      // LED 상태를 바꾼다
33       }
34     }
35   }
36
37   digitalWrite(ledPin,ledState);         // ledState 에 저장된 값으로 LED 를 on/off
39   // reading 변수의 값을 lastButtonState (다음 loop 에서 사용)에 저장
40   lastButtonState =reading;
41 }
```

디지털 입력 – 예제



실습 : RGB LED 제어
1) 1버튼 : RED, 2버튼 : GREEN, 3 버튼 : BLUE
2) 버튼 조합 가능