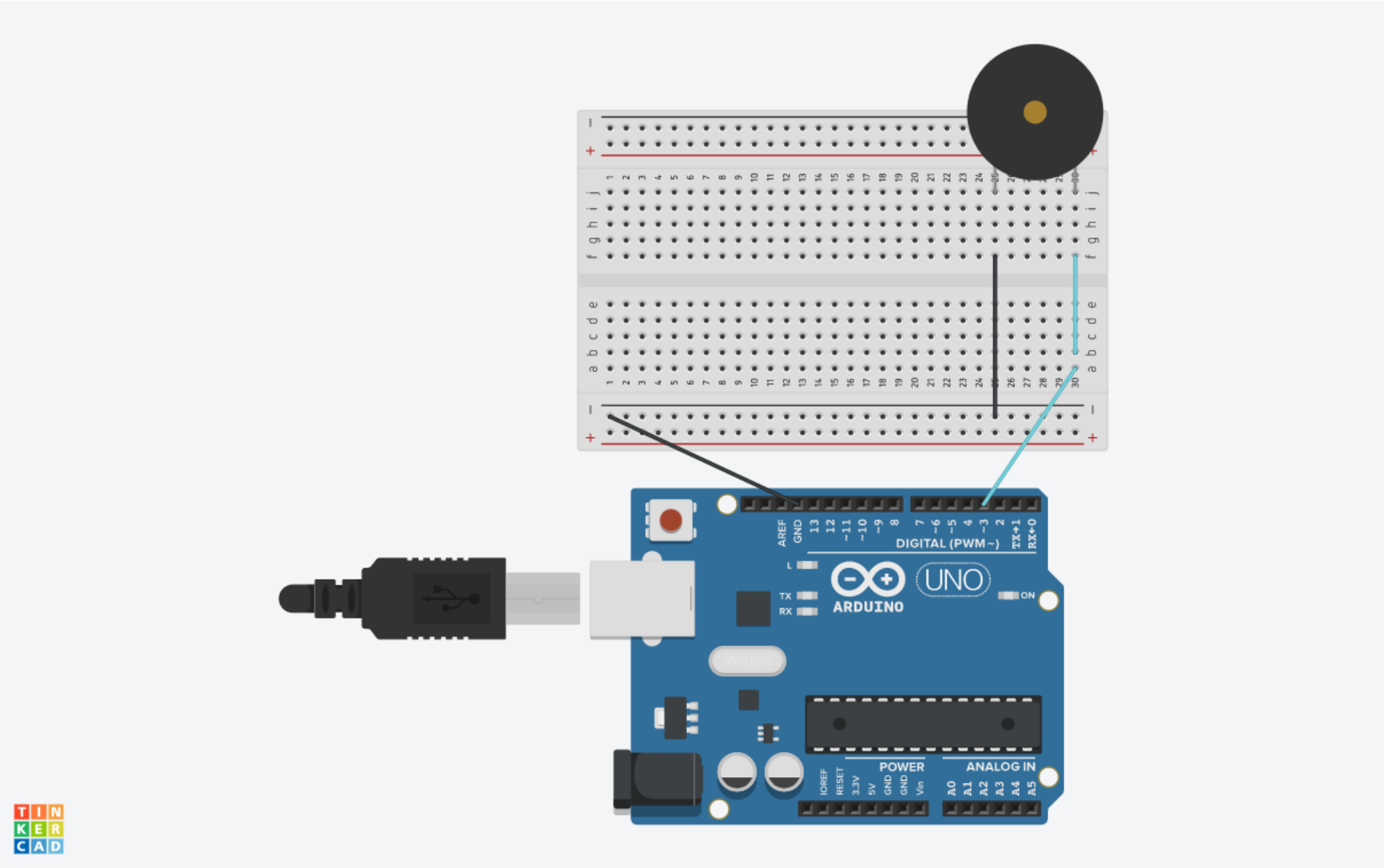
아두이노
센서 활용

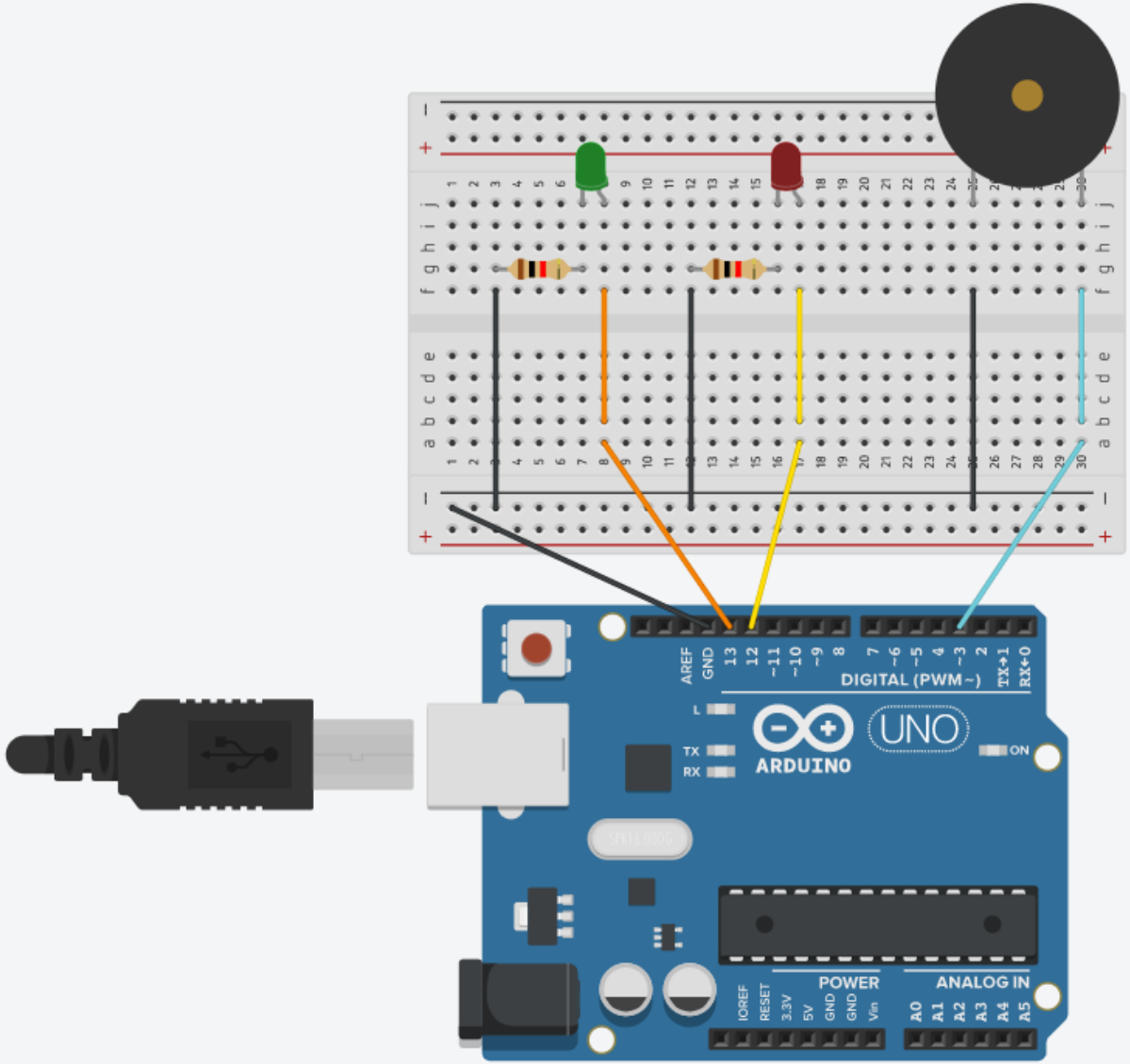
부저 사용하기



▶ 부저 사용하기

```
1  int buzzerPin = 3;              // 피에조 부저 핀 등록
2
3  void setup(){
4    pinMode(buzzerPin, OUTPUT);       // 피에조 부저 핀 선언
5  }
6
7  void loop() {
8    // 피에조 부저는 PWM(펄스 폭 진동 : 0 ~ 255)을 통하여
9    // 소리를 내기 때문에 ~표시가 있는 디지털 핀 번호를 사용해야 함
10
11   analogWrite(buzzerPin, 64);             // PWM 25% 적용
12   delay(1000);                            // 1초 대기
13
14   analogWrite(buzzerPin, 128);            // PWM 50% 적용
15   delay(1000);                            // 1초 대기
16
17  analogWrite(buzzerPin, 256);             // PWM 100% 적용
18   delay(1000);                            // 1초 대기
19 }
```

응용 (뮤직박스)



▶ 응용 (뮤직박스)

```
1  // 부저에서 PWM 50% 대역에서 사용할 주파수 설정 (31~65535Hz)
2  // like 풀잎피리의 떨림을 통한 소리
3  const int c = 261, d = 294, e = 329, f = 349, g = 391;
4  const int gS = 415, a = 440, aS = 455, b = 466, cH = 523;
5  const int cSH = 554, dH = 587, dSH = 622, eH = 659, fH = 698;
6  const int fSH = 740, gH = 784, gSH = 830, aH = 880;
7
8  const int buzzerPin = 3;
9  const int ledPin1 = 12;
10 const int ledPin2 = 13;
11
12 // LED를 번갈아 켜기 위한 변수
13 int counter = 0;
```

▶ 응용 (뮤직박스)

```
15 void setup()
16 {
17   // 핀 설정하기
18   pinMode(buzzerPin, OUTPUT);
19   pinMode(ledPin1, OUTPUT);
20   pinMode(ledPin2, OUTPUT);
21 }
22
23 void loop()
24 {
25
26   // 첫번째 파트 재생
27   firstSection();
28   // 두번째 파트 재생
29   secondSection();
30
31   // 중간 연결 부
32   beep(f, 250);
33   beep(gS, 500);
34   beep(f, 350);
35   beep(a, 125);
36   beep(cH, 500);
37   beep(a, 375);
38   beep(cH, 125);
39   beep(eH, 650);
40
41   delay(500);
42
43   // 두번째 파트 재생
44   secondSection();
45
46   // 마무리 음악
47   beep(f, 250);
48   beep(gS, 500);
49   beep(f, 375);
50   beep(cH, 125);
51   beep(a, 500);
52   beep(f, 375);
53   beep(cH, 125);
54   beep(a, 650);
55
56   delay(1000);
57 }
```

▶ 응용 (뮤직박스)

```
59 void beep(int note, int duration) {
60   // 부저로 소리를 재생하는 함수
61   tone(buzzerPin, note, duration);
62
63   //'counter' 변수에 따라 LED 깜빡이 변경
64   if(counter % 2 == 0) {
65     digitalWrite(ledPin1, HIGH);
66     delay(duration);
67     digitalWrite(ledPin1, LOW);
68   } else {
69     digitalWrite(ledPin2, HIGH);
70     delay(duration);
71     digitalWrite(ledPin2, LOW);
72   }
73
74   noTone(buzzerPin);  // 음악 재생 중지
75   delay(50);
76
77   counter++;     // counter 변수 증가
78 }
```

✓ **tone()**

tone(핀번호, 주파수, 재생 시간) :

특정 주파수에 맞춰 해당 음을 몇 초 동안 재생할 지 설정하여 음을 출력하는 함수

✓ **noTone()**

noTone(핀번호) :

재생했던 부저를 멈추는 함수

부저는 단순히 주파수에 따라 진동하여 음을 재생하므로 tone() – noTone() 의 순서를 가진다.

즉, tone() 함수의 반복 호출은 불가하다.

▶ 응용 (뮤직박스)

```
80  void firstSection(){
81    beep(a, 500);
82    beep(a, 500);
83    beep(a, 500);
84    beep(f, 350);
85    beep(cH, 150);
86    beep(a, 500);
87    beep(f, 350);
88    beep(cH, 150);
89    beep(a, 650);
90
91    delay(500);
92
93    beep(eH, 500);
94    beep(eH, 500);
95    beep(eH, 500);
96    beep(fH, 350);
97    beep(cH, 150);
98    beep(gS, 500);
99    beep(f, 350);
100   beep(cH, 150);
101   beep(a, 650);
102
103   delay(500);
104 }
```

```
106 void secondSection(){
107   beep(aH, 500);
108   beep(a, 300);
109   beep(a, 150);
110   beep(aH, 500);
111   beep(gSH, 325);
112   beep(gH, 175);
113   beep(fSH, 125);
114   beep(fH, 125);
115   beep(fSH, 250);
116
117   delay(325);
118
119   beep(aS, 250);
120   beep(dSH, 500);
121   beep(dH, 325);
122   beep(cSH, 175);
123   beep(cH, 125);
124   beep(b, 125);
125   beep(cH, 250);
126
127   delay(350);
128 }
```

▶ 응용 (뮤직박스)

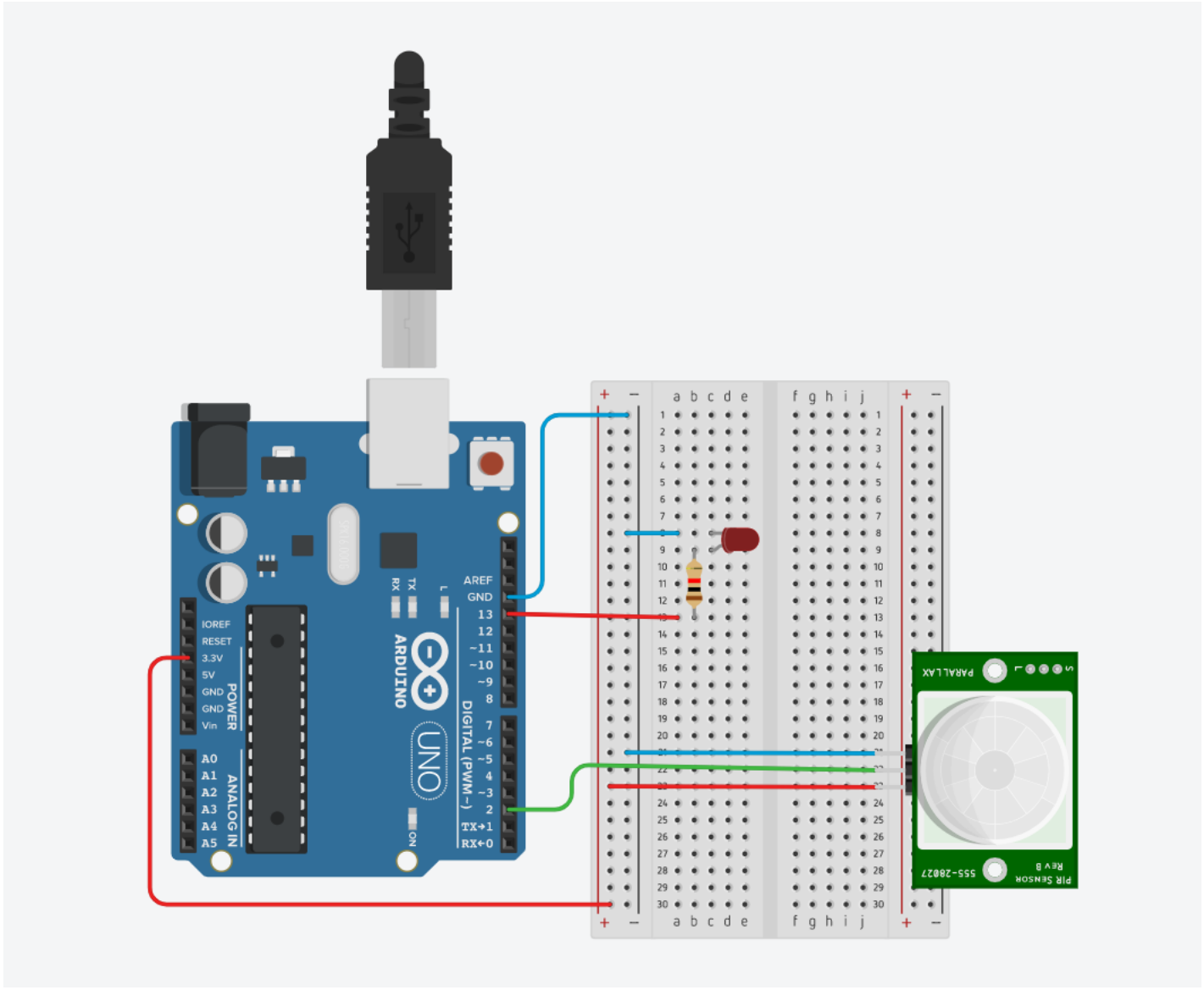
✓ 다른 음악 모음

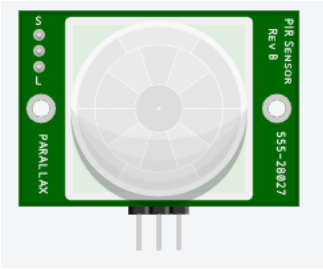
https://dragaosemchama.com/en/2019/02/songs-for-arduino/

https://kocoafab.cc/tutorial/view/626

https://postpop.tistory.com/64

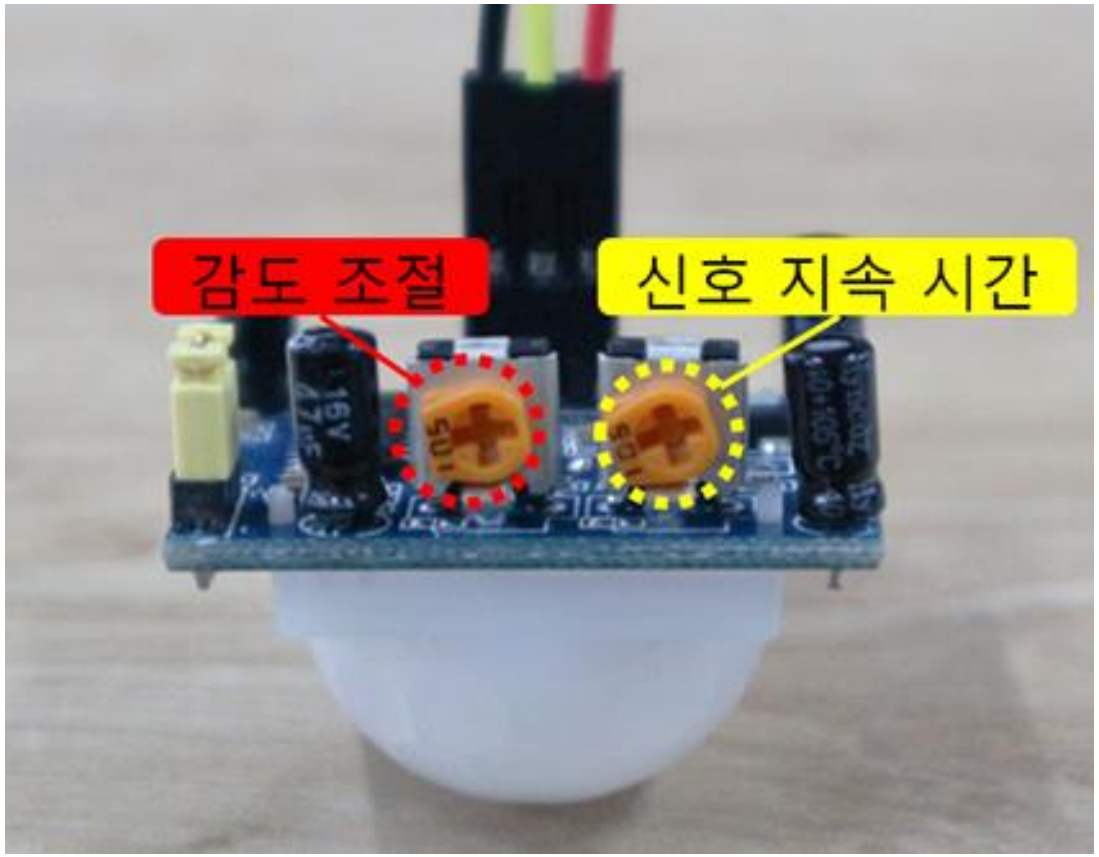
PIR 센서 제어하기





좌 측 기준		
GND	Data	VCC

▶ PIR 센서 제어하기



- **감도조절** (조금만 손대어도 켜지는지) :
  - **시계방향** – 민감도 ↓
  - **반시계방향** – 민감도 ↑
- **지속시간** (데이터 유지 시간) :
  - **시계방향** – 지속시간 ↑
  - **반시계방향** – 지속시간 ↓

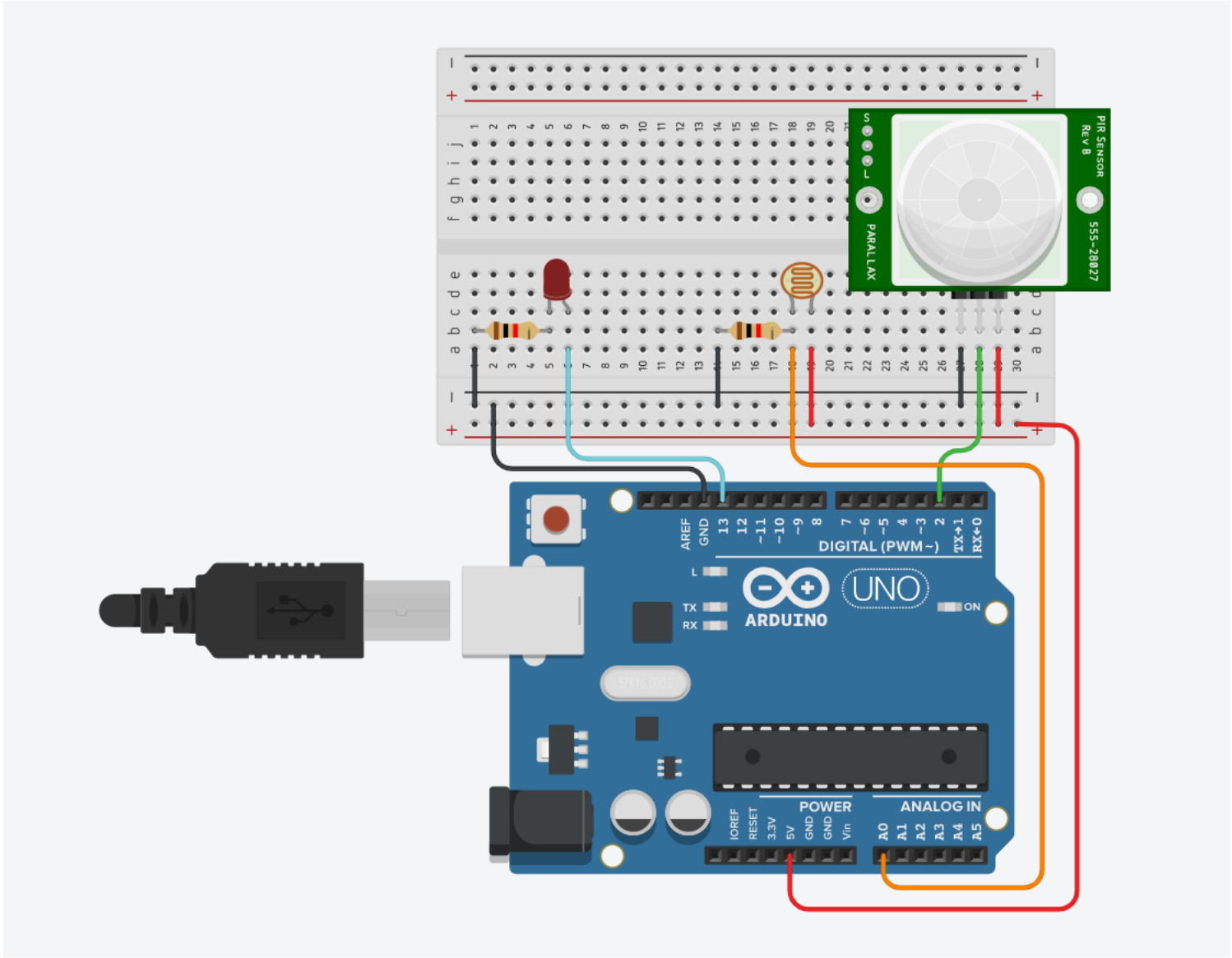
** 적외선 감지 센서는 무척 민감하기 때문에, 실행 후 약 1 ~ 5분 정도 현재 적외선 환경에 적응할 시간을 주어야 함

▶ PIR 센서 제어하기

```
1  const int PIRPin= 2;      // 디지털 핀
2  const int ledPin= 13;     // 디지털 핀
3
4  void setup()
5  {
6    pinMode(ledPin, OUTPUT);
7    pinMode(PIRPin, INPUT);
8
9    Serial.begin(9600); // print Serial interval
10
11 }
```

```
13 void loop()
14 {
15   int value1 = digitalRead(PIRPin);
16   Serial.println(value1); // 시리얼 값 확인
17
18   if (value1 == HIGH) {
19     digitalWrite(ledPin, HIGH);
20     delay(500);
21     digitalWrite(ledPin, LOW);
22
23   } else {
24     digitalWrite(ledPin, LOW);
25   }
26 }
```

▶ 응용 (PIR & 조도 센서)

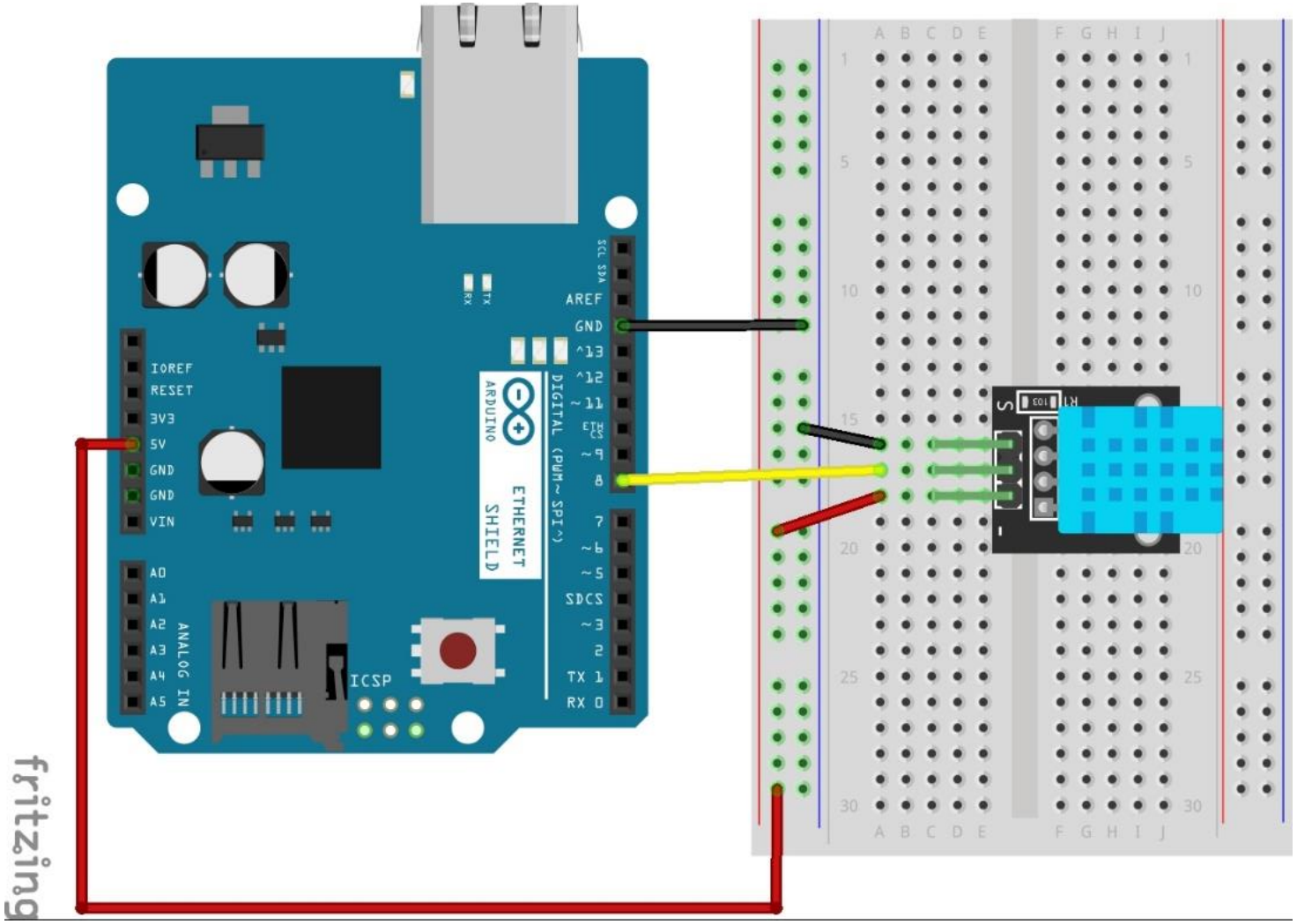

PIR SENSOR
Rev B
PARALLAX
555-28027
DIGITAL (PWM~)
ARDUINO
UNO
POWER
ANALOG IN

▶ 응용 (PIR & 조도 센서)

```
const int PhotoPin = A0; // 아날로그 핀
const int PIRPin= 2;     // 디지털 핀
const int ledPin= 13;    // 디지털 핀

void setup()
{
  pinMode(ledPin, OUTPUT);
  pinMode(PIRPin, INPUT);
  pinMode(PhotoPin, INPUT);

  Serial.begin(9600); // print Serial interval

}
```

```
void loop()
{
  int value1 = digitalRead(PIRPin);
  Serial.println(value1); // 시리얼 값 확인

  if (value1 == HIGH && analogRead(PhotoPin) < 100) {
    digitalWrite(ledPin, HIGH);
    delay(500);
    digitalWrite(ledPin, LOW);

  } else {
    digitalWrite(ledPin, LOW);
  }
}
```

▶ DHT11 센서 제어하기



▶ DHT11 센서 제어하기

```
#include "DHT.h"
#define DHTPIN 8
#define DHTTYPE DHT11
DHT dht(DHTPIN, DHTTYPE);

void setup() {
  Serial.begin(9600);
  dht.begin();
}

void loop() {
  delay(2000);
  int h = dht.readHumidity();
  int t = dht.readTemperature();
  Serial.print("Humidity: ");
  Serial.print(h);
  Serial.print(" %\t");
  Serial.print("Temperature: ");
  Serial.print(t);
  Serial.println(" C");
}
```