

펌웨어 구현

아두이노 기본 문법

▶ 아두이노 기본 문법

✓ 프로그램 기본 구조

```
1 // 라이브러리 포함 부
2 #include <파일명.h>
3 #include "파일명.h"
4
5 // 변수 선언 부
6 int led = 10;
7
8 // setup : 초기 설정 부
9 void setup(){
10
11 }
12 // loop : 반복 수행할 기능
13 void loop(){
14
15 }
```

▶ 아두이노 기본 문법

✓ 전처리문

➤ 파일 처리를 위한 전처리문 : **#include**

➤ 형태 정의를 위한 전처리문 : **#define**

➤ 조건 처리를 위한 전처리문 :

#if : 참이면

#ifdef : 정의 되어 있으면

#else : #if나 #ifdef가 아니면

#elif : else와 if의 의미

#endif : #if, #ifdef, #ifndef가 끝났음을 의미

▶ 아두이노 기본 문법

✓ 기초 함수

➤ **setup() { }** :

설정을 담당(설정과 관련된 함수를 {}안에 선언)

아두이노 스케치가 실행될 때 1회 실행

변수 초기화, 핀 모드 설정, 라이브러리 등을 사용

아두이노에 초기 전원 부여 및 reset 버튼 클릭시 스케치 실행/재실행

```
1 void setup{  
2     pinMode(4, OUTPUT); // 4번 핀을 출력으로 선언  
3     pinMode(5, INPUT); // 5번 핀을 입력으로 선언  
4 }
```

▶ 아두이노 기본 문법

✓ 기초 함수

➤ loop() {} :

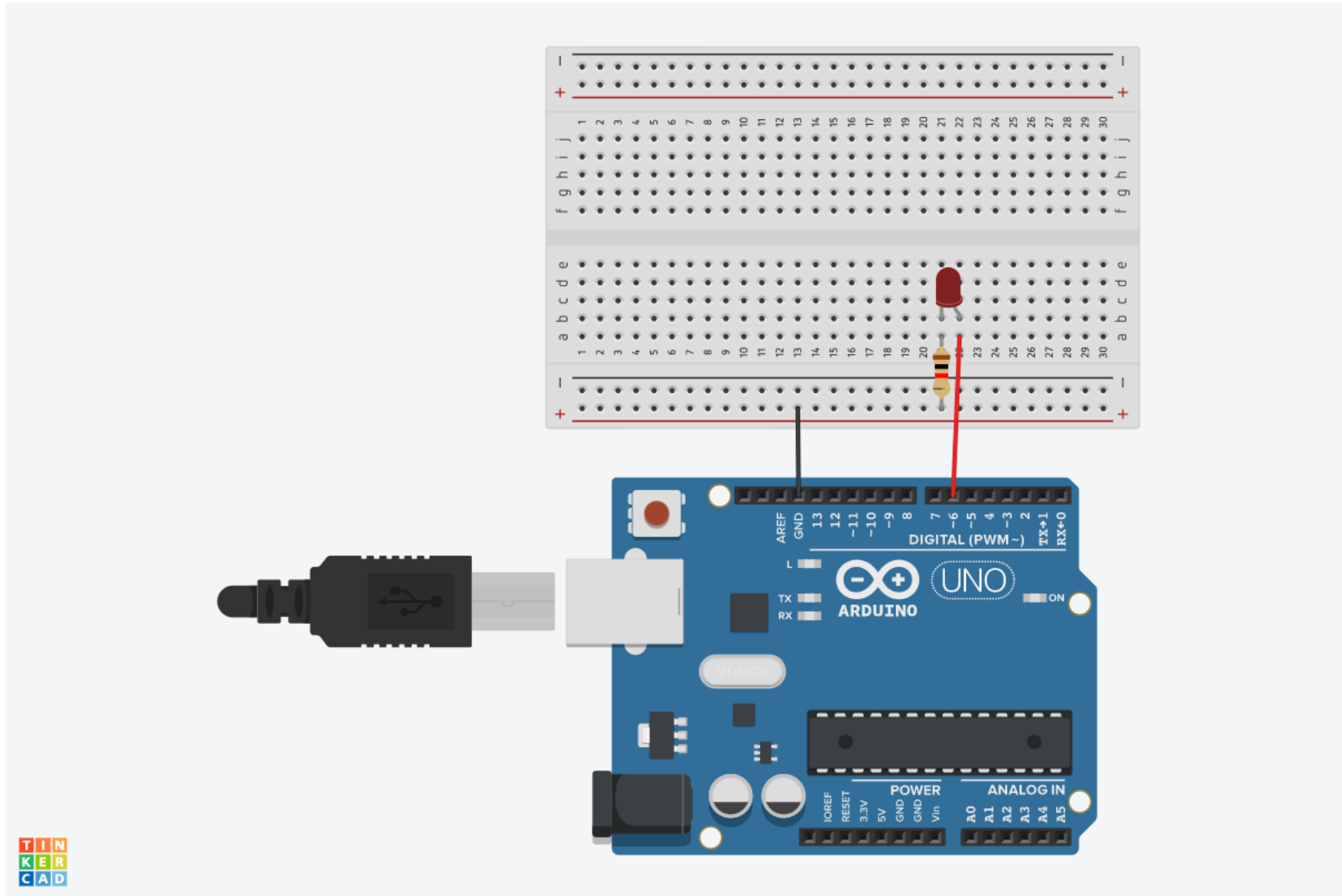
주 기능을 담당(반복할 내용을 {} 안에 선언)

아두이노 스케치가 실행되면 매번 반복

반복할 센서 통신, 주 기능을 작성

```
1 void loop() { // 계속 반복
2     digitalWrite(4, HIGH); // 4번 핀을 켜다.
3     delay(1000); // 1초간 지속한다. (1000밀리초)
4     digitalWrite(4, LOW); // 4번 핀을 끈다.
5     delay(1000); // 1초간 지속한다.
6 }
```

▶ LED 제어하기(깜빡이)



▶ LED 제어하기(깜빡이)

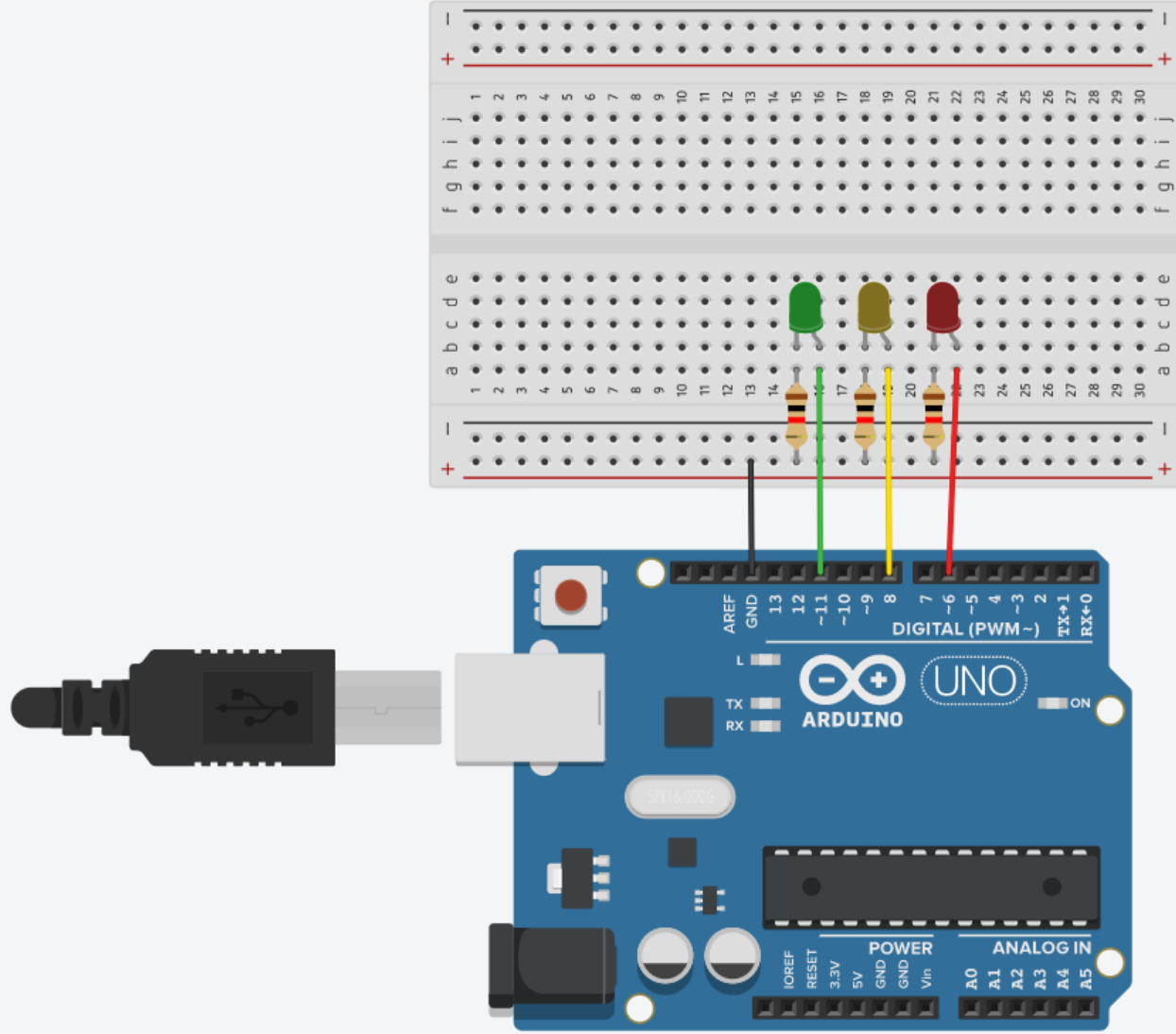
```
1 int redLED = 6; // 핀 설정
2
3 void setup() {
4     pinMode(redLED, OUTPUT); // 핀 선언
5 }
6
7 void loop() {
8     // LED ON / OFF : HIGH / LOW
9     digitalWrite(redLED, HIGH);
10    delay(1500); // 1.5초 지속
11
12    digitalWrite(redLED, LOW);
13    delay(1500); // 1.5초 지속
14 }
```


▶ 응용 (신호등 만들기)

➤ 조건

1. 빨간 LED pin : 6번 / 노란 LED pin : 8번 / 초록 LED pin : 11번
2. 노란 LED만 켜지고 3초 지속
3. 빨간 LED 켜지면 노란 LED가 꺼지고 5초 지속
4. 노란 LED 켜지고 2초 지속
5. 빨간 LED, 노란 LED 꺼지고
초록 LED가 켜진 상태에서 15초 지속
6. 초록 LED를 끄고, 1 ~ 5번 반복

▶ 응용 (신호등 만들기)

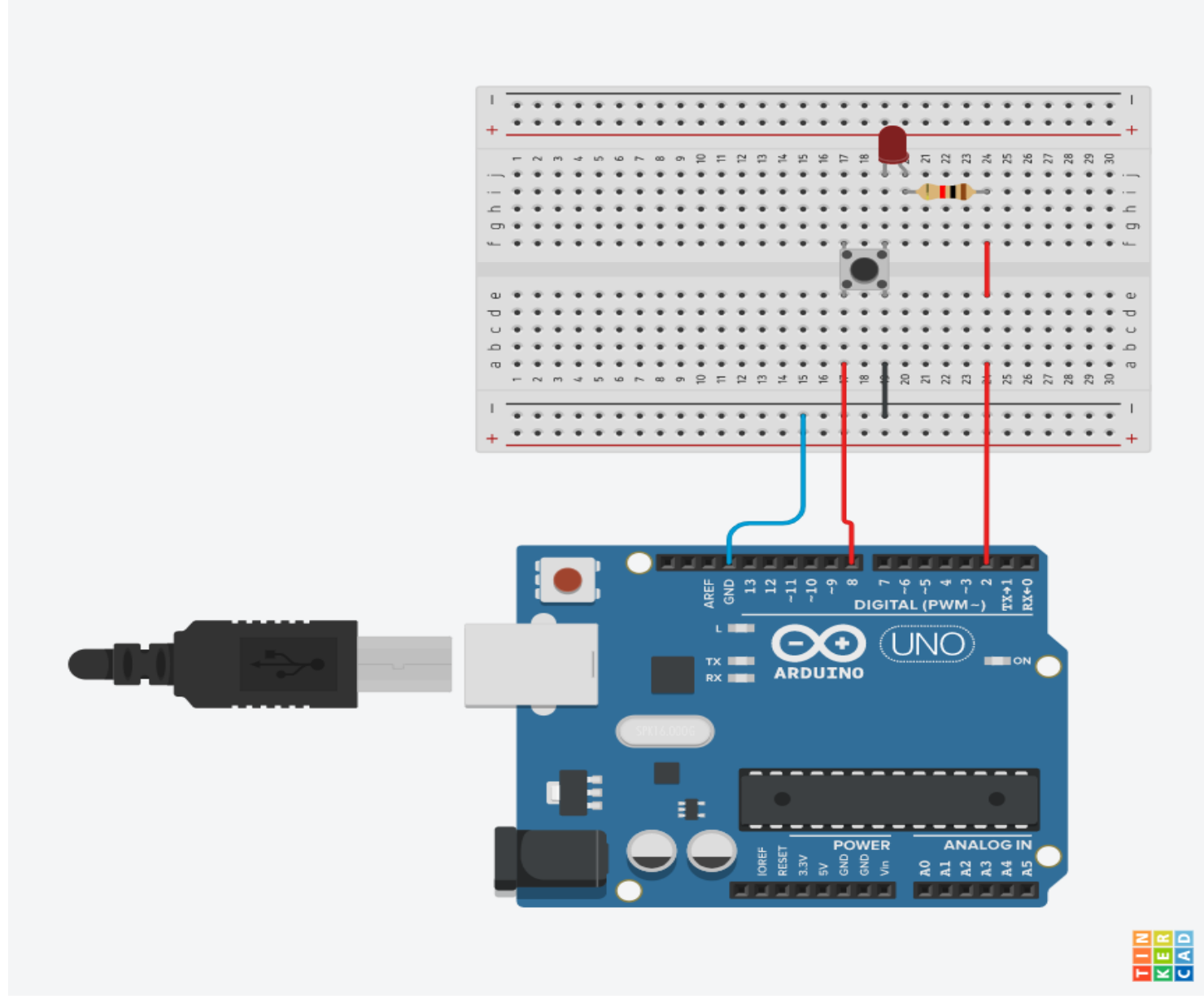


▶ 응용 (신호등 만들기)

```
1 int red = 6; // 핀 설정
2 int yellow = 8;
3 int green = 11;
4
5 void setup() {
6     pinMode(red, OUTPUT); // 핀 선언
7     pinMode(yellow, OUTPUT);
8     pinMode(green, OUTPUT);
9 }
10
11 void loop() {
12     changeLights();
13
14     delay(15000); // 각 반복마다 15초 씩 딜레이
15 }
```

```
17 void changeLights() {
18     // 초록색 LED OFF, 노란색 LED ON
19     digitalWrite(green, LOW);
20     digitalWrite(yellow, HIGH);
21     delay(3000);
22
23     // 노란색 LED OFF, 빨간색 LED ON
24     digitalWrite(yellow, LOW);
25     digitalWrite(red, HIGH);
26     delay(5000);
27
28     // 노란색 LED ON
29     digitalWrite(yellow, HIGH);
30     delay(2000);
31
32     // 노란색 LED OFF, 빨간색 LED OFF
33     digitalWrite(yellow, LOW);
34     digitalWrite(red, LOW);
35     // 초록색 LED ON
36     digitalWrite(green, HIGH);
37 }
```

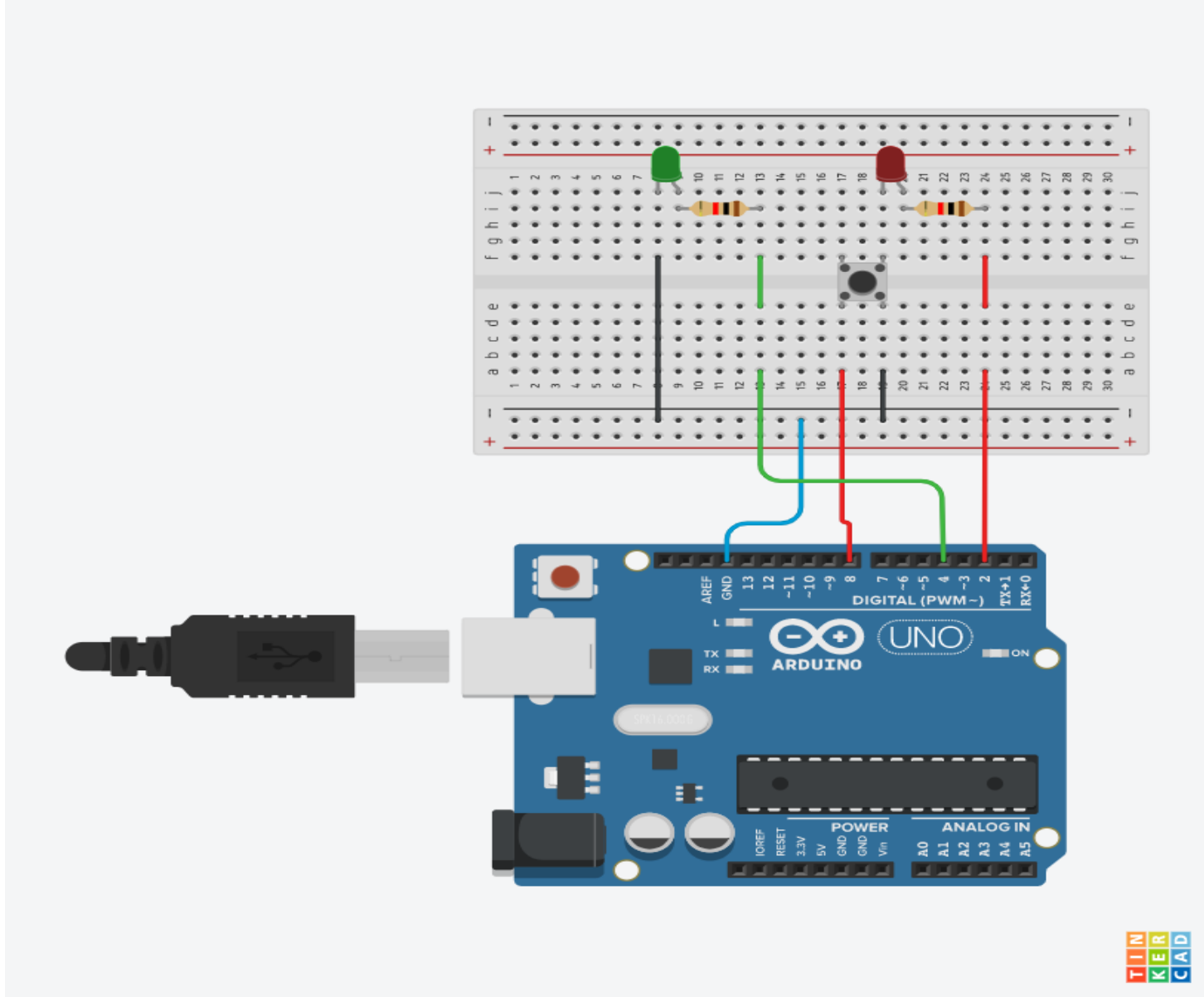
▶ 버튼 클릭



▶ 버튼 클릭

```
1 int redLED = 2; // LED 핀 설정
2 int sw = 8;     // 스위치 핀 설정
3
4 void setup() {
5     pinMode(redLED, OUTPUT); // LED 핀 연결
6     pinMode(sw, INPUT_PULLUP); // 스위치 핀 연결
7     // ** INPUT_PULLUP 으로 눌렀을 때 INPUT이 되도록 연결한다.
8 }
9
10 void loop() {
11     // digitalWrite(핀) 로 사용자가 누른 스위치의 값을 가져올 수 있다.
12     // 스위치가 눌리면 'LOW' / 스위치가 떨어지면 'HIGH'
13
14     if(digitalRead(sw) == LOW){ // 스위치를 눌렀다면
15         digitalWrite(redLED, HIGH); // LED 불 켜기
16     } else { // 스위치에서 손을 떼었다면
17         digitalWrite(redLED, LOW); // LED 불 끄기
18     }
19 }
```

▶ 버튼 응용(초록불 / 빨간불)



▶ 버튼 응용(초록불 / 빨간불)

```
1 int redLED = 2; // LED 핀 설정
2 int greenLED = 4; // LED 핀 설정
3 int sw = 8;      // 스위치 핀 설정
4
5 void setup() {
6   pinMode(redLED, OUTPUT); // LED 핀 연결
7   pinMode(greenLED, OUTPUT);
8   pinMode(sw, INPUT_PULLUP); // 스위치 핀 연결
9   // ** INPUT_PULLUP 으로 눌렀을 때 INPUT이 되도록 연결한다.
10 }
11
12 void loop() {
13   // digitalRead(핀) 로 사용자가 누른 스위치의 값을 가져올 수 있다.
14   // 스위치가 눌리면 'LOW' / 스위치가 떨어지면 'HIGH'
15
16   if(digitalRead(sw) == LOW){ // 스위치를 눌렀다면
17     digitalWrite(greenLED, LOW);
18     digitalWrite(redLED, HIGH); // LED 불 켜기
19   } else { // 스위치에서 손을 떼었다면
20     digitalWrite(redLED, LOW); // LED 불 끄기
21     digitalWrite(greenLED, HIGH);
22   }
23 }
```