

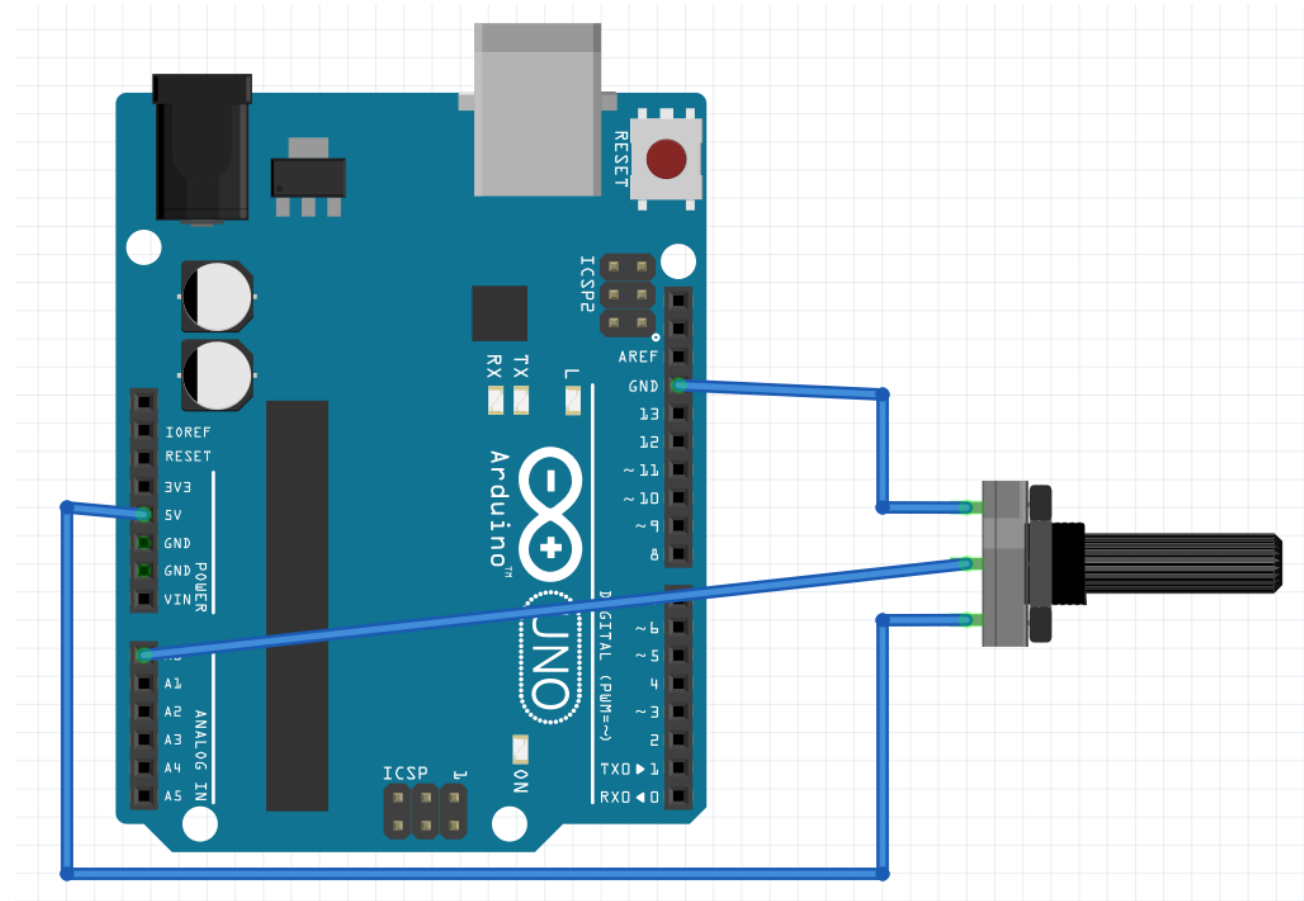
# 센서를 이용한 자동 제어

[https://github.com/juhong-rdv/2023\\_fall\\_du\\_sensor](https://github.com/juhong-rdv/2023_fall_du_sensor)



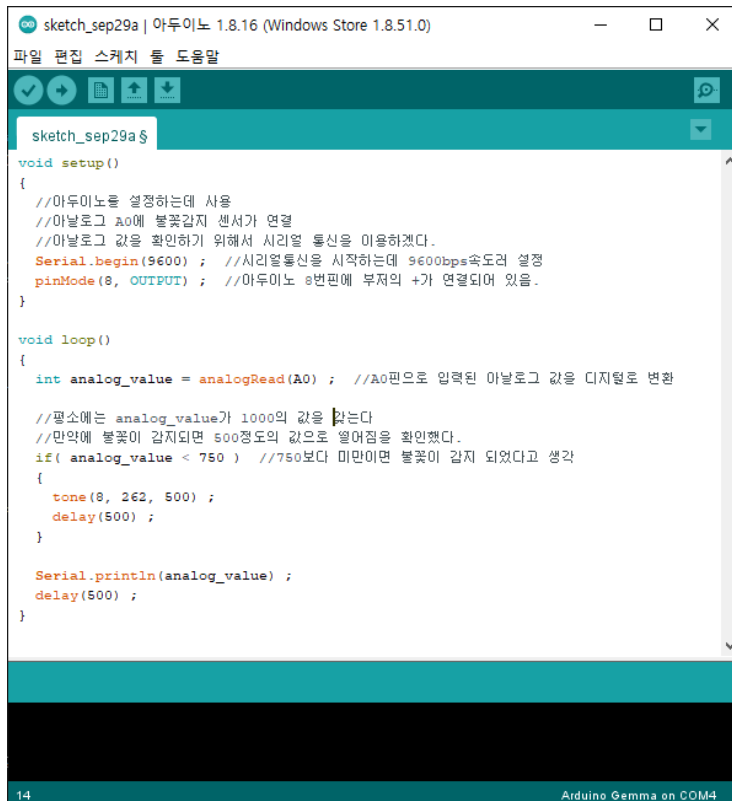
# 가변저항(Potentiometer, 볼륨)

```
void setup (){\n  Serial.begin(9600);\n}\n\nvoid loop(){\n  int val = analogRead(A0);\n  Serial.print("Analog : ");\n  Serial.println(val);\n}
```



# 아두이노 프로그래밍 과정

## • 업로드 & 실행

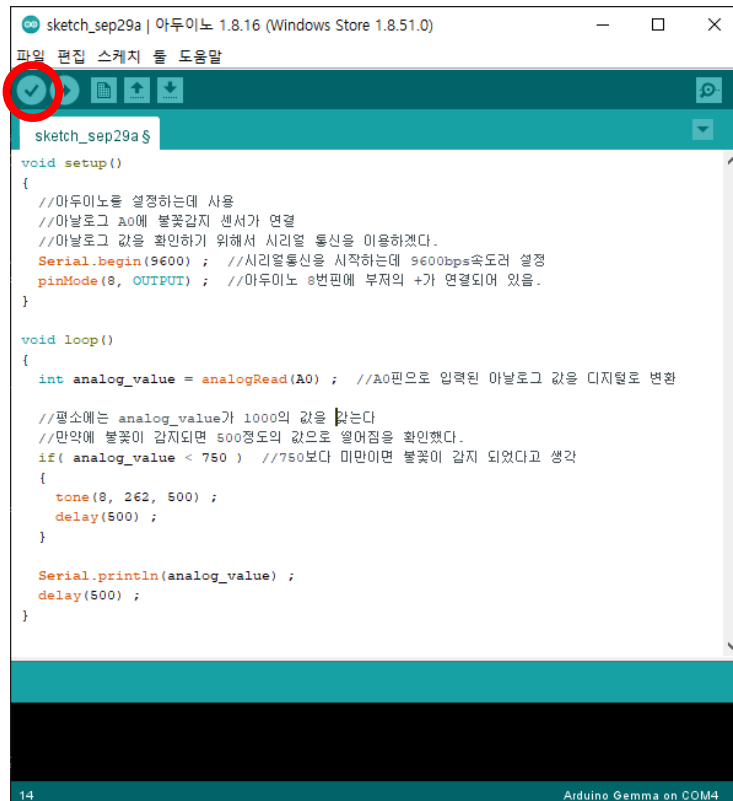


```
sketch_sep29a $
void setup()
{
  //아두이노를 설정하는데 사용
  //아날로그 A0에 불꽃감지 센서가 연결
  //아날로그 값을 확인하기 위해서 시리얼 통신을 이용하겠다.
  Serial.begin(9600); //시리얼통신을 시작하는데 9600bps속도려 설정
  pinMode(8, OUTPUT); //아두이노 8번핀에 부저의 +가 연결되어 있음.
}

void loop()
{
  int analog_value = analogRead(A0); //A0핀으로 입력된 아날로그 값을 디지털로 변환

  //평소에는 analog_value가 1000의 값을 갖는다
  //만약에 불꽃이 감지되면 500정도의 값으로 떨어짐을 확인했다.
  if( analog_value < 750 ) //750보다 미만이면 불꽃이 감지 되었다고 생각
  {
    tone(8, 262, 500);
    delay(500);
  }

  Serial.println(analog_value);
  delay(500);
}
```

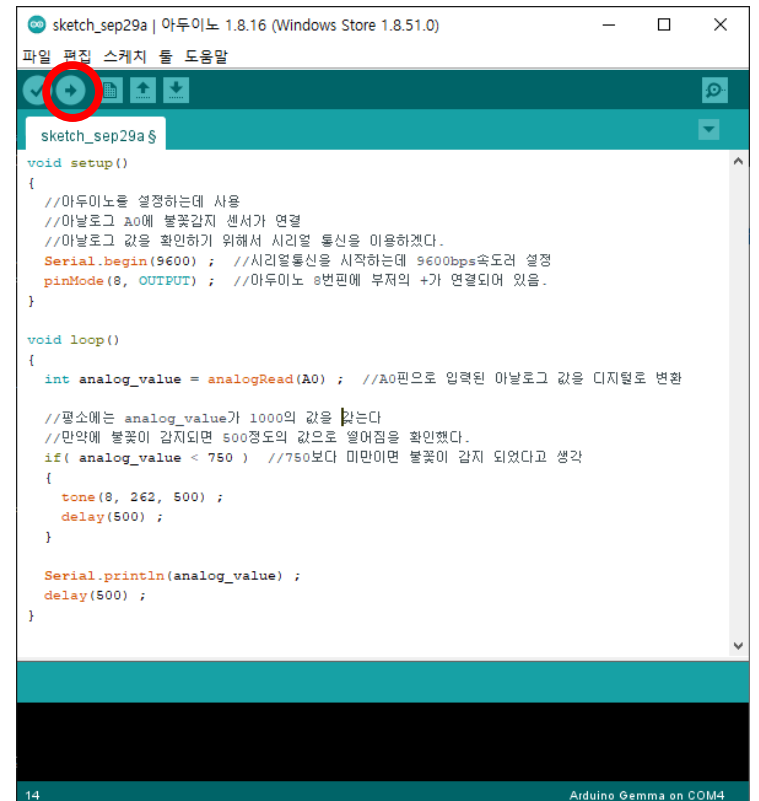


```
sketch_sep29a $
void setup()
{
  //아두이노를 설정하는데 사용
  //아날로그 A0에 불꽃감지 센서가 연결
  //아날로그 값을 확인하기 위해서 시리얼 통신을 이용하겠다.
  Serial.begin(9600); //시리얼통신을 시작하는데 9600bps속도려 설정
  pinMode(8, OUTPUT); //아두이노 8번핀에 부저의 +가 연결되어 있음.
}

void loop()
{
  int analog_value = analogRead(A0); //A0핀으로 입력된 아날로그 값을 디지털로 변환

  //평소에는 analog_value가 1000의 값을 갖는다
  //만약에 불꽃이 감지되면 500정도의 값으로 떨어짐을 확인했다.
  if( analog_value < 750 ) //750보다 미만이면 불꽃이 감지 되었다고 생각
  {
    tone(8, 262, 500);
    delay(500);
  }

  Serial.println(analog_value);
  delay(500);
}
```



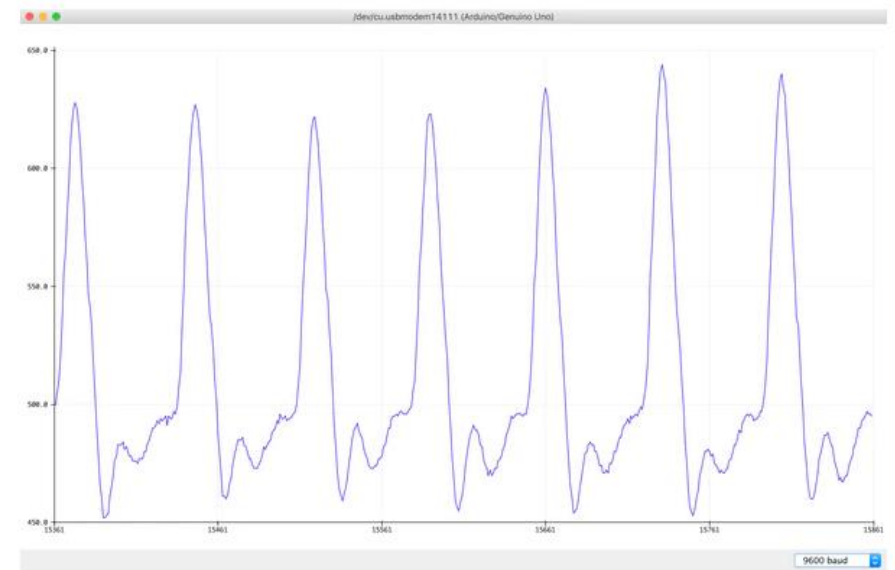
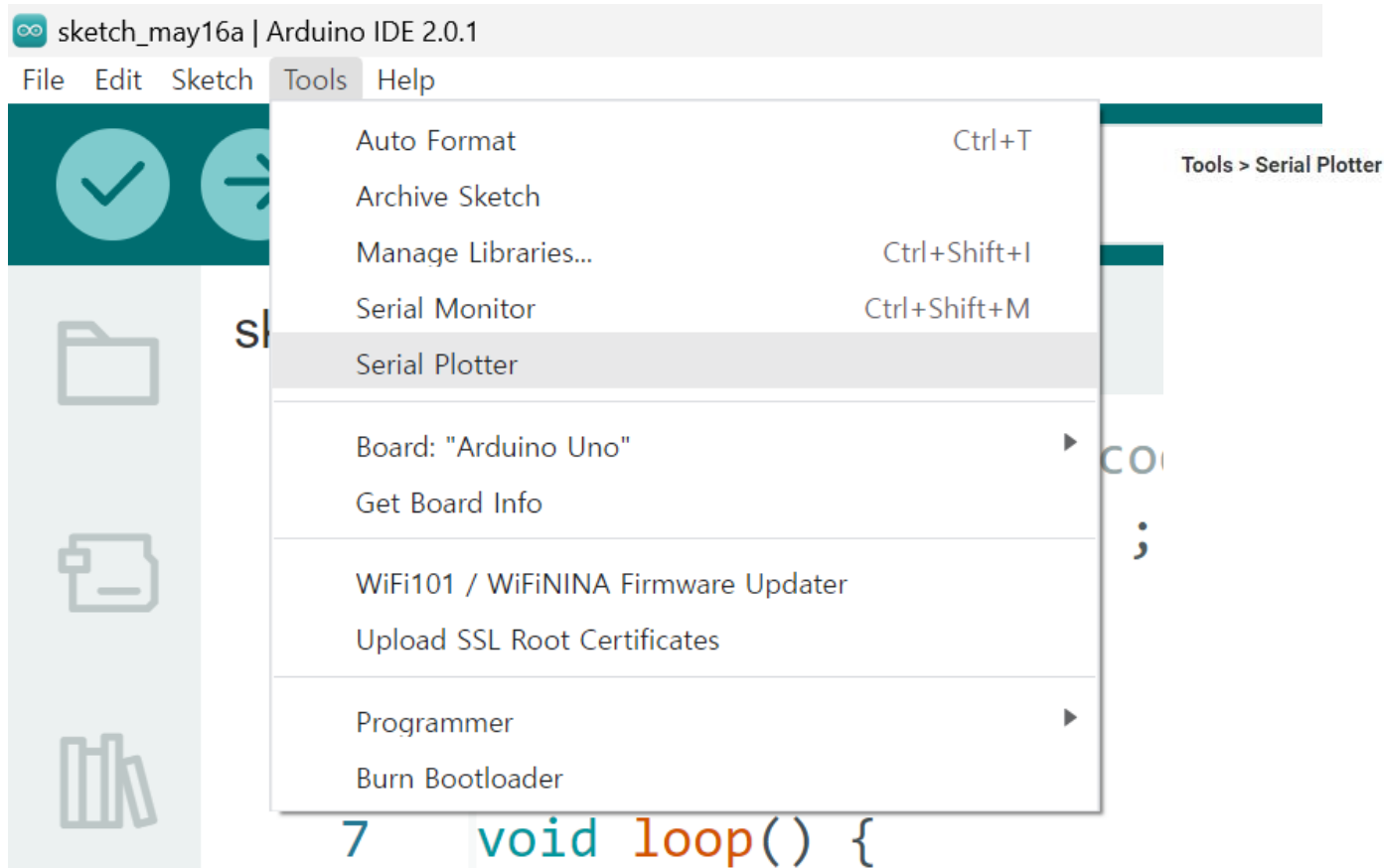
```
sketch_sep29a $
void setup()
{
  //아두이노를 설정하는데 사용
  //아날로그 A0에 불꽃감지 센서가 연결
  //아날로그 값을 확인하기 위해서 시리얼 통신을 이용하겠다.
  Serial.begin(9600); //시리얼통신을 시작하는데 9600bps속도려 설정
  pinMode(8, OUTPUT); //아두이노 8번핀에 부저의 +가 연결되어 있음.
}

void loop()
{
  int analog_value = analogRead(A0); //A0핀으로 입력된 아날로그 값을 디지털로 변환

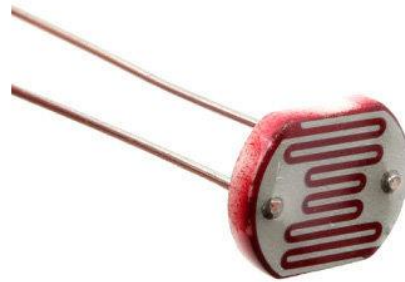
  //평소에는 analog_value가 1000의 값을 갖는다
  //만약에 불꽃이 감지되면 500정도의 값으로 떨어짐을 확인했다.
  if( analog_value < 750 ) //750보다 미만이면 불꽃이 감지 되었다고 생각
  {
    tone(8, 262, 500);
    delay(500);
  }

  Serial.println(analog_value);
  delay(500);
}
```

# 센서값을 그래프로 확인



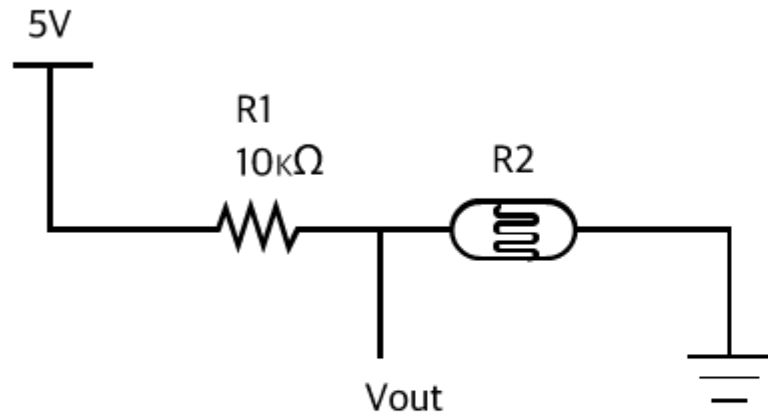
# 조도센서(CDS cell)



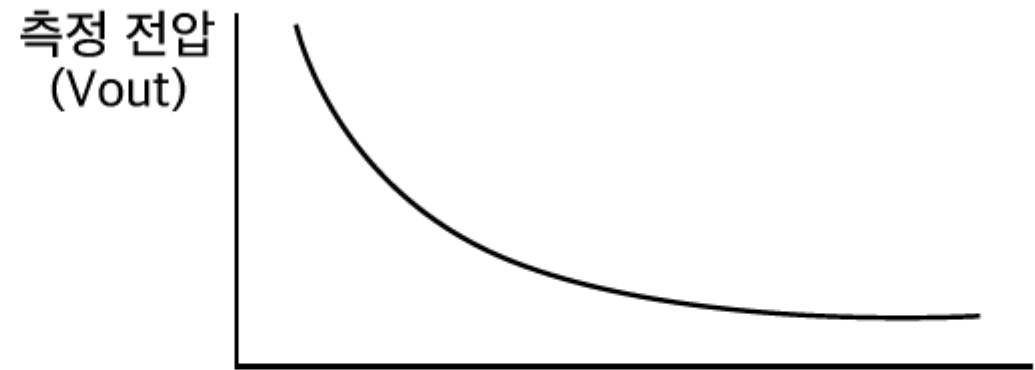
# 조도센서(CDS cell)

- 특징
  - 빛의 양(조도)에 따라 저항값이 변화(밝기값과 저항값은 반비례)
  - 극성이 없음(+,-가 없음)
  - 아날로그 입력
- 조도센서를 이용하여
  - 스마트 가로등 : 가로등을 자동으로 On/Off

# 조도 센서(CDS cell)



풀업 저항 사용

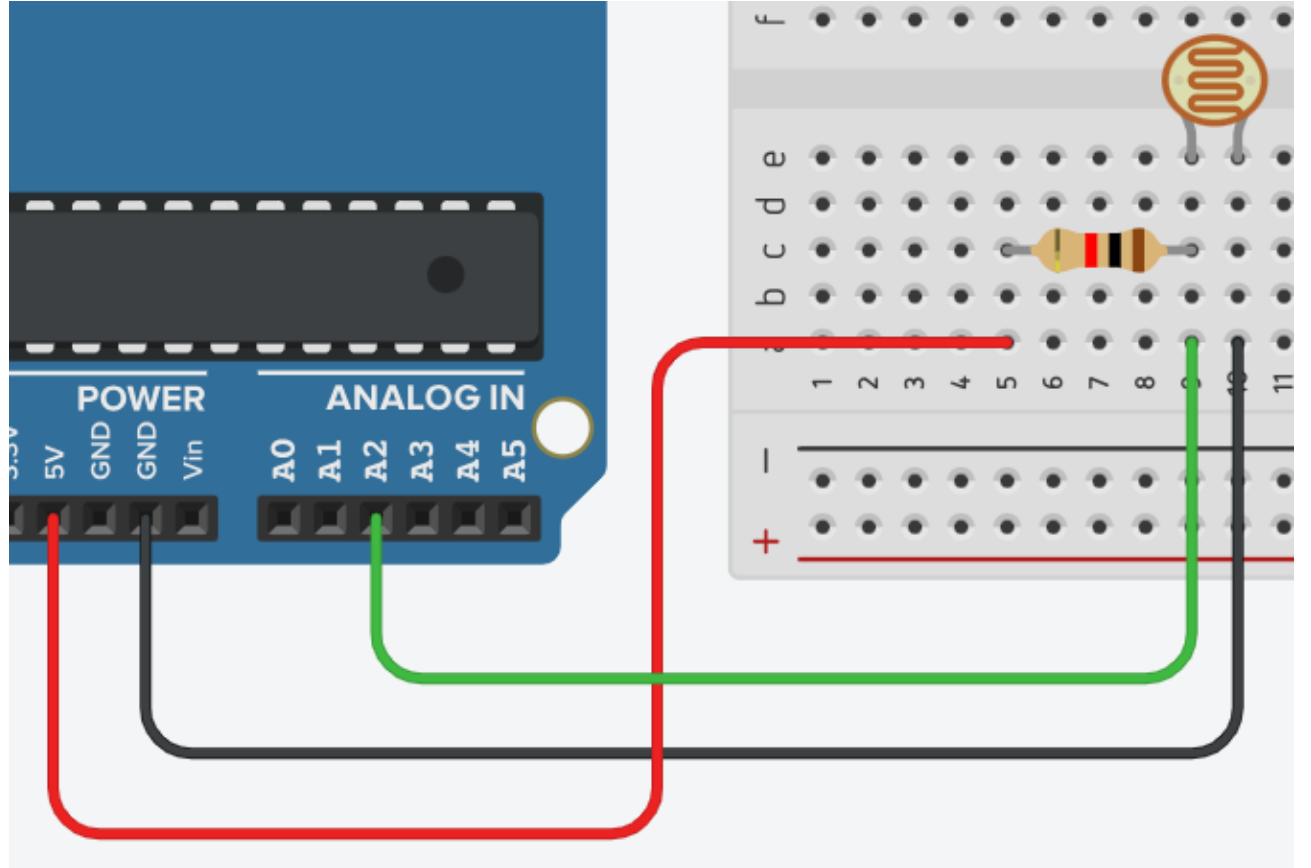


풀업 저항 사용시 밝기에 대한 측정 전압

# C D S (포토레지스터)

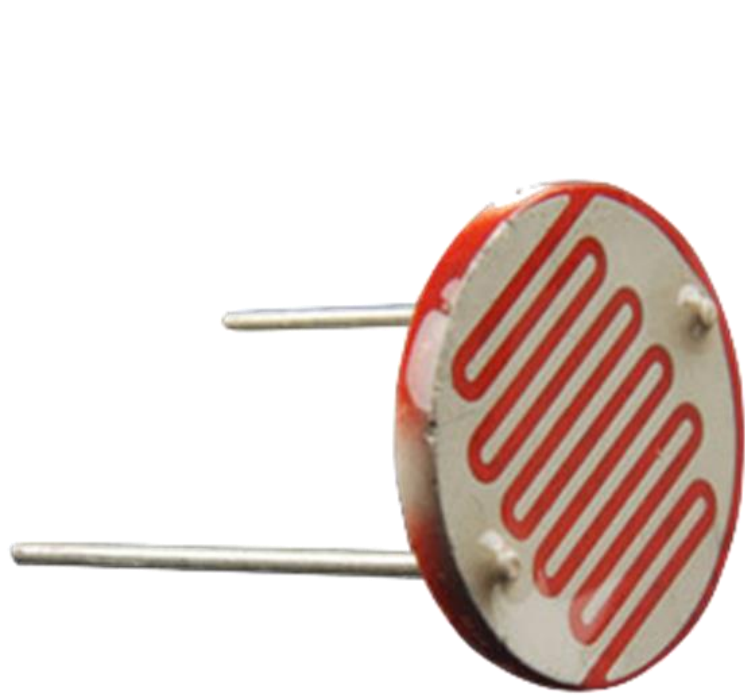
```
void setup(){
    Serial.begin(9600);
}

void loop(){
    int val = analogRead(A2);
    Serial.print("Analog : ");
    Serial.println(val);
}
```

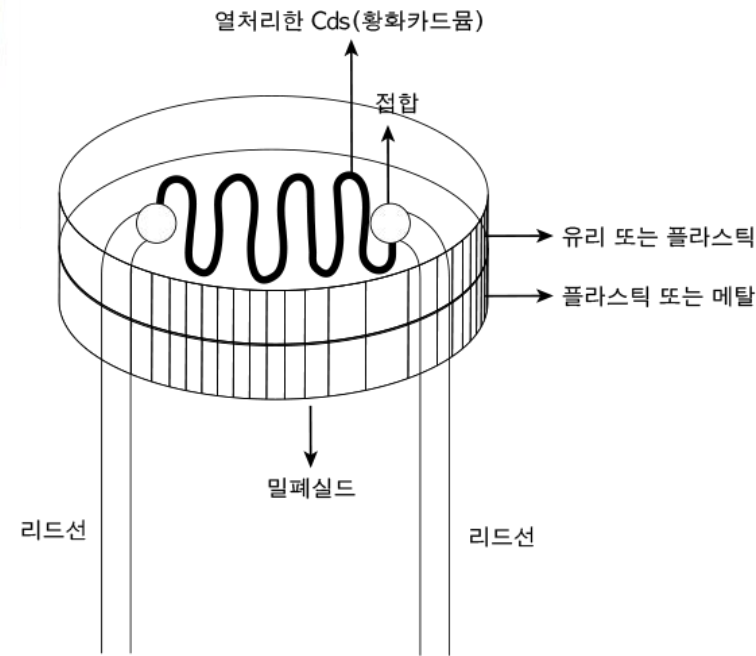




# 스마트 가로등 제작



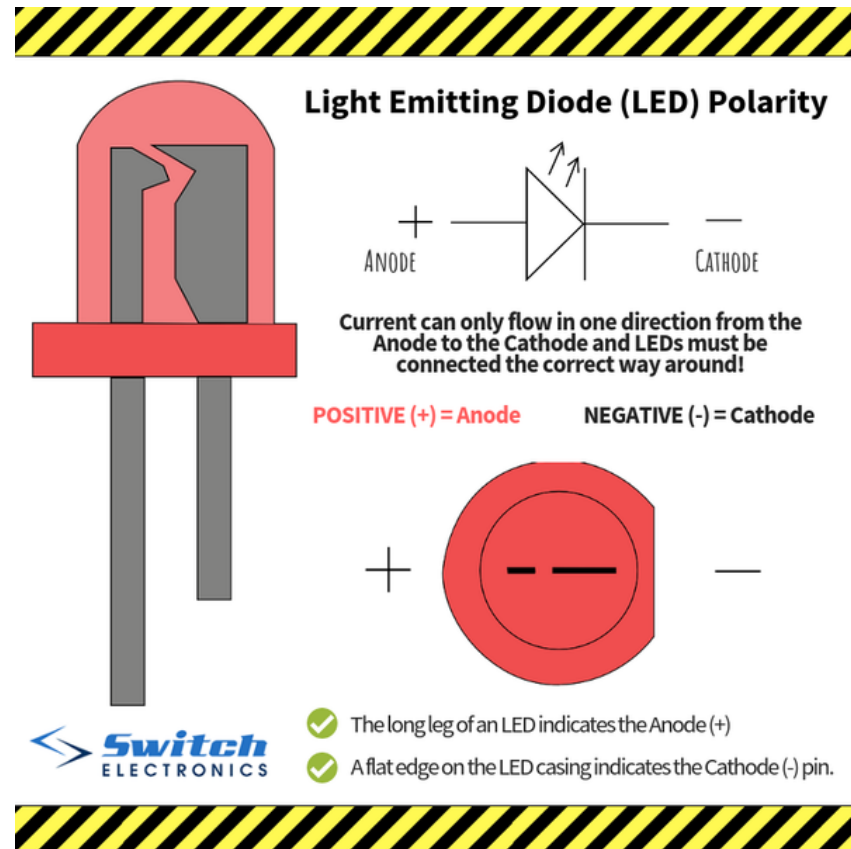
조도센서(Cds)



조도센서(Cds) 구조도

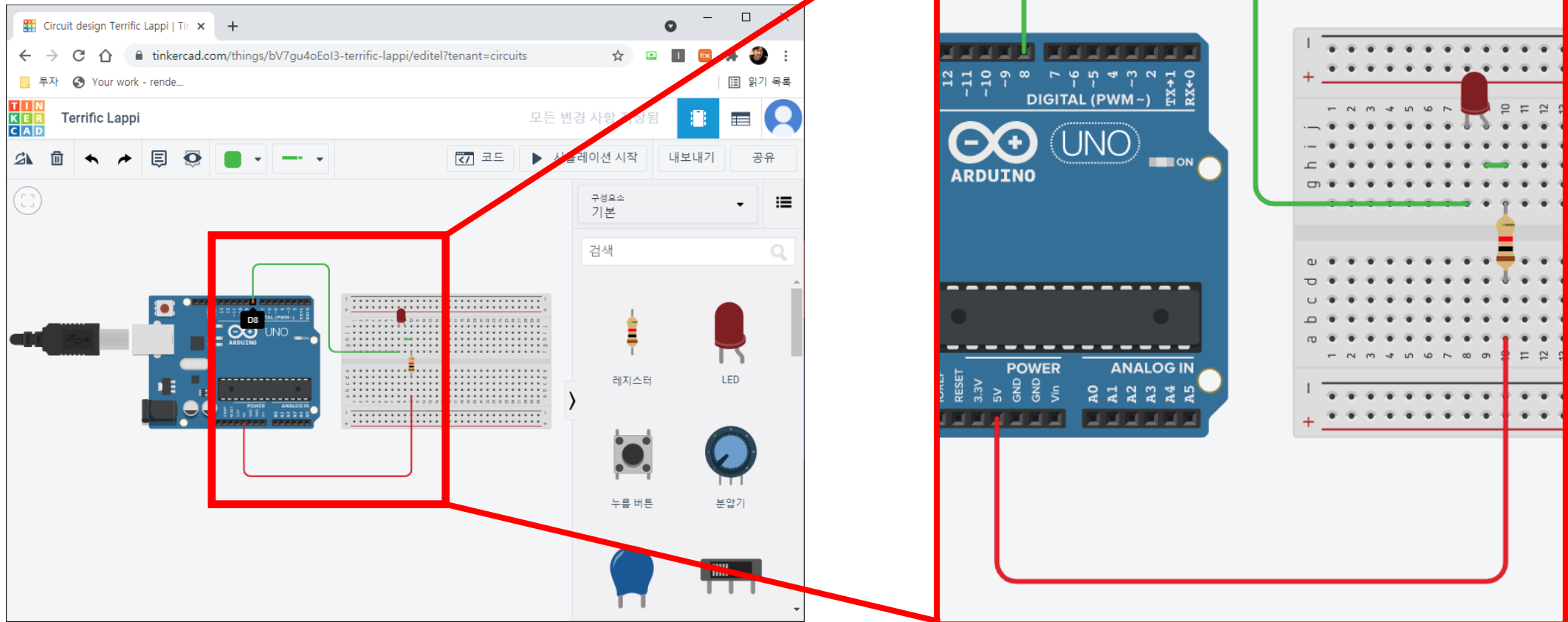
# 아두이노를 이용한 LED 실험

- LED를 이용한 포트 Output 테스트



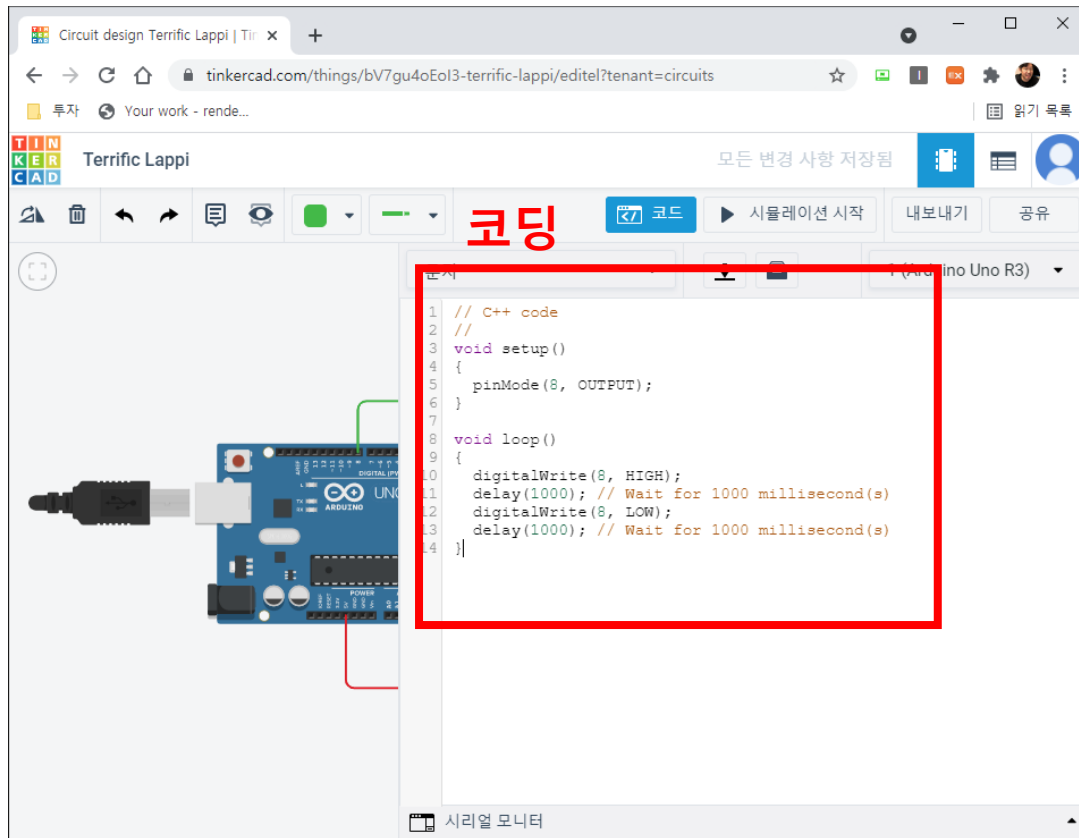
# 아두이노를 이용한 LED 실험

- tinkercad.com : 전체 회로 구성



# 아두이노를 이용한 LED 실험

- tinkercad.com : 코드 작성

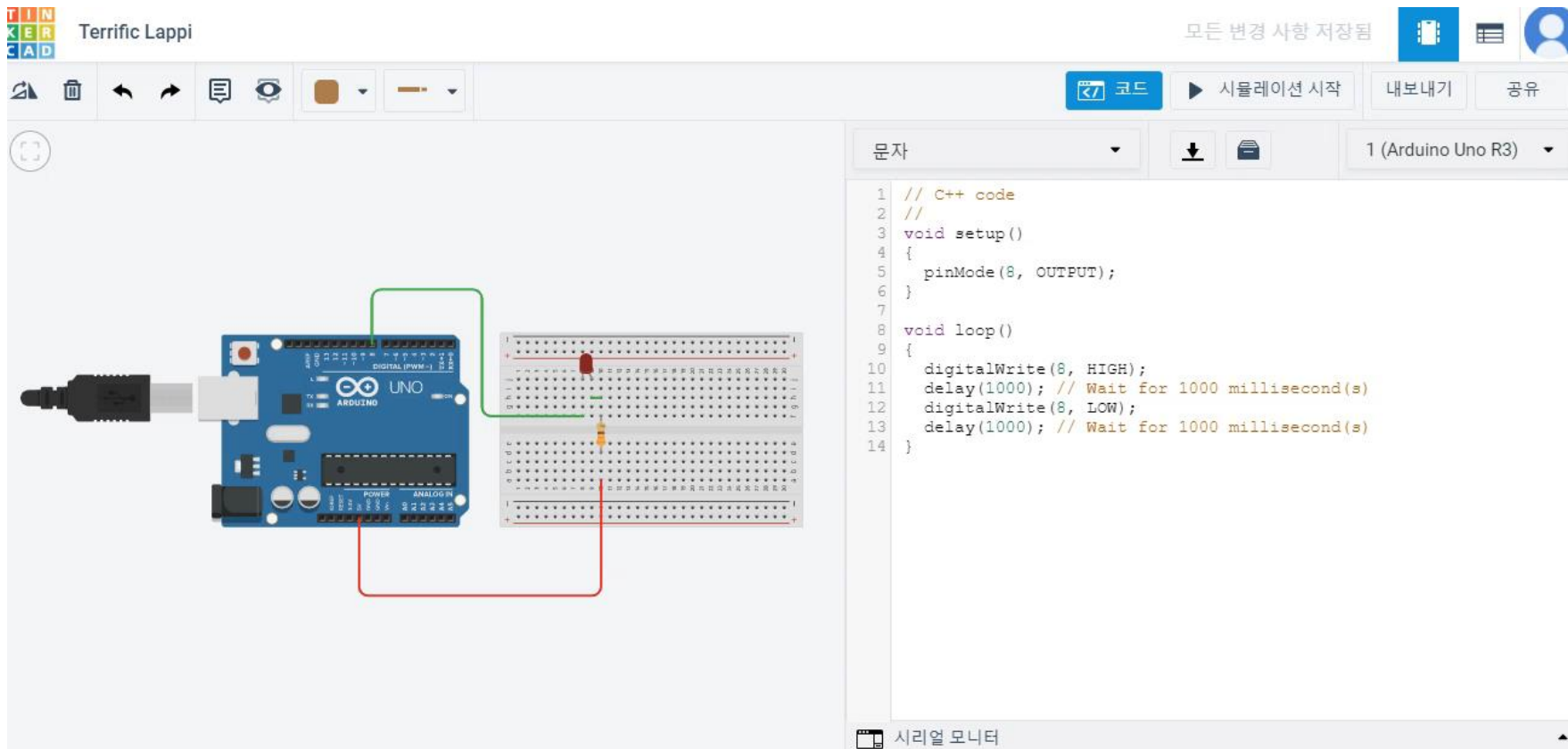


```
// C++ code
//
void setup()
{
  pinMode(8, OUTPUT);
}

void loop()
{
  digitalWrite(8, HIGH);
  delay(1000); // Wait for 1000 millisecond(s)
  digitalWrite(8, LOW);
  delay(1000); // Wait for 1000 millisecond(s)
}
```

# 아두이노를 이용한 LED 실험

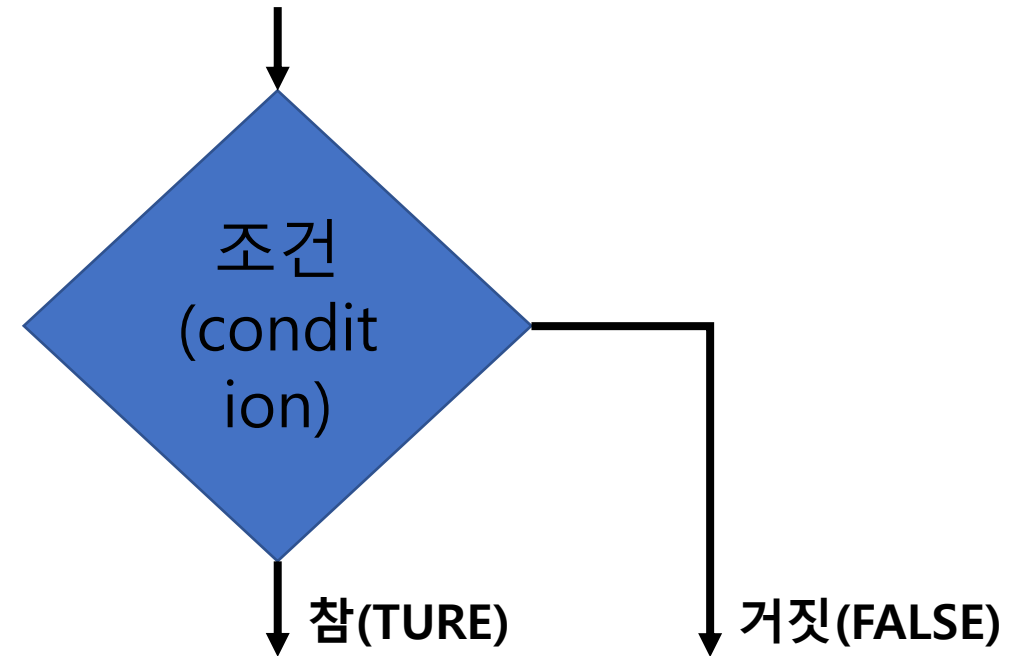
- tinkercad.com : 시뮬레이션 시작



# 특정 조건일때 실행(조건문/분기문)

- if ~ else

```
if( 조건식1 )  
{  
    //조건식1이 참이면 실행  
}  
else  
{  
    //그렇지 않으면(조건식1이 거짓이면) 실행  
}
```



# 특정 조건일때 실행(조건문/분기문)

- if ~ else

```
if( 조건식1 ) •  
{  
    //조건식1이 참이면 실행  
}  
else  
{  
    //그렇지 않으면(조건식1이 거짓이면) 실행  
}
```

## 조건식

- $A == B$  : A와 B가 같으면 참
- $A != B$  : A와 B가 다르면 참
- $A > B$  : A가 B보다 크면 참
- $A < B$  : A가 B보다 작으면 참
- $A \geq B$  : A가 B보다 크거나 같으면 참
- $A \leq B$  : A가 B보다 작거나 같으면 참

# 센서값이 특정 조건일때 실행

- if ~ else

```
if( 센서값 < 1000 )  
{  
    //불(LED)를 켜라  
}  
else  
{  
    //불(LED)를 꺼라  
}
```

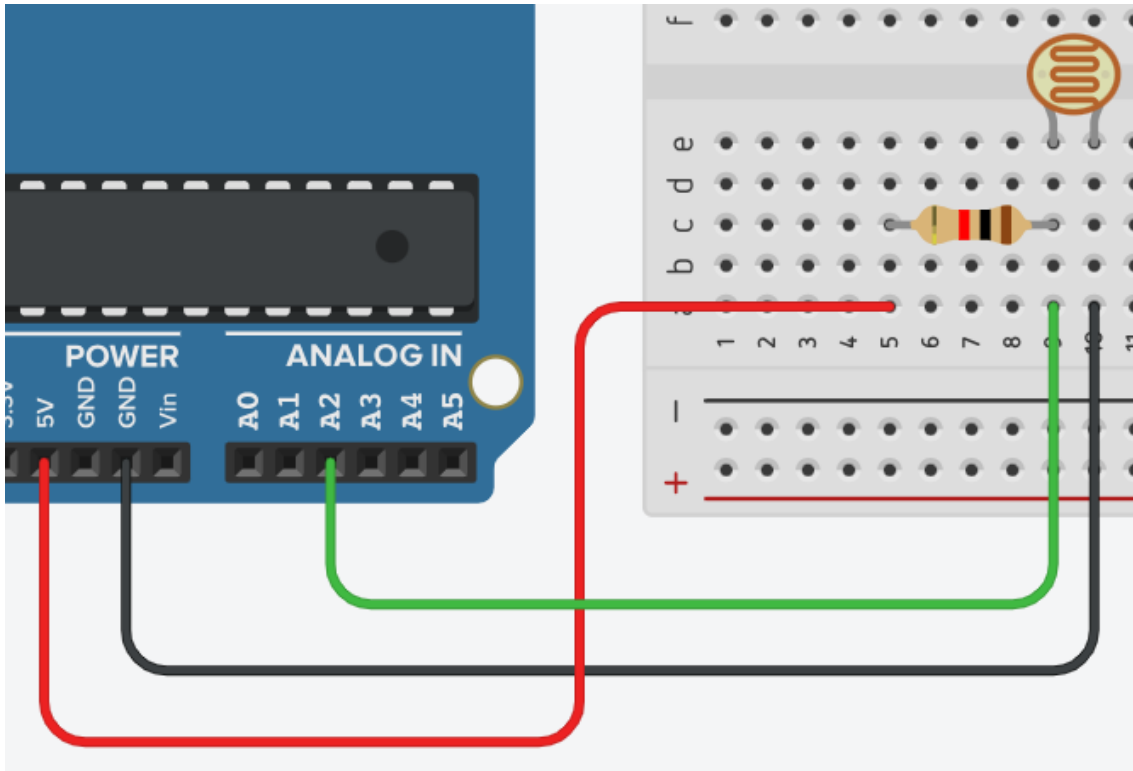
## 조건식

- $A == B$  : A와 B가 같으면 참
- $A != B$  : A와 B가 다르면 참
- $A > B$  : A가 B보다 크면 참
- $A < B$  : A가 B보다 작으면 참
- $A \geq B$  : A가 B보다 크거나 같으면 참
- $A \leq B$  : A가 B보다 작거나 같으면 참



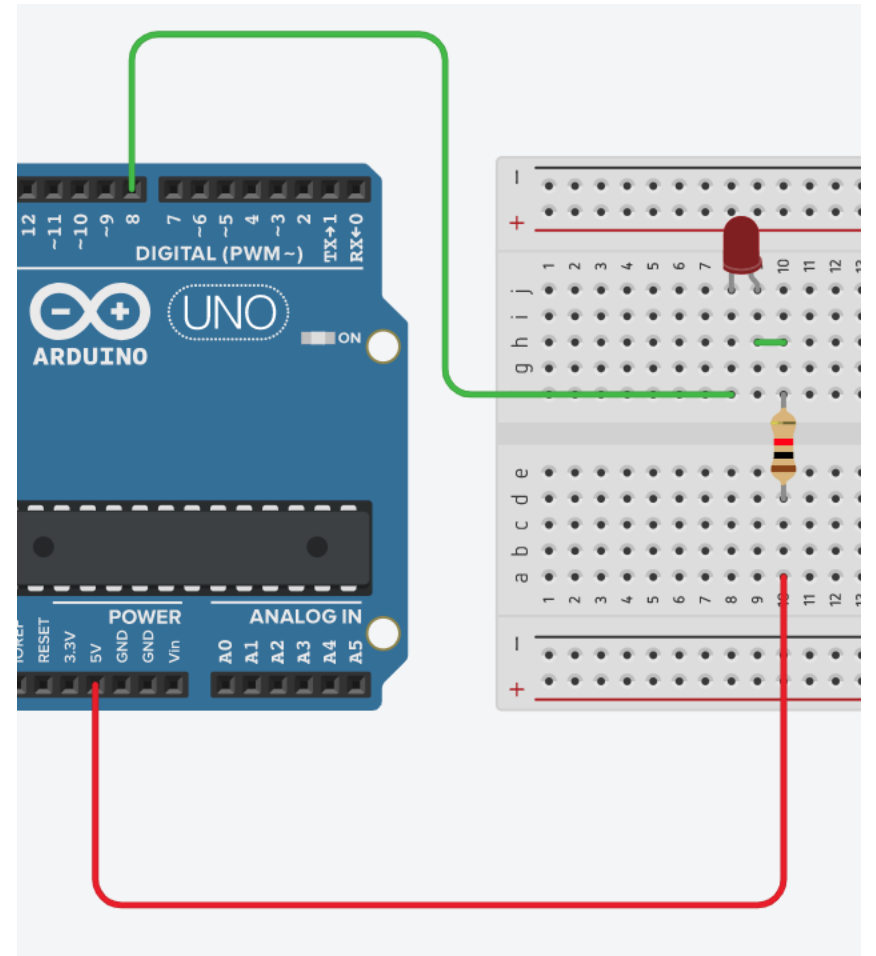
# 스마트 가로등 실험

## 1. CSD 센서 회로(빛의 세기 측정)

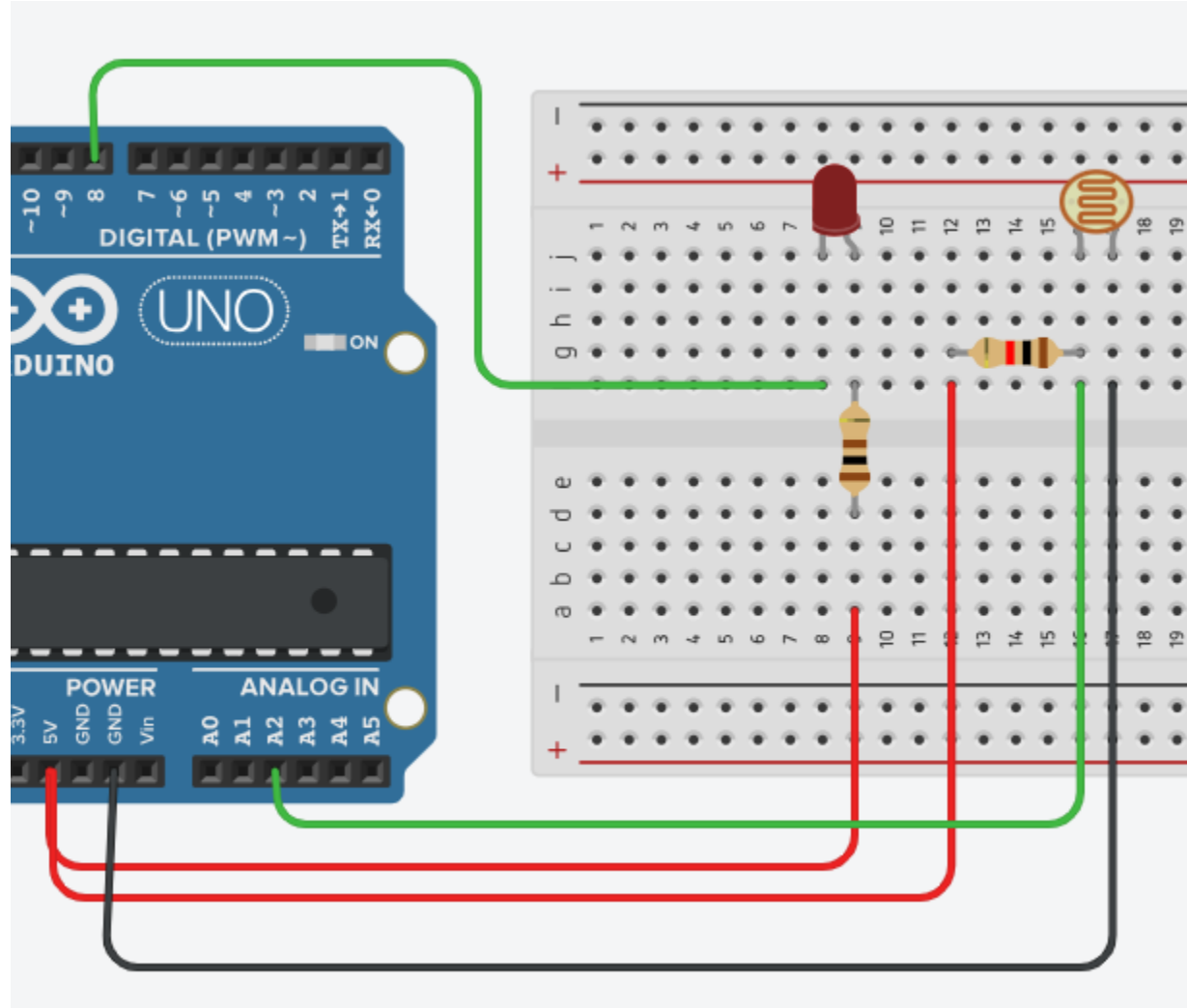


+

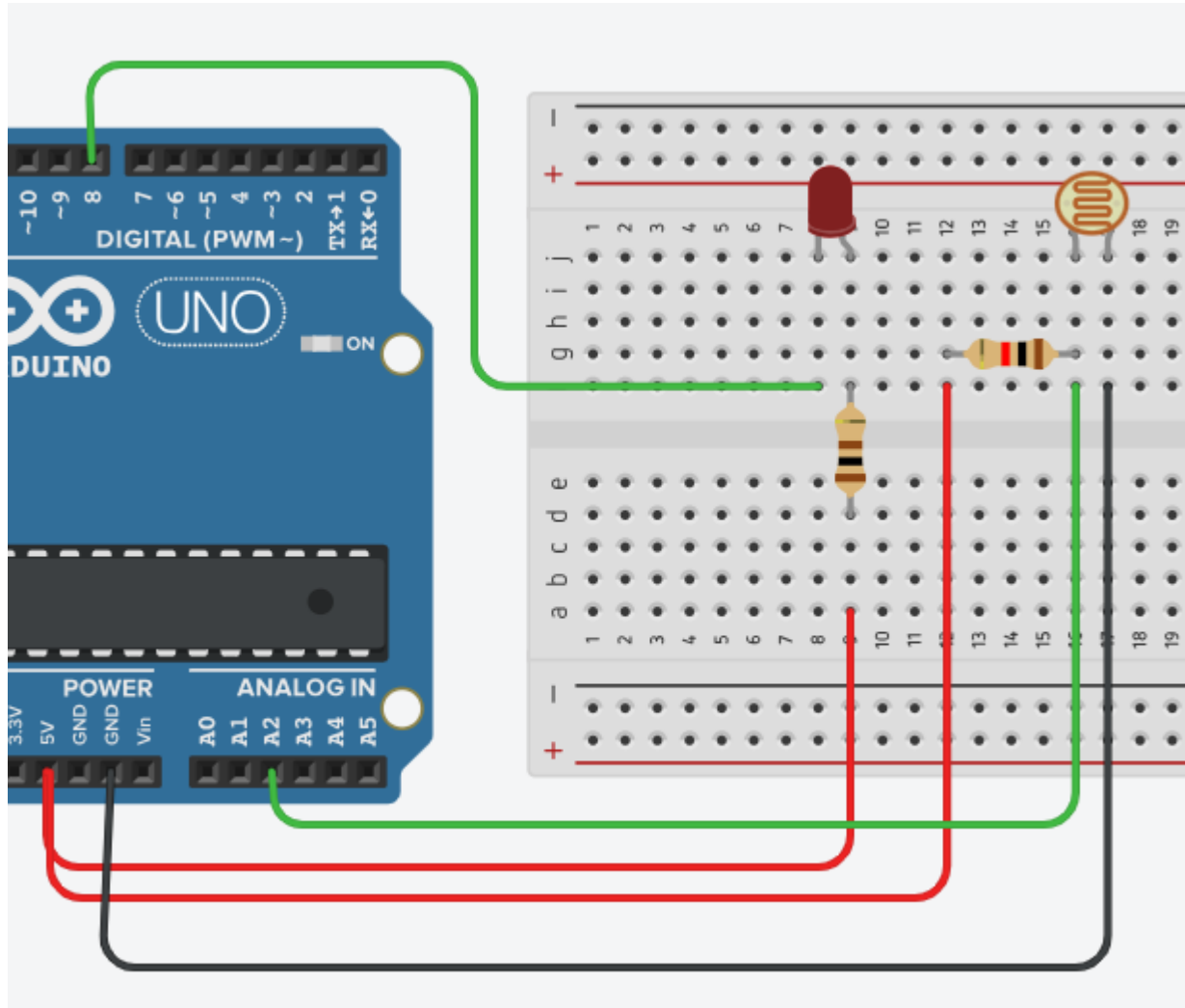
## 2. LED 제어 회로(조명 On/Off)



# 스마트 가로등 실험



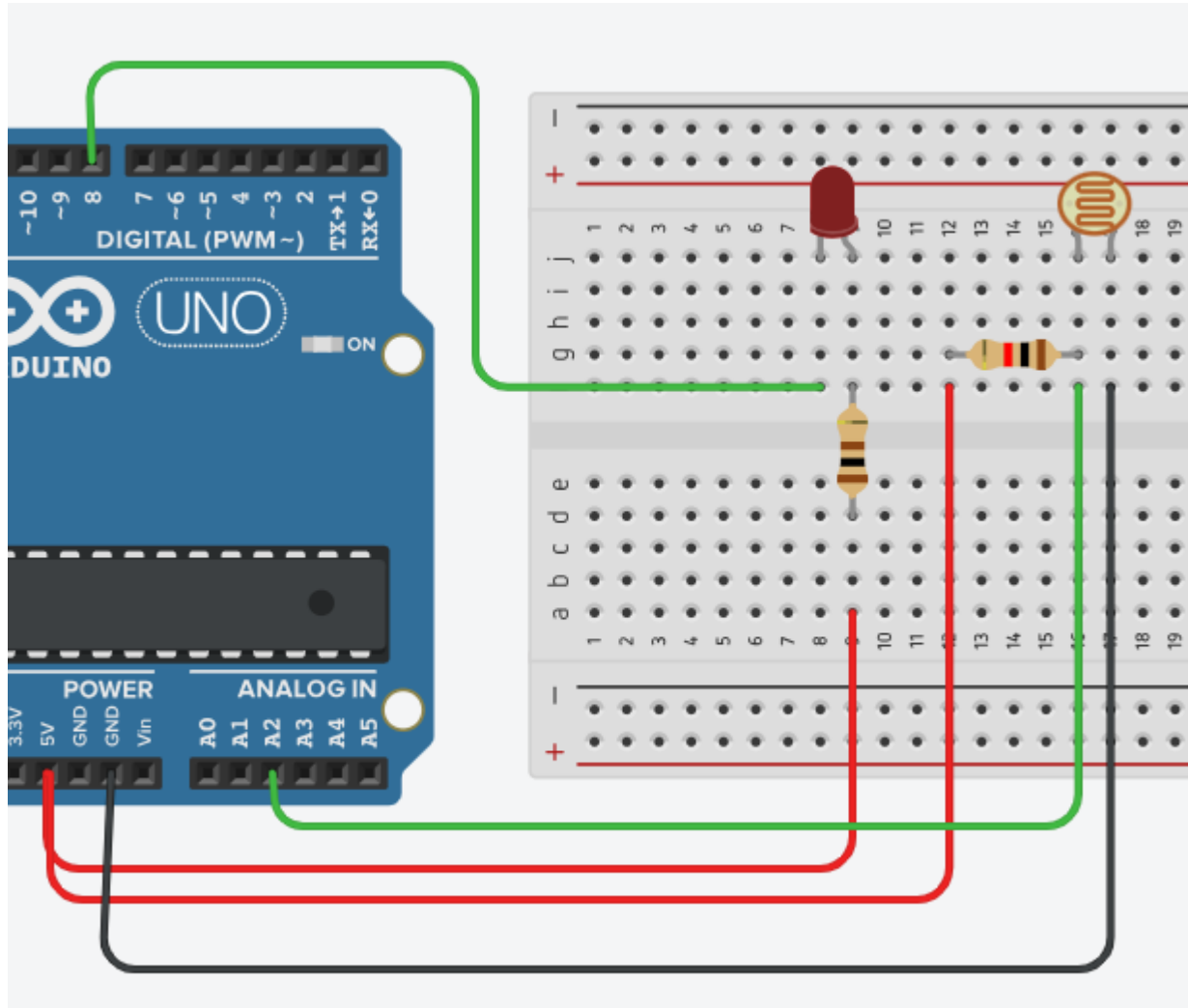
# 스마트 가로등 실험



1. 빛의 밝기에 따라 CDS 센서의 값을 확인

```
void setup (){\n  Serial.begin(9600);\n}\n\nvoid loop(){\n  int val = analogRead(A2);\n  Serial.println(val);\n}
```

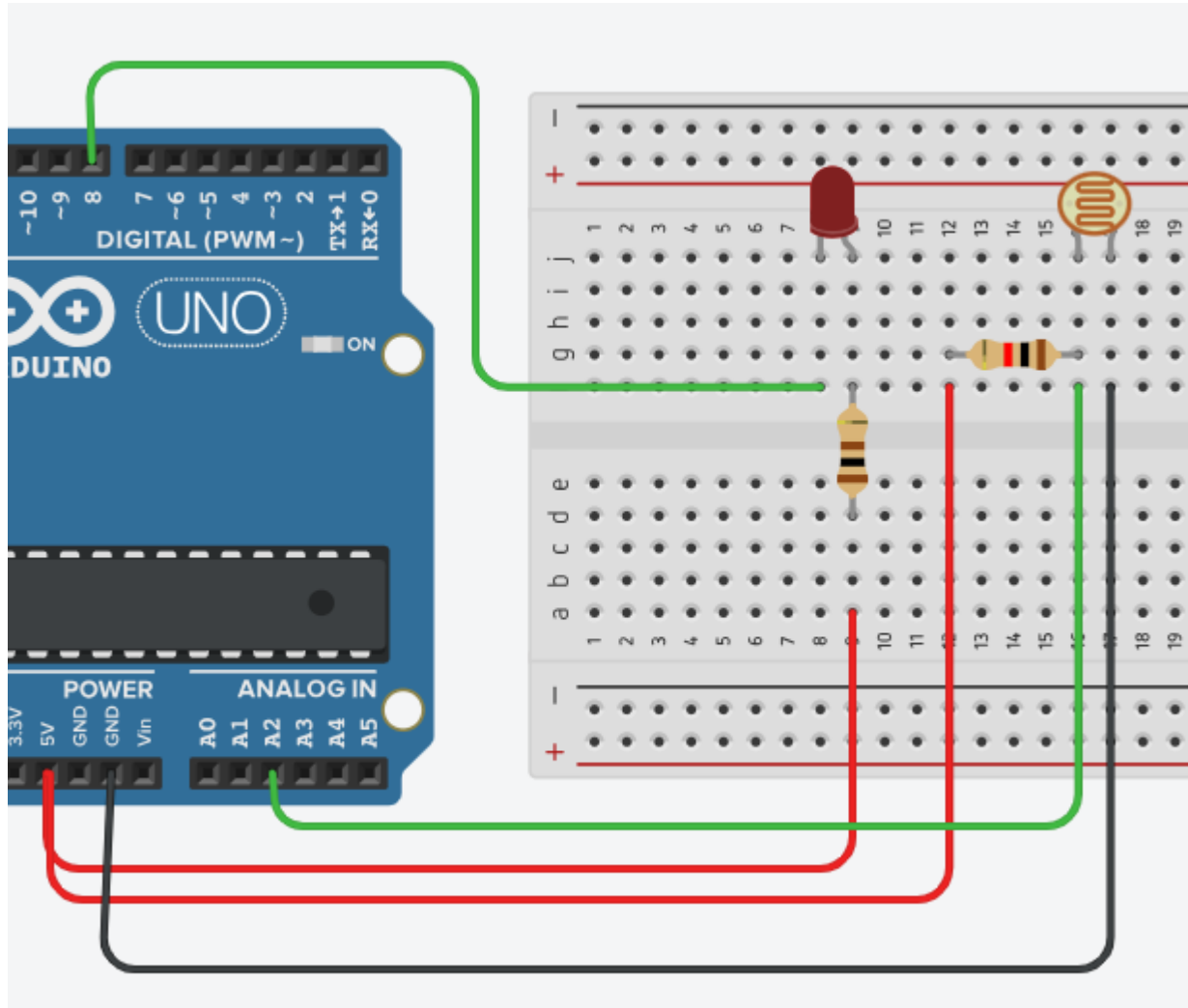
# 스마트 가로등 실험



## 2. LED를 On/Off 제어

```
void setup (){  
  Serial.begin(9600);  
}  
  
void loop(){  
  int val = analogRead(A2);  
  Serial.println(val);  
  
  //LED On/Off  
  digitalWrite(8, HIGH);  //Off  
  delay(1000);  
  digitalWrite(8, LOW);   //On  
  delay(1000);  
}
```

# 스마트 가로등 실험

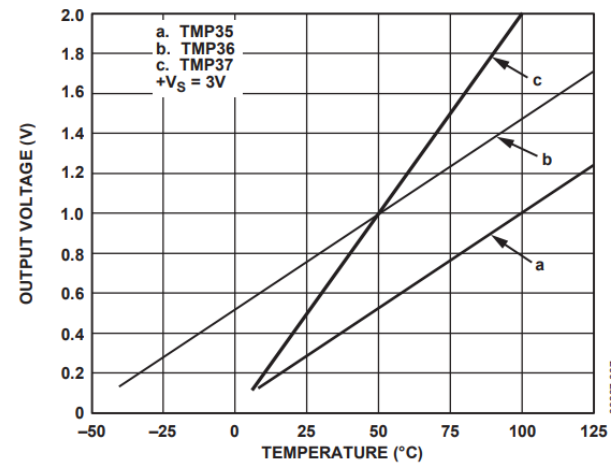


3. CDS로 빛이 어두운 경우 LED를 On  
그렇지 않으면 Off 자동 제어

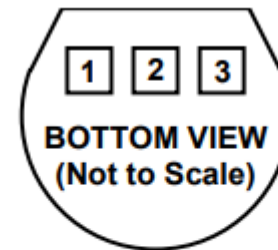
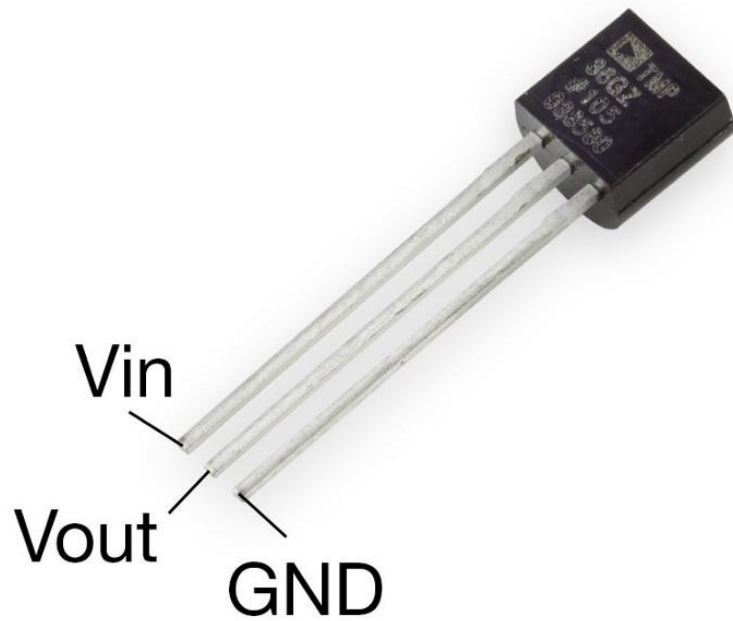
```
void setup () {  
  Serial.begin(9600);  
}  
  
void loop() {  
  int val = analogRead(A2);  
  Serial.println(val);  
  
  if( val > [      ] )  
  {  
    digitalWrite(8, [      ] );  
  }  
  else  
  {  
    digitalWrite(8, [      ] );  
  }  
}
```

# TMP36

- 온도센서는 온도를 감지해 전기신호로 바꿔주는 센서를 의미
- TMP36
  - 상온에서 대략 750mV를 출력
  - 온도 1 °C가 변화하면 10mV의 출력 전압이 변화 함
  - 정밀도는  $\pm 1$  °C로 정밀한 온도 감지는 어려움.
  - 사용하기 쉽고 저렴하여 정밀한 온도 감지가 필요 없는 어플리케이션이 많이 사용 됨.



# TMP36 핀 연결



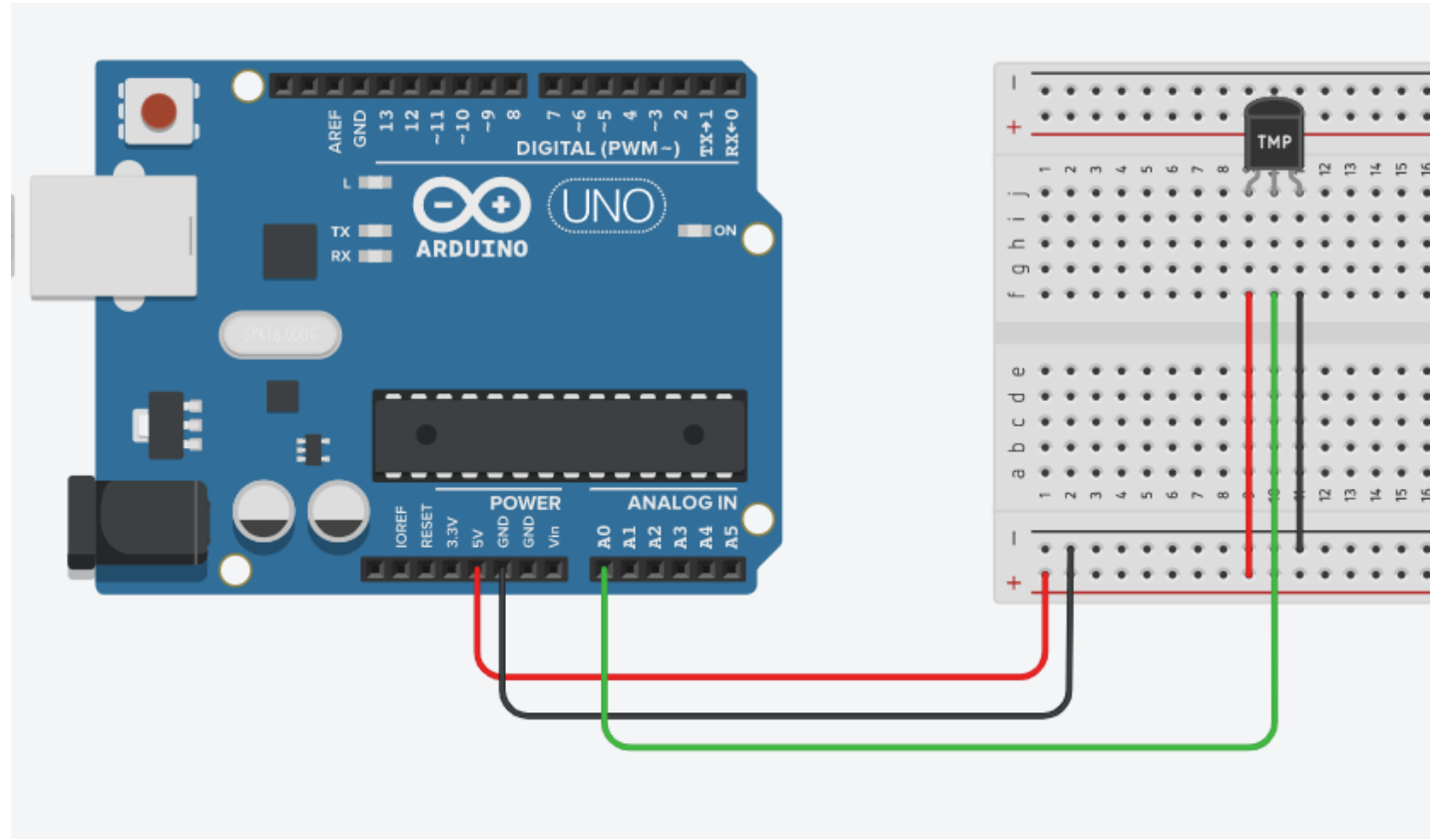
**PIN 1,  $+V_S$ ; PIN 2,  $V_{OUT}$ ; PIN 3, GND**

00337-004

Figure 4. T-3 (TO-92)

# TMP36 + 아두이노 실험

- TMP Vin <> 아두이노 5V
- TMP Vout <> 아두이노 A0
- TMP GND <> 아두이노 GND





# 코드 작성

```
void setup()
```

```
{  
  Serial.begin(9600);  
}
```

```
void loop()
```

```
{  
  int reading = analogRead(A0);  
  
  float voltage = reading * 5.0;  
  voltage /= 1024.0;  
  
  Serial.print(voltage); Serial.println(" volts");  
  
  float temperatureC = (voltage - 0.5) * 100 ;  
  Serial.print(temperatureC); Serial.println(" degrees C");  
  
  float temperatureF = (temperatureC * 9.0 / 5.0) + 32.0;  
  Serial.print(temperatureF); Serial.println(" degrees F");  
  
  delay(1000);  
}
```

The screenshot displays an Arduino IDE interface. At the top, a blue box labeled '온도 센서 [TMP36]' contains a text field with '이름 온도센서'. Below this, a breadboard circuit is shown with a TMP36 temperature sensor connected to an Arduino Uno R3. The sensor's VCC pin is connected to a 5V pin on the breadboard, GND to a GND pin, and the output pin to analog pin A0. The Arduino IDE code editor shows the following code:

```
1 void setup()  
2 {  
3   Serial.begin(9600);  
4 }  
5  
6 void loop()  
7 {  
8   int reading = analogRead(A0);  
9  
10  float voltage = reading * 5.0;  
11  voltage /= 1024.0;  
12  
13  Serial.print(voltage); Serial.println(" volts");  
14  
15  float temperatureC = (voltage - 0.5) * 100 ;  
16  Serial.print(temperatureC); Serial.println(" degrees C");  
17  
18  float temperatureF = (temperatureC * 9.0 / 5.0) + 32.0;  
19  Serial.print(temperatureF); Serial.println(" degrees F");  
20  
21  delay(1000);  
22 }
```

At the bottom, the '시리얼 모니터' (Serial Monitor) window is open, displaying the following output:

```
0.00 volts  
33.01 degrees C  
91.41 degrees F  
0.83 volts  
33.01 degrees C  
91.41 degrees F  
0.83 volts  
33.01 degrees C  
91.41 degrees F  
0.83 volts  
33.01 degrees C  
91.41 degrees F
```

# 불꽃감지센서 (Flame sensor)

- 불꽃 또는 화염은 사람의 눈으로 확인 할 수 없는 자외선과 적외선의 파장이 발생
- 불꽃감지센서는 적외선 감지센서로서 760nm ~ 1100nm파장을 감지한다.

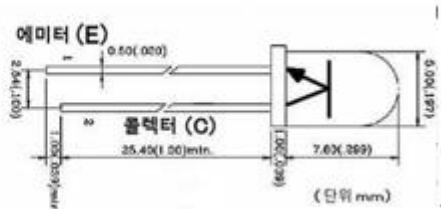
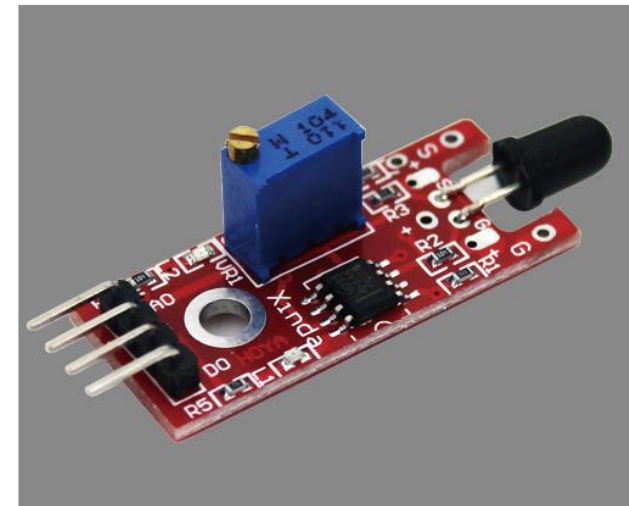
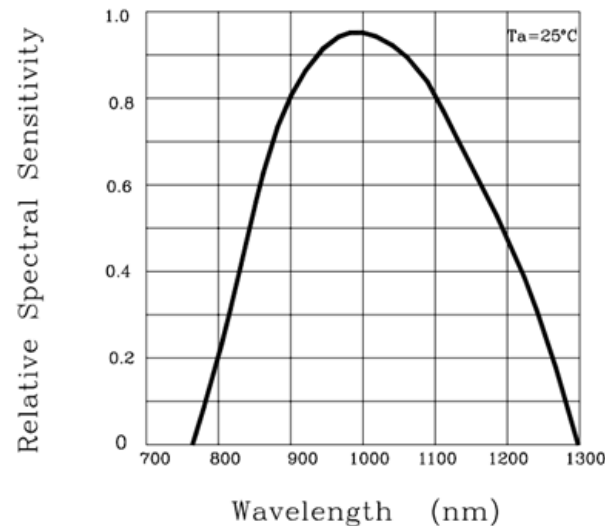
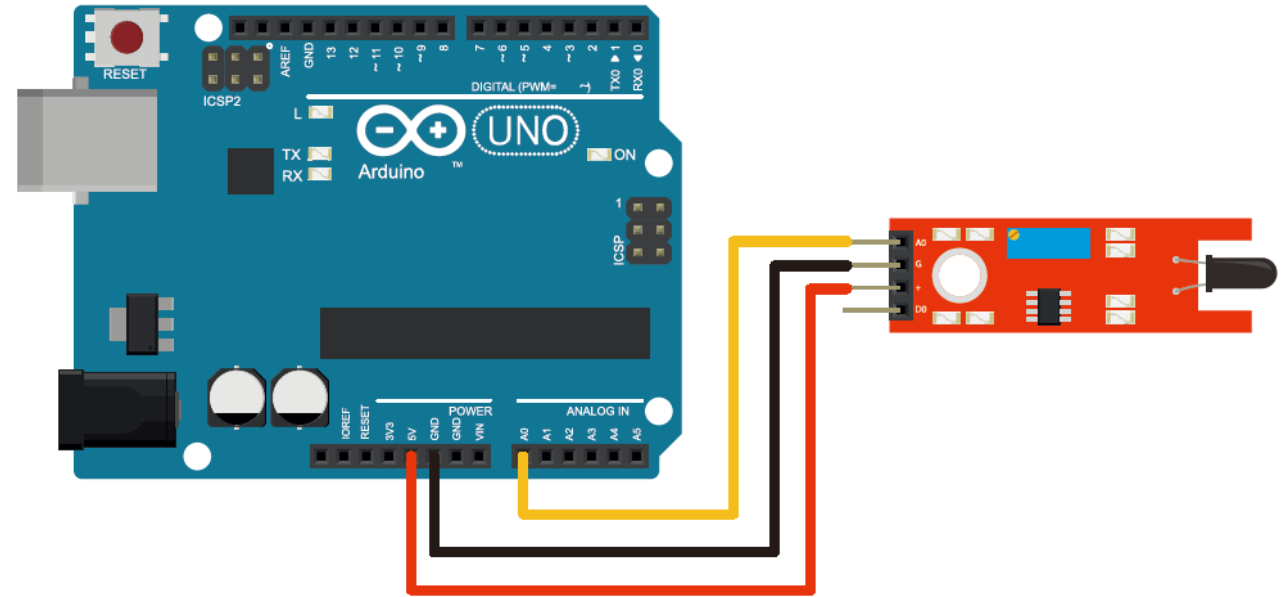


Fig. 5 Spectral Sensitivity



# 불꽃감지센서 (Flame sensor)

- 불꽃 감지 아두이노 실험 구성
  - 센서모듈 A0 <> 아두이노 A0
  - 센서모듈 G <> 아두이노 GND
  - 센서모듈 + <> 아두이노 5V



# 불꽃감지센서 (Flame sensor)

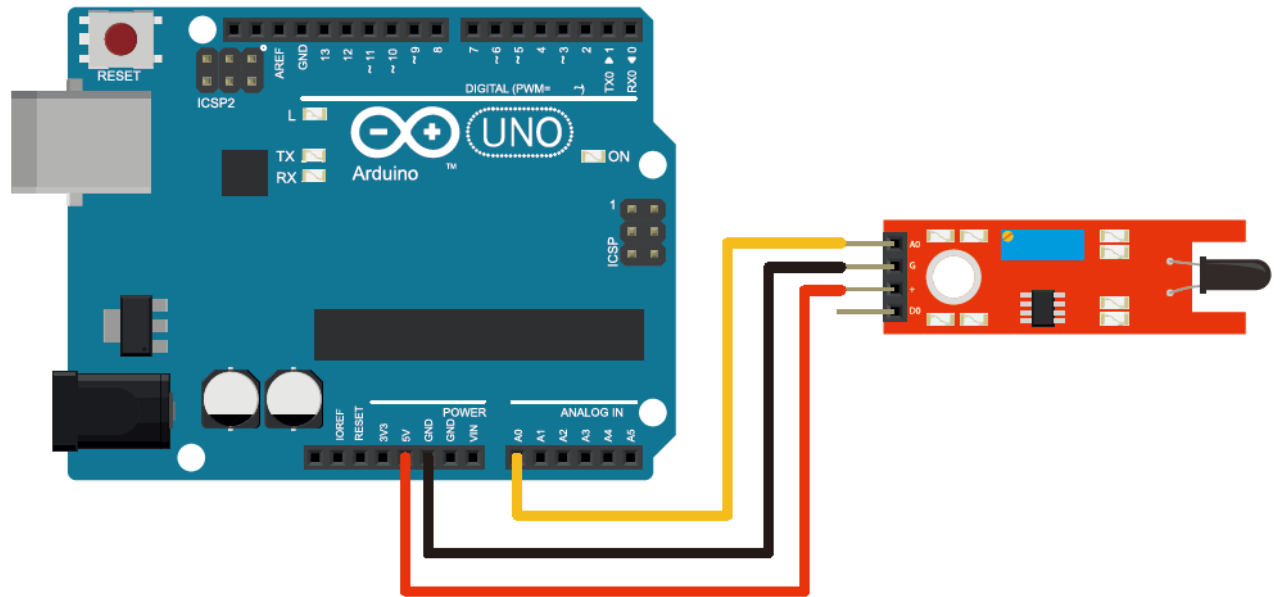
- 불꽃 감지 아두이노 실험 코드 작성

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

void loop()
{
  int analog_value = analogRead(A0);

  Serial.println(analog_value);

  delay(100);
}
```



# 불꽃감지센서를 이용한 화재감지 응용

- 불꽃이 감지 되면 자동으로 경고를 발생시키자!

