

신호등 LED, DotMatrix

신호등 LED, DotMatrix

시각 표현 소자

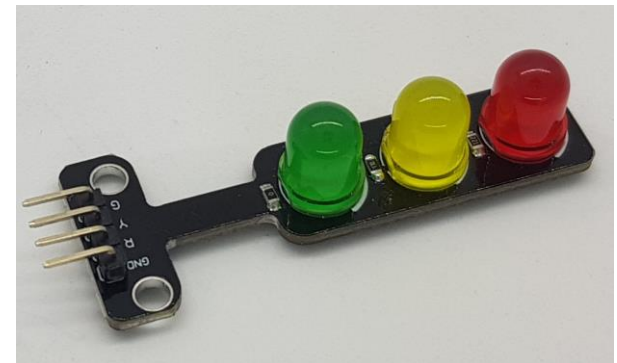
신호등 모듈 사용해보기

■모듈이란?

- ❖ 회로를 구성한 보드
- ❖ 기판위에 부품을 배치하고 결선하여 인터페이스를 간단하게 만든 전자 보드를 모듈이라 한다

■신호등 모듈

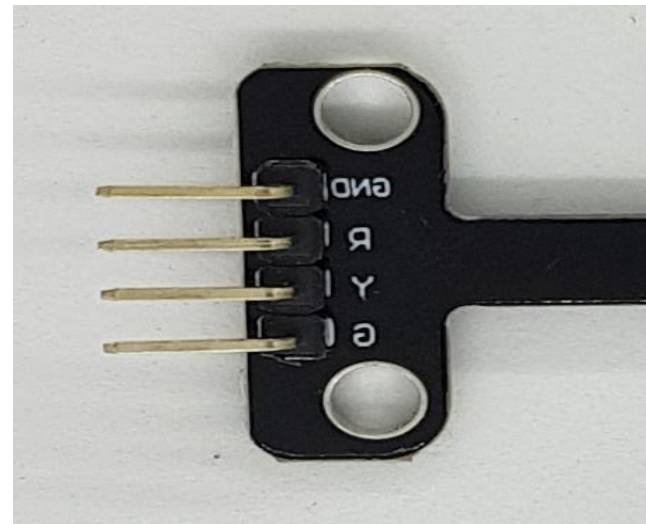
- ❖ 빨강, 노랑 초록 LED 모듈
- ❖ GND 공통에 각 LED에 HIGH 공급하여 켤 수 있다
- ❖ 핀이름은 R = 빨강, Y = 노랑, G = 초록, GND



신호등 모듈 핀

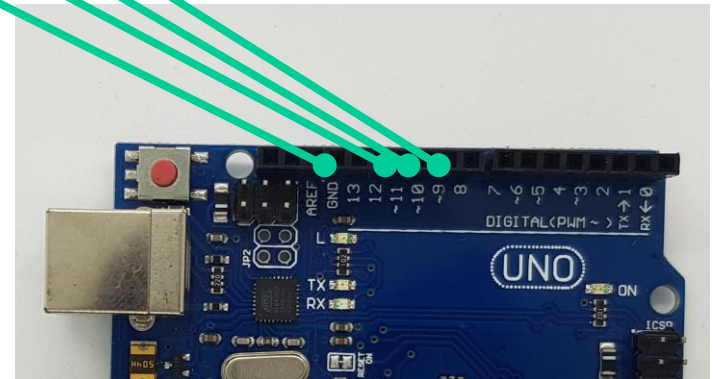
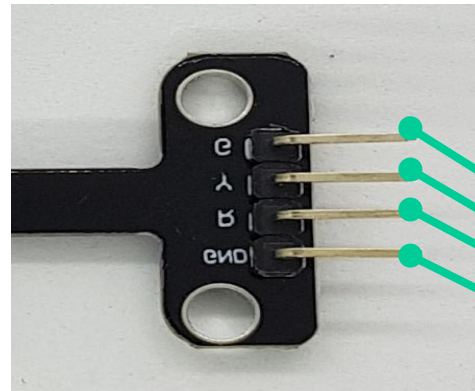
■핀 이름은 다음과 같이 연결

- ❖ GND – 0V or GND
- ❖ R – Red LED
- ❖ Y – Yellow LED
- ❖ G – Green LED



신호등 연결

신호등모듈	아두이노
GND	GND
R	11
Y	10
G	9



신호등 모듈 - 코드(1)

```
#define RED_LED 11
#define YELLOW_LED 10
#define GREEN_LED 9

void setup()
{
  pinMode(RED_LED, OUTPUT);
  pinMode(YELLOW_LED, OUTPUT);
  pinMode(GREEN_LED, OUTPUT);
}
```

신호등 모듈 - 코드(2)

```
void loop()
{
    digitalWrite(RED_LED, HIGH);
    digitalWrite(YELLOW_LED, LOW);
    digitalWrite(GREEN_LED, LOW);
    delay(1000);
    digitalWrite(RED_LED, LOW);
    digitalWrite(YELLOW_LED, HIGH);
    digitalWrite(GREEN_LED, LOW);
    delay(1000);
    digitalWrite(RED_LED, LOW);
    digitalWrite(YELLOW_LED, LOW);
    digitalWrite(GREEN_LED, HIGH);
    delay(1000);
}
```

과제 – 신호등 모듈 수정

```
static int status;

void setup()
{
    status = 0;
    pinMode(RED_LED, OUTPUT);
    pinMode(YELLOW_LED, OUTPUT);
    pinMode(GREEN_LED, OUTPUT);
}

void loop()
{
    digitalWrite(RED_LED, LOW);
    digitalWrite(YELLOW_LED, LOW);
    digitalWrite(GREEN_LED, LOW);
```

```
    switch(status){
        case 0:
            status = 1;
            _____ BLANK 1 _____
            break;
        case 1:
            status = 2;
            _____ BLANK 2 _____
            break;
        case 2:
            status = 0;
            _____ BLANK 3 _____
            break;
        default:
            break;
    };
    delay(1000);
}
```

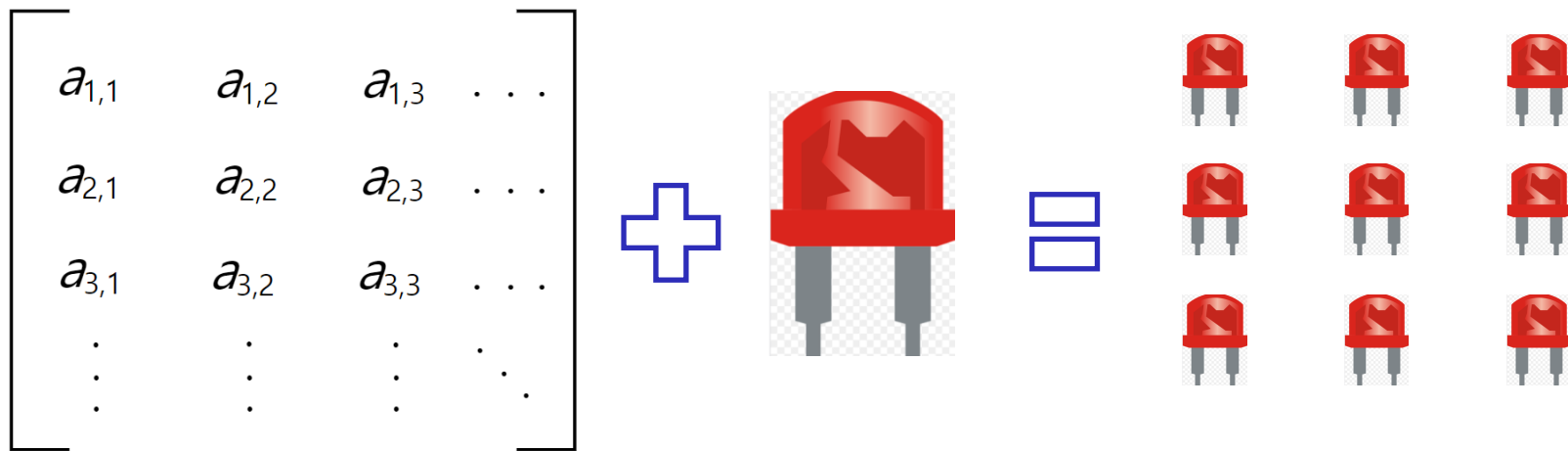
과제 – 기존 코드의 동작과 같이 1초에 한 개씩
켜지도록 **BLANK1, 2, 3** 을 채워주세요

시각 표현 소자

도트메트릭스(DotMatrix)

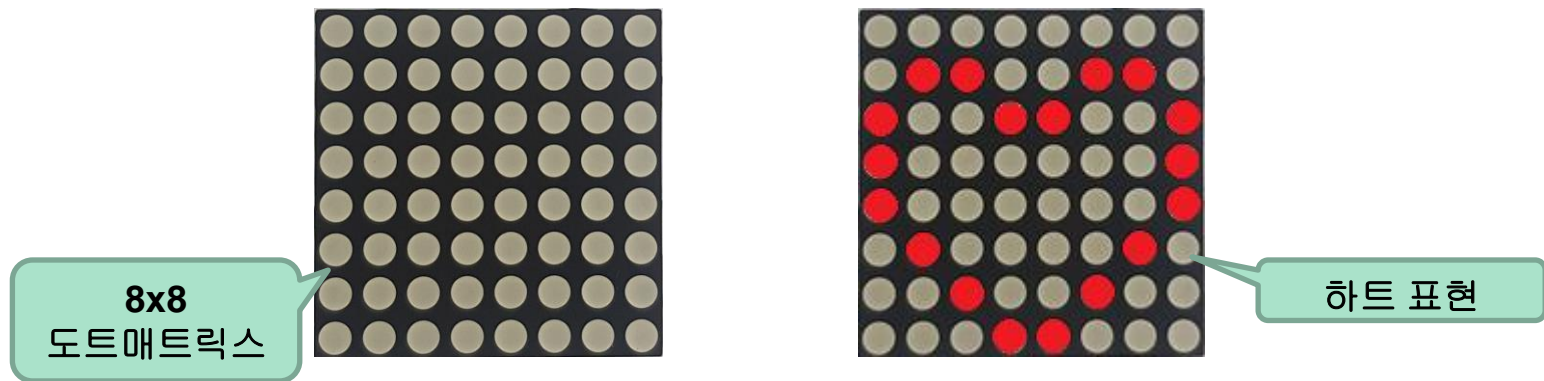
■ Dot + Matrix

- ❖ Dot는 LED를 의미한다.
- ❖ Matrix 는 행렬 배열 모양이다.
- ❖ LED 배열을 DotMatrix 라고 한다.



도트매트릭스(DotMatrix)

- 도트매트릭스는 행렬 LED 개수에 따라 사이즈가 가변이다
 - ❖8x8 또는 16x16. 그 이상의 배열도 있다.
- 문자, 숫자, 그림 등 Display 목적으로 사용 된다
- LED는 휘도가 높아 멀리서도 정확하게 보인다
- 전광판 또는 안내판 등에 사용되며, 일상 생활에서 많이 접할 수 있다

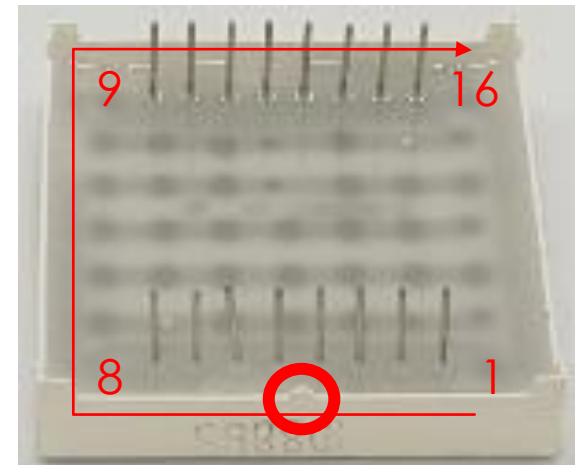


도트매트릭스 전광판



도트매트릭스 사용 방법

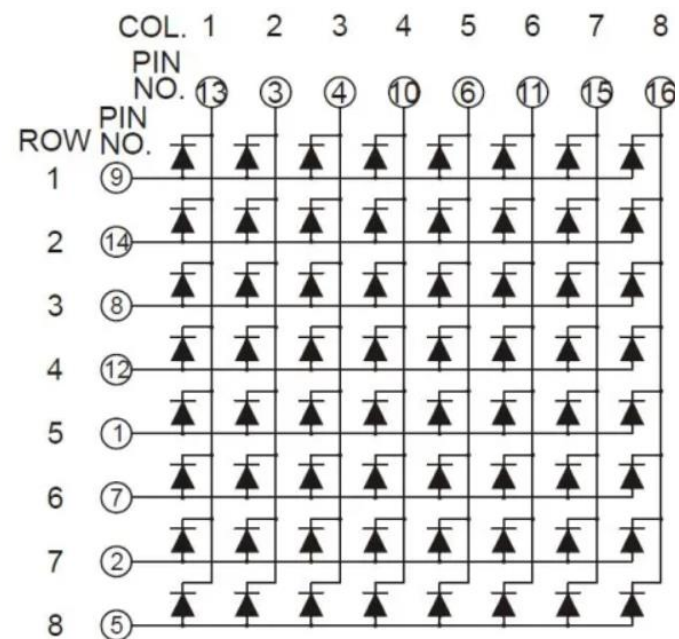
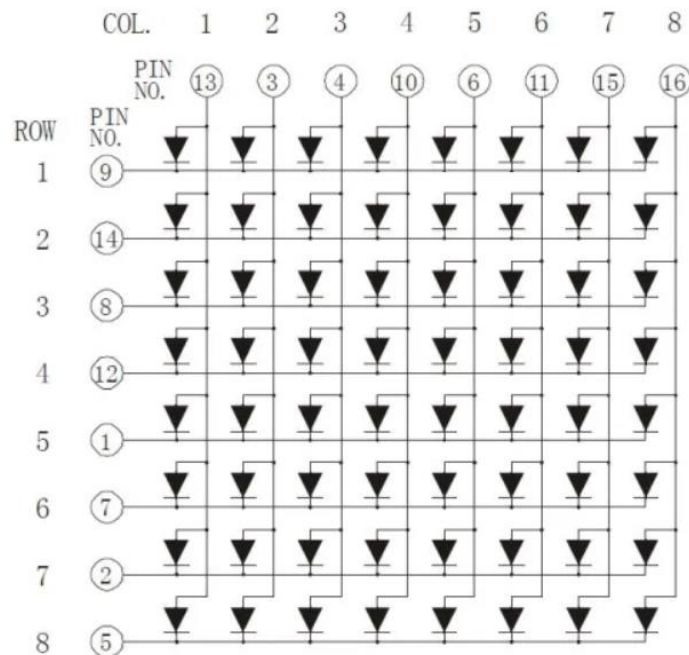
- 행렬의 핀이 있으며, 교차해서 선택할 수 있다
- 8x8 도트매트릭스는 16개의 핀이 있다
- 반시계 방향으로 핀이 카운트 된다
- COL과 ROW 핀이 섞여 있다
 - ❖ COL 13,3,4,10,6,11,15,16
 - ❖ ROW 9, 14, 8, 12, 1, 7, 2, 5



도트매트릭스 사용 방법

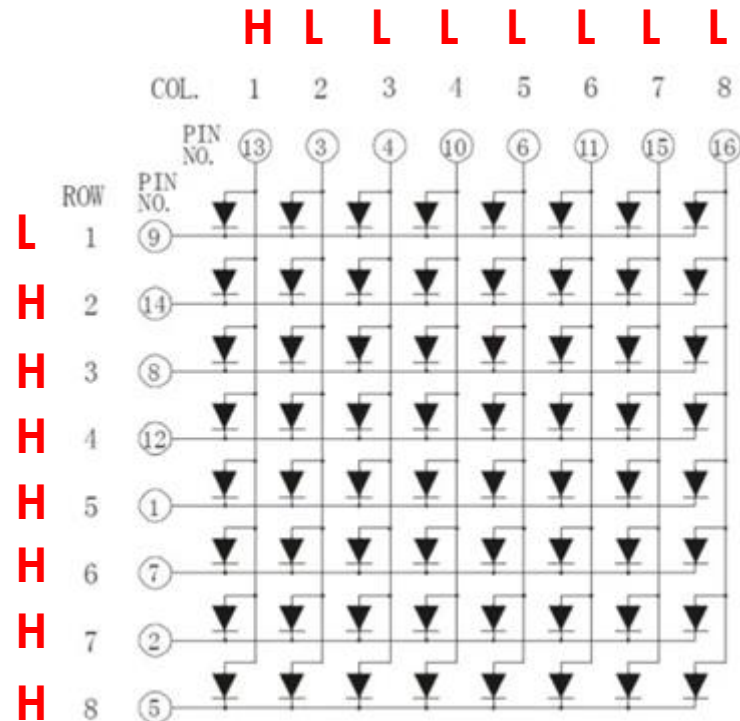
■ LED의 극성에 따라 두 가지 형태가 있다

❖ Common Cathode/Common Anode



도트매트릭스 사용 방법

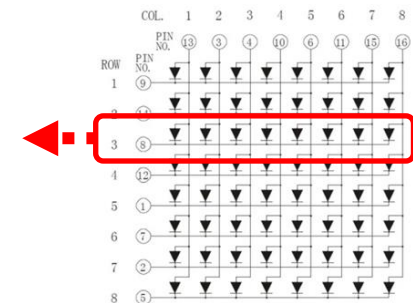
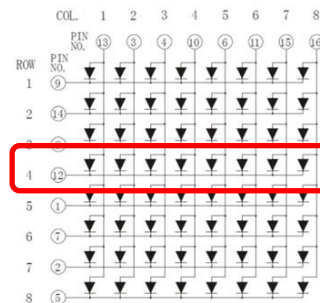
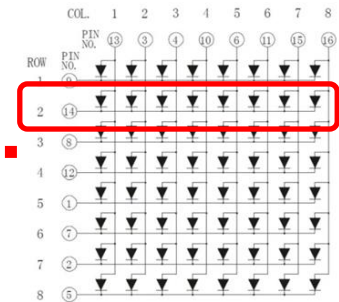
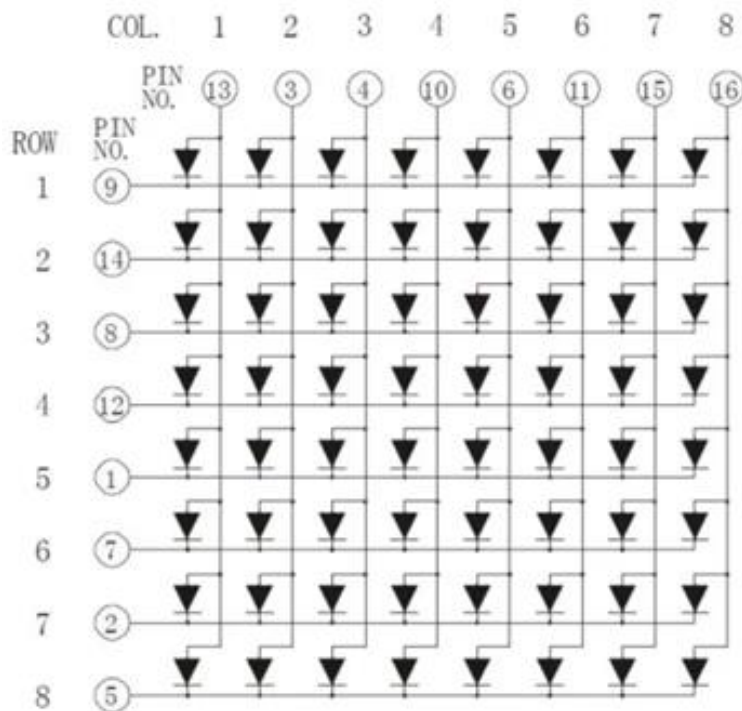
■ 한번에 16핀을 모두 제어 해야한다



도트매트릭스 사용 방법

■착시 현상을 이용하여 행 단위로 빠르게 제어한다

❖사람의 눈은 잔상이 사라지기전에 다시 비추면 계속 켜있다고 느낀다



도트매트릭스 연결도(Common Cathod)

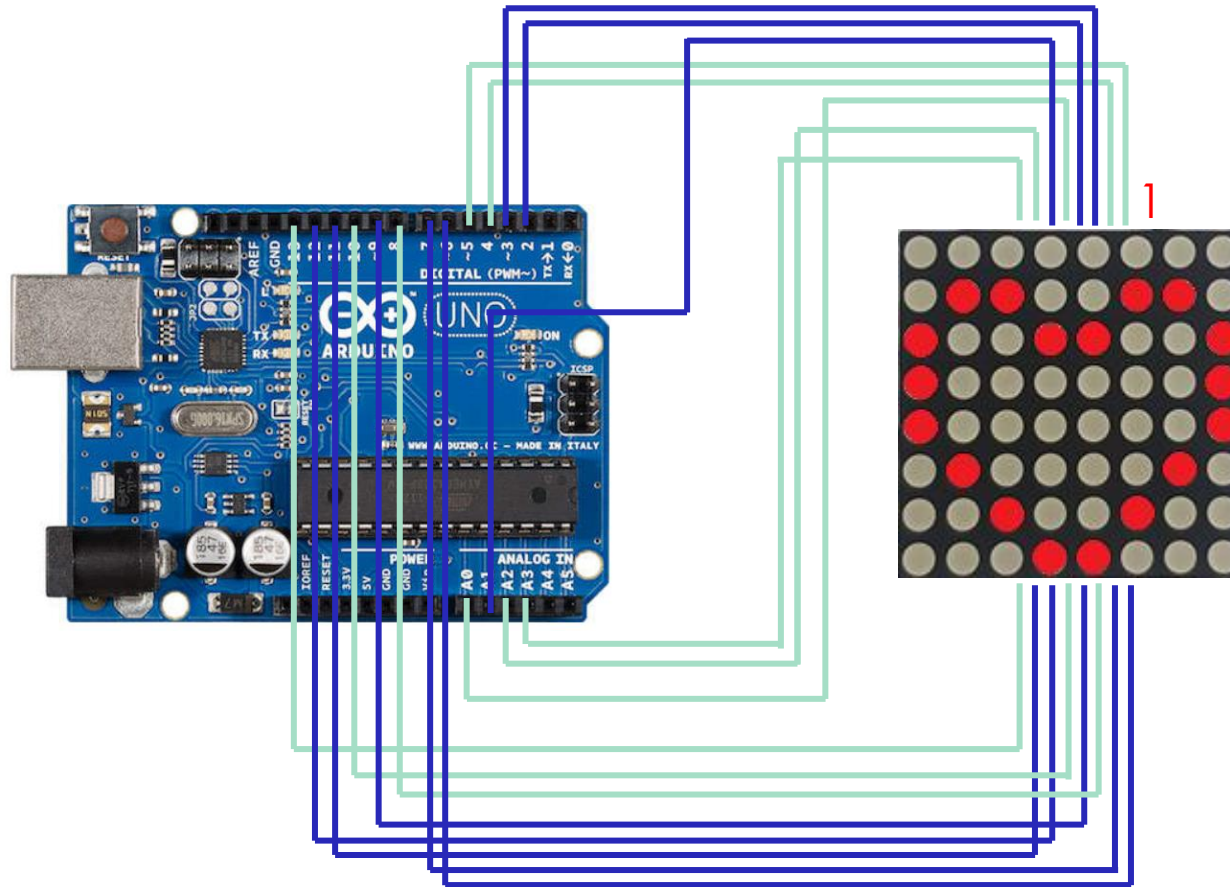
■ COL : 9,3,2,12,15,11,7,6

■ ROW: 13,8,17,10,5,16,4,14

COL	PIN	Arduino
1	13	9
2	3	3
3	4	2
4	10	12
5	6	15
6	11	11
7	15	7
8	16	6

ROW	PIN	Arduino
1	9	13
2	14	8
3	8	17
4	12	10
5	1	5
6	7	16
7	2	4
8	5	14

도트매트릭스 연결도(Common Cathode)



Common Anode Dotmatrix 켜기

```
void setup() {  
    pinMode( 9,OUTPUT);  
    pinMode( 3,OUTPUT);  
    pinMode( 2,OUTPUT);  
    pinMode( 12,OUTPUT);  
    pinMode( 15,OUTPUT);  
    pinMode( 11,OUTPUT);  
    pinMode( 7,OUTPUT);  
    pinMode( 6,OUTPUT);  
  
    pinMode( 13,OUTPUT);  
    pinMode( 8,OUTPUT);  
    pinMode( 17,OUTPUT);  
    pinMode( 10,OUTPUT);  
    pinMode( 5,OUTPUT);  
    pinMode( 16,OUTPUT);  
    pinMode( 4,OUTPUT);  
    pinMode( 14,OUTPUT);  
}
```

```
void loop() {  
    // Col  
    digitalWrite( 9,HIGH);  
    digitalWrite( 3,LOW);  
    digitalWrite( 2, LOW);  
    digitalWrite( 12, LOW);  
    digitalWrite( 15, LOW);  
    digitalWrite( 11, LOW);  
    digitalWrite( 7, LOW);  
    digitalWrite( 6, LOW);  
  
    // Row  
    digitalWrite( 13, LOW);  
    digitalWrite( 8, HIGH);  
    digitalWrite( 17, HIGH);  
    digitalWrite( 10, HIGH);  
    digitalWrite( 5, HIGH);  
    digitalWrite( 16, HIGH);  
    digitalWrite( 4, HIGH);  
    digitalWrite( 14, HIGH);  
}
```

스마일 예제 (1/5)

//표현할 문자를 2진수로 정의.

```
#define DOT { \  
    {0,0,1,1,1,1,0,0}, \  
    {0,1,0,0,0,0,1,0}, \  
    {1,0,1,0,0,1,0,1}, \  
    {1,0,0,0,0,0,0,1}, \  
    {1,0,1,0,0,1,0,1}, \  
    {1,0,0,1,1,0,0,1}, \  
    {0,1,0,0,0,0,1,0}, \  
    {0,0,1,1,1,1,0,0} \  
}
```

```
byte col = 0;  
byte leds[8][8];
```

스마일 예제 (2/5)

```
int pins[17] = {-1, 5, 4, 3, 2, 14, 15, 16, 17, 13, 12, 11, 10, 9, 8, 7, 6};
```

```
int cols[8] = {pins[13], pins[3], pins[4], pins[10],  
pins[6], pins[11], pins[15], pins[16]};
```

```
int rows[8] = {pins[9], pins[14], pins[8], pins[12], pins[1], pins[7],  
pins[2], pins[5]};
```

```
int pattern[8][8] = DOT;
```

스마일 예제 (3/5)

```
void setup() {  
  //아두이노 디지털 핀 출력 설정  
  for (int i = 1; i <= 16; i++) {  
    pinMode(pins[i], OUTPUT);  
  }  
  
  //LED를 켜기 전에 모두 끄기  
  for (int i = 1; i <= 8; i++) {  
    digitalWrite(cols[i - 1], LOW);  
  }  
  for (int i = 1; i <= 8; i++) {  
    digitalWrite(rows[i - 1], LOW);  
  }  
}
```

스마일 예제 (4/5)

```
void loop() {  
    //출력  
    display();  
    delay(2);  
}
```

```
void display() {  
    //다음 column으로 넘어가기 전에 이전led를 끈다  
    digitalWrite(cols[col], LOW);  
  
    //현재 column  
    col++;  
    if (col == 8) {  
        col = 0;  
    }  
}
```

스마일 예제 (5/5)

```
for (int row = 0; row < 8; row++) {  
    if (pattern[col][7 - row] == 1) {  
        digitalWrite(rows[row], LOW); // LED ON  
    }  
    else {  
        digitalWrite(rows[row], HIGH); // LED OFF  
    }  
}  
  
digitalWrite(cols[col], HIGH);  
} //display() 함수 종료
```


마무리

■ Delay를 1 초로 늘려보자

```
void loop() {  
  //출력  
  display();  
  delay(2);      → delay(1000);  
}
```

❖행렬의 그리는 동작이 정확히 보이게 된다.

■ Tinkercad 를 사용하여 확인해보자

❖Led 를 8x8로 배열하여 시뮬레이션할 수 있다.

❖코드 검증 및 이해하는데 효과적이다.

장애물감지센서, 진동감지센서,
액체센서(빗물감지, 수위, 토양수분)

1011110001010010100110001010011110101010101111100111110101000011111000010000101010110011111010100001111100001000010101011111001111101010000111110000100001010101111100111110101000011111000010000101011001111101010000111110000100001010101010000101

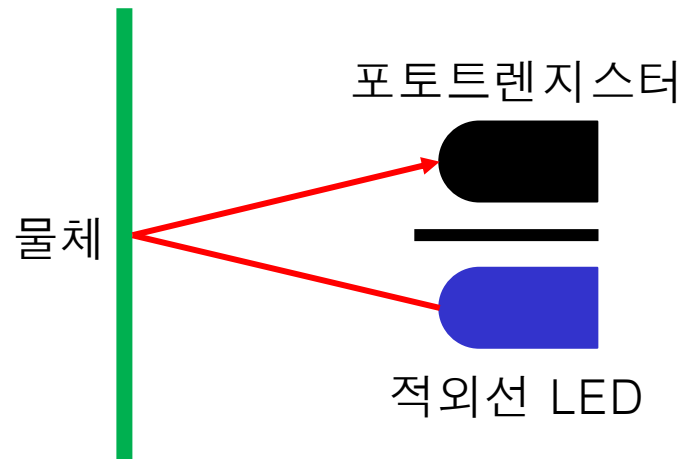
적외선 장애물감지센서 모듈

- 물체에 반사되는 적외선을 감지하여 디지털 신호로 출력.
- 감도조절부에서 감지거리를 조정 가능함(시계방향 거리증가).
- 검은색 물체는 빛을 흡수하는 성질을 이용해 흑백구분이 가능함.
- 로봇청소기, 라인트레이서 등에서 사용.
- 거리 측정에는 부적합함.



적외선 장애물감지센서 원리

- 적외선 LED에서 적외선을 보내게 되고, 물체에 닿아 반사되는 빛을 포토트랜지스터에서 감지함.



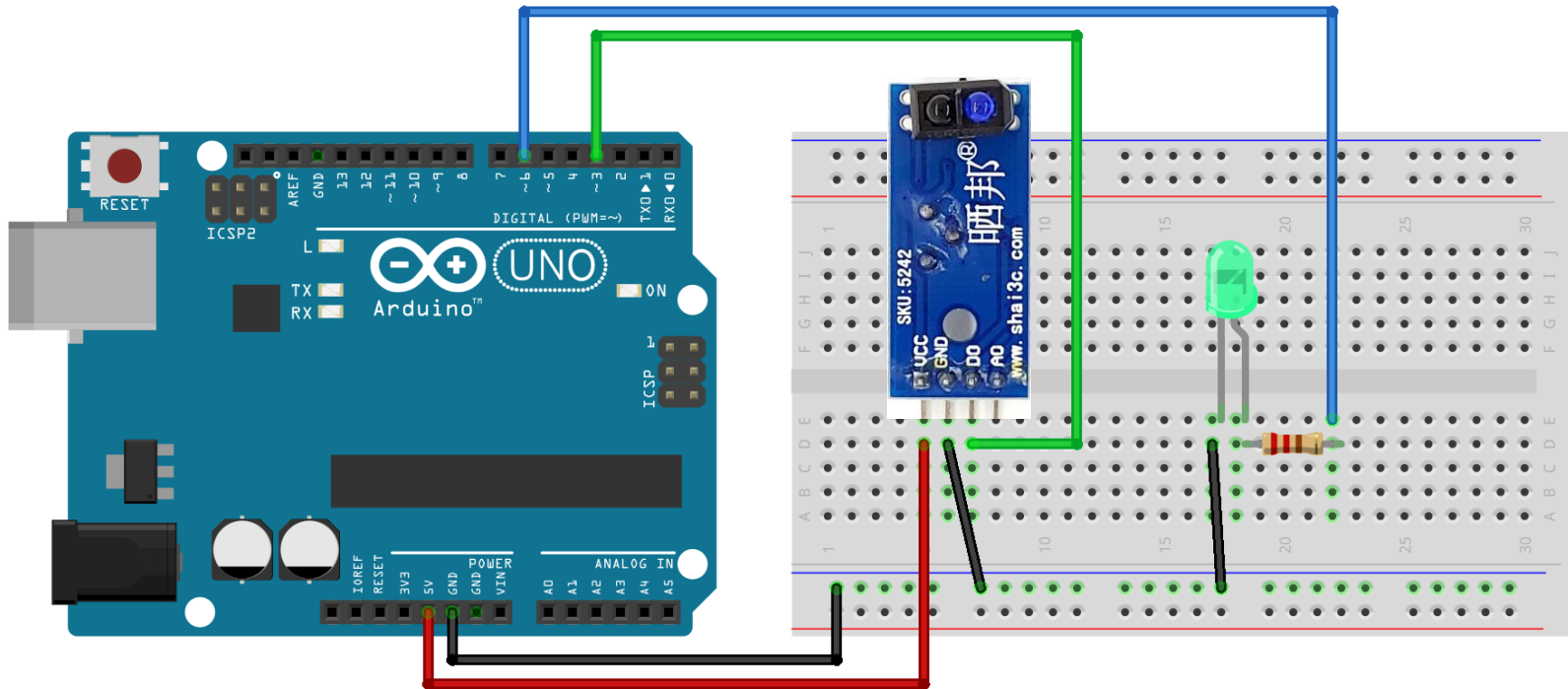
적외선 장애물감지센서 모듈 핀 아웃

■ 다음과 같이 연결

- ❖ GND – GND
- ❖ VCC – 5V
- ❖ D0 – 3



연결도



예제 코드

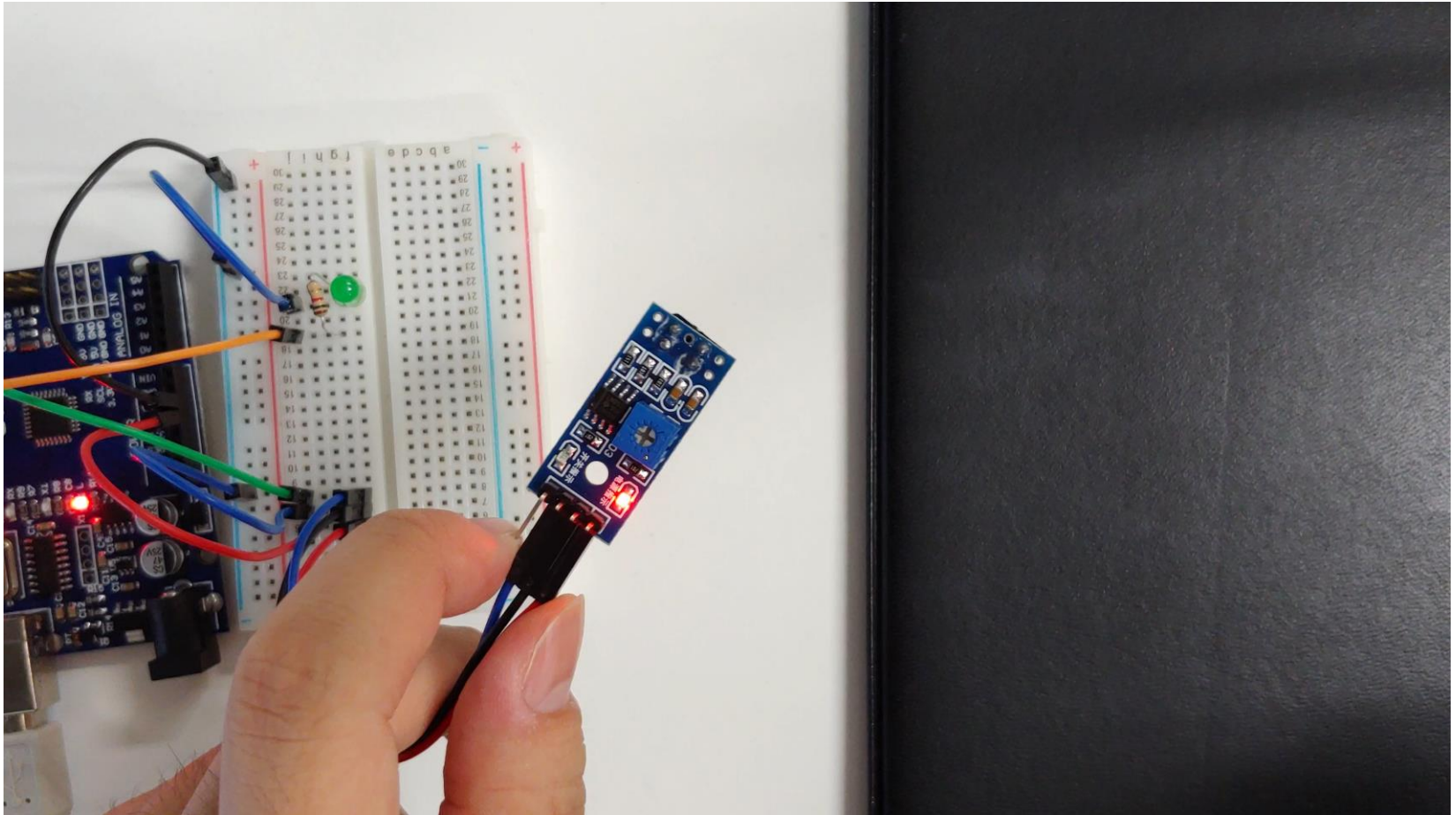
```
#define PHOTO_PIN 3
#define LED_PIN 6

void setup() {
  pinMode(PHOTO_PIN, INPUT);
  pinMode(LED_PIN, OUTPUT);
}

void loop() {
  int state = digitalRead(PHOTO_PIN);

  if(state == LOW){
    digitalWrite(LED_PIN, HIGH);
  }
  else{
    digitalWrite(LED_PIN, LOW);
  }
}
```


동작 확인



2. 진동감지센서

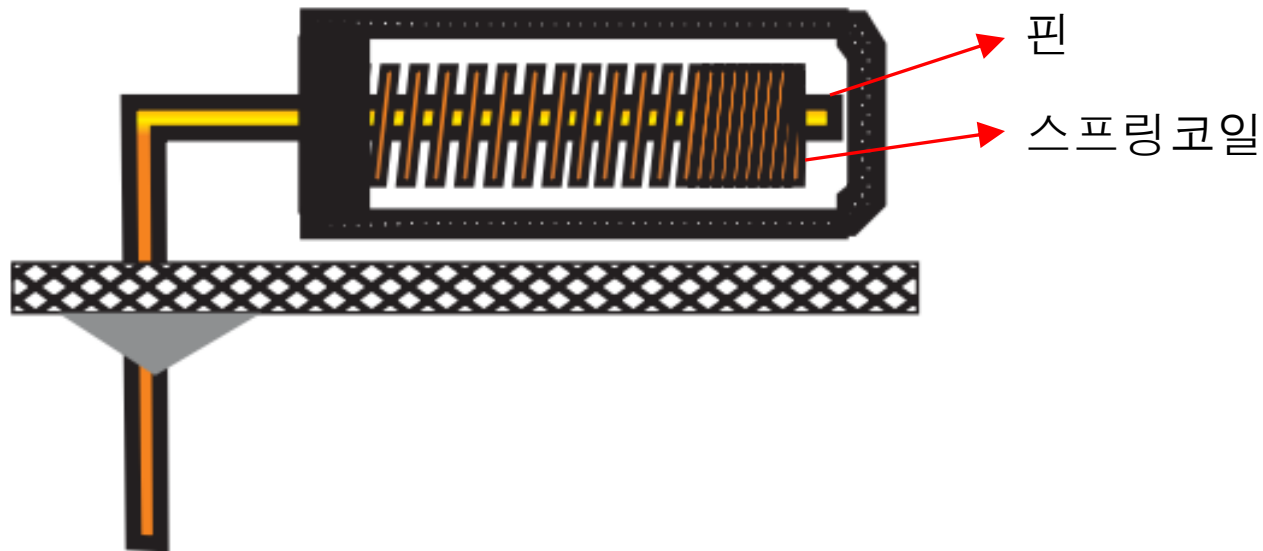
진동(충격)감지센서 모듈

- 센서안에 있는 스프링이 충격을 감지하여 디지털 신호로 출력.
- 민감도 조정이 가능한 모듈도 있음.
- 지진 측정기, 자동차의 도난경보장치, 만보기 등 다양한 곳에 사용.



진동(충격)감지센서 원리

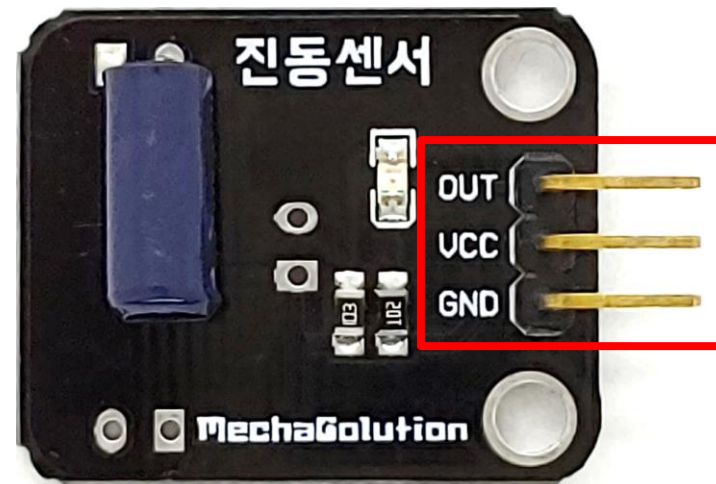
■ 일정강도 이상으로 진동하면 스프링코일이 핀에 접촉.



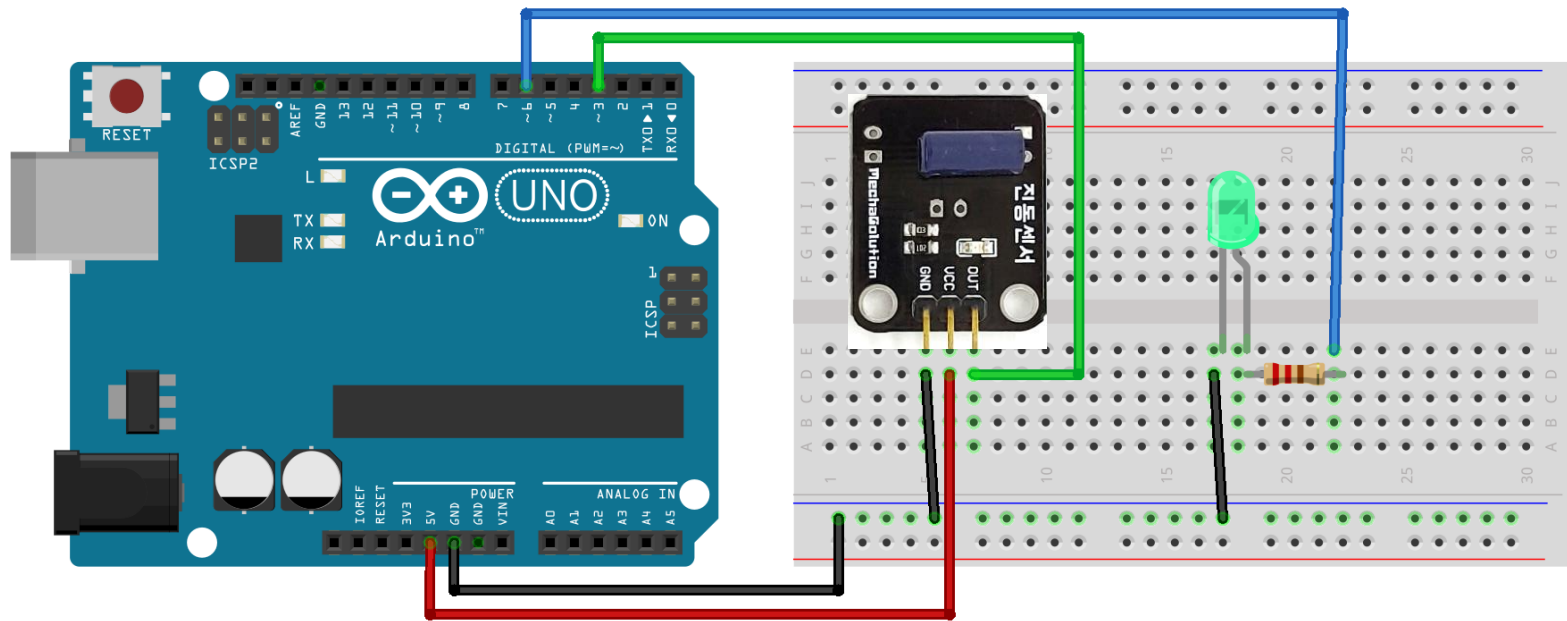
진동(충격)감지센서 모듈 핀 아웃

■ 다음과 같이 연결

- ❖ GND – GND
- ❖ VCC – 5V
- ❖ OUT – 3



연결도



예제 코드

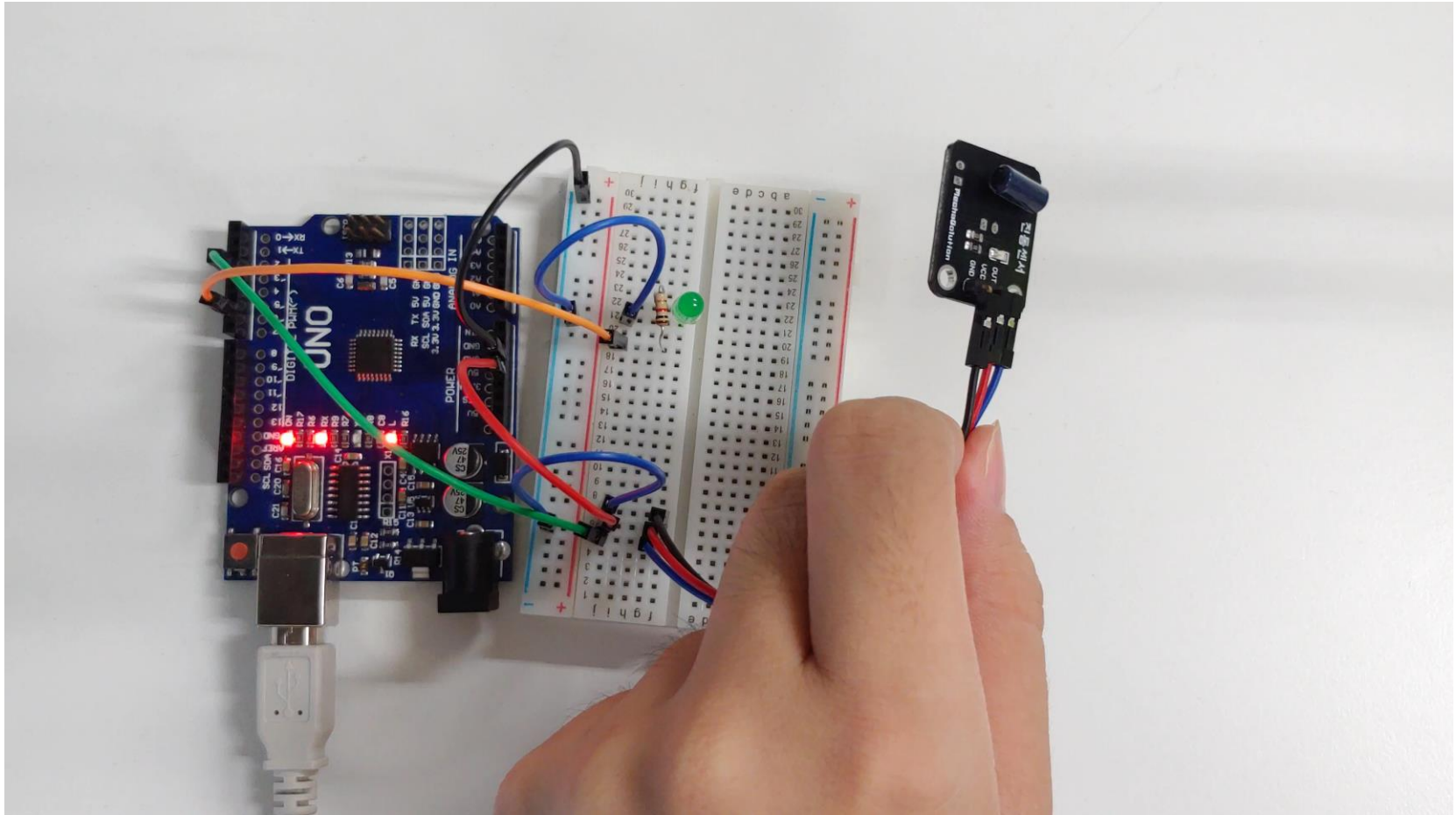
```
#define VIBRATION_PIN 3
#define LED_PIN 6

void setup() {
  pinMode(VIBRATION_PIN, INPUT);
  pinMode(LED_PIN, OUTPUT);
}

void loop() {
  int state = digitalRead(VIBRATION_PIN);

  if(state == LOW){
    digitalWrite(LED_PIN, HIGH);
    delay(1000);
  }
  else{
    digitalWrite(LED_PIN, LOW);
  }
}
```

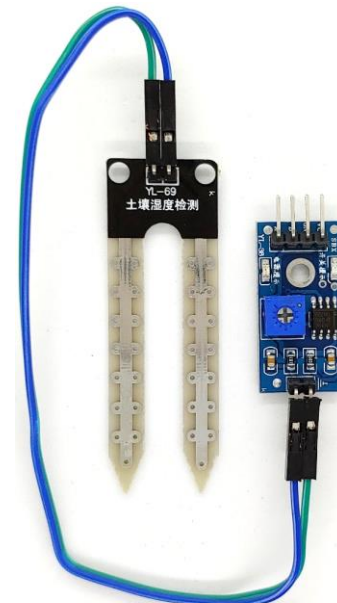
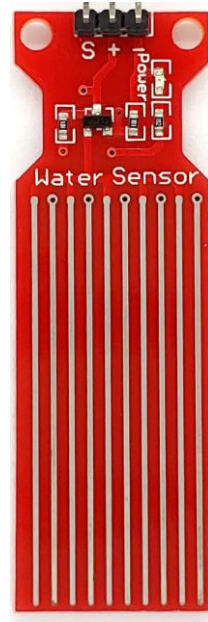
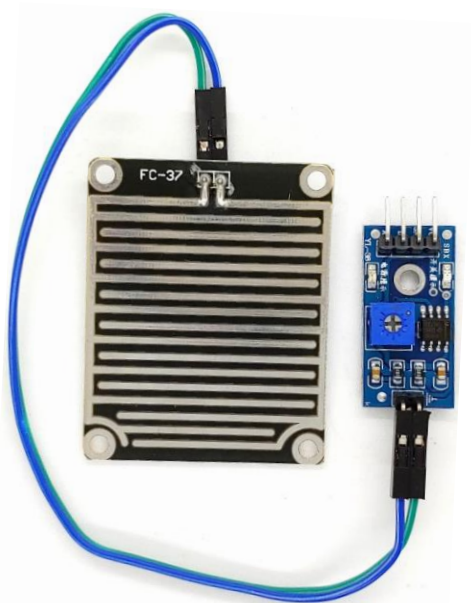

동작 확인



빗물감지, 수위, 토양수분

액체센서 란?

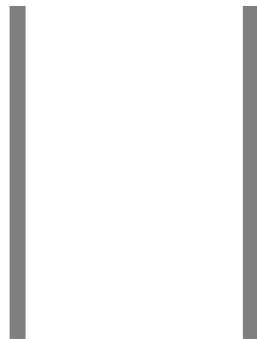
- 액체를 감지하는 센서들.
- 원리는 같지만 용도에 따라 분류된다.



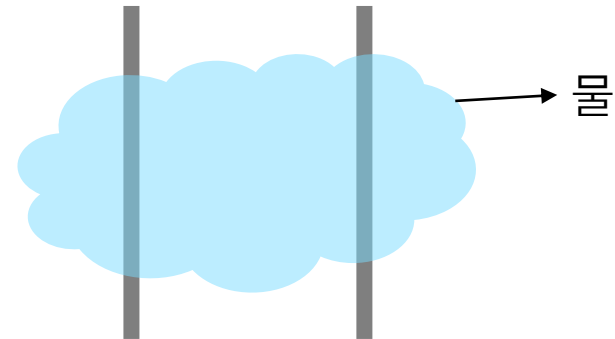
예시1. 빗물감지센서 예시2. 접촉식수위센서 예시3. 토양수분센서

액체센서 원리

- 일반 물은 불순물이나 미네랄 등이 함유되어 전류가 흐른다(불순물이 없는 증류수는 전류가 흐르지 않음).
- 액체센서는 저항 값이 무한대인 회로로 구성되어 있다(개방된 회로).
- 물이 두 도체에 닿으면 저항 값이 낮아져 전류가 흐른다(단힌 회로).
- 물에 접촉하는 표면적이 넓어질수록 저항 값이 더 낮아져 더 많은 전류가 흐른다.



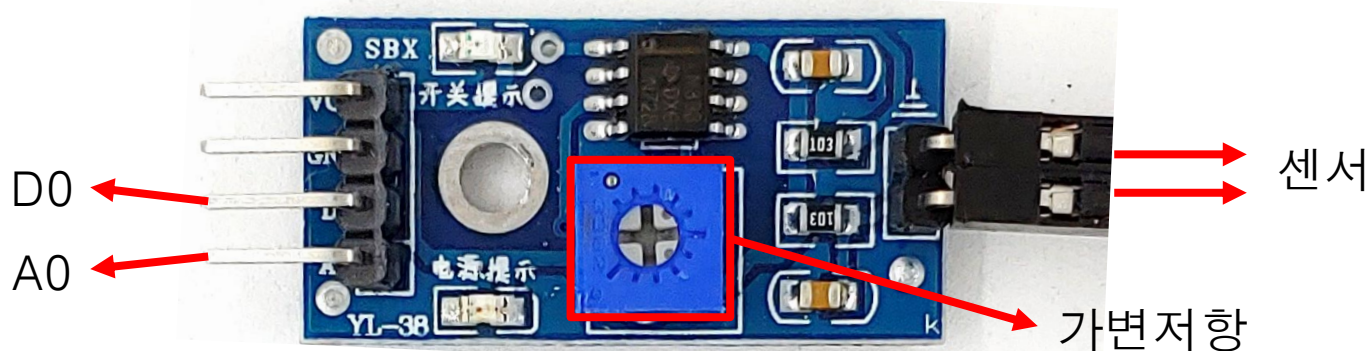
개방된 회로



단힌 회로

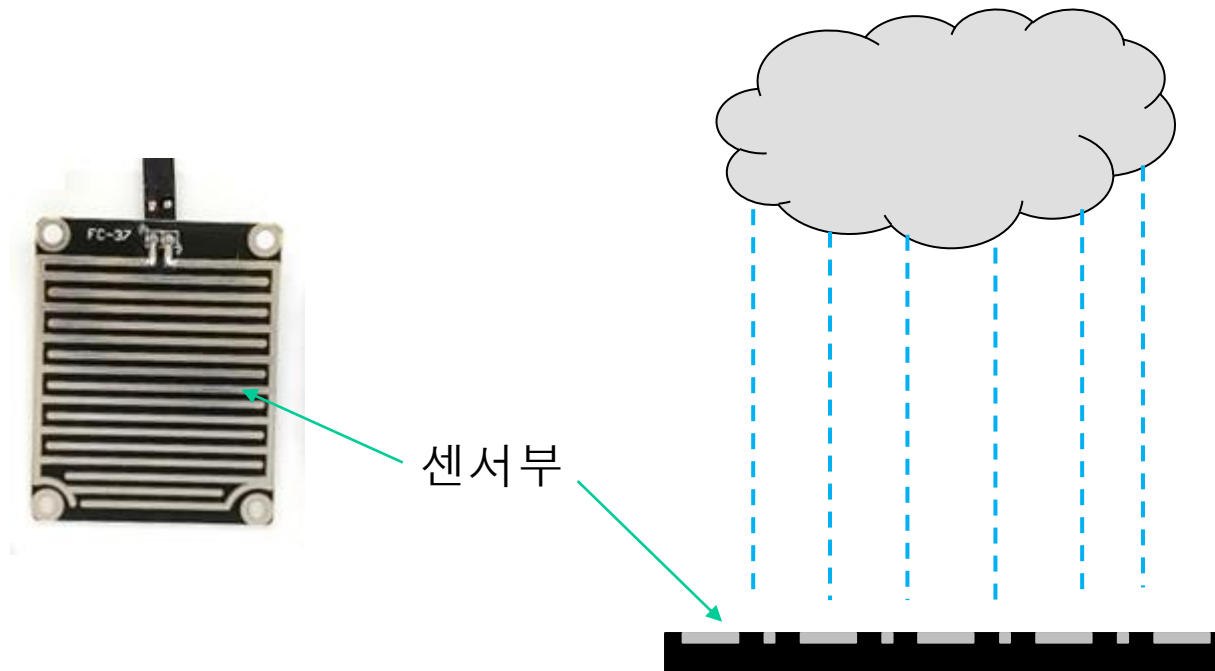
신호 증폭 회로 모듈

- 액체센서에서 감지된 신호는 미약하기 때문에 신호를 증폭 해주어야 한다.
- 디지털(D0)핀은 측정된 값이 기준치 이상이면 HIGH, 이하면 LOW를 출력하며, 이 기준치는 모듈에 포함된 가변저항을 사용하여 설정할 수 있다.
- 아날로그(A0)핀은 측정된 아날로그 값(0~1023)을 출력한다.



빗물감지센서 모듈 사용방법

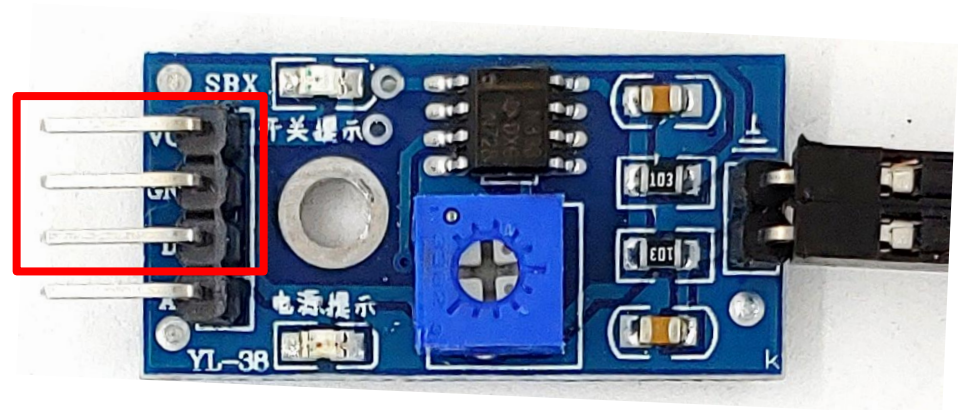
- 하늘에서 떨어지는 빗물을 잘 맞을 수 있도록 센서부를 하늘로 향하게 놓고 측정한다.



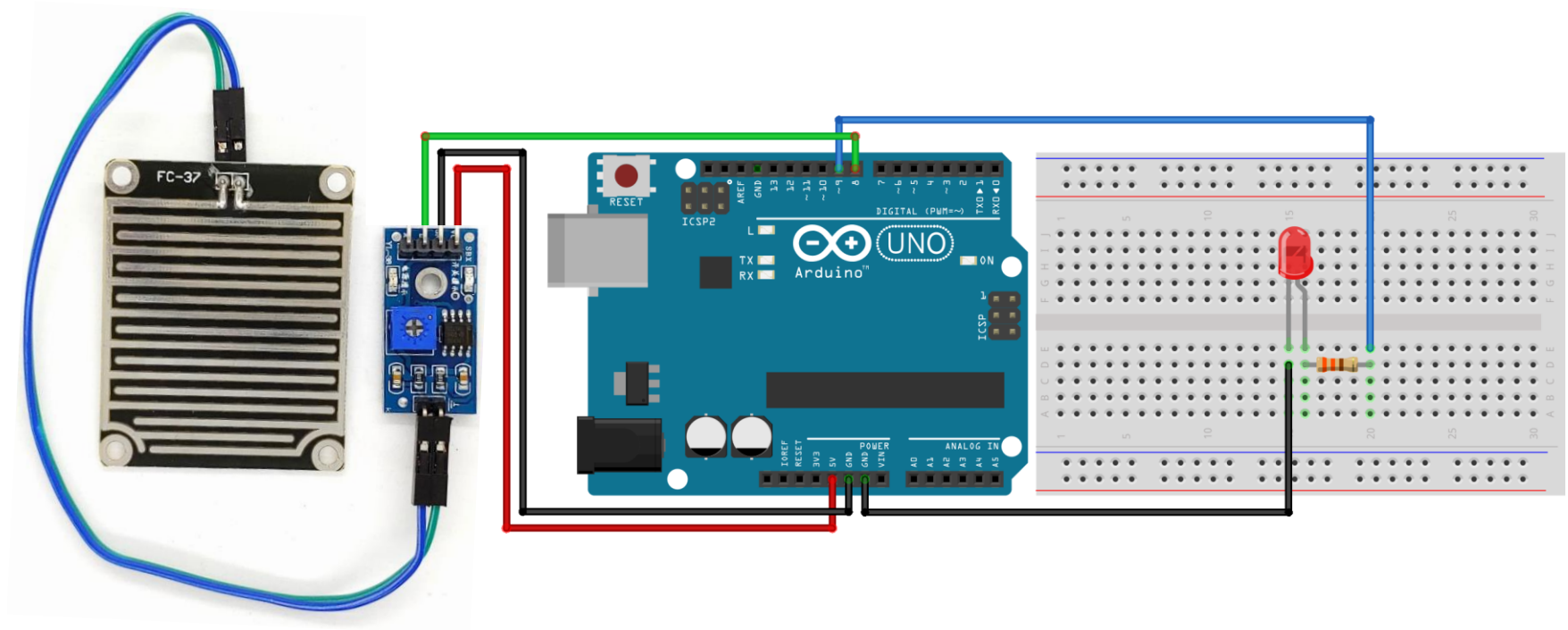
신호 증폭 회로 모듈 핀 아웃(빛물감지)

■ 다음과 같이 연결

- ❖ GND – GND
- ❖ VCC – 5V
- ❖ D0 – 8



연결도



예제 코드

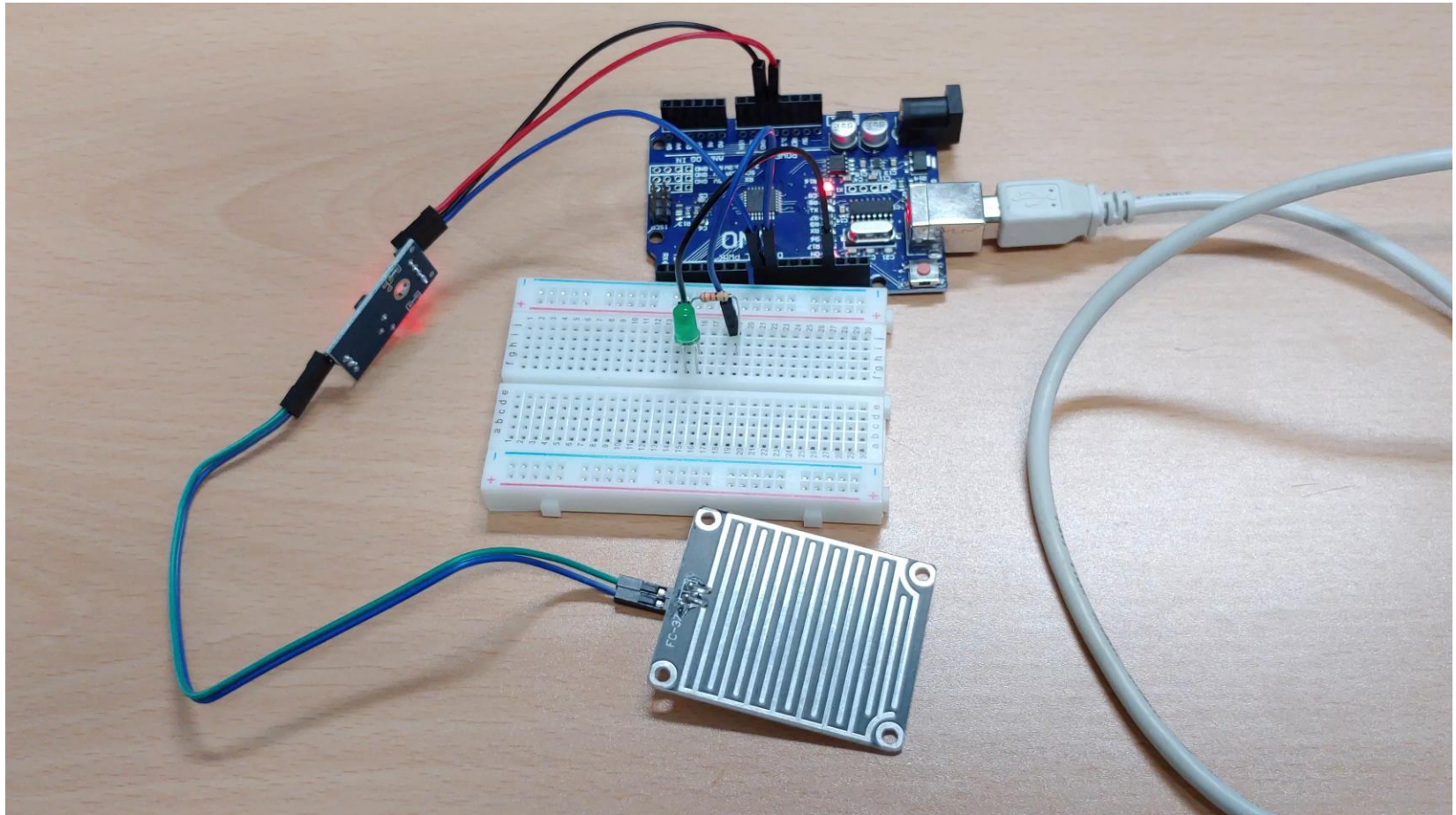
```
#define RAIN_PIN 8
#define LED_PIN 9

void setup() {
  pinMode(RAIN_PIN, INPUT);
  pinMode(LED_PIN, OUTPUT);
}

void loop() {
  int state = digitalRead(RAIN_PIN);

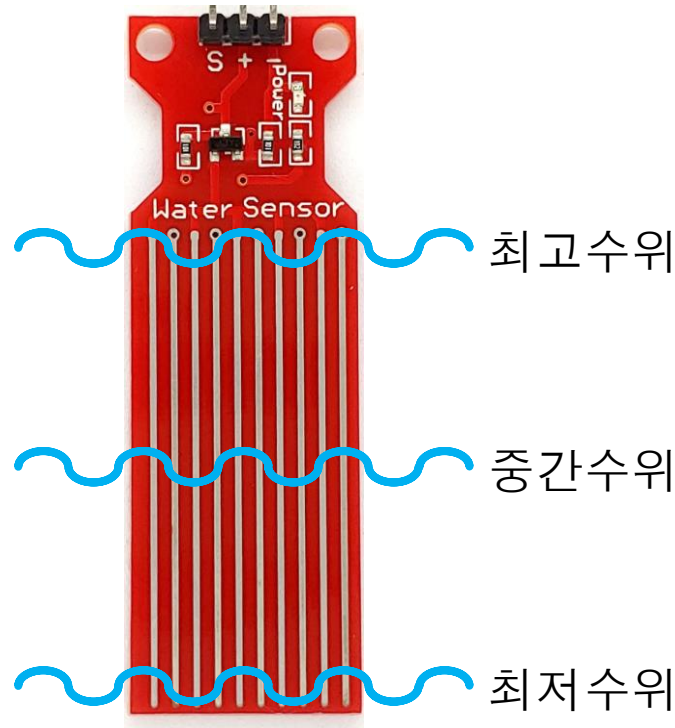
  if(state == LOW){
    digitalWrite(LED_PIN, HIGH);
  }
  else{
    digitalWrite(LED_PIN, LOW);
  }
}
```


동작 확인



접촉식 수위센서 모듈 사용방법

- 수위에 따라 변화가 잘 측정되도록 세워서 측정한다.
- 해당 수위센서는 이미 모듈로 구성되어 있어 별도로 신호 증폭 회로 모듈이 필요 없다.



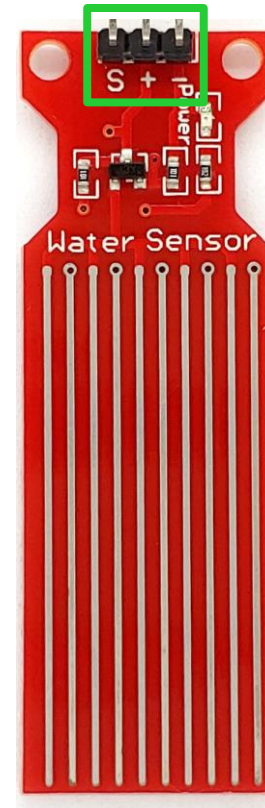
접촉식 수위센서 모듈 핀 아웃

■ 다음과 같이 연결

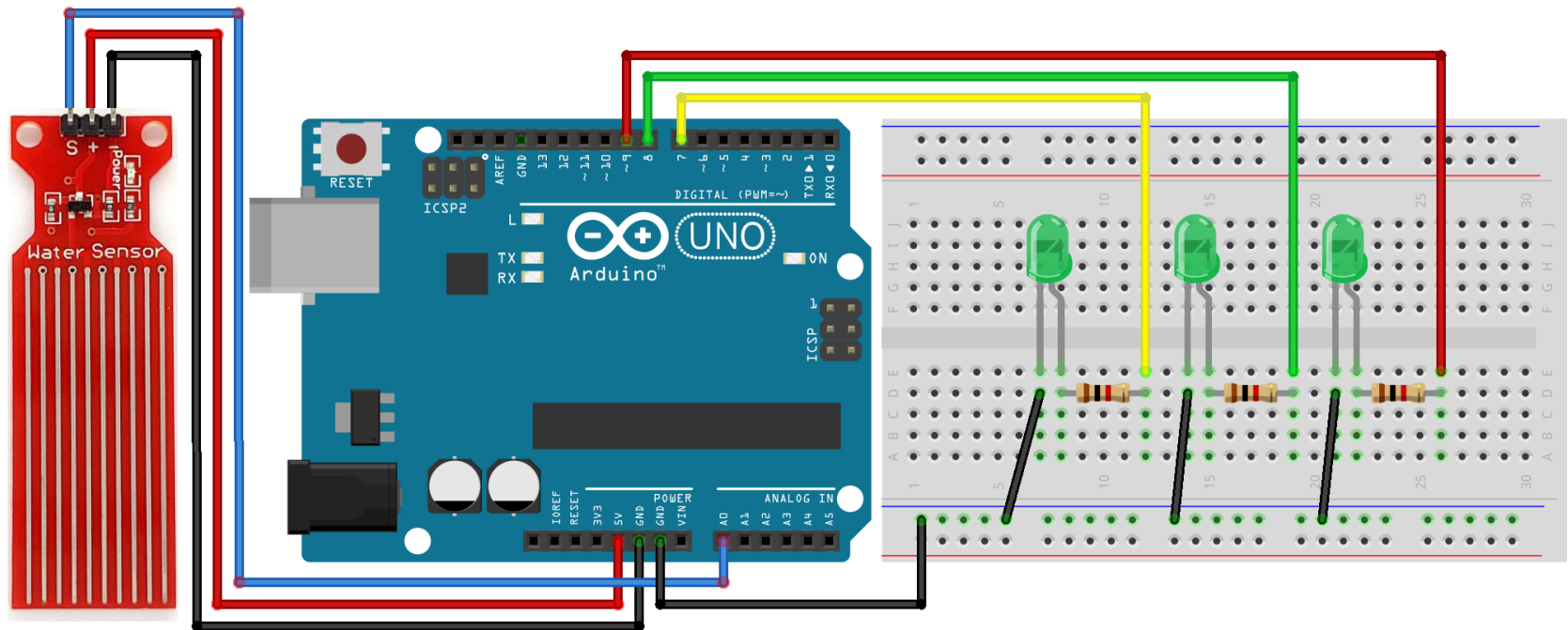
❖ - - GND

❖ + - 5V

❖ S - A0



연결도



예제 코드

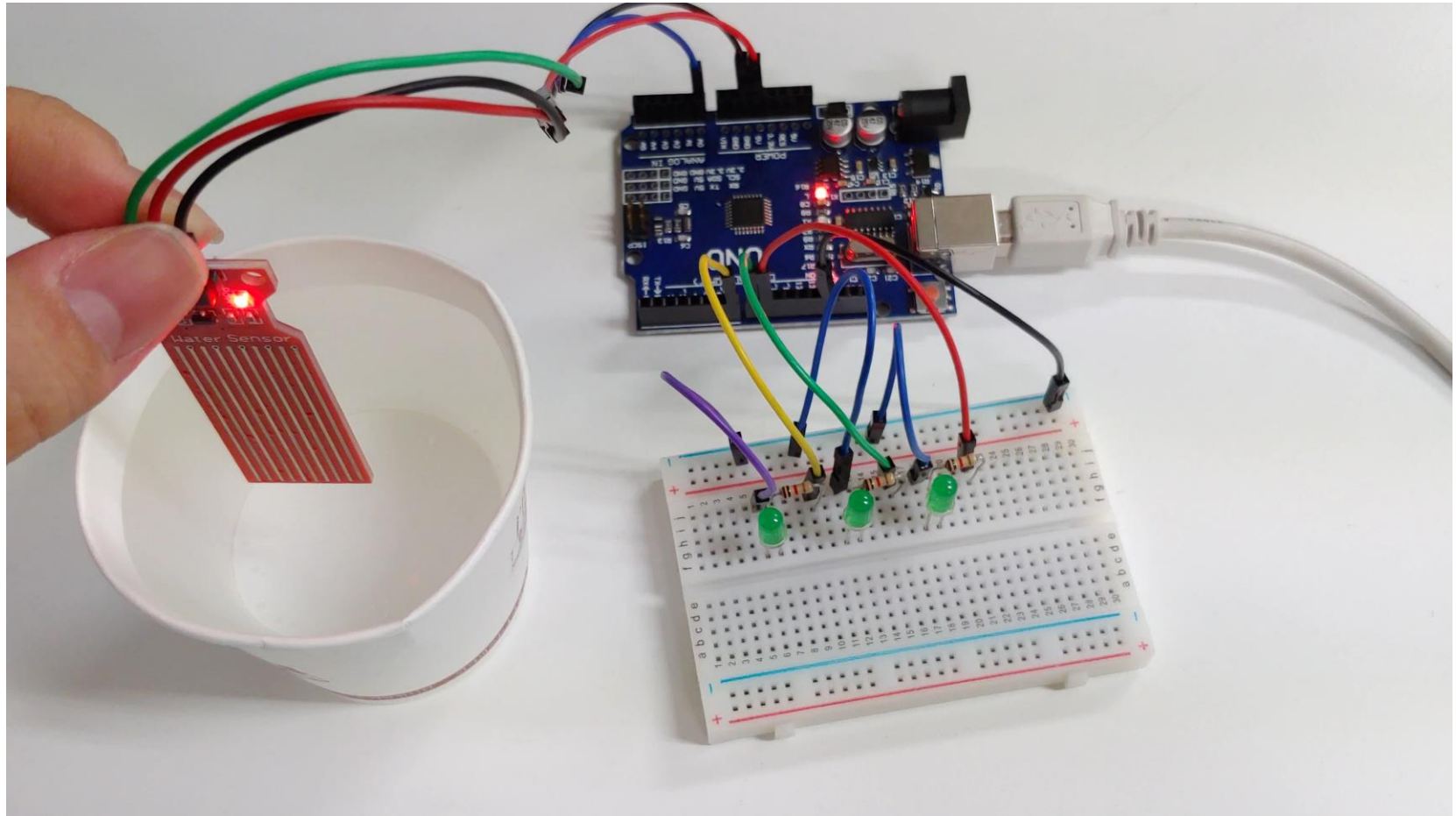
```
#define LEVEL_PIN A0
#define LED_PIN_1 7
#define LED_PIN_2 8
#define LED_PIN_3 9

void setup() {
  pinMode(LEVEL_PIN, INPUT);
  pinMode(LED_PIN_1, OUTPUT);
  pinMode(LED_PIN_2, OUTPUT);
  pinMode(LED_PIN_3, OUTPUT);
}

void loop() {
  int value = analogRead(LEVEL_PIN);
  int level_1 = 200;
  int level_2 = 560;
  int level_3 = 600;

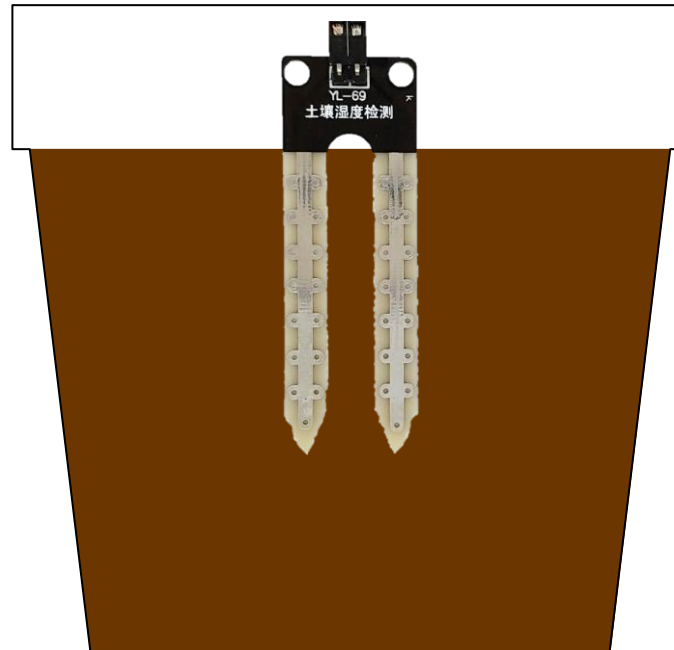
  if(value <= level_1){
    digitalWrite(LED_PIN_1, LOW);
    digitalWrite(LED_PIN_2, LOW);
    digitalWrite(LED_PIN_3, LOW);
  }
  if(value > level_1 && value <= level_2){
    digitalWrite(LED_PIN_1, HIGH);
    digitalWrite(LED_PIN_2, LOW);
    digitalWrite(LED_PIN_3, LOW);
  }
  if(value > level_2 && value <= level_3){
    digitalWrite(LED_PIN_1, HIGH);
    digitalWrite(LED_PIN_2, HIGH);
    digitalWrite(LED_PIN_3, LOW);
  }
  if(value > level_3){
    digitalWrite(LED_PIN_1, HIGH);
    digitalWrite(LED_PIN_2, HIGH);
    digitalWrite(LED_PIN_3, HIGH);
  }
}
```

동작 확인



토양수분센서 모듈 사용방법

- 흙과 접하는 면이 많도록 센서부를 최대한 흙 속에 넣는다.



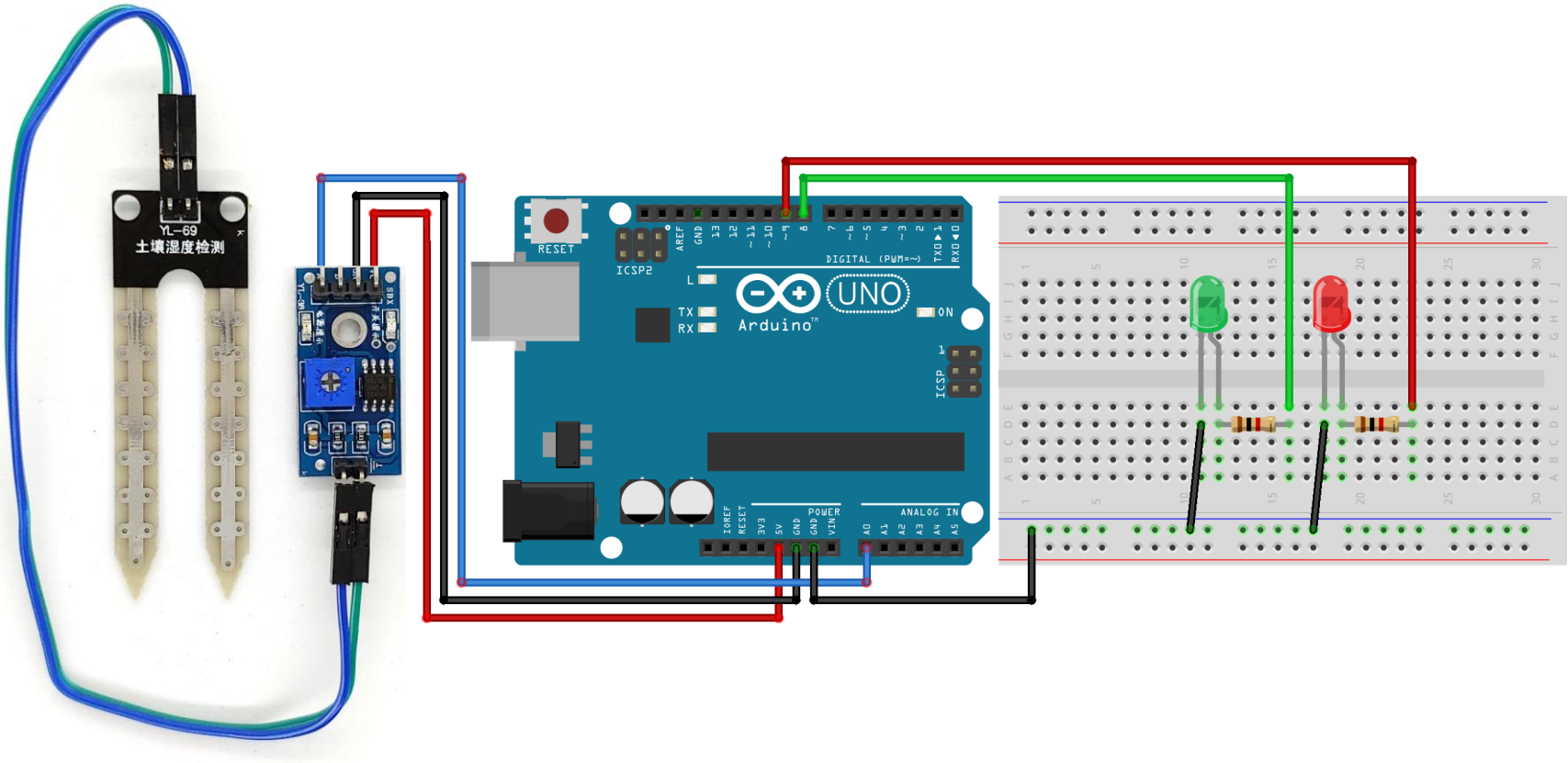
신호 증폭 회로 모듈 핀 아웃(토양수분)

■ 다음과 같이 연결

- ❖ GND – GND
- ❖ VCC – 5V
- ❖ A0 – A0



연결도



예제 코드

```
#define SOLID_PIN A0
#define GREEN_PIN 8
#define RED_PIN 9

void setup() {
  pinMode(SOLID_PIN, INPUT);
  pinMode(GREEN_PIN, OUTPUT);
  pinMode(RED_PIN, OUTPUT);
}

void loop() {
  int value = analogRead(SOLID_PIN);

  if(value <= 512){
    digitalWrite(GREEN_PIN, HIGH);
    digitalWrite(RED_PIN, LOW);
  }
  else{
    digitalWrite(GREEN_PIN, LOW);
    digitalWrite(RED_PIN, HIGH);
  }
}
```

동작 확인

