

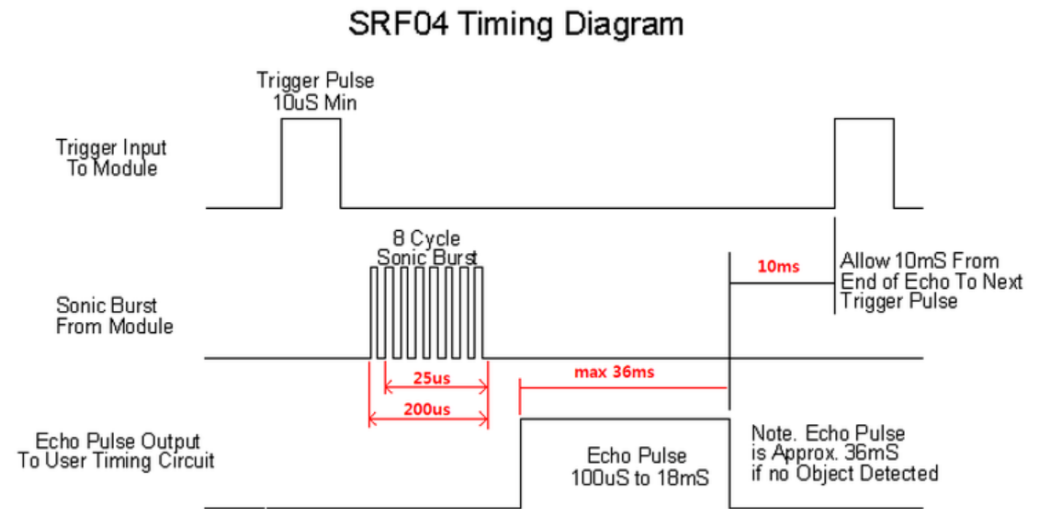
# ToF 인터럽트 실험

# 목표

- 초음파를 이용한 거리 측정 실험
- IR센서를 이용한 흰/검 측정 실험
- IR센서를 이용한 화재감지 및 화재경보기 제작
- 인터럽트를 이용한 디지털(ON/OFF) 센서 실험

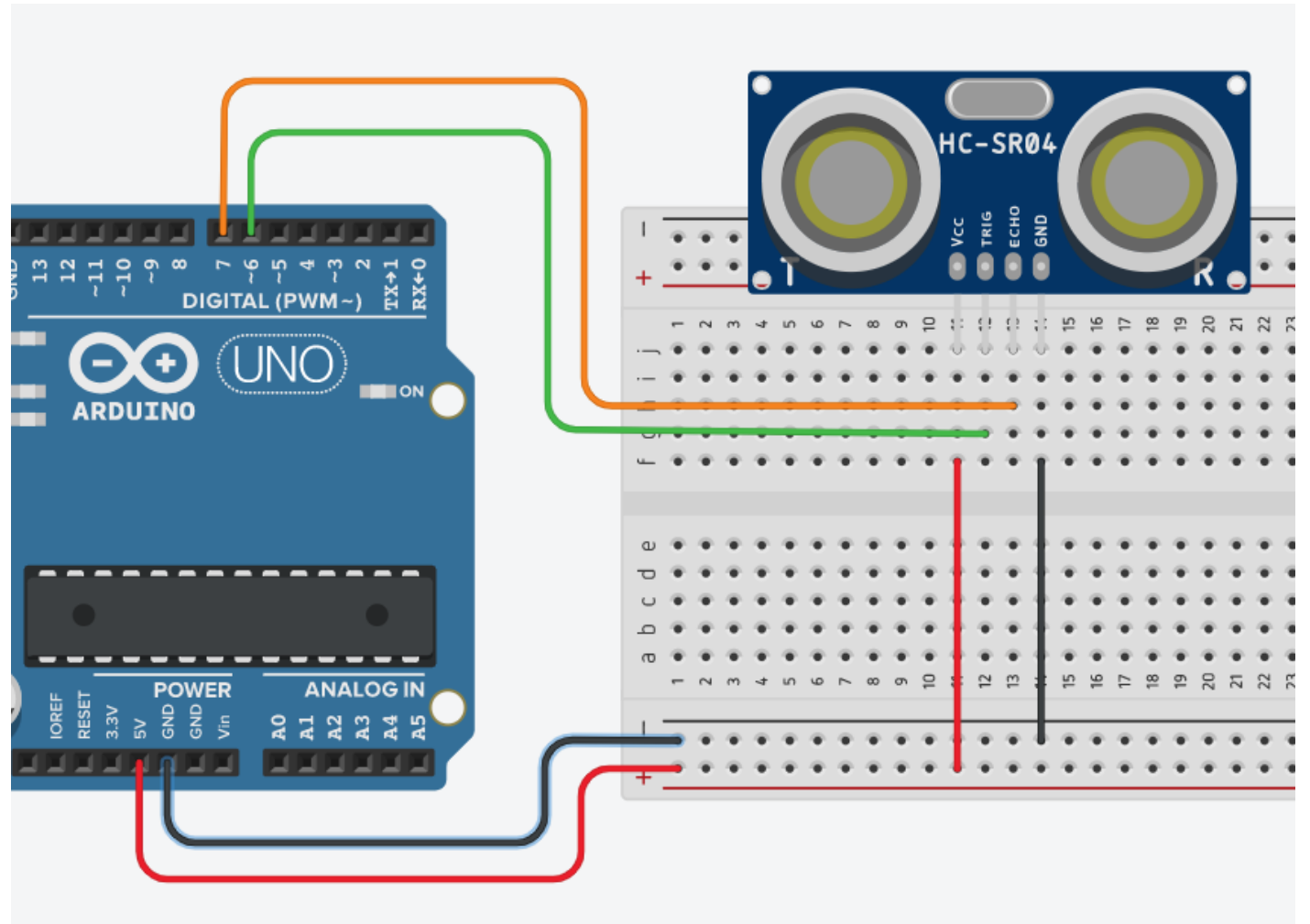
# 초음파 센서 모듈

- SRF04 초음파 모듈을 사용하여 장애물까지의 거리 측정



# 아두이노를 이용한 초음파 센서 실험

- VCC ↔ 아두이노 5V
- GND ↔ 아두이노 GND
- TRIG ↔ 아두이노 6
- ECHO ↔ 아두이노 7



# 아두이노를 이용한 초음파 센서 실험

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

  pinMode(6, OUTPUT);      //6 : Trigger
  pinMode(7, INPUT);       //7 : Echo
}

void loop()
{
  //trigger 발생
  digitalWrite(6, LOW) ;
  delayMicroseconds(2) ;
  digitalWrite(6, HIGH) ;
  delayMicroseconds(10) ;
  digitalWrite(6, LOW) ;

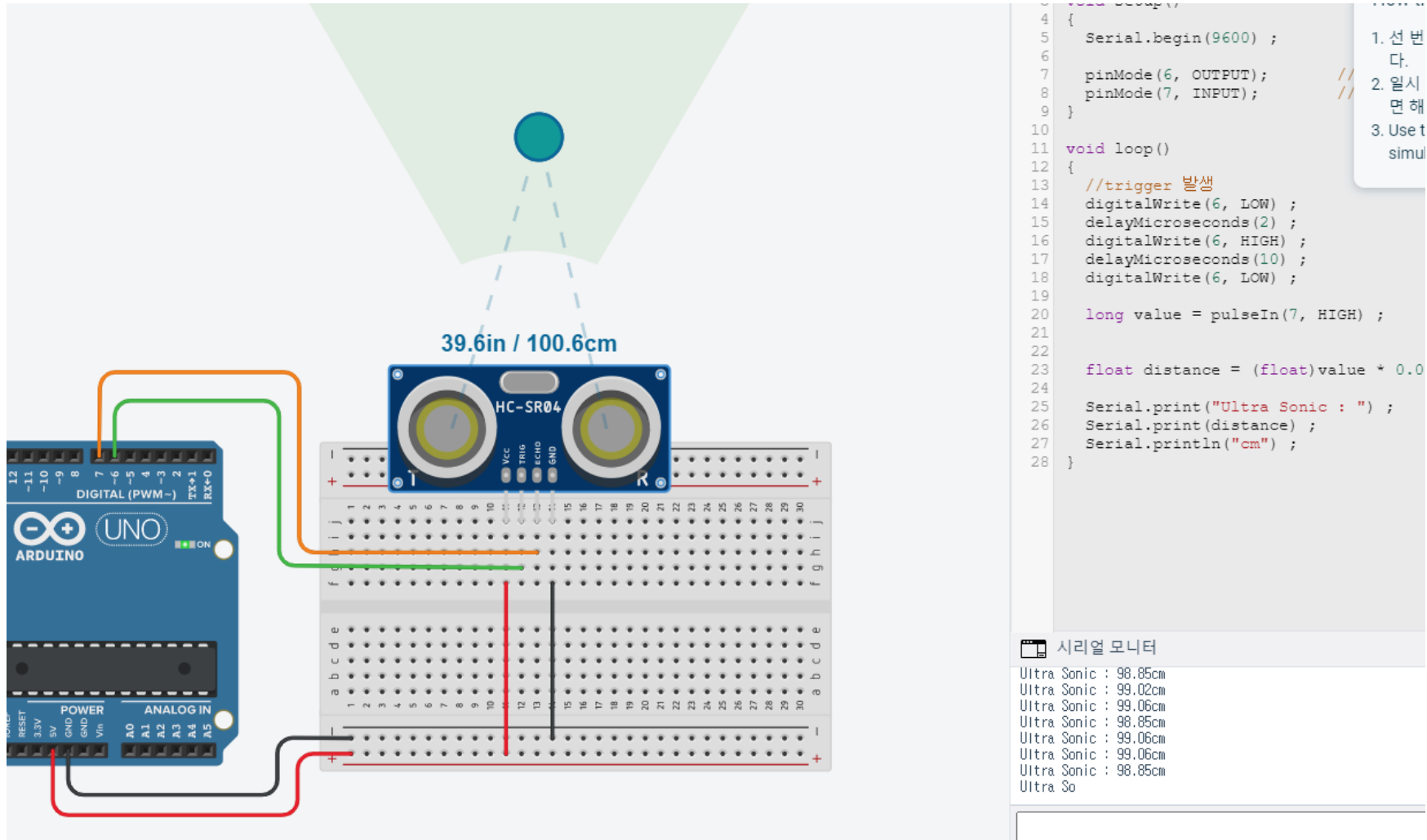
  long value = pulseIn(7, HIGH) ;

  float distance = (float)value * 0.01723 ;

  Serial.print("Ultra Sonic : ") ;
  Serial.print(distance) ;
  Serial.println("cm") ;
}
```

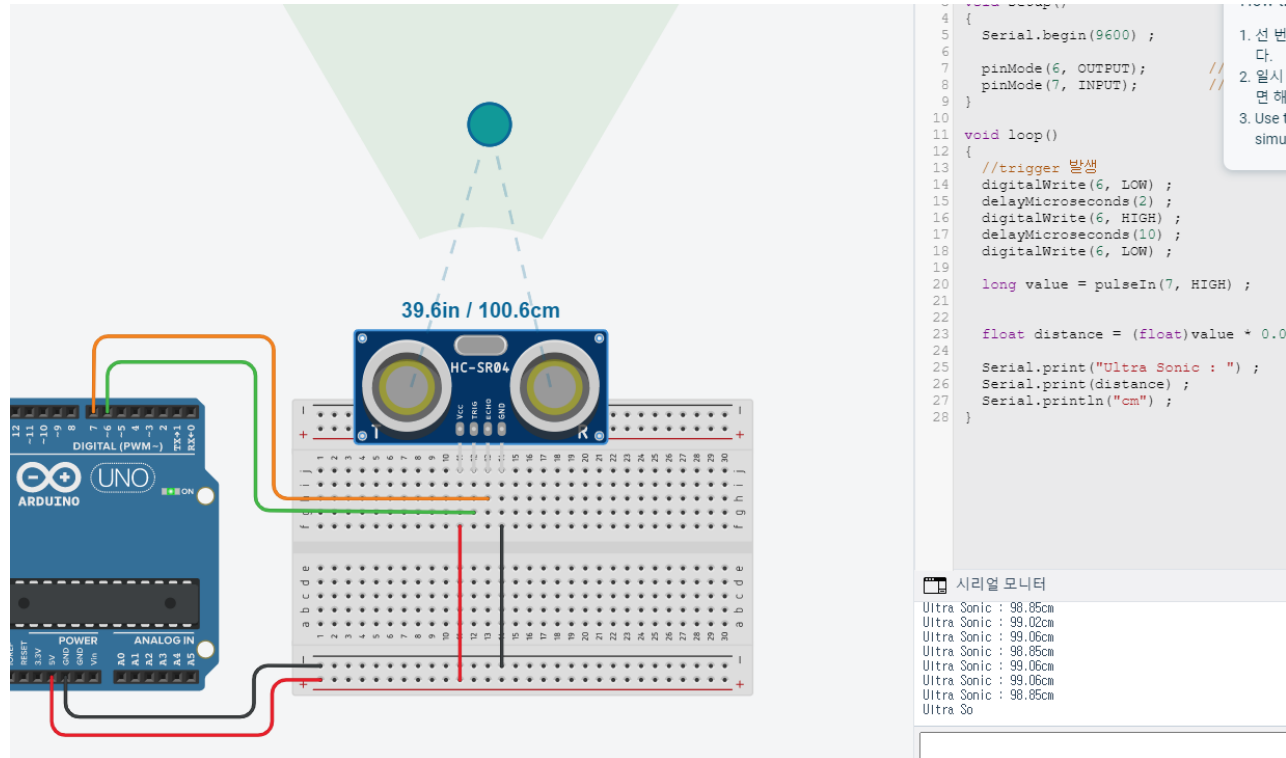
```
1  // C++ code
2  //
3  void setup()
4  {
5    Serial.begin(9600) ;
6
7    pinMode(6, OUTPUT);      //6 : Trigger
8    pinMode(7, INPUT);       //7 : Echo
9  }
10
11 void loop()
12 {
13   //trigger 발생
14   digitalWrite(6, LOW) ;
15   delayMicroseconds(2) ;
16   digitalWrite(6, HIGH) ;
17   delayMicroseconds(10) ;
18   digitalWrite(6, LOW) ;
19
20   long value = pulseIn(7, HIGH) ;
21
22
23   float distance = (float)value * 0.01723 ;
24
25   Serial.print("Ultra Sonic : ") ;
26   Serial.print(distance) ;
27   Serial.println("cm") ;
28 }
```

# 아두이노를 이용한 초음파 센서 실험



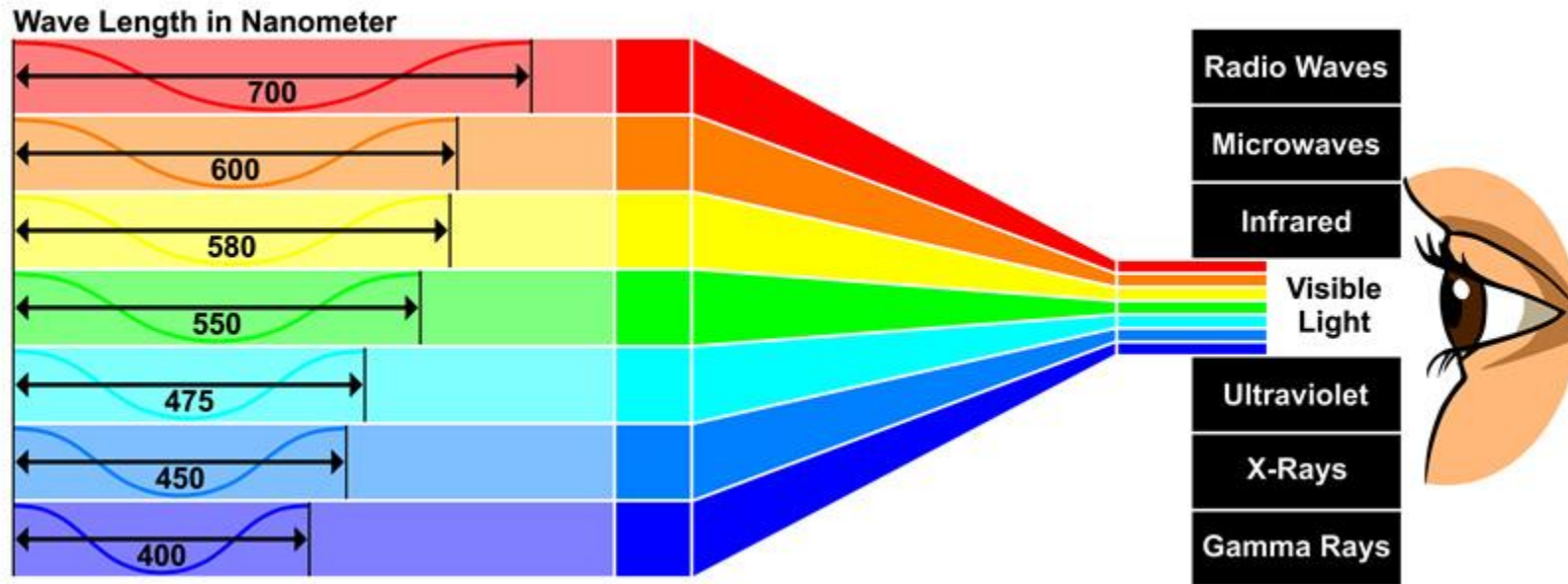
# 아두이노를 이용한 초음파 센서 실험

- QUIZ : 초음파 센서로 10cm이내에 장애물이 감지 되면 LED를 켜고 그렇지 않으면 LED를 끄는 회로와 프로그램을 완성 하시오.



# 광센서를 이용하여 장애물(물체) 인식

- IR(적외선)을 이용하여 장애물 인식



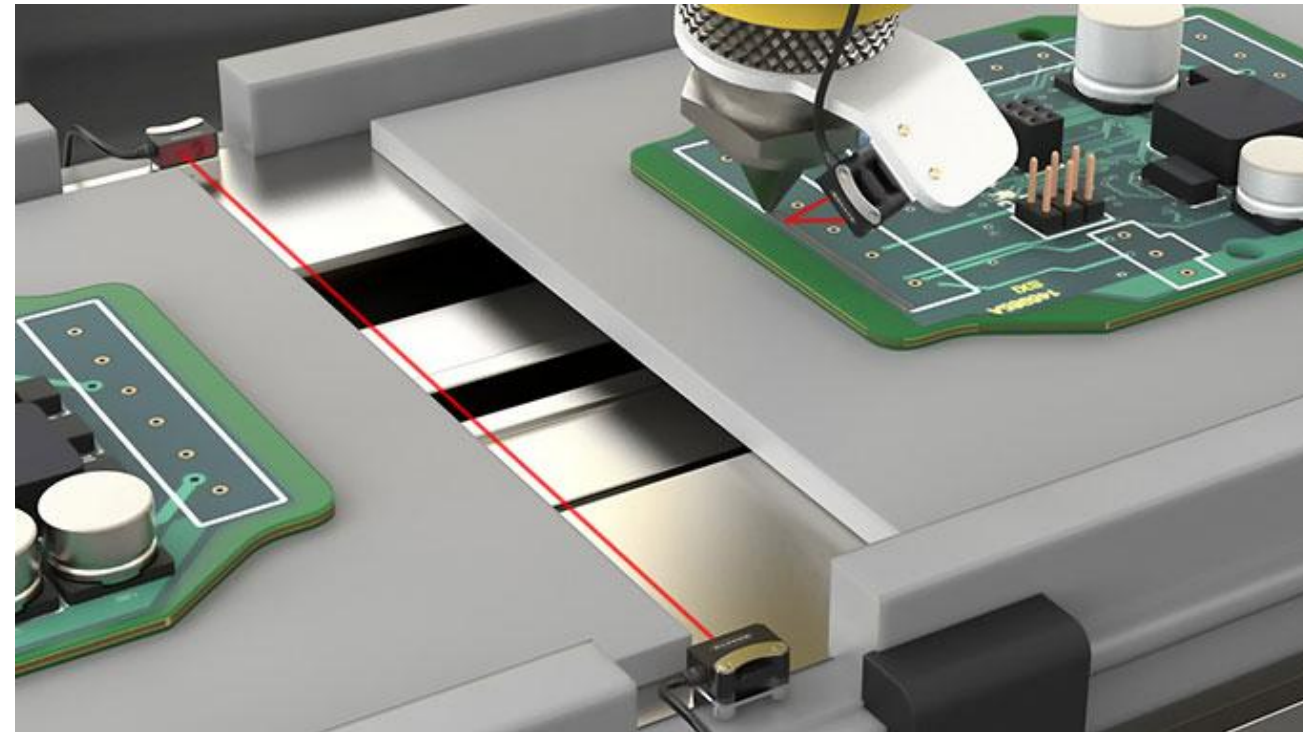


# 적외선 센서를 이용한 Line 추적 무인이동차

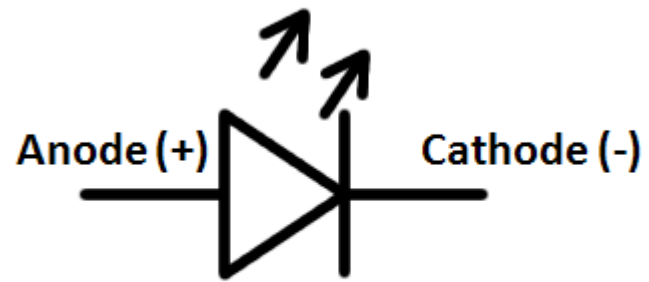
- 물류센터, 스마트팩토리, 스마트팜에서 물류를 자동으로 이동시키기 위해 가장 많이 사용하는 방식



# 광(IR, 레이저)센서를 이용한 생산라인의 생산품 관리

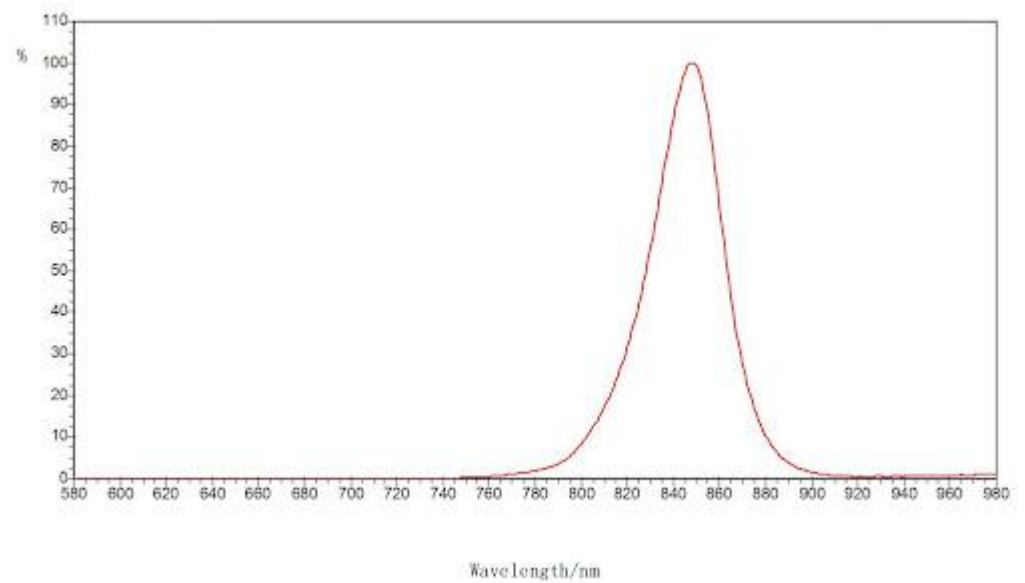


# 적외선 발광 다이오드 및 수신 모듈

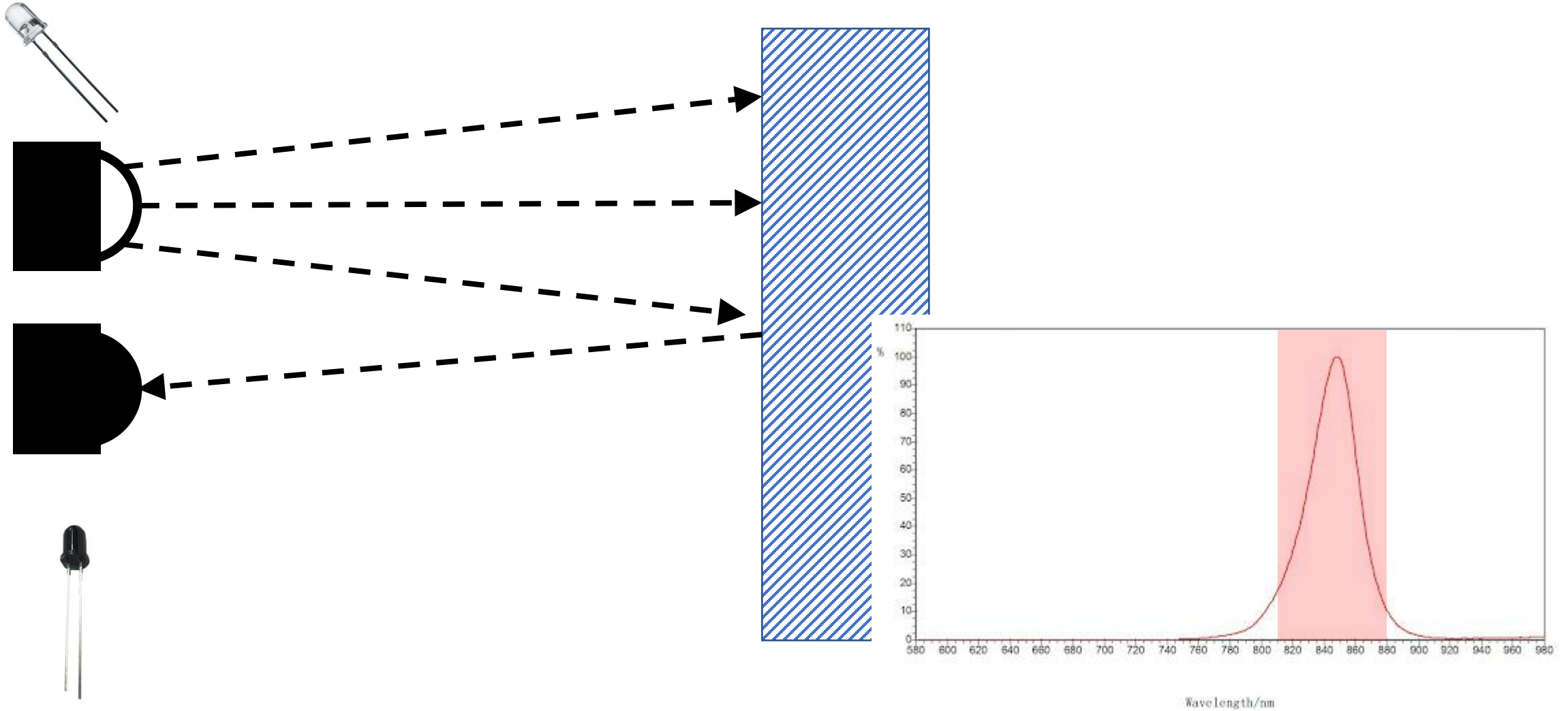


# 적외선 발광 다이오드의 파장

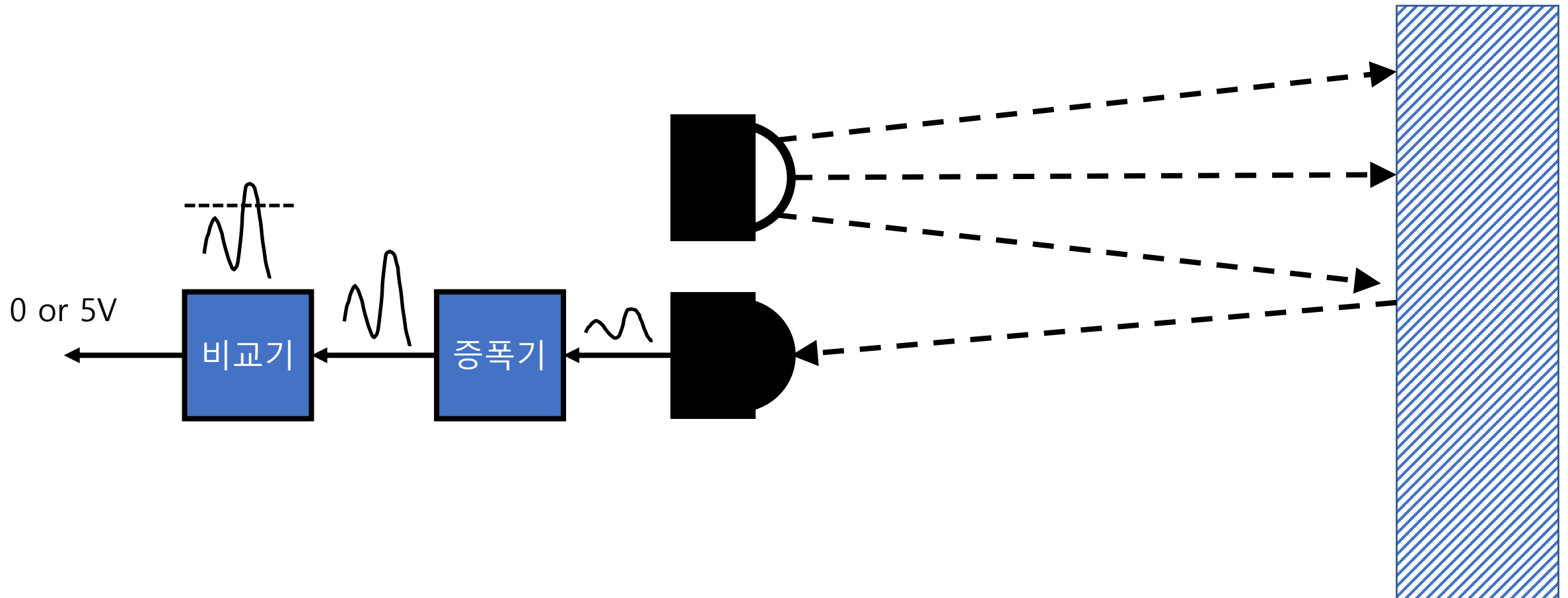
- 840 nm



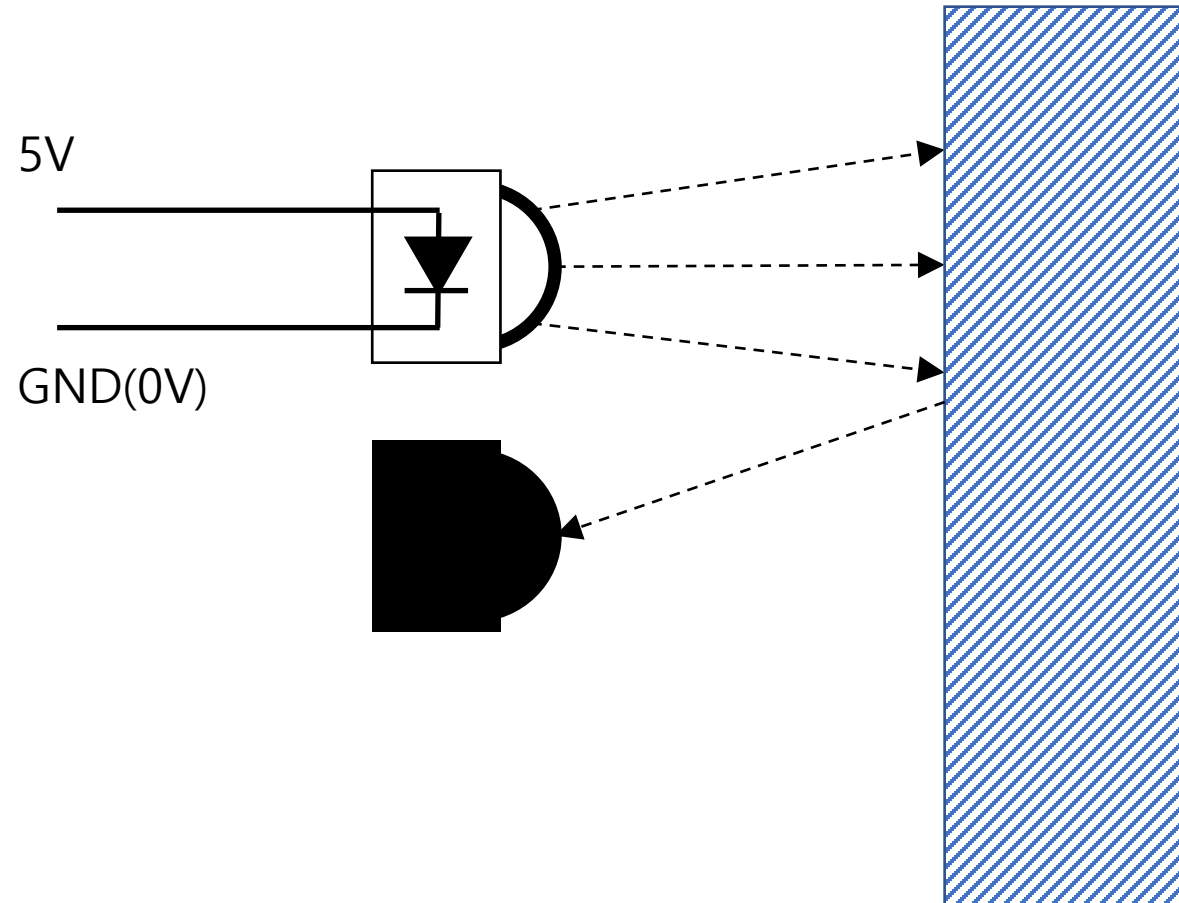
# 적외선 센서를 이용한 장애물 인식



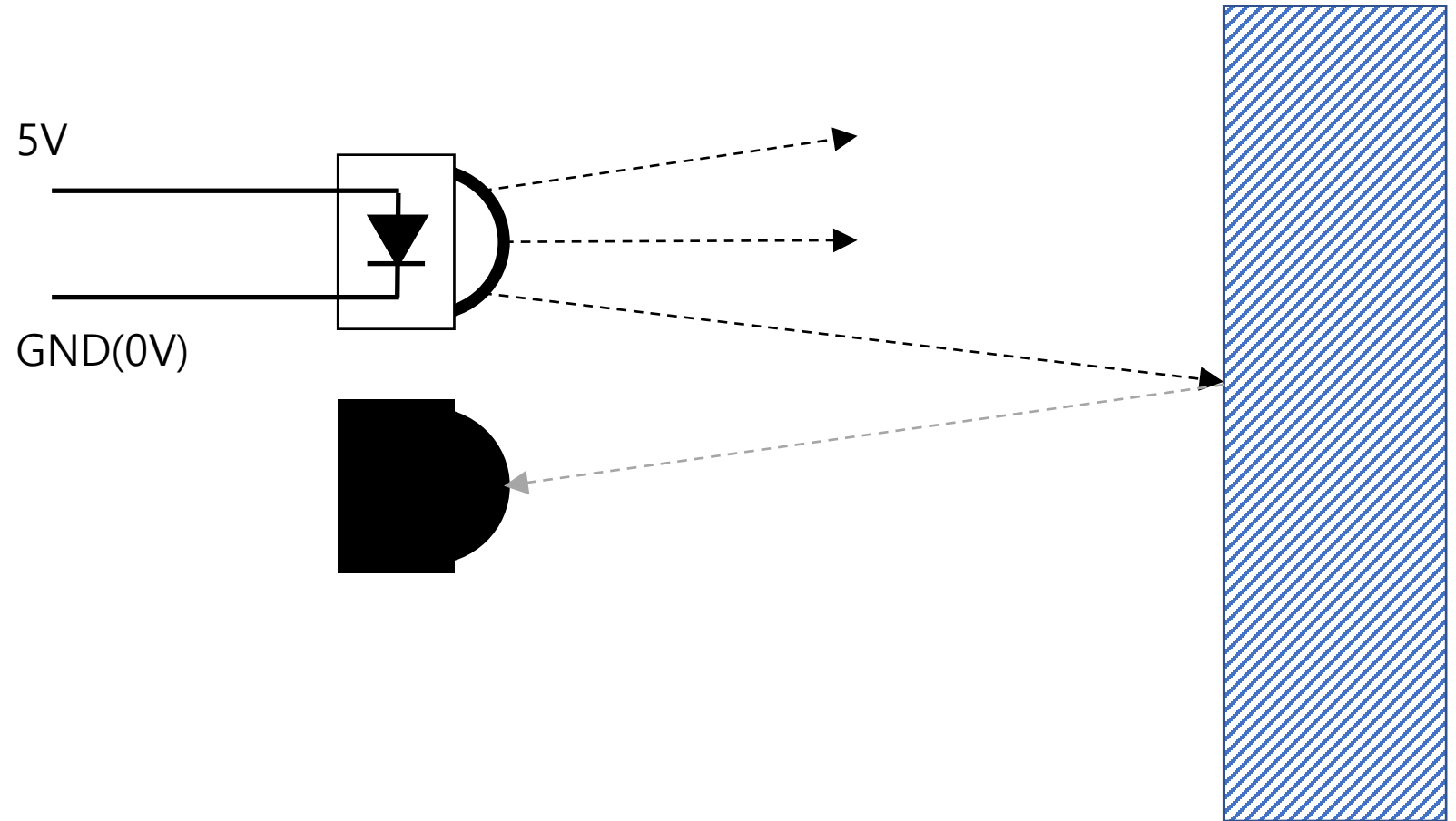
# 적외선 센서를 이용한 장애물 인식



적외선 센서를 이용해 더 먼 거리의 장애물을 인식하려면?

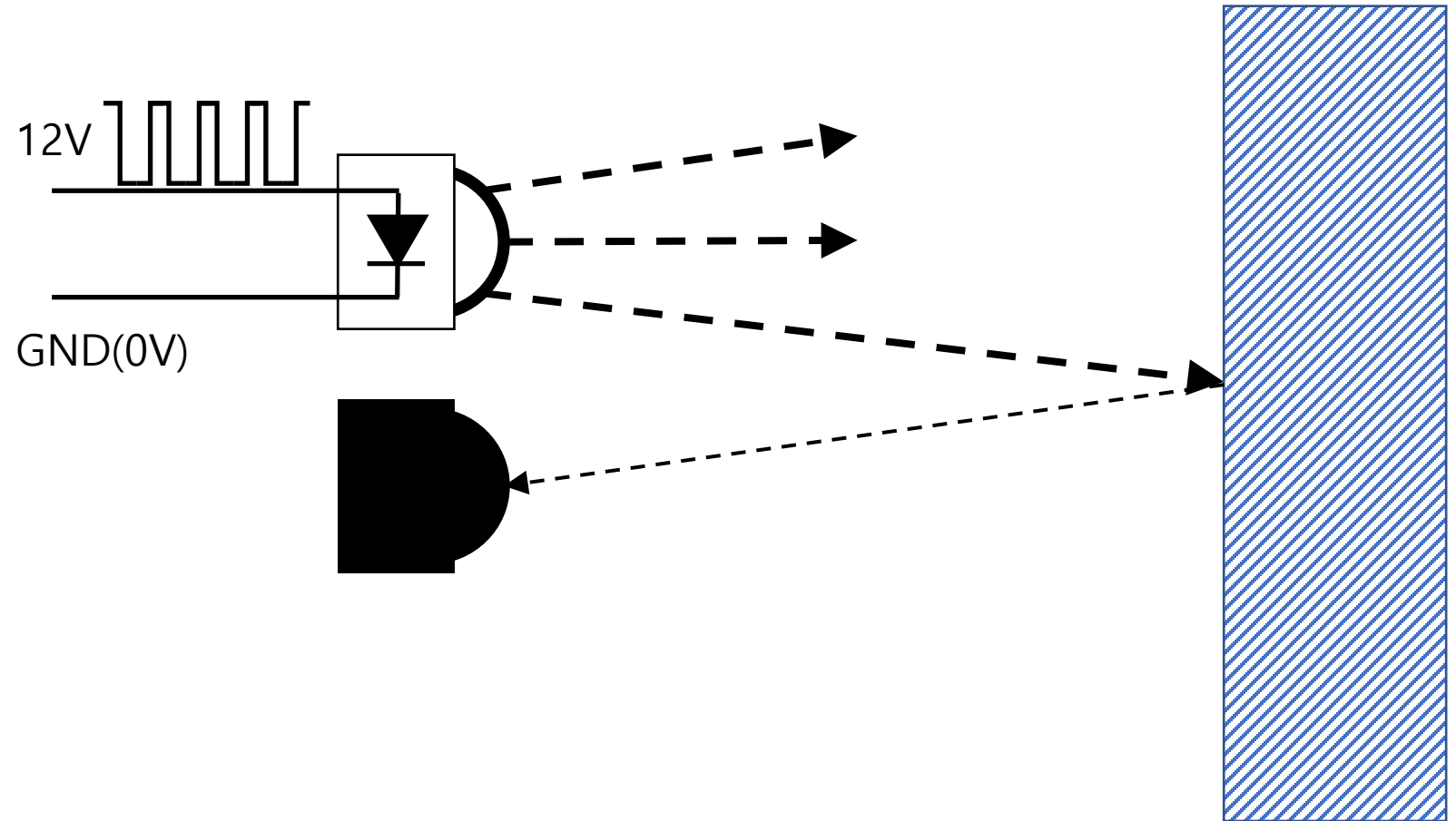


# 적외선 센서를 이용해 더 먼 거리의 장애물을 인식하려면?



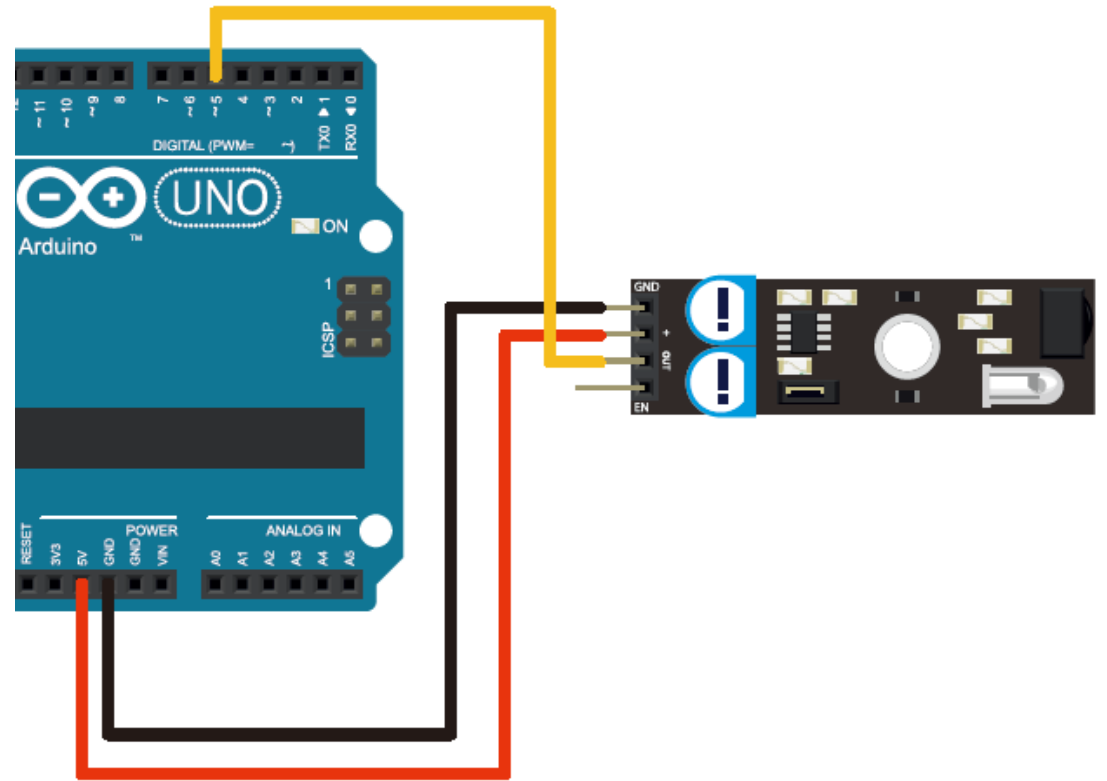
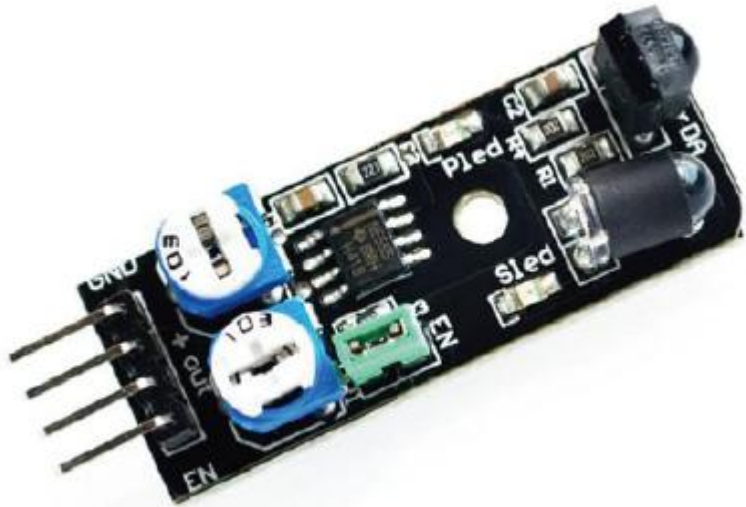


# 적외선 센서를 이용해 더 먼 거리의 장애물을 인식하려면?



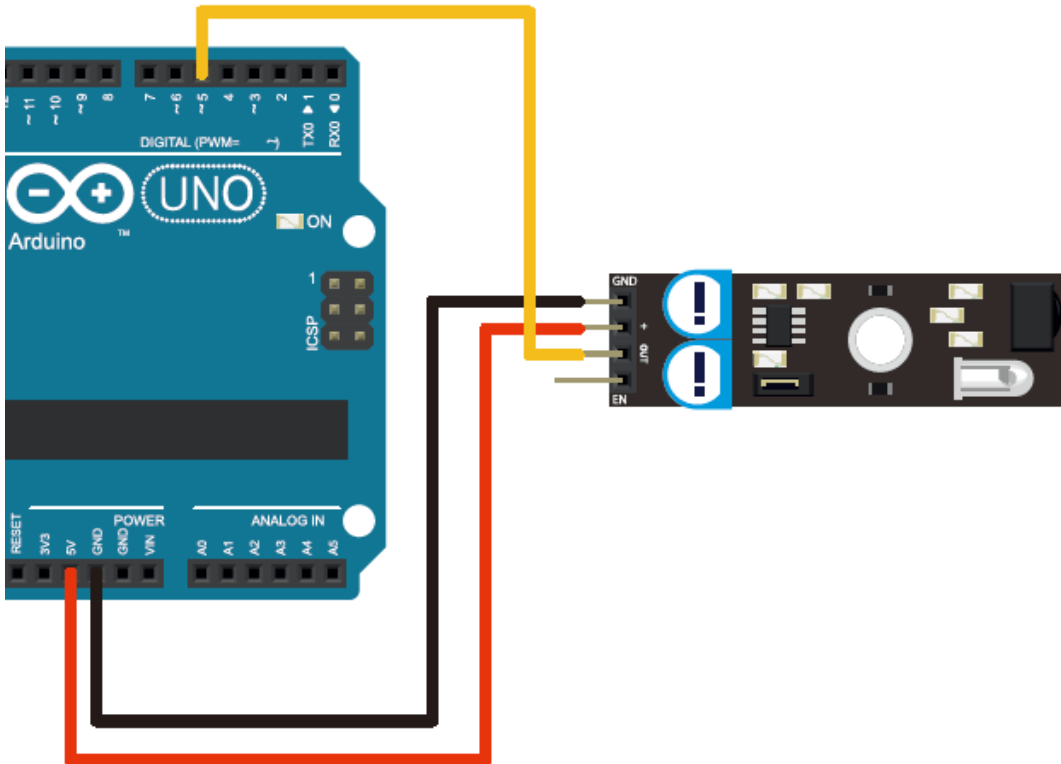
# 적외선 센서 모듈 실험

- 적외선 센서 기반의 장애물 인식 실험



# 적외선 센서 모듈 실험

- 적외선 센서 기반의 장애물 인식 실험

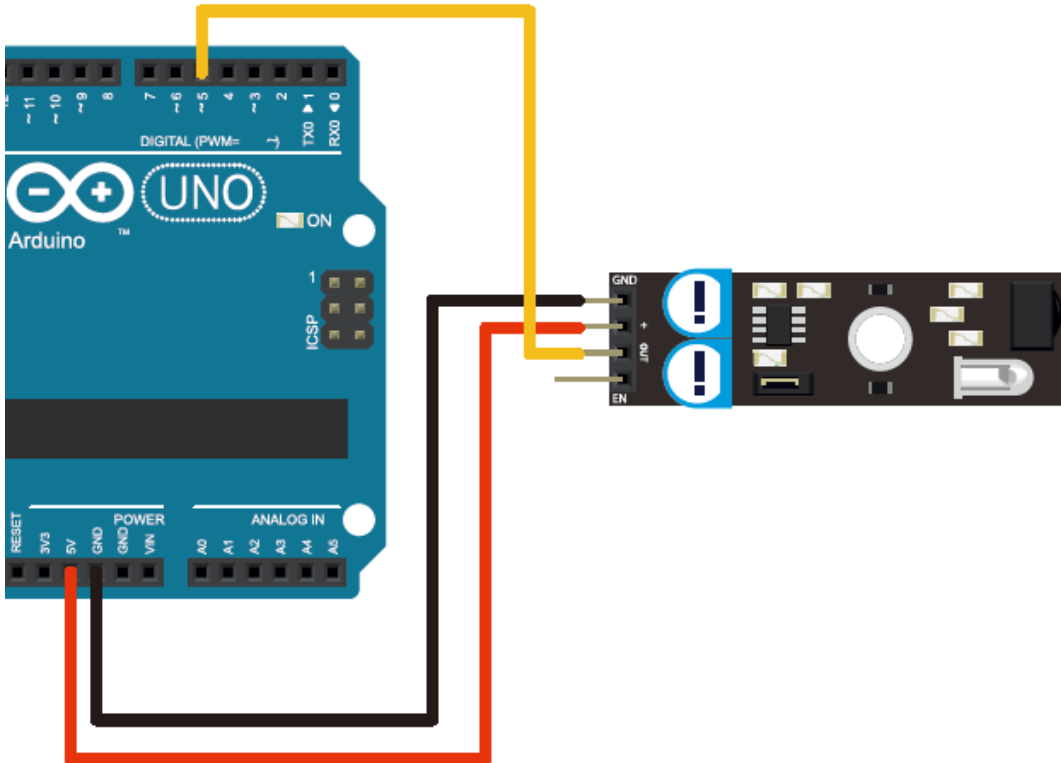


```
void setup()
{
  Serial.begin(9600) ;
  pinMode(5, INPUT) ;
}

void loop()
{
  int sensor = digitalRead(5) ;
  Serial.println(sensor) ;
}
```

# 적외선 센서 모듈 실험

- 적외선 센서 기반의 장애물 인식 실험



```
void setup()
{
  Serial.begin(9600) ;
  pinMode(5, INPUT) ;
}

void loop()
{
  if( digitalRead(5) == LOW )
  {
    Serial.println("Obstacle!") ;
  }
  else
  {
    Serial.println("No Obstacle!") ;
  }

  delay(500) ;
}
```

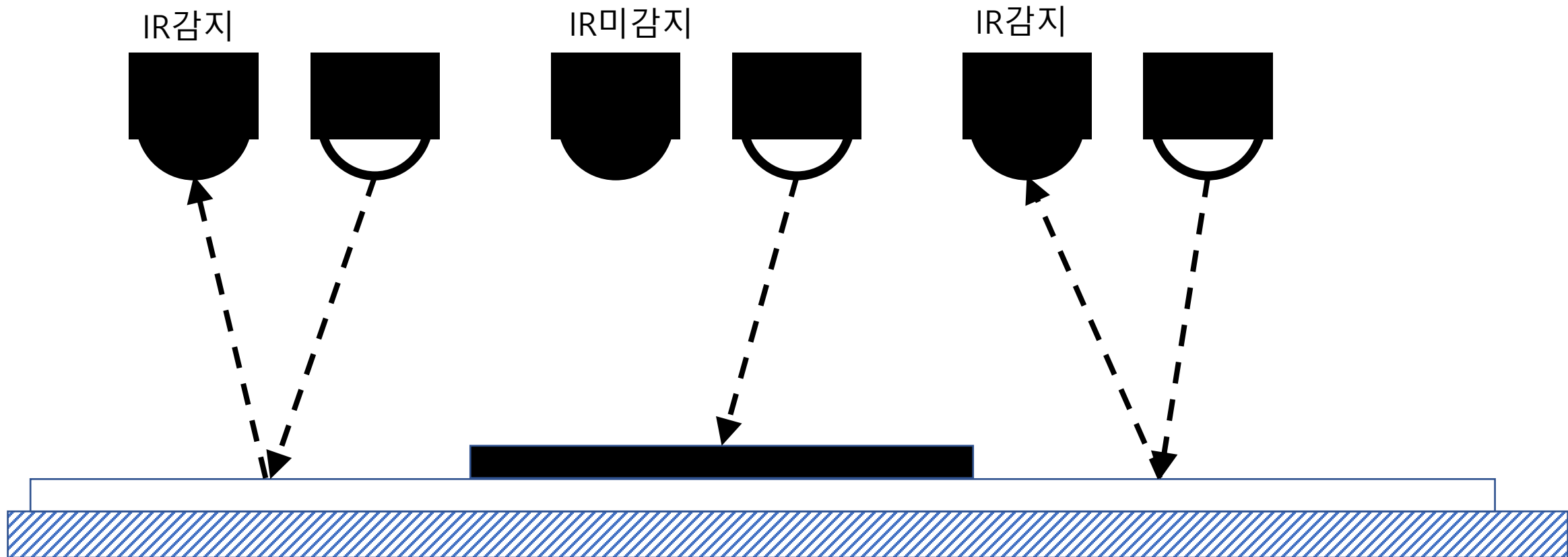
# 적외선 센서를 이용한 Line인식

- 바닥의 검은선을 인식

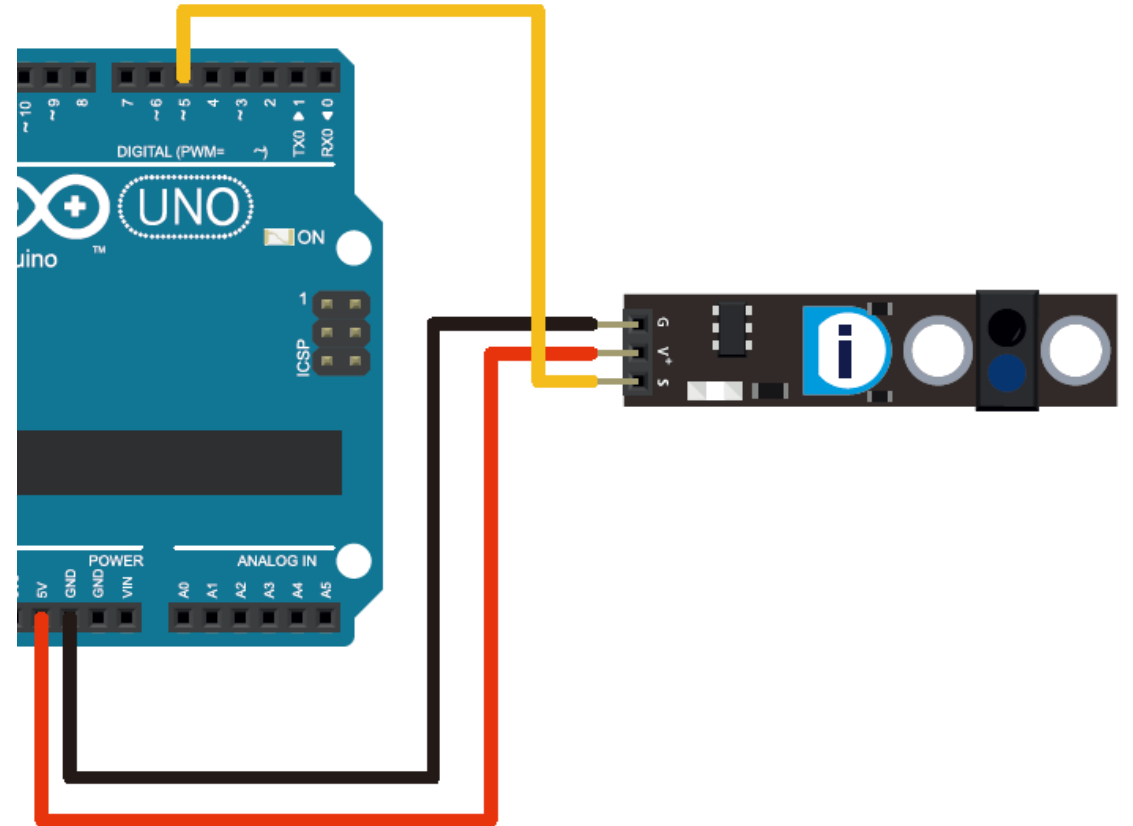


# 적외선 센서를 이용한 Line인식

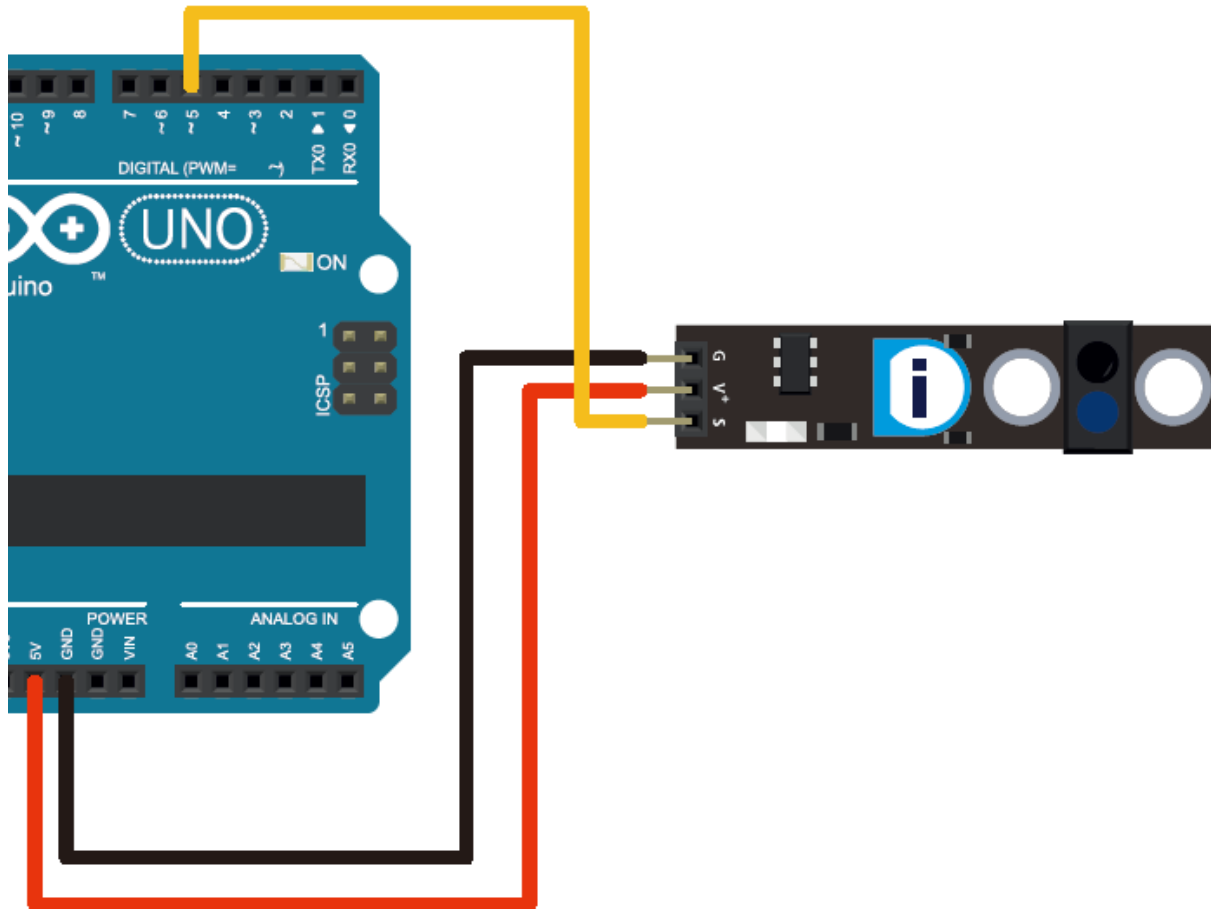
- 여러 개의 IR센서를 이용하여 바닥의 검은선의 위치를 인식



# 적외선 감지 센서 실험



# 적외선 라인감지 센서 실험



```
void setup()
{
  Serial.begin(9600) ;
  pinMode(5, INPUT) ;
}

void loop()
{
  if( digitalRead(5) == HIGH )
  {
    Serial.println("Check Line!") ;
  }
  else
  {
    Serial.println("Line") ;
  }

  delay(500) ;
}
```



# 불꽃감지센서 (Flame sensor)

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

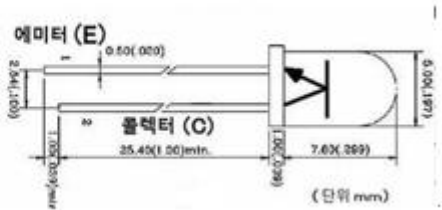
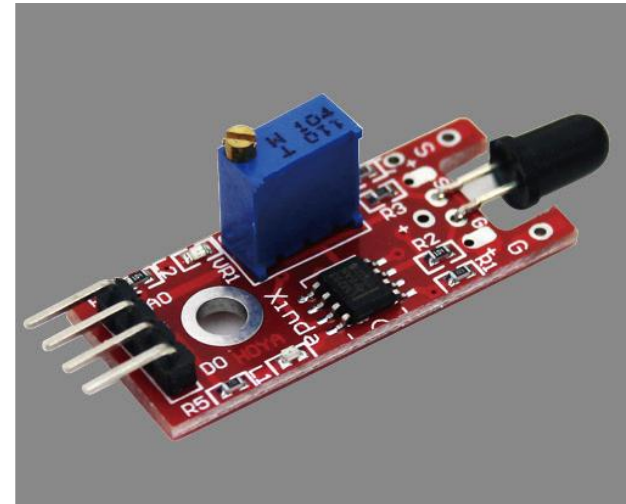
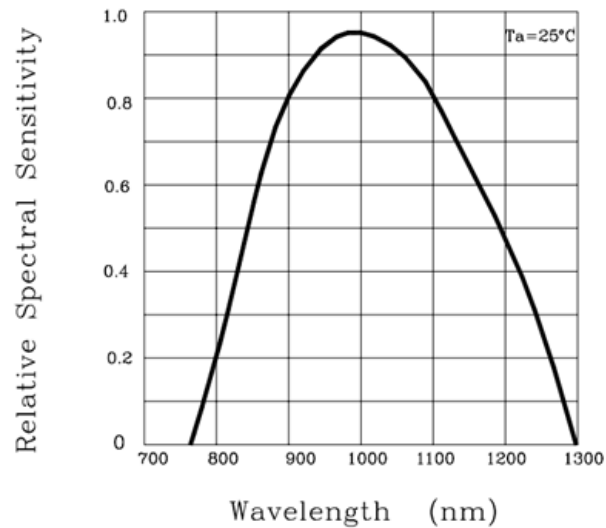
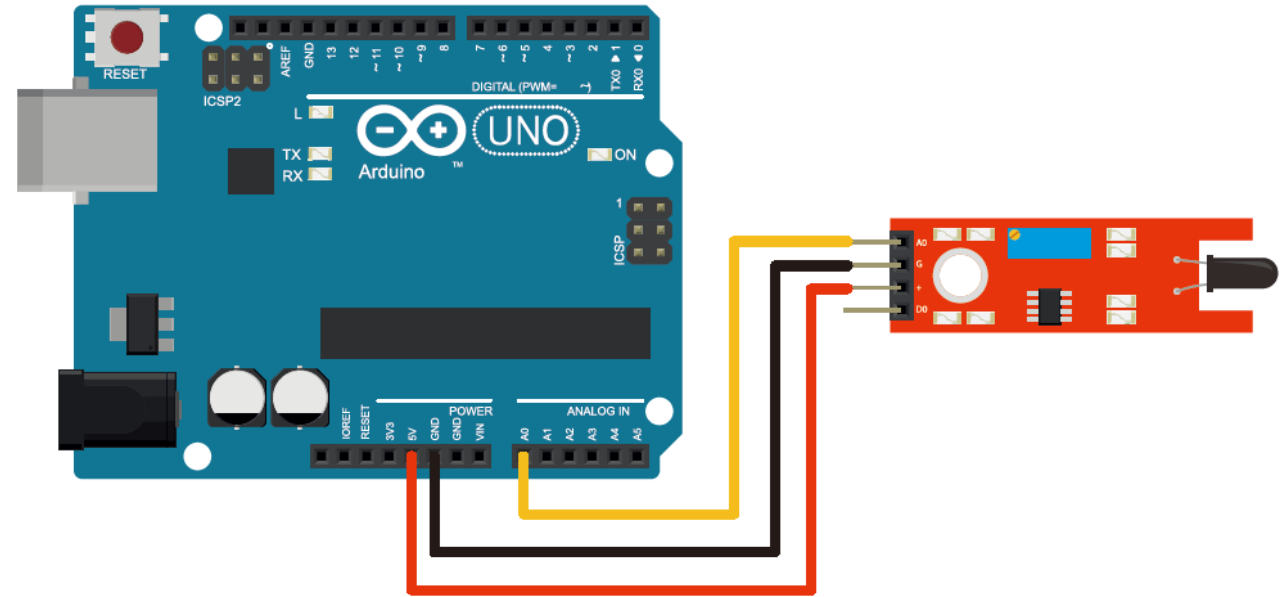


Fig. 5 Spectral Sensitivity



# 불꽃감지센서 (Flame sensor)

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



# 불꽃감지센서 (Flame sensor)

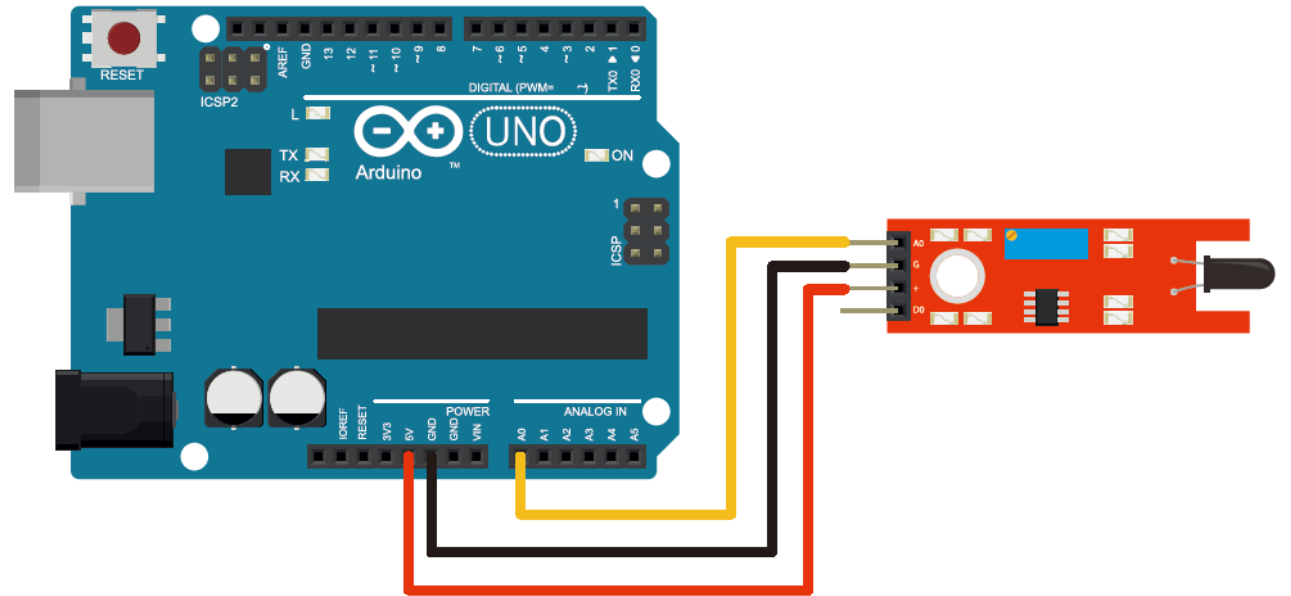
- 불꽃 감지 아두이노 실험 코드 작성

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

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

  Serial.println(analog_value);

  delay(100);
}
```



# 부저 실험

## • 부저(소리) 출력 실험

- 능동부저:전원을공급하면단음(빠)소리가출력
- 수동부저:진동을만들어특정주파수의소리를출력(다양한소리를출력할수있음,멜로디)



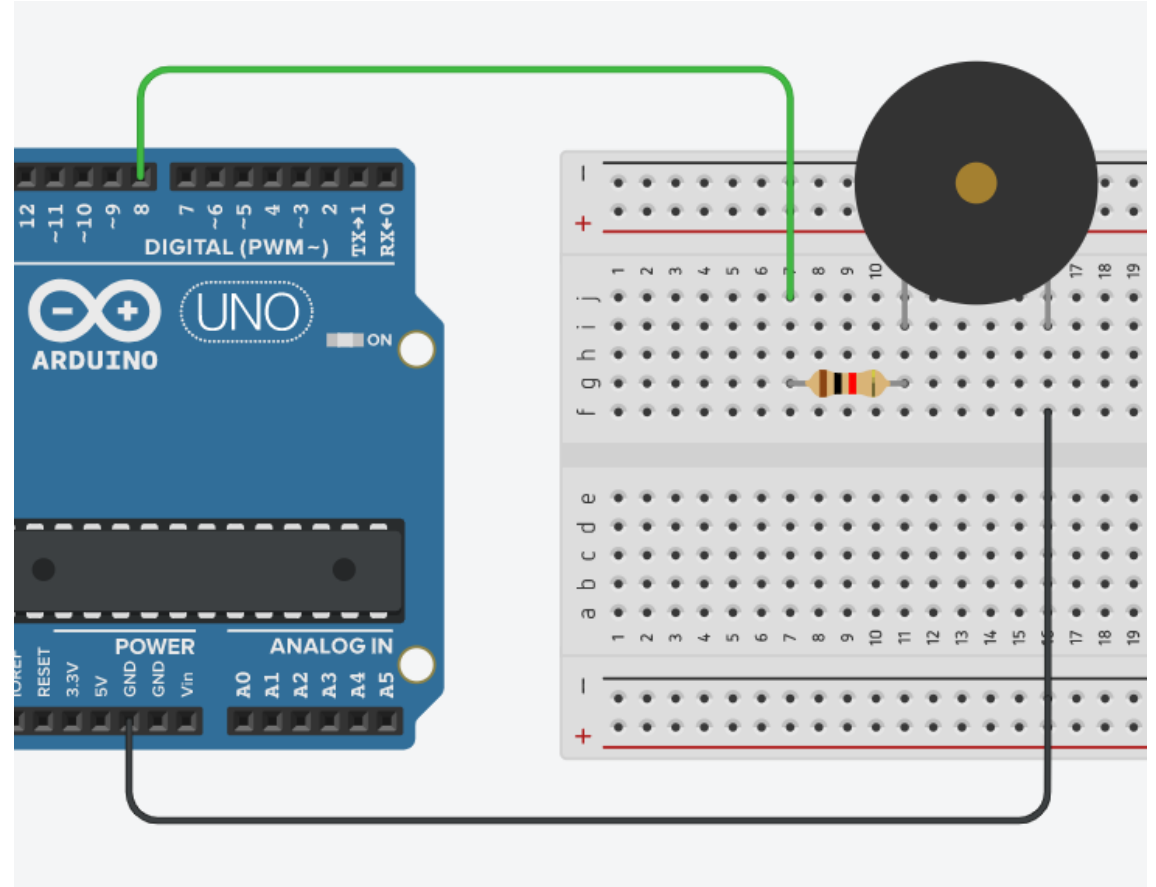
( 단위 : Hz )

음계 \ 옥타브	1	2	3	4	5	6	7	8
C(도)	32.7032	65.4064	130.8128	261.6256	523.2511	1046.502	2093.005	4186.009
C#	34.6478	69.2957	138.5913	277.1826	554.3653	1108.731	2217.461	4434.922
D(레)	36.7081	73.4162	146.8324	293.6648	587.3295	1174.659	2349.318	4698.636
D#	38.8909	77.7817	155.5635	311.1270	622.2540	1244.508	2489.016	4978.032
E(미)	41.2034	82.4069	164.8138	329.6276	659.2551	1318.510	2637.020	5274.041
F(파)	43.6535	87.3071	174.6141	349.2282	698.4565	1396.913	2793.826	5587.652
F#	46.2493	92.4986	184.9972	369.9944	739.9888	1479.978	2959.955	5919.911
G(솔)	48.9994	97.9989	195.9977	391.9954	783.9909	1567.982	3135.963	6271.927
G#	51.9130	103.8262	207.6523	415.3047	830.6094	1661.219	3322.438	6644.875
A(라)	55.0000	110.0000	220.0000	440.0000	880.0000	1760.000	3520.000	7040.000
A#	58.2705	116.5409	233.0819	466.1638	932.3275	1864.655	3729.310	7458.620
B(시)	61.7354	123.4708	246.9417	493.8833	987.7666	1975.533	3951.066	7902.133

- 도:261.6256Hz
- 레:293.1826Hz
- 미:329.6276 Hz
- 파:349.2282 Hz
- 솔:391.9954 Hz
- 라:440.0000 Hz
- 시:466.1638 Hz
- 도:523.2511 Hz

# 부저 실험

- 부저(소리) 출력 실험
  - 부저 + <> 아두이노 8번핀
  - 부저 - <> 아두이노 GND



# 부저 + LED 실험

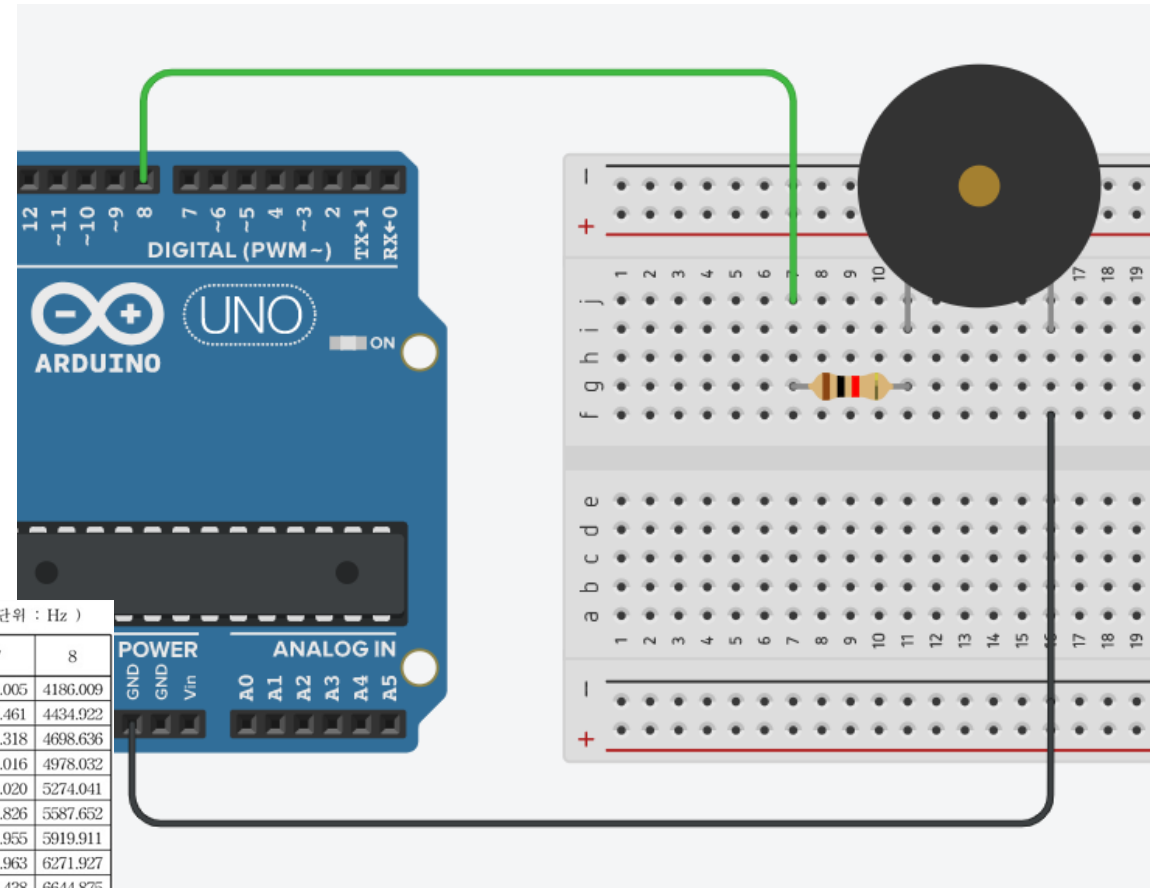
- 부저(소리) 출력 실험

```
void setup()
{
  pinMode(8, OUTPUT);
}

void loop()
{
  tone(8, 262, 500);
  delay(500);
}
```

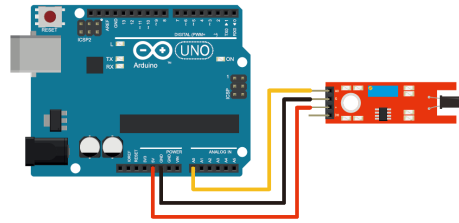
( 단위 : Hz )

음계 \ 옥타브	1	2	3	4	5	6	7	8
C(도)	32.7032	65.4064	130.8128	261.6256	523.2511	1046.502	2093.005	4186.009
C#	34.6478	69.2957	138.5913	277.1826	554.3653	1108.731	2217.461	4434.922
D(레)	36.7081	73.4162	146.8324	293.6648	587.3295	1174.659	2349.318	4698.636
D#	38.8909	77.7817	155.5635	311.1270	622.2540	1244.508	2489.016	4978.032
E(미)	41.2034	82.4069	164.8138	329.6276	659.2551	1318.510	2637.020	5274.041
F(파)	43.6535	87.3071	174.6141	349.2282	698.4565	1396.913	2793.826	5587.652
F#	46.2493	92.4986	184.9972	369.9944	739.9888	1479.978	2959.955	5919.911
G(솔)	48.9994	97.9989	195.9977	391.9954	783.9909	1567.982	3135.963	6271.927
G#	51.9130	103.8262	207.6523	415.3047	830.6094	1661.219	3322.438	6644.875
A(라)	55.0000	110.0000	220.0000	440.0000	880.0000	1760.000	3520.000	7040.000
A#	58.2705	116.5409	233.0819	466.1638	932.3275	1864.655	3729.310	7458.620
B(시)	61.7354	123.4708	246.9417	493.8833	987.7666	1975.533	3951.066	7902.133



# 불꽃감지센서를 이용한 화재감지 응용

- 불꽃이 감지 되면 자동으로 경고를 발생시키자!



# QUIZ

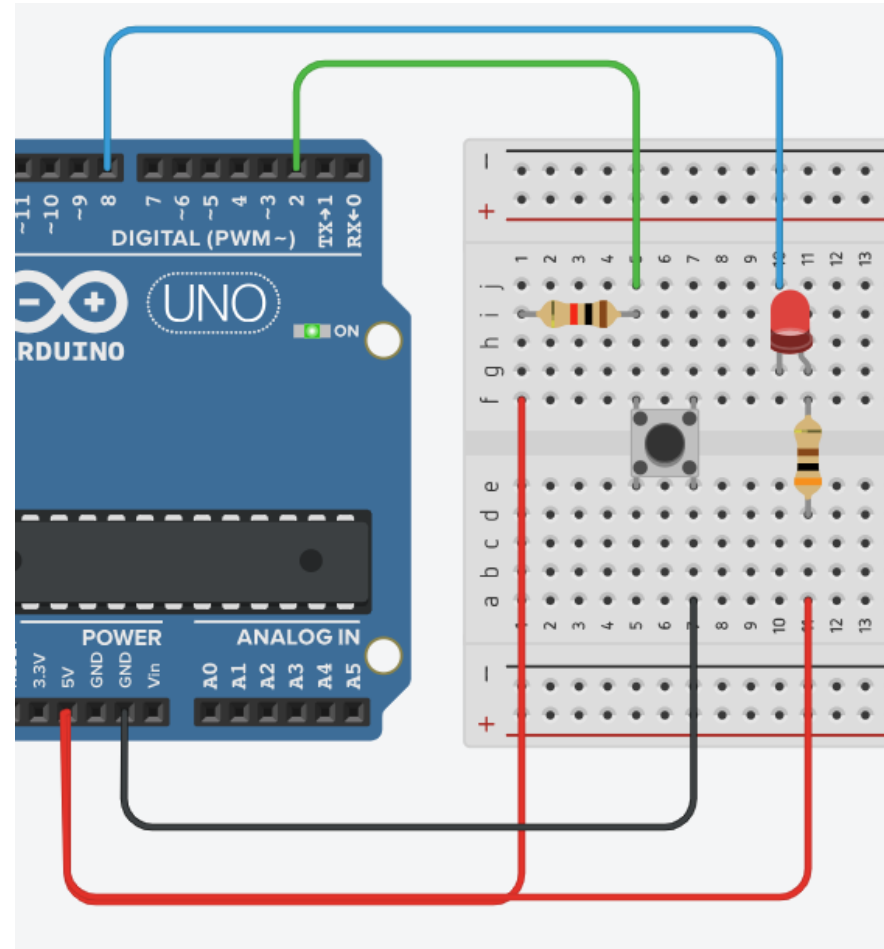
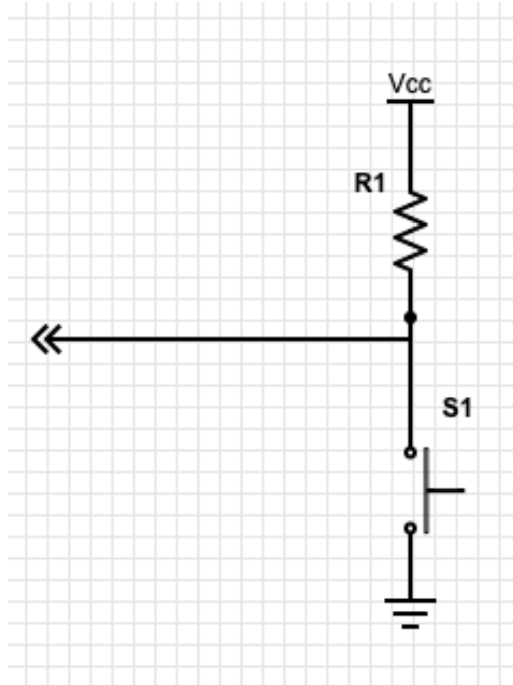


화재가 감지되면 경고음(부저)과 비상등을 켜세요!



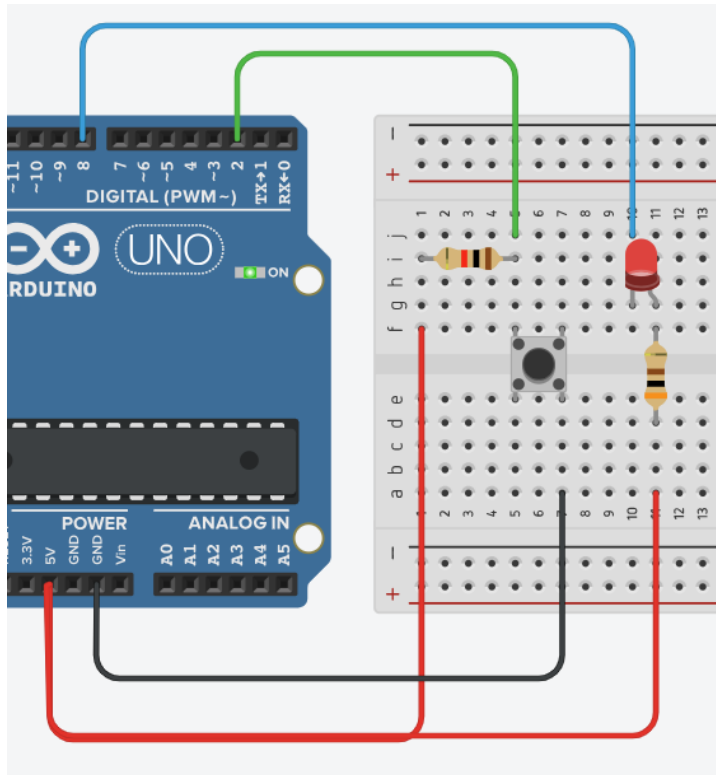
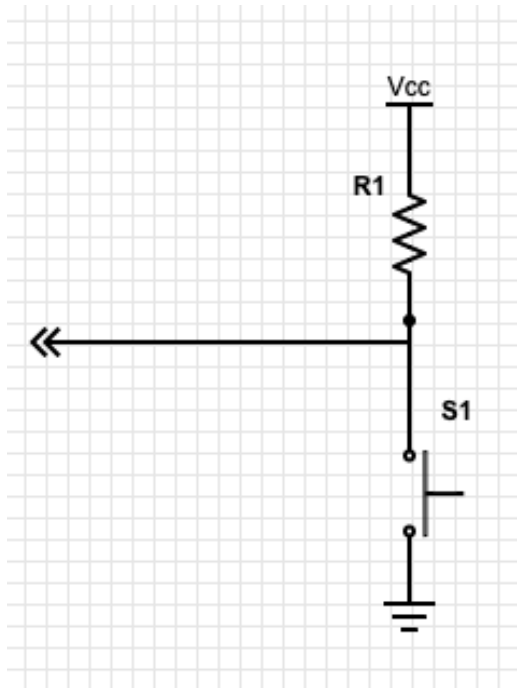
# 외부 인터럽트(External Interrupt)

- 풀링 vs 인터럽트



# 외부 인터럽트(External Interrupt)

- 풀링 vs 인터럽트



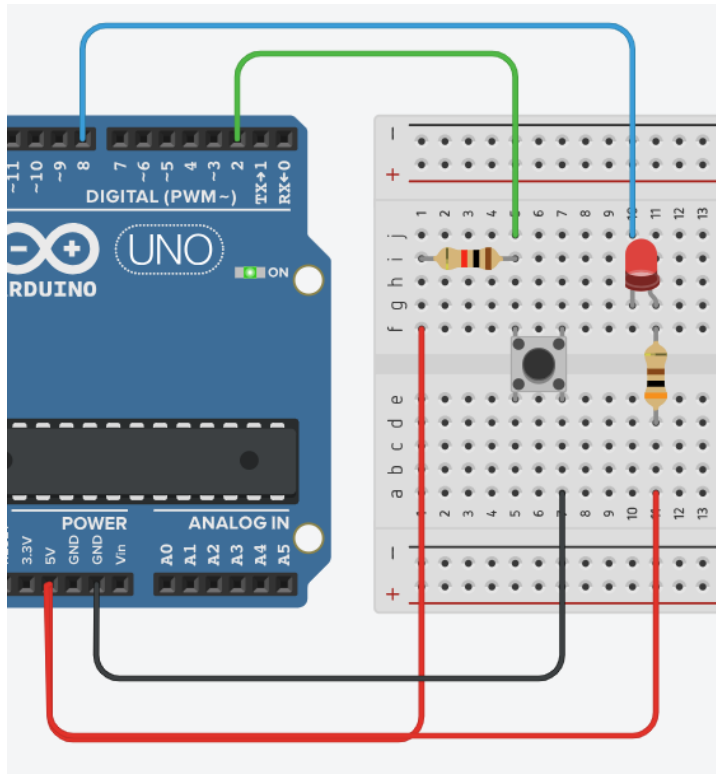
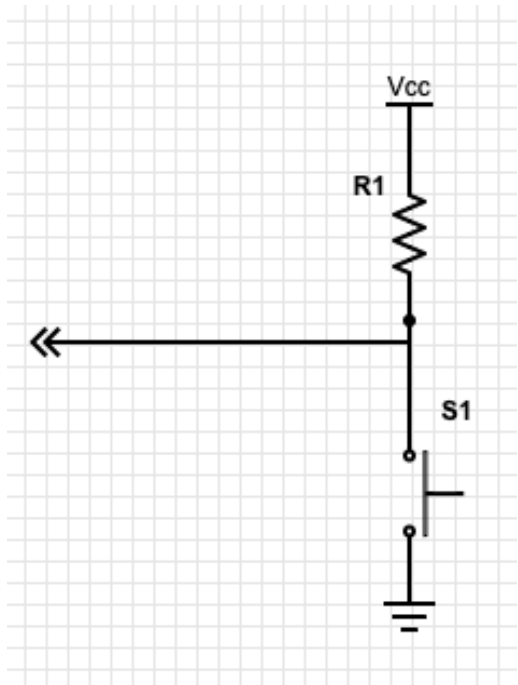
```
void setup()
{
  pinMode(2, INPUT) ;
  pinMode(8, OUTPUT) ;
}

void loop()
{
  int input = digitalRead(2) ;

  if( input == 0 )
  {
    digitalWrite(8, 0) ;
  }
  else
  {
    digitalWrite(8, 1) ;
  }
}
```

# 외부 인터럽트(External Interrupt)

- 풀링 vs 인터럽트



```
void setup()
{
  pinMode(2, INPUT) ;
  pinMode(8, OUTPUT) ;

  Serial.begin(9600) ;
}

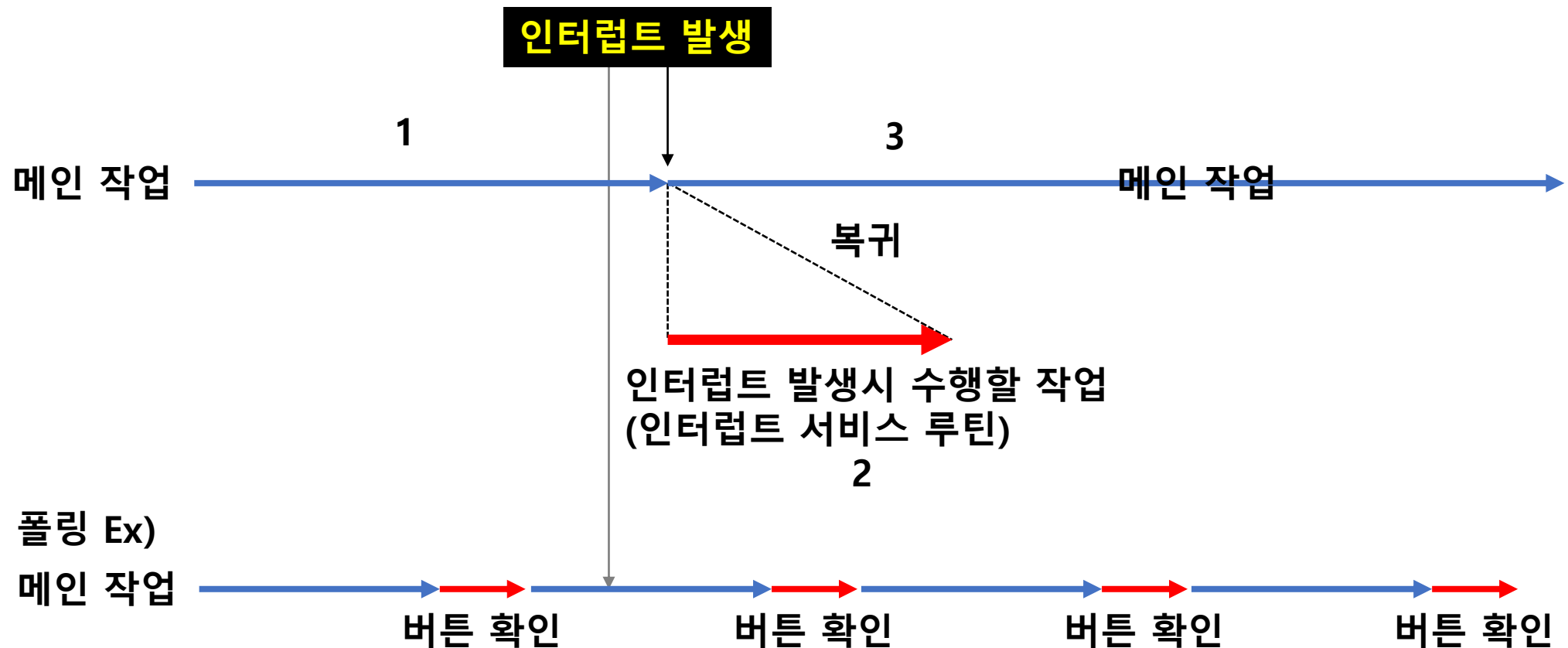
void loop()
{
  digitalWrite(8, 0) ;
  delay(1000) ;

  digitalWrite(8, 1) ;
  delay(1000) ;

  int input = digitalRead(2) ;
  if( input == 0 )
  {
    Serial.println("key") ;
  }
}
```

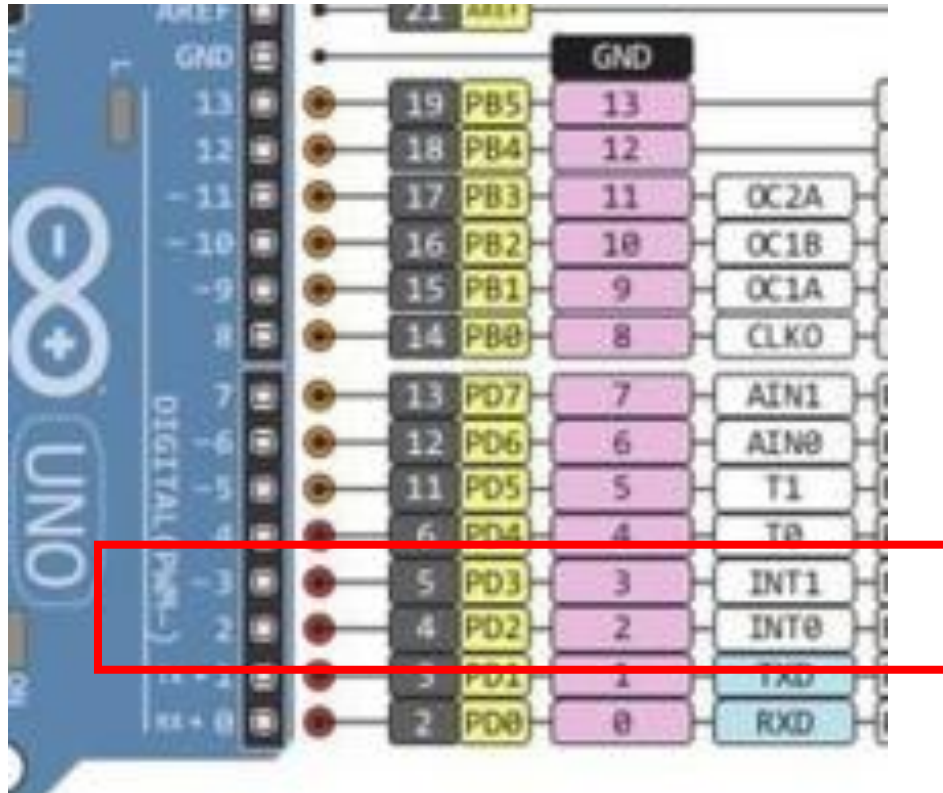
# 외부 인터럽트(External Interrupt)

- 폴링 vs 인터럽트



# 외부 인터럽트(External Interrupt)

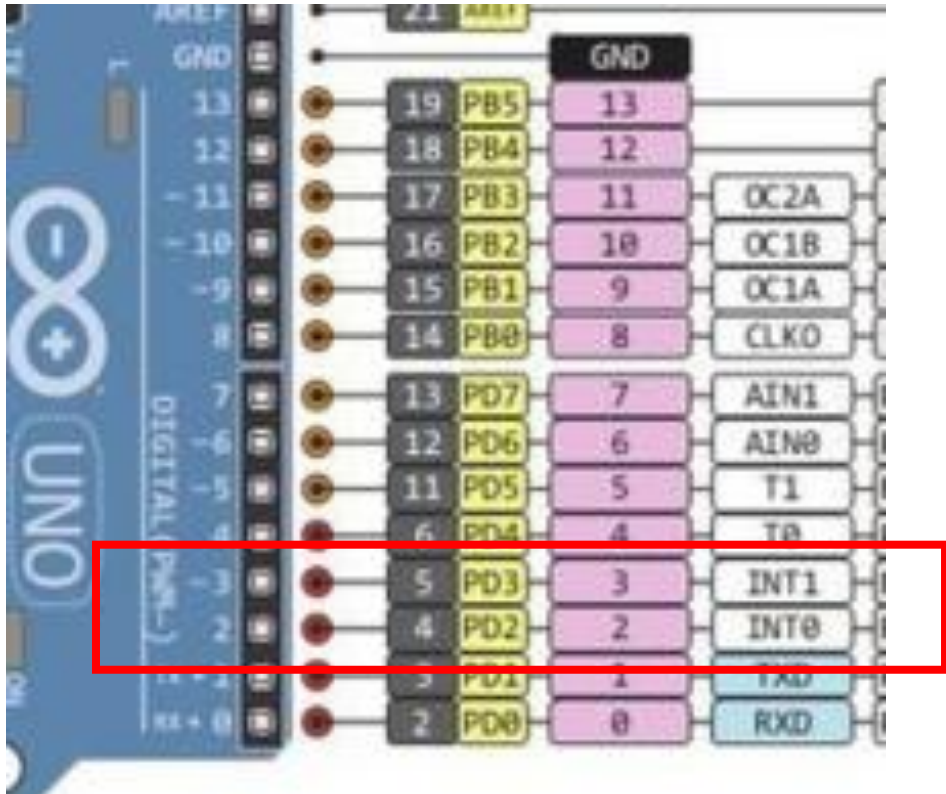
- 폴링 vs 인터럽트



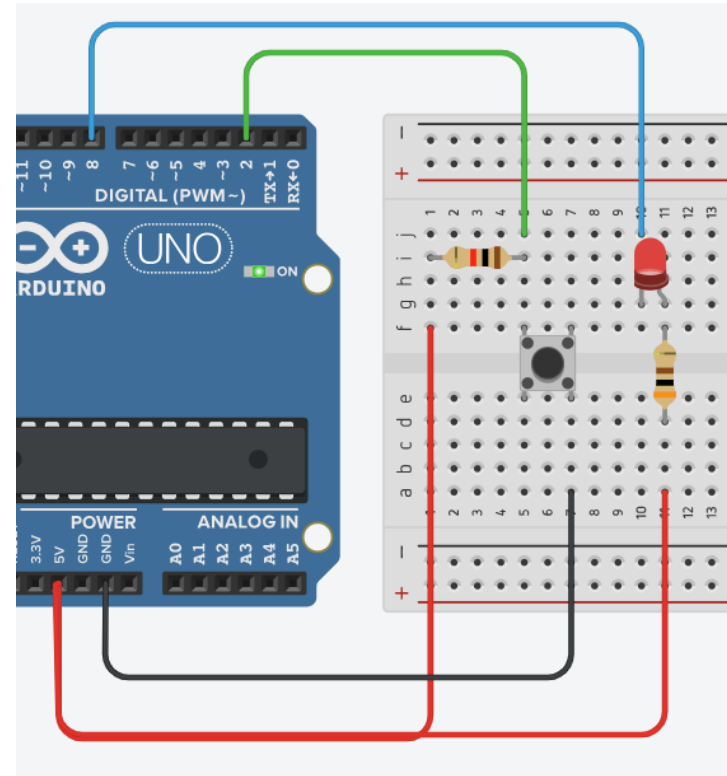
- INT1 : Interrupt #1
- INT0 : Interrupt #0

# 외부 인터럽트(External Interrupt)

- 폴링 vs 인터럽트



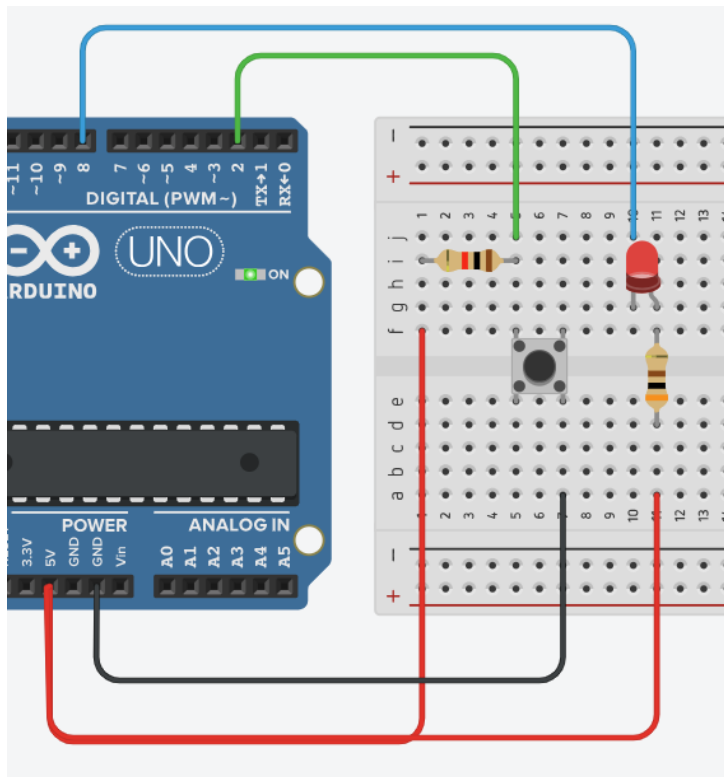
2 → INT0 : Interrupt #0



# 외부 인터럽트(External Interrupt)

## • 폴링 vs 인터럽트

2 → INT0 : Interrupt #0



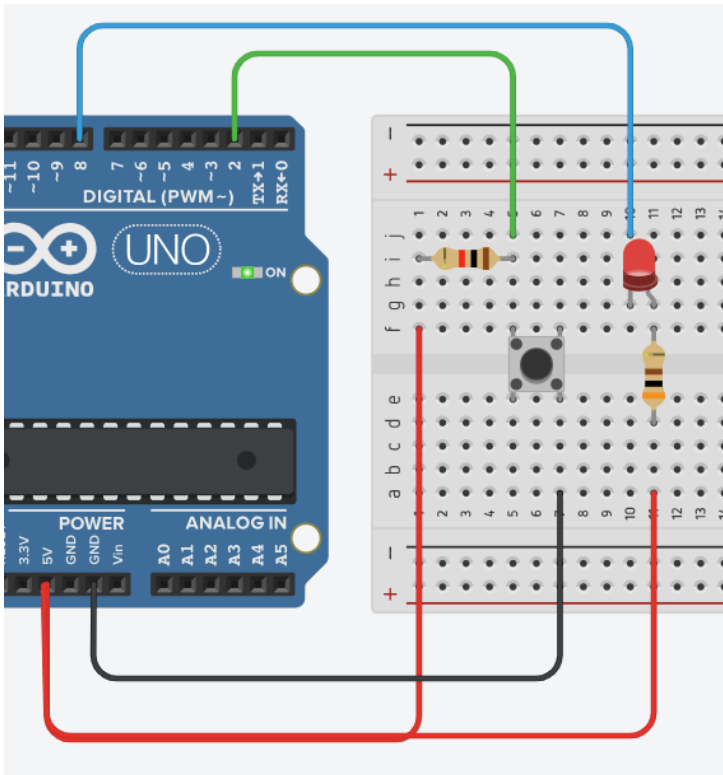
### 인터럽트 발동 조건 (mode)

모드	상태
LOW	핀이 LOW일때
CHANGE	LOW->HIGH or HIGH->LOW로 변할 때
RISING	LOW ->HIGH일때
FALLING	HIGH -> LOW일때
HIGH	핀이 HIGH일때

# 외부 인터럽트(External Interrupt)

- 폴링 vs 인터럽트

2 → INT0 : Interrupt #0



```
attachInterrupt( digitalPinToInterrupt(핀번호), 서비스루틴함수명, 모드 );
```

## 인터럽트 발동 조건 (mode)

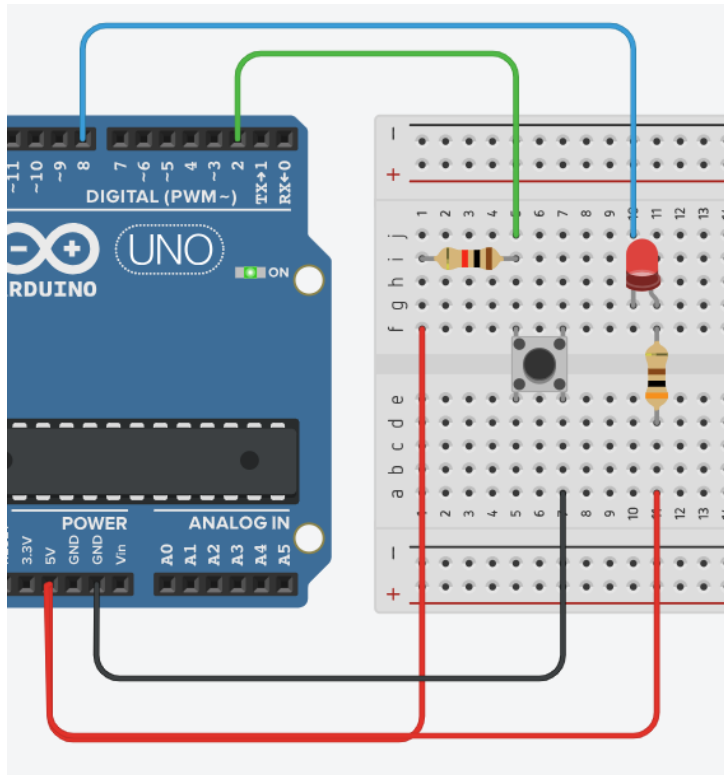
모드	상태
LOW	핀이 LOW일때
CHANGE	LOW->HIGH or HIGH->LOW로 변할 때
RISING	LOW -> HIGH일때
FALLING	HIGH -> LOW일때
HIGH	핀이 HIGH일때



# 외부 인터럽트(External Interrupt)

## • 폴링 vs 인터럽트

2 → INT0 : Interrupt #0



```
attachInterrupt( digitalPinToInterrupt(2), ExINT, FALLING );
```

```
attachInterrupt( digitalPinToInterrupt(핀번호), 서비스루틴함수명, 모드 );
```

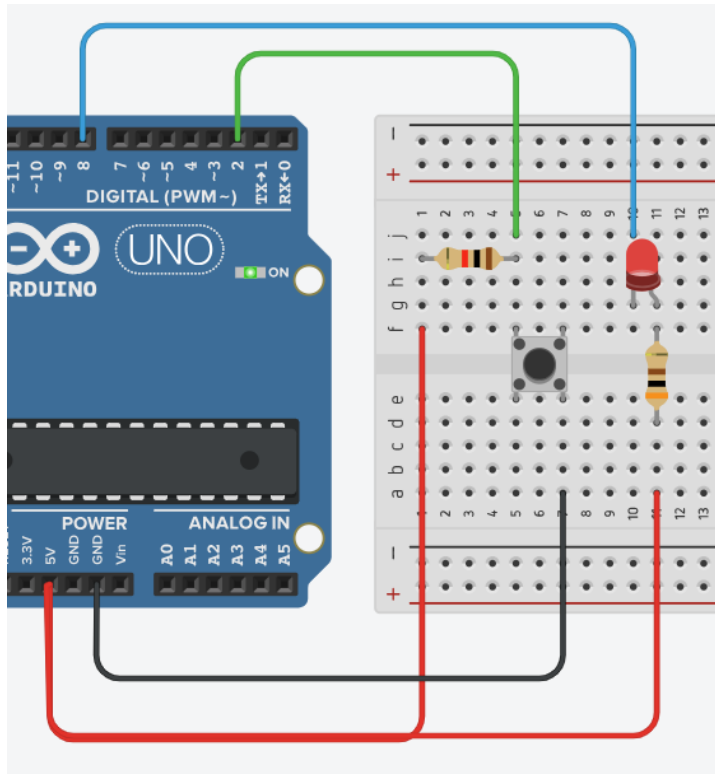
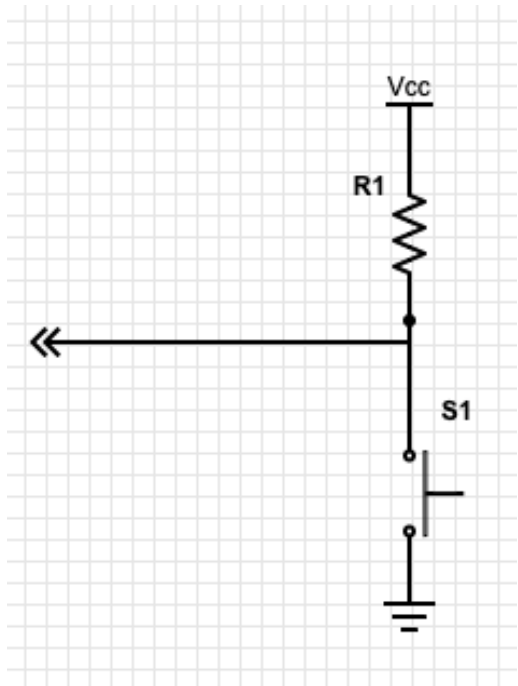
## 인터럽트 발동 조건 (mode)

모드	상태
LOW	핀이 LOW일때
CHANGE	LOW->HIGH or HIGH->LOW로 변할 때
RISING	LOW ->HIGH일때
FALLING	HIGH -> LOW일때
HIGH	핀이 HIGH일때

# 외부 인터럽트(External Interrupt)

## • 플링 vs 인터럽트

`attachInterrupt( digitalPinToInterrupt(2), ExINT, FALLING );`



```
void setup()
{
  pinMode(8, INPUT) ;
  pinMode(2, OUTPUT) ;

  attachInterrupt( digitalPinToInterrupt(2), ExINT, FALLING );

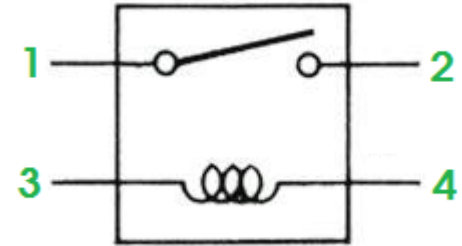
  Serial.begin(9600) ;
}

void loop()
{
  digitalWrite(8, 0) ;
  delay(1000) ;

  digitalWrite(8, 1) ;
  delay(1000) ;
}

void ExINT()
{
  Serial.println("ExINT") ;
}
```

# 마그네틱 도어센서 실험



# QUIZ



문이 열리면(버튼이 눌렸을 때) 경고음(부저)을 울려봅시다.