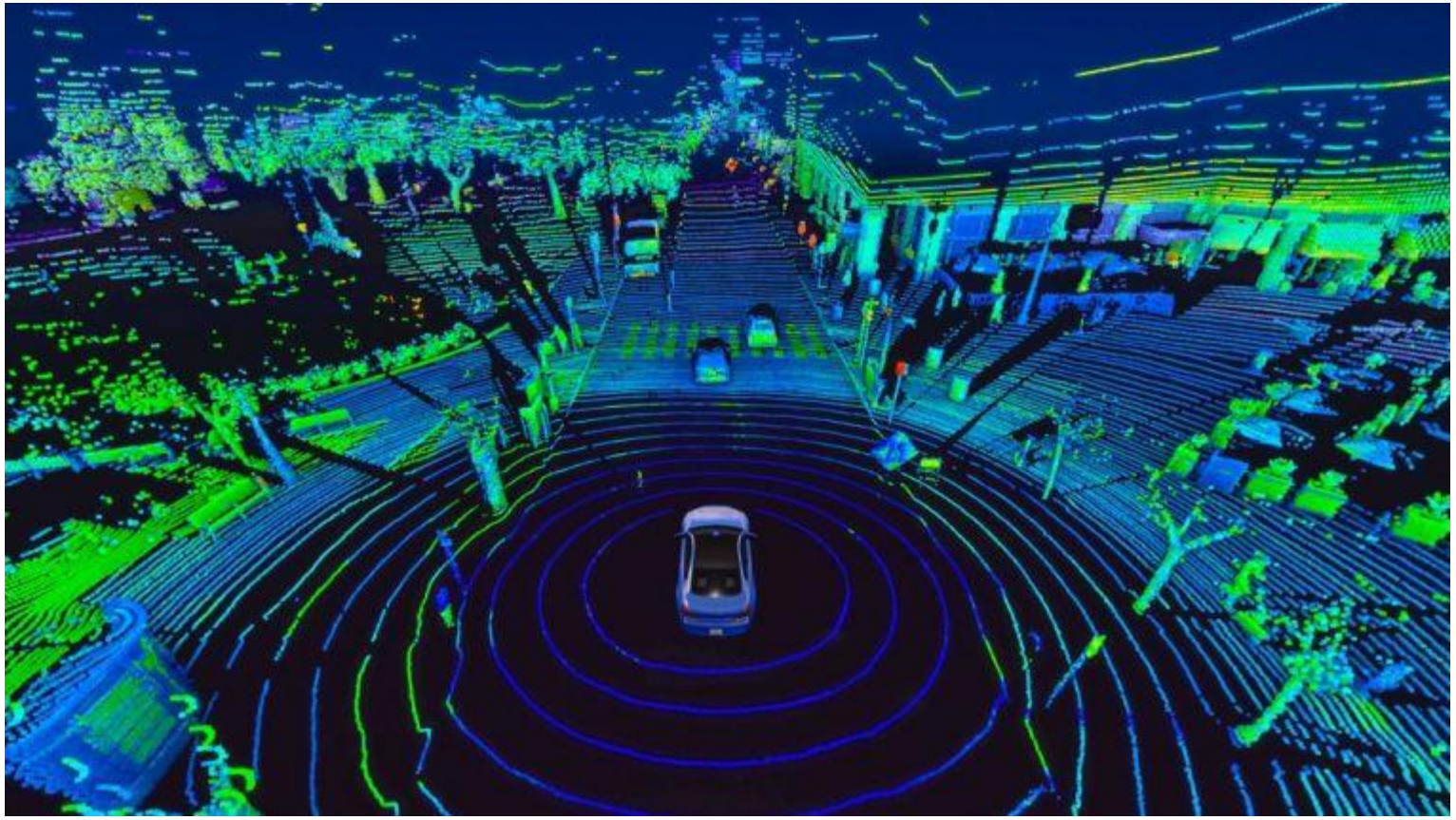
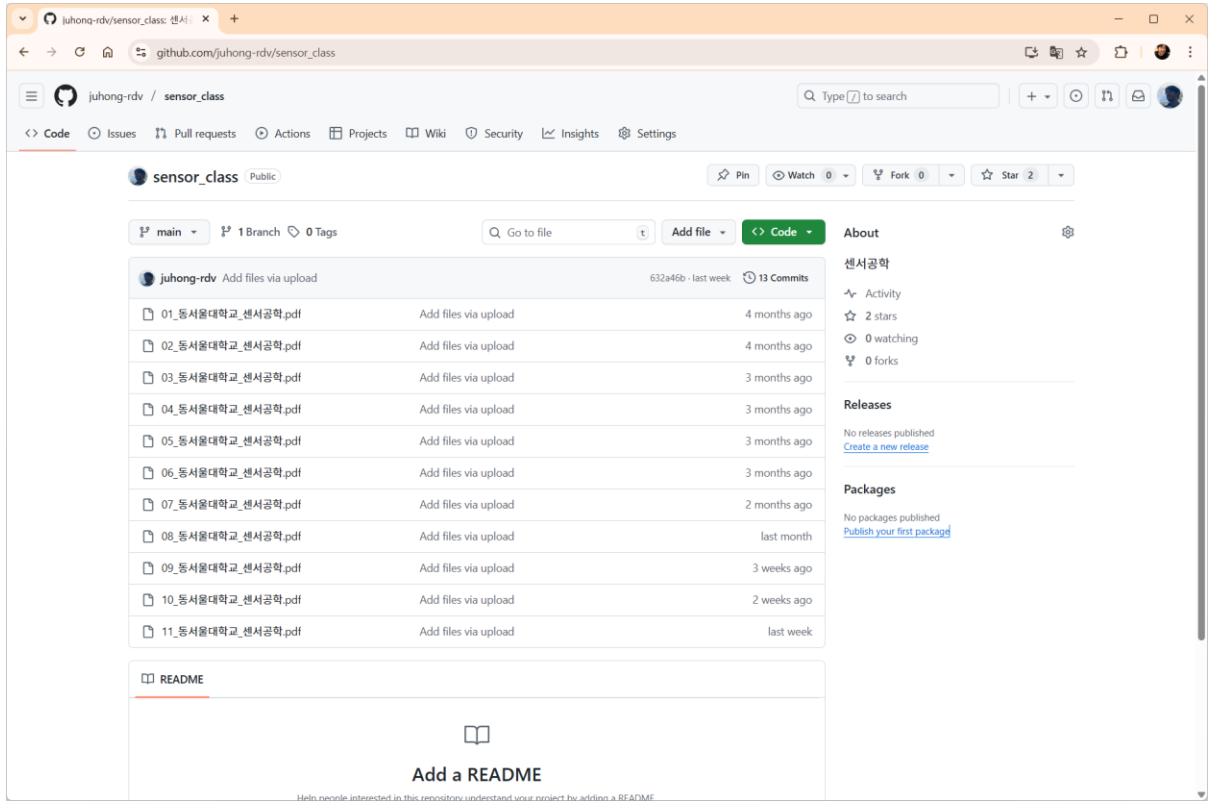
센서공학

2025 FALL



수업자료

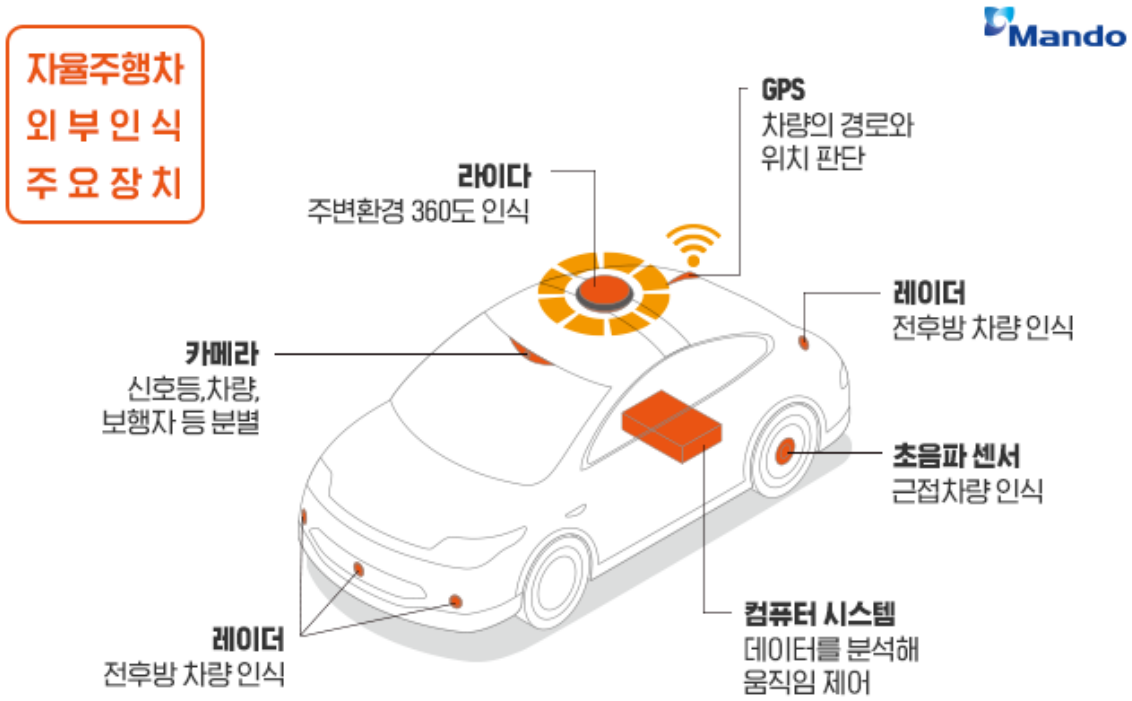
- https://github.com/juhong-rdv/sensor_class



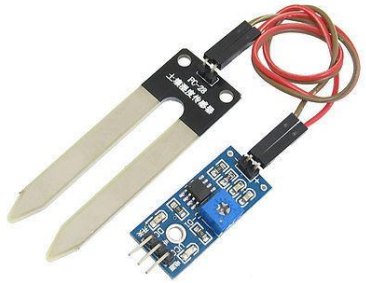
센서?

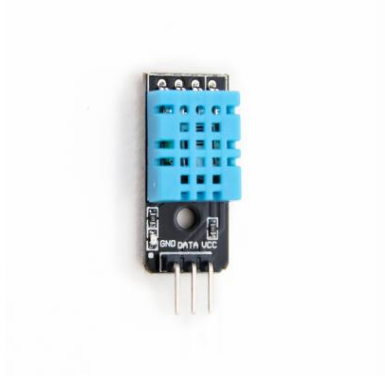
- 열, 빛, 온도, 압력, 소리등의 물리적인 양이나 그 변화를 감지하여 일정한 신호로 알려주는 부품이나 기구, 또는 계측기
- 사람의 오감

자율주행 자동차는 어떻게 운전자 없이 달릴 수 있을까?

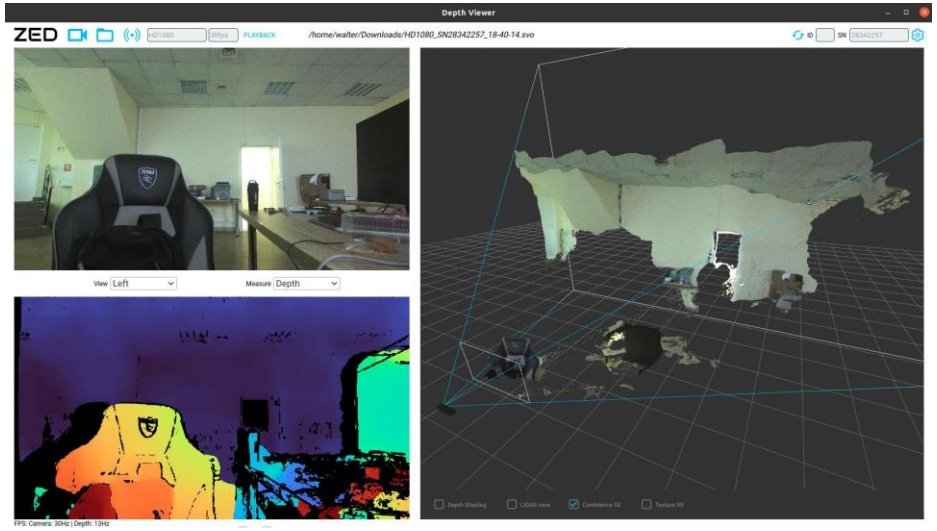


스마트팜에는 어떤 센서가 사용될까?



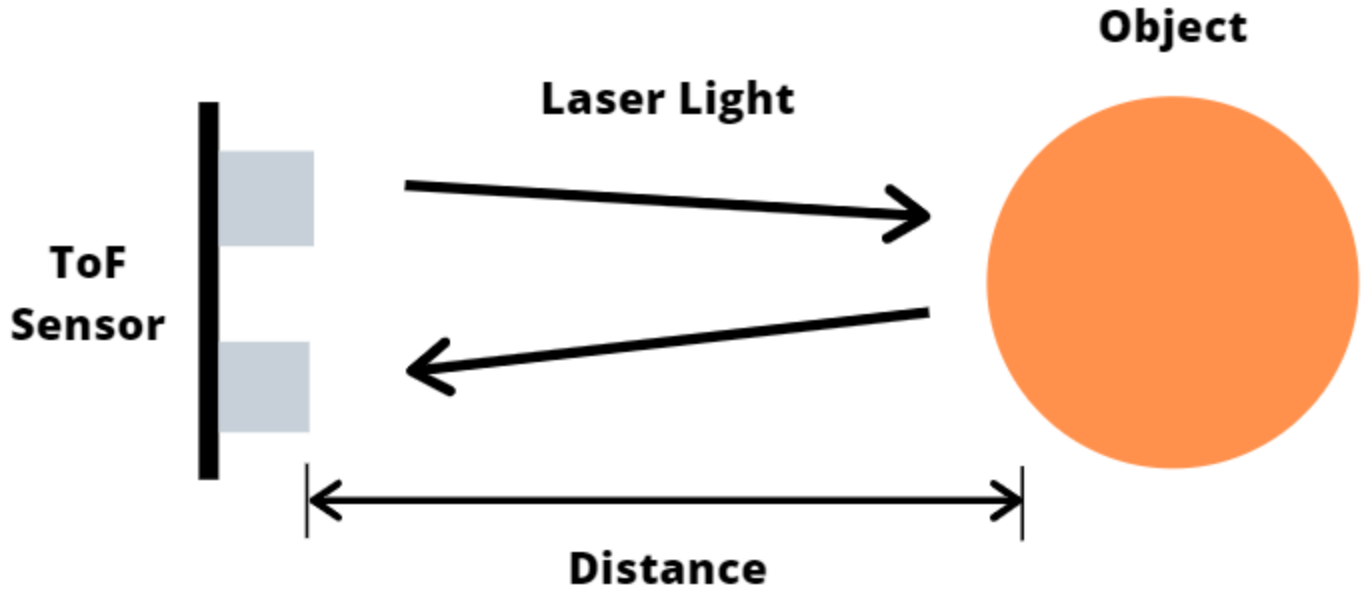


자율 주행 자동차에서 사용하는 센서의 원리



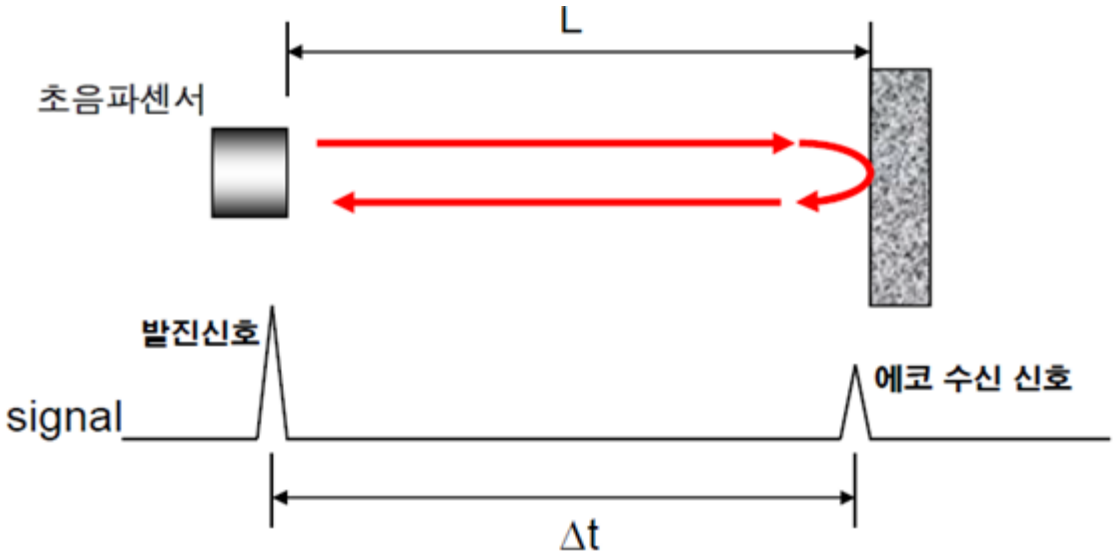
자율 주행 자동차에서 사용하는 센서의 원리

- 거리 측정 센서의 원리(ToF, Time of Flight)
  - 초음파 거리 센서
  - 레이저 거리 센서
  - LiDAR



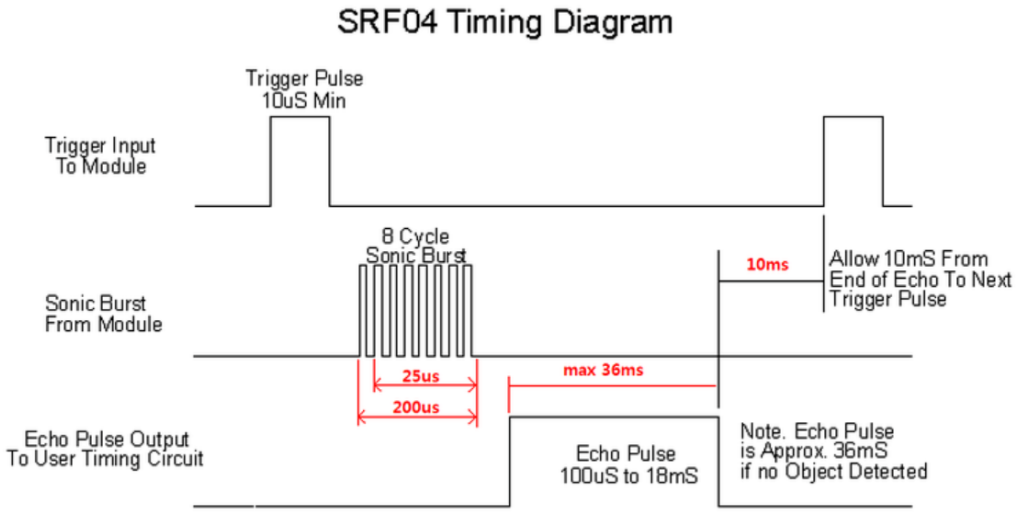
ToF(Time of Flight)

- **ToF**는 피사체를 향해 발사한 빛이나 소리가 반사돼 돌아오는 시간으로 거리를 계산해 사물의 입체감이나 공간 정보, 움직임 등을 인식하는 3D 센싱 기술이다



초음파 센서 모듈

- SRF04 초음파 모듈을 사용하여 장애물까지의 거리 측정



초음파를 이용한 거리 측정

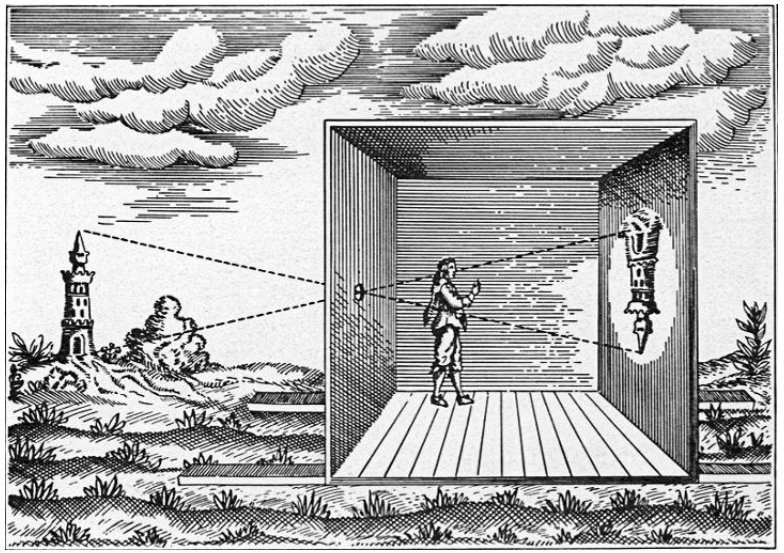
$$t = \frac{2 \times L(\text{물체와의 거리 m})}{Vs(\text{음속 m/s})}$$

t: 신호가 되돌아 올때까지 걸리는 시간(s)

재료	속 도 (m/s)
공기 (0℃)	331
공기 (20℃)	344
물 (25℃)	1498
목재 (소나무)	3300
유리	5000
철	5000
화강암	6000

카메라 센서의 원리

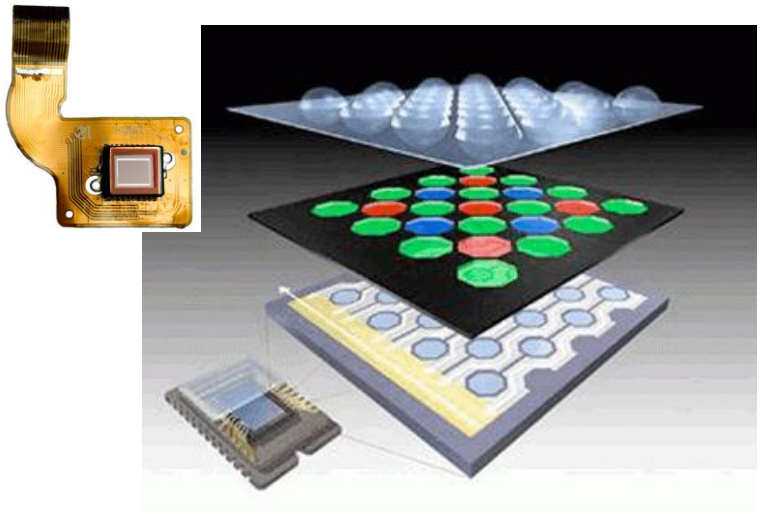
- 카메라 이미지 센서의 원리
  - 카메라의 이미지는 어떻게 수집 될까?
    - 빛 → 렌즈 → CCD센서 → Memory

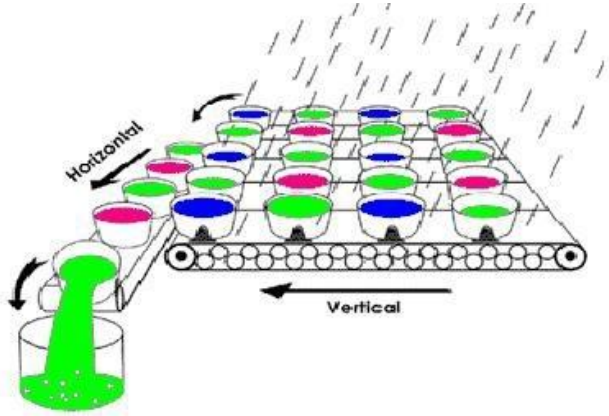


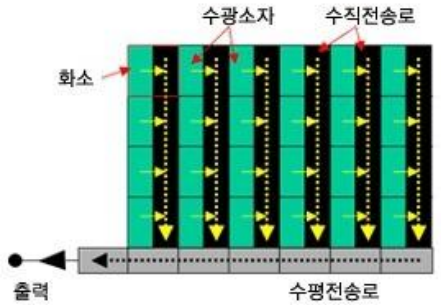
최초의 카메라 원리 '카메라 옵스큐라'

카메라 센서의 원리

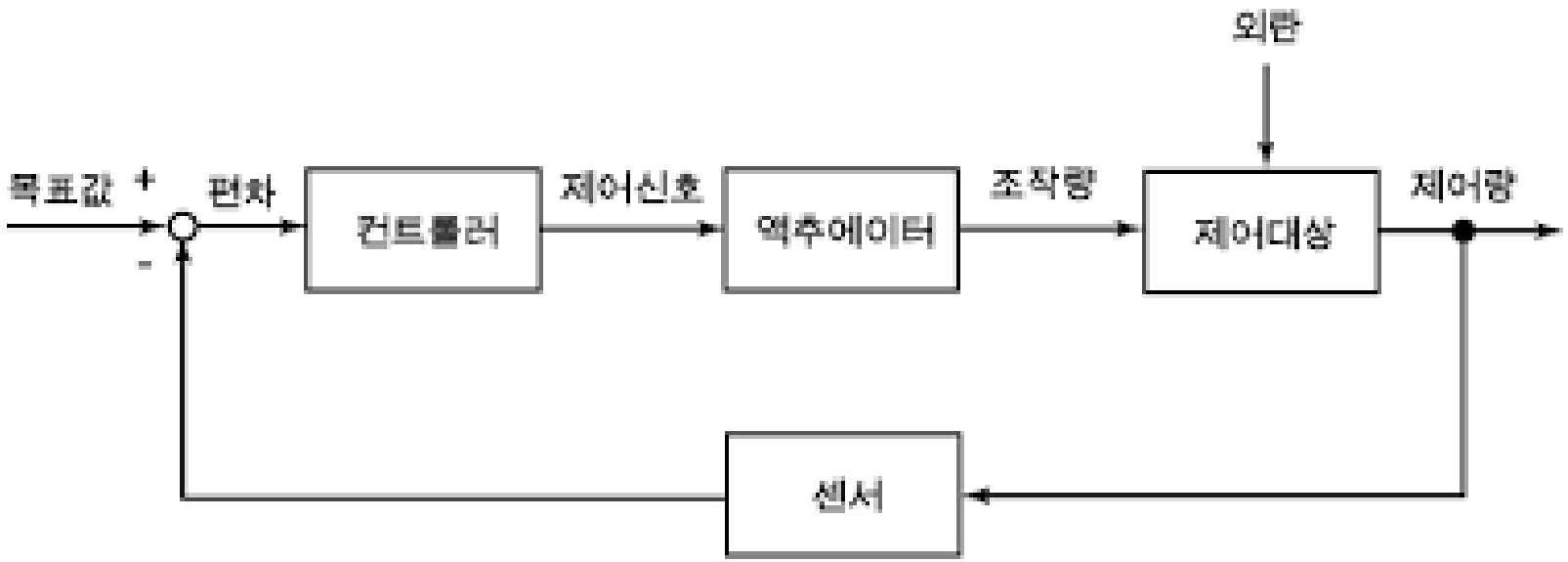
- 카메라 이미지 센서의 원리
  - 카메라의 이미지는 어떻게 수집 될까?
    - 빛 → 렌즈 → 센서 → Memory





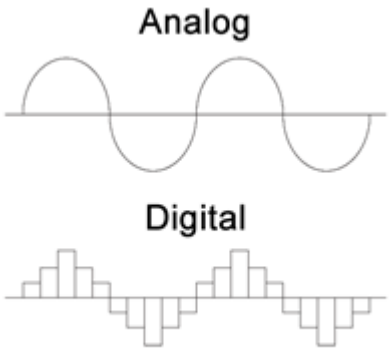


피드백 제어 시스템



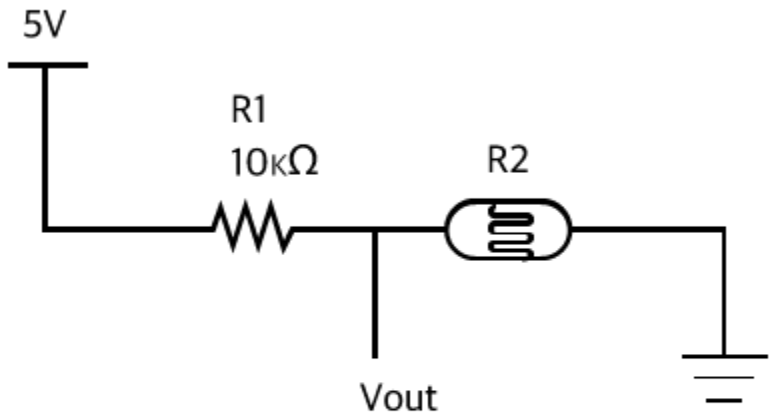
ADC(Analog to Digital Converter)

- 아날로그는 하나의 연속된 그래프(신호)
- 디지털은 구간별 레벨값으로 연속X
- 아날로그값인 센서의 값을 컴퓨터가 측정 하려면 디지털값으로 변환 해야 하는데, 이러한 역할을 해주는 것을 ADC라고 한다.

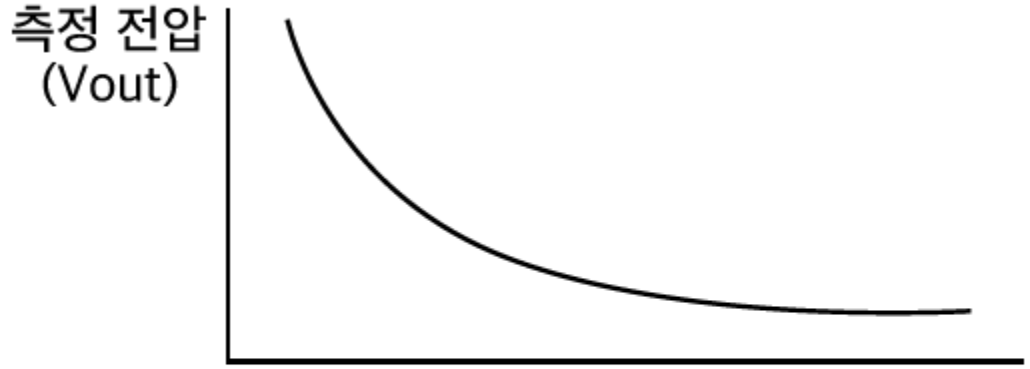


analogRead

조도센서(CDS cell)



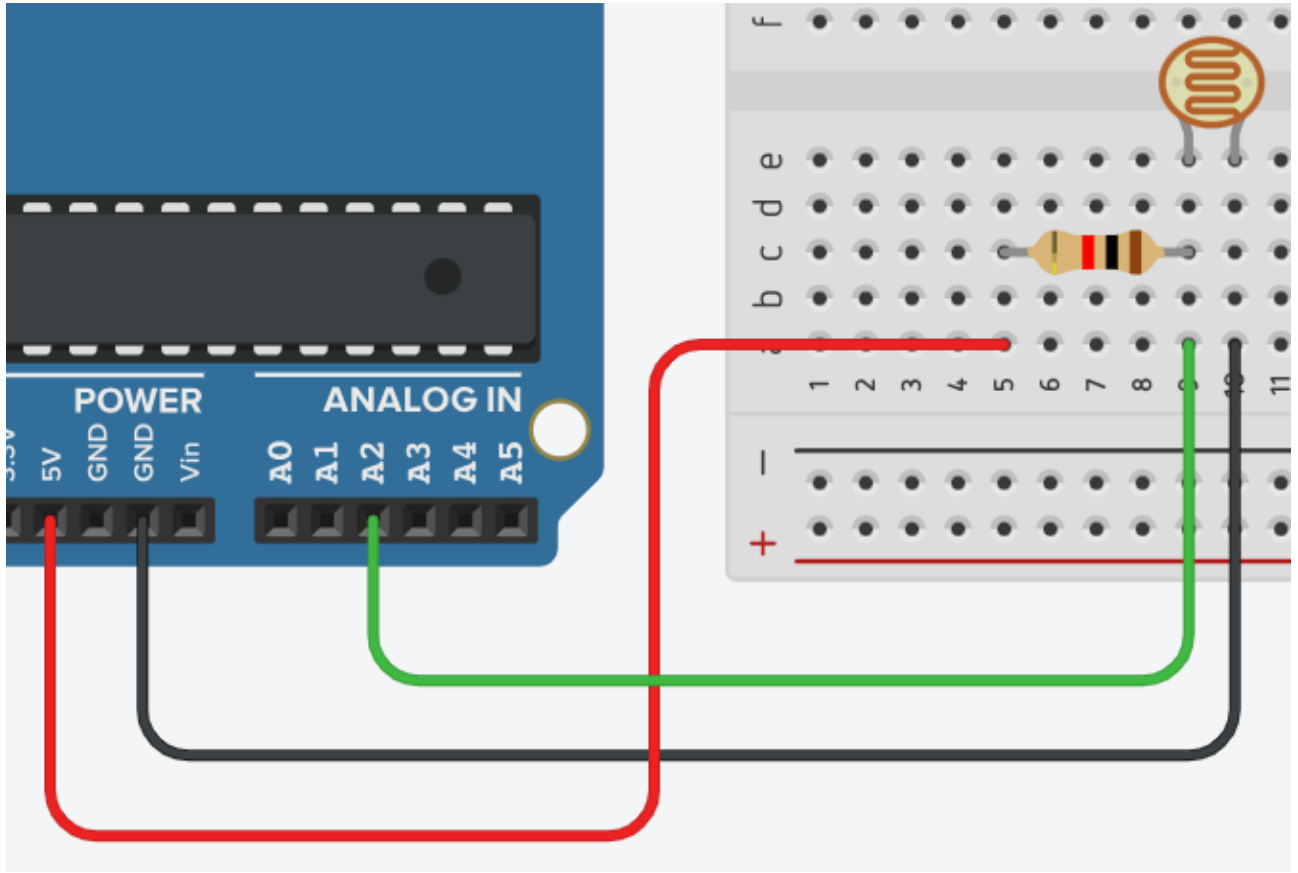
풀업 저항 사용



풀업 저항 사용시 밝기에 대한 측정 전압

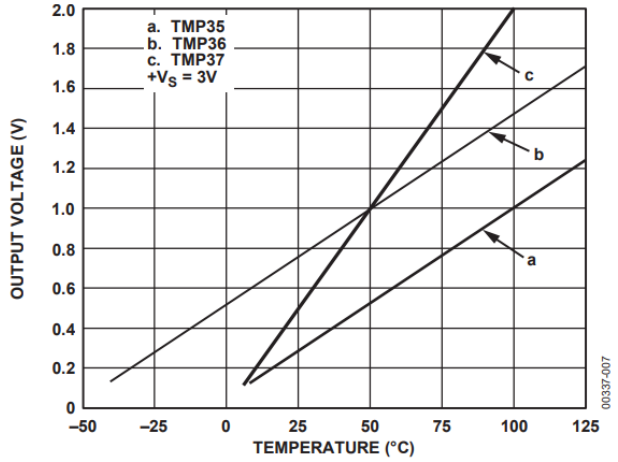
analogRead Example

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

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



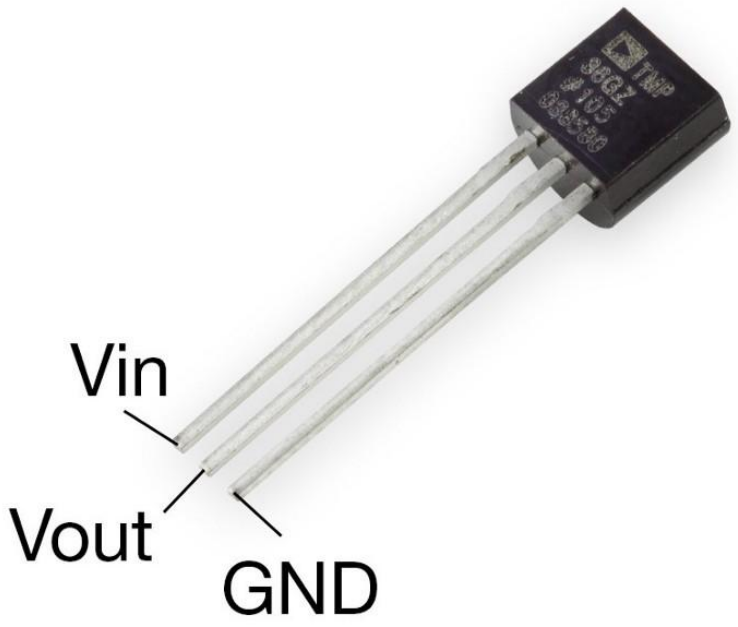
TMP36

- 온도센서는 온도를 감지해 전기신호로 바꿔주는 센서를 의미
- TMP36
  - 상온에서 대략 750mV를 출력
  - 온도 1 ℃가 변화하면 10mV의 출력 전압이 변화 함
  - 정밀도는 ±1 ℃로 정밀한 온도 감지는 어려움.
  - 사용하기 쉽고 저렴하여 정밀한 온도 감지가 필요 없는 어플리케이션이 많이 사용 됨.



TMP36 데이타시트 : https://www.analog.com/media/en/technical-documentation/data-sheets/TMP35_36_37.pdf

TMP36 핀연결



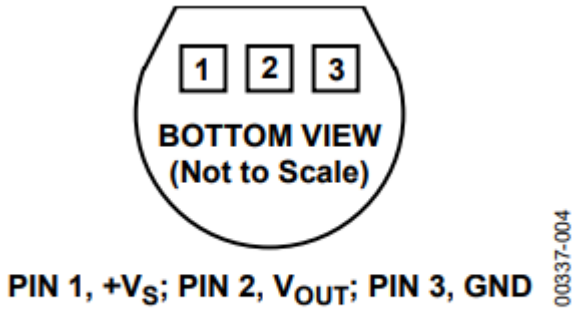
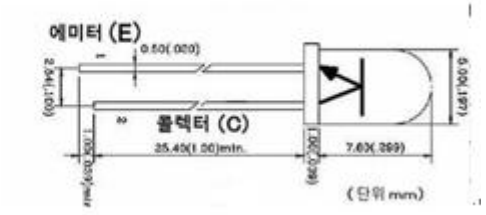
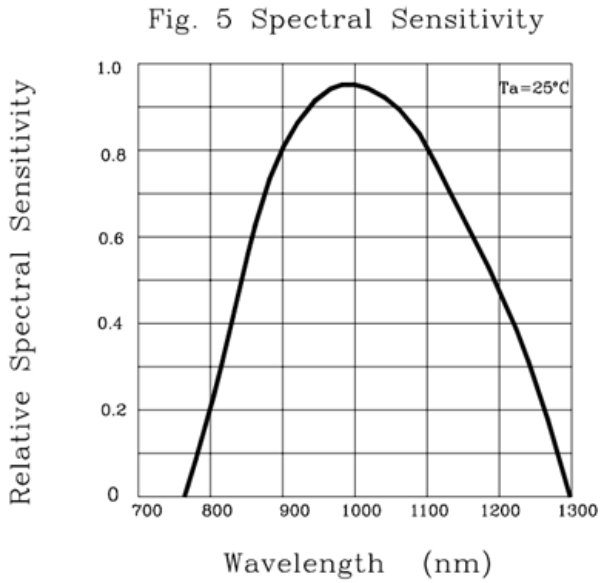


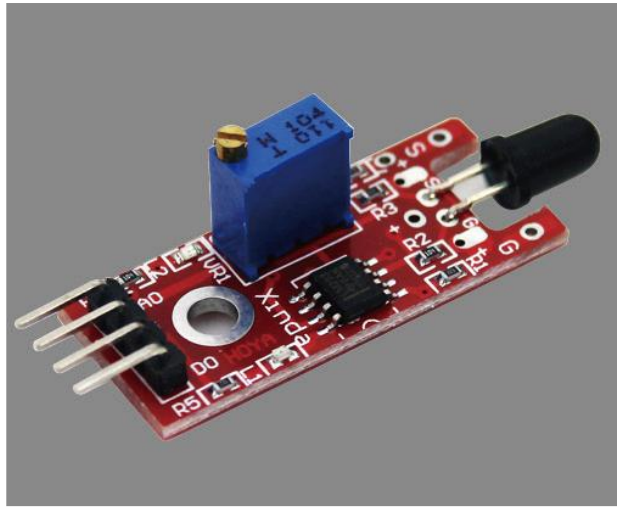
Figure 4. T-3 (TO-92)

불꽃감지센서 (Flame sensor)

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

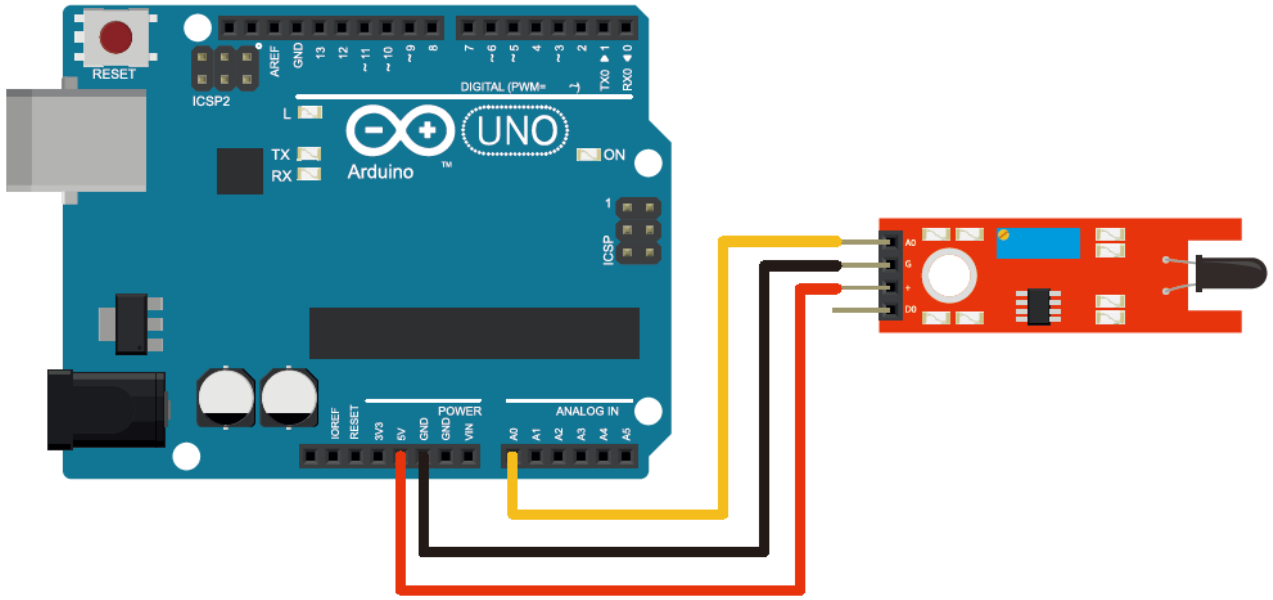






불꽃감지센서 (Flame sensor)

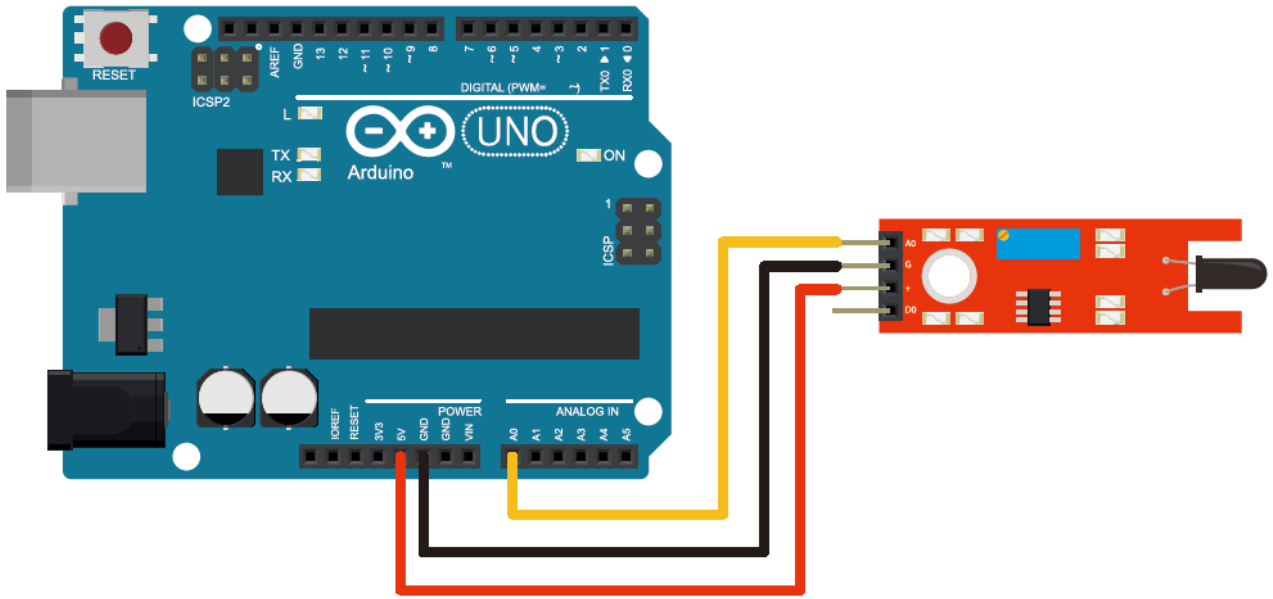
- 불꽃 감지 아두이노 실험 구성
  - 센서모듈 A0 <> 아두이노 A0
  - 센서모듈 G <> 아두이노 GND
  - 센서모듈 + <> 아두이노 5V



불꽃감지센서 (Flame sensor)

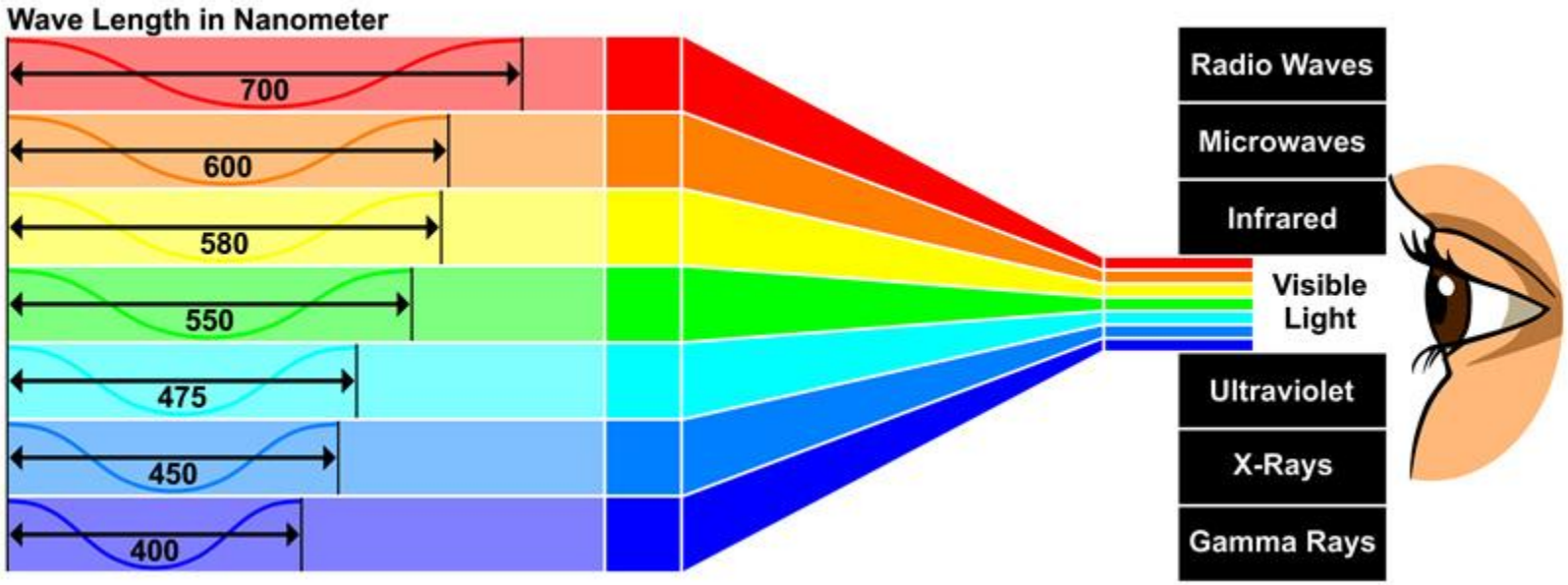
- 불꽃 감지 아두이노 실험 코드 작성

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

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

    Serial.println(analog_value);

    delay(100);
}
```



광센서를 이용하여 장애물(물체) 인식

- IR(적외선)을 이용하여 장애물 인식

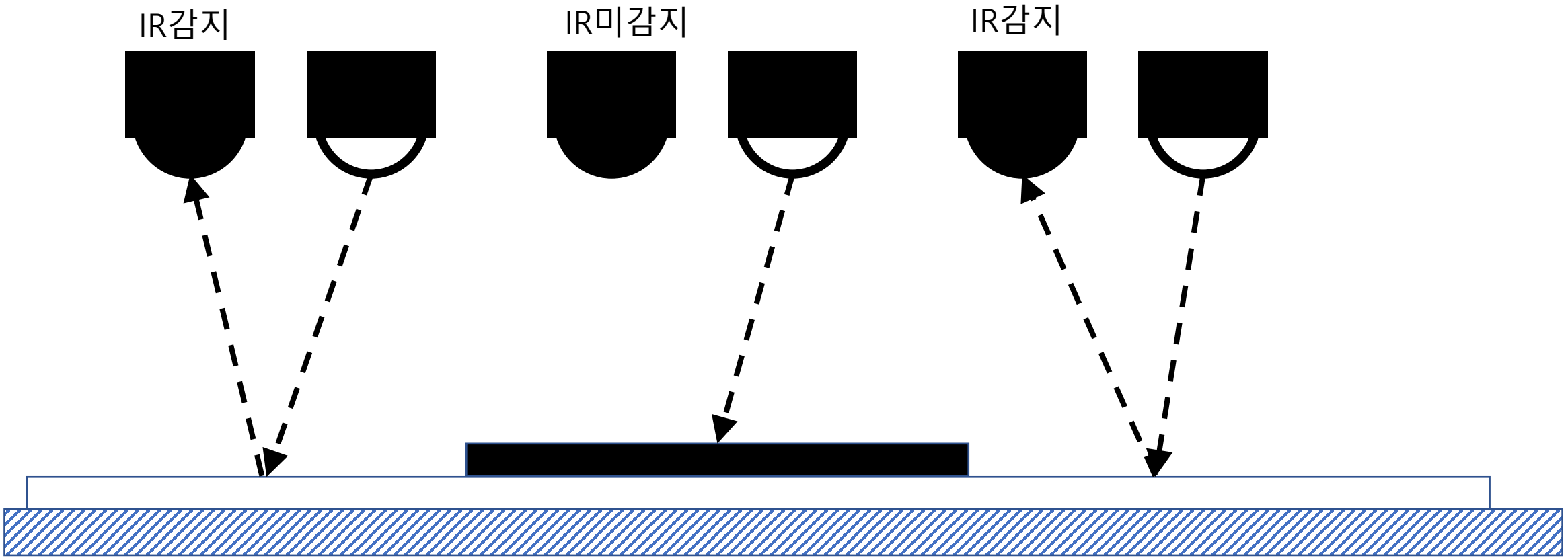


적외선 센서를 이용한 Line인식

- 바닥의 검은선을 인식

적외선 센서를 이용한 Line인식

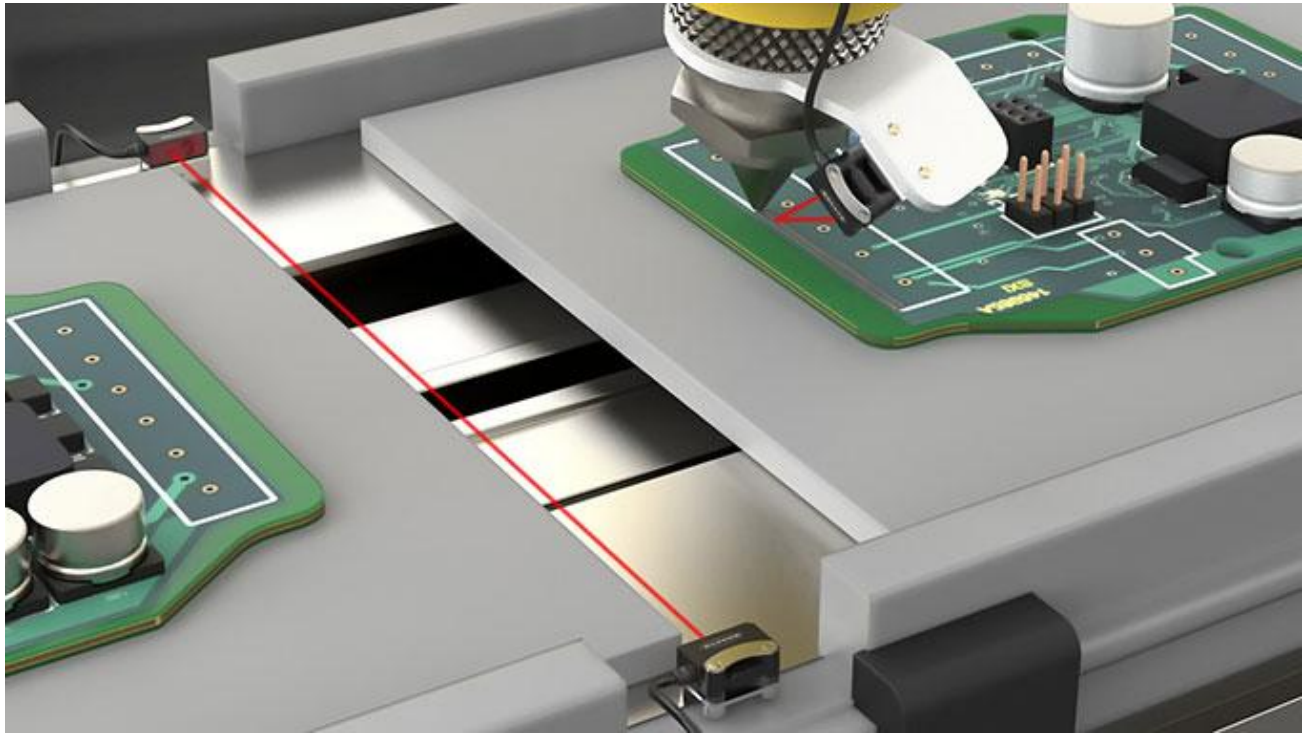
- 여러 개의 IR센서를 이용하여 바닥의 검은선의 위치를 인식



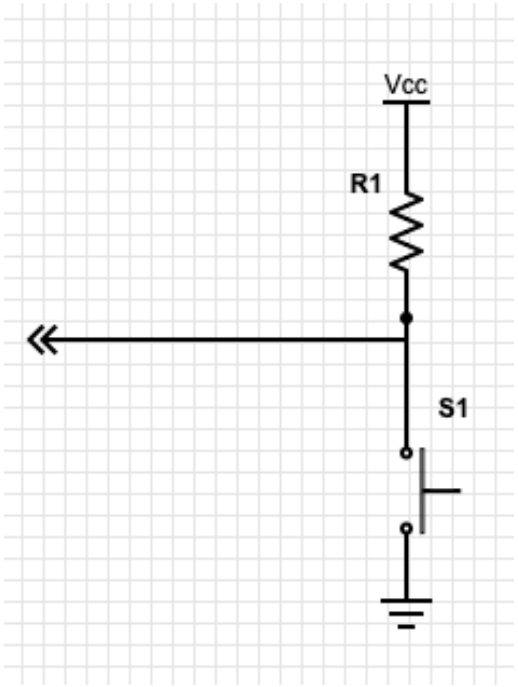
적외선 센서를 이용한 Line 추적 무인이동차

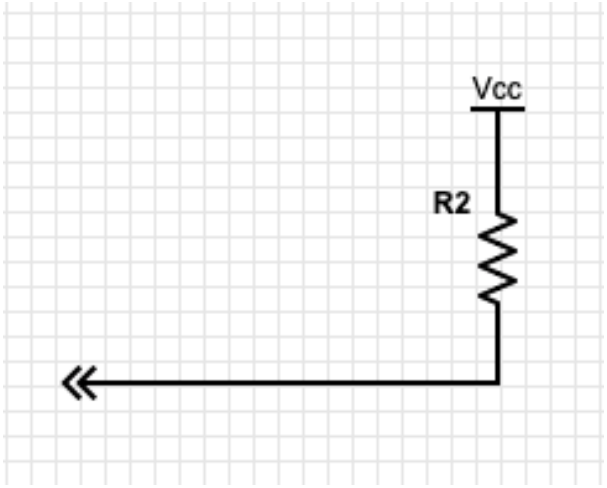
- 물류센터, 스마트펙토리, 스마트팜에서 물류를 자동으로 이동시키기 위해 가장 많이 사용하는 방식

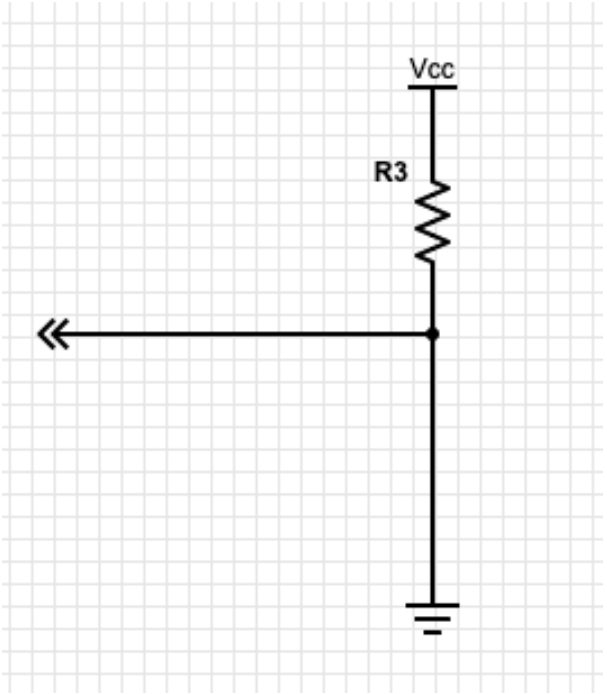
광(IR, 레이저)센서를 이용한 생산라인의 생산품 관리



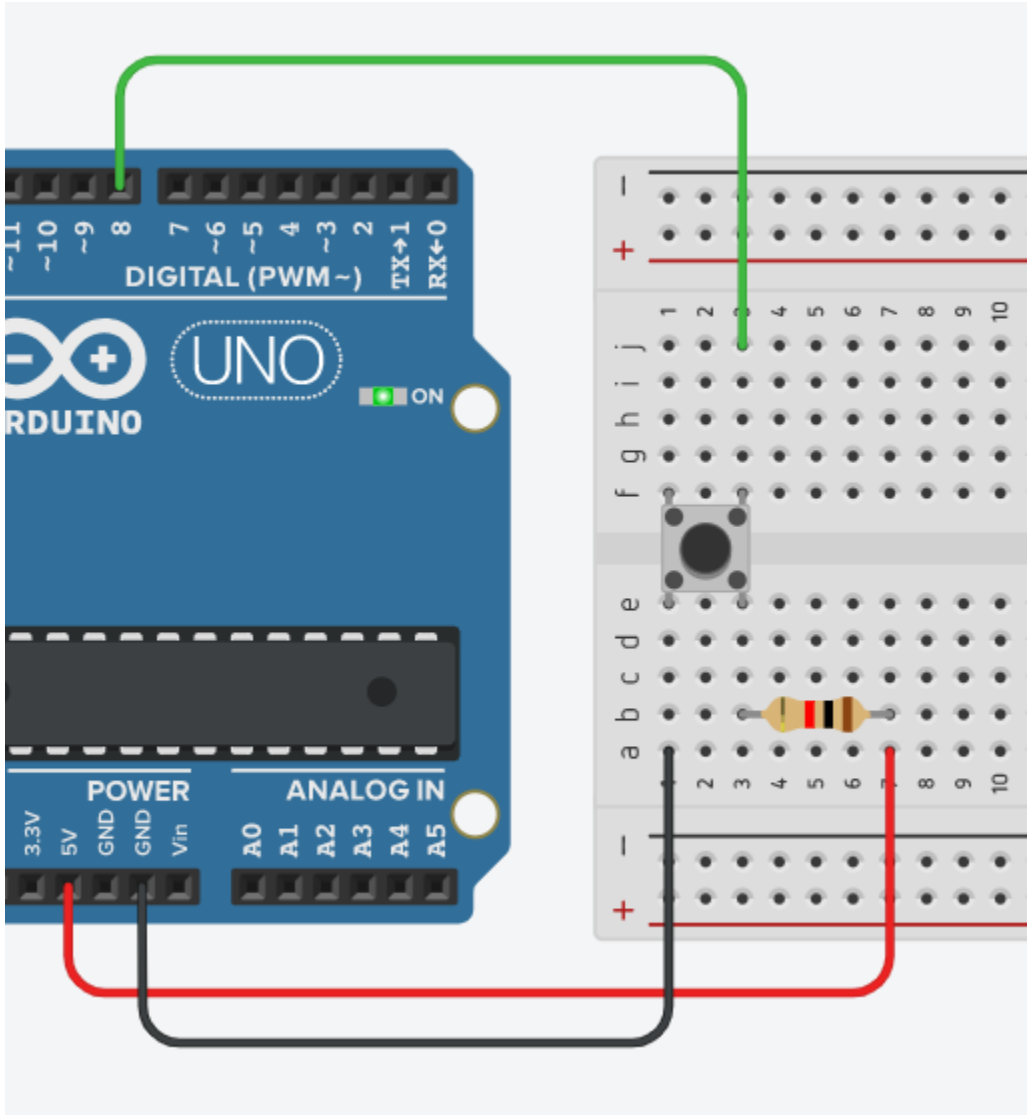
digitalRead





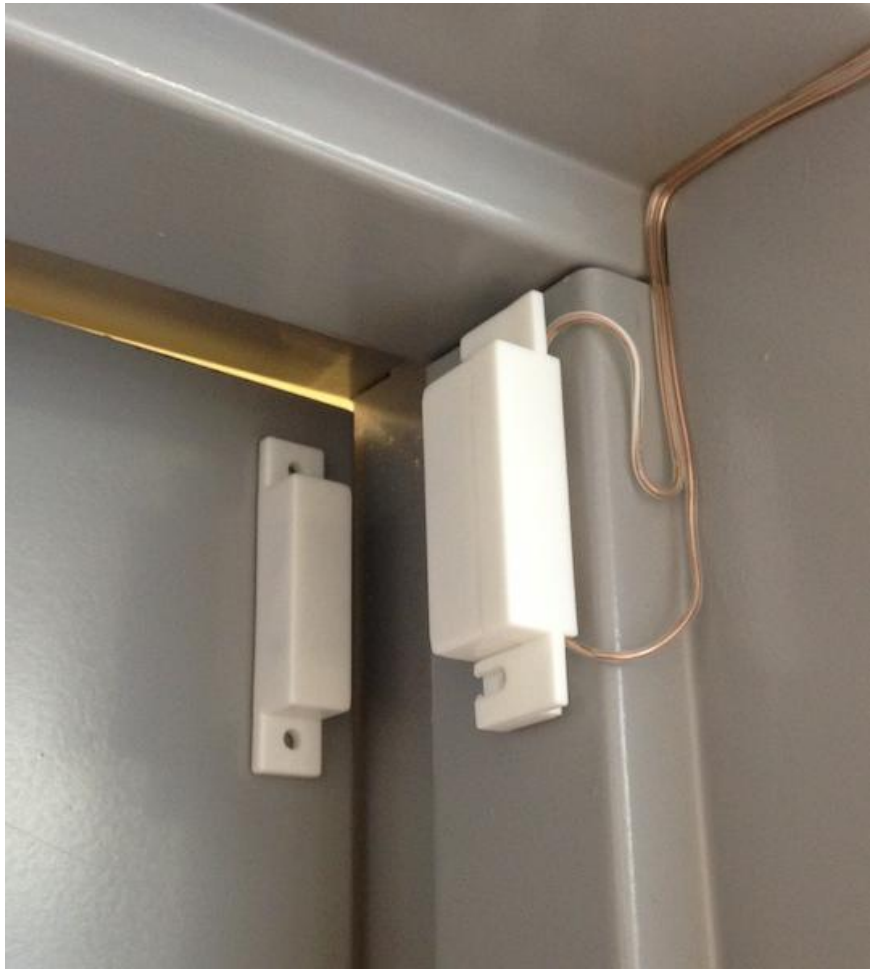


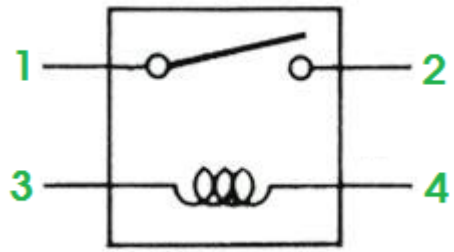
digitalRead




```
void setup()
{
  pinMode(8, INPUT) ;
  Serial.begin(9600) ;
}

void loop()
{
  int read = digitalRead(8) ;
  Serial.println(read) ;
}
```

마그네틱 도어센서 실험

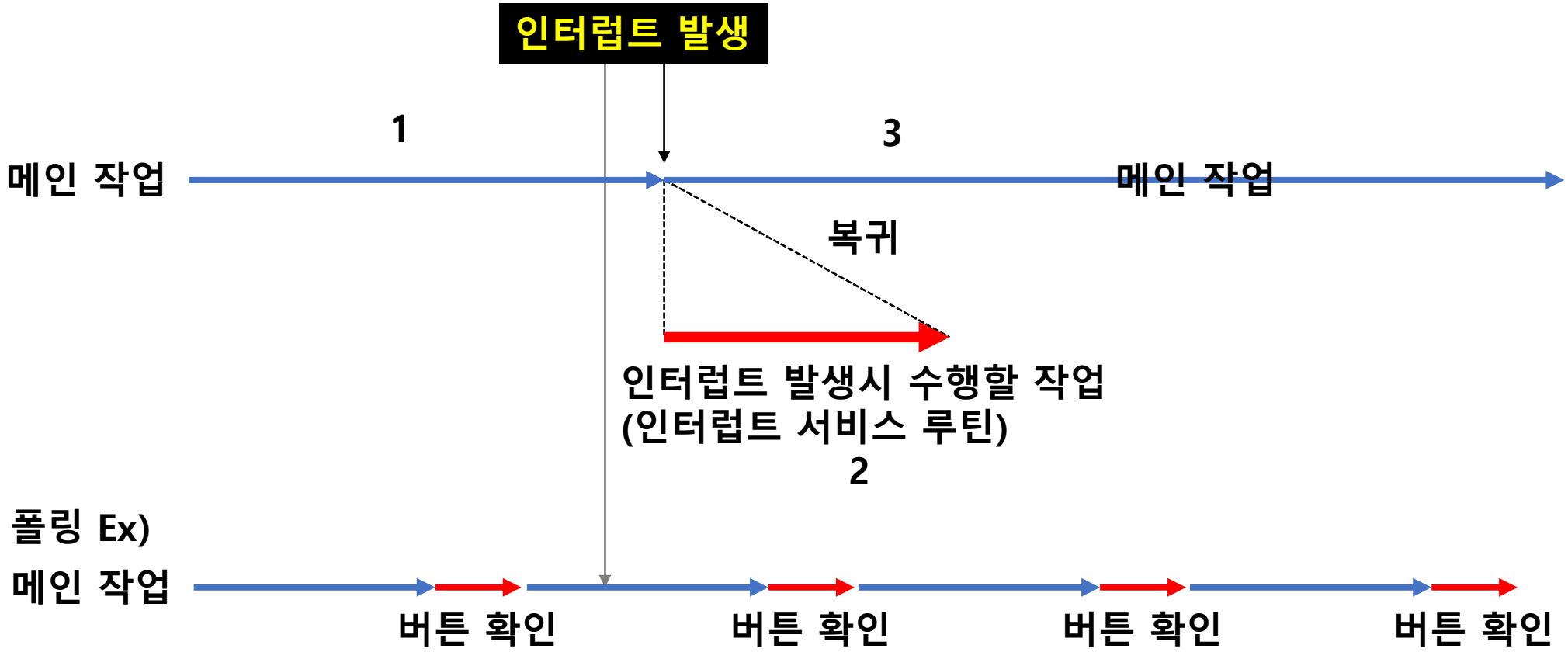



1
2
3
4


SONGLE
10A 250VAC 10A 125VAC
10A 30VDC 10A 28VDC
SRD-05VDC-SL-C

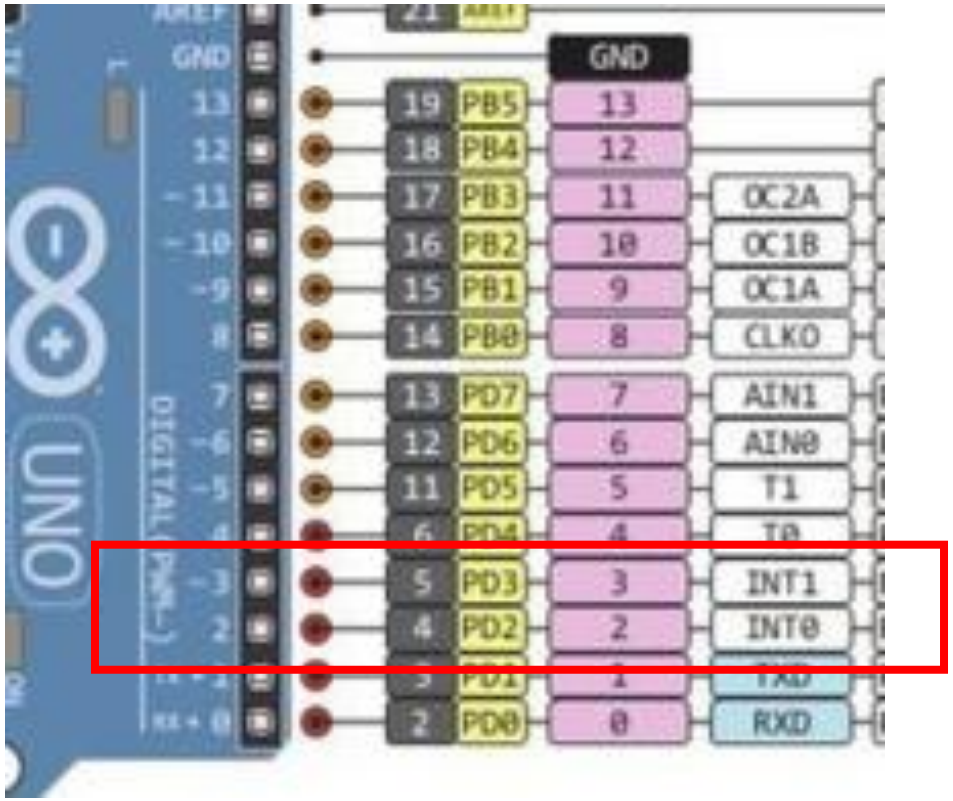
외부 인터럽트(External Interrupt)

- 폴링 vs **인터럽트**



외부 인터럽트(External Interrupt)

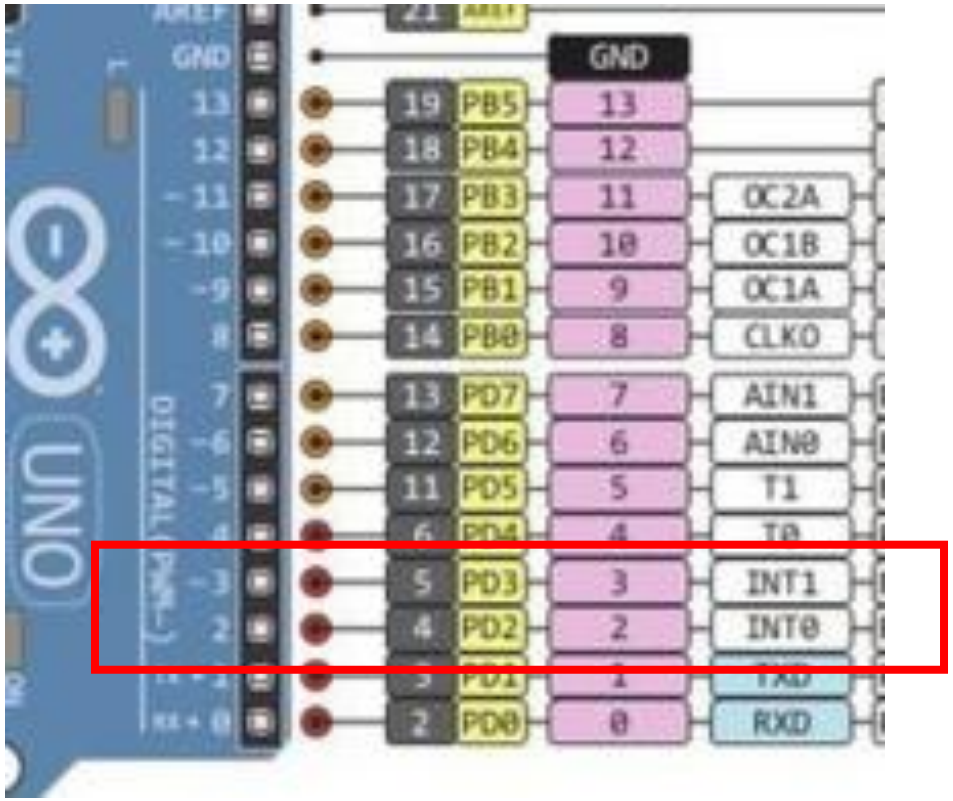
- 폴링 vs **인터럽트**



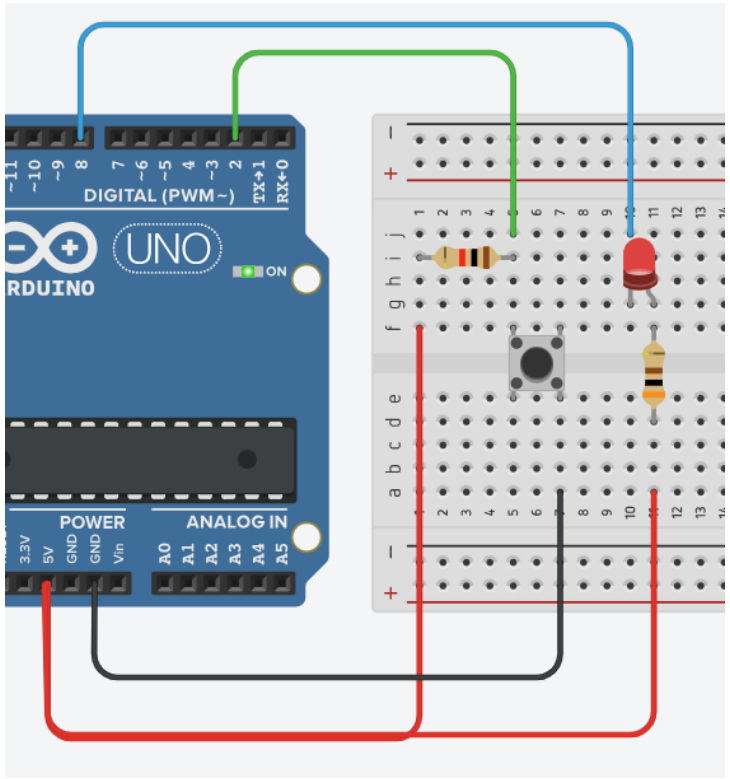
- **INT1 : Interrupt #1**
- **INT0 : Interrupt #0**

외부 인터럽트(External Interrupt)

- 폴링 vs **인터럽트**



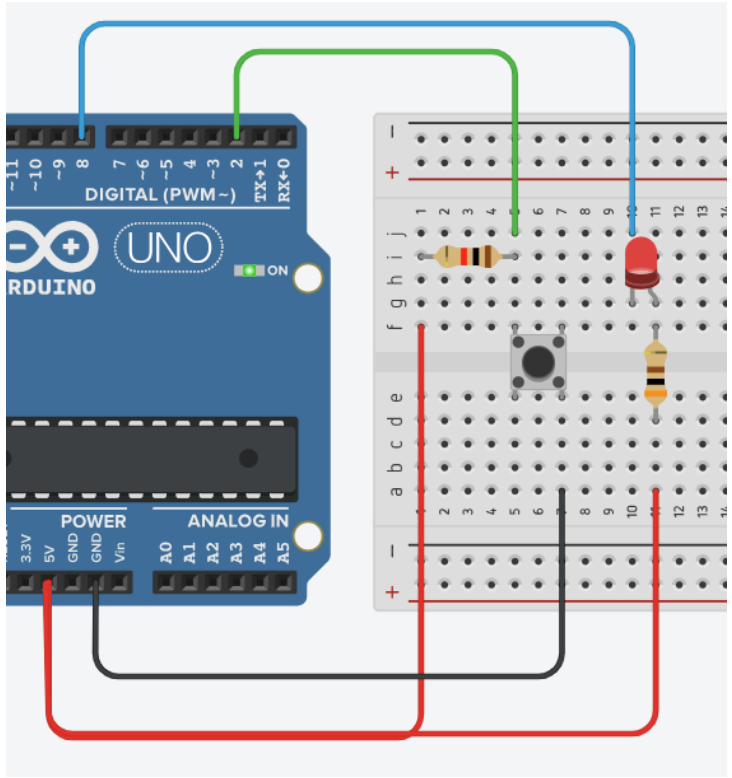
2 → INT0 : Interrupt #0



외부 인터럽트(External Interrupt)

- 폴링 vs **인터럽트**

2 → INT0 : Interrupt #0



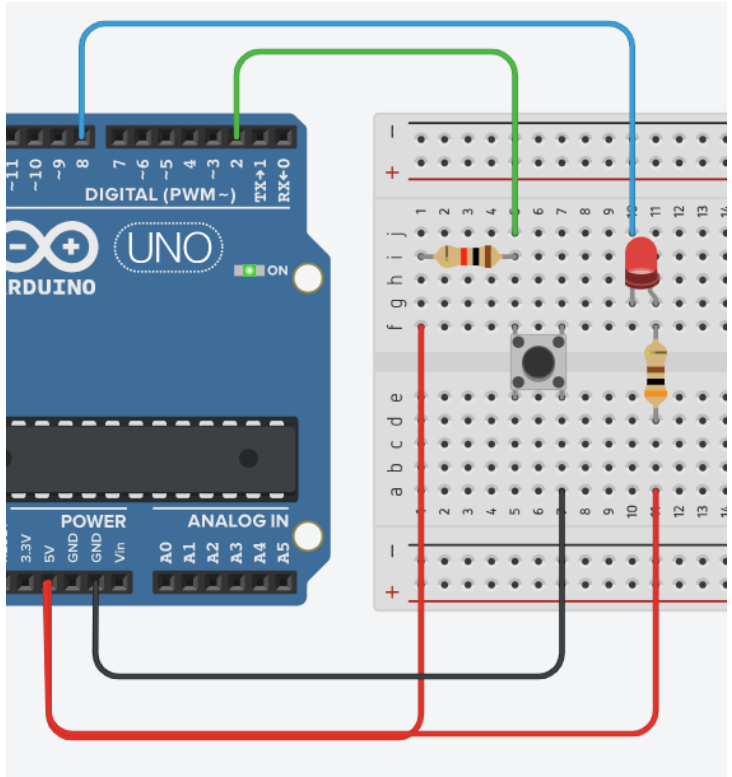
인터럽트 발동 조건 (mode)

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);

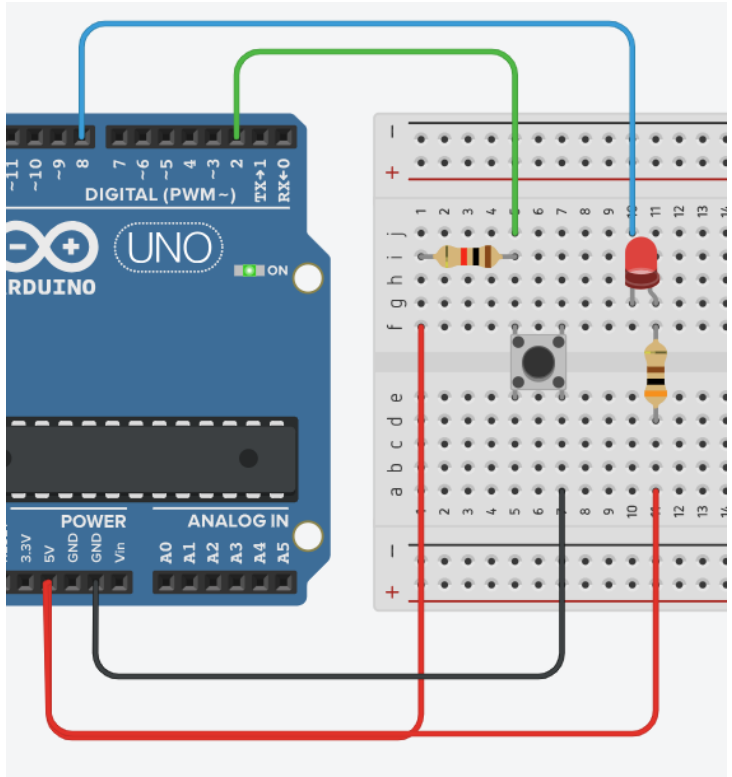
인터럽트 발동 조건 (mode)

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);

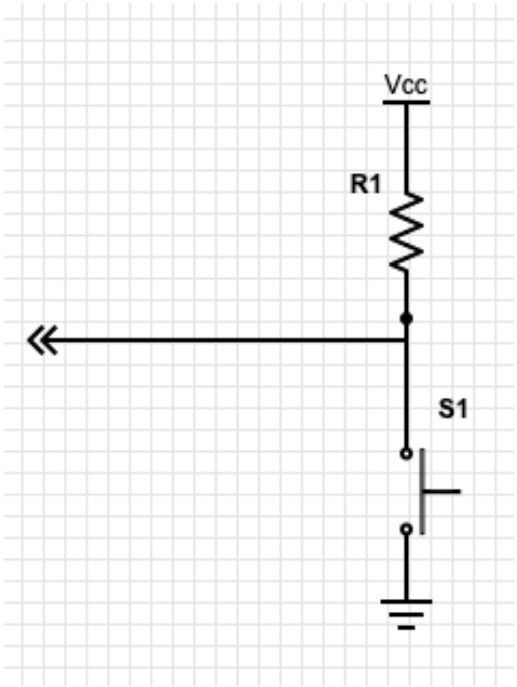
인터럽트 발동 조건 (mode)

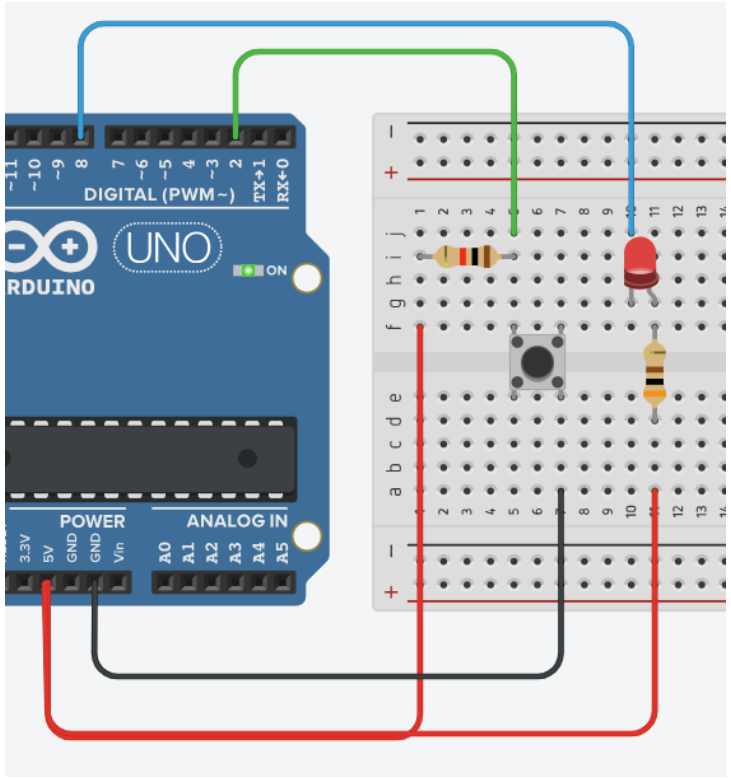
모드	상태
LOW	핀이 LOW일때
CHANGE	LOW->HIGH or HIGH->LOW로 변할 때
RISING	LOW ->HIGH일때
FALLING	HIGH -> LOW일때
HIGH	핀이 HIGH일때

외부 인터럽트(External Interrupt)

- 폴링 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") ;
}
```

라이브러리를 활용 : DHT11

- 동작 전압 (Power) 3~5 V
- 온도 측정 범위 (Temperature range) 0 ~ 50 ℃ (±2 ℃)
- 습도 측정 범위 (Humidity range) 20 ~ 80 % (±5 %)
- 최대소비전력 (Max. current) 2.5 mA
- 데이터 주기 (sampling rate) 1 Hz