

# Robotics Term Project

Team 5

---

202035510 Kim YooHyun

202235038 Moon HeeJin

202235092 Lee JungMin

202236947 Lee JaeYong

# Index

01

Why  
Made This

02

Project  
Schedule

03

Process & Role

04

Entrance Door  
Secure &  
Illumination  
Sensor

05

A Fire  
Alarm

06

Impact  
Sensor

07

Humidity  
Sensor &  
LCD

## Why Made This – IoT House

**Our team planned for some production using useful sensors in real life.**

**So, we talked about a house that we living in now, sensors that will essentially work, and made a plan for the production.**

**First, we talked about the Entrance door Secure Sensor.**

**Almost every house has a door lock system, so our plan is partially based on this.**

## Why Made This – IoT House

**Second, a Fire Alarm Sensor. Nowadays every building has to consider this.**

**So, we planned it placed in outplace.**

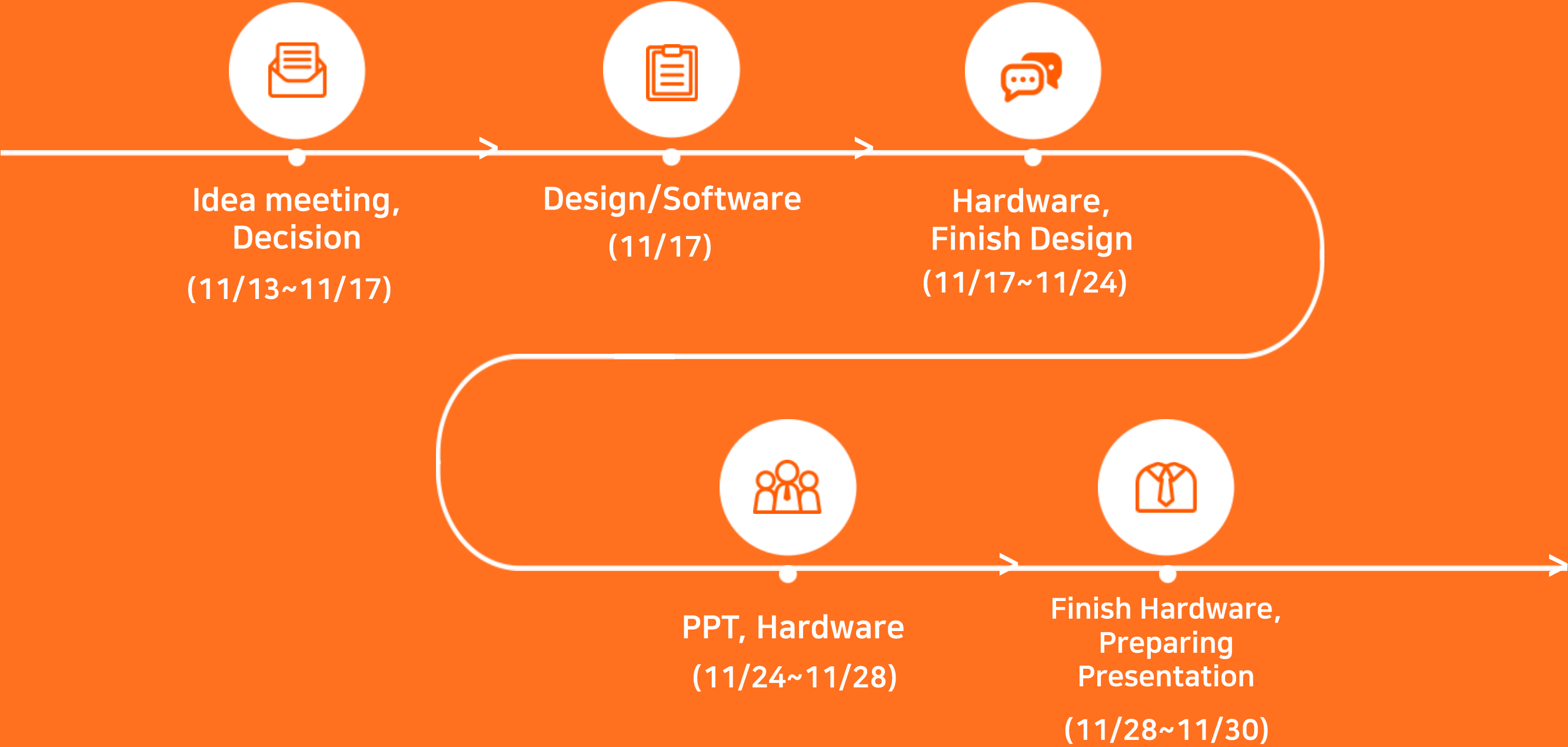
**Third, Impact Sensor. Usually, most buildings need to have Seismic Design.**

**But, we planned to install it house. Cause the house needs to be safe neither building.**

**Fourth, humidity sensor and LCD. We planned to see humidity and temperature (inside of the house) on the LCD screen.**

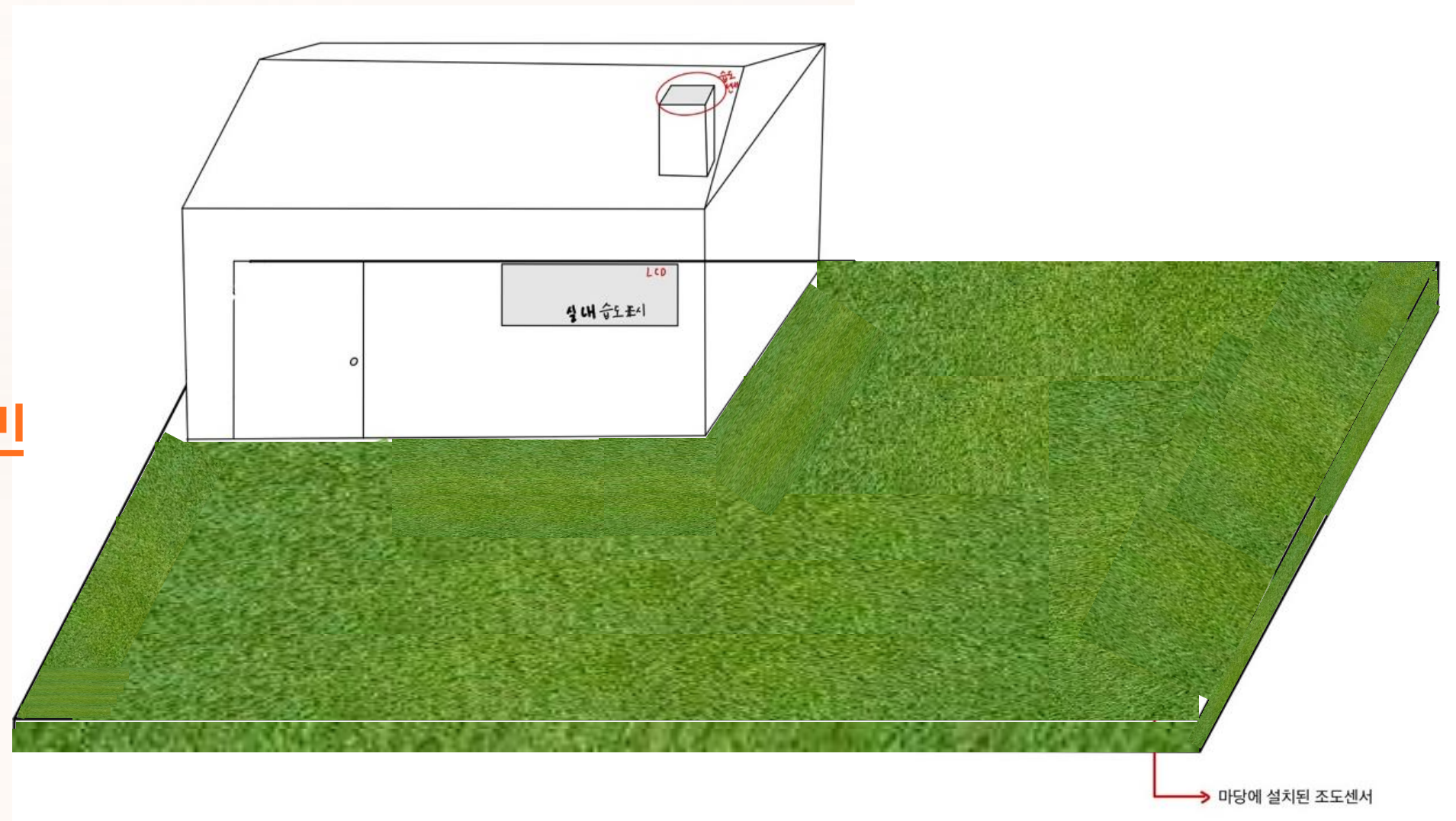
# Schedule

## Term Project Schedule



# Process & Role

- Ultrasonic Sensor : 이정민
- Illumination Sensor : 이정민
- Fire Alarm : 김유현
- Impact Sensor : 이재용
- Humidity Sensor & LCD: 문희진, 이정민
- Design: 김유현, 이재용
- PPT: 김유현, 문희진, 이정민
- Presentation: 이정민, 이재용





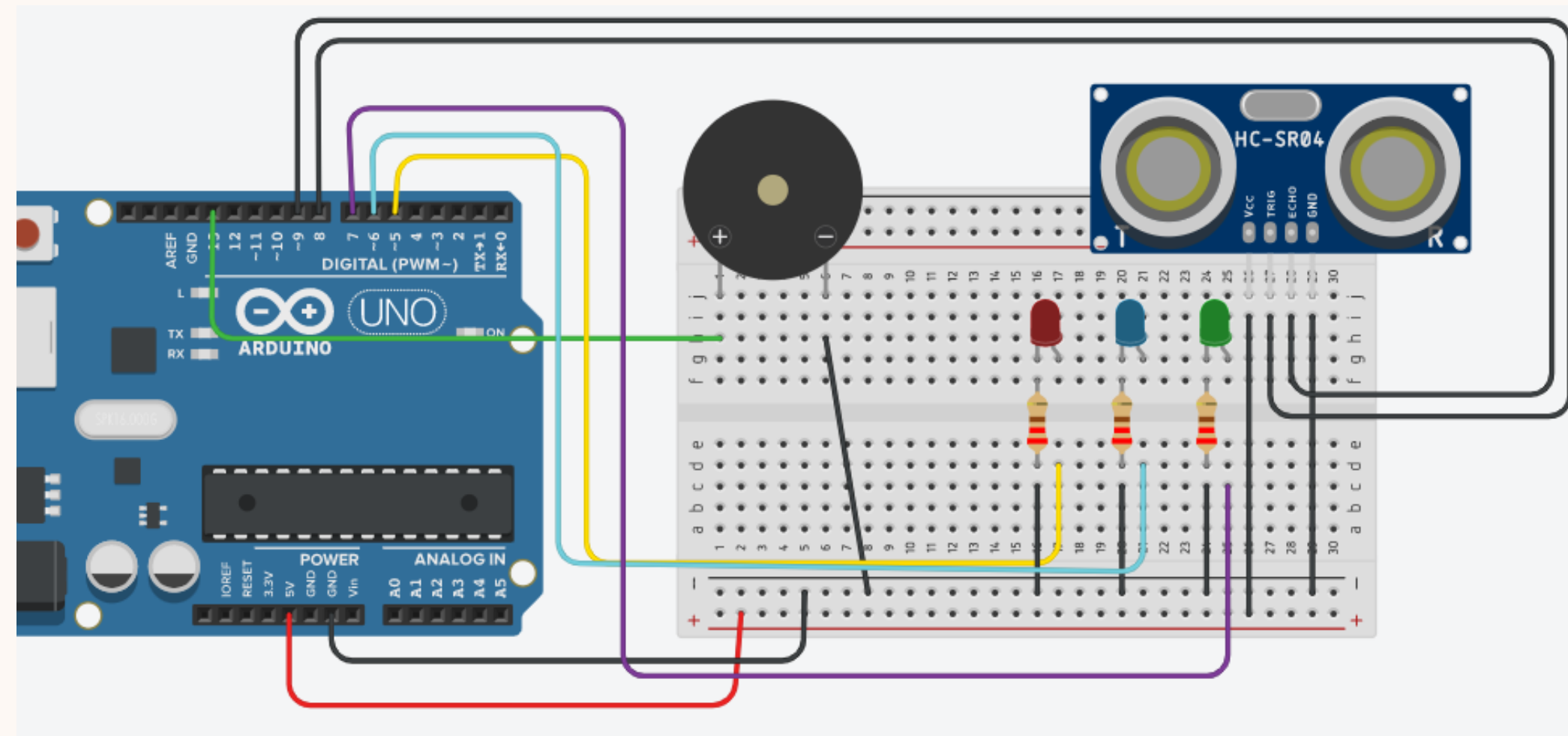
# Door Security System

## - Placing and Purpose

Install opposite the door to warn the user when the door is not fully closed

## - Used Sensors and kits

Arduino UNO, Ultrasonic Sensor, Piezo buzzer, LED



300%

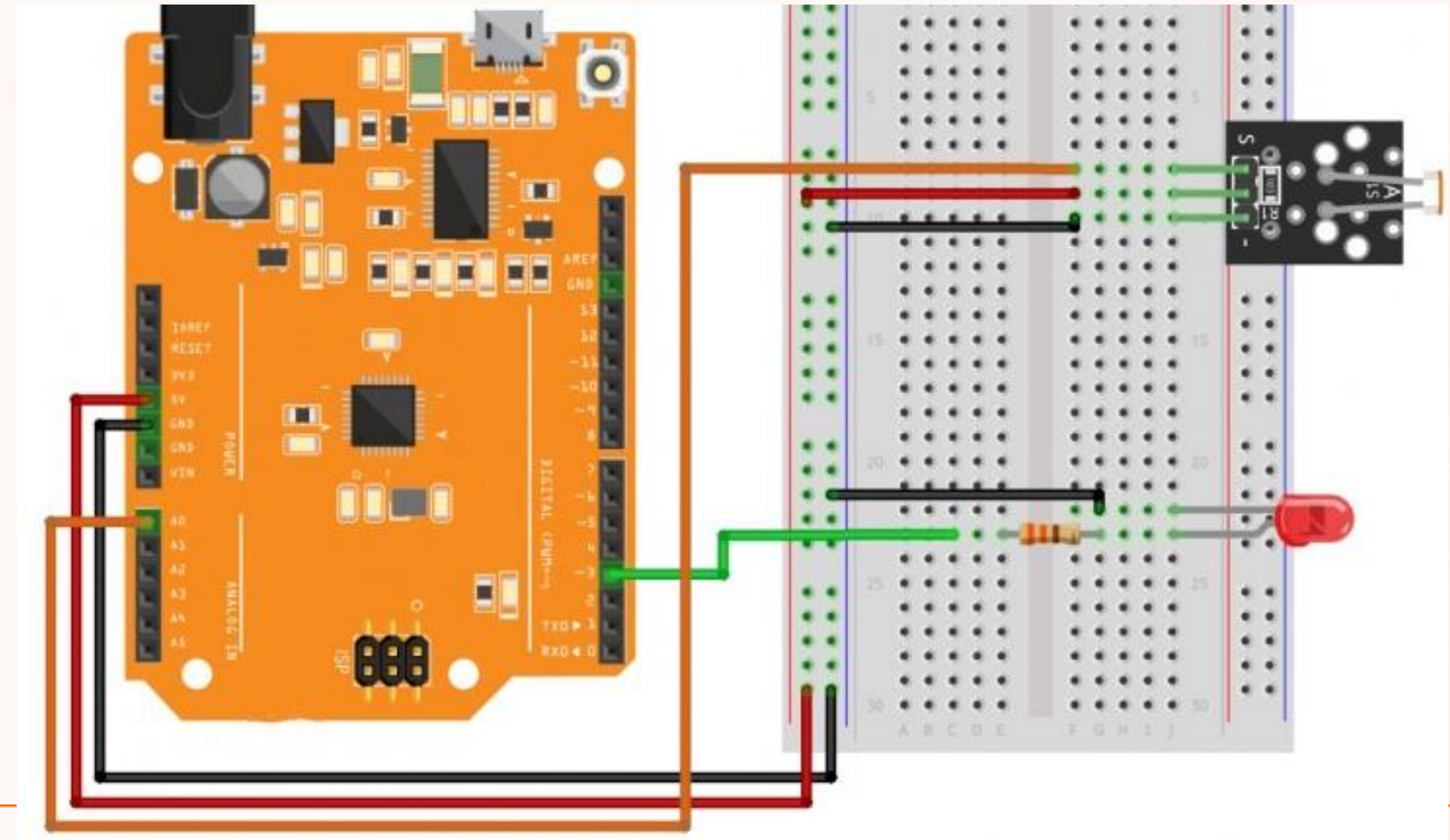
# Illumination System

## - Placing and Purpose

When the room is dim, the light is automatically turned on according to the sensor value

## - Used Sensors and kits

Arduino UNO, Illumination Sensor, LED





# Door Security System & Illumination Sensor

```
// 출입문 보안 경보기(초음파 + 피에조)

int echoPin = 8;
int trigPin = 9;
int buzzer = 13;
int cds = A0;
int led = 3; // 조도센서에 사용될 LED

int timer = 0;

void setup()
{
  pinMode(13, OUTPUT);
  pinMode(3, OUTPUT);
  pinMode(9, OUTPUT); // trig 핀 모드 설정
  pinMode(8, INPUT); // echo 핀 모드 설정
  pinMode(5, OUTPUT); // Red LED
  pinMode(6, OUTPUT); // Blue LED
  pinMode(7, OUTPUT); // Green LED
}

void loop()
{
  float Length, distance; // Length와 distance 지역변수로 선언

  // trigger 신호 발생(10us)
  digitalWrite(trigPin, LOW);
  delay(2);
  digitalWrite(trigPin, HIGH); // trigger 신호 발생(10us)
  delay(10);
  digitalWrite(trigPin, LOW);
```

```
// Echo 신호 입력
/* pulsIn()은 아두이노 기본 함수, 입력신호가 High 또는 Low가 되는 시간을 측정
 * pulseIn 함수를 통해 echo핀에 초음파가 돌아오는 시간을 측정*/

Length = pulseIn(echoPin, HIGH);

// 거리 계산, 거리를 계산하는 공식을 코딩, 10000은 cm로 단위 변환을 위해 사용함
distance = ((float) (340*Length)/10000)/2;

// Serial 모니터로 출력, 데이터 확인을 하려면 아두이노 창의 맨 오른쪽 위 검색 버튼
Serial.print(distance);
Serial.println(" cm");

// 1sec마다 출력
delay(100);

if(distance < 15){
  digitalWrite(7, LOW); // R
  digitalWrite(6, HIGH); // G
  digitalWrite(5, LOW); // B
}
else {
  digitalWrite(7, HIGH);
  delay(100);
  digitalWrite(7, LOW);
  delay(100);
  digitalWrite(6, HIGH);
  delay(100);
  digitalWrite(6, LOW);
  delay(100);
  digitalWrite(5, HIGH);
  delay(100);
  digitalWrite(5, LOW);
}
```

```
delay(100);
if(distance > 15)
{
  tone(buzzer, 262, 500);
  delay(500);
}

int cdsValue = analogRead(cds);

Serial.print("cds=");
Serial.println(cdsValue);

if(cdsValue > 50)
{
  digitalWrite(led, HIGH);
  Serial.println("LED ON(cds > 50)");
}
else
{
  digitalWrite(led, LOW);
  Serial.println("LED OFF(Cds < 50)");
}
else
{
  digitalWrite(led, LOW);
  Serial.println("LED OFF(cds < 50)");
}
delay(200);
}
```

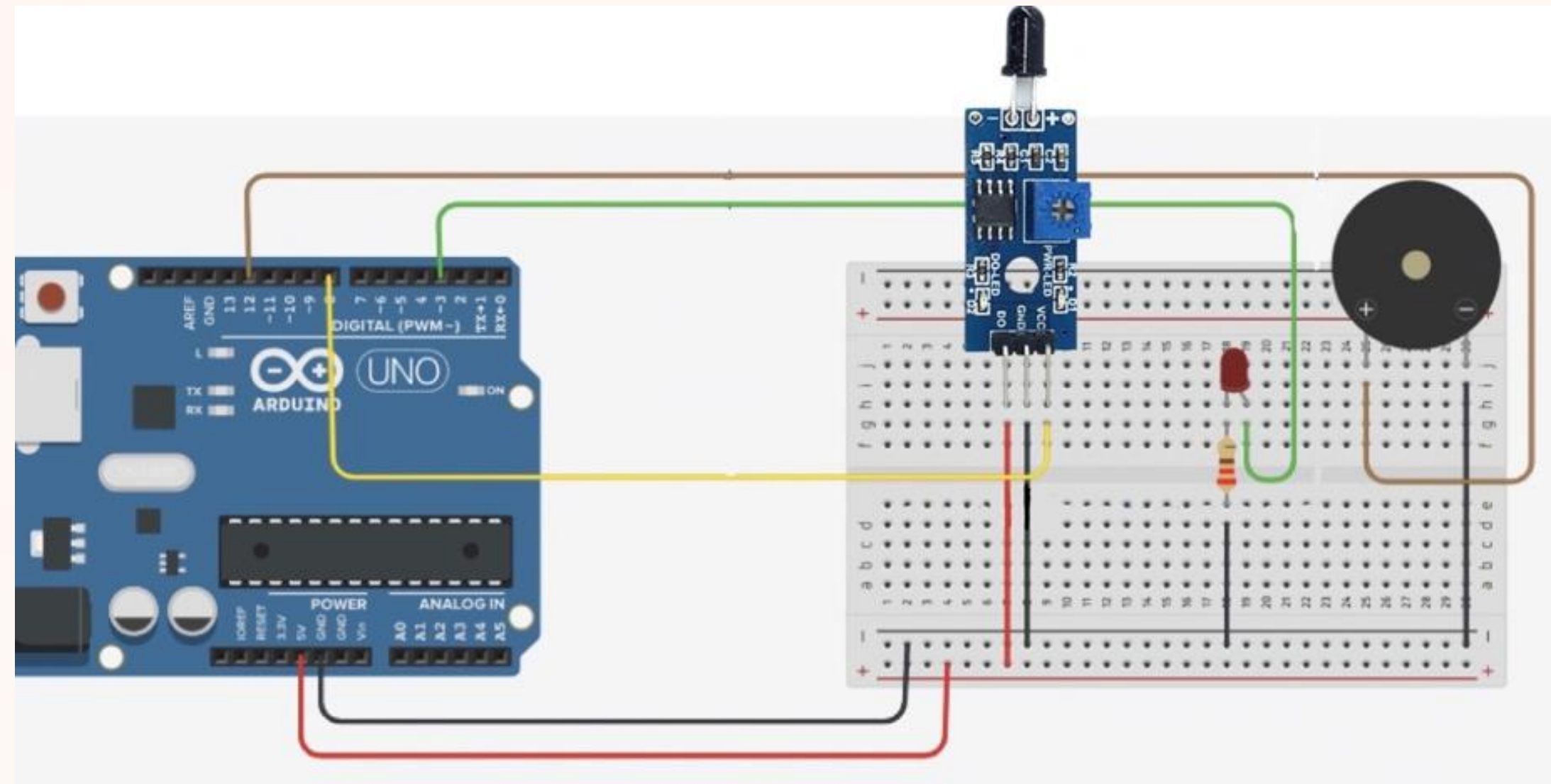
# A Fire Alarm

- **Placing and Purpose**

Designed to prevent fires in the yard

- **Used Sensors and kits**

Arduino UNO , Flame detection Module , Piezo buzzer, LED



300%

# A Fire Alarm

```
// 화재 경보기

int led=7;    // LED 핀 번호 7번
int flame=8;  // 화재 감지 센서 핀 번호 8번
int buzzer=9; // 스피커 핀 번호 6번
int value=0;  // 초기 value 값: 0

void setup(){
  pinMode(led, OUTPUT);    // LED 출력
  pinMode(flame, INPUT);   // 화재 감지 센서 입력
  pinMode(buzzer, OUTPUT); // 스피커 출력
  Serial.begin(9600);
}

void loop(){
  digitalWrite(led, LOW); // 기본 LED 상태: 꺼짐
  noTone(buzzer);         // 기본 스피커 상태: 꺼짐
  value=digitalRead(flame);
```

```
  if(value>0){ // 화재를 감지 했을 때
    Serial.println("DANGER"); // Danger를 시리얼 모니터에 출력
    digitalWrite(led, HIGH);  // LED 상태: 켜짐
    tone(buzzer, 490, 100);    // 스피커 상태: 490Hz 출력
    delay(100);
    digitalWrite(led, LOW);    // LED 상태: 꺼짐
    noTone(buzzer);            // 스피커 상태: 꺼짐
    delay(100);
  }
  else{ // 평상시
    Serial.println("SAFE");    // Safe 문구를 시리얼 모니터에 출력
    digitalWrite(led, LOW);    // LED 상태: 꺼짐
    delay(100);
    noTone(buzzer);            // 스피커 상태: 꺼짐
    delay(100);
  }
  delay(1000);
}
```

# Impact Sensor

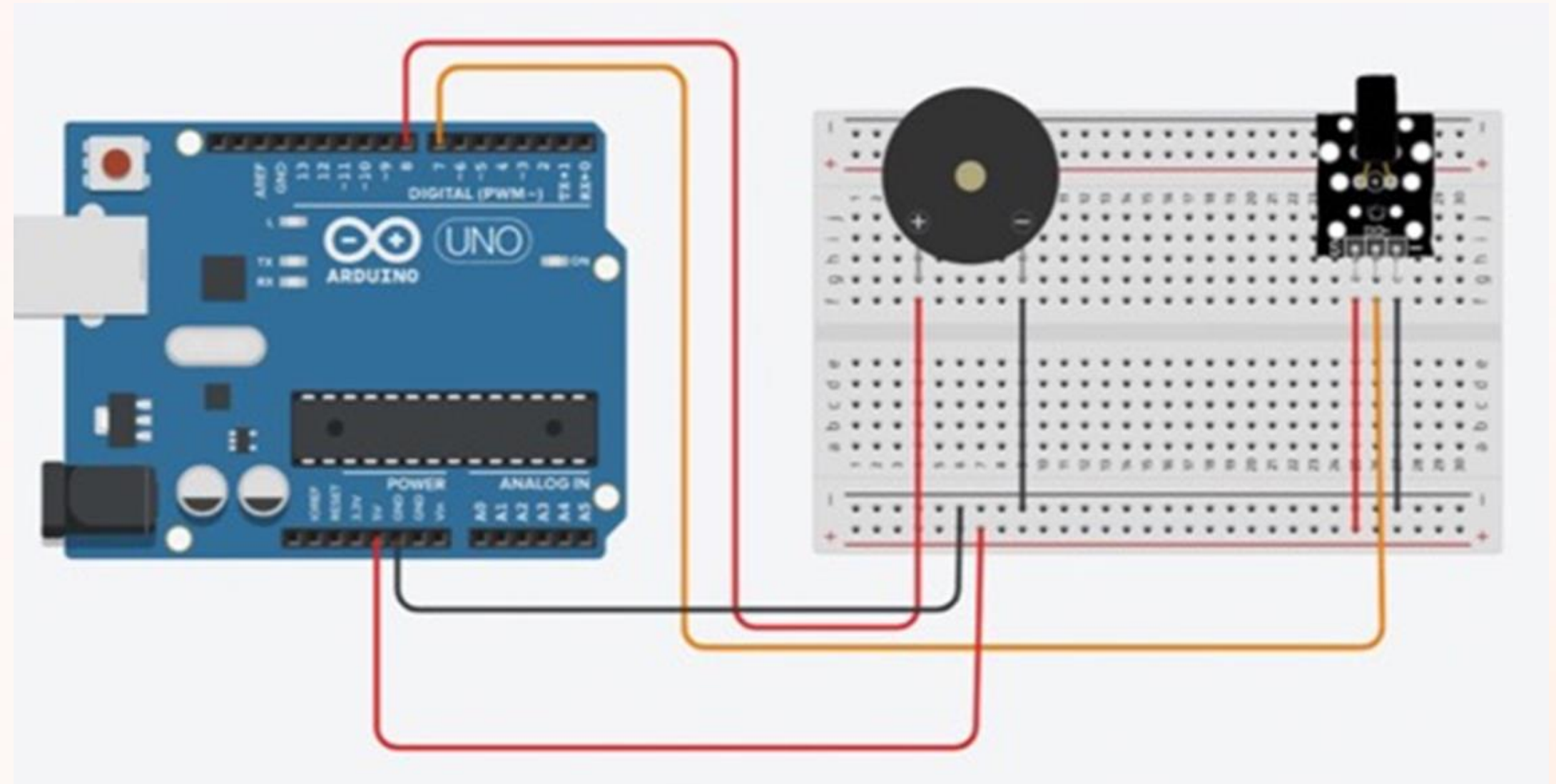
## - Placing and Purpose

## Placed Basement

# To warn users when an earthquake detects a shock to their home

## - Used Sensors and kits

# Arduino UNO , Shock Detection Module , Piezo buzzer



# Impact Sensor

```
// 충격 감지 센서

#define SHOCK 7

int speakerPin = 8;
int count = 0;

void setup()
{
    Serial.begin(9600); // 시리얼 통신 사용
    pinMode(SHOCK, INPUT);
    pinMode(speakerPin, OUTPUT);
}

void loop()
{
    if(digitalRead(SHOCK) == HIGH)
    {
        Serial.println("SHOCK!"); // Serial 모니터: SHOCK! 출력
        tone(speakerPin, 261);      // 스피커 상태: 261Hz 출력
        delay(500);
        noTone(speakerPin);         // 스피커 상태: 출력 x
        delay(500);
        count ++;

        if (count == 5)
        {
            // 충격 받으면 부저 울리고 5번 누적 시 부저 높은 음으로 지속
            Serial.println("WARNING"); // Serial 모니터: WARNING 출력
            tone(speakerPin, 523);      // 스피커 상태: 523Hz 출력
            delay(2000);
        }
    }
    noTone(speakerPin);
}
```



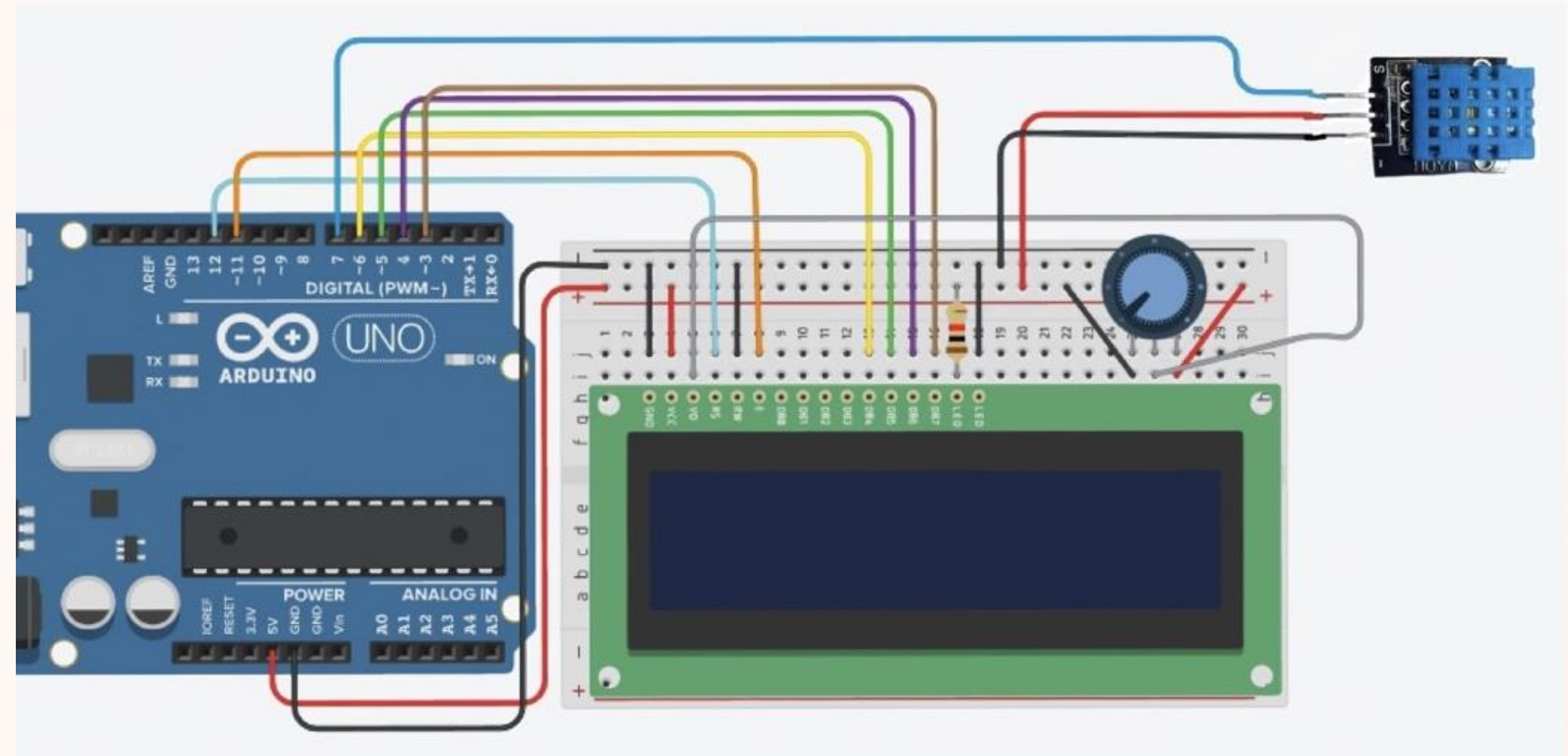
# Humidity Sensor & LCD

## - Placing and Purpose

Measure the humidity in the room and output it to the LCD monitor

## - Used Sensors and kits

Arduino UNO , DHT11 , Potentiometer, LCD



300%

# Humidity Sensor & LCD

```
// 습도 센서 + LCD

#include <DHT11>
#include <LiquidCrystal.h>
LiquidCrystal lcd(12, 11, 5, 4, 3, 2); // LCD 활용 핀
DHT11 dht11(7);

void setup()
{
    Serial.begin(9600); // 시리얼 통신 사용
    lcd.begin(16,2);    // LCD 화면 시작 위치
}
```

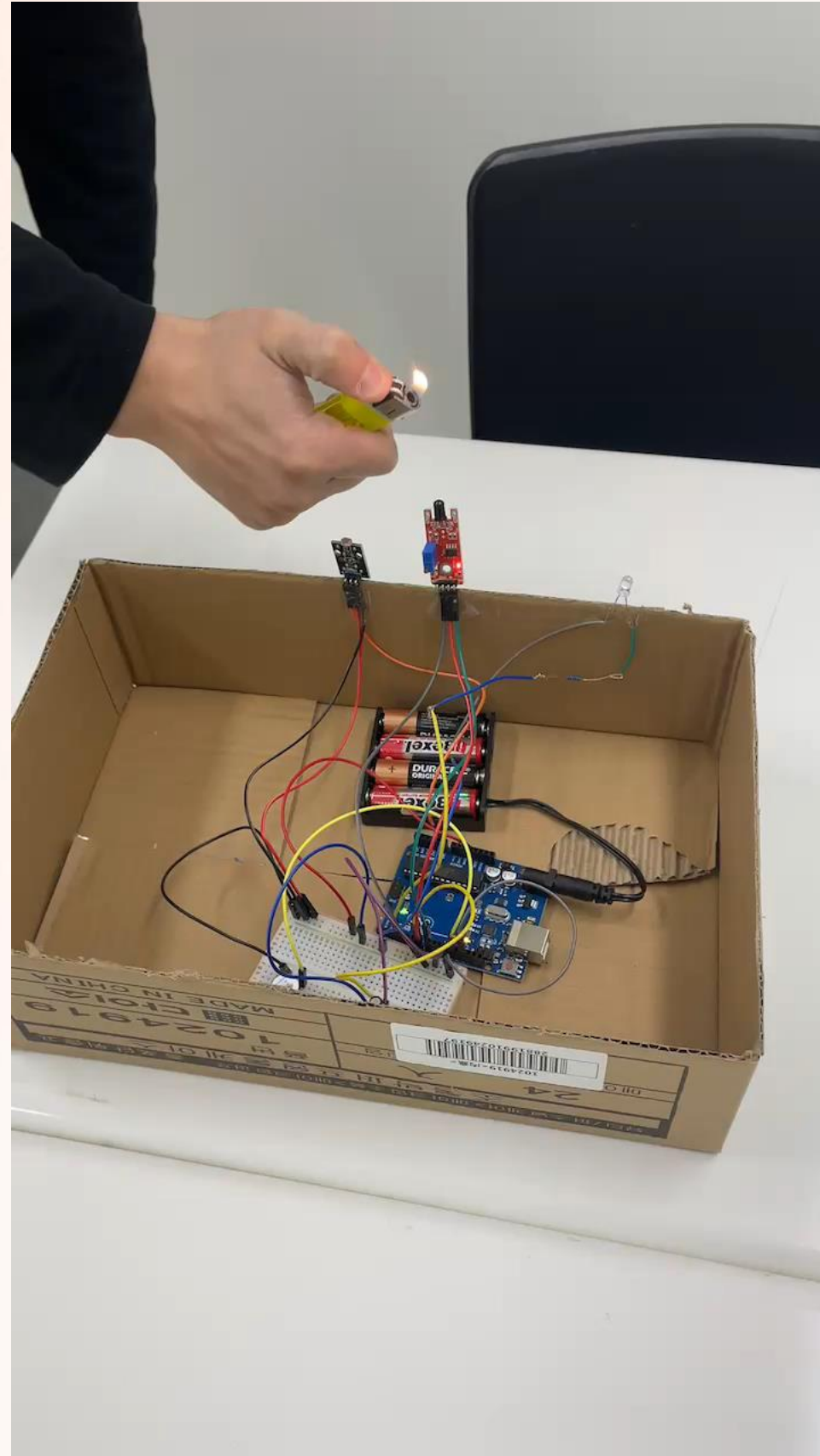
```
void loop()
{

    float h = dht.readHumidity();
    float t = dht.readTemperature();
    lcd.setCursor(0,0); // 0번째 줄 0번째 셀부터 출력
    lcd.print("Humi:"); // LCD 화면 Humi: 출력
    lcd.print(h);
    lcd.print("%d");

    lcd.setCursor(0,1); // 0번째 줄 1번째 셀부터 출력
    lcd.print("Temp:"); // LCD 화면 Temp: 출력
    lcd.print(t);
    lcd.print("C");
    delay(2000);
}
```

00  
동작

# Video



<https://youtu.be/wQKwCZvPQqg>



# Video



# Learned About

**이정민** : 우리 주변에서 쓰이는 IoT 기기들을 직접 만들고, 센서들을 통해 데이터 값들이 시리얼 모니터에 출력되는 것을 보고, 어떠한 원리로 IoT 기기들이 동작하는지 알게 되었으며 또한 우리가 만들만한 또다른 IoT 기기들은 무엇이 있을지 아이디어를 도출해낼 수 있는 귀중한 시간이었다. 비록 LCD 모니터에 출력하는 과정에서 저항 값 문제로 정상적인 출력문이 나오지 않았지만, 이를 해결하는 과정에서 나의 수행방식에 어떠한 착오가 있었는지 바로잡고 수정할 수 있어 뿌듯함을 느꼈다.

**이재용** : 사전에 역할을 분담하고 서로 이야기하며 의견을 조율하는 과정이 좋았다. 직접 작업물을 만들며 아두이노에 대해 이해도가 높아지는 프로젝트였다.



# Learned About

**김유현** : 한 달 정도의 시간을 활용하여 미숙하지만 완성된 작품을 만들게 된 것에 대해 보람을 느끼며  
실생활에 활용 할 수 있는 아두이노 센서가 많다는 점을 새롭게 알게 되었습니다.

**문희진** : 수업시간에 다루지 않은 센서들을 이용하는 데에 어려움이 있었지만 관련 정보를 찾으면서 직접 해보니  
다른 부품들에 대한 이해도 수월해져 많은 도움이 되었다. 회로를 연결하며 주어진 자료만 보는 게 아니라  
회로를 파악하며 연결해야 한다는 것도 깨닫게 되었다.

**Thank you**