

스마트 화재경보기

smarcle

~ 5조 ~

CONTENTS



01

3D프린팅

- 과정
- 결과



02

코딩 및 회로 정리

- 코딩
- 회로 정리



03

앱 개발

- 과정
- 결과

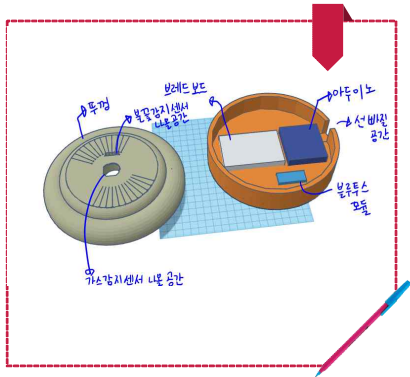


04

최종 완성

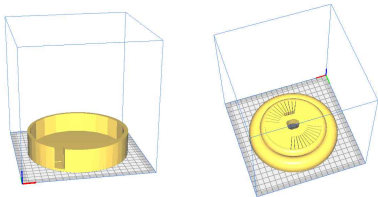
- 1차 완성
- 2차 완성
- 최종 완성

과정 및 결과



<과정>

1. Cad 프로그램을 통해 모델링
2. 모델링 한 것을 파일명.stl 파일로 저장
3. 이를 슬라이싱 프로그램을 이용해 기본 설정 및 파일명.zcode로 저장
4. 3D프린터를 통해 출력



출력된 사진 못찍음ㅠㅠㅠ

코딩

```

#include <Arduino.h>
#include <SoftwareSerial.h>

#define BT_RXD 9
#define BT_TXD 8
SoftwareSerial btSerial(BT_RXD, BT_TXD);

// 13번 핀을 LED로 지정합니다.
const int ledPin = 13;
// 디지털 입력을 분할용 저항과 함께 지정합니다.
const int inputPin = 2;
// 불꽃 감지 센서의 상태를 지정합니다. (여름 상태를 LOW로 설정)
int pirState = LOW;
// 센서 값을 읽기 위해 변수를 선언합니다.
int val = 0;
// 핀을 통해 스피커 혹은 피에조 센서를 연결합니다. (speaker 9, 12, or 11)
int pinSpeaker = 10;
int melody = 0; // 가스센서 입력을 위한 피에조 센

float val2;

void setup() {
  // LED를 출력으로 설정합니다.
  pinMode(ledPin, OUTPUT);
  // 센서를 입력으로 설정합니다.
  pinMode(inputPin, INPUT);
  // 스피커를 출력으로 설정합니다.
  pinMode(pinSpeaker, OUTPUT);
  // 피에조 센서(PIEZOE)를 연결합니다.
  pinMode(melodyPin, INPUT); // 피에조 센서 값을 입력모드로 설정

  Serial.begin(9600);
  btSerial.begin(9600);
}

void loop() {
  if (btSerial.available()) {
    Serial.write(btSerial.read());
  }
  if (Serial.available()) {
    btSerial.write(Serial.read());
  }
}

```

```

// 센서값을 읽어옵니다.
val = digitalRead(inputPin);
Serial.println(analogRead(A02pin)); // 가스센서로부터 아날로그 데이터를 받아와 시리얼 모니터로 출력함
btSerial.println(analogRead(A02pin));
delay(1000); // 1s 대기

```

```

val2 = analogRead(A02pin);
if (val2 > 500) {
  Serial.print("연가깝지!!!");
  btSerial.write("연가깝지!!!");
  delay(1000);
  playTone(300, 160);
  delay(150);
  Serial.println("");
}
delay(2000);

```

```

// 만약 값이 HIGH 일때,
if (val == HIGH) {
  // 13번 핀(보드에 내장되어 있는 LED)을 켜줍니다.
  digitalWrite(ledPin, HIGH);
  playTone(300, 160);
  delay(150);
}

```

```

if (pirState == LOW) {
  // 센서의 상태가 LOW일때
  Serial.println("FIRE!!!!");
  btSerial.write("FIRE!!!!");
  delay(1000);
  // 다음의 문구를 시리얼 모니터로 출력합니다.
  pirState = HIGH;
}

```

```

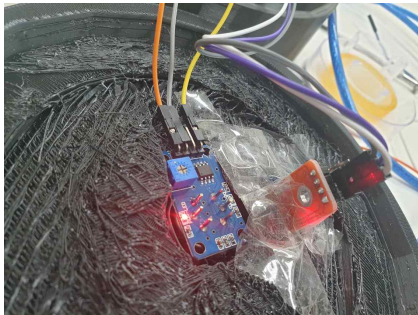
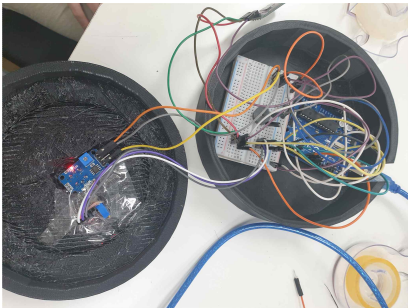
} else {
  // LED를 끕니다.
  digitalWrite(ledPin, LOW);
  playTone(0, 0);
  delay(30);
  if (pirState == HIGH) {
    // 센서값이 HIGH 일때
    Serial.println("Beware of fire.");
    btSerial.write("Beware of fire.");
    delay(1000);
    // 다음의 문구를 시리얼 모니터로 출력합니다.
    pirState = LOW;
  }
}

// 경보음을 만들어 냅니다.
void playTone(long duration, int freq) {
  duration /= 1000;
  int period = (1.0 / freq) * 1000000;
  long elapsed_time = 0;
  while (elapsed_time < duration) {
    digitalWrite(pinSpeaker, HIGH);
    delayMicroseconds(period / 2);
    digitalWrite(pinSpeaker, LOW);
    delayMicroseconds(period / 2);
    elapsed_time += (period);
  }
}

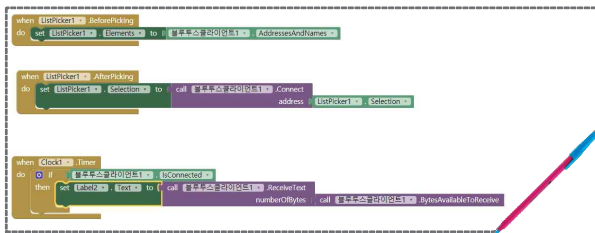
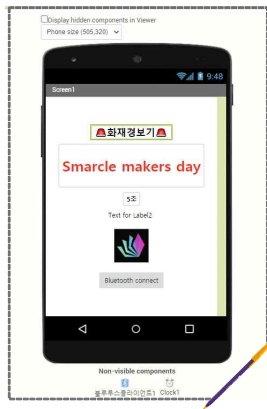
```

02 코딩 및 회로 정리

회로 정리



과정 및 결과



앱 인벤터 프로그램을 이용해 가스의 수치를 보여주고

불꽃의 여부를 표시할 수 있도록 프로그래밍

04

최종 완성

1차 완성

가스 감지 1차 ☺

불꽃 감지 1차 ☺

04

최종 완성

2차 완성

가스 감지 2차 ☺

불꽃 감지 2차 ☺

04

최종 완성

최종 완성



가스 감지 최종 ☺

불꽃 감지 최종 ☺



감사합니다

Q & A

smarcle

~ 5조 ~