

환경구축

코딩

3D 모델링

3D 프린팅

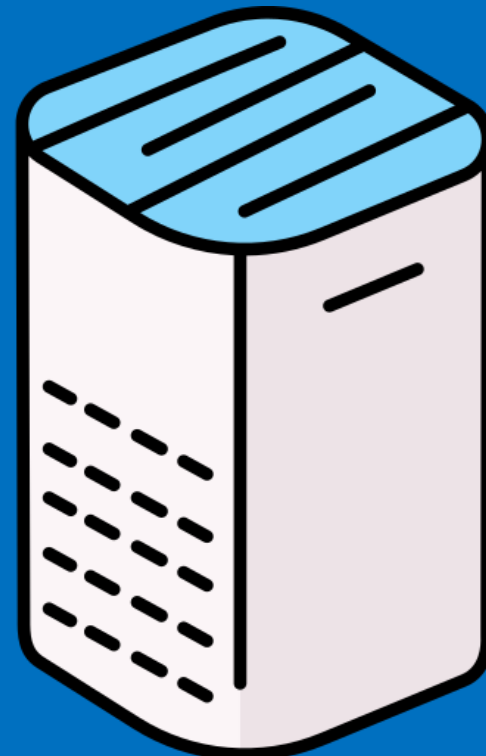
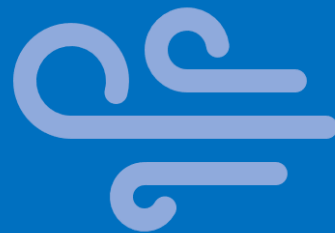
For Future Smart Home

2022 라이프 스타일 스마트 가전 메이커톤

IoT 스마트 제어 공기청정장치 설계 및 제작

공기 UP!

이소정 한재덕 신병근





목 차



환경구축

코딩

3D 모델링

3D 프린팅

배경

- 제작 취지 및 기획 의도
- 접근 방식

설계

- 하드웨어 구성
- 작동 시나리오 및 코드 구성
- 모델링

제작

- 제작 과정

결과물

작동

- Optional 미션 달성 결과
- 동작 시연

차별성

- 결과물의 차별점 및 특징
- 상품으로서의 경쟁력

발전방향

- 개선 방향 및 발전 가능성

» 가장 효과적인 위치에서 맑은 공기와 밝은 빛을 제공한다.

가장 효과적인 위치

미세먼지 고르게, 많이 없애려면 '가운데'가 효과적

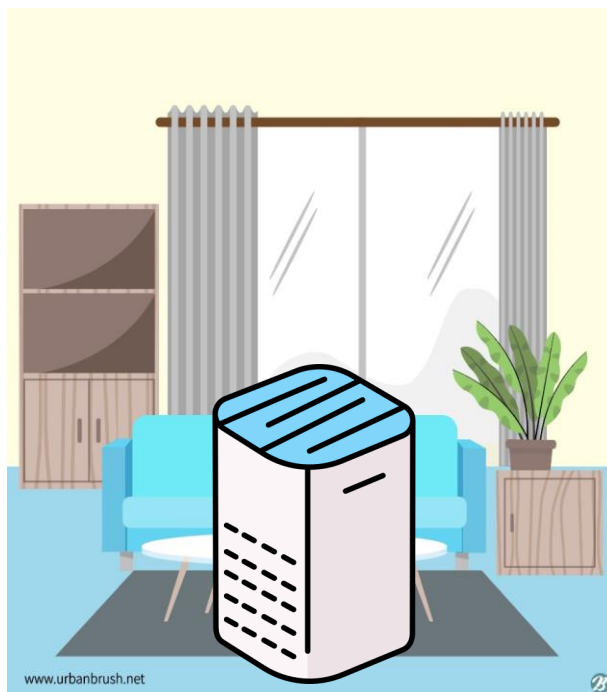
이진욱 웅진코웨이 연구원은 “공기청정기 위치 실험이 주는 결과는 단순하지만 명확하다”며 “미세먼지를 구석구석 고르게, 보다 많이 없애고 싶다면 되도록 해당 공간의 가운데에 두고 가동하는 게 효과적”이라고 말했다. 각종 광고에 등장하는 공기청정기가 거실 한 가운데에 주인공처럼 놓여있는 모습이 사실은 ‘올바른 사용법’이었던 셈이다.

[출처] 중앙일보

공기청정기, 어디에 두는 게 효과적일까?

✓ **공기청정기는
공간의 한가운데 있을 때
가장 효과적!**

공간의 효율을 고려한 배치



✓ BUT! 공간의 한가운데 놓으면 공간의 측면에서 비효율적

**공간과 공기청정의 효율을
모두 고려한다면?**

천장형 공기청정기 + 조명



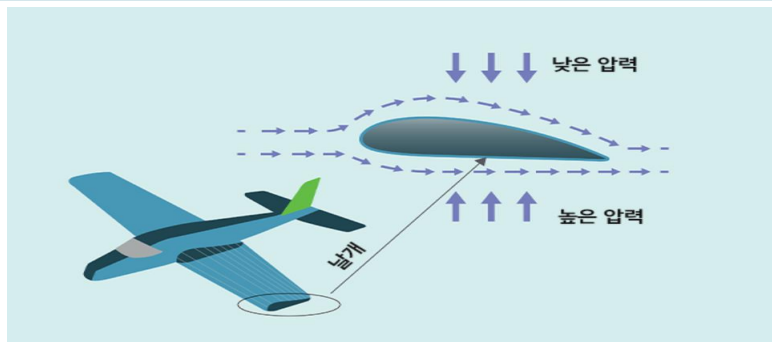
✓ **천장에서의 효율까지 고려한
조명과 공기청정기의 결합**

➔ **공기청정효과, 공간의 효율과
인테리어적 측면 고려**

접근 방식

» 보다 효율적인 공기청정을 위한 유체역학 적용

제트엔진 (베르누이 원리)



유체가 빠른 속력으로 흐르면 압력 감소
느린 속력으로 흐르면 압력 증가

흡입구 적용

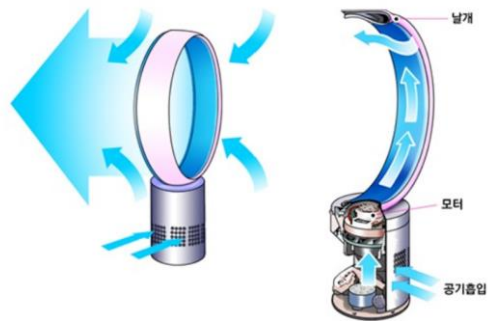
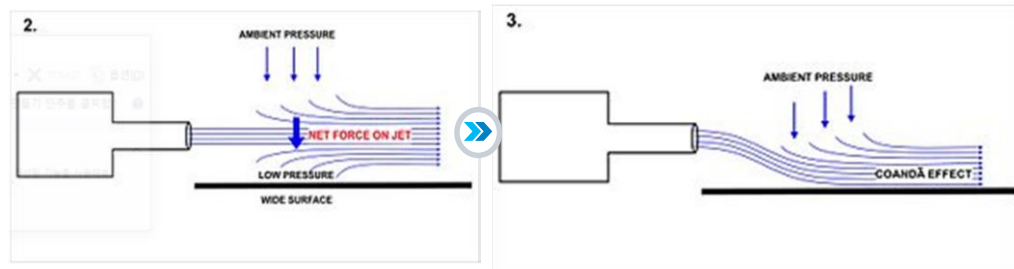


사진 출처=네이버 지식백과

본체에서 공기 흡입하여 원통으로 공기 보냄
→ 공기 속도 증가, 주변 공기까지 이동시켜 바람발생

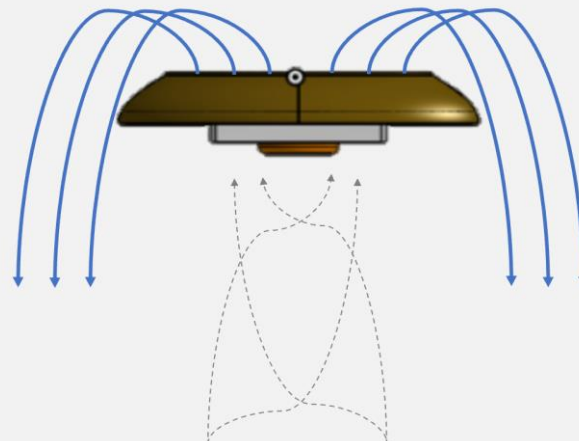
코안다 효과



유체 바로 아래에 벽 존재
아래쪽 압력 X

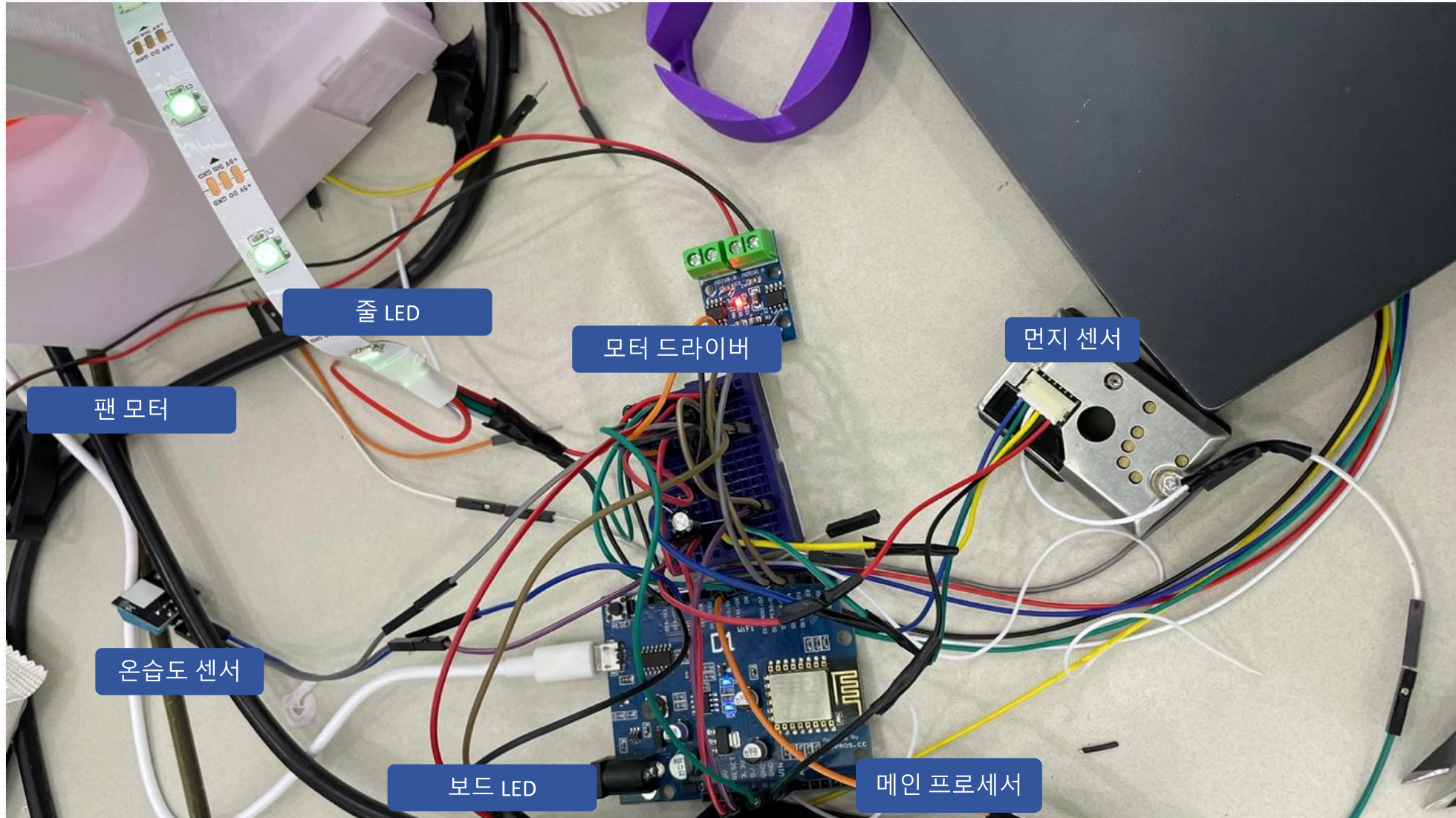
유체 벽 쪽에 붙어서 흐름
(곡면 존재해도 같은 원리)

배출구 적용



천장면을 따라서 분출된 기류는 그 면에 부착해서 흐르는 경향
→ 청정공기의 흐름의 속도 감쇠가 적고 도달거리가 증가하는 효과 기대

실제 회로 구성



작동 시나리오

작동 프로세스

스마트 공기청정기

IoT 스마트 제어 시스템

센서부

먼지 센서



온습도 센서



모터 제어



LED 제어



Wemos R1 D1



humidity: 57

number: 603

temperature: 28



미세먼지 농도 보정

실시간 데이터 처리

스마트 공기청정

Blynk 대시보드



실시간 대기질
모니터링



원격 기기 제어



예약 및 자동 정화



데이터
관리

온도

먼지

습도

상태값

Meta
Data Repository

인프라

Blynk 2.0



Blynk.io

WeMos R1 D1

2. 코드 구성

코드 아키텍처

Wemos R1 D1



Blynk 웹 서버

ID	Label	Value	Unit	Min	Max	Step	Color	Icon
1	Temp	25.0	°C	0	50	0.1	Blue	thermometer
2	Humidity	65	%	0	100	1	Green	humidity
3	Light	100	lx	0	1000	10	Yellow	light
4	Door	0	bool	0	1	1	Red	door
5	Fan Speed	50	rpm	0	100	5	Blue	fan
6	LED	0	bool	0	1	1	Green	led
7	Motor	0	bool	0	1	1	Red	motor
8	Relay	0	bool	0	1	1	Blue	relay
9	Switch	0	bool	0	1	1	Green	switch
10	Button	0	bool	0	1	1	Red	button
11	Slider	50	%	0	100	5	Blue	slider
12	Color	0	bool	0	1	1	Green	color
13	Temperature	25.0	°C	0	50	0.1	Blue	thermometer
14	Humidity	65	%	0	100	1	Green	humidity
15	Light	100	lx	0	1000	10	Yellow	light
16	Door	0	bool	0	1	1	Red	door
17	Fan Speed	50	rpm	0	100	5	Blue	fan
18	LED	0	bool	0	1	1	Green	led
19	Motor	0	bool	0	1	1	Red	motor
20	Relay	0	bool	0	1	1	Blue	relay
21	Switch	0	bool	0	1	1	Green	switch
22	Button	0	bool	0	1	1	Red	button
23	Slider	50	%	0	100	5	Blue	slider
24	Color	0	bool	0	1	1	Green	color

Blynk 대시보드



<송신부>

- 센서 라이브러리 정의
- WiFi를 이용한 blynk와 웹서버 연결
- 먼지 센서, 온/습도 센서, 장치 상태 값을 읽고 blynk로 전송

<수신부>

- blynk의 제어 요소가 업데이트 될 때 마다 데이터 수신
- 수신 값에 따라 FAN 속도, LED 등 제어

<제어부>

- 스마트 모드 활성화 혹은 웹서버와 연결 안될 시, 기본적으로 환경에 따라 센서 제어

<데이터부>

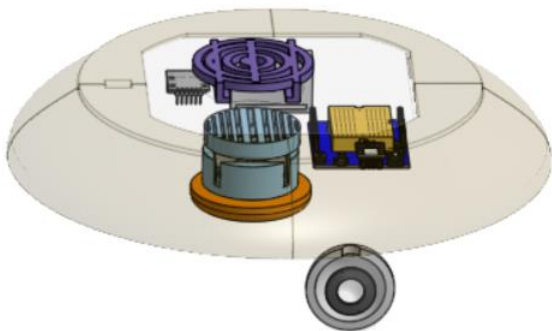
- 기기에 연결된 WIFI를 통해 어디서나 기기의 데이터를 수신
- 기기의 모든 센서 장치 및 변수 데이터 스트림
- 가상핀을 통한 하드웨어와의 입출력 통신
- 데이터 스트림, 오토메이션, 대시보드와의 연계로 하드웨어 컨트롤 수행

<모니터링부>

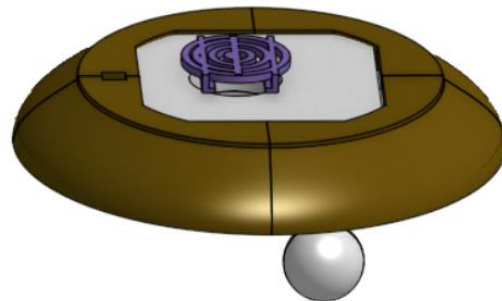
- 기기로부터 현재 연동되는 시간 및 센서 데이터 값 시각화
- 전원, 운전 모드 등 각 제어 변수들의 실시간 시각화

<제어부>

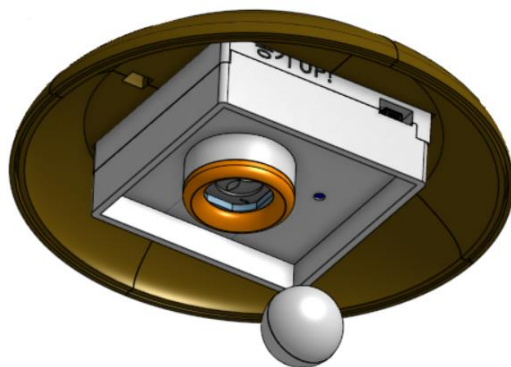
- 자동 운전 모드 지원
- 각 제어 변수 실시간 컨트롤
- 기기 연동시간에 맞춰 예약 시간대 작동 설정 가능



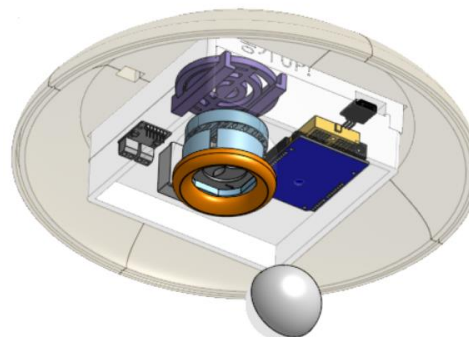
<정면> 배치도



<정면> 전체 모습



<후면> 배치도



<후면> 전체 모습

제작 과정

배경

설계

제작

결과물

작동

차별성

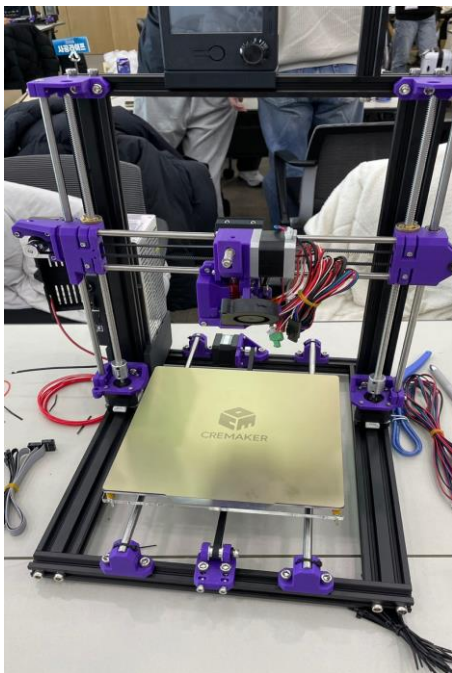
발전방향

1일차

3D 프린터 조립 및 작동

- 3D 프린터 조립 및 배선작업
- 예시 작업을 출력

3D 프린터 조립 [1일차]

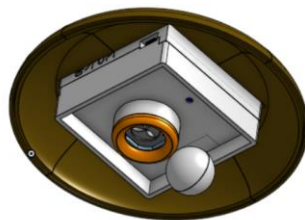
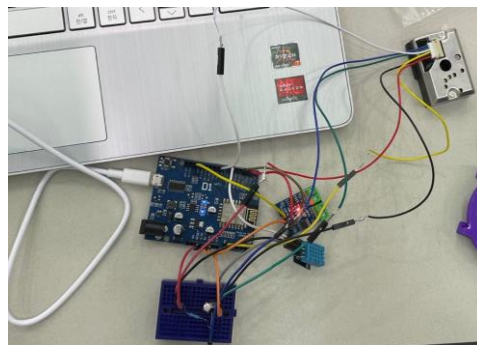


2일차

3D 모델링 및 코드 작동

- 고지된 미션 이해
- 회로 및 코드 구성 및 작동 확인
- 3D 프린팅 및 모델링

3D 모델링, 회로 및 코드 작동 확인 [2일차, 3일차]



3일차

회로 구성 및 앱, 웹 구현

- 3D 프린팅 및 모델링
- 회로 배선 작업
- 코드 상세히 구현 및 웹, 앱 구현

4일차

마무리, 점검

- 전체 조립 후 점검

```
1 #include <GP2YDustSensor.h>
2
3 const uint8_t SHARP_LED_PIN = D6; // Sharp Dust/particle sensor Led Pin
4 const uint8_t SHARP_VO_PIN = A0; // Sharp Dust/particle analog out pin used for reading
5
6 GP2YDustSensor dustSensor(GP2YDustSensorType::GP2Y1014A0UF, SHARP_LED_PIN, SHARP_VO_PIN);
7
8 GP2YDustSensor setBaseline(float zeroDustVoltage)
9 {
10 }
11
12 void setup() {
13   Serial.begin(115200);
14   dustSensor.setBaseline(0.5); // set no dust voltage according to your own experiments
15   //dustSensor.setCalibrationFactor(1.1); // calibrate against precision instrument
16   dustSensor.begin();
17 }
18
19 void loop() {
20   Serial.print("Dust density: ");
21   Serial.print(dustSensor.getDustDensity());
22   Serial.print(" ug/m3; Running average: ");
23   Serial.print(dustSensor.getRunningAverage());
24   Serial.print(" ug/m3");
25   delay(1000);
26 }
```

실생활 적용

AIR
PURIFIER

+

Illumination

AIRLUMINATION

미세먼지 농도 별 상태등 표시

좋음

0~30 [$\mu\text{g}/\text{m}^3$]

보통

~70 [$\mu\text{g}/\text{m}^3$]

나쁨

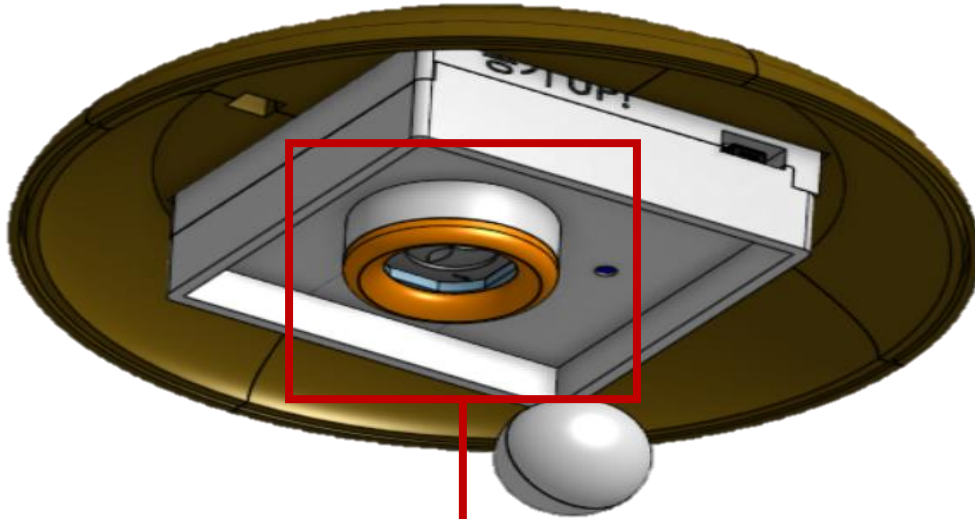
~150 [$\mu\text{g}/\text{m}^3$]

매우 나쁨

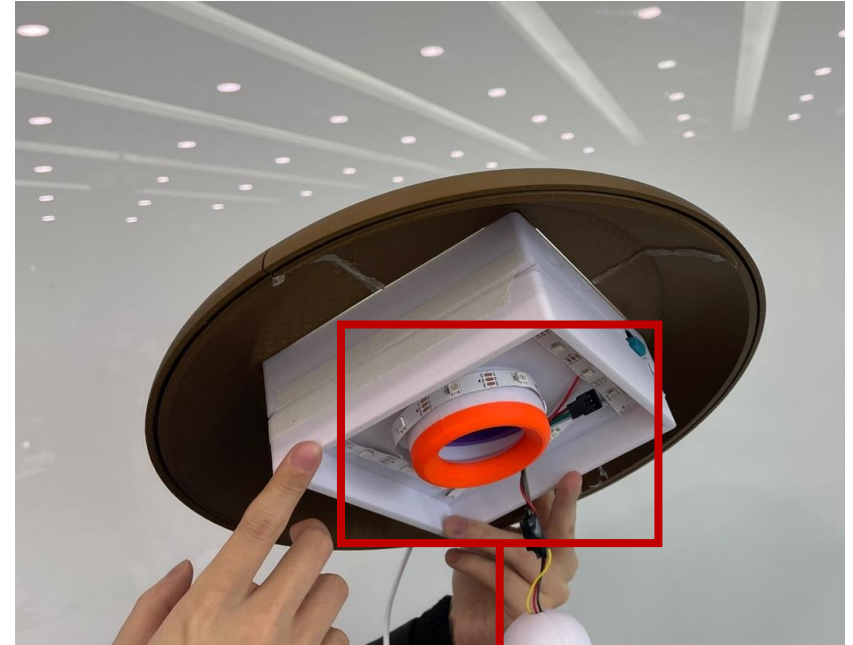
151 [$\mu\text{g}/\text{m}^3$]~

Optional mission 달성 결과

[Mission #6_Optional] 3D구성품 중 적어도 하나에 출력중 Filament Change 기능 사용



[모델링]



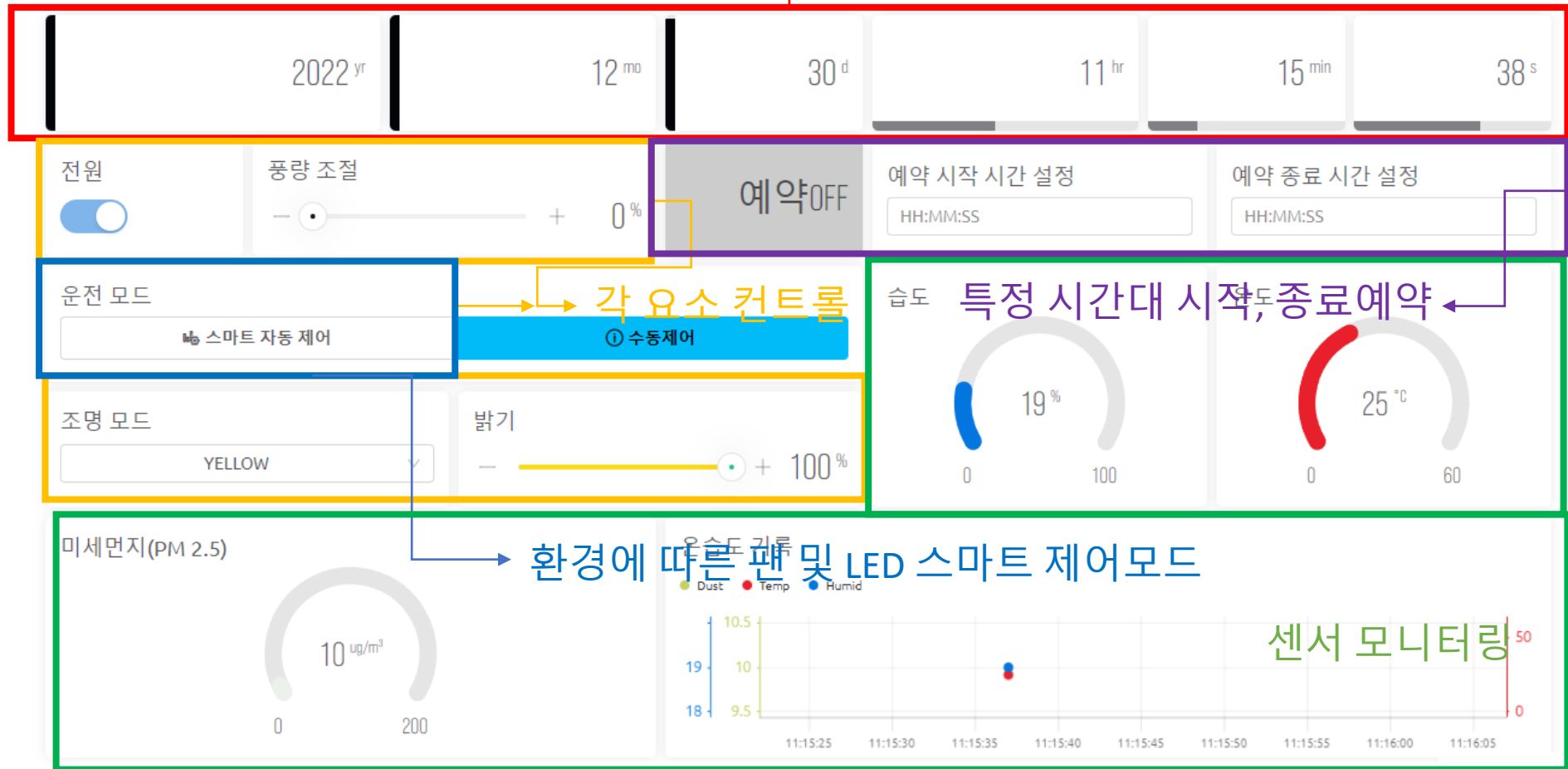
[실제 구현]

흡입구 부분을 filament
change로 구현

2. Optional mission 달성 결과

[Mission #4_Optional] IoT 장치에 기능모드를 구현하고 원격에서 설정할 수 있도록 구현

기기의 현재 장치 시간 출력



Optional mission 달성 결과

[Mission #3_Optional] 온도/습도 측정값을 활용하여 미세먼지 농도값을 적절하게 보상

- ✓ 습도 높아지면서 미세먼지 습기 흡수하며 커짐
→ 광산란 방식에서는 크기 2배 증가하면 무게 (농도) 8배 증가

The mean averages of the PA PM_{2.5} concentrations at each site tended to be higher than their FEM counterparts in both datasets, except at the lowest humidity levels (Fig. 3). For the training data, 23 % of the hourly observations were classified as “dry” ($RH \leq 33\%$), 56 % were classified as “moderate” ($30\% < RH < 70\%$), and 21 % were classified as “humid” ($RH \geq 70\%$).¹

[출처] Development and evaluation of correction models for a low-cost fine particulate matter monitor, Brayden Nilson, 2022

습도 보상

Humid : 보상계수 더해줌
Dry : 보상계수 빼줌

Dry : 30% 이하
Moderate : 30% ~ 70%
Humid : 70% 이상

- ✓ 계절별 미세먼지 농도 차이에 따르면 온도가 높을 때 미세먼지가 덜하다고 예측가능

초미세먼지 PM-2.5 (μg/m ³)					
구분		봄(3,4,5)	여름(6,7,8)	가을(9,10,11)	겨울(12,1,2)
서울시 평균	18	20	13	16	24

온도 보상

Room Temperature (25°C) 기준

25 °C 이상 : 보상계수 T1
25 °C 이하 : 보상계수 T2

기준 보상계수 T값
(800 - 100)/800 = 0.875

Optional mission 달성 결과

[Mission #3_Optional] 온도/습도 측정값을 활용하여 미세먼지 농도값을 적절하게 보상

가정

온도가 높을 때 미세먼지 농도 90
온도가 낮을 때 미세먼지 농도 110

습도가 humid 상태라 800이라
가정

$$T1 = (800-90)/800 = 0.8875$$

$$T2 = (800-110)/800 = 0.8625$$

〈코드 구현〉

```
int dustCalibration(int humid, int temperature, double dust){  
    float t = 0;  
    if (temperature > 25){  
        t = dust - dust*0.8875;  
        if (humid <= 30){  
            return dust+t;  
        }  
        else if (humid > 30 and humid < 70){  
            return dust;  
        }  
        else{  
            return dust-t;  
        }  
    }  
    else{  
        t = dust - dust*0.8625;  
        if (humid <= 30){  
            return dust+t;  
        }  
        else if (humid > 30 and humid < 70){  
            return dust;  
        }  
        else{  
            return dust-t;  
        }  
    }  
}
```

〈실제 결과 반영〉

Humid: 29

Temperature: 19

Dust density: 30.00

Dust Calibration: 34.00

Humid: 31

Temperature: 19

Dust density: 30.00

Dust Calibration: 30.00

습도가 dry 일 때
미세먼지 보정되어 값 증가

결과물로서 차별성 및 특징

제품 차별성

공기청정의 효과성 극대화

- 공간의 한 가운데 배치로 정화된 공기를 공간 전체에 보다 넓고 고르게 분배
- 코안다 효과와 베르누이 원리를 고려한 과학적 원리 이용하여 더욱 효과성 극대화
- 기존의 적층식 공기청정기와 달리 표면적을 넓이는 설계

공간의 효율성 고려한 배치

공기흐름
효과

+

공간
효율

- 천장 배치로 공기청정의 효과 극대화하면서 공간의 효율성까지 고려한 배치

전등과 결합하여 인테리어 효과

밝은 빛

+

맑은
공기

- 천장으로 올렸을 때 전등의 위치 고려하여 전등과 공기청정기의 융합

웹으로 제어 가능



- 수동으로 풍량 조절, 조명 색, 밝기 조정 가능
- 실시간으로 미세먼지, 온습도 값 모니터링 및 차트 기록

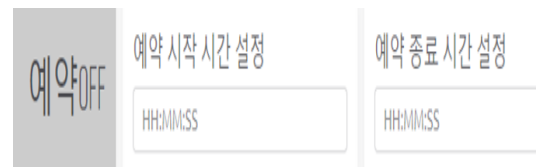
온도에 따른 팬 제어 기능

운전 모드

스마트 자동 제어

- 스마트 자동 제어 기능 선택 시 기존 스마트 공기청정기 기능 강화하여 주변의 환경에 맞게 풍량을 조절
- 온도가 25 °C 이상일 때부터 원래 팬에 온도의 10% power 가산해서 넣어주는 효과 추가

공기청정기의 예약기능



- 예약 시작 시간 설정 및 종료 시간 설정해 사용자가 원하는 시간대에 공기청정기 예약
- 수면시간, 휴식시간 등 사용자가 원하는 시간대에 예약 가능한 새로운 OPTION

SMART AIR PURIFIER AIRLUMINATION

감사합니다!

