

대한기계학회 주친

제12회 전국학생설계경진대회(2022년)

설계 최종 보고서

참가부	대학부 (☒)				
참가분야	공모주제 () / 자유주제 (☒)				
참가팀명	AK(Air Keeper)				
설계제목	스마트 환기 시스템				
지도교수/교사	(소속)세종대학교 (성명)전용태				
대표자 (신청인)	성명	소속	연락처 (휴대폰)	E-mail	주소
	장나경	세종대학교 기계공학과			

참가팀원 인적사항

NO	성명	소속 / 학년	E-MAIL
1	구승완	세종대학교 기계공학과 / 4학년	
2	김준혁	세종대학교 기계공학과 / 4학년	
3	이상수	세종대학교 기계공학과 / 4학년	
4	이승찬	세종대학교 기계공학과 / 4학년	
5			
6			

설계 요약문

참가분야	공모주제 () / 자유주제 (✓)
참가팀명	AK(Air Keeper)
설계제목	스마트 환기 시스템
대표자명	장나경
요약문	최근 호흡기 및 피부질환을 유발하는 미세먼지, 오염물질로 인한 대기오염의 심각성이 대두되고 있다. 더불어 코로나-19의 영향으로 집에서 생활하는 시간이 늘어나 실내 공기질을 청결하게 관리하는 것은 인간의 건강과 직결되는 문제로 자리 잡았다. 공기에 대한 국민의 걱정과 우려가 증가함에 따라 공기청정기의 시장은 급속도로 확대되었고, 팀 AK는 1인 가구에 초점을 맞춰 원룸, 오피스텔 등 온습도가 제대로 조절되지 않아 곰팡이가 생기기 쉬운 거주 공간에서 사용할 수 있는 ‘스마트 환기 시스템’을 제작해보고자 한다. 아이디어 등장 배경은 직장인과 학생들의 실생활에서 시작되었다. 출근(또는 등교)준비를 위해 샤워를 마친 직후 실내 온습도는 급격히 증가하여 환풍기의 작동을 필연적으로 요구하게 된다. 이 상황에서 종종 환풍기를 켜놓고 외출해버리는 상황이 발생하며 귀가 시간까지 환풍기는 끊임없이 작동하게 된다. 마찬가지로 취침 전 샤워를 마친 후 환풍기를 제때 끄지 못한 경우 아침까지 환풍기가 작동하는 것을 확인하고 서둘러 작동을 멈추는 상황이 있다. 팀 AK의 ‘스마트 환기 시스템’은 가정 내 온습도와 미세먼지, 이산화탄소와 같은 다양한 센싱 값을 측정해 쾌적한 공기 유지를 위한 기준치에 도달하게 도와주는 자동개폐창문과 환풍기를 설계할 예정이다. 센싱을 위한 모듈은 조립과 설치에 용이하도록 블록형으로 제작해 사용자의 희망에 따라 센서의 종류를 선택 및 추가 할 수 있으며, 개폐형창문의 날개 각도 조절을 통한 효과적인 환기가 이뤄지는 것을 목표로 하였다.

스마트 환기 시스템

장나경*·구승완*·김준혁*·이상수*·이승찬*·전용태**
*세종대학교 기계항공우주공학부 기계공학과

Smart Ventilation System

Na-Gyeong Jang*, Seungwan Gu*, Junhyeok Kim*,
Sangsu Lee*, Seungchan Lee* and Yongtae Jeon**

* School of Mechanical Engineering, Pusan National University

** School of Mechanical and Aerospace Engineering, Sejong University

Key Words: Fine Dust(미세먼지), Discomfort Index(불쾌지수), Turbomachines(터보기계)

초록: 최근 호흡기 및 피부질환을 유발하는 미세먼지, 오염물질로 인한 대기오염의 심각성이 대두되는 가운데, 코로나-19의 여파로 집에서 생활하는 시간이 늘어나며 실내공기질 관리는 인간의 건강과 직결되는 중요한 문제가 되었다. 실내오염물질은 자연환기와 기계식환기가 동시에 이뤄졌을 때 효과적으로 관리할 수 있으며, 실내공기질을 조절해주는 ‘스마트 환기 시스템’을 제작해보고자 한다. 공기상태를 자동으로 측정하고, 환기시스템을 on/off하는 번거로움을 해결함으로써 실내 공기를 관리할 수 있다. 측정된 수치를 애플리케이션을 통해 사용자가 관리할 수 있는 서버를 제공하여 오염물질로부터의 노출 위험성을 피하고, 쾌적하고 편리한 생활을 돕는다.

Abstract: As fine dust, pollutants, and COVID-19 increase the time to live at home, indoor air quality management has become an important problem. Indoor pollutants are effective when natural ventilation and mechanical ventilation are performed at the same time, and we want to produce a "smart ventilation system" that controls indoor air quality. Indoor air may be managed by measuring air conditions and automatically turning on/off the ventilation system. It provides a server that can manage measured values through an application to avoid danger from pollutants and help people live a pleasant and convenient life.

1. 서 론

최근 호흡기 및 피부질환을 유발하는 미세먼지, 오염물질로 인한 대기오염의 심각성이 대두되고 있다. 더불어 코로나-19의 영향으로 집에서 생활하는 시간이 늘어나 실내공기질을 청결하게 관리하는 것은 인간의 건강과 직결되는 문제로 자리 잡았다. 세계보건기구(WHO)는 폐암을 유발하는 미세먼지를 그룹1에 해당하는 발암물질로 지정했다. 그룹1은 인체에 발암성이 있다는 증거가 충분한 물질로 미세먼지와 경유차배출가스가 포함된다. AQI는 공기질 개선을 위한 대표적인 관리 대상을 온도, 먼지, 일산화탄소(CO)로 선정했고, 대상을 기준에 맞춰 조절함으로써 쾌적한 실내 공기질을 유지 및 조절하는 것을 목표로 하였다. 관리 대상은 환경부와 세계보건기구의 권고기준에 따라 조절한다. 실내 오염물질을 낮추는 방법은 세종대학교 기후에너지융합학과에서 진행한 실험 결과를 통해 양쪽 환기가 가장 효과적임을 확인했다. 코로나 19의 확산으로 바이러스 예방을 위한 환기는 필수적이고, 이는 대부분의 공기오염물질에도 해당된다.

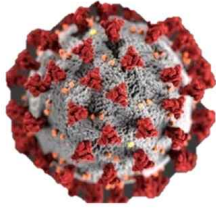


Fig 1 Increased need for ventilation to prevent coronavirus



Fig 2 Current status of fine dust



Fig 3 Relationship between indoor air quality and respiratory and skin diseases

대표적인 관리 대상을 온도, 먼지, 일산화탄소로 선정한 이유는 다음과 같다. 우선 매우 낮은 습도에서는 바이러스와 호흡기 질환을 퍼뜨리고, 높은 습도는 먼지 진드기와 곰팡이가 생기기 쉬운 환경이 조성되어 적절한 습도 조절은 필수적이다. 여름철 장마 기간과 겨울철 공기가 건조해 목이 갈라지는 상황을 떠올리면 습도와 온도가 일상생활에 어떤 영향을 주는지 대략 파악할 수 있고, 이는 불쾌지수로 정확하게 표현 가능하다. 불쾌지수로 68을 넘기면 불쾌감을 느끼기 시작하고, 이를 기준으로 설정했다.

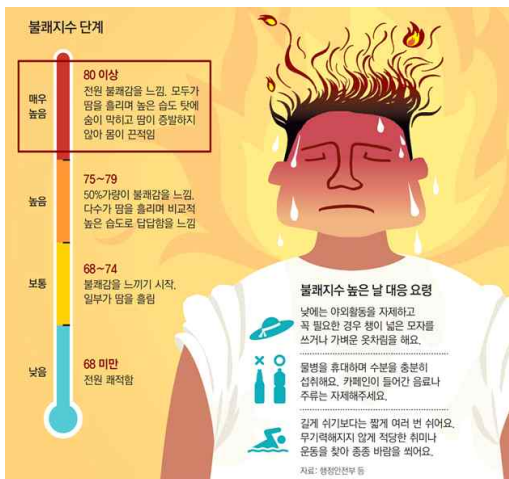


Fig 4 The discomfort index level

불쾌지수 단계		체감
매우 높음	80이상	전원 불쾌감을 느낌. 모두가 땀을 흘리며 숨이 막히고, 몸이 끈적임
높음	75-79	50%가량이 불쾌감을 느낌. 다수가 땀을 흘리고 비교적 답답함을 느낌.
보통	68-74	불쾌감을 느끼기 시작. 일부가 땀을 흘림.
낮음	68미만	전원 쾌적함

Table 9 Relationship between discomfort index and sentiment

먼지(분진)은 작은 입자는 폐에 깊이 침투하게 되며, 일반적으로 미세먼진보다 극미세분진이 많은 염증을 유발한다. 세종대학교에서 진행한 미세먼지 실험에 따르면 양쪽환기가 실내 미세먼지 농도를 낮추는데 제일 효과적임을 알 수 있다.

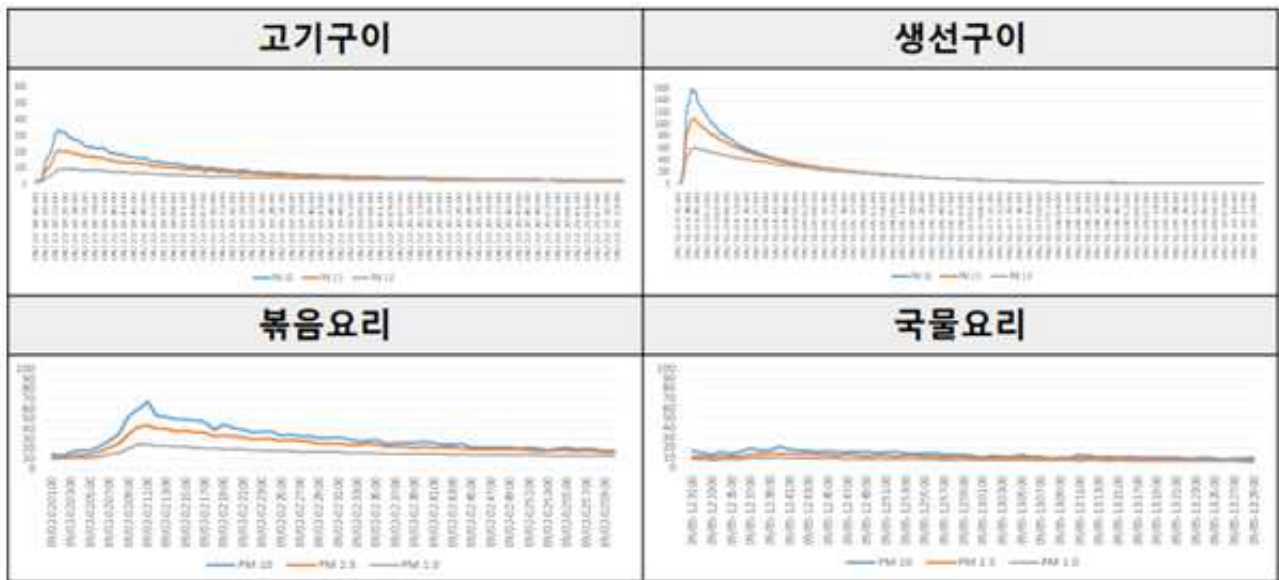


Fig 5 Concentration of indoor fine dust exposure when ventilation is not performed

아무런 동작을 취하지 않았을 때 미세먼지 농도는 급증하며, 농도가 낮아지는데 시간이 오래 소요된다. 구이요리를 할 때 제일 많은 미세먼지가 발생하고, 국물요리를 할 때 비교적 적은 미세먼지가 발생한다.

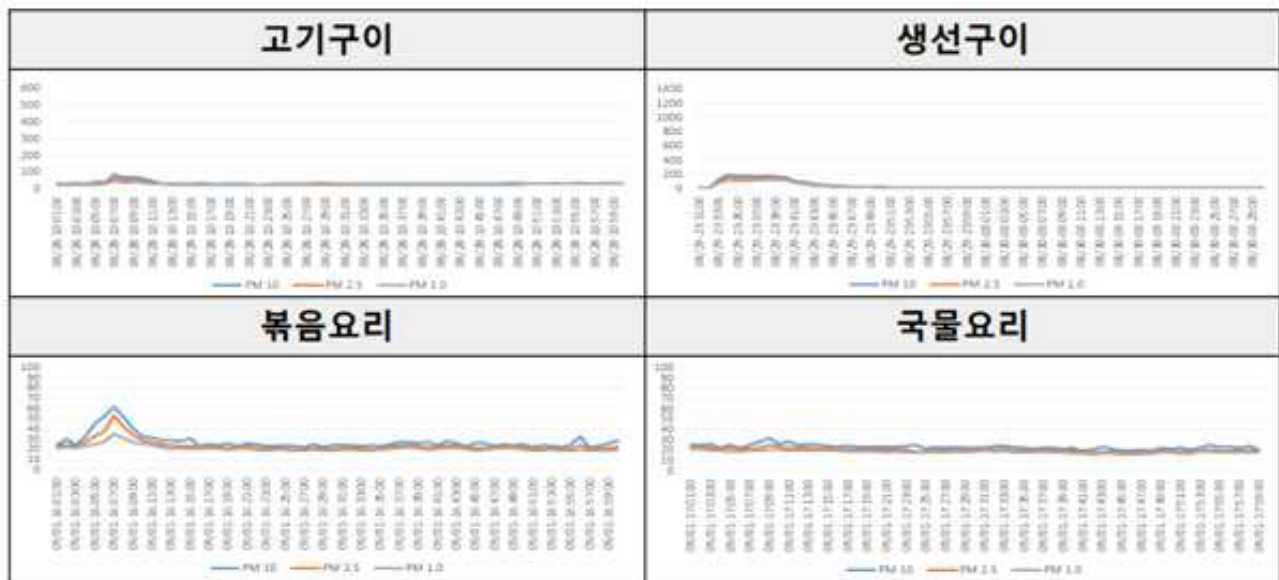


Fig 6 When both sides are ventilated, the level of exposure to fine dust in the room

양쪽 환기가 이뤄졌을 때 미세먼지 농도가 눈에 띄게 높아지지 않음을 알 수 있다.

세번째 지표값인 일산화탄소(CO)는 무색, 무취 유독성 가스로 탄소 성분이 불완전 연소시 발생한다. 일산화탄소는 산소운반 역할을 하는 헤모글로빈과 결합하여 산소운반능력을 저하시켜 일산화탄소에 노출되면 사망 위험이 있다. 때문에 세가지 지표값을 기반으로 팀 AK는 쾌적한 공기질을 위해 실내 공기질 상태를 주기적으로 센싱하고, 자동으로 on/off하는 환기시스템 제작을 목표로 한다.

2. 설계핵심내용

(1) 설계 문제 정의

스마트 환기 시스템은 분리형 센서 모듈을 사용해 사용자에게 편리성을 제공하고 애플리케이션 자동제어를 통해 실내 공기를 쾌적하게 만드는 자동화 시스템이다. 미세먼지와 일산화탄소 그리고 새집증후군에 영향을 주는 공기 중의 유기화합물과 같은 인체에 해로운 물질들을 실시간 센싱해 기준 수치를 넘어가면 사용자에게 부저와 LED를 통해 가시적으로 위험을 알리고 창문 개폐와 팬의 속도를 자동으로 조절해주는 스마트 장치이다.

(2) 설계 방법 및 제약조건

미세먼지, 유기화합물, 일산화탄소, 온도 및 습도 센서를 분리형 센서 모듈에 장착해 공기 중의 부유물질 및 온도와 습도를 센싱하고 해당 값을 디스플레이 모듈과 자체 제작 애플리케이션으로 가시성을 확보한다. 분리형 센서 모듈은 사용자가 원하는 센서를 선택해 탈부착할 수 있게 만들기 위해 제작했다. 창문은 외기 상태에 따른 자동 각도 조절과 창문 개폐를 통해 날씨가 좋지 않을 때의 공기 중의 부유물질이 실내로 들어오지 않게 설계했고 모터와 자체 제작한 기어를 이용해 창문 각도를 외부 공기 상태에 따라 적절한 각도로 변할 수 있게 만들었으며 또한 환풍기의 블레이드 개수와 각도 그리고 크기는 시연 공간의 크기에 맞게 제작했다. 공기청정기의 팬과 모듈 또한 자체 제작했고 필터는 해파 등급 필터를 사용했으며 에너지 효율을 높이기 위해 정전기 방식의 필터도 장착했다. 시연 공간은 아크릴판을 이용해 50x50x50으로 만들었다. 마지막으로 애플리케이션은 Mit App Inventor를 이용해 네이버 지역별 미세먼지 페이지 소스코드를 사용해 실시간으로 특정 지역의 미세먼지 농도를 검색할 수 있게 했고 센서의 센싱값인 미세먼지, 일산화탄소 온도 및 습도 등등을 기준치를 설정해 환풍기 바람세기를 수동으로 총 3단까지 그리고 오토모드가 가능하도록 제작했다. 센서의 기준치 설정에 있어서는 WHO 권고기준을 토대로 한다.

먼저 외기/내기 상태에 따라 창문의 각도를 조절하는 부분에 있어서 많은 제약조건이 걸린다. 포름 알데히드 같은 ‘새집증후군’을 유발하는 유기화합물을 만드는 것에 한계가 있고 공기 중의 모든 인체에 유해한 부유물질을 만드는 것이 쉽지 않기 때문에 대푯값을 미세먼지와 일산화탄소 두 가지만을 이용해 시연한다. 환풍기를 모터로 돌리는데 있어서도 문제가 발생했다. 시연 공간의 크기의 한계가 존재하기 때문에 큰 팬을 만드는데 어려움이 있으며 또한 현재 제작한 팬이 대부분의 모터 전압을 사용하는 상황이 발생했다. 이에 대한 해결책으로 모터 드라이브를 통해 극복하려 했지만 팬이 돌아가는 반응속도가 부족한 결과를 초래했고 시연 공간의 한계로 블레이드 개수 및 크기 그리고 각도가 미치는 영향이 미미함을 발견했다.

(3) 목표(QFD)

실내 공기질을 쾌적하게 유지해주는 스마트 환기 시스템 QFD 작성하면 다음과 같다.

1단계: 고객이 요구조건 조사

스마트 환기 시스템을 사용하는 소비자 입장에서 고객의 요구조건을 먼저 선정했다. 요구조건 항목은 다음과 같다. 환기 시스템의 기능이 많다, 가격이 저렴하다, 오랜기간 사용할 수 있다, 전기세가 적게 나온다, 사용 방법이 쉽다, 설치 또는 조립이 간편하다, 크기와 무게가 적당하다, 안전하게 사용할 수 있다, 환기가 잘 이뤄진다. 호흡기를 보호해준다, 오랜기간 부식이 발생하지 않는다, 물기에 고장나지 않는다, 굽힘에 강하다, 소음이 적다, 열에 강하다, 반응속도가 빠르다, 날개 청소가 가능하다, 공간을 많이 차지하지 않는다, 앱을 통

해 쉽게 관리할 수 있다. 날개 모양과 크기가 적당하다, 창문 형상이 알맞다.

2단계: 고객요구조건 가중치 산출

고객요구조건을 외부고객평가와 내부평가로 나눠 가중치를 산출했을 때 환기가 잘 이뤄지는 정도와 소음이 적게 나는 것이 제일 중요한 요구 조건으로 평가되었다.

고객 요구		외부고객	내부평가	Weight
경제성이 우수하다	기능이 많다	2	3	0.25
	가격이 저렴하다	4	4	0.4
	오랜기간 사용할 수 있다	5	3	0.4
	전기세가 적게 나온다	2	3	0.25
조작이 쉽다	사용방법이 쉽다	5	4	0.45
	설치(조립)가 간편하다	5	3	0.4
	크기, 무게가 적당하다	3	2	0.25
	안전하게 사용할 수 있다	5	5	0.5
환기가 잘된다	환기가 잘 이뤄진다	5	5	0.5
	호흡기를 보호해준다	4	4	0.4
품질이 좋다	오랜기간 부식이 발생하지 않는다	5	3	0.4
	물기에 고장나지 않는다	3	4	0.35
	균함에 강하다(생활기스가 발생하지 않는다)	3	2	0.25
	소음이 적다	5	5	0.5
	열에 강하다	3	4	0.35
	반응속도가 빠르다	4	2	0.3
사용하기 좋다	날개(Fan)청소가 가능하다	5	3	0.4
	공간을 많이 차지하지 않는다	5	4	0.45
	앱을 통해 쉽게 관리할 수 있다.	5	3	0.4
디자인이 예쁘다	날개 모양, 크기가 적당하다	1	3	0.2
	창문 형상이 알맞다	2	4	0.3

3단계: 공학사양

공학사양은 크게 조작용이성, 안정성, 품질, 기술성, 편리성이 있다. 조작이용성은 설치의 편의성, 조립의 용이성, 작동의 편리성으로, 안정성은 소음 발생정도, 고정성, 방수성, 청결성으로, 품질은 내구성, 무게, 강도로, 기술성은 제품의 크기, 날개의 형상, 정확성 및 신속성, 전기적 성능, 제품재료로, 편리성은 애플리케이션 서비스, 시스템 동작의 자동화로 세분화할 수 있다.

조작용이성	설치의 편의성
	조립의 용이성
	작동의 편리성
안정성	소음 발생정도(방음성)
	고정성
	방수성
	청결성
품질	내구성
	무게
	강도
기술성(배기량)	제품의 크기
	날개의 형상
	정확성 및 신속성
	전기적 성능
	제품 재료
편리성	애플리케이션 서비스
	시스템 동작의 자동화

4단계: 고객요구조건과 공학사양 관련정도과악

앞서 선정된 고객요구조건과 공학사양의 관련 정도에 따라 절대적으로 관련이 밀접하면 ‘◎’ 으로, 관련이 있으면 ‘○’ 으로, 관련이 있으나 미미하게 관련이 있으면 ‘△’ 으로, 아무런 관련이 없으면 ‘blank’ 로 표기하면 다음과 같은 관련성을 보인다.

절대적으로 관련 있음 ◎
 관련 있음 ○
 관련이 있으나 약함 △
 관련없음 blank

		조작성이성				안정성				품질			기술성 (배기량)				면 리 성	
		설치의 편의성	조립의 용이성	작동의 편리성	소음 발생정도(영음성)	고장성	방수성	청결성	내구성	무게	강도	제품의 크기	날개의 형상	정확성 및 신속성	전기적 성능	제품 재료	예물리케어선 서비스	시스템 동작의 자동화
경제성이 우수하다	기능이 많다			○													◎	◎
	가격이 저렴하다	○	○	○	◎	△	◎	△	◎	△	○	◎	△	◎	◎	◎	◎	◎
	오래기간 사용할 수 있다	○	○	◎	◎	○	◎	○	◎	△	△	△	△	○	◎	◎	◎	◎
	전기세가 적게 나온다				△									△	◎		△	△
조작이 쉽다	사용방법이 쉽다	◎	◎	◎										◎			◎	◎
	설치(조립)가 간편하다	◎	◎			◎				○		○	○			○	○	○
	크기, 무게가 적당하다	◎	◎	◎	○	◎			◎	◎	◎			△	○			
	안전하게 사용할 수 있다	○	○	◎	◎				◎	◎	◎	◎	△	△	○	○		
환기가 잘된다	환기가 잘 이뤄진다			◎	◎				◎		○	◎	◎	◎	◎	◎	◎	◎
	호흡기를 보호해준다							◎							○	○		
품질이 좋다	오래기간 부식이 발생하지 않는다			○				◎	◎			◎				◎	△	△
	물기에 고장나지 않는다						◎	◎	○				△		◎	◎		
	규힘에 강하다(생활기스가 발생하지 않는다)								◎		◎							
	소음이 적다								◎		◎		○			○		
	열에 강하다				○				◎		◎				◎	◎		
사용하기 좋다	반응속도가 빠르다			◎										◎	○		◎	◎
	날개(Fan)정소가 가능하다	◎	◎			○	○	◎				○	○			○		
	공간을 많이 차지하지 않는다	○	○	○						◎		◎						
	앱을 통해 쉽게 관리할 수 있다.			○										○	△		◎	◎
디자인이 예쁘다	날개 모양, 크기가 적당하다	◎	◎	△	○				○			◎	◎	△	△			
	창문 형상이 알맞다			○										○				

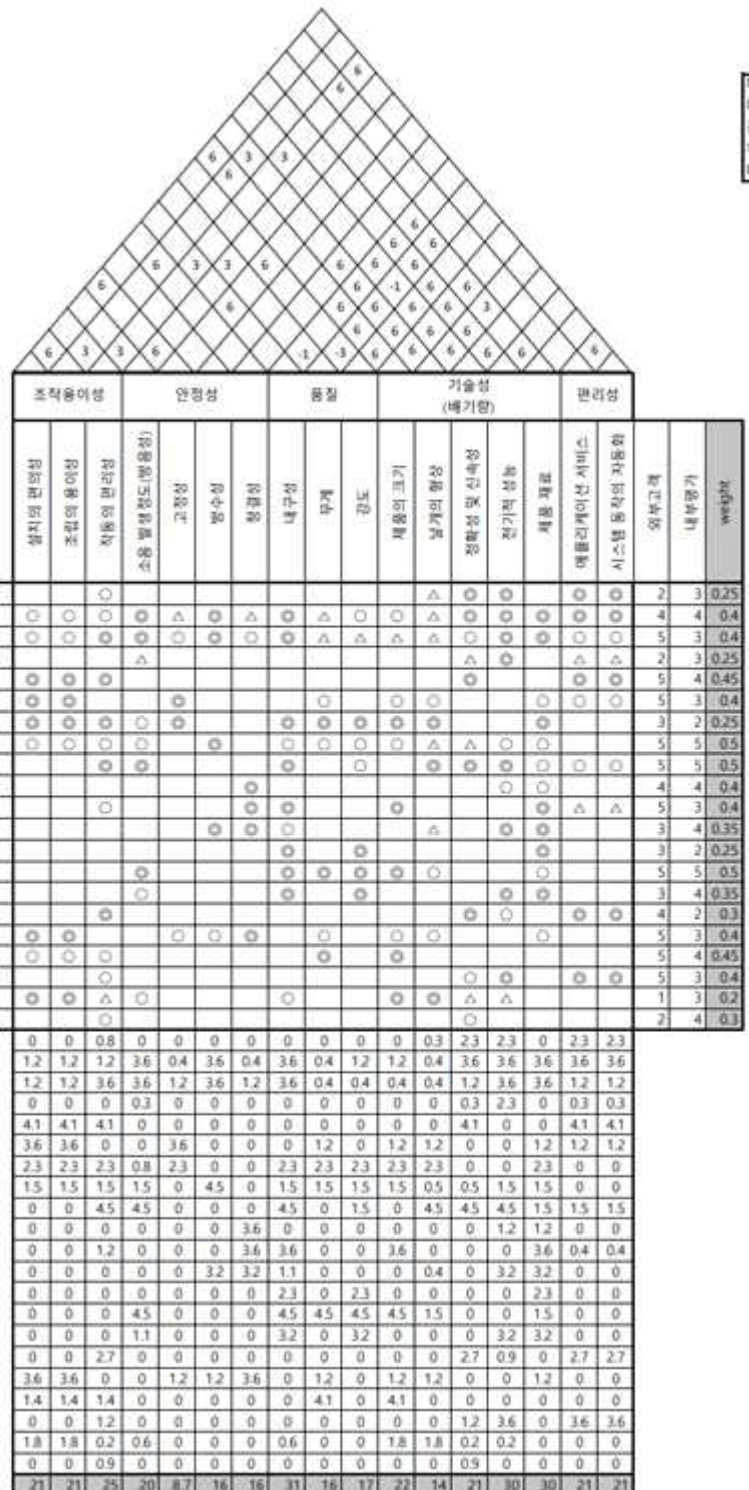
5단계: 설계특성결정

절대적으로 관련 있음 ◎ 9점
 관련 있음 ○ 3점
 관련이 있으나 약함 △ 1점
 관련없음 blank 0점

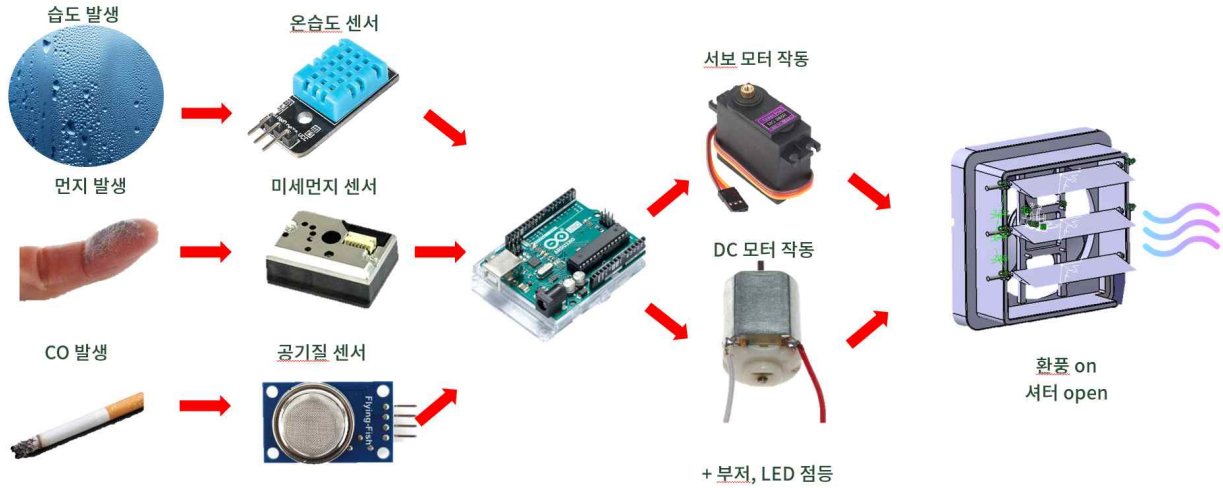
		조작용이성				안정성			품질			기술성 (배기량)				관리성			
		원자의 편리성	조작의 용이성	작업의 편리성	수동 발생 정도(발음 등)	고장성	방수성	정밀성	내구성	무게	강도	제품의 크기	배관의 형상	정확성 및 신속성	전기적 성능	제품 재료	애플리케이션 서비스	시스템 동작의 자동화	weight
경제성이 우수하다	기능이 많다			○									△	○	○		○	○	0.3
	가격이 저렴하다	○	○	○	○	△	○	△	○	△	○	△	△	○	○	○	○	○	0.4
	오랜기간 사용할 수 있다	○	○	○		○	○	○	○	△	△	△	△	○	○	○	○	○	0.4
조작이 쉽다	전기세가 적게 나온다				△										○		△	△	0.3
	사용방법이 쉽다	○	○	○										○			○	○	0.5
	설치(조립)가 간편하다	○	○			○						○	○			○		○	0.4
	크기, 무게가 적당하다	○	○	○	○	○			○	○	○	○	○			○			0.3
환기가 잘된다	안전하게 사용할 수 있다	○	○	○	○				○	○	○	○	△	△	○	○			0.5
	환기가 잘 이뤄진다			○	○									○	○	○	○	○	0.5
품질이 좋다	호흡기를 보호해준다							○							○	○			0.4
	오랜기간 부식이 발생하지 않는다			○					○			○				○	△	△	0.4
	물기에 고장나지 않는다						○		○				△		○				0.4
	균형에 강하다(생활기스가 발생하지 않는다)								○	○	○					○			0.3
	소음이 적다				○				○	○	○	○				○			0.5
	열에 강하다				○				○		○				○	○			0.4
사용하기 좋다	반응속도가 빠르다			○										○	○		○	○	0.3
	날개(fan)정소가 가능하다	○	○			○	○	○				○	○			○			0.4
	공간을 많이 차지하지 않는다	○	○	○															0.5
디자인이 예쁘다	앱을 통해 쉽게 관리할 수 있다			○										○	○		○	○	0.4
	날개 모양, 크기가 적당하다	○	○	△	○				○			○	○	△	△				0.2
장문 형상이 알맞다				○										○					0.3
		0	0	0.8	0	0	0	0	0	0	0	0	0.3	2.3	2.3	0	2.3	2.3	
		1.2	1.2	1.2	3.6	0.4	3.6	0.4	3.6	0.4	1.2	1.2	0.4	3.6	3.6	3.6	3.6	3.6	
		1.2	1.2	3.6	3.6	1.2	3.6	1.2	3.6	0.4	0.4	0.4	0.4	1.2	3.6	3.6	1.2	1.2	
		0	0	0	0.3	0	0	0	0	0	0	0	0	0.3	2.3	0	0.3	0.3	
		4.1	4.1	4.1	0	0	0	0	0	0	0	0	0	4.1	0	0	4.1	4.1	
		3.6	3.6	0	0	3.6	0	0	0	1.2	0	1.2	1.2	0	0	1.2	1.2	1.2	
		2.3	2.3	2.3	0.8	2.3	0	0	2.3	2.3	2.3	2.3	2.3	0	0	2.3	0	0	
		1.5	1.5	1.5	1.5	0	4.5	0	1.5	1.5	1.5	1.5	0.5	0.5	1.5	1.5	0	0	
		0	0	4.5	4.5	0	0	0	4.5	0	1.5	0	4.5	4.5	4.5	1.5	1.5	1.5	
		0	0	0	0	0	0	3.6	0	0	0	0	0	0	1.2	1.2	0	0	
		0	0	1.2	0	0	0	3.6	3.6	0	0	3.6	0	0	0	3.6	0.4	0.4	
		0	0	0	0	0	3.2	3.2	1.1	0	0	0	0.4	0	3.2	3.2	0	0	
		0	0	0	0	0	0	0	2.3	0	2.3	0	0	0	0	2.3	0	0	
		0	0	0	4.5	0	0	0	4.5	4.5	4.5	4.5	1.5	0	0	1.5	0	0	
		0	0	0	1.1	0	0	0	3.2	0	3.2	0	0	0	3.2	3.2	0	0	
		0	0	2.7	0	0	0	0	0	0	0	0	0	2.7	0.9	0	2.7	2.7	
		3.6	3.6	0	0	1.2	1.2	3.6	0	1.2	0	1.2	1.2	0	0	1.2	0	0	
		1.4	1.4	1.4	0	0	0	0	0	4.1	0	4.1	0	0	0	0	0	0	
		0	0	1.2	0	0	0	0	0	0	0	0	0	1.2	3.6	0	3.6	3.6	
		1.8	1.8	0.2	0.6	0	0	0	0.6	0	0	1.8	1.8	0.2	0.2	0	0	0	
		0	0	0.9	0	0	0	0	0	0	0	0	0	0.9	0	0	0	0	
		21	21	25	20	8.7	16	16	31	16	17	22	14	21	30	30	21	21	

고객 요구 weight 값과 각 대응 값을 곱하여 설계특성 weight를 계산하여 나타낸 값이다. 내구성, 전기적 성능, 제품 재료가 가장 중요한 설계 조건으로 산출되었다.

매우관련있음	6
다소관련있음	3
관련없음	0
부정적관련	-1
매우부정적관련	-3



(4) 시스템 구성



4.1) 팬의 수

스마트 환기 시스템을 구현하는 것에 있어서 가장 효율적인 배출을 위해 팬의 갯수를 결정하였다. 이를 확인하기 위해 날개수에 따른 공기의 유동해석을 참조하였으며 이때 3개, 4개, 5개 중에서 결정하였다. 보다 정확한 계산을 위하여 다음 방정식들을 고려하였다. 비압축성, 압축성에 통용되는 연속방정식을 사용했으며 S_m 은 다른 상에서 유입되는 부속물의 질량이며 무시해도 무방하다.

$$\frac{\partial \rho}{\partial t} + \frac{\partial}{\partial x_i}(\rho u_i) = S_m$$

또한 운동량 보존 방정식과 에너지 방정식을 함께 이용하였다.

$$\frac{\partial}{\partial x_i}(\rho u_i u_j) = -\frac{\partial p}{\partial x_i} + \frac{\partial \tau_{ij}}{\partial x_j} + \rho g_i$$

$$\frac{\partial}{\partial t}(\rho E) + \frac{\partial}{\partial x_i}(u_i(\rho E + p)) = \frac{\partial}{\partial x_i}\left(k_{eff} \frac{\partial T}{\partial x_i} - \sum_j h_j J_j + u_j(\tau_{ij})_{eff}\right)$$

동시에 난류 운동에너지와 난류 운동에너지의 소멸율을 나타낸 전달방정식을 같이 사용하며 보다 정확한 설계를 추구하고자 하였다.

$$\rho \frac{Dk}{Dt} = \frac{\partial}{\partial x_i} \left[\left(\mu + \frac{\mu_t}{\sigma_k} \right) \frac{\partial k}{\partial x_i} \right] + G_k + G_b - \rho \epsilon - Y_M$$

$$\rho \frac{D\epsilon}{Dt} = \frac{\partial}{\partial x_i} \left[\left(\mu + \frac{\mu_t}{\sigma_\epsilon} \right) \frac{\partial \epsilon}{\partial x_i} \right] + C_{1\epsilon} \frac{\epsilon}{k} (G_k + C_{3\epsilon} G_b) - C_{2\epsilon} \rho \frac{\epsilon^2}{k}$$

보다 정확한 공기의 유동을 계산하고자 한 것은 효과적인 공기의 배출량과 적합한 전력소비량을 고려하기 위함이다. 같은 속도를 내기 위해서 날개 수가 많을수록 큰 전력이 필요함을 확인하여 전력소비에 있어서 3엽의 날개가 가장 유리함을 확인했다. 동시에 공기의 유동해석에서 날개사이가 좁을수록 유동 속도와 바람의 세기가 약해짐을 확인하였다. 따라서 최대의 속력을 내며 효율적인 전력소비량을 유지할 수 있는 3엽의 날개를 선택하였다.

4.2) 팬의 구동방식

팬은 에너지를 통해 기계에 압력과 속도에너지로 변환시켜주는 유체기계로 주요한 환기장치이다. 이에 피치각의 크기에 따른 유동분석을 참고하였다. 팬의 rpm은 모두 동일하였고 정압별 풍량을 측정한 값을 참고하였다. 피치각은 20도~45도, 5도 간격으로 조정하여 팬의 효율을 구분하였다. 그 결과 피치각이 커질수록 풍량은 증가하였으나 35도에서 역전되었으며 축동력도 피치각에 따라 선형적으로 증가함을 확인했다. 위의 결과를 정리하여 팬 효율을 공기동력에 소비전력을 나눈 값을 기준으로 잡았으며 가장 이상적인 각도는 30도임을 확인하여 설계하였다. 정의된 유량계수와 정압계수를 통해 수치해석의 검증을 실행하는 과정으로 설계를 확인하였다.

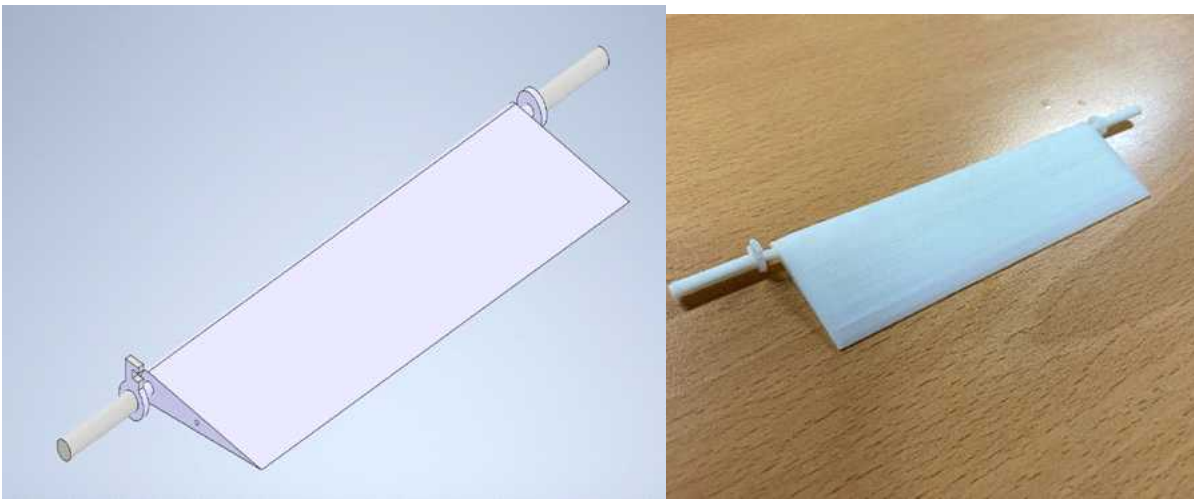
유량계수(flow coefficient)

$$\Phi = \frac{4Q}{\pi^2 (D_t^2 - D_h^2) D_t N}$$

정압계수(pressure coefficient)

$$\Psi_s = \frac{2gP_s}{\rho(\pi D_t N / 60)^2}$$

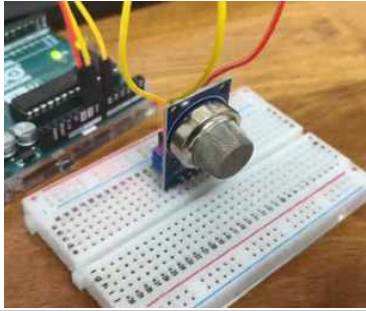
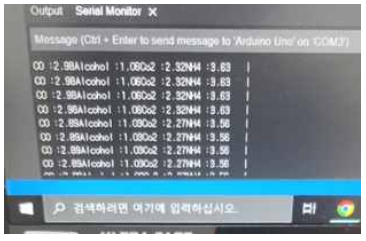
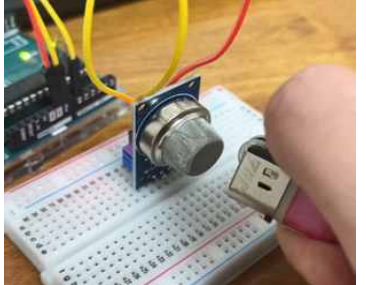
4.3) 환풍기 개폐 시스템



첫 설계시 기어를 이용하여 개폐를 하고자 하였으나 설계과정에서 케이스와의 길이를 조정하는데 있어서 문제가 발생하여 설계를 수정하였다. 결과적으로 shutter에 갈고리를 추가설계하여 프린팅하였다. 이후 아두이노와 결합한 서보모터를 사용하여 개폐시스템을 구현하였다. 정확한 작동을 위해 실을 갈고리에 고정시키고 길이를 조정하여 서보모터 작동 시 환기가 효과적으로 되도록 구현하였다.

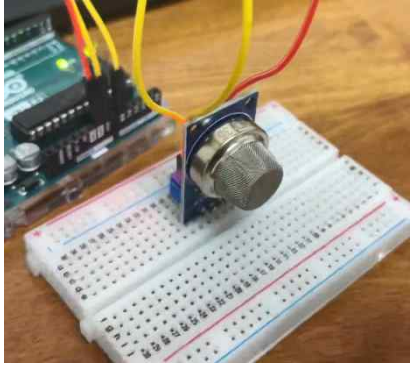
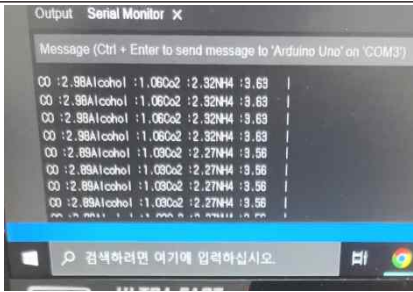
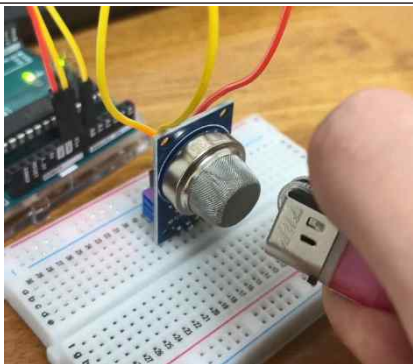
4.4) 미세먼지 센서

측정단위는 $\mu g/m^3$ 이고, 작동과정은 다음과 같다.

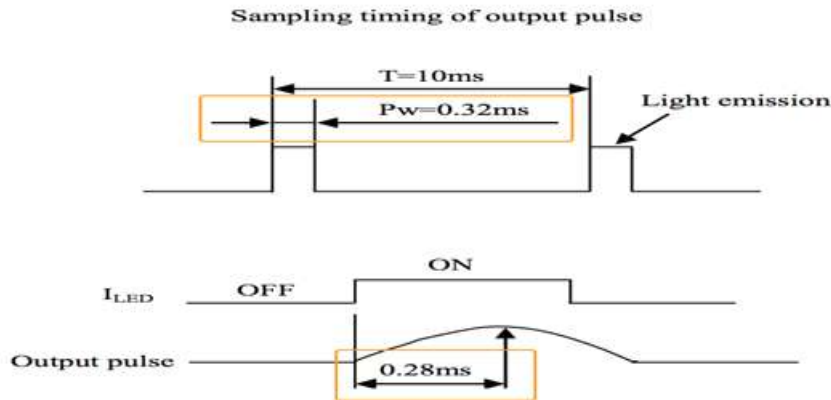
	평상시 공기질 센서 작동
	평상시 공기질 센서 수치 CO: 2.89 Alcohol: 1.03 CO₂: 2.27 NH₄: 3.56
	연소시 공기질 센서 작동
	연소시 공기질 센서 수치 CO: 786.18 Alcohol: 95.44 CO₂: 133.61 NH₄: 120.46

4.5) 공기질 센서

측정단위는 ppm이고, 작동과정은 다음과 같다.

	평상시 공기질 센서 작동
 <pre> Output Serial Monitor X Message (Ctrl + Enter to send message to 'Arduino Uno' on 'COM3') CO :2.98Alcohol :1.060c2 :2.32NH4 :3.63 CO :2.98Alcohol :1.060c2 :2.32NH4 :3.63 CO :2.98Alcohol :1.060c2 :2.32NH4 :3.63 CO :2.98Alcohol :1.060c2 :2.32NH4 :3.63 CO :2.98Alcohol :1.090c2 :2.27NH4 :3.56 CO :2.98Alcohol :1.090c2 :2.27NH4 :3.56 CO :2.98Alcohol :1.090c2 :2.27NH4 :3.56 CO :2.98Alcohol :1.090c2 :2.27NH4 :3.56 CO :2.98Alcohol :1.090c2 :2.27NH4 :3.56 </pre>	평상시 공기질 센서 수치 CO: 2.89 Alcohol: 1.03 : 2.27 : 3.56
	연소시 공기질 센서 작동
 <pre> Message (Ctrl + Enter to send message to 'Arduino Uno' on 'COM3') CO :2.80Alcohol :1.010c2 :2.22NH4 :3.49 CO :2.80Alcohol :1.010c2 :2.22NH4 :3.49 CO :2.80Alcohol :1.010c2 :2.22NH4 :3.49 CO :26.46Alcohol :5.170c2 :11.35NH4 :14.31 CO :629.15Alcohol :79.720c2 :115.63NH4 :104.72 CO :795.18Alcohol :95.440c2 :133.61NH4 :120.46 CO :854.78Alcohol :92.330c2 :115.69NH4 :107.99 CO :518.75Alcohol :89.210c2 :99.76NH4 :92.77 </pre>	연소시 공기질 센서 수치 CO: 786.18 Alcohol: 95.44 : 133.61 : 120.46

4.6) 미세먼지 센서 원리



Parameter	Symbol	Value	Unit
Pulse Cycle	T	10 ± 1	ms
Pulse Width	Pw	0.32 ± 0.02	ms
Operating Supply voltage	V _{CC}	5 ± 0.5	V

```

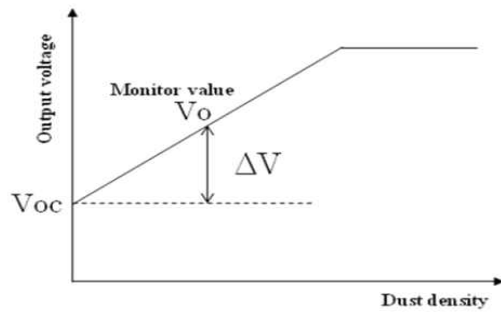
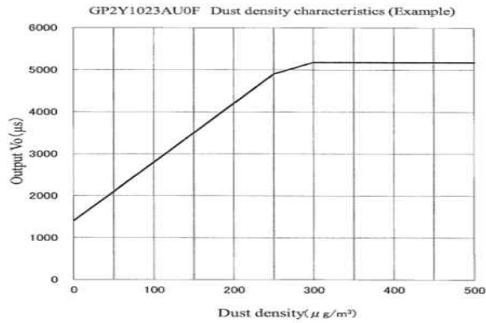
...
digitalWrite(no3Pin, LOW); // IRED 'on'
delayMicroseconds(280); // sampling timing

voltsMeasured = analogRead(no5Pin); // read the dust value

delayMicroseconds(40); // pulse width 0.32 - 0.28 = 0.04 msec
digitalWrite(no3Pin, HIGH); // IRED 'off'
...

```

IRED를 제어하고 미세먼지 신호의 정보를 읽어오는 과정에서의 타이밍이 매우 중요하다. 데이터시트에 언급된 내용을 살펴보면 위 사진과 같은데 IRED를 점등하는 시간은 약 0.32 msec의 펄스폭을 유지해야 하며 펄스가 HIGH가 되는 순간부터 약 0.28 msec 후에 아두이노 아날로그 입력으로 값을 읽어와야 한다.



GP2Y1023AU0F 센서의 PWM 펄스 값

이 센서의 출력전압(V_o)는 먼지가 없을 때의 출력전압(V_{oc})과 먼지농도(ΔV)에 비례하여 출력

■ Electro-optical Characteristics

(T _a =25°C, V _{CC} =5V)						
Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Sensitivity	K	*1 *2 *3	0.35	0.5	0.65	V/(0.1mg/m³)
Output voltage at no dust	V _{OC}	*2 *3	0	0.9	1.5	V
Output voltage range	V _{OH}	*2 *3 R _L =4.7kΩ	3.4	—	—	V
LED terminal current	I _{LED}	*2 LED terminal voltage = 0	—	10	20	mA
Consumption current	I _{CC}	*2 R _L =∞	—	11	20	mA

*1 Sensitivity is specified by the amount of output voltage change when dust density changes by 0.1 mg/m³.
And the dust density for detection is a value of the density of cigarette (MILD SEVEN®) smoke measured by the digital dust monitor (P-5L2; manufactured by SHIBATA SCIENTIFIC TECHNOLOGY LTD.).

*2 Input condition is shown in Fig. 1

*3 Output sampling timing is shown in Fig. 2

$$\text{typically } K = 0.5V/[100\mu g/m^3] = 0.005V/[\mu g/m^3]$$

$$\Delta V = V_o - V_{oc}$$

$$\text{미세먼지 농도}[\mu g/m^3] = (V_o - V_{oc}) / 0.005$$

GP2Y1023AU0F 센서의 데이터시트나 여러 자료를 찾아보면 미세먼지 센서의 값을 먼지 농도로 변환하기 위한 공식이 없다. 따라서 미세먼지 농도를 측정하기 위해 미세먼지 계측기를 기준으로 해당 미세먼지 센서에 대한 조정이 필요하지만 현재 학부 수준에서 계측기를 준비해 측정하는 것에는 한계가 있기 때문에 데이터시트와 애플리케이션 노트의 내용으로 미세먼지의 농도 값을 계산하고 보정하는 방법을 사용한다. 데이터시트에 기록되어 있는 K 매개변수를 이용하면 100 마이크로미터 퍼 세제곱미터로 단위 변환할 수 있으며 또한 최소, 최대값에 의해 오차는 발생할 수 있다.

4.7) 미세먼지 센싱 코드 및 적절한 제어 값

실내공기질 관리법 (환경부, 2018-10-18)

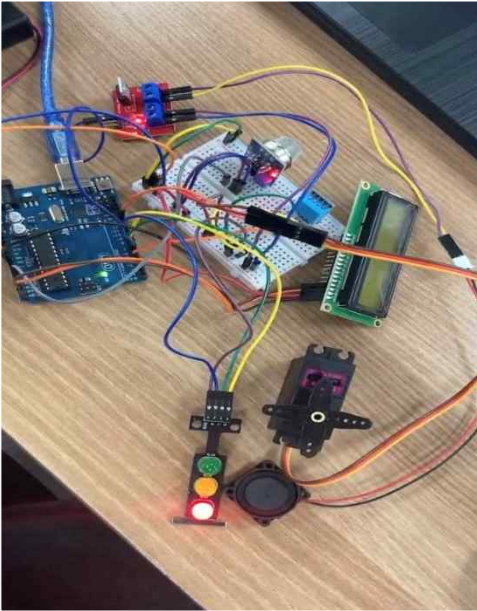
Standards for Indoor Pollutants in Public Facilities in Republic of Korea (Oct. 18, 2018)

오염물질항목	입자상물질 (Aerosol)		실내공기질 유지기준 Gas상 물질				방사성물질			
	미세먼지 (PM-10) ($\mu\text{g}/\text{m}^3$)	초미세먼지 (PM-2.5) ($\mu\text{g}/\text{m}^3$)	이산화탄소 (ppm)	폼알데하이드 ($\mu\text{g}/\text{m}^3$)	총부유세균 (CFU/ m^3)	일산화탄소 (ppm)	이산화질소 (ppm)	라돈 (Bq/ m^3)	총휘발성 유기화합물 ($\mu\text{g}/\text{m}^3$)	총부유곰팡이 (CFU/ m^3)
다중이용시설	유지기준 (1년 1회 측정)						권고기준 (2년 1회 측정)			
지하역사, 지하도상가, 철도역사의 대합실, 여객자동차터미널의 대합실, 항만시설 중 대합실, 공항시설 중 여객터미널, 도서관·박물관 및 미술관, 대규모 점포, 장례식장, 영화상영관, 학원, 전시시설, 인터넷컴퓨터게임시설제공업의 영업시설, 목욕장업의 영업시설	100	50	1000	100	-	10	0.1	148	500	-
의료기관, 산후조리원, 노인요양시설, 어린이집	75	35		80	800		0.05		400	500
실내주차장	200	-		100	-	25	0.3		1000	-
실내 체육시설, 실내 공연장, 업무시설, 둘 이상의 용도에 사용되는 건축물	200	-	-	-	-	-	-	-	-	-

<비고>
 1. 도서관, 영화상영관, 학원, 인터넷컴퓨터게임시설제공업 영업시설 중 자연환기가 불가능하여 자연환기설비 또는 기계환기설비를 이용하는 경우에는 이산화탄소의 기준을 1,500ppm 이하로 한다.
 2. 실내 체육시설, 실내 공연장, 업무시설 또는 둘 이상의 용도에 사용되는 건축물로서 실내 미세먼지(PM-10)의 농도가 $200\mu\text{g}/\text{m}^3$ 에 근접하여 기준을 초과할 우려가 있는 경우에는 실내공기질의 유지를 위하여 다음 각 목의 실내공기정화시설(덕트) 및 설비를 교체 또는 청소하여야 한다.
 가. 공기정화기와 이에 연결된 급·배기관(급·배기구를 포함한다) 나. 중앙집중식 냉·난방시설의 급·배기구
 다. 실내공기의 단순배기 라. 화장실용 배기관
 마. 조리용 배기관

위에 보이는 환경부에서 밝힌 자료에 따른 실내 미세먼지 농도는 50ppm이 넘지 않도록 한다. 때문에, 센서 기준값으로 50ppm을 설정하였다.

4.8) 공기질센서값에 따른 팬 구동



4.9) 공기질 센서의 원리

E. Sensitivity characteristic curve

Fig.2 sensitivity characteristics of the MQ-135

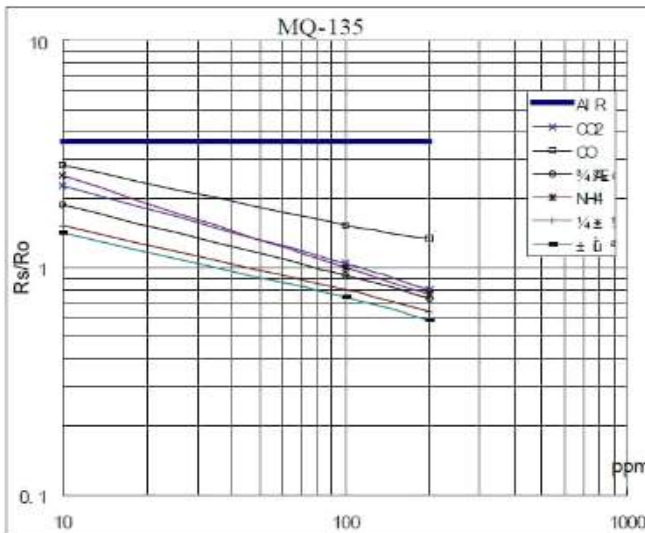


Fig.3 is shows the typical sensitivity characteristics of the MQ-135 for several gases. in their. Temp: 20[] Humidity: 65%[] O₂ concentration 21% RL=20k[]
Ro: sensor resistance at 100ppm of NH₃ in the clean air.
Rs: sensor resistance at various concentrations of gases.

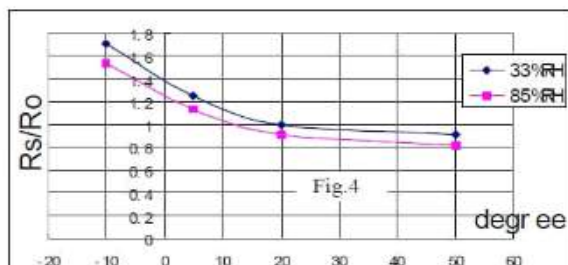


Fig.4 is shows the typical dependence of the MQ-135 on temperature and humidity. Ro: sensor resistance at 100ppm of NH₃ in air at 33%RH and 20 degree.
Rs: sensor resistance at 100ppm of NH₃ at different temperatures and humidities.

MQ-135 센서를 사용하여 PPM을 측정하려면 MQ-135 데이터시트에서 가져온 $(R_s / R_o)_V / s$ PPM 그래프를 살펴 봐야 한다. R_o 값은 신선한 공기의 저항 값이고 R_s 값은 가스 농도의 저항 값이다. R_o 값을 찾아 센서를 보정 한 다음 해당 값을 사용해 아래 공식을 적용한다.

RS (Resistance of sensor) : $RS = (V_c / V_{RL} - 1) \times R_L$

4.10) 공기질 센서에서 센싱할 화학 물질

공기질 센서에서 측정할 수 있는 물질은 이산화탄소(CO_2), 일산화탄소(CO), 벤젠(C_6H_6), 암모니아(NH_4), 알코올(C_2H_5OH), 포름알데하이드(H_2CO)가 있다. 이 중에서는 실내에서 위험한 물질인 일산화탄소 벤젠 포름알데하이드를 따로 측정할 수 있는 코드를 만들었으며, 벤젠과 포름알데하이드는 인체에 굉장히 위험하기도 하고 구하기도 어려운 화학물질이기 때문에 이번 모듈로 사용하기에는 힘들다.

4.11) 일산화 탄소의 위험성

따라서 공기질 센서 값 모듈은 ‘일산화탄소 증가에 따른 부저 울림과 환풍기 개방’ 이다. 먼저 일산화탄소 중독의 위험성에 대해 알아보자. 일산화탄소 중독은 일산화탄소에 중독되어 두통, 호흡 곤란이 생기고 결국 사망까지도 이를 수 있는 질환이다. 가정에서는 특히 연료를 연소해도 생기기 때문에 가정에서도 각별히 조심할 필요가 있다. 자료에 따르면 일산화탄소의 정상 농도는 20ppm 이하일 경우이며, 그 이상의 일산화탄소가 지속 될 경우 굉장히 위험하다.

농도		중독현상
9PPM	0.0009%	짧은 시간 노출됨에 대한 최대 허용치수
35PPM	0.0035%	연방법에 따른 8시간의 계속적인 노출에 대한 최대 허용치수
200PPM	0.02%	2~3시간 이내에 가벼운 두통, 피곤, 어지러움
400PPM	0.04%	1시간 이내에 두통, 어지러움, 구토 3시간 내에 사망
800PPM	0.08%	20분 이내에 두통, 어지러움, 구토 40분 이내에 의식불명, 1시간 이내에 사망
1600PPM	0.16%	10분 이내에 두통, 어지러움, 구토 30분 내에 사망
3200PPM	0.32%	5분 이내에 두통, 어지러움, 구토 20분 내에 사망
6400PPM	0.64%	2분 이내에 두통, 어지러움, 구토 10분 내에 사망
12800PPM	1.28%	3분 내에 사망

4.12) 일산화탄소 센싱 코드 및 적절한 제어 값 확립

MQ-135센서로 일산화탄소만 센싱을 할 수 있게 하였고, 일산화탄소 센싱값 결과 아무것도 하지않는 평상 시에는 2~3ppm 정도인 것을 확인 할 수 있다. 반면 일산화 탄소 발생을 위해 라이터의 가스를 가져다 대본 결과 40에서 60까지 뛰는 모습을 볼 수 있다. 평상시에는 2~3ppm이던 일산화 탄소 농도가 20ppm만 넘어도 굉장히 위험한 상황임을 실험 결과와 자료를 참고하여 알 수 있었다. 그래서 부저가 울리는 제어값을 20ppm으로 하였다.

```

#include <MQUnifiedsensor.h> // Include the library

#define placa "Arduino UNO"

#define Voltage_Resolution 5

#define pin_A0 // Analog input 0 of your arduino

#define type "MQ-135" // MQ135

#define ADC_Bit_Resolution 10 // For arduino UNO/MEGA/NANO

#define RatioMQ135CleanAir 3.6 // RS / R0 = 3.6 ppm

#define LED_R 3
#define LED_Y 4
#define LED_G 5

MQUnifiedsensor MQ135(placa, Voltage_Resolution, ADC_Bit_Resolution, pin, type);

void setup() {

    Serial.begin(9600);

    MQ135.setRegressionMethod(1); // _PPM = a*ratio^b (PPM 농도와 상수 값을 계산하기 위한 수학 모델 설정)
    MQ135.init();

    Serial.print("Calibrating please wait.");

    float calcR0 = 0;

    for(int i = 1; i<=10; i++) {

        MQ135.update(); // 데이터를 업데이트하면 arduino가 아날로그 핀의 전압을 읽습니다

        calcR0 += MQ135.calibrate(RatioMQ135CleanAir);

        Serial.print(".");

    }

    MQ135.setR0(calcR0/10);

    Serial.println(" done!");

    if(!sinf(calcR0)) { Serial.println("Warning: Conection issue founded, R0 is infite (Open circuit detected) please check your wiring and supply"); while(1); }

    if(calcR0 == 0){ Serial.println("Warning: Conection issue founded, R0 is zero (Analog pin with short circuit to ground) please check your wiring and supply"); while(1); }

    Serial.println("CO : ");

    pinMode(LED_R, OUTPUT);
    pinMode(LED_Y, OUTPUT);
    pinMode(LED_G, OUTPUT);

}

void loop() {

    MQ135.update(); // 데이터를 업데이트하면 arduino가 아날로그 핀의 전압을 읽습니다

```

```

MQ135.setA(605.18); MQ135.setB(-3.937); // CO 농도를 얻기 위해 equation 값 구성

float CO = MQ135.readSensor();          // 센서는 설정 전 또는 설정에서 설정된 모델과 a 및 b 값을 사용하여 PPM 농도를 읽습니다

float Acetona = MQ135.readSensor();

Serial.print("CO :");  Serial.print(CO);

delay(500);

if (CO > 20) {
  turnOffAll();

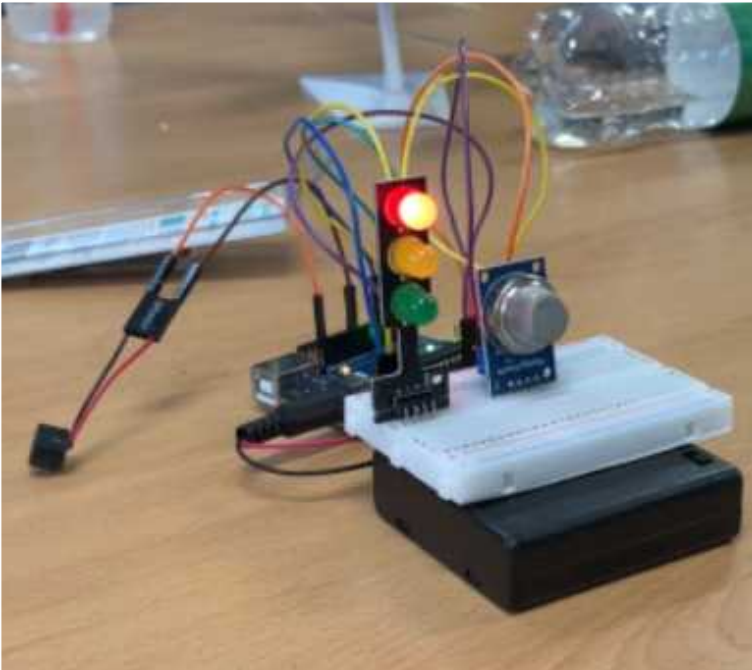
  digitalWrite(LED_R, HIGH);
  delay(1000);
  turnOffAll();
}

}

void turnOffAll() {
  digitalWrite(LED_R, LOW);
  digitalWrite(LED_Y, LOW);
  digitalWrite(LED_G, LOW);
}

```

4.13) 일산화탄소 센서 모듈



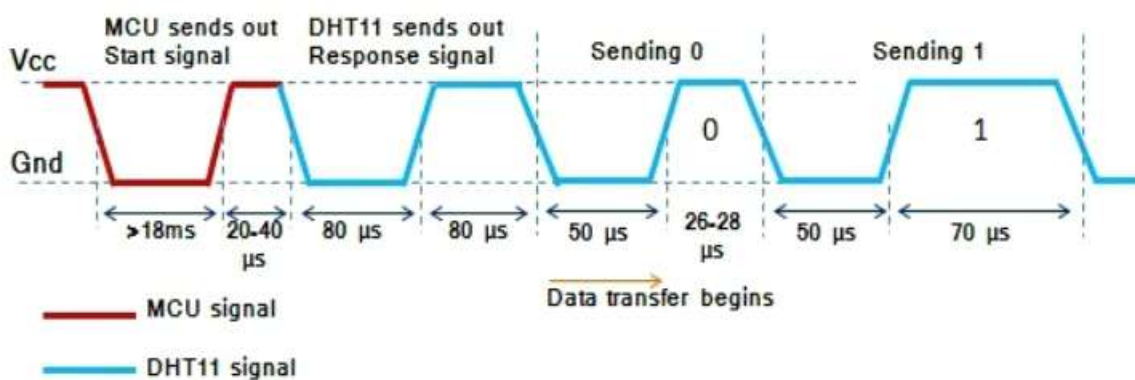
일산화탄소를 센싱하여 일산화탄소 농도가 20ppm이 넘을 경우 부저가 울리게 되고, 20ppm아래로 떨어지면 부저가 다시 울리지 않게 설정하였다.

4.14) 온습도센서값에 따른 팬 구동



4.15) 온습도 센서 원리

DHT11은 두 전극 사이의 전기 저항을 측정해 수증기를 감지한다. 습도 감지 구성 요소는 전극이 표면에 적용된 수분 보유 기관이다. 수증기가 기관에 흡수되면 이온이 기관에 의해 방출 돼 전극 사이의 전도성이 증가한다. 또한 두 전극 사이의 저항 변화는 상대 습도에 비례한다.



DHT11센서는 Signal 핀의 레벨을 통해 통신을 위 사진과 같이 수행한다. 초기 18ms 동안 LOW로 유지하다 High로 전환하여 DHT에 요청을 하게 되고 20us~40us 이후 DHT에서 50us간 LOW로 80us간 HIGH로 전환하여 진행에 대한 반환을 하게 된다. 이후 온습도 데이터를 전송해 비트전송시 50us LOW후 26~28us HIGH는 0bit로 50us LOW후 70us HIGH는 1bit로 간주하여 데이터를 전송하고 데이터의 끝에서는 50us LOW후 계속 HIGH상태를 유지한다.

4.16) 온습도 센싱 코드 및 적절한 제어 값

Table 2. Health Effect of the Discomfort Index.

Discomfort Index ($^{\circ}\text{C}$)	Possible phenomenon
> 80	Most of the total population suffers discomfort
75~80	The 50% of the total population feels discomfort
68~75	Few of the total population feels discomfort
< 68	No discomfort

($H > 80\%$) 거의 대부분의 사람들은 습도가 80을 초과할 때 불편함을 느낀다.

($75\% < H \leq 80\%$) 대부분의 사람들 중 50%는 습도가 75초과 80이하일 때 불편함을 느낀다.

($68\% < H \leq 75\%$) 몇몇의 사람들은 습도가 68초과 75이하일 때 불편함을 느낀다.

($H < 68\%$) 거의 대부분의 사람들은 습도가 68미만일 때 불편함을 느끼지 않는다.

```
//습도에 따른 DC 모터 작동
#include <DHT11.h>           // DHT11 라이브러리 추가
DHT11 dht11(5);             // DHT11 데이터핀을 5 번 핀에 연결
int err;
float temp, humi;            //임시 데이터를 받아 놓기 위한 변수 지정
void setup()
{
    pinMode(3, OUTPUT);      // 모터 연결한 핀
}

void loop()
{
    if((err=dht11.read(humi, temp))!=0) // 온습도 데이터 받아오기
    {
        if(humi > 40)          // 습도 40 이상일 때
        {
            digitalWrite(3, HIGH); // 모터 핀 신호 o (모터 구동)
        }
        else if(humi < 40)      // 습도 40 이하일 때
        {
            digitalWrite(3, LOW);  // 모터 핀 신호 x
        }
    }
}
```

4.17) 최종 아두이노 코드

최종 아두이노 코드에는 각 센서값의 우선순위에 따라 환풍기와 실내 간이공기청정기가 작동이 된다. 각 우선 순위는 다음과 같다.

1. 일산화탄소
2. 실외 미세먼지
3. 실내 미세먼지
4. 불쾌지수

```
#include <Wire.h> //LCD, 온습도센서 라이브러리
#include <LiquidCrystal_I2C.h> //I2C LCD 라이브러리
#include <DHT.h> // 라이브러리
#include <Emotion.h> //특수 문자 라이브러리
#include <SoftwareSerial.h>
#define DHTPIN 4 //온습도센서 핀 번호
#define DHTTYPE DHT11 //온습도센서 센서 타입
#include<Servo.h> //Servo 라이브러리
Servo servo; //Servo 클래스로 servo객체 생성
#include <MQUnifiedsensor.h> // Include the library
#define placa "Arduino UNO"
#define Voltage_Resolution 5
#define pin A1 // Analog input 0 of your arduino
#define type "MQ-135" // MQ135
#define ADC_Bit_Resolution 10 // For arduino UNO/MEGA/NANO
#define RatioMQ135CleanAir 3.6 // RS / R0 = 3.6 ppm

//모드가 총 4가지가있다
//off - 1단 - 2단 - 3단 - Auto
int mode = 0;
int Dir1Pin_A = 13; //DC모터 방향제어
int Dir2Pin_A = 11; //DC모터 방향제어
int SpeedPin_A = 3; //DC모터 속도제어
int Buzzer_Pin = 9; // 부저의 +측에 연결된 핀

int on_value = 30; //150 조정필요!
int off_value = 10; //30 조정필요!

bool is_auto_run = false;
bool is_fan_on = false;

MQUnifiedsensor MQ135(placa, Voltage_Resolution, ADC_Bit_Resolution, pin, type);

LiquidCrystal_I2C lcd(0x27, 16, 2); //LCD 초기화 (LCD 주소값, x축, y축)
DHT dht(4, DHT11); //온습도센서 초기화(온습도센서 핀 번호, 센서 타입)

//Carbonmonoxide라는 변수 선언과 미세먼지 값, 습도 값, 온도 값 문자열을 담기위한 배열 선언
int Carbonmonoxide = 0; //일산화탄소 초기값
int Tones[8] = {261, 294, 330, 349, 392, 440, 495, 524}; // 부저 소리 설정
int Tones_Number; // Tones에 들어갈 변수

char str1[10]; //일산화 탄소 값
char str2[10]; //습도 값
char str3[10]; //온도 값
```

```

//각 핀과 연결된 핀 번호를 저장할 변수 선언
int Vo = A0 ;
int Vo2 = A3 ;
int V_LED = 2 ;
int V_LED2 = 12 ;
int FAN = 8 ; //모스펫 8번핀 연결
int R_LED = 5 ; //LED 빨간불 5번핀 연결
int Y_LED = 6 ; //LED 노란불 6번핀 연결
int G_LED = 7 ; //LED 초록불 7번핀 연결
int angle = 0; // 서보모터 angle 값

//미세먼지값 측정을 위한 데이터값 변수 선언
float Vo_value = 0;
float Voltage = 0;
float a = 0;

//외기 미세먼지값 측정을 위한 데이터값 변수 선언
float Vo2_value = 0;
float Vo2ltage = 0;
float b = 0;

void setup()
{
  Wire.begin();
  // 시리얼 통신 시작
  Serial.begin(9600);

  while (!Serial);           // Leonardo: wait for serial monitor
  Serial.println("\nI2C LCD Scanner");
  // MQ-135에서 일산화탄소 농도만 추출
  MQ135.setRegressionMethod(1); // _PPM = a*ratio^b (PPM 농도와 상수 값을 계산하기 위한 수학 모델 설정)

  //전원을 켜올 때 LCD 초기값 설정
  lcd.begin();
  lcd.clear();
  lcd.noBacklight();
  delay(500);
  lcd.backlight();
  delay(500);
  lcd.setCursor(0,0);
  lcd.print("  AIR");
  delay(1000);
  lcd.setCursor(6,0);
  lcd.print("PURIFIER");
  delay(1000);
  lcd.setCursor(0,1);
  lcd.print("FANTASTIC TEAM 4");
  delay(1000);
  lcd.clear();

  //특수 문자 선언
  lcd.createChar(0, ug1);
  lcd.createChar(1, ug2);
  lcd.createChar(2, temp);
  lcd.createChar(3, C);
  lcd.createChar(4, humi);
  lcd.createChar(5, per);

```

```

MQ135.init();

Serial.print("Calibrating please wait.");

float calcR0 = 0;

for(int i = 1; i<=10; i++) {

    MQ135.update();    // 데이터를 업데이트하면 arduino가 아날로그 핀의 전압을 읽습니다

    calcR0 += MQ135.calibrate(RatioMQ135CleanAir);

    Serial.print(".");

}

MQ135.setR0(calcR0/10);

Serial.println("  done!.");


//핀모드 입출력 설정

pinMode(Dir1Pin_A, OUTPUT);
pinMode(Dir2Pin_A, OUTPUT);
pinMode(SpeedPin_A, OUTPUT);
pinMode(Vo, INPUT);
pinMode(V_LED, OUTPUT);
pinMode(Vo2, INPUT);
pinMode(V_LED2, OUTPUT);
pinMode(FAN, OUTPUT);
pinMode(R_LED, OUTPUT);
pinMode(Y_LED, OUTPUT);
pinMode(G_LED, OUTPUT);
servo.attach(10);    //멤버함수인 attach : 핀 설정

```

```

    byte error, address;
int nDevices;
// 시리얼 모니터 - I2C LCD 주소값을 출력

nDevices = 0;
for(address = 1; address < 127; address++ )
{
    // The i2c_scanner uses the return value of
    // the Write.endTransmission to see if
    // a device did acknowledge to the address.
    Wire.beginTransmission(address);
    error = Wire.endTransmission();

    if (error == 0)
    {
        Serial.print("I2C device found at address 0x");
        if (address<16)
            Serial.print("0");
        Serial.print(address,HEX);
        Serial.println(" !");

        nDevices++;
    }
    else if (error==4)
    {
        Serial.print("Unknown error at address 0x");
        if (address<16)
            Serial.print("0");
        Serial.println(address,HEX);
    }
}
if (nDevices == 0)
    Serial.println("No I2C devices found\n");
else
    Serial.print("");
}

```

```

    byte error, address;
int nDevices;
// 시리얼 모니터 - I2C LCD 주소값을 출력

nDevices = 0;
for(address = 1; address < 127; address++ )
{
    // The i2c_scanner uses the return value of
    // the Write.endTransmission to see if
    // a device did acknowledge to the address.
    Wire.beginTransmission(address);
    error = Wire.endTransmission();

    if (error == 0)
    {
        Serial.print("I2C device found at address 0x");
        if (address<16)
            Serial.print("0");
        Serial.print(address,HEX);
        Serial.println(" !");

        nDevices++;
    }
    else if (error==4)
    {
        Serial.print("Unknown error at address 0x");
        if (address<16)
            Serial.print("0");
        Serial.println(address,HEX);
    }
}
if (nDevices == 0)
    Serial.println("No I2C devices found\n");
else
    Serial.print("");
}

```

```

/*
전압[V] = (아날로그 핀 값) * 5.0 / 1024
미세먼지 센서의 출력전압 (Vo)는 먼지가 없을 때의 출력전압 (Voc) 과 미세먼지 농도에 비례하여 출력
*/
{Voltage = Vo_value * 5.0 / 1024.0; //내기 미세먼지 농도값 측정

if( Voltage < 0.1 ){

    in_Dust = 0 ;

}

else if ( Voltage < 0.8){

    in_Dust = ((10*(5*(Voltage - 0.1)))*0.6);

}

else if ( Voltage < 1 ){

    in_Dust = ((10 + 15*((Voltage - 0.8)*10))*0.6);

}

else if( Voltage > 1){

    in_Dust = ((40 + 16*((Voltage - 1)*10))*0.6);

}

}

void loop()
{
    unsigned char t = dht.readTemperature();
    unsigned char h = dht.readHumidity();
    MQ135.update(); // 데이터를 업데이트하면 arduino가 아날로그 핀의 전압을 읽습니다

    MQ135.setA(605.18); MQ135.setB(-3.937); // CO 농도를 얻기 위해 equation 값 구성

    float CO = MQ135.readSensor(); // 센서는 설정 전 또는 설정에서 설정된 모델과 a 및 b 값을 사용하여 PPM 농도를 읽습니다

    float in_Dust = 0; //내부 미세먼지 농도값 FLOAT
    float out_Dust = 0; //외부 미세먼지 농도값 FLOAT
    //적외선 발광 다이오드 (IRED)와 포토다이오드가 대각선으로 배치되어 공기 중 먼지의 반사광을 감지 (내기)
    //미세먼지 센서의 데이터값 계산을 통한 미세먼지값 측정 (내기)
    digitalWrite(V_LED, LOW);
    delayMicroseconds(280);
    Vo_value = analogRead(Vo);
    delayMicroseconds(40);
    digitalWrite(V_LED, HIGH);
    delayMicroseconds(9680);
    //적외선 발광 다이오드 (IRED)와 포토다이오드가 대각선으로 배치되어 공기 중 먼지의 반사광을 감지 (외기)
    //미세먼지 센서의 데이터값 계산을 통한 미세먼지값 측정 (외기)
    digitalWrite(V_LED2, LOW);
    delayMicroseconds(280);
    Vo2_value = analogRead(Vo2);
    delayMicroseconds(40);
    digitalWrite(V_LED2, HIGH);
    delayMicroseconds(9680);
}

```



```

{Vo2ltage = Vo2_value * 5.0 / 1024.0; //외기 미세먼지 농도값 측정

if( Vo2ltage < 0.1 ){

    out_Dust = 0 ;

}

else if ( Vo2ltage < 0.8){

    out_Dust = ((10*(5*(Vo2ltage - 0.1))));

}

else if ( Vo2ltage < 1 ){

    out_Dust = ((10 + 15*((Vo2ltage - 0.8)*10)));

}

else if( Vo2ltage > 1){

    out_Dust = ((40 + 16*((Vo2ltage - 1)*10)));

}

}

Serial.print(in_Dust);
Serial.print("ppm");
Serial.print(",");
Serial.print(out_Dust);
Serial.print("ppm");
Serial.print(",");
Serial.print(CO);
Serial.print("ppm");
Serial.print(",");
Serial.print(32+(1.8f*t)-(0.55f*((1-h/100.0f)*(1.8f*t-26)))); //볼레지수 계산법
Serial.println("°C");
delay(500);
//LCD 출력
lcd.setCursor(1,0);
lcd.print("CO : ");
lcd.print(CO);
lcd.write(0);
lcd.print("g/m³");
lcd.write(1); //특수 문자와 3글자로 된 미세먼지 값 출력

lcd.setCursor(1,1);
lcd.write(2);
lcd.print(" ");
sprintf(str2,"%02d",t);
lcd.print(str2);
lcd.write(3); //특수 문자와 2글자로 된 온도 값 출력
lcd.print(" ");
lcd.write(4);
lcd.print(" ");
sprintf(str3,"%02d",h);
lcd.print(str3);
lcd.write(5); //특수 문자와 2글자로 된 습도 값 출력

delay(1500);

```

```

if(Serial.available()){
//앱에서 사용자가 데이터를 전송해줬을 때
//'0', '1', '2', '3', '4'
char c = Serial.read(); //char 1개를 읽는 함수

if(c == '0'){
mode = 0;
is_auto_run = false;
is_fan_on = false;
analogWrite(FAN,0);
angle = 60; //환풍기 닫히는 각도
servo.write(angle); // 환풍기 닫음
digitalWrite(Dir1Pin_A,LOW);
digitalWrite(Dir2Pin_A,LOW);
digitalWrite(SpeedPin_A,0);
noTone(Buzzer_Pin);
digitalWrite(R_LED,LOW);
digitalWrite(Y_LED,LOW);
digitalWrite(G_LED,LOW);
//간이 공기청정기와 환풍 팬 모두 off
}

else if(c == '1'){
mode = 1;
analogWrite(FAN,0);
angle = 20; //환풍기 열리는 각도
servo.write(angle); //환풍기를 열
noTone(Buzzer_Pin);
digitalWrite(R_LED,LOW);
digitalWrite(Y_LED,LOW);
digitalWrite(G_LED,LOW);
digitalWrite(Dir1Pin_A,HIGH);
digitalWrite(Dir2Pin_A,LOW);
digitalWrite(SpeedPin_A,120);
//간이 공기청정기 off, 환풍 팬 1단 작동
}

if(Serial.available()){
//앱에서 사용자가 데이터를 전송해줬을 때
//'0', '1', '2', '3', '4'
char c = Serial.read(); //char 1개를 읽는 함수

if(c == '0'){
mode = 0;
is_auto_run = false;
is_fan_on = false;
analogWrite(FAN,0);
angle = 60; //환풍기 닫히는 각도
servo.write(angle); // 환풍기 닫음
digitalWrite(Dir1Pin_A,LOW);
digitalWrite(Dir2Pin_A,LOW);
digitalWrite(SpeedPin_A,0);
noTone(Buzzer_Pin);
digitalWrite(R_LED,LOW);
digitalWrite(Y_LED,LOW);
digitalWrite(G_LED,LOW);
//간이 공기청정기와 환풍 팬 모두 off
}

else if(c == '1'){
mode = 1;
analogWrite(FAN,0);
angle = 20; //환풍기 열리는 각도
servo.write(angle); //환풍기를 열
noTone(Buzzer_Pin);
digitalWrite(R_LED,LOW);
digitalWrite(Y_LED,LOW);
digitalWrite(G_LED,LOW);
digitalWrite(Dir1Pin_A,HIGH);
digitalWrite(Dir2Pin_A,LOW);
digitalWrite(SpeedPin_A,120);
//간이 공기청정기 off, 환풍 팬 1단 작동
}
}

```

```

else if(c == '2'){
    mode = 2;
    analogWrite(FAN,0);
    angle = 20; //환풍기 열리는 각도
    servo.write(angle); //환풍기를 열
    digitalWrite(Dir1Pin_A,HIGH);
    digitalWrite(Dir2Pin_A,LOW);
    digitalWrite(SpeedPin_A,150);
    noTone(Buzzer_Pin);
    digitalWrite(R_LED,LOW);
    digitalWrite(Y_LED,LOW);
    digitalWrite(G_LED,LOW);
    //간이 공기청정기 off, 환풍 팬 2단 작동
}
else if(c == '3'){
    mode = 3;
    analogWrite(FAN,0);
    angle = 20; //환풍기 열리는 각도
    servo.write(angle); //환풍기를 열
    digitalWrite(Dir1Pin_A,HIGH);
    digitalWrite(Dir2Pin_A,LOW);
    digitalWrite(SpeedPin_A,180);
    noTone(Buzzer_Pin);
    digitalWrite(R_LED,LOW);
    digitalWrite(Y_LED,LOW);
    digitalWrite(G_LED,LOW);
    //간이 공기청정기 off, 환풍 팬 3단 작동
}
else if(c == '4'){
    mode = 4;
    is_auto_run = true;
    analogWrite(FAN,0); //일단 멈춘 다음 조건을 본다
    is_fan_on = false;
    angle = 60; //환풍기 닫히는 각도
    servo.write(angle); //환풍기를 닫음
    digitalWrite(Dir1Pin_A,LOW);
    digitalWrite(Dir2Pin_A,LOW);
    digitalWrite(SpeedPin_A,0);
    noTone(Buzzer_Pin);
    digitalWrite(R_LED,LOW);
    digitalWrite(Y_LED,LOW);
    digitalWrite(G_LED,LOW);
    //AUTO모드로 작동
}

```

```

else if(c == '2'){
    mode = 2;
    analogWrite(FAN,0);
    angle = 20; //환풍기 열리는 각도
    servo.write(angle); //환풍기를 열
    digitalWrite(Dir1Pin_A,HIGH);
    digitalWrite(Dir2Pin_A,LOW);
    digitalWrite(SpeedPin_A,150);
    noTone(Buzzer_Pin);
    digitalWrite(R_LED,LOW);
    digitalWrite(Y_LED,LOW);
    digitalWrite(G_LED,LOW);
    //간이 공기청정기 off, 환풍 팬 2단 작동
}
else if(c == '3'){
    mode = 3;
    analogWrite(FAN,0);
    angle = 20; //환풍기 열리는 각도
    servo.write(angle); //환풍기를 열
    digitalWrite(Dir1Pin_A,HIGH);
    digitalWrite(Dir2Pin_A,LOW);
    digitalWrite(SpeedPin_A,180);
    noTone(Buzzer_Pin);
    digitalWrite(R_LED,LOW);
    digitalWrite(Y_LED,LOW);
    digitalWrite(G_LED,LOW);
    //간이 공기청정기 off, 환풍 팬 3단 작동
}
else if(c == '4'){
    mode = 4;
    is_auto_run = true;
    analogWrite(FAN,0); //일단 멈춘 다음 조건을 본다
    is_fan_on = false;
    angle = 60; //환풍기 닫히는 각도
    servo.write(angle); //환풍기를 닫음
    digitalWrite(Dir1Pin_A,LOW);
    digitalWrite(Dir2Pin_A,LOW);
    digitalWrite(SpeedPin_A,0);
    noTone(Buzzer_Pin);
    digitalWrite(R_LED,LOW);
    digitalWrite(Y_LED,LOW);
    digitalWrite(G_LED,LOW);
    //AUTO모드로 작동
}
}

```

//안드로이드 어플로 실시간 실내 미세먼지 농도, 실외 미세먼지 농도, 일산화탄소 농도, 불쾌지수를 확인 가능하게 하는Serial monitor

//앱 인벤터에 대한 자세한 설명은 4-18부터

```

//안드로이드 어플로 실시간 실내 미세먼지 농도, 실외 미세먼지 농도, 일산화탄소 농도, 불쾌지수를 확인 가능하게 하는 Serial monitor
//앱 인벤터에 대한 자세한 설명은 앱 인벤터 파트에서 설명
Serial.print(in_Dust);
Serial.print("ppm");
Serial.print(",");
Serial.print(out_Dust);
Serial.print("ppm");
Serial.print(",");
Serial.print(CO);
Serial.print("ppm");
Serial.print(",");
Serial.print(32+(1.8*f*t)-(0.55*f*((1-h/100.0f)*(1.8*f*t-26))));
Serial.println("°C");

delay(500);

//LCD 출력
lcd.setCursor(1,0);
lcd.print("CO : ");
lcd.print(CO);
lcd.write(0);
lcd.print("g/m³");
lcd.write(1); //특수 문자와 3글자로 된 미세먼지 값 출력

lcd.setCursor(1,1);
lcd.write(2);
lcd.print(" ");
sprintf(str2,"%02d",t);
lcd.print(str2);
lcd.write(3); //특수 문자와 2글자로 된 온도 값 출력
lcd.print(" ");
lcd.write(4);
lcd.print(" ");
sprintf(str3,"%02d",h);
lcd.print(str3);
lcd.write(5); //특수 문자와 2글자로 된 습도 값 출력

delay(1500);
}

```



```

if(is_auto_run){
  if (CO < 9){ //*****일산화탄소 대기상태 좋을 때-가장 높은 우선순위는 일산화탄소*****
    digitalWrite(FAN,LOW); //실내 간이 공기청정기 off
    digitalWrite(R_LED,LOW); // LED 빨간불 off
    digitalWrite(Y_LED,LOW); // LED 노란불 off
    digitalWrite(G_LED,HIGH); // LED 초록불 on(실내 대기와 실외 대기 모두 좋은 상태라는 뜻)
    angle = 60; //환풍기 달하는 각도
    servo.write(angle); //환풍기를 닫음
    digitalWrite(Dir1Pin_A,LOW);
    digitalWrite(Dir2Pin_A,LOW);
    digitalWrite(SpeedPin_A,0); //환풍 팬 작동 off
    noTone(Buzzer_Pin);
    if (out_Dust < 50){ //*****실외 미세먼지 좋을 때 (환경부에서 발표한 실외 미세먼지 PM2.5 권고 기준: 50ppm 미만)-두번째 우선순위*****
      digitalWrite(FAN,LOW); //실내 간이 공기청정기 off
      digitalWrite(R_LED,LOW); // LED 빨간불 off
      digitalWrite(Y_LED,LOW); // LED 노란불 off
      digitalWrite(G_LED,HIGH); // LED 초록불 on(실내 대기와 실외 대기 모두 좋은 상태라는 뜻)
      digitalWrite(Dir1Pin_A,LOW);
      digitalWrite(Dir2Pin_A,LOW);
      digitalWrite(SpeedPin_A,0); //환풍 팬 작동 off
      angle = 60; //환풍기 달하는 각도
      servo.write(angle); //환풍기를 닫음
      noTone(Buzzer_Pin);
      if (in_Dust < 35){ //*****실내 미세먼지 좋을 때 (환경부에서 발표한 실내 미세먼지 PM2.5 권고 기준: 35ppm 미만)-세번째 우선순위*****
        digitalWrite(FAN,LOW); //실내 간이 공기청정기 off
        digitalWrite(R_LED,LOW); // LED 빨간불 off
        digitalWrite(Y_LED,LOW); // LED 노란불 off
        digitalWrite(G_LED,HIGH); // LED 초록불 on(실내 대기와 실외 대기 모두 좋은 상태라는 뜻)
        digitalWrite(Dir1Pin_A,LOW);
        digitalWrite(Dir2Pin_A,LOW);
        digitalWrite(SpeedPin_A,0); //환풍 팬 작동 off
        angle = 60; //환풍기 달하는 각도
        servo.write(angle); //환풍기를 닫음
        noTone(Buzzer_Pin);
        if (32+(1.8*t)-(0.55*((1-h/100)*(1.8*t-26)))){ // 불쾌지수 68이상
          digitalWrite(FAN,LOW); //실내 간이 공기청정기 off
          digitalWrite(R_LED,LOW); // LED 빨간불 off
          digitalWrite(Y_LED,HIGH); // LED 노란불 on(실내, 실외 대기상태가 좋으나 실내 불쾌지수가 높아 노란불 on)
          digitalWrite(G_LED,LOW); // LED 초록불 off
          angle = 20; //환풍기 열리는 각도
          servo.write(angle); //환풍기를 엽
          digitalWrite(Dir1Pin_A,HIGH);
          digitalWrite(Dir2Pin_A,LOW);
          digitalWrite(SpeedPin_A,150); //환풍기 on
          noTone(Buzzer_Pin);
        }
      }
    }
  }
}

```



```

    }
}

else if (in_Dust >= 35) { //*****실내 미세먼지가 안 좋을 때 (환경부에서 발표한 실내 미세먼지 PM2.5 권고 기준: 35ppm 미만) *****
    digitalWrite(FAN,LOW); //실내 간이 공기청정기 off
    digitalWrite(R_LED,LOW); // LED 빨간불 off
    digitalWrite(Y_LED,HIGH); // LED 노란불 on(실내 대기상태가 좋지 못해 노란불 on)
    digitalWrite(G_LED,LOW); // LED 초록불 off
    angle = 20; //환풍기 열리는 각도
    servo.write(angle); //환풍기를 엮
    digitalWrite(Dir1Pin_A,HIGH);
    digitalWrite(Dir2Pin_A,LOW);
    digitalWrite(SpeedPin_A,150); //환풍기 on
    noTone(Buzzer_Pin);
}
}

else if(out_Dust >=50){ //*****실외 미세먼지 좋을 때 (환경부에서 발표한 실외 미세먼지 PM2.5 권고 기준: 50ppm 미만) *****
    digitalWrite(FAN,LOW); //실내 간이 공기청정기 off
    digitalWrite(R_LED,LOW); // LED 빨간불 off
    digitalWrite(Y_LED,HIGH); // LED 노란불 on(실내 미세먼지는 좋으나 실외 미세먼지가 높아 노란불 on)
    digitalWrite(G_LED,LOW); // LED 초록불 off
    angle = 60; //환풍기 닫히는 각도
    servo.write(angle); //환풍기를 닫음
    digitalWrite(Dir1Pin_A,LOW);
    digitalWrite(Dir2Pin_A,LOW);
    digitalWrite(SpeedPin_A,0); //환풍 팬 작동 off
    noTone(Buzzer_Pin);
    if(in_Dust >= 35){ //*****실내 미세먼지가 안 좋을 때 (환경부에서 발표한 실내 미세먼지 PM2.5 권고 기준: 35ppm 미만) *****
        digitalWrite(FAN,HIGH); // (실내 대기 상태가 좋지 않아 환풍팬이 돌아야 하지만 실외가 좋지 않음. 따라서 실내 간이 공기청정기가 돌아감)
        digitalWrite(R_LED,LOW); // LED 빨간불 off
        digitalWrite(Y_LED,HIGH); // LED 노란불 off
        digitalWrite(G_LED,LOW); // LED 초록불 off
        angle = 60; //환풍기 닫히는 각도
        servo.write(angle); //환풍기를 닫음
        digitalWrite(Dir1Pin_A,LOW);
        digitalWrite(Dir2Pin_A,LOW);
        digitalWrite(SpeedPin_A,0); //환풍기 off
        noTone(Buzzer_Pin);
    }
}
}

else if (CO >= 9) { //*****일산화탄소 실내 대기 상태 매우 안 좋을 때 (가장 높은 우선순위는 일산화탄소)*****
    digitalWrite(FAN,LOW); //실내 간이 공기청정기 off
    digitalWrite(R_LED,HIGH); // LED 빨간불 on(일산화탄소가 높을 때만 빨간불이 켜지며 가장 위험한 상태를 알려줌)
    digitalWrite(Y_LED,LOW); // LED 노란불 off
    digitalWrite(G_LED,LOW); // LED 초록불 off
    angle = 20; //환풍기 열리는 각도
    servo.write(angle); //환풍기를 엮
    digitalWrite(Dir1Pin_A,HIGH);
    digitalWrite(Dir2Pin_A,LOW);
    digitalWrite(SpeedPin_A,255); //환풍기 최대로 on
    for (Tones_Number = 0 ; Tones_Number < 8; Tones_Number++) {
        tone(Buzzer_Pin, Tones[Tones_Number]); // tone(출력 핀 번호, 주파수 값)
        delay(350); //실내에 일산화탄소 농도가 높아 매우 위험한 상태이기 때문에 부저가 울림
    }
}
}

```

```

Serial.print(in_Dust);
Serial.print("ppm");
Serial.print(", ");
Serial.print(out_Dust);
Serial.print("ppm");
Serial.print(", ");
Serial.print(CO);
Serial.print("ppm");

Serial.print(", ");
Serial.print(32+(1.8f*t)-(0.55f*((1-h/100.0f)*(1.8f*t-26))));
Serial.println("°C");
delay(500);

lcd.setCursor(1,0);
lcd.print("CO : ");
lcd.print(CO);
lcd.write(0);
lcd.print("g/m");
lcd.write(1);    //특수 문자와 3글자로 된 미세먼지 값 출력

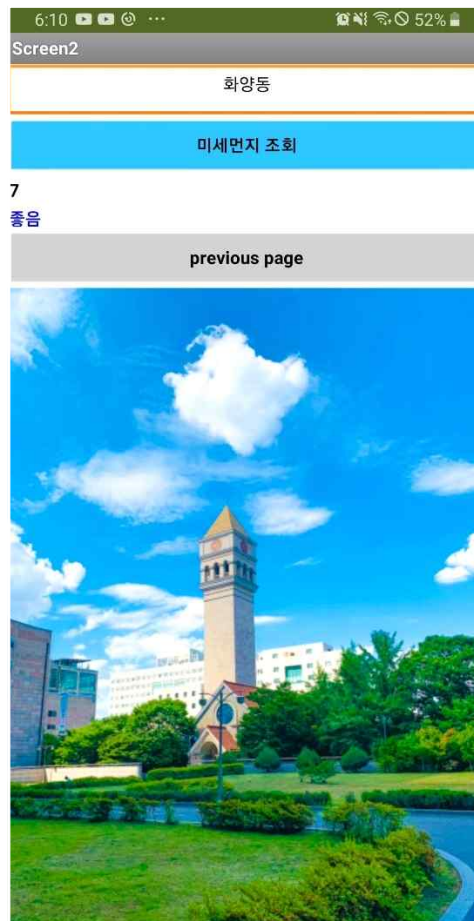
lcd.setCursor(1,1);
lcd.write(2);
lcd.print(" ");
sprintf(str2,"%02d",t);
lcd.print(str2);
lcd.write(3);    //특수 문자와 2글자로 된 온도 값 출력
lcd.print(" ");
lcd.write(4);
lcd.print(" ");
sprintf(str3,"%02d",h);
lcd.print(str3);
lcd.write(5);    //특수 문자와 2글자로 된 습도 값 출력

delay(1500);

}
}

```

4.18) 앱 인벤터 디자인



첫번째 페이지에서는 실내 미세먼지 농도와 실외 미세먼지 농도, 일산화탄소 농도, 불쾌지수를 실시간으로 확인할 수 있다. 그리고 두 번째 페이지에서는 실시간으로 원하는 위치를 검색하면 현재 미세먼지 농도와 상태를 확인할 수 있다.

첫 번째Screen

```

언제 ListPicker1 ▾ .선택전에
실행 지정하기 ListPicker1 ▾ . 요소 > 값 > BluetoothClient1 ▾ . 주소와이름 >

```

```

언제 ListPicker1 ▾ .선택후에
실행 지정하기 ListPicker1 ▾ . 선택된항목 > 값 > 호출 BluetoothClient1 ▾ .연결 주소 ListPicker1 ▾ . 선택된항목 >

```

-
- ```

when timer starts
 if (BluetoothClient1 is connected)
 call BluetoothClient1 to receive bytes 0
 else
 make local variable data with initial value call BluetoothClient1 to receive text bytes
 set data value to replace string in text text get data find string wrwn replace with space
 set data value to split all text text get data delimiter space
 set Label1 text value concatenate indoor PM2.5 concentration select item list get data position 1
 set Label2 text value concatenate outdoor PM2.5 concentration select item list get data position 2
 set Label3 text value concatenate carbon monoxide concentration select item list get data position 3
 set Label4 text value concatenate discomfort index select item list get data position 4

```

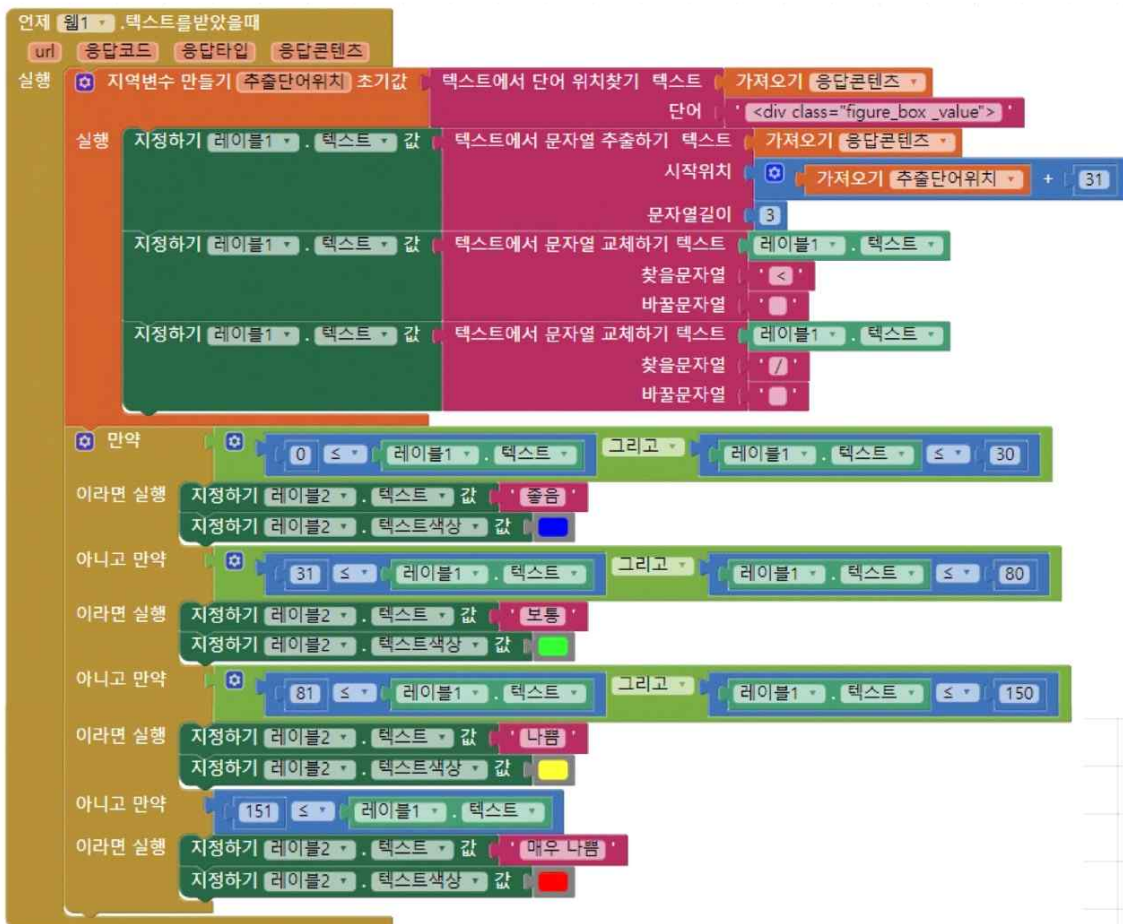
- ① data를 리스트 형식으로 만들어 각각의 Label을 선택할 때 알맞은 인덱스를 호출한다.  
 ② 합치기 블록을 통해 문자열을 서로 연결시킨다.  
 EX) [실내 미세먼지 농도, 실외 미세먼지 농도, 일산화탄소 농도, 불쾌지수]<sup>T</sup>



①블루투스가 연결되어 있을 때 해당 버튼을 누르면 해당 클라이언트로 메시지 전송  
→각각의 맞는 텍스트를 전달 EX) 0,1,2,3,4



## 두 번째 Screen



- ①버튼을 선택시 네이버 날씨정보 창에서 미세먼지 단어를 찾는다.
- ②미세먼지 페이지 소스코드에서 미세먼지에 대한 특정 코드를 찾는다.
- ③추출단어 개수를 선택하고 그 중 찾을 문자열길이를 선택한다.
- ④ 추출 확인 결과 미세먼지 농도 + "<" 와 "/" 가 같이 추출되기 때문에 문자열을 0으로 교체
- ⑤ 조건문을 통해 미세먼지 농도에 따른 날씨 상태를 표시한다.



## (5) 하드웨어

### 5.1) 터보기계에 대한 이해

터보기계(turbomachines)는 동역학적으로 유동 에너지를 더해 주거나 추출하는 유체기계를 말하며, 매우 다양한 형상으로 사용된다. 일반적으로 펌프(pump)는 유체에 에너지를 더해준다. 즉 펌프는 흐르는 유체에 일을 하여 유체의 압력을 증가시킨다. “펌프”라는 용어는 일반적으로 유체의 에너지를 끌어올리는 기계를 총칭하는데, 펌프(pump), 팬(fan), 송풍기(blower), 압축기(compressor) 등이 여기에 포함된다. 스마트 환기 시스템을 분석하기 위해 환풍팬이 펌프의 역할을 어떤 식으로 수행하고 있는지 계산해볼 필요가 있다. 스마트 환기 시스템의 효과를 유체역학 관점에서 분석하고자한다.

### 5.2) 환풍기 디자인

터보기계는 로터(rotor)에 부착된 여러 개의 날개로 구성되고, 유체와 기계 사이의 동역학적 상호결과로 유체에 에너지를 공급해주거나 유체로부터 에너지를 추출하는 기계장치이다. 사용유체는 기체, 증기, 액체로 스마트 환기 시스템에는 기체의 유동이 발생한다. 많은 터보기계는 로터를 둘러싸고 있는 여러 종류의 하우스(housing)이나 케이싱(casing)을 가지고 있지만, 창문용팬에 종종 날개를 둘러싸는 덕트를 가지고 있지 않은 것도 있다. AK의 환풍팬은 날개를 둘러싸고 있는 케이스가 존재하고, 셔터식 환풍팬을 설계했다.

### 5.3) 팬 날개 설계

스마트 환기 시스템 효과를 구현하기 위해 외부로부터의 유입되는 이물질 및 우천 시 영향을 최소화 하기 위하여 환풍팬 전면에 셔터로 보호하고자 한다. 동시에 환풍기 작동시 오염공기 배출에 방해가 되지 않도록 기체 유동의 흐름에 유의하였다. 특히 실내의 오염공기가 재유입되는 것을 최우선으로 지양하였으며 이는 관련 논문(1)을 참고하였다. 유동해석을 참조하여 재유입되는 30도의 모델을 확인하여, 이에 30도 이하의 각도는 배제하였다. 센서를 통해 받은 실내의 공기질 데이터에 따라 1,2,3단계로 구분하였으며 이때 팬의 각도는 각각 45도, 60도, 90도가 되도록 설계하였다.

$$\Delta P = \left(P_1 + \frac{\rho V_1^2}{2}\right) - \left(P_2 + \frac{\rho V_2^2}{2}\right) + \rho g(z_1 - z_2)$$

$\Delta P$  : 두 지점에서의 전체 압력강하, Pa

$P_1, P_2$  : 입력과 출력의 정압, Pa

$V_1, V_2$  : 두 지점의 유속, m/s

$\rho$  : 공기밀도, kg/m<sup>3</sup>

$g$  : 중력가속도, 9.81m/s<sup>2</sup>

$z_1, z_2$  : 입구와 출구의 높이, m

베르누이 방정식을 사용하여 실내와 실외를 비교하는 계산을 하였다. 높이에 의한 압력차는 없다고 가정하여 계산을 진행하였다.

$$\Delta P = P_j - P_i + P_s + P_w$$

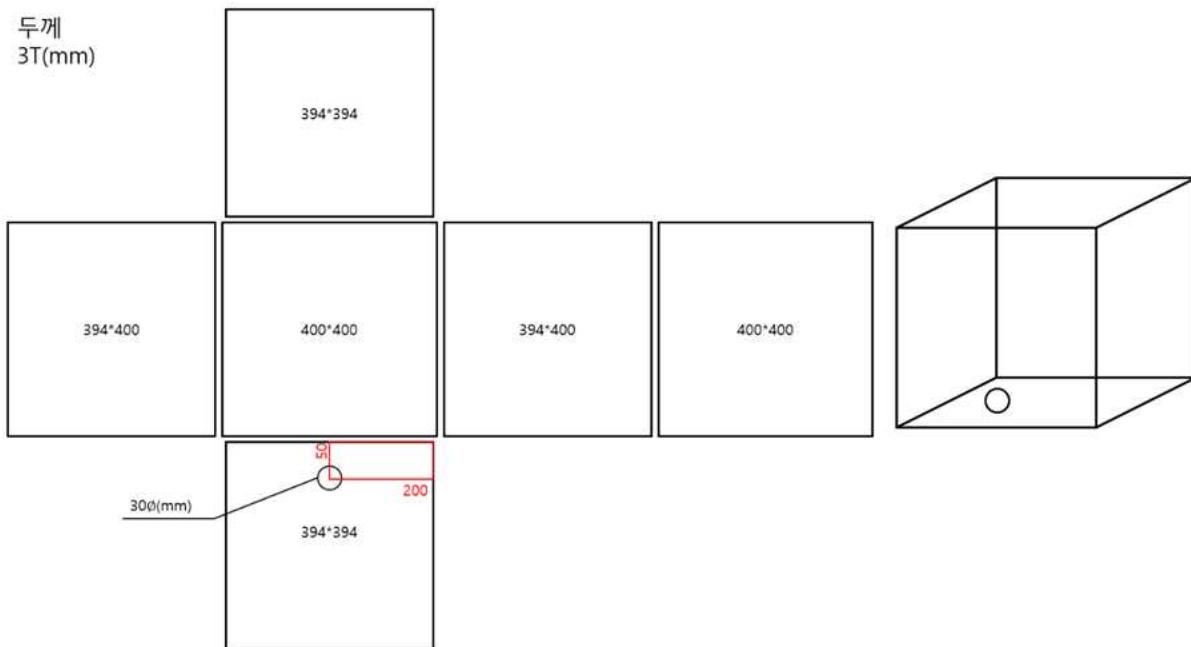
$P_i, P_j$  : 존 i와 j에서의 전체 압력, Pa

$P_s$  : 연돌효과에 의한 압력, Pa

$P_w$  : 바람에 의한 압력, Pa

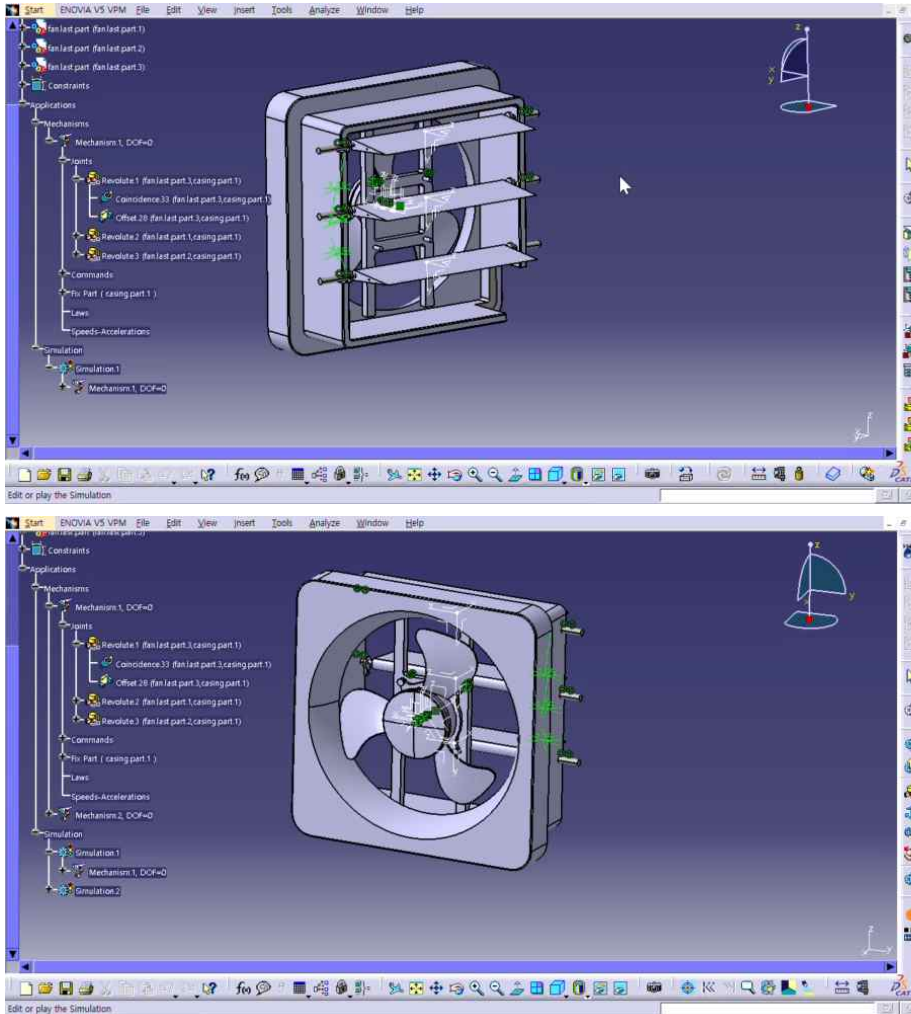
전체적인 결과를 더한 후 연돌효과와 바람에 의한 압력을 추가하여 계산하였다. 연돌효과란 내부공기의 상태와 외부공기 상태의 차이에서 발생하는 압력에 의해 발생하는 공기의 흐름을 의미한다. 실내와 실외의 경계선에 있는 팬의 각도 뿐 아니라 공기의 흐름을 고려하며 설계의 타당성을 확인하였다.

#### 5.4) 시연 공간 설계

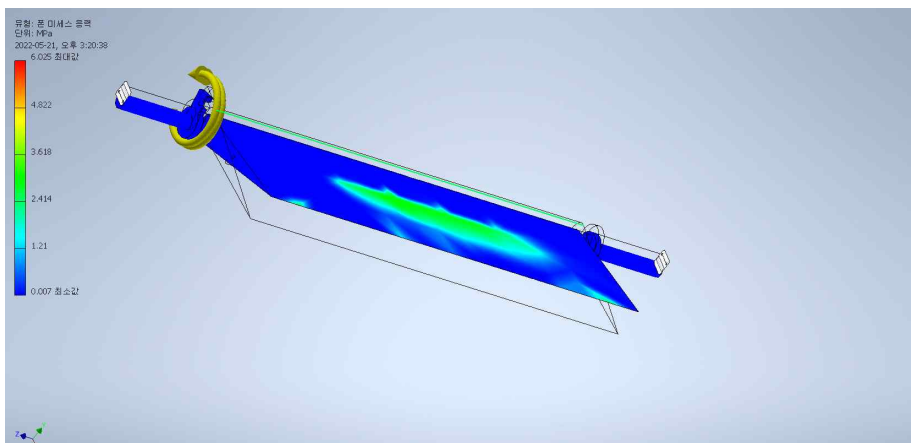


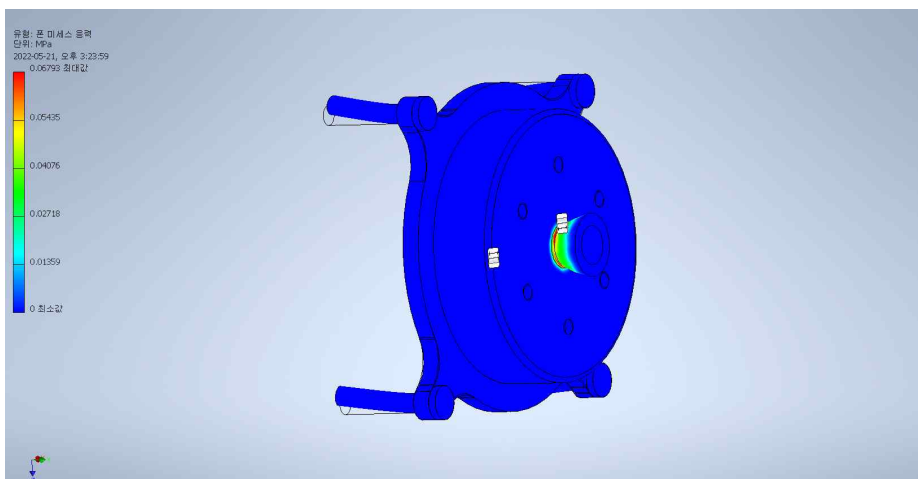
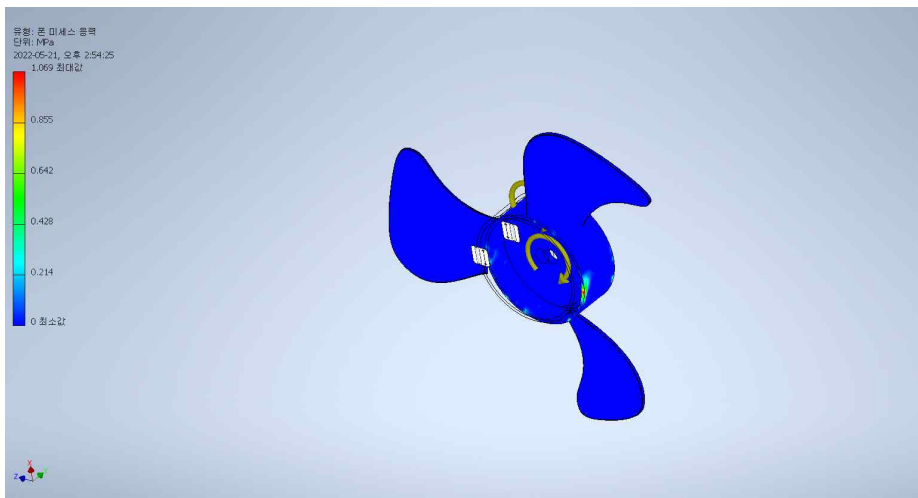
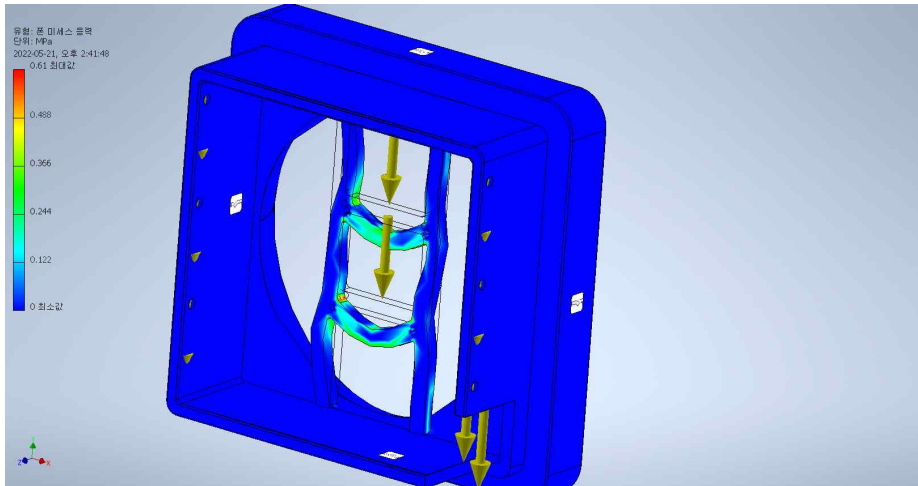
환기 시스템의 작동을 보여줄 수 있는 시연 공간을 설계하였다. 상방에는 시연할 스프레이 및 이산화탄소를 넣을 수 있도록 고려하였고 동시에 아두이노들을 넣을 수 있도록 설계하였다. 공기 투입구를 만들어 실제 집의 환기를 표현하기 위하여 공기 투입구를 설계하였다. 환풍기는 공기 투입구와 다른 면에 설계하였으며 아두이노 센서와 선 연결을 고려하여 모듈은 환풍기 반대쪽에 배치하도록 설계하였다. 시연을 보여주기 위해 재질은 아크릴을 선택하였고 크기는 부품배치와 시연을 고려하여 400mm x 400mm로 아크릴판을 구매하여 설계하였다.

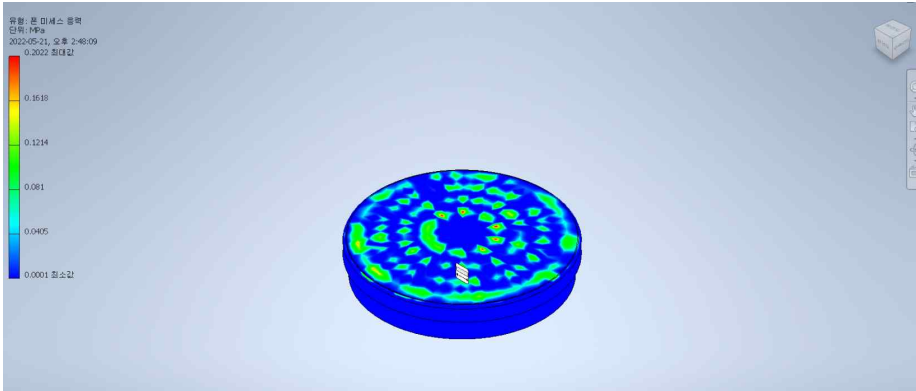
## 5.5) DMU 및 해석



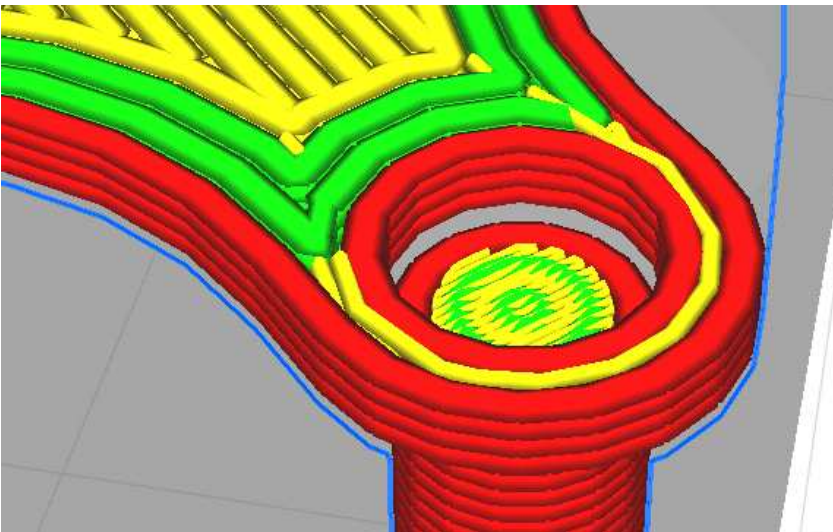
카티아를 이용한 DMU를 통해 부품의 결합과 4조가 설계한 의도와 맞는 것인지 확인하였다. 이를 통해 부품들의 결합과 치수를 확인하고 올바르게 작동할 수 있음을 알 수 있었다.







인벤터를 이용하여 설계한 부품들의 fem해석을 진행하였다. 각 부품의 조립 및 쓰임새에 맞게 구속조건을 부여하였다. 케이스는 4면에 고정을 부여, 회전하는 부품들은 조립 방향 법선에 고정하는 구속을 주었으며 동시에 3D프린터로 제작할 것을 고려하여 재질을 플라스틱으로 설정하였다. 각 부품들의 작동에 맞게 서보모터, DC 모터의 rpm과 토크를 계산하여 외력을 부여하였다.



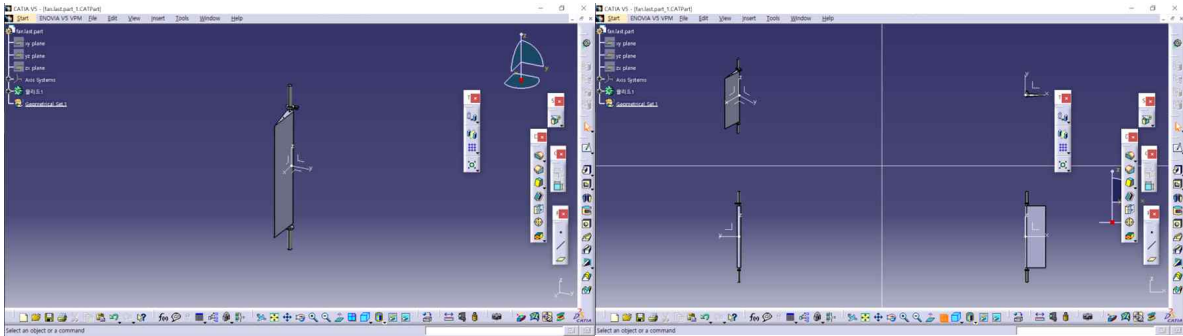
해석결과 안전을 확인하였으며 취약한 결합부분에 보강을 하는 조치를 취하였다.

$$\omega [rad/s] = \frac{2\pi}{60} N [rpm] \quad W' [kW] = \frac{T' [Nm] \omega [rad/s]}{1,000}$$

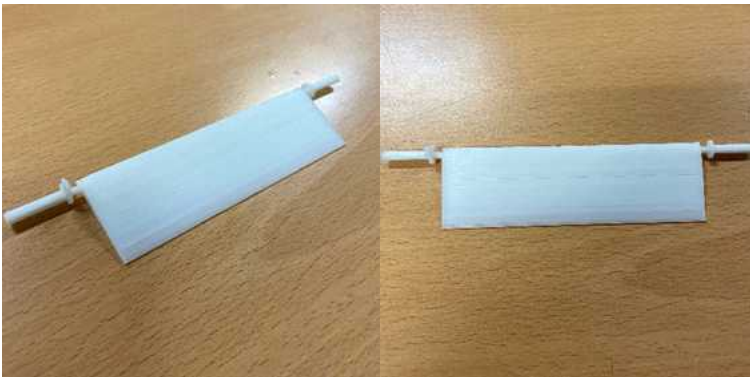


## (6) 설계 결과물 3D 모델링

### 6.1) 창문

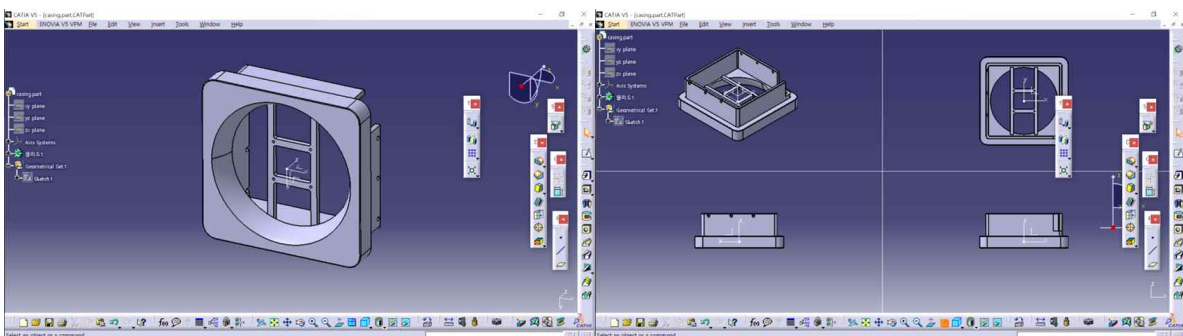


3개의 환풍기 창문을 제어하기 위해 고리를 창문 끝에 추가하여 설계한다.

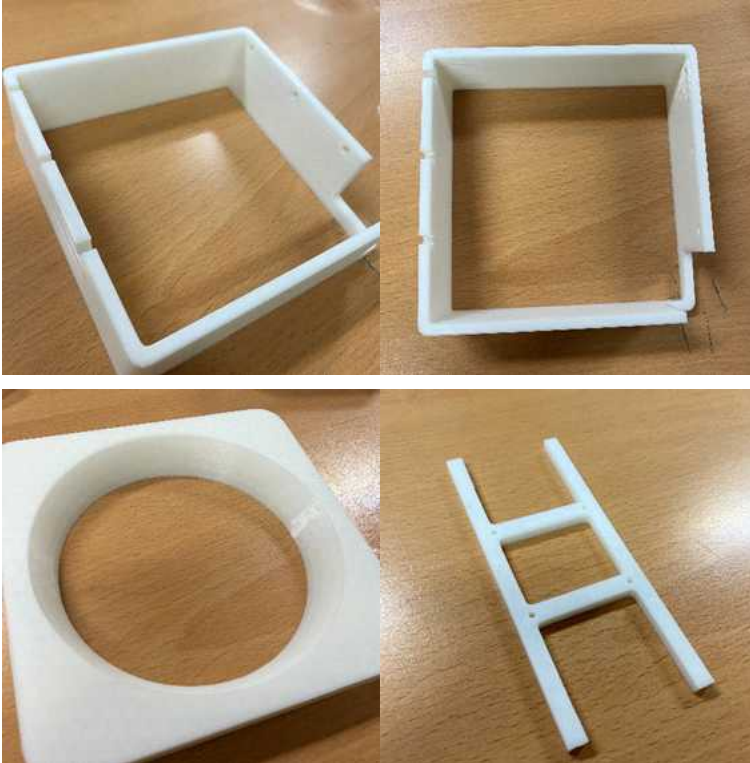


1차적으로, 같은 형상으로 3개 프린트 한다. 각각의 고리에 연결된 실을 서보모터로 한번에 제어하는 방법을 택한다. 추후 기어를 이용하여 3개의 shutter를 제어하는 방법으로 변경한다.

### 6.2) 환풍기 외곽

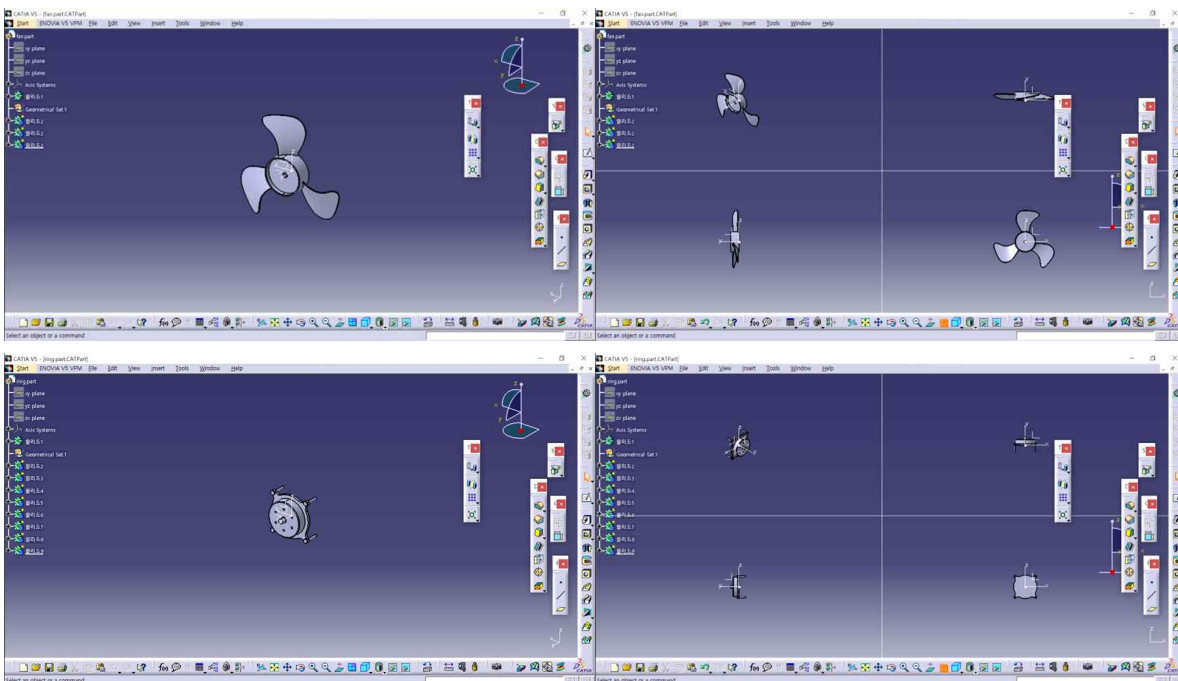


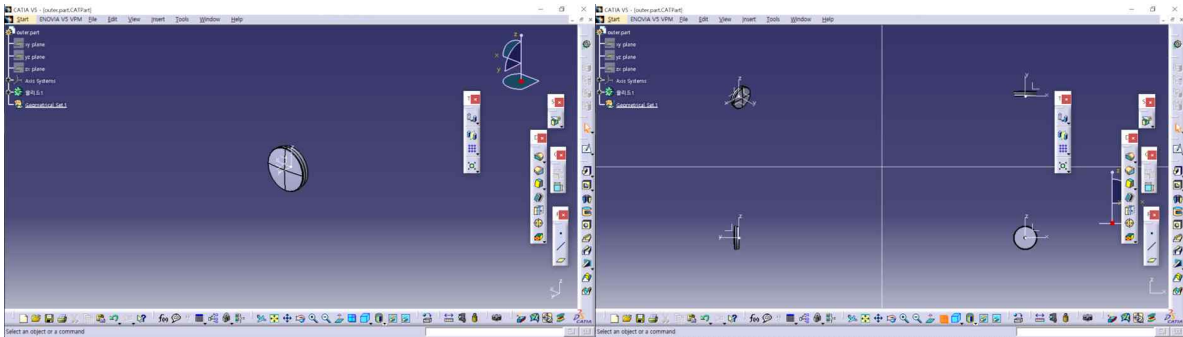
환풍기 외곽에 원활한 환풍기 창문 결합을 위해 홈을 파고 서보모터 장착을 위한 공간을 고려하여 설계한다.



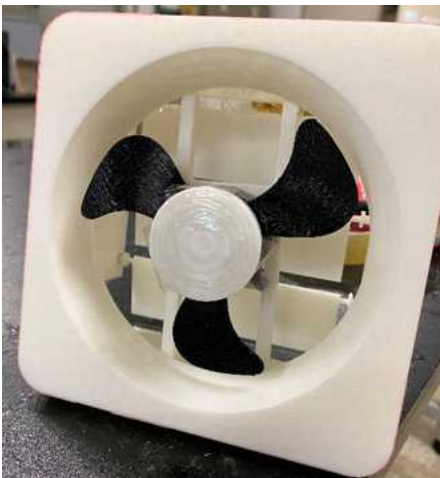
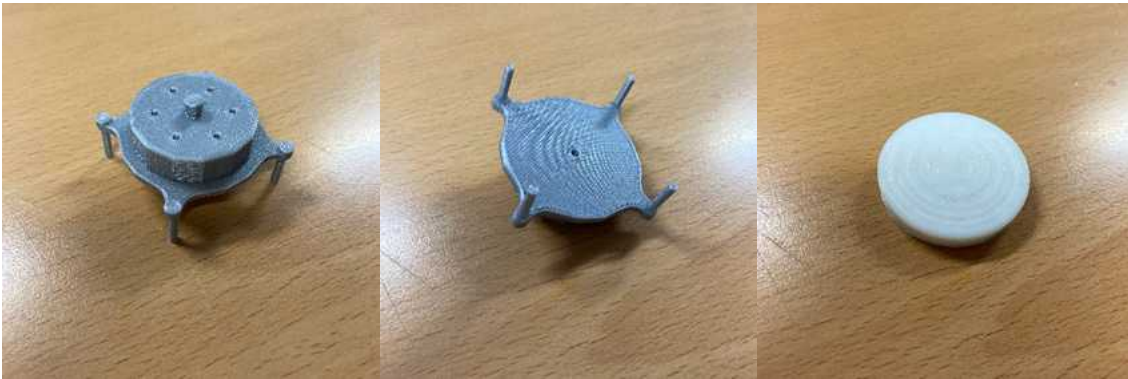
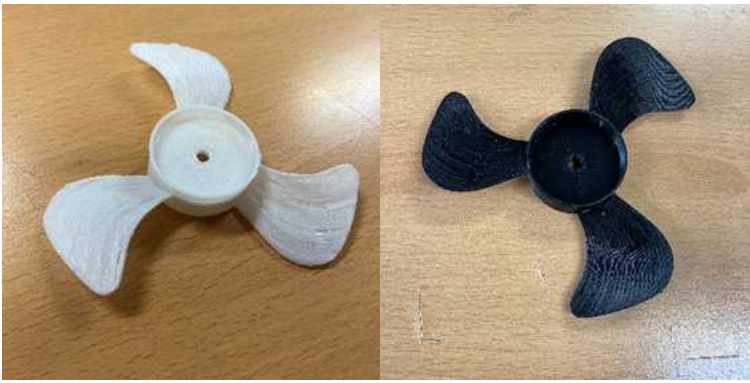
원활한 3D프린팅을 위해 외곽을 3부위로 나누어 각자 프린트 한다.

### 6.3) 팬



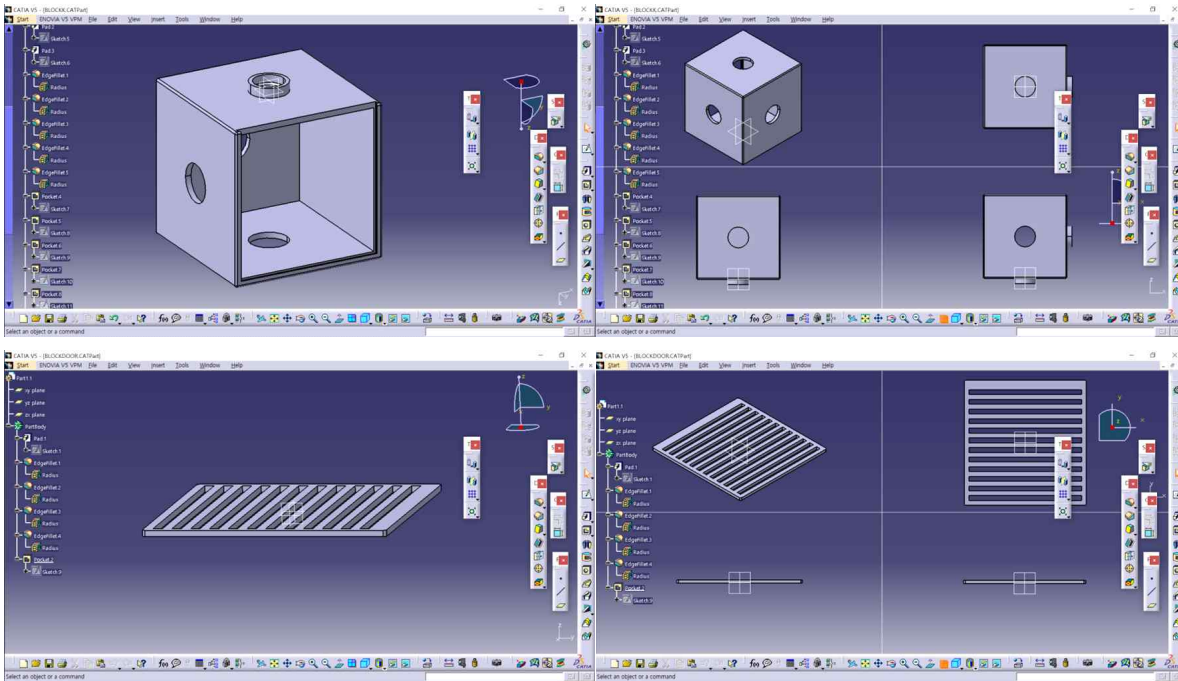


위 내용과 같이 팬의 개수와 날개를 고려하여 팬과 DC모터 결합 부품을 설계한다.



프린트 완료한 환풍기를 가결합 완료한 모습이다.

#### 6.4) 모듈화 틀

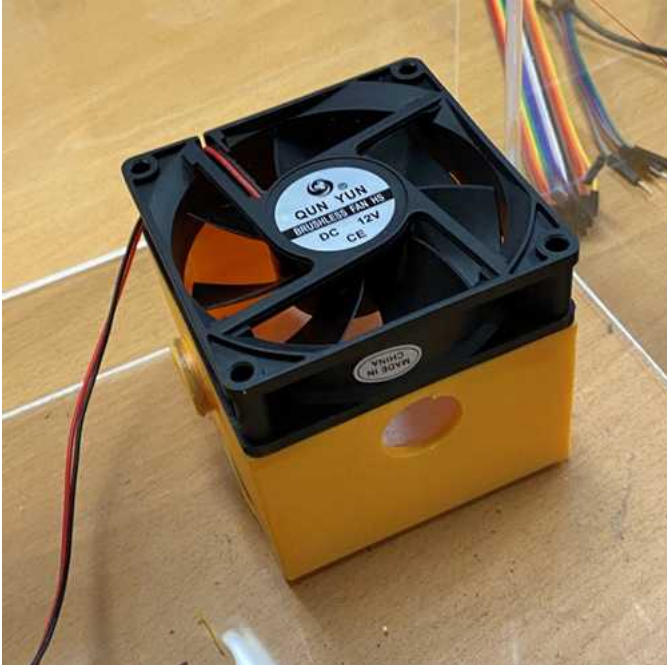


여러 개의 모듈을 서로 결합과 분리가 가능하게 pocket으로 파내고 치수 공차 크기를 고려하여 pad로 쌓았다. 모듈 안에 아두이노 부품들을 넣고 셔터형 창으로 닫는 형태를 택한다.



친환경적이고 무독성인 PLA필라멘트를 이용하여 프린트 한다. 3개의 모듈을 프린트하여 각각 온습도센서, 공기질센서, 미세먼지센서를 넣어 사용한다.

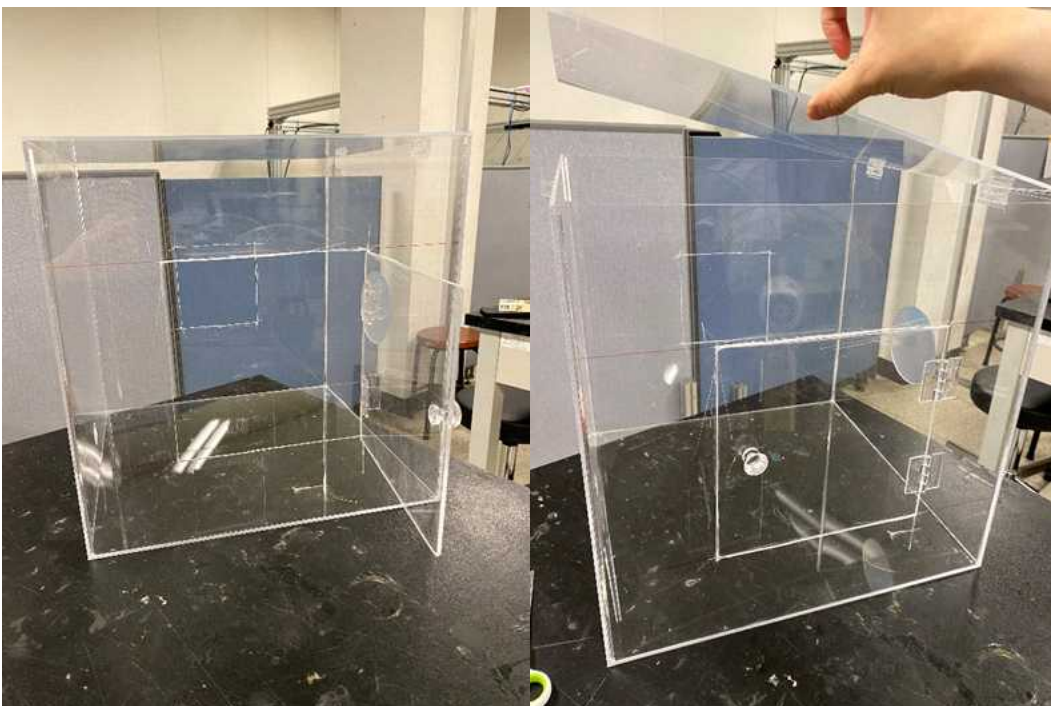




모듈 박스를 알맞은 크기로 절단하여 팬을 접착하여 간이 공기청정기를 만든다. 이는 실외 미세먼지 수치가 높아 환기를 할 수 없는 상황에 작동하여 실내 공기를 정화시키는 역할을 한다.

#### 6.5) 시연 공간

시연 공간은 3mm 두께의 아크릴 판으로 만든다. 모듈의 크기를 고려하여 400mm의 정육면체를 공간을 제작한다. 두께가 3mm인 것을 감안하여 측면의 판 크기를 결정한다. 왼쪽 아크릴판은 환풍기, 오른쪽 아크릴판은 수동으로 개폐 가능한 창문을 위해 정사각형의 크기로 절단한다. 뒤의 아크릴판은 여러 불순물 주입을 위한 지름 30mm의 원 모양의 구멍을 포함한다. 위의 아크릴판은 힌지를 이용하여 뚜껑 구조로 열고 닫을 수 있게 제작한다.



메틸렌 클로라이드 접착제를 이용하여 아크릴 판 공간을 제작한다. 습한 환경을 욕안으로 편하게 확인하기 위해 측면에 작은 거울을 접착한다.

## (7) 환기시스템 시장 분석

환기 시장은 미세먼지와 코로나와 같은 이슈들을 바탕으로 계속해서 발전해 나갈 것이다. 공기청정기와 실내 공기질 관리 시스템의 개발과 연구가 활발하게 진행되고 있으며 현재 시장에 나와있는 제품들은 다음과 같은 특징이 있다.

### 7.1) 미니빅 윈도우미니



미니빅 윈도우미니는 209,000원 상당의 제품으로 블루투스 통신 거리인 10~15m 내외에서만 제어가 가능하다. 1.5m\*1.5m 즉, 2.25m<sup>2</sup> 이상의 창문에서는 사용이 불가능하며, 미닫이 창문에만 사용할 수 있다. 설치과정에서 랙과 브라켓을 수평으로 부착하지 못할 시 파손 위험이 있다는 단점이 존재하는 반면에 AK의 스마트 환기 시스템은 조립과 설치가 용이하다. 미니빅 윈도우미니는 단순히 비가 오거나 오지 않는 상황만 제어 될 뿐, 실내 공기질의 상태를 센싱하고 유지 및 관리하는 기능이 없을 뿐만 아니라 기상 상황에 따른 상세한 제어가 어렵다.

### 7.2) SNUG 창문형 스마트 환풍기 창문 창틀



SNUG 창문형 스마트 환풍기 창문 창틀은 158,000원 상당의 제품으로 창문 레일 두께가 12mm 이하, 높이가 1300mm 까지의 창에만 설치가 가능하다는 제한요소가 존재한다. 즉 최소 200mm\*400mm 공간의 창문 크기를 가져야 설치가 가능하다. 작동과정 또한 기본의 창문을 열어둔 상태에서 끼워넣기 형식으로 설치하는 제품이라 틈새가 발생할 수 있으며 이런 틈새는 건축학적으로 열교가 되어 에너지효율을 떨어트리고, 외기의 미세먼지와 같은 오염물질의 유입을 완벽히 차단하기 어려운 구조를 만들어낸다. 설치에 제한적인 제품임과 동시에 환풍구의 수가 적고, 출력이 크지 않아 유량의 이동이 원활하게 이뤄지지 않는다는 점에서 환풍기의 성능이 부족하여, 환기 창문의 기능을 수행하기에 어려움이 있다.

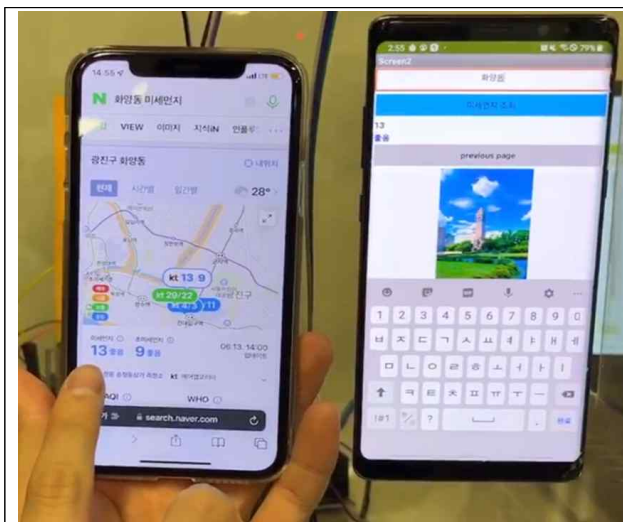
반면 AK의 스마트 환기 시스템은 시중에 나와있는 제품처럼 기존 창에 설치하는 형식이 아니므로 창 크기에 제약받지 않고, 소음이 전혀 없다는 점에서 소비자의 요구 조건을 충족한다. 소비전력 또한 앞선 제품보다 적으며 방충방과 같은 시설물의 간섭에서 자유로워 공간적인 제한에서 벗어날 수 있다. 양쪽 환기를 이용함으로



써 효율적인 실내 공기질 관리를 할 수 있을 뿐 아니라 상세하고 보다 다양한 기상 상황과 대기 상황에 따른 자동적인 환기와 창문 개폐가 가능하다는 점에서 단순한 기계적 작동이 아닌 정밀한 관측값에 따른 자동제어가 가능하다는 점이 AK의 스마트 환기 시스템의 장점이라 할 수 있다.

### 3. 결과 및 토의

#### (1) 시뮬레이션



→앱 인벤터로 만든 어플에서 실시간으로 ‘화양동’의 미세먼지 상태를 알 수 있다. 현재 미세먼지 농도를 알 수 있다면 원격으로 환기시스템을 on 또는 off 할 수 있다.

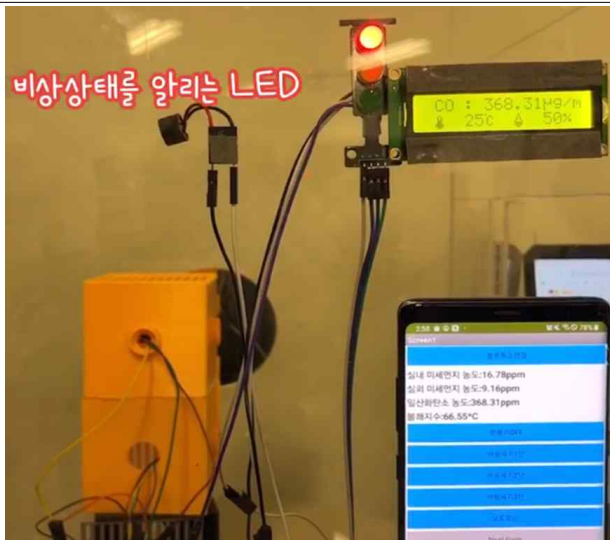


→앱으로 인해 환풍팬이 원격으로 작동하는 모습이다. 원격 버튼으로는 환풍 팬을 멈추게 하는 off 기능, 환풍기 속도에 따른 1단, 2단, 3단, 저절로 환풍팬을 조절하는 auto모드가 있다.

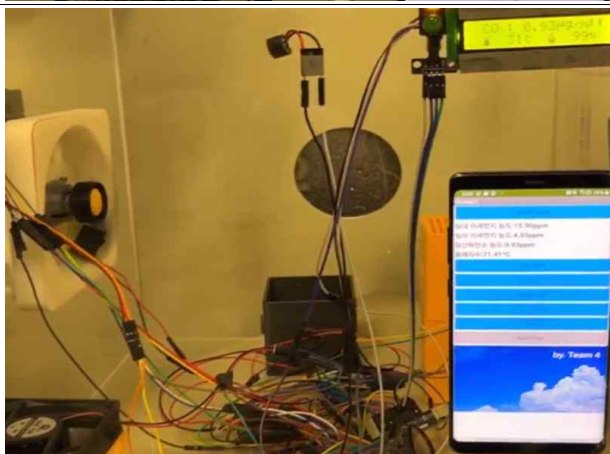
또한, 실시간으로 실내 미세먼지 농도, 실외 미세먼지 농도, 일산화탄소 농도, 불쾌지수를 실시간으로 앱을 통해 알 수 있다.



→실내 미세먼지 상태가 높아 환풍기가 도는 상태에서 실외 미세먼지 농도가 나빠진 상태이다. 실내 미세먼지 농도보다 실외 미세먼지 농도가 높기 때문에 환풍기는 멈추고 갓도 닫히게 된다. 하지만 실내 미세먼지 농도는 여전히 높기 때문에 실내 미세먼지 농도를 줄이기 위해서 간이 공기청정기가 자동으로 작동을 하는 모습을 보여주고 있다.



→센싱을 하는 값 중 가장 높은 우선순위는 일산화탄소의 농도이다. LED의 빨간불은 일산화탄소가 기준치 이상으로 높아졌을 때만 들어오게 된다. 이것은 매우 위험한 상태임을 알리며 부저도 같이 울리게 된다. 비상상태이기 때문에 환풍기는 최대로 돌아가게 설정해 놓았다.



→다음 상황은 모든 미세먼지 농도와 일산화탄소 농도가 정상일 때이다. 반면, 불쾌지수만 높은 상태이다. 불쾌지수는 가장 낮은 우선순위이기 때문에 모든 것이 정상일 때만 영향을 줄 수 있는 센싱 값이다. 다음 사진은 불쾌지수가 높아져 환풍기가 돌아가는 모습이다.

## (2) 의의 및 기대효과

어떤 오염물질이 실내 공기질에 부유하고 있는지 사람의 맨눈으로 파악하는 것은 쉽지 않다. 스마트 환기 시스템을 이를 정확하게 측정하는 것에 머무르지 않고, 창문 개폐와 공기청정기 및 사용자가 이를 확인할 수 있는 애플리케이션 제공을 한다는 점에서 안전하면서도 쾌적한 생활을 보장한다. 스마트 환기 시스템을 구성하고 있는 각종 센서 모듈은 상황에 따라 조절할 수 있으며, 간편하게 설치할 수 있다는 장점이 있다. 블록형 모듈로 센서들끼리의 조립과 설치, 관리가 용이할 뿐만 아니라 사용자가 원하는 센서들로만 스마트 환기 시스템을 직접 구성하여 사용할 수 있다. 모듈의 크기와 조립방법을 통일하여 대량생산시 제품마다 설계를 달리해야하는 번거로움을 제거했으며, 상용화가 유리하다.

## 4. 결 론

AK의 스마트 환기 시스템은 자동으로 실내 공기질 상태를 센싱하고 그 센싱 값에 따라 창문개폐가 이뤄진다. 실외 미세먼지 농도가 매우나쁨일 때는 자연환기가 이뤄지지 않고, 기계적 환기인 공기청정기를 작동함으로써 실내 공기질을 유지하는 것을 돕는다. 스마트폰을 통해 수동조절도 가능할 뿐만 아니라 실시간 모니터링을 가능하게 함으로써 사용자에게 편리함을 지원한다. 미세먼지, 일산화탄소, 온습도를 대표적인 지표로 설정해 실내를 쾌적하게 유지함으로써 삶의 질을 향상시키고 건강한 삶을 보조할 수 있다.

## 후 기

스마트 환기 시스템을 개발하면서 현재 1인 가구와 노후화된 건물은 실내 공기질 관리가 어렵다는 것을 파악했다. 협소한 공간에서의 높은 습도는 곰팡이와 바이러스의 확산이 빠르게 이뤄지며, 환기가 매우 필수적이지만 실질적으로 잘 이뤄지기 어려운 점에 중점을 두고 누구나 쉽게 설치하고 모니터링할 수 있는 스마트 환기 시스템을 개발했다. 상품화가 이뤄지기까지 개선이 필요한 점은 분명하나 작고 간편한 설치로 공기상태를 측정하고 조절하는 것은 장점이다.

| 설계 진행 내용               | 4월 | 5월 | 6월 | 7월 | 8월 | 9월 |
|------------------------|----|----|----|----|----|----|
| 설계, 재료 구매              | ■  |    |    |    |    |    |
| 3D도면 스케치, DMU & FEM 해석 |    | ■  |    |    |    |    |
| 1차 프린팅, 조립             |    |    | ■  |    |    |    |
| 아두이노 코드 개발, 제어         |    | ■  |    |    | ■  |    |
| 앱 인벤터 개발               |    |    | ■  | ■  |    |    |
| 시뮬레이션 및 피드백            |    |    | ■  | ■  |    |    |
| 도면 수정, 2차 프린팅          |    |    |    |    | ■  |    |
| 시스템 완성                 |    |    |    |    |    | ■  |

스마트 환기 시스템은 위와 같은 일정으로 제작되었으며, 제작비용은 다음과 같다.

| 항목                                 | 단가     | 개수 | 금액     |
|------------------------------------|--------|----|--------|
| 아두이노 우노 R3 정품                      | 22,500 | 2  | 45,000 |
| 아두이노우노R3호환보드                       | 13,200 | 2  | 26,400 |
| 온습도 센서(DHT11)                      | 2,100  | 2  | 4,200  |
| 미세먼지 센서 GP2Y1023AU0F               | 14,300 | 2  | 28,600 |
| 미세먼지센서 GP2Y1012AU0F                | 8,920  | 1  | 8,920  |
| 미세먼지센서 GP2Y1010AU                  | 5,500  | 1  | 5,500  |
| GP2Y10시리즈 미세먼지센서<br>전용케이블(MALE)    | 660    | 3  | 1,980  |
| 우노 메가 USB 전원 통신 케이블                | 220    | 3  | 660    |
| 16x2LCD모듈                          | 2,200  | 5  | 11,000 |
| 16x2LCD모듈with 어댑터                  | 2,970  | 3  | 8,910  |
| 40핀 커넥터M-M (Male -Male) 30         | 1,100  | 1  | 1,100  |
| 40핀 커넥터M-M (Male -Male) 20         | 990    | 2  | 1,980  |
| 40핀 커넥터M-M (Male -Male) 10         | 990    | 2  | 1,980  |
| 40핀 커넥터F-F (Female -Female) 20     | 990    | 1  | 990    |
| 40핀 커넥터F-F (Female -Female) 10     | 990    | 1  | 990    |
| 40핀 커넥터F-M / M-F (Female -Male) 20 | 990    | 1  | 990    |
| 40핀 커넥터F-M / M-F (Female -Male) 10 | 990    | 1  | 990    |
| 18650x 4리튬이온배터리홀더                  | 1,800  | 4  | 7,200  |
| 4xAA 배터리홀더                         | 1,210  | 4  | 4,840  |
| 6xAA 배터리홀더                         | 2,200  | 1  | 2,200  |
| 브레드보드                              | 880    | 5  | 4,400  |
| 아두이노3색신호등 LED모듈                    | 880    | 2  | 1,760  |

|             |                        |        |    |         |
|-------------|------------------------|--------|----|---------|
|             | 백셀AA 알카라인 건전지          | 880    | 15 | 13,200  |
|             | 공기질센서 모듈               | 2,200  | 2  | 4,400   |
|             | 5V 소형DC 모터             | 990    | 6  | 5,940   |
|             | 옥토퍼스5V DC 팬모터모듈        | 3,850  | 1  | 3,850   |
|             | MG946R호환서보모터           | 4,620  | 2  | 9,240   |
|             | MQ-7가스센서(일산화탄소)        | 1,540  | 2  | 3,080   |
|             | 5V 수동부저                | 220    | 2  | 440     |
|             | 2510쿨링팬(5V)            | 1,000  | 1  | 1,000   |
|             | 4010쿨링팬(5V)            | 990    | 1  | 990     |
|             | 8025쿨링팬(12V)           | 1,540  | 1  | 1,540   |
|             | HC06블루투스모듈             | 4,100  | 1  | 4,100   |
|             | 1채널 모스펫모듈(IRF520)      | 990    | 2  | 1,980   |
|             | L298N모터드라이버            | 2,200  | 2  | 4,400   |
|             | 3x2핀 전선연결 레버형커넥터       | 660    | 2  | 1,320   |
|             | 2x2핀 전선연결 레버형커넥터       | 550    | 2  | 1,100   |
| 아<br>크<br>릴 | 삼각줄대 500mm             | 500    | 4  | 2,000   |
|             | 손잡이                    | 600    | 2  | 1,200   |
|             | 시건장치                   | 800    | 1  | 800     |
|             | 경첩(중)                  | 600    | 4  | 2,400   |
|             | 재단칼                    | 3,000  | 2  | 6,000   |
|             | 투명 3T 394*400          | 9,000  | 4  | 36,000  |
|             | 투명 3T 394*394 (홀)      | 9,900  | 1  | 9,900   |
|             | 투명 3T 400*400          | 9,100  | 2  | 18,200  |
|             | 투명 3T 394*394          | 8,900  | 1  | 8,900   |
| 쿠<br>광      | 나무송곳 DIY도구             | 1,330  | 1  | 1,330   |
|             | 안개분무기(투명)160           | 4,900  | 1  | 4,900   |
|             | 안전거울 3mm, 동그라미, 100mm  | 3,000  | 1  | 3,000   |
|             | 아크릴 본드 대용량             | 12,000 | 1  | 12,000  |
|             | 1000 성냥개비              | 4,870  | 1  | 4,870   |
|             | 퓨리웨이 1kg PLA16 노랑 필라멘트 | 14,300 | 2  | 28,600  |
|             | 배송비1                   | 2,500  | 8  | 20,000  |
|             | 배송비2                   | 3,000  | 2  | 6,000   |
|             | 배송비3                   | 12,500 | 1  | 12,500  |
| 합계 금액       |                        |        |    | 405,770 |

## 참고문헌

- (1) 빌딩용 에어컨 실외기 주변의 유동 현상에 관한 연구
- (2) 지능형 스마트 홈네트워크를 위한 IoT기반 자동조절시스템 설계(심정연, 강남대학교 교양학부 컴퓨터학 전공)
- (3) An optimal design for axial-flow fan blade: theoretical and experimental studies(Cheng-Hung Huang and Chung-Wei Gau)
- (4) 센서를 이용한 자동 실내 온도 제어시스템 설계 및 구현(정규태 외 2, 제주대학교)
- (5) 기후변화 대응을 위한 RCP 시나리오 기반 국내 열지수 불쾌지수 예측(이수지 외 10, 고려대학교 의과대학 예방의학교실, 경북대학교 수학과)



- (6) 공동주택 세대 습기 제어를 위한 환기 시스템 운영 전략 효과 연구(곽병창 외 4, 한국토지주택공사 토지주택연구원 주택성능연구개발센터 책임연구원)
- (7) 공동주택의 필요환기량 산정방법(한화택, 국민대학교 기계자동차공학과)
- (8) 선풍기의 날개 수에 따른 공기 유동해석(조재웅 외 1, 한국기계항공학회지)
- (9) 아파트에서 환기장치와 공기청정기의 미세먼지 저감 실증 사례(노광철, 광주공기산업 전문위원)
- (10) 자연환기 및 공기청정기 가동에 따른 실내공기질 분석(송수원 외 3, 한국건설기술연구원)
- (11) 피치각과 날개 길이에 따른 축류팬의 성능(전성택 외1, 인천대학교 기계시스템공학부)