

팀명	13조		
개발 시스템명	미니 아두이노 공기청정기		
팀원	구명희	김선재	김재부

## 1. 시스템 개요 및 필요성

### 1) 미니 아두이노 공기 청정기란?

최근 몇년간 중국에서 날아오는 미세먼지와 기타 요인에 의해 미세먼지의 농도가 급격히 늘어나면서 공기청정기에 대한 수요가 크게 늘어나고 있습니다. 이런 유행에 따라 저희 팀에서는 기말고사 프로젝트로 미니 아두이노 공기청정기를 만들어보기로 했습니다. 아두이노 미니 공기청정기라고 이름을 붙인 이유는 제품에 사용되는 팬의 크기와 보드의 크기가 작기 때문에 전체적인 모양도 작아지게 되었기 때문에, 저희는 이 제품을 미니 이두이노 공기청정기라고 이름을 붙였습니다.

### 2) 필요성

- 올해 미세먼지가 유독 심해짐에 따라 외부활동을 자제하고 마스크를 사용하거나 공기청정기를 사용하는 등 우리는 미세먼지의 피해를 줄이기 위한 대책을 강구해야 하는 동시에, 개인의 노력이 필요합니다.
- 가장 간단한 방법으로는 마스크를 착용하는 것입니다. 외부에서는 마스크를 착용하고 활동할 수 있지만, 실내에서까지 지속적으로 마스크를 착용하기에는 한계가 있습니다.
- 공기청정기는 사용하기 간편하지만 가격면에서 봤을 때 선뜻 구입하기 꺼려지며, 이에, 저희 조는 수업시간에 사용한 센서들을 기반으로 간단히 제작할 수 있는 미니 아두이노 공기청정기를 제작했습니다.

### 3) 13팀에서 생각하는 미니 아두이노 공기청정기와 타 제품과의 차별점

- 제품을 아두이노로 제작함으로써 대체로 공기청정기를 따로 구입하는 비용을 줄일 수 있으며, 본인이 원한다면 공기청정기의 순기능 외에도 여러 가지 다른 기능들을 추가해서 자신만의 미니 아두이노 공기 청정기를 개발해서 사용 할 수 있습니다.

### 4) 제품의 개발 내용

- (1) 소프트웨어 시리얼을 이용한 유선통신과 블루투스 모듈을 이용한 무선통신을 구현합니다.
- (2) 팬의 동작을 제어합니다.
- (3) 제품의 동작상태를 표시합니다.
- (4) 디지털 시계를 구현합니다.
- (5) 제품이 동작중일 때, 미세먼지의 수치를 출력합니다.

## 2. 계획대비 변경내용

### - 팬 조절 문제가 있었습니다.

기존의 계획으로는 가변저항을 사용해 팬의 RPM을 조절하려고 했으나, 전력량을 줄이면 동작을 멈추거나, 릴레이 모듈에서 소음이 발생하는 문제가 있었습니다. 이를 해결하기 위해 RPM을 조절하는 항목을 삭제했습니다. 이와 비슷한 효과를 구현하기 위해 2개의 팬 중에서 1개의 팬만 OFF하여 RPM조절과 비슷한 효과를 낼 수 있도록 수정했습니다.

### - 쿨링 팬 제어 문제가 있었습니다.

5v의 팬을 다른 장치없이 보드에 연결하여 사용하려고 했으나, 디지털 신호와 전력을 동시에 전달할 수 없는 문제가 생겼습니다. 이를 해결하기 위해 릴레이 모듈을 추가해서 팬의 디지털 신호값을 전송해 팬의 온오프가 원활히 작동되도록 구현했습니다.

### - 추가적으로 기존 계획에 없었던 HW STC 모듈인 DS1302를 추가했습니다.

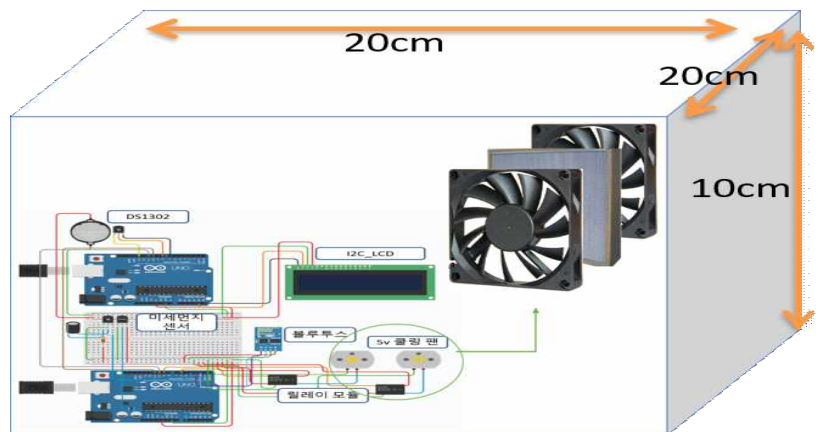
처음 계획으로는 LCD에 미세먼지 수치를 출력하면서 위에 시간 또는 바이트 문자를 사용해 미세먼지 상태에 따라 바뀌는 표정 등의 출력을 할 수 있도록 생각하고 있었습니다. 프로젝트를 만들면서 키트에 있는 RTC모듈을 사용해서 만드는 시계의 구현을 알게 되었고, 전원이 꺼져도 시간이 실시간으로 동기화되는 디지털 시계를 추가로 구현했습니다.

## 3. 최종 시스템 구성 및 기능

### 1) 제품 구상도 및 회로도

#### - 제품 구성

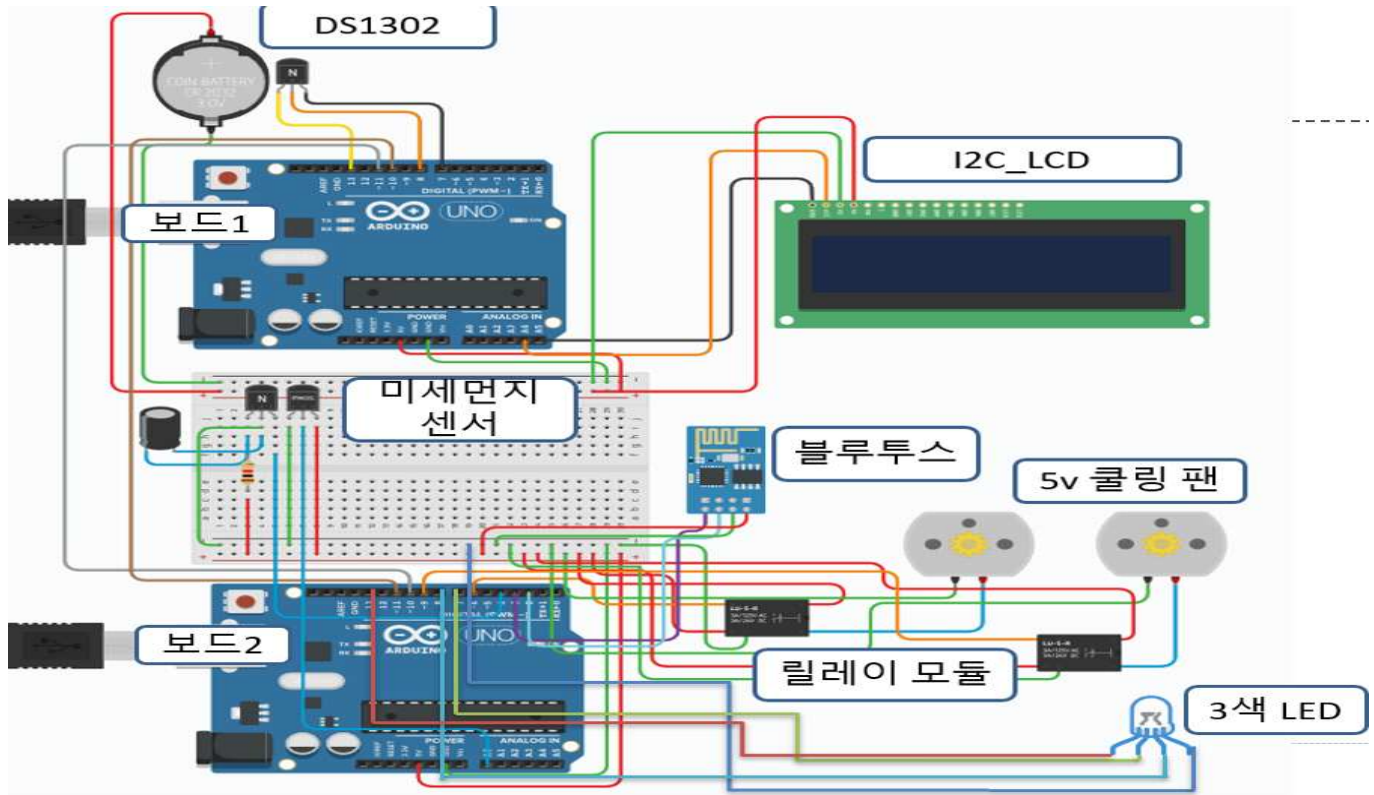
- (01) 20cm x 20cm x 10cm상자
- (02) 5v 70mm 쿨링팬 2개
- (03) 아두이노 우노 2개
- (04) 3색 LED 1개
- (05) I2C\_LCD 1개
- (06) 블루투스 모듈 1개
- (07) 미세먼지 센서 1개
- (08) 릴레이모듈 2개
- (09) DS 1302 모듈 1개



#### - 구조

- (10) 상자속에 쿨링팬과 헤파필터를 샌드위치 방식으로 구성했습니다.
- (11) 첫 번째 보드와 두 번째 보드는 서로 유선으로 연결되어 있습니다.
- (12) 회로도에서 연결한 센서중에서 모양이 다르게 있습니다.  
틀에 블루투스 모듈, 미세먼지 센서, 팬, 릴레이 모듈이 없어 비슷한 모양의 부품으로 대신했습니다.

## - 회로도



### 1) 보드1

- (1) I2C\_LCD를 연결했습니다.
- (2) DS1302 모듈을 연결했습니다.
- (3) 보드2와 연결했습니다.

### 2) 보드2

- (4) 5v 쿨링팬을 연결했습니다.
- (5) 릴레이 모듈을 연결했습니다.
- (6) 미세먼지 센서를 연결했습니다.
- (7) 블루투스 모듈을 연결했습니다.
- (8) 3색 LED를 연결했습니다.
- (9) 보드1과 연결했습니다.

### 3) 연결

저희 조의 미니 아두이노 공기청정기는 작은 브레드 보드 하나에 2개의 아두이노 보드를 연결했습니다.

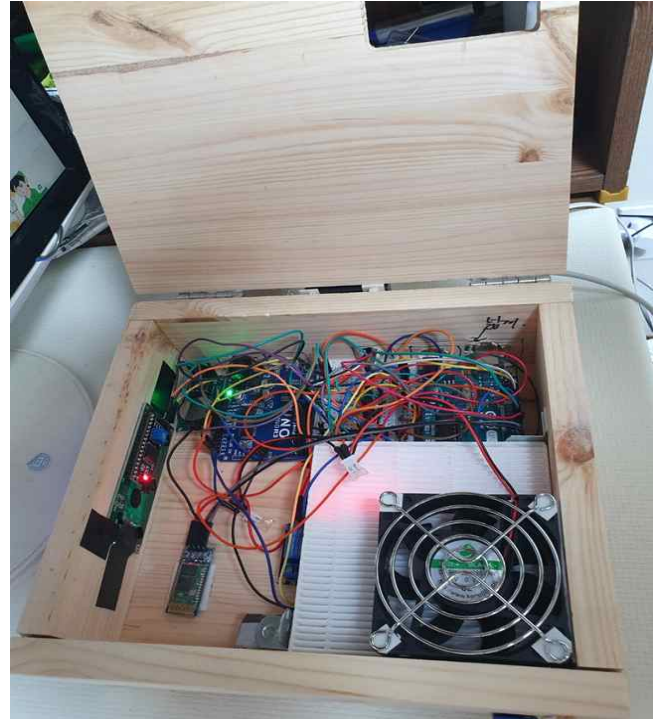
- (1) 모니터가 되어 줄 I2C\_LCD는 VCC - 5v, GND - gnd, SDL - A4, SCL - A5에 각각 연결되어 있습니다.
- (2) 디지털 시계의 역할을 수행해줄 DS1302 모듈은 D7, D8, D13에 연결되어있습니다.
- (3), (9) 보드1과 보드2는 각각 D10, D11핀이 교차로 연결되어 서로 소프트웨어 통신을 합니다.
- (4) 쿨링팬은 D6, D9에 각각 연결되어 있습니다.
- (5) 팬을 제어할 릴레이 모듈은 팬과 연결되어 있습니다.
- (6) 미세먼지 센서는 6개의 핀이 있습니다. 왼쪽부터 1번핀이라고 했을 때,  
1번 - 커패시터 +극과 5v, 2번 - 커패시터 -극과 gnd,  
3번 - D4, 4번 - gnd, 5번 - A0, 6번 - 5v에 각각 연결되어 있습니다.
- (7) 팬을 제어할 블루투스 모듈은 VCC - 5v, GND - gnd, RXD - D2, TXD - D3에 각각 연결되어 있습니다.
- (8) 미세먼지의 심각도를 알려줄 3색LED는 G - D7, B - D8, R - D13, GND - gnd에 각각 연결되어 있습니다.

## 2) 최종 시스템 구현 모습

### - 케이스 상단에서 내려다봤을때 모습입니다.

경첩을 사용해서 문을 여닫아 내부 관찰이 수월하도록 했습니다.

주로 부품을 수리 또는 교체할 때 사용할 부분입니다.



### - 케이스 옆면에서 봤을 때 모습입니다.

경첩을 사용해서 문을 여닫아 내부 관찰이 수월하도록 했습니다.

추가적으로, 손잡이와 걸쇠를 제작해서 문의 고정과 여닫음을 개선했습니다.

주로 필터를 교체할 때 사용할 부분입니다.

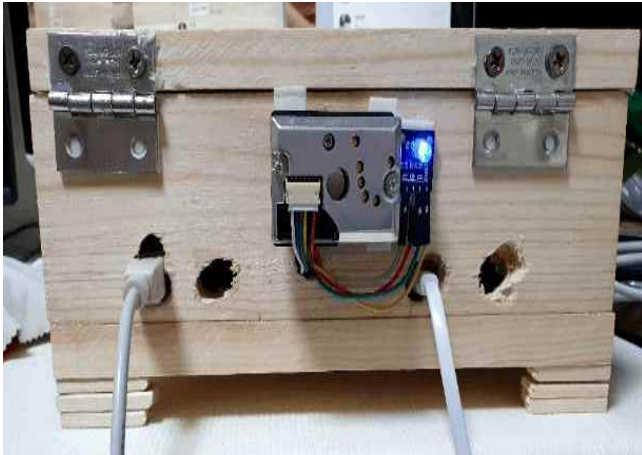




- 케이스 옆면에서 봤을 때 모습입니다.

한쪽 면에는 미세먼지 센서와 3색 LED를, 다른쪽 면에는 I2C\_LED를 부착했습니다.

원래 한 면에 전부 해결하고 싶었으나, 사정상 구멍을 뚫을 수 없어서 이와 같이 배치했습니다.



- 케이스 하단에서 봤을 때 모습입니다.

2개의 팬이 흡기와 배기로 작동하므로 공기가 들어가고 빠져나가야 할 공간이 필요했습니다.

그래서 하단의 네 귀퉁이에 다리를 제작함으로 공기의 흐름이 원활하도록 제작하였습니다.

미세먼지 센서 또한 센서의 특성상 공기의 흐름이 원활해야 합니다.

미세먼지 센서 부분을 내려다보면 역시 나무젓가락으로 다리를 제작해주었습니다.



## 4. 기능별 동작화면

사진 촬영이 용이하도록 각각의 미세먼지 수치를 조정해서 촬영했습니다. 조정한 수치는  $\pm 20$  정도입니다. 추가적으로 미세먼지 농도가 91 이상일 때 색의 랜덤 변화는 촬영하지 않았습니다.

- 미세먼지 수치가 0~30일 때 초록색입니다.

```
if(dust_average > 0 && dust_average < 30) // 초록
{ digitalWrite(color[0], 1); LEDreset(); }
```



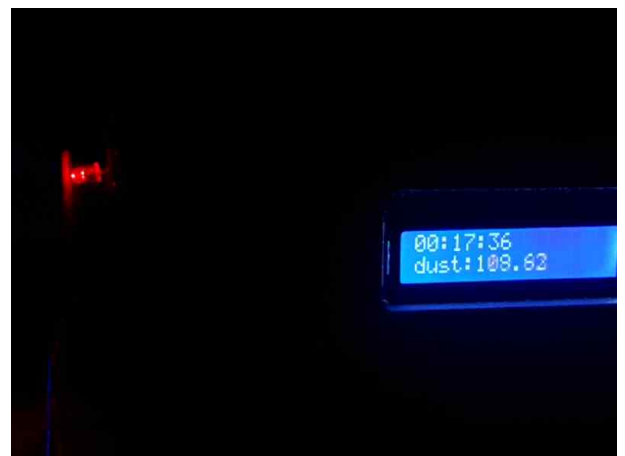
- 미세먼지 수치가 31~60일 때 파란색입니다.

```
else if(dust_average > 31 && dust_average < 60) // 파랑
{ digitalWrite(color[1], 1); LEDreset(); }
```



- 미세먼지 수치가 61~90일 때 빨간색입니다.

```
else if(dust_average > 61 && dust_average < 90) // 빨강
{ digitalWrite(color[2], 1); LEDreset(); }
```



- 사진에 보이는 시간과 미세먼지 농도가 잘 출력되고 있음을 알 수 있습니다.
- 블루투스를 이용해서 팬 조절하는 기능은 유튜브 영상으로 대체하였습니다.

## 5. 문제점 및 해결방법

----- 2. 계획대비 변경내용과 겹치는 내용의 설명은 추가하지 않았습니다. -----

- 팬 조절 문제가 있습니다. - 쿨링 팬 제어 문제가 있습니다.

- 시리얼 통신으로 미세먼지 센서의 값을 LCD에 출력하는 문제가 있습니다.

채팅 프로그램의 응용을 통해 센서의 값을 바로 읽어서 시리얼 모니터에 출력하려고 했으나, 왼쪽 사진과 같이 뒤의 숫자가 초기화되지 않고 계속 따라붙는 문제가 있었습니다.

이 문제는 시리얼 모니터에서 읽어온 문자가 `wn`, 즉 널 문자가 들어있는 `Msg`의 인덱스를 찾아서 그 인덱스까지 출력하게 했습니다. 그 결과, 오른쪽 사진과 같이 정상적으로 출력됨을 확인했습니다.

```
new_Msg[a] : Msg[a] : .
new_Msg[a] : Msg[a] : 5
new_Msg[a] : Msg[a] : 0
account : 6 RMsg.length() : 8 new_Msg : 14.60
new_Msg[a] : 1 Msg[a] : 1
new_Msg[a] : 4 Msg[a] : 0
new_Msg[a] : . Msg[a] : 0
new_Msg[a] : 6 Msg[a] : .
new_Msg[a] : 0 Msg[a] : 5
new_Msg[a] : Msg[a] : 7
account : 7 RMsg.length() : 9 new_Msg : 100.57
new_Msg[a] : 1 Msg[a] : 1
new_Msg[a] : 0 Msg[a] : 2
new_Msg[a] : 0 Msg[a] : .
new_Msg[a] : . Msg[a] : 5
new_Msg[a] : 5 Msg[a] : 2
account : 6 RMsg.length() : 8 new_Msg : 12.527
new_Msg[a] : 1 Msg[a] : 1
new_Msg[a] : 2 Msg[a] : 0
new_Msg[a] : . Msg[a] : 0
new_Msg[a] : 5 Msg[a] : .
new_Msg[a] : 2 Msg[a] : 5
new_Msg[a] : 7 Msg[a] : 7
account : 7 RMsg.length() : 9 new_Msg : 100.57
new_Msg[a] : 1 Msg[a] : 1
new_Msg[a] : 0 Msg[a] : 1
new_Msg[a] : 0 Msg[a] : .
new_Msg[a] : . Msg[a] : 5
new_Msg[a] : 5 Msg[a] : 4
account : 6 RMsg.length() : 8 new_Msg : 11.547
new_Msg[a] : 1 Msg[a] : 1
```

->

```
new_Msg[a] : Msg[a] : 1
new_Msg[a] : Msg[a] : 8
new_Msg[a] : Msg[a] : .
new_Msg[a] : Msg[a] : 2
new_Msg[a] : Msg[a] : 2
account : 6 RMsg.length() : 8 new_Msg : 18.22
-----
new_Msg[a] :
Msg[a] : 1
new_Msg[a] :
Msg[a] : 0
new_Msg[a] :
Msg[a] : 0
new_Msg[a] :
Msg[a] : .
new_Msg[a] :
Msg[a] : 5
new_Msg[a] :
Msg[a] : 7
account : 7 RMsg.length() : 9 new_Msg : 100.57
-----
new_Msg[a] :
Msg[a] : 1
new_Msg[a] :
Msg[a] : 2
new_Msg[a] :
Msg[a] : .
new_Msg[a] :
Msg[a] : 2
new_Msg[a] :
Msg[a] : 0
account : 6 RMsg.length() : 8 new_Msg : 12.20
```

하지만 여전히 왼쪽 사진과 같이 LCD화면엔 숫자가 초기화되지 않았습니다.

주말내내 여러 가지 검증을 통해 시리얼 모니터에 출력해보아도 해결할 수 없었습니다.

다행히 마지막으로 교수님에게 질문했던 날, 혹시나 했던 방법으로 문제를 해결할 수 있었는데, LCD의 같은 자리에 문자 2글자와 3글자를 각각 반복해서 출력해보았고, 그제서야 이 문제는 LCD를 출력하는 코딩 부분이라는 것을 알았습니다.

이 문제는 미세먼지 센서의 출력값이  $500\mu\text{g}/\text{m}^3$  이하이고 소수점 둘째 자리까지만 출력되므로, 센서의 값이 한 자리(x.xx), 두 자리(xx.xx), 세 자리(xxx.xx) 3가지 값을 가질 경우에 대해 위에서 구한 인덱스 값을 사용해서 각각의 소수점 위치를 구했습니다.

그 결과, 오른쪽 사진과 같이 정상적으로 출력됨을 확인했습니다.



->



## 6. 최종 구성요소

No.	Items	Purpose of use
-	<b>[센서 및 모듈]</b>	—
1	3색 LED	미세먼지 농도에 따라 색이 다르게 출력됩니다.
2	I2C_LCD	미세먼지의 수치를 표시합니다.
3	미세먼지 센서	주변 미세먼지 농도를 측정합니다.
4	릴레이 모듈	팬에 제어신호와 전류를 보낼 수 있도록 하는 모듈
5	블루투스 모듈	핸드폰을 이용해 무선으로 제품을 작동시킵니다.
6	DS1302 모듈	전원이 없어도 시간이 계속 흐름으로써, 디지털 시계의 역할을 합니다.
-	<b>[통신]</b>	—
-	유선통신	측정한 미세먼지 농도를 공기청정기 쪽으로 전송합니다.
-	블루투스 무선통신	블루투스 모듈을 사용해서 안드로이드 핸드폰과 무선 통신을 합니다.
-	<b>[그 외 부품]</b>	—
-	헤파필터	대기중의 미세먼지를 걸러주는 필터입니다.
7	5v 70mm 쿨링팬	대기중의 공기를 필터를 거쳐 흡입과 방출을 하게 합니다.



## 7. 개인별 구현 내용

이름	개발내용	기여도
구명회	1) 시리얼 통신을 통해 읽어온 미세먼지 센서의 값을 Lcd에 출력하는 코드를 구현했습니다. 2) 코드 전반 주석문을 작성했습니다. 3) Hwp, ppt 등 보고서 및 발표자료를 작성했습니다.	50%
김선재	1) 쿨링 팬 제어 및 블루투스 통신을 구현했습니다. 2) HW STC방식의 디지털 시계를 구현했습니다. 3) 최종 조립 및 테스트를 맡았습니다. 4) 최종 발표를 맡았습니다.	40%
김재부	1) 3색 LED 동작을 구현했습니다. 2) 제품을 담은 케이스를 선정하고 구매했습니다.	10%

## 8. 오픈소스HW 기말고사 팀 프로젝트를 마치고 개인별 소감문입니다.

### 구명회

왠지 IoT의 길에 눈을 뜬 것 같습니다! 컴퓨터 공학과에 와서 손으로 직접 뭔가를 만들어보는건 처음이라서 과제를 받았을 때 잘 만들수 있을지 불안했었는데, 막상 다 만들어놓고 나니까 의외로 별 것 아니어서 이런 작품을 다음에도 만들어 볼 기회가 있었으면 좋겠습니다.

### 김선재

기존의 실습과는 다르게 실시간 결과를 눈앞에서 볼 수 있었으며 컴퓨터로만 하는것이 아닌 외부기기를 사용하여 학습하는게 다른과목들 대비 다루는 재미가 있었습니다.

### 김재부

배운 내용을 직접 활용해보는 좋은 시간이었습니다. 수업에서 배운것은 작은 부분 부분이었지만 그것을 서로 연동하고 조합하여 실제로 사용가능한 하나의 체계를 만들어가는 모습이 즐거웠습니다.