Space Conditioner

{capstone design}

1조

김준식 · 남중훈 · 박보영 · 박조은 · 홍혜림

01 Research background

02 Content of study

03 Progress

04 Final model

05 Significance limitations

Research Background

Content of study

Progress

Final model

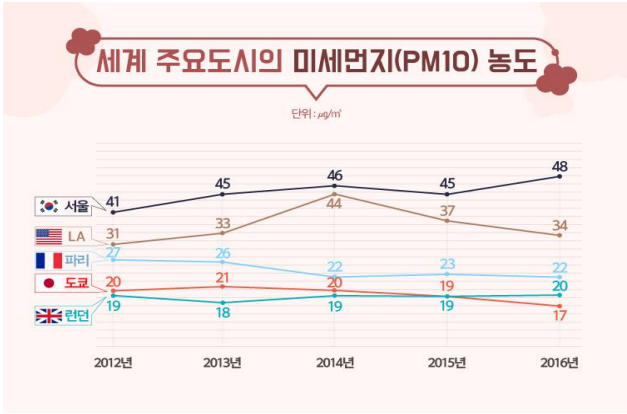
Significance/ limitations

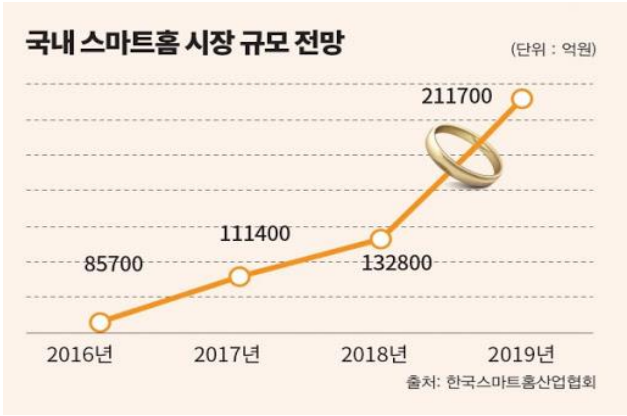
Research Background

- The average fine dust level in Seoul is more than doubled compared to other countries.

→ Fine dust in our country gets worse day by day

- The size of South Korea's smart home market is growing
- Increasing demand for Smart Home-related products and services

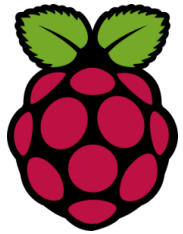




"A system fine dust/temperature/humidity in the room is required"

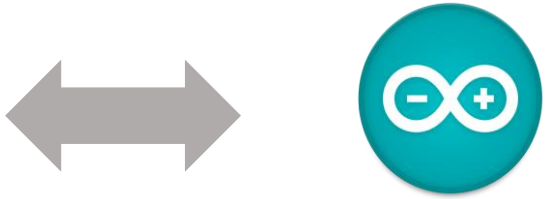
Project Topic

"Space Conditioner according to Real-time Weather Data"



Computer

- Communication with outside
- Real-time Weather Data Processing (fine dust)



Control Tower

- Monitoring Current State
- Make a Decision
- The Hub

Home
(with Display Interface)

- Window Opening / Closing
- Indoor Condition Sensing

As a Capstone Project

Computer Engineering

- System Design with Raspberrypi
  → Automatic Space Condition Control
- Further Research (Application..)

Electronic & Electrical Engineering (HW)

- Hardware Component Design with Arduino
  → Sensors, home appliances
  → Prototyping

Industrial Engineering

- Real-time fine dust Data Processing
  → Data Analysis, Data Science
- Project Managing
  → Data Visualization

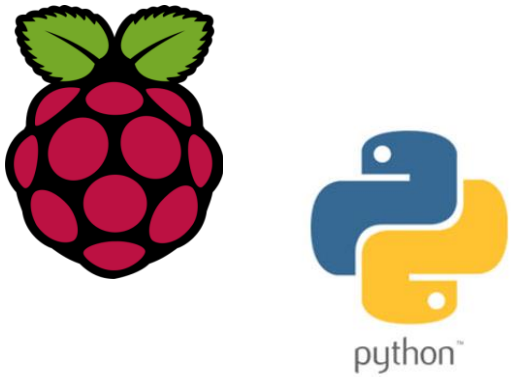
Electronic & Electrical Engineering (SW)

- Interfacing between HW & SW
  → Raspberrypi - Arduino
- Signal Processing

Content of study | Smart window

- Producing a smart window that opens and closes according to real-time fine dust concentration

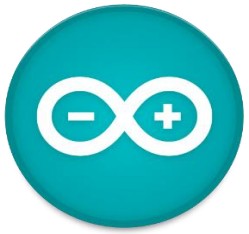
- Using real-time fine dust data provided by the public data portal



- Coding data with Python using Raspberrypi

- Checked data in Arduino and operated servo-motor to open and close windows

Content of study | Indoor control



Arduino

- Simplified IDE
- Various OS environment
- Cheap price
- Many open sources

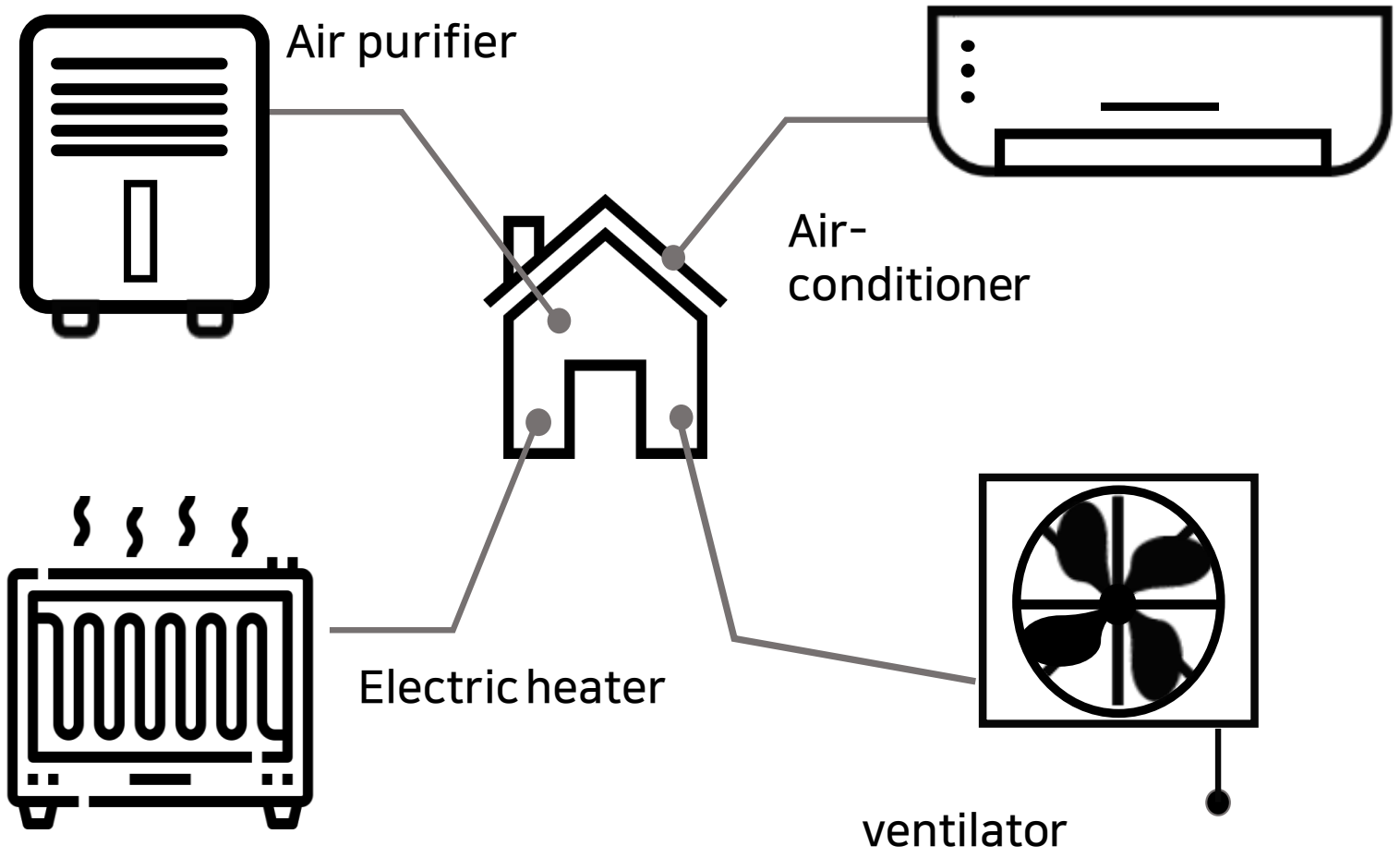
Sensor

- Quick results check
- Easy usability
- DHT11

Humidity

Temperature

Content of study | Indoor control



Total model

Research Background

Content of study

Progress

Final model

Significance/ limitations

	좋음	보통	나쁨	매우나쁨
미세먼지	0~30	31~80	81~150	150 이상
초미세먼지	0~15	16~50	51~100	101 이상

	매우 높은	높음	낮음
온도	32도 이상	28~32	10도 이하
습도	80% 이상	60~80	60% 이하

	매우높음	높음	보통	낮음
불쾌지수	80 이상	75~80	68~75	68 미만

Smart window

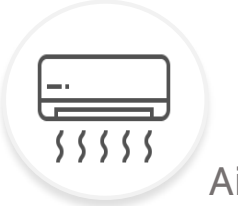
Air purifier

Display

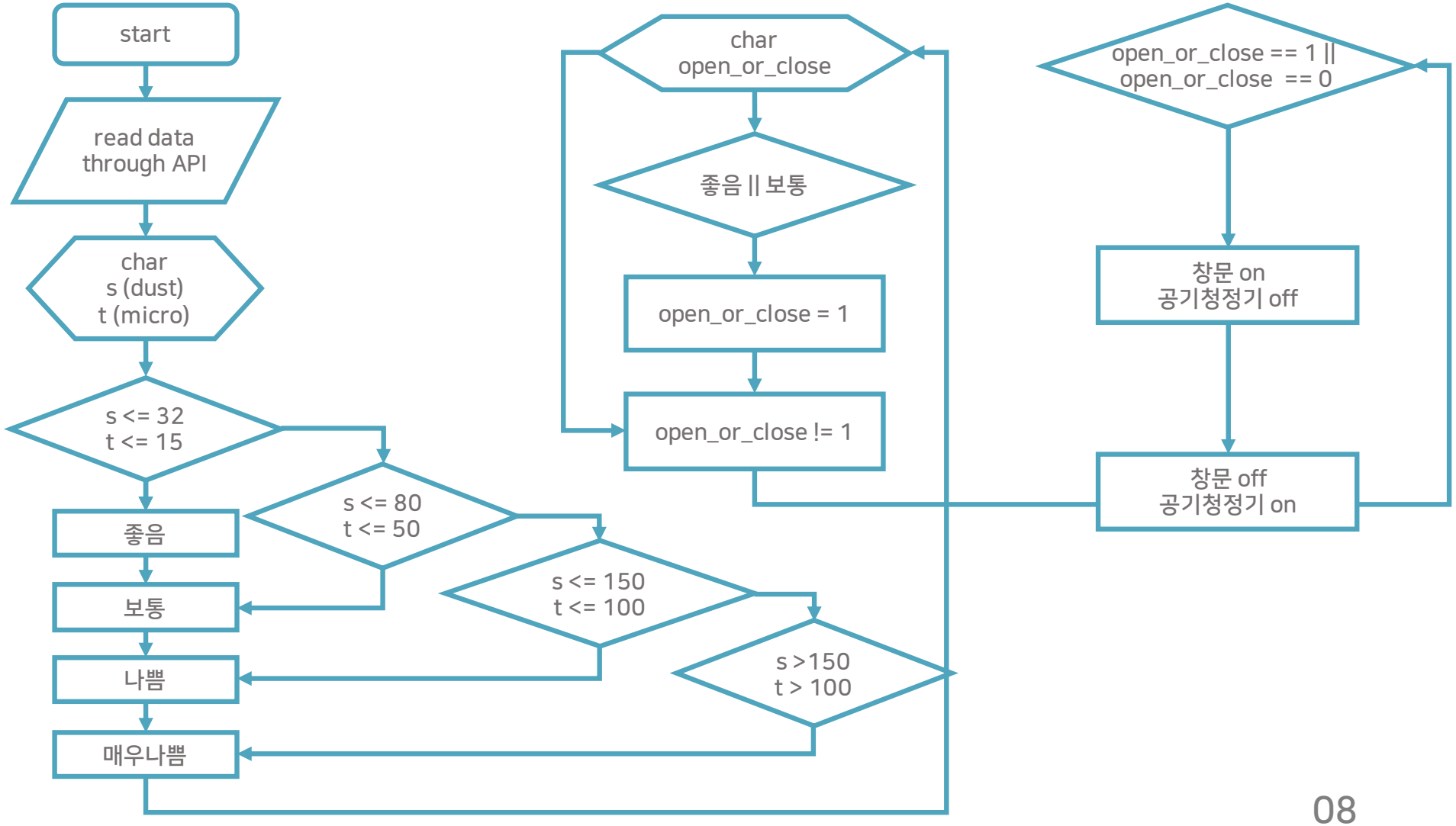
Electric heater

ventilator

Display



Air conditioner



```
ser = serial.Serial("/dev/ttyACM0" ,9600)
```
→ start – 값을 넘겨줄 아두이노의 port number

```
M = '&numOfRows=1&pageNo=1&stationName=마포구&dataTerm=DAILY&ver=1.3'
key =
'baHQwL7PU9aOAl5VEMy8uEvanZj%2F9NkUstZkCki%2FJDskKLQymF%2FRLMEBg2KXwCAKxQ6BkMVE8nFBpMHeilnNMQ%3D%3D'
url =
'http://openapi.airkorea.or.kr/openapi/services/rest/ArpltnInforInqireSvc/getMsrstnAcctoRltmMesureDnsty
?serviceKey=' + key + M

response = requests.get(url)
soup = BeautifulSoup(response.text, "html.parser")
ItemList = soup.findAll('item')
for item in ItemList:
    a = item.find('datatime').text
    g = item.find('pm10value').text
    i = item.find('pm25value').text
    s = item.find('pm10grade1h').text
    t = item.find('pm25grade1h').text
    print('측정소: 마포구')
    print('측정시간:' + a)
    print('미세먼지 농도:' + g + 'μg/㎥ ( ' + s + ' )')
    print('초미세먼지 농도:' + i + 'μg/㎥ ( ' + t + ' )')
    print('( 좋음: 1 ),( 보통: 2 ),( 나쁨: 3 ),( 매우나쁨: 4)' )
```
→ BeautifulSoup 이용하여 text 형태로 미세먼지 data 추출

```
open_or_close = '0'
if (s <= '2' and t <= '2'):
    open_or_close = '1'
else:
    open_or_close = '3'
```
→ open_or_close는 창문을 on off 여부를 결정하는 변수
→ open_or_close가 1이면 on, 1이 아니면 off
→ 미세먼지 상태가 좋음 또는 보통일 경우 open_or_close = '1'
→ 미세먼지 상태가 나쁨 또는 매우나쁨일 경우 open_or_close != '1'

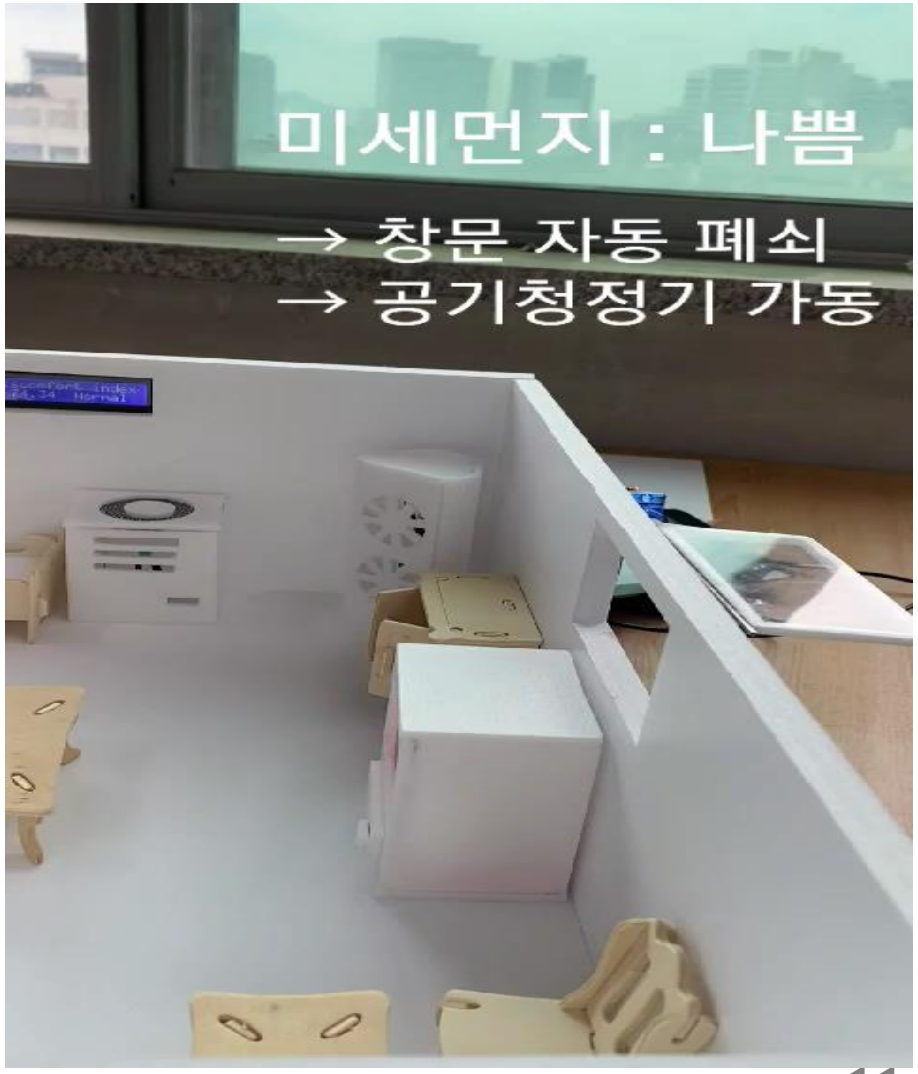
```
ser.write(open_or_close.encode())
```
→ 아두이노에 open_or_close 값을 넘겨줌

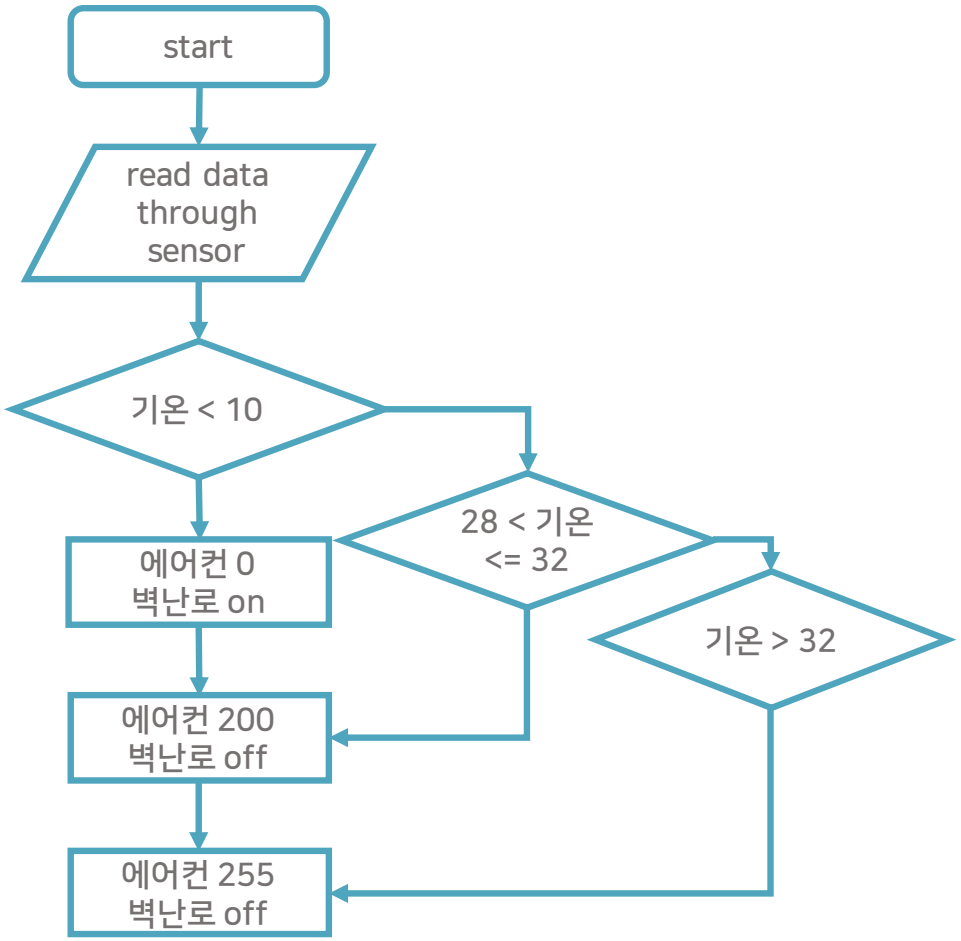
```
  pinMode(led_air,OUTPUT);
  pinMode(led_Tmotor,OUTPUT);
  pinMode(led_heater,OUTPUT);

}

void servofunc() {
    char in_data;
    in_data = Serial.read();

    if(in_data == '1')
    {
      angle = 90;
      digitalWrite(led_air,LOW);

    }
    else
    {
      angle = 0;
      digitalWrite(led_air,HIGH);
    }
    servo.write(angle);
}
```

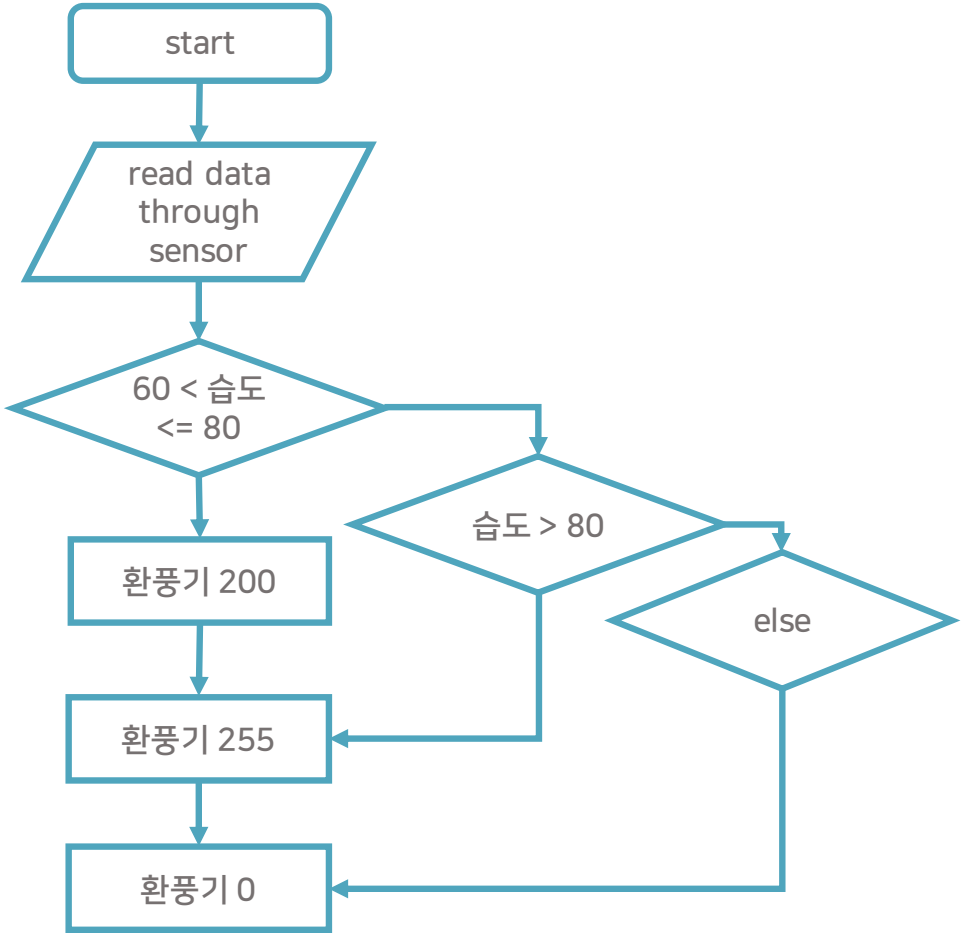
→ serial port를 통해 라즈베리파이로 부터 받은 값을 in_data로 저장

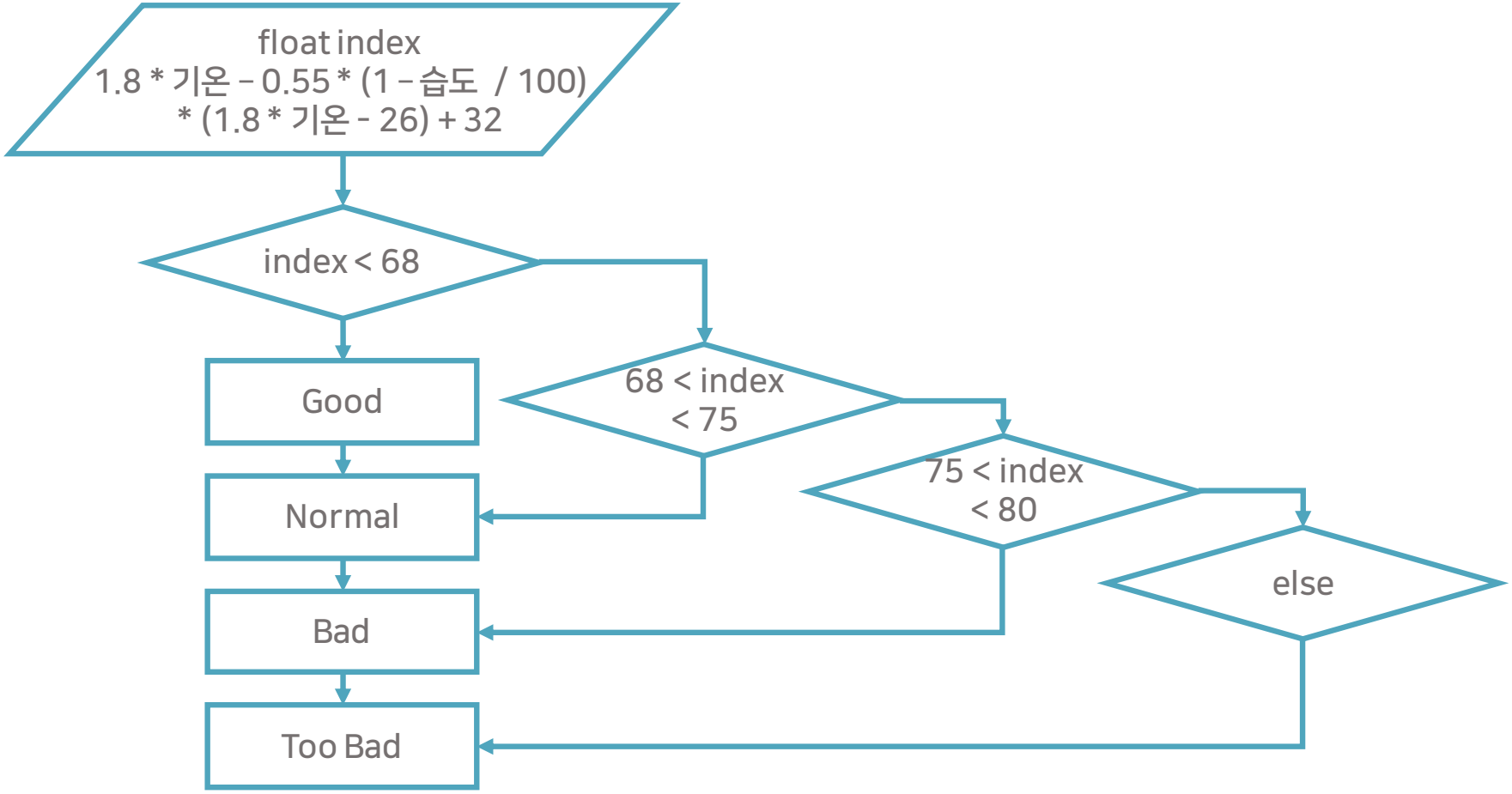
→ open_or_close가 1이면
서보모터 90° → 창문 열기
led off → 공기청정기 끄기

→ open_or_close가 1이 아니면
서보모터 0° → 창문 닫기
led on → 공기청정기 켜기









```
if (Temp > 30 & Temp <= 32)
  {
   analogWrite(Tmotor,200);
   digitalWrite(led_Tmotor,HIGH);
   digitalWrite(led_heater,LOW);
  }
 else if (Temp > 32)
  {
   analogWrite(Tmotor,255);
   digitalWrite(led_Tmotor,HIGH);
   digitalWrite(led_heater,LOW);
  }
 else if (Temp < 10)
 {
   digitalWrite(led_Tmotor,LOW);
   digitalWrite(led_heater,HIGH);
 }

 else
  {
   analogWrite(Tmotor,0);
   digitalWrite(led_Tmotor,LOW);
   digitalWrite(led_heater,LOW);
  }
```

→ 기온이 28℃ ~ 32 ℃일 때 에어컨(Tmotor) 작동

→ 기온이 32 ℃ 초과일 때 에어컨을 더 세게 작동 →모터에 더 높은 전압을 입력

→ 기온이 10 ℃ 미만일 때 전기난로 작동

→ 적정 온도

```
if (Humi >60 & Humi <= 80)
{
 analogWrite(Hmotor,200);
}
else if (Humi > 80)
{
 analogWrite(Hmotor,255);
}
else
{
 analogWrite(Hmotor,0);
}
}
```

→ 습도가 60%~ 80%일 때 환풍기(Hmotor) 작동

→ 습도가 80% 초과일 때 환풍기를 더 세게 작동

→ 적정 습도

"각 가전제품의 작동 여부는 LED로 확인 가능"

```
void loop(){

  if((err=dht11.read(Humi, Temp))==0)
  {
    lcd.setCursor(0,0);
    lcd.print("Temp : ");
    lcd.print(Temp);
    lcd.print(" C  ");

    lcd.setCursor(0,1);
    lcd.print("Humi : ");
    lcd.print(Humi);
    lcd.print(" %  ");
  }
  else
  {
    lcd.setCursor(0,0);
    lcd.print("ERROR NO.: dht11");
    lcd.setCursor(0,1);
    lcd.print("                ");
  }
```

→ LCD 패널에 센서가 측정한 온/습도 표시

```
  int i;
  for(i=0; i < 30; i++)
  {
   servofunc();
   delay(100);
  }
```

→ 센서 측정 주기보다 큰 주기로 자동 제어 실행을 위한 delay

```
index = 1.8*Temp-0.55*(1-Humi/100)*(1.8*Temp-26)+32;
```

→ 불쾌지수 계산

```
lcd.setCursor(0,0);
lcd.print("Discomfort index");

lcd.setCursor(0,1);
lcd.print(": ");
lcd.print(index);
if(index < 68)
{
  lcd.setCursor(7,1);
  lcd.print("   Good  ");
}
else if(index > 68 && index < 75)
{
  lcd.setCursor(7,1);
  lcd.print("  Normal ");
}
else if(index >= 75 && index <80)
{
  lcd.setCursor(7,1);
  lcd.print("   Bad   ");
}
else
  {
    lcd.setCursor(7,1);
    lcd.print(" Too Bad ");
  }


for(i=0; i < 30; i++)
{
 servofunc();
 delay(100);
}
```

→ LCD 패널에 불쾌지수 값과 등급 표시

Research Background

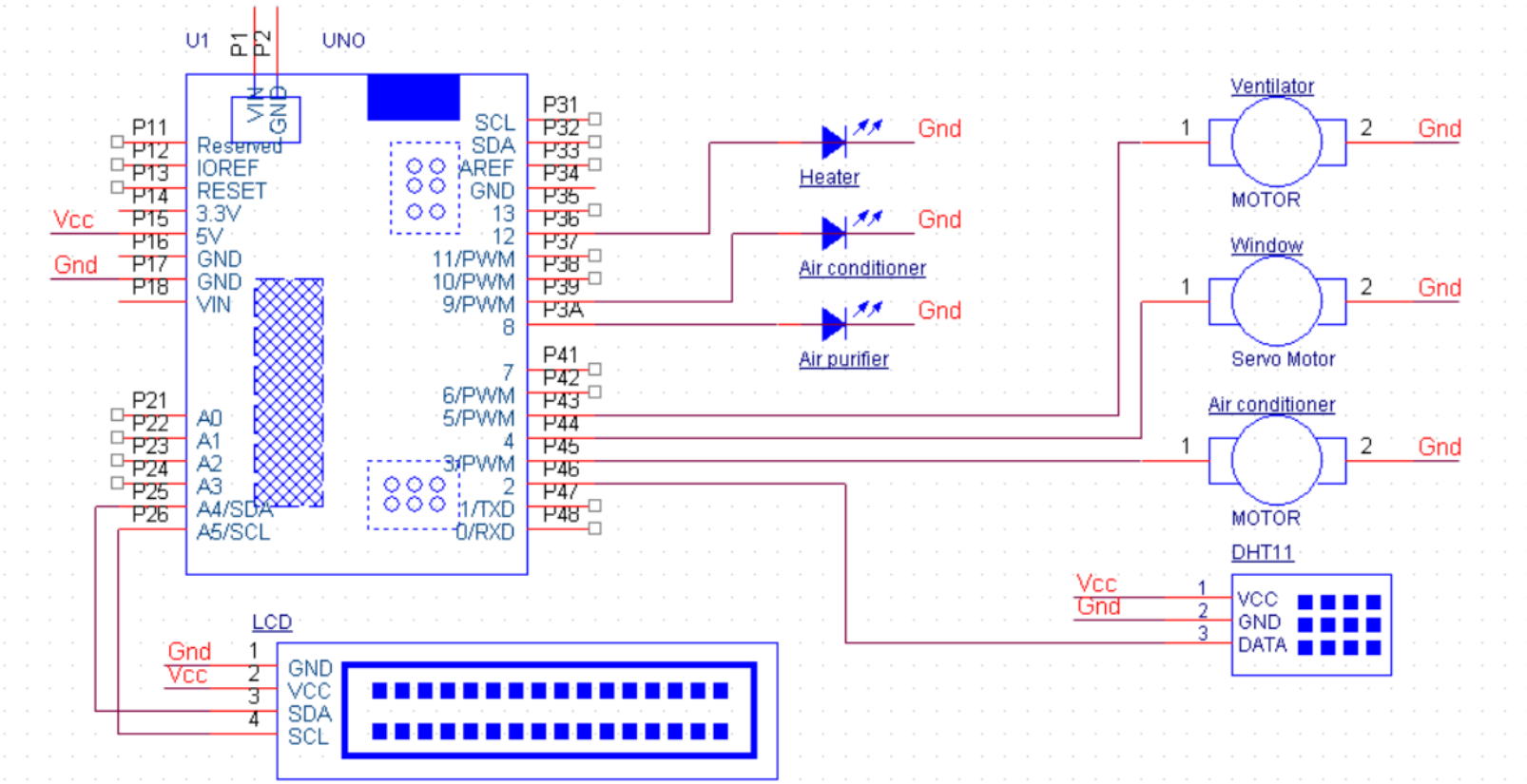
Content of study

Progress

Final model

Significance/ limitations

Indoor control | Circuit diagram



Significance/ limitations

Significance

- The rise of environmental problems & wellness trend
  → Market size & growth
- Potential High consumer interest

Limitations

- Fine dust data is not real-time because it does not use sensors and receives data values from monitoring stations.

Further research

- Create Application → Customization, Remote Control
- Building an iot environment with actual home appliances → wifi

Thank you