

<https://doi.org/10.7236/JIIBC.2021.21.1.121>
JIIBC 2021-1-17

IoT 센서를 활용한 미세먼지 분석 기술 개발

Development of Fine Dust Analysis Technology using IoT Sensor

신동진*, 이진**, 허민희**, 황승연***, 이용수****, 김정준*****

Dong-Jin Shin*, Jin Lee**, Min-Hui Heo**, Seung-Yeon Hwang***,
Yong-Soo Lee****, Jeong-Joon Kim*****

요약 최근 중국에서 발생하는 황사와 더불어 국내에서는 뉴스 등 미디어를 통해 미세먼지가 큰 화제가 되고 있다. 외부에서 발생하는 미세먼지도 있지만, 외부 미세먼지가 내부로 유입되면서 공기 청정기 제품의 구입율이 높아지고 있다. 공기 청정기는 내부에 필터를 사용하고 있으며, 센서를 통해 필터의 교체 유무를 LED 알람을 통해 사용자에게 알려 준다. 하지만, 현재 필터율이 얼마나 감소하였는지 측정하고, 작동하는 송풍기의 압력을 결정해주는 제품은 없는 상태이다. 따라서 본 논문에서는 IoT 기기인 아두이노와 미세먼지 센서, 차압 센서를 활용하여 데이터를 직접 생성하고, 측정된 미세먼지와 압력 값을 계산하여 필터율에 따라 필터가 얼마나 노화되었는지 확인할 수 있는 프로그램을 파이썬 프로그래밍을 이용하여 개발하였다.

Abstract In addition to yellow dust occurring in China, fine dust has become a hot topic in Korea through news and media. Although there is fine dust generated from the outside, the purchase rate of air purifier products is increasing as external fine dust flows into the inside. The air purifier uses a filter internally, and the sensor notifies the user through the LED alarm whether the filter is replaced. However, there is currently no product measuring how much the filter rate is reduced and determining the pressure of the blower to operate. Therefore, in this paper, data are generated directly using Arduino, fine dust sensor, and differential pressure sensor. In addition, a program was developed using Python programming to calculate how old the filter is and to analyze the wind power of the blower according to the filter rate by calculating the measured dust and pressure values.

Key Words : IoT Sensor, Arduino, Dust Analysis, Python Programming

*준회원, 안양대학교 컴퓨터공학과 박사과정

**준회원, 경기과학기술대학교 컴퓨터모바일융합공학과 학부생

***준회원, 안양대학교 컴퓨터공학과 석사과정

****정회원, 여주대학교 소프트웨어융합과 조교수

*****정회원, 안양대학교 ICT융합학부 소프트웨어전공 조교수 (교신저자)Korea.

접수일자 2020년 8월 31일, 수정완료 2020년 11월 3일

게재확정일자 2021년 2월 5일

Received: 31 August, 2020 / Revised: 3 November, 2020 /

Accepted: 5 February, 2021

****Corresponding Author: jkim@anyang.ac.kr

Dept. ICT Convergence Engineering at AnYang University.

I. 서 론

ICT 기술의 발달로 인해 4차 산업혁명의 주 기술인 빅데이터, 인공지능, IoT(Internet of Things), 3D 프린팅이 많이 활용되고 있다. 특히, IoT 기술은 다양한 센서를 활용하여 빅데이터 분석과 인공지능을 활용한 자동화 시스템을 운영할 때 주로 사용되고 있다.

대표적인 IoT 기기로는 라즈베리 파이, 아두이노가 많이 사용되고 있으며, 라즈베리 파이는 운영체제를 이용하여 활용성이 높고, 파이썬 프로그래밍이 가능한 장점이 있다. 하지만, 아두이노는 C언어 기반의 프로그래밍과 운영체제가 없기 때문에 불편할 수 있지만, LCD, 모터와 같은 단순 센서 역할만 수행할 때는 라즈베리 파이보다 더 활용성이 높다.

또한, 최근 미세먼지 이슈로 인해 이를 줄이기 위한 관련 기술 개발과 다양한 연구가 진행되고 있다. '초미세먼지 문제 해결을 위한 연구 및 정책 방향' 논문 저자는 초미세먼지의 발생 원인과 문제 해결을 위해 세계 각국의 정책과 우리나라에 필요한 연구 방향을 제안하였다^[1].

이처럼 미세먼지는 국가 환경에 대해 심각한 문제를 일으키고 있어 본 논문에서는 아두이노와 미세먼지 및 차압 센서를 이용하여 데이터를 생성하고, 데이터베이스로 실시간 전송하였다. 그리고 저장된 데이터를 파이썬 프로그래밍을 이용한 분석 프로그램을 구현하여 미세먼지의 값과 공기압 간의 관계를 분석하였다.

논문의 구성은 다음과 같다. 2장에서는 본 연구에서 사용되는 센서를 소개하고, 관련된 연구를 소개한다. 3장에서는 본 연구에서 제시하는 시스템의 구조와 상세 프로세스를 설명한다. 마지막으로 4장에서는 결론과 향후 연구 내용에 대하여 설명한다.

II. 관련 센서 및 연구

1. 아두이노

아두이노는 마이크로컨트롤러를 사용하여 만들어진 개발 보드이다. 다양한 제어장치를 사용하여 쉽고 간편한 개발환경을 갖추고 탈부착이 가능한 핀들로 구성되어 있어 보드를 수정하는 것이 쉽다^[2].

따라서 온도, 습도, 등 다양한 센서를 내부의 모듈을 사용하여 개발 보드를 완성할 수 있으며, 아두이노 전용 개발 IDE를 통해 센서 동작에 필요한 라이브러리를 사용자가 직접 설치할 수 있다^[3].

2. 미세먼지 센서(GP2Y1014AU0F)

다양한 미세먼지 센서 중에서 GP2Y1014AU0F는 적외선 발광 다이오드(IRED)와 포토다이오드가 대각선으로 배치되어 공기 중 먼지에서 발생한 반사광을 감지할 수 있다^[4]. 또한, 공기 중에 존재하는 먼지와 입자를 측정하는데 필요한 전류 소모량이 7V DC 전원으로 동작 가능하며 농도에 따라 아날로그 출력을 통해 아두이노의 아날로그 핀으로 값을 읽는 장점이 있다^[5]. 측정할 수 있는 미세먼지의 범위는 PM10을 포함하여 PM2.5까지 측정이 가능하여 PM2.5보다 낮은 수치인 초미세먼지까지 측정이 가능하다^[6].

3. 차압 센서(53A-L15H-2210)

차압 센서(53A-L15H-2210)는 아두이노 개발 환경에서 압력의 차를 측정하여 데이터를 생성하는 센서이다. 차압 센서는 오리지널 차압 센서(Differential pressure sensor)와 게이지식 차압 센서(Gauge pressure sensor) 두 가지로 구분한다.

오리지널 차압 센서는 두 환경의 압력의 차이를 측정하며 수압 요소인 다이어프램이나 벨로즈의 양쪽에 압력을 주는 방식과 2개의 수압 요소를 사용하는 방식이 있다. 또한, 오리피스 유량계용으로 많이 이용되고 정압이 차압 측정 범위의 몇 배가 되는 경우가 있어 많은 정압이나 오버레이 지압이 적용하는 조건에서 이용하는 데 걸리는 특성이 필요하다^[7].

게이지식 차압 센서의 경우 하나의 환경은 대기압을 보정하며, 다른 하나의 환경에서 가해지는 압력을 측정하여 대기압에 대한 압력을 측정하는 센서이다. 수압 요소인 다이어프램이나 벨로즈의 피 측정 쪽을 대기에 드러도록 하여 대기와의 차압을 측정한다. 차압 센서의 종류 중 하나지만 항상 대기압을 이용하기 때문에 차압 센서의 정압 특성을 필요로 하지 않아 압력센서로써 많이 사용한다.

4. 관련 연구

공공데이터 포털로 속해있는 '에어코리아'에서 제공하는 미세먼지 데이터를 이용한 연구가 있다^[8]. 이 연구는 미세먼지와 초미세먼지의 측정량 데이터를 사용하고, 이 변량 자료를 조사하여 이들 사이의 확률적 관계를 밝히는 것을 목적으로 한다. 이 연구의 결과로 미세먼지와 초미세먼지의 결합 범위의 확률을 나타내고 각각의 결합등급에 속하는 확률을 밝힌다.

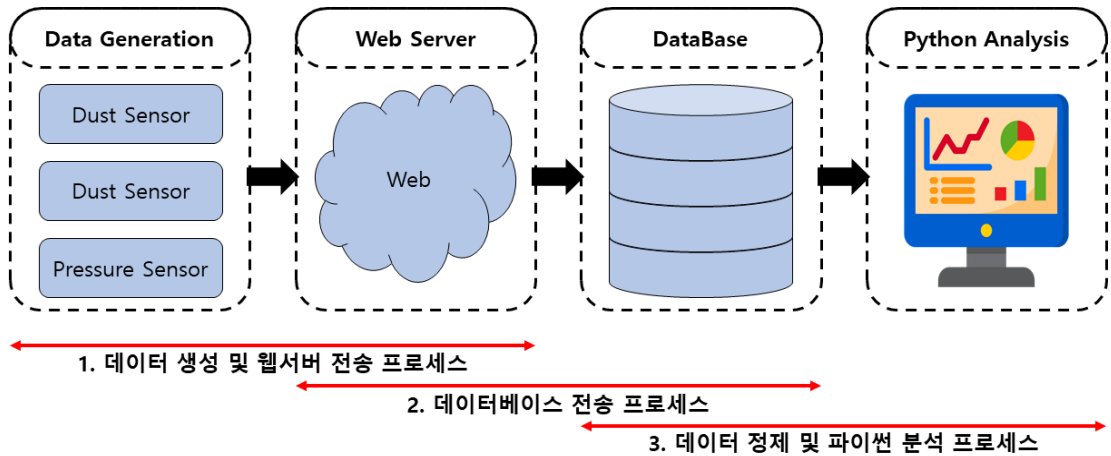


그림 1. 빅데이터 기반 센서 데이터 처리 및 분석 시스템 구조도
 Fig. 1. System Architecture of Big Data-based for Sensor Data Processing and Analysis

또한, 공공데이터 포털에 속해있는 ‘국립환경과학원’에서 제공하는 데이터를 이용하여 대기 중 미세먼지 농도를 이용한 연구가 있다^[9]. 이 연구는 미세먼지가 심한 포항 지역을 대상으로 2001년 이후로 측정된 미세먼지 농도자료를 이용하였다.

따라서 포항의 미세먼지 오염도에 관한 자연, 인위적 요인에 대해 종합적 특성을 알아내는 것이 목적이다. 이 연구의 결과로 대기오염물질은 많은 자연적인 기상요인에 의해 미세먼지 농도는 증감에 민감한 영향을 받는다고 언급하고 있다. 이처럼 미세먼지에 대한 국가 환경 문제가 증가하고 있으며, 본 논문에서는 직접 IoT 센서를 통해 측정된 미세먼지에 대해서 분석한다.

III. 시스템 구조 및 프로세스 소개

본 논문에서 제안하는 시스템 구조는 그림 1과 같이 3가지의 ‘데이터 생성 및 웹서버 전송 프로세스’, ‘데이터베이스 전송 프로세스’, ‘데이터 정제 및 파이썬 분석 프로세스’로 구성된다.

‘데이터 생성 및 웹서버 전송 프로세스’는 아두이노와 미세먼지 센서, 차압 센서를 이용하여 데이터를 생성하고, 웹서버로 데이터를 전송한다. 데이터베이스에 직접 저장하지 않고, 웹서버를 경유 하는 이유는 데이터를 네트워크를 통해 데이터베이스에 저장할 때 직접적인 접근은 보안상 허용되지 않기 때문이다. 또한, 웹서버는 기본적인 Apache 서버가 사용되며, PHP 문서를 통해서 아두이노에서 전송한 데이터를 받을 수 있다.

‘데이터베이스 전송 프로세스’는 웹서버의 PHP 문서로 데이터가 전송되면 전송된 데이터를 데이터베이스에 파싱하여 저장하는 역할을 한다.

‘데이터 정제 및 파이썬 분석 프로세스’는 데이터베이스에 저장되어있는 데이터를 정제하고, 파이썬 프로그램을 이용하여 분석하는 단계를 나타낸다.

1. 데이터 생성 및 웹서버 전송 프로세스

본 1절에서는 IoT 기기로 사용한 미세먼지 센서와 차압 센서를 아두이노와 브레드보드에 연결하고, 추가로 ESP8266 모듈을 이용하여 무슨 네트워크를 통해 웹서버에 있는 PHP 문서로 데이터를 전송하는 역할을 한다.

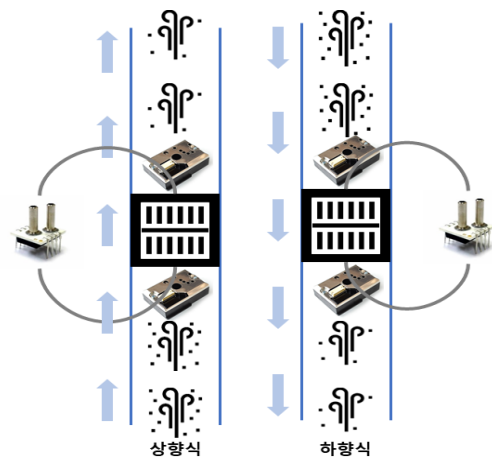


그림 2. 데이터 생성을 위한 시스템 시나리오
 Fig. 2. System Scenarios for Data Generation

그림 2는 데이터를 생성하여 수집하기 위한 시스템 시나리오를 보여준다.

상향식과 하향식 두 가지 방법을 사용하고 있으며, 상향식은 바람이 아래에서 위로 진행하여 바람이 통하는 통로에 필터를 설치한다. 필터 전 미세먼지 센서 1개, 필터 후 미세먼지 센서 1개, 필터 전후에 튜브를 이용해서 바람의 압력을 계산하는 차압 센서 1개가 사용된다.

하향식의 경우 바람이 흐르는 경로만 반대로 위에서 아래로 흐르는 방법이며, 사용한 센서와 방식은 상향식과 같기에 설명을 생략한다.



그림 3. 상향식(좌)과 하향식(우) 데이터 수집 방법
Fig. 3. Top-down (left) and top-down (right) data collection methods

그림 3은 시나리오를 기반으로 데이터를 수집 할 때 사용된 기기의 모습을 나타내며, 그림 3의 좌측이 상향식 방법이며 우측이 하향식 방법이다. 또한, 외부의 미세먼지를 직접 실내로 가져와 실험하는 방법은 환경상 어렵다. 따라서 고기집에 방문하여 센서를 연결하고, 고기를 구우면서 발생하는 연소 가스를 측정하여 데이터로 사용하였다. 그림 4는 사용한 아두이노 6개와 미세먼지 센서 4개, 차압 센서 2개의 모습을 보여준다.

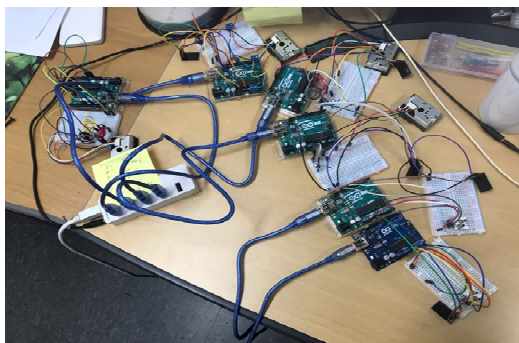


그림 4. 데이터 수집에 사용된 IoT 센서
Fig. 4. IoT sensors used to collect data

```

1  #include "ESP8266.h"
   #include <SoftwareSerial.h>
   #define SSID      "SSID"
   #define PASSWORD  "PASSWORD"
   #define HOST_NAME "192.168.0.15"

2  const int sharpLEDPin = 2;
   const int sharpVoPin = A0;
   SoftwareSerial mySerial(10, 9);
   ESP8266 wifi(mySerial);

3  void setup() {
   pinMode(sharpLEDPin, OUTPUT);
   ..... }

4  void loop() {
   digitalWrite(sharpLEDPin, LOW);
   .....
   Vo = Vo / 1024.0 * 5.0;
   float dV = Vo - Voc;
   if ( dV < 0 ) {
     dV = 0;
     Voc = Vo; }

5  String cmd = "AT+CIPSTART=\\"TCP\\","\\"";
   cmd += HOST_NAME;
   cmd += "\",80";
   Serial.println(cmd);
   mySerial.println(cmd);
   if (mySerial.find("Error")) {
     Serial.println( "TCP connect error" );
     return; }
   char Dust[20];

   String ValueB(floatToString(Dust, dustDensity, 2, 0));
   cmd = "GET /insert_dust1.php?dust=" + ValueB +
         "(ug/m3)'\r\n"; }
    
```

그림 5. 미세먼지 센서 핵심 코드
Fig. 5. Fine dust sensor core code

그림 5는 미세먼지 센서를 동작하기 위한 코드 중 핵심 부분을 보여준다.

1) 무선 와이파이 센서인 ESP8266을 활용하기 위해 라이브러리 헤더를 지정하고, ESP8266 센서가 무선 공유기를 통해 데이터베이스에 데이터를 저장하기 때문에 무선 공유기에 접속하기 위해 접속 관련 무선 공유기 이름(SSID), 무선 공유기 아이디(ID), 비밀번호(PW)를 지정한다.

2) 아두이노에 미세먼지 센서와 무선 와이파이 센서의 핀을 브레드보드와 연결하기 위해 핀 번호를 지정하며, 3) 아두이노의 전원을 연결하고, 초기에 발생하는 신호를 설정하기 위한 값을 지정한다. 4) 실제로 아두이노가 동작하면서 측정되는 값을 센싱하기 위한 부분이며, 아두이노는 라즈베리 파이와 다르게 아날로그값만 센싱할 수 있기 때문에 아날로그값을 디지털로 변환한다.

5) 센싱된 값을 기존 1)에서 설정한 무선 공유기 이름, 아이디, 비밀번호 값을 이용하여 웹서버로 전송하기 위한 연결을 한다. 웹서버에서 PHP로 제작된 문서를 이용하여 데이터베이스에 실시간으로 값을 저장하기 위해 cmd

변수에 HTML Get 요청 쿼리를 이용한다.

그림 6은 차압 센서를 동작하기 위한 코드 중 핵심 부분을 보여준다.

```

1  1) 미세먼지 센서 코드와 동일
2  2) 미세먼지 센서 코드와 동일
3  3) 미세먼지 센서 코드와 동일
4  4) 미세먼지 센서 코드와 동일

5  void loop() {
    void loop() {
      float sensor_volt;
      float sensorValue = 0;
      float pressure_value_PSI = 0;
      float pressure_range_PSI = 0.15;
      int pressure_value_PA = 0;
      int pressure_range_PA = 1034;

      sensorValue = sensorValue/100.0;
      sensor_volt = sensorValue/1024*5.0;
      pressure_value_PSI = (sensor_volt - 0.5) *
      (pressure_range_PSI / (5 * 0.8));
      pressure_value_PA = (sensor_volt - 0.5) *
      (pressure_range_PA / (5 * 0.8));
6  }

7  6) 미세먼지 센서 코드와 동일

      char Psi[20];
      String ValueB(floatToString(Psi,pressure_value_PSI, 2,
      0));
      char Pa[20];
      String ValueC(floatToString(Pa,pressure_value_PA, 2,
      0));

      cmd = "GET
      /insert_press.php?psi="+ValueB+"&pa="+ValueC+"pa'
      \r\n"; }
    
```

그림 6. 차압 센서 핵심 코드
 Fig. 6. Differential pressure sensor core code

1), 2), 3) 4), 6) 부분은 무선 공유기 접속을 위한 초기 설정값과 아두이노에서 센서를 데이터를 발생하기 위한 신호 처리 값을 나타내며, 미세먼지 코드를 설명할 때 작성했던 코드와 동일하기 때문에 생략하였다.

5) 차압 센서에서 발생하는 센서 값을 저장하고, 차압 센서의 초깃값을 설정하는 부분이다. 미세먼지 센서와 동일하게 아날로그값으로 데이터가 저장되기 때문에 디지털화하기 위한 계산식도 포함되어 있다. 7) 아두이노에서 웹서버로 데이터를 전송하는 부분으로, 차압 센서에서 발생하는 데이터를 문자열로 저장하는 Psi, Pa 변수를 선언하고, 값을 문자열 형태로 변환하여 HTML Get 요청 쿼리를 이용하여 웹서버에 있는 PHP 코드로 전송하게 된다.

2. 데이터베이스 전송 베이스

본 2절에서는 아두이노에서 데이터를 생성하여 웹서버로 데이터를 전송된 데이터를 파싱받아 데이터베이스

에 저장하는 역할을 한다. 그림 7은 웹서버 PHP 문서의 코드를 보여준다.

```

<?php
1  $con = mysqli_connect("localhost", "root", "tlseh12",
2  "db");
    if (mysqli_connect_errno()) {
        echo "Failed to connect to MySQL: " .
        mysqli_connect_error(); }

3  $Dust = $_GET["dust"];
    $sql = "insert into db.dust1(Dust) values($Dust)";
    $Psi = $_GET["psi"];
    $Pa = $_GET["pa"];
    $sql = "insert into db.press(Psi, Pa) values($Psi, $Pa)";

    mysqli_query($con, $sql);
    mysqli_close($con);
?>
    
```

그림 7. PHP 문서 코드
 Fig. 7. PHP Document Code

1) con 변수를 지정하고, 사용하고 있는 데이터베이스인 MySQL에 접속하기 위한 접속 호스트 주소, 아이디, 비밀번호, 사용할 DB를 설정한다. 2) 만약 오류가 발생할 경우를 대비해 한 번에 오류를 확인할 수 있도록 오류 발생 확인 코드를 추가한다.

3) Dust 변수에 미세먼지 센서 코드를 통해 생성되는 데이터를 가져오고, Psi 변수에 압력 센서 코드를 통해 생성되는 데이터 값을 가져온다. 그리고 Insert SQL 질의문을 이용하여 각 테이블에 데이터를 저장하고, 연결을 끊는다.

그림 8은 테이블 스키마 구조와 데이터 파싱이 완료되어 데이터베이스에 저장된 모습을 보여준다.

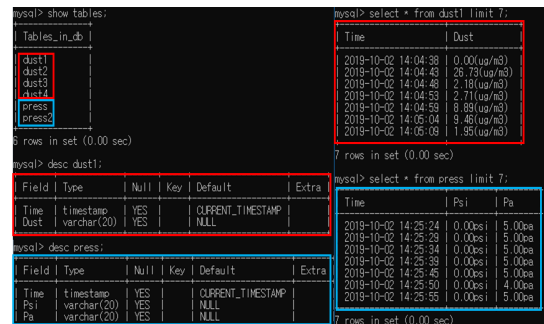


그림 8. 테이블 스키마와 데이터가 저장된 모습
 Fig. 8. Table schema and data stored

웹서버 PHP에서 사용할 DB를 선택하고, 미세먼지가 저장되는 테이블인 dust1, dust2, dust3, dust4의 스키마는 그림과 같이 "시간, 미세먼지 측정값"으로 구성되어

있다. 차압 센서가 저장되는 테이블은 press, press2 테이블이며, 저장되는 값은 “시간, Psi 단위 값, Pa 단위 값”으로 구성된다. 오른쪽 그림을 확인하면 실시간으로 미세먼지가 측정되어 데이터베이스에 저장되는 것을 확인할 수 있다.

3. 데이터 정제 및 파이썬 분석 프로세스

본 3절에서는 수집이 완료된 데이터들을 하나의 데이터로 합쳐서 정제하는 작업과 최종 정제된 데이터를 파이썬 분석 프로그램을 이용하여 분석이 수행된다.

A	B	A	B	A	B	C
time	beforedust	time	afterdust	Time	psi	pa
2019-12-01 10:48:02	90	2019-12-01 10:48:02	50	2019-12-01 10:48:02	140	958160
2019-12-01 10:48:06	100	2019-12-01 10:48:06	45	2019-12-01 10:48:06	135	923940
2019-12-01 10:48:10	90	2019-12-01 10:48:10	55	2019-12-01 10:48:10	135	923940
2019-12-01 10:48:14	85	2019-12-01 10:48:14	50	2019-12-01 10:48:14	140	958160
2019-12-01 10:48:18	80	2019-12-01 10:48:18	45	2019-12-01 10:48:18	145	992380
2019-12-01 10:48:22	130	2019-12-01 10:48:22	50	2019-12-01 10:48:22	280	1916320
2019-12-01 10:48:26	120	2019-12-01 10:48:26	50	2019-12-01 10:48:26	290	1904760
2019-12-01 10:48:30	145	2019-12-01 10:48:30	40	2019-12-01 10:48:30	300	2053200

A	B	C	D	E
Time	beforedust	afterdust	pressure	rate
1	90	50	140	40
2	100	45	135	55
3	90	55	135	35
4	85	50	140	35
5	80	45	145	35
6	130	50	280	80
7	120	50	290	70
8	145	40	300	105

그림 9. 수집된 데이터 및 최종 정제 데이터
Fig. 9. Data collected and final refining data

그림 9는 각 센서를 통해 수집된 데이터와 최종 정제된 데이터의 모습을 보여주며, 그림 9의 상단 좌측부터 필터 전 미세먼지 센서의 수집된 데이터, 중간은 필터 후 미세먼지 센서의 수집된 데이터, 우측은 차압 센서의 수집된 데이터를 의미한다. 그림 9의 아래는 최종 정제된 데이터의 모습을 보여준다.

최종 정제된 데이터에서 pressure 컬럼의 값은 차압 센서의 psi 컬럼을 의미하고, rate 컬럼의 값은 필터 전 미세먼지 값에서 필터 후 미세먼지 값의 빼기 연산을 통한 결과를 의미한다.

최종 정제된 데이터를 파이썬 프로그램을 이용하여 개발한 분석 프로그램의 입력 데이터로 사용한다. 또한, 입력 데이터는 이전 데이터와 현재 데이터 두 가지를 사용하며, 이전 데이터는 측정 시점으로부터 일정 시간 이전에 측정된 데이터를 의미하고, 현재 데이터는 최근에 측정된 데이터를 의미한다.

그림 10은 파이썬 프로그래밍을 이용하여 개발한 분석 프로그램의 UI를 보여준다.

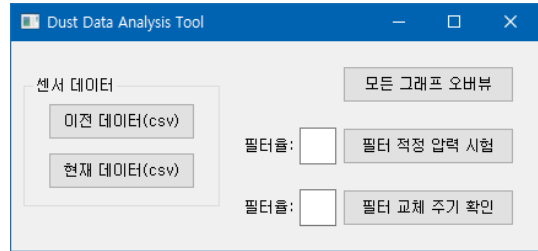


그림 10. 파이썬 분석 프로그램 UI
Fig. 10. Python Analysis Program UI

이전 데이터와 현재 데이터를 입력하고, 오른쪽에 있는 분석 버튼을 선택하면, 입력한 데이터를 기준으로 다양한 분석이 실행되어 그래프의 형태로 표현이 된다. 그림 11은 데이터를 입력할 때 모습을 보여준다.

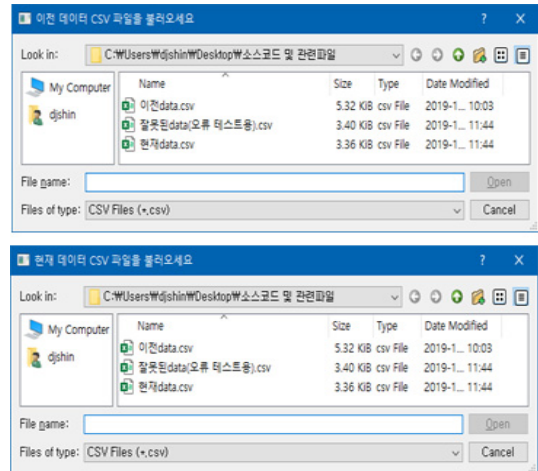


그림 11. 센서 데이터 입력 화면
Fig. 11. Sensor data entry screen

그림 11의 상단은 이전 데이터 버튼을 선택했을 때 파일 탐색기 창이 출력되며, 최종 정제된 데이터 중 이전 데이터 파일을 입력하면 된다. 또한, 그림 11의 하단처럼 현재 데이터 버튼을 선택하면 이전 데이터와 동일하게 현재 데이터 파일을 입력하면 된다.

그림 12는 파일의 입력이 완료된 후 그림 10에서 오른쪽에 위치한 분석 버튼 중 첫 번째인 '모든 그래프 오버뷰' 버튼을 선택했을 때 결과를 보여준다.

첫 번째 그래프는 x축을 시간, y축을 필터 전 미세먼지로 설정하여 그래프를 표현하였으며, 두 번째 그래프는 x축을 시간, y축을 필터 후 미세먼지로 설정하였고, 마지막 그래프는 x축을 시간, y축을 압력으로 지정하여 출력하였다.

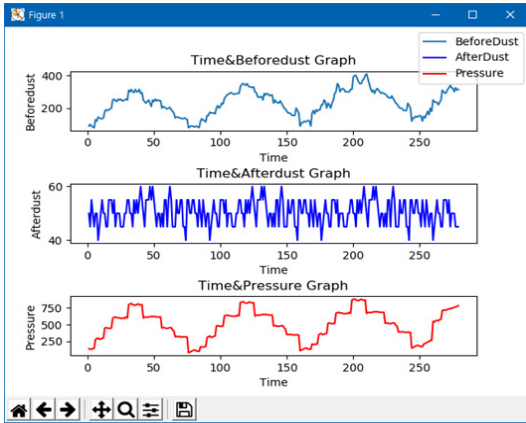


그림 12. 모든 그래프 오버뷰 결과
 Fig. 12. All Graph Overview Results

시간이 지남에 따라 미세먼지가 높은 수치로 들어올 때 필터를 통해 수치가 낮아지는 것을 볼 수 있으며, 수동으로 송풍기의 압력을 수동으로 강하게 돌릴 때 미세먼지가 낮아지는 것을 확인할 수 있다.

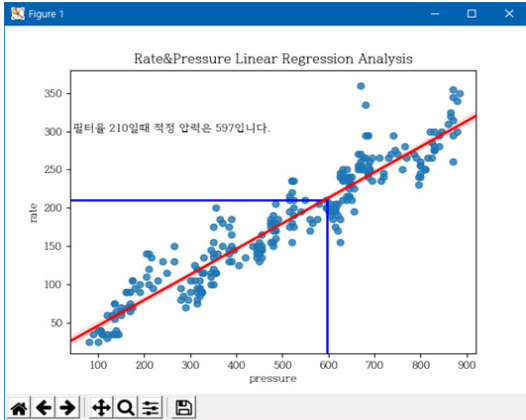


그림 13. 필터 적정 압력 시험 결과
 Fig. 13. Optimum filter pressure test results

그림 13은 그림 10에서 오른쪽에 위치한 분석 버튼 중 두 번째인 '필터 적정 압력 시험' 버튼을 선택했을 때 결과를 보여준다.

UI 창에서 필터율 부분에 130을 입력했을 때 출력되는 그래프이다. 필터율이 130일 때 회귀분석에 의해서 생성된 회귀곡선에 따라 적정 압력이 597이라는 값을 알려준다. 따라서 현재 동작하는 시스템에서 필터율을 확인하고, 필터율을 낮추기 위한 송풍기의 압력 값을 결정하고, 조절할 수 있다.

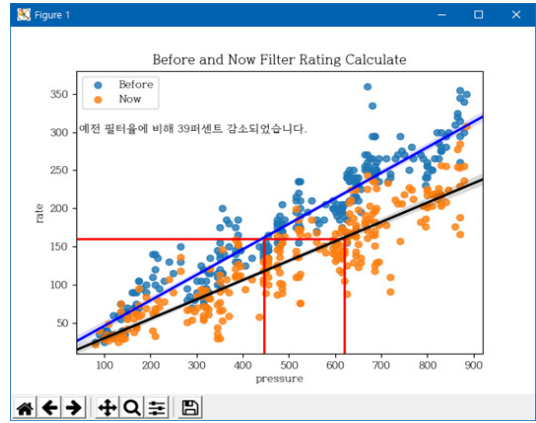


그림 14. 필터 교체 주기 확인 결과
 Fig. 14. Filter replacement cycle check results

그림 14는 그림 10에서 오른쪽에 위치한 분석 버튼 중 세 번째인 '필터 교체 주기 확인' 버튼을 선택했을 때 결과를 보여준다.

이전 데이터와 현재 데이터의 회귀분석 곡선을 통해 분석된 그래프를 출력하며, 필터의 교체 주기를 확인할 수 있다. 사실 필터의 교체주기는 필터가 얼마나 오염되었는지에 따라 교체를 해야 하며, 필터를 통과할 때 미세먼지의 양이 많지 않다면 필터가 오래 유지되기 때문에 사용 환경에 따라 필터 주기는 다르게 된다.

본 필터 교체 주기 확인 그래프는 이전 데이터를 활용하여 100%로 기준을 잡고, 일정 시간이 지난 후에 데이터인 현재 데이터를 입력하였을 때 얼마나 필터가 감소하였는지 출력하여 필터의 교체 주기를 알려준다.

따라서 필터율이 160일 때 이전 데이터의 압력 값은 약 450, 현재 데이터의 압력 값은 620으로 이전 데이터 필터율과 압력을 100%로 기준 잡고, 얼마나 감소하였는지를 그래프의 좌측 상단의 "예전 필터율에 비해 39 퍼센트 감소 되었습니다."라는 결과를 통해 필터가 얼마나 노화되었는지 확인할 수 있다.

그림 15부터 17까지는 파이썬 분석 프로그램을 이용할 때 발생하는 다양한 오류 경고창을 보여준다.

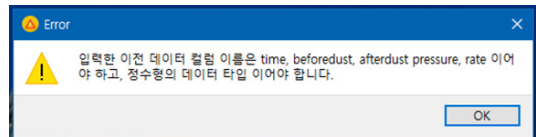


그림 15. 잘못된 데이터 입력 시 발생하는 오류 경고창
 Fig. 15. Error alert when entering wrong data

그림 15는 이전 데이터와 현재 데이터를 파일 탐색기를 통해 입력할 때 잘못된 컬럼의 형식을 가지고 있을 때 발생하는 오류 경고창이다. 즉, 최종 데이터 형식인 “시간, 필터 전 미세먼지 값, 필터 후 미세먼지 값, 압력 값, 필터율”로 이루어진 데이터가 아니면 경고 오류 창을 출력하여 사용자에게 알려준다.

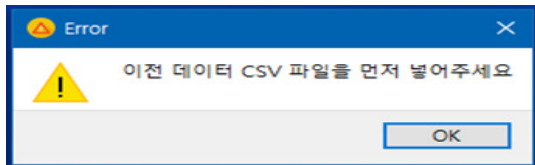


그림 16. 입력 데이터 없이 분석할 때 발생하는 오류 경고창
Fig. 16. Error warning window when analyzing without input data

그림 16은 입력 데이터와 현재 데이터 버튼을 선택하지 않아 입력된 데이터가 없이 분석 버튼을 선택했을 때 발생하는 오류 경고창이다.

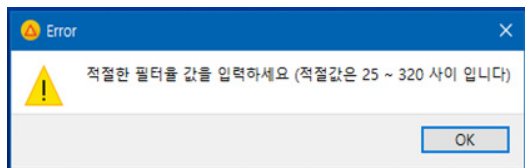


그림 17. 잘못된 필터율 값 입력 시 발생하는 오류 경고 창
Fig. 17. Error warning window when you enter an invalid filter rate value

그림 17은 ‘필터 적정 압력 시험’ 버튼과 ‘필터 교체 주기 확인’ 버튼 좌측에 위치한 필터율 텍스트 박스에 잘못된 값을 입력했을 때 발생하는 오류이다. 0을 비롯하여 정수만 입력되며, 실수와 같이 측정되지 않는 범위의 값을 입력하면 그림 17과 동일한 오류 경고창을 출력하게 된다. 또한, 공백과 특수 기호로 취급되는 더하기, 빼기, 곱하기와 같은 기호를 입력했을 때도 발생한다. 이처럼 분석기를 처음 접하는 사용자도 쉽게 오류 내용을 파악할 수 있는 예외 처리도 적용하였다.

IV. 결 론

본 논문에서는 IoT 환경에서 주로 사용되는 아두이노와 미세먼지 센서 및 차압 센서를 활용하여 필터율에 따라 적정 압력 값을 확인하며, 필터의 노화를 확인하고, 필

터를 교체할 수 있는 프로그램을 개발하여 적용하였다.

하지만, 필터 적정 압력 시험을 통해 현재 필터율 수치에 따라 송풍기의 압력 값을 조절할 수 있는 기능은 수동이 아닌 자동으로 이루어지고, 송풍기를 강하게 동작시켜야 필터를 조금 더 오랫동안 사용할 수 있다.

따라서 본 시스템을 더 확장하여 빅데이터와 인공지능 기반 알고리즘을 활용하여 현재 필터율에 따라 자동으로 송풍기의 압력을 조절할 수 있는 연구를 수행할 예정이다.

References

- [1] Yong-Pyo Kim, “Research and Policy Directions against Ambient Fine Particles,” *Journal of Korean Society for Atmospheric Environment*, Vol. 33, No. 3, pp. 191-204, 2017
DOI : <https://doi.org/10.5572/KOSAE.2017.33.3.191>
- [2] Hyoung-Ro Lee, Chi-ho Lin, “Design and Implementation of Arduino-based Efficient Home Security Monitoring System,” *The Journal of The Institute of Internet, Broadcasting and Communication (IIBC)*, Vol. 16, No. 2, pp. 49-54, 2016.
DOI : <http://dx.doi.org/10.7236/IIBC.2016.16.2.49>
- [3] Jin-Gyeong Kim, Sang-Yong Ra, Min-Seok Kim, Jung-Hoon Kim, Jun-Dong Lee, “The Implementation of Wireless Fine Dust Sensor System Based on Arduino,” *The Korean Society of Computer And Information Conference*, Vol. 26, No. 2, pp. 234-235, 2018.
- [4] Jin-Su Kim, You-Jung Jang, Jin-Seok Kim, Min-Woo Park, Chan-Jong Bu, Yun-Gu Lee, Youn-Ha Kim, Jung-Hun Woo, “Evaluation of the Usability of Micro-Sensors for the Portable Fine Particle Measurement,” *Journal of Environmental Impact Assessment*, Vol. 27, No. 4, pp. 378-393, 2018.
DOI : <http://dx.doi.org/10.14249/eia.2018.27.4.378>
- [5] Chang-Seok Kim, Dong-Guk Kim, “A Study on the Performance Comparison of Fine Dust Sensor,” *The Korean Institute of Electrical Engineers Information and Control Symposium*, pp. 253-254, 2018.
- [6] Sang-Ik Lee, Jin-Kook Lee, “Visualization of the Comparison between Airborne Dust Concentration Data of Indoor Rooms on a Building Model,” *Journal of the Korean Housing Association* Vol. 26, No. 4, pp. 55-62, 2015.
DOI : <http://dx.doi.org/10.6107/JKHA.2015.26.4.055>.
- [7] Kyu-Sik Shin, Sang-Woo Song, Kyun-Gil Lee, Dae-Sung Lee, Jae-Pil Jung, “Fabrication of Single Capacitive type Differential pressure sensor for Differential Flow meter,” *Journal of the Microelectronics and Packaging Society*, Vol. 24, No. 1, pp. 51-56, 2017.
DOI : <http://dx.doi.org/10.6117/kmeps.2017.24.1.051>

- [8] Hye-Sun Shin, "Analysis of fine dust data using bivariate distributions," EWHA Womans University Master's Degree Thesis, 2018.
- [9] Min-Seok Choi, Sung-Ok Baek, "Statistical Analysis of PM10 Data in a Steel-Industrial Area," Korean Society for Atmospheric Environment, Vol. 32, No. 3, pp. 329-341, 2016.
DOI : <https://doi.org/10.5572/KOSAE.2016.32.3.329>

저 자 소 개

신 동 진(준회원)



- Dong-Jin Shin received BS in Department of Computer Science and MS in Department of Smart Manufacturing Engineering at the Korea Polytechnic University in 2018 and 2020. He is currently studying Phd in Department of Computer Science at AnYang University. His research interests include Big Data, Internet of Things (IoT), Network&System security.

이 진(준회원)



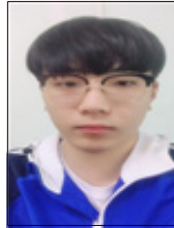
- Jin Lee is currently studying AS in Computer Mobile Convergence at Gyeonggi University of Science and Technology in 2020. Her research interests include Internet of Things (IOT), Game development, etc.

허 민 희(준회원)



- Min-hui Heo is currently studying AS in Computer Mobile Convergence at Gyeonggi University of Science and Technology in 2020. Her research interests include Big Data, Data Analysis, etc.

황 승 연(준회원)



- Seung-Yeon Hwang is received his BS in Department of Computer Science at Korea Polytechnic University in 2019. He is currently studying MS in Department of Computer Science at AnYang University. His research interests include Database System, Big Data, Data Analysis, Machine Learning, etc.

이 용 수(정회원)



- Yong-soo Lee received his MS in Computer Science at Konkuk University in 1989. In 2015, he received his PhD in Information & Control Engineering at Kwangwoon University. He is currently a professor at the Department of software convergence at Yeosu Institute of Technology. He is the Member of the Korea Institute of Internet, Broadcasting & Communication (IIBC). His research interests include Database Systems, Data Mining, BigData, Wireless Sensor Networks and Ubiquitous Sensor Network (USN), etc.

김 정 준(정회원)



- Jeong-Joon Kim received his BS and MS in Computer Science at Konkuk University in 2003 and 2005, respectively. In 2010, he received his PhD in at Konkuk University. He is currently a Professor in software major ICT Convergence Engineering at AnYang University. His research interests include Database Systems, Big Data, Semantic Web, Geographic Information Systems (GIS) and Ubiquitous Sensor Network (USN), etc.