

## 중학생들을 위한 아두이노 기반 구름 발생 실험의 개발 및 적용

원정재<sup>1</sup> · 정덕호<sup>2</sup> · 고 결<sup>1\*</sup>

<sup>1</sup>전북대학교 교육대학원 지구과학교육전공, 전주 54896

<sup>2</sup>전북대학교 과학교육학부, 과학교육연구소, 전주 54896

## Design and Implementation of an Arduino-Based Cloud Formation Experiment for the Middle School Students

Won Jeong-Jae<sup>1</sup>, Chung Duk-Ho<sup>2</sup>, and Ko Gyu<sup>1\*</sup>

<sup>1</sup>Division of Earth Science Education, Graduate School of Education, Jeonbuk National University, Jeonju 54896, Korea

<sup>2</sup>Division of Science Education, Institute of Science Education, Jeonbuk National University, Jeonju 54896, Korea

**초 록:** 학교 교육 현장에 디지털 기술을 접목해야 할 필요성이 증가하고 있다. 따라서 이 연구에서는 수업에 활용 가능한 디지털 기기로서 주목받고 있는 아두이노를 실제 교수학습에 활용할 수 있는지 알아보고자 실험 과정에서 측정된 물리량을 그래프의 형태로 실시간 시각화할 수 있는 모델을 개발하였다. 2015년 개정 교육과정 중학교 3학년 과학 교과서에는 구름 발생 실험장치가 제시되어 있다. 여기에 아두이노 우노 보드와 DHT11, BMP280 두 가지 센서를 설치하여 단열 과정에서 변화하는 물리량을 측정하였다. 측정된 데이터값은 Python의 시각화 도구를 사용하여 그래프 형태로 표현하였으며 BMP 280 센서에서 감지한 기압과 온도 값은 압력 감소에 따른 단열팽창과 온도 감소의 상관성을 명확히 보여주었다. 다만 DHT11 센서와 같이 센서의 응답 속도가 5~10초인 경우 단열 과정과 같이 물리량이 변하는 시간 규모가 1초보다 작은 현상을 관측하는 데 있어 어려움이 있었다. 이 아두이노 모델을 교수학습에 적용한다면 학생들의 구름 발생과 단열팽창에 대한 이론적 내용의 이해를 도울 수 있을 것으로 기대한다.

**중심어:** 아두이노, 디지털 기기, 구름 발생 실험, 단열팽창

**Abstract:** The need to integrate digital technology into school education is increasing. Accordingly, this study aimed to explore the feasibility of using Arduino, a digital device gaining attention for its applicability in classrooms, by developing a model that visualizes physical quantities measured during experiments in real time as graphs. The 2015 revised curriculum for 9th-grade science textbooks presents a cloud formation experiment apparatus. An Arduino Uno board equipped with two sensors, DHT11 and BMP280, was used to measure the physical quantities that change during the adiabatic process. The collected data were visualized as graphs using Python visualization tools. The data from the BMP280 sensor clearly demonstrated the correlation between adiabatic expansion, characterized by pressure decrease, and temperature decrease. However, sensors like the DHT11, with a response time of 5 to 10 seconds, had difficulty observing phenomena with time scales shorter than 1 second, such as those occurring during the adiabatic process. This Arduino model is expected to help students better understand theoretical concepts related to cloud formation and adiabatic expansion when applied to teaching and learning.

**Keywords:** Arduino, Digital inquiry tools, Cloud formation experiment, Adiabatic expansion

\*Corresponding Author: Ko Gyu

Phone: +82-63-270-3630

E-mail: kg2001@kigam.re.kr



All the content in Journal of Science & Science Education(JSSE) is Open Access, meaning it is accessible online to everyone, without fee and authors' permission. All JSSE content is published and distributed under the terms of the Creative Commons Attribution License (<http://creativecommons.org/licenses/by/4.0/>). Under this license, the authors retain full ownership of their work, while permitting anyone to use, distribute, and reproduce the content in any medium, as long as the original authors and source are cited. For any reuse, redistribution, or reproduction of a work, users must clarify the license terms under which the work was produced.

## I. 서론

21세기에 들어서 전자기기의 발전과 디지털화가 급격히 진행되고 있다. 이러한 관점에서 교육 현장에서도 학생들의 디지털 장비를 다루는 능력과 컴퓨팅 사고를 함양시키는 것이 강조되고 있다. 일례로 2022 개정 교육과정에서는 디지털 도구를 사용하여 과학 과목의 탐구 활동을 수행할 것을 제시하고 있다.

새롭게 강조되는 현대적 수업의 재료로서 가장 주목받는 것은 아두이노이다. 아두이노는 오픈소스 하드웨어의 일종으로, 마이크로컨트롤러 보드를 기반으로 한 소프트웨어 및 장치 전반을 일컫는다. 가격이 저렴하고, 다양한 종류의 실험에 응용 가능한 동시에 다루기 간편하다는 장점이 있어 컴퓨팅 수업에 대한 학생들의 진입장벽을 낮춰준다[1]. 또한, 컴퓨터와 아두이노의 메인보드를 연결하여 학생들이 코딩한 프로그램을 직접 사용해볼 수 있다. 디지털 화면을 통해서 센서가 감지한 물리량이나 프로그램의 동작 과정을 실시간으로 시각화할 수도 있다. 이러한 장점은 초·중학교 수업 현장에서 학습 흥미도와 성취도 영역에 효과적으로 작용한다[2]. 이처럼 궁극적으로 학생들의 창의적 문제해결력 향상에 기여하므로 아두이노의 사용은 현대적 수업의 목적에 부합한다[3].

실제로 아두이노는 다양한 과목의 수업에 적용되고 있다. 화학공학 수업에서 물질의 3D형태를 구현하는 것을 보조하거나[4], 천문학 수업에서 별자리를 구현하거나[5], 물리학 수업에서 수식으로 배운 고전물리 이론을 실험에 옮겨 보거나[6], SW(소프트웨어) 교육 수업에서 화염감지 모듈을 직접 만들어 보는 등[7] 다양한 선례가 존재한다. 그러나 중등교육의 지구과학 과목에 적용된 사례는 앞서 소개한 다른 과목들에 비해 상대적으로 적다. 해당 교과과정 중, 아두이노를 적용해볼 대상으로서 2022 교육과정에 디지털 도구의 사용이 권장된 탐구실험 중 하나인 구름 발생 실험을 선택했다. 기존의 교수학습에서 구름 발생 실험은 전통적인 아날로그 실험장치로 수행되고 있었는데, 이 실험 장치에는 변화하는 물리량을 눈으로 볼 수 없다는 단점이 있다. 아두이노를 사용하면 이러한 단점이 개선될 것이라 예상하였다. 따라서 본 연구는

기존의 아날로그 구름발생 실험 장치에 아두이노를 접목한 새로운 모델을 개발하였고, 이 모델을 이용해 실험을 수행하고 검토함으로써 교육 현장에의 적용이 문제가 없는지 검증하였다. 이 연구를 통해 앞으로 구름 발생 실험 수업에서 학생들의 학습 효과를 더욱 증진시킬 수 있을 것으로 기대한다.

## II. 연구 방법

### 2.1. 연구 소재

2022 개정 교육과정 중 과학과교육과정 [9과17-03]에서 명시하는 “상대습도, 단열 팽창 및 응결 현상의 관계를 이해하고, 구름의 생성과 강수 과정을 설명할 수 있다.”와 해당<탐구 활동>에 있는 “디지털 탐구 도구를 이용하여 구름 생성 과정 실험하기”를 재현해 보고자 한다. 실험 모델은 2015년 개정 교육과정 중학교 3학년 과학책에 등장하는 간이 기압 장치와 패트병을 활용한 구름 생성 실험을 사용하였다(Fig. 1).



70 II. 기압과 날씨  
Fig. 1. 중학교 3학년 과학 교과서 2단원 기권과 날씨 중 구름 발생 실험 장치(미래엔)

2.2. 아두이노(Arduino)와 시각화 프로그램

본 연구에서는 디지털 탐구 도구로 아두이노(Arduino)를 활용하였다. 아두이노는 오픈 소스를 기반으로 한 개발 플랫폼으로, 간편한 프로그래밍과 데이터 연동이 가능한 마이크로컨트롤러이다[8]. 저렴한 가격과 높은 확장성을 갖추고 있어 학교 현장뿐 아니라 다양한 연구 환경에서 활용도가 높다. 본 실험에서는 아두이노 우노 보드와 DHT11, BMP280 센서를

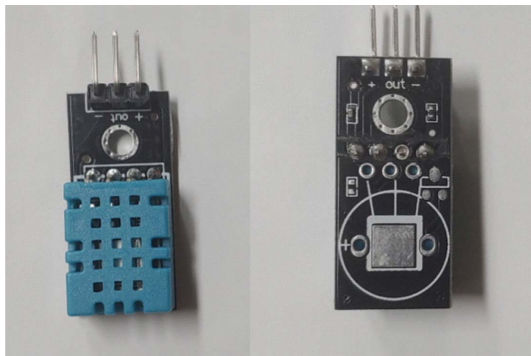


Fig. 2. 왼쪽부터 DHT11(온습도 센서), BMP280(온도, 기압 센서)

Table 1. 아두이노와 센서의 회로도

| 센서     | 센서 핀       | 아두이노 핀  |
|--------|------------|---------|
| DHT11  | VCC        | 5V      |
|        | GND        | GND     |
|        | DATA       | 디지털 핀 2 |
| BMP280 | VCC        | 3.3V    |
|        | GND        | GND     |
|        | SDI (MOSI) | 핀11     |
|        | SDO (MISO) | 핀12     |
|        | SCK        | 핀13     |
|        | CS         | 핀10     |

이용하여 데이터 수집과 시각화를 수행하였다.

실험과 관련된 데이터를 얻기 위해서 아두이노 우노 보드와 DHT11 센서, BMP280 센서를 사용하였다 (Fig. 2). 해당 센서는 각각 대기중 온도와 습도, 대기 압과 온도를 측정할 수 있는 장치이다. 아두이노 회로도 구성에 대해서는 Table 1에, 센서의 감도 및 응답시간은 Table 2에 정리 해주었으며 실제로 구성된 회로도 구성은 사진은 Fig. 3이다.

DHT11 센서의 VCC 센서 선은 아두이노 보드의 5V 핀에 연결한다. GND 선은 GND 핀에 연결하며 DATA 선은 디지털 핀 2번에 연결하였다. BMP280 센서의 VCC 선은 3.3V 핀에 연결한다. GND 선은 GND 핀에 연결, SDI선은 11번 핀, SDO선은 12번 핀, SCK 선은 13번 핀, CS 선은 10번 핀에 연결하여 회로도를 구성하였다. 회로 구성시 주의 사항으로는 BMP280 센서가 3.3V에서만 동작하므로 반드시 3.3V 핀에 연결해야 한다. 만약 5V 핀에 연결한다면 센서가 손상될 수 있다.

DHT11센서는 온도값과 습도값을 측정한다. 감도는



Fig. 3. 구름생성실험 장치에 센서와 우노보드를 연결한 사진

Table 2. DHT11과 BMP280 센서의 감도와 응답시간

| 센서     | 측정값 | 감도(분해능) | 정확도    | 응답시간     |
|--------|-----|---------|--------|----------|
| DHT11  | 온도  | 1°C     | ±2°C   | 약 5~10초  |
|        | 습도  | 1% RH   | ±5% RH | 약 6~10초  |
| BMP280 | 온도  | 0.01°C  | ±1°C   | 약 1~2초   |
|        | 기압  | 0.18 Pa | ±1 hPa | 약 7.5 ms |

```

1  #include <SPI.h>
2  #include <Adafruit_Sensor.h>
3  #include <Adafruit_BMP280.h>
4  #include <DHT.h>
5
6  #define DHTPIN 2          // DHT11 센서를 연결할 핀
7  #define DHTTYPE DHT11   // DHT11 센서 타입
8
9  DHT dht(DHTPIN, DHTTYPE);
10
11 // BMP280 SPI 핀 설정 (CS 핀은 10번으로 설정)
12 #define BMP_CS 10
13 Adafruit_BMP280 bmp(BMP_CS);
14
15 void setup() {
16   Serial.begin(9600);
17
18   // DHT11 초기화
19   dht.begin();
20
21   // SPI 통신을 통한 BMP280 초기화
22   if (!bmp.begin()) {
23     Serial.println("BMP280 초기화 실패");
24     while (1);
25   }
26 }
27
28 void loop() {
29   // DHT11에서 온도와 습도 읽기
30   float humidity = dht.readHumidity();
31   float temperatureDHT = dht.readTemperature(); // 섭씨 온도
32
33   // BMP280에서 온도와 기압 읽기 (SPI 통신)
34   float temperatureBMP = bmp.readTemperature();
35   float pressure = bmp.readPressure() / 100.0; // 기압을 hPa로 변환
36
37   // 시리얼로 각 센서의 데이터 전송
38   Serial.print(temperatureBMP); // BMP280 온도
39   Serial.print(",");
40   Serial.print(pressure); // BMP280 기압
41   Serial.print(",");
42   Serial.print(temperatureDHT); // DHT11 온도
43   Serial.print(",");
44   Serial.println(humidity); // DHT11 습도
45
46   delay(1000); // 1초 대기
47 }

```

Fig. 4. 아두이노 소스코드

두 물리량 각각 1°C와 1% RH를 가진다. 응답시간은 약 5~10초 이다. BMP280 센서는 온도와 기압값을 측정하며 0.01°C, 0.18의 감도를 가지고 있다. 응답 속도는 온도의 경우 1~2초, 기압은 7.5 milli 초를 가진다.

DHT11 센서는 변화하는 물리량에 응답하는 시간이 10초까지도 소요되는 반면 BMP280 센서는 비교적 빠른 응답속도를 가지고 있다. 아두이노와 센서의 회로도 구성을 마치면 아두이노가 일을 할 수 있게 소스코드를 작성해주어야 한다. 아두이노는 통합개발환경을 통하여 사람이 입력한 C언어 기반의 소스코드를 컴파일 하여 컴퓨터언어로 변환을 해주고 보드에 적재할 수 있다. Fig. 4는 DHT11 센서와 BMP280 센서에서 감지한 온습도 및 기압 값을 1초 단위로 받아볼 수 있는 소스코드이다. 해당 소스코드를 활용하면 각 센서에서 온도와 습도, 온도와 기압 값을 1초 단위로 얻을 수 있다. 이를 그래프의 형태로 시각화 하기 위해서는 Excel, Python 등을 이용할 수 있는데, 이번 실험에서 사용한 BMP280 센

서의 SPI 데이터 전송방식에서는 센서가 감지한 물리값을 Excel로 전송하기 어렵다. 따라서 본 연구에서는 센서에서 얻은 데이터 값을 Python을 활용하여 전송 받고 그래프의 형태로 시각화 하였다. Fig. 5에서 사용한 Python코드를 확인할 수 있다.

### III. 연구 결과

#### 3.1. 전통적 구름 생성 실험

전통적 구름 생성 실험은 공기 덩어리가 상승하며 단열팽창이 발생하는 대기의 구름 형성 과정을 재현한다. 본 실험에서는 PET병에 공기를 밀폐하여 압축한 뒤, 뚜껑을 열어 순간적으로 압력을 낮추어 단열팽창을 유도하였다. Fig. 6는 해당 실험 과정을 시각화한 결과를 보여준다. 이러한 실험을 통해 내부 온도 변화는 관찰 가능하지만, 기압 변화는 직접적으로 측정하기 어렵다는 한계가 있었다.

#### 3.2. 아두이노 센서를 활용한 구름생성과정 실험

아두이노 센서를 활용한 실험에서는 전통적 실험에 사용된 PET병을 수정하여 DHT11 및 BMP280 센서를 부착한 후 밀폐하였다. 센서는 아두이노 보드와 연결되어 USB 포트를 통해 PC와 연동되었으며, 실험 과정에서 온도, 습도, 기압 변화를 실시간으로 측정하였다. 수집된 데이터는 Python을 이용해 시각화하였으며, 그래프(Fig. 7)는 압력 감소에 따른 단열팽창과 온도 감소의 상관성을 명확히 보여준다. ‘BMP280 Pressure’ 값이 1000hPa 부근으로 감소하는 시각과 ‘BMP280 Temperature’ 값이 감소하는 시각이 일치한다. 이는 해당 구름생성실험 모델에서 압력 감소에 의한 단열팽창의 결과를 실시간으로 시각화 할 수 있다는 점을 시사한다. DHT11 센서는 BMP280 센서보다 늦은 응답속도를 보인다. 다만 ‘DHT11 Humidity’ 값이 십 초 경과 후 다시 원습도로 감소하는 경향을 확인할 수 있다. Fig. 8에서 단열압축을 진행함에 따라 온도와 기압, 습도 값이 변화하는 것을 확인 가능하며 응답속도가 빠른 BMP280 센서의 데이터는 초기 온도 15.95°C, 기압 1017.19 hPa에서 17.2°C, 1368.15 hPa 까지 상승하는 것을 볼 수 있다. 실험 과정에서 압력을 빠르게 감소시키며 단열팽창을 유도

```

import serial
import matplotlib.pyplot as plt
import matplotlib.dates as mdates
import pandas as pd
from datetime import datetime
import time

# 시리얼 포트 설정
ser = serial.Serial('COM7', 9600)
time.sleep(2)

# 데이터 저장을 위한 pandas DataFrame 설정
df = pd.DataFrame(columns=["Timestamp", "Temperature_BMP", "Pressure_BMP", "Temperature_DHT", "Humidity_DHT"])

# 그래프 설정
plt.ion()
fig, ((ax1, ax2), (ax3, ax4)) = plt.subplots(2, 2, figsize=(12, 8))

# 그래프 초기화
temperature_bmp_line, = ax1.plot([], [], label="BMP280 Temperature (°C)", color='tab:blue')
pressure_bmp_line, = ax2.plot([], [], label="BMP280 Pressure (hPa)", color='tab:red')
temperature_dht_line, = ax3.plot([], [], label="DHT11 Temperature (°C)", color='tab:green')
humidity_dht_line, = ax4.plot([], [], label="DHT11 Humidity (%)", color='tab:orange')

# 그래프 설정
for ax in [ax1, ax2, ax3, ax4]:
    ax.xaxis.set_major_formatter(mdates.DateFormatter('%H:%M:%S'))

try:
    while True:
        if ser.in_waiting > 0:
            line = ser.readline().decode('utf-8').strip()
            print("Received: {line}")
            try:
                temperature_bmp_val, pressure_bmp_val, temperature_dht_val, humidity_dht_val = map(float, line.split(","))
            except ValueError:
                continue

            # 현재 시간 기록
            timestamp = datetime.now()

            # 데이터를 pandas DataFrame에 추가
            new_data = pd.DataFrame([[timestamp, temperature_bmp_val, pressure_bmp_val, temperature_dht_val, humidity_dht_val]],
                                   columns=["Timestamp", "Temperature_BMP", "Pressure_BMP", "Temperature_DHT", "Humidity_DHT"])
            df = pd.concat([df, new_data], ignore_index=True)

            # 실시간 그래프 업데이트
            temperature_bmp_line.set_xdata(df["Timestamp"])
            temperature_bmp_line.set_ydata(df["Temperature_BMP"])

            pressure_bmp_line.set_xdata(df["Timestamp"])
            pressure_bmp_line.set_ydata(df["Pressure_BMP"])

            temperature_dht_line.set_xdata(df["Timestamp"])
            temperature_dht_line.set_ydata(df["Temperature_DHT"])

            humidity_dht_line.set_xdata(df["Timestamp"])
            humidity_dht_line.set_ydata(df["Humidity_DHT"])

            # 축 범위 동적 업데이트
            for ax in [ax1, ax2, ax3, ax4]:
                ax.relim()
                ax.autoscale_view()

            # X축 레이블 회전
            plt.setp(ax1.xaxis.get_majorticklabels(), rotation=45, ha='right')
            plt.setp(ax2.xaxis.get_majorticklabels(), rotation=45, ha='right')
            plt.setp(ax3.xaxis.get_majorticklabels(), rotation=45, ha='right')
            plt.setp(ax4.xaxis.get_majorticklabels(), rotation=45, ha='right')

            # 그래프 갱신
            plt.draw()
            plt.pause(1)

except KeyboardInterrupt:
    print("프로그램 종료")

```

Fig. 5. Python 데이터 시각화 코드

하였을 때 온도값 0.16°C, 기압값 약 350 hPa 만큼 급격히 하강하는 것을 확인하였다.

본 연구에서는 DHT11 및 BMP280 센서를 활용하여 구름 생성 실험의 데이터를 실시간으로 수집하고 시각화하였다. 실험 결과, BMP280 센서는 빠른 응답 속도로 압력 및 온도 변화를 즉각적으로 측정할 수 있었으나, DHT11 센서는 느린 응답 시간으로 인해

데이터 시각화에 한계가 있었다. 단일과정과 같이 물리량이 급변하는 현상을 측정할 때 센서의 응답속도를 충분히 고려하여 센서를 선정할 것을 제안한다.

## IV. 결 론

이 연구에서는 기존의 구름 발생 실험 장치에

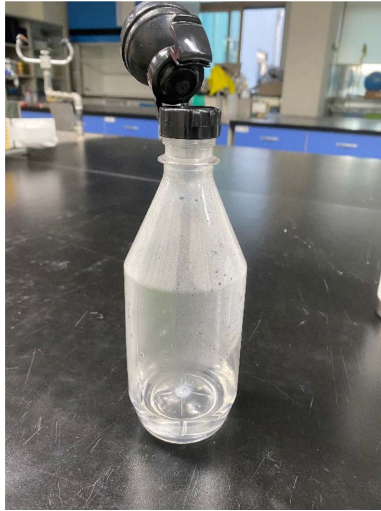


Fig. 6. 구름생성실험모델 수증기생성

아두이노를 결합한 새로운 모델을 개발하였다. 전통적 구름 발생 실험에서, 내부의 습도 변화는 육안으로 관찰이 가능했으나 온도와 기압의 변화는 볼 수 없었다. 새로운 실험 모델에서는 전통적 실험과 같이 육안을 통한 습도 변화의 관찰이 가능했다. 더불어 아두이노에 연결된 디지털 화면을 통해 기존에는 확인할 수 없었던 온도와 기압, 그리고 습도가 실시간으로 변하는 모습을 관찰할 수 있었다. 디지털 화면으로 도출된 물리량은 아날로그 실험 수업에서 사용

| Timestamp | Temperature_BMP | Pressure_BMP | Temperature_DHT | Humidity_DHT |
|-----------|-----------------|--------------|-----------------|--------------|
| 34:44.7   | 15.95           | 1017.19      | 15.2            | 35           |
| 34:45.9   | 16.04           | 1017.23      | 15.2            | 35           |
| 34:47.0   | 16.05           | 1017.22      | 15.2            | 35           |
| 34:48.1   | 16.06           | 1017.25      | 15.2            | 35           |
| 34:49.2   | 16.07           | 1017.24      | 15.2            | 35           |
| 34:50.2   | 16.07           | 1017.2       | 15.2            | 35           |
| 34:51.3   | 16.08           | 1017.34      | 15.2            | 35           |
| 34:52.4   | 16.1            | 1055.1       | 15.2            | 35           |
| 34:53.5   | 16.16           | 1088.99      | 15.2            | 35           |
| 34:54.5   | 16.21           | 1138.97      | 15.2            | 35           |
| 34:55.6   | 16.29           | 1167.94      | 15.2            | 35           |
| 34:56.7   | 16.35           | 1204.99      | 15.2            | 35           |
| 34:57.8   | 16.42           | 1220.79      | 15.2            | 35           |
| 34:58.9   | 16.47           | 1244.84      | 15.2            | 35           |
| 34:59.9   | 16.54           | 1271.6       | 15.2            | 35           |
| 35:01.0   | 16.57           | 1298.62      | 15.2            | 35           |
| 35:02.1   | 16.64           | 1303.89      | 15.2            | 35           |
| 35:03.2   | 16.67           | 1319.56      | 15.2            | 35           |
| 35:04.3   | 16.72           | 1333.49      | 15.3            | 36           |
| 35:05.3   | 16.75           | 1336.47      | 15.3            | 36           |
| 35:06.4   | 16.79           | 1355.12      | 15.3            | 36           |
| 35:07.4   | 16.82           | 1357.27      | 15.3            | 36           |
| 35:08.5   | 16.85           | 1371.11      | 15.3            | 36           |
| 35:09.6   | 16.88           | 1370.82      | 15.3            | 36           |
| 35:10.6   | 16.9            | 1383.78      | 15.3            | 37           |
| 35:11.7   | 16.94           | 1385.84      | 15.3            | 37           |
| 35:12.7   | 16.96           | 1385.24      | 15.4            | 37           |
| 35:13.8   | 16.98           | 1396.04      | 15.4            | 37           |
| 35:37.3   | 17.19           | 1370.65      | 15.8            | 39           |
| 35:38.4   | 17.19           | 1367.42      | 15.8            | 39           |
| 35:39.5   | 17.2            | 1368.15      | 15.8            | 39           |
| 35:40.5   | 17.04           | 1017.38      | 15.9            | 39           |
| 35:41.6   | 17.07           | 1017.33      | 15.9            | 39           |
| 35:42.7   | 17.08           | 1017.31      | 15.9            | 39           |
| 35:43.8   | 17.07           | 1017.28      | 15.9            | 39           |
| 35:44.8   | 17.07           | 1017.31      | 15.9            | 39           |
| 35:45.9   | 17.06           | 1017.26      | 15.9            | 39           |
| 35:47.0   | 17.06           | 1017.24      | 15.9            | 39           |
| 35:48.0   | 17.07           | 1017.26      | 15.9            | 39           |
| 35:49.2   | 17.07           | 1017.24      | 15.9            | 39           |
| 35:50.2   | 17.07           | 1017.24      | 15.9            | 39           |
| 35:51.4   | 17.08           | 1017.23      | 15.9            | 38           |
| 35:52.4   | 17.08           | 1017.21      | 15.9            | 38           |
| 35:53.5   | 17.08           | 1017.25      | 15.9            | 38           |
| 35:54.6   | 17.09           | 1017.23      | 15.9            | 38           |
| 35:55.7   | 17.1            | 1017.22      | 16              | 38           |
| 35:56.7   | 17.09           | 1017.2       | 16              | 38           |
| 35:57.8   | 17.1            | 1017.18      | 16              | 37           |

Fig. 8. 아두이노 센서로 측정된 물리량. 왼쪽부터 시간, BMP 온도, BMP 기압, DHT 온도, DHT 습도값

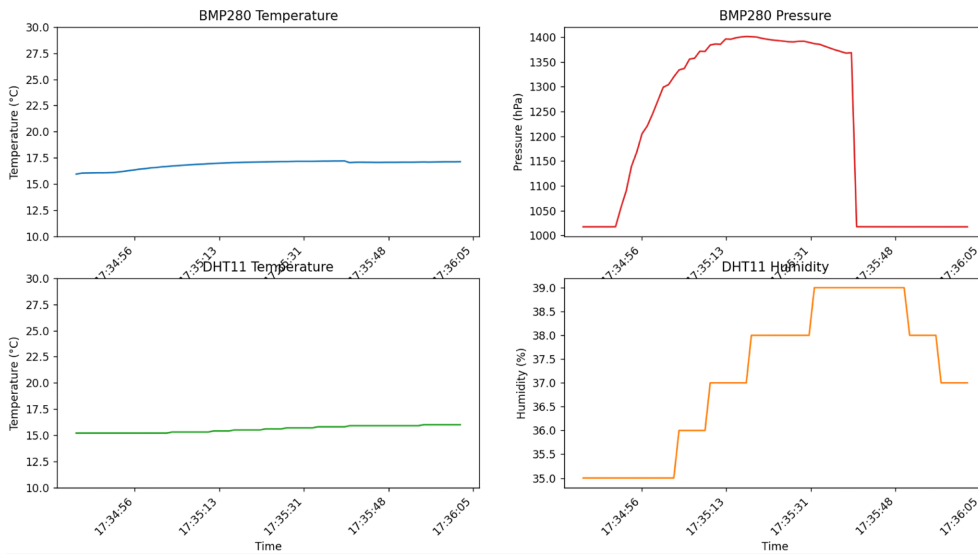


Fig. 7. Python에서 그래프로 시각화 한 자료. 왼쪽 위부터 BMP280 센서의 온도와 기압, DHT 센서의 온도와 습도

되던 이론과 일치하였다. 따라서 실제 교수학습 현장에서 아두이노를 활용한 구름 발생 실험 모델을 활용하는 것은 문제가 없는 것으로 판단된다. 이 새로운 실험 수업이, 물리량의 시각화가 가져다주는 교육적 효과와 피지컬 컴퓨팅 교육과 결합된다면 더 큰 교수학습 효과를 거둘 수 있을 것으로 기대된다.

다만 코딩 과정에서 AI 도구를 이용할 경우, 학생들이 직접 코드를 짜 보며 컴퓨팅 사고력을 기를 기회가 제한될 가능성이 있다. AI를 사용함으로써 절약한 시간을 토론이나 발표 등에 활용할 수 있다는 장점도 있으나 상기한 단점을 최소화하기 위한 새로운 교수학습법에 대한 추가 연구가 필요하다. 또한 아두이노 실험 장치를 만들 때, 센서마다 감도 차이가 존재하므로 실험 목적에 맞는 합리적인 선택이 권장된다.

### 참고문헌

- [1] 장미화, 이미형, 이선미, 조윤수, 조경숙, 아두이노를 활용한 고등학생 대상 환경교육 프로그램의 개발 및 적용: 미세먼지측정기 만들기를 중심으로, 한국 환경교육학회 학술대회 자료집 (2019).
- [2] 이은상, 초, 중등학교 아두이노 적용 교육의 효과에 대한 메타 분석, 한국컴퓨터교육학회 학술발표대회 논문집 (2020).
- [3] 박태정, 창의적 문제해결 기반 피지컬컴퓨팅 교육 프로그램 설계원리 개발 및 적용, 교육공학연구, **34**(3), 817-847 (2018).
- [4] J. J. Elmer and D. A. Kraut, 3-D Printing and Arduino in the Chemical Engineering Classroom: Protein Structures, Heat Exchangers, and Flow Cells, 2018 ASEE Annual Conference & Exposition (2018).
- [5] F. D. Giacomo and M. Sandri, Educational activities with Arduino to learn about astronomy, 4th Symposium on Space Educational Activities Barcelona (2022).
- [6] K. John, Using an Arduino in physics teaching for beginners, Physics Education, **53**(6), 063007 (2018).
- [7] 유종훈, 김용환, 양창은, 장명호, 김희주, 명노영, 김동정, 유현창. 아두이노 기반 피지컬컴퓨팅을 활용한 SW 개발 교육. 한국컴퓨터교육학회 학술발표대회논문집 (2015).
- [8] 장봉석, 중고등학생 대상 아두이노 활용 교육의 효과에 대한 메타분석, 산업융합연구, **19**(3), 61-65 (2021).

[1] 장미화, 이미형, 이선미, 조윤수, 조경숙, 아두이노를 활용한 고등학생 대상 환경교육 프로그램의 개발