ISSN 3022-8719

Journal of Science and Science Education | Vol. 50, No. 2, p. 103-116, December 2025

https://doi.org/10.23335/JSSE.2025.50.2.103

중학생 라디오존데 제작 활동을 통한 기상 데이터 분석

황호철[1] · 이수경[2*] · 송규정[2]

[1]전주우림중학교, 전주 55065

[2]전북대학교 사범대학 과학교육학부, 전주 54896

Analysis of Atomospheric Data through a Middle School Student-Built Radiosonde Project

Ho Cheol Hwang[1], Su Kyeong Lee[2]*, and Kyu Jeong Song[2]

[1]Jeonju Woolim Middle School, Jeonju 55065, Korea

[2]Division of Science Education, Jeonbuk National University, Jeonju 54896, Korea

초 록: 본 연구는 중학생이 제작한 라디오존데를 활용하여 실제 대기 데이터를 수집·분석하고, 대기 구조 학습을 탐구 중심 활동으로 확장하고자 수행되었다. 19명의 학생이 설계, 제작, 회수 및 데이터 분석 전 과정에 참여하였다. 라디오존데는 약 16 km까지 상승하여 온도, 습도, 기압 데이터를 수집하였고, 기압을 고도로 변환하여 분석하였다. 1차 실험 실패 이후 데이터 이중 저장 시스템을 도입하여 2차 실험에서 약 4,500초의 데이터를 확보하였다. 상승 속도는 6.19 m/s, 하강 속도는 7.98 m/s로 나타났으며, 대류권 내 온도 감소 경향이 확인되었다. 본 연구는 공학 설계 기반 활동이 이론과 실제 관측을 연결하는 효과적인 과학교육 방안임을 시사한다.

중심어: 라디오존데, 대기 구조, 탐구 기반 학습, 공학 설계 과정, 피지컬 컴퓨팅

Abstract: This study was conducted to expand the learning of atmospheric structure into an inquiry-based activity by collecting and analyzing real atmospheric data using a radiosonde designed and constructed by middle school students. Nineteen students participated in the entire process, including design, fabrication, recovery, and data analysis. The radiosonde ascended to approximately 16 km and collected temperature, humidity, and pressure data. Atmospheric pressure was converted into altitude for analysis. After the failure of the first experiment, a dual data storage system was implemented, enabling the acquisition of approximately 4,500 seconds of data during the second experiment. The ascent and descent rates were 6.19 m/s and 7.98 m/s, respectively, and the decreasing temperature trend within the troposphere was confirmed. The findings suggest that engineering design–based activities are an effective science education approach for connecting theoretical knowledge with real-world observations.

Keywords: Radiosonde, Atmospheric Structure, Inquiry-Based Learning, Engineering Design Process, Physical Computing

*Corresponding Author: Su Kyeong Lee
Phone: +82-(0)63-270-2803
E-mail: sklee92@jbnu.ac.kr

I. 서 론

대기 구조와 기상 요소는 중학교 및 고등학교 과학 교육과정에서 핵심적으로 다루어지는 내용으로, 대류권과 성층권의 구조, 고도에 따른 기압과 온도의 변화, 대류권계면의 특징 등을 이론적으로 학습한다[1]. 그러나 이러한 학습은 대부분 교과서와 모의 자료에 기반한 간접 경험에 머무르는 경우가 많아, 실제 대기에서 나타나는 물리적 현상을 체험적으로 이해하기에는 한계가 있다.

라디오존데(radiosonde)는 지상에서부터 상층으로 상승하면서, 기압, 상대습도, 기온 등의 기상 요소를 측정하는 센서를 탑재하고 측정한 정보를 지상으로 전송하는 장치로, 기상 관측 및 예보에 핵심적인 역할을 수행한다[2,3]. 기상청에서도 매일 정해진 시각에 라디오존데를 띄워 대기의 연직 구조를 분석하고 있다. 그러나 이러한 전문 장비는 교육 현장에서 직접 활용하기 어렵고, 학생이 제작 과정에 참여하는 사례는 제한적이다.

아두이노는 오픈 소스를 기반으로 한 단일 보드 마이크로컨트롤러로 완성된 보드, 관련 개발 도구 및 환경으로[4], 최근 저비용 관측 장비 제작이 가능해졌으며 이는 학교 현장에서 공학 설계 기반 탐구 활동을 구현할 수 있는 가능성을 제공한다. 하지만 고고도 비행을 통한 실제 대기 데이터 수집은 기체 회수, 데이터 저장 안정성, 센서 신뢰도 등 여러 기술적 난제를 포함하고 있어 체계적인 설계와 반복적 개선 과정이 요구된다[4,5].

이에 본 연구에서는 학생들이 직접 라디오존데를 설계·제작하고, 약 16 km 고도까지의 실제 대기 데이터를 수집·분석함으로써 교과서 이론과 실제 대기 환경을 비교·탐구하고자 하였다. 특히 상승 및 하강 과정에서의 고도-온도 관계를 분석하고, 센서의 응답 지연과 단열 효과가 데이터에 미치는 영향을 해석하는 것을 연구의 주요 목적으로 설정하였다.

본 연구는 단순한 체험 활동을 넘어, 공학적 설계 과정과 물리적 데이터 분석을 통합한 탐구 중심 과학교육 모델을 제시하고자 한다.

II. 연구 방법

본 연구에서는 고도에 따른 대기 변화 특성을 분석하기 위하여 온도, 기압, 습도, 성층권 촬영 영상 데이터를 수집하였다. 온도, 기압, 습도 데이터는 라디오존데에 탑재된 BME688 센서를 통해 수집하였다. 기압 데이터는 표준 대기 변환식을 이용하여 고도 데이터로 환산하였다. 영상 데이터는 성층권 하부의 대기 상태를 시각적으로 확인하기 위하여 액션캠을 이용해 촬영하였다. 라디오존데는 기상 풍선을 이용하여 고고도로 상승한 뒤 낙하산을 통해 지상으로 하강하도록 설계하였다. 데이터 확보를 위해 착륙 지점을 GPS 기반으로 추적한 후 현장에서 라디오존데를 직접 회수하여 SD카드 및 스마트폰 백업 데이터를 활용한 저장 장치로부터 온도, 습도, 기압 데이터를 수집하였으며, 기압 데이터는 기압-고도 변환식을 이용하여 고도로 환산하였다. 기압-고도 변환식은 정역학 평형식, 이상기체 상태방정식, 대류권에서의 선형 기온감률 가정식을 통해 유도 된다. 먼저 정역학 평형식은 식 (1)과 같이 나타낼수 있다[6].

$$\frac{dP}{dz} = -\rho g \tag{1}$$

이상기체 상태 방정식은 식 (2)와 같고,

$$P = \rho RT \tag{2}$$

대류권에서 선형 기온 감률을 가정하면 식 (3)과 같다.

$$T = T_0 - Lz \tag{3}$$

이때 ISA조건에서 L, T_0, P_0는 $L = 0.0065$ K/m(표준기온감률), $T_0 = 288.15$ K, $P_0 = 1013.25$ hPa로 가정하고, 식 (1)–(3)을 연립하여 적분하면 식 (4)와 같다.

$$h = \frac{L}{T_0}\left[1 - \left(\frac{P}{P_0}\right)^{\frac{RL}{g}}\right] \tag{4}$$

이때

$$\frac{T_0}{L} \approx 44330m \tag{5}$$

$$\frac{RL}{g} \approx 0.1903 \tag{6}$$

으로 근사하면 최종적으로 기압 고도 변환식은 식 (7)과 같이 나타낼 수 있다.

$$h = 44330\left(1-\left(\frac{P}{P_0}\right)^{0.1903}\right) \tag{7}$$

2.1. 연구 절차

연구 참여자는 표 1과 같이 중학교 1~3학년 학생 19명이며, 1학년 1명, 2학년 6명, 3학년 12명으로 구성되었으며, 남학생 10명, 여학생 9명이 참여하였다.

라디오존데 제작 활동을 위해 자료 조사를 먼저 시작하였으며 라디오존데 제작에 필요한 아두이노 보드, 센서 구입, 촬영을 위한 방안 등을 모색하였다. 1차 실험에서 데이터 수집에 실패하여 1차 실험의 과정을 수정하여 2차 실험을 시행 하였으며 수업 절차는 표 2와 같다.

2.2. 자료 수집 및 분석

온도, 습도, 기압 데이터는 피지컬 컴퓨팅 보드(Arduino 및 ESP32)와 BME688 센서를 이용하여 일정 시간 간격으로 자동 측정되도록 하였다. 수집된 기압 데이터는 표준 대기식에 따라 고도로 변환하였으며, 수집된 데이터를 바탕으로 시간-고도, 시간-온도, 시간-습도, 고도-온도, 고도-습도를 분석하였다.

시간-고도 그래프의 기울기를 이용하여 상승 속도와 하강 속도를 계산하였으며, 상승 구간과 하강 구간을 분리하여 분석하였다. 또한 고도-온도 관계를 통해 대류권 및 성층권 하부의 온도 변화 특성을 확인하였다. 분석 과정에서는 교과서에 제시된 표준 대기 모형과 실험 데이터를 비교함으로써 이론과 실제 관측값의 차이를 해석하였다.

2.2.1. 라디오존데 제작 자료 수집

1~2차시에는 협업 보드를 활용하여 라디오존데 제작을 위한 사전 자료 조사를 실시하였다. 학생들은 유튜브, 블로그 등 다양한 온라인 자료를 통해 라디오존데 제작 성공 사례를 분석하고, 라디오존데의

표 1. 연구 참여자

	남	여	전체(명)
1학년		1	1
2학년	3	3	6
3학년	7	5	12

그림 1. 라디오존데 모식도[5].

표 2. 라디오존데 프로그램

단 계	차시	주 제
제작 전 자료 수집	1차시~2차시	라디오존데 제작 자료 조사
	3차시~4차시	라디오존데 제작을 위한 피지컬 컴퓨팅
1차 라디오존데 제작 및 띄우기	5차시~6차시	라디오존데 낙하산 제작
	7차시~8차시	라디오존데 1차 띄우기 (충남 논산 착륙)
	9차시~10차시	라디오존데 1차 평가회
2차 라디오존데 제작 및 띄우기	11차시~12차시	라디오존데 2차 제작
	13차시~14차시	라디오존데 2차 띄우기(경북 영천 착륙)
	15차시~16차시	라디오존데 2차 평가회
	17차시~18차시	라디오존데 데이터 우드락 제작

표 3. 1차 라디오존데의 구성요소

목적	준비물
데이터 수집(온도, 습도, 기압)	아두이노보드, BME688센서, SD카드 모듈
영상 데이터 수집	스마트폰(갤럭시 S21)
GPS 데이터 수집	스마트폰(아이찾기 어플)
기상 풍선	1200 g, 약 2 kg 중량 탑재 가능
He 가스	47 L 헬륨가스 1통 주입
낙하산	45인치 페이로드
전원 공급	2000 mAh 보조배터리
라디오존데 중량(낙하산 포함)	약 1.1 kg
단열	스티로폼 상자(아이스크림 케이크)

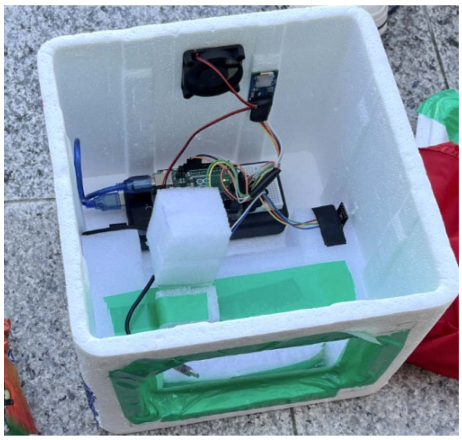

그림 2. 라디오존데 내부의 모습.

그림 3. 낙하산의 연결 모습.

구조와 형태에 대해 토의하였다. 이를 바탕으로 라디오존데의 외형과 내부 구성 방안을 협의하였으며 그림 1은 라디오존데의 모식도 이다.

또한 인터넷 포털을 통해 필요한 부품과 장비 목록을 정리하고, 띄울 시기의 기상 조건을 고려하여 북태평양 고기압의 영향을 받아 바람이 적은 여름철에 1차 실험을 계획하였다.

2.2.2. 1차 라디오존데 설계

1차 라디오존데의 구성 요소는 표 3과 같으며, 아두이노(Arduino) 보드에 환경 센서 BME688을 연결하여 온도, 습도, 기압 데이터를 측정하도록 설계하였다. 그림 2와 그림 3은 제작한 라디오존데 내부와 낙하산 모습이다. 라디오존데의 데이터 저장을 위해 SD카드 모듈을 추가 장착하였으며, 데이터 수집 프로그램은 부록 1에 제시하였다. 라디오존데 회수를 위해 GPS 기능이 활성화된 스마트폰을 내부에 탑재하고, 위치 추적 애플리케이션인 아이찾기 앱을 활용하여 착륙 지점을 확인하도록 하였다. 영상 데이터 또한 동일한 스마트폰을 이용하여 촬영 및 저장하였다.

착륙 시 라디오존데의 충격을 최소화하고 안전하게 회수하기 위해 45인치 낙하산을 연결하여 하강 속도를 감소시키도록 설계하였다. 띄우기 하루 전에는 학교 옥상에서 시험 비행을 실시하여 센서 데이터가 정상적으로 수집되는지 확인하고, 기체의 이상 여부를 점검하였다. 라디오존데의 총 중량은 낙하산을 포함하여 약 1.1 kg이었으며, 기상 풍선의 부력을 고려할 때 상승에 무리가 없는 수준이었다.

2.2.3. 1차 실험

1차 라디오존데는 2025년 8월 29일 오후 4시경 띄웠으며, 충청남도 논산 인근에 착륙하였다. 착륙 지점 확인 후 약 4일에 걸쳐 주변 야산을 탐색한 결과 라디오존데를 회수할 수 있었다. 그러나 SD카드에

표 4. 2차 라디오존데의 구성요소

목적	준비물
데이터 수집(온도, 습도, 기압)	GeekblenanoESP32, BME688, SD카드모듈, 블루투스 백업
영상 데이터 수집	액션캠(유프로)
GPS 데이터 수집	스마트폰(아이찾기 어플)
기상 풍선	1200 g, 약 2 kg 중량 탑재 가능
He 가스	47 L 헬륨가스 1통 주입
낙하산	45인치 페이로드
전원 공급	2000 mAh 보조배터리
라디오존데 중량(낙하산 포함)	약 1.2 kg
단열	스티로폼 상자(아이스크림 케이크)

데이터가 정상적으로 저장되지 않았으며, 라디오존데 내부에 탑재한 스마트폰 또한 발열 문제로 인해 영상 촬영이 중단되어 유의미한 데이터를 확보하지 못하였다. 데이터 수집에는 실패하였으나, 라디오존데가 목표 지점까지 안정적으로 비행하고 GPS 기반 위치 추적을 통해 회수가 가능함을 확인하였다. 이는 하드웨어적 설계는 유효하였음을 의미하며, 소프트웨어 및 데이터 저장 체계의 보완을 통해 후속 실험이 가능함을 시사하였다.

그림 4. 라디오존데 2차 실험.

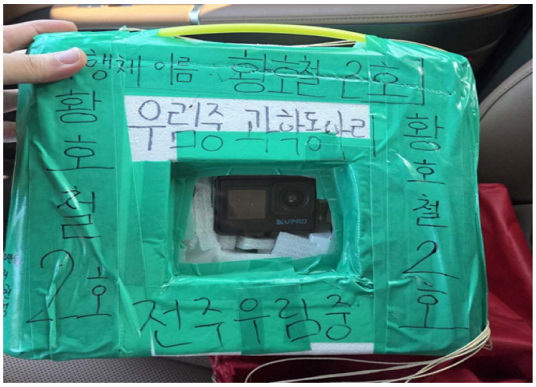

그림 5. 라디오존데 2차 회수.

2.2.4. 2차 라디오존데 설계

1차 실험에서 데이터 수집에는 실패하였으나, GPS 기반 위치 추적을 통해 라디오존데를 성공적으로 회수한 경험은 시스템 개선의 방향을 제시하였다. 이에 따라 2차 실험에서는 데이터 저장 안정성을 확보하는 것을 핵심 목표로 설정하였다. 기존 SD카드 저장 방식에 더하여, 라디오존데 내부의 스마트폰과 블루투스 통신을 이용한 이중 데이터 저장 시스템을 구축하였다. 이를 통해 저장 오류 발생 시에도 데이터를 확보할 수 있도록 하였다.

또한 1차 실험에서 스마트폰의 발열로 인해 영상 촬영이 중단된 점을 반영하여, 2차 실험에서는 별도의 액션캠을 탑재하여 영상 데이터를 수집하도록 설계하였다. 제어 보드는 블루투스 송신 기능이 가능한 Geekble Nano ESP32 보드로 교체하였으며, 환경 센서 BME688은 동일하게 사용하였다. 전원 공급 장치, 낙하산, 헬륨 가스, 기상 풍선은 1차 실험에서 정상 작동이 확인되었으므로 동일한 사양으로 구성하였다.

2차 실험은 실험 당일 서풍이 강하여 1차 실험보다 이동 거리가 증가할 가능성이 우려되었다. 2차 라디오존데의 구성은 표 4에 제시하였으며, 데이터 수집을 위한 프로그램 코드는 부록 2에 제시하였다.

2.2.5. 2차 라디오존데 실험

2차 라디오존데는 2025년 11월 12일 오후 1시경 띄웠으며, 경상북도 영천 인근에 착륙하였다. 2025년 11월 13일 인근 야산에서 라디오존데를 회수하였고, 영상 데이터와 기상 데이터를 모두 확보하였다. 회수된 데이터를 바탕으로 학생들과 함께 본격적인 데이터 분석을 수행할 수 있었다.

III. 결 과

3.1. 시간에 따른 고도, 습도, 온도 분석

분석 결과 비행 시간은 약 4,500초(약 1시간 15분)였으며, 기압 고도 변환식인 식 (7)을 활용한 시간에 따른 온도, 고도, 습도의 변화는 그림 6과 같다.

라디오존데는 최대 약 16 km 고도까지 상승하였으며, 최저 온도는 -30°C까지 감소하였다. 습도 데이터는 측정 당시 구름이 많은 기상 조건의 영향을 받아 고도에 따른 뚜렷한 경향성을 확인하기 어려웠다.

3.2. 고도-시간 그래프를 통한 상승 속력, 하강 속력 분석

고도를 y축, 시간을 x축으로 설정하여 시간에 따른 고도 변화를 분석하였으며 그림 7과 그림 8은 상승, 하강 결과이다. 고도–시간 그래프의 기울기는 속력을 의미하므로, 이를 통해 상승 및 하강 속력을 산출하였다.

그림 7을 바탕으로 상승 구간을 등속 운동으로 가정할 경우, 그래프의 기울기로부터 상승 속력은 6.19 m/s로 계산되었다. 그림 8에서 하강 구간 또한 낙하산 전개 이후 종단속력에 의해 등속 운동을 한다고 가정하면, 기울기로부터 하강 속력은 7.98 m/s로 산출되었다.

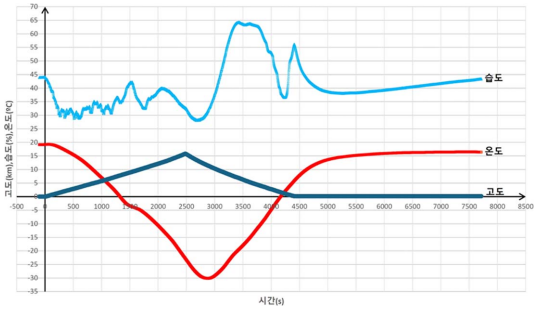

그림 6. 시간에 따른 온도, 습도, 고도 데이터.

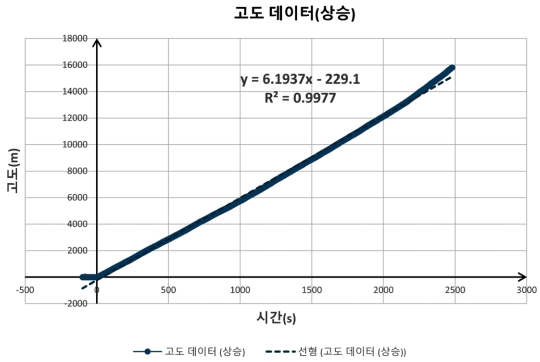

그림 7. 고도-시간 상승 그래프.

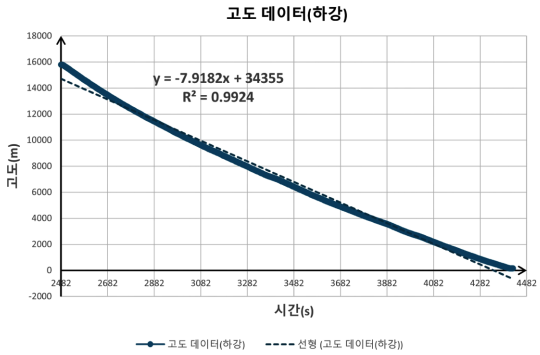

그림 8. 고도-시간 하강 그래프.

상승 속력(6.19 m/s)과 하강 속력(7.98 m/s)을 비교하면 하강 속력이 다소 크게 나타났다. 속력 차이는 센서의 열적 응답 시간에 영향을 줄 수 있으며, 특히 하강 시 외부 환경 변화에 대한 센서 반응 지연이 더 크게 나타날 가능성이 있다.

3.3. 고도-온도 그래프를 통한 고도와 온도의 상관 관계

온도를 x축, 고도를 y축으로 설정하여 고도에 따른 온도 변화를 분석하였으며, 결과는 그림 9와 10과 같다.

그림 9에서 확인할 수 있듯이, 대류권 영역에서는 고도가 증가함에 따라 온도가 감소하는 경향이 나타났다. 그러나 대류권계면(약 10 km) 부근에서도 온도는 지속적으로 감소하였으며, 최대 고도인 약 16 km 부근까지 감소가 이어졌다.

그림 10에서는 약 16 km에서 12 km 구간까지 고도가 감소함에 따라 온도가 증가하는 경향이 나타났으며,

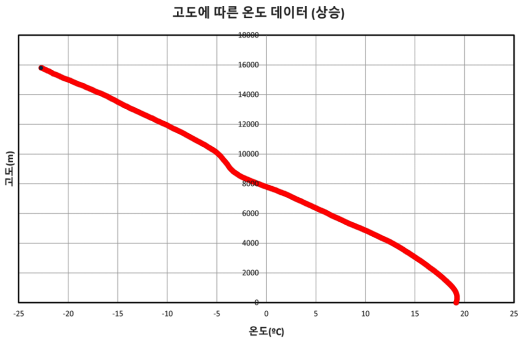

그림 9. 고도-온도 상승 그래프.

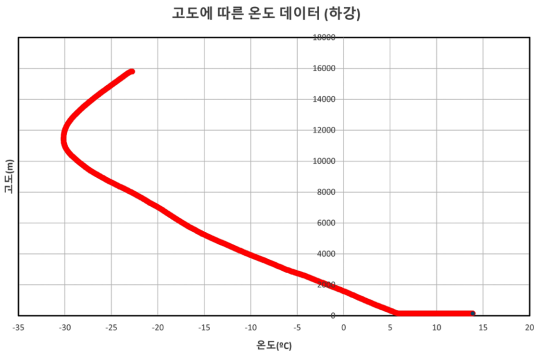

그림 10. 고도-온도 하강 그래프.

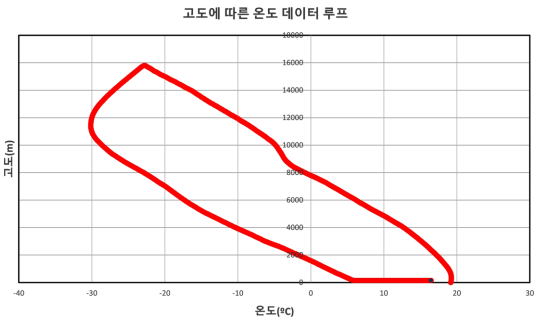

그림 11. 고도-온도 데이터 곡선.

이후 다시 감소하는 양상을 보였다. 이러한 현상은 온도 센서가 라디오존데 내부에 위치하여 단열 효과가 발생하였기 때문으로 판단된다. 실제 측정된 최저 온도는 -30ºC로, 이론적으로 예상되는 16 km 부근의 대기 온도(약 -50ºC)보다 높게 나타났다. 이는 내부와 외부 온도 차이에 따른 측정 지연의 영향으로 해석된다.

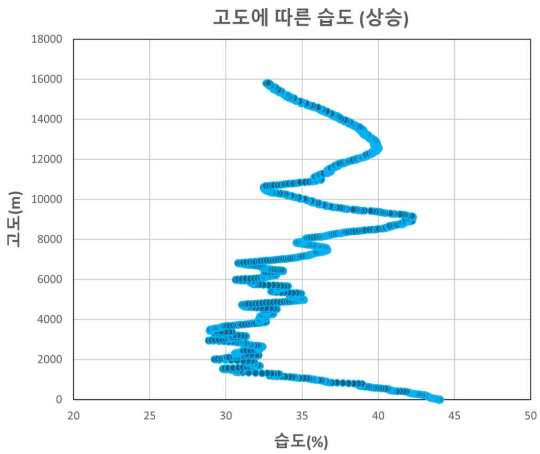

그림 12. 고도에 다른 습도 상승 데이터.

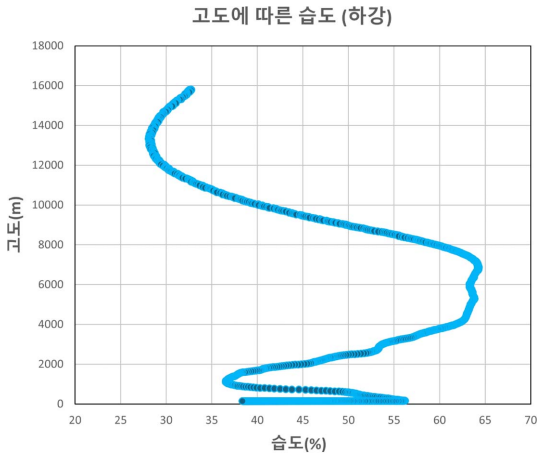

그림 13. 고도에 따른 습도 하강 데이터.

3.4. 고도-습도 그래프를 통한 고도와 습도의 상관관계

고도-습도에 대한 결과는 그림 12–14와 같다. 일반적으로 상대습도는 고도가 증가함에 따라 감소하는 경향을 보인다. 그러나 본 연구에서 수집된 데이터는 상승과 하강 구간 모두에서 증가와 감소가 반복되는 불규칙한 형태를 나타냈다. 상대습도는 실제 수증기량과 포화 수증기량의 영향을 동시에 받으며, 특히 구름이 많은 기상 조건에서는 국지적인 수증기 분포에 따라 급격한 변화가 나타날 수 있다. 본 실험이 수행된 날짜에는 그림 15와 같이 구름이 많은 날씨였으며, 이로 인해 고도에 따른 습도의 뚜렷한 경향성을 도출하기에는 한계가 있었다.

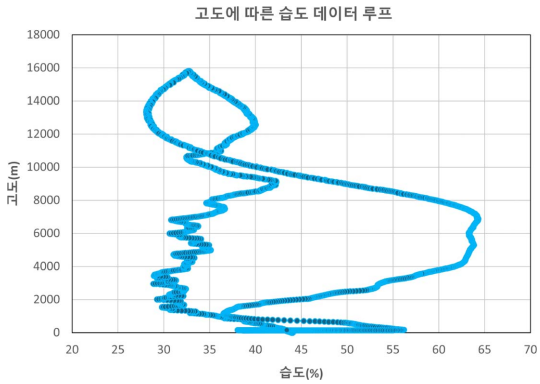

그림 14. 고도에 따른 습도 데이터 루프.

그림 15. 라디오존데에서 촬영한 성층권 사진.

IV. 결 론

본 연구는 중등 과학 교육과정에서 학습한 대기 구조와 기상 요소를 실제 대기 관측 활동과 연계하기 위하여 아두이노 및 ESP32 기반의 라디오존데를 제작·실험하고, 수집된 데이터를 분석함으로써 고도에 따른 기상 요소의 변화를 탐구하는 데 목적이 있다. 이를 위해 1차 및 2차 라디오존데를 설계·실험하였으며, 데이터 수집 및 분석을 통해 다음과 같은 결론을 도출하였다.

첫째, 1차 실험은 데이터 저장 실패라는 기술적 한계를 보였으나, GPS 기반 회수 시스템이 정상적으로 작동함을 확인하였다. 이는 하드웨어 구조와 비행 안정성은 확보되었음을 의미하며, 이후 소프트웨어적 보완을 통해 실험 완성도를 향상시킬 수 있음을 시사하였다.

둘째, 2차 실험에서는 SD카드 저장과 BLE 기반 스마트폰 백업 저장 방식을 병행함으로써 데이터 이중 저장 체계를 구축하였다. 그 결과 최대 고도 약 16 km까지의 온도, 습도, 기압 데이터를 성공적으로 회수할 수 있었으며, 약 4500초(약 1시간 15분) 동안의 비행 데이터를 확보하였다. 이는 학생 주도 제작 라디오존데로 성층권 하부까지의 실제 기상 데이터를 확보했다는 점에서 교육적·기술적 의의를 가진다.

셋째, 고도-시간 그래프 분석 결과 상승 속력은 6.19 m/s, 하강 속력은 7.98 m/s로 계산되었다. 하강 속력이 상승 속력보다 다소 크게 나타났으며, 이는 낙하산 전개 이후 종단속력 조건에서의 공기저항 차이에 기인한 것으로 판단된다.

넷째, 고도-온도 분석 결과 대류권 영역에서 고도 증가에 따른 온도 감소 경향은 확인되었으나, 대류권계면(약 10 km) 부근에서 이론적으로 예상되는 등온층 특성은 명확히 나타나지 않았다. 또한 16 km 부근에서 측정된 최저 온도는 -30°C로, 이론적 성층권 하부 온도(약 -50°C)보다 높게 나타났다. 이는 센서가 라디오존데 내부에 위치하여 단열 효과로 인해 외부 대기 온도가 지연 측정되었기 때문으로 해석된다.

다섯째, 고도-습도 관계는 이론적으로 고도 상승에 따라 감소 경향을 보여야 하나, 실제 데이터에서는 불규칙한 증가·감소 패턴이 나타났다. 이는 실험 당시 구름이 많은 기상 조건에서 상대습도가 실제 수증기량과 포화 수증기량의 영향을 복합적으로 받았기 때문으로 판단된다. 따라서 고도에 따른 습도 경향성을 명확히 도출하기에는 한계가 있었다.

본 연구는 학생들이 직접 라디오존데를 설계·제작·회수·데이터 분석까지 수행함으로써 교과서 속 이론을 실제 대기 환경과 연결하는 경험을 제공하였다는 점에서 큰 교육적 의의를 가진다. 특히 데이터 수집 실패를 기술적 피드백으로 전환하여 2차 실험에서 개선한 과정은 과학적 탐구의 본질인 문제 해결 및 반복적 설계 과정을 잘 보여준다.

그러나 본 연구는 센서의 내부 배치에 따른 열적 지연, 단일 센서 사용에 따른 오차, 기상 조건 통제의 어려움 등 한계를 가진다. 향후 연구에서는 외부 노출형 온도 센서 추가 장착, 스티로폼의 단열 정도를 파악하여 온도 데이터 보정 등 추가 실험의 논의가 필요하다.

결론적으로, 본 연구는 학교 현장에서 구현 가능한

저비용 라디오존데 시스템을 통해 성층권 하부까지의 실제 대기 데이터를 성공적으로 수집하고 분석하였으며, 이를 통해 대기 구조에 대한 이해를 심화시키고 탐구 중심 과학교육의 실천 가능성을 제시하였다. 이는 이론 중심의 대기 단원을 실제 관측 기반 탐구 활동으로 확장할 수 있는 교육적 모델을 제안한다는 점에서 의미가 있다.

참고문헌

[1] 교육부(2022), 2022 개정 과학과 교육과정(교육부 제 2022-33호 [별책 9]).
[2] 김기훈, 김연희, 장동연, KEOP-2007 라디오존데 관측자료를 이용한 장마 특성 분석 : Part I. 라디오존데 관측 자료 평가 분석, 한국기상학회, 19(2). 213-226 (2009).
[3] 전국과학교사협회 https://k-sta.org/
[4] 이영건, 이상현, 김종범, 김송현, 유승훈, 창의공학 설계 및 실습 수업을 위한 아두이노 기반 교육용 캔위성 개발 연구, 한국공학교육학회, **24**, 38-45 (2021).
[5] 눈이 즐거운 물리, 우주 풍선 프로젝트 https://m.blog.naver.com/sanghyup12/221213562484?recommendTrackingCode=2
[6] NASA Glenn Research Center, “Earth Atmosphere Model,” https://www.grc.nasa.gov/www/k-12/airplane/atmosmet.html

부 록

부록 1. 1차 라디오존데의 데이터 수집을 위한 코드

```
#include <Wire.h>
#include <SPI.h>
#include <SD.h>
#include <Adafruit_Sensor.h>
#include <Adafruit_BME680.h>

Adafruit_BME680 bme;
const int chipSelect = 10;
const float seaLevelPressure = 1013.25;

unsigned long startTime = 0;
unsigned long lastMeasure = 0;
const unsigned long interval = 5000; // 5초 간격

File dataFile;

void setup() {
  Serial.begin(9600);
  delay(1000);

  if (!SD.begin(chipSelect)) {
    Serial.println("SD카드 초기화 실패!");
    while (1);
  }

  dataFile = SD.open("data.csv", FILE_WRITE);
  if (dataFile) {
    dataFile.println("경과시간(초),온도(°C),습도(%),기압(hPa),고도(m)");
  } else {
    Serial.println("파일 열기 실패");
    while (1);
  }
  if (!bme.begin(0x76)) {
    Serial.println("BME688 연결 실패!");
    while (1);
  }

  bme.setTemperatureOversampling(BME680_OS_8X);
  bme.setHumidityOversampling(BME680_OS_2X);
  bme.setPressureOversampling(BME680_OS_4X);
  bme.setGasHeater(320, 150);

  startTime = millis();
  lastMeasure = millis();
}
```

```
void loop() {
  unsigned long now = millis();

  // 지정한 간격이 지났는지 확인
  if (now - lastMeasure >= interval) {
    lastMeasure += interval;  // 주기를 정확히 유지
    unsigned long elapsedSeconds = (now - startTime) / 1000;

    if (!bme.performReading()) {
      Serial.println("센서 읽기 실패!");
      return;
    }

    float temp = bme.temperature;
    float hum = bme.humidity;
    float pres = bme.pressure / 100.0;
    float alt = 44330.0 * (1.0 - pow(pres / seaLevelPressure, 0.1903));

    // 시리얼 출력
    Serial.print("T+"); Serial.print(elapsedSeconds); Serial.print("s | ");
    Serial.print("T: "); Serial.print(temp);
    Serial.print(" C, H: "); Serial.print(hum);
    Serial.print(" %, P: "); Serial.print(pres);
    Serial.print(" hPa, Alt: "); Serial.println(alt);

    // SD카드 저장
    if (dataFile) {
      dataFile.print(elapsedSeconds); dataFile.print(",");
      dataFile.print(temp); dataFile.print(",");
      dataFile.print(hum); dataFile.print(",");
      dataFile.print(pres); dataFile.print(",");
      dataFile.println(alt);
      dataFile.flush(); // 매번 저장
    }
  }

  // 루프는 빠르게 반복하지만, 측정은 정확히 5초마다 실행됨
}
```

부록 2. 2차 라디오존데의 데이터 수집을 위한 코드

```
#include <Wire.h>
#include <Adafruit_Sensor.h>
#include <Adafruit_BME680.h>
#include <SD.h>
#include <SPI.h>
#include <BLEDevice.h>
#include <BLEServer.h>
#include <BLEUtils.h>
#include <BLE2902.h>

// ===== BME688 =====
#define BME_ADDR 0x76
Adafruit_BME680 bme;

// ===== SD 카드 =====
#define SD_CS 10
File dataFile;

// ===== BLE =====
BLEServer* pServer = nullptr;
BLECharacteristic* pCharacteristic = nullptr;
bool deviceConnected = false;

#define SERVICE_UUID        "12345678-1234-1234-1234-1234567890ab"
#define CHARACTERISTIC_UUID "abcd1234-ab12-cd34-ef56-1234567890ab"

// ===== I2C 핀 =====
#define SDA_PIN 21
#define SCL_PIN 22

// 타이머
unsigned long lastMillis = 0;
const unsigned long interval = 2000; // 2초 간격

class MyServerCallbacks : public BLEServerCallbacks {
  void onConnect(BLEServer* pServer) {
    deviceConnected = true;
    Serial.println("BLE 장치 연결됨");
  }
  void onDisconnect(BLEServer* pServer) {
    deviceConnected = false;
    Serial.println("BLE 장치 연결 해제");
  }
};

void setup() {
  Serial.begin(115200);
```

```
  delay(1000);
  Serial.println("=== ESP32-S3 시작 ===");

  // ===== BLE 초기화 =====
  BLEDevice::init("ESP32S3_BME688");
  pServer = BLEDevice::createServer();
  pServer->setCallbacks(new MyServerCallbacks());

  BLEService *pService = pServer->createService(SERVICE_UUID);
  pCharacteristic = pService->createCharacteristic(
                      CHARACTERISTIC_UUID,
                      BLECharacteristic::PROPERTY_NOTIFY
                    );
  pCharacteristic->addDescriptor(new BLE2902());
  pService->start();

  BLEAdvertising *pAdvertising = BLEDevice::getAdvertising();
  pAdvertising->addServiceUUID(SERVICE_UUID);
  pAdvertising->setScanResponse(true);
  pAdvertising->setMinInterval(0x20);
  pAdvertising->setMaxInterval(0x40);
  pAdvertising->start();
  Serial.println("BLE 서버 시작, 스마트폰 연결 대기...");

  // ===== BME688 초기화 =====
  if (!bme.begin(BME_ADDR)) {
    Serial.println("BME688 연결 실패! 계속 진행...");
  } else {
    bme.setTemperatureOversampling(BME680_OS_8X);
    bme.setHumidityOversampling(BME680_OS_2X);
    bme.setPressureOversampling(BME680_OS_4X);
    bme.setGasHeater(320, 150);
    Serial.println("BME688 초기화 완료");
  }

  // ===== SD 초기화 =====
  if (!SD.begin(SD_CS)) {
    Serial.println("SD카드 초기화 실패! 계속 진행...");
  } else {
    Serial.println("SD 카드 초기화 완료");
    dataFile = SD.open("/data.csv", FILE_WRITE);
    if (dataFile) {
      dataFile.println("time(s),temperature(C),humidity(%),pressure(hPa),altitude(m)");
      dataFile.close();
    }
  }

  lastMillis = millis();
}
```

```
void loop() {
  unsigned long currentMillis = millis();
  if (currentMillis - lastMillis >= interval) {
    lastMillis += interval; // 정확한 간격 유지

    // BME688 읽기
    float temp=0, hum=0, press=0, alt=0;
    if (bme.performReading()) {
      temp = bme.temperature;
      hum = bme.humidity;
      press = bme.pressure / 100.0;
      alt = bme.readAltitude(1013.25);
    }

    // 시간 초 단위
    float timestamp_s = currentMillis / 1000.0;

    //1) 시리얼 출력 (항목명 포함)
    Serial.print("time(s): "); Serial.print(timestamp_s,2);
    Serial.print(" | temperature(C): "); Serial.print(temp,2);
    Serial.print(" | humidity(%): "); Serial.print(hum,2);
    Serial.print(" | pressure(hPa): "); Serial.print(press,2);
    Serial.print(" | altitude(m): "); Serial.println(alt,2);

    //2) CSV 형식 데이터
    String dataLine = String(timestamp_s,2) + "," + String(temp,2) + "," + String(hum,2) + ","
                      + String(press,2) + "," + String(alt,2);

    // SD 카드 저장
    if (SD.exists("/data.csv")) {
      dataFile = SD.open("/data.csv", FILE_APPEND);
      if (dataFile) {
        dataFile.println(dataLine);
        dataFile.close();
      }
    }

    //3) BLE 전송
    if (deviceConnected) {
      pCharacteristic->setValue(dataLine.c_str());
      pCharacteristic->notify();
    }
  }
}
```