JDCS
디지털콘텐츠학회논문지
Journal of Digital Contents Society
Vol. 23, No. 3, pp. 541-549, Mar. 2022

Check for updates

펠티어 모듈과 PID 제어를 이용한 온도 정밀제어 시스템 개발

나 성 주[1] · 송 주 환[2*]
[1]전주대학교 문화기술학과 박사과정
[2*]전주대학교 인공지능학과 교수

Development of temperature precision control system using Peltier module and PID control

Sung-Ju Na[1] · Ju-Whan Song[2*]

[1]Ph.D. Student, Department of Culture and Technology, Jeonju University, Jeonju 55069, Korea
[2*]Professor, Department of Artificial Intelligence, Jeonju University, Jeonju 55069, Korea

[요 약]

본 연구는 라즈베리 파이와 펠티어 모듈을 이용하여 식물 연구에 이용할 수 있는 소형 챔버 용 온도 정밀제어 시스템을 설계하고 개발하였다. 소형 챔버에 들어가는 기존의 시스템은 단순 On/Off 제어였다. 이것은 식물 생장에 밀접한 요인인 온도에 크게 영향을 미쳤다. 온도 유지가 되지 않고, 목표 온도 주위로 진동하는 현상이 발견되었다. 이 점을 해결하기 위해 본 연구에서는 PID(Proportional-Integral-Differential) 제어를 통해 1분마다 각 온도와 제어량을 산출하여 펠티어 모듈을 제어하도록 했다. 그 결과 외부 온도가 섭씨 22도에서 소수점 단위로 진동하는 현상이 일어날 때, 본 연구에서 구현한 챔버는 내부 온도 섭씨 20.2도를 약 20분간 유지하는 것을 확인하였다. 본 연구를 통해 소형 챔버에서도 요인 정밀제어를 구현하여 연구자가 연구하는 데에 큰 도움이 될 것이다.

[Abstract]

This study designed and developed a temperature precision control system for a small chamber that can be used for plant research using Raspberry Pi and Peltier module. The existing system that goes into a small chamber was a simple on/off control. This greatly affected the temperature, which is a factor closely related to plant growth. The temperature was not maintained, and the vibrating phenomenon around the target temperature was found. In order to solve this point, this study calculated each temperature and the control amount every 1 minute through PID control to control the Peltier module. As a result, when the external temperature vibrates in a decimal point unit at 22 degrees Celsius, the chamber implemented in this study was confirmed that the internal temperature of 20.2 degrees was maintained for about 20 minutes. Through this study, it will be of great help to researchers by realizing factor precision control even in a small chamber.

색인어 : 사물인터넷, 모니터링, 펠티어 모듈, PID 제어, 라즈베리 파이

Keyword : IoT(Internet of Things), Monitoring, Peltier Module, PID Control, Raspberry Pi

http://dx.doi.org/10.9728/dcs.2022.23.3.541

This is an Open Access article distributed under the terms of the Creative Commons Attribution Non-CommercialLicense(http://creativecommons.org/licenses/by-nc/3.0/) which permits unrestricted non-commercial use, distribution, and reproduction in any medium, provided the original work is properly cited.

Received 26 January 2022; **Revised** 21 February 2022
Accepted 17 March 2022

***Corresponding Author; Ju-Whan Song**

Tel: +82-63-220-2912
E-mail: jwsong@jj.ac.kr

Ⅰ. 서 론

2021년 10월 세계보건기구(WHO; World Health Organization)는 기후위기를 '인류가 직면한 최대 보건위협'으로 규정하고 각국 정부가 당장 행동에 나서야 한다고 촉구하였다. 같은 날 기후 분야 국제학술지 네이처 기후변화(Nature Climate Change)는 지구 면적의 80% 이상이 이미 지구온난화의 영향을 받는 중이고, 세계 인구의 85% 이상은 각종 기상이변을 경험하고 있다는 논문을 발표하였다[1]. 또 같은 해 겨울, 미국 중부에서 토네이도가 동시다발적으로 일어나 많은 재산 피해와 인명피해가 발생하게 되었다[2]. 이런 기상이변이 일어나는 원인은 지구온난화로 인한 따뜻한 겨울 때문이다[3].

이상 기후, 기상이변으로 인해 타격을 받는 것은 비단 농부뿐만이 아니다. 지구온난화로 지구의 온도가 높아짐에 따라 우리 인류가 현재 먹고 있는 식량 작물에도 큰 문제가 되고 있다. 식량 위기가 곧 닥칠 수 있다는 예측이 나오자 국가에서 위기에 대해 대응 및 극복을 위해서 식량 위기 극복을 위한 연구에 아낌없는 투자를 하고 있다[4].

식량 위기 극복의 한 방안으로 식물재배를 들 수 있다. 식물이 식량 위기를 대체할 정도의 양을 생산하려면 그에 맞는 적정한 생장 조건을 갖춰야 한다. 그런 조건을 맞추기 위한 식물 연구를 위해서는 식물을 키우기 위한 실험 환경이 마련되어야 한다. 그러므로 많은 연구자는 식물을 키우기 위한 챔버가 필요하다. 그렇기에 많은 이들은 식물 생장 전용 챔버를 구매한다. 그러나 시중에 판매하고 있는 식물 생장 전용 챔버는 대부분 중대형이다. 이것은 연구 공간이 부족한 사람들이 챔버를 구매하기 꺼려질 것이다. 또한, 소형은 일단 찾기 어렵다. 찾아도 대부분 On/Off 제어를 통해 온도나 습도를 조절한다. 이러한 제어 방식은 결국 온도 변화가 자주 일어나게 되므로 식물에 안 좋은 영향을 끼친다. 이 점을 해결하기 위해 본 연구에서는 PID 제어를 통해 온도 정밀제어를 구현하였다.

본 연구에서는 챔버를 구성하기 전 필요한 지식과 기존의 챔버를 2장에서 설명한다. 3장에서는 정밀 소형 챔버에 들어가는 센서와 구동기들을 나열해 설명하고, 챔버 조립까지 설명한다. 4장에서는 본 연구의 챔버에서 사용된 소프트웨어 중심으로 무엇을 구현하였는지 설명하고 실험을 통해 성능을 확인한다. 마지막 5장에서 본 연구의 결론을 서술한다.

Ⅱ. 관련 연구

본 연구는 IoT 기술을 활용하여, 본 시스템에서 데이터 수집 및 판단 그리고 제어를 하게 된다. 수집되는 데이터는 내부의 온도와 습도, 그리고 외부 온도를 수집한다.

2-1 MCU(Microcontroller unit)

MCU는 마이크로프로세서와 입출력 모듈을 하나의 칩으로 만들어 정해진 기능을 수행하는 컴퓨터를 말한다[5].

본 연구에서는 MCU 보드 중 대표적인 2개를 사용하였다. 그것은 바로 라즈베리 파이(Raspberry Pi)와 아두이노(Arduino)이다.

라즈베리 파이는 영국의 라즈베리 파이 재단이 학교와 개발도상국에서 기초 컴퓨터 과학의 교육을 증진시키기 위해 개발한 신용카드 크기의 싱글 보드 컴퓨터이다[6]. 2012년에 처음의 라즈베리 파이를 출시하였으며, 지금까지 다양한 버전의 보드를 내놨다. 본 연구에서는 현재까지 나온 버전 중 가장 최근에 나왔고, 가장 좋은 요건을 가진 라즈베리 파이 4 Model B를 사용하였다.

아두이노는 사용하기 쉬운 하드웨어 및 소프트웨어를 기반으로 하는 오픈 소스 전자 플랫폼이다[7]. 앞서 설명한 라즈베리 파이에 비교하여 설명한다면, 라즈베리 파이가 흔히들 생각하는 컴퓨터라고 한다면, 아두이노는 그에 반해 기계라고 볼 수 있다. 아두이노는 C 언어 기반 플랫폼으로써, 단순 처리에 능하다. 또한, 라즈베리 파이와 달리 아날로그 입력 핀이 존재하여, 아날로그 출력값을 받을 수 있다. 본 연구에서 사용한 아두이노는 Arduino UNO 버전으로, 아두이노 제품군에서 가장 많이 팔린 제품이다[8]. 가장 많이 팔리고, 가장 보편적인 제품의 이점은 여러 가지가 있는 데, 하나는 프로그래밍에 필요한 라이브러리와 자료가 많다는 것이고, 또 하나는 비교적 다른 제품보다 제품값이 저렴하다는 장점이 있다.

2-2 열전효과

두 도체 양단에 열과 전류를 동시에 흘렸을 때, 상호작용으로 서로 영향을 주는 것을 열전효과라고 한다[9]. 열전효과는 총 3가지가 있다. 그 중 첫 번째로는 두 개의 서로 다른 전도체의 접합부에 열을 가하였을 때, 양단에 전위차가 생기는 것을 Seebeck 효과, 두 번째로는 온도 기울기가 있는 도체에 전류를 흘리면, 도체 내부에서 열이 흡수되거나 열이 발생하는 것을 Thomson 효과라 하고, 마지막으로 서로 다른 물질에 전류를 통과시켰을 때, 접합부 근처에서 온도 차가 생긴다는 것이 펠티어 효과이다[10, 11].

이러한 열전효과 중 펠티어 효과를 바탕으로 구성된 모듈을 펠티어 모듈이라고 한다. 펠티어 모듈의 열전소자를 무엇으로 식히는 것인지에 따라 공랭식과 수랭식으로 분류된다. 본 연구에서 사용한 펠티어 모듈은 공랭식을 사용하였다.

그림 1은 펠티어 효과를 설명한 것으로 WATRONIX nic의 제품에서 사용한 이미지를 인용하였다.

(https://inbthermoelectric.com/peltier-seebeck/)

그림 2는 VISION BIONEX의 식물 생장상 제품(VS-3DM / 4DM) 이미지다 (http://www.visionbionex.com/bbs/board.php?bo_table=Chamber&wr_id=26&sfl=wr_subject&stx=VS-3DM&sop=and).

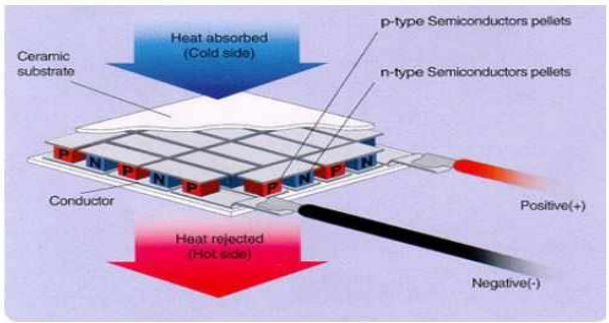


그림 1. 펠티어 효과
Fig. 1. Peltier Effect

그림 2. 중형 식물 생장상
Fig. 2. Medium Plant Growth Chamber

2-3 기존 시장의 챔버

기존 시장의 챔버는 앞서 설명했다시피 중대형 챔버가 연구용으로 우세하고 있다. 그러나 중대형의 단점이 존재한다[12]. 그림 2는 현재 판매되고 있는 중형 식물 연구용 챔버이다[13]. 그 크기만큼이나, 전력 소비량도 상당하며, 비용부담이 크고 협소한 공간에 설치하기에도 어려워 고비용, 저효율적이다. 상술한 이유로 여러 개 구매해서 다양한 실험 환경을 실험하려는 연구자에게는 부적합하다. 중대형을 대체할 소형 챔버는 주로 가정용 재배기로 개발되어 판매되고 있다[14]. 그러나 연구용 소형 챔버는 시장에 거의 출시되지 않았고, 연구를 통해 개발된 챔버의 경우에도 대부분 On/Off 제어를 통해 제어한다[15]. 소형 챔버를 여러 개 구비 해놓고 실험하는 환경을 구축하기 위해서 저가형 정밀제어 소형 챔버의 시스템 개발을 목표로 본 연구를 진행하였다.

2-4 PID Controller

PID Controller(제어기)란, 비례-적분-미분 제어로 구분되며, 상황에 따라서는 비례-적분만 사용하는 PI 제어를 사용하기도 한다. PID 제어는 오차값을 이용해서 비례와 적분, 미분을 통해 오차값을 점점 줄여나가는 피드백 제어이기 때문에, 안정적인 온도 제어 값을 얻을 수 있다.

그림 3은 PID 제어의 기본 구조이다.

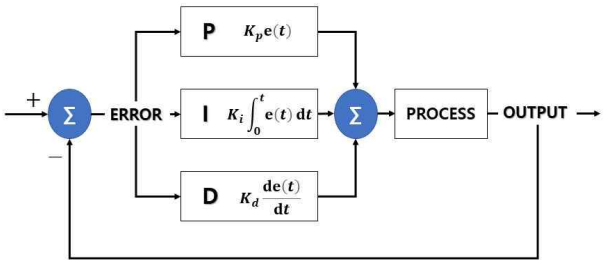


그림 3. PID 제어 기본 알고리즘
Fig. 3. PID Control Standard Algorithm

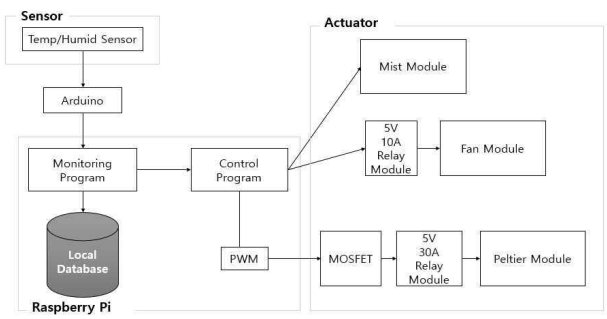


그림 4. 전체 시스템 설계
Fig. 4. System Configuration

Ⅲ. 시스템 설계

3-1 시스템 구성

본 연구에서 구성한 시스템은 그림 4와 같다. 그림에서 보는 바와 같이 크게 3개로 이루어져 있다. 환경 값을 수집하는 Sensor, 제어 값을 수신받고 이후 환경을 제어하는 Actuator 그리고 메인 PC의 역할을 맡는 Raspberry Pi로 구성되어 있다.

3-2 Sensor

먼저 Sensor 부문에 관해서 설명하겠다. 본 연구에서는 PT1000 센서만을 사용하였다. 그림 4의 Temp/Humid Sensor는 PT1000 센서를 뜻한다. 내부 온도, 내부 습도, 외부 온도를 이 센서로 측정하였다. 내부 온도와 내부 습도의 정확한 값을 구하기 위해서는 디지털 센서보다는 아날로그 센서를 써야 한다. 그래서 본 연구에서는 더 정확한 값을 측정하기 위해서, Measuring Box라는 것에 PT1000 센서를 넣어서 측정하도록 하였다[16]. Measuring Box는 생소할 수는 있으나 그림 5처럼 네덜란드의 농업 전문 기업 'Priva'에서도 제작 및 판매하고 있다[17]. 그림을 보면 알 수 있듯이, 기상대에 설치된 백엽상과 비슷한 모습을 띠는 데, 모습과 똑같이 기능도 백엽상과 똑같다. 이 Box에는 센서 Probe 2개와 그리고 물통이 하나 들어간다.

그림 5. Priva의 Measuring Box
Fig. 5. Priva's Measuring Box

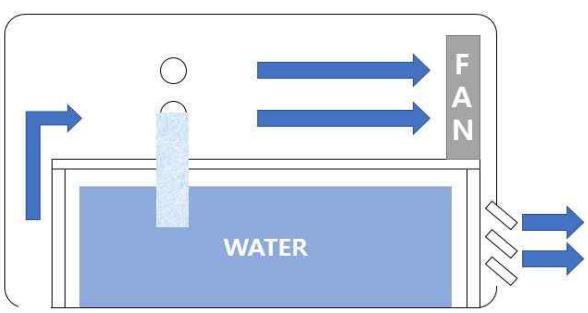


그림 6. Measuring Box의 구조도
Fig. 6. Measuring Box Structure Diagram

표 1. 포화수증기압 추출 코드
Table. 1. Saturated vapor pressure extraction code

```
def ASHREA_WaterVapor_SaturationrPressure(Tdb_C): #Tdb_C
    or Twb_C
    C1 = -5.6745359 * 10 ** 3
    C2 = 6.3925247
    C3 = -9.677843 * 10 ** -3
    C4 = 6.2215701 * 10 ** -7
    C5 = 2.0747825 * 10 ** -9
    C6 = -9.484024 * 10 ** -13
    C7 = 4.1635019
    C8 = -5.8002206 * 10 ** 3
    C9 = 1.3914993 * 10 ** 0
    C10 = -4.8640239 * 10 ** -2
    C11 = 4.1764768 * 10 ** -5
    C12 = -1.4452093 * 10 ** -8
    C13 = 6.5459673 * 10 ** 0

    K_Tdb = Tdb_C + 273.15 #Tdb_C or Twb_C

    if Tdb_C < 0: #Tdb_C or Twb_C
        AWS = math.exp(C1 / K_Tdb + C2 + C3 * K_Tdb + C4 *
    K_Tdb ** 2 + C5 * K_Tdb ** 3 + C6 * K_Tdb ** 4 + C7 *
    math.log(K_Tdb))
    else:
        AWS = math.exp(C8 / K_Tdb + C9 + C10 * K_Tdb + C11 *
    K_Tdb ** 2 + C12 * K_Tdb ** 3 + C13 * math.log(K_Tdb))

    print("saturated water vapor pressure", round(AWS,2))
    return AWS
```

표 2. 상대 습도 추출 코드
Table. 2. Relative Humidity Extraction Code

```
def relative_humidity(Altitude, Tdb_C, Twb_C):
    AP = 101325 * (1 - 2.25577 * 10 ** -5 * Altitude) **
    5.2559
    Saturated_water_vapor_pressure =
    ASHREA_WaterVapor_SaturationrPressure(Tdb_C)
   # saturated water vapor pressure
    Absolute_humidity = ASHREA_HumidityRatio_AWD(Altitude,
    Tdb_C, Twb_C)
   # absolute humidity

    RH = Absolute_humidity * AP /
    (Saturated_water_vapor_pressure * (0.62198 +
    Absolute_humidity))
    print("relative humidity", round(RH*100,2) , "%")

    # Saturated water content
    SMC = Absolute_humidity / RH * 1000
    print("Saturated water content", SMC)

    #humidity deficit
    HD = SMC - Absolute_humidity * 1000
    print("humidity deficit", HD)
    print("----------------------------")
```

표 3. 절대 습도 추출 코드
Table. 3. Absolute Humidity Extraction Code

```
#Absolute Humidity
def ASHREA_HumidityRatio_AWD(Altitude, Tdb_C, Twb_C):
    if Tdb_C < Twb_C:  # <
        AHA = "Error in Value of Temperature. DBT>=WBT"
    else:
        Saturated_HumidityRatio =
    ASHREA_Saturated_HumidityRatio(Altitude, Twb_C)
        AHA = ((2501 - 2.381047 * Twb_C) *
    Saturated_HumidityRatio - 1.006 * (Tdb_C - Twb_C)) /
    (2501 + 1.805 * Tdb_C - 4.186047 * Twb_C)
    print("Absolute Humidity", AHA)
    return AHA
```

그림 6은 Measuring Box의 전체 구조이다. 회사마다 겉부분은 다를지 몰라도 대부분 내부는 이 구조를 따라가고 있다. 그림 6에서 알 수 있듯이 물통에 무언가를 담는 데, 일종의 천으로, 이 천을 사용해 PT1000 센서에 연결하면, 환기팬을 통해 천에 흡수된 물이 기화하면서 열을 가져간다. 이를 통해 습구 온도를 얻을 수 있는 구조이다. Measuring Box를 이용해서 얻을 수 있는 것은 내부의 건구 온도와 습구 온도이다. 건구 온도는 일반적으로 생각하는 기온을 의미한다. 그러나 습구 온도는 습도가 아니므로 습도를 구하는 별도의 식을 사용해 구해야 한다. 본 연구에서는 절대 습도가 아닌 상대 습도를 사용해 판단하므로, 건구 온도와 습구 온도를 사용하여, 상대 습도 구하는 식을 사용 및 값을 도출하였다. 상대 습도를 구하기 위해서는 포화수증기압, 절대 습도, 대기압 등 다양한 값을 구해야 한다.

그중 표1과 표 2가 포화수증기압과 상대 습도를 구하는 Python 코드를 적은 것이다. 표 1에서 Tdb_C와 Twb_C는 각각 건구 온도와 습구 온도를 뜻한다. 표 1에서 나타낸 각각의 C 값은 각각의 고윳값이며 포화수증기압을 구하기 위한 값이다.

표 2는 상대 습도를 구하는 식을 표현한 코드이다. Altitude는 고도를 뜻하며, 본 연구에서는 고도를 임의의 값 100으로 설정하고 계산하였다. AP는 대기압을 뜻한다. AH는 절대 습도를 뜻한다. 절대 습도를 구하는 식은 표 3에서 알 수 있다.

3-3 Actuator

본 단락에서는 본 연구에서 사용한 모듈에 관해 설명한다. 총 4가지로 구성되어 있다. 온도 제어를 담당하는 온도 제어 모듈, 습도를 제어하기 위한 습도 제어 모듈, 온도 보조 제어와 환기를 담당하는 환기 모듈 그리고 전체 on/off 제어를 위한 전기 제어 모듈이 있다.

온도 제어 모듈은 Peltier Module(펠티어 모듈)과 Relay Module(릴레이 모듈)로 구성되어 있다. 펠티어 모듈은 앞서 설명한 대로 열전효과를 이용하여 만든 펠티어 소자를 통해 하나의 모듈로 만든 것이다. 큰 특징 중의 하나는 극을 바꾸면 온도 배출 및 흡수의 방향이 바뀐다. 그것을 구현하기 위해 릴레이 모듈을 이용하여 극 전환을 하도록 하였다. 펠티어 효과는 공기를 가열하거나 냉각시켜 간접적으로 작용하는 것이기 때문에, 효율이 낮다. 그래서 본 연구에서는 효율을 높이기 위해 출력을 높이기로 하였다. 펠티어 모듈은 12V 24A이고, 릴레이 모듈은 입력전압 5V, 출력전압 12V, 최대전류량은 30A이다. 다음 그림 7은 본 연구에서 사용한 펠티어 모듈이다.

PWM(Pulse-Width-Modulation; 펄스 폭 변동 제어)을 사용하여 PID 제어를 할 수 있다. 그러나 앞서 설명한 펠티어 모듈을 제어하기 위해서는 릴레이를 거쳐야 하는 데 릴레이는 이 값을 받아들일 수 없다. 그러므로 전원과 릴레이 사이에 새로운 모듈을 설치하였다. 그것은 MOSFET(Metal Oxide Semiconductor Field Effect Transistor) Module이라는 것이다. 스위칭 속도가 극히 빠르며, N형과 또는 P형의 반도체 벌크에 Drain, Source라 불리는 벌크와 반대되는 반도체형의 확산영역을 만들고, 절연층으로 분리된 게이트 전극을 높인 구조로 구성되어 있다[18].

그림 7. 펠티어 모듈
Fig. 7. Peltier Module

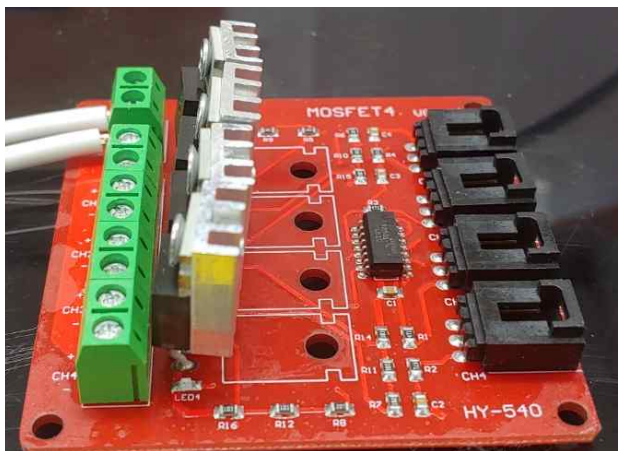


그림 8. MOSFET 모듈
Fig. 8. MOSFET Module

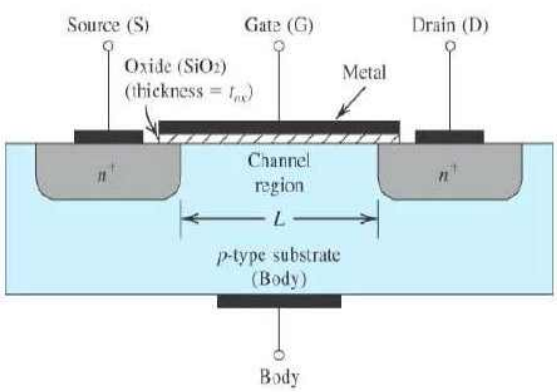


그림 9. MOSFET 모듈의 구조
Fig. 9. MOSFET module structure

이 모듈에는 2가지 특징이 있는데, 하나는 릴레이 모듈처럼 스위칭 기능이 있다. 또 다른 하나는, 그림 9에서 볼 수 있듯이 Source와 Drain 사이의 Gate를 사용하여 Gate의 유지시간을 통해 전압과 전류를 조절할 수 있다. 이 방법을 사용하여 PWM을 통해 제어할 수 있고, 더 나아가 펠티어 모듈을 정밀 제어할 수 있게 하였다.

습도 제어 모듈은 초음파를 이용한 가습기 센서와 물을 공급해줄 물통으로 되어 있다. 본 연구에서 사용한 부품은 'SZH-HUM003' 제품이다.

환기 제어 모듈에는 일반적인 냉각 팬이 하나 들어간다. 펠티어 모듈 자체가 냉방 효율이 많이 떨어지기 때문에, 냉각할 시에 팬이 돌아가고, 또 습도를 제어할 때, 조건부로 돌아가게 하여, 습도 제어에도 보조적인 역할을 하도록 하였다.

3-4 조립

이 단락에서 조립 과정을 설명한다. 소형 챔버를 사용하여 앞서 설명한 각종 Sensor와 Actuator를 설치하였다. 내부 온도를 제어하기 위해 펠티어 모듈을 내부에 설치하였다. 벽

한가운데에 펠티어 모듈을 탑재하였다. 그림 10은 벽에 삽입 및 부착한 펠티어 모듈이다. 습도 모듈을 조립하고 작동까지 확인하였다. 나머지 모듈들도 용도에 맞게 조립하였다. 또한, 내부 데이터 측정을 위해 챔버 내부에 Measuring Box를 설치하였다. 외부 온도 측정을 위한 센서는 임의의 변수에 영향을 받지 않는 챔버 밖에 설치하였다. 그림 11은 최종 조립이 완료된 모습이다.

3-5 Database 설계

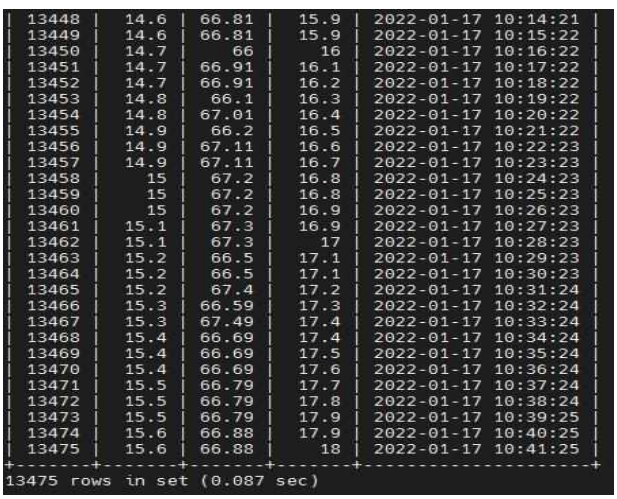
그림 10. 벽에 삽입한 펠티어 모듈
Fig. 10. Peltier module attached to the wall

표 4. 구성 테이블
Table. 4. Main Table

Field	Type	Null	Key	Default	Extra
_id	int(11) unsigned	NO	PK	NOT NULL	auto_increment
no1t1	float	YES		NULL	
no1t2	float	YES		NULL	
no2t1	float	YES		NULL	
tim	datetime	NO		current_timestamp	

그림 11. 조립 완성
Fig. 11. Assembly Complete

본 연구에서는 매1분마다 환경 데이터를 저장한다. 저장된 데이터를 관리하는 데에 필요한 Database를 설계하고 구성하였다. 저장되는 데이터는 내부의 온도와 내부의 습도, 그리고 외부의 온도가 저장된다. 표 4를 보면 어떤 데이터가 저장되는지 알 수 있다.



13448	14.6	66.81	15.9	2022-01-17 10:14:21
13449	14.6	66.81	15.9	2022-01-17 10:15:22
13450	14.7	66	16	2022-01-17 10:16:22
13451	14.7	66.91	16.1	2022-01-17 10:17:22
13452	14.7	66.91	16.2	2022-01-17 10:18:22
13453	14.8	66.1	16.3	2022-01-17 10:19:22
13454	14.8	67.01	16.4	2022-01-17 10:20:22
13455	14.9	66.2	16.5	2022-01-17 10:21:22
13456	14.9	67.11	16.6	2022-01-17 10:22:23
13457	14.9	67.11	16.7	2022-01-17 10:23:23
13458	15	67.2	16.8	2022-01-17 10:24:23
13459	15	67.2	16.8	2022-01-17 10:25:23
13460	15	67.2	16.9	2022-01-17 10:26:23
13461	15.1	67.3	16.9	2022-01-17 10:27:23
13462	15.1	67.3	17	2022-01-17 10:28:23
13463	15.2	66.5	17.1	2022-01-17 10:29:23
13464	15.2	66.5	17.1	2022-01-17 10:30:23
13465	15.2	67.4	17.2	2022-01-17 10:31:24
13466	15.3	66.59	17.3	2022-01-17 10:32:24
13467	15.3	67.49	17.4	2022-01-17 10:33:24
13468	15.4	66.69	17.4	2022-01-17 10:34:24
13469	15.4	66.69	17.5	2022-01-17 10:35:24
13470	15.4	66.69	17.6	2022-01-17 10:36:24
13471	15.5	66.79	17.7	2022-01-17 10:37:24
13472	15.5	66.79	17.8	2022-01-17 10:38:24
13473	15.5	66.79	17.9	2022-01-17 10:39:25
13474	15.6	66.88	17.9	2022-01-17 10:40:25
13475	15.6	66.88	18	2022-01-17 10:41:25

13475 rows in set (0.087 sec)

그림 12. 내부 DB에 저장된 데이터
Fig. 12. Data stored in the internal DB

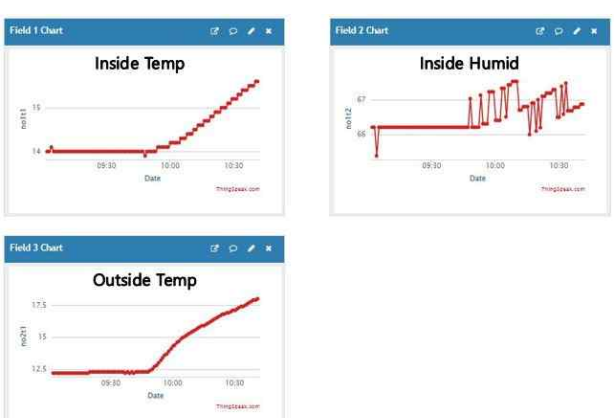


그림 13. Thingspeak에 저장된 데이터
Fig. 13. Data stored in Thingspeak

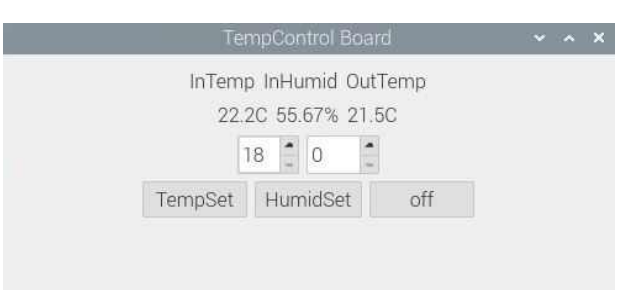


그림 14. 사용자 인터페이스
Fig. 14. User Interface

Ⅳ. 구 현

4-1 모니터링 시스템 구현

라즈베리 파이 4를 사용하였고 Python 3.7 버전을 사용하여 구현하였다. 또한, 아날로그값을 받아들이기 위해 아두이노를 사용하였고, 라즈베리 파이와 아두이노 간 통신은 시리얼 통신을 사용하였다. 표 4와 그림 12와 그림 13을 보면 데이터 저장을 위해서 내부 DB를 만들고, 데이터 시각화를 하기 위해, Thingspeak 사이트를 이용하였다. 사용자가 기계를 조작하려면 UI가 필요하므로 UI를 구현하였다. UI 구현 라이브러리는 PyQt5 라이브러리를 사용하였다. 다음 그림 14는 PyQt5로 구현한 UI이다. 이 UI를 사용하여 사용자가 내부 온도를 조정할 수 있다. 또한, 습도도 조정이 가능하다.

또한, 그림 15를 보면 알 수 있듯이 UI를 통해서 현재 내부 온도와 현재 내부 습도, 그리고 현재 외부 온도를 알 수 있도록, 실시간으로 데이터를 불러온다. 또, 펠티어 모듈의 최대 제어 온도와 최저 제어 온도를 측정하고, 제어 한도제한을 두어 과부하나 출력저하가 일어나지 않게 설계 및 구현하였다. 또, 문제가 생기거나 급히 꺼야 할 상황이 생겼을 때를 대비하여 모든 제어 장치가 꺼지는 버튼도 구현하였다.

4-2 제어 시스템 구현

본 연구에서는 메인 PC인 Raspberry Pi를 기준으로 주로 Relay Module을 이용하여 Actuator를 제어하였다. 그림 14를 보면 UI를 통해 사용자가 임의의 온도나 습도로 제어할 수 있도록 하였다. 또한, 그림 15를 보면 라즈베리 파이 공식 7인치 터치스크린을 이용하여 사용자가 컴퓨터 화면이 아닌 외부 간섭으로도 제어할 수 있도록 하였다.

펠티어에 연결된 Relay Module을 통해 극 전환을 할 수 있도록 하였다. 극 전환을 통해 냉난방을 쉽게 전환할 수 있다. 그림 16과 그림 17은 PID 제어가 아닌 극 전환을 통한 냉난방 전환이 원활히 되는지에 대해 확인하기 위한 과정이다. 또한, 각각 그림은 냉난방 조작 이후에 데이터를 모아 그래프를 그린 것이 아니라 실시간으로 데이터를 모으고 있는 도중에 냉난방 조작을 가해 다음과 같은 결과가 나온 것이다.

먼저 설명에 들어가기 전 챔버에 아무런 작용을 하지 않았을 시에 챔버와 외부 온도와의 차이는 약 1도 내외다. 그림 16과 그림 17은 각각 냉난방 시에 내부의 온도 변화와 습도 변화 그리고 외부 온도를 관측함으로써, 펠티어의 냉난방 성능을 확인하고 냉난방을 하였을 시 내부의 습도가 어떻게 변하는지에 대해 확인한 전체적인 그래프를 나타낸 것이다.

그림 16에서 확인할 수 있듯, 13시 50분부터 챔버의 냉방을 시작하였다. 외부 환경은 난방 에어컨이 작동하고 있는 상태다. 습도는 제어하지 않았다. 냉방을 시작하자 내부 온도가 섭씨 20도에서 약 2도 정도 내려가는 것을 확인하였고, 동시에 외부 온도가 섭씨 20도 이상 가열되고 있는 양상을 보여 외부 온도와의 차이를 보여주었다. 이것을 통해 냉방 기능이 잘 작동되는 것을 확인할 수 있었다. 내부의 습도가 요동치는 양상이 보인 것은 상대 습도의 소수점 2자리까지 저장하는 바람에 생긴 일이다. 또한, 본 논문에서 습도는 온도에 대한 값을 정밀하게 보기 위해 측정한 것이다. 이후 실험에서는 소수점 1자리까지만 그래프에 그려지도록 하였다.

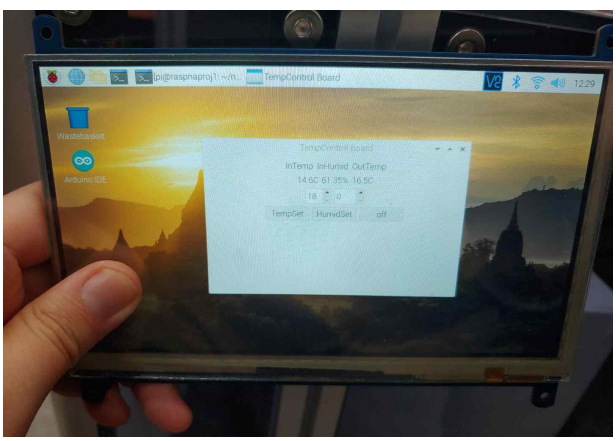


그림 15. 터치스크린
Fig. 15. Touch Screen

그림 16. 냉방 시 온도 변화
Fig. 16. Temperature change during cooling

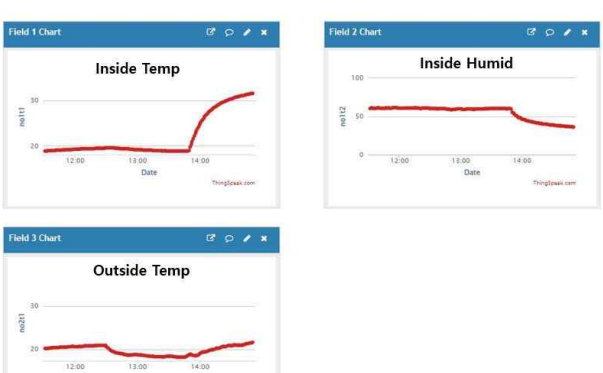


그림 17. 난방 시 온도 변화
Fig. 17. Temperature change during heating

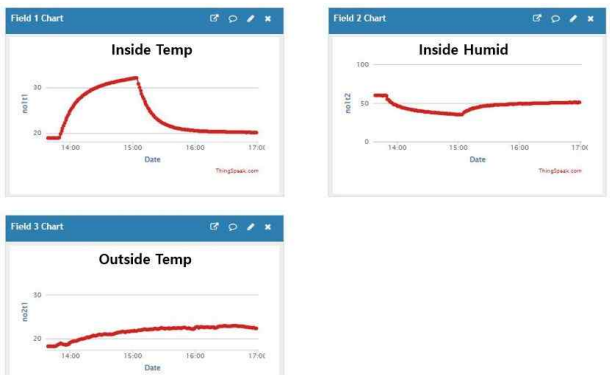


그림 18. PID 제어 시의 온도 변화
Fig. 18. Temperature change during PID control operation

그림 17은 같은 시각 다른 날짜에 실험한 그래프이다. 13시 50분부터 난방 기능을 가동하였다. 15시경 내부 온도는 섭씨 32도에 도달하였고, 당시 외부의 온도는 섭씨 21.7도이었다. 또한, 펠티어 모듈이 냉방보다 난방의 성능이 높다는 것을 확인할 수 있었다.

마지막으로 그림 18은 PID 제어를 사용하여 냉방제어에 적용한 것이다. 15시부터 PID 제어를 이용해 냉방 하도록 하였다. 16시 30분 외부 온도가 섭씨 22.8도를 기록하였을 때, 내부 온도는 섭씨 20.2도를 기록하였고, 약 20분간 유지하는 것을 확인할 수 있었다. 이 같은 실험을 약 열흘간 하루에 2번씩 실험하였고, 똑같은 결과를 얻을 수 있었다. 추가로 그림 18의 외부 온도 데이터를 보면 사무실 내 설정 온도인 22도 주위로, 온도 값이 소수점 한자리 단위로 미세하게 진동하는 양상이 보였다. 그러나 본 연구에서 구현한 것을 토대로 나온 데이터는 일정한 값을 약 20분간 유지하고 소수점 한자리까지 유지함으로써, 간단한 성능 비교가 되었다고 생각한다.

Ⅴ. 결 론

본 연구에서 펠티어 모듈과 PID 제어를 이용하여 소형 챔버의 정밀제어 시스템을 구현하였다. 기존 중대형 챔버에 들어가던 정밀제어 시스템을 소형 챔버에도 구현하였다. 펠티어 모듈 하나만을 사용해서 냉방기와 난방기 두 개를 탑재하지 않아도 냉난방을 제어할 수 있도록 하였다. 또한, PID 제어 방식을 사용하여 On/Off 제어 보다 안정적인 온도 값을 도출하도록 하였다. 그러나 총 2개의 문제가 존재하였다. 구성 모듈에서 문제가 있었고, 펠티어의 냉방 성능 부족으로 인한 문제가 있었다. 첫 번째 문제는 펠티어 모듈의 PID 제어를 하기 위한 필수 모듈인 MOSFET 모듈에서 문제가 발생하였다. 30A의 고전류를 본 논문에서 사용한 모듈이 버티질 못해, 약 10분 정도 작동되다가 화재가 발생하였다. 새로운 모듈을 구해서 교체하면 되는 일이지만, 최근 코로나 사태로 인한 2020년대 반도체 공급 부족으로 부품 구하기가 다소 어려워져 제때 교체할 수 없었다. 두 번째 문제는 펠티어 소자의 방식이 간접적인 온도 제어 방식이기 때문에 기존의 냉방기, 난방기보다 효율이 낮다. 또한, 냉방은 난방과 비교하면 효율이 극도로 낮으며, 냉방 효율을 증가시키려면 내부의 크기와 비례하여, 크기와 전기 사용량이 증가해야 할 것이다. 또는 냉각 방식을 공랭식이 아닌 수랭식으로 바꾸어 냉각 효율을 증가시키는 방법이 있다.

이 두 가지의 문제점을 개선한다면, 소형 챔버에서도 손쉽게 온도 전환이 쉬울 것이고, 식물 연구에도 쉬울 것으로 보인다. 추후 연구에서는, 앞서 설명한 두 가지의 문제를 해결한 모델을 마저 구현하는 것이고, 온도뿐만이 아닌 습도와 상관관계 사이에서 PID 제어를 이용하여 온도와 습도를 동시에 제어하는 것을 목표로 연구할 계획이다.

참고문헌

[1] E. T. Lee, “85% of the world’s population experience extreme weather… Ecosystem could collapse by the end of this century,” *Dong-A Ilbo* [Internet]. Available: https://www.donga.com/news/Inter/article/all/20211013/109676298/1.

[2] D. H. Jeon, “U.S. tornado kills more than 70: "Federal resources are immediately put in",” *MBC* [Internet]. Available: https://imnews.imbc.com/replay/2021/nw1200/article/6323212_34908.html.

[3] J. E. Kim, “Central US tornado death toll likely to exceed 100The cause is "warm winter",” *NEWSIS* [Internet]. Available: https://newsis.com/view/?id=NISX20211213_0001685291&cID=10101&pID=10100.

[4] I. Kang, “Rural Development Administration confirmed next year’s budget of 1.18 trillion won… Focus on digital agriculture,” *Financial News* [Internet]. Available: https://www.fnnews.com/news/202112091535095538

[5] Wikipedia. Microcontroller [Internet]. Available: https://en.wikipedia.org/wiki/Microcontroller

[6] Wikipedia. Raspberry Pi [Internet]. Available: https://en.wikipedia.org/wiki/Raspberry_Pi

[7] Arduino. [Internet]. Available: https://www.arduino.cc/en/Guide/Introduction

[8] Arduino. [Internet]. Available: https://store-usa.arduino.cc/products/arduino-uno-rev3/?selectedStore=us

[9] Wikipedia. Thermoelectric_effect [Internet]. Available: https://en.wikipedia.org/wiki/Thermoelectric_effect

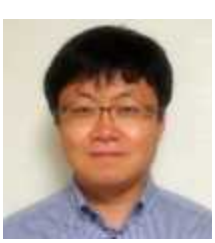
[10] H. S. Yoon, A Study on Cooling Performance using Thermoelectric Module in a Small Cooling System, Ph.D.

dissertation, Graduate School of Chosun University, Gwangju, 2013.
[11] D. J. Lee, Development of Temperature Control System in Aquariums applying Thermoelectric Device, Master's thesis, Kangwon National University Graduate School of Industrial Science, Gangwon-do, 2016
[12] M. G. Kim, W. J. Shin, C. W. Lee, “Development of Small Test Chamber Method for Furniture 1,” in *Proceeding of the 43rd Meeting of Korea Society for Atmospheric Environment*, Suwon, pp. 183-184, 2006.
[13] Vision Bionex. [Internet]. Available: http://www.visionbionex.com/bbs/board.php?bo_table=Chamber&wr_id=26&sfl=wr_subject&stx=VS-3DM&sop=and
[14] Kyowon Wells. [Internet]. Available: http://www.kyowonwells.com/Product/Detail?grpIdx=1235&productIdx=496
[15] O. H. Kwon, I. C. Kang, D. S. Min, H. B. Im, Y. W. Park, “A Study on the Smart Farm Characteristics Using Multiple Sensors,” *The Journal of the Korea Institute of Electronic Communication Sciences*, Vol. 16, No. 4, pp. 719-724, Aug 2021., http://dx.doi.org/10.13067/JKIECS.2021.16.4.719
[16] Adafruit. [Internet]. Available: https://www.adafruit.com/product/3984
[17] Priva. [Internet]. Available: https://www.priva.com/solutions/horticulture/priva-sensors-for-horticulture
[18] I. C. Shin, A study on Design of 100V Class Super-junction Power MOSFET, Master's thesis, Woosong University Graduate School, Daejeon, 2016

나성주(Sung-Ju Na)

2020년 : 전주대학교 문화융합대학 스마트미디어학과 학사 졸업
2022년 : 전주대학교 일반대학원 스마트 Agro_ICT 융합학과 석사 졸업
현 재 : 전주대학교 일반대학원 문화기술학과 박사과정 재학

2020년～2022년: 전주대학교 일반대학원 스마트 Agro_ICT 융합학과 석사
2022년～현 재: 전주대학교 일반대학원 문화기술학과 박사과정
※관심분야 : 사물인터넷(IoT), 임베디드 시스템, 스마트팜 등



송주환(Ju-Whan Song)

1997년 : 전북대학교 대학원 (이학석사)
2003년 : 전북대학교 대학원 (이학박사)

2006년～현 재: 전주대학교 인공지능학과 교수
2019년～현 재: 전주대학교 일반대학원 Agro AI학과 교수
※관심분야 : 인공지능, 스마트팜, 영상처리, 사물인터넷(IoT), 디지털콘텐츠, 모바일 프로그래밍 등