Journal of Institute of Control, Robotics and Systems (2017) 23(11):927-934
https://doi.org/10.5302/J.ICROS.2017.17.0155 ISSN:1976-5622 eISSN:2233-4335

Raspberry Pi를 이용한 원격 제어 및 모니터링 시스템 개발과 HILS를 통한 시스템 검증

Development of a Remote Control and Monitoring System using Raspberry Pi and Its Verification through HILS

김 태 윤, 이 영 삼*
(Tae Yoon Kim[1] and Young-Sam Lee[1,*])
[1]Department of Electrical Engineering, Inha University

Abstract: In this paper, we propose a remote control and monitoring system using the Raspberry Pi as an IoT device. The proposed system has the following features. First, various sensor information can be acquired through Arduino Due's rich peripherals. Second, the acquired information is delivered to a PC through Ethernet communication of Raspberry Pi, which is an IoT device. Third, easy use is supported through a lab-built user interface. In the proposed system, the acquisition of sensor information and processing of control commands are performed on Arduino Due, and the measured information is designed to be transmitted to Raspberry Pi through SPI communication. A lab-built GUI program is introduced to efficiently display the received the sensor information on a PC efficiently. To illustrate the usefulness of the proposed system, an HILS (hardware-in-the-loop simulator), which simulates the greenhouse phenomenon, is implemented using Arduino Nano. The entire proposed system is implemented using open-source hardware and can be easily deployed at low cost.

Keywords: Remote control and monitoring system, Open-source hardware, IoT, Raspberry Pi, Arduino Due

I. 서론

최근 4차 산업혁명에 대한 관심도가 증대됨에 따라 IoT를 접목시킨 시스템에 대한 관련 기술 개발이 활발히 이루어지고 있다[1-3]. IoT 기술의 발전으로 인해 각종 센서, 카메라, 전자제품 등의 사물에 인터넷이 연결되어 다양한 협업 기능을 가능하게 하고 있다. IoT 기술들은 다양한 분야의 기술들과 접목되어 많은 분야에서 활용되고 있으며, 특히 인터넷에 연결된 사물에 대한 실시간 모니터링과 같은 다양한 서비스를 제공할 수 있다. 또한 사용자의 목적을 파악하여 일정 수준의 자동화 결정을 함으로써 사용자에게 편의를 제공할 수 있는 장점이 있다[4,5].

이러한 장점으로 인해 IoT 기술에 대한 연구가 활발히 이루어지고 있다. [6]에서는 디지털 도어락 제어와 관련하여 마이크로 컨트롤러와 블루투스 모듈, 그리고 스마트폰 플랫폼을 사용하여 제어한 연구를 수행하였다. [7]에서는 주차 공간을 블루투스와 초음파 센서를 통하여서 관리하는 시스템을 제안하였다. 또한 마이크로 컨트롤러가 블루투스 모듈을 이용하여 LED 식물공장을 자동으로 제어하는 시스템을 만든 연구도 있다[8]. 하지만 [6-8]의 연구에서는 블루투스 모듈을 이용하여 근거리에서만 통신을 수행하며 또한 제한된 센서만을 이용할 수 있기 때문에 시스템 구현에 한계점이 있다. 따라서 이러한 문제점을 개선하기 위하여 통신 시스템으로 이더넷을 이용하는 Raspberry Pi를 사용해 볼 수 있다. [9]에서는 Raspberry Pi를 이용하여 의료 데이터를 수집하고 분석하는 시스템을 다루고 있다. 또한 [10]에서는 Raspberry Pi에 각종 센서를 연결하여 가정의 환경을 모니터링하고 제어할 수 있는 시스템을 구현한 연구를 수행하였다. [11]에서는 환경 센서를 이용하여 실내 공기의 질을 측정하고, 측정 데이터를 분석하여 환기 개선에 필요한 액츄에이터 신호를 생성하고 실시간으로 실내 공기 질 측정 모니터링 시스템을 구현한 기술을 구현하였다.

이러한 시스템은 Sensor Shield나 추가적인 센서를 이용한다. 하지만 이러한 방식에서 제공되는 센서의 종류는 매우 제한적이어서 Encoder counter 또는 PWM과 같은 심화된 기능은 제공하지 못한다.

따라서 본 논문에서는 Raspberry Pi에서 다양하고 많은 I/O 기능을 이용하기 위해서 Arduino Due를 함께 사용한다. 또한 개발된 시스템은 모두 오픈소스 하드웨어를 이용하여 시스템 개발비용을 낮추었다. 하지만 이러한 많은 기능들을 구현할 때에는 디스플레이 할 수 있는 프로그램을 찾기 힘들다. 따라서 모니터링 및 제어가 가능한 GUI 윈도 프로그래밍을 직접 개발하여 사용한다. 제안된 시스템의 효용을 검증하기 위하여 HILS (Hardware In the Loop Simulation)를 이용한 비닐하우스 시스템과 실내 온실 시스템으로 구현하여 테스트하게 된다.

HILS는 임베디드 시스템을 효율적으로 검증할 수 있는 테스트 방식이다. 실제 시스템을 테스트하기 위해서는 시간

* Corresponding Author
Manuscript received July 24, 2017 / revised September 26, 2017 / accepted September 29, 2017
김태윤, 이영삼: 인하대학교 전기공학과
(kty9304@naver.com/lys@inha.ac.kr)
※ 이 논문은 2017년도 정부(교육부)의 재원으로 한국연구재단의 지원을 받아 수행된 연구임(No. 2017R1D1A1B03029578).

과 비용이 많이 소요되며 위험도 따를 수 있다. HILS를 이용하면 비용을 줄일 수 있으며 실현 불가능한 상황이나 위험한 상황도 실험할 수 있는 장점이 있다. 이런 이유로 자동차, 철도, 항공 우주, 의료기기, 산업기계, 전력 생성 시스템 등과 같은 다양한 분야에서 HILS를 이용한 시스템 검증을 수행하고 있다[12].

본 논문에서는 시간이 오래 걸리고 큰 장소와 비용이 많이 소요되는 비닐하우스와 실내 온실을 HILS를 이용하여 구현하고 제안된 시스템을 검증해 본다. 관련하여 온수난방 시스템 온실의 온도 제어 시뮬레이션 시행한 연구가 있다[13]. 이 연구에서 온수 공급에 의한 실내 온도 변화는 열전달을 구하여 식을 구현하였다. 그리고 온도 제어를 위해서 온수의 온도 변화를 상세히 구현하여 실험하였다. 본 논문에서는 실내 온도 변화는 열전달을 이용하여 같은 방식으로 구현하고 온수의 온도 변화는 관련이 없어서 고려하지 않는다.

본 논문의 구성은 다음과 같다. II절에서는 전반적인 시스템의 개요를 소개하고 제안된 시스템의 구성된 장치와 시스템 애플리케이션에 대하여 설명한다. III절에서는 제안된 시스템을 실험해보기 위해 HILS를 구현한다. 구현된 HILS를 이용하여 제안된 시스템을 테스트하고 테스트한 결과를 보여준다. 마지막으로 IV절에서 결론을 맺는다.

II. 시스템 개요

제안된 시스템은 그림 1과 같이 오픈소스 하드웨어와 IoT 디바이스 그리고 모니터링 및 제어할 수 있는 컴퓨터로 구성된다. 오픈소스 하드웨어인 Arduino Due와 저가형 IoT 디바이스 Raspberry Pi를 이용하여 고가의 장비 없이 시스템을 구성한다. Raspberry Pi는 이더넷을 이용할 수 있다는 큰 장점이다. 하지만 GPIO와 serial 통신 핀(UART, SPI, I2C)만 제공하기 때문에 ADC나 PWM 같은 주변 장치를 이용하지 못한다. 다양한 주변장치를 이용하기 위하여 Arduino Due를 이용하였고 두 장치를 SPI 통신을 통해 연결해 준다. Arduino Due에는 2 channel의 encoder counter, 8 channel의 ADC, 여러 channel의 PWM, 2 channel의 DAC (Digital to Analog Converter), 다수의 digital input/output 핀이 있다. 그중에서 기능 구현의 예시를 위하여 ADC 2 channel 그리고 DAC, encoder counter, PWM, 각각 1 channel 그리고 D/I 핀 8 channel을 구현한다.

그림 2는 위에서 설명한 개략적 수행 방식을 흐름도를 이용하여 나타내고 있다. 우선 Arduino Due에서 주변장치를 이용하여 입력 값들을 측정하고 SPI 통신을 통하여 Raspberry Pi와 데이터 교환을 하게 된다. 여기서 Raspberry Pi를 SPI 통신 master, Arduino Due는 slave로 설정한다. Arduino Due는 측정, 데이터 교환, 출력을 반복하게 된다.

Raspberry Pi에서는 이더넷 통신을 이용하여 컴퓨터와 연결하게 된다. 이더넷 통신 중에서는 TCP/IP (Transmission Control Protocol / Internet Protocol)통신을 이용하였는데 이것은 UDP (User Datagram Protocol)통신 방법보다 절차는 많아 조금 느릴 수 있지만 안전하게 정보를 전달할 수 있기 때문이다. TCP/IP 통신에서 컴퓨터를 server로 Raspberry

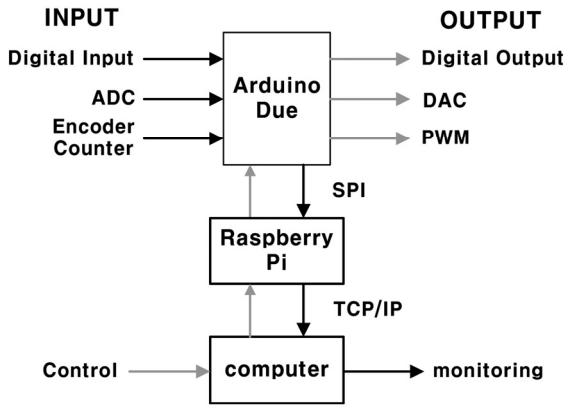

그림 1. 제안된 시스템 구성도.

Fig. 1. The conceptual diagram of the proposed system.

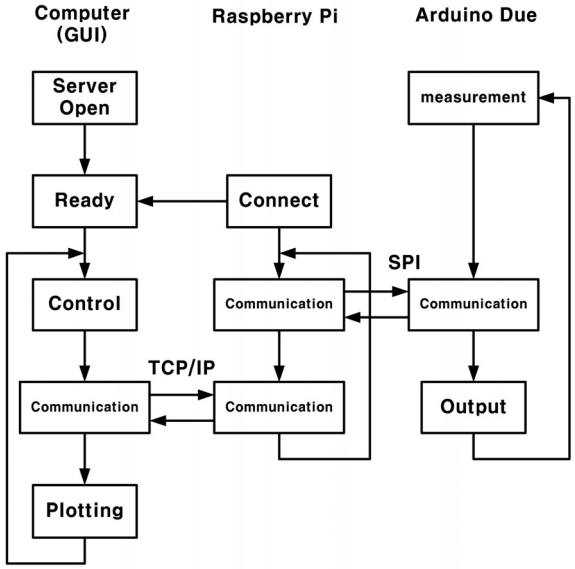

그림 2. 제안된 시스템의 수행 방식에 대한 흐름도.

Fig. 2. The flowchart on the operation of the proposed system.

Pi를 client로 설정한다. Raspberry Pi는 컴퓨터에 server가 먼저 열린 것을 확인하고 접속을 하게 된다. 접속할 때 server와 client 간에 IP (Internet Protocol)주소와 포트 번호를 맞추어 주어야 한다[14]. 그 뒤에 Raspberry Pi는 Arduino Due로부터 받은 데이터를 컴퓨터로 보내주고 컴퓨터로부터 받은 데이터를 Arduino Due로 보내준다. Raspberry Pi는 한번 접속한 후에는 Arduino Due와 데이터 교환, 컴퓨터와 데이터 교환을 반복한다.

컴퓨터에서는 server를 열고 접속을 대기한다. 그 후에 client의 접속이 확인되면 데이터를 교환하여 받은 데이터는 계산을 통해 GUI를 통하여 사용자에게 보여준다. 컴퓨터에서는 접속이 확인된 다음에는 지속적으로 데이터 교환과 GUI 업데이트가 이루어진다.

1. 시스템 구성 장치

제안되는 시스템의 하드웨어 구성은 다음과 같다. 센서 정보를 취득할 수 있는 오픈소스 하드웨어 Arduino Due와 IoT 디바이스인 Raspberry Pi, 그리고 HILS를 구현한 Arduino

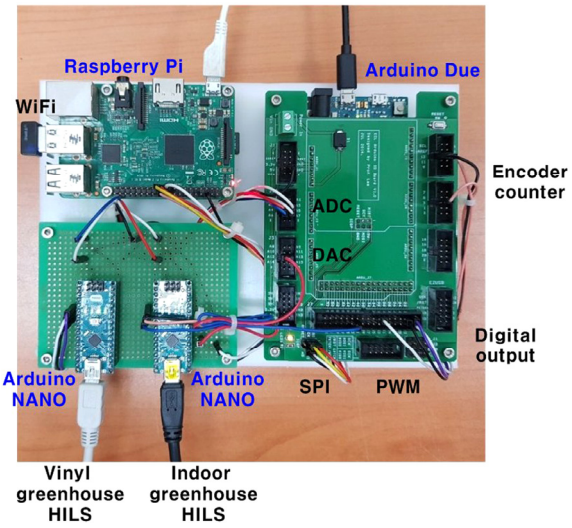

그림 3. 구현된 시스템의 실물 사진.

Fig. 3. Picture of an implemented system.

Nano로 구성된다. 그리고 모니터링과 제어를 수행하는 컴퓨터가 포함된다. 그림 3은 구현된 시스템의 하드웨어적 구성을 보여주고 있다.

Arduino Due는 32 비트 프로세서를 사용하여 성능이 높은 제품이고 동작 clock이 84MHz로 빠른 CPU 연산속도를 가진다. 다수의 PWM, ADC, 디지털 I/O 핀을 가지고 있어서 다양한 입출력을 구현하기 위하여 Arduino Due를 이용한다.

Raspberry Pi에는 운영체제를 올릴 수 있어 다양한 기술분야를 사용할 수 있다. 특히 많은 IoT 장치들이 인터넷을 중심으로 개발되고 있기 때문에, Raspberry Pi에 WiFi 기능이 내장된 것은 큰 장점이라 할 수 있다. 게다가 Raspberry Pi에는 GPIO이 내장되어 있어 기본적인 Digital I/O 기능을 처리할 수 있다. Raspberry Pi의 장점으로는 저가, 저전력이면서도 비교적 높은 성능을 갖추고 있다. C/C++, Java, Python 들의 개발 툴을 사용할 수 있다[15].

2. 시스템 애플리케이션

컴퓨터에서 제작된 GUI는 윈도용 응용 프로그램의 통합 개발 환경인 MFC (Microsoft Foundation Class)를 이용하여 제작한다. MFC는 윈도용 응용 프로그램의 통합 개발 환경인 마이크로소프트 비주얼 C++에 부속되는 클래스 라이브러리이다[16]. 또한 코딩 량을 줄여주면서도, 실행 속도도 빠르다는 장점을 가진 라이브러리이다. 다른 윈도 개발 방법엔 자바 Swing, python, Tkinter, Adobe AIR가 있지만 이것들은 결국 MFC나 Windows API로 작성한 라이브러리를 이용해서 작동하기 때문에, 전환시간이 걸려서 결국 MFC보다 느릴 수밖에 없다. C로 직접 Windows API를 사용하는 것은 기반이 되는 것이고, 굉장히 힘든 과정이지만 이것을 C++로 쓰기 좋게 포장해서 OOP의 장점을 얻는 것이 MFC이고 이 MFC를 이용해서 더 쓰기 좋게 만든 것이 다른 윈도 프로그램 개발 방법들이다[17].

그림 4는 자체 제작된 GUI로 구성된 메인 화면이다. 오른쪽 위에는 컨트롤 버튼이 있고 왼쪽에는 출력을 조절하고 입력을 볼 수 있는 창이 칸이 있다. 왼쪽 아랫부분에 입력 박스에서는 encoder counter, ADC, digital input을 확인할 수 있다.

컨트롤 버튼에 연결은 TCP/IP server을 오픈하여 client의 접속을 기다린다. 저장은 측정된 데이터를 한 시간 단위로

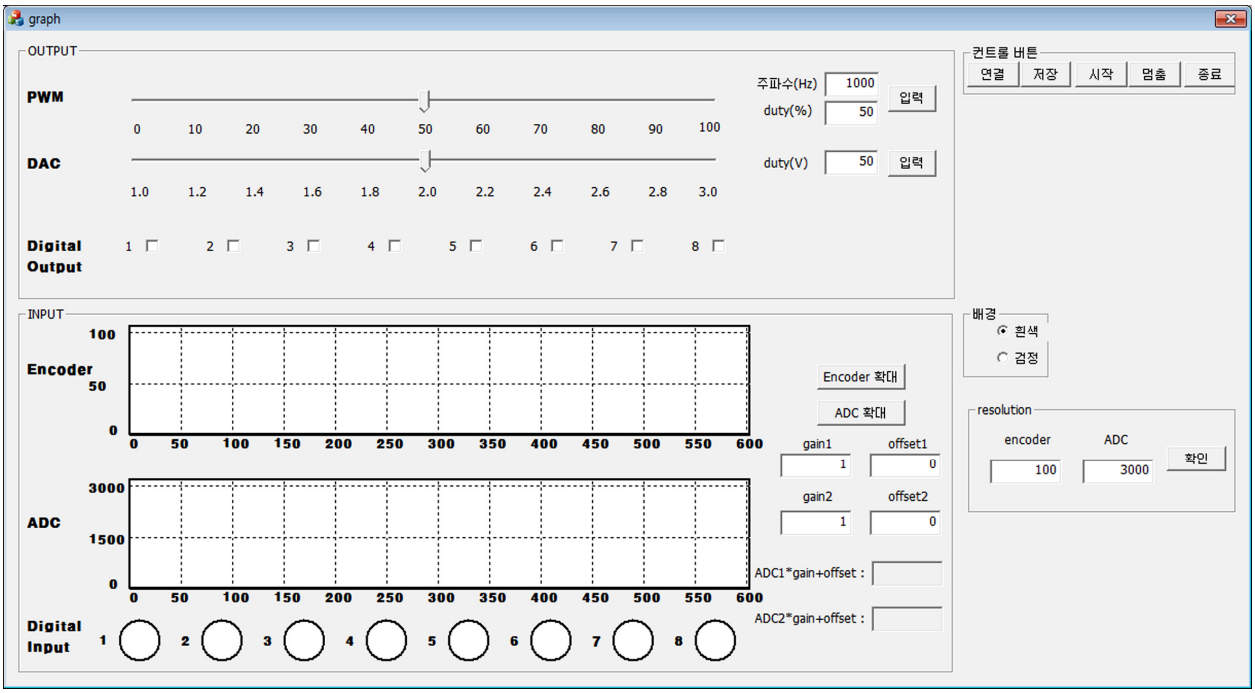

그림 4. 시스템 모니터링 및 제어를 위한 GUI 화면.

Fig. 4. GUI screen for system monitoring and control.

엑셀파일로 만들어 저장하게 된다. 실험에서는 결과를 빠르게 확인하기 위해서 1초에 한 번씩 저장하게 설정하였다. 시작은 그래프를 그리기 시작하는 버튼이고 멈춤은 그래프 그리다가 즉시 멈추었다가 시작을 누르면 다시 그리게 된다. 마지막으로 종료 버튼으로 창을 닫을 수 있다.

좌측 상단부분에 출력에는 PWM, DAC와 digital output을 설정할 수 있는 부분이다. PWM은 주파수와 duty를 숫자를 네모 칸 안에 넣어 옆에 입력 버튼을 누르면 변경이 되고 슬라이드 바를 이용하여서 duty를 조절할 수도 있다. 마찬가지로 DAC도 duty를 네모 칸 안에 입력하여 버튼을 누르거나 슬라이드 바를 이용하여서 조절할 수 있다. 그리고 밑은 digital output을 설정해주는 부분인데 체크 박스로 되어 있어서 누르면 logic high를, 체크를 해제하면 logic low를 출력하게 된다.

좌측 하단부인 입력 부분은 encoder counter와 ADC 그래프를 그려주고 digital input 값을 표시해준다. 그릴 때 입력 박스 옆쪽에 resolution 박스를 설정해주어 encoder와 ADC의 resolution에 대한 정보를 제공함으로써 그래프를 보기 쉽게 그릴 수 있도록 설정해준다. Resolution 값을 넣어서 확인을 누른 후에 encoder 부분에서는 encoder counting 한 값을 시간에 따라서 그리게 된다. 마찬가지로 ADC 부분도 시간에 따라 ADC 측정치를 그려주는데 이때 측정된 값에 gain이나 offset을 주어서 실제 원하는 값으로 환산에 도움을 주도록 하는 기능이 있다. 그리고 그래프를 크게 보기 위해서 확대 버튼을 누르면 새로운 창에 그래프를 확대하여 그려준다. 밑에 digital input에서는 logic high이면 원이 노란색으로 채워지고 logic low면 흰색으로 표시되어 확인할 수 있도록 하였다.

추가적으로 입력 박스 옆에 배경 박스에서는 그래프를 그려지는데 배경을 검은색으로 선을 노란색과 분홍색으로 바꿔서 보기 편한 방법으로 볼 수 있도록 하는 기능을 제공한다.

III. 시스템을 이용한 사례

본 논문의 제안된 통신 시스템을 검증하기 위해서 예시 시스템을 만들어 확인해보도록 한다. 예시로 HILS를 이용한 비닐하우스에서의 온도와 토양 습도 정보 모니터링과 제어하는 시스템을 구현한다. 그리고 실내 온실에서 온도, 습도, 조도를 모니터링하고 온도, 습도, 조도 량을 제어하는 시스템을 구현한다. 실제 비닐하우스에서 측정해볼 수도 있지만 그 경우에는 시간의 변화에 따른 측정이 비효율적으로 되기 때문에 HILS를 이용하여 실용적으로 테스트한다.

1. 비닐하우스 시스템

우선 HILS를 구현하기 위하여 외부의 온도는 연평균 기온 데이터를 이용하여 만든다[18]. 한 해의 각 날마다 최고기온과 최저기온을 저장하고 그날의 최고기온은 14시 최저기온은 4시로 정해서 한 시간 간격으로 선형적으로 기온 값을 구한다. 외부 기온이 변함에 따라 비닐하우스에 온도가 변함을 구하기 위해서 비닐하우스의 온도를 23℃로 유지한다고 가정하고 외부 온도 변화에 따른 열 손실을 아래 식 (1)을 통하여 구한다. 비닐하우스의 크기는 가로 8 m 세로 4 m로 가정하고 높이 직선으로 1 m 반지름 2 m인 반원으로 지붕을 가진다고 가정한다. 모식도는 그림 5와 같다.

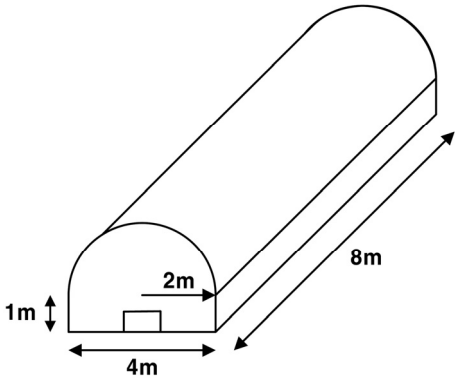

그림 5. 비닐하우스 모식도.
Fig. 5. Vinyl greenhouse mimetic diagram.

$$Q = UA(t_r - t_0) \quad (1)$$

Q는 외벽, 지붕, 바닥, 유리창을 통한 열 손실, A는 외벽, 지붕, 바닥, 유리창 등의 면적으로 86.83 m^2, U는 총괄 열전달계수로 비닐은 폴리프로플렌 (Polypropylene)으로 0.12W/m·K이고, t_r[K]은 실내공기 온도, t_0[K]는 외부 온도이다.

추가적으로 비닐하우스 안에 1800W/h 효율을 가진 난방기를 설치한다고 하고 비닐하우스 안에서 난방기로부터 전도, 대류, 복사 에너지까지 고려하여 열전달을 구해보면 다음과 같은 식 (2)가 된다.

$$Q = \kappa A\frac{T_1 - T_2}{L} + \epsilon\sigma A_s(T_1^4 - T_2^4) + hA_s(T_1 - T_2) \quad (2)$$

Q는 열전도율, κ는 열전도계수로 공기 중에 20℃에서 값은 0.0257 W/m·K, A는 86.83 m^2, T_1은 난방기의 온도로 650 K, T_2는 실내 온도 300 K, ϵ은 표면의 방사율 1, σ는 Stefan-Boltzmann의 상수 $5.67\times10^{-8}\,W/m^2K^4$, h는 대류 열전달계수로 공기 중에선 0.5 W/m^2·K이다[19]. 식 (1)과 식 (2)를 더하여 난방기 효율로 나누어주면 총 걸리는 시간을 구할 수 있게 된다. 걸리는 시간만큼 시간당 비례적으로 온도가 상승할 수 있도록 HILS를 구성한다.

습도의 경우에는 건조한 날에는 6시간이면 땅이 마르고 습한 날에는 24시간이 걸린다. 5 V 오픈소스 하드웨어를 이용하여 토양습도센서로 측정하였을 때 물이 말라서 필요할 경우에는 3.81 V가 측정되었고 물을 줬을 때에는 2.15 V가 측정되었다. 이를 이용하여 HILS를 구성할 때 1년간 하루에 증발량 수치를 저장하여서 증발량이 최대일 때 6시간 최소일 때 24시간으로 잡고 그날에 증발량을 이용하여 완전히 증발하는데 걸리는 시간을 비례적으로 계산하여 적용하였다[20].

$$Y = -4.5X + 28.5 \quad (3)$$

식 (3)은 증발량과 완전히 증발하는데 걸리는 시간에 대한 관계식이다. Y는 증발하는데 걸리는 시간 X는 증발량(mm)

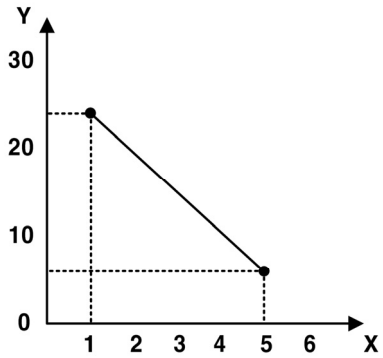

그림 6. 증발량-증발하는데 걸리는 시간 그래프.

Fig. 6. Graph for the evaporation rate vs time.

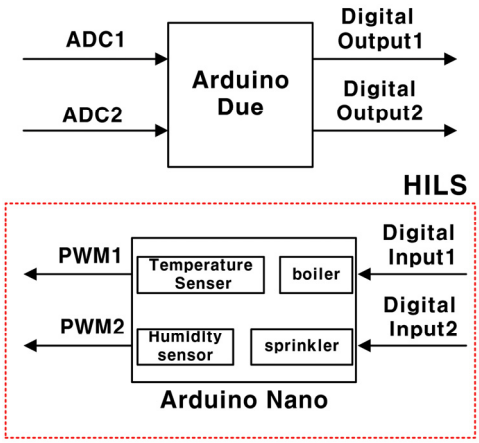

그림 7. 비닐하우스의 HILS 개념도.

Fig. 7. Conceptual diagram for the vinyl greenhouse HILS.

이다. 그림 6은 증발량에 따른 증발하는데 걸리는 시간 그래프이다.

이러한 가정을 가지고 그림 7과 같이 HILS를 구현해 보았다. Arduino Due에서 ADC로 측정한 온도 및 습도정보를 사용자가 판단하여 보일러 스프링클러를 작동하는 스위치를 누를 수 있다. 이 신호는 digital output으로 출력되고 Arduino Nano에서 digital input으로 받는다. 비닐하우스 HILS인 Arduino Nano에서는 위에서 설명한 방법으로 온도와 습도를 조절하게 되고 이는 센서를 통해 출력되는 analog 값으로 출력해야 하지만 Arduino Nano에는 DAC가 없으므로 PWM으로 출력 후 LPF (Low Pass Filter)를 통하여 analog 출력을 만들어낸다. 여기서 사용한 LPF의 회로는 그림 8과 같다.

다음은 비닐하우스를 HILS로 구현된 시스템을 연결하여 측정한 결과이다. 그림 9는 전체적인 모니터링 GUI이고 그림 10은 ADC 부분을 확대 버튼을 눌러서 확대하여 확인한 결과이다. 빨간 선(진한 선)은 온도이며 평상시에는 외부 기온에 영향을 받다가 급상승하는 시작 부분에 보일러를 켰다고 입력받아서 온도가 상승하여 23°C를 맞추게 된다. 파란 선(연한 선)은 토양습도이며 물을 주지 않았을 때는 건조한 상태로 있다가 그래프가 떨어지는 부분에서 물을 주어 낮아졌다가 증발하면서 다시 서서히 높아지는 것을 볼 수 있다.

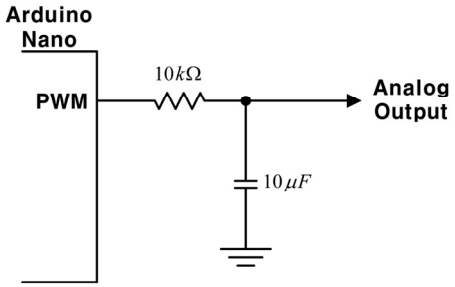

그림 8. Arduino Nano의 DAC 기능구현을 위한 저역 필터.

Fig. 8. Low pass filter for DAC implementation in Arduino Nano.

2. 실내 온실 시스템

햇빛이 차단되고 인공조명으로 키우는 온실에 화초를 재배한다고 생각해본다. 우선 식물을 키우는데 있어서 제일 중요한 요소들 중에 온도와 습도 그리고 조도만을 주기적으로 체크해야 되는 시스템을 구성해 본다. 이러한 시스템에서 온실 안에서는 온도와 습도, 조도를 측정하고 온실 밖에서 측정한 값들을 볼 수 있으며 이를 통하여 사람이 주기적으로 확인하여 물을 주거나 빛의 밝기를 조절할 수 있도록 한다.

이러한 가정을 가지고 그림 11과 같이 HILS를 구현 하였다. 그림은 Arduino Due와 HILS를 구현한 Arduino Nano 간의 입력과 출력을 나타낸다. Arduino Due와 Raspberry Pi의 연결은 위에서 설명한 것 같이 SPI 통신으로 이루어지며 이는 그림에서는 생략한다. Arduino Due에 출력에선 PWM의 duty로 온도를 조절하고 DAC로 조도를 조절하고 digital output으로 스프링클러를 작동시켜 물을 공급한다. 이는 HILS인 Arduino Nano에서 입력으로 받는데 PWM과 DAC는 외부 LPF를 거쳐서 ADC로 들어오고 digital output은 digital input으로 들어오게 된다. 출력값은 PWM 두 개와 quadrature pulse 한 개로 PWM에서는 온도와 습도 값이 duty 값으로 나오고 이는 LPF를 통해 Arduino Due로 전달되어 ADC를 통해 읽어진다. Quadrature pulse는 digital output pin 두 개를 이용하여 만들어 내고 Arduino Due의 encoder counter를 통해서 읽어진다. 조도 값의 경우 PWM을 이용하여도 되지만 ADC를 두 개의 채널만 만들어 놓았고 encoder counter를 검증하기 위하여 quadrature pulse를 사용했다. 마지막으로 토양이 건조해 지면 경고를 알리기 위해 digital output으로 신호를 보내 Arduino Due에서 digital input으로 받아서 led에 연결하여 경고 신호를 보내게 된다.

온도는 거의 변화 없다고 가정하고 실내 온도 23°C를 기준으로 온실 안의 온도가 높으면 비례해서 떨어지고 온도가 낮으면 비례해서 올라가도록 한다. 여기에 사람이 온실 밖의 패널을 통해서 조절하면 그에 따라 값이 변화하게 된다. 토양습도는 물을 주었을 때부터 자연적으로 증발하게 되는데 12시간 정도면 완전히 마르게 된다. 증발량은 선형적으로 식을 세워 시간에 따라 조금씩 증발할 수 있도록 한다. 사람이 온실에 들어와서 직접 확인하는 것이 아닌 밖에서 습도를 확인한 후에 물을 주는 여부를 판단할 수 있게 할 수 있다. 조도는 식물의 영향을 받지 않고 오로지 광채의 영향만 받으므로 패널에서 조절한 조도의 값만 변화하도록 한다.

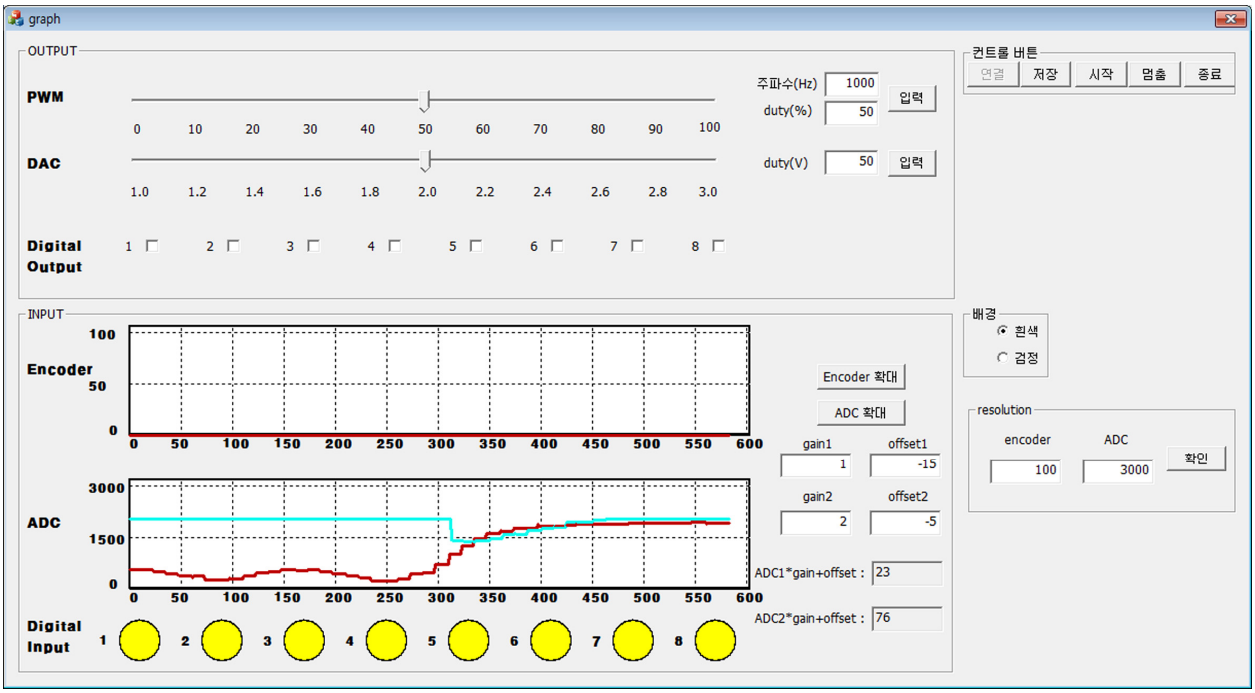

그림 9. 비닐하우스 HILS의 테스트 결과.

Fig. 9. Test results of vinyl greenhouse HILS.

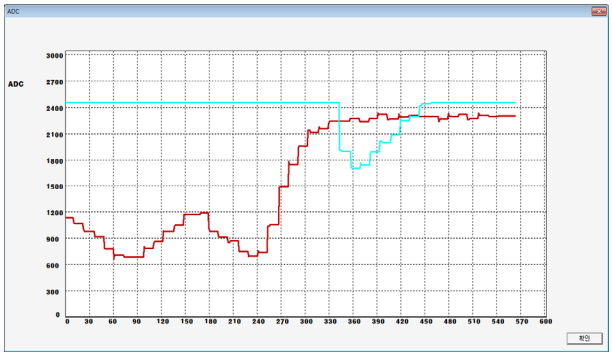

그림 10. ADC 결과의 확대 그래프.

Fig. 10. Expanded graph of ADC result.

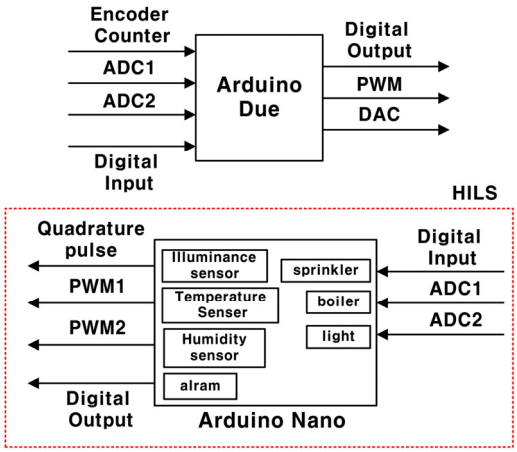

그림 11. 실내 온실의 HILS 개념도.

Fig. 11. Conceptual diagram for the indoor greenhouse HILS.

그림 12는 실내 온실 시스템을 HILS로 구현한 것으로 연결하여 측정한 결과이다. Encoder counter 값은 조도 값으로 50% 근처에서 움직여서 측정해 보았다. ADC 부분에서는 전 실험과 마찬가지로 빨간 선(진한 선)은 온도이며 23°C 부근에 있고 파란 선(연한 선)은 습도이며 물을 주었을 때 내려갔다가 증발해서 올라오는 것을 볼 수 있다. 이때 물이 건조했을 때는 digital input 1번으로 경고등이 켜지게 된다.

IV. 결론

본 논문에서는 오픈소스 하드웨어와 IoT 디바이스를 이용하여 원격으로 제어 및 모니터링이 가능한 통신 시스템을 제안하였다. 제안된 시스템은 오픈소스 하드웨어를 이용하여 개발비용을 절감하였다. Raspberry Pi에서는 이더넷 기능을 이용하고 부족한 주변 장치를 Arduino Due를 연결하여 기본적인 입출력 기능을 구현하였다. Arduino Due는 ADC 2 channel 그리고 DAC, encoder counter, PWM, 각각 1 channel 그리고 D/I 핀 8 channel을 구현했다. 또한 자체 개발된 윈도용 GUI를 통하여 구현된 기능을 제작하고 사용자가 쉽게 프로그램에 접근하여 사용할 수 있도록 제작하였다. 제안된 시스템을 실험해보기 위하여 HILS를 이용한 비닐하우스 시스템과 실내 온실 시스템을 구현하였다. 그리고 구현한 HILS를 이용하여서 제안된 시스템을 테스트해보았다. 제안된 시스템의 기능을 이용하면 가정이나 사무실, 공장 등에서도 다양한 방면으로 이용될 수 있을 것이다.

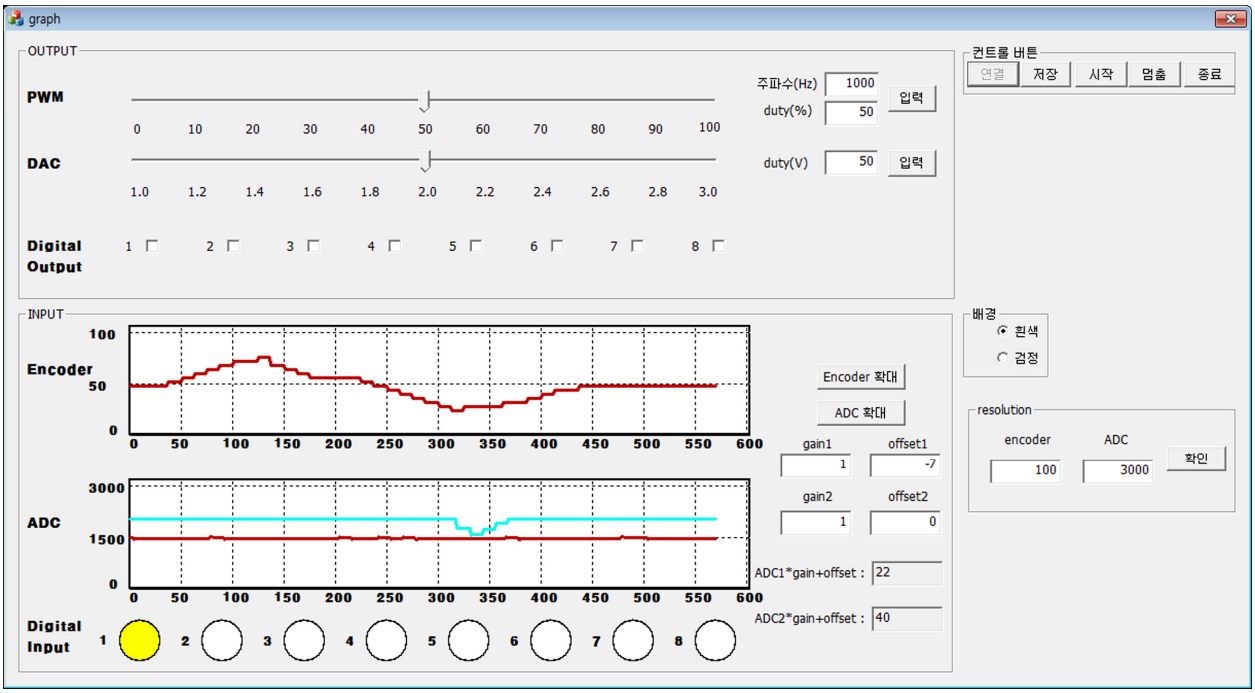

그림 12. 실내 온실 HILS의 테스트 결과.

Fig. 12. Test results of indoor greenhouse HILS.

REFERENCES

[1] S. W. Lee, S. M. Park, and K. B. Sim, "One time password-based SEED algorithm for IoT systems," *Journal of Institute of Control, Robotics and Systems (in Korean)*, vol. 22, no. 9, pp. 766-772, 2016.

[2] K. Delic and J. Riley, "Current and future trends in AI," *Information, Communication and Automation Technologies (ICAT), 2013 XXIV International Symposium on. IEEE*, 2013.

[3] J. A. Stankovic, "Research directions for the internet of things," *IEEE Internet of Things Journal*, vol. 1, no. 1, pp. 3-9, 2014.

[4] R. Itschner, C. Pommerell, and M. Rutishauser, "GLASS: remote monitoring of embedded systems in power engineering," *IEEE Internet Computing*, vol. 2, no. 3, pp. 46-52, 1998.

[5] C. O. Park, K. K. Ahn, and I. S. Song, "A study of web-based remote pneumatic servo control system using java language," *Journal of Control, Automation, and Systems Engineering*, vol. 9, no. 3, pp. 196-203, 2003.

[6] D. G. Seo, H. S. Ko, and Y. D. Noh, "Design and implementation of digital door lock by IoT," *KIISE Transactions on Computing Practices*, vol. 21, no. 3, pp. 215-222, 2015.

[7] C. S. Lee, Y. T. Han, S. B. Jeon, D. M. Seo, and I. B. Jung, "Smart parking system using ultrasonic sensor and bluetooth communication in internet of things," *KIISE Transactions on Computing Practices*, vol. 22, no. 6, pp. 268-277, 2016.

[8] T. H. Yeom, S. M. Park, H. I. Kwon, D. K. Hwang, and J. C. Kim, "A smart farming system based on visible light communications," *The Journal of Korean Institute of Communications and Information Sciences*, vol. 38C, no. 5, pp. 479-485, 2013.

[9] H. J. La, M. K. Kim, and S. D. Kim, "A health assessment platform with IoT devices," *KIISE Transactions on Computing Practices*, vol. 22, no. 5, pp. 255-234, 2016.

[10] J. W. Kim, "A smart home prototype implementation using raspberry Pi," *The Journal of the Korea Institute of Electronic Communication Sciences*, vol. 10, no. 10, pp. 1139-1144, 2015.

[11] C. S. Oh, M. S. Seo, J. H. Lee, S. H. Kim, Y. D. Kim, and H. J. Park, "Indoor air quality monitoring systems in the IoT environment," *The Journal of Korean Institute of Communications and Information Sciences*, vol. 40, no. 5, pp. 886-891, 2015.

[12] J. S. Choi and Y. S. Lee, "The implementation of a hardware-in-the-loop simulator for an inverted pendulum system using open-source hardware," *Journal of Institute of Control, Robotics and Systems (in Korean)*, vol. 23, no. 2, pp. 117-125, 2017.

[13] Y. B. Min, T. S. Chung, and J. K. Ha, "A simulation of temperature control of greenhouse with hot-water heating system," *Journal of Bio-Environment Control*, vol. 8, no. 3, pp. 152-163, 1999.

[14] J. S. Lee, Y. D. Kim, and I. Y. Moon, "Design and im-

plementation of real-time surveillance system," *Journal of Advanced Navigation Technology*, vol. 12, no. 1, 2008.
[15] H. J. Yoo, "Developing an IT course utilizing raspberry Pi," *Journal of Practical Engineering Education*, vol. 7, no. 2, pp. 89-95, 2015.
[16] P. S. Choi, "Visual C++ MFC Window Programming", 2nd Ed., Daallbook, 2013.
[17] J. H. Kim, S. G. Kang, K. B. Shin, and Y. J. Lee, "A development of automated design and structural analysis aided-program based on GUI environment for aluminum extrusion carbody structures of railway vehicle for design engineers," *Journal of the Korean Society for Railway*, vol. 15, no. 4, pp. 323-328, 2012.
[18] http://www.kma.go.kr.
[19] J. M. Choi and S. W. Cho, "The characteristic of convective heat transfer coefficient by natural heat transfer coefficient and forced heat transfer coefficient," *Journal of the Architectural Institute of Korea Planning & Design*, vol. 27, no. 6, pp. 205-212, 2011.
[20] B. Y. Lee, "Short-term variation in class a pan evaporation," *Atmosphere. Korean Meteorological Society*, vol. 12, no. 3, pp. 196-199, 2002.

김 태 윤

2016년 인하대학교 전기공학과 졸업
2016년~현재 인하대학교 전기공학과 석사과정 재학 중. 관심분야는 사물인터넷(IOT), 윈도우 프로그래밍, 임베디드 시스템.

이 영 삼

제어 · 로봇 · 시스템학회 논문지, 제15권 제4호 참조.