



저작자표시-비영리-변경금지 2.0 대한민국

이용자는 아래의 조건을 따르는 경우에 한하여 자유롭게

- 이 저작물을 복제, 배포, 전송, 전시, 공연 및 방송할 수 있습니다.

다음과 같은 조건을 따라야 합니다:



저작자표시. 귀하는 원저작자를 표시하여야 합니다.



비영리. 귀하는 이 저작물을 영리 목적으로 이용할 수 없습니다.



변경금지. 귀하는 이 저작물을 개작, 변형 또는 가공할 수 없습니다.

- 귀하는, 이 저작물의 재이용이나 배포의 경우, 이 저작물에 적용된 이용허락조건을 명확하게 나타내어야 합니다.
- 저작권자로부터 별도의 허가를 받으면 이러한 조건들은 적용되지 않습니다.

저작권법에 따른 이용자의 권리는 위의 내용에 의하여 영향을 받지 않습니다.

이것은 [이용허락규약\(Legal Code\)](#)을 이해하기 쉽게 요약한 것입니다.

[Disclaimer](#)

碩士學位論文

오픈소스 기반 MCU의 농업적
활용성에 관한 연구

濟州大學校 産業大學院

機械工學科

金 炫 秀

2021年 7月



오픈소스 기반 MCU의 농업적 활용성에 관한 연구

指導教授 玄 明 澤

金 炫 秀

이 論文을 工學 碩士學位 論文으로 提出함

2021年 7月

金炫秀의 機械工學 碩士學位 論文을 認准함

審査委員長 _____ 印

委 員 _____ 印

委 員 _____ 印

濟州大學校 産業大學院

2021年 7月

A Study on The Agricultural Utility of Open Source Based MCU

Kim Hyun-Su
(Supervised by professor Myung-Taek Hyun)

A thesis submitted in partial fulfillment of the requirement for the degree of
Master of Engineering

2021. 7.

This thesis has been examined and approved.

.....
Thesis director, Myung-Taek Hyun,
Prof. of Mechanical Engineering
.....
.....
.....

(Name and signature)

2021. 7.

.....
Date

Department of Mechanical Engineering
GRADUATE SCHOOL OF ENGINEERING
JEJU NATIONAL UNIVERSITY

목 차

I. 서론	1
1. 연구 배경 및 필요성	1
1) 4차 산업혁명의 도래	1
2) 스마트팜의 개념	2
3) 연구의 필요성	3
2. 선행연구 검토	7
II. 재료 및 방법	10
1. 프로세서	10
2. 장치제어모듈	12
3. 농업용 장치	14
4. 하드웨어 제작	16
III. 연구 내용	18
1. 시스템 구성 및 동작 원리	18
2. 웹서버 및 데이터베이스 구축	20
3. 웹서버와 ESP32 통신	26
4. 시스템 농가실증	30
IV. 결론 및 향후 연구	32
참고문헌	34

LIST OF TABLES

Table 1. Comparison between PIC(Micro Controller) and PLC	4
Table 2. Comparison between ESP32 and ESP8266	11
Table 3. Agricultural control device to be applied to the system	15

LIST OF FIGURES

Fig 1. 1st-4th industrial revolution process	2
Fig 2. Composition of smart farm facility horticultural field ..	3
Fig 3. PLC-based smart farm system	5
Fig 4. Arduino Uno (left) and Raspberry Pi (right)	6
Fig 5. SKT Smart Farm Green House	7
Fig 6. SKT Smart Farm Service Configuration Diagram	7
Fig 7. Farm Labs Smart Farm Configuration Diagram	8
Fig 8. Block diagram of ESP32	12
Fig 9. DC motor forward and reverse control circuit	13
Fig 10. Basic operation circuit of solid state relay	14
Fig 11. Hardware configuration diagram	16
Fig 12 Hardware production	17
Fig 13. System configuration diagram	18
Fig 14. System data delivery process	19
Fig 15. Flow chart of data when operating the switch (Example) ..	20
Fig 16. HTTP communication between web server and web client ·	21
Fig 17. Smartphone control screen	21
Fig 18. User interface (UI) design	22
Fig 19. Database management using HeidiSQL(device history table) ··	23
Fig 20. Database management using HeidiSQL (device operation table) ··	23
Fig 21. Code that sends variables to PHP file using Ajax in index.html file ··	24
Fig 22. Code to modify and create database data in device operation.php file ·	25
Fig 23. AP connection between ESP32 and Client	26

Fig 24. PHP file that retrieves the data of the device operation table ·	27
Fig 25. Actuation code of the switch according to the value of the device operation table ··	28
Fig 26. PHP file that modifies the data of the device operation table ··	29
Fig 27. Blueberry Farm Empirical Results	31

SUMMARY

With the growing interest in smart farms that combine ICT technology and agriculture around the world, research related to smart farms has been activated in Korea. However, it is a reality that even the most basic first-generation smart farm is difficult to introduce because it has the disadvantage that the initial cost is burdened and A/S is difficult for farms to introduce smart farms. Since most of the smart farm systems commercialized in Korea are expensive PLC-based systems that are inefficient for facility agriculture, we attempted to develop a smart farm model using an open source based MCU that is cheaper and easier to handle. This system can control the environment of the house through remote control of agricultural equipment such as air circulation fan. It built its own web server and database using open source software, and the control board transmits and receives data through WiFi communication. In addition, data can be transferred smoothly through PHP. Since the system structure is simple, if only the software source code is tailored to the farmhouse environment, it can be a farm-customized smart farm system for only the functions that the farmhouse wants. As a result of demonstrating the system at a facility house that grows blueberries, the introduction cost was significantly reduced compared to commercially available smart farms, and the satisfaction of the farmers was high. As MCU performance continues to improve around the world, it is expected that low-cost smart agriculture can be realized with low cost and simple configuration when an open source based system is developed.

I. 서 론

1. 연구 배경 및 필요성

1) 4차 산업혁명의 도래

최근 전 세계적으로 4차 산업혁명 시대가 열리면서 많은 산업 전반에 걸쳐 정보통신 기술 기반의 디지털 연결성 변화로 급속히 산업의 패러다임이 바뀌고 있다.

4차 산업혁명이란 정보통신기술(ICT)의 융합으로 이루어지는 차세대 산업혁명이다[1]. 1차 산업혁명은 기계의 발명으로 인한 자동화의 탄생과 증기기관의 발명을 통한 국가 내의 연결성 강화를 그 특징으로 했다. 2차 산업혁명은 전기 등 에너지원의 활용과 작업의 표준화를 통해 기업 간, 국가 간 노동부문의 연결성을 강화하고 대량생산체제의 수립한 것이 그 특징이다. 3차 산업혁명은 전자장치, ICT를 통하여 급진적인 정보처리 능력의 발전, 이를 바탕으로 사람, 환경, 기계를 아우르는 연결성이 강화되는 변화를 의미하며 4차 산업혁명은 인공지능에 의해 자동화와 연결성이 극대화되는 단계를 의미한다.

Fig. 1과 같이 4차 산업혁명의 핵심 요소는 개별적으로 발달한 각종 기술의 ‘융합’이라고 할 수 있다. 디지털, 바이오, 오프라인 기술들이 다양하고 새로운 형태로 융합되어 새로운 부가가치를 창출해내는 것이다. 4차 산업혁명은 제품, 설비, 인간이 연결되는 ‘사물인터넷(IoT: Internet of Things) 혁명’으로 표현할 수 있다[2].

Han 등[3]에 의하면 사물인터넷이란 사람이나 사물 프로세스 등의 모든 객체들이 인터넷으로 연결되어 소통하는 미래 네트워크 기술로 정의되며, 스마트홈, 스마트리테일, 스마트팩토리 등 다양한 산업현장에서 활용된다고 하였다.

이 흐름은 최근 농업에서도 활발히 이루어지고 있으며 농업과 ICT기술을 융합한 ‘스마트팜(Smart farm)’ 기술이 주목받고 있다.



Fig 1. 1st-4th industrial revolution process

2) 스마트팜의 개념

스마트팜은 PC 또는 모바일을 통해 온실의 온·습도, 이산화탄소 등 작물 재배 환경을 모니터링하고 창문 개폐, 영양분 공급 등을 원격 자동으로 제어하며 재배하는 작물의 최적 생육 환경을 유지 관리할 수 있는 농장으로 정의된다[4].

스마트팜의 주요 기능으로 최근 사회문제 중 하나인 농업 현장의 고령화, 인력난 등을 해소할 수 있으며, 농장이 아닌 다른 장소에 있어도 인터넷만 연결되어 있다면 어디서든 농장의 환경을 모니터링하고 제어할 수 있다는 간편함도 있다.

Kim 등[5]에 의하면 전 세계적으로 네덜란드를 비롯해 미국, 일본, 중국 등 지능정보 및 ICT를 활용해 산업 경쟁력을 높이고 부가가치를 창출하기 위해 스마트팜에 대한 연구가 활발히 진행 중이다. 세계 스마트팜 시장규모는 2022년에 한화 약 491조에 이를 것으로 전망되며, 한국의 경우 2017년 4조 4,493억 원에서 연평균 5%씩 성장해 2022년에는 5조 9,588억 원의 규모를 형성할 것으로 전망된다고 하였다.

한국형 스마트팜의 기본적인 구성은 Fig. 2와 같이 온·습도, CO₂, 강우 등 센서와 환기, 난방, 차광커튼 등의 시설별 제어 및 통합 제어 장비가 갖추어져 있다 [4].

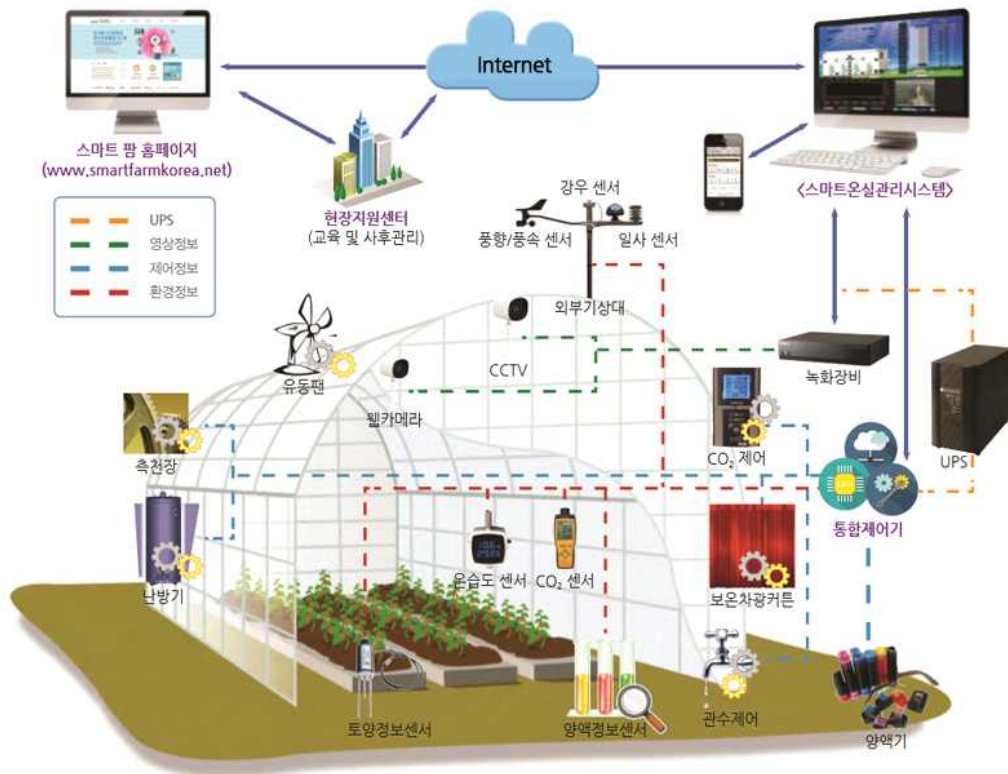


Fig 2. Composition of smart farm facility horticultural field

3) 연구의 필요성

현재 상용화된 스마트팜 시스템은 대부분 높은 초기 비용이 필요하기 때문에 접근성이 매우 떨어진다는 단점이 있다. Fig. 3과 같이 국내 상용화된 대부분의 스마트팜 제어시스템은 상대적 고가의 PLC(Programmable Logic Controller) 기반의 제어시스템이다. PLC는 전용 소프트웨어를 통해 가상의 릴레이 시퀀스 회로를 만드는 것이므로 배선이나 릴레이 등 물리적인 변경점 없이 제어회로의 수정이 가능하다는 장점이 있다. 하지만 Table. 1과 같이 PLC기반 제어시스템은 PIC시스템에 비해 제조사 간 표준화가 되어 있지 않아 호환성이 떨어지고 제조사마다 다른 프로그램 언어를 사용해 사용자가 익히기도 쉽지 않다[6]. PLC는 많은 산업 분야에서 활용되고 있지만, 농업에서는 농가별, 재배작목별 수요가 다르고 특히 토경 과수재배농가와 같이 제한적인 환경 데이터 측정과 농업용장치의 제어목적이라면 PLC시스템이 투자비용 대비 비효율적일 수 있

다.

이러한 문제점들로 인해 저렴하면서도 제품 간의 호환성, 신뢰성 등 농가의 요구 사항에 맞는 시스템 제작이 필요하다.

Table 1. Comparison between PIC(Micro Controller) and PLC

Properties	PIC (Micro Controller)	PLC
Cost	Cheap	Expensive
Size	Smaller	Bulky
Board Area	Can be reduced	Need large area compare to PIC
Programming	Easy to change programme	Changing the program requires physically
Handle Complex Command At One Time	Easier	Difficult
Ease of Design	Simpler	Harder
Can choose any I/o as output	Yes	No
Required Voltage	5V	12V, 24V
Maintenance Cost	Low	High (Require Specialist)

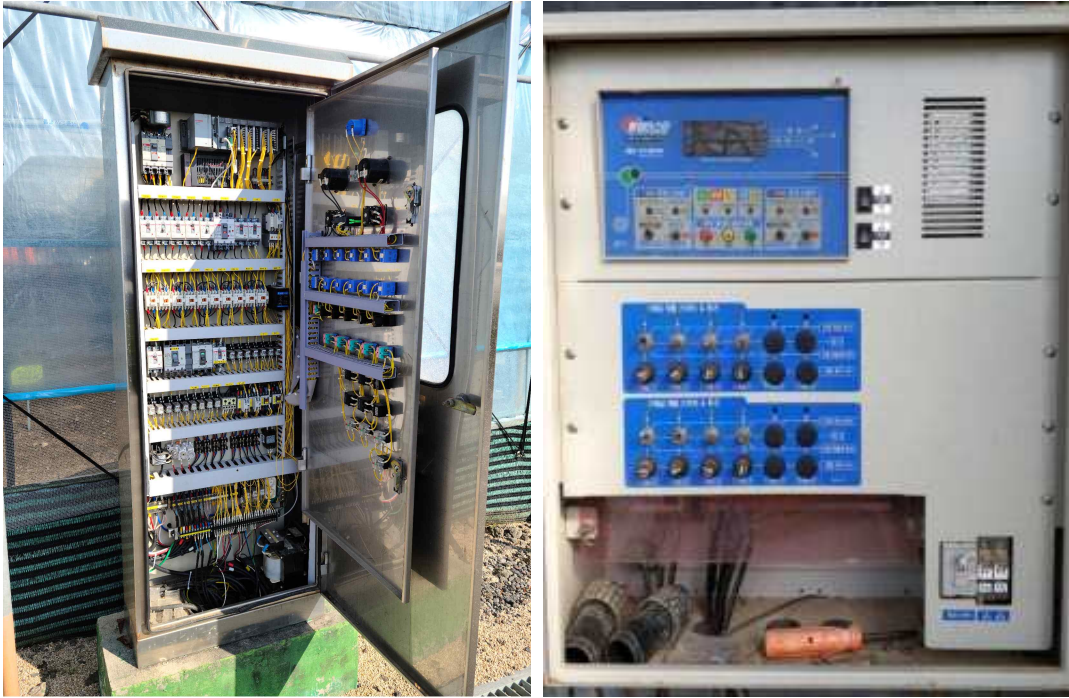


Fig 3. PLC-based smart farm system

본 연구에서는 앞서 언급한 여러 문제를 해결하기 위해 누구든지 쉽고 저렴하게 구할 수 있는 오픈소스(Open source) 기반 MCU(Micro Controller Unit)를 내장한 기기 제어용 보드를 이용하여 스마트팜을 구축하는데 목적이 있다. Frolund 등[7]에 의하면 오픈소스란 특정 제품 제작에 필요한 기술 정보를 일반 대중에게 공개하여 누구나 필요한 제품을 만들 수 있도록 하는 것이다.

오픈소스 하드웨어는 전문가가 아니더라도 일반 사용자들이 자신의 요구사항에 맞춰 직접 자신만의 사물(Thing)을 만들고 이를 통해 스마트 사물인터넷 환경을 설계하는 ICT(Information and Communications Technologies), DIY(do-it-yourself) 패러다임을 제공할 수 있어 누구라도 사물인터넷 서비스 플랫폼을 구축할 수 있게 해준다고 하였다. 즉, 오픈소스는 하드웨어 구성과 목적에 맞게 구동하는 응용프로그램 등 소스코드를 무상으로 공개함으로써 사용자에 의해 공유·논의되어 발전시키는데 목적을 둔다.

Fig. 4와 같이 대표적인 오픈소스 하드웨어는 아두이노, 라즈베리파이 등이 있다. 일반적으로 AVR 프로그래밍은 winAVR로 컴파일하여, ISP장치를 통해

업로드를 해야하는 등 번거로운 과정을 거쳐야 하는데, 아두이노 등은 컴파일된 펌웨어를 USB를 통해 업로드를 쉽게 할 수 있다[8]



Fig 4. Arduino Uno (left) and Raspberry Pi (right)

본 연구에서는 Fig. 2과 같은 스마트팜 기본적인 구성에서 온·습도, 강우 센서 등과 천·측창 개폐, 공기순환팬, 전자밸브 등 시설하우스 환경관리에 필요한 기능을 오픈소스 기반 MCU를 내장한 기기제어용 보드를 이용하여 구현하고 농업 현장에서 활용성 향상을 도모하기 위해서 수행하였다.

2. 선행연구 검토

Fig. 5는 SKT 지능형 비닐하우스이다. 이 시스템은 SK 이동 통신망으로 자체 서버와 통신을 하여 하우스의 개폐, 관수, 보일러 등을 제어할 수 있다. 스마트팜 시스템에서 개폐기, 전자밸브, 난방기와 연결하여 원격으로 작동을 할 수 있으며 온습도 센서 등 환경 데이터를 수집하여 사용자에게 통계 정보를 전송한다.

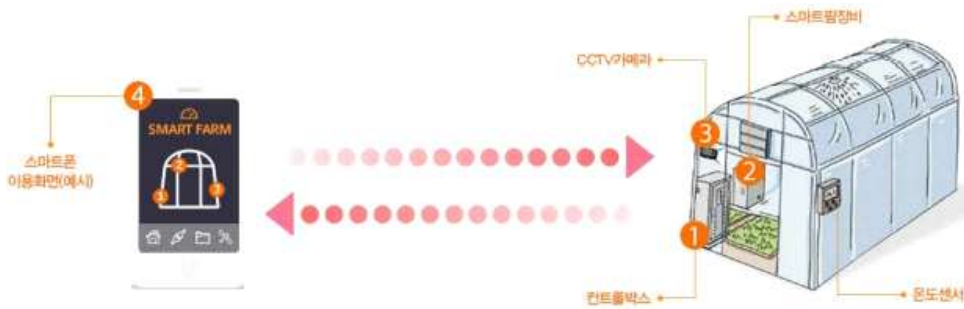


Fig 5. SKT Smart Farm Green House

Fig. 6은 SKT 스마트팜 서비스 구성도이다. 해당 시스템은 SKT 이동통신망에 의해 개인 스마트폰에서 비닐하우스까지 제어데이터가 송·수신되며 개인 스마트폰을 통해 하우스의 환경 데이터를 조회할 수 있다.



Fig 6. SKT Smart Farm Service Configuration Diagram

SKT 스마트팜은 주요 6가지 기능이 있다. ① LTE 통신망을 이용하여 전용 앱을 통해 어디서나 제어, ② CCTV 카메라로 하우스 제어 과정, 작물의 생육 상태 등을 원격 모니터링, ③ 천·측창, 보온덮개, 전자밸브 등을 스마트폰으로 원격제어, ④ 실시간 정전 및 침입 알림과 고온 또는 저온경보 등 알림 서비스 제공, ⑤ 각종 센서 정보를 전용앱에서 실시간 확인 및 통계정보 제공, ⑥ 시스템의 높은 호환성, 확장성, 소형화로 설치가 용이함 등이 있다. 이 시스템은 LTE 통신망을 이용하여 통신한다는 점 외에 원격으로 앱을 통한 제어, 모니터링, 센서 데이터 활용 등은 타 스마트팜과 비슷하다고 판단된다.

Fig. 7는 팜랩스라는 스마트팜 제조업체의 시스템 구성도이다. 이 시스템은 메인 컨트롤박스에서 WiFi 통신을 이용하여 CCTV, 전동기, 환풍기 등을 제어한다. 이 시스템은 전국에 약 130농가에 적용되어 있다. 팜랩스 스마트팜 시스템은 과수원, 온실, 축사에 적용 가능하며 제어부분에 있어서 타사 스마트팜 시스템과 별다른 차이가 없다. 이 시스템만의 특징은 생육 환경 변화와 기기 제어 정보가 기록되는 기능이 있다. 하지만 이 기능도 다른 스마트팜 장비에서 지원되고 있으므로 차이점은 거의 없다.



Fig 7. Farm Labs Smart Farm Configuration Diagram

앞에서 설명한 것처럼 최근 여러 업체에서 스마트팜 시스템을 개발하고 있지만 시스템 간에 기능은 큰 차이점이 없다. 또한 이러한 시스템은 월정액과 같은 정기적인 요금을 지불해야하고 특정 사업자 전용의 스마트 기기를 사용해야 하는 경우가 많으며, 기기의 종류나 기능이 사용자 요구 수준에 비해 제한적이라는 문제점을 갖고 있다. 이러한 문제점들로 농가 현장에 저렴한 제품을 사용하면서도 제품의 호환성 등 농가의 요구 사항에 맞는 스마트팜 시스템을 개발해야 할 필요가 있다고 판단된다.

II. 재료 및 방법

1. 프로세서

우리가 흔히 말하는 프로세서(processor) 또는 CPU(Central Processor Unit)는 제어장치, 연산장치, 레지스터 그리고 데이터 버스로 구성된 디지털 시스템의 핵심 부분으로, 프로그램을 기억 장치로부터 읽어 연산 처리, 비교처리, 데이터 전송, 편집, 변환 등의 데이터 처리를 하고, 각종 장치를 구동하는 역할을 담당한다.

마이크로프로세서(Micro-processor)는 1개의 조그마한 IC(Integrated Circuit) 칩 속에 CPU의 모든 내용을 내장한 칩을 말한다. 이와 같이 CPU의 모든 내용이 하나의 칩 속에 내장됨으로써 가격이 훨씬 싸고, 부피가 줄어든다는 장점이 있다[9].



본 연구에서는 시스템 개발보드로 ESPRESSIF사의 ESP32를 이용하여 스마트팜 시스템을 설계하고 구현하였다. 오픈소스 기반 MCU 대표적인 제품인 Arduino Uno와 비교하여 ESP32는 크기가 매우 작고 와이파이 통신모듈이 내장되어있어 저렴한 가격으로 네트워크 통신이 가능하다[10]. 또한 Table. 2와 같이 ESP32는 CPU 속도(Clock Speed)가 160MHz로 직전 모델인 ESP8266(80Mhz)와 Arduino Uno보다(16MHz) 훨씬 빠르며, Flash Memory 또한 ESP32(16MB)가 Arduino Uno(32KB)보다 크다. ESP32와 ESP8266의 아키텍처는 32bits로 Arduino Uno모델(8Bits)보다 성능이 좋고 RAM의 크기도 ESP32와 ESP8266이 각각 512KB, 160KB로 Arduino Uno모델(2KB)보다 훨씬 성능이 좋다. 이와 같이 Arduino Uno의 용도는 학생들의 교육용과 제품 개발 관련 프로토타입 모델개발을 위해 자주 쓰이지만 실제 산업현장에서는 개발 규모에 따라 ESP32와 ESP8266을 선택해서 사용한다.

ESP32와 ESP8266의 통합개발환경(IDE)은 ArduinoIDE에서도 호환되어 제품

개발의 편의성, 접근성의 뛰어나다는 장점도 있다.

본 연구에서는 온·습도, 강우감지 등 여러 센서 등을 이용하고 그 센서 신호들로 개폐기, 환풍기들의 기기를 제어할 목적으로 상대적으로 성능이 좋은 ESP32가 시스템 개발에 적절한 제품이라고 판단하여 선택했다.

Table 2. Comparison between ESP32 and ESP8266

Properties	ESP32	ESP8266
Number of Cores	2	1
Architecture	32Bit	32Bit
CPU Frequency	160MHz	80MHz
WiFi	YES	YES
BLUETOOTH	YES	NO
RAM	512KB	160KB
FLASH	16MB	16MB
GPIO PINS	36	17
Busses	SPI, I2C, UART, I2S, CAN	SPI, I2C, UART, I2S
DAC Pins	2	0
ADC Pins	18	1
Size(mm)	25.0 × 48.0	25.0 × 48.0
Image		

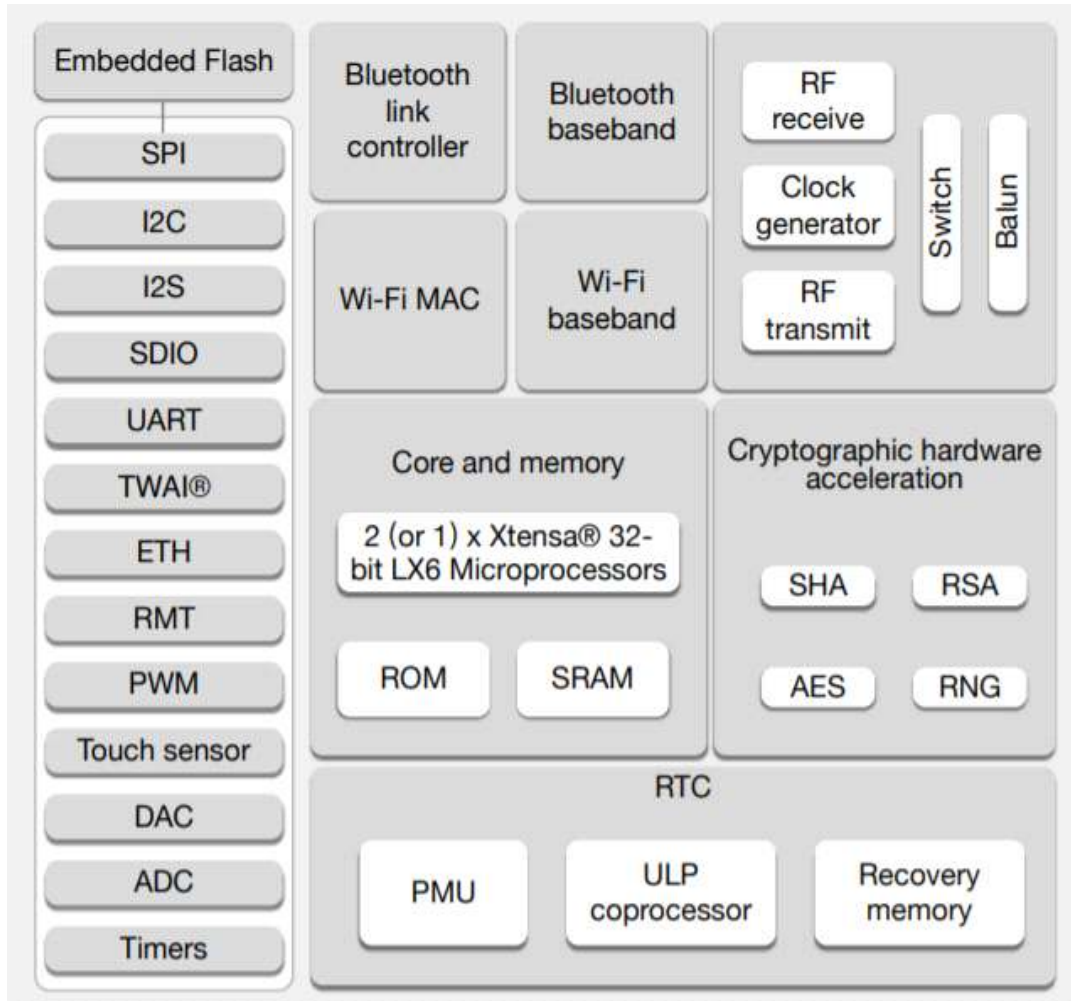


Fig 8. Block diagram of ESP32

2. 장치제어모듈

농업 현장에서는 주로 개폐기, 전자밸브 등과 같은 직류(Direct Current)모터와 공기순환팬, 환풍기 등과 같은 교류(Alternating Current)모터를 사용한다. 대부분의 농가에서는 개폐기 등과 같은 DC모터를 2개의 마그네틱커넥터를 이용하여 정·역 방향 제어를 하고 있는데 이와 같은 방법은 컨트롤박스의 규모가 커지고 재료비용이 증가하며 만약을 대비한 회로의 수정이 번거로운 문제점이 있다. DC모터는 편이 2개 밖에 없으므로 모터 회전의 정역제어를 위해서 따로

회로를 구성해야한다. Fig. 9는 정역 제어 회로로 모양이 알파벳 H 모양과 비슷하다고 해서 H-Bridge 회로라고 불린다. S1과 S4가 닫힘으로써 모터의 왼쪽이 +, 오른쪽이 -가 되며, S2와 S3이 닫힘으로써 왼쪽이 -, 오른쪽이 +가 된다. 이를 응용하여 쉽게 DC모터의 회전방향 및 속도를 제어할 수 있도록 한 것이 모터드라이버이다[11]. 모터는 보통 MCU보드가 허용하는 전류보다 전류를 많이 사용하기 때문에 추가적인 회로가 필요하며 모터드라이버가 그 역할도 수행한다. MDD10A 모터드라이버는 5~30V에 해당되는 DC모터 2개를 제어할 수 있으며, 각 모터 당 10A의 연속 전류를 허용하여 비교적 저렴한 가격으로 고용량의 모터를 제어할 수 있다.

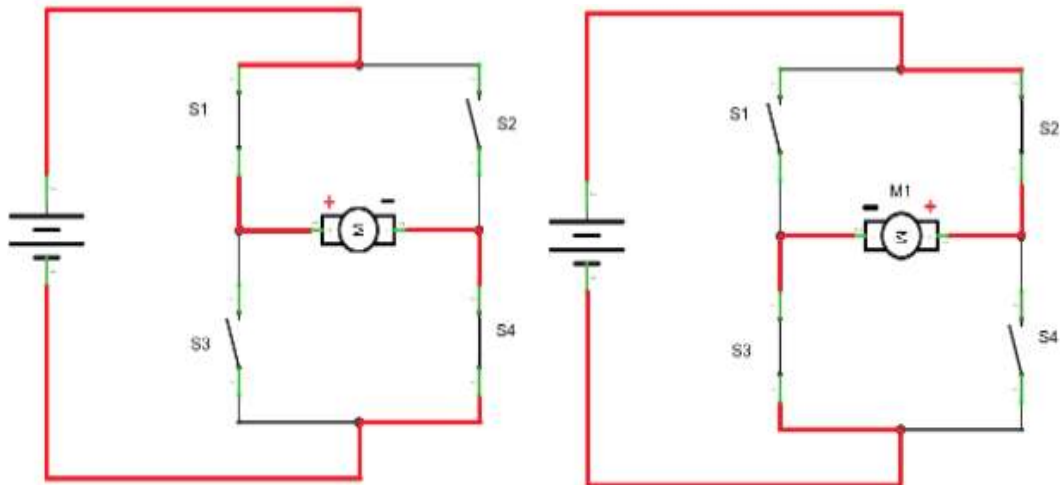


Fig 9. DC motor forward and reverse control circuit

그리고 공기순환팬, 환풍기 등과 같은 AC모터의 ON,OFF 제어하기 위하여 MCU보드와 원활하게 호환이 되는 릴레이가 필요하다. 본 연구에서는 기존의 기계식 릴레이는 전류 ON, OFF제어를 위해 전자석으로 접점을 붙이고 떨어뜨리는 과정에서 발생하는 딸각하는 소음과 접점 부분이 마모되는 단점을 보완한 무접점(Solid State)릴레이를 사용하였다. 무접점 릴레이는 MCU의 소용량 출력신호를 이용하여 대용량의 전기기기를 제어하기 위한 릴레이의 한 종류로써 소음이 전혀 발생되지 않으며 수명도 훨씬 길고 전류의 스위칭 속도가 빠르

다는 장점이 있다. 무접점 릴레이의 기본 작동방식은 Fig. 10과 같이 주로 스위치가 ON되면 발광다이오드에 전류가 흐르고, 광학적으로 결합시킨 포토트랜지스터가 동작한다. 그리고 제로크로스회로가 동작해 교류전원접압의 제로전압 근방에서 출력회로의 트라이악이 도통된다. 전원에서 트라이악을 통해 부하에 전류가 통하는데, 다음으로 스위치가 OFF되면 무접점릴레이는 트라이악의 동작특성에 따라 부하전류의 제로점 근방에서 차단된다. 무접점릴레이는 단락보호, 과부하 보호 및 과열 보호기능을 가질 수 있다[12].

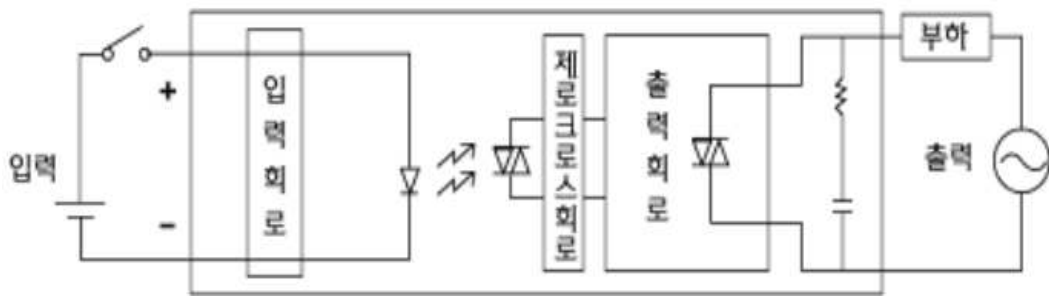


Fig 10. Basic operation circuit of solid state relay

3. 농업용 장치





본 연구에서 개발하는 시스템은 토경 시설과수인 키위를 재배하는 하우스에서 수행하였기 때문에 시설과수에 적합한 농업용 장치들을 제어하고자 한다.

본 시스템에 적용한 농업용 장치는 Table. 3과 같이 농가에서 흔히 사용하는 제품인 개폐기, 공기순환팬, 환풍기 등의 전동기 종류이다. 본 연구에서 다루는 농업용 장치는 농가마다 수요가 다르고 AC모터와 DC모터별로 다루는 원리는 거의 비슷하기 때문에 해당 장치 외의 양수기 등 수경재배용 장치 등도 해당 모델에 적용이 가능하다.

이 전동기 중 하우스 천·측창, 전자밸브 등 DC모터를 제어하기 위한 구동 제어 모듈로 모터드라이버(MDD10A)를 사용하였다. MDD10A를 이용하여 제어

할 농업용 개폐기는 우성하이텍사의 WSM-4035이고 전자밸브는 VALCON사의 AD24-2T이다. 그 외에도 공기순환팬, 환풍기 등과 같은 AC모터는 무접점(Solid State)릴레이모듈(SZH-RLBG-043)과 마그네트커넥터를 이용하여 제어하였다.

Table 3. Agricultural control device to be applied to the system

Properties	Ventilation fan (KMI-200K6)	Air circulator (SG-M220S-4)	Top Ventilator (WSM-4035)	Magnetic Valve (AD24-2T)
Power	AC	AC	DC	DC
Rated Voltage (V)	220/380	220	24	24
Rated Current (A)	2.2/1.3	0.6	2.6	1.0
Power Consumption (W)	400	130	62	2.4
Image				

4. 하드웨어 제작

Fig. 11은 하드웨어 구성도이다. 시스템 동작은 ESP32 제어보드에 내장되어 있는 ESP32 WiFi모듈을 통해 웹서버의 데이터를 전달받아서 장치를 제어한다. ESP32는 개폐기, 전자밸브 등과 같은 DC모터를 제어할 때 모터드라이버를 이용하고, 환풍기와 공기순환팬과 같은 AC모터를 제어할 때 무접점 릴레이와 마그네틱 스위치를 이용하는 구조로 제작하였다.

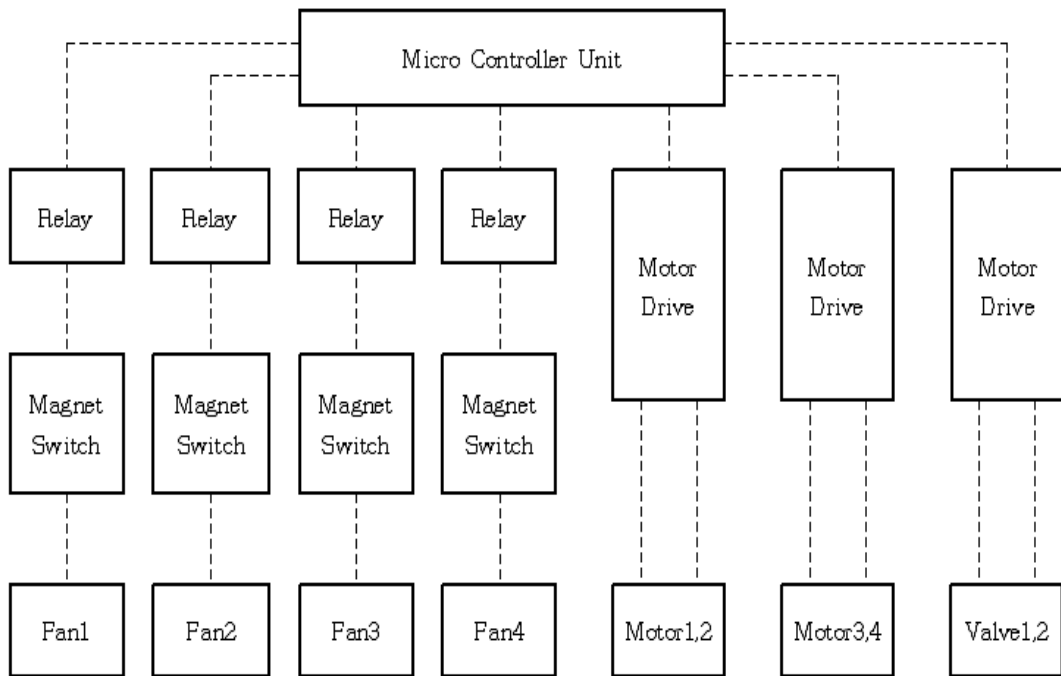


Fig 11. Hardware configuration diagram

Fig. 12와 같이 시스템이 설치된 하우스의 개폐기는 천창 4개, 측창 4개이며 이를 동시에 제어하기 위해서 최소 300W의 220-24V 변압기를 사용하였다. 그리고 제어보드는 웹서버와 WiFi 통신을 하기 때문에 원활한 WiFi 통신을 위하

여 WiFi Expander를 하우스에 추가로 설치하였다. 제어함에는 ESP32 제어보드가 각각 환풍기와 공기순환팬 제어용, 천·측창 개폐기 제어용, 전자밸브 제어용으로 설치하였으며 모터드라이버도 천창 제어용, 측창 제어용, 전자밸브 제어용으로 나누어 설치하였다.

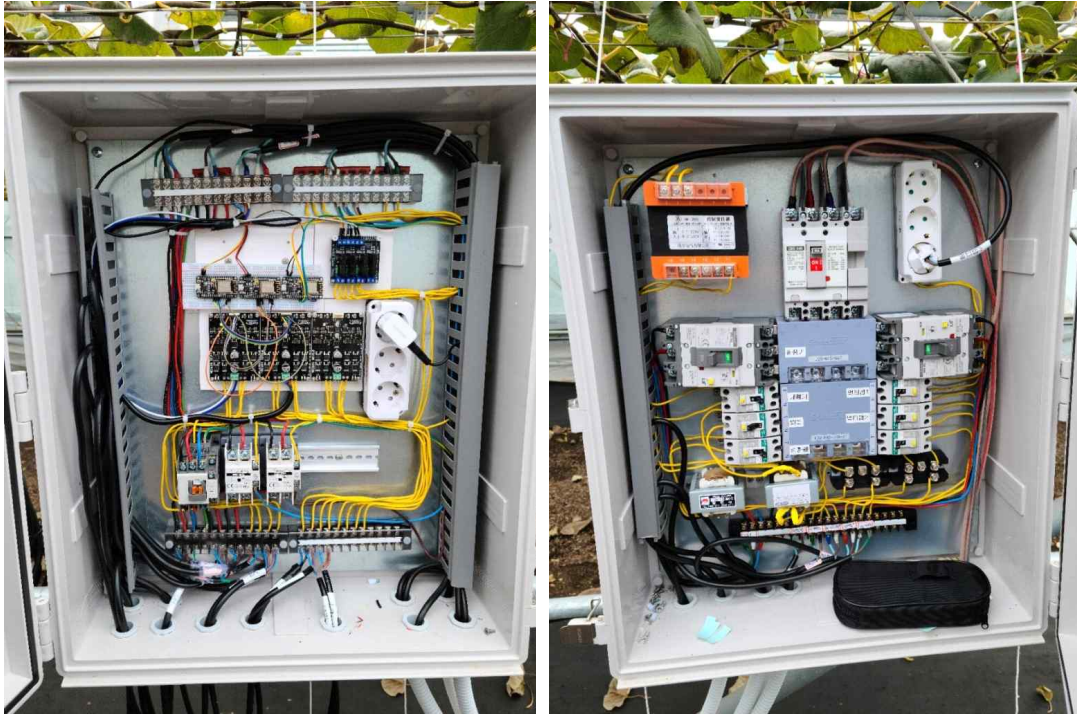


Fig 12. Hardware production

Ⅲ. 연구 내용

1. 시스템 구성 및 동작 원리

시스템 구성은 크게 웹서버, 데이터베이스, WiFi통신, 클라이언트로 구분된다. 웹서버는 HTML과 Apache 서버프로그램 등을 이용하여 자체 서버를 구축하였다. 그리고 전용 데이터베이스는 MariaDB를 이용하여 구축하였고 HeidiSQL을 이용하여 데이터베이스를 관리하였다. 웹서버와 데이터베이스, 제어보드 사이에 데이터 전송은 PHP를 이용하였다. Fig. 31과 같이 시스템 구성은 농가 스마트폰으로 장치를 원격제어를 한다면 인터넷을 통해 웹서버에 데이터가 전송되고 웹서버와 데이터베이스 간에 데이터가 HTTP통신으로 전달되어 WiFi통신으로 제어보드에 작동데이터가 전송된다.

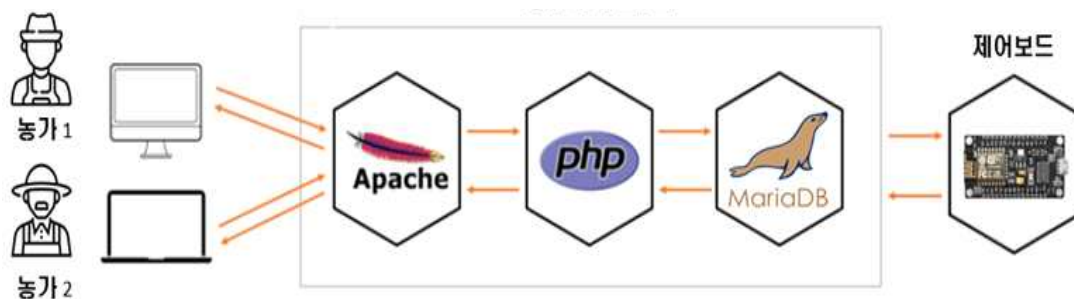


Fig 13. System configuration diagram

Fig. 14와 같이 클라이언트에서 구동기까지 시스템 내 데이터 전달과정을 살펴보면, ① 사용자(클라이언트)가 애플리케이션을 통해 전용 웹서버에 HTTP 통신을 이용하여 데이터를 보낸다. ② 웹서버에서는 PHP엔진을 통해 스크립트 실행요청을 하고 파싱한 스크립트는 웹서버와 데이터베이스의 연결을 수행한다. ③ PHP는 데이터베이스와 연결된 후 해당 SQL명령을 전달한다. ④ 데이터베이스는 SQL명령을 받아 데이터를 처리 및 저장하고 그 결과를 PHP엔진

에 되돌려주어 웹서버와 제어보드의 WiFi통신모듈로 데이터를 전송한다. ⑤ WiFi 통신 모듈로부터 전송받은 데이터로 제어보드를 통해 릴레이모듈과 모터 드라이버 등 구동 제어 장치를 이용하여 개폐기, 환풍기 등 농업용 장치를 제어한다. ⑥ 웹서버는 PHP엔진을 통해 받은 데이터를 웹서버로 HTML형식에 맞게 반환하여 사용자에서 현재 상태 등의 정보를 전달한다.

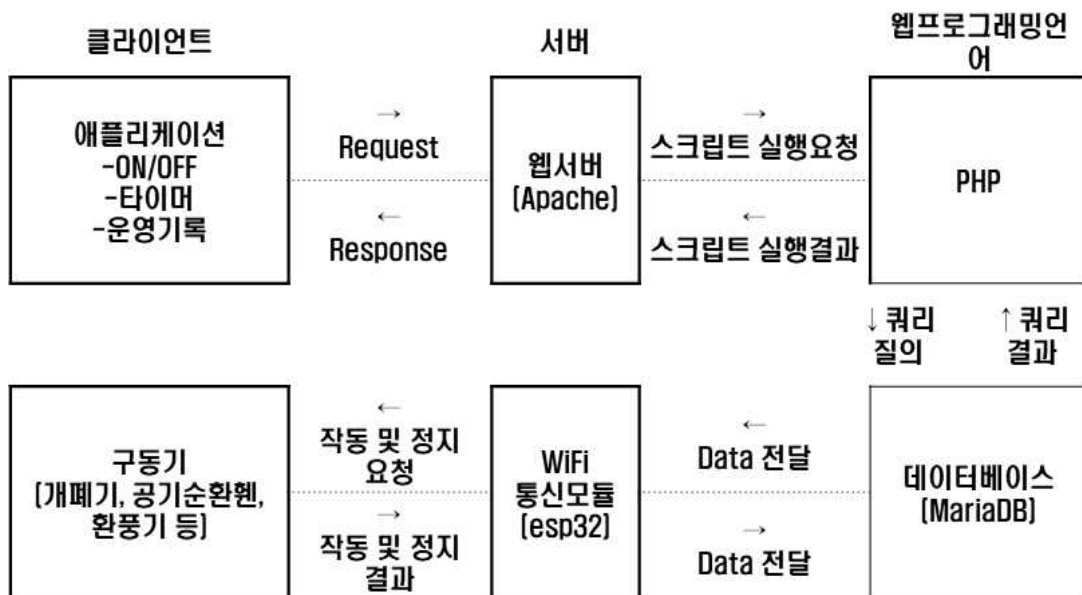


Fig 14. System data delivery process

Fig. 15과 같이 본 시스템의 작동원리 한 예시로 개폐기 작동 시 데이터 흐름을 살펴보면, 사용자가 원격으로 하우스 천·측창 개폐기를 열고 싶은 상황이다.

만약 사용자가 ‘개폐기 열림’ 버튼을 누른다면 웹서버에서 데이터베이스로 ‘개폐기 열림’ 데이터가 전송되어 데이터베이스의 개폐기 상태 데이터가 ‘개폐기 열림’ 데이터로 업데이트된다. 이때 제어보드는 데이터베이스의 데이터를 계속해서 주시하다가 데이터가 ‘개폐기 열림’으로 업데이트되면 곧바로 개폐기를 작동시키고 작동 결과를 다시 데이터베이스의 히스토리 테이블로 반환한다. 그러면 데이터베이스에는 작동 결과 데이터가 저장되고 이 데이터는 웹서버를 통해 사용자가 작동 결과를 확인할 수 있다.

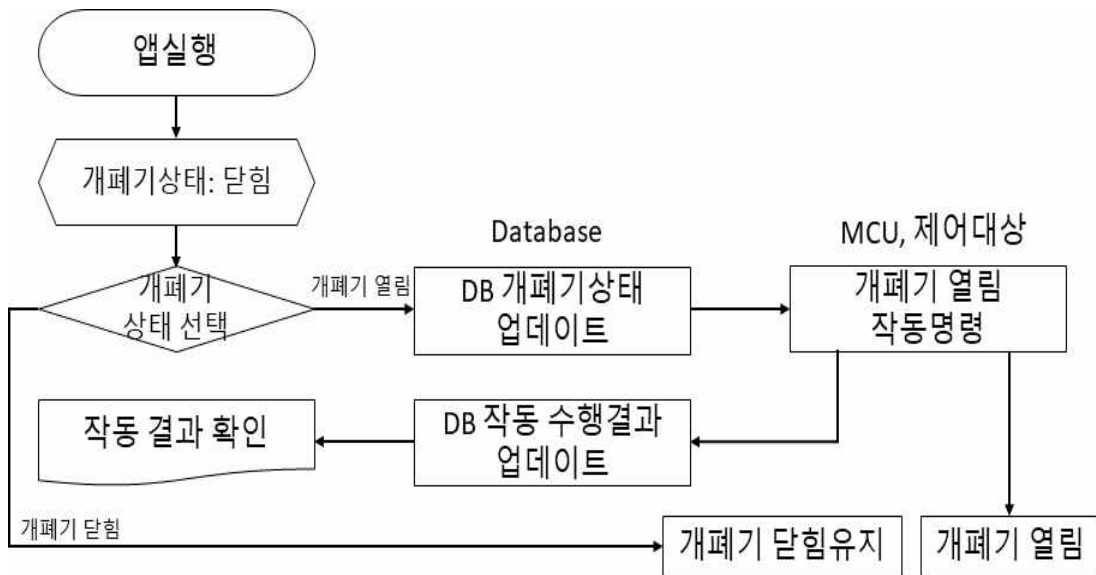


Fig 15. Flow chart of data when operating the switch (Example)

2. 웹서버 및 데이터베이스 구축

본 연구에서 인터넷을 사용하여 웹을 통해 데이터를 송수신하려면 웹서버를 먼저 구축해야한다. 웹서버 구축을 위해 많은 개발자들이 보통 오픈소스 소프트웨어인 Apache, PHP, MySQL 또는 MariaDB 조합을 많이 사용한다. 본 시스템의 웹서버는 Apache를 이용하여 웹서버를 구축하고 PHP모듈을 붙임으로써 웹 어플리케이션 서버(WAS: Web Application Server)기능을 추가하였다. 그리고 MariaDB를 사용함으로써 일반적인 웹서비스 구조를 가진다.

이러한 웹서버 구축을 위하여 주로 사용되는 Apache, PHP, MariaDB의 앞글자를 따서 APM이라고한다. Fig. 16과 같이 웹서버는 클라이언트가 웹서버에서 요청받은 파일을 필요로 할 때, 웹브라우저는 HTTP 통신을 통해 파일을 요청한다. 요청이 올바른 웹서버에 수신되었을 때, 웹서버는 요청받은 문서를 HTTP 통신을 이용해 보내준다.

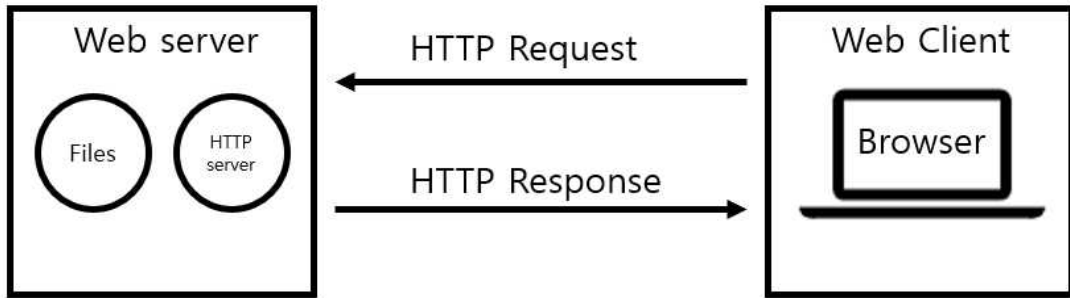


Fig 16. HTTP communication between web server and web client

클라이언트에 있어서 가장 중요한 요소 중 하나인 UI(User Interface)설계를 하였다. UI 설계란 인터랙션이 일어나는 인간과 도구 사이의 인터페이스를 디자인하는 작업을 총칭한다[13]. 주로 농업인들은 고령의 비중이 많기 때문에 UI를 최대한 간편하고 단순하게 설계하였다. Fig. 17과 같이 사용자가 제어할 장치(밸브, 천·측창, 환풍기 등)를 선택한 후 제어 버튼을 누르면 상태창이 업데이트되면서 해당 장치가 작동하게 하였다.



Fig 17. Smartphone control screen


```

<div class="container-fluid">
  <div class="col-x1-12 col-md-6 mb-4">
    <div class="row justify-content-center">
      <div class="card shadow text-center">
        <div class="card-header text-white bg-dark">
          <h5 class="card-title">오픈스스형 MCU기반 스마트팜</h5>
        </div>
        <div class="card-body text-center">
          <ul class="nav nav-tabs card-header-tabs nav-fill" id="SmartTab" role="tablist">
            <li class="nav-item">
              <a class="nav-link active" id="window-tab" data-toggle="tab" href="#window" role="tab" aria-controls="window" aria-selected="false">개폐</a>
            </li>
            <li class="nav-item">
              <a class="nav-link" id="water-tab" data-toggle="tab" href="#water" role="tab" aria-controls="water" aria-selected="true">관수</a>
            </li>
            <li class="nav-item">
              <a class="nav-link" id="fan-tab" data-toggle="tab" href="#fan" role="tab" aria-controls="fan" aria-selected="false">환풍기</a>
            </li>
            <li class="nav-item">
              <a class="nav-link" id="circular-tab" data-toggle="tab" href="#circular" role="tab" aria-controls="circular" aria-selected="false">순환팬</a>
            </li>
          </ul>
          <div class="tab-content mt-3" id="SmartTabContent">
            <div class="tab-pane fade" id="water" role="tabpanel" aria-labelledby="water-tab">
              <table>
                <tr>
                  <thead class="text-center">
                    <tr>
                      <th>밸브</th>
                    </tr>
                    <tr>
                      <th>제어버튼</th>
                    </tr>
                    <tr>
                      <th>상태</th>
                    </tr>
                  </thead>
                <tr>
                  <td>
                    <button type="button" class="btn btn-info btn-lg" disabled="disabled">밸브 1</button>
                  </td>
                  <td>
                    <div class="btn-group ml-2 mr-2" role="group" aria-label="Second group">
                      <button type="button" class="btn btn-primary btn-lg" onClick="runFunc('mowi_open')>OPEN</button>
                      <button type="button" class="btn btn-warning btn-lg" onClick="runFunc('mowi_close')>CLOSE</button>
                    </div>
                  </td>
                </tr>
              </table>
            </div>
          </div>
        </div>
      </div>
    </div>
  </div>
</div>

```

Fig 18. User interface (UI) design

시스템 웹서버는 단순히 사용자에게 텍스트 기반의 정보전달 목적인 ‘정적 웹서버’가 아니라 데이터베이스를 기반으로 사용자에게 실시간 정보전달이 목적인 ‘동적 웹서버’이다. 그러므로 MariaDB와 같은 관계형 데이터베이스 관리 시스템이 필요하며 데이터베이스와 웹서버를 연결할 서버사이드 언어인 PHP가 필요하다[14]. 데이터베이스의 구축은 Fig. 19, 20과 같이 장치작동테이블과, 장치히스토리테이블로 나누었는데, 장치작동테이블에서 각 행(id)당 하나의 장치이다. 만약 사용자가 UI를 통해 어떠한 장치를 ‘열림’으로 설정하였다면, 데이터베이스의 장치작동테이블에서 actvalue값은 1로 바뀌고 동시에 status값은 ‘열림’으로 업데이트된다. 반대로, 사용자가 ‘닫힘’으로 설정하였다면 actvalue값은 0으로 바뀌고 동시에 status값은 ‘닫힘’으로 업데이트된다. 그리고 작동결과는 데이터베이스 장치히스토리테이블에서 각 장치의 작동 시간과 열림, 닫힘을 1과 0으로 표현한다.

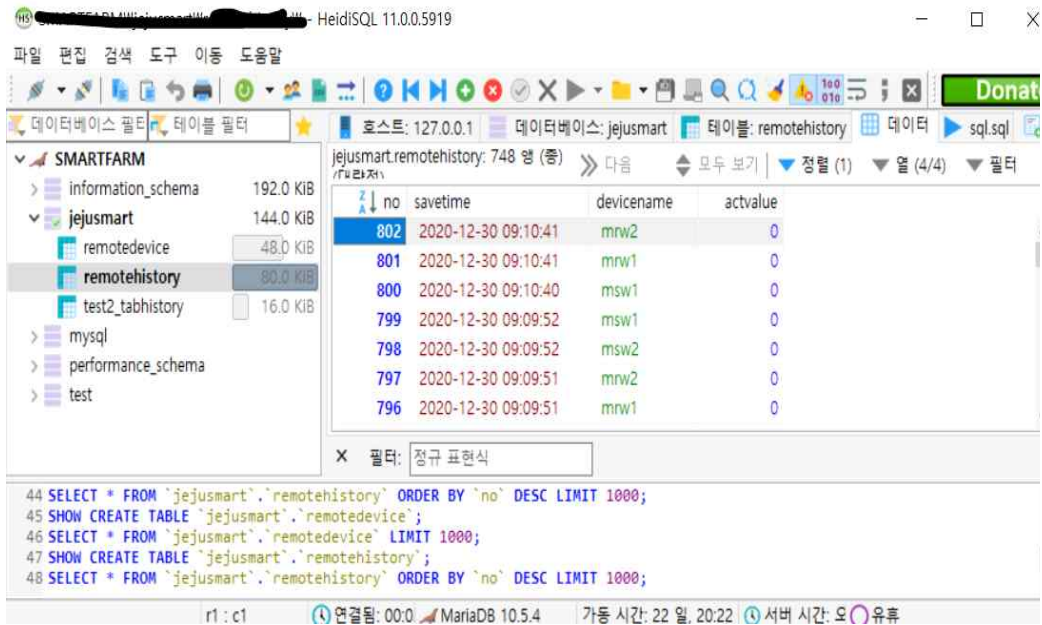


Fig 19. Database management using HeidiSQL(device history table)

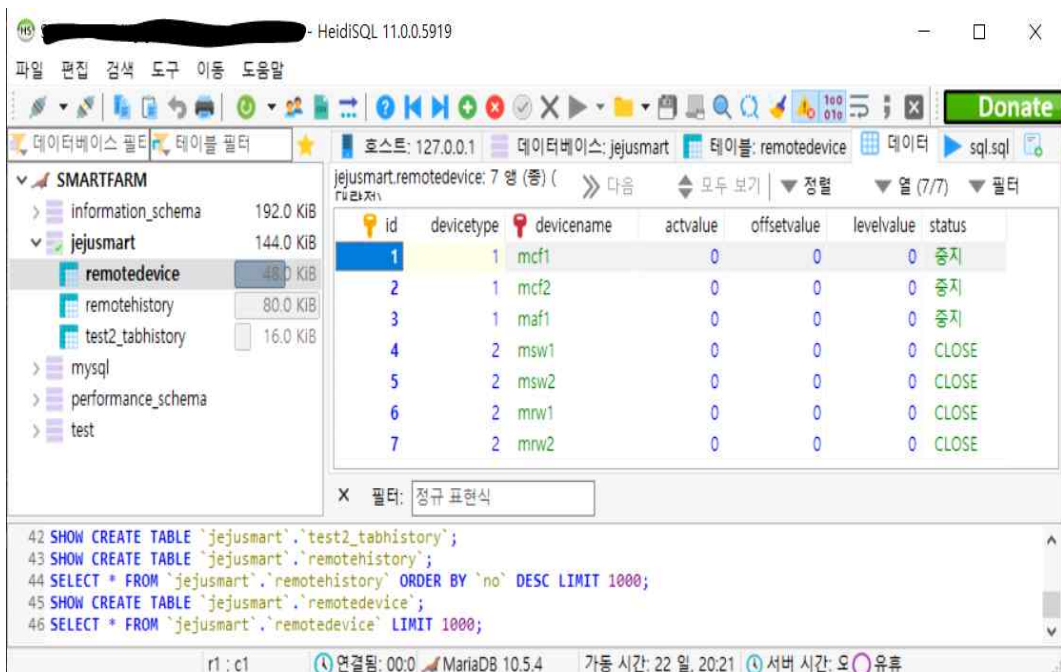


Fig 20. Database management using HeidiSQL (device operation table)

웹서버에서는 데이터베이스의 실시간 데이터를 조회, 생성, 수정, 삭제할 수 있다. 이를 위해서 PHP와 jQuery의 Ajax라는 서버사이드 웹프로그래밍 언어가 필요하다. 여기서 jQuery는 HTML의 클라이언트사이드 조작을 단순화하도록 설계된 크로스 플랫폼의 자바스크립트 라이브러리이며 Ajax는 페이지의 이동 없이 실시간으로 데이터베이스나 다른 웹페이지의 내용을 가져올 수 있는 jQuery의 메소드이다[15]. Fig. 21는 웹서버 index.html 파일에서 Ajax를 이용하여 장치작동.php파일에 변수를 보내는 코드이다. 여기서 HTTP 통신으로 POST 요청을 수행하여 성공여부에 따라서 success 속성의 함수가 호출된다.

```
<script src="resources/js/jquery-3.5.1.min.js"></script>
<script>
function runFunc(value){
    var data = {};
    data.function_type = value;

    $.ajax({
        url : '/장치작동.php',
        type : 'post',
        data : data,
        success : function(data) {

        },
        error : function(request, status, error) {

        }
    })
};
</script>
```

Fig 21. Code that sends variables to PHP file using Ajax in index.html file

Fig. 22는 장치작동.php 파일에서 데이터베이스에 접속 후 해당 장치의 작동 여부와 히스토리 데이터를 수정 및 생성하는 코드이다. 만약 사용자가 UI에서 ‘개폐기 열림’ 버튼을 눌렀을 때 웹서버 index.html 파일에서 제어데이터가 장치작동.php 파일에 ‘개폐기 열림’ 변수가 전송된다. 그러면 SQL 명령문에 따라서 장치작동테이블의 actvalue값은 1, status는 열림으로 수정된다. 그리고 장

장치히스토리테이블의 작동 여부 데이터가 실시간으로 생성된다. 사용자가 '개폐기 닫힘' 버튼을 누르면 정반대로 데이터가 수정 및 생성된다.

```
<?php
$servername = "서버URL";
$username = "유저ID";
$password = "비밀번호";
$db_name = "데이터베이스명";

//create connection
$mysqli = new mysqli($servername, $username, $password,
                    $db_name);

// check connection
if ($mysqli->connect_errno) {
    printf("Connect failed: %s\n", $mysqli->connect_error());
;
    exit();
}

$funcType = $_POST['function_type'];
if($funcType === '개폐기열림')
{
    $allowed = "update 장치작동테이블 SET actvalue = 1,
                status='열림' WHERE id = 1;";
    $allowed_res = $mysqli->query($allowed);
    $allowed_ins = "insert into 장치히스토리테이블
                    (devicename, actvalue)
                    values ('개폐기', 1) ORDER BY no DESC LIMIT 1;";
    $allowed_rins = $mysqli->query($allowed_ins);
    $mysqli->close();
}
if($funcType === '개폐기닫힘')
{
    $allowed = "update 장치작동테이블 SET actvalue = 0,
                status='닫힘' WHERE id = 1;";
    $allowed_res = $mysqli->query($allowed);
    $allowed_ins = "insert into 장치히스토리테이블
                    (devicename, actvalue)
                    values ('개폐기', 0) ORDER BY no DESC LIMIT 1;";
    $allowed_rins = $mysqli->query($allowed_ins);
    $mysqli->close();
}
}
```

Fig 22. Code to modify and create database data in device operation.php file

3. 웹서버와 ESP32 통신

ESP32가 인터넷에 연결되기 위해서 통합 개발환경인 Arduino IDE에는 WiFi 라이브러리를 제공하고 있으며, 인터넷의 기본이 되는 TCP/IP 프로토콜과 서비스 제공을 위한 다양한 프로토콜이 존재하므로 간단히 인터넷에 연결할 수 있다[17]. 우선 와이파이 접속을 위하여 Fig. 23과 같이 ESP32에 내장된 ESP32 WiFi모듈이 AP(Access Point)에 접속해야한다.

ESP32가 데이터베이스 장치작동테이블의 데이터를 조회 및 수정하기 위해서 PHP를 이용한다. 그래서 PHP파일을 장치작동테이블의 데이터 조회용과 수정용으로 나누었다. Fig. 24은 장치작동테이블의 데이터를 조회하는 PHP파일이다. 이 파일은 데이터베이스에 접속 후 장치작동테이블의 장치명, 작동 여부, 작동상태 등 모든 데이터를 배열형식으로 변환하여 변수를 정의한다. 그리고 배열형식의 변수를 json형식으로 변환하여 ESP32에 데이터를 보내준다.

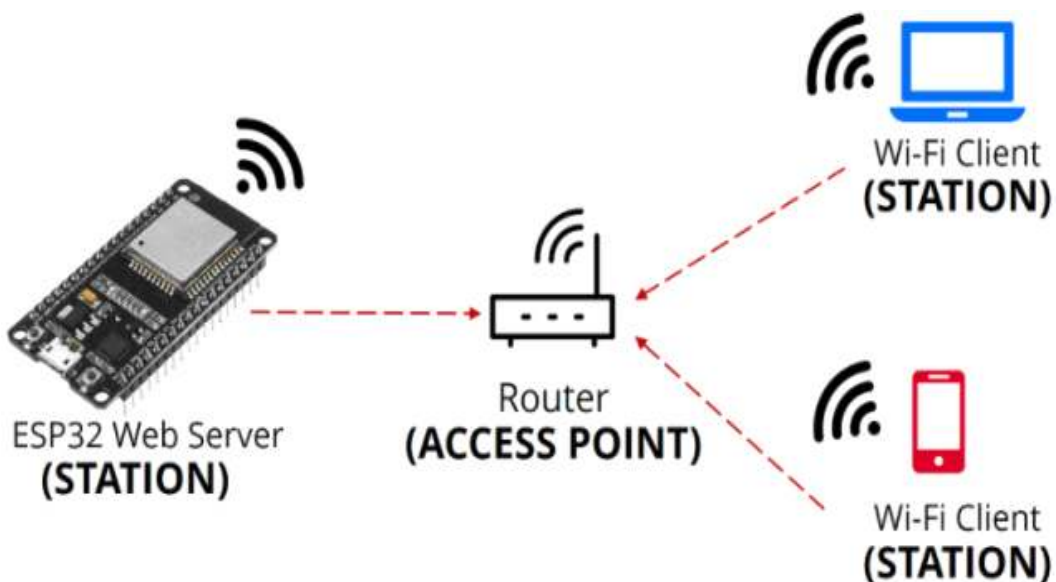


Fig 23. AP connection between ESP32 and Client

```

<?php
$servername = "서버URL";
$username = "유저ID";
$password = "비밀번호";
$db_name = "데이터베이스명";

//create connection
$con = new mysqli($servername, $username, $password,
                 $db_name);
// check connection
if (mysqli_connect_errno())
{
    printf("Connect failed: %s\n", mysqli_connect_error());
    exit();
}

$res = mysqli_query($con, "select * from 장치작동테이블");
// 결과를 배열로 변환하기 위한 변수 정의
$result = array();
// 쿼리문의 결과(res)를 배열형식으로 변환(result)
while($row = mysqli_fetch_array($res))
{
    array_push($result, array('id'=>$row[0],
                              'devicetype'=>$row[1], 'devicename'=>$row[2], 'actvalue'
                              =>$row[4], 'offsetvalue'=>$row[5], 'levelvalue'=>$row[6]
                              , 'status'=>$row[7]));
}
// 배열형식의 결과를 json으로 변환
echo json_encode($result, JSON_UNESCAPED_UNICODE);
// DB 접속 종료
mysqli_close($con);
?>

```

Fig 24. PHP file that retrieves the data of the device operation table

Fig. 25는 ESP32에서 장치작동테이블의 장치 작동 데이터를 조회하여 개폐기를 작동시키는 코드이다. ESP32는 데이터베이스의 데이터를 수시로 조회하여 값이 업데이트되면 개폐기 등 농업용 장치를 작동시킨다. 여기서는 actvalue값이 1일 때 열리고, 2일 때 닫히는 조건이다. 하지만 만약 actvalue값이 1이나 2일 경우 개폐기 장치의 리미트 조정에 따라서 특정 시점이 되면 자동으로 정지가 되지만 실제로 제어보드 내부에서는 계속해서 장치 작동을 지시하며 제어

보드에 부담을 준다. 결론적으로 제어보드에 부담을 덜기 위해 개폐기가 특정 시간동안 작동되었을 때 작동을 멈추고 장치 작동이 중지되었다는 뜻으로 actvalue에 0을 데이터 베이스에 반환해주는 것이 좋다.

```

nodeMCU_msw | 아두이노 1.8.13
파일 편집 스케치 툴 도움말
nodeMCU_msw
String deviceOperation(int No, int actValue, int deviceType)
{
    if(actValue==1)
    {
        digitalWrite(DevicePinNo[No-3][0], HIGH);
        digitalWrite(DevicePinNo[No-3][1], HIGH);
        digitalWrite(LED_BUILTIN, LOW);
        if (deviceType==2)
        {
            delay(motorTime*1000);
        }
        else
        {
            delay(valveTime*1000);
        }
        digitalWrite(DevicePinNo[No-3][0], LOW);
        digitalWrite(DevicePinNo[No-3][1], LOW);
        digitalWrite(LED_BUILTIN, HIGH);
        String d_status="OPEN";
        updateDeviceStatus(No, d_status);
    }
    else if(actValue==2)
    {
        digitalWrite(DevicePinNo[No-3][0], HIGH);
        digitalWrite(DevicePinNo[No-3][1], LOW);
        digitalWrite(LED_BUILTIN, LOW);
        if (deviceType==2)
        {
            delay(motorTime*1000);
        }
        else
        {
            delay(valveTime*1000);
        }
        digitalWrite(DevicePinNo[No-3][0], LOW);
        digitalWrite(DevicePinNo[No-3][1], LOW);
        digitalWrite(LED_BUILTIN, HIGH);
        String d_status="CLOSE";
        updateDeviceStatus(No, d_status);
    }
    else
    {
        digitalWrite(DevicePinNo[No-3][0], LOW);
        digitalWrite(DevicePinNo[No-3][1], LOW);
        delay(100);
    }
    return strResult;
}

```

Fig 25. Actuation code of the switch according to the value of the device operation table

Fig. 26은 장치작동테이블의 데이터를 수정하는 PHP파일이다. Fig. 25과 같이 ESP32에서 장치의 작동이 끝난 후 Fig. 26의 PHP파일로 장치가 중지되었다는

뜻인 actvalue에 0을 반환하는 코드이다. 그러면 PHP파일을 통해 데이터베이스로 작동이 중지되었다는 데이터가 전송되고 이를 사용자가 정상적인 작동 여부를 조회할 수 있다.

```
<?php
if ($_SERVER['REQUEST_METHOD'] == 'POST')
{
    $data = json_decode(file_get_contents("php://input"),true)
;
    print_r($data);
};
//echo php info
$servername = "서버URL";
$username = "유저ID";
$password = "비밀번호";
$db_name = "데이터베이스명";
$id = $data['id'];
$act_value = $data['actvalue'];
$status = $data['status'];
//create connection
mysqli = new mysqli($servername, $username, $password, $d
b_name);
// check connection
if (mysqli_connect_errno()) {
    printf("Connect failed: %s\n", mysqli_connect_error());
    exit();
}

//작동 데이터를 mysql에 집어 넣는 쿼리
if(isset($id))
{
    $pup_allowed = "update 장치작동테이블 SET actvalue=
'$act_value',status= '$status' WHERE `id` = $id;";
    $result = $mysqli->query($pup_allowed);
}
$mysqli->close();
?>
```

Fig 26. PHP file that modifies the data of the device operation table

4. 시스템 농가실증

본 연구는 제주특별자치도 농업기술원 키위 연구포장에서 시스템 개발을 수행하였고 개발된 시스템의 농가 실증은 Fig. 27과 같이 제주시 구좌읍에 위치한 블루베리 시설하우스에서 수행하였다.

대부분 고령 농업인들은 직접 시스템을 구축하는데 거부감을 느끼는 경우가 많아서 전자기기를 다루는데 상대적으로 능숙한 청년농을 대상으로 농가를 선정하였다.

해당 농가는 제어시스템 도입 초기 비용의 부담을 느껴서 관수, 천·측창 개폐를 수동으로 작동하고 있었다. 오랫동안 각종 농업용 장치를 수동으로 작동시키는데 불편함을 느껴 본 시스템을 현장에 적용하게 되었다. 농가에서 실증한 시스템은 관수용 전자 밸브와 천·측창 개폐기를 실증하였다.

실증농가에 활용한 시스템은 소프트웨어 제어방식으로 실제 하드웨어 구성은 단순하기 때문에 제어보드와 구동 제어 장치 등을 이용하여 설치하였다. 웹서버와 데이터베이스의 경우 농가 자체 서버를 구축하였는데 시스템 코드를 해당 농가 환경에 맞게 일부 수정작업을 거쳐 실증하였다.

농가에 시스템 실증 후 농가 반응은 ‘수동에서 원격제어로 작동방식이 바뀌니 작물 재배 관리에 편리함을 느낀다.’라는 반응과 ‘평소에 물을 줄 때나 비가 올 때 자택에서 일부러 농가까지 이동 후 작업하는데 이제는 그럴 필요가 없어서 좋다.’라는 반응이 있었다.

본 시스템은 원격 수동제어용으로 상용화된 스마트팜과 단순비교가 어려워 해당 시스템은 온·습도 및 강우 감지 센서 등을 추가한 알고리즘을 개발하여 시스템 보완이 필요하다.



Fig 27. Blueberry Farm Empirical Results

IV. 결론 및 향후 연구

본 시스템은 스마트팜 초기 비용 부담을 낮추고 농가 환경에 따라 맞춤형으로 적용이 가능한 오픈소스 기반 MCU를 이용한 시스템이다. 현재 국내 상용화된 대부분의 스마트팜 시스템은 상대적으로 고가의 PLC기반의 제어시스템으로 농가가 스마트팜을 도입하기에는 높은 초기비용이 필요하기 때문에 접근성이 떨어진다. 또한 PLC기반의 제어시스템은 많은 산업 분야에서 활용되고 있지만, 농업 현장에서와 같이 제한적인 환경 데이터 측정과 농업용장치의 제어목적이라면 PLC시스템이 투자비용 대비 비효율적일 수 있다.

과거에는 MCU 기반 제어시스템은 내구성과 신뢰성 등 여러 문제로 농가 현장에서 활용이 어려웠다. 그러나 내구성과 신뢰성 등이 개선된 현재 저비용 MCU보드로 충분히 스마트팜 현장실증이 가능하게 되었다.

제어보드는 크기가 작고 저비용이며 와이파이 통신모듈이 내장되어있는 ESP32를 이용하였고 개폐기, 전자밸브 등과 같은 DC모터를 제어하기 위한 모터드라이버는 MDD10A를 공기순환팬과 환풍기를 제어하기 위해 무접점 릴레이모듈과 마그네트 커넥터를 이용하였다. 웹서버는 HTML과 Apache 서버프로그램을 이용하여 자체 서버를 구축하였고 전용 데이터베이스는 MariaDB를 이용하여 HeidiSQL로 데이터베이스를 관리하였다. 장치 작동 데이터의 흐름은 농가(클라이언트)가 어떤 장치를 제어하면 그 제어데이터가 농가 → 웹서버 → PHP → 데이터베이스 → PHP → 제어보드 → 해당 장치로 전송이 되어 장치가 작동된다. 그리고 장치 작동 완료 데이터는 장치 작동 데이터와 반대로 이루어져 웹서버를 통해 농가가 장치 작동여부를 조회할 수 있다.

본 시스템은 소프트웨어 제어방식으로, 농가에 실증한 결과 실제 하드웨어 구성은 단순하기 때문에 농가의 요구 및 환경에 따라 내부 소스코드만 변경하여 구축을 할 수 있었다. 대부분 고령 농업인들은 직접 시스템을 구축하는데 거부감을 느끼는 경우가 많아서 전자기기를 다루는데 상대적으로 능숙한 청년농을

대상으로 농가를 선정하였다. 농가 실증결과, ‘수동에서 원격제어로 전환되어 농작업이 훨씬 편리하고 인터넷이 가능한 곳이라면 어디든 원격으로 제어가 가능하여 시간을 절약할 수 있다’ 는 반응이었다.

본 시스템은 원격 수동제어용으로 상용화된 스마트팜과 단순비교가 어려워 해당 시스템은 온·습도 및 강우 감지 센서 등을 추가한 알고리즘을 개발하여 시스템 보완이 필요하다.

참고문헌

1. 위키백과사전, 「제 4차 산업혁명」
2. 클라우드 슈밥의 제4차 산업혁명, 2016 다보스리포트, 메가스터디BOOKS
3. Sang-Yeal Han, Jae-Heung Yoo and Hang-Jung Zo, 2015, “Business Strategies for IoT Ecosystem Utilizing Open Source Hardware”. SKtelecom. pp. 12-13
4. 스마트팜코리아, 시설원예분야 구성,
<https://www.smartfarmkorea.net/contents/view.do?menuId=M01010103>
5. 김지은, 이정우, 2019, “스마트팜 기술 및 시장동향 보고서”, 과학기술일자리진흥원, p. 6
6. 시퀀스 릴레이 제어와 PLC제어 장단점, <https://k-issue.com/26>
7. C. Frolund, J. H. Jeon, Y. K. Lee, S. Y. Lee and H. J. Kim, 2014 “Study for strengthening the ICT DIY ecosystem”, information and Communication Technology Convergence(ICTC), pp. 269-274
8. 위키백과사전, 「아두이노」, <https://ko.wikipedia.org/wiki/arduino>
9. 프로세서(processor)란?, <https://soccom.tistory.com/59>
10. 아두이노 우노와 ESP32의 성능비교, MCU 속도 벤치마크, <https://postpop.tistory.com/78>
11. 모터 쉴드란? 모터 드라이버란? 모터 제어방법,정역제어, DC모터,스텝핑 모터, 서보모터, 모터종류, http://makeshare.org/bbs/board.php?bo_table=Parts&wr_id=26
12. 무접점 릴레이 [SSR] 원리, <https://k-whale.tistory.com/40>
13. 이금구, 2003, “UI 디자인에서의 컨셉 단계를 위한 확산적 사고 지향 수법 연구”, 국민대학교
14. 루크 웰링, 2017, 로라 톰슨, “성공적인 웹 프로그래밍 PHP와 MySQL”, 제5판, 정보문화사, pp. 209-210
15. 위키백과사전. 「jQuery」, <https://ko.wikipedia.org/wiki/JQuery>

16. 허경용, 2017, “사물인터넷을 품은 아두이노 사물인터넷에 필요한 연결의 모든 것”, 제이펍, pp. 20-21
17. 김성민, 최환석, 이우섭, 2016, “오픈소스 하드웨어를 활용한 ACOME 기반의 IoT 홈 게이트웨이 환경 개발”, 한국컨텐츠학회논문지, vol. 16, no. 3, pp. 296-304
18. Chang-Gyu Seong, 2017, “A Study on the Smart Home System under Internet of Things Using Open Source”, Korea Maritime and Ocean University, pp. 13-14