Journal of KIIT. Vol. 22, No. 12, pp. 133-143, Dec. 31, 2024. pISSN 1598-8619, eISSN 2093-7571
http://dx.doi.org/10.14801/jkiit.2024.22.12.133

유선 센서 시스템의 IoT 전환을 위한 Zigbee 기반 데이터 통합관제 시스템 개발

김민영*, 한유정**[1], 최민**[2], 류준호***

Development of Zigbee-based Data Integration Control System for IoT Conversion of Wired Sensor System

Minyoung Kim*, Yujung Han**[1], Min Choe**[2], and Junho Ryu***

요 약

본 논문은 기존의 유선 데이터 통신 네트워크(TIA/EIA-485)를 기반으로 구축된 센서 데이터 모니터링 시스템을 무선 데이터 통신 네트워크로 전환하여 모든 센서 데이터를 한 곳에서 실시간으로 모니터링 할 수 있는 IoT 시스템으로 개선하는 연구 결과를 제시한다. 기존 유선 네트워크의 한계로 인해 센서 설치 위치에 제약이 있으며, 추가 설치 비용이 발생하고, 중앙에서 센서 데이터를 통합적으로 수집·관찰하는 데 어려움이 있었다. 이를 해결하기 위해 본 연구에서는 Zigbee를 기반으로 하는 무선 네트워크 전환을 통해 센서로 수집된 데이터를 실시간으로 하나의 데이터베이스에 저장하고, 웹 기반 플랫폼을 통해 사용자가 센서 데이터를 일원화된 방식으로 모니터링할 수 있는 IoT 시스템을 구현하였다. 본 연구는 유선 네트워크의 한계를 극복하고 효율적인 데이터 수집 및 모니터링 환경을 제공함으로써 향후 다양한 IoT 응용 분야에서 활용될 가능성을 제시한다.

Abstract

This paper presents the results of a study aimed at improving an existing sensor data monitoring system, which was originally based on a wired data communication network(TIA/EIA-485), by transitioning to a wireless data communication network. The goal is to enable real-time monitoring of all sensor data in a unified IoT system. The limitations of the existing wired network include constraints on sensor installation locations, additional installation costs, and difficulties in centrally collecting and monitoring sensor data. To address these challenges, this research proposes the implementation of a wireless network based on Zigbee technology, enabling real-time data collection from sensors and centralized storage in a single database. Additionally, a web-based platform allows users to monitor sensor data in a unified manner. This study overcomes the limitations of wired networks and offers an efficient environment for data collection and monitoring, suggesting potential applications in various IoT fields.

Keywords

IoT system, zigbee technology, wireless network, real-time monitoring, sensor data collection, web-based platform

* 동의대학교 ICT융복합연구소 조교수(교신저자)
- ORCID: https://orcid.org/0009-0009-5184-7440
** 동의대학교 ICT융복합연구소 학사 후 과정
- ORCID[1]: https://orcid.org/0009-0004-1009-8037
- ORCID[2]: https://orcid.org/0009-0009-0142-5469
*** 부산대학교 정보융합공학과 석사과정
- ORCID: https://orcid.org/0009-0007-7444-2951

· Received: Sep. 25, 2024, Revised: Oct. 28, 2024, Accepted: Oct. 31, 2024

· Corresponding Author: Minyoung Kim
Research Institute of ICT Fusion and Convergence, Dong-eui Univ.
Republic of Korea
Tel.: +82-51-890-4267, Email: kmyco@deu.ac.kr

Ⅰ. 서 론

최근 IT 기술의 급속한 발전과 함께, IoT(Internet of Things) 기반 시스템이 다양한 데이터를 수집하고 분석하는 데 핵심적인 역할을 하고 있다. 이러한 IoT 시스템은 스마트 디바이스에 내장된 센서를 통해 실시간으로 데이터를 수집하고, 대부분 무선 네트워크를 통해 이를 전송함으로써 데이터의 저장과 분석이 이루어진다. 특히, 웹 기반 플랫폼을 통해 통합적으로 데이터를 모니터링하고 분석된 정보를 다양한 예측자료에 활용하는 것이 가능해졌다. 예를 들어, 화재 감지 시스템의 경우 스마트 센서에서 수집된 데이터를 실시간으로 분석하여 화재 발생을 조기에 탐지하고, 나아가 화재 예측까지 가능하게 하는 IoT 기반 솔루션이 널리 도입되고 있다. 이를 통해 우리의 삶은 화재로부터 더욱 안전하게 보호받을 수 있게 되었다[1][2].

그러나, IoT가 도입되기 이전에는 대부분의 유사 시스템이 TIA/EIA-485(RS-485)와 같은 유선 직렬 통신 방식을 사용하여 데이터를 수집하고 전송했다. 당시 이러한 통신 방식은 비교적 저렴한 비용으로 넓은 범위의 네트워크를 구축할 수 있는 장점이 있었지만, 여러 한계를 가지고 있었다. 첫째, 새로운 장치(노드)를 네트워크에 추가하려면 설치 수량과 위치에 제한이 있었으며, 추가 공사와 그에 따른 비용이 발생하는 문제점이 있었다. 둘째, Bus 형식의 토폴로지로 인해 노드 수가 증가할수록 데이터 전송 속도가 저하되었고, 특히 실시간 데이터를 수집하는 데 어려움이 있었다. 이러한 한계로 인해 데이터 수집 주기가 최대 수초 단위로 설정될 수밖에 없다[3][4].

본 논문은 이러한 기존 시스템의 문제를 해결하고자, 유선 기반의 EIA-485 통신 방식을 무선 네트워크 기반의 IoT 시스템으로 전환하는 연구를 다루고자 한다. 특히, 기존의 유선 네트워크를 Zigbee를 기반으로 한 무선 통신 기술로 대체하여 네트워크의 유연성과 확장성을 향상시키고, 초 단위의 시계열 데이터를 실시간으로 수집하고 모니터링 할 수 있는 시스템을 설계 및 구현한 결과를 제시한다. 이를 통해 기존 시스템의 한계를 극복하고, 데이터 수집 및 분석의 효율성을 대폭 향상시킬 수 있는 방안을 소개하고자 한다.

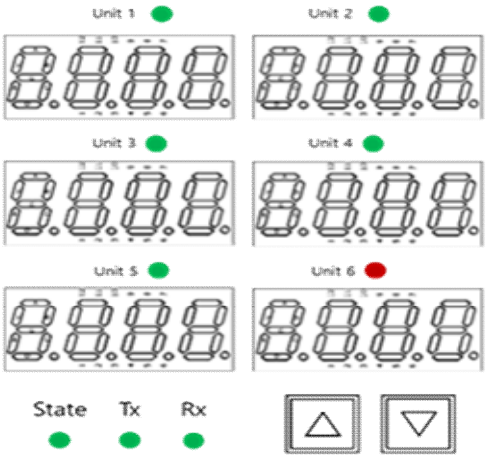
본 논문은 ‘2022년도 한국정보기술학회 추계종합학술대회 및 대학생 논문경진대회’[3]와 ‘2023년도 한국정보기술학회 추계종합학술대회 및 대학생 논문경진대회’[4][5]에서 발표한 연구를 기반으로, 추가적인 연구를 수행하여 이전 연구 내용을 개선하고 기능을 추가해 확장했다.

Ⅱ. 기존 시스템 분석

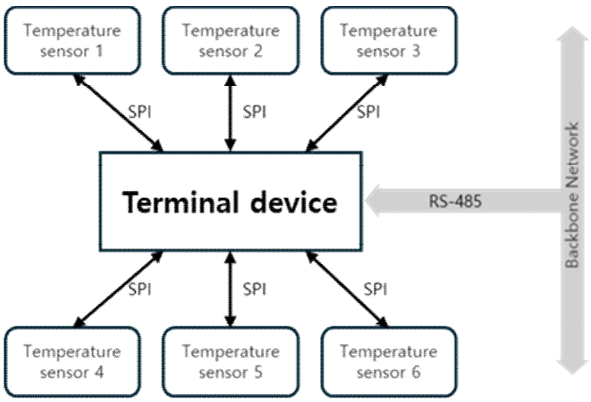
본 논문에서는 건물 또는 시설에 설치된 여러 배전반의 화재 발생 전 상태를 감시하는 기존 시스템의 문제를 개선하기 위한 연구를 다룬다. 해당 시스템은 화재를 사전에 방지해 사용자의 안전과 재산을 보호하고, 배전반의 화재가 전소되기 전에 조처해 시설 재구축에 걸리는 시간과 비용을 절감하는 역할을 한다. 기존 시스템은 배전반 온도를 측정하는 ‘단말장치’, 이러한 데이터를 수집하고 확인할 수 있는 ‘모니터링 장치’로 구성되며, 이를 운영하기 위해 TIA/EIA-485(RS-485) 통신 기반 Bus형 유선 네트워크가 구축된다.(단, 본 연구의 주관 기관의 요청에 따라 본 논문에서는 기존 시스템명과 각 요소의 일부 상세한 사양 등의 내용을 명시하지 않는다.)

2.1 단말장치

본 논문에서 분석한 기존 시스템은 배전반 내부에 설치된 비접촉식 온도 센서를 활용하여, 배전반의 주요 지점의 온도를 실시간으로 측정하고 그림 1의 (a)와 같이 이를 7-segment로 구성된 디스플레이에 출력하는 방식으로 운영된다. 또한, 설정된 온도 범위를 초과할 때 스피커를 이용한 경고 알람 기능을 수행한다. 이 시스템은 시설이나 건물 내 여러 배전반에 설치되어 운영되며, 그림 1의 (b)와 같이 이 단말장치와 온도 센서들은 직렬 통신 방식 중 하나인 SPI(Serial Peripheral Interface Bus)를 통해 연결되어 실시간으로 온도 데이터를 수집한다. 이렇게 수집된 데이터는 모니터링 장치(PC)에서 요청 시 통신 네트워크를 통해 전송된다.



(a) 단말장치 화면 인터페이스
(a) Terminal device screen interface



(b) 단말장치 내부 연결 구성
(b) Terminal device internal connection configuration

그림 1. 단말장치의 화면구성 및 내부 구성도
Fig. 1. Screen interface and internal configuration of the terminal device

2.2 통신 네트워크 및 프로토콜

기존 시스템은 그림 2와 같이 TIA/EIA-485 통신 방식을 기반으로 Bus형 토폴로지(반이중 통신)를 채택하여 네트워크를 구성하고 있다. 이 구조는 전파 노이즈가 많은 산업 환경에서도 안정적인 데이터 통신을 보장하도록 설계되어 개발된 당시에는 통신의 효율성과 안정성을 동시에 고려한 방식이다. 이 네트워크에서는 데이터 송수신은 최소 몇 분 간격으로 이루어진다. 그리고 통신 케이블의 부하와 노드 간 충돌을 줄이기 위해 최대 노드 수와 통신 거리는 네트워크 운영 환경인 공급 전압에 따라 제한된다. 이 통신 환경에서 효율성과 시스템 안정성을 보장하기 위해 표 1에 해당 설정값을 기술하였다[6][7].

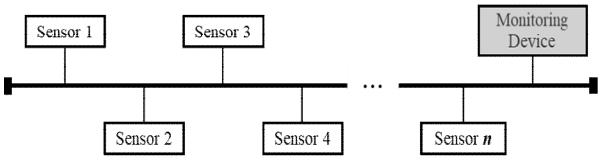
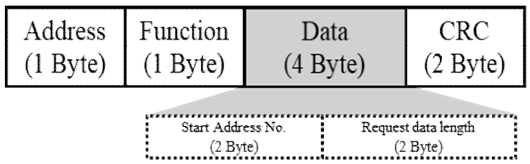


그림 2. 기존 시스템 네트워크 토폴로지 구성
Fig. 2. Network topology diagram of the existing system

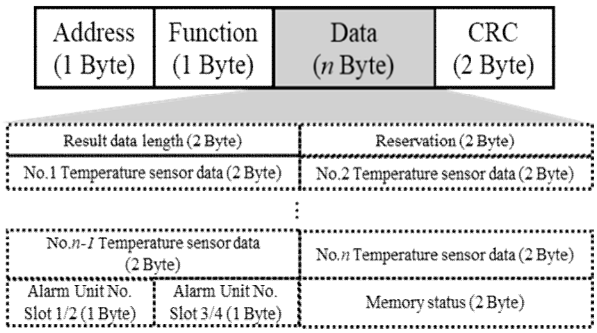
표 1. 기존 네트워크 설정
Table 1. Existing network settings

Para-meters	Baud rate (bit/sec)	Data bit	Parity bit	Stop bit	Flow control
Setting	9,600	8	None	1	None

이 네트워크에 연결된 모든 노드, 즉 단말장치와 모니터링 장치 간의 통신은 Modbus 프로토콜을 기반으로 이루어지며, RTU(Remote Terminal Unit) 방식을 채택하고 있다(그림 3 참조)[8]. 기존 시스템에서는 Modbus 프로토콜의 주요 필드 중 address와 function 부분이 Modbus 표준에 따라 설정되었으며, 데이터 전송 시 data 필드가 모두 삽입된 후 CRC-16 알고리즘을 사용하여 오류 검출을 위한 값을 CRC 필드에 추가한다. Data 필드는 그림 3과 같이 여러 종류의 데이터로 구성되어 있다.



(a) 전체 통신 프로토콜 구성
(a) Overall communication protocol layout



(b) Data 필드 내 프로토콜 구성
(b) Protocol layout within the data field

그림 3. 기존 시스템의 통신 프로토콜
Fig. 3. Communication protocols of the existing systems

그림 3의 (a)는 모니터링 장치가 네트워크에 연결된 각 단말장치로 데이터를 전송할 때 사용하는 프로토콜이다. 모든 단말장치는 이 프로토콜에 따라 요청을 받고, 각 단말장치는 자신이 수집하고 저장 중인 데이터를 그림 3의 (b) 형식에 맞춰 모니터링 장치로 전송한다. (단, 본 논문에서는 기존 시스템에서 사용 중인 프로토콜의 필드별 설정값과 data 필드의 세부 내용은 2장에 처음 단락의 마지막에 언급한 이유로 상세하게 설명하지 않았다.)

2.3 모니터링 장치

기존 시스템에서 모니터링 장비는 전용 모니터링 소프트웨어(이하 'SW')가 설치된 PC를 의미한다. 이 SW는 최소 몇 분마다 네트워크에 연결된 모든 단말장치에 2.2절에서 설명한 프로토콜(그림 3(a))에 맞춘 요청데이터를 전송하고, 이를 통해 수집된 데이터를 모니터에 표시한다. 이때 화면구성은 그림 1의 (a)와 유사하며, 여러 단말장치가 있을 때 모니터에 각 단말장치의 데이터를 보여주는 여러 개의 화면이 동시에 나타나는 인터페이스로 구성된다.

이 SW는 수집된 데이터를 경량 관계형 데이터베이스에 저장하며, 추후 특정한 파일로 데이터를 내보내는 기능을 제공한다. 이를 통해 사용자는 수집된 데이터를 손쉽게 활용하여 보고서 작성이나 분석에 사용할 수 있다.

2.4 기존 시스템의 한계

기존 시스템에서는 1장에서 논의한 바와 같이 새로운 노드(단말장치)를 네트워크에 추가하는 것이 어렵기 때문에 시스템 확장성이 제한된다. 이로 인한 데이터 수집에도 어려움이 발생할 수 있다. TIA/EIA-485 기반 네트워크는 물리적인 통신 거리의 한계로 인해 넓은 지역에서는 별도의 네트워크 구성이 필요하다. 각 네트워크에는 별도의 모니터링 장치가 설치되며, 사용자는 이 장치들에 저장된 데이터를 개별적으로 수집하고 통합하여 분석해야 한다. 이러한 방식으로 운영되는 기존 시스템에서는 데이터 취합과 분석의 복잡성으로 인해 사용자가 수동으로 데이터를 정리하고 보고서를 작성해야 하는 불편함이 있다[4][5].

III. 새로운 시스템 설계

본 논문에서 제안하는 새로운 시스템은 2.4절에서 언급된 기존 시스템의 한계를 극복하기 위해 설계되었으며, 전체적인 구조는 그림 4에 제시되어 있다. 기존 시스템의 단말장치와 통신 프로토콜을 제외한 모든 요소는 사물인터넷(IoT) 기술을 기반으로 새롭게 설계되었다. 새로운 시스템에서는 Zigbee 기반의 무선 네트워크가 사용되며, 이 네트워크에 연결된 모든 단말장치로부터 GW(Gateway)가 실시간으로 데이터를 수집한다. 수집된 데이터는 DCS(Data Collector Server)로 전송되며, DCS는 각 단말장치에서 받은 데이터를 분류한 후 데이터베이스(DB, Database)에 저장한다. 사용자는 이 DB에 저장된 데이터를 웹사이트(Web)를 통해 실시간으로 모니터링 할 수 있도록 시스템이 설계되었다.

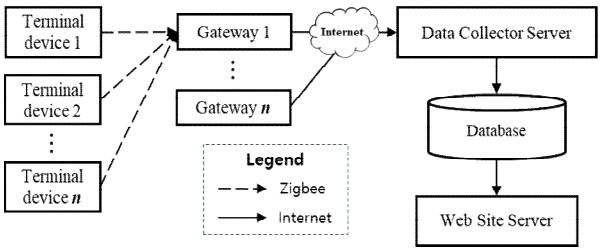


그림 4. 새로운 시스템 구성도
Fig. 4. Schematic diagram of the new system

3.1 통신 네트워크

새로운 시스템에서는 2.2절에서 설명한 기존 시스템 네트워크의 여러 한계를 극복하기 위해 Zigbee 기반의 IoT 무선 네트워크를 도입하였다. Zigbee는 물리 계층과 매체 접근 제어(MAC) 계층에서 IEEE 802.15.4 표준을 사용하며, 같은 표준을 사용하는 다른 통신망에 비해 신호 감쇠가 적고, 통신 범위가 넓다는 장점을 가지고 있다. 또한, Mesh 네트워크 구성할 수 있어 노드 간 통신을 통해 네트워크의 유효 범위를 확장할 수 있으며, 하나의 노드가 고장나더라도 네트워크는 계속 정상적으로 운영될 수 있어, 새로운 시스템에 적합한 네트워크 기술이다[9].

이 시스템에서 단말장치가 Zigbee 네트워크에 연결되기 위해서는 그림 5에 나타난 통신 모듈이 추

가된다. 이 모듈은 기존 단말장치의 TIA/EIA-485 통신 I/O 포트를 그대로 사용하면서, Zigbee 통신을 가능하게 한다. 구체적으로, Zigbee 통신 모듈은 UART(범용 비동기 송수신기)를 사용하여 다른 장치와 통신하며, TIA/EIA-485 신호를 UART로 변환하는 컨버터를 통해 Zigbee 네트워크와 연결된다.

새로운 시스템에서는 단말장치와 GW가 Zigbee 네트워크로 연결된다. 이때 단말장치는 End-device로, 게이트웨이는 Coordinator로 설정되며, 네트워크 범위를 확장할 필요가 있을 때 단말장치는 Router 역할도 할 수 있도록 설정할 수 있다.

그림 5. 단말장치용 통신 모듈
Fig. 5. Communication module for the terminal device

3.2 GW

새로운 시스템에서 게이트웨이(GW)는 단말장치의 데이터를 수집하여 DCS로 전달하는 역할을 담당한다. GW는 Zigbee 네트워크에 연결된 단말장치에 대해, 사용자가 설정한 시간 간격으로 그림 3의 (a)와 같은 프로토콜을 사용하여 데이터를 요청한다. 단말장치는 이 요청을 받은 후, 현재 수집된 데이터를 그림 3의 (b)와 같은 프로토콜을 따라 GW로 전송한다.

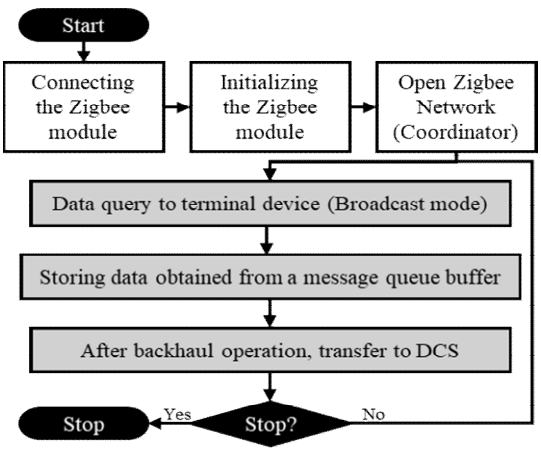


그림 6. GW의 전체 프로세스 흐름도
Fig. 6. GW's overall process flow chart

이 과정에서 GW는 Broadcast 방식으로 단말장치에 데이터를 요청하고, 순차적으로 응답 데이터를 받기 위해 메시지 큐(Message Queue) 방식을 사용한다. 그림 6은 GW의 전체 프로세스를 보여주며, 회색 음영으로 표시된 부분은 멀티스레드 방식으로 실행된다. 또한, 실행 중간에 설정을 변경할 수 있는 프로세스도 포함되어 있다.

단말장치로부터 데이터를 수집한 후, GW는 이를 DCS로 전송하며, 이때 그림 7에서 설명된 전용 프로토콜을 사용한다. 이 프로토콜은 TCP(Transmission Control Protocol)를 기반으로 하며, 그림 7의 데이터 부분에 그림 3(b)에서 정의된 프로토콜 데이터를 포함하여 전송된다.

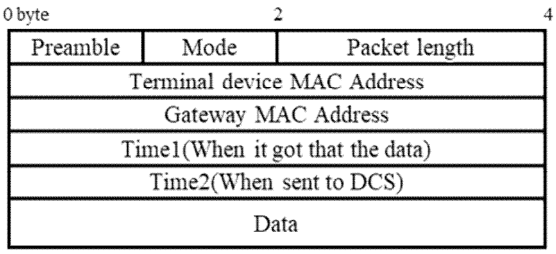


그림 7. GW와 DCS간 전송 프로토콜 구성
Fig. 7. transmission protocol configuration between GW and DCS

마지막으로 GW의 하드웨어는 임베디드 리눅스를 구동할 수 있는 임베디드 장비를 사용한다. 또한, Zigbee 모듈은 UART와 같은 직렬 통신이 가능한 모듈을 채택하였다. 그림 6에서 설명된 절차를 처리하는 소프트웨어는 C언어를 기반으로 설계 및 개발한다.

3.3 DCS

DCS는 새로운 시스템에서 여러 게이트웨이(GW)로부터 수집된 데이터를 데이터베이스에 저장하는 역할을 담당한다. 이때, 지정되지 않은 단말장치나 GW에서 수신된 데이터는 DCS에서 삭제된다. 단말장치와 GW의 지정 여부는 웹사이트를 통해 관리되며, 각 장치는 Zigbee 모듈의 MAC 주소로 구분되고 관리된다. 또한, 여러 GW로부터 동시에 들어오는 패킷을 처리하기 위해 3.2절에서 설명한 메시지 큐(Message queue) 방식을 사용하고, 그림 7의 프로토콜을 기반으로 데이터 파싱을 수행한다.

그림 8은 이 전체 프로세스의 흐름을 나타내며, 회색 음영 부분은 멀티스레드 방식으로 처리된다. 처리해야 할 GW의 수가 많을 경우, 여러 대의 서버를 사용하여 병렬로 운영할 수 있다.

DCS의 하드웨어는 다수의 패킷을 동시에 처리할 수 있는 서버급 PC로 구성된다. 그림 8에 설명된 프로세스를 처리하는 소프트웨어는 C언어를 기반으로 설계되고 개발되었다.

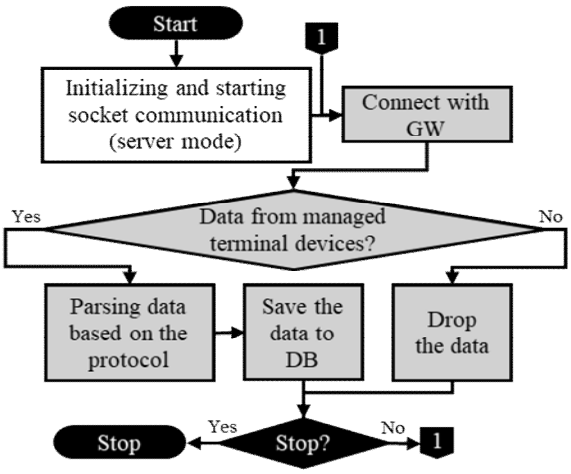


그림 8. DCS의 전체 프로세스 흐름도
Fig. 8. DCS's overall process flow chart

3.4 DB

DB는 DCS에서 파싱 후 저장하는 데이터를 보관하는 역할을 담당한다. 그림 9는 새로운 시스템에서 사용할 DB의 실체 엔티티 형식의 설계도이다. 이 DB는 크게 웹사이트를 사용할 'Web User' 엔티티, DCS에서 온 데이터를 처리할 때 지정할 장치의 정보를 포함한 'Management Device', 그리고 각 단말장치로부터 온 데이터들을 저장할 'Terminal Device' 엔티티로 구성된다.

이때 관리될 단말장치마다 각 Terminal Device 엔티티가 생성되며, 엔티티 이름은 단말장치의 Zigbee 통신모듈의 MAC 주소로 설정된다. 이런 엔티티명으로 설정된 이유는 DCS와 웹사이트에서 DB에서의 조회 및 기록을 쉽게 하기 위해서다. 이 엔티니는 웹사이트에서 관리대상 단말장치 또는 게이트웨이의 정보를 저장하면 Management Device 엔티티에 해당 정보를 저장함과 동시에 해당 테이블도 자동으로 생성되도록 설계된다.

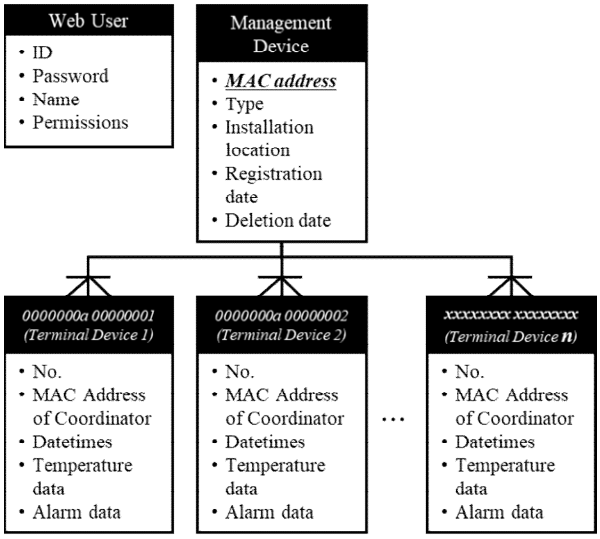


그림 9. 새로운 시스템 DB의 실체 엔티티 설계도
Fig. 9. Entity blueprint of the new system DB

DB의 하드웨어는 DCS와 동일한 이유로 서버급 PC에서 기존 오픈소스 기반의 DBMS(DataBase Management System) 소프트웨어를 이용해 구현 및 운영된다.

3.5 Website

Website는 DB에 저장된 데이터를 실시간으로 모니터링 하기 위한 역할을 담당한다. 이때 DB에 저장되는 데이터를 일정 주기로 조회해 Web에서 모니터링 하기 위해 Websocket 기반으로 기능을 구현한다. 이때 Web과 해당 기능 사이에 데이터는 JSON(JavaScript Object Notation) 프로토콜 기반으로 이루어진다.

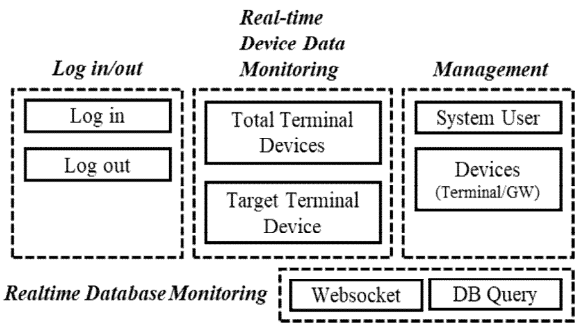


그림 10. WebSite 기능 구성도
Fig. 10. Functional structure of the WebSite

그림 10은 Web의 기능 구성도이다. 크게 로그인/아웃, 실시간 단말장치 데이터 모니터링, 사용자/장치(단말장치 및 GW) 관리, 그리고 실시간 DB 모니터링 기능으로 구분된다. 그리고 그림 11은 Web의 페이지별 구성 및 흐름을 나타낸다. 그림 11의 Log in 페이지와 Terminal Device Real-time Data Monitoring 페이지 이외에는 One 페이지에서 기능을 제공하도록 설계된다.

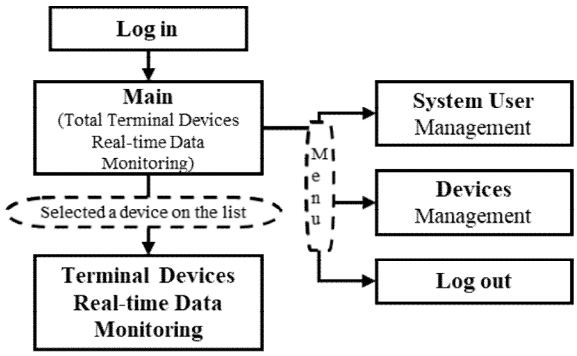


그림 11. WebSite 페이지 구성도
Fig. 11. Page structure of the WebSite

Ⅳ. 새로운 시스템 구현 및 성능 테스트

4.1 구현 결과

새로운 시스템의 하드웨어 구성 요소는 기존 상용 제품을 활용하여 프로토타입으로 제작되었으며, 응용 소프트웨어는 직접 개발되었다.

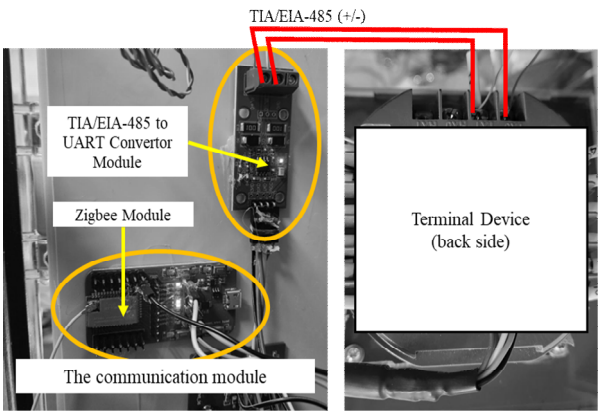


그림 12. 게이트웨이용 Zigbee 통신모듈 프로토타입 구현 결과
Fig. 12. Implementation result of Zigbee communication module prototype for the terminal device

그림 12는 3.1절에서 설명한 단말장치용 Zigbee 통신모듈의 프로토타입 구현을 보여주며, PCB 제작 전 그림 5와 같은 구조로 구현되었다. 이때 Zigbee 통신모듈과 단말장치는 표 1에 제시된 통신 환경으로 설정되어 작동된다[4].

그림 13은 3.2절에서 설명한 GW 하드웨어의 구현 결과물이다. 하드웨어는 Raspberry Pi 3 Model B를 사용하였고, Zigbee 모듈은 USB로 연결 가능한 제품을 선택하여 Raspberry Pi의 USB 포트에 연결하였다. 인터넷 연결은 Wi-Fi를 통해 이루어졌다[4].

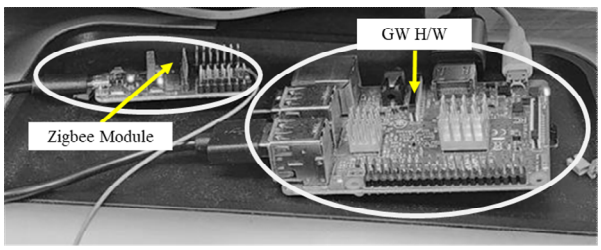


그림 13. 게이트웨이용 하드웨어 프로토타입 구현 결과
Fig. 13. Results of hardware prototype implementation for gateway

그림 14는 3.2절에 언급된 프로세스를 실행하는 소프트웨어의 구현 장면이다. 이 소프트웨어는 Zigbee 네트워크에 연결된 단말장치로부터 실시간 데이터를 수집하고, 이를 그림 7의 프로토콜로 변환하여 DCS로 전송하는 작업을 동시에 수행한다. 해당 소프트웨어는 백그라운드에서 실행되며, 그림 14의 모니터링 화면은 이 프로세스를 시각화하는 추가 소프트웨어로 개발되었다. 또한, 환경변수 변경을 위한 별도의 소프트웨어도 추가로 개발되었다.

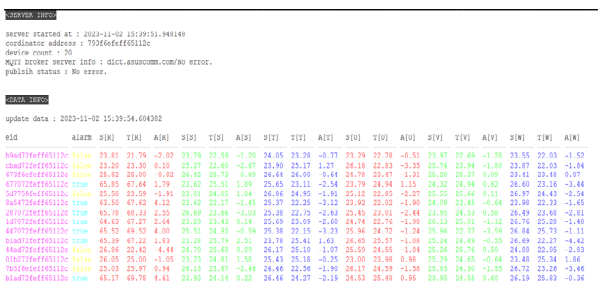


그림 14. GW 소프트웨어 중 '실시간 데이터 모니터링 소프트웨어' 실행화면
Fig. 14. Execution screen of 'real-time data monitoring S/W' of GW software

이 소프트웨어는 Raspberry Pi OS(Linux Kernel 6.1.21-v7+)에서 실행되며, GCC(GNU Compiler Collection) 10.02.1 버전을 사용하여 개발되었다[4].

그림 15는 DCS용 프로그램의 실행 결과 화면을 보여준다. 이 소프트웨어는 GW 소프트웨어와 마찬가지로 백그라운드에서 실행되도록 개발되었다. 그림 16은 리눅스의 systemctl 명령어로 실제 실행 중인 프로세스 정보와, 현재 GW로부터 수집된 데이터를 그림 7의 프로토콜로 파싱하여 로그 형태로 출력하는 화면이다. 이 소프트웨어는 Ubuntu Server 버전 22에서 실행되며, GW 소프트웨어와 동일하게 GCC 11.4.0 버전을 사용하여 개발되었다.

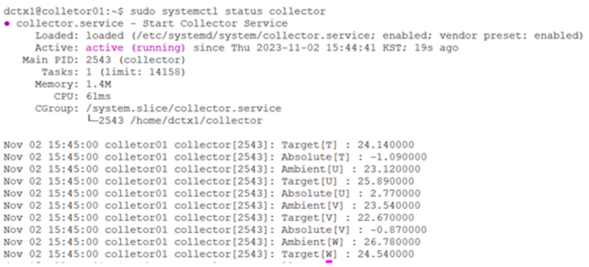


그림 15. DCS 소프트웨어의 실행 확인 결과
Fig. 15. Results confirming execution of DCS software

그림 16은 또한 3.4절에서 설명된 DB가 실제로 DBMS에 구현된 결과를 보여준다. 그림 16은 데이터베이스에 구현된 일부 테이블의 E-R 다이어그램이다. 3.4절의 내용 외에도 로그인 상태를 확인하기 위한 Session 테이블이 추가되었다. 이 데이터베이스는 MariaDB Server 10.6.12 버전을 사용하여 Ubuntu Server 버전 22 환경에서 구축되었다.

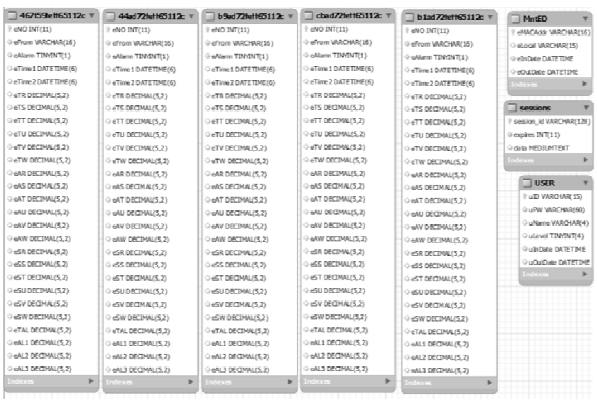


그림 16. 데이터베이스 구현 결과
Fig. 16. Implementation results of the database

그림 17부터 19는 3.5절에서 설명한 Website 구현 결과를 보여준다. 해당 Website에 웹 브라우저로 처음 접속하면 로그인 페이지가 나타나며, 로그인에 성공하면 그림 17과 같이 메인 페이지가 출력된다. 이 페이지에서는 관리되는 단말장치의 데이터를 실시간으로 모니터링할 수 있도록 구현되었다.

그림 17에서 하늘색 배경의 영역은 정상적인 단말장치의 실시간 데이터를 나타내며, 빨간색 배경의 영역은 설정된 임계 온도를 초과한 단말장치의 데이터를 표시한다. 이 배경 색상은 배전반의 상황이 위급함을 직관적으로 알리기 위해 선택되었다. 그림 17의 화면은 일부 단말장치에서 임계 온도에 해당하는 이벤트를 발생시켜 출력된 결과이다.

그림 17. 구현된 Website 중 main 페이지
Fig. 17. Main page of implemented Websites

그림 17에서 하나의 단말장치를 선택하면, 그림 18과 같이 별도의 창이 열리며 해당 단말장치의 실시간 데이터와 시각화된 그래프가 함께 출력된다. 이를 통해 사용자는 데이터를 더 직관적으로 분석할 수 있다.

그림 18. 구현된 Website 중 선택 단말장치 실시간 데이터 모니터링 페이지
Fig. 18. Selected a terminal device real-time data monitoring page from the implemented Website

그림 19는 메인 페이지 우측 상단의 '장치 관리' 버튼을 클릭했을 때 나타나는 화면이다. 이 화면에서는 현재 Website에서 관리할 수 있는 단말장치 목록을 확인할 수 있으며, 사용자가 새로운 단말장치 정보를 추가할 수도 있다. 새로운 단말장치가 추가되면, 해당 장치의 데이터를 저장하기 위한 테이블이 DB에 자동으로 생성된다.

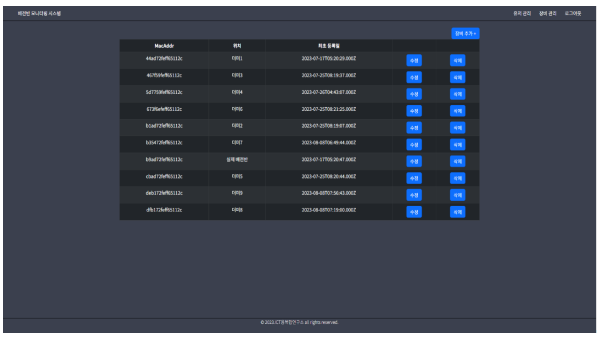


그림 19. 구현된 Website 중 장치 관리 기능 페이지
Fig. 19. Device management function page of implemented Websites

Website 운영을 위해 Ubuntu Server 버전 22가 설치된 환경에서 Node.js 버전 12.22.9와 React 버전 18.2.0을 함께 사용하여 구현되었다. Websocket을 통해 DBMS에 연결하여 일정 주기마다 SQL로 DB 데이터를 실시간 조회하고, 웹사이트에서 필요한 데이터의 읽기 및 쓰기 작업은 Node.js로 구현하였다. 사용자에게 제공되는 화면 구성은 React를 사용하여 개발되었으며, 이 Website는 데스크탑 웹 브라우저뿐만 아니라 모바일 운영체제의 웹 브라우저에서도 실행 가능하도록 구현되었다.

구현된 DCS, DB, Website는 Virtual Box 버전 7을 이용해 하나의 서버급 컴퓨터에서 가상화 환경으로 구축되었다. 네트워크는 Bridge 모드로 설정하여 실제 연결된 인터넷 네트워크에서 IP 주소를 할당받아 인터넷에 연결되었으며, 이를 통해 시스템이 하나의 노드로 인식되어 관리가 용이해졌다.

4.2 성능 테스트

새로운 시스템의 성능을 검증하기 위해 다음과 같은 테스트 환경을 구축하고, 테스트 방법을 계획하여 진행하였다.

- 한 GW에 7대의 단말장치를 연결하고, 20초 간격으로 100번의 데이터 수집을 시도하였다. 이 과정에서 각 프로그램의 Log를 기록하고 처리 시간을 측정하였다. (Zigbee 통신 모듈은 Broadcast 모드로 설정)
- 테스트에서는 1대의 실제 단말장치와 6대의 더미 장치를 사용하였다. 더미 장치는 고정된 온도 데이터를 제공하도록 설정하여 추후 데이터 확인이 용이하도록 하였다.
- 각 요소는 그림 20에 나타난 대로 테스트 장소에 배치되었다.

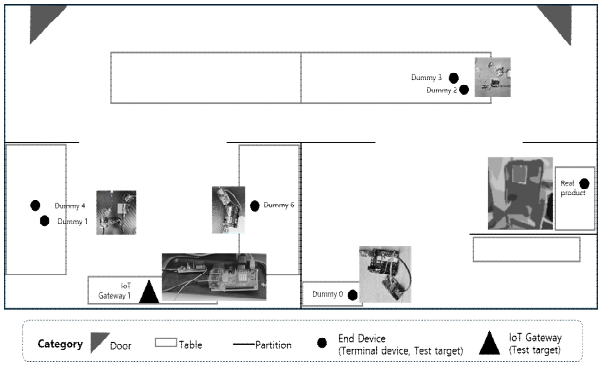


그림 20. 해당 테스트 환경 구축 현황
Fig. 20. Diagram of the test Environment setup state

- 실제 단말장치(그림 20의 'Real product')는 테스트 환경의 온도를 측정하여 저장하며, Zigbee 모듈을 통해 데이터를 저장하였다. TIA/EIA-485 버스 토폴로지를 구성하고, 노트북 PC를 이 버스에 연결하여 전용 프로그램으로 테스트 동안의 데이터를 수집했다.
- GW와 DCS에서는 테스트를 위해 소프트웨어에 Log 기록 기능을 추가했다.
- GW는 조회 요청 시간, 수신된 단말장치, DCS로 전송된 데이터 패킷을 CSV 파일로 기록하였으며, 데이터 수신 시간과 DCS 전송 시간은 그림 7의 프로토콜에 포함되어 있어 별도로 기록하지 않았다.
- DCS는 패킷 수신 시간, 수신된 원본 패킷, DB 저장 시간을 CSV 파일로 기록했다.
- DB에서는 테스트 기간 동안 각 단말장치별로 저장된 데이터를 추출하여 CSV 파일로 저장했다.
- 각 구성 요소에서 기록된 CSV 파일과 DB 기록 내용을 비교하여 데이터 일치 여부를 확인하고 정확도를 산출했다.

표 2. 테스트 결과 (일부 회차의 결과만 추출)
Table 2. Test results (only some rounds of results are extracted)

Round	Start date and time	Transmission/ reception data accuracy	Quantity of end device data received by IoT gateway
1	2024-07-11 23:29:04.305	All match	7
2	2024-07-11 23:29:24.792	All match	7
10	2024-07-11 23:32:08.607	All match	7
11	2024-07-11 23:32:29.097	All match	7
30	2024-07-11 23:38:58.218	All match	6
31	2024-07-11 23:39:18.637	All match	7
60	2024-07-11 23:49:52.987	All match	7
61	2024-07-11 23:50:13.457	All match	7
80	2024-07-11 23:56:42.482	All match	7
81	2024-07-11 23:57:02.970	All match	6
92	2024-07-12 00:00:48.239	All match	7
93	2024-07-12 00:01:29.155	All match	7

위의 조건과 방법대로 테스트 한 결과, 한 번 데이터를 조회하고 수집할 때마다 6~7대 단말장치로부터 응답을 받았다. 총 테스트 시간(2,000초) 동안 단말장치별로 98~99회 응답해 최소 98%의 수신 정확도가 확인되었다. 그리고 수신된 모든 데이터는 오류 없이 정확하게 수신되었다. 표 2는 해당 테스트 결과를 정리한 자료에서 일부만 추출한 내용이다.

V. 결론 및 향후 과제

본 논문에서는 기존 유선 데이터 통신 네트워크(TIA/EIA-485) 기반의 배전반 온도 센서 모니터링 시스템을 무선 데이터 통신 네트워크인 Zigbee로 전환하여, 모든 센서 데이터를 한 곳에서 실시간으로 모니터링할 수 있는 IoT 시스템으로 개선한 연구 결과를 제시했다.

이를 위해 2.4절에서 언급한 기존 시스템의 문제를 해결하고자, GW에서 Zigbee 네트워크에 연결된 단말장치들로부터 일정 주기로 데이터를 요청하고 수집하는 방식이 도입되었다. 수집된 데이터는 DCS를 통해 DB에 저장되며, Website에서 실시간으로 확인할 수 있어 모든 단말장치의 데이터를 통합적으로 관리할 수 있다. 자체 테스트 결과, 20초 간격으로 7대의 단말장치 중 6~7대의 데이터를 안정적으로 수집하고 모니터링할 수 있는 성능이 확인되었다.

본 연구를 통해 단말장치를 설치할 때 장소의 제약을 받지 않으며, 데이터를 통합적으로 수집하여 실시간 모니터링이 가능해졌다. 이는 기존 시스템에 비해 사용 효율성을 크게 향상시켰으며, 유선 네트워크 노드 추가 비용이나 데이터 통합을 위한 추가 작업 비용을 절감할 수 있을 것으로 기대된다.

앞으로는 데이터 수집 주기를 1초로 줄여 모든 단말장치의 데이터를 더 빠르게 수집하는 방법을 연구하고, Website의 기능을 추가 및 고도화하여 사용자에게 더욱 유용한 도구로 발전시키는 연구를 지속할 예정이다.

Acknowledgement

본 논문은 2023년도 한국정보기술학회 추계종합학술대회에서 발표한 논문 “TIA/EIA-485 통신 기반 센서 데이터 수집 시스템에서 Zigbee 기반 IoT 시스템으로 변환하기 위한 연구”[4]를 확장한 것임

References

[1] S. Park, D. Kim, and S. Kim, "Analysis of Unwanted Fire Alarm Signal Pattern of Smoke / Temperature Detector in the IoT-Based Fire Detection System", Journal of the Korean Society of Safety, Vol. 37, No. 2, pp. 69-75, Apr. 2022. https://doi.org/10.14346/JKOSOS.2022.37.2.69.

[2] I. Ehsan, A. Mumtaz, M. I. Khalid, J. Iqbal, S. Hussain, S. S. Ullah, and F. Umar, "Internet of Things-Based Fire Alarm Navigation System: A Fire-Rescue Department Perspective", Mobile Information Systems, Vol. 2022, pp. 1-15, Sep.

2022. https://doi.org/10.1155/2022/3830372.

[3] M. Kim, M. Choe, H. Jeong, J. Ryu, I. Lee, and B. Kim, "Research on the technology required to change the existing sensor monitoring system into an integrated IoT type sensor monitoring system", The Proceedings of the 2022 KIIT Autumn Conference, Jeju, Korea, pp. 96-197, Dec. 2022.

[4] Y. Han, M. Choe, J. Ryu, and M. Kim, "Research on Converting TIA/EIA-485 Communication-BasedSensor Data Collection Systems into Zigbee-Based IoT System", The Proceedings of the 2023 KIIT Autumn Conference, Jeju, Korea, pp. 599-603, Nov. 2023.

[5] M. Choe, Y. Han, J. Ryu, and M. Kim, "Research on How to Monitor for IoT data stored in real-time as relational database on the wesite based on WebSocket", The Proceedings of the 2023 KIIT Autumn Conference, Jeju, Korea, pp. 612-614, Nov. 2023.

[6] TIA, "Electrical Characteristics of Generators and Receivers for Use in Balanced Digital Multipoint Systems (TIA/EIA-485-A Standard)", TIA, pp. 1-16, Mar. 1998.

[7] L. Shi and B. Guo, "RS485/422 Solution in Embedded Access Control System", 2009 2nd International Conference on Biomedical Engineering and Informatics, Tianjin, China, pp. 1-4, Oct. 2009. https://doi.org/10.1109/BMEI.2009.5305573.

[8] Modbus Protocol, https://www.modbus.org/specs.php [accessed: Aug. 02, 2024]

[9] M. Erol-Kantarci and H. T. Mouftah, "Wireless Sensor Networks for Cost-Efficient Residential Energy Management in the Smart Grid", IEEE Transactions on Smart Grid, Vol. 2, No. 2, pp. 314-325, Jun. 2011. https://doi.org/10.1109/TSG.2011.2114678.

저자소개

김 민 영 (Minyoung Kim)

2020년 2월 : 동의대학교 컴퓨터공학과(공학박사)
2019년 3월 ~ 현재 : 동의대학교 ICT융복합연구소 조교수
관심분야 : IoT, 정보시스템, 임베디드 시스템, 양자 컴퓨팅

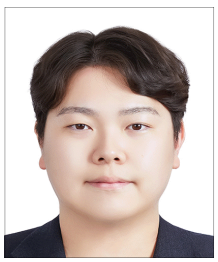
한 유 정 (Yujung Han)

2024년 8월 : 동의대학교 컴퓨터공학과(공학사)
2024년 9월 ~ 현재 : 동의대학교 ICT융복합연구소 학사 후 과정
관심분야 : IoT, 컴퓨터 운영체제, 컴퓨터 네트워크, 드라이버 프로그래밍

최 민 (Min Choe)

2024년 2월 : 동의대학교 컴퓨터공학과(공학사)
2024년 3월 ~ 현재 : 동의대학교 ICT융복합연구소 학사 후 과정
관심분야 : IoT 서비스, 웹 서비스, 빅데이터, 핀테크

류 준 호 (Junho Ryu)



2024년 2월 : 동의대학교 컴퓨터공학과(공학사)
2024년 3월 ~ 8월 : 동의대학교 ICT융복합연구소 학사 후 과정
2024년 9월 ~ 현재 : 부산대학교 정보융합학과 석사과정
관심분야 : IoT, 컴퓨터 비전, 딥러닝