

사물의 이동성 강화를 위한 MQTT 기반의 IoT 사물 제어 기법

장우린*, 정동원**¹, 이석훈**²

A Method of IoT Object Control based on MQTT for Enhancing Object Mobility

Woolin Jang*, Dongwon Jeong**¹, and Sukhoon Lee**²

이 연구는 2019년도 정부(교육부)의 재원으로 한국연구재단의 지원을 받아 수행된 연구임.
(NRF-2019R1I1A3A01060826).

요 약

최근 사물인터넷 기술의 발전으로 새로운 IoT 디바이스들이 개발되고 그 활용 분야가 방대해지면서, 다양한 분야에서 IoT 디바이스를 제어하기 위한 연구가 요구된다. 하지만 기존의 사물 제어 기법은 유연하게 변경되는 네트워크 환경에 대해 확장성 및 이동성 저하, IP 주소인지의 문제 등의 한계를 지닌다. 따라서 본 논문은 IoT 환경에서 사물의 이동성 강화를 위해 디바이스 레지스트리를 활용한 MQTT(Message Queuing Telemetry Transport) 기반의 IoT 사물 제어 기법을 제안한다. 다양한 디바이스의 정보를 동시에 등록 및 관리하기 위해서 디바이스 레지스트리를 사용한다. 이후, 모니터링 툴을 통해 사용자는 디바이스 레지스트리에 등록된 IoT 디바이스의 제어를 요청하며, MQTT를 기반으로 IoT 디바이스로 제어 요청을 전달해 사물 제어 기법을 구현한다. 또한, 본 논문은 시나리오를 통해 제안 시스템이 실생활에서 어떻게 활용될 수 있는지 검증한다.

Abstract

With the recent development of IoT technology, new IoT devices have been developed and their application fields have become vast, and research for controlling IoT devices is required in various fields. However, the existing object control techniques have limitations such as the reduced scalability and mobility, and the recognition problem of IP addresses in flexible network environments. Therefore, this paper proposes an IoT object control technique based on MQTT(Message Queuing Telemetry Transport) using a device registry to enhance the mobility of objects in an IoT environment. A device registry is used to simultaneously register and manage information on various devices. Thereafter, through a monitoring tool, users request control of the IoT device registered in the device registry, and deliver a control request to the IoT device based on MQTT to implement an object control technique. In addition, this paper verifies how the proposed system can be used in real world through scenarios.

Keywords

object control method, MQTT protocol, internet of things, device registry, object mobility

* 군산대학교 소프트웨어학과 학부생
- ORCID: <https://orcid.org/0000-0002-9020-6023>
** 군산대학교 소프트웨어학과 교수(교신저자)
- ORCID¹: <https://orcid.org/0000-0001-9881-5336>
- ORCID²: <https://orcid.org/0000-0002-3390-5602>

• Received: Nov. 11, 2021, Revised: Dec. 06, 2021, Accepted: Dec. 09, 2021
• Corresponding Author: Sukhoon Lee and Dongwon Jeong
Dept. of Software Science & Engineering, Kunsan National University,
Korea
Tel.: +82-63-469-8914, Email: {leha82, djeong}@kunsan.ac.kr

1. 서 론

최근 사물인터넷(IoT, Internet of Things) 관련 기술의 발전으로 새로운 IoT 디바이스들이 개발되면서 연결되는 장치의 수와 그 종류가 증가하고 있다[1]. 이러한 IoT 디바이스들은 센서를 통해 수집되는 다양한 데이터를 분석 및 처리하여 헬스케어, 스마트팜, 일상 생활용품 등 다양한 분야에서 활용되고 있다[2].

IoT 디바이스의 수가 증가함에 따라 디바이스를 효율적으로 관리하고 모니터링 및 제어하기 위한 플랫폼 역시 요구된다[3]. 하지만 이러한 IoT 플랫폼들은 자체적으로 정의된 표준 및 프로토콜을 사용함으로써 그 확장성이 저하되는 문제를 지닌다.

또한, IoT 디바이스는 기술 발전으로 인해 소형화 및 휴대성이 증대되었으며, 이는 이동성이 높아져 다양한 장소에서 사용자에게 서비스를 제공할 수 있도록 발전되었다. 즉, IoT 디바이스들이 점차 작아지고 이동이 쉬워지므로, 한 장소에서 다른 장소로 이동하여 배치되기 쉬우며 이러한 행위는 IoT 플랫폼 내에서 IoT 디바이스의 관리 및 추적 시 그 복잡성이 높아지는 문제를 발생시킨다.

디바이스 레지스트리(Device registry)는 이러한 문제들의 해결을 위하여 본 연구의 선행 연구로서 개발되었다[4]. 디바이스 레지스트리는 IoT 플랫폼 내에서 디바이스의 정보를 등록하고 관리하기 위한 시스템으로 디바이스의 메타데이터를 사용자가 직접 정의하여 키-값의 쌍(Key-value pair) 형태로 등록시켜 유연한 정보 관리를 할 수 있게 한다. 하지만, 기존의 디바이스 레지스트리는 정보 관리 및 센서 데이터의 모니터링에 초점을 두고 있어 사물 제어 기능에 대해서는 고려하지 않고 있다.

현재 다양한 분야에서 IoT 디바이스를 제어하기 위한 시스템에 대한 연구가 활발히 진행되고 있다[5]-[7]. IoT 디바이스의 제어 방식에는 크게 근거리 통신을 사용하거나 원거리 통신으로 인터넷을 통해 제어하는 방식, 추가로 모듈을 연결하는 방식 등이 있다. 예를 들어, 근거리 통신인 NFC 기술을 활용해 차량용 내비게이션의 목적지 입력에 편리성과 안정성 증대를 위한 연구가 있다[8]. 이는 사물인터

넷의 등장으로 새로운 디바이스들이 빠르게 개발되면서, 변화에 적응하기 힘든 사용자들을 위해 개발되었다. 두 번째로 원거리 통신을 이용하여 제어할 때는, 웹 서비스 또는 모바일 애플리케이션을 통해 사물을 모니터링 및 제어한다[9][10]. 마지막으로 인터넷에 연결되지 않은 기기는 별도의 게이트웨이나 허브를 추가로 연결하거나, 사물 자체에 Wi-Fi 모듈을 탑재시킴으로써 사물을 제어할 수 있다[11].

그러나 이러한 사물 제어 기법들은 사물의 이동성 측면에서 보았을 때 유연하게 사물을 이동하기 힘들다는 문제를 지닌다. 이는 사용자가 IoT 디바이스가 설치된 장소 외의 장소에서 IoT 디바이스를 제어하기를 원할 수 있다는 점과 IoT 디바이스들이 다른 네트워크 환경으로 이동 시 IP 주소가 바뀔 수 있다는 점에서 이동성이 저하된다.

예를 들어, 많은 IoT 디바이스들이 집이나 사무실에서 사용이 될 수 있으며, 특히 Wi-Fi를 사용하는 IoT 디바이스의 경우 DHCP(Dynamic Host Configuration Protocol) 기능을 지니는 유무선 공유기를 통하여 인터넷에 접속될 가능성이 높다. 이러한 인터넷 환경은 공유기의 외부에서 IoT 디바이스에 접근하기가 어려우며, 이를 위해서는 포트 포워딩 기능을 공유기에 설정해 놓아야 한다. 또한, 이동이 쉬운 IoT 디바이스는 공유기를 바꾸거나 다른 공유기가 설치된 환경으로 이동될 경우, 공유기 설정을 수정하거나 IoT 디바이스의 플랫폼 설정을 수정해야 하므로 이동성이 저하되는 문제를 지닌다.

이를 위하여, IoT 디바이스의 메시지 프로토콜로 MQTT(Message Queuing Telemetry Transport)와 같은 기술들이 개발되었다. MQTT란 가벼운 발행(Publish) 및 구독(Subscribe) 메시징 프로토콜 형태를 지닌다. 따라서, MQTT는 이동성이 높은 무선 네트워크 환경에서 IoT 디바이스의 메시지 교환을 위해 주로 사용된다[12].

본 논문은 IoT 환경에서 사물의 이동성 강화를 위해 디바이스 레지스트리를 활용한 MQTT 기반의 IoT 사물 제어 기법을 제안한다. 먼저, 다양한 IoT 디바이스 정보를 통합 및 관리하기 위해 디바이스 레지스트리를 활용하고, 디바이스의 센서와 액추에이터에 대한 메타데이터를 정의한다.

제안 기법의 구현을 위하여 환경 데이터를 수집하기 위한 IoT 디바이스를 개발하고, 센서값 및 액추에이터를 제어할 수 있는 웹 기반 모니터링 시스템을 개발한다. 또한, 시나리오를 통하여 제안 기법의 특성을 검증한다.

본 논문의 구성은 다음과 같다. 제2장은 관련 연구를 기술하며, 제3장에서는 제안 시스템과 MQTT 기반 IoT 디바이스 제어 기법에 대해 기술한다. 제4장에서는 구현 결과를 기술하며, 제5장에서는 시나리오를 자세하게 기술한다. 마지막으로, 제6장에서는 결론과 함께 본 연구의 기대효과 및 향후 연구에 대하여 기술한다.

II. 관련 연구

이 장에서는 사물 제어 프로토콜과 사물 제어 시스템에 관한 연구를 소개하고 각 연구의 문제점을 기술한다.

2.1 사물 제어 프로토콜 관련 연구

IoT 환경에서 발생하는 데이터 및 메시지를 전달하기 위한 다양한 프로토콜이 개발되었다[13]-[16].

MQTT는 IoT 디바이스와 서버 간의 통신을 가능하게 하기 위해 IoT 환경에서 사용하도록 고안된 경량의 양방향 통신 프로토콜이다[13]. TCP를 사용하여 패킷 전송의 신뢰성을 보장한다. 지속해서 전송이 이루어져 메시지를 보낼 때 최대한 효율적으로 전송해야 하므로 MQTT와 같은 발행/구독 프로토콜은 IoT의 센서 장치에 특히 중요하다.

CoAP(Constrained Application Protocol)는 HTTP, RESTful과 상호 운용이 가능한 클라이언트/서버 기반의 프로토콜이다[14]. UDP만을 사용하므로 NAT 환경에서 동작하기 위해서는 터널링, 포트 포워딩 같은 방법이 필요하다.

XMPP(eXtensible Message and Presence Protocol)는 XML을 기반으로 IM(Instance Message) 전송을 위한 프로토콜이다[15].

AMQP(Advanced Message Queuing Protocol)는 MQ(Message Queue) 기반의 메시징 전송 프로토콜

이다[16]. 간단한 큐 형태에서부터 발행/구독과 같은 복잡한 구조까지 다양하게 사용할 수 있는 것이 특징이다.

[17]은 IoT 환경에서 사용할 수 있는 통신 프로토콜인 MQTT, CoAP의 장단점을 도출하고 비교·분석한다. 비교 결과, MQTT는 CoAP보다 메시지 전송에서 신뢰성이 높으며 비교적 가벼운 보안 기법을 제공한다는 측면에서 IoT 환경에서 사용하기에 적합하다.

[18]은 XMPP, MQTT 통신 프로토콜을 마이크로 그리드의 내부 혹은 외부적 관점에 적용해 비교하여 적합한 프로토콜을 도출한다. XMPP는 QoS를 따로 지원하지 않고, 스트링 기반의 메시지 통신을 지원하여 오버헤드가 다소 높다. 통신 속도 측면에서도 MQTT가 XMPP보다 약 10ms 정도 빠르다. 따라서 이 연구에서는 장치의 상태 정보를 주기적으로 수신하기 위해서 기능 면에서도 가볍고 통신 성능도 우수한 MQTT가 더 적합하다고 판단하였다.

[19]는 발행/구독 구조의 프로토콜들의 전송 성능을 비교 분석한다. 이 중 MQTT와 AMQP의 비교 결과, 대기 시간이 짧고 비교적 가볍다는 특징을 가진 MQTT가 실시간 모니터링 어플리케이션 환경에서 사용하기에 적합하다고 판단하였다.

이 외에도 Mobile IPv4, Mobile IPv6, NEMO, MIPv6 등의 이동성 관리 프로토콜이 있지만, 전력 및 리소스의 제한 등의 특징으로 인해 IoT 환경에서 사용하기에 적합하지 않다[20].

따라서 네트워크 환경에 유연하고 통신 속도나 전력 소모 면에서 우수하므로 MQTT기반의 사물 제어 기법을 제안한다.

2.2 사물 제어 시스템 관련 연구

IoT 디바이스를 모니터링하거나 제어하기 위한 다양한 연구들이 진행되어 왔다[21]-[24]. 이러한 연구들은 대부분 스마트 홈 환경에서 IoT 디바이스들을 제어하기 위하여 각종 시스템 및 플랫폼을 개발하였다.

[21]은 가정에서 수집한 데이터를 기반으로 MQTT 프로토콜을 사용해 홈 디바이스들을 자동 또는 수

동으로 제어한다. 해당 연구에선 온/습도의 설정값에 따라 DC 모터가 자동으로 동작하며, 조도 값에 따라 LED의 밝기를 조절한다. 하지만 액추에이터의 제어 형태가 모두 on/off의 형태를 이루고, 에어컨 온도 입력과 같이 세부적인 동작의 수행은 불가능하다는 단점이 있다.

[22]는 블루투스를 기반으로 한 스마트 홈 IoT 시스템을 구현한다. 이 연구는 기본적으로 1:1 통신을 지원하는 블루투스를 이용해 하나의 키트에 여러 가지 센서를 부착하여 사용한다. 이로 인해 다수의 디바이스를 모니터링, 원격 제어하기 어렵다는 단점이 있다.

[23]은 IoT 디바이스와 서버 간 통신에 전력선 통신 기반 UDP/IP 통신을 활용한 시스템을 제안한다. 이 연구에서는 UDP의 낮은 신뢰성을 보완하기 위해 오류 검출 기능을 구현하였다. 하지만 오류 검출 기능의 구현을 위해서 서버와 디바이스 간 메시지 전달 시 추가적인 오버헤드가 발생하는 문제를 지닌다.

[24]는 IoT 센서와 라즈베리파이를 이용한 스마트 창문을 개발하고, 스마트폰 애플리케이션을 통해 수동 제어한다. 그러나 사용자가 애플리케이션을 통해 제어하기 위해선 창문과 애플리케이션이 같은 네트워크를 공유하고 있어야 한다는 문제를 지닌다.

이러한 연구들은 대부분 사물 제어를 목표로 하지만, 특정 사물만 제어하거나 네트워크 환경의 변화에 있어 취약하다는 문제가 있다.

따라서 본 논문은 디바이스 레지스트리를 활용하여 상호운용성을 높이고 네트워크 환경의 변화와 관계없이 유연하게 대처하는 것을 목표로 한다.

III. 제안 시스템

이 장에서는 제안 시스템에 대한 구조를 설명한다. 먼저, 제안 시스템의 모듈 중 디바이스 레지스트리의 구조를 기술하고, 센싱, 액추에이팅을 위한 메타데이터의 정의와 MQTT 기반의 메시지 토픽 정의 방법을 기술한다.

3.1 제안 시스템 구조

그림 1은 제안 시스템의 구조이다. 위 그림에서 빨간 선은 센싱 데이터의 흐름을, 파란 선은 제어 요청 데이터의 흐름을 나타낸다. 구조는 크게 IoT 디바이스, 서버(Server), 클라이언트(Client)로 구성되어 있으며, 각 구성 요소와 주요 기능은 다음과 같다.

IoT 디바이스는 센싱과 액추에이팅 부분으로 나누어지며 아두이노에 부착된 센서를 통해 데이터를 수집한다. 이때 수집된 데이터는 블루투스 통신을 통해 MQTT 브로커(Broker)인 라즈베리파이로 전송된다.

서버는 MQTT 브로커, 센서 DB 관리자(Sensor DB manager), 데이터베이스, 모니터링 툴(Monitoring tools), 디바이스 레지스트리로 구성된다.

MQTT 브로커는 IoT 디바이스와 클라이언트들이 보내는 메시지를 수집하고 이를 다시 필요한 장치에 전달해주는 중계 서버 역할을 한다. 제안 시스템은 IoT 디바이스로부터 수집된 데이터를 데이터베이스에 저장하며 사용자로부터의 제어 요청을 IoT 디바이스로 전달한다. 토픽을 사용함으로써 다수의 디바이스로부터 데이터가 들어왔을 때 어떤 디바이스로 데이터를 발행해야 하는지 구분한다.

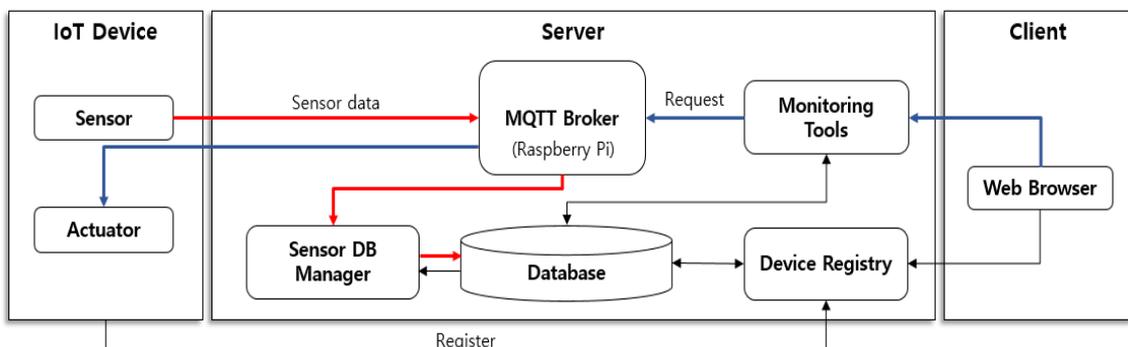


그림 1. 제안 시스템의 구조
Fig. 1. Architecture of proposed system

센서 DB 관리자는 MQTT 브로커로 전송된 센서 데이터를 전송받아, 데이터베이스로 저장하는 역할을 한다. 즉, 센서 DB 관리자는 디바이스 레지스트리에 등록된 각 디바이스에서 발생하는 토픽을 구독하다가, IoT 디바이스가 MQTT 브로커로 센싱된 데이터를 전송하면 MQTT 브로커가 센서 DB 관리자로 해당 데이터를 전송하고, 센서 DB 관리자는 이를 데이터베이스에 저장한다.

모니터링 툴은 IoT 디바이스의 센서 데이터를 보여주고 사물 제어를 요청하는 도구이다. 클라이언트는 IoT 디바이스로 수집된 데이터를 웹 브라우저를 통하여 실시간으로 감시한다. 또한, 모니터링 툴의 사물 제어 기능을 통해 IoT 디바이스의 액추에이터로 제어를 요청할 수 있다. 이때, 제어 요청은 MQTT 브로커에게 전송되고, MQTT 브로커가 이를 IoT 디바이스로 전송함으로써 사물을 제어한다.

3.2 디바이스 레지스트리 데이터 구조

이 절에서는 다양한 종류의 IoT 디바이스의 정보를 등록 및 관리하기 위해 선행 연구인 디바이스 레지스트리를 활용한다[4]. 디바이스 레지스트리는 다양한 스펙의 IoT 디바이스의 메타데이터를 유연하게 등록하고, 통합 관리할 수 있는 시스템이다. 다수의 디바이스를 유연하게 등록하고 관리하기 위해 디바이스의 스펙을 아이템으로 정의하여 등록한다. 이때 디바이스 레지스트리는 아이템의 정보를 키-값의 쌍 형태로 정의한다.

본 연구에서는 디바이스 레지스트리를 통하여 IoT 디바이스의 키-값 정보를 특정 정보(Specific information) 항목에서 정의한다. 표 1은 디바이스 레지스트리의 특정 정보 항목의 데이터 구조를 보인다.

표 1. 특정 정보의 데이터 구조
Table 1. Data structure for specific information

Specific information	Data type	Description
Group	integer	A number that binds the same type of key and value values
Key	string	Classification name of various specifications of the device
Value	string	Value corresponding to key

특정 정보 항목의 구조는 하나의 IoT 디바이스에 대하여 group, key, value로 구성된다. 예를 들어, 온습도 센서(DHT11)가 부착된 IoT 디바이스의 스펙 정보를 등록한다고 가정한다. 이 경우, 표 2와 같이 센서에 대한 메타데이터를 등록할 수 있다.

표 2. 특정 정보에서 정의된 IoT 디바이스의 예
Table 2. Example of IoT device defined in specific information

Group	Key	Value
1	sensor_model	DHT11
1	sensor	Temperature
1	unit_of_measure	C
2	sensor_model	DHT11
2	sensor	Humidity
2	unit_of_measure	%

3.3 센싱, 액추에이팅 메타데이터 정의

디바이스 레지스트리의 메타데이터 정의를 위해 한국정보통신기술협회의 ‘스마트측사 내기 센서 메타데이터 구조’의 표준[25]과 국립전파연구원의 ‘스마트 온실 구동기 메타데이터’의 표준[26]을 참조한다.

디바이스 레지스트리에 다수의 사물을 등록 및 관리하는 과정에서 특정 정보 항목을 작성하고 IoT 디바이스로 수집된 데이터의 저장 및 사물 제어를 위해 이러한 메타데이터 정의 과정이 꼭 필요하다. 따라서 센서 및 액추에이터 등록 정보 메타데이터를 정의한다.

표 3은 본 논문에서 정의한 IoT 디바이스의 센서 메타데이터이다. 키-값 형식으로 IoT 디바이스에 부착된 다양한 센서들의 정보를 표현하며 이는 센서의 모델명, 종류, 최소값, 최대값, 측정 단위, 데이터 타입이 있다. 표 4는 온습도 센서(DHT11), 조도 센서(CDS), 미세먼지 센서(GP2Y10)의 정보를 정의한 메타데이터 예시이다.

표 5는 본 논문에서 정의한 액추에이터와 관련된 메타데이터이다. 이 정보에는 액추에이터의 모델명, 종류, 제어 요청 종류, 최소값, 최대값, 측정 단위, 데이터 타입이 있다.

표 6은 led, RGB led, 서보 모터의 액추에이터 정

보를 메타데이터로 표현한 예시이다. request_type은 액추에이터 제어 요청의 종류를 기술한다. 제어 요청의 종류로는 이진법의 숫자를 의미하는 binary_number, 순서가 있는 값인 ordered_values, 범위가 있는 숫자인 range_of_numbers가 있다. 예를 들어, led의 on/off의 경우 0 또는 1의 값을 가짐으로써 request_type이 binary_number이다. 모터의 각도인 경우 1~180°

사이의 값을 가지기 때문에 request_type이 range_of_numbers가 된다.

메타데이터를 정의함으로써 디바이스 레지스트리에 IoT 디바이스를 등록하여 관리하고, 데이터를 수집할 테이블을 생성할 수 있다. 또한, 모니터링 툴을 통한 센서 데이터의 모니터링 및 IoT 디바이스의 제어가 가능하다.

표 3. 센서 메타데이터 정의
Table 3. Definitions of sensor metadata

Key	Data type	Description	Example
sensor_model	string	Describe the model name of the sensor.	DHT11
sensor	string	Describe the type of sensor.	humidity, temperature
min_value	integer	Describes the minimum value of the sensor value.	20
max_value	integer	Describes the maximum value of the sensor value.	90
unit_of_measure	string	Describe the unit for each type of sensor.	%
data_type	string	Describes the data type of the sensor value.	integer

표 4. 센서 정보의 예
Table 4. Examples of sensor information

sensor_model	Sensor	min_value	max_value	unit_of_measure	Data type
DHT11	humidity	20	90	%	integer
	temperature	0	50	°C	integer
CDS	light	0	1023		integer
GP2Y10	dust	0	250	ug/m3	float

표 5. 액추에이터 메타데이터 정의
Table 5. Definitions of actuator metadata

Key	Data type	Description	Example
actuator_model	string	Describe the actuator model name.	DM562
actuator	string	Describe the type of actuator.	red_LED
request_type	string	Describes the type of actuator control request.	binary_number
min_value	integer	Describes the minimum value of actuator control requests.	0
max_value	integer	Describes the maximum value of actuator control request.	1
unit_of_measure	string	Describe the unit for each type of actuator.	°
data_type	string	Describes the data type of the actuator control request.	integer

표 6. 액추에이터 정보 메타데이터의 예
Table 6. Example of actuator information metadata

actuator_model	Actuator	request_type	min_value	max_value	unit_of_measure	data_type	Example
DM562	red_LED	binary_number	0	1		integer	0
DM52	rgb_LED	ordered_values	0	255		string	0 0 255
SH90	servo_motor	range_of_numbers	1	180	°	integer	60

3.4 MQTT 메시지의 토픽 정의

이 절에서는 MQTT에서 전달되는 메시지를 위한 토픽의 정의 방법에 대해 기술한다.

MQTT는 토픽을 통해 슬래시(/)를 분리 문자(Delimiter)로 사용하여 서로 다른 클라이언트 간 메시지를 주고받을 수 있다. 제안 시스템에서는 'iot/sensor' 또는 'iot/actuator/system_id' 형식으로 토픽을 정의한다. 이때 'iot'는 데이터를 받는 모듈인 센서 DB 관리자를 나타내며 sensor, actuator를 통해 해당 데이터가 센서 데이터인지 제어 요청인지를 구분한다. 그림 2는 제안 시스템의 MQTT 발행/구독의 구조를 통한 메시지 전달 과정을 보인다. 사용자의 제어 요청은 웹 브라우저로 입력받아 모니터링 툴을 통해 MQTT 브로커로 발행된다. IoT 디바이스로 수집된 센서 데이터는 토픽을 통해 MQTT 브로커로 발행되며, 동시에 MQTT 브로커로부터 제어 요청을 전달받아 다양한 사물을 제어한다.

IV. 구현 결과

이 장에서는 제안 시스템의 구현 결과를 기술한다. 이를 위하여 IoT 디바이스를 실제로 개발하고 이를 통하여 제안 기법이 어떻게 디바이스를 제어

하는지를 예제를 통하여 보인다.

표 7은 제안 시스템 구성 요소들의 구현 환경이다. MQTT 브로커는 보편적으로 사용되는 mosquitto를 사용한다[27]. Mosquitto는 MQTT 브로커가 가져야 할 대부분의 기능을 충분히 지원하고 소형 디바이스에 올려 사용하기에 적합해 IoT 환경에서 많이 사용된다[28]. 본 연구는 서버에서 설치해 아두이노와 센서 DB 관리자 사이의 데이터 처리를 위해 사용된다.

표 7. 서버 구현 환경

Table 7. Server implementation environment

Features	Specification
OS	Window 10(x64)
MQTT broker	mosquitto
Database	MySQL
Device registry	JSP, Tomcat 8.5
Sensor DB manager	Python
Monitoring tools	Java script, Node.js

평가에 사용될 IoT 디바이스는 미세먼지 수집기로서 메카 솔루션에서 제공하는 교육용 미세먼지 센서 키트를 기반으로 센서 데이터 측정뿐만 아니라 액추에이터의 원격 제어도 가능하도록 구현한다 [29]. 표 8은 미세먼지 키트의 스펙이고, 그림 3은 구현된 미세먼지 수집기를 보인다.

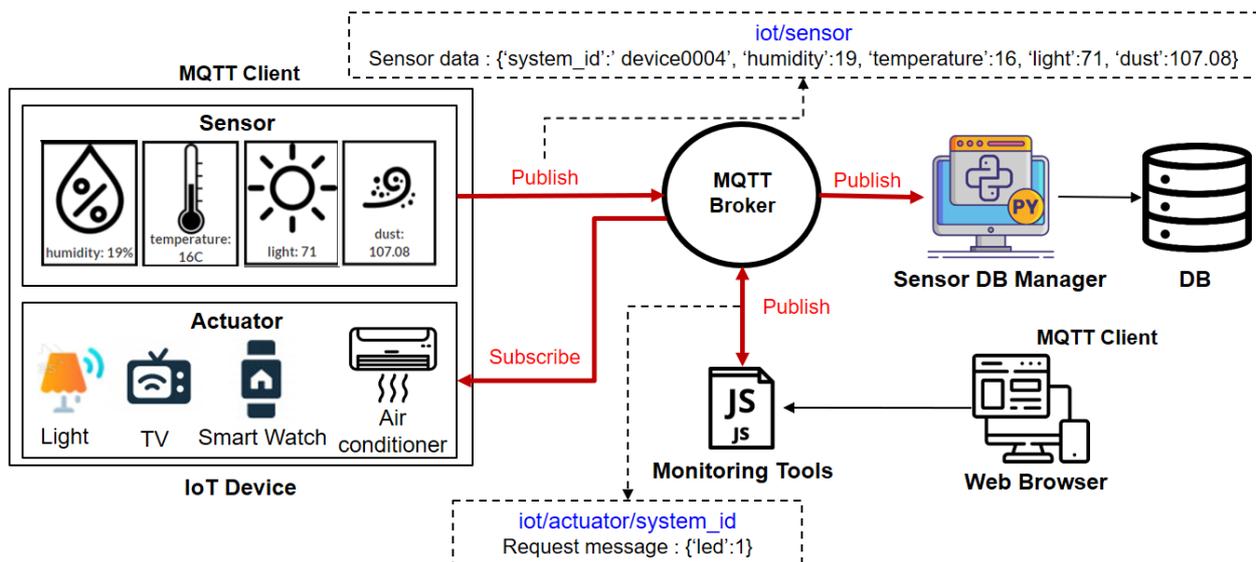


그림 2. MQTT 기반의 메시지 전달 과정
Fig. 2. MQTT-based message delivery process

표 8. 미세먼지 키트의 스펙

Table 8. Specifications of the fine dust kit

Features	Specification
Platform	Arduino Uno, Raspberry Pi 4
Sensors	DHT11 (temperature, humidity), CDS (light), GP2Y10 (dust)
Actuator	RGB led, 1602 LCD
Communication	HC-06 (bluetooth)



그림 3. 미세먼지 수집기의 구현
Fig. 3. Implementation of fine dust collector

부착된 센서 및 액추에이터의 종류에는 온습도 센서, 조도 센서, 미세먼지 센서, RGB led 등이 있다. 이 외에도 LCD 모니터, 블루투스 모듈이 있다.

부착된 센서로 데이터를 측정한 후, LCD 모니터에 온도(T), 습도(H), 미세먼지(D), 조도 값(L)을 출력한다. 이는 다시 블루투스 통신을 통해 라즈베리 파이로 전송된다. RGB led 색상은 기본적으로 미세먼지 값에 따라 좋음, 평균, 나쁨 세 가지 기준으로 분류되며 사용자의 입력에 따라 제어가 가능하다.

표 9. 특정 정보에 등록된 디바이스의 메타데이터
Table 9. Metadata of a device registered in specific information

Group	Key	Value
1	sensor_model	DHT11
1	sensor	temperature
1	unit_of_measure	C
2	sensor_model	DHT11
2	sensor	humidity
2	unit_of_measure	%
3	sensor_model	CDS
3	sensor	light
4	sensor_model	GP2Y10
4	sensor	dust
4	unit_of_measure	ug/m ³
5	actuator_model	rgb_LED
5	actuator	led
5	measure_type	binary_number

미세먼지 수집기를 제어하기 위해서 먼저, 디바이스 레지스트리에 등록한다. 표 9는 특정 정보에 등록된 디바이스의 메타데이터를 보인다. 사물 제어 실험에 사용될 액추에이터는 led를 사용한다. 디바이스 레지스트리에 등록된 정보를 이용해 모니터링 툴이 구성된다. 모니터링 툴 화면에 나타난 led 액추에이터를 클릭하면 MQTT를 통해 미세먼지 수집기가 동작한다. 그림 4는 구현된 미세먼지 수집기로부터 수집된 센서 데이터 및 액추에이터의 상태를 보인다. 그림 5는 웹 브라우저로 접속한 모니터링 툴의 스크린샷을 보인다.

id	timestamp	humidity	temperature	light	dust	led
22874	2021-08-13 15:50:01	55	22	137	766.9922	0
22873	2021-08-13 15:49:56	55	22	129	770.3125	0
22872	2021-08-13 15:49:50	54	22	123	769.4824	0
22871	2021-08-13 15:49:44	54	22	128	766.9922	1
22870	2021-08-13 15:49:39	54	22	127	772.8028	1
22869	2021-08-13 15:49:35	54	22	146	771.1426	1
22868	2021-08-13 15:44:19	55	22	115	104.5898	1
22867	2021-08-13 15:44:14	55	22	118	101.2695	1
22866	2021-08-13 15:44:09	55	22	118	114.5508	1
22865	2021-08-13 15:44:04	55	22	118	101.2695	1

그림 4. 수집된 센서 데이터 및 액추에이터 상태
Fig. 4. Collected sensor data and status of actuator

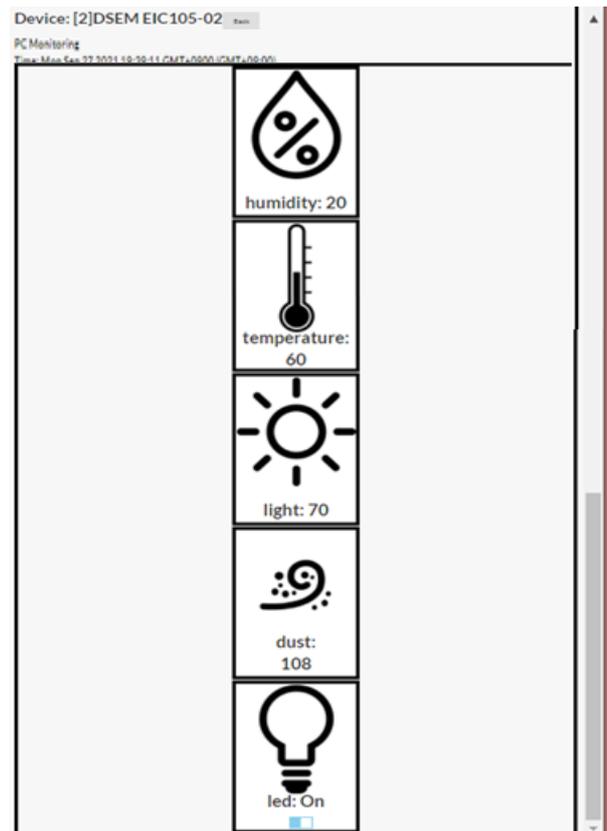


그림 5. 모니터링 툴의 스크린샷
Fig. 5. Screenshot of a monitoring tool

실험 결과, 제어 요청 입력으로부터 IoT 디바이스의 제어까지 약 1~2초 정도의 시간이 소요되며, 사용자의 요청에 따라 실시간으로 사물을 제어할 수 있음을 확인한다.

V. 시나리오

이 장은 제안 시스템이 제한된 환경에서 어떻게 이동성 및 상호운용성이 강화되는지를 보여주는 시나리오를 기술한다.

본 연구는 시나리오를 위하여 다음과 같이 가정한다. 일반적인 사무실 및 집에서는 에어컨, 전등, 블라인드, 등과 같은 다양한 사물이 존재하며, 모든 사물이 인터넷이 연결되어 있지 않을 수도 있다. 이러한 사물들을 인터넷에 연결하여 IoT 환경을 구현하고자 할 때, 스위치봇(Switchbot)[30]과 같은 IoT 디바이스는 좋은 해결책이 될 수 있다. 스위치봇은 인터넷에 연결되어 원격으로 버튼을 클릭해 줄 수 있는 IoT 디바이스이다. 따라서, 이 시나리오는 서보모터(Servo motor)를 이용한 스위치봇을 개발하고, 이 디바이스가 제안 시스템에서 어떻게 동작하는지를 보이게 한다. 또한, 기존 스위치봇과 제안 시스템에서의 스위치봇을 비교하여 사물의 위치가 이동했을 시 어떻게 변화하는지를 기술한다.

먼저, 사용자는 사무실 내의 전등 스위치에 스위치봇을 부착하여 전등을 원격으로 켜고 끌 수 있도록 하고자 한다. 이때, 스위치봇은 1~180° 사이의 각도를 가지는 서보모터를 이용하여 각도 조절에 의해 스위치를 누르도록 설계하였으며, Wi-Fi를 통해 인터넷에 직접 접속할 수 있도록 개발한다.

이때, 상호운용성의 향상을 위하여 디바이스 레지스트리에 개발된 스위치봇의 정보를 등록하도록 한다. 그림 6은 디바이스 레지스트리에 등록된 스위치봇의 정보를 보여주는 스크린샷이다. 그림과 같이 등록이 완료되면 해당 디바이스의 상태 정보를 수집하기 위한 테이블이 데이터베이스에 자동으로 생성된다. 그림 7은 개발된 스위치봇의 모니터링을 위한 테이블을 보인다.

모니터링을 위하여 제안 시스템은 액추에이터인 서보모터의 상태 값을 5초 주기로 측정하여 MQTT

브로커로 발행한다. 이때, 'iot/sensor'의 토픽으로 {'system_id': 'switchbot01', 'servo_motor':160}과 같은 메시지가 전송된다. 이 메시지를 센서 DB 관리자가 구독하여 전송받고 파싱하여 모니터링 테이블에 저장한다. 이후, 사용자는 모니터링 툴을 이용하여 해당 테이블의 데이터를 실시간으로 받아와 스위치봇의 상태를 모니터링한다. 현재 디바이스 레지스트리에 등록된 센서와 액추에이터 정보는 서보모터 하나만 존재하므로 하나만 존재하므로 이 모듈만 모니터링 된다. 그림 8은 스위치봇의 모니터링 툴의 스크린 샷을 보인다.

Item Detail

[6] Switchbot01

[modify](#) [back](#)

Common Information

Attribute	Value
Item id	6
Registration time	2021-10-29 13:05:59
Model name	Switchbot01
Device type	Raspberry Pi 4 (Arduino UNO)
Manufacturer	Dsem Lab.
Category	Actuator Device

Specific Information

Group	Key	Value
1	actuator_model	servo_motor
1	actuator	SG90
1	unit_of_measure	°
1	request_type	range of numbers
1	min_value	1
1	max_value	180

그림 6. 스위치봇에 대한 디바이스 레지스트리 스크린샷
Fig. 6. Screenshot of device registry for switchbot

	id	timestamp	servo_motor
▶	1	2021-09-27 19:39:11	0
	2	2021-09-27 19:39:16	0
	3	2021-09-27 19:39:21	0
	4	2021-09-27 19:39:26	160
	5	2021-09-27 19:39:31	160

그림 7. 스위치봇의 모니터링을 위한 테이블
Fig. 7. DB table for monitoring switchbot

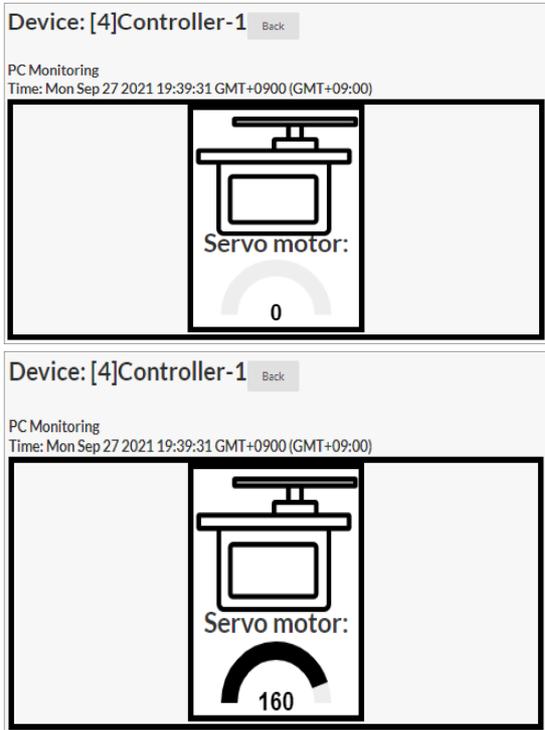


그림 8. 서보모터 설정 변경 스크린샷
Fig. 8. Screenshots of change of servo motor setting

사용자는 모니터링 툴을 이용하여 스위치봇의 서보모터 설정 값을 입력하면, 모니터링 툴은 MQTT 브로커로 ‘iot/actuator/switchbot01’의 토픽으로 {‘servo_motor’:60}과 같은 메시지를 발행한다. 스위치봇은

해당 메시지를 구독하여 전송받아 스위치를 작동시킴으로 전등을 제어할 수 있다.

두 번째 시나리오는 기존에 장착된 스위치봇을 다른 지역의 스위치로 옮기고자 할 때를 가정한다. 그림 9는 기존의 스위치봇과 제안 시스템을 이용한 스위치 봇의 차이를 보인다. 기존의 스위치봇 제품들은 대부분 스마트폰과 블루투스로 연결되거나, 자체적인 Wi-Fi AP(Access Point)를 통해 스마트폰의 특정 앱으로 설정을 변경하도록 한다. 이러한 방식은 사용자가 사물 제어를 위해 해당 제품 주변에 접근해야 한다는 문제를 지닌다.

또한, 스위치봇의 위치를 옮길 때 네트워크 환경의 변화가 발생할 경우 해당 네트워크의 설정을 다시 해야 하는 문제를 지닌다. 이는 스위치봇이 해당 지역에서 인터넷에 다시 접근하는 것과 함께 해당 플랫폼 내에서도 네트워크 설정을 다시 하도록 해야 한다.

하지만, 제안 시스템의 경우 스위치봇이 인터넷에 접속하여 플랫폼에 접근할 수 있다면 모니터링 및 사물 제어는 자동으로 진행할 수 있다. 또한, 네트워크가 변경될 시 스위치봇을 제어할 수 없는 기존과 달리 제안 시스템을 활용하면 디바이스의 네트워크가 변경되어도 쉽게 사물을 제어할 수 있다.

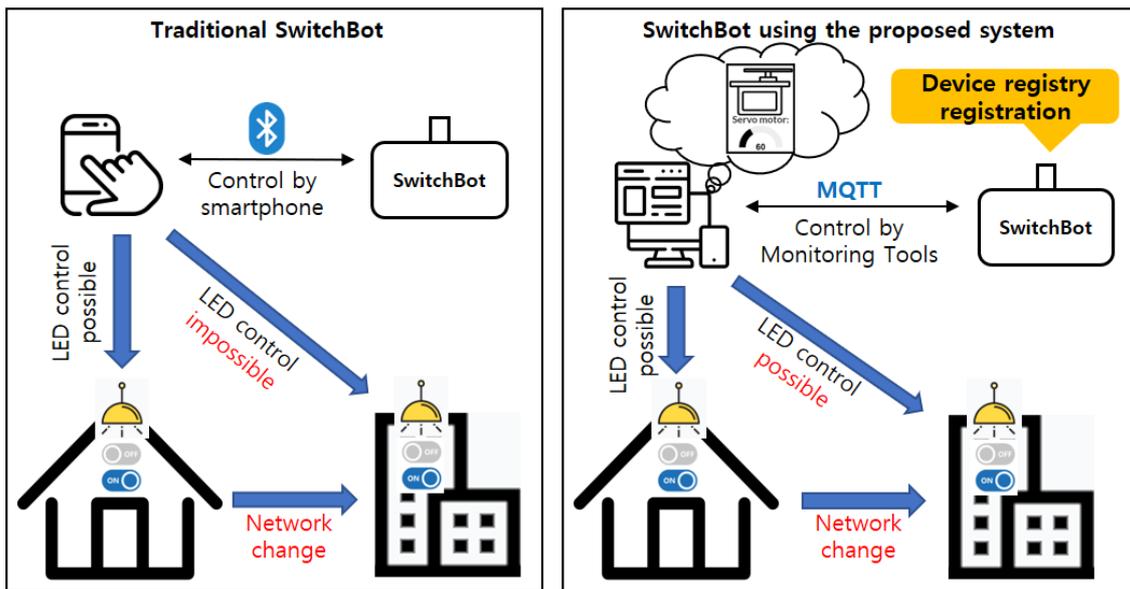


그림 9. 기존 스위치봇의 사용 방법과 제안 시스템을 적용한 방법
Fig. 9. How to use the existing switchbot and how to apply the proposed system

본 논문은 시나리오를 통하여 다음과 같이 제안 시스템의 두 가지 장점을 확인할 수 있다. 첫 번째는 디바이스 레지스트리를 통해 상호운용성을 향상시킬 수 있다. 기존 연구는 모니터링 툴을 통해 제어 요청이 DB에 저장되며 중간에 서버가 이를 가져와 처리한 후 IoT 디바이스로 전달하던 방식으로, 제어 요청의 메타데이터 정의가 이루어지지 않아, 제어할 수 있는 액추에이터의 종류가 한정되어 있다는 단점이 있다. 그러나 제안 시스템은 메타데이터를 정의하고 이를 디바이스 레지스트리에 활용함으로써 한정적인 제어 요청 종류를 가지는 기존 연구에 비해 다른 액추에이터 또한 제어할 수 있다.

두 번째는 MQTT를 사용하여 사물의 이동성을 강화시킬 수 있다. 예를 들어 사물이 집에서 회사로 네트워크 환경이 변경되었을 때, 집 와이파이에서 연결되어 있던 기존의 스위치봇은 회사로 이동된 사물을 제어할 수 없다. 그러나 제안 시스템은 모니터링 툴의 버튼을 클릭하면 제어 요청이 MQTT 브로커를 통해 바로 IoT 디바이스로 전달되므로 기존 스위치봇이 지니는 이동성의 한계를 극복할 수 있다.

VI. 결 론

기존 IoT 사물 제어 기법 관련 연구들은 자체적으로 정의된 표준 및 프로토콜을 사용해 확장성이 떨어지며 사물이 다른 장소로 이동했을 시 변화되는 네트워크 환경에 유연하게 대처할 수 없다는 문제를 지닌다.

따라서 본 논문은 다양한 네트워크 환경으로 이동하는 사물의 이동성 강화를 위해 MQTT 기반의 IoT 사물 제어 기법을 구현하였다. 이를 위하여, 네트워크 환경과 관계없이 사물을 제어하기 위한 방안으로 MQTT 기반의 메시지 전송 프로세스를 구현하였다. 또한, 다양한 IoT 디바이스의 정보를 유연하게 다루기 위해 디바이스 레지스트리를 사용하며, 모니터링뿐만 아니라 사물 제어를 위한 메타데이터를 정의함으로써 상호운용성을 높일 수 있었다.

본 연구의 검증을 위하여 스위치봇을 개발하는 시나리오를 작성하여 새로운 사물에 대한 디바이스 레지스트리 등록 과정, 제어 과정을 설명하였다. 그

결과, 네트워크 이동에 따른 기존 스위치봇과 제안 기법을 비교함으로써 제안 기법의 우수성을 검증하였다.

향후 연구로는 제안 기법을 이용하여 다양한 분야에서 IoT 디바이스를 직접 개발하고, 이동성이 강조된 환경에서 제안 시스템이 기존의 시스템에 비해 얼마나 환경에 잘 적응할 수 있는지를 평가할 계획이다.

Acknowledgement

2021년도 한국정보기술학회 하계종합학술대회에서 발표한 논문(유연한 사물 이동을 위한 웹 어플리케이션 기반 사물 제어 기법)을 확장한 것임

References

- [1] J. Jeon, N. Kim, J. G. Go, T. Park, H. Gang, and C. Pyo, "IoT device products and technology trend", Korea Institute Of Communication Sciences, Vol. 31, No. 4, pp. 44-52, Mar. 2014.
- [2] J. Jeon, "A study on the Techniques Trends and Prospects for Internet of Things", Convergence Security Journal, Vol. 14, No. 7, pp. 65-73, Dec. 2014.
- [3] SmartThings, <https://www.samsung.com/sec/apps/smartthings/> [accessed: Nov. 11, 2021]
- [4] S. Lee, D. Jeong, Y. Seo, and S. Lee, "Development of Device Registry for Flexible Information Management in IoT Platform", Proceedings of KIIT Conference, Cheongju Korea, pp. 323-327, Oct. 2020.
- [5] Sk. F. Islam, Md. I. Hasan, M. Akter, and M. S. Uddin, "Implementation and Analysis of an IoT-Based Home Automation Framework", Journal of Computer and Communications, Vol. 9, No. 3, pp. 143-157, Mar. 2021. <https://doi.org/10.4236/jcc.2021.93011>.
- [6] H. Singh, V. Pallagani, V. Khandelwal, and U. Venkanna, "IoT based smart home automation

- system using sensor node", International Conference on Recent Advances in Information Technology, Dhanbad, India, pp. 1-5, Mar. 2018. <https://doi.org/10.1109/RAIT.2018.8389037>.
- [7] Stolojescu-Crisan C, Crisan C, and Butunoi B-P, "An IoT-Based Smart Home Automation System", *Sensors*, Vol. 21, No. 11, 3784, May 2021. <https://doi.org/10.3390/s21113784>.
- [8] Y. Sin, "A Study on Device Control Methods and Platform in IoT Environment", Korea University of Technology and Education, Feb. 2016.
- [9] J. Seo and C. Kim, "Design and Implementation of Realtime Things Control System Using MQTT and WebSocket in IoT Environment", *Journal of the Korea Electronics and Telecommunications Society*, Vol. 13, No. 3, pp. 517-524, Jun. 2018. <https://doi.org/10.13067/JKIECS.2018.13.3.517>.
- [10] Y. Kim, S. Lee, and J. Kim, "Device-flexible Access Control in IoT", *The Korea Institute of Information and Communication Engineering*, pp. 392-394, Dec. 2015.
- [11] H. Park, "Development of IoT Hub System for Smart Home Control based on Cloud", *The Society of Convergence Knowledge*, Vol. 6, No. 2, pp. 59-65, Jul. 2018.
- [12] MQTT, <https://ko.wikipedia.org/wiki/MQTT>. [accessed: Sep. 24, 2021]
- [13] Olivier Deschambault, Abdelouahed Gherbi, and Christian Légaré, "Efficient Implementation of the MQTT Protocol for Embedded Systems", *JIPS(Journal of Information Processing Systems)*, Vol. 13, No. 1, pp. 26-39, Feb. 2017. <http://dx.doi.org/10.3745/JIPS.04.0028>.
- [14] K. Lee, S. Seol, and S. Kang, "A Study on Applying CoAP to Remote Control Service", *Asia-pacific Journal of Multimedia Services Convergent with Art, Humanities, and Sociology*, Vol. 8, No. 7, pp. 809-818, Jul. 2018. <http://dx.doi.org/10.35873/ajmahs.2018.8.7.079>.
- [15] XMPP, www.xmpp.org/about-xmpp/history/ [accessed: Oct. 5, 2021]
- [16] W. Jang, B. Kim, and M. Kang, "Analysis of the AMQP for Data Message Queuing of Korean e-Navigation Operation", *Proceedings of the Korean Institute of Navigation and Port Research Conference*, pp. 22-24, Nov. 2017.
- [17] Y. Jang, J. Shim, and S. Park, "Analysis Standardized of IoT-based Low-power·Light-weight Protocol", *Journal of the Korea Institute of Information and Communication Engineering*, Vol. 20, No. 10, pp. 1895-1902, Oct. 2016. <https://doi.org/10.6109/jkiice.2016.20.10.1895>.
- [18] H. Jun, D. Lee, and H. Yang, "Performance Analysis of Data Exchange using XMPP and MQTT in Microgrid Platform", *The Korean Institute of Electrical Engineers*, Vol. 14, No. 16, pp. 369-370, Jul. 2018. <https://doi.org/10.3390/en14165024>.
- [19] Zujie Fan and J. Kim, "Transmission Performance Comparison and Analysis with Different Publish/Subscribe Protocol", *Proceedings of the Korean Society of Computer Information Conference*, Vol. 28, No. 2, pp. 77-80, Jul. 2020.
- [20] S. Chun, H. Kim, C. Ham, Y. Chung, and J. Park, "Reliable Mobility Management Using CoAP in Internet of Things Environments", *Journal of The Institute of Electronics and Information Engineers*, Vol. 53, No. 8, pp. 13-18, Aug. 2016. <https://doi.org/10.5573/ieie.2016.53.8.013>.
- [21] U. Kim and J. Choi, "Implementation of IoT Home System based on MQTT", *The Journal of The Institute of Internet, Broadcasting and Communication (IIBC)*, Vol. 20, No. 1, pp. 231-237, Feb. 2020. <http://dx.doi.org/10.7236/IIBC.2020.20.1.231>.
- [22] J. Lee, H. Kim, and D. Kim, "Implementation of a Smart Home IoT system based on Bluetooth controlled by Users", *Journal of Digital Contents Society*, Vol. 22, No. 2, pp. 331-338, Feb. 2021. <http://dx.doi.org/10.9728/dcs.2021.22.2.331>.

- [23] S. Eom, H. Noh, C. Kim, and K. Lee, "Implementation of IoT Control System with Device Virtualization Based on Power Line Communication", *Asia-pacific Journal of Multimedia Services Convergent with Art, Jumanities, and Sociology*, Vol. 9, No. 1, pp. 687-698, Jan. 2019. <http://dx.doi.org/10.35873/ajmahs.2019.9.1.069>.
- [24] J. Park, O. Kwon, B. Kim, W. Noh, J. Park, J. Jeon, and B. Ko, "Controlling of Remote Smart Window Using IoT Sensor Technology", *Korea Information Science Society*, pp. 414-416, Dec. 2018.
- [25] TTA, http://www.tta.or.kr/data/ttas_list_N.jsp [accessed: Sep. 24, 2021]
- [26] National Radio Research Agency, <https://www.rra.go.kr/ko/index.do> [accessed: Oct. 6. 2021]
- [27] Mosquitto, <https://mosquitto.org/download/> [accessed: Oct. 28. 2021]
- [28] S. Kim, D. Kim, H. Oh, H. Jeon, and H. Park, "The Data Collection Solution Based on MQTT for Stable IoT Platforms", *Journal of the Korea Institute of Information and Communication Engineering*, Vol. 20, No. 4, pp. 728-738, Apr. 2016. <http://dx.doi.org/10.6109/jkiice.2016.20.4.728>.
- [29] MecahSolution, <https://mechasolution.com/shop/main/index.php> [accessed: Sep. 24, 2021]
- [30] Switchbot, https://www.wakers.shop/product/switchbot_w [accessed: Oct. 28. 2021]

정 동 원 (Dongwon Jeong)



1997년 2월 : 군산대학교
컴퓨터과학과(학사)
1999년 2월 : 충북대학교
전자계산학과(석사)
2004년 2월 : 고려대학교
컴퓨터학과(박사)
2005년 4월 ~ 현재 : 군산대학교

소프트웨어학과 교수

관심분야 : 데이터베이스, 시맨틱 서비스, 빅데이터,
사물인터넷, 엣지컴퓨팅, 지능형 융합 서비스

이 석 훈 (Sukhoon Lee)



2009년 2월 : 고려대학교
전자및정보공학부(학사)
2011년 2월 : 고려대학교
컴퓨터·전파통신공학과(공학석사)
2016년 2월 : 고려대학교
컴퓨터·전파통신공학과(공학박사)
2016년 3월 ~ 2017년 3월 :
아주대학교 의료정보학과

연구강사

2017년 4월 ~ 현재 : 군산대학교 소프트웨어학과 부교수
관심분야 : 사물인터넷, 메타데이터, 센서 레지스트리,
시맨틱 웹, 경로 예측

저자소개

장 우 린 (Woolin Jang)



2022년 2월 : 군산대학교
소프트웨어융합공학과(학사)
2022년 3월 ~ 현재 :
(주)윈드밀소프트 사원
관심 분야 : 사물인터넷, 센서
레지스트리