JKIICE Journal of the Korea Institute of Information and Communication Engineering

한국정보통신학회논문지(J. Korea Inst. Inf. Commun. Eng.) Vol. 21, No. 10 : 1943~1949 Oct. 2017

oneM2M 표준 프로토콜 기반의 개방형 IoT 플랫폼 설계

명상일[1] · 김성대[2*]

The design of Open IoT Platform based on oneM2M Standard Protocol

Sang-Il Myung[1] · Sung-Dae Kim[2*]

[1,2*]Department of of Electrical Eng., Dongmyong University, Busan, 48520, Korea

요 약

최근 메이커 문화는 스스로 '만들다'라는 개념을 하이테크놀로지 최신 기술을 통해 실현시키고 있다. 개인의 제작 방법 및 정보를 인터넷을 통해 서로 공유함으로서 메이커 운동으로 빠르게 확산되고 있다. 메이커 운동을 선도하는 국가들은 메이커 운동을 사물인터넷 관점에서 새로운 가치 창조 및 경제 성장의 동력이 될 것이라고 전망하고 있다. DIY (Do It Yourself)를 통해 제작하고 개발된 다양한 사물인터넷 디바이스와 서비스를 등록하고, 상호 연동하여 서비스를 실현할 수 있는 다양한 형태의 개방형 IoT (Internet of Things) 플랫폼을 개발하고 있다. 일반적인 개방형 IoT 플랫폼들은 전문적인 API (Application Programming Interface) 분석 능력이 필요하며, 까다로운 플랫폼 등록조건과 절차로 인해 일반 사용자들의 접근이 어려운 실정이다. 본 논문에서는 일반 메이커 사용자의 관점에서 보다 간편한 구조의 사물인터넷 디바이스와 서비스 애플리케이션을 제작하고, 실제 서비스로 구동 할 수 있는 개방형 IoT 플랫폼을 구성하고자 한다.

ABSTRACT

In recent years, the maker culture has realized the concept of 'making' itself through the latest technology of high technology. It is rapidly spreading as a maker's movement by sharing the production methods and information of individuals through the Internet. Countries that are leading the makers' movement expect the maker's movement to be the driving force for new value creation and economic growth from the point of view of things and the Internet. We are developing various types of open IoT platforms that can register various Internet devices and services manufactured and developed through DIY and realize interoperable services. Typical open IoT platforms require specialized API analysis capabilities and difficult access to general users due to difficult platform registration conditions and procedures. In this paper, we try to construct an open IoT platform that can be operated as a real service by creating a simple Internet application device and service application from a general manufacturer 's point of view.

키워드 : 클라우드, 사물인터넷, 원엠투엠, MQTT 프로토콜, 플랫폼

Key word : Cloud, IoT, oneM2M, MQTT Protocol, Platform

Received 15 September 2017, **Revised** 21 September 2017, **Accepted** 28 September 2017

* **Corresponding Author** Sung-Dae Kim(E-mail : jbksd@tu.ac.kr, Tel:+82-51-629-1315)
Department of of Electrical Eng., Dongmyong University, Busan, 48520, Korea

Open Access https://doi.org/10.6109/jkiice.2017.21.10.1943 print ISSN: 2234-4772 online ISSN: 2288-4165

Ⅰ. 서 론

최근 들어 자신이 필요한 무언가를 자기 스스로 만드는 DIY(Do It Yourself)나, 메이커(Maker) 문화를 많이 볼 수 있다. 이러한 메이커 문화는 스스로 '만들다'라는 개념을 오픈소스 소프트웨어, 하드웨어, 3D 프린팅과 같은 하이테크놀로지 최신 기술을 통해 실현시키고 있으며, 개인의 제품 제작 방법 및 정보를 온·오프라인을 통해 서로 공유함으로서 메이커 운동으로 빠르게 확산되고 있다.

현재 미국, 호주 등 메이커 운동을 선도하는 국가들은 메이커 운동을 사물인터넷 관점에서 중요한 신산업 아이템으로 인식하고 있으며 메이커 활동이 일자리 창출, 제조업 혁신 등의 새로운 가치 창조 및 경제 성장의 동력이 될 것이라고 전망하고 있다[1,2].

이에 우리나라에서도 메이커 문화를 활성화하기 위해 정부 기관들의 주관으로 메이커 스페이스를 설립하고, 사물인터넷 DIY 교육과 장비 보급을 위한 다양한 정책적 지원을 확대하고 있다. 하지만 현재 정부의 정책은 단순히 메이커 스페이스 설립 · 운영 위주로 편중되어 있으며 낮은 사용자 접근성, 전문 인력의 부족, 특화 장비의 미비, DIY 연계 인프라 부족 등 기술적·환경적 문제로 인해 메이커 문화의 활성화가 어려운 실정이다. 특히 DIY로 제작되는 대부분의 사물인터넷 디바이스는 IoT (Internet of Things) 기술을 통해 서비스 애플리케이션과 연동되어 되지 못하고 단순한 제품 조립과 제작에 그치고 있는 실정이다. 사물인터넷 서비스의 경우 인터넷 네트워크를 통한 다양한 IoT 디바이스 및 서비스 간의 연동이 필수적인 요소이다. 이에 전 세계적으로 DIY를 통해 제작하고 개발된 다양한 사물인터넷 디바이스와 서비스를 등록하고, 상호 연동하여 서비스를 실현할 수 있는 다양한 형태의 개방형 IoT 플랫폼을 제공하고, 이를 활용한 실증 서비스를 유도하고 있다[3,4].

이러한 개방형 IoT 플랫폼들은 oneM2M (one machine-to-machine) 표준 플랫폼 구조로 각기 다른 서비스군의 디바이스 특징별로 효율적인 통신 프로토콜과 연동을 위한 상세한 API (Application Programming Interface)와 라이브러리를 제공하고 있다. 그러나 전문 개발자가 아닌 일반 메이커들이 전문 지식 없이 이들 API를 분석하여 자신이 필요한 디바이스와 서비스를 개발하기란 매우 어려운 일이며, 까다로운 플랫폼 등록 조건과 절차는 간단하게 만들어보고 구동해보고자 하는 일반 사용자들의 접근이 어려운 실정이다. 따라서 본 논문에서는 전문 지식이 없는 일반 메이커 사용자의 관점에서 보다 쉽고 간편한 구조로 자신이 필요로 하는 사물인터넷 디바이스와 서비스 애플리케이션을 제작하고, 이를 실제 서비스로 연동/구동 할 수 있는 경량화된 개방형 IoT 플랫폼을 제한한다. oneM2M 표준 플랫폼의 구조를 따라 MQTT (MQ Telemetry Transport) 프로토콜(protocol)를 기반으로 애플리케이션 연동을 위한 웹 소켓 (web socket) 통신 구조를 설계하고 이를 적용한 클라우드 플랫폼 서버를 구현하였다.

Ⅱ. 관련 연구

2.1. MQTT

IoT 서비스를 구성하기 위해서는 IoT 디바이스로부터 측정된 특정 정보에 대한 실시간 커뮤니케이션 기술이 필수적인 요소로 작용하면서 주요 실시간 프로토콜로 XMPP (Extensible Messaging and Presence Protocol), CoAP (Constrained Application Protocol), MQTT 등이 주목 받고 있다[5].

특히 MQTT는 다양한 IoT 애플리케이션과 서비스가 개발되면서 기존의 HTTP와 같은 프로토콜만으로는 장치 간 커뮤니케이션의 요구사항을 수용할 수 없게 됨에 따라 제한된 통신 환경을 고려하여 낮은 전력, 낮은 대역폭 환경에서 사용이 가능하도록 설계되었으며 임베디드, 모바일 기기에 최적화된 라이트 메시징 프로토콜로써 M2M (machine-to-machine)과 IoT를 위한 대표적인 표준 프로토콜로 활용되고 있다[6].

MQTT 프로토콜의 가장 큰 특징은 메시지를 발행(publishing) 하고, 관심 있는 주제를 구독(subscribe) 하는 것을 기본 원칙으로 한다는 것이다.

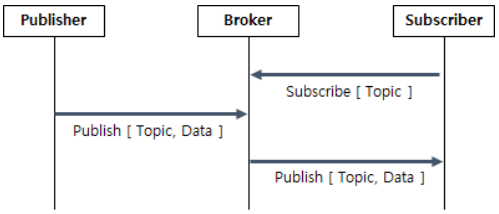

Fig. 1 MQTT structure

그림 1은 발행과 구독 간의 MQTT 프로토콜을 통한 메시지 전송의 기본 구조로써 발행자는 토픽을 발행하기 위한 목적으로 구독자는 토픽을 구독하기 위한 목적으로 브로커(broker)에 연결되며 발행자와 구독자는 모두 브로커에 대한 클라이언트로 동작한다. 이때 구독자가 특정 토픽(topic)을 구독한 상태라면 발행자로부터 해당 토픽으로 발행된 메시지를 브로커를 통해 수신 받을 수 있으며 다수의 구독자가 하나의 토픽을 구독하거나 하나의 수신자가 다수 토픽에 대한 구독을 할 수 있다[7]. 이처럼 MQTT 프로토콜에서 발행과 구독간의 메시지 교환은 토픽을 기준으로 이루어진다. 토픽은 문자열로 표현하고 슬래시(/)를 이용해서 계층적으로 구성할 수 있어서 다수의 디바이스로부터 송수신되는 대량의 데이터들을 효율적으로 관리 할 수 있다. 예를 들어 MQTT 프로토콜 기반의 스마트 팜 시스템을 구축하여 온도/습도와 같은 농장의 환경 요소나 생상/소비되는 전력에 대한 모니터링을 한다고 가정한다면 그림 2와 같은 계층 구조를 가지는 토픽을 구성할 수 있다.

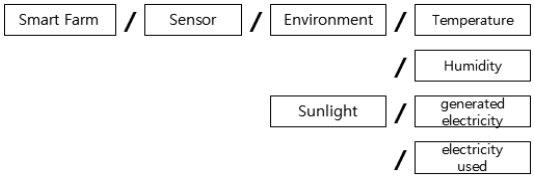

Fig. 2 Topic hierarchy

2.2. Web Socket

웹 소켓 프로토콜은 전이중(full- duplex)을 제공하는 양방향 통신채널로 ws 프로토콜을 기반으로 클라이언트와 서버 사이에 지속적인 완전 양방향 연결 스트림을 만들어 주는 기술이다. 웹 소켓은 하나의 소켓을 통해 확장성과 실시간성을 보장하여 웹 애플리케이션을 구축하는 것이 가능하며 웹 서버와 웹 브라우저 상에 구현되어 두 지점 간에 실시간 상호 작용하도록 지원한다. 기존의 HTTP 프로토콜 기반으로 한 서버 시스템의 전송 효율 문제, 시스템 과부하의 문제점을 해결하고 IoT 서비스와 같이 실시간성이 요구되는 응용 프로그램을 한층 효과적으로 구현할 수 있으며 그림 3과 같이 Opera Mini 브라우저를 제외한 대부분의 최신 버전 의 웹 브라우저에서 웹 소켓을 지원하고 안드로이드 브라우즈와 iOS 사파리(Safari) 등 모바일 웹 환경에서도 웹

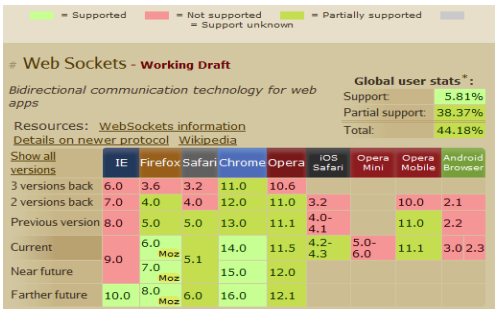

Fig. 3 Web Socket Supported browsers

소켓을 지원하기 함에 따라 모바일 환경에서도 다양한 활용이 가능하다[8,9]. 또한 웹 소켓은 그림 4와 같이 특정 네임스페이스 (namespace)를 통해 임의의 채널을 지정하는 룸 (room) 기능을 지원한다. 룸은 말 그래도 그 방에 있는 사람들끼리만 메시지를 주고받을 수 있는 기능으로 만일 MQTT 프로토콜을 지원하는 않는 특정 디바이스나 애플리케이션의 경우 특정 토픽 명으로 룸을 생성하고 클라이언트를 Join 시킴으로써 MQTT 프로토콜의 발행과 구독과 동일한 통신 구조의 메시지 교환 기능을 지원할 수 있다.

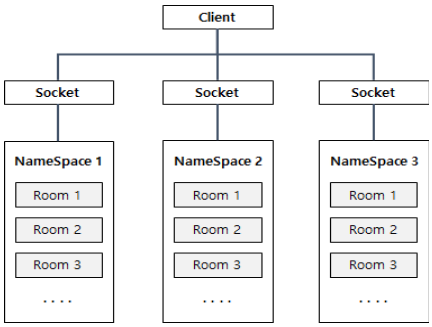

Fig. 4 Web Socket Room

Ⅲ. 시스템의 설계

본 논문에서는 메이커 활동을 지원하기 위한 개방형 클라우드 서버를 구축하기 위해 oneM2M 국제 표준 프로토콜 기반의 메시지 송수신이 가능한 MQTT 프로토콜을 활용하며 서버와 애플리케이션 간의 효율성과 실시간성을 보장하기 위해 웹 소켓을 활용한다. 또한 클라우드 서버와 연계하여 다양한 환경에서의 클라이언

트 애플리케이션 제작을 지원하기 위한 서비스 모듈을 설계·구현 하였다.

3.1. IoT Cloud Server

IoT 클라우드 서버는 IoT 디바이스와 서비스 애플리케이션 간의 MQTT 메시지를 교환하고 수집된 데이터에 대한 관리/활용을 위한 웹 서버로써 MQTT 프로토콜을 지원하기 위해 MQTT 브로커를 서버 내부에 구축하였으며 서비스 서버가 MQTT 브로커의 클라이언트로 동작하도록 구성하였다. 또한 서비스 애플리케이션의 확장성과 실시간성을 고려하여 즉각적인 메시지 전송을 위해 웹 소켓을 활용하였다.

그림 5는 본 논문에서 제안하는 IoT 크라우드 서버의 소프트웨어 모듈 구성도이다. 클라우드 서비스 서버의 MQTT 모듈은 서비스 서버를 MQTT 브로커에 연결하고 발행과 구독을 수행하기 위한 모듈로써 MQTT 요구자(requester)를 통해 애플리케이션으로부터 수신된 요청에 따라 해당 토픽에 대한 발행과 구독을 제어하고 메시지 리스너(message listener)를 통해 IoT 디바이스로부터 발행된 메시지를 수신한다. 소켓 모듈은 서비스 애플리케이션과의 서비스 서버 간 웹 소켓 통신을 지원하기 위한 모듈로써 Event Listener를 통해 애플리케이션에서 발생한 이벤트에 따라 MQTT 모듈로 발행과 구독 명령을 전송하고 Message Emitter를 통해 애플리케이션으로 구독에 대한 발생 메시지를 전송하거나 이벤트 처리에 대한 응답 메시지를 전송한다.

HTTP API 모듈은 서비스 애플리케이션과의 서비스 서버 간 API 통신을 지원하기 위한 모듈로써 사용자 계정 관리 또는 프로젝트 관리와 같은 IoT 서비스를 위한 애플리케이션의 API 요청에 따라 기능을 수행하고 결과 정보를 전송한다. Custom Data 모듈은 수신 데이터의 특성에 따라 개인 맞춤형 정보를 제공하기 위한 데이터의 커스터마이징을 제공하는 모듈로써 사용자의 설정에 따라 데이터의 생성, 수집, 분석을 기능을 수행한다.

Authority 모듈는 API 또는 소켓통신을 통한 애플리케이션의 기능 요청 시 사용 권한을 검증하기 위한 모듈로써 API 통신에서는 Session의 존재 유무를 통해 권한을 확인하고 웹 소켓 통신에서는 JWT(Json Web Token)을 통해 Token 데이터를 디코딩하여 올바른 토큰인지를 검증하여 권한을 확인한다. SQL Manager는 데이터베이스 조회, 등록, 수정 등의 Query를 수행하기 위한 모듈로 서비스 애플리케이션으로부터 수신되는 요청에 따라 SQL문을 생성하고 수행하며 결과 정보를 응답한다.

그림 6은 IoT 클라우드 서버의 서비스 서버에서의 구독이 진행되는 과정을 나타낸 것이다. 먼저 서비스 애플리케이션에서 사용자 로그인을 요청하면 서비스 서버에서는 수신된 사용자 정보를 통해 사용자를 확인하고 검증이 완료되면 고유 Token을 생성하여 응답 메시지와 함께 애플리케이션으로 전송한다. 이후 애플리케이션에서 구독 이벤트가 발생하면 서버에서는 이벤트 리스너를 통해 Token과 구독 할 토픽에 대한 정보를 수

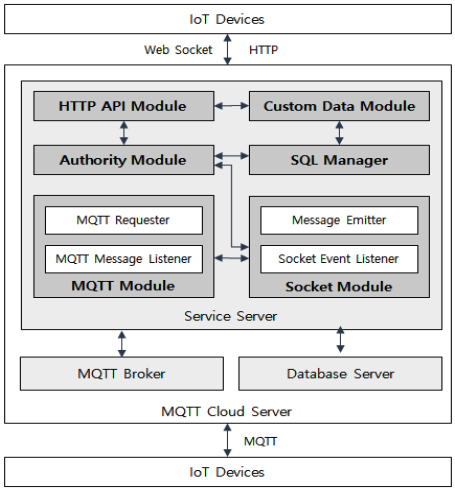

Fig. 5 IoT Cloud Server Structure

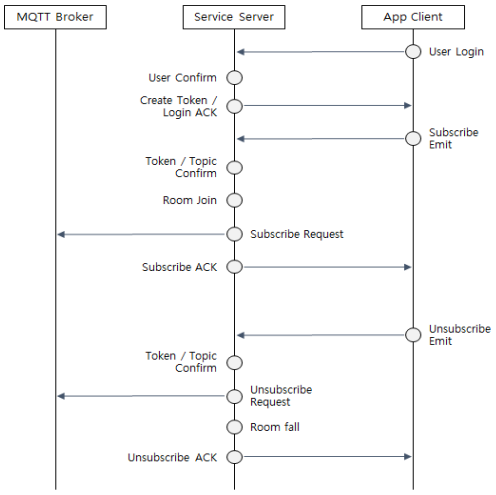

Fig. 6 Service Server Subscribe Process

신하고 해당 정보의 검증을 진행하여 검증이 완료되면 해당 토픽으로 소켓 품을 생성하고 클라이언트를 생성된 룸에 연결한다. 그 뒤 MQTT 브로커를 통해 수신된 토픽에 대한 구독을 진행하고 수행 결과를 애플리케이션으로 전송한다.

미구독의 경우 서비스 서버에서는 구독과 동일한 검증 과정을 거친 후 MQTT 브로커를 통해 해당 토픽의 구독을 중지하고 클라이언트를 소켓 룸에서 제거한 뒤 수행 결과를 응답한다. 발행의 경우도 역시 구독과 동일한 검증 과정을 거친 후 수신된 토픽 및 데이터를 MQTT 브로커로 발행하고 수행 결과를 응답한다.

3.2. Application Service Module

서비스 모듈은 IoT 크라우드 서버와 서비스 애플리케이션 간의 MQTT 메시지 교환을 지원하고 API 통신을 통해 특정 정보를 요청/응답 받는 모듈로써 애플리케이션에 라이브러리 형태로 적용된다.

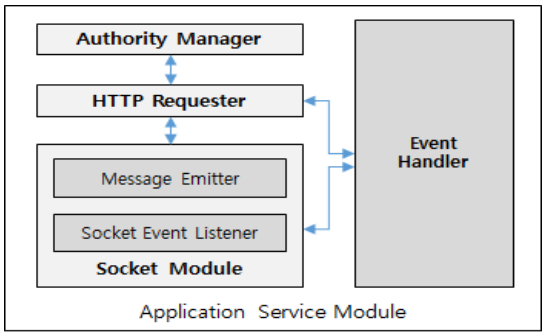

Fig. 7 Application Service Module Structure

그림 7의 소켓 모듈은 서비스 서버와 웹 소켓 연결을 관리하고 구독에 대한 발행 메시지를 수신하거나 발행과 구독에 대한 이벤트를 발생 시킨다.

HTTP Requester는 사용자 인증 또는 특정 정보에 대해 요청/응답을 위해 서비스 서버와의 API 통신을 지원하며 사용자 인증 시 수신된 Token 정보는 Authority Manager에 저장한다.

Event Handler는 소켓 모듈과 HTTP Requester에서 수신된 데이터를 애플리케이션으로 전송하기 위한 핸들러(handler)로써 메시지 수신 시 이벤트를 통해 메인 쓰레드로 전송한다.

Ⅳ. 구현 및 구동 테스트

4.1. 구현 결과

본 논문에서는 제안한 IoT 클라우드 서버를 구현하기 위해 서비스 서버를 싱글 쓰레드/이벤트 기반의 Nodjs를 활용하여 구현하였으며 MQTT 브로커로는 경량화되고 소형디바이스를 사용하는 사물인터넷 환경에서 사용하기에 적합한 오픈 소스인 Mosquitto를 사용하였다. 또한 실제 서비스를 제공하기 위해 AWS(Amazon Web Services)를 통해 윈도우 서버를 구축하였다[10].

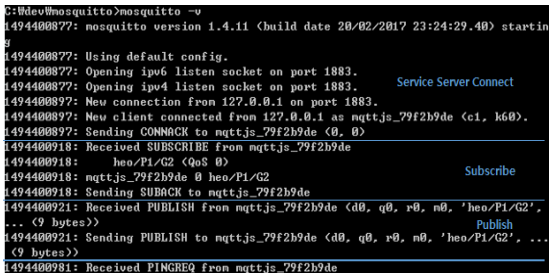

Fig. 8 MQTT Broker(Mosquitto) log

그림 8은 IoT 클라우드 서버의 내부에 구축한 Mosquitto의 동작 화면으로 웹 서버의 초기 구동 시 Mosquitto가 같이 구동되고 IoT 클라우드 서버 내에서 클라이언트 역할을 하는 서비스 서버가 연결되는 것을 확인 할 수 있다. 또한 디바이스나 서비스 애플리케이션의 구독 요청에 따라 해당 토픽에 대한 구독을 시작하고 그에 대한 SUBACK를 클라이언트로 전송하며 발

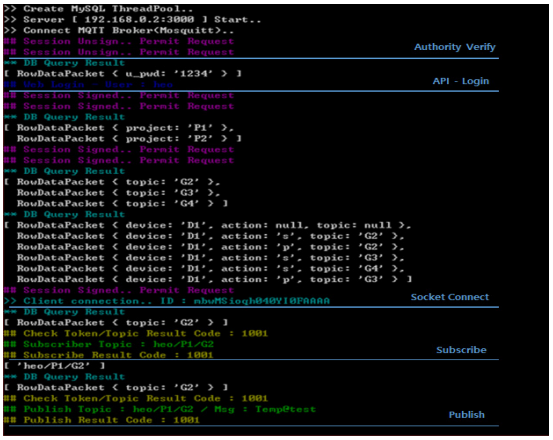

Fig. 9 Service Server log

행를 수신 받으면 해당 토픽에 대한 구독을 신청한 클라이언트에게 발행 메시지를 전송하는 것을 확인할 수 있다. 그림 9는 IoT 클라우드 서버의 서비스 서버 동작 화면으로 서비스 애플리케이션의 API 또는 소켓 통신을 통한 기능 수행 시 사용자 권한을 지속적으로 확인하는 것을 볼 수 있으며 발행과 구독을 위한 소켓이 연결되면 해당 클라이언트의 정보를 확인하고 저장한다. 또한 클라이언트로부터 구독이나 발행 요청이 수신되면 Token과 토픽 정보를 확인하고 해당 토픽에 대한 발행과 구독이 진행되는 것을 확인할 수 있다.

4.2. 구동 테스트

본 논문에서는 구현된 IoT 클라우드 서버와 어플리케이션 서비스 모듈의 성능을 검증하기 위해 아두이노 기반의 IoT 디바이스 활용하여 안드로이드와 C#을 대상으로 스마트 홈 모니터링 시스템을 구축하였다.

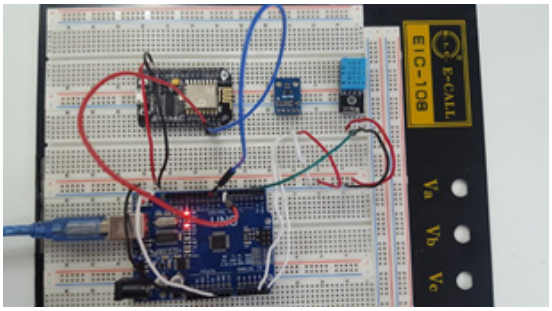

Fig. 10 DIY Smart Home Device Sample

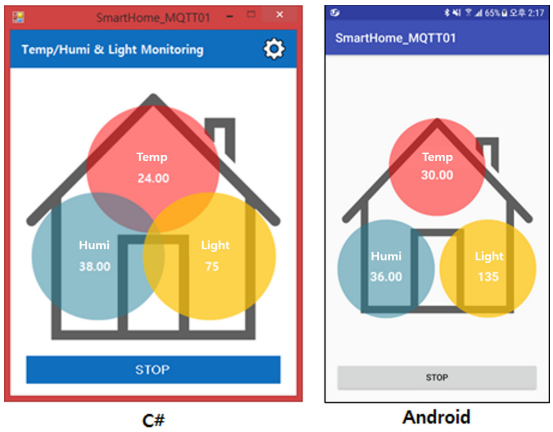

Fig. 11 DIY Smart Home Application Sample

그림 10은 아두이노에 온/습도 및 조도 센서를 부착한 IoT 디바이스로써 스마트 홈 모니터링을 위해 센서로 부터 측정 환경 정보를 지정 토픽을 통해 IoT 클라우드 서버로 발행하며 서비스 서버에는 해당 토픽에 대한 구독한 애플리케이션으로 메시지를 전송한다.

그림 11은 안드로이드와 윈도우(C#) 환경에서 동작하고 있는 스마트 홈 모니터링 애플리케이션의 화면으로 사용자 인증 후 특정 Topic을 통해 IoT 클라우드 서버로 구독을 요청하면 스마트 홈 IoT 디바이스에서 발행되는 환경 정보를 수신하며 이를 화면에 출력하는 것을 확인할 수 있다.

Ⅴ. 결 론

본 논문에서는 관련 전문 개발자가 아닌 일반 사용자의 관점에서 보다 손쉽게 활용할 수 있는 개방형 IoT 경량화 플랫폼을 설계하여 구현하였다.

MQTT 프로토콜을 활용하여 DIY IoT 디바이스와 발행과 구독을 통해 상호간 효율적이고 최적화된 데이터 교환을 위해 MQTT 브로커를 서버 내부에 구성하고, 웹 소켓 기반의 서비스 웹 서버를 통해 애플리케이션으로 알림 및 데이터를 실시간으로 전송하는 MQTT 클라우드 서버를 설계·구현하였다.

본 논문에서 제안한 MQTT 클라우드 서버는 oneM2M 국제 표준 기반의 개방형 사물인터넷 환경에 적합한 통신 프로토콜을 활용함에 따라 사용자가 제작한 DIY 디바이스를 통해 실제 사물인터넷 서비스를 개발·사용할 수 있으며 이를 통해 메이커 운동에 대한 사용자 인식 변화 및 활성화에 도움이 될 것이라 기대한다. 그러나 본 논문에서 구현된 개방형 IoT 경량화 플랫폼의 디바이스는 아두이노를 기반으로 하였으며, 애플리케이션은 안드로이드, C#으로 개발하여 개발환경이 다소 한정적이라 할 수 있다. 통신 프로토콜 또한 MQTT와 웹 소켓만을 사용하여 oneM2M 표준 플랫폼에서 제공하는 다양한 통신 프로토콜을 지원하지 못하였다.

추후 라즈베리파이, 비글본블랙 등 디바이스 제작을 위한 하드웨어 플랫폼과 XMPP, COAP 등 프로토콜 적용 범위를 단계적으로 늘려 실증서비스를 제공할 수 있도록 지속적인 연구개발을 진행할 것이다.

REFERENCES

[1] C. Frolund, J. H. Jeon, Y. K. Lee, S. Y. Lee, and H. J. Kim, “Study for strengthening the ICT DIY ecosystem,” *Information and Communication Technology Convergence (ICTC)*, pp. 269-274, Oct. 2014.

[2] H. Shen, Z. Li, L. Yu, and C. Qiu, “Efficient Data Collection for Large-Scale Mobile Monitoring Applications,” *IEEE Transactions on Parallel and Distributed Systems*, vol. 25, no. 6, pp. 1424-1436, Apr. 2013.

[3] M. Meddeb, M. Ben Alaya, T. Monteil, A. Dhraief, and K. Drira, “M2M Platform with Autonomic Device Management Service,” *International Workshop on Recent Advances on Machine-to-Machine Communication*, vol. 32, pp. 1063-1070, 2014.

[4] S. K. Datta, C. Bonnet, and N. Nikaein, “An IoT gateway centric architecture to provide novel M2M services,” *Internet of Things (WF-IoT), 2014 IEEE World Forum on*, pp. 514-519, Mar. 2014.

[5] K. Tang, Y. Wang, H. Liu, Y. Sheng, X. Wang, and Z. Wei, “Design and implementation of push notification system based on the MQTT protocol,” *2013 International Conference on Information Science and Computer Applications (ISCA 2013)*, pp. 116-119, 2013.

[6] Z. Suryady, G. Sinniah, S. Haseeb, M. Siddique, and M. Ezani "Rapid development of smart parking system with cloud-based platforms," in *The 5th International Conference on Information and Communication Technology for The Muslim World (ICT4M)*, Kuching Malaysia, Nov. 2014.

[7] P. Bellavista, A. Corradi, and A. Reale, “Quality of Service in Wide Scale Publish-Subscribe Systems,” *IEEE Communications Surveys & Tutorials*, vol. 16, no. 3, Apr. 2014.

[8] N. Witthayawiroj, and P. Nilaphruek, “The Development of Smart Home System for Controlling and Monitoring Energy Consumption using WebSocket Protocol,” in *IOP Conference Series: Materials Science and Engineering*, vol. 185, 2017.

[9] Y. Zhangling, and D. Mao, “A Real-Time Group Communication Architecture Based on WebSocket,” *International Journal of Computer and Communication Engineering*, vol. 1, no. 4, pp. 408-411, Nov. 2012.

[10] Y. H. Jang, J. S. Shim, and S. C. 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.

명상일(Sang-Il Myong)

2017년 동명대학교 전기공학과 공학박사수료
1998년 ~ 현재 코리아시스템 대표
※관심분야 : 지능제어, 신호처리, 로봇공학

김성대(Sung-Dae Kim)

1984년 동아대학교 물리학과 졸업(이학사)
1986년 동아대학교 대학원 전자공학과 졸업(공학석사)
1996년 동아대학교 대학원 전자공학과 졸업(공학박사)
1991년 ~ 현재 동명대학교 전기공학과 부교수
※ 관심분야 : 자동제어, 인공지능