

FCM을 이용한 라즈베리파이 기반의 센서정보 알림 구현

오세진

경운대학교 항공공과대학 항공소프트웨어공학부 교수

Implementation of Sensors Information Alarm Service using an FCM based on Raspberry Pi

Sejin Oh

Professor, Dept. of Aeronautical Software Engineering, College of Aeronautical Engineering, Kyungwoon University

요약 사물인터넷(IoT)은 4차 산업혁명의 핵심적인 기술이다. 사물인터넷은 다양한 센서로부터 획득된 정보를 사용자에게 의미 있는 정보로 재가공 및 정보 전달하는 것을 의미한다. 이러한 정보는 WIFI, 블루투스, 서버를 이용하여 센서의 정보를 획득한다. 근거리 네트워크로 구성된 사물인터넷 시스템은 외부의 사용자가 접근하기에는 어려움이 존재한다. 이러한 이유로 사물인터넷 시스템의 정보를 외부에 있는 사용자에게 알림을 전달하는 데 어려움이 있다. 본 논문에서는 라즈베리파이 기반의 사용자 알림 서비스를 구현하였다. 구글의 FCM을 이용하여 라즈베리파이에서 제어되는 온·습도 센서의 정보를 안드로이드 디바이스로 알림을 통해 전달한다.

키워드 : 사물인터넷, 라즈베리파이, 클라우드 메시징, 알림 서비스, 안드로이드

Abstract The Internet of Things(IoT) is one of the key technologies in the Fourth Industrial Revolution. The IoT is a system that acquires information from various sensors and provides meaningful information to users. The method of obtaining information from sensor is using WIFI, Bluetooth and Server. is not accessible to external users because of different type of networks or local area communication. For this reason, there is a problem that external user cannot receive notification in regard to sensor information. In this paper, we want to establish a cloud message environment using Google's FCM(Firebase Cloud Messaging) and find out through experiments how users can receive notifications even if they are outside.

Key Words : IoT, Raspberry Pi, Cloud Messaging, Alarm Service, Android

1. 서론

4차 산업혁명은 인공지능(AI), 사물인터넷(IoT), 휴머노이드 로봇, 드론, 자율주행, 가상현실(VR/AR) 등이 핵심인 차세대 산업혁명을 말한다. 인공지능은 인간의 인식 판단, 문제해결, 학습 기능과 같은 인간의 두뇌에서 처리하는 것과 같이 컴퓨터가 스스로 추론, 학습, 판단하고 특정 작업을 처리하는 시스템을 가리킨다. 사물인터넷은 일상생활 속의 모든 사물에 유무선 네트워크

장비를 연결해서 다양한 정보를 사용자에게 공유 및 제공하는 환경을 말한다. 즉, 각종 사물에 통신 기능을 내장해 인터넷에 연결되도록 해 사람과 사물, 사물과 사람 간의 인터넷 기반 상호 작용을 이루는 것이다. 이를 통해 가전제품과 전자기기를 연결할 수 있고 다양한 하드웨어의 정보를 수집 및 재가공하여 사용자에게 새로운 형태로 서비스를 제공할 수 있다[1, 2].

사물인터넷과 인공지능은 스마트 센서 시장을 크게

*Corresponding Author : Sejin Oh(sjoh@ikw.ac.kr)

증폭시키는 계기가 되었다. 이는 광범위한 센서를 통해서만이 사물인터넷 구축이 가능하며, 수집된 빅데이터가 있어야 한다. 센서는 압력, 온도, 가속도, 생체신호 등 정보를 감지해 전기적 신호로 변환시켜 주는 장치로 사물인터넷의 핵심요소 중 하나이다. 이러한 센서들의 정보는 특정 시스템에 저장되어 되고 사용자의 스마트폰과 같은 디바이스로 전달되어야 할 것이다.

센서의 정보를 사용자에게 전달하는 방법으로는 WIFI, 블루투스, 웹서버 등과 같은 방법을 활용할 수 있을 것이다. 센서를 제어하기 위해 라즈베리파이와 같은 별도의 시스템이 필요한데 이 경우 시스템상에서 센서의 정보를 확인 및 제어를 할 수 있지만, 모바일 디바이스를 이용하여 사용자에게 센서 정보를 전달하기 위해서는 WIFI나 블루투스 등을 이용하여 네트워크를 구성해야 한다[3]. WIFI와 블루투스의 경우는 사용자의 디바이스가 같은 네트워크에 연결되거나 근거리 통신을 해야 하므로 외부에서는 센서의 정보를 받기 어렵다. 사용자와 센서 시스템이 서로 다른 네트워크를 사용할 때는 웹 서버를 이용해야 하는데, 사용자가 외부에서 센서 시스템에 접근하여 정보를 요청하여야 센서 정보를 응답을 받을 수 있다. 그러므로 센서 시스템에 특정 이벤트가 있을 때, 사용자가 접근하지 않고 알림 정보를 전달받기에는 어려움이 있다[4].

본 논문에서는 센서를 사용하는 시스템의 정보 획득에 있어 사용자가 외부에서도 알림을 받을 수 있는 시스템을 구현하였다. 사용자와 센서 시스템의 네트워크가 다르더라도 센서 정보를 획득할 수 있으며, 사용자의 요청이 없더라도 센서 시스템이 사용자에게 이벤트 메시지를 전달할 수 있도록 구현하였다. 구현에 사용된 시스템은 라즈베리파이 기반에서 구글의 FCM을 이용하여 특정 이벤트를 사용자의 안드로이드 폰으로 알림으로 전달한다.

2. 관련연구

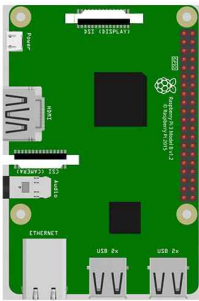
2.1 라즈베리파이

라즈베리파이(Raspberry Pi)는 영국의 라즈베리파이 재단에서 만든 초소형 저가 컴퓨터로서 한 장의 PCB(Printed Circuit Board) 보드 위에 중앙처리장치(CPU), 그래픽처리장치(VGA), 메모리, 입출력장치를 모두 장착한 작은 컴퓨터를 뜻한다. 한 장의 작은 PCB

보드 위에 컴퓨터를 구성하기 때문에 유지보수 및 관리 측면에서 장점이 존재한다. 또한 SoC(System on Chip) 기술을 사용하기 때문에 소형이고 저전력의 장점이 있으며 가격 대비 성능이 뛰어나다[5, 6].

라즈베리파이는 라즈비안, Ubuntu, Windows 10 IoT Core, OSMC, LibreELEC, Mozilla Web Things, PiNet, RISC OS 등 다양한 운영체제를 제공하고 있다. 다양한 운영체제로 인해 물리적 컴퓨팅, 미디어 센터, 게임기 등 다양하게 활용할 수 있다.

라즈베리파이는 다양한 센서와 같은 장치들을 사용할 수 있는데 이때 사용되는 것이 GPIO(General Purpose Input Output)이다. GPIO는 핀으로 제공되며, 전원을 포함한 총 40개의 핀으로 구성되어있어 사용자가 입력과 출력을 자유자재로 사용할 수 있다. GPIO는 사용자가 임의로 입력과 출력을 정할 수 있어 다양한 센서들을 연결하여 하드웨어 제어가 가능하다. GPIO를 제어하기 위해서는 GPIO 라이브러리가 필요하다. GPIO 라이브러리는 Adafruit, WiringPi, RPi.GPIO, WebIOPi 등 종류가 매우 다양하다[7, 8]. Fig. 1은 라즈베리파이와 GPIO 핀을 나타낸 것이다.



Pin#	NAME	NAME	Pin#
01	3.3v DC Power	DC Power 5v	02
03	GPIO2 (SDA1, I2C)	DC Power 5v	04
05	GPIO3 (SCL1, I2C)	Ground	06
07	GPIO4 (GPIO_GCLK)	(TXD0) GPIO14	08
09	Ground	(RXD0) GPIO15	10
11	GPIO17 (GPIO_GEN0)	(GPIO_GEN1) GPIO18	12
13	GPIO27 (GPIO_GEN2)	Ground	14
15	GPIO22 (GPIO_GEN3)	(GPIO_GEN4) GPIO23	16
17	3.3v DC Power	(GPIO_GEN5) GPIO24	18
19	GPIO10 (SPL_MOSI)	Ground	20
21	GPIO9 (SPL_MISO)	(GPIO_GEN6) GPIO25	22
23	GPIO11 (SPL_CLK)	(SPL_CEO_N) GPIO8	24
25	Ground	(SPL_CEL_N) GPIO7	26
27	ID_SD (I2C ID EEPROM)	(I2C ID EEPROM) ID_SC	28
29	GPIO5	Ground	30
31	GPIO6	GPIO12	32
33	GPIO13	Ground	34
35	GPIO19	GPIO16	36
37	GPIO26	GPIO20	38
39	Ground	GPIO21	40

Fig. 1. Raspberry Pi GPIO Pins

2.2 Firebase Cloud Messaging

센서의 정보를 사용자에게 제공하기 위해서는 WIFI, 블루투스와 같은 방법을 활용해서 사용자의 디바이스로 정보를 받을 수 있다. 이 경우는 사용자 디바이스가 센서 시스템과 같은 네트워크에 연결되거나 근거리 통신을 해야 한다. 하지만 사물인터넷의 다양한 센서 정보들은 사용자가 같은 네트워크가 아니더라도 외부에서 정보를 받을 수 있어야 한다[9]. 이러한 경우는 웹 서버를 구축하여 사용자가 센서의 정보를 획득할 수 있

다. 하지만 웹 서버와 클라이언트의 프로토콜은 클라이언트가 서버에게 요청(Request)하면 응답(Response)하는 구조이기 때문이다. 즉, 서버가 사용자에게 이벤트 또는 메시지를 먼저 보낼 수 없다[10, 11]. Fig. 2는 서버와 클라이언트 메시지 전달 구조이다.

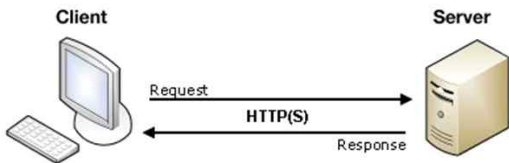


Fig. 2. Basic HTTP Client/Server Communication

FCM(Firebase Cloud Messaging)은 구글의 GCM(Google Cloud Messaging)의 새 버전을 의미한다. Firebase가 지원하는 다양한 기능 가운데 서버가 사용자에게 이벤트를 전달할 수 있게 해준다. FCM은 안드로이드, iOS, 모바일 웹 등 다양한 플랫폼을 지원한다[12]. Fig. 3은 FCM의 구조와 메시지 전달과정을 그림으로 나타낸 것이다.

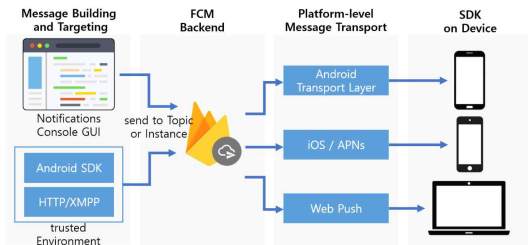


Fig. 3. FCM Architectural Overview

FCM의 과정은 먼저 클라이언트 애플리케이션은 FCM에 연결하여 Sender ID, API Key, Application ID, API Key를 FCM에 전달하여 등록한다. 이후 FCM은 클라이언트 애플리케이션에 등록 토큰을 반환받는다. 마지막으로 클라이언트 애플리케이션이 등록 토큰을 애플리케이션 서버에 전달한다[13]. Fig. 4는 FCM의 등록 과정을 나타낸 것이다.

FCM 등록 이후 사용자의 디바이스로 전달한다. 애플리케이션 서버에서 FCM으로 메시지를 보낸다. 그리고 FCM 서버는 메시지를 큐에 저장한다. 메시지를 큐에 저장하는 과정에서 필요에 따라 메시지를 저장할 수 있다. 이후 클라이언트 애플리케이션에서 FCM 메시지

를 수신하고, 사용자에게 화면에 메시지를 표시하게 된다[14]. Fig. 5는 사용자 디바이스로 이벤트 메시지 전달과정을 나타낸 것이다.

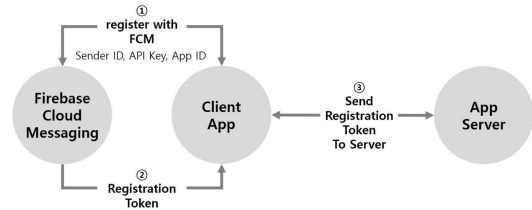


Fig. 4. Registration with FCM

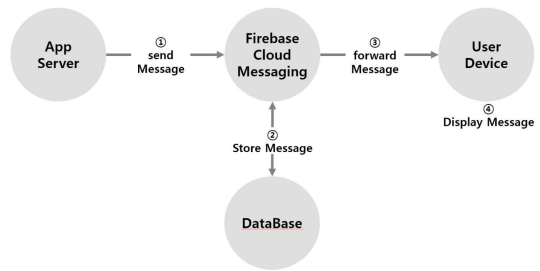


Fig. 5. Downstream Messaging

3. 센서정보 사용자 알림 서비스 구현

본 장에서는 사용자와 센서 시스템이 같은 네트워크가 아니며, 사용자가 먼저 센서 시스템에 접근하지 않더라도 이벤트 메시지를 전달받을 수 있는 센서 정보 사용자 알림 서비스를 구현하고자 한다.

3.1 시스템 구성

앞서 언급한 바와 같이 근거리 네트워크로 구성된 사물인터넷 시스템상에서 사용자 접근의 어려움과 항상 사용자가 먼저 시스템에 접근해야 센서 정보를 획득하는 문제점을 개선하기 위해 FCM을 이용하고자 한다. 본 절에서는 라즈베리파이에 구성된 센서의 정보를 구글의 FCM을 통해 근거리 네트워크 유무에 상관없이 사용자가 접근하지 않더라도 시스템이 능동적으로 정보를 제공하는 시스템을 구현하였다.

구현된 시스템은 사용자의 디바이스와 센서가 연결된 시스템이 같은 네트워크가 아니더라도 FCM을 통해 센서의 정보를 전달받을 수 있다. 또한, Server-Client 특성상 사용자 요청에 대한 서버 응답을 받는 구조가

아닌 서버가 사용자에게 정보를 전달할 수 있는 구조를 가지도록 구현하였다.

FCM의 App Server의 역할을 라즈베리파이를 사용하였다. 라즈베리파이에는 DHT11 센서를 연결하여 파이썬으로 제어하며, 특정 이벤트 발생 시 라즈베리파이에서 FCM으로 이벤트 메시지를 전송한다. FCM은 전달받은 이벤트 메시지를 사전에 등록된 사용자 디바이스로 메시지를 전달한다. FCM의 이벤트 메시지를 전달받은 사용자는 디바이스에 알람이 나타나게 된다. Fig. 6은 본 논문에서 구현한 시스템 구성을 그림으로 나타낸 것이다.

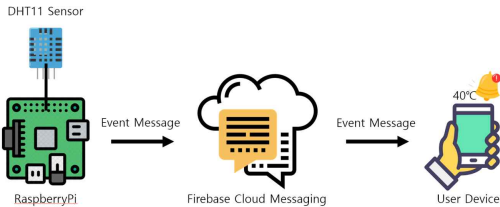


Fig. 6. System Architecture of Alarm Service

3.2 GPIO를 이용한 센서정보 획득

GPIO(General Purpose Input Output)는 마이크로프로세서가 주변장치와 통신하기 위해 범용으로 사용되는 입출력 포트이다. 입력이나 출력을 포함한 동작이 런타임 시에 사용자에게 의해 제어할 수 있으며, 입력과 출력을 마음대로 선택할 수 있고 Low(0)와 High(1)의 출력 신호를 임의로 만들어줄 수 있는 구조를 가진다. 라즈베리파이의 GPIO는 사용자가 임의로 입력과 출력을 정할 수 있어 다양한 센서들을 연결하여 하드웨어 제어가 가능하다.

온습도 센서는 온도와 습도를 측정할 수 있는 센서 두 개를 가지고 있다.

본 절에서 사용된 온습도 센서는 DHT11이며, VCC, DATA, GND 총 3개의 핀이 있다. VCC는 3V, DATA는 GPIO 4번 핀에 연결하였다. 동작을 위한 프로그램은 파이썬을 이용하여 작성하였고, DHT11를 위한 라이브러리는 Adafruit_DHT를 사용하였다. Fig. 7은 라즈베리파이에 DHT11 센서를 연결한 모습이고, Fig. 8은 DHT11의 온도와 습도 값을 획득하는 모습을 나타낸 것이다.

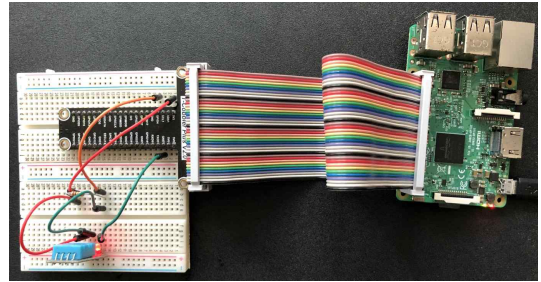


Fig. 7. Setting up a DHT11 on the Raspberry Pi

```

pi@raspberrypi: ~/gpio
pi@raspberrypi:~/gpio $ sudo python dht11.py
Temperature = 26.0°C Humidity = 34.0%
Temperature = 26.0°C Humidity = 34.0%
Temperature = 26.0°C Humidity = 34.0%
Temperature = 25.0°C Humidity = 34.0%
Temperature = 26.0°C Humidity = 34.0%
Temperature = 26.0°C Humidity = 33.0%
  
```

Fig. 8. To read Temperature and Humidity values

3.3 FCM 프로젝트 생성

본 절에서는 구글의 GCP(Google Cloud Platform) 내 FCM을 이용한 라즈베리파이 기반 사용자 알림 서비스를 구현한다.

사용자 알림 서비스를 구현하기 위해서는 FCM 프로젝트를 생성하여야 한다. FCM 프로젝트는 구글 계정이 필요하며, Firebase에 접속한 후 프로젝트를 추가한다. Fig. 9는 FCM 프로젝트 추가 화면이다.

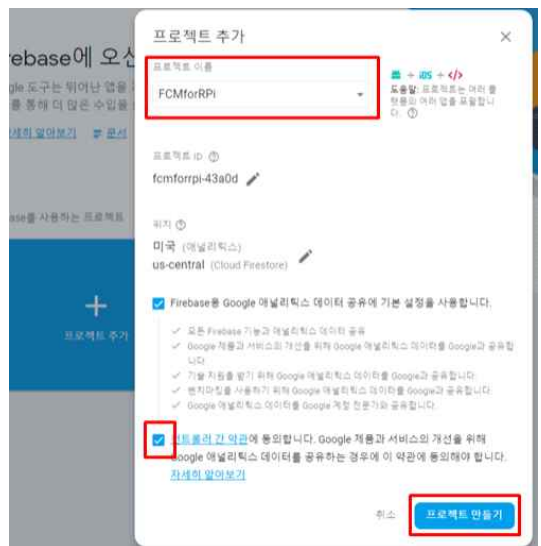


Fig. 9. To create a Firebase Project

FCM 프로젝트를 생성하였다면 FCM 설정, 안드로이드 애플리케이션 작성, FCM 메시지 전달과정을 진행한다. FCM 설정은 안드로이드 애플리케이션 패키지 명을 작성하고 google-service.json 파일은 다운로드하여 Google 서비스 플러그 정보를 확인하고 검증과정을 수행한다. Fig. 10은 google-service.json 파일 다운로드 및 설정하는 화면이다.



Fig. 10. To get FCM Credentials

안드로이드 애플리케이션은 안드로이드 스튜디오를 사용하여 애플리케이션을 생성하고 작성한다. 이때 생성하는 프로젝트 이름은 Firebase 프로젝트 생성 시 사용된 패키지 명이므로 Firebase 패키지 명과 동일하게 생성한다. 생성된 안드로이드 프로젝트는 FCM의 FCM 메시지를 수신할 수 있는 애플리케이션을 작성한다. 생성 클래스는 FirebaseInstanceIdService와 FirebaseMessagingService를 작성한다.

FCM 메시지 전송은 Firebase Cloud Messaging에서 "Send your first message"로 가능하다. Fig. 11은 메시지 전송화면이다.

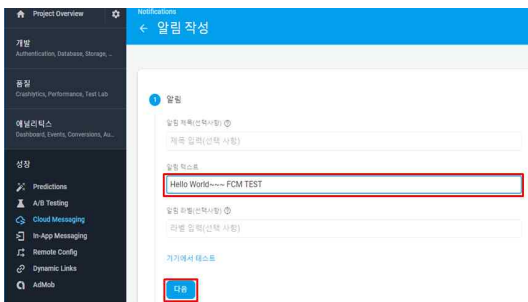


Fig. 11. To send messaging in FCM

라즈베리파이에서 FCM 서버로 메시지를 게시하고 FCM에 게시된 메시지를 스마트폰으로 알림 처리하는 과정은 FCM 클라우드 메시징 서버 키와 안드로이드 애플리케이션의 토큰이 필요하다. 준비된 서버 키와 토큰을 활용하여 라즈베리파이에서 메시지 전송 시 FCM을 통하여 안드로이드 폰으로 메시지 수신이 가능하다. Fig. 12은 FCM의 서버 키, Fig. 13는 안드로이드 스튜디오 상의 디바이스 토큰 키, Fig. 14는 FCM에서 메시지 전송 시 사용자 디바이스에서 메시지를 수신한 화면을 나타낸 것이다.



Fig. 12. The Server Key of Cloud Messaging API



Fig. 13. To generates Device Token Key on Android

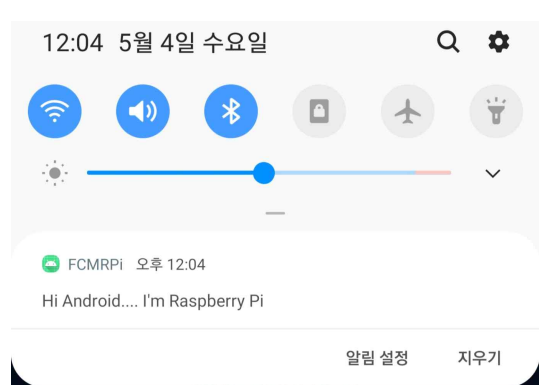


Fig. 14. The Result of Cloud Messaging

3.4 라즈베리파이 센서와 FCM 연동

앞 절에서 FCM과 사용자 디바이스와의 메시지 송수신 준비가 되었으니 본 절에서는 앞서 GPIO를 이용하여 획득한 온도와 습도 정보를 FCM을 이용하여 사용자의 휴대전화로 전달하는 과정을 다룬다.

파이썬 상에서 FCM을 사용하기 위해서는 PYFCM(Python Client for FCM) 패키지가 필요하다. 라즈베리파이에서 pip 명령을 통해 설치할 수 있다. PYFCM이 설치가 되었다면 FCM에 접근하기 위한 API_KEY와 휴대전화 PushToken을 이용하여 전달할 수 있다. API_KEY와 PushToken은 앞 절의 클라우드 메시징 서버 키와 안드로이드 토큰 키이다.

이제 온습도 정보를 사용자의 디바이스로 전달할 준비는 다 되었다. 앞서 작성된 DHT11 코드에 PYFCM 코드를 추가하여 온도와 습도 정보를 FCM으로 전달하게 되면 Fig. 15와 같이 사용자의 휴대전화로 알림을 나타내게 된다.

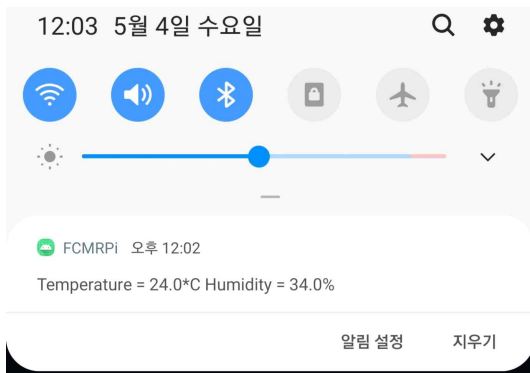


Fig. 15. The Sensors Information using FCM

4. 결론

최근 4차 산업혁명과 관련하여 사물인터넷은 여러 기술 등 핵심적인 기술로 주목받고 있다. 사물은 인터넷은 다양한 센서 디바이스의 센서 정보를 획득하고 사용자에게 다양한 형태로 데이터를 재가공하여 의미 있는 정보로 제공되고 있다. 그 가운데 데이터의 정보를 획득 받는 구조 중 WIFI, 블루투스의 경우 같은 네트워크이거나 근거리 통신으로 이루어짐에 따라 외부에서 정보를 획득하기 위한 연구가 활발히 진행되었다. 기존의 서버를 구축한 방법은 클라이언트가 먼저 서버 측에 접근하여 정보를 요청하여야 응답받을 수 있는 구조로 인해 사물인터넷 상에서 요청 없이도 응답받는 방법이 필요하다.

본 논문에서는 라즈베리파이와 센서 장치가 구성된 환경에서 구글의 FCM(Firebase Cloud Messaging)을 이용하여 안드로이드 사용자에게 알림 메시지를 전달하는 방법에 대해 실험을 통해 확인 및 구현하였다.

이는 FCM을 이용하게 되면, 센서 시스템과 사용자가 같은 네트워크가 아니더라도 접근할 수 있으며, 사용자가 요청 없이도 센서 시스템의 이벤트를 바로 받을 수 있다. 아울러 FCM 내부적으로 데이터베이스를 가지고 있어 센서 시스템에 별도의 데이터베이스를 구축하지 않아도 되는 장점이 있다. 클라우드 상의 데이터베이스에 센서 정보를 저장하여 사용자가 실시간으로 확인 가능하며, 다양한 형태로 서비스를 제공할 수 있는 장점이 있다.

매년 화재, 도난 등의 사고가 끊이지 않는 상황에서 사물인터넷 서비스는 사고를 예방하거나 초동조치에 큰 역할을 하는 실정이다. 건물 내외부에 화염 감지 센서, 침입 감지 센서 등을 활용한 시스템에 본 논문에서 사용된 FCM을 이용하여 외부의 사용자가 센서의 정보를 전달받는다면 많은 사고에 대처하는 데 큰 역할을 할 것으로 기대된다. 현재 클라우드 환경이 확산하는 시점에서 FCM은 다양한 형태로 구성 및 활용을 할 수 있을 것으로 예측한다.

REFERENCES

- [1] A. Al-Fuqaha, M. Guizani, M. Mohammadi, M. Aledhari & M. Ayyash. (2015). Internet of Things: A Survey on Enabling Technologies, Protocols, and Applications. *IEEE Communications Surveys & Tutorials*, 17(4), 2347-2376.
- [2] B. C. Kim. (2015). A Internet of Things(IoT) based exploration robot design for remote control and monitoring. *Journal of Digital Convergence*, 13(1), 185-190. DOI : 10.14400/JDC.2015.13.1.185
- [3] J. H. Hyeong, H. S. Jeon, C. H. Shin, M. H. Chang & B. K. An. (2016). Smart LED Push Notification System based on Android. *The Journal of The Institute of Internet, Broadcasting and Communication*, 16(1), 97-102. DOI : 10.7236/JIIBC.2016.16.1.97
- [4] J. H. Noh & H. H. Tack. (2017). The Implementation of the Fine Dust Measuring System based on Internet of Things(IoT). *Journal of the Korea Institute of Information and Communication Engineering*, 21(4), 829-835. DOI : 10.6109/jkiice.2017.21.4.829
- [5] E. Upton & G. Halfacree. (2013). *Raspberry Pi User Guide*. John Wiley & Sons.
- [6] A. Bradbury & B. Everard. (2014). *Learning Python with Raspberry Pi*. John Wiley & Sons.

- [7] H. H. Hadwan & Y. P. Reddy. (2016). Smart Home Control by using Raspberry Pi & Arduino UNO. *International Journal of Advanced Research in Computer and Communication Engineering*, 5(4), 283-288. DOI : 10.17148/IJARCCCE.2016.5473
- [8] S. Liu, L. Guo, H. Webb, X. Ya & X. Chang. (2019). Internet of Things Monitoring System of Modern Eco-Agriculture Based on Cloud Computing. *Data Mining for Internet of Things*, 7, 37050-37058.
- [9] Y. M. Lee & K. R. Sohn. (2015). Fabrication of smart alarm service system using a tiny flame detection sensor based on a Raspberry Pi. *Journal of Advanced Marine Engineering and Technology*, 39(9), 953-958. DOI : 10.5916/jkosme.2015.39.9.953
- [10] Y. W. Cha, J. K. Jeong, C. H. Kim & K. G. Kim. (2016). Research of Event Notification Subsystem Based on Websocket and GCM Push Services. *Journal of Korean Institute of Information Technology*, 14(10), 123-131. DOI : 10.14801/jkiit.2016.14.10.123
- [11] Y. M. Na, D. H. Hyun, D. H. Park, S. H. Hwang & S. H. Lee. (2020). AI Fire Detection & Notification System. *Journal of The Korea Society of Computer and Information*, 25(12), 63-71. DOI : 10.9708/jksoci.2020.25.12.063
- [12] C. Rojas. (2019). Push Notifications. *Building Progressive Web Applications with Vue.js: Reliable, Fast, and Engaging Apps with Vue.js*. 187-203. DOI : 10.1007/978-1-4842-5334-2
- [13] L. Moroney. (2017). Firebase Cloud Messaging. *The Definitive Guide to Firebase*. 163-188. DOI : 10.1007/978-1-4842-2943-9_9
- [14] M. B. Chung. (2022). Guidance system about closure using push notification of smart device. *Journal of Internet Computing and Services*, 23(2), 19-28.
- [15] G. Albertengo, F. G. Debele, W. Hassan & D. Stramandino. (2020). On the performance of web service, google cloud messaging and firebase cloud messaging. *Digital Communications and Networks*, 6(1), 31-37. DOI : 10.1016/j.dcan.2019.02.002

오 세 진(Sejin Oh)

[정회원]



- 2009년 2월 : 경운대학교 컴퓨터 공학과(공학사)
- 2011년 2월 : 경북대학교 전자전기 컴퓨터공학부(공학석사)
- 2014년 8월 : 경북대학교 컴퓨터 학부(공학박사)
- 2017년 3월~현재 : 경운대학교 항공소프트웨어공학부 교수
- 관심분야 : 임베디드 시스템, 사물인터넷, 정보보안
- E-Mail : sjoh@ikw.ac.kr