

EdgeX Foundry 기반의IoT 제어 시스템 구현

권도형*, 임현교*, 한연희**1), 김민석***, 홍용근***

*한국기술교육대학교 창의융합공학협동과정

**한국기술교육대학교 컴퓨터공학과

***한국전자통신연구원

e-mail : {dohk, glenn89, yghan}@koreatech.ac.kr, {mskim16, yghong}@etri.re.kr

Implementation of IoT control system based on EdgeX Foundry

Do-Hyung Kwon*, Hyun-Kyo Lim*, Youn-Hee Han*, Min-suk Kim**, Yong-Geun Hong**

*Korea University of Technology and Education, Republic of Korea

**Electronics and Telecommunications Research Institute, Korea

요 약

본 논문에서는 운영체제나 하드웨어에 종속되지 않고 IoT 환경을 구축할 수 있는 EdgeX Foundry를 이용하여 EdgeX와 두 대의 라즈베리파이 간의 통신을 구현한다. 두 라즈베리파이가 보낸 센서값은 EdgeX를 거쳐 EdgeX Client로 전해지며, 처리된 값은 두 라즈베리파이 중 특정 명령을 수행할 라즈베리파이로 command 메시지가 전송되어 디바이스를 작동시킨다.

1. 서론

사물 인터넷(IoT: Internet of Things)은 모든 사물이 네트워크상에 상호간 연결되어, 센서와 액추에이터간 상호 작용하는 환경을 의미한다 [1]. 각종 IoT 장치는 방대한 양의 데이터를 연속적으로 생성하며, 일반적으로 그러한 대량의 센서 데이터를 저장하고 계산하는 데 원격의 중앙 클라우드 시스템이 사용된다. 하지만, 원격지에 존재하는 클라우드 시스템의 사용은 데이터 저장 및 장치 간의 통합에 많은 어려움이 발생하게 된다 [2]. 이 문제를 극복하기 위해 제안된 Edge Computing 기술은 데이터 수집, 프로세스 및 분석을 원격지의 클라우드 시스템이 아닌 데이터 생성 소스 근처에서 수행하도록 한다. 최근, Edge Computing 기술은 실제 애플리케이션을 사용하는 소비자 중심의 IoT 서비스 및 애플리케이션을 구현하기 위한 중요한 플랫폼이 되고 있다.

최근, IoT 서비스에 대한 수요는 증가하고 있는 추세인 반면, 연결하려는 장치끼리 서로 다른 프로토콜을 사용하는 것으로 인해, 장치 간 호환성이 떨어진다는 문제점이 있다. EdgeX Foundry는 이러한 호환성 문제를 해결하기 위하여 제안된 오픈 프레임워크이자 플랫폼으로서, 특정 하드웨어나 운영체제의 종류에 종속되지 않고서 Edge

Computing 환경을 구성할 수 있게 해준다 [3-4].

2. 본론

여러 대의 라즈베리파이와 별도의 데스크탑을 묶어 하나의 네트워크 환경을 구축하여 통신이 가능하게 하는 방법은 여러 가지가 있다. UDP나 TCP를 이용하여 통신이 이루어지게 할 수도 있으며, IoT 기기들간의 통신 프로토콜로 주목 받는 MQTT 프로토콜을 이용하여 통신이 이루어지게 할 수도 있다.

그러나 라즈베리파이 이외의 CCTV와 같은 디바이스들은 이와 같은 프로토콜을 지원하지 않을 수 있다. 이와 같은 디바이스로부터 데이터를 전송 받고 조작하기 위해서는 기기들끼리 합의된 프로토콜이 있어야 한다. 디바이스의 종류가 다양해질수록 이러한 문제는 더욱 커진다.

EdgeX는 이와 같이 서로 상이한 프로토콜을 가진 여러 대의 디바이스를 하나의 서비스로 구성할 수 있는 프레임워크를 제공해준다. 이를 위하여 EdgeX는 복수의 마이크로 서비스를 하나의 워크플로우로 묶어서 실행할 수 있는 컨테이너 기반의 환경을 제공해준다 [5]. 공식 홈페이지의 가이드에 따르면 12개의 마이크로 서비스들이 기본적으로 제공되며, 12개 중 필요한 마이크로 서비스만을 선택적으로 사용할 수 있다. 또한 개발자가 직접 필요한 마이크로서비스를 만들어 배포하는 것도 가능하다.

EdgeX를 실행시키기 위한 방법은 크게 세 가지로,

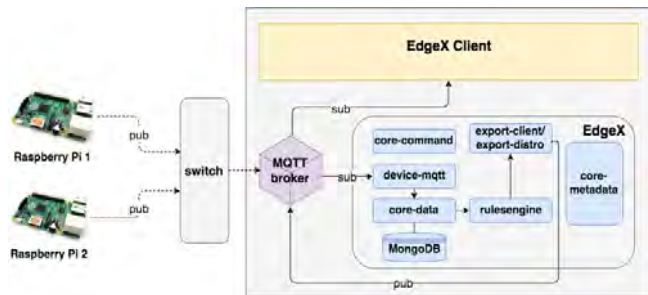
1)교신 저자: 한연희(한국기술교육대학교)

이 논문은 2016년도 및 2018년도 정부(교육부)의 재원으로 한국연구재단의 지원을 받아 수행된 기초연구사업임 (No. NRF-2016R1D1A3B03933355, No.2018R1A6A1A03025526)

User mode, Developer mode, Hybrid mode가 있다. 본 논문에서는 실험과 디버깅을 위하여 Developer mode로 core-data, core-metadata, core-command, export-client, export-distro, rulesengine 그리고 device-mqtt를 구성하며 별도로 mongoDB를 실행한다.

실험 환경 구성은 그림 1과 같다. 먼저 스위치를 통하여 두 대의 라즈베리파이가 EdgeX의 device-mqtt서비스로 센서값을 보낸다. 이 때 보내지는 센서값은 비동기방식으로, 각 디바이스가 독립적으로 센서값을 전송한다.

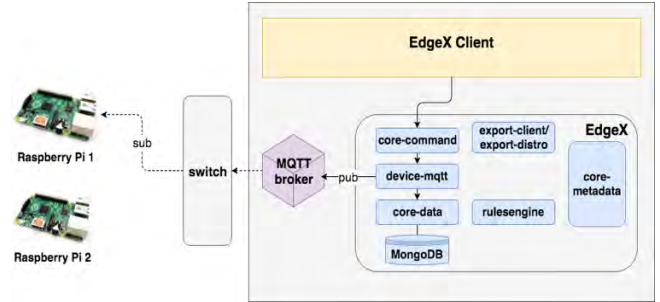
전송은 MQTT방식으로 이루어지며, 값을 보내는 주체인 라즈베리파이가 MQTT broker의 client가 되어, 각자 독립적인 토픽으로 EdgeX에게 publish를 수행한다.



(그림 1) device-mqtt서비스를 이용한 센서값 수신

이 값들은 EdgeX외부에 위치해 있는 EdgeX client가 subscribe하는데, 이 때 device-mqtt 서비스를 거쳐야만 하며, REST API 방식으로 값을 받는다. 따라서 device-mqtt도 MQTT broker의 client가 된다. EdgeX 외부에 위치해 있는 EdgeX client는 EdgeX의 rulesengine서비스와 export-distro서비스를 통해 값을 전송 받는다.

EdgeX의 device-mqtt서비스는 전송 받은 값을 다시 EdgeX client로 전송하며, 값을 처리한다. 처리된 값은 두 라즈베리파이 중 특정 라즈베리파이(이하 라즈베리파이 1번)를 동작 시키는 데에 쓰일 command이다. command는 다시 EdgeX의 device-mqtt서비스를 거쳐 라즈베리파이 1번으로 전송된다. command를 전송할 때, EdgeX의 device-mqtt 대한 호출은 REST API 방식으로 이루어진다. EdgeX의 core-command서비스가 이 과정에 개입하며, 모든 로그와 주고받는 데이터는 core-data서비스에 의하여 mongoDB 데이터베이스에 쌓인다. 센서값을 받은 후, 라즈베리파이 1번에게 command를 내리는 상황은 그림 3과 같다. EdgeX Client에서 라즈베리파이 1번에게 보낼 command는 먼저 EdgeX에게 보내어진 후, EdgeX는 이를 device-mqtt 서비스를 통해 라즈베리파이 1번에게 보낸다.



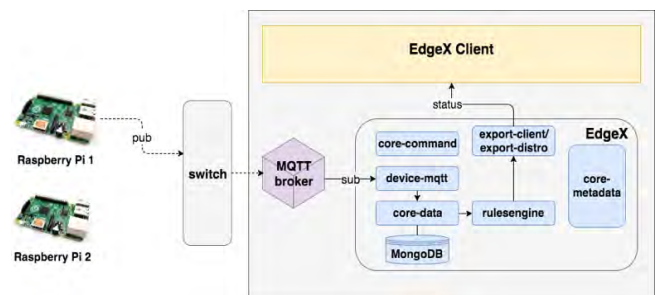
(그림 3) device-mqtt서비스를 이용한 command 전송

EdgeX의 device-mqtt서비스는 command가 내려질 때마다 이에 대한 response를 디바이스로부터 전송 받게 되어있다. response를 받은 모습은 그림 4와 같다. 이때, command마다 unique한 센서 식별자(uuid)가 주어진다.

```
object -
Matching response received for uuid: e37144fa-8c2e-4473-ba8f-b220df69624d
response data for element found - 1
RESULT - 1
object -
Matching response received for uuid: f556dbba-60be-4785-bf1e-876c96004fdc
response data for element found - 1
RESULT - 1
```

(그림 4) EdgeX에서 출력하는 response 메시지

Command 메시지를 수신 받은 후, 이에 대한 response 메시지를 받는 상황은 그림 5와 같다. command 메시지를 수신 받은 라즈베리파이 1번에서 MQTT broker에게 REST API를 통해 response 메시지를 전송하여 EdgeX Client에게 해당 메시지를 보내는 상황을 나타내고 있다.



(그림 5) device-mqtt서비스를 이용한 response 전송

3. 결론

본 논문에서는 EdgeX foundry를 이용하여, 복수의 라즈베리파이로부터 센서값을 받아들이고 후, EdgeX Client에서 값을 처리하여 해당 값을 또 다른 라즈베리파이에 command 메시지로 전송하고 그에 대한 response 메시지를 수신하는 일련의 과정을 구현함을 보였다.

참고문헌

- [1] 김아영, 이준우, “대용량 IoT 데이터 처리를 위한 클라우드 프레임워크 설계,” 한국컴퓨터종합학술대회, 2015.
- [2] Hyunseok Chang, Adishesu Hari, Sarit Mukherjee and T.V. Lakshman, “Bringing the Cloud to the Edge,” IEEE INFOCOM Workshop on Mobile Cloud Computing, 2014.
- [3] “EdgeX Foundry,” <https://www.edgexfoundry.org>
- [4] 이승형, 권진철, 김종원, “EdgeX Open Framework 기반의 IoT-Cloud 대응 서비스의 시험적 구성,” 한국통신학회, 2018.
- [5] 김승룡, 강문중, 김종원, “컨테이너 기반 IoT-Cloud 서비스를 위한 워크플로우를 활용한 자동화된 서비스 합성,” 한국통신학회, 2017.