|  |
| --- |
| EdgeX Foundry 기반의 IoT 제어 시스템 구현 |
|  |

|  |
| --- |
| 권도형\*, 임현교\*, 한연희\*\*, 김민석\*\*\*, 홍용근\*\*\*[[1]](#footnote-2)  \*한국기술교육대학교 창의융합공학협동과정  \*\*한국기술교육대학교 컴퓨터공학과  \*\*\*한국전자통신연구원  e-mail : {dohk, glenn89, yhhan}@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 두 대의 라즈베리파이 간의 통신을 구현하였다. 두 라즈베리파이가 보낸 센서 값은 EdgeX를 거쳐 EdgeX Client로 전해지며, 처리된 값은 두 라즈베리파이 중 특정 명령을 수행할 라즈베리파이로 command 메세지가 전송되어 디바이스를 작동 시키게 된다. |
|  |

1. 서론

사물 인터넷(IoT: Internet of Things) 서비스에 대한 수요는 증가하고 있는 추세인 반면, 연결하려는 장치들끼리 서로 다른 프로토콜을 사용하는 것으로 인해, 장치 간 호환성이 떨어진다는 문제점이 있다. EdgeX Foundry는 이러한 호환성 문제를 해결하기 위하여 제안 된 오픈 프레임워크로서, 특정 하드웨어나 운영체제의 종류에 종속되지 않고서 IoT 서비스 환경을 구성할 수 있게 해준다.[1]

1. 본론

여러 대의 라즈베리파이와 별도의 데스크탑을 묶어 하나의 네트워크 환경을 구축하여 통신이 가능하게 하는 방법은 여러가지가 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 수도 있다. 이와 같은 디바이스들로부터 데이터를 전송 받을 뿐만 아니라 디바이스들을 조작하기 위해서는 기기들끼리 합의된 프로토콜이 있어야 한다. 디바이스의 종류가 다양해질수록 이러한 문제는 더욱 커진다.

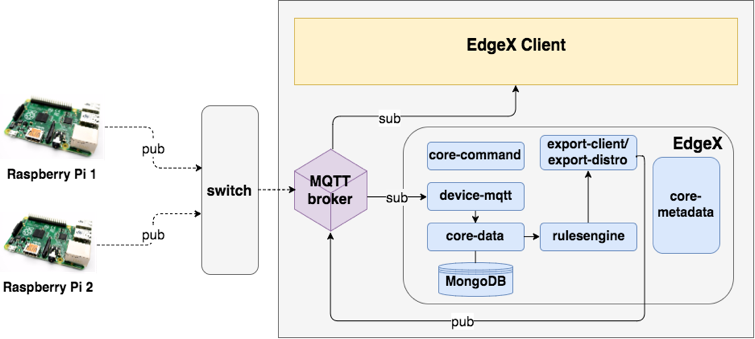
EdgeX는 이와 같이 서로 상이한 프로토콜을 가진 여러 대의 디바이스들을 하나의 서비스로 구성할 수 있는 프레임워크를 제공해준다. 이를 위하여 복수의 마이크로서비스 묶어서 실행할 수 있는 환경을 제공해준다. 공식 홈페이지의 가이드에 따르면 12개의 마이크로서비스들이 기본적으로 제공되며[2], 12개 중 필요한 마이크로서비스만을 선택적으로 사용할 수도 있다. 또한 개발자가 직접 필요한 마이크로서비스를 만들어 배포하는 것도 가능하다.

EdgeX를 실행시키기 위한 방법은 크게 세 가지로, 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 구성하였으며 별도로 mongoDB를 실행하다.

실험 환경 구성은 그림 1과 같다. 먼저 스위치를 통하여 두 대의 라즈베리파이가 EdgeX의 device-mqtt서비스로 센서 값을 보낸다. 이 때 보내어지는 센서 값은 비 동기방식으로서, 각 디바이스가 독립적으로 센서 값을 전송한다.

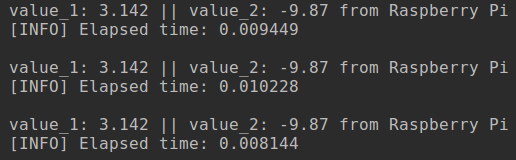
전송은 MQTT방식으로 이루어지며, 값을 보내는 주체인 라즈베리파이가 MQTT broker의 client가 되어, 각각 독립적인 토픽으로 EdgeX에게 publish를 수행한다. 이 값들은 EdgeX외부에 위치해 있는 EdgeX client가 subscribe하는데, 이 때 device-mqtt 서비스를 거쳐야만 하며, REST API 방식을 통해 값을 받게 된다. 따라서 device-mqtt도 MQTT broker의 client가 된다. EdgeX 외부에 위치해 있는 EdgeX client는 EdgeX의 rulesengine서비스와 export-distro서비스를 통해 값을 전송 받을 수 있다.

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.



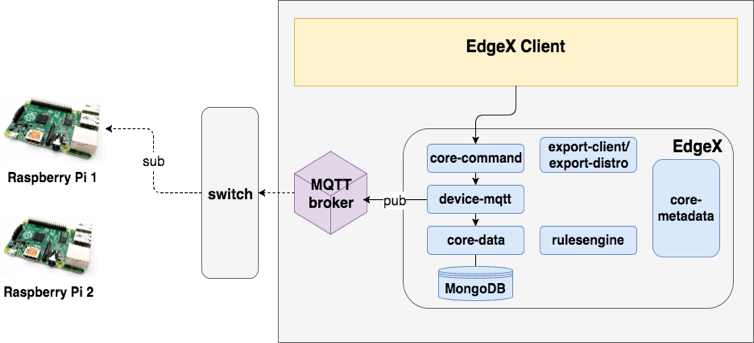
(그림 1) device-mqtt서비스를 이용한 센서 값 수신

EdgeX Client에서 값을 받은 결과는 그림 2와 같다. REST API를 통해 센서 값을 받았으며, 약 0.01초 정도의 time delay가 생기는 것으로 나타난다.



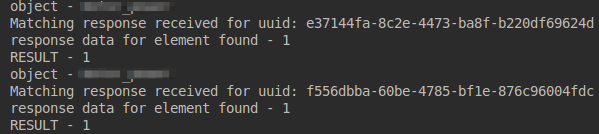
(그림 2) EdgeX Client가 수신 받은 센서 값

센서 값을 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.



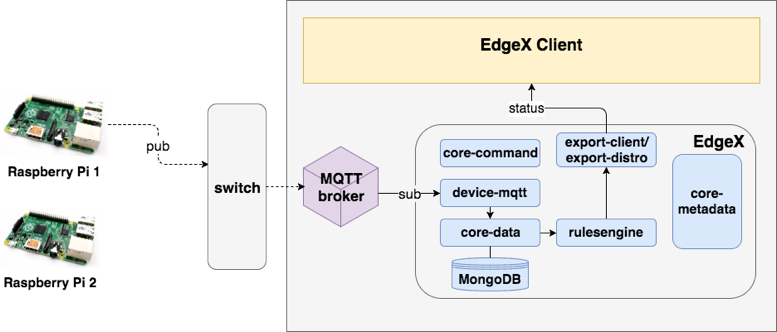
(그림 3) device-mqtt서비스를 이용한 command 전송

EdgeX의 device-mqtt서비스는 command가 내려질 때마다 이에 대한 response를 디바이스로부터 전송 받게 되어있다. response를 받은 모습은 그림 4와 같다. 특이하게도, command마다 unique한 uuid가 주어지고 있는 것을 관찰할 수 있다.



(그림 4) EdgeX에서 출력하는 response 메세지

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



(그림 5) device-mqtt서비스를 이용한 response 전송

1. 결론

본 논문에서는 EdgeX foundry 기반의 마이크로서비스를 이용하여, 복수의 라즈베리파이로부터 센서 값을 받아들인 후, EdgeX Client에서 값을 처리하여 해당 값을 또 다른 라즈베리파이에게 command 메세지로 전송하고 그에 대한 response 메세지를 수신하는 일련의 과정을 구현함을 보였다.

**참고문헌**

[1] 이승형, 권진철, 김종원, “EdgeX Open Framework 기반의 IoT-Cloud 대응 서비스의 시험적 구성,” 한국통신학회, 2018

[2] https://github.com/edgexfoundry

1. 교신 저자: 한연희(한국기술교육대학교)

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

   "이 논문은 2018년도 정부(교육부)의 재원으로 한국연구재단의 지원을 받아 수행된 기초연구사업임(No. 2018R1A6A1A03025526)" [↑](#footnote-ref-2)