

ARTIK 클라우드를 통한 분산된 원격 IoT 센서 데이터 모니터링 및 제어

배점한*, 김종태*

Distributed Remote IoT Sensor Data Monitoring and Control through the ARTIK Cloud

Jum-Han Bae*, Jong Tae Kim*

요약

IoT 시스템은 로컬의 센서들과 인터넷에 연결된 서버 간의 동작으로 이루어진다. 센서가 서로 다른 도시에 위치하고 있고, 이것을 원격으로 모니터링 하고 제어 하며, 동작하는 모드에 따라 제3자의 클라우드 서버를 호출하여 활용 할 수 있다. 이런 동작은 IoT로서의 모든 시나리오를 포함하는 것이다. 본 논문에서는 ARTIK 하드웨어와 ARTIK 클라우드를 사용하여 예상 할 수 있는 중요한 경우의 IoT 시스템을 모두 다룬 튜토리얼 논문이다. WiFi가 연결된 환경 내에서는, 인터넷을 통해서 거리의 제한이 없어진다. 그리고 IoT 시스템은 기능적으로 비교적 단순하지만 센스 층, 허브 층, 클라우드 층을 가지는 수직적인 구조 이다. 이렇게 IoT 층간을 잘 설계하기 위해서는, 층간을 잘 이해 해야만 비로소 설계가 가능하다. 본 논문에서는 node-red라는 GUI 방식의 언어를 사용하여 층간 통신의 어려움을 용이하게 극복 하였다. 또한 본 논문에서 사용한 방법은, 양산적용이 용이하고 IoT 본연의 기능 연구에 집중 할 수 있다. 따라서 최적의 IoT 설계 방법이라 할 수 있다.

Key Words : Artik Cloud, IoT, Remote Device, Local Device, Node-Red

ABSTRACT

The IoT system consists of operations between local sensors and servers connected to the Internet. The sensors are located in different cities, which can be monitored and controlled remotely, and a third-party cloud server can be called and utilized according to the operating mode. This operation includes all scenarios as IoT. This paper is a tutorial paper on all of the important cases of IoT systems that can be expected using ARTIK hardware and the ARTIK cloud. In the environment where WiFi is connected, the distance limitation is eliminated through the Internet. The IoT system is relatively simple in function but has a vertical structure with a sense layer, a hub layer, and a cloud layer. In order to design the IoT interlayer well, it is only necessary to understand the interlayer to design it. In this paper, we use GUI-based language called node-red to overcome the difficulties of inter-layer communication easily. Also, the method used in this paper can be easily applied in mass production and can focus on the function research of IoT. Therefore, this is the optimal IoT design method.

* First Author : (ORCID:0000-0002-4865-3356) Soongsil University, Department of IT Convergence, jhabe17@ssu.ac.kr,정회원
/ Electronic, Electrical and Computer Eng., PhD Course, Sungkyunkwan Univ., Republic of Korea

* Sungkyunkwan Univ

논문번호 : 201810-327-B-RN, Received October 16, 2018; Revised November 3, 2018; Accepted November 5, 2018

I. 서 론

IoT (Internet of Things) 시스템은 사물과 인터넷이 연결된 융합의 영역으로 여러 가지 응용이 있을 수 있다. 그 중에서도 가정 내 기기를 지능적으로 제어하기 위한 스마트 홈이나 작업장 내에 센서를 배치하여 공장을 효율적으로 제어하기 위한 스마트 팩토리 등의 응용에서 많은 발전이 있었다. IoT의 일반적인 데이터 흐름은, 센서를 통해서 입력을 받고 클라우드 데이터 센터에서 센서 데이터를 가공하며, 가공된 결과인 제어 데이터는 클라이언트 하드웨어에 액션 (Action)을 전달하는 형태이다.

스마트 홈은 통신사를 통한 AI(Artificial Intelligence) 스피커가 중심으로 되어 가전을 제어하는 기능들이 모여 있다. 스마트 팩토리는 반드시 표준을 따르지 않아도 되는 영역이다. 생산성 향상은 바로 이익과 직결되기 때문에 다양한 연구들이 진행되고 있다. 스마트 팩토리를 실현하기 위해서, 설비들에 장착되는 제조용 IoT 기기 및 각 기기간의 통신을 실증적으로 제시하고 구현한 연구가 있다¹⁾. 하드웨어는 Intel사의 IoT 플랫폼 중 하나인 Edison을 기반으로 하여 임베디드 시스템을 확장하고 여기에 새로운 통신을 구현하기 위한 방식을 제시하였다²⁾. 또한 제조업에서 공정의 불량률을 줄이고 생산량을 증가시키기 위해서는 공정시스템이 자동화되고 실시간으로 정보들을 수집하여 공정마다 최적화된 의사결정을 내릴 수 있어야한다. 이를 위한 실시간 모니터링 시스템을 위해 분산형 구조 기반의 임베디드 네트워킹 시스템을 제안한 논문도 소개 되었다³⁾. 설치장소나 공간에 제약이 없도록, 소형, 저비용 및 친환경의 대기환경 측정 장치로 구성되는 IoT 기반의 대기환경 측정 및 분석시스템이 소개 되었다⁴⁾.

IoT는 로컬영역에 있는 클라이언트 하드웨어 장치와 인터넷으로 연결된 클라우드 서버와 연결되어, 일체화된 시스템으로 연동하여 동작하여야 한다. 이렇게 하기 위해서는 클라이언트와 서버간의 네트워크 통신을 정의하고 구현해야 한다. 이것을 위해서는 방대한 양의 소프트웨어를 개발하여야 하여야 한다. 이런 요소는 IoT 시스템을 구축하는데 큰 장애요소가 되고 있다. 인터넷 서버를 설계하기 위해서는 자바스크립트 언어에서 파생한 서버용 언어인 Node.js로 설계할 수 있다. Node.js의 GUI(Graphical User Interface) 형태의 언어인 Node-red로 기술하여 소프트웨어 설계의 부담을 대폭 경감한 방법이 소개 되었다⁵⁾. Node-red는 IBM에서 제안하였고, 아직은 1.0 버전이

아닌 0.18버전까지 나와 있다.

IoT 기술은 디바이스, 네트워크, 플랫폼, 서비스를 모두 아우르는 융합기술이다. 기존에는 개별 서비스 분야 별로 별도의 플랫폼을 구축하고 서비스하는 방식으로서, 새로운 서비스를 제공하기 위해서는 별도의 플랫폼을 구축해야 하는 문제를 가지고 있다⁶⁾. 이러한 수직적으로 파편화된 플랫폼의 문제점을 극복하고 IoT 생태계 활성화를 위하여 개방형 구조의 수평적으로 통합된 플랫폼이 필요하다고 주장하는 논문이 소개되었다⁶⁾.

본 논문에서는 IoT전용의 서버인 삼성의 아티클라우드(ARTIK Cloud)와 세미커스텀 (Semi- custom) 하드웨어인 아티 보드(ARTIK Board) 그리고 인터넷 서버를 설계하기 위해 만들어진 Node.js 언어의 GUI 방식 언어인 Node-red를 이용해서 IoT 시스템을 설계하는 기술을 소개한다. 삼성 아티 플랫폼을 기준으로 한 이유는 이 플랫폼의 모든 구성요소가 양산이 가능하도록 설계되어 있다. 교육용으로만 만들어진 것은 모든 사용자 경우의 수와 모든 동작환경 조건을 충족하지 못한다. 따라서 교육과 양산이 모두 가능한 플랫폼을 기준으로 연구하였다. 지금까지의 많은 논문들은 아티 IoT 클라우드의 역할에 해당하는 통신과 인터넷 서버 관련된 부분에 많은 할당을 하고 있다⁷⁾⁸⁾. 기존의 논문들은 한 지역에 있는 IoT 디바이스를 원격에서 모니터링 하는 반면¹⁾²⁾⁴⁾ 본 논문에서는 로컬과 다수의 원격지역의 디바이스를 통합하고 모니터링 하는 기술을 소개 한다. 또한 다른 기능을 필요로 하는 경우는 제3자 클라우드를 호출하여 통합된 동작을 하도록 한다. GUI방식 언어인 Node-red를 이용해서 클라이언트와 서버 간을 설계한다.

II. 본 론

IoT 시스템은 센서 소자에서부터 인터넷 클라우드 서버까지의 많은 구성요소를 포함하고 있다. 본 논문에서는 개요에서부터 동작까지를 톱다운(Top-Down)의 형태로 기술하기로 한다. IoT 플랫폼은 완전한 표준이 있을 수 없고, 기술을 선도하는 회사의 플랫폼이 사실상의 표준(de facto standard)이 된다. 따라서 상용화된 오픈 플랫폼 부분은 그대로 이용하기로 한다.

2.1 전체 시스템의 구성

그림 1은 본 논문의 IoT 시스템의 간략화된 구성을 보여 주고 있다. IoT 디바이스는 WiFi를 통해서 인터넷에 연결된다. WiFi를 통한 무선 접속을 하는

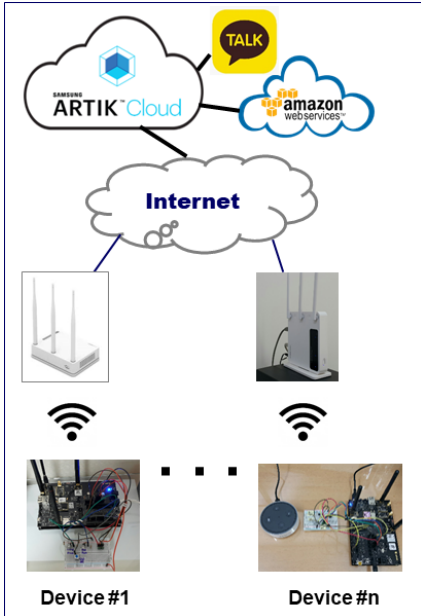


Fig. 1. Conceptual Structure of IoT System

과정에는 많은 절차가 있다. 각 단계마다 개별 명령어를 실행을 하여 접속하고 최종적으로 “ping 8.8.8.8”을 실행해서 인터넷 망에 접속되었는지를 확인 후 비로소 실험단계를 진행하여야 한다.

그림1에서 1번 IoT 디바이스와 n번 IoT 디바이스는 서로 다른 공간에 위치해 있고, 서로 다른 무선인터넷공유기에 연결되어 있다. 무선인터넷공유기가 사설 IP를 이용해서 250개 이상의 IoT 디바이스를 연결할 수 있고, 무선인터넷공유기 또한 제한 없이 붙일 수 있어서 IoT 디바이스의 연결 숫자에는 큰 제약이 없다. 클라우드에 사람이 사용자 등록 한 뒤, 클라우드에서 고유한 ID를 갖는 클라우드 디바이스를 만든다. 이 클라우드 디바이스와 로컬의 물리 디바이스(Things)를 소프트웨어를 설계하여 서로 연결을 지어 주면, 물리 디바이스와 클라우드 디바이스 간에 통신이 이루어진다. 해당 클라우드에 없는 기능은 제3자의 클라우드를 호출해서 더 많은 확장 기능을 실행할 수 있다.

기본 클라우드는 IoT 전용의 아티클라우드이다. 아티클라우드는 설계자가 정의한 디바이스로부터 입력을 받고, 전체 입력을 모니터(Monitor)하고, 설계자가 정해진 룰(Rule)에 따라 정해진 액션(Action)을 출력한다. 만약 추가적인 기능이 필요할 때는 그에 맞는 제3자(Third Party)의 클라우드를 API(Application Programming Interface)로 호출한다. 본 논문에서는

음성제어를 위해서 아마존(Amazon) 클라우드를 이용했고, 센서 값이 일정 이상일 때 경고를 스마트폰으로 전송하기 위해서는 카카오톡(KakaoTalk)클라우드를 이용하였다. 카카오톡과 아마존은 무료 서비스이다. 아티클라우드는 월 100KB까지 데이터는 무료이다. 따라서 하드웨어 모듈 이외는 별도의 비용이 발생하지 않는다.

2.2 원격에 있는 디바이스 제어 및 데이터 모니터링

IoT는 사물(Things)이 인터넷에 연결되어 정보를 교환하고, 클라우드에서 정해진 규칙에 따라서 로컬 HW에 제어출력을 전송하는 형태로 이루어진다. 사물 부분은 센서의 값을 출력하기도 하고, 제어결과에 따라 동작하기도 하는, 가장 말단 부분이다. 허브는 라우터 역할을 하는 부분으로 센서 입력과 클라우드 간을 보안이 보장된 방법으로 접속하는 역할을 한다. 외부에서 인터넷 망을 통한 해킹이 가장 취약하기 때문에 허브에서 클라우드 간의 보안은 무엇보다도 신뢰할 만한 방법을 사용해야 한다.

그림 2는 인터넷 WiFi가 연결되었지만 지역적으로 멀리 떨어진 서울(A지역)과 분당(B지역) 두 지역 간에 센서 데이터를 아티클라우드 한 곳에 모으고, 서울에서의 음성제어를 통해서 분당에 있는 램프를 켜도록 하는 실험을 하였다. AI 스피커는 아마존 에코닷(Amazon Echo Dot)을 사용하였고, 램프는 LED(Light Emitting Diode)로 대체하여 제어 신호가 전달되는 것을 확인 하였다.

AI 스피커 음성과 아티클라우드에서 설계자가 정의한 문장이 같은지 아마존 클라우드가 판단한다. 음성과 정의된 문장이 일치하면 아티클라우드는 아티보드로 액션 데이터를 보낸다. 아마존 클라우드를 아티클라우드에 연결하기 위해서는 아마존에서 계정등록

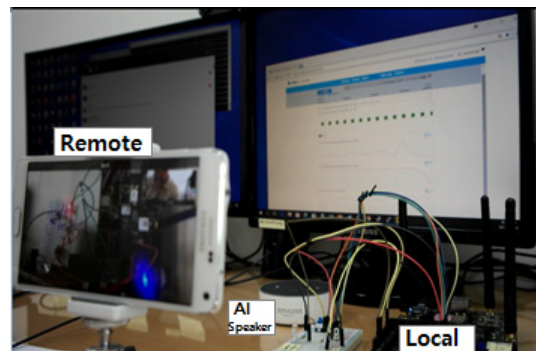


Fig. 2. Test between Local and Remote Area

을 하고 아틱 클라우드를 검색하여 이것을 활성화 시키는 방법으로 연결한다. 아마존 음성제어의 대상이 되려면, 영어문자로 된 아틱 클라우드의 디바이스를 만들어야 한다. 만약 거실 등을 제어 하려고 할 때는 영문으로 된 "Living Room" 이라는 이름의 아틱 클라우드의 디바이스를 만들고, 제어를 할 때는 음성에서 "Living Room"이라는 키워드가 들어가야 한다. 제어 대상의 행위는 아틱 클라우드의 룰(Rules)에서 정의한다. "Turn on" "Turn off"의 말을 액션의 "setOn" 혹은 "setOff"와 연결시킨다. 음성제어의 룰은 다음 장에서의 아틱 클라우드의 룰을 정할 때 더 기술하도록 한다.

그림 3은 온도와 습도 센서 데이터를 아틱 클라우드의 차트(Charts)화면을 통해서 모니터링 한 것이다. Y축의 눈금은 센서의 현재의 상태를 정밀하게 보여주기 위해서 Y축의 범위가 동적으로 바뀐다. 원격으로 온도의 변화를 보기 위하여, B지역의 센서에 헤어 드라이어로 순간 가열하였을 때 온도가 순간 45℃까지 치솟고 습도가 급격히 낮아지는 것을 볼 수 있었다. A 지역은 24℃ 정도의 온도와 40% 정도의 습도를 나타

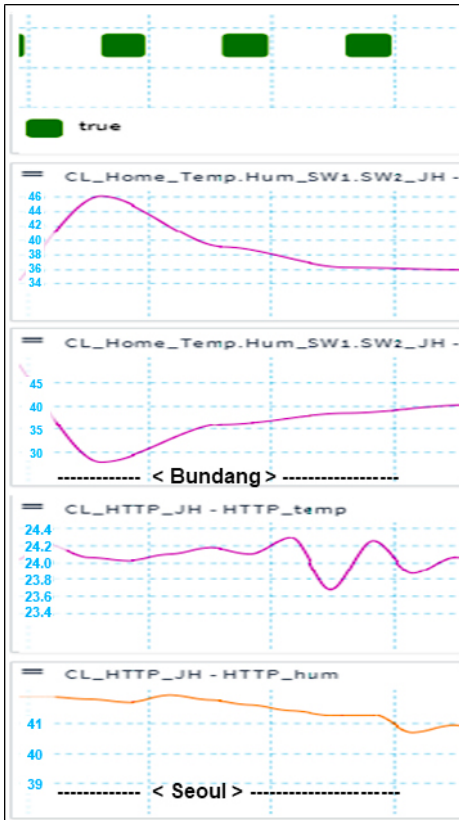


Fig. 3. Sensor Data in Seoul & Bundang>

내고 있다.

2.3 센서의 이상 상황에 대한 경고 전송

IoT 시스템이 기능을 실행중일 때, 사람이 계속 지켜보고 있기는 어렵다. 따라서 이상 상황이 되면 가장 일상적인 방법으로 경보를 알려주는 기능이 필수적이다. 가장 보편적인 방법으로 전자 메일과 메시지 서비스를 이용하는 것이다. 전자 메일은 해킹을 막기 위해 보안이 강화되어 있다. 전자 메일 보안등급을 낮추어야 만이 데이터 전송이 가능하였다. 그래서 여기서는 카카오톡 문자 서비스를 이용하였다. 카카오톡을 용이하게 활용하도록 이미 "KakaoTalk Send ToMe" 라는 디바이스 타입이 만들어져 있다. 이것을 불러 들여서 자신의 디바이스로 만들어 사용하면 된다.

그림 4는 온도센서를 강제로 가열하고 난 뒤, 상용의 문자 서비스로부터 경고 안내 메시지를 전달 받은 실험을 한 것이다. 안내 문자를 받는 주기는 1초 이상의 값으로 조절 할 수 있다. 센서 데이터의 수신 주기와 안내 문자를 보내는 주기는 모두 클라우드를 사용하는 과금에 해당하기 때문에 주기는 필요이상으로 빠르게 할 필요는 없다.

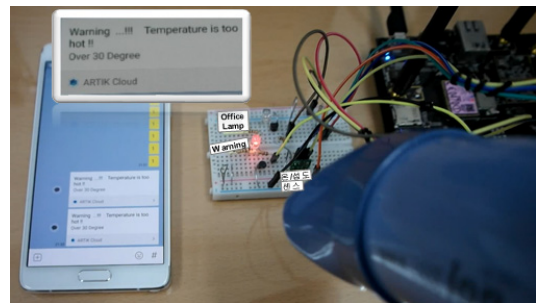


Fig. 4. Send Alerts via KakaoTalk

2.4 아틱클라우드의 디바이스

아틱 클라우드는 각 개인마다 계정을 등록해서 사용한다. 각 개인은 자신이 사용할 클라우드 디바이스(Device)를 만들 수 있다.

디바이스는 그림 5의 예와 같이 만든다. 먼저 디바이스 타입에서 필요한 기능에 대한 변수를 정의 한다. 필드(Fields)는 센스 값이 전달되는 변수의 역할을 한다. 액션은 룰(Rules)에 의해서 결정된 결과가 다시 사물로 전달되는 통로이다. "Fields"가 입력이고, "Actions"가 출력어 된다. 자신이 필요한 디바이스를 만들 때는 자신이 직접 만든 디바이스 타입을 불러와서 자신의 디바이스로 만들 수도 있고, 남들이 만들어

Fields	Actions																
<table border="1"> <tr><td>hum</td><td>Double</td><td>%</td></tr> <tr><td>sw1</td><td>Boolean</td><td>bit</td></tr> <tr><td>sw2</td><td>Boolean</td><td>bit</td></tr> <tr><td>temp</td><td>Double</td><td>°C</td></tr> </table>	hum	Double	%	sw1	Boolean	bit	sw2	Boolean	bit	temp	Double	°C	<table border="1"> <tr><td>lock</td></tr> <tr><td>setOff</td></tr> <tr><td>setOn</td></tr> <tr><td>unlock</td></tr> </table>	lock	setOff	setOn	unlock
hum	Double	%															
sw1	Boolean	bit															
sw2	Boolean	bit															
temp	Double	°C															
lock																	
setOff																	
setOn																	
unlock																	

DEVICE ADDED ON	DEVICE TYPE
15/Jul/2018	Home_Temp_Hum_SW1_SW2_JH
DEVICE ID	DEVICE TYPE ID
1283f30bebacc43c9ac28c8ced50d4171	dtb87c7be859334ec8b5bec147793341fc
DEVICE TOKEN	
e07083e6d18a49caa05fd0af17ad08ce	REVOKE TOKEN

Fig. 5. An Example of a Defined Device

서 공개한 디바이스 타입을 이용해서 자신의 디바이스로 만들 수도 있다. 각 디바이스는 유일한 디바이스 ID와 토큰을 갖는다. ID와 토큰은 각각 32개의 16진수 숫자로 표현된 충분히 긴 값이다. 이 데이터는 인증과 디바이스 구분을 위해서 각각 사용된다. 모든 입력력은 이 디바이스를 통해서 이루어진다.

2.5 아틱 클라우드의 룰

그림 6은 카카오톡으로 경고 안내 문자를 보내는 경우와 음성으로 원격지역의 램프를 온/오프 하는 경우를 요약한 것을 나타 낸 것이다. 더 많은 디바이스와 더 많은 변수를 동원하여 더 복잡한 경우도 나타낼 수 있다. 경고 안내문의 “Warning...!!! Temperature is too hot”의 문자는 그림 6에서 볼 수 있는 것처럼 룰에서 이미 정의하였기 때문에 이에 따라 안내 문자가 발송된 것이다.

그림 2에서의 음성제어는 “turn on bed room lamp in pundang” 이라고 그림 6에 정의되어 있다. 룰에서 이렇게 정의가 되어 있기 때문에 이 단어가 나오면 아마존 클라우드의 알렉사(Alexa) 서비스는 “setOn”을 “Bed Room Lamp_JH”라는 아틱 클라우드 디바이스에 보낸다. 이 디바이스는 출력을 아틱 보드에 전달하는 디바이스 이다.

Fig. 6. Example of Rule Defined in ARTIK Colud

2.6 Node-red를 이용한 기능 기술

그림 7은 Node-red를 이용하여, 아틱 보드에서 입력된 데이터를 아틱 클라우드로 보내고 아틱 클라우드에서 처리된 결과를 다시 아틱 보드로 전송하는 동작을 기술한 것이다. 아틱보드는 클라이언트로 동작하고 아틱 클라우드는 서버로 동작한다. 각 박스(Box)안에도 코드를 설계해서 넣어야 하지만, 인터페이스와 신호흐름에 대해서는 이미 표현되어 있기 때문에 해당하는 기능만 설계하면 된다. 클라우드 기반으로 돌아가는 Node-red를 이용하면 클라이언트와 서버간의 방대한 네트워크 SW를 개발하는 노력을 대폭 줄일 수 있다.

그림 7의 윗부분은 B지역에서의 제어를 나타낸 것이다. 아틱 보드의 입력이 아틱 클라우드 입력으로 전달되는 것을 나타내고 있다. 맨 앞단의 “timestamp” 노드는 입력의 전달 주기를 조절 한다. 최소 1초이고, 그 이상의 값으로 주기를 조절 한다. 이 주기는 데이터의 과금과 직결되기 때문에 필요이상 빠르게 할 필요가 없다.

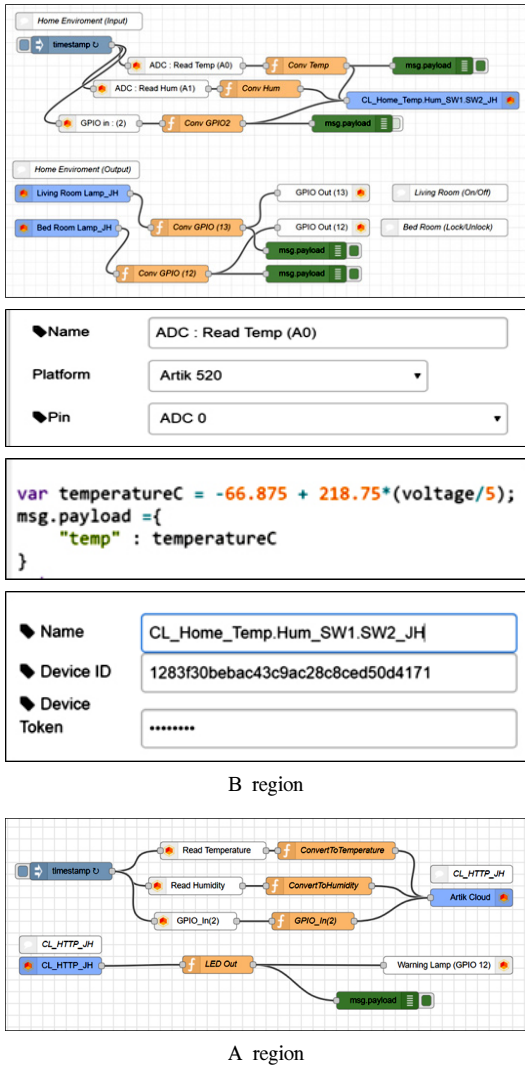


Fig. 7. Function Description using Node_Red

이렇게 받은 아틱 보드의 입력은 기능(Function) 노드를 거쳐 아틱 클라우드에 입력된다. 클라우드의 입력과 출력은 룰에 의해서 관계가 정의 된다. 클라우드의 입출력은 모두 클라우드의 디바이스를 통해서 이루어진다. 데이터 처리나 변환이 필요한 경우는 해당 기능을 노드 박스 안에 Node.js언어로 기술한다. 그림 7에는 실제로 사용된 센서의 출력 값을 °C로 표현된 온도 수치와 맞추기 위해서 관계식이 추가 되었다. 특정 센서 부품의 전압출력을 온도로 정의된 값과 맞추기 위한 관계식이다. Node-red의 방식으로 설계를 하면, 노드를 불러올 때 일반 네트워크 프로그램은 모듈로 따라오기 때문에 설계자는 목표하는 기능에 집중해서 설계 할 수 있다. 그림 7의 아래 부분은 A 지역

의 제어를 나타낸 것이다.

III. 결론

본 논문에서는 인터넷 WiFi 연결이 가능한, 복수의 지역에 분포되어 있는 센서들의 데이터를 하나의 클라우드로 모아서 모니터링 한 결과를 확인해 보았다. 또한 아틱 클라우드가 동작하는 중에 외부 클라우드를 API로 호출하여 융합된 동작을 하는 결과도 확인하였다. 특정 센스에 정해진 일정 이상의 과도한 값이 감지 될 때, 상용 서비스의 문자 메시지를 통해 경고 안내를 보내는 동작도 확인 하였다. 이러한 동작은 IoT 시스템에서 일어나는 대부분의 동작이다.

그리고 IoT 시스템은 기능적으로 비교적 단순한 편이지만 센서, 허브, 클라우드 라는 수직적인 계층을 가지고 있어서 넓은 범위의 기술개발을 동시에 해야 한다. 따라서 넓은 범위의 일을 추진해 가야 할 때에는 이에 상응하는 잘 정립된 방법론이 필요하다. IoT 전용의 아틱 클라우드를 활용하고, 클라우드 기반의 Node-red를 사용하여 기능을 설계하고, 세미커스텀 하드웨어인 아틱 보드를 활용하여 IoT 시스템을 구성하여 테스트를 하였다. 사물, 허브, 클라우드의 3단 구성을 모두 할 수 있었다. 데이터 보안이 적용된 형태로 양산까지를 고려한다면 본 논문에서의 솔루션이 최적의 방법론이라 할 수 있다.

References

- [1] H. Lee, "Embedded system framework and its implementation for device-to-device intelligent communication of manufacturing IoT device considering smart factory," *J. KIIS*, vol. 27, no. 5, pp. 459-465, Oct. 2017.
- [2] G.-S. Park, T. T. Tran, V. C. Dang, K.-J. Gil, Y.-B. Shin, J.-W. Choi, and J.-W. Kim "A development of real-time monitoring system in industrial factory based on cloud platform using IoT device," *IEMEK J. Embedded Syst. and Appl.*, vol. 13, no. 1, pp. 25-32, Feb. 2018.
- [3] S.-H. Kim, J.-M. Jeong, M.-T. Hwang, and C.-S. Kang, "Development of an IoT-based atmospheric environment measurement and analysis system," *J. KICS*, pp. 1750-1764, vol. 42, no. 09, Sep. 2017.

- [4] Ademir F. da Silva, R. L. Ohta, arcelo N. dos Santos, and Alecio P. D. Binotto “A cloud-based architecture for the internet of things targeting industrial devices remote monitoring and control,” *IFAC-PapersOnLine*, vol. 49, no. 30, pp. 108-113, 2016.
- [5] A. Saxena and R. Prakash, “Universal BLDC controller -With IIoT set of features,” *2017 Int. Conf. Nextgen Electron. Technol.*, pp. 419-425, Mar. 2017.
- [6] S. I. Hwang, S. Y. Joo, and J. M. Ju, “A study of IoT platform for the smart farm factory,” in *Proc. Symp. KICS*, pp. 520-521, Jan. 2016.
- [7] C. Ahn, *4th Industry IoT Platform Training*, 1st Ed., ICOM Press, 2018.

배 접 한 (Jum-Han Bae)



1985년 : 경북 대학교 전자공학
과 졸업

2013년~현재 : 성균관 대학교 전
자전기컴퓨터 공학 박사과정

1985년~2014년 : 삼성전자
DMC 연구소

2015년~2016년 SK하이닉스 메
모리시스템 연구소

현재 : 숭실대학교 IT대학 전자정보공학부 산학중점
교수

<관심분야> SoC Design, Embedded System, IoT.

김 종 태 (Jong Tae Kim)

한국통신학회 논문지 제 34권 5호 참조

현재 성균관대학교 전자전기공학부 교수