

http://dx.doi.org/10.17703/JCCT.2022.8.2.355

JCCT 2022-3-46

## 마이크로 데이터센터의 환경 모니터링 시스템 개발 연구

# A Study on the Development for Environment Monitoring System of Micro Data Center

이갑래\*, 김영식\*\*

Kap Rai Lee\*, Young Sik Kim\*\*

**요약** 본 연구에서는 마이크로 데이터 센터의 환경 모니터링 시스템(EMS)을 설계하고 개발하는 방법을 나타낸다. 설계하는 환경 모니터링 시스템은 IoT 센서 및 싱글보드 컴퓨터를 이용하여 데이터센터 운영환경을 실시간으로 감지하여 데이터를 해석하고 운영환경을 모니터링한다. 먼저, EMS 하드웨어 플랫폼 및 IoT 센싱 패키지 설계 방법에 대하여 나타낸다. 또한 서로 다른 프로토콜을 갖는 IoT 센서에 대한 데이터 수집 SW를 설계하고 원격에서 모니터링이 가능한 EMS 모니터링 SW를 개발한다. 데이터 수집 SW는 센서 수집 모듈과 수집 매니저 모듈로 구성된다. EMS 소프트웨어는 마이크로 서비스 아키텍처(MSA) 개념을 적용하여 설계하며 각각의 서비스는 독립적인 비즈니스 로직을 포함한다.

**주요어** : 마이크로 데이터센터, 환경 모니터링 시스템, IoT 센싱패키지, 싱글보드 컴퓨터, 모니터링 SW, 데이터 수집 SW

**Abstract** In this paper, we present design and developing method for EMS(environment monitoring system) of micro data center. This developing EMS monitors operating environment of micro data center and analyze sensing data through IoT(Internet of things) sensors in real time. Firstly we present configuration method of IoT sensing package and design method EMS hardware platform. Secondly we design data collector software for data collection of IoT sensor with different protocol and develop monitoring software of EMS. The data collector software consists of sensor collector module and collector manager module. Also we design EMS software which has micro service architecture structural style and component based business logic.

**Key words** : Micro data center, Environment monitoring system, IoT sensing package, Single board computer, Monitoring SW, Data collector SW

### 1. 서론

사물인터넷 서비스 및 빅데이터 사용이 늘어나면서 대규모 데이터 센터는 물론 소규모 마이크로 데이터 센터 필요성도 증가하고 있다. 마이크로 데이터센터는 옛 지컴퓨팅의 확산, 정부 주도 스마트팩토리 확산 등으로

그 수요가 최근 증가하고 있다. 마이크로 데이터센터는 옛 지 컴퓨팅을 위해 네트워크, 서버, 스토리지, 보안 등 컴퓨팅 인프라를 제공하는 통합 랙(Rack)솔루션이다. 데이터센터 인프라에 필수적인 전력, 쿨링, 보안, 원격 감시, 인프라 및 환경 관리 등의 일체로 제공하는 소형 데이터센터 그 자체를 의미한다. 데이터센터의 안정적인

\*정회원, 평택대학교 스마트자동차학과 교수 (제1저자, 교신저자)

\*\*정회원, (주)비바엔에스 대표이사 (공동저자)

접수일: 2022년 1월 27일, 수정완료일: 2022년 3월 1일

게재확정일: 2022년 3월 8일

Received: January 27, 2022 / Revised: March 1, 2022

Accepted: March 8, 2022

\*Corresponding Author: krlee@ptu.ac.kr

Dept. of Smart Automobile, Pyeongtaek Univ., Korea

운영관리를 위해서는 위치별 장비의 온도, 습도, 전력, 도어 개폐 등의 다양한 정보를 실시간으로 수집하고 분석되어야 한다[1]. 또한 IoT 센서 기반으로 운영환경을 모니터링 및 관리하는 EMS(Environmental Monitoring System) 운영이 필수적이다.

데이터 센터 모니터링에 대한 기존 연구로는 데이터 센터 구축 시에 다양한 설비들을 통합 관리할 수 있는 시스템에 설계에 관한 연구[2, 3]와 데이터 센터의 다양한 센서에서 발생하는 환경 정보를 효율적으로 모니터링하기 위한 센서 데이터 처리 방안에 대한 연구가 있다[4]. 또한 데이터 센터의 전력 소비 패턴을 유형별로 분석하여 상황인지 기반의 전력 절감 패턴을 추천하는 전력 절감 모니터링 시스템을 개발한 연구가 있다[5]. 현재 마이크로 데이터센터용 EMS 시장은 라리탄(Raritan) 및 NTI 사 등의 고가의 외산 솔루션들이 주도하고 있으므로 국산화 개발도 시급한 실정이다. EMS은 구성요소들이 가지는 고유의 역할을 유기적으로 통합하여 운영 소프트웨어, DBMS, IoT센서, 설비 모니터링 등을 하나의 솔루션으로 관리하여야 한다.

본 연구에서는 마이크로 데이터 센터용 환경 모니터링 시스템(EMS)을 설계하고 개발하는 방법을 나타낸다. 단일 랙(Rack) 기반의 마이크로 데이터센터 운영환경을 IoT 센서 기반으로 모니터링하는 하드웨어 및 소프트웨어 일체형 EMS을 설계하고 구현한다. 본 논문의 구성은 다음과 같다. 2장에서는 EMS 하드웨어 및 모듈형 IoT 센싱 패키지 개발에 대하여 나타낸다. 3장에서는 별도 서버 없이 EMS 자체만으로 원격에서 모바일 기반으로 모니터링 및 관리가 가능한 반응형 웹 GUI를 개발한다. EMS 소프트웨어는 독립적인 비즈니스 로직을 포함하는 마이크로 서비스 아키텍츠(MSA) 개념으로 설계한다.

## II. EMS HW 및 센싱 패키지 개발

### 1. IoT 센싱 패키지 개발

#### 1) 구성 방식 설계

IoT 센싱 패키지는 그림 1과 같이 시리얼 허브 기반 방식에 적합하게 개발한다. 아날로그 온습도 센서는 아날로그-시리얼 컨버터에 연결하고 아날로그-시리얼 컨버터는 시리얼 허브에 직접 연결하게 설계하였다. 또한 시리얼 온·습도센서 및 시리얼 차압센서는 시리얼 허브에

직접 연결하게 설계하였다. 시리얼 허브는 엣지 컴퓨팅 모듈과 직접 연결된다.

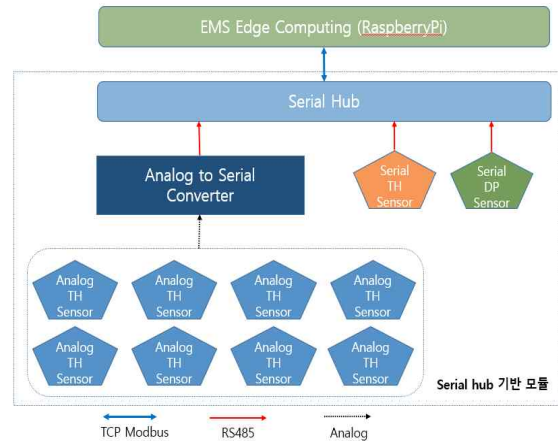


그림 1. 시리얼 허브 기반의 데이터 수집  
Figure 1. Data collection based on serial hub

### 2) IoT 센싱 패키지 구현

#### o 온·습도 센서 모듈

온·습도 센서는 공기중의 상대습도와 온도를 측정하는 초정밀 센서로 안정적인 운영을 보장하면서 오염과 응축이 일어나는 환경에 강해야 한다. 케이스를 열지 않고도 설치가 가능한 모델(EE160)과 버튼과 함께 LED 디스플레이를 일체형으로 제공하는 모델(RTH 204D)로 구성한다. 센서당 온도와 습도 2개 데이터를 시리얼 데이터로 출력하며 RS485 통신용 RJ45 port로 데이터 수집기와 연결한다. 다수의 온·습도센서는 데이터 체인(Daisy Chain) 방식으로 서로 연결하여 사용할 수 있다.

#### o 차압 센서 모듈

차압센서는 공기 또는 비부식성 기체의 차압, 정압, 대기압 등을 측정한다. 내부 버튼 또는 덤스위치를 이용하여 센서 설정 변경이 가능한 모델(RTA101D)로 구성하였다. 센서당 1개 데이터를 시리얼 데이터로 출력하며 RS485 통신용 RJ45 포트 데이터 수집기와 연결할 수 있다.

#### o 전압·전류 센서 모듈

계측기능이 포함된 미터링 랙(Metering Rack) PDU(Power Distribution Unit)을 사용한다. 계측기능이 없는 랙 PDU가 이미 구성되어 있으면 미터링 릴레이

유닛을 공급 전력선과 랙 PDU 사이에 설치하여 사용한다. PDU당 전압 및 전류 2개의 데이터를 이더넷 포트 또는 RS485로 출력한다. 브릿지모드(Bridge Mode) 연결로 PDU 상호 간에 데이터 체인 구성이 가능하다.

o 도어락 센서 모듈

엣지 컴퓨팅모듈 뿐만 아니라 관제실이나 원격 모바일로 랙 도어(Rack Door)의 잠금 상태를 감시하고 랙 도어를 제어할 수 있게 장치 모듈을 구성한다. 필요한 구성 요소로는 전기식 도어락 핸들러(Electric Door Lock Handle), 도어락 센싱스위치(Door Closure Sensing Switch), 신호 수집중계기가 필요하다. 도어락 핸들의 잠금장치는 전기식 락(Electric Lock)과 기계식 락(Mechanical Lock)으로 구성되어 있으며, 기계식 락은 회로의 이상 등에 의해 전기식 락이 작동하지 않을 경우를 대비하여 비상용으로 사용한다. 정상적인 상태에서 기계식 락은 잠금 상태를 유지해야 하며, 기계식 락이 열리면 전기식 락의 상태와 상관없이 락은 해제된다. 기계식 락은 비상 용도이므로 공통으로 사용할 마스터(Master Key)를 사용한다. 전기식 도어락 핸들은 8핀 케이블로 제어 유닛(Control Unit)과 연결되며 그림 2와 같이 구성된다.

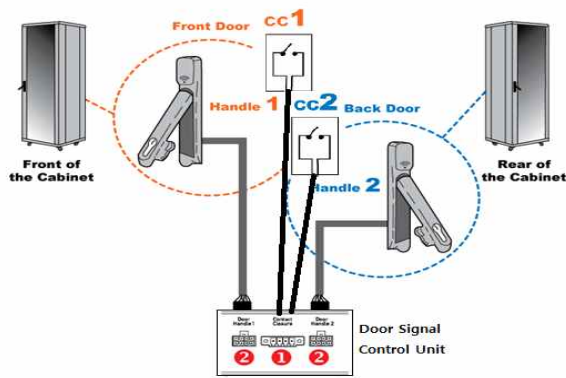


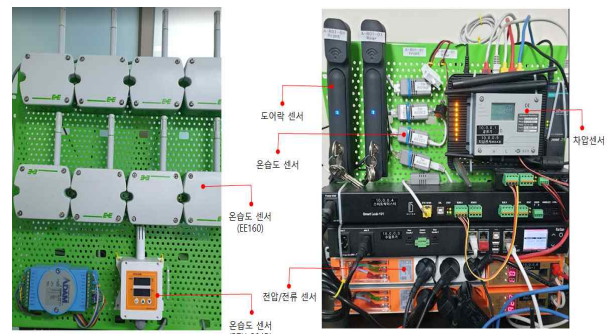
그림 2. 도어락 센서 구성도  
 Figure 2. Configuration of door lock sensor

엣지 컴퓨팅 모듈에서 제어하는 신호는 전기식 락 상태(Electric Lock Status), 기계식 락 상태(Mechanical Lock Status), 전기식 락 해제( Electric Lock Release) 3개 신호가 사용되며, 별도로 도어가 닫혀있는 지 여부를 감시하기 위한 신호선이 추가로 사용되어, 총 4개 데이터가 사용된다. 도어락 제어 유닛은 랙 별로 1개씩 구성되면 이들 유닛은 컨테이너먼트(Containment) 단위로

RS485 데이터 체인으로 연결되어 서버와 TCP로 연결된다.

o 개발된 IoT 센싱 패키지

온·습도 센서, 전압·전류 센서, 도어락 센서 등을 포함하는 다양한 IoT 센서를 수용한 IoT 센싱 패키지는 아래 그림 3과 같다.



다양한 IoT 센서 수용

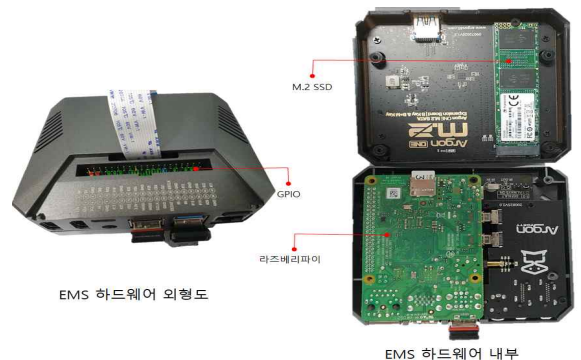
그림 3. IoT 센싱 패키지  
 Figure 3. IoT sensing package with various sensors

2. EMS 컴퓨팅 하드웨어 플랫폼

EMS의 컴퓨팅 하드웨어 플랫폼으로 싱글보드 컴퓨터인 라즈베리파이를 사용한다. 라즈베리파이 싱글보드 컴퓨터의 대략적인 사양은 다음과 같다.

- 4GB RAM, 500G M.2 SSD, Gigabit Ethernet
- 40 pin GPIO, 범용 OS (Ubuntu) 사용

또한 냉각을 위하여 알루미늄 케이스를 사용하여 24시간 상시 구동이 가능하도록 설계하였다. 그림 4는 설계 구현된 EMS 엣지 컴퓨팅 하드웨어 모듈을 나타낸다.



EMS 하드웨어 외형도

EMS 하드웨어 내부

그림 4. EMS 컴퓨팅 하드웨어 플랫폼  
 Figure 4. Computing hardware platform of EMS

III. EMS 소프트웨어 개발

1. EMS 소프트웨어 구조

그림 5는 EMS 소프트웨어 스택의 모듈 구성을 나타낸다.

- 라즈베리파이에 리눅스 커널을 설치하고 이 위에 오픈소스 프로젝트인 도커(docker)를 설치하여, 응용 프로그램들을 ‘소프트웨어 컨테이너’ 안에 배치시키는 일을 자동화한다.
- 인플렉스 DB(influxDB) 시계열 데이터베이스를 설치하고, 관계형 데이터베이스인 mySQL과 비관계형 데이터 베이스 관리시스템인 레디스(Redis)를 설치한다.
- 데이터 수집모듈(Collectors)과 레스트(Rest) 기반의 서비스 API인 레스트풀API(RESTful API)를 구현한다. - 또한 모바일 기반으로 원격에서 모니터링 및 관리 가능한 반응형 웹 GUI를 개발한다.

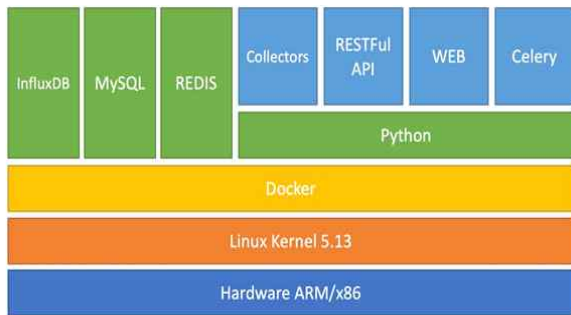


그림 5. EMS 컴퓨팅 스택  
Figure 5. Computing stacks of EMS

2. EMS SW 개발

설치된 센서의 종류에 따라 통신방법 및 프로토콜이 다양하게 사용되기 때문에 다양한 통신 방식을 수용하도록 EMS SW를 구현하였다.

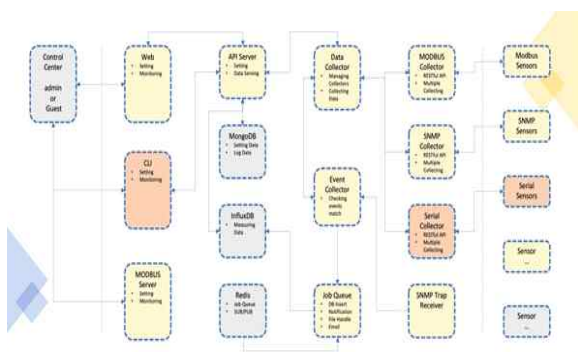


그림 6. EMS 소프트웨어 MSA 컨테이너  
Figure 6. MSA container of EMS SW

EMS SW는 서비스 디자인 스타일로 개발하였으며, 그림 6과 같이 각각의 서비스는 독립적인 비즈니스 로직을 포함하게 하였다. 작은 서비스의 결합을 통해 하나의 큰 응용 프로그램을 개발하는 마이크로 서비스 아키텍처(MSA) 개념을 적용하여 개발하였다.

(1) 수집 데이터 모듈 및 수집 매니저 모듈

컨테이너에서 사용하는 설비 및 구성요소는 상황에 따라 이동 또는 재편될 수 있기 때문에 데이터 수집 (Collector) 모듈은 그림 7과 같이 센서 중심의 수집 모듈로 개발하였다. 프로토콜별 수집 데이터 모듈은 수집 매니저 모듈과 연결된다.

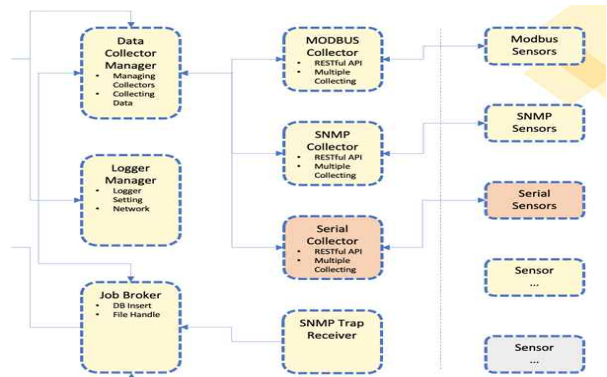


그림 7. 수집 데이터 모듈과 수집 매니저 모듈  
Figure 7. Collector data and collector manager module

MODBUS, SNMP, Rest API 등 다양한 프로토콜을 하나의 모듈이 처리하면 복잡도 상승과 의존성이 높아지므로 개별 전용 수집 모듈로 분리하였다. 프로토콜별 개별 수집 모듈 프로그램으로 작성하여 현장 상황에 쉽게 대응할 수 있도록 파이썬 스크립트 명령어를 사용하여 센서모듈 수집 부분을 작성하였다. 수집된 데이터는 길이에 따라 최대 64Bit의 문자열과 정수, 실수중 하나를 선택하여 정의된 데이터 타입 테이블을 이용하여 변환한다. 수집된 데이터는 관리자가 정의한 최대 보관 개수만큼 메모리에 저장된다. 저장된 최대 개수에 도달하면 가장 오래된 데이터부터 순차적으로 버리면서 최대 보관 수를 유지한다. 그림 7에 나타난 것과 같이, 데이터 콜렉터 매니저의 요청에 따라 전부 또는 가장 최신데이터를 RESTful API로 제공한다. 모든 수집 모듈은 통일된 REST API 를 제공한다. 데이터 콜렉터 매니저는 각 프로토콜 별 콜렉터가 수집중인 데이터에서 마지막 데이터를 가져오며, 데이터는 잡 브로커에 요청



하여 데이터 베이스에 저장하도록 한다. 데이터 콜렉터 매니저는 데이터를 가져올 수집기를 지정하고 가져올 데이터 항목과 DB에 저장되기 위한 메타정보(InfluxDB의 measurement, tags)를 가지고 있다. 이 메타정보를 이용하여 센서로부터 수집된 데이터를 가진 콜렉터에 접근하여 데이터를 가져온다.

데이터의 수집, 처리, 저장을 개별 프로세스로 분산 처리하여 개발한다. 데이터를 수집하는 동작은 스케줄링에 의해 등록된 수집기가 동시에 실행된다. 데이터는 그림 8과 같은 단계를 거쳐 최종적으로 데이터 베이스에 들어가게 된다.

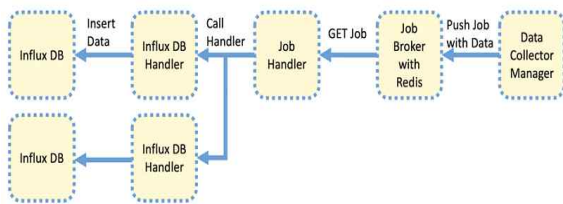


그림 8. 수집된 데이터의 이동 흐름  
 Figure 8. Collector data and collector manager module

데이터 콜렉터 모듈에서 수집된 데이터는 메타 정보에 따라 잡 오더 메시지로 만들고 레디스에 PUSH한다. 잡 브로커는 레디스에 들어온 잡들을 처리하는 프로그램으로서 처리할 수 있을 만큼 레디스에서 잡을 가져오고 요청받은 핸들러를 실행한다. 멀티프로세싱을 이용하여 병목 없도록 핸들러를 분산 처리한다.

SNMP를 지원하는 센서 모듈을 사용할 경우 SNMP 트랩(Trap)을 이용하여 등록된 이벤트를 즉시 처리할 수 있도록 한다. 도어 개폐, 소방 화재신호와 같은 즉시성이 필요한 이벤트를 트랩 리시브(TrapReceiver)에 등록하여 디바이스가 보내는 트랩 신호를 처리할 수 있게 한다.

(2) DB 연동 모니터링 모듈

DB가 연동되어 작동하는 모니터링 SW의 실행에 필요한 전체 구성 컨테이너는 데이터 로거 컨테이너, 시계열 데이터베이스 컨테이너, 관계형 데이터베이스 컨테이너, 데이터 체어 컨테이너, 웹 서버 컨테이너로 구성한다.

o 시계열 데이터 베이스 컨테이너

시계열 데이터베이스는 시간이라는 기본 기준열을 기반으로 정수나 실수와 같은 정보를 담고 있는 필드와 해당 내용을 필터링할 수 있는 태그 정보를 이용하여 데이터를 저장한다. 시계열 데이터베이스는 시간 축에 따라 데이터가 생성만 될 뿐 변경이나 삭제는 되지 않아 비교적 단순하고 빠르게 정보를 처리할 수 있다.

o 데이터 로거 컨테이너

데이터 로거 컨테이너는 도커파일(Dockerfile)로 구성된다. 데이터 로거의 소스는 파이썬 언어로 작성되었으며 우분투(18.04) 기본 이미지에서 출발하여 동작한다. 실제 작동되는 start.sh 쉘스크립트는 InfluxDB 시계열 DB 서비스가 돌고 있는지 확인한다. 확인하여 돌고 있으면 환경변수에 기술된 설정파일(CONFIG\_FILE)에 기술된 연동 장치 중의 특정 장치 (DEVICE\_SERVER)를 이용하여 특정 IP 별로 별도의 컨테이너로 수집한다. 시계열 데이터베이스는 도커 허브(Docker Hub)에 존재하는 인플렉스 DB를 이용한다.

o 센서 데이터 매핑

센서 연동 모듈은 실제 센서와 수집한 데이터를 매핑하는 것이 필요하며, 매핑 테이블은 다음과 같은 규칙을 따른다.

- 우선 디바이스 서버(device-servers) 항목은 개별 IP 주소를 갖는 모드 버스 접속 장치를 의미한다.
- 그 하위의 모듈(module)은 모드 버스의 슬레이브를 의미하고, 그 하위 센서 항목이 센서와 데이터 목록의 매핑 테이블에 해당한다.

3. EMS의 연동 성능

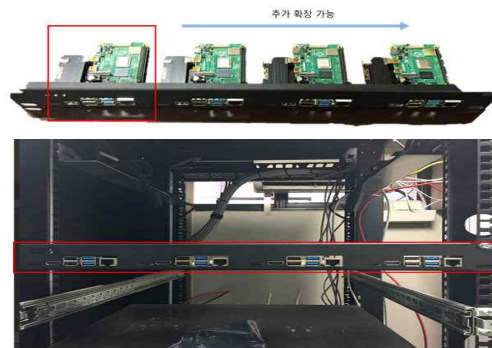


그림 9. EMS 컴퓨팅 랙 유닛  
 Figure 9. EMS computing rack unit of EMS

그림 9는 개발된 EMS 컴퓨팅 하드웨어 플랫폼이 랙에 장착된 모습을 나타낸다. 그림 9의 상위그림은 EMS 컴퓨팅 하드웨어 플랫폼이 랙 유닛에 장착된 모습이고 하위 그림은 랙 유닛이 랙에 장착된 모습이다.

그림 10은 개발된 EMS가 랙에 장착되어 연동되어 나타난 마이크로데이터 센터의 모니터링 시계열 그래프이다. 시계열 DB인 인플릭스 DB에서 데이터를 가지고 와서 그라파나(Grafana) 오픈소스를 이용하여 그래프 형식으로 나타낸 모니터링 결과이다.



그림 10. EMS 모니터링 그래프  
Figure 10. EMS monitoring graph

#### IV. 결 론

본 논문에서는 마이크로 데이터 센터용 환경 모니터링 시스템(EMS)을 설계하고 개발하는 방법을 나타내었다. 단일 랙(Rack) 기반의 마이크로 데이터센터 운영환경을 IoT 센서 기반으로 모니터링하는 하드웨어 및 소프트웨어 일체형 EMS를 설계하고 구현하였다. 먼저, EMS 하드웨어 및 모듈형 IoT 센싱 패키지 개발 등의 하드웨어 설계 방법에 대하여 나타내었다. 또한 IoT 센서 데이터 수집 SW 모듈을 개발하고, 별도 서버 없이 EMS 자체만으로 원격에서 모바일 기반으로 모니터링 및 관리가 가능한 반응형 웹 GUI를 개발하였다. EMS 소프트웨어는 독립적인 비즈니스 로직을 포함하는 마이크로 서비스 아키텍처(MSA) 개념으로 설계하였다.

#### References

- [1] S. Meng, L. Liu, et al., "State monitoring in cloud data centers," Proc. of IEEE Trans. Knowledge Data Eng. Vol 23, no. 9, pp. 1328-1344, 2011. DOI:10.1109/TKDE.2011.70
- [2] W. H. Choi, H. S. Hwang and C. S. Kim, "The study on the design of the integrated monitoring system of facilities in data center", Journal of the Korea Institute of Information and Communication Engineering, Vol. 19, No. 4, pp. 909-916, Apr. 2015. DOI:10.6109/jkiice.2015.19.4. 909
- [3] T. G. Ahn, G. J. Kim, H. M. Kim, K. Y. Park, J. W. Song and B. S. Sim, "Configure intelligent field test study on a comprehensive monitoring system," Korea Railroad Association Conference Proceedings, pp 1816-1820, 2012.
- [4] D H. Sun, S. J. Kim, K. G. Choi and S. H. Hwang, "Streaming Massive Sensor Data Processing in Data Center Monitoring System," The Journal of Korean Institute of Next Generation Computing, Vol. 13, no. 6, pp. 77-87, 2017.
- [5] H. J. Lee, M. Y. Jung, C. G. Kim and H. J. Kim, "A Study on Energy Saving Monitoring System of Data Center based on Context Awareness", Journal of Convergence for Information Technology. vol. 9, no.1, pp. 19-27, 2019. <https://doi.org/10.22156/CS4SMB.2019.9.1.019>

※ 본 연구는 교육부와 한국연구재단의 재원으로 지원을 받아 수행된 산학협력 선도대학(LINC+) 육성사업의 연구결과입니다.