

---

# KETI의 Mobius플랫폼과 LoRa를 활용한 oneM2M규격의 IoT 환경 구축에 관한 연구

이소행\* 송범성\*\*

청암대학교\*, (주)티이에프

Study on Construction of IoT environments oneM2M standards utilizing the KETI  
Mobius platform and LoRa

So-Haeng Lee\*, Bub-Sung Song\*\*

Cheongam College\*, (주)티이에프\*\*

E-mail : sohaeng.lee@gmail.com\*, peicsong@gmail.com\*\*

## 요 약

최근 국내에서는 IoT를 기반으로 한 신 성장동력 모델들이 기업과 국가기관에 의해 제안되고 있다. 이러한 인터넷 기반의 융합중심 IoT가 현장에 적용되면서 다양한 경제적 가치의 향상과 효율성의 증대등이 현실화될 전망이다. 본 논문에서는 KETI(전자부품연구원)에서 개발 보급중인 oneM2M기반 표준 플랫폼 Mobius와 Gateway용 플랫폼 &Cube, 그리고 저전력, 장거리 RF통신이 가능한 LoRa 통신모듈을 이용하여 현장 적용이 가능한 환경을 구축하여 제안한다.

## ABSTRACT

In recent domestic new growth model based on the IoT it has been proposed by the industry and national authorities. As applied to these Internet-based IoT convergence center of the field is expected to be realized, including the improvement and increase of efficiency of various economic values. In this paper, using KETI (Korea Electronics Technology Institute) developed distribution being oneM2M standards-based platform from Mobius and Gateway Platforms & Cube, and low power, capable of long-range RF communication LoRa communication module is proposed to build the site applicable environment.

## 키워드

Mobius, IoT, 사물인터넷, oneM2M,

## 1. 서 론

사물인터넷은 2012년 12월 oneM2M을 설립(한,EU,미,일,중의 7개 표준기관, 267개 업체가 참여)하여 자동차, 의료, 가전, 전력 등 IoT 호환성 확보를 위해 공동 플랫폼 표준화를 추진하였다. 이에 국내에서는 TTA에 기반 한 13개의 업체가

참여하고 있으며 KETI (전자부품연구원 - 이하 KETI)에서는 2015년부터 oneM2M기반의 Mobius라는 플랫폼을 개발 배포하고 있다. Mobius는 KETI에서 개발한 oneM2M기반의 사물인터넷 플랫폼이다. 사물인터넷은 특정 기관이나 기업, 개별적으로 운영되던 형태에서 개방형과 표준화를 접목한 인터넷 서비스로 진화되면서 주변 기술과 융합되

어 하나의 생태계를 구성하고 있다. 또 모든 것이 연결되는 과정에서 방대한 비정형 데이터의 처리 및 분석을 위한 정보처리 산업에도 상당한 시장이 형성될 것으로 보인다. 국내 사물인터넷 시장은 2015년 3.8조 원에서 2022년 22.9조 원까지 성장할 것이며, 서비스 관련 매출의 비중이 52.6%까지 증가하며 성장을 주도할 전망이다. [1]

본 논문에서는 Mobius플랫폼을 이용하여 Server, Gateway를 구성하고 장거리, 저속 통신 모듈인 LoRa를 활용하여 Test End node를 통해 Mobius Server에 데이터를 전송하는 것을 목표로 하였고, non oneM2M규격의 데이터를 Gateway에서 oneM2M표준의 json포맷으로 변환하는 convert module을 제작 테스트 하였다.

## II. 본 론

Mobius플랫폼은 일종의 미들웨어 이다. 물리적 기반인 통신 수단을 활용하여 인터넷등 다양한 통신 장치에 연결된 장치들을 관리하고 장치의 데이터를 저장하는 서버 기반의 플랫폼이다. 이처럼 장치와 어플리케이션의 통신 연결을 쉽고 편리하게 연결하여 활용할 수 있는 개방형 개발 환경으로 IoT 생태계를 구축할 수 있게 도와준다. [2]

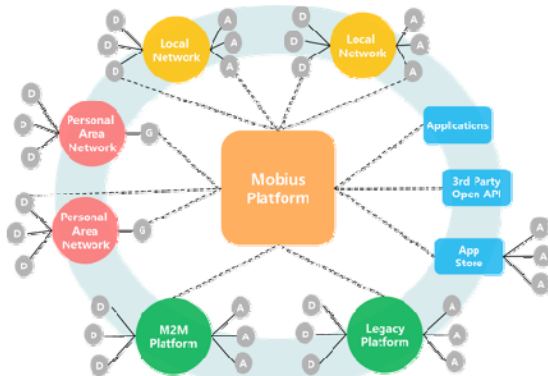


그림 1. Mobius platform 개념도[2]

Mobius플랫폼은 IoT관련 서비스를 제공하기 위해 Gateway, Device, End node의 정보를 관리하고 사용자들이 Device나 End node에 접근하여, 정보 검색, 제어, 인증, 사용자 관리등의 서비스를 제공하여 어플리케이션을 통해 정보를 제공하기 위한 플랫폼이다. 이러한 환경을 구축하기 위해 Mobius 플랫폼은 node.js로 개발되어 있다.

node.js는 서버환경에서 작동하는 자바스크립트로서 구글의 크롬 V8 자바스크립트 엔진을 기반으로 한 고성능 네트워크 서버이다. Single Thread 기반의 Event Loop를 기반으로 하고 있으며, File, Network 등에 대해서 비동기 IO 처리를 하는 서버 미들웨어이다. 최근 node.js 가 많은 주목을 받고 있는데, Paypal이나 Groupon과 같은 굴

지의 서비스 기업들이 내부 서버 플랫폼을 node.js로 전환하는 것이 계기가 되고 있다. Node.js는 일반적으로 성능이 매우 빠른 고성능 서버로 알려져 있다.

이 고성능은 자바스크립트등에서 오는 것이 아니라, node.js의 기본적인 구조인 Single Thread 기반의 비동기 IO 처리에서 온다. 하나의 스레드가 request를 받으면, 처리를 하고, File IO나 Network 처리 (데이터 베이스 접근)등이 있을 경우에는 IO 요청을 보내 놓고, 작업을 처리하다가, IO 요청이 끝나면 이벤트를 받아서 처리하는 이벤트 방식을 사용한다.

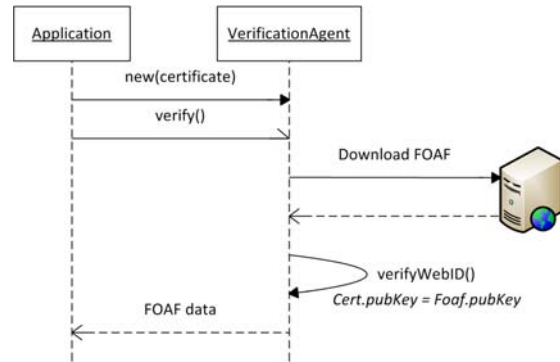


그림 2. node.js의 이벤트 처리방식

이로 인해서, CPU가 IO 응답을 기다리는 시간이 필요 없고, 대부분의 연산 작업에 사용되기 때문에 높은 효율성을 가질 수 있으며, 특히 하나의 Thread로 여러 개의 요청을 처리하는 구조로 되어 있기 때문에, C10K 문제를 처리할 수 있는데 최적화 되어 있다.[3]



그림 3. Mobius platform 개요

Mobius플랫폼은 oneM2M 표준에서 요구하는 Http, MQTT, CoAP 등 다양한 프로토콜과 Open API를 제공한다. 계층적인 리소스 구조를 가지고 있으며, Http RESTful로 Mobius서버에 Request/Response를 요청한다. 최근에 공개되어 배포되는 Mobius플랫폼은 Http, MQTT를 지원한다. Mobius플랫폼은 Http Open API를 위한 Http서버와 MQTT 지원을 위한 MQTT 서버로구성되어 있으며, 리소스 저장을 위한 DB는 MySQL DBMS를 사용한다. node.js, MQTT, MySQL은 모두 무료로 사용할 수 있는 공개 툴이다. 특히 node.js는 매우 활발한 개발자들의 활동을 통해 끊임없이 버전업되고 있으며 현재 약 30여만개의 모듈이 공개 배

포되고 있다.



그림 4. Mobius Server install 순서

Mobius서버를 설치하기 위해서는 nodejs, MQTT, MySQL이 설치되어야 한다. 설치 순서는 먼저 MySQL을 설치하고 MQTT Server를 설치한 뒤 node.js를 설치한다. node.js는 nodejs.org에서 다운받아 설치할 수 있다. 다양한 OS플랫폼을 지원하므로 쉽게 설치 운영이 가능하다.



그림 5. nodejs.org

MQTT는 센서,장치간의 기기들을 연결하기 위한 프로토콜이다. 이 프로토콜의 설계의도는 리소스 점유를 최소화하고 저성능의 네트워크의 장애와 단절에 대비한다. 또 단순하고 미니멀한 Pub/Sub 메시지 체제를 갖추고 있으며 오버헤드를 최소화하기 위해서 가장 작은 메시지 사이즈를 2byte로 정의 하였다. 신뢰도 보장을 위한 QoS레벨을 제공하여 pub/Sub의 메시지 전송을 보장한다. MQTT는 HTTP처럼 TCP/IP 네트워크 상에서 동작한다. 사용하기 위해서는 디바이스등에 반드시 TCP/IP물리적 장치가 설치되어 있어야 한다.



그림 6. mqtt.org

mysql.com사이트에 접속한 뒤 MySQL Community Server를 다운받아 사용자의 운영체제에 맞게 설치를 한다. 설치 중 반드시 root암호 설정을 해야 하는데 이것은 만들어진 database에 접근을 위한 것으로 MySQL Workbench등으로 database에 접근할 때 사용하기 위함이다. MySQL을 사용하기 위해서는 SQL에 대한 사전지식이 반드시 필요하다.



그림 7. mysql.com

서버가 설치되고 시스템을 재부팅하면 서버는 node.js, MQTT, MySQL이 미들웨어 형태로 구동되고 있다. MySQL Workbench로 mobiusdb를 만들기 위해 KETI에서 제공하는 sql파일을 import하여 oneM2M표준 database를 구성한다.

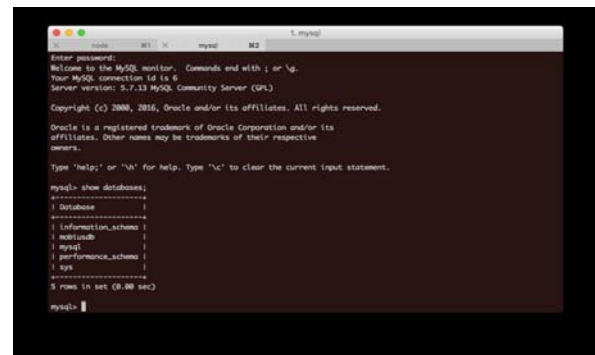


그림 8. mobiusdb table

기본적인 Server의 설정이 끝나면 Mobius Server플랫폼이 설치되어야 하고, 필수적으로 설치되어야 하는 모듈도 반드시 설치 되어야 한다. 필요한 모듈은 package.json에 정의되어 있다. node.js로 코딩된 Server플랫폼은 별도의 수정이 없이 즉시 실행이 가능하다. 서버는 HTTP기반의 RESTful형태의 API를 지원하므로 API 인터페이스 정의를 참고하여 CRUD요청을 할 수 있다.

Gateway로 사용되는 RaspberryPI에는 node.js가 설치되고 별도의 설정없이 즉시 사용이 가능하다. Gateway에는 &Cube와 TAS가 설치되고 서버와의 연결을 위해 반드시 환경설정으로 통해 서버의 주소, MQTT 서버의 주소,

MySQL서버의 주소등을 설정해야 한다. 기본 설정이 끝나면 서버와 gateway를 구동하여 정상적으로 통신이 이루어지는지 확인해야 한다.



그림 9. mobius server connection

### III. 실험

본 논문에서는 TAS와 통신을 위한 end device의 통신모듈을 RoLA를 사용하였다. 로라(LoRa)란 Long Range의 약어로서 대규모 저전력 장거리 무선통신 기술로 대기 전력이 적고 모듈 가격이 저렴하여 스마트 시티와 옥외 등에서 쓰이는 기술이다. 초고속, 광대역 네트워크를 필요로 하는 기존의 스마트 기기 연결 환경과는 다르게 별도의 기지국이나 중계 장비 없이 기기에 칩셋을 올려 저전력으로 소규모 데이터를 주고받고 있으며, 보안 양방향 통신, 이동성, 현지화 서비스와 같은 인터넷에서 요구되는 사항을 목표로 사용자 및 개발자에게 원활한 상호 운용성을 제공한다.

이번 실험에서 사용된 PLM100은 호스트 CPU(MCU)와 UART 통신으로 상호 연결할 수 있으며, UART Serial Interface 기반으로 최적화된 text command/response 인터페이스를 제공한다. 이 인터페이스를 이용하여 호스트는 PLM100 설정을 할 수 있으며, 무선 데이터 전송 및 수신할 수 있다. 무선 데이터 전송 시 적용된 PLM100 MAC 프로토콜은 IEEE802.15.4e MAC 기반으로 구현되어 있어, LoRa를 사용한 저전력 장거리 양방향 통신이 가능하게 한다.

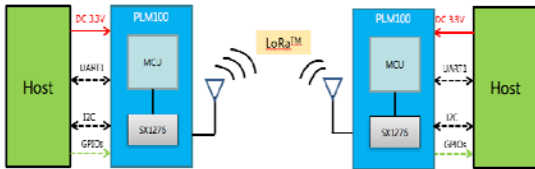


그림 10. mobius server connection

또한, 1대 1 혹은 1대 N 접속을 위한 Connectivity 설정이 가능하며, 이 설정 값은 PLM100 내의 지정된 별도의 Flash 영역에 저장되어 전원 ON/OFF 시에도 설정이 유지된다. 다양한 Application의 요구사항에 맞추어(저전력,

데이터 전송속도, 도달 거리 및 전송 지연 등) MAC 및 LoRa 설정을 할 수 있어 사용자의 요구에 맞는 Application 구현 및 성능을 가능하게 한다.

원활한 인터페이스를 위해 LoRa모듈과 EV kit을 활용하여 USART통신을 구현하고, 115200bps의 속도로 통신을 하도록 하였다. 다만 LoRa간 통신은 최대 5400bps이므로 내부통신의 속도는 외부통신과는 연관성이 없다.

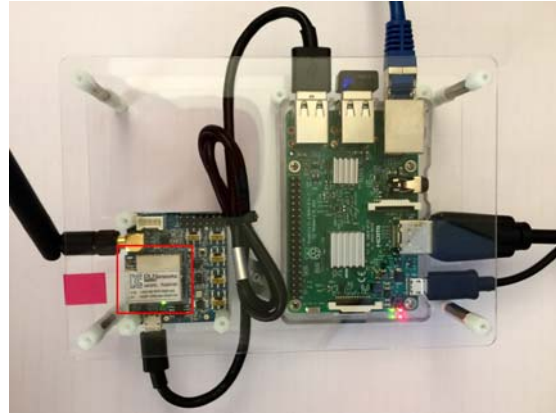


그림 11. mobius server connection

그림 11은 RaspberryPI와 LoRa로 구성된 Gateway이다. Gateway에서는 end node에서 전송되는 데이터를 수신 후 oneM2M 표준 포맷으로 변환한 뒤 Server로 데이터를 전송하는 역할을 한다. &Cube라는 Gateway용 소프트웨어와 TAS라고 불리는 end node제어용 소프트웨어로 구성되어 있다. 본 논문에서는 TAS를 직접 제작하여, LoRa에서 전송되는 non onM2M 규격의 데이터를 oneM2M규격으로 컨버전하여 Mobius server에 전송 하도록 하였다.

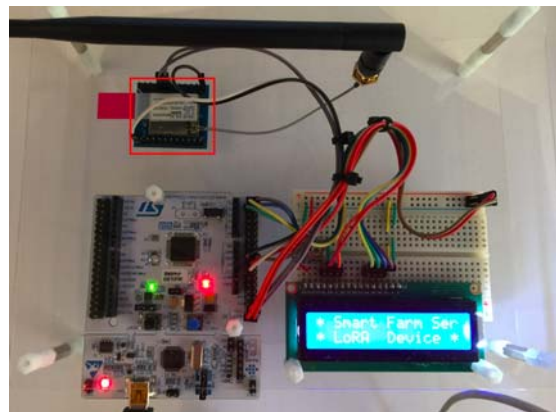


그림 12. mobius server connection

테스트용 end node는 5초 간격으로 데이터를



전송한다. Cortex-M4기반의 임베디드 보드에 LoRa를 연결하여 SF7, 5.4kbps의 속도로 약 70바이트의 데이터를 전송한다. 전송 후 성공여부에 대한 결과는 사용자가 지정한 retry횟수에 의거하여 평가하고 이를 다시 LoRa모듈로 전송 확인한다.

### 참고문헌

- [1] 사물인터넷 수요 및 시장동향, NIA 한국정보화진흥원, 2015
- [2] Mobius Installation Guide v1.0 2015.8
- [3] <http://bcho.tistory.com/876>

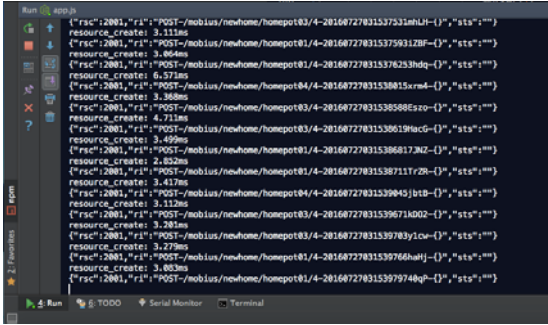


그림 13. Mobius server 데이터전송

그림 13은 Mobius서버에서 수신한 end node의 데이터 리스트이다. 데이터가 전송될 때마다 데이터 처리 시간과 전송된 container목록을 디스플레이 한다. 약 2주간 연속 테스트를 통해 end node에서 발생한 non oneM2M 데이터가 정상적으로 mobius server에 전송되는 것을 확인 하였다.

### IV. 결 론

최근 저렴한 장거리, 저속통신 모듈인 LoRa에 대한 관심이 커지고 있다. SK에서는 전국에 LoRa 망을 구축하여 디바이스 개발사를 지원하고 있고, 이를 통해 새로운 수익모델의 창출을 기대하고 있다.

국내, 국외에서 수십억개의 디바이스가 연결되고 또 다양한 통신모듈을 통해 데이터가 전송되는 환경을 고려하면 향후 저비용, 장거리 통신용 모듈의 개발이 계속될 것으로 예상된다. LoRa모듈은 장거리 통신의 조건을 만족하지만 저속으로 인해 응답성이 떨어지는 단점을 가지고 있다. 물론 CDMA나 LTE를 활용하는 방안도 있지만 고비용으로 인해 사용자들의 활용이 미비한 편이다.

따라서 장거리, 중속의 조건을 만족하는 통신 모듈의 개발이 필요할 것으로 사료되며 이는 장거리에서 현장의 상황 데이터를 즉시 받아볼 수 있을 것으로 예상된다.