

OM2M Release 2 오픈 소스 설치 가이드 및 예제

2018년 5월

경북대학교 통신프로토콜연구실

김소용

(thdyd324@gmail.com)

요 약

최근 4차 산업혁명이 이슈가 되면서 사물 인터넷(Internet of Things)이 그 대표 기술 중 하나로 주목받고 있다. 본 문서는 이 사물 인터넷에 대하여 알아보고 이를 구현하기 위한 표준의 하나인 oneM2M에 대해 설명한다. 또한 oneM2M의 표준규격들을 살펴보면서 각 특징을 알아보고 이를 오픈소스 OM2M을 이용하여 Raspberry Pi에 구현한다 그리고 OM2M에서 제공하는 sample plugin 예제를 통해 oneM2M의 동작 구조를 살펴본다.

목 차

1. 서론	3
2. oneM2M의 구조	5
2.1 1차 표준규격	7
2.2 2차 표준규격	8
3. OM2M 설치 및 사용법	9
3.1 OM2M	9
3.2 OM2M 실행환경 및 설치법	10
3.3 OM2M 다운로드 및 설치	13
3.4 OM2M 예제 테스트	17
4. 결론	23
참고 문헌	23

1. 서론

사물 인터넷이란 사람, 사물, 공간, 데이터 등 모든 것이 인터넷으로 서로 연결되어, 정보가 생성, 수집, 공유, 활용되는 초 연결 인터넷을 의미한다. 사물 인터넷은 자율 판단하는 지능을 보유하고 수백 억 이상의 사물을 관리하는 서비스 규모를 보이며 의미 기반의 모니터링 및 자율 제어기능을 가지고 있다. 이러한 기능들은 이미 존재하거나 향후 등장할 상호 운용 가능한 정보 통신 기술과 활용하여 다양한 실제 및 가상 사물 간의 상호 연결을 통해서, 진보된 서비스를 제공할 수 있다. 이를 위해 많은 기업과 표준화 기구에서 표준화를 위한 노력을 보이고 있다.

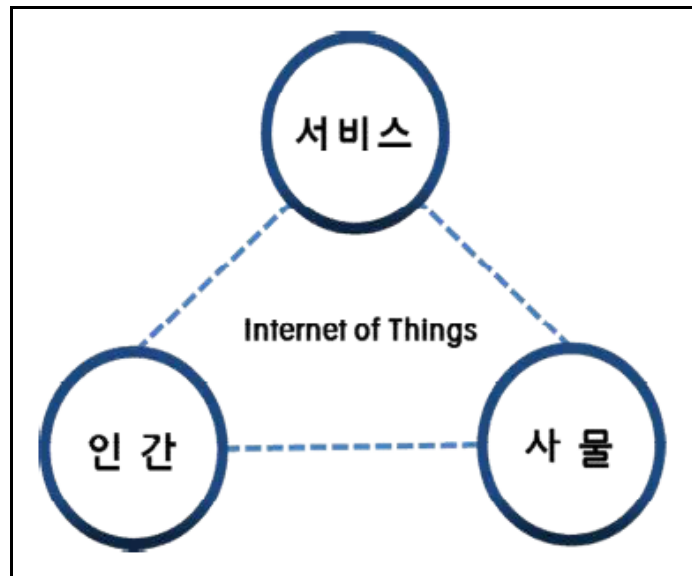


Figure 1. 사물인터넷 구조

oneM2M은 에너지, 교통, 국방, 공공서비스 등 산업별로 종속적이고 폐쇄적으로 운영되는, 파편화된 서비스 플랫폼 개발 구조를 벗어나 응용서비스 인프라(플랫폼) 환경을 통합하고 공유하기 위한 사물인터넷 공동서비스 플랫폼 개발을 위해 발족된 표준화 단체이다. oneM2M은 2015년 1월 요구사항, 용어정의, 아키텍처 등 10개의 표준규격을 포함하는 1차 규격을 발표하였고 현재 표준확산에 주력하는 사실상 표준화 단체로 성장하였다. 현재 많은 기업에서 oneM2M을 이용하여 서비스 창출 및 제품 개발을 하고 있으며 그 입지를 다지고 있다.

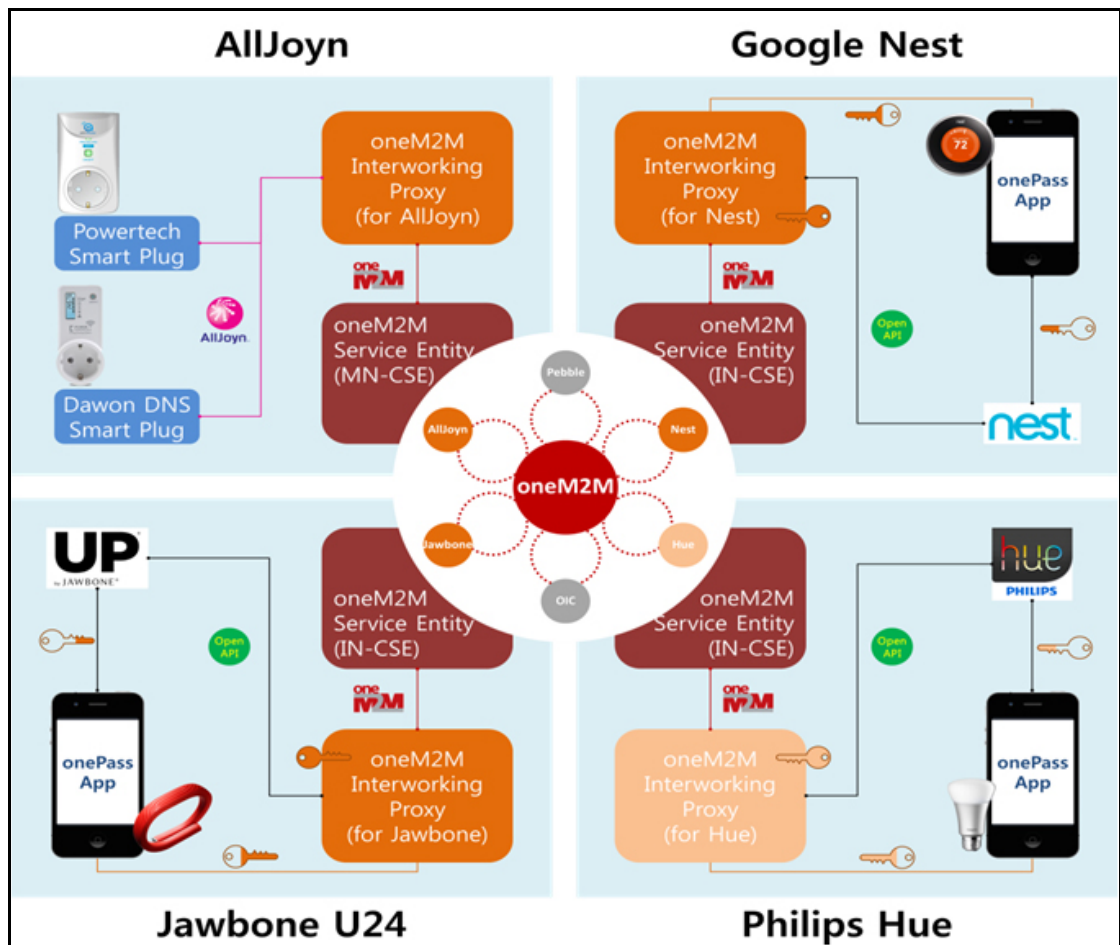


Figure 2. oneM2M 표준을 이용하는 플랫폼들

본 문서에서는 oneM2M의 구조에 대해 알아보고 1차 표준규격과 2차 표준규격에서 제시된 특징들을 살펴본다. 또한 통합 개발 환경인 이클립스(Eclipse)를 이용하여 오픈소스 OM2M을 Raspberry Pi에 설치하고 이를 이용하여 oneM2M 표준을 구현한다.

2. oneM2M의 구조

oneM2M은 기능에 따라 애플리케이션 전용 노드(ADN: Application Dedicated Node), 애플리케이션 서비스 노드(ASN: Application Service Node), 중간 노드(MN: Middle Node) 및 인프라 노드(IN: Infrastructure Node)로 구성되어 있다.

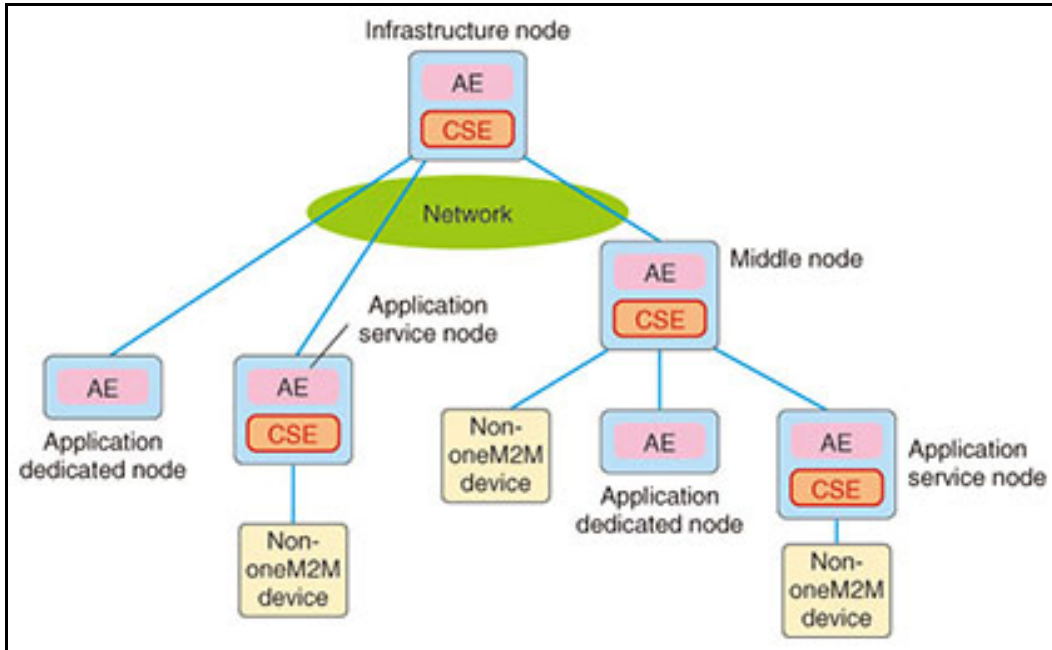


Figure 3. oneM2M의 노드 구조

애플리케이션 서비스 노드는 M2M Application 뿐만 아니라 공통의 서비스 기능을 포함하는 일반 노드이고 애플리케이션 전용 노드는 M2M Application을 포함하는 M2M 디바이스로 M2M 서비스만을 포함하는 제한된 기능을 가지는 디바이스이다. 중간 노드는 디바이스 노드들과 네트워크 인프라 노드를 연결해주는 게이트웨이 역할을 하는 노드이고 인프라 노드는 네트워크 인프라에 위치해 M2M 서비스를 제공하는 노드이다.

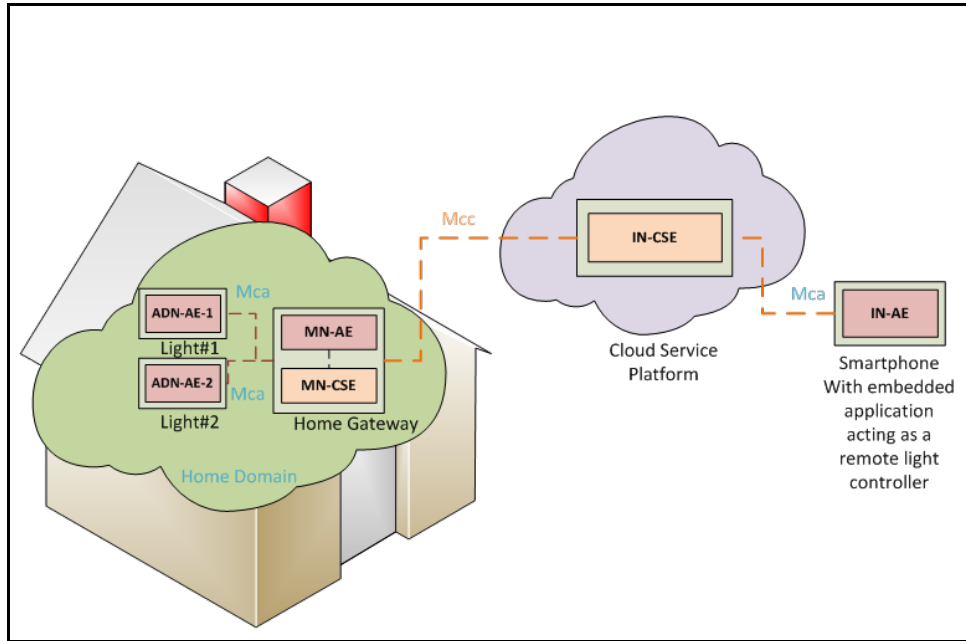


Figure 4. oneM2M 구조 레퍼런스 모델

이 노드들을 구성하는 엔티티는 세 가지 계층으로 이루어져 있으며, 각 엔티티의 기능은 다음과 같다.

구분	내용
애플리케이션 엔티티 (Application Entity)	<ul style="list-style-type: none"> • 애플리케이션 엔티티는 End-to-End 사물인터넷 솔루션을 위한 애플리케이션 로직을 제공함 - 화물추적, 원격모니터링, 원격점검 및 제어 등
공통 서비스 엔티티 (Common Service Entity)	<ul style="list-style-type: none"> • 공통 서비스 엔티티는 사물인터넷의 다양한 애플리케이션 엔티티들이 공통적으로 사용할 수 있는 기능들로 이루어진 플랫폼임
네트워크 서비스 엔티티 (Network Service Entity)	<ul style="list-style-type: none"> • 네트워크 서비스 엔티티는 공통 서비스 엔티티에 네트워크 서비스를 제공함. 3GPP(3rd Generation Partnership Project)네트워크 연동 중심으로 되어 있음. - 장치관리, 위치서비스, 장치 트리거링 등

Figure 5. oneM2M 엔티티

2.1 1차 표준규격

앞서 언급하였던 oneM2M의 구조는 1차 표준 규격에서 처음으로 정의되었다. 1차 표준 규격은 네트워크 아키텍처와 아키텍처를 구성하는 엔티티 및 공통 서비스 기능(CSF, Common Service Function)과 이를 제공하기 위한 공통 서비스 계층에서의 레퍼런스 포인트를 정의한다.

공통 서비스 기능은 사물인터넷 서비스 애플리케이션에서 자주 사용되는 기능을 정의한 것으로 데이터 저장/공유, 장치 관리, 그룹 관리, 구독/통지(Subscription/Notification), 위치 정보, 과금 등의 기능을 포함하며, 보안 기능은 기본적인 인증, 접근 제어 등의 기능을 제공한다.

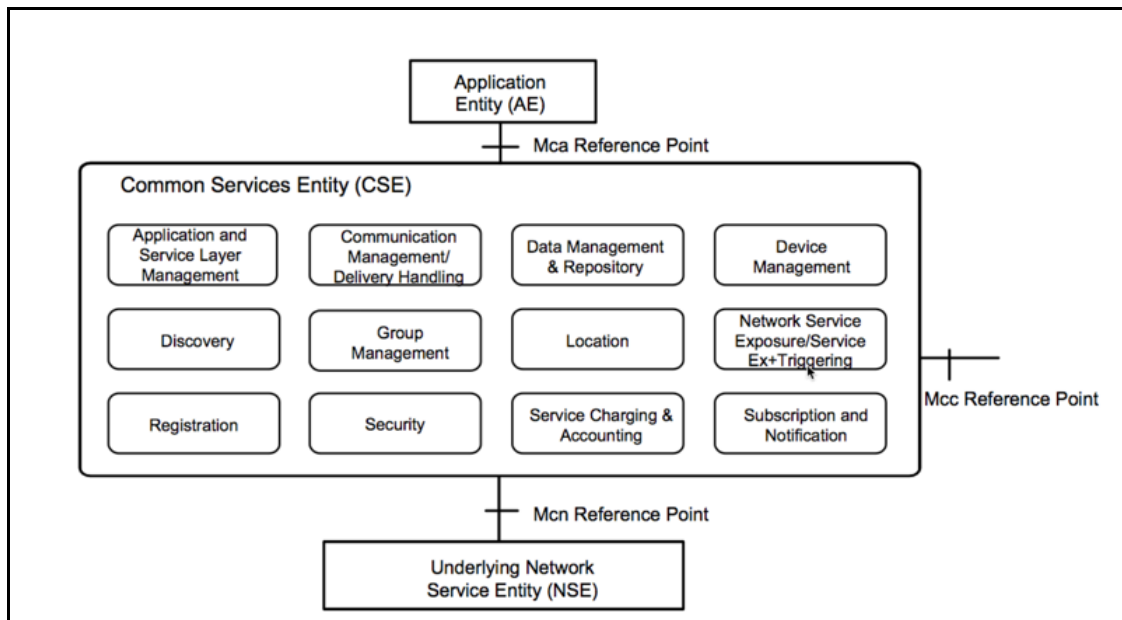


Figure 6. 공통 서비스 기능(CSF, Common Service Function)

또한, oneM2M 코어 프로토콜 메시지(primitive)는 CoAP, HTTP 및 MQTT 프로토콜 메시지를 통해 전송된다. oneM2M의 코어 프로토콜은 향후 추가 프로토콜 바인딩을 지원할 수 있도록 특정 메시지 프로토콜에 종속성을 가지지 않도록 개발되었다.

2.2 2차 표준규격

2차 표준 규격은 다양한 산업 사물인터넷 플랫폼 및 네트워크 연동을 주 목적으로 하기 위해 발표되었다. 사물인터넷 연동으로는 AllJoyn, OCF(Open Connectivity Foundation) 및 Lightweight M2M 기술과의 연동 규격을 제공한다.

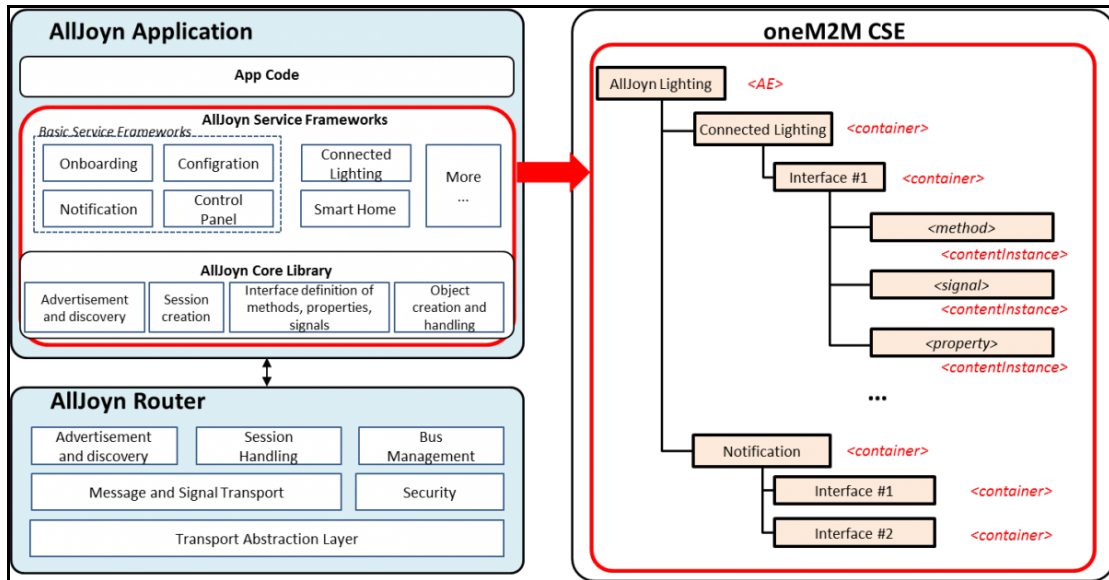


Figure 7. AllJoyn과 oneM2M의 상호 연동 구조

네트워크 연동으로는 3GPP Rel-13 네트워크와 연동을 위한 트래픽 패턴 설정(Traffic pattern Configuration) 기능을 정의하고 있으며 3차 표준규격에 모니터링 등의 연동 기능을 추가하기 위한 기술 보고서 작업을 지속하고 있다.

이전의 1차 표준규격에서는 가전 제어 및 센싱 정보를 교환하기 위해 사전에 애플리케이션 간 정의한 데이터 모델로 container 및 content Instance 자원 타입을 이용했다.

이에 비해 2차 표준규격에서는 oneM2M 플랫폼을 이용하는 모든 애플리케이션이 표준에 정의된 가전 디바이스 데이터 모델(Appliances device data model)을 사용함으로써 가전제품 제조사 및 애플리케이션 개발자 간에 별도의 데이터 모델을 정의하는 번거로움을 없애고 제품과 애플리케이션 간의 호환성을 보장한다.

3. OM2M 설치 및 사용법

3.1 OM2M

OM2M은 ETSI-M2M(European Telecommunication Standards Institute-Machine To Machine)표준을 기반으로 한 M2M(Machine To Machine) 상호 운용을 지원하는 오픈 소스 서비스 플랫폼을 제공하고, 개방형 인터페이스인 RESTful 접근법을 따르므로 기본 네트워크와 독립적으로 서비스 및 응용 프로그램을 개발할 수 있다. 또한 OM2M은 OSGi(Open Service Gateway initiative) 계층 위에 실행되는 모듈 방식 구조를 제안하여 플러그인을 통해 확장성을 높인다. OSGi란 네트워크상에 연결된 디바이스들이 다양한 서비스를 공유할 수 있도록 하는 자바 언어 기반의 동적 플랫폼을 말한다.

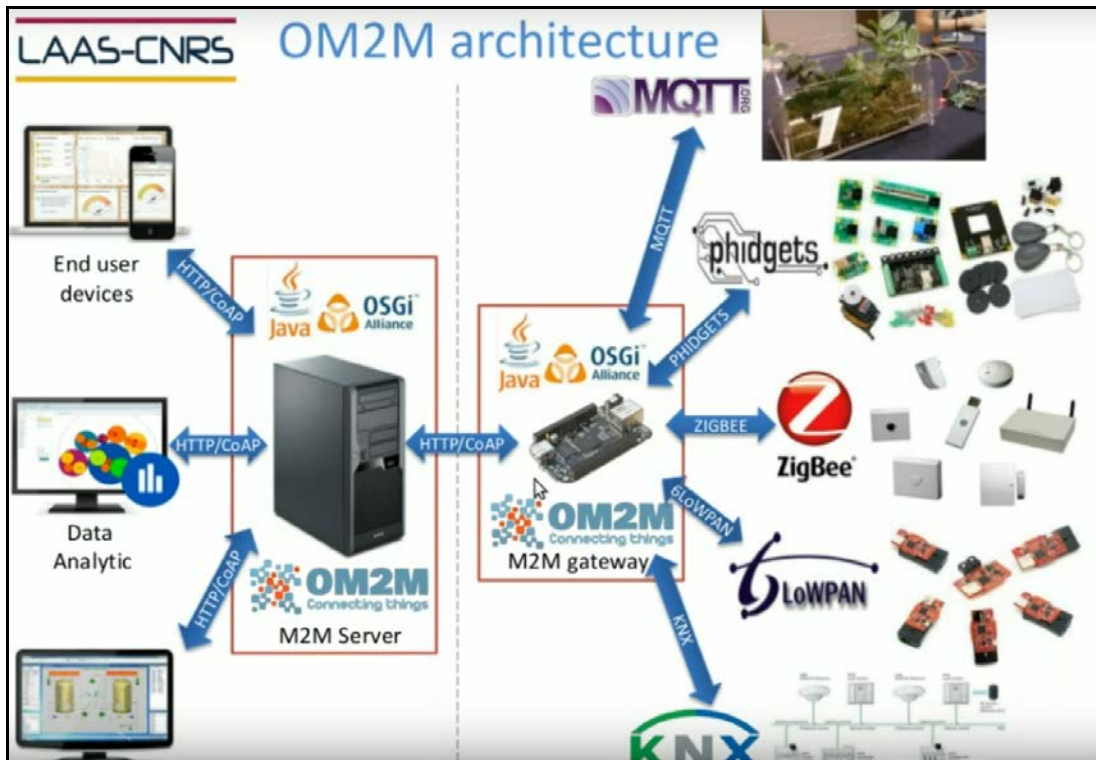


Figure 8. OM2M의 구조

OM2M은 2017년 8월 1.1.0 버전을 출시하면서 oneM2M의 2차 표준규격을 지원하고 동시에 MQTT 통신 바인딩기능, NoSQL MongoDB 연동기능 등이 추가되면서 다양한 활용방식을 제공한다.

3.2 OM2M 실행 환경 및 설치법

본 문서의 실행환경으로는 서버 역할을 하는 OM2M의 IN-CSE를 Raspberry Pi 2 Model B에 설치하고 sample plugin 예제를 가지고 있는 MN-CSE를 같은 Raspberry Pi에 설치하여, IN-CSE에 연결하였다. 그리고 OM2M에서 제공하는 웹페이지를 클라이언트로 이용하였다.

OM2M의 설치법은 두 가지가 있는데 OM2M의 소스코드를 받아 컴파일하여 설치하는 방법과 이미 빌드 된 OM2M의 바이너리 파일을 받아 설치하는 방법이 있다. 본 문서에서는 전자인 소스 코드를 컴파일 하여 설치하는 방법을 설명한다.

먼저 Raspberry Pi에 이클립스를 설치하고 Git과 Maven plugin을 설치하는 과정을 설명한다.

터미널에 `sudo apt-get install eclipse` 명령어를 입력하여 이클립스를 설치한다. 이 후 이클립스를 실행하여 상단에 위치한 메뉴에서 Help -> Install New Software를 클릭하게 되면 Figure9와 같은 plugin install창이 생성된다.

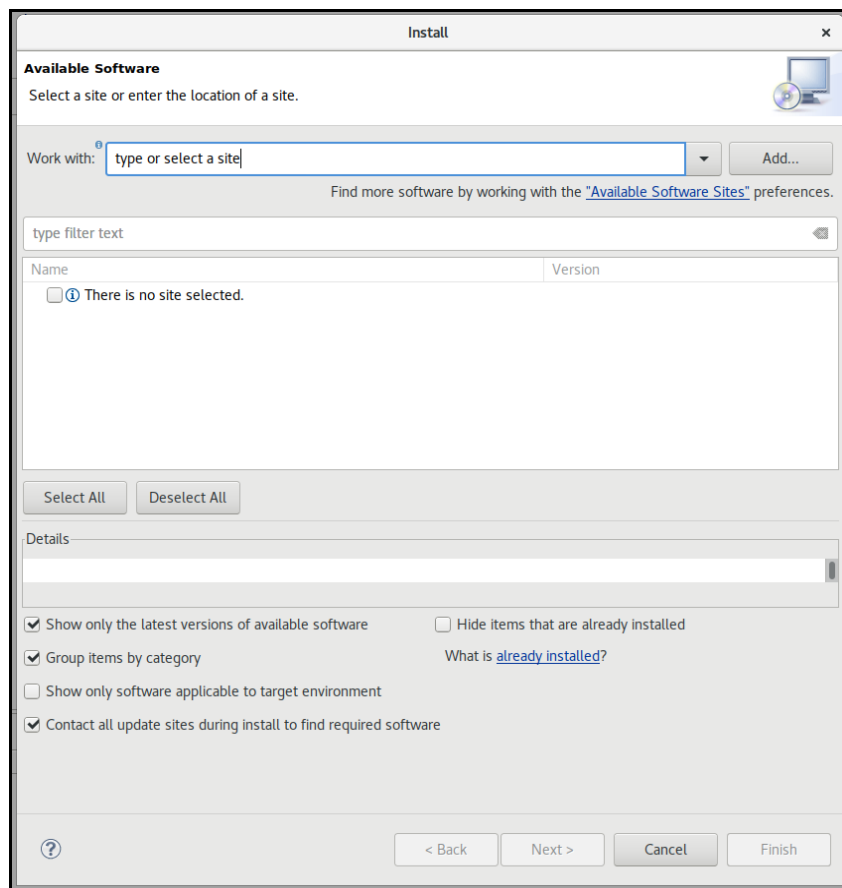


Figure 9. plugin install창

install창에서 Add를 클릭하면 Repository를 입력하는 창이 생성된다. location 부분에 아

래의 주소를 입력하고 OK를 클릭하면 Git plugin 목록을 install창에서 확인할 수 있다.

`http://archive.eclipse.org/egit/updates-2.1`

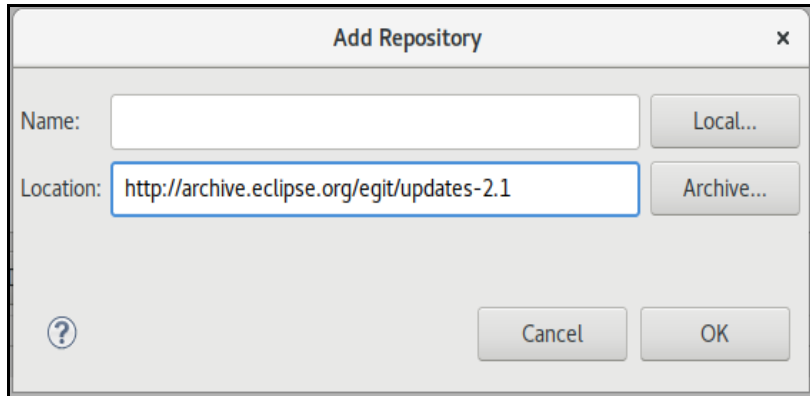


Figure 10. Git Repository 추가

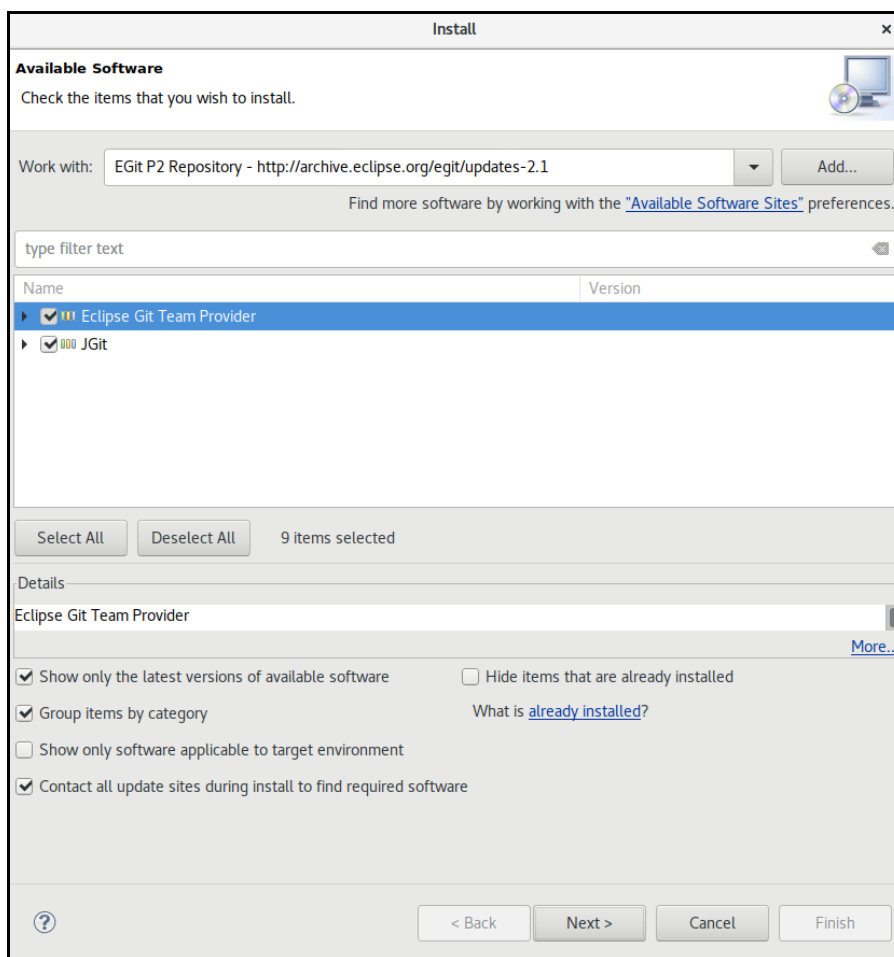


Figure 11. 설치할 Git plugin 목록

Next를 클릭하여 설치를 진행하게 되면 Figure12와 같은 플러그인 적용을 위한 메시지창

이 생성된다. Yes를 눌러 재시작을 하면 플러그인 적용이 완료된다.

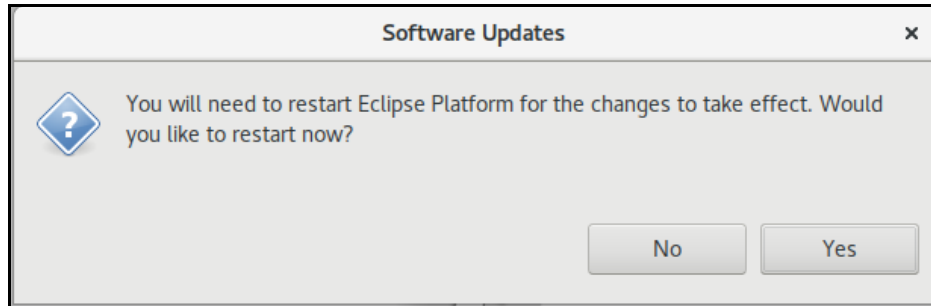


Figure12. 플러그인 적용

같은 방식으로 location 부분에 아래의 주소를 입력하고 Maven plugin을 설치한다.

<http://download.eclipse.org/technology/m2e/releases/1.4>

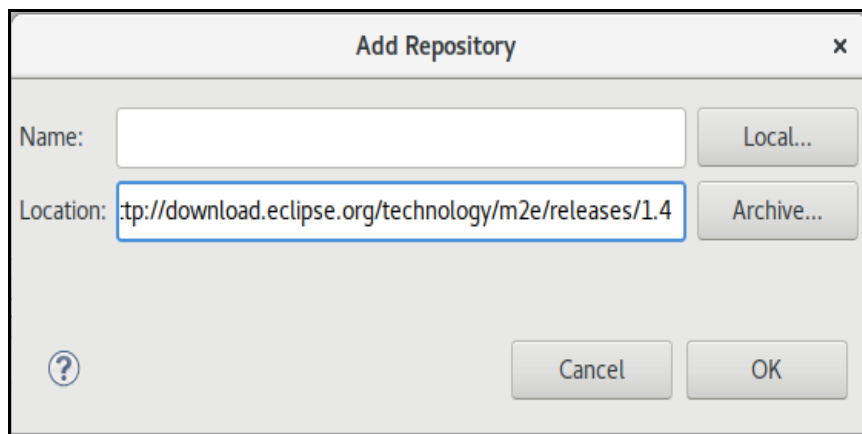


Figure 13. Maven Repository 추가

3.3 OM2M 다운로드 및 설치

이클립스 상단 메뉴에 Window -> Show View -> Other... 을 클릭하면 Figure14처럼 Show View 창이 생성되고 Git을 클릭하면 Git Repositories를 볼 수 있다. 이것을 선택한 채 OK를 클릭하면 Git Repository를 관리하는 뷰어가 이클립스에 생성된다.

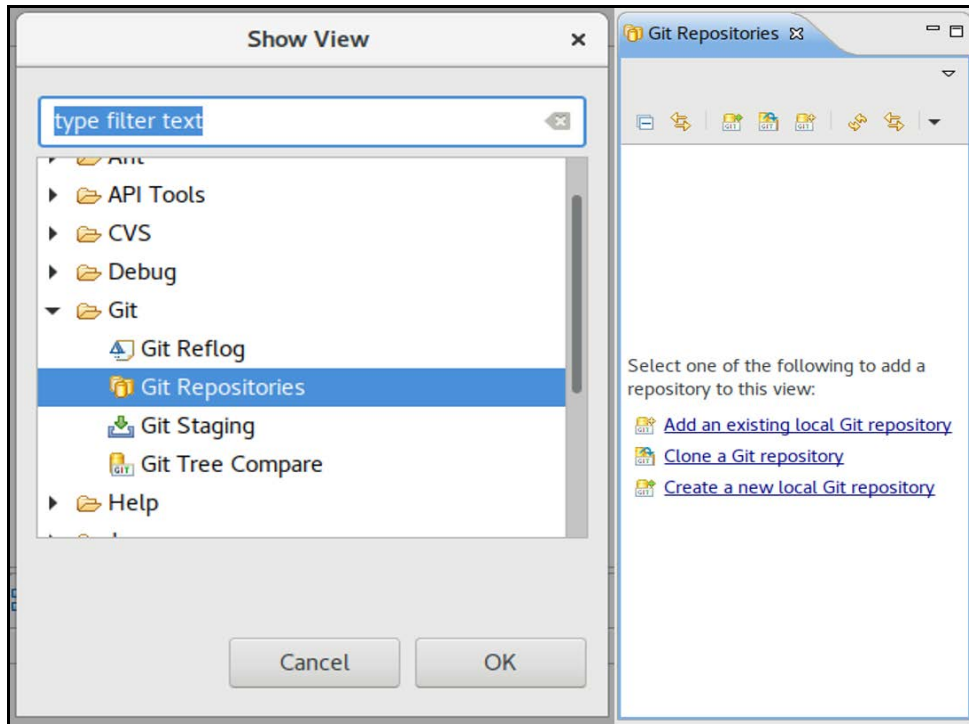


Figure 14. Show View 와 Git Repositories

Git Repositories 내부에 Clone a Git repository를 클릭하거나 상단에 Clone 아이콘을 클릭하여 URI 박스에 아래의 주소를 입력한다.

<https://git.eclipse.org/r/om2m/org.eclipse.om2m>

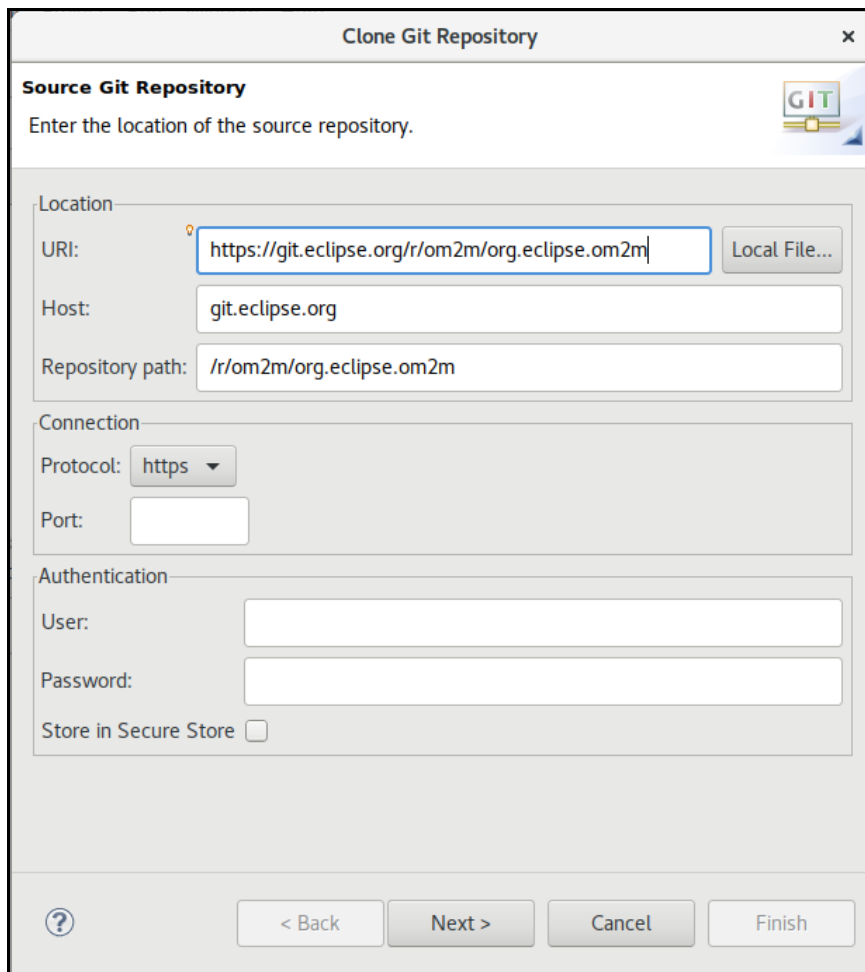


Figure 15. Clone Git Repository

Next를 클릭하여 Local Destination 설정 구간으로 오게 되면 Figure16과 같이 Projects 탭에 있는 Import all existing projects after clone finishes 옵션을 체크한다. 그리고 Finish를 클릭하면 해당 주소에서 Git Repositories에 org.eclipse.om2m의 이름을 가진 Repository를 추가하게 된다.

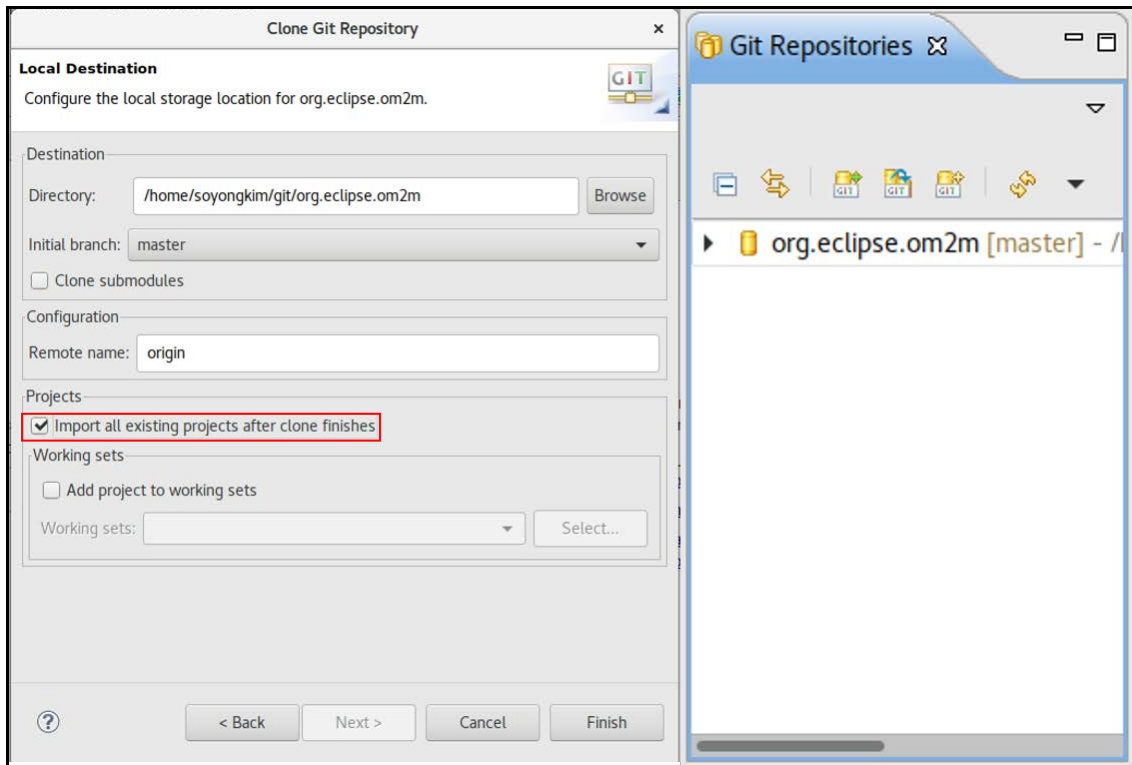


Figure 16. Local Destination 설정 및 org.eclipse.om2m 추가

Package Explorer창에 Figure16과 같이 org.eclipse.om2m 항목이 나타나지 않을 경우, org.eclipse.om2m을 클릭하면 Figure17과 같이 Working Directory를 확인할 수 있는데, 이것을 우 클릭하여 생성되는 메뉴에서 Import Projects를 클릭하면 org.eclipse.om2m 항목을 Package Explorer에 생성할 수 있다.

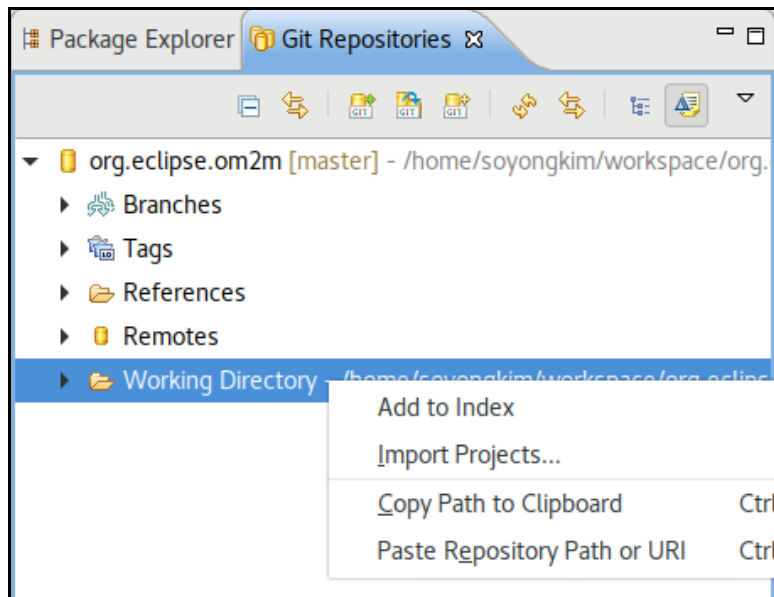


Figure 17. Working Directory

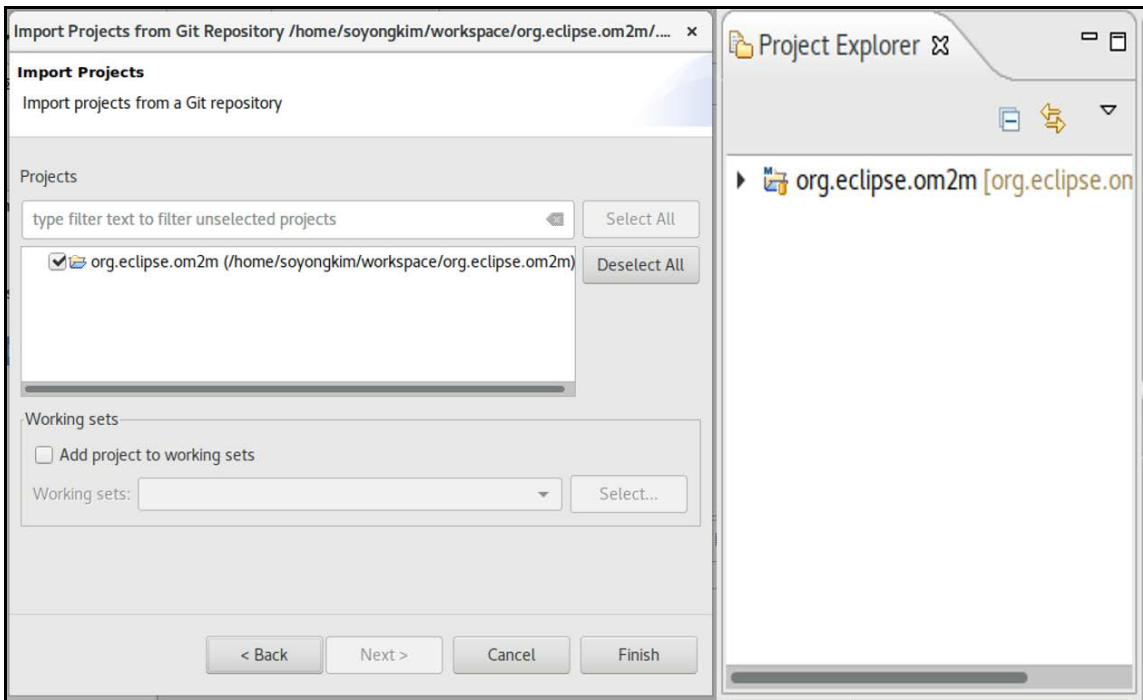


Figure 18. Repository를 통한 Project 생성

Package Explorer에 위치한 org.eclipse.om2m 디렉토리를 우 클릭 한 후 Run As > Maven Install을 클릭하면 빌드가 시작되고 Build Success가 콘솔창에 표시되면서 빌드가 완료 된다.

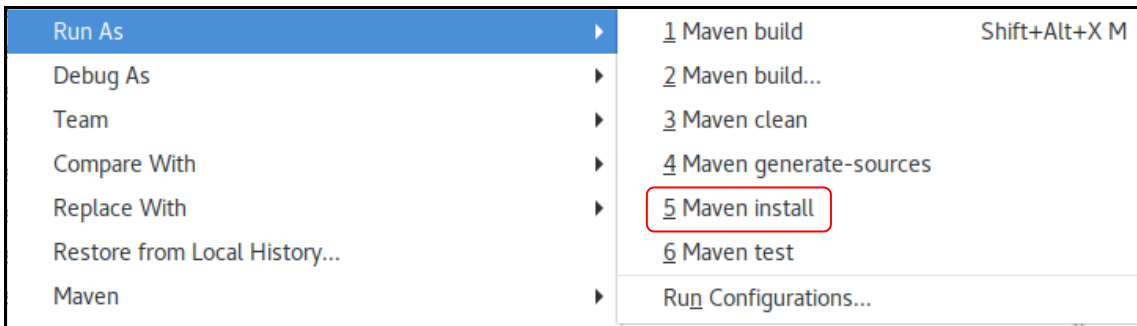


Figure 19. maven install 명령 실행

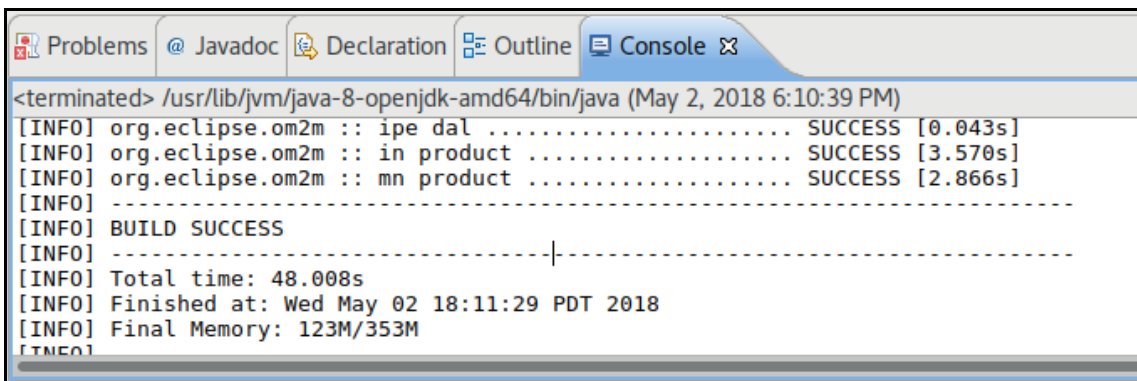
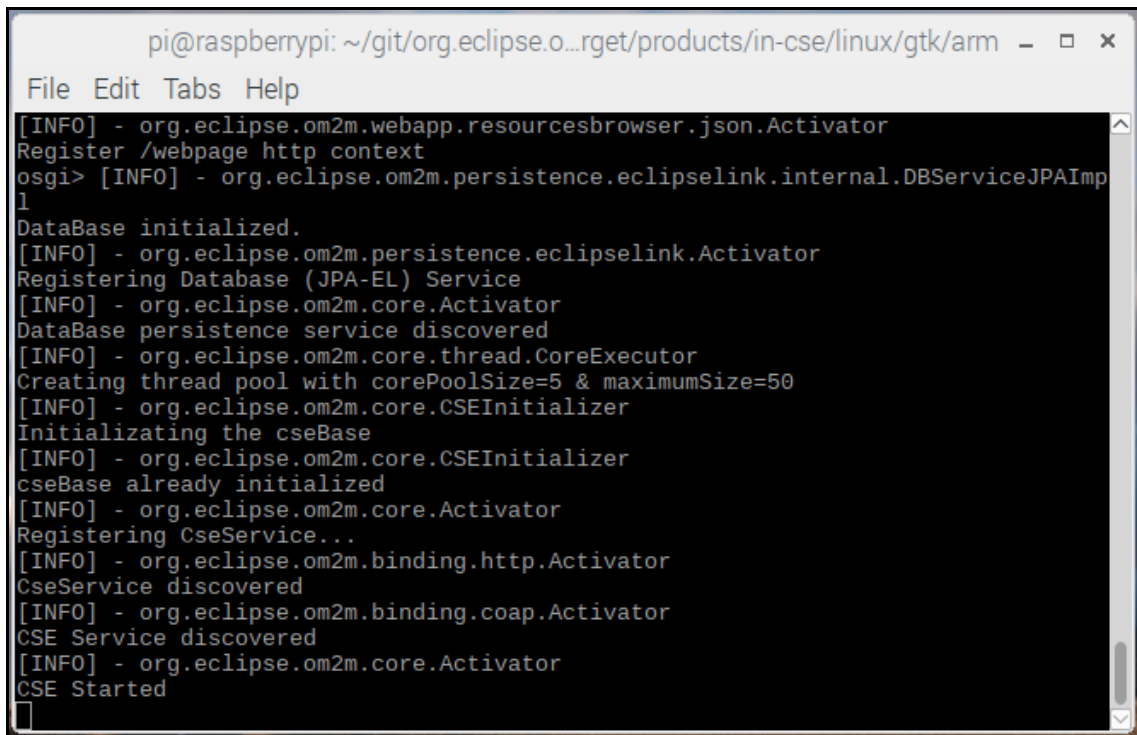


Figure 20. 빌드 완료 상태 창

3.4 OM2M 예제 테스트

빌드가 완료되면 org.eclipse.om2m/org.eclipse.om2m.site.in-cse폴더 안에 target 폴더가 생성된다. 터미널을 실행하여 target/products/in-cse/linux/gtk/arm폴더로 들어가 start.sh 파일을 bash start.sh 명령으로 실행시켜주면 Figure21처럼 IN-CSE가 실행된다.



```
pi@raspberrypi: ~/git/org.eclipse.o...rget/products/in-cse/linux/gtk/arm - □ ×
File Edit Tabs Help
[INFO] - org.eclipse.om2m.webapp.resourcesbrowser.json.Activator
Register /webpage http context
osgi> [INFO] - org.eclipse.om2m.persistence.eclipselink.internal.DBServiceJPAImpl
DataBase initialized.
[INFO] - org.eclipse.om2m.persistence.eclipselink.Activator
Registering Database (JPA-EL) Service
[INFO] - org.eclipse.om2m.core.Activator
DataBase persistence service discovered
[INFO] - org.eclipse.om2m.core.thread.CoreExecutor
Creating thread pool with corePoolSize=5 & maximumSize=50
[INFO] - org.eclipse.om2m.core.CSEInitializer
Initializing the cseBase
[INFO] - org.eclipse.om2m.core.CSEInitializer
cseBase already initialized
[INFO] - org.eclipse.om2m.core.Activator
Registering CseService...
[INFO] - org.eclipse.om2m.binding.http.Activator
CseService discovered
[INFO] - org.eclipse.om2m.binding.coap.Activator
CSE Service discovered
[INFO] - org.eclipse.om2m.core.Activator
CSE Started
```

Figure 21. IN-CSE 실행화면

이제 웹 브라우저를 실행하여 주소창에 127.0.0.1:8080/webpage를 입력하면 OM2M 웹 페이지를 볼 수 있고 id와 password에 admin을 입력한 뒤 로그인하면 OM2M의 리소스 트리를 볼 수 있다.

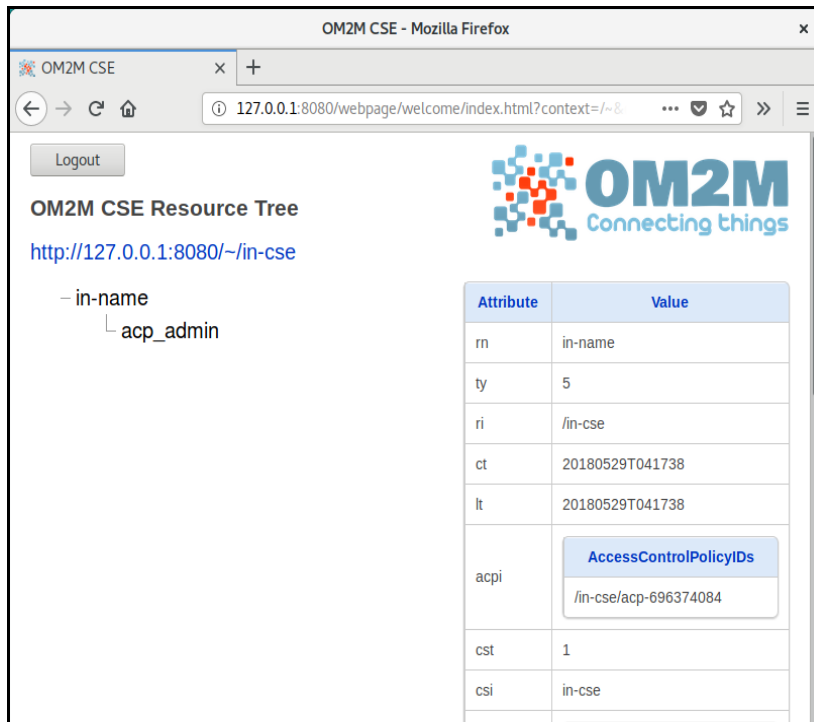


Figure 22. OM2M 웹 페이지

IN-CSE 리소스 하단에 MN-CSE 리소스를 등록하기 위해 터미널을 하나 더 실행하여 `org.eclipse.om2m.site.mn-cse/target/products/mn-cse/linux/gtk/arm` 폴더로 이동해 `bash start.sh` 명령을 하여 MN-CSE를 실행한다.

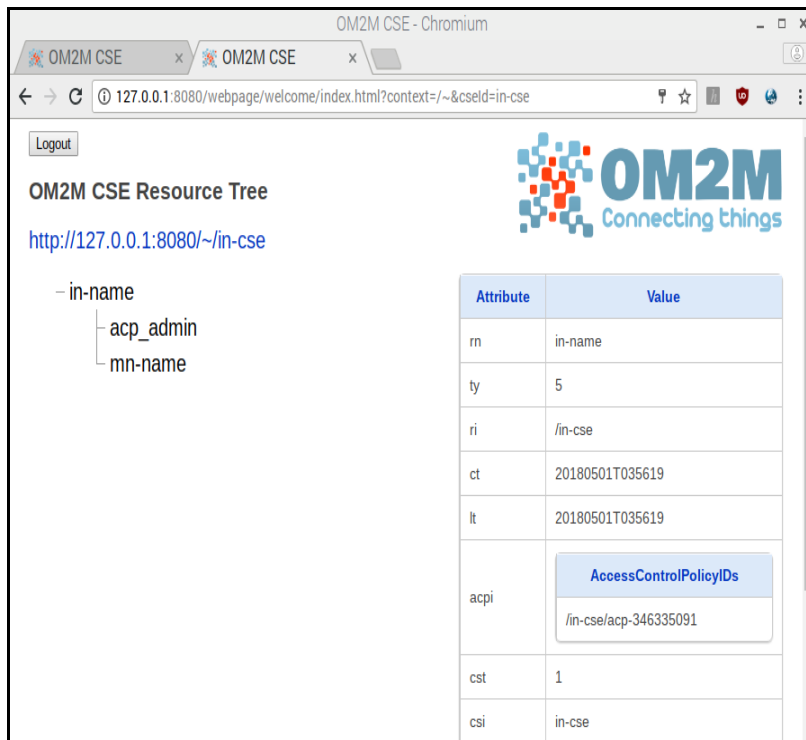


Figure 23. MN-CSE 리소스 추가

MN-CSE를 실행중인 터미널에서 `ss` 명령을 주면 현재 실행할 수 있는 플러그인의 목록이 출력된다.

```

pi@raspberrypi: ~/Downloads/MN-CSE
File Edit Tabs Help
14  RESOLVED  org.eclipse.jetty.continuation_8.1.16.v20140903
15  RESOLVED  org.eclipse.jetty.http_8.1.16.v20140903
16  RESOLVED  org.eclipse.jetty.io_8.1.16.v20140903
17  RESOLVED  org.eclipse.jetty.security_8.1.16.v20140903
18  RESOLVED  org.eclipse.jetty.server_8.1.16.v20140903
19  RESOLVED  org.eclipse.jetty.servlet_8.1.16.v20140903
20  RESOLVED  org.eclipse.jetty.util_8.1.16.v20140903
21  ACTIVE    org.eclipse.om2m.binding.http_1.0.0.20160708-0821
22  RESOLVED  org.eclipse.om2m.binding.service_1.0.0.20160708-0821
23  RESOLVED  org.eclipse.om2m.commons_1.0.0.20160708-0821
24  RESOLVED  org.eclipse.om2m.commons.logging_1.0.0.20160708-0821
Master=4
25  ACTIVE    org.eclipse.om2m.core_1.0.0.20160708-0821
26  RESOLVED  org.eclipse.om2m.core.service_1.0.0.20160708-0821
27  ACTIVE    org.eclipse.om2m.datamapping.jaxb_1.0.0.20160708-0821
28  RESOLVED  org.eclipse.om2m.datamapping.service_1.0.0.20160708-0821
29  RESOLVED  org.eclipse.om2m.interworking.service_1.0.0.20160708-0821
30  RESOLVED  org.eclipse.om2m.ipe.sample_1.0.0.20160708-0821
31  ACTIVE    org.eclipse.om2m.persistence.eclipselink_1.0.0.20160708-0821
32  RESOLVED  org.eclipse.om2m.persistence.service_1.0.0.20160708-0821
33  ACTIVE    org.eclipse.om2m.webapp.resourcesbrowser.xml_1.0.0.20160708-
0821
34  RESOLVED  org.eclipse.osgi.services_3.4.0.v20140312-2051
osgi>

```

Figure 24. MN-CSE의 플러그인 목록

플러그인 목록에서 30번에 해당하는 `org.eclipse.om2m.ipe.sample`을 실행하면 Figure25와 같이 `sample plugin`이 실행된다.

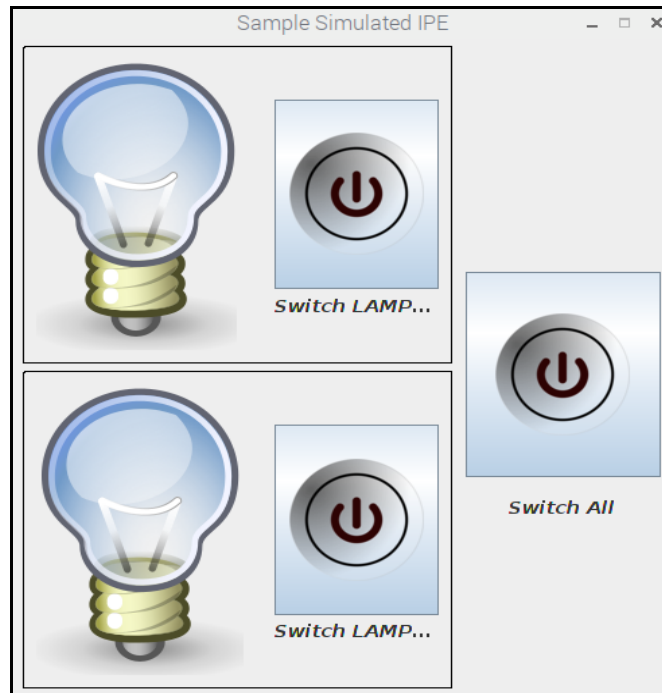


Figure 25. LAMP sample plugin

웹 브라우저에서 mn-cse 리소스를 클릭하고 csi Attribute의 value값인 /mn-cse를 클릭하면 mn-cse의 리소스 트리를 볼 수 있다.

OM2M CSE Resource Tree

<http://127.0.0.1:8080/~in-cse/csr-858183787>

- in-name

- acp_admin
 - mn-cse

Attribute	Value
ty	16
ri	/in-cse/csr-858183787
pi	/in-cse
ct	20180531T005358
lt	20180531T005358
acpi	<div style="border: 1px solid #ccc; padding: 5px;"> <p style="text-align: center; margin: 0;">AccessControlPolicyIDs</p> <p style="margin: 0;">/in-cse/acp-22681632</p> </div>
poa	<div style="border: 1px solid #ccc; padding: 5px;"> <p style="text-align: center; margin: 0;">Point Of Access</p> <p style="margin: 0;">http://127.0.0.1:8282/</p> </div>
cb	//om2m.org/mn-cse
csi	/mn-cse
rr	true

Figure 26. IN-CSE의 MN-CSE 정보

OM2M CSE - Chromium

OM2M CSE Resource Tree

<http://127.0.0.1:8080/~mn-cse>

Logout

– mn-name

- acp_admin
- acpae-560340890
- acpae-278184004
- acpae-39765429
- LAMP_0
- LAMP_1
- LAMP_ALL
- in-name

Attribute	Value
ty	5
ri	/mn-cse
ct	20180531T005356
lt	20180531T005356
acpi	<div style="border: 1px solid #ccc; padding: 2px;"> <p style="text-align: center; margin: 0;">AccessControlPolicyIDs</p> <p style="margin: 0;">/mn-cse/acp-412029881</p> </div>
cst	1
csi	mn-cse
srt	1 2 3 4 5 9 14 15 16 17 23
poa	<div style="border: 1px solid #ccc; padding: 2px;"> <p style="text-align: center; margin: 0;">Point Of Access</p> <p style="margin: 0;">http://127.0.0.1:8282/</p> </div>

Figure 27. MN-CSE의 리소스 트리

mn-name 트리 하단에 있는 LAMP_0를 클릭하면 “cin_임의의 숫자”형태를 가지는 LAMP_0의 컨테이너 객체를 확인할 수 있다. 이 컨테이너 객체를 클릭하면 Figure28과 같이 테이블 목록에 switchON을 확인할 수 있는데 이것을 클릭하면 Figure29와 같이 sample plugin에서 LAMP_0에 해당하는 전구가 켜지는 것을 확인할 수 있다.

OM2M CSE - Chromium

OM2M CSE Resource Tree

<http://127.0.0.1:8080/~mn-cse/cin-20771404>

mn-name

- acp_admin
- acpae-560340890
- acpae-278184004
- acpae-39765429
- LAMP_0
 - DESCRIPTOR
 - cin_20771404
 - DATA
- LAMP_1
- LAMP_ALL
- in-name

Attribute	Value
ty	4
ri	/mn-cse/cin-20771404
pi	/mn-cse/cnt-427981997
ct	20180531T012935
lt	20180531T012935
st	0
cnf	application/obix
cs	658

Attribute	Value
type	LAMP
location	Home
appld	LAMP_0
getState	/mn-cse/mn-name/LAMP_0/DATA/la
getState(Direct)	/mn-cse/mn-name/LAMP_0?op=getStateDirect&lampid=LAMP_0
switchON	/mn-cse/mn-name/LAMP_0?op=setOn&lampid=LAMP_0
switchOFF	/mn-cse/mn-name/LAMP_0?op=setOff&lampid=LAMP_0
toggle	/mn-cse/mn-name/LAMP_0?op=toggle&lampid=LAMP_0

Figure 28. LAMP_0의 컨테이너 객체

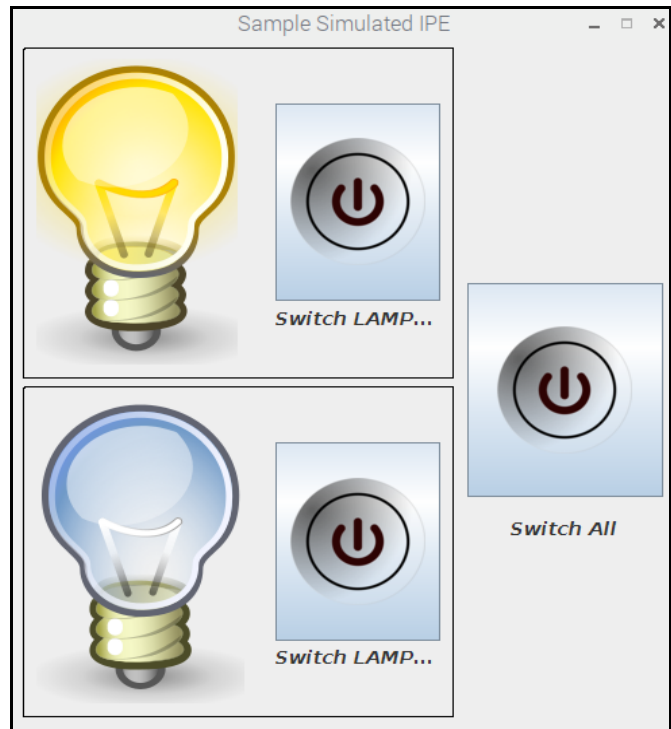


Figure 29. sample plugin의 LAMP_0 변화

4. 결론

지금까지 본 문서에서는 사물인터넷과 oneM2M 표준에 대하여 설명하였고 OM2M을 이용하여 oneM2M 표준을 직접 구현해보았다. oneM2M은 시장 적용 확대를 위해 산업체, 차량 등 도메인에 대한 기술 지원을 강화하고, 제품 개발 확대를 위한 가이드, 제품 프로파일, 시험 인증 규격 개발을 지속적으로 하고 있다. 또한 oneM2M은 블록 체인, 지능형 IoT, 가상화 등 최신 기술 트렌드를 지원하기 위한 논의가 본격적으로 진행하면서 좋은 전망을 보이고 있다.

OM2M은 홈페이지에 자세한 설명과 함께 쉬운 사용법으로 개발자라면 누구나 쉽게 구현할 수 있도록 되어 있어 접근성이 아주 좋아 추후 개발자들이 IoT 서비스를 개발할 때, 유용하게 쓰일 것으로 기대된다.

참고 문헌

- [1] 위키백과 OneM2M, <https://ko.wikipedia.org/wiki/OneM2M>
- [2] OM2M 오픈 소스 설치 가이드, <http://protocol.knu.ac.kr/tech/CPL-TR-14-03-OM2M.pdf>
- [3] 정승명, oneM2M Release 3 표준, IoT 표준/기술 동향 2017-9호
- [4] 백승범, IoT 오픈플랫폼 기반 제품 서비스 개발
- [5] oneM2M 개발자 가이드, <http://www.onem2m.org/application-developer-guide/welcome>