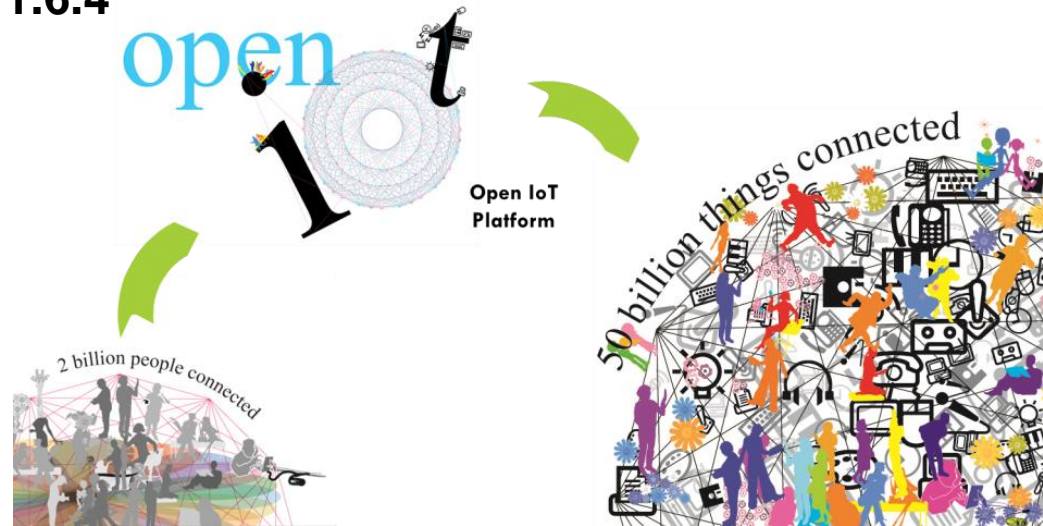




nCube:Thyme

Software Version: 1.6.4



KETI (Korea Electronic Technology Institute)

안 일 엽 (iyahn@keti.re.kr)

1. 개요

1.1 오픈소스 종류 및 oneM2M 표준 준수

1.2 &Cube: Thyme 활용 디바이스 개발 방법

1.3 &Cube 연동 구조

OCEAN Open Sources

- <http://www.iotocean.org>
- Have more platforms → become IoT complicated

OCEAN About License Download Community Showcas

A global alliance based on open source and IoT standards

OCEAN's aim is to share the open source developed based on IoT standards and to promote the development and commercialization of diverse IoT services.

NOTICE	+ MORE	DOWNLOAD
IoT Korea Conference & Int'l Exhibition 2015	2015-09-30	Blue Octopus v1.1
OCEAN Publication Plan	2015-08-06	Yellow Turtle v1.1
OCEAN Site Renewal Guide	2015-07-22	Yellow Turtle v1.0
IoT Innovation Forum 2015 행사 [2015.5.27]	2015-05-15	Blue Octopus v1.0
[ICT DIY Related exhibitions] Community participa...	2015-04-24	Lavender v1.0

Membership application
 The OCEAN services including downloading source codes are provided for free under the OCEAN membership. In principle, the OCEAN membership is permitted only to a person affiliated to a company or organization.

Download
 You can download the source code and documents of the standard IoT server platform, openMobius® and IoT device platform, &Cube®.

Showcase
 All OCEAN members are encouraged to publish and share their developments based on the OCEAN's open sources for further collaboration.

OCEAN About License Download Community Showcas

Download - Yellow Turtle

Yellow Turtle v1.1 2015-09-30

Mobius: Yellow Turtle

Yellow Turtle is an open source software of oneM2M-based IoT Server Platform based on Node.js. The source code and files of Yellow Turtle are under the OCEAN license terms and conditions, i.e.

OCEAN About License Download Community Showcas

Download - Lavender

Lavender v1.0 2015-08-06

&Cube: Lavender

Lavender is an open source software of oneM2M-based IoT Device Platform based on the Java Virtual Machine. The source code and files of Lavender are under the OCEAN license terms and conditions, i.e., 3-clause open source license.

Versions

Code Name	Framework	Version	Ref. Standards
Lavender	Java	1.0	oneM2M Release 1
Lavender	Java	0.8	oneM2M Candidate Release 1

PREREQUISITES

Lavender software is running on a platform-independent Java Virtual Machine.

SYSTEM REQUIREMENTS

System Requirements	Remarks
Operating System	Windows, Linux, Mac OSX
Java Virtual Machine	Java 7
MQTT Broker	Mosquitto 1.4.x

Files

Name	Download Link

1.1 오픈소스 종류 및 oneM2M 표준 준수

■ 오픈소스 종류 별 oneM2M 표준 지원 여부

		AE	CSE			Framework
			ASN	MN	IN	
Mobius	Blue Octopus				✓	Spring
	Yellow Turtle				✓	Node.js
nCube	Rosemary			✓		Node.js
	Lavender		✓			Node.js
	Thyme	✓				Java
		✓				Node.js

■ Latest Release

	oneM2M Release 1				
	Software name	Framework	version	Publication date	Standard
Server(IN-CSE)	Mobius : Blue Octopus	Spring Framework	v1.1	2015-09	TS-0001 Functional Architecture v2.10.0 TS-0004 Service Layer Core Protocol v2.7.1 TS-0009 HTTP Protocol Binding v2.6.1 TS-0010 MQTT Protocol Binding v2.4.1
	Mobius : Yellow Turtle	Node.js	v2.1.12	2016-10	
Gateway(MN-CSE)	&Cube : Rosemary	Node.js	v2.1.12	2016.10	
Device(ASN-CSE)					
Application(AE)	&Cube : Thyme	Node.js	v1.6.4	2016-10	

1.1 오픈소스 종류 및 oneM2M 표준 준수

■ Mobius: Yellow Turtle

- Node.js Java Script 기반 IoT Server Platform (oneM2M IN-CSE)
- HTTP, MQTT 연동 지원
- 경량/저용량 시스템으로 간편한 설치 가능
- 시스템 요구사항

System requirements	Remarks
Operating System	WindowsX, Linux Redhat and CentOS, Mac, Raspbian
Open Source Framework	Node.js
Web Application Server	Node.js
Database	MySQL
CoAP Framework	-
MQTT Broker	Mosquitto 1.4.x

- 표준 리소스 지원

- CSEBase, remoteCSE, AE, container, contentInstance, subscription, latest, oldest, timeSeries, timeSeriesInstance, locationPolicy, group, fanOutPoint, semanticDescriptor

- 표준 Primitive 지원

- XML/Json 지원
- short-name 리소스 지원

1.1 오픈소스 종류 및 oneM2M 표준 준수

■ &Cube: Thyme

- HTTP, MQTT 연동 지원
- Node.js Java Script 기반 IoT Application (oneM2M AE)
- 경량/저용량 시스템으로 간편한 설치 가능
- 시스템 요구사항

System requirements	Remarks
Operating System	WindowsX, Linux Redhat and CentOS, Mac, Raspbian
Open Source Framework	Node.js
Web Application Server	Node.js
Database	MySQL
CoAP Framework	-
MQTT Broker	Mosquitto 1.4.x

- 표준 리소스 지원
 - CSEBase, remoteCSE, AE, container, contentInstance, subscription, notification, latest, oldest
- 표준 Primitive 지원
 - XML/Json 지원
 - short-name 리소스 지원

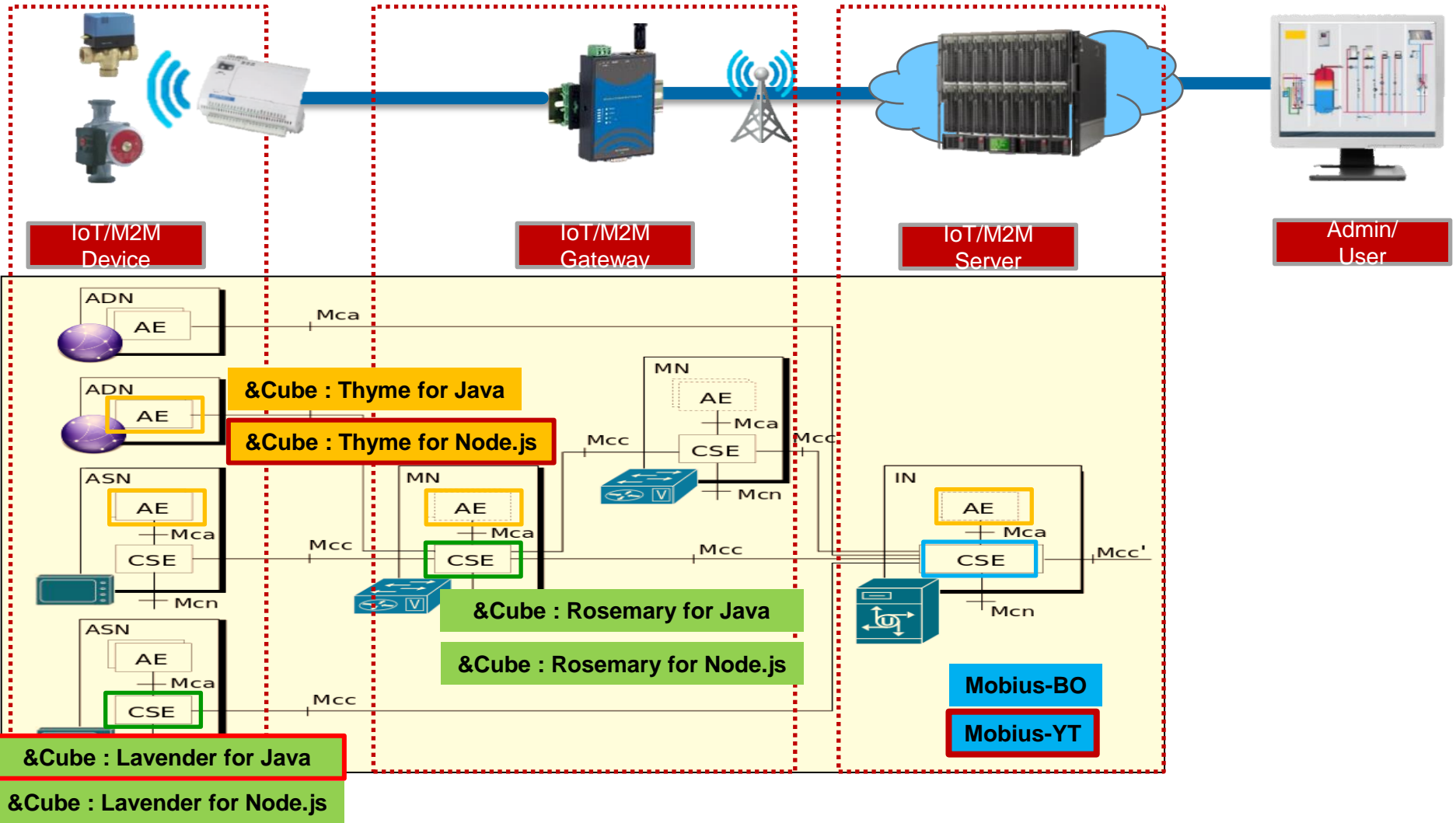
1. 개요

1.1 오픈소스 종류 및 oneM2M 표준 준수

1.2 &Cube: Thyme 활용 디바이스 개발 방법

1.3 &Cube 연동 구조

1.2 Select open source platform in OCEAN to build the use case with oneM2M



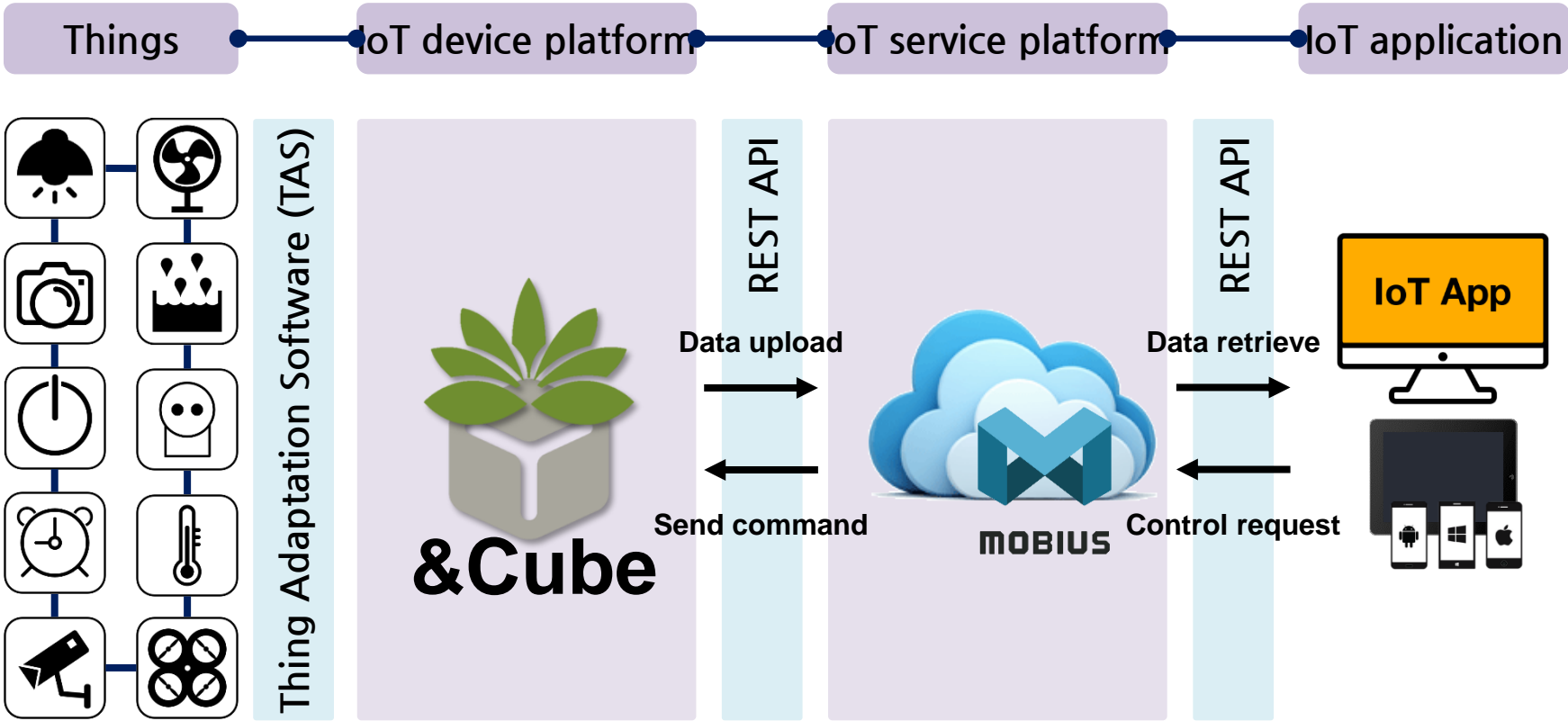
Construction Server Platform

S/W name \ oneM2M Nodes		AE	CSE		IN	Framework
			ASN	MN		
Mobius	Blue Octopus				✓	Spring
	Yellow Turtle				✓	Node.js
&Cube	Rosemary			✓		Java
				✓		Node.js
	Lavender		✓			Java
			✓			Node.js
	Thyme	✓				Java
		✓				Node.js

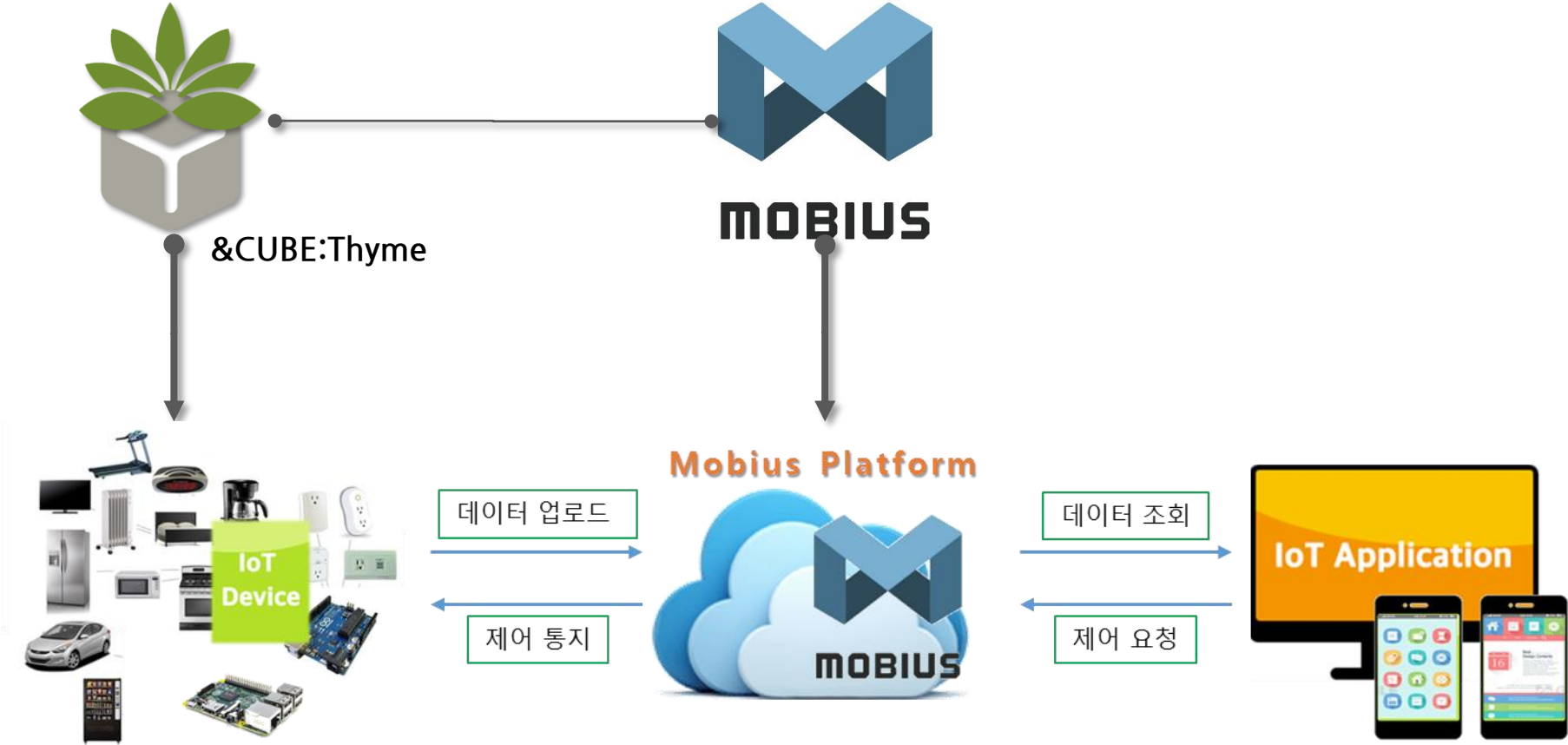
Construction Device Application Platform

S/W name \ oneM2M Nodes		AE	CSE			Framework
			ASN	MN	IN	
Mobius	Blue Octopus				✓	Spring
	Yellow Turtle				✓	Node.js
&Cube	Rosemary			✓		Java
				✓		Node.js
	Lavender		✓			Java
				✓		Node.js
	Thyme	✓				Java
✓					Node.js	

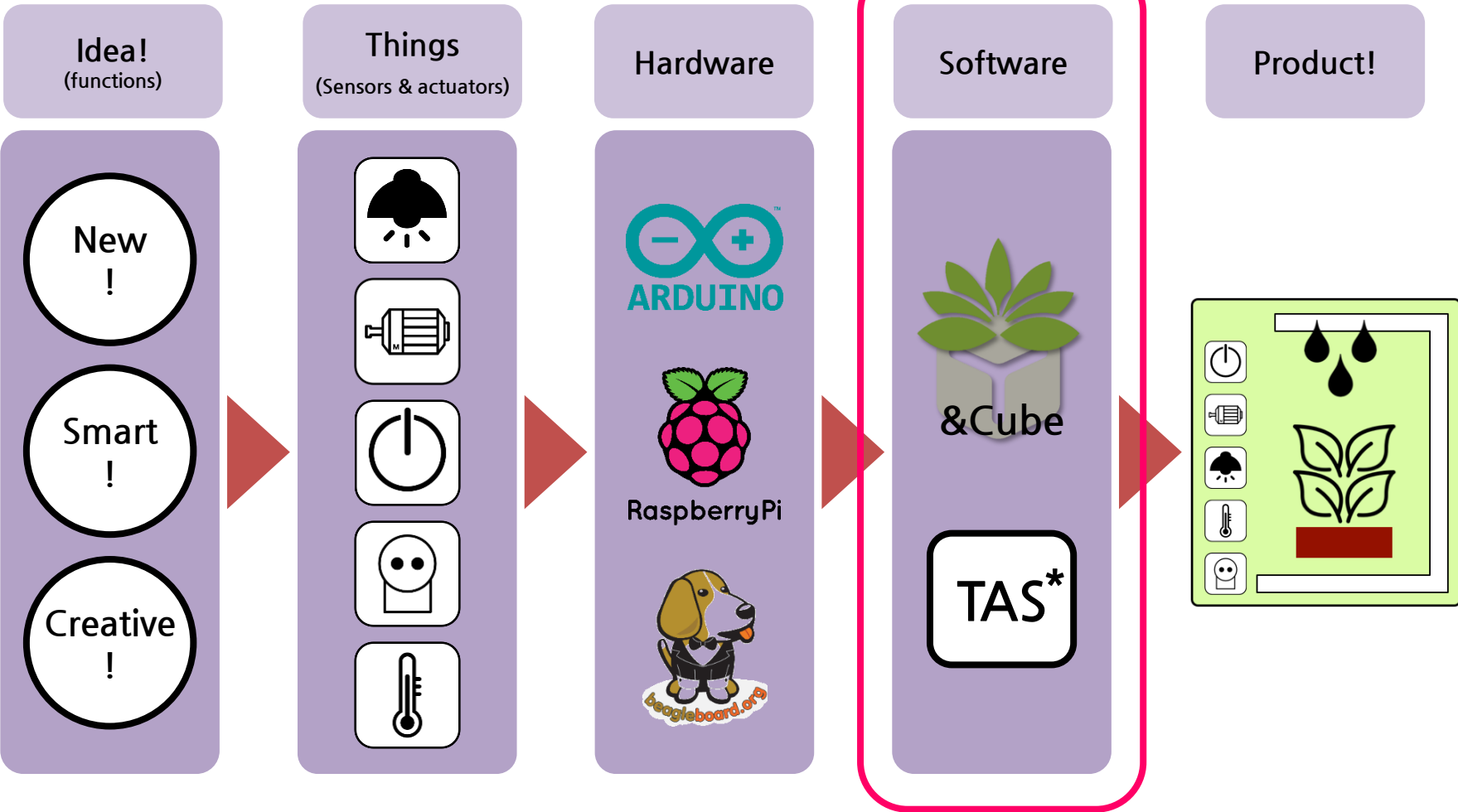
How IoT Devices Work?



1.2 &Cube:Thyme 활용 디바이스 개발 방법



Development Procedure for IoT Product

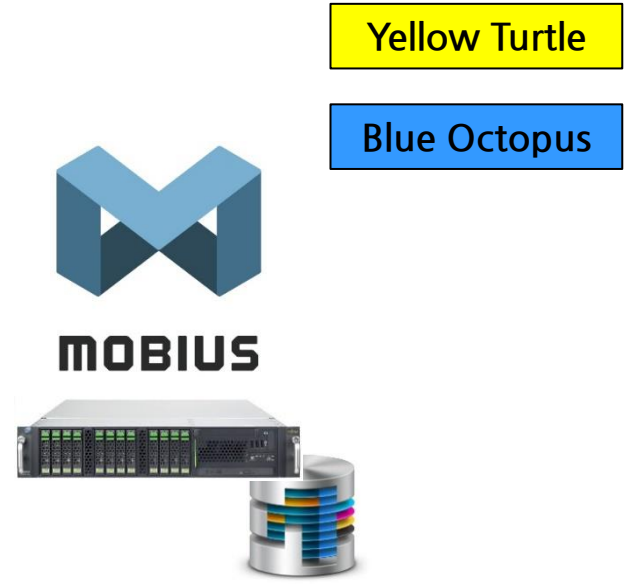
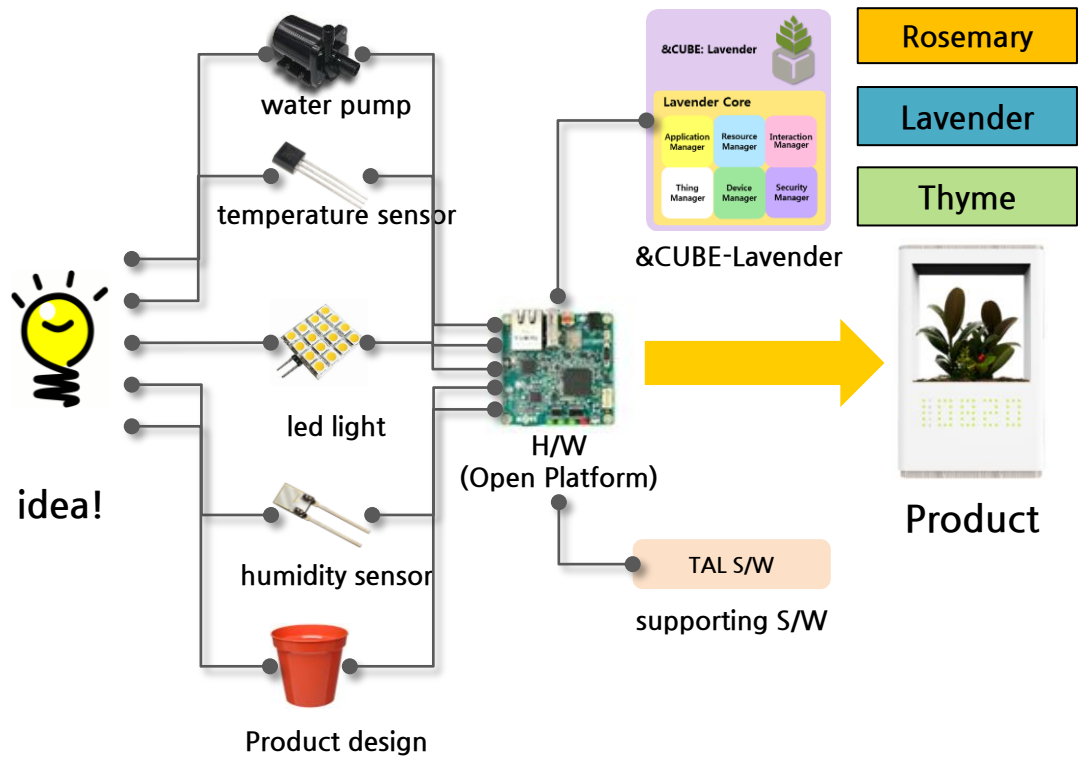


*TAS: Thing Adaptation Software

1.2 &Cube:Thyme 활용 디바이스 개발 방법

Product = Idea + H/W + &CUBE + TAS

IoT Server = Server(H/W) + Mobius



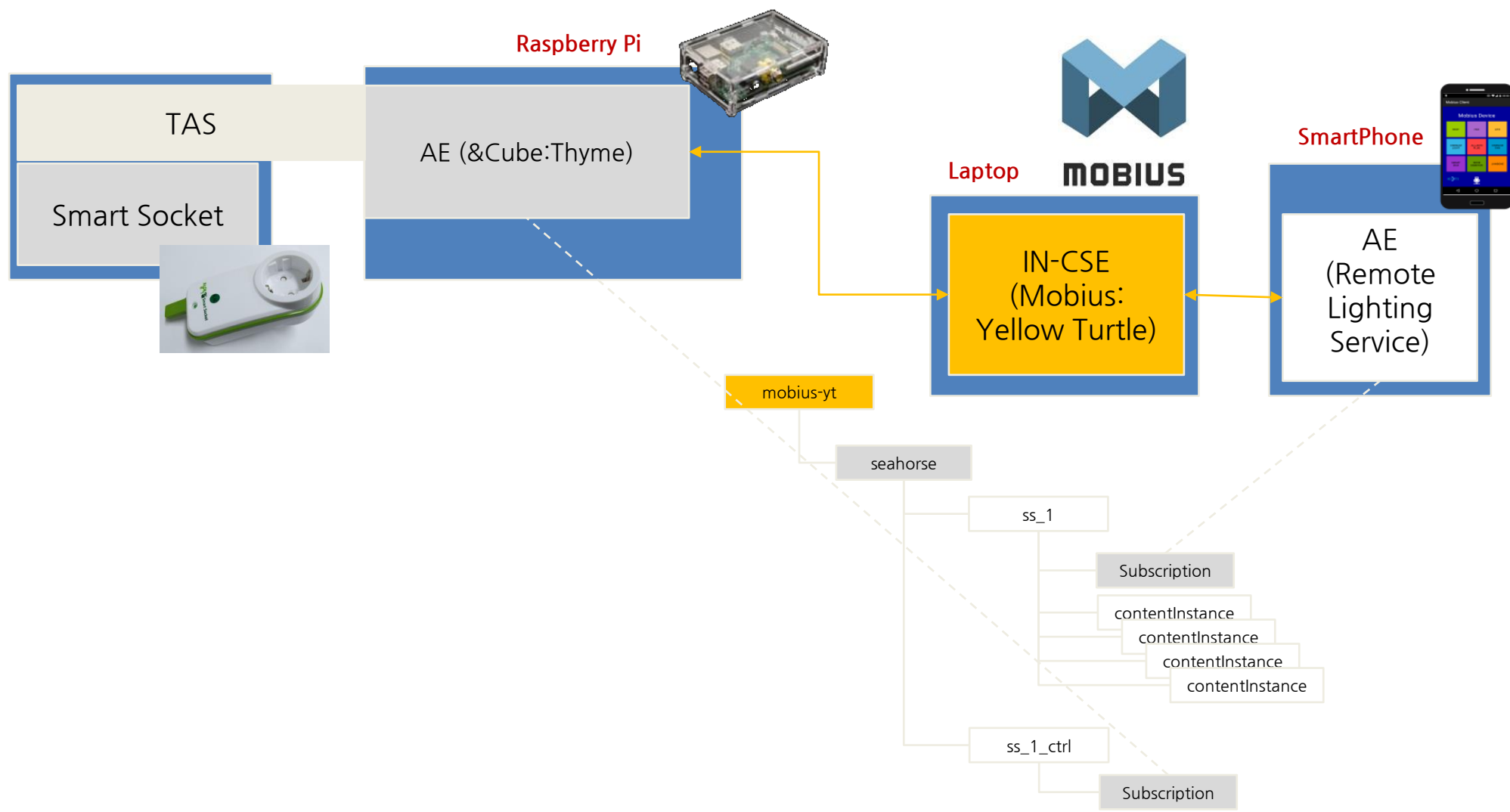
1. 개요

1.1 오픈소스 종류 및 oneM2M 표준 준수

1.2 &Cube: Thyme 활용 디바이스 개발 방법

1.3 &Cube 연동 구조

1.3 &Cube 연동 구조



2. 사물인터넷 디바이스 개발환경 구축

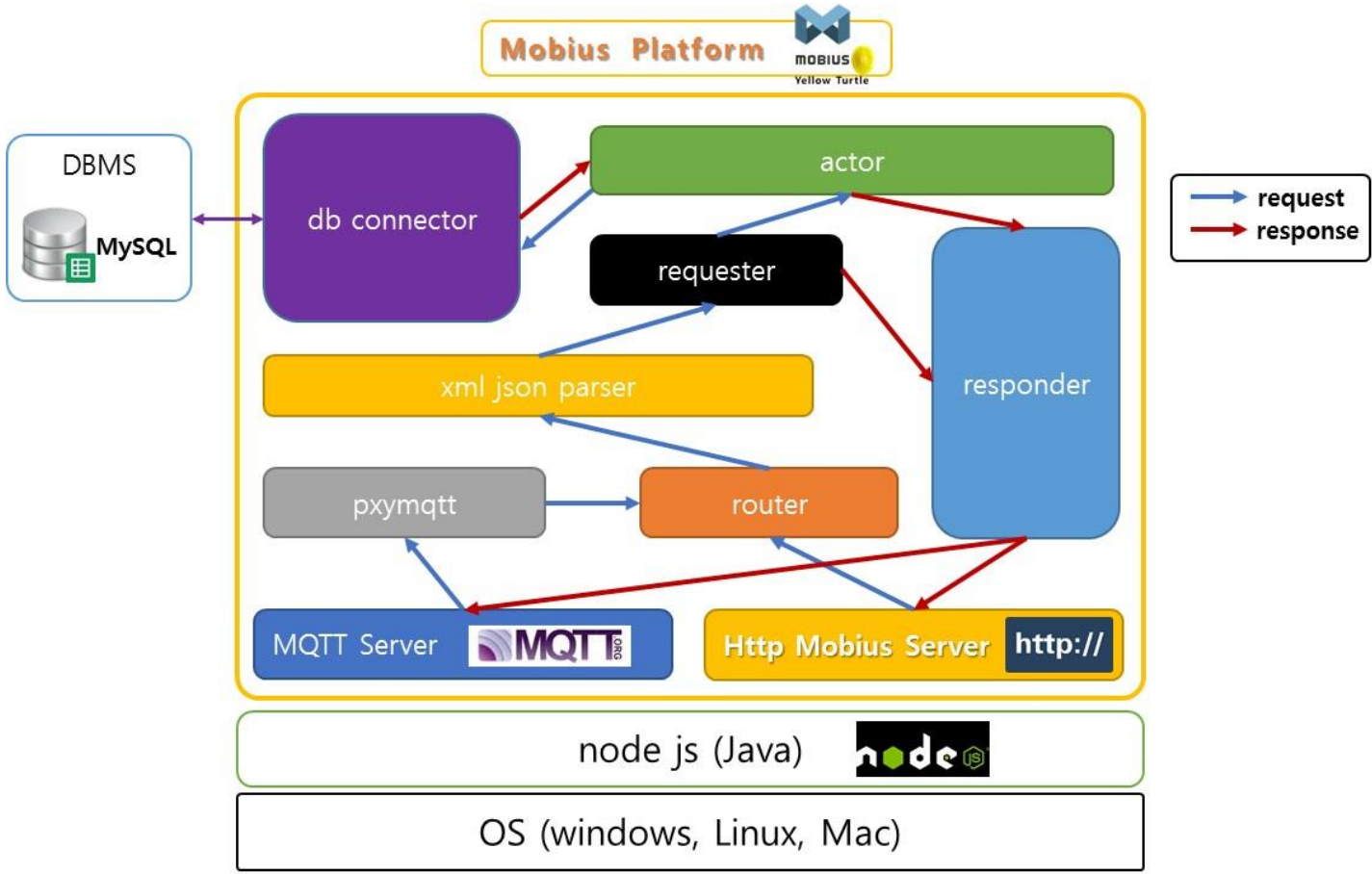
2.1 사물인터넷 모비우스 플랫폼 구동 환경 구축

2.2 사물인터넷 디바이스 Open H/W 및 개발환경 구축

2.3 &Cube: Thyme 구동환경 구축

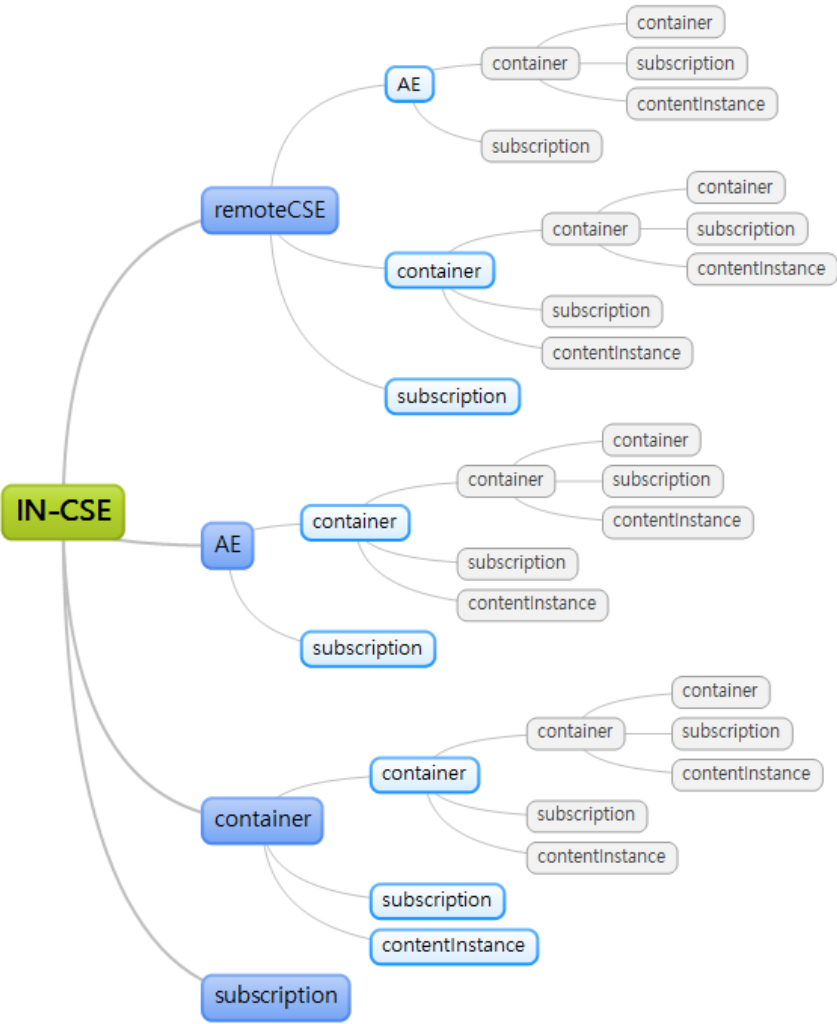
Mobius Yellow Turtle 서버 플랫폼 S/W 아키텍처

- Mobius Yellow Turtle : Node JS를 기반으로 Java Script로 개발
- DB는 MySQL을 지원하고 있으며 통신 프로토콜은 HTTP, MQTT를 지원



Mobius Yellow Turtle 서버 지원 Resource 구조

http://hostname:port/IN-CSE_name/AE_name/container_name/contentInstance_name



Construction Server Platform with Mobius : Yellow Turtle

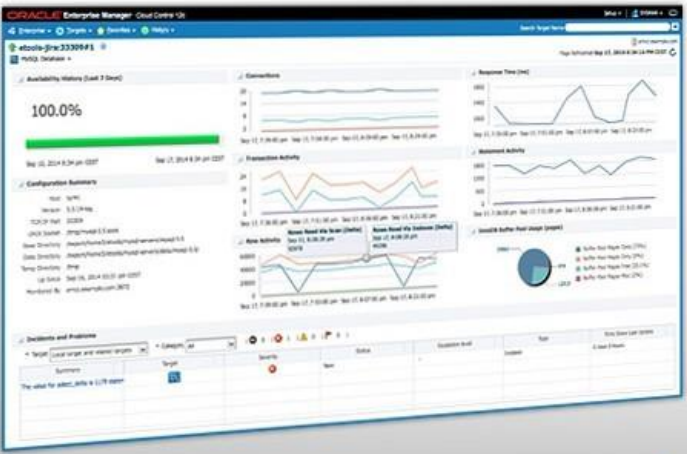
- Reference install guide of Yellow Turtle in OCEAN

Mobius Platform  설치 순서



Construction Server Platform with Mobius : Yellow Turtle

- MySQL 설치 (MySQL Server , MySQL Workbench)



Oracle

Generally Available (GA) Releases | Development Releases

MySQL Community Server 5.6.22

Select Platform: Looking for previous GA versions?

Recommended Download:

MySQL Installer 5.6 for Windows

All MySQL Products. For All Windows Platforms. In One Package.

Starting with MySQL 5.6 the MySQL Installer package replaces the server-only MSI packages.

Windows (x86, 64-bit), MySQL Installer MSI [Download](#)

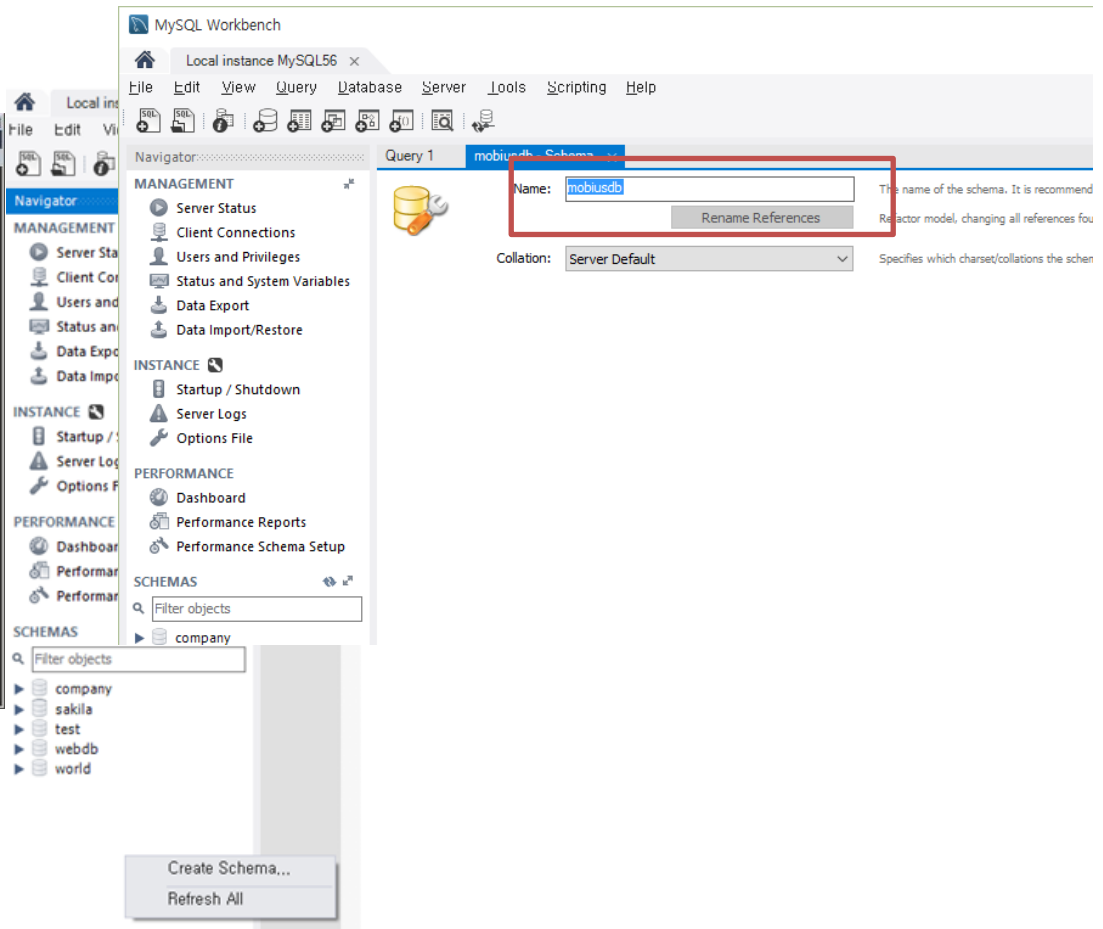
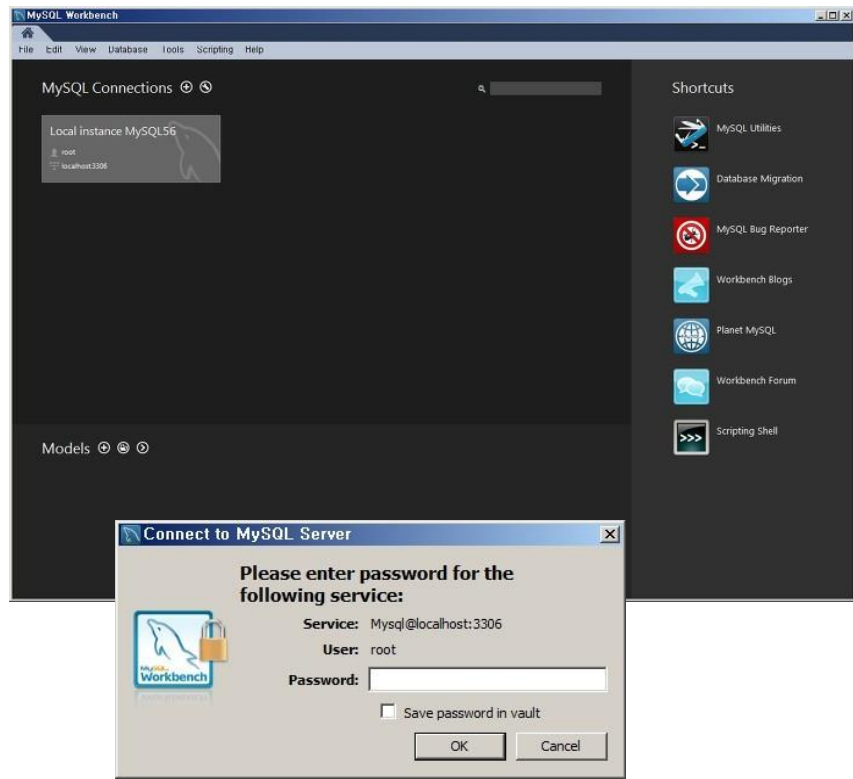
Other Downloads:

Windows (x86, 32-bit), ZIP Archive	5.6.22	342.0M	Download
(mysql-5.6.22-win32.zip)		MDS: 00abcb99a71708d372ff073f870deabd Signature	
Windows (x86, 64-bit), ZIP Archive	5.6.22	347.5M	Download
(mysql-5.6.22-winx64.zip)		MDS: 8810b875ff1651e3c91473faa7ed6509 Signature	

i We suggest that you use the MD5 checksums and GnuPG signatures to verify the integrity of the packages you download.

Construction Server Platform with Mobius : Yellow Turtle

■ MySQL 데이터베이스 생성 (mobiusdb)



Construction Server Platform with Mobius : Yellow Turtle

■ MySQL mobiusdb 테이블 가져오기 (www.iotocean.org)

OCEAN

About License Download Community Showcase Contact

Download

Mobius

- Blue Octopus
- Yellow Turtle
- Cube
- Open Contribution

Yellow Turtle v1.0

Mobius: Yellow Turtle

Yellow Turtle is an open source software of oneM2M-based IoT Server Platform. The source code and files of Yellow Turtle are under the OCEAN license ter BSD open source license.

Versions

Code Name	Framework	Version	Ref. Standards
Yellow Turtle	Node.js Java Script	1.0	oneM2M Release

PREREQUISITES

-

SYSTEM REQUIREMENTS

System Requirements	Remarks
Operating System	WindowsX, Linux Redhat and CentOS, MacOS
Open Source Framework	Node.js
Web Application Server	Node.js
Database	MySQL
CoAP Framework	
MQTT Broker	Mosquitto 1.4.x

Files

Name	Download Link
Mobius Installation Guide Korea	Installation Guide_Yellow_Turtle_v1_0_KR.pdf
Mobius Source 1.0	Mobius_Yellow_Turtle_v1_0.zip
MySQL Script	YellowTurtle_script.sql

File: Installation Guide_Mobius_Yellow_Turtle_v1_0_KR.pdf | Mobius Yellow Turtle.zip | YellowTurtle_script.sql

LAST PAGE

Local Instance MySQL56

Data Import

Import from Disk Import Progress

Options

Import from Dump Project Folder C:\Users\W\yeubi\Documents\W\dumps

Select the Dump Project Folder to import. You can do a selective restore.

Import from Self-Contained File C:\Users\W\yeubi\Documents\W\dumps\Wexport.sql

Select the SQL/dump file to import. Please note that the whole file will be imported.

Default Schema to be Imported To

Default Target Schema: company The default schema to import the dump into. NOTE: this is only used if the dump file doesn't contain it otherwise it is ignored.

Select Database Objects to Import (only available for Project Folders)

Imp... Schema Imp... Schema Objects

Press [Start Import] to start...

company

mobiusdb

Tables

- ae
- cb
- cin
- cnt
- csr
- lcp
- lookup
- mms
- sd
- sub
- ts
- tsi

Views

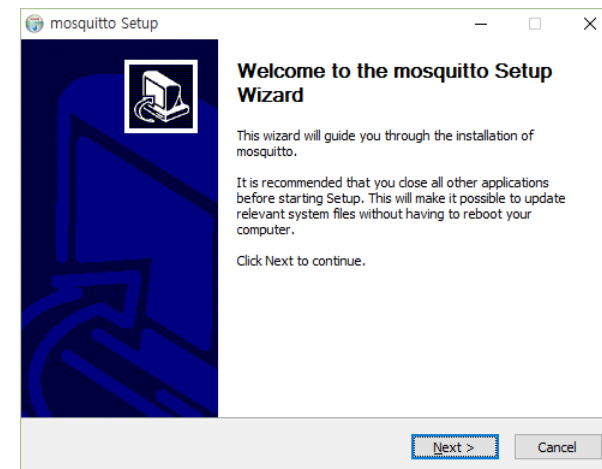
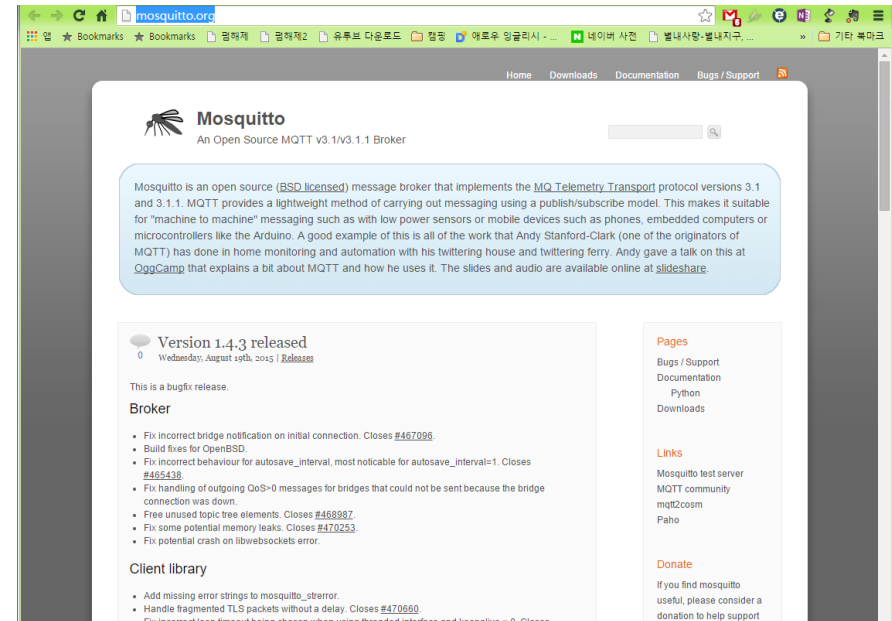
Stored Procedures

Functions

Construction Server Platform with Mobius : Yellow Turtle

■ MQTT 서버 설치

- Mobius Yellow Turtle Platform은 디바이스와의 통신을 위해 MQTT 프로토콜을 지원하고 있으며 이를 위해서 오픈소스로 제공되는 MQTT broker Mosquitto 서버를 설치
- <http://mosquitto.org> 사이트에서 다운로드
- MQTT를 지원하지 않는다면 설치하지 않음



Construction Server Platform with Mobius : Yellow Turtle

■ Node.js

- 고성능의 비동기 IO (Async/Non-blocking IO)를 지원하는 single thread 기반 네트워크 서버
- 2009년 Ryan Dahl에 의해 개발이 시작되었고 현재 수많은 지원 모듈을 가지고 있는 오픈 소스 프로젝트 중 하나
- Google Chrome V8 엔진으로 개발되어 있고 Event 기반의 프로그래밍 모델로써 프로그래밍 언어로는 Java script를 사용
- 현재 많은 인터넷 기업들이 node.js를 도입

■ Node.js 설치

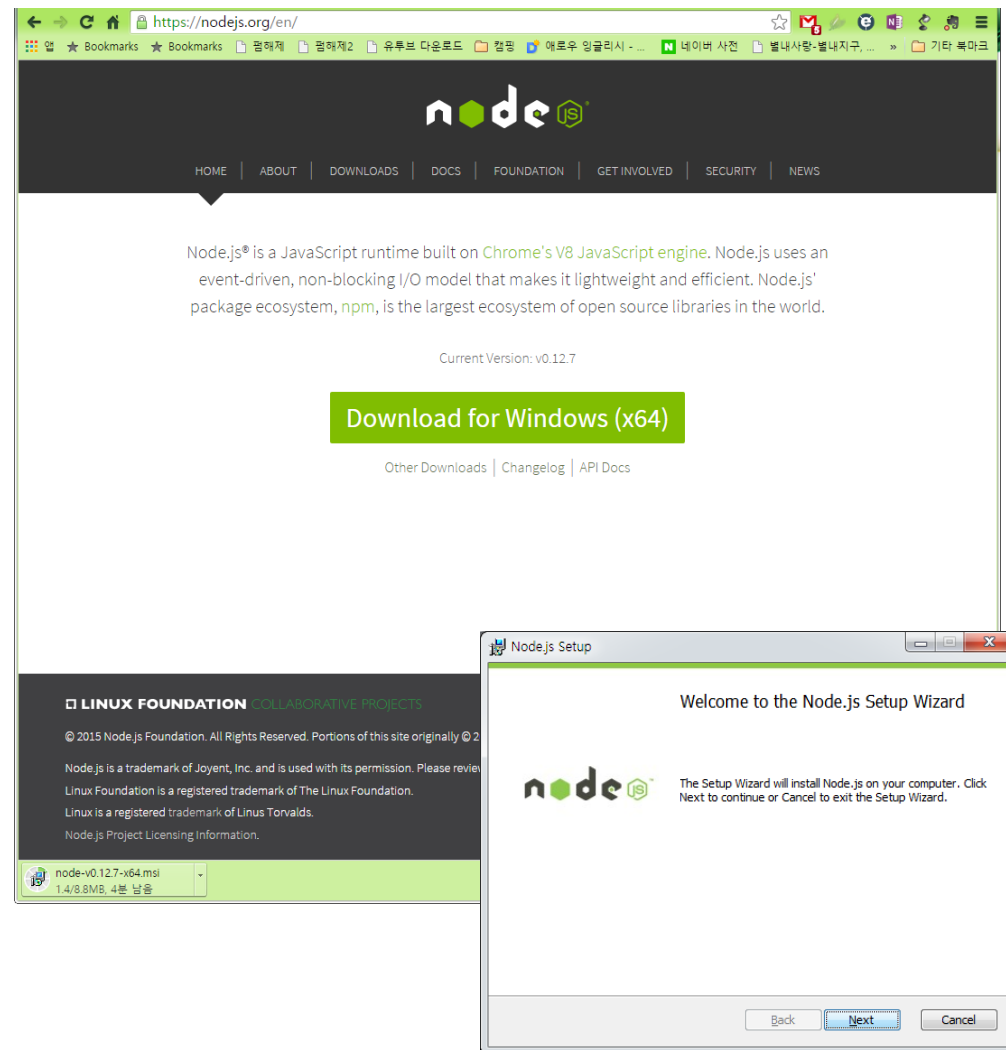
- <http://www.nodejs.org>

■ Node.js 설치 확인

```

cmd 명령 프롬프트 - node
Microsoft Windows [Version 10.0.10240]
(c) 2015 Microsoft Corporation. All rights reserved.

C:\Users\ryeubi>node
> console.log('hello node.js');
hello node.js
undefined
>
  
```



2. 사물인터넷 디바이스 개발환경 구축

2.1 사물인터넷 모비우스 플랫폼 구동 환경 구축

2.2 사물인터넷 디바이스 Open H/W 및 &Cube 개발환경 구축

2.3 &Cube: Thyme 구동환경 구축

2.2 사물인터넷 디바이스 Open H/W 소개

■ Raspberry Pi 3 Model B

- 영국의 라즈베리 파이 재단이 개발
- 기초 컴퓨터 과학 교육을 증진시키기 위해 만든 싱글 보드 컴퓨터
- Raspbian (Debian 계열 Linux) 운영체제 사용
- 기타 운영체제 포팅 가능
- 상세 정보는 <http://www.raspberrypi.org/> 홈페이지에서 확인 가능



■ CM1106 CO2 Sensor

- 공기 CO2 평균 농도값 측정
- UART 통신방식 지원



■ PL2303 USB UART Board

- UART 데이터 USB로 변경
 - 3.3v 5v 출력



■ RGB 3 Color LED

- Red, Green, Blue 동시 표시 가능



2.2 Open H/W 기반 개발환경 구축

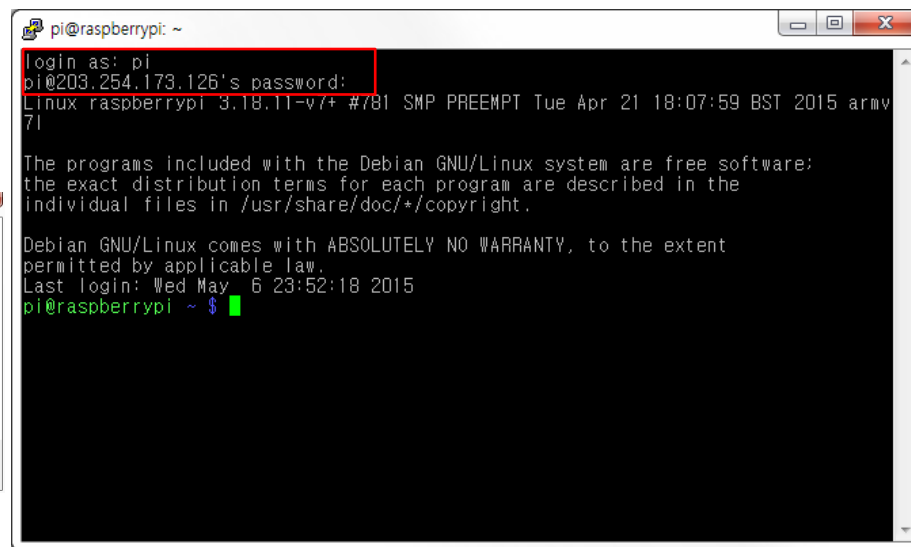
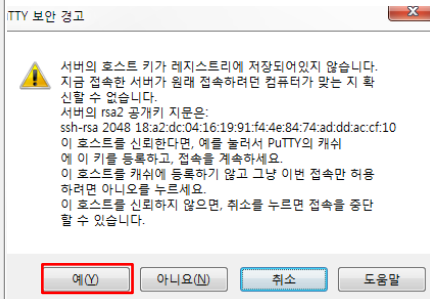
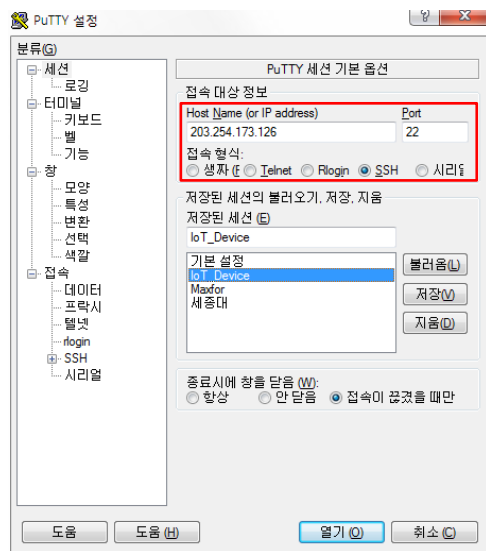
■ 원격 접속환경 구축 (SSH client 사용)

● SSH client 프로그램 다운로드 및 설치 (PuTTY)

– <http://www.chiark.greenend.org.uk/~sgtatham/putty/>

● PuTTY 프로그램 설정 및 연결

- Raspberry-Pi의 IP address 입력을 통해 접속
- 호스트 키 확인 메시지 출력 시 '예'를 눌러 캐쉬에 등록
- 접속 완료 시 로그인 진행 (Raspberry-Pi 초기 설정 → ID : pi, PW : raspberry)



2.2 Open H/W 기반 개발환경 구축

■ Samba 서버 구축

● Repository 업데이트

```
pi@raspberrypi ~ $ sudo apt-get update
.....
Reading package lists... Done
```

● Samba 서버 설치

```
pi@raspberrypi ~ $ sudo apt-get install samba samba-common-bin
.....
Do you want to continue [Y/n]? Y
```

■ Samba 서버 구축

● Samba 서버 사용자 추가

```
pi@raspberrypi ~ $ sudo smbpasswd -a pi
New SMB password: (원하는 패스워드 입력)
Retype new SMB password: (원하는 패스워드 입력)
Added user pi.
```

● Samba 서버 사용자 설정

```
pi@raspberrypi ~ $ sudo nano /etc/samba/smb.conf
..... (가장 마지막 줄 밑에)
[pi]
comment = raspberry pi folder
path = /home/pi
valid user = pi
writable = yes
browseable = yes
<Ctrl>+<X> → Y → <Enter>
```

2.2 Open H/W 기반 개발환경 구축

■ Samba 서버 구축

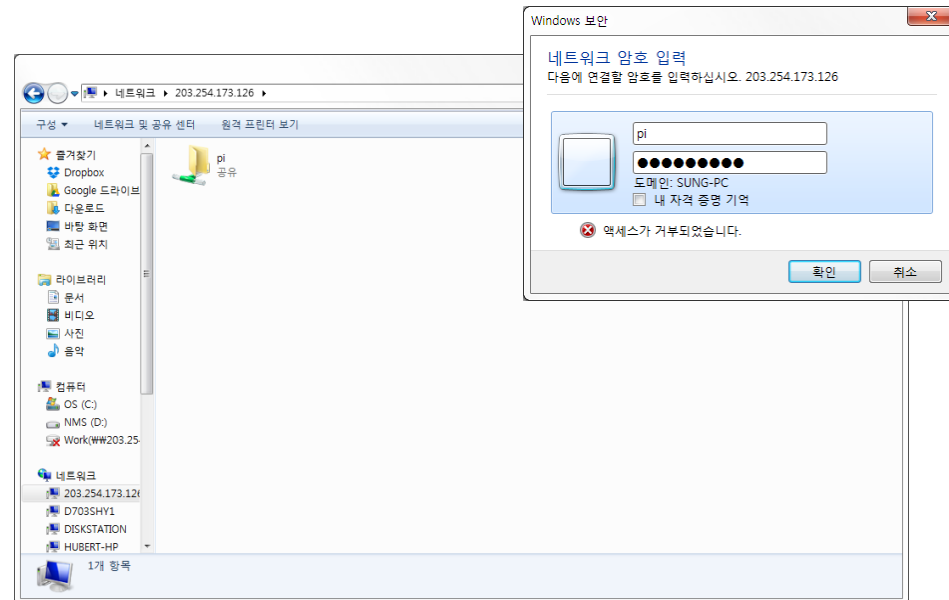
- Samba 서버 재시작

```
pi@raspberrypi ~ $ sudo service samba restart
[ ok ] Stopping Samba daemons: nmbd smbd.
[ ok ] Starting Samba daemons: nmbd smbd.
```

■ Samba 폴더 연결

- Windows 탐색기 실행
- 주소입력 창에 \\Raspberry-Pi IP 주소 입력
 - 예) \\203.254.173.126
- pi 폴더 더블클릭
- 계정 및 패스워드 입력
 - 계정 : pi, 패스워드 : Samba서버 설치 시 지정한 패스워드

```
pi@raspberrypi: ~
Adding group `smbshare' (GID 111) ...
Done.
update-alternatives: using /usr/bin/smbstatus.samba3 to provide /usr/bin/smbstat
us (smbstatus) in auto mode
[ ok ] Starting Samba daemons: nmbd smbd.
Setting up samba-common-bin (2:3.6.6-6+deb7u5) ...
update-alternatives: using /usr/bin/nmblookup.samba3 to provide /usr/bin/nmblook
up (nmblookup) in auto mode
update-alternatives: using /usr/bin/net.samba3 to provide /usr/bin/net (net) in
auto mode
update-alternatives: using /usr/bin/testparm.samba3 to provide /usr/bin/testparm
(testparm) in auto mode
Setting up tdb-tools (1.2.10-2) ...
update-alternatives: using /usr/bin/tdbbackup.tdbtools to provide /usr/bin/tdbba
ckup (tdbbackup) in auto mode
pi@raspberrypi ~ $ sudo smbpasswd -a pi
New SMB password:
Retype new SMB password:
Added user pi.
pi@raspberrypi ~ $ sudo nano /etc/samba/smb.conf
pi@raspberrypi ~ $ sudo service samba restart
[ ok ] Stopping Samba daemons: nmbd smbd.
[ ok ] Starting Samba daemons: nmbd smbd.
pi@raspberrypi ~ $
```



2. 사물인터넷 디바이스 개발환경 구축

2.1 사물인터넷 모비우스 플랫폼 구동 환경 구축

2.2 사물인터넷 디바이스 Open H/W 및 &Cube 개발환경 구축

2.3 &Cube: Thyme 구동환경 구축

Install Node.js

```

pi@raspberrypi: ~/node
pi@raspberrypi ~ $ ls
Desktop Documents Downloads Music Pictures Public python_games Templates Videos
pi@raspberrypi ~ $ mkdir node
pi@raspberrypi ~ $ ls
Desktop Documents Downloads Music node Pictures Public python_games Templates Videos
pi@raspberrypi ~ $ cd node
pi@raspberrypi ~/node $ sudo wget https://node-arm.herokuapp.com/node_archive_armhf.deb
--2016-03-24 13:40:57-- https://node-arm.herokuapp.com/node_archive_armhf.deb
Resolving node-arm.herokuapp.com (node-arm.herokuapp.com)... 23.21.73.34
Connecting to node-arm.herokuapp.com (node-arm.herokuapp.com)|23.21.73.34|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 6728900 (6.4M) [application/x-debian-package]
Saving to: 'node_archive_armhf.deb'

node_archive_armhf.deb      100%[=====>] 6.42M  373KB/s  in 22s

2016-03-24 13:41:21 (294 KB/s) - 'node_archive_armhf.deb' saved [6728900/6728900]

pi@raspberrypi ~/node $ ls
node_archive_armhf.deb
pi@raspberrypi ~/node $ sudo dpkg -i node_archive_armhf.deb
Selecting previously unselected package node.
(Reading database ... 118707 files and directories currently installed.)
Preparing to unpack node_archive_armhf.deb ...
Unpacking node (0.12.6-1) ...
Setting up node (0.12.6-1) ...
Processing triggers for man-db (2.7.0.2-5) ...
pi@raspberrypi ~/node $ █

```

Check the version of Node.js and package manager

```

>> pi@raspberrypi ~/node $ node -v
>> pi@raspberrypi ~/node $ npm -v

```


2.2 &Cube: Thyme 구동환경 구축

■ Node.js 패키지 설치

● Node.js 다운로드 및 설치

```

pi@raspberrypi ~ $ mkdir node
pi@raspberrypi ~ $ cd node
pi@raspberrypi ~/node $ sudo apt-get update
pi@raspberrypi ~/node $ sudo apt-get upgrade
pi@raspberrypi ~/node $ sudo apt-get remove nodejs
pi@raspberrypi ~/node $ sudo wget https://node-arm.herokuapp.com/node_archive_armhf.deb
pi@raspberrypi ~/node $ sudo dpkg -i node_archive_armhf.deb (패키지 설치 명령어)
pi@raspberrypi ~/node $ node -v (버전 확인 명령어)
pi@raspberrypi ~/node $ npm -v (추가 라이브러리 설치도구 버전 확인 명령어)

```

```

Resolving node-arm.herokuapp.com (node-arm.herokuapp.com)... 23.21.97.86
Connecting to node-arm.herokuapp.com (node-arm.herokuapp.com)|23.21.97.86|:443...
. connected.
HTTP request sent, awaiting response... 200 OK
Length: 6728900 (6.4M) [application/x-debian-package]
Saving to: 'node_archive_armhf.deb'

node_archive_armhf. 100%[=====>] 6.42M 187KB/s in 28s

2015-10-20 15:20:40 (236 KB/s) - 'node_archive_armhf.deb' saved [6728900/6728900]

pi@raspberrypi ~ $ ls
Desktop Music Pictures Templates thyme.zip
Documents node_archive_armhf.deb Public thyme Videos
Downloads node_latest_armhf.deb python_games thyme_tas
pi@raspberrypi ~ $

```

```

(Reading database ... 116809 files and directories currently installed.)
Preparing to unpack node_archive_armhf.deb ...
Unpacking node (0.12.6-1) ...
Setting up node (0.12.6-1) ...
Processing triggers for man-db (2.7.0.2-5) ...
pi@raspberrypi ~ $

```

3. 사물인터넷 디바이스 개발 실습

3.1 &Cube:Thyme 구동 실습

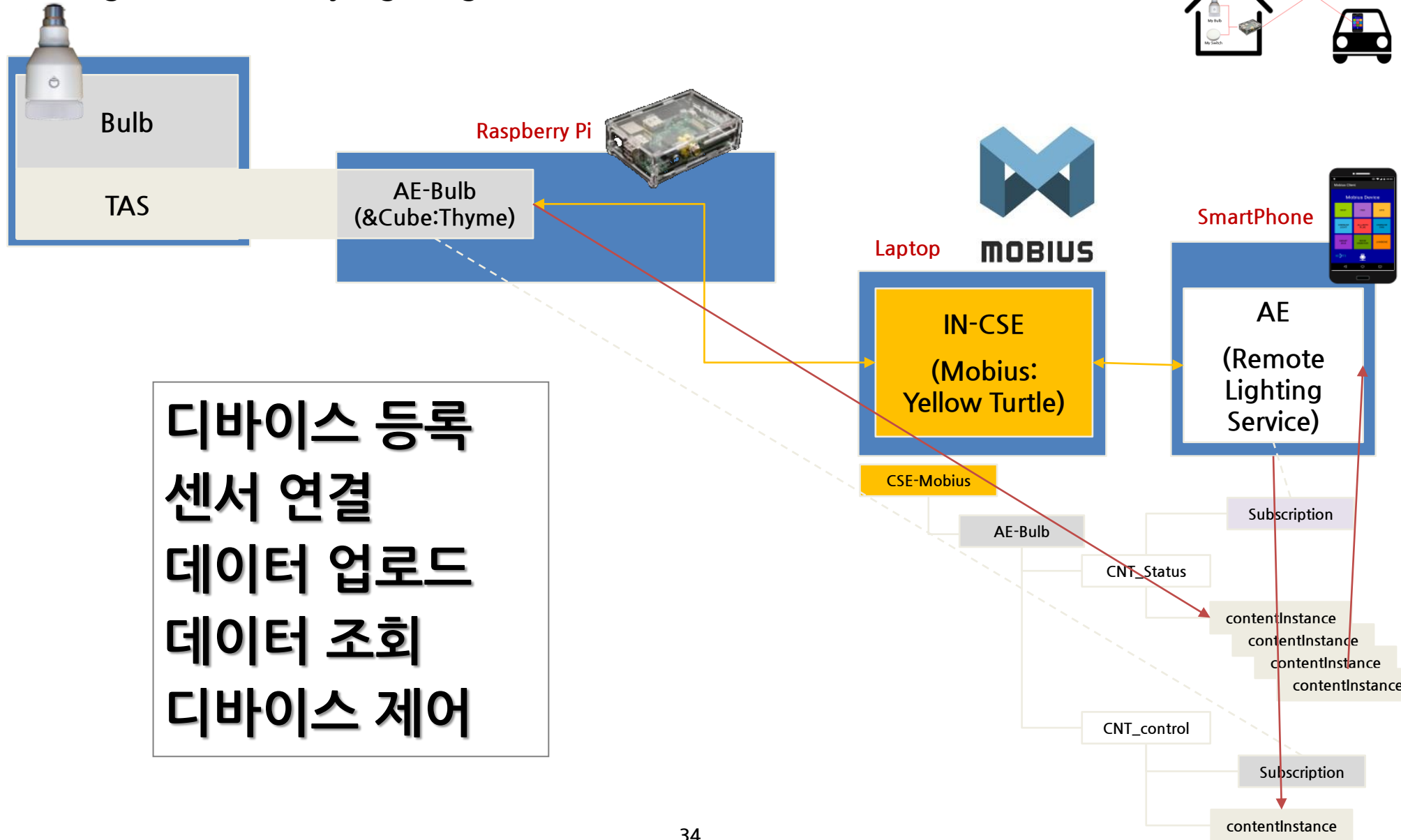
3.2 &Cube:TAS 개발 실습

3.3 Test Device

3.4 서비스 시나리오

3.5 Yellow Turtle 구축 실습

Configuration for My Lighting Service



디바이스 등록
 센서 연결
 데이터 업로드
 데이터 조회
 디바이스 제어

How to Develop Software for IoT Devices?

Install &Cube

- Download &Cube
- Install Node.js
- Configure &Cube
- Run &Cube

Develop TAS

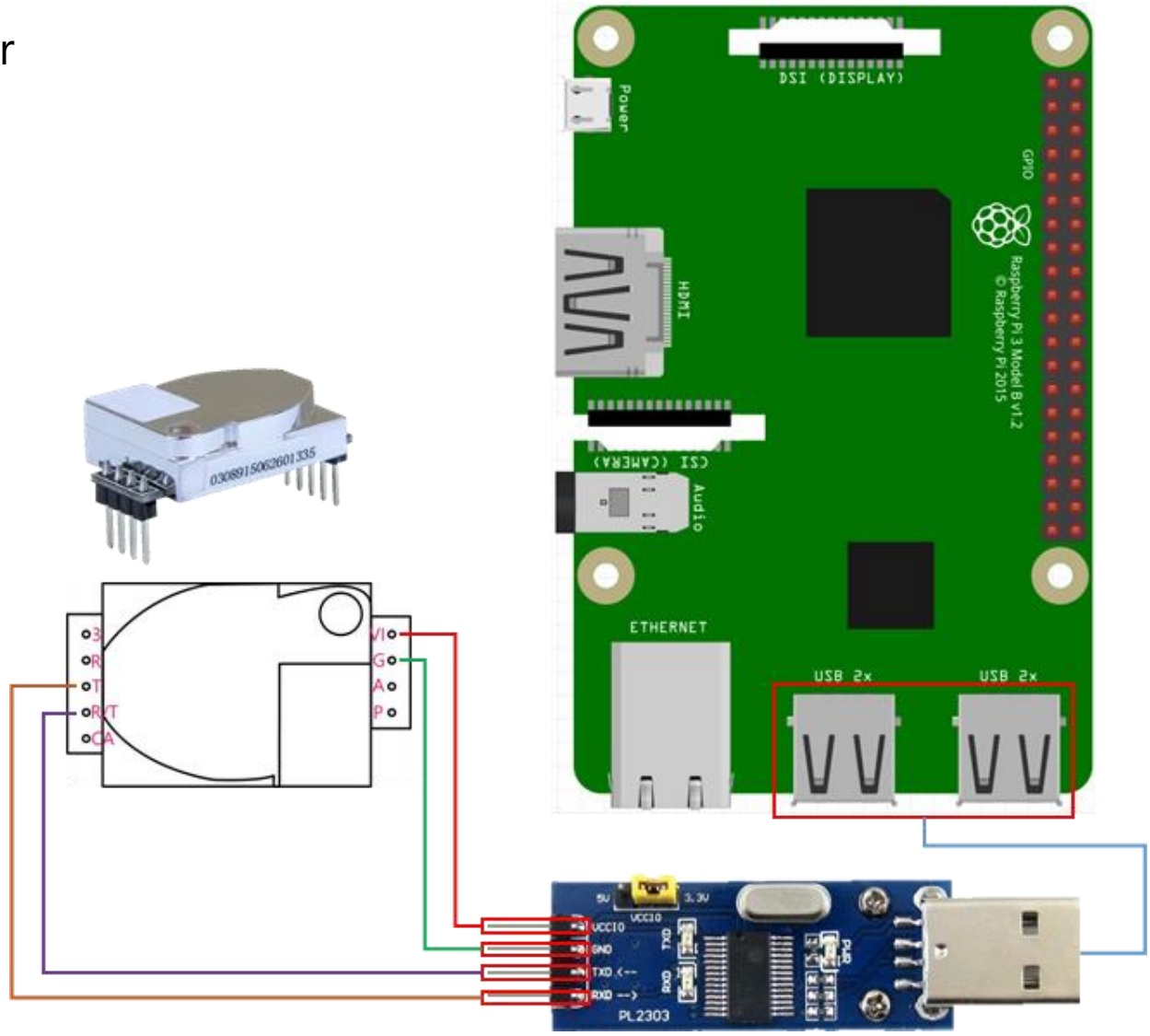
- Download TAS sample
- Develop TAS
- Configure TAS
- Run TAS

Test Device

- Download Mobius API collection
- Install test program
- Test device

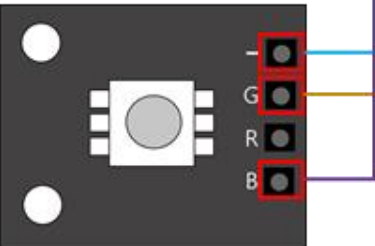
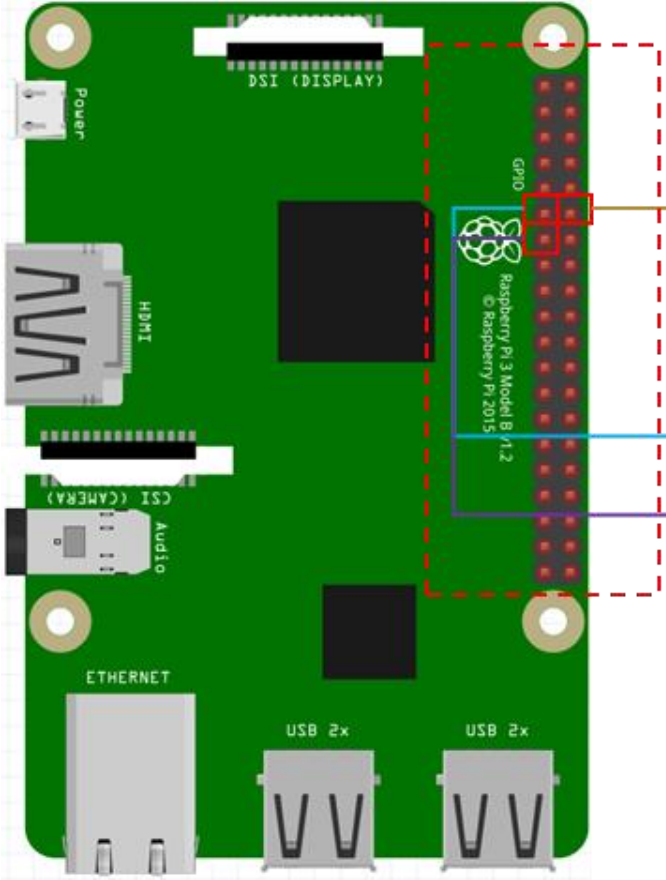
하드웨어 준비

Connect CO2 Sensor



하드웨어 준비

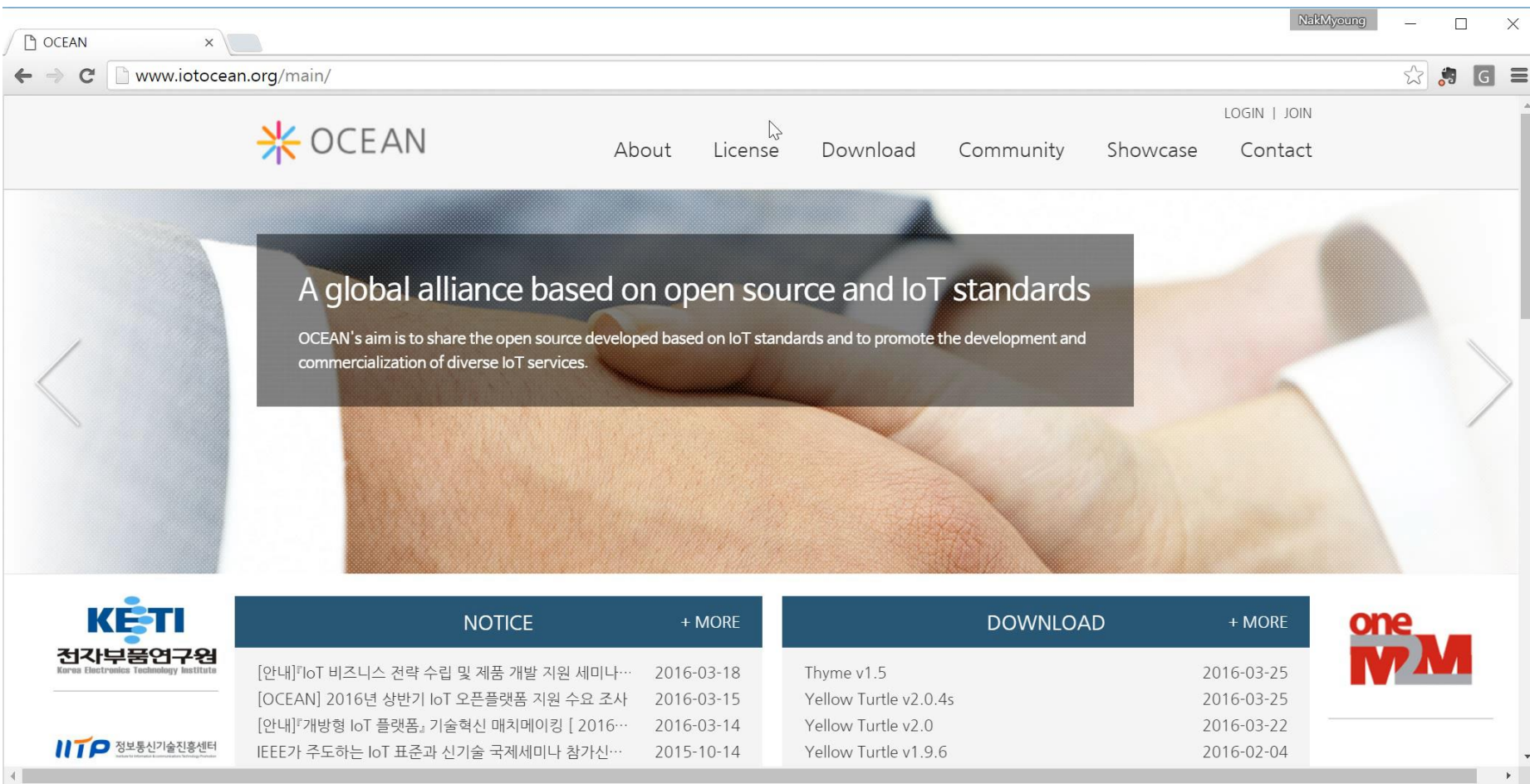
Connect RGB LED



```
pi@raspberrypi: ~
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| BCM | wPi | Name | Mode | V | Physical | V | Mode | Name | wPi | BCM |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| 2 | 8 | 3.3v | | | 1 | 2 | | | 5v | | |
| 3 | 9 | SDA.1 | IN | 1 | 3 | 4 | | | 5v | | |
| 4 | 7 | SCL.1 | IN | 1 | 5 | 6 | | | 0v | | |
| 7 | 15 | GPIO.7 | IN | 1 | 7 | 8 | 0 | IN | TxD | 15 | 14 |
| 8 | 16 | 0v | | | 9 | 10 | 1 | IN | RxD | 16 | 15 |
| 17 | 0 | GPIO.0 | OUT | 1 | 11 | 12 | 1 | OUT | GPIO.1 | 17 | 18 |
| 27 | 2 | GPIO.2 | OUT | 1 | 13 | 14 | | | 0v | | |
| 22 | 3 | GPIO.3 | IN | 0 | 15 | 16 | 0 | IN | GPIO.4 | 4 | 23 |
| 10 | 12 | 3.3v | | | 17 | 18 | 0 | IN | GPIO.5 | 5 | 24 |
| 9 | 13 | MOSI | IN | 0 | 19 | 20 | | | 0v | | |
| 11 | 14 | MISO | IN | 0 | 21 | 22 | 0 | IN | GPIO.6 | 6 | 25 |
| 0 | 30 | SCLK | IN | 0 | 23 | 24 | 1 | IN | CE0 | 10 | 8 |
| 5 | 21 | 0v | | | 25 | 26 | 1 | IN | CE1 | 11 | 7 |
| 6 | 22 | SDA.0 | IN | 1 | 27 | 28 | 1 | IN | SCL.0 | 31 | 1 |
| 13 | 23 | GPIO.21 | IN | 1 | 29 | 30 | | | 0v | | |
| 19 | 24 | GPIO.22 | IN | 1 | 31 | 32 | 0 | IN | GPIO.26 | 26 | 12 |
| 26 | 25 | GPIO.23 | IN | 0 | 33 | 34 | | | 0v | | |
| | | GPIO.24 | IN | 0 | 35 | 36 | 0 | IN | GPIO.27 | 27 | 16 |
| | | GPIO.25 | IN | 0 | 37 | 38 | 0 | IN | GPIO.28 | 28 | 20 |
| | | 0v | | | 39 | 40 | 0 | IN | GPIO.29 | 29 | 21 |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| BCM | wPi | Name | Mode | V | Physical | V | Mode | Name | wPi | BCM |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
pi@raspberrypi:~$
```

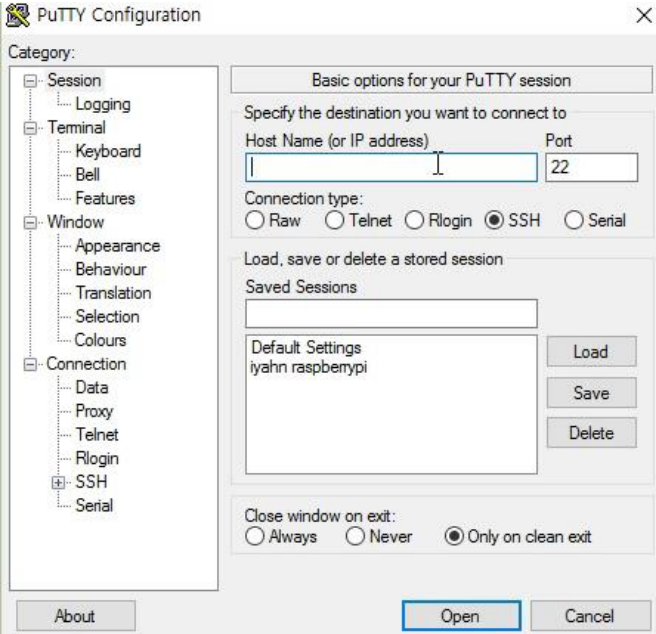
“-” -> pin0(wiringPi)
“G” -> pin1(wiringPi)
“B” -> pin2(wiringPi)

Download &Cube



Download &Cube:Thyme ([Download](#) → [&Cube](#) → [Thyme](#))

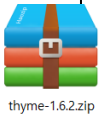
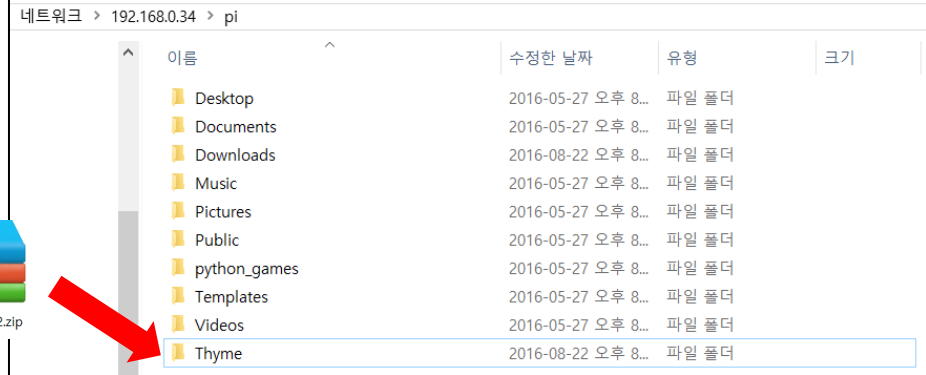
IoT 장치 연결



IoT 장치 (Raspberry-Pi) 연결

Install &Cube: Thyme

- &Cube: Thyme Download and Run with Raspberry Pi
 - Download &Cube:Thyme in www.iotocean.org
 - Copy the thyme source file downloaded to Raspberry Pi with samba
 - Login Raspberry Pi with putty SSH program
 - Create Thyme folder



```

pi@raspberrypi ~/node/thyme $ cd thyme
pi@raspberrypi ~/node/thyme $ unzip thyme-1.6.x.zip
pi@raspberrypi ~/node/thyme $ sudo npm install
  
```

```

pi@raspberrypi:~/Thyme $ unzip thyme-1.6.2.zip
Archive: thyme-1.6.2.zip
  inflating: thyme.js
  inflating: thyme/adn.js
  inflating: thyme/mqtt_adn.js
  inflating: app.js
  inflating: conf.json
  inflating: conf.json.old
  inflating: mqtt_app.js
  inflating: notificationURI.txt
  inflating: package.json
pi@raspberrypi:~/Thyme $ █
  
```

```

mqtt@1.14.0 node_modules/mqtt
├── inherits@2.0.1
├── reinterval@1.1.0
├── xtend@4.0.1
├── help-me@0.1.0
├── minimist@1.2.0
├── readable-stream@1.0.34 (string_decoder@0.10.31, isarray@0.0.1, core-util-is@1.0.2)
├── commist@1.0.0 (leven@1.0.2)
├── mqtt-connection@2.1.1 (through2@0.6.5, reduplexer@1.1.0)
├── mqtt-packet@3.4.7 (bl@0.9.5)
├── end-of-stream@1.1.0 (once@1.3.3)
├── pump@1.0.1 (once@1.3.3)
├── concat-stream@1.5.1 (typedarray@0.0.6, readable-stream@2.0.6)
├── split2@2.1.0 (through2@2.0.1)
├── websocket-stream@3.2.1 (ws@1.1.1, through2@2.0.1, duplexify@3.4.5)
├─┬─ xmlbuilder@2.6.5 node_modules/xmlbuilder
│   └── lodash@3.10.1
├─┬─ xml2js@0.4.17 node_modules/xml2js
│   ├── sax@1.2.1
│   └── xmlbuilder@4.2.1 (lodash@4.15.0)
pi@raspberrypi:~/Thyme $ █
  
```

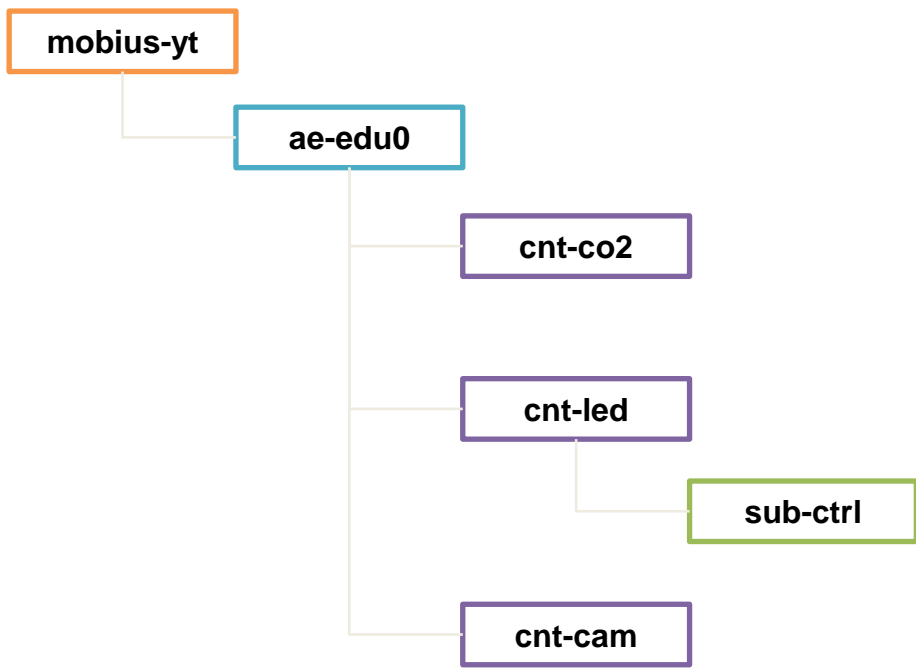
Configure &Cube : Thyme

- Configure &Cube:Thyme for resource structure created into Mobius

```

{
  "useprotocol": "http",
  "cse": {
    "cbhost": "203.253.128.151",
    "cbport": "7579",
    "cbname": "mobius-yt",
    "cbcseid": "/mobius-yt"
  },
  "ae": {
    "aeid": "S",
    "appid": "0.2.481.1.1",
    "appname": "ae-edu0",
    "appport": "9727",
    "bodytype": "json",
    "tasport": "3105"
  },
  "cnt": [
    {
      "parentpath": "/ae-edu0",
      "ctname": "cnt-co2"
    },
    {
      "parentpath": "/ae-edu0",
      "ctname": "cnt-led"
    },
    {
      "parentpath": "/ae-edu0",
      "ctname": "cnt-cam"
    }
  ],
  "sub": [
    {
      "parentpath": "/ae-edu0/cnt-led",
      "subname": "sub-ctrl",
      "nu": "mqtt://autoset"
    }
  ]
}

```



Run &Cube: Thyme

```
pi@raspberrypi ~/node/thyme $ node thyme.js
```

```
pi@raspberrypi: ~/Thyme
pi@raspberrypi:~/Thyme $ node thyme.js █
```

```
pi@raspberrypi: ~/Thyme
CT00000000000000000000000039</ri><pi>AE000000000000000000000096</pi><ct>20160822T144317</c
t><lt>20160822T182327</lt><lb1>cnt-co2</lb1><et>99991231T000000</et><st>267</st>
<cr>SHJ-tF-Og0000000000004931720896</cr><cni>267</cni><cbs>1068</cbs><la>/CB00061
/18644122219/ae-edu0/cnt-co2/la</la><ol>/CB00061/18644122219/ae-edu0/cnt-co2/ol<
/ol></m2m:cnt>
[crtct response] : 409
x-m2m-rsc : 4105 <----<m2m:cnt rn="cnt-led" xmlns:m2m="http://www.onem2m.org/xml
/protocols" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"><ty>3</ty><ri>
CT000000000000000000000038</ri><pi>AE000000000000000000000096</pi><ct>20160822T144317</c
t><lt>20160822T182327</lt><lb1>cnt-led</lb1><et>99991231T000000</et><st>6</st><cr>
r>SHJ-tF-Og0000000000004931720896</cr><cni>6</cni><cbs>6</cbs><la>/CB00061/186441
22219/ae-edu0/cnt-led/la</la><ol>/CB00061/18644122219/ae-edu0/cnt-led/ol</ol></m
2m:cnt>
[sh_state] : delsub
x-m2m-rsc : 2002 <----
[sh_state] : crtsub
x-m2m-rsc : 2001 <----<m2m:sub rn="sub-ctrl" xmlns:m2m="http://www.onem2m.org/xml
/protocols" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"><ty>23</ty><ri>
i>SS000000000000000000000029</ri><pi>CT000000000000000000000038</pi><ct>20160822T205712<
/ct><lt>20160822T205712</lt><et>99991231T000000</et><enc><net>3</net></enc><nu>m
qtt://oneM2M/req/+/SHJ-tF-Og0000000000004931720896</nu><cnt>2</cnt></m2m:sub>
[sh_state] : crtci
TCP Server (192.168.0.34) is listening on port 3105
█
```

3. 사물인터넷 디바이스 개발 실습

3.1 &Cube: Thyme 구동 실습

3.2 &Cube:TAS 개발 실습

3.3 Test Device

3.4 서비스 시나리오

3.5 Yellow Turtle 구축 실습

Download TAS Sample

The screenshot shows the OCEAN website interface. At the top, there is a navigation bar with the OCEAN logo and links for About, License, Download, Community, Showcase, and Contact. A main banner features the text: "A global alliance based on open source and IoT standards" and "OCEAN's aim is to share the open source developed based on IoT standards and to promote the development and commercialization of diverse IoT services." Below the banner, there are two columns of content. The left column is titled "NOTICE" and includes a link to "[안내] IoT 비즈니스 전략 수립 및 제품 개발 지원 세미나...". The right column is titled "DOWNLOAD" and lists two items: "Thyme v1.5" and "Yellow Turtle v2.0.4s". At the bottom of the browser window, a download bar shows "Thyme_v1.5.zip" has been downloaded.

Download TAS Sample ([Download](#) → [&Cube](#) → [Thyme](#))

Develop TAS

pi@raspberrypi: ~/node/thyme_tas

```
pi@raspberrypi ~/node/thyme_tas $
```

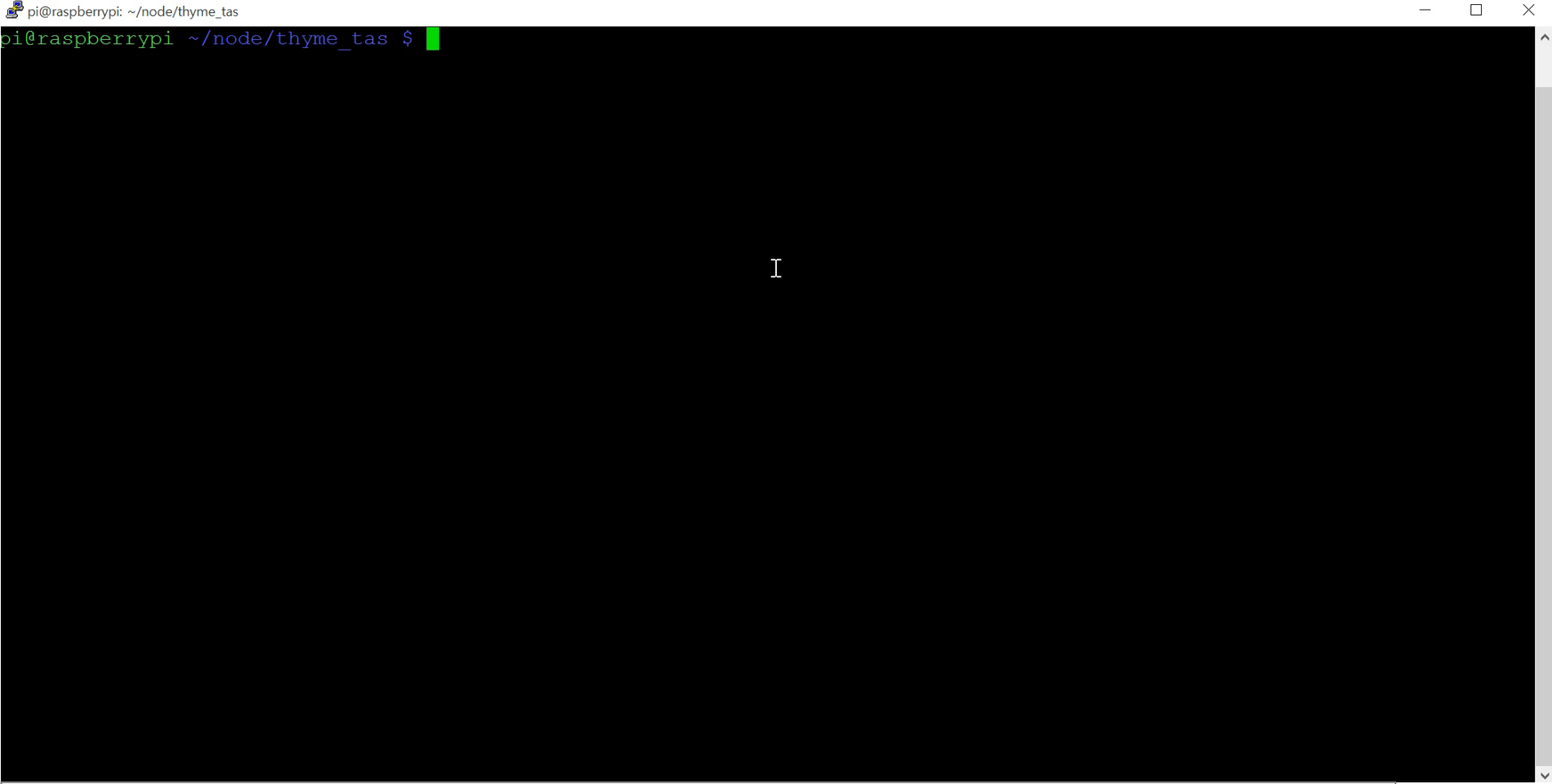
I

Edit TAS sample source, `app.js`

```
>> pi@raspberrypi ~/node/thyme_tas $ sudo nano app.js
```

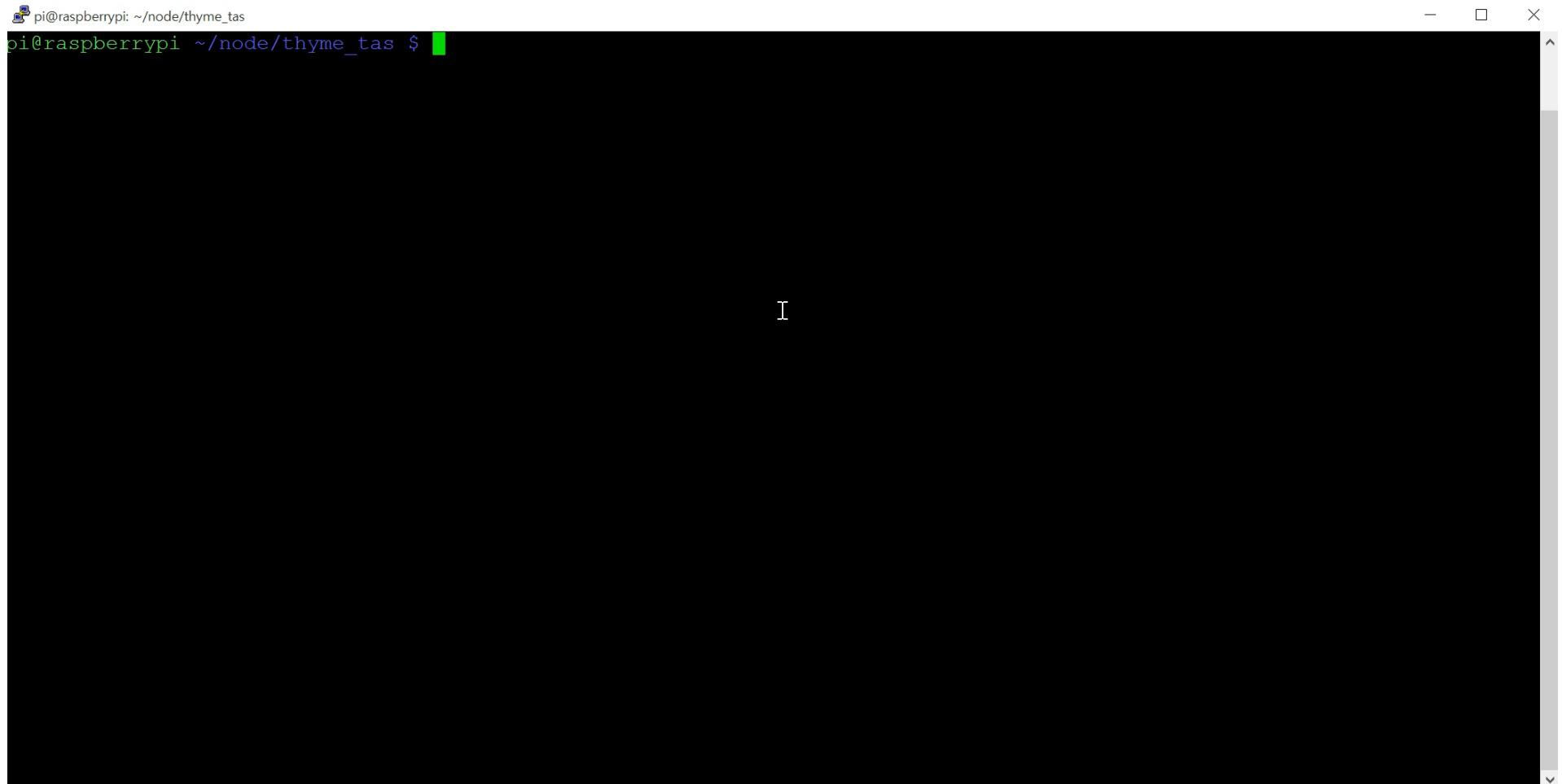
Configure TAS

```
pi@raspberrypi: ~/node/thyme_tas
pi@raspberrypi ~/node/thyme_tas $
```

A terminal window with a black background and white text. The prompt is 'pi@raspberrypi ~/node/thyme_tas \$' with a green cursor. A white cursor is visible in the center of the terminal area.

Edit TAS config file, `conf.xml` (enter information about Things)

Run TAS

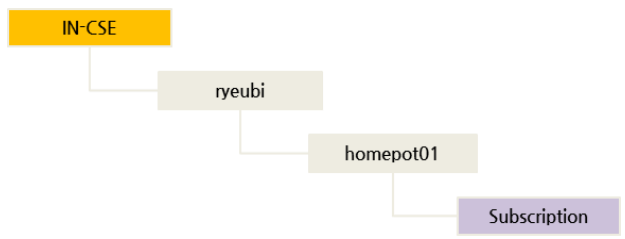
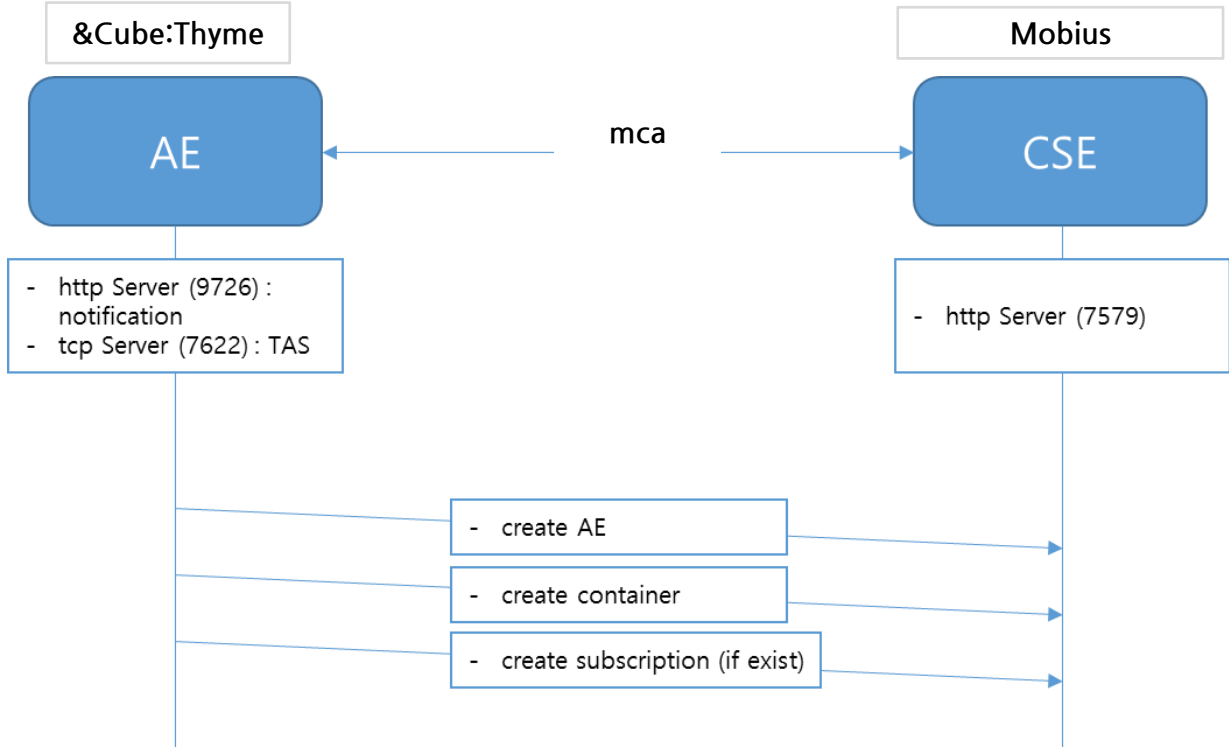
A terminal window titled "pi@raspberrypi: ~/node/thyme_tas" with standard window controls (minimize, maximize, close) in the top right. The terminal shows a shell prompt "pi@raspberrypi ~/node/thyme_tas \$" with a green cursor. The rest of the terminal area is black and empty.

```
pi@raspberrypi: ~/node/thyme_tas $
```

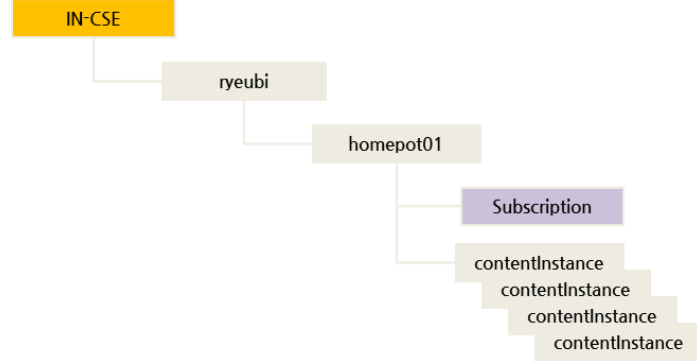
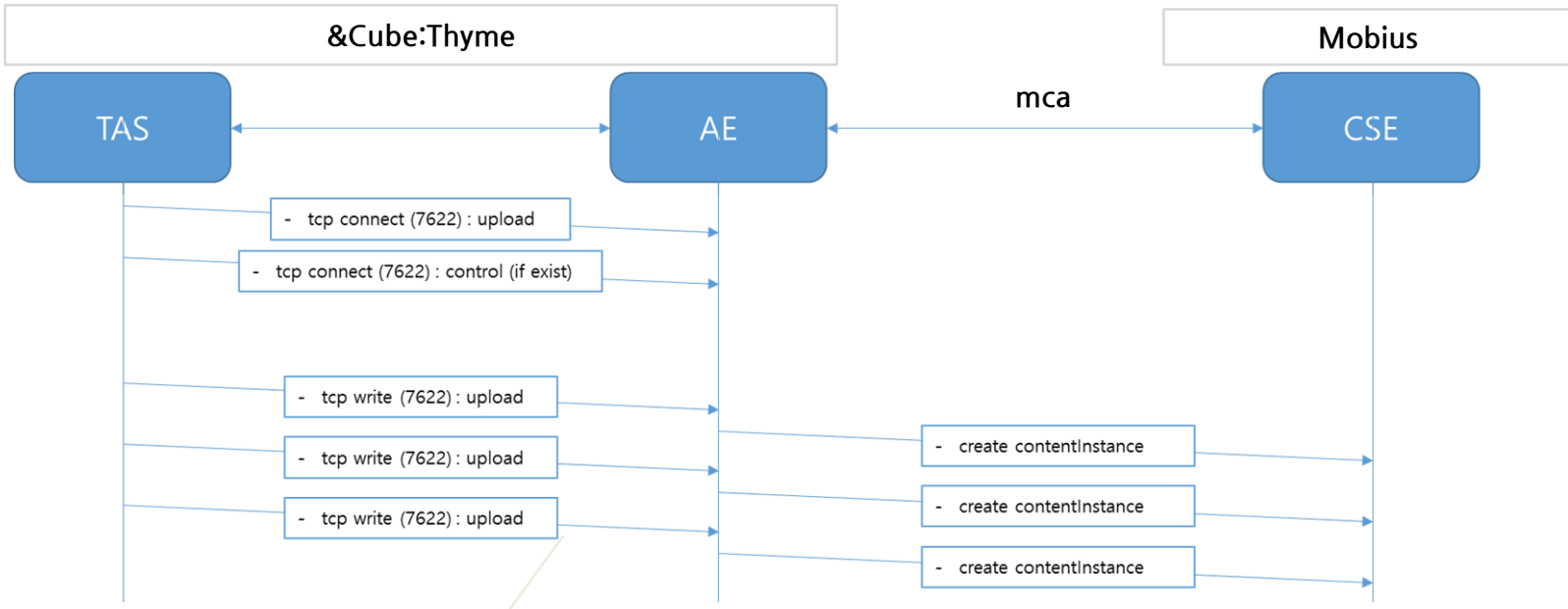
Install Node.js libraries of the working project and run TAS

```
>> root@raspberrypi ~/node/thyme_tas $ sudo npm install  
>> root@raspberrypi ~/node/thyme_tas $ node app.js
```


AE-CSE (&Cube-Mobius) MSC Example



TAS (Thing Adaptation Software) Example



{“ctname”:”<Target container name>”, “con”:”<data>”}

예)

{“ctname”:”cnt_1”, “con”:”TAS2015”}

{“ctname”:”temp”, “con”:”29.8”}

{“ctname”:”temphumi”, “con”:”29.8, 47”}

{“ctname”:”4491”, “con”:”29.8, 47”}

Configure TAS and run TAS

```
pi@raspberrypi ~/node/tas_co2 $ nano conf.xml
```

```
pi@raspberrypi ~/node/tas_co2 $ node app.js
```

```

pi@raspberrypi: ~/node/thyme_tas
GNU nano 2.2.6 File: conf.xml
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<m2m:conf xmlns:m2m="http://www.onem2m.org/xml/protocols" xmlns:
  <tas>
    <comport>/dev/ttyUSB0</comport>
    <baudrate>115200</baudrate>
    <parenthostname>localhost</parenthostname>
    <parentport>3105</parentport>
  </tas>
  <upload>
    <ctname>ss_1</ctname>
    <id>fe80:0000:0000:0000:0212:4b00:0235:bc89</id>
  </upload>
  <download>
    <ctname>ss_1_ctrl</ctname>
    <id>fe80:0000:0000:0000:0212:4b00:0235:bc89</id>
  </download>
</m2m:conf>

```

```

pi@raspberrypi: ~/node/thyme_tas
pi@raspberrypi ~/node/thyme_tas $ node app.js
port open. Data rate: 115200
{"con": "0.00W,1,6245", "id": "fe80:0000:0000:0000:0212:4b00:0235:bba0"}
Tas_develop_test
{"con": "0.00W,1,6246", "id": "fe80:0000:0000:0000:0212:4b00:0235:bba0"}
Tas_develop_test
upload Connected
download Connected - control_test_container hello
reconnect
Received: {"ctname": "control_test_container", "con": "hello"}
{"con": "0.00W,1,6247", "id": "fe80:0000:0000:0000:0212:4b00:0235:bba0"}
Tas_develop_test
ACK : {"ctname": "test_container", "con": "2001"} <----
{"con": "0.00W,1,6248", "id": "fe80:0000:0000:0000:0212:4b00:0235:bba0"}
Tas_develop_test
ACK : {"ctname": "test_container", "con": "2001"} <----
{"con": "0.00W,1,6249", "id": "fe80:0000:0000:0000:0212:4b00:0235:bba0"}
Tas_develop_test
ACK : {"ctname": "test_container", "con": "2001"} <----

```

3. 사물인터넷 디바이스 개발 실습

3.1 &Cube: Thyme 구동 실습

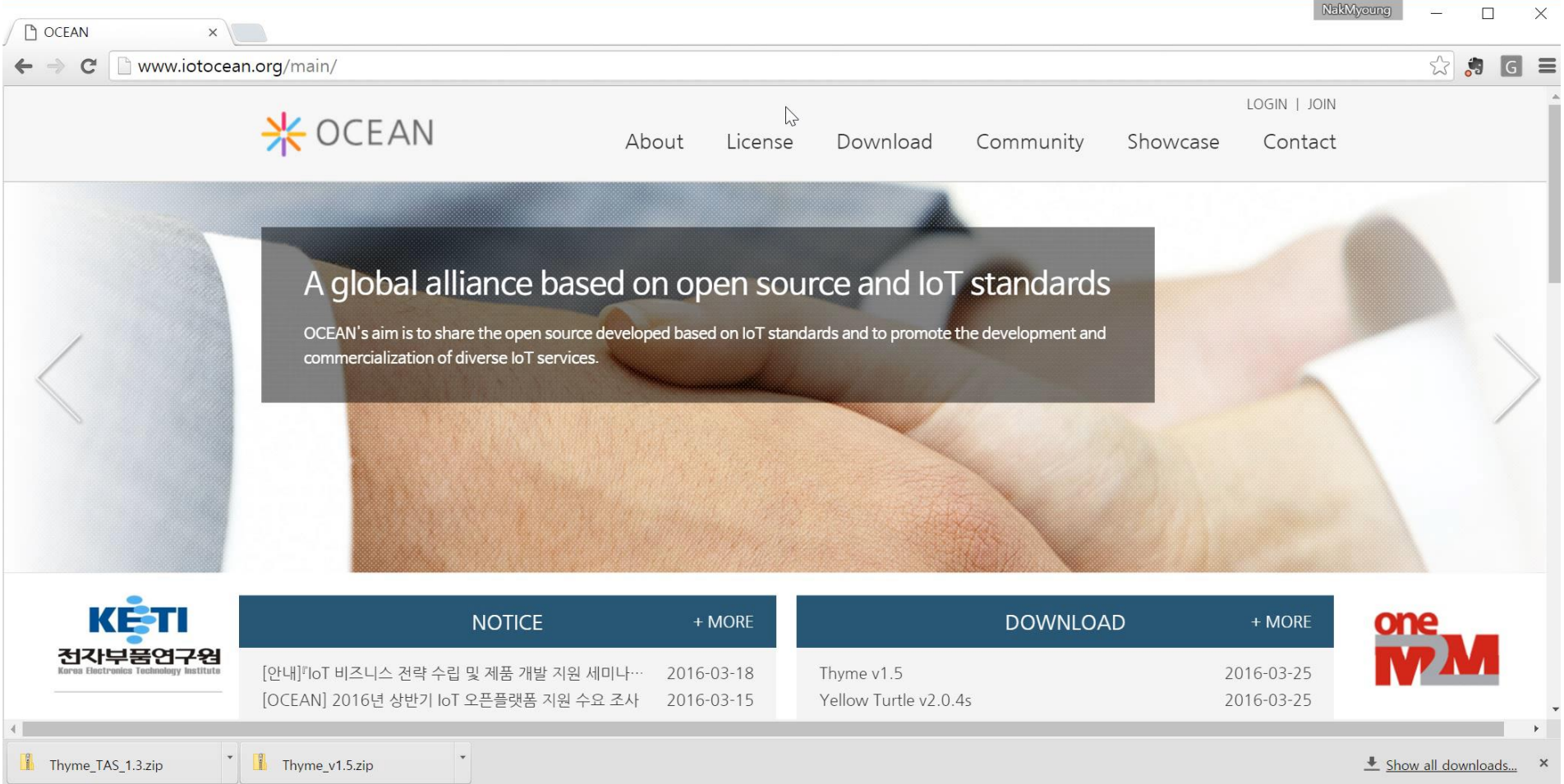
3.2 &Cube:TAS 개발 실습

3.3 Test Device

3.4 서비스 시나리오

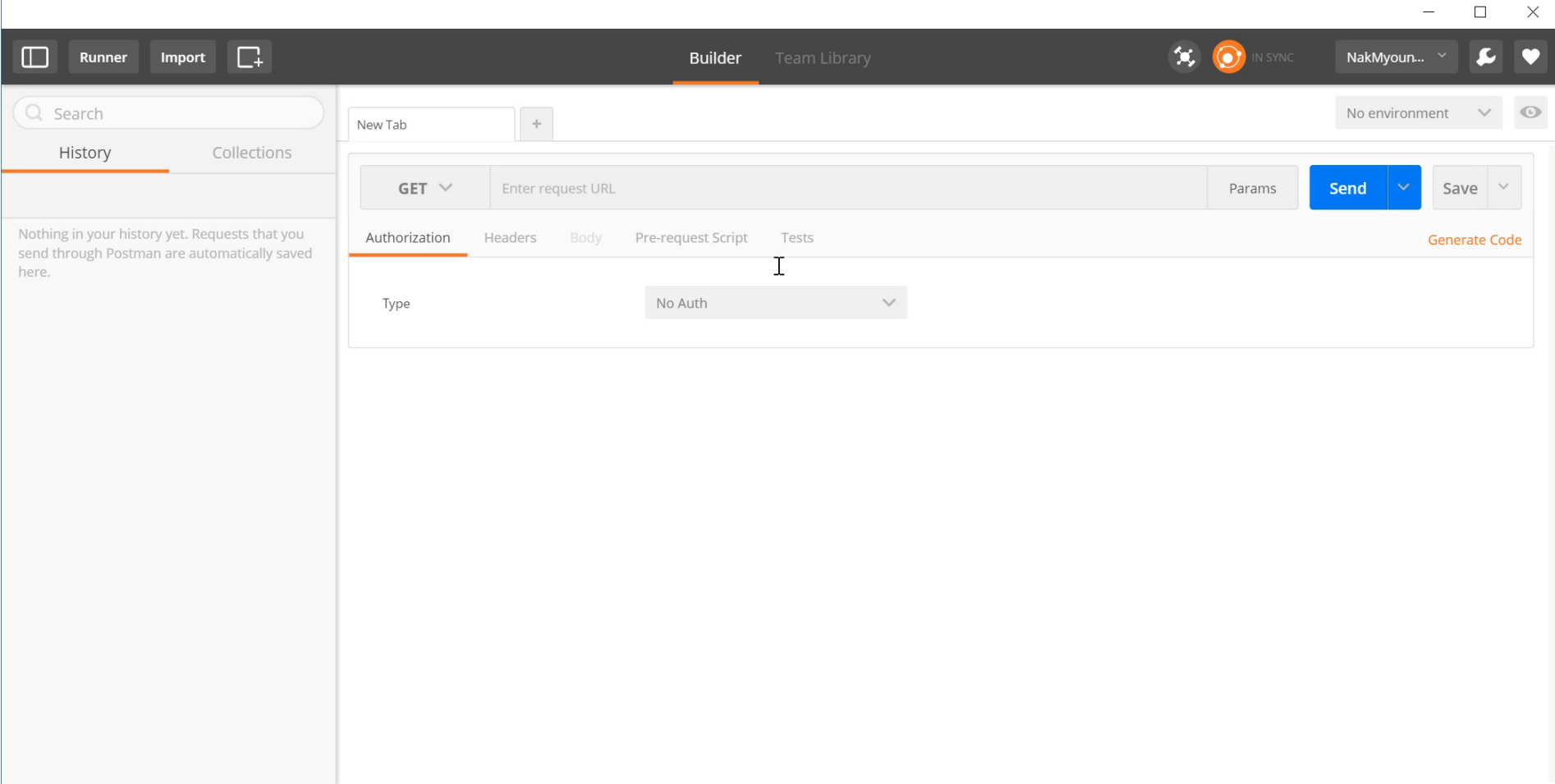
3.5 Yellow Turtle 구축 실습

Download Mobius API Collection



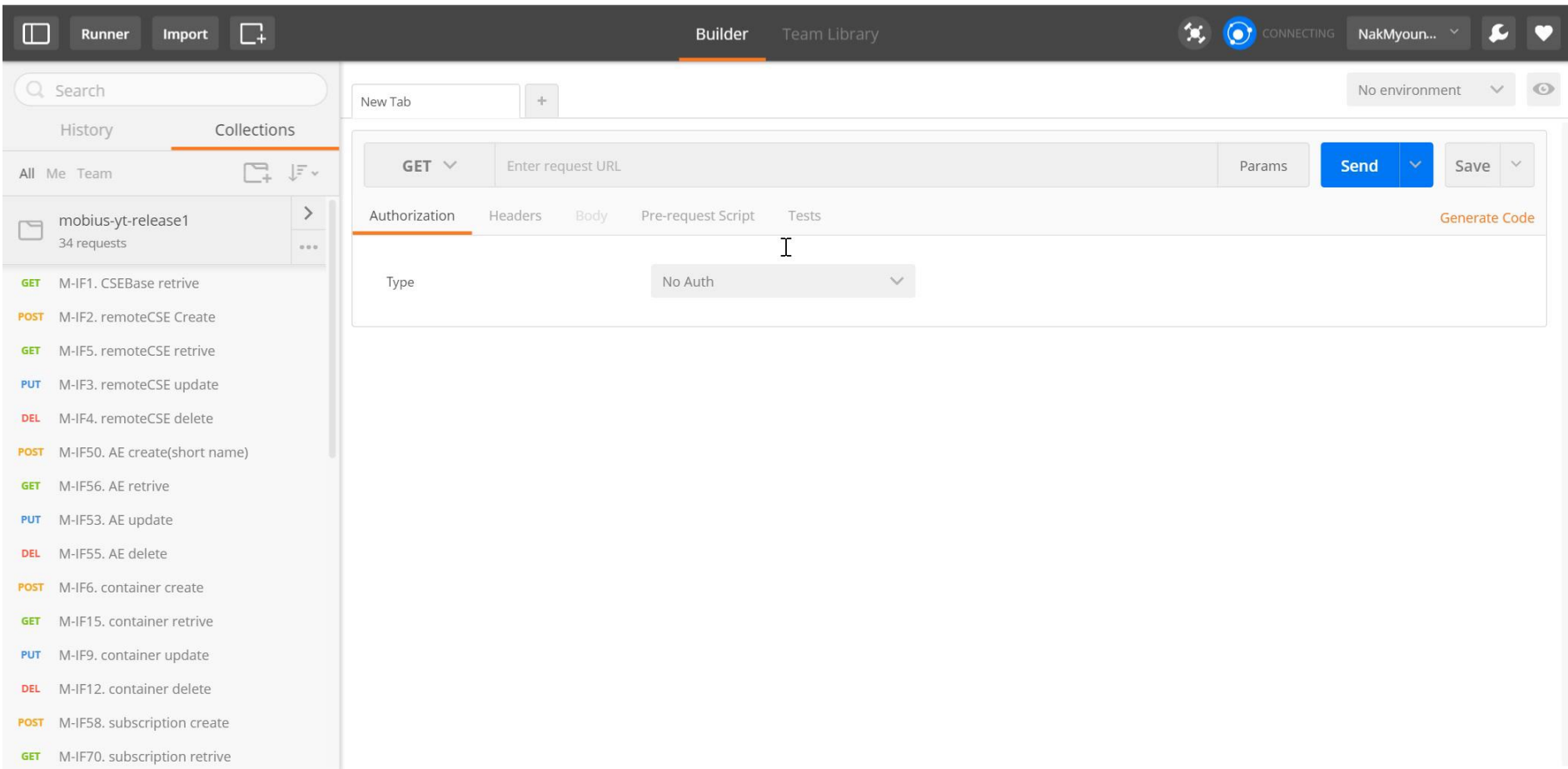
Download Mobius API Collection ([Download](#) → [Mobius](#) → [Yellow Turtle](#))
 Version example: `mobius-yt-releaseXXX.json.postman_collection`

Install Test Program: Install Postman REST Client & Import Collection



Import Test API collection
Collections → Import Collection
Upload files → Choose "mobius-yt-releaseXXX.json.postman_collection" → Import

Test Device: Data Retrieval



Data retrieval example using Mobius API
Collections → mobius-yt-releaseXXX → contentInstance retrieve

Test Device: Control Request



Builder Team Library

mp_url

Instances +

URL: `{{mp_url}}/mobius-yt/thyme_test/test_container/latest` Params Send Save

Content-Type	application/xml	≡	×	Presets
Content-Length	12345	≡	×	
Origin	Origin	≡	×	
Accept	short	≡	×	
Value	value			Bulk Edit

Cookies Headers (8) Tests I Status: 200 OK Time: 210 ms

Raw Preview XML Save Response

```

<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<cin xmlns:m2m="http://www.onem2m.org/xml/protocols" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" rn="4-201603250758470680kGo">
  <pi>/mobius-yt/thyme_test/test_container</pi>
  <ty>4</ty>
  <ri>/mobius-yt/thyme_test/test_container/4-201603250758470680kGo</ri>
  <ct>20160325T075847</ct>
  <lt>20160325T075847</lt>
  <st>0</st>
  <mni>18446744073709551615</mni>
  <cs>12</cs>
  <con>0.00W,0,1171</con>
</cin>
  
```

Control request example using Mobius API
 Collections → mobius-yt-releaseXXX → contentInstance create

3. 사물인터넷 디바이스 개발 실습

3.1 &Cube:Thyme 구동 실습

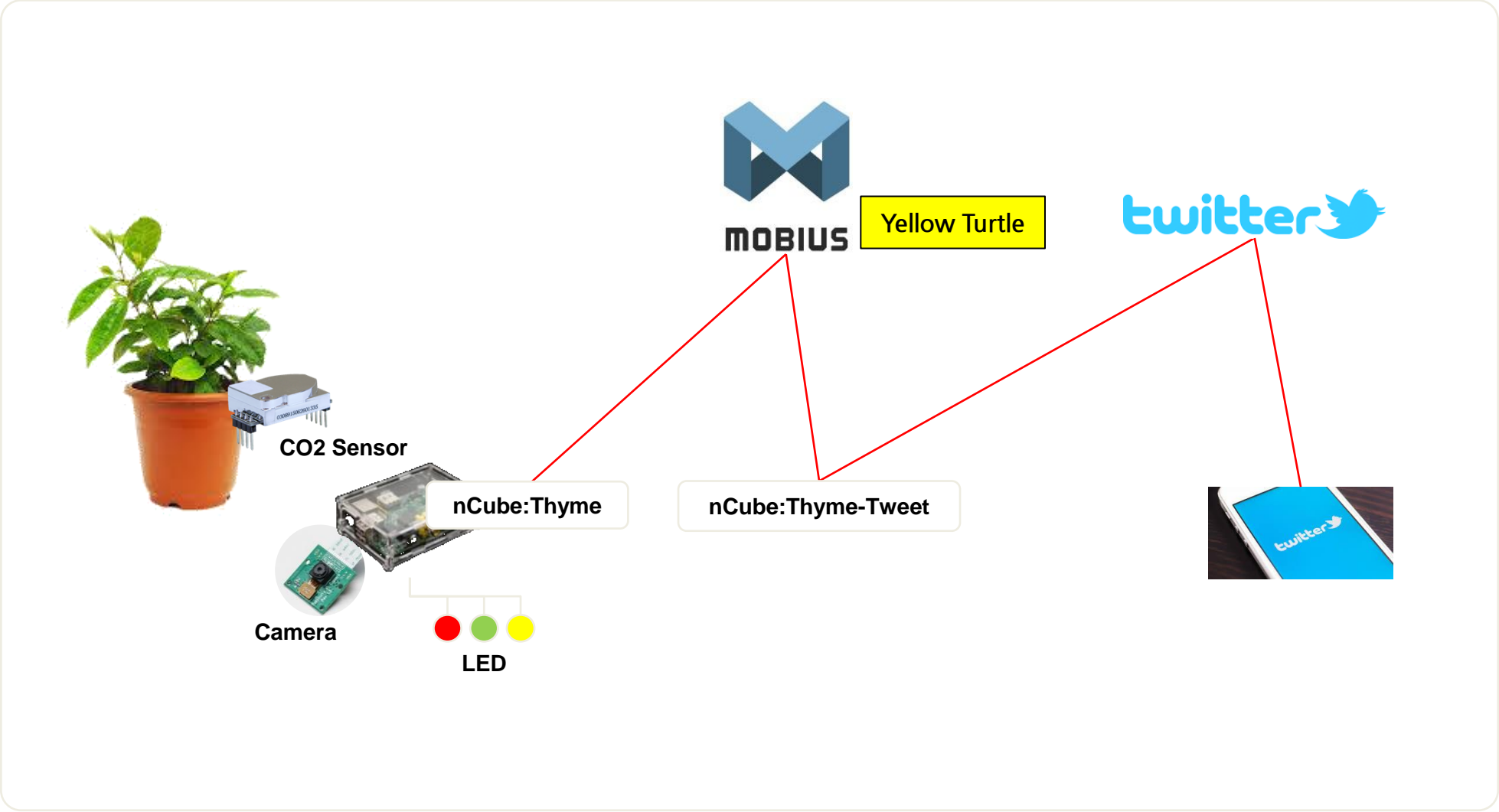
3.2 &Cube:TAS 개발 실습

3.3 Test Device

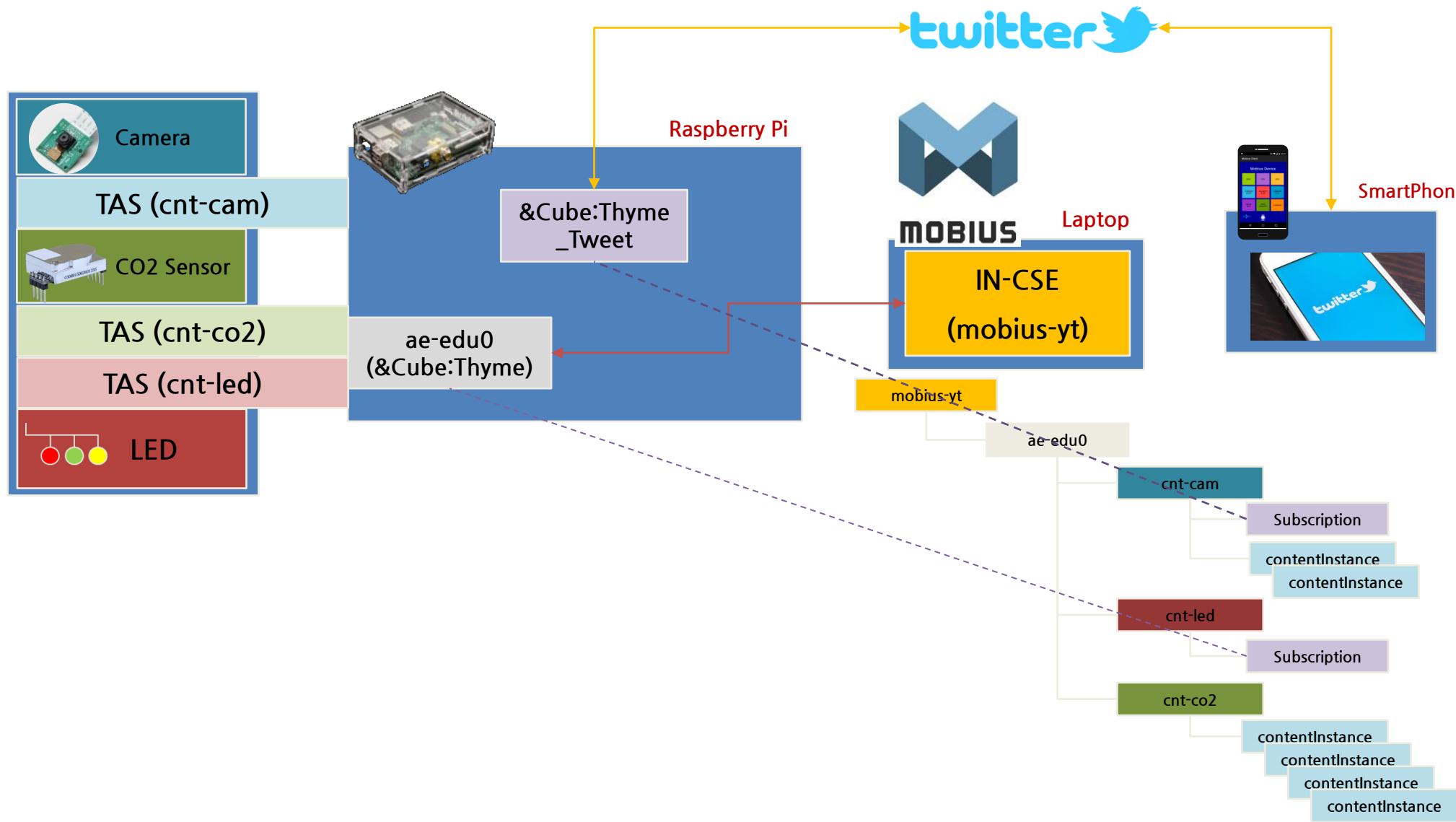
3.4 서비스 시나리오

3.5 Yellow Turtle 구축 실습

Case study for Smart Pot Service (Demo)



Configuration for Smart Pot Service (Demo)

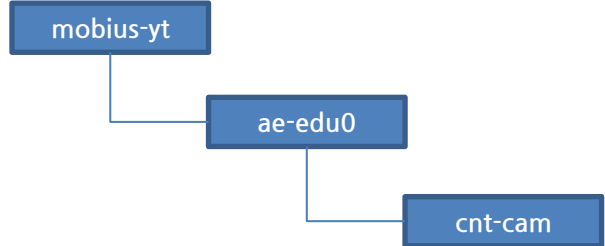
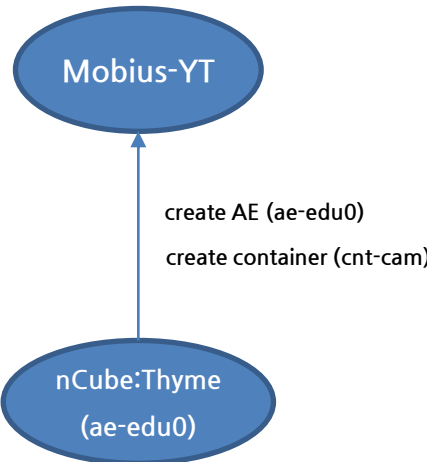


ToT (Tweet of Thing)

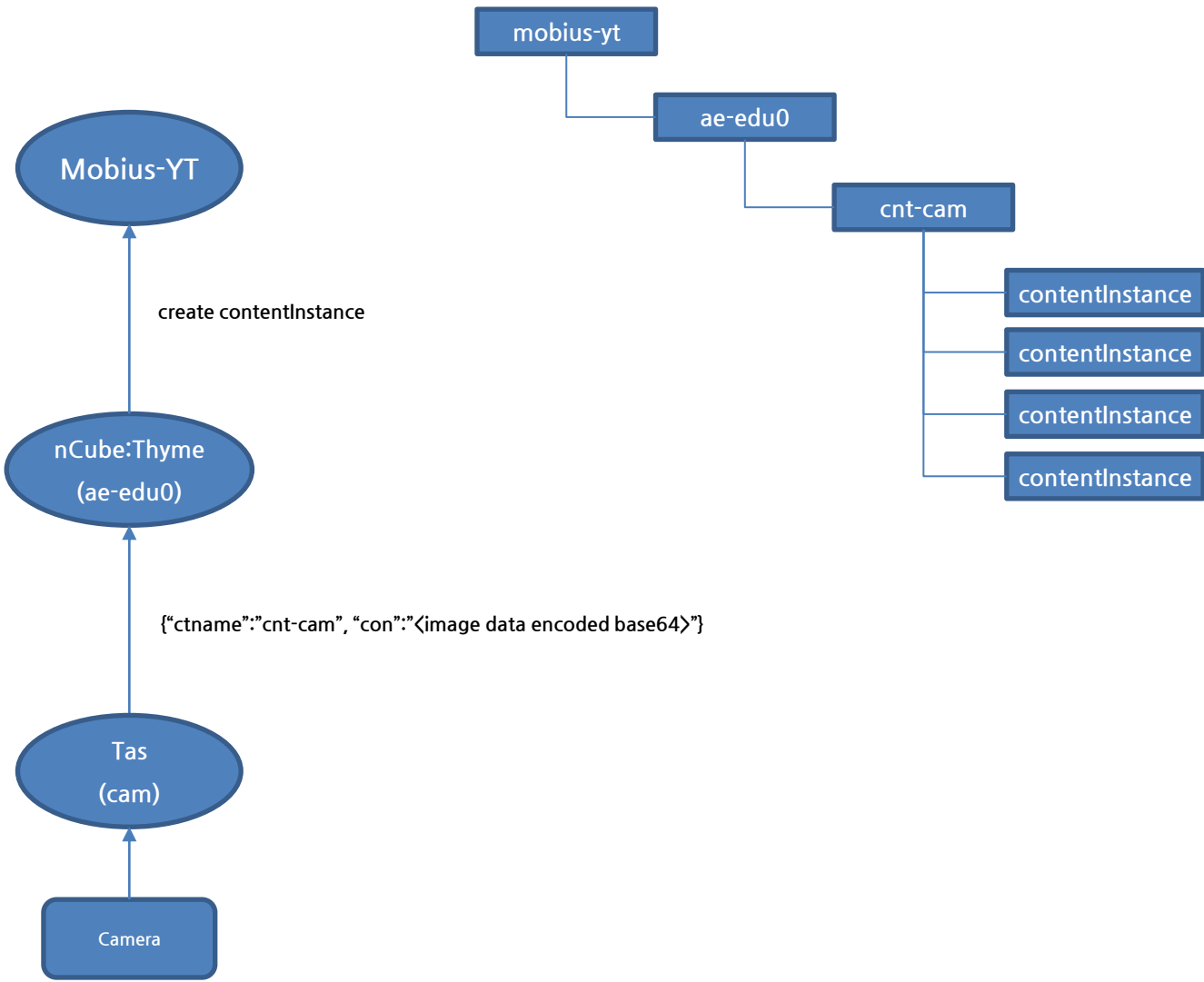
Mobius-YT

mobius-yt

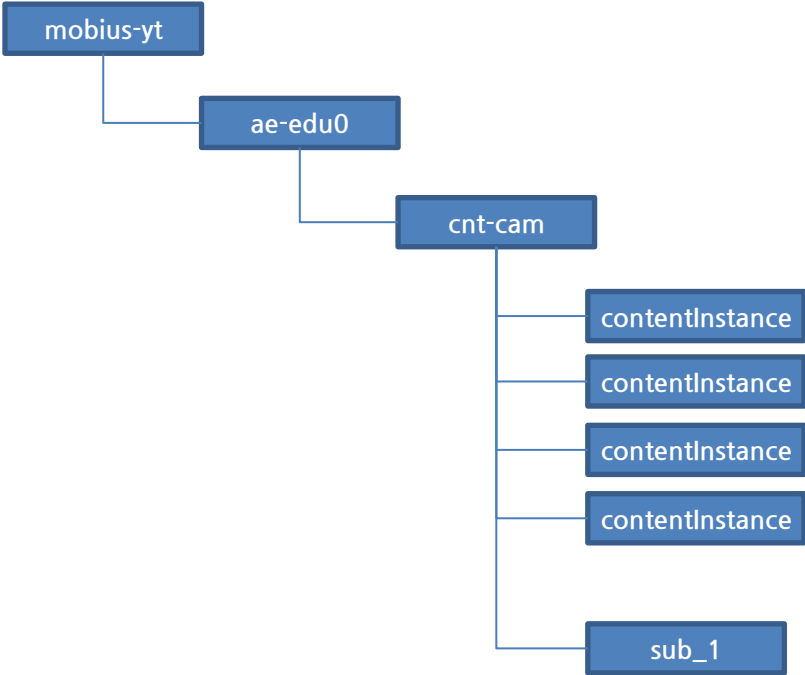
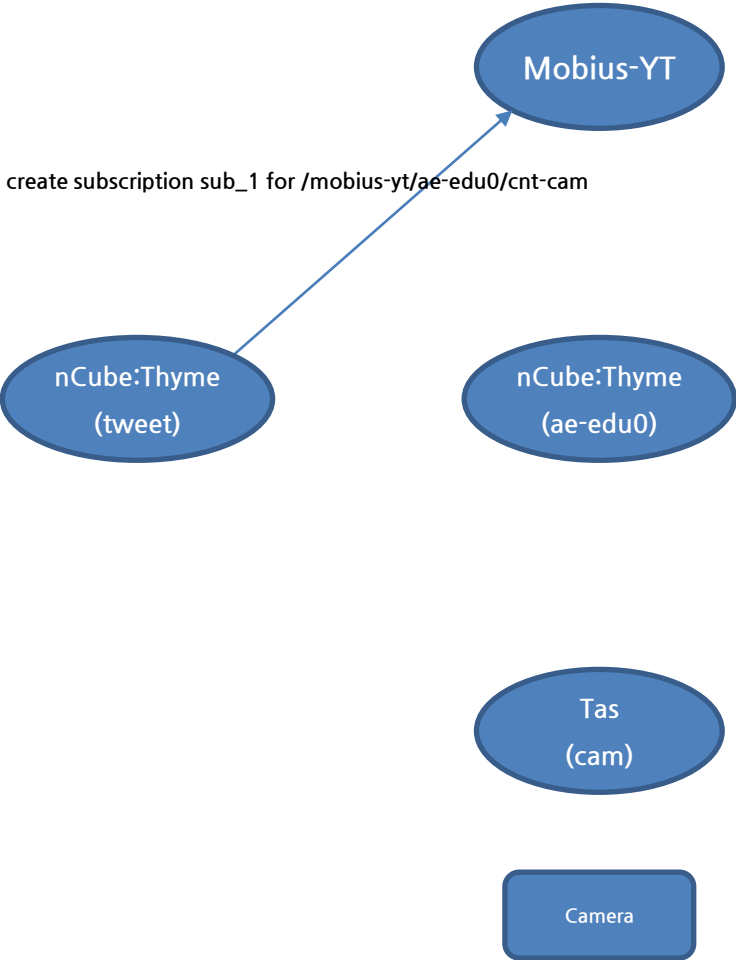
ToT (Tweet of Thing)



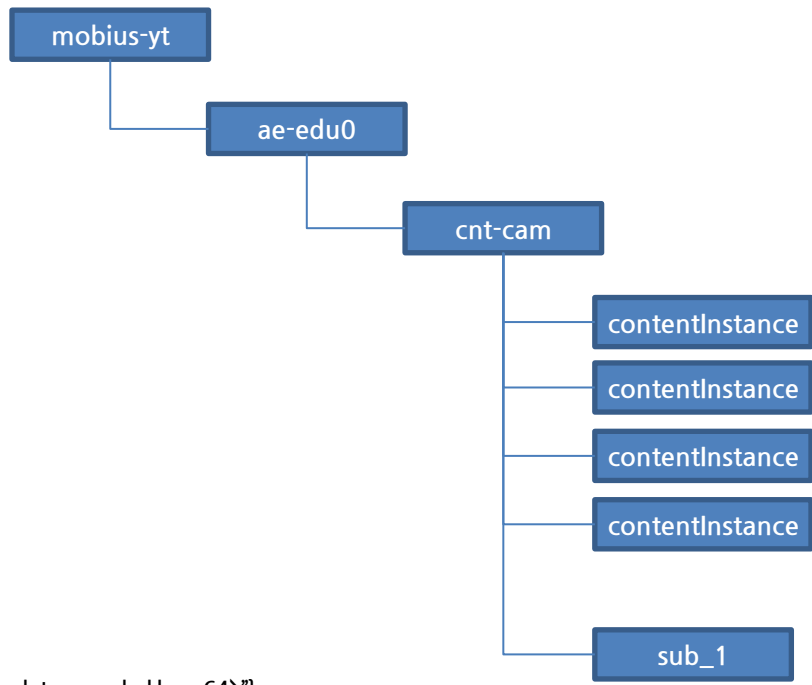
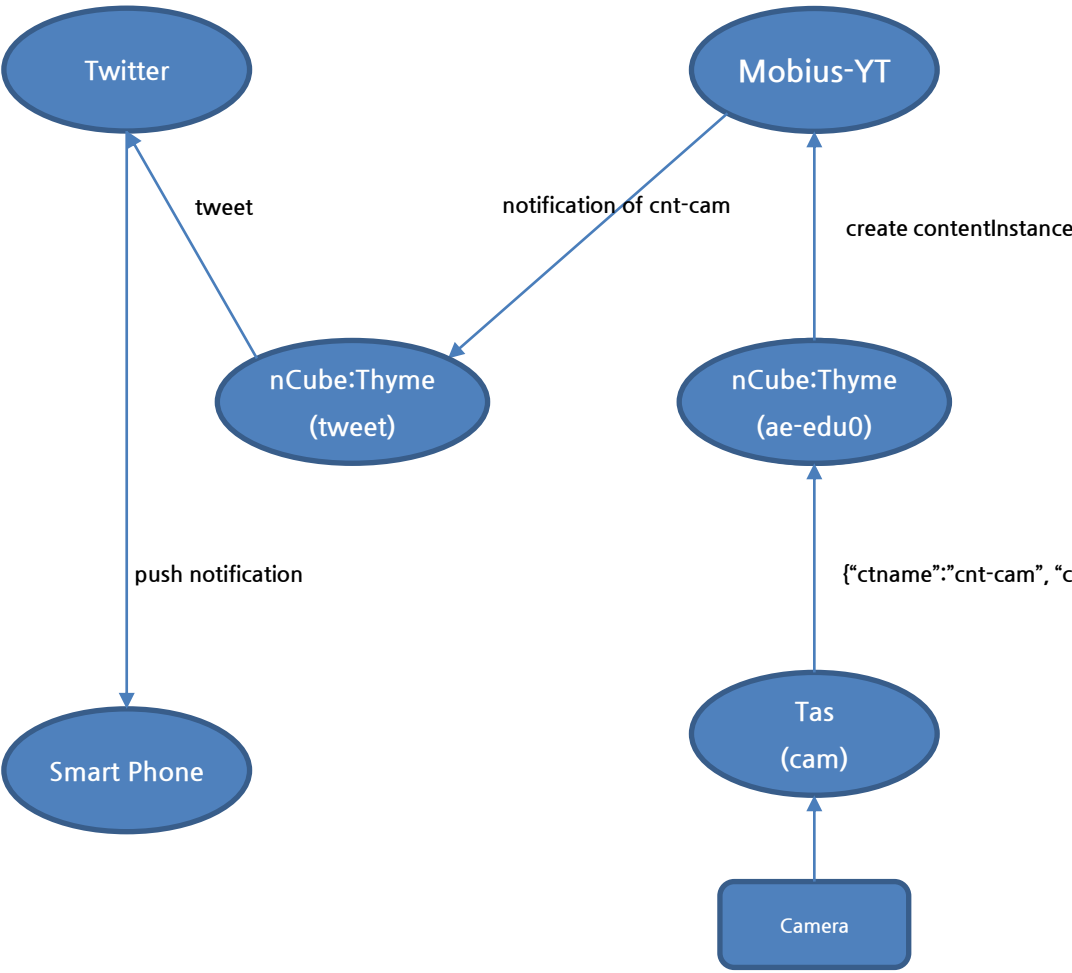
ToT (Tweet of Thing)



ToT (Tweet of Thing)



ToT (Tweet of Thing)



3. 사물인터넷 디바이스 개발 실습

3.1 &Cube: Thyme 구동 실습

3.2 &Cube:TAS 개발 실습

3.3 Test Device

3.4 서비스 시나리오

3.5 Yellow Turtle 구축 실습

Construction Server Platform with Mobius : Yellow Turtle

■ Mobius-YT 서버 설치

- OCEAN Alliance 사이트 (<http://www.iotocean.org>)

- Download Mobius:Yellow Turtle from OCEAN

Download
Open allianCE for iot stANDARD

Download - Yellow Turtle

Latest Version

- Mobius >
- Yellow Turtle
- Blue Octopus
- &CUBE
- Interworking
- Tools
- App Sample
- Old
- Open Contribution

Yellow Turtle v2.1.2 2016-08-17 15:45

Mobius: Yellow Turtle

Yellow Turtle is an open source software of oneM2M-based IoT Server Platform based on Node.js JavaScript.

The source code and files of Yellow Turtle are under the OCEAN license terms and conditions, i.e., 3-clause BSD open source license.

Versions

Code Name	Framework	Version	Ref. Standards
Yellow Turtle	Node.js	2.1.2	oneM2M Release 1

PREREQUISITES

-

SYSTEM REQUIREMENTS

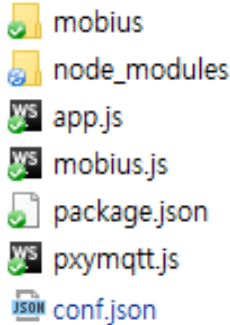
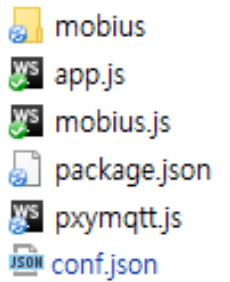
System Requirements	Remarks
Operating System	WindowsX, Linux Redhat and CentOS, Mac, Rasbian
Open Source Framework	Node.js
Web Application Server	Node.js
Database	MySQL
CoAP Framework	
MQTT Broker	Mosquitto 1.4.x

Files

Name	Download Link
Mobius Installation Guide Korea	Installation Guide_Mobius_Yellow_Turtle_v2.1_...
Mobius Yellow Turtle Source 2.0.12	mobius-yt-2.1.2.zip
POSTMAN Environment Script	mobius-yt-7579_postman_environment.json
POSTMAN Script	mobius-yt-release1_postman_collection.json
	test-as-virtual-device_postman_collection.json
	test-for-group-resource-fanoutpoint_postman_collection.json

Construction of Mobius:Yellow Turtle

- Extract zip and run 'npm install' in command prompt at the folder of mobius



```
명령 프롬프트
finalhandler@0.4.0 (unpipe@1.0.0)
proxy-addr@1.0.8 (forwarded@0.1.0, ipaddr.js@1.0.1)
send@0.13.0 (destroy@1.0.3, statuses@1.2.1, ms@0.7.1, http-errors@1.3.1,
mime@1.3.4)
type-is@1.6.7 (media-typer@0.3.0, mime-types@2.1.5)
accepts@1.2.12 (negotiator@0.5.3, mime-types@2.1.5)

xmlbuilder@2.6.4 node_modules\xmlbuilder
└── lodash@3.10.1

mqtt@1.4.0 node_modules#mqtt
├── inherits@2.0.1
├── xtend@4.0.0
├── minimist@1.2.0
├── readable-stream@1.0.33 (isarray@0.0.1, string_decoder@0.10.31, core-util-
is@1.0.1)
├── commist@1.0.0 (leven@1.0.2)
├── mqtt-packet@3.3.1 (bl@0.9.4)
├── end-of-stream@1.1.0 (once@1.3.2)
├── help-me@0.1.0 (pump@1.0.0)
├── concat-stream@1.5.0 (typedarray@0.0.6, readable-stream@2.0.2)
├── mqtt-connection@2.1.1 (through2@0.6.5, reduplexer@1.1.0)
├── websocket-stream@2.0.2 (through2@0.6.5, duplexify@3.4.2, ws@0.8.0)

C:\Users\ryeubi\Dropbox\Downloads\mobius yellow turtle>node app.js
server running at 7579 port
[IC create] - 2015-09-03T01:29:50+09:00
```

Construction of Mobius:Yellow Turtle

- Setting of configuration file (conf.json) for mobius
 - Mobius 정보 설정
 - csebaseport : 서버가 오픈하는 포트 번호
 - dbpass: DB 접속 암호

```
{  
  "csebaseport": "7579",  
  "dbpass": "dksd1fduq2"  
}
```

Conclusion - OCEAN (Open alliance for iot stANdard) - <http://iotocean.org>

Find, Connect, Control

Dream, Do, Realize

OCEAN

2 billion people connected

50 billion things connected

Member logos include: KEITI, NAVER, BIT&PULSE, NST, obih, M2MDS, TOT, innoPia, GNS, Fsystems, KAON, A3SECURITY, QRZ, KNU, Shimera, EinsWare, REAL-TIME, LG, DONGNAM EnS, Axstone, DC, SystemBase, NEXUSCHIPS, DYG, THOMNET, WON, ID-Fone, AP전자(주), AJANTECH, KOPTI, 세종대학교, SEIL Technology, R@remote Solution, dadam MICRO INC., LG, WON, ID-Fone.

To Be Continue...