

# oneM2M 기반의 디바이스 플랫폼 &Cube 설치 및 실행 가이드

2015 8월

경북대학교 통신프로토콜연구실

정중화, 강형우

[godopu16@gmail.com](mailto:godopu16@gmail.com), [hwkang0621@gmail.com](mailto:hwkang0621@gmail.com)

## 요 약

사물인터넷 (Internet of Things: IoT)의 시대가 도래하고 있다. 다양한 분야에서 사물인터넷을 활용한 서비스가 개발되고 있다. 본 문서에서는 사물인터넷 국제표준인 oneM2M을 기반으로 만들어진 사물인터넷 디바이스 플랫폼인 &Cube에 대한 설명과 함께 설치 및 구동 방법에 대하여 설명하도록 한다.

## 목 차

1. 서론 .....	2
2. &CUBE란 .....	2
3. &CUBE 설치 및 실행 .....	4
3.1 라즈비안(RASPBIAN 설치) .....	4
2.2 JAVA JDK 설치 .....	14
2.3 &CUBE 설치 및 실행 .....	14
4. 결론 .....	25
참고 문헌 .....	26

## 1. 서론

요즘 사물인터넷이란 말을 어렵지 않게 들을 수 있다. 사물인터넷이라는 개념이 널리 퍼지면서 다양한 분야에서 사물인터넷을 활용한 서비스들이 개발되고 공개되고 있으며, 사물인터넷 표준화 활동 역시 활발히 진행되고 있다. 우리나라에서도 사물인터넷을 위한 연구가 활발히 진행되고 있으며, KETI, nTels 등에서 개발한 국제표준 기반의 플랫폼인 Mobius와 &Cube가 바로 그 중심에 있다.

본 문서에서는 KETI에서 개발한 사물인터넷 국제표준 oneM2M 기반의 디바이스 플랫폼인 &Cube에 대하여 설명하도록 하겠다.

## 2. &Cube란

&Cube는 사물인터넷 디바이스/게이트웨이에 탑재되는 S/W 플랫폼으로 디바이스에 연결된 사물을 개방형 사물인터넷 서버 플랫폼 Mobius와 연동할 수 있도록 지원하는 플랫폼이다. 6개의 core block으로 구성되어 &Cube란 이름이 지어졌으며 디바이스의 종류에 따라 4가지 버전을 지원한다. 해당 파일은 사물인터넷 Alliance인 Ocean 사이트에서 회원가입 후 다운받을 수 있으며, 현재 사물인터넷 국제표준인 oneM2M Release1 버전 기반으로 Lavender v1.0을 공개하였다.

- &Cube: Rosemary: 게이트웨이 버전 S/W 플랫폼
- &Cube: Lavender: 디바이스 버전 S/W 플랫폼
- &Cube: Chamomile: CoAP 지원 S/W 플랫폼
- &Cube: Mint: 초 경량 디바이스 버전 S/W 플랫폼

Figure 1. &Cube 버전

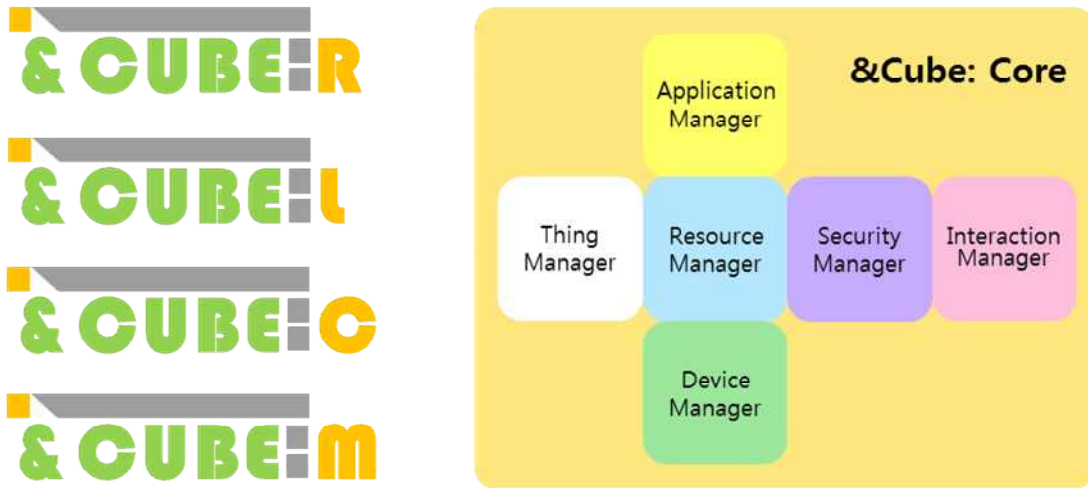


Figure 2. &Cube Core Block

&Cube는 IoT 디바이스 속에서 여러 IoT 기능을 수행합니다. 예를 들면 각종 센서가 탑재된 다른 기기로부터 수집한 데이터를 Mobius 서버에 전송하기도 하며, Mobius로부터 명령을 받아 연결된 다른 기기들을 조정하는 역할을 수행하기도 한다.

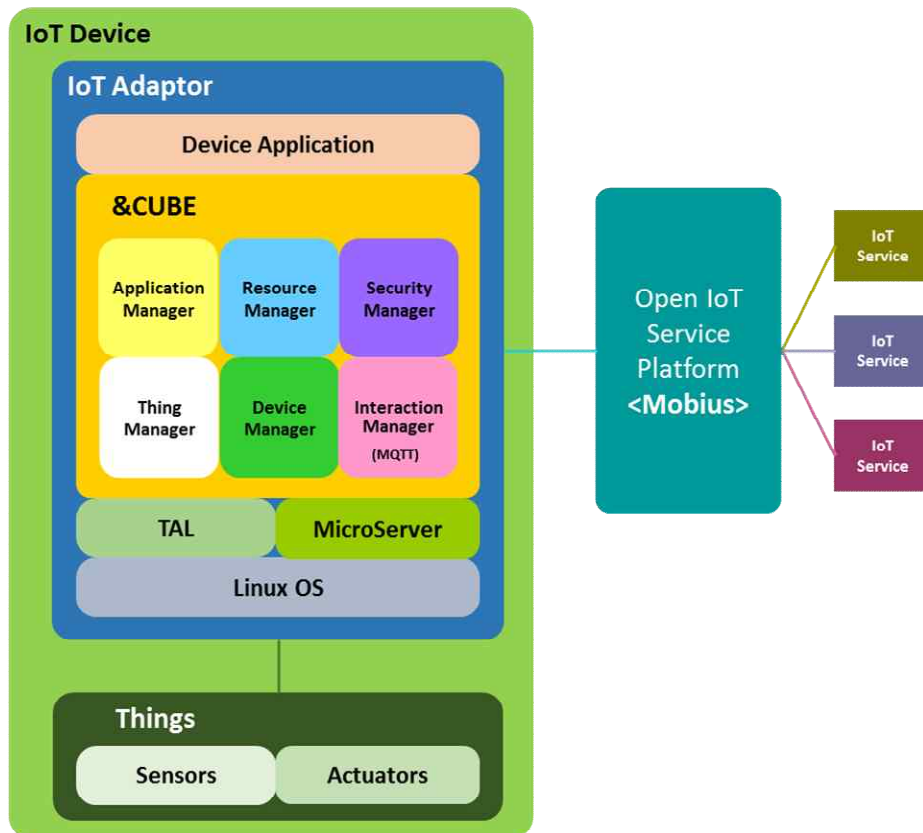


Figure 3. 전반적인 &Cube의 다이어그램

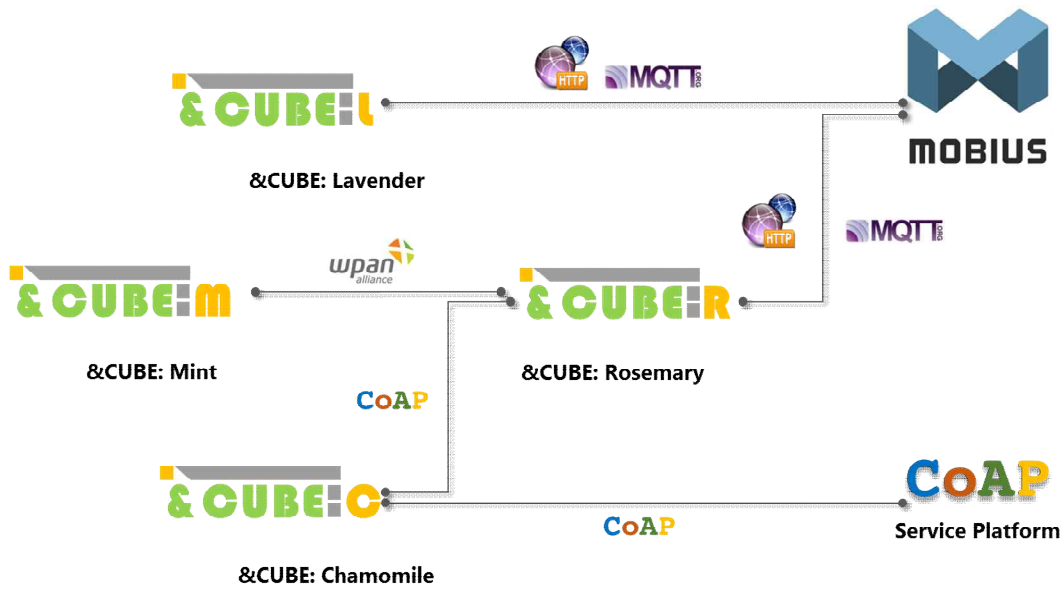


Figure 4. &Cube 버전 별 연동 구조

&Cube 플랫폼은 각 버전에 따라 다른 구성으로 Mobius와 통신한다. 일반 디바이스 버전인 Lavender는 HTTP REST와 MQTT 프로토콜을 활용하여 통신을 수행하고, 경량 디바이스 버전인 Mint와 CoAP 프로토콜 기반의 Chamomile은 각각 wpan, Bluetooth, zigbee 등의 인터페이스를 활용하여 게이트웨이 버전의 Rosemary와 통신을 수행한다. Rosemary 역시 HTTP REST 및 MQTT 프로토콜을 활용하여 통신을 수행한다.

본 문서에서는 Ocean에서 공개한 Lavender v1.0 만을 고려하였다.

### 3. &Cube 설치 및 실행

&Cube는 6개의 구성요소로 이루어져 있다. 각 구성요소는 모두 자바 패키지로 개발되었으며, 우리는 라즈베리파이에 &Cube Lavender를 설치하는 방법을 알아보기로 한다.

#### 3.1 라즈비안(raspbian 설치)

앞에서 말했듯이 라즈베리파이에 &Cube를 설치하기 위해서는 먼저 라즈베리파이에 raspbian 운영체제를 설치하여야 한다. 먼저 raspbian을 설치할 micro sd 카드가 필요한데, 가급적이면 4GB 이상의 sd카드를 사용하길 권장한다. micro sd 카드가 준비되었다면 <http://sourceforge.net/projects/win32diskimager> 사이트에 가서 win32diskmager.exe 프로그램을 다운로드 받는다. win32가 들어있지만 Win7 64비트 환경에서도 문제없이 돌아간다. 아

래의 그림을 참고하시면 쉽게 설치 하실 수 있다.

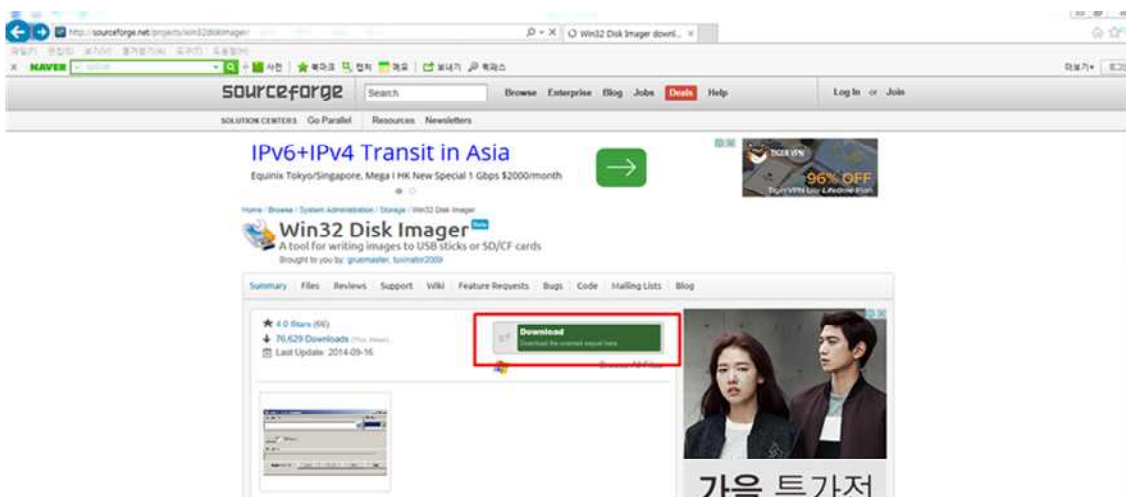


Figure 5. Win32DiskImager.exe 다운로드 1

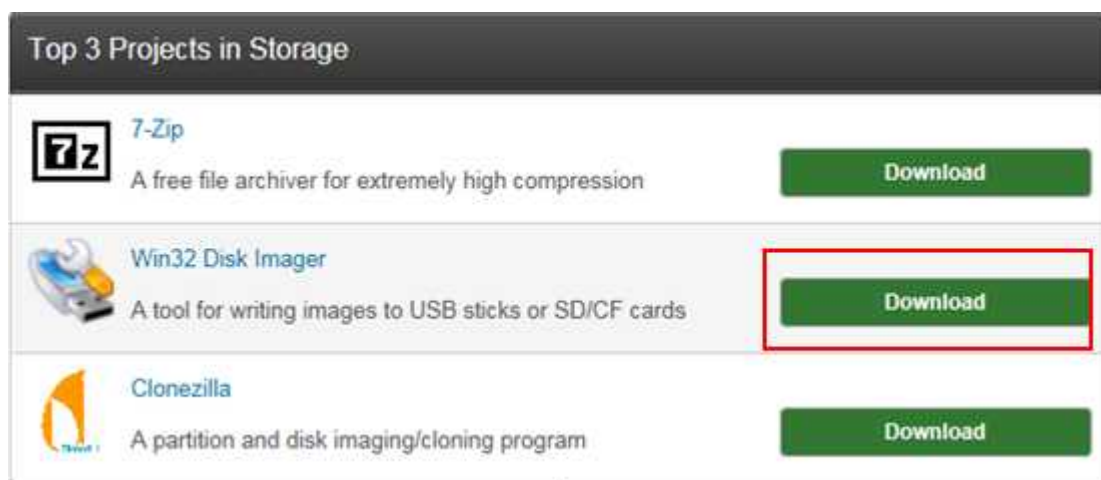


Figure 6. Win32DiskImager.exe 다운로드 2

win32diskimager.exe의 다운로드가 완료되었다면, 설치를 시작한다. 아래의 그림을 따라 설치한다.

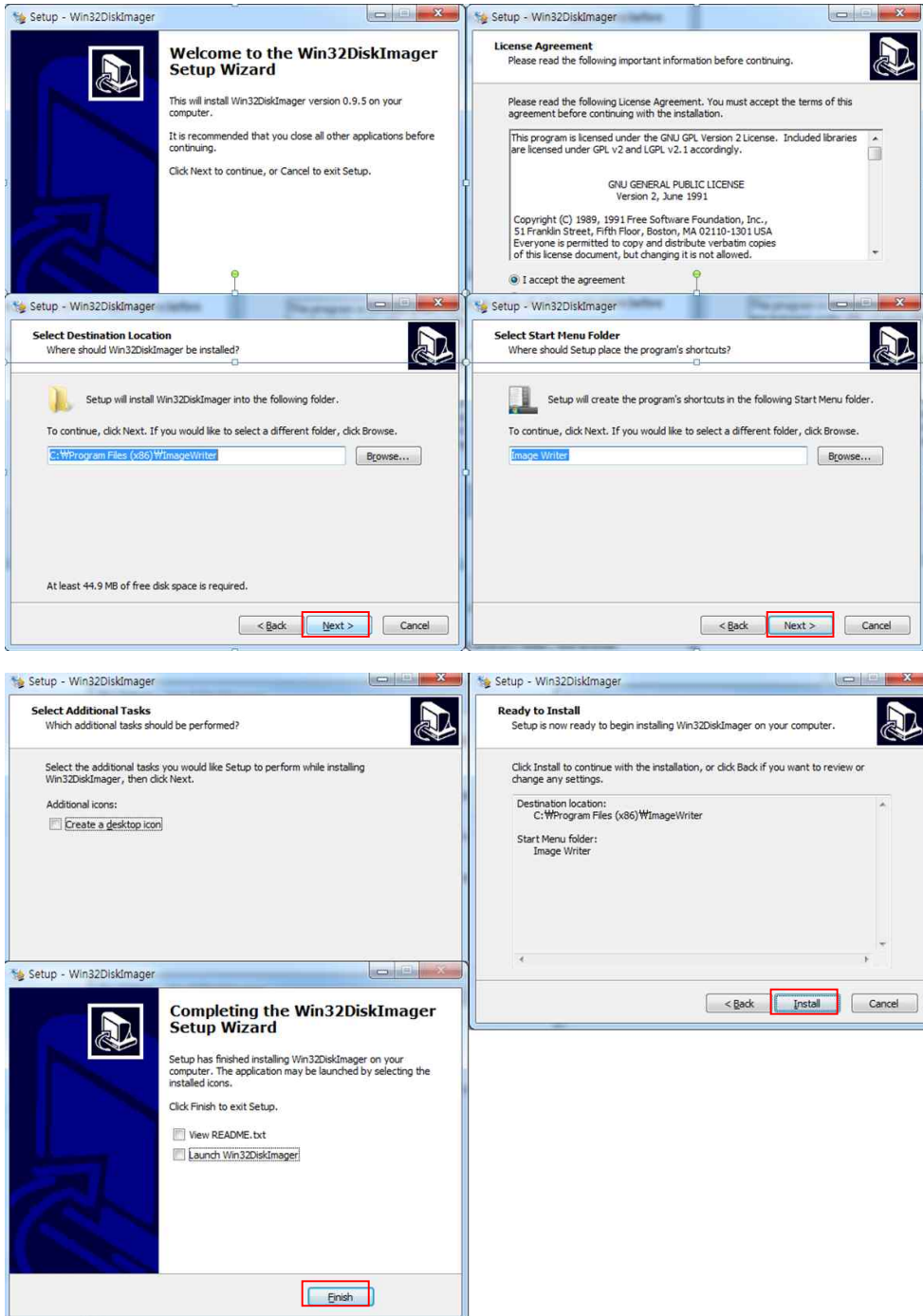


Figure 7. Win32DiskImager.exe 설치

win32diksimer.exe를 설치하였다면, <http://downloads.raspberrypi.org/> 에 가서 raspbian 이미지 파일을 다운로드를 한다.

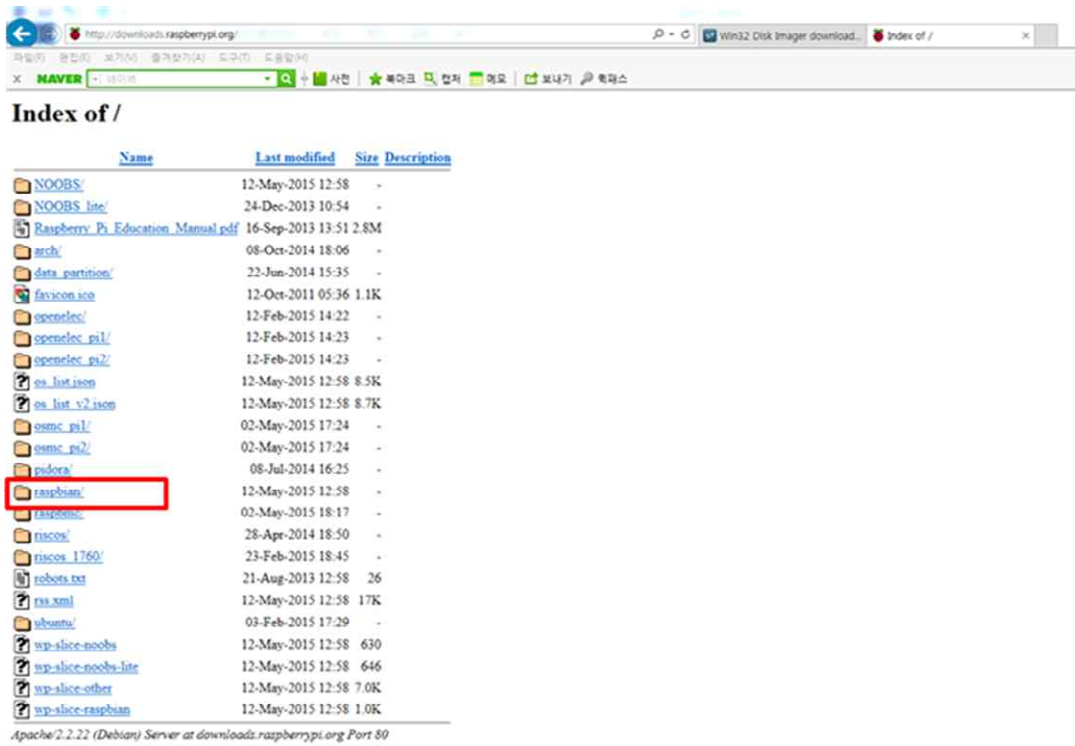


Figure 8. Raspbian 이미지 파일 다운로드 1

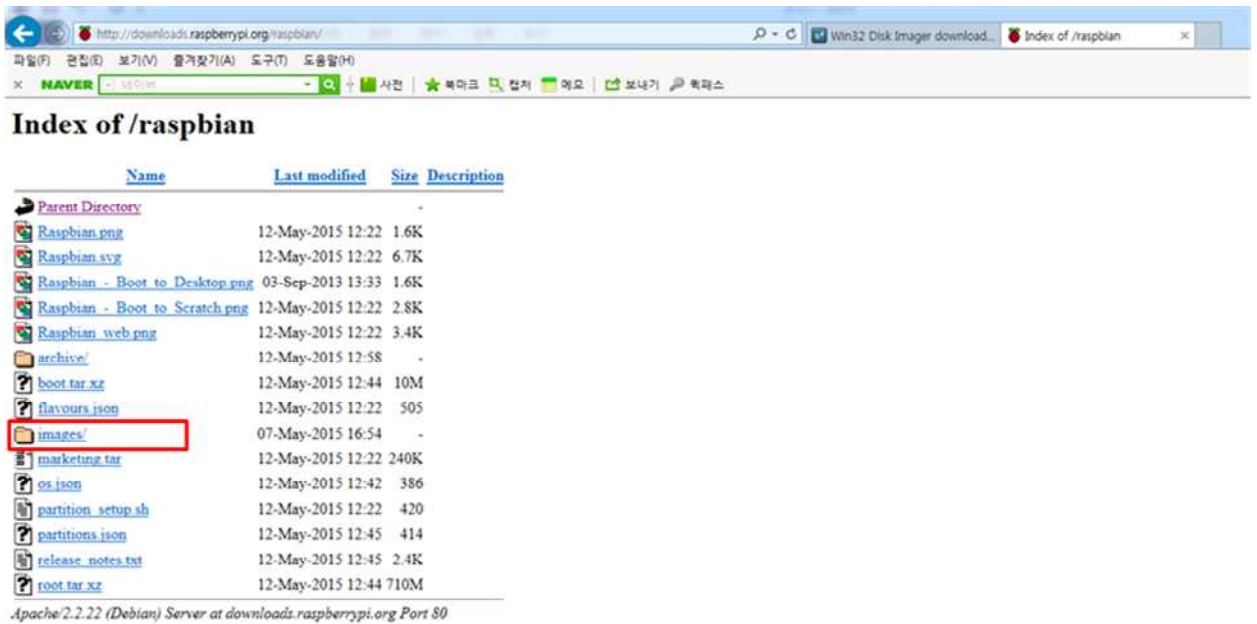


Figure 9. Raspbian 이미지 파일 다운로드 2

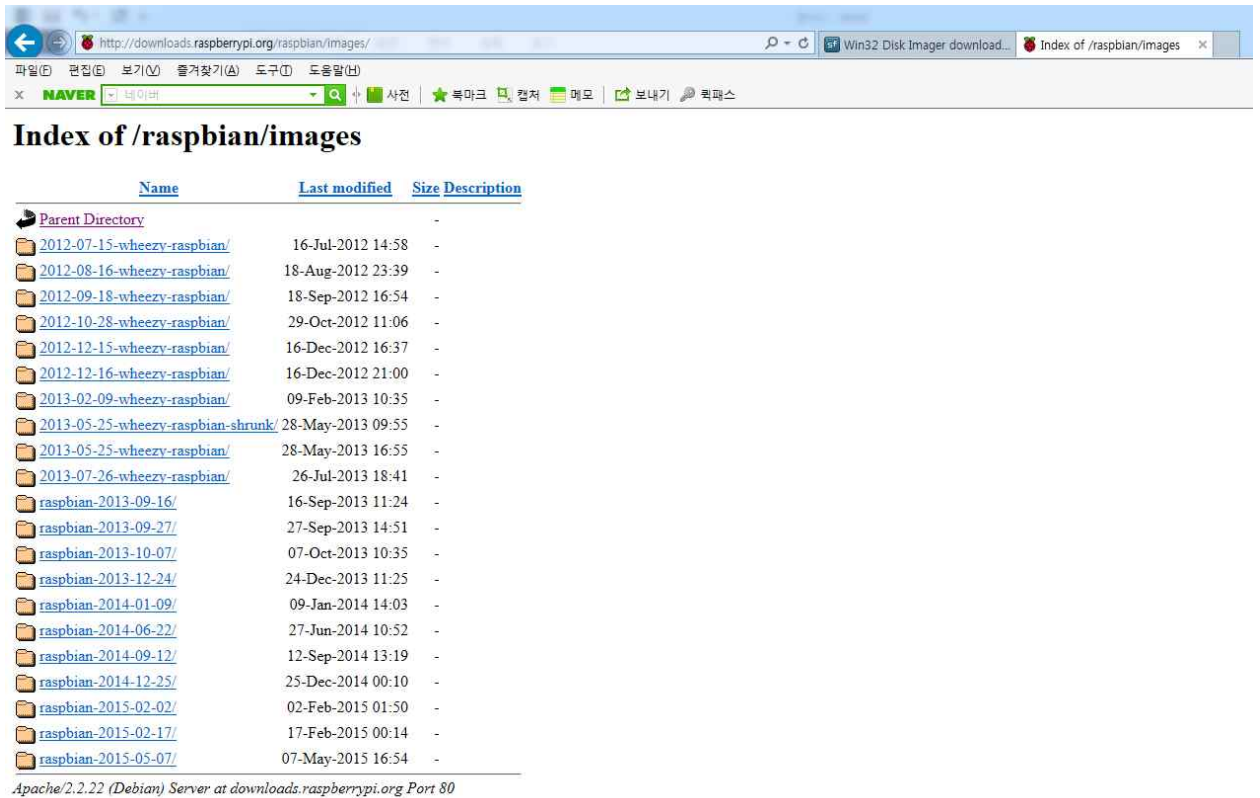


Figure 10. Raspbian 이미지 파일 다운로드 3

raspbian 을 다운로드 받았다면 win32DiskImager.exe 를 실행한 후 아래의 그림을 따라 SD 카드에 raspbian을 설치한다.

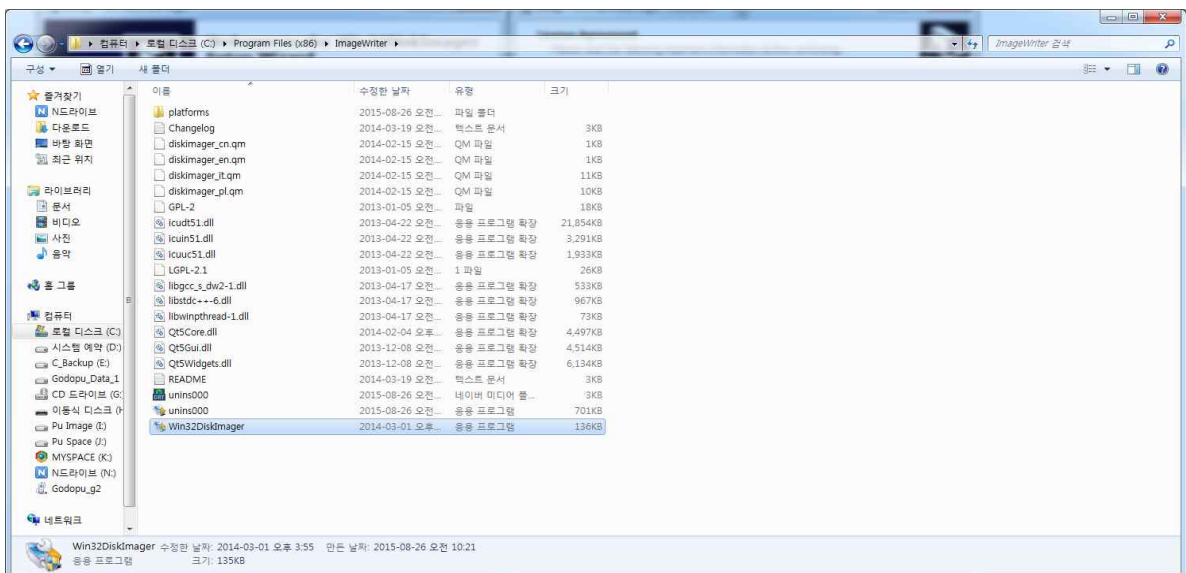


Figure 11. Win32DiskImager.exe 실행



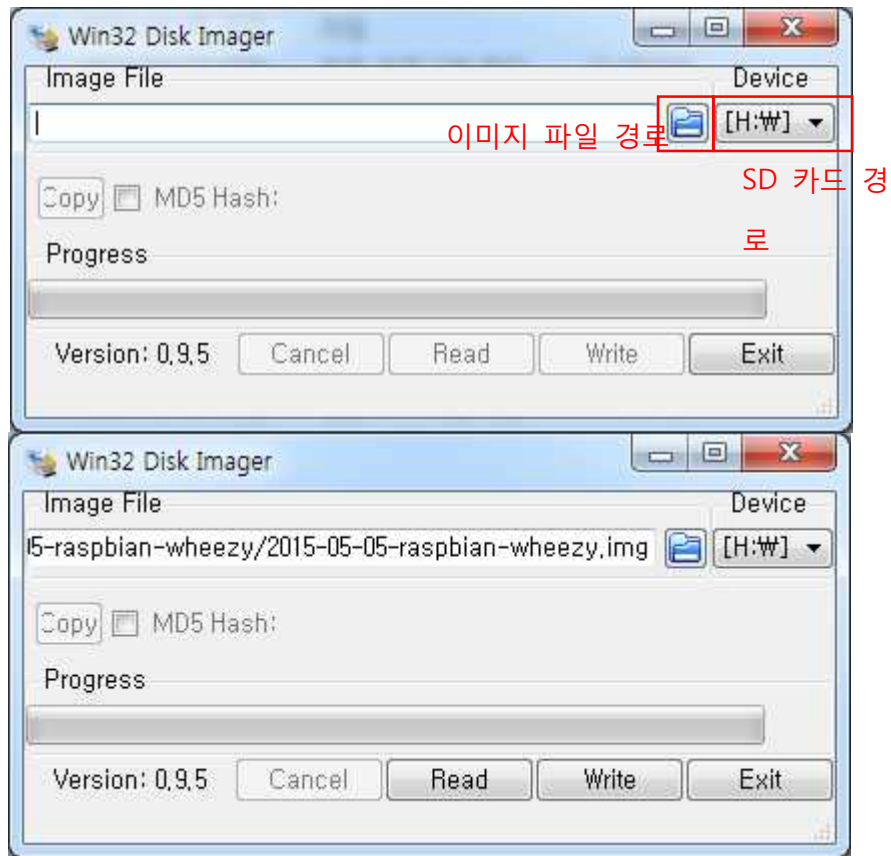


Figure 12. Raspbian 설치

이제 라즈베리 파이에 sd카드를 삽입한 후 5핀 충전기를 이용해 전원을 공급하면 전원이 켜지고 로그인 화면이 나온다. 기본 id 는 'pi' 이며, password 는 'raspberry' 이다. 로그인을 하게 되면 사용의 편의성을 위해 아래의 Figure14 ~ 19의 내용에 따라 기본 구성을 설정한다.



Figure 13. 라즈베리파이 전원 연결 및 로그인

처음에 기본적으로 생성되는 기본 구성 화면이 나오면 Internationalisation Options를 선택하여 설정을 수행한다.

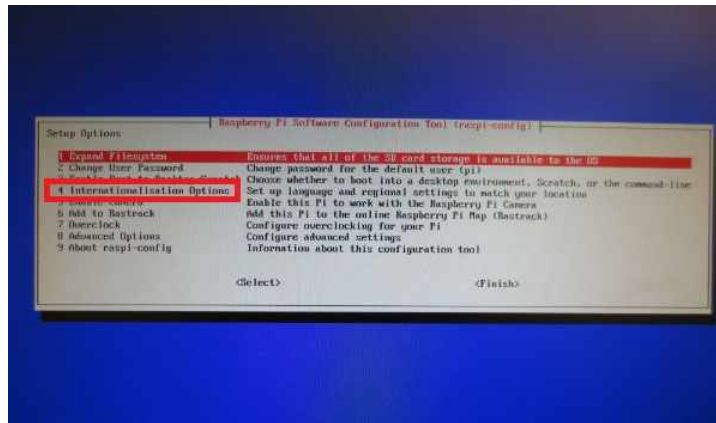


Figure 14. Internationalisation Options 선택

Internationalisation Option을 선택하면 메뉴 3개가 나오는데 위에서부터 차례대로 설정

한다. 우선 Change Locale을 선택한 후 설정을 알아보도록 하겠다.

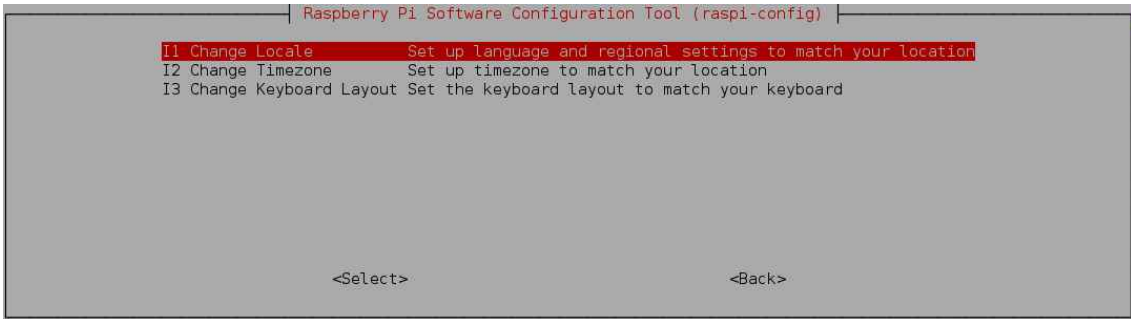


Figure 15. Change Locale 선택

'Change Locale'을 선택하면 다음과 같은 화면을 볼 수 있다. 여기서 'ko\_KR.UTF-8 UTF-8'을 찾아서 선택한다. (스페이스 바를 사용하면 선택이 가능하다)

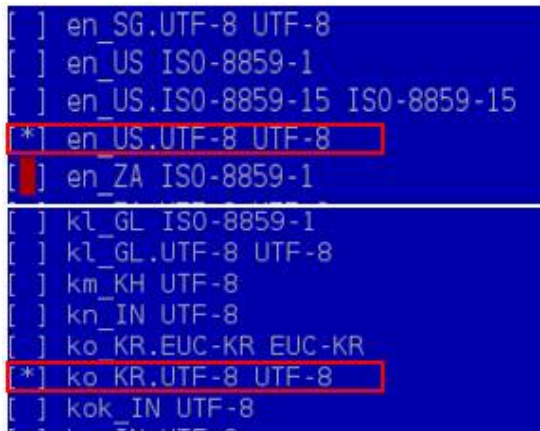


Figure 16. 'ko\_KR.UTF-8 UTF-8' 선택

그리고 Enter를 눌러주면 Figure 17과 같은 화면을 볼 수 있다. 여기서는 'en\_US.UTF-8'을 선택하고 계속해서 Enter 키를 눌러 초기화면으로 돌아가도록 한다.

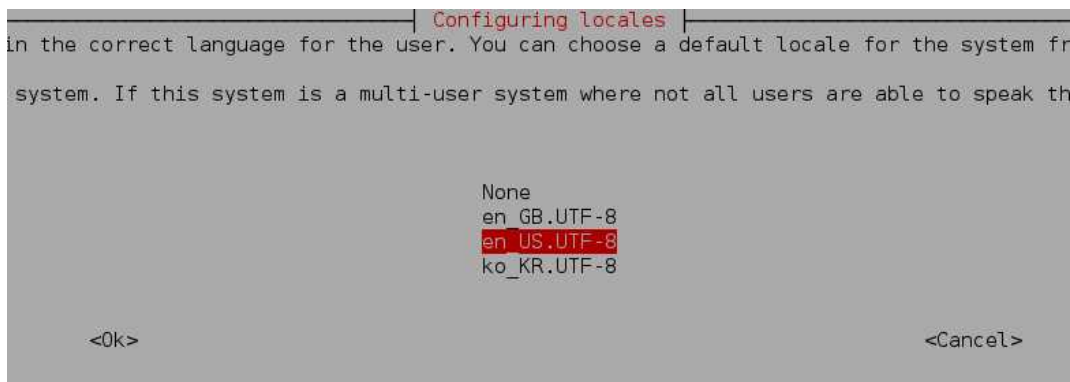


Figure 17. Configuring locales 화면

Figure 14번 화면으로 돌아오면 다시 4번 Internationalisation Options로 들어가서 Change Timezone을 선택한다. 여기서 Figure 18과 같이 locale을 Asia -> Seoul을 선택한다.

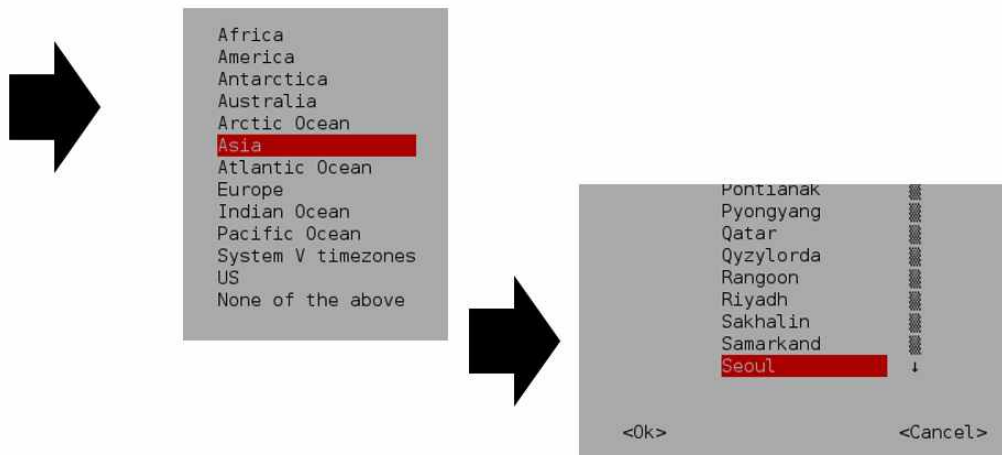
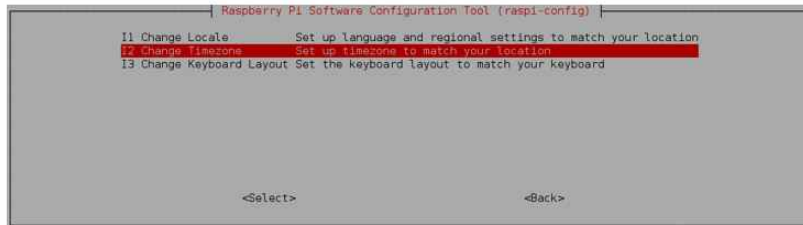


Figure 18. Change Timezone 설정과정

마지막으로 다시 4번 Internationalisation Options에서 Change Keyboard Layout을 선택 다음의 순서대로 셋팅한다. (Figure 19 참조)

1. Generic 105-key (Intl) PC
2. Other
3. Korean
4. Korean - Korean (101/104 Key compatible)
5. The default for the keyboard layout
6. No compose key
7. Yes

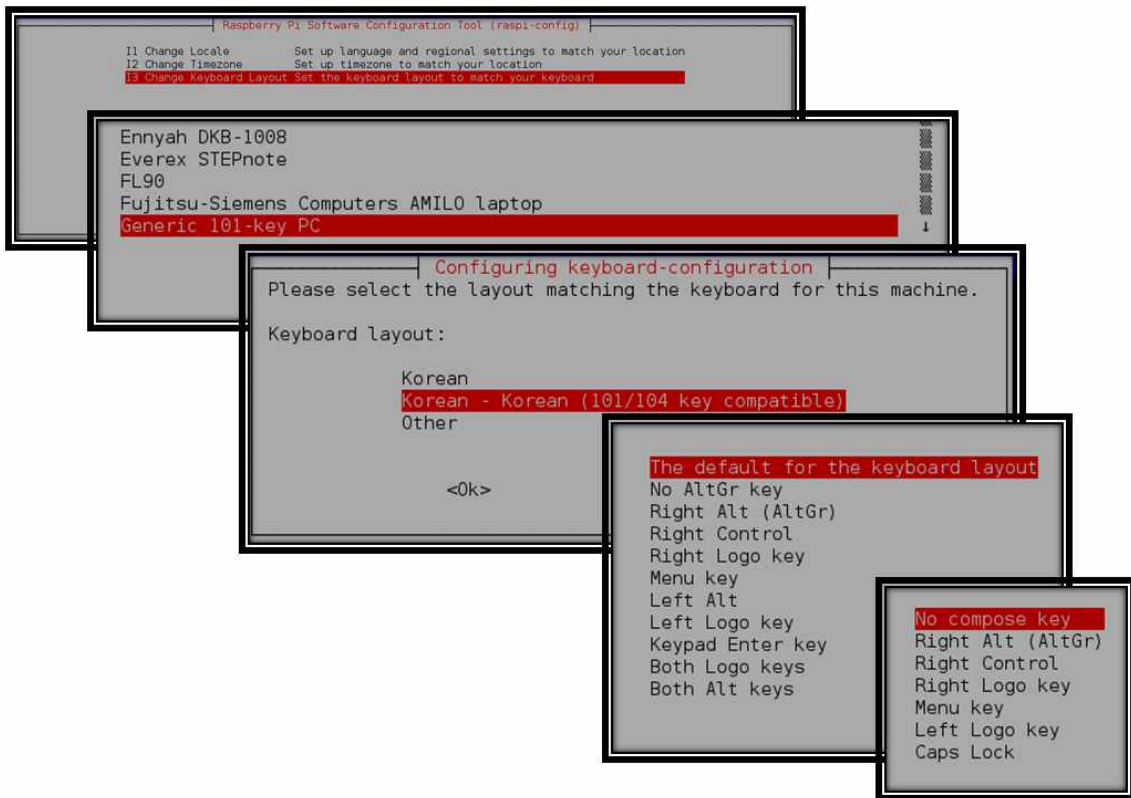


Figure 19. Change Keyboard Layout 설정과정

설정이 끝났다면 Finish 버튼을 누르고 shell 창으로 이동한다. 이때 혹시 '#' 혹은 '@' 키를 눌렀을 때 다른 문자가 출력된다면 “sudo cat /etc/default/keyboard” 명령어를 사용하여 XKBLAYOUT의 속성이 'gb' 가 아닌지 확인한다. 혹시 gb 라면 “sudo vim /etc/default/keyboard” 명령어를 사용하여 'kr' 또는 'en' 으로 수정한다.

**---변경법---**

변경은 vim 또는 vi에디터를 사용하여 변경한다. vim을 사용하실 경우 “sudo apt-get install vim” 명령어를 사용하여 vim을 설치하셔야 합니다. 변경하고 싶은 곳에 커서를 가져다 놓으신 후 'i' 를 눌러 변경 ESC -> ':' -> wq입력 절차를 통해 내용을 변경할 수 있다.

Shell 창에서 startx 를 입력하면 GUI모드로 들어갈 수 있으며, 운영체제를 설치한 직후에는 아래의 명령어를 사용하여 운영체제를 업그레이드 하는 것을 권장한다.

# startx	GUI 모드 실행
# sudo apt-get install upgrade	운영체제 업그레이드
# sudo apt-get install update	

Figure 20. Raspbian OS 업그레이드 및 GUI 모드 명령어

## 2.2 Java jdk 설치

이제 라즈베리 파이에서 jar 파일 및 java 파일을 실행시키기 위해 jdk를 설치한다. 설치하는 런타임(jre) 환경과 개발 환경(jdk)을 모두 설치 가능한 open jdk로 설치한다. 아래의 명령어를 입력하면 자동으로 설치가 완료된다.

```
# sudo apt-get install openjdk-7-jdk
```

Figure 21. openjdk-7-jdk 설치 명령어

java -version명령어를 이용하여 제대로 jdk가 설치되었는지 확인할 수 있다. 제대로 설치가 완료되었다면, Figure 22와 같은 화면을 볼 수 있다.

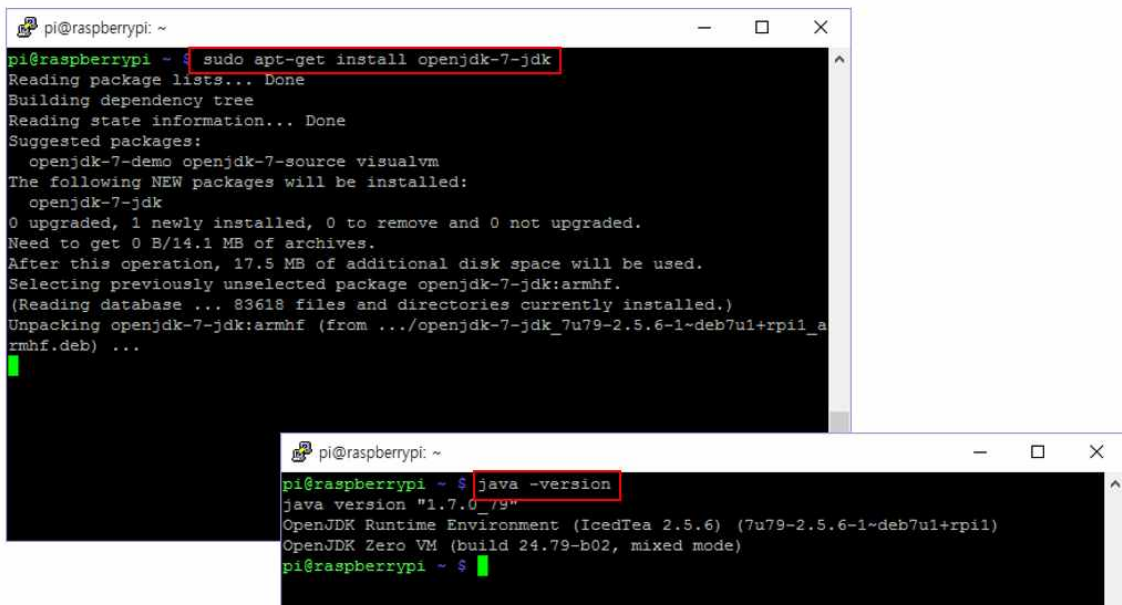


Figure 22. openjdk-7 설치 및 확인

## 2.3 &Cube 설치 및 실행

이제 &Cube를 설치해보도록 하겠다. &Cube는 사물인터넷 디바이스/게이트웨어에 탑재되는

S/W 플랫폼으로 개방형 사물인터넷 서버 플랫폼 Mobius와 연동할 수 있도록 지원하는 역할을 한다. &Cube는 이클립스 환경에서 자바로 개발 되었기 때문에 이클립스 홈페이지인 <http://www.eclipse.org/downloads> 로 접속하여 이클립스를 다운 받는다.

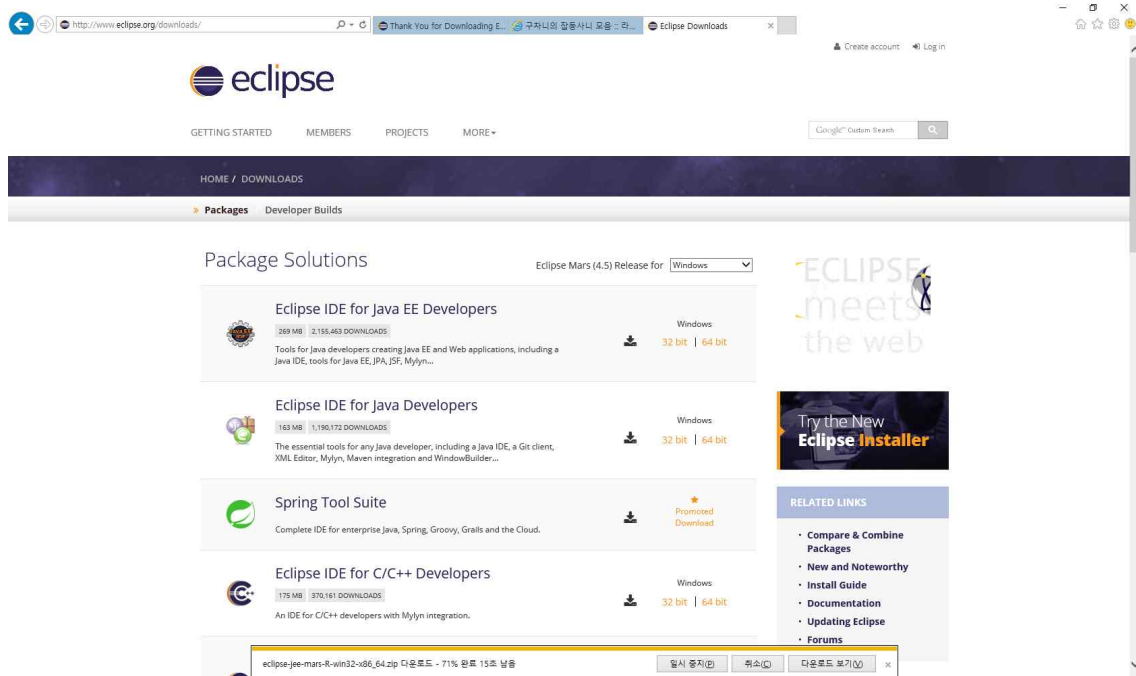


Figure 23. 이클립스 다운로드

라즈베리 파이에서 &Cube를 실행시키기 위해서는 .jar(java archive) 파일을 만들어야 한다. 일단 Ocean ([www.iotocean.org](http://www.iotocean.org))에 회원가입을 한 후 nCubeLavenderVer1.0.zip를 다운로드 받습니다.

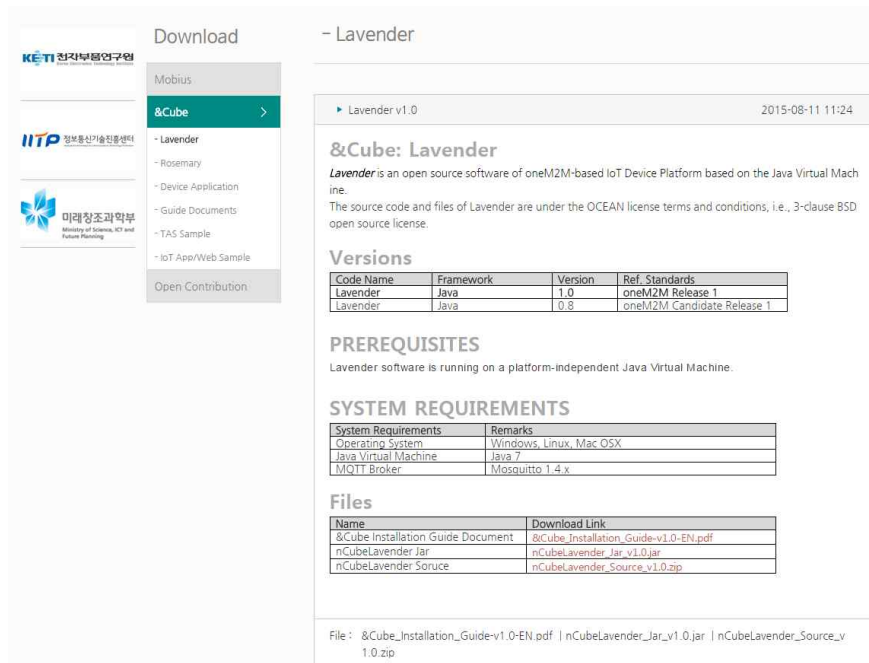


Figure 24. Lavender 다운 (Ocean 홈페이지)

그리고 eclipse에 import 시킨다. eclipse에 import 시키는 방법은 아래와 같다.

Procedure	<ul style="list-style-type: none"> <li>&gt; nCubeLavenderVer1.0.zip 압축풀기</li> <li>&gt; File -&gt; import -&gt; Existing Project into Workspace-&gt;Browse-&gt; nCubeLavenderVer1.0폴더 선택 &gt; Finish</li> </ul>
-----------	--

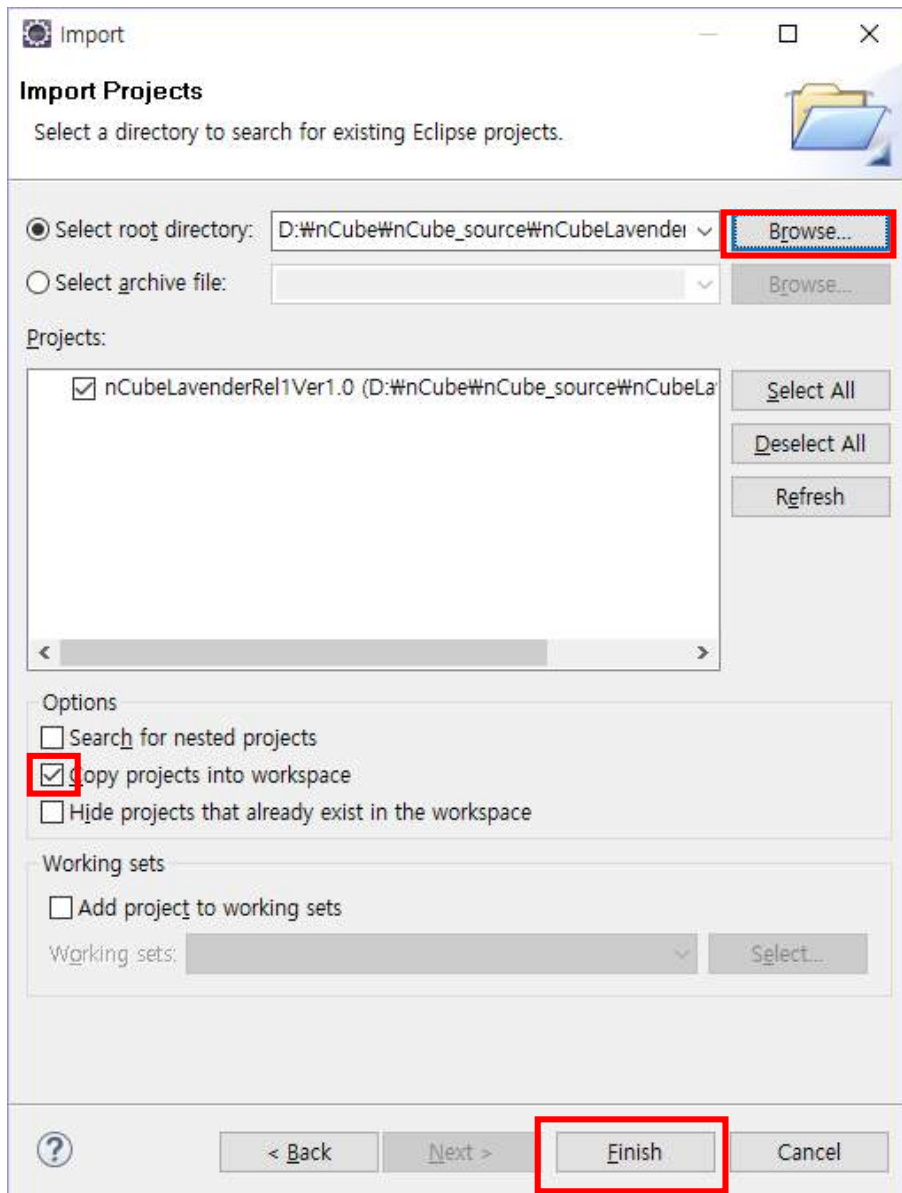


Figure 25. nCube Lavender v1.0 프로젝트 import

해당 프로젝트를 import 했을 때 에러가 생긴다면 다음과 같이 해결하도록 한다. 우선 프로젝트 위에 마우스 커서를 가져다 놓고 마우스 오른쪽 키를 눌러 Properties > java Build Path 로 들어갑니다. 그리고 import 되어있던 외부 jar 파일들을 모두제거 하고 다시 jar



파일들을 import 시켜줍니다. (Figure 26 ~ 30 참조)

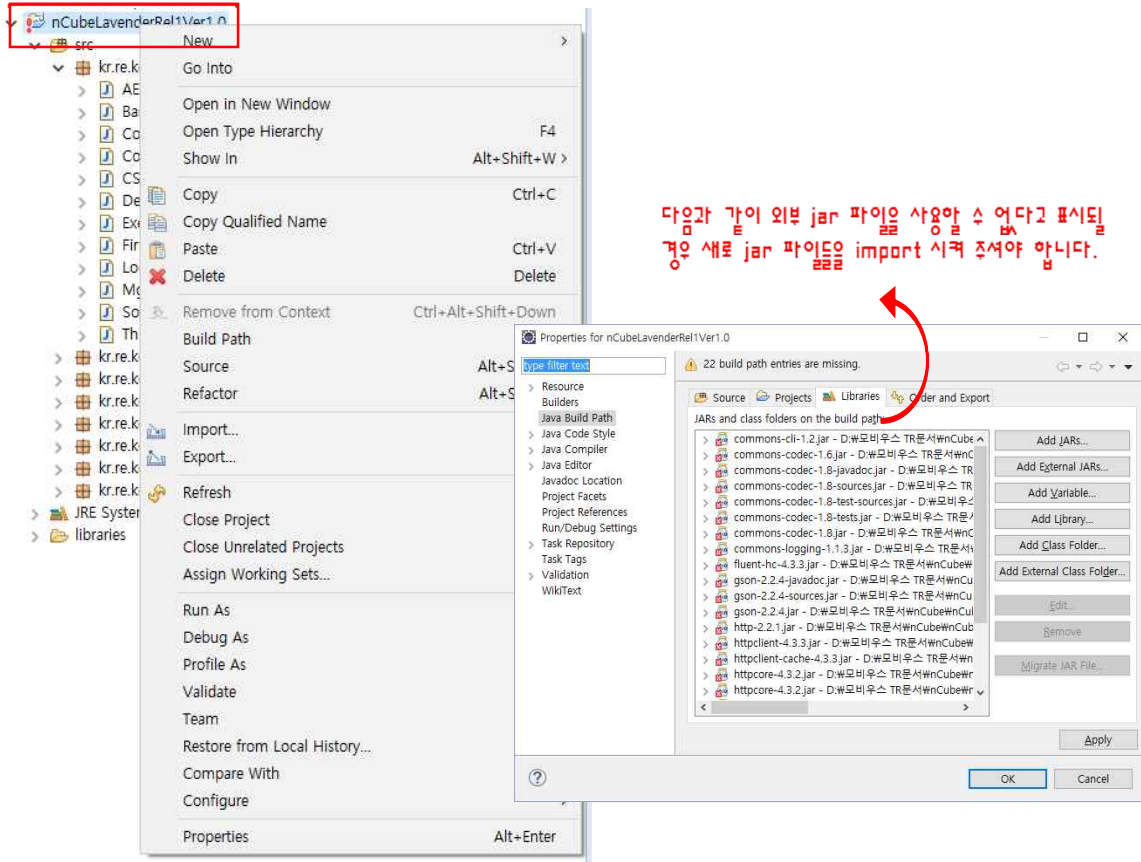


Figure 26. 외부 jar 파일 제거 1

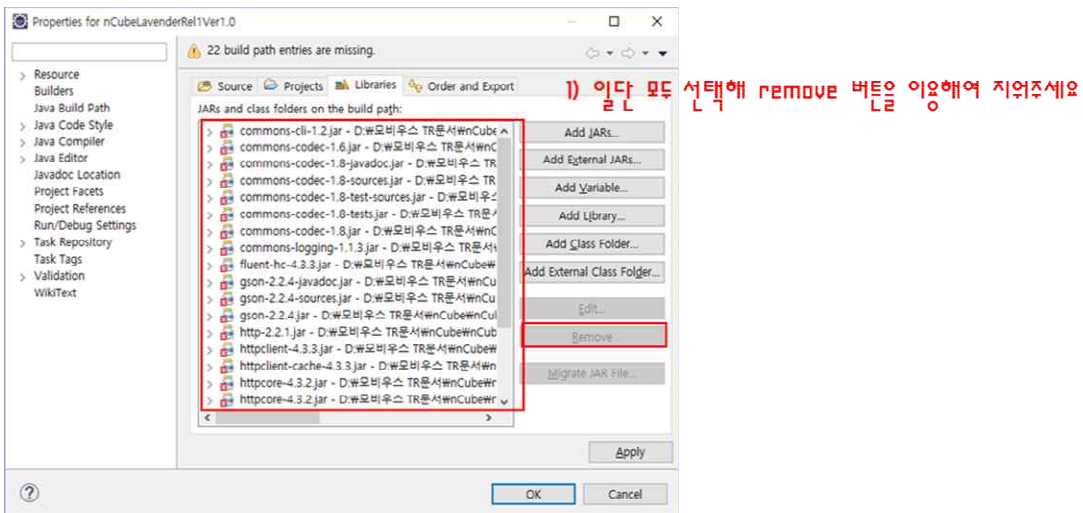


Figure 27. 외부 jar 파일 제거 2

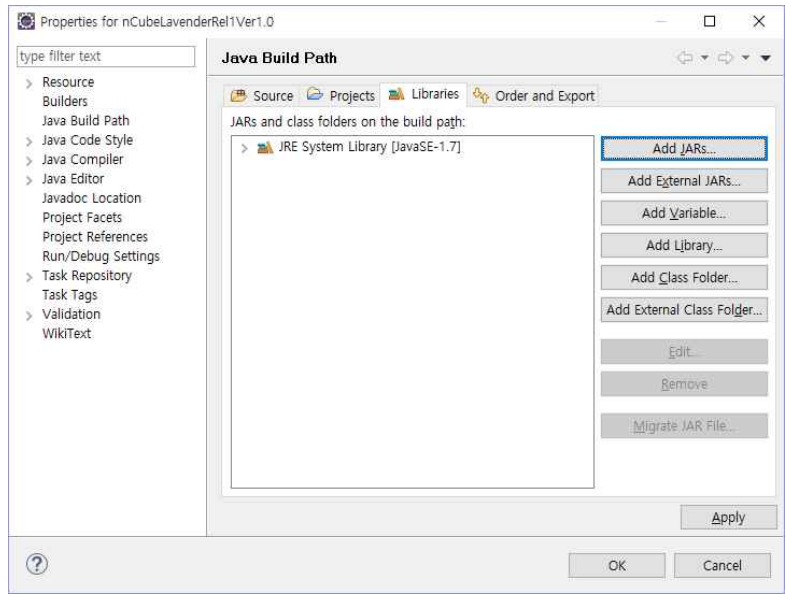


Figure 28. 외부 jar 파일 제거 3

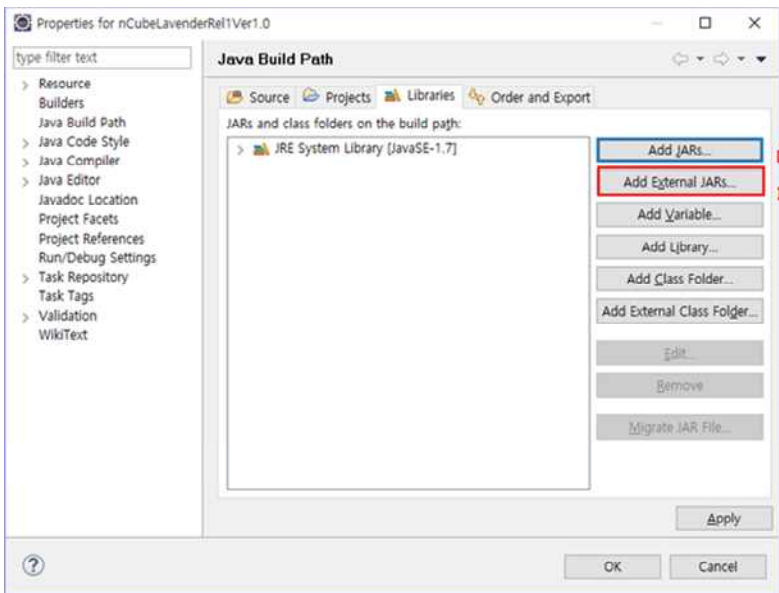


Figure 29. 외부 jar 파일 import 1

import 하실 때 libraries 폴더 내에 있는 httpcomponents-client-4.3.3\lib, httpcomponents-core-4.3.2\lib 폴더에 있는 jar 파일과 commons-codec-1.8, google-gson-2.2.4 폴더 내에 있는 jar 파일 모두를 import 시켜줘야 한다.

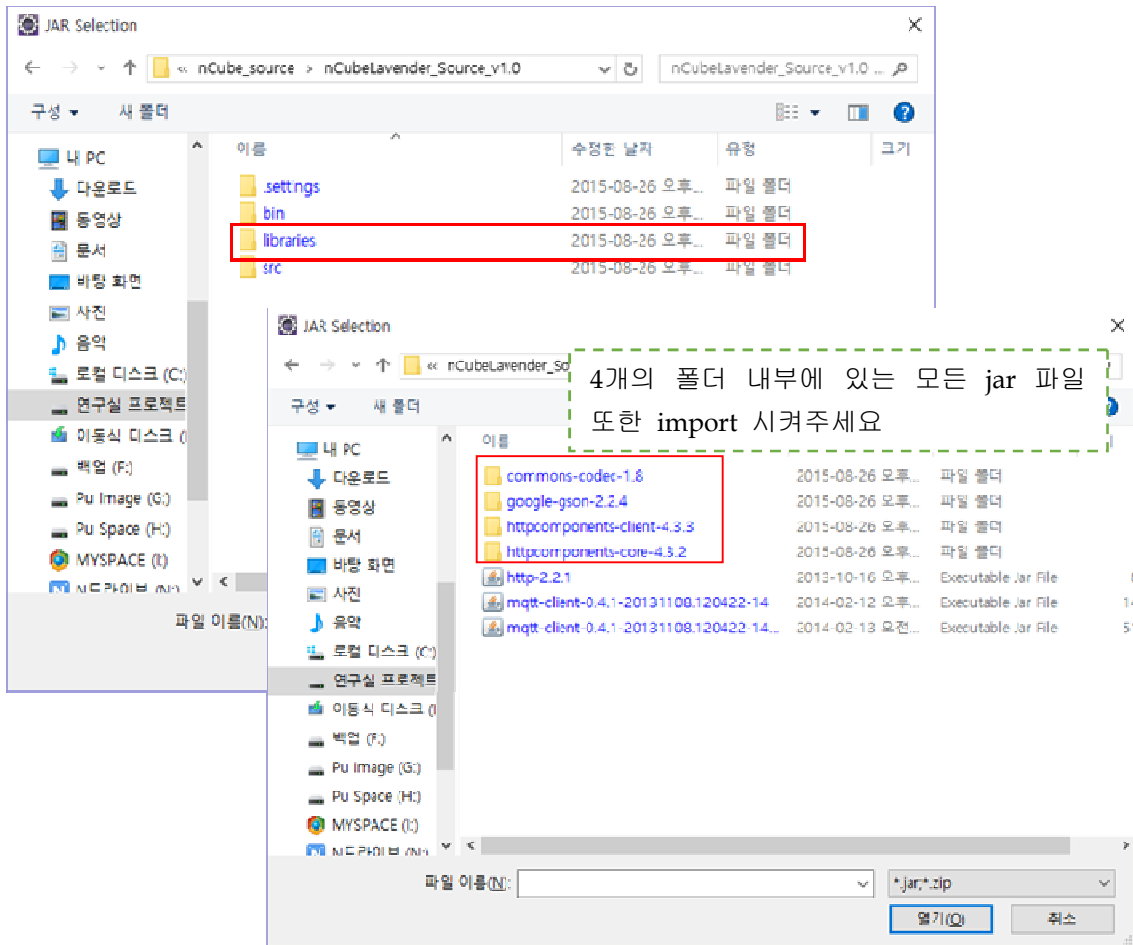


Figure 30. 외부 jar 파일 import 2

이렇게 외부 jar 파일들을 다시 import 시켜주시면 에러가 사라질 것이다. &Cube를 실행 시키기 위해서 소스코드를 수정해주셔야 한다. Figure 31과 같이 kr.re.ncube.devicemanager 패키지 내에 있는 DeviceManager.java 파일을 열어서 수정하도록 한다.

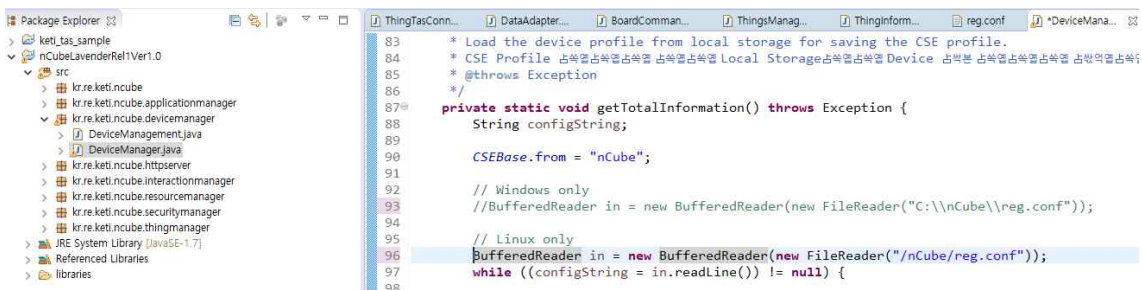


Figure 31. kr.re.ncube.devicemanager.DeviceManager.java

코드의 //Linux only 밑의 BufferedReader in = new BufferedReader(new FileReader("/nCube/reg.conf")); 에 있는 주석을 풀고, Windows only 밑의 문장을 주석처리

한다. 지금과 같이 windows 의 이클립스 환경에서 프로그램을 돌리실 때는 windows only 밑의 문장에 주석을 해제하고, Linux only 밑에 있는 문장을 주석처리를 하도록 한다. 본 문서에서는 라즈베리 파이의 raspbian 환경에서 이 프로그램을 돌릴 것이기 때문에 Windows only 밑의 문장을 주석처리 하였다.

이제 모든 수정이 완료되었다. 이제 라즈베리 파이에서 돌릴 수 있도록 Runnable Jar 파일을 만드는 과정을 알아보도록 한다. (Figure 32 ~ 33 참조)

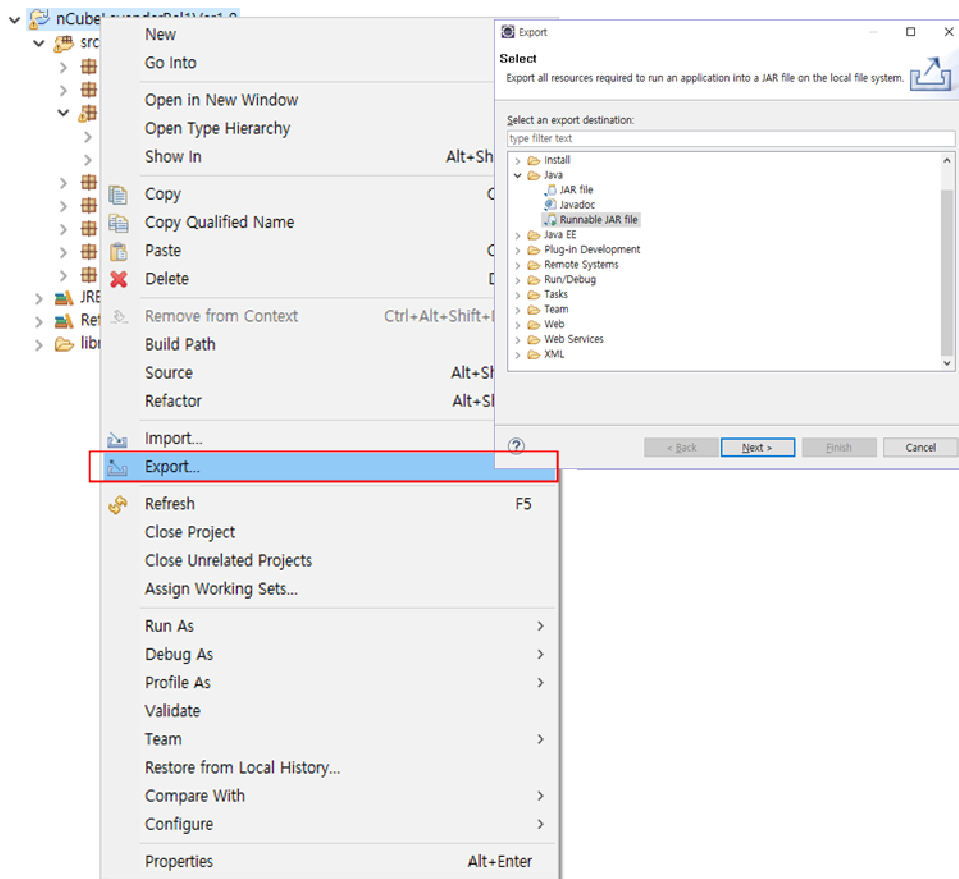


Figure 32. Export Project 1

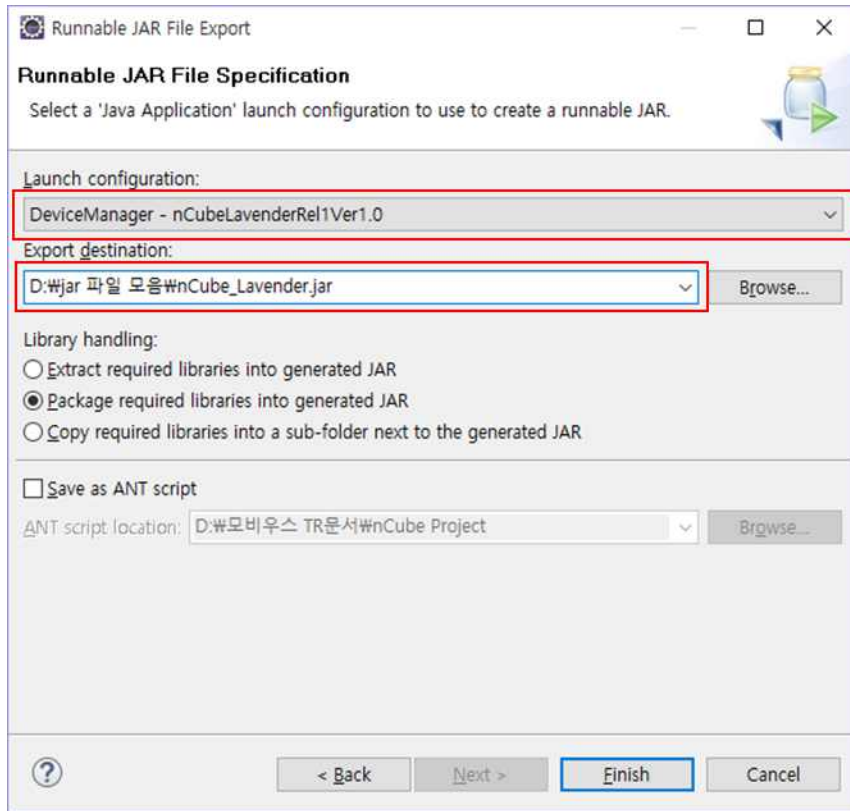


Figure 33. Export Project 2

nCube\_Lavender.jar 파일이 만들어 졌다면 라즈베리 파이로 이동한다. 이동시키기 전 루트 폴더 내에 nCube 폴더를 만든다. nCube 폴더는 Figure 34와 같이 만들 수 있다.

```
# mkdir /nCube
```

Figure 34. /nCube 디렉터리 생성 명령어

그리고 /nCube 폴더에 nCube\_Lavender.jar 파일을 위치 시킨 후 reg.conf 파일을 생성한다. Reg.config 파일의 경우 Figure 35의 형식에 맞게 작성해야 한다.

Procedure	<pre>&gt; # sudo vim reg.conf &gt; 아래의 내용을 따라 내용을 입력</pre> <hr/> <pre>CSEid=0.2.481.1.0001.001.7591 CSEpasscode=1234 CSEName=SNM CSEPointOfAccess=MQTT 0.2.481.1.0001.001.7591 requestReachability=true firmwareName=nCube_Lavender</pre>
-----------	---

	<pre> firmwareDescription=nCube1.0_Test_version firmwareVersion=1.0 firmwareURL=<a href="http://www.keti.pe.kr">http://www.keti.pe.kr</a> firmwareStatus=0 deviceName=KETI_SmartCity_Device deviceLabel=KETI_SmartCity_Device deviceDescription=KETI's SmartCity Device deviceManufacturer=KETI deviceModel=KETI001 deviceType=Sensor device deviceFwVersion=1.0 deviceSwVersion=1.0 deviceHwVersion=1.0 INCSEAddress=155.230.105.169:9000 MQTTBrokerAddress=155.230.105.169 interopType=0 primitiveType=0 protocol Binding=1 debugPrint=1 </pre> <p>위의 내용은 oneM2M 표준을 따른 것이며, INCSEAddress 및 MQTTBrokerAddress를 제외한 나머지는 밑의 Parameter 표를 참고하여 수정하기 위해서는 각 형식에 맞게 작성하길 바란다. INCSEAddress 와 MQTTBrokerAddress 파라미터는 각각 Mobius 서버 주소와 MQTTBroker 주소로 서버를 따로 가지고 있지 않다면 그대로 사용하길 바란다 (예제로 든 서버의 주소는 연구실 서버 주소). 또한 CSEid 와 CSEPointOfAccess 의 MQTT 뒤에 오는 id는 CSEid이다.</p>								
Parameter	<table border="1"> <thead> <tr> <th>Name</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>CSEid</td> <td>CSE id for oneM2M standard (e.g., 0.2.481.1.0001.001.7591)</td> </tr> <tr> <td>CSEpassword</td> <td>ECSE password for password (e.g., 1234)</td> </tr> <tr> <td>CSEName</td> <td>CSE access point for oneM2M standard -HTTP : HTTP  [Address] -MQTT : MQTT   [CSEid] (e.g., HTTP  <a href="http://155.230.105.165">http://155.230.105.165</a> MQTT  0.2.481.1.0001.001.7591)</td> </tr> </tbody> </table>	Name	Description	CSEid	CSE id for oneM2M standard (e.g., 0.2.481.1.0001.001.7591)	CSEpassword	ECSE password for password (e.g., 1234)	CSEName	CSE access point for oneM2M standard -HTTP : HTTP  [Address] -MQTT : MQTT   [CSEid] (e.g., HTTP   <a href="http://155.230.105.165">http://155.230.105.165</a> MQTT  0.2.481.1.0001.001.7591)
Name	Description								
CSEid	CSE id for oneM2M standard (e.g., 0.2.481.1.0001.001.7591)								
CSEpassword	ECSE password for password (e.g., 1234)								
CSEName	CSE access point for oneM2M standard -HTTP : HTTP  [Address] -MQTT : MQTT   [CSEid] (e.g., HTTP   <a href="http://155.230.105.165">http://155.230.105.165</a> MQTT  0.2.481.1.0001.001.7591)								

CSEPointOfAccess	CSE requestReachability for oneM2M standard - Unreachable : 0 - Reachable : 1
requestReachability	CSE requestReachability for oneM2M standard - Unreachable : 0 - Reachable : 1
firmwareName	Firmware name for oneM2M standard (e.g., ComplexSensorDeviceFirmware)
firmwareDescription	Firmware description for Mobius (e.g., This firmware is made for complex sensor)
firmwareVersion	Firmware version for oneM2M standard (e.g., 1.0)
firmwareURL	Firmware URL for oneM2M standard (e.g., http://www.keti.re.kr)
firmwareStatus	Firmware status for oneM2M standard - Stop : 0 - Running : 1
deviceName	Device name for Mobius (e.g., ComplexSensorDevice)
deviceLabel	Device label for Mobius (e.g., complexesensor, KETI)
deviceDescription	Device label for Mobius (e.g., This device is complex sensor.)
deviceManufacturer	Device manufacturer for Mobius (e.g., KETI)
deviceModel	Device model for Mobius (e.g., KETI001)
deviceType	Device type for Mobius (e.g., Sensor device)
deviceFwVersion	Device firmware version for Mobius (e.g., 1.0)
deviceSwVersion	Device software version for Mobius (e.g., 1.0)
deviceHwVersion	Device hardware version for Mobius (e.g., 1.0)
INCSEAddress	IN-CSE address for oneM2M standard (e.g., 155.230.105.165:9000)
MQTTBrokerAddressdress	MQTT Broker address for oneM2M standard (e.g., 155.230.105.165)
interopType	oneM2M standard or Mobius interworking - Mobius : 0

		- oneM2M : 1
primitiveType	Long or short type content	-Short type content : 0 -Long type content : 1
protocolBinding	HTTP or MQTT protocol binding	- MQTT : 0 - HTTP : 1
debugPrint	print the debug message	- Not present : 0 - Present : 1
['Parameters' 설명]		

Figure 35. reg.conf 파일 생성

각 파라미터에 대한 정의는 Figure 35에 정의되어 있다. &Cube는 oneM2M 표준을 기반으로 구현되었기 때문에 각 파라미터에 대한 내용은 oneM2M 표준에 맞게 정의되어야 한다. 따라서 원활한 &Cube의 실행을 위해 위 Figure 35의 형식을 꼭 지켜주길 바란다.

```

pi@raspberrypi: ~
pi@raspberrypi ~ $ ls -l /nCube
total 4056
-rw-r--r-- 1 pi pi 4143334 Aug 26 14:08 nCubeLavender_Jar_v1.0.jar
-rw-r--r-- 1 root root 639 Aug 31 14:08 reg.conf
pi@raspberrypi ~ $ cat /nCube/reg.conf
CSEid=0.2.481.1.0001.001.7591
CSEpasscode=1234
CSEName=SNM
CSEPointOfAccess=MQTT|0.2.481.1.0001.001.7591
requestReachability=true
firmwareName=nCube_Lavender
firmwareDescription=nCube1.0_Test_version
firmwareVersion=1.0
firmwareURL=http://www.keti.re.kr
firmwareStatus=1
deviceName=KETI_SmartCity_Device
deviceLabel=KETI_SmartCity_Device
deviceDescription=KETI's SmartCity Device
deviceManufacturer=KETI
deviceModel=KETI001
deviceType=Sensor device
deviceFwVersion=1.0
deviceSwVersion=1.0
deviceHwVersion=1.0
INCSEAddress=155.230.105.165:9000
MQTTBrokerAddress=155.230.105.165
interopType=0
primitiveType=0
protocol|Binding=1
debugPrint=1
pi@raspberrypi ~ $

```

Figure 36. cat reg.conf 결과화면



reg.conf 파일을 만든 후에 nCube\_Lavender.jar 파일을 실행시키고, Device Registration 완료를 확인한다.

Procedure	<pre>&gt; #java -jar nCube_Lavender.jar &gt; 결과화면 확인 (e.g.) ..... [DeviceManager] Received labels : appInstall [DeviceManager] Received body : MCMDXXXXXXXXXX</pre>
-----------	---

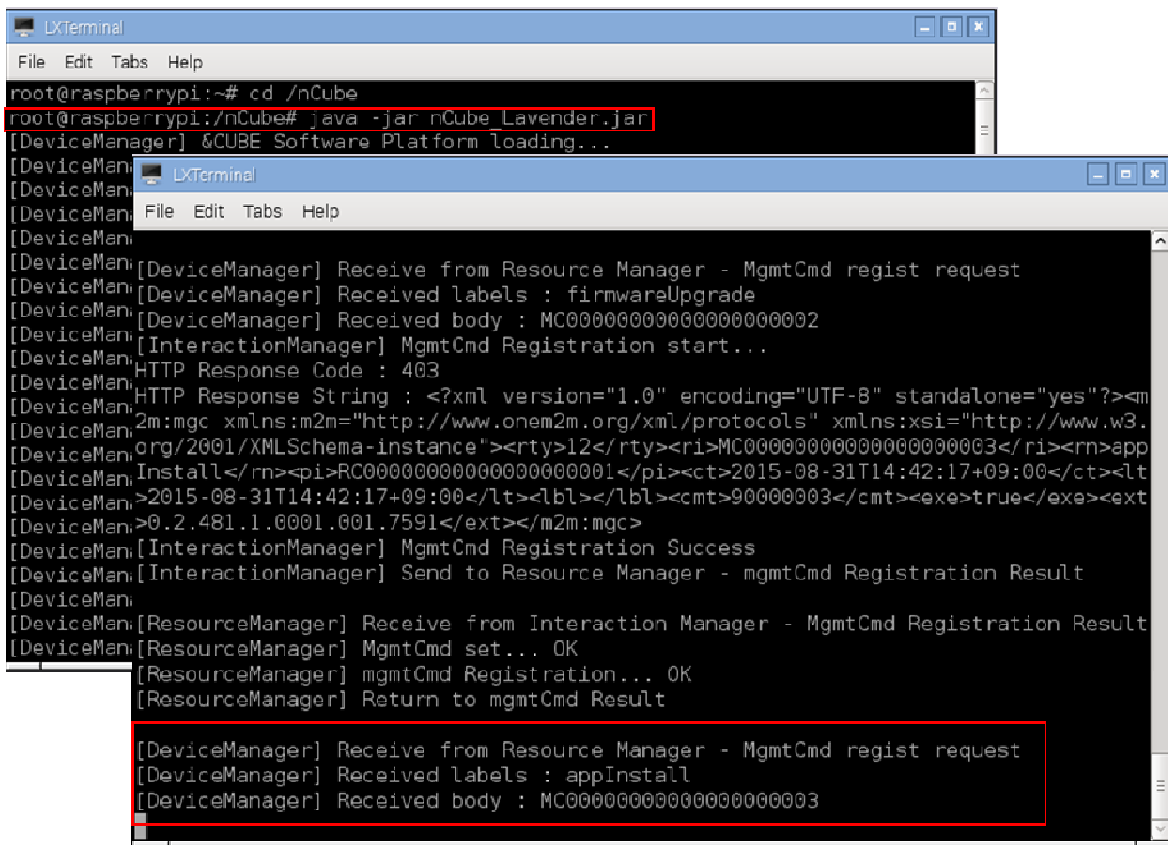


Figure 37. nCube\_Lavender.jar 실행 및 결과화면

#### 4. 결론

지금까지 사물인터넷 국제표준인 oneM2M 기반의 디바이스 플랫폼인 &Cube에 대하여 알아보았다. &Cube는 IoT 서버 플랫폼인 Mobius와 연동하여 데이터를 업로드하고 제어 메시지를 수신하여 제어하는 등의 서비스를 위한 기능을 제공한다. 본 문서에서는 사물인터넷

Alliance인 Ocean에서 공개한 &Cube를 설치하고 구동하는 내용을 다뤘다. Ocean 홈페이지에서 &Cube Lavender v1.0 및 설치 가이드 문서를 제공하고 있지만, 초보자도 쉽게 &Cube를 설치하여 구동할 수 있도록 본 문서를 작성하였다. 또한, 직접 설치 가이드 문서를 따라서 진행할 때 생기는 문제들의 해결방안을 언급하였다.

이와 같이, 사물인터넷 서버 플랫폼인 Mobius와 디바이스 플랫폼인 &Cube는 국제표준인 oneM2M 기반으로 구현되어 있기 때문에 상호운용성을 제공하며, 앞으로 중소기업 및 개인들도 쉽게 사물인터넷 서비스를 직접 구현하고 제공하는데 많은 도움이 될 것으로 생각된다.

## 참고 문헌

- [1] 사물인터넷 Alliance Ocean 홈페이지, <http://www.iotocean.org/main/>
- [2] &Cube 설치 가이드 문서, &Cube Install Guide-1.0.doc