**物聯網核心技術**

**實驗作業四**

**Linkit One and oneM2M**

**中正大學資工系黃仁竑教授**

**Fall, 2017**

**實驗目的**

本實驗主要在教導學生如何使用Linkit One開發板、oneM2M及智慧手機透過藍芽的方式進行資料傳輸。

**實驗所需軟體**

* PC
  + Windows7/8/10
  + Vmware
  + Ubuntu 14..04 LTS 以上穩定版本
  + Java SDK1.7以上
  + Network Enable Environment
  + Maven 3以上版本
  + Node.js
  + Node-red
  + Postman
  + 智慧型手機(Andriod 6.0)

目錄

[Part1 oneM2M環境設置 3](#_Toc479254330)

[1. oneM2M 介紹 3](#_Toc479254331)

[2. oneM2M 環境建置 4](#_Toc479254332)

3. oneM2M藍芽環境建置…………..……………….…………………………………….…….21

4. 使用Java新增oneM2M Application………………………………………………………23

5. 使用Node-red完成IN-AE subscription to MN-AE…………………….……………26

6. 使用App Inventor完成App讀取oneM2M資料…………………….……………26

Part1 oneM2M環境設置

1. oneM2M介紹

oneM2M是ETSI（European Telecommunications Standards Institute，歐洲電信標準協會）旗下的專門對M2M（機器到機器通信）以及IoT（物聯網）的一個國際標準組織。其目的在於研發一種物聯網架構及相關技術標準，以實現跨部門或者跨行業不同機器之間的訊息傳遞。目前的應用大部分都用於: 車載通訊系統、智慧交通系統、智慧醫療、智慧電網、工業物聯網、智慧家居等。

 oneM2M採用簡單的橫向平台架構，分為三層，包括了應用層、通用服務層、網路服務層，如右圖。

應用層(AE)會使用在端點設備例如智能手機及Sensor應用中。通用服務層(CSE)可視為在oneM2M的服務功能實例，透過不同的參考點與不同的應用服務連結做資訊交換並存取網路服務層。網路服務層則是提供通用服務層實體所需的服務，例如:

裝置管理、定位、裝置啟動等等。oneM2m透過以上標準使得可以被便捷的遷入各種硬體或者軟體中的通用的機器到機器的通訊服務，以實現將無數終端設備連結到全球佈署的應用伺服器中。

目前on2M2M open source implementations有以下四個:

1. Eclipse OM2M

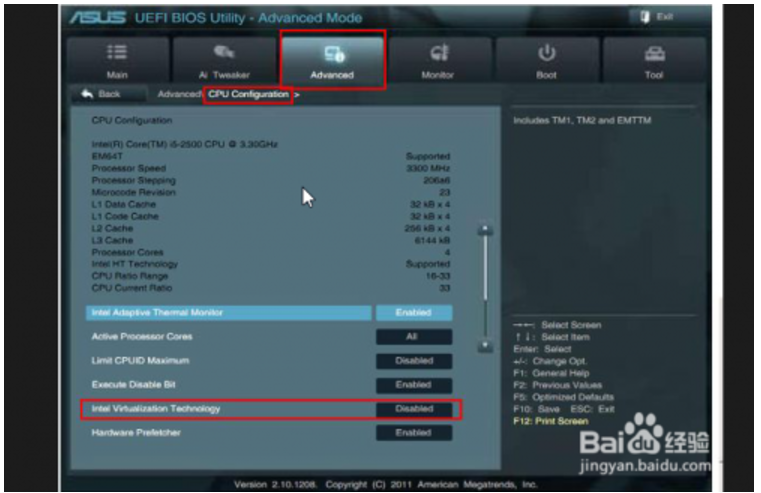
2. OCEAN Mobius

3. IoTDM

4. OASIS SI

本次Lab將會選用1 Eclipse OM2M作為使用。

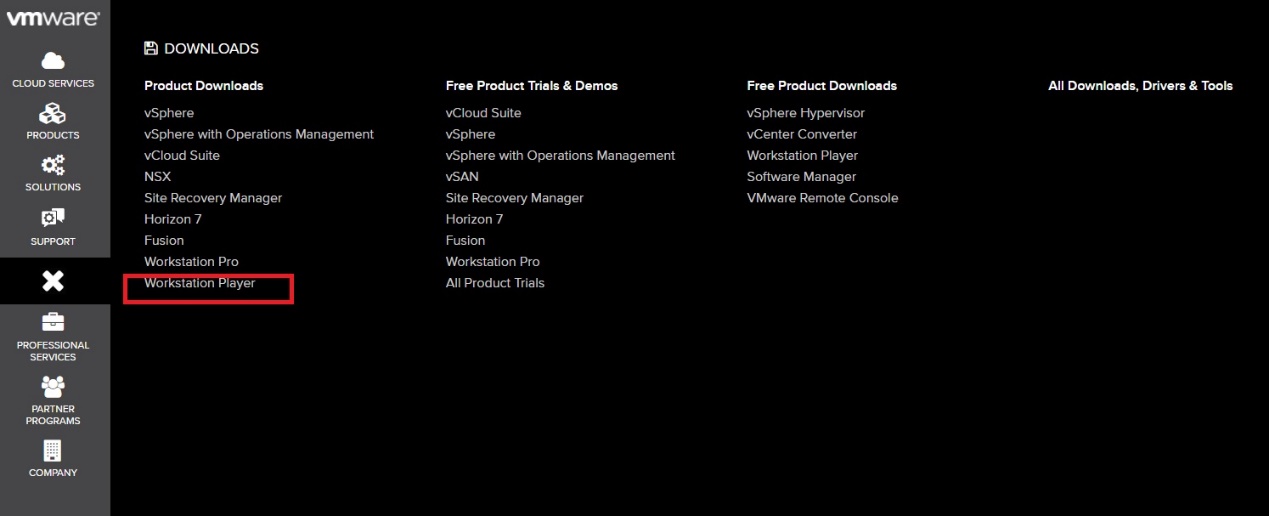
1. oneM2M 環境建置
2. oneM2M 環境建置
3. 請先重新啟動電腦按F2或者Del進入到Bios系統後，到CPU Options -> Advanced Setup，看到Intel Virtualization Technology將選項改成Enable儲存並重新開機即可。

以下為網路擷取範例示意圖，依每個人環境不同而有差異。

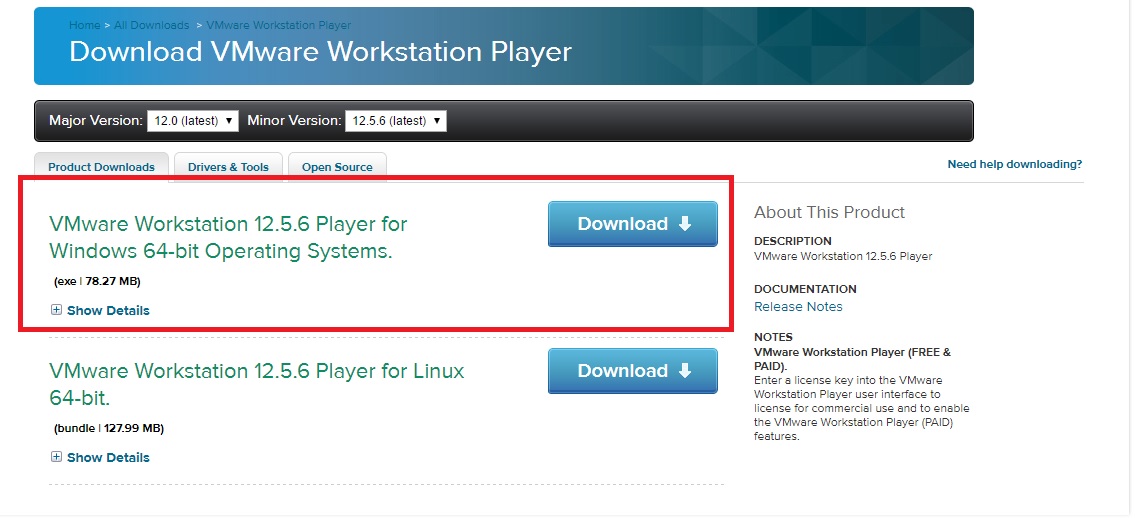
1. 請至<https://www.ubuntu-tw.org/modules/tinyd0/>下載ubuntu14.01LTS以上的版本並選擇好位元版本後開始下載。
2. 前往[VMware官方網站](https://www.vmware.com/tw.html)點選左方下載🡪Workstation Player🡪選取Windows版本Download

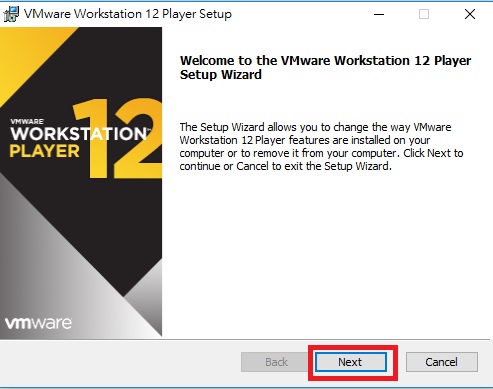
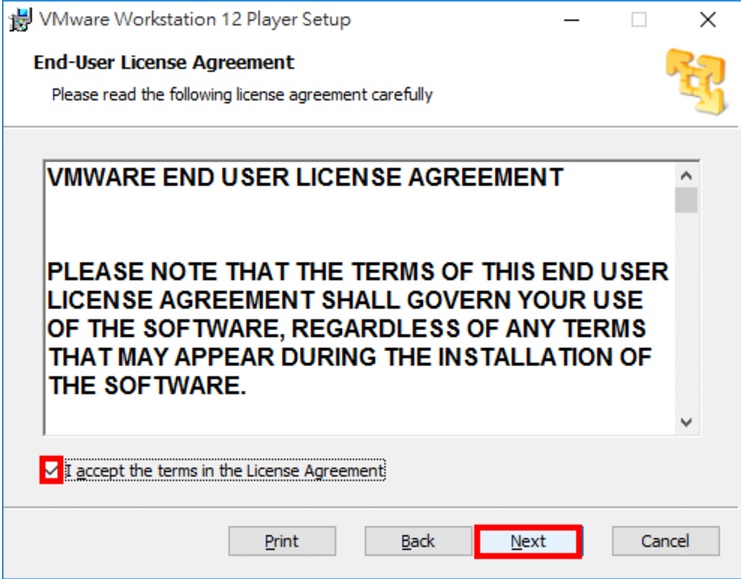
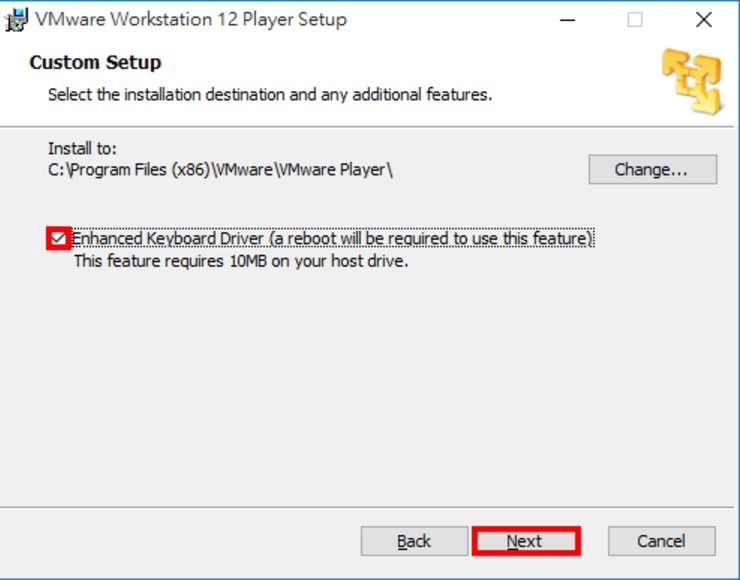
左方下載

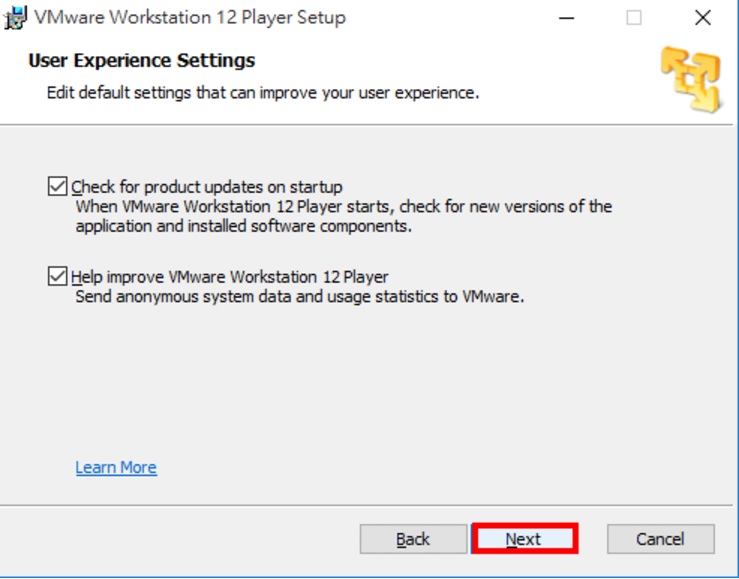
Workstation Player

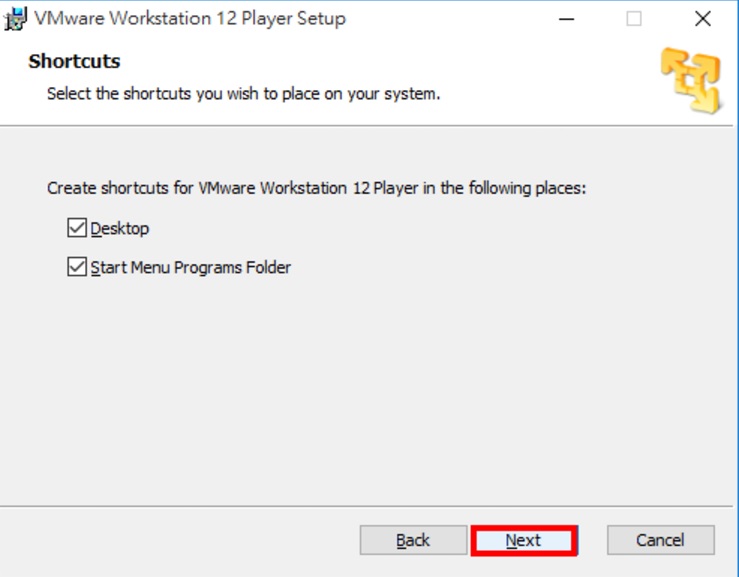


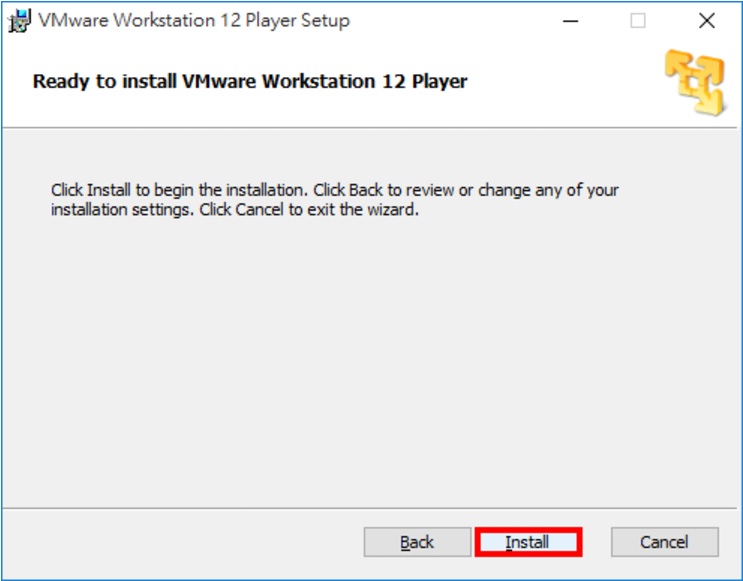
Windows版本Download

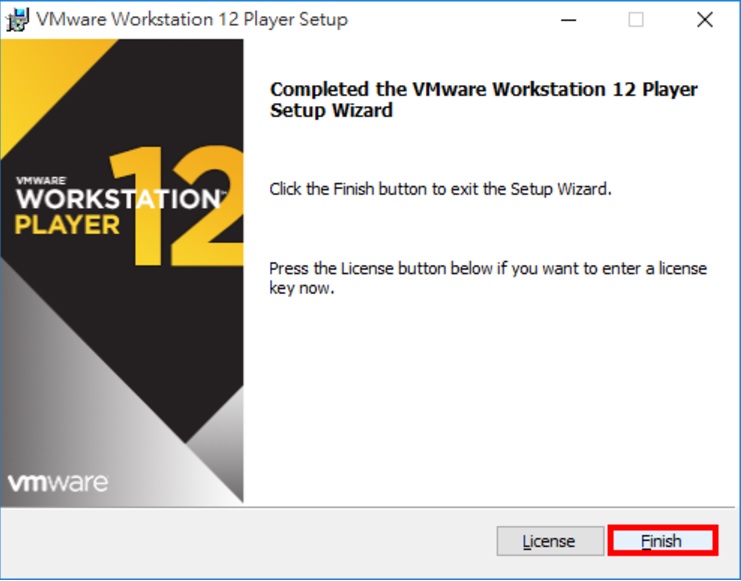
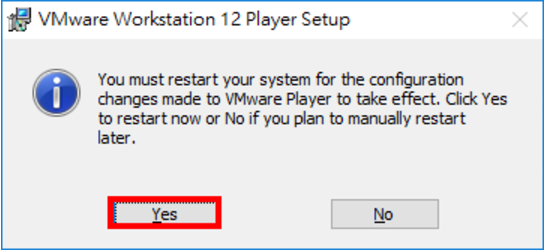
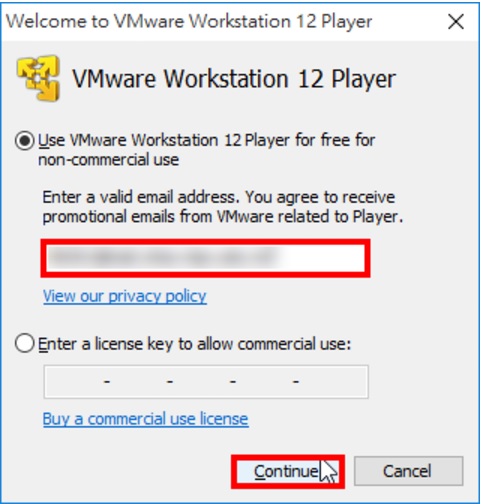


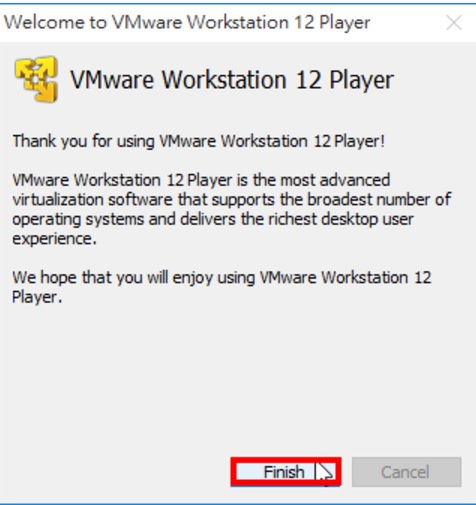
1. 開啟剛剛下載的VMware安裝檔案開始安裝點選next
2. 勾選”I accept the terms in the License Agreement”後按下Next。
3. 勾選”Enhanced Keyboard Driver”後按下Next。
4. 使用預設選項後按下Next。

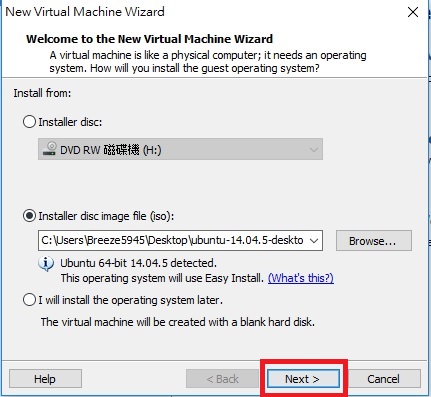
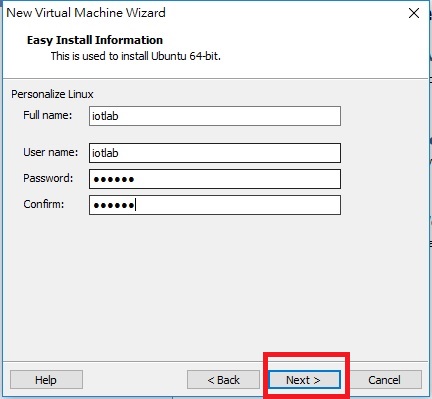
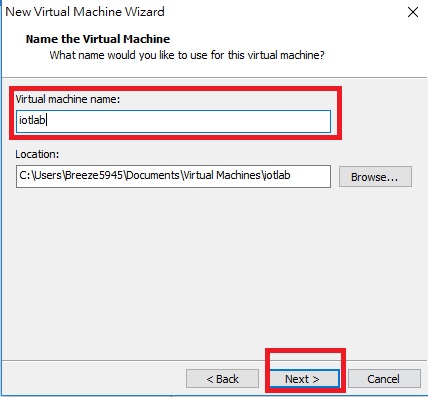
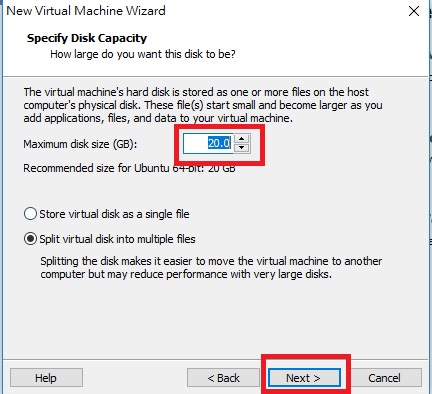
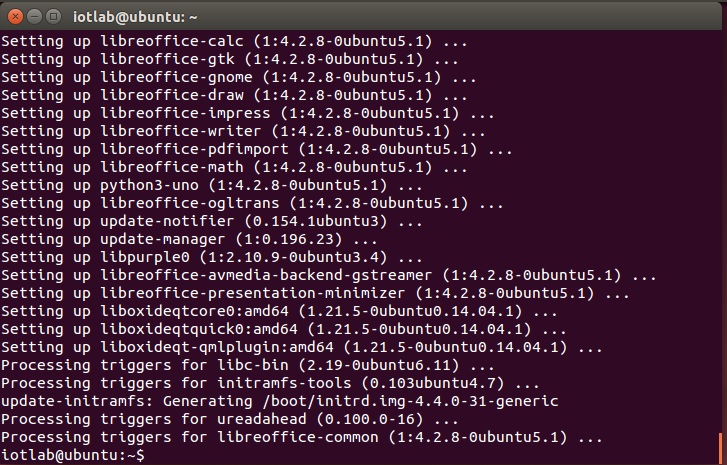
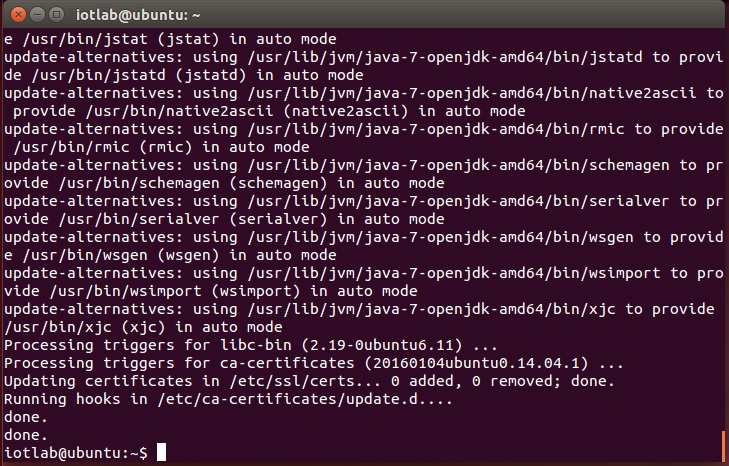
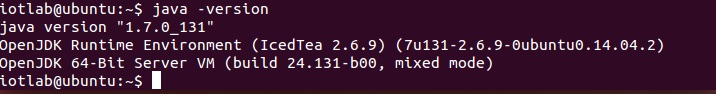
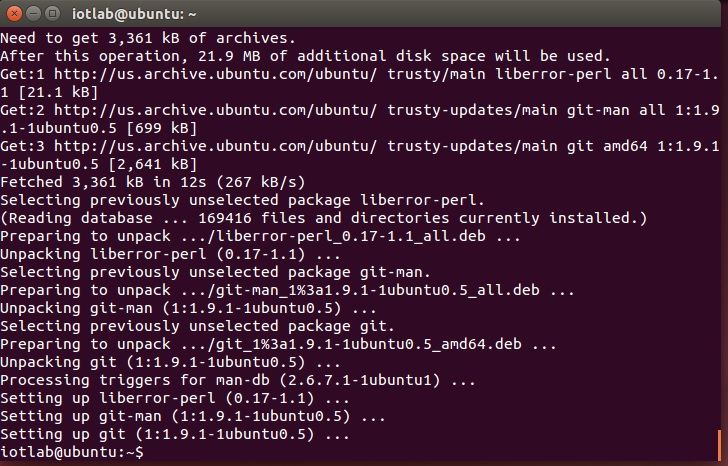
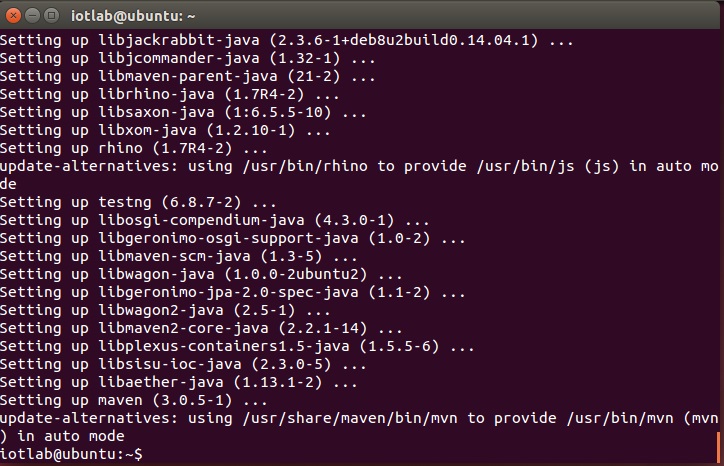
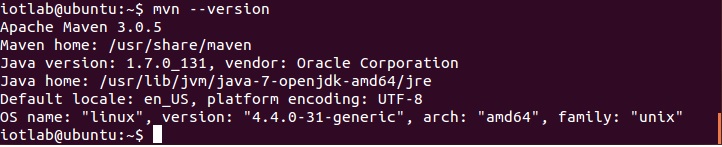
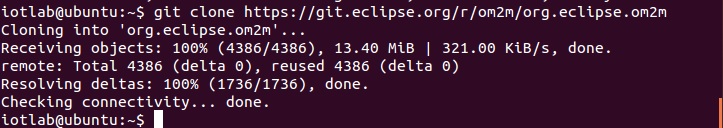
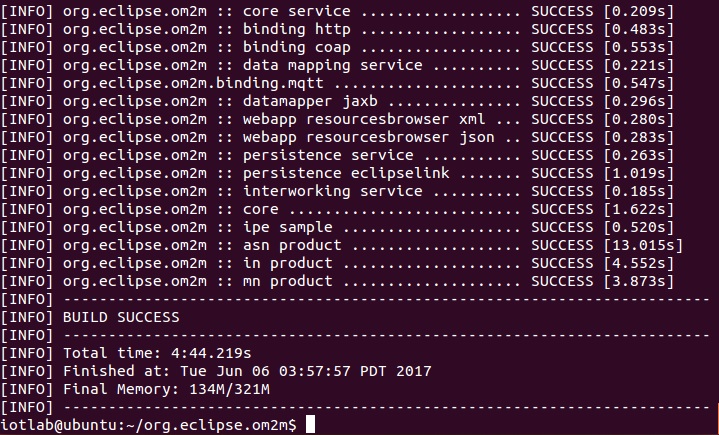
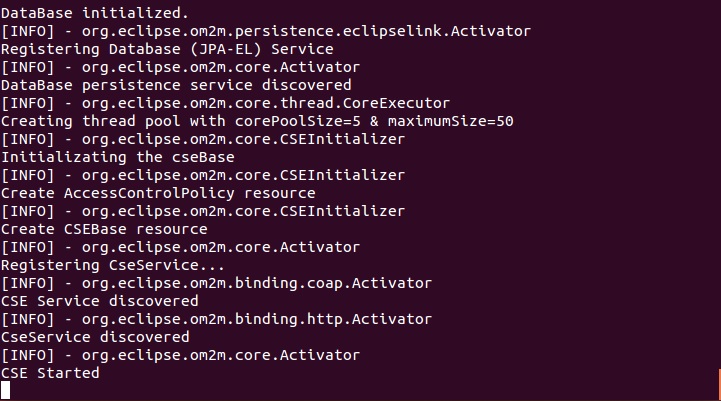
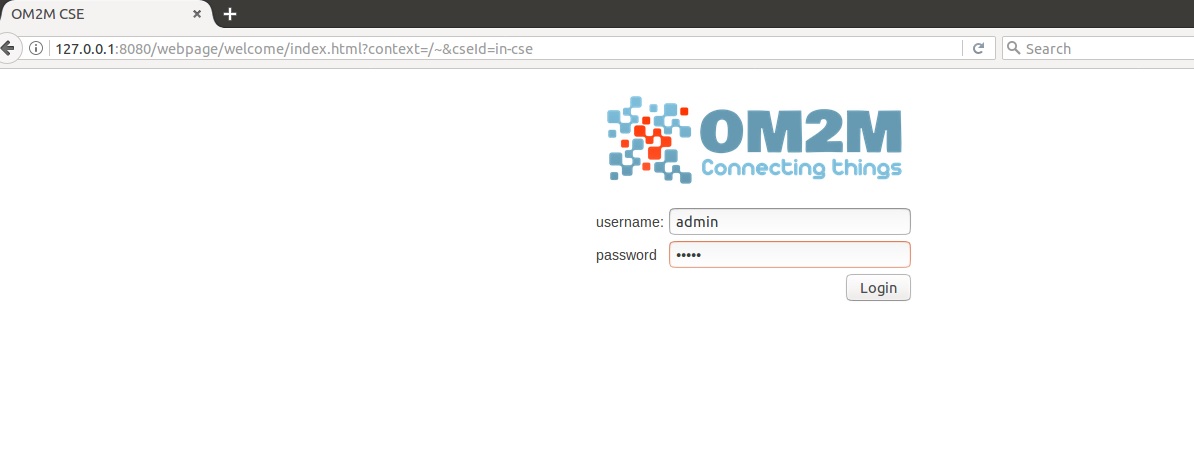
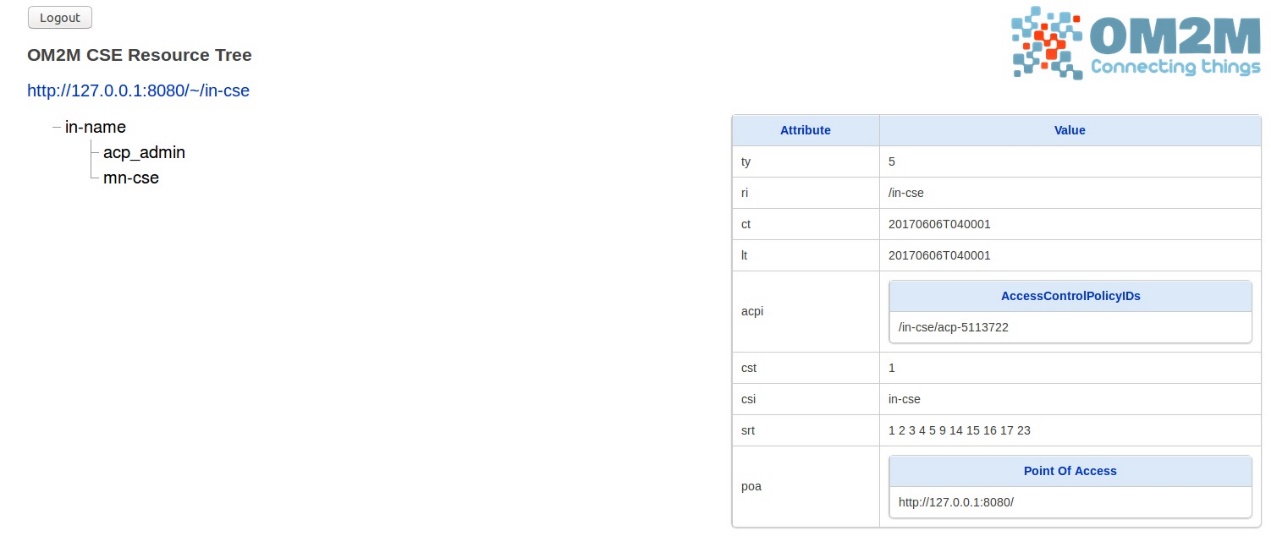
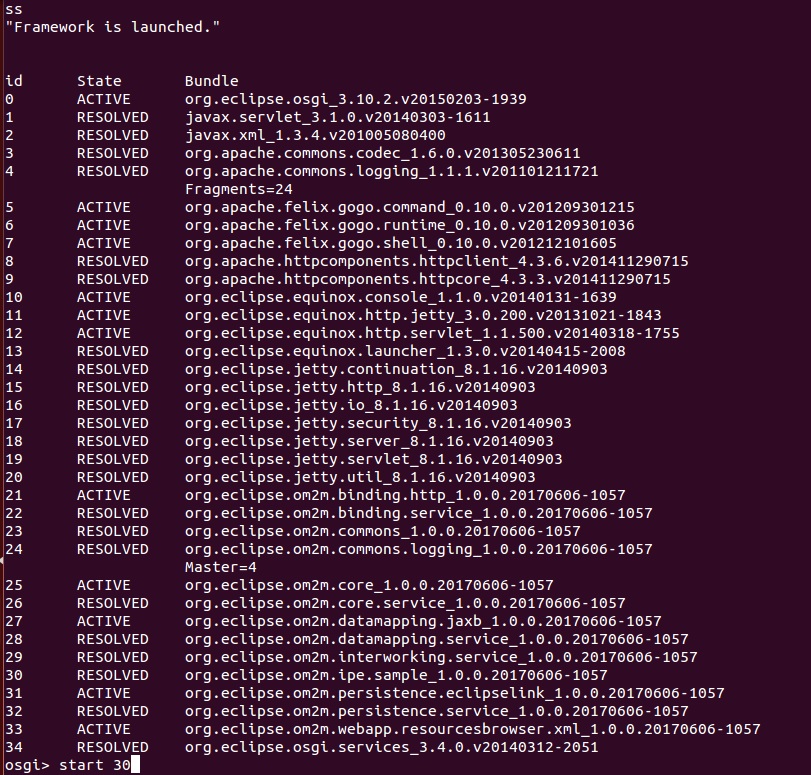
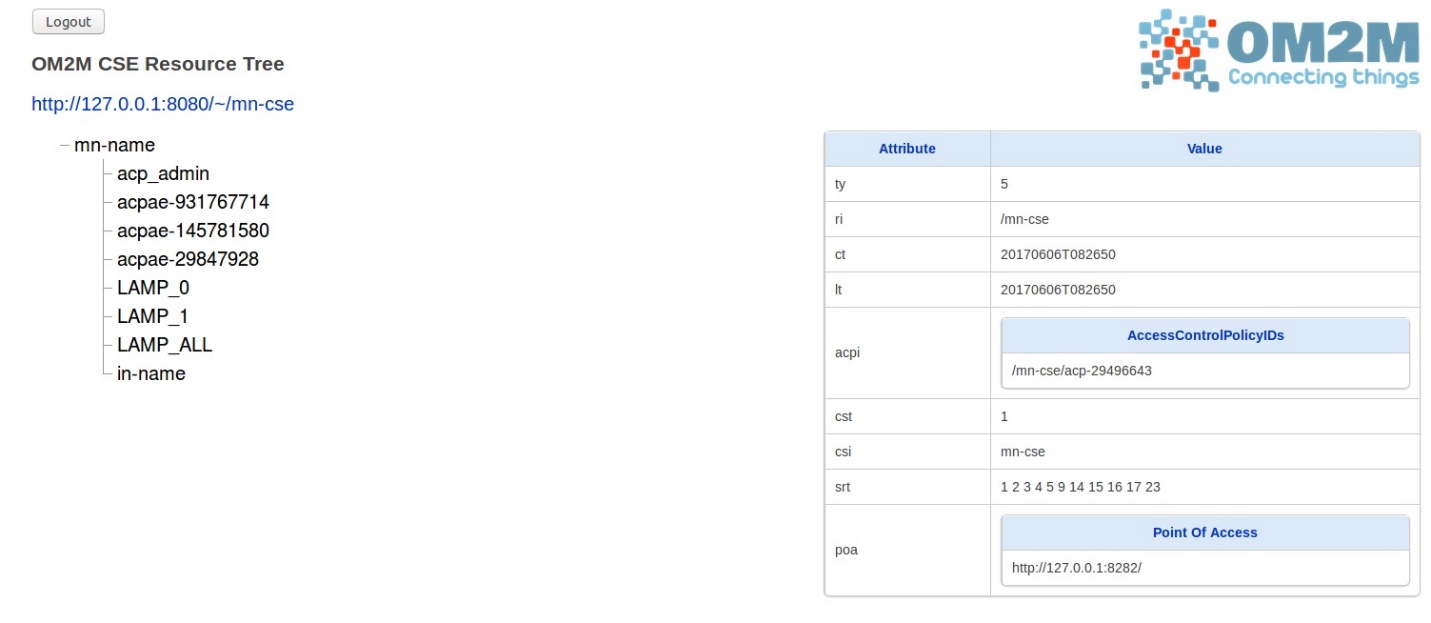
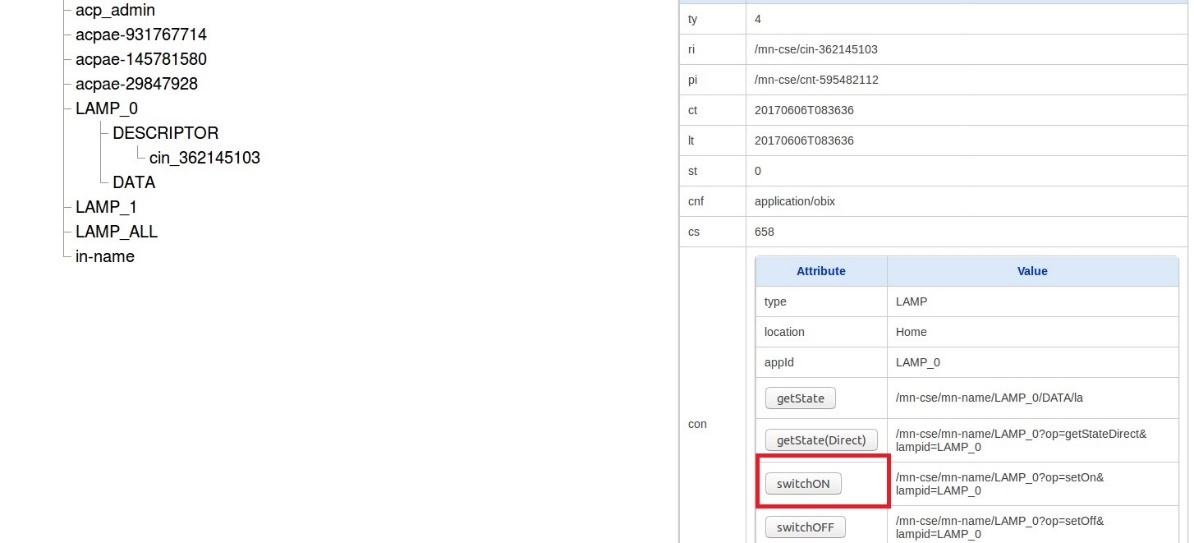
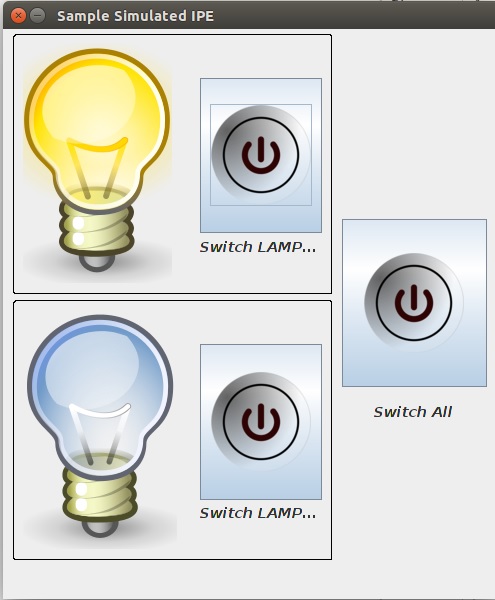


1. 使用預設選項後按下Next。
2. 點選Install



1. 完成後按下Finish
2. 點選Yes後重新開機。
3. 開啟VMware後選擇”Use VMware Workstation 12 Player for free for non-commercial use”，輸入Email點選Continue。
4. 按下Finish後結束並開始使用。



1. 開啟剛剛安裝好的VMware，點選” Create a New Virtual Machine”
2. 選擇中間的”Installer disc image file”並點選Browse選擇下載完成的Ubuntu系統的Image檔，按下Next。
3. 將Full name，Username、password、Confirm 填寫完成後，按下Next，以下欄位可自由填寫。
4. 填寫”virtual machine name”後按下Next。virtual machine name”可自由填寫。
5. 選擇Disk size，預設為20Gb，按下Next。
6. 按下Finish後開始安裝。
7. 安裝完成後輸入密碼進入系統桌面並開啟終端機以root身分執行Update和Upgrade。  
   執行: sudo apt-get update && sudo apt-get upgrade –y  
   此為完成之後之狀態圖。  
   
8. 安裝Java SDK 1.7  
   指令: sudo apt-get install openjdk-7-jdk  
   完成如下  
   
9. 輸入 java –version 檢查版本是否無誤。  
   
10. 安裝git。  
    指令: sudo apt-get install git –y  
    完成如下  
    
11. 安裝Maven。指令: sudo apt-get install maven –y  
    完成如下  
      
    檢查Maven版本 : mvn –version  
    
12. Git Clone Om2m one。  
    指令: git clone <https://git.eclipse.org/r/om2m/org.eclipse.om2m>  
    
13. 到剛剛clone完的org.eclipse.om2資料夾中執行 mvn clean install。完成如下  
    
14. 啟動Om2m one，  
    到org.eclipse.om2m/org.eclipse.om2m.site.in-cse/target/products/in-cse/linux/gtk/x86\_64裏頭執行start.sh，成功執行後會看到”CSE starteed”。  
    
15. In-cse啟動後到org.eclipse.om2m/org.eclipse.om2m.site.mn-cse/target/products/mn-cse/linux/gtk/x86\_64裏頭執行start.sh，成功執行後會看到mn-cse成功向in-cse註冊成功。  
    
16. Mn-cse成功向in-cse註冊成功後，可以開啟網頁瀏覽器在網址列打上http:127.0.0.1:8080/webpage開啟Om2m的頁面。帳號密碼皆為admin。
17. 登入之後可以看到在in-cse下有mn-cse存在。
18. 在啟動mn-cse終端機下達ss可以看到目前提供哪些服務 輸入start 30的指令可以開啟om2m內建的燈泡sample。
19. 啟動之後可以看到mn-cse出現了LAMP\_0、LAMP\_1及LAMP\_ALL。
20. 點選其中一個LAMP再點選DESCRIPTOR出現cin\_random\_number點選之後可以透過出現的按鈕控制燈泡發亮。  
      
      
    
21. oneM2M藍芽環境建置

在linux環境下安裝OM2M的平台，然後透過藍芽來進行傳值，那我們藍芽所使用的套件是用bluecove-2.1.0.jar跟bluecove-gpl-2.1.0.jar。

(bluecove-2.1.0.jar載點：<https://sourceforge.net/projects/bluecove/>

bluecove-gpl-2.1.0.jar載點：<https://zh-tw.osdn.net/projects/sfnet_bluecove/downloads/BlueCove/2.1.0/bluecove-gpl-2.1.0.jar/>)

接下來將介紹一些藍芽基本的function

LocalDevice.getLocalDevice()這個可以取得你電腦本身藍芽裝置的資訊。

getDiscoveryAgent()這個就像是你派出探員來搜尋附近的藍芽裝置。

startInquiry()開始搜尋附近裝置。

藍牙本身的搜尋機制就是Listener的機制，叫做DiscoveryListener。這個

Interface 包括了四個abstract的method

1. deviceDiscovered()：如果搜尋中找到一個裝置，就由JVM代為呼叫此method，通常在此method中，會把remoteDevice丟到一個Vector內做儲存的動作。
2. inquiryCompleted()：當搜尋結束(在藍牙可搜尋到的範圍內沒有新的裝置時，稱為結束)
3. servicesDiscovered()：每"一個"裝置找到所有services便由JVM呼叫此method
4. serviceSearchCompleted()：所有的裝置的service都被找到了，JVM呼叫此method

取到想要的藍芽裝置的address後，就可以進行連線與傳值。

RemoteDevice：此型態為附近的藍芽裝置。

RemoteDevice.getBluetoothAddress()：用來取得藍芽裝置的address。

StreamConnection streamConnection = (StreamConnection)：這裡宣告的streamConnection，主要是要用來傳值，以及接收值的處理。

Connector.open(Url)：將拿到的address(如url)，丟進去connector後，進行連線。

OutputStream os = streamConnection.openOutputStream()：為傳值給板子，所使用的物件。

os.write("1".getBytes())：傳值給板子的範例，範例為傳1給板子。

InputStream is = streamConnection.openInputStream()：為接收板子傳來的值，讀取用的物件。

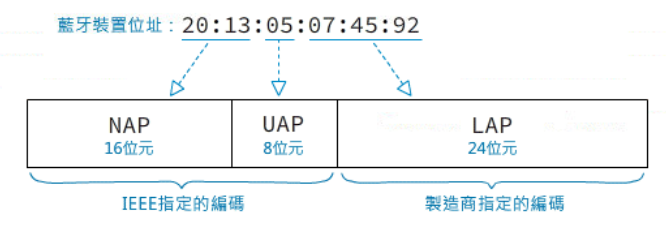
is.read(b)：用變數b，去接板子傳送過來的值。

藍芽address介紹

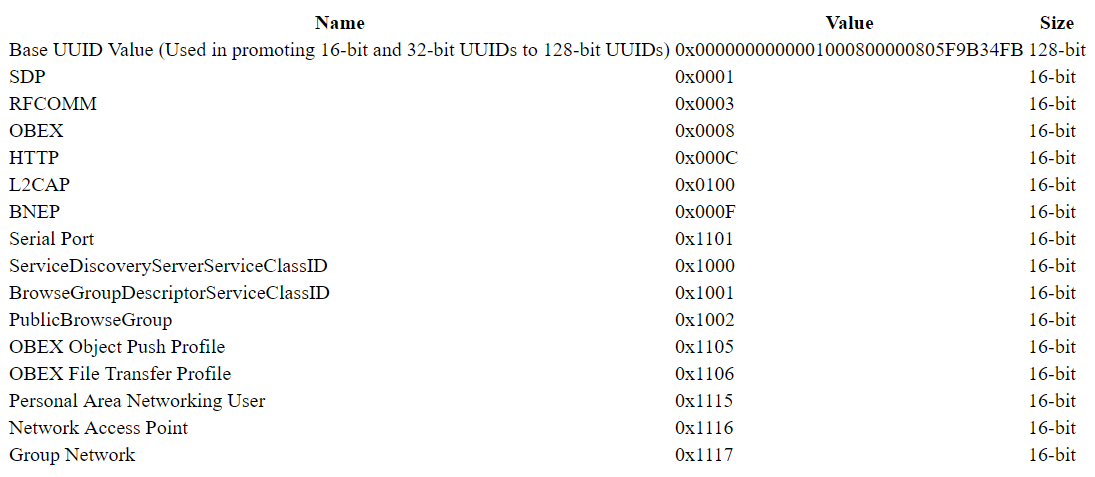
藍牙裝置位址（Bluetooth Device Address，簡稱BD\_ADDR）由一連串16進位數字組成，總長48位元。裝置位址相當於藍牙裝置的序號，每個裝置都不同，它可細分成三個部份：

NAP：未定義位址部分（Non-significant Address Part），由IEEE分配給製造商的編碼。

UAP：較高位址部分（Upper Address Part），用於產生錯誤偵測碼以及藍牙裝置的跳頻順序。

LAP：較低位址部分（Lower Address Part），製造商設定的編碼。

UUID介紹

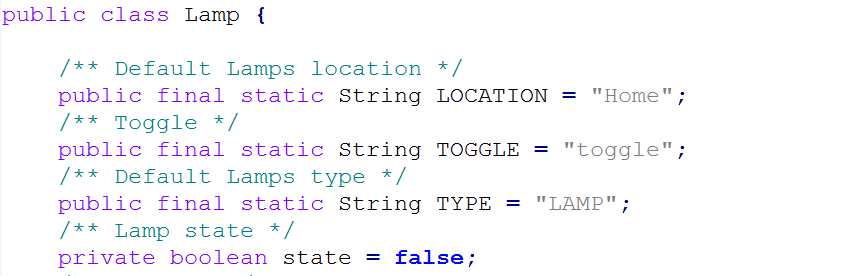
下圖為各種UUID，例如要查詢藍芽address的port時，在searchServices的function後面的引數有UUID[] uuidSet，這個陣列，uuidSet要是存放0x1101 ，則可以取得藍芽裝置的serial port。

1. Java新增oneM2M Application

以OM2M所提供的開放原始碼Lamp為範例進行講解功能，並結合上述所提到的藍芽功能，進行更改。

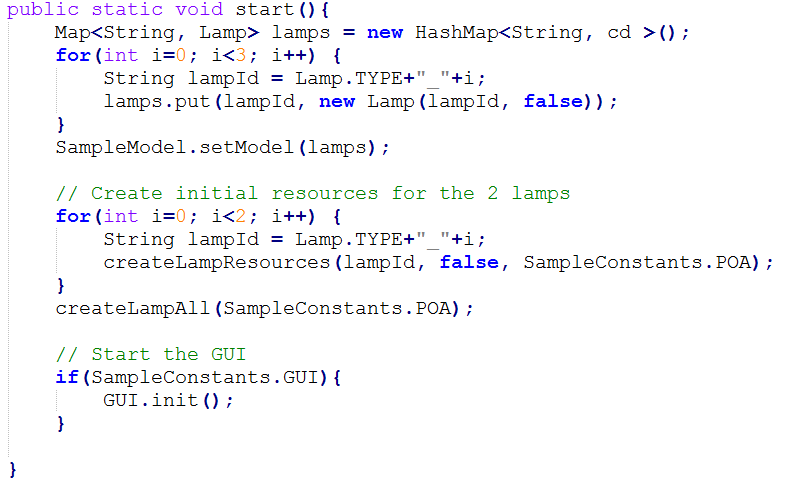
Lamp.java:

在Lamp.java裡面設置關於要新增的Application的基本設定，例如:Type，State等等。

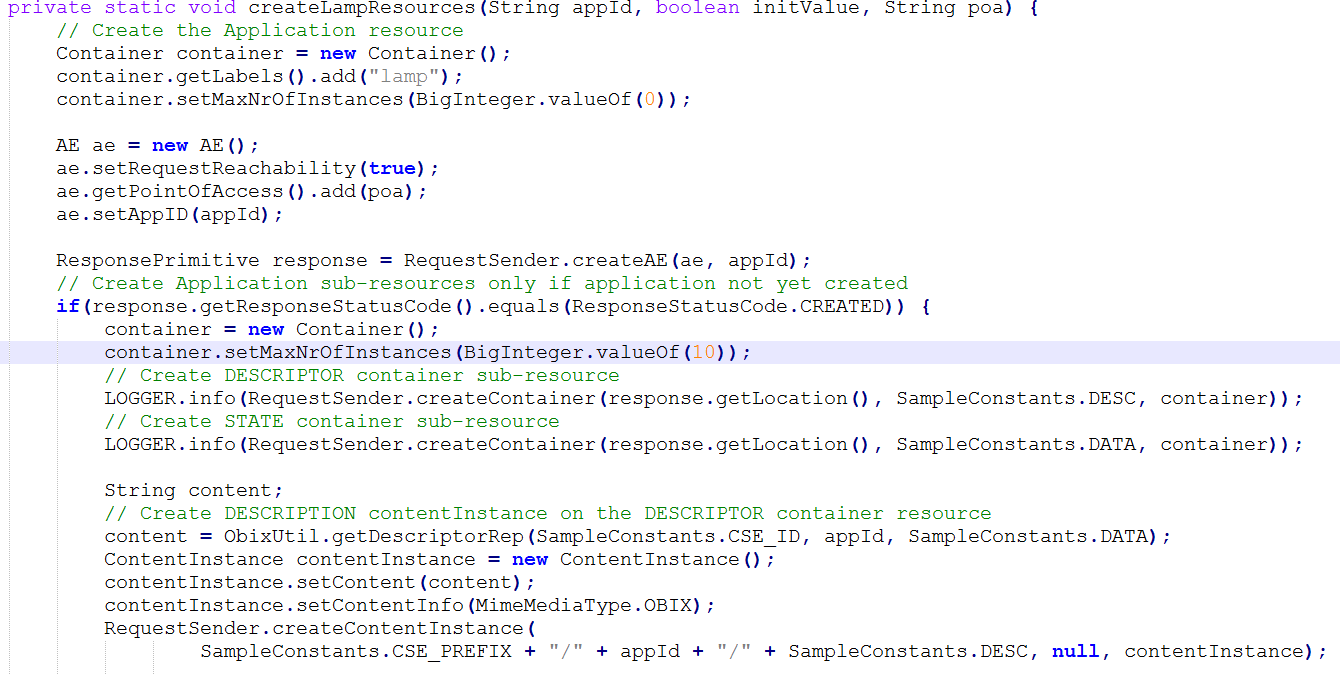


LifeCycleManger.java:

Start():設定要新增的Application名稱，並呼叫creatLampResource()。

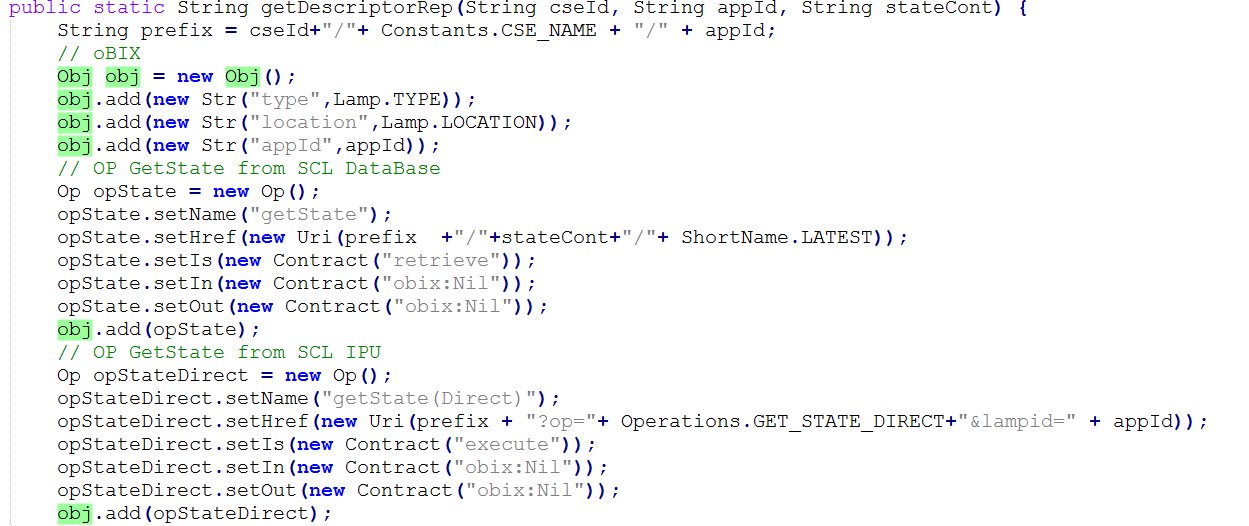


creatLampResources():透過裡面的Function創建剛剛新增的Application 吉它的Dercriptor跟Data的Container跟內容。

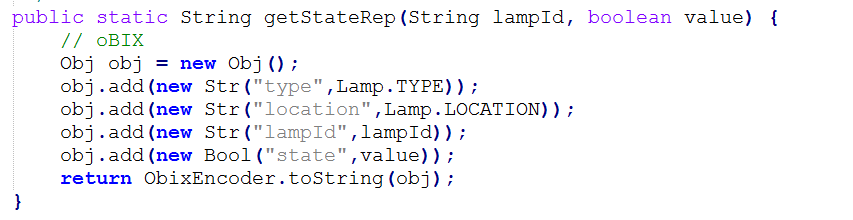


ObixUtil.java:

getDescriptorRep():設定Application 的Descriptor內容。

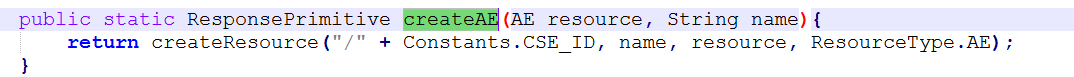


getStateRep():設定Application 的Data內容。

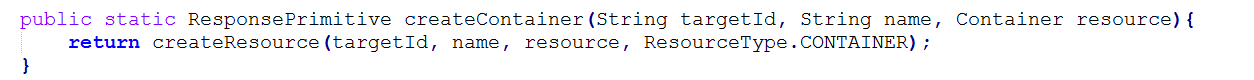


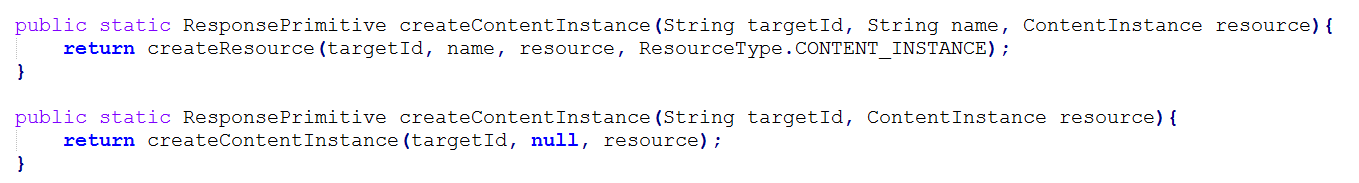
RequestSender.java:

createAE():創建Application。



createContainer():創建Application中的Descriptor和Data的container。



createContentInstance():透過這個function上傳要放到oneM2M網頁的資料。兩個createContentInstance()主要在於一個上傳Descriptor的資料疑另一個上傳Data的資料。

透過修改上述Function及結合藍芽之後可以將Linkit One開發版的值透過藍芽的方式將感測器讀取到的值上傳到OM2M的網頁上，在用手機可以透過HTTP或者藍芽的方式取得。

1. 使用Node-red完成IN-AE subscription to MN-AE

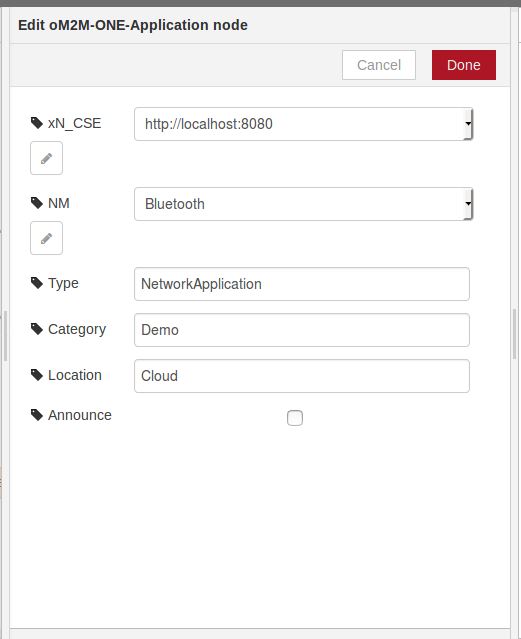
首先透過Lab3完成MN-AE的架構，包含了DESCRIPTOR Container & ContentInstance、DATA Container & ContentInstance及HTTP POST Service。

完成了MN-AE後再接著繼續完成IN-AE。

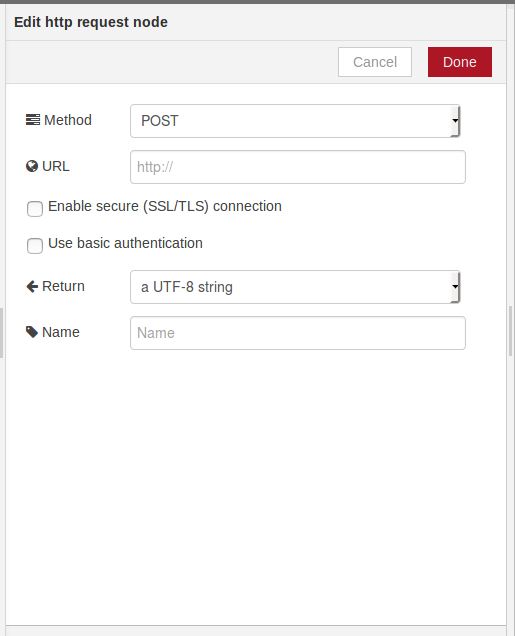
Step1. 首先創建一個oneM2M Application，如下圖



Step2. 雙擊oneM2M Application進行設定，如圖所示。



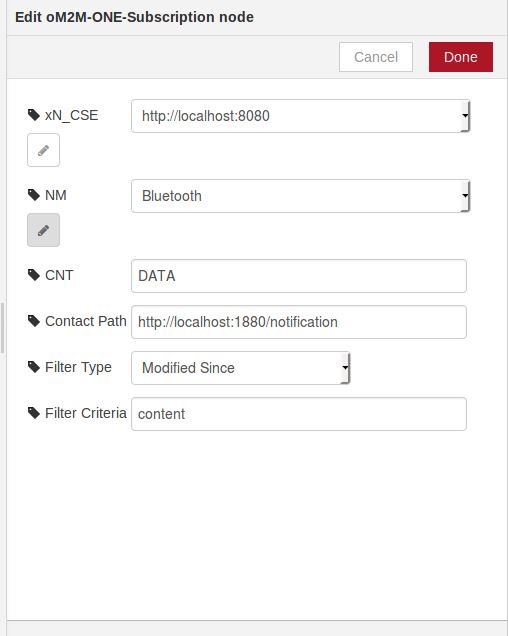
Step3. 點擊http request進行設定，如圖所示。



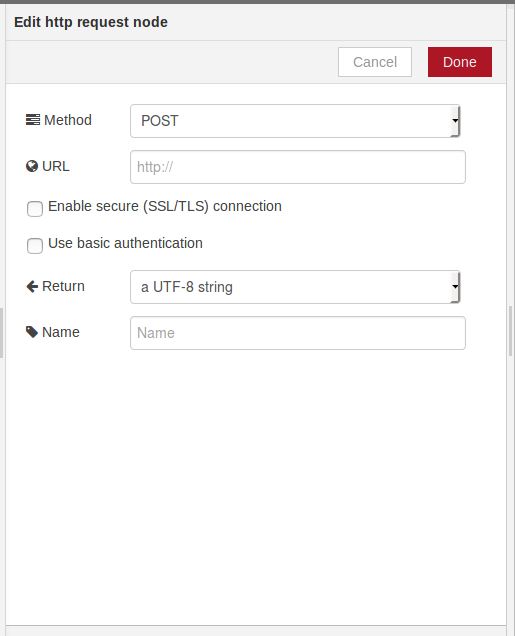
Step4. 創建一個oneM2M Subcription，如圖所示。



Step5. 點擊oneM2M Subcription進行設定，如圖所示。



Step6. 點擊http request進行設定，如圖所示。



Step7. 創建web service listening for notifications如圖所示。

Lissten HTTP POST: Input->http。

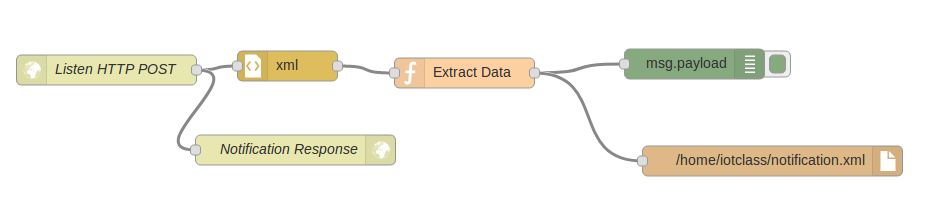
Extract Data: function->function 。

Xml: function->xml。

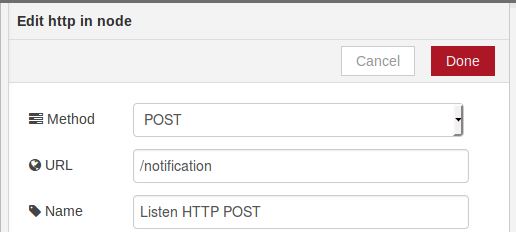
Notification Response: output->http response。

/home/iotclass/notification.xml: storage->file。

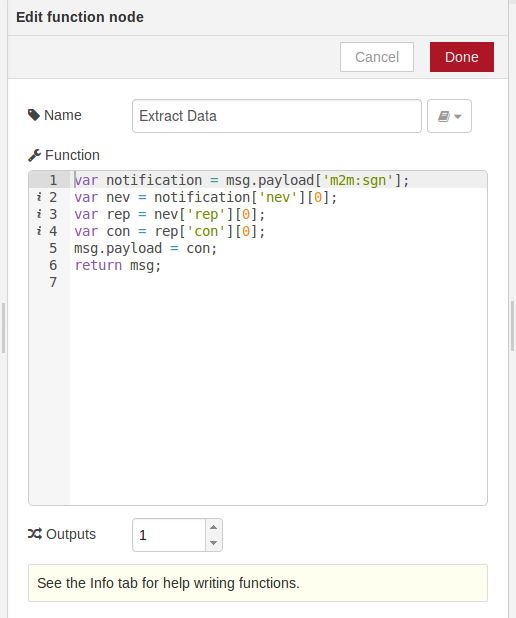
msg.payload: output->debug。



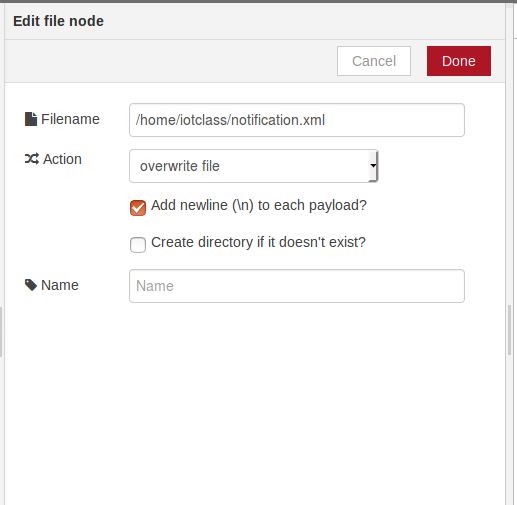
Step8. 點擊http修改資訊，如圖所示。



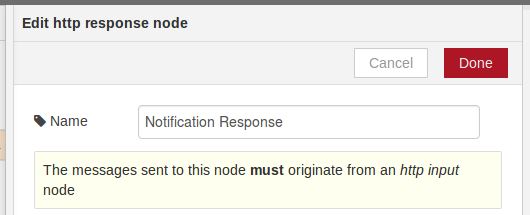
Step9. 點擊function進行修改，如圖所示。



Step10. 點擊File進行修改，如圖所示。



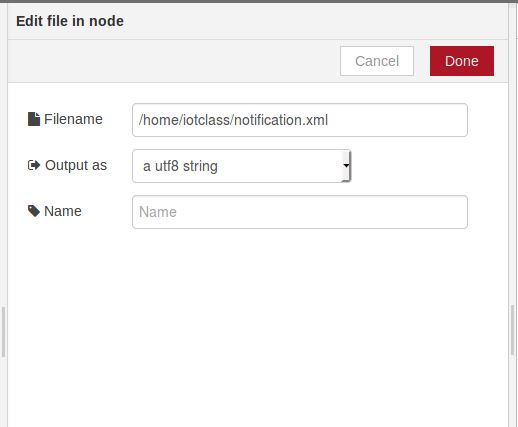
Step11. 點擊http response進行修改，如圖所示。



Step12. 創建HTTP GET web service如下圖。



Step13.修改file內容，如圖所示。



Step14. 創建additional HTTP GET Web Service，如下圖。

HTTP GET: function->http resquest。

/home/iotclass/notification.xml: storage->file。

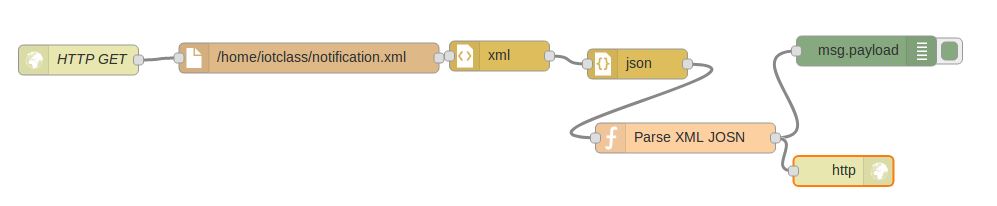
Xml: function->xml。

Json:function->json。

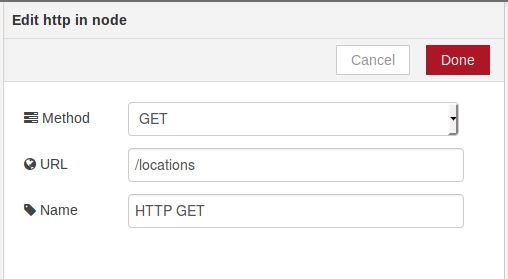
Parse XML JSON:function->function。

http:output->http response。

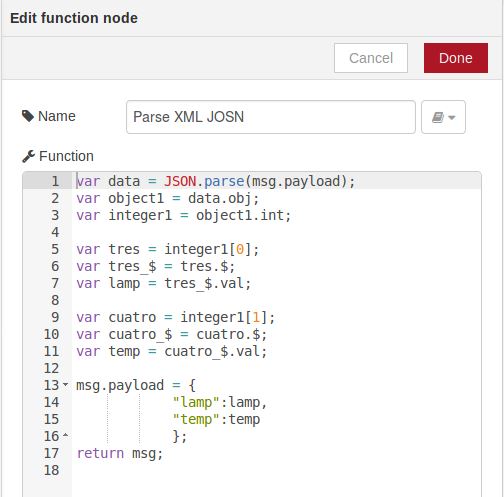
msg.payload: output->debug。



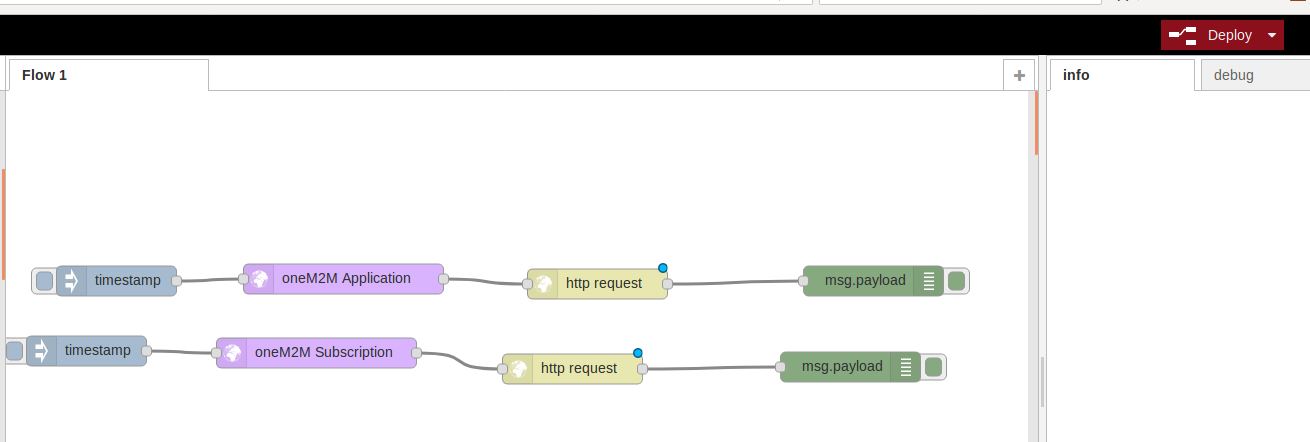
Step15. 修改http request，如圖所示。



Step16. 修改function內容，如圖所示。



Step17. 按下Deploy之後，在按下timestamp已啟用。

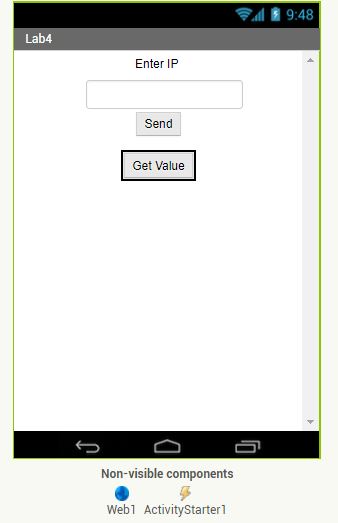


Step18. 可看到OM2M上，出現了SUB\_DATA，如圖所示。

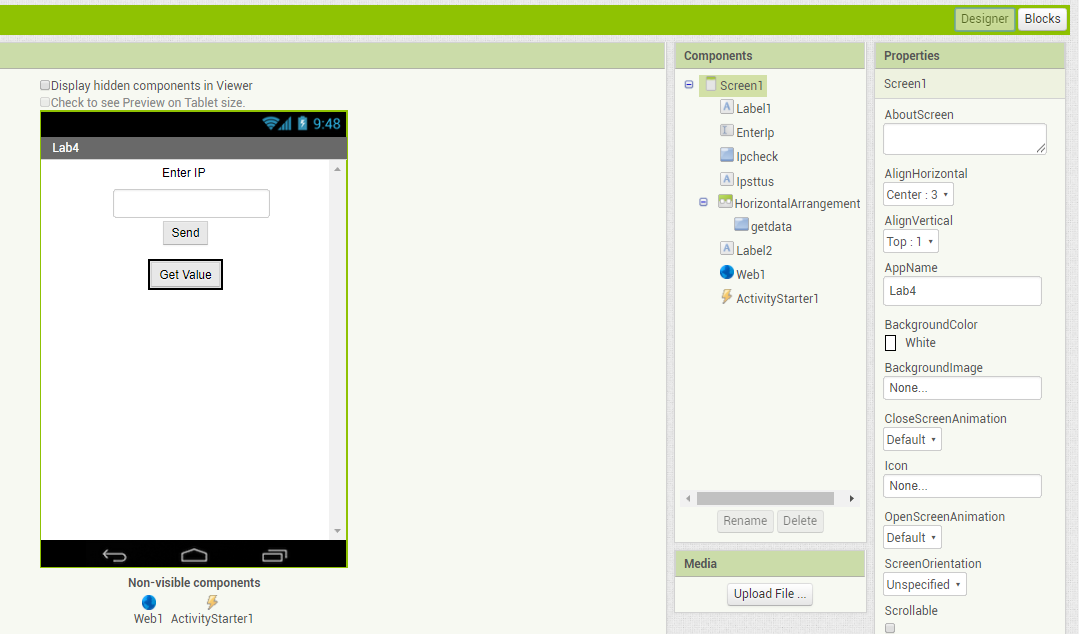


6. 使用App Inventor完成App讀取oneM2M資料

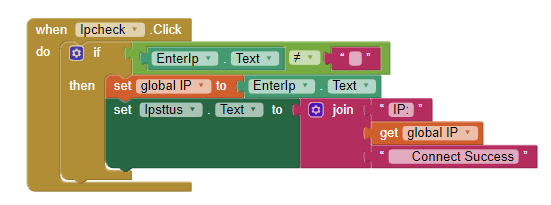
Step1. 新增Lab4 project，並完成以下元件。



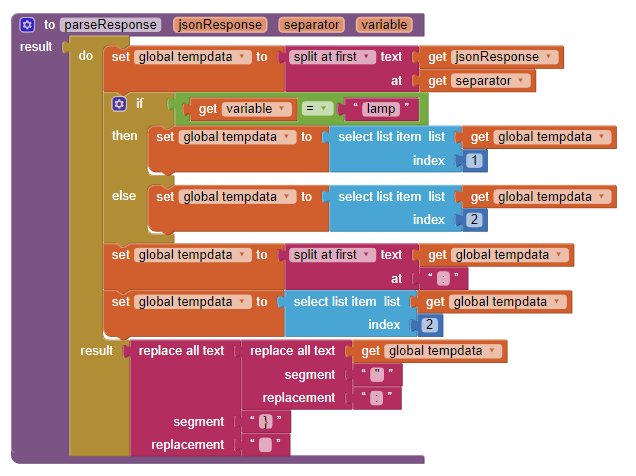
Step2. 切換到右上方blocks開始設計code。



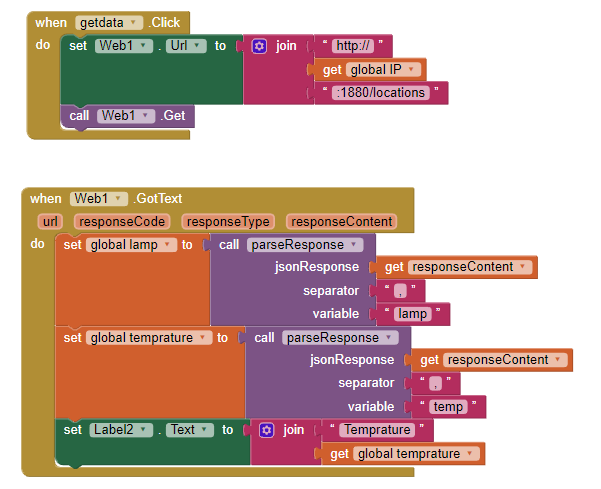
IP check:



Parse Json:



Get Data from IN-AE:



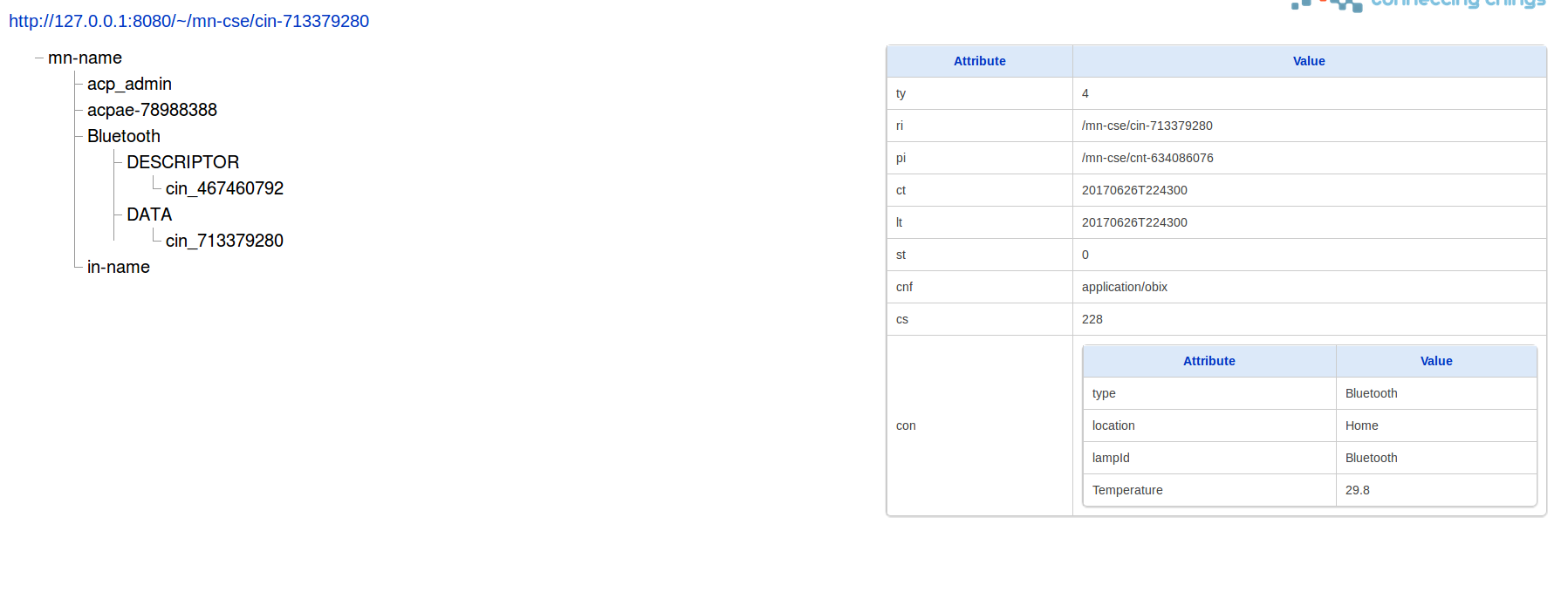
Lab4藍芽傳值，讀取資料

Part1 Linkit One透過藍芽傳送感測器得到的值到OM2M網頁上自己新增的Application上顯示，這邊使用溫度感測器作為範例。

1. 修改OM2M提供的Lamp範例來新增自己的Application包含Descriptor跟Data。

2. 將Linkit One讀取溫度感測器讀取到的值藉由藍芽上傳到OM2M的網頁上。

完成結果如下:



Part2手機可以透過App inventor進行開發app使其取得OM2M網頁上的資料。

1.使用Node-red完成IN-AE subscription to MN-AE

2.透過完成的App讀取OM2M上的資料。

完成結果如下。

操作方式:

1. 首先輸入OM2M所在的IP Address，按下Send後，會顯示Connect Success。
2. 按下Get Value後讀取資料，顯示於下方，如下圖。

