

此講義紀錄了 111-1 學期，在實驗上遇到問題之解套方式。其內容有：

1. CH5 解決舊版的 node-red 無法使用 manage Platte 的功能
2. CH8 解決新版 AWS 的安裝和執行 MQTT 的步驟
3. CH9 Docker 需留意的地方 (by 蔡嘉倫)

以上實驗皆跑在 RPi 環境：

```
Raspberry Pi Model3 B+ (2017)
Linux kernel: 4.19.66-v7+
Rip os version: Raspbian GNU / Linux version 9.13
GCC C++ compiler: Raspbian 6.3.0-18+rpi1+deb9u1 20170516
Glib: 2.24-11+deb9u4
setuptools 65.5.1
wheel 0.38.4
```

CH5 解決舊版的 node-red 無法使用 manage Platte 的功能

安裝 RPi OS 時，OS 內亦會建好 node-red (v1.X)系統，但原本系統內建的 node-red 版本太舊，這會影響我們使用 node-red 中 manage Platte 的功能(實驗步驟 5.3 會用到)，因此我們需要先刪除原本內建在 RPi 裡的版本，並安裝新的 node-red(v3.0.2)。

可以正確運行本實驗的環境：

```
Python 3.8.0
Npm 8.19.2
Node 16.18.0
Nore-red 3.0.2
```

Step1: 刪除 RPi 內有關 node-red 所使用的套件 Npm, Node, Node-red

```
sudo apt-get remove npm
sudo apt-get remove node
sudo apt-get remove node-red
```

Step2: 這邊要從 source code 安裝 npm。切記勿用 apt install npm 安裝，因為會此指令仍然會抓到舊版的 npm(v1.X)。首先到 RPi 的 terminal 打以下指令來下載

source code 。

```
wget https://nodejs.org/dist/v16.18.0/node-v16.18.0-linux-armv7l.tar.gz
```

Step3: 打以下指令來解壓縮檔案

```
tar -xzf node-v16.18.0-linux-armv7l.tar.gz
```

step4: 複製 node 到 /usr/local，請打指令

```
cd node-v16.18.0-linux-armv7l/  
sudo cp -R * /usr/local/
```

step5: 檢查 npm 和 node 是否有安裝好。
版本應是：Npm 8.19.2, Node 16.18.0

```
node -v  
npm -v
```

step6: 使用 npm 來安裝 node-red，請打指令

```
sudo npm install -g --unsafe-perm node-red
```

另外，此實驗還會用到 node-red 中 serialprot 的套件，若要透過 node-red 安裝此套件，請注意，RPI 的 Python3 環境必須 $\geq v3.6.0$ 以上。通常 RPI 內建的 Python3 環境通常是 v3.5.X。若要升級 Python3 到版本，請參考以下步驟：

Step1: 下載 Python 3.8.0

```
wget https://www.python.org/ftp/python/3.8./Python-3.8.0.tgz
```

step2: 解壓縮並進入 Python-3.8.0 的資料夾

```
tar -zxvf Python-3.8.0.tgz  
cd Python-3.8.0
```

step3: 執行 configuration command

```
./configure --enable-optimizations
```

Step4: 安裝 any missing dependencies

```
sudo apt update  
sudo apt install -y build-essential tk-dev libncurses5-dev libncursesw5-dev libreadline6-dev libdb5.3-dev libgdbm-dev libsqlite3-dev libssl-dev libbz2-dev libexpat1-dev liblzma-dev zlib1g-dev libffi-dev
```

step5: 安裝 Python

```
sudo make altinstall
```

CH8 解決新版 AWS 的安裝和執行 MQTT 的步驟

由於 AWS 提供使用 MQTT 的 package 在 2022 年 4 月有更新，因此原本 CH8 的實驗講義有些安裝和執行步驟亦須更改，才能完成此次實驗。詳細步驟請參考下方紀錄。

可以正確運行本實驗的環境：

```
Python 3.8.0  
cmake 3.7.2  
libssl-dev 1.1.0l-1~deb9u6  
git 2.11.0  
awscrt 0.14.7  
awsiot-sdk 1.11.9  
AWSIoTPythonSDK 1.5.2
```

step1: 同學在執行實驗步驟 8.1 前，請先在 terminal 執行以下步驟：

```
sudo apt-get update  
sudo apt-get upgrade  
sudo apt-get install cmake  
sudo apt-get install libssl-dev  
sudo apt-get install git  
git --version
```

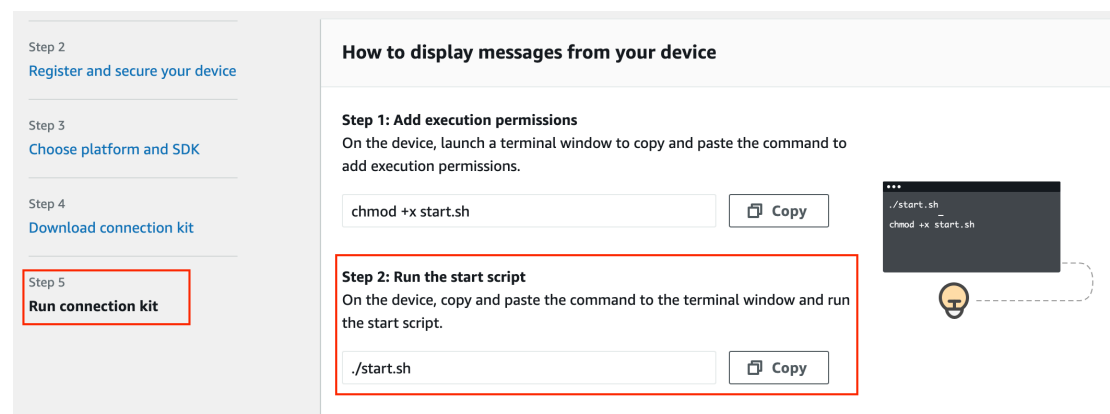
step2: 從 github 安裝 awscrt，請執行以下指令：

```
git clone https://github.com/aws-labs/aws-crt-python.git
cd aws-crt-python
git submodule update --init
python3 -m pip install .
```

來源：<https://pypi.org/project/awscrt/>

step3: 請執行原本實驗講義 CH8-1 和 8-2 的步驟，此步驟是為了讓 RPi 和 AWS 做溝通，因此需要透過 AWS 的平台註冊和下載相對應執行檔在 RPi 上。AWS 現在網頁上的介面和講義 CH8-1 和 8-2 的步驟所截圖的畫面會有些不同，但不用擔心，這不影響註冊和安裝流程。不過請注意，當 AWS 的官網上的步驟請我們要執行 start.sh 的檔案時[如圖一]，請先更改 start.sh 的第 26 行程式碼如下，並請注意要安裝 awsiotsdk 的套件時，python3 的環境需 ≥ 3.5 以上。

```
python3 -m pip install awsiotsdk
```



圖一

Step4: 請將 PM2.5 sensor 接至 Arduino，再接至 RPi 上。詳細接法請參考實驗講義 5.3。接著，請下載 testPM2.5_aws.py (此為因應上述新版 AWS 套件所寫)，連結如下：

<https://www.space.ntu.edu.tw/navigate/s/1C295D5106664E979EBFE8EBE5CD847DQQY>

Step5: 將 testPM2.5_aws.py 的檔案放在路徑 aws-iot-device-sdk-python-v2/samples/ 下，並執行以下指令。

```
cd ../../  
python3 aws-iot-device-sdk-python-v2/samples/testPM2.5_aws.py --endpo  
int [your API endpoint] --ca_file [your Root CA] --cert [your certific  
ate]--key [your private key]--client_id [your client id] --topic sdk/t  
est/Python --count 0
```

關於 [your API endpoint] / [your Root CA]/ [your certificate] / [your private key] / [your client id] 的訊息可以在 start.sh 的第 36 行找到，此為我們在安裝 AWS MQTT 的執行檔到 RPi 時，AWS 幫我們設定好的。以我的例子而言，我會下指令：

```
pi@raspberrypi:~/aws $ python3 aws-iot-device-sdk-python-v2/samples/testPM2.5_aws.py --e  
ndpoint a3c7o1yavmawig-ats.iot.ap-northeast-1.amazonaws.com --ca_file root-CA.crt --cert  
myRPi.cert.pem --key myRPi.private.key --client_id basicPubSub --topic sdk/test/Python  
--count 0
```

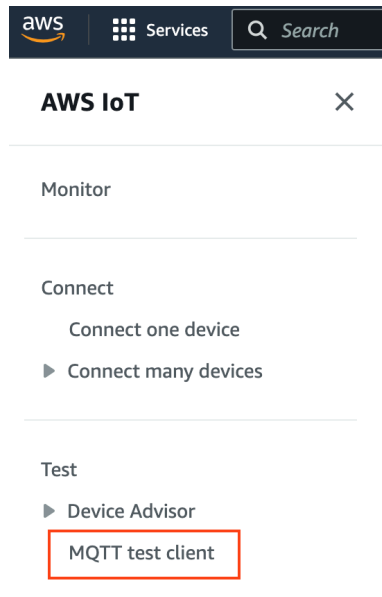
Step5: 如果 RPi 有順利連接至 AWS，則可以看到 PM2.5 sensor 的數值[如圖二]。

```
Connecting to a3c7o1yavmawig-ats.iot.ap-northeast-1.amazonaws.com with client ID 'basicP  
ubSub'...  
Connected!  
Subscribing to topic 'sdk/test/Python'...  
Subscribed with QoS.AT_LEAST_ONCE  
Sending messages until program killed  
-149.52 0  
Publishing message to topic 'sdk/test/Python': -149.52 [1]  
Received message from topic 'sdk/test/Python': b'" -149.52 [1]"'  
-149.52 0  
Publishing message to topic 'sdk/test/Python': -149.52 [2]  
Received message from topic 'sdk/test/Python': b'" -149.52 [2]"'  
-149.52 0  
Publishing message to topic 'sdk/test/Python': -149.52 [3]  
Received message from topic 'sdk/test/Python': b'" -149.52 [3]"'  
-149.52 0  
Publishing message to topic 'sdk/test/Python': -149.52 [4]  
Received message from topic 'sdk/test/Python': b'" -149.52 [4]"'  
-149.52 0  
Publishing message to topic 'sdk/test/Python': -149.52 [5]  
Received message from topic 'sdk/test/Python': b'" -149.52 [5]"'  
-149.52 0  
Publishing message to topic 'sdk/test/Python': -149.52 [6]  
Received message from topic 'sdk/test/Python': b'" -149.52 [6]"'
```

圖二

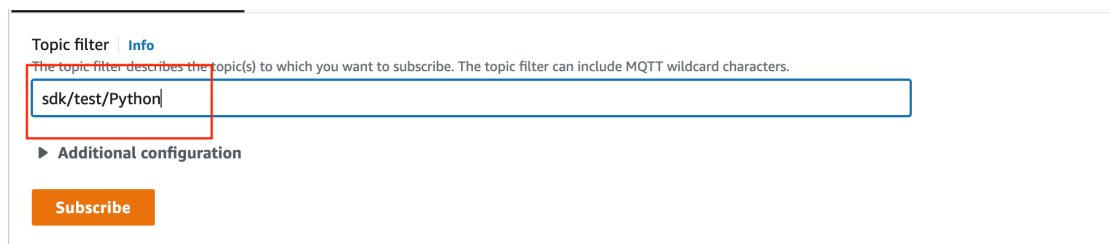
[註]: 但因為助教的 Arduino pin 腳讀值有些問題，所以偵測的值為負。正常來說，PM2.5 的值會落在 0-3000。

Step6: 我們亦可以用 AWS 提供的 client service 去確認 RPi 是否有連上 AWS。首先，進入 AWS IoT，並進入 [Aws IoT > Test > MQTT test client](#) [如圖三]

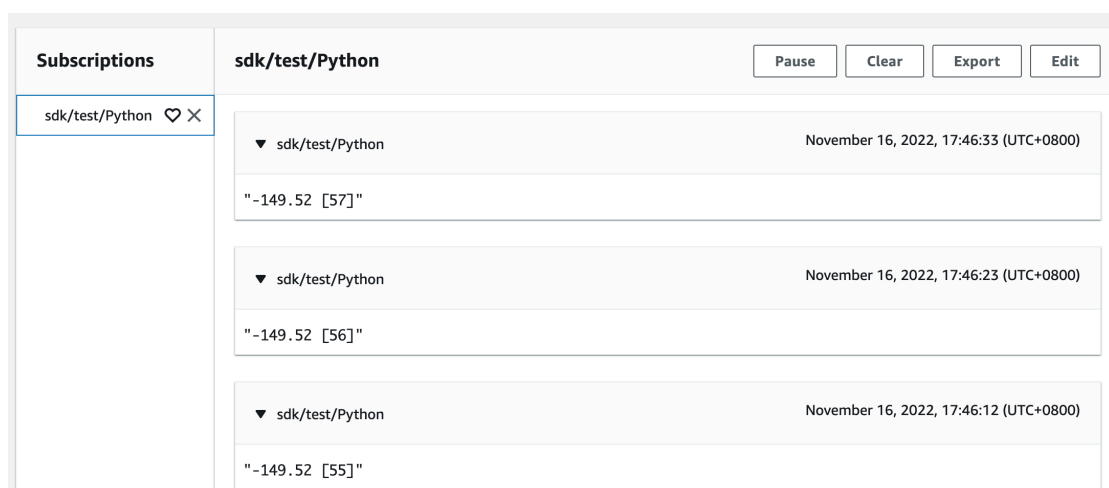


圖三

Step7: 並訂閱 sdk/test/Python [如圖四]，如果連線成功，AWS 的平台會顯示 PM2.5 的值[如圖五]。



圖四



圖五

CH9 Docker 需留意的地方

可以正確運行本實驗的環境：

Docker 20.10.21

Rancher 1.21.6

K8s 1.21.6

- 在 9.6.1 的 `rancher/rancher:latest` 的地方，得改成 `rancher/rancher:v1.21.6` 的 `version`，才能避免遇到無法連線到 `k8s kernel` 而無法啟動的問題。
- 在使用 `k8s` 時，會有網頁不穩的情況，會有可能在連線成功後仍無法連線，偶爾需要刷新頁面。