

國立台北科技大學電子工程系 106 年度

實務專題成果報告書

基於 LoRa 之網頁物聯網控制技術  
(LoRa based Web of Things Control Technology)

專題執行期間：106 年 07 月 01 日起至 107 年 05 月 31 日止

專題發表時間：一百零七年六月八號

四子三甲 104360017 張丞志

四子三甲 104360043 陳宥軒

指導老師 李昭賢

中 華 民 國 106 年 05 月 26 日

# 目錄

一、摘要 .....	1
1.1 中文摘要.....	1
1.2 英文摘要.....	1
二、專題預期研究目標.....	2
2.1 研究背景.....	2
2.2 研究動機.....	2
2.3 研究目的.....	2
三、專題研究方法與進行步驟.....	3
3.1 預期完成功能.....	3
3.2 研究方法.....	8
四、研究成果 .....	16
4.1 server 端.....	16
4.2 web 端.....	18
五、參考文獻 .....	23

## 圖目錄

圖 1 整體系統架構圖 .....	3
圖 2 DASHBOARD 頁面 .....	4
圖 3 DASHBOARD 頁面之圖表顯示 .....	5
圖 4 FREEBIRD. JS 啟動時的輸出訊息 .....	5
圖 5 FREEBIRD. JS 收到資料及結束畫面 .....	6
圖 6 完成模組之輸出訊息畫面 .....	7
圖 7 完成系統之輸出訊息畫面 .....	7
圖 8 LoRA MODULE 架構圖(一) .....	9
圖 9 LoRA MODULE 架構圖(二) .....	9
圖 10 MQTT PUB/SUB 流程 .....	10
圖 11 SERVER 系統流程圖 .....	11
圖 12 LoRA PICO 無線型閘道器 .....	12
圖 13 SMART BLOCK 架構圖 .....	12
圖 14 GATEWAY 相關設定之一 .....	13
圖 15 GATEWAY OTAA 認證設定 .....	13
圖 16 GATEWAY ABP 金鑰設定 .....	14
圖 17 SIVANN WEATHER STATION MODULE .....	15

圖 18 COAP 協議 .....	15
圖 19 LoRA 架構圖 .....	17
圖 20 智慧工廠概念圖 .....	17
圖 21 WEBPACK 打包訊息 .....	17
圖 22 FREEBIRD 啟動訊息 .....	18
圖 23 裝置狀態改變訊息 .....	18
圖 24 感測器的數值改變訊息 .....	18
圖 25 系統訊息輸出 .....	19
圖 26 裝置的向力圖 .....	20
圖 27 裝置的概念圖 .....	20
圖 28 裝置圖表 持續更新 .....	21
圖 29 裝置圖表 單日 .....	21
圖 30 裝置圖表 特定範圍日期 .....	22
圖 31 設定上限值後的圖表 .....	22
圖 32 數值超過範圍後的圖示 .....	23

# 一、摘要

## 1.1 中文摘要

現今的物聯網(IoT)技術可以被應用在各式各樣的場景，例如:智慧城市、智慧工廠和智慧農業。即使目前已經有許多無線存取技術的存在，像：3G/4G、Wi-Fi、Zigbee或藍牙(Bluetooth)，它們能夠使大量的感測器及裝置互相連接，但是全新的無線通訊仍然被持續的修改及定義。舉個例子，LoRa 技術被提出用來提供長距離及低功耗的無線存取。隨著物聯網的快速發展，物聯網的下個階段應該是結合物聯網及網頁服務的 Web of Things (WoT)。在這個專題裡，我們將會聚焦在 WoT 的軟體開發，也就是說，我們基於一套知名的開源專案 FreeBird.js，它支援使用 Zigbee、Bluetooth、CoAP 和 MQTT 交換資料，本次專題將延伸並使它更進一步地支援 LoRa，最後，我們將會實作一個網頁介面以視覺化顯示數據。

關鍵字：Web of Things (WoT)、LoRa、Node.js

## 1.2 英文摘要

A The Internet of Things (IoT) technology can be applied to various scenarios, e.g., smart city, smart manufacturing, and smart agriculture. Even if a lot of wireless access techniques, e.g., 3G/4G, Wi-Fi, Zigbee, or Bluetooth, enable to interconnect a mass of sensors and devices, brand-new wireless communications are still defined and devised continuously. For example, the LoRa technology is proposed to provide long-distance and low-power wireless access. With the rapid development of IoT, the next stage of IoT should be the integration of IoT and web service called Web of Things (WoT). In this project, we focus on the software development of WoT. In other words, we utilize and extend one famous open-source project called Freebird.js, that exchanges IoT data over Zigbee, Bluetooth, CoAP and MQTT, to further support LoRa. Finally, we implement one Web interface to realize data visualization in this project.

Keywords：Web of Things (WoT)、LoRa、Node.js

## 二、專題預期研究目標

### 2.1 研究背景

在近年來物聯網快速的發展下，各式各樣不同的協定一直被制定出來，其相關服務與應用也越來越多，一台手機上可能安裝了各種不同產品的應用程式，因此，物聯網的下一個階段就是要將每個物件(things)與網路(web)相結合，使網路可以存取每個物件，且使用者可以透過網路來使用各種不同協定的服務，這即是 Web of Things(WoT)。

雖然目前有各種無線存取技術的存在像是:Zigbee、藍芽(Bluetooth)、3G/4G、Wi-Fi，但是沒有一種是為長距離低功耗而設計的，所以 LoRa 被提出來解決這個問題，像是在物流追蹤中，電池的壽命對定位及追蹤來說都十分重要，因此相當適合利用 LoRa 技術。

### 2.2 研究動機

隨著物聯網科技的進步，每個人的手機多少都有一兩款可以控制智慧物件的應用程式，但是每當要使用不同產品的服務時往往需要使用其專屬的應用程式，程式一多便顯得十分雜亂。因此我們想試著用 WoT 技術並且結合近幾年來被制定出的 LoRa 無線存取技術，搭建出一套物聯網系統，讓使用者可以通過網路頁面存取各式各樣不同裝置，並且可以在我們設計出的網路頁面上觀測到各種不同的數據。

### 2.3 研究目的

透過設計一伺服器可供各種異質網路(Zigbee、Bluetooth、CoAP、MQTT 及 LoRa)連接，結合不同無線存取技術的感測器來收集數據，且可智能判斷與決策，並主動與其他設備交換資訊，不需人力介入即可依據實際狀況進行運轉調配，提升材料、能源、人力等資源之利用效率並主動與其他設備交換資訊。

### 三、專題研究方法與進行步驟

#### 3.1 預期完成功能

##### (一) 專題計畫之架構

此計畫主要分為兩個部分:物聯網設備管理平台及 Web 端

##### 物聯網設備管理平台:

本專題採用 Node.js 做為開發平台，並使用 Express.js 作為 Web 的開發框架，另外還搭載了 Freebird.js 套件，使其可與多項不同協定的設備連接。除此之外，本專題將會撰寫一 JS Module 使其可與 LoRa 設備溝通，並將自行撰寫的 JS Module 接上 Freebird.js。

##### Web 端:

網頁端用到了 HTML、Javascript、CSS，框架則使用 JQuery、Bootstrap 及 D3.js，HTML 和 CSS 做網頁編排的部分，Javascript 用來控制 HTML 的呈現方式，

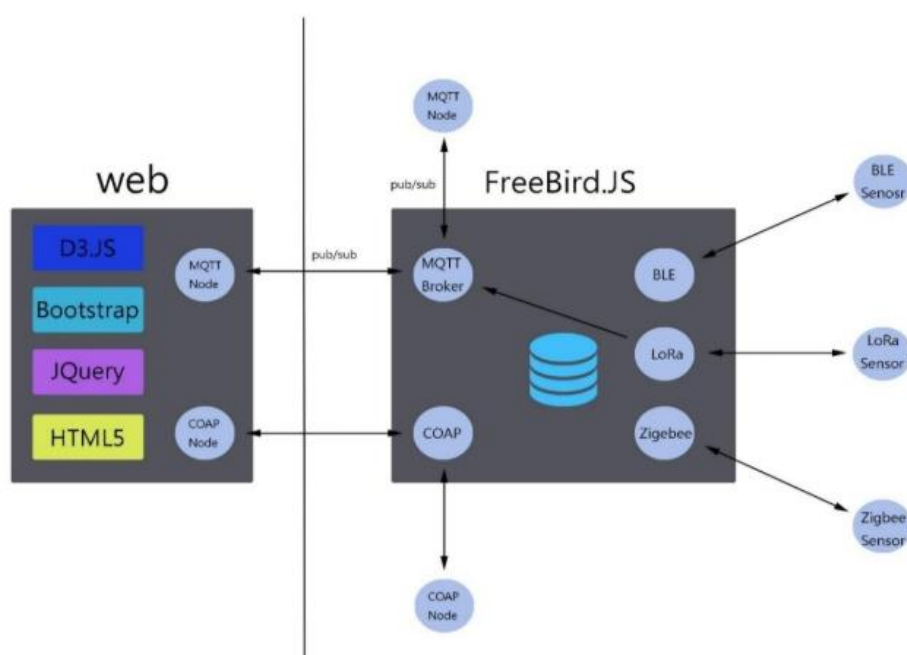


圖 1 整體系統架構圖

JQuery 是 Javascript 的框架之一，利用現成的函式加速開發速度，Bootstrap 用來美化網，D3.js 負責數據可視化的部分。

各種不同協定的裝置可以連上 Freebird.js 伺服器，伺服器會將收到的資料存進資料庫裡，並且使用者可以通過網頁端的 MQTT/CoAP Node 像伺服器獲取資料庫的資料，並透過各種框架顯示出來。

預期完成項目 1：

設計 Dashboard 頁面，在此頁面會顯示所有連接過的裝置，並且能顯示裝置的狀態，還可以查看此設備收到的資料圖表。

完成度：

已將 Dashboard 頁面完成如下圖所示，目前已經可將連接過的裝置顯示出來，並且當滑鼠移到裝置圖示上時可以顯示該裝置的資料及狀態如，並且畫面左側會顯示執行了哪些動作的訊息如 Figure 2。

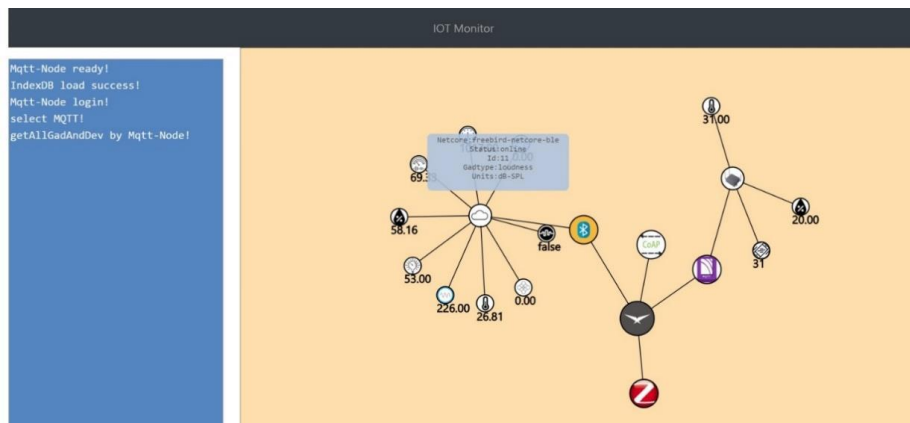


圖 2 Dashboard 頁面

且當滑鼠點擊裝置圖示時，將會彈出視窗顯示圖表，在視窗左上方有一選單可以選擇顯示的模式，Tracking 模式代表觀測網頁自連上伺服器後收到的感測器值，另外兩種模式是可以觀看單一日期及特定日期。圖表的 Y 軸為感測器數值，X 軸為日期，Figure 3 為 Tracking 模式下繪製的分貝計圖表。



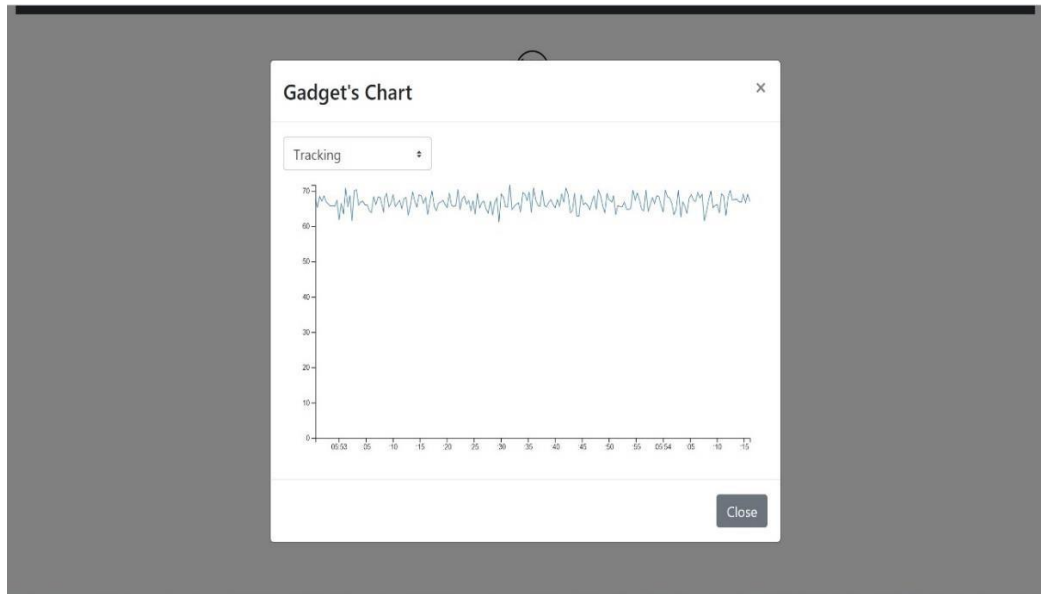


圖 3 Dashboard 頁面之圖表顯示

預期完成項目 2：

架設一個 Freebird.js 的伺服器，並使設備可以連接上該伺服器。且可通過 Freebird.js 介面來對設備進行操作及接收資料。

完成度：。

Freebird.js 啟動時會輸出目前監聽的埠、網頁的網址，接著會顯示此專案支援的協定及一些資料如 Figure 4，並且當收到資料時會以 Json 格式顯示出來，如 Figure 5 收到的資料有新的感測器數值、上一次收到的感測器數值、裝置的實體位置、裝置在 freebird 裡的編號及裝置是屬於何種通訊方式

```
"D:\JetBrains\WebStorm 2017.2.4\bin\runnerw.exe" "C:\Program Files\nodejs\
==> Listening on port 8000.
==> Open up http://localhost:8000/ in your browser.

Welcome to freebird-demo webapp...

77                                freebird v0.1.9                                77
72927
288888477                                77488882
7888888889277                                7710888888807
708888888888427                                714888888888847
70888880888888                                88888808888847
70888800409807880040888847
7088809488099088847
7480049490857
80888888                                A network server and
888470888                                manager for
1887                                7887                                heterogeneous
77                                27                                machine network

>>> Protocols: BLE, Zigbee, MQTT, CoAP, Modbus
>>> Version: freebird v0.1.9
>>> Document: https://github.com/freebirdjs/freebird/wiki
>>> Copyright (c) 2017, The MIT License (MIT)

The server is up and running, press Ctrl+C to stop server.

-----
```

圖 4 freebird.js 啟動時的輸出訊息

```

auxId: '0xbb80.0xcc08.61',
id: 5,
ncName: 'freebird-netcore-ble' }
{ data: { sensorValue: 57.97247314453125 },
  _data: { sensorValue: 57.7969970703125 },
  permAddr: '0x20c38ff1c09d',
  auxId: '0xbb80.0xcc08.61',
  id: 5,
  ncName: 'freebird-netcore-ble' }
{ data: { sensorValue: 58.12506103515625 },
  _data: { sensorValue: 57.97247314453125 },
  permAddr: '0x20c38ff1c09d',
  auxId: '0xbb80.0xcc08.61',
  id: 5,
  ncName: 'freebird-netcore-ble' }
{ data: { sensorValue: 58.239501953125 },
  _data: { sensorValue: 58.12506103515625 },
  permAddr: '0x20c38ff1c09d',
  auxId: '0xbb80.0xcc08.61',
  id: 5,
  ncName: 'freebird-netcore-ble' }
{ data: { sensorValue: 58.27001953125 },
  _data: { sensorValue: 58.239501953125 },
  permAddr: '0x20c38ff1c09d',
  auxId: '0xbb80.0xcc08.61',
  id: 5,
  ncName: 'freebird-netcore-ble' }

```



```

>>> This is a simple demonstration of how the freebird works.
>>> Please visit the link to know more about this project:
>>> https://github.com/freebirdjs/freebird/wiki

```

圖 5 freebird.js 收到資料及結束畫面

預期完成項目3：

設計出一JS Module使其可以接收LoRa設備傳送的資料。

完成度：

此模組啟動時將會顯示啟動訊息，當收到資料時會以 Json 的格式顯示出來，如 Figure 6 ，訊息裡會有此次傳輸的頻道、時間、閘道器的資料及訊號強度還有感測器的資料等等。

```
"D:\JetBrains\WebStorm 2017.2.4\bin\runnerw.e
server is started...
Server is ready.
[ { channel: 923375000,
  sf: 10,
  time: '2018-04-29T12:04:17',
  gwip: '192.168.1.104',
  gwid: '00001c497bc0dde9',
  repeater: '00000000ffffffff',
  systype: 10,
  rssi: -12,
  snr: 26.3,
  snr_max: 40,
```

圖 6 完成模組之輸出訊息畫面

預期完成項目4：

將自己設計的JS Module與Freebird.js 結合，使Freebird.js可以收到LoRa設備傳送的資料。

完成度：

最後修改完成的系統執行時會先顯示一些啟動訊息，當有裝置的狀態改變時會把該裝置的編號、實體位置、連接方式、及狀態顯示出來，收到訊息時則以 Json 格式顯示。

```
"D:\JetBrains\WebStorm 2017.2.4\bin\runnerw.exe" "C:\Program Files\nodejs\node.exe" D:\code\freebird-demo-master
Express server started on port 8000
LoRa server is started...
db load success!
webpack built 9384fd7ebe6a05834b70 in 4893ms
Version: webpack 3.10.0
Time: 4893ms
   Asset      Size  Chunks             Chunk Names
  main.js  2.33 MB          0  [emitted]  [big]  main
  index.html  4.29 kB          0  [emitted]
Child html-webpack-plugin for "index.html":
   Asset      Size  Chunks             Chunk Names
  index.html  1.49 MB          0
webpack: Compiled successfully.
dev is undefined
[   ready   ]
[Server Started]
[ devNetChanged] dev.id:3 |dev.Addr:00:11:22:aa:bb:cc/iNode |dev.ncName:my_iot_mqtt |dev.status:online
[ devNetChanged] dev.id:1 |dev.Addr:00:11:22:aa:bb:cc/LoRaGateway1 |dev.ncName:my_iot_mqtt |dev.status:online
{ data: { sensorValue: '25' },
  _data: { sensorValue: 31 },
  permAddr: '00:11:22:aa:bb:cc/LoRaGateway1',
  auxId: 'temperature/0',
  id: 1,
  ncName: 'my_iot_mqtt' }
```

圖 7 完成系統之輸出訊息畫面

## 3.2 研究方法

### A. 軟體方面：

#### Node.js:

Node.js 是一個能夠在伺服器端運行 JavaScript 的開源、跨平台 JavaScript 執行環境。Node.js 使用事件驅動、非阻塞和非同步輸入輸出模型等技術來提高效能，可優化應用程式的傳輸量和規模。

Node.js 大部分基本模組都用 Javascript 語言編寫。在 Node.js 出現以前，Javascript 通常作為網頁的用戶端語言使用，以 Javascript 寫出的程式常在用戶的瀏覽器上執行。Node.js 的出現使 Javascript 也能用於伺服器端。Node.js 含有一系列內置模組，使得程式可以作為獨立伺服器執行。並讓前端開發者可以更容易的處理後端伺服器的系統。

#### Freeird.js:

這是一套 Node.js 開源的物聯網管理平台套件，它可以搭建一異質網路，像是 BLE、MQTT、Zigbee、CoAP。並可以輕鬆的推送資料至許多平台，像是 PC、Raspberry Pi、Beaglebone 或者是其他嵌入式設備。這個套件也提供了一致的介面讓開發者可以同時操作多項不同協定的設備，像是 IoT/WoT 的基礎設施。藉著 Freebird.js 可以更簡單的將雲端、前端及機器結合在一起。他有以下幾種特色：

- 跨協定:像是 BLE、MQTT、Zigbee、CoAP
- Smart Object (IPSO)裡分層的資料模組
- 本地的網路管理中心及應用閘道器
- 基於 Node.js 使它可以簡易的結合機器的應用及其他框架或服務，像是 HTTP Server、Express.js、React.js、Angular.js
- 處理了 IoT 最底層的部分和幫助前端開發者建置出更迷人的圖形介面

### LoRa module :

目的在撰寫一模組介於 LoRa Gateway 及 Freebird 之間負責將收到的數據格式化為 FreeBird 的資料型態後再發送至 FreeBird。

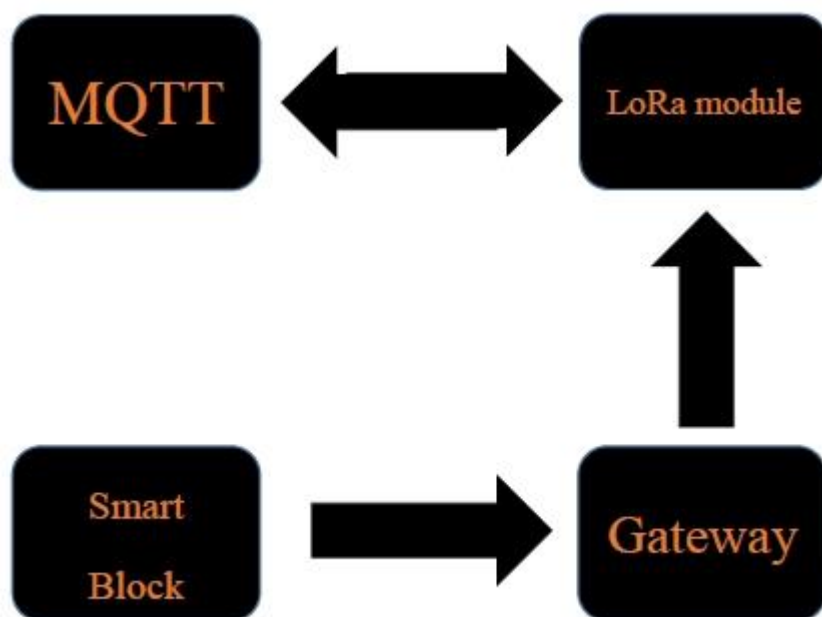


圖8LoRa module架構圖(一)

伺服器

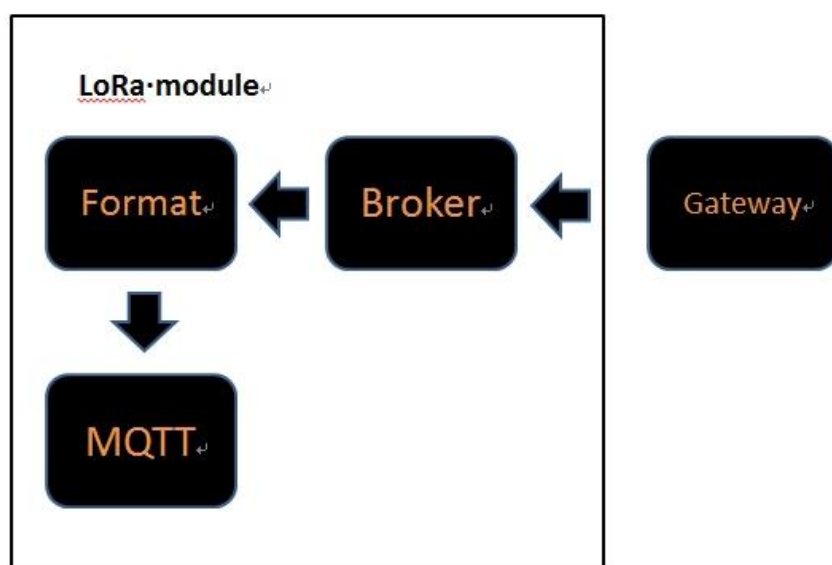


圖9LoRa module架構圖(二)

### Lora Pico無線型閘道器MQTT PUB/SUB流程:

WEB一開始會訂閱gad、dev和cmd下的主題，server的內部mqtt-node則是訂閱cmd主題，當web端的mqtt-node準備好，會發布訊息至cmd/getAllGadAndDev來向server要求所有連接過裝置的資料。每當web端收到一個新的裝置便會訂閱devAddr/gadId/sensorValue，當裝置的數值發生改變時，會發布新的數據至devAddr/gadId/sensorValue，WEB收到數據後則會更新裝置列表。

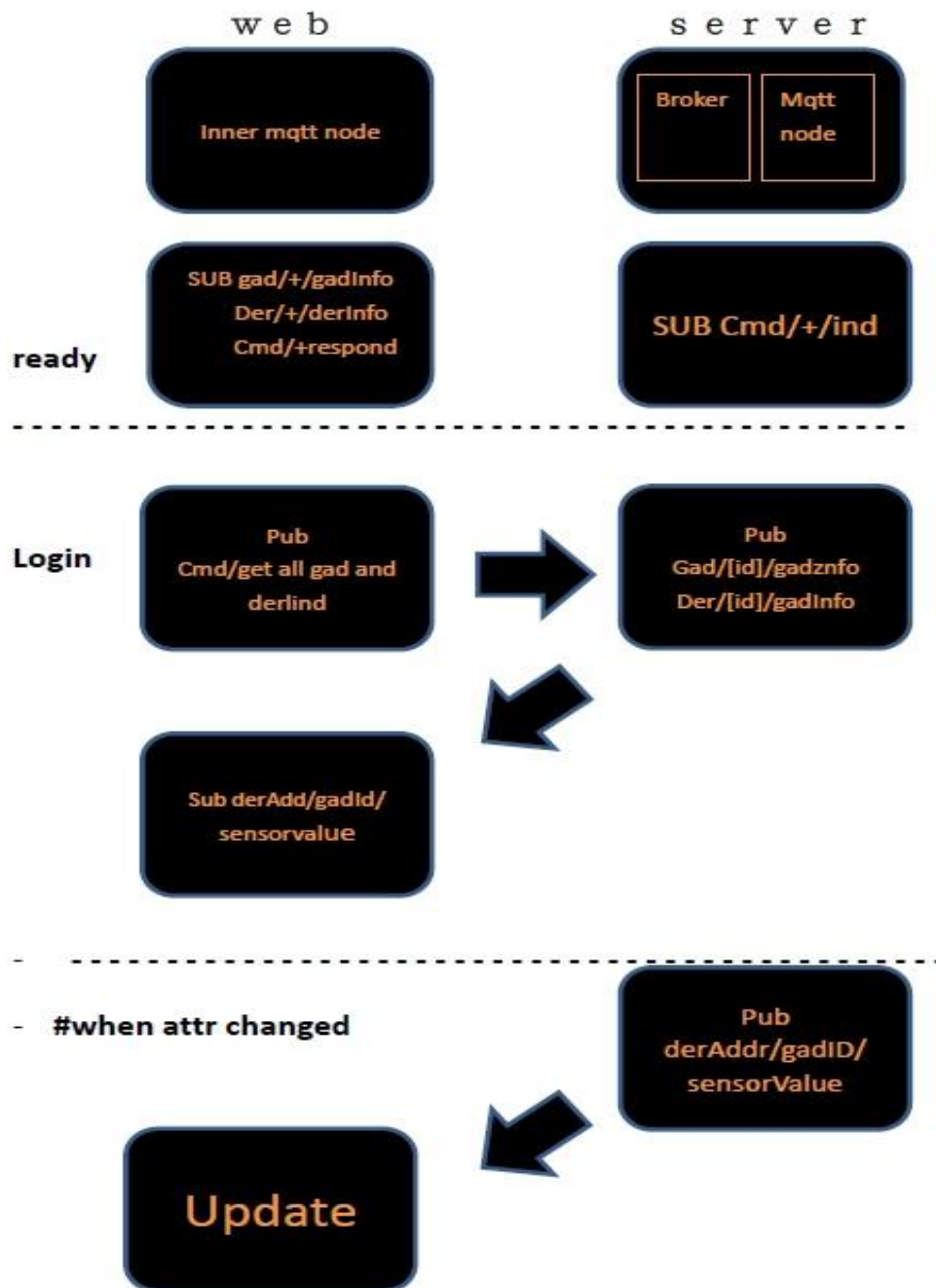


圖10 MQTT PUB/SUB流程

### Server系統流程圖：

伺服器會依照express→webpack→Lora server→dB→FreeBird的順序啟動，每當裝置上線，則會關注裝置上的感測器，當感測器的數值改變會將其寫入資料庫並發布至derAddr/gadid/sensorValue

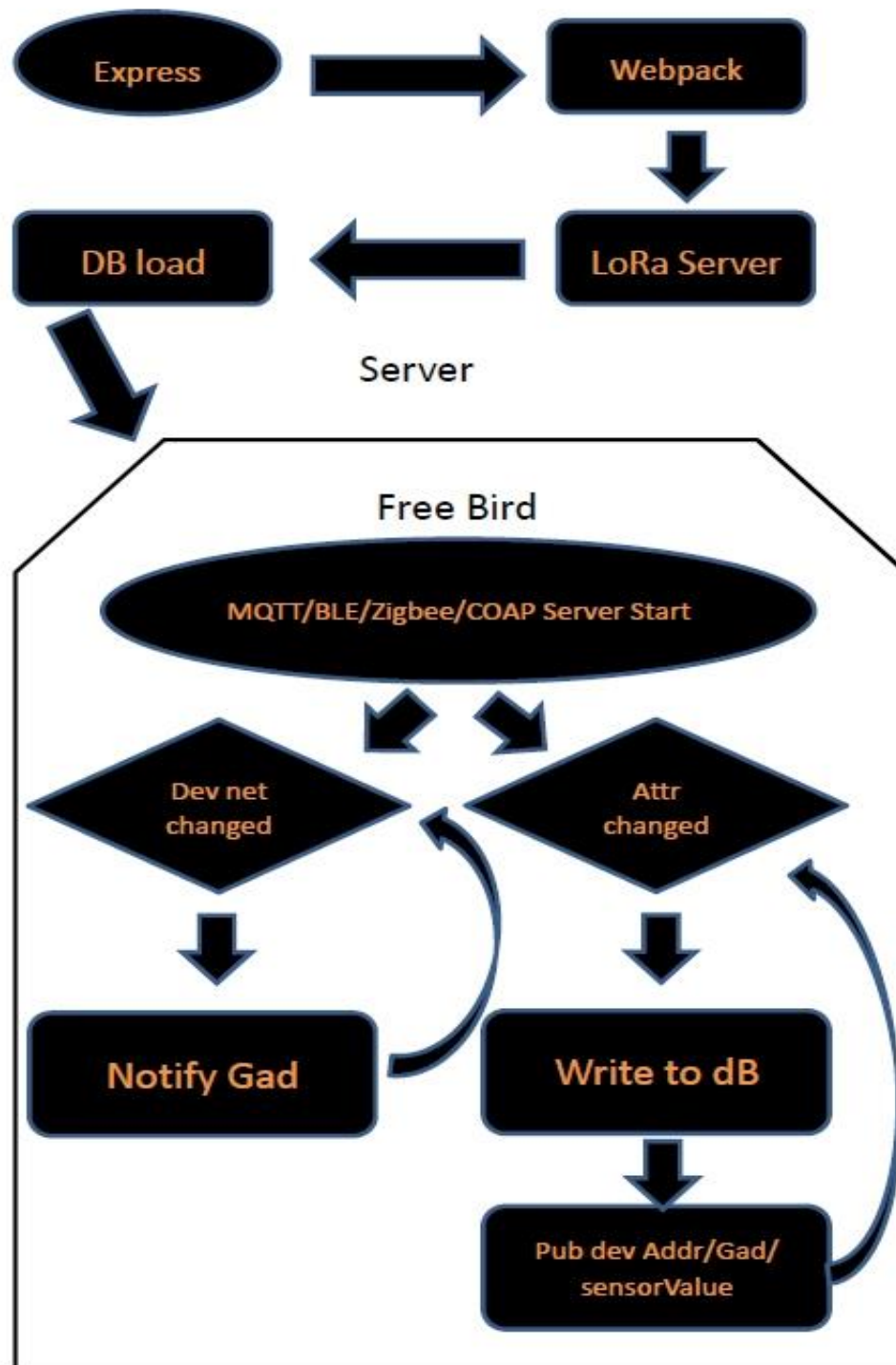


圖11 Server系統流程圖

## B. 硬體方面：

本專題採用正文科技(Gemtek)的室內 Lora Pico 無線型閘道器如圖 12 以及 sivann-module(天氣觀測站模組)來完成。



圖 12Lora Pico 無線型閘道器

### (一) Lora Pico 無線型閘道器及 SmartBlock

它遵守 LoRaWan 1.0.2 規範，支援 16 通道同時傳送，並支援多種網路連接方式。

Lora 感測器部分則採用群登科技(Acsip)的 Smart Block，它是由電源模組、Lora 模組、Arduino 模組組成的 Lora 感測器，本次專題將會使用 Smart Block 上原本就有的濕度及溫度感測器來偵測數值。



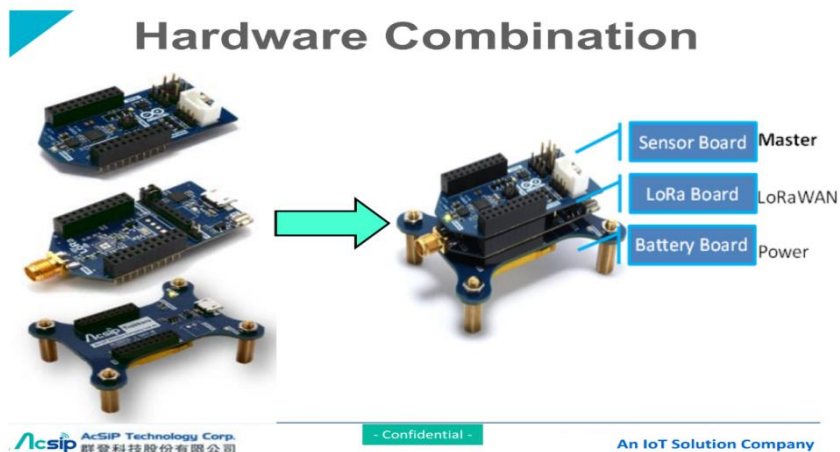


圖 13 Smart Block 架構圖

如果要使 LoRa 設備連至閘道器必須先至路由器頁面設置該裝置的憑證，如要用 OTAA 方式連接需要填入 AppEUI、DevEui、DevAddr，ABP 方式的話則需填入 NwkSkey、AppSkey。且為了使閘道器能正確的連上伺服器的 Mqtt Broker 還需要設定相關的資料。

InDoor

- Status
- System
- GloT
- LoraWan
  - OTAA Status
  - OTAA
  - ABP
  - Network Server
- Network
- Logout

## Network Server

The description for the Network Server.

Protocol:	MQTTS
Hostname:	192.168.88.134
Username:	admin
Password:	admin
Publish topic:	GIOT-GW/DL/
Subscribe topic:	GIOT-GW/UL/
Downlink ACK:	GIOT-GW/DL-report/
Port:	1883

APPLY

Powered by LuCI Trunk (git-71cd45f) / OpenWrt Barrier Breaker unknown

圖 14 Gateway 相關設定之一

InDoor

Status

System

GloT

LoraWan

OTAA Status

**OTAA**

ABP

Network Server

Network

Logout

OTAA

The description for the OTAA.

1 / 1

Group Index	AppEUI Start	App Counts	DevEUI Start	DevEUI Counts	DevAddr Start	DevAddr Counts	AppKey	Aging Out Time (Minutes)
0	00000000 00000000	1	9c65f9ff fe5b8d9c	1	0a12 bc34	1	11223344 55667788 99aabbcc ddeeff01	60

DELETE

ADD

11223344

55667788

99aabbcc

ddeeff01

EDIT

Powered by LuCI Trunk (git-71cd45f) / OpenWrt Barrier Breaker unknown

圖 15 Gateway OTAA 認證設定

InDoor

Status

System

GloT

LoraWan

OTAA Status

OTAA

**ABP**

Network Server

Network

Logout

ABP

The description for the ABP.

1 / 1

INDIVIDUAL

NETID GROUP

DevAddr	NwkSKey	AppSKey
0a12bc34	112233445566778899aa bbccddeeff00	112233445566778899aa bbccddeeff00

DELETE

ADD

0a12bc34

112233445566778899aa

bbccddeeff00

112233445566778899aa

bbccddeeff00

EDIT

Powered by LuCI Trunk (git-71cd45f) / OpenWrt Barrier Breaker unknown

圖 16 Gateway ABP 金鑰設定

## (二)sivann-module

它的無線感測模組由三種電路模組堆疊而成如圖 17，包括上層感測模組、中層 BLE 無線模組，以及底層電源模組，其中包含光度、大氣壓力、溫濕度以及聲音等感測器，模組上的感測值皆可透過低功耗藍牙(BLE)以無線方式傳出。

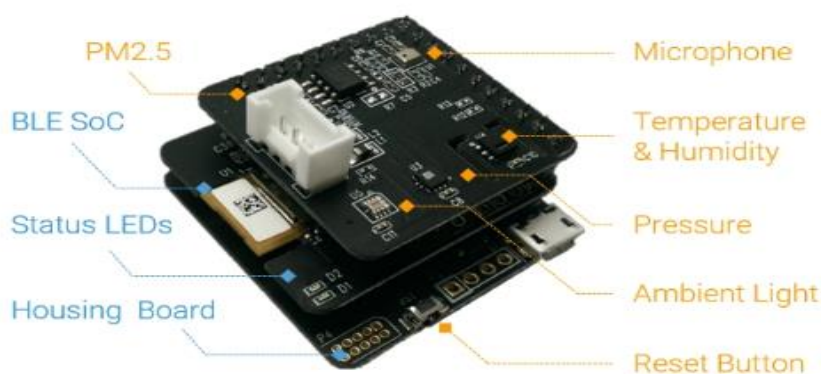


圖 17 Sivann weather station module

## C.無線存取技術

### (一)CoAP:

CoAP 是受限制的應用協議(Constrained Application Protocol)，它考慮到資源限制、低處理能力以及低功耗限制，而重新設計了 HTTP 的部分功能而可以適應設備的條件。另外，為了使 CoAP 適應物聯網和 M2M 應用，CoAP 協議改進了一些機制之外也增加了一些功能。圖 18 顯示了 HTTP 和 CoAP 的協議。CoAP 和 HTTP 在傳輸層有明顯的區別。HTTP 協議的傳輸系統訊息輸出層採用了 TCP 協議，而 CoAP 協議的傳輸層使用 UDP 協議，消耗的資源降低許多。

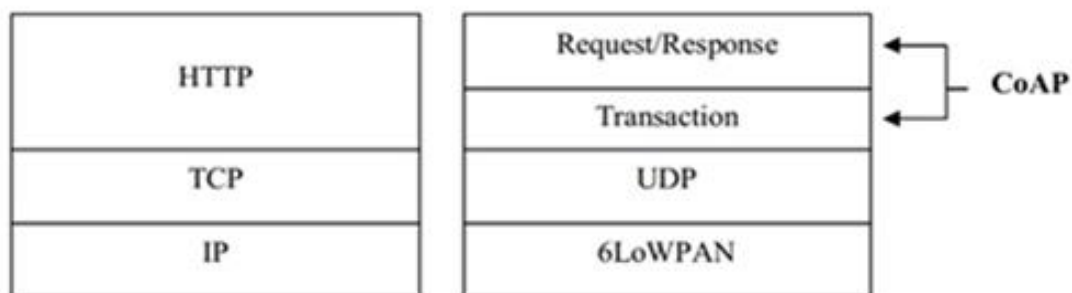


圖 18 CoAP 協議

## (二)MQTT:

MQTT 的全名為 Message Queuing Telemetry Transport，為 IBM 和 Eurotech 共同制定出來的協議，它是為了物聯網而設計的協議，並且它是透過 publish/subscribe 的方式來做訊息傳送。由於是為了物聯網而設計的協議，因此它所需要的網路頻寬是很低的，而所需要的硬體資源也很低。

## (三)LoRa:

LoRa 是 LPWAN 通信技術中的一種，是美國 Semtech 公司採用和推廣的一種基於擴頻技術的超遠距離無線傳輸方案。而它的特點有傳輸距離遠以及工作的功耗低，終端設備可能是各種設備，比如溫溼度感測器、氣體感測器等。這些節點通過 LoRa 無線通信首先與 LoRa 網關連接，再通過 3G 網絡或者乙太網絡，連接到網絡伺服器中，而 LoRa 的架構圖如下：

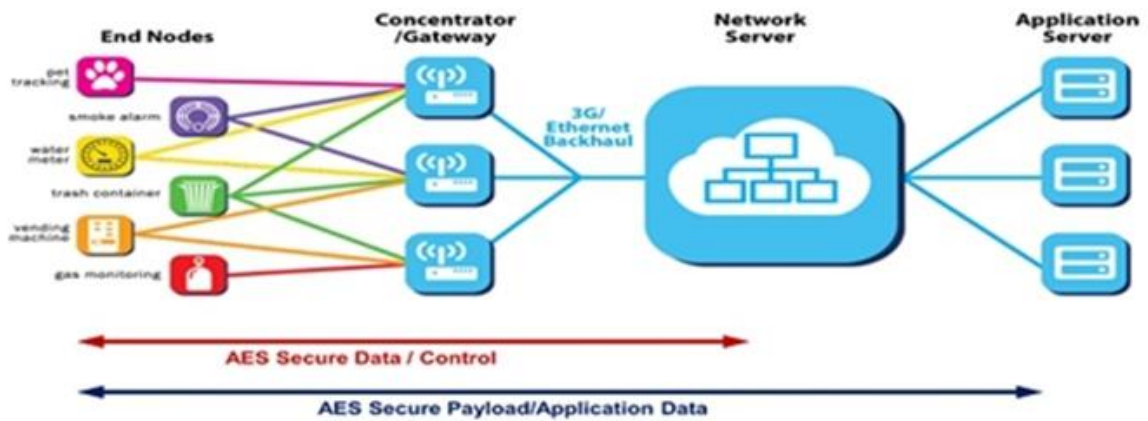


圖 19 LoRa 架構圖

## 四、研究成果

### 4.1 假想情境－智慧工廠

我們以智慧工廠作為假想情境，此工廠設有多種 LoRa 感測器，例如溫度、濕度以及氣體感測器，收集到的數據會透過 LoRa 來傳至伺服器端，使用者可以透過網頁來全面監控工廠狀況，概念圖如下：

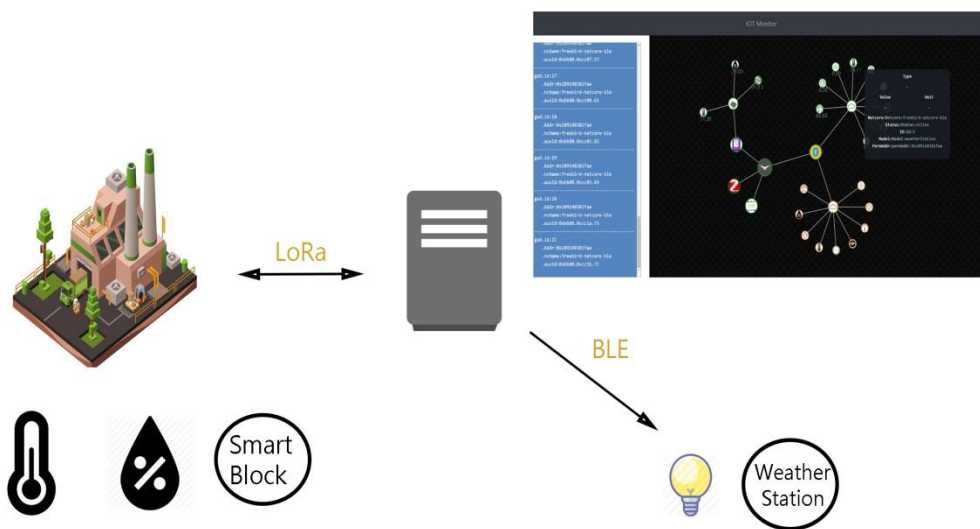


圖 20 智慧工廠概念圖

本次實驗使用一塊 sivvan 的 weather station 模組及 LoRa Smart Block。

### 4.2 server 端

當程式執行時會依序啟動

Express 伺服器 → webpack 打包 → 自行撰寫的 LoRa 模組 → Database → Freebird.js 並顯示相關訊息

```
webpack built d2d9a4328abfc444b84d in 26280ms
[fwdm]: Version: webpack 4.6.0
Time: 26280ms
Built at: 2018-05-29 23:45:39
    Asset      Size  Chunks             Chunk Names
  main.js    2.22 MiB       0  [emitted]  main
  index.html  7.25 KiB       0  [emitted]
Entrypoint main = main.js
Child html-webpack-plugin for "index.html":
    Asset      Size  Chunks             Chunk Names
  index.html  1.43 MiB       0
Entrypoint undefined = index.html
[fwdm]: Compiled successfully.
```

圖 21 Webpack 打包訊息

當 Freebird 啟動後會印出啟動訊息(如圖 22)，

```
[ ready ]  
[Server Started]
```

圖 22 Freebird 啟動訊息

當裝置狀態改變時(上線或離線)會印出該裝置的編號、實體位置、連接方式及裝置狀態(如圖 23)。

```
[ devNetChanged ] dev.id:5 |dev.Addr:0x209148381fae |dev.ncName:freebird-netcore-ble |dev.status:online
```

圖 23 裝置狀態改變訊息

當裝置上的感測器數值改變時也會印出該感測器的編號及新的數值(如圖 24)。

```
[ gadAttrChanged ] gad.id:14 |gad.data:219  
[ gadAttrChanged ] gad.id:17 |gad.data:55.40899658203125  
[ gadAttrChanged ] gad.id:20 |gad.data:67.97282409667969  
[ gadAttrChanged ] gad.id:20 |gad.data:70.57881164550781  
[ gadAttrChanged ] gad.id:14 |gad.data:221  
[ gadAttrChanged ] gad.id:20 |gad.data:64.4468994140625  
[ gadAttrChanged ] gad.id:20 |gad.data:68.73686218261719  
[ gadAttrChanged ] gad.id:14 |gad.data:197  
[ gadAttrChanged ] gad.id:20 |gad.data:68.57144927978516  
[ gadAttrChanged ] gad.id:20 |gad.data:65.99978637695312  
[ gadAttrChanged ] gad.id:17 |gad.data:55.47003173828125  
[ gadAttrChanged ] gad.id:20 |gad.data:69.12979125976562  
[ gadAttrChanged ] gad.id:14 |gad.data:195  
[ gadAttrChanged ] gad.id:20 |gad.data:69.9757308959961  
[ gadAttrChanged ] gad.id:17 |gad.data:55.53106689453125  
[ gadAttrChanged ] gad.id:20 |gad.data:60.67974090576172  
[ gadAttrChanged ] gad.id:20 |gad.data:71.43867492675781  
[ gadAttrChanged ] gad.id:20 |gad.data:69.26478576660156  
[ gadAttrChanged ] gad.id:17 |gad.data:55.59210205078125
```

圖 24 感測器的數值改變訊息

## 4.2 web 端

當伺服器啟動後就可於瀏覽器輸入 <http://localhost:8000> 進入監測頁面  
畫面左邊會有一個訊息輸出欄，這邊會顯示系統的相關訊息，像是內部  
Mqtt-node 的狀態及收到的裝置資訊(如圖 25)



```
Mqtt-Node ready!
IndexDB load success!
Mqtt-Node login!
getAllGadAndDev by Mqtt-Node!
-----
dev.id:1
  .Addr:00:11:22:aa:bb:cc/LoRaGateway1
  .ncName:my_iot_mqtt
-----
dev.id:4
  .Addr:0x20c38ff1c09d
  .ncName:freebird-netcore-ble
-----
dev.id:5
  .Addr:0x209148381fae
  .ncName:freebird-netcore-ble
-----
gad.id:1
  .Addr:00:11:22:aa:bb:cc/LoRaGateway1
  .ncName:my_iot_mqtt
  .auxId:temperature/0
-----
gad.id:2
  .Addr:00:11:22:aa:bb:cc/LoRaGateway1
  .ncName:my_iot_mqtt
  .auxId:humidity/0
-----
gad.id:3
  .Addr:00:11:22:aa:bb:cc/LoRaGateway1
```

圖 25 系統訊息輸出



畫面右邊則會顯示以 D3 的力向圖呈現的裝置列表(如圖 26)

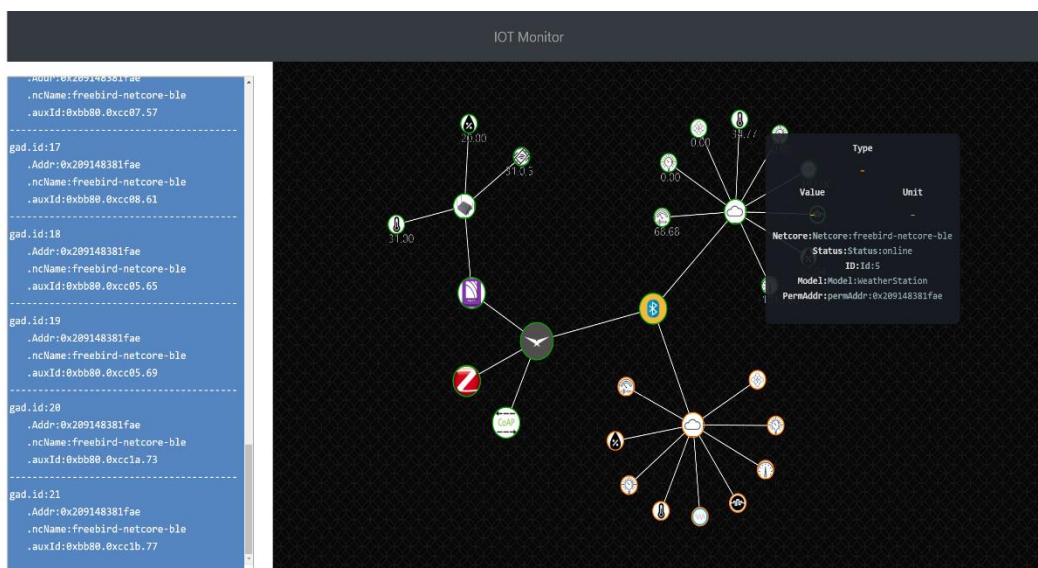


圖 26 裝置的力向圖

當鼠標移至裝置圖示上時會顯示該裝置的屬性，像是感測器種類、感測器數值、感測器單位、連接方式、狀態、編號、模組及實體位址(如圖 27)，



圖 27 裝置的資訊

點擊裝置圖示會彈出圖表，圖表分為三種模式，分別是可以顯示目前該裝置至網頁連上伺服器後收到的數值圖表(如圖 28)，此模式會將收集到的數據以線條表示。



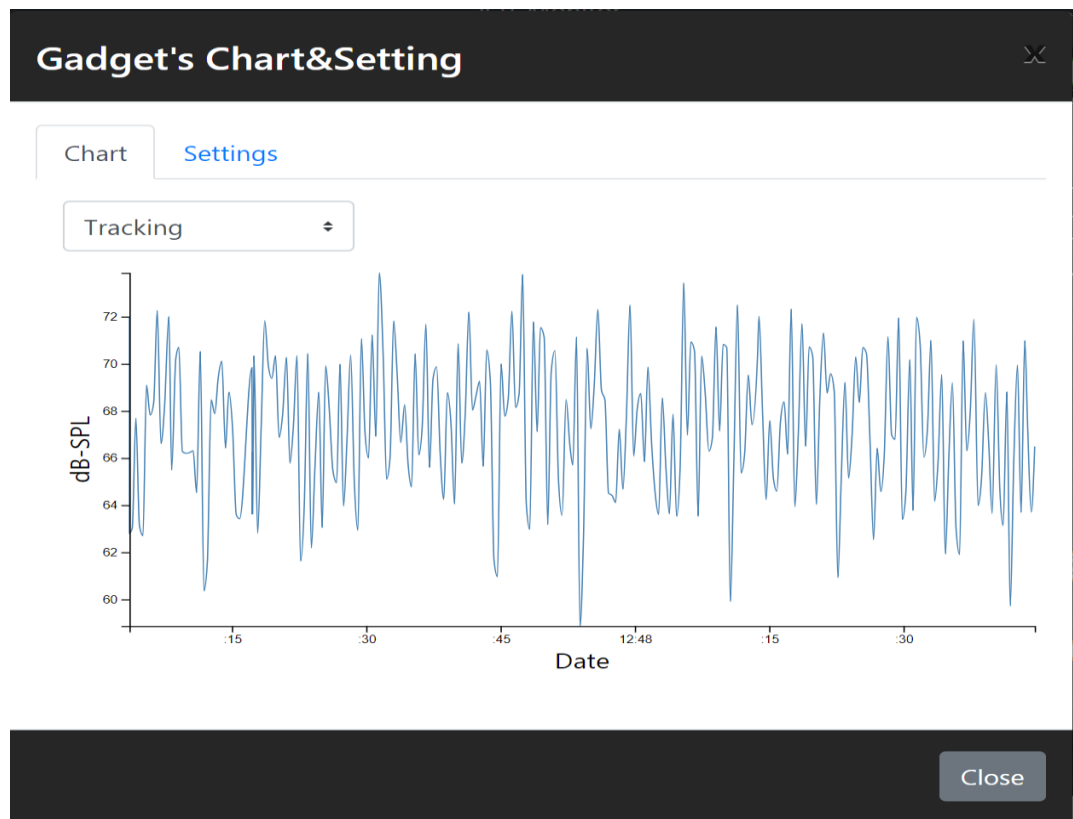


圖 28 裝置圖表 持續更新

特定日期的數據(如圖 29)，此模式會將特定日期的所有數據以圓點的方式顯示出來，並當滑鼠移至圓點上方時會顯示出該筆資料的時間及數值，且可以滾輪進行縮放。

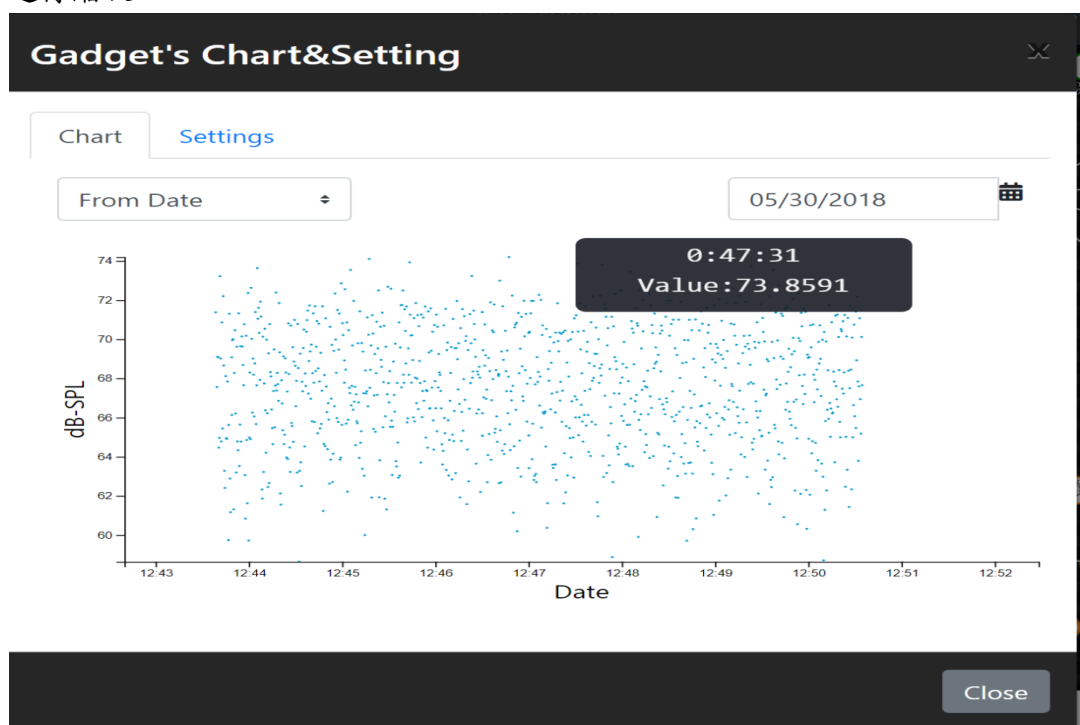


圖 29 裝置圖表 單日

特定日期範圍內的數據，此模式會顯示特定日期範圍內的所有數據，如圖 30 就是 5/27 至 5/28 中間蒐集到的數據，一樣可以滾輪進行縮放及顯示該筆資料。

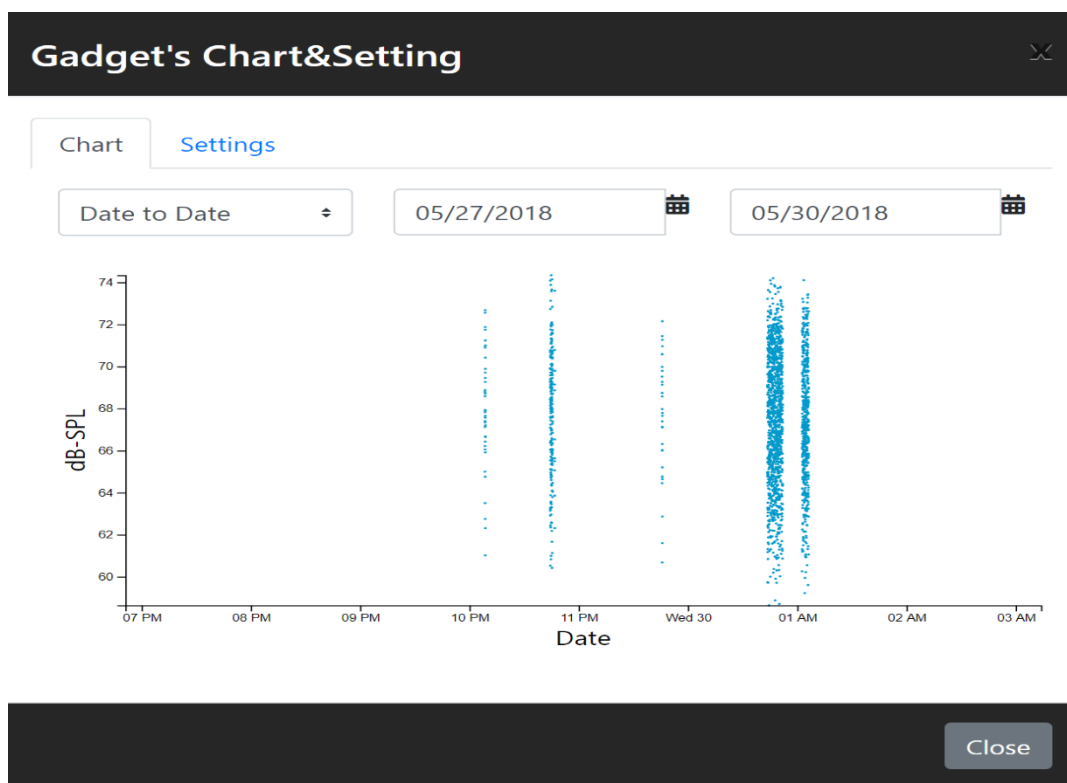


圖 30 裝置圖表 特定範圍日期

此外還可以對該感測器設定最大值及最小值(如圖 31)

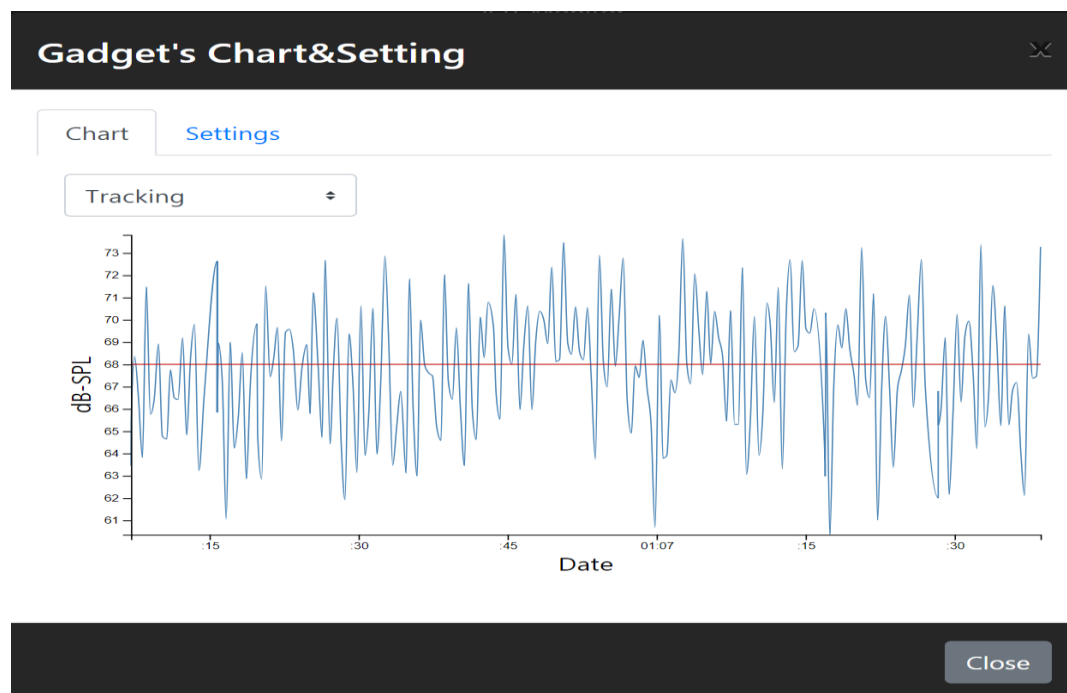


圖 31 設定上限值後的圖表

當數值超過設定的範圍時裝置的圖示會變紅(如圖 32)。



圖 32 數值超過範圍後的圖示

## 五、參考文獻

- [1] Node.JS 官方網站 <https://nodejs.org/en/>
- [2] Freebird.js 框架介紹 <https://www.npmjs.com/package/freebird#Overview>
- [3] D3.js 官方網站 <https://d3js.org/>
- [4] OMA 官方網站 <https://www.omaspecworks.org/about/>
- [5] JQuery <http://jquery.com/>
- [6] sivann 藍芽模組說明  
<https://github.com/sivann-tw/sivann-modules/blob/master/ble/WeatherStation.md>