

### 三.製作理論探討

#### 1.系統架構

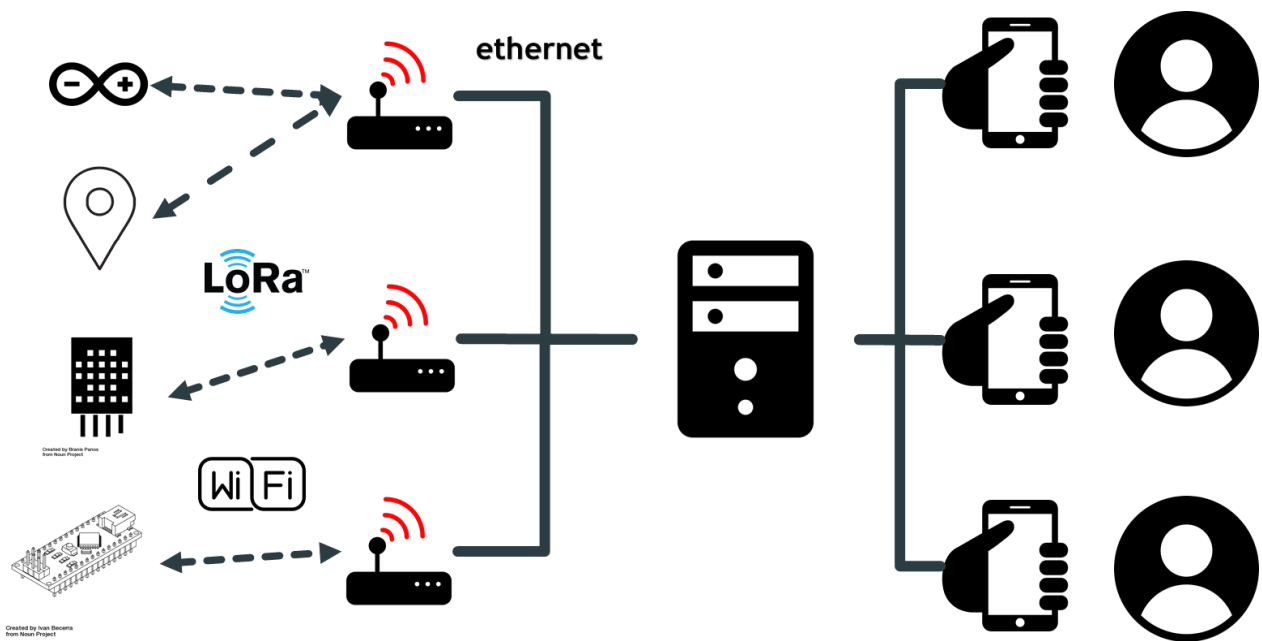


圖 系統架構

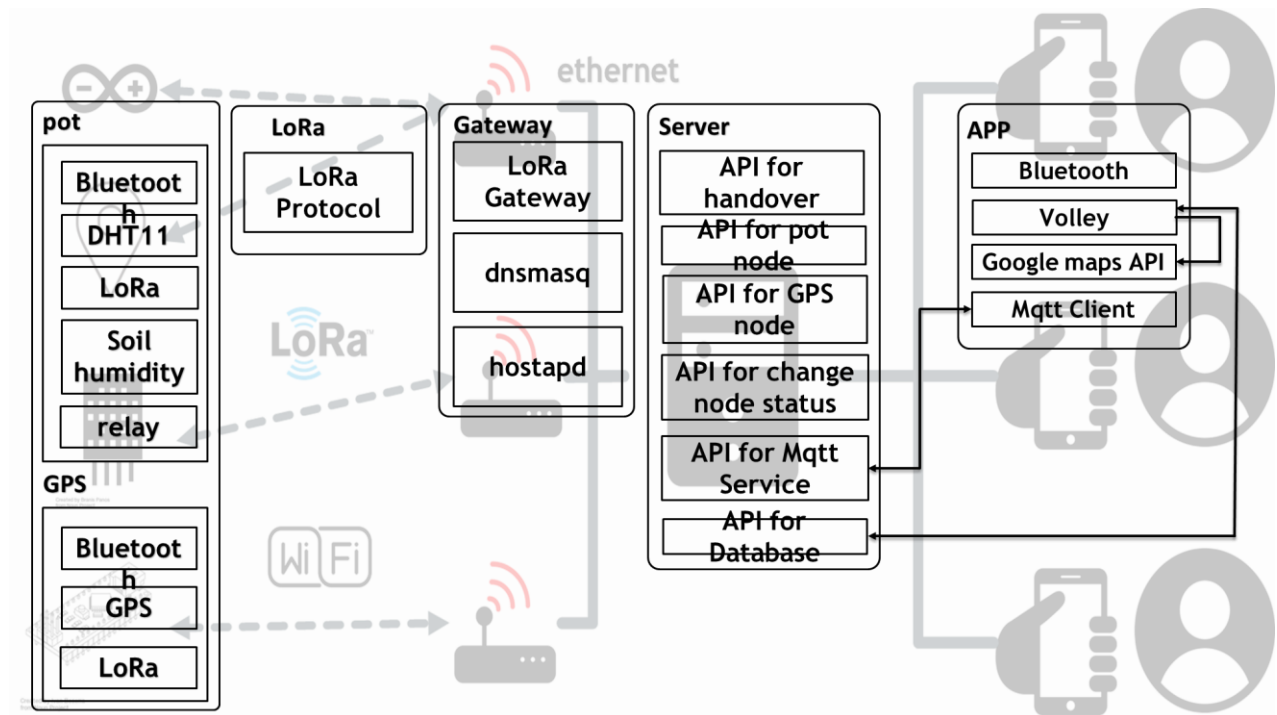


圖 系統架構細節

## 2.LoRaWan 封包格式

封包格式:

<b>cjp (3 bytes)</b>	<b>type (1 byte)</b>	<b>destination (6 bytes)</b>	<b>source (6 bytes)</b>	<b>MessageID (1 byte)</b>	<b>length (1 byte)</b>	<b>payload</b>
--------------------------	--------------------------	----------------------------------	-----------------------------	-------------------------------	----------------------------	----------------

圖 LoRaWan 封包格式

cjp:

因為 LoRa 並無一個統一的傳輸協定所以任何人都可使用,最前面 cjp 三個字元只是為了識別是否是我可以讀懂的封包,過濾掉那些別人送出來但是我讀不懂的封包

type:

T(transfer), D(data), A(ack), S(search), I(Inform)

destination:

每個裝置都有一個唯一的 SID,此封包的 destination 欄位即是這個封包的目的地 SID,000000 為廣播位址

sid:

發送端的 SID

messageID:

此封包的識別碼(0~255)

length:

後面 payload 的長度

payload:

訊息

### 3.LoRaWan 各類封包的運作

#### (1)Data

所有 LoRa node 要將感測到的資料(例如:溫濕度,GPS 經緯度) 裝置狀態(例如:某個開關被打開…等),只要是要需要轉送到 internet 上的訊息,就必須使用這類封包

cjp	D	Lgw001	pot001	0~255	0~255	payload
-----	---	--------	--------	-------	-------	---------

#### (2)Transfer

一般的資料封包, 大部分用於 internet 上的訊息透過 gateway 轉送到 LoRaWan 內

cjp	T	pot001	Lgw001	0~255	0~255	payload
-----	---	--------	--------	-------	-------	---------

#### (3)Ack

用來確保封包順利抵達目的地,只要接收端順利接收到一個正確的封包就發一個 ack 給發送端告知此封包傳送成功,ask 封包的 messageID 欄位會填入接收端要回給發送端的封包的識別碼,傳送端若在送出封包後 2 秒內沒收到 ack 傳送端就會判定這個封包傳送失敗

node po001 送出一個封包識別碼為 5 的封包給 Lgw001:

cjp	D	Lgw001	pot001	5	0~255	payload
-----	---	--------	--------	---	-------	---------

Lgw001 回一個 ack 給 pot001 告知送給 Lgw001 的編號 5 封包成功送達:

cjp	A	pot001	Lgw001	5	3	ack
-----	---	--------	--------	---	---	-----

#### (4)Search

Node 目前剛開機尚未連線或原本連線的 gateway 消失,就會廣播這類封包尋找周邊有哪些 gateway 可用,所有收到這類封包的 gateway 都必須回應發送端,此時發送端就會從眾多 gateway 回應的 ack 中依照 rssi 值,挑選訊號最強的那個 gateway 連線,此時就會產生 handover,通知 server 這個 node 要換 gateway,此時 server 就會修改 routine table

pot001 廣播 search 封包

cjp	S	000000	pot001	0~255	0~255	search
-----	---	--------	--------	-------	-------	--------

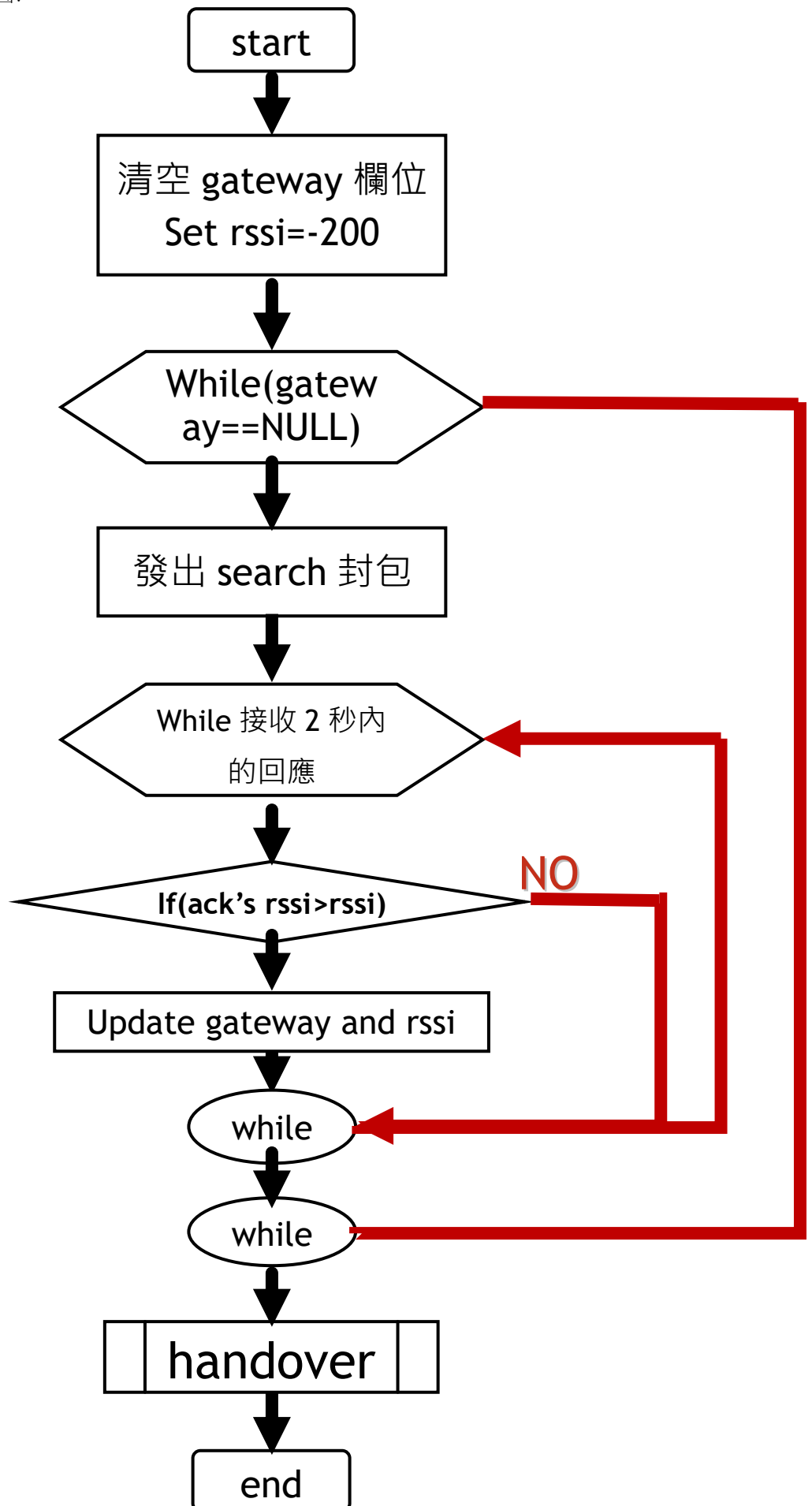
Lgw001 回應 pot001

cjp	A	pot001	Lgw001	0~255	0~255	ack
-----	---	--------	--------	-------	-------	-----

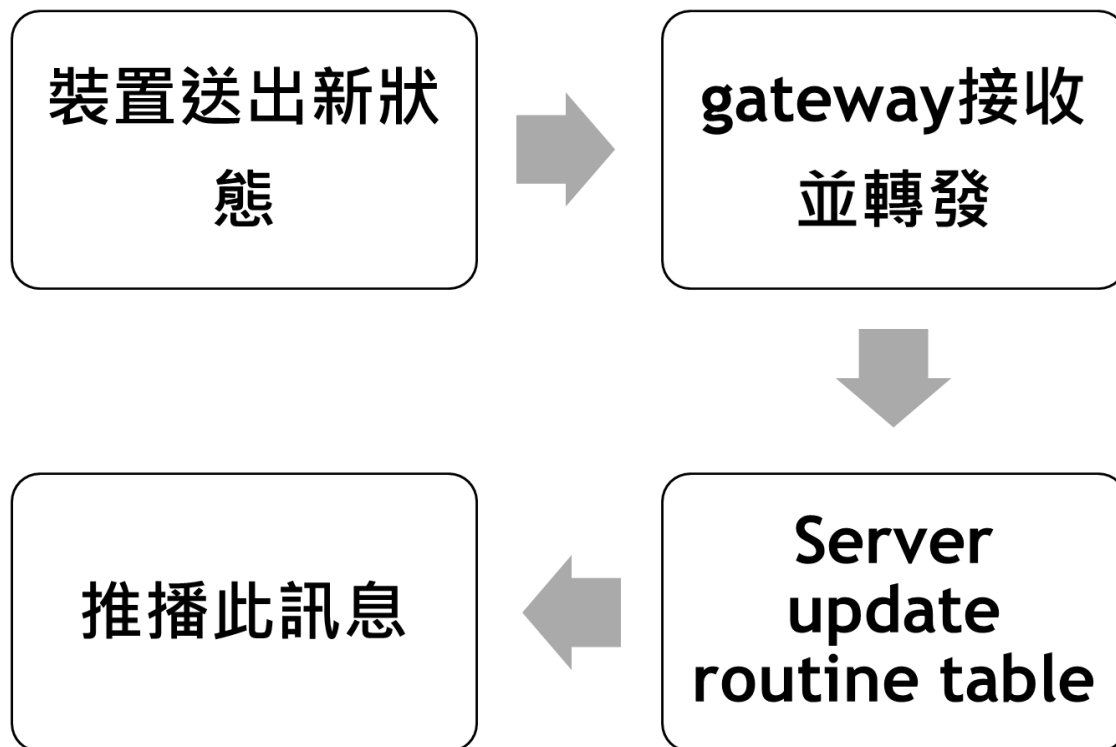
通知 server pot001 要與 Lgw001 連線

cjp	D	Lgw001	pot001	0~255	0~255	<a href="http://104.155.210.211:8080/registerLgw?user=user1&amp;sid=Ldht01&amp;gw=Lgw01">http://104.155.210.211:8080/registerLgw?user=user1&amp;sid=Ldht01&amp;gw=Lgw01</a>
-----	---	--------	--------	-------	-------	---

search gateway 流程圖:



Handover 流程:



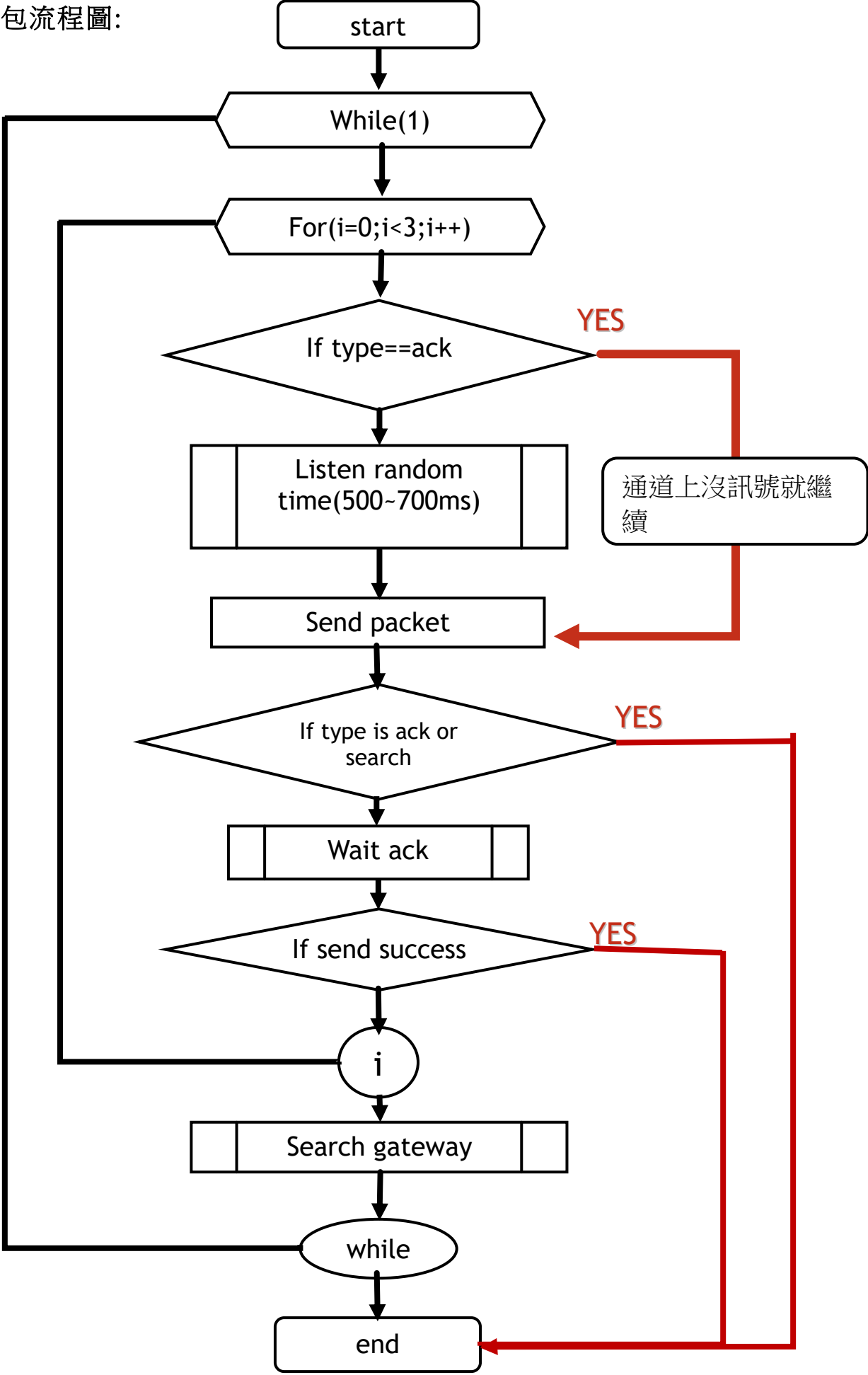
(5) Inform:

每個 LoRa gateway 每隔 3 分鐘就會廣播一個 Inform 封包給四周的 node,讓 node 知道它的周邊有哪些 gateway, 每個 node 都會 maintain 一個與此裝置連線的那台 gateway 的 rssi 值,node 每次只要收到來自與此裝置連線的 gateway 傳來的封包就會更新一次該數值,node 只要收到不是與他連線的 gateway 廣播出來的 Inform 封包,就會與該數值做比對,只要有大大於該數值的情況出現,就代表有一個 gateway 比目前連線的 gateway 訊號更好,此時就會 handover 換到跟訊號比較好的 gateway 連線

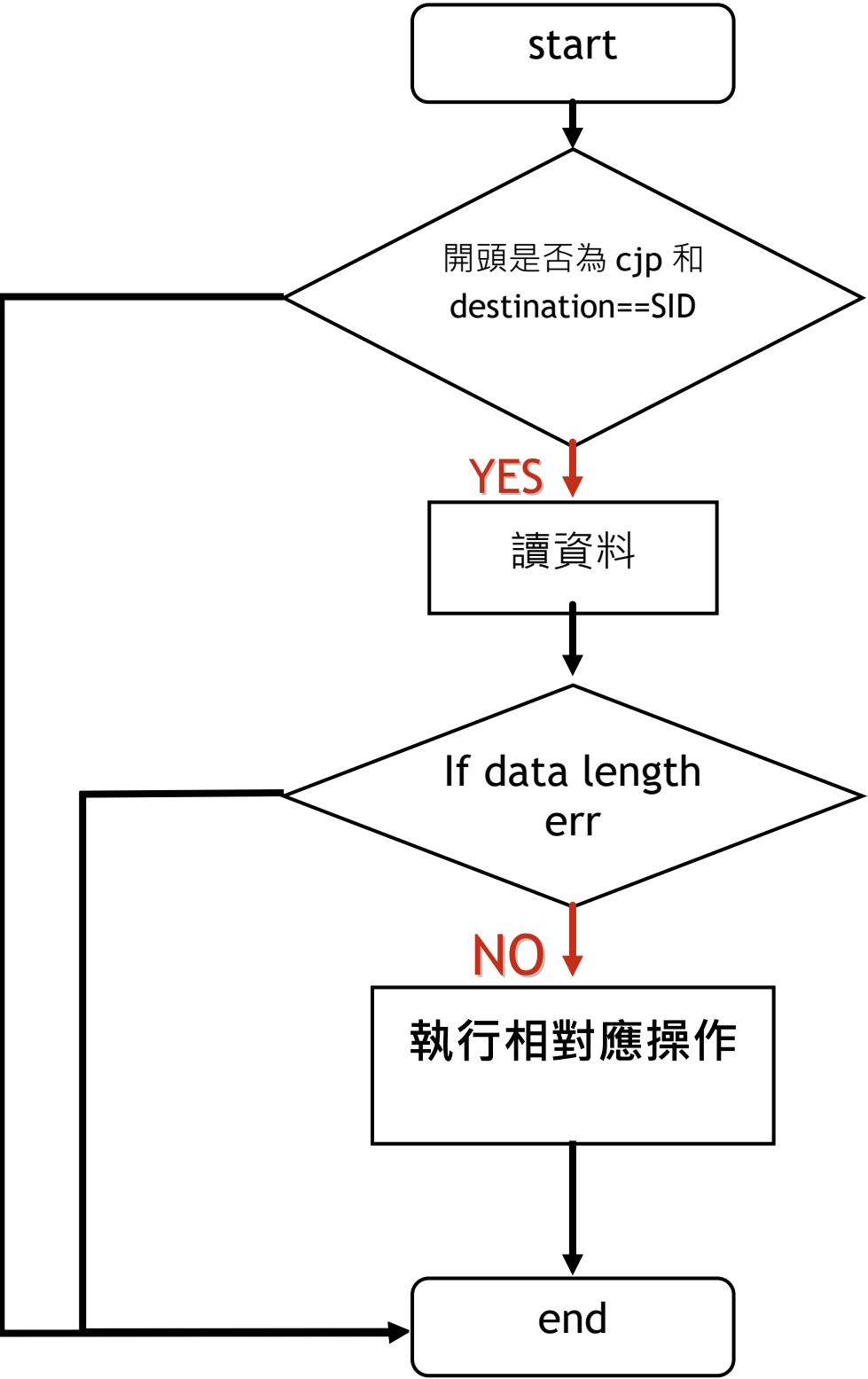
Lgw001 廣播 publish 封包

cjp	l	000000	Lgw001	0~255	0~255	Inform
-----	---	--------	--------	-------	-------	--------

4.送封包流程圖:



5.收封包流程:



## 6.LoRa gateway

LoRa gateway 是由一片 raspberry pi 加一片配有 Dragino Lora Shield 868MH 的 arduino 組成,raspberry pi 和 arduino 之間使用 USB 線連接

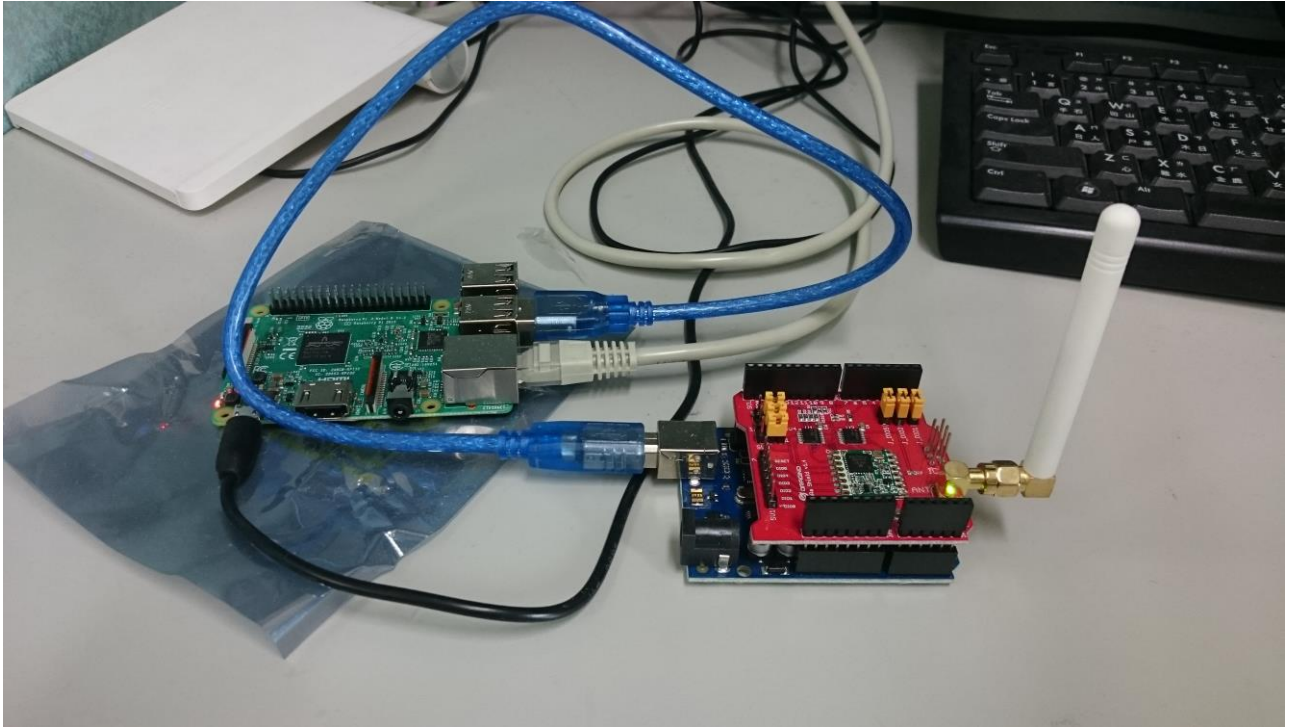


圖 LoRa gateway

當 arduino 一接上 raspberry pi 時,arduino 會傳送一段訊息(111+SID)告訴 raspberry pi 上的轉送服務這 arduino 的 SID 序號,此時這服務就會向 server 的 mqtt service 訂閱這個 topic,例如:arduino 傳 111Lgw001 過來,raspberry pi 上的轉送服務就會向 mqtt server 訂閱 Lgw001 topic,所以只要向 Lgw001 topic 發布訊息,這台 raspberry pi 就能收到

Raspberry pi 上的轉送服務由 javascript 撰寫,使用方式為

`$sudo node [程式位置] [USB 連接 port] [server IP] [mqtt 登入帳號]`

例如: `$sudo node /root/serial.js /dev/ttyACM0 104.155.210.211 test`

此程式會將收到的訊息的前 6 碼視為要轉送到 LoRaWan 中的目的地,所以如果向 Lgw001 topic 發布訊息 pot001hello,raspberry pi 收到此訊息時便會透過 USB 告知要 arduino 要送一封包給 SID 為 pot001 的 node ,訊息為 hello

cjp	T	pot001	Lgw001	0~255	5	hello
-----	---	--------	--------	-------	---	-------

gateway 保留 wifi 功能使 wifi 與 LoRa 並存,並且導入 docker 技術將每一個 service 獨立成一個一個的 container,可使用本專題所撰寫的 docker-compose.yml,即可快速地將所有功能部屬完畢



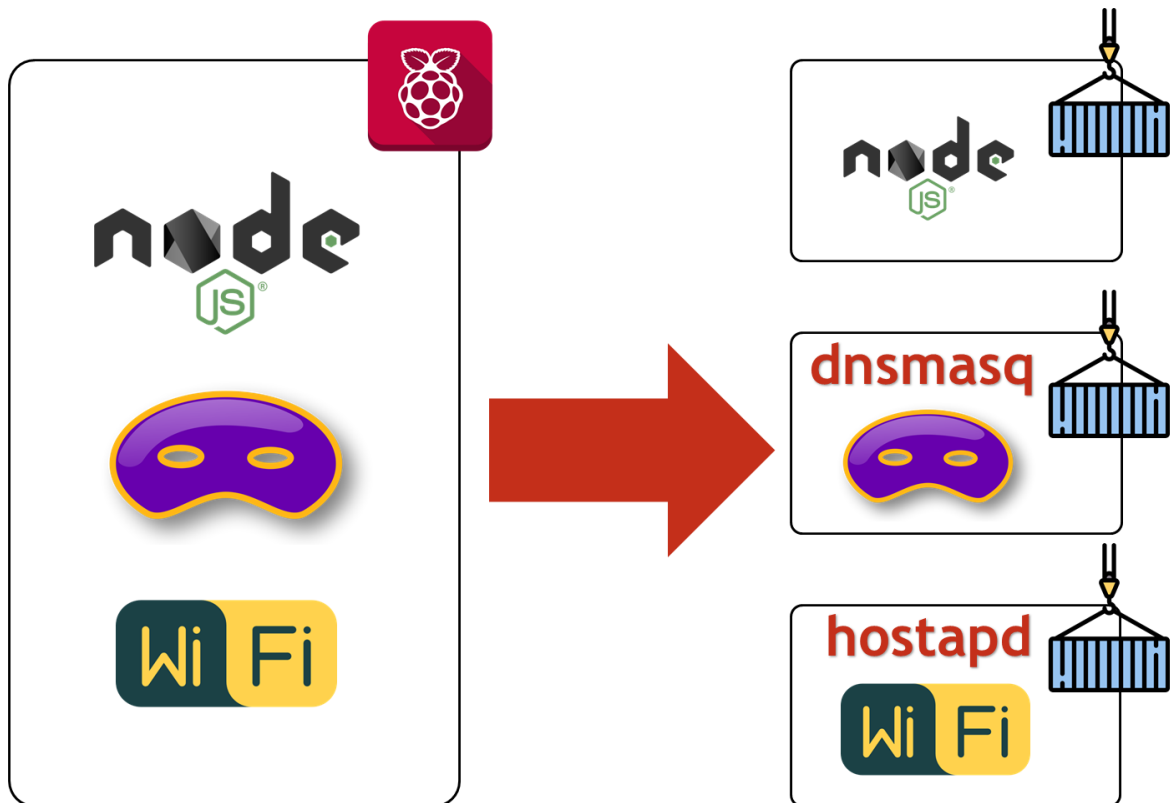


圖 LoRa gateway microservice 概念

```

pi@raspberrypi: ~
login as: pi
pi@192.168.31.128's password:

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Tue May  8 21:54:24 2018 from 192.168.31.106
pi@raspberrypi:~$ sudo docker ps
CONTAINER ID        IMAGE               COMMAND                  CREATED            STATUS              PORTS
d1aef19e971b        nodejs:0.1         "node /root/serial.j..." 43 hours ago       Up 43 hours
056eb38eb0ff        hostapd:0.1        "hostapd -dd /etc/ho..." 43 hours ago       Up 43 hours
f19ce288e902        dnsmasq:0.1        "dnsmasq --keep-in-f..." 43 hours ago       Up 43 hours
pi@raspberrypi:~$

```

圖 LoRa gateway microservice 運作

## 7.農業應用 node

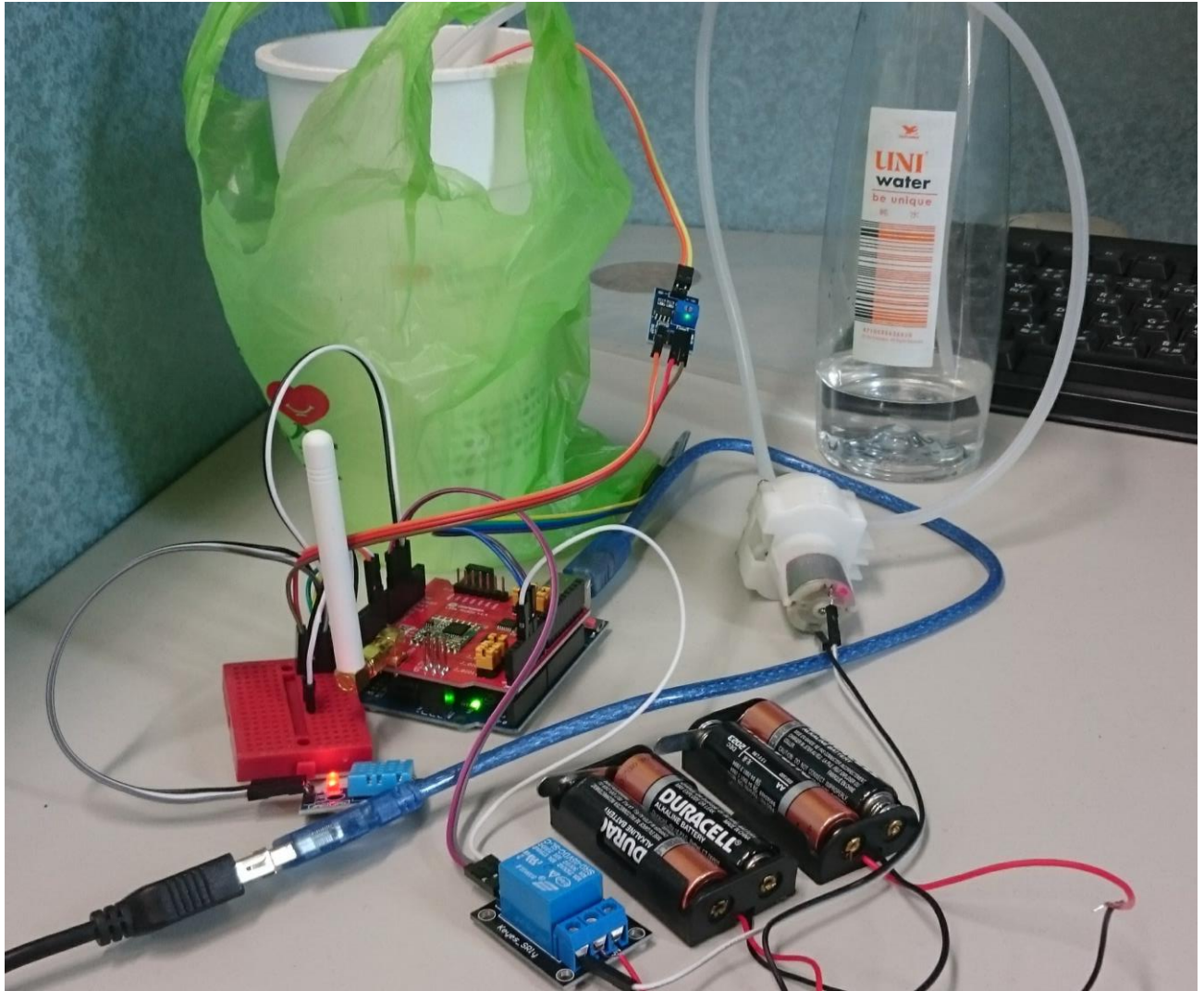


圖 農業應用 node

這類 node 有一片 arduino,一片 Dragino Lora Shield 868MH,一片藍芽模組(CH-06)加上空氣溫濕度感測模組(DHT-11),土壤溼度感測模組,及一個繼電器開關,外加 6 伏特的電源給抽水馬達使用

在使用前必須先使用藍芽與此裝置配對並透過藍芽告知此裝置的 user,設定字串為 (SET+USER+"user")例如這裝置的 user 為 user1,在使用前就必須先傳一段 SET+USERuser1 給此裝置,此時 arduino 就會將這些資料記錄至 arduino 上的 EEPROM,所以如果在 node 開機時發現 EEPROM 內沒 user 資料的話就會進入 busy wait 直到有使用者使用藍芽告知此裝置的 user 操作才會繼續

EEPROM 內資料擺放位置:

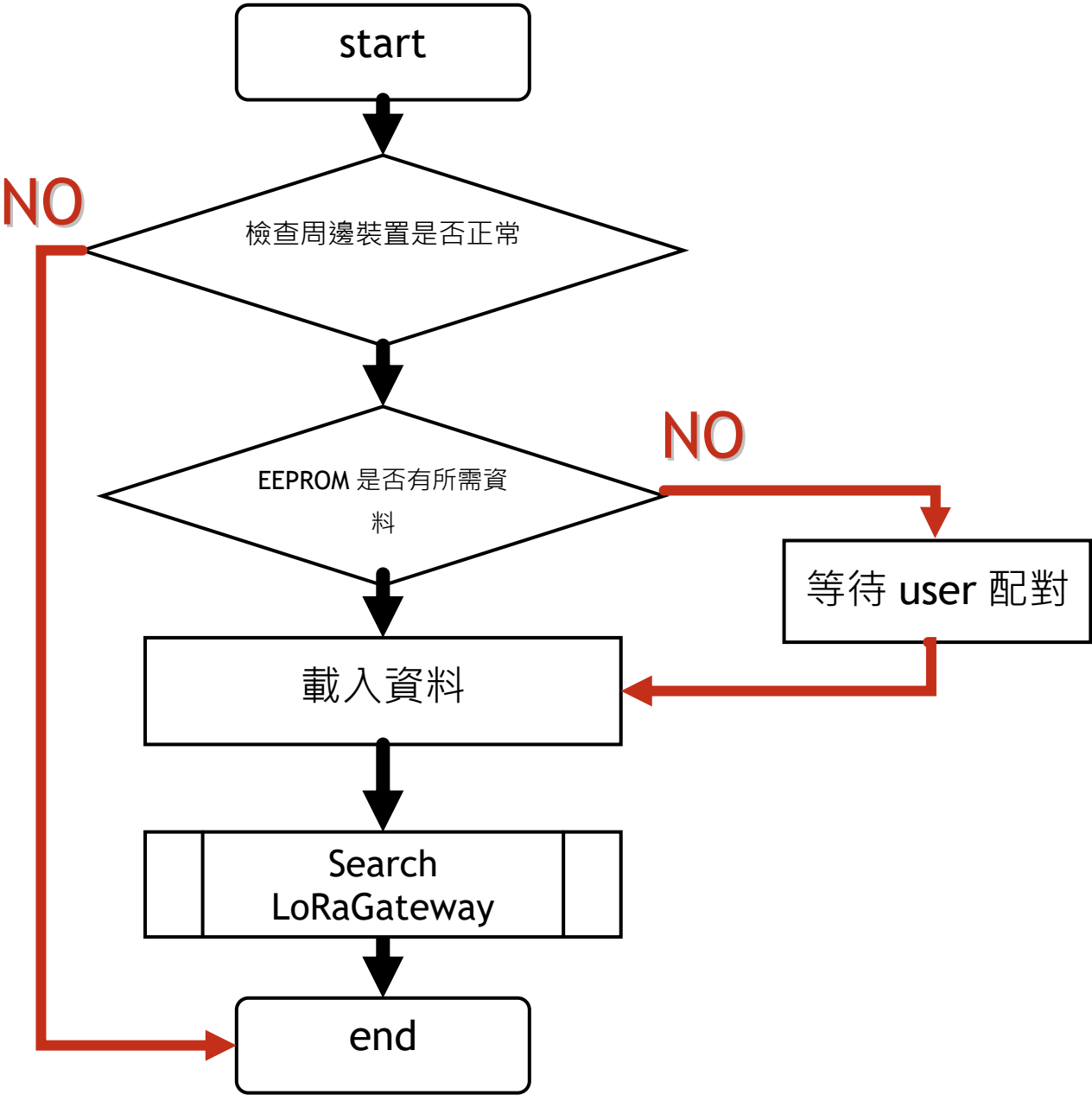
User:起始位置為 0

是否開啟自動澆水功能:位置 20

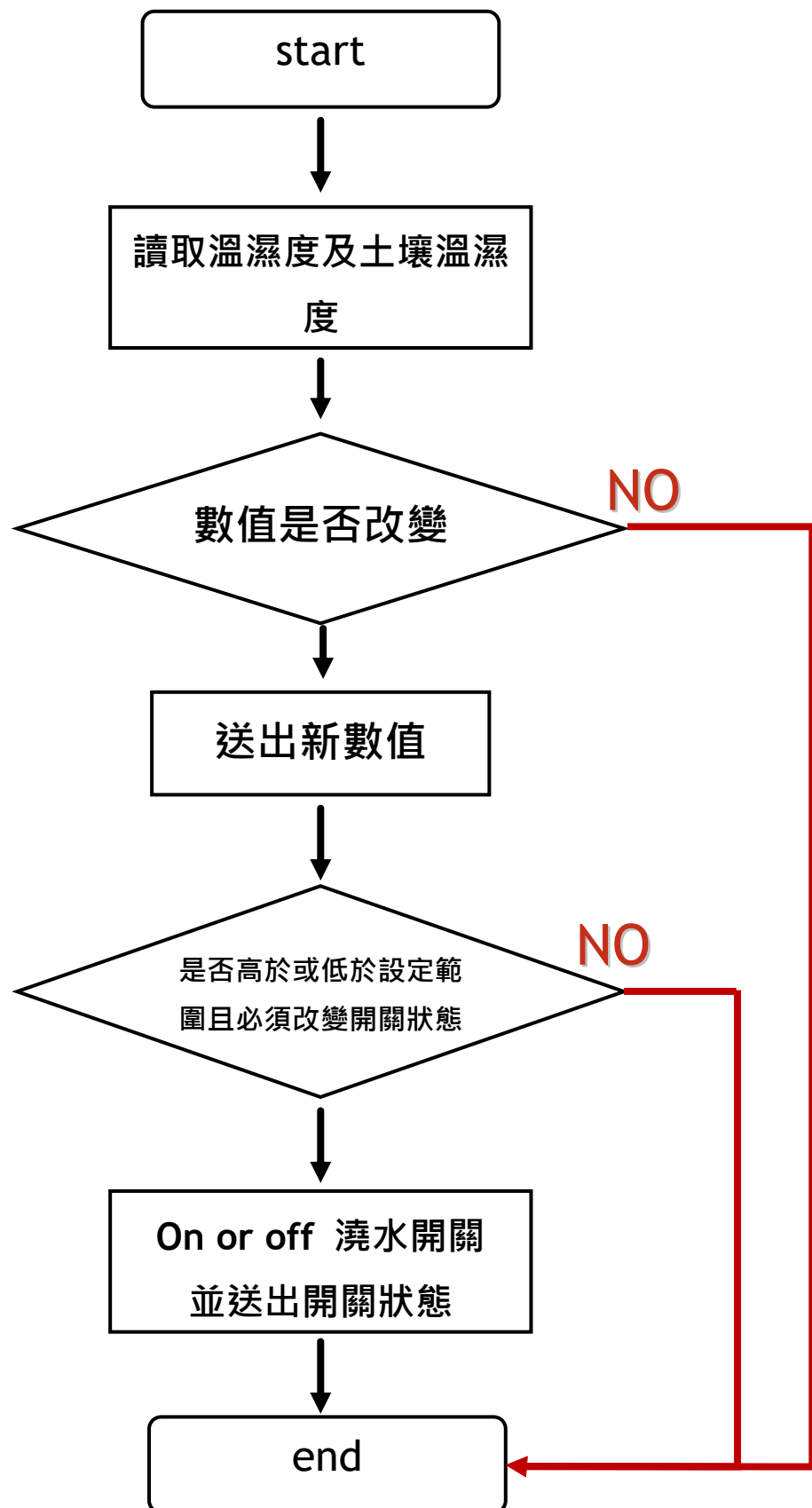
自動澆水時,啟動抽水法達的下限值:位置 22

自動澆水時,關閉抽水法達的上限值:位置 24

Node 開機流程



每 5 秒執行一次感測流程(pot)



可使用 APP 在遠端打開抽水馬達開關和設定是否開啟自動澆水設定及設定自動澆水的上下限值  
例如：

以下步驟為在遠端打開抽水馬達的流程

- ①查 routine table 找出要設定的 node 是與哪個 gateway 連線
- ②向此 gateway 訂閱的 mqtt topic 發布訊息
- ③訊息前 6 碼填入 SID,後面接著要送到 node 上的內容(pot001SET+SW011)
- ④node 收到訊息並打開開關(如果是設定自動澆水開關,自動澆水的上限值或下限值,還會將這些數值紀錄至 EEPROM)
- ⑤node 送出訊息給 server 告知此開關的狀態已被打開
- ⑥server 更新這個 node 的狀態
- ⑦將這訊息推播給 user

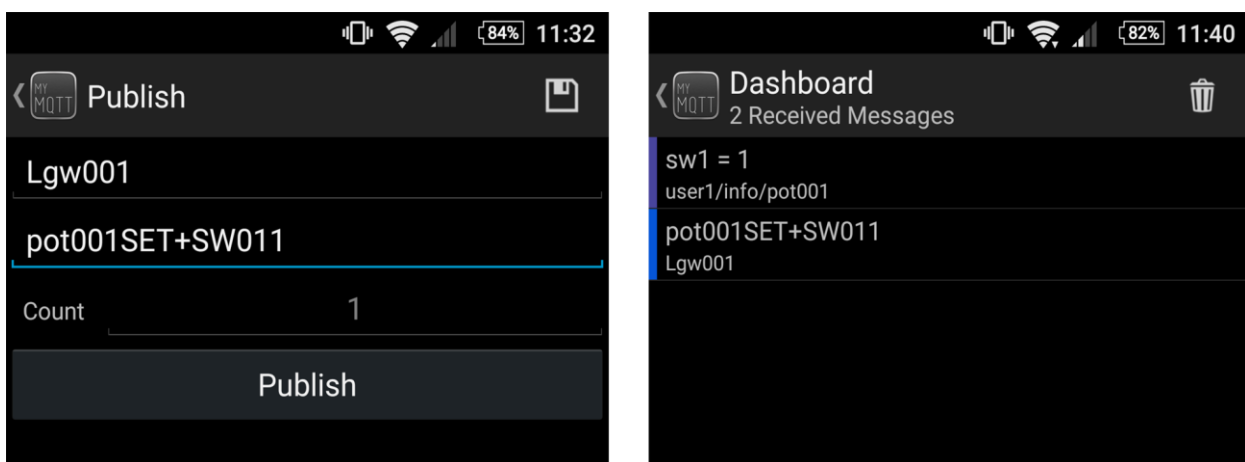


圖 使用 mqtt 開啟 node 上的開關

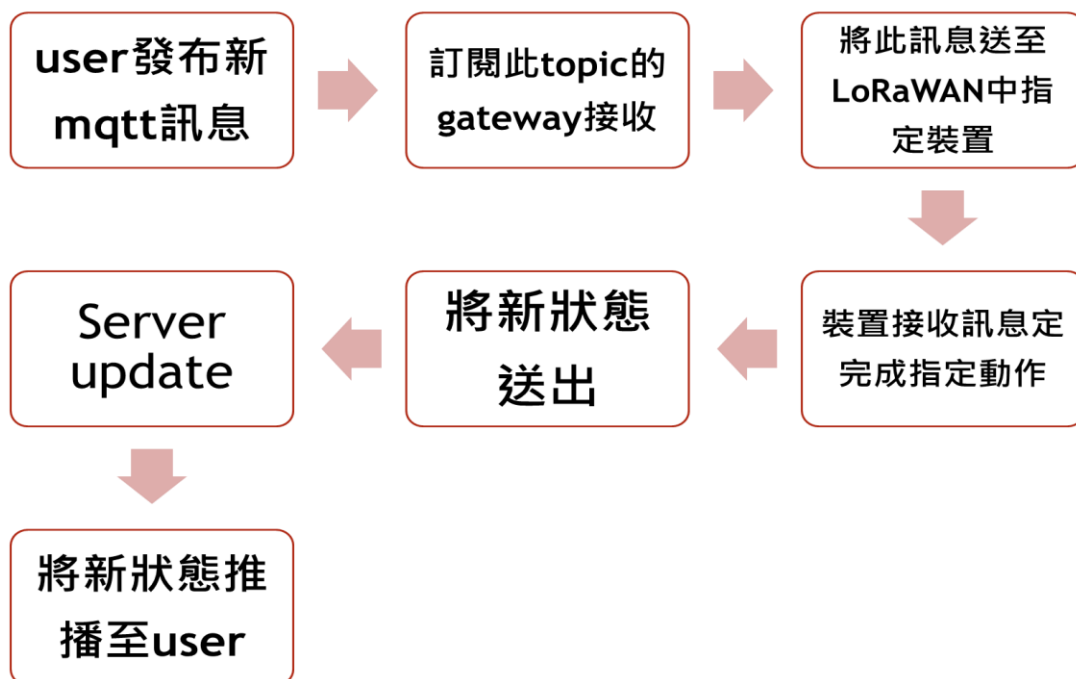


圖 使用 mqtt 控制 node 流程

Node 設定字串:

開關 01 設定:

SET+SW01?-----問號為要設定的狀態,1 為開 0 為關

自動澆水設定:

SET+AUTO?-----問號為要設定的狀態,1 為開 0 為關

自動澆水上下限設定:

SET+STWT[下限],[上限]-----SET+STET20,40

User 設定

SET+USER?????-----SET+USERtest

接線圖:

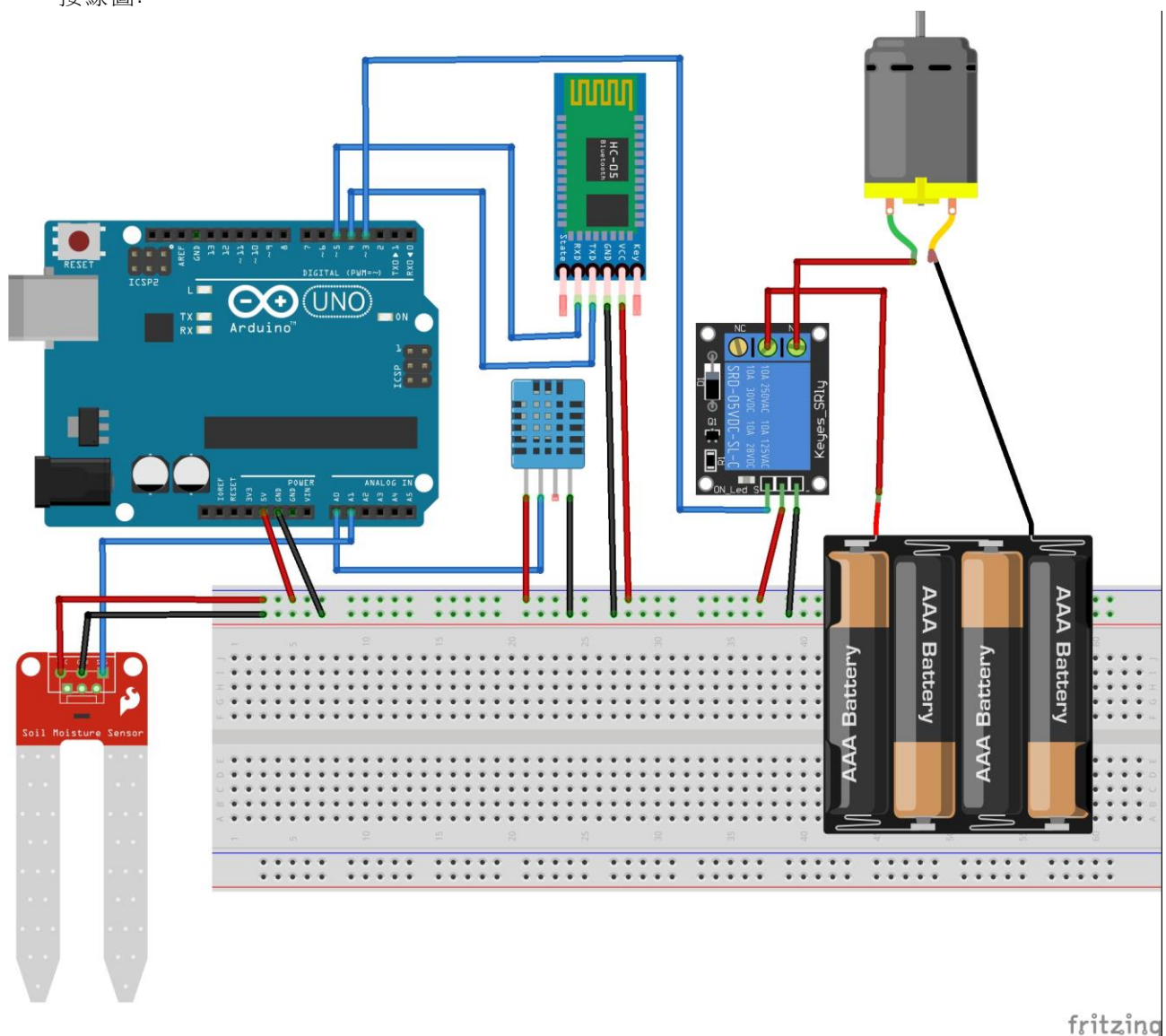


圖 農業應用 node 接線圖



## 8.定位用 node

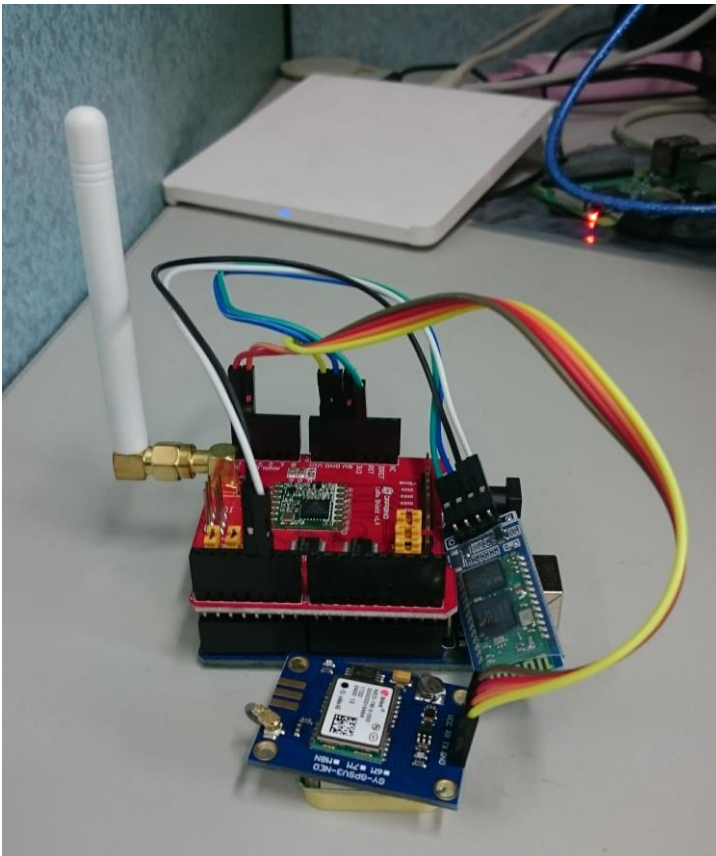


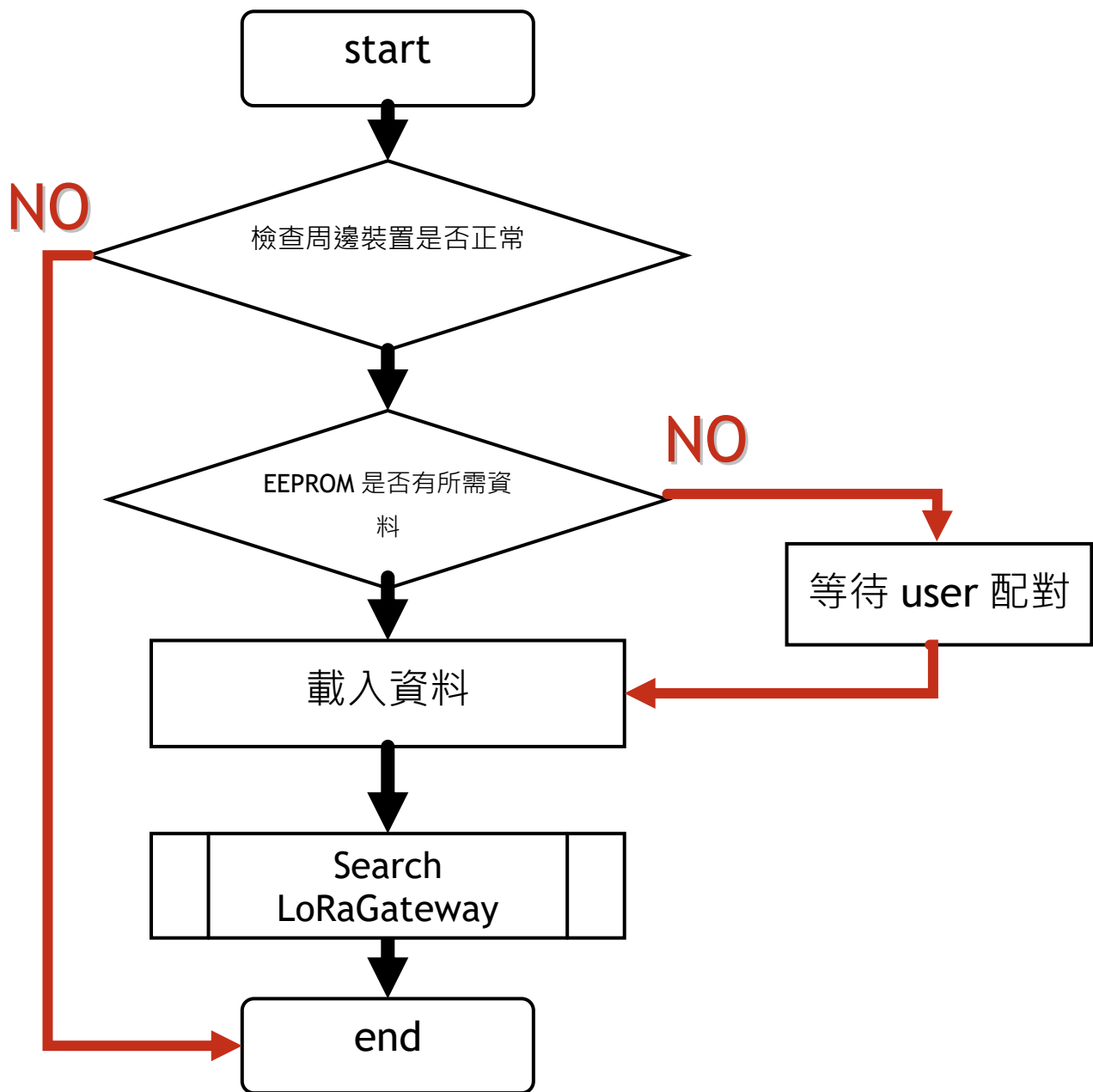
圖 定位用 node

這類 node 由一片 arduino,一片 Dragino Lora Shield 868MH,一片藍芽模組(CH-06)加上 GPS 衛星定位模組 (Ublox NEO-7M)組成

在使用前必須先使用藍芽與此裝置配對並透過藍芽告知此裝置的 user,設定字串為(SET+USER+"user")例如這裝置的 user 為 user1,在使用前就必須先傳一段 SET+USERuser1 給此裝置,此時 arduino 就會將這些資料記錄至 arduino 上的 EEPROM,所以如果在 node 開機時發現 EEPROM 內沒 user 資料的話就會進入 busy wait 直到有使用者使用藍芽告知此裝置的 user 操作才會繼續

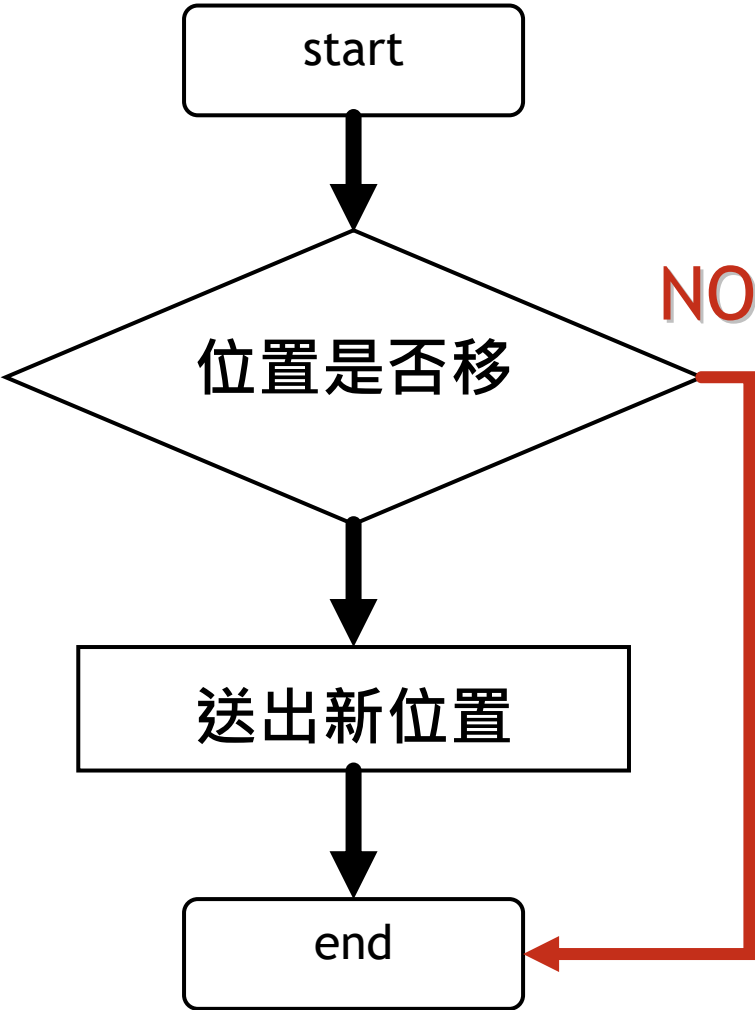
EEPROM 內資料擺放位置:

User:起始位置為 0





每 30 秒執行一次定位流程：



Node 設定字串:

User 設定:

SET+USER?????-----SET+USERtest

GPS node 接線圖:

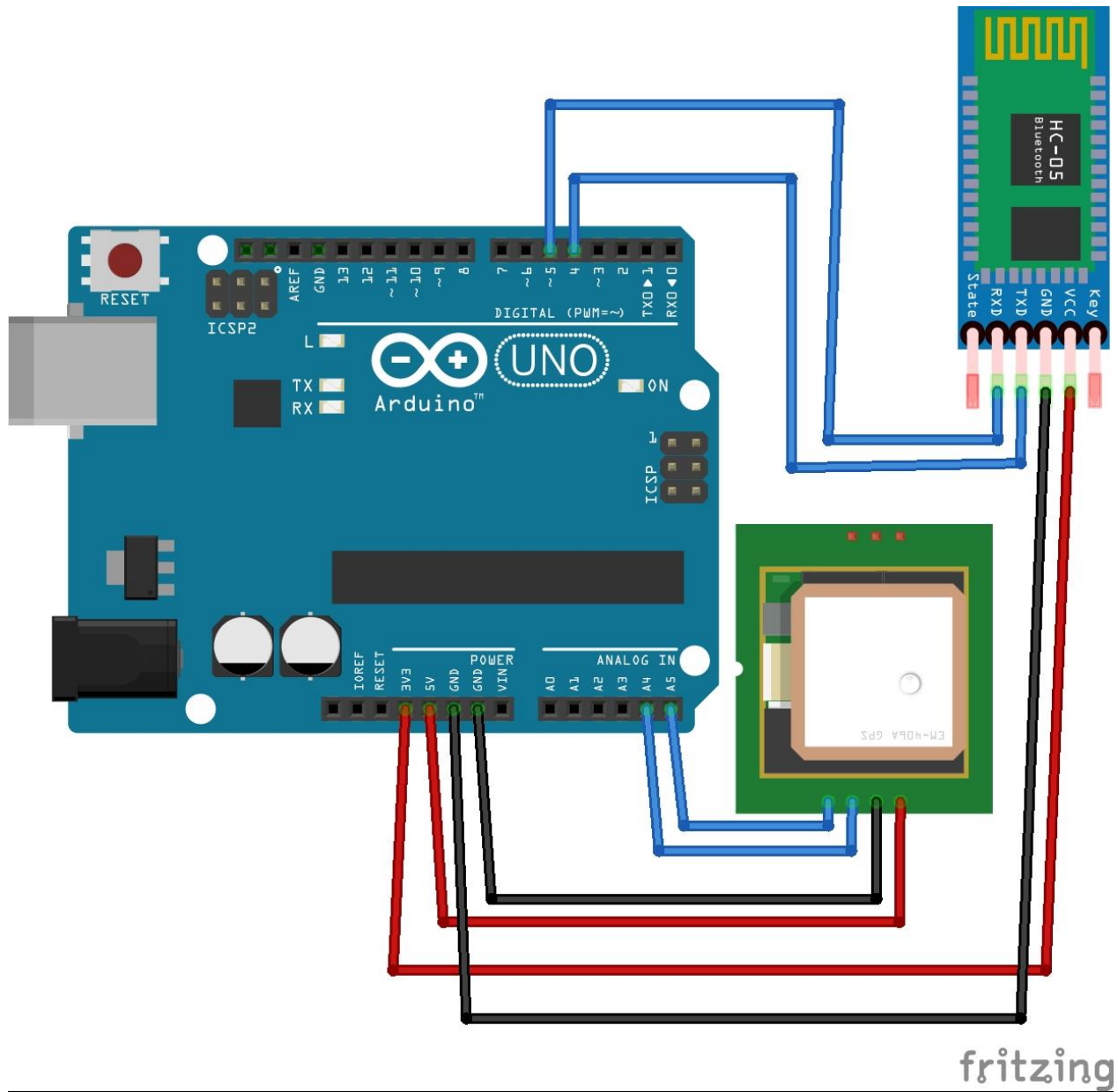


圖 定位用 node 接線圖

## 9.server 資料庫設計:

每個 node 都有一個資料表記錄著這 node 的狀態,相關設定及感測資料

每個 user 都有一個獨立的資料庫

資料庫內有該 user 所有的 node 的資料表,及這些 node 的 routine table



圖 user 資料庫

LoRaNode 即為這些 node 的 routine table



圖 routine table

## 10.mqtt 推播設定

Server 有訊息要告知使用者會向 mqtt user/info/ topic 發布訊息

例如:server 要告知 user1 的 pot001 node 上的開關被打開,就會向 user1/info/pot001 topic 發布訊息 sw1 = 1,所以 user1 只要訂閱 user1/info/# topic 就可以接收到 server 要發給 user1 的所有訊息

Topic info 的下一層為觸發此訊息的 node 的 SID

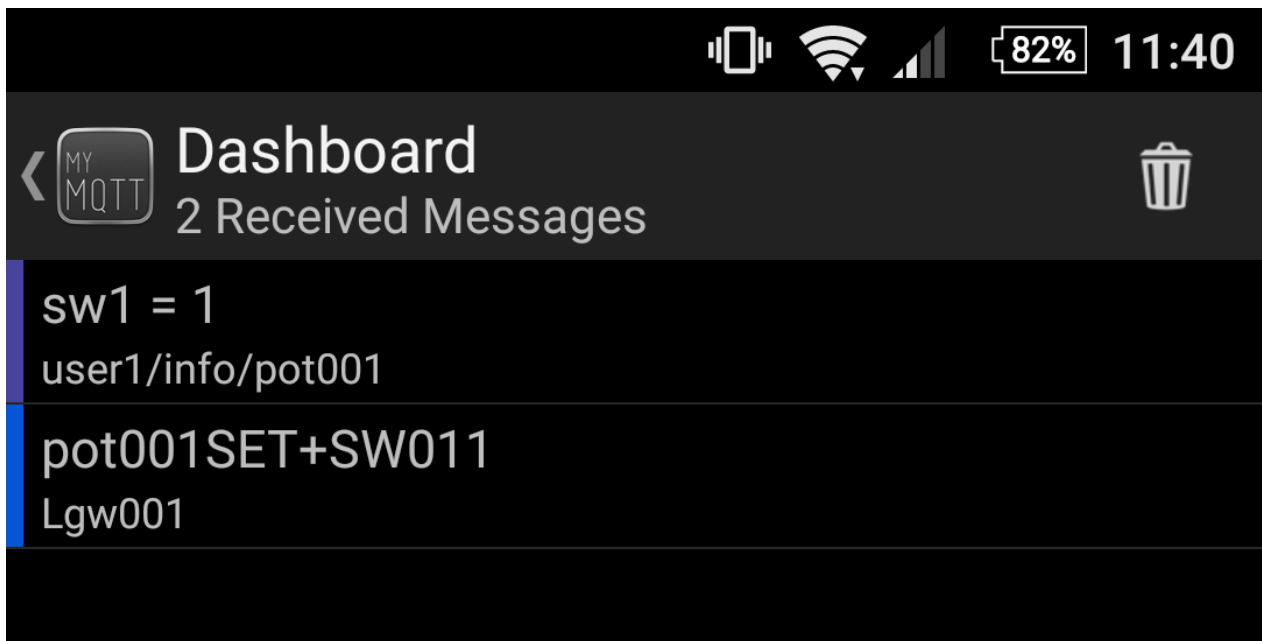


圖 mqtt 推播

## 11.server API:

API 使用 javascript 撰寫

使用方式為:

```
$sudo node dataHandleDocker.js [server ip] [mqtt port] [API port] [data base container name]
[data base root] [data base root password] &
```

例:

```
$sudo node dataHandleDocker.js 104.155.210.211 1883 8000 mysql root 123456789 &
```

### (1)addPot API:

將農業用 node 感測到之後送上來的資料記錄到資料庫中這個 node 的資料表並且檢查感測到的空氣溫度,濕度及土壤溼度是否超過使用者所設定的合理範圍,如果超過就會發出推播通知使用者  
<http://104.155.210.211:8000/addPot?user=user1&sid=pot001&t=27&h=37.00&sh=50>

### (2)Chval API:

用來 update node 狀態

例如:有個開啟自動澆水開關的 node 因為土壤濕度低於使用者設定而打開連接抽水馬達的繼電器時,就會送出

```
http://104.155.210.211:8080/chval?user=user1&sid=pot001&vari=sw1&val=1
```

update 紀錄 pot001 的資料表,並且向 mqtt topic user1/info/pot001 發出訊息 sw1 = 1 通知灑水開關被打開

### (3)chWater API

用來改變開啟自動澆水功能時,打開抽水馬達的下限值及關閉抽水馬達的上限值,所以當 node 上的設定更改時就會送出

```
http://104.155.210.211:8000/chWater?user=user1&sid=pot001&start=10&stop=60
```

update 紀錄 pot001 的資料表,並且向 mqtt topic user1/info/pot001 發出訊息 waterStart=10,waterStop=60 通知更改後的數值

### (4)addGPS API:

將定位用 node 送上來的經緯度資料記錄到資料庫中這個 node 的資料表

```
http://104.155.210.211:8080/addGPS?user=user1&sid=GPS001&lat=24.143387&lon=120.728520
```

### (5)registerLgw API:

用來新增或改變 routine table 當 pot001 要跟 Lgw01 連線時就會送出

```
http://104.155.210.211:8080/registerLgw?user=user1&sid=pot001&gw=Lgw01
```

