



# 用Python學智慧聯網

## Chapter 05 : IoT 應用 – 體溫通報器

# 踏入 AIoT 的世界



1 何謂 AIoT



5 IoT 應用 – 體溫通報器



2 用Python 玩轉 AI



6 二元分類 – 雲端步頻紀錄儀



3 AI 的小大腦 – 微控制器



7 多元分類 – 無線體感鍵盤



4 迴歸問題 – 體溫監測站



8 CNN – 智慧聲控燈



# 前言

- 如果想隨時監控體溫變化，甚至發燒時傳送訊息提醒，這要怎麼做呢？
- 將體溫上傳至 IoT 平台，並監測體溫是否超過 37.5 度，如有發生直接傳送 LINE 訊息給緊急聯絡人。

# 前言

- 5-1 ESP32 連接至網際網路
- 5-2 中華電信 IoT 智慧聯網大平台
  - LAB06 體溫紀錄儀
- 5-3 使用 IFTTT 發送 LINE 通知
  - LAB07 防疫作戰：體溫推播系統

# 5-1 ESP32 連接至網際網路

ex5-1

- ESP32 本身就是可連接無線網路的晶片。
- 匯入 network 模組，並建立 WLAN 物件：

Thonny

```
>>> import network
>>> sta=network.WLAN(network.STA_IF)
```

# ESP32 連接至網際網路

ex5-1

| 網路介面           | 說明  |
|----------------|---|
| network.STA_IF | 工作站 (station) 介面，專供連上現有的 Wi-Fi 無線網路基地台，以便連上網際網路 |
| network.AP_IF  | 熱點 (access point) 介面，可以讓 ESP32 變成無線基地台，建立區域網路   |

- 啟用網路介面：

```
>>> sta.active(True)
```

Thonny

# ESP32 連接至網際網路

ex5-1

Thonny

```
>>> sta.connect('無線網路名稱', '無線網路密碼')
```

Thonny

```
>>> sta.connect('FLAG', '12345678')
```

- 使用 `isconnected()` 確認是否連上無線網路：

Thonny

```
>>> while not sta.isconnected():  
    pass
```

## 5-2 中華電信 IoT 智慧聯網大平台

- 註冊會員
- 新增專案
- 測試上傳資料
- 程式上傳資料

# 註冊會員

ex5-2

1 輸入 <https://iot.cht.com.tw/iot/> 連接至中華電信 IoT 平台



2 點擊登入

歡迎登入

中華電信會員中心登入

3 點擊中華電信會員中心登入

密碼

請輸入密碼



保持登入狀態 (公用電腦不建議使用)

登入

[忘記帳號?](#) | [忘記密碼?](#)

4 將視窗下滑

沒有帳號嗎? 立刻加入會員

5 點擊沒有帳號嗎? 立刻加入會員

# 註冊會員

6 輸入手機號碼、驗證碼

手機號碼

請輸入手機號碼

圖形驗證碼

請輸入圖片中內容

46483

重新產生驗證碼 | 語音播放

7 輸入完畢後點擊

發送簡訊驗證碼

8 輸入手機收到的驗證碼

簡訊驗證碼

請輸入簡訊收到的驗證碼

9 輸入 2 次密碼

密碼

請輸入8碼以上英數字混合密碼

密碼強度：

確認密碼

請再次輸入您的密碼

10 選擇同意

同意  不同意 \*中華電信個資共同行銷條款

11 兩者皆勾選

\*我已詳閱 中華電信個資蒐集告知聲明

\*我了解並同意 中華電信會員條款

12 點擊確定送出

確定送出

# 註冊會員

member.cht.com.tw 顯示  
如不設定身分證號，將無法以本會員帳號使用身分證號下紅利點數與其他個人服務。

**14** 點擊確定

設定證號

確定 取消

**13** 點選現在還不要

確定

現在還不要

**15** 看到此畫面後，重新連接至 <https://iot.cht.com.tw/iot/> 並點選登入

註冊完成!如需查詢或異動會員資料，請於帳號管理操作。

前往會員中心

# 註冊會員

**登入 IOT智慧聯網平台**

**會員手機簡訊認證**  
使用您的手機取得簡訊驗證碼進行登入。

**會員帳號**  
請輸入會員帳號或手機號碼

**密碼**  
請輸入密碼

保持登入狀態 (公用電腦不建議使用)

**登入**

**16 輸入會員帳號、密碼**

**17 點擊登入**

**請輸入您的聯絡信箱**  
設定後您亦可使用此信箱作為您的登入帳號。

**E-mail帳號**  
請輸入您的E-mail信箱

**18 輸入 E-mail**

**確定**

**19 點擊確定**

# 註冊會員

**\* 名稱**

平台帳號名稱(共享專案使用會員名稱)

**\* 暱稱**

在 IoT 智慧聯網平台使用者的顯示名稱

**\* 電子郵件**

行業別

**\* 用戶身分**

組織/公司

備註

帳號金鑰

**\* 為必填欄位!**

歡迎加入中華電信會員!

您已經完成所有步驟了，請開始使用服務吧。

會員帳號

登入手機

**21 更改用戶身分為創客**

**22 點擊註冊會員**

**20 點擊確定**

# 新增專案

ex5-2

## 專案精靈

您尚未建立任何專案，是否立即透過專案精靈一次新增專案、設備與感測器？

否

是

1 點選是

## 1.新增專案

### 基本資料

專案名稱：

體溫監測站

2 專案名稱輸入體溫監測站

專案描述：

記錄體溫

3 專案描述輸入記錄體溫

應用領域：

其他

4 選擇其他

取消

下一頁

5 點擊下一頁

# 新增專案

ex5-2

2.新增設備

基本資料

設備名稱 ESP32

設備描述 物聯網裝置

6 設備名稱輸入 ESP32、  
設備描述輸入物聯網裝置

模組元件  設備是否採用中華電信硬體安全元件

設備金鑰 D10F2277E7D49F1E04C

專案最多允許建立 4096 個設備

關閉 上一頁 下一頁

7 點擊下一頁

# 新增專案

ex5-2

### 3.新增感測器 ×

基本資料

識別編號(ID)   
識別編號只允許輸入英文或數字或底線符號

顯示名稱

描述

類型  數值  文字  開關  圖像

單位

**8** 識別編號輸入 **thermistor**、  
顯示名稱輸入**熱敏電阻**、  
描述輸入**感測溫度**、  
單位輸入**°C**

**9** 點擊**儲存**

先切換成中文輸入法，並點擊 **Ctrl** + **Alt** + **,** 即可從小鍵盤中找到**°C**。

關閉
上一頁
儲存

# 新增專案

新增完成
✕

新增專案成功，Project ID : 189012  
 新增設備成功，Device ID : 21901240987  
 新增感測器成功，Sensor ID : 1

1.可參考下圖透過設備金鑰上傳資料，DEVICE\_KEY : 00000000-0000-0000-0000-000000000000  
 URI : <https://iot.cht.com.tw/iot/v1/device/21901240987/rawdata>  
 CK : 00000000-0000-0000-0000-000000000000

| POST         | /iot/v1/device/\${device_id}/rawdata   | 儲存感測資料 |
|--------------|--|--------|
| 協定名稱         | 儲存感測資料   |        |
| URI          | <a href="https://iot.cht.com.tw/iot/v1/device/\${device_id}/rawdata">https://iot.cht.com.tw/iot/v1/device/\${device_id}/rawdata</a>  |        |
| Request 格式說明 | headers:<br>CK:\${PROJECT_KEY or DEVICE_KEY}<br>Content-Type:application/json<br><br>body:<br><pre>[   {     "id": "temperature",     "time": "2017-08-04T10:43:58",     "value": ["26.8"]   } ]</pre> |        |

2.上傳完成後，可點選進入專案內查看是否成功上傳  
 3.如何使用IoT平台提供之API：[API文件](#)

若需更詳細上傳資訊請點選下方按鈕前往「開發者中心」，或點選「結束」直接開始使用

開發者中心
10 點擊結束

11 看到此畫面，代表新增專案完成


智慧聯網大平台  
SMART PLATFORM

首頁 / 專案管理

✎
🔍
👥
🗑️



## 體溫監測站

記錄體溫

# 測試上傳資料

『 ESP32 』設備資訊 (編號:XXXXXXXXXXXX)

感測器 設備內容 事件驅動 憑證

共有 1 個感測器

°C

熱敏電阻

1 點擊圖示

2 點擊 >-

# 測試上傳資料

請輸入感測數值或控制與確認訊息

感測數值(Rawdata)  3 輸入 36 4 點擊確定

取消 確定



5 看到 36 °C 顯示於此

開發者中心 專案管理 應用服務 ▾ 聯絡我們

- 儀表板
- 大數據分析
- 通用型監控

6 點選應用服務 / 儀表板

# 測試上傳資料

旗標創客



# 測試上傳資料

### Widget 參數設定

數據折線圖：指定單一感測項目的感測數據以折線圖表方式呈現

請選擇 Chart 類型  
Chart.js

請選擇專案  
體溫監測站 **10 專案選擇體溫監測站**

請選擇設備  
ESP32 **11 設備選擇 ESP32**

請選擇感測器  
熱敏電阻 **12 感測器選擇熱敏電阻**

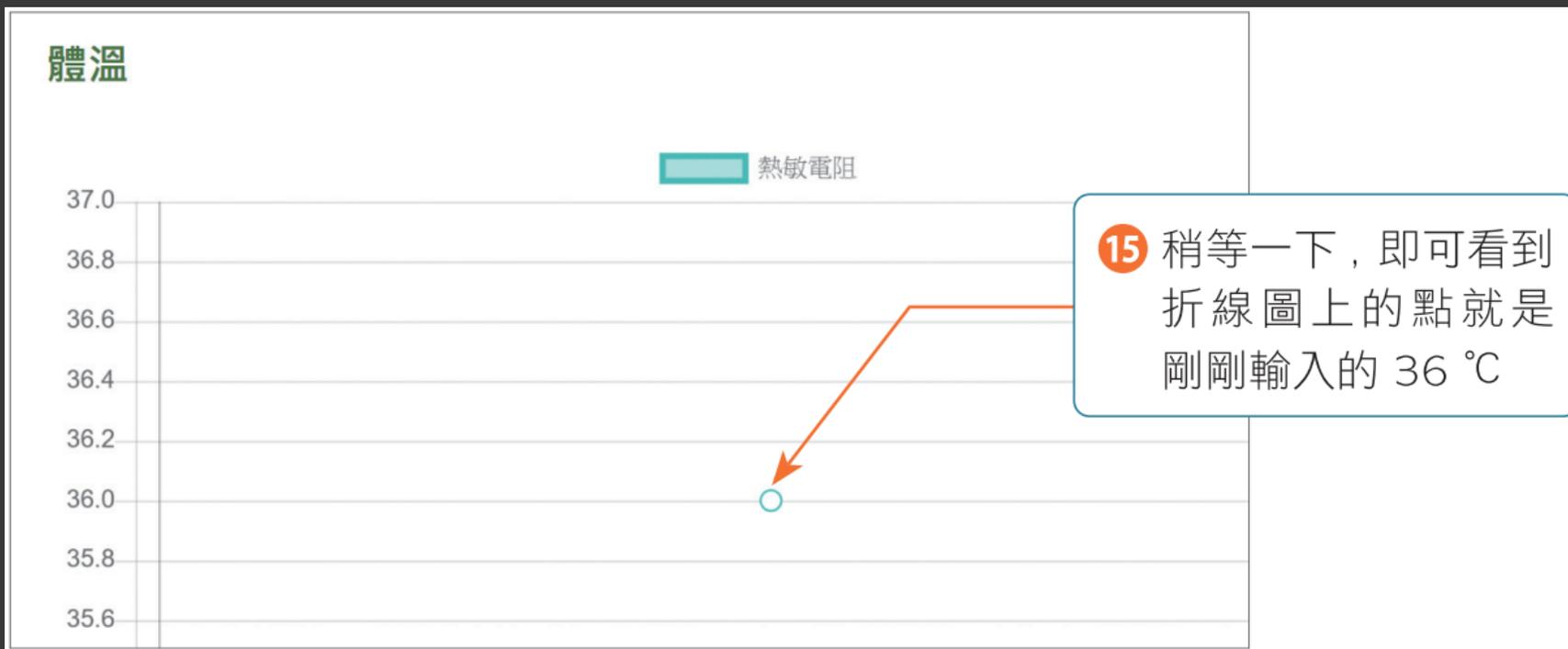
- 元件顯示感測器名稱  
熱敏電阻

- 顯示數量  
30

選擇顯示大小  
大 (以畫面寬度大小呈現) **13 更改為大 (以畫面寬度大小呈現)**

[返回 Dashboard](#) [新增 Widget](#) **14 點擊新增 Widget**

# 測試上傳資料



# 程式上傳資料

## 1 點選開發者中心

IoT 智慧聯網大平台  
SMART PLATFORM

開發者中心 專案管理 應用服務 聯絡我們 繁體中文

首頁 / 開發者中心

### 開發者中心

我們提供中華電信物聯網之【**平台介紹**】，說明平台架構與特色；  
使用者可參考【服務說明/核心服務】中「**裝置連線管理**」的指引，將您的設備連上物聯網平台；  
並透過【服務說明/核心服務】中的「**應用開發支援**」說明及【**API文件**】等說明協助您快速打造應用服務。  
【**應用案例**】與【**相關下載**】中提供豐富應用實例與相關範例程式下載，可幫助您更輕易上手，一同加入物聯網生態圈。

平台介紹 服務說明 **API文件** 應用案例 相關下載

## 2 點選 API 文件

# 程式上傳資料

## 3 點選核心服務



## 4 點選感測資料



https://iot.cht.com.tw/iot/v1 - Server URL

此區塊即是各種服務的 API 網址

一般資料 感測資料為數字或文字類型資料

POST /device/{device\_id}/rawdata 儲存感測資料

POST /device/{device\_id}/rawdata 儲存數字類型資料

# 程式上傳資料

API 網址前都需要加上此段來變成完整路徑

https://iot.cht.com.tw/iot/v1 - Server URL ▾

一般資料 感測資料為數字或文字類型資料 ▾

**POST** /device/{device\_id}/rawdata 儲存感測資料

**POST** /device/{device\_id}/rawdata 儲存數字類型資料

儲存數字類型資料

Parameters Try it out

| Name                           | Description   |
|--------------------------------|---|
| <b>CK</b> * required<br>string | 專案金鑰或設備金鑰，專案金鑰可以從網站->專案管理-->單個專案-->編輯專案-->權限資料查詢獲得；設備金鑰可以從專案管理-->單個專案-- |

5 點擊儲存數字類型資料

# 程式上傳資料

- 將上面兩段網址合併：

```
http://iot.cht.com.tw/iot/v1/device/設備編號/rawdata
```

# LAB06 體溫紀錄儀

- 實驗目的：  
上傳 LAB05 的體溫值至中華電信 IoT 平台
- 材料、線路圖：同 LAB02
- 開發環境：Thonny

# 實驗原理

- requests 模組可讓程式扮演瀏覽器的角色，MicroPython 中提供的是精簡版的 urequests 模組。匯入此模組，即可使用該模組連線網路服務。
- POST 代表 HTTP 協定的 POST 方法，因此可使用 urequests 模組中的 post() 來處理。

# 實驗原理

ex5-3

Thonny

```
import urequests  
data = {"value":["36"]} ←以 Python 字典來設定要上傳的資料  
urequest.post("http://xxxx.xxxx", json=data)
```

# 實驗原理

- 中華電信 IoT 平台上傳資料時除了要 API 網址，還需要提供 3 個相關資訊：
  1. 設備金鑰 (CK)
  2. 設備編號 (device\_id)
  3. 感測器識別編號 (id)

# 實驗原理

## 1 感測器識別編號

The screenshot displays the IOT Smart Platform interface. The top navigation bar includes the logo 'IOT 智慧聯網大平台 SMART PLATFORM' and a breadcrumb trail '首頁 / 專案管理'. Below the navigation bar is a toolbar with icons for edit, search, group, and delete. A folder icon is highlighted with a red box. A callout box on the right side of the interface shows a close button, a refresh button, and an edit button, with the edit button highlighted by a red box. The main content area shows a temperature monitoring station with a large green '36 °C' display, a timestamp '2020-05-29T12:58:10.241Z', and the sensor type '熱敏電阻'.

- 1 回到專案管理
- 2 點擊資料夾
- 3 點擊

# 實驗原理

## • ① 感測器識別編號

感測器管理

基本資料 其他資料

識別編號 (ID) thermistor 感測器識別編號  
識別編號只允許輸入英文或數字或底線符號

顯示名稱 熱敏電阻

設備最多允許建立 128 個感測器

4 點擊關閉 關閉 下一頁

# 實驗原理

## ② 設備金鑰和設備編號

點擊設備內容

感測器 設備內容 事件驅動 憑證申請

共有 1 個感測器

36 °C

2020-05-29T13:09:05.625Z

熱敏電阻

感測器 設備內容 事件驅動 憑證申請

設備內容

編號: 4C1M3HGBEGCGRRTE 設備編號

名稱: ESP32

描述: 物聯網裝置

類型: general

URI: 4C1M3HGBEGCGRRTE

擴充屬性  
資訊: 屬性尚未設定 金鑰

設備金鑰: D0808C74707454545454545454545454



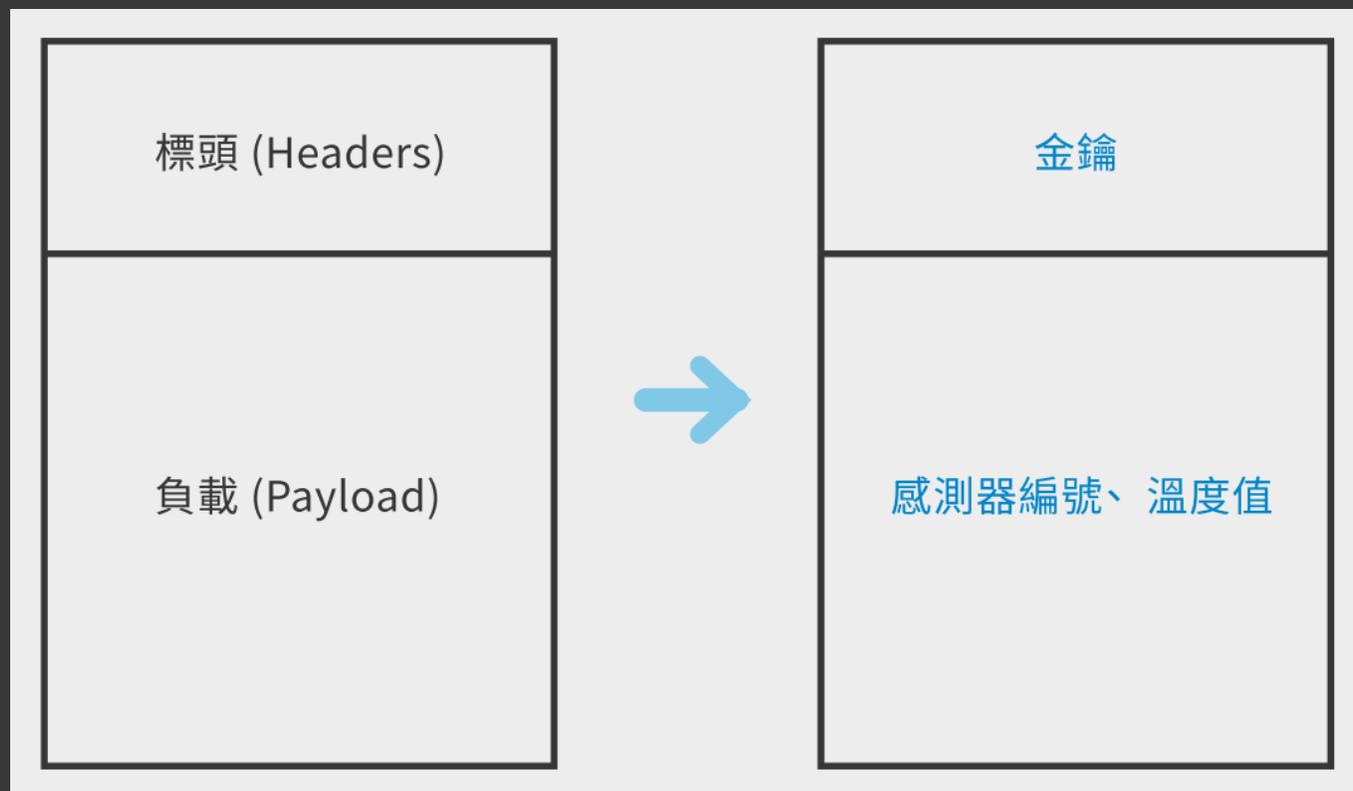
# 實驗原理

Thonny

```
device_id = "請填入中華電信設備編號"  
headers={"CK":"請填入中華電信設備金鑰"}  
url_CHT="http://iot.cht.com.tw/iot/v1/device/"+ \  
        device_id+"/rawdata"  
CHT_data=[{"id":"請填入感測器識別編號","value":["36.5"]}]  
urequests.post(url_CHT,json=CHT_data,headers=headers)
```

# 實驗原理

- POST 內容包含標頭、負載。



# 程式設計

LAB06.py

體溫紀錄儀

Thonny

```
1  from machine import Pin,ADC
2  import time
3  from keras_lite import Model
4  import ulab as np
5  import network          # 匯入network模組
6  import urequests       # 匯入urequests模組
7
8  # 連線至無線網路
9  sta=network.WLAN(network.STA_IF)
10 sta.active(True)
11 sta.connect('無線網路名稱','無線網路密碼')
12
13 while not sta.isconnected() :
```

# 程式設計

```
14     pass
15
16     print('Wifi連線成功')
17
18     device_id = "請填入中華電信設備編號"
19     headers={"CK":"請填入中華電信設備金鑰"}
20
21     # 中華電信IoT平台
22     url_CHT="http://iot.cht.com.tw/iot/v1/device/" \
23           +device_id+"/rawdata"
24
25     mean=170.98275862068965
26     std=90.31162360353873
27     model = Model('temperature_model.json')
28
29     adc_pin = Pin(36)
```

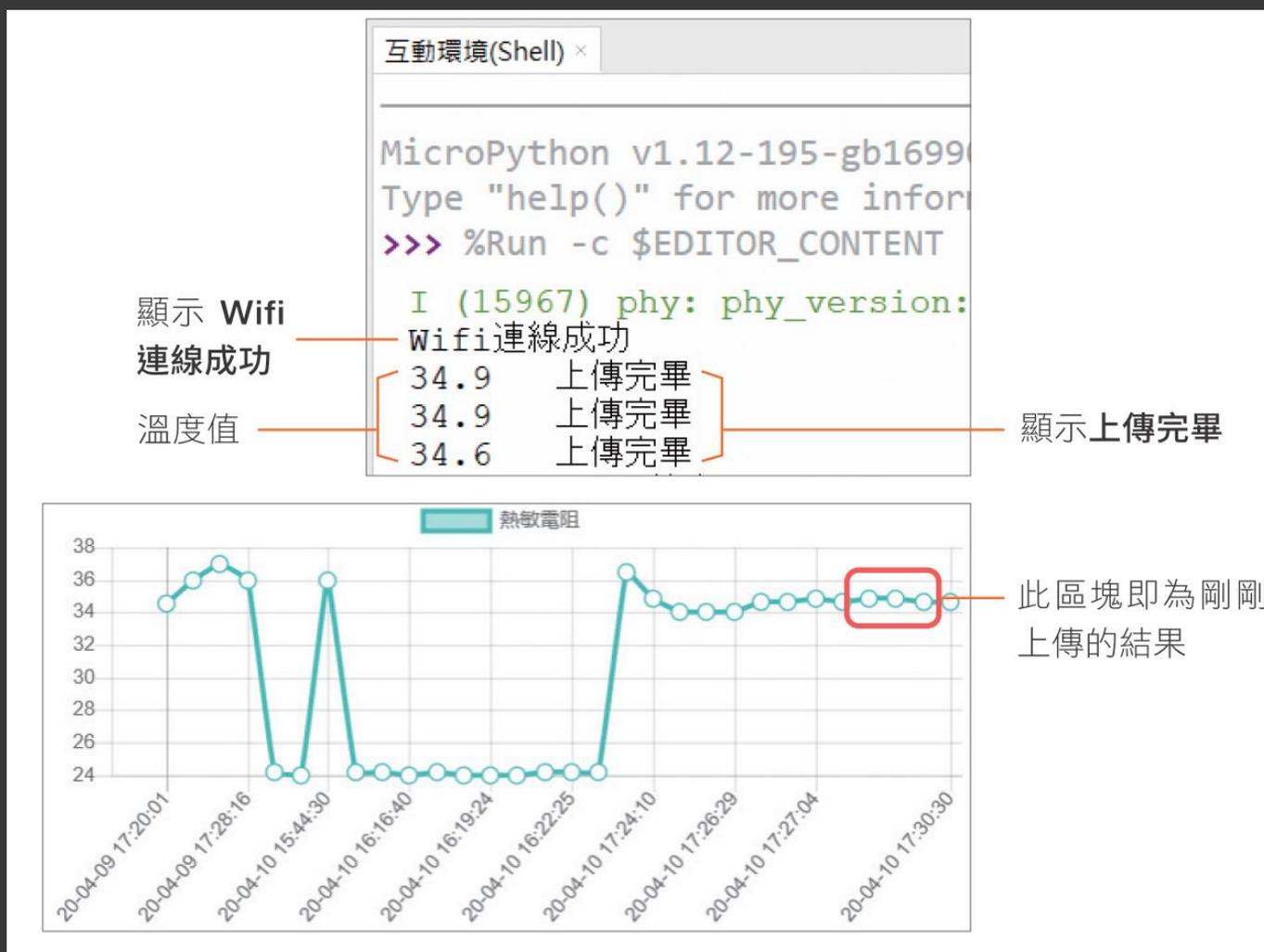
# 程式設計

```
30 adc = ADC(adc_pin)
31 adc.width(ADC.WIDTH_9BIT)
32 adc.atten(ADC.ATTN_11DB)
33
34 data=0
35
36 while True:
37
38     for i in range(20):
39         thermal=adc.read()
40         data=data+thermal
41         time.sleep(0.01)
42
43     data=data/20
44
45     data = np.array([int(data)])
```

# 程式設計

```
46     data = data-mean
47     data = data/std
48     tem = model.predict(data)
49     tem = round(tem[0]*100,1)
50     print(tem,end='    ')
51
52     CHT_data=[{"id":"請填入中華電信感測器編號",
53               "value":[str(tem)]}]
54
55     urequests.post(url_CHT,json=CHT_data, # 上傳資料
56                   headers=headers)
57
58     print("上傳完畢")
59
60     data=0
61     time.sleep(60)          # 暫停60秒
```

# 測試程式



## 5-3 使用 IFTTT 發送 LINE 通知

- IFTTT 是英文 “IF This, Then That” 的縮寫，其服務的精神就是『如果 A 然後就 B』，：

**If** **+** **This** **Then** **That**

**A** **B**

# 註冊 IFTTT 會員

ex5-4

## Get started with IFTTT



Continue with Apple

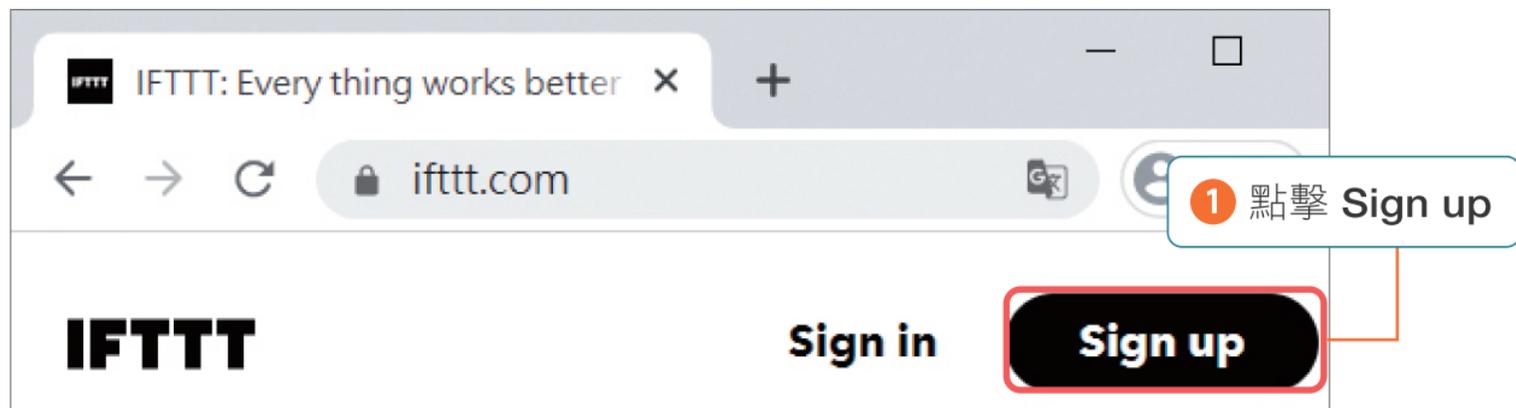


Continue with Google



Continue with Facebook

Or use your password to **sign up** or [sign in](#)



- 2 可以用 Apple、Google 和 Facebook 帳號註冊，或者用其它信箱。我們選擇用其它信箱，點選下方 **sign up**

# 註冊 IFTTT 會員

The image shows a 'Sign up' form for IFTTT. The form includes an 'Email' input field, a 'Password' input field, a checkbox for 'Get updates for products available on IFTTT', and a large black 'Sign up' button. Below the button is a link that says 'Continue with Apple, Google, or Facebook'. Three numbered annotations are present: 3 points to the Email field, 4 points to the Password field, and 5 points to the Sign up button.

**Sign up**

Email

Password

Get updates for products available on IFTTT

**Sign up**

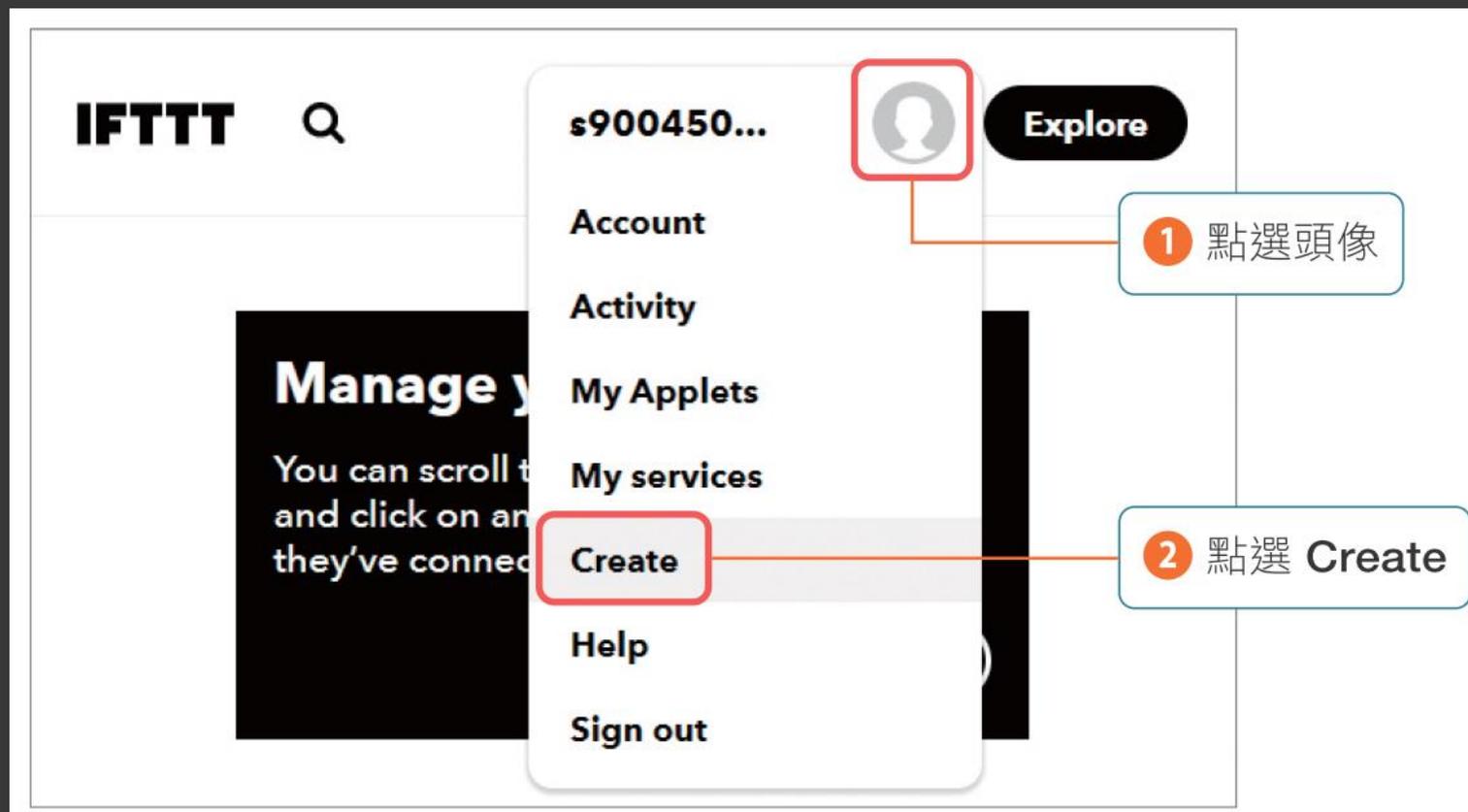
[Continue with Apple, Google, or Facebook](#)

3 輸入 Email 信箱做為會員帳號

4 輸入會員密碼

5 點選 Sign up 完成註冊

# 設定事件 A 當作觸發條件：



If **+ This** Then That

3 設定事件 A。點選 +This

# 設定事件 A 當作觸發條件：

webhooks

4 在搜尋欄位輸入 webhooks



Webhooks

5 選擇 Webhooks

6 點擊 Connect

Connect

## Receive a web request

This trigger fires every time the Maker service receives a web request to notify it of an event. For information on triggering events, go to your Maker service settings and then the listed URL (web) or tap your username (mobile)

7 選擇 Receive a web request

# 設定事件 A 當作觸發條件：

Event Name

temperature

The name of the event, like "button\_pressed" or "f

8 輸入 temperature 當名稱

**Create trigger**

9 輸入完按 Create trigger

If  Then  That

10 設定事件 B。點選 +That

# 設定事件 B :



# 設定事件 B :

## IFTTT

IFTTT, Inc.

將提供用戶名稱及聊天室列表給IFTTT服務的提供者。您可於LINE Notify的個人頁面解除連動。

同意後便會自動將「LINE Notify」官方帳號加入好友。

取消

同意並連動

5 點擊同意並連動

### Send message

This Action will post a message to LINE.

6 選擇 Send message



# 設定事件 B :

ex5-5

Recipient

透過1對1聊天接收LINE N... ▼

Message destination

Message

Value 1: {{Value1}}<br>  
Value 2: {{Value2}}<br>  
Value 3: {{Value3}}

Add ingredient

Photo URL

Add ingredient

Create action

7 點擊 Message 區塊

警告!現在體溫為{{Value1}}°C

8 刪除原內容, 並填入警告!現在體溫為 {{Value1}}°C

⚠ Value1 旁的前後大括號都各有 2 個

9 修改完成點擊 Create action

**Finish**

10 點擊 Finish

# 手動請求發出 LINE 訊息：

1 點擊左上圖示



2 點擊 Documentation



# 手動請求發出 LINE 訊息：

Your key is: [REDACTED]

◀ Back to service

## To trigger an Event

Make a POST or GET web request to:

```
https://maker.ifttt.com/trigger/temperature/with/key/[REDACTED]
```

With an optional JSON body of:

```
{ "value1" : 37.5, "value2" : " ", "value3" : " " }
```

The data is converted to variables. This **value1 輸入 37.5** so pass value1, value2, and value3 as query parameters or form Action in your Recipe.

You can also try it with `curl` from a command line.

```
curl -X POST -H "Content-Type: application/json" -d '{"value1": "37.5"}'  
https://maker.ifttt.com/trigger/temperature/with/key/[REDACTED]
```

**1** 這裡請改成 Webhooks 中設定的事件名稱 **temperature**

這裡是上面看到的 key

**2** value1 輸入 **37.5**

**3** HTTP 請求網址，請先複製下來

**4** 按此鈕測試

Test It

# 手動請求發出 LINE 訊息：

5 打開 LINE 選擇 **LINE Notify**



6 收到 IFTTT 傳來的通知

# LAB07 防疫作戰：體溫推播系統

- 實驗目的：  
LAB05 體溫超過 37.5 °C時傳送 LINE 訊息
- 材料、線路圖：同 LAB02
- 開發環境：Thonny

# 設計原理

- 用 GET 方法來傳送 LINE 訊息。  
GET 方法就是剛從 IFTTT 複製的網址：

Thonny

```
urequests.get("http://maker.ifttt.com/trigger/temperature/  
with/key/您的 key")
```

# 設計原理

- 如果直接執行上述的程式碼，會看到：



```
http://maker.ifttt.com/trigger/temperature/with/key/您的  
key?value1=37.8
```

Thonny

```
urequests.get("http://maker.ifttt.com/trigger/temperature/  
with/key/您的 key?value1=37.8")
```

# 程式設計

**LAB07.py****防疫作戰：體溫推播系統****Thonny**

```
24 # LINE (請記得更改為 http)
25 url_line="請填入 IFTTT 複製的網址 (請記得將 https 更改為 http)"
...
60     if (tem>=37.5): # 當體溫超過 37.5 度時, 傳 LINE 做警告
61         urequests.get(url_line+"?value1="+str(tem))
62         print("警告!!!發燒了!!!")
```

# 測試程式

```
60     if (tem>=10.5)
61         urequests.get
62         print("警告!!!")
63
64     data=0
65     time.sleep(60)
66
67
```

1 發燒值調低成 10.5

2 24.7>10.5, 所以顯示警告!!! 發燒了!!!

```
互動環境(Shell) ×
MicroPython v1.12-195-gb1e
Type "help()" for more info
>>> %Run -c $EDITOR_CONTENT
I (1073895) phy: phy_ver
Wifi連線成功
24.7 上傳完畢
警告!!!發燒了!!!
```