

# 物聯網感測元件實務

REST API

王昱景 Brian Wang

[brian.wang.frontline@gmail.com](mailto:brian.wang.frontline@gmail.com)

# 全球資訊網

- 全名為 World Wide Web
- 簡稱為 WWW 或是 Web
- 是一個由許多互相連結的超文字組成的系統，通過網際網路存取

- 1980 年英國科學家提姆·柏內茲-李於 CERN（歐洲核子研究組織）編寫 ENQUIRE 軟體計劃，它是全球資訊網的前身
- 1989 年 3 月撰寫了“關於資訊化管理的建議”一文，文中提及 ENQUIRE 並且描述了一個更加精巧的管理模型
- 1990 年 11 月 12 日和羅伯特·卡里奧合作提出了一個更加正式的關於全球資訊網的建議

- 1990 年 11 月 13 日在一台 NeXT 工作站上寫了第一個網頁實現想法
- 1990 年的聖誕節製作了第一個網頁瀏覽器 WorldWideWeb（同時也是網頁編輯器）和第一個網頁伺服器

<http://info.cern.ch/hypertext/WWW/TheProject.html>

- 1991 年 8 月 6 日全球資訊網公共服務的首次亮相
- 1993 年 4 月 30 日，歐洲核子研究組織宣布全球資訊網對任何人免費開放
- 1994 年 10 月全球資訊網協會（W3C）在麻省理工學院電腦科學實驗室成立

# HTTP

- 超文本傳輸協定，HyperText Transfer Protocol
- 一種用於分佈式、協作式和超媒體資訊系統的應用層協議
- 全球資訊網的資料通訊的基礎
- HTTP 是一個用戶端終端（用戶）和伺服器端（網站）請求和應答的標準（TCP）

# 用戶端請求

```
GET / HTTP/1.1  
Host: www.google.com
```

# 伺服器應答

```
HTTP/1.1 200 OK  
Content-Length: 3059  
Server: GWS/2.0  
Date: Sat, 11 Jan 2003 02:44:04 GMT  
Content-Type: text/html  
Cache-control: private  
Set-Cookie: PREF=ID=73d4aef52e57bae9:TM=1042253044:LM=1042253044:S=SMCc_HRPCQiqy  
X9j; expires=Sun, 17-Jan-2038 19:14:07 GMT; path=/; domain=.google.com  
Connection: keep-alive
```

- 請求方法
  - GET
    - 向指定的資源發出「顯示」請求
  - HEAD
    - 與GET方法一樣，都是向伺服器發出指定資源的請求，只不過伺服器將不傳回資源的本文部份



- POST

- 向指定資源提交資料，請求伺服器進行處理（例如提交表單或者上傳檔案），資料被包含在請求本文中

- PUT

- 向指定資源位置上傳其最新內容

- DELETE
  - 請求伺服器刪除 Request-URI 所標識的資源
- TRACE
  - 回顯伺服器收到的請求，主要用於測試或診斷

- OPTIONS

- 這個方法可使伺服器傳回該資源所支援的所有 HTTP 請求方法

- CONNECT

- HTTP/1.1 協定中預留給能夠將連線改為管道方式的代理伺服器

- 狀態碼

- 1xx 訊息—請求已被伺服器接收，繼續處理
- 2xx 成功—請求已成功被伺服器接收、理解、並接受
- 3xx 重新導向—需要後續操作才能完成這一請求
- 4xx 請求錯誤—請求含有詞法錯誤或者無法被執行
- 5xx 伺服器錯誤—伺服器在處理某個正確請求時發生錯誤

# HTTPS

- 超文字傳輸安全協定，Hypertext Transfer Protocol Secure
- 常稱為 HTTP over TLS，HTTP over SSL 或 HTTP Secure
- HTTPS 經由 HTTP 進行通訊，但利用 SSL/TLS 來加密封包

# URL

- 統一資源定位符、統一資源定位器/定位位址或URL位址等，Uniform Resource Locator
- 俗稱為網頁位址、網址
- 在網路上的門牌，是網際網路上標準的資源的位址 (Address)

協定類型:[//[存取資源需要的憑證資訊@]伺服器位址[:埠號]][/資源層級UNIX檔案路徑]檔案名[?查詢][#片段ID]

# Web service

- Web 服務是一種服務導向架構的技術，透過標準的 Web 協議提供服務，目的是保證不同平台的應用服務可以互操作
- 網絡服務通常是許多應用程式接口（API）所組成的

```
<?xml version="1.0"?>
<quiz>
  <qanda seq="1">
    <question>
      Who was the forty-second
      president of the U.S.A.?
    </question>
    <answer>
      William Jefferson Clinton
    </answer>
  </qanda>
  <!-- Note: We need to add
  more questions later.-->
</quiz>
```

**XML**

可延伸標記式語言，Extensible Markup Language，一種標記式語言



# API

- 應用程式介面，Application Programming Interface
- 目的是讓開發人員以呼叫一組常式功能而無須考慮其底層的原始碼為何、或理解其內部工作機制的細節
- API 本身是抽象的，僅定義了一個介面而不涉及應用程式實際實現過程中的具體操作

# REST

- 表示狀態傳送，Representational State Transfer
- Roy Thomas Fielding 博士於 2000 年博士論文中提出來的一種全球資訊網軟體架構風格，目的是便於不同軟體/程式在網路中互相傳遞資訊
- 是設計風格而不是標準

- 資源是由 URI 來指定
- 對資源的操作包括取得、建立、修改和刪除資源，正好對應 HTTP 協定提供的 GET、POST、PUT 和 DELETE 方法。
- 通過操作資源的表現形式來操作資源
- 資源的表現形式則是 XML 或者 HTML，取決於讀者是機器還是人，當然也可以的任何其他的格式

- 符合具象狀態傳輸設計風格的 Web API 稱為 RESTful API
- 直觀簡短的資源位址：URI，比如：`http://example.com/resources/`
- 傳輸的資源：Web 服務接受與返回的網際網路媒體類型，比如：JSON，XML，YAML等
- 對資源的操作：Web 服務在該資源上所支援的一系列請求方法（比如：POST，GET，PUT 或 DELETE）

資源	GET	PUT	POST	DELETE
一組資源的URI，比如 <a href="https://example.com/resources/">https://example.com/resources/</a>	列出URI，以及該資源組中每個資源的詳細資訊（後者可選）。	使用給定的一組資源替換當前整組資源。	在本組資源中 <b>建立/追加</b> 一個新的資源。該操作往往返回新資源的URL。	<b>刪除</b> 整組資源。
單個資源的URI，比如 <a href="https://example.com/resources/142">https://example.com/resources/142</a>	取得指定的資源的詳細資訊，格式可以自選一個合適的網路媒體類型（比如：XML、JSON等）	<b>替換/建立</b> 指定的資源。並將其追加到相應的資源組中。	把指定的資源當做一個資源組，並在其下 <b>建立/追加</b> 一個新的元素，使其隸屬於當前資源。	<b>刪除</b> 指定的元素。

# JSON

- JavaScript Object Notation
- 由道格拉斯·克羅克福特構想和設計、輕量級的資料交換語言
- 以易於讓人閱讀的文字為基礎，用來傳輸由屬性值或者序列性的值組成的資料物件

```
1 {
2   "firstName": "John",
3   "lastName": "Smith",
4   "sex": "male",
5   "age": 25,
6   "address":
7   {
8     "streetAddress": "21 2nd Street",
9     "city": "New York",
10    "state": "NY",
11    "postalCode": "10021"
12  },
13  "phoneNumber":
14  [
15    {
16      "type": "home",
17      "number": "212 555-1234"
18    },
19    {
20      "type": "fax",
21      "number": "646 555-4567"
22    }
23  ]
24 }
```

# 雲端氣象站

Arduino + pushingbox + Google Forms

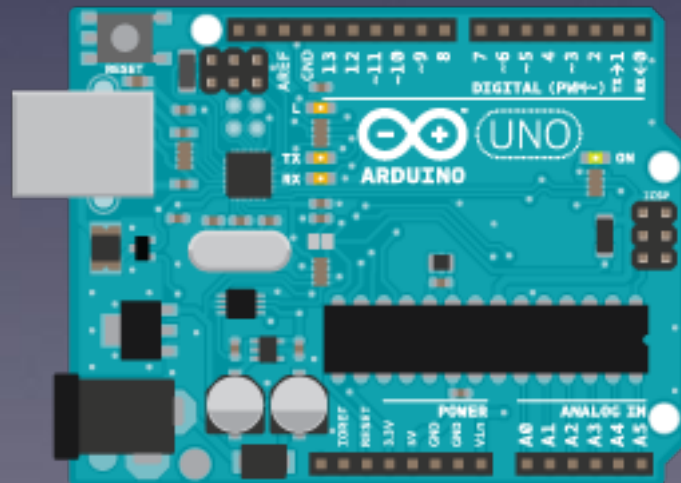




HTTPS



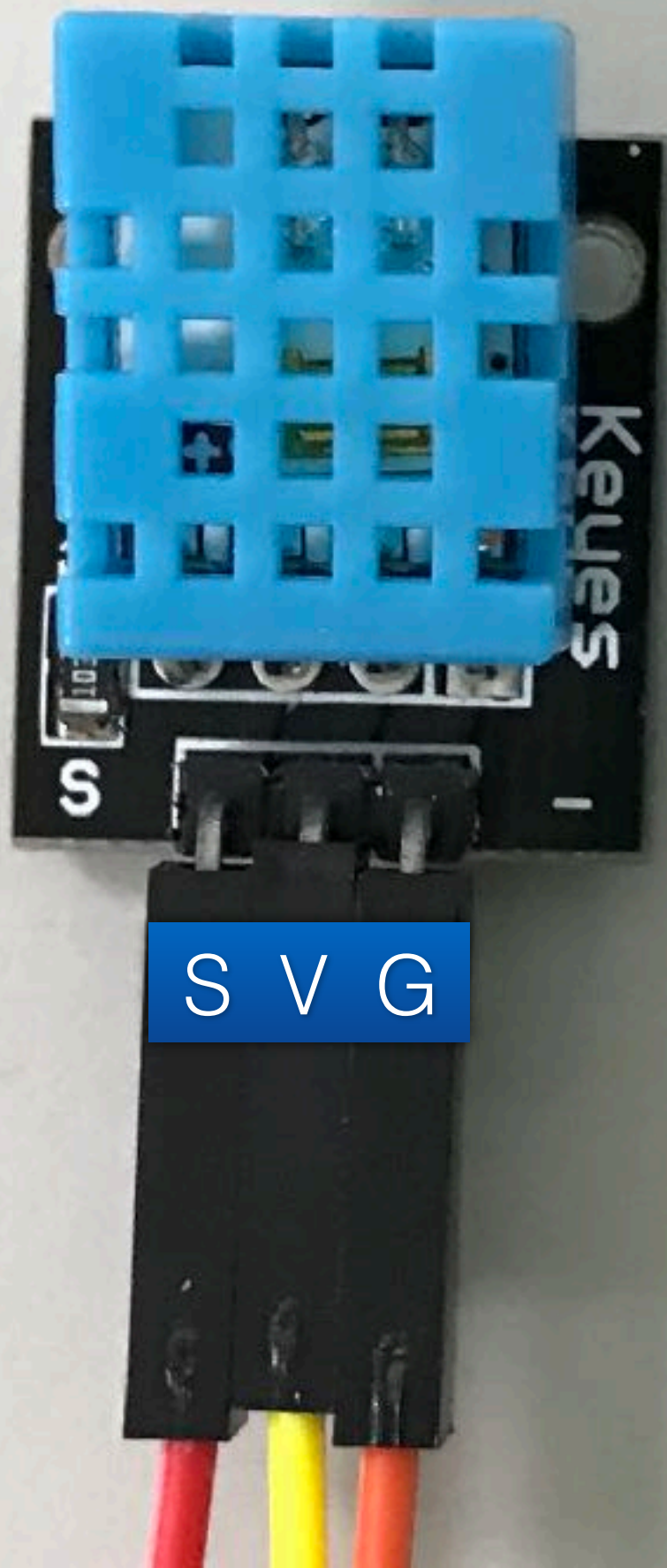
HTTPS



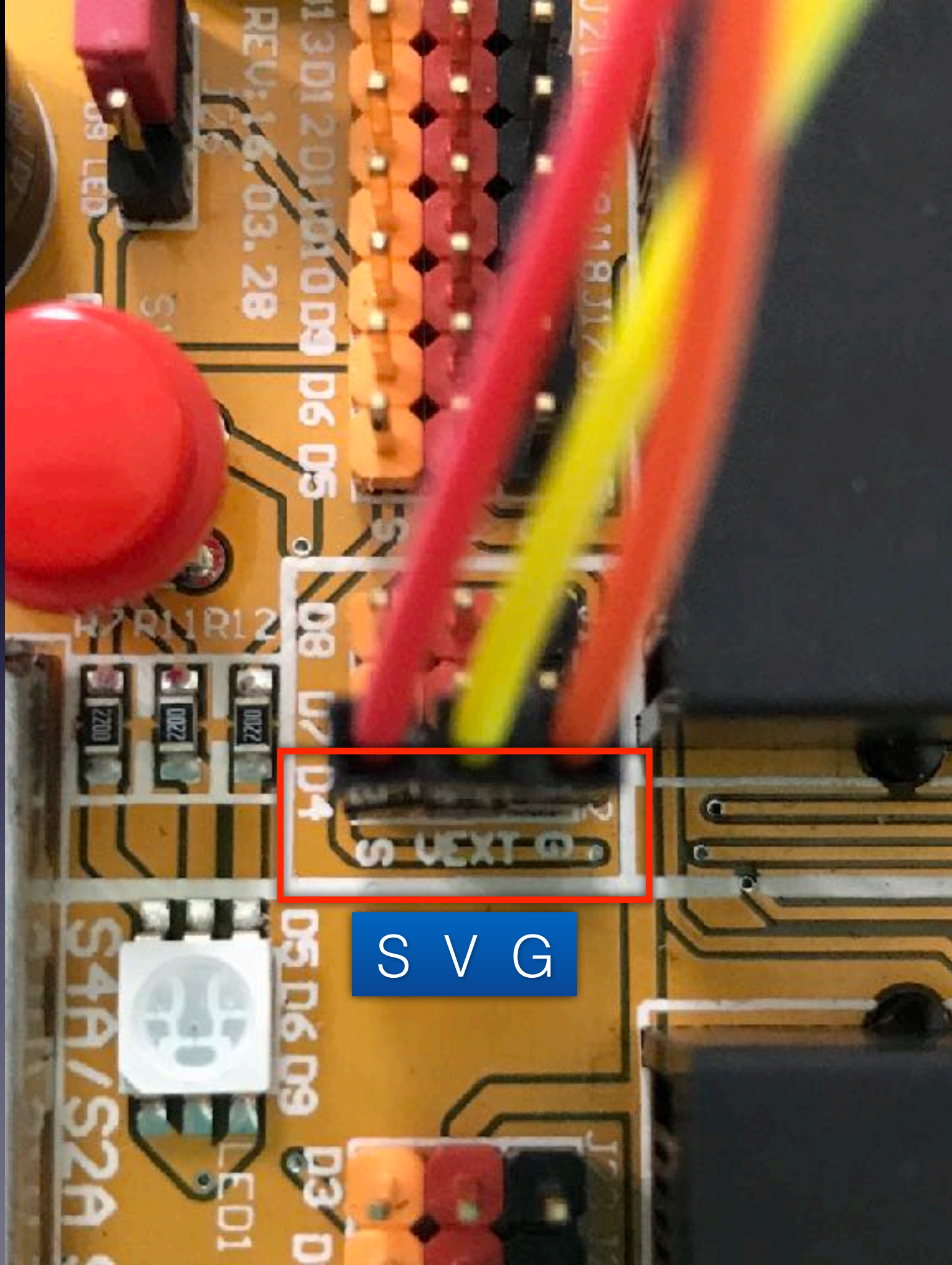
HTTP



# Arduino 與溫濕度感測器







# Google Forms 設定



A close-up photograph of a person's hands holding a s'more. The s'more consists of two graham crackers sandwiching a piece of melted chocolate and a marshmallow. The background is a blurred campfire with flames and a blue tent. The overall tone is warm and inviting.

# 建立精美的表單

Tom

Google 表單可以幫您收集及整理資料，無論資料多寡都能幫您輕鬆搞定，而且完全免費。

前往 Google 表單



[http://www.google.com/intl/zh-TW\\_tw/forms/about/](http://www.google.com/intl/zh-TW_tw/forms/about/)



最近的表單

不限歸屬 ▾



AZ



目前沒有任何表單

按一下 [+] 即可建立新表單。





傳送



問題

回覆

# Arduino 溫濕度監控 **1**

表單說明

**3**

溫度

簡答文字

**4**

數字



數字



自訂錯誤訊息



= 簡答



**2**



必填



**5**







傳送



問題

回覆

表單說明

溫度 \*

簡答文字

2

濕度

簡答文字

已新增回應驗證。 移除

3

數字

數字

自訂錯誤訊息

= 簡答

1

必填

4



## Arduino 溫濕度監控

\*必填

溫度 \*

您的回答

濕度 \*

您的回答

提交

# 填寫測試取得URL

[https://docs.google.com/forms/d/e/1FAIpQLSfr71wA7TGNL0kl96Sckr\\_ZRv6UacDFY0lpI0DiBuWzJo8qw/formResponse](https://docs.google.com/forms/d/e/1FAIpQLSfr71wA7TGNL0kl96Sckr_ZRv6UacDFY0lpI0DiBuWzJo8qw/formResponse)



問題

回覆

95

## Arduino 溫濕度監控

表單說明

溫度

簡答文字

濕度

簡答文字

復原

建立複本

移至垃圾桶

取得預先填入的連結

列印

新增協作者...

指令碼編輯器

外掛程式...

偏好設定...

## Arduino 溫濕度監控

溫度

1

濕度

2

取得連結

3

請勿利用 Google 表單送出密碼。

共用這個連結即可加入預先填妥的回應內容

複製連結

4

Google 並未認可或建立這項內容。檢舉濫用情形 · 服務條款 · 其他條款

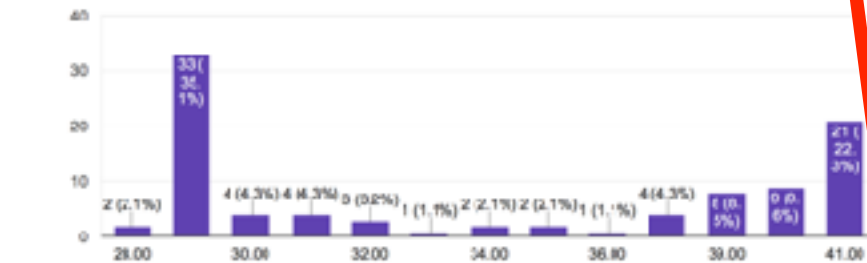
[https://docs.google.com/forms/d/e/1FAIpQLSfrr71wA7TGNL0kI96Sckr\\_ZRv6UacDFY0IpI0DiBuWzJo8qw/viewform?usp=pp\\_url&entry.192329928=0&entry.1478584319=0](https://docs.google.com/forms/d/e/1FAIpQLSfrr71wA7TGNL0kI96Sckr_ZRv6UacDFY0IpI0DiBuWzJo8qw/viewform?usp=pp_url&entry.192329928=0&entry.1478584319=0)

94 則回應

摘要 個別

溫度

94 則回應



濕度

94 則回應

29

Arduino 溫濕度監控 (回應)

檔案 編輯 檢視 插入 格式 資料 工具 表單 外掛程式 說明 上次編輯是在 20 分鐘前

時間戳記	溫度	濕度
2018/4/3 上午 2:58:05	29	30
2018/4/3 上午 2:58:57	29	40
2018/4/3 上午 2:59:09	29	40
2018/4/3 上午 2:59:20	29	40
2018/4/3 上午 2:59:32	29	40
2018/4/3 上午 2:59:44	29	40
2018/4/3 上午 2:59:56	29	40
2018/4/3 上午 3:00:07	29	40
2018/4/3 上午 3:00:19	29	40
2018/4/3 上午 3:00:31	29	40
2018/4/3 上午 3:00:43	29	40
2018/4/3 上午 3:00:54	29	40
2018/4/3 上午 3:01:06	29	40
2018/4/3 上午 3:01:17	29	40
2018/4/3 上午 3:01:29	28	41
2018/4/3 上午 3:01:40	29	40
2018/4/3 上午 3:01:51	29	40
2018/4/3 上午 3:02:03	29	40
2018/4/3 上午 3:02:14	29	40
2018/4/3 上午 3:02:41	29	40
2018/4/3 上午 3:02:53	29	40
2018/4/3 上午 3:03:05	29	40
2018/4/3 上午 3:03:16	29	40
2018/4/3 上午 3:03:28	29	40
2018/4/3 上午 3:03:40	32	39
2018/4/3 上午 3:03:53	34	39
2018/4/3 上午 3:04:05	36	37
2018/4/3 上午 3:04:14	38	36
2018/4/3 上午 3:04:25	38	36
2018/4/3 上午 3:04:37	39	36
2018/4/3 上午 3:04:49	39	36
2018/4/3 上午 3:05:01	39	36

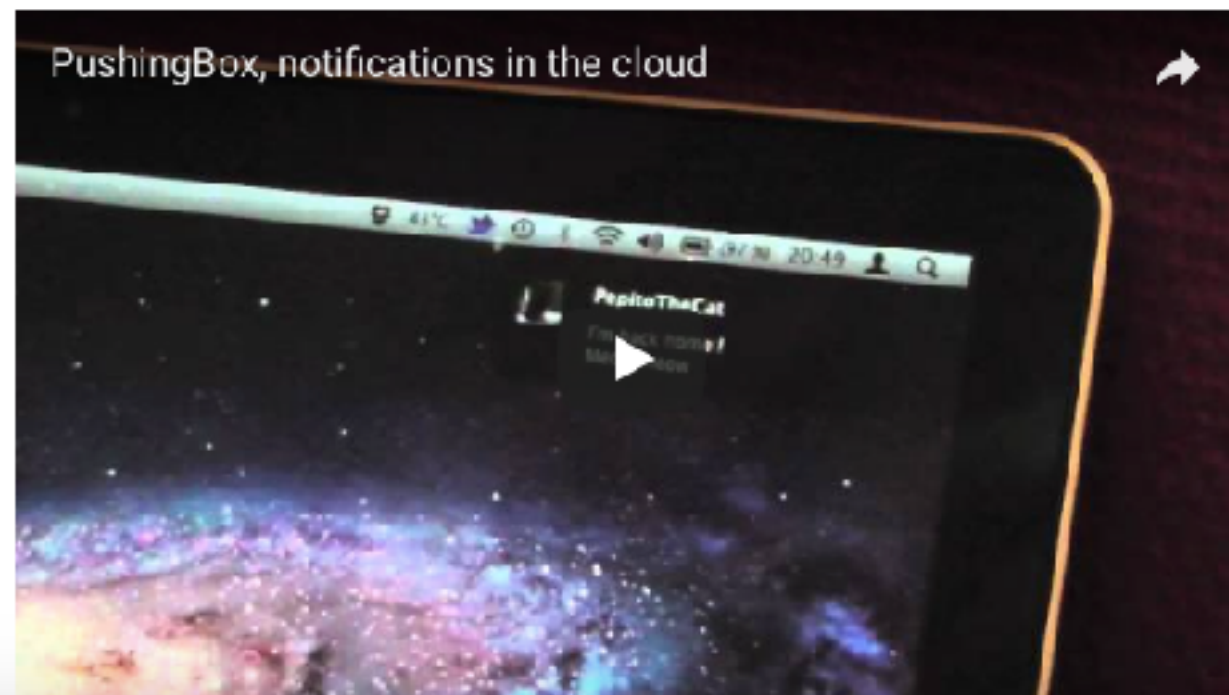
pushingbox 設定

## PushingBox, Notifications for your Internet of Things devices

### One API for all your notifications.

PushingBox is a cloud that can send notifications based on API calls.

From one request, you can send several notifications like a Push, a Tweet, an Email... All this in real time.



<https://www.pushingbox.com/index.php>

## Services

Add a service

Here you can select one of the various services that are available on PushingBox.

Add a service

## Your services

Here you can find the services you have subscribed to.



Email Brian Wang

brian.wang.frontline@gmail.com

Edit

Delete



## Services

### Add a service

Here you can select one of the various services that are available on PushingBox.

**Email**

Just a more email service for being notified by email

[Select this service](#)**Twitter**

Tweet notification

[Select this service](#)**Karotz**

Voice notification (tts), make your Karotz speaking on an event

[Select this service](#)**Prowl**

Push notification service for iPhone

[Select this service](#)**Pushalot**

Push notification service for Windows 8 and Windows Phone

[Select this service](#)**Tostty**

Push notification service for Windows Phone

[Select this service](#)**Notify My Android**

Push notification service for Android

[Select this service](#)**Newtify**

Push notification service for Android

[Select this service](#)**Pushbullet**

Push notification service for iOS, Android, Windows, Firefox and Chrome

[Select this service](#)**CustomURL**

Set your own service!

[Select this service](#)

### Your services


Here you can find the services you have subscribed to.





Email Brian Wang


brian.wang.frontline@gmail.com


[Edit](#)[Delete](#)

**toasty**  
Push notification service for Windows Phone

**Notify My Android**  
Push notification service for Android

**Newtify**  
Push notification service for iOS

**Pushbullet**  
Push notification service for Android and iOS

**CustomURL**  
Set your own service URL


Select this service

Select this service

Select this service

Select this service

Select this service

**CustomURL Service**

CustomURL let you interface with any API of any product.  
Enter the Root URL of the service you want.

Name of your CustomURL configuration:

Root URL:


Method:

Cancel

Submit

Your services

Here you can find the services you have subscribed to.



Email Brian Wang

brian.wang.frontline@gmail.com

Edit

Delete

# Scenarios

1

Create a scenario or add a device

Enter the name of your new Scenario or your Notifon's DeviceID (starting with "h")

2

ArduinoToGoogleForm

Add

3

## Your virtual scenarios

Here you can find all your scenarios previously created. Click on Manage to add actions.

Scenario name

DeviceID

Demo scenario

v1FC48B6984CD76C

Test

Manage

Delete

## Scenarios

ArduinoToGoogleForm [Rename](#)

DeviceID: v34D1138D151C5FB

Deactivate

Test Scenario

Well done, this is your scenario, now clic on Add an Action...

Back

Add an Action



## Scenario

Add an action for the scenario *ArduinoToGoogleForm*



Email Brian Wang

Add an action with this service



ArduinoToGoogleForm

Add an action with this service

Back

## Scenarios

## ArduinoToGoogleForm

DeviceID: v34D1138D151C5EB



ArduinoToGoogleForm

Back

Add an Action

Deactivate

Test Scenario

Edit

Delete

Privacy

Blog



ArduinoToGoogleForm



Enter the data you need to send.

Example for GET method: ?token=12345&amp;message=Hello

Example for POST method: token=12345&amp;message=Hello

Data

1

Cancel

Update

2

?entry.192329928=\$temp\$&entry.1478584319=\$hum\$&submit=Submit

## Scenarios

### ArduinoToGoogleForm [Rename](#)

[Deactivate](#)[Test Scenario](#)

DeviceID: v34D1138D151C5FB



ArduinoToGoogleForm

**Data:** ?entry.192328928=\$temp\$&entry.1478584319=\$hum\$&submit=Submit[Edit](#)[Delete](#)[Back](#)[Add an Action](#)



## Scenarios

Create a scenario or add a device

Enter the name of your new Scenario or your Notifon's DeviceID (starting with "h")

"New mail in the mailbox" or "h1234567890ABCDE"

Add

### Your virtual scenarios

Here you can find all your scenarios previously created. Click on Manage to add actions.

Scenario name	DeviceID	
ArduinoToGocgleForm	v34D1138D151C5EB	<a href="#">Test</a> <a href="#">Manage</a> <a href="#">Delete</a>
Demo scenario	v1FC48B6904CD76C	<a href="#">Test</a> <a href="#">Manage</a> <a href="#">Delete</a>



motoBlockly 積木邏輯

## 設定

設定串列埠 serial 傳輸率 9600 bps



ESP8266 設定 Ver 0.40

WiFi 模式 STATION

串列輸出腳位 A5

串列輸入腳位 A4

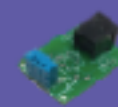
SSID(分享器名稱) " WiFi "

Password(密碼) " 0928583777 "

印出訊息後換行 取得裝置IP位址

## 迴圈

宣告 temp 當 String 資料



DHT11溫溼度感測器 腳位 4 讀取數值 溫度

宣告 hum 當 String 資料



DHT11溫溼度感測器 腳位 4 讀取數值 濕度

如果



連線到遠端伺服器(Server)

≠ 0

伺服器位置(Address) " api.pushingbox.com "

伺服器連接埠(Port) 80

執行



WiFi發送資料到所指定網址

WiFi: 發送資料 " "

停止遠端連線

印出訊息後換行

建立字串使用 " 溫度: "

temp

" °C "

" , 濕度: "

hum

" % "

延遲毫秒 5000

```

1  /* Sketch was generated by motoblockly
2  Website: http://www.motoblockly.com
3  Author: www.motoduino.com
4  Date: Tue Apr 03 2018 02:04:02 GMT+0800
5  */
6  #include "motoWiFiEsp.h"
7
8  #include <SoftwareSerial.h>
9
10 #include <motoDHT.h>
11
12 SoftwareSerial esp8266_Serial(A5,A4);
13
14 WiFiEspClient esp_client;
15
16 int connect_status = WL_IDLE_STATUS;
17
18 String ipAddressToString(const IPAddress& ipAddress ) {
19     return String(ipAddress[0]) + String(".") +
20     String(ipAddress[1]) + String(".") +
21     String(ipAddress[2]) + String(".") +
22     String(ipAddress[3]);
23 }
24 DHT motoDHT(4, DHT11);
25
26 String temp;
27 String hun;
28 void setup()
29 {
30     Serial.begin(9600);
31     esp8266_Serial.begin(9600);
32
33     pinMode(4, INPUT);
34     motoDHT.begin();
35     esp8266_Serial.listen();
36     WiFi.init(&esp8266_Serial);
37     if(WiFi.status()==WL_NO_SHIELD) {
38         Serial.println("Esp8266 module no present");
39         while(true);
40     }
41     while(connect_status != WL_CONNECTED) {
42         Serial.println("Connect to router...");
43         connect_status = WiFi.begin("WiFi","0928583777");
44     }
45     Serial.println(ipAddressToString(WiFi.localIP()));
46
47 }
48
49
50 void loop()
51 {
52     temp = motoDHT.readTemperature();
53     hun = motoDHT.readHumidity();
54     if (esp_client.connect("api.pushingbox.com",80) != 0) {
55         String restapi_request = "";
56         esp_client.print(restapi_request);
57         esp_client.stop();
58         Serial.println((String("温度:") + String(temp) + String("℃") + String("。湿度:") + String(hun) + String("%")));
59         delay(5000);
60     }
61 }
62
63 }

```

# Arduino 程式編輯燒錄

```
1  /* Sketch was generated by motoblockly
2  Website: http://www.motoblockly.com
3  Author: www.motoduino.com
4  Date: Tue Apr 03 2018 02:04:02 GMT+0800
5  */
6  #include "motoWiFiEsp.h"
7
8  #include <SoftwareSerial.h>
9
10 #include <motoDHT.h>
11
12 // Your secret DevID from PushingBox.com.
13 char DEVID1[] = "v34D1138D151C5EB";
14 char serverName[] = "api.pushingbox.com";
15
16 String temp;
17 String hum;
18
19 SoftwareSerial esp8266_Serial(A5,A4);
20
21 WiFiEspClient esp_client;
22
23 int connect_status = WL_IDLE_STATUS;
24
25 String ipAddressToString(const IPAddress& ipAddress ) {
26     return String(ipAddress[0]) + String(".") +
27     String(ipAddress[1]) + String(".") +
28     String(ipAddress[2]) + String(".") +
29     String(ipAddress[3]);
30 }
31 DHT motoDHT(4, DHT11);
32
```

```
32
33 void setup()
34 {
35     Serial.begin(9600);
36     esp8266_Serial.begin(9600);
37
38     pinMode(4, INPUT);
39     motoDHT.begin();
40     esp8266_Serial.listen();
41     WiFi.init(&esp8266_Serial);
42     if(WiFi.status()==WL_NO_SHIELD) {
43         Serial.println("Esp8266 module no present");
44         while(true);
45     }
46     while(connect_status != WL_CONNECTED) {
47         Serial.println("Connect to router...");
48         connect_status = WiFi.begin("WiFi","0928583777");
49     }
50     Serial.println(ipAddressToString(WiFi.localIP()));
51
52 }
53
54
55 void loop()
56 {
57     temp = motoDHT.readTemperature();
58     hum = motoDHT.readHumidity();
59     if (esp_client.connect("api.pushingbox.com",80) != 0) {
60         esp_client.print("GET /pushingbox?devid=");
61         esp_client.print(DEVID1);
62         esp_client.print("&temp=");
63         esp_client.print(String(temp));
64         esp_client.print("&hum=");
65         esp_client.print(String(hum));
66         esp_client.println(" HTTP/1.1");
67         esp_client.print("Host: ");
68         esp_client.println(serverName);
69         esp_client.println("User-Agent: Arduino");
70         esp_client.println();
71         esp_client.stop();
72         Serial.println((String("溫度:") + String(temp) + String("(") + String(", 濕度:") + String(hum) + String("%)"));
73         delay(5000);
74     }
75
76
77 }
```

測試





# Arduino 溫濕度監控 (回應) ☆

檔案 編輯 檢視 插入 格式 資料 工具 表單 外掛程式 說明 所有變更都已儲存到雲端硬碟



共用



98% NT\$ % .0 .00 123 Arial 10 B I U A

fx 2018/4/3 上午 3:10:30

	A	B	C	D	E	F	G	H
1	時間戳記	溫度	濕度					
2	2018/4/3 上午 2:58:46	29	39					
3	2018/4/3 上午 2:58:57	29	40					
4	2018/4/3 上午 2:59:09	29	40					
5	2018/4/3 上午 2:59:20	29	40					
6	2018/4/3 上午 2:59:32	29	40					
7	2018/4/3 上午 2:59:44	29	40					
8	2018/4/3 上午 2:59:56	29	40					
9	2018/4/3 上午 3:00:07	29	40					
10	2018/4/3 上午 3:00:19	29	40					
11	2018/4/3 上午 3:00:31	29	40					
12	2018/4/3 上午 3:00:43	29	40					
13	2018/4/3 上午 3:00:54	29	40					
14	2018/4/3 上午 3:01:06	29	40					
15	2018/4/3 上午 3:01:17	29	40					
16	2018/4/3 上午 3:01:29	28	41					
17	2018/4/3 上午 3:01:40	29	40					
18	2018/4/3 上午 3:01:51	29	40					
19	2018/4/3 上午 3:02:03	29	40					
20	2018/4/3 上午 3:02:14	29	40					
21	2018/4/3 上午 3:02:41	29	40					
22	2018/4/3 上午 3:02:52	29	40					

+ 三

表單回應1 工作表2



進階即時圖表顯示



# Arduino 溫濕度監控 (回應)



檔案 編輯 檢視 插入 格式 資料 工具 表單 外掛程式 說明 所有變更都已儲存到雲端硬碟



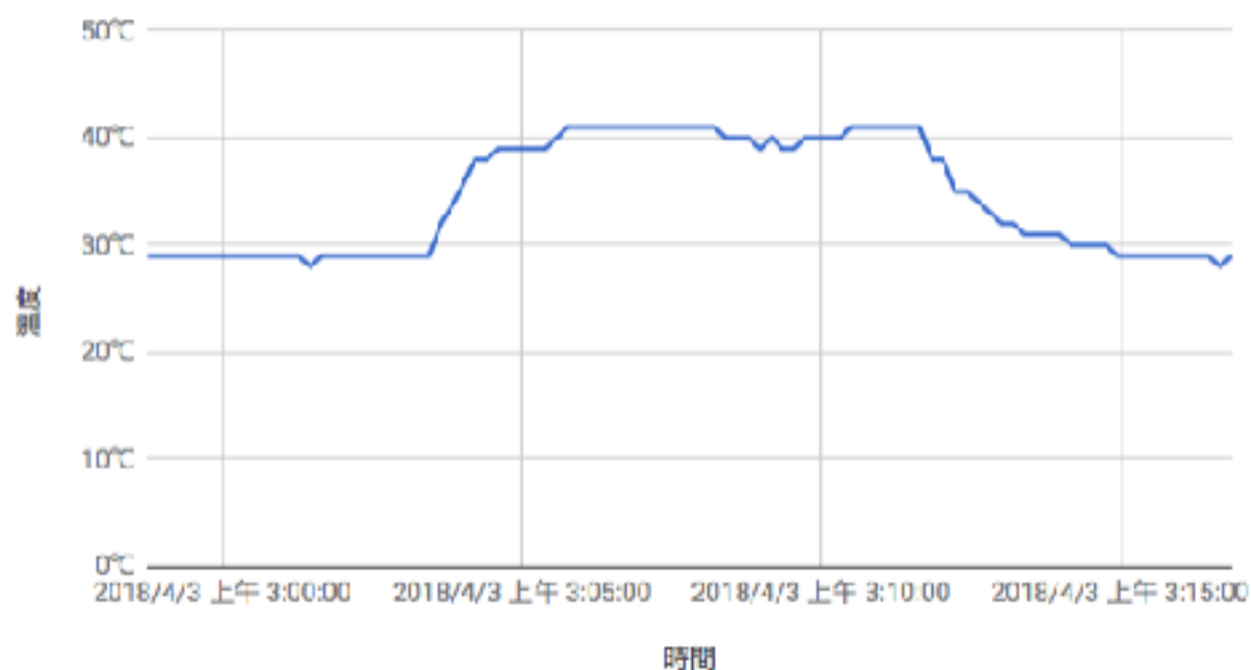
共用



98% NT\$ % .0 .00 123 Arial 10 B I U A

fx

溫度



濕度



+

≡

表單回應1 工作表2

