

****

7103056201 蘇茂銓

7104056022 楊東祐

7104056043 白書珉

目錄

[GasWeight 2](#_Toc440464409)

[使用情境 2](#_Toc440464410)

[專題檔案連結 2](#_Toc440464411)

[流程圖 2](#_Toc440464412)

[硬體材料需求 3](#_Toc440464413)

[環境架設 3](#_Toc440464414)

[Arduino 3](#_Toc440464415)

[接線 3](#_Toc440464416)

[WiFi模組 3](#_Toc440464417)

[電子秤套件 4](#_Toc440464418)

[線路圖 5](#_Toc440464419)

[程式碼 6](#_Toc440464420)

[基本設定和函式 6](#_Toc440464421)

[Wifi設定 6](#_Toc440464422)

[電子秤-抓值 7](#_Toc440464423)

[傳到資料庫 8](#_Toc440464424)

[Web 11](#_Toc440464425)

[環境 11](#_Toc440464426)

[資料庫 11](#_Toc440464427)

[資料庫設定 11](#_Toc440464428)

[PHP 11](#_Toc440464429)

[使用API 15](#_Toc440464430)

[頁面 16](#_Toc440464431)

[主頁面 16](#_Toc440464432)

[使用量表 20](#_Toc440464433)

[尋找瓦斯行 26](#_Toc440464434)

[設定/連絡瓦斯行 31](#_Toc440464435)

[導航功能 34](#_Toc440464436)

# GasWeight

## 使用情境

　　不管是餐飲店家或家庭，常常要等到瓦斯用完了才知道瓦斯沒了，店家通常會預留好一兩桶瓦斯當作預備，但瓦斯桶本身就占空間，所以如果我們可以提早告知瓦斯快耗盡的訊息，就可以避免沒瓦斯用的情況發生，也不用在預留瓦斯。

## 專題檔案連結

GitHub - [https://github.com/Brownies1201/iot\_gas\_weight](%20https:/github.com/Brownies1201/iot_gas_weight)

# 流程圖

Database(server)

首頁

使用量表

尋找瓦斯行

設定/連絡瓦斯行

電子秤秤重



Arduino取值

WiFi

設定/連絡瓦斯行

將資料送到資料庫

Arduino

Reply

Request

Web(client)

# 硬體材料需求

* Arduino UNO \*1
* 麵包板 \*1
* 10 k ohm電阻 \*1
* 5KG電子秤套件 \*1
* WiFi模組 - ESP8266 \*1
* 5V轉3.3V穩壓器 \*1
* 杜邦線(M/M), (F/M) 數條

# 環境架設

* 安裝XAMPP - <https://www.apachefriends.org/zh_tw/index.html>

# Arduino

偵測瓦斯桶的重量，並且將值傳到資料庫

## 接線

### WiFi模組

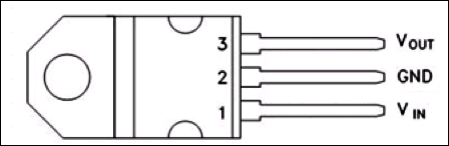
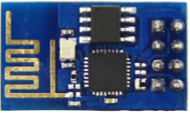
1 – 5V

2 – 3.3V

3 - GND

LD-1117

ESP8266



UTXD

CH\_PD

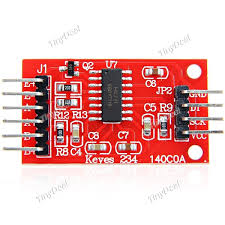
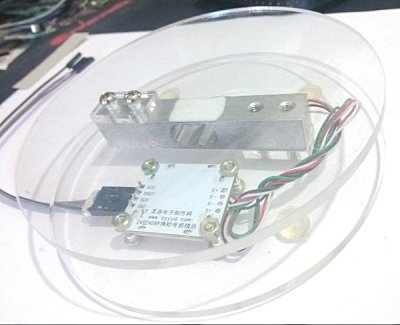
RST

VCC

GND  
GPIO2  
GPIO0  
URXD

|  |  |  |
| --- | --- | --- |
| **ESP8266** | **Arduino** | **LD-1117** |
| **UTXD** | **RX** |  |
| **URXD** | **TX** |  |
|  | **5V** | **V\_in** |
|  | **GND** | **GND** |
| **CH\_PD** |  | **V\_out** |
| **VCC** |  | **V\_out** |
| **GND** |  | **GND** |

### 電子秤套件

  
  
  
  
  
  
  
  
  
另外要準備壓克力板和支架當作放置瓦斯桶的平台

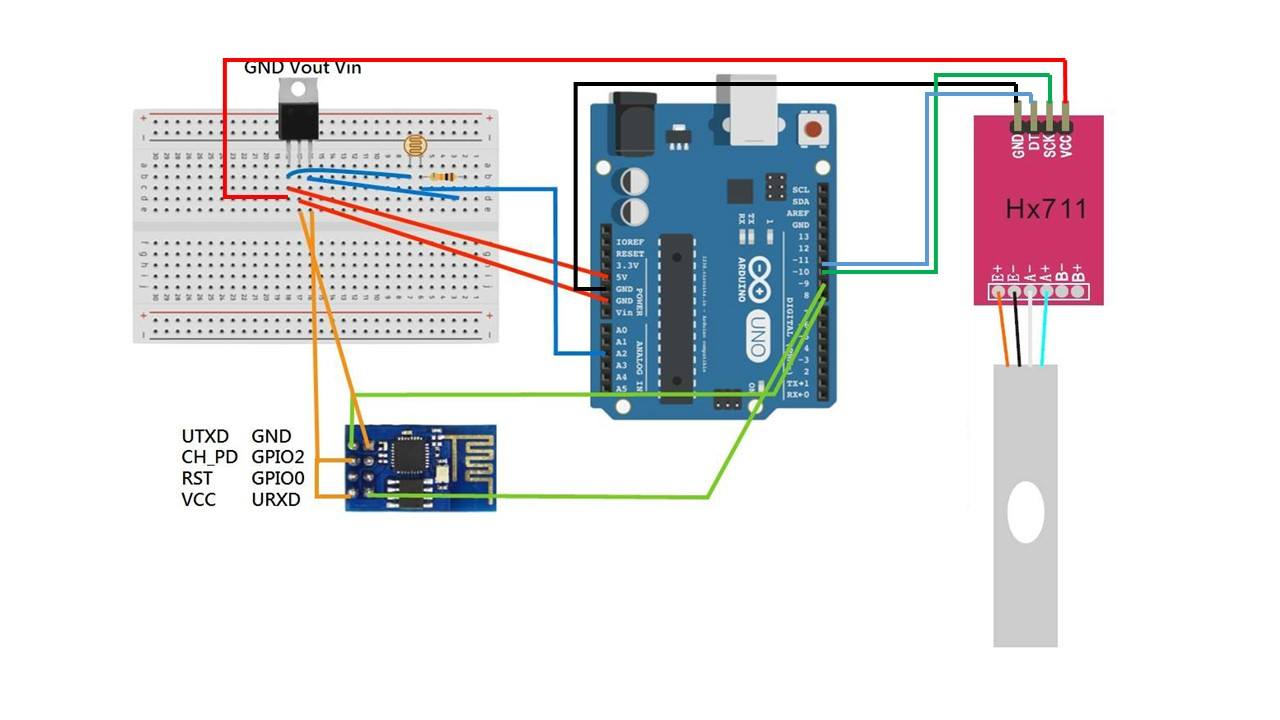
穩壓器+數位類比轉換器+時脈晶片所構成，電壓轉換與A/D轉換

由四個電阻片構成，利用壓力產生電阻改變測量重量

HX-711

懸臂平行樑

### 線路圖

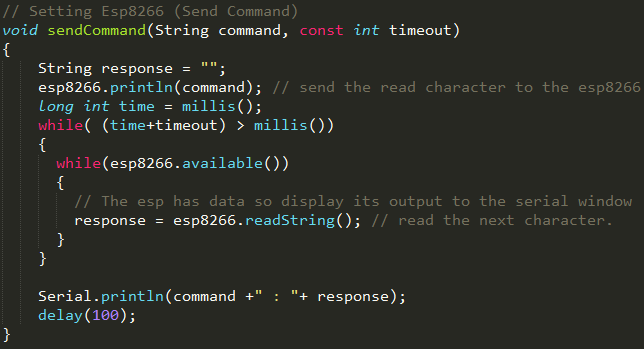


### 成果圖

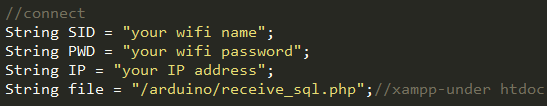
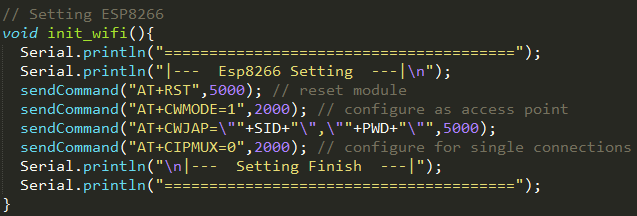


## 程式碼

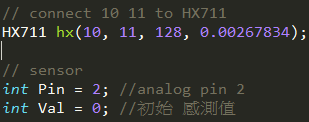
### 基本設定和函式

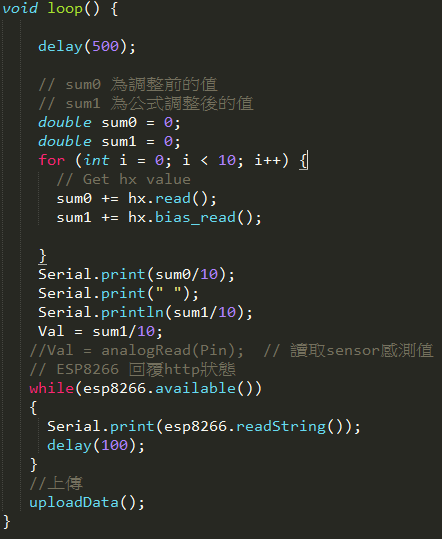
[標頭檔]  
  
[傳送指令]

### Wifi設定

[變數設定]  
SID – WiFi名稱  
PWD – WiFi密碼  
IP – DB所在的IP位址  
file – php檔案所在的路徑(XAMPP – htdoc目錄底下)  
  
  
[WiFi 初始化]  


### 電子秤-抓值

[設定]  
  
  
[公式調整基準值]  
  
[抓取感測出來的值]



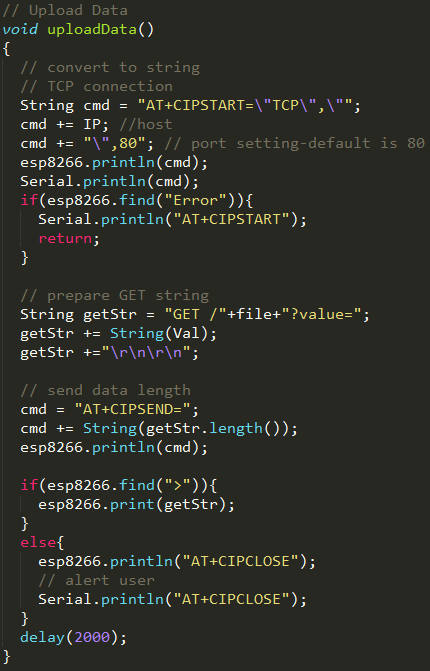
取值後還需要套公式調整

### 傳到資料庫

[HTTP request]  
在 http 通訊協定中，有兩種較為常見的方法（HTTP methods）可以將資料送到 Web Server 端，分別是GET 和 POST。(本次實作採用GET的方法)

當使用 GET 的方法時，會將表單資訊附加在 URL 上，並作為 QueryString 的一部分，QueryString 是一種 key/value 的組合，從問號「?」開始，每一組值都是用「&」隔開，例如：

**GET /receive.php?value1=10&value2=20 HTTP/1.1   
Host: your IP address**

[程式碼]

[PHP]



# Web

## 環境

安裝XAMPP - [https://www.apachefriends.org/zh\_tw/index.html](%20https:/www.apachefriends.org/zh_tw/index.html)

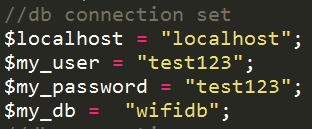
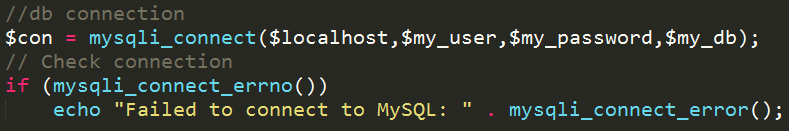
## 資料庫

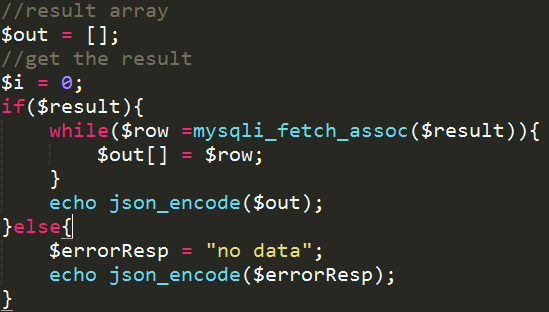
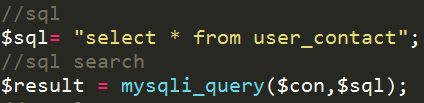
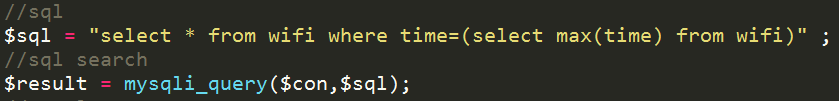
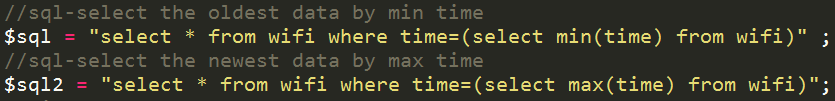
### 資料庫設定

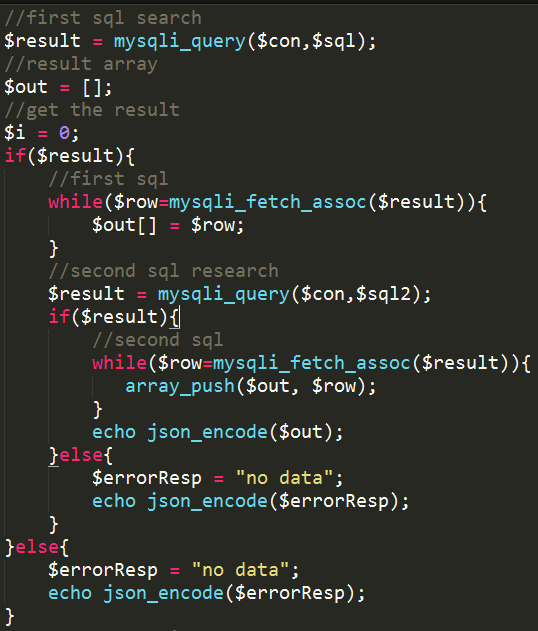
* **資料庫名稱: wifidb**
* **資料表設定**#user\_contact(id, phone, address, name)  
  [id] - primary key(AI)  
  [phone] - text  
  [address] – text  
  [name] – text  
    
  #wifi(id, data, time)  
  [id] - primary key(AI)  
  [data] – int  
  [time] - timestamp

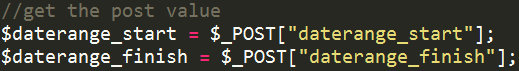
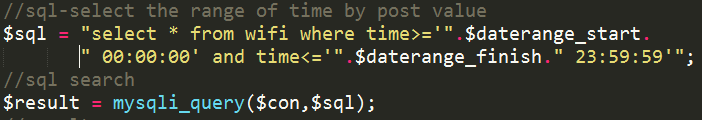
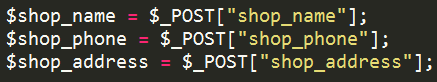
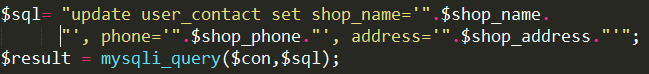
### PHP

* **連線資料庫和設定**  
  編碼方式  
    
  連線設定

  
  
連線資料庫  
  
  
設置字元編碼的資料庫連接  


* **與資料庫結束連結**
* **Sql query得結果存取在json陣列中回傳**  
  如果沒有資料或sql query失敗($result == null)，就回傳沒有資料等訊息。
* **home\_user\_contact.php**  
  sql query - 選擇user\_contact中所有的欄位的資料。
* **home\_alert.php**  
  sql query - 先從wifi table中選擇time欄位中時間最新的(max time)，然後選擇所有欄位中time和max time一樣的。
* **get\_min\_max.php**  
  query query - 要作兩次query  
  (1). 找出wifi table中time最早的所有欄位資料(min time)  
  (2). 找出wifi table中time最新的所有欄位資料(max time)  
    
  query要作兩次，將結果放在同一個陣列中，紅框為第二次query。



* **chart.php  
  **先接收從請求端傳過來的資料  
    
  sql query - 尋找符合daterange\_start和daterange\_finish之間範圍的所有欄位資料。
* **set\_change.php**先接收從請求端傳過來的資料****  
    
  sql query – 不使用插入(insert)，而是將資料更新(update)，因為此系統的常用店家只有一個。  
  ****

## 使用API

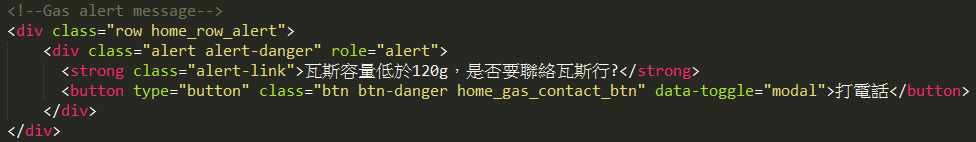
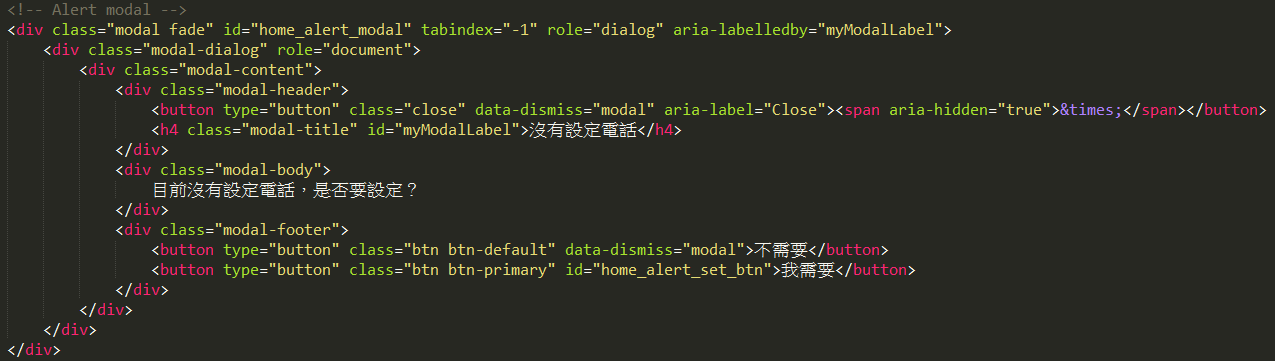
* **Google Maps Javascript API**[**https://developers.google.com/maps/documentation/javascript/?hl=zh-tw**](https://developers.google.com/maps/documentation/javascript/?hl=zh-tw)
* **jQuery**[**https://jquery.com/**](https://jquery.com/)
* **HighCharts**[**http://www.highcharts.com/**](http://www.highcharts.com/)
* **Bootstrap**[**http://getbootstrap.com/**](http://getbootstrap.com/)
* **Bootstrap-datepicker**  
  <https://bootstrap-datepicker.readthedocs.org/en/latest/>

## 頁面

### 主頁面

選單功能，可直接進入其他功能頁面

* **程式碼**
* HTML  
  [menu bar] – 這裡只有放logo，其他選單放在主畫面  
  

[選單] – 使用bootstrap grid system配置menu icon  
  
[警告訊息] – 警告使用者瓦斯即將用罄，並且添加可以撥打電話的按鈕  
  
  
[提醒設定電話] - 如果沒有設定電話，會顯示modal告訴使用者是否要設定電話。  


* Javascript  
  [參數設定] – 設定最小的瓦斯重量為120g  
    
  [請求預設店家資料]   
    
    
  [請求最新一筆瓦斯資料]   
  

如果要設定電話，就跳轉到設定的頁面

沒有設定電話，就會顯示modal，並且詢問是否要設定電話

如果有設定電話，就觸發打電話的程式(這邊只有手機能用)

如果小於最小值，就顯示警告訊息

* **畫面  
  [主畫面] – 放置主要的三個功能  
  [警示訊息]**如果有設定電話，會直接撥打電話  
    
  如果沒有設定電話，會跳出modal提醒使用者是否要設定電話 ****

連結set.html – 設定預設瓦斯行，直接聯絡/定位瓦斯行

連結search.html – 找尋使用者所在位置附近的瓦斯行

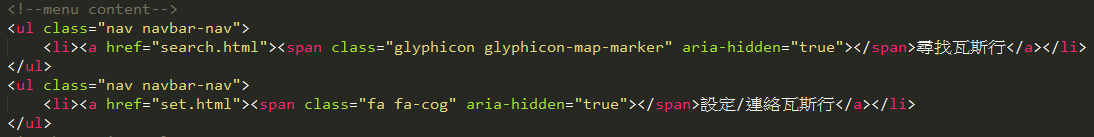
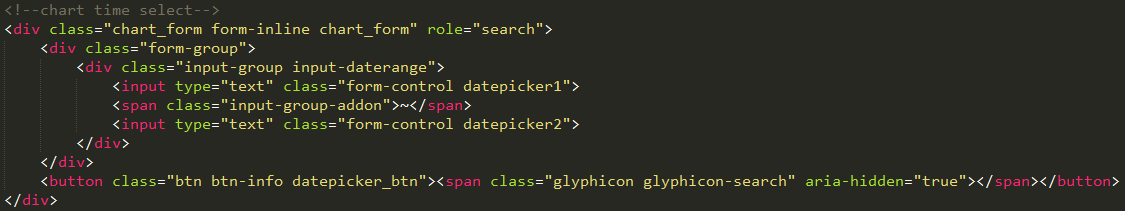
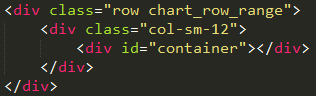
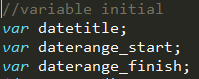
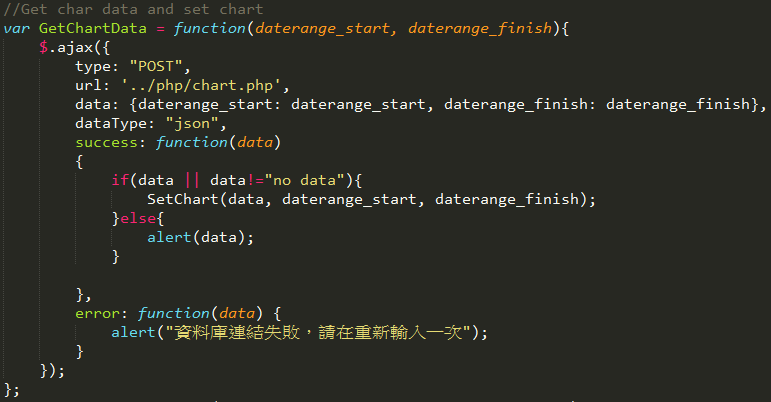
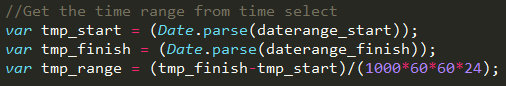
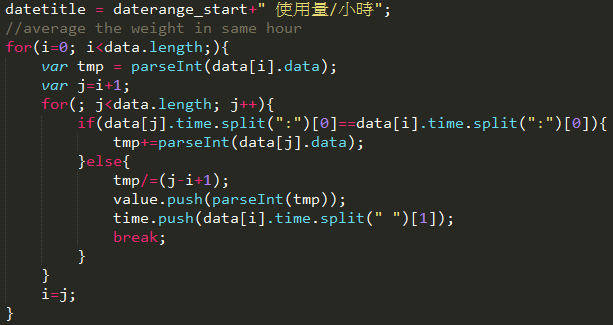
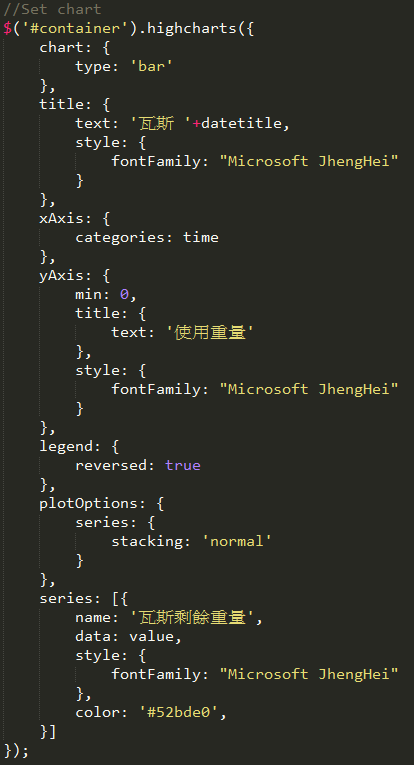
連結chart.html – 顯示時間範圍內瓦斯剩餘重量

選擇需要，頁面會跳轉到設定/聯絡瓦斯行

選擇不需要，會直接關掉Modal

### 使用量表

使用圖表方式顯示時間範圍內的瓦斯剩餘用量，頁面載入時預設從資料庫中抓出最新日期的所有資料(ex. 2016-01-08是最新日期)，那所有2016-01-08的資料都會被post出來，然後透過前端作一些運算處理。  
圖表的呈現方式分為兩種，一天或多天  
一天： 會以一小時為單位，顯示當天每小時的平均重量  
多天： 會以天為單位，顯示時間範圍每天平均重量  
※Arduino是30分鐘傳一次資料，所以資料會依據單位作處理。

* **程式碼**
* HTML  
  [menu] – 可連結至其他頁面  
    
    
  [選擇日期範圍] – 使用bootstrap-datapicker  
    
    
  [HighCharts]  
  
* Javascript  
  [變數設定]  
    
    
  [初始datepicker] – 向資料庫請求最舊的時間和最新的時間，讓使用者只能選擇資料庫有的時間範圍，並且透過取得最新的日期來初始HighCharts。  
  DB丟過來的資料的time格式為timestamp，我們只需要日期的部分，所以用split分割出來。  
    
    
  [請求圖表所需的資料] – 依據時間範圍向DB請求資料  
    
    
  [畫圖表] – 使用請求得資料和選取的時間範圍來畫圖表  
    
    
  參數設定 – 圖表需要顯示時間和瓦斯剩餘重量  
    
  圖表有兩種模式：一天和多天，用時間範圍來判斷，所以會先做運算  
    
    
  一天(tmp\_range==0) – 將資料每一小時做一次平均  
    
  多天(tmp\_range>0) – 將資料用天當單位作平均  
    
  圖表設定  
  

[datatile] – 設定HightChart標題  
[daterange\_start] – 起始時間  
[daterange\_finish] – 結束時間

data[1].time取得的是最新的timestamp，將日期切割出來然後依據日期畫出圖表

設定datapicker可以選取的時間範圍還有時間格式

用請求來的資料畫出圖表

傳參數到chart.php

將日期轉換成毫秒

作完運算後再轉換成天數

判斷是否為同一小時內的資料

將同一小時內的資料作平均，並且存進陣列中

判斷是否為同一天內的資料

X軸：時間

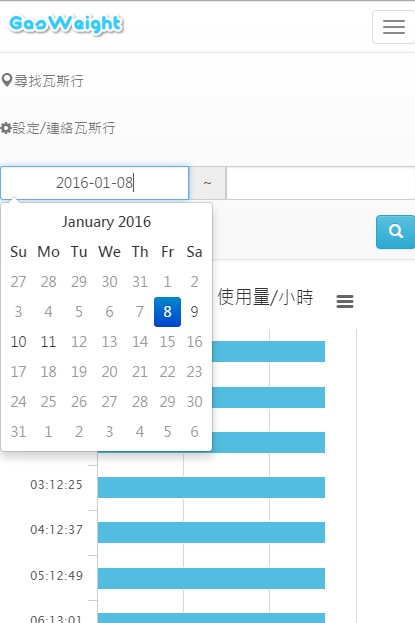
y軸：瓦斯剩餘重量

x軸和y軸反轉

y軸最小值設為0

設定圖表標題，還有設定樣式

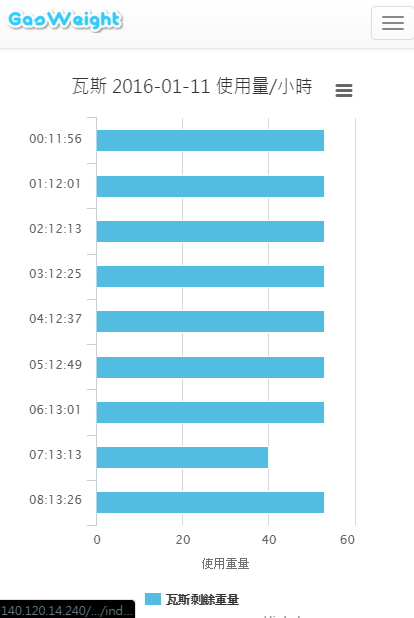
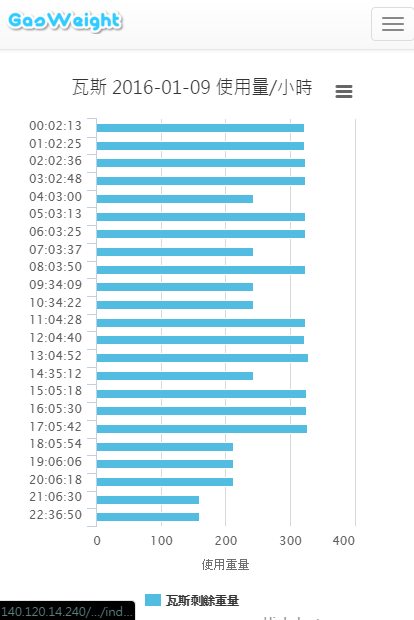
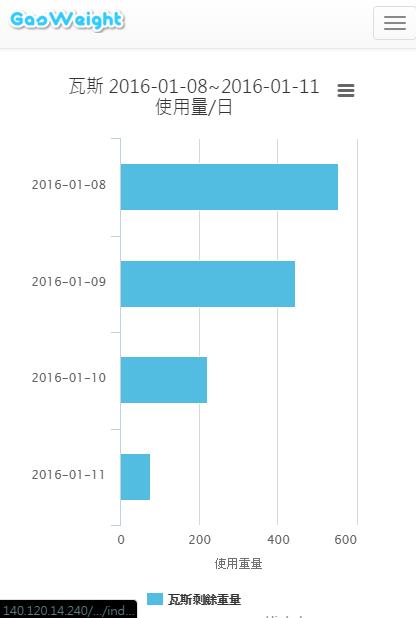
設為柱狀圖

* **畫面**[初始畫面] – 顯示資料庫最新日期的資料/小時[圖表顯示]

一天：小時當單位

多天：一天當單位

按下選單顯示datepickers



點擊input欄位就會跳出月曆

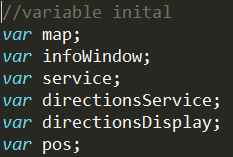
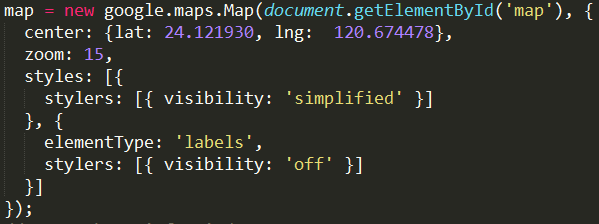
### 尋找瓦斯行

**藉由Google maps API提供的功能，完成：**

1. GPS定位：定位使用者目前的位置
2. Keyword搜尋：透過Geocoder來尋找使用者附近的所有瓦斯行

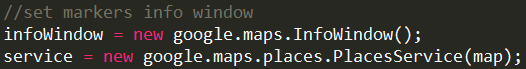
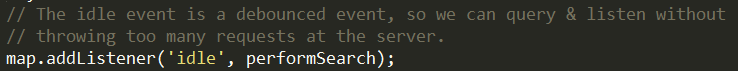
其他功能

1. 設定常用瓦斯行，方便使用者可以直接聯絡常用的瓦斯行
2. 打電話
3. 定位(此系統本來為瓦斯行端的功能，但時間不夠所以合併)

* **程式碼**
* HTML  
  [Google map] – 只需要用一個div就可以呈現地圖了
* Javascript  
  [地圖初始化] – 要將地圖載入頁面中需要一些設定  
    
    
  

地圖的基本參數設定

變數設置：會使用到Google maps功能和服務以及所需到參數

  
定位  
  
  
  
  
  
參考網址：<https://developers.google.com/maps/documentation/javascript/examples/map-geolocation>  
  
[搜尋附近瓦斯行] – 當地圖載入完時才進行搜尋，所以使用idle事件來監聽  


取得定位的位置並且進行所需要的動作

建立Keywords(瓦斯行)的服務

建立markers的訊息視窗(定位後要標註定位的位置)

判斷是否瀏覽器是否支援定位

如果定位處理錯誤，就執行此函式

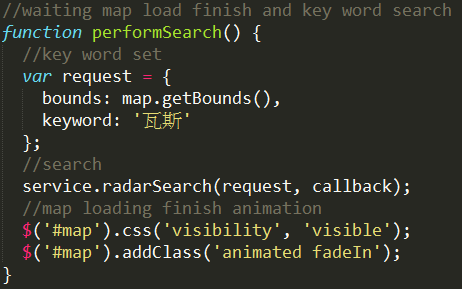
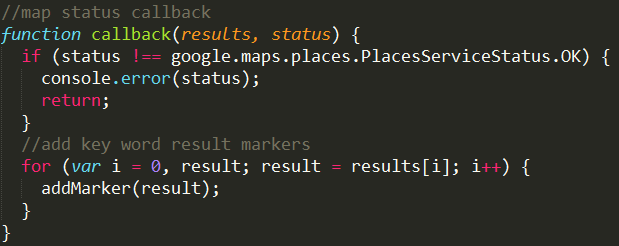
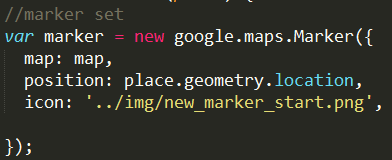
將地圖的中心設在定位位置

在Marker上建立一個infoWindow

在定位的座標上設一個Marker

存取定位出來的座標(使用JSON)

如果瀏覽器不支援定位，就執行此函式

當”idle”發生時，就執行performSearch  
  
執行callback  
  
  
設定Marker – addMarker()  
  
每個Marker都設置click 監聽事件，並且將事件觸發的動作寫在監聽函式中  


執行search，結果使用callbeck執行

設定keyword

失敗的話就return

成功的話，將搜尋出來的結果位置一個個標上marker

Marker作標使用搜尋出來得座標

抓出每個搜尋結果的詳細資料(包含店名，地址，電話)

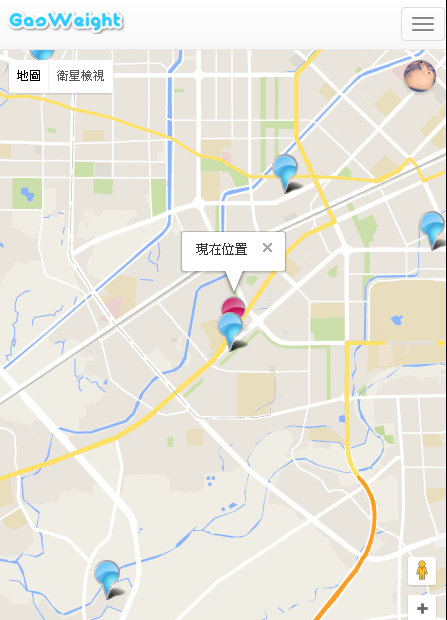
功能按鈕的設置

功能按鈕的觸發事件

將每個Marker設置infoWindow並且在裡面放入功能按鈕

事件監聽函式

參考網址：<https://developers.google.com/maps/documentation/javascript/examples/infowindow-simple>  
  
[改變常用瓦斯行]  


* **畫面**[初始畫面]

1. 點擊瓦斯行的Marker回顯示瓦斯行店名，地址，電話
2. 提供三個功能：

路線 – 會連到set\_get\_direction.html?地址

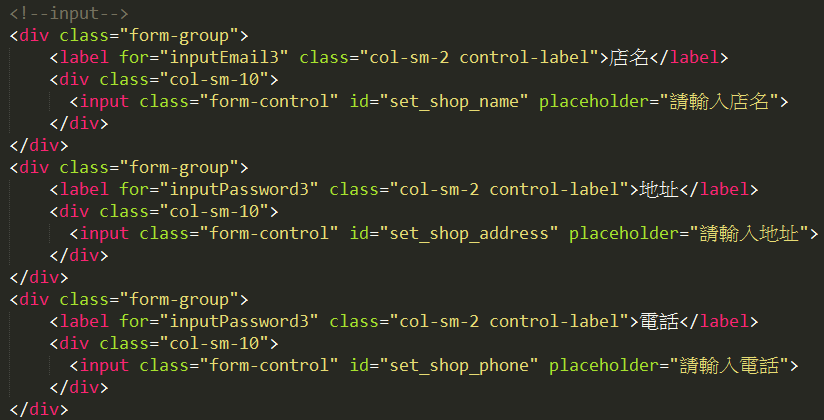
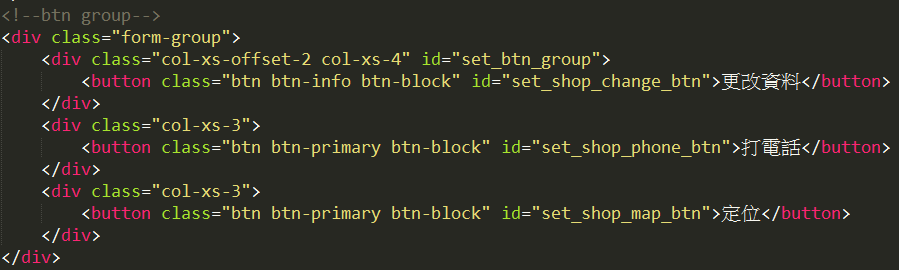
打電話：撥打電話給瓦斯行

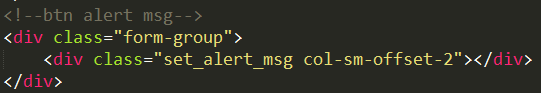
設為常用：可以將該瓦斯行設為常用瓦斯行

瓦斯行位置

使用者現在位置

### 設定/連絡瓦斯行

* **程式碼**
* HTML  
  [表單] – 欄位包含店名、地址和電話  
    
  [功能按鈕] – 分為更改資料，打電話和定位  
  

[警告訊息] – 如果使用者沒有建立資料，那使用打電話與定位時會出現警告訊息告知使用者設定資料

Javascript  
[載入設定的資料到欄位中以及按鈕事件觸發的設定]

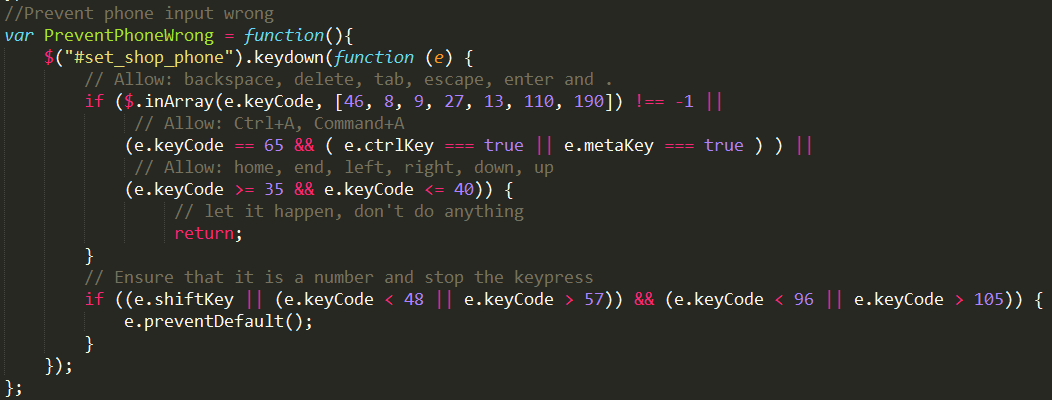


將請求來的資料放入欄位中

撥打電話，會先檢查是否有值可以撥打電話，沒有的話顯示警告訊息

定位功能，會先檢查是否有值可以定位，沒有的話顯示警告訊息

會在跳轉頁面的位址後面加上地址當作參數

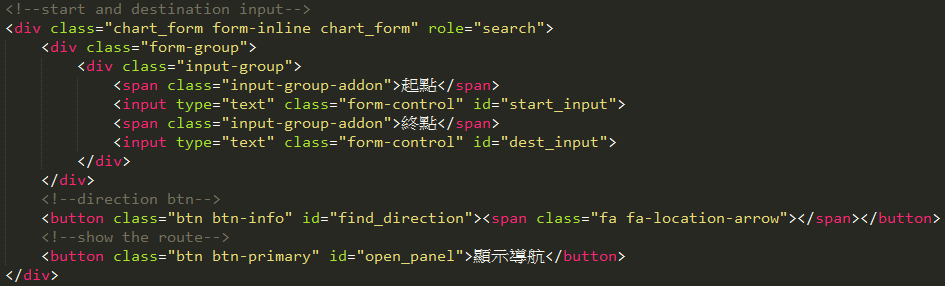
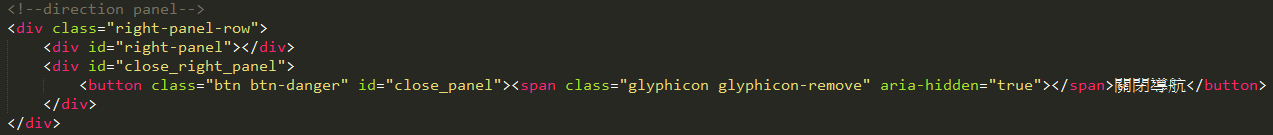
* [電話欄位的輸入限制] – 讓使用者只能輸入數字  
    
  [更新DB資料]  
  

取出每個欄位的值

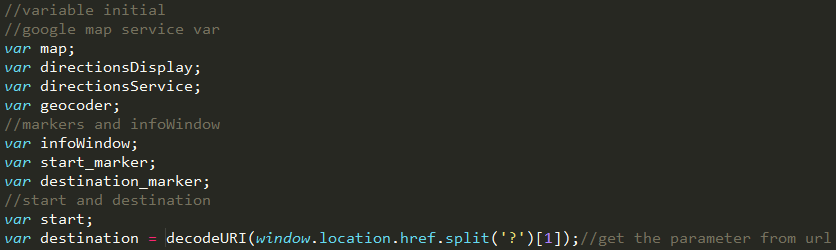
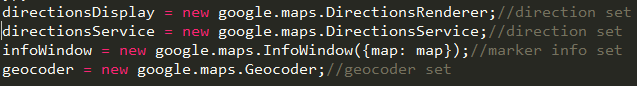
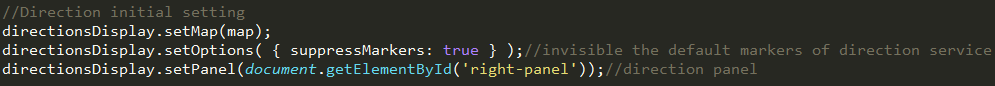
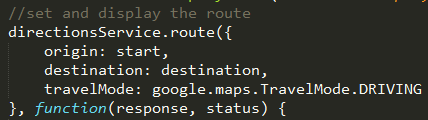
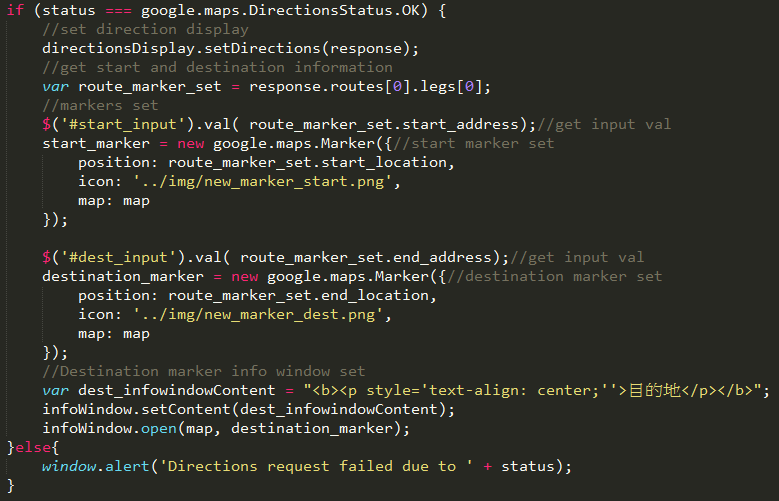
* **畫面**

### 導航功能

從「設定/聯絡瓦斯行」和「尋找瓦斯行」的頁面中跳轉過來的，必須要從兩個頁面中傳參數過來，從參數中的地址進行導航

* **程式碼**
* HTML  
  [起點&終點欄位] – 頁面一載入時會先定位使用者現在的位置並且轉換成地址，然後將值放入起點欄位中；終點的值是分出網址後面的參數然後放入欄位中。  
    
    
  [地圖] – 可參考尋找瓦斯行  
  [導航] – 可顯示詳細的路線規劃

起點和終點的欄位

* Javascript  
  [變數設置]   
    
  [建立功能和服務]  
    
  [設定導航]  
    
    
    
    
    
    
  [定位] – 可參考尋找瓦斯行  
  [導航]   
  參數設定  
    
  [開始導航]-導航路徑確定後，會加上自訂的markers在起點和終點上  
  參考網址：<https://developers.google.com/maps/documentation/javascript/examples/directions-panel>

設置起點和終點點

導航模式：汽車

設定導航的路線要顯示在哪個元件

這邊把導航預設的起點和終點的markers隱藏，之後就可以自訂markers了

起點和終點，這邊就將網址中的參數存在終點裡

Markers 和infoWindow

建立Google map服務

Markers和infoWindow的設置

取得起點和終點得資訊

* **畫面**[地圖] – 一載入頁面就會抓取網址的參數值並且顯示路徑

使用者現在位置

瓦斯行位置

****[欄位與功能表]

按下後可重新導航

起點和終點的欄位，可更改內容

取得起點和終點得資訊