城市空氣監控系統

City Air Monitor System –CAMS

**Team Name：**ME2(第四組)

**Team Members：**林勁維、張恆昇、李東翰、崔策

指導教授：周瑞仁

1. **目的與動機Goal and Motivation：**

去年8月1日凌晨，高雄前鎮區與苓雅區的多起石化氣爆炸事件，造成32人死亡、321人受傷的悲劇。雖然責任歸咎於化工廠並未發現氣體外洩，但是除了工廠端的檢查，我們真的沒有別的方法能阻止這個悲劇的發生嗎？7月31日20時左右就有民眾通報瓦斯外洩，然而卻因為消防人員未善加利用隨身佩帶，可判斷現場危害氣體脂濃度的5用氣體偵測器，導致現場部署及封鎖延滯。如果我們能更早的發現氣體外洩，更早的知道外洩的範圍，是否就能將傷害降到最低？然而現在要判斷是否有氣體外洩多靠人拿著儀器檢測，範圍小，且無法即時檢測。再者，隨著都市快速的開發，汽機車數量降劇增，造成空氣品質逐漸下降，油煙廢棄充斥的我們的生活空間，汙染環境的同時，也傷害我們的健康。光靠氣象局定點式或以人力測量空氣品質，也難以即時的監測。

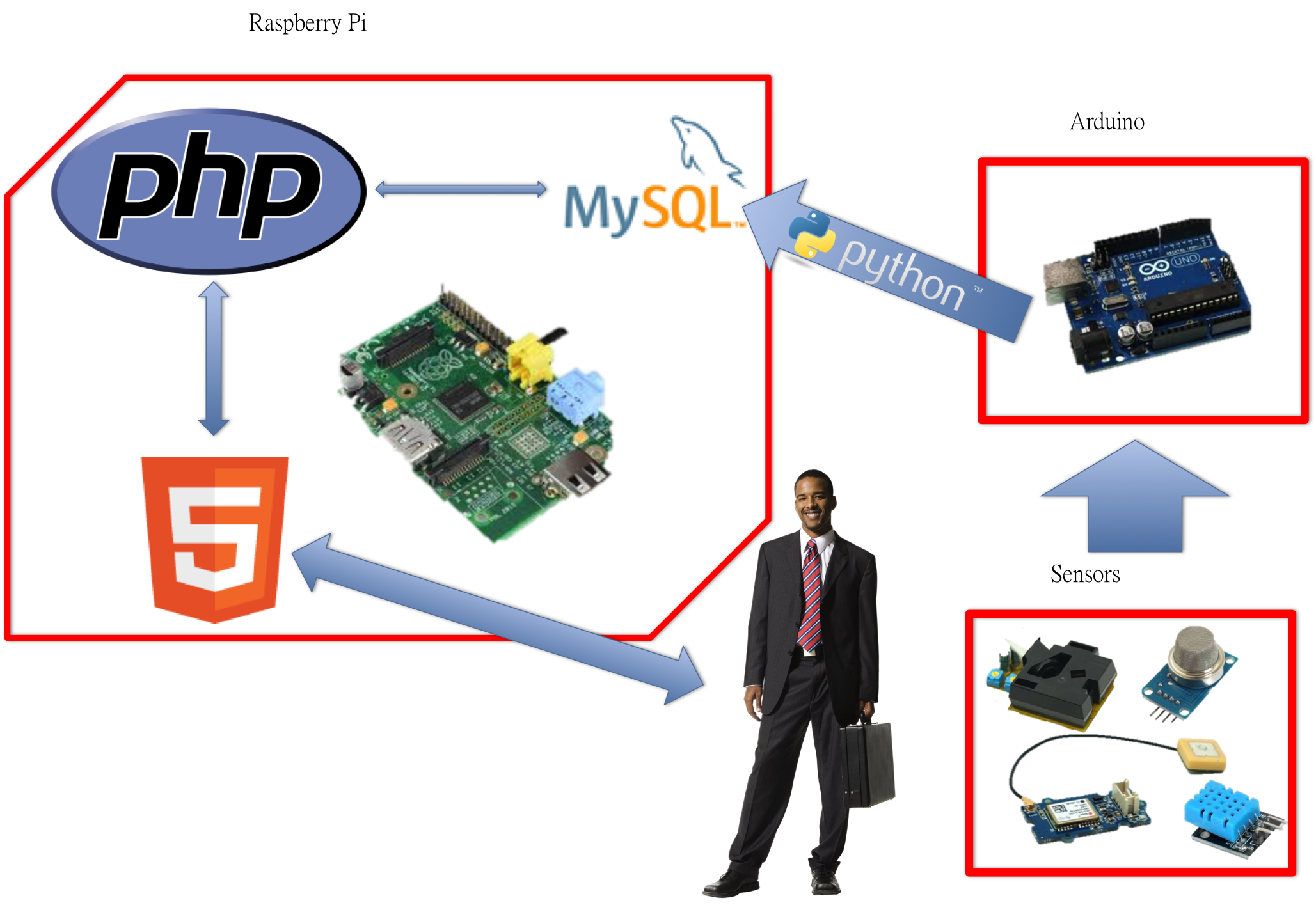
本次計畫的目的是，如果能在遍佈城市的公車系統上加裝感測器，並隨時回報，再與電子地圖做疊合，就能即時的掌握各種氣體在各個地點的相對濃度。一方面，當發現空氣品質異常時，可派人拿較精密的儀器去檢驗，就能在最短時間內畫出警戒範圍，將可發生的災害減少到最低。另一方面，我們也希望能利用這個系統，長期且大量的累積數據，經由記錄且分析城市內空氣品質的變化，檢測城市內污染物質的分佈，並且能即時反應，藉由與地圖的比對，如果發現某一區域內的濃度特別高，也能即時派人前往檢測，查出空氣污染的來源，並配合環保單位加以取締、舉發違法廠商，讓城市的生活環境更為健康。

由於公車是較為普及的大眾交通運輸工具，所在的地方幾乎為人口聚集的據點，其分布廣達全台，且為固定路線移動，較易追蹤、評估，藉由與客運公司的合作，加上設計簡單、便宜、好安裝的設備，利於市場上的推廣。

1. **技術內容Technical Description：**

利用Arduino收集液化石油氣、一氧化碳、可燃氣體、溫濕度及GPS位置資訊，透過Python寫入Raspberry PI內的MySQL資料庫，當使用者連上我們的網站呼叫時透過PHP呼叫MySQL內的資料，顯示出使用者需要的資料，並且透過Google Maps繪製簡易圖表。

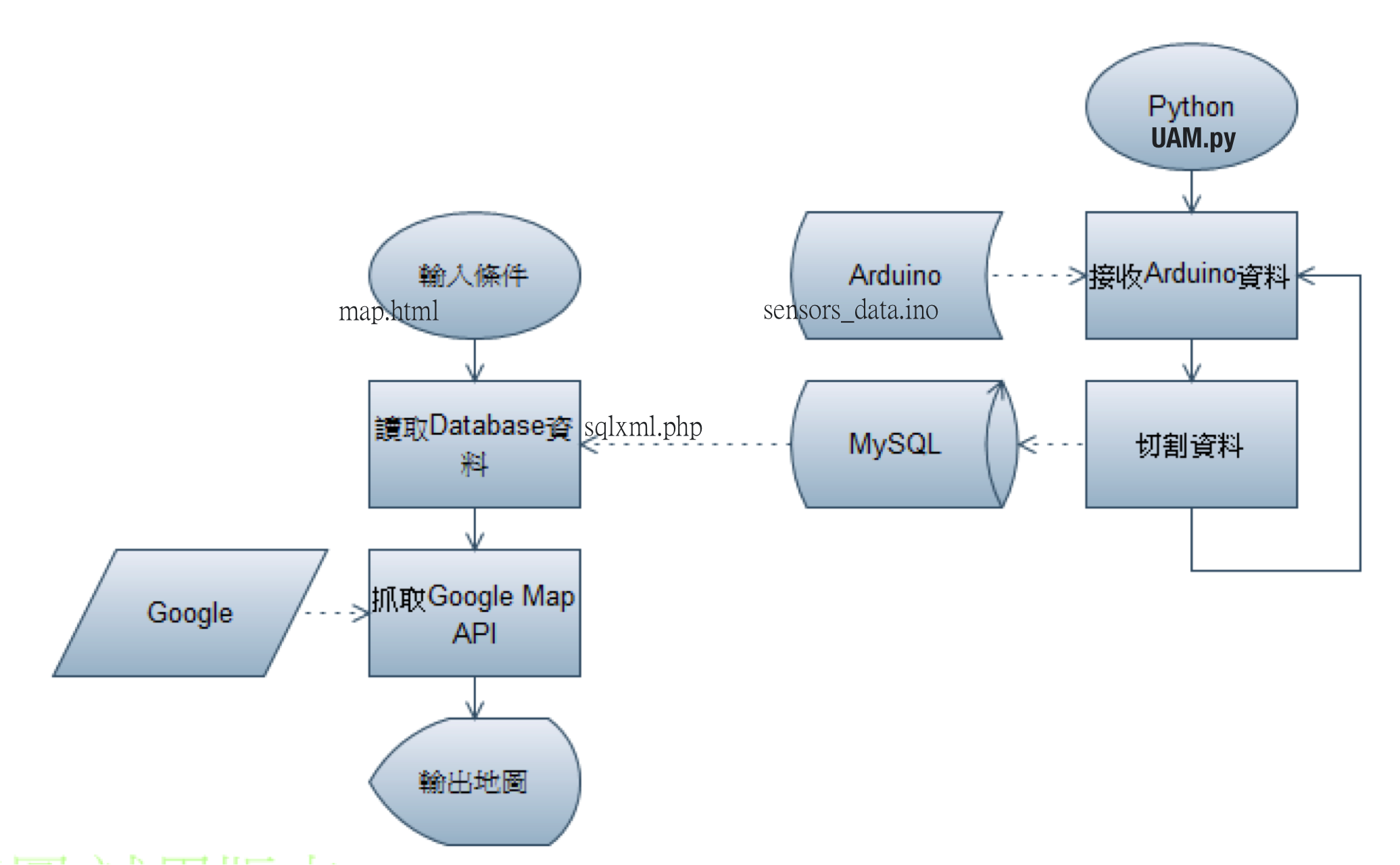
**系統架構圖System Framework：**

****

**硬體設計Hardware Description：**

**在Arduino端，為了方便感測器接線，我們加了擴充版，便不會造成線路太過複雜，也省去了麵包版的空間。我們希望能將感測器、Arduino board、Raspberry Pi 放置在同一個箱子(以下稱為感測箱)中，比較能接近即顯現出未來產品的外貌。由於控制板運作時會產生熱量，我們將感測箱設計有百葉窗扇，方便空氣的流動、帶走熱量，另外也將感測器的偵測端固定在感測箱的外牆洞上，偵測溫濕度、空氣品質、易燃、有毒氣體外洩等等，且為避免感測器淋雨也加裝了向外延伸的屋頂，並有傾斜設計，讓雨水不會從孔縫中流入造成損壞。最後也設計了一台遙控車，將感測箱裝置在上端，模擬在公車上端裝置。**

**軟體設計Software Design：**

****

**測試程序Teating Process：**

**將Raspberry Pi連接至內網，並確定GPS可讀值，用腳踏車將感測箱載至校園偵測各種感測器數值。回到教室後連結至網路，確定可以將所讀到的數值，繪至Google Map上。重複以上步驟至可簡便完成所有流程。**

1. **結果與討論Result and Discussion：**

在製作的過程中，幾乎是每走一步，就是一個問題，還好前面有老師的引導，讓起步的一切還算順利，正所謂萬事起頭難。有著老師的帶頭指點，後面的問題克服也就順利許多。在arduino讀值時，每個sensor給的數據表都需了解，一一解碼，也感謝網路上許多的參考範例，另外偵測儀器給的頻率也不盡相同，例如：DHT11就必須要最少兩秒讀一次值，才不會造成失準，而GPS則一定要到空曠處才能收訊，而且一次解碼就包括零零碎碎的許多資訊，必須從中節取出來。在過程中，最有效的學習即是solve by problem，一一面對所需處理的問題，一一克服，才能有今天我們現在的作品成果。

網頁的程式是一個之前從來沒有碰過的部分，只能不斷的從網路上找類似的範例，不斷的改，出了問題再不斷的上網找，發現這個方法行不通再找另外一個範例，在不斷的try-and-error中總算是做出了還算滿意的網頁。

1. **結論Conclusion：**

在產品設計過程中，計畫是必要，但不是唯一。有些問題只有在真正的設計與製作過程，才能發現其中的問題所在，也因為有這次的計畫實作，才能更深入的了解到平常不會思考的部分，並能更精進自己的學習領域，將所學加以應用。

軟體的細節Debug很重要，舉例來說，經緯度的”分”若當成小數直接輸入，座標就會從台北跑到桃園。此外今天想做出一個作品，真的不必自己從0開始，尤其是Google提供了許多強大好用的範例程式，僅需稍加修改與整合。

即使是在軟體，依然有顯著的trial & error的過程，儘管程式邏輯似乎沒有錯，卻會產生許多奇怪的bug，透過不停參考、調整程式碼，到突然work的時候，其實也不知道真正的問題在哪裡。在這樣的實作中，體認到為何軟體是當今的重要技術：假設硬體工程師跟軟體工程師都不犯錯，開發一個產品的成本可能一樣，但因為兩者都會犯錯，同樣的問題用軟體解決，修改的成本遠比硬體低許多。

1. **未來展望Future Development Ideas：**

在與客運公司合作之後，更可以跟計程車業者、汽車廠商…等運輸業合作，擴大偵測範圍，提供更完整的資料作分析。之後硬體部分可以改用更為堅固的材質，而軟體可以跟網路做結合，讓GPS地位更為精準。

在往後的發展，將不會是一個Arduino配一個Raspberry Pi，將會是一個無數個裝載在公車上arduino，透過Wi-Fi、php將資料傳回統一的一個server進行紀錄，同時也需要提升軟體的可靠性及能自動排除data lost的狀況。我們這次使用的氣體感測器經過矯正過後可以得到非常精確的濃度，但是由於我們沒有儀器，只能用相對的濃度來代表，往後真的上線後就會需要絕對的濃度來提供決策者更為可靠的資訊。

產品有什麼發展？有什麼設計需要再加入或移除？有什麼測試要做?

1. **參考資料References：**

<https://developers.google.com/maps/web/>

<http://www.seeedstudio.com/wiki/Main_Page>

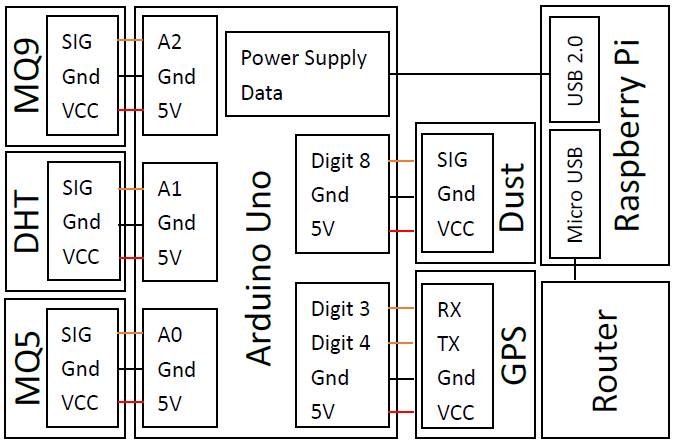
<http://developer.android.com/index.html>

1. **附錄Appendices：**

**附錄1 Appendix 1：**費用 Bill of Materials

|  |  |  |  |
| --- | --- | --- | --- |
| **日期** | **項目** | **單價** | **數量** |
| April 25, 2015 | Grove-氣體偵測(MQ5) | 232 | 1 |
| April 25, 2015 | Base Shield V2 | 299 | 1 |
| May 3, 2015 | Grove-灰塵傳感器 | 468 | 1 |
| May 9, 2015 | Grove-氣體偵測（MQ9） | 252 | 1 |
| May 13, 2015 | 雷射切割 | 450 | 1 |
| May 16, 2015 | Grove-GPS | 1160 | 1 |

**附錄2 Appendix 2：**電路圖Schematic

****

**附錄3 Appendix 3：**程式 Source Code

**UAM.py**

**import** MySQLdb

**import** serial

**from** time **import** strftime

**from** datetime **import** datetime**,** time

dbhost **=** 'localhost'

dbuser **=** 'root'

dbpass **=** 'raspberry'

dbname **=** 'MyDB'

ser **=** serial**.**Serial**(**'/dev/ttyACM0'**,**9600**)**

startTime **=** datetime**.**now**()**

conn **=** MySQLdb**.**connect**(**dbhost**,** dbuser**,** dbpass**,** dbname**)**

cursor **=** conn**.**cursor**()**

createTable **=** "CREATE TABLE IF NOT EXISTS UAM(DATE DATE, TIME TIME, LATITUDE FLOAT(10, 6), LONGITUDE FLOAT(10 ,6), TEMP FLOAT, HUMID FLOAT, MQ5 FLOAT, MQ9 FLOAT, Dust FLOAT)"

cursor**.**execute**(**createTable**)**

line **=** ser**.**readline**()**

line **=** ser**.**readline**()**

line **=** ser**.**readline**()**

line **=** ser**.**readline**()**

**try:**

**while** 1**:**

line **=** ser**.**readline**().**rstrip**()**

lat**,**lon**,**temp**,**humid**,**mq5**,**mq9**,**dust **=** line**.**split**(**";"**)**

now **=** datetime**.**now**()**

#print ( "%s, %s, %s, %s, %s, %s, %s" % (lat, lon, temp, humid, mq5, mq9, dust))

sql **=** "INSERT INTO UAM(DATE, TIME, LATITUDE, LONGITUDE, TEMP, HUMID, MQ5, MQ9, Dust) VALUES ('%s', '%s', '%f', '%f', '%f', '%f', '%f', '%f', '%f' )" **%** **(**now**.**strftime**(**"%Y-%m-%d"**),**now**.**strftime**(**"%H:%M:%S"**),** float**(**lat**),** float**(**lon**),** float**(**temp**),** float**(**humid**),** float**(**mq5**),** float**(**mq9**),** float**(**dust**))**

**try:**

cursor**.**execute**(**sql**)**

conn**.**commit**()**

**except:**

conn**.**rollback**()**

**except:**

conn**.**close**()**

**map.html**

<!DOCTYPE html >

<head>

<meta charset=**"utf-8"**>

<title>**城市空氣監測系統**</title>

<style>

**html, body, #map-canvas {**

**height: 100%;**

**margin: 0px;**

**padding: 0px**

**}**

**#panel {**

**position: absolute;**

**top: 5px;**

**left: 40%;**

**margin-left: -180px;**

**z-index: 5;**

**background-color: #fff;**

**padding: 5px;**

**border: 1px solid #999;**

**}**

</style>

<script type=**"text/javascript"** src=**"https://maps.googleapis.com/maps/api/js"**></script>

<script type=**"text/javascript"**>

//<![CDATA[

var date;

var time;

var show="temp";

var timer;

function loadeveryminute(){

var Today=new Date();

var date= Today.getFullYear()+ "-"+ (Today.getMonth()+1) +"-"+ Today.getDate();

var time= Today.getHours()+":"+(Today.getMinutes())+":"+(Today.getSeconds());

show=form1.show.value;

form1.date.value=date;

form1.time.value=time;

draw(date,time,show);

clearInterval(timer);

timer= setInterval(loadeveryminute, 60000);

}

function load(){

clearInterval(timer);

date=form1.date.value;

time=form1.time.value;

show=form1.show.value;

draw(date,time,show);

}

function draw(date,time,show){

var map = new google.maps.Map(document.getElementById('map-canvas'), {

center: new google.maps.LatLng(25.02, 121.54),

zoom: 15,

mapTypeId: 'roadmap'

});

var infoWindow = new google.maps.InfoWindow;

// Change this depending on the name of your PHP file

downloadUrl("sqlxml.php?date="+date+"&time="+time, function(data) {

var xml = data.responseXML;

var markers = xml.documentElement.getElementsByTagName("marker");

if(show=="temp"){

for (var i = 0; i < markers.length; i++) {

var temp = markers[i].getAttribute("temp");

var point = new google.maps.LatLng(

parseFloat(markers[i].getAttribute("lat")),

parseFloat(markers[i].getAttribute("lng")));

var html = "<b>" + temp + "</b> <br/>";

var marker = new google.maps.Marker({

map: map,

position: point,

icon: {

path: google.maps.SymbolPath.CIRCLE,

strokeWeight: 0,

fillColor: 'hsl('+((1-((Math.min(temp, 35) - 20)/(35 - 20)))\*120)+', 100%, 50%)',

fillOpacity: 0.4,

scale:6

}

});

bindInfoWindow(marker, map, infoWindow, html);

}

}

if(show=="humid"){

for (var i = 0; i < markers.length; i++) {

var humid = markers[i].getAttribute("humid");

var point = new google.maps.LatLng(

parseFloat(markers[i].getAttribute("lat")),

parseFloat(markers[i].getAttribute("lng")));

var html = "<b>" + humid + "</b> <br/>";

var marker = new google.maps.Marker({

map: map,

position: point,

icon: {

path: google.maps.SymbolPath.CIRCLE,

strokeWeight: 0,

fillColor: 'hsl('+((1-((Math.min(humid, 90) - 30)/(90 - 30)))\*120)+', 100%, 50%)',

fillOpacity: 0.4,

scale:6

}

});

bindInfoWindow(marker, map, infoWindow, html);

}

}

if(show=="mq5"){

for (var i = 0; i < markers.length; i++) {

var mq5 = markers[i].getAttribute("mq5");

var point = new google.maps.LatLng(

parseFloat(markers[i].getAttribute("lat")),

parseFloat(markers[i].getAttribute("lng")));

var html = "<b>" + mq5 + "</b> <br/>";

var marker = new google.maps.Marker({

map: map,

position: point,

icon: {

path: google.maps.SymbolPath.CIRCLE,

strokeWeight: 0,

fillColor: 'hsl('+((1-((Math.min(mq5, 100) - 10)/(100 - 10)))\*120)+', 100%, 50%)',

fillOpacity: 0.4,

scale:6

}

});

bindInfoWindow(marker, map, infoWindow, html);

}

}

if(show=="mq9"){

for (var i = 0; i < markers.length; i++) {

var mq9 = markers[i].getAttribute("mq9");

var point = new google.maps.LatLng(

parseFloat(markers[i].getAttribute("lat")),

parseFloat(markers[i].getAttribute("lng")));

var html = "<b>" + mq9 + "</b> <br/>";

var marker = new google.maps.Marker({

map: map,

position: point,

icon: {

path: google.maps.SymbolPath.CIRCLE,

strokeWeight: 0,

fillColor: 'hsl('+((1-((Math.min(mq9, 100) - 20)/(100 - 20)))\*120)+', 100%, 50%)',

fillOpacity: 0.4,

scale:6

}

});

bindInfoWindow(marker, map, infoWindow, html);

}

}

if(show=="dust"){

for (var i = 0; i < markers.length; i++) {

var dust = markers[i].getAttribute("dust");

var point = new google.maps.LatLng(

parseFloat(markers[i].getAttribute("lat")),

parseFloat(markers[i].getAttribute("lng")));

var html = "<b>" + dust + "</b> <br/>";

var marker = new google.maps.Marker({

map: map,

position: point,

icon: {

path: google.maps.SymbolPath.CIRCLE,

strokeWeight: 0,

fillColor: 'hsl('+((1-((Math.min(dust, 10) - 0)/(10 - 0)))\*120)+', 100%, 50%)',

fillOpacity: 0.4,

scale:6

}

});

bindInfoWindow(marker, map, infoWindow, html);

}

}

});

}

function bindInfoWindow(marker, map, infoWindow, html) {

google.maps.event.addListener(marker, 'click', function() {

infoWindow.setContent(html);

infoWindow.open(map, marker);

});

}

function downloadUrl(url, callback) {

var request = window.ActiveXObject ?

new ActiveXObject('Microsoft.XMLHTTP') :

new XMLHttpRequest;

request.onreadystatechange = function() {

if (request.readyState == 4) {

request.onreadystatechange = doNothing;

callback(request, request.status);

}

};

request.open('GET', url, true);

request.send(null);

}

function doNothing() {}

//]]>

</script>

</head>

<body onload=**"loadeveryminute()"**>

<div id=**"panel"**>

<form name=**"form1"** method=**"post"** action=**""** >

<p>

**項目**

<select name=**"show"** id=**"show"**>

<option value=**"temp"**>**Temperture**</option>

<option value=**"humid"**>**Humidity**</option>

<option value=**"mq5"**>**MQ5**</option>

<option value=**"mq9"**>**MQ9**</option>

<option value=**"dust"**>**Dust**</option>

</select>

**date**

<input type=**"text"** name=**"date"** id=**"date"**>

**time**

<input type=**"text"** name=**"time"** id=**"time"**>

<input type=**"button"** name=**"btn"** id=**"btn"** value=**"送出"** onClick=**"load()"**>

<input type=**"button"** name=**"btn2"** id=**"btn2"** value=**"即時"** onClick=**"loadeveryminute()"**>

</p>

</form>

</div>

<div id=**"map-canvas"**></div>

</body>

</html>

**sqlxml.php**

<?php

**require**("dbinfo.php");

// Start XML file, create parent node

$dom = **new** DOMDocument("1.0");

$node = $dom->createElement("markers");

$parnode = $dom->appendChild($node);

// Opens a connection to a MySQL server

$connection=**mysql\_connect** ('localhost', $username, $password);

**if** (!$connection) { **die**('Not connected : ' . **mysql\_error**());}

// Set the active MySQL database

$db\_selected = **mysql\_select\_db**($database, $connection);

**if** (!$db\_selected) {

**die** ('Can\'t use db : ' . **mysql\_error**());

}

// Select data in the UAM table

$query = "SELECT \* FROM UAM where DATE='".$\_GET['date']."' AND TIME BETWEEN ADDTIME('".$\_GET['time']."','-0:30:0') AND ADDTIME('".$\_GET['time']."','0:30:0')";

$result = **mysql\_query**($query);

**if** (!$result) {

**die**('Invalid query: ' . **mysql\_error**());

}

header("Content-type: text/xml");

// Iterate through the rows, adding XML nodes for each

**while** ($row = @**mysql\_fetch\_assoc**($result)){

// ADD TO XML DOCUMENT NODE

$node = $dom->createElement("marker");

$newnode = $parnode->appendChild($node);

$newnode->setAttribute("lat", $row['LATITUDE']);

$newnode->setAttribute("lng", $row['LONGITUDE']);

$newnode->setAttribute("temp", $row['TEMP']);

$newnode->setAttribute("humid", $row['HUMID']);

$newnode->setAttribute("mq5", $row['MQ5']);

$newnode->setAttribute("mq9", $row['MQ9']);

$newnode->setAttribute("dust", $row['Dust']);

}

**echo** $dom->saveXML();

?>

**dbinfo.php**

<?php

$username="root";

$password="raspberry";

$database="MyDB";

?>

**sensors\_data.ino**

#include <TinyGPS++.h>

#include <SoftwareSerial.h>

static const int RXPin **=** 3**,** TXPin **=** 4**;**

static const uint32\_t GPSBaud **=** 9600**;**

// The TinyGPS++ object

TinyGPSPlus gps**;**

double latitude **=** 100**;**

double longitude **=** 100**;**

unsigned long starttime\_gps**;**

// The serial connection to the GPS device

SoftwareSerial ss**(**RXPin**,** TXPin**);**

//from "-" to "s" 5V, ground, A1

#include <dht.h>

dht DHT**;**

#define gasMQ5\_pin A0

#define dht\_pin A1

#define gasMQ9\_pin A2

#define dust\_pin 8

#define gpsLedPin 13

double humidity**;**

double temperature**;**

unsigned long duration**;**

unsigned long starttime\_dust**;**

unsigned long sampletime\_ms **=** 30000**;**//sampe 30s ;

unsigned long lowPulseOccupancy **=** 0**;**

float ratio **=** 0**;**

float concentration **=** 0**;**

float vol\_MQ5**;**

float vol\_MQ9**;**

void setup**()** **{**

Serial**.**begin**(**9600**);**

delay**(**300**);**

ss**.**begin**(**GPSBaud**);**

InitGPS**();**

InitDustSensor**();**

InitGasMQ5Sensor**();**

InitGasMQ9Sensor**();**

delay**(**700**);**

**}**

void InitGPS**()** **{**

starttime\_gps **=** millis**();**

pinMode**(**gpsLedPin**,** OUTPUT**);**

pinMode**(**12**,** OUTPUT**);**

**}**

void InitDustSensor**()** **{**

pinMode**(**dust\_pin**,**INPUT**);**

starttime\_dust **=** millis**();**

**}**

void InitGasMQ5Sensor**()** **{**

pinMode**(**gasMQ5\_pin**,** INPUT**);**

**}**

void InitGasMQ9Sensor**()** **{**

pinMode**(**gasMQ5\_pin**,** INPUT**);**

**}**

void InitDHT**(){**

pinMode**(**dht\_pin**,**OUTPUT**);**

digitalWrite**(**dht\_pin**,**HIGH**);**

**}**

void loop**()** **{**

GPS**();**

DHT11**();**

GasSensorMQ5**();**

GasSensorMQ9**();**

DustSensor**();**

Serial**.**print**(**latitude**,** 6**);**

Serial**.**print**(**";"**);**

Serial**.**print**(**longitude**,** 6**);**

Serial**.**print**(**";"**);**

Serial**.**print**(**temperature**);**

Serial**.**print**(**";"**);**

Serial**.**print**(**humidity**);**

Serial**.**print**(**";"**);**

Serial**.**print**(**vol\_MQ5**);**

Serial**.**print**(**";"**);**

Serial**.**print**(**vol\_MQ9**);**

Serial**.**print**(**";"**);**

Serial**.**println**(**concentration**/**283**);**

**}**

void GPS**()** **{**

**while((**millis**()-**starttime\_gps**)<**3000**)** **{**

**while** **(**ss**.**available**()** **>** 0**)** **{**

**if** **(**gps**.**encode**(**ss**.**read**()))** **{**

displayInfo**();**

**}**

**}**

**}**

starttime\_gps **=** millis**();**

**}**

void displayInfo**()**

**{**

**if** **(**gps**.**location**.**isValid**())**

**{**

latitude **=** gps**.**location**.**lat**();**

longitude **=** gps**.**location**.**lng**();**

digitalWrite**(**gpsLedPin**,**HIGH**);**

**}**

**else** //GPS INVALID

**{**

latitude **=** 100**;**

longitude **=** 100**;**

digitalWrite**(**gpsLedPin**,**LOW**);**

**}**

**}**

void DustSensor**()** **{**

duration **=** pulseIn**(**dust\_pin**,** LOW**);**

lowPulseOccupancy **+=** duration**;**

**if** **((**millis**()-**starttime\_dust**)** **>** sampletime\_ms**)**//if the sampel time == 30s

**{**

ratio **=** lowPulseOccupancy**/(**sampletime\_ms**\***10.0**);** // Integer percentage 0=>100

concentration **=** 1.1**\***pow**(**ratio**,**3**)-**3.8**\***pow**(**ratio**,**2**)+**520**\***ratio**+**0.62**;** // using spec sheet curve

lowPulseOccupancy **=** 0**;**

starttime\_dust **=** millis**();**

**}**

**}**

void GasSensorMQ5**()** **{**

int sensorValue **=** analogRead**(**gasMQ5\_pin**);**

vol\_MQ5 **=** **(**float**)**sensorValue**;**

delay**(**10**);**

**}**

void GasSensorMQ9**()** **{**

int sensorValue **=** analogRead**(**gasMQ9\_pin**);**

vol\_MQ9 **=** **(**float**)**sensorValue**;**

delay**(**10**);**

**}**

void DHT11**()** **{**

DHT**.**read11**(**dht\_pin**);**

humidity **=** DHT**.**humidity**;**

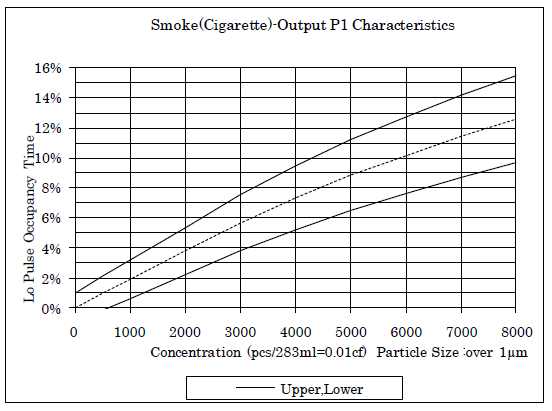
temperature **=** DHT**.**temperature**;**

delay**(**1000**);**

**}**

**附錄4 Appendix 4：** Data sheets and Application Notes

Dust Sensor



DHT11(溫溼度感應器)

供電電壓：3.3～5.5V DC

輸 出 ：單總線數字信號

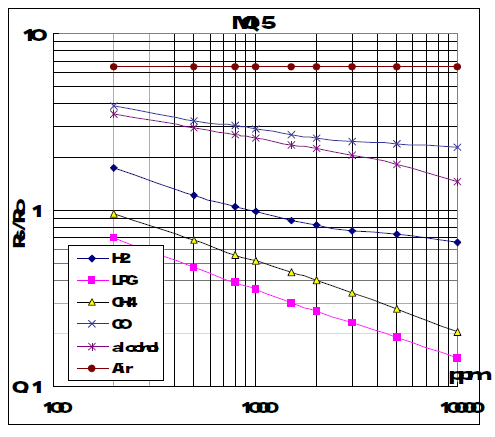
量測範圍：濕度：20-90% RH 溫度：0-50oC

量測精度：濕度：±5%RH 溫度：±2oC

分 辨 率：濕度：1%RH 溫度：1oC

Data format: 8bit integral RH data + 8bit decimal RH data + 8bit integral T data + 8bit decimal T data + 8bit check sum.

MQ5：



MQ9

