

國立勤益科技大學電子工程系
四技日實務專題報告

題目：
環境溫溼度無線監控

指導教授：陳英傑
學生：邱柏瑄、林家賢

中華民國 106 年 6 月 27 日

專題名稱: 環境溫溼度無線監控

姓名：邱柏瑄 學號：3A313041 班級：四子三甲 座號：41

姓名：林家賢 學號：3A313007 班級：四子三甲 座號：07

一.摘要

溫濕度無線監控首先是硬體和軟體兩大部分組成。

硬體就是溫濕度的管理平台、伺服器、客戶端和感測器組成。

軟體用來遠程即時監控和歷史溫濕度監控，通過主機發送代碼給伺服器，經由伺服器回傳資料就可以在遠程監控溫濕度。

各感測器能夠對周邊環境溫濕度進行實時數據的蒐集、傳送；管理主機可對各測點終端監測數據進行收集、處理和記錄。

目錄

摘要	03
目錄	04
圖目錄	05
第一章 緒言.....	06
第二章 簡介.....	06
第三章 軟體組成.....	07
第四章 軟硬體分析.....	08
第五章 測試結果.....	20
第六章 結論.....	24
第七章 給未來學弟妹的建議.....	24
第八章 參考資料.....	25

圖目錄

圖一 電路板接線圖	08
圖二 Arduino 流程圖	09
圖三 Raspberry pi 流程圖	10
圖四 MySQL(網上資料庫)	20
圖五 Python tkinter 程式視窗化	21
圖六 整體完成圖	22
圖七 電路板實體圖	23

第一章

緒言

本專題為環境溫溼度無線監控，透過將感測器連接網路，並經由網路連接可快速傳遞資料至各地亦可經由各地取的資料，且利用儲存至資料庫中各類資料進行分析，以利使用者運用之參考，經過使用者參考後可操控遠端的機器做相對應的處理。

第二章

簡介(研究動機 方法)

研究動機：

本校地區有時會發生下暴雨的情形，一般來說，要下雨之前，該地區的溼度 會比平常還來的高出許多，所以藉此來了解到底濕度到達多少，則之後可能會下雨，而濕度低 於多少，該地區可能就會讓人覺得口乾舌燥、心浮氣躁等等。所以尋找 Arduino、Raspberry Pi 相關的書籍，並與老師討論決定製作此專題。

方法：

利用伺服端將各個感測器的溫溼度資訊傳輸至客戶端，經客戶端查詢後可當下控制伺服端的 LED 燈開關，亦可瀏覽網上資料庫進行查閱溫溼度的歷史資料。

第三章

軟體組成

Raspberry Pi 是基於 Linux 的單板機電腦。使用 SD 卡作儲存媒體，有一個 Ethernet、兩個 USB 介面、及 HDMI 和 RCA 端子輸出支援。支援 Python 作為主要程式語言。

Python 是一種物件導向、直譯式的電腦程式語言，包含一組功能完備的標準庫，使用縮排來定義語句塊，具垃圾回收功能，能自動管理記憶體使用，常用於腳本語言用於處理系統管理任務和網路程式編寫，適合完成各種高階任務。

MySQL 是一個開放原始碼的關聯式資料庫管理系統，包含結構化查詢語言 (SQL)、超文字預處理器(PHP)、Apache HTTP 伺服器。

SQL 是一種高階的非過程化編程語言，用於資料庫中的標準資料查詢語言，它允許用戶在高層資料結構上工作。它不要求用戶指定對資料的存放方法，也不需要用戶了解其具體的資料存放方式。而它的介面，能使具有底層結構完全不同的資料庫系統和不同資料庫之間，使用相同的 SQL 作為資料的輸入與管理。它以記錄專案〔records〕的合集 (set) 作為操縱物件，所有 SQL 語句接受項集作為輸入，回送出的項集作為輸出，這種項集特性允許一條 SQL 語句的輸出作為另一條 SQL 語句的輸入，所以 SQL 語句可巢狀，使有極大的靈活性和強大的功能。

PHP 是一種開源通用電腦手稿語言，適用於網路開發可嵌入 HTML 中使用，語法吸收 C 語言、Java 和 Perl 等流行電腦語言的特點，主要目標是允許網路開發人員快速編寫動態頁面。

Apache HTTP 伺服器開放原始碼的網頁伺服器軟體，可在大多數電腦作業系統中運行，快速、可靠且可通過簡單的 API 擴充及跨平台 and 安全性，將 Python 直譯器編譯到伺服器中。

Tk 是一開放原始碼的圖形用戶界面開發工具，提供許多常用的圖形介面元件。具有以下特性：

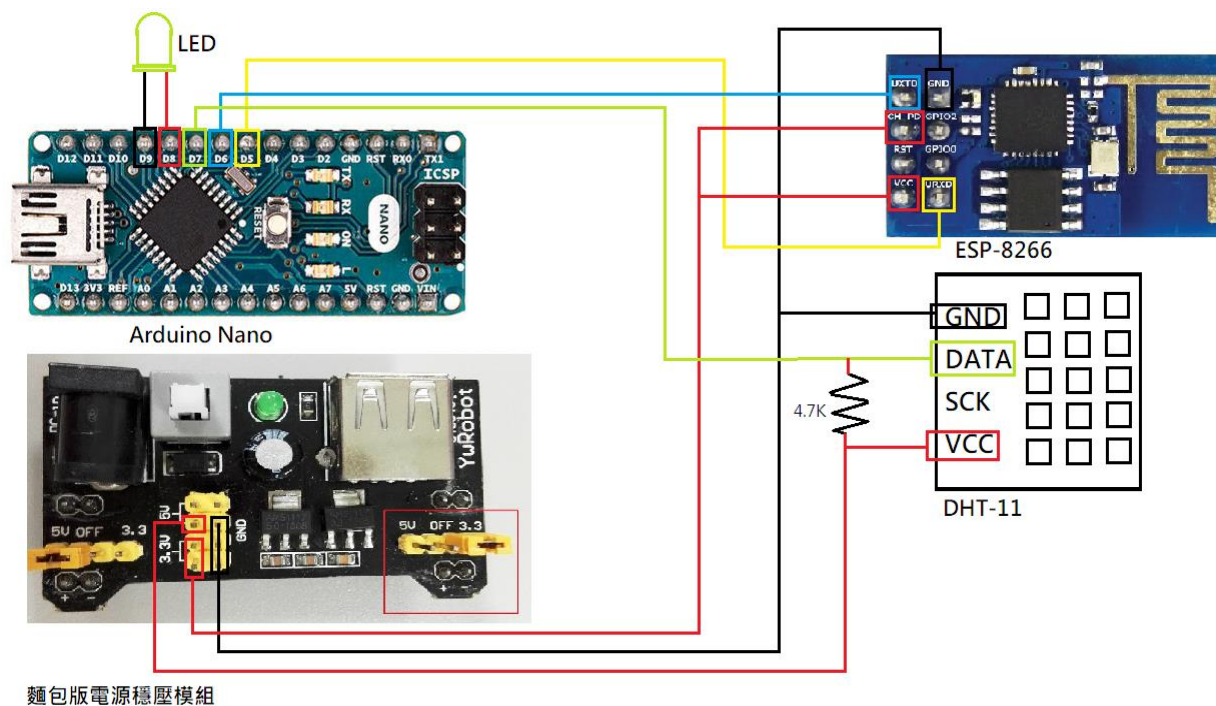
平台獨立：Tk 是解釋型的。各種平台下的 Tcl 實現是統一的，因此 Tk 程序可無需修改地移植至各種平台。

可定製：幾乎所有的特性都是可以修改的，可通過初始化選項或運行時的命令修改。

可保存：很多選項出存在資料庫中，例如界面的顏色設置。這些選項能保存下來，在再次載入程序時讀取。

第四章

硬體分析

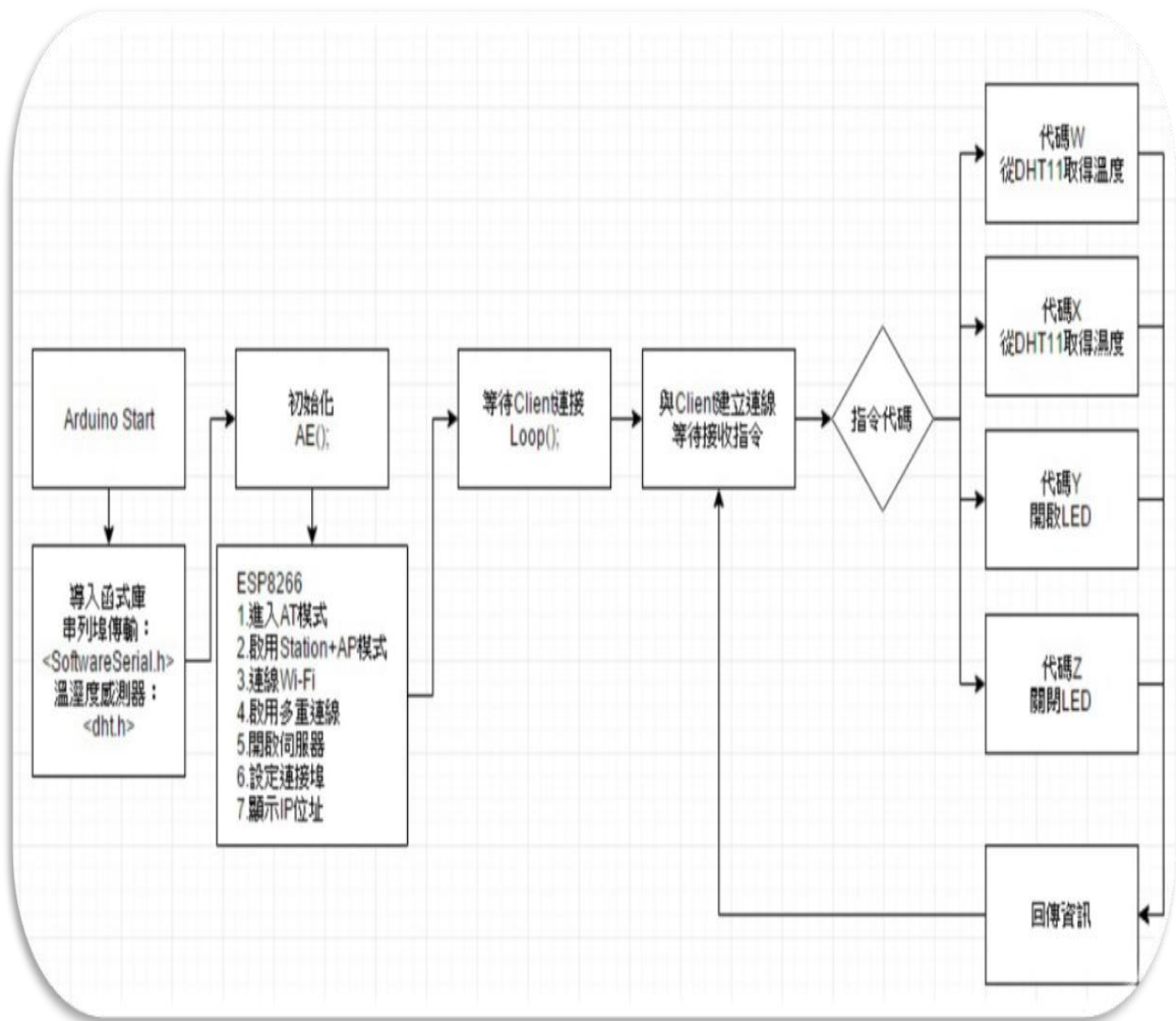


圖一

Arduino	ESP-8266	DHT-11
D5 → URXD (ESP-8266)	UTXD → D6(Arduino)	GND → GND
D6 → UTXD (ESP-8266)	URXD → D5(Arduino)	DATA → D7(Arduino)
D7 → DATA (DHT-11)	CH_PD → +3.3V	VCC → +5V
D8 → +(LED)	VCC → +3.3V	
D9 → -(LED)	GND → GND	

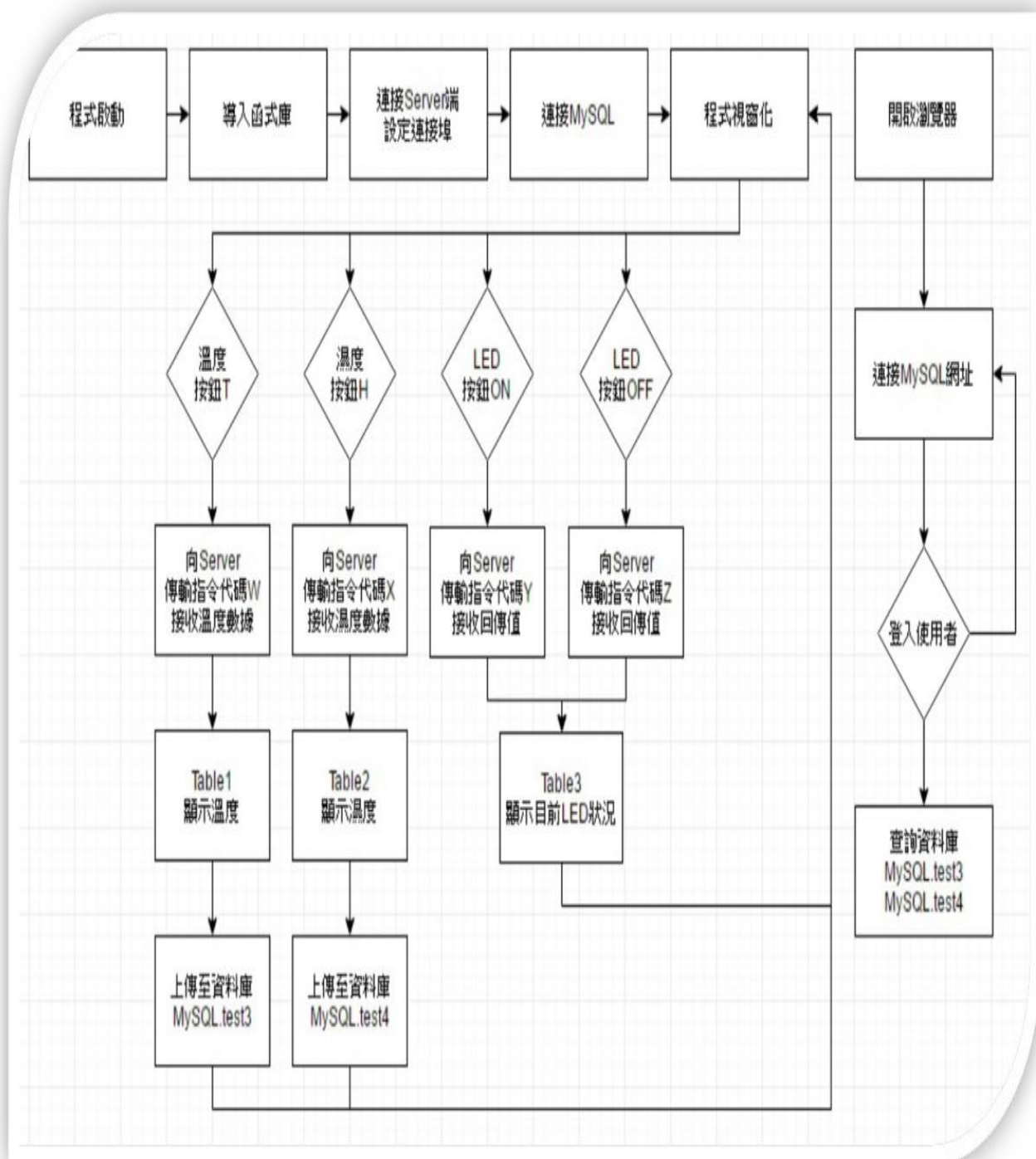
軟體分析

1. Arduino 自繪流程圖



圖二

2.Raspberry Pi 自繪流程圖



圖三

3. (arduino)

Arduino 是一個很常見的軟體，如果單單指運用 Arduino 的話會有點單調所以我們這次專題除了用 Arduino 來做之外也利用他來配合 raspberry pi，在 Arduino 的部分我們主要是用他來和 ESP8266 連線當作 SERVER 端，並讓擔任 CLIENT 端的 raspberry pi 來進行連接，並將 DHT11 收集到的溫溼度數據由 SERVER 傳送到 CLIENT 端，最後則是上傳到資料庫進行資料保存，方便日後要需要用到資料時可以隨時查看。

以下是 ARDUINO 的程式碼

```
#include <SoftwareSerial.h>//導入含括檔 序列埠傳輸

#include <dht.h> //導入含括檔 DHT-11

#define _baudrate 9600 //定義鮑率

#define dht_dpin 7 //設定 DHT-11 的資料傳輸 Pin 腳

SoftwareSerial ESP(5,6);// 設定 ESP-8266 RX ,T

dht DHT;//設定變數 DHT

char a , b='A';//設定變數 a 為代碼，變數 b 為初始化 ESP8266 設定確認

int ledPin = 13;// 設定變數 ledPin 作為 Pin 腳輸出為 13 腳

int ledPin2 = 8;// 設定變數 ledPin2 作為 Pin 腳輸出為 8 腳

int ledPin3 = 9;// 設定變數 ledPin3 作為 Pin 腳輸出為 9 腳

//ESP8266 連接無線網路

#define Name "111"//私人網路名稱：“111”

#define Pass "11111111"//網路密碼：“11111111”

//#define Name "TANetRoaming"//公開網路名稱：“TANetRoaming”

void setup(){

    pinMode(ledPin, OUTPUT);// 設定 ledPin 為輸出

    pinMode(ledPin2, OUTPUT);// 設定 ledPin2 為輸出

    pinMode(ledPin3, OUTPUT);// 設定 ledPin3 為輸出

    digitalWrite(ledPin, LOW);// 設定 ledPin 輸出電位為低電位
```

```

digitalWrite(ledPin2, LOW);// 設定 ledPin2 輸出電位為低電位
digitalWrite(ledPin3, LOW);// 設定 ledPin3 輸出電位為低電位
Serial.begin(_baudrate);//設定 Arduino 序列埠通訊速率
ESP.begin(_baudrate);//設定 ESP-8266 序列埠通訊速率
delay(1000);//延遲時間 1 秒
Serial.println("ESP8266 is reading...");
AE();//初始化
Serial.println("ESP8266 is ready...");
digitalWrite(ledPin, HIGH);// 設定 ledPin 輸出電位為高電位
}

void loop(){
    if (ESP.available()){//接收代碼
        a = ESP.read();
        Serial.write(a);
        Ma();}
}

void Ma(){//代碼判斷並傳輸資料
    DHT.read11(dht_dpin);    //去 library 裡面找 DHT.read11
    if( a == 'W'){Serial.println("");
        ESP.println("AT+CIPSEND=0,5");Re();
        ESP.println(DHT.temperature);Re();}//傳輸溫度資料
    if( a == 'X'){Serial.println("");
        ESP.println("AT+CIPSEND=0,5");Re();
        ESP.println(DHT.humidity);Re();}//傳輸濕度資料
    if( a == 'Y'){Serial.println("");
        digitalWrite(ledPin3, HIGH);delay(100);

```

```

        ESP.println("AT+CIPSEND=0,1");re();

        ESP.println('1');re();}//傳輸 LED 燈號亮

        if( a == 'Z'){Serial.println("");

        digitalWrite(ledPin3, LOW);delay(100);

        ESP.println("AT+CIPSEND=0,1");re();

        ESP.println('0');re();}//傳輸 LED 燈號暗

    }

    void AE(){//初始化

        ESP.println("AT"); re();//進入 AT 模式

        ESP.println("AT+CWMODE=3"); re();//Station+AP 模式

        //String cmd="AT+CWJAP=\"";cmd+=Name;cmd+="\"";//連線公開 Wi-Fi
網路

        String
cmd="AT+CWJAP=\"";cmd+=Name;cmd+="\"";cmd+="\"";cmd+=Pass;cmd+="\"";ESP.printl
n(cmd);//連線私人 Wi-Fi 網路

        ESP.println(cmd); re();

        ESP.println("AT+CIPMUX=1"); re();//設定多重連線

        ESP.println("AT+CIPSERVER=1,8087"); re();//開啟伺服器 設定連接埠
8087

        ESP.println("AT+CIPSTO=7200"); re();//設定伺服器主動斷開的逾時時
間

        ESP.println("AT+CIFSR"); re();//取得 ESP8266 IP 位址

    }

    void re(){//判斷初始化是否 OK

        while(b == 'A'){

            if (ESP.available()){

                a = ESP.read();

```

```

        if (a == 'K') b='z';

        Serial.write(a);}}

Serial.println("");

b='A';

delay(1000);

}

void Re(){//設定序列埠傳輸

//ESP8266 to Arduinp

if ( ESP.available() ){

    while( ESP.available() > 0 )

        Serial.write(ESP.read());}

//Arduino to ESP8266

if (Serial.available()){

    while(Serial.available() > 0)

        ESP.write(Serial.read());}

delay(100);

}

```

4. (raspberry pi)

Raspberry pi 的主要語言是 python 在 python 中我們也使用到了 tkinter 程式視窗化語言，我們的執行程式包括執行視窗通通都是我們一行一行程式打出來的，在這方面也有一些難度，因為這些語言都是以往沒學過，課堂上沒有教的，我們專題有分上下學期，我們花了一整個上學期的時間在學習 python 語言，由此可知，其並不是非常容易，在學習過程中，我們從買書、看文章、看網路上看影片又或者每個禮拜跟老師請教，一路上走過來實在不容易，在打程式的過程中，常常要找出 BUG 可能就花了一整晚的時間。raspberry pi 除了要把程式打好之外，還要跟 arduino 的程式配合，只要一邊有錯誤，那麼軟體就無法運作。

以下是 raspberry pi 的程式碼

```
#導入函式庫
```

```
import socket
```

```
import time
```

```
import MySQLdb
```

```
import Tkinter as tk
```

```
#設定 Server 端 IP 及 埠號
```

```
SERVER_IP="172.20.10.14"
```

```
SERVER_PORT=8087
```

```
print("starting socket:TCP...")
```

```
#TCP 連接
```

```
socket_tcp=socket.socket(socket.AF_INET,socket.SOCK_STREAM)
```

```
server_addr=(SERVER_IP,SERVER_PORT)
```

```
socket_tcp.connect(server_addr)
```

```
#設定視窗化
```

```
window = tk.Tk()
```

```
window.title('Environmental monitoring')
```

```
window.geometry('180x300')
```

```

#設定變數

var=tk.StringVar()

var1=tk.StringVar()

var2=tk.StringVar()

var3=tk.StringVar()

//設定表格

l=tk.Label(window,textvariable=var,bg='red',font=('Arial',12),width=15,height=2)

l1=tk.Label(window,textvariable=var1,bg='green',font=('Arial',12),width=15,height$

l2=tk.Label(window,textvariable=var2,bg='yellow',font=('Arial',12),width=15,heigh$

l3=tk.Label(window,textvariable=var3,bg='orange',font=('Arial',12),width=15,heigh$

#表格啟動設置

l.pack()

l1.pack()

l2.pack()

#設定資料庫

db =MySQLdb.Connect(host="localhost",user="root",passwd="aaa321",db="mysql")

cur =db.cursor()


#設定按鈕溫度事件

on_hit=False

def hit_me():

    global on_hit

    if on_hit == False:

        on_hit = True

        #發送代碼 W

        socket_tcp.send("W")

```



```

#取得資料，長度為 512Bits

dataw=socket_tcp.recv(512)

#顯示溫度資料

print("T:%s"%dataw)

var.set(dataw)

#中斷 TCP 連線

socket_tcp.close

#上傳資料庫

cur.execute("INSERTINTOtest3(number,t)VALUES(%s,%s)",(time.time(),dataw))

#資料庫更新

db.commit()

on_hit=False

#設定按鈕濕度事件

on_hit1=False

def hit_me1():

    global on_hit

    if on_hit == False:

        on_hit = True

        socket_tcp.send("X")

        datax=socket_tcp.recv(512)

        print("H:%s"%datax)

        var1.set(datax)

        socket_tcp.close

cur.execute("INSERTINTOtest4(number,h)VALUES(%s,%s)",(time.time(),datax))

db.commit()

on_hit=False

```

```
#設定按鈕 LED 亮事件
```

```
on_hit2=False
```

```
def hit_me2():
```

```
    global on_hit
```

```
    if on_hit == False:
```

```
        on_hit = True
```

```
        socket_tcp.send("Y")
```

```
        datay=socket_tcp.recv(512)
```

```
        print("LED:%s"%datay)
```

```
        var2.set(datay)
```

```
        socket_tcp.close
```

```
        on_hit=False
```

```
#設定按鈕 LED 暗事件
```

```
on_hit=False
```

```
def hit_me3():
```

```
    global on_hit
```

```
    if on_hit == False:
```

```
        on_hit = True
```

```
        socket_tcp.send("Z")
```

```
        dataz=socket_tcp.recv(512)
```

```
        print("LED:%s"%dataz)
```

```
        var2.set(dataz)
```

```
        socket_tcp.close
```

```
        on_hit=False
```

```
#設定按鈕
```

```
b=tk.Button(window,text="T",width=15,height=2,command=hit_me)
```

```
b1=tk.Button(window,text='H',width=15,height=2,command=hit_me1)
b2=tk.Button(window,text='led on',width=15,height=2,command=hit_me2)
b3=tk.Button(window,text='led off',width=15,height=2,command=hit_me3)
#按鈕啟動設置
b.pack()
b1.pack()
b2.pack()
b3.pack()
window.mainloop()
```

第五章

測試結果

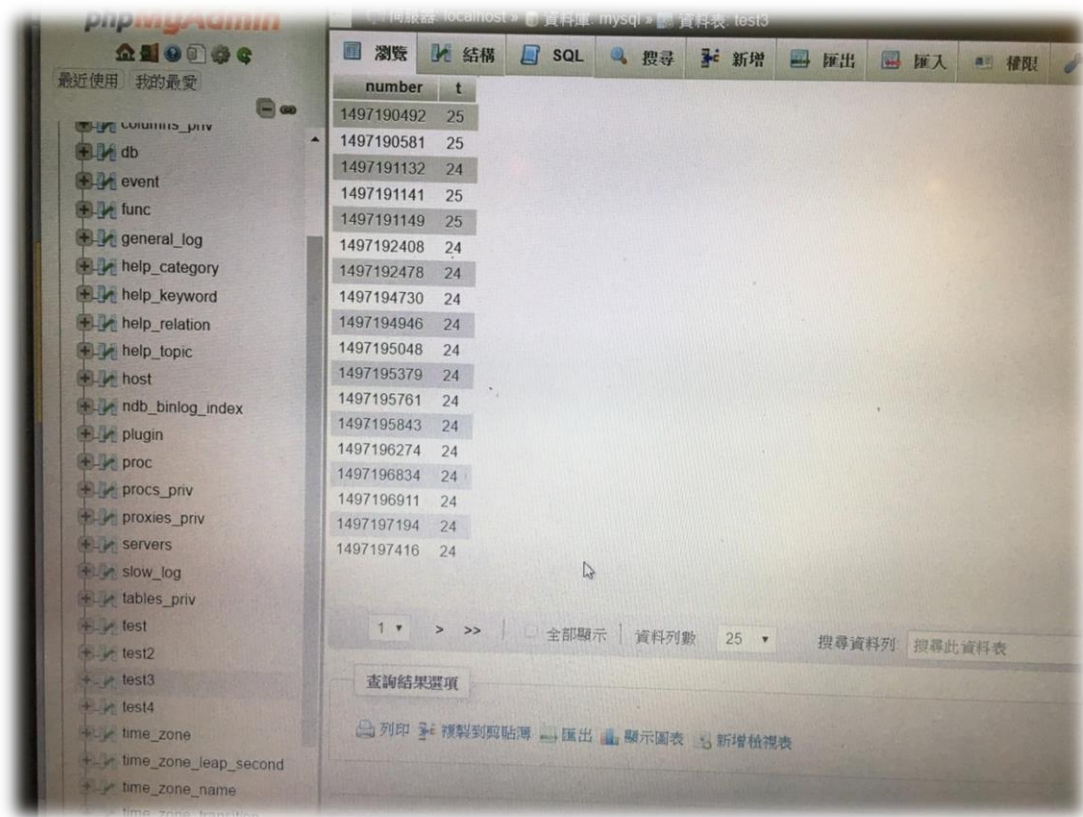
我們可以使用此專題程式透過網路連線連接到 SERVER 端後，在依你想要得到的數據透過按鈕 T(溫度)或者按鈕 H(溼度)來得到，又或者你想要將某個地方的某樣東西打開則可以按下 LED ON 按鈕(這邊以 LED 舉例)，如果要關閉的話則按下 LED OFF 即可。

圖四中我們將得到的每一筆數據上傳至資料庫，以利將來查詢資料。

我們可以利用輸入其網址登入網站之後，在輸入帳密登入後查看資料庫李完整的內容。

資料庫內的每一筆資料的時間點均都是不同的，這是由於我們在 python 的語言中引入 TIME 的時間函數形成，我們可以看到圖中有一個 NUMBER 的項目，在這項目下面每一筆資料時間點都沒有重複。

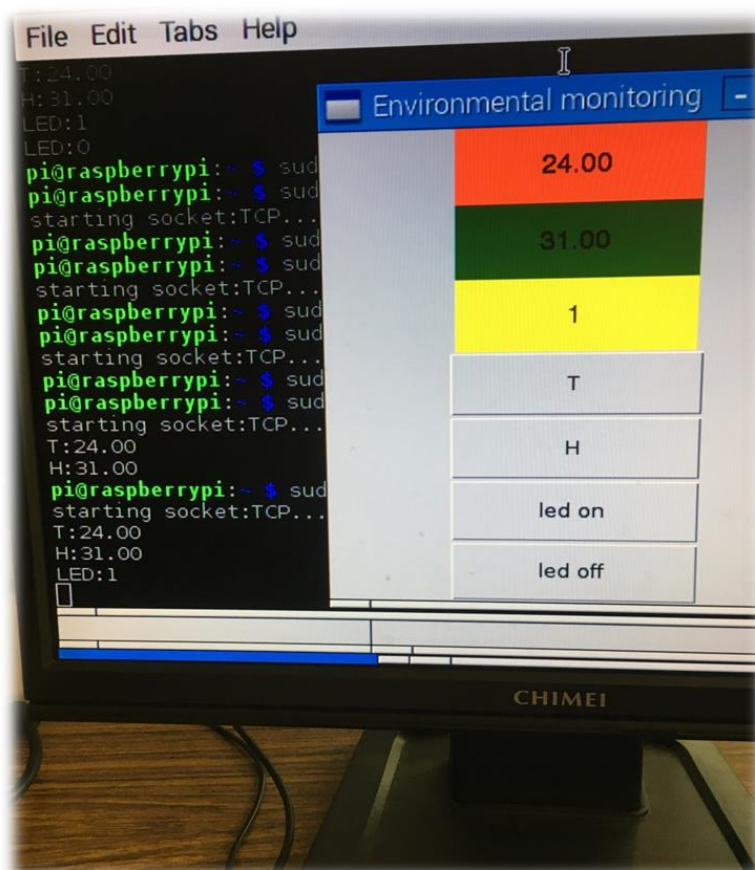
同時，可以看到旁邊的資料夾欄類有許多資料夾，當中的 TEST3 以及 TEST4 是我們分別用來儲存溫度及溼度的資料夾。



圖四

圖五是我們執行軟體時會產生的視窗，這個視窗是利用 Tkinter 的語法所構成的，包括整個視窗的大小(長寬高)，以及有幾個視窗，視窗的顏色，每一個按鈕的按鍵按下去會得到什麼效果，都是我們一行行的程式碼打出來的。

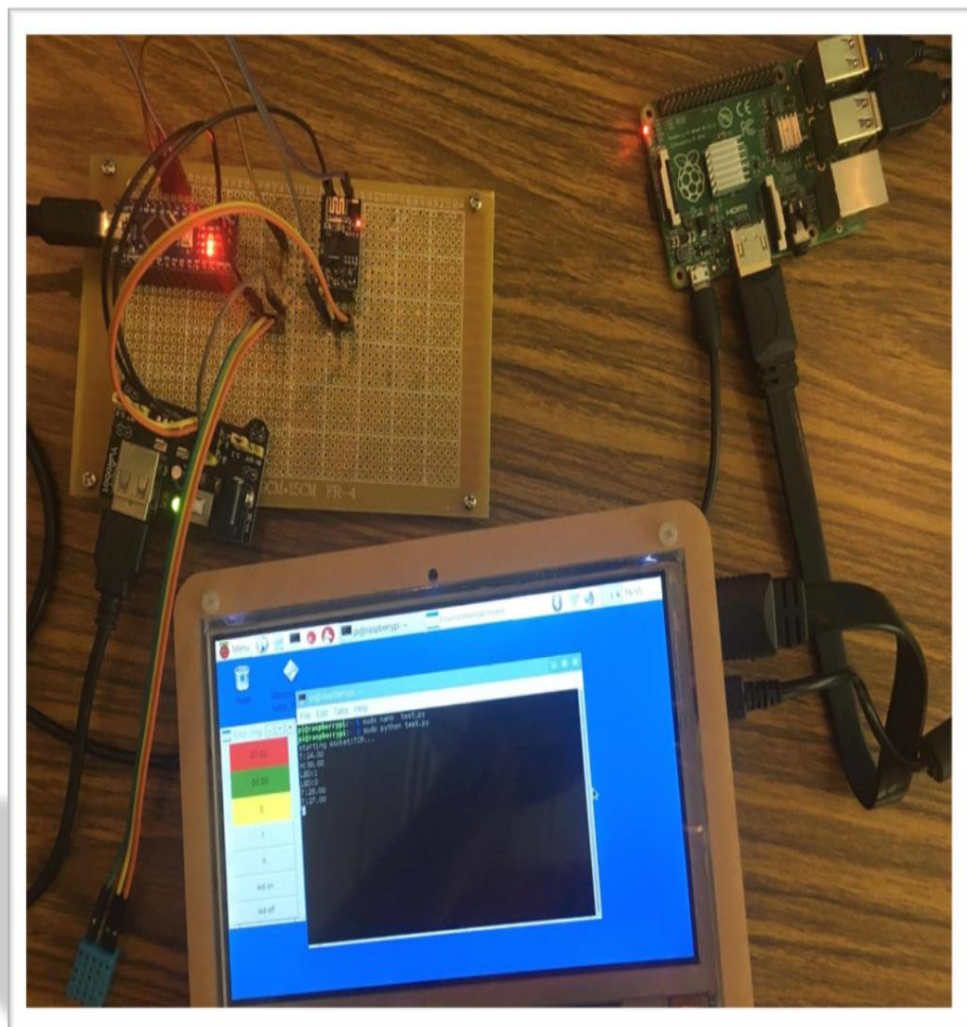
在圖五中，假如我們按下的按鈕是 T，那麼程式就會向 Server 端傳送指令代碼 W(代碼是自行設定的，我們是使用 WXYZ)，那 Server 端接受到這個字元時，就會回傳由 DHT-11 溫溼度感測器測量得到的數值，並且將得到的值顯示在表格上面，相同的按下 H 按鈕也會回傳當下溼度的值到 Client 端，並顯示在表格上，並且還會將得到的數值紀錄下來上傳至 MySQL 裡面的 TEST3 以及 TEST4 資料夾裡面保存(資料夾 TEST3 是用來放置溫度數據，TEST4 則是用來放置溼度數據)比較特別的是按鈕 LED ON，他可以用來控制 LED 的開關，當我們按下這個按鈕，他就會將 LED 燈打開，反之，LED OFF 則是關閉。



圖五

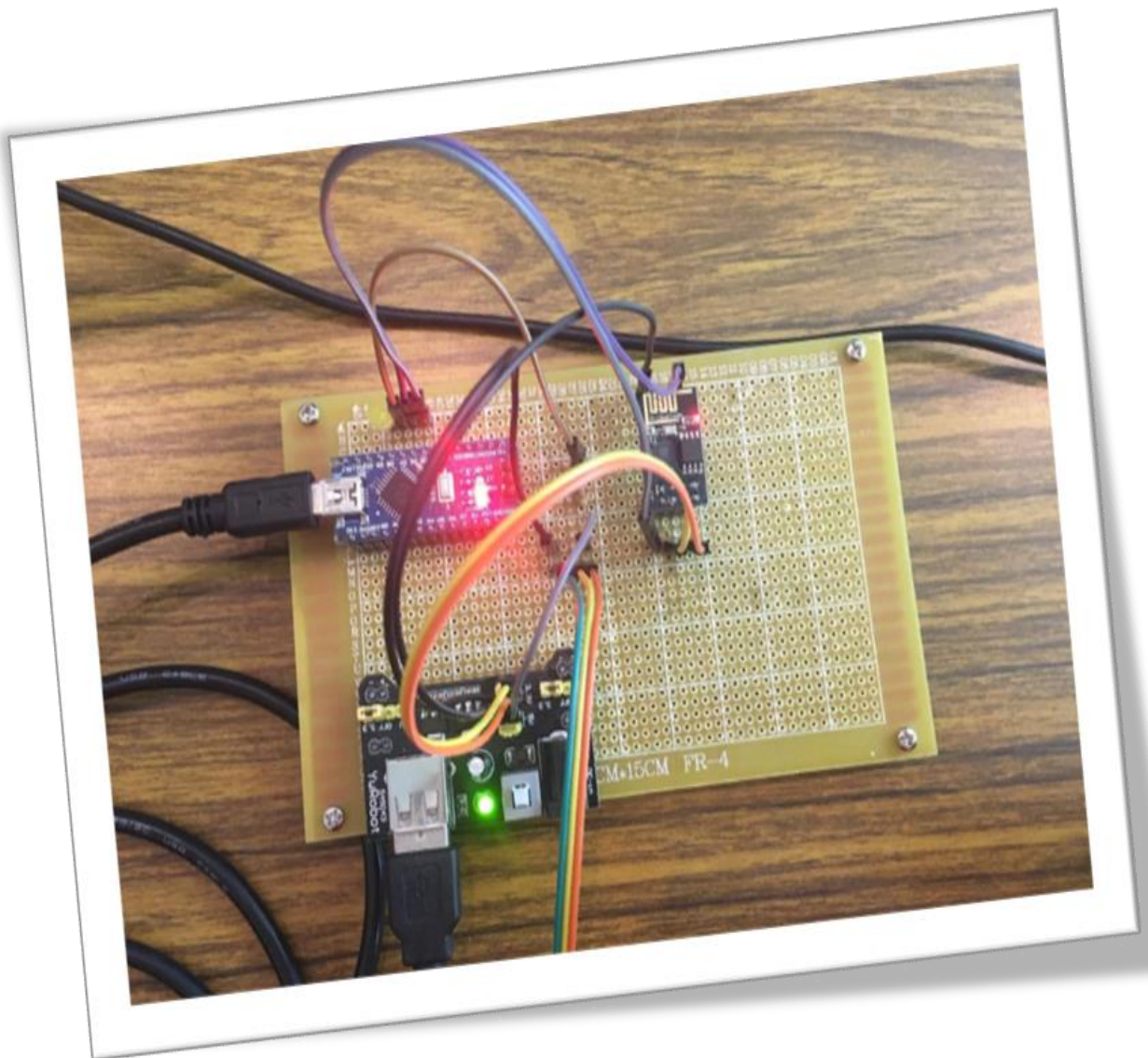
我們可以看到圖六，圖中有我們的 Raspberry pi2 以及 Arduino 的 nano 板還有 ESP-8266 跟電源穩壓模組，我們這個專題是軟體為主，那我們希望將這些東西焊接到電路板上，可以讓他成為一體，而不是零零散散的，也能達到軟體為主，硬體為輔的概念。

在圖六也可以看到我們原來的程式是放在電腦螢幕上執行的，到我們作品成形時，我們將整個作品移植到觸控的螢幕上，可以讓人們更方便使用，更方便收納，也更貼近人們的生活。



圖六

在圖七中我們可以看到我們自己焊接电路板的元件面，雖然焊接及元見面看起來固然簡單，但是翻到背後的焊接面則是有大大的學問，背後的每一條線皆是我們自己一條一條線慢慢配置的，只要有一條線拉錯或者是空焊，那這個电路就不會動作了。



圖七

第六章

結論

近年來受到地球的溫室效應影響，天氣變化越來越大，早晚溫度也有明顯的差別，藉由此作品我們可以方便監控某處當下的溫溼度變化，以利了解各地溫溼度的改變，也讓我們更了解大局觀。

透過此專題，讓我們從中學習到了運用服務端(Linux)架設資料庫(MySQL)以程式語言(Python、Arduino)來操作，雖然在這各程中遇到了許許多多的問題，常常因為一小段程式碼的錯誤，為了要把它找出來，可能一找就是一個晚上，學習方式也是自己買書來參考或者是上網看文章主要以自主學習居多，所以遇到問題我們也都試著自己解決，如果真的遇到問題無法解決，在專題開會的時候也會提出來跟老師討論，在完成專題的同時也期望我們能學以致用。

第七章

給未來學弟妹的建議

這個專題雖然看似簡單，但對完全沒有接觸過的人來說是非常困難的，因為它不單單應用到 Arduino 還用到了 Raspberry pi，尤其是 pi，完完全全是跟以前學的東西是沒有相關的，他用的主要語言是 Python，對沒有深入了解的人可能連聽都沒聽過。專題分為上下學期，我們花了整個上學期的時間在學習 Python，一路過來遇到了無數的 bug 每次除錯都花了大把大把的時間，所以對這方面沒有興趣的同學，要做這方面的專題要三思而後行，反之，如果對這方面有興趣的同學，一定也會樂此不疲，祝未來學地專題順利。

第八章參考資料

- 利用 DHT11 抓取溫度和濕度

<http://ming-shian.blogspot.tw/2014/05/arduino19dht11.html>

- Arduino 教學

<http://thomasprivate.blogspot.tw/2016/02/arduino-esp8266-wifidht-thingspeak.html>

- Arduino 語法參考

http://www.86duino.com/?page_id=2255&lang=TW

- MySQL 語法參考

<https://dev.mysql.com/doc/refman/5.5/en/select.html>

- Python 語法參考

<http://www.runoob.com/python/python-socket.html>

- 菜鳥編程

<http://www.runoob.com/>

- Linux 教學

http://linux.vbird.org/new_linux.php#

- Streaming Tweets from Twitter API v1.1 into MySQL Database with Python

https://www.youtube.com/watch?annotation_id=annotation_2383914567&feature=iv&src_vid=i6PVshqAUGw&v=LC8VpW03Ksw