**目的**

智能濕度提醒精靈包含了兩套智能濕度提醒裝置，一個Gateway (Raspberry Pi)負責將裝置傳來之溫濕度資料上傳至RainbowTree雲端平台。使用者可以自行透過手機APP設定環境的濕度上下限，若當環境濕度未在此範內，手機APP端會發出提醒聲音。

**http://youtu.be/BiFOyHTiC9A**

**材料**

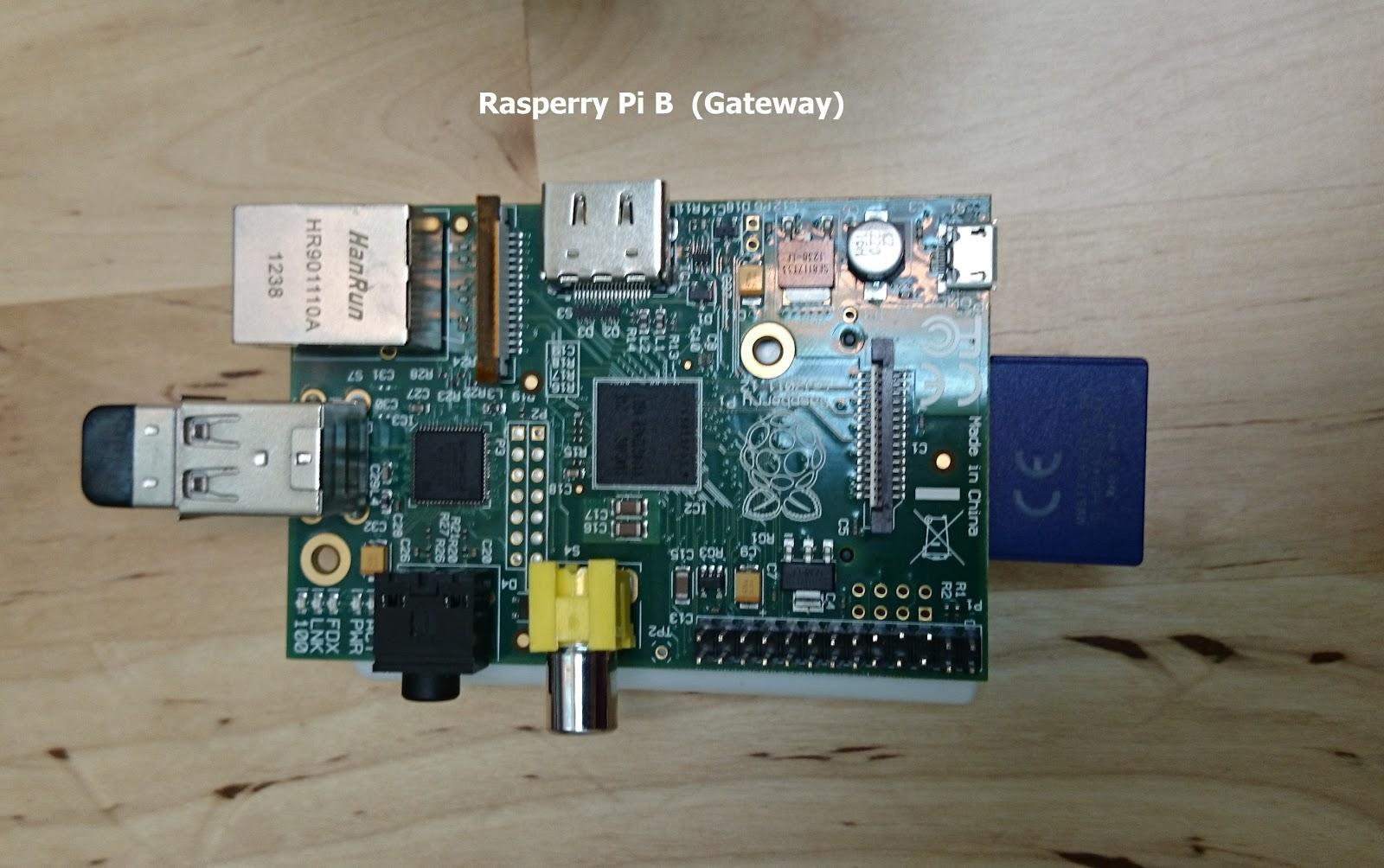
|  |  |
| --- | --- |
| Arduino主板 Seeeduino | × 2 |
| DHT-11 | × 2 |
| Android手機 | × 1 |
| Raspberry Pi | × 1 |
| 杜邦線 | × N |
| 麵包板 | × 2 |

**接線**

請參考以下圖表，以下提供我們的接法，詳細可根據個人的專案設計來決定針腳的接法，但VCC/GND務必不要接反

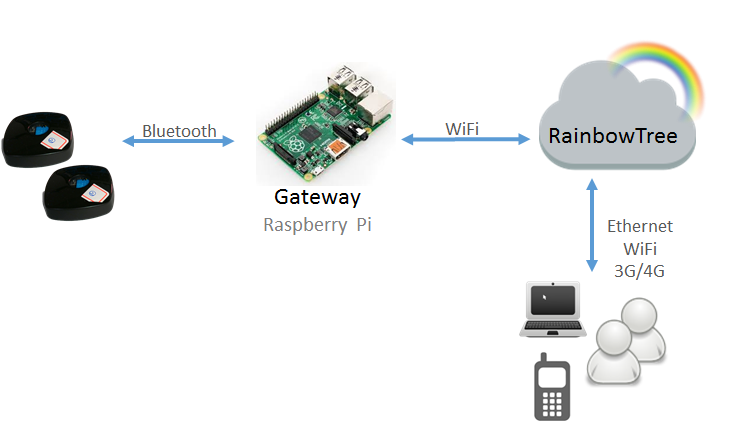
|  |  |
| --- | --- |
| **DHT-11 --- Arduino端**  VCC---5V  DATA---A0  NC---  GND---GND |  |
| **HC-06藍芽模組 --- Arduino端**  VCC---5V  GND---GND  TXD---D2  RXD---D3  \*HC-06的TX需接至Arduino主板上的RX(D2)  HC-06的RX需接至Arduino主板上的TX(D3) | 20140710_121952.jpg |





\*燒錄程式請務必將各模組電源切斷，或是燒錄完畢後，再斷電重新將線路連通，以防模組等燒壞。藍芽模組接至Arduino板上的RX、TX，在燒錄程式時會影響，故請不要接上線路，待燒錄完畢，再將各模組接上通電執行。

**智能濕度提醒精靈運作架構**



兩組智能濕度裝置會定期將所感測到的環境溫濕度，透過藍芽傳輸至Gateway端(RPi)，Gateway端負責將接收到的資料透過WiFi傳送至RainbowTree雲端平台上。使用者可以透過手機或是電腦瀏覽RainbowTree上資料，而在手機APP端可以設定所想要之濕度上下限，當環境濕度未在範圍內，即會發出提醒音效。

**Arduino端程式操作**

溼度裝置的開發，是採用Grove Temperature and Humidity Sensor所提供的library，  
<https://github.com/Seeed-Studio/Grove_Temperature_And_Humidity_Sensor>

請至此下載。  
溼度裝置會定期將感測到的資料透過藍芽傳送，而裝置本身有寫入預設之濕度上下限(30%~65%)，若環境濕度未在範圍內，板子上LED會亮起。

1. 藍芽模組採用SoftwareSerial的連線方式，其pin腳為D2、D3。設定DHT-11連線pin腳，及宣告DHT物件。

|  |
| --- |
| SoftwareSerial BTSerial(2, 3); //rx,tx  #define DHTPIN A0 // what pin we're connected to  #define DHTTYPE DHT11 // DHT 11  DHT dht(DHTPIN, DHTTYPE); |

1. 初始化各模組，及設定對應buad rate。

(HC-06出廠預設baud rate為9600，藍芽設備名稱為HC-06，配對密碼為1234，若想要修改baud rate，請參考HC-06 AT command的修改方式)。

|  |
| --- |
| //==================================setup()====================================  void setup()  {  Serial.begin(5700);  BTSerial.begin(9600);  pinMode(ledPin, OUTPUT);  digitalWrite(ledPin, LOW);  dht.begin();  } |

1. 重複執行Loop內的動作

讀取DHT-11的溫度、及濕度值，再透過BTSerial將資料透過藍芽傳送。

|  |
| --- |
| void loop()  {  //…  float h = dht.readHumidity();  float t = dht.readTemperature();  dtostrf(h,2,2,ftmp);  outputData = outputData+ftmp+",";  dtostrf(t,2,2,ftmp);  outputData = outputData+ftmp+",";    if(h>=upperBound)  {  digitalWrite(ledPin, HIGH);  outputData += "1";  }else if(h<=lowerBound)  {  digitalWrite(ledPin, HIGH);  outputData += "2";  }  else  {  outputData += "0";  }  BTSerial.println(outputData);  //...  } |

**Gateway端 (Raspberry Pi)**

我們在Raspberry Pi上是透過一支Python程式來接收來自裝置端的資料，並同時將資料上傳至RainbowTree平台上。首先先確認Rapberry Pi上的一些套件已安裝完成，並設定好裝置與藍芽間的配對，確保Raspberry Pi已是連網狀態。

\*藍芽及Wifi設定可參考以下文章<http://www.dotblogs.com.tw/bowwowxx/archive/2014/04/17/144774.aspx>

\*藍芽配對問題請參考以下文章

<http://www.correlatedcontent.com/blog/bluetooth-keyboard-on-the-raspberry-pi/>

在python端的實作，主要是透過threading library來接收來自多個裝置的資料，而濕度裝置所對應之RainbowTree上之device id, datastream id等，皆定義於MoistureDevice的物件中。而在上傳RainbowTree平台的部分，我們將資料轉成JSON格式，以http post的方式將JSON格式的資料上傳至雲端平台。

**Android手機端程式實作與操作**

手機端APP部分主要是至RainbowTree平台上讀取最新的資料，再將最新的資料數值顯示於APP頁面上，而若當濕度超過上限值、或是低於下限值，手機端會發出提醒音，頁面的溼度顯示會以紅字顯示。透過點選兩個濕度計的上下限數字值，即會跳出對話框，可輸入所預設定的濕度限制值。



**AndroidManifest.xml**

加入Internet的uses-permission，因APP需至RainbowTree上擷取資料，故需有上網的權限。

|  |
| --- |
| <uses-permission android:name=**"android.permission.INTERNET"** /> |

**MainActivity.java**由於APP需需使用去RainbowTree上讀取新的資料值，我們採用**ScheduledExecutorService類別**來實現，這邊設定週期長度為10秒。

|  |
| --- |
| **import** java**.**util**.**concurrent**.**ScheduledExecutorService**;**  //…  private final ScheduledExecutorService scheduler **=** Executors  **.**newScheduledThreadPool**(**1**);**  //…  scheduler.scheduleAtFixedRate(new Runnable() {  public void run() {  monitoring();  //…  // If you need update UI, simply do this:  runOnUiThread(new Runnable() {  public void run() {  //…  }  }  }, 0, 10, TimeUnit.SECONDS); |

若需使用RainbowTree平台，僅需加入我們的rainbowtree SDK至專案中。在宣告RainbowtreeClient實例時，需加入RainbowTree的IP及平台使用的api key。rainbowtree類別中亦提供各種方法讓我們可以進行資料的上傳(post)、查詢等。詳細請參考rainbowtree SDK。

|  |
| --- |
| **import com.hereapps.rainbowtree.android.RainbowtreeClient;**  **import com.hereapps.rainbowtree.android.listener.JsonArrayListener;**  //…  **RainbowtreeClient client = new RainbowtreeClient("106.186.30.234", "e095f4538bac4870");**  //…  **client.getAllDatastreamStatusAsync(dev\_id[0], new JsonArrayListener() {**  //…  **}**  **});** |

**參考資料**

**Arduino**

[**http://arduino.cc/**](http://arduino.cc/en/Reference/SoftwareSerial)

**SoftwareSerial**

[**http://arduino.cc/en/Reference/SoftwareSerial**](http://arduino.cc/en/Reference/SoftwareSerial)

**藍芽模組HC-06**

[**http://gsyan888.blogspot.tw/2014/03/arduino-hc-06-at-command.html**](http://gsyan888.blogspot.tw/2014/03/arduino-hc-06-at-command.html)

[**http://www.mikrokopter.de/ucwiki/en/HC-06**](http://www.mikrokopter.de/ucwiki/en/HC-06)

**RainbowTree**

[**http://rainbowtree.here-apps.com/**](http://rainbowtree.here-apps.com/)