**目的**

智能偵水度提醒精靈包含了一組智能偵水提醒裝置，一個Gateway (Raspberry Pi)負責將裝置傳來之偵水狀態上傳至RainbowTree雲端平台。偵水裝置是由

一顆偵水Sensor、藍芽模組、及Arduino主板所組成。若當裝置偵測到有水的存在(淹水、滿水位等狀態)，手機APP端即會發出提醒聲音。

[**http://youtu.be/mSDgg5gNE8M**](http://youtu.be/mSDgg5gNE8M)

**嵌入語法**

**<iframe width="560" height="315" src="//www.youtube.com/embed/mSDgg5gNE8M" frameborder="0" allowfullscreen></iframe>**

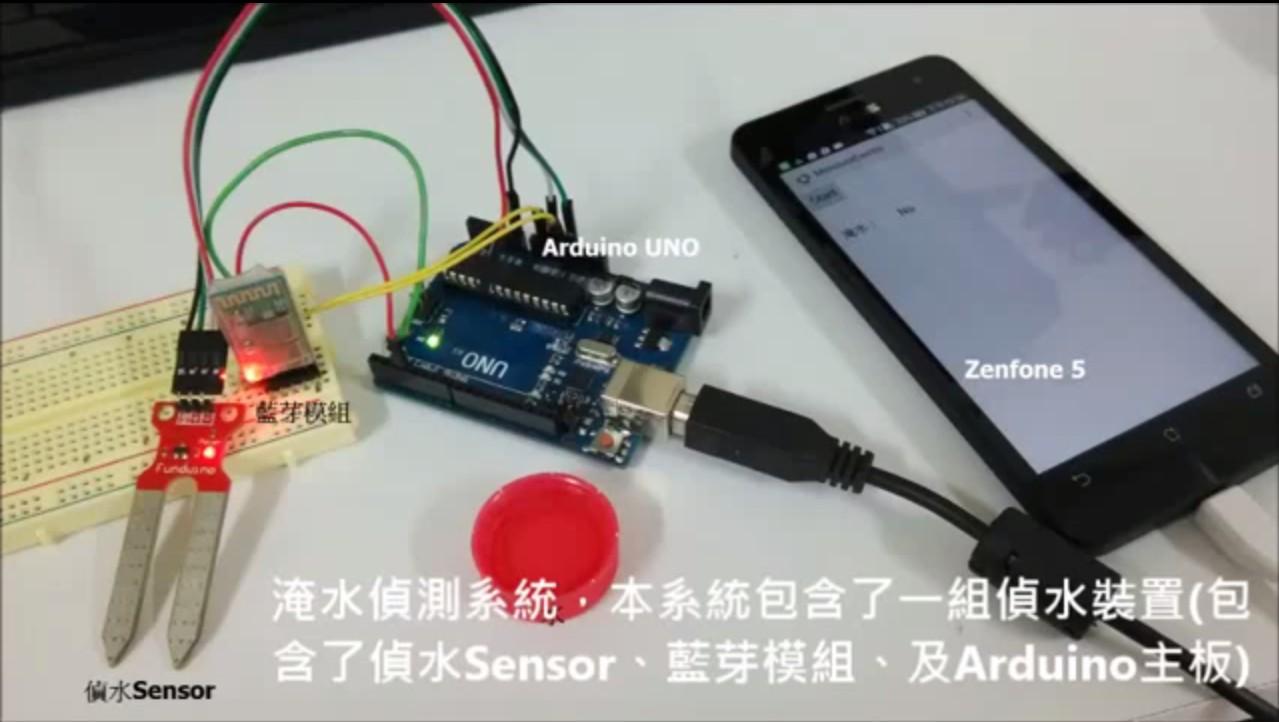
**材料**

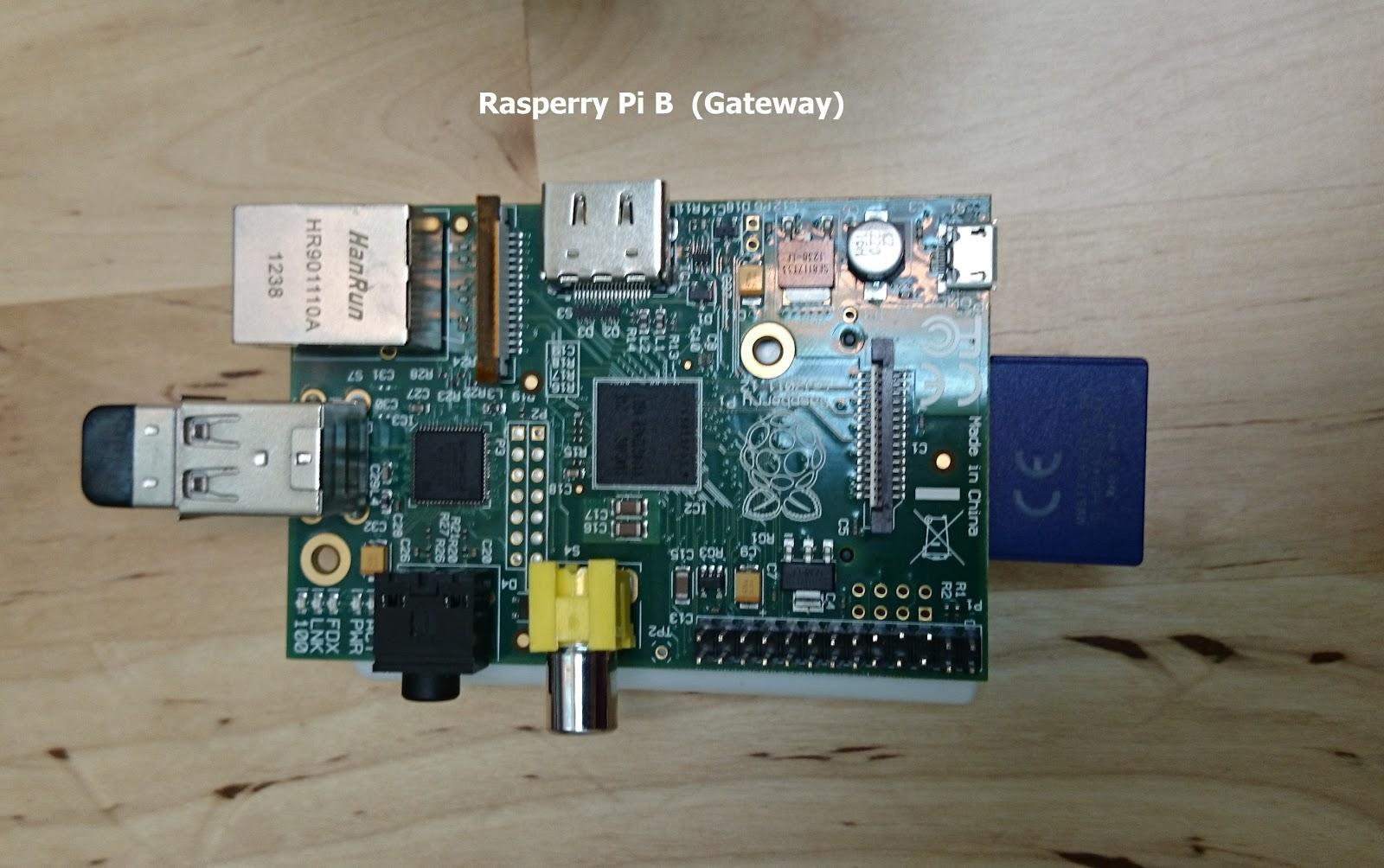
|  |  |
| --- | --- |
| Arduino主板 UNO | × 1 |
| 土壤水分Sensor | × 1 |
| Android手機 | × 1 |
| Raspberry Pi | × 1 |
| 杜邦線 | × N |

**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

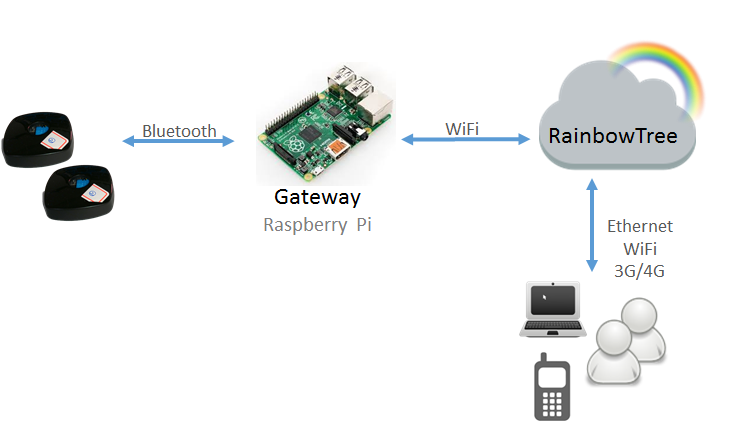
|  |  |
| --- | --- |
| **水份Sensor--- Arduino端**  S --- A0  + ---5V  - --- GND |  |
| **HC-06藍芽模組 --- Arduino端**  VCC---5V  GND---GND  TXD---D2  RXD---D3  \*HC-06的TX需接至Arduino主板上的RX(D2)  HC-06的RX需接至Arduino主板上的TX(D3) | 20140710_121952.jpg |





\*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。

**智能偵水提醒精靈運作架構**



偵水裝置會偵測是否有水存在，並將所感測到的資料，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，當裝置偵測到有水時(淹水、滿水位等狀態)，即會發出提醒音效。

**Arduino端程式操作**

偵水裝置會檢查裝置兩端電極是否有接觸到水，並將感測到的資料透過藍芽傳送，若偵測到有水，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。

|  |
| --- |
| SoftwareSerial BTSerial(2, 3); //rx,tx |

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(57600);  BTserial.begin(9600);  pinMode(ledpin, OUTPUT);  digitalWrite(ledpin, LOW);  } |

1. 重複執行Loop內的動作

讀取偵水Sensor的數值，判斷是否有水存在(value>0)，若有水，LED會亮起；反之則滅，再將此讀數資料透過BTSerial將資料透過藍芽傳送。

|  |
| --- |
| void loop()  {  //…  value = analogRead(0);    if(value>0)  {  flag = 1;  digitalWrite(ledpin, HIGH);  }  else  {  flag = 0;  digitalWrite(ledpin, LOW);  }  output = output+flag+","+value;  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>

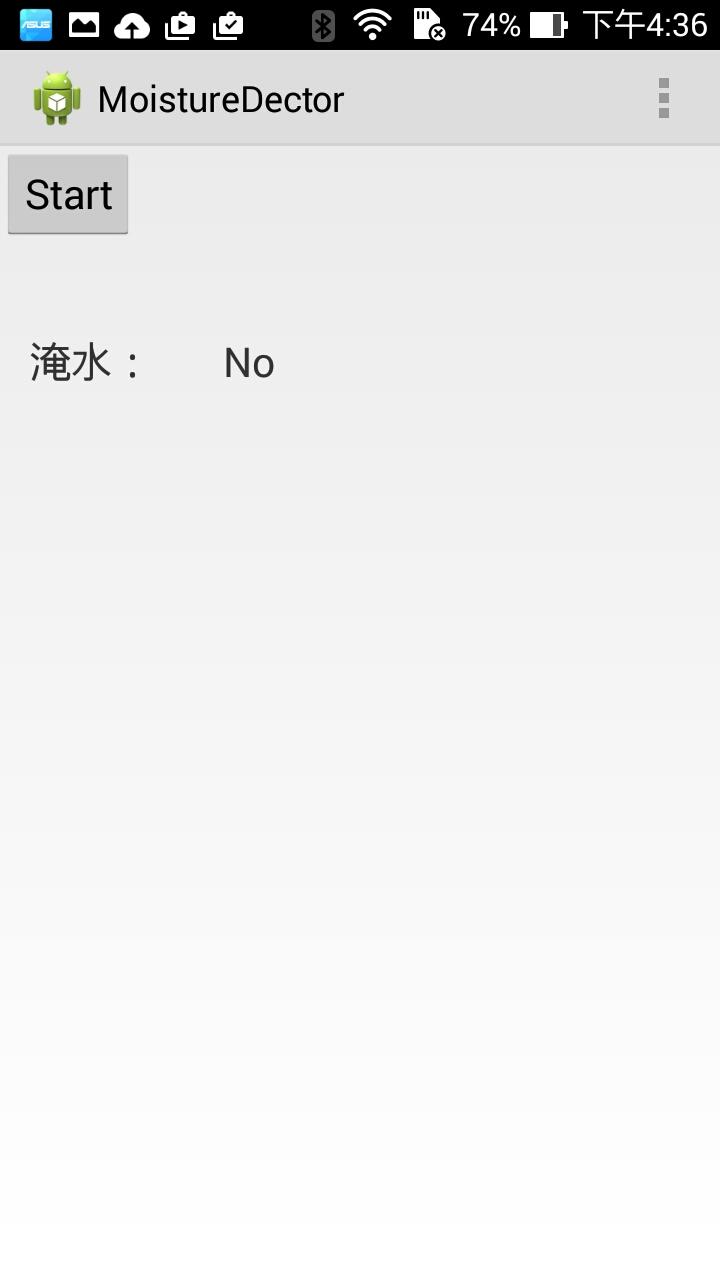
\*藍芽配對問題請參考以下文章

<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，而若當裝置端偵測到有水的狀態，手機端會發出提醒音，淹水資訊會以紅字顯示。



**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1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, 1, 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/)