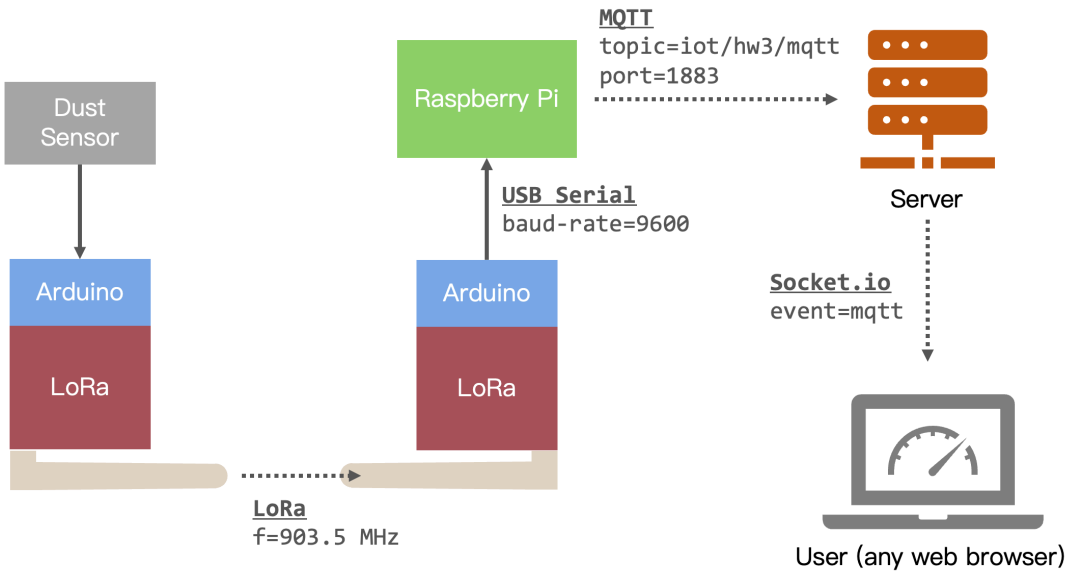
IOT HW3 REPORT

B08901097 徐有齊
B08901132 任瑨洋

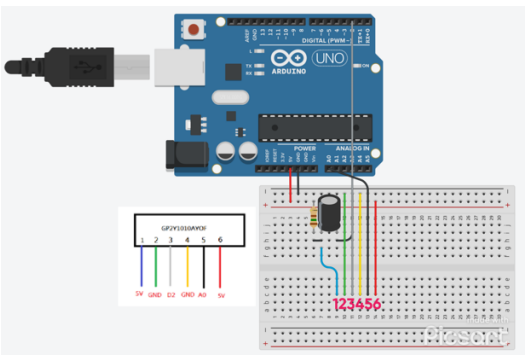
原始碼

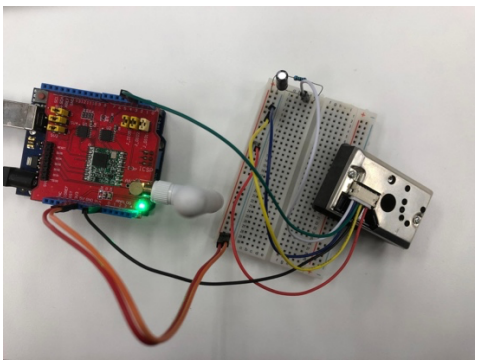
https://github.com/MartianSheep/Intro2IoT_HW3

系統架構



我們使用 Arduino Uno 、 Dragino LoRa shield 和 SHARP GP2Y1014AU PM2.5 Sensor 架設遠端 sensor 偵測空氣品質。此處的 Arduino Uno 將會是 LoRa 傳輸中的 client 。 以下是 PM2.5 sensor 連接到 Arduino Uno 上的腳位圖和實際圖片：





我們使用一個 Arduino Uno 作為 LoRa Server ，接收來自 Dragino LoRa shield 的訊號，並將訊息以 UART 方式透過 USB 傳給 RPi ，RPi 再透過專案資料

夾中的 PM25.py 把 LoRa Server 傳過來的資料轉成 MQTT 。

我們另外使用一個 node 伺服器作為 MQTT 的 subscriber，並把 MQTT 資訊轉換成更常見的 Web Socket Protocol，這樣便能使 UI 的設計更加簡單也更通用。最後在使用者介面上，我們採用 React 作為 UI 設計框架。

各項設置

此作業所有的原始碼皆可以在 https://github.com/MartianSheep/Intro2IoT_HW3 存取。

- **Arduino**

  以下為操作步驟：

  1. 於燒錄用的電腦上安裝 Arduino IDE 。
  2. 從 http://www.airspayce.com/mikem/arduino/RadioHead/RadioHead-1.121.zip 下載 RadioHead 程式庫，並放在 C:\Users%USERNAME%\Documents\Arduino\libraries\ 。
  3. 改動 RadioHead 程式庫中的 RH_RF95.cpp 與 RH_RF95.h ，將這兩個檔案替換成我們於 github 上提供的 Arduino\Modified rf95 files\ 中的 RH_RF95.cpp 與 RH_RF95.h 。 (備註：我們更改過其中的 rf95.init() ，使其可以更輕鬆地更改 LoRa 頻率。)
  4. 關於 LoRa client 端，燒錄 Arduino\PM2.5_client 中的 Arduino code 至 LoRa client 端的 Uno 開發版上，使其可以讀取 PM2.5 sensor 並傳至 LoRa server 。
  5. 關於 LoRa server 端，燒錄 Arduino\PM2.5_server 中的 Arduino 程式碼至 LoRa server 端的 Uno 開發版上，使其可以接收來自 client 的訊息並傳送至 RPi。

  關於各項係數：

  - LoRa 方面，我們使用 903.5 MHz 作為我們的通訊頻段。
  - Arduino Uno 的 UART 溝通方面，我們使用 9600 baud-rate 溝通，LoRa client 端的 debug message 和 LoRa server 端給 RPi 訊息皆是。
  - 更改過的 RH_RF95 方面，原本的 rf95.init() 並沒有參數傳入，我們將其改為 rf95.init(float initFrequency = 434.0) 後稍微更改其內容，使其在兼容原本程式碼的 434 MHz 的情況下允許我們輕鬆地更改傳輸頻率。

- **Raspberry Pi**

  以下為操作步驟：

  1. 於燒錄用的電腦上安裝 Raspberry Pi Imager 。
  2. 將 RPi 的 SD Card 插到電腦上，開啟 Imager ，選擇「擦除」，將 SD Card 格式化。
  3. 於 Imager 上選擇 Raspberry Pi OS (32-bit) ，燒錄至 SD Card 。
  4. 將 SD Card 插入 RPi ，上電，按螢幕指示完成基本設定。
  5. 於 RPi 上開啟 terminal ，執行以下指令：

     ```
     sudo apt-get update
     sudo apt-get upgrade
     sudo apt-get install mosquitto
     sudo apt-get install mosquitto-clients
     pip3 install paho-mqtt
     git clone https://github.com/MartianSheep/Intro2IoT_HW3
     cd Intro2IoT_HW3/RPi/
     ```

  6. 確定 LoRa server 端的 Arduino Uno 已經插上 RPi 。使用 ls -l \dev 來確定此 Arduino Uno 使用哪一個 port 。我們在實作時 RPi 皆給 /dev/ttyACM0 ，因此程式中預設使用此 port 。若有不同，請在 RPi/PM25.py 中第 10 行更改 `com_port` 參數。

  7. 使用 `python3 PM25.py` 執行程式碼。

  關於各項係數：

  - MQTT broker 的部份，在 RPi 安裝 mosquitto 後便會自動啟動。
  - 這邊由 PM25.py 擔任 MQTT publisher 的部分，這支程式會從 COM port 抓取從 Arduino Uno 傳來的資料，並將其 publish 出去。
  - 由於 MQTT broker 本身便架在 RPi 上，因此 PM25.py 中 broker 的位址便是 localhost ， port 使用預設的 1883 。

- **伺服器和前端**

由於沒有實際的伺服器，我們在同一台電腦同時運行伺服器以及前端，但只要是能夠互相透過網際網路連接的設備，都可以把伺服器跟前端分開運行。

以下為操作步驟：

1. 確認 RPi 的 IP 位置
   我們建議把 RPi 跟運行伺服器的電腦連到同一個區域網路下，這樣就能保證兩者可以互相連線。後續設置需要知道 RPi，也就是 MQTT publisher 的 IP 位置，可以用 `ifconfig` 指令進行確認。

2. 安裝 Node 和 Yarn（若已有安裝則不需要操作此步驟）
   (1) 安裝 Node (https://nodejs.org/en/download/)，版本建議為 16
   (2) 開啟一個終端機，執行 `npm install -g yarn`
   (3) 在終端機執行 `yarn --version` 檢查是否安裝成功。

3. 伺服器設置
   (1) 進到專案資料夾，打開 ./mqtt-server/app.js 檔案，把 RPi 的 IP 位置填入第 35 行：
   ```
   var client = mqtt.connect('mqtt://<IP ADDR>', opt);
   ```

   (2) 使用終端機進入 ./mqtt-server ，執行以下指令：

   ```
   npm install
   npm start
   ```

   完成指令後，伺服器會監聽 port 1883 的 MQTT 資訊，並轉成 Web Socket 協定於 port 8000 送出。

4. Client 設置
   (1) 使用終端機進入 ./mqtt-client ，執行以下指令：

   ```
   yarn
   yarn start
   ```

(2) 此時瀏覽器會自動跳出對應的頁面，並顯示 sensor 的讀值：