[嵌入式系統設計](https://flipclass.stust.edu.tw/course/13883)第十一次作業

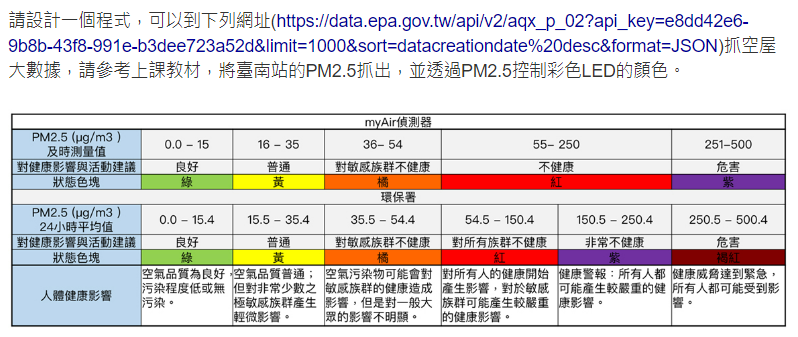
班級:五專資工三甲

姓名:陳紋誼

學號:5A9G0006

指導老師:吳建中 老師

1. 題目



1. 程式

#include <Arduino.h>

#include <WiFi.h>

#include <WiFiMulti.h>

#include <HTTPClient.h>

#include <ArduinoJson.h>

#define USE\_SERIAL Serial

WiFiMulti wifiMulti;

String site, county, datacreationdate, itemunit;

int pm25;

void setup() {

USE\_SERIAL.begin(115200);

USE\_SERIAL.println();

USE\_SERIAL.println();

USE\_SERIAL.println();

for (uint8\_t t = 4; t > 0; t--) {

USE\_SERIAL.printf("[SETUP] WAIT %d...\n", t);

USE\_SERIAL.flush();

delay(1000);

}

wifiMulti.addAP("WY1010", "20041010311");

}

void loop() {

// wait for WiFi connection

if ((wifiMulti.run() == WL\_CONNECTED)) {

HTTPClient http;

USE\_SERIAL.print("[HTTP] begin...\n");

// configure traged server and url

//http.begin("https://www.howsmyssl.com/a/check", ca); //HTTPS

http.begin("https://data.epa.gov.tw/api/v2/aqx\_p\_02?api\_key=e8dd42e6-9b8b-43f8-991e-b3dee723a52d&limit=1000&sort=datacreationdate%20desc&format=JSON"); //HTTP

USE\_SERIAL.print("[HTTP] GET...\n");

// start connection and send HTTP header

int httpCode = http.GET();

// httpCode will be negative on error

if (httpCode > 0) {

// HTTP header has been send and Server response header has been handled

USE\_SERIAL.printf("[HTTP] GET... code: %d\n", httpCode);

// file found at server

if (httpCode == HTTP\_CODE\_OK) {

String payload = http.getString();

//USE\_SERIAL.println(payload);

//new

DynamicJsonDocument doc(12288);

DeserializationError error = deserializeJson(doc, payload);

if (error) {

Serial.print("deserializeJson() failed: ");

Serial.println(error.c\_str());

return;

}

for (JsonObject field : doc["fields"].as<JsonArray>()) {

const char\* field\_id = field["id"]; // "site", "county", "pm25", "datacreationdate", "itemunit"

const char\* field\_type = field["type"]; // "text", "text", "text", "text", "text"

const char\* field\_info\_label = field["info"]["label"]; // "測站名稱", "縣市名稱", "細懸浮微粒濃度", "資料建置日期", "測項單位"

}

const char\* resource\_id = doc["resource\_id"]; // "c1f31192-babd-4105-b880-a4c2e23a3276"

const char\* extras\_api\_key = doc["\_\_extras"]["api\_key"]; // "e8dd42e6-9b8b-43f8-991e-b3dee723a52d"

bool include\_total = doc["include\_total"]; // true

const char\* total = doc["total"]; // "78"

const char\* resource\_format = doc["resource\_format"]; // "object"

const char\* limit = doc["limit"]; // "1000"

const char\* offset = doc["offset"]; // "0"

const char\* links\_start = doc["\_links"]["start"];

const char\* links\_next = doc["\_links"]["next"];

for (JsonObject record : doc["records"].as<JsonArray>()) {

const char\* record\_site = record["site"]; // "大城", "富貴角", "麥寮", "關山", "馬公", "金門", "馬祖", "埔里", "復興", ...

const char\* record\_county = record["county"]; // "彰化縣", "新北市", "雲林縣", "臺東縣", "澎湖縣", "金門縣", "連江縣", "南投縣", ...

const char\* record\_pm25 = record["pm25"]; // "14", "6", "14", "2", "12", "20", "15", "17", "28", "9", ...

const char\* record\_datacreationdate = record["datacreationdate"]; // "2022-11-19 14:00", "2022-11-19 ...

const char\* record\_itemunit = record["itemunit"]; // "μg/m3", "μg/m3", "μg/m3", "μg/m3", "μg/m3", ...

if (record["site"] == "臺南") {

site = record["site"].as<String>();

USE\_SERIAL.print("測站："); USE\_SERIAL.println(site);

county = record["county"].as<String>();

USE\_SERIAL.print("縣市："); USE\_SERIAL.println(county);

pm25 = record["pm25"].as<int>();

USE\_SERIAL.print("PM2.5："); USE\_SERIAL.println(pm25);

datacreationdate = record["datacreationdate"].as<String>();

USE\_SERIAL.print("資料建置日期："); USE\_SERIAL.println(datacreationdate);

itemunit = record["itemunit"].as<String>();

USE\_SERIAL.print("測項單位："); USE\_SERIAL.println(itemunit);

}

}

if (pm25 <= 500 && pm25 >= 251) {

USE\_SERIAL.println("PM2.5值251~500，紫燈");

} else if (pm25 >= 55) {

USE\_SERIAL.println("PM2.5值55~250，紅燈");

} else if (pm25 >= 36) {

USE\_SERIAL.println("PM2.5值36~54，橘燈");

} else if (pm25 >= 16) {

USE\_SERIAL.println("PM2.5值16~35，黃燈");

} else {

USE\_SERIAL.println("PM2.5值0.0~15，綠燈");

}

}

} else {

USE\_SERIAL.printf("[HTTP] GET... failed, error: %s\n", http.errorToString(httpCode).c\_str());

}

http.end();

}

delay(5000);

}

1. 程式說明

可以到網址抓空汙大數據，將臺南的PM2.5抓出，並透過PM2.5控制彩色LED的顏色

1. 執行結果

一張含有 文字 的圖片

自動產生的描述

影片連結： <https://youtu.be/3LyJqpX6aGM>