

INSPIRED

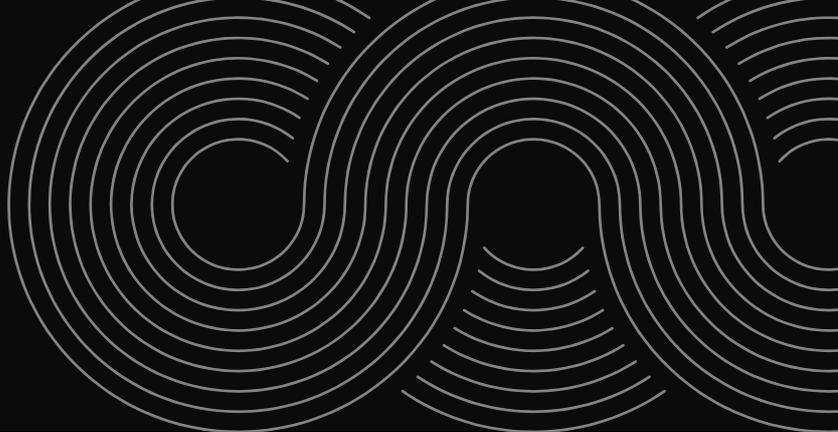


**LET'S HAVE A
FUN COURSE.**

This handouts is a publication
focused on IoT Project and design.

This course explores IoT lighting,
weather data, and recording
module integration.

TABLE OF CONTENTS



1

物聯網燈控的原理氣象資訊平台

實驗器材介紹	1
在ESP8266上架設wifi基站與網頁控制	3
網頁內容設計與測試	6
利用HTML網頁控制燈色	8
網頁控制原理與美化	11

2

空氣品質資訊站設計

平台的選擇與設定	15
程式範例	15

3

硬體互動與功能切換

錄音模組的控制	17
按鈕電路與程式	19
多模式切換程式架構	20

4

外觀設計與組裝

外觀設計流程	21
外觀設計工具	23
組裝說明	25

物聯網燈控原理

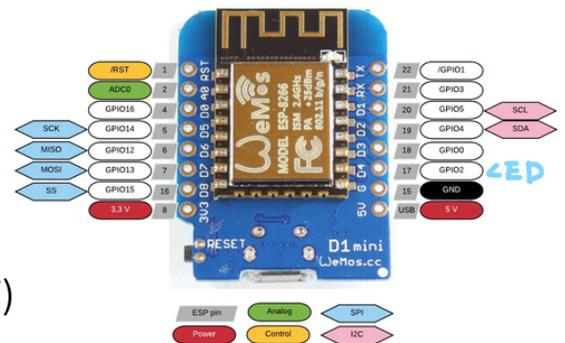
本次工作坊主題以燈塔造型為創作發想，將照亮全台灣各地碼頭，引領討海漁民方向的燈塔作為主題象徵，隱喻燈塔對船舶的引導就像家對人們的溫暖和歸屬感，同時設計加入聯網控制開發板，另外設計語音留言功能及全彩燈光控制模組，將燈塔與留言合而為一，創作出獨一無二的生活應用功能。

本次課程除了向大家介紹程式的運作和原理，我們也向各位介紹外觀結構是如何被設計和製作出來的

一、實驗器材介紹

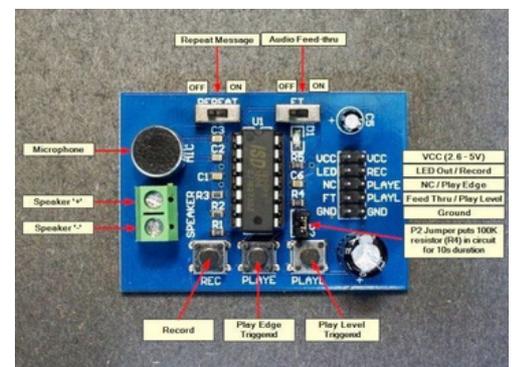
1. 控制板：D1 mini

- 控制晶片：ESP8266
- 工作電壓：3V
- 數位I/O腳位數量：11
- 類比I/O腳位數量：1(最大輸入電壓3.3V)



2. 錄放音模組

- 板載ISD1820晶片
- 板載咪頭，可以直接錄音
- 可以進行10秒左右的語音錄放
- 高品質自然的語音還原，可用作喊話器模組
- 帶迴圈播放，點動播放，單遍播放功能
- 晶片主要的引腳已經引出
 - 可以通過單晶片(如ESP8266)控制操作
- 工作電壓：3~5V
- 板子尺寸54(mm)x38(mm)



物聯網燈控原理

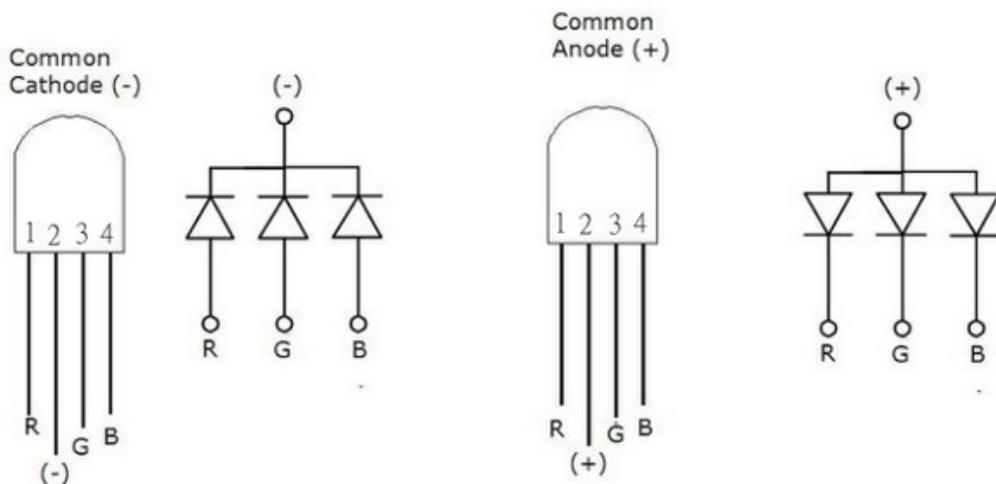
3. 杜邦線

- 杜邦線又稱做跳線(Jumper)
- 搭配麵包板及實驗用零件上的接角可以讓我們快速的接線。
- 市售的杜邦線共有以下三種款式：
- 公-公
- 公-母
- 母-母



4. RGB共陰LED燈

RGB燈可分為共陽和共陰兩種，由於我們使用D1 mini的腳位來輸出控制，因此需要使用共陰RGB。



二、在ESP8266上架設WiFi基站與網頁控制

ESP8266是一款價格經濟、功能強大的微控制器，它內建了WiFi功能，因此成為物聯網項目的首選。除了能夠作為WiFi客戶端連接其他網絡外，ESP8266還可以作為獨立的WiFi基站（或稱作熱點），允許其他設備連接它。

當ESP8266設置為WiFi基站模式時，它可以建立自己的無線網絡，並分配IP地址給連接的客戶端。這使得ESP8266成為一個理想的選擇，用於建立臨時的、本地的無線通信網絡。這種模式尤其適合於那些不需要連接到外部網絡，但需要多個設備間互相通訊的應用，例如家居自動化、無人機控制、或其他即時通信項目。

此外，利用ESP8266作為WiFi基站，還可以實現如網頁伺服器、MQTT代理等功能，使其成為真正的物聯網中樞。

現在，我們來看看如何將esp8266變成一個可以用手機控制開關的燈吧！
我們會學到如何利用HTTP來控制ESP8266

```
1  #include <ESP8266WiFi.h> // 匯入ESP8266WiFi庫
2  #include <ESP8266WebServer.h> // 匯入ESP8266WebServer庫
3
4  // 設定熱點的名稱和密碼
5  const char* ssid = "fangGu_LED"; // 熱點的名稱(請務必改成自己的)
6  const char* password = "12345678"; // 熱點的密碼
7
8  ESP8266WebServer server(80); // 創建一個在80端口的Web伺服器對象
9
10 int ledR = D7; // 定義紅色 LED 燈的腳位
11
12 // 創建一個HTML頁面，該頁面有兩個按鈕，用於開啟和關閉LED燈
13 String html = "<html>\
14 <head>\
15   <title>LED Control</title>\
16 </head>\
17 <body>\
18   <h1>LED Control</h1>\
19   <button onclick=\"fetch('/led/on')\">Turn On</button>\
20   <button onclick=\"fetch('/led/off')\">Turn Off</button>\
21 </body>\
22 </html>";
23
```

```

24 void setup(void){
25     pinMode(ledR, OUTPUT); // 設定LED燈腳位為輸出模式
26     digitalWrite(ledR, LOW); // 預設LED燈為關閉狀態
27
28     Serial.begin(115200); // 開始序列通訊，並設定波特率為115200
29
30     WiFi.softAP(ssid, password); // 將ESP8266設定為熱點模式
31     IPAddress myIP = WiFi.softAPIP(); // 獲取ESP8266的IP地址
32     Serial.print("AP IP address: "); // 在序列監視器上打印IP地址
33     Serial.println(myIP);
34
35     // 當在瀏覽器輸入ESP8266的IP地址時，將顯示html頁面
36     server.on("/", []() {
37         server.send(200, "text/html", html);
38     });
39     // 處理開燈請求
40     server.on("/led/on", []() {
41         digitalWrite(ledR, HIGH); // 開燈
42         server.send(200, "text/plain", "LED is on"); // 在瀏覽器回應LED燈已經開啟
43     });
44     // 處理關燈請求
45     server.on("/led/off", []() {
46         digitalWrite(ledR, LOW); // 關燈
47         server.send(200, "text/plain", "LED is off"); // 在瀏覽器回應LED燈已經關閉
48     });
49
50     server.begin(); // 啟動服務器
51 }
52
53 void loop(void){
54     server.handleClient(); // 處理來自客戶端的請求
55 }
56

```

使用手機/平板或電腦打開wifi，連上程式第五行的熱點名稱後，打開瀏覽器輸入192.168.4.1，即可看到以下畫面。

LED Control

HINT → 本範例網頁語法介紹

`<html>`：HTML文件以這個標籤開始，表示這是一個HTML程式碼。

`<head>`：頭部標籤。它包含了所有的頭部標籤元素，比如 `<title>`、`<style>`、`<meta>` 等。

`<title>`：網頁標題標籤。它的內容會顯示在瀏覽器的標題欄或標籤上。在這裡，它的內容是 "LED Control"。

`</head>`：頭部標籤的結束。

`<body>`：主體標籤。網頁的所有內容，比如文字、圖片、連結、按鈕等，都會放在這裡。

`<h1>`：一級標題標籤。在這裡，它的內容是 "LED Control"。瀏覽器會將 `<h1>` 到 `<h6>` 的內容顯示為不同級別的標題，`<h1>` 是最大的標題。

`<button>`：按鈕標籤。用於創建一個可點擊的按鈕。在這裡，我們有兩個按鈕，分別有 "Turn On" 和 "Turn Off" 的文字。

`onclick`：一個JavaScript事件屬性，當點擊該元素時會觸發其中的JavaScript代碼。在這裡，它會觸發 `fetch('/led/on')` 和

`fetch('/led/off')` 的操作，即向伺服器發送開啟和關閉LED燈的請求。

`</body>`：主體標籤的結束。

`</html>`：HTML文件的結束。

三、網頁內容設計與測試

網頁內容設計不僅僅是關於美觀的視覺呈現，更是如何有效地傳遞資訊、引導用戶流程，以及創建一個流暢的用戶體驗。從選擇合適的顏色方案和字體，到創建結構清晰的佈局和易於導航的介面，每一步都需要細心的考量。

而網頁測試則是網頁設計的關鍵組成部分。這涉及檢查網站在不同的瀏覽器、裝置和操作系統上的兼容性，確保無論用戶使用何種方式訪問，都能獲得一致和優質的體驗。測試還包括檢查網頁的功能性，如鏈接、表單、和其他互動元素是否正常運作。

網頁內容設計與測試是確保一個網站不僅外觀吸引，而且功能健全、用戶友善的必要步驟。

若每次都要上傳到ESP8266才能看到介面就太麻煩了
本節我們將使用CodePen來及時預覽網頁介面
網址為：<https://codepen.io/>

3.1 按鈕太小?你需要使用CSS規範網頁元件

- 1 我們可以在HTML的<head>標籤中加入<style>標籤來定義 CSS 樣式。
- 2 body元素的font-size為30像素，使得網頁中的文字變大。
- 3 設定button元素的font-size為30像素，並且設定了height和width使得按鈕變大，
- 4 margin則是設定按鈕之間的間距。這些都可以根據你的需求進一步調整。

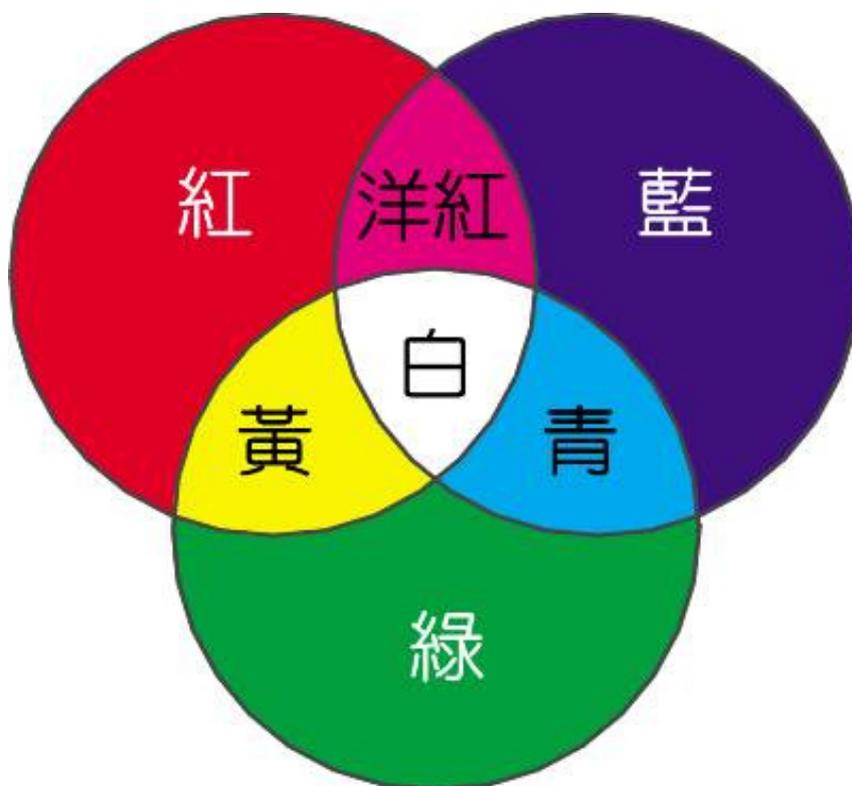
修改完後效果如下：

```
1  #include <ESP8266WiFi.h> // 引入 ESP8266WiFi 庫
2  #include <ESP8266WebServer.h> // 引入 ESP8266WebServer 庫
3
4  const char* ssid = "fangGu_LED"; // 熱點的名稱(請務必改成自己的)
5  const char* password = "12345678"; // 熱點密碼
6
7  ESP8266WebServer server(80); // 創建一個在 80 端口的伺服器
8
9  int ledR = D7; // 定義 LED 腳位
10
11 // 定義網頁 HTML 內容
12 String html = "<html>\
13 <head>\
14   <title>LED Control</title>\
15   <style>\
16     body { font-size: 30px; }\
17     button { font-size: 30px; height: 60px; width: 200px; margin: 10px; }\
18   </style>\
19 </head>\
20 <body>\
21   <h1>LED Control</h1>\
22   <button onclick=\"fetch('/led/on')\">Turn On</button>\
23   <button onclick=\"fetch('/led/off')\">Turn Off</button>\
24 </body>\
25 </html>";
26
27 void setup(void){
28   pinMode(ledR, OUTPUT); // 將 LED 腳位設置為輸出模式
29   digitalWrite(ledR, LOW); // 初始設置 LED 為關閉狀態
30
31   Serial.begin(115200); // 開始串行通信，並設定波特率為 115200
32
33   WiFi.softAP(ssid, password); // 設置 ESP8266 為 AP 模式並設定名稱和密碼
34   IPAddress myIP = WiFi.softAPIP(); // 獲取 ESP8266 的 IP 地址
35   Serial.print("AP IP address: ");
36   Serial.println(myIP); // 在串行監視器打印 IP 地址
37
38   // 處理根路徑的 HTTP 請求
39   server.on("/", []() {
40     server.send(200, "text/html", html);
41   });
42
43   // 處理開啟 LED 的 HTTP 請求
44   server.on("/led/on", []() {
45     digitalWrite(ledR, HIGH); // 開啟 LED
46     server.send(200, "text/plain", "LED is on"); // 發送回應
47   });
48
49   // 處理關閉 LED 的 HTTP 請求
50   server.on("/led/off", []() {
51     digitalWrite(ledR, LOW); // 關閉 LED
52     server.send(200, "text/plain", "LED is off"); // 發送回應
53   });
```

```
54  
55     server.begin(); // 啟動伺服器  
56 }  
57  
58 void loop(void){  
59     server.handleClient(); // 處理客戶端的請求  
60 }  
61
```

我們可以透過調整<style></style>內的數值來控制文字大小

四、利用HTML網頁控制燈色



目前程式只能控制紅色燈光，
請根據上一個範例並利用三原色的原理來創造屬於你的燈色吧!

HINT → 觀察程式中創造一個按鈕所需的語法並新增按鈕控制三原色

! 複製html進CodePen，記得先將HTML中每一行結尾的"\"刪掉

4.1 將LED燈變成你的RGB調色器

現在我們知道透過程式控制RGB，我們可以排列出不同的顏色，如果我們想要無段的條整RGB數值來達到調色器的效果呢？

我們需要根據原本的程式做一點小修改：

1. 網頁須由原來的按鈕變成滑動條
2. 腳位輸出由原本的數位輸出(digitalWrite) 變成 類比輸出(analogWrite)

完成後如下：

```
// 引入需要的庫
#include <ESP8266WiFi.h>
#include <ESP8266WebServer.h>

// 定義無線網路名稱 (SSID) 和密碼
const char* ssid = "fangGu_LED";
const char* password = "12345678";

// 建立一個在端口 80 上運行的 web 服務器對象
ESP8266WebServer server(80);

// 定義 RGB LED 燈的腳位
int ledR = D7; // 紅色 LED 燈腳位
int ledG = D6; // 綠色 LED 燈腳位
int ledB = D5; // 藍色 LED 燈腳位

// 定義網頁 HTML 內容
String html = "<html>\n\
<head>\n\
  <title>LED Control</title>\n\
  <style>\n\
    body { font-size: 30px; }\n\
    input[type=range] {\n\
      width: 400px;\n\
      height: 60px;\n\
    }\n\
  </style>\n\
</head>\n\
<body>\n\
  <h1>LED Control</h1>\n\
  <p>Red <input type='range' min='0' max='255' onchange=\"fetch('/set/red?value=' + value)\"></p>\n\
  <p>Green <input type='range' min='0' max='255' onchange=\"fetch('/set/green?value=' + value)\"></p>\n\
  <p>Blue <input type='range' min='0' max='255' onchange=\"fetch('/set/blue?value=' + value)\"></p>\n\
</body>\n\
</html>";
```

```
void setup() {
  // 將 RGB LED 的腳位設為 OUTPUT
  pinMode(ledR, OUTPUT);
  pinMode(ledG, OUTPUT);
  pinMode(ledB, OUTPUT);

  // 啟動串行通訊並設定速率為 115200
  Serial.begin(115200);

  // 建立一個軟接入點 (soft-AP) 供其他裝置連線
  WiFi.softAP(ssid, password);
  IPAddress myIP = WiFi.softAPIP();
  Serial.print("AP IP address: ");
  Serial.println(myIP);

  // 當訪問根目錄 ("/") 時，返回 HTML 內容
  server.on("/", []() {
    server.send(200, "text/html", html);
  });

  // 當訪問 "/set/red"、"/set/green"、"/set/blue" 時，讀取顏色強度的值並設定對應的 LED 顏色強度
  server.on("/set/red", []() {
    int value = server.arg("value").toInt();
    analogWrite(ledR, value);
    server.send(200, "text/plain", "Red intensity set to " + String(value));
  });

  server.on("/set/green", []() {
    int value = server.arg("value").toInt();
    analogWrite(ledG, value);
    server.send(200, "text/plain", "Green intensity set to " + String(value));
  });

  server.on("/set/blue", []() {
    int value = server.arg("value").toInt();
    analogWrite(ledB, value);
    server.send(200, "text/plain", "Blue intensity set to " + String(value));
  });

  // 啟動伺服器
  server.begin();
}

void loop() {
  // 持續處理來自客戶端的請求
  server.handleClient();
}
```

五、網頁控制原理與美化

📖 網頁控制原理解說

現在，我們已經知道如何撰寫HTML網頁並利用HTTP協定控制ESP8266。

讓我們完整的看一遍原理：

1. ESP8266設定為一個Web伺服器，並在特定的路由（例如/led/on和/led/off）上監聽HTTP GET請求。
2. HTML網頁有兩個按鈕，當你點擊其中一個按鈕時，會觸發瀏覽器發送一個HTTP GET請求到對應的路由（也就是對應的URL）。這是由JavaScript的fetch()函數實現的，這個函數用來執行網路請求。
3. 當ESP8266接收到來自HTML網頁的HTTP GET請求後，會執行對應路由的處理函數，例如開啟或關閉LED燈。

這個程式碼的基本流程就是：HTML網頁（作為客戶端）發送HTTP請求 -> ESP8266（作為伺服器）接收請求並執行對應的動作。這一切都是透過HTTP協定來實現的。

🌟 網頁美化 CSS程式碼介紹

你可以使用Codepen來進行網頁美化練習，你可以使用以下程式碼：

```
<html>
<head>
  <title>LED Control</title>
  <style>
    body { font-size: 30px; }
    input[type=range] {
      width: 400px;
      height: 60px;
    }
  </style>
</head>
<body>
  <h1>LED Control</h1>
  <p>Red <input type='range' min='0' max='255' onchange=\"fetch('/set/red?value=' + value)\"></p>
  <p>Green <input type='range' min='0' max='255' onchange=\"fetch('/set/green?value=' + value)\"></p>
  <p>Blue <input type='range' min='0' max='255' onchange=\"fetch('/set/blue?value=' + value)\"></p>
</body>
</html>
```

輸入CodePen後會看到：

LED Control



將以下內容加入原本的程式碼，看看會發生什麼事

```
body {  
  font-family: system-  
ui; background: #f06d06;  
  color: white;  
  text-align: center;  
}
```

font-family = 字型
background = 背景顏色
color = 字體顏色

text-align = 文字的位置(預設是在左上角，這邊設定為正中央)

5.1 更多網頁美化技巧

HINT → 如何更換網頁的顏色?

你需要Canva色輪 <https://www.canva.com/colors/color-wheel/>
並修改以下程式碼中"#"的部分
background: #f06d06;

HINT → 如何更換字型?

常見字型有以下幾種：

serif

sans-serif

cursive

fantasy

monospace

MingLiU (細明體)

PMingLiU (新細明體)

DFKai-sb (標楷體)

Microsoft JhengHei (微軟正黑體)

若想修改字型，修改下面這行即可。

```
font-family: DFKai-sb;
```

補充說明

在CSS中，body和input是所謂的選擇器（Selectors）。

選擇器用於選擇你想要樣式化的HTML元素。

body和input選擇器分別選擇HTML文檔中的<body>和<input>元素。

body：這個選擇器選擇的是HTML文檔的<body>元素。

在一個標準的HTML頁面上，<body>元素包含了所有你可以看到的內容，包括頭部（headings）、段落（paragraphs）、圖像（images）等。

當你對<body>應用樣式時，該樣式將應用於<body>元素及其所有子元素，除非子元素有自己的樣式規則。

input：這個選擇器選擇的是所有的HTML<input>元素。

<input>元素常用於HTML表單，它可以創建不同類型的輸入欄位，如文字欄位、核取方塊、單選按鈕、提交按鈕等。

當你對<input>應用樣式時，該樣式將應用於所有的<input>元素。

空氣品質資訊站設計

簡易空氣品質資訊站是為了快速、方便地提供本地空氣品質數據而設計的裝置。其設計重點是結合高效的空氣監測感測器和直觀的用戶介面，使居民和遊客能夠瞭解當前的空氣狀況。

一、平台的選擇與設定

既然我們有LED燈和ESP8266,當然要善用其物聯網特性，將網路上的現成資訊抓下來供我們使用，以下是空氣品質資訊站的範例：

👉 你會需要申請OpenWeather的API

(網址：<https://openweathermap.org/>)

👉 空汙API使用說明

(網址：<https://openweathermap.org/api/air-pollution>)

二、程式範例

```
#include <ESP8266WiFi.h> // 引入ESP8266WiFi庫，可以讓ESP8266連接到WiFi
#include <ESP8266HTTPClient.h> // 引入ESP8266HTTPClient庫，可以讓ESP8266透過HTTP協定來取得網路資料
#include <ArduinoJson.h> // 引入ArduinoJson庫，用於處理JSON數據

// 你的WiFi信息
const char* ssid = "要連上的WiFi名稱"; // WiFi名稱
const char* password = "要連上wifi的密碼"; // WiFi密碼

// 你的OpenWeatherMap API Key
String apiKey = "你的API Key";

// LED針腳
int ledR = D7; // 紅色LED連接的腳位
int ledG = D6; // 綠色LED連接的腳位
int ledB = D5; // 藍色LED連接的腳位
// 你的OpenWeatherMap API URL
// 地區資訊使用座標做查詢，預設為新北市，座標lat=25.01&lon=121.47
String serverPath = "http://api.openweathermap.org/data/2.5/air_pollution?lat=25.01&lon=121.47&appid=" +
apiKey;
void setup() {
  pinMode(ledR, OUTPUT); // 將紅色LED的針腳設定為輸出模式
  pinMode(ledG, OUTPUT); // 將綠色LED的針腳設定為輸出模式
  pinMode(ledB, OUTPUT); // 將藍色LED的針腳設定為輸出模式

  Serial.begin(115200); // 開始序列通訊，設定傳輸速率為115200
  WiFi.begin(ssid, password); // 連接到WiFi
```

空氣品質資訊站設計

```
    } else {  
      // 等級5 (Very Poor): 紅色  
      digitalWrite(ledR, HIGH); // 打開紅色LED  
      digitalWrite(ledG, LOW); // 關閉綠色LED  
      digitalWrite(ledB, LOW); // 關閉藍色LED  
    }  
  }  
  WiFi.disconnect(); // 斷開與WiFi的連接  
  Serial.println("wifi disconnect");  
  
  http.end(); // 結束HTTP連接  
}  
  
delay(60000); // 每分鐘更新一次空氣質量指數  
}
```

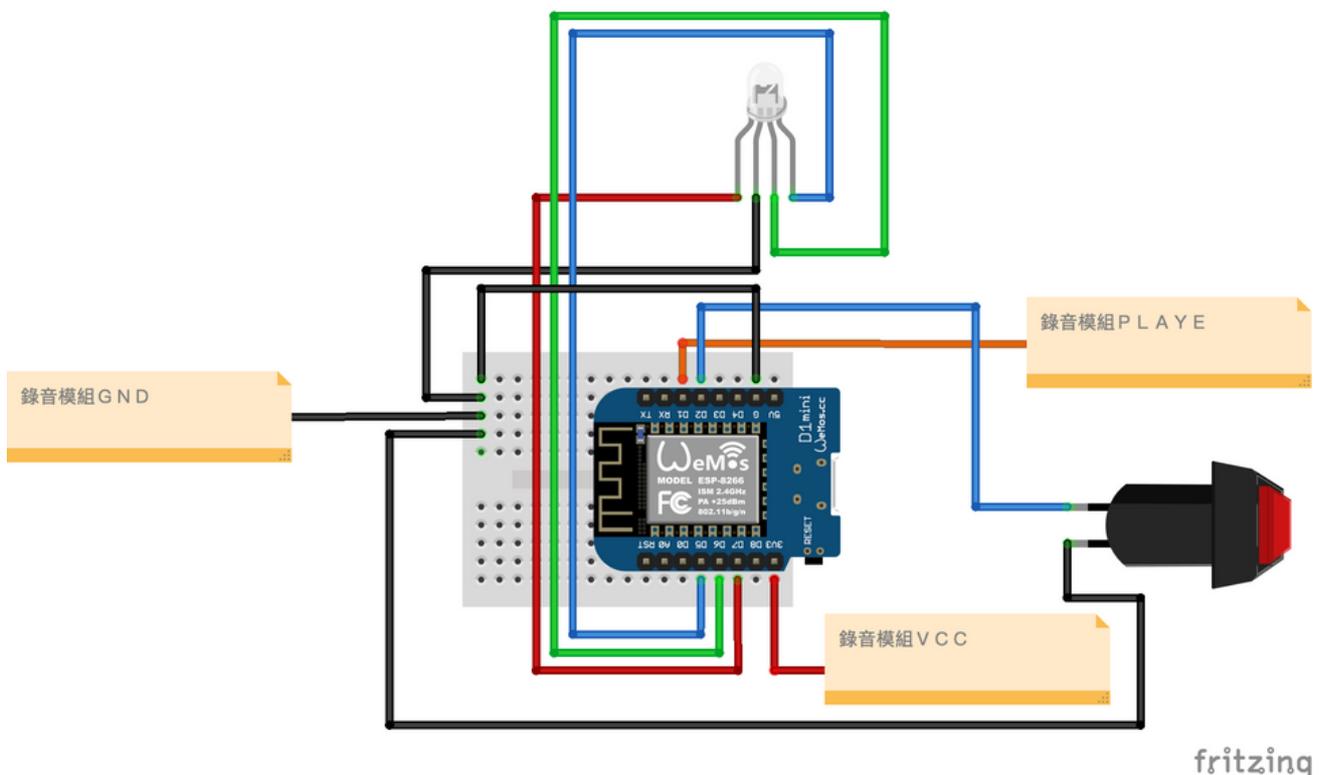
硬體互動與功能切換

硬體互動與功能切換是智慧裝置設計中的核心元素，它允許使用者直接透過物理介面（如按鈕、旋鈕或觸摸屏）與裝置進行互動。

功能切換特別重要，因為現代的裝置越來越複雜，且具有多種操作模式。透過明確的功能切換設計，用戶可以輕鬆選擇和轉換不同的操作模式，例如從音量控制切換到頻道選擇，或從一種燈光模式切換到另一種。

硬體元件的選擇和佈局、以及其與軟體之間的互動，都是設計過程中必須仔細考慮的要點。目的是確保無論是新手還是熟練用戶，都能夠快速、直覺地使用該裝置，而無需過多的學習或指南。

一、路音模組的控制



硬體互動與功能切換

```
int ledR = D7;
int ledG = D6;
int ledB = D5;
int rec = D2;
int ply = D1;
int recState;
int plyState;

void setup() {
  // put your setup code here, to run once:
  pinMode(ledR, OUTPUT);
  pinMode(ledG, OUTPUT);
  pinMode(ledB, OUTPUT);
  pinMode(rec, OUTPUT);
  pinMode(ply, OUTPUT);
  digitalWrite(rec, LOW);
  digitalWrite(ply, LOW);
}

void loop() {
  // put your main code here, to run repeatedly:
  /*
   * 在這個loop中·錄音開啟後的delay必須比放音後的delay短
   * 因為放音後錄音模組會有一個重置動作·如果不等他重置下一輪就會錄不到
   音 */
  digitalWrite(rec, HIGH); //錄音開啟
  digitalWrite(ledR, HIGH);
  digitalWrite(ledG, LOW);
  delay(3000);
  digitalWrite(rec, LOW); //錄音停止
  digitalWrite(ledR, LOW);
  delay(2000);
  digitalWrite(ply, HIGH); //放音按鈕按下
  delay(100);
  digitalWrite(ply, LOW); //放音按鈕放開
  digitalWrite(ledG, HIGH);
  delay(4900); //delay必須比錄音時間長
}
```

二、按鈕電路與程式

若要測試，請參考P.19的接線圖並將以下程式灌入ESP8266：

```
const int buttonPin = D3; // 按鈕連接到的腳位
const int ledRed = D7; // 紅色LED連接到的腳位
const int ledGreen = D6; // 綠色LED連接到的腳位
const int ledBlue = D5; // 藍色LED連接到的腳位

void setup() {
  pinMode(buttonPin, INPUT_PULLUP); // 設定按鈕為輸入，內部啟用上拉電阻
  pinMode(ledRed, OUTPUT); // 設定LED為輸出模式
  pinMode(ledGreen, OUTPUT);
  pinMode(ledBlue, OUTPUT);

  // 初始時所有LED燈關閉
  digitalWrite(ledRed, LOW);
  digitalWrite(ledGreen, LOW);
  digitalWrite(ledBlue, LOW);
}

void loop() {
  if (digitalRead(buttonPin) == LOW) { // 檢查按鈕是否被按下（接地）
    blinkLed(ledRed, 1); // 閃爍紅色LED
    delay(500); // 等待0.5秒
    blinkLed(ledGreen, 1); // 閃爍綠色LED
    delay(500); // 等待0.5秒
    blinkLed(ledBlue, 1); // 閃爍藍色LED
    delay(1000); // 等待1秒以便觀察
  }
}

void blinkLed(int led, int times) {
  for (int i = 0; i < times; i++) {
    digitalWrite(led, HIGH);
    delay(500);
    digitalWrite(led, LOW);
    delay(500);
  }
}
```

三、多模式程式架構

多模式程式架構是一種軟體設計策略，旨在使單一的應用程式或系統具有多種操作模式或功能。這樣的架構允許應用程式在不同的情境或需求下適應並改變其行為。

這種架構通常透過以下方式實現：

- 模組化設計：應用程式被劃分為多個獨立的模組或部件，每一個都有其專屬功能。
- 狀態管理：程式內部維護一個或多個狀態，根據當前狀態決定如何響應用戶的輸入或其他事件。
- 互動介面：用戶可以透過一系列的介面元件（例如按鈕、選單或滑桿）來選擇不同的模式或調整參數。

利用多模式程式架構，開發者可以為用戶提供更多的靈活性和定制性。這尤其適用於那些需要在不同情境下操作或具有多重功能的系統。

篇幅關係，請上線上講義觀看主程式

(網址 https://hackmd.io/@allenyang30/ByXZ_-rWT)

進入講義後，在左側書籤列找到：★多模式切換主程式

切換模式的方法：

關閉電源 ➡ 按著紅色按鈕不放 ➡ 打開電源 ➡ 看到藍燈後放開紅色按鈕

➡ 打開手機Wifi搜尋fangGu_LED並連線 ➡ 連線成功後打開瀏覽器

➡ 輸入192.168.4.1 ➡ 若要使用氣象站模式需填入表格中的資訊後送出。

外觀設計與組裝

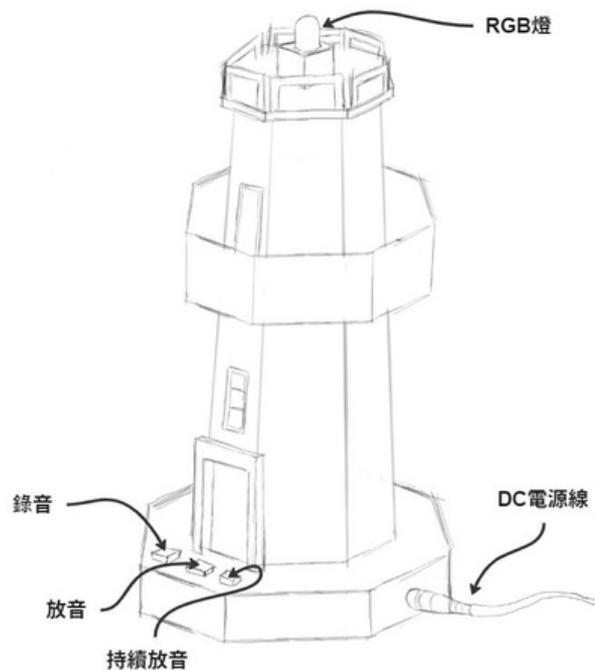
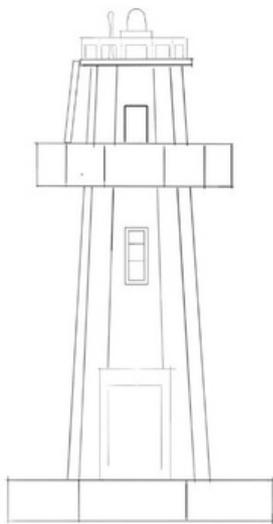
本單元將介紹筆者的設計流程與工具，本次設計將燈塔造型與物聯網功能做結合，因此需考量造型與硬體的配合，下面將簡單分享草圖、建模工具以及參考書籍。

一、外觀設計流程

草圖

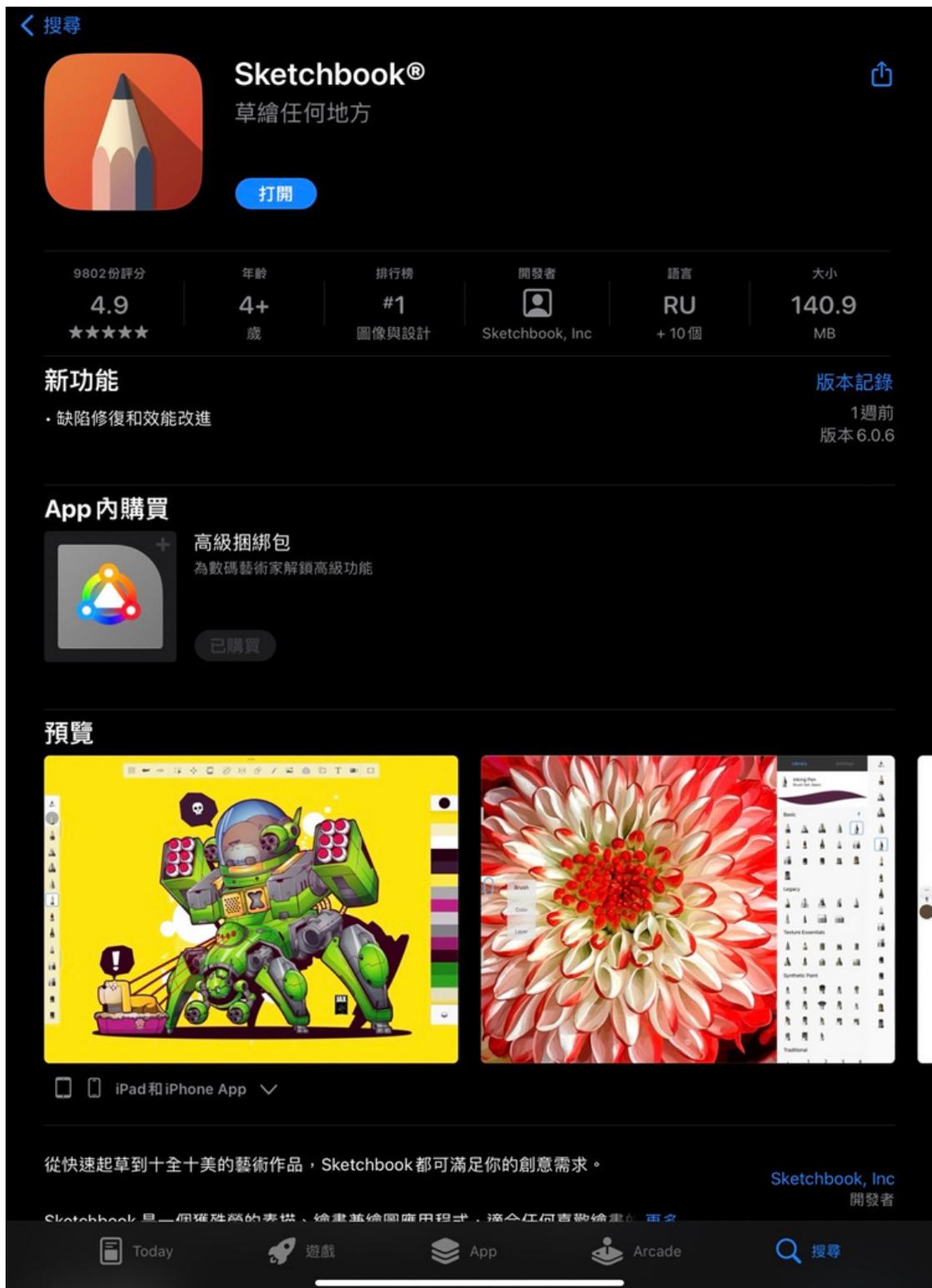
草圖與說明

- 可WiFi操控燈色與效果
- 可錄音留言
- 基隆島燈塔造型



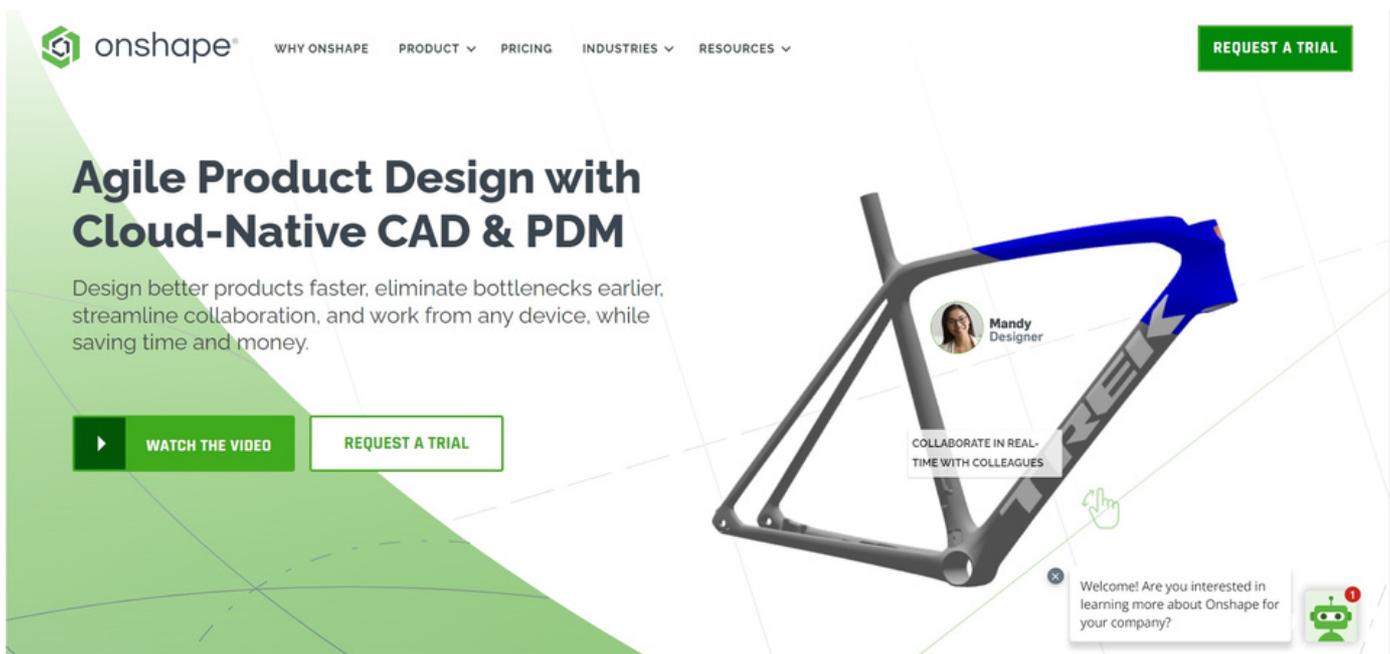
二、外觀設計工具

草圖：Adobe Sketchbook (於iPad上使用)



建模：Onshape (使用電腦瀏覽器操作)

請參考電子講義中的申請流程



推薦書籍-設計素描：基礎原理



博客來 > 中文書 > 藝術設計 > 繪畫 > 繪畫方法 > 素描技法 > 商品介紹



設計素描：基礎原理

作者：庫斯·埃森·羅薩琳·史都爾 [追蹤作者](#) ?
 出版社：龍溪 [訂閱出版社新書快訊](#) ?
 出版日期：2012/08/09
 語言：繁體中文
 定價：800元
 優惠價：95折 760元
 本商品單次購買10本85折 680元
 運送方式：[臺灣與離島](#) [海外](#)
 🚚 可配送點：台灣、蘭嶼、綠島、澎湖、金門、馬祖
 📍 可取貨點：台灣、蘭嶼、綠島、澎湖、金門、馬祖
 📦 台北、新北、基隆宅配快速到貨(除外地區)

中午前訂 可明天 拿

庫存=1

放入購物車

直接結帳

加入下次再買清單

我要評鑑

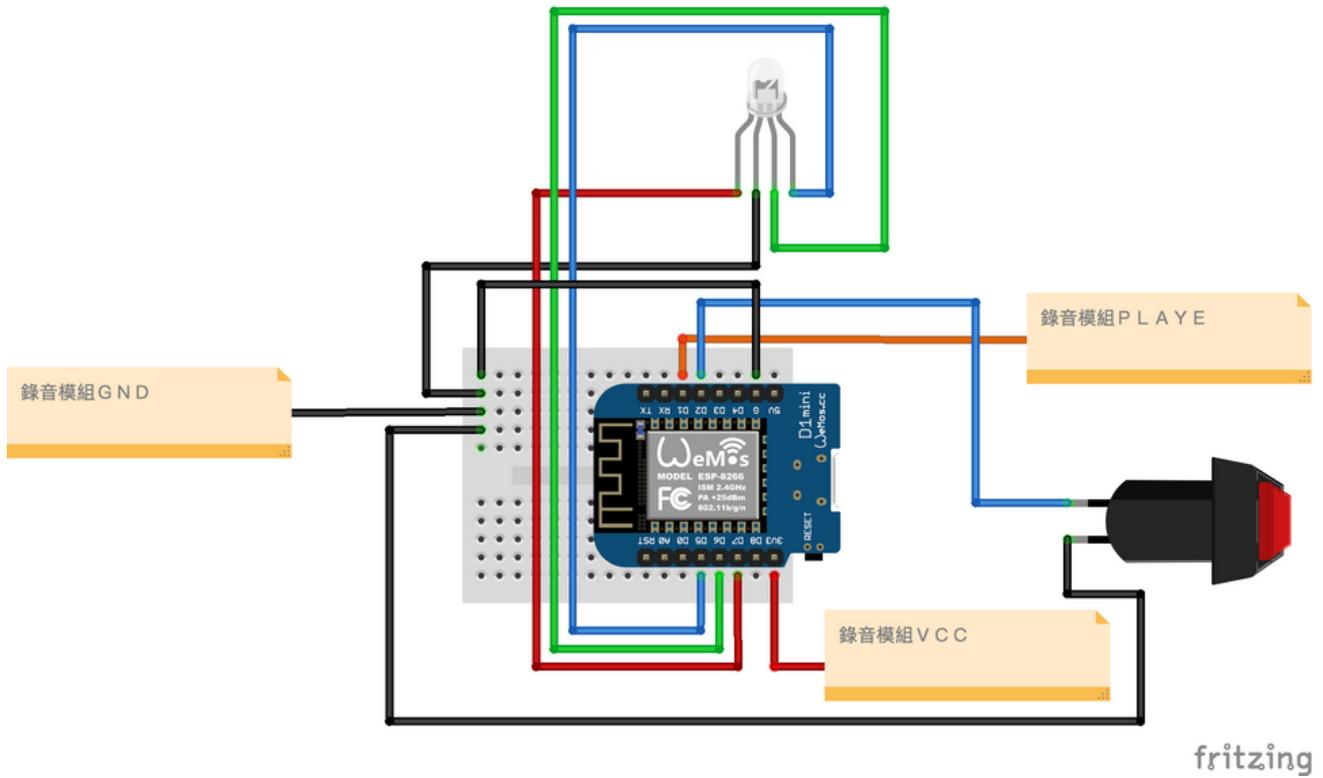
分享

讚 12



三、組裝說明

請參考此接線圖與組裝影片



電子講義👉



組裝影片👉

筆記欄：



INSPIRE

2023.10.27

13

LECTURE FOR
DESIGN & CODING

WITH SPEAKER

ALLEN YANG



WWW.FANGGU.ORG