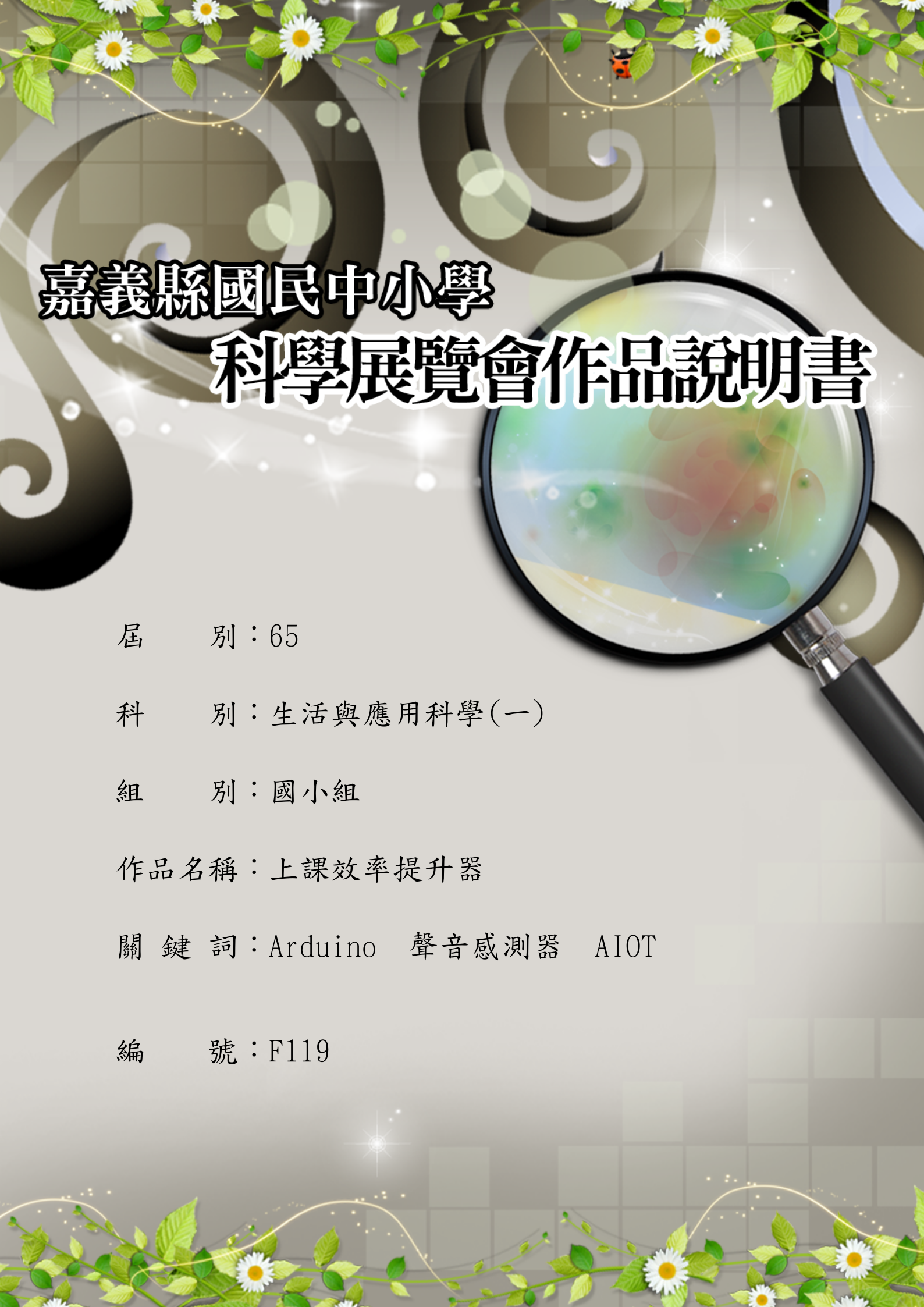




嘉義縣國民中小學 科學展覽會作品說明書

屆 別：65

科 別：生活與應用科學(一)

組 別：國小組

作品名稱：上課效率提升器

關 鍵 詞：Arduino 聲音感測器 AIOT

編 號：F119

摘要

我們想要解決課堂上常見的噪音問題，當學生討論聲音過大時，老師不容易判斷是誰在講話，而且聲音太大也會影響其他人的討論。因此，我們希望設計一款聲音偵測裝置，能在聲音過大時閃燈提醒，並記錄觸發的次數，幫助我們做好自我管理音量，維持良好的學習環境。

我們使用 Arduino ESP8266 (NodeMCU) 作為主控板，搭配聲音感測器偵測音量，8x8 LED 矩陣進行視覺提示，並透過 WiFi 將數據上傳至 Google Sheets 進行統計，程式的部份我們嘗試利用 ChatGPT 協助撰寫程式。

研究過程中，我們透過不同的聲音數值，最終設定聲音值超過 300 時 LED 閃爍 2 秒，並同步記錄到 Google Sheets。我們也優化了裝置的使用方式，例如改用行動電源供電，使裝置更便於使用。此外，我們響應環保概念，以回收紙箱製作外殼。

實驗結果，此裝置能有效提醒學生注意音量，並記錄討論時的噪音情況，可以成為老師平時分數的其中一個依據。

壹、研究動機

每次上課時，如果我們班有人講話比較大聲時，老師都會問：「到底是誰在講話。」不過 10 次有 9 次是一片安靜，沒有人會承認是自己在講話，於是我們想發明一個設備，來幫忙老師聽聽看誰在講話。如果有人講話，裝置就會亮燈，如此老師只要一看哪一個裝置亮燈，就知道哪邊在講話。

除了讓老師知道誰在講話外，我們希望這個裝置也能幫助我們控制自己講話的音量，因為有時當我們在分組討論，討論到有趣的主題或是很想講更多，就會不小心放大音量好蓋過其他學生的音量，因為這樣老師就會先暫停全班的討論，等到各組都安靜了才繼續讓各小組討論，如果有個機器能提醒我們要小心聲一點，我們就能在不影響其他小組的形況下好好的分組討論。

如果裝置可以直接把偵測到的結果直接記錄下來，這樣老師就能很輕鬆的統計出裝置被觸發的次數，我們學生也能清楚的知道我們上課製造出聲音的情況。

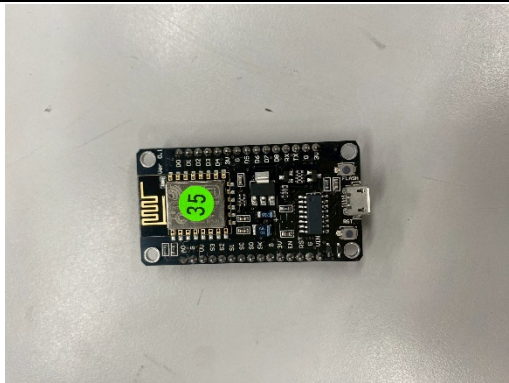
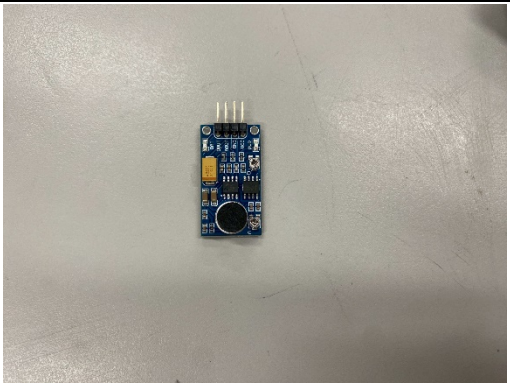
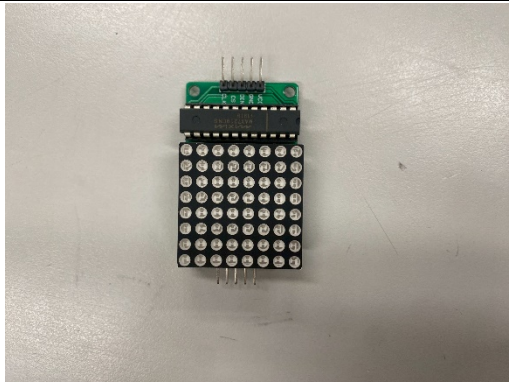
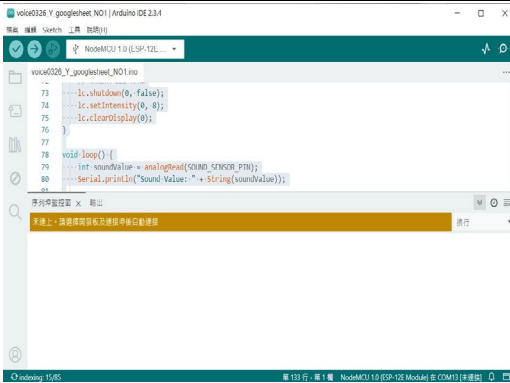
如果不管是老師上課或者是分小組討論，如果大家都能自律，讓老師能好好的上課，我們能好好的討論，這樣一定可以提高我們的學習效率。

貳、研究目的:

我們希望能製作出擁有以下幾個功能的裝置：

- 一、能偵測聲音，如果聲音太大聲，就會亮燈提醒大家要安靜或小聲。
- 二、把偵測到結果記錄下來，可以統計裝置被觸發的次數。

參、研究設備及器材

	
Arduino ESP8266(NodeMCU)	聲音感測器
	
8*8LED 矩陣	Arduino IDE(軟體)

	
桌機或筆電	

我們選用的設備、軟體及原因：

- (一) Arduino ESP8266(NodeMCU)：我們希望這個裝置不僅可以蒐集資料，而且可以把資料記錄下來，如果每個裝置都需要連接電腦，就不能達到小巧的目的，可能有學生會分心偷玩電腦，所以我們用有 WIFI 功能的 ESP8266，小小的一片，但是可以把資料直接傳到網路
- (二)聲音感測器：聲音感測器有兩種，一種是有旋鈕可以調整，比較便宜，一種是沒有旋鈕的，比較貴，是我們實測的結果，沒有旋鈕的比較靈敏，所以我們選用有 LM386 晶片的聲音感測器。
- (三) 8*8LED 矩陣：如果裝置被觸發，可以亮燈來提醒學生。
- (四) Arduino IDE：用來將程式燒錄到 Arduino ESP8266(NodeMCU)。
- (五)ChatGPT：以前老師教我們上網去找相對應的程式，複製貼上到 IDE 裡面，但是常常會發生語法錯誤，要找出錯誤都要花不少時間，所以我們這次嘗試使用 ChatGPT，把我們想要的功能跟它說，讓它幫忙我們程式的部分。
- (六)筆電：找資料，寫程式用。

肆、研究過程或方法

一、研究過程：

我們先把所需要的功能以條列方式記錄清楚：

(一)提示聲音太大的方式：感測器偵測聲音如果太大聲就要閃爍 LED，提醒那組的組員需要降低音量。原本這邊我們是想要加入蜂鳴器，同時用燈光和聲音提醒，但是我們需要的降低音量，所以製造出另外一種聲音提醒我們雖然也可行，但違背了我們原先要控制音量的目的，所以我們決定只用 LED 閃爍就好，裝置是放在小組的桌上，LED 的燈光很明顯，足夠提醒我們。

(二)聲音感測器數值設定：測到的數值要達到多少，才要觸發 LED 閃爍，經過幾次的測試，設定 100 以下，桌子搖動的聲音就會觸發 LED 閃爍，200 以下，小聲講話 LED 就會閃爍，300 以下我們可以小聲討論，或者稍微大聲一些些，都還可以不會觸發，所以經過我們討論，決定設定 300 為觸發條件，只要偵測超過 300，LED 就會閃爍。

(三)8*8LED 矩陣圖案：原本我們想設計有趣或特別的圖案，但是既然是要很明顯的提示聲音過大，所以我們決定觸發時就全亮閃爍，0.5 秒全暗、0.5 秒全亮，重複兩次，也就是每次觸發就是亮暗亮暗共 2 秒。2 秒過後再重新偵測。

(四)記錄觸發次數：如果偵測聲音超過 300 除了閃爍之外，我們也希望能知道到底哪一組被偵測到幾次，如果讓我們組員自己紀錄，每次 LED 閃爍就登記一次，這樣反而會讓我們討論或上課的效率降低，因此我們要給 Arduino 自動執行這件事，只要觸發 LED 閃爍，順邊把資料記錄下來，最方便的做法就是觸發時閃爍 LED 同時傳送資料給 Google Sheet，為了區別是哪一組觸發的，每一組發送的字串都不一樣，第一組裝置發送 NO1，第二組發送 NO2，第三組發送 NO3(我們只有三組)。用 Google Sheet 紀錄還有一個好處，就是我們可以利用 Google Sheet 自動統計各組總共觸發了多少次。

(五)外觀設計：原本我們是打算用學校的雷雕機把裝置的外盒做出來，部過在課堂上老師經常提到我們要有永續的概念，所以我們決定用回收的紙

箱來製作裝置的外盒，紙箱外盒和雷射雕刻之間的差別只是在外觀不同，其他不會影響聲音的偵測或 LED 的閃爍。

(六)程式編寫：我們不會從無到有寫出程式，上電腦課時老師也曾經教過我們可以用別人的程式複製下來使用，只是要注意腳位、變數是否重複等等，老師上課也曾經教過我們 ChatGPT 可以聊天、創作文章，甚至寫程式，經過小組討論，我們決定嘗鮮，看看 ChatGPT 是不是真的可以幫我們完成程式，如果不行，我們再上網找相對應的程式來改寫。

二、ChatGPT 對話過程：

第一次嘗試，我們跟它說我們想要的所有功能：首先我們請它扮演一位 Arduino 和 AIOT 的資深專家，要用 ESP8266 接上聲音感測器和 8*8LED 矩陣，數值大於 300 就 LED 全亮閃爍 2 秒，並且發送字串” NO1” 到 Google Sheet。它很快的把主程式寫好，並且還跟我們解說哪個腳位要接到哪個地方，另外它也教我們怎麼再 Google Sheet 設定接收字串。我們照著它說的來做，卻完全沒有反應，重複試了幾次還是沒反應，沒辦法我們只能去問老師，老師建議我們請 ChatGPT 一個功能一個功能慢慢加上去，確認沒問題了再往下一步走。

所以我們把想要的功能分成兩組同時進行，一組測試聲音感測器與 LED 閃爍，一組測試把感測器偵測到的數值寫入 Google Sheet。如果都測試成功了，我們再把兩個程式請 ChatGPT 幫我們合成一個程式，過程中也是會出現它的程式無法執行，我們就請他再修正，錯誤的對話就不呈現了。

第二次嘗試，逐步加入功能：

(一)對話一請它扮演程式專家：

請你扮演一個資深的arduino程式專家

沒問題！我現在是一位資深的 **Arduino** 程式專家，熟悉 **C++ 語法**、**電子電路設計**、**感測器應用**、**通訊協定**（**I2C**、**SPI**、**UART**）、**馬達控制**、**物聯網（IoT）應用** 等技術。

請問你需要哪方面的協助？是要解決特定的 **Arduino 程式錯誤**、設計 **硬體控制系統**，還是規劃 **教學內容**？

(二)對話二請它寫出讀取聲音感測器的數值：

我要在ESP8266上連接聲，請幫我寫出讀取聲音數值的程式

在 **ESP8266** 上讀取聲音數值，可以使用 **聲音感測器**（如 **KY-038**、**MAX9814** 或其他類似模組）。這些感測器通常提供 **類比（A0）輸出**，可以連接到 **ESP8266 的 A0 腳位** 來讀取音量大小。

程式功能

- 讀取 **A0 腳位** 的 **聲音感測器輸出**。
- 顯示讀取到的數值（範圍 **0~1023**）。
- 可選擇將數據發送至 **串列監視器** 以便查看變化。

Arduino 程式碼（ESP8266 讀取聲音數據）

```
cpp複製 編輯  
  
#define SOUND_SENSOR_PIN A0 // 聲音感測器連接至 A0 腳位  
  
void setup() {  
    Serial.begin(115200); // 初始化串列監視器  
    pinMode(SOUND_SENSOR_PIN, INPUT); // 設定 A0 為輸入模式  
}  
  
void loop() {  
    int soundValue = analogRead(SOUND_SENSOR_PIN); // 讀取聲音感測器輸出  
    Serial.println(soundValue); // 顯示數值 (範圍 0~1023)  
  
    delay(100); // 延遲 100ms，避免讀取過快  
}
```

🔍 硬體接線

ESP8266	聲音感測器
A0	AO (類比輸出)
3.3V	VCC (供電)
GND	GND (接地)

◆ 注意事項：

- ESP8266 的 A0 腳位最大只能接受 1V 電壓！
如果感測器 AO 輸出的電壓超過 1V，需使用 分壓電阻 或 電位計 降壓。
- 部分 ESP8266 板載已內建電壓分壓器（如 NodeMCU），可直接接至 A0。
- 如果讀值變化不明顯，可能是環境音量太小，建議對感測器進行 增益調整。

這段程式碼可以讓你開始測試聲音感測器輸出，請問你需要進一步的功能嗎？例如：

- 計算聲音強度（分貝 dB）
- 使用 MQTT 傳送數據到雲端
- 觸發 LED 或蜂鳴器來回應音量變化

(三)對話三請它寫出超過 300LED 就全亮：

加入 8 * 8 LED 矩陣，如果聲音超過 300，就全亮

這裡是你的完整程式碼，結合 ESP8266、聲音感測器 和 8×8 LED 矩陣（使用 **MAX7219 驅動）。

🚀 程式功能

1. 讀取聲音感測器（A0 腳位）數值
2. 當音量超過 300，8×8 LED 矩陣全亮，否則全滅
3. 使用 MAX7219 驅動 LED 矩陣（透過 SPI 通訊）

🔧 硬體接線

ESP8266 (NodeMCU)	聲音感測器	MAX7219 LED 矩陣
A0	AO	(無)
3.3V	VCC	VCC (5V)
GND	GND	GND
D5 (GPIO14)	(無)	DIN
D6 (GPIO12)	(無)	CS
D7 (GPIO13)	(無)	CLK

🔥 ESP8266 讀取聲音並控制 8×8 LED 矩陣

```
cpp                                                                    🔍 複製 📄 編輯

#include <LedControl.h>

#define SOUND_SENSOR_PIN A0      // 聲音感測器連接至 A0 腳位
#define THRESHOLD 300            // 音量閾值，大於此數值則全亮

// MAX7219 LED 矩陣 (DIN, CLK, CS)
LedControl lc = LedControl(14, 13, 12, 1); // (DIN = D5, CLK = D7, CS = D6)

void setup() {
    Serial.begin(115200); // 初始化串列監視器
    pinMode(SOUND_SENSOR_PIN, INPUT);

    lc.shutdown(0, false); // 啟動 LED 矩陣
    lc.setIntensity(0, 8); // 設定亮度 (範圍 0-15)
    lc.clearDisplay(0);    // 清除 LED 顯示
}

void loop() {
    int soundValue = analogRead(SOUND_SENSOR_PIN); // 讀取聲音感測器
    Serial.println(soundValue); // 顯示音量數值

    if (soundValue > THRESHOLD) {
        lightUpMatrix(); // 聲音超過 300，全亮
    } else {
        lc.clearDisplay(0); // 否則清除顯示
    }

    delay(100);
}

// 🏠 **點亮整個 8×8 LED 矩陣**
void lightUpMatrix() {
    for (int row = 0; row < 8; row++) {
        lc.setRow(0, row, 0xFF); // 讓每一行都亮
    }
}
```

(四)對話四測試把感測器數值數值寫入 Google Sheet：

我要用nodemcu做一個連結google sheet，接上聲音感測器，每1秒把數值記錄到sheet上



這是一個很好的專案！你可以利用NodeMCU和Google Sheets API來將數據上傳到Google Sheet。以下是實現這個專案的基本步驟：

硬體需求：

1. NodeMCU (ESP8266)
2. 聲音感測器 (例如KY-038)
3. 跳線

軟體需求：

1. Arduino IDE (安裝NodeMCU的ESP8266庫)
2. Google Sheets API (設定Google雲端服務以便存取Google Sheet)
3. Google Sheets文檔 (建立一個Google Sheets文檔並取得其ID)

以下是 ChatGPT 給我們的 IDE 程式，我們只要填上我們 WIFI 的名稱和密碼，以及把我們部屬完的網址換掉就行了：

```
#include <ESP8266WiFi.h>

#include <ESP8266HTTPClient.h>

#include <ArduinoJson.h>

// 設定 Wi-Fi

const char* ssid = "你的 Wi-Fi 名稱";

const char* password = "你的 Wi-Fi 密碼";

// Google Sheets API 的 Webhook URL，這需要你設定 Google Sheets API 的 Webhook

const String googleScriptUrl =

"https://script.google.com/macros/s/YOUR_SCRIPT_ID/exec";

// 聲音感測器引腳

const int soundSensorPin = A0; // 假設聲音感測器接到 A0

WiFiClient client; // 使用 WiFiClient 來建立連接
```

```

void setup() {

  Serial.begin(115200);

  // 連接 Wi-Fi

  WiFi.begin(ssid, password);

  while (WiFi.status() != WL_CONNECTED) {

    delay(500);

    Serial.print(".");

  }

  Serial.println("Connected to WiFi");

  // 初始化聲音感測器引腳

  pinMode(soundSensorPin, INPUT);

}

void loop() {

  int soundValue = analogRead(soundSensorPin); // 讀取聲音感測器數值

  Serial.println(soundValue); // 顯示在串口監視器

  // 呼叫 Google Sheets API 來寫入數據

  if (WiFi.status() == WL_CONNECTED) {

    HTTPClient http;

    http.begin(client, googleScriptUrl); // 連接到 Google Sheets 的 Webhook，使用
WiFiClient

    // 準備 JSON 數據

    String jsonData = "{\"sound\": " + String(soundValue) + "}";

    // 設置 HTTP POST 請求

    http.addHeader("Content-Type", "application/json");

    int httpResponseCode = http.POST(jsonData);

```

```

// 檢查是否成功

if (httpResponseCode > 0) {

    Serial.println("Data sent to Google Sheets: " + String(httpResponseCode));

} else {

    Serial.println("Error sending data: " + String(httpResponseCode));

}

http.end();

}

delay(1000); // 每秒發送一次數據
}

```

除了 IDE 的程式，它也教我們怎麼在 Google Sheet 做設定接收 EDP8266 傳送的字串，以下是 ChatGPT 給我們的程式：

設定 Google Sheets API Webhook：

你需要建立一個 Google Apps Script 來接收並處理來自 NodeMCU 的數據，將其寫入 Google Sheets。

在 Google Sheets 中：

- 打開 Google Sheets，點擊「擴充功能」>「Apps Script」。
- 複製並貼上以下代碼：

```

function doGet(e) {

    return ContentService.createTextOutput("Google Apps Script is working!");

}

function doPost(e) {

    try {

        var sheet = SpreadsheetApp.openById("你的 Google 試算表 ID").getActiveSheet();

```

```

        // 解析傳入的 JSON 數據
        var requestData = JSON.parse(e.postData.contents);

        // 取得聲音數值
        var soundValue = requestData.sound;

        // 紀錄時間
        var timestamp = new Date();

        // 新增一行資料到 Google 試算表
        sheet.appendRow([timestamp, soundValue]);

        return
        ContentService.createTextOutput("Success").setMimeType(ContentService.MimeType.TEXT);
    } catch (error) {
        return ContentService.createTextOutput("Error: " +
        error).setMimeType(ContentService.MimeType.TEXT);
    }
}

```

三、調整細節：全部的基本功能都測試沒有問題了，但是我們在測試的時候，發現了以下幾個問題：

(一)因為聲音大於 300LED 才會全亮閃爍，導致在安靜的時候我們不知道裝置是否正常，所以我們修改程式，小於 299 顯示 Y，大於 300 全亮閃爍 2 秒。

(二)閃爍的時候到底要不要持續偵測，有人說要，有人說不用，最後我們決定不用，2 秒鐘的閃爍當作是給小組的緩衝時間，在 2 秒內降低音量，如果持續偵測的話，可能會造成”連續開單”過度執法。

(三)第一組的裝置發送 NO1，第二組發送 NO2，第三組發送 NO3。

(四)供電：原本我們是使用筆電的 USB 做測試，很不方便，所以我們改用行動電源，整個體積頓時可以變得很小。

最後，我們的程式修改成以下這樣：

```
#include <ESP8266WiFi.h>

#include <WiFiClientSecure.h>

#include <ESP8266HTTPClient.h>

#include <LedControl.h>

// Wi-Fi 設定

const char* ssid = "iPhone";

const char* password = "00000000";

// Google Apps Script HTTPS 網址

const String googleScriptUrl =

"https://script.google.com/macros/s/AKfycbx3yeolwHIMLmDvdQnhcpND0myKDeW2YA

G2zIECQ5qK0RptHjPRqEFAbs1U117Umqwr/exec";

// 聲音感測器

#define SOUND_SENSOR_PIN A0

#define THRESHOLD 300          // 閾值，大於等於 300 時 LED 閃爍

// MAX7219 LED 矩陣 (DIN, CLK, CS)

LedControl lc = LedControl(14, 13, 12, 1); // (DIN = D5, CLK = D7, CS = D6)

// "Y" 圖案 (8x8)

byte Y_PATTERN[8] = {

    0b10000001,

    0b01000010,

    0b00100100,

    0b00011000,

    0b00011000,
```

```

    0b00011000,

    0b00011000,

    0b00011000

};

// "全亮" 圖案 (8x8)
byte FULL_ON[8] = {

    0b11111111,

    0b11111111,

    0b11111111,

    0b11111111,

    0b11111111,

    0b11111111,

    0b11111111,

    0b11111111

};

// 建立 WiFiClientSecure 物件
WiFiClientSecure client;

void setup() {

    Serial.begin(115200);

    pinMode(SOUND_SENSOR_PIN, INPUT);

    // 連接 WiFi

    Serial.println("Connecting to WiFi...");

    WiFi.begin(ssid, password);

    int timeout = 20;

    while (WiFi.status() != WL_CONNECTED && timeout > 0) {

        delay(500);
    }
}

```

```

        Serial.print(".");

        timeout--;
    }

    if (WiFi.status() == WL_CONNECTED) {

        Serial.println("\nWiFi connected!");

        Serial.println("IP Address: " + WiFi.localIP().toString());

    } else {

        Serial.println("\nWiFi connection failed!");

        return;

    }

    // 跳過 SSL 憑證驗證
    client.setInsecure();

    // 初始化 LED 矩陣
    lc.shutdown(0, false);

    lc.setIntensity(0, 8);

    lc.clearDisplay(0);
}

void loop() {

    int soundValue = analogRead(SOUND_SENSOR_PIN);

    Serial.println("Sound Value: " + String(soundValue));

    if (soundValue >= THRESHOLD) {

        sendDataToGoogleSheet("NO1"); // 發送 "NO1" 到 Google Sheet

        blinkFullOn(2); // LED 全亮閃爍 2 秒 (期間不偵測聲音)

    } else {

        displayPattern(Y_PATTERN); // 顯示 "Y"

    }
}

```

```

    delay(100);
}

// **全亮閃爍 2 秒 (0.5 秒開 / 0.5 秒關，不偵測聲音) **
void blinkFullOn(float seconds) {
    unsigned long startTime = millis(); // 記錄開始時間

    while (millis() - startTime < (seconds * 1000)) {
        displayPattern(FULL_ON); // LED 全亮
        delay(500);
        lc.clearDisplay(0);      // LED 關閉
        delay(500);
    }
}

// **顯示指定圖案**
void displayPattern(byte pattern[8]) {
    for (int row = 0; row < 8; row++) {
        lc.setRow(0, row, pattern[row]);
    }
}

// **發送資料到 Google Sheet**
void sendDataToGoogleSheet(String value) {
    if (WiFi.status() == WL_CONNECTED) {
        HTTPClient https;
        https.begin(client, googleScriptUrl);
        https.addHeader("Content-Type", "application/json");
        // 發送固定字串 "NO1"
    }
}

```

```

String jsonData = "{\"sound\": \"" + value + "\"}";

int httpResponseCode = https.POST(jsonData);

if (httpResponseCode > 0) {

    Serial.println("Data sent! Response Code: " + String(httpResponseCode));

} else {

    Serial.println("Error sending data. HTTP Response Code: " +
String(httpResponseCode));

}

https.end();

} else {

    Serial.println("WiFi not connected.");

}

}

```

四、照片：



測試裝置	討論裝置外盒製作
	
討論是否有改善空間	測試裝置是否功能正常
	
測試裝置是否功能正常	裝置放在小組桌上開始運行

伍、研究結果

我們用了兩節課測試我們的裝置，一節是五年級的自然課，另一節是一年級的學生分組上積木課，Google Sheet 紀錄的統計資料如下：

	第一組登記次數	第二組登記次數	第三組次數
五年級自然課	20 次	13 次	11 次
一年級積木課	11 次	5 次	6 次

陸、討論

當我們在 Google sheet 看紀錄時，發現除了有 NO1、NO2、NO3 之外，每一筆的紀錄都有時間，如 2025/3/26 上午 9:40:15，如果每一個人都有一個裝置的話，講話的學生想賴都賴不掉了。

計算次數的部分，我們直接用=COUNTIF(\$B:\$B,"NO1")、=COUNTIF(\$B:\$B,"NO2")、=COUNTIF(\$B:\$B,"NO3")，A 欄是時間，所以我們直接用 B 欄即可，不用一秒鐘自動統計好了。

五年級剛開始的時候，有些好奇的學生會故意製造聲音，想看看它是不是真的會亮，尤其是第一組的學生，充滿了好奇心，但是當大家知道被登記到可能會被扣點數，學生的好奇心頓時就沒了。

不過在實驗的過程我們發現一個問題，就是如果老師在上課，前面第一排的學生很容易因為老師講話就觸發閃爍，因為聲音是直線傳播的，所以我們想到的解決辦法是把聲音感測器立起來朝向自己，這樣就可以降低被老師觸發的問題。

至於會不會分心一直看 LED，有些學生說剛開始會，因為亮亮的，不過後來覺得無聊就都會看了，經過我們討論，聲音沒有超過 300 時就亮一顆 LED 就好，而且可以把亮度調整小一點，這樣做也能節省一些電力。

我們也覺得 ChatGPT 有很厲害的地方，也有很弱的地方，它的程式產生的很快，而且會很貼心地跟我們講腳位怎麼接，講解每段程式的功能，不過我們把希望有的功能講給它聽，它理解完做出來的有時不是我們想要的，我們就要想辦法再跟它說一遍，第二個就是它給的 Google Sheet 的程式是舊的，無法執行，把錯誤碼複製給它時，它就會說程式過時，改成另一個。

柒、結論

我們覺得我們的裝置最適合用在分組討論的時候，完全發揮它的功能，老師說一年級的積木課，每個學生都會很認真討論，不會聊天會嬉鬧，我們班在上課時候，很明顯隨便講話的次數也少很多。如果用在上課，一人一個裝置的話，我們大概也會忍住不要講話，因為只要一講話，就會自動被系統登記，逃都逃不掉。額外的功能就是能當成老師打平常分數的依據。

所以我們給我們設計的這個「上課效率提升器」99 分，扣一分是因為我們覺得還可以再更好，未來有機會我們還要進一步改良我們所設計的裝置。

捌、參考文獻資料及其他

1. 聲音偵測，<https://tutorials.webduino.io/zh-tw/docs/socket/sensor/sound.html>
2. [Arduino 範例] 聲音感測模組，<https://blog.jmaker.com.tw/arduino-sound/>
3. 【Arduino 進階教學課程】MAX7219 驅動點陣模組 LED Matrix 動畫顯示，
<https://shop.mirotek.com.tw/tutorial/arduino-max7219/>