

ESP32-GYRO 教學

112年ESP32-GYRO 教學操作手冊



<https://reurl.cc/v6p2Ea>

LINE:

https://line.me/ti/g2/OyvguDdPYjxECX1ZRi17aWRay0K20iL_4cEtNA?utm_source=invitation&utm_medium=link_copy&utm_campaign=default



施朝斌

shihkw1@gmail.com

感謝:

MASON

夜市小霸王

慧手科技-sam lin

大安高工冷凍科-歐鎮寬

虎尾科技大學-林正敏

MIT Media Lab - Mitch Resnick

https://www.ted.com/talks/mitch_resnick_let_s_teach_kids_to_code?language=zh-TW

範例程式

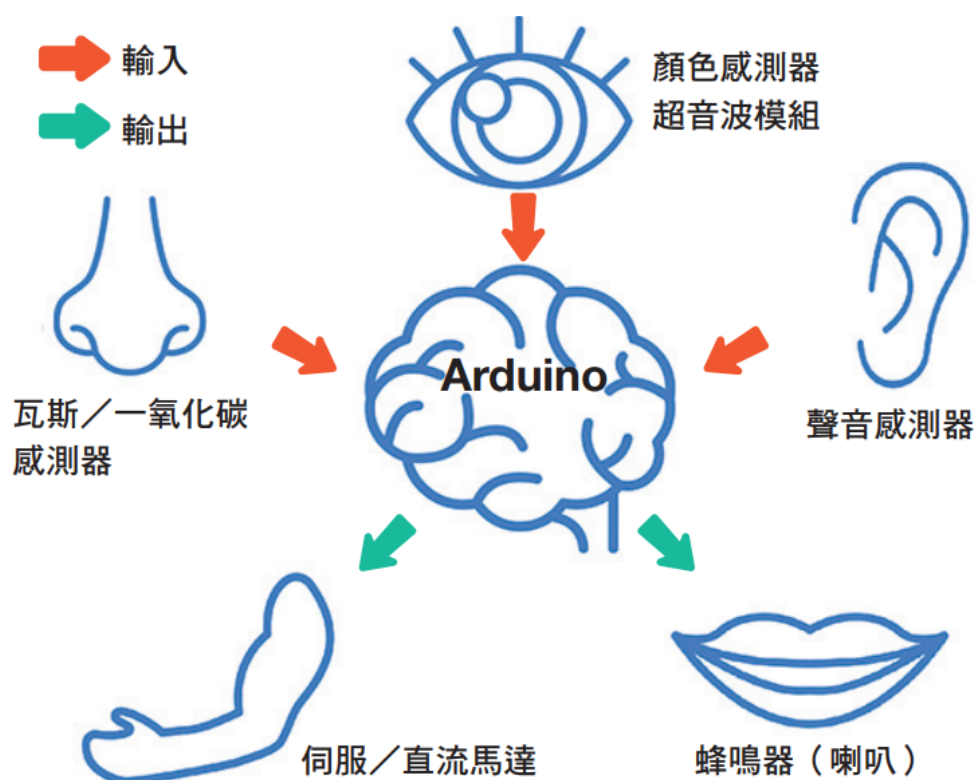
<https://drive.google.com/drive/folders/1ulmrCQXNe1XfYnHK33VEGZVq5wqeqQfD?usp=sharing>



硬體介紹

基本輸入與輸出

如同電腦主機擁有鍵盤、滑鼠與螢幕等不同的輸出入裝置，**ESP32 Gyro** 也擁有屬於自己的輸出入模組，這些模組大多是由一些特殊的感測元件組成。也由於 **ESP32 Gyro** 可以使用許多特殊的感測元件，所以 **ESP32 Gyro**與使用者之間便多了許多和電腦大相逕庭 的互動方式。



1-1.輸入裝置

如上圖紅箭頭所示，若將 **ESP32** 視為控制人體各器官的指揮中樞--大腦，那麼可以感測外界資訊、並將相關資訊回傳給 **ESP32**(大腦)的元件(器官)，都是屬於「輸入裝置」的一種。例如可以辨別顏色的顏色感測器和可以偵測障礙物距離的超音波模組，就像人體的眼睛一樣；而可以辨別聲音大小的聲音感測器，則有著如同耳朵般的功能；另外可以偵測瓦斯、一氧化碳濃度的感測器，則是扮演了鼻子的角色。如同以上這些會將所偵測到的環境數據回報給 **ESP32** 的感測元件，便統稱為 **ESP32** 的輸入裝置。

1-2.輸出裝置

反之則如上圖綠箭頭所示，若是由 **ESP32** (大腦) 主動控制、並使其能做出一些對應動作的元件 (器官)，便是屬於「輸出裝置」。例如可由 **ESP32** 決定旋轉方向及速度的伺服或直流馬達，就像是由大腦控制動作的手腳；而由 **ESP32** 決定發出何種旋律的蜂鳴器 (喇叭)，則像是由大腦決定要發出什麼聲音的嘴巴。諸如以上可由 **ESP32** 主動控制其動作的元件，均通稱為 **ESP32** 的輸出裝置。



<https://sites.google.com/view/esp-gyro/%E9%A6%96%E9%A0%81?authuser=3>

Gyro 常用腳位表			
ESP32	功能一	功能二	功能三
IO22	I2C - SCL		
IO21	I2C - SDA		
IO19	GVS (SG90)		
IO5	GVS (SG90)		
IO16	GVS (SG90)	紅綠燈- G	按鈕-2
IO4	GVS (SG90)	GVS 母座	
IO26	GVS (SG90)	L9110-A (TT馬達)	
IO25	GVS (SG90)		
IO33	GVS (SG90)	L9110-B (TT馬達)	按鈕-4
IO32	GVS (SG90)		按鈕-1
IO27	紅綠燈- R	L9110-C (TT馬達)	TFT 液晶
IO14	紅綠燈- Y		TFT 液晶
IO13	超音波- ECHO	L9110-D (TT馬達)	TFT 液晶
IO23	超音波- TRIG		
IO17	內建 WS2812		
IO12	TFT 液晶	內建 LED	
IO15	TFT 液晶	按鈕-3	

Gyro 擴充板常用腳位表

I2C 可外接其它模組:

如 OLED / PCA9685 / AI 模組 / 紅外線溫度感測

SPI 可接 TFT 液晶螢幕

GVS 腳位主要可用來接伺服馬達或其它感測器

1. 最多八組伺服馬達: 機械手臂、四足蜘蛛
2. 最多四組 TT 馬達: 麥克納姆輪車
3. 兩顆 TT 馬達+四顆伺服馬達: 遙控手臂車

硬體名稱: ESP Gyro 擴充板

尺寸規格: 58x80mm

支援兩款開發板: (ESP32 或 ESP32CAM 擇一使用)

A - ESP32 nodemcu-32s (窄版 38 pins)

B - ESP32CAM (進階使用者), 見 **ESP32CAM** 區說明

固定孔: 3mm

孔位: 四個相容 arduino uno 的固定孔, 方便對接現有的車體平台

供電方式

1. 直接使用 usb 接 ESP32 供電 (適合物聯網等相關低耗電應用)
2. 外接電源供電請注意, 擴充板只支援標準規範的接頭,

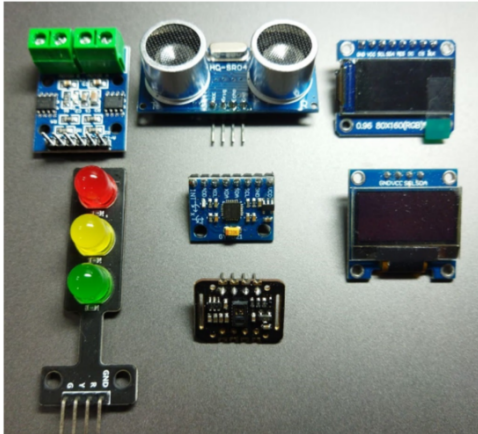
外部為負極, 內部為正極,

使用前請先確認, 避免損壞相關設備.

3. 外接電源, 可使用兩顆 3.7 V 鋰電 (14500 或 18650) 串聯成 7.4V (適合使用馬達等大電流裝置)
4. 外接 DC2.1 接頭的變壓器, 電壓 6.5V ~ 9V, 電流建議 1A (適合定點展示時)

擴充板使用時注意事項:

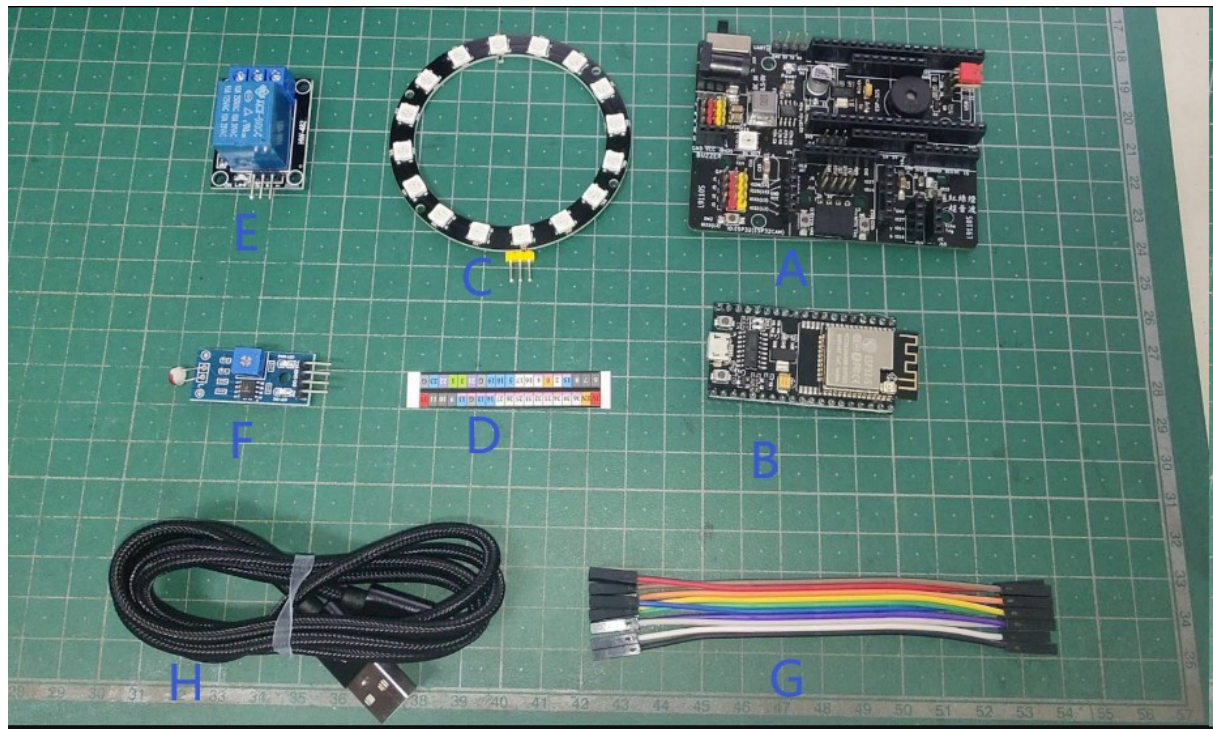
1. 只支援 **ESP32 NodeMCU-32s** 窄版 **38 腳位** 版本. 寬版不能使用
2. 插上開發板時, 請注意 **usb** 那邊請朝擴充版箭頭指的方向才是正確, 不然會造成開發板損壞
3. 使用多顆伺服馬達或 **TT** 直流馬達時, 請務必採用**外部電源供電**, 避免電流不夠造成誤動作
4. 燒錄程式時, 請務必把**外部電源關閉**, 避免燒錄失敗
5. 外接模組時, 請務必確認**正負極方向**是否正確, 避免開發板及模組損壞



支援常見的模組

- **SSD1306 OLED** (請注意腳位順序)
- **MPU6050** 六軸陀螺儀
- **HC-SR04** 超音波模組
- 紅綠燈模組
- 兩組 **L9110S DC** 馬達模組 (最多四路電機, 可玩麥克納姆輪的車子)
- **ST7789** or **ST7735 TFT** 液晶模組 (需特定腳位的排列可直接插上) - 見 **Arduino** 範例十一
- **MAX30102** 血氧模組 (共用超音波插座, 程式需特別處理) - 見 **Arduino** 範例十三
- 其它未列在上面的模組也可以用杜邦線再自行外接使用

檢查您的硬體清單



A: ESP-GYRO

B: ESP32

C: W2812 燈環

D: ESP32 腳位貼紙

E: 繼電器模組

F: 亮度檢知模組

開發環境介紹

安裝 **Arduino IDE** 與驅動程式 **Arduino IDE** 是由 **Arduino** 官方所提供的程式開發軟體，不論是 **Arduino** 程式的編寫上傳，還是 **Arduino** 各式開發板驅動程式的安裝，都需要使用到這套軟體。請先至 **Arduino** 官網 <http://arduino.cc> 下載 **Arduino** 官方的程式開發環境軟體 **Arduino IDE**。

Arduino IDE 環境設定

<https://sites.google.com/view/esp-gyro/%E9%A6%96%E9%A0%81/arduino-ide-%E7%92%B0%E5%A2%83%E8%A8%AD%E5%AE%9A?authuser=3>

直接下載打包版本：

https://drive.google.com/drive/folders/1LwcvH1k_BdsYTxfBFnAR3OJc6bb24Sli

內建範例所需安裝的 **library** 已打包好，若還需額外的 **library**，請再自行安裝

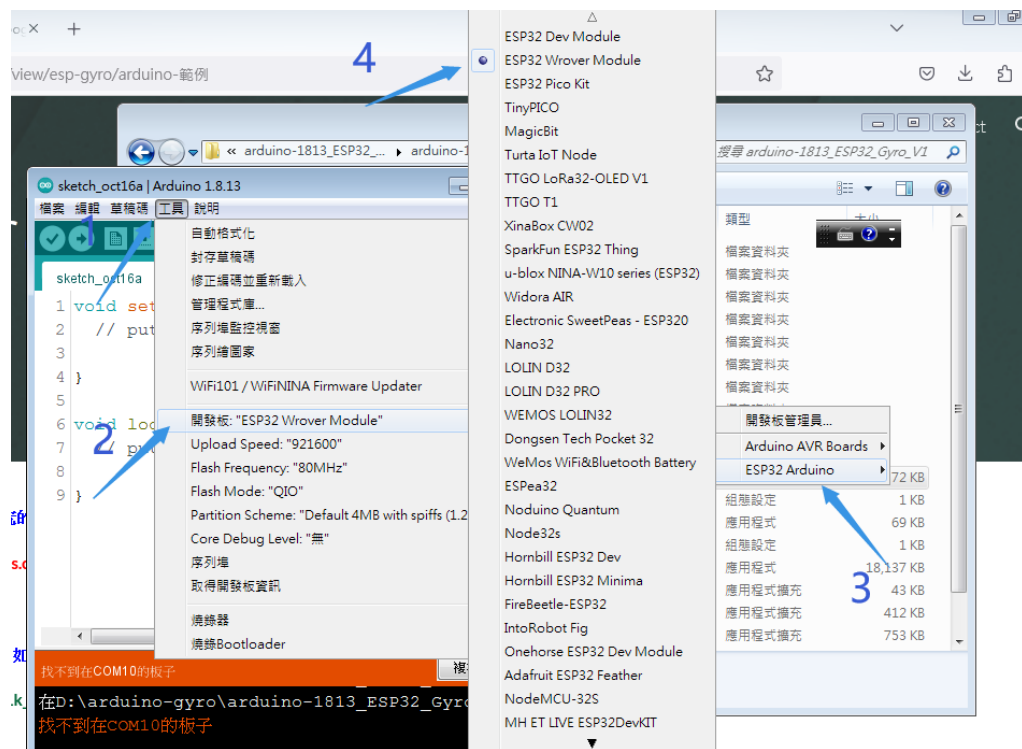
-> 如果為初學者或想較方便使用的話，建議可直接使用已打包好 **library** 的 **arduino ide** 離線版

-> 解開時請放在根目錄下，避免路徑過長的問題。

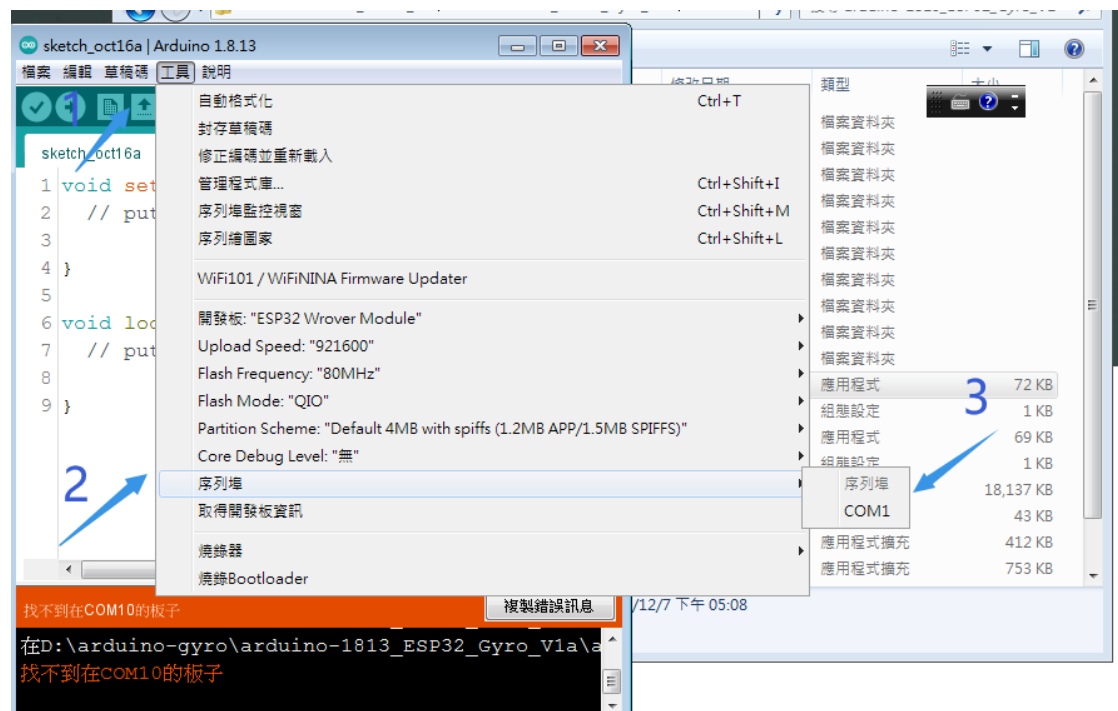
-> 解開後，請直接用目錄下的 **arduino.exe** 即可使用

可選相關範例來修改使用：

設定開發板



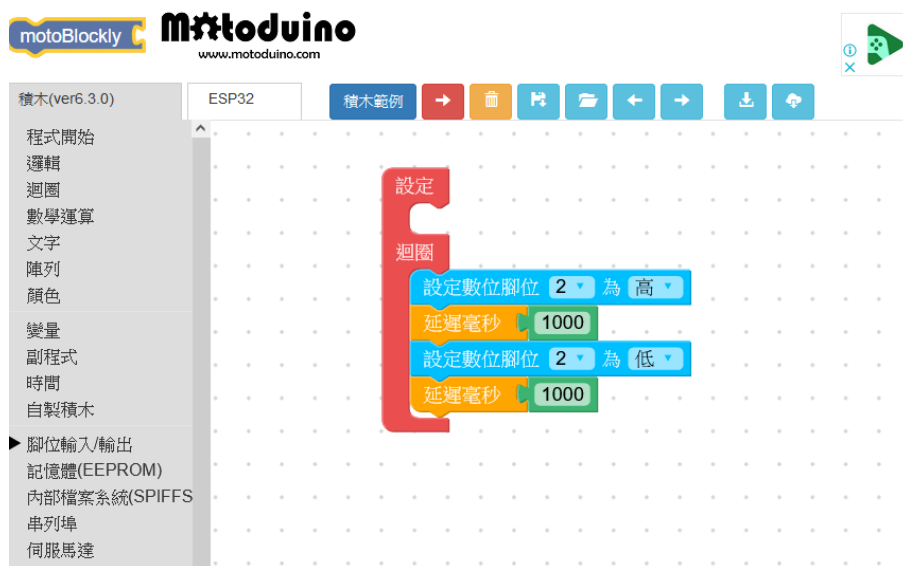
確定通訊埠



Motoduino

motoBlockly 簡介

Arduino IDE 安裝完畢後就可以開始編寫 **Arduino** 程式，但若是對 **Arduino** 或寫 **Code** 不熟悉的初學者，建議可試著從簡單的圖控式程式編輯軟體開始入門。**motoBlockly** 是由慧手科技所開發的 **Arduino** 線上圖控式程式編輯軟體，其利用 程式積木堆疊來編寫 **Arduino** 程式的方式，與另一種圖控式編輯軟體 **App Inventor** 非常類似，對於想嘗試自行編寫程式的初學者來說非常容易上手。使用者只須把 **Arduino** 要執行的動作依序將積木堆疊起來，**motoBlockly** 便可將所堆疊的程式積木 轉換成 **Arduino** 的程式碼，而且還能將程式直接就編譯上傳到 **Arduino** 開發板上。



設定開發板

The image displays two screenshots of the Motoduino web IDE interface, illustrating the steps to set up a development board.

Top Screenshot: The interface shows the "ESP32" board selected in the top right dropdown menu. A blue arrow labeled "1" points to the dropdown menu, and another blue arrow labeled "2" points to the "ESP32" option in the list. The main workspace contains a block diagram with a "設定" (Setup) block, a "迴圈" (Loop) block, and two "設定數位腳位" (Set digital pin) blocks, each followed by a "延遲毫秒" (Delay in milliseconds) block set to 1000.

Bottom Screenshot: The interface shows the "ESP32" board selected in the top right dropdown menu. A blue arrow labeled "Save file" points to the "Save" icon (a floppy disk) in the top toolbar. Another blue arrow labeled "Load file" points to the "Load" icon (a folder) in the top toolbar. The main workspace contains a block diagram with a "設定" (Setup) block, a "迴圈" (Loop) block, and an "RGBLED燈條" (RGB LED strip) block set to "腳位# 4" (Pin # 4) and "全暗" (All off).

活動：積木程式上傳使LED開啟和關閉

選擇開啟 [motoBlockly 積木範例](#)裡的「LED 閃爍」程式範例，其為控制 ESP32 開發板上數位腳位 IO 2上的 LED 程式，點選後 motoBlockly 便會匯入並顯示此範例程式的程式積木堆疊狀態。

首先，您的程式必須告訴 ESP32 使用pinMode函數設置IO 2 (或IO12) 的輸出方向：`pinMode(pin, mode)`。所述腳位pin參數是一個數字I/O管腳的編號，和接腳模式必須是INPUT或OUTPUT (輸入/輸出)。

範例程式積木開啟完成後，點選下圖中 motoBlockly 的「ESP32」標籤頁，先將整個範例的程式積木轉換成 Arduino 程式碼後，再按下迴文針會複製程式碼，再貼到arduino IDE。

按下上傳即可傳到ESP32 電路板中。可以觀查電路板上IO12LED 閃爍。



積木(ver6.3.0)

ESP32



```
/* Sketch was generated by motoblockly
Website: http://www.motoblockly.com
Author: www.motoduino.com
Date: Mon Oct 16 2023 21:30:10 GMT+0800
*/
void setup()
{
  pinMode(12, OUTPUT);
}

void loop()
{
  digitalWrite(12,HIGH);
  delay(1000);
  digitalWrite(12,LOW);
  delay(1000);
}
```

1

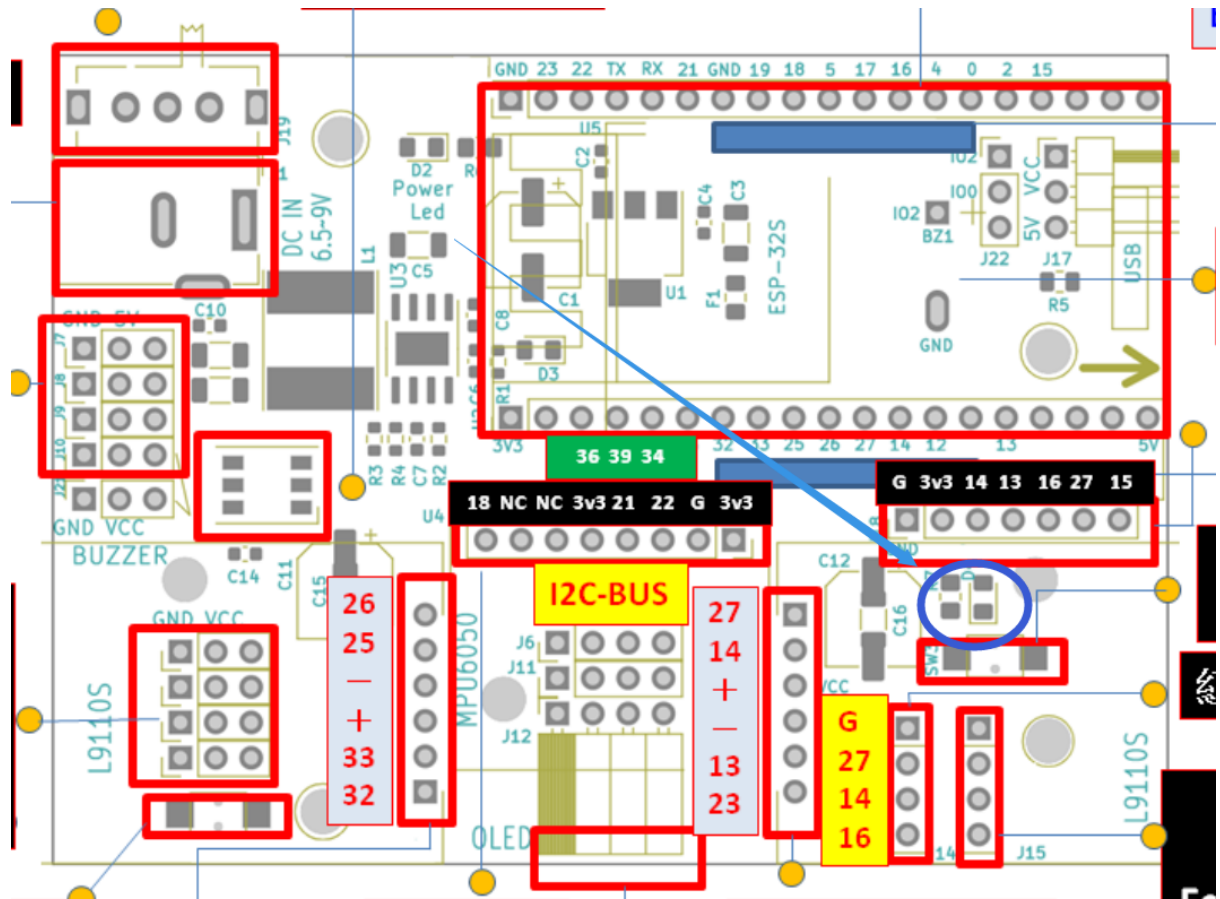
2

sketch_oct16b | Arduino 1.8.13

檔案 編輯 草稿碼 工具 說明

✓ ↻ 📄 ⬆ ⬇

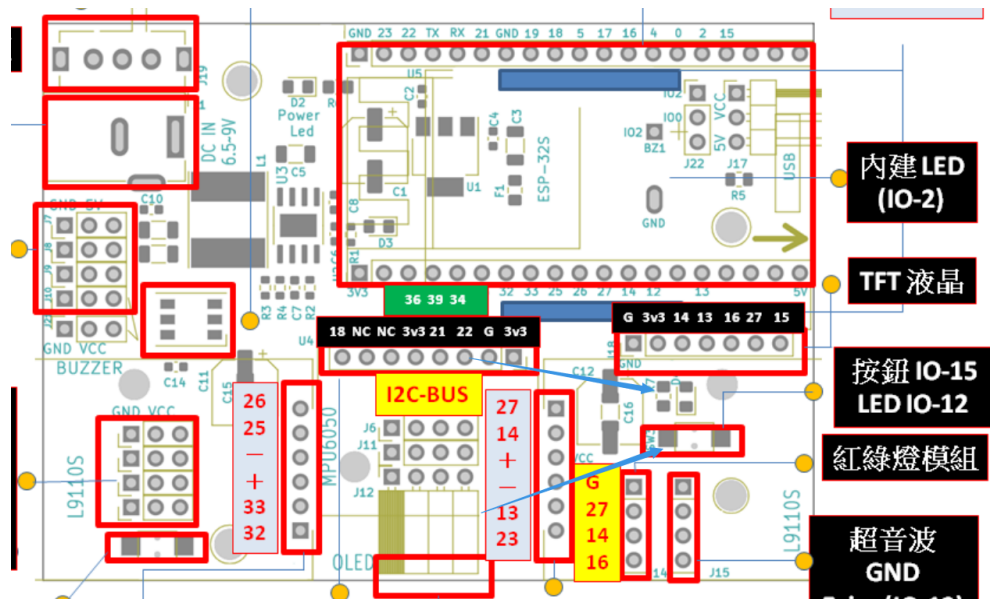
```
sketch_oct16b $
1  Author: www.motoduino.com
2
3  Date: Mon Oct 16 2023 21:27:24 GMT+0800
4  */
5
6  void setup()
7  {
8    pinMode(2, OUTPUT);
9
10 }
11
12
13 void loop()
14 {
15   digitalWrite(2,HIGH);
16   delay(1000);
17   digitalWrite(2,LOW);
18   delay(1000);
19
20 }
```



活動：按鍵讀取與LED控制



```
void setup()  
{  
  pinMode(15, INPUT_PULLUP);  
  pinMode(LED_PIN, OUTPUT);  
}  
  
void loop()  
{  
  if (digitalRead(15) == 0) {  
    digitalWrite(LED_PIN, HIGH);  
  } else {  
    digitalWrite(LED_PIN, LOW);  
  }  
}
```



燒錄完成後,按下IO15按鈕,LED 22 會亮起來,放開後會熄滅

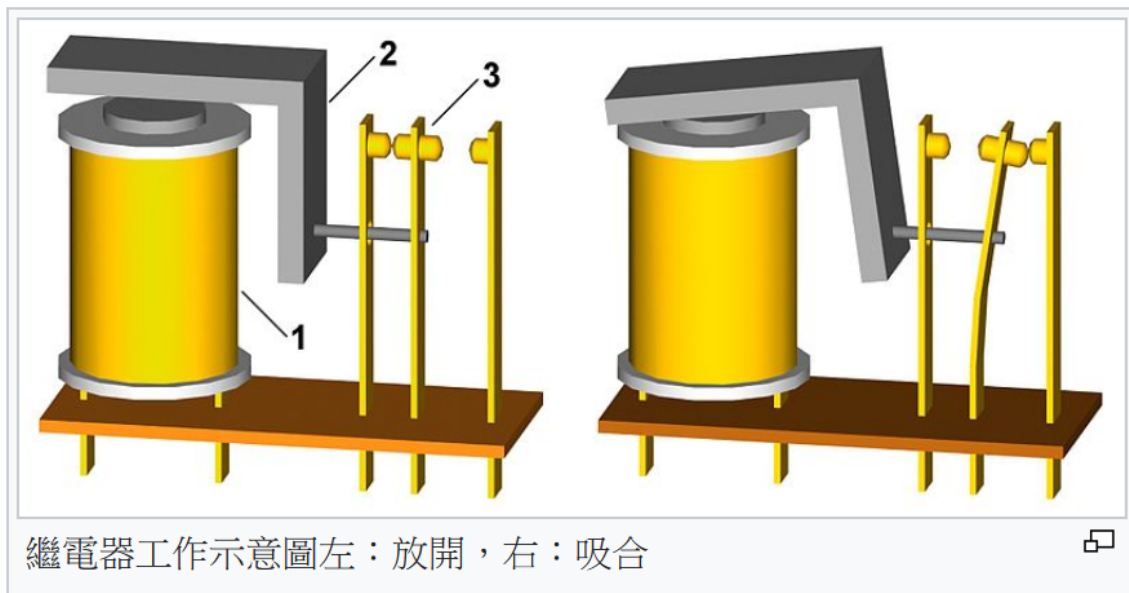
活動：按鍵讀取與RELAY控制

RELAY的作用:電路板只能控制小的電壓和電流,如果要控制家電或其他電路需要**RELAY**.

一般**RELAY**除了有線圈控制接點外還會有三個接點

其中一個為共點(**com**)

另兩個為常開(**NO**)及常閉(**NC**)



資料來源wiki

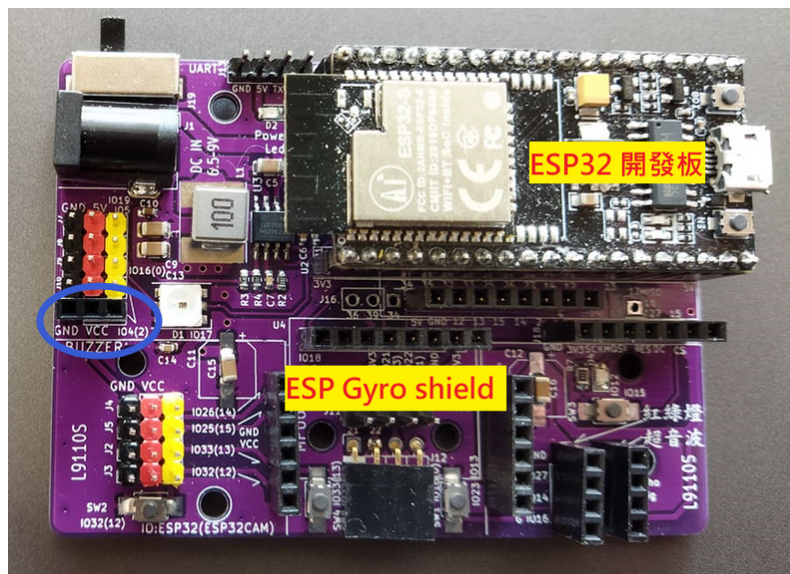
<https://zh.wikipedia.org/zh-tw/%E7%BB%A7%E7%94%B5%E5%99%A8>

將Relay 模組接到IO4母座上,注意接腳方向

-接GND ；

+接VCC ；

S接IO4





```
/* Sketch was generated by motoblockly
Website: http://www.motoblockly.com
Author: www.motoduino.com
Date: Mon Oct 16 2023 21:30:10 GMT+0800
*/
int RELAY_PIN=4;

void setup()
{
  pinMode(15, INPUT_PULLUP);
  pinMode(RELAY_PIN,OUTPUT);
}

void loop()
{
  if (digitalRead(15) == 0) {
    digitalWrite(RELAY_PIN,HIGH);
  } else {
    digitalWrite(RELAY_PIN,LOW);
  }
}
```

燒錄完成後，

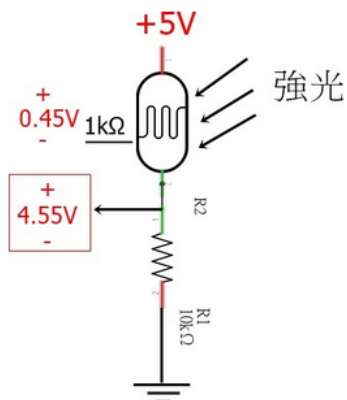
按下IO15按鈕，**RELAY**吸合：**NO**導通；**NC**斷開

放開IO15按鈕，**RELAY**吸合：**NO**斷開；**NC**導通

活動：SENSOR讀取與RELAY控制

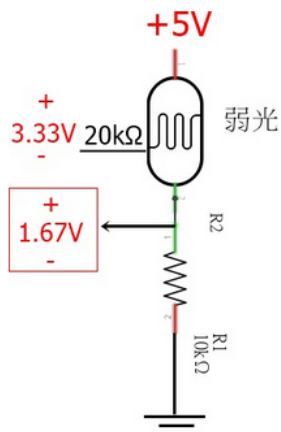
光敏電阻模組：用來偵測環境亮度，一般有三隻腳和四隻腳的，三隻腳大多為類比輸出，直接把分壓過後電壓傳出，四隻腳的比三隻腳內部多一個比較器和一個可變電阻，可以調整可變電阻設定閾值，把資料傳到數位輸出。

本活動分為兩部分，一個部分單純數位輸出到**ESP32** 去控制**RELAY**，第二個部分加上類比輸出及波型的顯示。



假設在強光時，光敏電阻阻值為 $1k\Omega$ 。

從中間擷取到的電壓則為4.55V。



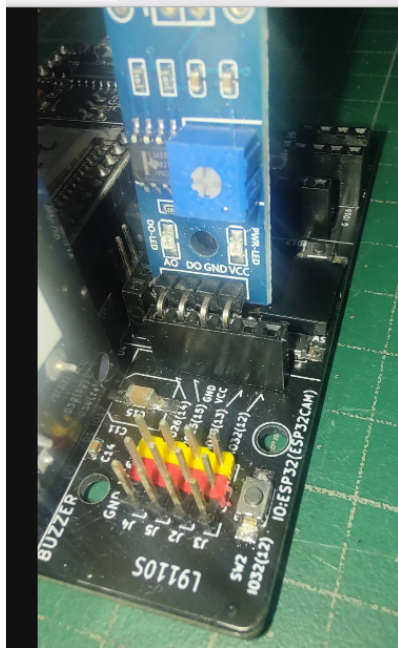
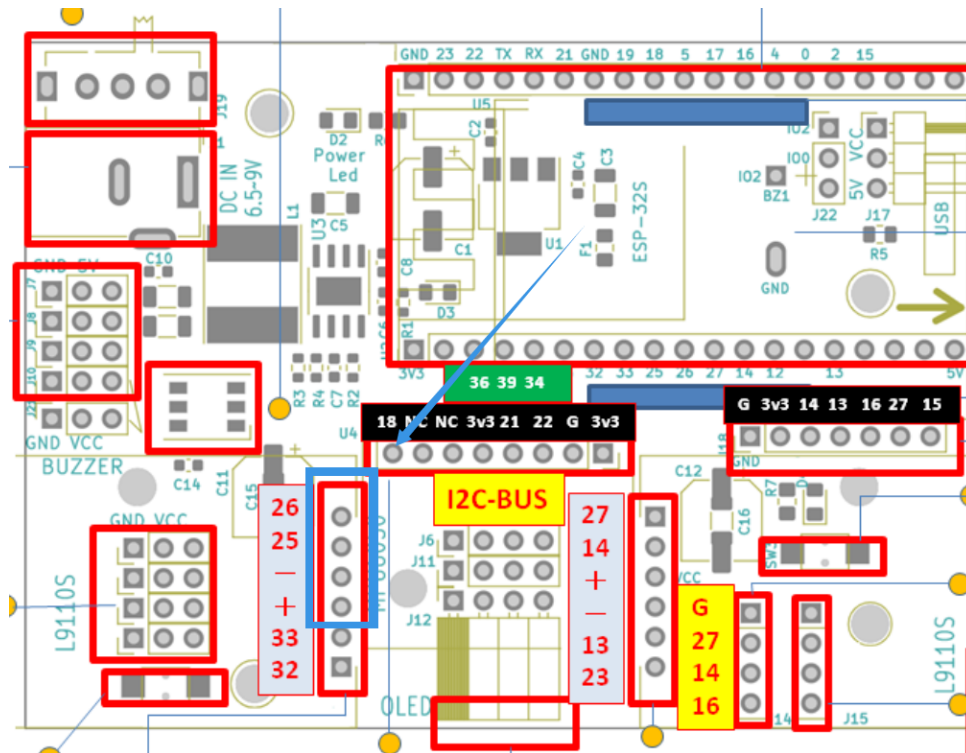
假設在弱光時，光敏電阻阻值為 $20k\Omega$ 。

從中間擷取到的電壓則為1.67V。

從上面電路即可實現，當光越強，電壓越高，反之光越弱，電壓越低。

活動接線方式：

繼電器模組延續上一活動接在IO4，再將光敏電阻模組接到下圖位置，VCC接VCC；GND接GND；DO接IO25；AO接IO26。



宣告全域變數 RELAY_PIN 為 int 資料 4
宣告全域變數 SENSOR_PIN 為 int 資料 25

設定

設定 Serial 傳輸率 115200 bps
腳位模式設定(PinMode) 腳位# RELAY_PIN 為 輸出
腳位模式設定(PinMode) 腳位# SENSOR_PIN 為 輸入上拉(電阻)

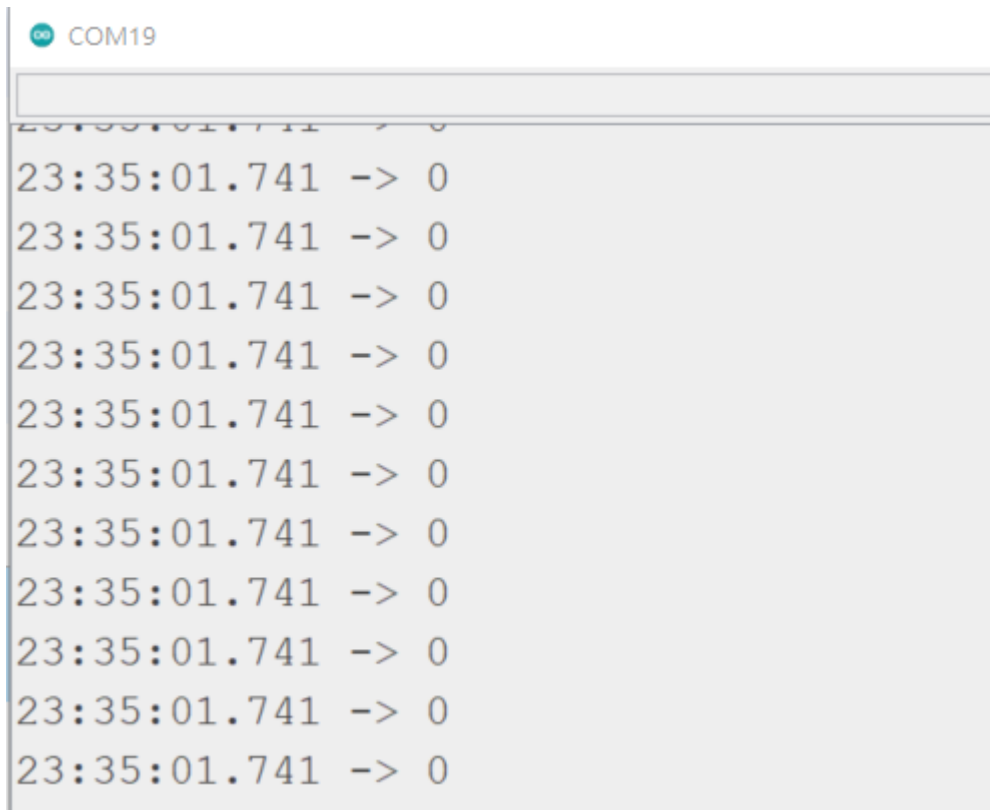
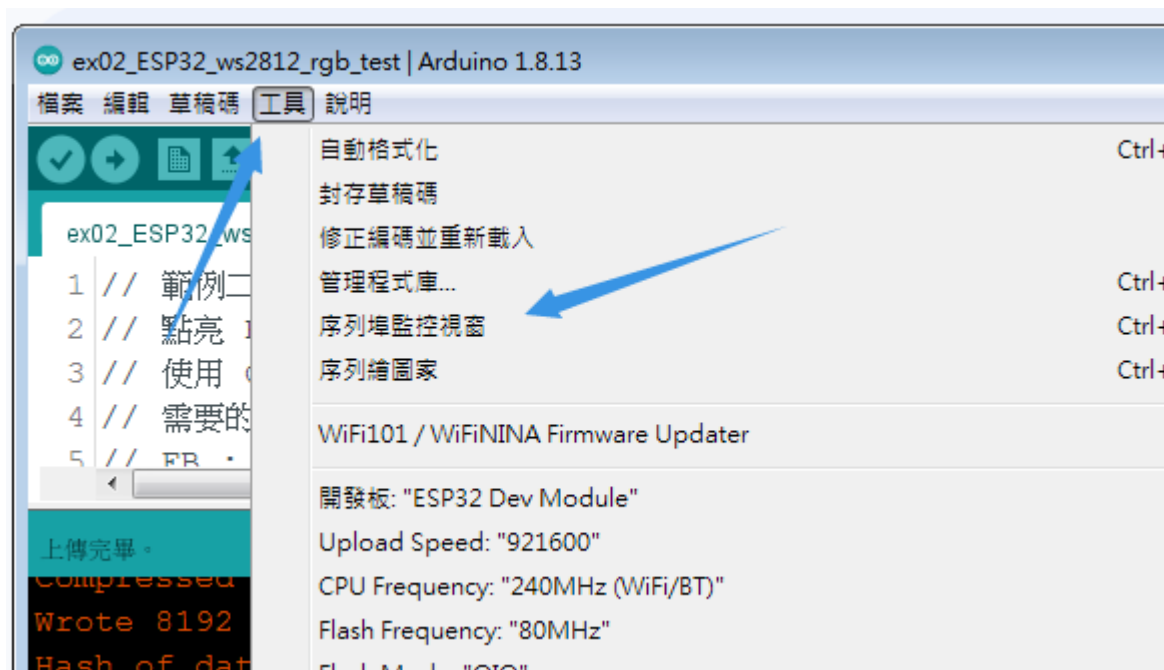
迴圈

Serial 輸出格式 10進制 數位讀出變數腳位 SENSOR_PIN
如果 數位讀出變數腳位 SENSOR_PIN = 0
執行 設定數位腳位 RELAY_PIN 為 高
否則 設定數位腳位 RELAY_PIN 為 低

```
/* Sketch was generated by motoblockly
Website: http://www.motoblockly.com
Author: www.motoduino.com
Date: Mon Oct 16 2023 21:30:10 GMT+0800
*/
int RELAY_PIN=4;
int SENSOR_PIN=25;

void setup()
{
  Serial.begin(115200);
  pinMode(RELAY_PIN,OUTPUT);
  pinMode(SENSOR_PIN,INPUT_PULLUP);
}

void loop()
{
  Serial.println(digitalRead(SENSOR_PIN),DEC);
  if (digitalRead(SENSOR_PIN) == 0) {
    digitalWrite(RELAY_PIN,HIGH);
  } else {
    digitalWrite(RELAY_PIN,LOW);
  }
}
```



燒錄完成後開啟串列埠監控視窗，可以由串列埠監控視窗觀查到光敏電阻模組輸出的情況，正常光照顧下會顯示0
當用手摀住光敏電阻模擬天黑後，則可以看到串列埠輸出為1，同一時間繼電器吸合。

宣告全域變數 RELAY_PIN 為 int 資料 4

宣告全域變數 SENSOR_PIN 為 int 資料 25

宣告全域變數 SENSOR_AO_PIN 為 int 資料 26

設定

設定 Serial 傳輸率 115200 bps

腳位模式設定(PinMode) 腳位# RELAY_PIN 為 輸出

腳位模式設定(PinMode) 腳位# SENSOR_PIN 為 輸入上拉(電阻)

迴圈

Serial 輸出格式 10進制 數位讀出變數腳位 SENSOR_PIN

Serial 印出訊息到同一行 “亮度: ”

Serial 輸出格式 10進制 類比讀出變數腳位 SENSOR_AO_PIN

如果 數位讀出變數腳位 SENSOR_PIN = 0

執行 設定數位腳位 RELAY_PIN 為 高

否則 設定數位腳位 RELAY_PIN 為 低

```
/* Sketch was generated by motoblockly
Website: http://www.motoblockly.com
Author: www.motoduino.com
Date: Mon Oct 16 2023 21:30:10 GMT+0800
*/
int RELAY_PIN=4;
int SENSOR_PIN=25;
int SENSOR_AO_PIN=26;

void setup()
{
  Serial.begin(115200);
  pinMode(RELAY_PIN,OUTPUT);
  pinMode(SENSOR_PIN,INPUT_PULLUP);
}

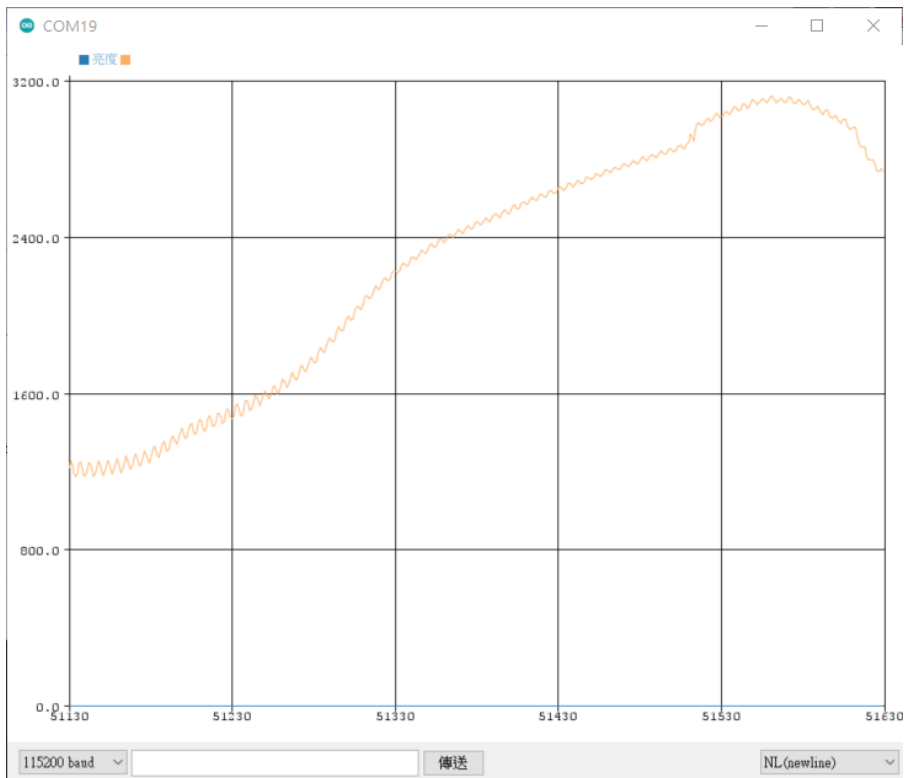
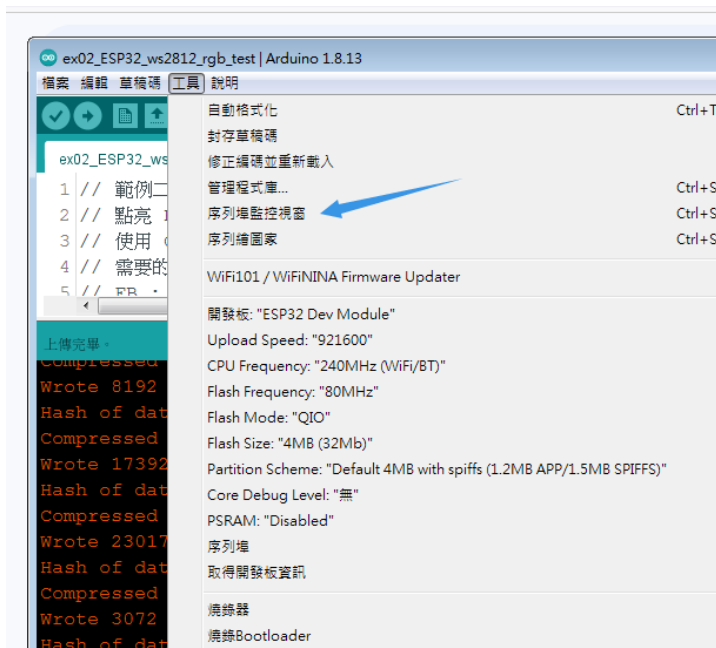
void loop()
{
  Serial.println(digitalRead(SENSOR_PIN),DEC);
  Serial.print("亮度: ");
  Serial.println(analogRead(SENSOR_AO_PIN),DEC);
  if (digitalRead(SENSOR_PIN) == 0) {
    digitalWrite(RELAY_PIN,HIGH);
  } else {
    digitalWrite(RELAY_PIN,LOW);
  }
}
```



```
COM19
23:37:28.223 -> 0
23:37:28.223 -> 亮度: 884
23:37:28.223 -> 0
23:37:28.223 -> 亮度: 866
23:37:28.223 -> 0
23:37:28.223 -> 亮度: 887
23:37:28.223 -> 0
23:37:28.223 -> 亮度: 865
23:37:28.223 -> 0
23:37:28.223 -> 亮度: 823
23:37:28.223 -> 0
23:37:28.223 -> 亮度: 812
23:37:28.223 -> 0
23:37:28.223 -> 亮度: 831
23:37:28.223 -> 0
23:37:28.223 -> 亮度: 867
```

顯示波型圖：

先關閉串列埠監控視窗，再由下圖指示開啟序列繪圖家，用手不斷撫斷光線，觀查序列繪圖家顯示結果。



活動:藍牙通訊

手機端:下載arduino blue control

 Google Play Games Apps Movies Books Kids

Arduino Bluetooth Control

broxcode
Contains ads

4.4★
790 reviews

100K+
Downloads

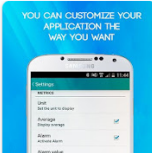
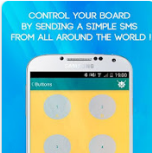
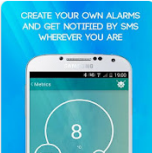
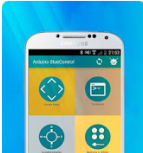
 3+
Rated for 3+ 

Install on more devices

 Share

 This app is available for your device





App support 

More apps to try 

 Disney+



```

/* Sketch was generated by motoblockly
Website: http://www.motoblockly.com
Author: www.motoduino.com
Date: Mon Oct 16 2023 21:30:10 GMT+0800
*/
#include "BluetoothSerial.h"

BluetoothSerial SerialBT;

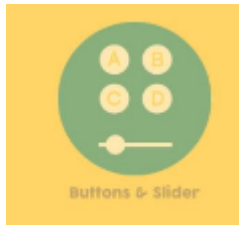
void setup()
{
  Serial.begin(115200);
  SerialBT.begin("ESP32test");
}

void loop()
{
  if (SerialBT.available()) {
    Serial.write(SerialBT.read());
  }
  if (Serial.available() > 0) {
    char c=Serial.read();
    SerialBT.println((String(c)));
  }
}

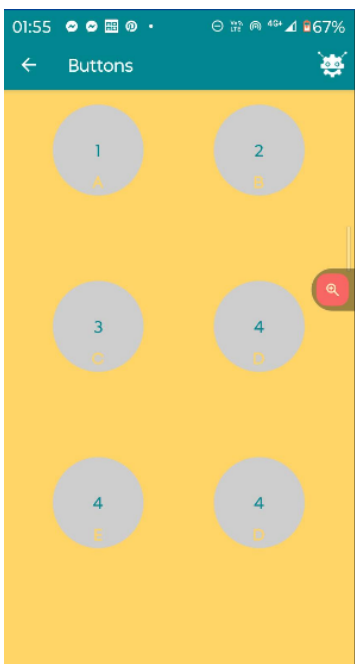
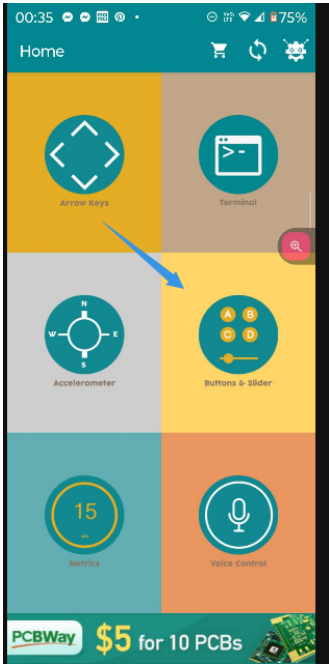
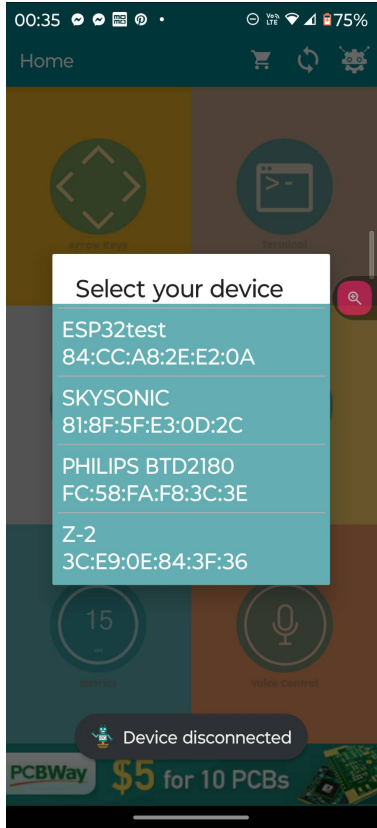
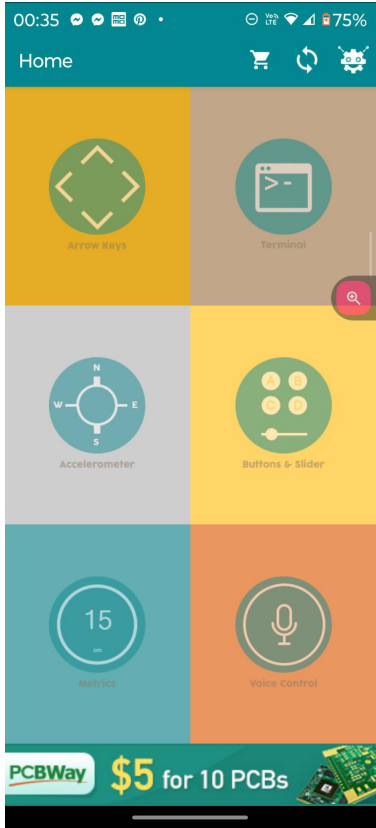
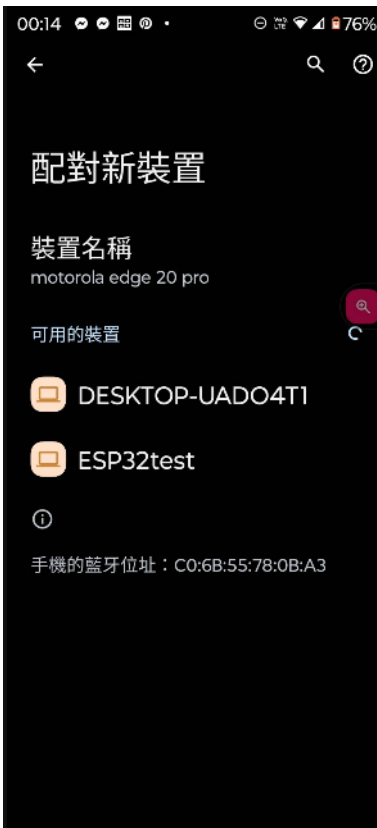
```

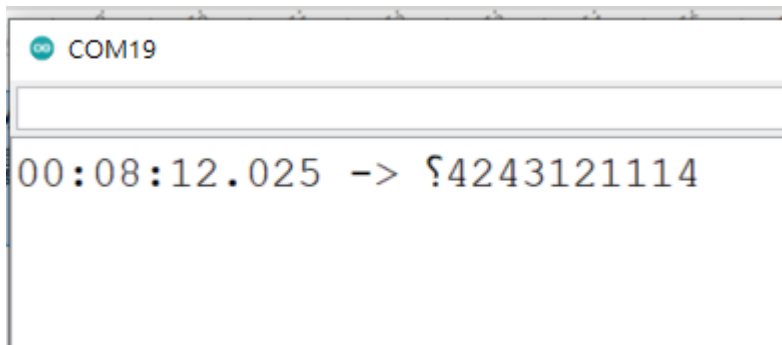
燒錄完成後在手機>設定>已連線裝置>配對新裝置 中找到你的設備並連絡後，開啟**Arduino Bluetooth control**

點左上角 ，找尋你的裝置，點擊後連線，按



後選擇要輸入數字並觀查串列埠監控視窗會顯示你所數入的數值。





活動：藍牙通訊與RELAY控制

延續上一個活動，我們對藍牙傳送過來的字元做判斷後去控制RELAY。

將RELAY接到IO4。



```

/* Sketch was generated by motoblockly
Website: http://www.motoblockly.com
Author: www.motoduino.com
Date: Mon Oct 16 2023 21:30:10 GMT+0800
*/
#include "BluetoothSerial.h"

int RELAY_PIN=4;
BluetoothSerial SerialBT;

void setup()
{
  Serial.begin(115200);
  pinMode(RELAY_PIN,OUTPUT);
  SerialBT.begin("ESP32test");
}

void loop()
{
  char BLE_CMD=0;
  if (SerialBT.available()) {
    BLE_CMD = SerialBT.read();
    if (BLE_CMD == 49) {
      digitalWrite(RELAY_PIN,HIGH);
    }
    if (BLE_CMD == 50) {
      digitalWrite(RELAY_PIN,LOW);
    }
  }
}

```

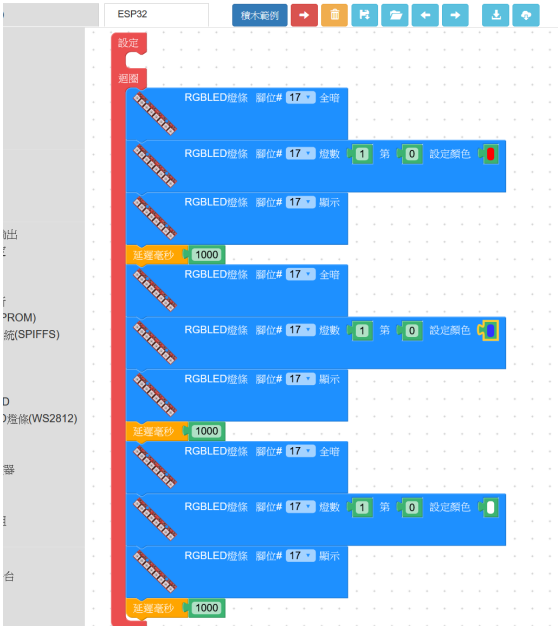
燒錄完成後，當手機輸入1時繼電器吸合，當手機輸入2時繼電器放開。

活動: WS2812燈環

WS2812 RGB LED，它是內建驅動晶片功能的LED，驅動方式採串列進出。

因此不管你要控制幾顆 **RGB LED**，都只需要使用 **3 支腳位**就足夠了，它可以將數據用串列的方式傳輸下去。

ESP32-GYRO電路版上預設有一個**W2812**，本次活動先驅動這個**W2812**，後續再外接**16位W2812**燈環。



```
/* Sketch was generated by motoblockly
Website: http://www.motoblockly.com
Author: www.motoduno.com
Date: Mon Oct 16 2023 21:30:10 GMT+0800
*/
#include <Adafruit_NeoPixel.h>

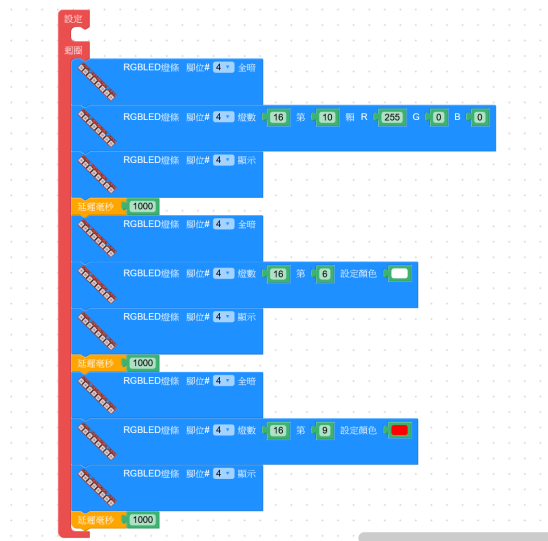
Adafruit_NeoPixel strip17(17, NEO_GRB + NEO_KHZ800);

void setup()
{
  strip17.begin();
}

void loop()
{
  strip17.clear();
  strip17.show();
  colorled17(0, "ff0000");
  strip17.show();
  delay(1000);
  strip17.clear();
  strip17.show();
  colorled17(0, "3333ff");
  strip17.show();
  delay(1000);
  strip17.clear();
  strip17.show();
  colorled17(0, "ffffff");
  strip17.show();
  delay(1000);
}

void colorled17(int number, String c) {
  long rgb_number = strtol(&c[1], NULL, 16);
  int r = rgb_number >> 16;
  int g = rgb_number >> 8 & 0xFF;
  int b = rgb_number & 0xFF;
  strip17.setPixelColor(number, strip17.Color(r, g, b));
}
```

將16位WS2812光環接到IO4，



```
/* Sketch was generated by motoblockly
Website: http://www.motoblockly.com
Author: www.motoduno.com
Date: Mon Oct 16 2023 21:30:10 GMT+0800
*/
#include <Adafruit_NeoPixel.h>

Adafruit_NeoPixel strip4(16,4, NEO_GRB + NEO_KHZ800);

void setup()
{
  strip4.begin();
}

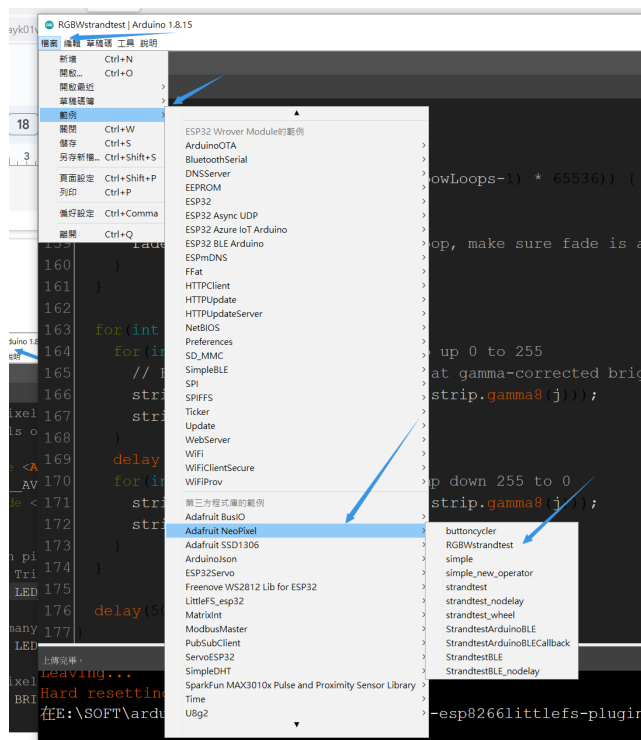
void loop()
{
  strip4.clear();
  strip4.show();
  strip4.setPixelColor(10,strip4.Color(255,0,0));
  strip4.show();
  delay(1000);
  strip4.clear();
  strip4.show();
  colorled4(6,"ffffff");
  strip4.show();
  delay(1000);
  strip4.clear();
  strip4.show();
  colorled4(9,"ff0000");
  strip4.show();
  delay(1000);
}

void colorled4(int number,String c) {
  long rgb_number = strtol(&c[1], NULL, 16);
  int r = rgb_number >> 16;
  int g = rgb_number >> 8 & 0xFF;
  int b = rgb_number & 0xFF;
  strip4.setPixelColor(number,strip4.Color(r,g,b));
}
```

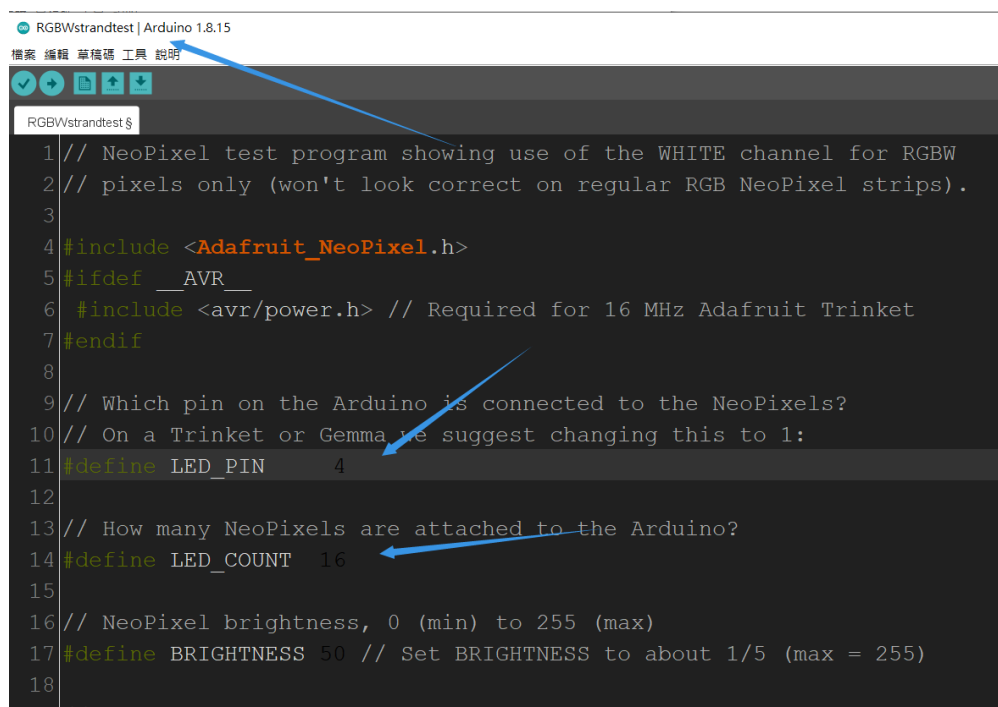
燒錄後觀查WS2812光環變化

活動: **ARDUINO** 裡的**WS2812**範例

Arduino IDE有很多**WS2812**的範例, 我們一起來測試一下吧。



修改腳位和WS2812數量



```
buttoncycler | Arduino 1.8.15
檔案 編輯 草稿碼 工具 說明
buttoncycler $
1 // Simple demonstration on using an input device to trigger changes on y
2 // NeoPixels. Wire a momentary push button to connect from ground to a
3 // digital IO pin. When the button is pressed it will change to a new pi
4 // animation. Initial state has all pixels off -- press the button once
5 // start the first animation. As written, the button does not interrupt
6 // animation in-progress, it works only when idle.
7
8 #include <Adafruit_NeoPixel.h>
9 #ifdef __AVR__
10 #include <avr/power.h> // Required for 16 MHz Adafruit Trinket
11 #endif
12
13 // Digital IO pin connected to the button. This will be driven with a
14 // pull-up resistor so the switch pulls the pin to ground momentarily.
15 // On a high -> low transition the button press logic will execute.
16 #define BUTTON_PIN 15
17
18 #define PIXEL_PIN 4 // Digital IO pin connected to the NeoPixels.
19
20 #define PIXEL_COUNT 16 // Number of NeoPixels
21
22 // ...
```

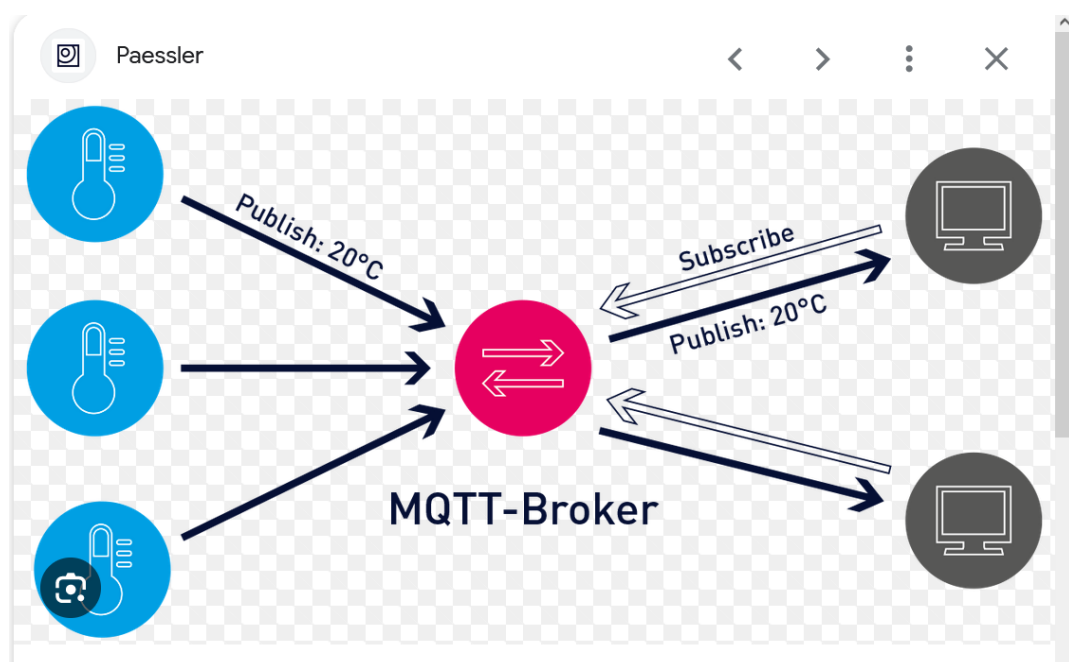
活動:MQTT 與RELAY控制

MQTT [編輯]

條目 討論 漢 漢 臺灣正體 ▾

維基百科，自由的百科全書

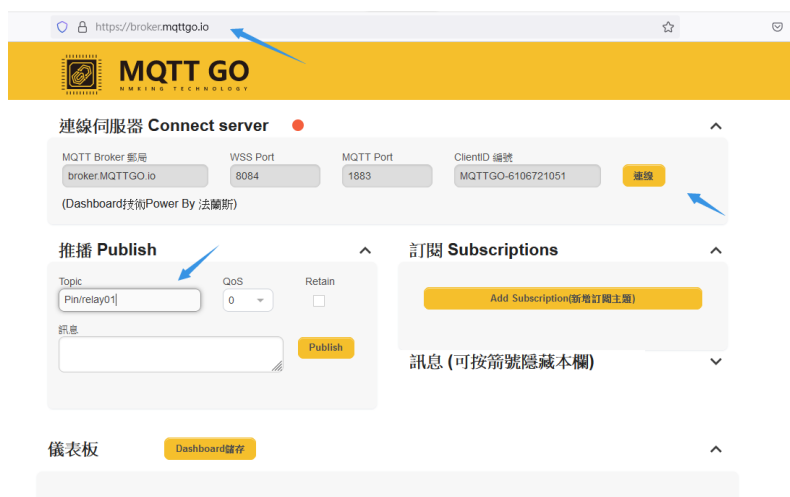
訊息佇列遙測傳輸（英語：Message Queuing Telemetry Transport，**MQTT**^[1]）是 **ISO 標準**（ISO/IEC PRF 20922）^[2]下基於發布（Publish）/訂閱（Subscribe）範式的訊息協定，可視為「資料傳遞的橋梁」^[3]。它工作在**TCP/IP協定族**上，是為硬體效能低下的遠端裝置以及網路狀況糟糕的情況下而設計的**發布/訂閱型**訊息協定。為此，它需要一個**訊息中介軟體**（如**HTTP**），以解決當前繁重的資料傳輸協定。^[3]



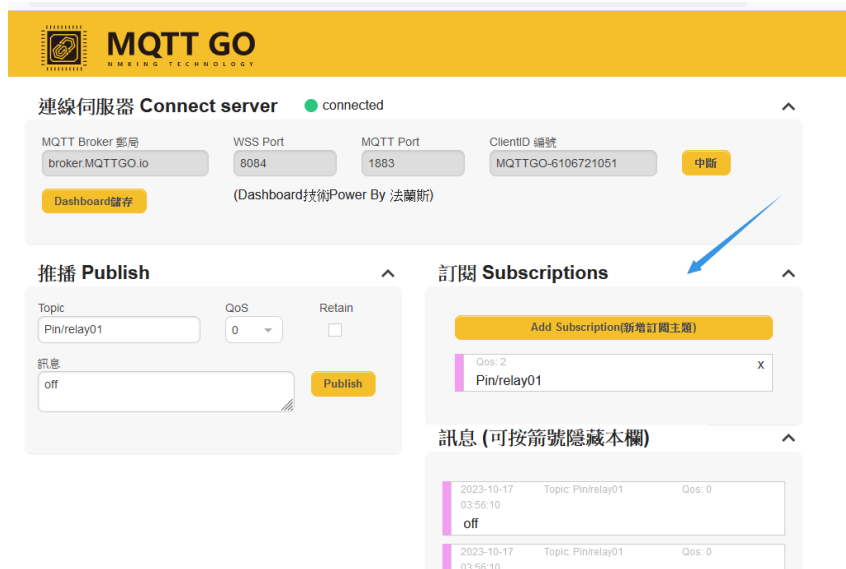
MQTT broker

<https://broker.mqttgo.io/>

連接到**MQTT GO** 網站後直接連線再設置推播及訂閱主題。



注意在**Topic**中請輸入自己訂義的主題.並且和板子上的設定一樣。

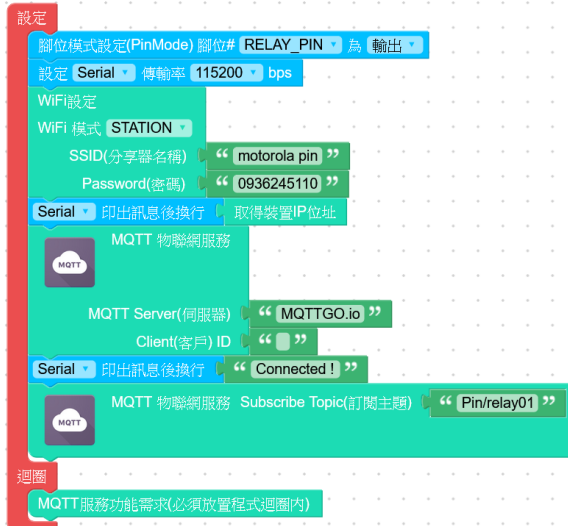


設定完後在推播的訊息輸入文字，可以在訂閱的訊息中得到該訊息及收到的時間。

設置esp32 為Subscription (設置網路連線,MQTT Broker及Topic)

宣告全域變數 RELAY_PIN 為 int 資料 4

宣告全域變數 GetData 為 String 資料 “ ”



```
#include <WiFi.h>
#include <PubSubClient.h>

int RELAY_PIN=4;
String GetData="";
WiFiClient client;
PubSubClient mqtt_client(client);

void setup()
{
  Serial.begin(115200);
  mqtt_client.setCallback(callback);
  pinMode(RELAY_PIN,OUTPUT);
  WiFi.mode(WIFI_STA);
  WiFi.begin("motorola pin","0936245110");
  while(WiFi.status() != WL_CONNECTED) {
    delay(500);
    Serial.print(".");
  }
  Serial.println((WiFi.localIP().toString()));
  mqtt_client.setServer("MQTTGO.io",1883);
  while (!mqtt_client.connect("")) {
    Serial.print(".");
    delay(1000);
  }
  Serial.println("Connected !");
  mqtt_client.subscribe("Pin/relay01");
}
```

```
void loop()
{
  mqtt_client.loop();
}

String getTopicData(String myTopic, String inTopic, byte* payload ,unsigned int length){
  String TopicData;
  if(inTopic == myTopic){
    for (int i = 0; i < length; i++) {
      TopicData += (char)payload[i];}
    return TopicData;
  }else{
    return "";
  }
}

void callback(char* topic, byte* payload, unsigned int length){
  GetData = getTopicData("Pin/relay01",topic, payload,length);
  Serial.println(GetData);
  if (GetData == "on") {
    digitalWrite(RELAY_PIN,HIGH);
  } else if (GetData == "off") {
    digitalWrite(RELAY_PIN,LOW);
  }
}

<
```

推播 Publish

Topic: Pin/relay01 QoS: 0 Retain: ☐

訊息: off Publish

訂閱 Subscriptions

Add Subscription(新增訂閱主題)

Qos: 2 Pin/relay01 X

訊息 (可按箭號隱藏本欄)

Time	Topic	Qos	Message
2023-10-17 04:03:57	Pin/relay01	0	off
2023-10-17 04:03:56	Pin/relay01	0	on
2023-10-17	Pin/relay01	0	

燒錄完成後:MQTT GO 網站中Publish >訊息分別輸入on ; off (注意大小寫)後可以在網站中的Subscriptions看到訂閱的資料, 同時板子上的繼電器也可以被控制吸合和斷開。

App inventor2 與MQTT

App inventor2 為MIT 開發專為快速開發安卓手機APP的程式

<https://appinventor.mit.edu/>

加入MQTT 功能

由於App inventor2 沒有MQTT功能還好他們開放其他人可以開發外卦的套件, 到

<https://ullisroboterseite.de/android-AI2-PahoMQTT-en.html>

點擊

Download

[The ZIP archive UrsAI2PahoMqtt for download](#). The archive contains the

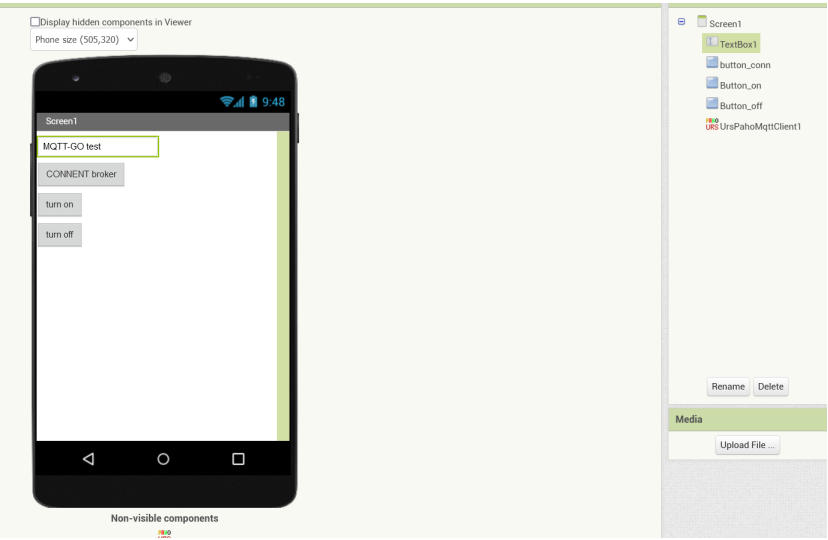
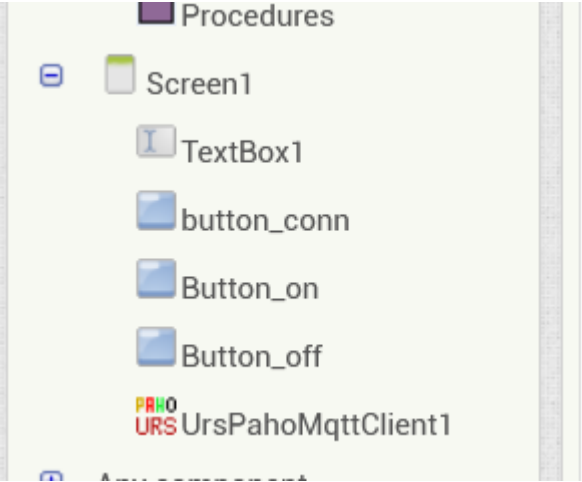
下載後解壓，再由App inventor2 點擊Extension>
Import extension 匯入”
de.ullisroboterseite.ursai2pahomqtt.aix”



參考資料

<https://kittenbothk.readthedocs.io/en/latest/Wifibrick/AI2/ai2.html>

App inventor2 APP 開發
設置以下元件



TextBox1
button_conn
Button_on
Button_off
UrsPahoMqttClient1

RenameDelete

Media
Upload File ...

(UrsPahoMqttClient)
▼ Unspecified
Broker ?
broker.mqttgo.io
ClientCertFile ?
None...
ClientID ?
MQTTGO-519981123456
ClientKeyFile ?
None...
ClientKeyPassword ?

ClientKeystoreFile ?
None...
ClientKeystorePassword ?

ClientPemFormatted ?
☐
ConnectionTimeout ?
30
KeepAlive ?
60
MaxInflight ?
10
Port ?
1883
Protocol ?
TCP
TimeToWait ?
-1
TrustedCertFile ?
None...
TruststoreFile ?
None...
TruststorePassword ?

Protocol ?
TCP
TimeToWait ?
-1
TrustedCertFile ?
None...
TruststoreFile ?
None...
TruststorePassword ?

Username ?

UserPassword ?

```

when button_conn.Click
do
  call UrsPahoMqttClient1.Connect
  CleanSession true

```

```

when UrsPahoMqttClient1.ConnectionStateChanged
  NewState StateString
do
  set button_conn.Text to join "Connect = " UrsPahoMqttClient1.IsConnected

```

```

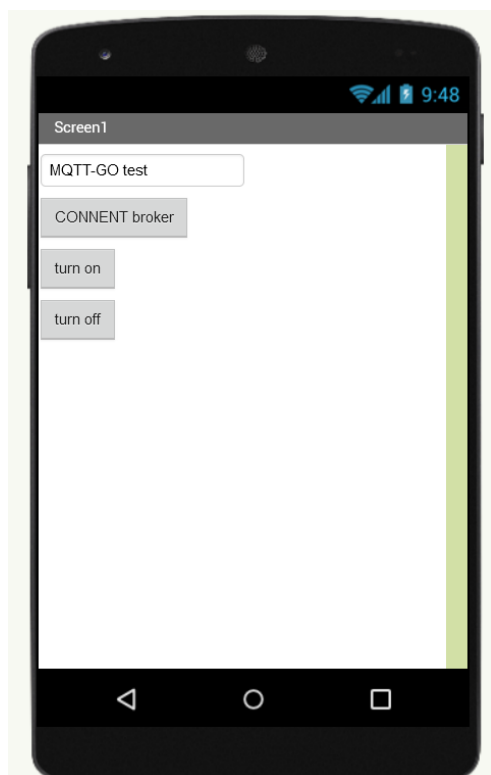
when Button_on.Click
do
  call UrsPahoMqttClient1.Publish
    Topic "Pin/relay01"
    Message "on"

```

```

when Button_off.Click
do
  call UrsPahoMqttClient1.Publish
    Topic "Pin/relay01"
    Message "off"

```



上傳到手機後，點擊**CONNENT broker**連接MQTT GO，出現"**Connent = true**"表示連接成功。連線成功後，分別點擊**trun on / trun off** 可以在MQTT GO 網站的**Subscriptions/訊息**中得到人**on/off**，同時板子上的繼電器也可以被控制吸合和斷開。

請把全組每個所完成的畫面以及成果說明，做成一份文件上傳

MQTT的實作		分組作業		
☐ 截止: 2023.11.22 17:10	3.0%	範圍: 全部小組	10 / 5 / 5 / 0	更多
章節 / 單元: 第十一週		分組方案: IOT分組		
場域主題設計		分組作業		
☐ 截止: 2023.11.30 16:16	2.0%	範圍: 全部小組	10 / 6 / 4 / 0	更多
章節 / 單元: 第十二週		分組方案: IOT分組		

11/29 布袋戲館場域主題設計

1. 場域規劃:與期中報告相同
2. 燈光設計:靠近感應、動態燈光、顏色變換
(雲林虎尾鎮/布袋戲館一樓/燈組 1)
3. 聲音:按鈕觸發、台詞設計
(雲林虎尾鎮/布袋戲館一樓/按鈕 1)

雲林故事館/室外/QR-code/前院/導覽介紹
雲林故事館/室外/QR-code/後院/導覽介紹
雲林故事館/室外/QR-code/咱兜/導覽介紹
雲林故事館/QR-code/館內人數
雲林故事館/室內/QR-code/起居空間/導覽介紹
雲林故事館/室內/QR-code/服務空間/導覽介紹
雲林故事館/室內/QR-code/通行空間/導覽介紹
雲林故事館/室內/QR-code/書齋/展覽介紹

雲林記憶cool 二館/設置可由app遠端操控的照明燈與裝飾燈條/由MQTT GO下指令，以讓整個讓電燈運作的程式開始運作/會用到副程式

虎尾驛的售票處，售票數量
虎尾驛的售票處，買票中/on
虎尾驛的售票處，售完/off
虎尾驛的售票處，共有多少人買
虎尾驛的售票處 電燈/on
虎尾驛的售票處 電燈/off
MQTT設計燈光控制

請依照您期中考的場域，進行小組討論，您們想為場域提供那些服務，MQTT主題如何設計？

以我們上課教室為例：

虎科大校園/機械工程館/4F/AME0406/電腦數量
虎科大校園/機械工程館/4F/AME0406/電腦/01/on
虎科大校園/機械工程館/4F/AME0406/電腦01/off
虎科大校園/機械工程館/4F/AME0406/出席人數
虎科大校園/機械工程館/4F/AME0406/電燈數量
虎科大校園/機械工程館/4F/AME0406/電燈/01/on
虎科大校園/機械工程館/4F/AME0406/電燈/01/off



檔名規則:

組別_主題

內容要素:

組別, 組員姓名, 班別, 日期, 主題, 說明