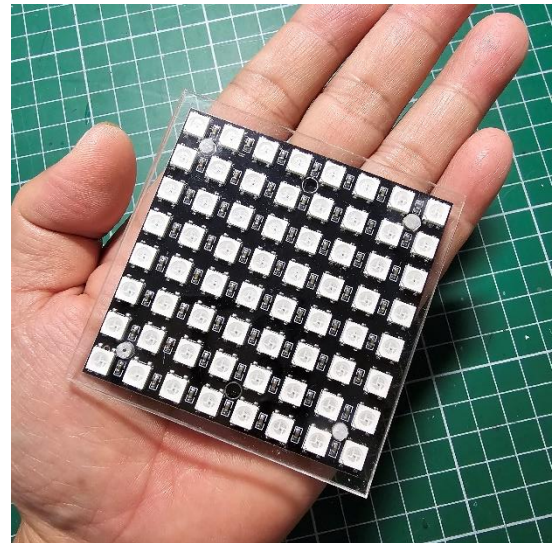
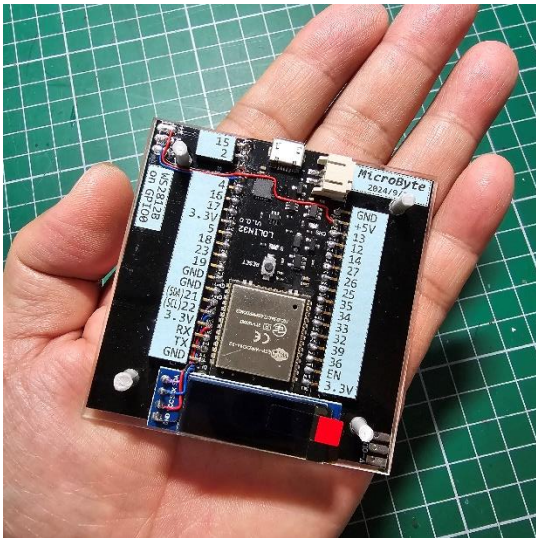
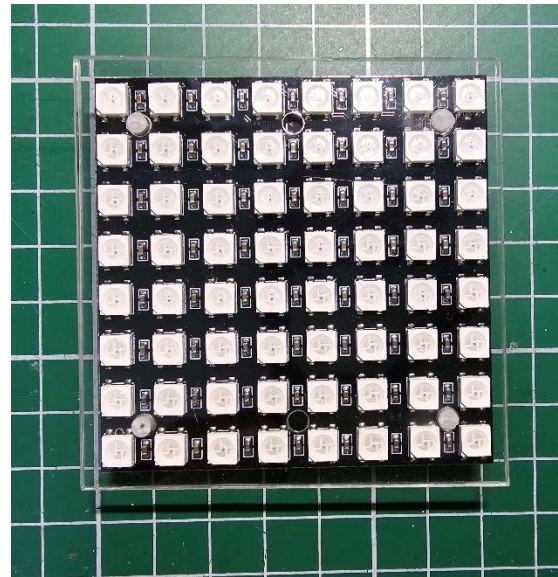
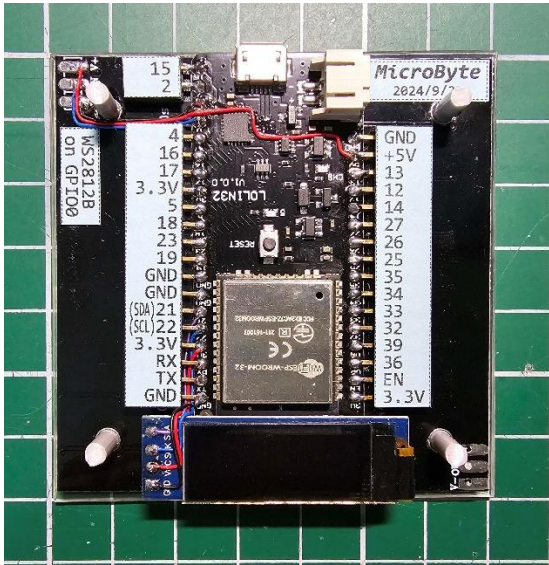


# 我的 MicroByte

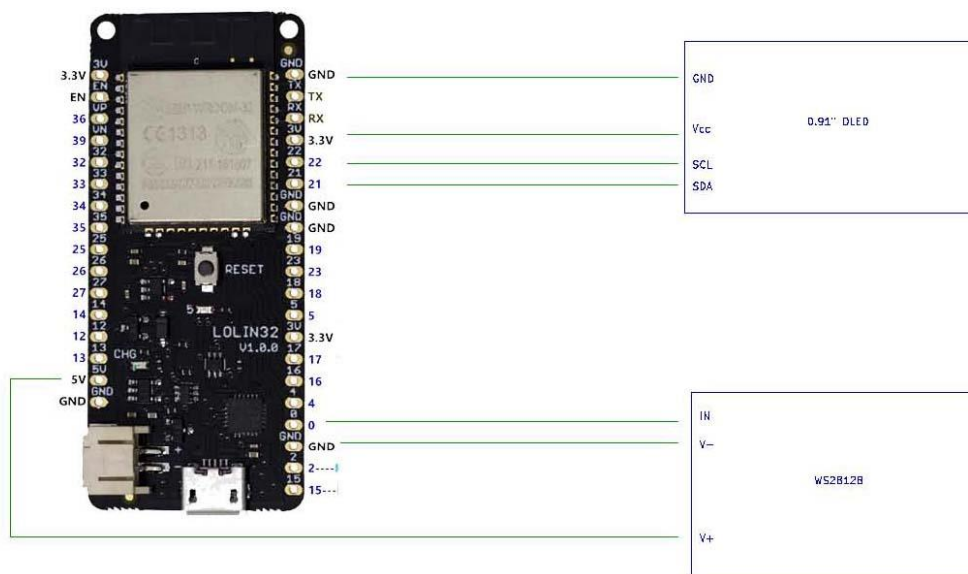


大家都知道 MicroBit，可以用積木式程式簡單快速地做一些好玩的東西，背面還附了 5\*5 的 LED，方便顯示一些小圖案。

我是沒有去好好玩它啦，但我手邊有 WS2812B，8\*8 的全彩 LED 矩陣，又有 ESP32，把它們貼到一塊兒也差不多像一個大一點的 MicroBit 了，不過還缺了三軸感應等一些 MicroBit 有的功能，不過我有無線功能呀，還加上一個 0.91 吋的 OLED 顯示，免強可用啦，因為體積較大，剛好又是 8\*8 的全彩矩陣，就叫它 MicroByte 吧~。

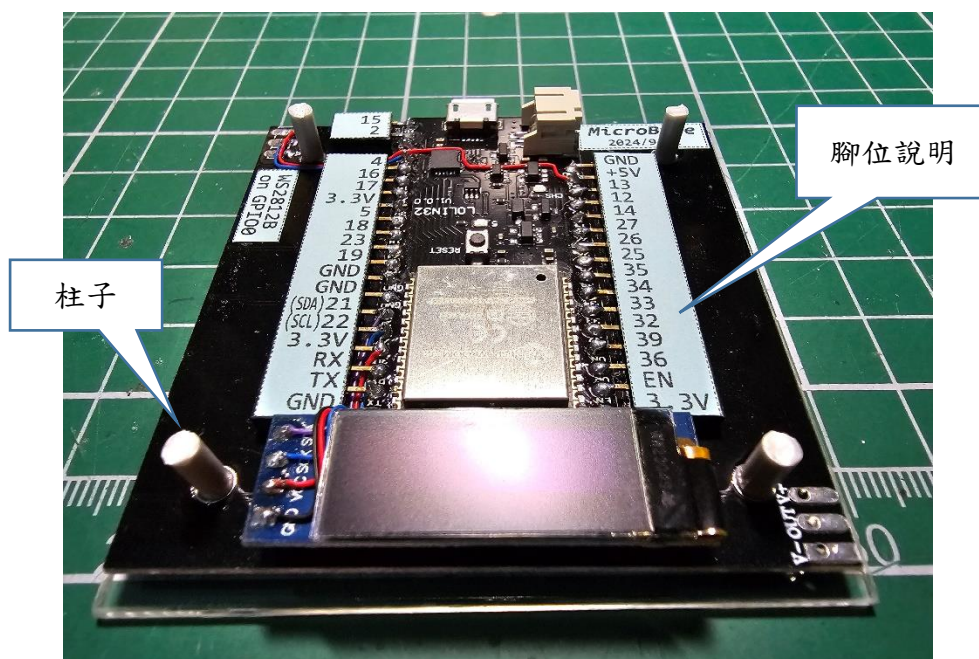
(其實市面上已經有人將 ESP32 背面結合全彩 LED 矩陣的產品了，不過手做 DIY 還是比較有樂趣啦!)

線路圖如下，其實只是經由 IIC 介面(SCL、SDA)接上 OLED，用 GPIO0 去控制 WS2812，其它腳都用排針座接出來，方便將來還可以作其它實驗。



除了接線，其它元件都用快乾黏在 WS2812 背面就行了，我在 WS2812 上加了一片透明壓克力，避免放在桌上磨來磨去磨到燈珠。我使用做模型剩下的邊角料，粗細剛好可以穿過 WS2812 預留的孔，將 WS2812 與壓克力黏在一起，而它的長度裁剪成剛好超過 ESP32 和 OLED 的高度，當”柱子”用，當這面朝下時可以保持 ESP32 及 OLED 不直接碰到桌面。

貼在排針座上的腳位說明標示用 Word 編排好印出來再剪下來用膠水貼上就行了，我是用白膠，感覺牢固一點。Word 是所視及所得的，所以量一下排針的長寬，或你想要的長寬，在 Word 裡用文字方塊控制範圍就行了，印出來的大小是非常準確的。



以下是簡單的 WS2812B 及 OLED 測試程式，使用 Arduino IDE 環境。必須使用”程式庫管理員”先安裝 Adafruit\_SSD1306 給 OLED，安裝 FastLED 給 WS2812。



```
/*-----  
  測試 OLED  
  連上 NTP 伺服器，取得正確時間，存入 ESP32 內部 RTC，  
  並將現在時間顯示在 OLED 上  
-----*/  
  
#include <SPI.h>  
#include <Wire.h>  
#include <WiFi.h>  
#include <time.h>  
#include <Adafruit_GFX.h>  
#include <Adafruit_SSD1306.h>  
  
#define SCREEN_WIDTH 128 // OLED display width, in pixels  
#define SCREEN_HEIGHT 32 // OLED display height, in pixels  
  
// Declaration for an SSD1306 display connected to I2C (SDA, SCL pins)
```

```

#define OLED_RESET    -1 // Reset pin # (or -1 if sharing Arduino reset pin)
Adafruit_SSD1306 display(SCREEN_WIDTH, SCREEN_HEIGHT, &Wire, OLED_RESET);

const char* ssid      = "Your_WiFi_AP_name";
const char* password = "password_here";

const char* ntpServer = "pool.ntp.org";
const uint16_t utcOffest = 28800;    // UTC+8 偏移量
const uint8_t daylightOffset = 0;    // 夏令時間
String time_msg;    // 用來儲存時間字串
struct tm now;    // 宣告「分解時間」結構變數

void setup() {
  Serial.begin(9600);

  // SSD1306_SWITCHCAPVCC = generate display voltage from 3.3V internally
  if(!display.begin(SSD1306_SWITCHCAPVCC, 0x3C)) { // Address 0x3C for 128x32
    Serial.println(F("SSD1306 allocation failed"));
    for(;;); // Don't proceed, loop forever
  }
  // Clear the buffer
  display.clearDisplay();
  display.cp437(true);    // Use full 256 char 'Code Page 437' font

  WiFi.mode(WIFI_STA);
  WiFi.begin(ssid, password);
  Serial.println("");
  while (WiFi.status() != WL_CONNECTED) {
    Serial.print(".");
    delay(500);
  }
  Serial.print("IP 位址：");
  Serial.println(WiFi.localIP());

  //取得 NTP 伺服器的網路時間，存入內部 RTC
  configTime(utcOffest, daylightOffset, ntpServer);
  delay(1000);
}

void loop() {
  if(!getLocalTime(&now)){    // 取得本地 RTC 內時間
    Serial.println("無法取得 RTC 時間～");
    display.setCursor(0, 16);    // Start at top-left corner

```

```

    display.println(F("RTC Error!")); // Show RTC Error message
    display.display();
    digitalWrite(LED_BUILTIN, LOW);
    return;
}else{
    digitalWrite(LED_BUILTIN, HIGH);
}
Serial.println(&now, "%Y/%m/%d %H:%M:%S"); // 輸出格式化時間字串到串列埠
//每一小時取得 NTP 伺服器的網路時間(校正時間)，存入內部 RTC
if(now.tm_min == 0 && now.tm_sec == 0){
    configTime(utcOffset, daylightOffset, ntpServer);
    Serial.println("NTP Update.");
}

display.clearDisplay();
display.setTextSize(1); // Normal 1:1 pixel scale
display.setTextColor(SSD1306_WHITE); // Draw white text
display.setCursor(0, 0); // Start at top-left corner
display.print("My IP: ");
display.println(WiFi.localIP());

display.setTextColor(SSD1306_WHITE);
//display.setTextColor(SSD1306_BLACK, SSD1306_WHITE); //以反白方式顯示
display.setTextSize(2); // Draw 2X-scale text
display.setCursor(15, 16);
time_msg = (String)now.tm_hour + ":" + (String)now.tm_min + ":" +
(String)now.tm_sec;
display.print(time_msg);
display.display();
delay(1000);
}

```

```

/*-----
    測試 WS2812B
    顯示各點各色是否正常
    顯示順序：
        依序點亮紅色-->一半邊紅色增強到最亮再減弱
        依序點亮綠色-->一半邊綠色增強到最亮再減弱
        依序點亮藍色-->一半邊藍色增強到最亮再減弱
        顯示七彩，一色一排
        隨機一個燈珠顯示隨機顏色及亮度，隨機熄滅兩個燈珠
        顯示七彩，一色一排
        重複執行以上
    -----*/

#include "FastLED.h"
#define uchar unsigned char
#define RGB_PIN 0          /*RGB 燈使用 GPIO0 進行控制*/
#define NUM_LEDS 64        /*RGB 燈是串聯的，RGB 燈的數量*/
CRGB leds[NUM_LEDS];      /*CRGB 是結構類型*/

/*
 *功能:RGB 燈初始化
 *參數:無
 *返回值:無
 */
void RGB_Init(void)
{
    FastLED.addLeds<WS2812, RGB_PIN, GRB>(leds, NUM_LEDS);
}

/*
 *功能：RGB 燈控制
 *參數：
    creset：燈號，控制哪個 LED 燈
    red:三原色中的紅色(0-255)
    green:三原色中的綠色(0-255)
    blue:三原色中的藍色(0-255)
 *返回值:無
 */
void RGB_Control(uchar creset, uchar red, uchar green, uchar blue)
{
    leds[creset] = CRGB(red, green, blue);
    FastLED.show();
}

```

```
void Rainbow(void)    //顯示七彩色，一種顏色一行
{
    //---RAINBOW-----
    for(int k=0; k<8; k++)
        RGB_Control(k, 20, 0, 0);    //紅
    delay(1000);
    for(int k=8; k<16; k++)
        RGB_Control(k, 20, 6, 0);    //橙
    delay(1000);
    for(int k=16; k<24; k++)
        RGB_Control(k, 20, 20, 0);    //黃
    delay(1000);
    for(int k=24; k<32; k++)
        RGB_Control(k, 0, 20, 0);    //綠
    delay(1000);
    for(int k=32; k<40; k++)
        RGB_Control(k, 2, 14, 20);    //淺藍
    delay(1000);
    for(int k=40; k<48; k++ )
        RGB_Control(k, 0, 0, 25);    //藍
    delay(1000);
    for(int k=48; k<56; k++)
        RGB_Control(k, 4, 1, 20);    //靛
    delay(1000);
    for(int k=56; k<64; k++)
        RGB_Control(k, 10, 0, 14);    //紫
}

void AllOff(void)    //所有燈珠熄滅
{
    for(int i=0; i<64; i++)
        RGB_Control(i, 0, 0, 0);    //燈全滅
}

void setup()
{
    Serial.begin(115200);    /*設置串列傳輸速率*/
    Serial.print("hello world! RGB Test\r\n");
    RGB_Init();    /*RGB 燈初始化*/
}

void loop()
{

```

```
//---RED-----
for(int i=0; i<64; i++)    //依序點亮紅色
{
    RGB_Control(i, 10, 0, 0);
    delay(50);
}
for(int i=1; i<=255; i+=10)    //前半紅色漸亮，測試亮度變化

{
    for(int j=0; j<32; j++)
        RGB_Control(j, i, 0, 0);
    delay(5);
}
for(int i=255; i>1; i-=10)    //前半紅色漸暗，測試亮度變化
{
    for(int j=0; j<32; j++)
        RGB_Control(j, i, 0, 0);
    delay(5);
}
AllOff();
delay(100);
//---GREEN-----
for(int i=0; i<64; i++)    //依序點亮綠色
{
    RGB_Control(i, 0, 10, 0);
    delay(50);
}
for(int i=1; i<=255; i+=10)    //後半綠色漸亮，測試亮度變化
{
    for(int j=32; j<64; j++)
        RGB_Control(j, 0, i, 0);
    delay(5);
}
for(int i=255; i>1; i-=10)    //後半綠色漸暗，測試亮度變化
{
    for(int j=32; j<64; j++)
        RGB_Control(j, 0, i, 0);
    delay(5);
}
AllOff();
delay(100);
//---BLUE-----
for(int i=0; i<64; i++)    //依序點亮藍色
```

```

{
    RGB_Control(i, 0, 0, 10);
    delay(50);
}
for(int i=1; i<=255; i+=10)    //兩側藍色漸亮，測試亮度變化
{
    for(int j=0; j<16; j++)
    {
        RGB_Control(j, 0, 0, i);
        RGB_Control(j+48, 0, 0, i);
    }
    delay(5);
}
for(int i=255; i>1; i-=10)    //兩側藍色漸暗，測試亮度變化
{
    for(int j=0; j<16; j++)
    {
        RGB_Control(j, 0, 0, i);
        RGB_Control(j+48, 0, 0, i);
    }
    delay(5);
}
Alloff();
delay(1000);

//-----Rainbow-----
Rainbow();    //顯示七彩
delay(5000);
Alloff();
delay(1000);

//-----Random color and position-----
for(int i=0; i<=1000; i++)    //隨機閃爍一段時間
{
    RGB_Control(random(64), random(30), random(30), random(30));
    RGB_Control(random(64), 0,0,0);
    RGB_Control(random(64), random(30), random(30), random(30));
    RGB_Control(random(64), 0,0,0);
    delay(150);
    RGB_Control(random(64), 0,0,0);
}
Alloff();
delay(3000);

```

```
//-----Rainbow-----  
Rainbow();    //顯示七彩  
delay(5000);  
AllOff();  
delay(1000);  
}
```