

Name Badge (使用 ESP32 WITH TTGO 開發板)

使用基於 ESP32 的 TTGO-T-Display 開發板建構的簡單名稱徽章。因同事要離職了.. 想送點特別的禮物祝福他,想了很久~ 突然看到名稱徽章覺得好特別 ^^ 就決定送一個客製化電子名片可顯示基本個人資訊、臉書 QRCode 及 LINE QRCode 加入好友,在這裡非常謝謝 Oxinon 專案。該靈感來自於 oxinon 的 github 名稱徽章專案
<https://github.com/oxinon/TTGO-T-Display-Name-Badge>。



LILYGO TTGO T-Display ESP32 可在露天拍賣或其它蝦皮網拍應可買得到
<https://goods.ruten.com.tw/item/show?21931296978153>

安裝 Arduino IDE 開發環境

Arduino IDE 在 1.6.5 版之後，支援第三方的硬體，因此我們可以在 Arduino IDE 上開發 Ameba，並共享 Arduino 上面的範例程式。在 Arduino 官方網站上可以找到下載程式
<https://www.arduino.cc/en/Main/Software>。目前最新 Arduino IDE 版本為 1.8.12。

← → ↻ [arduino.cc/en/Main/Software](https://www.arduino.cc/en/Main/Software) ⓘ ☆

∞ STORE SOFTWARE **EDU** PRO RESOURCES COMMUNITY HELP 🔍 📁 SIGN IN

Download the Arduino IDE



ARDUINO 1.8.12
The open-source Arduino Software (IDE) makes it easy to write code and upload it to the board. It runs on Windows, Mac OS X, and Linux. The environment is written in Java and based on Processing and other open-source software.
This software can be used with any Arduino board. Refer to the [Getting Started](#) page for installation instructions.

Windows Installer, for Windows XP and up
Windows ZIP file for non-admin install

Windows app Requires Win 8.1 or 10
[Get](#)

Mac OS X 10.8 Mountain Lion or newer

Linux 32 bits
Linux 64 bits
Linux ARM 32 bits
Linux ARM 64 bits

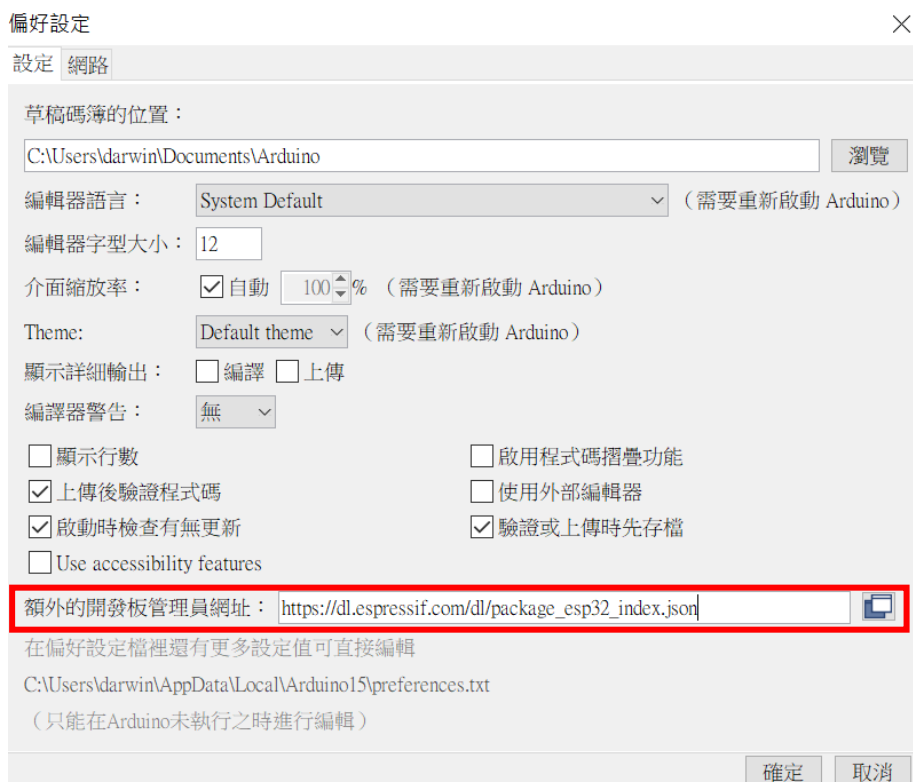
[Release Notes](#)
[Source Code](#)
[Checksums \(sha512\)](#)

- 步驟 1: 安裝後,打開 Arduino IDE,為了讓 Arduino IDE 找到 ESP32 的設定檔，先到【檔案】→【偏好設定】。



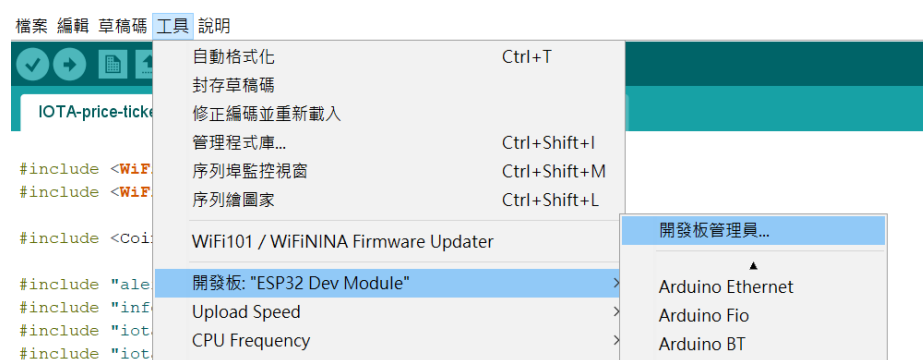
- 步驟 2: 然後在【額外的開發板管理員網址】：填入如下網址。

https://dl.espressif.com/dl/package_esp32_index.json

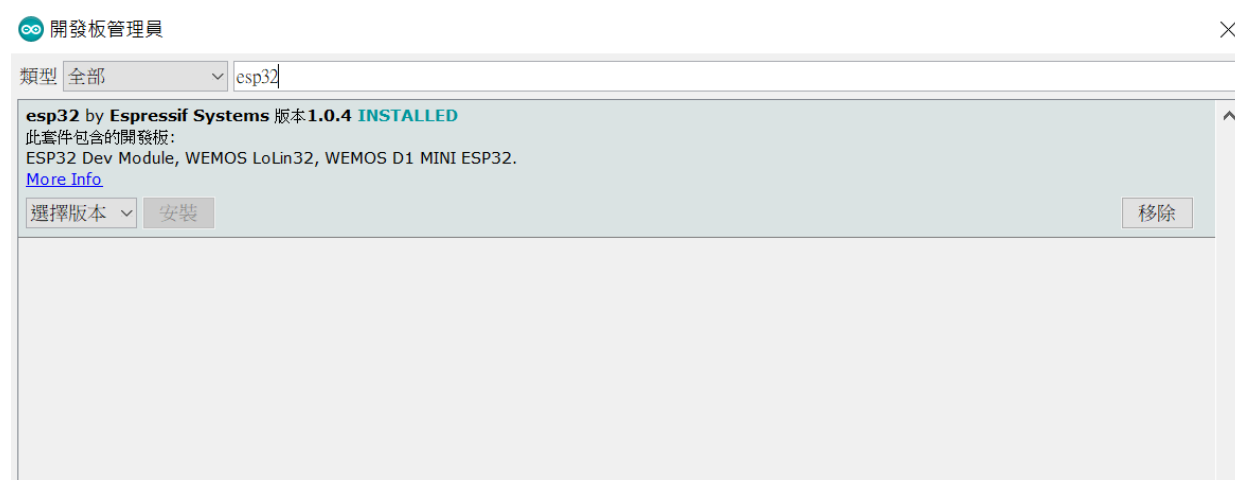


填完之後按 OK，我們將 Arduino IDE 關掉之後重開。

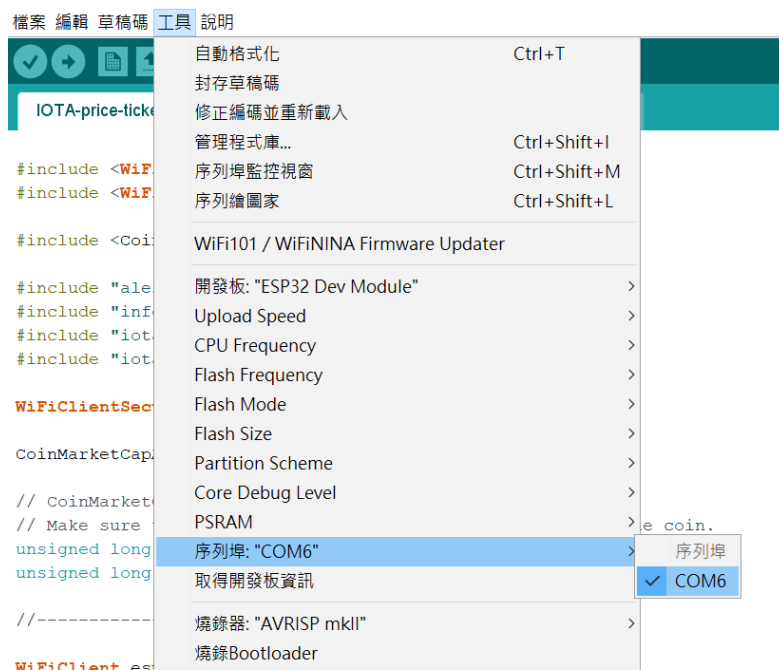
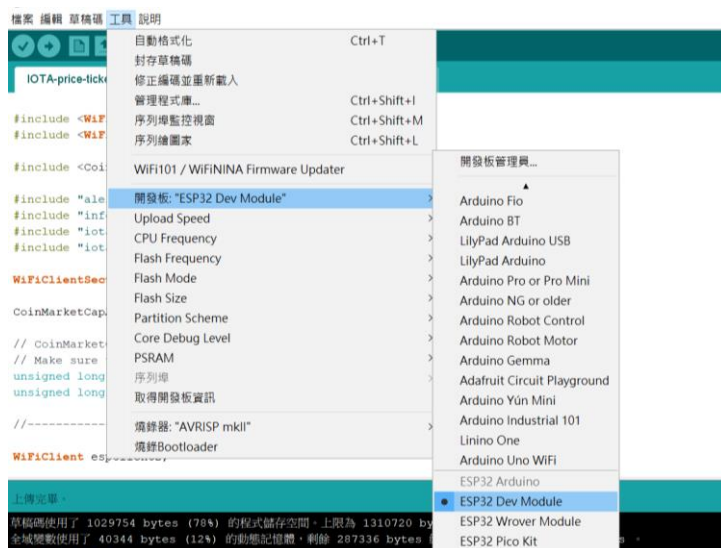
接著到【工具】→【開發板】→【開發板管理員】如下圖所示。



在【開發板管理員】裡，它需要約十幾秒鐘整理所有硬體檔案，如果網路狀況不好可能會等上數分鐘。每當有新的硬體設定，我們需要重開【開發板管理員】，所以我們等一會兒之後，關掉【開發板管理員】，然後再打開它，將捲軸往下拉找到【esp32】，點右邊的 Install，這時候 Arduino IDE 就根據 esp32 的設定檔開始下載 esp32 所需要的檔案，安裝成功後，關閉開發板管理員視窗。



- 步驟 3: 接著將板子選成 ESP32，選取【工具】→【開發板】→【ESP32 Dev Module】並正確選擇對應的序列埠。



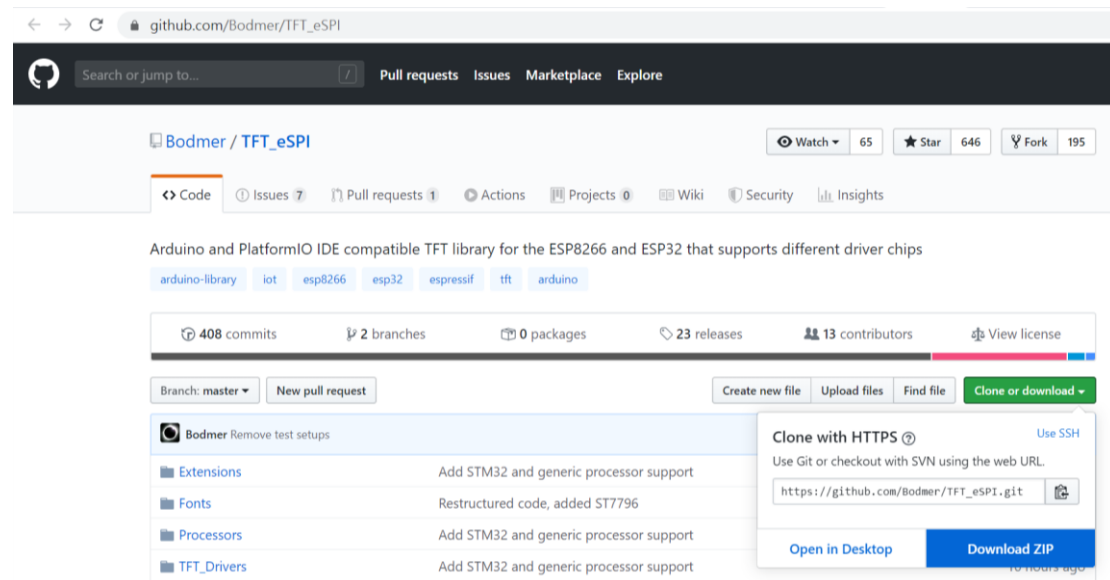
這樣開發環境就設定完成了。

小提醒: COM6 序號埠是筆者目前的,你們可能和我不一樣,可至裝置管理員→通用序列埠匯流排控制器去查看哦~

參考: <https://randomnerdtutorials.com/installing-the-esp32-board-in-arduino-ide-windows-instructions/>

安裝相關 library

- 步驟 1: 下載 TFT_eSPI 並解壓縮檔案，並將 TFT_eSPI 裡整個 lib 目錄的檔案到【我的文件\Arduino\library】目錄裡。 https://github.com/Bodmer/TFT_eSPI



- 步驟 2: 編輯 User_Setup_Select.h 檔案。

在第 22 行及 52 行將註解取消，如下所示

```
#include <User_Setup.h>           // Default setup is root library folder
#include <User_Setups/Setup25_TTGO_T_Display.h> // Setup file for ESP32 and TTGO T-Display ST7789V SPI
bus TFT
```

最後存檔離開。

- 步驟 3: 編輯 User_Setup.h 檔案。

在第 22 行及 132 行~134 行將註解，如下所示

```
///
```

在第 37 將註解取消，如下所示

```
#define ST7789_DRIVER // Full configuration option, define additional parameters below for this display
```

在第 167 將加入，如下所示

```
// For ESP32 Dev board (only tested with ILI9341 display)
```

// The hardware SPI can be mapped to any pins

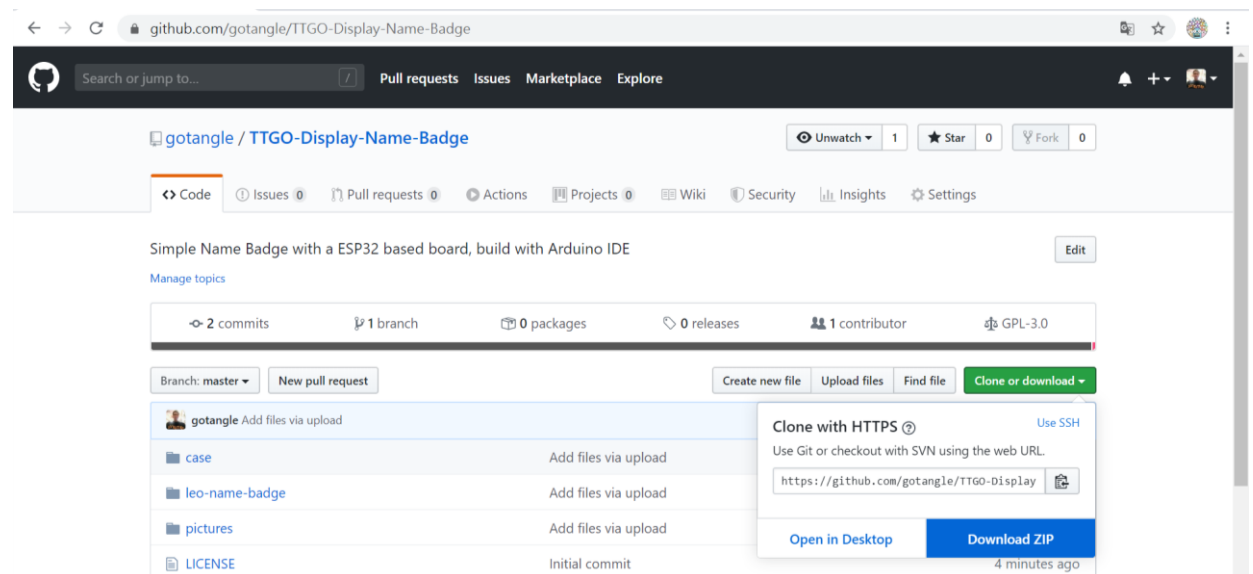
```
#define TFT_MOSI      19
#define TFT_SCLK      18
#define TFT_CS        5
#define TFT_DC        16
#define TFT_RST       23
#define TFT_BL        4 // Display backlight control pin
```

最後存檔離開。

- 步驟 4: 下載 Button2 並解壓縮檔案並將裡整個 lib 目錄的檔案到【我的文件\Arduino\library】目錄裡。<https://github.com/LennartHennigs/Button2>。

參考: <https://github.com/Xinyuan-LilyGO/TTGO-T-Display>

- 步驟 5: 下載【TTGO-Display-Name-Badge】專案並解壓縮檔案。
<https://github.com/gotangle/TTGO-Display-Name-Badge>。



- 步驟 6: 準備圖檔。

解壓縮後 TTGO-Display-Name-Badge-master\pictures 目錄,這裡共有 4 張圖檔,有開機畫面用的、主畫面 Leo 名稱徽章、以及 FB/LINE QRCode 圖檔, 這裡可依自己需求自己設計更改, 這裡以開機畫面為例(自行設計)。

圖片大小:240x135px 副檔:png

轉換圖檔:

此時必須將做好的圖檔轉換成 c 檔案格式,可透過線上圖檔做轉換。

http://www.rinkydinkelectronics.com/t_imageconverter565.php

上傳圖檔:

Supported fileformats: .png, .jpg and .gif
Maximum filesize: 300KB

Picture to convert: gamania.png

Convert to: ☒ .c file ☐ .raw file

下載 c 檔案:

ImageConverter (UTFT)

The picture 'gamania.png' is being processed...



[Click here to download your file](#)

可將下載的 c 檔案將副檔名改成.h 標頭檔案, 並編輯 h 標頭檔案,例如 gamania.h

新增內容如下

```
#include <pgmspace.h>

// Icon width and height

const uint16_t gamaniaWidth = 240;

const uint16_t gamaniaHeight = 135;
```

刪除如下:

```
#if defined(__AVR__)

    #include <avr/pgmspace.h>

#elif defined(__PIC32MX__)

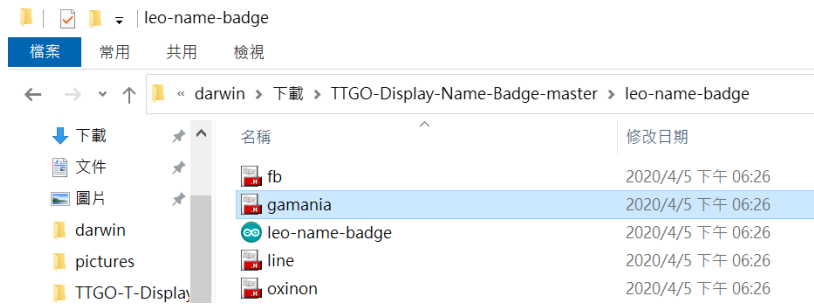
    #define PROGMEM

#elif defined(__arm__)

    #define PROGMEM

#endif
```

最後將所有圖檔轉換好的.h 標頭檔準備好後和主程式一起放在一起,要開始來編譯主程式及這些標頭檔案。



■ 步驟 7: 開啟主程式【leo-name-badge.ino】專案。



■ 步驟 8: 點選左上角【→】編譯/上傳按鈕，上傳程式至開發板。



編譯/上傳成功!

```
上傳完畢  
Writing at 0x00008000... (100 %)  
Wrote 3072 bytes (128 compressed) at 0x00008000 in 0.0 seconds (effective 4096.0 kbit/s)...  
Hash of data verified.  
Leaving...  
Hard resetting via RTS pin...
```

顯示結果:



長按【A 按鈕】約 2 秒會顯示臉書 QRCode 再按一秒會顯示 LINE QRCode。依據 QRCode 顯示讓新朋友加入自己。長按【B 按鈕】則回到主畫面 Leo 名稱徽章。



長按【B 按鈕】約 2 秒會顯示電力狀態,如果有安裝鋰電池可隨時監控電力狀態。



參考資料來源: <https://github.com/oxinon/TTGO-T-Display-Name-Badge>