

# IOTA Price Monitor (使用 ESP32 WITH TTGO 開發板)

使用基於 ESP32 的 TTGO-T-Display 和 CoinMarketcap API 的簡單 IOTA 價格程式碼來實現 Arduino IDE 平台 IOTA 價格 monitor 。 Youtube: <https://youtu.be/vdw2eKsP-xM>



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 1.8.10 。



### Download the Arduino IDE



**ARDUINO 1.8.10**  
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

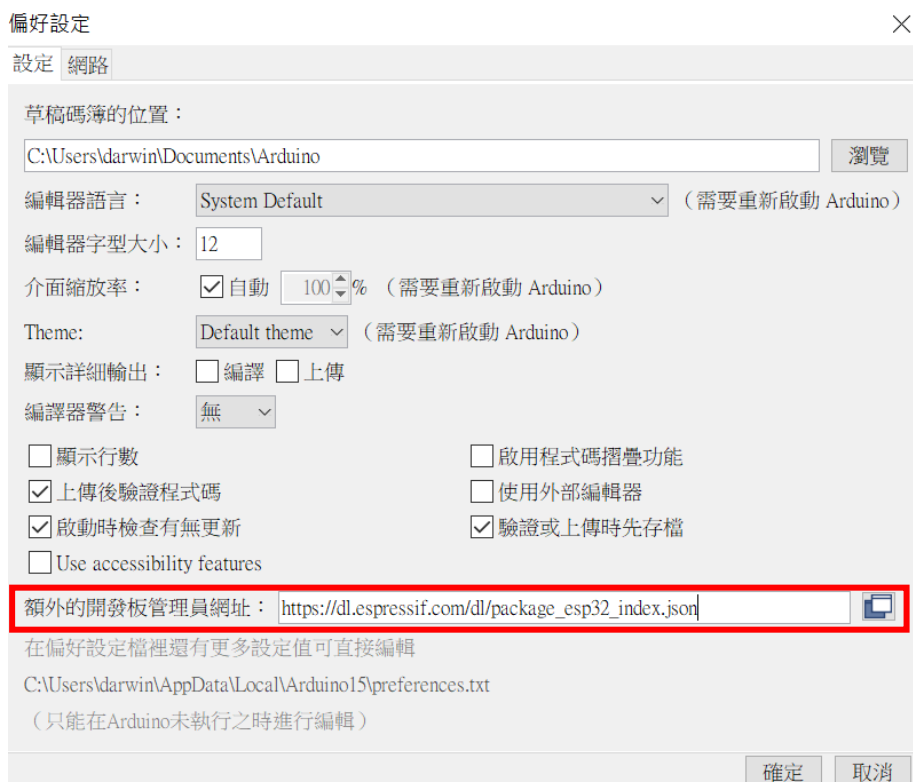
[Release Notes](#)  
[Source Code](#)  
[Checksums \(sha512\)](#)

- 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](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。

IoT-price-ticker-V2-TTGO-T-Display - IOTA-price-ticker-v2-ttgo-t-display.ino | Arduino 1.8.10

檔案 編輯 草稿碼 工具 說明



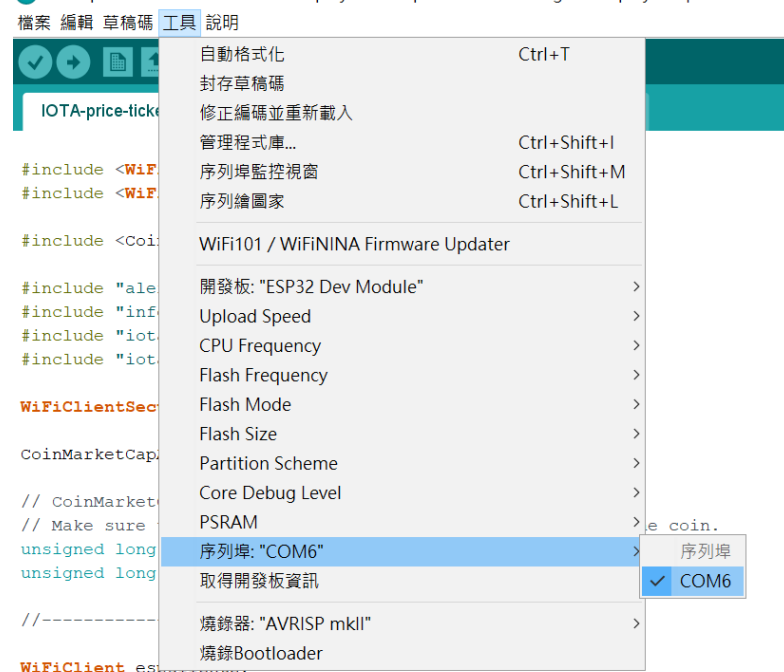
在【開發板管理員】裡，它需要約十幾秒鐘整理所有硬體檔案，如果網路狀況不好可能會等上數分鐘。每當有新的硬體設定，我們需要重開【開發板管理員】，所以我們等一會兒之後，關掉【開發板管理員】，然後再打開它，將捲軸往下拉找到【esp32】，點右邊的 Install，這時候 Arduino IDE 就根據 esp32 的設定檔開始下載 esp32 所需要的檔案，安裝成功後，關閉開發板管理員視窗。



- 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。



IOTA-price-ticker-v2-TTGO-T-Display - IOTA-price-ticker-v2-ttgo-t-display.ino | Arduino 1.8.



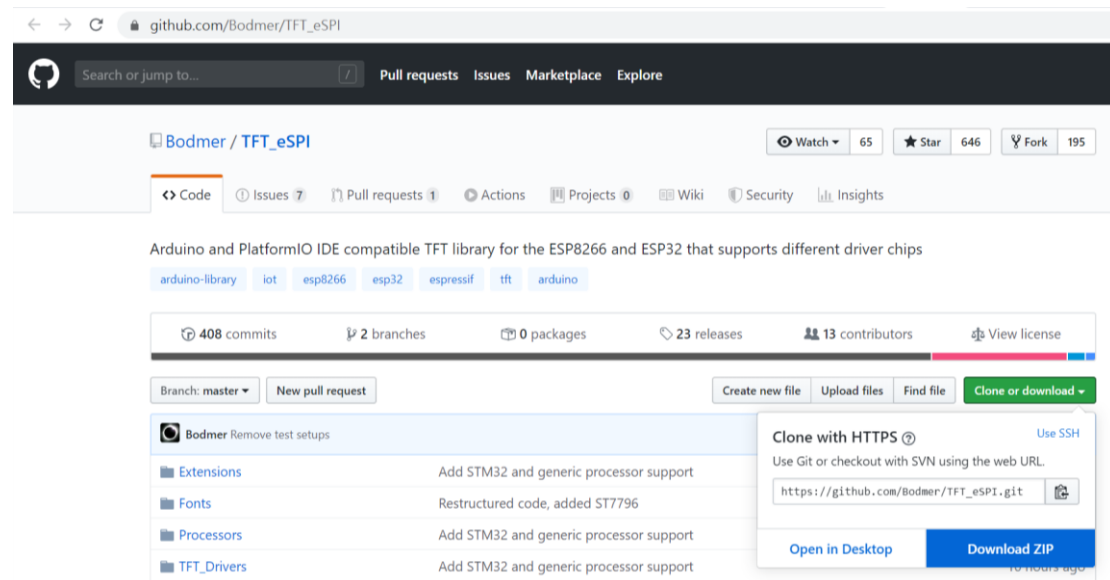
這樣開發環境就設定完成了。

小提醒: COM6 序號埠是筆者目前的,你們可能和我不一樣,可至裝置管理員裡面→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](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

```
///define ILI9341_DRIVER
///define TFT_CS   PIN_D8 // Chip select control pin D8
///define TFT_DC   PIN_D3 // Data Command control pin
///define TFT_RST  PIN_D4 // Reset pin (could connect to NodeMCU RST, see next line)
```

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。

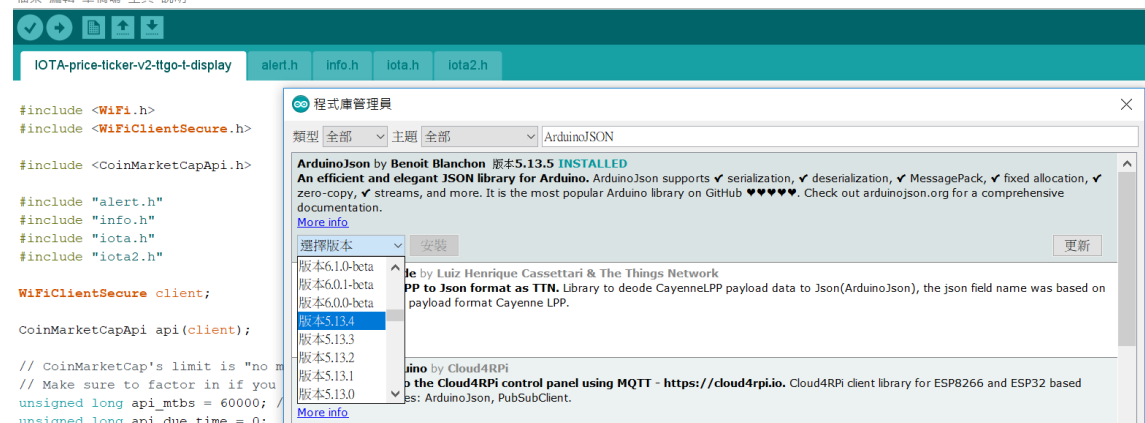
- 步驟 4: 下載 Button2 並解壓縮檔案並將裡整個 lib 目錄的檔案到【我的文件\Arduino\library】目錄裡。<https://github.com/LennartHennigs/Button2>。

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

- 步驟 5: 下載 arduino-coinmarketcap-api 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/witnessmenow/arduino-coinmarketcap-api>。

- 步驟 6: 開啟【管理程式庫...】安裝 ArduinoJSON version 5.13.5 版本(6 版本有問題)。

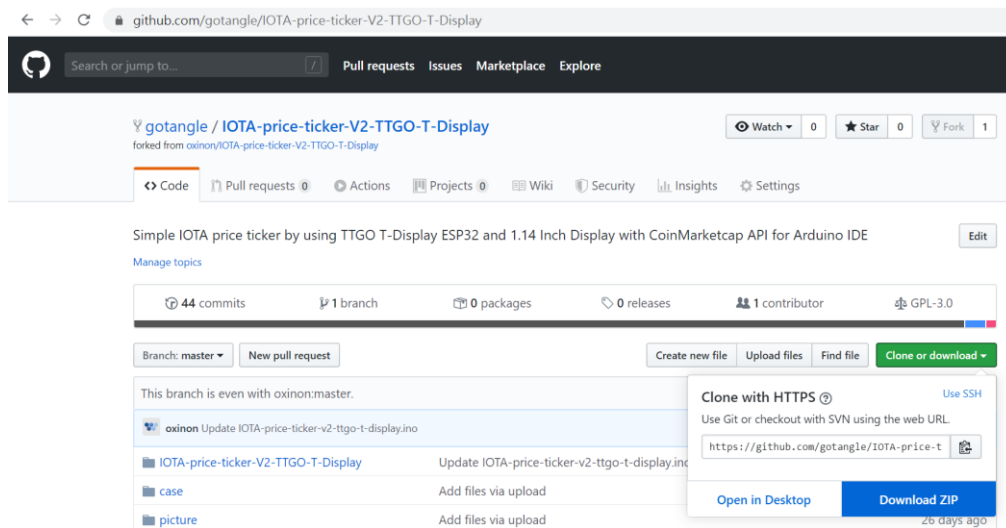




檢查【我的文件\Arduino\library】目錄裡有 4 個目錄; TFT\_eSPI 目錄/ Button2 目錄/ ArduinoJson 目錄和 arduino-coinmarketcap-api-master 目錄。

← → ↕ ↗ 本機 > 文件 > Arduino > libraries			
	名稱	修改日期	類型
★ 快速存取 Desktop 下載 文件 圖片 IOTA-price-ticker-V2-TTGO-T libraries picture 新增資料夾 OneDrive 本機 網路	Adafruit_GFX_Library	2020/1/7 下午 01:25	檔案資料夾
	Adafruit_Sensor-master	2020/1/7 下午 01:25	檔案資料夾
	Adafruit_SHT31-master	2020/1/7 下午 01:25	檔案資料夾
	Adafruit_SSD1306-master	2020/1/7 下午 01:25	檔案資料夾
	arduino_566205	2019/10/7 下午 07:19	檔案資料夾
	Arduino_Cryptography_Library	2020/1/7 下午 01:25	檔案資料夾
	arduino-coinmarketcap-api-master	2020/1/12 下午 03:20	檔案資料夾
	ArduinoJson	2020/1/12 下午 03:29	檔案資料夾
	Blynk	2020/1/7 下午 01:25	檔案資料夾
	Button2	2020/1/12 下午 01:39	檔案資料夾
	DHT-sensor-library-master	2019/10/6 上午 10:46	檔案資料夾
	SimpleTimer	2020/1/7 下午 01:25	檔案資料夾
	TFT_eSPI	2020/1/12 下午 04:50	檔案資料夾
	readme	2019/10/6 上午 02:23	文字文件

■ 步驟 7: 下載【IOTA-price-ticker-V2-TTGO-T-Display】專案並解壓縮檔案。  
<https://github.com/gotangle/IOTA-price-ticker-V2-TTGO-T-Display>。



## ■ 步驟 8: 開啟主程式【IOTA-price-ticker-v2-ttgo-t-display.ino】專案。

🔗 IOTA-price-ticker-V2-TTGO-T-Display - IOTA-price-ticker-v2-ttgo-t-display.ino | Arduino 1.8.10  
檔案 編輯 草稿碼 工具 說明



編輯 IOTA-price-ticker-v2-ttgo-t-display.ino 檔案。

在第 26~27 行將無線 wifi 改成你們自家連線的 SSID 及密碼, 如下所示

```
// edit this for your network
```

```
#define wifi_ssid "WIFISSID"
```

```
#define wifi_password "WIFIPASSWD"
```

## ■ 步驟 9: 點選左上角【→】編譯/上傳按鈕, 上傳程式至開發板。



IOTA-price-ticker-V2-TTGO-T-Display - IOTA-price-ticker-v2-ttgo-t-display.ino | Arduino 1.8.10

檔案 編輯 草稿碼 工具 說明

```
// Startup

tft.fillRect(0, 0, 240, 135, ST7735_BLACK);
Serial.print("Connecting to ");
tft.drawString("Connecting to ",15, 10, 2);
Serial.println(wifi_ssid);
tft.drawString(wifi_ssid,15, 25, 2);
tft.pushImage(200, 2, infoWidth, infoHeight, info);
delay(1000);
```

編譯/上傳成功!

IOTA-price-ticker-V2-TTGO-T-Display - IOTA-price-ticker-v2-ttgo-t-display.ino | Arduino 1.8.10

檔案 編輯 草稿碼 工具 說明

```
// Startup

tft.fillRect(0, 0, 240, 135, ST7735_BLACK);
Serial.print("Connecting to ");
tft.drawString("Connecting to ",15, 10, 2);
Serial.println(wifi_ssid);
tft.drawString(wifi_ssid,15, 25, 2);
tft.pushImage(200, 2, infoWidth, infoHeight, info);
delay(1000);

WiFi.begin(wifi_ssid, wifi_password);

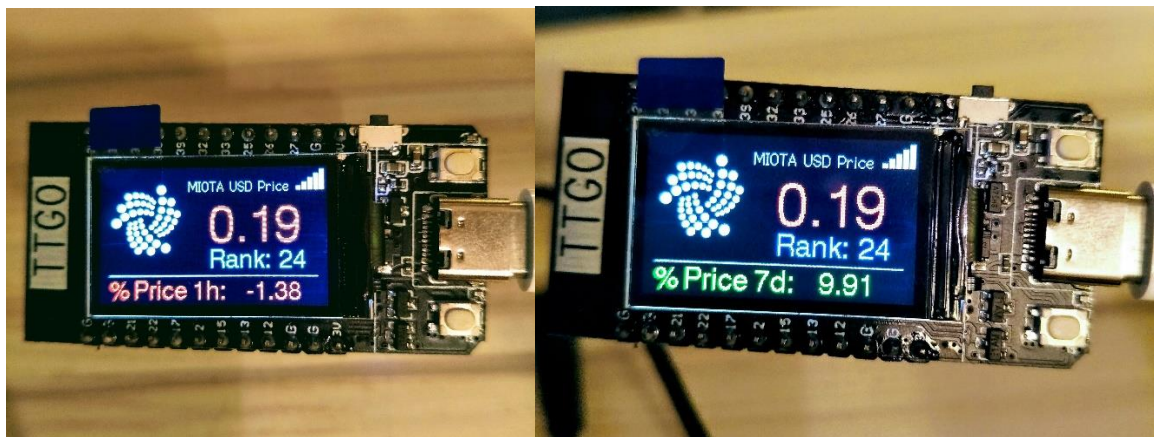
while (WiFi.status() != WL_CONNECTED) {
  delay(500);
  Serial.print(".");
}

Serial.println("");
Serial.println("WiFi connected");
tft.setTextColor(ST7735_GREEN);
tft.drawString("WiFi connected",15, 40, 2);
tft.setTextColor(ST7735_WHITE);
Serial.println("IP address: ");
Serial.println(WiFi.localIP());
```

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 RST pin...

顯示結果:



參考資料來源: <https://github.com/oxinon/IOTA-price-ticker-V2-TTGO-T-Display>