

電腦網路實驗實驗報告

< 聯網式感測網路-進行無線感測網路實驗 >

姓名：翁佳煌

學號：409430030

1. 實驗名稱

Esp32 與 OLED 顯示比特幣價錢

2. 實驗目的

使用 OLED 顯示器以及 ESP32，連接到 Wi-Fi 網絡並從 Yahoo 股市中抓取比特幣的實時價格數據並從獲取的 JSON 數據中提取比特幣的實時價格數據，然後將這些數據顯示在 OLED 顯示器上，這樣可以方便地查看比特幣價格的變化，並提供一種即時的數據展示方式。

3. 實驗設備

Windows 作業系統之電腦。

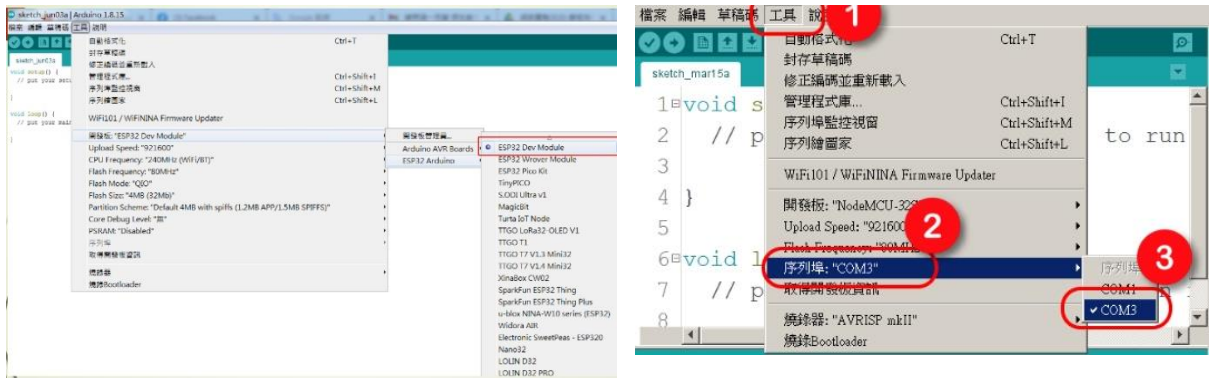
Arduino 1.8.15

Esp32、ArduinoJson、OLED 程式庫

CP210x 驅動程式

4. 實驗步驟

1. 首先先到官網安裝 Arduino 1.8.15 的版本。
2. 安裝相關函式庫，並選擇 esp32 Dev Module，以及確認裝置的連接埠，如下圖 1。



▲圖 1

3. 撰寫相關程式

首先看到下圖 2，

1~6 行:引入所需的函式庫，這些函式庫提供了與 WiFi 通訊、發起 HTTP 請求、解析 JSON 資料以及控制 OLED 顯示屏的功能。

第 11 行:程式碼初始化了一個 SSD1306 物件，用於與 OLED 顯示屏進行通訊。0x3c 是 OLED 顯示屏的 I2C 位址，21 和 22 是用於連接 OLED 顯示屏的引腳。

第 17~18 行: 這些變數儲存了要連接的 WiFi 熱點的名稱 (SSID) 和密碼。

第 23 行: 要擷取的網頁 URL，在這裡為使用 Yahoo 財經的 API 來獲取 BTC-USD 的即時價格。

```

1  #include <ssl_client.h>
2  #include <ArduinoJson.h>
3  #include <Wire.h>
4  #include "SSD1306.h"
5  #include <WiFi.h>
6  #include <HttpClient.h>
7
8
9  ///
10 /* OLED 顯示設定 */
11 SSD1306 display(0x3c, 21, 22);
12 ///
13
14
15 /* 要連接的wifi熱點 */
16 ///
17 const char* ssid = "Reno";
18 const char* password = "a0907863836";
19 ///
20
21 /* 想透過http抓取的網頁 */
22 ///
23 const char* url = "https://tw.stock.yahoo.com/_td-stock/api/";
24 ///

```

▲圖 2

再來看到下圖 3，

29~30 行：這行程式碼初始化 OLED 顯示屏，並設定 OLED 顯示屏的字型為 ArialMT_Plain_24，即使用 24 號普通字型。

第 34 行：這行程式碼初始化串口通訊，並將傳輸速率設定為 115200。

第 36~48 行：先透過串口輸出印出訊息，提示程式準備連接 WiFi，並顯示 "ready to connect" 字串，再透過第 39 行程式碼開始連接 WiFi，使用事先定義的 SSID 和密碼進行連接。第 42~45 行使用迴圈等待的方式，檢查 WiFi 連線狀態，直到連線成功。迴圈內的 delay(500) 函式會延遲 500 毫秒，而 Serial.println(...) 則透過串口輸出印出 "..." 字串，用於表示正在等待 WiFi 連線的過程。

```
26 void setup()
27 {
28
29     display.init();
30     display.setFont(ArialMT_Plain_24);
31
32
33     /* Serial: uart0 : print on putty */
34     Serial.begin(115200);
35
36     Serial.println("ready to connect\n");
37     /* Wifi 連線 */
38     ///
39     WiFi.begin(ssid,password);
40     ///
41
42     while (WiFi.status() != WL_CONNECTED) {
43         delay(500);
44         Serial.println("...");
45     }
46
47     Serial.print("WiFi connected with IP: ");
48     Serial.println(WiFi.localIP());
49
50
51 }
```

▲圖 3

接下來看到下圖 4，

首先第 58 行：宣告一個 HTTPClient 物件 http，用於發送 HTTP 請求。

60~63 行：使用 http 物件開始與指定的 URL 建立連線，物件發送 GET 請求，並將回應的 HTTP 狀態碼存入 httpCode 變數中，並使用串口輸出將 httpCode 值印出，以檢查 HTTP 連線狀態。

65~71 行：程式碼 65 行進入條件判斷，檢查 httpCode 是否等於 HTTP_CODE_OK，表示連線成功。進入 if 後使用 http 物件的 getString() 方法，獲取 HTTP 回應的網頁原始碼內容，並將其存入 payload 字串變數中，並在 70~71 行建立一個 DynamicJsonDocument 物件 doc，並指定其大小為原始碼內容的長度乘以 2。然後使用 deserializeJson 函式將 payload 字串中的 JSON 資料解析到 doc 物件中。

第 76~84 行：doc 物件中擷取出想要查看的資訊。JSON 資料的結構是 data 陣列中的第一個元素的 price 欄位。將取得的值存入 v 變數中，並使用串口輸出將 "BTC : " 字串以及 v 變數的值印出，以便檢查。

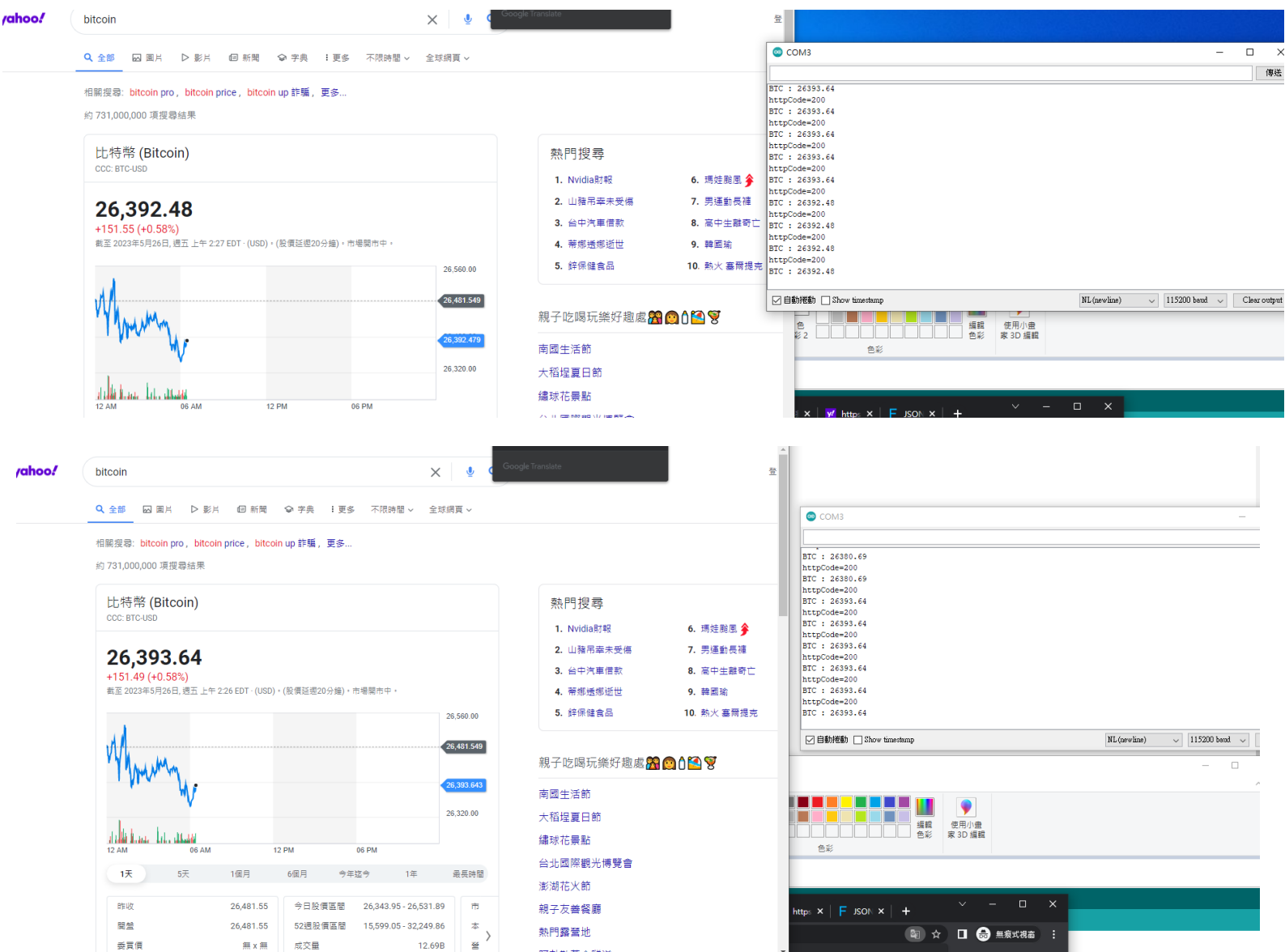
86~90 行：結束 HTTP 連線，釋放相關資源，並在之後的迴圈中加入延遲 3 秒。

```
53  /* 當setup結束後會開始跑loop() */
54
55  void loop()
56  {
57
58      HTTPClient http;
59
60      http.begin(url);
61      int httpCode = http.GET();
62      Serial.print("httpCode=");
63      Serial.println(httpCode);
64
65      if(httpCode == HTTP_CODE_OK){ // 確認連線是否正常
66
67          String payload = http.getString(); // 網頁原始碼內容
68
69          /* 透過ArduinoJson來解析 */
70          DynamicJsonDocument doc(payload.length()*2);
71          deserializeJson(doc,payload);
72
73          /* 擷取想要看的資訊 */
74          ///
75
76          const double v = doc["data"][0]["price"];
77          ///
78
79          Serial.print("BTC : ");
80          Serial.println(v);
81
82          display.clear();
83          display.drawString(0,0,"BTC :\n "+String(v));
84          display.display();
85
86          http.end();
87
88      }
89      delay(3000);
90
91  }
```

▲圖 4

5. 觀察實驗結果

從下圖 5 可發現，當 bitcoin 價格發生變動時，esp32 也成功即時做出更新，並也成功地將變動後的價格顯示在電腦螢幕以及 OLED 上。



▲圖 5

5. 問題與討論

1. 無法成功連接 wifi:

我看許多同學都無法成功連接到自己的手機熱點，以下是我覺得可能的原因：

1. Wi-Fi 熱點的問題：可能是 Wi-Fi 熱點本身出現了問題，例如信號不穩定、連接人數過多等。

2. Wi-Fi 設定的問題：請確保您在程式中正確設置了正確的 SSID 和密碼，以確保與 Wi-Fi 熱點的連接資訊是正確的。

3. 硬體或環境干擾：可能存在硬體或環境上的問題，例如信號干擾、電磁干擾等。這些因素可能導致連接不穩定或無法連接。

4. 程式錯誤：我看許多人是忘記在 `WiFi.begin()` 裡面加上 `ssid, password`，因此會導致錯誤。

2. JSON 解析與資料提取：

在程式碼中，使用 `ArduinoJson` 庫來解析 JSON 檔案並提取比特幣價格。可以深入討論 JSON 解析的過程、如何選擇適當的鍵來擷取資料，以及如何處理嵌套的 JSON 結構。

3. 資料存儲和歷史數據：

目前僅獲取實時比特幣價格，但如果想要存儲並檢索歷史數據呢？以下是討論如何將數據存儲到外部儲存器或雲端服務中。

外部儲存器：您可以使用外部儲存器（例如 SD 卡或 EEPROM）將數據保存在裝置上。這可以通過將數據以特定格式（如 CSV 或 JSON）寫入文件或儲存區來實現。您可以討論如何定義數據格式、如何實現數據的寫入和讀取操作，以及如何管理和組織存儲的數據。

雲端服務：您可以使用雲端服務（例如數據庫服務或物聯網平台）將數據保存在遠程伺服器上。這可以通過建立數據庫表或集合來實現，並使用 API 或協議將數據傳輸到遠程服務器。您可以討論如何設計數據庫結構、如何建立連接和通訊，以及如何進行數據的查詢和檢索。

6. 心得與感想

在進行這個實驗中，我運用了 OLED 顯示器、esp32 開發板和 Wi-Fi 連接，以獲取比特幣的實時價格並將其顯示在 OLED 上。

這次實驗讓我獲得了寶貴的經驗，不僅深入瞭解了如何連接外部設備和進行網絡通信，還學習了數據解析和顯示的技巧。這讓我對於物聯網和數據處理領域有了更深入的了解，並激發了我進一步探索的動力。透過不斷的實驗和學習，我相信我可以不斷提升自己在物聯網和數據處理領域的技術和應用方面。

同時，在過程中，我遇到了一些挑戰和問題，例如如何處理網絡請求的錯誤、如何正確解析 JSON 數據以及如何適應不同的數據格式。這些問題激勵我進一步深入學習，通過研究和尋找解決方案，我得以克服這些困難並取得進展。

綜上所述，這次實驗讓我獲得了許多寶貴的經驗和知識。我不僅學會了如何連接和控制外部設備，還學習了如何處理網絡數據並將其可視化。這次實驗激發了我對物聯網和數據處理的興趣，並鼓勵我繼續深入學習和探索相關領域的知識和技術。

7. 參考文獻

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

https://tw.stock.yahoo.com/_td-stock/api/resource/StockServices.stockList;autoRefresh=1659720073465;fields=avgPrice,orderbook;symbols=BTC-USD?bkt=&device=desktop&ecma=modern&feature=ecmaModern,useNewQuoteTabColor&intl=tw&lang=zh-Hant-TW&partner=none&prid=3ec7fd9heqk4r®ion=TW&site=finance&tz=Asia/Taipei&ver=1.2.1432&returnMeta=true