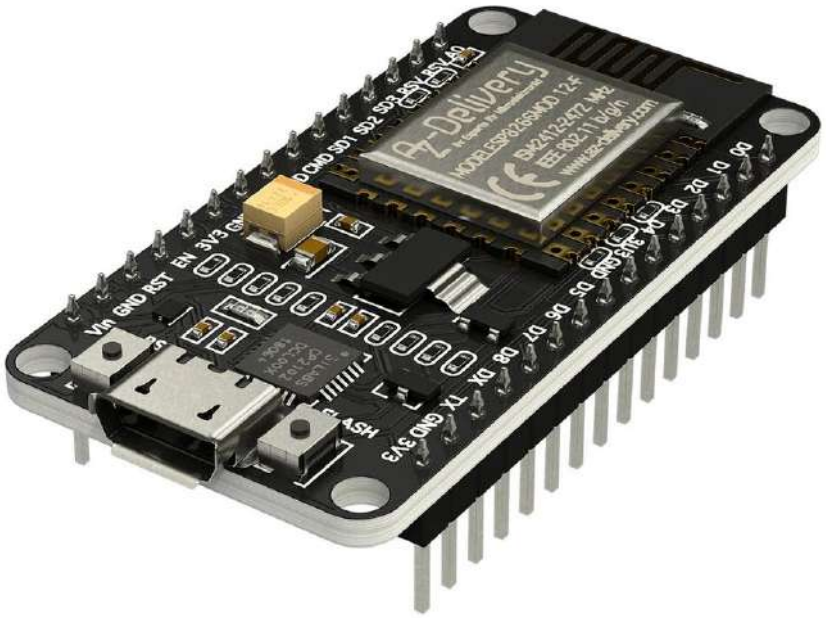
Arduino實作 - ESP32智慧聯網時鐘

資一孝-蔡宥勝



一. 研究動機:

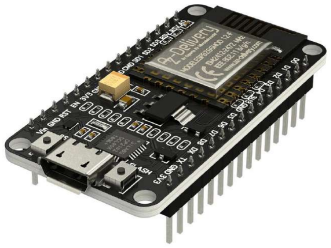
因為前陣子很流行那種DIY智慧魔境，我那時候就特別喜歡裡面的天氣功能，一看就知道外面的天氣，剛好現在有這個機會，我自己也想要做出一個，只是通常那種大型螢幕價格比較昂貴，所以我就在想有沒有用手邊現有零件，或是使用價格便宜的零件，做出可以達到類似功能的東西

二. 功能介紹:

可以在連線後在第一排顯示日期，第二排顯示現在時間與讀取出來的溫度，最後一排可以顯示OpenWeatherMap提供的資訊，並且可以使用手機設定時區與要顯示的東西，也可以很方便的連線到網路

三.使用的元器件:

1.NodeMCU v2



工作電壓:5V
板載通訊模組:WiFi 2.4 / 藍芽4.2
閃存容量:32Mbit

2.麵包板

尺寸:170 洞

3.I2c LCD 2004

工作電壓:5V
通訊協議:I2c

4.SPI SD Card Reader

工作電壓:3.3V / 5V
通訊協議:SPI

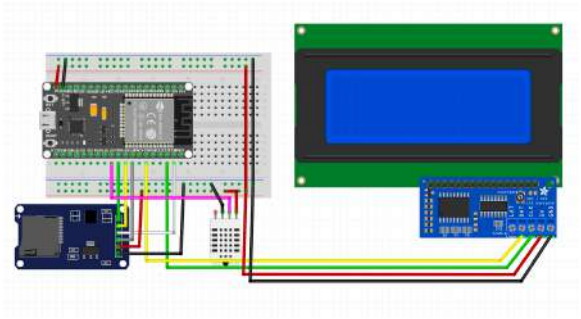
6.DHT22

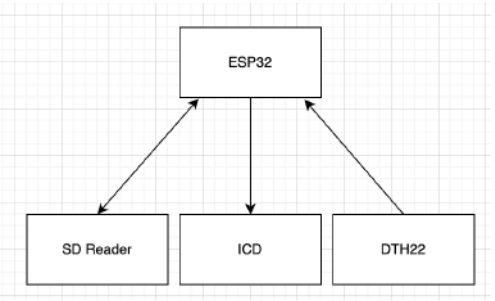
工作電壓:5V
最佳工作溫度: -40~80°C ±0.5°C
濕度感測: 0~100% ±2-5%
最高讀取頻率: 0.5 Hz

5.杜邦線

尺寸: 26 AWG

四.電路連接方式:





線路對照表:

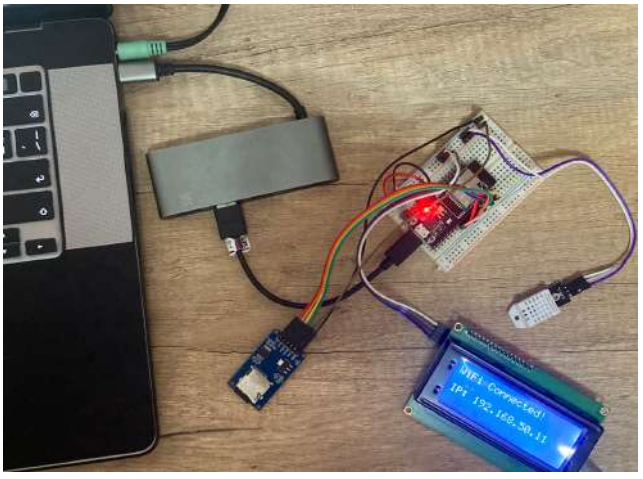
ESP32	I2C LCD 20x4
VCC	VCC
GND	GND
D1	SCL
D2	SDA

ESP32	I2c LCD
VCC	VCC
GND	GND
D1	SCL
D2	SDA

ESP32	SPI SD Card Reader
VCC	VCC
GND	GND
VSPI SS	CS
VSPI SCK	SCK
MOSI	MOSI
MISO	MISO

ESP32	DHT22
VCC	VCC
GND	GND
GPIO 16	OUT

電路實體圖方式:



五.效果示範:

Wi-Fi連線中:

程式514~550行:
持續判斷連線狀態重試時間大於設定值(15秒秒)，過程中游標重複指定同一個地方顯示字符

如果連線失敗(或第一次連線)開啟熱點供使用者設定:

程式607~642行:
當連線失敗後設定為AP mode並顯示SSID和密碼

當手機連線到Wi-Fi:

程式940~945:
當連線用戶數大於1(有裝置連線至此AP) 顯示設定提示

SSID選擇:

當瀏覽器取得root目錄(index.html)且狀態為未連線回傳程式256行HTML變數，等待手機端回傳資料
(過程中持掃描WIFI)

密碼輸入:

時鐘連線到取得IP:

當程式第825行路由取得到回應data時嘗試連接WIFI，如果成功顯示IP，並將狀態改變為已連接，並將帳號密碼保存到SD卡

(以連接後會持續判斷連線狀態，如果中斷會再重新連線，如果失敗會將狀態設定為未連線並且開啟AP mode重試)

連線成功後設定頁面

六.構建介紹:

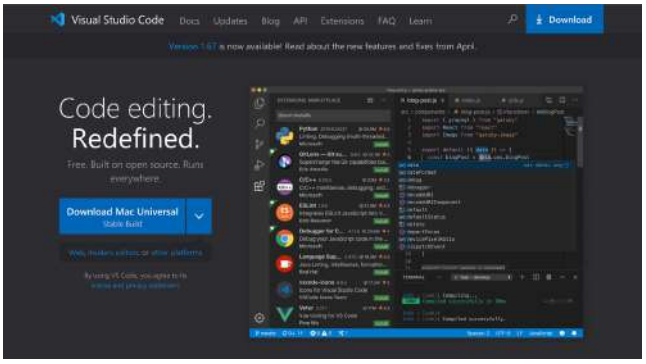
撰寫環境架設(MacOS 10.15 搭配 VScode Arduino擴充套件):

1. 先到Arduino IDE下載頁面安裝IDE (我使用是2.0 RC)

2. 安裝開啟後由於Arduino 函式庫並不會內建ESP開發版，
所以必須先到Arduino IDE設定頁面添加開發版網址

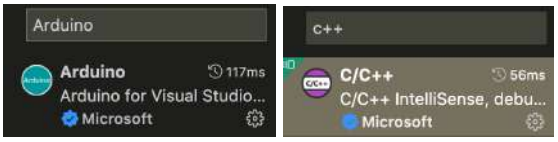
然後在開發版管理員添加 ESP32 才能進行開發

3. 然後再來到VSCode下載頁面下載編輯器



安裝完成後開啟

4. 接下來請到擴充套件安裝C++/Arduino擴充套件



5. 然後將ESP連接到電腦後下方選擇開發板跟連接阜

七.使用的函式庫 (提供GitHub或是Arduino Librarys連結):

1.ESP32(WiFi/WiFiClient/SD) (上面安裝開發版就會一併安裝)

2.NTPClient (同步時間)

3.LiquidCrystal_I2C (LCD函式庫)

4.Arduino JSON (JSON DATA 解析)

5.ESPAsyncWEBServer (網頁伺服器)

6.HTTPClient (HTTP發送請求)

7.DHT Sensor (濕溫度感測)

八.安裝函式庫:

將下載的ZIP包解壓縮放置到放置到Arduino/Library

函式簡略介紹:

1. WIFi連線部分:

a. `WiFi.begin(WIFI_ssid,WIFI_password)`
這個函式可以用來開始WiFi連線(char* char*)

b. `WiFi.status() == WL_CONNECTED`
WiFi.status本身用來回傳是否連線，可以判斷是否為WL_CONNECTED來確認連線(可配合While等待連線)

c. `WiFi.localIP()`
這個函式可以用來取得連線到的IP
(*註:回傳的類型為IPAddress)

d. `WiFi.softAPConfig(local_IP, gateway, subnet);`
這個函式可以用來設定熱點
(*註:參數的類型皆為IPAddress,
需要使用 IPAddress local_IP(192, 168, 0, 1); 這種定義方法
local_IP為主機IP名稱，gatway為遮罩IP,subnet則為子網域)

e. `WiFi.softAP(SSID, PASSWORD, 1, 0, 8);`
這個函式是用來開啟熱點的，會回傳0/1代表成功/失敗
(*註:後面那三個數字是可選的，分別代表意思為
頻道/是否隱藏/最多連線裝置)

f. `WiFi.softAPgetStationNum();`
這個函式用來取得連線的裝置數量

2. LCD控制部分:

a. `lcd.clear();`
這個函式可以用來清除LCD

b. `lcd.setCursor(0, 0);`
這格個函式是用來設定縣市游標的

c. `lcd.print("Initializing SD card");`
這個函式可以用來取得連線到的IP
(*註:回傳的類型為IPAddress)

3. Server設定部分:

a. server.on("/", HTTP_GET, [](AsyncWebServerRequest *request)
可以用來設定路由

b. 如果想要回在取得路由時回覆
   I. request->send_P(200, "text/html", wait_html);
   send_P用來傳送HTML網頁
   II. request->redirect("/");
   用來把使用者導向到已經設定過的路由

c. 取得表單
   I. request->getParam("password")->value().toCharArray(WIFI_password, 50);

4. 濕溫度感測器:

a. DHT dht(17, DHT22);
用來設定類型為DHT22, 角位在17

b. dht.readHumidity();
取得濕度

c. dht.readTemperature();
取得溫度
(裡面可以放置一組BOOL來設定要不要取得華氏)

程式重點介紹:

這段程式碼用來取代Delay: (使用在Wi-Fi連線中時的轉圈效果)

```
void loop(){
   x = millis();
   if (millis() - x> 1000)
   {
       x = millis();
   }
}
```

用來設定熱點模式 (使用在連線失敗開啟熱點):

```
IPAddress local_IP(192, 168, 0, 1);
IPAddress gateway(192, 168, 0, 1);
IPAddress subnet(255, 255, 255, 0);
lcd.setCursor(1, 0);
lcd.print("Setting AP Mode...");
lcd.setCursor(2, 2);
lcd.print("Waitting..");
WiFi.softAPConfig(local_IP, gateway, subnet);
String Ap_ssid_idtemp = "SetupClock_" + Ap_ssid_id;
String Ap_passwordtemp = "Setup" + Ap_password;
boolean result = WiFi.softAP(Ap_ssid_idtemp.c_str(),
Ap_passwordtemp.c_str(), 1, 0, 1);
```

滾動顯示:

```
else if (scroolli <= 20)
{
   lcd.setCursor(0, 3);
   lcd.print(scroll_text.substring(20 - scroolli, 20));
}
else if (scroolli > 20 && scroolli < 25)
{
   lcd.setCursor(0, 3);
   lcd.print(scroll_text);
}
else if (scroolli >= 25)
{
   lcd.setCursor(scroolli - 25, 3);
   lcd.print(scroll_text.substring(0, 20 - (scroolli - 25)));
}
```

計時取得時間:

```
if (millis() - temptime2 >= 30000)
   {
     {
       timeClient.update();
       h = dht.readHumidity();
       c = dht.readTemperature();
       f = dht.readTemperature(true);
       temptime2 = millis();
       weather();
     }
}
```

問題討論:

問1.

Q.為何使用ESP32而不選擇常見的UNO板

A.因為Arduino本身並不具備時鐘計數器功能，所以就換一個思路使用網路時間對時(NTP)作為時間來源，好處是只要有網路，連線的到伺服器就一直回有準確的時間，缺點是如果沒網路會不能用

問2.

Q.那是如何拿到現在的天氣資訊的?

A.因為上面有提到可以聯網了，那這個功能當然好好利用一下，使用Arduino上的Http Client Library(HTTP客戶端)連線到OpenWeatherMap(因為完全免費，只要註冊就可以使用)使用他們提供的API取得天氣狀況

心得:

在撰寫這種單晶片我覺得最難的地方不是程式要怎麼寫，而是要怎麼執行，因為在單晶片效能本身並不會太強，記憶體與ROM可以儲存的東西很少，所以可能會在執行過程高出限制而強制重開/當機，而且如果當機掉沒有象VS這樣會幫你標示出錯誤在哪，因為程式已經燒錄進去了，還有第二難的是去找適用於這塊開發版可以使用的函式庫，以及包裝起來不會產生衝突。

附加文件:

1. 程式碼檔案(.ino)
2. 電路圖檔案(.fzz)