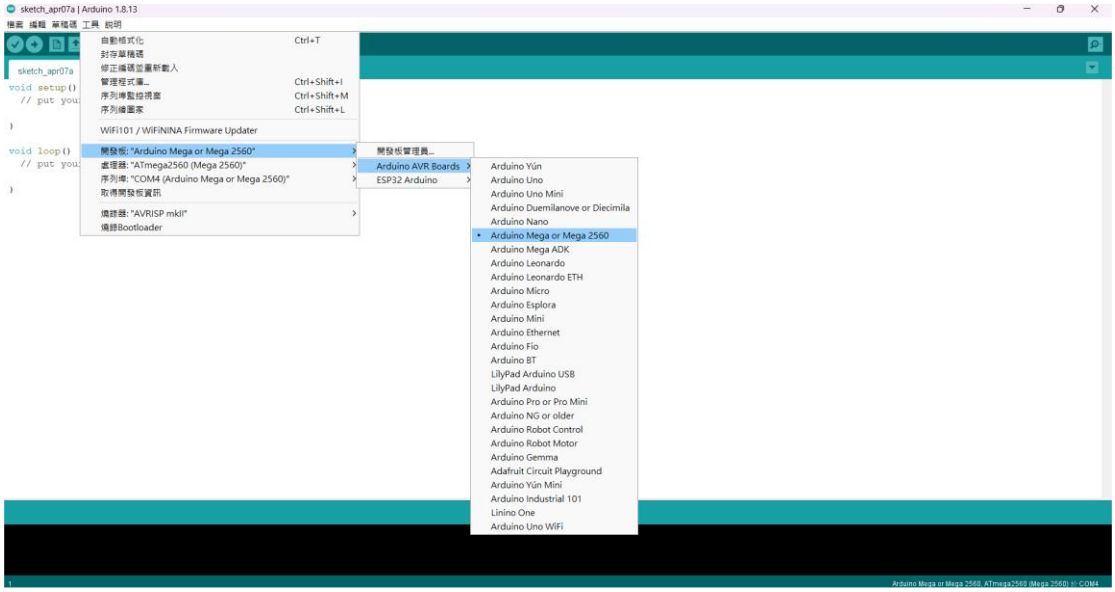
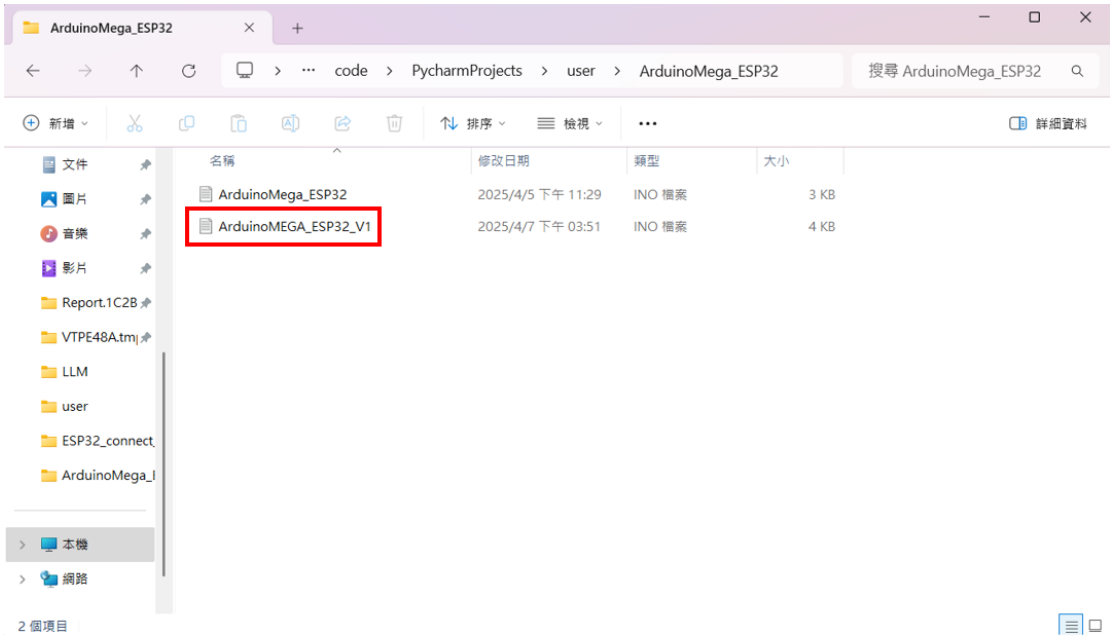
1. 燒錄程式碼進 Arduino Mega

工具 → 開發版 → Arduino AVR → Arduino Mega 2560





```
#include <Keypad.h>

// LED 腳位定義
const int successLED = 2;
const int failureLED = 3;

const byte ROWS = 4;
const byte COLS = 4;

// 定義鍵盤按鍵排列
char keys[ROWS][COLS] = {
  {'D','C','B','A'},
  {'#','9','6','3'},
  {'0','8','5','2'},
  {'*','7','4','1'}
};

byte rowPins[ROWS] = {4, 5, 6, 7};
byte colPins[COLS] = {8, 9, 10, 11};

// 初始化鍵盤物件
Keypad keypad = Keypad(makeKeymap(keys), rowPins, colPins, ROWS, COLS);

const int maxInputLength = 6;
```

```
const int maxOTPCount = 10;          // 陣列最多儲存 10 組 OTP
String otpList[maxOTPCount];         // 儲存 OTP 的陣列
int otpCount = 0;                    // 當前 OTP 數量
String enteredOTP = "";

void setup() {
  Serial.begin(115200);          // PC 監控
  Serial1.begin(115200);         // 與 ESP32 通信
  pinMode(successLED, OUTPUT);
  pinMode(failureLED, OUTPUT);
  Serial.println("系統已啟動，等待按鍵輸入...");
}

void loop() {
  // 檢查是否有新的 OTP 從 ESP32 傳入
  if (Serial1.available()) {
    String newOTP = Serial1.readStringUntil('\n');
    newOTP.trim();
    Serial.print("收到來自 ESP32 的 OTP：");
    Serial.println(newOTP);

    // 加入新 OTP
    addNewOTP(newOTP);
  }
```

```
char key = keypad.getKey();
if (key) {
    feedbackBlink();
    Serial.print("按鍵：");
    Serial.println(key);

    if (key == '#') {
        Serial.print("已輸入 OTP：");
        Serial.println(enteredOTP);
        if (verifyOTP(enteredOTP)) {
            Serial.println("OTP 正確！");
            activateSuccess();
        } else {
            Serial.println("OTP 錯誤！");
            activateFailure();
        }
        enteredOTP = "";   // 重置輸入
    } else if (key == '*') {
        enteredOTP = "";
        Serial.println("輸入已重置");
        feedbackBlink();
    } else {
        if (enteredOTP.length() < maxInputLength) {
```

```
            enteredOTP += key;
            Serial.print("當前輸入：");
            Serial.println(enteredOTP);
        } else {
            Serial.println("輸入長度已達上限");
            feedbackBlink();
        }
    }
  }
}

// 加入新 OTP 並移除最舊的一組
void addNewOTP(String newOTP) {
  if (otpCount < maxOTPCount) {
    otpList[otpCount] = newOTP;
    otpCount++;
  } else {
    // 移除最舊 OTP
    for (int i = 0; i < maxOTPCount - 1; i++) {
      otpList[i] = otpList[i + 1];
    }
    otpList[maxOTPCount - 1] = newOTP;
  }
  Serial.println("當前 OTP 列表：");
```

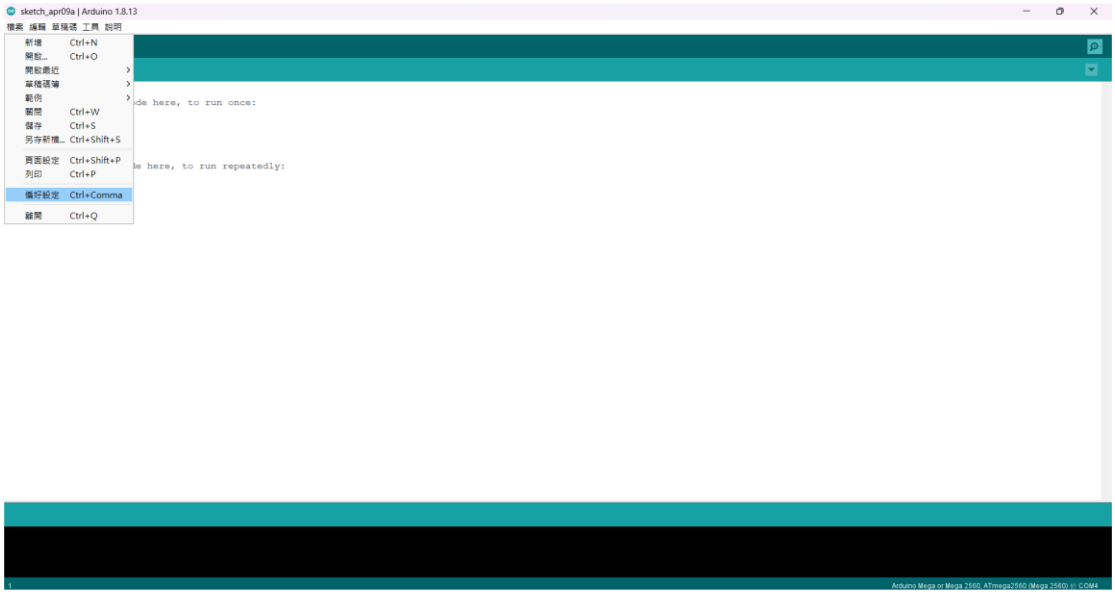
```
    for (int i = 0; i < otpCount; i++) {
        Serial.println(otpList[i]);
    }
}

// 驗證輸入的 OTP 是否存在
bool verifyOTP(String inputOTP) {
    for (int i = 0; i < otpCount; i++) {
        if (otpList[i] == inputOTP) {
            // 刪除已驗證的 OTP
            for (int j = i; j < otpCount - 1; j++) {
                otpList[j] = otpList[j + 1];
            }
            otpCount--;
            return true;
        }
    }
    return false;
}

// 成功提示：LED 輪流亮滅
void activateSuccess() {
    for (int i = 0; i < 5; i++) {
        digitalWrite(successLED, HIGH);
```

```
        digitalWrite(failureLED, LOW);
        delay(300);
        digitalWrite(successLED, LOW);
        digitalWrite(failureLED, HIGH);
        delay(300);
    }
    digitalWrite(successLED, LOW);
    digitalWrite(failureLED, LOW);
}

// 失敗提示：紅燈慢閃
void activateFailure() {
    for (int i = 0; i < 3; i++) {
        digitalWrite(failureLED, HIGH);
        delay(400);
        digitalWrite(failureLED, LOW);
        delay(400);
    }
}

// 按鍵回饋：短暫亮燈
void feedbackBlink() {
    digitalWrite(successLED, HIGH);
    delay(100);
```

```
    digitalWrite(successLED, LOW);

}
```

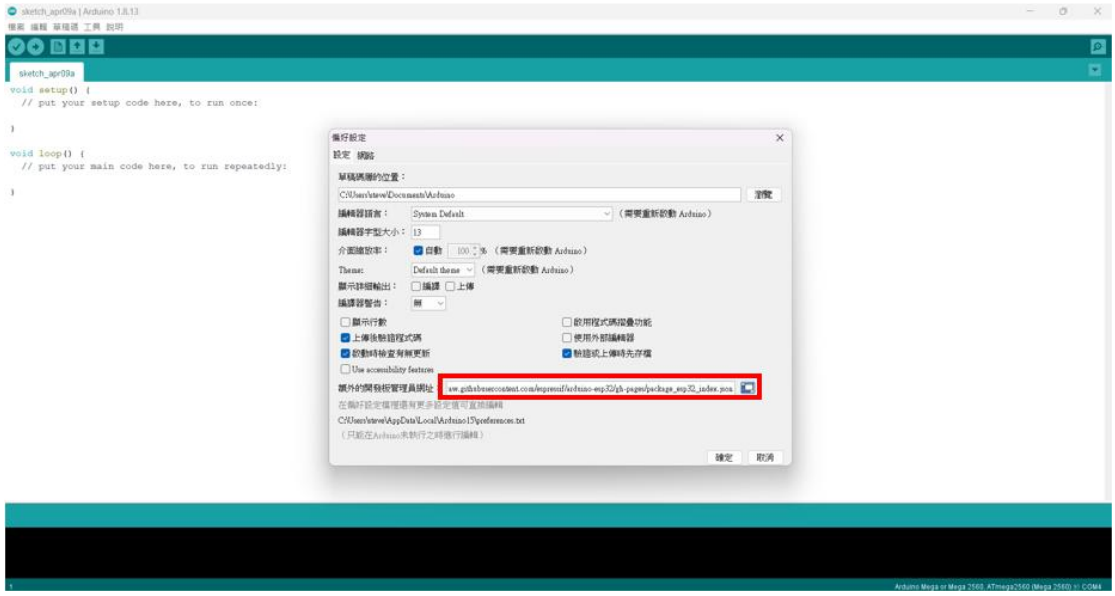
2. 安裝 ESP32

Step1.檔案 → 偏好設定

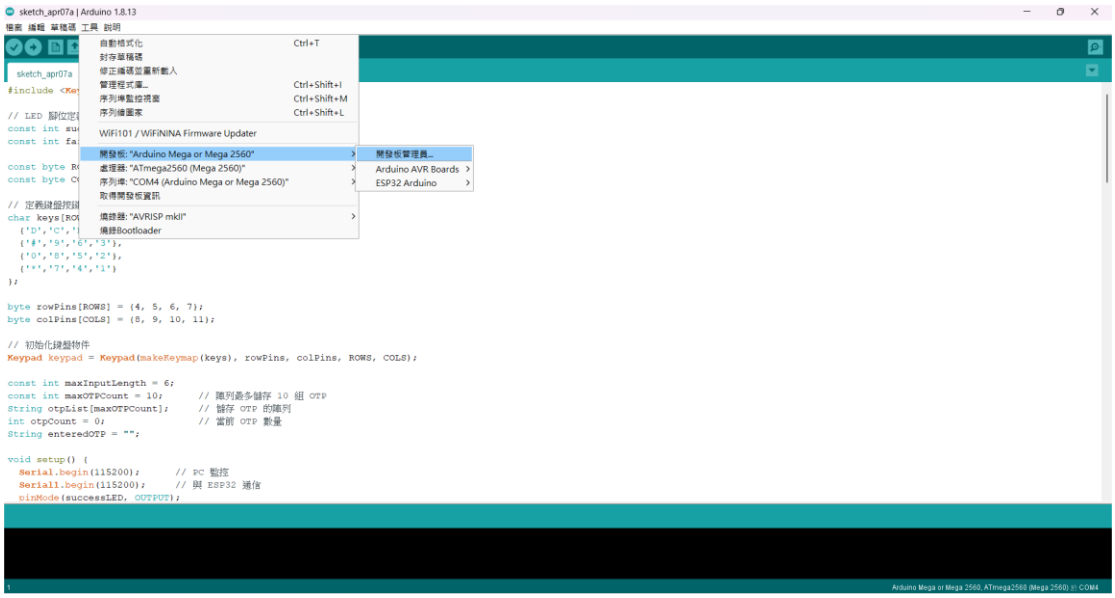


Step2.將下面文字取代紅框內文字，完成後按下確定

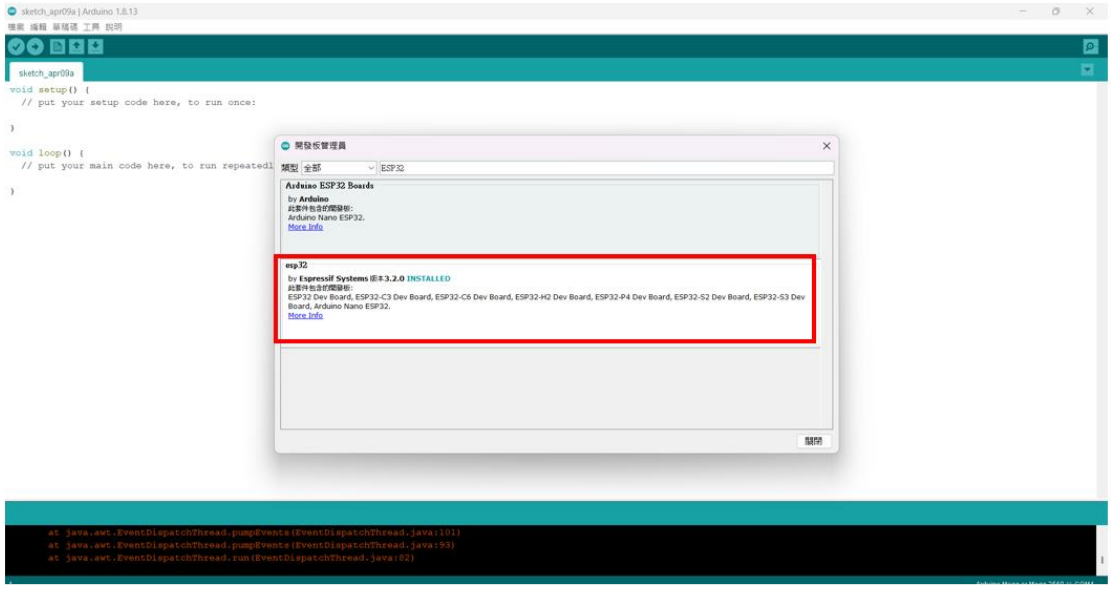
https://raw.githubusercontent.com/espressif/arduino-esp32/gh-pages/package_esp32_index.json



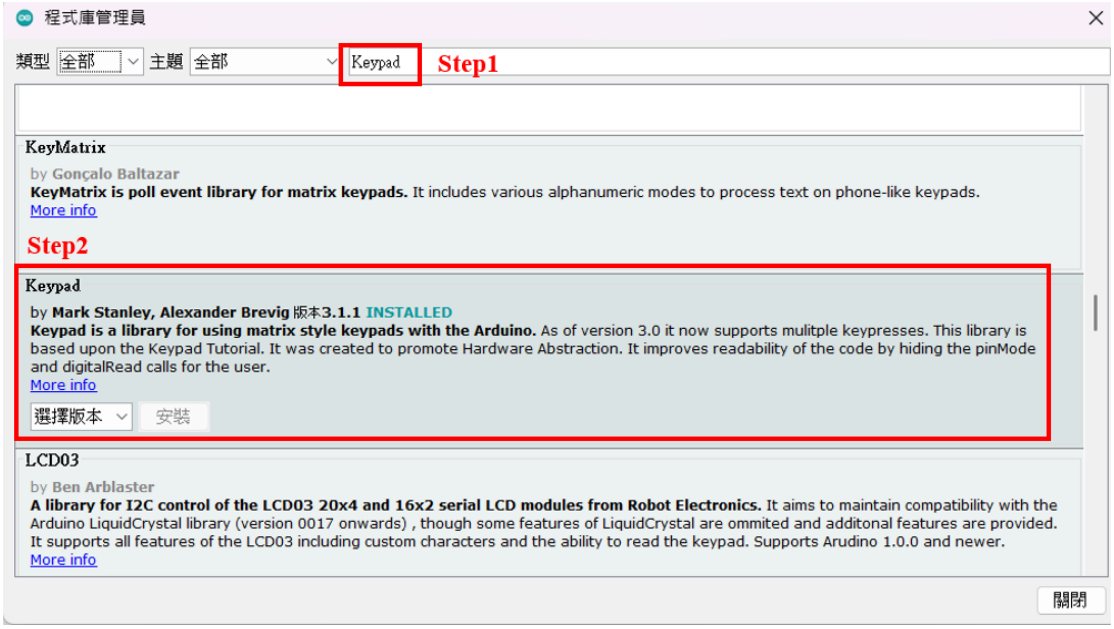
Step3.工具 → 開發版 → 開發版管理員



Step4.輸入 ESP32，選擇 by Espressif Systems 這個

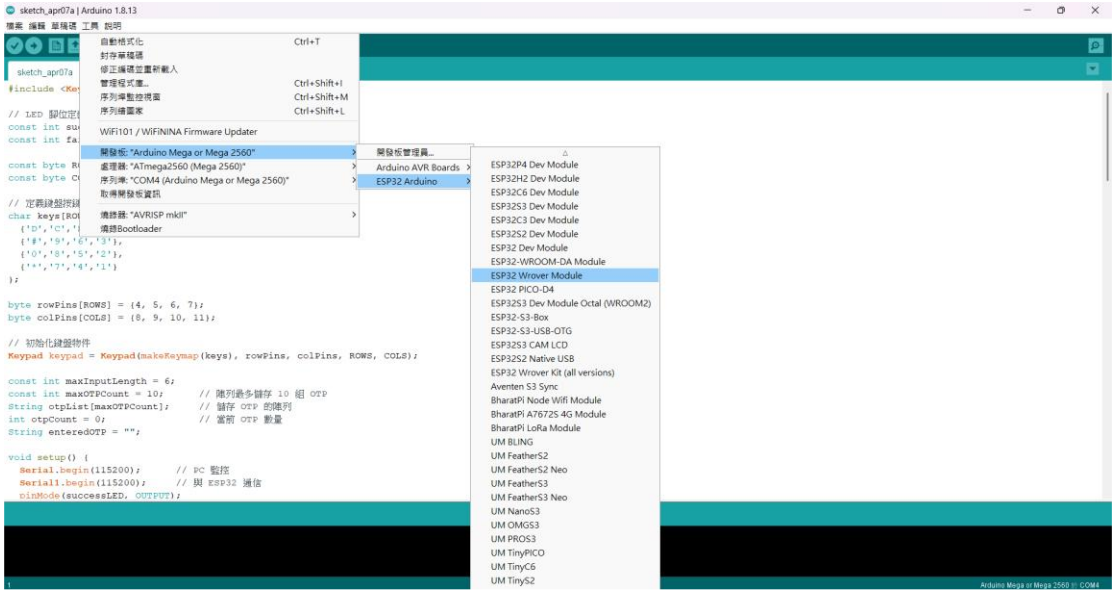


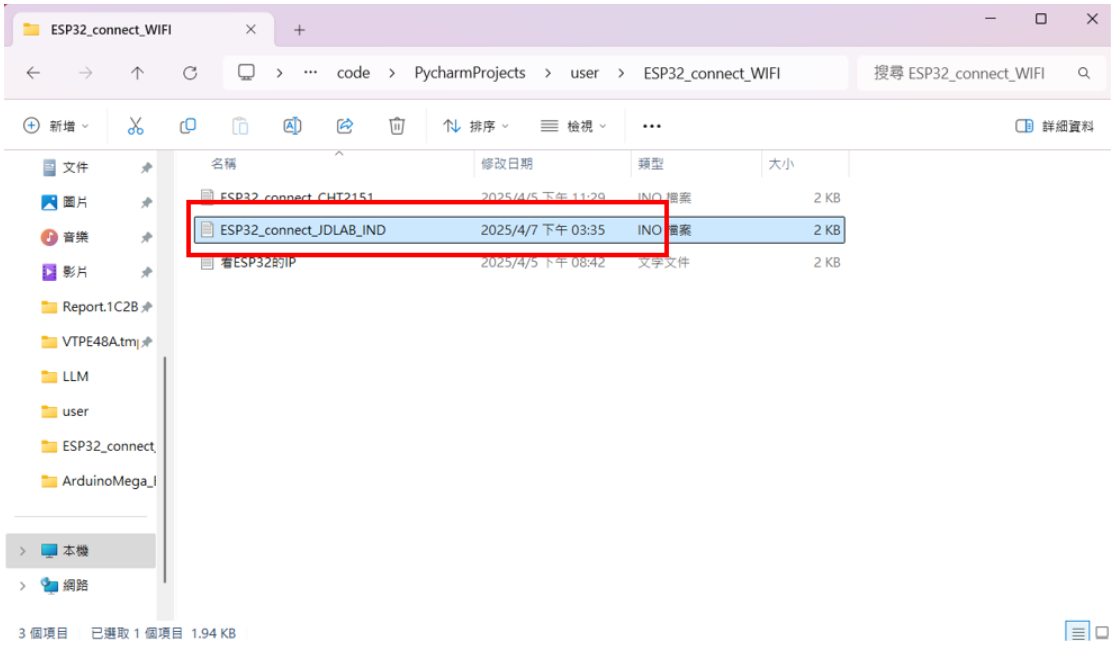
Step5.輸入 Keypad，選擇 by Mark Stanley 這個



3. 燒錄程式進 ESP32

工具 → 開發版 → ESP32 Arduino → ESP32 Wrover Module (因為我的板子型號是 ESP32 Wroom 32E)

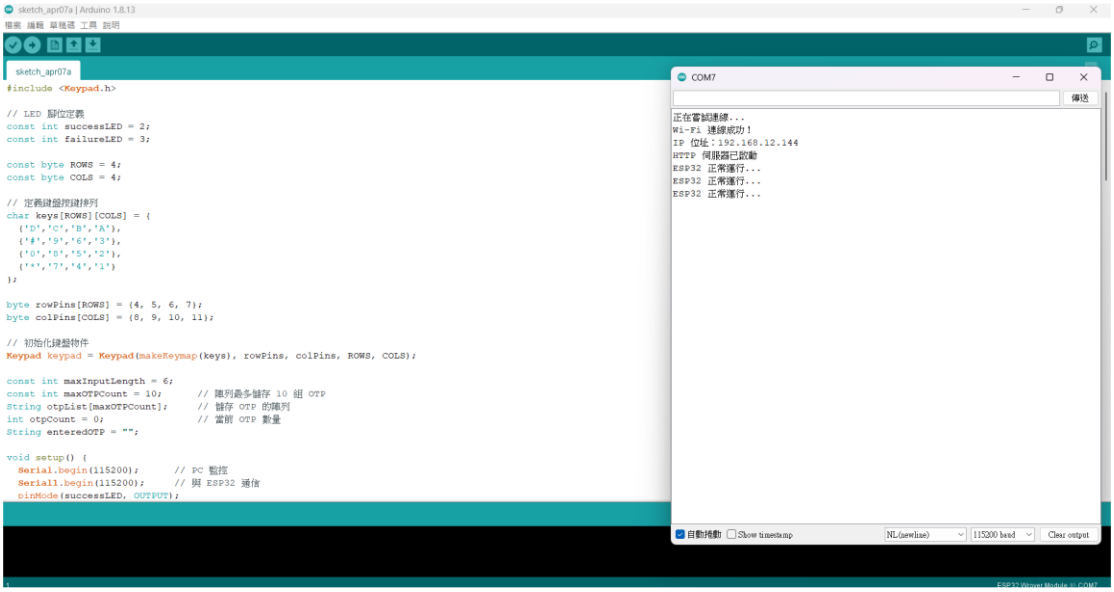




注意!!!

一開始請將固定 IP 設定註解，燒錄後開啟監控視窗查看 IP，完成後再將 ESP32 的固定 IP 改成剛剛看到的 IP，避免 IP 衝突

比如下面我上電後開啟監控視窗，看到 IP 位置為 192.168.12.144



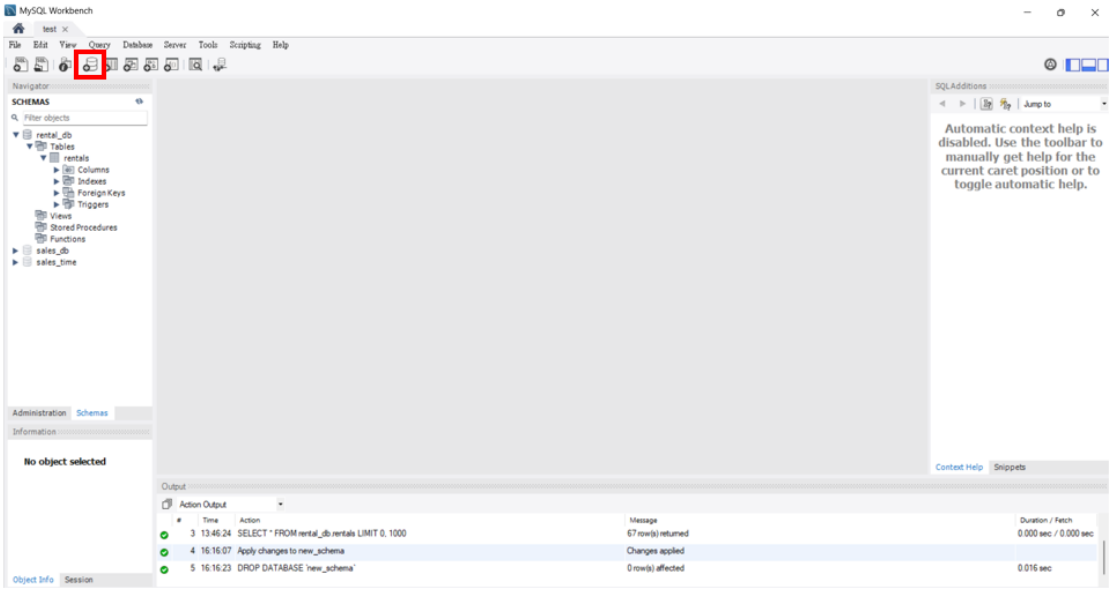

```
#include <WiFi.h>
#include <WebServer.h>

const char* ssid = "JDLAB_IND";            // Wi-Fi 名稱
const char* password = "40427200";   // Wi-Fi 密碼

// 固定 IP 設定
IPAddress local_IP(192, 168, 12, 144);   // ESP32 保持的固定 IP
IPAddress gateway(192, 168, 12, 1);       // 路由器的 IP 位址
IPAddress subnet(255, 255, 255, 0);      // 子網路遮罩
IPAddress primaryDNS(8, 8, 8, 8);         // 首選 DNS
IPAddress secondaryDNS(8, 8, 4, 4);      // 備用 DNS


WebServer server(80);   // 設定 HTTP 伺服器在端口 80

String received_otp = "";
```

```
// 當收到 /send-otp 請求時的處理
void handleSendOTP() {
  if (server.hasArg("otp")) {
    received_otp = server.arg("otp");
    Serial.println("收到 OTP：");
    Serial.println(received_otp);
    Serial2.println(received_otp);  // 傳送 OTP 至 Arduino
    server.send(200, "text/plain", "OTP 接收成功");
  } else {
    Serial.println("未收到有效 OTP 參數");
    server.send(400, "text/plain", "缺少 OTP 參數");
  }
}

void setup() {
  Serial.begin(115200);
  Serial2.begin(115200, SERIAL_8N1, 16, 17);  // 與 Arduino 通信

  // 設定固定 IP
  if (!WiFi.config(local_IP, gateway, subnet, primaryDNS, secondaryDNS))
  {
    Serial.println("STA 設定失敗！");
  }
  WiFi.begin(ssid, password);
  Serial.println("連線中...");
  Serial.println("ESP32 測試 - 串口初始化成功");
  while (WiFi.status() != WL_CONNECTED) {
    delay(1000);
    Serial.println("正在嘗試連線...");
  }

  Serial.println("Wi-Fi 連線成功！");
  Serial.print("IP 位址：");
```

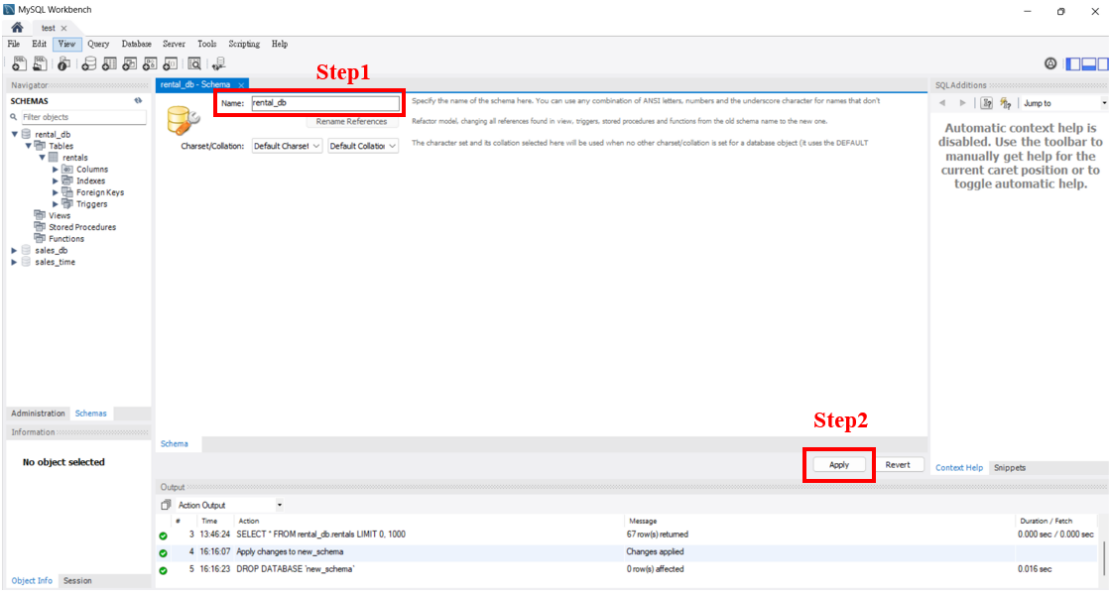
```
  Serial.println(WiFi.localIP());

  // 註冊處理程序
  server.on("/send-otp", HTTP_POST, handleSendOTP);

  // 啟動伺服器
  server.begin();
  Serial.println("HTTP 伺服器已啟動");
}

void loop() {
  Serial.println("ESP32 正常運行...");
  delay(1000);
  server.handleClient();  // 處理客戶端連線
  delay(100);  // 防止卡死
}
```

4. 建立資料庫

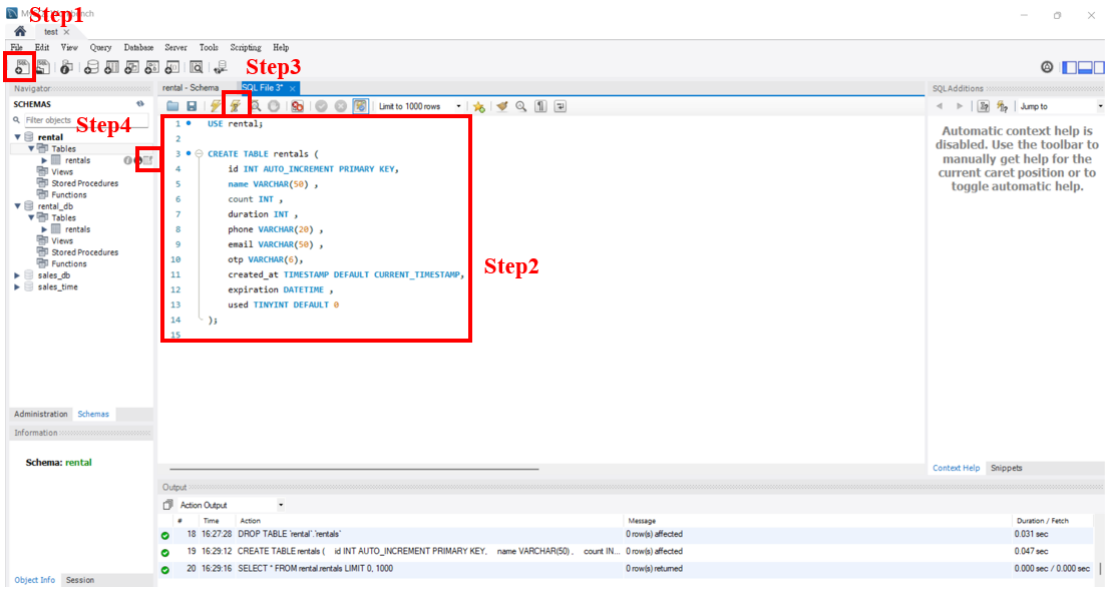
Step1.開啟 mysql → Create a new Schema

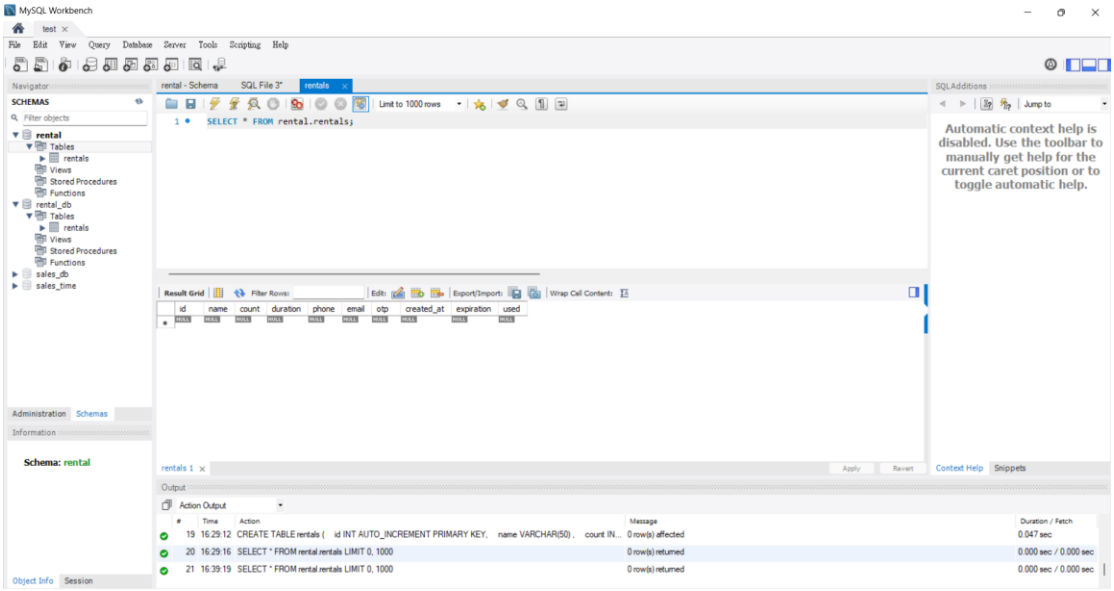


Step2. 輸入 rental_db，完成後 apply



Step3. 左上角的 Create a SQL tab → 中間白色區域貼上程式碼 → 中間偏上的閃電(Execute the statement) → 展開資料庫後點擊最右邊的表格即可看到列表

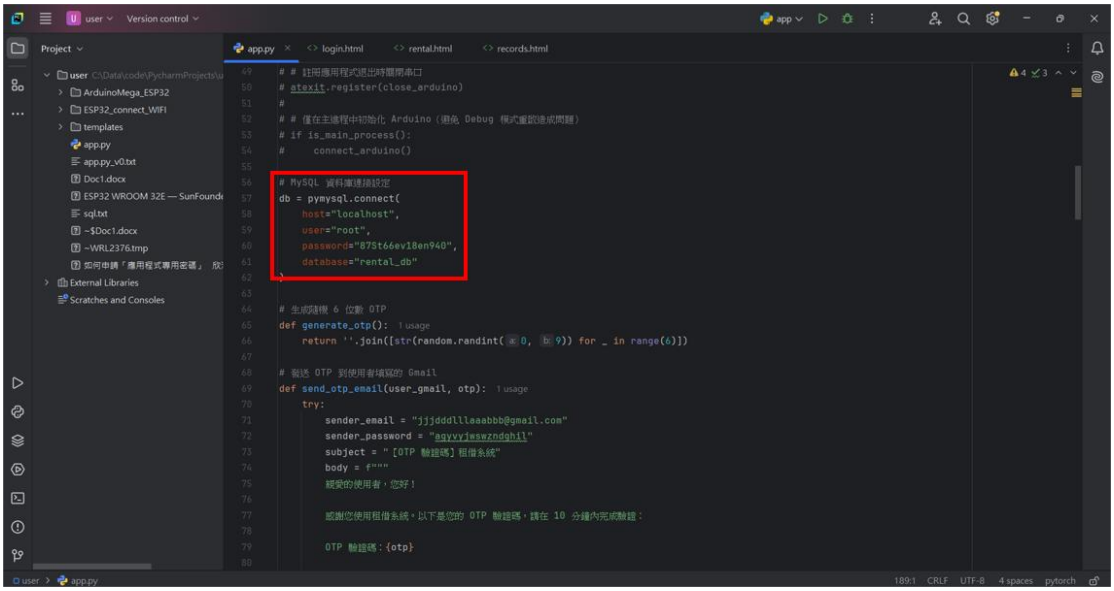




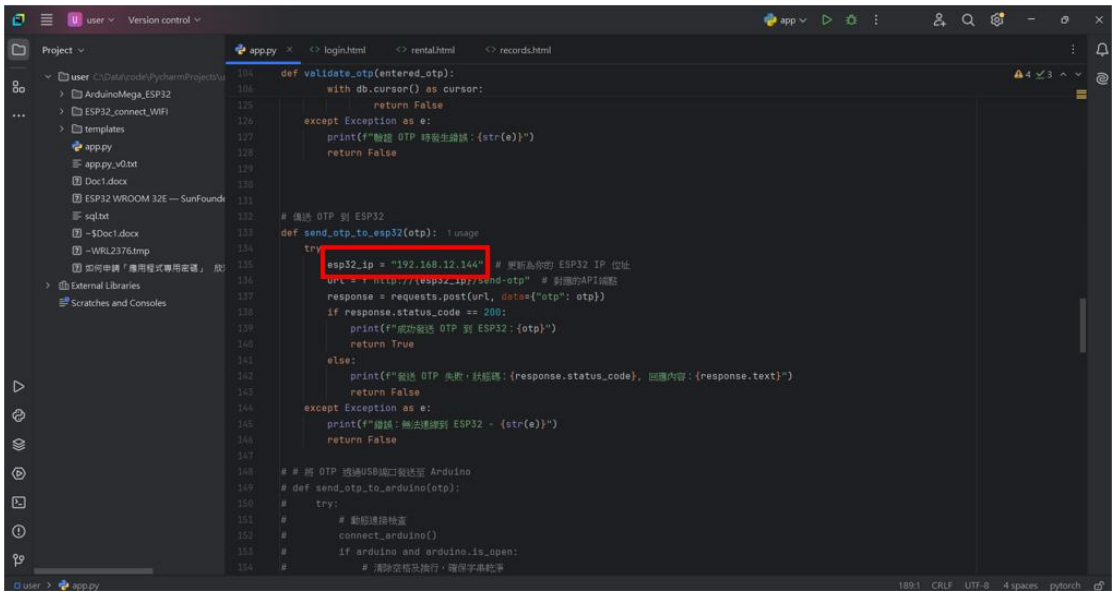
```
USE rental_db;

CREATE TABLE rentals (
    id INT AUTO_INCREMENT PRIMARY KEY,
    name VARCHAR(50) ,
    count INT ,
    duration INT ,
    phone VARCHAR(20) ,
    email VARCHAR(50) ,
    otp VARCHAR(6),
    created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
    expiration DATETIME ,
    used TINYINT DEFAULT 0
);
```

5. 使用 Pycharm 開啟 app.py (環境自己調整)

Step1. 更改資料庫設定

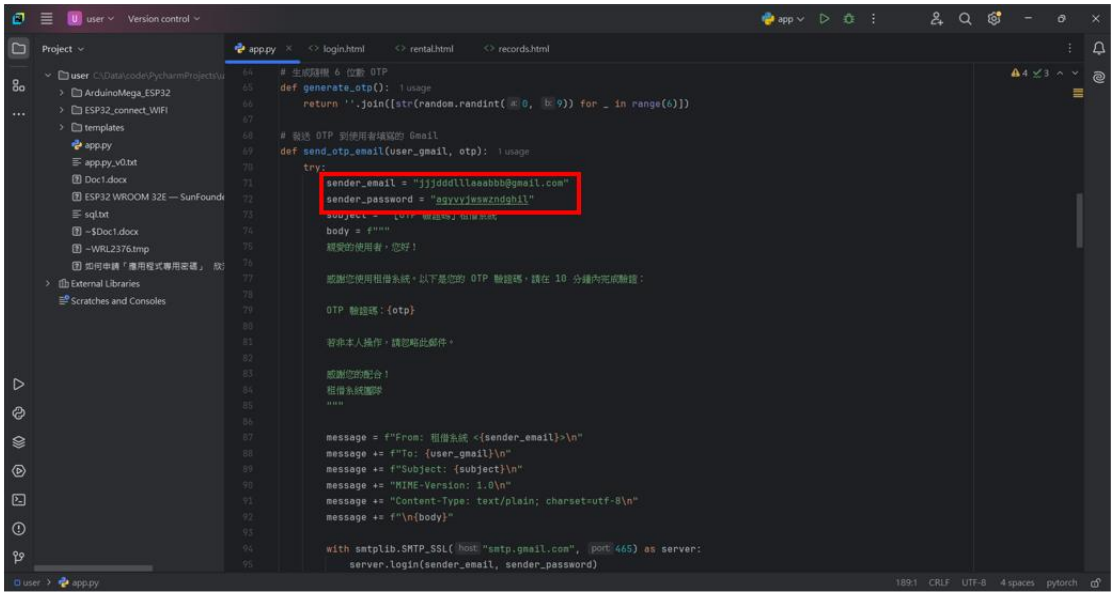


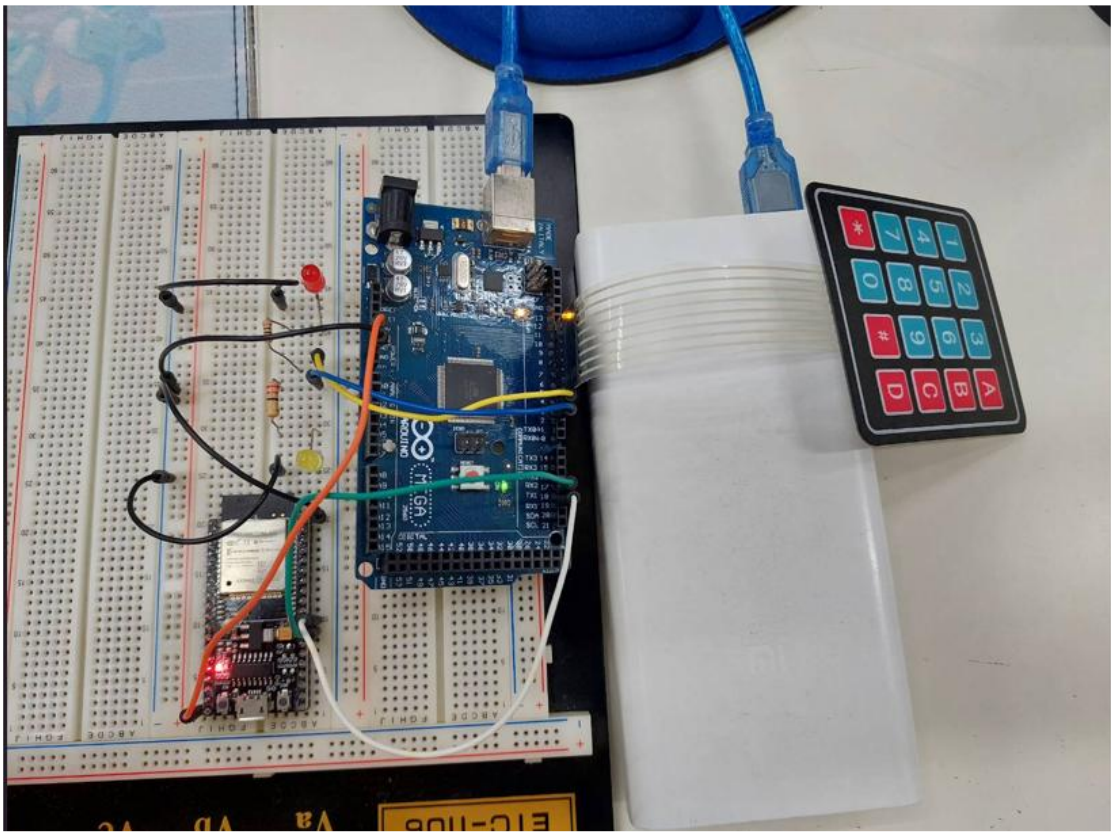
Step2. 更改 ESP32 的 IP 為剛剛設定的固定 IP



Step3.參考以下網址並更改專用密碼，完成後執行即可

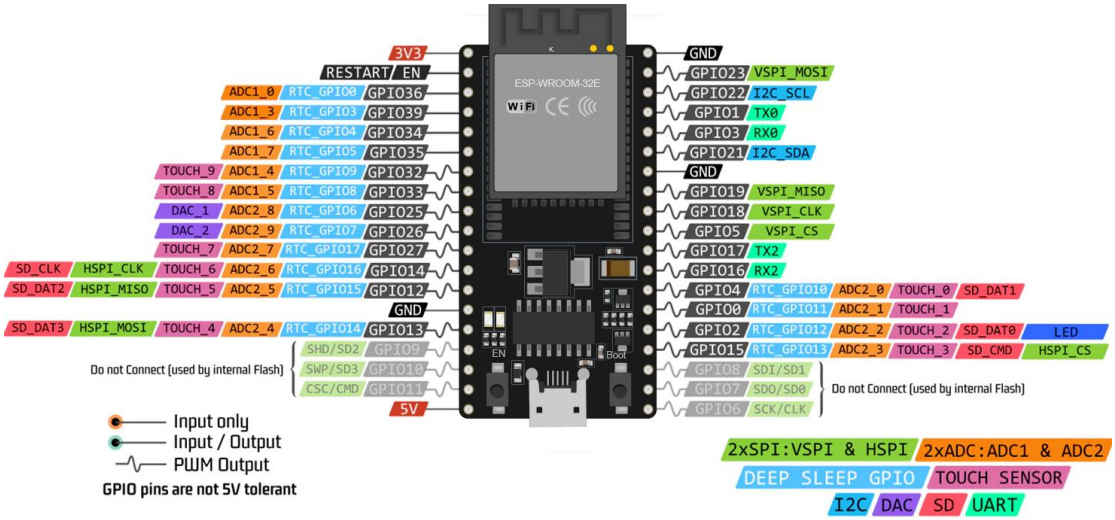
https://shinher.gitbook.io/shinher/ru-he-shen-qing-ying-yong-cheng-shi-zhuan-yong-mi-ma

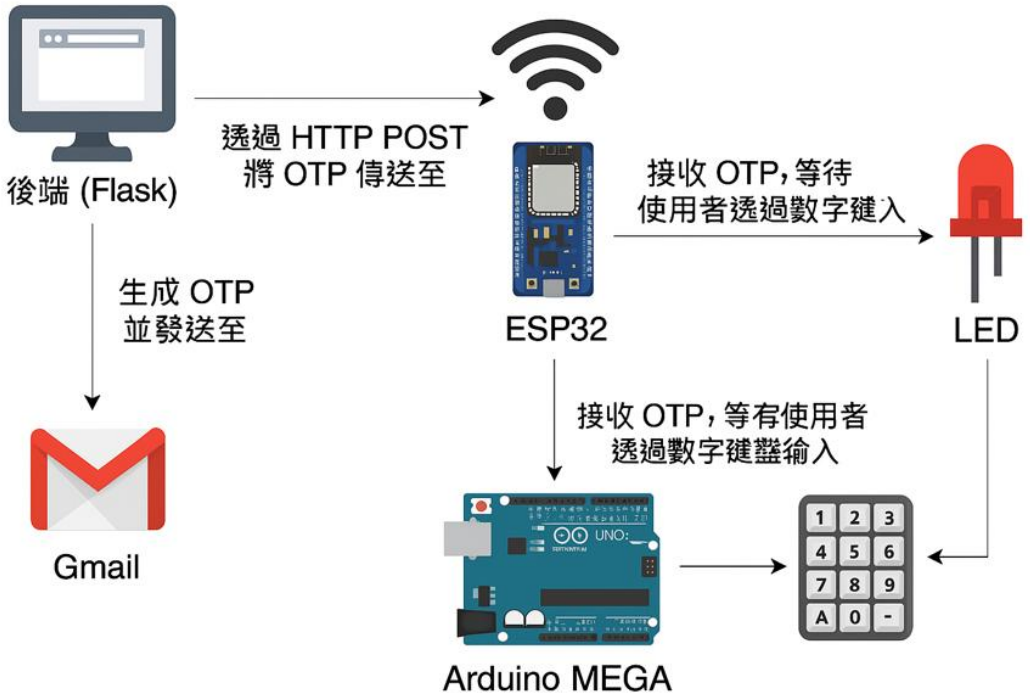


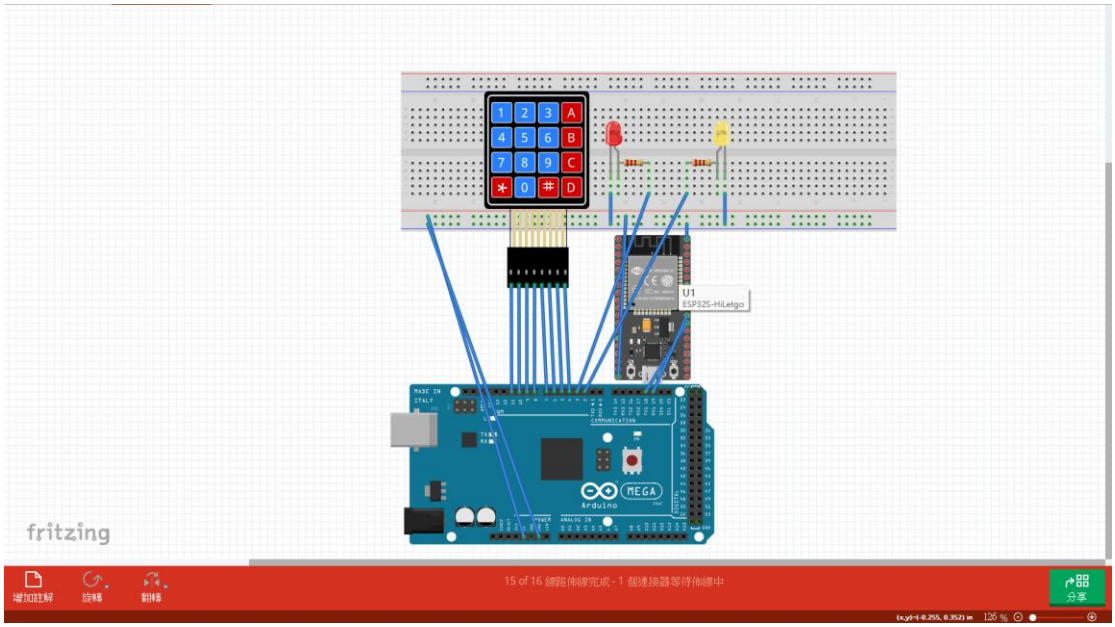


Arduino Mega	ESP32 WROOM-32E
TX1 (18)	RX2 (GPIO16)
RX1 (19)	TX2 (GPIO17)
GND	GND
5V	VIN

ESP32 WROOM 32E Pinout




後端 (Flask)
透過 HTTP POST
將 OTP 傳送至
接收 OTP，等待
使用者透過數字鍵入
生成 OTP
並發送至
ESP32
LED
接收 OTP，等有使用者
透過數字鍵盤輸入
Gmail
Arduino MEGA


fritzing