變數不能大寫。有單引號(')、換行(\n)要設為脫逸字元。~~字串變數不用留雙引號~~

★字首單引號前加空格,避免被excel去掉。

若積木只有單行程式碼,換行依據：尾部有沒有；分號

|  |  |
| --- | --- |
| \_01imi\_espgoods\_init  iMi物流模組ESP32初始化  WiFi設定  MQTT設定  UART通訊  Google試算表  \_02imi\_espgoods\_wifi  ssid: 密碼:  “ “  “ “  '//WiFi'  'char \_lwifi\_ssid[] = '+value\_ssid+';'  'char \_lwifi\_pass[] = '+value\_wifipwd+';'  \_03imi\_espgoods\_mqtt  帳號 密碼 port 服務器  “ “  “ “  8883  “ “  '//HiveMQ設定'  'const char\* mqtt\_server =' +value\_mqtt\_server+';'  'const char\* mqtt\_username =' +value\_mqtt\_user+';'  'const char\* mqtt\_password =' +value\_mqtt\_pwd+';'  'const int mqtt\_port =' +value\_port+';'  \_04imi\_espgoods\_uart  RX: TX:  **16**  **17**  '//UART通訊'  'SoftwareSerial ArduinoSerial('+value\_rx+','+value\_tx+');'  \_05imi\_espgoods\_sheet  ID 工作表名稱  “ “  “ “  '//google 試算表'  'String sheetId='+value\_sheetid+';//試算表id'  'const char\* sTag = '+value\_stag+';//工作表名稱' | #include <WiFi.h>  #define ARDUINOJSON\_DECODE\_UNICODE 1  #include <ArduinoJson.h>  #include <WiFiClientSecure.h>  #include <PubSubClient.h>  #include <SoftwareSerial.h>  //===設定變數 Start===  Statements區塊  //===設定變數 End===  String sheetTag="";  //asId不要變更  const char\* asId="AKfycbyR-Yp-uu4nIvnjvnkILaQ5AX8yFxp-UpBO-Sqs0su3ai1N\_BvQsz\_Q";  DynamicJsonDocument docSheet(2048);  //收件人資料(格式:姓名,商品,倉庫X,倉庫Y,收件人X,收件人Y)  String recipient[6];  String strRecipient="";  //Topic主題  const char\* TOPIC\_MAP\_SET = "imiRobot/map/set";  const char\* TOPIC\_CAR\_STANDBY = "imiRobot/car/standby";  const char\* TOPIC\_CAR\_GPS = "imiRobot/car/gps";  const char\* TOPIC\_GOODS\_LOAD = "imiRobot/goods/load";  const char\* TOPIC\_CAR\_LOWPOWER = "imiRobot/car/lowPower";  //UART  const char\* GOODS\_LOAD = "goodsLoad";  //發佈者 傳送的消息內容  char\* mqttSendMsg = "";  //訂閱者 接收的消息內容  String mqttGetMsg = "";  WiFiClientSecure espClient;  PubSubClient client(espClient);  // HiveMQ Cloud  static const char\* root\_ca PROGMEM = R"EOF(  -----BEGIN CERTIFICATE-----  MIIFazCCA1OgAwIBAgIRAIIQz7DSQONZRGPgu2OCiwAwDQYJKoZIhvcNAQELBQAw  TzELMAkGA1UEBhMCVVMxKTAnBgNVBAoTIEludGVybmV0IFNlY3VyaXR5IFJlc2Vh  cmNoIEdyb3VwMRUwEwYDVQQDEwxJU1JHIFJvb3QgWDEwHhcNMTUwNjA0MTEwNDM4  WhcNMzUwNjA0MTEwNDM4WjBPMQswCQYDVQQGEwJVUzEpMCcGA1UEChMgSW50ZXJu  ZXQgU2VjdXJpdHkgUmVzZWFyY2ggR3JvdXAxFTATBgNVBAMTDElTUkcgUm9vdCBY  MTCCAiIwDQYJKoZIhvcNAQEBBQADggIPADCCAgoCggIBAK3oJHP0FDfzm54rVygc  h77ct984kIxuPOZXoHj3dcKi/vVqbvYATyjb3miGbESTtrFj/RQSa78f0uoxmyF+  0TM8ukj13Xnfs7j/EvEhmkvBioZxaUpmZmyPfjxwv60pIgbz5MDmgK7iS4+3mX6U  A5/TR5d8mUgjU+g4rk8Kb4Mu0UlXjIB0ttov0DiNewNwIRt18jA8+o+u3dpjq+sW  T8KOEUt+zwvo/7V3LvSye0rgTBIlDHCNAymg4VMk7BPZ7hm/ELNKjD+Jo2FR3qyH  B5T0Y3HsLuJvW5iB4YlcNHlsdu87kGJ55tukmi8mxdAQ4Q7e2RCOFvu396j3x+UC  B5iPNgiV5+I3lg02dZ77DnKxHZu8A/lJBdiB3QW0KtZB6awBdpUKD9jf1b0SHzUv  KBds0pjBqAlkd25HN7rOrFleaJ1/ctaJxQZBKT5ZPt0m9STJEadao0xAH0ahmbWn  OlFuhjuefXKnEgV4We0+UXgVCwOPjdAvBbI+e0ocS3MFEvzG6uBQE3xDk3SzynTn  jh8BCNAw1FtxNrQHusEwMFxIt4I7mKZ9YIqioymCzLq9gwQbooMDQaHWBfEbwrbw  qHyGO0aoSCqI3Haadr8faqU9GY/rOPNk3sgrDQoo//fb4hVC1CLQJ13hef4Y53CI  rU7m2Ys6xt0nUW7/vGT1M0NPAgMBAAGjQjBAMA4GA1UdDwEB/wQEAwIBBjAPBgNV  HRMBAf8EBTADAQH/MB0GA1UdDgQWBBR5tFnme7bl5AFzgAiIyBpY9umbbjANBgkq  hkiG9w0BAQsFAAOCAgEAVR9YqbyyqFDQDLHYGmkgJykIrGF1XIpu+ILlaS/V9lZL  ubhzEFnTIZd+50xx+7LSYK05qAvqFyFWhfFQDlnrzuBZ6brJFe+GnY+EgPbk6ZGQ  3BebYhtF8GaV0nxvwuo77x/Py9auJ/GpsMiu/X1+mvoiBOv/2X/qkSsisRcOj/KK  NFtY2PwByVS5uCbMiogziUwthDyC3+6WVwW6LLv3xLfHTjuCvjHIInNzktHCgKQ5  ORAzI4JMPJ+GslWYHb4phowim57iaztXOoJwTdwJx4nLCgdNbOhdjsnvzqvHu7Ur  TkXWStAmzOVyyghqpZXjFaH3pO3JLF+l+/+sKAIuvtd7u+Nxe5AW0wdeRlN8NwdC  jNPElpzVmbUq4JUagEiuTDkHzsxHpFKVK7q4+63SM1N95R1NbdWhscdCb+ZAJzVc  oyi3B43njTOQ5yOf+1CceWxG1bQVs5ZufpsMljq4Ui0/1lvh+wjChP4kqKOJ2qxq  4RgqsahDYVvTH9w7jXbyLeiNdd8XM2w9U/t7y0Ff/9yi0GE44Za4rF2LN9d11TPA  mRGunUHBcnWEvgJBQl9nJEiU0Zsnvgc/ubhPgXRR4Xq37Z0j4r7g1SgEEzwxA57d  emyPxgcYxn/eR44/KJ4EBs+lVDR3veyJm+kXQ99b21/+jh5Xos1AnX5iItreGCc=  -----END CERTIFICATE-----  )EOF";  void reconnect() {  while (!client.connected()) {  Serial.print("Attempting MQTT connection… ");  String clientId = "ESP32Client";  if (client.connect(clientId.c\_str(), mqtt\_username, mqtt\_password)) {  Serial.println("connected!");  } else {  Serial.print("failed, rc = ");  Serial.print(client.state());  Serial.println(" try again in 5 seconds");  delay(5000);  }  }  }  //======google 試算表 函數 Start======  String URLEncode(const char\* msg)  {  const char \*hex = "0123456789abcdef";  String encodedMsg = "";  while (\*msg!=\'\\0\'){  if( (\'a\' <= \*msg && \*msg <= \'z\')  || (\'A\' <= \*msg && \*msg <= \'Z\')  || (\'0\' <= \*msg && \*msg <= \'9\') ) {  encodedMsg += \*msg;  } else {  encodedMsg += \'%\';  encodedMsg += hex[\*msg >> 4];  encodedMsg += hex[\*msg & 15];  }  msg++;  }  return encodedMsg;  }  void searchSheet(const String& fname, const String& sname){  static WiFiClientSecure sheetClient;  sheetClient.setInsecure();  const char\* host="script.google.com";  String newUrl="";  sheetClient.connect(host, 443);  while (!sheetClient.connected());  const String url = String() +"https://"+host+"/macros/s/"+asId+"/exec?type=search&sheetId="+sheetId+"&sheetTag="+sheetTag+"&fname="+fname+"&sname="+sname;  sheetClient.print("GET " + url + " HTTP/1.1\\\n"+String()+"Host: "+host+"\\\nAccept: \*/\*\\\nConnection: close\\\n\\\n\\\n");  while (!newUrl.startsWith("https:")){  newUrl = sheetClient.readStringUntil('\\\n');  if (newUrl.startsWith("Location: https://")) {  newUrl.replace("Location: ","");  break;  }  }  sheetClient.stop();  String jsonData ="";  sheetClient.connect(host, 443);  while (!sheetClient.connected());  sheetClient.print("GET " + newUrl + " HTTP/1.1\\\n"+String()+"Host: "+host+"\\\nAccept: \*/\*\\\nConnection: close\\\n\\\n\\\n");  while(!jsonData.startsWith("{")){  jsonData = sheetClient.readStringUntil('\\\n');  if (jsonData.startsWith("{")) {  DeserializationError error = deserializeJson(docSheet, jsonData);  break;  }  }  sheetClient.stop();  }  void readGoogleSheet(String field,String name){  //讀取Google試算表  sheetTag=URLEncode(sTag);//工作表名稱  searchSheet(field,URLEncode(String(name).c\_str()).c\_str());  recipient[0] = String(docSheet["A"].as<String>());//姓名  recipient[1] = String(docSheet["B"].as<String>());//商品  recipient[2] = String(docSheet["C"].as<String>());//倉庫X  recipient[3] = String(docSheet["D"].as<String>());//倉庫Y  recipient[4] = String(docSheet["E"].as<String>());//收件人X  recipient[5] = String(docSheet["F"].as<String>());//收件人Y    strRecipient="";  for(int i=0;i<6;i++){  strRecipient += recipient[i] + ",";  }  }  //======google 試算表 函數 End======  //傳送訊息：ESP32→Arduino  void UartSentToArduino(String msg){  ArduinoSerial.print(msg);  }  void pickGoods(String strX,String strY){  UartSentToArduino(strX+strY);//格式xy(例如13)  }  void setup(){  Serial.begin(9600);  ArduinoSerial.begin(9600);  //Wifi初始  WiFi.disconnect();  WiFi.softAPdisconnect(true);  WiFi.mode(WIFI\_STA);  WiFi.begin(\_lwifi\_ssid, \_lwifi\_pass);  while (WiFi.status() != WL\_CONNECTED) { delay(500); }  delay(300);    //MQTT初始  espClient.setCACert(root\_ca);  client.setServer(mqtt\_server, mqtt\_port);  client.setCallback(callback);  if (!client.connected()) {  reconnect();  }  } |
| \_06imi\_espgoods\_loop  iMi物流模組ESP32 重複區 | '//MQTT啟動'  ' client.loop();'  ' '  ' //接收訊息：Arduino→ESP32'  ' UartGetFromArduino();' |
| \_07imi\_espgoods\_submqtt  setup()  [訂閱]MQTT主題 | '//MQTT Topic訂閱'  'client.subscribe('+dropdown\_mqtt\_topic+');' |
| \_08imi\_espgoods\_mqttcallback\_func  MQTT回覆函數  中區：statements\_msg | //訂閱MQTT的主題回覆  void callback(char\* topic, byte\* payload, unsigned int length) {  mqttGetMsg = "";  for (int i = 0; i < length; i++) {  mqttGetMsg += (char)payload[i];  }  mqttGetMsg.trim();  Serial.println(String(topic));  Serial.println(mqttGetMsg);    //接收Topic主題的消息Msg  Statements區塊  } |
| \_09imi\_espgoods\_ismqtttopic  GOODS\_LOAD TOPIC\_GOODS\_LOAD  MQTT主題是 | 'String(topic) ==' +dropdown\_mqtt\_topic |
| \_10imi\_espgoods\_readsheet  讀取Google試算表 欄位為 的資料  “A“  “陳大春“ | '//google表單查詢收貨人'  'readGoogleSheet('+value\_field+','+value\_name+');' |
| \_11imi\_espgoods\_sendarduino\_pick  [傳送]訊息至Arduino:依商品xy座標 自動撿貨 | '//傳送訊息至UNO:依商品xy座標 自動撿貨'  'pickGoods(recipient[4],recipient[5]);' |
| \_12imi\_espgoods\_getarduino\_func  接收Arduino訊息函數 | //接收訊息：Arduino→ESP32  void UartGetFromArduino(){  while(ArduinoSerial.available()){  String val=ArduinoSerial.readString();  Serial.println(val);  Statements區塊  }  } |
| \_13imi\_espgoods\_isarduinomsg  類型GOODS\_LOAD GOODS\_LOAD  Arduino傳來類型是 | 'val=='+dropdown\_msg |
| \_14imi\_espgoods\_pubmqtt\_goods  [發佈]MQTT主題：GOODS\_LOAD | '//發送MQTT：TOPIC\_GOODS\_LOAD'  'const char\* msg = strRecipient.c\_str();'  'mqttSendMsg = const\_cast<char\*>(msg);'  'client.publish(TOPIC\_GOODS\_LOAD, mqttSendMsg);' |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |