123

4位

MQTT訂閱主題

**10**

RX: TX:

**10**

**10**

傳送訊息至Arduino 訊息

4位數字顯示模組 顯示數字(CLK=2,DIO=3,數值,是否補0=false)

“ “

當收到Arduino訊息時

1883

帳號: 密碼:

發佈MQTT主題 訊息

Joystick搖桿 方向是

Joystick搖桿 方向是

|  |  |  |  |
| --- | --- | --- | --- |
| 只出現1次 | 使用者自訂 | 每拖曳1次出現1次 | 使用n個**實體**感測器出現n次 |

變數：value\_rx value\_tx value\_ssid value\_wifi\_pwd value\_line\_token

value\_mqtt\_user value\_mqtt\_pwd value\_mqtt\_port value\_mqtt\_server

dropdown\_mqtt\_topic dropdown\_uart\_topic

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

|  |  |
| --- | --- |
| \_01imi\_esp32car\_init  iMi小車模組ESP32初始化  UART通訊  WiFi設定  LINE設定  MQTT設定  \_02imi\_esp32car\_init\_uart  RX: TX:  **16**  **17**  '//UART通訊'  'SoftwareSerial MegaSerial('+value\_rx+',' +value\_tx+');'  \_03imi\_esp32car\_init\_wifi  ssid: 密碼:  “ “  “ “  '//WiFi設定'  'const char\* ssid =' +value\_ssid+';'  'const char\* password =' +value\_wifi\_pwd+';'  (變數不能大寫)  \_04imi\_esp32car\_init\_line  權杖  “ “  '//LINE權杖'  'String lineToken =' +value\_line\_token+';'  \_05imi\_esp32car\_init\_mqtt  帳號 密碼 port 服務器  “ “  “ “  8883  “ “  '//---- HiveMQ設定 Start -----'  '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\_mqtt\_port+';'  脫逸字元\\\\n \\n | #include <SoftwareSerial.h>  #include <WiFi.h>  #include <PubSubClient.h>  #include <WiFiClientSecure.h>  //與Mega通訊  const char\* MAP\_SET = "mapSet";  const char\* GOODS\_LOAD = "goodsLoad";  const char\* LINE\_NOTIFY = "lineNotify";  const char\* CAR\_GPS = "carGps";  //===設定程式碼input Start===  //===設定程式碼input End===  //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";  //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);  }  }  }  void sendLineMsg(String myMsg) {  static WiFiClientSecure line\_client;  line\_client.setInsecure();  myMsg.replace("%", "%25");  myMsg.replace("&", "%26");  myMsg.replace("§", "&");  myMsg.replace("\\n", "\n");  if (line\_client.connect("notify-api.line.me", 443)) {  line\_client.println("POST /api/notify HTTP/1.1");  line\_client.println("Connection: close");  line\_client.println("Host: notify-api.line.me");  line\_client.println("Authorization: Bearer " + lineToken);  line\_client.println("Content-Type: application/x-www-form-urlencoded");  line\_client.println("Content-Length: " + String(myMsg.length()));  line\_client.println();  line\_client.println(myMsg);  line\_client.println();  line\_client.stop();  } else {  Serial.println("Line Notify failed");  }  }  void setup\_wifi() {  delay(10);  Serial.println();  Serial.print("Connecting to ");  Serial.println(ssid);  WiFi.mode(WIFI\_STA);  WiFi.begin(ssid, password);  while (WiFi.status() != WL\_CONNECTED) {  delay(500);  Serial.print(".");  }  Serial.println("");  Serial.println("WiFi connected");  Serial.println("IP address: ");  Serial.println(WiFi.localIP());  }  //傳送訊息：ESP32→Mega  void UartSentToMega(String msg) {  MegaSerial.print(msg);  }  void setup() {  delay(500);  Serial.begin(9600);  delay(500);  MegaSerial.begin(9600);  delay(500);  setup\_wifi();  //MQTT初始  espClient.setCACert(root\_ca);  client.setServer(mqtt\_server, mqtt\_port);  client.setCallback(callback);  if (!client.connected()) {  reconnect();  }    } |
| \_06imi\_esp32car\_loop  iMi小車模組ESP32 重複區 | void loop() {  //MQTT啟動  client.loop();  //接收訊息：Mega→ESP32  UartGetFromMega();  } |
| \_07imi\_esp32car\_getarduino\_func  接收Arduino訊息函數  中區：statements\_msg | //接收訊息：Mega→ESP32  void UartGetFromMega() {  while (MegaSerial.available()) {  String str = MegaSerial.readString();  Serial.println(str);  //完成送貨，傳LINE通知收貨人  if (str.indexOf(LINE\_NOTIFY) != -1) {    //雲端平台模擬GPS(格式:xy)  } else if (str.indexOf(CAR\_GPS) != -1) {    }  }//while  } |
| \_08imi\_esp32car\_submqtt  setup()  [訂閱]MQTT主題 | //MQTT Topic訂閱  client.subscribe(TOPIC\_MAP\_SET);  'client.subscribe('+dropdown\_mqtt\_topic+');\n' |
| \_09imi\_esp32car\_mqttcallback\_func  MQTT回覆函數  中區：statements\_msg | //訂閱的主題回覆  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);  //接收多個Topic主題的消息Msg  if (String(topic) == TOPIC\_MAP\_SET) {  } else if (String(topic) == TOPIC\_GOODS\_LOAD) {    }  } |
|  |  |
| \_10imi\_esp32car\_ismqtttopic  類型MAP\_SET TOPIC\_MAP\_SET  MQTT主題是 | String(topic) == TOPIC\_MAP\_SET  'String(topic) ==' +dropdown\_mqtt\_topic  (不用換行) |
| \_11imi\_esp32car\_sendarduino\_mapset  [傳送]至Arduino 類型：MAP\_SET | //儲存地圖陣列(格式:MAP\_SET,4x6地圖陣列)  UartSentToMega(String(MAP\_SET) + "," + mqttGetMsg);  delay(1000); |
| \_12imi\_esp32car\_pubmqtt\_carstandby  [發佈]MQTT主題：CAR\_STANDBY | //發送MQTT：TOPIC\_CAR\_STANDBY  mqttSendMsg = "1";  client.publish(TOPIC\_CAR\_STANDBY, mqttSendMsg); |
| \_13imi\_esp32car\_sendarduino\_goodsload  [傳送]至Arduino 類型：GOODS\_LOAD | //開始送貨(格式:GOODS\_LOAD,姓名,商品,倉庫X,倉庫Y,收件人X,收件人Y)  UartSentToMega(String(GOODS\_LOAD) + "," + mqttGetMsg); |
| 當收到Arduino訊息時 |  |
| \_14imi\_esp32car\_isarduinomsg  類型LINE\_NOTIFY LINE\_NOTIFY  Arduino傳來類型是 | str.indexOf(LINE\_NOTIFY) != -1  'str.indexOf('+dropdown\_uart\_topic+') != -1'  (不用換行) |
| \_15imi\_esp32car\_getarduino\_recipient  [接收]Arduion傳來的收貨人資料 | //收貨人資料字串轉陣列  String tmpArray[3];  char\* token = strtok((char\*)str.c\_str(), ",");  int tokenLen = 0;  while (token != NULL && tokenLen < 3) {  tmpArray[tokenLen] = token;  token = strtok(NULL, ",");  tokenLen++;  } |
| \_16imi\_esp32car\_line\_recipient  發送LINE通知收貨人 | //發送LINE  sendLineMsg(tmpArray[1] + "您好：您的商品：「" + tmpArray[2] + "」已送達，請至門口進行人臉識別簽收。"); |
| \_17imi\_esp32car\_getarduino\_gps  [接收]Arduino傳來的GPS座標 | //座標字串轉陣列  String arryGps[3];  char\* token = strtok((char\*)str.c\_str(), ",");  int tokenLen = 0;  while (token != NULL && tokenLen < 3) {  arryGps[tokenLen] = token;  token = strtok(NULL, ",");  tokenLen++;  } |
| \_18imi\_esp32car\_pubmqtt\_cargps  [發佈]MQTT主題：TOPIC\_CAR\_GPS | //發送MQTT：TOPIC\_CAR\_GPS(格式:xy)  // 使用String結合x和y的內容  String xy = arryGps[1] + arryGps[2];  // 將String轉換為const char\*  const char\* msg = (xy).c\_str();  // const char\*轉char\*  mqttSendMsg = const\_cast<char\*>(msg);  client.publish(TOPIC\_CAR\_GPS, mqttSendMsg); |