Yêu cầu: kiểm tra xem hoạt động MQTT là theo cơ chế ngắt hay Polling và xem tác dụng của hàm client.loop()

Theo tài liệu API của bộ thư viện PubSubClient:

https://pubsubclient.knolleary.net/api.html#callback

* callback : a pointer to a [message callback](https://pubsubclient.knolleary.net/api.html#callback) function called when a message arrives for a subscription created by this client.

boolean**loop**()

This should be called regularly to allow the client to process incoming messages and maintain its connection to the server.

**Returns**

* false - the client is no longer connected
* true - the client is still connected

Mỗi lần nhận subscribe được 1 chuỗi thì sẽ echo lại server

**Test 1:** bỏ client.loop() ra khỏi chương trình

kết quả: publish lên không có echo về

**Test 2**: kiểm tra xem cơ chế hoạt động

Thêm vào đầu và cuối hàm func\_state\_control hàm in ra monitor báo dấu hiệu, thêm vào các hàm delay để dễ dàng theo dõi.

Publish liên tục từ server, khi nhận được thì ESP sẽ echo về và truyền 1 frame qua UART

Nếu:

* frame truyền qua UART ngay, không bị delay thì cơ chế là ngắt
* phải có 1 khoảng delay và frame truyền nằm giữa 2 dòng “vao control” và “thoat control” thì cơ chế là Polling

void func\_state\_controll()

{

g\_next\_state = STATE\_CONTROLL;

**Serial.println("vao control");**

button\_config();

if (WiFi.status() == WL\_CONNECTED)

{

if (!client.connected())

{

Serial.println("client connected");

if (client.connect(macID))

{

client.set\_callback(callback);

client.subscribe(topic\_in);

}

}

if (client.connected())

{

**delay(200);**

client.loop();

}

send\_data\_server();

**delay(1000);**

}

else

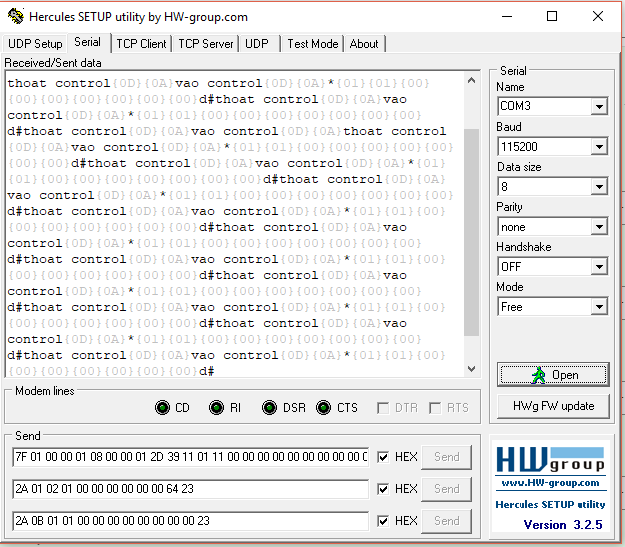
{

g\_next\_state = STATE\_CONNECT\_WIFI;

}

**Serial.println("thoat control");**

}



**Kết quả**: trường hợp 2 đúng

**Kết luận**: cơ chế hoạt động là Polling, khi nhận được dữ liệu, dữ liệu sẽ được lưu vào buffer, đợi đến khi vào hàm client.loop() nó sẽ kiểm tra buffer, nếu có dữ liệu sẽ gọi hàm callback ra thực thi.

**Test 3**: Test tác dụng của hàm client.loop()

Dựa vào đoạn code test trên nhưng tăng thời gian delay lên.

Kết quả:

|  |  |
| --- | --- |
| Thời gian delay | Kết quả nhận được |
| 1 phút | Mỗi lần func\_control thì phải kết nối server và sub lại. Pubish lên thì không đến được với ESP |
| 30s | Như trên |
| 15s | Như trên |
| 10s | Như trên |
| 5s | Như trên |
| 3s | Kết nối bình thường, nhận dữ liệu bình thường  Nếu gửi từ server liên tục thì thấy echo lại từ 3 đến 4 cái -> buffer lưu dữ liệu của MQTT khoảng từ 3 đến 4 lần 1 chuỗi json |

Kết luận:

Hàm client.loop() có chức năng kiểm tra xem buffer có dữ liệu không, nếu có thì gọi hàm callback để thực thi và còn có chức năng giữ kết nối với server (nếu trong khoảng 5s không gọi lại hàm thì sẽ bị ngắt kết nối)