

Trong bài trước, mình đã giới thiệu về giao thức MQTT và mình đã nhanh chóng test qua về MQTT trên esp8266 với arduino. Tiếp tục phần này, mình sẽ giới thiệu về giao tiếp MQTT qua tập lệnh AT

Các bạn cần đọc bài 3 và bài 5 trước thì mới hiểu bài này nhé !

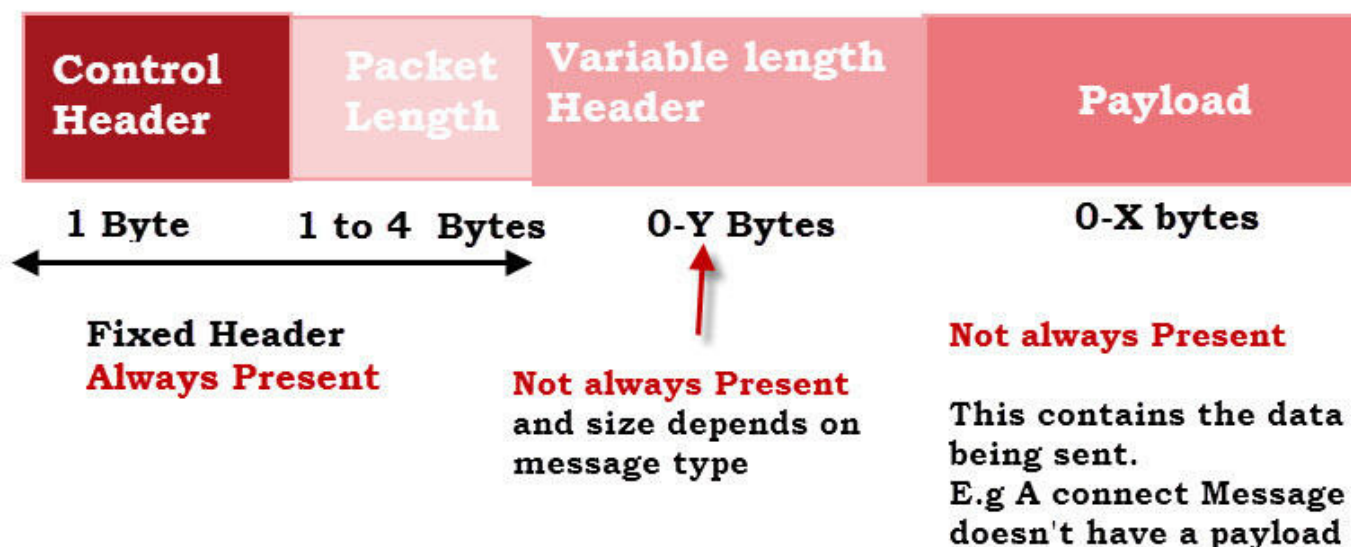
MQTT hoạt động trên nền TCP hoặc Socket, trong bài này mình sẽ chỉ demo qua trên nền tảng TCP, còn socket mình sẽ tìm hiểu ở các bài sau nhé

Cấu trúc của các gói tin MQTT

Các gói tin liên quan đến MQTT thì nhiều lắm, mình chỉ giới thiệu qua 3 gói tin cơ bản

- Gói tin connect
- Gói tin publish
- Gói tin subscriber

1, Connect - Client gửi 1 thông điệp kết nối đến broker



MQTT Standard Packet Structure

LinhKienThaoMay.Com
Zalo: 0389937723

Control Header : 1 byte

Dưới đây là bảng mô tả của **Control Header**

CONNECT	0x10
CONNACK	0x20
PUBLISH	0x30
PUBACK	0x40
PUBREC	0x50

PUBREL	0x60
PUBCOMP	0x70
SUBSCRIBE	0x80
SUBACK	0x90
UNSUBSCRIBE	0xA0
UNSUBACK	0xB0
PINGREQ	0xC0
PINGRESP	0xD0
DISCONNECT	0xE0

Packet Length (1 byte đến 4 byte)

Packet Length mô tả phía sau nó còn bao nhiêu byte nữa (không tính bản thân nó)

Trường này không cố định số lượng byte, tối đa 4 và tối thiểu 1. Bit cao nhất của byte sẽ xác định xem byte phía sau nó có thuộc Packet Length không ! Do đó chỉ có 7bit dùng để mã hóa dữ liệu

Ví dụ 1: Gói tin phía sau còn 31 byte thì ta chỉ cần 1 byte cho Packet Length
=> Packet Length = 0x1F

Ví dụ 2: Gói tin phía sau còn 321 byte thì ta sẽ cần 2 byte cho Packet Length với byte1 là byte thấp và byte2 là byte cao. Cứ mỗi giá trị của byte cao sẽ = 128 lần byte thấp. Ta tách $321 = 65 + 2 \cdot 128$

=> Packet Length = 0x41 0x02

Bit cao nhất của byte1 phải được set lên 1 để báo vẫn còn 1 byte nữa cho Packet Length nên phải sửa thành Packet Length = 0xC1 0x02

Với 4 byte Packet Length, ta sẽ mã hóa được tối đa 268,435,455 byte dữ liệu

From	To
0 (0x00)	127 (0x7F)
28 (0x80, 0x01)	16 383 (0xFF, 0x7F)
16 384 (0x80, 0x80, 0x01)	2 097 151 (0xFF, 0xFF, 0x7F)
2 097 152 (0x80, 0x80, 0x80, 0x01)	268 435 455 (0xFF, 0xFF, 0xFF, 0x7F)

Tiếp tục tới trường thứ 3 là Variable length Header, nó gồm 4 trường nhỏ sau: Protocol Name, Protocol Level, Connect Flags, and Keep Alive

- Protocol Name: (6byte) 0x00 0x04 0x4D 0x51 0x54 0x54
- Protocol Level: (1byte) 0x04
- Connect Flags: (1 byte) 0x??
- Keep Alive: (2 byte) 0x?? 0x??

Các bạn chỉ cần quan tâm tới Connect Flags

Tài liệu này được tải từ website: <http://linhkienthaomay.com>. Zalo hỗ trợ: 0389937723

Cấu trúc của nó như sau:

Bit	7	6	5	4	3	2	1	0
	User Name Flag	Password Flag	Will Retain	Will QoS		Will Flag	Clean Session	Reserved
byte 8	X	X	X	X	X	X	X	0

Bit 1 mình sẽ mặc định để bằng 1

Các bit khác các bạn xài cái nào thì set nó lên. Ví dụ mình không cần user, password thì chỉ cần 0x02 là ok, nếu set cái nào lên thì phải mô tả nó trong payload nhé

Payload là trường sẽ chứa các thông tin để phục vụ việc connect như ID_Client, User, Password. Thông tin trong trường này phải tuân thủ thứ tự sau:

Client ID -> Will Topic -> Will Message -> User Name -> Password

Chú ý: Ở trên byte Connect Flags cái nào không set thì bỏ qua nó trong Payload nhé !

Còn Keep Alive là 2 byte chứa thời gian được tính bằng giây. Đó là khoảng thời gian tối đa được phép trôi qua giữa điểm mà Client hoàn thành việc truyền một Control Packet và điểm mà nó bắt đầu gửi tiếp theo. Nói chung cứ để từ 10 đến 60s tùy các bạn

Dưới đây là 1 ví dụ về 1 gói tin connect vào broker với Client id là IOT47 và không sử dụng tên user lẫn password

0x10 0x11 0x00 0x04 0x4D 0x51 0x54 0x54 0x04 0x02 0x00 0x3C 0x00 0x05 0x49 0x4F 0x54 0x34 0x37

Trong đó:

0x10 là Control Header

0x11 = 17 là trường Packet Length báo phía sau nó có 17 byte

0x00 0x04 0x4D 0x51 0x54 0x54 là Protocol Name thuộc trường Variable length Header

0x04 là Protocol Level

0x02 là Connect Flag

0x00 0x3C là Keep Alive mình để 60 giây

0x00 0x05 0x49 0x4F 0x54 0x34 0x37 là gói Packet, với 0x05 ám chỉ phía sau nó có 5byte data. Do Connect Flag mình để 0x02 (tức không xài user name, password hay gì hết) nên thông tin ở packet chỉ cần id client là được (với id client là IOT47 = 0x49 0x4F 0x54 0x34 0x37) (id client là bắt buộc phải có nha, nó là gì cũng được tùy các bạn)

Publish - gửi 1 tin nhắn đến 1 topic

Cấu trúc như sau

1byte Control Header = 0x30
1 đến 4 byte Packet Length
1 byte 0x00
1byte chứa độ dài của topic
còn lại là nội dung của tin nhắn

Ví dụ: mình sẽ publish tin nhắn tới topic ESP8266_sent_data và nội dung là xin chào

0x30 0x1A 0x00 0x11 0x45 0x53 0x50 0x38 0x32 0x36 0x36 0x5F 0x73 0x65 0x6E 0x74
0x5F 0x64 0x61 0x74 0x61 0x78 0x69 0x6E 0x63 0x68 0x61 0x6F

Trong đó:

0x30 là Control Header
0x1A = 26 là số lượng byte phía sau
0x11 = 17 là độ dài của topic
0x45 0x53 0x50 0x38 0x32 0x36 0x36 0x5F 0x73 0x65 0x6E 0x74 0x5F 0x64 0x61 0x74
0x61 = ESP8266_sent_data là tên của topic
0x78 0x69 0x6E 0x63 0x68 0x61 0x6F = xin chào là nội dung tin nhắn

Subscriber - đăng kí nhận tin nhắn từ 1 topic

Cấu trúc như sau

1byte Control Header = 0x82
1 đến 4 byte Packet Length
1 byte 0x00 và 1 byte 0x01 (Variable header)
1 byte 0x00
1byte chứa độ dài của topic
1 byte 0x00

Ví dụ, mình sẽ đăng kí topic ESP8266_read_data

0x82 0x16 0x00 0x01 0x00 0x11 0x45 0x53 0x50 0x38 0x32 0x36 0x36 0x5F 0x72 0x65 0x61
0x64 0x5F 0x64 0x61 0x74 0x61 0x00

Trong đó:

1byte Control Header = 0x82
0x16 = 22 là số lượng byte phía sau
0x11 = 17 là độ dài của topic
0x45 0x53 0x50 0x38 0x32 0x36 0x36 0x5F 0x72 0x65 0x61 0x64 0x5F 0x64 0x61 0x74 0x61
= ESP8266_read_data là tên của topic

Mình đã giới thiệu qua 1 vài cấu trúc cơ bản để làm việc với MQTT, các bạn tự tìm hiểu thêm ở đây nhé !

<https://docs.solace.com/MQTT-311-Prtl-Conformance-Spec/MQTT%20Control%20Packet%20format.htm>

Tài liệu này được tải từ website: <http://linhkienthaomay.com>. Zalo hỗ trợ: 0389937723

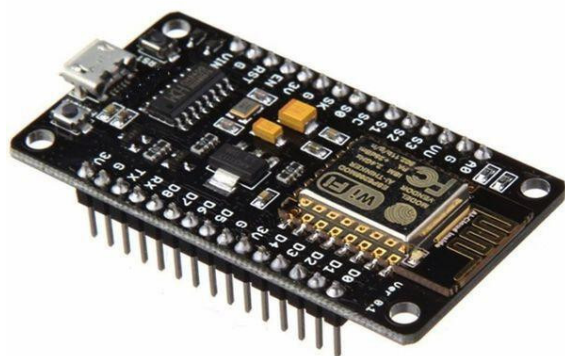
DEMO giao tiếp MQTT với tập lệnh AT Command

Trước tiên mình sẽ test bằng cách gửi thủ công bằng tay qua phần mềm Hercules luôn nhé

Các bạn chuẩn bị

- Mạch chuyển đổi USB-UART PL2303
- Laptop/PC đã cài Driver cho PL2303
- Phần mềm nạp firmware cho ESP8266

Hoặc sử dụng module esp8266 node-MCU cho nó tiện, chỉ việc cắm dây usb vào là xong, bao nhanh bao phê



LinhKienThaoMay.Com
Zalo: 0389937723

Nhắc lại 1 vài lệnh cơ bản ở bài 2

```
AT+CWJAP="IOT47","12345678"<CR><LF> //kết nối vào wifi nhà bạn
AT+CWMODE=1<CR><LF> // yêu cầu module hoạt động ở chế độ Station/Client
AT+CIPMUX=0<CR><LF>
ATE0<CR><LF> //tắt chế độ phản hồi ngửa mắt
```

Các bạn gọi các lệnh cơ bản phía trên để khởi tạo module trước nhé ! <CR><LF> là 2 byte 0x0D 0x0A hay rn đấy nhé ! Trên phần mềm Hercules thì là \$0D\$0A

Còn đây là lệnh cho giao thức TCP mà mình sẽ dùng để phục vụ cho các kết nối MQTT

```
AT+CIPSTART="TCP","yourserver.com",80<CR><LF> // khởi động 1 kết nối TCP đến server nào đó ở 1 port nào đó (ví dụ ở đây là cổng 80)
AT+CIPSEND=X<CR><LF> // bắt đầu gửi 1 gói tin TCP với độ dài X
```

Ở demo này, mình kết nối tới broker MQTT là broker.hivemq.com và cổng sử dụng là 1883 - đây là cổng chuyên dụng cho các kết nối TCP. Do broker.hivemq.com là broker free không bảo mật nên mình không cần user và password gì hết ☐

Các bạn có thể thử với mã html mình đã viết và hướng dẫn ở bài 7 hoặc mình sẽ show giao diện web đó trực tiếp ở đây cho các bạn test luôn

Demo MQTT

Điều khiển thiết bị qua WIFI - MQTT

Tin nhắn từ esp8266: ...

Bat 1 Tat 1

Bat 2 Tat 2

Bat 3 Tat 3

Bat 4 Tat 4

Note pad của mình đây nhé:

```
//kết nối tới broker mqtt
```

```
AT+CIPSTART="TCP","broker.hivemq.com",1883$0D$0A
```

```
// kết nối tới broker
```

```
AT+CIPSEND=19$0D$0A
```

```
$10$11$00$04$4D$51$54$54$04$02$00$3C$00$05$49$4F$54$34$37
```

```
//gửi tin nhắn xin chào tới topic ESP8266_sent_data
```

```
AT+CIPSEND=28$0D$0A
```

```
$30$1A$00$11$45$53$50$38$32$36$36$5F$73$65$6E$74$5F$64$61$74$61$78$69$6E$63$68$61$6F
```

```
//đăng kí nhận tin nhắn từ topic ESP8266_read_data
```

```
AT+CIPSEND=24$0D$0A
```

```
$82$16$00$01$00$11$45$53$50$38$32$36$36$5F$72$65$61$64$5F$64$61$74$61$00
```

Lập trình giao tiếp Arduino với esp8266 qua tập lệnh AT - giao thức MQTT điều khiển 4 thiết bị

Kết nối

Arduino ESP8266

3.3V 3.3V

GND GND

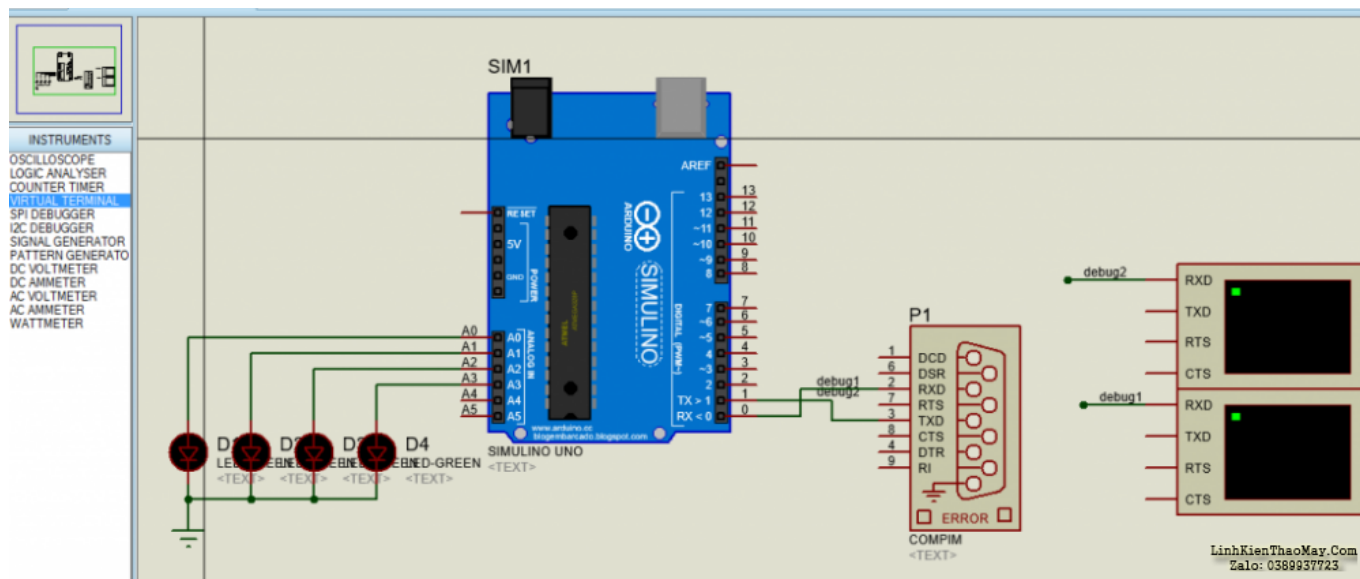
RX TX

TX RX

Giống như bài 3, mình sẽ sử dụng phần mềm mô phỏng proteus để giao tiếp arduino ảo với

Tài liệu này được tải từ website: <http://linhkienthaomay.com>. Zalo hỗ trợ: 0389937723

esp8266 thật thông qua cổng COMPIM (nghèo quá không có tiền mua arduino ☹)



Sơ đồ kết nối

Mình thêm 2 cái virtul terminal để debug dữ liệu vào ra trên cổng UART

Lập trình

Hàm ESP8266_SendCommand có nhiệm vụ gửi 1 AT command tới cho esp8266 và chờ cho tới khi trả về value, mình cũng cho thêm 1 cái timeout để thoát ra khi hết thời gian chờ

C

```
//hàm này có nhiệm vụ gửi 1 lệnh AT đến
```

```
bool ESP8266_SendCommand(
```

```
{
```

```
String rx_data="";
```

```
while(Serial.available())Serial.read();
```

```
Serial.print(cmd);  
// gửi lệnh AT
```

```
unsigned long t=millis();
```

```
while(1)
```



```

{
if((millis() - t) >
timeout)return
if
(Serial.available())
{
char inChar =
(char)Serial.read();
rx_data+=inChar;
if(rx_data.indexOf(v
alue) != -1)return
}
}
}
}

```

```

1 //hàm này có nhiệm vụ gửi 1 lệnh AT đến cho esp8266 và chờ xem nó có phản hồi đúng
2 data về không
3 bool ESP8266_SendCommand(String cmd,String value,int timeout)
4 {
5 String rx_data="";
6 while(Serial.available())Serial.read(); //xóa sạch bộ đệm RX
7 Serial.print(cmd); // gửi lệnh AT đi
8 unsigned long t=millis();
9 while(1)
10 {
11 if((millis() - t) > timeout)return false; //nếu tới time out thì thoát
12 if (Serial.available() > 0)
13 {
14 char inChar = (char)Serial.read();
15 rx_data+=inChar;
16 if(rx_data.indexOf(value) != -1)return true;
17 }
18 }
19 }

```

Hàm ESP82666_init sẽ khởi tạo module esp8266 và trả về err_code nếu gặp phải lỗi

C


```
int ESP8266_init()
```

```
{
```

```
int i=0;
```

```
while(1)//GÕI LỆNH  
AT
```

```
{
```

```
if(ESP8266_SendComma  
nd("ATrn","OK",500)
```

```
i++; if(i==10)return  
1; //trở về false
```

```
}
```

```
//1 s setup cơ bản
```

```
delay(100);Serial.pr  
int("ATE0rn"); //tắt
```

```
delay(100);Serial.pr  
int("AT+CWMODE=1rn")
```

```
delay(100);Serial.pr  
int("AT+CIPMUX=0rn")
```

```
delay(100);
```

```
String data_connect  
= "AT+CWJAP="" +
```

```
if(ESP8266_SendComma  
nd(data_connect,"OK"
```

```
return 0;
```

```
}
```

```

int ESP8266_init()
1 {
2   int i=0;
3
4   while(1)//GỬI LỆNH AT
5   {
6     if(ESP8266_SendCommand("ATrn","OK",500) == true)break; // liên tục gửi lệnh AT
7   xem esp8266 có hoạt động không
8     i++; if(i==10)return 1; //trả về false báo init thất bại
9   }
10
11 //1 số setup cơ bản
12 delay(100);Serial.print("ATE0rn"); //tắt phản hồi
13 delay(100);Serial.print("AT+CWMODE=1rn"); //chế độ station
14 delay(100);Serial.print("AT+CIPMUX=0rn"); //chế độ đơn kênh
15
16 delay(100);
17 String data_connect = "AT+CWJAP=" + ssid + "," + pass + "rn";
18 if(ESP8266_SendCommand(data_connect,"OK",10000) == false)return 2; //kết nối vào
19 wifi, timeout 10 giây
20 return 0;
  }

```

Tiếp tục là hàm khởi tạo giao thức MQTT

C

void MQTT_init()
{
Connect_packet[13] = ID_Client.length();
Connect_packet[1] = Connect_packet[13] +
Connect_Packet_Length =
}
int MQTT_broker_connect(
{

```
MQTT_init();

int i=10;

while(i--)

{

if(ESP8266_SendCommand("AT+CIPSTART="TCP

{

String CIPSEN =
"AT+CIPSEND=" +

Serial.print(CIPSEN)
;

delay(50);

//gửi gói tin
connect đi

for(int
i=0;i<14;i++)Serial.

Serial.print(ID_Client);

ESP8266_SendCommand(
"", "SEND OK", 3000);

return 0;

}

}

return 1; //không
thành công tới

}
```

```

1 void MQTT_init()
2 {
3   Connect_packet[13] = ID_Client.length(); //byte 14 ghi độ dài của id
4   Connect_packet[1] = Connect_packet[13] + 12; //byte 1 ghi Packet Length
5   Connect_Packet_Length = Connect_packet[1] + 2;
6 }
7 int MQTT_broker_connect()
8 {
9   MQTT_init();
10  int i=10;
11  while(i--)
12  {
13    if(ESP8266_SendCommand("AT+CIPSTART="TCP","broker.hivemq.com",1883rn","CONNECT",3000)
14    == true) //kết nối tcp tới broker
15    {
16      String CIPSEN = "AT+CIPSEND=" + String(Connect_Packet_Length) + "rn";
17      Serial.print(CIPSEN);
18      delay(50);
19      //gửi gói tin connect đi
20      for(int i=0;i<14;i++)Serial.write(Connect_packet[i]);
21      Serial.print(ID_Client);
22      ESP8266_SendCommand("", "SEND OK",3000); //chờ phản hồi SEND OK
23      return 0;
24    }
25  }
26  return 1; //không thể kết nối tới server

```

FULL code arduino, các bạn tự ngâm cứu nhé

C

```
String ssid
="Tang5";
```

```
String pass
="99999999";
```

```
String
ID_Client="IOT47";
```

```
unsigned char
Connect_packet[14]={
```

```
int
Connect_Packet_Lengt
```

```
//hàm này có nhiệm
vụ gửi 1 lệnh AT đến
```

```
bool
ESP8266_SendCommand(
```

```
{  
  
String rx_data="";  
  
while(Serial.available()  
Serial.read());  
  
Serial.print(cmd);  
    // gửi lệnh AT  
  
unsigned long  
t=millis();  
  
while(1)  
  
{  
  
if((millis() - t) >  
timeout)return  
  
if  
(Serial.available()  
{  
  
char inChar =  
(char)Serial.read();  
  
rx_data+=inChar;  
  
if(rx_data.indexOf(v  
alue) != -1)return  
  
}  
  
}  
  
}  
  
int ESP8266_init()  
  
{  
  
int i=0;  
  
  
  
while(1)//GỬI LỆNH  
AT  
  
{
```

```
if(ESP8266_SendCommand("ATrn","OK",500)
```

```
i++; if(i==10)return  
1; //trở về false
```

```
}
```

```
//1 s setup cơ bản
```

```
delay(100);Serial.pr  
int("ATE0rn"); //tắt
```

```
delay(100);Serial.pr  
int("AT+CWMODE=1rn")
```

```
delay(100);Serial.pr  
int("AT+CIPMUX=0rn")
```

```
delay(100);
```

```
String data_connect  
= "AT+CWJAP=" +
```

```
if(ESP8266_SendComm  
and(data_connect,"OK"
```

```
return 0;
```

```
}
```

```
void  
ESP8266_pub(String
```

```
{
```

```
}
```

```
void  
ESP8266_sub(String
```

```
{  
//size
```

```
unsigned char  
Sub_packet[6]={0x82,
```

```
//tính độ dài gói  
packet
```

```
Sub_packet[5] =  
topic.length();
```

```
Sub_packet[1] = 4 +  
Sub_packet[5] + 1;
```

```
int  
Sub_packet_length =  
  
String CIPSEN =  
"AT+CIPSEND=" +  
  
Serial.print(CIPSEN)  
;  
  
delay(50);
```

```
//gửi gói tin sub đi
```

```
for(int  
i=0;i<6;i++)Serial.w
```

```
Serial.print(topic);
```

```
Serial.write(0x00);
```

```
ESP8266_SendCommand(  
"", "SEND OK", 3000);
```

```
}
```

```
void MQTT_init()
```

```
{
```

```
Connect_packet[13] =  
ID_Client.length();
```

```
Connect_packet[1] =  
Connect_packet[13] +
```

```
Connect_Packet_Lengt  
h =
```

```
}
```

```
int  
MQTT_broker_connect(  

```

```
{
```

```
MQTT_init();
```

```
int i=10;
```

```
while(i--)
```

```
{
```



```
if(ESP8266_SendCommand("AT+CIPSTART="TCP
{
String CIPSEN =
"AT+CIPSEND=" +
Serial.print(CIPSEN)
;
delay(50);
//gửi gói tin
connect đi
for(int
i=0;i<14;i++)Serial.
Serial.print(ID_Client);
ESP8266_SendCommand(
"", "SEND OK", 3000);
return 0;
}
}
return 1; //không
thành kết nối tới
}
void
MQTT_reconnect()
{
}
void setup()
{
pinMode(A0,OUTPUT);digitalWrite(A0,LOW);
pinMode(A1,OUTPUT);digitalWrite(A1,LOW);
pinMode(A2,OUTPUT);digitalWrite(A2,LOW);
```

```
pinMode(A3,OUTPUT);d  
igitalWrite(A3,LOW);
```

```
delay(500);
```

```
Serial.begin(9600);
```

```
int err_code = 0;
```

```
err_code=ESP8266_ini  
t();
```

```
if(err_code != 0)
```

```
{
```

```
/*
```

```
* err_code=1 lỗi kết  
nối
```

```
* err_code=2 không  
thể truy cập vào
```

```
*/
```

```
Serial.print("Error  
code=");Serial.print
```

```
while(1);
```

```
}
```

```
int  
err_code_connect=MQT
```

```
if(err_code != 0)
```

```
{
```

```
/*
```

```
* err_code_connect=1  
//không thể kết nối
```

```
*
```

```
*/
```

```
Serial.print("err_co  
de_connect=");Serial
```



```
.println(err_code_connect);
```

```
while(1);
```

```
}
```

```
ESP8266_sub("ESP8266  
_read_data"); //đăng
```

```
}
```

```
String  
inputString="";
```

```
void loop()
```

```
{
```

```
if  
(Serial.available()
```

```
{
```

```
delay(100);
```

```
while(Serial.availab  
le() > 0) //data bít
```

```
{
```

```
char inChar =  
(char)Serial.read();
```

```
if(inChar == 0)  
break;
```

```
}
```

```
inputString="";
```

```
while(Serial.availab  
le() > 0)
```

```
{
```

```
char inChar =  
(char)Serial.read();
```

```
inputString+=inChar;
```

```
}
```

```
int topic_length =  
inputString[0];
```

```
String topic =  
inputString.substrin
```

```
String mess =  
inputString.substrin
```

```
if(mess == "Bat  
1")digitalWrite(A0,H
```

```
if(mess == "Tat  
1")digitalWrite(A0,L
```

```
if(mess == "Bat  
2")digitalWrite(A1,H
```

```
if(mess == "Tat  
2")digitalWrite(A1,L
```

```
if(mess == "Bat  
3")digitalWrite(A2,H
```

```
if(mess == "Tat  
3")digitalWrite(A2,L
```

```
if(mess == "Bat  
4")digitalWrite(A3,H
```

```
if(mess == "Tat  
4")digitalWrite(A3,L
```

```
}
```

```
}
```

```
1
2 String ssid ="Tang5";
3 String pass ="99999999";
4 String ID_Client="IOT47";
5
6 unsigned char Connect_packet[14]={0x10, 0xFF, 0x00, 0x04, 0x4D, 0x51, 0x54, 0x54, 0x04, 0x02, 0x00, 0x3C, 0x00, 0xFF};
7 int Connect_Packet_Length;
8 //hàm này có nhiệm vụ gửi 1 lệnh AT đến cho esp8266 và chờ xem nó có phản hồi đúng data về không
9 bool ESP8266_SendCommand(String cmd,String value,int timeout)
10 {
11 String rx_data="";
12 while(Serial.available())Serial.read(); //xóa sạch bộ đệm RX
13
14 Serial.print(cmd); // gửi lệnh AT đi
15 unsigned long t=millis();
16 while(1)
17 {
18 if((millis() - t) > timeout)return false; //nếu quá thời gian thì thoát
19 if (Serial.available() > 0)
20 {
21 char inChar = (char)Serial.read();
22 rx_data+=inChar;
23 if(rx_data.indexOf(value) != -1)return true;
24 }
25 }
26 }
27 int ESP8266_init()
28 {
29 int i=0;
30
31 while(1)//GỬI LỆNH AT
32 {
33 if(ESP8266_SendCommand("ATrn","OK",500) == true)break; // liên tục gửi lệnh AT xem esp8266 có hoạt động không
34 i++; if(i==10)return 1; //trả về false báo init thất bại
35 }
36
37 //1 số setup cơ bản
38 delay(100);Serial.print("ATE0rn"); //bắt phản hồi
39 delay(100);Serial.print("AT+CWMODE=1rn"); //chế độ station
40 delay(100);Serial.print("AT+CIPMUX=0rn"); //chế độ đơn kênh
41
42 delay(100);
43 String data_connect = "AT+CWJAP="+ ssid + " "" + pass + ""rn";
44 if(ESP8266_SendCommand(data_connect,"OK",10000) == false)return 2; //kết nối vào wifi, timeout 10 giây
45 return 0;
46 }
47 void ESP8266_pub(String topic,String mess) //max topic length + mess =127
48 {
49
50 }
51 void ESP8266_sub(String topic) //max topic length =255
52 {
53 //size size
54 unsigned char Sub_packet[6]={0x82, 0xFF, 0x00, 0x01, 0x00, 0xFF};
55 //tính độ dài gói packet
56 Sub_packet[5]= topic.length();
57 Sub_packet[1]= 4 + Sub_packet[5] + 1;
58
59 int Sub_packet_length = Sub_packet[1] + 2;
60 String CIPSEN = "AT+CIPSEND="+ String(Sub_packet_length) + "rn";
61 Serial.print(CIPSEN);
62 delay(50);
63 //gửi gói tin sub đi
64 for(int i=0;i<6;i++)Serial.write(Sub_packet[i]);
65 Serial.print(topic);
66 Serial.write(0x00);
67 ESP8266_SendCommand("", "SEND OK",3000); //chờ phản hồi SEND OK
68 }
69 void MQTT_init()
70 {
71 Connect_packet[13] = ID_Client.length(); //byte 14 ghi độ dài của id
72 Connect_packet[1] = Connect_packet[13] + 12; //byte 1 ghi Packet Length
73 Connect_Packet_Length = Connect_packet[1] + 2;
74 }
75 int MQTT_broker_connect()
76 {
77 MQTT_init();
78 int i=10;
79 while(i--)
80 {
81 if(ESP8266_SendCommand("AT+CIPSTART="TCP","broker.hivemq.com",1883rn","CONNECT",3000) == true) //kết nối tcp tới broker
82 {
83 String CIPSEN = "AT+CIPSEND="+ String(Connect_Packet_Length) + "rn";
84 Serial.print(CIPSEN);
85 delay(50);
86 //gửi gói tin connect đi
87 for(int i=0;i<14;i++)Serial.write(Connect_packet[i]);
88 Serial.print(ID_Client);
89 ESP8266_SendCommand("", "SEND OK",3000); //chờ phản hồi SEND OK
90 return 0;
91 }
92 }
93 return 1; //không thể kết nối tới server
94 }
95 void MQTT_reconnect()
96 {
97
98 void setup()
99 {
100 pinMode(A0,OUTPUT);digitalWrite(A0,LOW);
101 pinMode(A1,OUTPUT);digitalWrite(A1,LOW);
102 pinMode(A2,OUTPUT);digitalWrite(A2,LOW);
103 pinMode(A3,OUTPUT);digitalWrite(A3,LOW);
104 delay(500);
105 Serial.begin(9600);
106
107 int err_code = 0;
108 err_code=ESP8266_init();
109 if(err_code != 0)
110 {
111 /*
112 * err_code=1 lỗi kết nối
113 * err_code=2 không thể truy cập vào wifi
114 */
115 Serial.print("Error code=");Serial.println(err_code);
116 while(1);
117 }
118 int err_code_connect=MQTT_broker_connect(); // khởi tạo giao thức MQTT
119 if(err_code != 0)
120 {
121 /*
122 * err_code_connect=1 //không thể kết nối tới server
123 *
124 */
125 Serial.print("err_code_connect=");Serial.println(err_code_connect);
126 while(1);
127 }
128 ESP8266_sub("ESP8266_read_data"); //đăng kí nhận mess từ topic
129 }
130
131 String inputString="";
132 void loop()
133 {
134 if (Serial.available() > 0) //nhận tin nhắn từ topic
135 {
136 delay(100);
137 while(Serial.available() > 0) //data bắt đầu từ byte 0x00 , byte tiếp theo chưa độ dài của topic
138 {
139 char inChar = (char)Serial.read();
140 if(inChar == 0) break;
141 }
142 inputString="";
143 while(Serial.available() > 0)
144 {
145 char inChar = (char)Serial.read();
146 inputString+=inChar;
147 }
148 int topic_length = inputString[0];
149 String topic = inputString.substring(1,1+topic_length); //lấy được tên topic
150 String mess = inputString.substring(1+topic_length); //lấy được tên mess
151
152 if(mess == "Bat 1")digitalWrite(A0,HIGH);
153 if(mess == "Tat 1")digitalWrite(A0,LOW);
154 if(mess == "Bat 2")digitalWrite(A1,HIGH);
155 if(mess == "Tat 2")digitalWrite(A1,LOW);
156 if(mess == "Bat 3")digitalWrite(A2,HIGH);
157 if(mess == "Tat 3")digitalWrite(A2,LOW);
158 if(mess == "Bat 4")digitalWrite(A3,HIGH);
159 if(mess == "Tat 4")digitalWrite(A3,LOW);
160 }
161 }
162 }
```

Demo mô phỏng trên proteus

Chú ý: Các bạn tự viết thêm hàm publish tương tự nhé, và phải thường xuyên gửi gói tin keep alive để giữ kết nối với broker

Download

TRUNG TÂM SỬA CHỮA ĐIỆN TỬ QUẢNG BÌNH

MR. XÔ - 0901.679.359 - 80 Võ Thị Sáu, Phường Quảng Thuận, tx Ba Đồn, tỉnh Quảng Bình

GIÁ RẺ

NHANH CHÓNG

LINH KIẾN CHÍNH HÃNG



TRUNG TÂM SỬA CHỮA ĐIỆN TỬ XÔ NGUYỄN

- Dịch vụ sửa chữa điện tử tại nhà
- Cung cấp linh kiện điện tử
- Tư vấn lắp đặt nhà thông minh

Đc: Quảng Thuận. tx Ba Đồn,
tỉnh Quảng Bình - 0901.679.359

SANYO ELEC MSUNG

Panasonic TOSHIBA BISHI

Các bạn có thể tải về project tại đây

<https://drive.google.com/open?id=1Pi-MqD1HbaRWmgQhB6Kbmwe4dpg6SDjf>

Em chào anh! Tập lệnh AT commands có hỗ trợ AT+MQTT, anh đã từng làm qua chưa ạ? Em đã test nhưng hiện vẫn đang lỗi. Với cả trong phần anh làm, tại sao phải đổi ASCII sang HEX ở một số chỗ ạ?

Các bài viết tương tự:

1. [\[IoT\] Bài 4: Giới thiệu ngôn ngữ HTML và mô hình http request/reponse](#)
2. [\[IoT\] Bài 7: ESP8266 - arduino ide và giao thức MQTT](#)
3. [\[IoT\] Bài 9: Tìm hiểu giao thức TCP và UDP](#)
4. [Bài 1: Làm quen với module WiFi Esp8266](#)
5. [Bài 2: Giao tiếp với module WiFi ESP8266 bằng tập lệnh AT](#)
6. [Bài 6: Demo hệ thống theo dõi, giám sát thông số môi trường](#)
7. [Bếp MD c18f21 - Em nay bị bệnh bẫy nguồn lên kêu bip bip 30s rồi tắt.kt nguồn 18v,5v oki,kt và thay máy em 8550 với 8050 ,ic LM339 ,cam biến tốt mà vẫn ốm.](#)

Tài liệu này được tải từ website: <http://linhkienthaomay.com>. Zalo hỗ trợ: 0389937723

8. [điều hòa sumikura - báo lỗi F4](#)
9. [gtu - utu](#)
10. [Mạch sạc pin 18650 chống ngược cực](#)
11. [máy hàn que điện tử 2 pha - do khách cắm điện 1 pha nên em nó bốc khói](#)
12. [tivi Darling 21F93U tổng 11106 - con tụ 10mi 250v gần sò ngan, bật máy chạy chừng 1 phút rất nóng phù đầu, mong các bác chỉ giáo em mới vào nghề cong yếu quá](#)