目前我ESP01S支持的指令：

* +CMD:70,"AT+MQTTUSERCFG",0,0,1,0
* +CMD:71,"AT+MQTTCLIENTID",0,0,1,0
* +CMD:72,"AT+MQTTUSERNAME",0,0,1,0
* +CMD:73,"AT+MQTTPASSWORD",0,0,1,0
* +CMD:74,"AT+MQTTCONNCFG",0,0,1,0
* +CMD:75,"AT+MQTTCONN",0,1,1,0
* +CMD:76,"AT+MQTTPUB",0,0,1,0
* +CMD:77,"AT+MQTTPUBRAW",0,0,1,0
* +CMD:78,"AT+MQTTSUB",0,1,1,0
* +CMD:79,"AT+MQTTUNSUB",0,0,1,0
* +CMD:80,"AT+MQTTCLEAN",0,0,1,0

官方文档介绍的MQTTAT指令：

* • AT+MQTTUSERCFG：设置 MQTT 用户属性
* • AT+MQTTLONGCLIENTID：设置 MQTT 客户端 ID
* • AT+MQTTLONGUSERNAME：设置 MQTT 登陆用户名
* • AT+MQTTLONGPASSWORD：设置 MQTT 登陆密码
* • AT+MQTTCONNCFG：设置 MQTT 连接属性
* • AT+MQTTCONN：连接 MQTT Broker
* • AT+MQTTPUB：发布 MQTT 消息（字符串）
* • AT+MQTTPUBRAW：发布 MQTT 消息（二进制）
* • AT+MQTTSUB：订阅 MQTT Topic
* • AT+MQTTUNSUB：取消订阅 MQTT Topic
* • AT+MQTTCLEAN：断开 MQTT 连接

（每条信息要确认相应回复）

（可以写一个通用函数，参数有要发送的和要检测的响应，n次t内没有成功就重复发送，超过n次就报错需人工干预）

（人工干预报错函数，提示需要人工干预）

（发给ESP8266的AT指令需要在末尾换行）

1. **开启消息回显时各种消息交流情况**
   1. **收到乱码消息**
      1. **收**
         1. **?**
         2. **JJJJ\*J峛!迤Cc椅?筷#?#a壟Jrd$湠|刲?<  
            ?l?**
         3. **c|弬?況?#?c匊gn焏og溾?c8膰lsd;lx鹢?\_x0010\_  
            ?$**
   2. **上电默认消息**
      1. **################################################  
         arch:ESP8266, 1  
         compile\_time:Ai-Thinker|B&T  
         wifi\_mac:dc4F226750F8  
         sdk\_version:v3.4-22-g967752e2  
         firmware\_version:2.2.0  
         compile\_time:Jul 31 2021 11:41:32  
           
         ready  
           
         ################################################**
   3. **没有连上WiFi时AT+CWJAP?，不管有无WIFI连接配置**
      1. **发**
         1. **AT+CWJAP?**
      2. **收，成功**
         1. **AT+CWJAP?  
              
            OK**
   4. **自动连接上WiFi时收到消息**
      1. **WIFI CONNECTED**
      2. **WIFI GOT IP**
   5. **没有MQTT连接配置时，WIFI断联通知**
      1. **收**
         1. **WIFI DISCONNECT**
   6. **连上WiFi时AT+CWJAP?**
      1. **发**
         1. **AT+CWJAP?**
      2. **收，成功**
         1. **AT+CWJAP?  
            +CWJAP:"{SSID}","ae:40:5a:6f:ce:b3",6,-44,0,1,3,0,0  
              
            OK**
   7. **配置WiFi连接AT+CWJAP**
      1. **发**
         1. **AT+CWJAP="321","111000111"**
      2. **收1**
         1. **AT+CWJAP="321","111000111"**
      3. **收2，如果曾经有WIFI连接**
         1. **WIFI DISCONNECTED**
      4. **收3**
         1. **WIFI CONNECTED**
      5. **收4**
         1. **WIFI GOT IP**
      6. **收5，不确定是否相连**
         1. **OK**
   8. **配置MQTT用户配置**
      1. **发**
         1. **AT+MQTTUSERCFG=0,1,"NULL","{DEVICE\_ID}","{MQTT\_PWD}",0,0,""**
      2. **收，成功**
         1. **AT+MQTTUSERCFG=0,1,"NULL","{DEVICE\_ID}","{MQTT\_PWD}",0,0,""  
              
            OK**
   9. **配置MQTT用户端ID**
      1. **发**
         1. **AT+MQTTCLIENTID=0,"{CLIENT\_ID}"**
      2. **收，成功**
         1. **AT+MQTTCLIENTID=0,"{CLIENT\_ID}"  
              
            OK**
   10. **配置MQTT连接**
       1. **发**
          1. **AT+MQTTCONN=0,"{HOSTNAME}",1883,1**
       2. **收，成功**
          1. **AT+MQTTCONN=0,"{HOSTNAME}",1883,1  
             +MQTTCONNECTED:0,1,"{HOSTNAME}","1883","",1  
               
             OK**
       3. **收，失败**
          1. **AT+MQTTCONN=0,"{HOSTNAME}",1883,1  
               
             ERROR**
   11. **配置MQTT订阅**
       1. **发**
          1. **AT+MQTTSUB=0,"$oc/devices/{DEVICE\_ID}/sys/commands/#",0**
       2. **收**
          1. **AT+MQTTSUB=0,"$oc/devices/{DEVICE\_ID}/sys/commands/#",0  
               
             OK**
   12. **检查MQTT连接**
       1. **发**
          1. **AT+MQTTCONN?**
       2. **收，有MQTT配置，但不管有无WIFI连接上**
          1. **AT+MQTTCONN?  
             +MQTTCONN:0,6,1,"{HOSTNAME}","1883","",1  
               
             OK**
       3. **收，无MQTT配置**
          1. **AT+MQTTCONN?  
             +MQTTCONN:0,0,0,"","","",0  
               
             OK**
   13. **WiFi断联，已经配置MQTT，重复响应**
       1. **收**
          1. **+MQTTDISCONNECTED:0**
   14. **收到下行命令**
       1. **收**
          1. **+MQTTSUBRECV:0,"$oc/devices/{DEVICE\_ID}/sys/commands/request\_id={request\_id}",63,{"paras":{"{para\_name}":{para\_value}},"service\_id":"All","command\_name":"WPVRC"}**
   15. **发送上行响应**
       1. **发**
          1. **AT+MQTTPUB=0,"$oc/devices/{DEVICE\_ID}/sys/commands/response/request\_id={request\_id}","{}",0,1**
       2. **收，成功发送**
          1. **AT+MQTTPUB=0,"$oc/devices/{DEVICE\_ID}/sys/commands/response/request\_id={request\_id}","{}",0,1  
               
             OK**
   16. **发送属性上报**
       1. **发**
          1. **AT+MQTTPUB=0,"$oc/devices/AQAQ25032901/sys/properties/report","{\"services\":[{\"service\_id\":\"All\"\,\"properties\":{\"ISV\":3\,\"ALVR\":3\,\"PGLVR\":3\,\"FRS\":1\,\"AT\":1\,\"AH\":1}}]}",0,1**
       2. **收**
          1. **AT+MQTTPUB=0,"$oc/devices/AQAQ25032901/sys/properties/report","{\"services\":[{\"service\_id\":\"All\"\,\"properties\":{\"ISV\":3\,\"ALVR\":3\,\"PGLVR\":3\,\"FRS\":1\,\"AT\":1\,\"AH\":1}}]}",0,1  
               
             OK**
2. **ESP8266发给STM32的所有项目涉及的消息**
   1. **ESP8266上电消息(最后一行无\r\n，其他都有)  
        
      ################################################  
      arch:ESP8266, 1  
      compile\_time:Ai-Thinker|B&T  
      wifi\_mac:dc4F226750F8  
      sdk\_version:v3.4-22-g967752e2  
      firmware\_version:2.2.0  
      compile\_time:Jul 31 2021 11:41:32  
        
      ready  
        
      ################################################**
      1. **关键词："ESP"**
   2. **成功响应  
      \r\nOK\r\n**
      1. **关键词："OK"**
   3. **错误响应  
      \r\nERROR\r\n**
      1. **关键词："ERR"**
   4. **WiFi连接**
      1. **（开头前无换行，其他都有）  
         WIFI CONNECTED**
         1. **关键词："FI CO"**
      2. **（开头前无换行，其他都有）  
         WIFI GOT IP  
           
         //有时没有GOT IP**
         1. **关键词："FI GO"**
   5. **WiFi断联**
      1. **（开头前无换行，其他都有）  
         WIFI DISCONNECT**
         1. **关键词："FI DI"**
      2. **连接MQTT时改为（开头前无换行，其他都有）  
         +MQTTDISCONNECTED:0**
         1. **关键词："+"&&"TTDI"**
   6. **MQTT断联（开头前无换行，其他都有）  
      +MQTTDISCONNECTED:0**
      1. **关键词："+"&&"TTDI"**
   7. **AT+MQTTCONN?（开头前无换行，其他都有）  
      +MQTTCONN:0,4,1,"ad0ce5c71f.st1.iotda-device.cn-north-4.myhuaweicloud.com","1883","",1  
        
      OK**
      1. **关键词："+"&&MQTT\_HOSTNAME**
   8. **AT+CWJAP?（开头前无换行，其他都有）  
      【WARN】小心即使没连接到指定wifi也会识别出OK  
      +CWJAP:"321","ae:40:5a:6f:ce:b3",6,-35,0,1,3,0,0  
        
      OK**
      1. **关键词"+"&&"\"321\"" //WiFi\_USSID**
   9. **下行命令  
      +MQTTSUBRECV:0,"$oc/devices/AQAQ25032901/sys/commands/request\_id=1659d65f-497c-4f03-b68d-3f894c2a8136",62,{"paras":{"WPVR":1},"service\_id":"All","command\_name":"WPVRC"}**
      1. **关键词："+"&&"RECV'**
3. **ESP8266常见单向消息**
   1. **ESP8266上电消息**
      1. **（末尾没有换行）  
         ################################################  
         arch:ESP8266, 1  
         compile\_time:Ai-Thinker|B&T  
         wifi\_mac:dc4F226750F8  
         sdk\_version:v3.4-22-g967752e2  
         firmware\_version:2.2.0  
         compile\_time:Jul 31 2021 11:41:32  
           
         ready  
           
         ################################################**
   2. **ESP8266WiFi上电自动重连消息**
      1. **（开头没有换行）**
         1. **WIFI CONNECTED**
         2. **WIFI GOT IP**
   3. **ESP8266WiFi断开连接消息**
      1. **WIFI DISCONNECT**
4. **初始化**
   1. **AT+RESTORE**
      1. **成功响应（结尾无换行）**
         1. **################################################**
         2. **arch:ESP8266, 1  
            compile\_time:Ai-Thinker|B&T  
            wifi\_mac:dc4F226750F8**
         3. **sdk\_version:v3.4-22-g967752e2  
            firmware\_version:2.2.0**
         4. **compile\_time:Jul 31 2021 11:41:32**
         5. **ready  
              
            ################################################**
5. **常用指令**
   1. **[仅适用透传模式] +++：退出透传模式**
      1. **退出透传模式，进入透传接收模  
         +++**
   2. **取消消息重返  
      ATE0**
      1. **成功响应  
         ATE0  
           
         OK**
      2. **打开消息重返  
         ATE1**
         1. **成功响应  
              
            OK**
   3. **AT**
      1. **成功响应（ATE0）  
           
         OK**
6. **快捷连WiFi（WiFi连接一次后，未来重新上电自动连接）**
   1. **AT+CWMODE=1**
      1. **成功响应（ATE0）  
           
         OK**
   2. **断开与 AP 连接  
      AT+CWQAP**
      1. **成功响应，本就无连接（ATE0）  
           
         OK**
      2. **成功响应，原本有连接（ATE0）  
           
         WIFI DISCONNECT**
   3. **只有普通传输模式 (AT+CIPMODE=0)，才能设置为多连接  
      AT+CIPMUX=0**
      1. **成功响应（ATE0）  
           
         OK**
   4. **AT+CWJAP="WIFI\_SSID","WIFI\_PASSWORD"**
      1. **成功响应（ATE0）**
         1. **WIFI CONNECTED**
         2. **WIFI GOT IP**
      2. **自动重连响应  
         WIFI CONNECTED  
         WIFI GOT IP**
7. **常用查询**
   1. **AT+CWJAP?**
      1. **成功响应（ATE0）  
         +CWJAP:"321","ae:40:5a:6f:ce:b3",1,-35,0,1,3,0,0  
           
         OK**
   2. **AT+CMD?**
      1. **成功响应（ATE0）  
         太多，略...**
   3. **AT+MQTTCONN?**
      1. **空MQTTCONN响应（ATE0）  
         +MQTTCONN:0,0,0,"","","",0  
           
         OK**
      2. **成功响应（ATE0）  
         +MQTTCONN:0,4,1,"ad0ce5c71f.st1.iotda-device.cn-north-4.myhuaweicloud.com","1883","",1  
           
         OK**
8. **总结：3个AT指令即可接入：**
   1. **AT+MQTTUSERCFG=0,1,"NULL","AQAQ25032901","5eecee18613e8930c19fb93469c89b153fd48a3bb2fb606e1ce6b27fd35af531",0,0,""**
      1. **成功响应（ATE0）  
           
         OK**
   2. **AT+MQTTCLIENTID=0,"AQAQ25032901\_0\_0\_2025032907"**
      1. **成功响应（ATE0）  
           
         OK**
   3. **AT+MQTTCONN=0,"ad0ce5c71f.st1.iotda-device.cn-north-4.myhuaweicloud.com",1883,1**
      1. **成功响应（ATE0）  
         +MQTTCONNECTED:0,1,"ad0ce5c71f.st1.iotda-device.cn-north-4.myhuaweicloud.com","1883","",1  
           
         OK**
9. **上报**
   1. **MQTTX模拟**
      1. **{"services":[{"service\_id":"All","properties":{"WSD":2,"WQSV":2,"SMSV":2,"WT":2,"WPVR":2,"APRS":2,"WHRS":2,"ISV":5,"ALVR":2,"PGLVR":2,"FRS":2,"AT":2,"AH":2}}]}**
   2. **AT+MQTTPUB=0,"$oc/devices/AQAQ25032901/sys/properties/report","{\"services\":[{\"service\_id\":\"All\"\,\"properties\":{\"WSD\":2\,\"WQSV\":2\,\"SMSV\":2\,\"WT\":2\,\"WPVR\":2\,\"APRS\":2\,\"WHRS\":2\,\"ISV\":5\,\"ALVR\":2\,\"PGLVR\":2\,\"FRS\":2\,\"AT\":2\,\"AH\":2}}]}",0,1**
   3. **分段发送（电脑串口助手->ESP8266）**
      1. **AT+MQTTPUB=0,"$oc/devices/AQAQ25032901/sys/properties/report","{\"services\":[{\"service\_id\":\"All\"\,\"properties\":{\"WSD\":3\,\"WQSV\":3\,\"SMSV\":3\,\"WT\":3\,\"WPVR\":3\,\"APRS\":3\,\"WHRS\":3}}]}",0,1**
      2. **AT+MQTTPUB=0,"$oc/devices/AQAQ25032901/sys/properties/report","{\"services\":[{\"service\_id\":\"All\"\,\"properties\":{\"ISV\":3\,\"ALVR\":3\,\"PGLVR\":3\,\"FRS\":1\,\"AT\":1\,\"AH\":1}}]}",0,1**
   4. **分段发送（电脑串口助手->STM32->ESP8266）**
      1. **AT+MQTTPUB=0,"$oc/devices/AQAQ25032901/sys/properties/report","{\"services\":[{\"service\_id\":\"All\"\,\"properties\":{\"WSD\":9\,\"WQSV\":9\,\"SMSV\":9\,\"WT\":9\,\"WPVR\":9\,\"APRS\":9\,\"WHRS\":9}}]}",0,1**
      2. **AT+MQTTPUB=0,"$oc/devices/AQAQ25032901/sys/properties/report","{\"services\":[{\"service\_id\":\"All\"\,\"properties\":{\"ISV\":9\,\"ALVR\":9\,\"PGLVR\":9\,\"FRS\":9\,\"AT\":9\,\"AH\":9}}]}",0,1**
   5. **分段发送（STM32->ESP8266）**
      1. **AT+MQTTPUB=0,\"$oc/devices/AQAQ25032901/sys/properties/report\",\"{\\\"services\\\":[{\\\"service\_id\\\":\\\"All\\\"\\,\\\"properties\\\":{\\\"WSD\\\":%d\\,\\\"WQSV\\\":%.2f\\,\\\"SMSV\\\":%.2f\\,\\\"WT\\\":%d\\,\\\"WPVR\\\":%d\\,\\\"APRS\\\":%d\\,\\\"WHRS\\\":%d}}]}\",0,1\r\n**
      2. **AT+MQTTPUB=0,\"$oc/devices/AQAQ25032901/sys/properties/report\",\"{\\\"services\\\":[{\\\"service\_id\\\":\\\"All\\\"\\,\\\"properties\\\":{\\\"ISV\\\":10\\,\\\"ALVR\\\":9\\,\\\"PGLVR\\\":9\\,\\\"FRS\\\":9\\,\\\"AT\\\":9\\,\\\"AH\\\":9}}]}\",0,1\r\n**
   6. **成功响应（ATE0）  
        
      OK**
10. **订阅**
    1. **AT+MQTTSUB=0,"$oc/devices/AQAQ25032901/sys/commands/#",0**
       1. **成功响应（ATE0）  
            
          OK**
11. **接收的命令演示**
    1. **+MQTTSUBRECV:0,"$oc/devices/AQAQ25032901/sys/commands/request\_id=4723272a-c555-40d9-a863-11c4f49250d0",81,{"paras":{"WaterPVRatio":50},"service\_id":"Water","command\_name":"WaterPVConfig"}**
    2. **+MQTTSUBRECV:0,"$oc/devices/AQAQ25032901/sys/commands/request\_id=ef68f2a7-3b38-4ac4-a001-b35aa2157429",80,{"paras":{"WaterPVRatio":3},"service\_id":"Water","command\_name":"WaterPVConfig"}**
    3. **+MQTTSUBRECV:0,"$oc/devices/AQAQ25032901/sys/commands/request\_id=ef68f2a7-3b38-4ac4-a001-b35aa2157429",80,{"paras":{"WPVR":3},"service\_id":"Water","command\_name":"WPVRC"}**
12. **设备返回上行响应**
    1. **报文：  
       $oc/devices/AQAQ25032901/sys/commands/response/request\_id={request\_id}**
    2. **PC  
       AT+MQTTPUB=0,"$oc/devices/AQAQ25032901/sys/commands/response/request\_id=af4d23a2-72c0-46aa-8b8b-29f467351c30","{}",0,1**
    3. **STM32  
       AT+MQTTPUB=0,\"$oc/devices/AQAQ25032901/sys/commands/response/request\_id=af4d23a2-72c0-46aa-8b8b-29f467351c30\",\"{}\",0,1**
       1. **成功响应（ATE0）  
            
          OK**
    4. **MQTTX**
       * 1. **Topic: $oc/devices/67e22c942902516e866abb29\_AQAQ26032501/sys/commands/response/request\_id=25de16df-6630-4d51-9f7f-c9afeaecd189**
         2. **QoS: 0**
         3. **{}**

我要干的

1. 总体配置
   1. 设置 MQTT 用户属性  
      AT+MQTTUSERCFG=<LinkID>,<scheme>,<"client\_id">,<**"**username**"**>,<"password">,<cert\_key\_ID>,<CA\_ID>,<"path">
      1. AT+MQTTUSERCFG=0,1,"esp01s\_id","esp01s","esp01s",0,0,"/mqtt"
      2. **我写  
         AT+MQTTUSERCFG=0,1,"67e22c942902516e866abb29\_AQAQ26032501\_0\_0\_2025032607","67e22c942902516e866abb29\_AQAQ26032501","781117a499c134e294e0012691541cc379b8a994630a3651b62b8c1b6d3d85ae",0,0,""  
         AT+MQTTUSERCFG=0,1,"NULL","67e22c942902516e866abb29\_AQAQ26032501","781117a499c134e294e0012691541cc379b8a994630a3651b62b8c1b6d3d85ae",0,0,""  
         AT+MQTTUSERCFG=0,1,"NULL","67e22c942902516e866abb29\_AQAQ26032501","781117a499c134e294e0012691541cc379b8a994630a3651b62b8c1b6d3d85ae",0,0,""**
      3. <LinkID>
         1. 只能写0
         2. **我写  
            0**
      4. <scheme>
         1. 1：MQTT over TCP
         2. 6：MQTT over WebSocket（基于TCP）
         3. **我写  
            1**
      5. <client\_id>
         1. MQTT客户端ID，最大长度是256字节
         2. **我写  
            67e22c942902516e866abb29\_AQAQ26032501\_0\_0\_2025032509**
      6. <username>
         1. 用户名，用于登陆 MQTT broker，最大长度：64 字节
         2. **我写  
            67e22c942902516e866abb29\_AQAQ26032501**
      7. <password>
         1. 密码，用于登陆 MQTT broker，最大长度：64 字节。
         2. **我写  
            0ab0e141cdcb44ac5ba1e7fcf56cdae32d106e0e66e50c90b78a1fa30c0abd8b**
      8. <cert\_key\_ID>
         1. 证书ID,目前 ESP-AT 仅支持一套 cert 证书，参数为 0
         2. **我写  
            0**
      9. <CA\_ID>
         1. CA ID，目前 ESP-AT 仅支持一套 CA 证书，参数为 0。
         2. **我写  
            0**
      10. <path>
          1. 资源路径，最大长度：32 字节
          2. **我写  
             ？**
   2. 设置 MQTT 连接属性  
      AT+MQTTCONNCFG=<LinkID>,<keepalive>,<disable\_clean\_session>,<"lwt\_topic">,<"lwt\_msg">,<lwt\_qos>,<lwt\_retain>
      1. **我写  
         AT+MQTTCONNCFG=0,1,0,"$oc/devices/67e22c942902516e866abb29\_AQAQ26032501/sys/messages/up","AQAQ26032501",0,0**
      2. <LinkID>：当前仅支持 link ID 0。
         1. **我写  
            0**
      3. <keepalive>：MQTT ping 超时时间，单位：秒。范围：[0,7200]。默认值：0，会被强制改为 120 秒
         1. **我写  
            1**
      4. <disable\_clean\_session>：设置 MQTT 清理会话标志，有关该参数的更多信息请参考 MQTT 3.1.1 协议
         1. – 0: 使能清理会话
         2. – 1: 禁用清理会话
         3. **我写  
            0**
      5. <lwt\_topic>：遗嘱 topic，最大长度：128 字节。  
         格式例如：$oc/devices/{device\_ID}/...
         1. **我写  
            $oc/devices/{device\_id}/sys/messages/up**
      6. <lwt\_msg>：遗嘱 message，最大长度：64 字节。  
         遗嘱消息  
         一般含：设备ID，离线时间  
         大小<=64字节  
         格式未JSON或纯文本  
         禁止特殊字符
         1. **我写  
            AQAQ20250325163601001**
      7. <lwt\_qos>：遗嘱 QoS，参数可选 0、1、2，默认值：0  
         华为云支持：0、1
         1. 0：最多1次
         2. 1：最少1次
         3. **我写  
            0**
      8. <lwt\_retain>：遗嘱 retain，参数可选 0 或 1，默认值：0  
         即配置遗嘱是否保留/存储
         1. 1：在服务器存储，新订阅者可见。适用于持久化状态
         2. **我写  
            0**
      9. 遗嘱  
         设备意外离线时服务器自动发布遗嘱  
         通过此配置，设备意外离线时，华为云平台将自动收到遗嘱消息，实现设备状态的可靠监控。建议在代码中加入 AT+MQTTCONNCFG 配置，并在首次连接后用 AT+MQTTCONNCFG? 验证参数正确性。
   3. 连接 MQTT Broke  
      查询命令：AT+MQTTCONN?  
      设置命令：AT+MQTTCONN=<LinkID>,<"host">,<port>,<reconnect>
      1. **我写  
         AT+MQTTCONN=0,"ad0ce5c71f.st1.iotda-device.cn-north-4.myhuaweicloud.com",8883,0  
         AT+MQTTCONN=0,"ad0ce5c71f.st1.iotda-device.cn-north-4.myhuaweicloud.com",1883,1**
      2. <LinkID>：当前仅支持 link ID 0。
         1. **我写  
            0**
      3. <host>：MQTT broker 域名，最大长度：128 字节。
         1. **我写  
            ad0ce5c71f.st1.iotda-device.cn-north-4.myhuaweicloud.com**
      4. <port>：MQTT broker 端口，最大端口：65535。
         1. **我写  
            8883**
      5. <reconnect>：
         1. – 0: MQTT 不自动重连；
         2. – 1: MQTT 自动重连，会消耗较多的内存资源。
         3. **我写  
            1**
   4. 订阅 MQTT Topic  
      查询命令：AT+MQTTSUB?  
      设置命令：AT+MQTTSUB=<LinkID>,<"topic">,<qos>
      1. **我写  
         AT+MQTTSUB=0,"$oc/devices/67e22c942902516e866abb29\_AQAQ26032501/sys/commands/#",0**
      2. 查询命令响应  
         +MQTTSUB:<LinkID>,<state>,<"topic1">,<qos>  
         +MQTTSUB:<LinkID>,<state>,<"topic2">,<qos>  
         +MQTTSUB:<LinkID>,<state>,<"topic3">,<qos>  
         ...  
         OK
      3. 设置命令响应  
         OK
      4. 当AT 接收到已订阅的 topic 的 MQTT 消息时，返回：  
         +MQTTSUBRECV:<LinkID>,<"topic">,<data\_length>,data
      5. 若已订阅过该 topic，则返回：  
         ALREADY SUBSCRIBE
      6. <state>：MQTT状态
         1. 0: MQTT 未初始化；
         2. 1: 已设置 AT+MQTTUSERCFG；
         3. 2: 已设置 AT+MQTTCONNCFG；
         4. 3: 连接已断开；
         5. 4: 已建立连接；
         6. 5: 已连接，但未订阅 topic；
         7. 6: 已连接，已订阅过 MQTT topic。
      7. <"topic">
         1. 格式  
            $oc/devices/{device\_id}/sys/commands/request\_id={request\_id}
         2. 使用 + 匹配 request\_id 层（单层通配）  
            AT+MQTTSUB=0,"$oc/devices/+/sys/commands/+",1\r
         3. 避免过度通配：# 用于多层匹配（如 sys/commands/#），可能导致非预期消息。
   5. 取消订阅 MQTT Topic  
      AT+MQTTUNSUB=<LinkID>,<"topic">
      1. **我写  
         AT+MQTTSUB=0,"$oc/devices/67e22c942902516e866abb29\_AQAQ26032501/sys/commands/+"**
      2. 成功的响应  
         OK
      3. 未订阅过该topic 的响应  
         NO UNSUBSCRIBE  
         OK
   6. 断开MQTT连接  
      AT+MQTTCLEAN=<LinkID>
      1. **我写  
         AT+MQTTCLEAN=1**
      2. 成功的响应  
         OK
2. 子项独立配置
   1. 设置 MQTT 客户端 ID  
      AT+MQTTLONGCLIENTID=<LinkID>,<length>  
      发出命令后会返回’>’，然后才是发送Client\_ID
      1. **我写  
         AT+MQTTCLIENTID=0,"ioieobDLZef.esp01-s|securemode=2\,signmethod=hmacsha256\,timestamp=1704945210243|"  
         AT+MQTTCLIENTID=0,"67e22c942902516e866abb29\_AQAQ26032501\_0\_0\_2025032608"**
   2. 设置 MQTT 登陆用户名  
      AT+MQTTLONGUSERNAME=<LinkID>,<length>  
      发出命令后会返回’>’，然后才是发送User\_Name
      1. **我写  
         AT+MQTTLONGUSERNAME=0,"67e22c942902516e866abb29\_AQAQ26032501"**
   3. 设置 MQTT 登陆密码  
      AT+MQTTLONGPASSWORD=<LinkID>,<length>  
      发出命令后会返回’>’，然后才是发送Password
      1. **我写  
         AT+MQTTLONGPASSWORD=0,"9768d8186f4632bf52aae0edc9709382875d318083d2a17e4ee3408d7e4adc13"**
3. **上报MQTT**
   1. **发布 MQTT 消息（字符串）  
      AT+MQTTPUB=<LinkID>,<"topic">,<"data">,<qos>,<retain>**
      1. **我写  
         AT+MQTTPUB=0,"$oc/devices/67e22c942902516e866abb29\_AQAQ26032501/sys/properties/report","{"services":[{"service\_id":"Water","properties":{"WaterTemperature":18,"WaterSurfaceDistance":16,"WaterQualitySensorPortVoltage":3}}]}",0,1**
      2. <LinkID>我选0
      3. <topic>MQTT Topic  
         最大长度：128字节
      4. <data>MQTT字符串消息
      5. <qos>发布消息的QoS  
         可选：0、1、2  
         默认：0
      6. <retain>发布retain
      7. 本命令不能发送数据 \0，若需要发送该数据，请使用AT+MQTTPUBRAW 命令。
   2. 发布 MQTT 消息（二进制）
      1. AT+MQTTPUBRAW=<LinkID>,<"topic">,<length>,<qos>,<retain>  
         发送后ESP8166会返回相应信号“OK”  
         符号 > 表示 AT 准备好接收串口数据，此时您可以输入数据，当数据长度达到参数 <length> 的值时，数据传输开始。
      2. <length>：MQTT 消息长度，不同 ESP 设备的最大长度不同：
         1. – 对于 ESP32 设备：最大长度受到可利用内存的限制；
         2. – 对于 ESP8266 设备，最大长度受到可利用内存和 MQTT\_BUFFER\_SIZE\_BYTE 宏的限制。该宏的默认值为 512，可在 build.py menuconfig 中设置它的值，以此更改对最大长度的限制。该宏的值 = 消息的最大长度 + MQTT 报头长度（取决于 topic 名称的长度）。

每条 AT 命令的总长度不能超过 256 字节。

AQAQ20250325163601001的MQTT连接参数

{

"username": "67e22c942902516e866abb29\_AQAQ26032501",

"password": "eaede6a5685802224ba73cf89523c23de3a5e909cfc7fed3ff6a2201b20a9cdc",

"clientId": "67e22c942902516e866abb29\_AQAQ26032501\_0\_0\_2025032511",

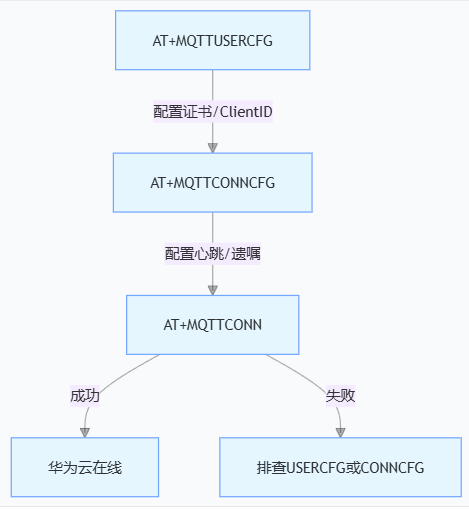
"hostname": "ad0ce5c71f.st1.iotda-device.cn-north-4.myhuaweicloud.com",

"port": 8883,

"protocol": "MQTTS"

}

命令顺序



遵规则：

USERCFG 必须在 CONNCFG 和 CONN 之前：证书和 ClientID 是连接的基础。

CONNCFG 可选：若使用默认值（如无遗嘱），可省略。

CONN 每次连接必执行：即使参数未变，也需手动触发

四、场景化差异：以华为云重连为例

场景 1：首次连接

执行 USERCFG（配置证书）

执行 CONNCFG（配置遗嘱）

执行 CONN（连接成功）

场景 2：断电后重连

仅需执行 CONN（自动复用之前的 USERCFG 和 CONNCFG 配置）

若遗嘱主题变更，需先执行 CONNCFG 再 CONN

场景 3：设备更换证书

执行 USERCFG（更新证书路径）

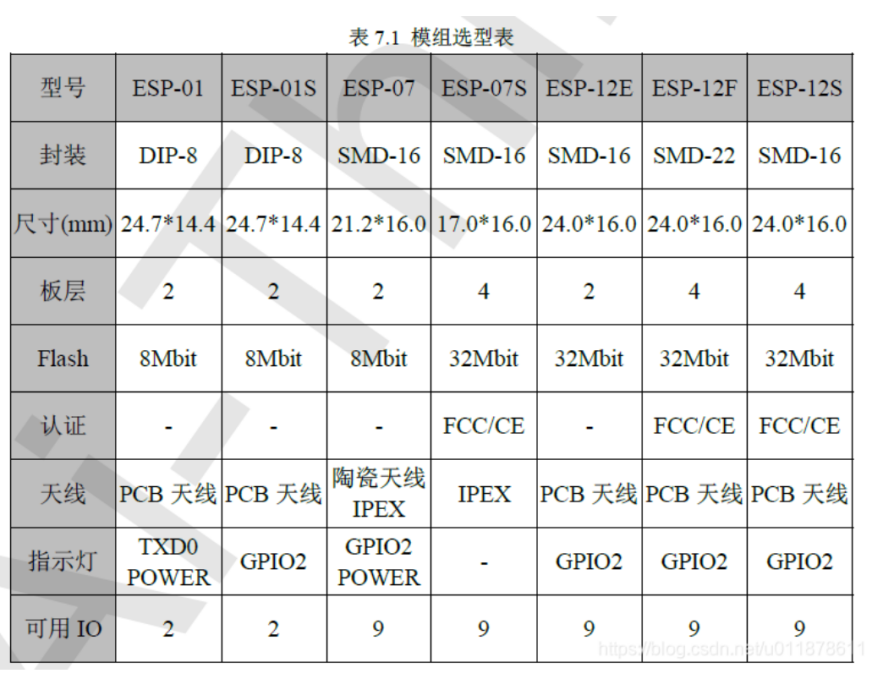
执行 CONN（强制使用新证书连接）

Device\_ID

67e22c942902516e866abb29\_AQAQ26032501

华为云TOPIC

| **Topic分类** | **Topic** | **Publisher(发布者)** | **Subscriber(订阅者)** | **用途** |
| --- | --- | --- | --- | --- |
| 设备消息相关Topic | $oc/devices/{device\_id}/sys/messages/up | 设备 | 平台 | [设备消息上报](https://support.huaweicloud.com/api-iothub/iot_06_v5_3016.html" \l "ZH-CN_TOPIC_0000002033145930) |
| $oc/devices/{device\_id}/sys/messages/down | 平台 | 设备 | [平台下发消息](https://support.huaweicloud.com/api-iothub/iot_06_v5_3017.html" \l "ZH-CN_TOPIC_0000002033304306) |
| 设备命令相关Topic | $oc/devices/{device\_id}/sys/commands/request\_id={request\_id} | 平台 | 设备 | [平台下发命令](https://support.huaweicloud.com/api-iothub/iot_06_v5_3014.html" \l "ZH-CN_TOPIC_0000002069345277) |
| $oc/devices/{device\_id}/sys/commands/response/request\_id={request\_id} | 设备 | 平台 | [设备返回命令响应](https://support.huaweicloud.com/api-iothub/iot_06_v5_3014.html" \l "ZH-CN_TOPIC_0000002069345277__zh-cn_topic_0000001426695400_zh-cn_topic_0208194431_section98871406444) |
| 设备属性相关Topic | $oc/devices/{device\_id}/sys/properties/report | 设备 | 平台 | [设备上报属性](https://support.huaweicloud.com/api-iothub/iot_06_v5_3010.html" \l "ZH-CN_TOPIC_0000002069223921) |
| $oc/devices/{device\_id}/sys/gateway/sub\_devices/properties/report | 设备 | 平台 | [网关批量上报属性](https://support.huaweicloud.com/api-iothub/iot_06_v5_3006.html" \l "ZH-CN_TOPIC_0000002033145934) |
| $oc/devices/{device\_id}/sys/properties/set/request\_id={request\_id} | 平台 | 设备 | [平台设置设备属性](https://support.huaweicloud.com/api-iothub/iot_06_v5_3008.html" \l "ZH-CN_TOPIC_0000002033304310) |
| $oc/devices/{device\_id}/sys/properties/set/response/request\_id={request\_id} | 设备 | 平台 | [属性设置的响应结果](https://support.huaweicloud.com/api-iothub/iot_06_v5_3008.html" \l "ZH-CN_TOPIC_0000002033304310__zh-cn_topic_0000001476455145_zh-cn_topic_0208194429_section98871406444) |
| $oc/devices/{device\_id}/sys/properties/get/request\_id={request\_id} | 平台 | 设备 | [平台查询设备属性](https://support.huaweicloud.com/api-iothub/iot_06_v5_3011.html" \l "ZH-CN_TOPIC_0000002069345285) |
| $oc/devices/{device\_id}/sys/properties/get/response/request\_id={request\_id} | 设备 | 平台 | [属性查询响应结果](https://support.huaweicloud.com/api-iothub/iot_06_v5_3011.html" \l "ZH-CN_TOPIC_0000002069345285__zh-cn_topic_0000001476614913_zh-cn_topic_0208194430_section98871406444) |
| $oc/devices/{device\_id}/sys/shadow/get/request\_id={request\_id} | 设备 | 平台 | [设备侧主动获取平台的设备影子数据](https://support.huaweicloud.com/api-iothub/iot_06_v5_3012.html" \l "ZH-CN_TOPIC_0000002069223925) |
| $oc/devices/{device\_id}/sys/shadow/get/response/request\_id={request\_id} | 平台 | 设备 | [设备侧主动获取平台设备影子数据的响应](https://support.huaweicloud.com/api-iothub/iot_06_v5_3012.html" \l "ZH-CN_TOPIC_0000002069223925__zh-cn_topic_0000001426536156_zh-cn_topic_0208194428_section98871406444) |
| 设备事件相关Topic | $oc/devices/{device\_id}/sys/events/up | 设备 | 平台 | 设备事件上报与平台事件下发，可用于：[设备网关管理](https://support.huaweicloud.com/api-iothub/iot_06_v5_3018.html" \l "ZH-CN_TOPIC_0000002033145938)、[软固件升级](https://support.huaweicloud.com/api-iothub/iot_06_v5_3027.html" \l "ZH-CN_TOPIC_0000002033145954)、[文件上传/下载](https://support.huaweicloud.com/api-iothub/iot_06_v5_3032.html" \l "ZH-CN_TOPIC_0000002033304338)、[设备时间同步](https://support.huaweicloud.com/api-iothub/iot_06_v5_3039.html" \l "ZH-CN_TOPIC_0000002033145966)、[设备信息上报](https://support.huaweicloud.com/api-iothub/iot_06_v5_30403.html" \l "ZH-CN_TOPIC_0000002069223957)、[设备日志收集](https://support.huaweicloud.com/api-iothub/iot_06_v5_1301.html" \l "ZH-CN_TOPIC_0000002033304354)、[远程配置](https://support.huaweicloud.com/api-iothub/iot_0801.html" \l "ZH-CN_TOPIC_0000002033145982) |
| $oc/devices/{device\_id}/sys/events/down | 平台 | 设备 |
| **Topic分类** | **Topic** | **Publisher(发布者)** | **Subscriber(订阅者)** | **用途** |
| 设备消息相关Topic | $oc/devices/{device\_id}/sys/messages/up | 设备 | 平台 | [设备消息上报](https://support.huaweicloud.com/api-iothub/iot_06_v5_3016.html" \l "ZH-CN_TOPIC_0000002033145930) |
| $oc/devices/{device\_id}/sys/messages/down | 平台 | 设备 | [平台下发消息](https://support.huaweicloud.com/api-iothub/iot_06_v5_3017.html" \l "ZH-CN_TOPIC_0000002033304306) |
| 设备命令相关Topic | $oc/devices/{device\_id}/sys/commands/request\_id={request\_id} | 平台 | 设备 | [平台下发命令](https://support.huaweicloud.com/api-iothub/iot_06_v5_3014.html" \l "ZH-CN_TOPIC_0000002069345277) |
| $oc/devices/{device\_id}/sys/commands/response/request\_id={request\_id} | 设备 | 平台 | [设备返回命令响应](https://support.huaweicloud.com/api-iothub/iot_06_v5_3014.html" \l "ZH-CN_TOPIC_0000002069345277__zh-cn_topic_0000001426695400_zh-cn_topic_0208194431_section98871406444) |
| 设备属性相关Topic | $oc/devices/{device\_id}/sys/properties/report | 设备 | 平台 | [设备上报属性](https://support.huaweicloud.com/api-iothub/iot_06_v5_3010.html" \l "ZH-CN_TOPIC_0000002069223921) |
| $oc/devices/{device\_id}/sys/gateway/sub\_devices/properties/report | 设备 | 平台 | [网关批量上报属性](https://support.huaweicloud.com/api-iothub/iot_06_v5_3006.html" \l "ZH-CN_TOPIC_0000002033145934) |
| $oc/devices/{device\_id}/sys/properties/set/request\_id={request\_id} | 平台 | 设备 | [平台设置设备属性](https://support.huaweicloud.com/api-iothub/iot_06_v5_3008.html" \l "ZH-CN_TOPIC_0000002033304310) |
| $oc/devices/{device\_id}/sys/properties/set/response/request\_id={request\_id} | 设备 | 平台 | [属性设置的响应结果](https://support.huaweicloud.com/api-iothub/iot_06_v5_3008.html" \l "ZH-CN_TOPIC_0000002033304310__zh-cn_topic_0000001476455145_zh-cn_topic_0208194429_section98871406444) |
| $oc/devices/{device\_id}/sys/properties/get/request\_id={request\_id} | 平台 | 设备 | [平台查询设备属性](https://support.huaweicloud.com/api-iothub/iot_06_v5_3011.html" \l "ZH-CN_TOPIC_0000002069345285) |
| $oc/devices/{device\_id}/sys/properties/get/response/request\_id={request\_id} | 设备 | 平台 | [属性查询响应结果](https://support.huaweicloud.com/api-iothub/iot_06_v5_3011.html" \l "ZH-CN_TOPIC_0000002069345285__zh-cn_topic_0000001476614913_zh-cn_topic_0208194430_section98871406444) |
| $oc/devices/{device\_id}/sys/shadow/get/request\_id={request\_id} | 设备 | 平台 | [设备侧主动获取平台的设备影子数据](https://support.huaweicloud.com/api-iothub/iot_06_v5_3012.html" \l "ZH-CN_TOPIC_0000002069223925) |
| $oc/devices/{device\_id}/sys/shadow/get/response/request\_id={request\_id} | 平台 | 设备 | [设备侧主动获取平台设备影子数据的响应](https://support.huaweicloud.com/api-iothub/iot_06_v5_3012.html" \l "ZH-CN_TOPIC_0000002069223925__zh-cn_topic_0000001426536156_zh-cn_topic_0208194428_section98871406444) |
| 设备事件相关Topic | $oc/devices/{device\_id}/sys/events/up | 设备 | 平台 | 设备事件上报与平台事件下发，可用于：[设备网关管理](https://support.huaweicloud.com/api-iothub/iot_06_v5_3018.html" \l "ZH-CN_TOPIC_0000002033145938)、[软固件升级](https://support.huaweicloud.com/api-iothub/iot_06_v5_3027.html" \l "ZH-CN_TOPIC_0000002033145954)、[文件上传/下载](https://support.huaweicloud.com/api-iothub/iot_06_v5_3032.html" \l "ZH-CN_TOPIC_0000002033304338)、[设备时间同步](https://support.huaweicloud.com/api-iothub/iot_06_v5_3039.html" \l "ZH-CN_TOPIC_0000002033145966)、[设备信息上报](https://support.huaweicloud.com/api-iothub/iot_06_v5_30403.html" \l "ZH-CN_TOPIC_0000002069223957)、[设备日志收集](https://support.huaweicloud.com/api-iothub/iot_06_v5_1301.html" \l "ZH-CN_TOPIC_0000002033304354)、[远程配置](https://support.huaweicloud.com/api-iothub/iot_0801.html" \l "ZH-CN_TOPIC_0000002033145982) |
| $oc/devices/{device\_id}/sys/events/down | 平台 | 设备 |



　引脚－接法  
　GND－GND  
　VCC－３ｖ３  
　CH-PD－VCC  
　GPIO0－GND  
　RXD－TXD  
　TXD－RXD



AT+CMD?

+CMD:0,"AT",0,0,0,1

+CMD:1,"ATE0",0,0,0,1

+CMD:2,"ATE1",0,0,0,1

+CMD:3,"AT+RST",0,0,0,1

+CMD:4,"AT+GMR",0,0,0,1

+CMD:5,"AT+CMD",0,1,0,0

+CMD:6,"AT+GSLP",0,0,1,0

+CMD:7,"AT+SYSTIMESTAMP",0,1,1,0

+CMD:8,"AT+SLEEP",0,1,1,0

+CMD:9,"AT+RESTORE",0,0,0,1

+CMD:10,"AT+SYSRAM",0,1,0,0

+CMD:11,"AT+SYSFLASH",0,1,1,0

+CMD:12,"AT+RFPOWER",0,1,1,0

+CMD:13,"AT+SYSMSG",0,1,1,0

+CMD:14,"AT+SYSROLLBACK",0,0,0,1

+CMD:15,"AT+SYSLOG",0,1,1,0

+CMD:16,"AT+SYSSTORE",0,1,1,0

+CMD:17,"AT+SLEEPWKCFG",0,0,1,0

+CMD:18,"AT+SYSREG",0,0,1,0

+CMD:19,"AT+USERRAM",0,1,1,0

+CMD:20,"AT+CWMODE",0,1,1,0

+CMD:21,"AT+CWSTATE",0,1,0,0

+CMD:22,"AT+CWJAP",0,1,1,1

+CMD:23,"AT+CWRECONNCFG",0,1,1,0

+CMD:24,"AT+CWLAP",0,0,1,1

+CMD:25,"AT+CWLAPOPT",0,0,1,0

+CMD:26,"AT+CWQAP",0,0,0,1

+CMD:27,"AT+CWSAP",0,1,1,0

+CMD:28,"AT+CWLIF",0,0,0,1

+CMD:29,"AT+CWQIF",0,0,1,1

+CMD:30,"AT+CWDHCP",0,1,1,0

+CMD:31,"AT+CWDHCPS",0,1,1,0

+CMD:32,"AT+CWSTAPROTO",0,1,1,0

+CMD:33,"AT+CWAPPROTO",0,1,1,0

+CMD:34,"AT+CWAUTOCONN",0,1,1,0

+CMD:35,"AT+CWHOSTNAME",0,1,1,0

+CMD:36,"AT+CWCOUNTRY",0,1,1,0

+CMD:37,"AT+CIFSR",0,0,0,1

+CMD:38,"AT+CIPSTAMAC",0,1,1,0

+CMD:39,"AT+CIPAPMAC",0,1,1,0

+CMD:40,"AT+CIPSTA",0,1,1,0

+CMD:41,"AT+CIPAP",0,1,1,0

+CMD:42,"AT+CIPV6",0,1,1,0

+CMD:43,"AT+CIPDNS",0,1,1,0

+CMD:44,"AT+CIPDOMAIN",0,0,1,0

+CMD:45,"AT+CIPSTATUS",0,0,0,1

+CMD:46,"AT+CIPSTART",0,0,1,0

+CMD:47,"AT+CIPSTARTEX",0,0,1,0

+CMD:48,"AT+CIPTCPOPT",0,1,1,0

+CMD:49,"AT+CIPCLOSE",0,0,1,1

+CMD:50,"AT+CIPSEND",0,0,1,1

+CMD:51,"AT+CIPSENDEX",0,0,1,0

+CMD:52,"AT+CIPDINFO",0,1,1,0

+CMD:53,"AT+CIPMUX",0,1,1,0

+CMD:54,"AT+CIPRECVMODE",0,1,1,0

+CMD:55,"AT+CIPRECVDATA",0,0,1,0

+CMD:56,"AT+CIPRECVLEN",0,1,0,0

+CMD:57,"AT+CIPSERVER",0,1,1,0

+CMD:58,"AT+CIPSERVERMAXCONN",0,1,1,0

+CMD:59,"AT+CIPSSLCCONF",0,1,1,0

+CMD:60,"AT+CIPSSLCCN",0,1,1,0

+CMD:61,"AT+CIPSSLCSNI",0,1,1,0

+CMD:62,"AT+CIPSSLCALPN",0,1,1,0

+CMD:63,"AT+CIPSSLCPSK",0,1,1,0

+CMD:64,"AT+CIPMODE",0,1,1,0

+CMD:65,"AT+CIPSTO",0,1,1,0

+CMD:66,"AT+SAVETRANSLINK",0,0,1,0

+CMD:67,"AT+CIPSNTPCFG",0,1,1,0

+CMD:68,"AT+CIPSNTPTIME",0,1,0,0

+CMD:69,"AT+CIPRECONNINTV",0,1,1,0

+CMD:70,"AT+MQTTUSERCFG",0,0,1,0

+CMD:71,"AT+MQTTCLIENTID",0,0,1,0

+CMD:72,"AT+MQTTUSERNAME",0,0,1,0

+CMD:73,"AT+MQTTPASSWORD",0,0,1,0

+CMD:74,"AT+MQTTCONNCFG",0,0,1,0

+CMD:75,"AT+MQTTCONN",0,1,1,0

+CMD:76,"AT+MQTTPUB",0,0,1,0

+CMD:77,"AT+MQTTPUBRAW",0,0,1,0

+CMD:78,"AT+MQTTSUB",0,1,1,0

+CMD:79,"AT+MQTTUNSUB",0,0,1,0

+CMD:80,"AT+MQTTCLEAN",0,0,1,0

+CMD:81,"AT+MDNS",0,0,1,0

+CMD:82,"AT+WPS",0,0,1,0

+CMD:83,"AT+CWSTARTSMART",0,0,1,1

+CMD:84,"AT+CWSTOPSMART",0,0,0,1

+CMD:85,"AT+PING",0,0,1,0

+CMD:86,"AT+FACTPLCP",0,0,1,0

+CMD:87,"AT+LEDTEST",0,0,1,0

+CMD:88,"AT+MCUTEST",0,0,1,0

+CMD:89,"AT+UART",0,1,1,0

+CMD:90,"AT+UART\_CUR",0,1,1,0

+CMD:91,"AT+UART\_DEF",0,1,1,0

ESP12F:

RX：AT+CMD?

+CMD:0,"AT",0,0,0,1

+CMD:1,"ATE0",0,0,0,1

+CMD:2,"ATE1",0,0,0,1

+CMD:3,"AT+RST",0,0,0,1

+CMD:4,"AT+GMR",0,0,0,1

+CMD:5,"AT+CMD",0,1,0,0

+CMD:6,"AT+GSLP",0,0,1,0

+CMD:7,"AT+SYSTIMESTAMP",0,1,1,0

+CMD:8,"AT+SLEEP",0,1,1,0

+CMD:9,"AT+RESTORE",0,0,0,1

+CMD:10,"AT+SYSRAM",0,1,0,0

+CMD:11,"AT+SYSFLASH",0,1,1,0

+CMD:12,"AT+RFPOWER",0,1,1,0

+CMD:13,"AT+SYSMSG",0,1,1,0

+CMD:14,"AT+SYSROLLBACK",0,0,0,1

+CMD:15,"AT+SYSLOG",0,1,1,0

+CMD:16,"AT+SYSSTORE",0,1,1,0

+CMD:17,"AT+SLEEPWKCFG",0,0,1,0

+CMD:18,"AT+SYSREG",0,0,1,0

+CMD:19,"AT+USERRAM",0,1,1,0

+CMD:20,"AT+CWMODE",0,1,1,0

+CMD:21,"AT+CWSTATE",0,1,0,0

+CMD:22,"AT+CWJAP",0,1,1,1

+CMD:23,"AT+CWRECONNCFG",0,1,1,0

+CMD:24,"AT+CWLAP",0,0,1,1

+CMD:25,"AT+CWLAPOPT",0,0,1,0

+CMD:26,"AT+CWQAP",0,0,0,1

+CMD:27,"AT+CWSAP",0,1,1,0

+CMD:28,"AT+CWLIF",0,0,0,1

+CMD:29,"AT+CWQIF",0,0,1,1

+CMD:30,"AT+CWDHCP",0,1,1,0

+CMD:31,"AT+CWDHCPS",0,1,1,0

+CMD:32,"AT+CWSTAPROTO",0,1,1,0

+CMD:33,"AT+CWAPPROTO",0,1,1,0

+CMD:34,"AT+CWAUTOCONN",0,1,1,0

+CMD:35,"AT+CWHOSTNAME",0,1,1,0

+CMD:36,"AT+CWCOUNTRY",0,1,1,0

+CMD:37,"AT+CIFSR",0,0,0,1

+CMD:38,"AT+CIPSTAMAC",0,1,1,0

+CMD:39,"AT+CIPAPMAC",0,1,1,0

+CMD:40,"AT+CIPSTA",0,1,1,0

+CMD:41,"AT+CIPAP",0,1,1,0

+CMD:42,"AT+CIPV6",0,1,1,0

+CMD:43,"AT+CIPDNS",0,1,1,0

+CMD:44,"AT+CIPDOMAIN",0,0,1,0

+CMD:45,"AT+CIPSTATUS",0,0,0,1

+CMD:46,"AT+CIPSTART",0,0,1,0

+CMD:47,"AT+CIPSTARTEX",0,0,1,0

+CMD:48,"AT+CIPTCPOPT",0,1,1,0

+CMD:49,"AT+CIPCLOSE",0,0,1,1

+CMD:50,"AT+CIPSEND",0,0,1,1

+CMD:51,"AT+CIPSENDEX",0,0,1,0

+CMD:52,"AT+CIPDINFO",0,1,1,0

+CMD:53,"AT+CIPMUX",0,1,1,0

+CMD:54,"AT+CIPRECVMODE",0,1,1,0

+CMD:55,"AT+CIPRECVDATA",0,0,1,0

+CMD:56,"AT+CIPRECVLEN",0,1,0,0

+CMD:57,"AT+CIPSERVER",0,1,1,0

+CMD:58,"AT+CIPSERVERMAXCONN",0,1,1,0

+CMD:59,"AT+CIPSSLCCONF",0,1,1,0

+CMD:60,"AT+CIPSSLCCN",0,1,1,0

+CMD:61,"AT+CIPSSLCSNI",0,1,1,0

+CMD:62,"AT+CIPSSLCALPN",0,1,1,0

+CMD:63,"AT+CIPSSLCPSK",0,1,1,0

+CMD:64,"AT+CIPMODE",0,1,1,0

+CMD:65,"AT+CIPSTO",0,1,1,0

+CMD:66,"AT+SAVETRANSLINK",0,0,1,0

+CMD:67,"AT+CIPSNTPCFG",0,1,1,0

+CMD:68,"AT+CIPSNTPTIME",0,1,0,0

+CMD:69,"AT+CIPRECONNINTV",0,1,1,0

+CMD:70,"AT+MQTTUSERCFG",0,0,1,0

+CMD:71,"AT+MQTTCLIENTID",0,0,1,0

+CMD:72,"AT+MQTTUSERNAME",0,0,1,0

+CMD:73,"AT+MQTTPASSWORD",0,0,1,0

+CMD:74,"AT+MQTTCONNCFG",0,0,1,0

+CMD:75,"AT+MQTTCONN",0,1,1,0

+CMD:76,"AT+MQTTPUB",0,0,1,0

+CMD:77,"AT+MQTTPUBRAW",0,0,1,0

+CMD:78,"AT+MQTTSUB",0,1,1,0

+CMD:79,"AT+MQTTUNSUB",0,0,1,0

+CMD:80,"AT+MQTTCLEAN",0,0,1,0

+CMD:81,"AT+MDNS",0,0,1,0

+CMD:82,"AT+WPS",0,0,1,0

+CMD:83,"AT+CWSTARTSMART",0,0,1,1

+CMD:84,"AT+CWSTOPSMART",0,0,0,1

+CMD:85,"AT+PING",0,0,1,0

+CMD:86,"AT+CIUPDATE",0,1,1,1

+CMD:87,"AT+FACTPLCP",0,0,1,0

+CMD:88,"AT+LEDTEST",0,0,1,0

+CMD:89,"AT+MCUTEST",0,0,1,0

+CMD:90,"AT+UART",0,1,1,0

+CMD:91,"AT+UART\_CUR",0,1,1,0

+CMD:92,"AT+UART\_DEF",0,1,1,0

OK

Qos是什么

MQTT 的 QoS（服务质量，Quality of Service） 是消息传递的可靠性等级，定义了消息从发布者到订阅者的传输保障机制。MQTT 协议支持 3 种 QoS 级别，分别对应不同的传输策略和可靠性：

1. QoS 0（最多一次，At Most Once）

定义：消息发送后不确认是否送达，可能丢失或重复。

传输机制：发布者发送消息后，不等待确认，直接结束流程。

保证：不保证消息到达，也不保证唯一性。

适用场景：对实时性要求高但允许少量丢失的场景（如传感器数据上报）。

2. QoS 1（至少一次，At Least Once）

定义：消息至少送达一次，可能重复但不丢失。

传输机制：

发布者发送消息并等待订阅者的确认（PUBACK）；

若未收到确认，重复发送直到成功。

保证：消息必达，但可能因重传导致重复。

适用场景：需要可靠性但允许去重的场景（如设备控制指令）。

3. QoS 2（恰好一次，Exactly Once）

定义：消息仅送达一次，确保不丢失且不重复。

传输机制：

发布者发送消息（PUBLISH），订阅者回复 PUBREC；

发布者确认 PUBREL，订阅者回复 PUBCOMP，完成四次握手。

保证：通过复杂的确认机制，确保消息唯一且可靠。

适用场景：对可靠性要求极高的场景（如金融交易、医疗数据）。

retain

MQTT 的 retain（保留消息） 是一种消息标记机制，允许服务器存储最新消息并推送给新订阅者或重连的旧订阅者。它解决了 “离线期间错过的关键消息” 问题，常见于设备状态同步、传感器数据等场景。以下是核心细节：

一、本质：服务器存储的 “最后已知值”

触发条件：当发布消息时设置 retain=1，服务器会永久存储该消息（直到被覆盖或清除）。

作用对象：仅对订阅该主题的客户端生效（包括订阅时离线的客户端）。

核心价值：新订阅者上线时，立即获取最新有效状态，无需等待发布者主动推送。