

MQTT 协议 知识点讲解



淘宝店铺：

PC 端：

<http://n-xytrt8gqu585po94mwj5atokcyd4.taobao.com/index.htm>

手机端：

https://shop.m.taobao.com/shop/shop_index.htm?sellerId=755668508&shopId=104493595&inShopPageId=423890608&pathInfo=shop/index2



资料下载地址：

链接：https://pan.baidu.com/s/1kCjD8yktZECSGmHomx_veg?pwd=q8er

提取码：q8er

源码下载地址：

<https://gitee.com/vi-iot/esp32-board.git>

一、MQTT 协议基本格式介绍

MQTT（Message Queuing Telemetry Transport，消息队列遥测传输协议），是一种基于发布/订阅（publish/subscribe）模式的"轻量级"通讯协议，该协议构建于 TCP/IP 协议上，由 IBM 在 1999 年发布。MQTT 最大优点在于，可以以极少的代码和有限的带宽，为连接远程设备提供实时可靠的消息服务。作为一种低开销、低带宽占用的即时通讯协议，使其在物联网、小型设备、移动应用等方面有较广泛的应用。

MQTT 是基于 TCP 的应用层协议，与 http、ftp 协议是处于同一层。几乎所有使用 ESP32 接入 IOT 物联网的项目中，都用的是 MQTT 协议，因此我们有必要学习 MQTT 协议基本内容。

1. MQTT 基本格式

MQTT 协议基本框架内容如下图所示

固定包头 (fixed header)	可变包头 (variable header)	荷载 (payload)
所有报文都包含	部分报文包含	部分报文包含

1) 固定包头

固定包头由如下构成

第一字节 4-7 位用于表示报文类型，0-3 位用于每种报文的特殊标志位。

报文类型(4-7 位)：

类型名称	类型值	流动方向	报文说明
Reserved	0	禁止	保留
CONNECT	1	客户端到服务器	发起连接
CONNACK	2	服务端到客户端	连接确认
PUBLISH	3	两个方向都允许	发布消息
PUBACK	4	两个方向都允许	QoS1消息确认
PUBREC	5	两个方向都允许	QoS2消息回执（保证交付第一步）
PUBREL	6	两个方向都允许	QoS2消息释放（保证交付第二步）
PUBCOMP	7	两个方向都允许	QoS2消息完成（保证交付第三步）
SUBSCRIBE	8	客户端到服务端	订阅请求
SUBACK	9	服务端到客户端	订阅确认
UNSUBSCRIBE	10	客户端到服务端	取消订阅
UNSUBACK	11	服务端到客户端	取消订阅确认
PINGREQ	12	客户端到服务端	心跳请求
PINGRESP	13	服务端到客户端	心跳响应
DISCONNECT	14	客户端到服务端	断开连接
Reserved	15	禁止	保留

报文的特殊标志位（0-3 位）：

控制报文	固定包头标志	Bit 3	Bit 2	Bit 1	Bit 0
CONNECT	Reserved	0	0	0	0
CONNACK	Reserved	0	0	0	0
PUBLISH	Used in MQTT 3.1.1	DUP ¹	QoS ²	QoS ²	RETAIN ³
PUBACK	Reserved	0	0	0	0
PUBREC	Reserved	0	0	0	0
PUBREL	Reserved	0	0	1	0
PUBCOMP	Reserved	0	0	0	0
SUBSCRIBE	Reserved	0	0	1	0
SUBACK	Reserved	0	0	0	0
UNSUBSCRIBE	Reserved	0	0	1	0
UNSUBACK	Reserved	0	0	0	0
PINGREQ	Reserved	0	0	0	0
PINGRESP	Reserved	0	0	0	0
DISCONNECT	Reserved	0	0	0	0

- DUP¹ = 控制报文的重复分发标志
- QoS² = PUBLISH 报文的服务质量等级
- RETAIN³ = PUBLISH 报文的保留标志

https://blog.csdn.net/qq_35720307

这个表中被标识为预留的，也需要填相应的值，如果接收方收到不可用的标识，接收方必须关闭网络连接，由此可见，这个特殊标志表对 PUBLISH 发布报文是最有用的。

- DUP——发布消息的副本。用来在保证消息的可靠传输，如果设置为 1，则在下面的变长中增加 MessageId，并且需要回复确认，以保证消息传输完成，但不能用于检测消息重复发送。
- QoS——发布消息的服务质量，QoS0:接收方最多能接收到一次，QoS1: 接收方最少收到一次，QoS2: 接收方只能收到一次
- RETAIN—— PUBLISH 数据包中 Retain 表示为 1 的消息，Broker 收到消息后，将会为该主题保存该 Retained 消息。当有新的订阅者订阅该主题时，Broker 会将这个消息立即发送给新的订阅者

固定包头第二字节开始表示剩余长度，该长度是后面的可变报文头加消息负载的总长度，该字段最多允许四个字节。剩余长度字段单个字节最大值为 16 进制 0x7F，也就是说，单个字节可以描述的最大长度是 127 字节。MQTT 协议规定，单个字节第 7 位（最高位）若为 1，则表示后续还有字节存在，第 7 位（最高位）起“延续位”的作用，所以最后一字节最大值只能是 0x7F。

例子如下：

比如剩余长度三个字节如下

0b1110 0111 = 0xE7 去掉第 7 位值为 0x67(第 7 位是 1，说明后面一个字节也表示剩余长度)

0b1000 1101 = 0x8D 去掉第 7 位值为 0x0D(第 7 位是 1，说明后面一个字节也表示剩余长度)

0b0001 0001 = 0x11 去掉第 7 位值为 0x0D(第 7 位是 0，表示后面没有长度)

计算方法位：剩余长度 len = 0x67 + (0x0D<<7) + (0x11<<14)

2) 可变报头与载荷

不一定每个报文都有，内容根据报文类型不同而不同，以下结合不同的报文类型说明一下可变包头与载荷内容

二、CONNECT 与 CONNACK 报文

CONNECT 报文是用于客户端向服务器发起连接请求的报文，服务器会验证其中的信息并且返回 CONNACK 告知客户端结果，在向 Broker 发布、订阅消息之前，需要先向服务器发起 CONNECT 报文，得到肯定回应后才能进行后续操作。CONNECT 包含固定包头、可变包头、消息载荷。CONNACK 包含固定包头、可变包头。

CONNECT 报文可变包头由如下构成：

协议名(Protocol Name)、协议等级(Protocol Level)、连接标志(Connect Flags)、保持连接(Keep Alive)

协议名长度MSB	协议名长度LSB	协议名	协议等级	连接标志	保持连接时间
1字节	1字节	N字节	1字节	1字节	2字节
00	04或06	MQTT 或MQIsdp	4或3		

主要看看连接标志，连接标志用于指示 payload 域存在哪些内容

Bit	7	6	5	4	3	2	1	0
每一位的含义	User Name Flag	Password Flag	Will Retain	Will QoS		Will Flag	Clean Session	Reserved
取值	X	X	X	X	X	X	X	0
描述	0 无用户名 1 有用户名	0 没有密码 1 有密码					是否存储会话状态	必须为 0

Clean Session:标识客户端是 0 否 1 建立一个持久化的会话，当 Clean Session 的标识设为 0 时，代表客户端希望建立一个持久会话的连接，代理服务器将存储该客户端订阅的主题和未接受的消息，否则（设置为 1）代理服务器不会存储这些数据，同时在建立连接时清除这个客户端之前存在的持久化会话所保存的数据。

Will Retain、Will QoS、Will Flag 这三个与遗嘱相关。当 Broker 检测到 Client 非正常断开连接时，就会向 Client 遗嘱主题中发布相应的遗嘱消息 LWT

- Will Flag: 是否使用 LWT 遗嘱消息；
- Will QoS: 发布遗嘱消息时的 QoS 等级
- Will Retain: 遗嘱消息的 Retain 标识

如果连接标志包含所有内容，必须按这个顺序出现：客户端标识符、遗嘱主题、遗嘱消息、用户名、密码。假设我们不使用 Will 功能，那么 payload 内容就是如下

ClientID长度MSB	ClientID长度LSB	username长度MSB	username长度LSB	username	password长度MSB	password长度LSB	password
1字节	1字节	N字节	1字节	N字节	1字节	1字节	N字节

CONNECT ACK 报文可变包头由如下构成：

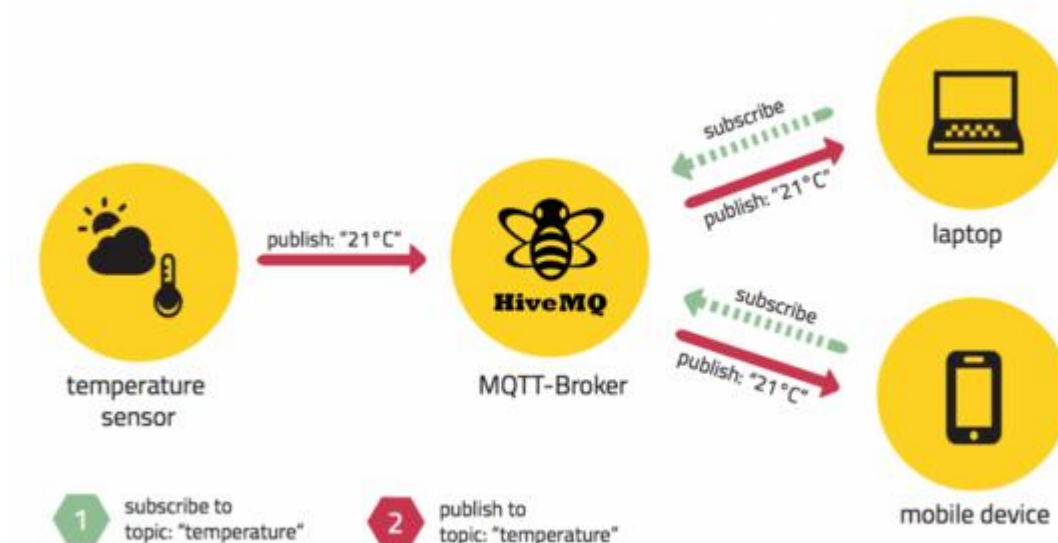
连接确认标志，返回码

连接确认标志	返回码
1字节	1字节
[7-1]位必须为0,[0]位表示当前会话标志	0: 表示CONNECT被接受, 其他均为拒绝

当 CONNECT 报文中的 Clean Session 标志设置为 1 时, 当前会话标志为 0

三、PUBLISH（发布）与 SUBSCRIBE（订阅）报文

这里先说一下订阅与推送过程



终端设备手机、电脑订阅了（Subscribe）主题“temperature”，这时候温度传感器向主题“temperature”发布了（Publish）了一条消息，内容是“21℃”，MQTT-Broker 收到这条消息后，会查看有哪些客户端订阅了这条消息主题，找到之后就会把这条消息内容转发给相应的客户端（手机、电脑）

Topic（主题）：

topic 在 MQTT 里面是消息传递的基础，代表了消息的流向，从本质上看，topic 是一串字符串，可以使用正斜杠对 topic 进行分级。

例子：

比如我向 A 代理服务器订阅了如下主题消息

“/编程知识/嵌入式/C 语言”，当有其他人向 A 服务器推送了主题“/编程知识/嵌入式/C 语言”的消息时候，那么代理服务器就会把这条消息推送给我，如果别人推送的是“/编程知识/嵌入式/C++”，由于主题不匹配，那么代理服务器就不会把这条消息推送给我们。

#和+通配符

topic 支持通配符进行订阅，比如说

订阅 “/编程知识/嵌入式/#” 的主题，那么订阅者就会收到诸如 “/编程知识/嵌入式/C++”、 “/编程知识/嵌入式/C”、” /编程知识/嵌入式/C/函数指针用法 “等等之类的消息。+与#的区别在于+只能匹配当前一层，#能匹配往下所有。

订阅报文 (SUBSCRIBE)

固定报头	byte1	[7-4]位报文类型	[3-0]位报文标志
		8(订阅报文)	2(固定QoS1)
	byte2-byteN	剩余表示数据长度	
可变报头	byte1	PacketID(MSB)	
	byte2	PacketID(LSB)	
帧载荷	byte1	主题长度(MSB)	
	byte2	主题长度(LSB)	
	byte3-byteN	主题内容	
	byteN+1	服务质量要求QoS (表示收这主题的服务质量最大等级)	

订阅回应报文(SUBACK)

固定报头	byte1	[7-4]位报文类型	[3-0]位报文标志
		9(订阅回应)	0
	byte2-byteN	剩余表示数据长度	
可变报头	byte1	PacketID(MSB)	
	byte2	PacketID(LSB)	
帧载荷	byte1	服务质量要求QoS	

取消订阅报文与 ACK 与上述两个报文类似，不同在于没有服务质量要求 Qos

发布报文(PUBLISH)

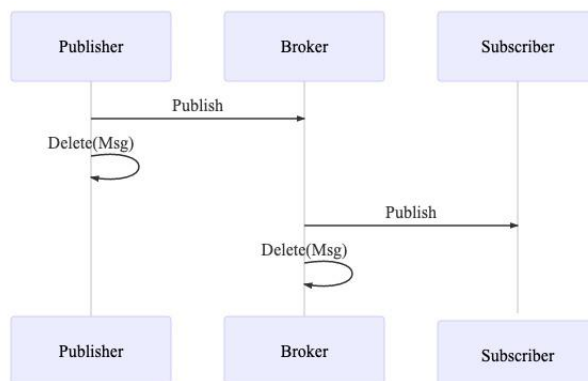
固定报头	byte1	[7-4]位报文类型	[3]位 DUP	[2-1]位 Qos	[0]位 Retain
		3(推送报文)	重发标志	Qos	表示消息持久保存
	byte2-byteN	剩余表示数据长度			
可变报头	byte1	主题长度(MSB)			
	byte2	主题长度(LSB)			
	byte3-byteN	主题内容			
	byteN+1	PacketID(MSB)			
	byteN+2	PacketID(LSB)			
帧载荷	byte1-byteN	自定义的帧载荷内容			

发布回应报文(PUBACK)

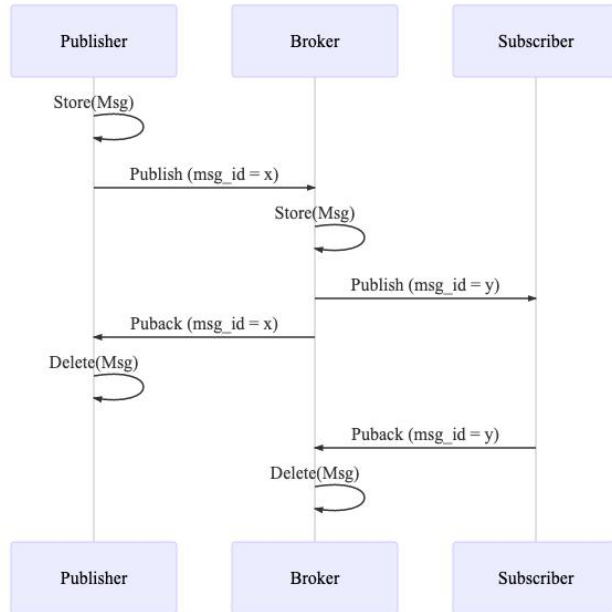
固定报头	byte1	[7-4]位报文类型	[3-0]位报文标志
		4(推送回应)	0
	byte2-byteN	剩余表示数据长度	
可变报头	byte1	PacketID(MSB)	
	byte2	PacketID(LSB)	

发布消息回执(PUBREC)、发布释放(PUBREL)、发布完成(PUBCOMP)报文与发布回应除报文类型外均一致。接下来看下三种不同的消息等级他们的消息流程以及发送的报文是怎样的。

QoS=0（接收方最多收到一次）



QoS=1（接收方最少收到一次）



QoS=2 (接收方只收到一次)

