

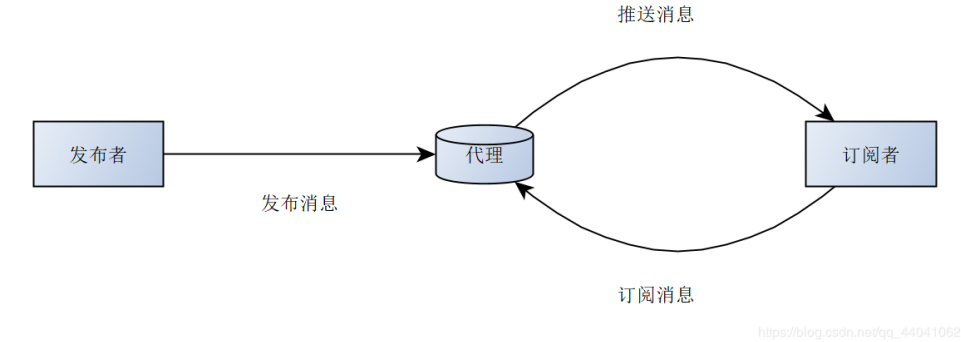
(1条消息)在linux下使用c语言实现MQTT通信(一.MQTT原理介绍及流程图) - qq_44041062的博客 - CSDN博客

版权声明：本文为博主原创文章，遵循CC 4.0 BY-SA 版权协议，转载请附上原文出处链接和本声明。

最近实验室的项目是要求在linux下使用C语言编程实现MQTT的通信。通信的内容是发布端（publish）发布从实验室树莓派获取的时间和温度，订阅端（suscribe）获取时间和温度。通过这几天的百度搜索总结了以下经验：

MQTT(消息队列遥测传输)是ISO 标准(ISO/IEC PRF 20922)下基于发布/订阅范式的消息协议。它工作在 TCP/IP协议族上，是为硬件性能低下的远程设备以及网络状况糟糕的情况下而设计的发布/订阅型消息协议，为此，它需要一个消息中间件。MQTT是一个基于客户端-服务器的消息发布/订阅传输协议。MQTT协议是轻量、简单、开放和易于实现的，这些特点使它适用范围非常广泛。在很多情况下，包括受限的环境中，如：机器与机器（M2M）通信和物联网（IoT）。其在，通过卫星链路通信传感器、偶尔拨号的医疗设备、智能家居、及一些小型化设备中已广泛使用。

它的基本模型如下：



MQTT控制报文的结构基本由三部分组成：固定报头（Fixed header），可变报头（Variable header），有效载荷（Payload）。控制报文的类型如下图所示：

名字	值	报文流动方向	描述
Reserved	0	禁止	保留
CONNECT	1	客户端到服务端	客户端请求连接服务端
CONNACK	2	服务端到客户端	连接报文确认
PUBLISH	3	两个方向都允许	发布消息
PUBACK	4	两个方向都允许	QoS 1消息发布收到确认
PUBREC	5	两个方向都允许	发布收到（保证交付第一步）
PUBREL	6	两个方向都允许	发布释放（保证交付第二步）
PUBCOMP	7	两个方向都允许	QoS 2消息发布完成（保证交互第三步）
SUBSCRIBE	8	客户端到服务端	客户端订阅请求
SUBACK	9	服务端到客户端	订阅请求报文确认
UNSUBSCRIBE	10	客户端到服务端	客户端取消订阅请求
UNSUBACK	11	服务端到客户端	取消订阅报文确认
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	客户端断开连接
Reserved	15	禁止	保留

https://blog.csdn.net/qq_44041062

网络连接 Network Connection

MQTT使用的底层传输协议基础设施。
客户端使用它连接服务端。 它提供有序的、可靠的、双向字节流传输。

客户端 Client

使用MQTT的程序或设备。客户端总是通过网络连接到服务端。它可以：

- 1.发布应用消息给其它相关的客户端。
- 2.订阅以请求接受相关的应用消息。
- 3.取消订阅以移除接受应用消息的请求。
- 4.从服务端断开连接。

服务端 Server

一个程序或设备，作为发送消息的客户端和请求订阅的客户端之间的中介。服务端：

- 1.接受来自客户端的网络连接。
- 2.接受客户端发布的应用消息。
- 3.处理客户端的订阅和取消订阅请求。
- 4.转发应用消息给符合条件的已订阅客户端。

订阅 Subscription

订阅包含一个主题过滤器（Topic Filter）和一个最大的服务质量（QoS）等级。订阅与单个会话（Session）关联。会话可以包含多于一个的订阅。会话的每个订阅都有一个不同的主题过滤器。

主题名 Topic Name

附加在应用消息上的一个标签，服务端已知且与订阅匹配。服务端发送应用消息的一个副本 给每一个匹配的客户端订阅。

主题过滤器 Topic Filter

订阅中包含的一个表达式，用于表示相关的一个或多个主题。主题过滤器可以使用通配符。 会话 Session 客户端和服务端之间的状态交互。一些会话持续时长与网络连接一样，另一些可以在客户端和服务端的多个连续网络连接间扩展。

控制报文 MQTT Control Packet

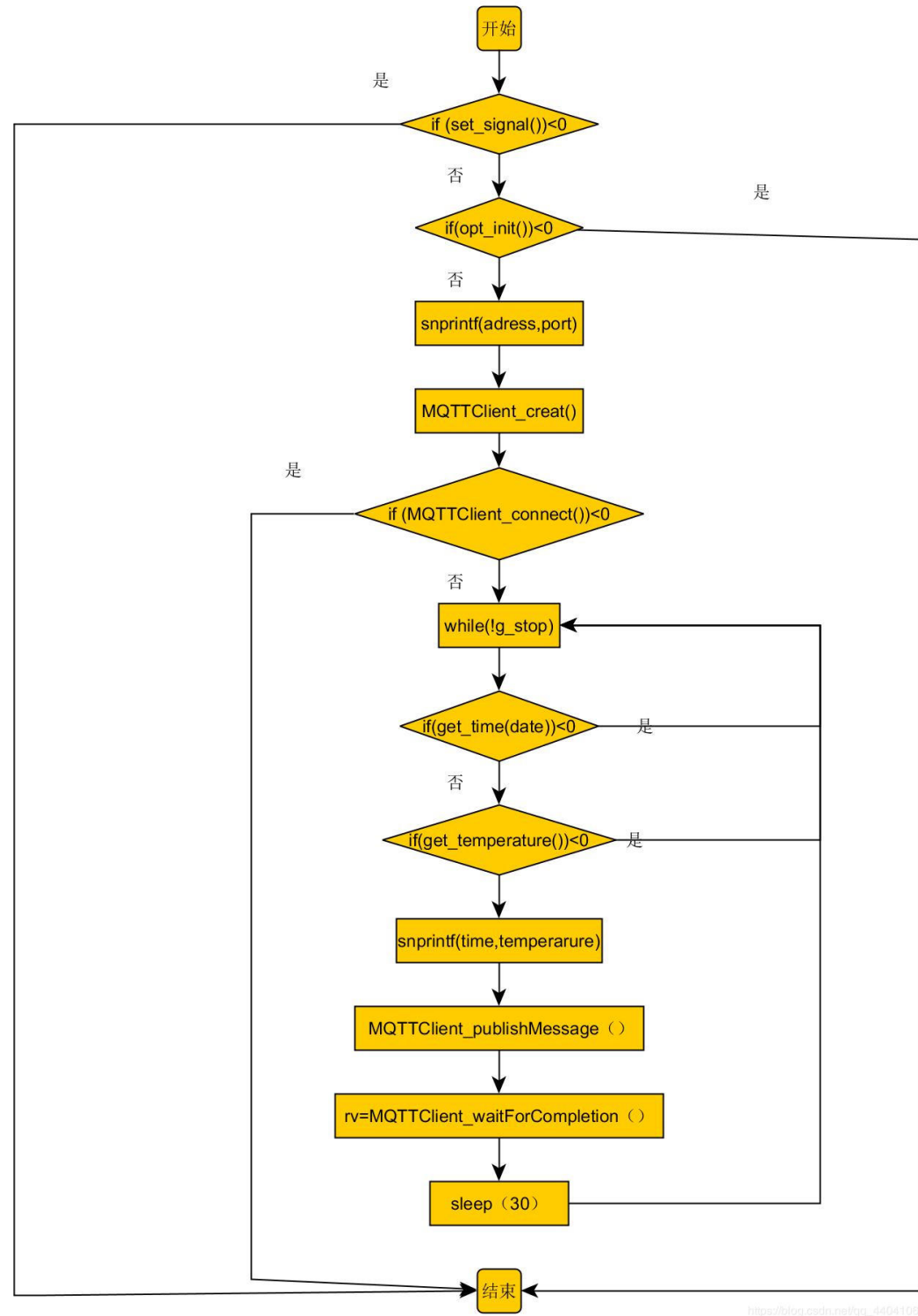
通过网络连接发送的信息数据包。MQTT规范定义了十四种不同类型的控制报文，其中一个（PUBLISH报文）用于传输应用消息。

MQTT具有三种消息发布服务质量：

三种消息发布服务质量:

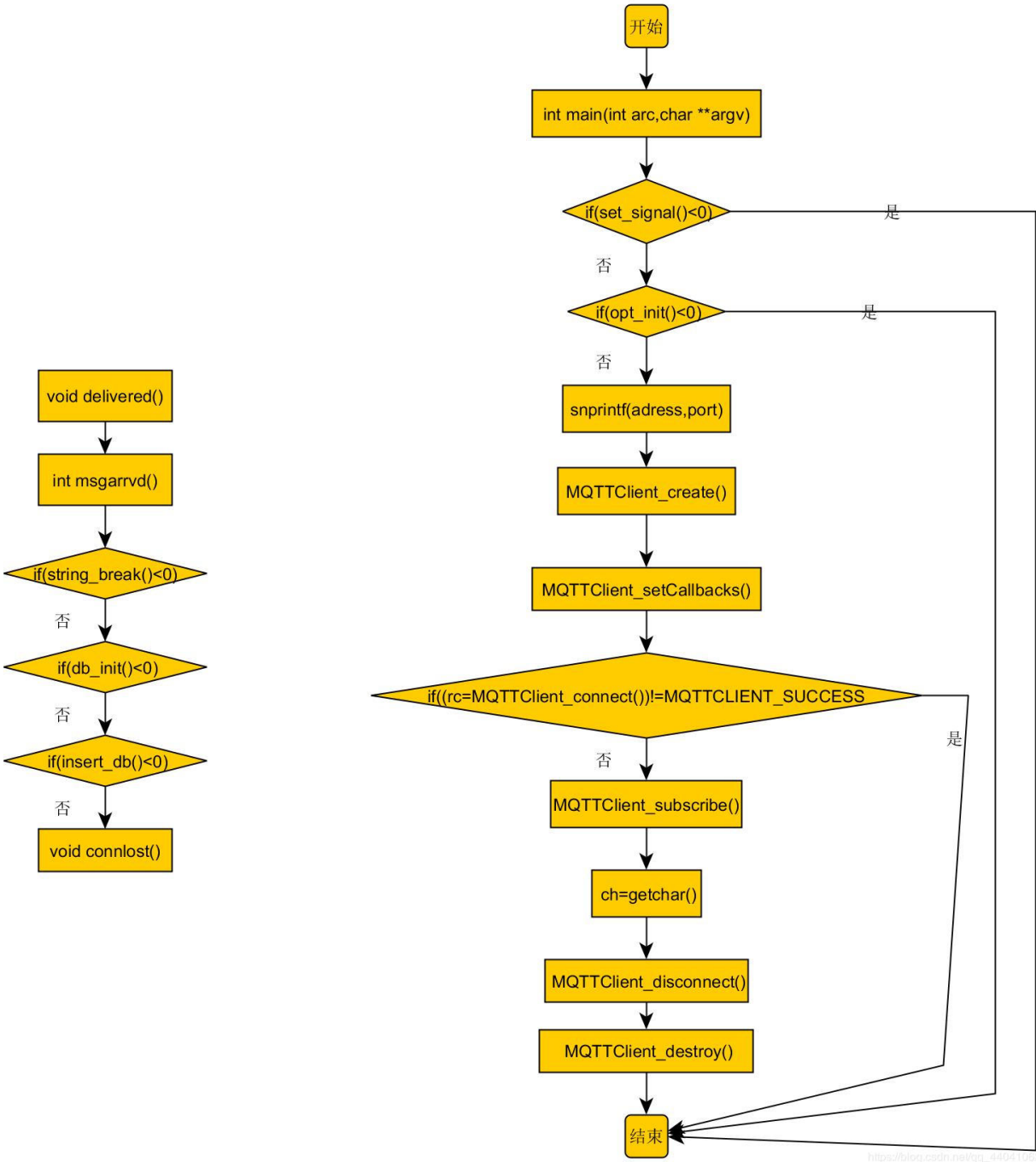
- 1.“至多一次”（QoS 0），消息发布完全依赖底层 TCP/IP 网络。会发生消息丢失或重复。这一级别可用于如下情况，环境传感器数据，丢失一次记录无所谓，因为不久后还会有第二次发送。
- 2.“至少一次”（QoS 1），确保消息到达，但消息可能会发生重复。
- 3.“只有一次”（QoS 2）确保消息到达一次。这一级别可用于如下情况，在计费系统中，消息重复或丢失会导致不正确的结果。

更多详细介绍请移步：
MQTT协议中文版: <https://github.com/mcxiaoke/mqtt>
publish端流程图：



subscribe端流程图：

https://blog.csdn.net/qq_44041062



文章最后发布于: 2019-07-22 16:46:49