

Please visit website: <http://cxyroad.com>

物联网通信，学学MQTT

=====

写在前面

=====

前段时间沉迷电子DIY，没事搞了个四足机器人玩，用的esp WiFi模块来做的通信，记录一下通信中学到的MQTT技术。

MQTT

=====

MQTT，全称为Message Queuing Telemetry Transport，是一种轻量级的消息传输协议。它与KafKa在架构上有相似之处，都采用了发布-订阅模式进行通信。不过，Kafka是一个消息队列系统，可以进行数据存储，而MQTT仅仅是一个类似于TCP的传输协议。

在实际应用中，MQTT涉及到三个角色：代理服务器（Broker），订阅者（Subscriber）和发布者（Publisher）。它们通过唯一的标识符——topic，来进行数据传输。在MQTT中，数据的收发双方无需像TCP那样建立连接，这一点在表现形式上与UDP广播有些相似。

订阅者通过订阅一个topic，在运行过程中，当有发布者对这个topic发送数据时，订阅者就能够接收到。MQTT的轻量级特性使其在物联网领域得到了广泛的应用。

MQTT相比HTTP特别轻量级，经常应用在物联网领域。

示例

==

这里我用python来演示，代理服务器Broker有很多公网免费的，直接网上搜索

用就行，topic是自己自定义的，用来作为自己“发布频道”的标识，这里我测试配置的config.py如下：

```
...  
# config.py  
# MQTT代理服务器的地址  
broker_address = "mqtt.eclipseprojects.io"  
  
# 自定义主题  
topic = "ssremex_test/topic"  
...
```

随后，来模拟发布者和订阅者，这里用到了一个第三方库`paho`，它是一个相对稳定的专门用来实现MQTT客户端的库。

```
...  
pip install paho  
...
```

在这个库中，不管是发布者还是订阅者都是被定义为客户端Client，只不过发布订阅调用的方法不同。下面分别来实现来进行实现

```
...  
# 发布者 publisher.py  
import paho.mqtt.client as client  
from config import broker_address, topic  
  
# 创建客户端实例，赋予ID P1  
publisher = client.Client(client.CallbackAPIVersion.VERSION2,  
client_id="P1")  
  
# 连接Broker代理  
publisher.connect(broker_address)  
  
message = "Hello MQTT"  
  
publisher.publish(topic, message)  
...
```

```

...

import paho.mqtt.client as client
from config import broker_address, topic
import time

# 连接成功回调函数
def on_connect(subscriber, userdata, flags, reason_code, properties):
    print(f"Connect with result code: {reason_code}")
    # 订阅topic
    subscriber.subscribe(topic)

# 消息接收回调函数
def on_message(subscriber, userdata, msg):
    print(f"Received message: {msg.payload.decode()}, topic: {msg.topic}")

subscriber = client.Client(client.CallbackAPIVersion.VERSION2,
client_id="S1")

subscriber.on_connect = on_connect
subscriber.on_message = on_message

subscriber.connect(broker_address)

subscriber.loop_forever()

...

```

代码很简单，通过构造回调函数的方式，对发布者和订阅者进行功能定制，关键地方也注释了，就不用过多解释。

演示
==

最后，让我们演示一下。由于MQTT本质上是实时通信技术，所以我们先运行订阅者，再运行发布者。

可以看到，数据成功发送，成功被订阅者接收！收工！

原文链接: <https://juejin.cn/post/7360512664315625506>