

| Mqtt通信实验

| 1. 实验目的

使用Mqtt网络通信协议实现通信。

| 2. 实验要求

- 软件要求：Win10/Win11系统；RflySim工具链；emqx^[1]。
- 硬件要求：笔记本/台式电脑1台^[2]。

| 3. 实验地址

例程目录：[\[安装目录\]\RflySimAPIs\9.RflySimComm\0.ApiExps\e4.MQTTDepsTestExps](#)

- [./MqttSender.py](#)：Mqtt发布者程序。
- [./MqttReceive.py](#)：Mqtt订阅器程序。
- [./Python38Run.bat](#)：Python环境一键启动脚本。

| 4. 实验内容或步骤

本实验分为两个小实验。

| 4.1 步骤1：emqx安装

从官网下载5.2.0版本emqx-5.2.0-windows-amd64.zip下载完成直接安装：

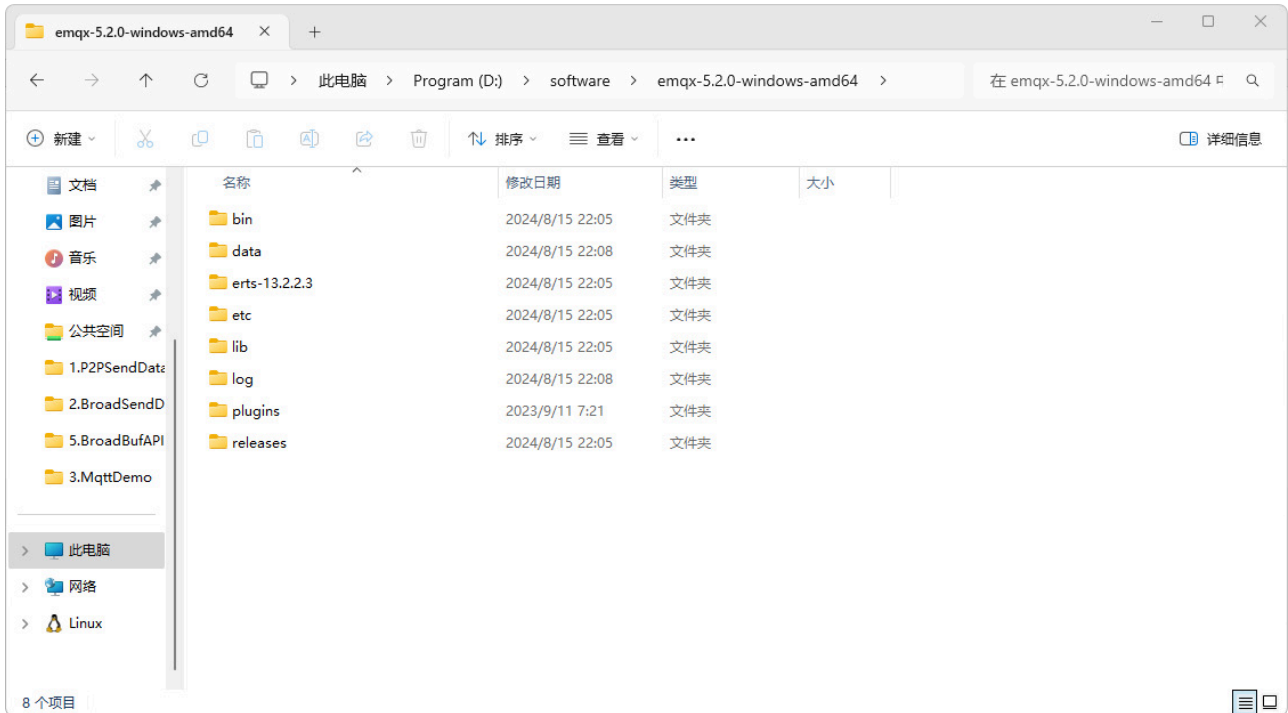
<https://www.emqx.com/en/downloads/broker>

或者从百度网盘进行下载：

链接：<https://pan.baidu.com/s/1FOYJfSmoJ2SU5L-mXpuD8A?pwd=7wso>

提取码：7wso

将下载好的emqx-5.2.0-windows-amd64.zip压缩包，进行解压缩。右键压缩包，点击"全部解压缩..."，会将所有文件解压缩到与压缩包同名的目录种。将该目录放置到想要安装的位置中，这里的安装路径如下图所示。

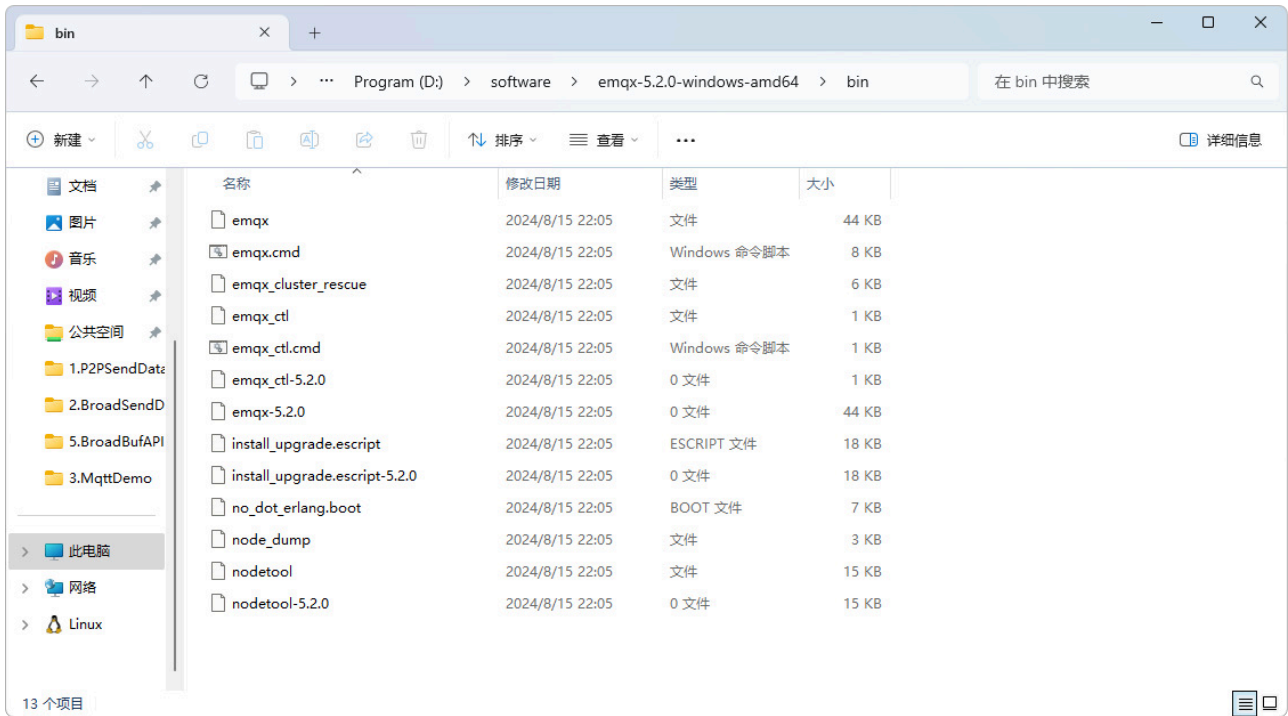


调试工具安装，MQTTX-Setup-1.9.6可以从官网（MQTTX 下载 <https://mqttx.app/zh/downloads>）获取。

该工具用于调试，如果不需要调试，可以跳过该步骤。

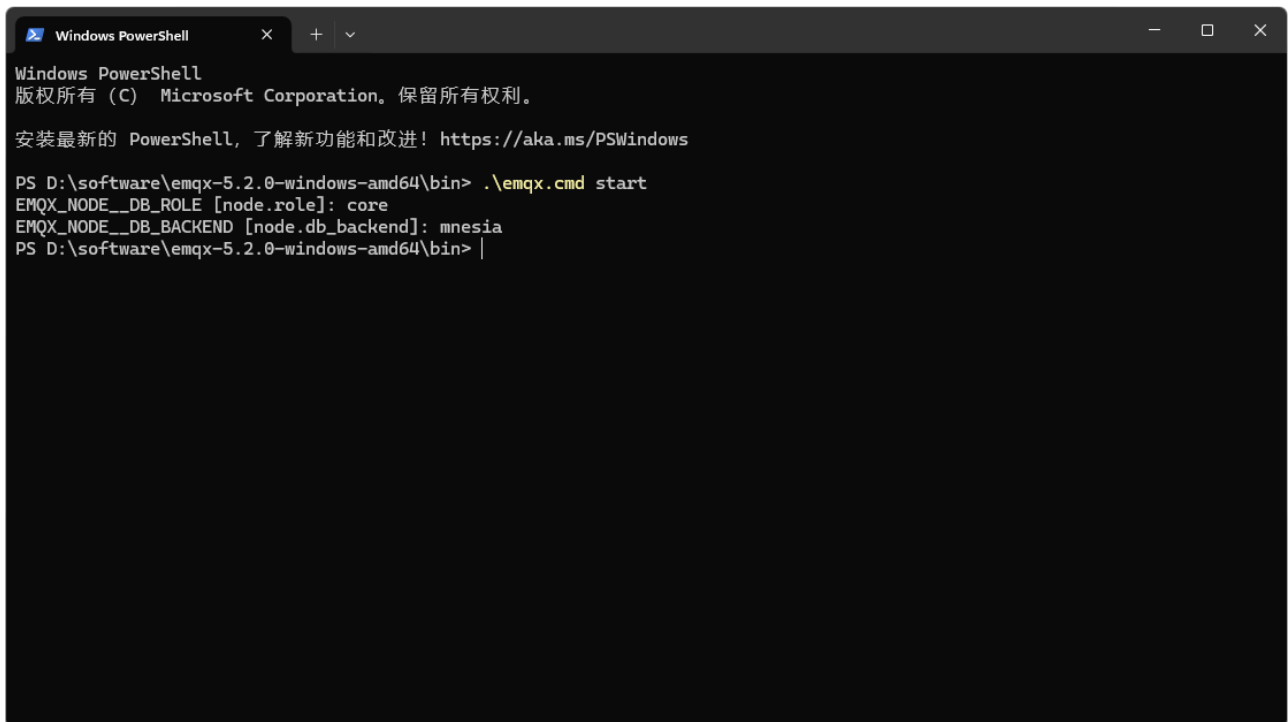
4.2 步骤2：本机实验

进入emqx的"bin"目录下。



在该目录中打开cmd，输入如下的命令，来开启mqtt服务器。

```
.\emqx.cmd start
```



注意，该服务是后台运行的服务，如果不手动进行关闭，则一直会在后台运行。如果需要关闭，可以在该目录中输入如下的命令，来关闭mqtt服务。

```
.\emqx.cmd stop
```

```
Windows PowerShell
版权所有 (C) Microsoft Corporation。保留所有权利。

安装最新的 PowerShell，了解新功能和改进！ https://aka.ms/PSWindows

PS D:\software\emqx-5.2.0-windows-amd64\bin> .\emqx.cmd start
EMQX_NODE__DB_ROLE [node.role]: core
EMQX_NODE__DB_BACKEND [node.db_backend]: mnesia
PS D:\software\emqx-5.2.0-windows-amd64\bin> .\emqx.cmd stop
PS D:\software\emqx-5.2.0-windows-amd64\bin> |
```

双击运行"Python38Run.bat"，在其中输入"python MqttReceive.py"，监听订阅话题的消息。

```
C:\windows\system32\cmd.exe
Python3.8 environment has been set with openCV+pymavlink+numpy+pyulog etc.
You can use pip or pip3 command to install other libraries
Put Python38Run.bat into your code folder
Use the command: 'python XXX.py' to run the script with Python

E:\summerwork\work1\9.RflySimComm\0.ApiExps\3.MqttDemo>python MqttReceive.py
Connected to MQTT with result code 0
|
```

双击运行"Python38Run.bat"，在其中输入"python MqttSender.py"，发布话题信息。

```
C:\windows\system32\cmd.exe x + v
Python3.8 environment has been set with openCV+pymavlink+numpy+pyulog etc.
You can use pip or pip3 command to install other libraries
Put Python38Run.bat into your code folder
Use the command: 'python XXX.py' to run the script with Python

E:\summerwork\work1\9.RflySimComm\0.ApiExps\3.MqttDemo>python MqttSender.py
E:\summerwork\work1\9.RflySimComm\0.ApiExps\3.MqttDemo>
```

观察运行"[MqttReceive.py](#)"监听程序的窗口，可以看到收到的信息被打印。

```
C:\windows\system32\cmd.exe x + v
Python3.8 environment has been set with openCV+pymavlink+numpy+pyulog etc.
You can use pip or pip3 command to install other libraries
Put Python38Run.bat into your code folder
Use the command: 'python XXX.py' to run the script with Python

E:\summerwork\work1\9.RflySimComm\0.ApiExps\3.MqttDemo>python MqttReceive.py
Connected to MQTT with result code 0
Received message on topic mytopic: this is my topic !!!
|
```

多次运行之后，还可以继续接受消息。

```
C:\windows\system32\cmd.exe x + v
Python3.8 environment has been set with openCV+pymavlink+numpy+pyulog etc.
You can use pip or pip3 command to install other libraries
Put Python38Run.bat into your code folder
Use the command: 'python XXX.py' to run the script with Python

E:\summerwork\work1\9.RflySimComm\0.ApiExps\3.MqttDemo>python MqttReceive.py
Connected to MQTT with result code 0
Received message on topic mytopic: this is my topic !!!
Received message on topic mytopic: this is my topic !!!
Received message on topic mytopic: this is my topic !!!
```

5. 关键知识点

关键知识点1：MQTT协议基础

MQTT (Message Queuing Telemetry Transport) 是一种轻量级的网络通信协议，它基于发布/订阅模式。在MQTT实验中，客户端（如传感器设备或应用程序）首先与MQTT代理服务器建立连接。然后，客户端可以向代理服务器发送订阅请求，指定要接收的消息的主题。当其他客户端或其他来源发布消息到某个主题时，代理服务器会将这些消息转发给已订阅该主题的客户。

关键知识点2：创建MQTT客户端实例

创建MQTT客户端实例。

```
# 使用 MQTTv31 或 MQTTv50 协议版本创建客户端
client = mqtt.Client()
```

关键知识点3：连接MQTT服务器

连接MQTT服务器。

```
# 连接服务器,端口为1883,维持心跳为60秒
client.connect(MqttServerIp, 1883, 60)
```

关键知识点4：发布消息

发布消息。

```
data = "this is my topic !!!"
client.publish(REPORT_TOPIC, data, 1)
```

关键知识点5：处理连接回调函数

如果需要连接之后调用回调函数，可以按照如下方式进行注册。

```
# 如果需要处理回调函数，可以使用以下方法注册
def on_connect(client, userdata, flags, rc):
    print("Connected with result code " + str(rc))
client.on_connect = on_connect
```

关键知识点6：处理消息回调函数

对于订阅者，需要实现对消息回调函数的注册。

```
def on_message(client, userdata, msg):
    message = msg.payload.decode()
    print(f"Received message on topic {msg.topic}: {message}")
client.on_message = on_message
```

6.参考资料

1. [MQTT协议官方文档](#)
2. [EMQ X Broker官方文档](#)
3. [MQTT Python客户端库paho-mqtt文档](#)

7. 常见问题

Q1: 无法连接到MQTT服务器怎么办?

A1: 请检查以下几点: 1) EMQ X Broker是否已经正确安装并成功启动; 2) 确认连接参数(服务器IP、端口号等)是否正确; 3) 防火墙设置是否阻止了连接。

Q2: 为什么发布消息后订阅端收不到消息?

A2: 可能的原因包括: 1) 发布端和订阅端的话题(Topic)不匹配; 2) MQTT服务器配置问题; 3) 网络连接不稳定导致消息传递失败。请确认Topic完全一致, 且服务器正常运行。

Q3: 如何测试MQTT通信是否正常?

A3: 可以使用MQTTX等调试工具来辅助测试。在运行Python程序前, 先用调试工具连接到MQTT服务器, 验证服务器是否正常工作, 然后再运行Python程序进行测试。

1. <https://rflysim.com/> ↩

2. 推荐配置请见: <https://rflysim.com/doc/zh/HowToInstall.pdf> ↩