

MQTT 协议集中器 应用指南

Ver 1.4

2020.07

上海睿据科技

目录

1. 概述.....	3
2. MQTT server/broker 配置.....	4
2.1. 配置文件示例.....	4
2.1.1. Mosquitto.....	4
2.1.2. ActiveMQ.....	6
2.2. TopicRoot.....	8
2.3. QoS.....	9
2.4. MQTT 客户端工具.....	9
3. 集中器配置.....	9
3.1. 编码.....	10
3.2. MQTT 服务器.....	10
3.3. MQTT 登录密码.....	10
3.4. TopicRoot.....	10
3.5. NTP.....	11
4. 集中器在线升级包.....	11

1. 概述

MQTT 协议智能集中器(即智能网关), 是为方便第三方系统使用睿据科技的智能断路器(即智能空气开关)产品来集成实现智慧用电等相关功能。

集中器作为 MQTT client, 连接第三方系统的 MQTT server/broker。集中器实时发布采集到的其下各线路(即断路器)的相关电力数据、状态信息等, 同时通过订阅获得第三方系统发布的线路开关等控制命令, 来对其下的相关线路执行相应的实际操作。

同时, 第三方系统需要开发一个管理模块, 也作为 MQTT client 连接在这同一 MQTT server/broker。一方面通过订阅来获得所有集中器发布的实时电力数据和状态信息等, 另一方面根据自身系统功能需要, 向对应集中器发布相应的远程控制操作等等命令。

相较于第三方系统直接使用断路器设备来进行功能集成, 使用 MQTT 协议集中器的集成方式有如下的优势:

- 开发效率高

集中器已经实现了对各类断路器的采集、控制和数据解析功能, 不需要再从断路器层面来进行开发。

- 采集策略

集中器定义了一套通用型的采集策略, 第三方系统可以直接采用。

- 智能控制

可以在集中器上直接驻留多个定时控制功能, 其效率更高。同时可以防止网络故障, 从而比服务端的定时控制功能更可靠。

- 免费在线升级

如果直接使用断路器来进行开发集成, 则实现的代码将局限于现有断路器的功能。如果系统今后使用断路器新一代产品功能, 则需要进行新的开发。

而睿据全线的集中器设备都是支持在线升级的。当断路器有新产品发布时, 所有类型集中器的功能模块将同步进行升级并免费发布。第三方系统可以使用该升级包对其下的所有集中器自动进行在线的热升级, 即可无缝接入新产品功能。

2. MQTT server/broker 配置

2.1. 配置文件示例

为保证数据通讯的安全性，集中器使用 MQTT over TLS (without client certificate)的单向认证方式来连接 MQTT server/broker。

同时，MQTT server/broker 中需要为每个将要连接的集中器创建登录用户及密码。约定使用集中器编码(12 位数字编码，在集中器端设置)直接作为 Login 用户名，而管理模块使用的用户名和密码则有第三方系统自行决定。

以 mosquitto(<https://mosquitto.org/>) 为例，来说明如何配置 MQTT server/broker。

2.1.1. Mosquitto

mosquitto.conf 相关内容：

```
# -----  
# Certificate based SSL/TLS support  
# -----  
...  
#cafile  
#capath  
cafile xxxxxxxxxxxxxxxx  
  
# Path to the PEM encoded server certificate.  
#certfile  
certfile xxxxxxxxxxxxxxxx  
  
# Path to the PEM encoded keyfile.  
#keyfile  
keyfile xxxxxxxxxxxxxxxx
```

同时, `require_certificate` 和 `use_identity_as_username` 保持默认 `false` 设置。

```
# =====  
# Security  
# =====  
  
...  
#allow_anonymous true  
allow_anonymous false  
  
# -----  
# Default authentication and topic access control  
# -----  
  
...  
#password_file  
password_file /etc/mosquitto/pwfile  
  
...  
#acl_file  
acl_file /etc/mosquitto/aclfile
```

`pwfile` 文件可以直接用 `pwfile.example` 拷贝生成, 然后使用:

```
mosquitto_passwd -b passwordfile username password
```

命令来创建写入各用户/密码。

如下为 `pwfile` 文件内容示例:

```
admin_client:$6$2q5FFRU9FGCifP6P$CxCeENzDJkmhMslkw6j9WXiJtFC3myvkiO  
Q+oyZvL3ojXuVWepANcjG4ZfzWEjrFHpmWTnJF+onWe0CBr7DHdA==  
202005010001:$6$IC9XnZ9oOUb1Sa1n$gN/qByXcVqKa8nZer5uOCct0W2S8llnP  
02sV73cs+cFD2P2lwy8lZkcByXvdXp9yHHfDO3p53cJ9EM9tqvks6A==
```

`admin_client` 为第三方系统的管理模块客户端所使用用户, `202005010001` 为该编码的集中器所对应的用户。

aclfile 文件可以直接用 aclfile.example 拷贝生成后直接编辑修改。

如下为 aclfile 文件内容示例：

```
# This only affects clients with username "admin_client".
user admin_client
topic read smart/devs+/fromdev
topic write smart/devs+/todev
# This affects all clients.
pattern write $SYS/broker/connection/%c/state
pattern write smart/devs/%u/fromdev
pattern read smart/devs/%u/todev
```

上面一段为针对管理模块的 MQTT 客户端用户，表示其对所有集中器的 fromdev 主题可读，对所有集中器的 todev 主题可写。

最后两行为针对所有集中器的 MQTT 客户端用户，表示集中器对其用户 id(如前描述，即为该集中器的编码)下的 fromdev 主题可写，对其用户 id 下的 todev 主题可读。

其中 smart/devs 为 TopicRoot，由第三方系统自行决定。请参见 TopicRoot 一节中的说明。

2.1.2. ActiveMQ

使用证书 pfx 生成 keystore 文件：

```
keytool -importkeystore -srckeystore "你的证书.pfx" -destkeystore "activeMQ
用的 keystore.jks" -srcstoretype PKCS12 -deststoretype JKS
```

注意，建议转换后的文件使用和源文件相同的密码。

在 activemq.xml 中的<broker>中设置 sslContext:

```
<sslContext>
  <sslContext keyStore="file: activeMQ 用的 keystore.jks 全路径"
keyStorePassword="密码" />
</sslContext>
```

因为使用单向认证，所以不需要设置客户端的文件。

在 `activemq.xml` 中的 `<broker>` 中设置用户和 `topic` 访问权限：

注意，以下为了确保每个断路器只访问自己的键值，对它们进行了分开设 `group`。可自行决定是否统一设置 `group`。

```
<plugins>
  <simpleAuthenticationPlugin>
    <users>
      <authenticationUser username="admin_client"
password="admin_client 的密码" groups="admin_client"/>
      <authenticationUser username="202001010001"
password="202001010001 的密码" groups="202001010001"/>
    </users>
  </simpleAuthenticationPlugin>

  <authorizationPlugin>
    <map>
      <authorizationMap>
        <authorizationEntries>
          <authorizationEntry
topic="smart.devs.202001010001.fromdev" read="admin_client"
write="202001010001" admin="202001010001,admin_client" />
          <authorizationEntry
topic="smart.devs.202001010001.todev" write="admin_client"
read="202001010001" admin="202001010001,admin_client" />
        </authorizationEntries>
      </authorizationMap>
    </map>
  </authorizationPlugin>
</plugins>
```

在 `activemq.xml` 中的 `<broker>` 中设置访问入口：

```
<transportConnectors>
    <!-- DOS protection, limit concurrent connections to 1000 and frame size to
100MB -->
    <!-- <transportConnector name="openwire"
uri="tcp://0.0.0.0:61616?maximumConnections=1000&wireFormat.maxFrameS
ize=104857600"/> -->
    <!-- <transportConnector name="amqp"
uri="amqp://0.0.0.0:5672?maximumConnections=1000&wireFormat.maxFrame
Size=104857600"/> -->
    <!-- <transportConnector name="stomp"
uri="stomp://0.0.0.0:61613?maximumConnections=1000&wireFormat.maxFra
meSize=104857600"/> -->
    <transportConnector name="mqtt"
uri="mqtt+ssl://0.0.0.0:9883?maximumConnections=1000&wireFormat.maxFra
meSize=1048576&needClientAuth=false"/>
    <transportConnector name="wss"
uri="wss://0.0.0.0:9884?maximumConnections=1000&wireFormat.maxFrameSi
ze=1048576&needClientAuth=false"/>
</transportConnectors>
```

如果连接失败，服务端提示针对断路器用户的 `Advisory` 错误。则可根据提示创建对应权限，或者如下方式关闭 `Advisory`：

```
<broker advisorySupport="false" .....
```

2.2. TopicRoot

为对系统的键值进行统一管理，约定一键值起始位置，称为 `TopicRoot`。该

值需要被写入集中器中。

假设 TopicRoot 设置为 **smart/devs**，某一集中器编码为 **202005010001**，则该集中器对应的各类 Topic 如下：

Topic	说明
smart/devs/202005010001/fromdev	管理模块订阅，本集中器发布。 本集中器在此发布配置请求、采集的实时电力相关数据、集中器及线路的各类状态信息等等。
smart/devs/202005010001/todev	管理模块发布，本集中器订阅。 管理在此发布各类配置、线路操作命令等信息给本集中器。

提示：在集中器客户端发布的一些 Message 中，包含有该集中器编码键值，管理应该验证 Message 中的集中器编码键值是否与 Topic 的编码数值是否一致，以及时发现异常消息，保证系统安全性。

2.3. QoS

消息约定以 QoS0 级别收发。

2.4. MQTT 客户端工具

建议使用 MQTT.fx(<http://mqttfx.jensd.de/index.php>)软件来帮助 MQTT 相关程序的开发和调试。

3. 集中器配置

集中器在使用之前，需要配置一些信息。配置方法请参见相关文档说明。提示，发送串口命令时需要带上回车换行。

和 MQTT 协议集中器相关的配置项如下：

3.1. 编码

set sn <编码>

例如: set sn 202005010001

编码必须是 12 位的数字。建议采用生产厂商缺省的编号模式: yyyymmddssss(4 位年份 2 位月份 2 位日 4 位流水号)。

编码在集中器出厂时应该已经设定, 不建议再另行修改!

get sn 命令可以查看集中器当前编码。

3.2. MQTT 服务器

set server <MQTT 服务器> <端口>

例如: set server rejuvee.com.cn 8883

MQTT 服务器项可以指定服务器名或 IP, 端口为 MQTT 连接端口。

get server 命令可以查看集中器当前的 MQTT 服务器设置。

3.3. MQTT 登录密码

set user <登录密码>

例如: set user 0501passwd

如上所述, 集中器固定以其编码作为登录用户名, 所以不需要设置登录用户名, 只需要设置登录密码。该用户和密码需要事先在 MQTT server/broker 端创建, 详见前面章节说明。

get user 命令可以查看集中器的 MQTT 登录密码。

3.4. TopicRoot

set topic <TopicRoot>

例如: set topic smart/devs

关于 TopicRoot, 详见前面章节说明。

get topic 命令可以查看集中器的 TopicRoot。

3.5. NTP

注意，此配置命令需要集中器达到特定版本。

```
set ntp <NTP 服务器> <端口> <时区>
```

例如：set ntp ntp1.aliyun.com 123 8

集中器配置有缺省的公网 NTP 设置，一般不需要修改。如果使用环境无法访问公网，则可以设置为内部的 NTP 服务器。建议使用 NTP 来对集中器进行自动校时。

get ntp 命令可以查看集中器的 NTP 设置。

4. 集中器在线升级包

将在官网实时发布。

如果第三方系统可以接入互联网，则可指定官网的升级包 URL 来进行在线升级。如果第三方系统在专有网络中，则可以通过在官网下载升级包文件放置在自己的 Web 系统中，然后通过指定内部的 URL 来实现在线更新。