



<MQTT 功能使用手册 V1.0>

作者	蒯鑫	 中科时代 基于PC技术的工智机新时代 深圳市南山区粤海街道百度国际大厦西塔楼 官网: www.sinsegye.com.cn 邮箱: Sales@sinsegye.com.cn 热线电话: 400-013-2158
日期	2024.11.29	
版本	V1.0	
Email		

更新说明:

本文记录了<MQTT 功能>的变更情况。

<2024.11.29> / <V1.0>

功能	变更类型	说明	相关文档
MQTT client 功能块 连接功能 订阅消息 发布消息 取消订阅 保留消息 遗嘱消息	<新增>	初版 V1.0	

目录

前言	1
一、文件说明	1
二、安全声明	1
概述	3
安装卸载	5
一、安装要求	5
二、安装过程	5
三、更新安装	16
四、卸载过程	16
技术说明	22
一、快速启动	22
（一）本例软、硬件配置	22
（二）本例实验操作步骤	22
二、示例	31
创建工程	31
建立 MQTT 客户端	36
连接 MQTT Broker	43
发布消息	46
订阅消息	50
取消订阅	56
保留消息	59
遗嘱消息	61
功能介绍	65
附录	71
一、工控机安装 deb 组件方法	71
二、支持与服务	76

前言

一、文件说明

本说明专为熟悉相关国家标准且经过专业培训的控制与自动化技术专家而制定。

在安装与调试部件时，务必仔细审阅所有相关文件及以下说明。

合格人员应始终采用最新的有效文档进行操作。

责任人员必须确保所述产品的应用或使用完全符合所有安全要求，涵盖所有相关法律法规、指导原则及标准。

1、免责声明

本文件经过精心编制，但鉴于所描述产品处于持续的开发与升级过程中，中科时代（深圳）计算机系统有限公司保留随时对文件进行修改和更新的权利，且无需事先通知。请注意，禁止依据数据图及本文件描述对已交付的产品进行任何改动。

对于因使用或信赖本手册所载明或未明示的信息而造成的任何损失或损害，中科时代计算机系统有限公司不承担任何责任。

2、版权所有

本手册的所有权归中科时代计算机系统有限公司所有。未经书面许可，任何人不得以任何形式复制、分发、翻译或以其他方式使用本手册的全部或部分内容。

本手册受版权法保护。任何对本手册内容的复制、分发、翻译、展示、表演、演绎或使用，无论出于何种目的，均需得到中科时代计算机系统有限公司的明确许可。未经许可，任何行为均视为侵犯中科时代计算机系统有限公司的版权。

二、安全声明

1、安全规程

为了您的安全，请阅读以下说明。始终遵守产品特定的安全说明，您可以在本文档的适当位置找到这些说明。

2、责任免除

所有组件都提供了硬件和软件配置。不允许对文件中所述以外的硬件或软件配置进行修改，中科时代不对文件所述外的硬件或软件负责。

3、人员资格

本说明仅适用于熟悉适用国家标准的经过培训的控制、自动化和驱动技术专家。

4、信号词

文档中使用的信号词分类如下。为了防止人员和财产受到伤害和损害，请阅读并遵守安全和警告通知。

5、个人伤害警示

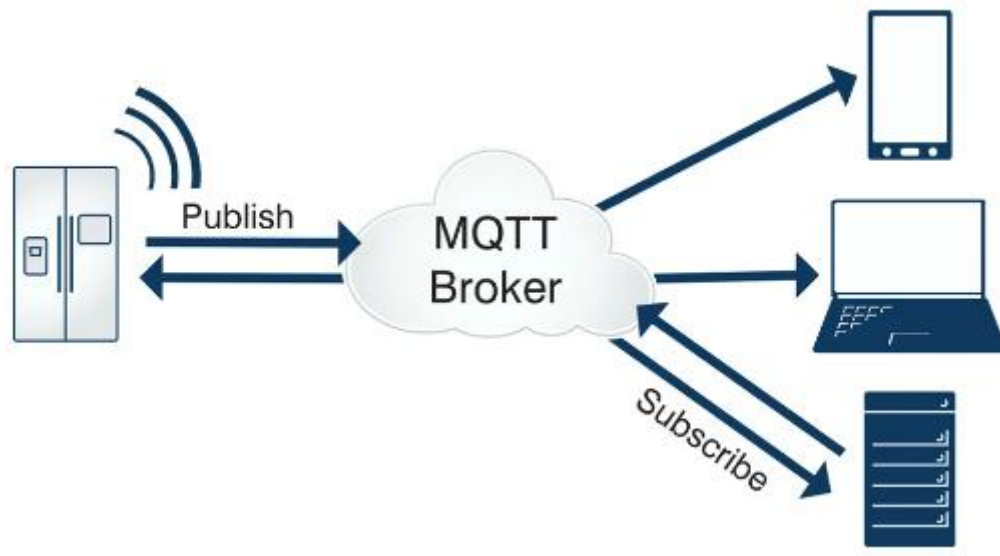
	<div><div>警告</div><div>危险的类型 说明不避开危险的后果 说明如何避免危险的发生</div></div>	警告表示一种潜在的危險情况, 如果不加以避免, 可能会导致严重的伤害或死亡。
	<div><div>注意</div><div>危险的类型 说明不避开危险的后果 说明如何避免危险的发生</div></div>	注意表示潜在的危險情况, 如果不避免, 可能会导致轻度受伤或中度受伤, 或导致设备损坏。
<div><div>提醒</div><div>危险的类型 说明不避开危险的后果 说明如何避免危险的发生</div></div>	注意表示一种潜在的危險情况, 如果不加以避免, 可能只导致设备的损坏。	

6、对财产或环境造成损坏的警告

<div><div><div>注意</div><div>危险的类型</div><div>说明不避开危险的后果</div><div>说明如何避免危险的发生</div></div><div></div></div>	环境、设备或数据可能会被损坏。
---	-----------------

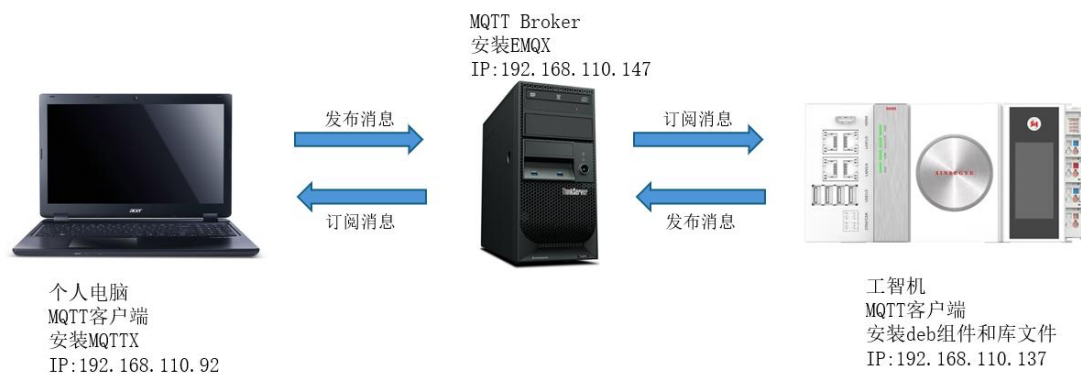
概述

MQTT (Message Queuing Telemetry Transport) 是一种轻量级、基于发布-订阅模式的消息传输协议，适用于资源受限的设备和低带宽、高延迟或不稳定的网络环境。它在物联网应用中广受欢迎，能够实现传感器、执行器和其它设备之间的高效通信。



- **QoS**
 - QoS (Quality of Service) 指消息传输的服务质量。分别可在消息发送端和消息消费端设置。
 - 发送端的 QoS 设置: 影响发送端发送消息到服务器消息队列 MQTT 版的传输质量。
 - 消费端的 QoS 设置: 影响服务器消息队列 MQTT 版服务端投递消息到消费端的传输质量。
 - QoS 包括以下级别:
 - QoS0: 代表最多分发一次。
 - QoS1: 代表至少达到一次。
 - QoS2: 代表仅分发一次。

本手册使用一台 WINDOWS 电脑作为 MQTT Broker，一台个人电脑和一台工控机作为客户端，实现相互订阅和发布信息。如下图所示：



下表概述了本手册各个产品组件

产品组件	描述说明
iotmqtt_0.0.2_amd64.deb	工控机 MQTT 组件
MQTT_Client.projectarchive	PLC 工程存档
sf4202_iotmqtt_1.0.2.0.library	PLC 工程库文件
emqx-5.3.2-windows-amd64.zip	EMQX 压缩文件 (MQTT Broker)
MQTTX-Setup-1.11.0-x64.exe	MQTT 客户端安装程序

安装卸载

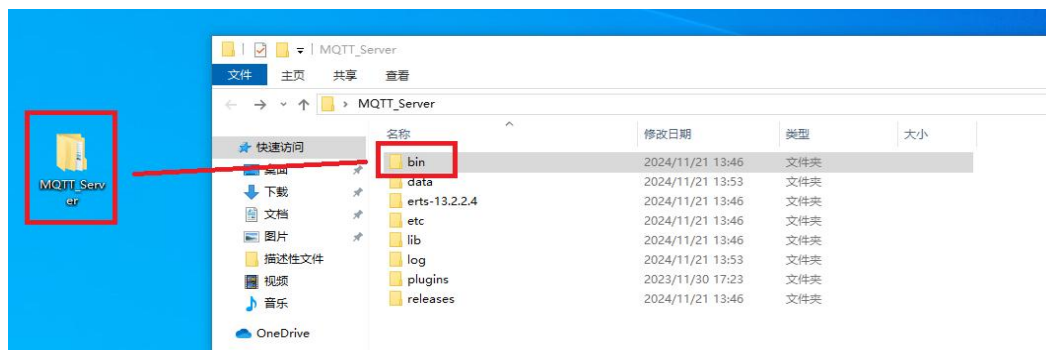
一、安装要求

- 中科时代出厂的工智机的自带系统；
- SX、SP 全系工智机
- 工智机可以访问互联网；

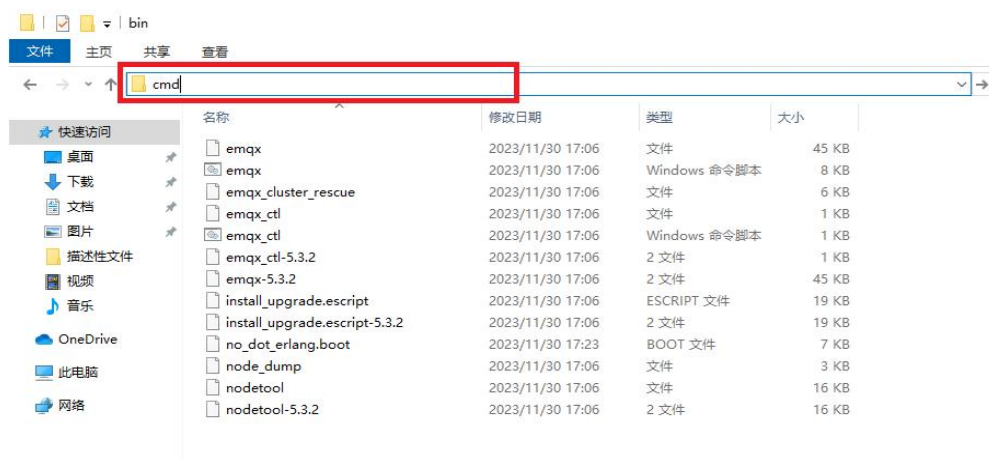
二、安装过程

1、安装 EMQX (MQTT Broker)

将官网下载的 emqx-5.3.2-windows-amd64.zip 进行解压缩，自行解压到相应的文件夹中。



打开文件夹目录 bin，在文件夹位置处输入“cmd”，回车。



在命令行中输入——emqx start——回车

```
C:\Windows\System32\cmd.exe
Microsoft Windows [版本 10.0.19045.5131]
(c) Microsoft Corporation. 保留所有权利。

C:\Users\sinsegye\Desktop\MQTT_Server\bin>emqx start_
```

启动完成

```
C:\Windows\System32\cmd.exe
Microsoft Windows [版本 10.0.19045.5131]
(c) Microsoft Corporation. 保留所有权利。

C:\Users\sinsegye\Desktop\MQTT_Server\bin>emqx start
EMQX_NODE__DB_ROLE [node.role]: core
EMQX_NODE__DB_BACKEND [node.db_backend]: mnesia

C:\Users\sinsegye\Desktop\MQTT_Server>
```

EMQX 端口号:

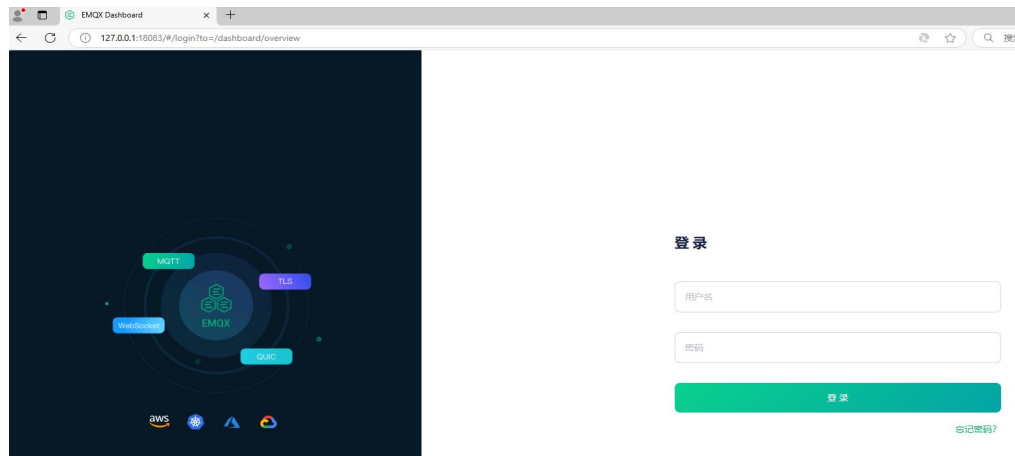
端口号	说明
1833	TCP 端口
8883	WebSocket 端口
8884	WebSocket Secure 端口
8883	SSL/TLS 端口
18083	Broker 的 Dashboard 访问端口

可以通过 18083 端口访问 MQTT Broker 可视化界面

本机访问：127.0.0.1:18083/

外部访问：192.168.110.147:18083/

(192.168.110.147 是本例 MQTT Broker 的 IP 地址)



用户名：admin

密码：public

首次登录会提示修改密码，也可以选择跳过。

修改密码

为保护账户安全，请修改默认密码：

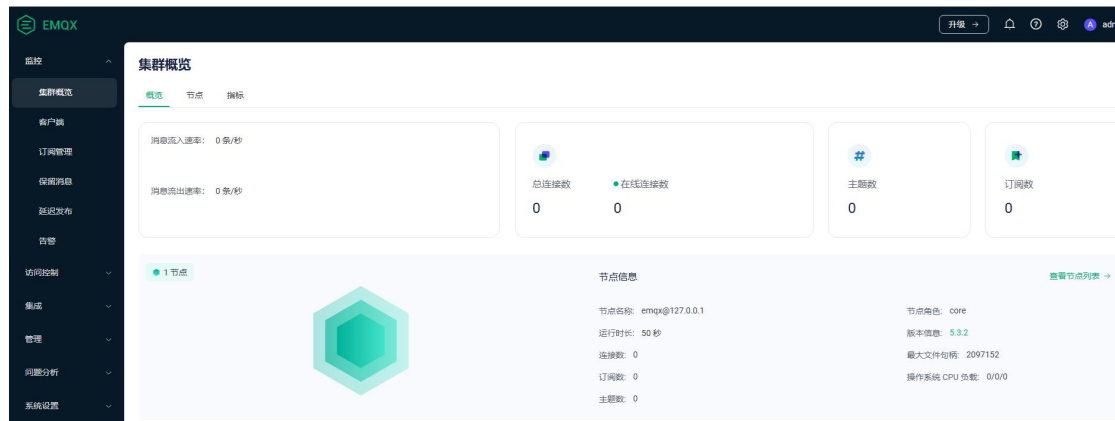
- 密码长度必须为 8 ~ 64 个字符
- 至少包含字母、数字、符号中的两种

确定

跳过

下次登录时将再次提示您修改默认密码，您也可以稍后在系统 > 用户菜单中进行修改。

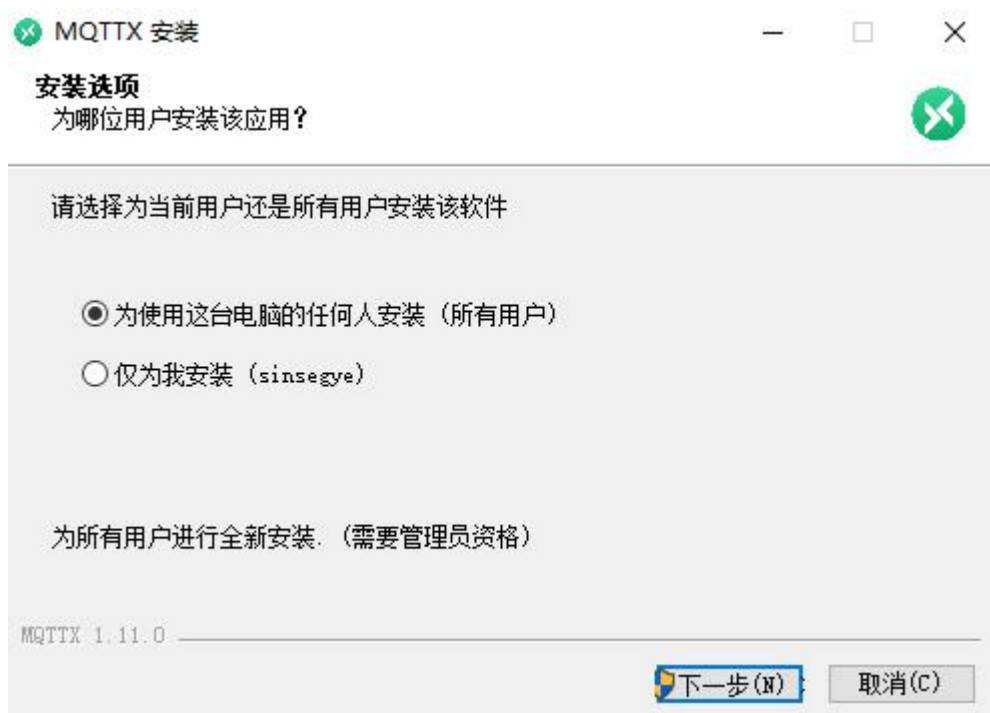
Dashboard 界面，可以进行 Broker 的设置以及查看客户端连接情况、消息订阅等等。



2、WINDOWS 个人电脑安装 MQTTX

本手册使用 MQTTX 作为客户端进行消息的发布和订阅。

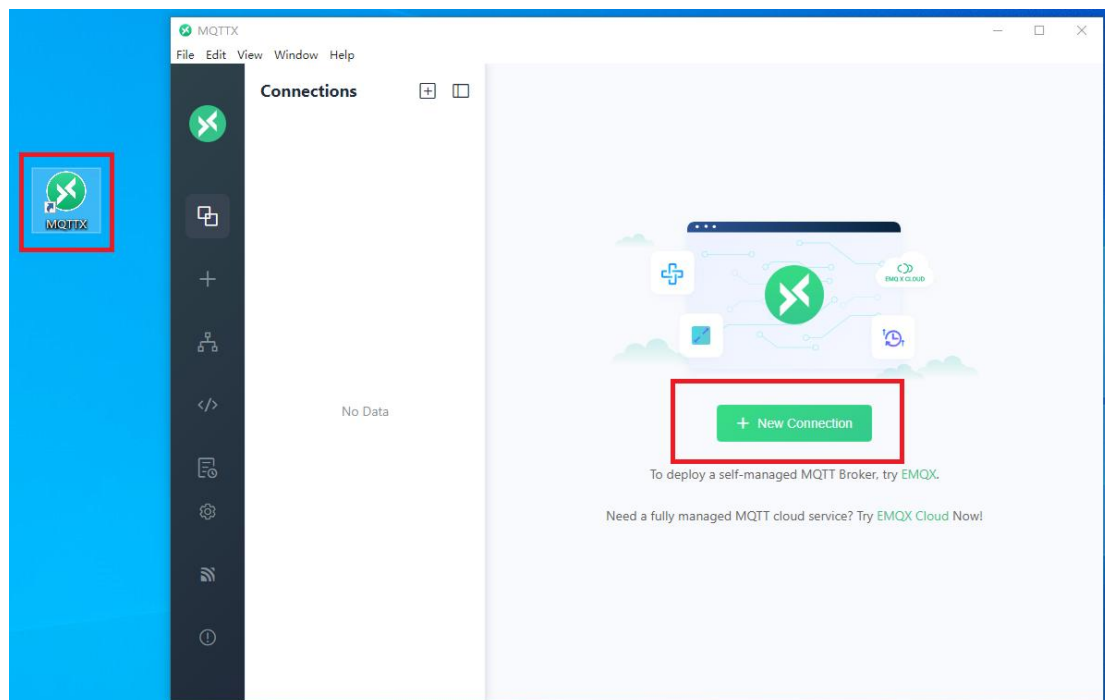
打开从官方网站下载的 MQTTX-Setup-x64.exe 程序，选择安装选项下一步。



选择安装目录—安装

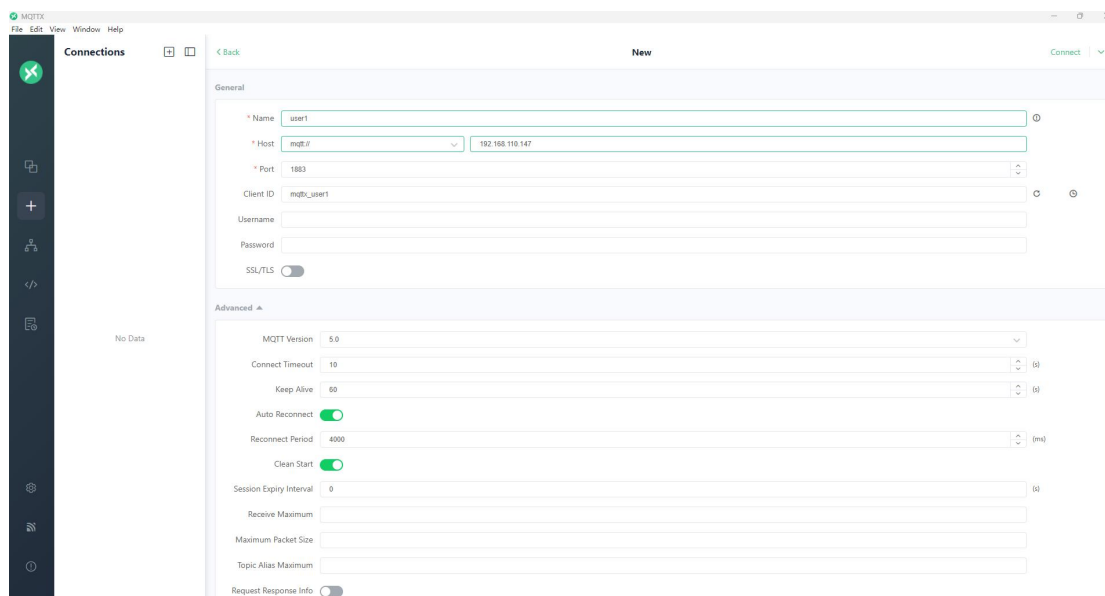


安装完成后，点击 MQTTX 图标——新建连接。



填写客户端

参数	说明	设定例
Name	客户名字	user1
Host	主机 IP 地址	192.168.110.147
Port	端口	1883
Client ID	客户端 ID，必须是唯一标识	mqtt_user1



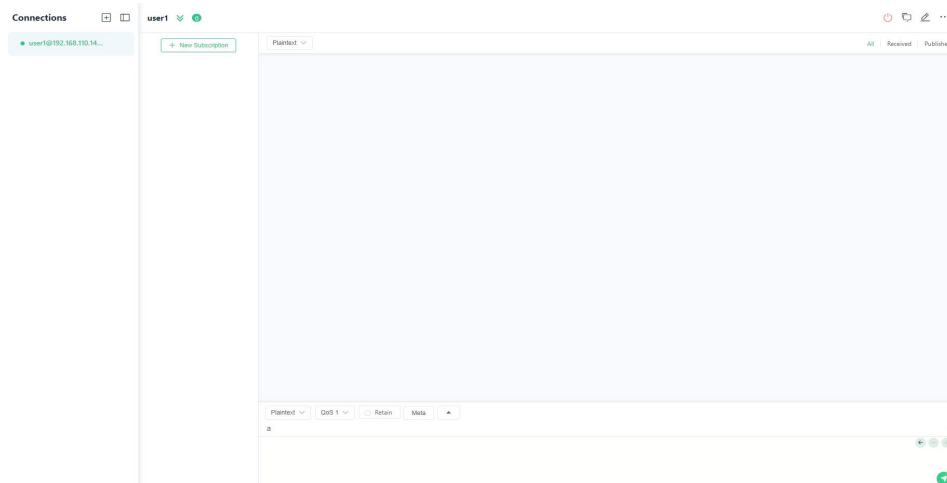
The screenshot shows the MQTTX application window with the 'New' connection configuration panel open. The 'General' tab is selected, and the following fields are filled out:

- Name:** user1
- Host:** mqtt:// 192.168.110.147
- Port:** 1883
- Client ID:** mqtt_user1
- Username:** (empty)
- Password:** (empty)
- SSL/TLS:** (disabled)

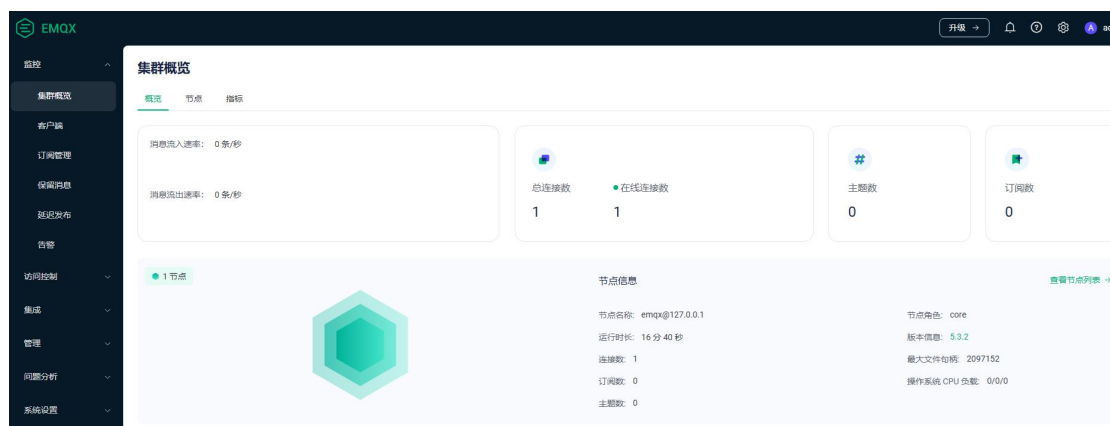
The 'Advanced' tab is also visible, showing the following settings:

- MQTT Version:** 5.0
- Connect Timeout:** 10 (s)
- Keep Alive:** 60 (s)
- Auto Reconnect:** (enabled)
- Reconnect Period:** 4000 (ms)
- Clean Start:** (enabled)
- Session Expiry Interval:** 0 (s)
- Receive Maximum:** (empty)
- Maximum Packet Size:** (empty)
- Topic Alias Maximum:** (empty)
- Request Response Info:** (disabled)

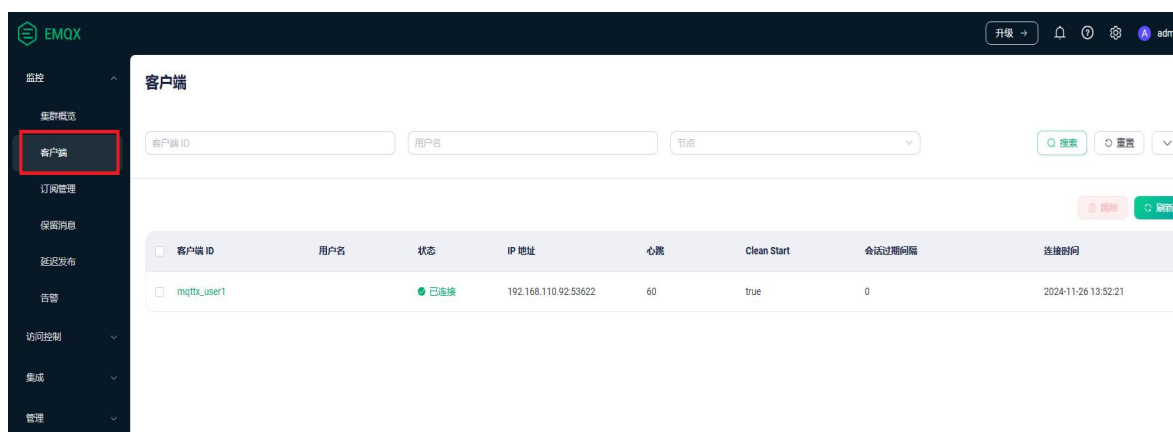
点击右上角“Connect”进行连接，连接成功后：



在 MQTT Broker 界面可以看到连接数变为 1.



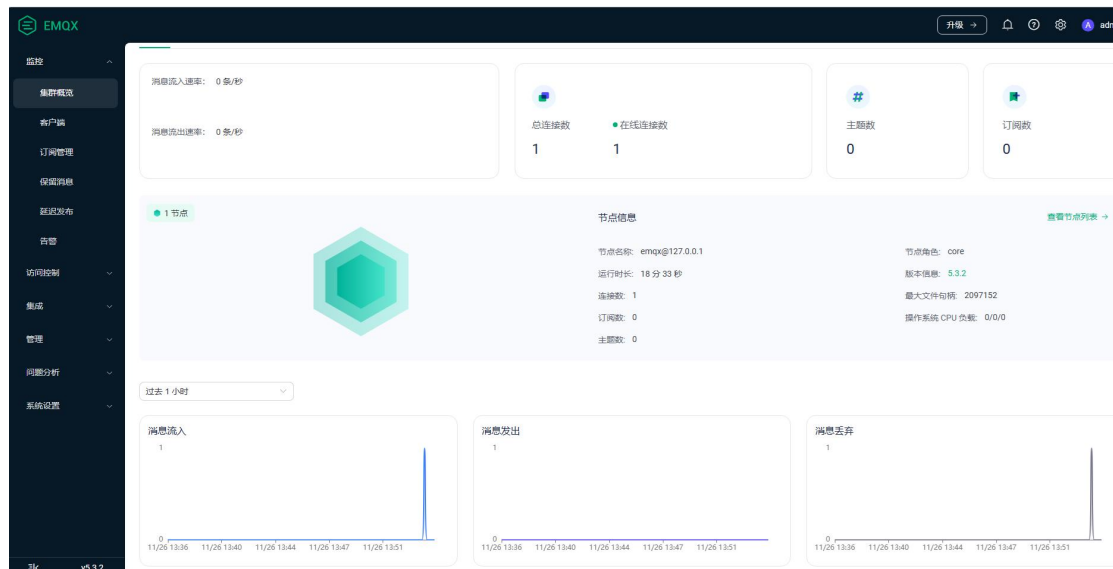
点击客户端，显示客户端 ID、连接情况、IP 地址等信息：



回到 MQTTX 客户端，发布消息，格式为 Plaintext，主题为 a，负载内容为 a=0



Broker 界面可以看到消息流入数为 1，由于没有客户端订阅主题 a，所有消息被丢弃。



3、工智机安装 iotmqtt_0.0.2_amd64.deb 组件

参考附录一，使用 Windows cmd 输入命令将 iotmqtt_0.0.2_amd64.deb 文件安装至工智机中。

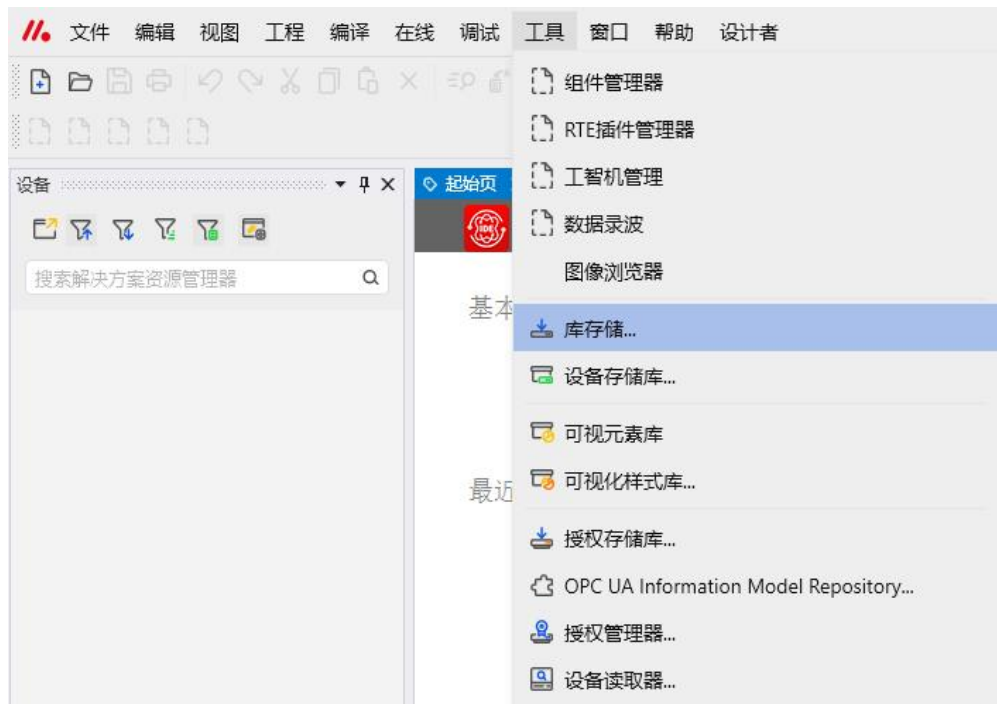
```
scp -P 2224 iotmqtt_0.0.2_amd64.deb sinsegye@192.168.110.137:~  
ssh -p 2224 sinsegye@192.168.110.137  
sudo dpkg -i iotmqtt_0.0.2_amd64.deb  
sudo nano /usr/local/etc/SinsegyeRTE/SinsegyeRTE.cfg  
修改 ComponentManager  
sudo reboot
```



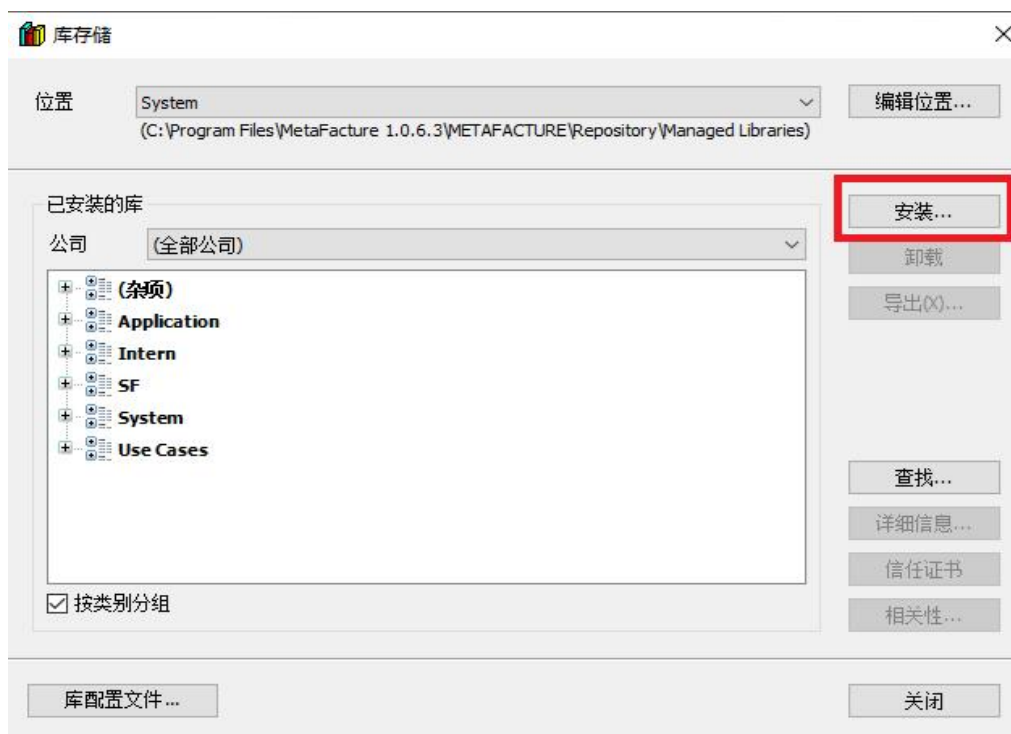
```
[ComponentManager] 6504  
# Component.0=SE4100  
Component.0=CmpLedManager  
Component.1=retainDaemon  
Component.2=CmpCanBusUtils  
Component.3=CmpSinsegyeLibs  
Component.4=CmpBroadcastServer  
Component.5=sf4401db  
Component.6=sf2100  
Component.7=acoreadwrite  
Component.8=iotmqtt 4212
```


4、工控机安装 iotmqtt.library 库文件

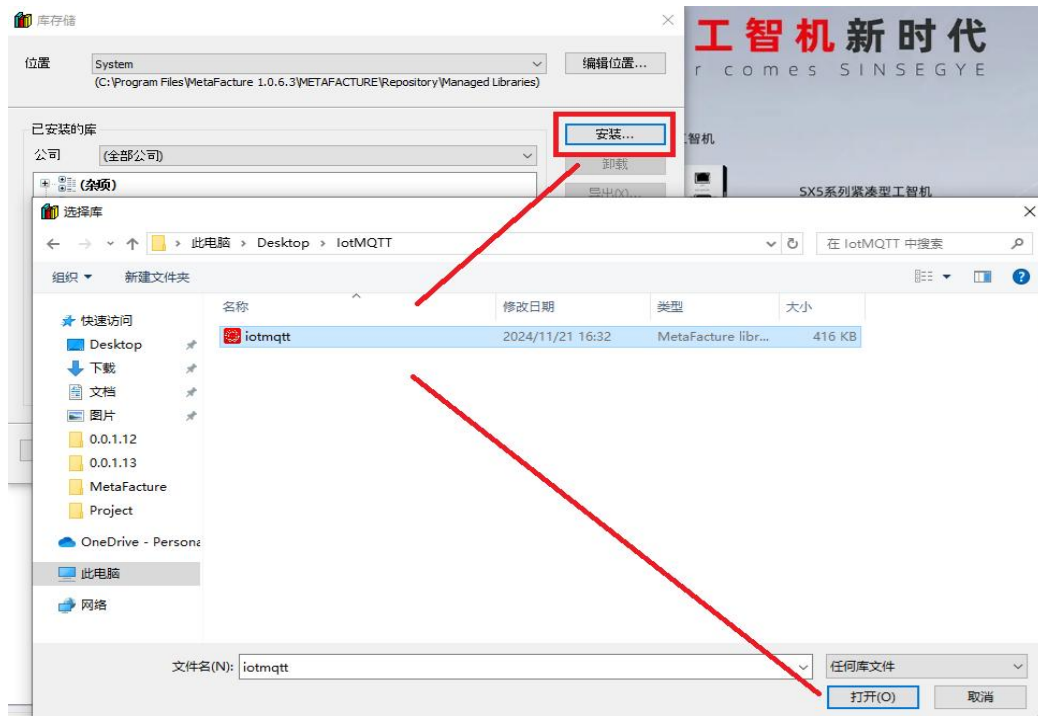
打开 MetaFacture 程序，选择“工具”——“库存储”



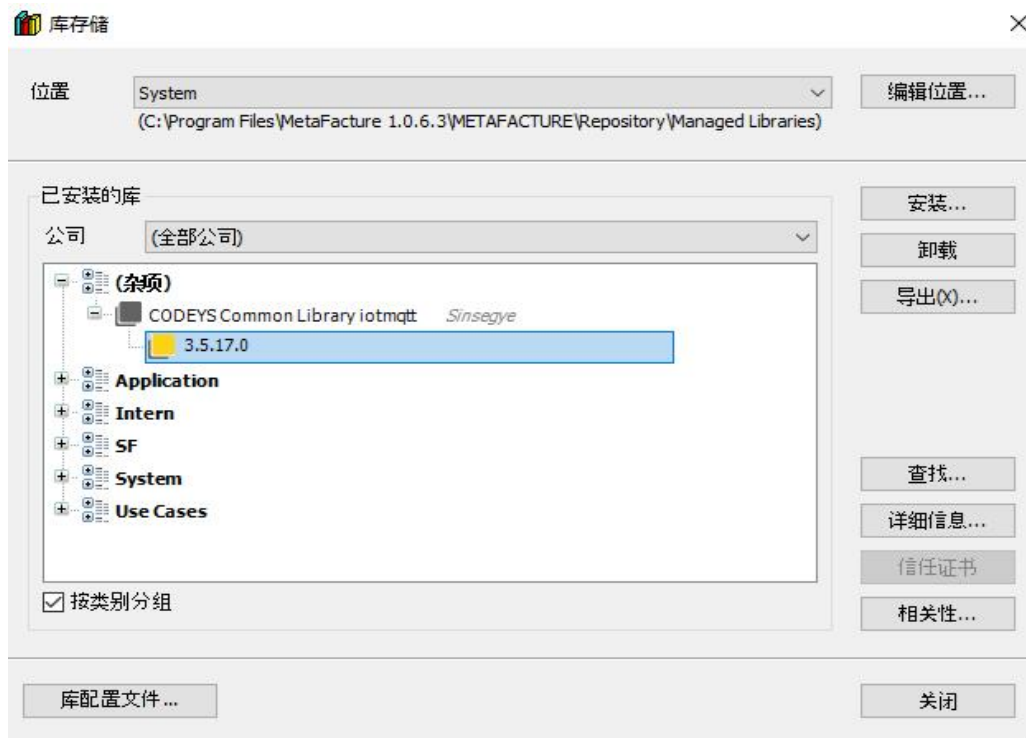
选择“安装”



选择从中科时代官方下载的 **iotmqtt.library** 库文件



安装完成



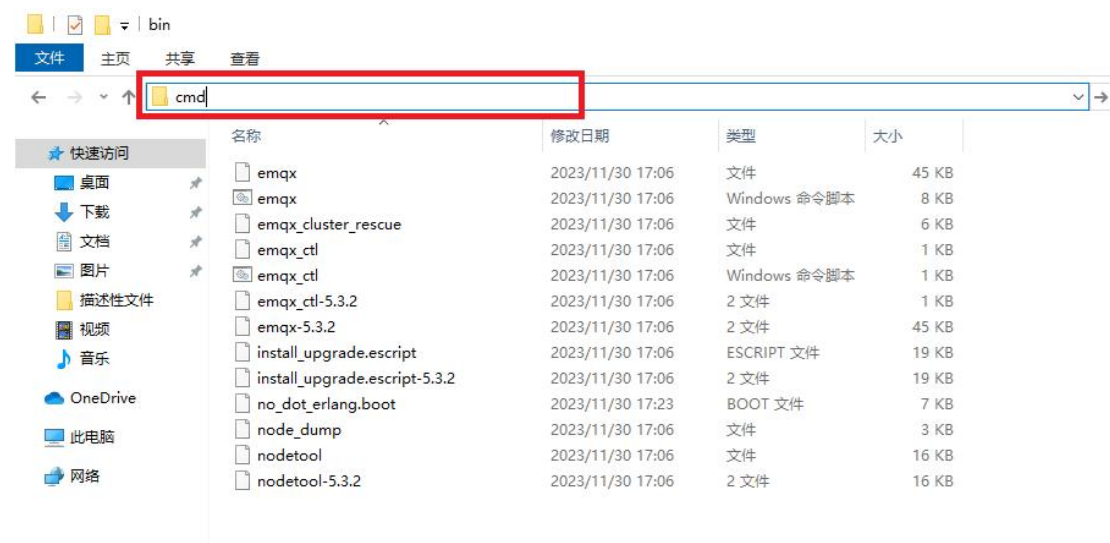
三、更新安装

参考上一章《二、安装过程》，将新的 deb 文件和库文件覆盖安装即可更新。

四、卸载过程

1、停止 EMQX (MQTT Broker)

打开 EMQX 文件夹——打开目录 bin，在文件夹位置处输入“cmd”，回车。



在命令行中输入——emqx stop——回车



等待片刻后，停止 EMQX 服务。

```

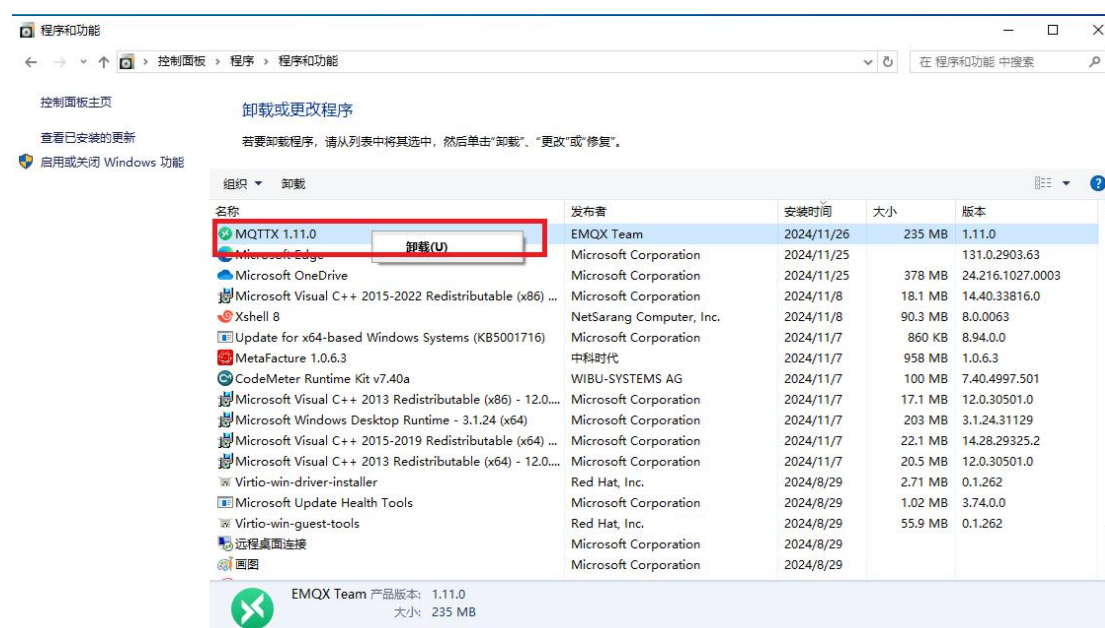
C:\Windows\System32\cmd.exe

Microsoft Windows [版本 10.0.19045.5131]
(c) Microsoft Corporation. 保留所有权利。

C:\Users\sinsegye\Desktop\MQTT_Server\bin>emqx stop

C:\Users\sinsegye\Desktop\MQTT_Server\bin>
  
```

2、使用控制面板卸载 MQTTX







3、卸载工控机 iotmqtt_0.0.2_amd64.deb 组件

3.1 使用 dpkg -r 命令进行卸载

```
sudo dpkg -r iotmqtt
```

3.2 修改 RTE 的配置文件，ComponentManger 模块下去掉 iotmqtt

```
sudo nano /usr/local/etc/SinsegyeRTE/SinsegyeRTE.cfg
```

删除 Component.8=iotmqtt



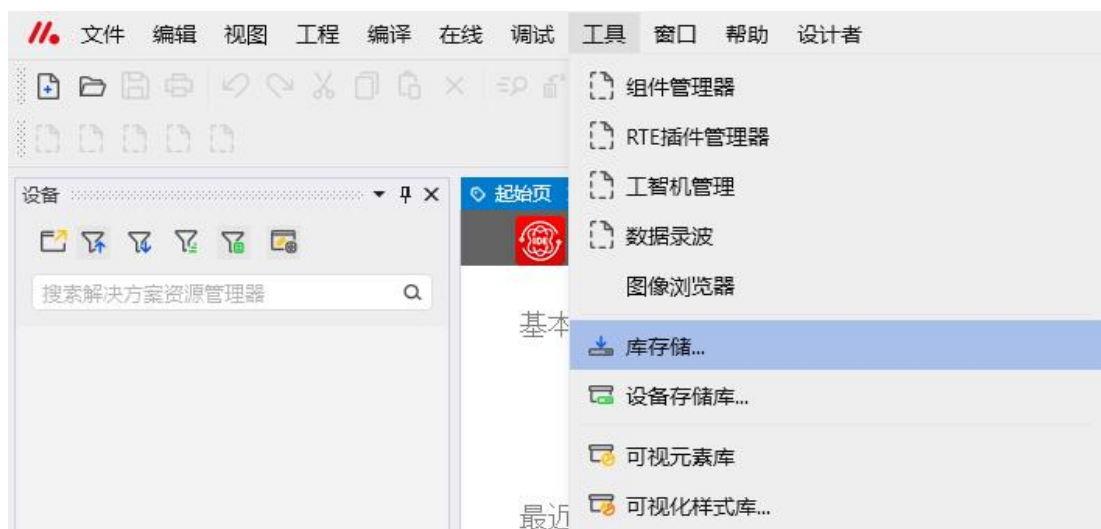
```
[ComponentManager]
+ Component.0=SE4100
Component.0=CmpLedManager
Component.1=retainDeamon
Component.2=CmpCanBusUtils
Component.3=CmpSinsegyeLibs
Component.4=CmpBroadcastServer
Component.5=sf4401db
Component.6=sf2100
Component.7=acbreawrite
Component.8=iotmqtt
```

3.3 重启工智机

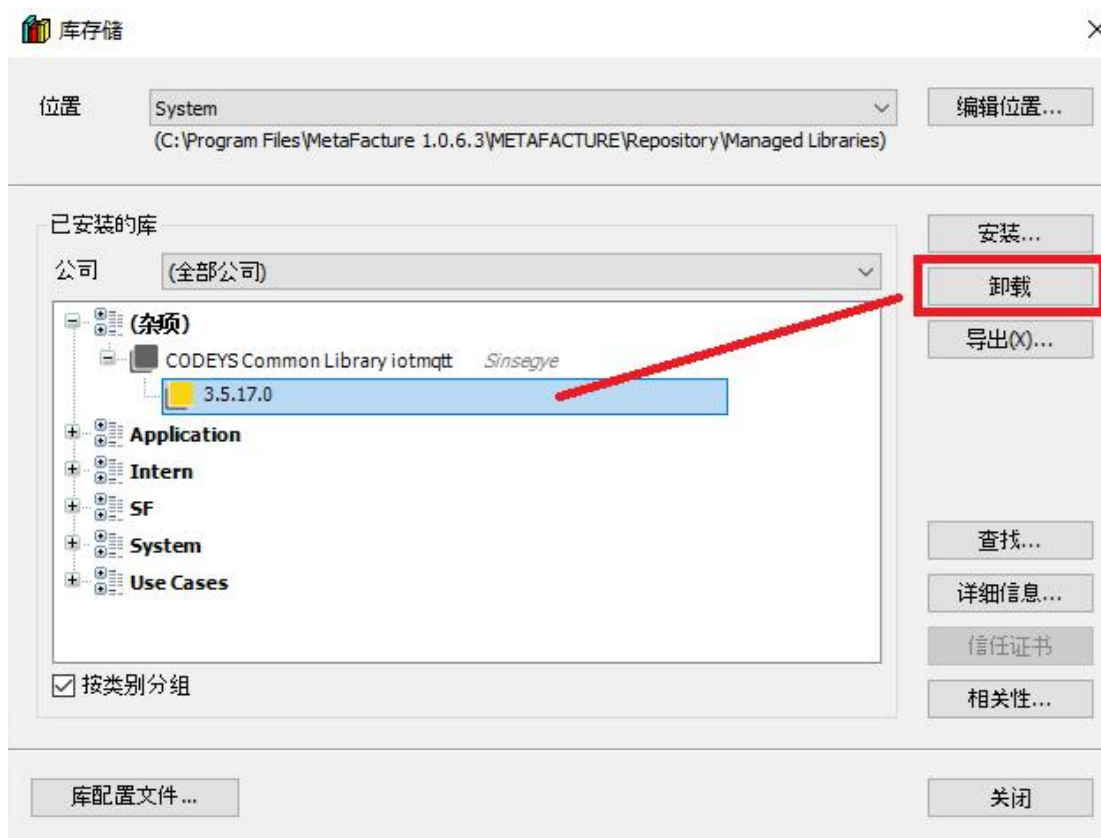
```
sudo reboot
```

4、卸载 MetaFacture 中的库文件 iotmqtt.library

4.1 打开软件，点击“工具”—“库存储”



4.2 在“杂项”中找到需要卸载的库文件，选中后点击“卸载”



技术说明

一、快速启动

（一）本例软、硬件配置

硬件：

1. 工控机 SX20
2. 服务端电脑 (WINDOWS)
3. 个人电脑 (WINDOWS)

软件：

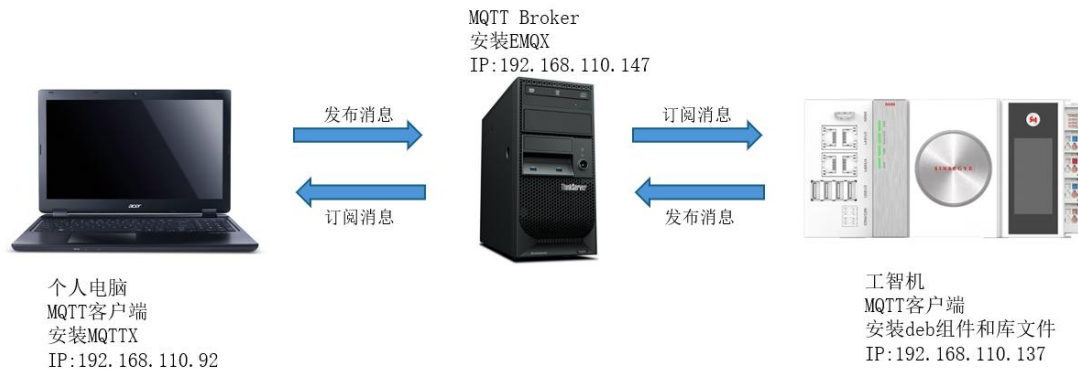
1. 服务端软件 emqx-5.3.2
2. 客户端软件 MQTTX 1.11.0
3. 工控机组件 iotmqtt_0.0.2_amd64.deb
4. PLC 库文件 iotmqtt.library
5. PLC 存档工程 MQTT_Client.projectarchive
6. 中科时代 MetaFacture V1.0.6.3

（二）本例实验操作步骤

1.实验要求

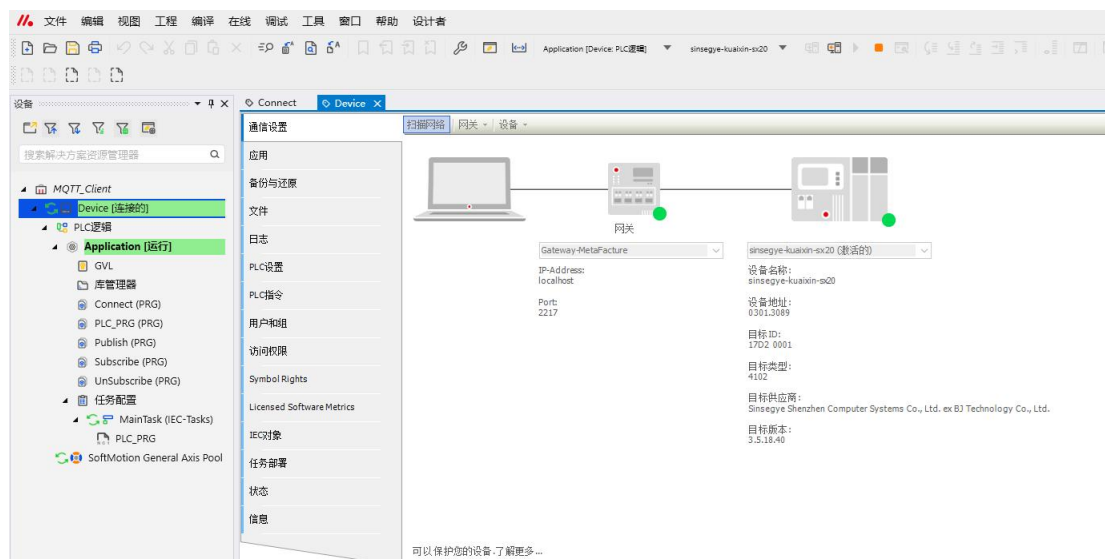
- a.MQTT Broker 成功运行 EMQX;
- b.个人电脑安装 MQTTX, 并与 Broker 成功连接, 并订阅主题“zkzd”;
- c.工控机成功安装 iotmqtt_0.0.2_amd64.deb 组件
- d.MetaFacture 成功安装库文件 iotmqtt.library

2.实验原理图



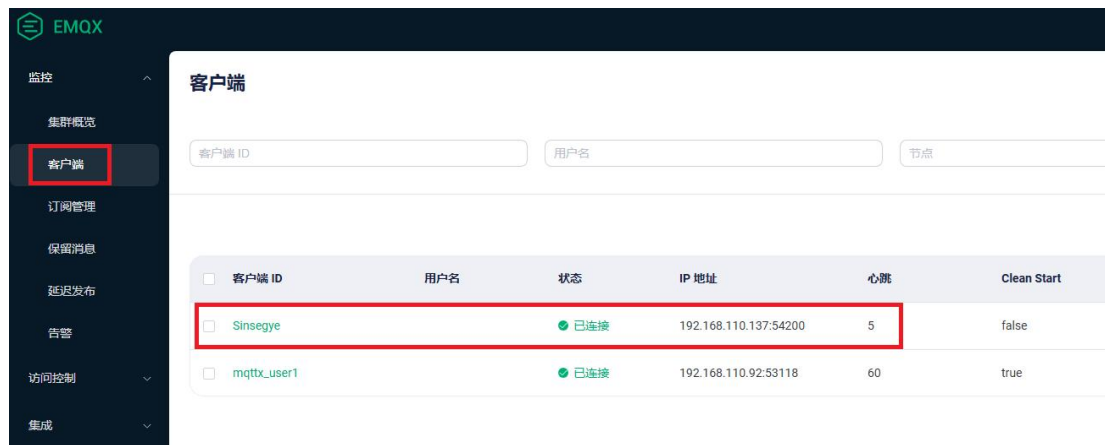
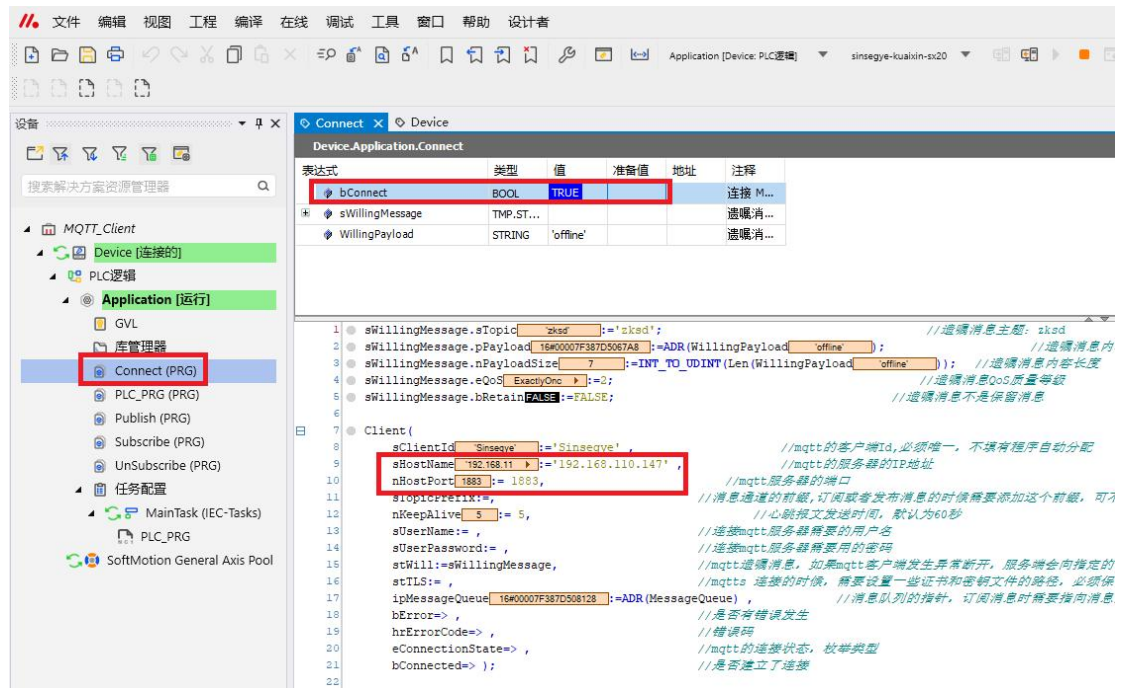
3.实验步骤

3.1 使用 MetaFactory 打开 PLC 存档工程“MQTT_Client.projectarchive”。登录设备，下载 PLC 程序并启动运行。

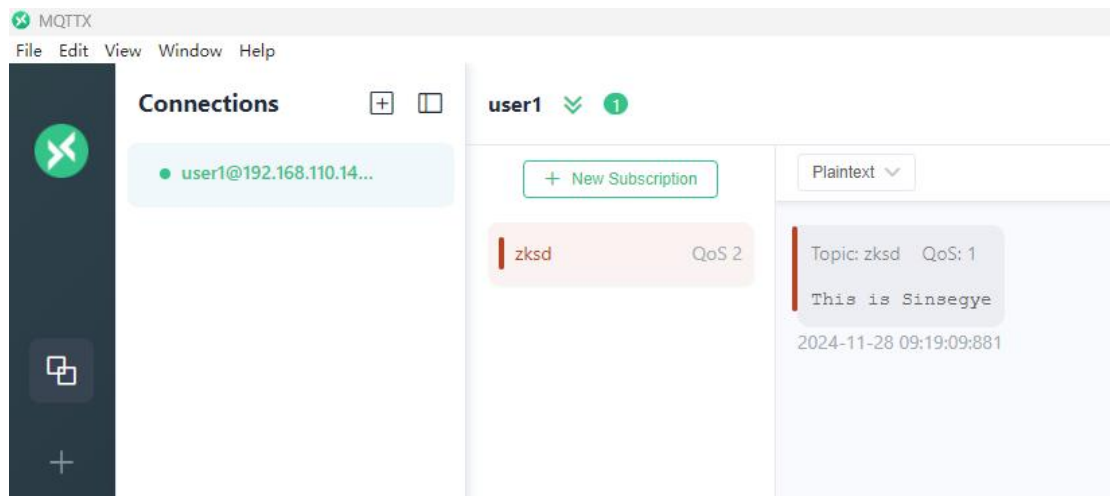
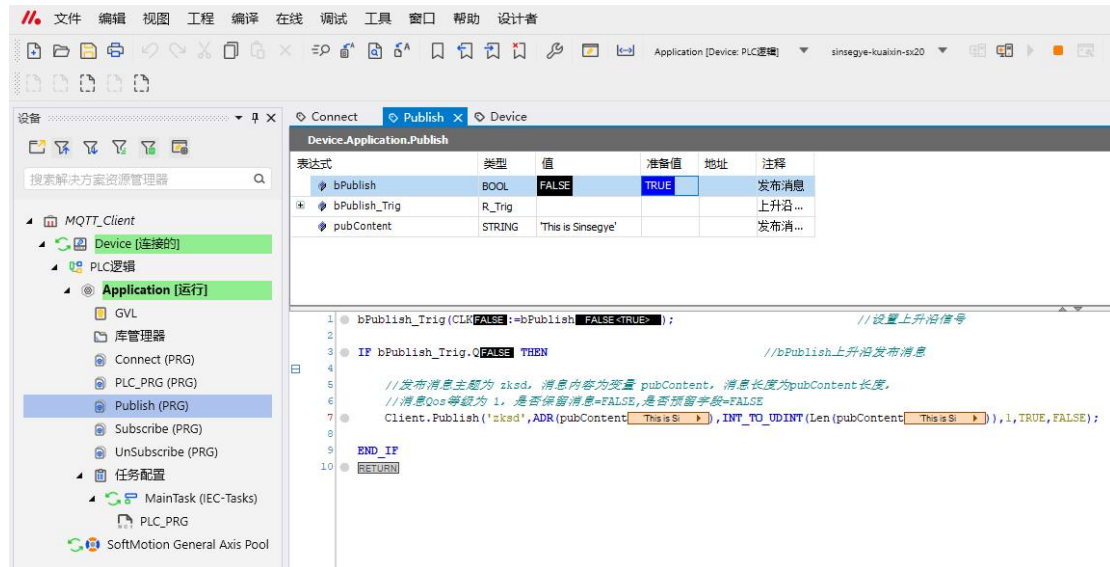


3.2 点击 PLC 程序“Connect”，配置 Broker 端的 IP 地址和端口，将变量 bConnect 置 TRUE，工控机连接 Broker。

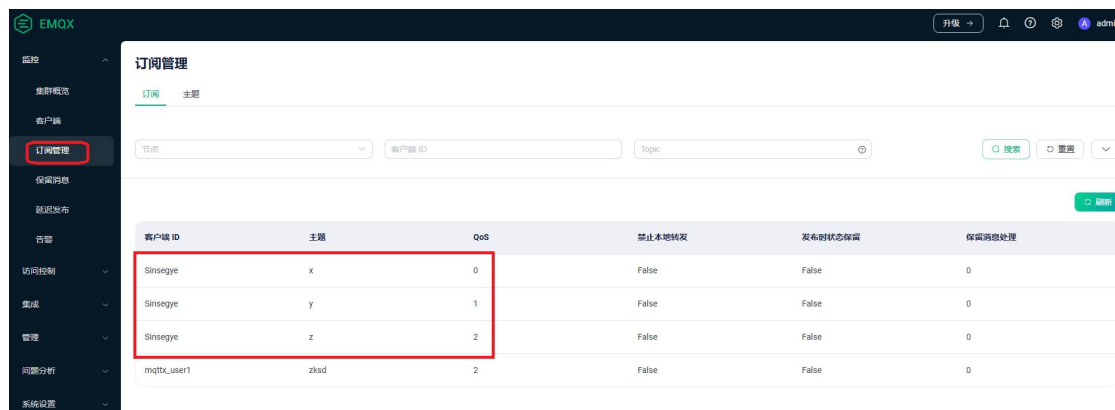
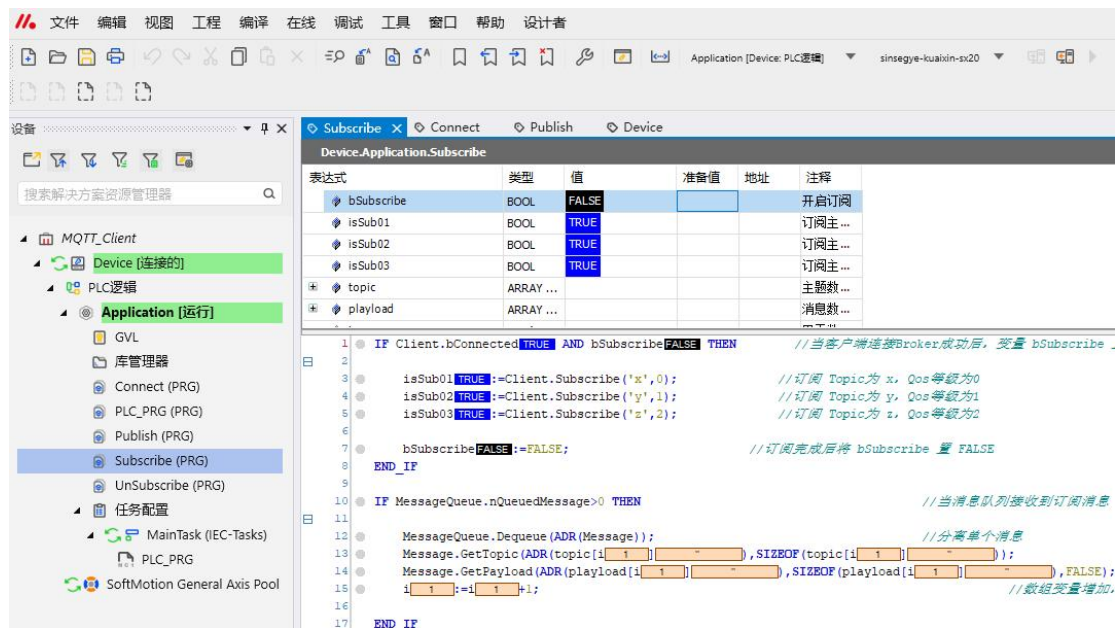
可以在 Broker Dashboard 界面查看到连接的客户端情况。



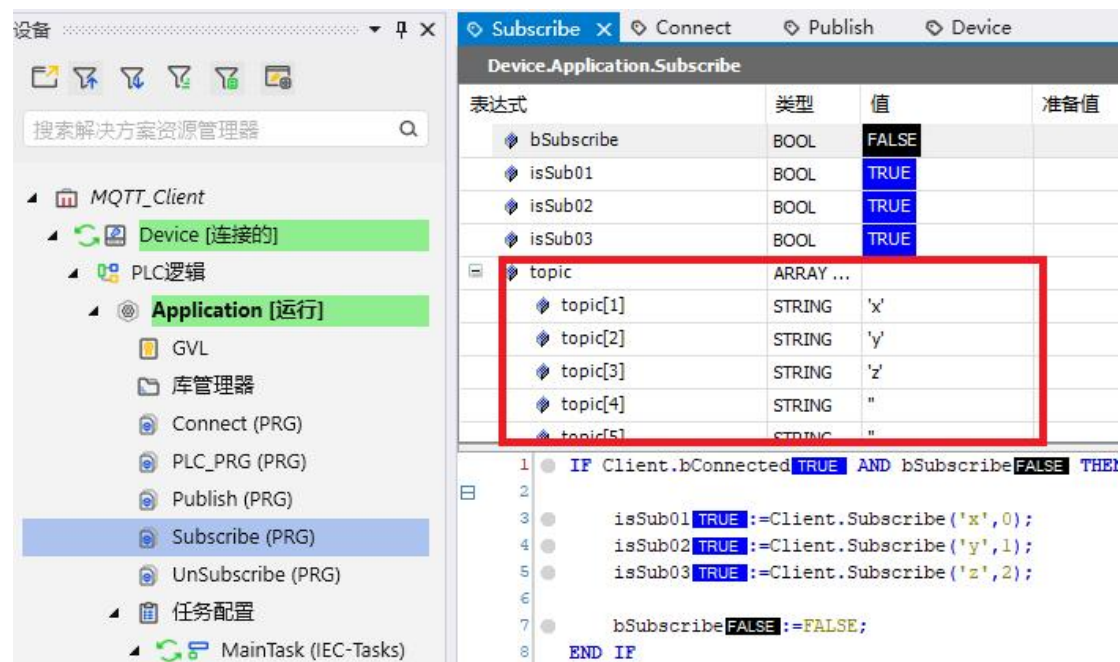
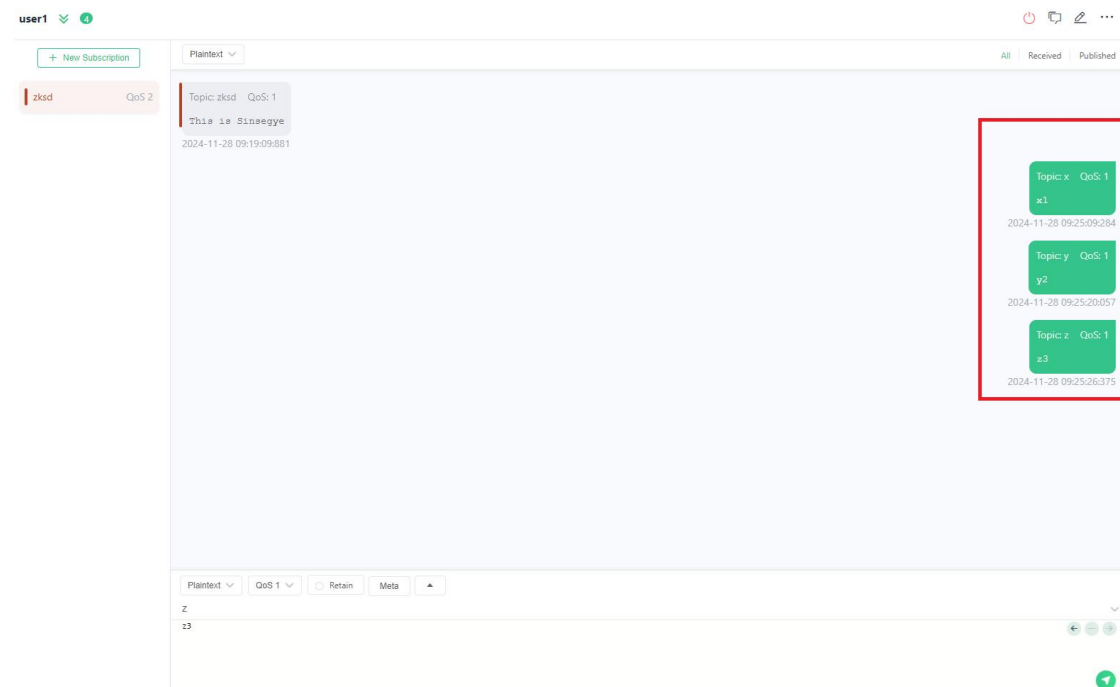
3.3 点击 PLC 程序“Publish”，将变量 bPublish 置 TRUE。在 MQTTX 客户端可以收到订阅消息。

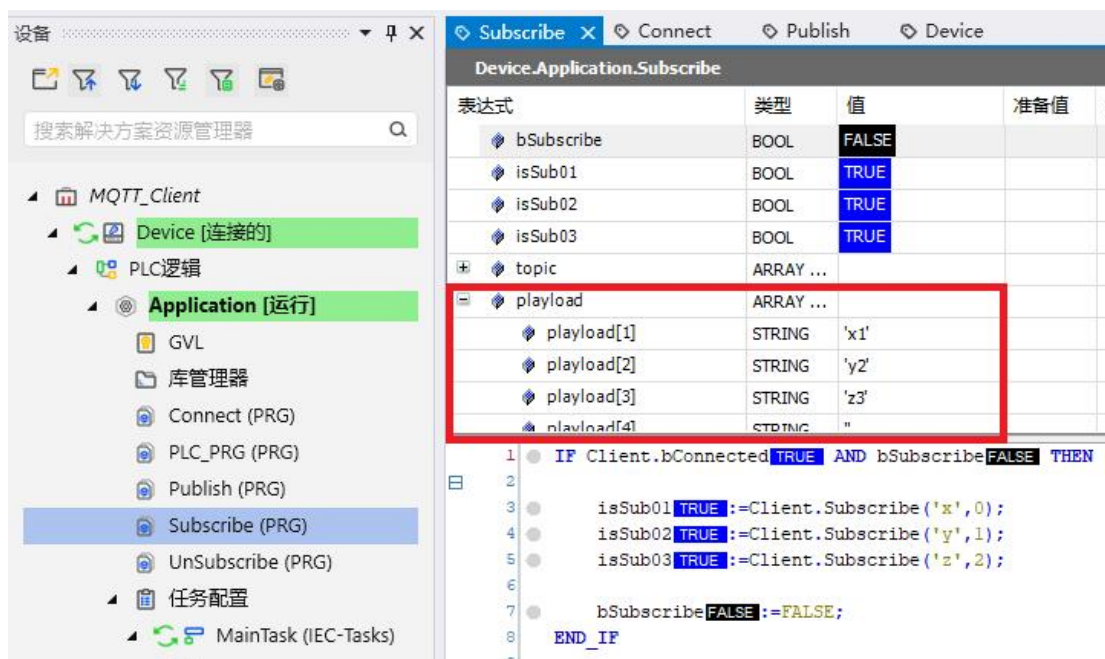


3.4 点击 PLC 程序“Subscribe”，将变量 bSubscribe 置 TRUE。订阅三个主题“x”“y”“z”。在 MQTTX 客户端中发送三个主题消息。PLC 程序变量 topic 和 payload 可以收到主题和内容。



发送主题消息：





3.5 点击 PLC 程序“Unsubscribe”，将变量 bUnSubscribe 置 TRUE。取消订阅“x”“y”“z”。在 Broker Dashboard 界面—订阅管理可以看到已经取消订阅。

文件 编辑 视图 工程 编译 在线 调试 工具 窗口 帮助 设计者

Application [Device: PLC逻辑] sinseg

设备

搜索解决方案资源管理器

- MQTT_Client
 - Device [连接的]
 - PLC逻辑
 - Application [运行]**
 - GVL
 - 库管理器
 - Connect (PRG)
 - PLC_PRG (PRG)
 - Publish (PRG)
 - Subscribe (PRG)
 - UnSubscribe (PRG)**
 - 任务配置
 - MainTask (IEC-Tasks)
 - PLC_PRG

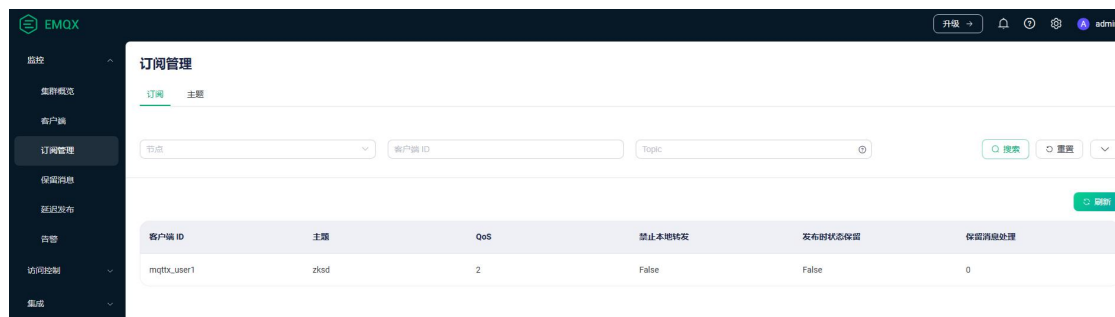
UnSubscribe x Subscribe Connect Publish Device

Device.Application.UnSubscribe

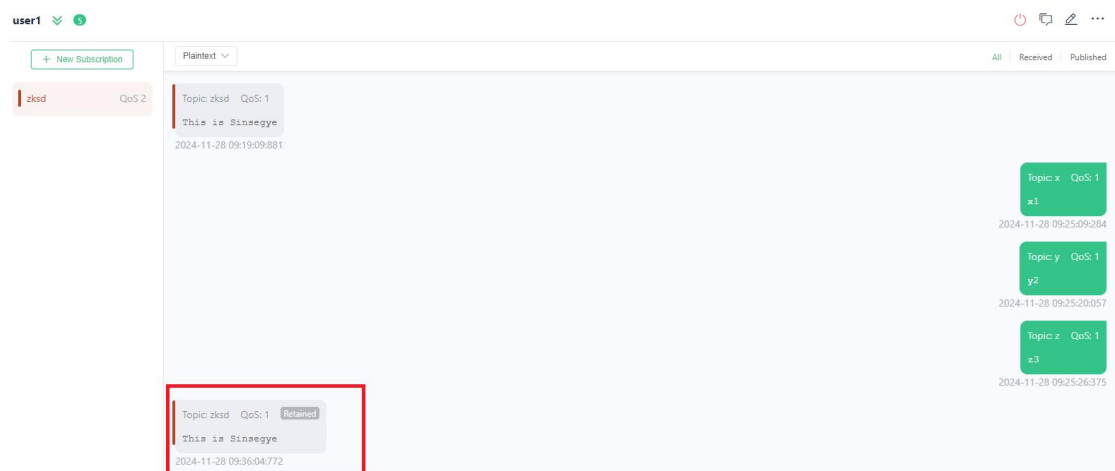
表达式	类型	值	准备值	地址	注释
bUnsubscribe	BOOL	FALSE			取消订阅
isUnsub01	BOOL	TRUE			是否取...
isUnsub02	BOOL	TRUE			是否取...
isUnsub03	BOOL	TRUE			是否取...

```

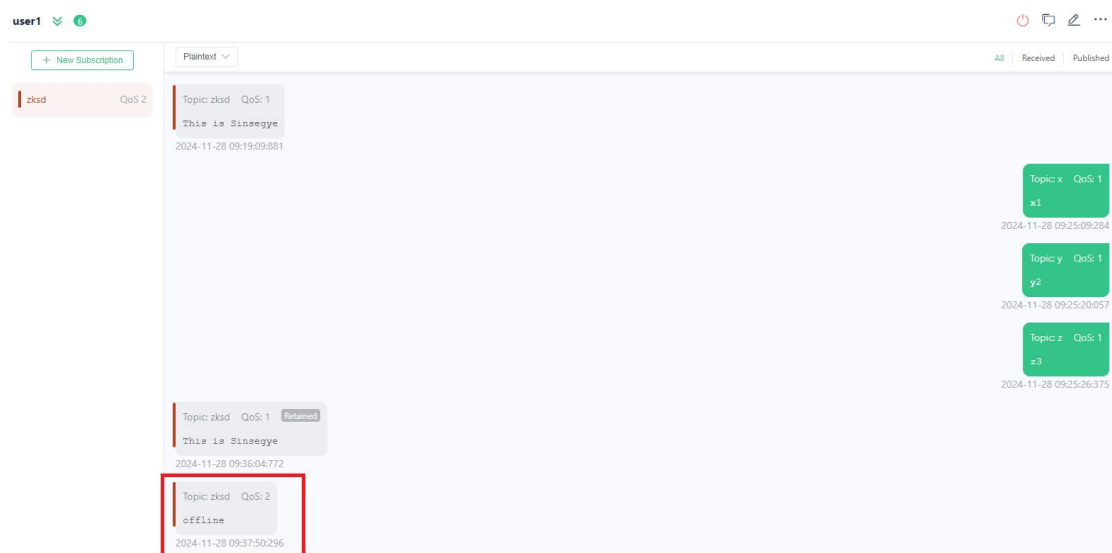
1 IF bUnsubscribeFALSE THEN //bUnsubscribeTRUE, 取消;
2
3   isUnsub01TRUE:=Client.Unsubscribe('x'); //取消订阅主题 x
4   isUnsub02TRUE:=Client.Unsubscribe('y'); //取消订阅主题 y
5   isUnsub03TRUE:=Client.Unsubscribe('z'); //取消订阅主题 z
6
7   bUnsubscribeFALSE:=FALSE; //完成取消订阅, 将bUnsubscr
8 END_IF
9
10 RETURN
  
```

3.6 MQTTX 客户端断开重连后，可以收到标记为 Retained 的保留消息。



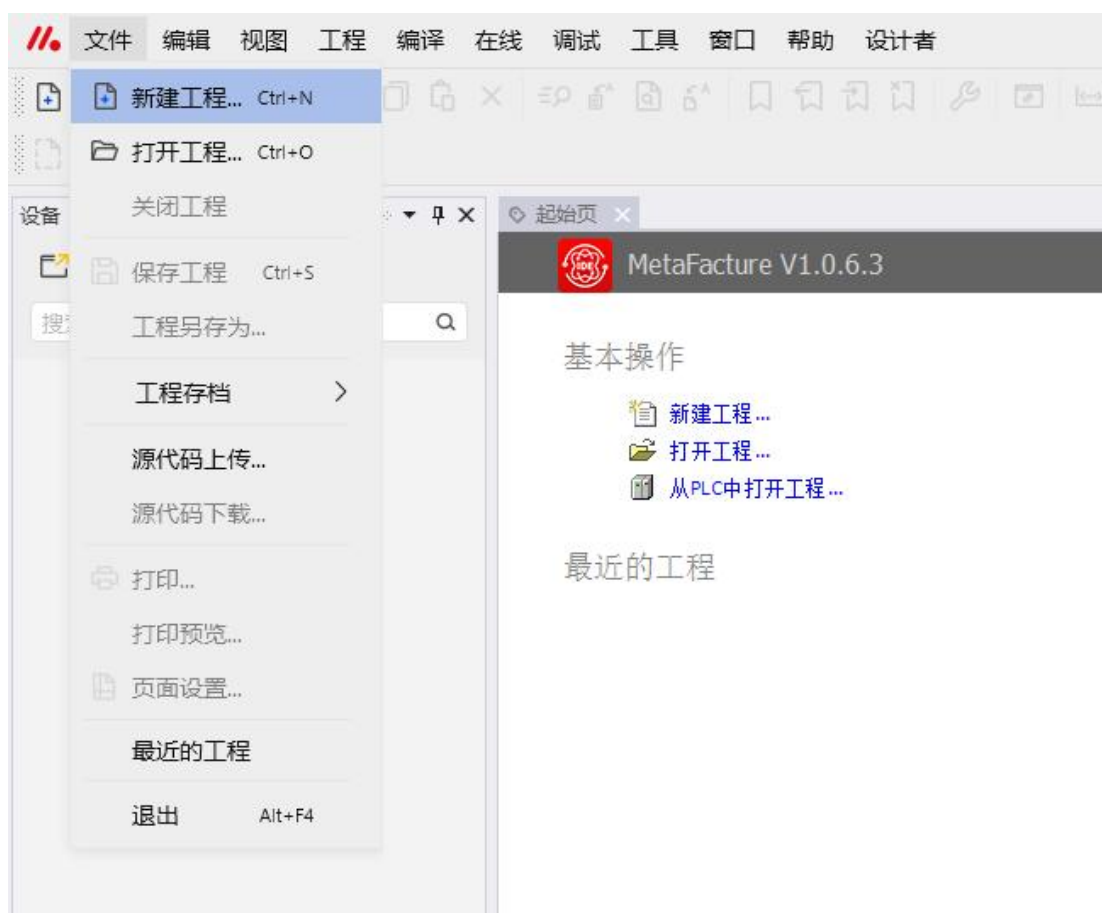
3.7 工智机手动拔除网线，约 5 秒后，MQTTX 客户端收到一条“offline”的遗嘱消息。



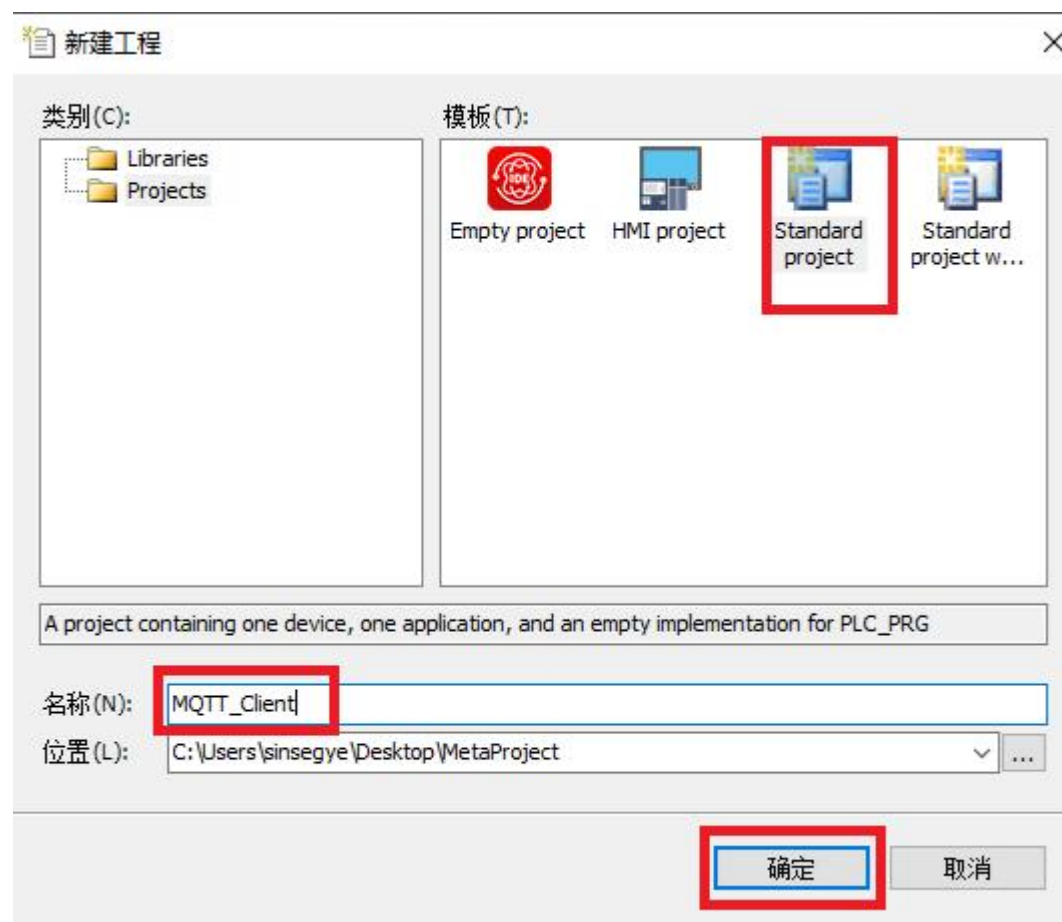
二、示例

创建工程

打开中科时代 MetaFactory V1.0.6.3,选择“文件”——“新建工程”



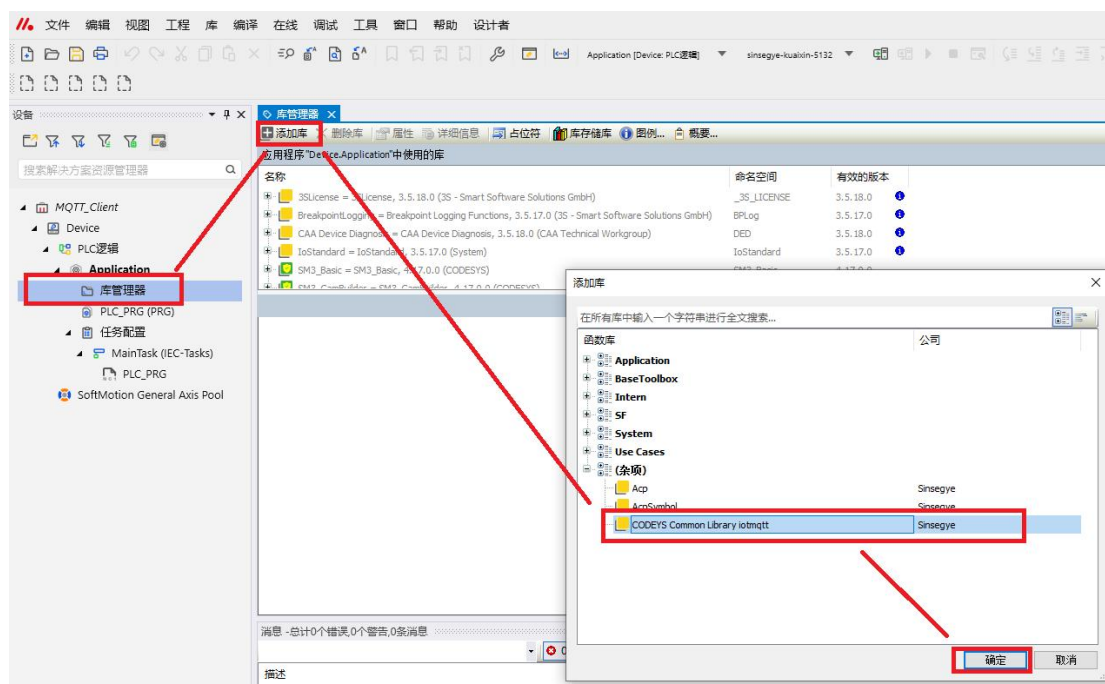
选择标准工程——命名为 MQTT_Client——点击“确定”



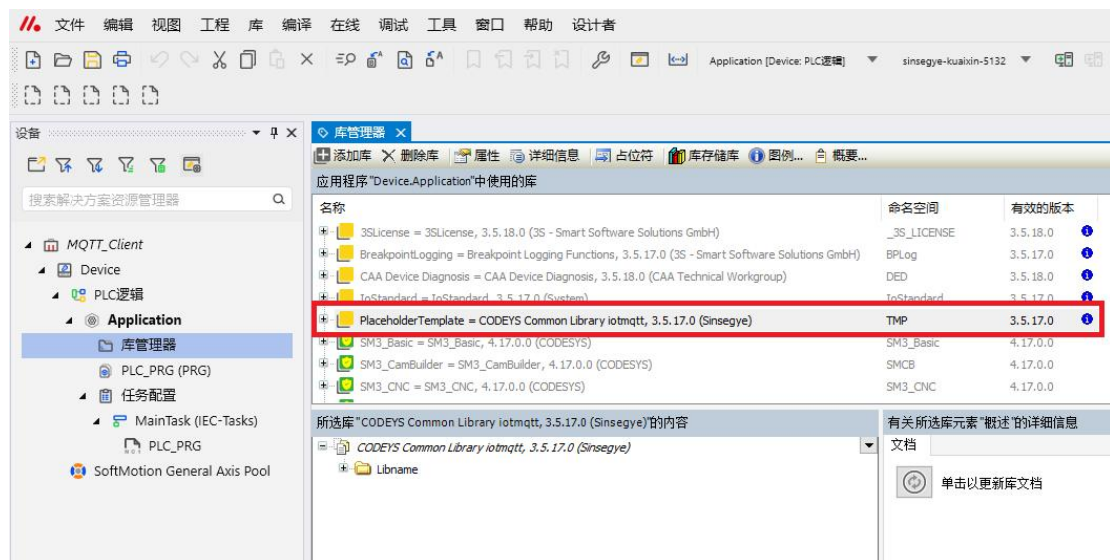
选择 Sinsegye 工智机以及编程语言（本例为 ST 语言）



选择“库管理器”——“添加库”——在杂项中选择“iotmqtt”——点击“确定”。

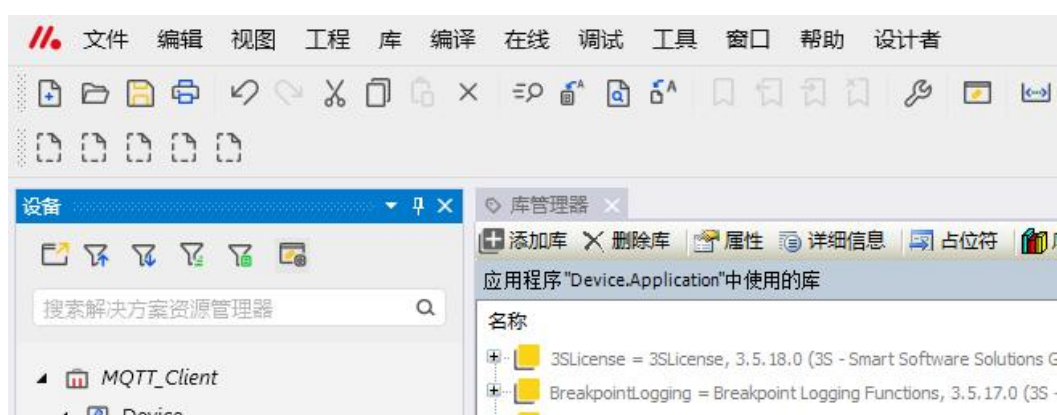


PLC 工程中完成库文件添加。

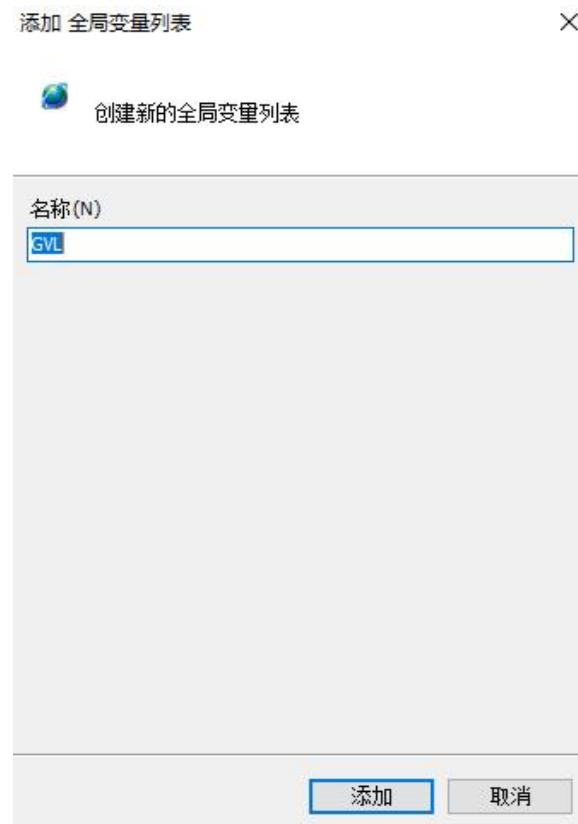


建立 MQTT 客户端

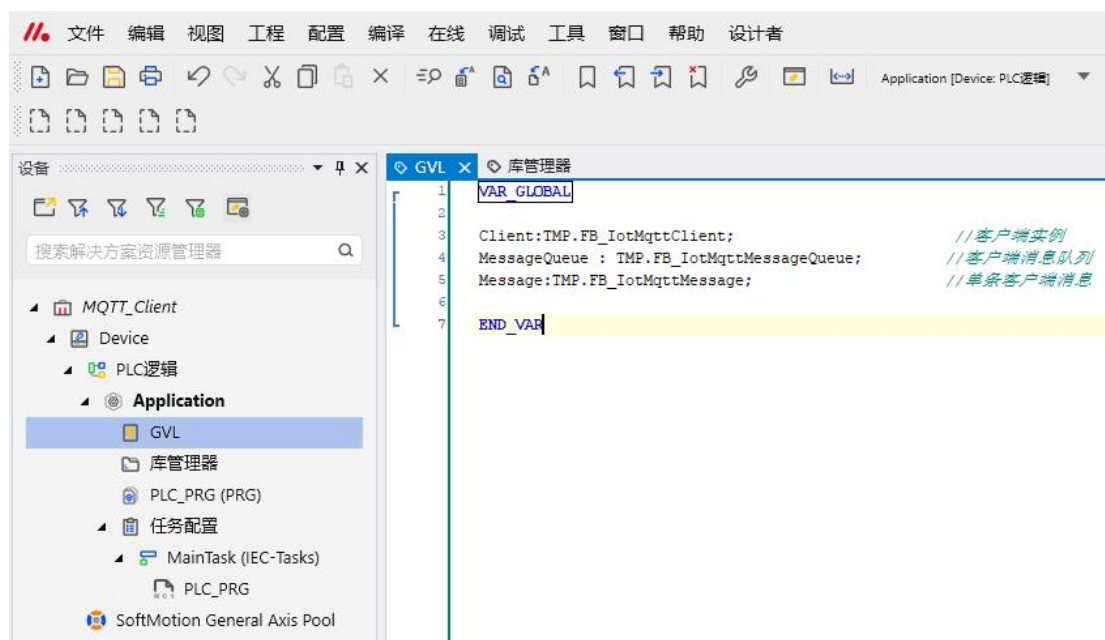
鼠标右键“Application”——“添加对象”——“全局变量列表”



命名全局变量列表名称

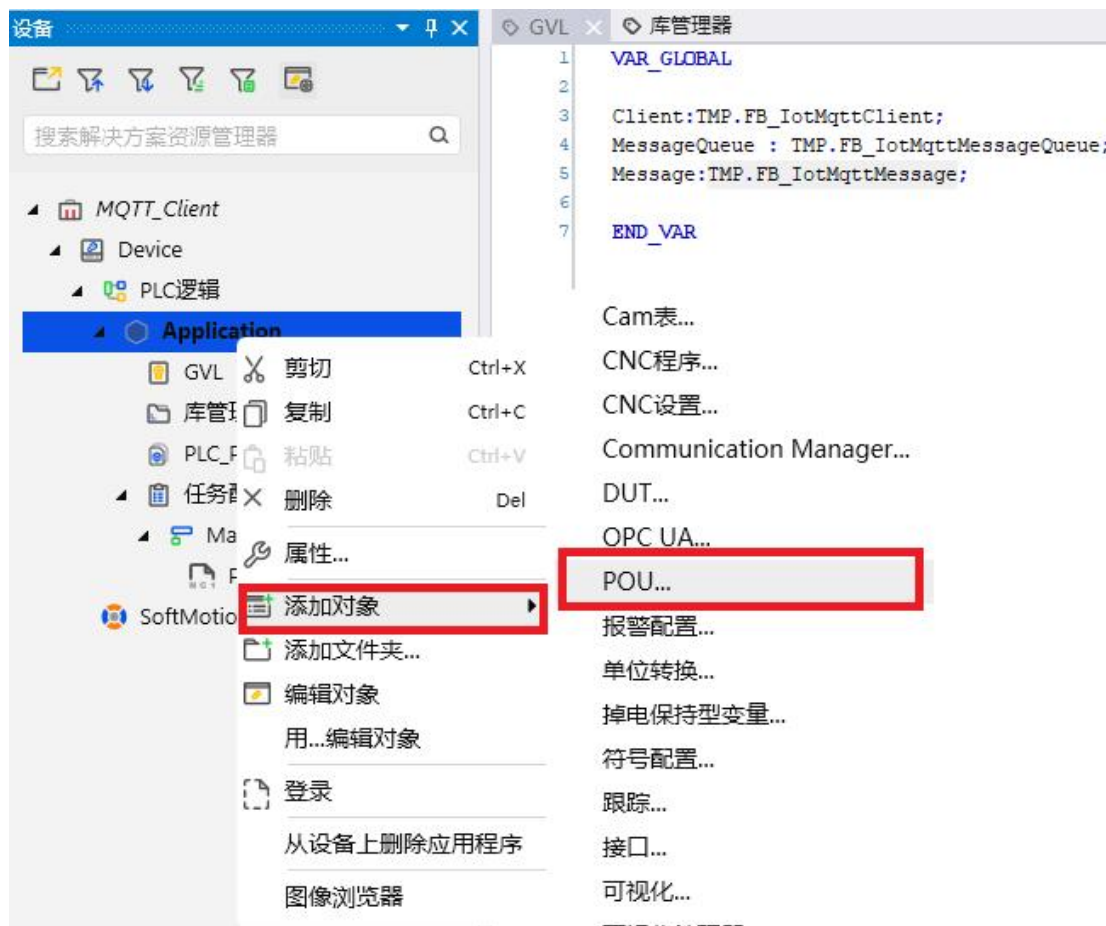


添加如下变量：



实例名称	功能块名称	说明
Client	TMP.FB_IotMqttClient	实例化 MQTT 客户端
MessageQueue	TMP.FB_IotMqttMessageQueue	存放订阅的消息队列
Message	TMP.FB_IotMqttMessage	取出的单条订阅消息

鼠标右键“Application”——“添加对象”——“POU”



将程序命名为“Connect”，选择 ST 语言后，点击“添加”

添加 POU ×

 创建新的POU(程序组织单元)

名称(N)

类型(T)

☒ 程序(P)

☐ 功能块(B)

☐ 扩展(X) ...

☐ 实现(I) ...

☐ 最终 ☐ 抽象(S)

访问说明符(A)

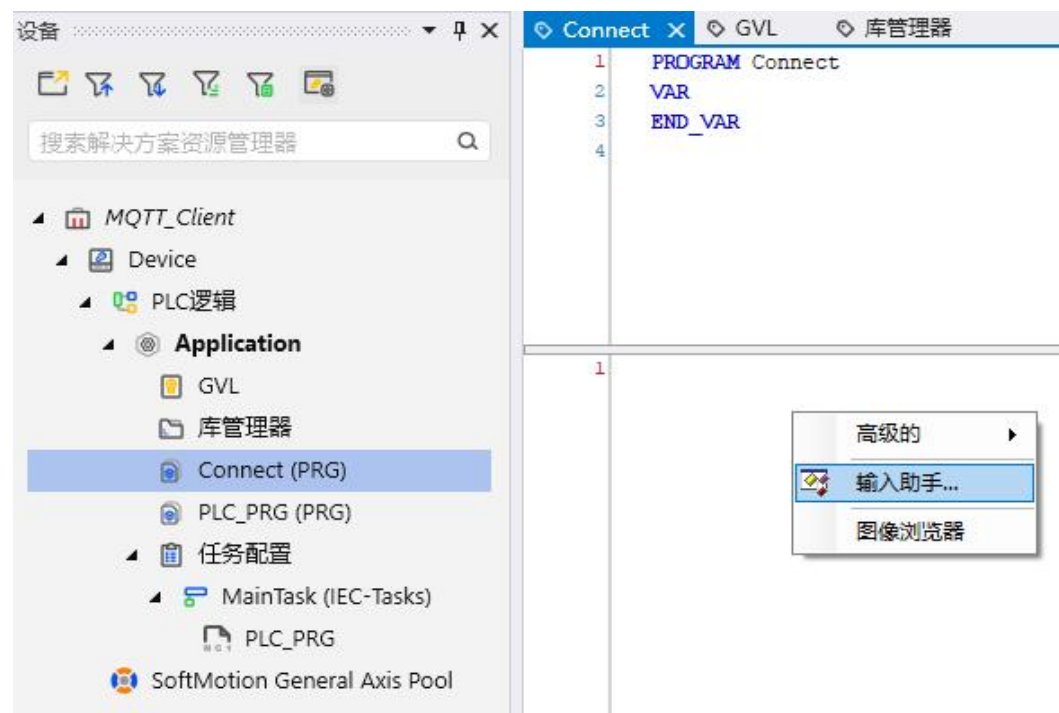
方法实现语言(M):

☐ 函数(F)

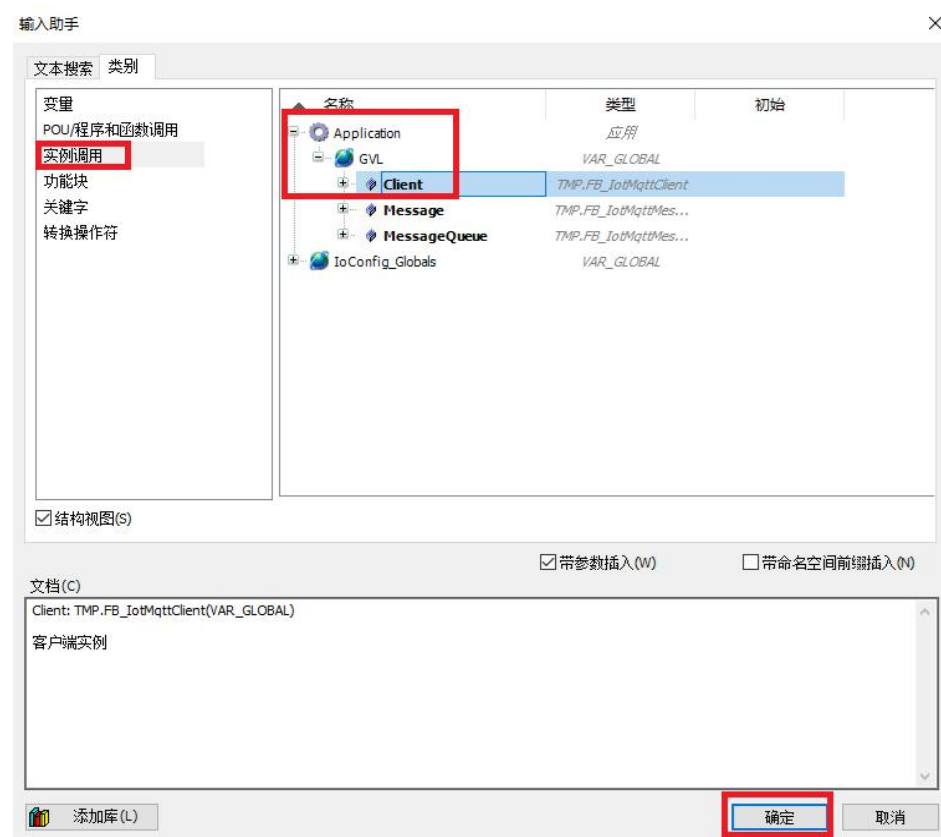
返回类型(R) ...

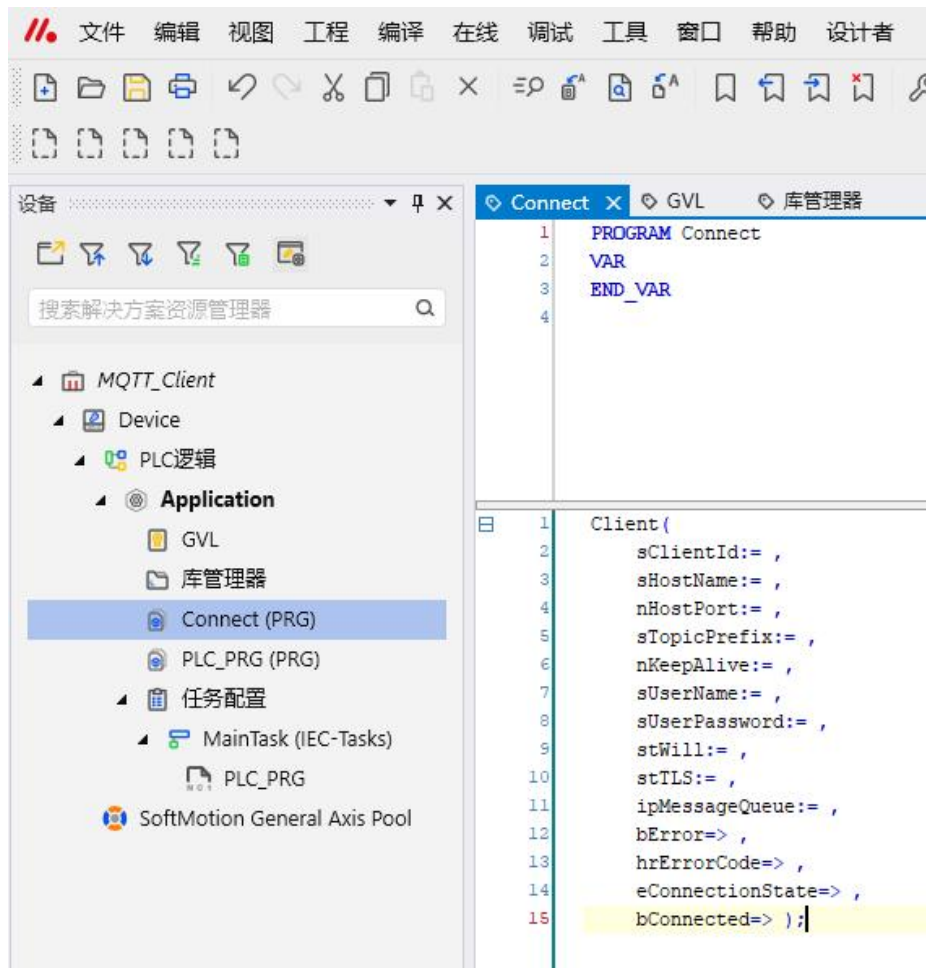
实现语言(L)

在 Connect 的 PLC 中，鼠标右键空白处，选择“输入助手”



点击“实例调用”——找到工程 Application 的 GVL 中的 Client，点击“确定”





添加参数 Client ID、Host IP、Host Port、心跳时间、消息队列指针。

关于功能块的参数说明，详见[功能介绍](#)章节。

参数	说明	设定例
sClientId	客户端 ID	'Sinsegye'
sHostName	连接 IP 地址	'192.168.110.147'
nHostPort	连接端口号	1883
nKeepAlive	心跳报文发送间隔时间，单位 s	60
ipMessageQueue	指向消息队列	ADR(MessageQueue)

声明 BOOL 型连接变量 bConnect

```
1 PROGRAM Connect
2 VAR
3     bConnect:BOOL:=FALSE; //連接 MQTT Broker 的變量
4 END_VAR
5
```

关于方法参数说明详见功能介绍章节。

The screenshot shows the Siemens SIMATIC Manager interface. On the left, the project tree is visible with 'Application' selected. The main window displays the 'Connect' dialog box, where the 'Connect' button is highlighted. The 'Connect' dialog box contains the following text:

```

1 PROGRAM Connect
2 VAR
3   bConnect:BOOL:=FALSE; //连接 MQTT Broker 的变量
4 END_VAR

```

The 'Client' tab is also visible, showing the 'Client' object configuration. The 'Client' tab contains the following text:

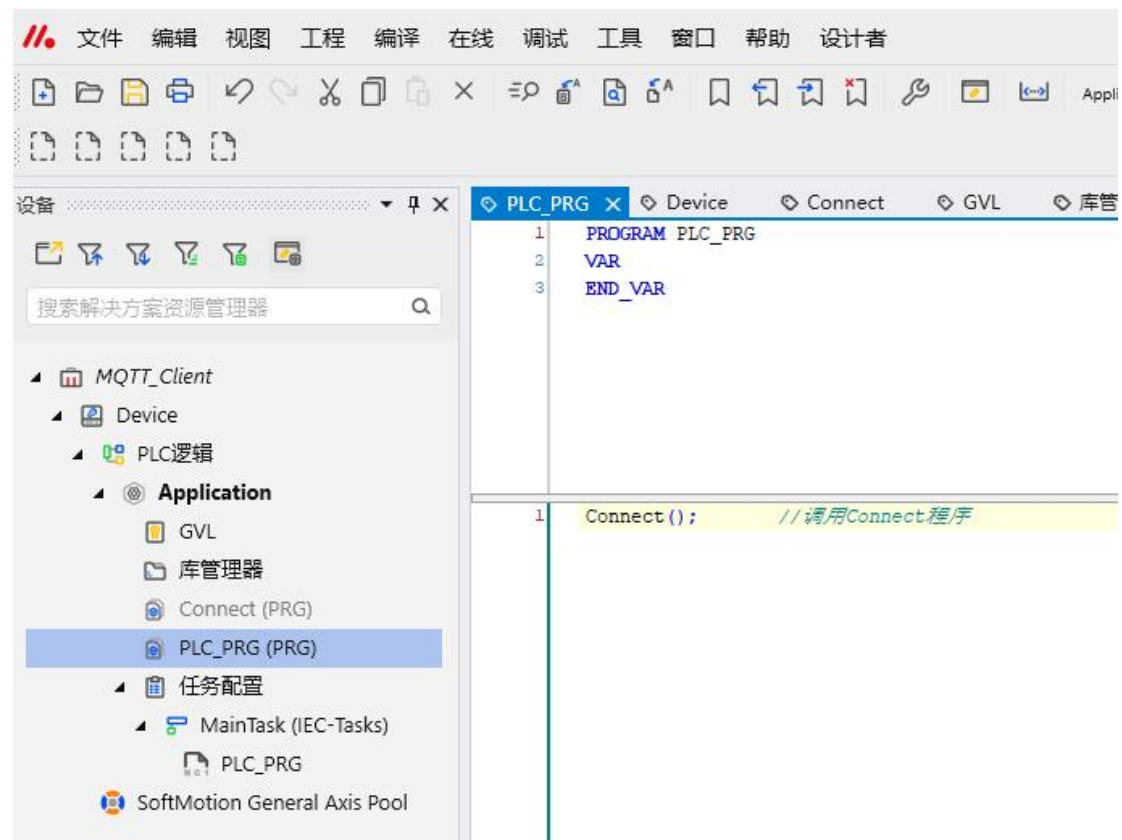
```

1 Client{
2   sClientId:='Sinsageye' //mqtt的客户端ID,必须唯一,不填有程序自动分配
3   sHostName:='192.168.110.147' //mqtt的服务器的IP地址
4   nHostPort:=1883 //mqtt服务器的端口
5   sTopicPrefix:='' //消息通道的名称,订阅或者发布消息的时候需要添加这个前缀,可不填
6   nKeepAlive:=60 //心跳报文发送时间,默认为60秒
7   sUserName:='' //连接mqtt服务器需要的用户名
8   sUserPassword:='' //连接mqtt服务器需要的密码
9   sWill:='' //mqtt连接时,如果mqtt客户端发生异常断开,服务器会在指定的topic发送消息
10  sTls:='' //mqtt连接的时候,需要设置一些证书和密钥文件的路径,必须保证路径存在
11  ipMessageQueue:=ADR(MessageQueue) //消息队列的指针,订阅消息时需要指向消息队列
12  bError:= //是否有错误发生
13  hzErrorCode=> //错误码
14  sConnectionState=> //mqtt的连接状态,枚举类型
15  bConnected:= //是否建立了连接

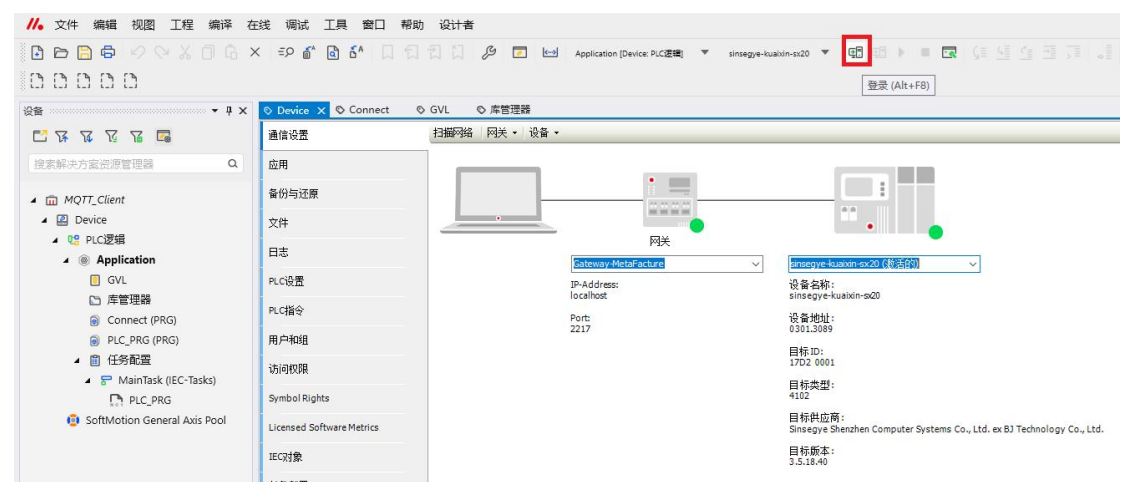
```

The 'Client.Execute(bConnect); //Execute方法: 传入TRUE时, 连接MQTT Broker' line is highlighted in yellow.

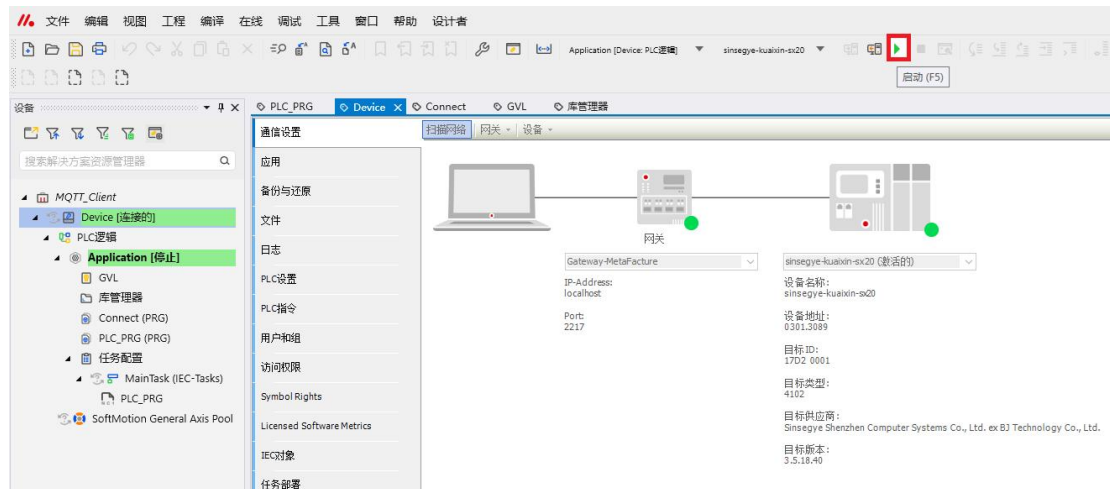
点击 PLC_PRG 程序，调用 Connect 程序。



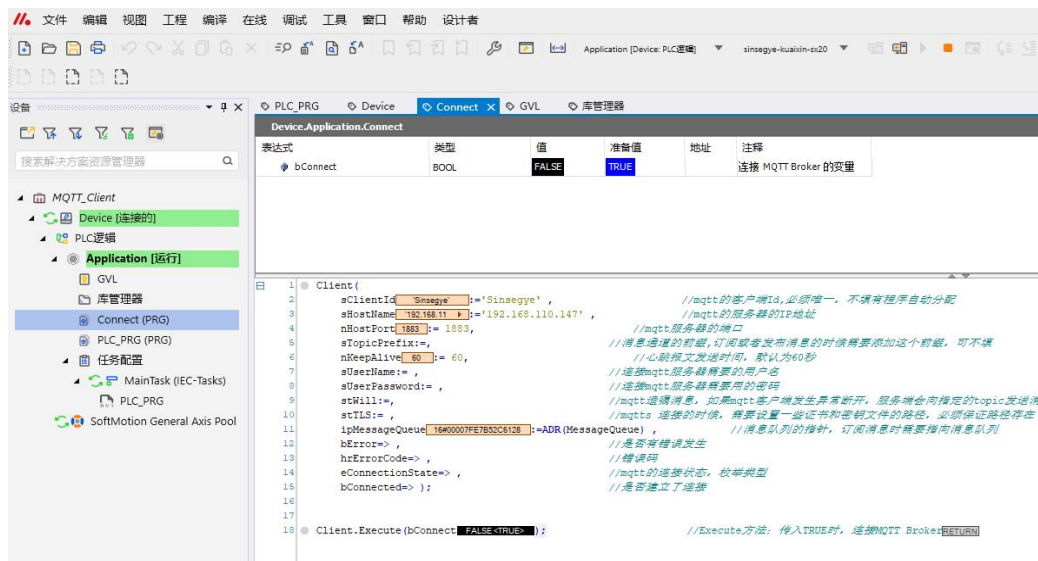
登录工智机，将 PLC 下载至工智机中。



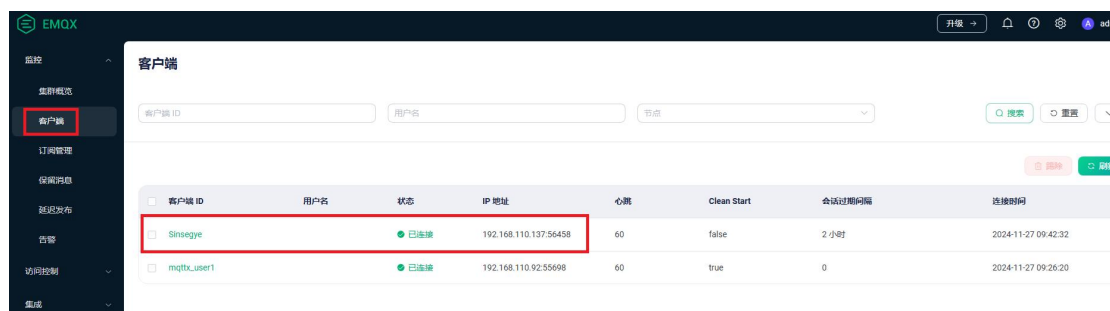
启动 PLC



将 bConnect 变量置为 TRUE, Ctrl+F7 写入变量，客户端与 MQTT Broker 进行连接。

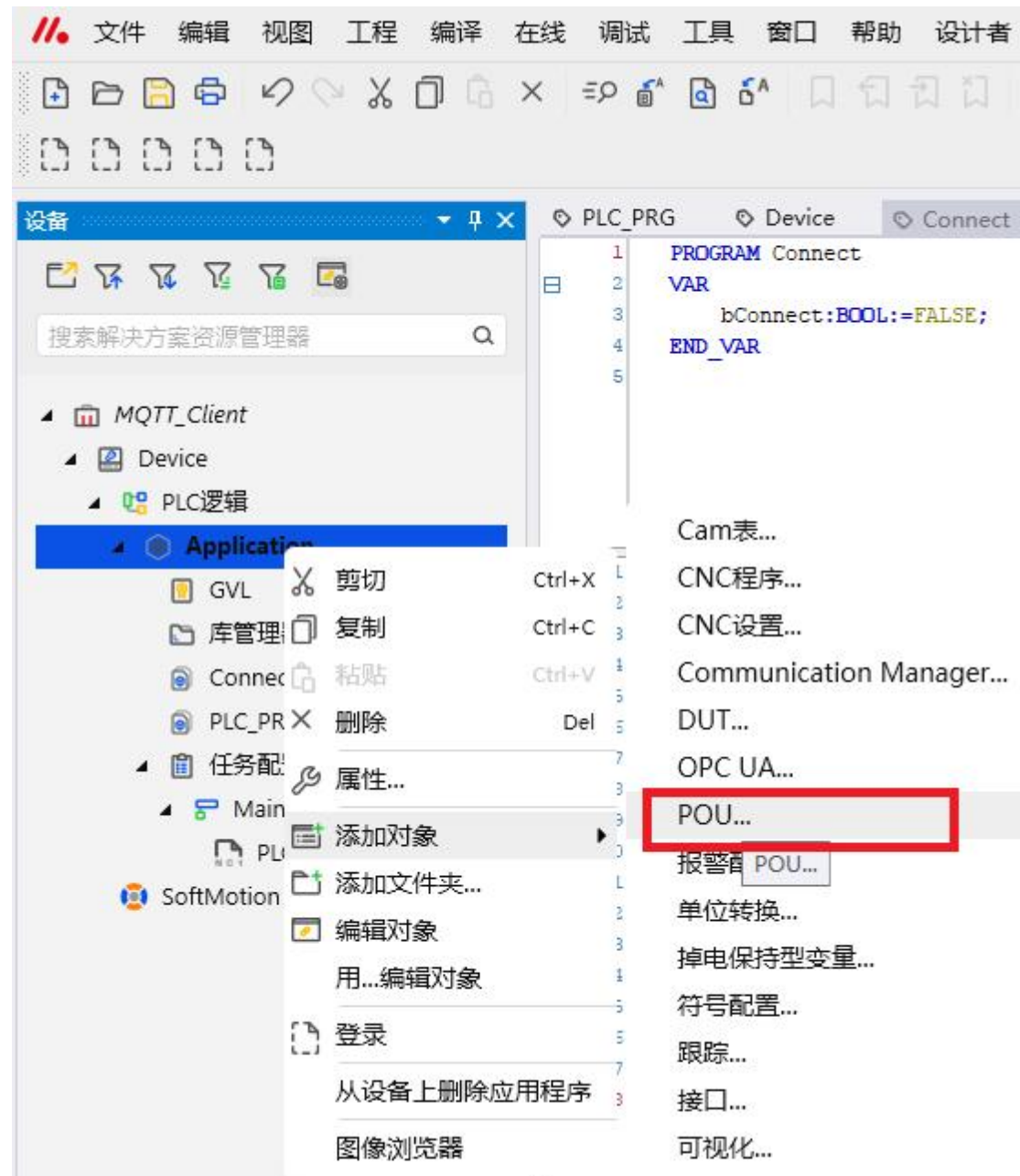


查看 Broker 的 Dashboard 的客户端界面，可以看到 mqttx_user1 是个人电脑使用 MQTTX 创建的客户端，而 Sinsegge 是工智机创建的客户端。

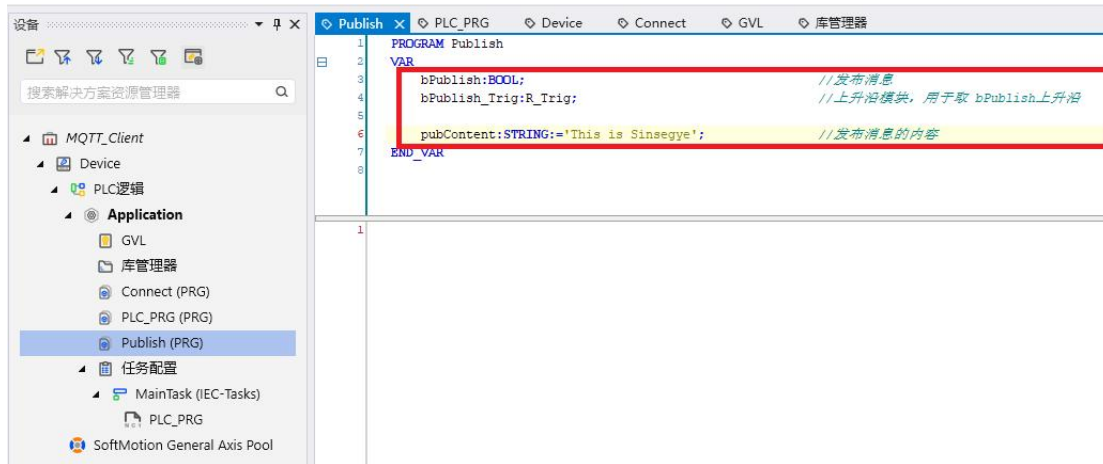


发布消息

新建一个 POU 程序，命名为 Publish。



声明变量

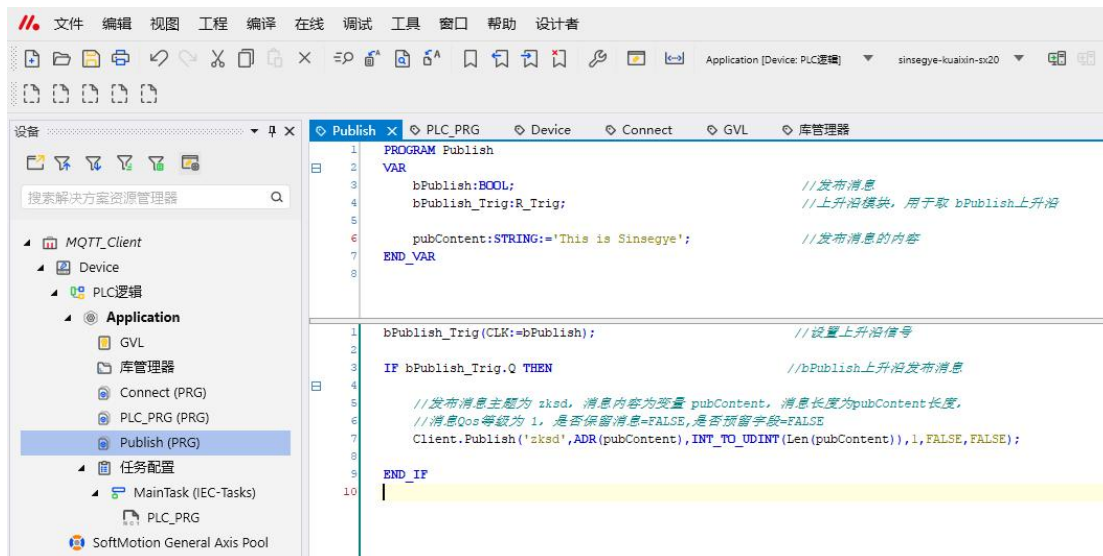


添加 PLC 程序，其中 bPublish_Trig 功能块作用是取 bPubish 变量的上升沿。

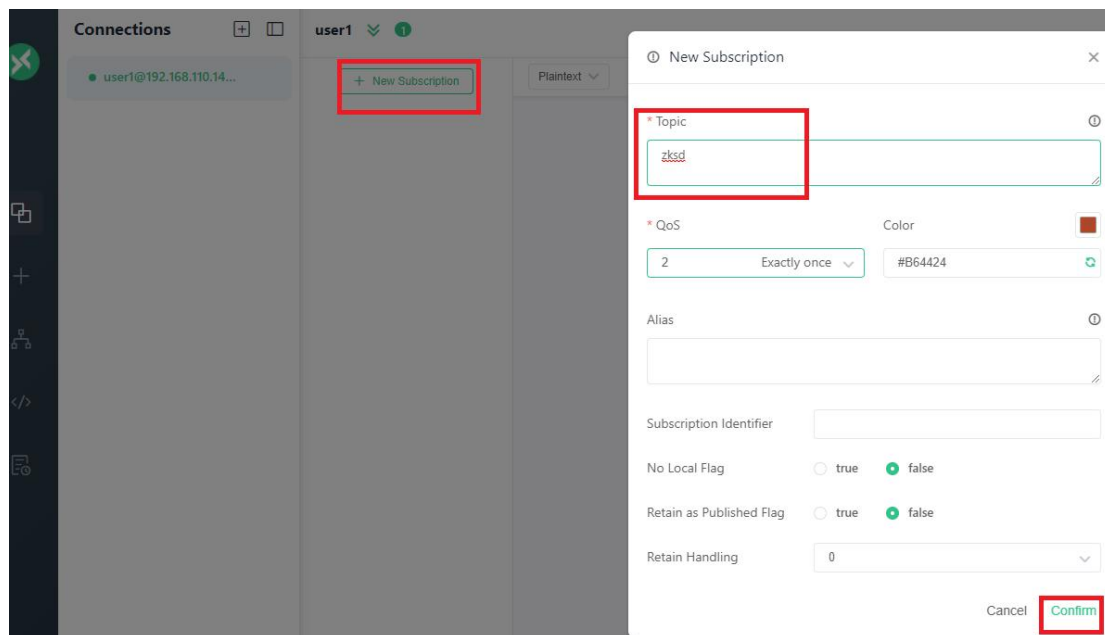
使用 Client.Publish()方法发布消息，

发布主题为 zksd，内容为 This is Sinsegye 的消息。

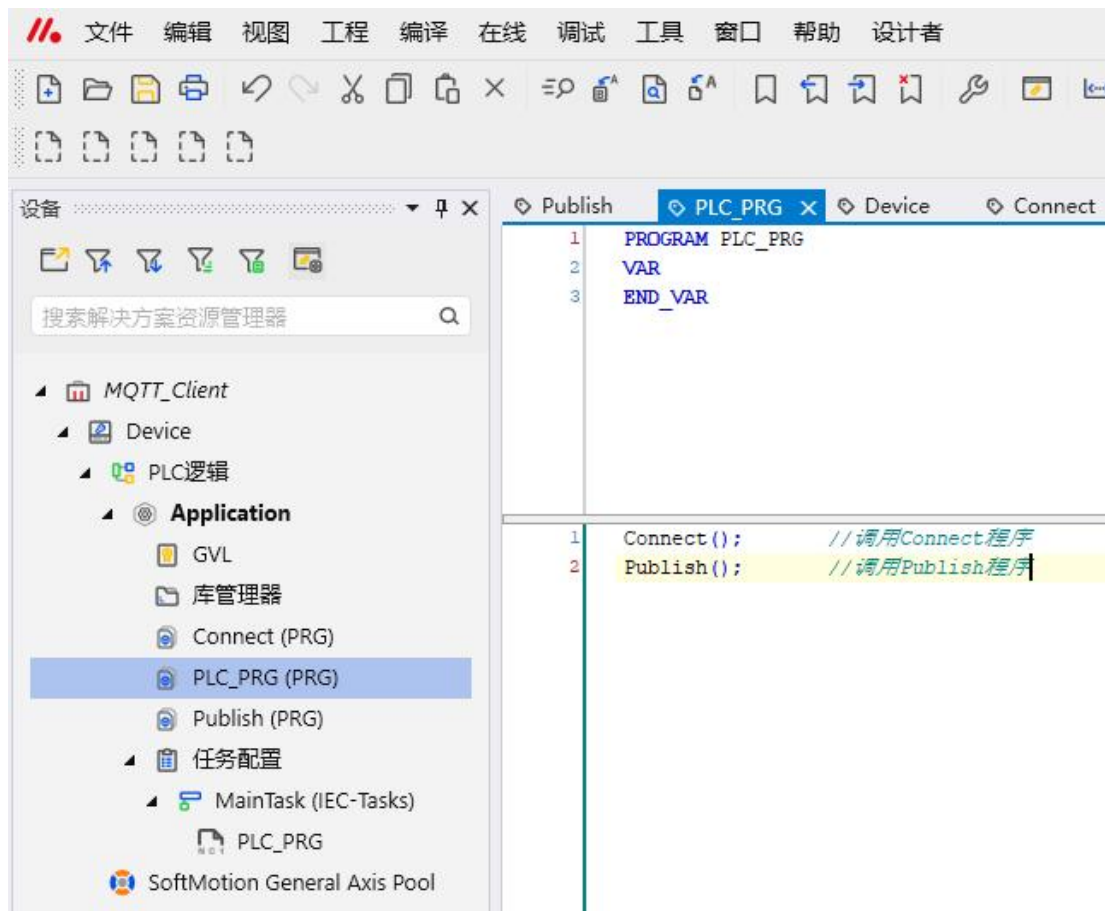
关于方法参数说明详见功能介绍章节



在个人电脑侧 MQTTX 客户端，订阅 zksd 主题

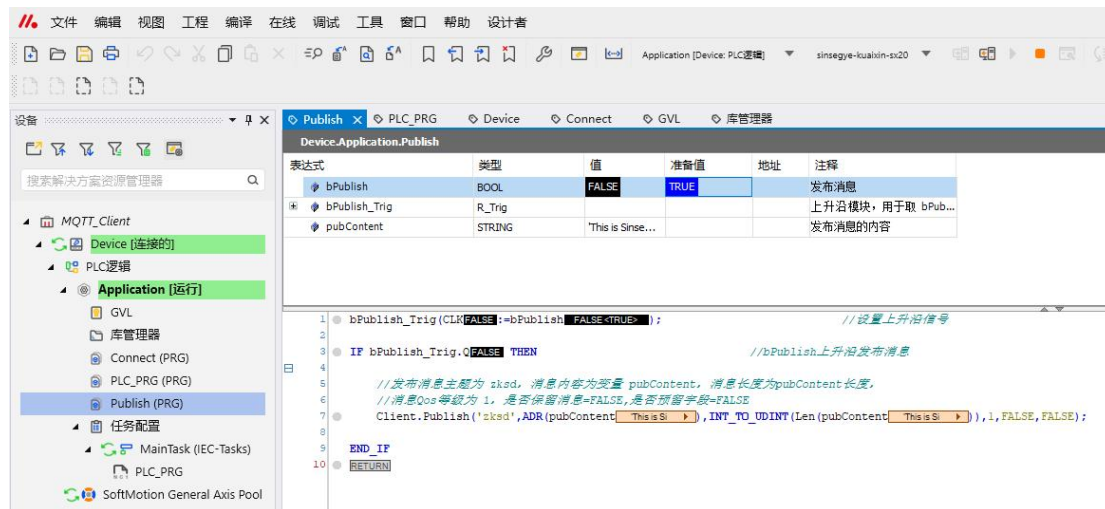


点击 PLC_PRG 程序，调用 Publish 程序。

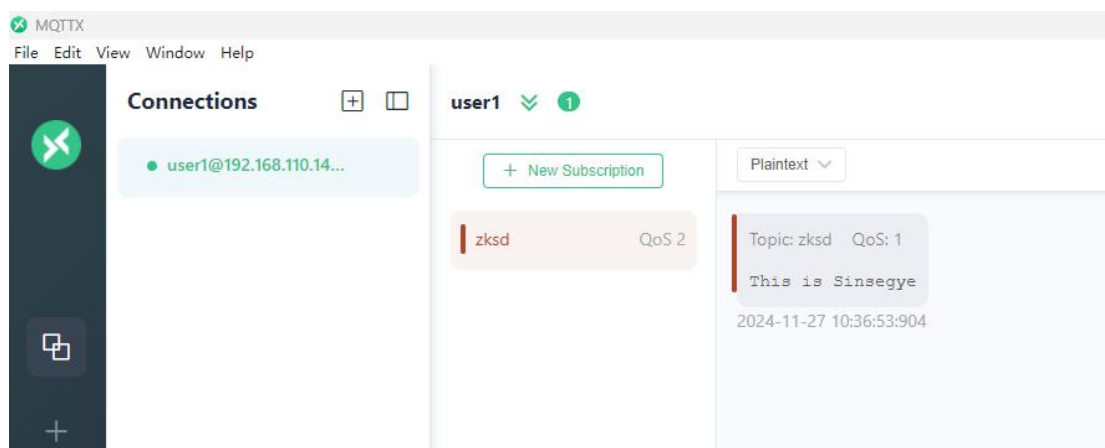


登录工智机，将 PLC 下载至工智机中，启动 PLC。

- 1.先将 bConnect 变量置为 TRUE,Ctrl+F7 写入变量,客户端与 MQTT Broker 进行连接。
- 2.再将 bPublish 变量置为 TRUE,Ctrl+F7 写入变量，发布消息。



MQTTX 客户端收到主题为 zksd 的消息



订阅消息

新建一个 POU 程序，命名为 Subscribe。

添加 POU ×

 创建新的POU(程序组织单元)

名称(N)

类型(T)

- ☒ 程序(P)
- ☐ 功能块(B)
 - ☐ 扩展(X) ...
 - ☐ 实现(I) ...
 - ☐ 最终 ☐ 抽象(S)
- ☐ 函数(F)
 - 返回类型(R) ...

访问说明符(A)

方法实现语言(M):

结构化文本(ST)

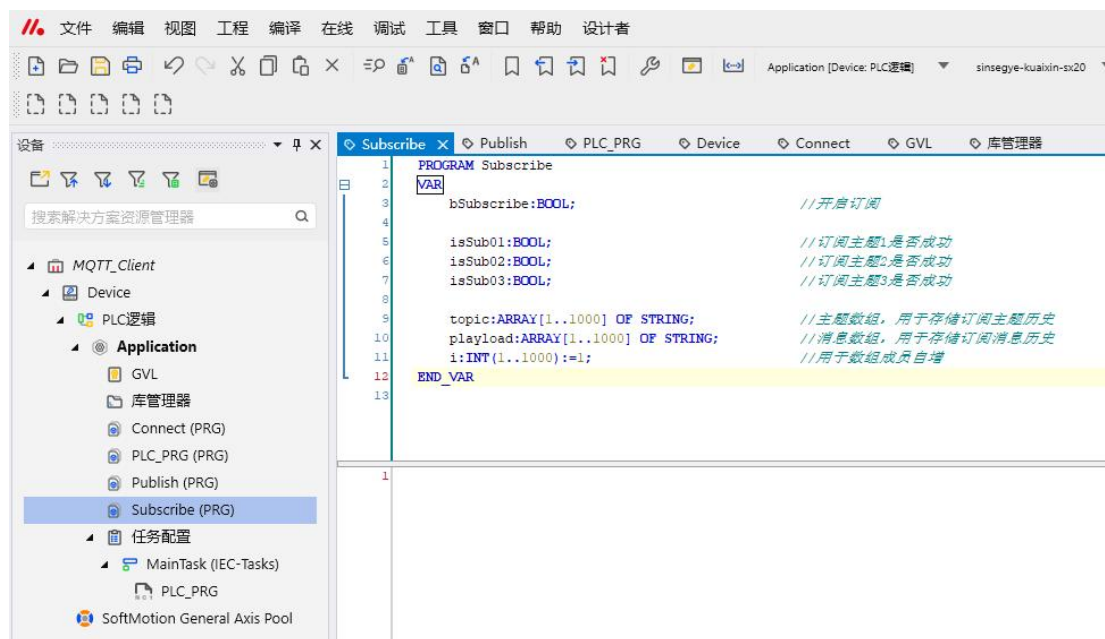
实现语言(L)

结构化文本(ST)

添加

取消

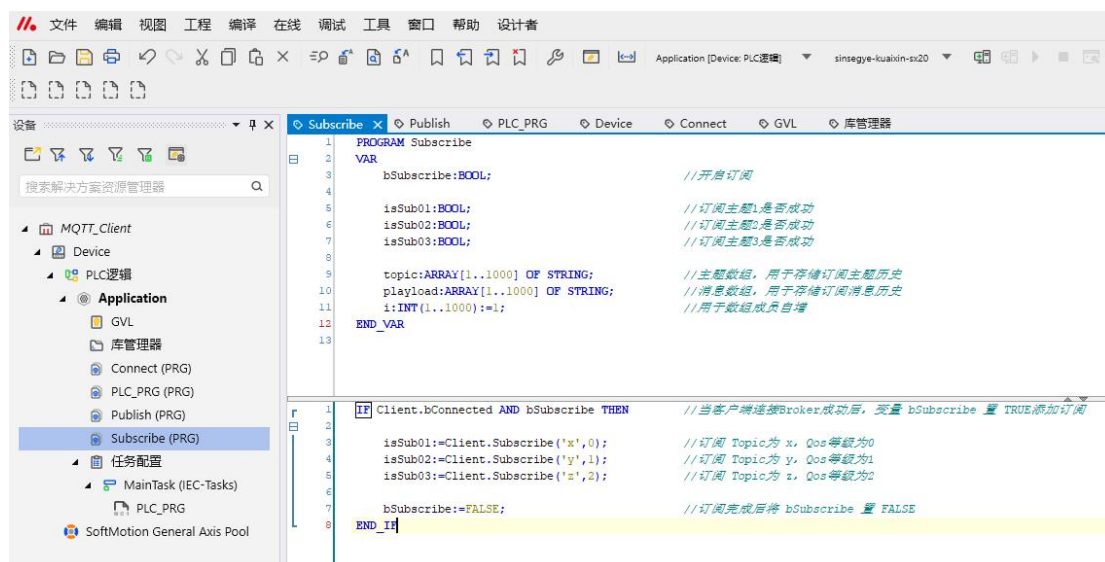
声明变量



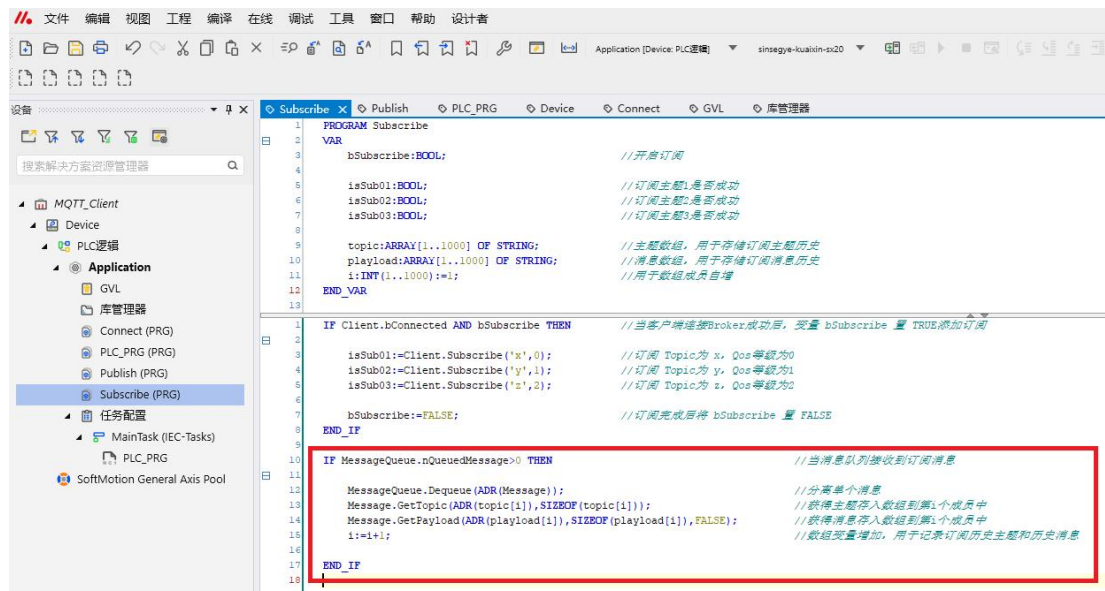
添加 PLC 程序。使用 Client.Subscribe()方法订阅消息。

本例订阅了三个主题，分别为‘x’、‘y’和‘z’

关于方法参数说明详见功能介绍章节。



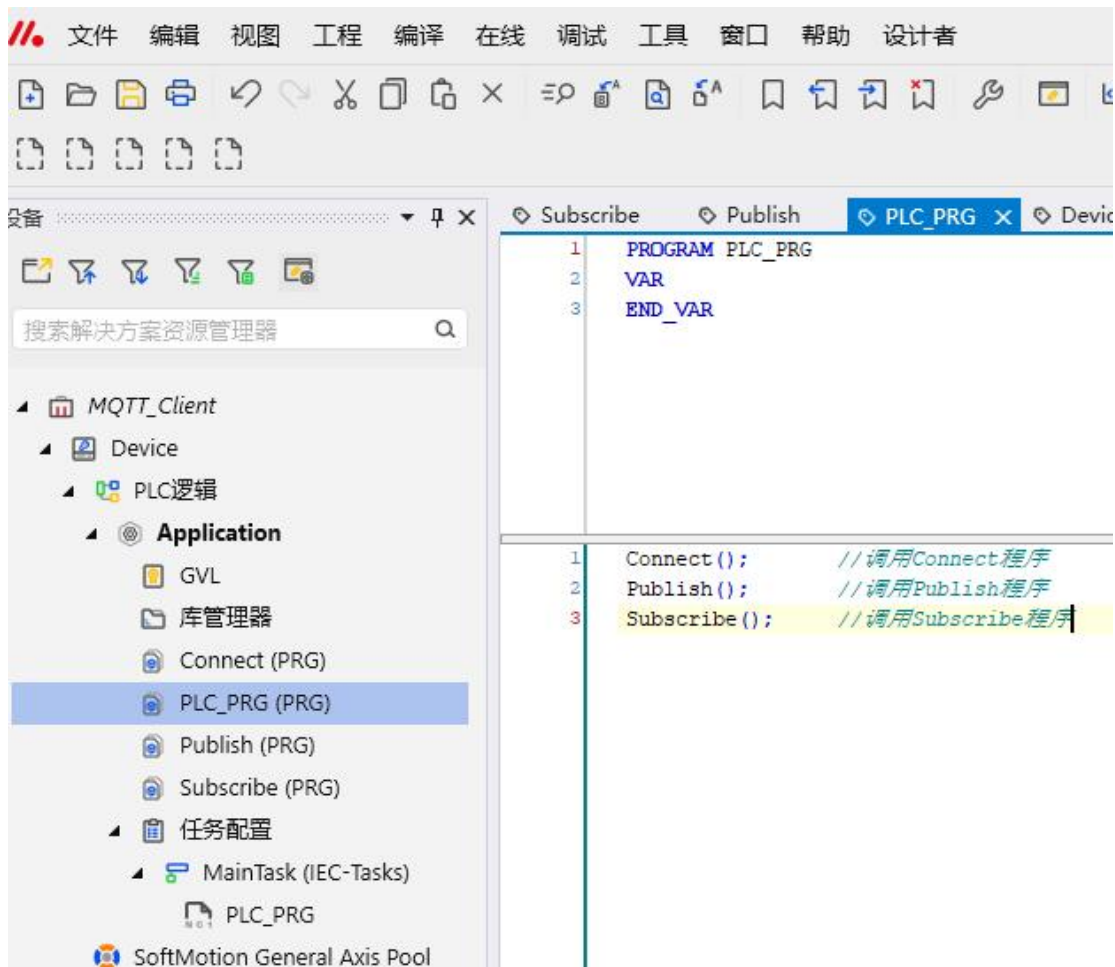
当消息发送过来后，存储再消息队列中，需要取出消息，添加消息取出的 PLC 代码：



```

1 PROGRAM Subscribe
2 VAR
3   bSubscribe:BOOL;           // 开启订阅
4
5   isSub01:BOOL;              // 订阅主题1是否成功
6   isSub02:BOOL;              // 订阅主题2是否成功
7   isSub03:BOOL;              // 订阅主题3是否成功
8
9   topic:ARRAY[1..1000] OF STRING; // 主题数组，用于存储订阅主题历史
10  payload:ARRAY[1..1000] OF STRING; // 消息数组，用于存储订阅消息历史
11  i:INT(1..1000):=1;          // 用于数组成员自增
12 END_VAR
13
14 IF Client.bConnected AND bSubscribe THEN // 当客户端连接Broker成功后，变量 bSubscribe 置 TRUE 添加订阅
15   isSub01:=Client.Subscribe('x',0);      // 订阅 Topic 为 x, Qos 等级为0
16   isSub02:=Client.Subscribe('y',1);      // 订阅 Topic 为 y, Qos 等级为1
17   isSub03:=Client.Subscribe('z',2);      // 订阅 Topic 为 z, Qos 等级为2
18   bSubscribe:=FALSE;                     // 订阅完成后将 bSubscribe 置 FALSE
19 END_IF
20
21 IF MessageQueue.nQueuedMessage>0 THEN // 当消息队列接收到订阅消息
22   MessageQueue.Dequeue(ADR(Message)); // 分离单个消息
23   Message.GetTopic(ADR(topic[i]),SIZEOF(topic[i])); // 获得主题存入数组到第i个成员中
24   Message.GetPayload(ADR(payload[i]),SIZEOF(payload[i]),FALSE); // 获得消息存入数组到第i个成员中
25   i:=i+1;                               // 数组变量增加，用于记录订阅主题和历史消息
26 END_IF
  
```

在 PLC_PRG 中调用 Subscribe 程序



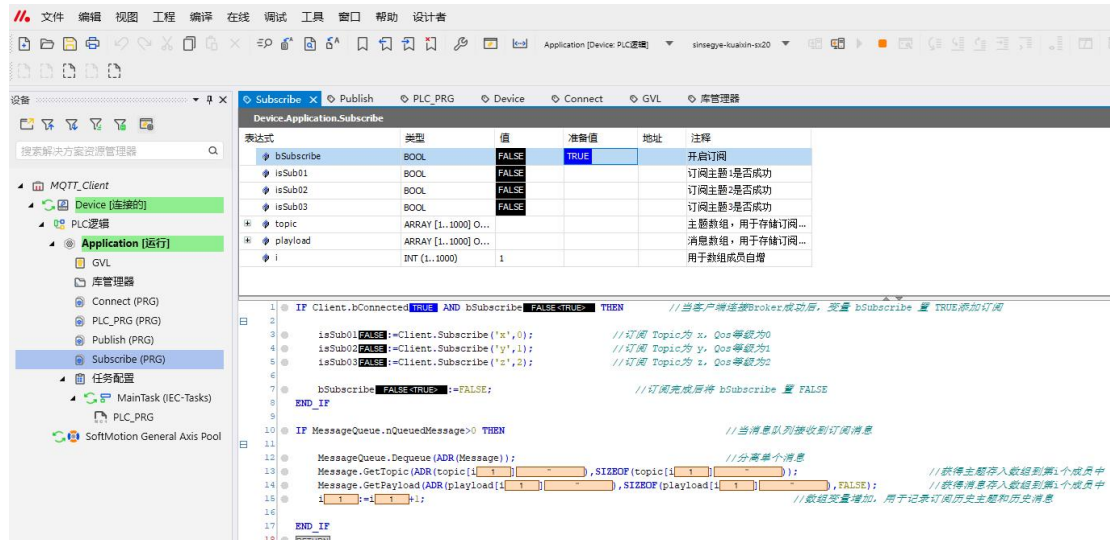
```

1 PROGRAM PLC_PRG
2 VAR
3 END_VAR
4
5 Connect(); // 调用Connect程序
6 Publish(); // 调用Publish程序
7 Subscribe(); // 调用Subscribe程序
  
```

登录工智机，将 PLC 下载至工智机中，启动 PLC。

1.先将 bConnect 变量置为 TRUE, Ctrl+F7 写入变量, 客户端与 MQTT Broker 进行连接。

2.再将 bSubscribe 变量置为 TRUE, Ctrl+F7 写入变量, 订阅三个主题的消息。

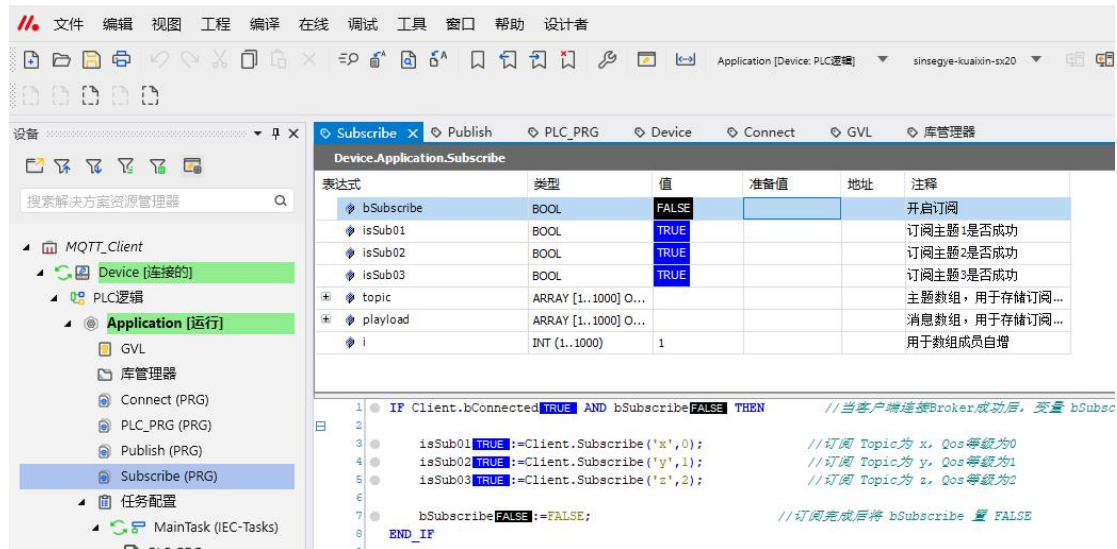


表达式	类型	值	准备值	地址	注释
bSubscribe	BOOL	FALSE	TRUE		开启订阅
isSub01	BOOL	FALSE			订阅主题1是否成功
isSub02	BOOL	FALSE			订阅主题2是否成功
isSub03	BOOL	FALSE			订阅主题3是否成功
topic	ARRAY [1..1000] O...				主题数组, 用于存储订阅...
payload	ARRAY [1..1000] O...				消息数组, 用于存储订阅...
i	INT (1..1000)	1			用于数组成员自增

```

1 IF Client.bConnected TRUE AND bSubscribe FALSE THEN //当客户端连接Broker成功后, 变量 bSubscribe 置 TRUE 添加订阅
2
3   isSub01 FALSE := Client.Subscribe('x', 0); //订阅 Topic 为 x, Qos 等级为 0
4   isSub02 FALSE := Client.Subscribe('y', 1); //订阅 Topic 为 y, Qos 等级为 1
5   isSub03 FALSE := Client.Subscribe('z', 2); //订阅 Topic 为 z, Qos 等级为 2
6
7   bSubscribe FALSE TRUE := TRUE; //订阅完成后将 bSubscribe 置 TRUE
8 END_IF
9
10 IF MessageQueue.nQueueMessage > 0 THEN //当消息队列接收到订阅消息
11
12   MessageQueue.Dequeue(ADR(topic[i])); //分离单个消息
13   Message.GetTopic(ADR(topic[i]), SIZEOF(topic[i]), FALSE); //获得主题存入数组到第 i 个成员中
14   Message.GetPayload(ADR(payload[i]), SIZEOF(payload[i]), FALSE); //获得消息存入数组到第 i 个成员中
15   i := i + 1; //数组变量增加, 用于记录订阅主题和历史消息
16
17 END_IF
18 RETURN
  
```

订阅成功

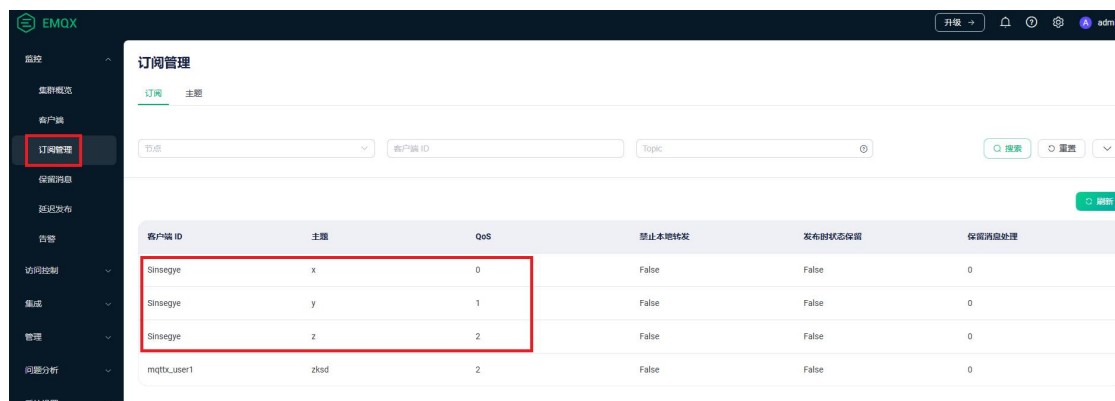


表达式	类型	值	准备值	地址	注释
bSubscribe	BOOL	FALSE			开启订阅
isSub01	BOOL	TRUE			订阅主题1是否成功
isSub02	BOOL	TRUE			订阅主题2是否成功
isSub03	BOOL	TRUE			订阅主题3是否成功
topic	ARRAY [1..1000] O...				主题数组, 用于存储订阅...
payload	ARRAY [1..1000] O...				消息数组, 用于存储订阅...
i	INT (1..1000)	1			用于数组成员自增

```

1 IF Client.bConnected TRUE AND bSubscribe FALSE THEN //当客户端连接Broker成功后, 变量 bSubsc
2
3   isSub01 TRUE := Client.Subscribe('x', 0); //订阅 Topic 为 x, Qos 等级为 0
4   isSub02 TRUE := Client.Subscribe('y', 1); //订阅 Topic 为 y, Qos 等级为 1
5   isSub03 TRUE := Client.Subscribe('z', 2); //订阅 Topic 为 z, Qos 等级为 2
6
7   bSubscribe FALSE := FALSE; //订阅完成后将 bSubscribe 置 FALSE
8 END_IF
9
10 IF MessageQueue.nQueueMessage > 0 THEN //当消息队列接收到订阅消息
11
12   MessageQueue.Dequeue(ADR(topic[i])); //分离单个消息
13   Message.GetTopic(ADR(topic[i]), SIZEOF(topic[i]), FALSE); //获得主题存入数组到第 i 个成员中
14   Message.GetPayload(ADR(payload[i]), SIZEOF(payload[i]), FALSE); //获得消息存入数组到第 i 个成员中
15   i := i + 1; //数组变量增加, 用于记录订阅主题和历史消息
16
17 END_IF
18 RETURN
  
```

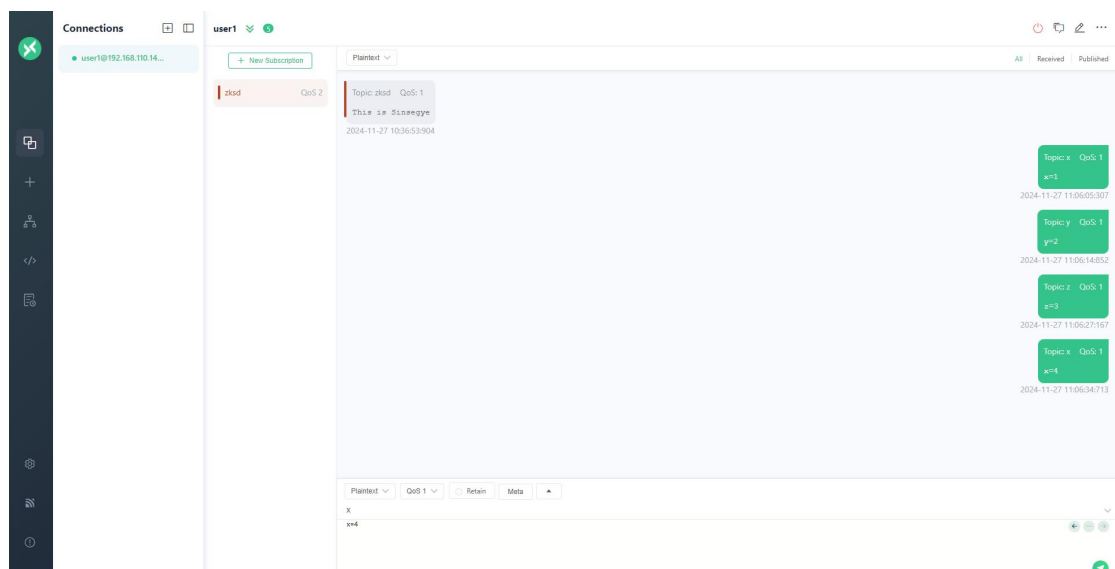

可以在 Broker 的 订阅管理中，看到订阅成功



客户端 ID	主题	QoS	禁止本地转发	发布时状态保留	保留消息处理
Sinsgeye	x	0	False	False	0
Sinsgeye	y	1	False	False	0
Sinsgeye	z	2	False	False	0
mqttx_user1	zkzd	2	False	False	0

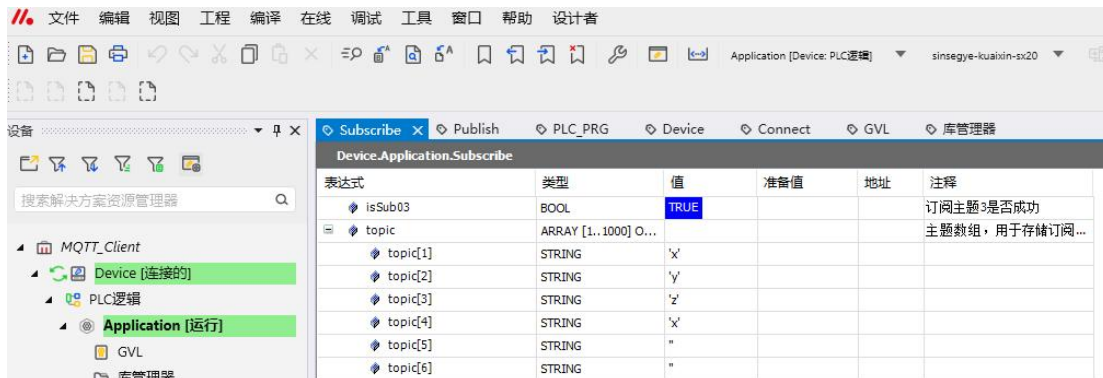
在个人电脑 MQTTX 客户端中发布消息进行测试

发布消息序列	主题	内容
1	x	x=1
2	y	y=2
3	z	z=3
4	x	x=4



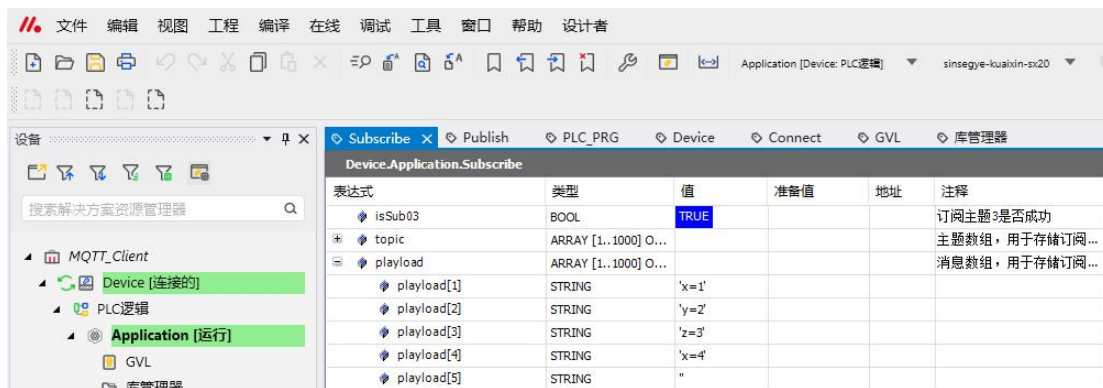
Topic	QoS	Content	Timestamp
zkzd	2	x=1	2024-11-27 11:06:05.307
zkzd	2	y=2	2024-11-27 11:06:14.852
zkzd	2	z=3	2024-11-27 11:06:27.167
zkzd	2	x=4	2024-11-27 11:06:34.713

在 PLC 程序中，变量 topic 记录了历史消息的主题



表达式	类型	值	准备值	地址	注释
isSub03	BOOL	TRUE			订阅主题3是否成功
topic	ARRAY [1..1000] O...				主题数组，用于存储订阅...
topic[1]	STRING	'x'			
topic[2]	STRING	'y'			
topic[3]	STRING	'z'			
topic[4]	STRING	'x'			
topic[5]	STRING	"			
topic[6]	STRING	"			

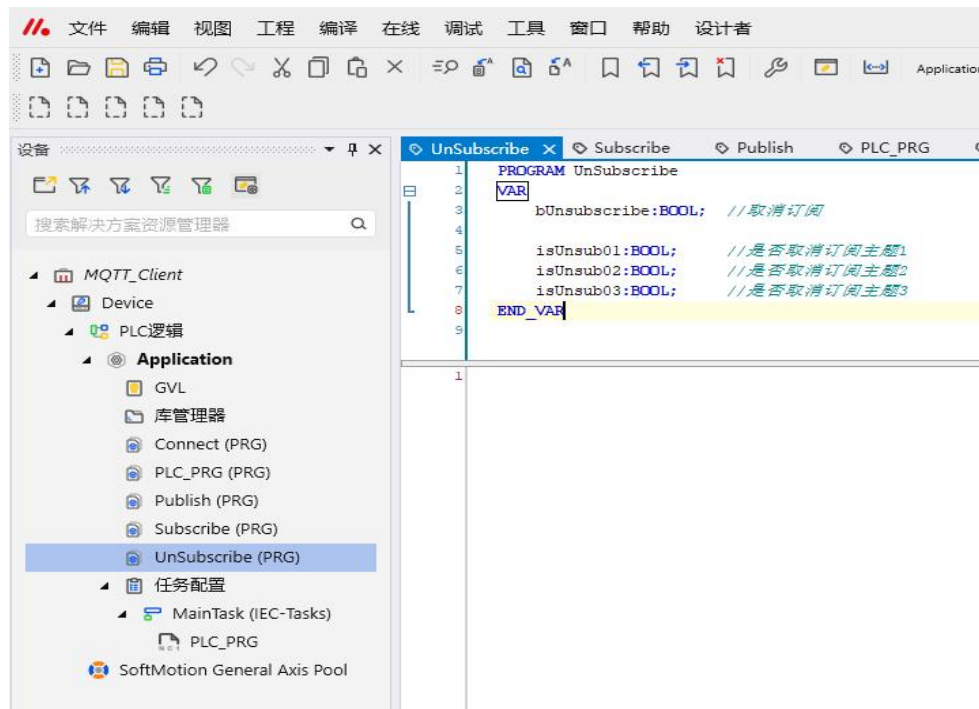
变量 payload 记录了历史消息的内容



表达式	类型	值	准备值	地址	注释
isSub03	BOOL	TRUE			订阅主题3是否成功
topic	ARRAY [1..1000] O...				主题数组，用于存储订阅...
payload	ARRAY [1..1000] O...				消息数组，用于存储订阅...
payload[1]	STRING	'x=1'			
payload[2]	STRING	'y=2'			
payload[3]	STRING	'z=3'			
payload[4]	STRING	'x=4'			
payload[5]	STRING	"			

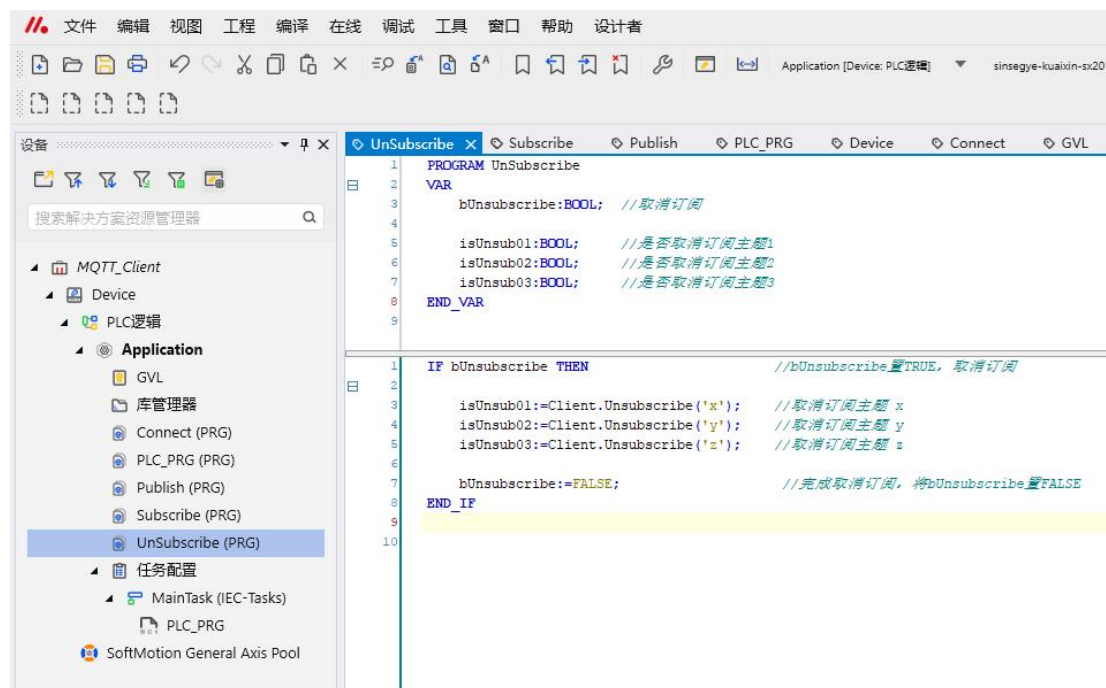
取消订阅

新建一个 POU 程序，命名为 UnSubscribe，声明如下变量。

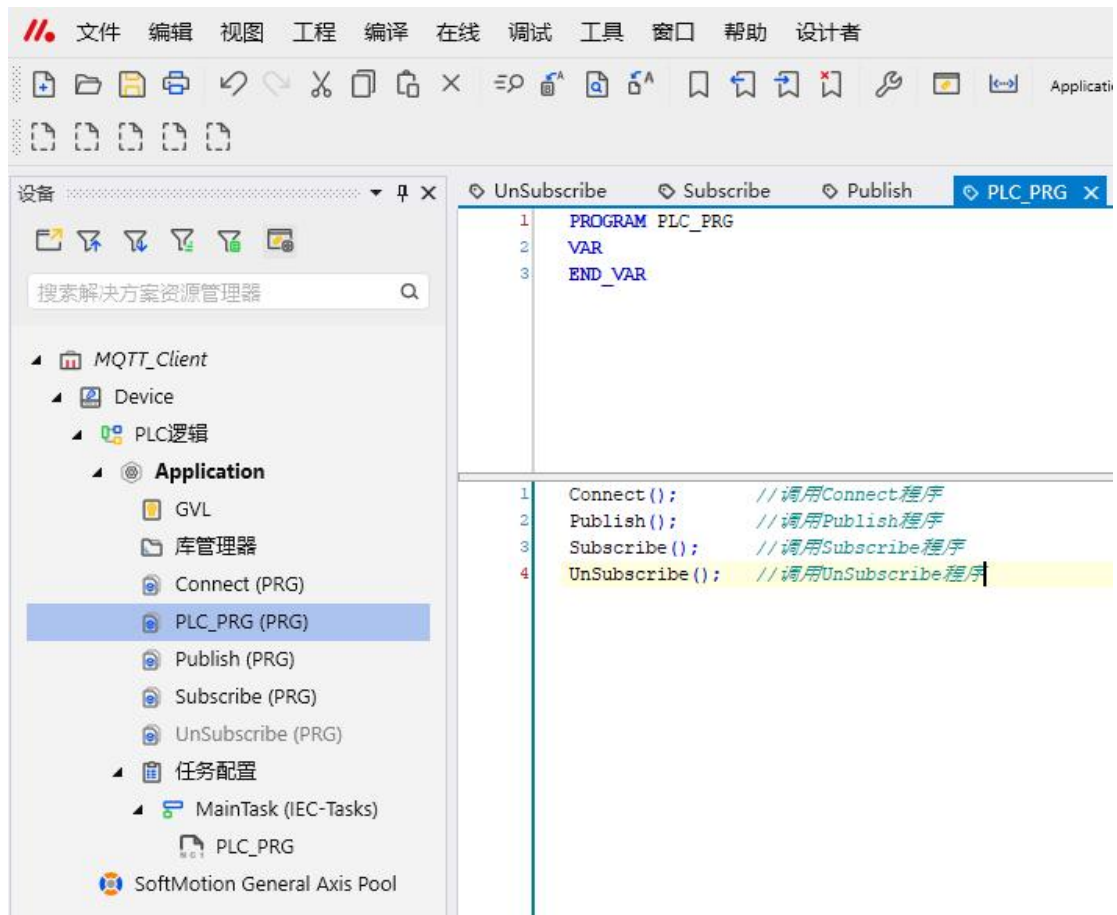


添加 PLC 程序。使用 Client.UnSubscribe()方法取消订阅。传入 String 类型的主题字符，取消订阅‘x’、‘y’和‘z’主题消息

关于方法参数说明详见功能介绍章节。



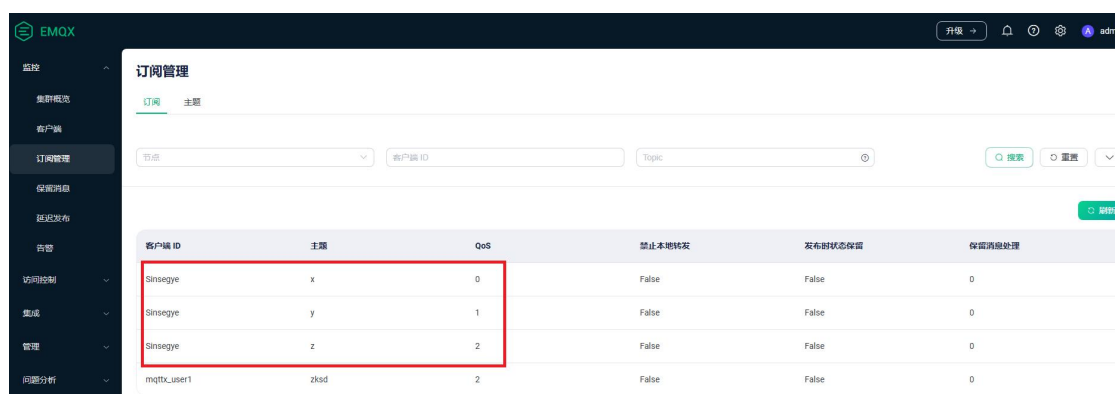
在 PLC_PRG 中调用 UnSubscribe 程序



登录工智机，将 PLC 下载至工智机中，启动 PLC。

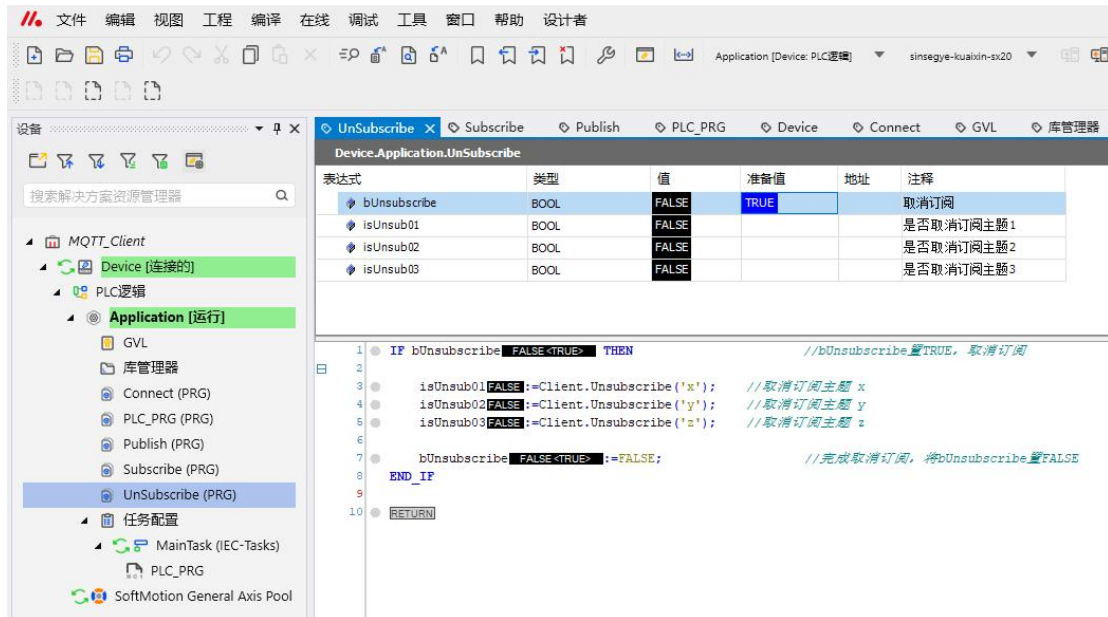
1.先将 bConnect 变量置为 TRUE, Ctrl+F7 写入变量, 客户端与 MQTT Broker 进行连接。

查看 Broker 的订阅管理，Sinsegye 订阅了三个主题



客户端 ID	主题	QoS	禁止本地转发	发布时状态保留	保留消息处理
sinsegye	x	0	False	False	0
sinsegye	y	1	False	False	0
sinsegye	z	2	False	False	0
mqttc_user1	zkzd	2	False	False	0

2.再将 bUnsubscribe 变量置为 TRUE, Ctrl+F7 写入变量，取消订阅三个主题的消息。



设备: MQTT_Client

库管理器

Connect (PRG)

PLC_PRG (PRG)

Publish (PRG)

Subscribe (PRG)

UnSubscribe (PRG)

任务配置

MainTask (IEC-Tasks)

PLC_PRG

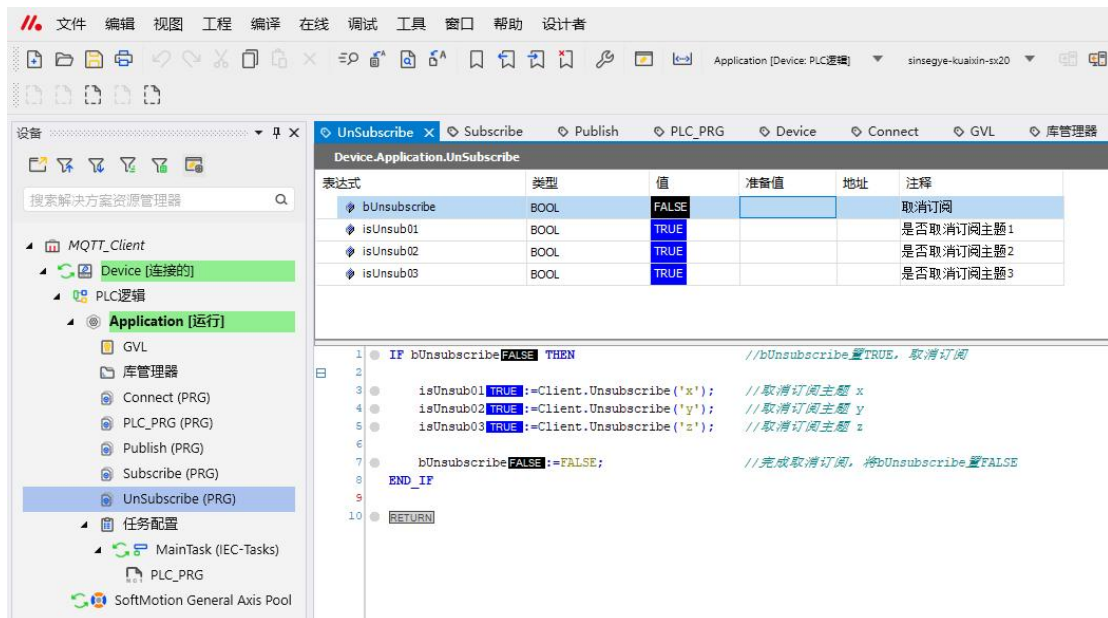
SoftMotion General Axis Pool

表达式	类型	值	准备值	地址	注释
bUnsubscribe	BOOL	FALSE	TRUE		取消订阅
isUnsub01	BOOL	FALSE			是否取消订阅主题 1
isUnsub02	BOOL	FALSE			是否取消订阅主题 2
isUnsub03	BOOL	FALSE			是否取消订阅主题 3

```

1 IF bUnsubscribe FALSE<TRUE> THEN //bUnsubscribe置TRUE, 取消订阅
2
3   isUnsub01 FALSE :=Client.Unsubscribe('x'); //取消订阅主题 x
4   isUnsub02 FALSE :=Client.Unsubscribe('y'); //取消订阅主题 y
5   isUnsub03 FALSE :=Client.Unsubscribe('z'); //取消订阅主题 z
6
7   bUnsubscribe FALSE<TRUE> :=FALSE; //完成取消订阅, 将bUnsubscribe置FALSE
8 END_IF
9
10 RETURN
  
```

取消成功



设备: MQTT_Client

库管理器

Connect (PRG)

PLC_PRG (PRG)

Publish (PRG)

Subscribe (PRG)

UnSubscribe (PRG)

任务配置

MainTask (IEC-Tasks)

PLC_PRG

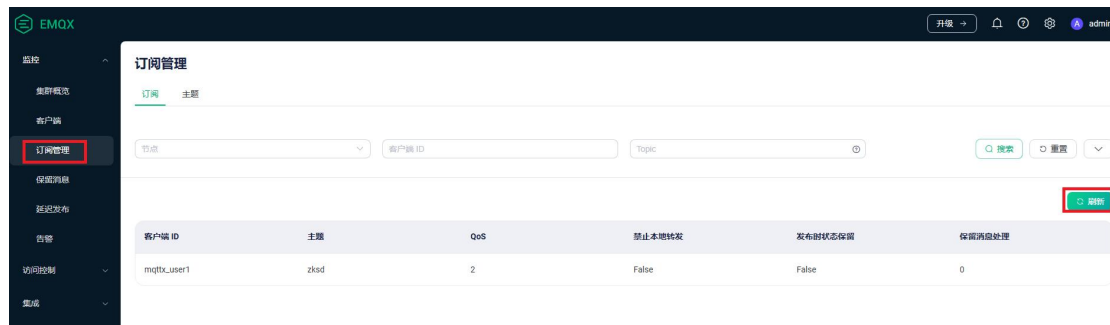
SoftMotion General Axis Pool

表达式	类型	值	准备值	地址	注释
bUnsubscribe	BOOL	FALSE			取消订阅
isUnsub01	BOOL	TRUE			是否取消订阅主题 1
isUnsub02	BOOL	TRUE			是否取消订阅主题 2
isUnsub03	BOOL	TRUE			是否取消订阅主题 3

```

1 IF bUnsubscribe FALSE THEN //bUnsubscribe置TRUE, 取消订阅
2
3   isUnsub01 TRUE :=Client.Unsubscribe('x'); //取消订阅主题 x
4   isUnsub02 TRUE :=Client.Unsubscribe('y'); //取消订阅主题 y
5   isUnsub03 TRUE :=Client.Unsubscribe('z'); //取消订阅主题 z
6
7   bUnsubscribe FALSE :=FALSE; //完成取消订阅, 将bUnsubscribe置FALSE
8 END_IF
9
10 RETURN
  
```

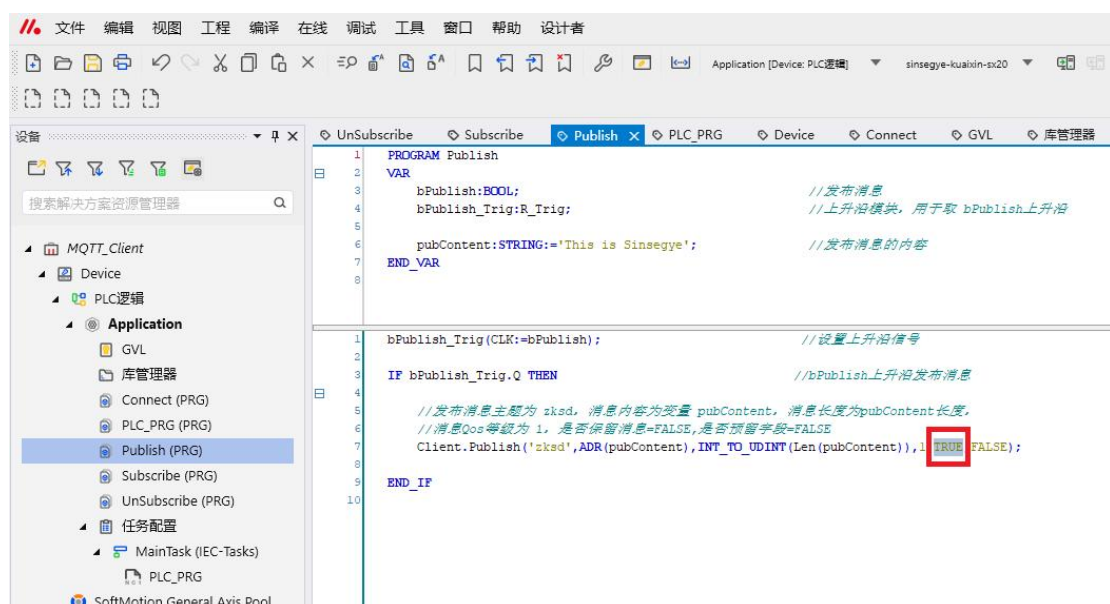
查看 Broker 的订阅管理，点击“刷新”，Sinseggye 取消订阅。



保留消息

MQTT 保留消息(Retained Messages)是一种在 MQTT 中发布持久化消息的方式。当客户端发布一条保留消息，它会在 MQTT 服务器上保存这条消息。当新的订阅者连接到服务器时，服务器会将这条保留消息发送给新的订阅者。一个主题只会存储一条保留消息，新的保留消息会替换旧的保留消息。

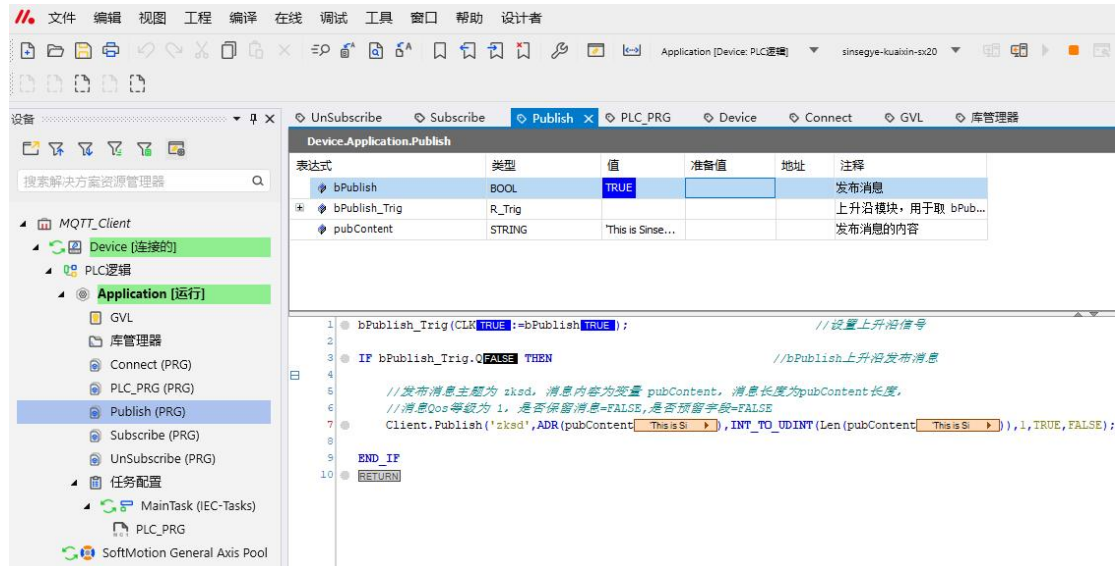
设置使用 Client.Publish()方法的 bRetain 参数为 TRUE，发布的消息即为保留消息。



登录工智机，将 PLC 下载至工智机中，启动 PLC。

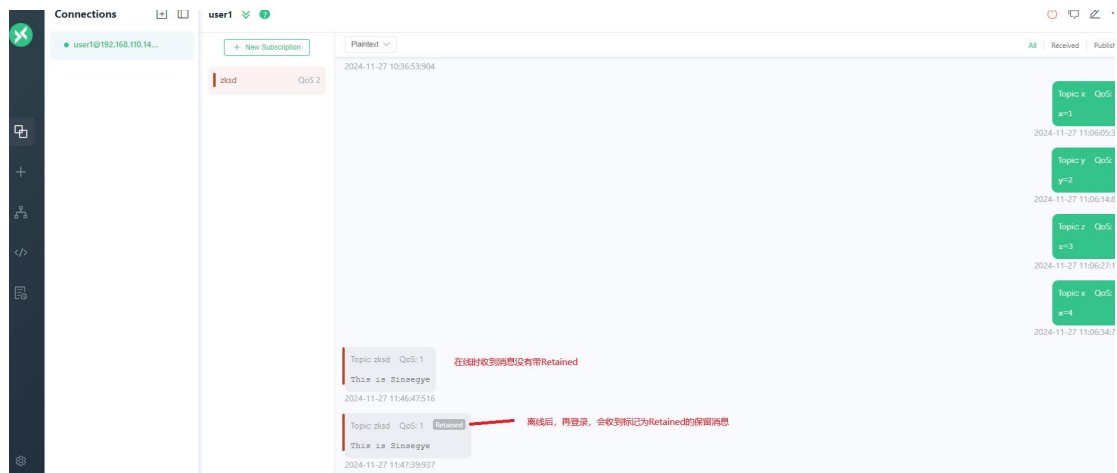
1.先将 bConnect 变量置为 TRUE,Ctrl+F7 写入变量，客户端与 MQTT Broker 连接。

2.再将 bPublish 变量置为 TRUE,Ctrl+F7 写入变量，发布保留消息。



如果发送保留消息时，订阅 zksd 的 MQTTX 客户端在线，则正常收到消息。

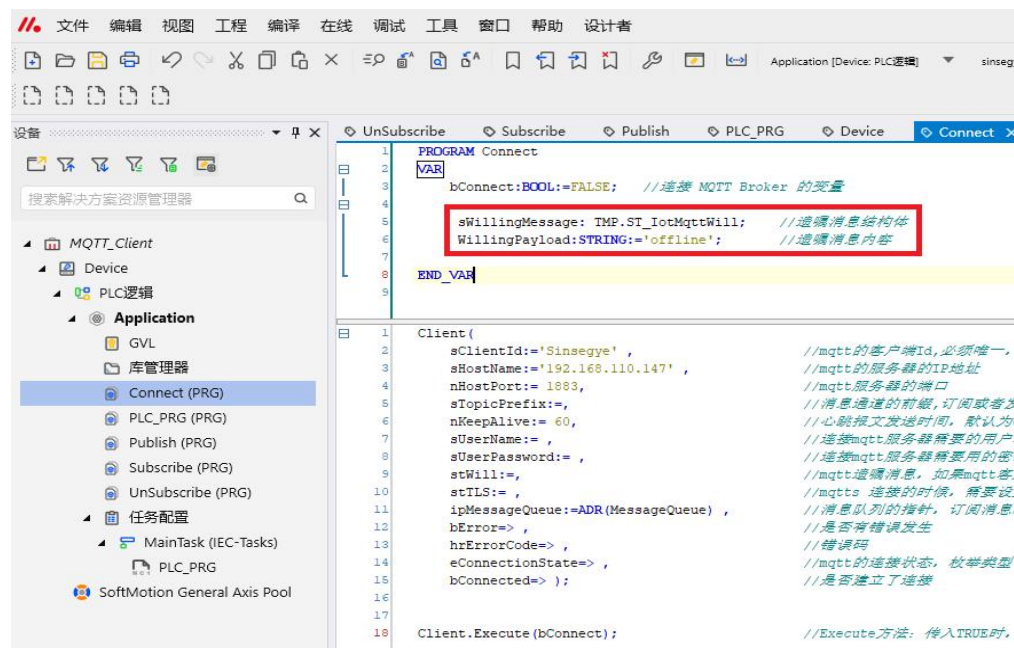
如果发送保留消息时，订阅 zksd 的 MQTTX 客户端离线，则当 MQTTX 上线后会收到标记为 Retained 的保留消息。



遗嘱消息

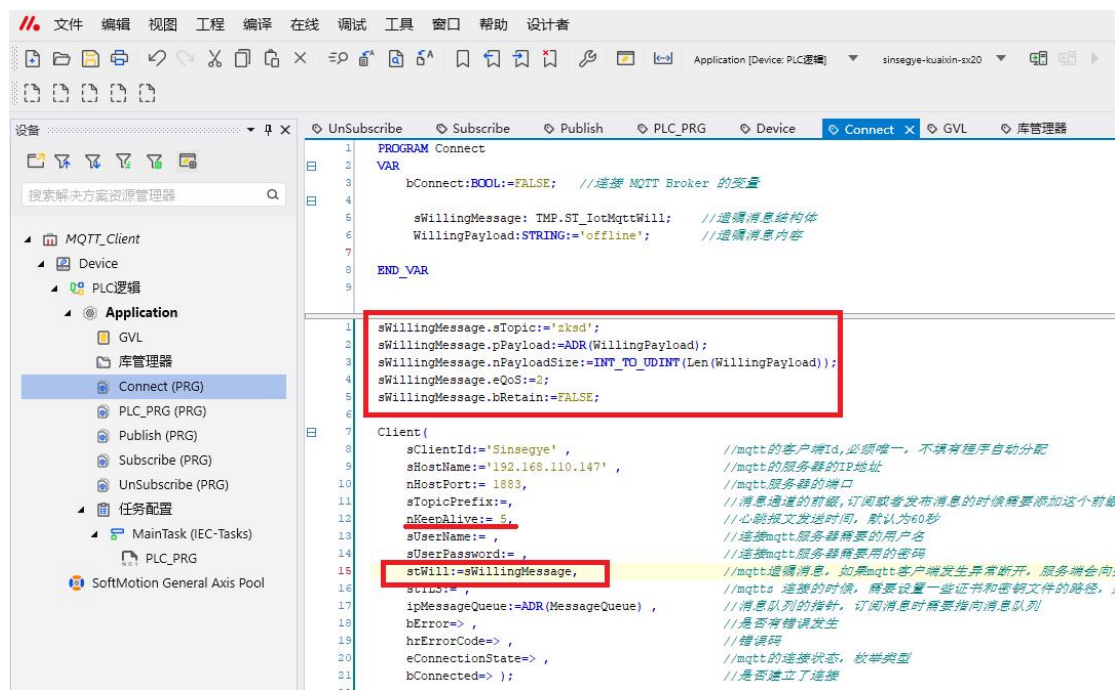
遗嘱消息是一种特殊的消息，用于客户端**异常**断开连接时，向其他订阅者通知其离线状态的消息。

在 Connect 程序中，声明遗嘱消息。



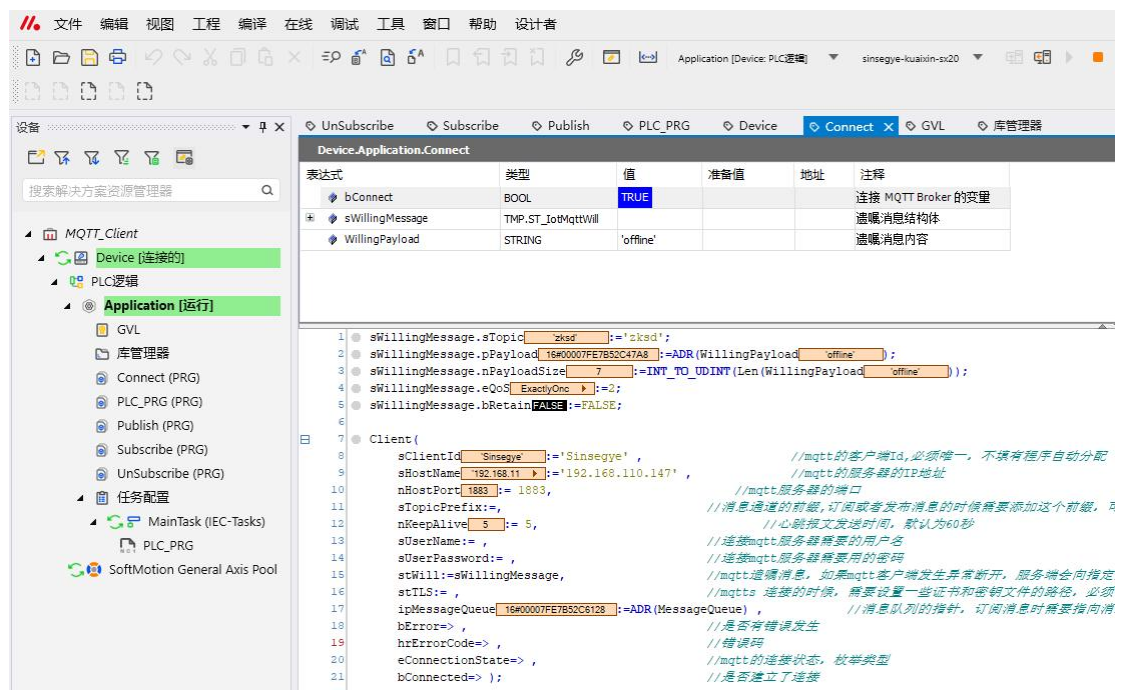
设定遗嘱消息结构体成员，并将结构体传入 Client 客户端的 stWill 参数。同时，为了方便演示异常断开，将心跳报文发送时间设为 5s

遗嘱结构体成员	类型	设定例	说明
sTopic	String	'zksd'	遗嘱消息主题为 zksd
pPayload	Pointer	ADR(WillingPayload)	遗嘱消息内容指针
nPayloadSize	UDINT	INT_TO_UDINT(Len(WillingPayload))	遗嘱消息数据长度
eQoS	UINT	2	eQoS 消息质量等级 2
bRetain	Bool	FALSE	不是保留消息



登录工智机，将 PLC 下载至工智机中，启动 PLC。

1.先将 bConnect 变量置为 TRUE,Ctrl+F7 写入变量，客户端与 MQTT Broker 连接。

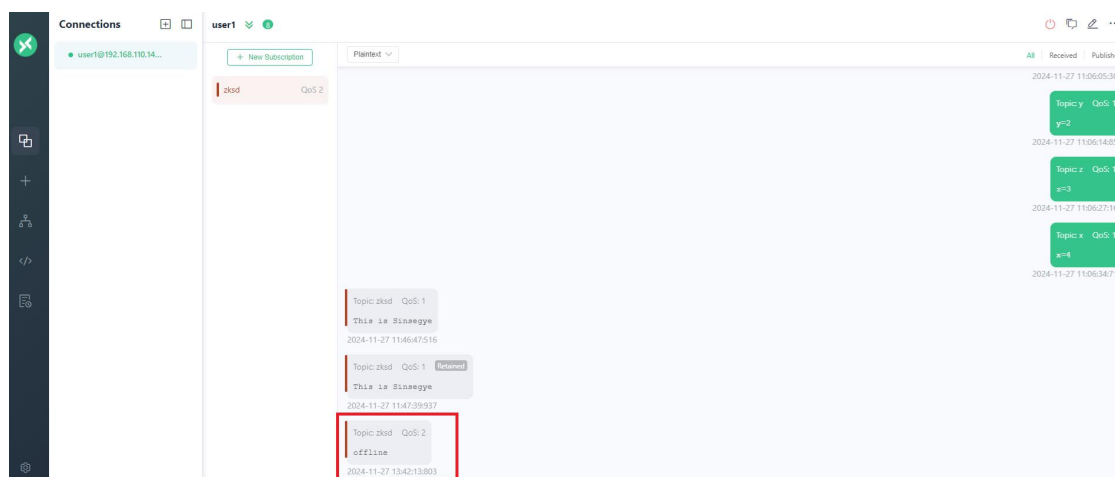


2.手动断开工智机网络连接线。

手动断开网络



3.在 MQTTX 客户端，收到遗嘱消息。



功能介绍

1、功能块 FB_IotMqttClient 介绍



- 输入参数

参数名称	参数类型	描述说明
sClientId	String	mqtt 的客户端 Id,必须是唯一标识, 不填有程序自动分配
sHostName	String	mqtt 的服务器的 IP 地址, 默认为本机 IP
nHostPort	UINT	mqtt 服务器的端口, 默认为 1883
sTopicPrefix	String	消息通道的前缀,订阅或者发布消息的时候需要添加这个前缀, 可不填
nKeepAlive	UINT	心跳报文发送时间, 默认为 60 秒
sUserName	String	设置连接 mqtt 服务器用户名。如果 Broker 需要提供用户名和密码连接。
sUserPassword	String	设置连接 mqtt 服务器密码。如果 Broker 需要提供用户名和密码连接。
stWill	结构体 ST_IotMqttWill	mqtt 遗嘱消息, 如果 mqtt 客户端发生异常断开, 服务端会向指定的 topic 发送消息

stTLS	结构体 ST_lotMqttTls	mqtt 连接的时候，需要设置一些证书和密钥文件的路径，必须保证路径存在
ipMessageQueue	指针	消息队列的指针，订阅消息时需要指向消息队列

- 输出参数

参数名称	参数类型	描述说明
bError	Bool	是否报错，True 为有错误，False 为正常
hrErrorCode	UINT	错误码
eConnectionState	UINT	mqtt 的连接状态，枚举类型
bConnected	Bool	是否建立连接，True 为连接，False 为没有连接

2、方法 FB_lotMqttClient.Execute()介绍

Void **Execute**(BOOL bConnect);

接口	类型	说明
bConnect	BOOL	TRUE 表示建立连接

3、方法 FB_lotMqttClient.Publish()介绍

Bool **Publish**(String sTopic, POINTER TO UDINT pPayload,UDINT nPayloadSize,TclotMqttQos eQoS,Bool bRetain, Bool bQueue);

接口	类型	说明
sTopic	String	发布消息的主题
pPayload	PVOID	指针指向发布消息内容

nPayloadSize	UDINT	发布消息内容长度
eQoS	枚举类型 TclotMqttQos	AtMostOnceDelivery := 0, //最多一次 AtLeastOnceDelivery :=1, //至少一次 ExactlyOnceDelivery := 2 //确切一次
bRetain	Bool	TRUE 表示发布保留消息
bQueue	Bool	保留勿用
返回值	Bool	发送是否成功，TRUE 成功，FALSE 不成功

4、方法 **FB_lotMqttClient.Subscribe()**介绍

Bool Subscribe(String sTopic, TclotMqttQos eQoS);

接口	类型	说明
sTopic	String	订阅消息的主题
eQoS	枚举类型 TclotMqttQos	AtMostOnceDelivery := 0, //最多一次 AtLeastOnceDelivery :=1, //至少一次 ExactlyOnceDelivery := 2 //确切一次
返回值	Bool	订阅是否成功，TRUE 成功，FALSE 不成功

5、方法 **FB_IotMqttClient.Unsubscribe()**介绍

Bool Unsubscribe(String sTopic);

接口	类型	说明
sTopic	String	取消订阅消息的主题
返回值	Bool	取消订阅是否成功，TRUE 成功，FALSE 不成功

6、方法 **FB_IotMqttMessageQueue.Dequeue()**介绍

Bool Dequeue(POINTER TO FB_IotMqttMessage fbMessage);

接口	类型	说明
fbMessage	POINTER TO FB_IotMqttMessage	指针，指向 FB_IotMqttMessage 从当前的消息队列里面获取一条消息，并赋值给 FB_IotMqttMessage 消息的对象
返回值	Bool	获取消息是否成功，TRUE 成功，FALSE 不成功

7、方法 **FB_IotMqttMessageQueue.ResetQueue()**介绍

Bool ResetQueue();

接口	类型	说明
----	----	----

返回值	Bool	清空当前消息队列里面的所有消息，TRUE 成功，FALSE 不成功
-----	------	-----------------------------------

8、方法 **FB_lotMqttMessage.CompareTopic()**介绍

Bool CompareTopic(String sTopic);

接口	类型	说明
sTopic	String	比较输入的 Topic
返回值	Bool	比较输入的 Topic 与当前消息的 Topic 是否一致，TRUE 为一致，FALSE 不一致

9、方法 **FB_lotMqttMessage.GetTopic()**介绍

Bool GetTopic(POINTER TO STRING pTopic, UINT nTopicSize);

接口	类型	说明
pTopic	POINTER TO STRING	指针指向主题消息内容
nTopicSize	UINT	主题消息长度
返回值	Bool	获取消息的 Topic，TRUE 获取成功，FALSE 获取失败

10、方法 **FB_IotMqttMessage.GetPayload()**介绍

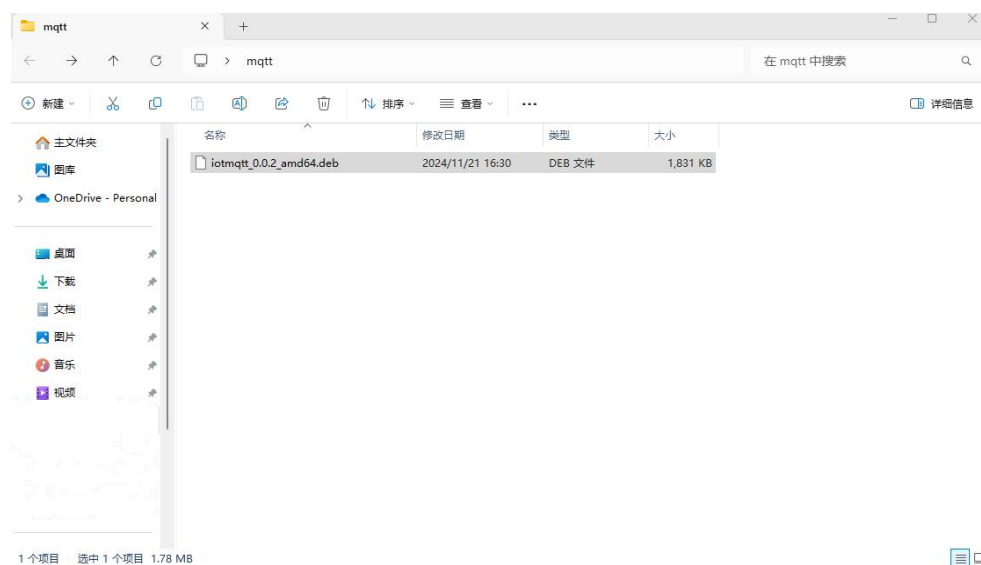
Bool **GetPayload**(PVOID pPayload, UDINT nPayloadSize, BOOL bSetNullTermination);

接口	类型	说明
pPayload	PVOID	指针指向负载消息内容
nPayloadSize	UDINT	负载消息内容长度
bSetNullTermination	Bool	消息是否以 NULL 字符结束(即'\0'), TRUE 为是, FALSE 为否
返回值	Bool	获取消息的负载, 也就是订阅收到的消息内容, TRUE 获取成功, FALSE 获取失败

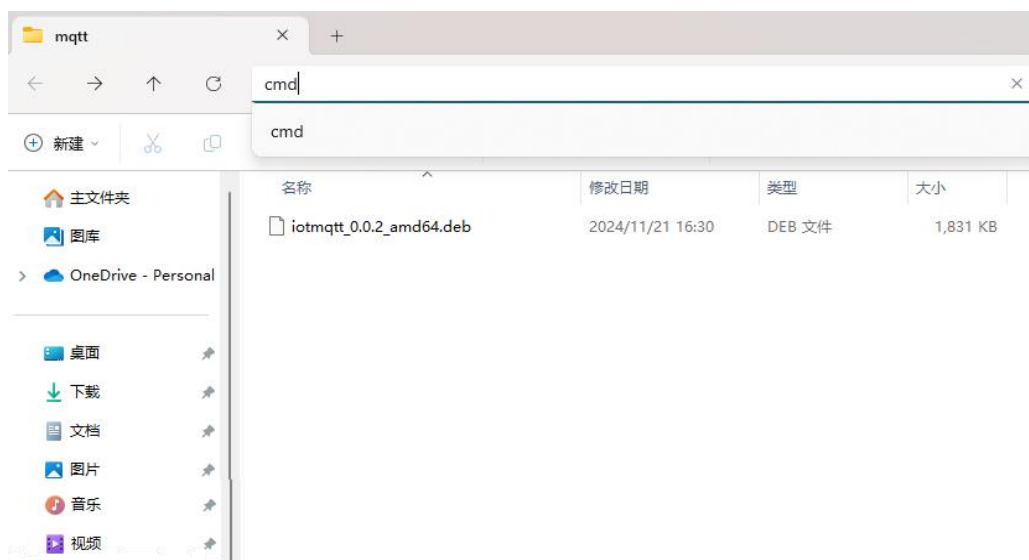
附录

一、工智机安装 deb 组件方法

- 1.将个人电脑连接至交换机中，并使其 IP 地址与工智机在同一网段中。
- 2.将工智机组件 `iotmqtt_0.0.2_amd64.deb` 下载至个人电脑中，放置在文件夹下。

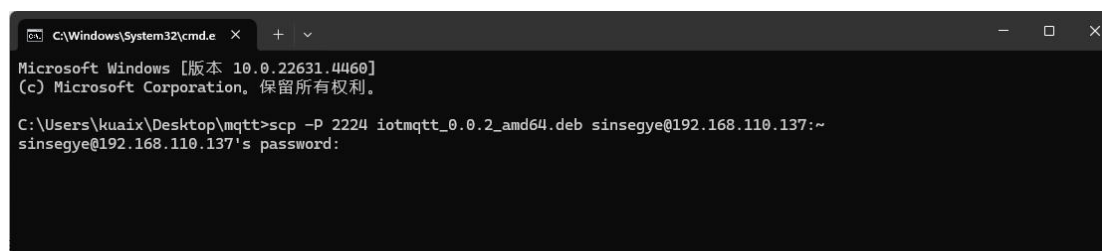


- 3.在文件夹位置处输入“cmd”，回车。



4.通过 scp 命令将 iotmqtt_0.0.2_amd64.deb 文件传输到客户端工智机中。

```
scp -P 2224 iotmqtt_0.0.2_amd64.deb sinsegye@192.168.110.137:~
```



密码为 1.

```
C:\Windows\System32\cmd.e x + v
Microsoft Windows [版本 10.0.22631.4460]
(c) Microsoft Corporation. 保留所有权利。

C:\Users\kuaix\Desktop\mqtt>scp -P 2224 iotmqtt_0.0.2_amd64.deb sinsegye@192.168.110.137:~
sinsegye@192.168.110.137's password:
iotmqtt_0.0.2_amd64.deb                                100% 1830KB  10.2MB/s   00:00

C:\Users\kuaix\Desktop\mqtt>
```

通过 ssh 远程连接到工控机命令行

```
ssh -p 2224 sinsegye@192.168.110.137
```

```
sinsegye@sinsegye-sx21: ~ x + v

C:\Users\kuaix\Desktop\mqtt>scp -P 2224 iotmqtt_0.0.2_amd64.deb sinsegye@192.168.110.137:~
sinsegye@192.168.110.137's password:
iotmqtt_0.0.2_amd64.deb                                100% 1830KB  10.2MB/s   00:00

C:\Users\kuaix\Desktop\mqtt>ssh -p 2224 sinsegye@192.168.110.137
sinsegye@192.168.110.137's password:
Welcome to Ubuntu 22.04.1 LTS (GNU/Linux 5.15.51-sinsegye-s1.5-rc.1 x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:    https://landscape.canonical.com
 * Support:       https://ubuntu.com/advantage

System information as of Tue Nov 26 03:35:53 PM CST 2024

System load:  2.3359375      Processes:            148
Usage of /:   17.7% of 211.12GB Users logged in:      1
Memory usage: 69%           IPv4 address for enp1s0f0: 192.168.110.137
Swap usage:   0%            IPv4 address for virbr0:  192.168.3.1

 * Strictly confined Kubernetes makes edge and IoT secure. Learn how MicroK8s
   just raised the bar for easy, resilient and secure K8s cluster deployment.

https://ubuntu.com/engage/secure-kubernetes-at-the-edge
New release '24.04.1 LTS' available.
Run 'do-release-upgrade' to upgrade to it.

Last login: Tue Nov 26 15:21:11 2024 from 192.168.110.92
sinsegye@sinsegye-sx21:~$
```

通过 dpkg 命令，安装 iotmqtt_0.0.2_amd64.deb 文件

```
sudo dpkg -i iotmqtt_0.0.2_amd64.deb
```

```
sinsegye@sinsegye-sx21: ~  
Welcome to Ubuntu 22.04.1 LTS (GNU/Linux 5.15.51-sinsegye-s1.5-rc.1 x86_64)  
  
* Documentation:  https://help.ubuntu.com  
* Management:    https://landscape.canonical.com  
* Support:       https://ubuntu.com/advantage  
  
System information as of Tue Nov 26 03:35:53 PM CST 2024  
  
System load:  2.3359375      Processes:            148  
Usage of /:   17.7% of 211.12GB Users logged in:       1  
Memory usage: 69%          IPv4 address for enp1s0f0: 192.168.110.137  
Swap usage:   0%            IPv4 address for virbr0:  192.168.3.1  
  
* Strictly confined Kubernetes makes edge and IoT secure. Learn how MicroK8s  
  just raised the bar for easy, resilient and secure K8s cluster deployment.  
  
  https://ubuntu.com/engage/secure-kubernetes-at-the-edge  
New release '24.04.1 LTS' available.  
Run 'do-release-upgrade' to upgrade to it.  
  
Last login: Tue Nov 26 15:21:11 2024 from 192.168.110.92  
sinsegye@sinsegye-sx21:~$ sudo dpkg -i iotmqtt_0.0.2_amd64.deb  
(Reading database ... 127374 files and directories currently installed.)  
Preparing to unpack iotmqtt_0.0.2_amd64.deb ...  
Unpacking iotmqtt (0.0.2) over (0.0.2) ...  
Setting up iotmqtt (0.0.2) ...  
Welcome! The package has been successfully installed.  
Processing triggers for libc-bin (2.35-0ubuntu3.1) ...  
sinsegye@sinsegye-sx21:~$
```

使用 nano 命令打开 SinsegyeRTE.cfg 文件

```
sudo nano /usr/local/etc/SinsegyeRTE/SinsegyeRTE.cfg
```

```

GNU nano 6.2 /usr/local/etc/SinseggyeRTE/SinseggyeRTE.cfg
[SysTask]. ]
; Max Linux priority(99)-OSPriority = effective Linux priority
; e.g. map IEC prio 0..15 to Linux prio 54..39 ,
; OSPriority.Realtime.Base=45
; OSPriority.Realtime.End=77
; avoid overlapping ranges:
; OSPriority.High.Base=78
; OSPriority.High.End=95
; OSPriority.Realtime.Base=0
; OSPriority.Realtime.End=31
Linux.MaxOSPriority=99

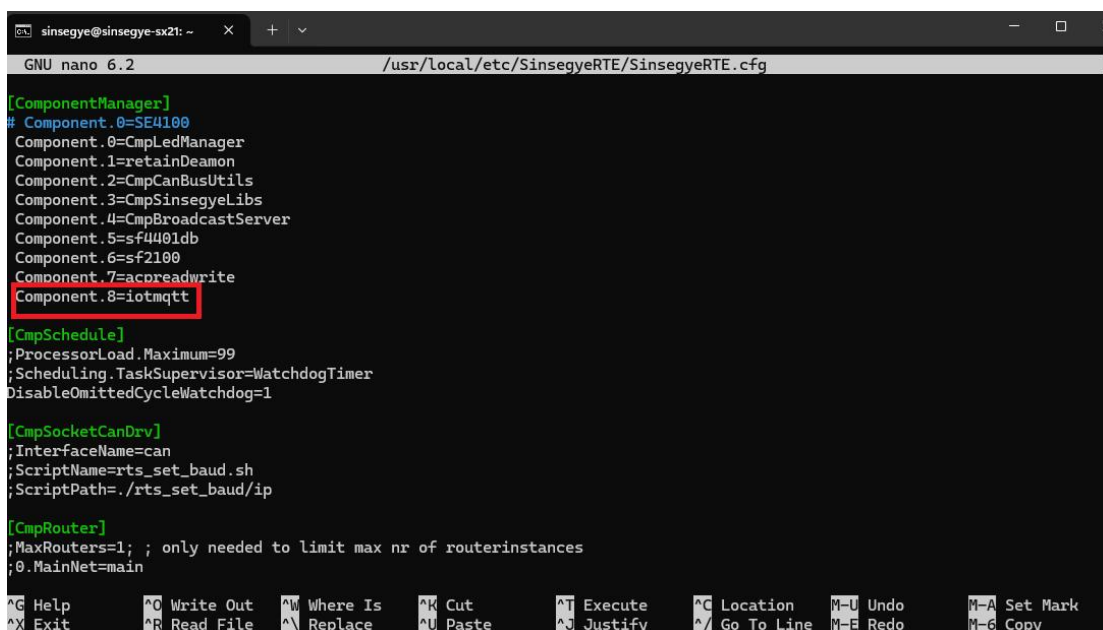
[CompSettings]
FileReference.0=SysFileMap.cfg, SysFileMap
; SplitDatabases=1
; FileReference.1=/etc/Configurationfile.cfg

[SysFile]
PlcLogicPrefix=1
FilePath=/opt/SinseggyeRTE
FilePath.1=/usr/local/etc,3S.dat
; ;FilePath.1=/home/plc

; Example for retains in shared memory, see runtime docu for details
; [CompRetain]
; Retain.SHM.Size=0x1FFFF

[ Read 152 lines (Converted from DOS format) ]
^G Help      ^O Write Out  ^W Where Is   ^K Cut        ^T Execute    ^C Location   M-U Undo     M-A Set Mark
^X Exit      ^R Read File  ^_ Replace    ^U Paste      ^J Justify    ^_ Go To Line M-E Redo     M-C Copy
  
```

在 [ComponentManager]下，添加 **Component.0=AcpPlcAccess**（红色部分视具体情况而增加）。Ctrl+X 退出一输入 Y—回车确定。



```
GNU nano 6.2 /usr/local/etc/SinseggyeRTE/SinseggyeRTE.cfg
[ComponentManager]
# Component.0=SE4100
Component.0=CmpLedManager
Component.1=retainDaemon
Component.2=CmpCanBusUtils
Component.3=CmpSinseggyeLibs
Component.4=CmpBroadcastServer
Component.5=sf4401db
Component.6=sf2100
Component.7=acpreadwrite
Component.8=iotmqtt
[CompSchedule]
;ProcessorLoad.Maximum=99
;Scheduling.TaskSupervisor=WatchdogTimer
DisableOmittedCycleWatchdog=1
[CompSocketCanDrv]
;InterfaceName=can
;ScriptName=rts_set_baud.sh
;ScriptPath=./rts_set_baud/ip
[CompRouter]
;MaxRouters=1; ; only needed to limit max nr of routerinstances
;0.MainNet=main
^G Help      ^O Write Out  ^W Where Is   ^K Cut        ^T Execute    ^C Location   M-U Undo     M-A Set Mark
^X Exit      ^R Read File  ^N Replace    ^U Paste      ^J Justify    ^_ Go To Line  M-E Redo     M-C Copy
```

重启工智机

```
sudo reboot
```

二、支持与服务

中科时代为公司产品及解决方案提供全方位支持与服务，确保针对相关问题给予快速且专业的响应。

资料下载

我们的资料下载专区涵盖了丰富的文件资源，包括应用案例、技术文档、产品介绍等，满足您的多样化需求。

资料下载地址：<https://help.sinsegye.com.cn/>

获取支持

如需中科时代产品的本地支持与服务，请随时联系我们。您可以通过访问我们的官方网站 www.sinsegye.com.cn，查找中科时代的分公司地址，并获取更多关于中科时代的信息。

此外，您还可以通过以下方式联系我们：

- 热线电话：400-013-2158
- 邮箱地址：support@sinsegye.com.cn