


Modbus Rtu Master操作手册



<Modbus Tcp Master V1.0>

作者	闫荣宝	 中科时代 SINSEGYE	基于PC技术的工智机新时代
----	-----	---	---------------

日期	2024/12/18	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
版本	V1.0	
Email	yanrongbao@sinsegye.com.cn	

更新说明

2024.12.18 / V1.0

功能	变更类型	说明	相关文档
1. 支持线圈功能码 1、5、15 2. 支持保持寄存器功能码 3、6、16 3. 支持离散输入功能码 2 4. 支持输入寄存器功能码 4	首次更新	/	/



中科时代

SINSEGYE

基于PC技术的工智机新时代

Modbus Rtu Master使用介绍

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险情况，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>	注意表示潜在的危险情况，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险情况，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>注意</div><div>危险的类型 说明不避开危险后果 说明如何避免危险的发生</div></div>	环境、设备或数据可能会被损坏。
--	-----------------

7、产品处理信息

例如，这些信息包括：行动建议、援助或有关产品的进一步信息。

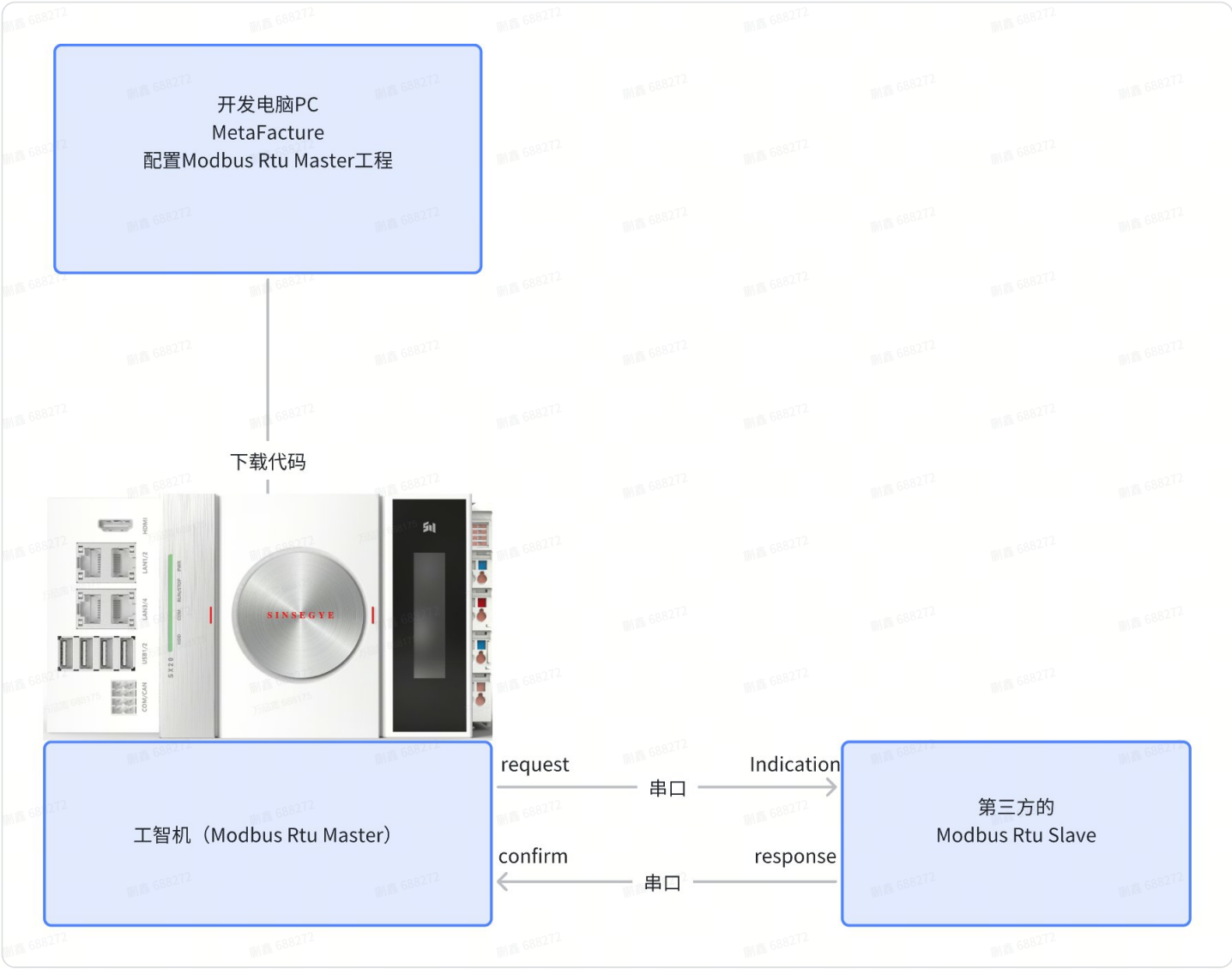
概述

Modbus 本身是一种信息交换的规范，Modbus Rtu则是透过串口来实现Modbus 的一种方式，因此所有的信息都是通过串口来传输；Modbus协议属于 C/S 架构，Modbus Rtu Master可以读写Rtu Slave的地址，实现数据交互；

使用场景

- 设备间数据采集和监控（如PLC与传感器、HMI的连接）。
- 过程自动化中的数据传输。
- 远程监控和控制系统。

整体架构



注：本手册中用到的中科时代的软件包，均可以从官网的子页面获取。官网提供的版本可能比本手册中提到的版本更高，一般情况下这不会影响您按照本手册的例子执行相应的操作

- Modbus Rtu是透过串口来实现Modbus的一种方式，因此所有讯息是透过串口来传输的，Modbus Rtu Master可以对Rtu Slave发送读或者写的指令，Rtu Slave收到指令后会回复确认信息，整个Modbus的沟通建立在一来一回的讯息交换上；
- 下表概述了各个产品组件

产品组件	描述说明
simodbusmaster_1.0.9_amd64.deb	Modbus rtu Master RTE组件
SF4100_ModbusMaster_1.0.0.2.library	Modbus rtu Master Metafacture 库文件

安装卸载

一、安装要求

- 中科时代出厂的工智机；
- 熟悉基础的Linux操作命令



开始安装前，请熟读[linux基础操作](#)中的操作示例

二、安装过程

1、工智机端安装Modbus Rtu Master RTE组件

- 上传deb包到工智机Linux环境的/home/sinsegye目录下
- 上传完成后在工智机上执行命令安装（参考下方截图，如果模块文件名发生变化则命令行中的文件名做相应更改）

```
1 cd $HOME
2 sudo dpkg -i simodbusmaster_1.0.9_amd64.deb
```

- 修改RTE的配置文件，ComponentManger模块下加入simodbusmaster

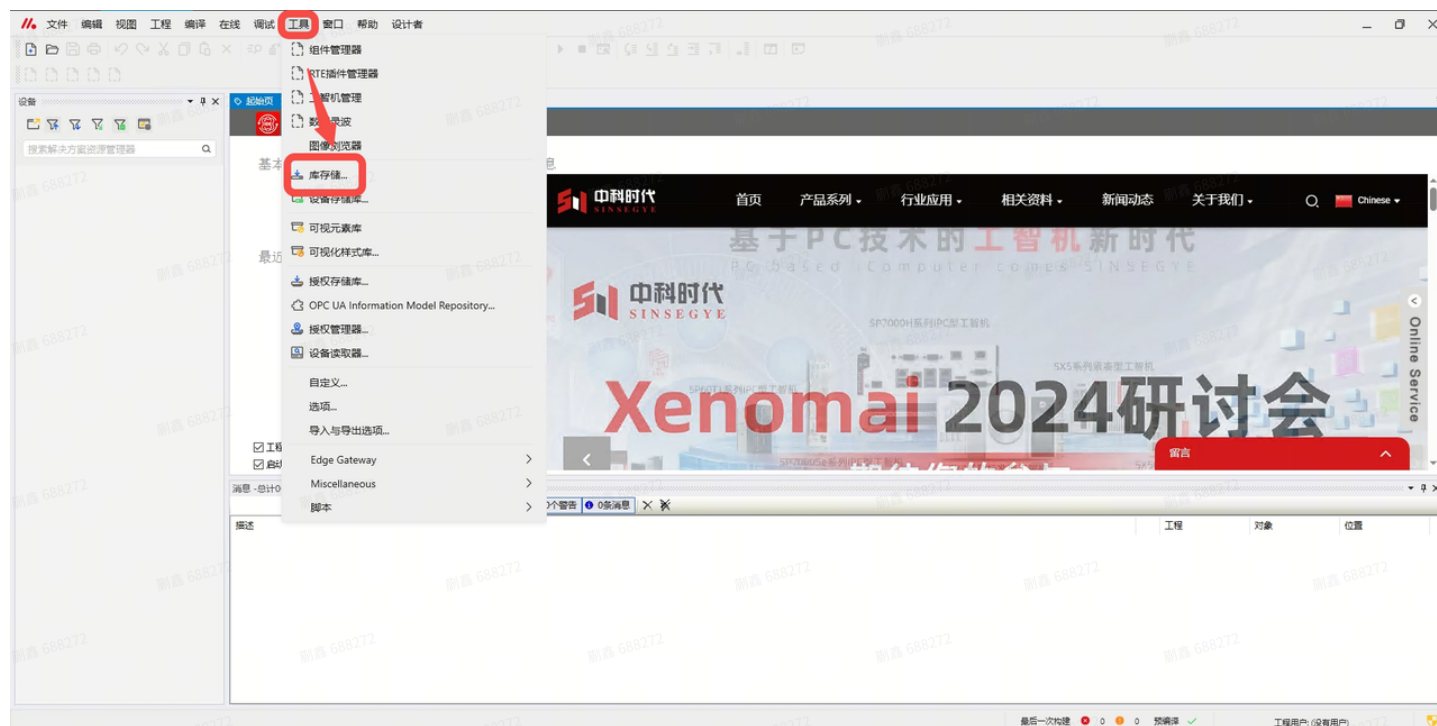
```
1 [ComponentManager]
2 Component.0=retainDeamon
3 Component.1=CmpCanBusUtils
4 Component.2=CmpSinsegyeLibs
5 Component.3=SinsegyeCmp
6 Component.4=simodbusmaster
```

- 重启RTE服务，使新加入的simodbusmaster被调用

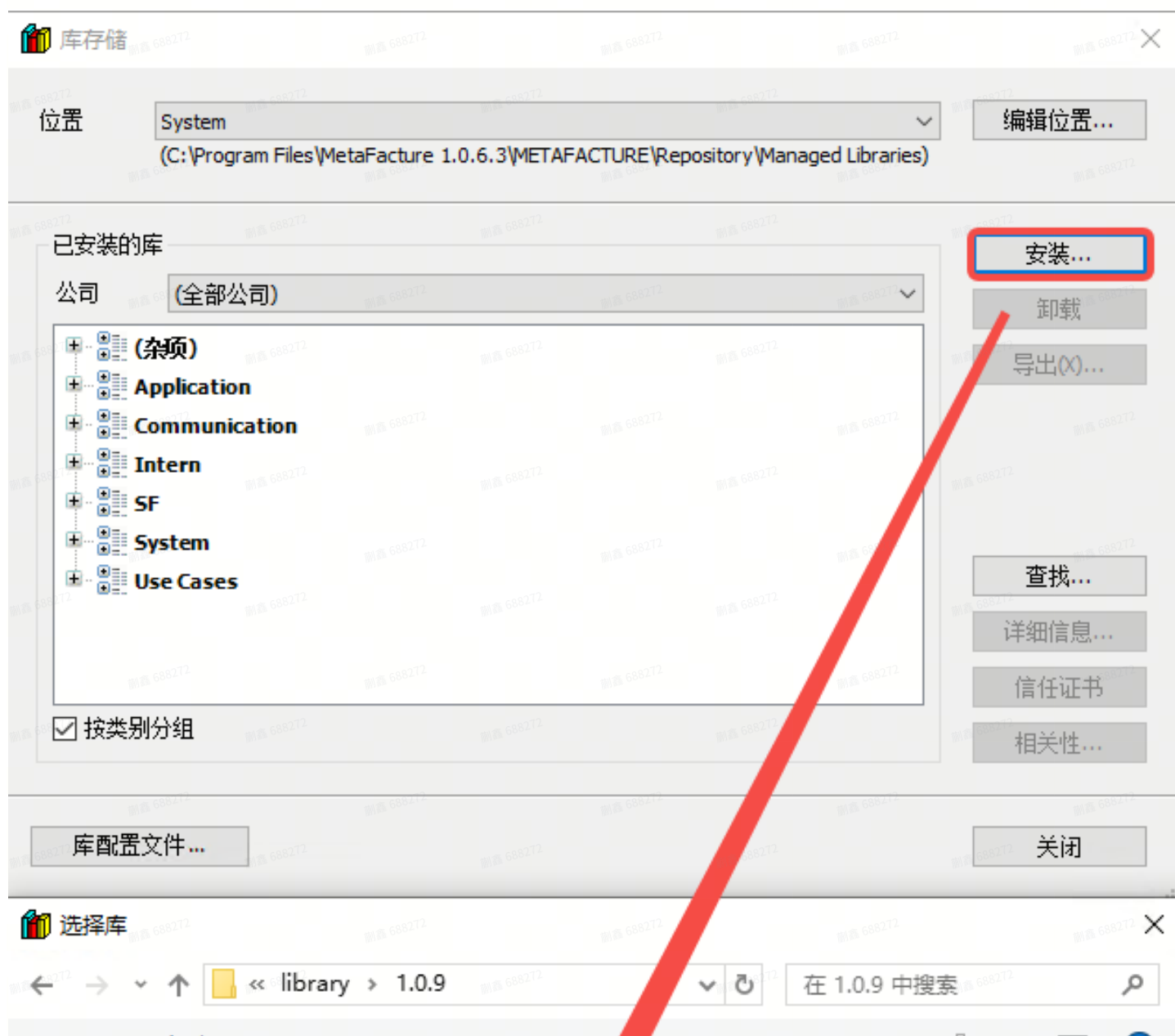
```
1 sudo systemctl restart sinsegyerte.service
```

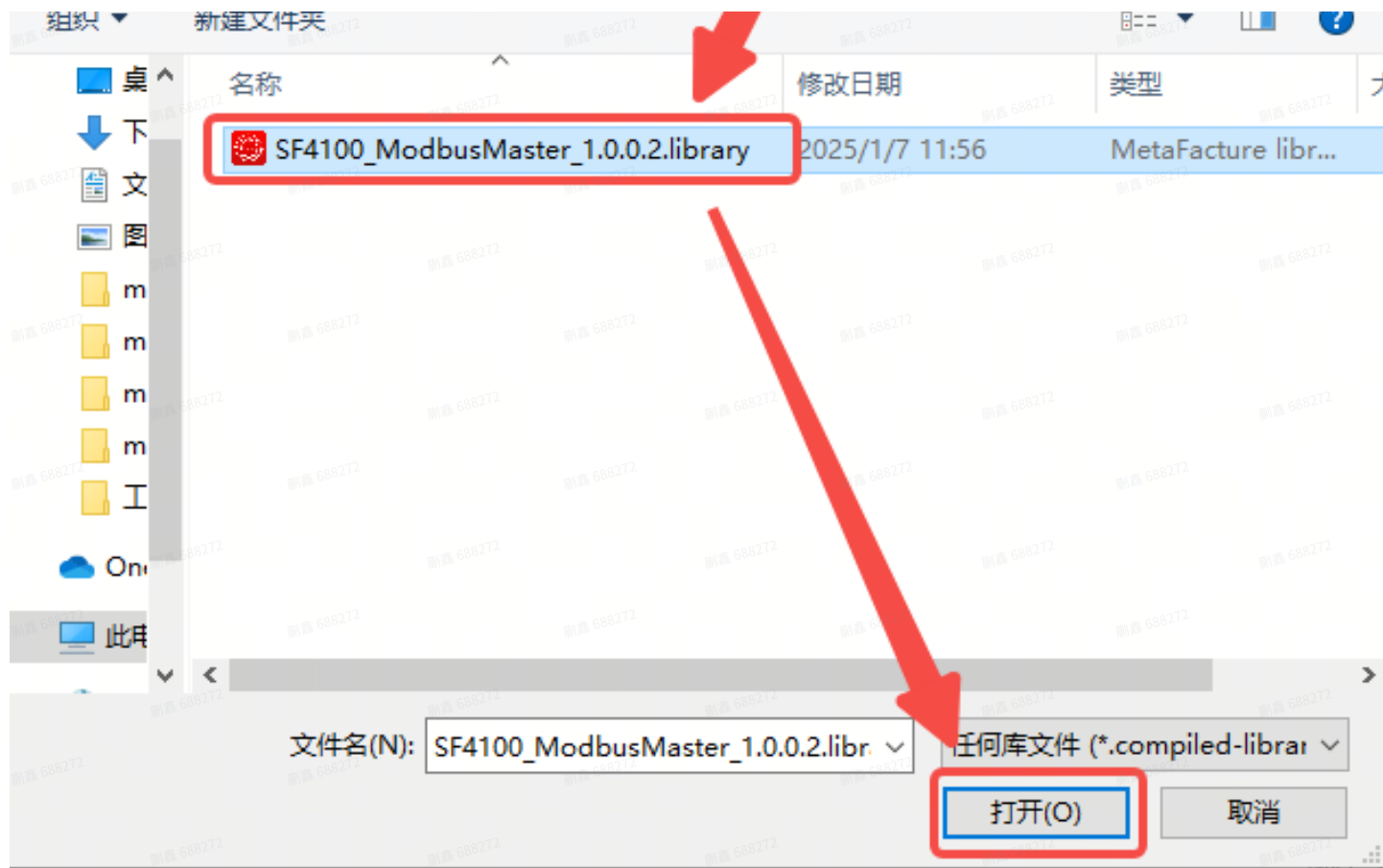
2、Metafacture安装library

- 打开Metafacture，点击“工具” -- “库存储”

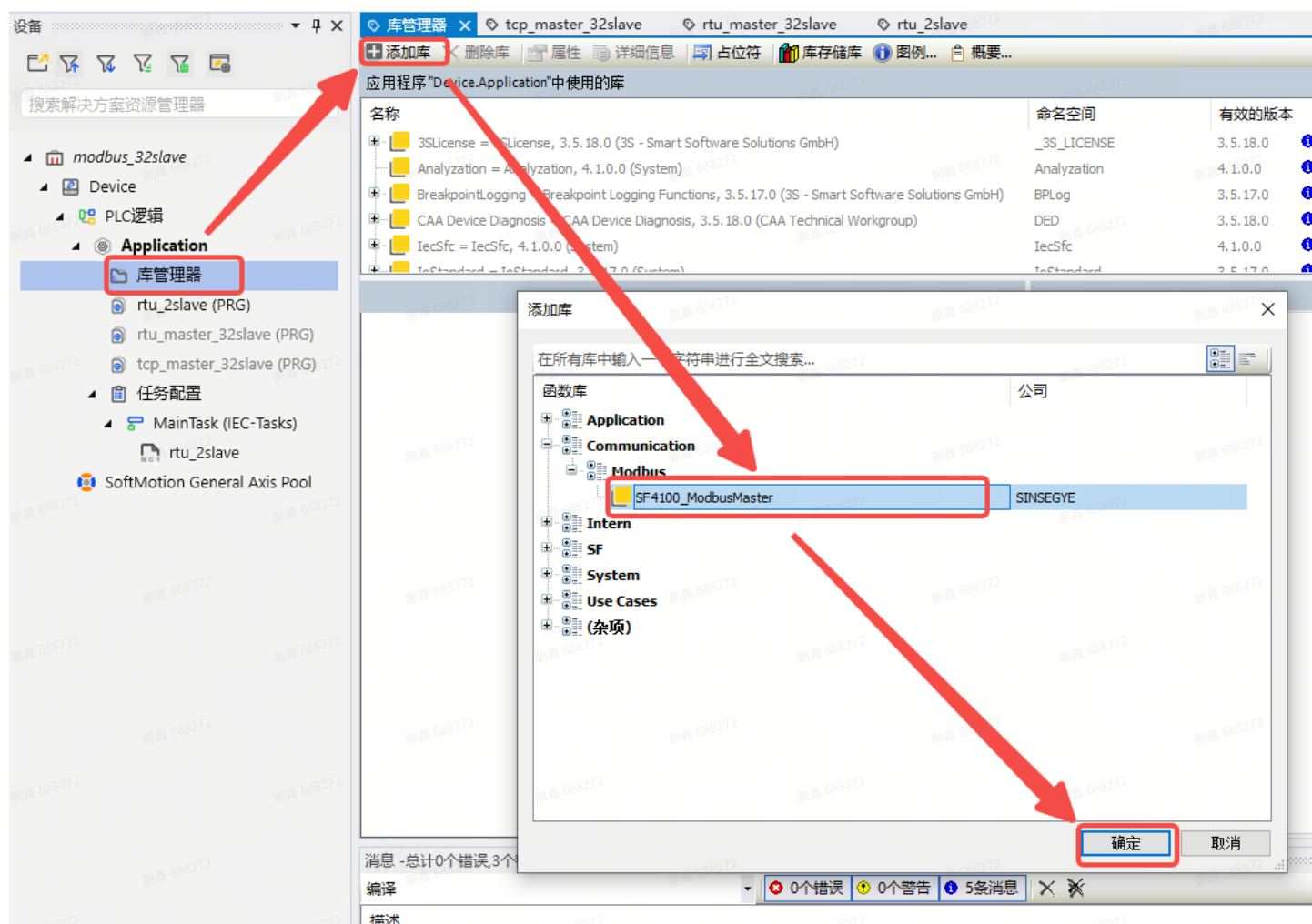


- 点击“安装” -- 选中modbus master的库文件，点击“打开”





- 工程中点击“库管理器” -- “添加库” -- 选中modbus库点击“确定”



三、更新安装

1、工控机端升级Modbus Rtu Master RTE组件

- 上传deb包到工控机Linux环境的/home/sinsegye目录下
- 上传完成后在工控机上执行命令安装（参考下方截图，如果模块文件名发生变化则命令行中的文件名做相应更改）

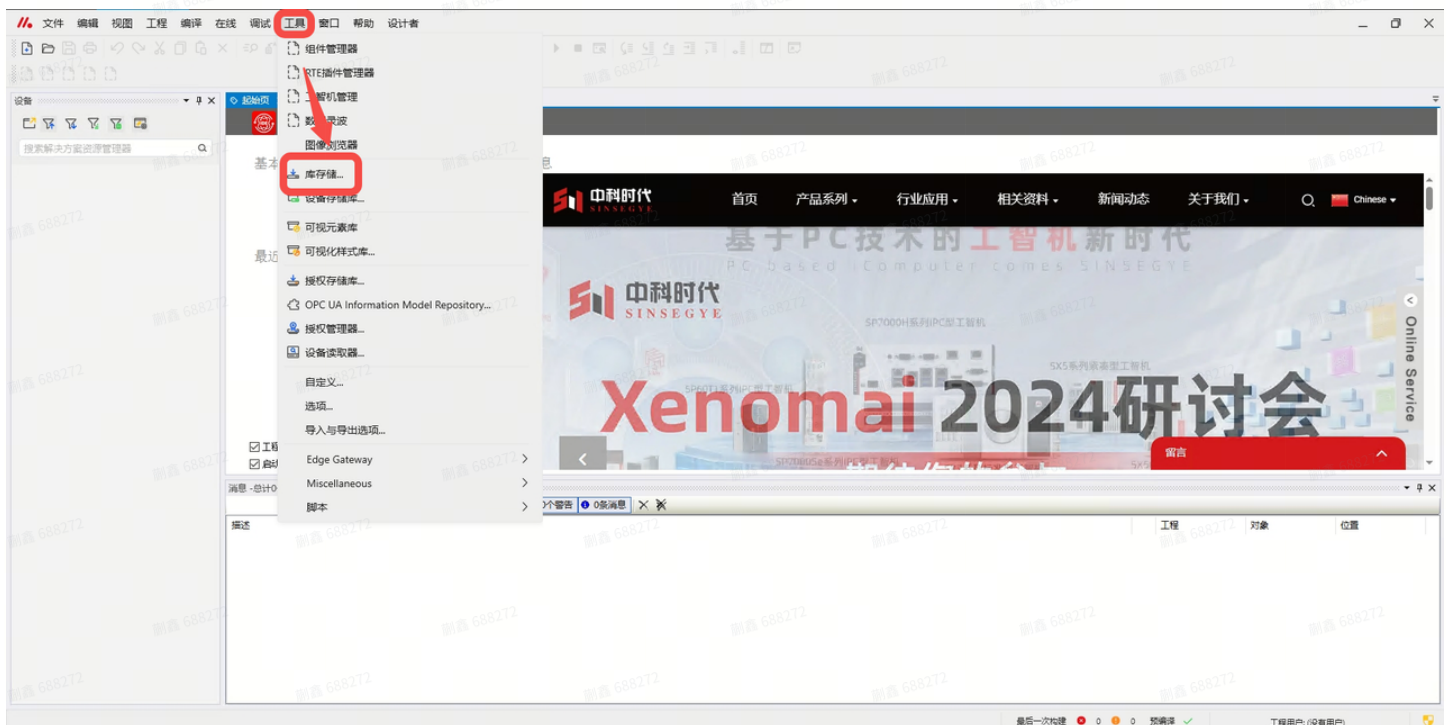
```
1 cd $HOME
2 sudo dpkg -i simodbusmaster_1.0.9_amd64.deb
```

- 重启RTE服务，使新升级的simodbusmaster生效

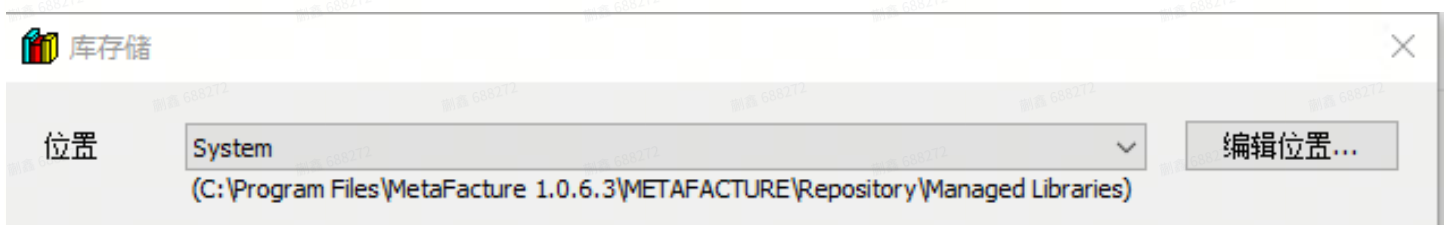
```
1 sudo systemctl restart sinsegyerte.service
```

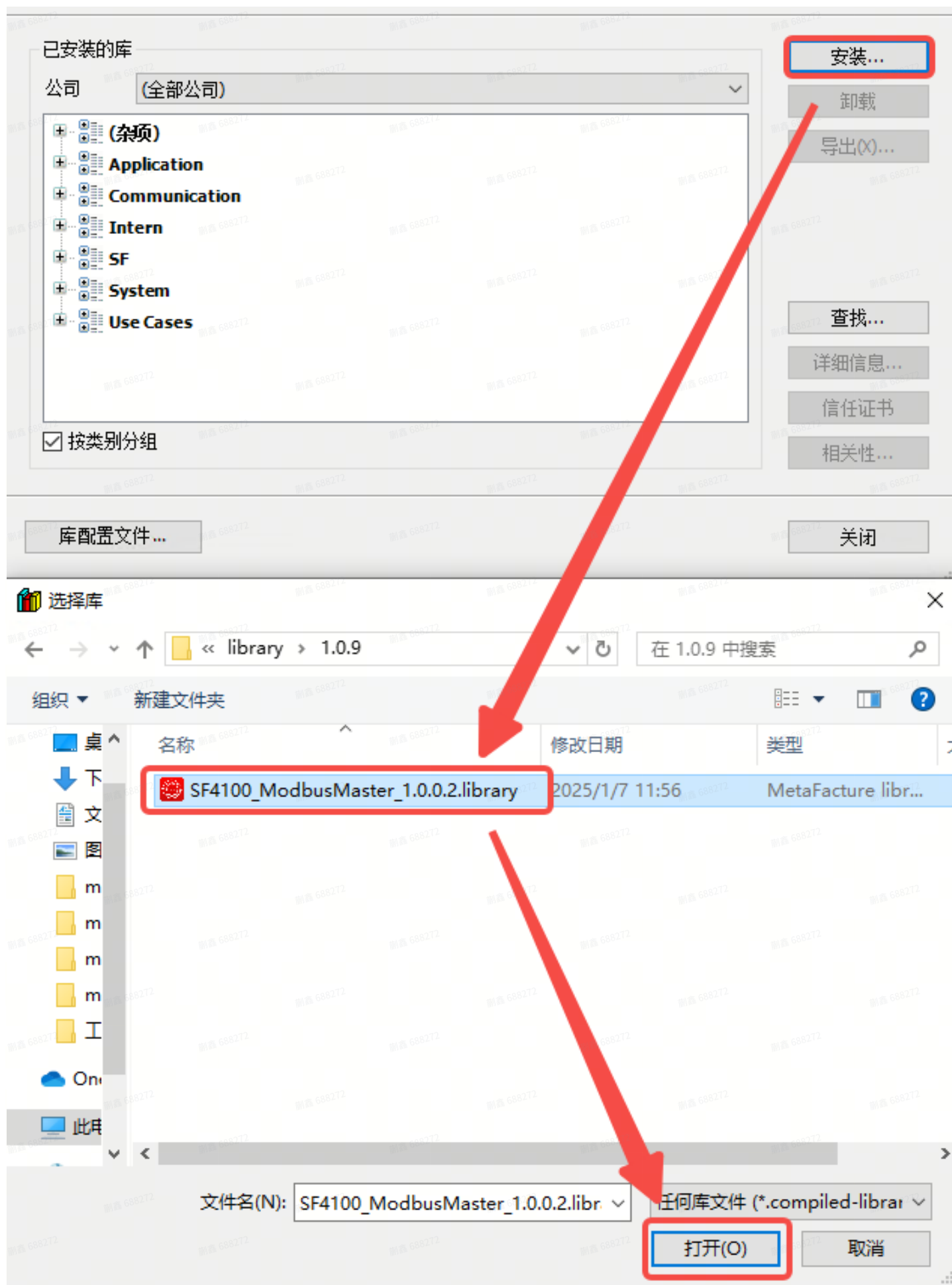
2、Metafacture升级library

- 打开Metafacture，点击“工具” -- “库存储”

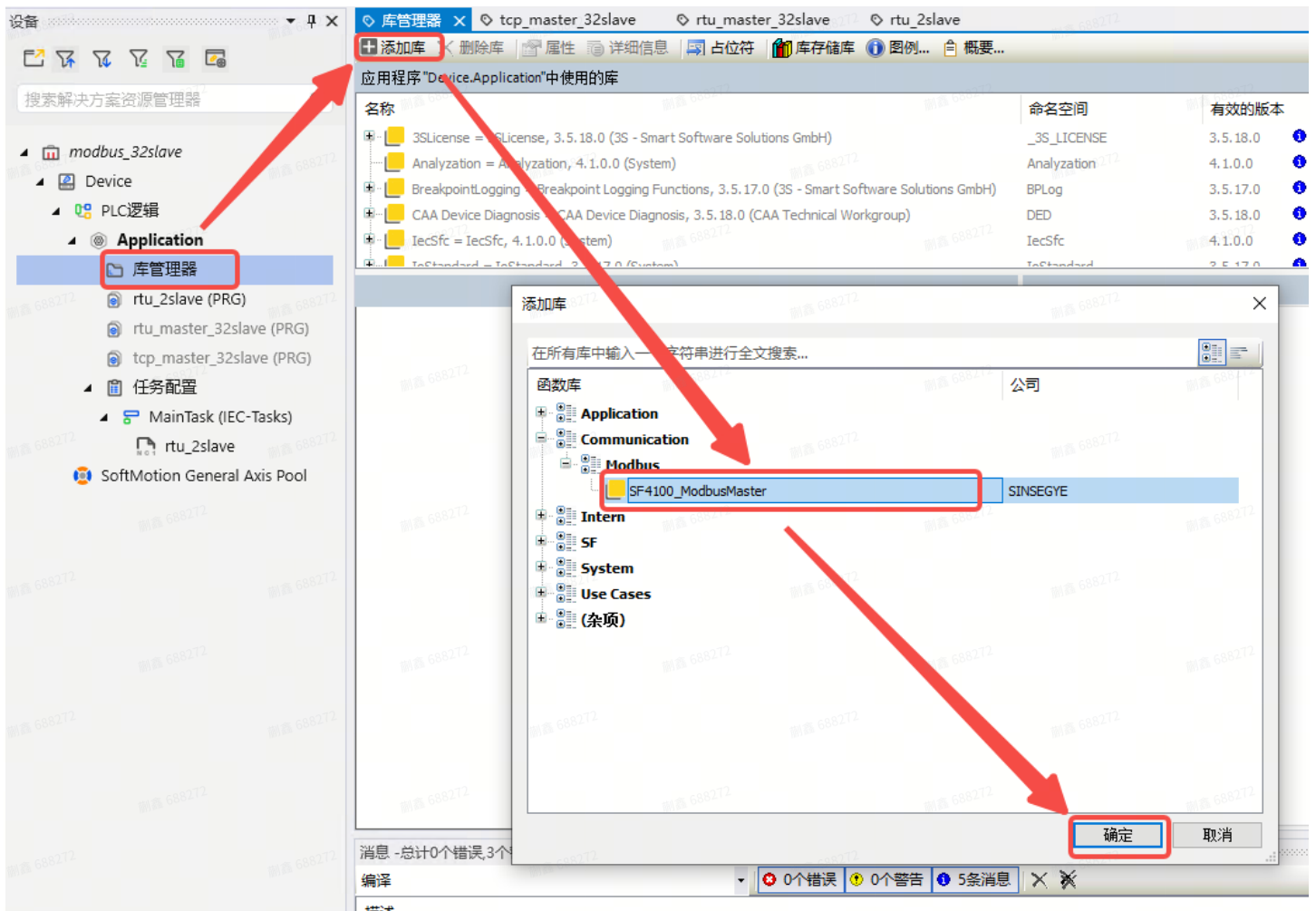


- 点击“安装” -- 选中新的modbus master的库文件，点击“打开”





- 工程中点击“库管理器” -- “添加库” -- 选中新的modbus master库点击“确定”



四、卸载过程

1、工智机端卸载Modbus Rtu Master RTE组件

- 工智机上执行命令卸载modbusmaster

```
1 sudo dpkg -r simodbusmaster
```

- 修改RTE的配置文件，ComponentManger模块下去掉simodbusmaster

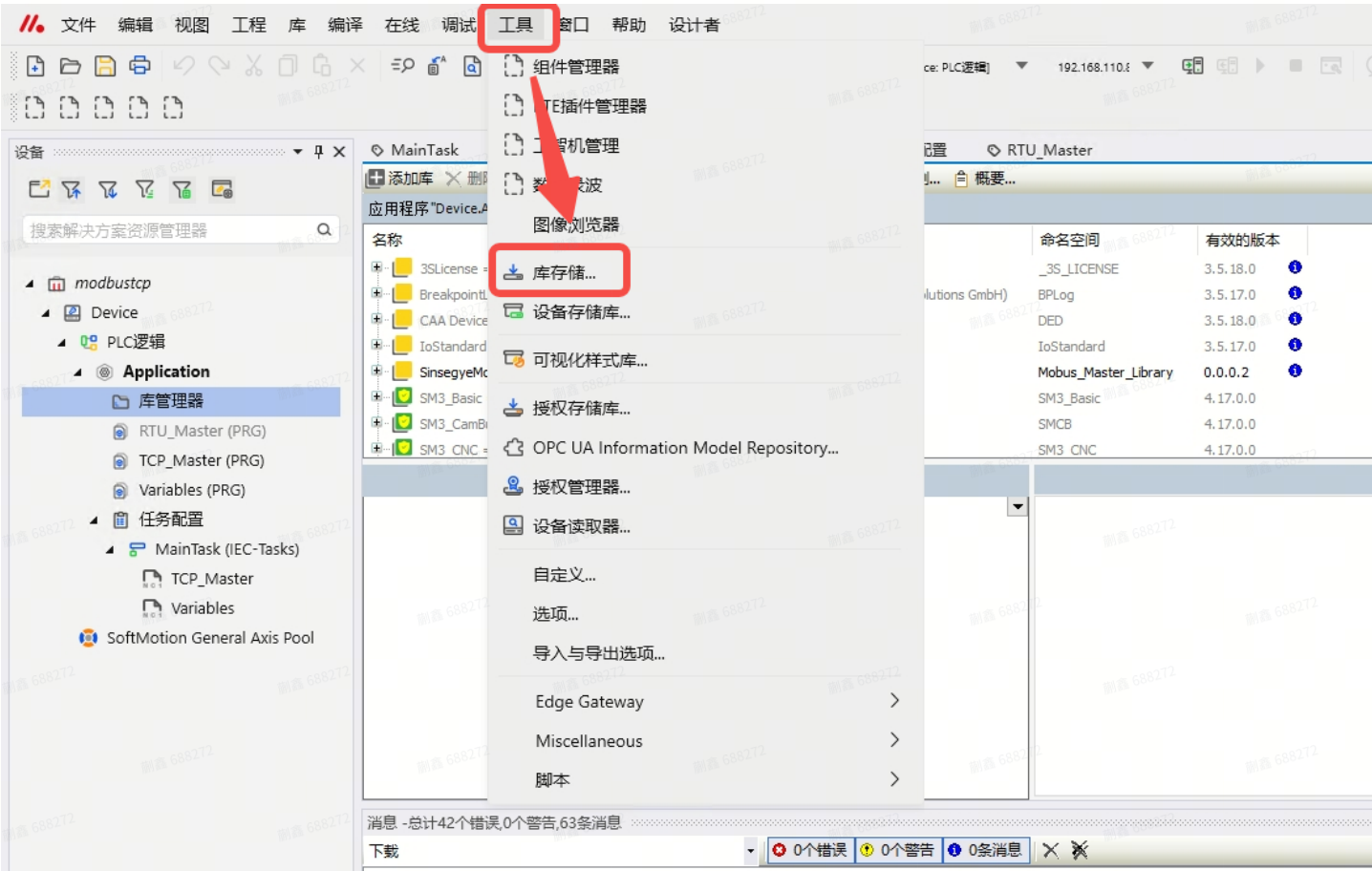
```
1 sudo vim /usr/local/etc/SinsegyeRTE/SinsegyeRTE.cfg
```

- 重启RTE服务

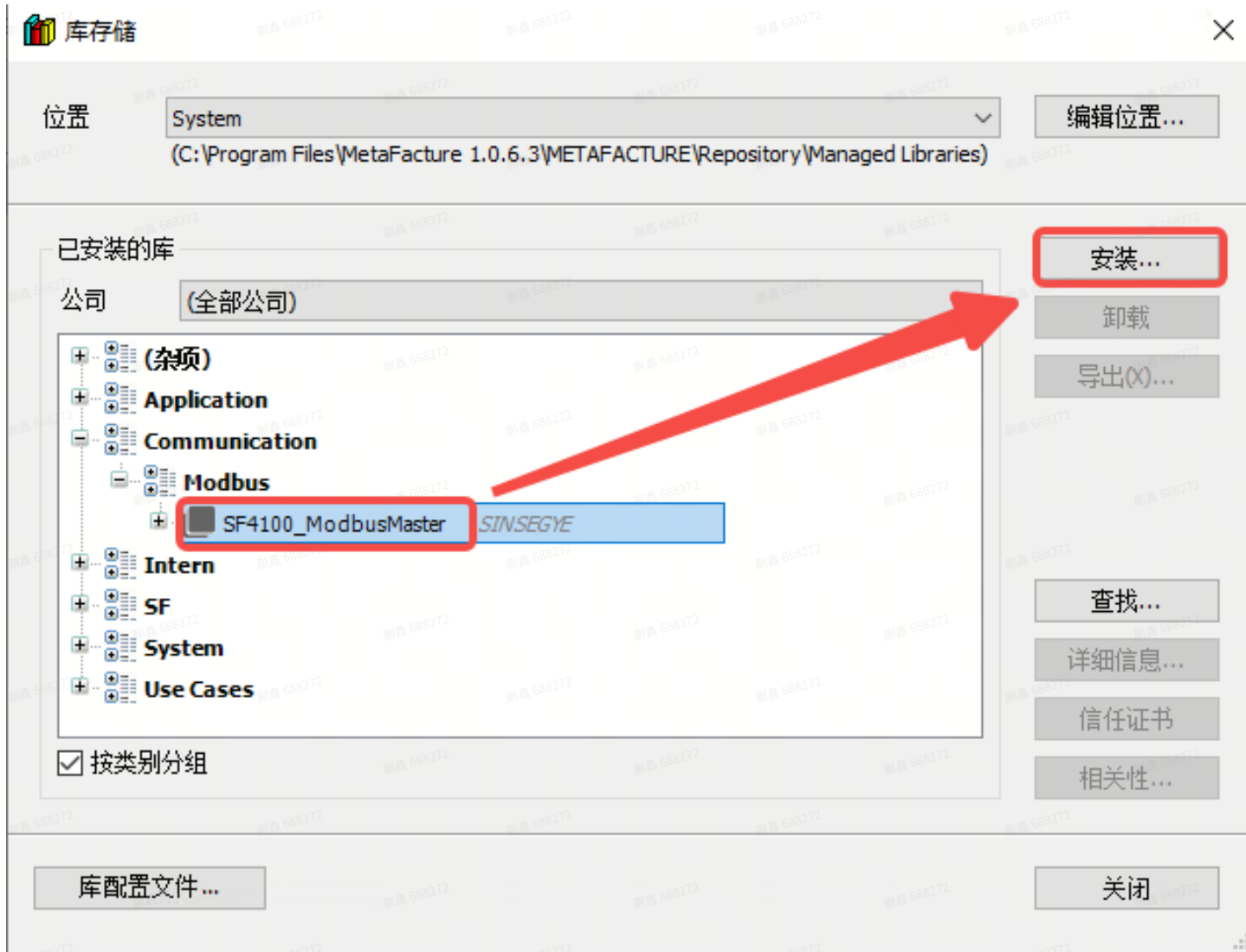
```
1 sudo systemctl restart sinsegyerte.service
```

2、卸载MetaFacture侧的Modbus Rtu master library

- MetaFacture界面点击“工具” -- “库存储”



- 对话框中选中安装的modbus master的库，点击“卸载”



技术说明

一、快速启动

(一) 本例软、硬件配置

硬件：

1. SX5100工智机 MetaOS V24.08.15_SX5
2. Win10 PC

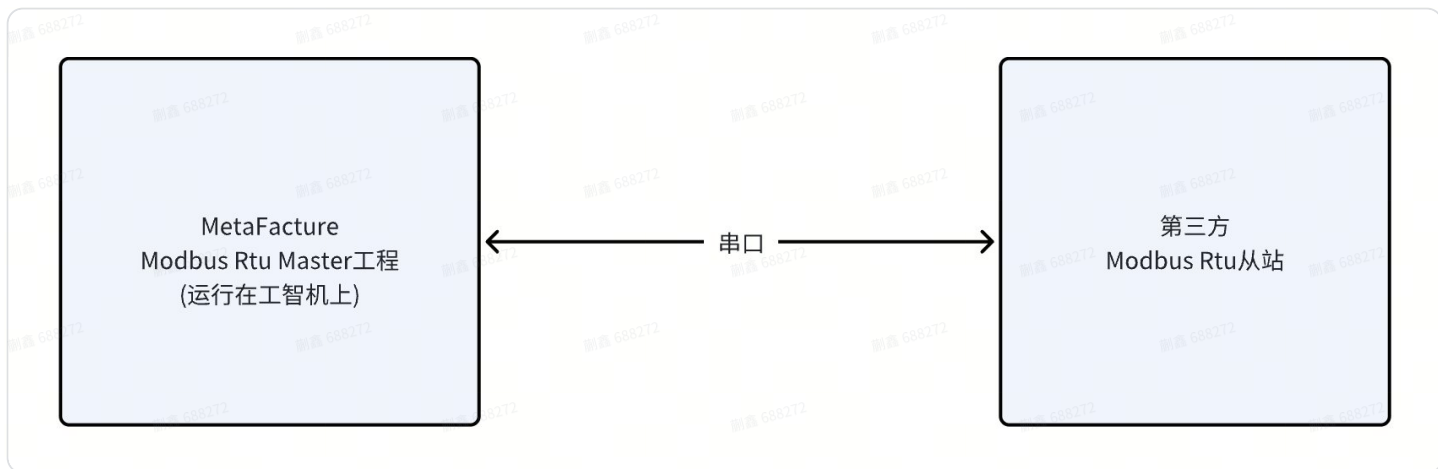
软件：

1. MetaFacutre V1.0.6.3
2. Modbus slave从站工具

本例实验要求及原理

- 实验要求：按照“安装卸载”部分中的“安装过程”配置完成Modbus Rtu Master环境

1、实验原理



- Metafacture Modbus Rtu站通过串口连接向第三方Modbus Rtu 从站发送请求，主要包含：功能码、目标寄存器的地址和数量、写操作时的数据；从站收到请求后先检查数据的合法性，然后执行请求的操作，响应主站请求；
- 上位机和工智机使用EtherNet连接；
- 上位机上，MetaFacture下装工程到工智机。工程中会包含下面的试验操作步骤中的配置；
- 工智机与第三方TCPIP客户端通过串口连接；

(二) 本例实验操作步骤

1、主站初始化串口实验步骤如下

- POU中声明区域调用modbus rtu master初始化串口功能块

```
1 Fb_serialInit : FB_MBRtuSerialPortInit;
```

- POU中程序区域调用Fb_serialInit

```
1 Fb_serialInit(Execute:=bTEST,  
2   PortName := '/dev/ttyS6',           //串口号  
3   BaudRate := 115200,                 //串口波特率  
4   DataBits := 8,  
5   StopBits := 1,  
6   Parity := 'N',  
7   Timeout:= ,  
8   BUSY=>,  
9   Error=>,  
10  ErrorId=>);
```

2、读多线圈的实验步骤如下：

- POU中声明区域调用modbus rtu master读多线圈功能块

```

1 bTEST          : BOOL ;
2 Fb_readCoils   : FB_MBRtuReadCoils;
3 arrReadCoils   : ARRAY [1..10] OF BOOL;

```

- POU中程序区域调用Fb_readcoils

```

1 Fb_readCoils(Execute:=bTEST,
2   Quantity:= 10,           //读取线圈长度
3   pMemoryAddr := ADR(arrReadCoils),
4   cbLength := SIZEOF(arrReadCoils),
5   MAddr:=0,               //读取线圈的起始位置
6   Timeout:= ,
7   UnitID:=1,
8   BUSY=>,
9   Error=>,
10  ErrorId=>);

```

- 工程运行后触发Fb_readCoils中的bExecute上升沿执行读取线圈的值

3、读取离散输入的实验步骤如下：

- POU中声明区域调用modbus rtu master读离散输入功能块

```

1 bTEST          : BOOL ;
2 Fb_readInputs   : FB_MBRtuReadInputs;
3 arrReadInputs   : ARRAY [1..10] OF BOOL;

```

- POU中程序区域调用Fb_readInputs

```

1 Fb_readInputs(Execute:=bTEST,
2   Quantity:=10,           //读取的离散输入长度
3   pMemoryAddr := ADR(arrReadInputs),
4   cbLength := SIZEOF(arrReadInputs),
5   MAddr:=0,               //读取离散输入起始位置
6   Timeout:= ,
7   UnitID:=1,
8   BUSY=>,
9   Error=>,
10  ErrorId=>);

```


- 工程运行后触发Fb_readInputs中的bExecute上升沿执行读取离散输入的值；

3、读取保持寄存器的实验步骤如下：

- POU中声明区域调用modbus rtu master读保持寄存器功能块

```
1 bTEST                : BOOL ;
2 Fb_readRegs           : FB_MBRtuReadRegs;
3 arrReadRegs           : ARRAY [1..10] OF WORD;
```

- POU中程序区域调用Fb_readRegs

```
1 Fb_readRegs(Execute:=bTEST,
2   Quantity:= 10,                //读取的保持寄存器长度
3   pMemoryAddr := ADR(arrReadRegs),
4   cbLength := SIZEOF(arrReadRegs),
5   MAddr:=0,                    //读取的保持寄存器起始位置
6   Timeout:= ,
7   UnitID:=1,
8   BUSY=>,
9   Error=>,
10  ErrorId=>);
```

- 工程运行后触发Fb_readRegs中的bExecute上升沿执行读取保持寄存器的值；

4、读取输入寄存器的实验步骤如下：

- POU中声明区域调用modbus rtu master读输入寄存器功能块

```
1 bTEST                : BOOL ;
2 Fb_readInputRegs      : FB_MBRtuReadInputRegs;
3 arrReadInputRegs      : ARRAY [1..10] OF WORD;
```

- POU中程序区域调用Fb_readInputRegs

```
1 Fb_readInputRegs(Execute:=bTEST,
2   Quantity:= 10,                //读取的保持寄存器长度
3   pMemoryAddr := ADR(arrReadInputRegs),
4   cbLength := SIZEOF(arrReadInputRegs),
5   MAddr:=0,                    //读取的保持寄存器起始位置
6   Timeout:= ,
7   UnitID:=1,
```

```

8     BUSY=>,
9     Error=>,
10    ErrorId=>);

```

- 工程运行后触发Fb_readInputRegs中的bExecute上升沿执行读取输入寄存器的值；

5、写多线圈的实验步骤如下：

- POU中声明区域调用modbus rtu master写多线圈功能块

```

1  bTEST          : BOOL ;
2  Fb_writeCoils   : FB_MBRTuWriteCoils;
3  arrWriteCoils   : ARRAY [1..10] OF BOOL;

```

- POU中程序区域调用Fb_writeCoils

```

1  Fb_writeCoils(Execute:=bTEST,
2      Quantity:= 10,           //要写入线圈的长度
3      pMemoryAddr := ADR(arrWriteCoils),
4      cbLength := SIZEOF(arrWriteCoils),
5      MAddr:=10,              //要写入的线圈的起始位置
6      Timeout:= ,
7      UnitID:=1,              //从站的ID
8      BUSY=>,
9      Error=>,
10     ErrorId=>);

```

- 工程运行后触发Fb_writecoils中的bExecute上升沿执行写多线圈的值；

6、写多保持寄存器的实验步骤如下：

- POU中声明区域调用modbus rtu master写多保持寄存器功能块

```

1  bTEST          : BOOL ;
2  Fb_writeRegs    : FB_MBRTuWriteRegs;
3  arrWriteRegs    : ARRAY [1..10] OF WORD;

```

- POU中程序区域调用Fb_writeRegs

```

1  Fb_writeRegs(Execute:=bTEST,
2      Quantity:= 10,           //要写入的保持寄存器长度

```

```

3    pMemoryAddr := ADR(arrWriteRegs),
4    cbLength := sizeof(arrWriteRegs),
5    MAddr:=10, //要写入的保持寄存器的起始位置
6    Timeout:= ,
7    UnitID:=1,
8    BUSY=>,
9    Error=>,
10   ErrorId=>);

```

- 工程运行后触发Fb_writeRegs中的bExecute上升沿执行写多保持寄存器的值；

7、写单线圈的实验步骤如下：

- POU中声明区域调用modbus rtu master写单个线圈功能块

```

1 bTEST          : BOOL ;
2 Fb_writeSingleCoil : FB_MBRtuWriteSingleCoil;
3 writeSinglecoil   : BOOL:=True;

```

- POU中程序区域调用Fb_writeSingleCoil

```

1 Fb_writeSingleCoil(Execute:=bTEST,
2   Value := writeSinglecoil,
3   MAddr:=20, //写入单线圈的位置
4   Timeout:= ,
5   UnitID:=1,
6   Busy=>,
7   Error=>,
8   ErrorId=>);

```

- 工程运行后触发Fb_writeSingleCoil中的bExecute上升沿执行写单个线圈；

8、写单个保持寄存器的实验步骤如下：

- POU中声明区域调用modbus rtu master写单个保持寄存器功能块

```

1 bTEST          : BOOL ;
2 Fb_writeSingleReg : MBRtuWriteSingleReg;
3 writeSingleReg    : WORD:=20;

```

- POU中程序区域调用Fb_writeSingleReg

```

1 Fb_writeSingleReg(Execute:=bTEST,
2   Value := writeSingleReg,
3   MAddr:=20,           //写入保持寄存器的位置
4   Timeout:= ,
5   UnitID:=1,
6   Busy=>,
7   Error=>,
8   ErrorId=>);

```

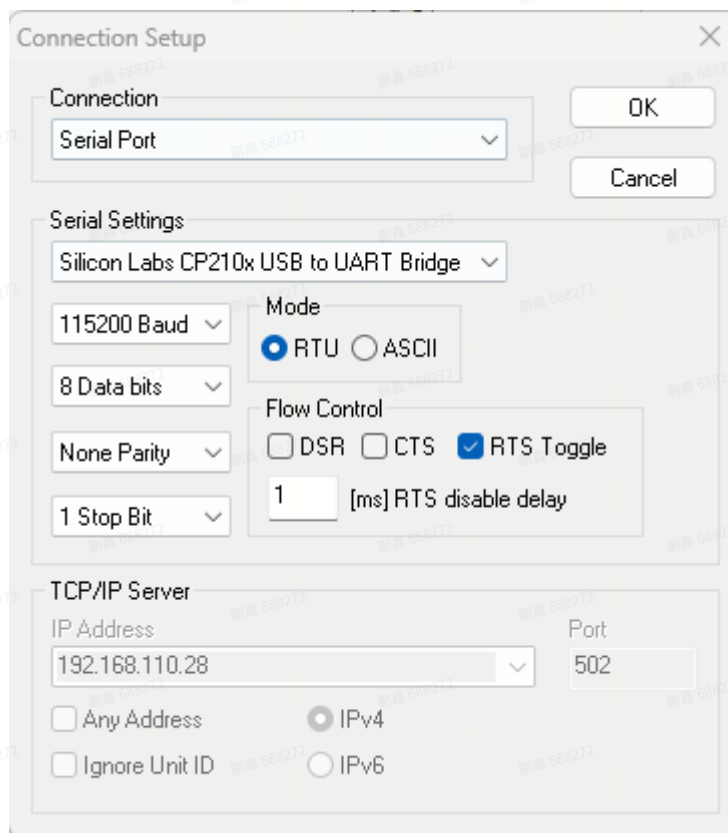
- 工程运行后触发Fb_writeSingleReg中的bExecute上升沿执行写单个保持寄存器；

(三) 实验注意点

1、实验中用到的从站COM口以实际为准；

2、实验中使用的从站案例如下：

- Modbus Slave中的从站COM口及串口波特率等配置以实际为准：



- 线圈配置21位

Mbslave1

ID = 1: F = 01

	Alias	00000	Alias	00010	Alias	00020
0		0		0		0
1		0		0		
2		0		0		
3		0		0		
4		0		0		
5		0		0		
6		0		0		
7		0		0		
8		0		0		
9		0		0		

- 离散输入配置10位



ID = 1: F = 02

	Alias	00000
0		0
1		0
2		0
3		0
4		0
5		0
6		0
7		0
8		0
9		0

- 保持寄存器配置21位

ID = 1: F = 03

	Alias	00000	Alias	00010	Alias	00020
0		0		0		0
1		0		0		
2		0		0		
3		0		0		
4		0		0		
5		0		0		
6		0		0		
7		0		0		
8		0		0		
9		0		0		

- 输入寄存器配置10位



Mbslave4

ID = 1: F = 04

	Alias	00000
0		0
1		0
2		0
3		0
4		0
5		0
6		0
7		0
8		0
9		0

功能介绍

一、主站配置初始化串口

(一) 功能块FB_MBRtuSerialPortInit介绍



(二) 参数介绍

• 输入参数

参数名称	参数类型	描述
PortName	STRING	串口号
BaudRate	UINT	波特率设置，常用的波特率类型如9600、38400、115200均支持
DataBits	UINT	数据位
StopBits	UINT	停止位
Parity	STRING	校验位
Execute	BOOL	触发执行
Timeout	TIME	超时

• 输出参数

参数名称	参数类型	描述
BUSY	BOOL	功能块激活时设置，直到确认接收
Error	BOOL	如果命令在传输过程中出现错误，则设置true直到bBusy输出被重置
ErrorId	UINT	当bError输出被设置，提供错误号

二、主站配置读取从站多线圈

(一) 功能块FB_MBRtuReadCoils介绍



(二) 参数介绍

• 输入参数

参数名称	参数类型	描述
UnitID	UINT	从站的ID
Quantity	UINT	读取的线圈位数
MBAAddr	WORD	读取的线圈起始位置
cbLength	UINT	存储读取的字节大小
pMemoryAddr	POINTER TO BYTE	要读的存放数据的存放地址
Execute	BOOL	触发读取动作
tTimeout	Time	超时

• 输出参数

参数名称	参数类型	描述
BUSY	BOOL	功能块激活时设置，直到确认接收
Error	BOOL	如果命令在传输过程中出现错误，则设置true直到bBusy输出被重置
ErrorId	UINT	当bError输出被设置，提供错误号

三、主站配置读取从站输入寄存器

(一) 功能块FB_MBRtuReadInputRegs介绍



(二) 参数介绍

• 输入参数

参数名称	参数类型	描述
UnitID	UINT	从站的ID
Quantity	UINT	读取的输入寄存器位数，一次最多读取125位
MBAAddr	UINT	读取的输入寄存器起始位置
cbLength	UINT	存储读取的字节大小
pMemoryAddr	POINTER TO BYTE	要读的存放数据的存放地址
Execute	BOOL	触发读取动作
tTimeout	Time	超时

• 输出参数

参数名称	参数类型	描述
BUSY	BOOL	功能块激活时设置，直到确认接收
Error	BOOL	如果命令在传输过程中出现错误，则设置true直到bBusy输出被重置
ErrorId	UINT	当bError输出被设置，提供错误号

四、主站配置读取从站离散输入

(一) 功能块FB_MBRtuReadInputs介绍



(二) 参数介绍

• 输入参数

参数名称	参数类型	描述
UnitID	UINT	从站的ID

Quantity	UINT	读取的离散输入位数
MBAAddr	UINT	读取的离散输入起始位置
cbLength	UINT	存储读取的字节大小
pMemoryAddr	POINTER TO BYTE	要读的存放数据的存放地址
Execute	BOOL	触发读取动作
tTimeout	Time	超时

• 输出参数

参数名称	参数类型	描述
BUSY	BOOL	功能块激活时设置，直到确认接收
Error	BOOL	如果命令在传输过程中出现错误，则设置true直到bBusy输出被重置
ErrorId	UINT	当bError输出被设置，提供错误号

五、主站配置读取从站保持寄存器

(一) 功能块FB_MBRtuReadRegs介绍



(二) 参数介绍

• 输入参数

参数名称	参数类型	描述
UnitID	UINT	从站的ID
Quantity	UINT	读取的保持寄存器位数，一次最多读取125位
MBAAddr	UINT	读取的保持寄存器起始位置

cbLength	UINT	存储读取的字节大小
pMemoryAddr	POINTER TO BYTE	要读的存放数据的存放地址
Execute	BOOL	触发读取动作
tTimeout	Time	超时

• 输出参数

参数名称	参数类型	描述
BUSY	BOOL	功能块激活时设置，直到确认接收
Error	BOOL	如果命令在传输过程中出现错误，则设置true直到bBusy输出被重置
ErrorId	UINT	当bError输出被设置，提供错误号

六、主站配置写从站多线圈

(一) 功能块FB_MBRtuWriteCoils介绍



(二) 参数介绍

• 输入参数

参数名称	参数类型	描述
UnitID	UINT	从站的ID
Quantity	WORD	要写入的线圈位数
MAddr	WORD	要写入的线圈起始位置
cbLength	UINT	存储写入的字节大小
pMemoryAddr	POINTER TO BYTE	要写入的数据的地址

Execute	BOOL	触发写入动作
tTimeout	Time	超时

• 输出参数

参数名称	参数类型	描述
BUSY	BOOL	功能块激活时设置，直到确认接收
Error	BOOL	如果命令在传输过程中出现错误，则设置true直到bBusy输出被重置
ErrorId	UINT	当bError输出被设置，提供错误号

七、主站配置写从站多保持寄存器

(一) 功能块FB_MBRtuWriteRegs介绍



(二) 参数介绍

• 输入参数

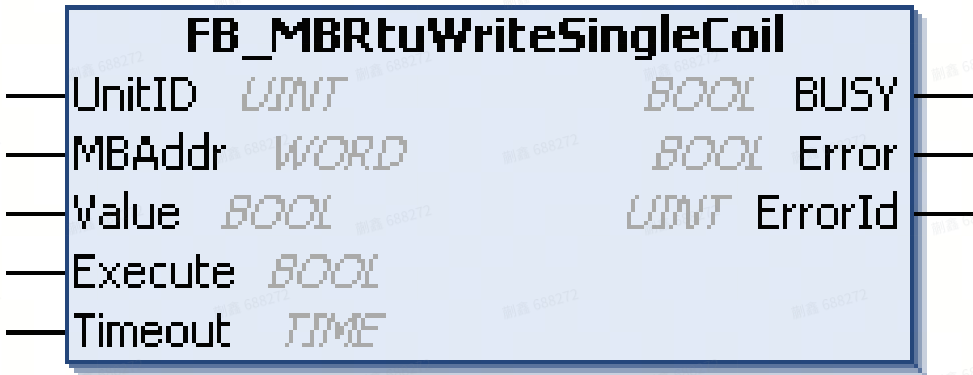
参数名称	参数类型	描述
UnitID	UINT	从站的ID
Quantity	UINT	要写入的保持寄存器位数
MBAddr	UINT	要写入的保持寄存器的起始位置
pMemoryAddr	POINTER TO WORD	要写入的数据的地址
Execute	BOOL	触发写入动作
tTimeout	Time	超时

● 输出参数

参数名称	参数类型	描述
BUSY	BOOL	功能块激活时设置，直到确认接收
Error	BOOL	如果命令在传输过程中出现错误，则设置true直到bBusy输出被重置
ErrorId	UINT	当bError输出被设置，提供错误号

八、主站配置写从站单线圈

(一) 功能块FB_MBRtuWriteSingleCoil介绍



(二) 参数介绍

● 输入参数

参数名称	参数类型	描述
UnitID	UINT	从站的ID
MBAAddr	WORD	要写入的线圈位置
Value	BOOL	要写入的线圈值
Execute	BOOL	触发写入动作
tTimeout	Time	超时

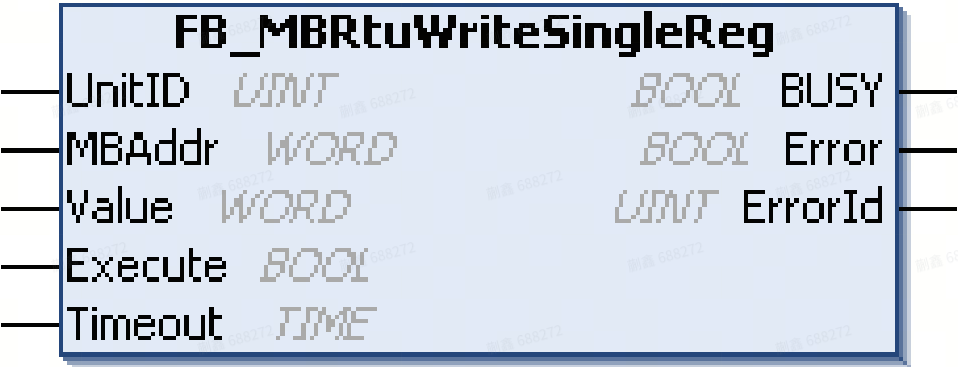
● 输出参数

参数名称	参数类型	描述
------	------	----

BUSY	BOOL	功能块激活时设置，直到确认接收
Error	BOOL	如果命令在传输过程中出现错误，则设置true直到bBusy输出被重置
ErrorId	UINT	当bError输出被设置，提供错误号

九、主站配置写从站单保持寄存器

(一) 功能块FB_MBRtuWriteSingleReg介绍



(二) 参数介绍

- 输入参数

参数名称	参数类型	描述
UnitID	UINT	从站的ID
MAddr	WORD	要写入的保持寄存器位置
Value	WORD	要写入的保持寄存器值
Execute	BOOL	触发写入动作
tTimeout	Time	超时

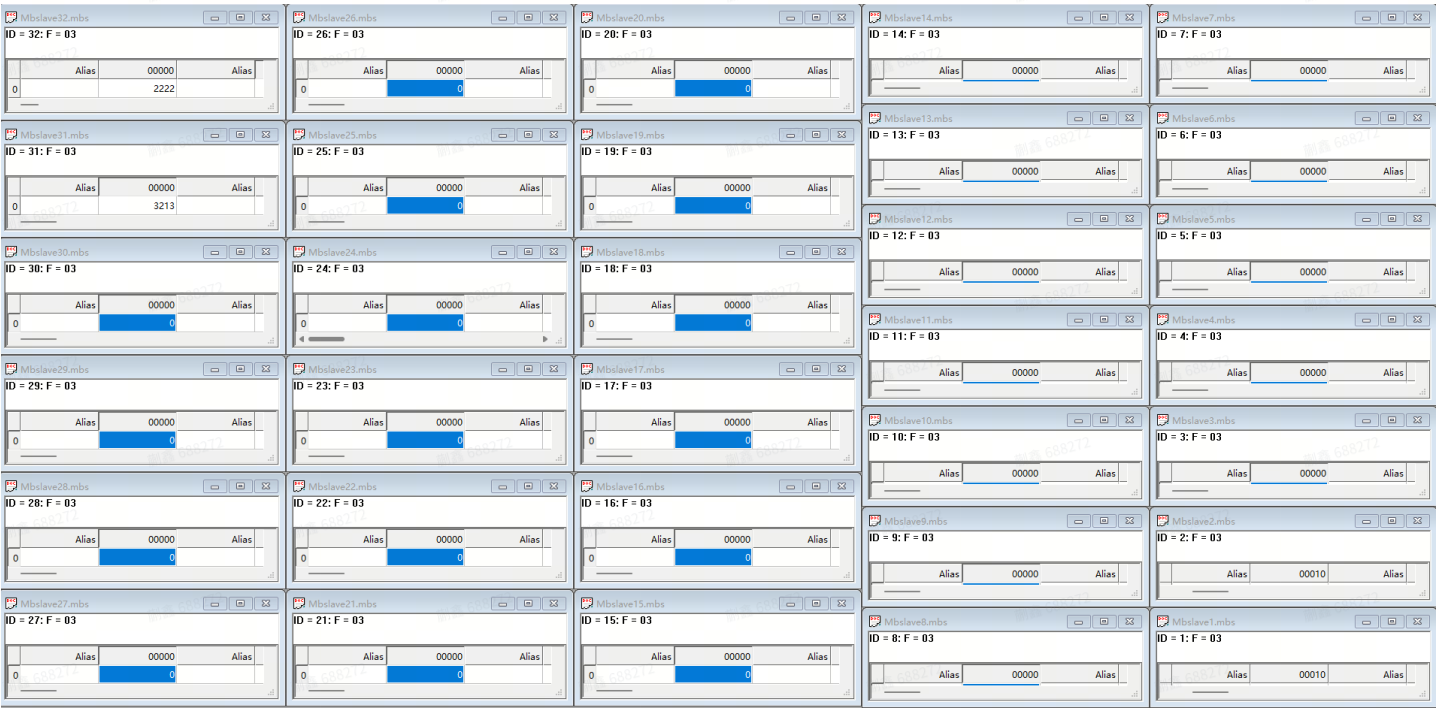
- 输出参数

参数名称	参数类型	描述
BUSY	BOOL	功能块激活时设置，直到确认接收
Error	BOOL	如果命令在传输过程中出现错误，则设置true直到bBusy输出被重置
ErrorId	UINT	当bError输出被设置，提供错误号

ErrorId	UINT	当bError输出被设置，提供错误号
---------	------	--------------------

十、主站配置连接32从站

- 使用模拟软件配置32个RTU从站（各个从站ID配置不一样），参考如下：
















- 主站工程中配置连接32个从站进行读写操作，测试读写正常

附录

一、错误处理

- 如果串口不通或者主站请求的从站地址不存在，工程运行后功能块的Error会置True

 Fb_readCoils	FB_MBRtuReadCoils	
 UnitID	UINT	1
 Quantity	UINT	200
 MAddr	WORD	0
 cbLength	UINT	200
  pMemoryAddr	POINTER TO BYTE	16#00007F166008...
 Execute	BOOL	TRUE
 Timeout	TIME	T#0ms
 BUSY	BOOL	FALSE
 Error	BOOL	TRUE
 ErrorId	UINT	0
 bOldExecute	BOOL	TRUE

- 需要先确定串口通讯正常，然后确定主要要请求的从站地址都存在；

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