



Q 系列可编程逻辑控制器

Q1简明教程

前言

Q1控制器是以CODESYS 为编程调试平台的多功能中型控制器，本文档会介绍如何在CODESYS平台测试Q1，其中包括如何使用Q1本地高速输入输出、Q系列耦合器和远程扩展IO（目前暂时介绍数字量和模拟量输入输出），单轴和两轴运动控制。由于Q1控制器会进行软硬件更新，文档也会不定期更新。本教程需要的文档也会在川禾自动化学院供用户免费获取。

川禾自动化学院网站地址：川禾自动化学院 (hcfa.cn)

文档版本更新

更新日期	版本	更新内容	更新人员
2018.11.19	V1.0		
2018.11.19	V1.01	更新附录A关于描述文件安装部分	
2018.12.18	V1.1	更新接线示意；更改追踪为Trace	
2019.07.01	V2.0	Q1更新Linux平台，更新高速IO和远程扩展模块使用	
2020.04.07	V3.0	Q1更新3.0版本，包括本地扩展，高速输入输出库，通讯库等	
2022.05.	V4.0	重新	周墨馨

免责声明

我们对文档内容都进行了测试与检查，但可能仍有些差错，请您谅解。如果您对本文档有个人的意见或建议，欢迎发送邮件联系作者：400@hcfa.cn。

浙江禾川科技股份有限公司

电话：0570- 7117888

地址：浙江省龙游县工业园阜财路9号

地址：杭州市余杭区五常街道文一西路1001号D幢4楼

杭州研发中心

技术支持热线：400 126 969

技术支持邮箱：400@hcfa.cn

目录

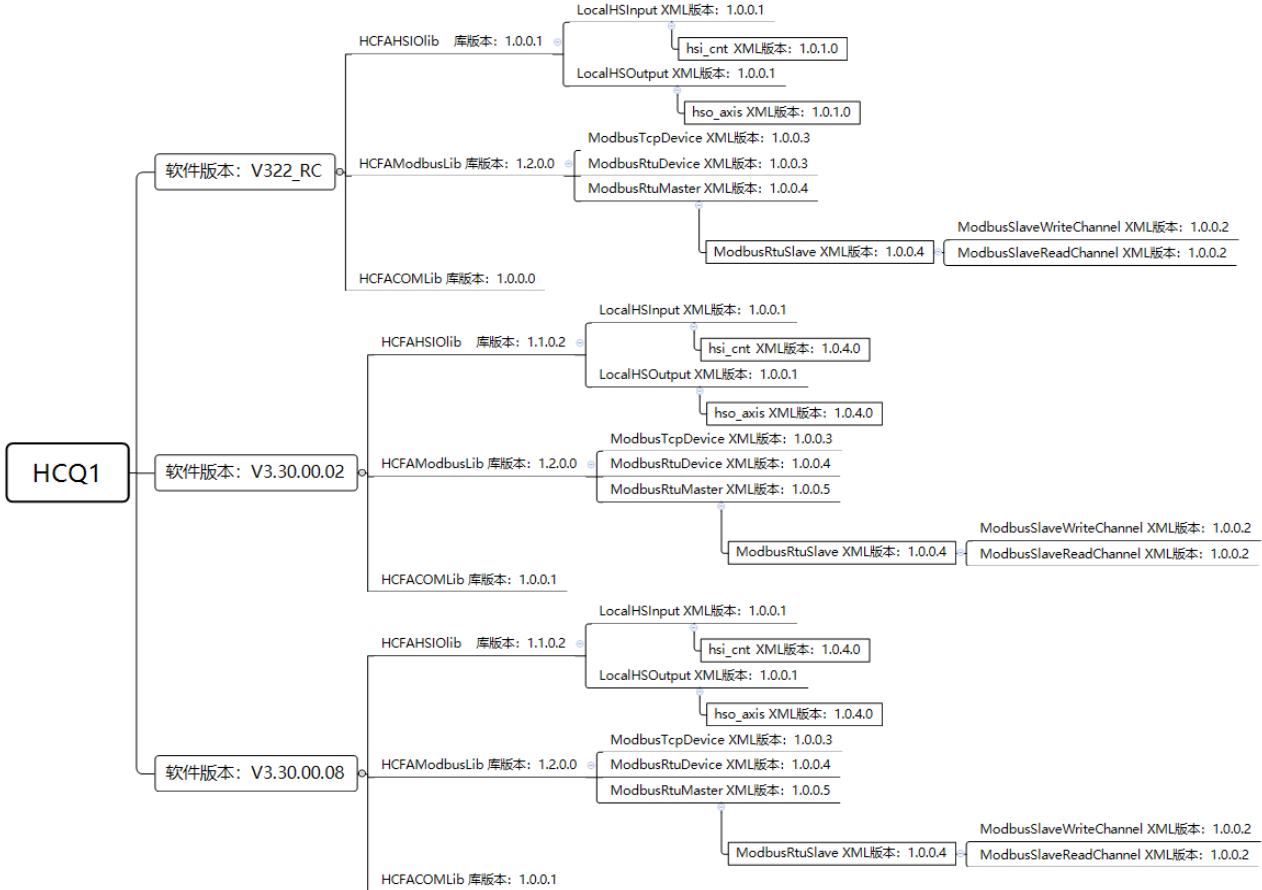
第一章	概述.....	1
1.1	版本信息.....	1
1.2	测试平台搭建.....	1
1.2.1	示例文件.....	1
1.2.2	硬件测试平台.....	2
第二章	工程创建.....	3
2.1	新建 CODESYS 工程项目.....	3
2.1.1	工程创建.....	3
2.1.2	语言设置.....	4
2.1.3	变量中文设置.....	4
2.1.4	快速索引设置.....	5
第三章	通讯.....	6
3.1	PC 与 Q1 建立通讯.....	6
3.1.1	与 PC 通讯.....	6
3.1.2	IP 地址修改.....	7
3.2	Modbus RTU 通讯.....	7
3.2.1	外部硬件介绍.....	7
3.2.2	Q1 做主站.....	8
3.2.3	Q1 做从站.....	10
3.3	Modbus TCP 通讯.....	11
3.3.1	Q1 做主站.....	11
3.3.2	Q1 做从站.....	14
3.4	OPCUA 通讯.....	15
3.4.1	程序编写及配置.....	15
3.4.2	配置 Codesys Security Agent package.....	17
第四章	程序运行.....	18
4.1	任务配置设置.....	18
4.1.1	任务配置.....	18
4.1.2	任务.....	19
4.2	程序 (PRG).....	20
4.2.1	POU 的程序 (PRG) 添加.....	20
4.2.2	程序运行.....	22
4.2.3	Trace.....	24
4.2.4	在线监控.....	26
4.2.5	软件复位.....	27
第五章	Q1 本体 IO.....	29
5.1	Q1 自带普通输入输出.....	29
5.1.1	Q1 硬件接线.....	30
5.1.2	程序示例.....	31
5.2	Q1 自带高速输入输出.....	34

5.2.1	Q1 硬件接线.....	35
5.2.2	程序示例.....	35
第六章	扩展模块.....	37
6.1	硬件接线介绍.....	37
6.1.1	数字量模块硬件接线.....	38
6.1.2	模拟量模块硬件接线.....	39
6.2	程序示例.....	40
6.2.1	EtherCAT 主站.....	40
6.2.2	数字量模块程序示例.....	42
6.2.1	模拟量模块程序示例.....	44
第七章	运动控制.....	46
7.1	运动控制配置.....	46
7.2	运动控制程序示例.....	49
7.2.1	单轴程序示例.....	49
7.2.2	多轴程序示例.....	54
附录	59
描述文件安装.....	59
Modbus RTU 数据报文结构.....	59
Modbus TCP 数据报文结构.....	61
AD04 对象字典总表.....	63
DA04 对象字典总表.....	66
模拟量模块模数值转换及显示表.....	69
轴状态转移图.....	70

第一章 概述

1.1 版本信息

软硬件版本对应安装包/描述文件下载



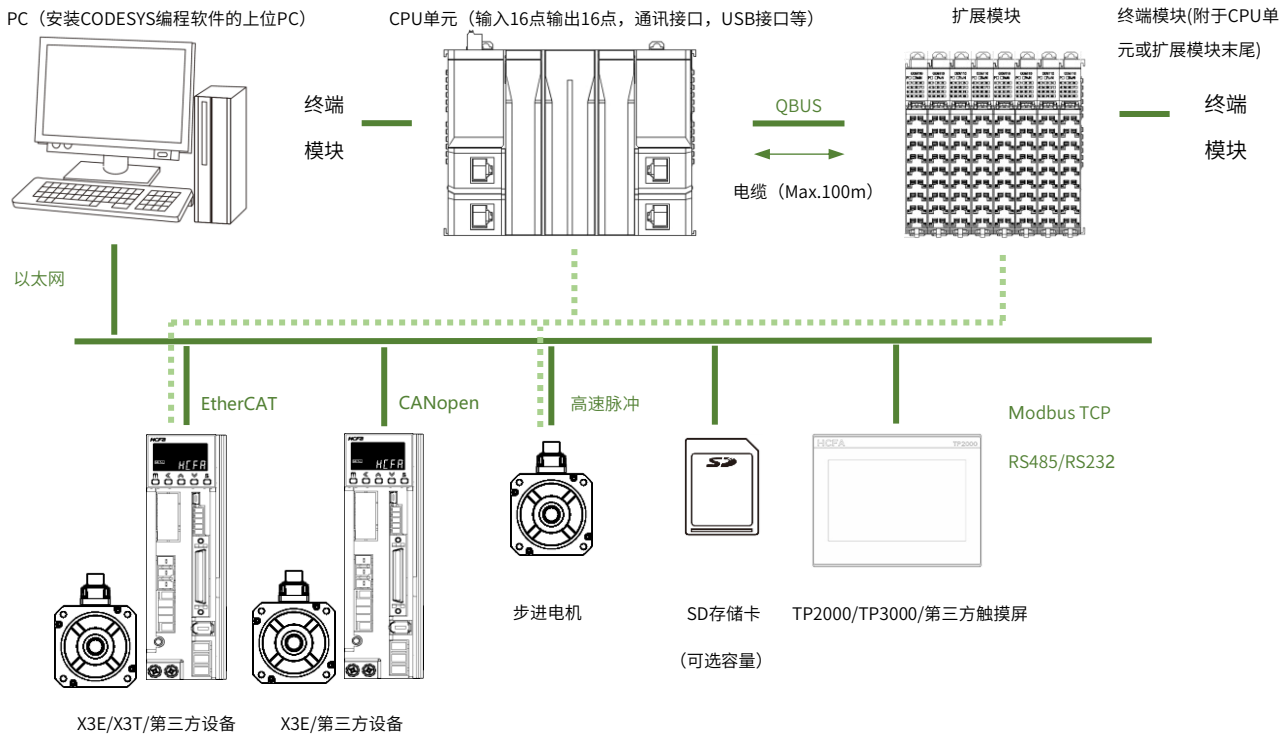
1.2 测试平台搭建

1.2.1 示例文件

文件名称	文件说明
Q1 简明调试教程.pdf	操作步骤说明
Q1test.pro	测试程序
HCQ1-1300-D.package	Q1 所需设备描述文件(对应版本下载)
HCFA_X3E_ESI.xml	X3E 描述文件
运动控制手册.pdf	运动控制功能块使用说明
Q 系列 Modbus RTU 通讯功能介绍.pdf	Modbus RTU 通讯使用教程
Q 系列 Modbus TCP 通讯功能介绍.pdf	Modbus TCP 通讯使用教程
Q1 高速输入输出使用介绍-Q13.30	高速输入输出使用教程

1.2.2 硬件测试平台

测试平台由Q1控制器、EC耦合器（HCQX-EC）、数字量输入输出（HCQX-ID/OD16-D2）、模拟量输入输出（HCQX-AD/DA04-D2）、X3E（SV-X3EA010A-A2-EC）驱动器和X3（SV-X3MM010A-N2LA）电机组成，硬件接线图示例如下：



注： 1. 基础CPU单元默认配置16通道普通输入输出，8通道高速输入输出，4个标准RJ45口，其中2个网口支持作为EtherCAT主站使用，其他额外的功能需要用户选配（定制型号除外）；

2. 扩展模块工作供电由EC耦合器提供（EC耦合器模块需要使用定制网线，随EC耦合器赠送定制网线及转接头），目前只支持挂载16个从站模块（实际还是根据功率计算），用户需要挂载更多的从站需要添加EC耦合器。Q1控制器内置EC耦合模块，禾川的远程扩展模块直接插在右端，通过金属片通讯；

3. 从站模块末端需要配合终端模块以保护从站模块/CPU单元金属片不受外界环境影响，否则最终导致设备无法正常工作甚至造成模块损坏。

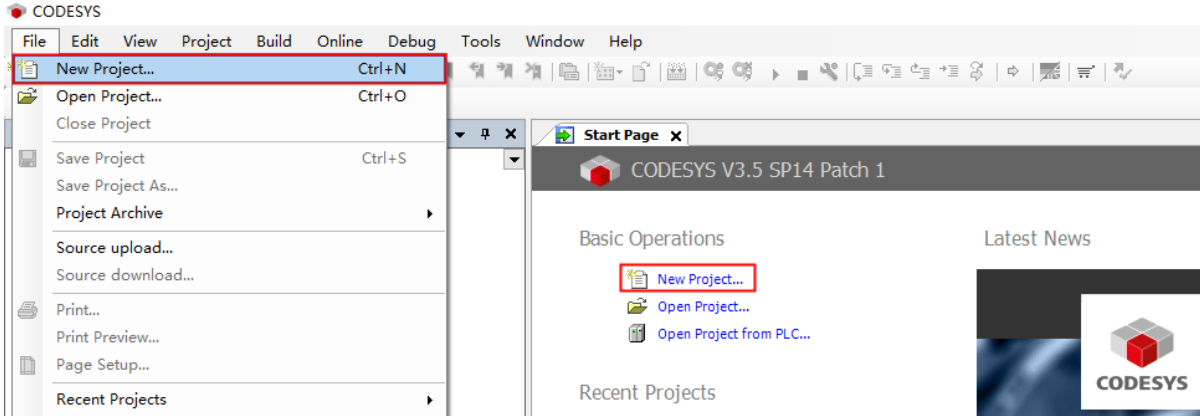
第二章 工程创建

2.1 新建CODESYS工程项目

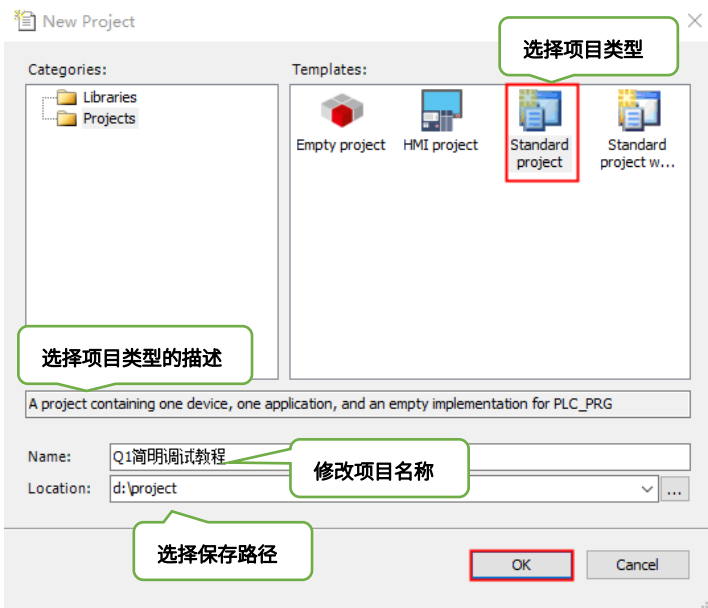
2.1.1 工程创建

在PC端安装好CODESYS软件后，双击打开软件。

新建工程可以选择【File】→【New Project】，也可以直接选择【New Project】两种方式



用户可以选择需要的项目类型，命名项目工程以及存储路径，完成后选择【OK】



选择连接设备【HCQ1-1300-D】（在此之前PC已经安装Q1安装包），选择编程语言，教程示例以【ST语言】为例



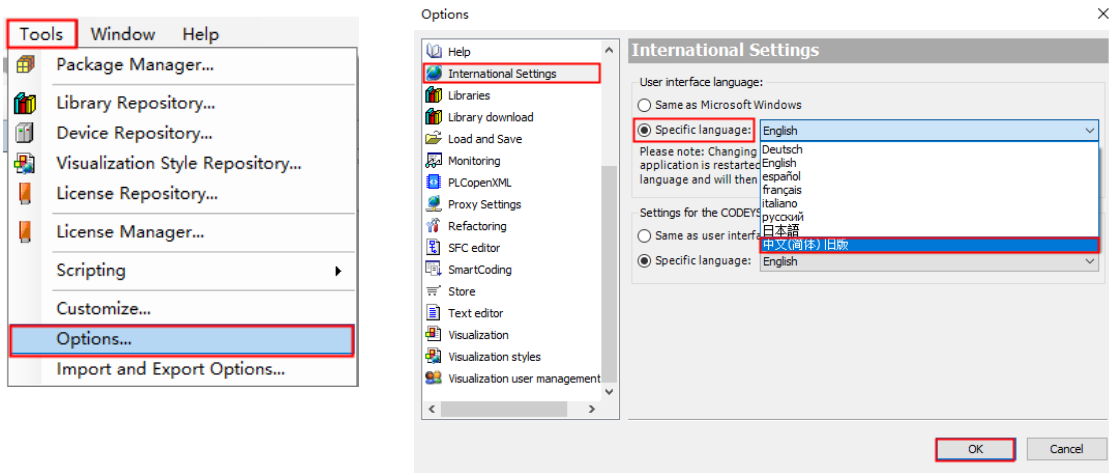
一个标准的工程如下图所示，工程左侧会生成设备树，自动添加设备、Application、PRG（程序）、Main Task任务，（若添加的标准工程打开后没有设备树，可在菜单栏选择【视图】→【设备】）



2.1.2 语言设置

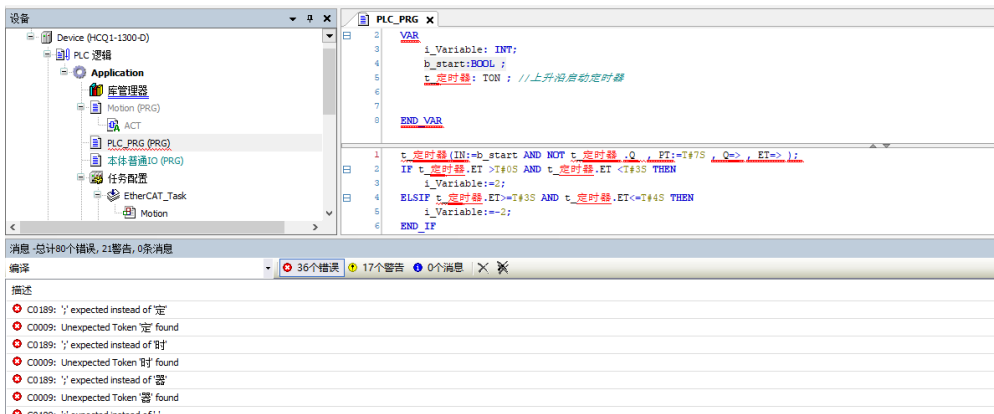
首次使用CODESYS为英文模式，切换中文模式如下。

单击【Tool】→【Options】→【International Settings】→【Specific language】，选择【中文（简体）旧版】，选择【OK】，关闭工程，再次打开则为中文模式，之后创建的工程都为中文模式。

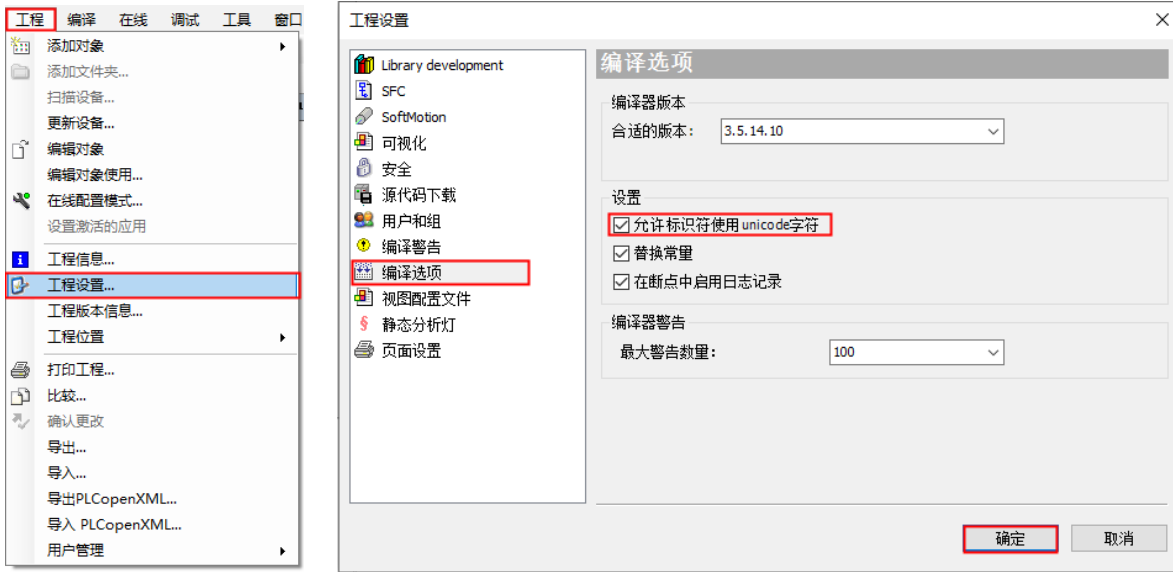


2.1.3 变量中文设置

CODESYS编程时，若想使用中文变量名称，需要以下设置，否则会标号报错

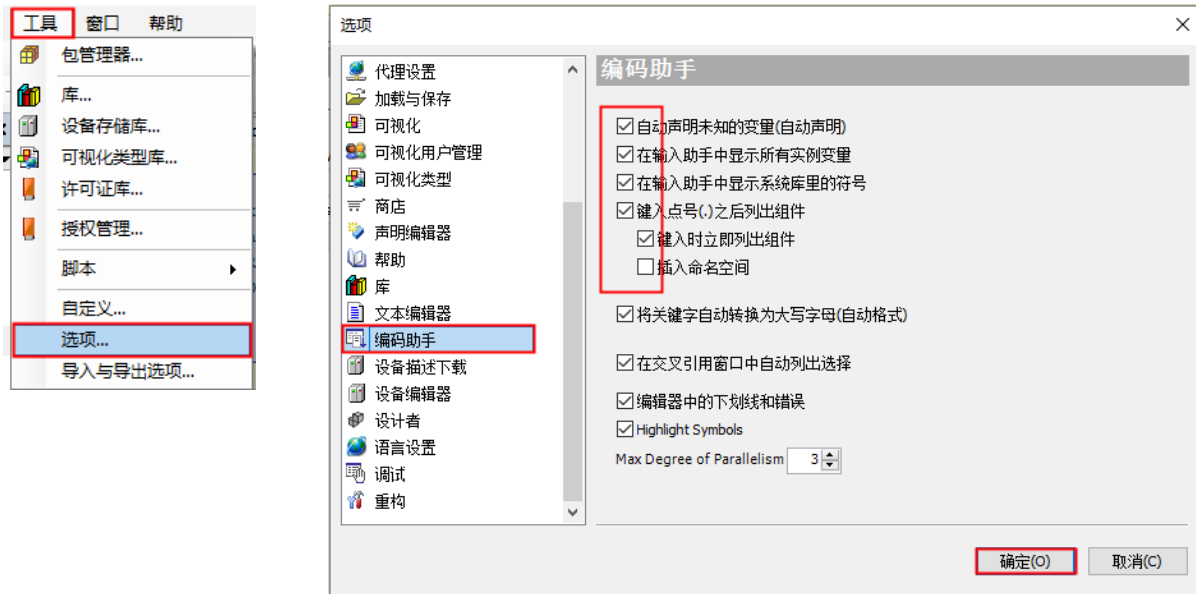


选择【工程】→【工程设置】→【编译选项】，勾选【允许标识符使用Unicode字符】后，选择【确定】



2.1.4 快速索引设置

菜单栏中选择【工具】→【选项】，找到【编码助手】，如下图勾选上后，【确定】



在程序中可索引调用，例如声明调用定时器功能块TON

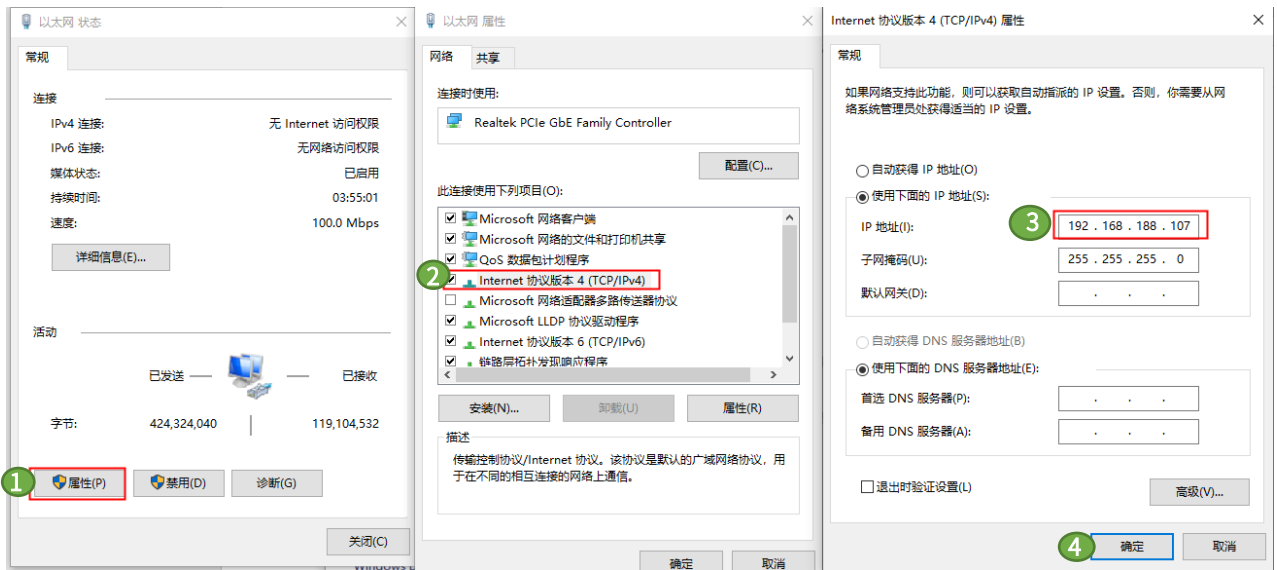


3.1 PC与Q1建立通讯

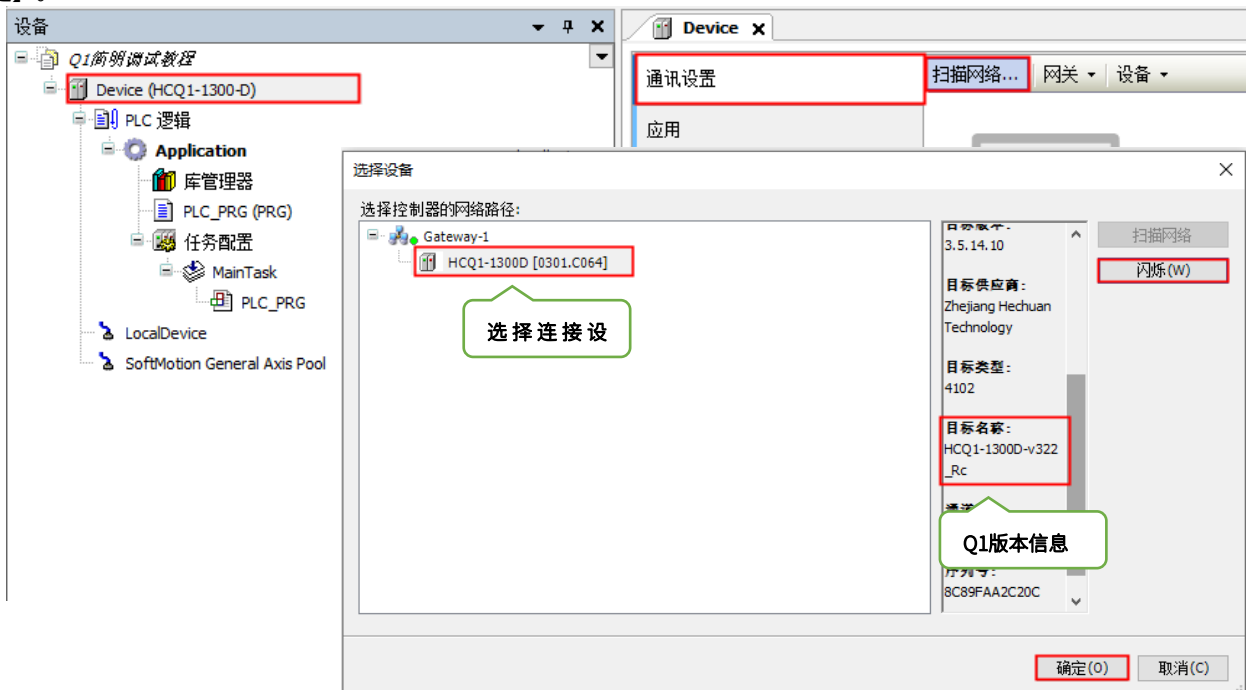
3.1.1 与PC通讯

Q1的Port1默认IP地址为**192.168.188.100**，Port2的默认IP地址为**192.168.88.100**，本次实验与上位机连接的Port1口，需打开以太网设置。

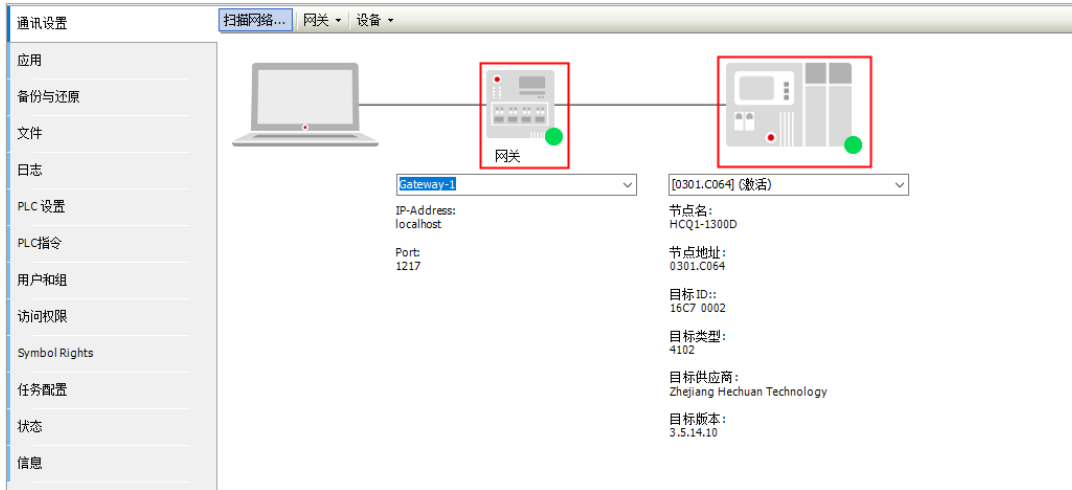
点击【属性】→【Internet 协议版本4 (TCP/IPv4)】→ 修改上位机IP地址，使其与Q1的Port1的IP地址在同一网段（此处设置的IP地址不可与Q1 Port口的IP地址完全一致），最后点击【确定】。



修改完成后，回到CODESYS，双击【Device】→【通讯设置】→【扫描网络】，单击设备，右侧可查看Q1版本信息（可根据版本下载对应的安装包或者描述文件），点击【闪烁】，查看PLC面板为4个0，则连接正确。设备连接正确，选择【确定】。



通讯成功，绿灯亮起。

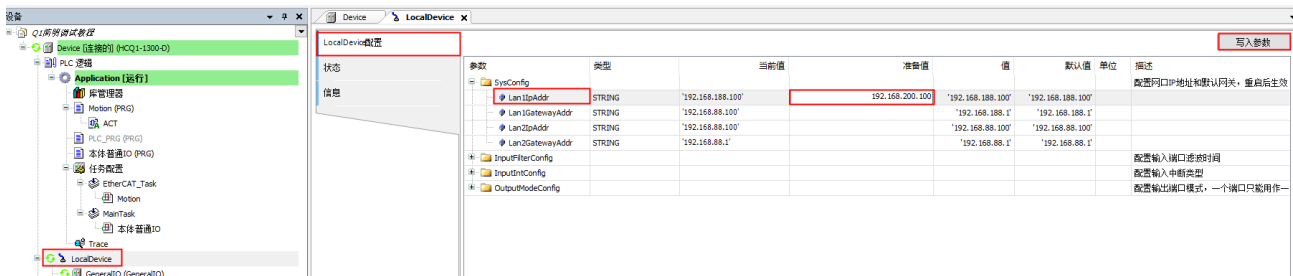


3.1.2 IP地址修改

Port口的IP地址也可以在软件中在线修改，登录

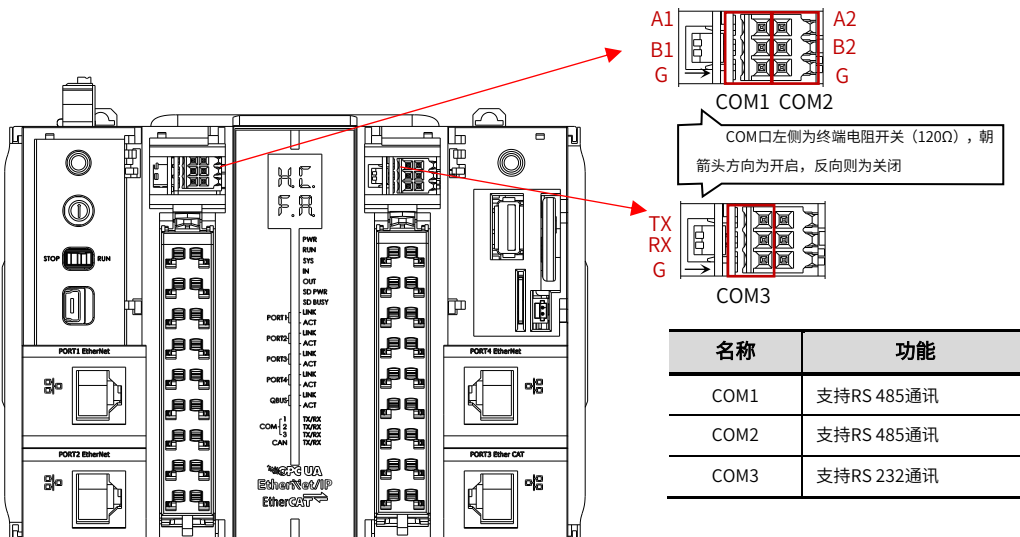


选择树形菜单中的【LocalDevice】，在右侧窗口中展开【SysConfig】→【IPAddr】，在准备值中输入需要修改的IP地址（示例以Port1为例），点击写入参数，在设备断电重启后生效



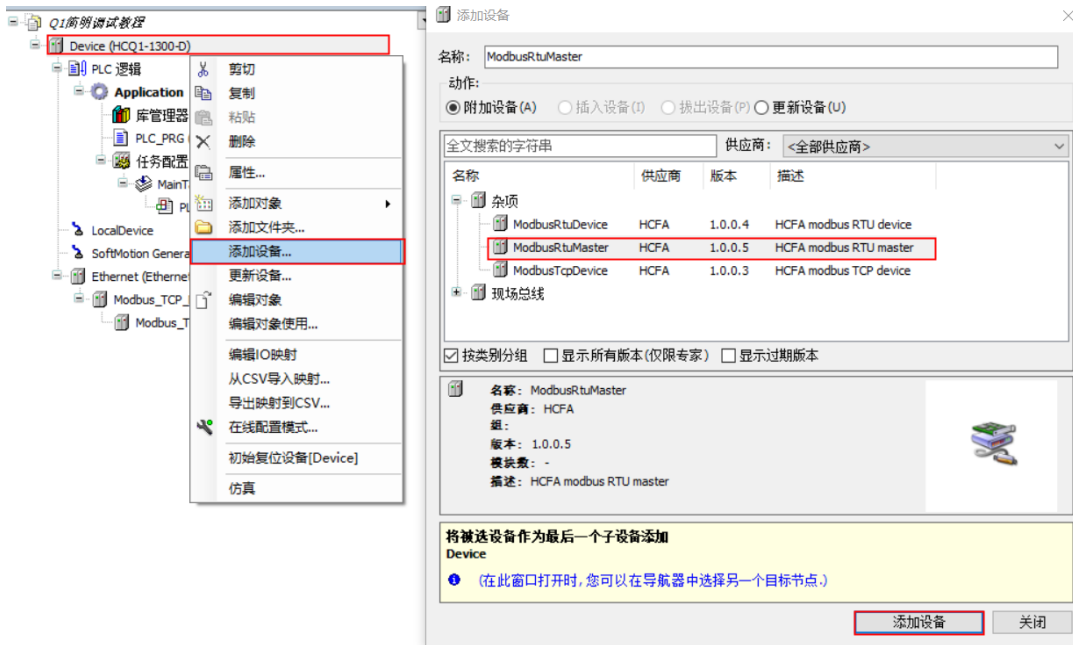
3.2 Modbus RTU通讯

3.2.1 外部硬件介绍

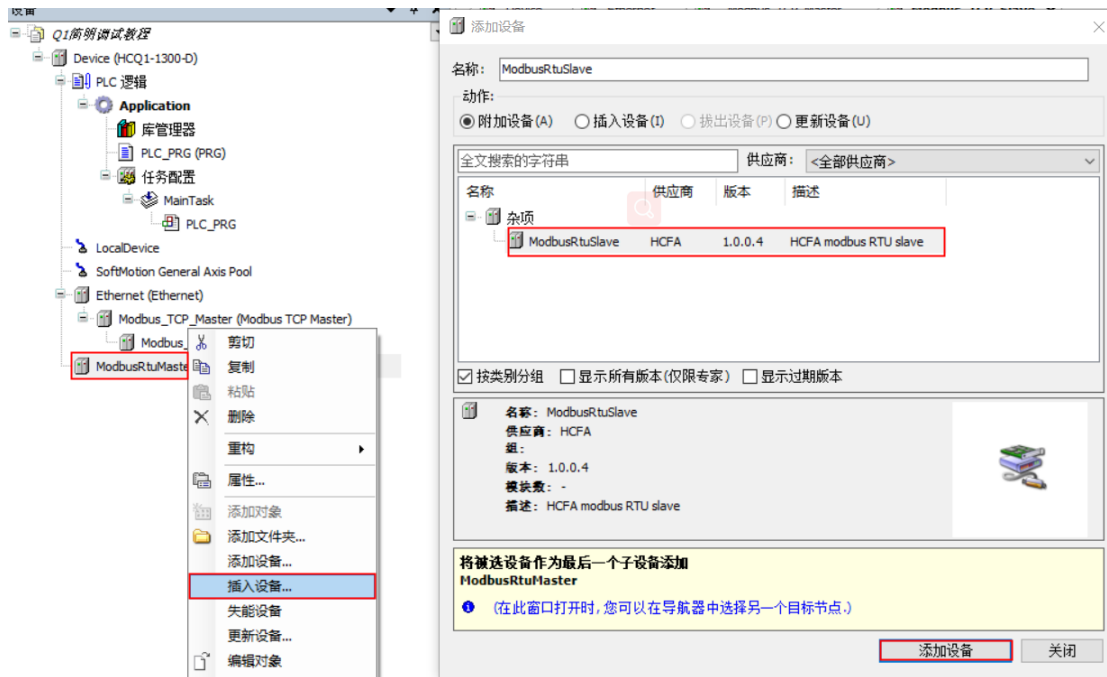


3.2.2 Q1做主站

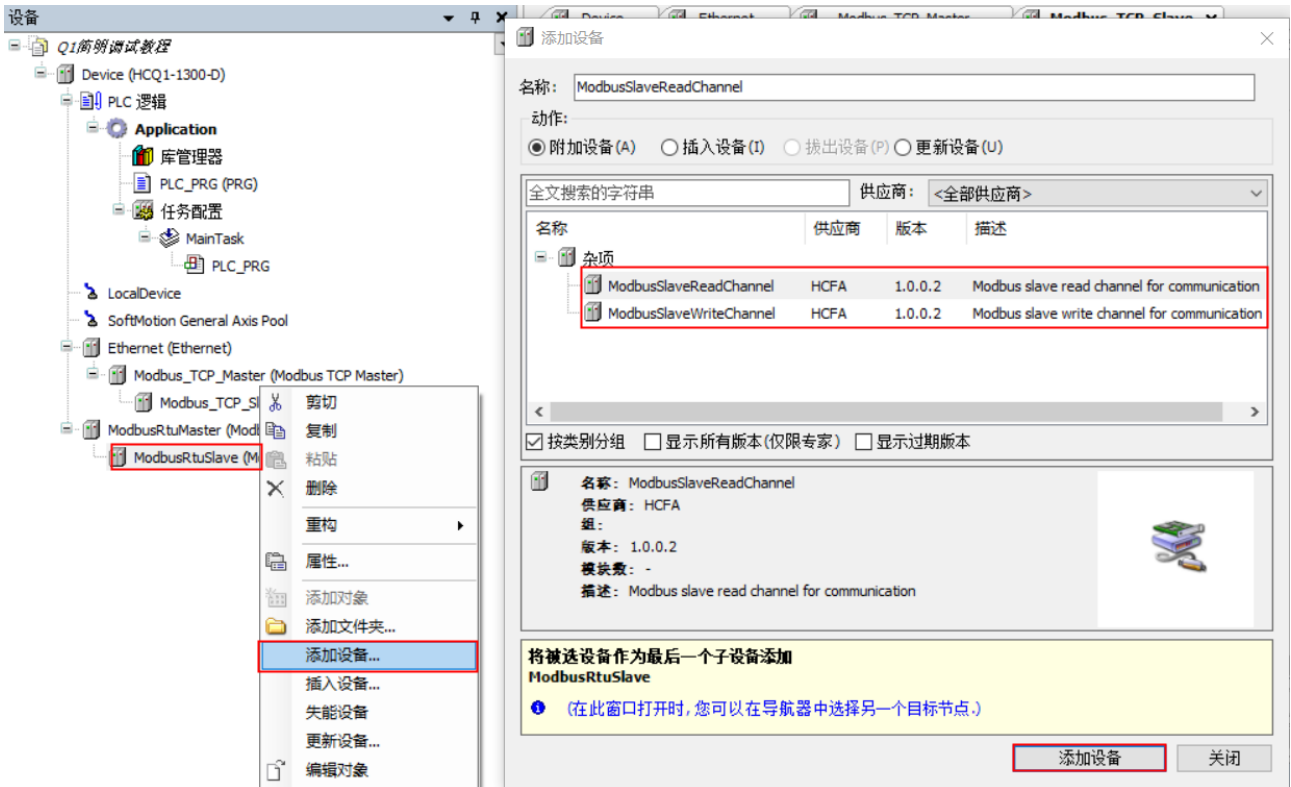
右击左侧树型菜单【Device】→选择【添加设备】，在弹出对话框中打开【杂项】，选择【ModbusRtuMaster】，最后点击【添加设备】。



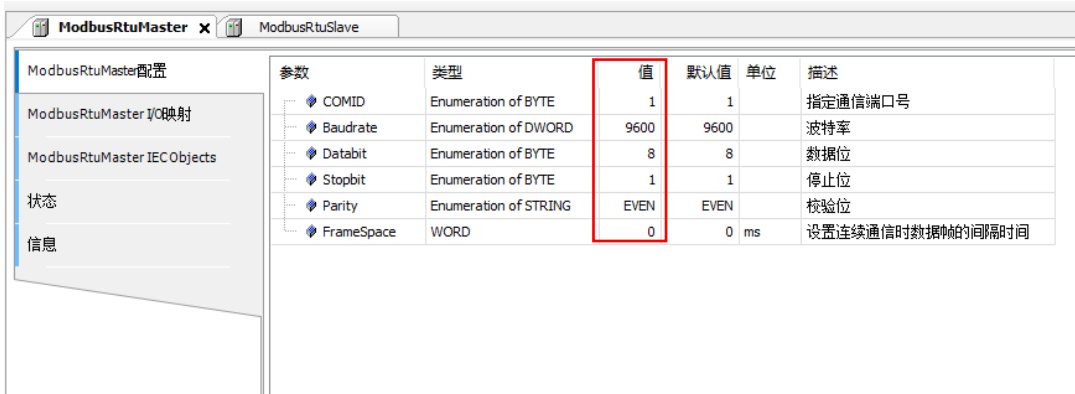
当使用Q1作为ModbusRTU主站时，同样需要在CODESYS的IDE界面中配置从站相关信息，如下图示，右击添加的【Modbus_RTU_Master】，选择【添加设备】，在弹出对话框里选择【ModbusRtuSlave】，最后点击【添加设备】完成添加。



接下来，还需在从站中配置输入和输出通道，右击【ModbusRtuSlave】→【添加设备】，在弹出窗口中打开【杂项】，选择【ModbusSlaveReadChannel】和【ModbusSlaveWriteChannel】，最后点击【添加设备】完成添加。



双击【ModbusRtuMaster】，按照下图所示进行配置：



各项参数说明：

COMID: 指定用于通信的COM端口，COM1、COM2、COM3均支持Modbus RTU，针对同一台PAC，不支持两个或三个COM口同时作Modbus RTU主站，目前Q1作主站最多支持添加50个从站，每个从站最多支持50个I/O通道。

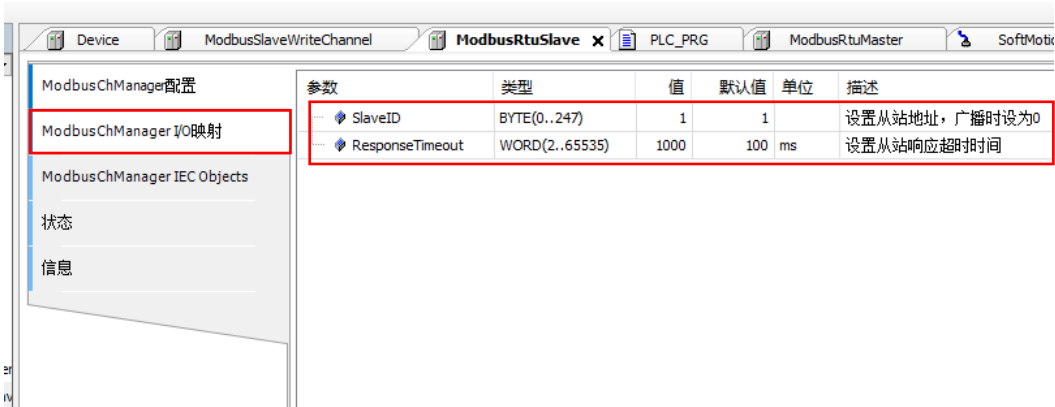
Baudrate: 所支持的波特率。可选1200、2400、4800、9600、19200、38400、57600、115200

Datebit: 数据位8位

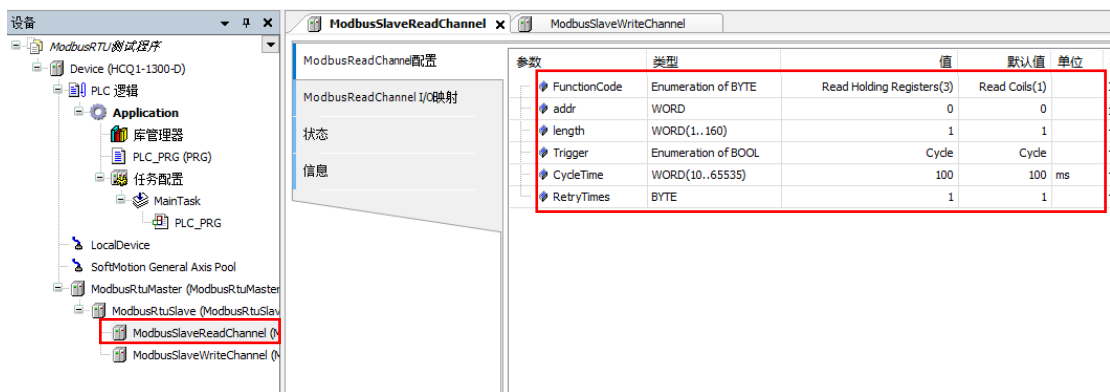
Stopbit: 停止位1位

Parity: 校验位，可选EVEN（偶校验）、ODD（奇校验）、NONE（无校验）

双击【ModbusRtuSlave】，【SlaveID】为从站地址，默认为1；【ResponseTimeout】为从站响应超时时间，设置为【1000】。

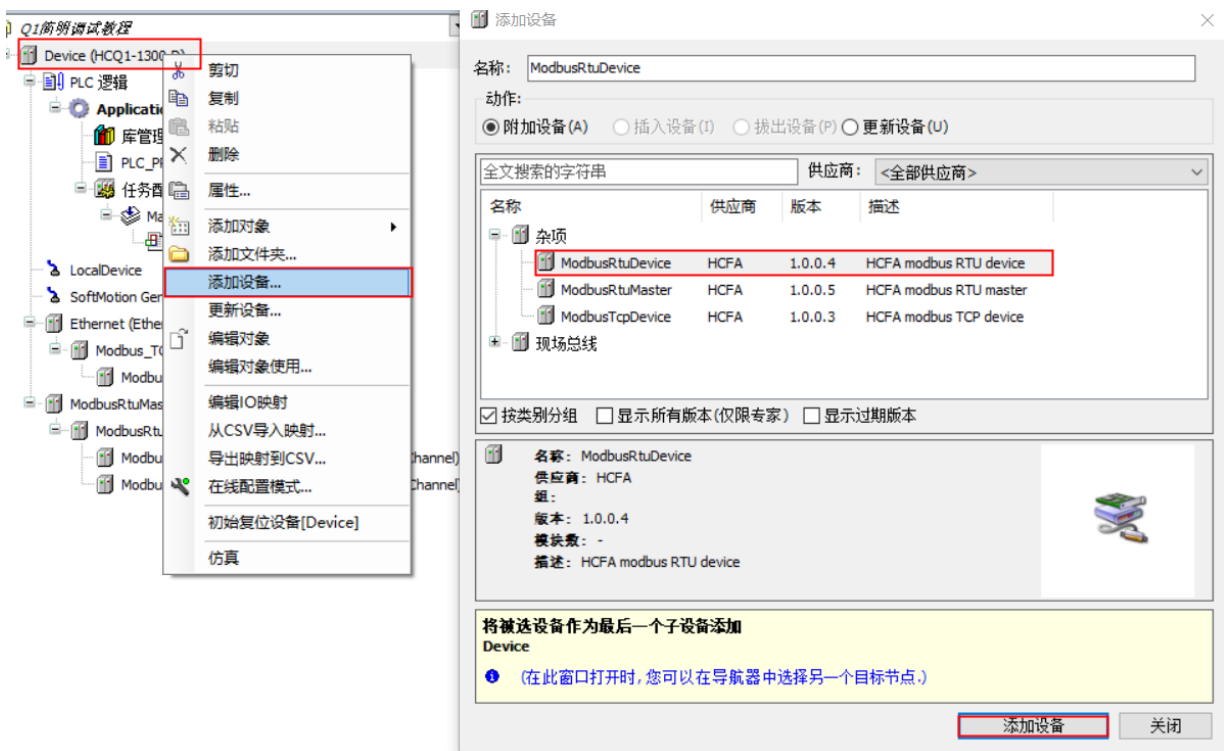


接下来在Modbus RTU从站中配置输入和输出通道，如下图所示进行配置：

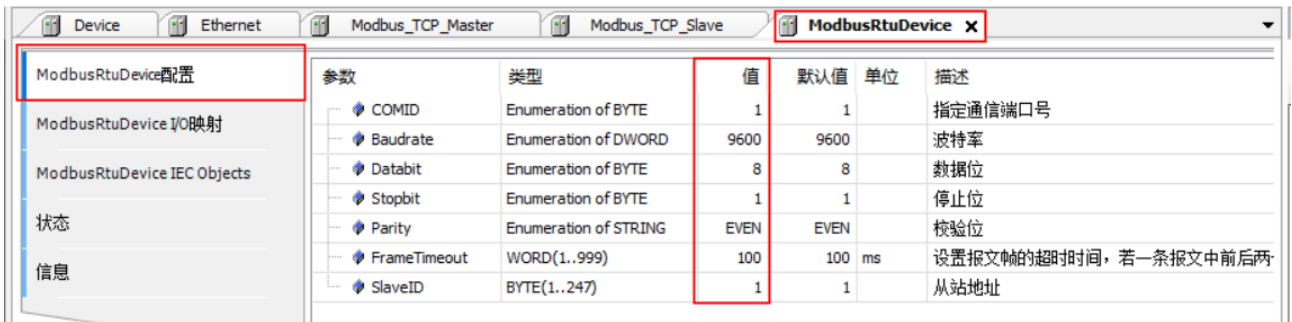


3.2.3 Q1做从站

右击左侧树型菜单【Device】→【添加设备】，在弹出对话框中打开【杂项】→【ModbusRtuDevice】，最后点击【添加设备】。



打开【ModbusRtuDevice】→【ModbusRtuDevice配置】，如下图所示进行配置：



各项参数说明：

COMID: 指定用于通信的COM端口，COM1、COM2、COM3均支持Modbus RTU，针对同一台PAC，不支持两个或三个COM口同时作Modbus RTU从站，Q1作为从站，仅能连接一个主站

Baudrate: 所支持的波特率。可选1200、2400、4800、9600、19200、38400、57600、115200

Datebit: 数据位8位

Stopbit: 停止位1位

Parity: 校验位，可选EVEN（偶校验）、ODD（奇校验）、NONE（无校验）

FrameTimeout: 设置帧超时时间，单位是ms。如果接收到的两个字节间隔超过超时时间则认为帧接收异常，忽略此帧。应根据实际情况设一个合理的值，并非越小越好；否则可能会造成误判，导致正常帧也被忽略。

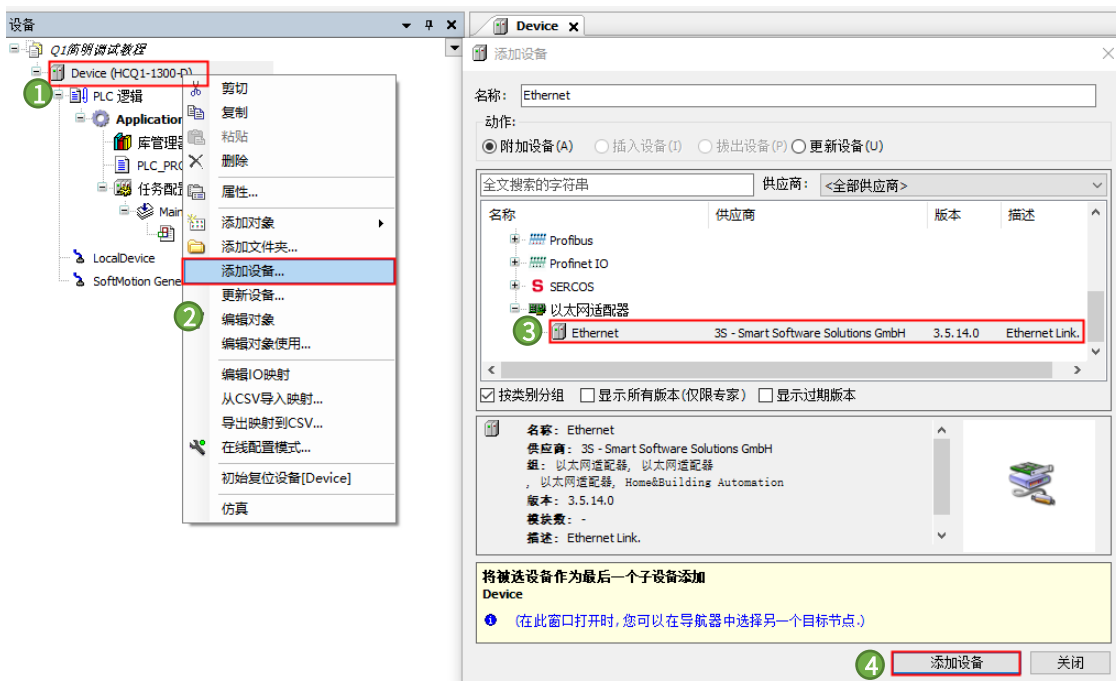
SlaveID: 从站的ID，设置范围：1~247。

本手册介绍简单的配置，具体示例参照Q系列Modbus RTU通讯功能介绍.pdf（可在川禾自动化学院免费获取）。

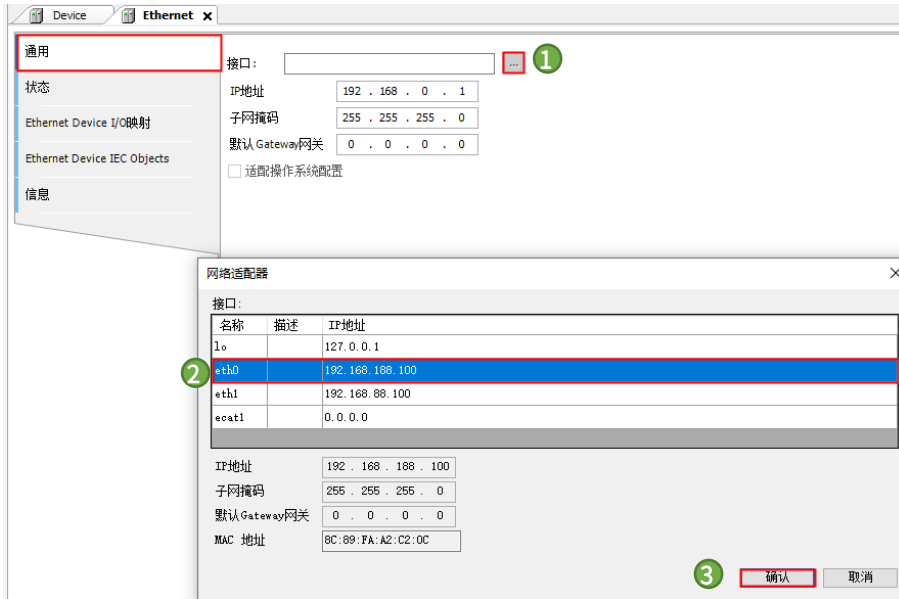
3.3 Modbus TCP通讯

3.3.1 Q1做主站

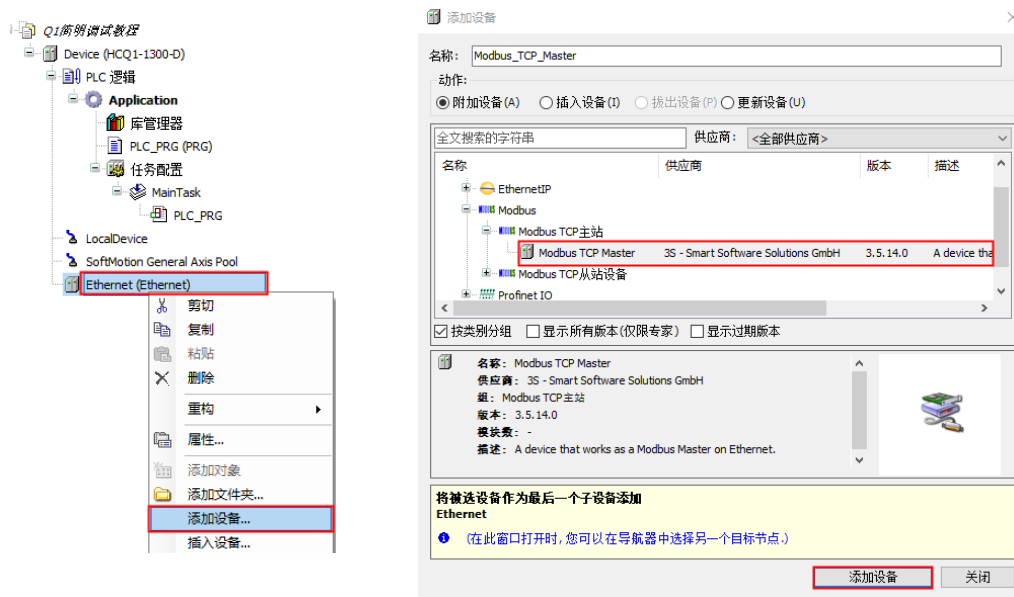
右击左侧树型菜单【Device】→选择【添加设备】，在弹出对话框中下拉选择【以太网适配器】下的【Ethernet】，最后点击【添加设备】。



双击添加的【Ethernet】，在右侧通用界面中点击接口右侧按钮，选择eth0（对应Q1的Port1，请根据实际连接的Port口选择），最后点击【确认】。



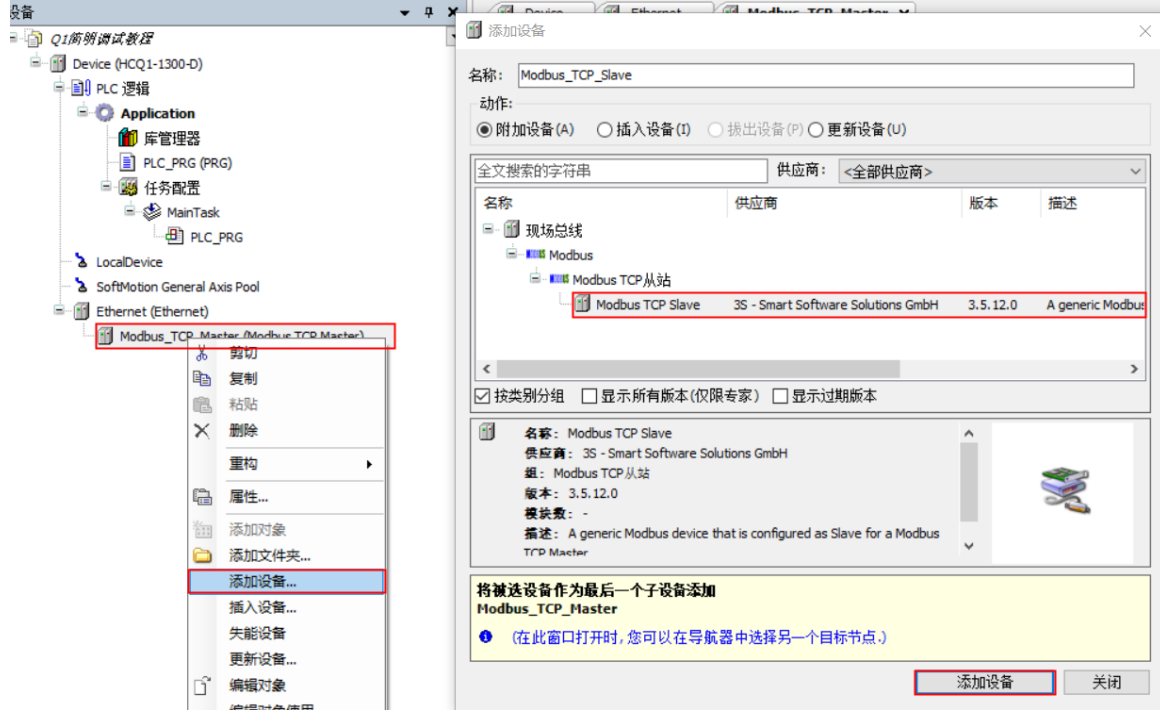
确定后左侧设备树中会出现【Ethernet】，右击【Ethernet】选择【添加设备】，在弹出对话框中选择【Modbus】→【ModbusTCP主站】→【ModbusTCPMaster】，最后点击【确定】。



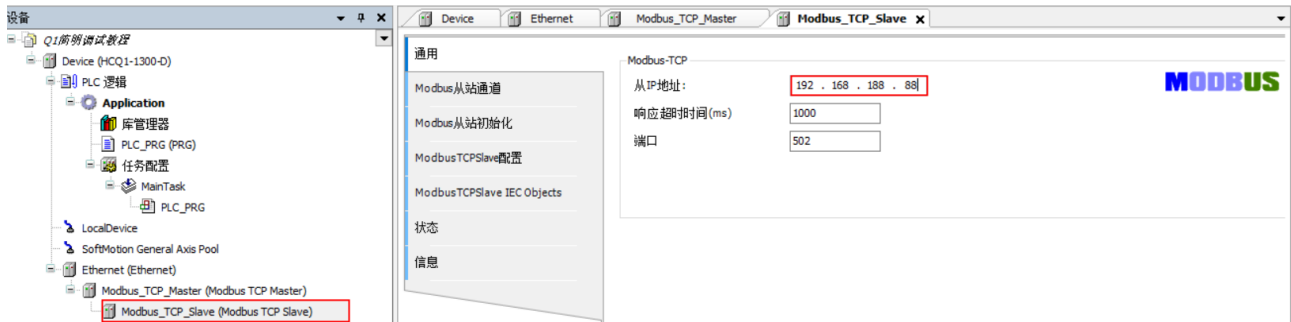
双击【Modbus_TCP_Master】，在【通用】选项卡下勾选【自动重新连接】，防止断电重启后通讯不连接。



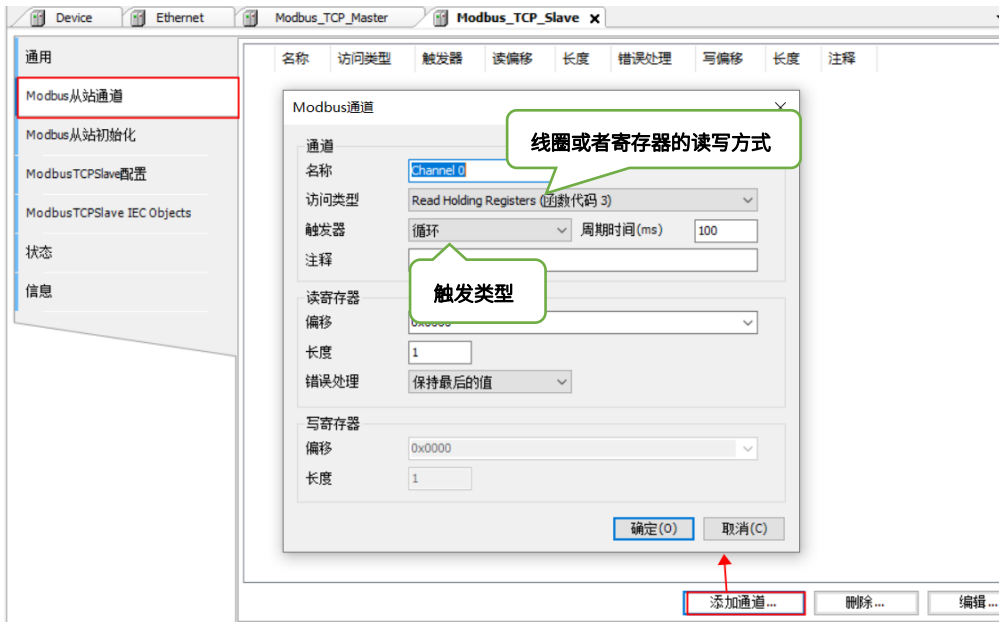
当使用Q1作为Modbus TCP主站时，同样需要在CODESYS的IDE界面中配置从站相关信息，如下图示，右击添加的【Modbus_TCP_Master】，选择【添加设备】，在弹出对话框里选择【Modbus TCP Slave】。



双击【Modbus_TCP_Slave】，在通用选项卡修改从站IP地址为192.168.188.88（从站设备IP地址，确保同一个网段），端口设置为502。

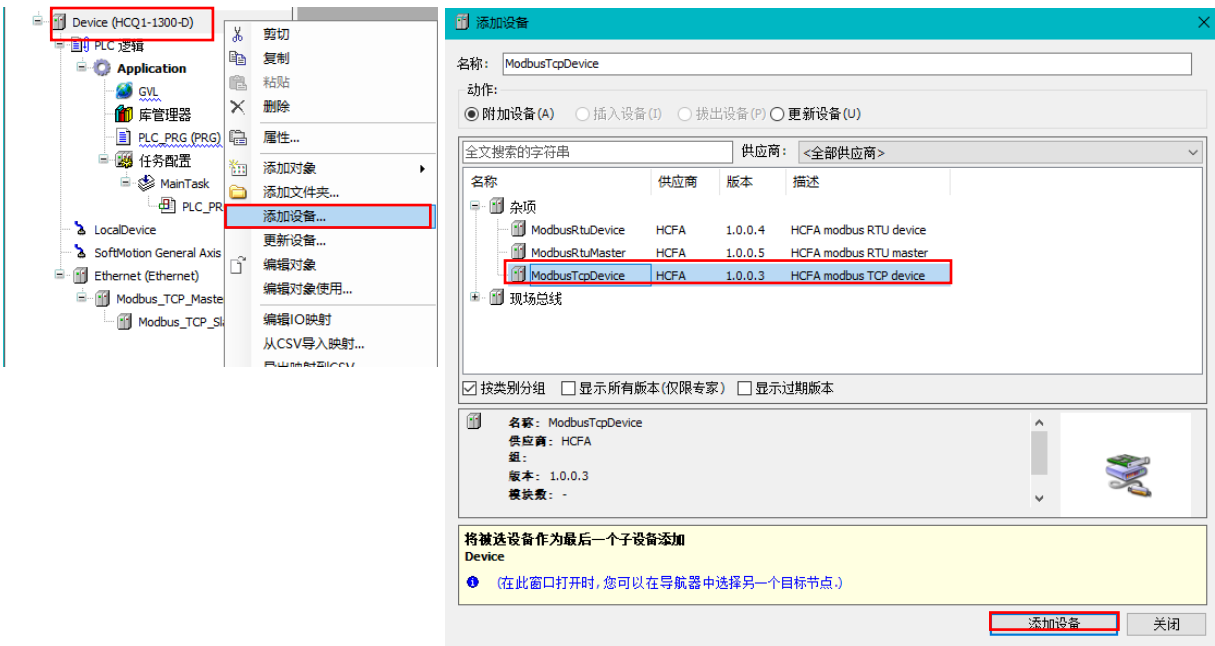


选择【Modbus从站通道】，点击右下角【添加通道】，设置通道名以及访问类型还有长度跟偏移量，如下图示，在本次测试中，设置通道名为channel0，访问类型为【Read Holding Registers】（访问类型函数代码与Modbus报文功能码一致），长度为【5】，偏移量为【0】，点击【确定】建立通道。

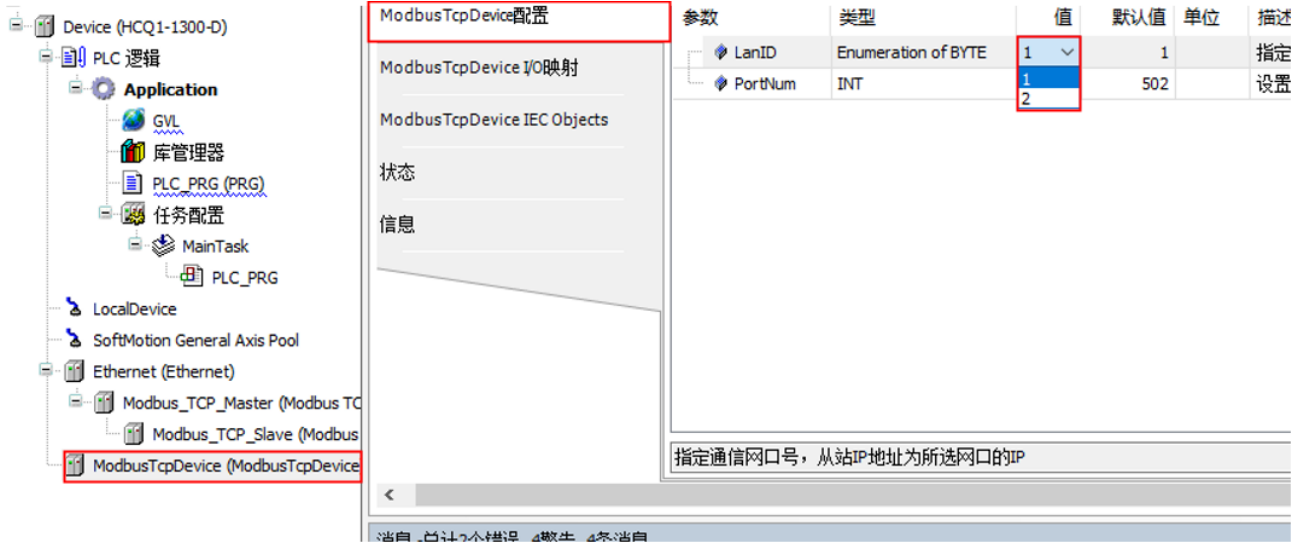


3.3.2 Q1做从站

右击左侧树型菜单【Device】→选择【添加设备】，在弹出对话框中打开【杂项】，选中【ModbusTcpDevice】，最后点击【添加设备】。



打开【ModbusTcpDevice】→【ModbusTcpDevice配置】，根据上位机与Q1所连的Port口来选择【LanID】（1为Port1口；2为Port2口），本次实验连接的是Port1口，故此处【LanID】选择【1】；端口号【PortNum】默认为502。

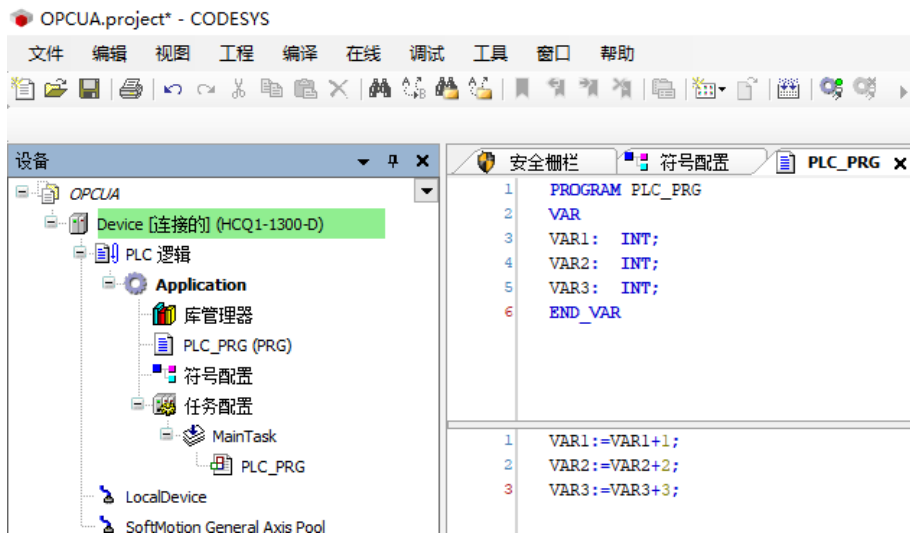


本手册介绍简单的配置，具体示例参照Q系列Modbus TCP通讯功能介绍.pdf（可在川禾自动化学院免费获取）。

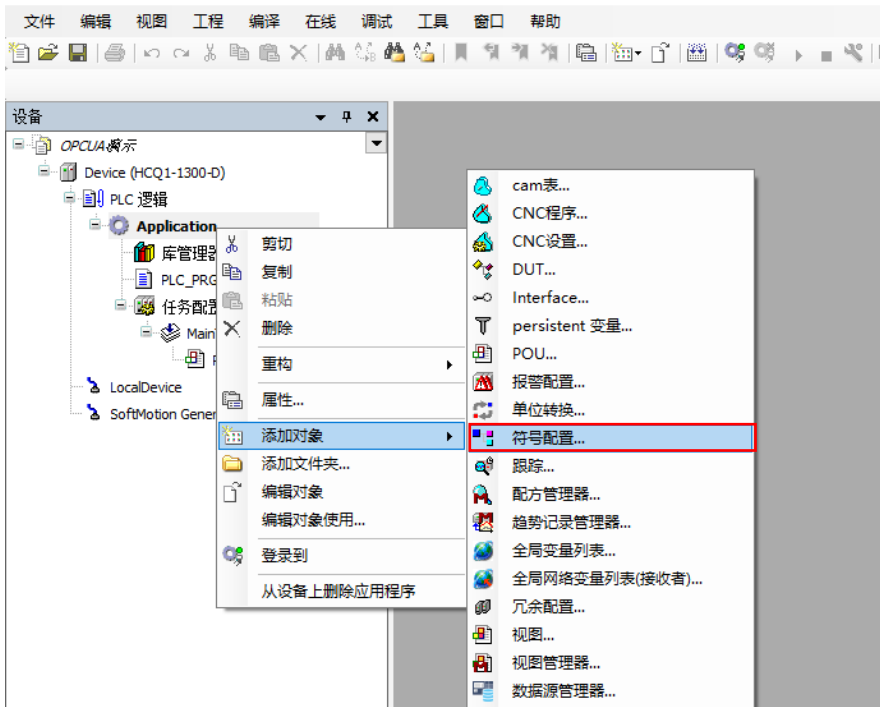
3.4 OPCUA通讯

3.4.1 程序编写及配置

在工程中写入以下程序（可随意写入一段有数据变化的程序）：



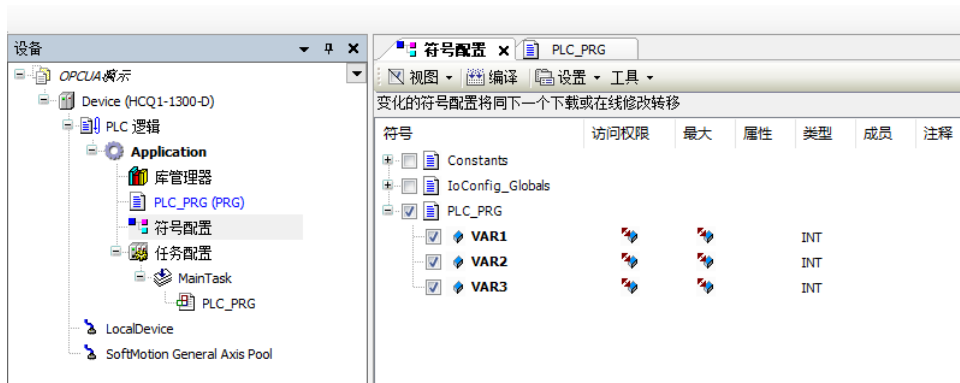
在【Application】对象下添加符号配置对象



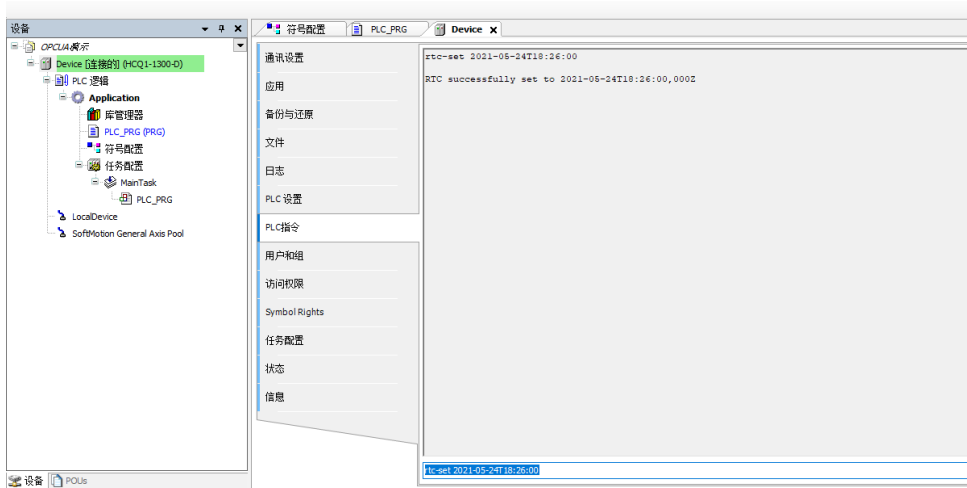
在符号配置对话框中勾选【支持OPC UA特征】，单击【打开】



编译程序，在符号配置对象中勾选需要添加到OPCUA服务器中的变量



双击设备树中的【Device】，进入【PLC指令选项卡】，在下方命令行输入设定当前的时间指令，如：【rtc-set 2021-05-24T18:26:00】



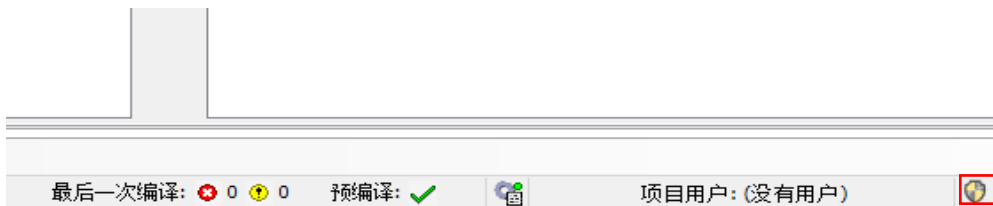
登录程序到Q1中并启动程序。



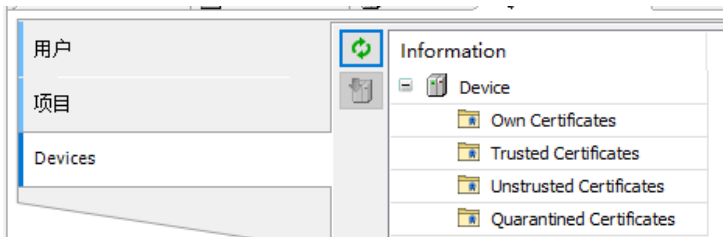
3.4.2 配置Codesys Security Agent package

安装Codesys Security Agent package，描述文件的安装方法见（附录：描述文件安装）。

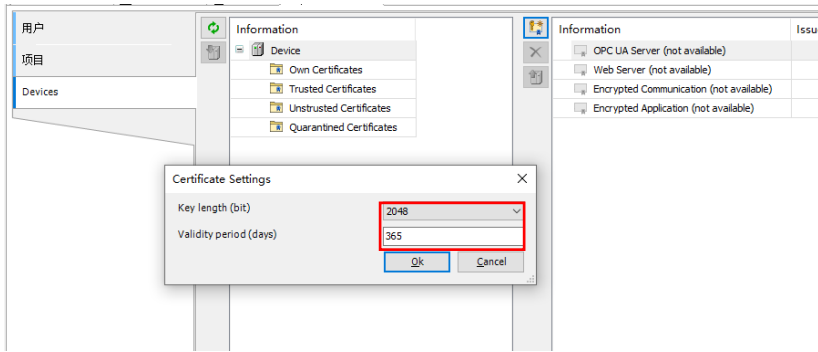
重启Codesys软件，打开右下角盾牌图标：



单击【Device】选项卡，单击绿色的刷新按钮，【Information】列出现【Device对象】：



单击【Device】对象，选择右边的【OPC UA Server(not available)】，单击【创建】按钮，会跳出【CertificatesSettings】弹框，【KeyLength】选【2048】，【Validity period】默认【365】，单击【OK】，创建OPC UA Server证书，Information下的OPC UA Server会处于【OPC UA Server (calculating)】状态，等待一段时间这个状态消息即创建完成。



本手册介绍简单的配置，具体示例参照Q系列OPC UA使用说明.pdf（可在川禾自动化学院免费获取）。

第四章 程序运行

4.1 任务配置设置

每一个PLC应用下都可以创建多个POU，在同一个应用下，不同POU之间的执行顺序是通过“任务”来配置的。在“任务”中完成POU的配置后，POU会按照任务配置周期性或由一个特定的事件触发开始执行。

4.1.1 任务配置

本教程中仅介绍监视和属性，其他可参考软件手册。

监视

当程序进入在线模式后，在“任务配置”右侧的监视窗口中，用户可以监控当前状态、循环次数、程序实际执行时间，平均/最大/最小循环时间等任务执行相关参数。

Task	Status	IEC-Cycle Count	Cycle Count	Last Cycle Time (μs)	Average Cycle Time (μs)	Max. Cycle Time (μs)	Min. Cycle Time (μs)	Jitter (μs)	Min. Jitter (μs)	Max. Jitter (μs)
⊕ EtherCA...	有效的	143297	144544	150	141	2750	50	49	-25	24
⊕ MainTask	有效的	143295	144542	0	0	50	0	400	-	400

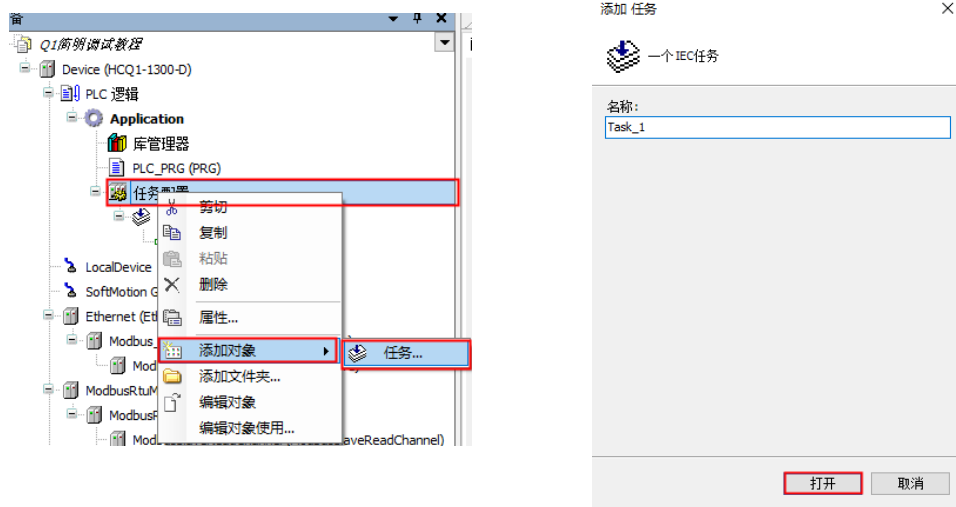
属性

属性选项卡下显示当前任务配置的基础属性。

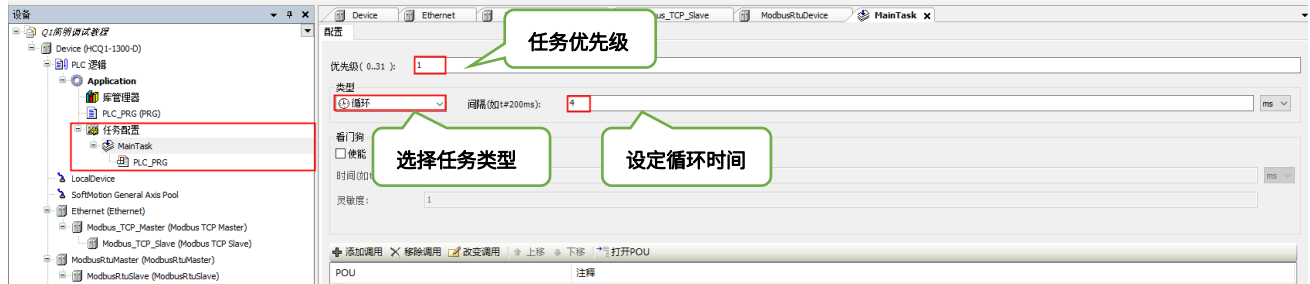


4.1.2 任务

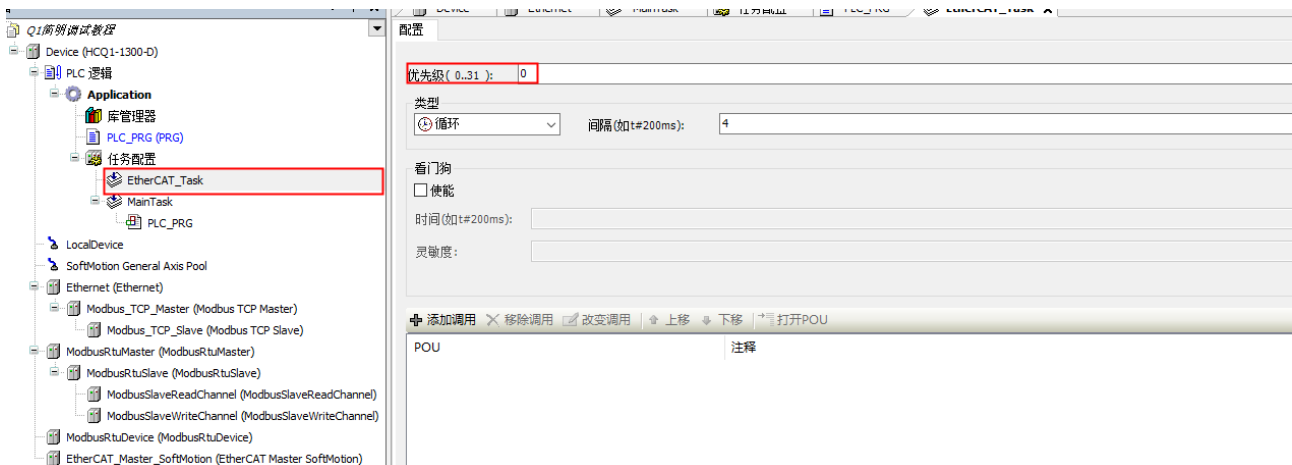
右击【任务配置】→【添加对象】→【任务】，定义任务名称后完成新任务的创建，不同类型的任务最多可以创建100个（任务配置属性中有描述），按照用户设定的优先级顺序执行，数字越小优先级越高，优先级相同的情况下，按照在任务配置当中的顺序从上往下执行。



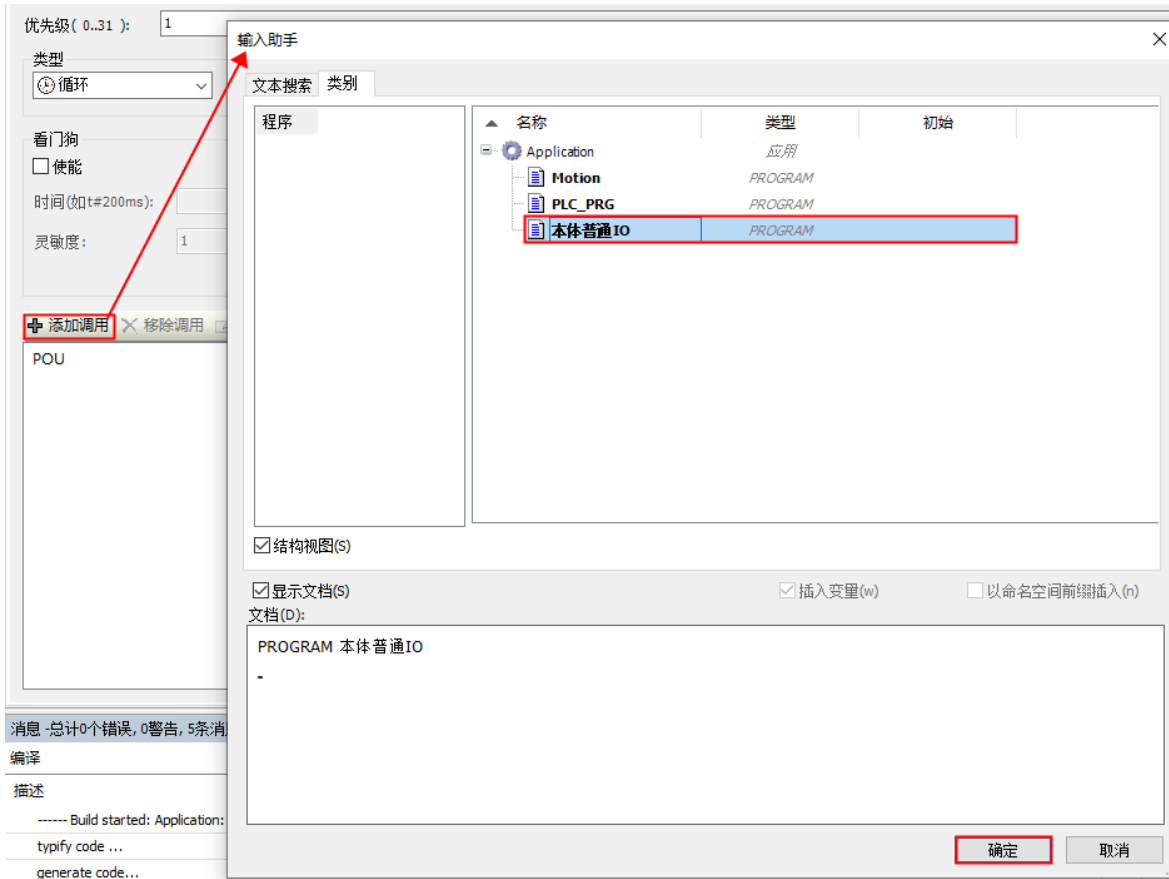
在树形菜单任务配置中可以进行管理，新建一个标准的PLC项目会自动生成一个循环执行的任务，该任务自动关联PLC_PRG，任务类型默认为循环，周期是4ms，优先级为1，PLC程序只有被任务调用才会参与编译和实际执行。



若添加【EtherCAT_Master_SoftMotion】,则会自动生成【EtherCAT_Task】,由于运动控制的优先级需要最高,添加后需要将修改优先级为【0】



程序需要在任务中调用才能运行，点击【添加调用】，选择你需要调用的程序，也可以在此【移除调用】

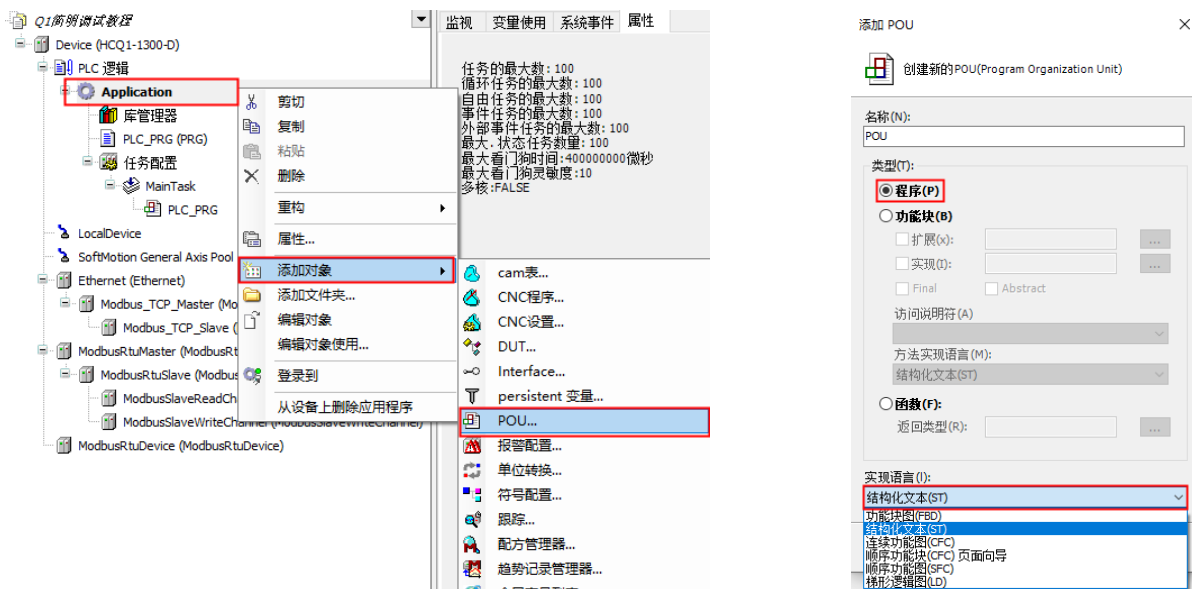


4.2 程序 (PRG)

4.2.1 POU的程序 (PRG) 添加

由于创建时选择的是标准工程，会自动创建一个PLC_PRG，若想创建多个程序如下。

新建POU (Program Organization Unit)，首先右击【Application】→【添加对象】→【POU】，填写POU的名称，选择程序块类型，支持六种编程语言，选择结构化文本 (ST) 编写示例程序。

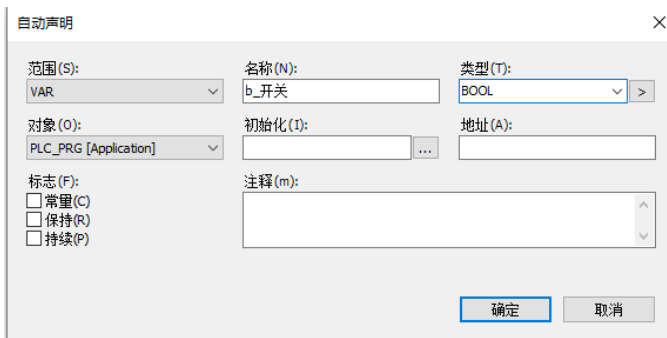


双击【PLC_PRG】进入编程界面，从上往下分别是【变量声明窗口】，【主程序窗口】，【消息窗口】，程序编译产生

的错误、警告和编译信息都会显示在【消息窗口】中



变量声明区域窗口中Shift+F2



范围：变量作用域

名称：变量名称

类型：变量数据类型

对象：变量所在应用

初始化：变量初始值

地址：变量和外部硬件点之间的映射关系

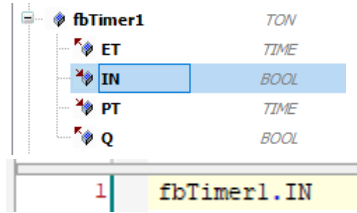
标志：可以设置变量类型为常量、保持和持续型

注释：注释，格式为 (*注释内容*)

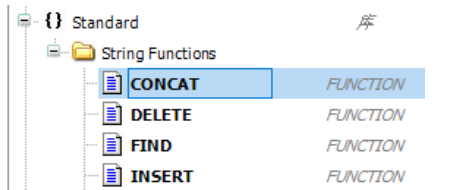


在主程序窗口编写示例程序，正确声明的变量可以在主程序窗口中通过F2调用，调用时需要注意调用类别

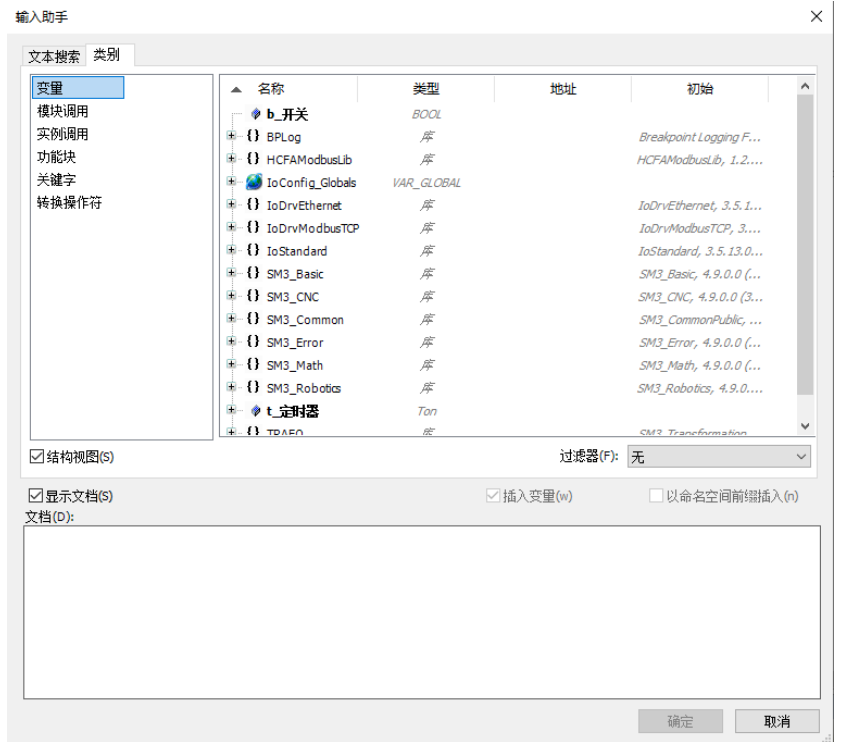
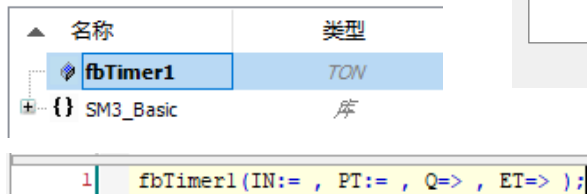
变量：单个变量，需声明



模块调用：函数调用，无需声明，直接调用

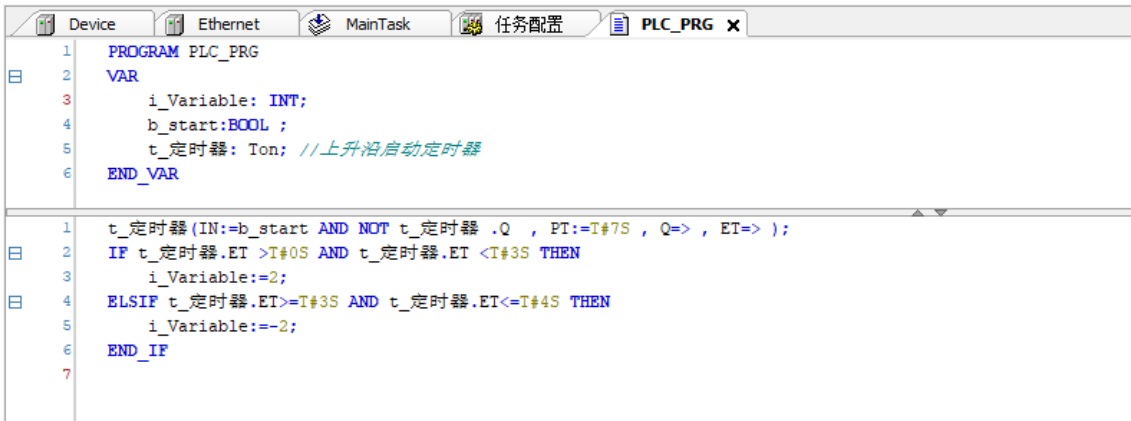


实例调用：功能块语句，需声明



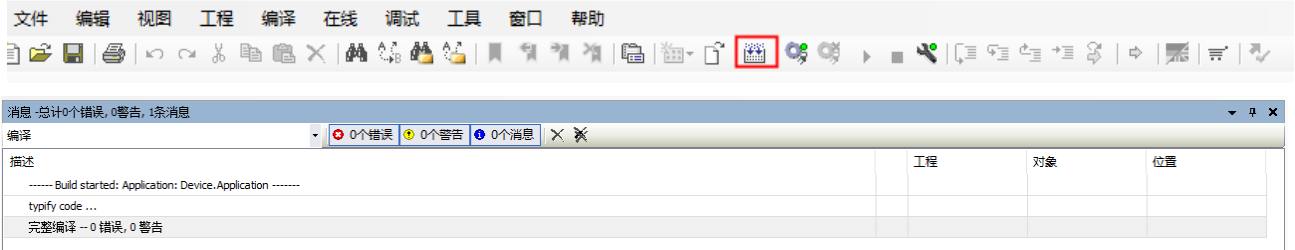
4.2.2 程序运行

程序示例如下所示



编译后，查看报错窗口

Q1简明调试教程.project* - CODESYS

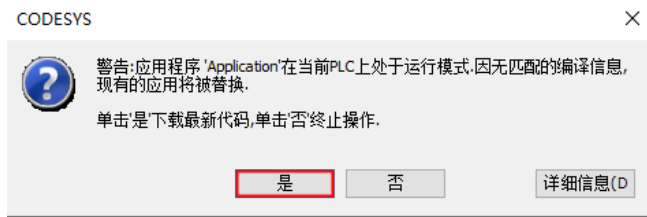


若无报错，则登录运行程序

Q1简明调试教程.project* - CODESYS

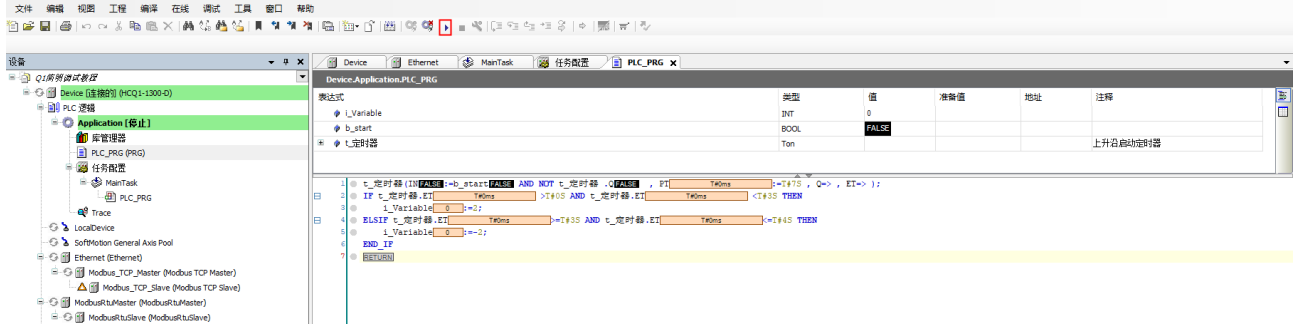


跳出弹窗，【警告】下载程序到PLC中，选择【是】

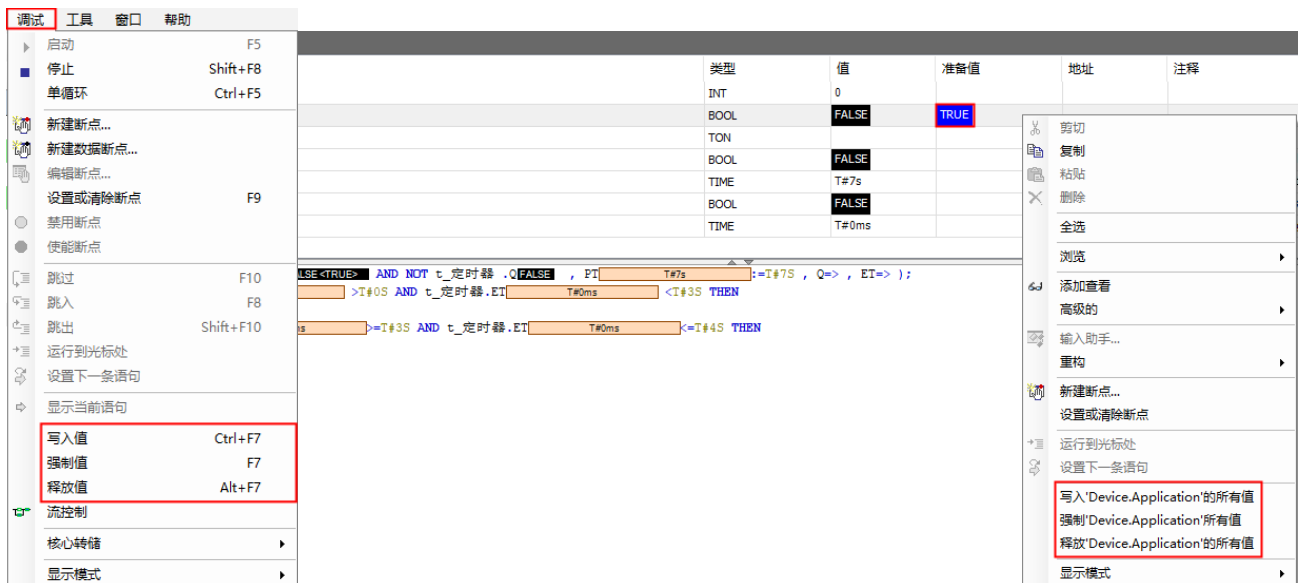


运行程序

Q1简明调试教程.project* - CODESYS



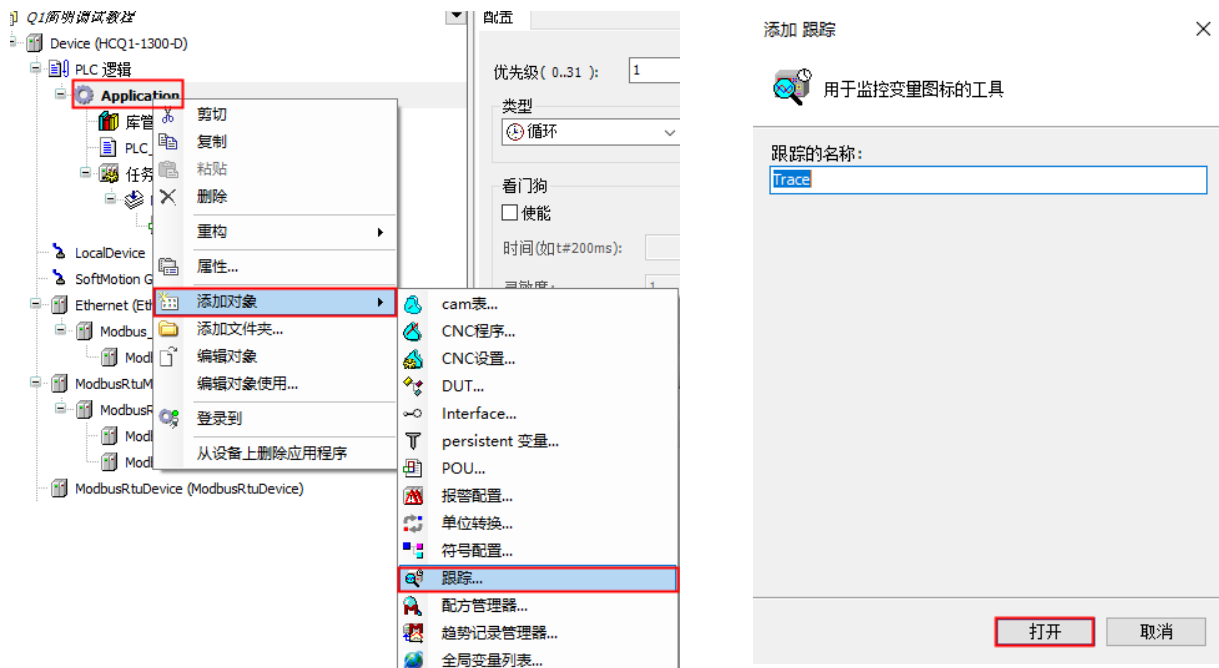
若想在线写入数值，填入准备值后，右击空白处选择【写入 'Device Application' 的所有值】或者菜单栏【调试】→【写入值】，又或者快捷键【Ctrl+F7】。通过强制值写入的数值左侧 **F** 会有标识，可以通过释放值将其释放还原。



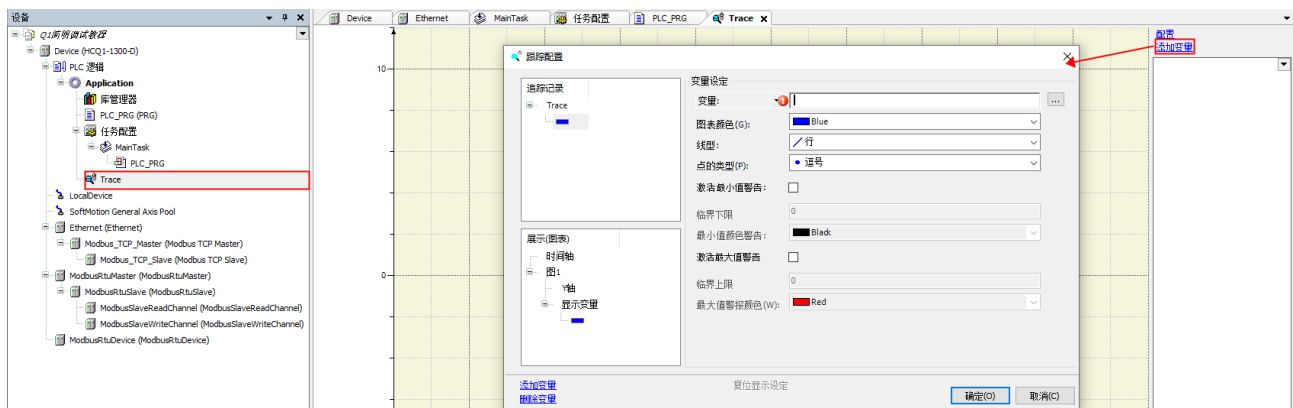
4.2.3 Trace

采样追踪是CODESYS提供的图形化数据监控软件，类似示波器，在程序调试和诊断过程中，是非常实用和有效的工具。在程序运行过程中产生的数据都是转瞬即逝的，无法根据实时产生的数据进行分析，通过采样追踪，可以把程序执行过程中产生的过程数据全部记录下来，例如运动控制过程中的电机当前位置、速度、加速度等。通过对采集到的数据分析，就可以清楚的观察到系统运行的整个过程。

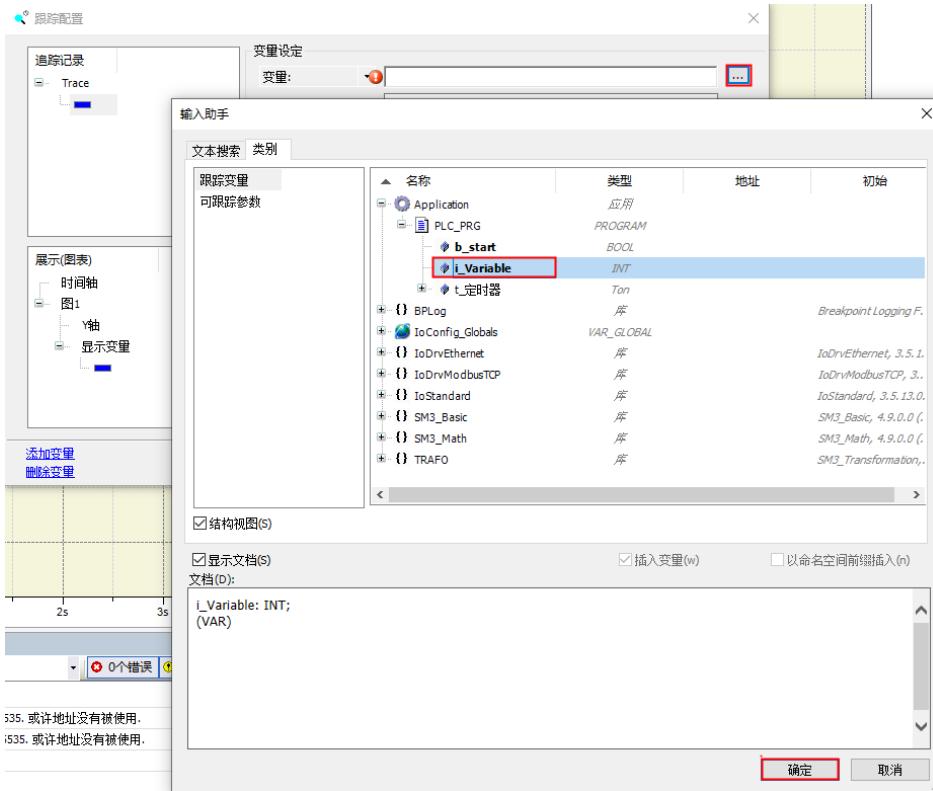
右击【Application】→【添加对象】→【跟踪】，命名【跟踪的名称】（只能识别ASCII，不能使用中文）



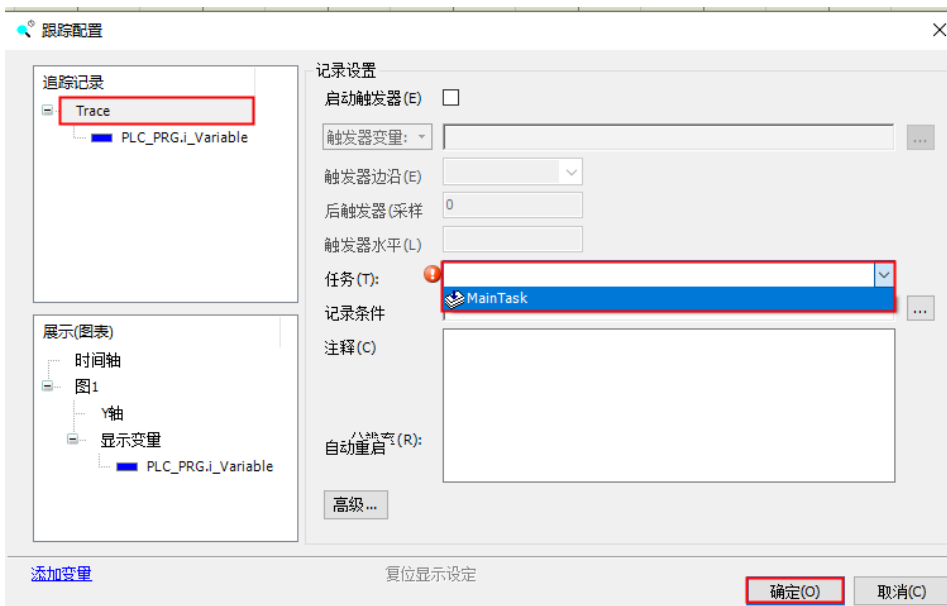
单击【Trace】，以示例程序为例，添加追踪变量



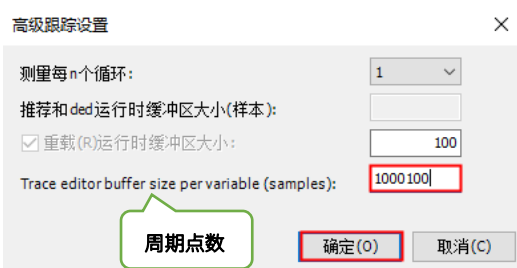
点击变量右侧的【输入助手】，在【Application】下找到对应程序下需要跟踪的变量，点击【确定】



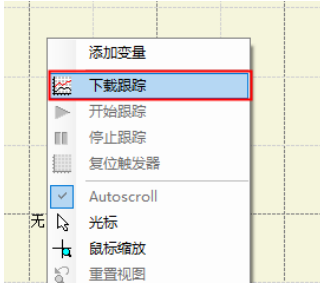
单击【Trace】，选择采样周期追随任务【Main Task】



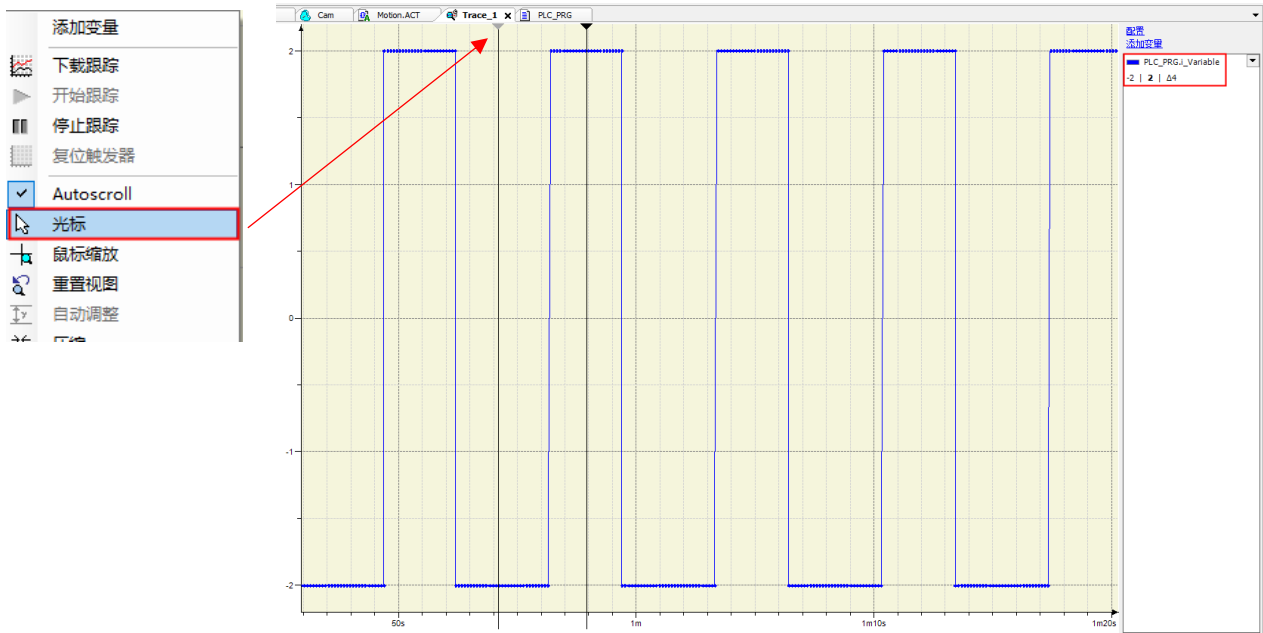
选择【高级】，增大跟踪个数，选择【确定】



在线监控示例，程序登陆后，在Trace的空白处右击，选择【下载跟踪】

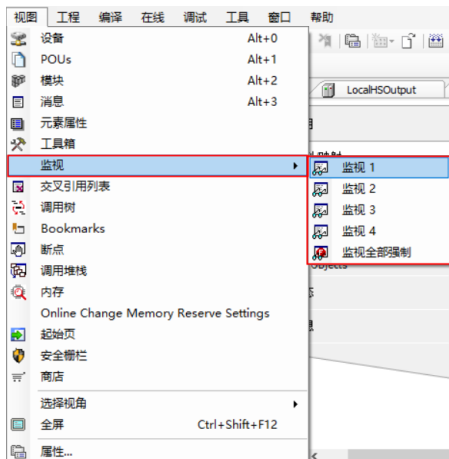


观察追踪结果，方便查看可以右键空白处，选择【光标】（最多只有两个光标），可查看右侧变量下方数据

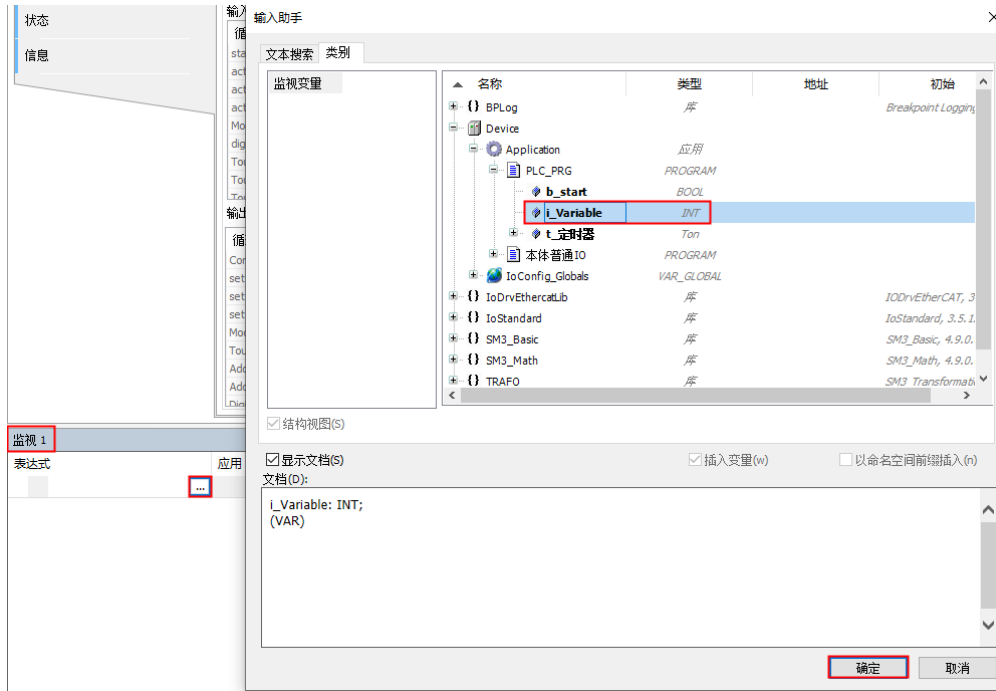


4.2.4 在线监控

程序成功登录运行后，主程序窗口和声明窗口都会显示变量的实际值。如果用户想要同时监视多个POU，可以把需要观察的变量集合起来，在菜单栏找到【视图】→【监视】→【监视1】，则程序会自动建立监视1列表，最多可以创建4个监视列表。



打开监视1，单击表达式下的空栏，可以通过点索引的方式直接调用不同POU中的变量或全局变量，也可以通过点击右侧展开输入助手，添加需要在线监视的变量。



配置完成后在程序正常运行过程中可以通过消息窗口下的“监视1”选项卡查看变量的实时值，以及对当前变量进行赋值等操作，即使是程序正常运行过程中也可以手动添加或者删除监控变量，非常方便



4.2.5 软件复位

在调试中，如果用户希望对程序进行复位操作，CODESYS提供了以下三种复位方式：热复位、冷复位和初始值复位。

热复位：模拟程序运行时断电，或者通过控制开关闭控制器，然后再打开（热复位）。

除了保持型变量外其他当前的应用都被重新初始化。

冷复位：模拟程序下载到PLC，运行之前。

与热复位不同，冷复位不但将普通变量重置为初始值，也将保持型变量的值设为初始值0。

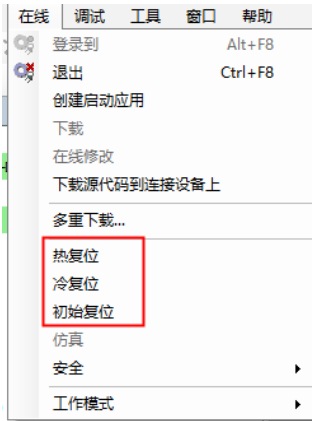
初始值复位：将设备复位到初始状态。设备中的任何应用、引导工程和剩余变量都将被清除。

不同指令执行后会对程序中定义的不同类型的变量的影响见下表。

在线命令	VAR	VAR RETAIN	VAR RETAIN PERSISTENT
热复位	✓	×	×
冷复位	✓	✓	×

初始值复位	√	√	√
下载	√	√	×
在线改变	×	×	×

菜单栏中选择【在线】，选择需要复位的方式（只能在线复位）。

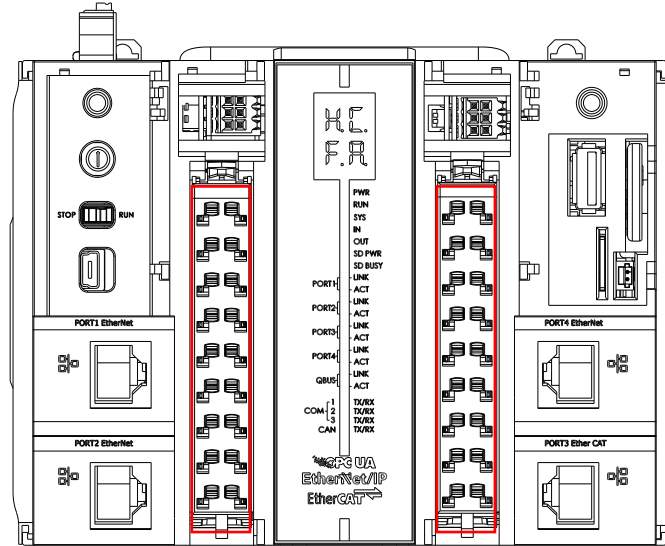


第五章 Q1本体IO

Q1本机自带16位普通输入输出，8位高速输入输出。

5.1 Q1自带普通输入输出

左侧为16位普通IO输入，右侧为16位普通IO输出。



16路普通输入端子排布

Input	I0	I10
Input	I1	I11
Input	I2	I12
Input	I3	I13
Input	I4	I14
Input	I5	I15
Input	I6	I16
Input	I7	I17
公共端	SS	SS

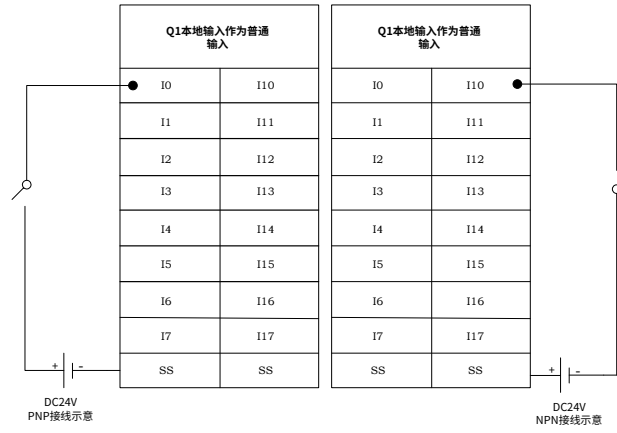
16路普通输出端子排布

Output	Q0	Q10
Output	Q1	Q11
Output	Q2	Q12
Output	Q3	Q13
Output	Q4	Q14
Output	Q5	Q15
Output	Q6	Q16
Output	Q7	Q17
公共端	COM	COM

5.1.1 Q1硬件接线

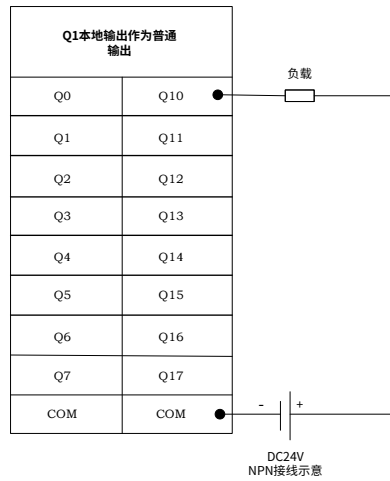
■ 本地输入作为普通输入外部接线示意图

输入接法支持NPN和PNP型



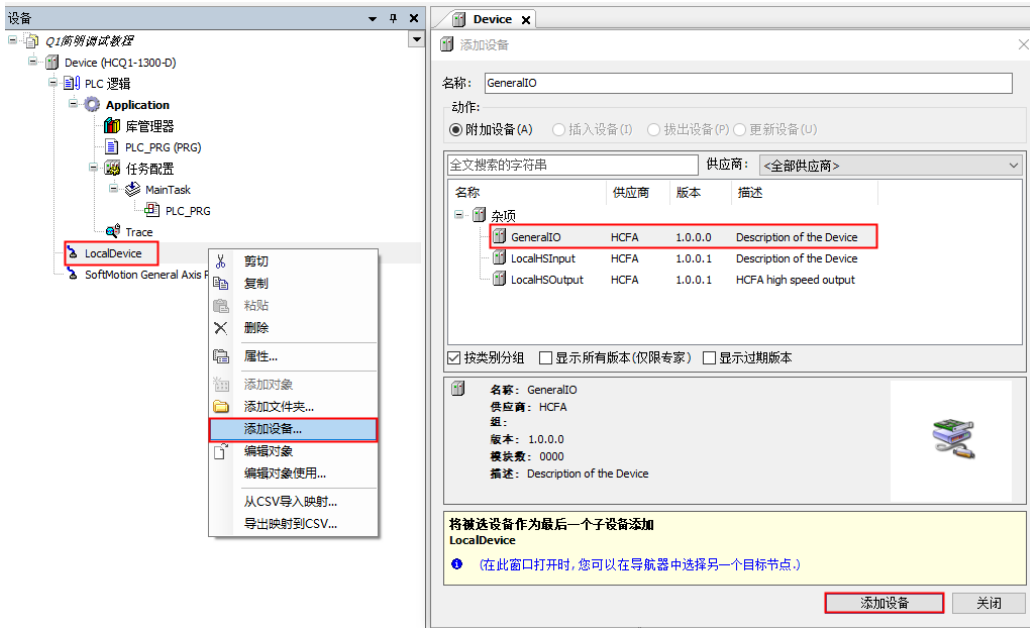
■ 本地输出作为普通输出外部接线示意图

输入接法支持NPN

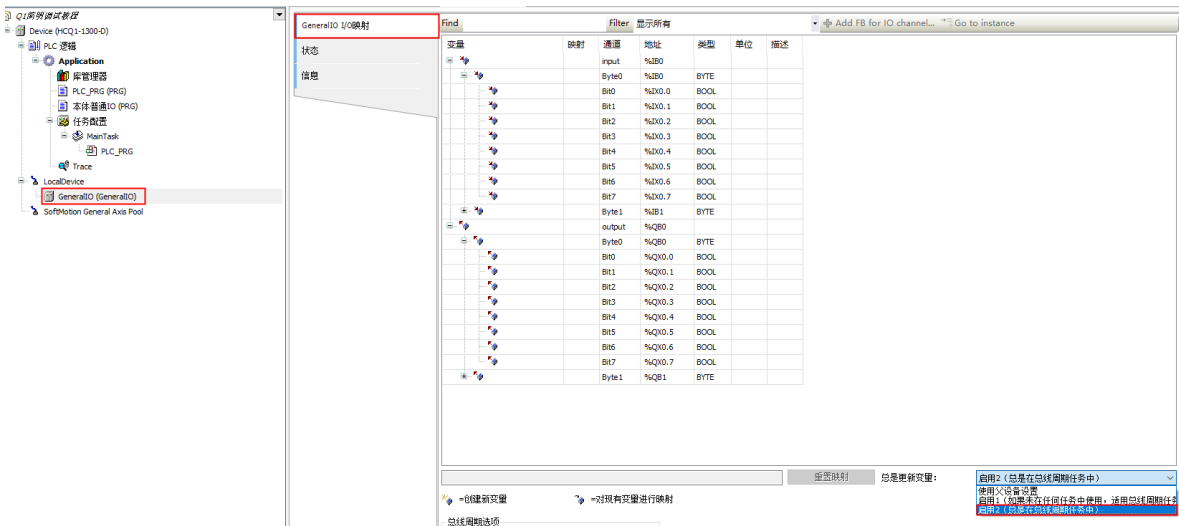


5.1.2 程序示例

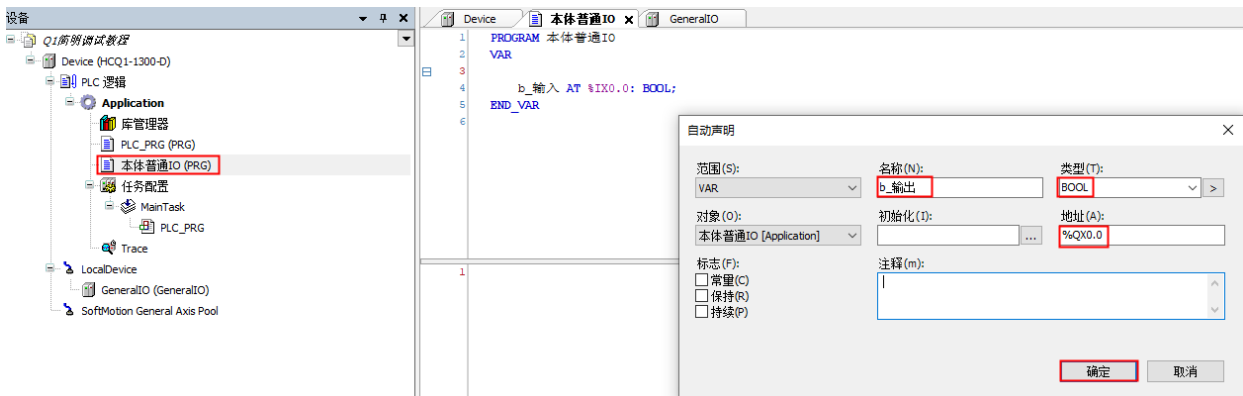
右击【LocalDevice】→【添加设备】→【GeneralIO】→【确定】



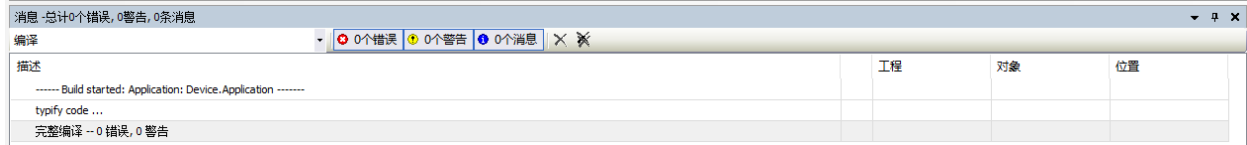
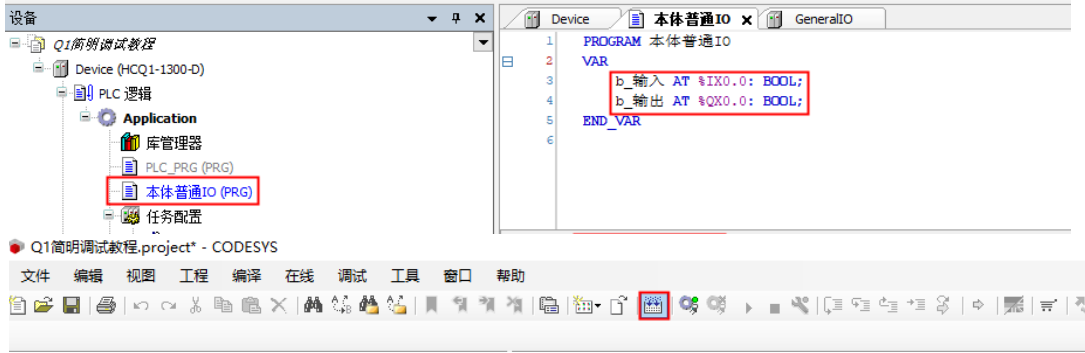
单击【GeneralIO】→【GeneralIO I/O映射】，修改【总是更新变量】为【总是在总线周期任务中】



单击程序【本体普通IO】（程序的添加参考3.1POU的程序），【Shift+F2】自动声明映射输入输出端口



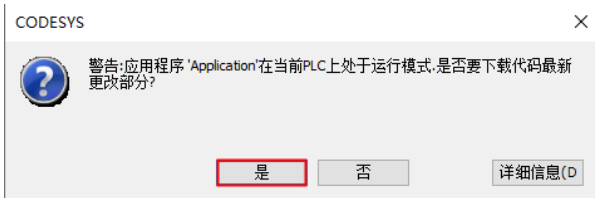
程序如下所示，编译，查看消息窗口是否报错



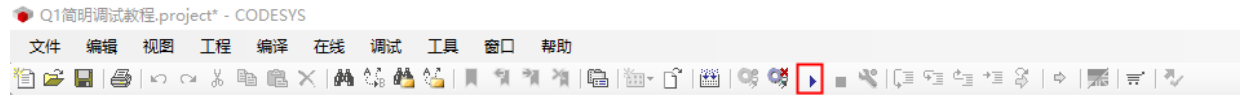
登录，下载程序到PLC中



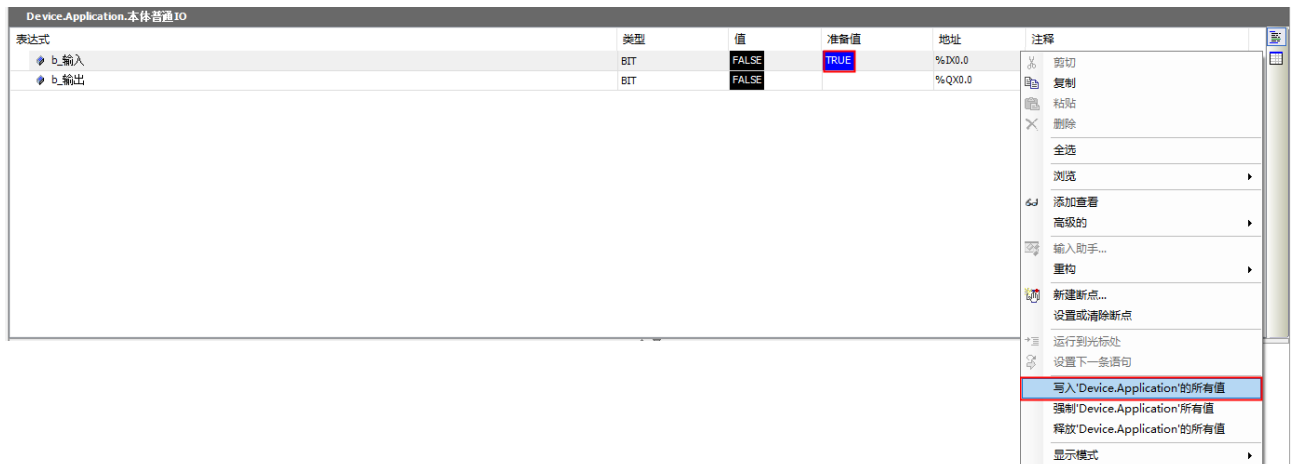
跳出弹窗，【警告】下载程序到PLC中，选择【是】



运行



【b_输入】准备值置【TRUE】，右击空白处【写入 'Device.Application' 的所有值】



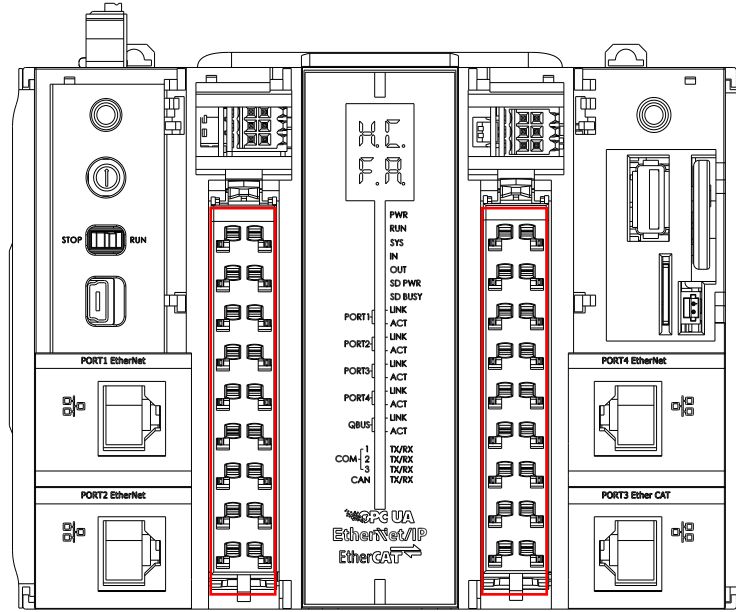
查看映射结果，观察灯板

The screenshot displays the 'GeneralIO I/O映射' (GeneralIO I/O Mapping) window. The left sidebar shows a project tree where 'GeneralIO (GeneralIO)' is selected. The main area contains a table of I/O channels. The table has columns for '变量' (Variable), '映射' (Mapping), '通道' (Channel), '地址' (Address), '类型' (Type), '当前值' (Current Value), '准备值' (Setpoint Value), '单位' (Unit), and '描述' (Description). The '当前值' column shows 'TRUE' for input channels and '0' for output channels.

变量	映射	通道	地址	类型	当前值	准备值	单位	描述
input			%IB0	BYTE	1			
Bit0			%IX0.0	BOOL	TRUE			
Bit1			%IX0.1	BOOL	FALSE			
Bit2			%IX0.2	BOOL	FALSE			
Bit3			%IX0.3	BOOL	FALSE			
Bit4			%IX0.4	BOOL	FALSE			
Bit5			%IX0.5	BOOL	FALSE			
Bit6			%IX0.6	BOOL	FALSE			
Bit7			%IX0.7	BOOL	FALSE			
Byte1			%IB1	BYTE	0			
output			%QB0	BYTE	0			
Byte0			%QB0	BYTE	1			
Bit0			%QX0.0	BOOL	TRUE			
Bit1			%QX0.1	BOOL	FALSE			
Bit2			%QX0.2	BOOL	FALSE			
Bit3			%QX0.3	BOOL	FALSE			
Bit4			%QX0.4	BOOL	FALSE			
Bit5			%QX0.5	BOOL	FALSE			
Bit6			%QX0.6	BOOL	FALSE			
Bit7			%QX0.7	BOOL	FALSE			
Byte1			%QB1	BYTE	0			

5.2 Q1自带高速输入输出

左侧为8位高速IO输入，右侧为8位高速IO输出。



8通道高速输入端子排布

his_cnt	I0	I10	his_cnt4
	I1	I11	
his_cnt1	I2	I12	his_cnt5
	I3	I13	
his_cnt2	I4	I14	his_cnt6
	I5	I15	
his_cnt3	I6	I16	his_cnt7
	I7	I17	
公共端	SS	SS	公共端

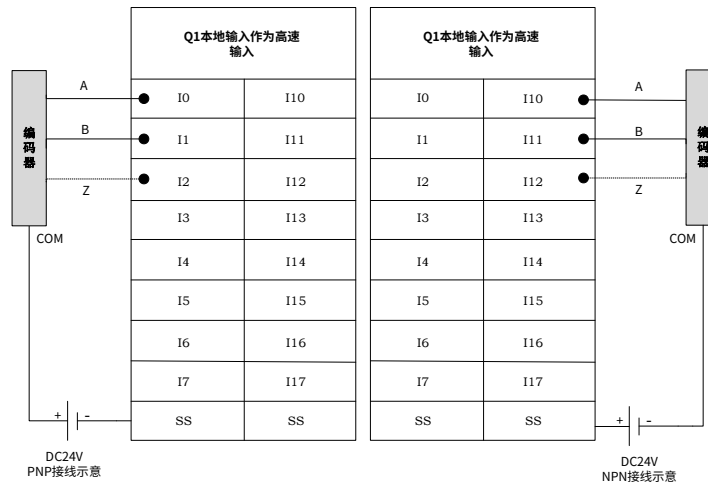
8通道高速输出端子排布

hso_axis	Q0	Q10	hso_axis4
	Q1	Q11	
hso_axis1	Q2	Q12	hso_axis5
	Q3	Q13	
hso_axis2	Q4	Q14	hso_axis6
	Q5	Q15	
hso_axis3	Q6	Q16	hso_axis7
	Q7	Q17	
公共端	COM	COM	公共端

注：整机版本V2.XX的支持4通道高速输出；V3.XX支持8通道高速输出。

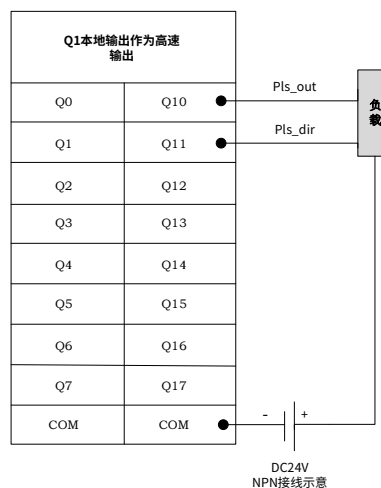
5.2.1 Q1硬件接线

本地输入作为高速输入外部接线示意图



注：在某些模式下，需要使用Z相或锁存端子，可选择空闲的任意端口，对程序中的对应参数进行配置即可。

本地输出作为高速外部接线示意图

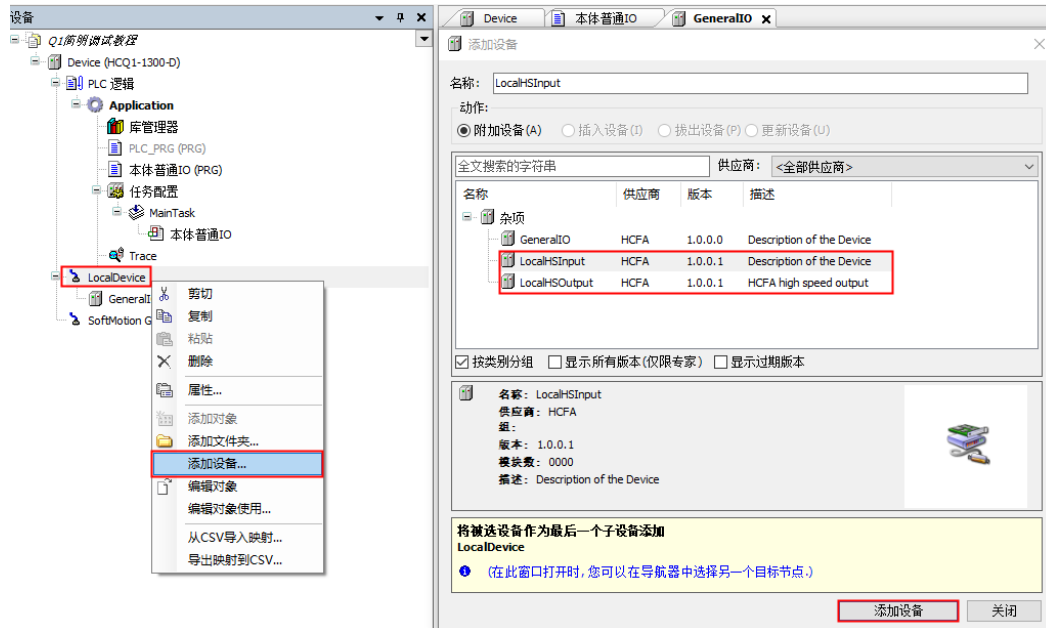


5.2.2 程序示例

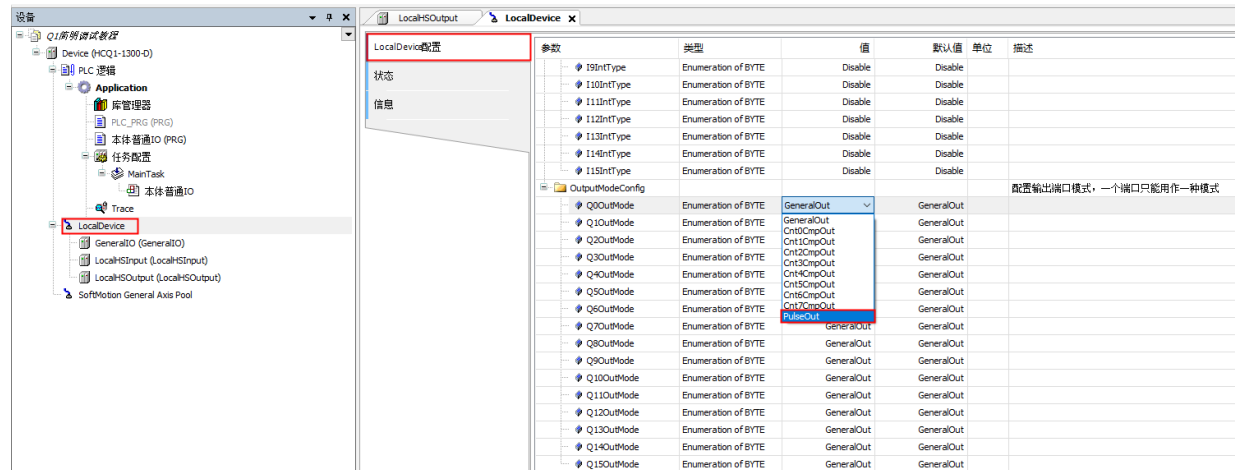
使用高速IO可实现单轴的基本定位功能，最高频率可达200k。

高速IO有多种模式，为了方便使用，对这些模式进行了二次封装，生成HCFA_HSIO_ATCLib_1.0.0.0.compiled-library (库版本可能会有升级) 库文件，封装的库文件包括普通高速输出和插补两种模式，两种模式互斥，只可同时运行一种模式 (下载的库文件需要先查看Q1的版本信息，下载对应库文件，详情查看1.1版本信息，功能块使用查看【Q1高速输入输出使用介绍】，库文件按照查看附录：描述文件安装)。

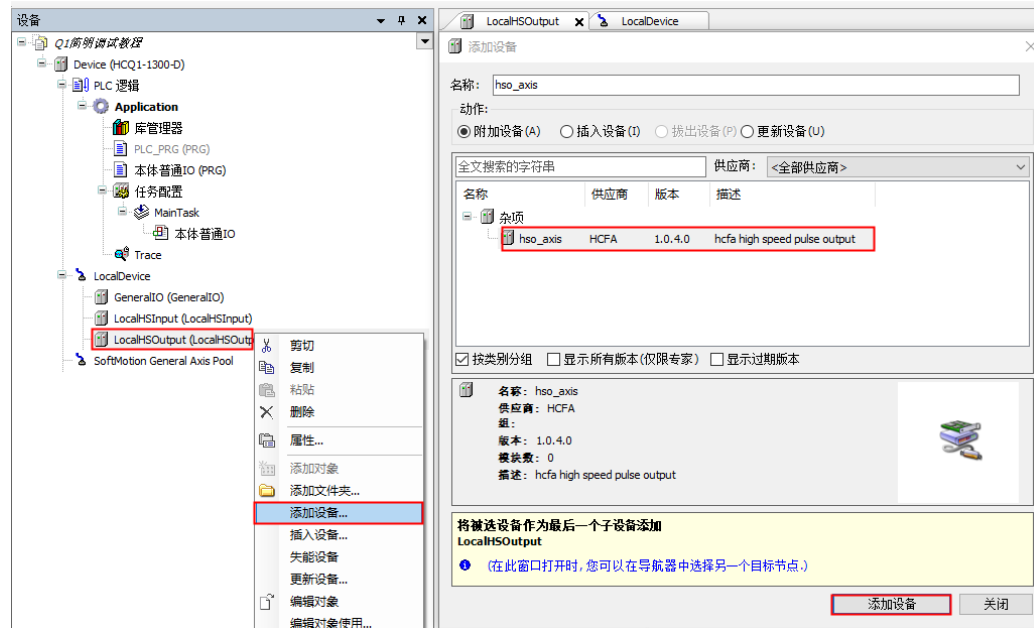
右击【LocalDevice】→【添加设备】，添加【LocalHSInput】、【LocalHSOutput】



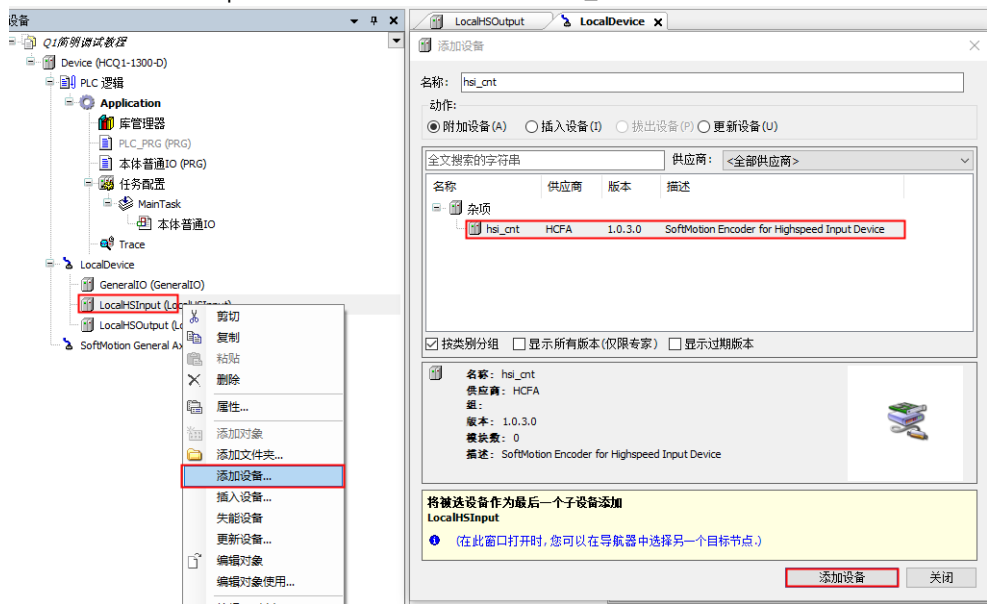
在使用高速输出前，需要在【LocalDevice】→【LocalDevice配置】中将对应通道的输出方式选择为【PulseOut】



使用功能块还需要以下配置，右击【LocalHSInput】在添加设备页面中选择【hsi_cnt】，单击【添加按钮】



右击【LocalHSOutput】在添加设备页面中选择【hsi_axis】，单击【添加按钮】



第六章 扩展模块

6.1 硬件接线介绍

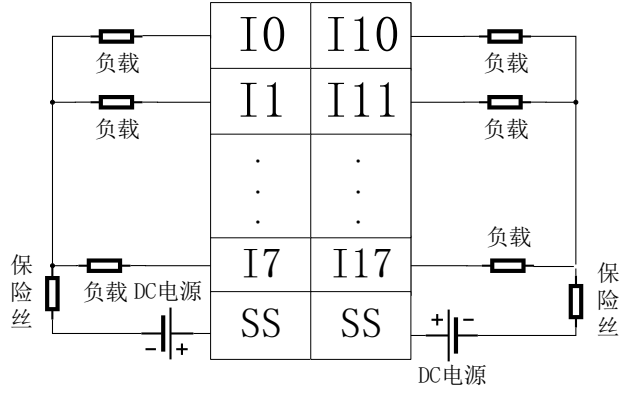
Q1 V301以上版本支持在右侧扩展本地IO（金属片），最大功率为16W，下表为Q1支持的扩展模块的功率（数据都是按照上浮10%~15%计算，选型时候不要把功率卡的太死，金属片传输总线电流信号会随着时间产生一定电阻，从而消耗一部分功率）

模块型号	参考功率
LocalEtherCATDevice	16W
HCQX-EC-D	16W
HCQX-EC-D2	16W
HCQX-ID16-D	0.78W
HCQX-ID16-D2	1.35W
HCQX-OD16-D	1.32W
HCQX-OD16-D2	1.32W
HCQX-MD-D	1.032W
HCQX-MD-D2	1.35W
HCQX-AD04-D	1.044W
HCQX-DA04-D	1.056W

6.1.1 数字量模块硬件接线

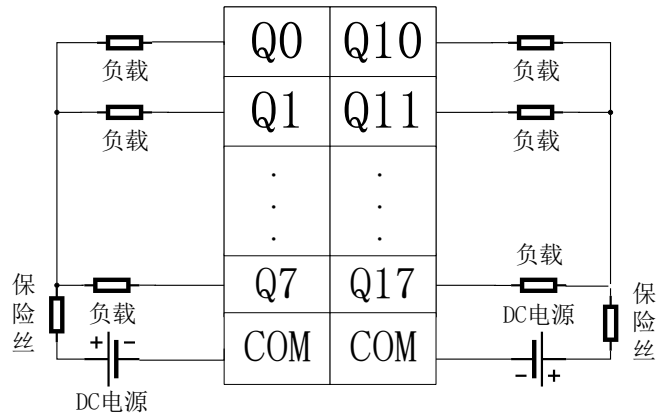
■ 数字量输入外部接线示意图

输入接法有NPN和PNP型



■ 数字量输出外部接线图

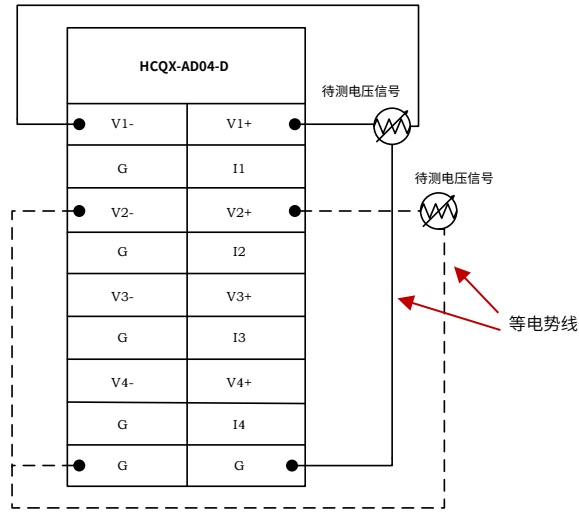
标准版本输出为NPN型，可定制PNP型。



6.1.2 模拟量模块硬件接线

模拟量需要外部24V供电作为负载供电。

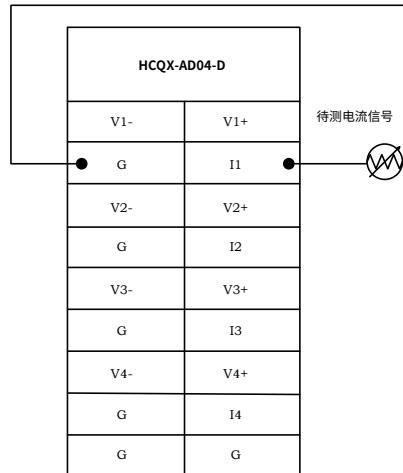
■ 模拟量电压输入外部接线



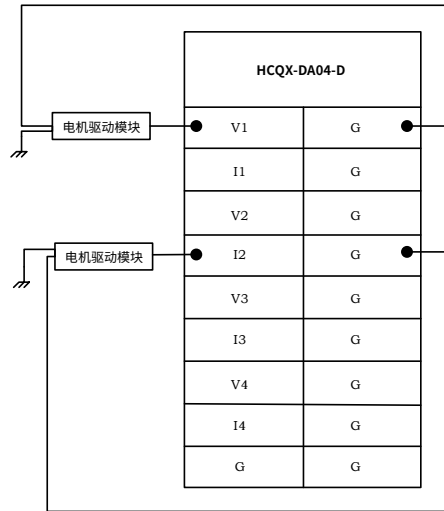
注：1. 模拟量输入输出模块外部需要24V直流供电

2. 模拟量输入模块有差分模式和单端模式，直线为电压差分输入接线，虚线为电压单端输入接法。

■ 模拟量电流输入外部接线



■ 模拟量输出外部接线

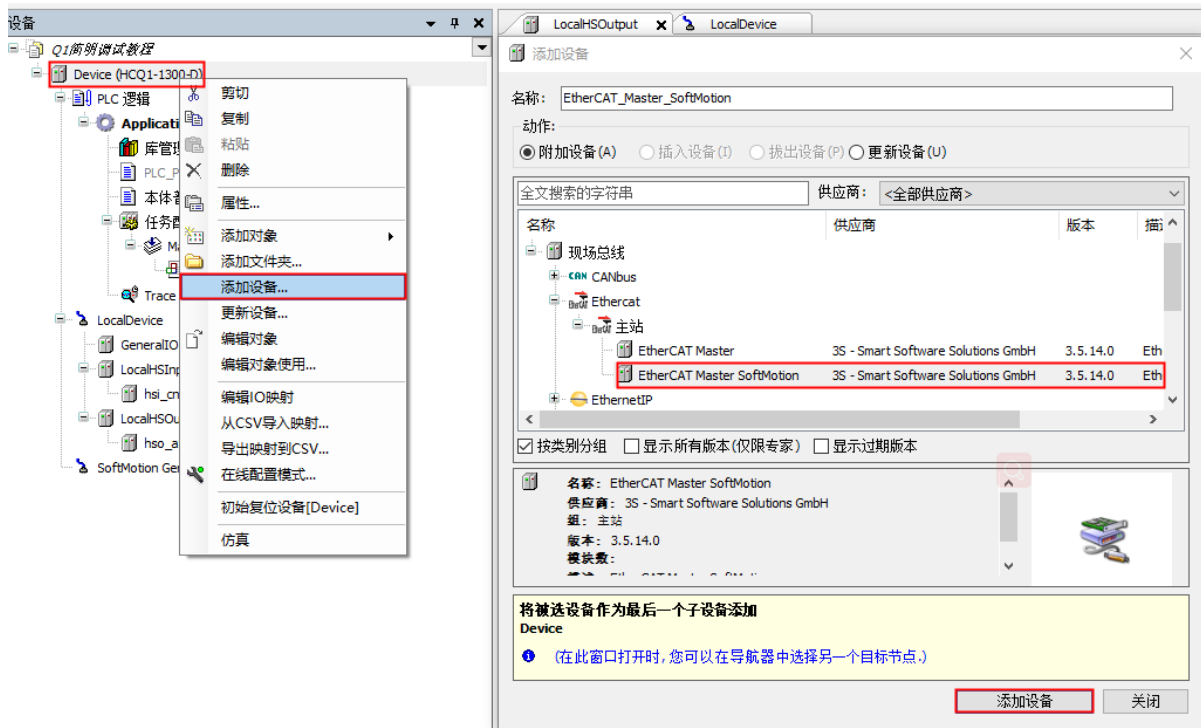


6.2 程序示例

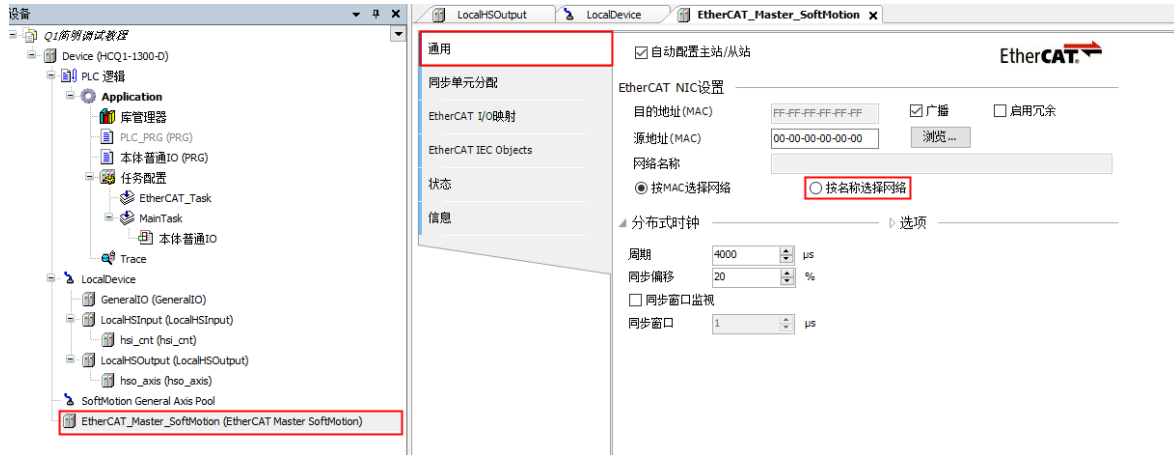
6.2.1 EtherCAT主站

扩展模块使用的通讯模式都是EtherCAT总线型，所以用户需要添加EtherCA主站

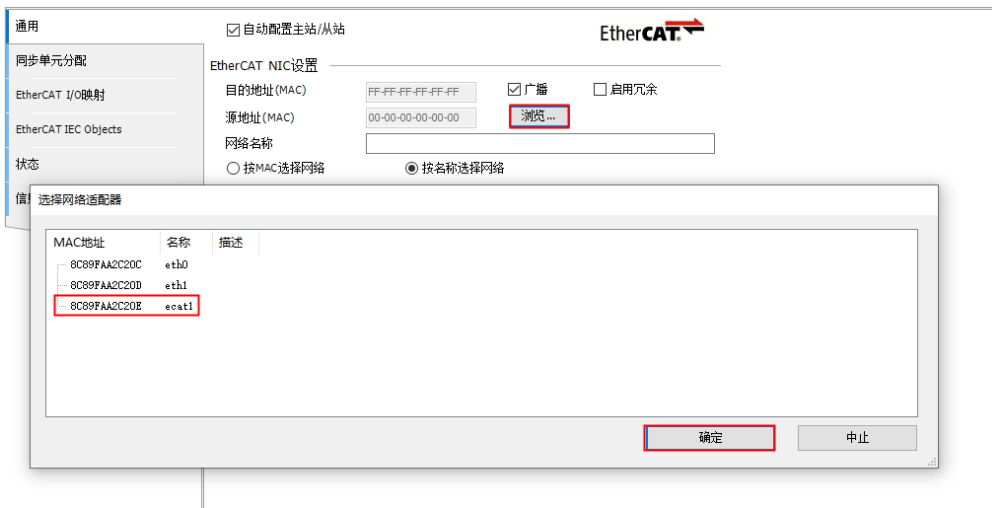
在左侧树形菜单找到【Device】→【添加设备】→【EtherCAT_Master_SoftMotion】



双击左侧树形菜单【EtherCAT_Master_SoftMotion】，在右侧【通用】下找到【按名称选择网络】

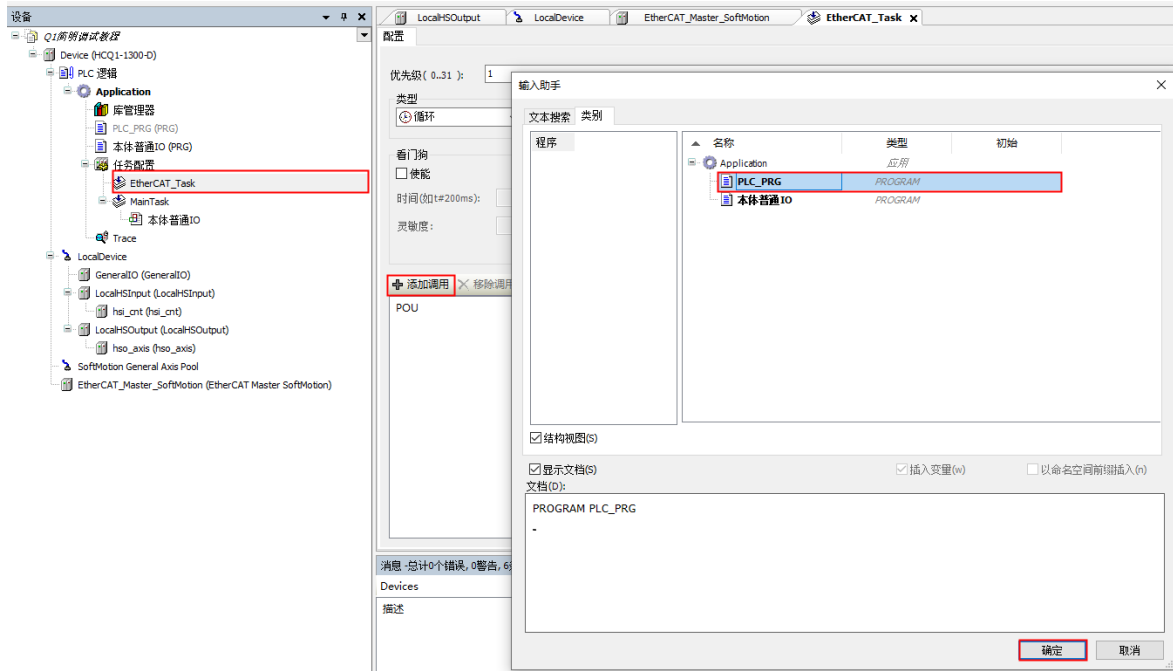


点击【源地址 (MAC)】→【浏览】，选择正确的网络适配器，点击【确定】



添加主站后，会自动生成【EtherCAT Task】，若需要使用运动控制程序，则需要要在【EtherCAT Task】的任务下调用

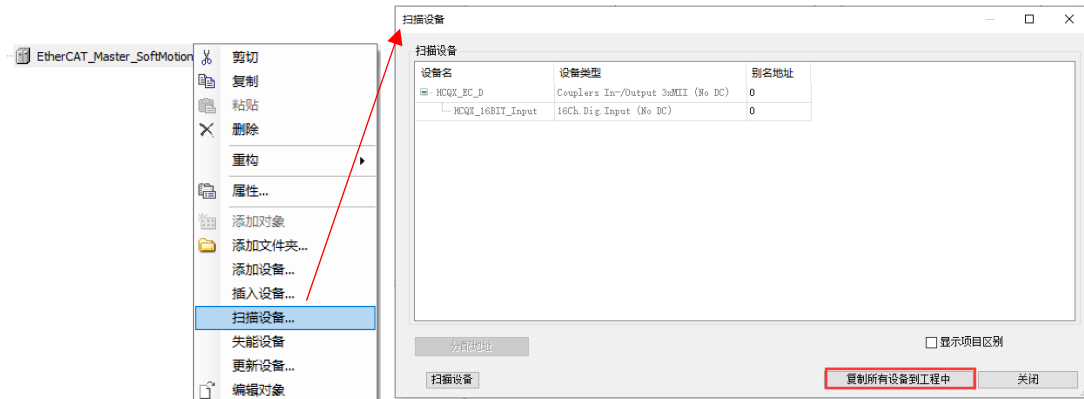
单击【EtherCAT Task】→【添加调用】，选择要调用的程序



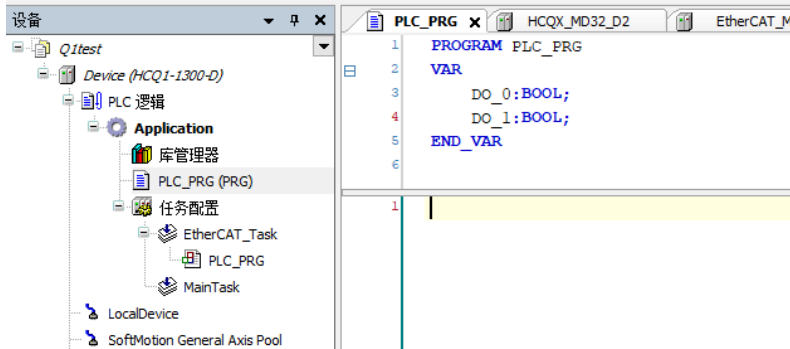
6.2.2 数字量模块程序示例

本示例以HCQ1-1300-D2 CPU单元+HCQX-EC01-D耦合器+HCQX-MD16-D2数字量混合模块搭建的系统作为示例进行说明。

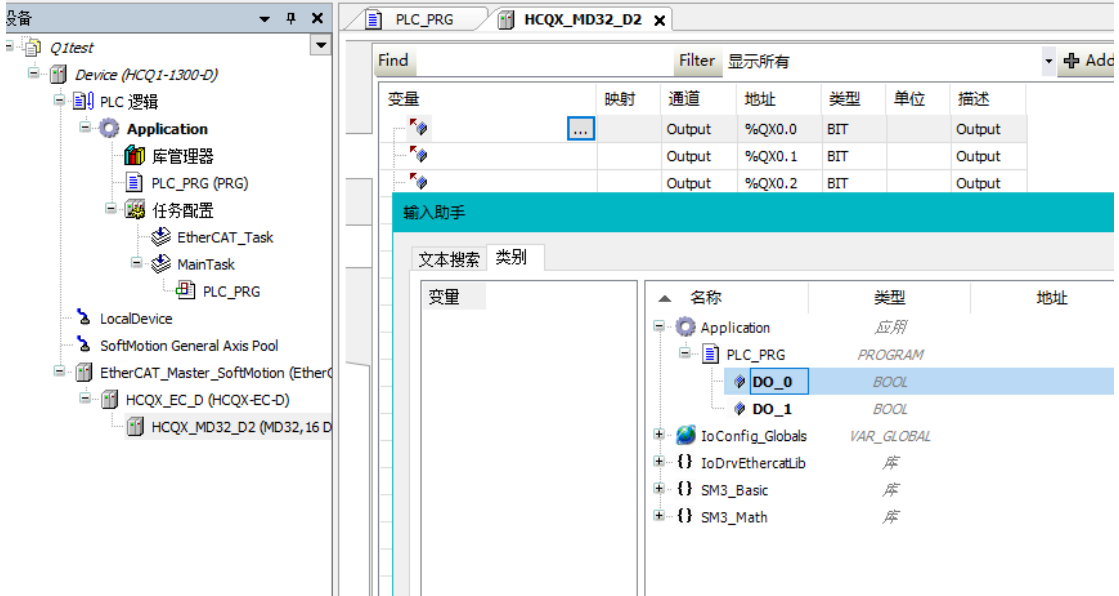
右击【EtherCAT_Master_SoftMotion】选择扫描设备，正常工作并建立通讯的模块，可以在【扫描设备】窗口中找到并通过右下角【复制所有设备到工程中】，将扫描到的模块添加到工程中



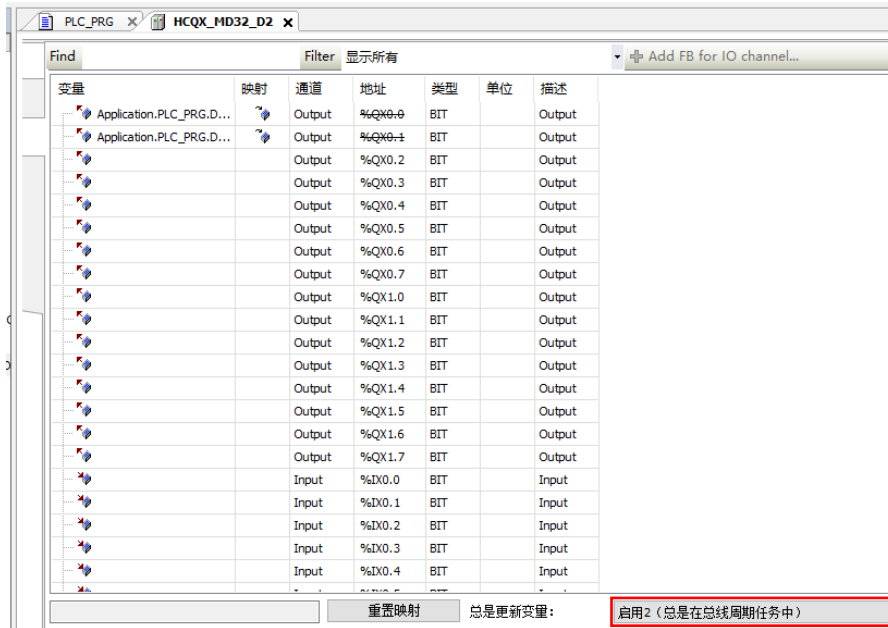
在PLC_PRG中用ST编程语言定义两个BOOL类型的变量，并映射到对应的输出变量上。



将【HCQX-MD16-D2】的两组通道分别映射到程序中定义好的变量【DO_0】和【DO_1】，如下



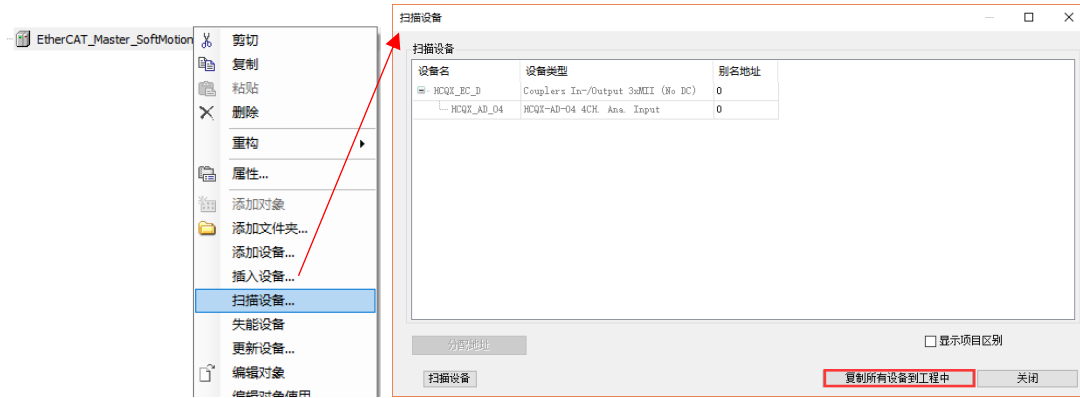
将右下角的【总是更新变量】修改为【启用2（总是再总线周期任务中）】（在无程序测试的情况下需要将IO刷新周期选择为：总是在总线周期任务中，否则未映射变量的IO不会刷新，映射了程序变量则当前IO会随程序变量更新时间刷新）



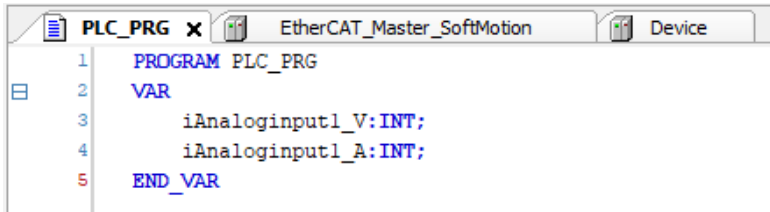
编译无错误后，登录并运行程序，在给对应输出通道输出信号时，能检测到通道有响应信号输出（外部接线正确的前提下，若无响应，查看接线是否出现问题）。

6.2.1 模拟量模块程序示例

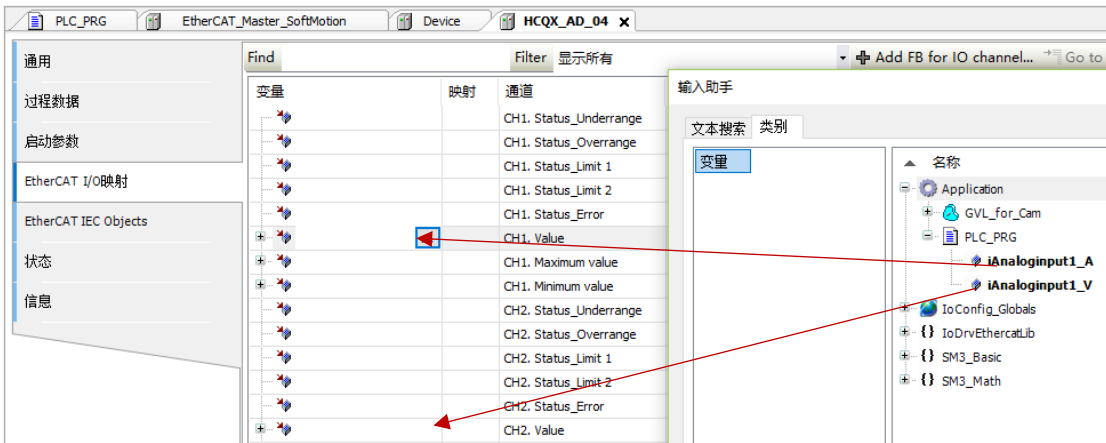
右击【EtherCAT_Master_SoftMotion】选择扫描设备，正常工作并建立通讯的模块，可以在【扫描设备】窗口中找到并通过右下角【复制所有设备到工程中】将扫描到的模块添加到工程中



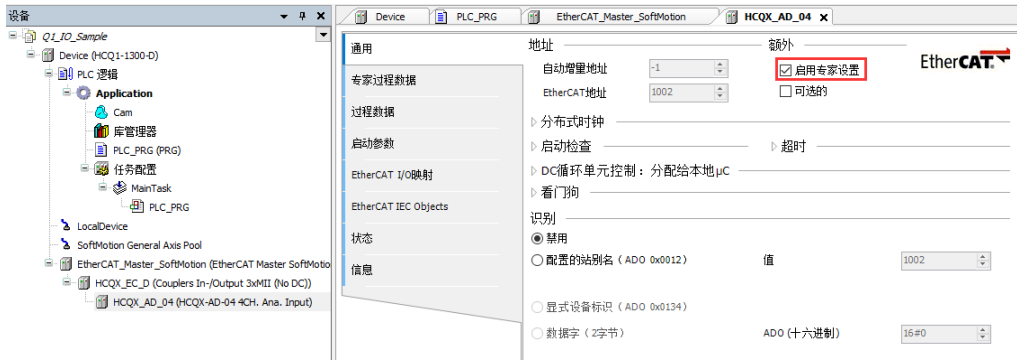
在PLC_PRG中用ST编程语言定义两组INT类型的变量，并映射到对应的模拟量输入上



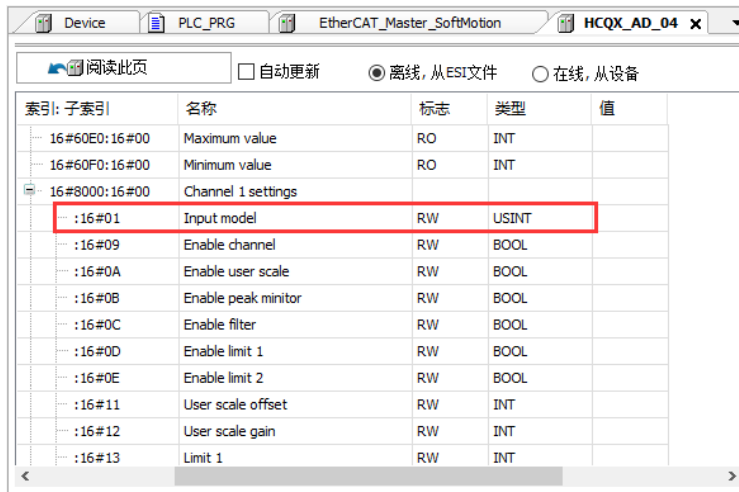
将【HCQX-AD04-D】的前两组通道分别映射到程序中定义好的变量【iAnaloginput1_V】和【iAnaloginput1_A】，如下：



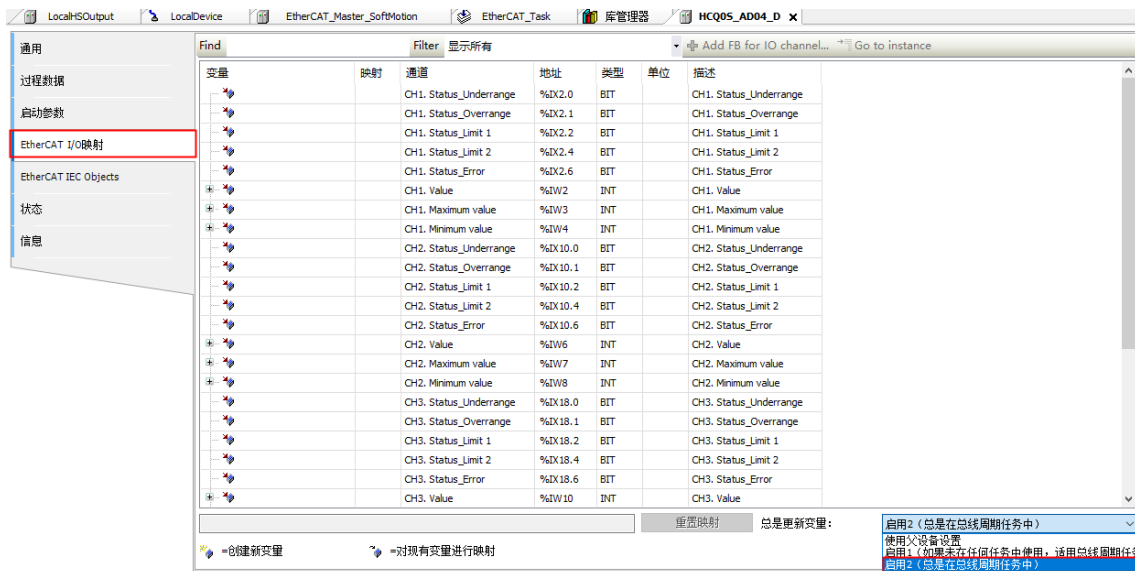
编译无错误后，登录并运行程序，按照需要测量的信号类型设定当前通道工作模式，首先选中需要设置的通道，在【通用】中勾选【启用专家模式】



在CoE在线页面【16#0x80n: 01】下设置模块的工作模式，并确认使用中的通道【16#0x80n: 09】处于开启状态，更详细的参数设置请参考附录：对象字典总表的说明。



修改【总是更新变量】为【启用2（总是在总线周期任务中）】（在无程序测试的情况下需要将IO刷新周期选择为：总是在总线周期任务中，否则未映射变量的IO不会刷新，映射了程序变量则当前IO会随程序变量刷新时间刷新）



配置完成后，在【EtherCAT I/O映射】下查看当前输入通道的数值。

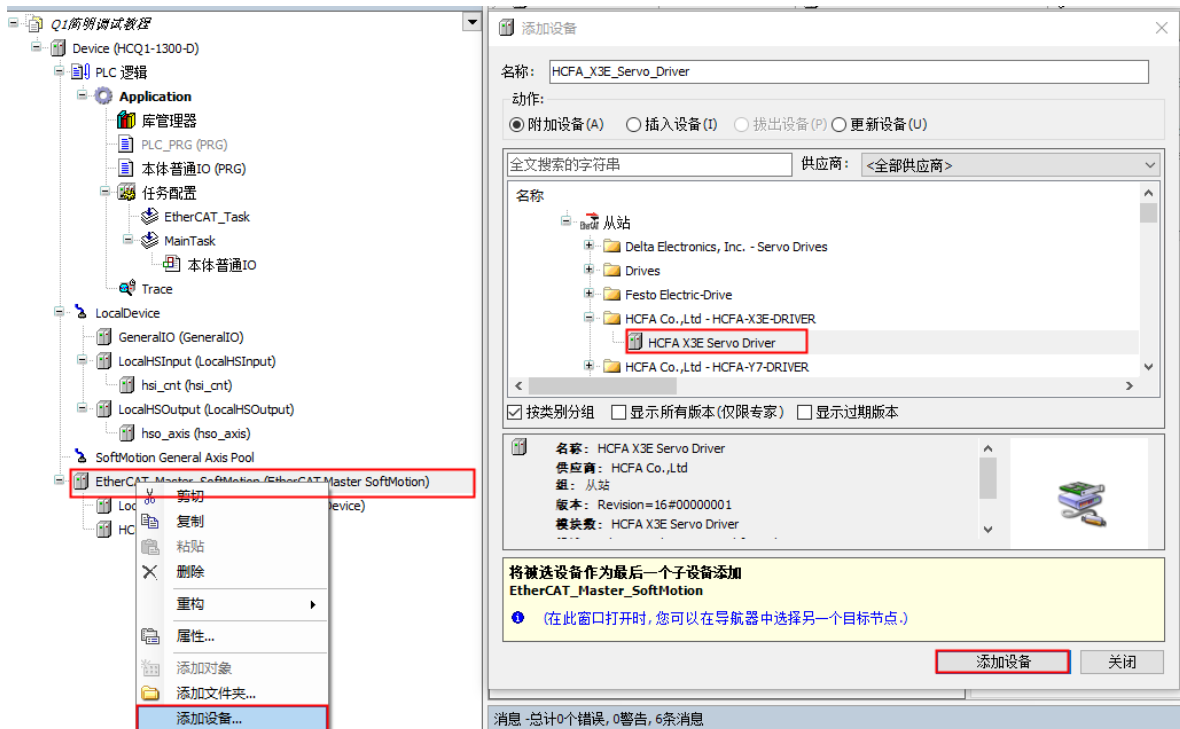
变量	映射	通道	地址	类型	单位	描述
CH1.Status_Underrange			%IX0.0	BIT		CH1.St...
CH1.Status_Ovrange			%IX0.1	BIT		CH1.St...
CH1.Status_Limit 1			%IX0.2	BIT		CH1.St...
CH1.Status_Limit 2			%IX0.4	BIT		CH1.St...
CH1.Status_Error			%IX0.6	BIT		CH1.St...
CH1.Value			%IW1	INT		CH1.Va...
CH1.Maximum value			%IW2	INT		CH1.M...
CH1.Minimum value			%IW3	INT		CH1.M...
CH2.Status_Underrange			%IX8.0	BIT		CH2.St...
CH2.Status_Ovrange			%IX8.1	BIT		CH2.St...
CH2.Status_Limit 1			%IX8.2	BIT		CH2.St...
CH2.Status_Limit 2			%IX8.4	BIT		CH2.St...
CH2.Status_Error			%IX8.6	BIT		CH2.St...
CH2.Value			%IW5	INT		CH2.Va...
CH2.Maximum value			%IW6	INT		CH2.M...
CH2.Minimum value			%IW7	INT		CH2.M...
CH3.Status_Underrange			%IX16.0	BIT		CH3.St...
CH3.Status_Ovrange			%IX16.1	BIT		CH3.St...
CH3.Status_Limit 1			%IX16.2	BIT		CH3.St...
CH3.Status_Limit 2			%IX16.4	BIT		CH3.St...
CH3.Status_Error			%IX16.6	BIT		CH3.St...
CH3.Value			%IW9	INT		CH3.Va...
CH3.Maximum value			%IW10	INT		CH3.M...
CH3.Minimum value			%IW11	INT		CH3.M...

第七章 运动控制

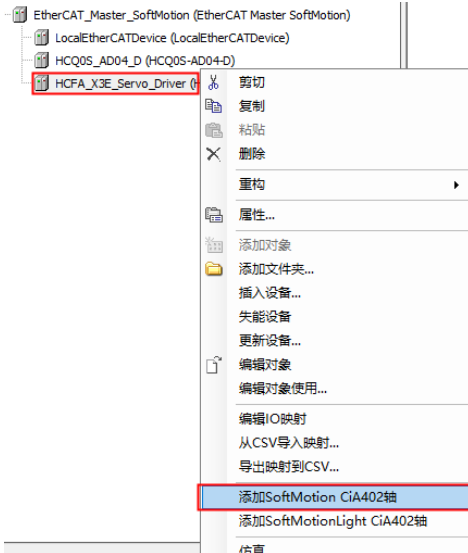
7.1 运动控制配置

以X3E为例，通讯方式为EtherCAT总线型，所以添加方式与扩展模块相似（若已添加扩展模块，可忽略添加主站的步骤，主站的添加查看6.2.1EtherCAT主站）

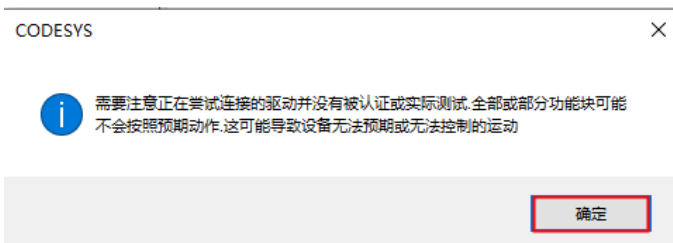
在【EtherCAT_Master_SoftMotion】右击选择【添加设备】，此时添加用户所需要的驱动器，本次示例使用【HCFA X3E Servo Drive】，选中后点击【添加设备】



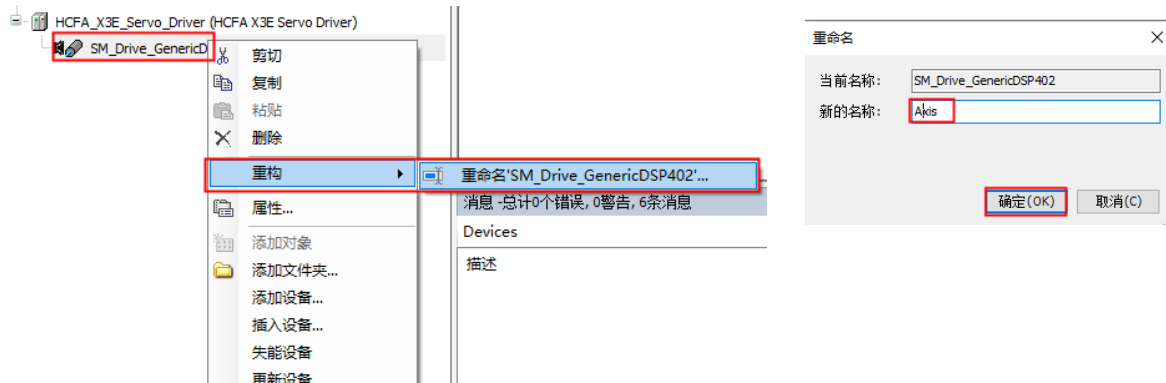
右击【HCFA X3E Servo Drive】，选择【SoftMotion CiA 402轴】，



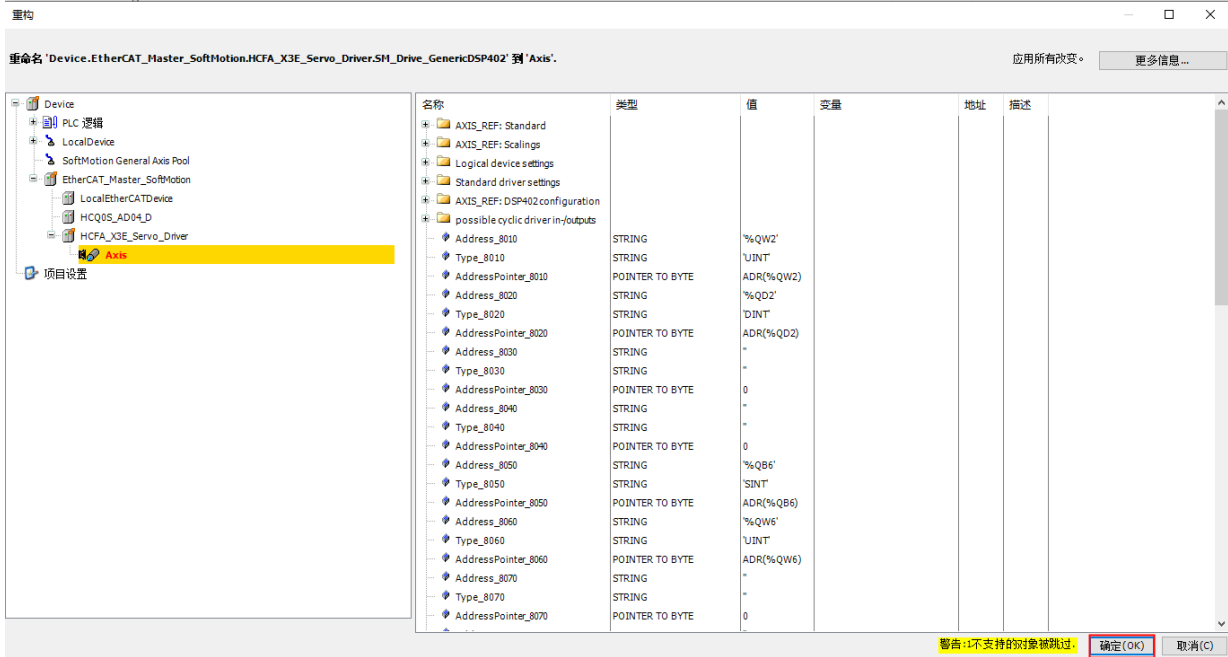
跳出弹窗，提示查看连接驱动，点击【是】



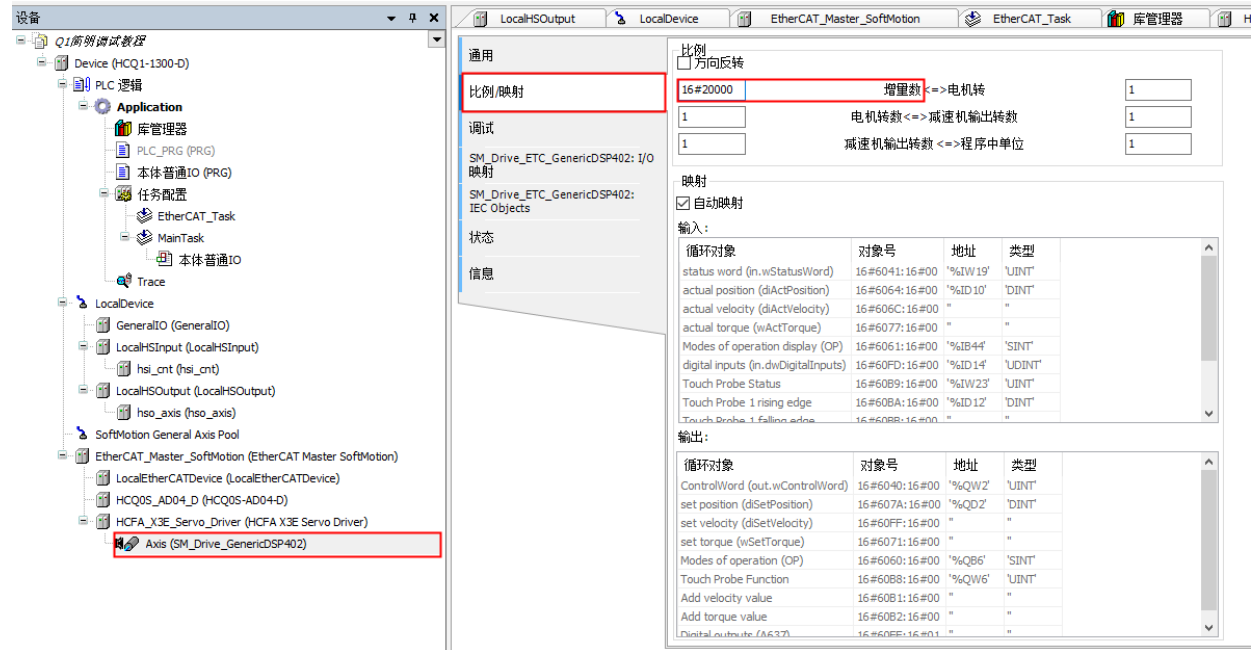
为了方便后续调用轴，可更改轴名字，右击【SM_Drive_GenericDSP402】→【重构】→【重命名】，重命名为【Axis】



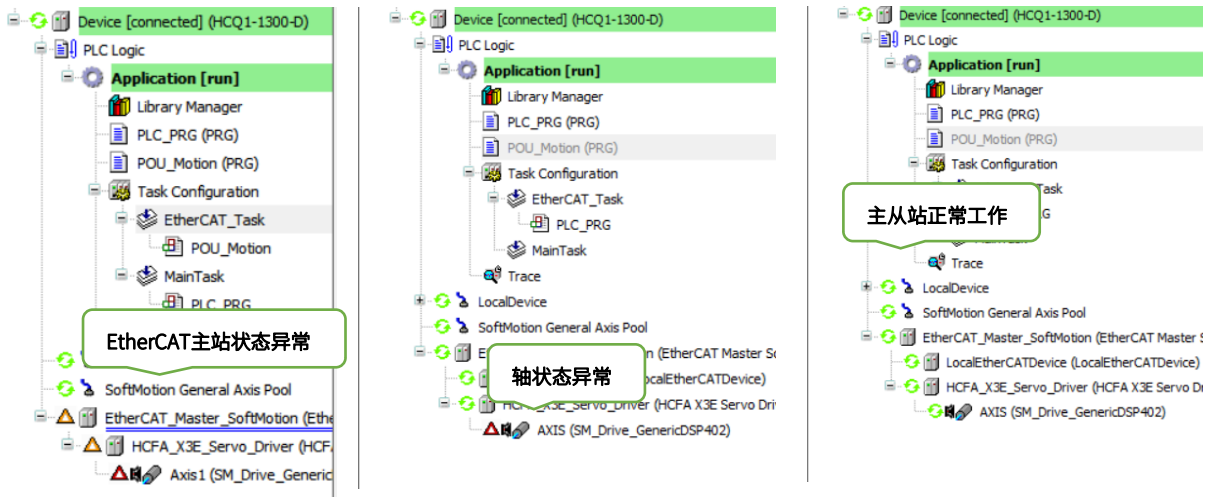
系统会自动完成轴变量重命名后的映射，点击【确定】完成轴的重命名



由于X3E为17位编码器，修改增量为16#20000，双击【Axis】，选择【比例映射】，修改【增量数】



完成硬件配置的修改之后，登录并运行程序，等待驱动器和轴状态正常后，则可开始启动运动

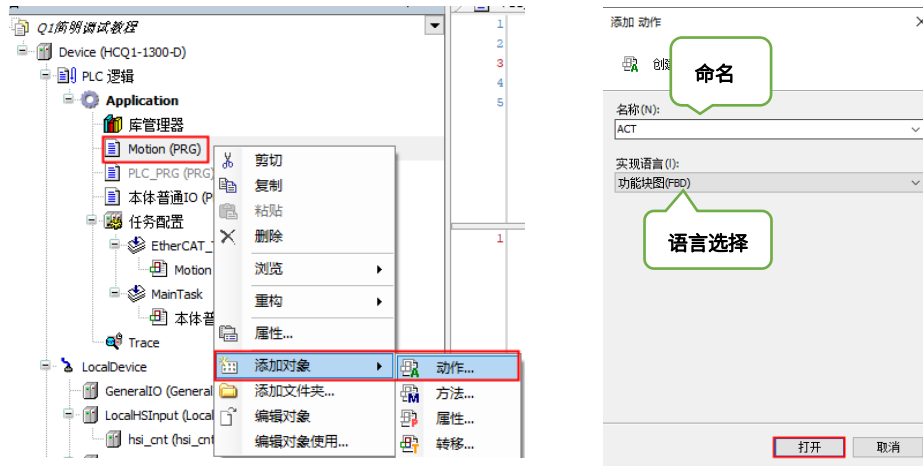


7.2 运动控制程序示例

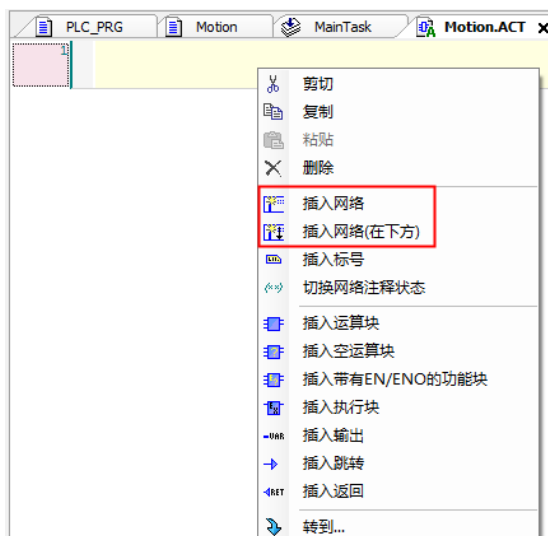
7.2.1 单轴程序示例

新建程序（具体步骤查看3.2.1POU的程序添加），取名为【Motion】（英文方便用户再Trace中跟踪），程序需要在【EtherCAT Task】中调用，调用步骤参考3.1.2 任务。

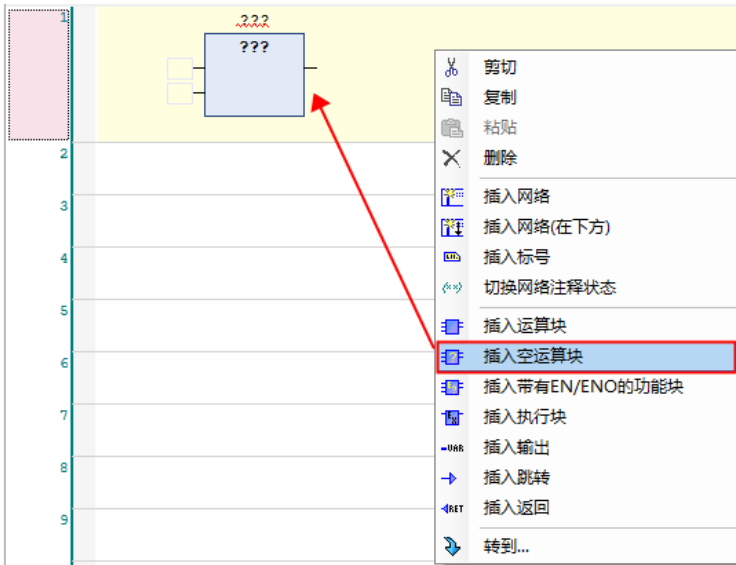
为方便用户使用功能块，右击【Motion】→【添加对象】→【动作】，命名后选择【功能块图】，点击【确定】



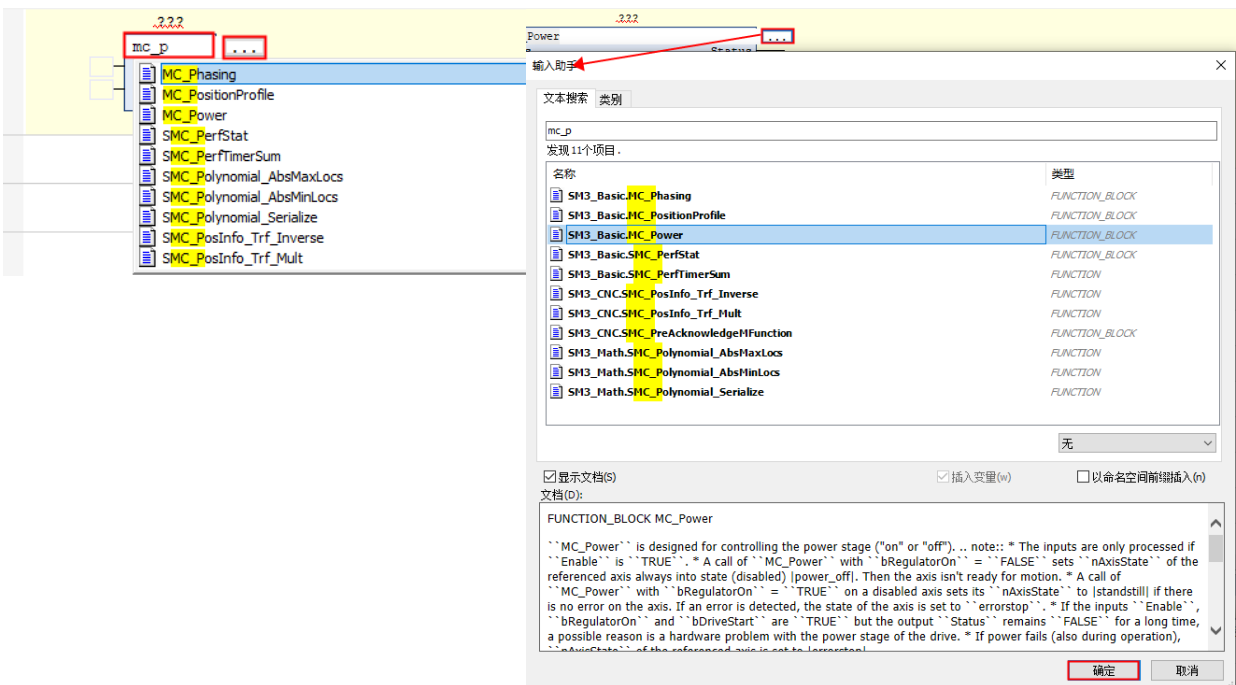
插入网络



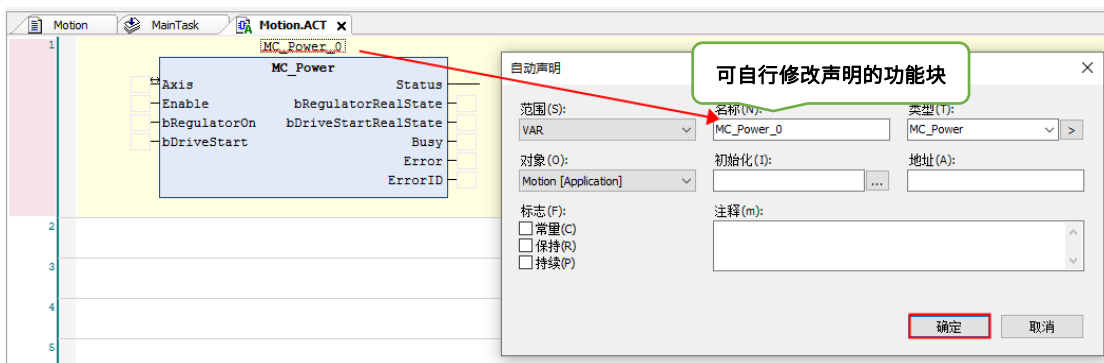
添加功能块，选择【插入空运输块】（方便自动声明）



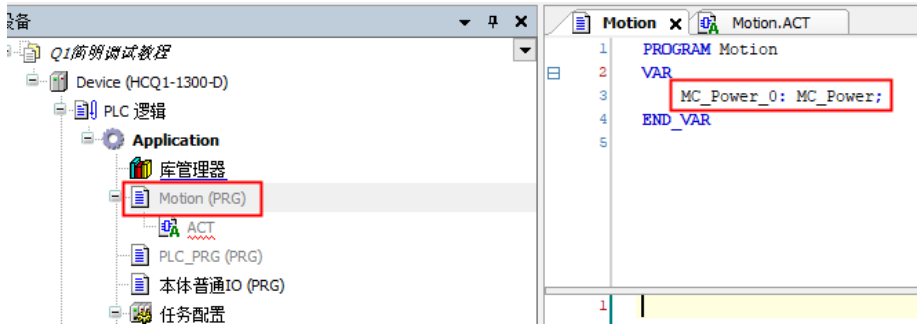
在功能块中直接输入你需要调用的功能块（快速索引设置可查看2.1.4），或者使用输入助手查找



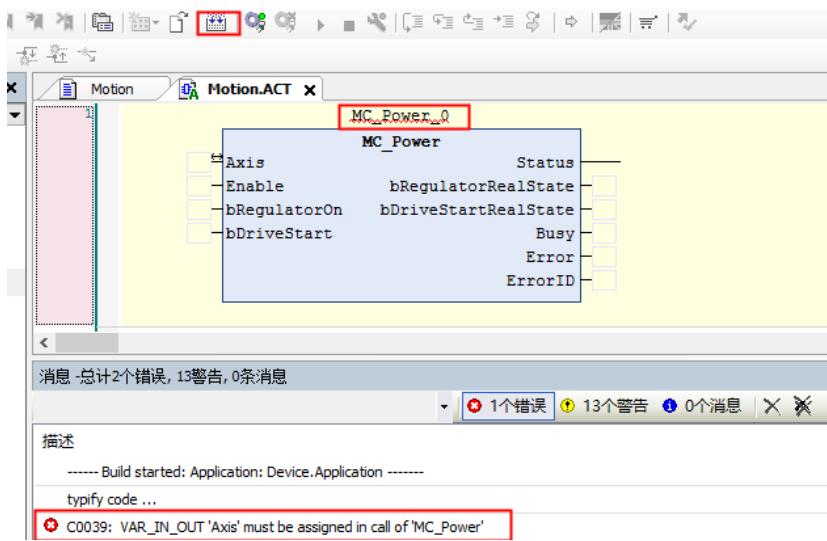
添加所需功能块后，点击空白处，将会自动在程序中声明用户调用的功能块，点击【确定】



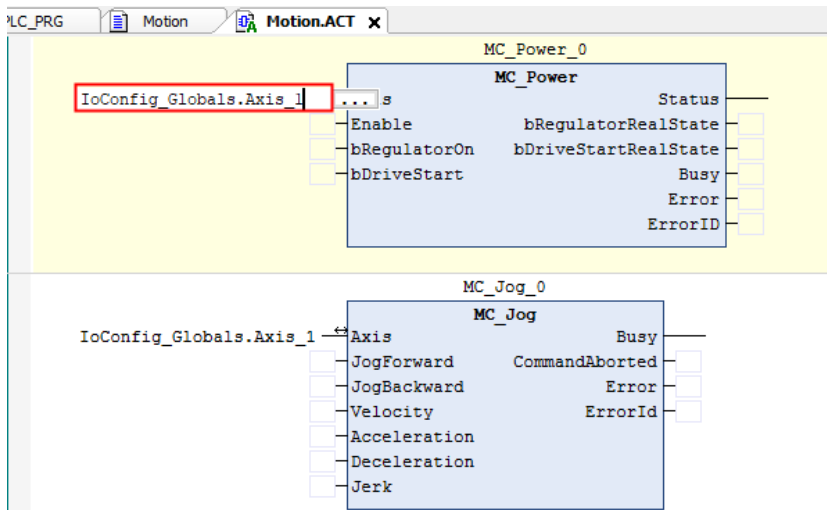
可以回到【Motion】程序中查看变量声明区域声明的功能块



回到【Motion.ATC】中，可以看到功能块下方标红色波浪线，编译后可查看报错窗口，表示需要链接轴

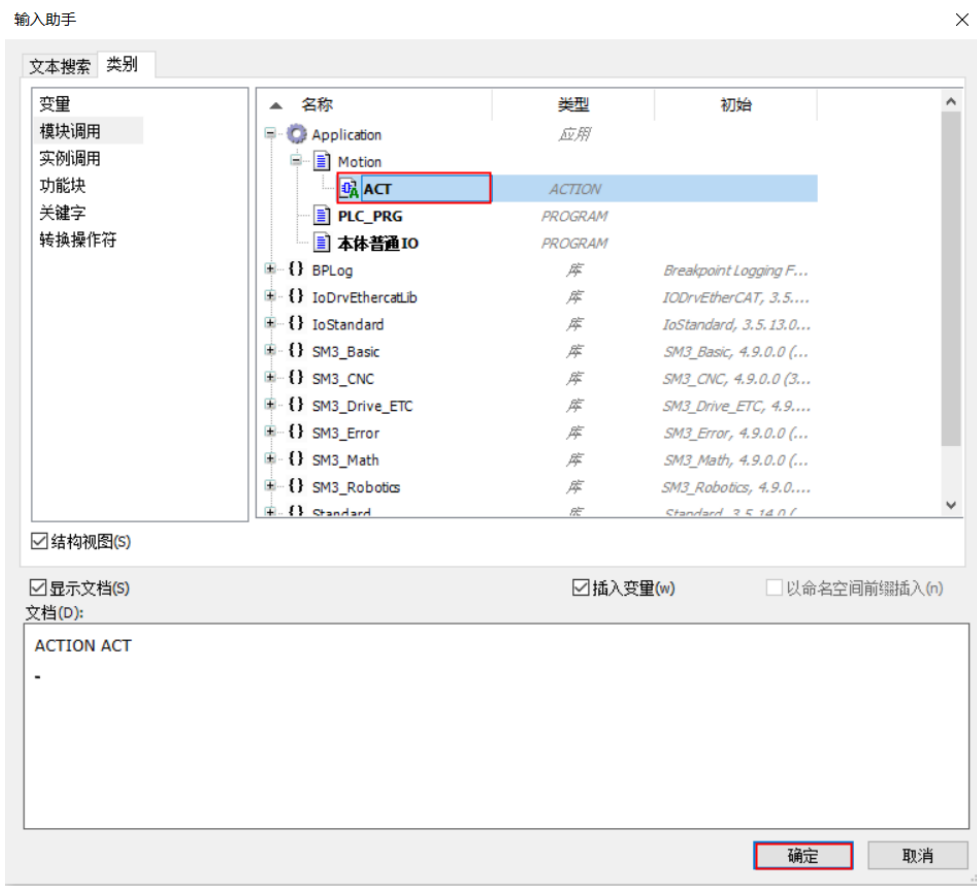


功能块引脚链接轴



此时【Motion.ATC】并没有在程序中被调用，所以程序运行时并不会刷新运行的（可自行登录尝试运行）。

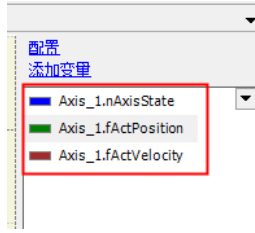
回到【Motion】程序中，在主程序窗口按F2（输入助手快捷键），选择【模块调用】→【Application】→【Motion】→【ACT】，点击确定



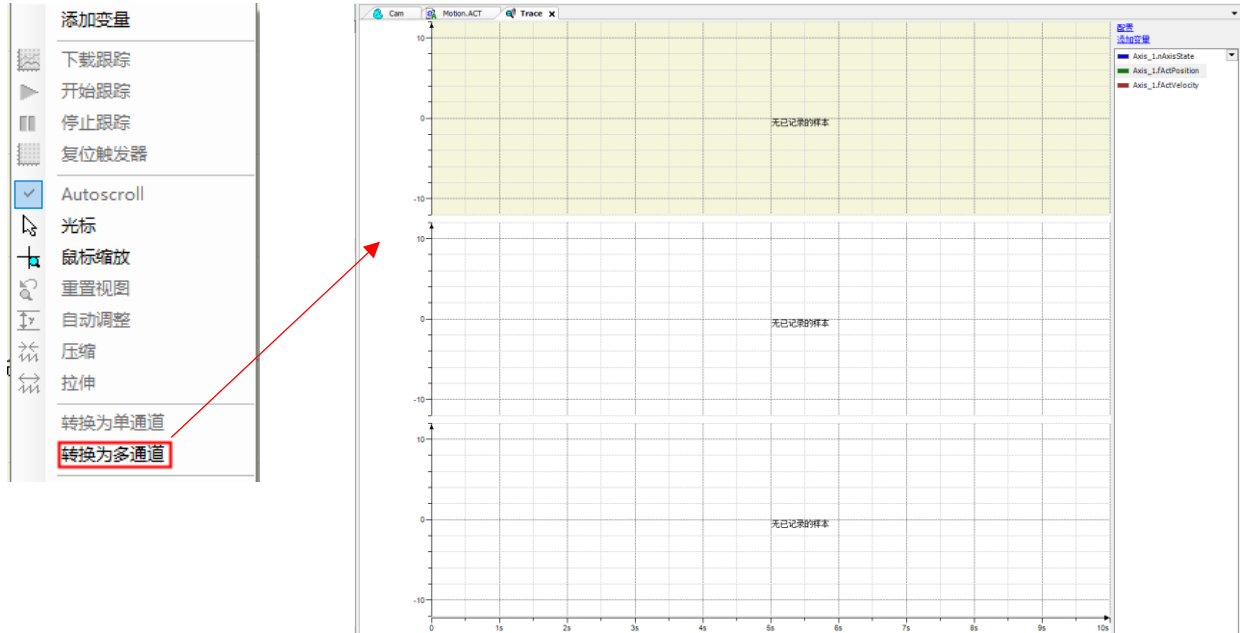
此时消息窗口会有报错，表示需要加上【;】



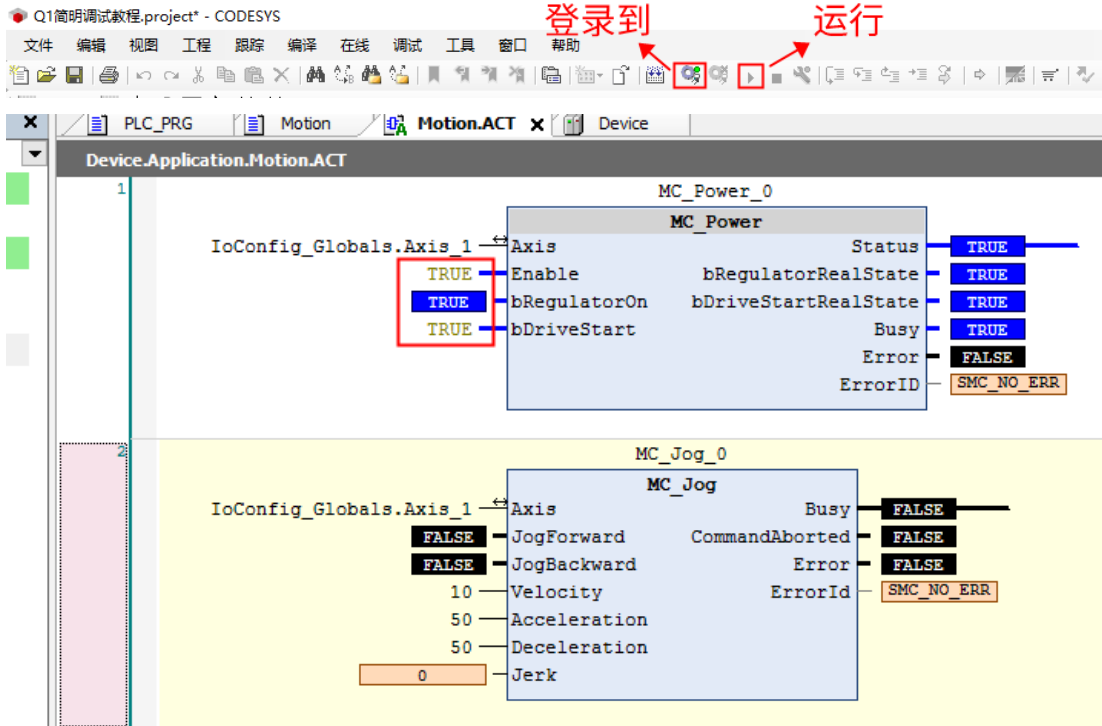
为方便用户追踪，可添加Trace（可查看4.2.3Trace）。可追踪轴状态，当前位置，当前速度等



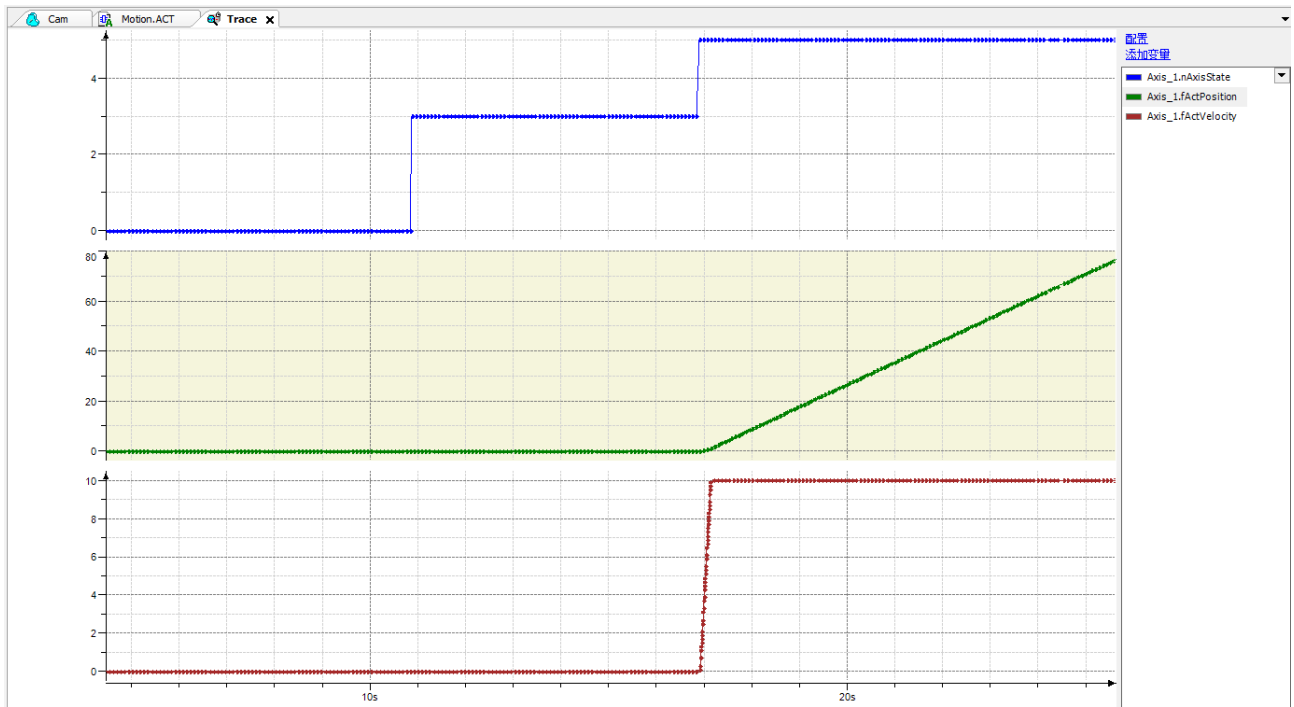
由于是多个变量，可以分多个图表，右键空白处



登录运行，给轴使能，【MC_Power】功能块引脚【Enable】【bRegulatorOn】【bDriveStart】置TRUE，运动控制功能块需要给定加减速度，不然轴会报错，处于【errorstop】状态（可查看附录：轴状态转移图）



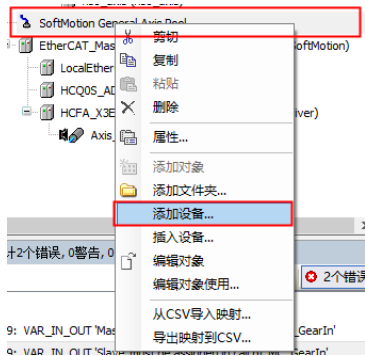
完成对Axis1的使能之后，通过【Jogforwwad】置TRUE对轴正向点动，通过给【Jogforwwad】置TRUE对轴反向点动，其他轴功能块的编辑和使用可以直接参考CODESYS在线帮助：<https://help.CODESYS.com/>，也可以参考【运动控制手册】，运动追踪可查看Trace是否有异常



7.2.2 多轴程序示例

多轴运动需要两轴及以上，此处可添加【SoftMotion General Axis Pool】（通用轴池），SoftMotion驱动器接口是一个标准接口，可用于在IEC程序中链接，配置和寻址驱动器硬件。通过将不同的硬件映射到一个接口，可以轻松交换驱动器并重复使用IEC程序。该接口将驱动器耦合到I/O映射，并负责更新所需的运动数据并将其传输到驱动器控件。虚拟驱动器

右击【SoftMotion General Axis Pool】→【添加设备】

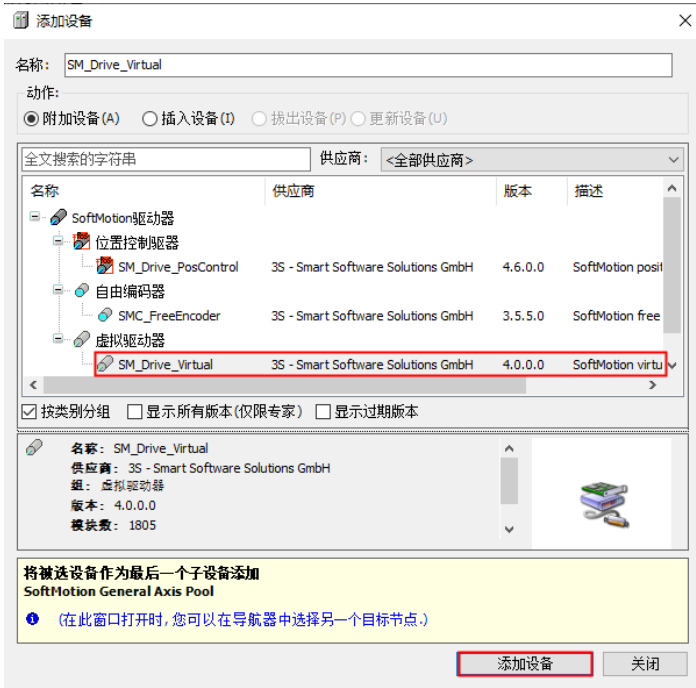


位置控制器：使用SM_Drive_PosControl驱动器控制，可以运行CODESYS轴的位置控制。要求是由设置的速度控制并返回其当前位置的设备。例如，它可以是具有位置反馈的速度控制设备（变频器）。

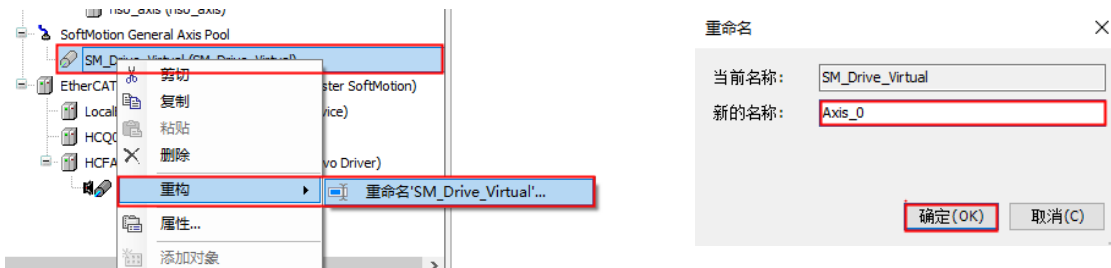
自由编码器：使用SMC_FreeEncoder集成未永久耦合到I/O或硬件的编码器。将编码器的输入值分配给变量<FREE_ENCODER_AXIS>.diEncoderPosition。这可以作为IEC代码或通过映射输入数据的存储器来实现。

虚拟驱动器：虚拟驱动器SM_Drive_Virtual是软件中的模拟驱动器。可以在没有连接硬件的情况下测试程序或实现扩展功能。这些类型的功能包括，例如，轴运动的控制。

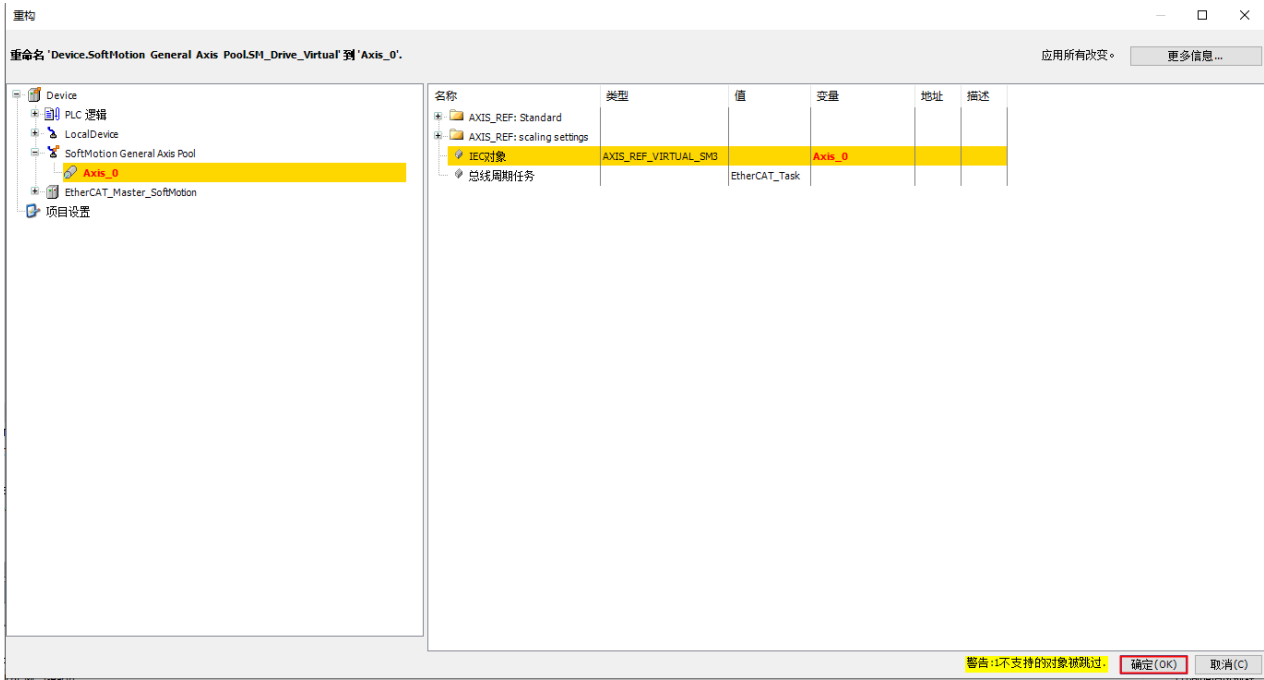
此处选择【虚拟驱动器】→【SM_Drive_Virtual】，点击【添加设备】



为方便后续功能块链接链接,可以修改轴名称为,右击【SM_Drive_Virtual】→【重构】→【重命名SM_Drive_Virtual】,修改名称为【Axis_0】,点击【确定 (OK)】

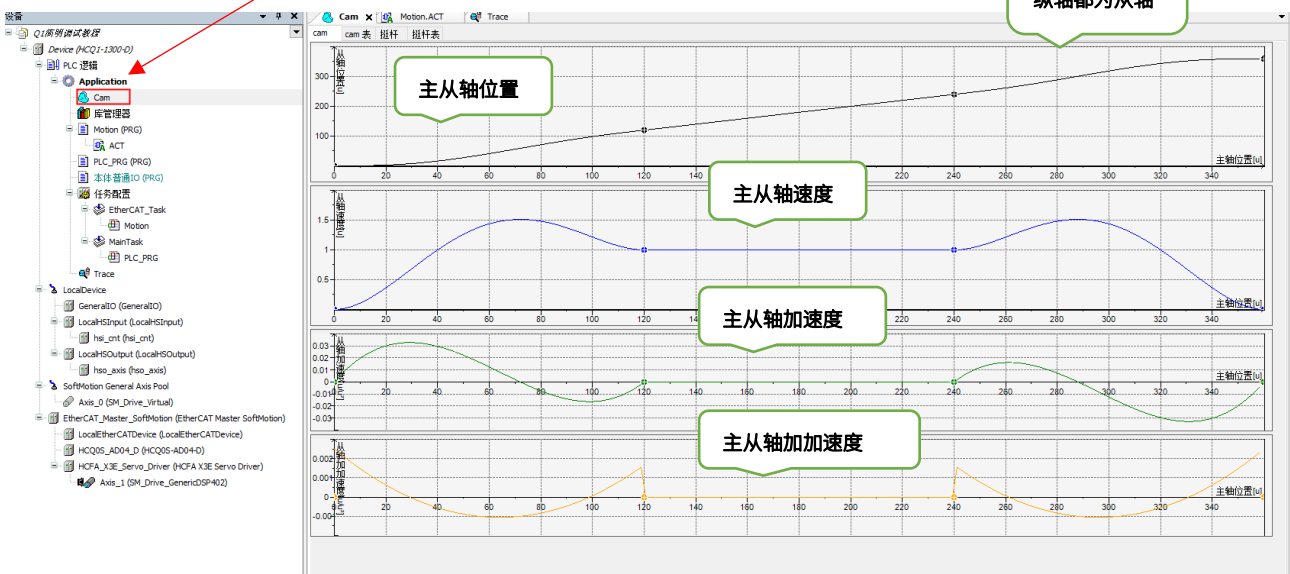
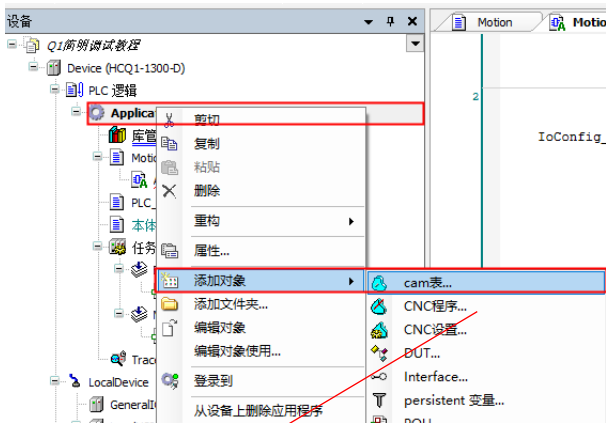


系统会自动完成轴变量重命名后的映射，点击【确定】完成轴的重命名

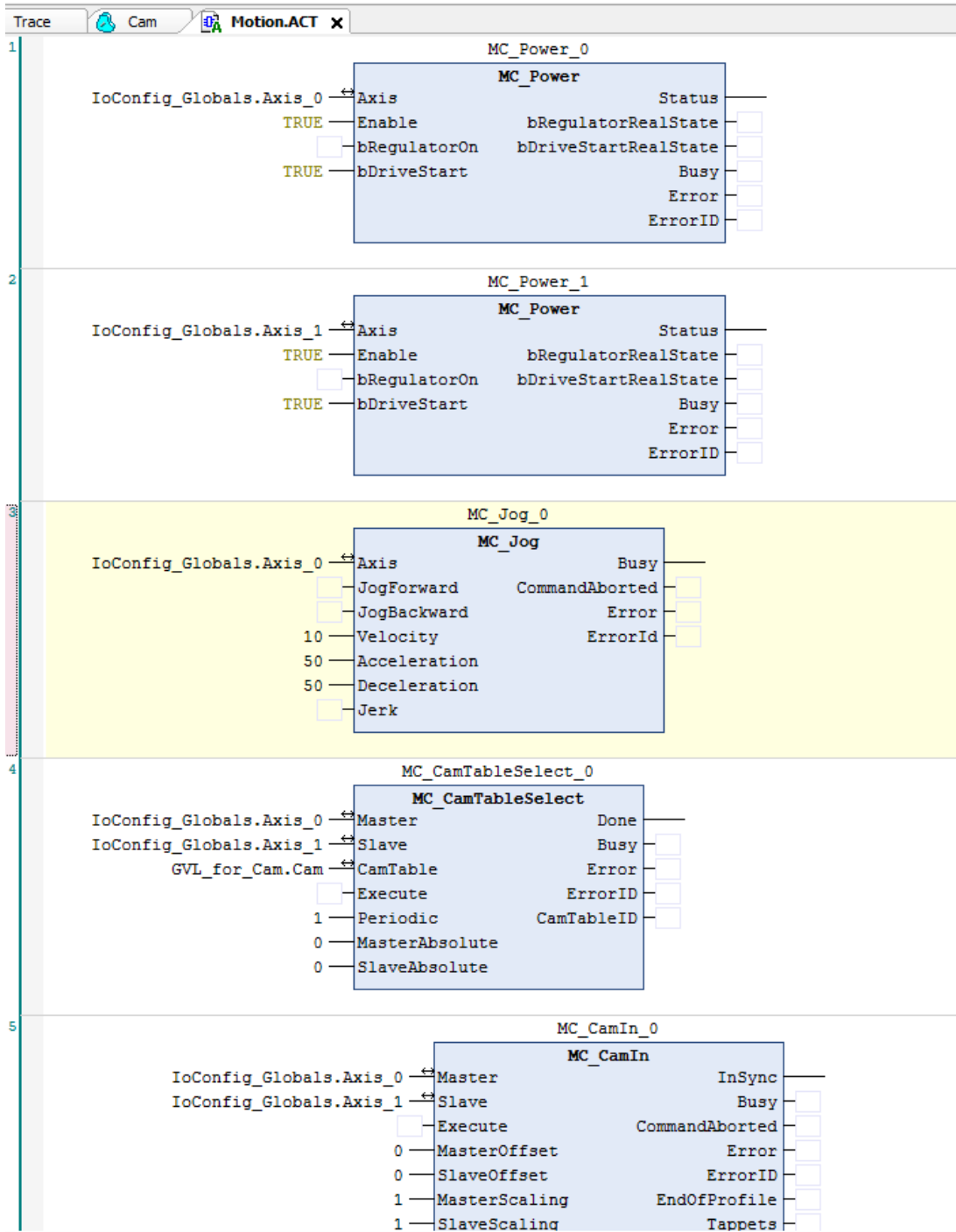


多轴运动以凸轮为例

右击【Application】→【添加对象】→【cam表】



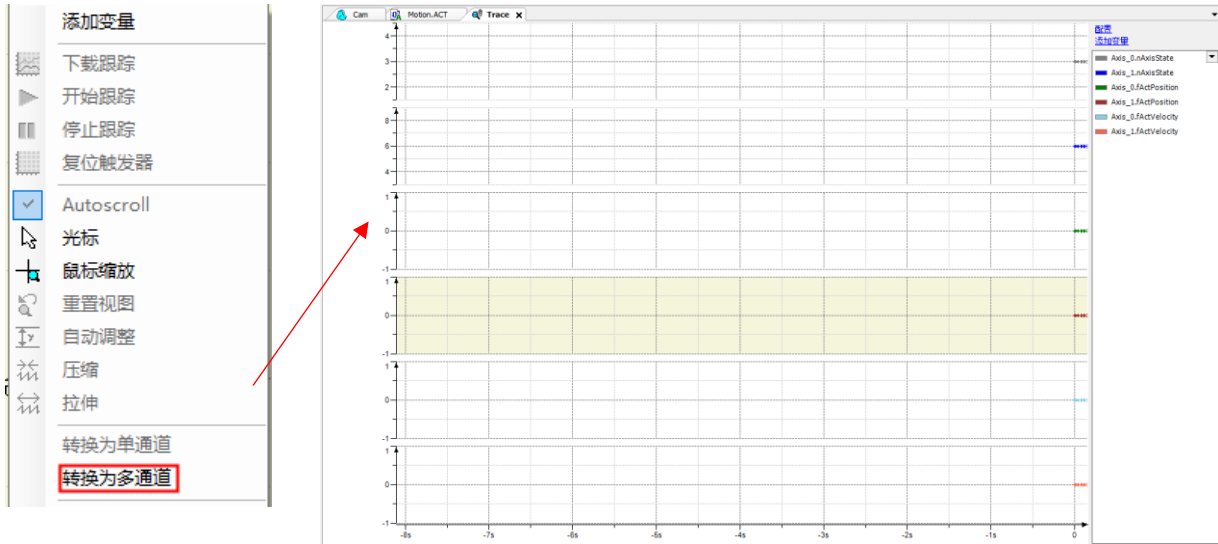
在【Motion.ATC】中添加【MC_Power】使能功能块，【MC_CamTableSelect】链接Cam表功能块，【MC_CamIn】轴耦合功能块



为方便用户追踪，可添加Trace（可查看4.2.3Trace）。可追踪两轴的轴状态，当前位置，当前速度等



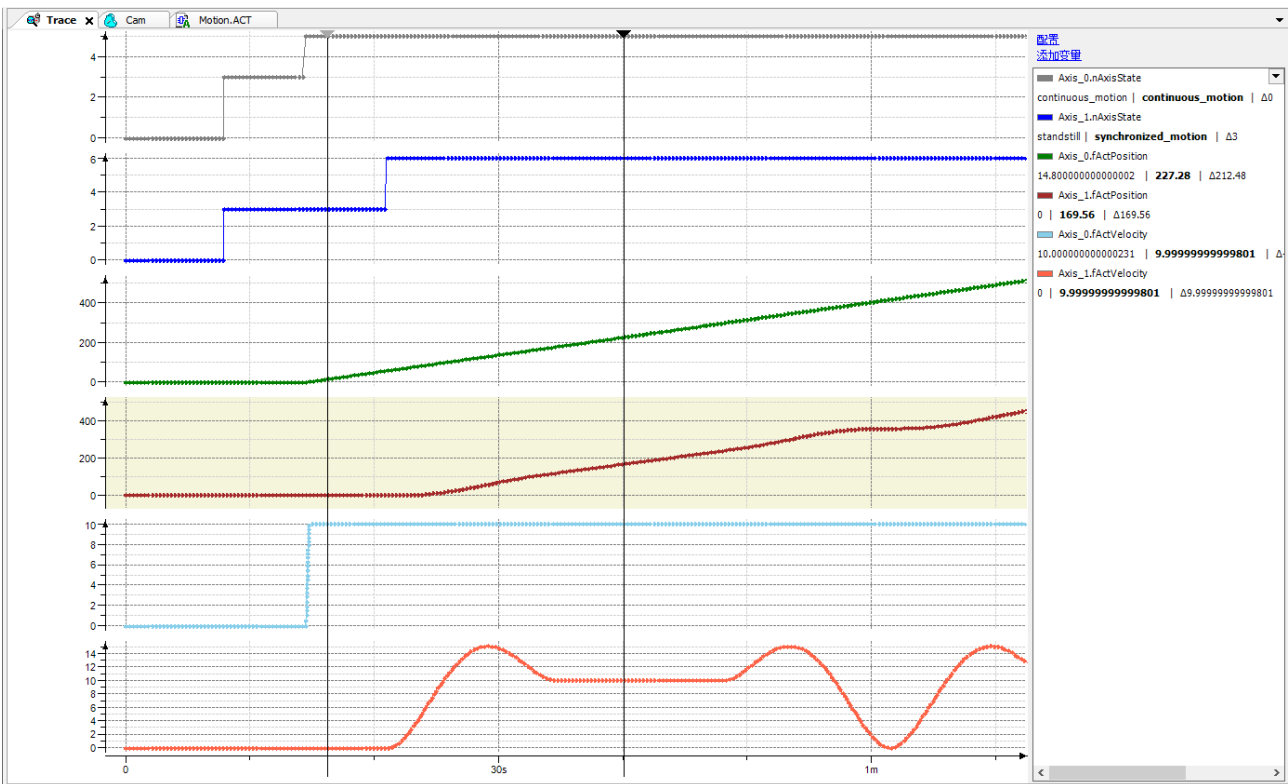
由于是多个变量，可以分多个图表，右键空白处



登录运行，给轴使能，【MC_Power】功能块引脚【Enable】【bRegulatorOn】【bDriveStart】置TRUE，运动控制功能块需要给定加减速度，不然轴会报错，处于【errorstop】状态（可查看附录：轴状态转移图）



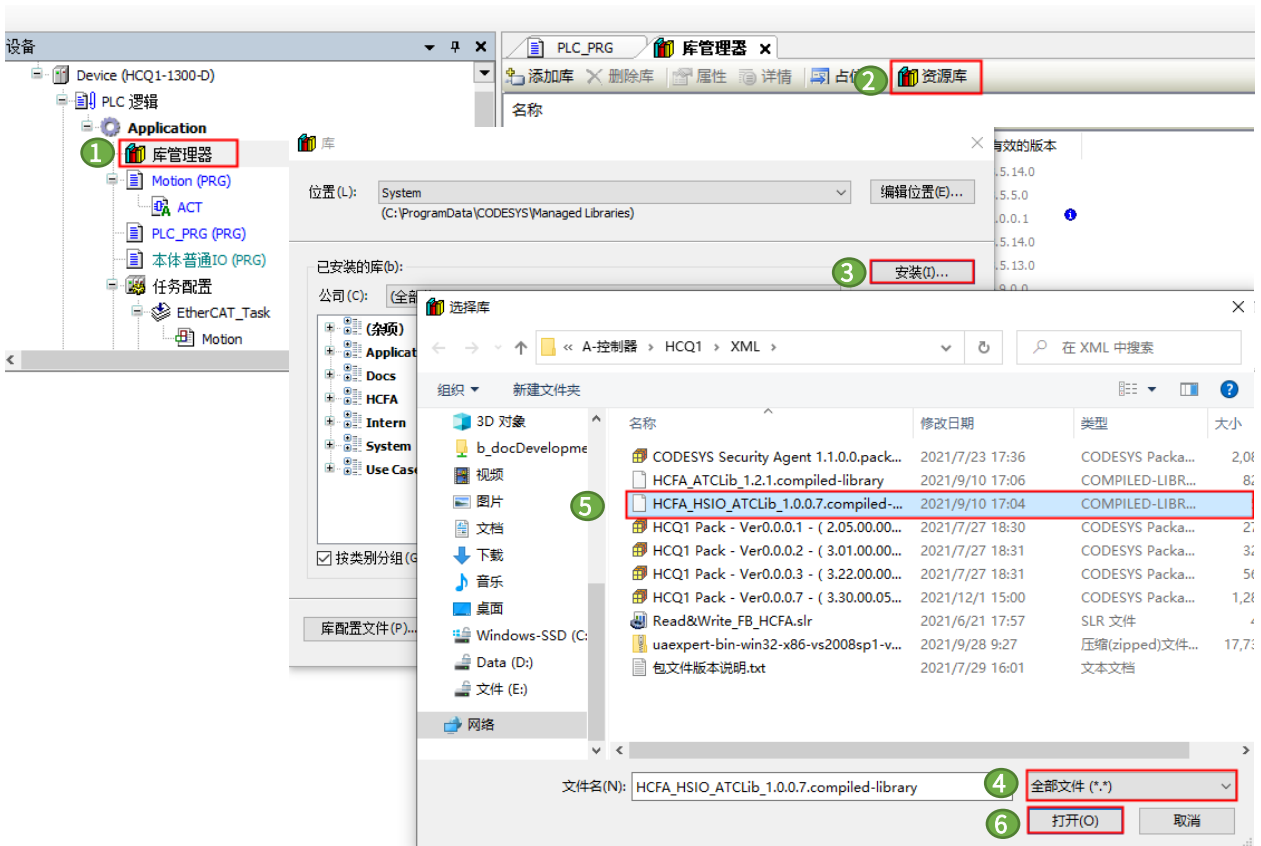
运动后查看Trace，可观察到从轴的当前位置和当前速度都以Cam表提前规划的轨迹运动



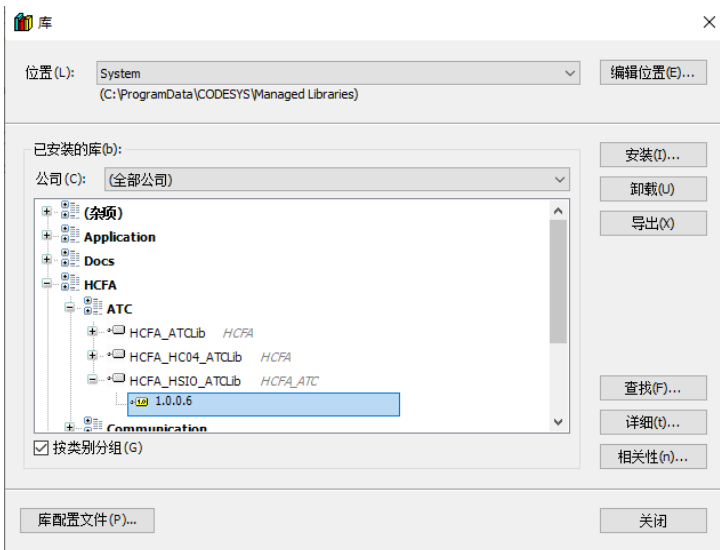
附录

描述文件安装

单击【库管理器】→【资源库】→【安装】，选择【全部文件】，单击需要安装的描述文件，点击【打开】



跳转出如下图，则安装成功



Modbus RTU数据报文结构

Modbus 是一个请求/应答协议，并且提供功能码规定的服务，下面介绍ModbusRTU的报文结构。

以下列请求及应答报文为例：

请求：01 03 01 16 00 03 E5 F3

应答：01 03 06 17 84 17 80 17 8A 58 47

请求报文：

01 03 01 16 00 03 E5 F3				
名称	示例	长度	说明	备注
域名	0x01	1	从站地址	从站地址范围1~247
功能码	0x03	1	功能码，读多寄存器	参考标准modbus协议
数据	0x0116	2	起始地址	
	0x 0003	2	寄存器数量	读取3个寄存器（共6个字节）
校验	E5 F3	2	校验码	

应答报文：

01 03 06 17 84 17 80 17 8A 58 47				
名称	示例	长度	说明	备注
域名	0x01	1	从站地址	从站地址范围1~247
功能码	0x03	1	功能码，读多寄存器	参考标准modbus协议
数据	0x06	1	寄存器长度	3个寄存器共6个字节
	0x1784	2	读取数值	地址为0116内存的内容
	0x1780	2	读取数值	地址为0117内存的内容
	0x178A	2	读取数值	地址为0118内存的内容
校验	58 47	2	校验码	

部分功能码定义如下：

功能码	描述	访问类型	Q1从站地址	数据类型	操作数量
0x01	线圈	读	%QX0.0 - %QX8191.7	位	单/多
0x02	离散输入	读	%IX0.0 - %IX8191.7	位	单/多
0x03	保持寄存器	读	%MW0 - %MW65535	字	单/多
0x04	输入寄存器	读	%MW0 - %MW65535	字	单/多
0x05	单个线圈	写	&QX0.0 - &QX8191.7	位	单
0x06	单个保持寄存器	写	%MW0 - %MW65535	字	单
0x0F	多个线圈	写	&QX0.0 - &QX8191.7	位	多
0x10	多个保持寄存器	写	%MW0 - %MW65535	字	多
0x17	多个保存寄存器	读/写	%MW0 - %MW65535	字	多

Modbus 寄存器地址分配如下表：

Q1从站地址	适用功能	寄存器种类	读写状态
%QX0.0 - %QX8191.7	01H	线圈	可读
	05H 0FH		可写
%IX0.0 - %IX8191.7	02H	离散输入	可读
%MW0 - %MW65535	04H	输入寄存器	可读

%MW0 - %MW65535	03H 06H 10H 17H	保存寄存器	可读可写
-----------------	-----------------	-------	------

寄存器种类说明如下表：

寄存器种类	说明	PLC类比	举例说明
线圈	输出端口，可设定端口的输出状态，也可以读取该位的输出状态。可分为两种不同的执行状态，例如保持型和边沿触发型。	DO数字量输出	电磁阀输出、MOSFET输出、LED显示等。
离散输入	输入端口，通过外部设定该案输入状态，可读但不可写。	DI数字量输入	拨码开关、接近开关等。
输入寄存器	输入参数，控制器运行时从外部设备获得的参数，可读但不可写。	AD模拟量输入	模拟量输入
保存寄存器	输出参数或保存参数，控制器运行时被设定的某些参数，可读可写。	DA模拟量输出	模拟量输出设定值，PID运行参数，变量阀输出大小，传感器报警上限下限。

Modbus TCP数据报文结构

Modbus 是一个请求/应答协议，并且提供功能码规定的服务，下面介绍Modbus TCP的报文结构。

以下列请求及应答报文为例：

请求：97 76 00 00 00 06 04 04 00 00 00 01

应答：97 76 00 00 00 05 04 04 02 12 34

请求报文：

97 76 00 00 00 06 04 04 00 00 00 01				
名称	说明	长度	说明	备注
Map报文头	0x97	1	事务处理标识符Hi	客户机发起，服务器复制，用于事务处理配对
	0x76	1	事务处理标识符Lo	
	0x0000	2	协议标识符号	客户机发起，服务器复制Modbus协议= 0.
	0x0006	2	长度	从本字节下一个到最后
	0x04	1	单元标识符	客户机发起，服务器复制，串口链路或其他总线上远程终端标识
功能码	0x04	1	功能码，读寄存器	参考标准modbus协议
数据	0x0000	2	起始地址	
	0x 0001	2	寄存器数量	
校验				

应答报文：

97 76 00 00 00 05 04 04 02 12 34				
名称	示例	长度	说明	备注
Map报文头	0x97	1	事务处理标识符Hi	客户机发起，服务器复制，用于事务处理配对
	0x76	1	事务处理标识符Lo	
	0x0000	2	协议标识符号	客户机发起，服务器复制 Modbus协议= 0.
	0x0005	2	长度	从本字节下一个到最后
	0x04	1	单元标识符	客户机发起，服务器复制 串口链路或其他总线上远程终端标识
功能码	0x04	1	功能码，读寄存器	参考标准modbus协议
数据	0x02	1	字节个数	
	0x ----		数据	此报文中数据为12 34
校验				

部分功能码定义如下表：

功能码	描述	访问类型	Q1从站地址	数据类型	操作数量
0x01	线圈	读	%QX0.0 - %QX8191.7	位	单/多
0x02	离散输入	读	%IX0.0 - %IX8191.7	位	单/多
0x03	保持寄存器	读	%MW0 - %MW65535	字	单/多
0x04	输入寄存器	读	%MW0 - %MW65535	字	单/多
0x05	单个线圈	写	%QX0.0 - %QX8191.7	位	单
0x06	单个保持寄存器	写	%MW0 - %MW65535	字	单
0x0F	多个线圈	写	%QX0.0 - %QX8191.7	位	多
0x10	多个保持寄存器	写	%MW0 - %MW65535	字	多

Modbus 寄存器地址分配如下表：

Q1从站地址	适用功能	寄存器种类	读写状态
%QX0.0 - %QX8191.7	01H	线圈	可读
	05H 0FH		可写
%IX0.0 - %IX8191.7	02H	离散输入	可读
%MW0 - %MW65535	04H	输入寄存器	可读

%MW0 - %MW65535	03H 06H 10H 17H	保存寄存器	可读可写
-----------------	-----------------	-------	------

寄存器种类说明如下表：

寄存器种类	说明	PLC类比	举例说明
线圈	输出端口，可设定端口的输出状态，也可以读取该位的输出状态。可分为两种不同的执行状态，例如保持型和边沿触发型。	DO数字量输出	电磁阀输出、MOSFET输出、LED显示等。
离散输入	输入端口，通过外部设定该案输入状态，可读但不可写。	DI数字量输入	拨码开关、接近开关等。
输入寄存器	输入参数，控制器运行时从外部设备获得的参数，可读但不可写。	AD模拟量输入	模拟量输入
保存寄存器	输出参数或保存参数，控制器运行时被设定的某些参数，可读可写。	DA模拟量输出	模拟量输出设定值，PID运行参数，变量阀输出大小，传感器报警上限下限。

AD04对象字典总表

对象字典	子索引	名称	属性	类型	默认值	单位	备注
0x1000	00	设备类型	RO	UDINT	5001	-	
0x1008	00	设备名称	RO	STRING(20)		-	HCQX_AD04-D
0x1009	00	硬件版本	RO	STRING(5)	--	-	
0x100A	00	软件版本	RO	STRING(5)	--	-	
0x1011	00	恢复默认参数				-	
	01	映射对象	RW	BOOL	FALSE	-	
0x1018	00	标识对象					
	01	供应商ID	RO	UDINT			
	02	产品代码	RO	UDINT			
	03	修订号	RO	UDINT			
	04	序列号	RO	UDINT			
0x1A00	00	子索引个数				-	组1默认映射对象个数
	01~12	TXPDO映射对象组1	RO	UDINT	略	-	组1默认TXPDO映射对象
0x1A01	00	子索引个数					组1默认映射对象个数
	01	TXPDO映射对象组1	RO	UDINT	略		组1默认TXPDO映射对象
0x1A02	00	子索引个数	R/W	USINT	0	-	组2默认映射对象个数
	01~12	TXPDO映射对象组2	R/W	UDINT	略	-	组2默认TXPDO映射对象
0x1C00	00	同步管理器类型				-	

	01~0C	映射对象001~004	RO	USINT	1~4	-	
0x1C13	00	TXPDO分配				-	
	01~10	映射对象001~016	RO	UINT			
0x1C33	00	输入参数				-	
	01	同步模式	RW	UINT	01		
	02	循环时间	RO	UDINT	7122000 (DEC)		SM通讯循环时间 (ns)
	04	支持的同步模式	RO	UINT	3		支持的同步模式: 0x0001: FREE RUN 0x0002: SM
	05	最小循环时间	RO	UDINT	100000 (DEC)		(ns)
	06	计算和复制时间	RO	UDINT	0		
	08	获取循环时间	RW	UINT	0		
	09	延迟时间	RO	UDINT	0		
	0A	SYNC0循环时间	RW	UDINT	0		
	0B	SM事件丢失	RO	UINT	0		
	0C	循环时间太小	RO	UINT	0		
	20	SYNC错误	RO	BOOL	FALSE		
	0x6000	00	通道1状态				
01		量程内	RO	BOOL	FALSE		
02		超量程	RO	BOOL	FALSE		
03		限制1	RO	BOOL	FALSE		
05		限制2	RO	BOOL	FALSE		
07		错误	RO	BOOL	FALSE		
0x6010	00	数值	RO	INT	0		
0x6020	00	最大值	RO	INT	0		
0x6030	00	最小值	RO	INT	0		
0x6040	00	通道2状态					
	01	量程内	RO	BOOL	FALSE		
	02	超量程	RO	BOOL	FALSE		
	03	限制1	RO	BOOL	FALSE		
	05	限制2	RO	BOOL	FALSE		

	07	错误	RO	BOOL	FALSE		
0x6050	00	数值	RO	INT	0		
0x6060	00	最大值	RO	INT	0		
0x6070	00	最小值	RO	INT	0		
0x6080	00	通道3状态					
	01	量程内	RO	BOOL	FALSE		
	02	超量程	RO	BOOL	FALSE		
	03	限制1	RO	BOOL	FALSE		
	05	限制2	RO	BOOL	FALSE		
	07	错误	RO	BOOL	FALSE		
0x6090	00	数值	RO	INT	0		
0x60A0	00	最大值	RO	INT	0		
0x60B0	00	最小值	RO	INT	0		
0x60C0	00	通道4状态					
	01	量程内	RO	BOOL	FALSE		
	02	超量程	RO	BOOL	FALSE		
	03	限制1	RO	BOOL	FALSE		
	05	限制2	RO	BOOL	FALSE		
	07	错误	RO	BOOL	FALSE		
0x60D0	00	数值	RO	INT	0		
0x60E0	00	最大值	RO	INT	0		
0x60F0	00	最小值	RO	INT	0		
0x80n0	00	通道n输入设置					n=0、1、2、3
	01	输入模式选择	RW	USINT	0		设定值： 模式0：0~10V、 模式1：-10~10V、 模式2：-5~5V、 模式3：0~5V、 模式4：1~5V、 模式5：0~20mA、 模式6：4~20mA
	09	使能通道	RW	BOOL	FALSE		设定值：True、False
	0A	使能比例调整	RW	BOOL	FALSE		设定值：True、False
	0B	使能峰值监测	RW	BOOL	FALSE		设定值：True、False

0C	使能滤波器	RW	BOOL	FALSE		设定值: True、False
0D	使能限位1	RW	BOOL	FALSE		设定值: True、False
0E	使能限位2	RW	BOOL	FALSE		设定值: True、False
11	用户比例偏置	RW	INT	0		设定值: 模式0: 0; 模式1: 未开启用户比例模式时0、 开启用户比例模式时0~User Scale Gain-1000; 模式2: 0; 模式3: 0; 模式4: 1000; 模式5: 未开启用户比例模式时0、 开启用户比例模式时0~User Scale Gain-3000; 模式6: 4000;
12	用户比例增益	RW	INT	5000		设定值: 模式0: 5000; 模式1: 未开启用户比例模式时 5000、开启用户比例模式时User Scale Offset+1000~10000; 模式2: 5000; 模式3: 5000; 模式4: 3000; 模式5: 未开启用户比例模式时 20000、开启用户比例模式时User Scale Offset+3000~20000; 模式6: 20000;
13	限制1	RW	INT	0		设定值: 模式0: -32~Limit2-1; 模式1: -32768~ Limit2-1; 模式2: -16384~ Limit2-1; 模式3: -16~ Limit2-1; 模式4: -16~ Limit2-1; 模式5: -32~ Limit2-1; 模式6: -32~ Limit2-1;
14	限制2	RW	INT	16383		设定值: 模式0: Limit1+1~32767; 模式1: Limit1+1~32767; 模式2: Limit1+1~16383; 模式3: Limit1+1~16383; 模式4: Limit1+1~16383; 模式5: Limit1+1~32767; 模式6: Limit1+1~32767;
15	滤波设置	RW	INT	1		设定值: 0~4096;

DA04对象字典总表

对象字典	子索引	名称	属性	类型	默认值	单位	备注
0x1000	00	设备类型	RO	UDINT	5001	-	
0x1001	00	错误寄存器	RO	USINT	0		

0x1008	00	设备名称	RO	STRING(20)		-	HCQX_DA04-D
0x1009	00	硬件版本	RO	STRING(3)	--	-	
0x100A	00	软件版本	RO	STRING(3)	--	-	
0x1011	00	恢复默认参数				-	
	01	恢复默认参数	RW	BOOL	FALSE	-	
0x1018	00	标识对象					
	01	供应商ID	RO	UDINT			
	02	产品代码	RO	UDINT			
	03	修订号	RO	UDINT			
	04	序列号	RO	UDINT			
0x1600	00	DA通道1参数				-	
	01	映射对象001	RO	UDINT		-	
0x1601	00	DA通道2参数				-	
	01	映射对象001	RO	UDINT		-	
0x1602	00	DA通道3参数				-	
	01	映射对象001	RO	UDINT		-	
0x1603	00	DA通道4参数				-	
	01	映射对象001	RO	UDINT		-	
0x1C00	00	同步管理器类型				-	
	01~0C	映射对象001~004	RO	USINT	1~4	-	
0x1C12	00	RXPDO 分配				-	
	01~04	映射对象001~004	RO	UINT			
0x1C32	00	输出参数				-	
	01	同步模式	RW	UINT	01		
	02	循环时间	RO	UDINT			SM通讯循环时间 (ns)
	04	支持的同步模式	RO	UINT	3		支持的同步模式: 0x0001: FREE RUN 0x0002: SM
	05	最小循环时间	RO	UDINT	100000 (DEC)		(ns)
	06	计算和复制时间	RO	UDINT	0		

	08	获取循环时间	RW	UINT	0		
	09	延迟时间	RO	UDINT	0		
	0A	SYNC0循环时间	RW	UDINT	0		
	0B	SM事件丢失	RO	UINT	0		
	0C	循环时间太小	RO	UINT	1		
	20	SYNC错误	RO	BOOL	FALSE		
0x7010	00	DA通道输出数值					
	01	通道1	RW	INT	0		
	02	通道2	RW	INT	0		
	03	通道3	RW	INT	0		
	04	通道4	RW	INT	0		
0x7011	00	DAC错误					
	01	通道1错误	RO	BOOL	FALSE		
	02	通道2错误	RO	BOOL	FALSE		
	03	通道3错误	RO	BOOL	FALSE		
	04	通道4错误	RO	BOOL	FALSE		
0x80n0	00	通道n输出设置					n=0、1、2、3
	01	输出模式选择	RW	USINT	0		设定值： 模式0：0~10V、 模式1：-10~10V、 模式2：-5~5V、 模式3：0~5V、 模式4：1~5V、 模式5：0~20mA、 模式6：4~20mA
	2	通道使能	RW	BOOL	TRUE		设定值：True、False
	3	使能用户校准	RW	BOOL	FALSE		
	4	使能比例调整	RW	BOOL	FALSE		暂不支持
	6	看门狗	RW	UINT	0		暂不支持
	7	错误/停止输出模式	RW	UINT	0		暂不支持
	8	用户输出值	RW	UINT	0		暂不支持
	9	用户比例偏差	RW	INT	10000		暂不支持
	A	用户比例增益	RW	DINT	0		暂不支持
	B	用户校准值	RW	INT	16000		暂不支持

	C	用户最大校准值	RW	INT	0		暂不支持
--	---	---------	----	-----	---	--	------

模拟量模块模数值转换及显示表

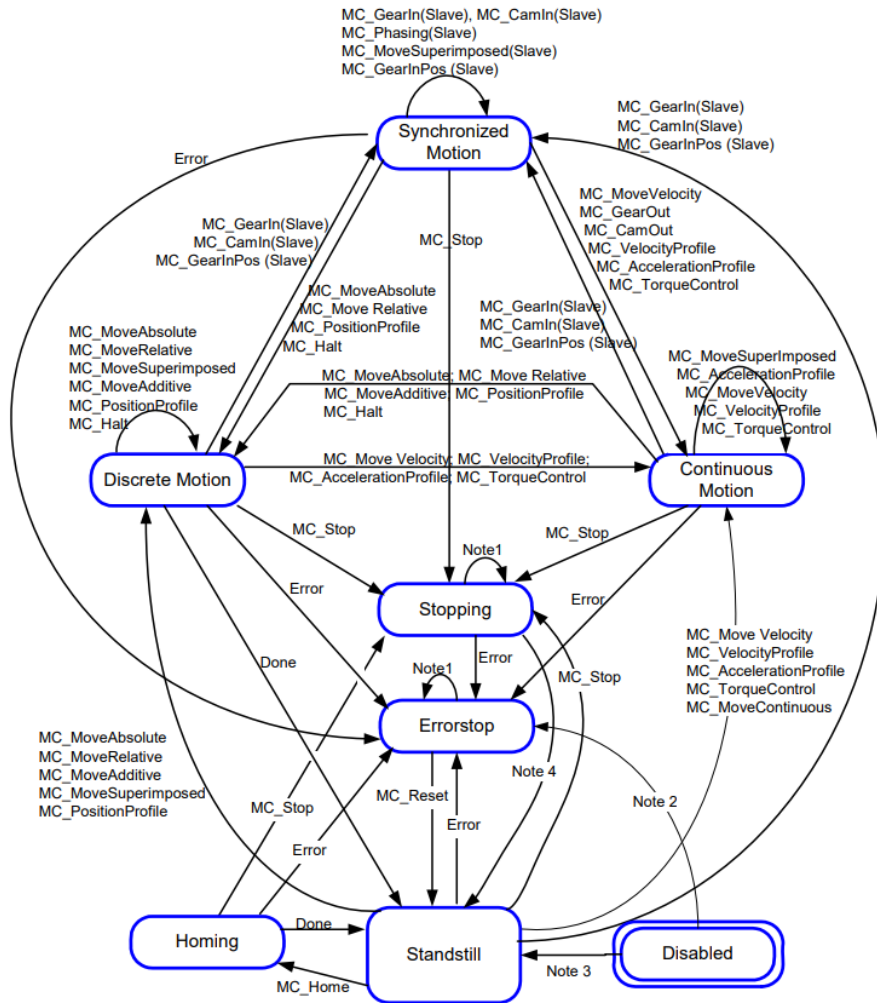
用户显示配置由index 0x80n0:01决定，在该索引下用户可以选择修改该通道的工作模式以显示不同测量范围下的数值。

模拟量模块为16位分辨率，最大显示范围为 -32768_{dec} 到 $+32767_{dec}$ ，不同模式下测量值和显示值对应如下：

输入/输出信号								数值	
HCQX-AD04-D				HCQX-DA04-D				Decimal	Hexadecimal
10V		20mA		10V		20mA		32000	0x7D00
5V				5V				16000	0x3E80
									0x0001
0V	1V	0mA	4mA	0V	1V	0mA	4mA	0	0x0000
									0x0001
-5V				-5V				-16000	0xC180
-10V				-10V				-32000	0x8300

轴状态转移图

根据PLC OPEN规定，轴状态可做如下转换：



Note1	在Errorstop或Stopping状态时，所有的功能块都可以被调用，但是他们不会并不会执行，除了MC_Reset可以把状态转换为Standstill。如果错误发生在轴状态为Stopping时，轴状态会转换成Errorstop
Note2	MC_Power的Enable引脚为TRUE且轴发生错误
Note3	MC_Power的Enable引脚为TRUE且轴没有发生错误
Note4	MC_Stop的Done引脚为TRUE且MC_Stop的Execute引脚为FALSE

注：codesys通过枚举体SMC_AXIS_STATE列出了PLC OPEN所规定的轴状态，具体如下：

- | | |
|---------------------------------------|------------------------|
| 0: power_off (在PLC OPEN的定义中为Disabled) | 4: discrete_motion |
| 1: errorstop | 5: continuous_motion |
| 2: stopping | 6: synchronized_motion |
| 3: standstill | 7: homing |