



XS3 系列可编程序控制器

用户手册[功能篇]

无锡信捷电气股份有限公司

资料编号 PS02 20191203 3.5

基本说明

- ◆ 感谢您购买了信捷 XS3 系列中型 PLC 产品。
- ◆ 本手册主要介绍 Codesys 编程的相关应用。
- ◆ 在使用产品之前，请仔细阅读本手册，并在充分理解手册内容的前提下进行操作。
- ◆ XS3 产品的基本使用、规格等内容请查阅相关手册。
- ◆ 请将本手册交付给最终用户。

用户须知

- ◆ 只有具备一定的电气知识的操作人员才可以对产品进行接线等其他操作，如有使用不明的地方，请咨询本公司的技术支持人员。
- ◆ 手册等其他技术资料中所列举的示例仅供用户理解、参考用，不保证一定动作。
- ◆ 将该产品与其他产品组合使用的时候，请确认是否符合有关规格、原则等。
- ◆ 使用该产品时，请自行确认是否符合要求以及安全，对于本产品故障而可能引发机器故障或损失时，请自行设置后备及安全功能。

责任申明

- ◆ 手册中的内容虽然已经过仔细的核对，但差错难免，我们不能保证完全一致。
- ◆ 我们会经常检查手册中的内容，并在后续版本中进行更正，欢迎提出宝贵意见。
- ◆ 手册中所介绍的内容，如有变动，请谅解不另行通知。

联系方式

如果您有任何关于本产品的使用问题，请与购买产品的代理商、办事处联系，也可以直接与信捷公司联系。

- ◆ 电话：400-885-0136
- ◆ 传真：0510-85111290
- ◆ 地址：无锡市滴翠路 100 号创意产业园 7 号楼 4 楼
- ◆ 邮编：214072

WUXI XINJE ELECTRIC CO., LTD. 版权所有

未经明确的书面许可，不得复制、传翻或使用本资料及其中的内容，违者要对造成的损失承担责任。保留包括实用模块或设计的专利许可及注册中提供的所有权力。

二〇一九年 十二月

目 录

1. 前言	3
1-1. 参考资料	4
1-2. 概述	4
1-3. 手册更新日志	4
2. 软件背景概述	5
2-1. PLCOPEN 概述	6
2-2. IEC 61131-3 概述	6
2-3. CODESYS 概述	6
3. XS3 运动控制系统组成	7
3-1. 系统构成	8
3-2. 系统通讯规格	8
4. ETHERCAT 系统通讯原理及配置	10
4-1. ETHERCAT 总线网络原理分析	11
4-1-1. EtherCAT 总线网络概述	11
4-1-2. EtherCAT 帧结构	11
4-1-3. EtherCAT 伺服驱动器控制应用协议—CoE 协议	12
4-1-4. EtherCAT 总线通信同步模式—DC (SYNC0 事件同步)	12
4-2. CiA402 数据对象字典与伺服驱动器常用对象	13
4-2-1. CiA402 对象字典概述	13
4-2-2. 伺服驱动器常用对象	14
4-3. ETHERCAT 主从站通讯的配置	22
4-3-1. 安装 EtherCAT 主从站设备描述文件	22
4-3-2. EtherCAT 主站配置	24
4-3-3. EtherCAT 从站配置	29
4-4. 伺服轴电机参数的配置	34
4-4-1. 基本参数配置	34
4-4-2. Scaling/Mapping (“缩放/映射”) 配置	35
4-4-3. I/O Mapping (I/O 映射)	37
4-4-4. IEC Objects	37
4-4-5. 电机轴状态	37
4-4-6. 电机轴信息	38
5. 以太网通讯原理及配置	39
5-1. 简介	40
5-2. ETHERNET 参数介绍	40
5-3. 组网方式	40
5-3-1. 直连	41
5-3-2. 中转连接方式	41
5-4. 以太网通讯原理分析	41
5-4-1. TCP/IP 协议	41
5-4-2. MODBUS TCP 协议概述	41
5-4-3. MODBUS 通讯地址	42
5-4-4. MODBUS 通讯功能码	42

5-5. 以太网通讯配置.....	42
5-6. 以太网通讯案例.....	45
6. 用户程序的组成及执行机制	47
6-1. 用户程序基本结构.....	48
6-2. 用户程序中的任务与配置.....	48
6-2-1. 任务类型	48
6-2-2. 任务配置	49
6-3. 用户程序由多个 POU 组成的优势	52
7. 用户程序应用编程	53
7-1. 基于 ETHERCAT 运动控制编程	54
7-1-1. XS3 EtherCAT 运控应用的编程要点	54
7-1-2. 单轴控制常用的 MC 功能块	54
7-1-3. MC 指令与 PDO/SDO 配置.....	55
7-2. 多轴高速定位运动控制编程.....	56
7-2-1. 高速定位控制功能概述及编程要点	56
7-2-2. 编程示例	58
7-3. 多路高速输入信号采集功能.....	59
7-3-1. 高速输入采集功能概述及编程要点	59
7-3-2. 编程示例	60
7-4. 扩展模块应用编程.....	61
7-4-1. 扩展模块应用功能概述及编程要点	61
7-4-2. 编程示例	63
7-5. MODBUS 通讯应用编程	65
7-5-1. MODBUS 通讯功能概述及编程要点.....	66
7-5-2. 编程示例	67
8. 自开发功能块详解	69
8-1. 高速计数指令.....	70
8-1-1. 指令格式	70
8-1-2. 相关变量	70
8-1-3. 功能说明	72
8-2. 以太网通讯指令.....	73
8-2-1. 指令格式	73
8-2-2. 相关变量	73
8-2-3. 功能说明	76
8-3. MODBUS 通讯指令	77
8-3-1. 指令格式	77
8-3-2. 相关变量	77
9. 附录	79
9-1. 小工具的使用.....	80

1. 前言

本章主要介绍本手册的参考资料、著作说明以及版本更新。

1. 前言	3
1-1. 参考资料	4
1-2. 概述	4
1-3. 手册更新日志.....	4

1-1. 参考资料

本手册在参考以下资料的基础上完成。

注：对于本手册和下述参考资料的记载内容的不同点，以本手册的记载内容为准。

序号	文档名	出版社	日期
1	开放式控制系统编程技术——基于 IEC 61131-3 国际标准	人民邮电出版社	2018-08
2	PLC 综合开发利器——CODESYS 基础编程及应用指南	--	--

1-2. 概述

信捷 XS3 系列控制器选用的是德国 3S 公司的 Codesys 编程平台，该平台完整支持 PLCopen 规范，用户可以引用许多标准的功能函数库；高级语言的编程方式，易于 PLC 厂家和用户开发自己专有的功能块和指令库，可显著提高用户编程效率。

本手册的主要目的是要帮助广大 XS3 产品使用者正确、快速的使用产品，手册中较为详细地介绍了 XS3 系列控制器支持的通讯协议、应用系统、执行机制以及在 Codesys 编程平台如何实现运动控制。

1-3. 手册更新日志

序号	更新时间	更新说明
1	2019.12.15	手册第一版

2. 软件背景概述

本章主要介绍 PLCopen、IEC61131-3 以及 Codesys 编程平台的基本概念。

2. 软件背景概述	5
2-1. PLCOPEN 概述	6
2-2. IEC 61131-3 概述	6
2-3. CODESYS 概述	6

2-1. PLCopen 概述

PLCopen 国际组织是独立于生产商和产品的全球性机构。其宗旨是致力于提高控制软件编程方法、效率、规范等相关方面问题，从而支持使用该领域国际标准。

2-2. IEC 61131-3 概述

PLCopen 的一项主要活动是致力于 IEC61131-3，它是第一个为工业控制系统提供标准化编程语言的国际标准。长期以来，用户可在很大范围内选择实现运动控制的硬件，不过，每种硬件都要求独自而无法兼容的开发软件。即使所要求的功能完全相同，在更换另一种硬件时，也需要重新编写软件。这一困扰运动控制用户的问题，其实质就是如何实现运动控制软件的标准化。PLCopen 组织考虑到用户存在运动控制软件标准话的需求，从 1996 年就建立了运动控制规范组，历时十多年完成这一挑战性工作。

IEC61131-3 标准针对工业控制系统所阐述的软件设计概念、模型等，适应当今世界软件、工业控制系统的发展方向，是一种非常先进的设计技术。他极大的推动了工业控制系统软件设计的发展，对现场总线设备的软件也产生了很大的影响。

该标准定义了用于构造程序内部结构的 SFC（顺序功能图）语言和四个互操作编程语言：IL（指令表）、LD（梯形图）、FBD（功能块图）和 ST（结构文本）。通过分解成逻辑元素、模块化以及现代软件技术来组成每个程序，从而提高了其重复使用性，减少了错误，提高了编程和用户的效率。

2-3. Codesys 概述

信捷 XS3 系列控制器选用的是德国 3S 公司的 Codesys 编程平台。Codesys 是一款工业信息技术、自动化编程软件及智能制造装备编程开发平台，向全球用户提供开放灵活、稳定可靠的一系列先进的工业信息技术、软件产品和行业解决方案。目前，全球大约有 350 家控制系统生产商是 Codesys 的用户。该平台完全支持 PLCopen 规范，提供了所有由 IEC 国际标准定义和支持的用于自动化应用开发的编辑器。

Codesys 包括 PLC 编程、可视化 HMI、安全 PLC、控制器实时核、现场总线及运动控制，是一个完整的自动化软件。

3. XS3 运动控制系统组成

本章主要介绍 XS3 的系统构成和通讯规格。

3. XS3 运动控制系统组成	7
3-1. 系统构成	8
3-2. 系统通讯规格.....	8

3-1. 系统构成

这是CODESYS基于传统硬PLC的控制系统。此方案中，PLC开发系统一般在普通PC机上运行，而传统硬PLC只是作为一个硬件平台，将软PLC的实时核安装在传统硬PLC中，将开发系统编写的系统程序下载到硬PLC中。XS3是一个具有运动控制功能的通用型可编程控制器，通过EtherCAT总线控制多个运动轴，典型的控制总线网络如下图：



整个总线网络为线型结构，其中XS3系列控制器为主站，信捷DS5C系列总线控制型伺服为从站。XS3系列PLC带有上下两个网口，上面的网口为Ethernet/IP，用于连接CODESYS上位机；下面的网口为EtherCAT连接口，用于连接XINJE DS5C系列伺服实现EtherCAT通讯。信捷DS5C系列伺服驱动器的两个通讯网口则需遵循“下进上出”的原则。

3-2. 系统通讯规格

项目	规格
物理层	100BASE-TX（IEEE802.3）
波特率	100[Mbps]（full duplex）
拓扑	Line
连接电缆	JC-CA 双绞线（屏蔽双绞线）
电缆长	节点间最长 50m
通信口	2 Port（RJ45）
EtherCAT Indicators（LED）	[Run] RUN Indicator [L/A IN] Port0 Link/Activity Indicator（Green） [L/A OUT] Port1 Link/Activity Indicator（Green）
Station Alias（ID）	设定范围：0～65535；设定地址：2700h
Explicit Device ID	不支持
邮箱协议	COE（CANopen Over EtherCAT）
SyncManager	4

项目	规格			
FMMU	3			
Modes of operation 控制模式	位置	csp	Cyclic synchronous position mode (Cyclic 位置控制模式)	
		PP	Profile position mode (Profile 位置控制模式)	
		hm	Homing mode (原点复位位置控制模式)	
	速度	csv	Cyclic synchronous velocity mode (Cyclic 速度控制模式)	
		pv	Profile velocity mode (Profile 速度控制模式)	
	转矩	cst	Cyclic synchronous torque mode (Cyclic 转矩控制模式)	
		tq	Torque profile mode (Profile 转矩控制模式)	
	Touch Probe	2 路		
	同期模式	DC (SYNCO 事件同期) SM (SM事件同步)		
Cyclic time (DC 通信周期)	500, 1000, 2000, 4000[μs]			
通信对象	SDO[服务数据对象], PDO[过程数据对象]			
单站 PDO 最大分配数	TxPDO: 4 [个] RxPDO: 4 [个]			
单站 PDO 最大字节数	TxPDO: 24[byte] RxPDO: 24[byte]			
PreOP 模式下邮箱通讯间隔	1ms			
电子邮箱	SDO 请求和 SDO 信息			

4. EtherCAT 系统通讯原理及配置

本章主要介绍 EtherCAT 系统通讯原理、常用对象、电机参数配置等内容。

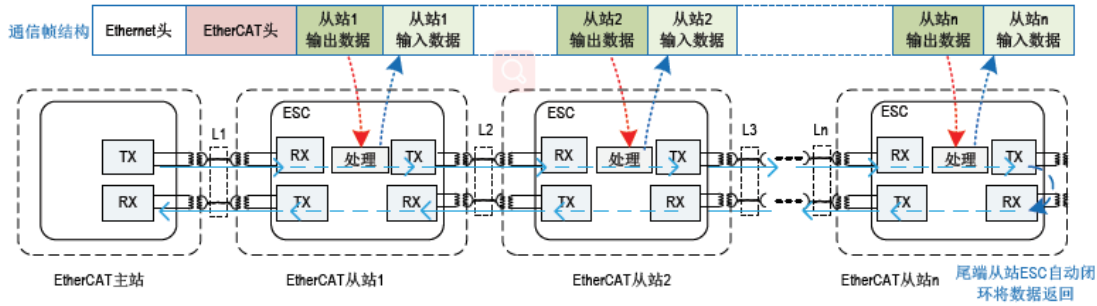
4. ETHERCAT 系统通讯原理及配置	10
4-1. ETHERCAT 总线网络原理分析	11
4-1-1. EtherCAT 总线网络概述	11
4-1-2. EtherCAT 帧结构	11
4-1-3. EtherCAT 伺服驱动器控制应用协议—CoE 协议	12
4-1-4. EtherCAT 总线通信同步模式—DC (SYNC0 事件同步)	12
4-2. CiA402 数据对象字典与伺服驱动器常用对象	13
4-2-1. CiA402 对象字典概述	13
4-2-2. 伺服驱动器常用对象	14
4-3. ETHERCAT 主从站通讯的配置	22
4-3-1. 安装 EtherCAT 主从站设备描述文件	22
4-3-2. EtherCAT 主站配置	24
4-3-3. EtherCAT 从站配置	29
4-4. 伺服轴电机参数的配置	34
4-4-1. 基本参数配置	34
4-4-2. Scaling/Mapping (“缩放/映射”) 配置	35
4-4-3. I/O Mapping (I/O 映射)	37
4-4-4. IEC Objects	37
4-4-5. 电机轴状态	37
4-4-6. 电机轴信息	38

4-1. EtherCAT 总线网络原理分析

4-1-1. EtherCAT 总线网络概述

EtherCAT，全称 Ethernet for Control Automation Technology，由 Beckhoff 开发，是一种实时以太网用于主站和从站开放式的网络通信。EtherCAT 总线采用常见的连接形式是采用 RJ45 插座、多芯以太网电缆连接，推荐采用超 5 类电缆，可以提高网络的抗干扰能力。

信捷XS3系列控制器仅支持标准的EtherCAT拓扑结构--线型。如果该拓扑结构中任何一个节点出现网络中断的现象，则所有从站通信也会相应中断。网络内部的等效连接关系与通信数据的流动如下图：

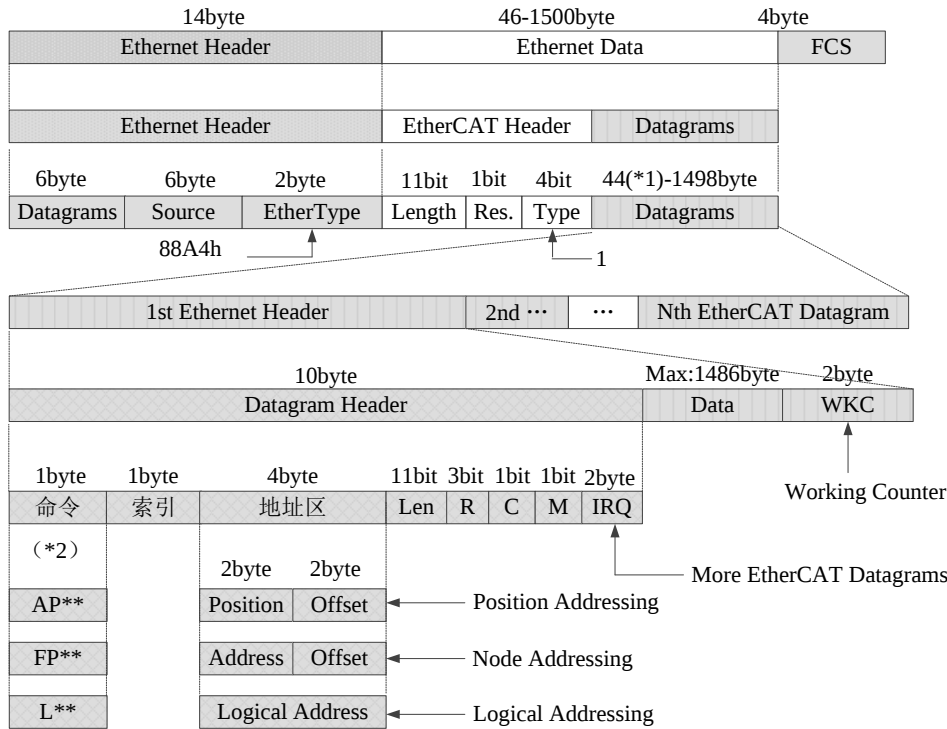


EtherCAT网络有与普通以太网明显不同的方面，在一个网络里只有一个EtherCAT主站，另外是从站内部的网络有专用控制芯片ESC（EtherCAT Slave Controller），它能在一个通信帧中实时收取发给本站通信数据并将本站的应答数据插入该数据帧，使得主站在一个通信帧中即可完成多个从站的访问，大大提高了通信效率。

4-1-2. EtherCAT 帧结构

EtherCAT是基于Ethernet可实时控制的工业用通信协议，只是对IEEE 802.3Ethernet规格进行扩充，并未对基本结构进行任何变更，所以可以转送标准的Ethernet帧内的数据。

EtherNet/EtherCAT 帧结构



注：Ethernet 帧比 64byte 短时，追加 1~32byte。

(Ethernet Header + Ethernet Data + FCS)

因为Ethernet Header的EtherType为「88A4h」,所以将之后的Ethernet Data作为EtherCAT帧来处理。“EtherCAT 帧”是以某种协议来进行定义和解析,只要主站和从站都遵守这个协议,就可以实现数据通信了。一般采用的协议有CANopen Over EtherCAT (CoE),还有Sercos Over EtherCAT (SoE)等。

4-1-3. EtherCAT 伺服驱动器控制应用协议—CoE 协议

XS3的EtherCAT总线通信应用层采用的是CANopen DS402协议,也称CiA402协议,它属于CANopen协议的“伺服与运动控制”行规,该协议行规广泛应用于基于CAN总线、EtherCAT总线网络的运动控制。不同设备厂家按照这个协议规范开发的控制器、伺服驱动器(从站设备)均可配合运行,或可替换使用,给用户增加了选择机会,符合PLCopen规范的宗旨。

其通信机制包括对象字典、PDO(过程数据对象)、SDO(服务数据对象)、与CANopen相似的网络管理。其通讯特点如下所述:

- 为了提高通信效率,主站-从站不是采用问答的方式来访问,而是在总线网络初始化阶段,主站事先把需要发送的数据项目清单给从站,这样从站在接收到主站数据帧后知道如何解析,也可以事先准备好所需要的应答数据。当主站的数据帧到来时,每个从站的网络控制芯片(ESC)就可以拿取发给本站的数据段,供本站的处理器按照配置表进行解析,并在EtherCAT通信帧的合适阶段及时插入本站的应答数据块,返回给主站。
- 将用户所需通信的数据按照实时性要求,分为“过程数据PDO”和“服务数据SDO”,前者PDO 安排为高频度周期性循环收发,后者SDO 则只在有需要的时候才进行通信;
- 伺服驱动器的控制命令参数、运行状态参数、功能码设置参数,最多者达到数百个,每个品牌的伺服参数的命名方法各不相同,为了保证不同品牌的主站、从站都具有互换性,CiA402协议中制定了一个“对象字典OD”,将伺服驱动器中所要用到的所有功能码、运行命令及其设定值意义、运行状态参数及量纲等,都作了具体定义,形成了一个专业的技术规范,不同品牌的设备供应商只要按照这个CiA402协议规范开发的产品,能保证通用性和互换性,就可以与XS3配合运行;
- 主从站之间的通信对象配置,是保证运控功能块功能成功执行的条件。控制器在执行用户程序中MC功能块时,内部需要使用具体的“通信数据对象”来向伺服从站发送命令、读取从站轴状态,编程人员需要在TPDO、RPDO中填入所需要用到的数据对象,主控器才能对伺服从站进行控制,不要有遗漏;
- 用户在编写用户工程时,按控制需要,选择和配置TXPDO、RXPDO数据对象表,运行时主站会自动以通信方式转发给对应的从站;尽量只选择需要用到的配置项,减少无关数据对象的配置项,可以减轻EtherCAT通讯的负荷,利于提高通信效率;
- 服务数据对象SDO 配置项目一般用于系统运行之初对从站设备的功能码初始化,运行过程中也可以通过MC_SDOread之类的功能块进行参数的访问,其通信及时性会比较低,而且会占用额外的EtherCAT通信开销,对于总线负载率比较高的应用中,还会引起同步超时故障,使用要慎重。

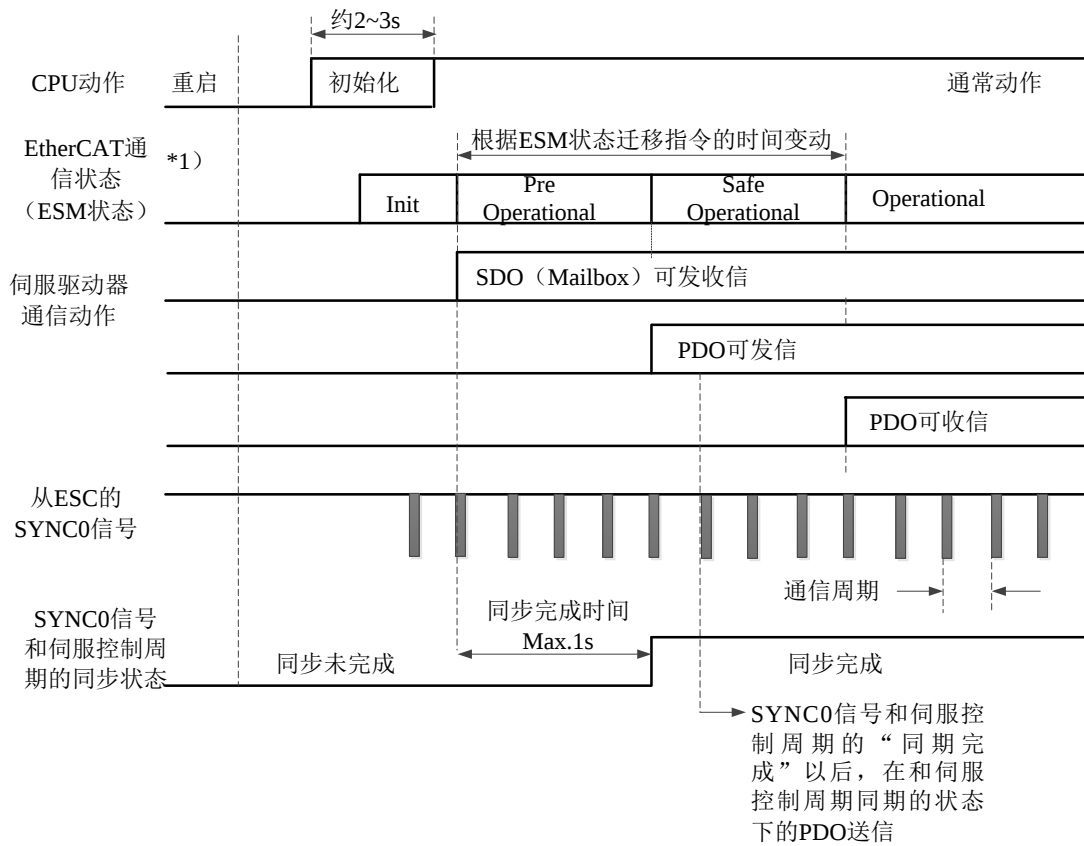
4-1-4. EtherCAT 总线通信同步模式— DC (SYNC0 事件同步)

XS3系列可以选择以下的同步模式。

同步模式	内容	同步方法	特征
DC	SYNC0事件同步	以第1轴的时间为基准同步其他从站的时刻信息	高精度 需要在主站侧进行补偿处理

EtherCAT通信的同步是基于此DC进行的。依据DC从站通过共有相同基准的时钟(System Time)实现同步。因为从站的处理(伺服处理)是开始于SYNC0事件周期,所以总是与SYNC0事件同步。

每个从站内部都有一个高分辨率的内部本地时钟,在EtherCAT 总线初始化阶段,主站会读取每一个从站的当前时间,将第一个从站的本地时间作为网络的“参考时钟”,即可算得每个从站相对于参考时钟的“时钟偏移”,将每个从站的时钟偏移写给对应的从站,以便其修正时钟,消除静态误差。下图表示从控制电源投入到SYNC0事件和从站的处理(伺服处理)的同步完成的过程。



4-2. CiA402 数据对象字典与伺服驱动器常用对象

4-2-1. CiA402 对象字典概述

XS3系列控制器的EtherCAT网络通讯都是由系统自动完成的，并不需要用户过多的干预，用户只需理解CiA402对象概念，掌握伺服轴常用的“对象”类型，进行TXPDO、RXPDO 配置表的对象选择就可以了。

CiA402协议将伺服运动控制应用中具有代表性的设置参数、控制参数、状态参数等项目，归纳整理为具有固定编号（索引号+子索引号）的“对象”，完整的对象定义表就是“对象字典”。

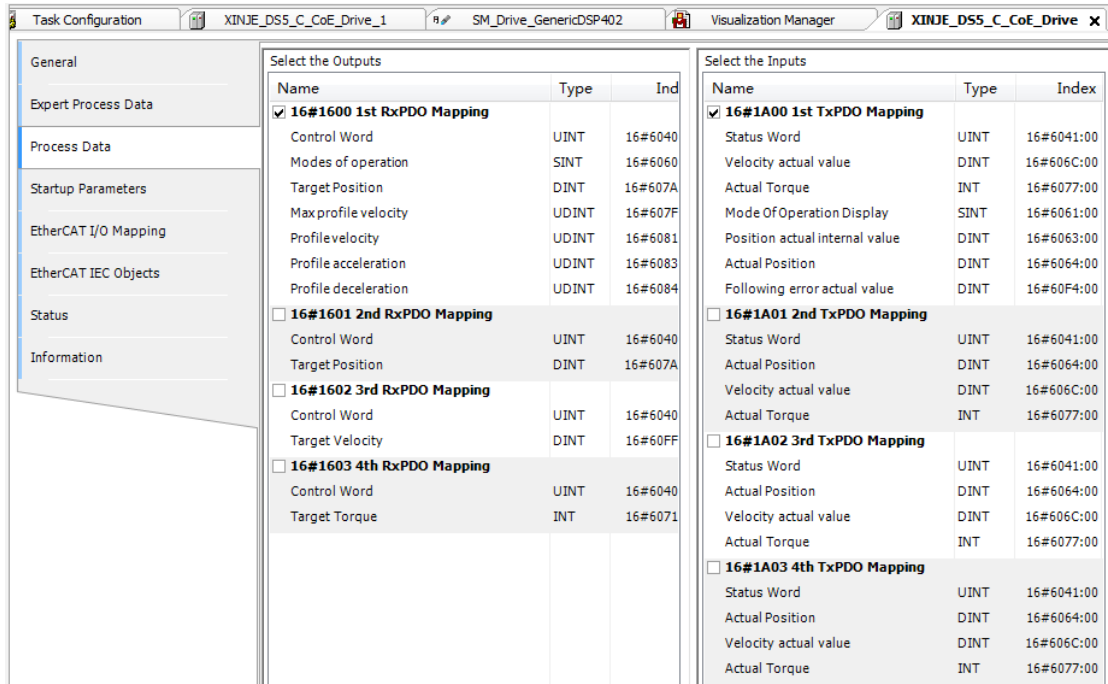
CiA402对象类型按属性分成索引号段，如下表：

CiA402规定的对象字典		说明
Index	内容	
0000h~0FFFh	数据类型区域	协议类型描述、制造商信息、行规类型描述、配置表描述信息等。由设备制造商进行信息初始化，配置信息由系统软件自动完成，无需控制器用户干预。
1000h~1FFFh	COE通信区域	
2000h~5FFFh	厂商自定义区域	由设备商自行定义对象的功能属性，设备商可以将主索引号设计成伺服驱动器的功能码，用设置功能码参数，用于静态参数的设置。
6000h~9FFFh	Profile区域	行规定义的数据对象，用于设备的控制与监视。
A000h~FFFFh	保留	-

由上表可以看出，运动控制所需的对象，主要在（0x6000~0x9FFF）索引号段部分，如果是希望SDO配置修改伺服功能码，需要关注（0x2000~0x5FFF）索引号段部分。

对于PDO 配置表，用户只需添加控制器运行时需要用到的数据对象，不需要填写具体的参数值或变量名，CODESYS软件编译时，会自动将MC功能块中的变量与PDO的对象进行关联；对于SDO 配置表，一般用于控制器初始化伺服功能码的操作（写操作），其中的写入值为确定的常数量，因此填写的常数必需符合DS402规范定义，有些则是按伺服驱动器内部特有的功能码定义的常数值。

下图为CODESYS中PDO 配置表的画面，其中的每种控制参数的索引号、数据类型，均由CiA402 行规规定，通过“索引号”我们可以在“对象字典OD”中反查得知参数名称、参数类型等信息。



4-2-2. 伺服驱动器常用对象

XS3控制器控制伺服的运行，大致上有这样的几种命令类型：

- ① 控制伺服的运行状态，如使能、原点回归、启停运行、告警复位等；
- ② 设定伺服的运行模式，如设为位置模式、速度模式、力矩模式；
- ③ 设定伺服运行的目标位置、运行速度、输出力矩等；
- ④ 读取伺服系统的运行状态，如运行状态、运行模式、位置、当前速度、输出力矩等；
- ⑤ 设定或修改伺服系统的功能码参数，运行约束参数等。

要完成这些运动控制操作，有一些常用的数据对象，编程时必需在PDO或SDO配置表中设定，还有一些数据对象，则是根据用户程序中使用到的功能，才酌情添加。以下则对一些常用对象进行阐述：

1、控制字6040H（Control word）

该对象是主控制器单元控制伺服的运行状态的命令字，用于控制伺服的使能、启停、告警复位等运行状态，这是最基本的控制命令字。

Index	Sub-Index	Name/Description	Range	DateType	Access	PDO	Op-mode		
6040h	00h	Controlword	0~65535	U16	rw	RxPDO	All		
		设定对PDS状态转换等伺服驱动器的控制命令。							
		bit信息							
		15	14	13	12	11	10	9	8
		r					oms	h	
		7	6	5	4	3	2	1	0
		fr	r		eo	qs	ev	so	
		r = reserved（未对应）			fr = fault reset				
		oms = operation mode specific			eo = enable operation				
		（控制模式依存bit）			qs = quick stop				
h = halt			ev = enable voltage						
			so = switch on						

控制字的每一个bit位单独赋值无意义，必须与其他位共同构成某一控制指令。

bit0~bit3和bit7在各伺服模式下意义相同，必须按顺序发送命令，才可将伺服驱动器按照CiA402状态机切换流程引导入预计的状态，每一命令对应一确定的状态。

Command	bits of the controlword				
	bit7	bit3	bit2	bit1	bit0
	Fault reset	Enable operation	quick stop	Enable voltage	Switch on
Shutdown	0	-	1	1	0
Switch on	0	0	1	1	1
Switch on + Enable operation	0	1	1	1	1
Enable operation	0	1	1	1	1
Disable voltage	0	-	-	0	-
Quick stop	0	-	0	1	-
Disable operation	0	0	1	1	1
Fault reset	0->1	-	-	-	-

bit4~bit6与各伺服模式相关。以下表示控制模式（Op-mode）固有的oms bit的变动。

Op-mode	Bit9	Bit6	Bit5	Bit4
pp	change on set-point	absolute /relative	change set immediately	new set-point
pv	-	-	-	-
tq	-	-	-	-
hm	-	-	-	start homing
csp	-	-	-	-
csv	-	-	-	-
cst	-	-	-	-

2、状态字6041H（Control word）

从站（伺服驱动器）的状态确认，通过6041h（状态字）进行。

Index	Sub-Index	Name/Description	Range	Date Type	Access	PDO	Op-mode																																
6041h	00h	Statusword	0-65535	U16	ro	TxPDO	All																																
表示伺服驱动器的状态。																																							
bit信息																																							
<table border="1"> <tr> <td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td></tr> <tr> <td colspan="2">r</td><td colspan="2">oms</td><td>ila</td><td>oms</td><td>rm</td><td>r</td></tr> <tr> <td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr> <tr> <td>w</td><td>sod</td><td>qs</td><td>ve</td><td>f</td><td>oe</td><td>so</td><td>rtso</td></tr> </table>								15	14	13	12	11	10	9	8	r		oms		ila	oms	rm	r	7	6	5	4	3	2	1	0	w	sod	qs	ve	f	oe	so	rtso
15	14	13	12	11	10	9	8																																
r		oms		ila	oms	rm	r																																
7	6	5	4	3	2	1	0																																
w	sod	qs	ve	f	oe	so	rtso																																
r = reserved（未对应） oms = operation mode specific（控制模式依存bit） ila = internal limit active rm = remote w = warning sod = switch on disabled qs = quick stop ve = voltage enabled f = fault oe = operation enabled so = switched on rtso = ready to switch on																																							

bit6,5,3-0（switch on disabled/quick stop/fault/operation enabled/switched on/ready to switch on）：根据此Bit可以确认PDS的状态。以下表示状态和对应的bit。

StatusWord	PDS State	
xxxx xxxx x0xx 0000 b	Not ready to switch on	初始化未完成状态
xxxx xxxx x1xx 0000 b	Switch on disabled	初始化完成状态
xxxx xxxx x01x 0001 b	Ready to switch on	初始化完成状态
xxxx xxxx x01x 0011 b	Switched on	伺服使能关闭/伺服准备
xxxx xxxx x01x 0111 b	Operation enabled	伺服使能开启
xxxx xxxx x00x 0111 b	Quick stop active	立即停止
xxxx xxxx x0xx 1111 b	Fault reaction active	异常（报警）判断
xxxx xxxx x0xx 1000 b	Fault	异常（报警）状态

bit4（voltage enabled）：1的情况下，表示电源电压印加到PDS。

bit5（quick stop）：0的情况下，表示PDS接收quick stop要求。quick stop的bit逻辑是在0下有效。请注意执行其他的bit逻辑和相反的动作。

bit7（warning）：1的情况下，表示警告正在发生。警告时PDS状态不变，电机也继续动作。

bit9（remote）：0（local）的情况下，表示6040（Controlword）无法处理的状态。1（remote）的情况下，表示6040（Controlword）处于可处理的状态。ESM状态是转换到PreOP以上时变为1。

bit13,12,10（operation mode specific）：以下，表示控制模式固有的oms bit的变化。（详情请参照各控制模式的关联对象的章节）

Op-mode	bit13	bit12	Bit10
Pp	following error	set-point acknowledge	target reached
Pv	-	speed	target reached
Tq	-	-	target reached
Hm	homing error	homing attained	target reached
Csp	following error	drive follows command value	-
Csv	-	drive follows command value	-
Cst	-	drive follows command value	-

bit11（internal limit active）：内部限制的主要原因是发生时6041h（Statusword）的bit11（internal limit active）变为1。

bit15,14（reserved）：此bit未使用（0固定）。

3、伺服运行模式6060H（Modes of operation）

控制模式的设定通过6060h（Modes of operation）进行。

Index	Sub-Index	Name/Description	Range	DateType	Access	PD0	Op-mode
6060h	00h	Mode of operation	-128~127	I8	rw	RxPDO	All

设定伺服驱动器的控制模式。 非对应的控制模式设定禁止。			
bit	Mode of operation	缩写	对应
-128~ -1	Reserved	-	-
0	No mode changed/No mode assigned (没控制模式改变/没控制模式分配)	-	-
1	Profile position mode (Profile位置控制模式)	pp	YES
3	Profile velocity mode (Profile速度控制模式)	pv	YES
4	Torque profile mode (Profile转矩控制模式)	tq	YES
6	Homing mode (原点复位位置模式)	hm	YES
8	Cyclic synchronous position mode (Cyclic位置控制模式)	csp	YES
9	Cyclic synchronous velocity mode (Cyclic速度控制模式)	csv	YES
10	Cyclic synchronous torque mode (Cyclic转矩控制模式)	cst	YES
11~127	Reserved	-	-

因为6060h (Modes of operation) 是default= (No mode change/no mode assigned)，电源投入后请一定设定使用的控制模式值。6060h的设定值是0并且6061h的设定值是0时，如果将PDS状态迁移到Operation enabled，发生E-881（控制模式设定异常保护）。

初期状态6060h=0 (No mode assigned) 转换到可支持的控制模式 (pp, pv, tq, hm, csp, csv, cst) 后，再次设定6060h=0的情况作为“No mode changed”，控制模式的切换无法执行。（保持前次的控制模式）。

伺服驱动器处于任何状态下，从内部速度模式 (PV)、内部转矩模式 (tq)、周期同步速度模式 (CSV)、周期同步转矩模式 (CST) 切入其他模式后，首先执行斜坡停机，停机完成后，可切入其他模式。

4、伺服运行模式确认6061h (Modes of operation display)

伺服驱动器内部的控制模式的确认根据6061h (Modes of operation display) 执行。6060h (Modes of operation) 设定后，请确认通过检测设定此对象动作是否可行。

Index	Sub-Index	Name/Description	Range	DateType	Access	PD0	Op-mode
6061h	00h	Mode of operation display	-128-127	I8	ro	TxPDO	All

		表示现在的控制模式。			
		bit	Mode of operation	缩写	对应
		-128~-1	Reserved	-	-
		0	No mode changed/No mode assigned (没控制模式改变/没控制模式分配)	-	-
		1	Profile position mode (Profile位置控制模式)	pp	YES
		3	Profile velocity mode (Profile速度控制模式)	pv	YES
		4	Torque profile mode (Profile转矩控制模式)	tq	YES
		6	Homing mode (原点复位位置模式)	hm	YES
		8	Cyclic synchronous position mode (Cyclic位置控制模式)	csp	YES
		9	Cyclic synchronous velocity mode (Cyclic速度控制模式)	csv	YES
		10	Cyclic synchronous torque mode (Cyclic转矩控制模式)	cst	YES
		11-127	Reserved	-	-

5、目标位置607Ah (Target position)

由主控器发送的伺服运行的目标位置命令。伺服多数情况下以位置模式运行，在XG3 的MC应用中，多以周期同步位置模式（CSP），控制器命令伺服在下一EtherCAT 周期运行的目标位置，其量纲为用户设置的物理量纲。

Index	Sub-index	Name	Units	Range	Datatype	Access	PDO
607Ah	00h	Target position	指令单位	-2147483648~ 2147483647	I32	rw	RxPDO

6、目标速度60FFh (Target velocity)

当伺服以速度模式运行时（内部速度模式PV与周期同步速度模式CSV），需要设置该速度指令。

Index	Sub-index	Name	Units	Range	Datatype	Access	PDO
60FFh	00h	Target velocity	指令单位/s	0~4294967295	U32	rw	RxPDO

7、目标转矩6071h (Target Torque)

当伺服以转矩模式运行时（内部转矩模式tq与周期同步转矩模式CST），需要设置该转矩指令，量纲为电机额定转矩的百分比（0.1%）。

Index	Sub-index	Name	Units	Range	Data type	Access	PDO
6071h	00h	Target torque	0.1%	-32768~32767	I16	rw	RxPDO

8、最大内部速度607Fh (Max profile velocity)

该对象用于设置伺服的最大运行速度，在内部速度模式（PV）下，作为最大速度的限制；在同步力矩模式（CST）、内部力矩模式（tq）时，为避免电机转速过高，造成机械冲击，使用该设定值作为最大速度的限制。

例如，当伺服运行于力矩模式，当电机的实际的负荷转矩小于给定转矩时，运行转速会越来越快，当运行转速达到最大内部速度Max profile velocity后，速度会被限制在该速度值。

注意：该对象在周期同步位置模式下无效。

Index	Sub-index	Name	Units	Range	Datatype	Access	PDO
607Fh	00h	Max profile velocity	指令单位/s	0~4294967295	U32	rw	RxPDO

9、位置反馈6064h (Position actual value)

伺服多数情况下以位置模式运行，控制器必需实时监测伺服的当前位置，主控器通过该对象读取伺服当前的实际位置。该对象为PDO配置表的必选项目之一。

Index	Sub-index	Name	Units	Range	Datatype	Access	PDO
6063h	00h	Position actual internal value	pulse	-2147483648~ 2147483647	I32	ro	TxPDO
6064h	00h	Position actual value	指令单位	-2147483648~ 2147483647	I32	ro	TxPDO

位置反馈6064h (指令单位) × 位置因子6091h (齿轮比) = 实际内部位置6063h (脉冲单位)

10、速度反馈606Ch (Velocity ActualValue)

显示伺服内部速度反馈。该对象为PDO 配置表的必选项目之一。

Index	Sub-index	Name	Units	Range	Datatype	Access	PDO
606Ch	00h	Velocity actual value	指令单位/s	-2147483648~ 2147483647	I32	ro	TxPDO

11、转矩反馈6077h (Torque Actual Value)

显示伺服内部转矩反馈，量纲为伺服电机额定值的百分比（0.1%）。该对象为PDO 配置表的常选项目之一。

Index	Sub-index	Name	Units	Range	Datatype	Access	PDO
6077h	00h	Torque actual value	0.1%	-32768~32767	I16	ro	TxPDO

12、探针功能相关对象

通过伺服的探针功能，可以检测特定DI信号变化时的伺服位置信号，伺服以中断方式记录DI信号变动时的伺服位置，供上位机读取，这样可以提高系统控制的精度。

注意：该功能的详细使用解释请详见EtherCAT通讯规格篇手册。

(1) 探针功能设定60B8h (Touch probe function)

该对象用于主控器设定伺服的探针功能方式及启停。

Index	Sub-index	Name	Units	Range	Datatype	Access	PDO	OP-mode
60B8h	00h	Touch probe function	-	0~65535	U16	rw	RxPDO	ALL
执行Touch probe功能的设定。								

对应bit说明

bit	vaule	note	
0	0	Switch off touch probe 1	Touch Probe 1 执行/停止
	1	Enable touch probe 1	
1	0	Trigger first event	Touch Probe 1 事件模式选择
	1	Continuous	
2	0	Trigger with touch probe 1 input	Touch Probe 1 触发选择（外部输入/Z相）
	1	Trigger with zero impulse signal of position encoder	
3	-	Reserved	未使用
4	0	Switch off sampling at positive edge of touch probe 1	Touch Probe 1 上升沿选择
	1	Enable sampling at positive edge of touch probe 1	
5	0	Switch off sampling at negative edge of touch probe 1	Touch Probe 1

bit	vaule	note	
	1	Enable sampling at negative edge of touch probe 1	下降沿选择
6-7	-	Not Supported	未使用
8	0	Switch off touch probe 2	Touch Probe 2 执行/停止
	1	Enable touch probe 2	
9	0	Trigger first event	Touch Probe 2 事件模式选择
	1	Continuous	
10	0	Trigger with touch probe 2 input	Touch Probe 2 触发选择（外部输入/Z相）
	1	Trigger with zero impulse signal of position encoder	
11	-	Reserved	未使用
12	0	Switch off sampling at positive edge of touch probe 2	Touch Probe 2 上升沿选择
	1	Enable sampling at positive edge of touch probe 2	
13	0	Switch off sampling at negative edge of touch probe 2	Touch Probe 2 下降沿选择
	1	Enable sampling at negative edge of touch probe 2	
14-15	-	Not Supported	未使用

注意：

- ①目前暂不支持 Z 相触发模式，只支持外部信号作为触发源；
- ②在同一探针下，请勿同时设定上升沿和下降沿。

（2）探针状态60B9h（Touch probe status）

Index	Sub-index	Name	Units	Range	Datatype	Access	PD0	OP-mode
60B9h	00h	Touch probe status	-	0~65535	U16	ro	TxPDO	ALL
表示Touch probe功能的状态。								

对应bit说明

bit	vaule	note	
0	0	Touch probe 1 is switch off	Touch Probe 1动作停止
	1	Touch probe 1 is enabled	Touch Probe 1动作中
1	0	Touch probe 1 no positive edge value stored	上升沿Touch Probe 1未完成状态
	1	Touch probe 1 positive edge value stored	上升沿Touch Probe 1完成状态
2	0	Touch probe 1 no negative edge value stored	下降沿Touch Probe 1未完成状态
	1	Touch probe 1 negative edge value stored	下降沿Touch Probe 1完成状态
3-5	-	Reserved	未使用
6-7	-	Not Supported	未使用
8	0	Touch probe 2 is switch off	Touch Probe 2动作停止
	1	Touch probe 2 is enabled	Touch Probe 2动作中
9	0	Touch probe 2 no positive edge value stored	上升沿Touch Probe 2未完成状态
	1	Touch probe 2 positive edge value stored	上升沿Touch Probe 2完成状态
10	0	Touch probe 2 no negative edge value stored	下降沿Touch Probe 2未完成状态
	1	Touch probe 2 negative edge value stored	下降沿Touch Probe 2完成状态
11-13	-	Reserved	未使用
14-15	-	Not Supported	未使用

（3）获取的箝位位置0x60BA~0x60BD（Touch probe 1/2 positive value）

表示获取的箝位位置。

Index	Sub-index	Name	Units	Range	Data type	Access	PDO	OP-mode
60BAh	00h	Touch probe pos1 pos value	指令 单位	-2147483648~ 2147483647	I32	ro	TxPDO	ALL
		表示Touch probe1的上升沿箝位位置。						
60BBh	00h	Touch probe pos1 neg value	指令 单位	-2147483648~ 2147483647	I32	ro	TxPDO	ALL
		表示Touch probe1的下降沿箝位位置。						
60BCh	00h	Touch probe pos2 pos value	指令 单位	-2147483648~ 2147483647	I32	ro	TxPDO	ALL
		表示Touch probe2的上升沿箝位位置。						
60BDh	00h	Touch probe pos2 neg value	指令 单位	-2147483648~ 2147483647	I32	ro	TxPDO	ALL
		表示Touch probe2的下降沿箝位位置。						

13、回原点功能相关对象

对于采用相对定位方式的应用中，需要首先进行回零操作，便于伺服驱动器、运动控制器确定位置的参考零点。

(1) 回零方式6098h (Homing method)

在通过EtherCAT通信让伺服回零操作前，需要设定伺服的回零方式，主站通过该数据对象设定伺服的回零方式，请参见附录中的归零方式详细说明。

Index	Sub-Index	Name/Description	Range	DateType	Access	PDO	Op-mode
6098h	00h	Homing method	-128~127	I8	rw	RxPDO	All
		设定原点方法					

注意：当选用绝对位置编码器方式，可采用第35种回零方式，回零操作的结果是以当前位置作为零点，电机并不转动。主站通过该数据对象设定伺服的回零方式，请详见EtherCAT通讯规格篇手册。

(2) 回零速度6099h (Homing speeds)

在伺服驱动器在寻找原点信号时的运行速度可以通过该对象进行设定，该对象有2个子索引。

Index	Sub-Index	Name/Description	Range	Date Type	Access	PDO	Op-mode
6099h	-	Homing speeds	-	-	-	-	-
		设定原点复位位置控制模式(hm)时的速度。					
	00h	Number of entries	2	U8	ro	NO	HM
		表示6099h (Homing speeds) 的Sub-Index的数					
	01h	Speed during search	0~4294967295	U32	rw	RxPDO	HM
		设定到Switch信号检出的动作的速度。 最大值是用内部处理的6080h (Max motor speed) 和2147483647任意小的一方进行限制。					
	02h	Speed during search for zero	0~4294967295	U32	rw	RxPDO	HM
		设定到原点检出的动作速度。 如果Switch信号的边沿作为原点检出位置，为了减小检出误差请设定尽量小的值。 最大值是用内部处理的6080h (Max motor speed) 和2147483647任意小的一方进行限制。					

(3) 回零加速度609Ah (Homing acceleration)

在伺服驱动器在寻找原点信号时的运行加速度可以通过该对象进行设定。

Index	Sub-Index	Name/Description	Range	Date Type	Access	PDO	Op-mode
609Ah	00h	Homing acceleration	0~4294967295	U32	rw	RxPDO	All
设定原点复位位置控制模式(hm)时的加速度以及减速度。 原点复位位置控制模式(hm)的减速度兼用于此对象。 各Homing method最终停止时(原点位置检出时)无需使用此对象的设定, 伺服锁定停止。 如果设定为0, 内部处理作为1处理。							

14、位置偏差反馈60F4h (Following error actual value)

该对象用于指示当前位置与目标位置的偏差, 量纲为指令单位, 应用于CSP、PP、HM三种模式, 用于判断位置到达情况。

Index	Sub-index	Name	Units	Range	Datatype	Access	PDO
60F4h	00h	Following error actual value	指令单位	-2147483648~ 2147483647	I32	ro	TxPDO

$60F4h \text{ (Following error actual value)} = 6062h \text{ (Position demand value)} - 6064h \text{ (Position actual value)}$

15、伺服告警码603Fh (Error code)

此对象给出驱动器最近发生的故障码或警告码。该对象为PDO配置表的常用项目之一。被定义的值(FF00h~FFFFh)的下位8bit如下表表示伺服异常(报警)的报警编号的主码。(不读取报警编号的辅码。)另外, 报警编号的主码用16进制数表示。

Index	Sub-Index	Name/Description	Range	Date Type	Access	PDO	Op-mode
603Fh	00h	Error code	0~65535	U16	ro	TxPDO	All
现在伺服驱动器发生的报警(只有主编号)。 报警未发生时, 显示0000h。 报警发生时, 显示报警。 FF**h 报警(主)编号(00h~FFh) (例) FF03h ... 03h=3d E-030(过压保护)发生 FF55h ... 55h=85d E-850(TxPDO配置异常保护)、E-851(RxPDO配置异常保护)其中任意一个发生 作为例外, E-817(SyncManager2/3设定异常)的情况下, 显示A000h。							

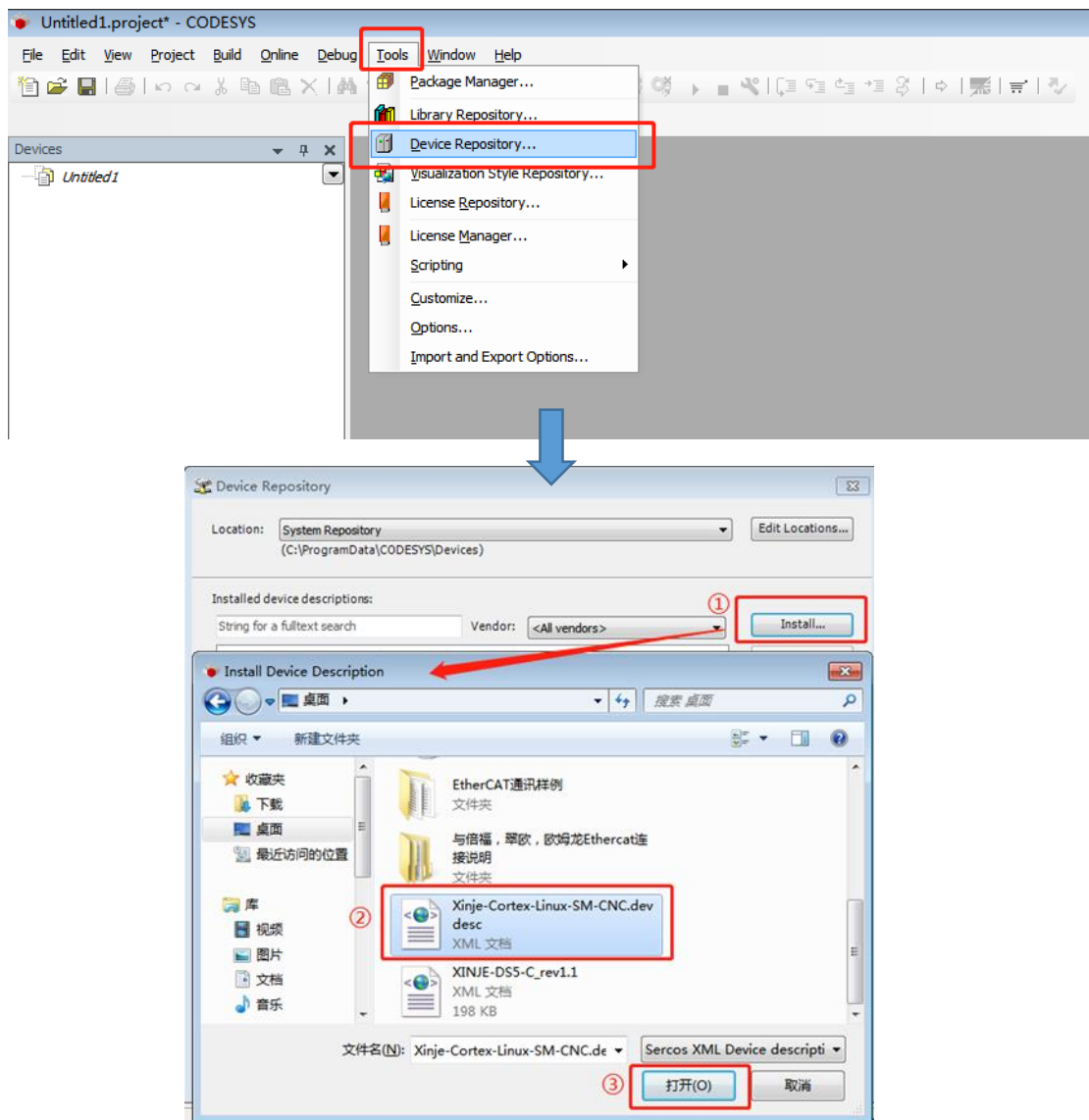
4-3. EtherCAT 主从站通讯的配置

首先添加EtherCAT主从站设备描述文件, 然后配置EtherCAT主站, 随后通过在主站卡菜单下添加从站, 从而完成配置。这里以信捷XS3和信捷DS5C伺服的通信为例, 对主从站通信配置进行如下详解。

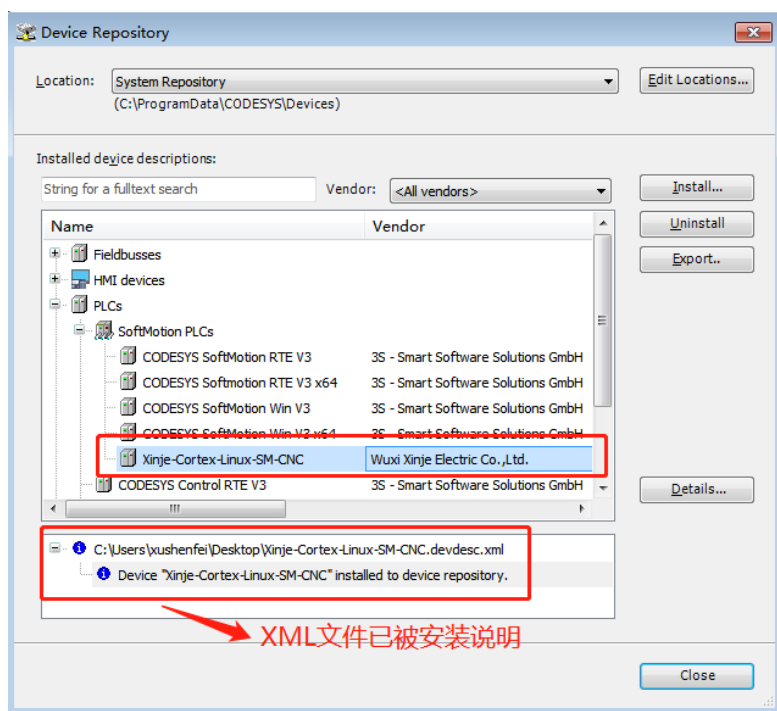
4-3-1. 安装 EtherCAT 主从站设备描述文件

为了在设备目录中插入和配置EtherCAT设备, 主站和从站必须使用硬件提供的设备描述文件, 通过“Device Repository (设备库)”对话框安装。主站的设备描述文件(*.devdesc.xml)定义了可以插入的从站。从站的描述为“xml”文件格式(文件类型: EtherCAT XML Device description Configuration Files)。

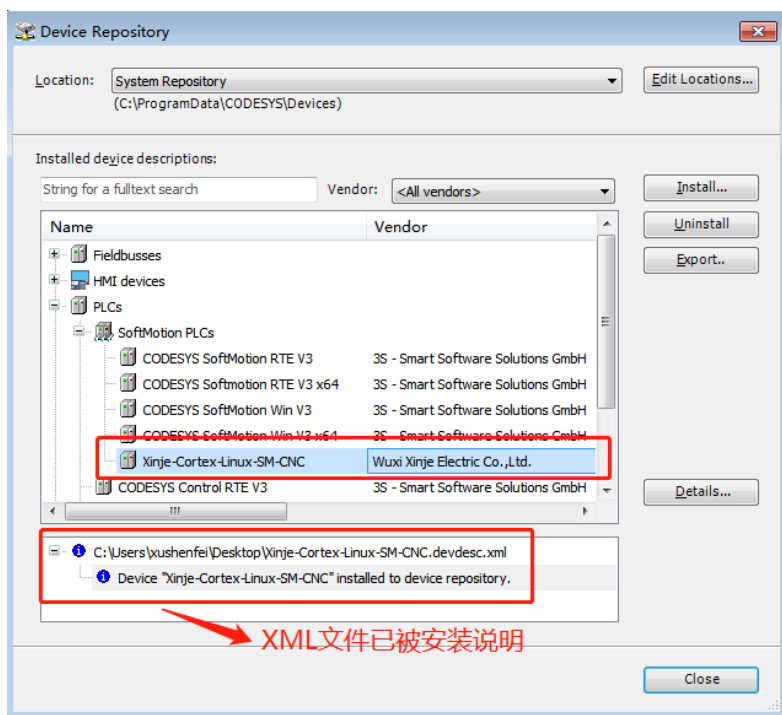
在应用的工程中打开工具设备库, 依次点击“Tools→Device Repository”, 在弹出的对话框中点击“安装 Install”, 选择 XML 文件所在的路径找到 XML 文件, 选中后点击打开。如下图所示:



打开后即安装完成，如下图所示。



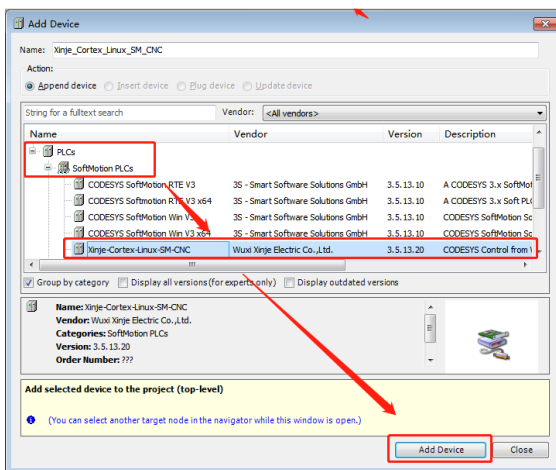
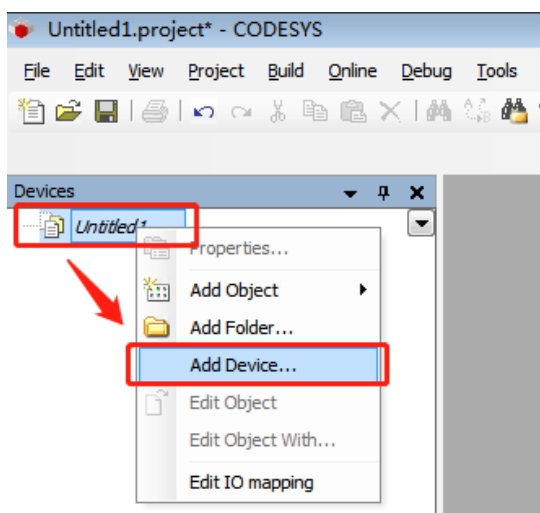
同理，用同样的方式安装从站XML文件（XINJE-DS5-C_rev1.1）。



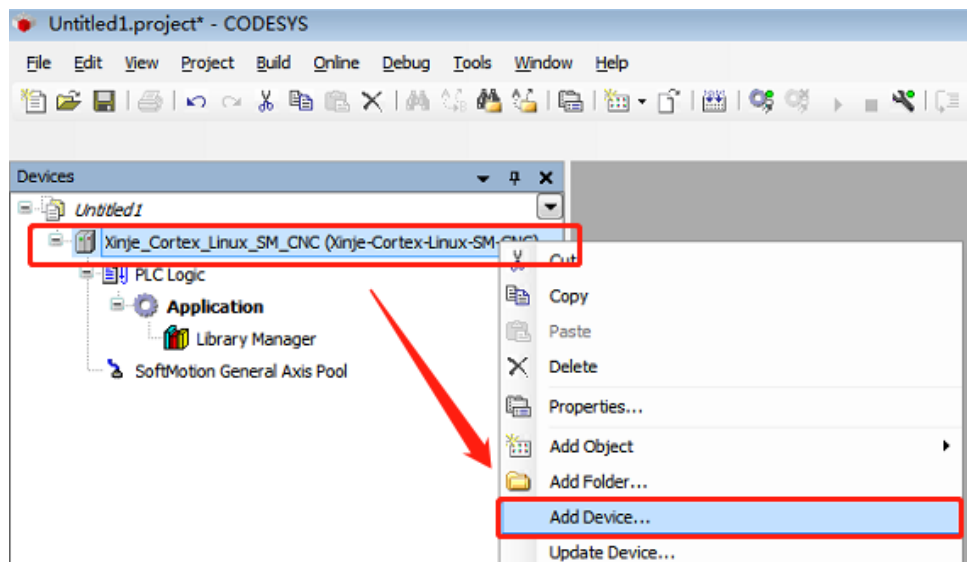
4-3-2. EtherCAT 主站配置

1、添加主站

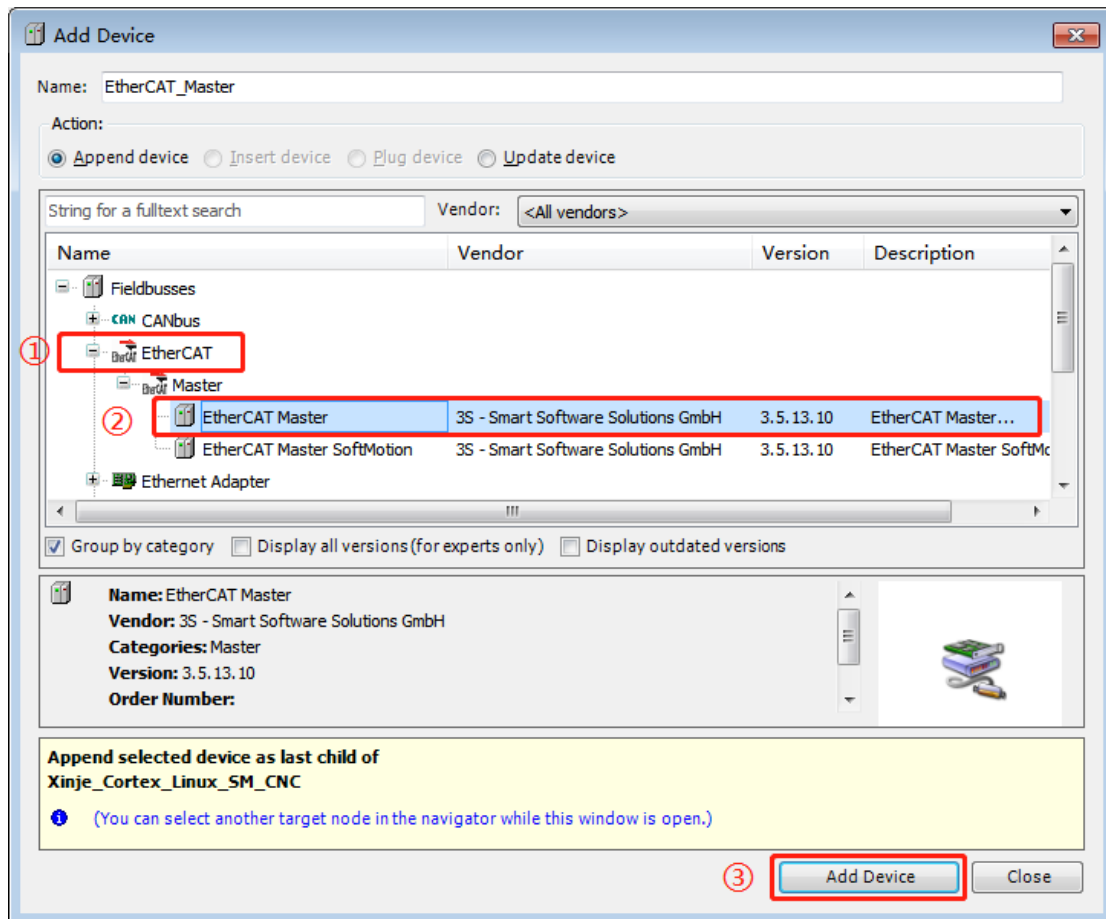
首先右击Untitled，点击Add Device添加设备，在“Add Device”对话框中选择“PLCs→SoftMotionPLCs→Xinje –Cortex-Linux-SM-CNC”，最后点击Add Device添加PLC。



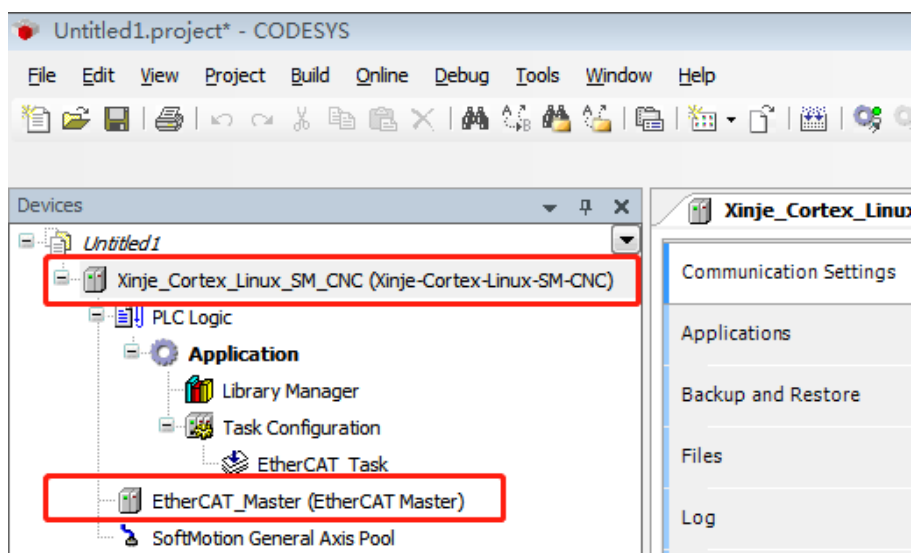
添加PLC完成后，在界面右侧就会出现设备管理器，选中Xinje –Cortex-Linux-SM-CNC右击，点击Add Device添加设备。



在“Add Device”对话框中选择“EtherCAT—Master—EtherCAT Master”，最后点击Add Device添加。

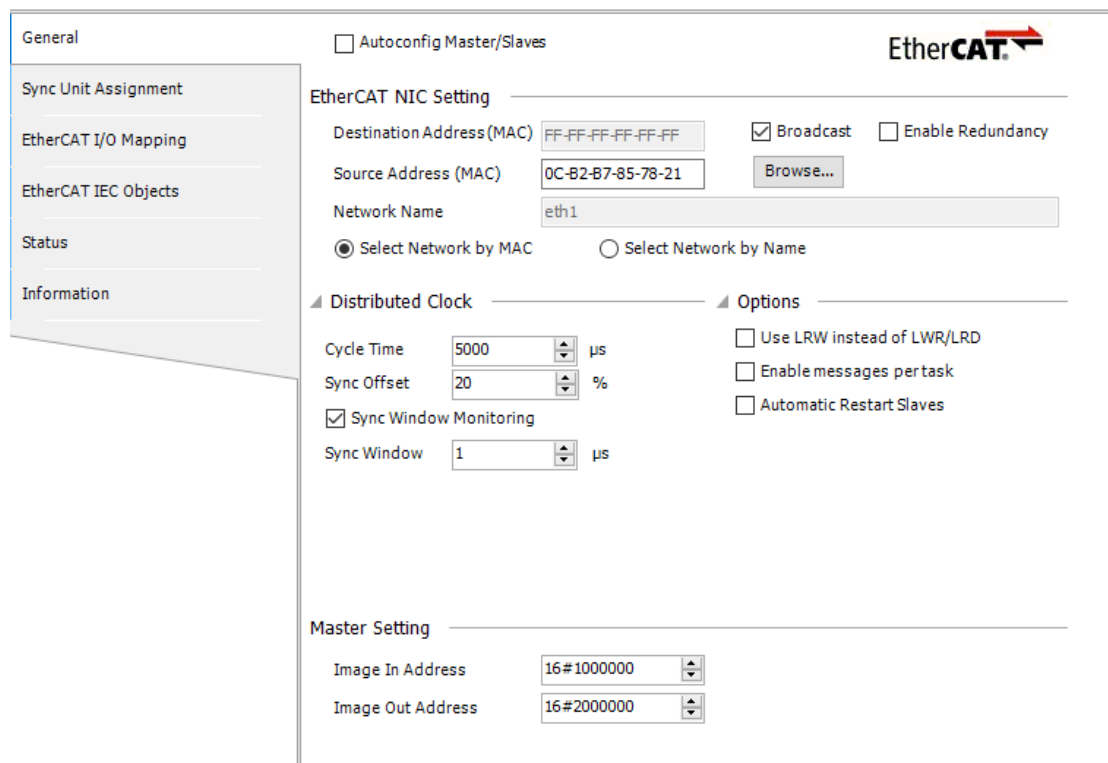


添加主站设备完成，如下图所示：



2、EtherCAT主站配置

主站添加后，可以通过点击选中主站从而对其进行配置。如图为EtherCAT_Master “General” 选项卡中的设置，用于配置主站、分布式时钟及从站相应设置。



【Autoconfig Master/Slaves（自动配置主站/从站）】:

主从站的配置，大部分是基于设备描述文件和隐式计算自动完成的。勾选后FMMU/Sync（现场总线内存管理单元/同步管理器）设置的对话框不可用。

【EtherCAT NIC Setting】:

- Destination Address（目的地址）（MAC）：为接受 EtherCAT 报文的目标地址，如果“Broadcast（广播）”选项被激活，则不需要输入目标地址，系统会自动通过广播搜索目标地址。
- Source Address（源地址）（MAC）：PLC 网络接口的 MAC 地址，可以选择“Select Network by MAC（根据 MAC 选择网络）”或者选择“Select Network by Name（根据名称选择网络）”。用户可以选择“Browse（浏览）”，选择想要设置的源地址。
- Enable Redundancy（启用冗余）：该选项使能后，则正式开启 EtherCAT 冗余模式，其支持环形拓扑结

构（信捷 XS3 不支持此功能）。

【Disrtibuted Clock（分布时钟）】:

- Cycle Time（循环时间）：**一般设为默认配置**。如果分布式时钟功能被激活，主站发送将会根据该循环时间向从站发送相应数据报文。因此数据交换可以实现精确同步，在分布式过程中要求同步动作时（例如几个伺服轴执行同时联动任务），此功能尤为重要的。可以在网络范围内提供信号抖动小于 1 微秒的主时钟。一般设为默认配置。
- Sync offset（同步偏移）：**一般设为默认配置**。通常当 PLC 任务开始 20%后，同步报文开始影响从站，这就意味着 PLC 的任务周期可以有 80%的延迟，在此延迟内不会有数据会丢失。
- Sync Window Monitoring（同步窗口监控）：如果该选项打开，可以监控从站的同步状态。
- Sync Window（同步窗口）：用于监控同步窗口的时间。以 μs 为单位。如果所有的从站在同步窗口时间内，则变量 xSyncInWindow（IoDrvEthercat）将会被置为 TRUE，否则为 FALSE。

【Option（选项）】:

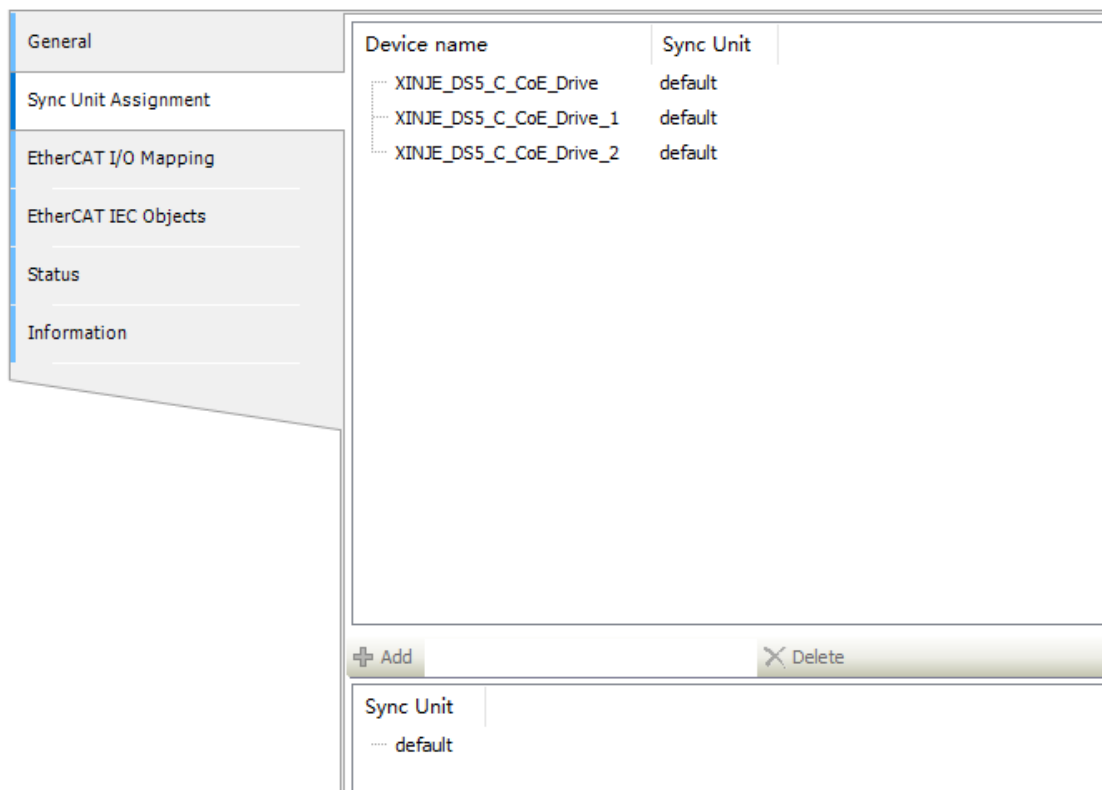
- Use LRW instead of LWR/LRD（使用 LRW 代替 LWR/LRD）：激活此项会使能从设备-从设备的通信。组合读写命令（LRW）会替代单独的读（LR）和写命令（LRW）。
- Enable messages per task（在每个任务中激活消息）：激活此项，处理输入输出信息的读写命令可以由不同任务完成。
- Automatic Restart Slaves（自动重启从站）：激活此项，主机在通信终止后立即尝试重启从站。

【Mater Setting（主站设置）】: 只有在自动配置选项被禁用时才能编辑，否则将自动执行这些设置，并且在界面中不可见。

- Image In Address（输入地址映射）：第一个从站的输入数据的第一个逻辑地址。
- Image Out Address（输出地址映射）：第一个从站的输出数据的第一个逻辑地址。

3、EtherCAT_Master “Sync Unit Assignment（同步单元分配）” 选项卡

如下图为 EtherCAT_Master “Sync Unit Assignment（同步单元分配）” 选项卡，此选项卡显示了插入到主站下的所有从站，并为同步单元分配了一个任务。



【Device name（设备名称）】: 从站名称。

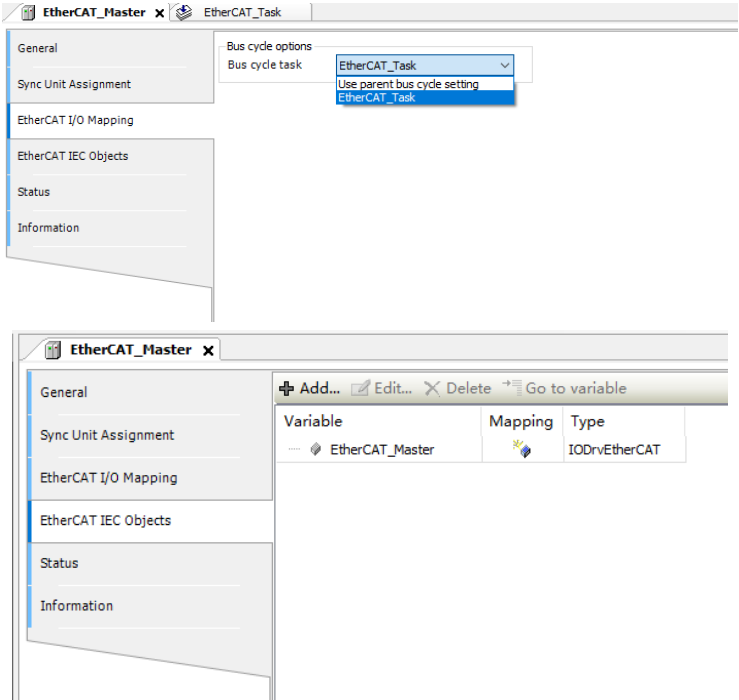
【Sync Unit（同步单元）】: 所选同步单元的名称。

【Add（添加）】：创建新的同步单元。

【Delete（删除）】：移除所选同步单元。

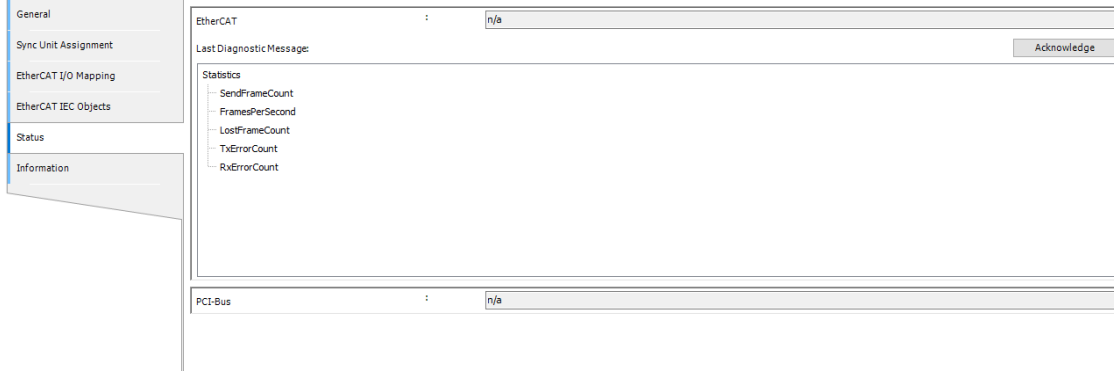
4、EtherCAT I/O映射

选项卡 EtherCAT I/O Mapping（EtherCAT I/O 映射）中是由 EtherCAT 主站提供的 I/O 映射表，PLC 中的变量可以连接其中的 I/O，为用户提供控制的接口，如图所示。



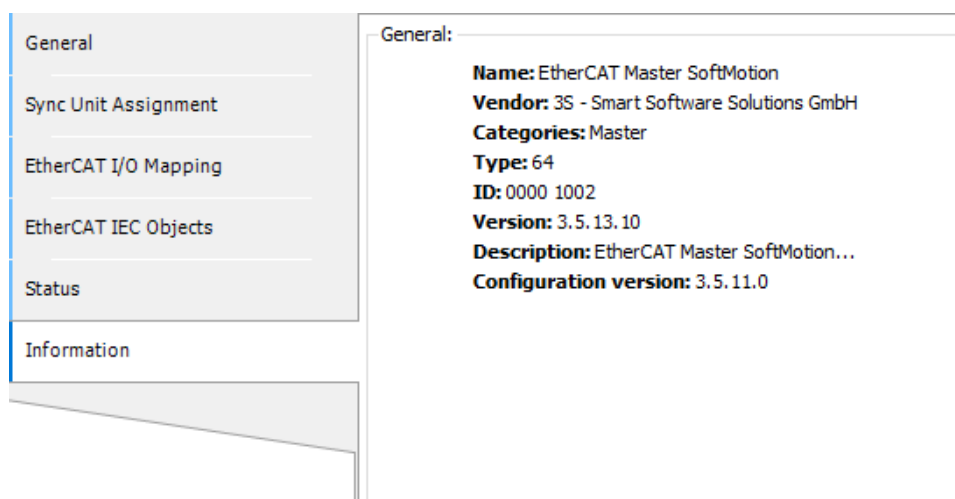
5、主站状态

如下图为 EtherCAT_Master 中的“Status（状态）”选项卡，用于显示主站的状态信息。例如“正在运行”或“停止”，以及来自各自设备的特定诊断消息，还显示有关所使用的卡和内部总线系统的信息。



6、主站信息

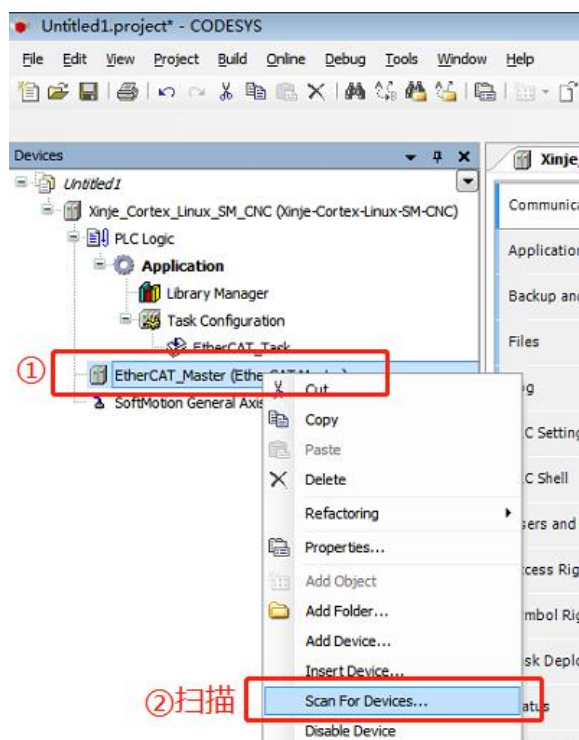
如下图为 EtherCAT_Master 中的“Information（信息）”选项卡。此选项卡显示来自设备描述文件的一般信息：名称、供应商、类别、版本、订单号、描述等。



4-3-3. EtherCAT 从站配置

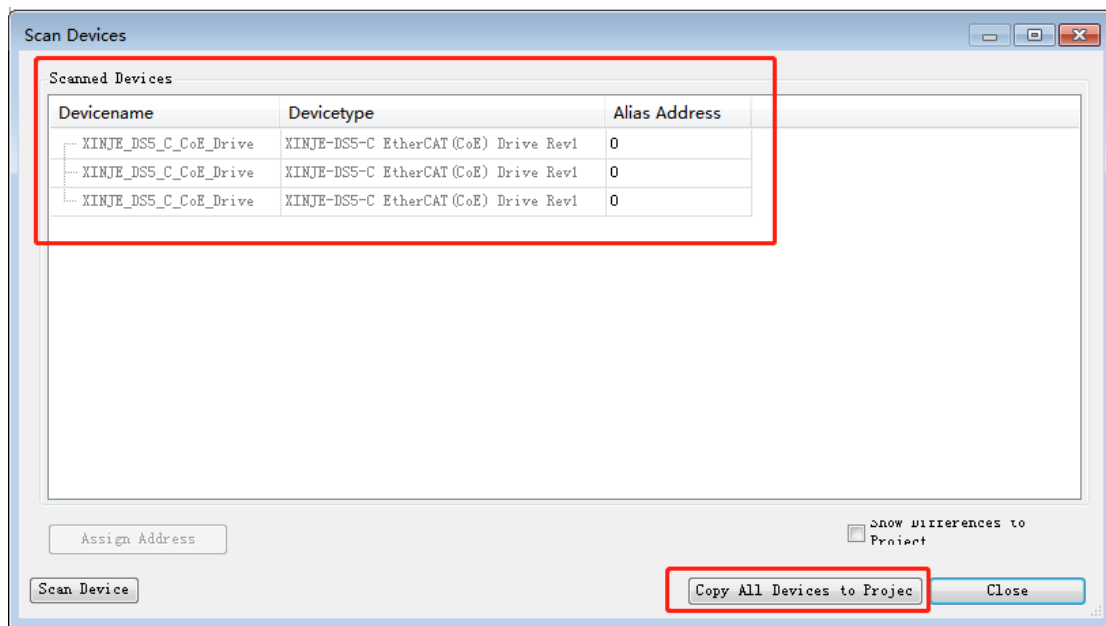
1、添加从站设备

选中主站选项卡右击鼠标，选择“Add Device（添加设备）”，系统会自动弹出从站设备添加框，用户可以根据实际连接的从站进行添加。此外，也可选择“Scan For Devices（扫描设备）”进行实际从站自动扫描搜索。如下图所示：

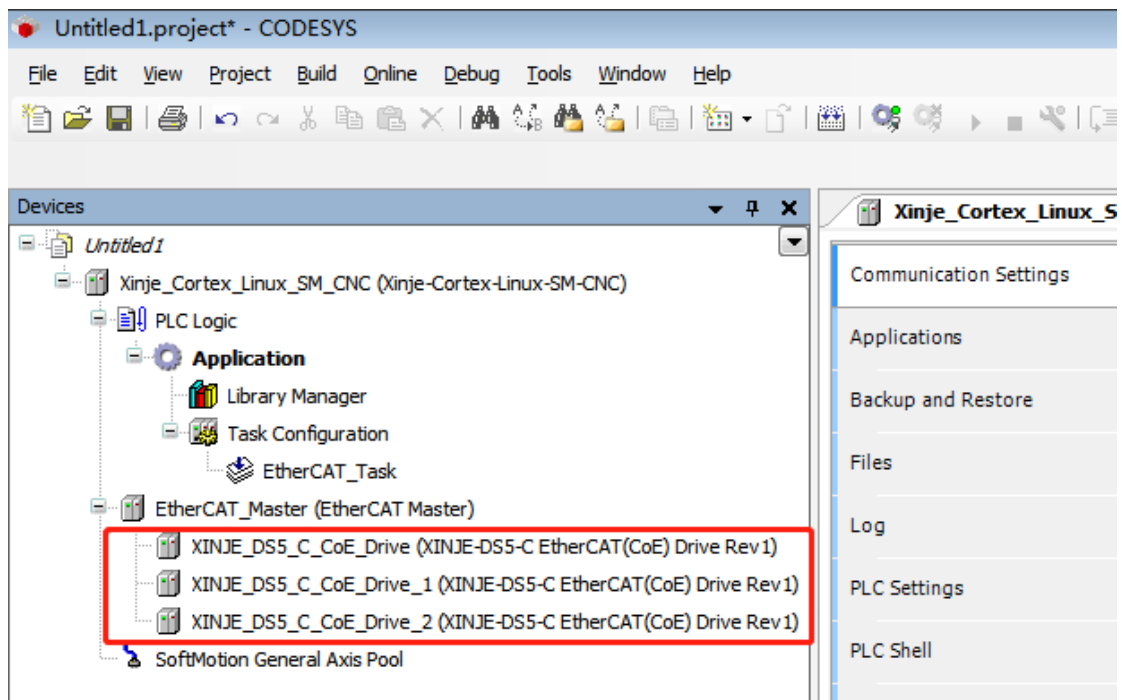


注意：使用“添加设备”或“扫描设备”功能之前，必须确保从站的EtherCAT设备描述文件已经安装在调试PLC的CODESYS内，否则无法使用此功能。

本例中连接了三台DS5C系列的伺服，扫描结果如下图所示，单击Copy All to Project 将扫描到的所有从站添加到项目中去。



从站设备成功添加后“Devices”一栏如下图所示。



2、EtherCAT从站配置

从站添加后，可以通过点击选中从站从而对其进行配置。如下图为从站“General”选项卡中的设置，用于配置从站地址、分布式时钟及从站相应设置。其从站配置信息如下图所示：

【Address（地址）】:

只有在禁用“自动配置主站/从站”模式时，才能编辑字段。

- **AutoInc Address（自动配置地址）**: 由从站在网络中的位置确定。这个地址仅仅在启动时使用，主站需要向从站分配 EtherCAT 地址。当用于此目的的首个报文通过从站时，每一个经过的从站将自身的自动增量地址加 1。
- **EtherCAT Address（EtherCAT 地址）**: 从站的最终地址，由主站在启动时分配。

【Distributed Clock（分布式时钟）】:

- **Select DC（选择 DC）**: 下拉菜单提供了由设备描述文件提供的所有关于分布时钟的设置，可以选择同步或不同步，**一般设置为同步模式**。
- **Enable**: 使能同步时钟。
- **Sync Unit Cycle（同步单元周期）**: 选择同步时钟时，与主站周期时间相同，在此处不可更改。

【Additional（附加）】:

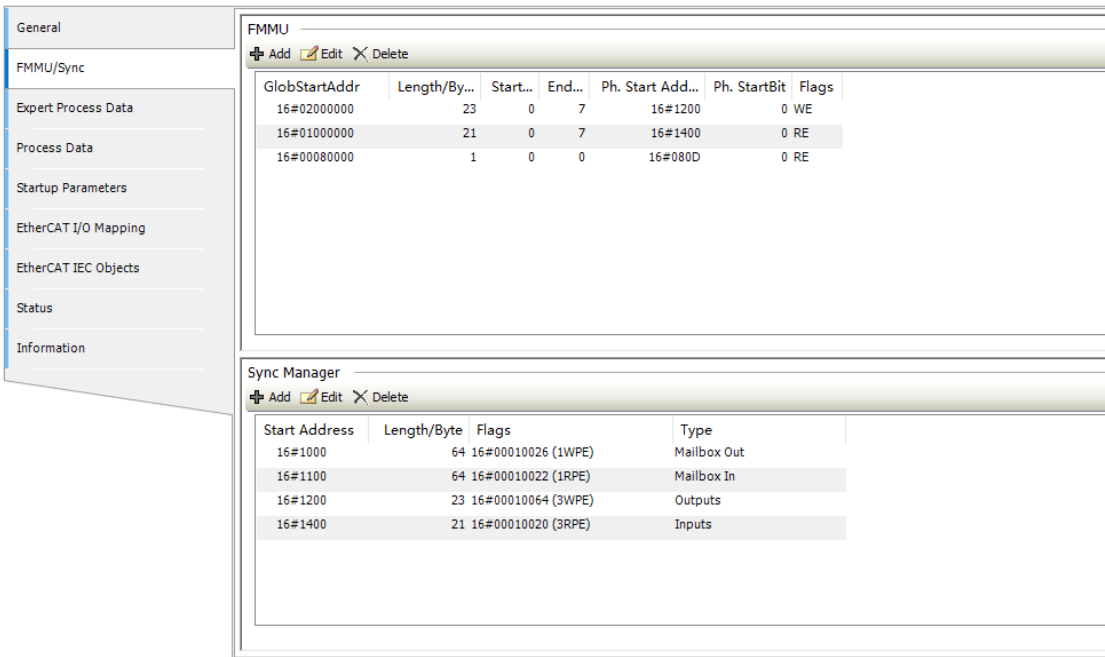
- **Enable Expert Setting（启用专家设置）**: 使能该选项后，能够修改更多 EtherCAT 从站的内部数据。通常情况下需勾选。

【Startup checking（启动检查）】:

- **Check Vendor ID（检查供应商 ID）**: 默认情况下，在系统启动时，将根据当前设置检查设备的供应商 ID 和产品 ID。若没通过，那么总线将被停止，不再执行进一步的操作。
- **Check Product ID（检查产品 ID）**

3、FMMU/Sync

只有主站配置中禁用“自动配置主站/从站”时，该选项卡才可使用。



【FMMU】: Fieldbus Memory Management Units（现场总线内存管理单元）

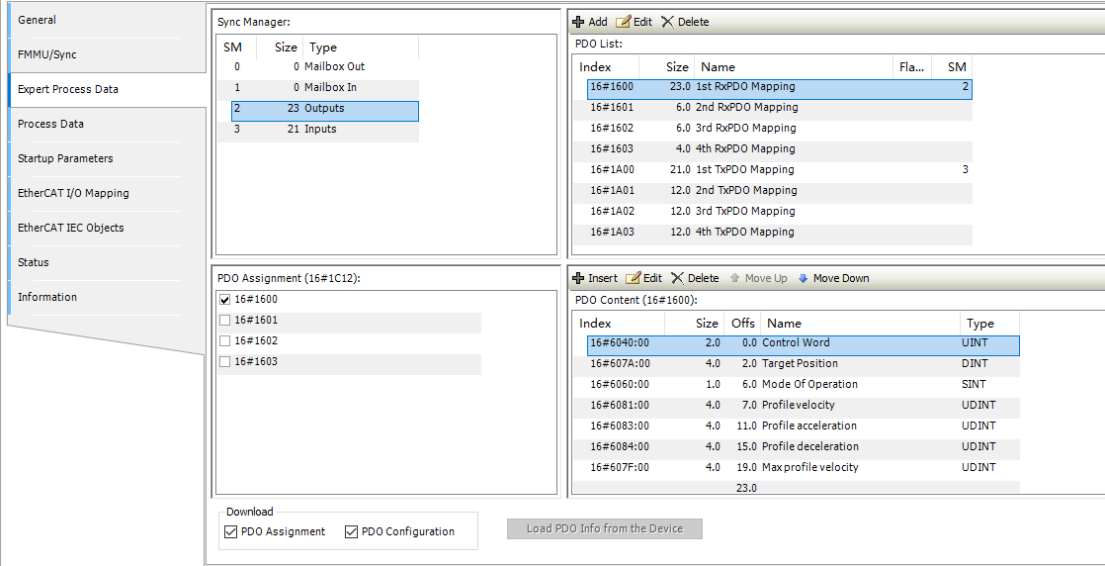
用于处理过程数据。FMMU的功能是将主站分配的逻辑地址转换为本地地址，每一个FMMU通道将一段连续的物理存储地址映射到一段连续的逻辑地址中。这样就实现了主站的逻辑寻址和实际物理地址的映射。

【SM: Synchronization Manager（同步管理器）】:

每种可用的同步管理器类型（邮箱输入、邮箱输出、输入和输出）定义了物理启动地址、访问类型、缓冲区和发送中断的物理地址。

4、专家过程数据

如下图为“Expert Process Data（专家过程数据）”选项卡。勾选从站的“General--Enable Expert Setting（启用专家设置）”复选框时，该功能才可使用。



用户在编写用户工程时，按控制需要，选择和配置TXPDO、RXPDO数据对象表，运行时主站会自动以通信方式转发给对应的从站；尽量只选择需要用到的配置项，减少无关数据对象的配置项，可以减轻EtherCAT通讯的负荷，利于提高通信效率。

5、过程数据

显示了设备描述文件所描述的从站输入和输出过程数据，这些变量永远将来与PLC的实时通信，其设置界面如图所示。

Name	Type	Index
16#1600 1st RxPDO Mapping		
Control Word	UINT	16#6040:00
Target Position	DINT	16#607A:00
Mode Of Operation	SINT	16#6060:00
Profile velocity	UDINT	16#6081:00
Profile acceleration	UDINT	16#6083:00
Profile deceleration	UDINT	16#6084:00
Max profile velocity	UDINT	16#607F:00
16#1601 2nd RxPDO Mapping		
Control Word	UINT	16#6040:00
Target Position	DINT	16#607A:00
16#1602 3rd RxPDO Mapping		
Control Word	UINT	16#6040:00
Target Velocity	DINT	16#60FF:00
16#1603 4th RxPDO Mapping		
Control Word	UINT	16#6040:00
Target Torque	INT	16#6071:00

Name	Type	Index
16#1A00 1st TxPDO Mapping		
Status Word	UINT	16#6041:00
Mode Of Operation Display	SINT	16#6061:00
Position actual internal value	DINT	16#6063:00
Actual Position	DINT	16#6064:00
Velocity actual value	DINT	16#606C:00
Actual Torque	INT	16#6077:00
Following error actual value	DINT	16#60F4:00
16#1A01 2nd TxPDO Mapping		
Status Word	UINT	16#6041:00
Actual Position	DINT	16#6064:00
Velocity actual value	DINT	16#606C:00
Actual Torque	INT	16#6077:00
16#1A02 3rd TxPDO Mapping		
Status Word	UINT	16#6041:00
Actual Position	DINT	16#6064:00
Velocity actual value	DINT	16#606C:00
Actual Torque	INT	16#6077:00
16#1A03 4th TxPDO Mapping		
Status Word	UINT	16#6041:00
Actual Position	DINT	16#6064:00
Velocity actual value	DINT	16#606C:00
Actual Torque	INT	16#6077:00

6、启动参数

该参数根据添加从站设备的不同而不同，在系统启动时可由SDOs（服务数据对象）或IDN将需要写入的参数发给设备。该对话框只有在设备支持“CoE（Canopen over EtherCAT）”或“SoE（Sercos over EtherCAT）”时才显现。包含的必要数据的对象字典由EtherCAT XML描述文件或由EtherCAT XML描述文件引用的EDS文件提供。

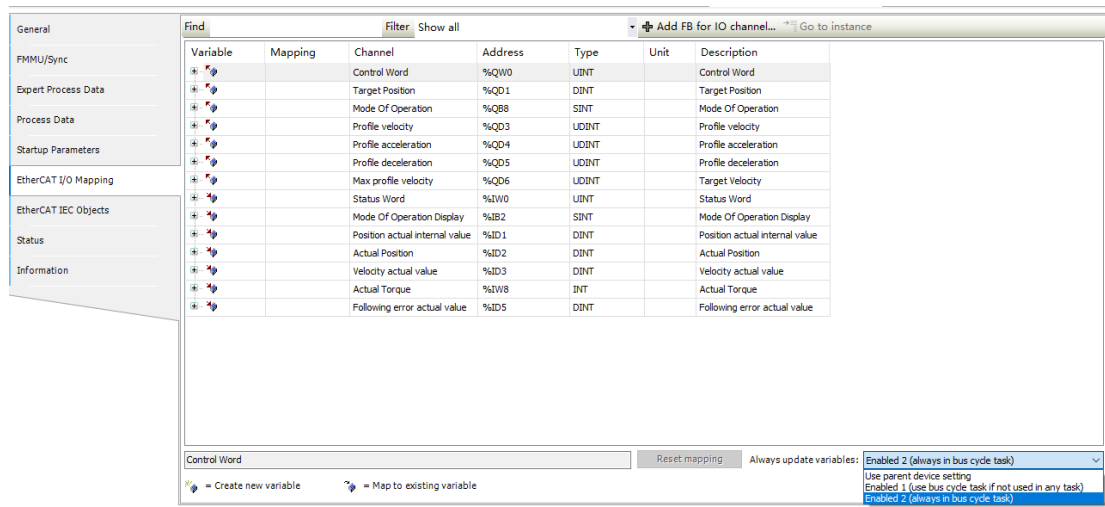
Line	Index/Subindex	Name	Value	Bitlength	Abort if error	Jump to line if error	Next line	Comment
1	16#6060:16#00	Op mode	1	8	☐	☐	0	Op mode
2	16#60C2:16#01	Interpolation time period	100	8	☐	☐	0	Interpolation time period
3	16#60C2:16#02	Interpolation time index	-5	8	☐	☐	0	Interpolation time index

当选择“Add（添加）”或“Edit（编辑）”修改对象时，会字典弹出如图所示的对话框，用户可以在其中选择要写入的参数对象及具体数值。“Move Up（向上移动）”和“Move Down（向下移动）”按钮可以改变当前选中条目在列表中的位置，从而改变参数下载的先后顺序。

Index/Subindex	Name	Flags	Type	Default
* 16#10F1:16#00	Error Settings			
* 16#1600:16#00	1st RxPDO Mapping			
* 16#1601:16#00	2nd RxPDO Mapping			
* 16#1602:16#00	3rd RxPDO Mapping			
* 16#1603:16#00	4th RxPDO Mapping			
* 16#1A00:16#00	1st TxPDO Mapping			
* 16#1A01:16#00	2nd TxPDO Mapping			
* 16#1A02:16#00	3rd TxPDO Mapping			
* 16#1A03:16#00	4th TxPDO Mapping			
* 16#1C12:16#00	RxPDO assign			
* 16#1C13:16#00	TxPDO assign			
* 16#1C32:16#00	SM output parameter			
* 16#1C33:16#00	SM input parameter			
16#2000:16#00	PO-00	RW	UINT	

7、EtherCAT I/O映射

在“EtherCAT I/O Mapping (EtherCAT I/O映射)”选项卡中，可以将控制设备的输入、输出和内存地址映射到应用程序的变量。这样就创建了所谓的“I/O映射”。



4-4. 伺服轴电机参数的配置

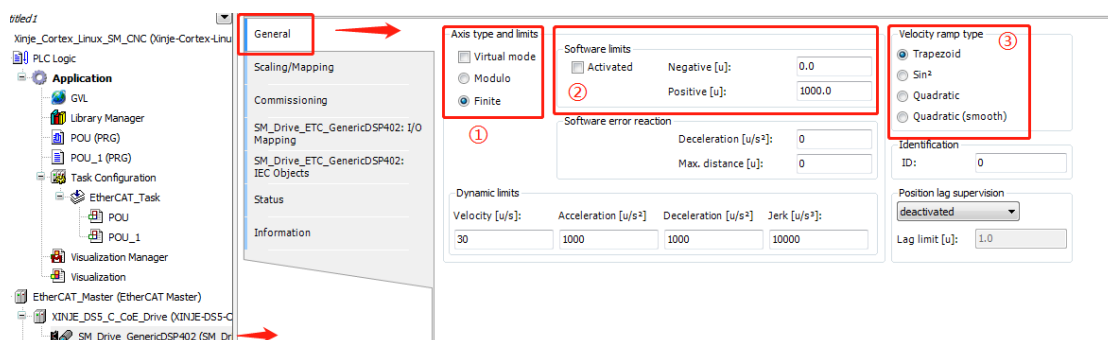
运动控制的动作，最终是通过伺服电机的运转来实现的，若要伺服电机按照我们所希望的，控制器需要知道伺服电机参数、应用系统机械传动机构特性参数，以及用户希望的运转特性，才能发送合适的运行位置指令，这需编程时对控制器设置这些特性参数。

设置伺服电机参数的方法如下图说明，双击伺服驱动器下伺服电机，在右侧的窗口即可对电机相关的参数进行设置了。

4-4-1. 基本参数配置

在General（基本参数）标签下，主要设置轴类型。如下图所示：

伺服电机是往返运行的特点若，如丝杆的往复运转，可以选择“线性模式”，（也称多圈模式、有限长模式），方便在伺服电机旋转多圈数的情况下，能进行绝对位置模式的定位；若伺服电机是单方向无限运转，例如飞剪轧辊的运转，就可以“周期模式”，其位置计数器在每个运转周期内，从0开始计数，不会产生位置计数器溢出的情况。



【Axis type（轴类型）】:

- Virtual mode（虚轴类型）：可以在没有实际接入伺服的情况下运行，可用于仿真运行。
- Modulo（周期模式）：沿一个方向无限运转的机构，其位置计数器只需记住本周期的计数值。
- Finite（线性模式）：运动机构在有限范围内往返运动，如驱动丝杆机构的伺服电机，其位置计数器可以记住丝杆全运行范围内的位置。

【Software limits（软件限制）】:

运行行程超限保护。是通过XS3的软件,对伺服电机的行程超限进行保护,让XS3不会发出超限的定位指令,这在绝对位置编码器、绝对定位指令的MC指令的应用系统中非常有用。

【Software error reaction（软件错误反应）】:

可设置Deceleration（减速）值和Max distanc（最大距离）。

【Velocity ramp type（速率斜坡模式）】:

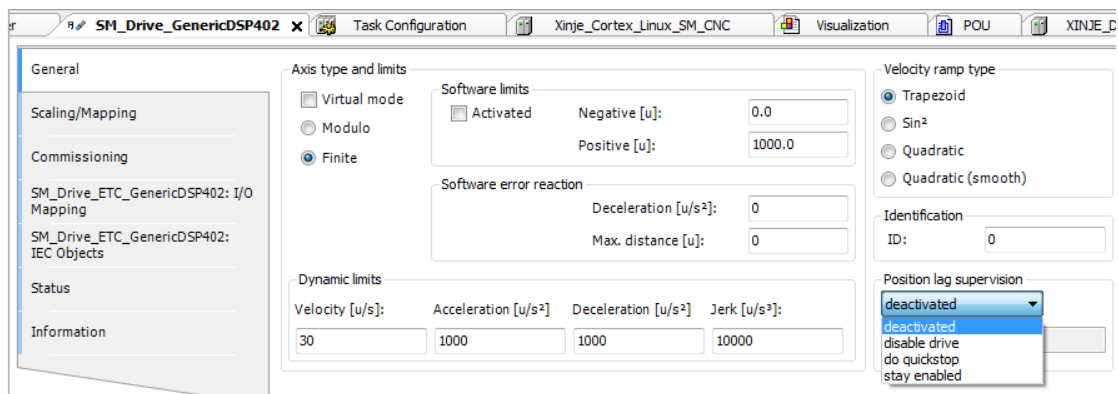
对加减速特性曲线的选择。可选Trapeziod（梯形）、Sin2、Quadratic（二次方）、Quadratic(smooth)（二次（平滑））四种模式。在调试中可酌情选择,以使机械系统运转更为平稳。

【Dynamic limits（动态限制）】:

可设置Velocity（速度）、Acceleration（加速度）、Decleration（减速）、Jerk（跃度）。

【Position lag supervision（位置滞后检测）】:

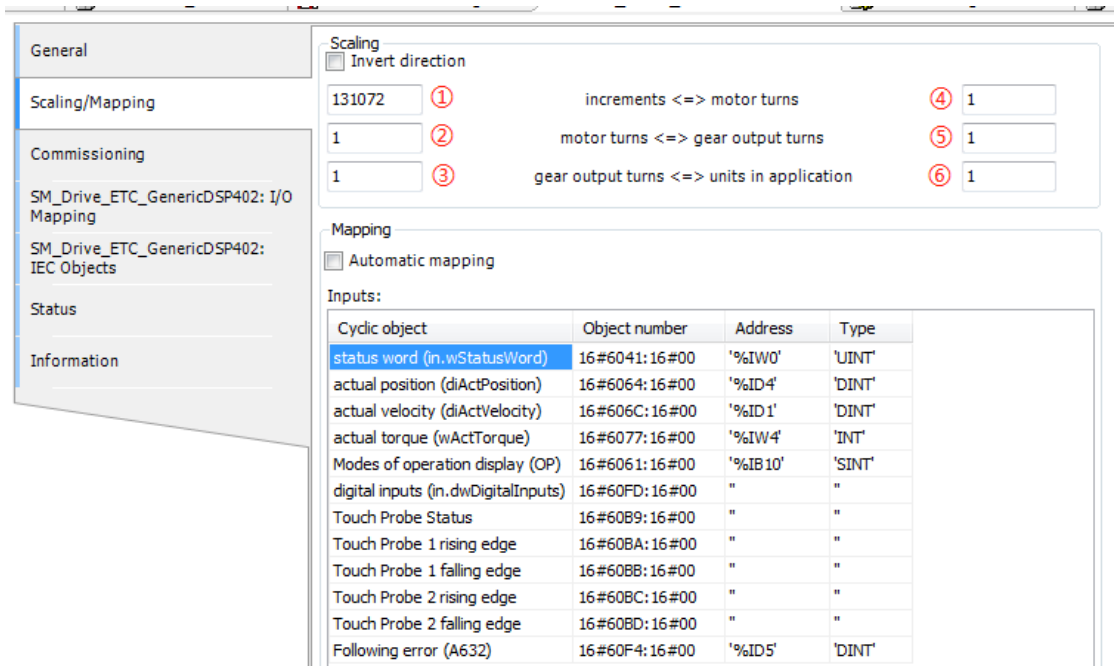
可选deactivated（使失效）、disable drive（禁用驱动）、do quickstop（快速停止）、stay enabled（保持使能）四种设置。



注意: 上述设置规则,既适用于增量编码器伺服电机,也适合绝对编码器的伺服电机,上述设定值并没有送给伺服驱动器,而电机的当前位置,是由AM600根据电机反馈的位置信号,以及上述设定参数,自动进行位置的累计和量程计算,因此,若需对伺服位置有掉电保持特性,需要用户程序中对轴的当前位置备份到掉电保持的变量中,上电后再及时恢复给相关参数。

4-4-2. Scaling/Mapping（“缩放/映射”）配置

因控制器总是以令伺服运转多少脉冲数来作为运行位置命令的,因此,控制器必需知道伺服电机的编码器每圈脉冲值,同时还需知道运转机构的减速比、丝杆导程、皮带轮周长等机械参数,在电机参数Scaling/Mapping（“缩放/映射”）标签下可输入这些参数,如下图:



【Scaling】:

- Invert direction（反转反向）：该选项可以修改电机的旋转方向；
- increments<=>motor turns（增量<=>电机转）：电机编码器每圈脉冲数，可以十六进制输入。17bit编码器填131072或16#20000；23bit编码器填8388608或16#800000。实际运行时，控制器以EtherCAT通信方式向伺服驱动器发送所需要运行的脉冲数，来控制伺服运行，因此编码器分辨率，需要根据实际情况准确设定。
- motor turns<=>gear output turns（电机转动<=>齿轮输出转）：减速机构的减速比，若非整数，可以用两侧整数的比值来表示。例如若输入5:1表示伺服电机轴每转5圈，减速机输出轴旋转1圈；若没有使用减速机，则减速比为1:1。
- gear output turns<=>units in application（减速机输出转<=>应用的单元）：用于设定输出轴每旋转1圈，工件运行的物理距离。若非整数，可以用两侧整数的比值来表示。

通过以上参数的准确设置，就可以按照应用系统的运动距离物理单位输入distance命令了，可以实现应用系统物理单位与MC运行指令单位一致，使得用户程序的指令直观明了，方便变量设置，不易发生错误。

另外还需注意，图中的参数输入栏中只能输入整形数，因左右两边对应行中的参数之比为有效比例值，可以通过在左右两边对应行输入合适的整数值。例如何伺服电机经过变比为4:1机械减速机构后，驱动导程为5.6mm的丝杆（即丝杆转动1 圈，丝杆滑块运动5.6mm）运动，设定如下图：

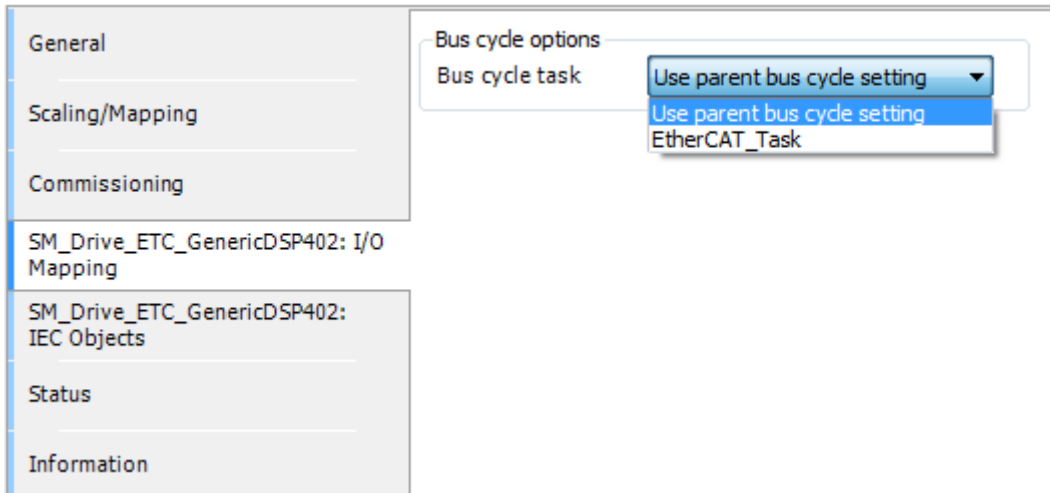
16#20000	increments <=> motor turns	1
4	motor turns <=> gear output turns	1
10	gear output turns <=> units in application	56

注意：设置电机参数，是用于XS3在发送最终的（脉冲数）位置指令时，进行电子齿轮比的换算，并没有下载到伺服驱动器中；而伺服中功能码设置的电子齿轮比同样会对运行指令进行衰减处理。

【Mapping】:

- Automatic mapping：自动映射。禁用该选项后，可以手动编辑映射。

4-4-3. I/O Mapping (I/O 映射)



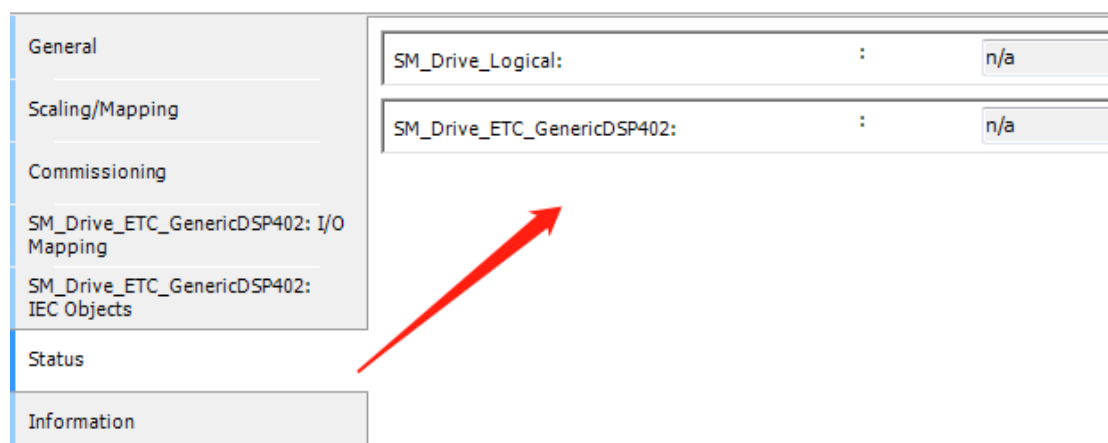
4-4-4. IEC Objects

此对话框仅在隐式创建可在应用程序中访问的设备对象的实例时出现。例如，这可以用于重新启动总线或查询信息。在线可通过此页面修改参数。

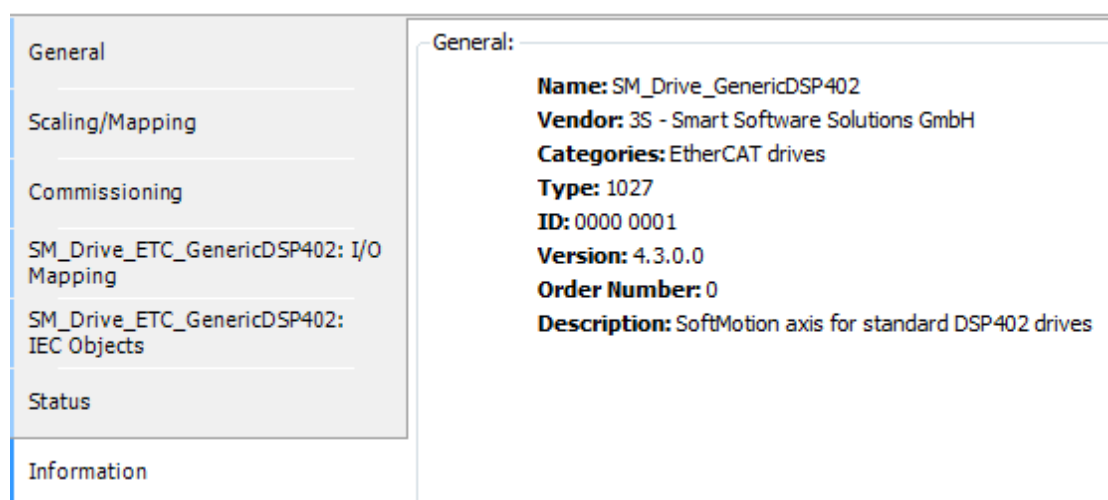
Expression	Type	Value	Prepared value	Address	Comment
Xinje_Cortex_Linux_SM_CNC.Application.SM_Drive_GenericD...	SM3_Drive_DS...				
wAxisStructID	WORD	65042			
nAxisState	SMC_AXIS_STATE	standstill			State of the axis according to the "PLCopen...
bRegulatorOn	BOOL	TRUE			Parameter number: 1010
bDriveStart	BOOL	TRUE			Parameter number: 1011
bCommunication	BOOL	TRUE			"TRUE": Communication OK
wCommunicationState	WORD	100			Parameter number: 1013
uDriveInterfaceError	UINT	0			Drive interface error number
bRegulatorRealState	BOOL	TRUE			Parameter number: 1015
bDriveStartRealState	BOOL	TRUE			Parameter number: 1016
wDriveId	WORD	0			Parameter number: 1021
iOwner	INT	0			Parameter number: 1022
iNoOwner	INT	0			Parameter number: 1023
fCycleTimeSpent	LREAL	0			Parameter number: 1024
fTaskCycle	LREAL	0.004			Parameter number: 1025
bError	BOOL	FALSE			Parameter number: 1030
dwErrorID	DWORD	0			Parameter number: 1031
bErrorAckn	BOOL	FALSE			Parameter number: 1032
bDisableErrorLogging	BOOL	FALSE			Parameter number: 1036
bFeedbackError	ARRAY [0..g_SMC_...				
dwRatioTechUnitsDenom	DWORD	65536			Parameter number 1051
iRatioTechUnitsNum	DINT	1			Parameter number 1052
nDirection	MC_DIRECTION	positive			Parameter number 1053
fScalefactor	LREAL	65536			Parameter number 1054

4-4-5. 电机轴状态

通用设备编辑器的这个选项卡显示状态信息，例如“正在运行”或“停止”，以及来自各自设备的特定诊断消息，还显示有关所使用的卡和内部总线系统的信息。



4-4-6. 电机轴信息



5. 以太网通讯原理及配置

本章主要介绍以太网通讯的原理及配置。

5. 以太网通讯原理及配置	39
5-1. 简介	40
5-2. ETHERNET 参数介绍	40
5-3. 组网方式	40
5-3-1. 直连	41
5-3-2. 中转连接方式	41
5-4. 以太网通讯原理分析.....	41
5-4-1. TCP/IP 协议.....	41
5-4-2. MODBUS TCP 协议概述	41
5-4-3. MODBUS 通讯地址.....	42
5-4-4. MODBUS 通讯功能码.....	42
5-5. 以太网通讯配置.....	42
5-6. 以太网通讯案例.....	45

5-1. 简介

Ethernet 通讯主要应用于 XS3 系列第一个 RJ45 物理接口, 具有比串口通讯方式更迅捷稳定的特点。CodeSys 编程工具软件对 PLC 进行数据监控时, 通过 Ethernet 通讯可以获得数据实时性更好、程序上下载更快速的用户体验。而 Ethernet 通讯本身运用的接线方式较 RS485、USB 也有着明显的优势, 可靠易用的接线方式在多台 PLC 通讯的场合中尤为适用, 用户只需通过一台交换机便可实现对现场任意 PLC 进行通讯, 同时高速、稳定的以太网组网也为大容量数据的交互提供有力的保障。

5-2. Ethernet 参数介绍

1、IP 地址相关参数

XS3 系列 Ethernet 基于标准的 TCP/IP 协议开发, 在通讯中需设定 IP 地址作为每台设备的编号。IP 地址的设定共有四项参数, 如下图所示, 就是 XS3 配置工具设置 IP 地址的界面。



2、IP 地址

XS3 系列 PLC 的 IP 地址默认设置为 C 类地址 192.168.6.6, C 类地址的用户设定格式为 192.168.A.B (A 的取值范围 0~255, B 的取值范围 1~254)。

3、子网掩码

XS3 系列 PLC 的默认子网掩码为 255.255.255.0, 当 PLC 所在局域网 IP 设备数量大于 254 个时, 可以适当改小。

4、默认网关

局域网连接中, 只和本地网络内的设备通信时, 默认网关可以不用设置。

5-3. 组网方式

XS3 系列 PLC 在使用 Ethernet 通讯时物理接口为 RJ45, 连接方式有直连方式和中转连接方式。接线时选择超五类 UTP 和 STP 网线, 单段网线长度建议不超过 100M。交换机类型建议为百兆/千兆自适应交换机, 例如 TP-LINK 交换机。接线完成后可以通过观察 PLC 端 RJ45 口的指示灯状态判定物理连接是否正常, 如果橙色指示灯常亮、绿色指示灯闪烁, 则网口连接生效。

5-3-1. 直连

直连方式是将两台设备的 RJ-45 网口用一根网线直连，此接法仅适用于两台设备一对一通讯如 PLC 与上位机通讯或 2 台 PLC 通讯的情况，在这种接线方式下，没有提供 DHCP 服务的设备，需要将设备 IP 设置为固定 IP 通信。

5-3-2. 中转连接方式

中转连接方式适用于存在多设备组网通讯时，所有通讯设备通过交换机中转连接，连接方式为每台 PLC（包括涉及到的其他设备，如上位机，触摸屏，路由器）通过网线将 PLC 的 RJ-45 接口与交换机上的 RJ-45 接口连接。

5-4. 以太网通讯原理分析

5-4-1. TCP/IP 协议

TCP/IP 协议是现在比较通用的以太网通信协议，与开放互联模型 ISO 相比，采用了更加开放的方式，它已经被美国国防部认可，并被广泛应用于实际工程。TCP/IP 协议可以用在各种各样的信道和底层协议（如 T1、X.25 以及 RS232 串行接口）之上。确切地说，TCP/IP 协议是包括 TCP 协议、IP 协议、UDP 协议、ICMP 协议和其他一些协议的协议组。

5-4-2. MODBUS TCP 协议概述

MODBUS TCP 结合了以太网物理网络和网络标准 TCP/IP 以及以 MODBUS 作为应用协议标准的数据表示方法。MODBUS TCP 通信报文被封装于以太网 TCP/IP 数据包中，MODBUS 协议规范一帧数据的最大长度为 256 个字节。MODBUS TCP/IP 的通信系统中有两种类型的设备：MODBUS TCP/IP 客户端和服务器设备。

1、MODBUS 客户端

客户端（TCP Client）主动向服务器（TCP Server）发起连接请求，连接建立成功，仅允许客户端主动发起通讯请求。

以太网机型作为 MODBUS TCP 客户端时，通过 S_OPEN 指令建立 TCP 连接，通过 M_TCP 指令发起 MODBUS 请求。

2、MODBUS 服务器

服务器主动监听 502 端口，等待客户端连接请求，连接建立成功，响应符合 Modbus TCP 协议规范的数据通讯请求。

以太网机型上电默认开启此服务，最大响应不超过 4 个 TCP 连接。

3、端口号

在以太网中，基于 TCP 协议的通信必须使用端口号才能与上层应用进行通信，端口号的范围从 0 到 65535，有一些端口号对应有默认功能，比如用于浏览网页服务的 80 端口，用于 FTP 服务的 21 端口等等，在 XS3 系列 PLC 中仅支持 MODBUS TCP 通信，用于 MODBUS TCP 通信的端口为 502 端口。

4、套接字（Socket）的基本概念

套接字（Socket）是通信的基石，是支持 TCP/IP 协议的网络通信的基本操作单元。它是网络通信过程中端点的抽象表示，包含进行网络通信必须的五种信息：连接使用的协议、本地主机的 IP 地址、本地进程的协议端口、远端主机的 IP 地址、远端进程的协议口。应用层通过传输层进行数据通信时，TCP 会遇到同时为多个应用程序进程提供并发服务的问题。多个 TCP 连接或多个应用程序进程可能需要通过同一个 TCP 协议端口传输数据。为了区别不同的应用程序进程和连接，许多计算机操作系统为应用程序与 TCP/IP 协议交互提供了套接字接口。应用层可以和传输层通过套接字接口，区分来自不同应用程序进程或网络连接的通信，实现数据传输的并发服务。

XS3 系列 PLC 中作为客户端时支持的套接字为 0-63，最多可以同时建立 32 个 Socket 连接。

5、建立套接字（Socket）连接

建立套接字连接至少需要一对套接字，其中一个运行于客户端（也称之为 TCP 客户端），称为 ClientSocket，另一个运行于服务端（也称之为 TCP 服务器），称为 ServerSocket。

套接字之间的连接过程分为三个步骤：服务端监听、客户端请求、连接确认。

① 服务端监听：

服务端套接字并不定位具体的客户端套接字，而是处于等待连接的状态，实时监控网络状态，等待客户端的连接请求。

② 客户端请求：

指客户端的套接字提出连接请求，要连接的目标是服务端的套接字。为此，客户端的套接字必须首先描述它要连接的服务端的套接字，指出服务端套接字的地址和端口号，然后就向服务端套接字提出连接请求。

③ 连接确认：

当服务端套接字监听到或者说接收到客户端套接字的连接请求时，就响应客户端套接字的请求，建立一个新的线程，把服务端套接字的描述发给客户端，一旦客户端确认了此描述，双方就正式建立连接。而服务端套接字继续处于监听状态，继续接收其他客户端套接字的连接请求。创建套接字连接时，可以指定使用的传输层协议，套接字可以支持不同的传输层协议，当使用 TCP 协议进行连接时，该套接字连接就是一个 TCP 连接。

5-4-3. MODBUS 通讯地址

XS3 系列 PLC 作为 Modbus 服务器时，内部软元件编号对应 Modbus 地址编号如下表所示：

类型	名称	属性	个数	Modbus 地址
word、字对象	In 0-499	RW	500	0-499
word、字对象	Out0-499	RO	500	500-999
Bool、位对象	In 500-599	RW	100	0-99
bool、位对象	Out500-599	RO	100	500-599

5-4-4. MODBUS 通讯功能码

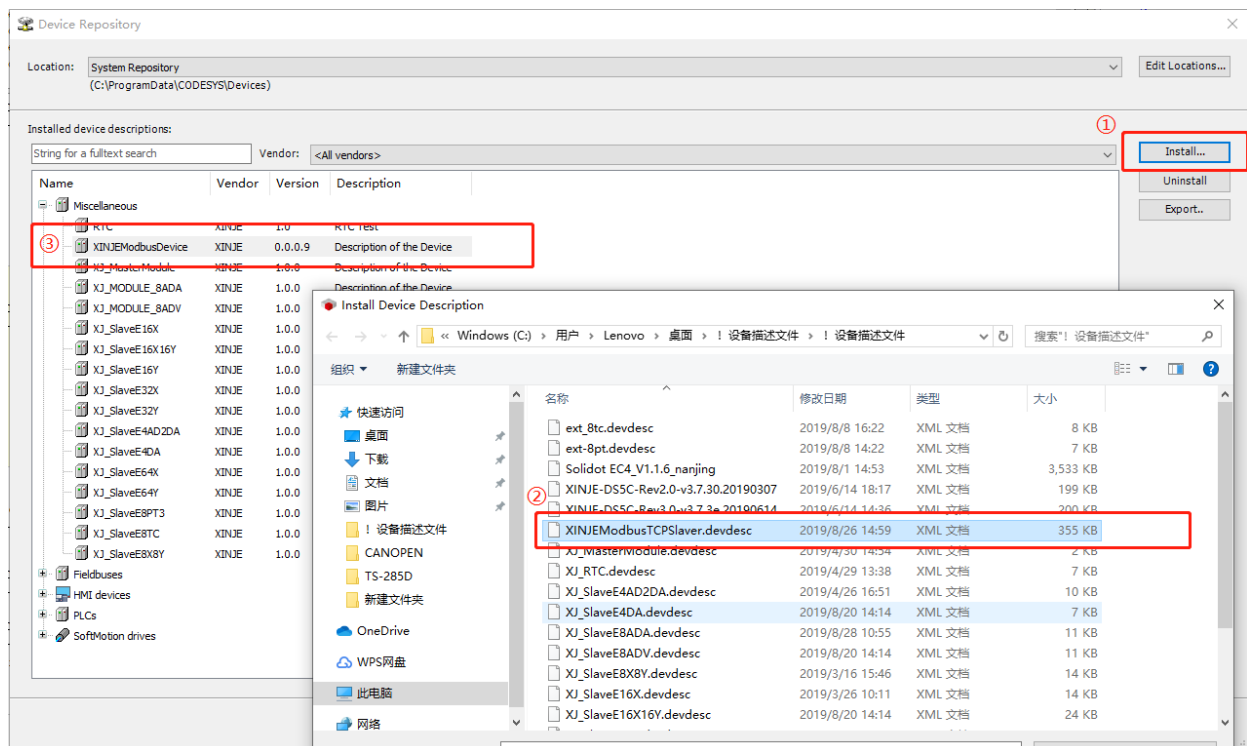
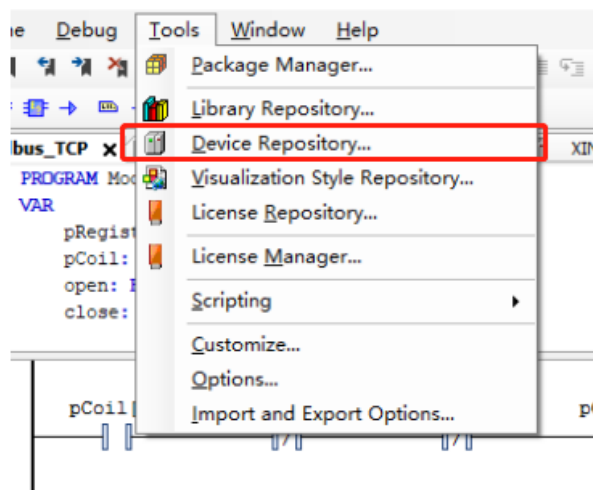
功能码	功能	说明
K1	读线圈	读取 0X 类型地址，最大数量 2000 个
K2	读输入离散量	读取 1X 类型地址，最大数量 2000 个
K3	读寄存器	读取 4X 类型地址，最大数量 120 个
K4	读输入寄存器	读取 3X 类型地址，最大数量 120 个
K5	写单个线圈	写单个 0X 类型地址
K6	写单个寄存器	写单个 4X 类型地址
K15	写多个线圈	写 0X 类型地址，最大数量 2000 个
K16	写多个寄存器	写 4X 类型地址，最大数量 120 个

5-5. 以太网通讯配置

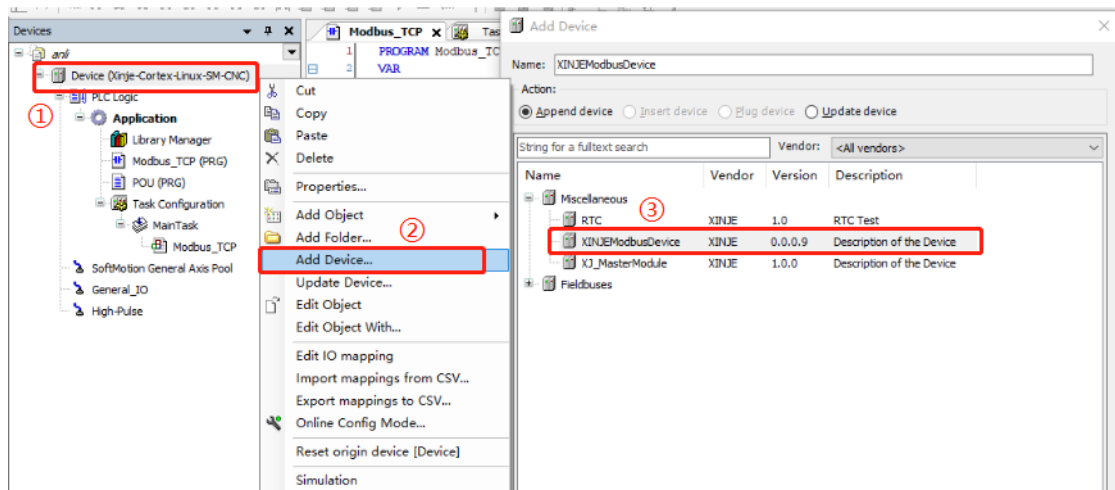
1、添加 Modbus 设备描述文件

为了使用以太网通讯配置功能，必须添加硬件提供的设备描述文件，通过“Device Repository（设备库）”对话框安装。该设备描述文件（*.devdesc.xml）定义了 XS3 系列 PLC 作为客户端和服务端时的通讯映射地址，以及通讯串口参数。

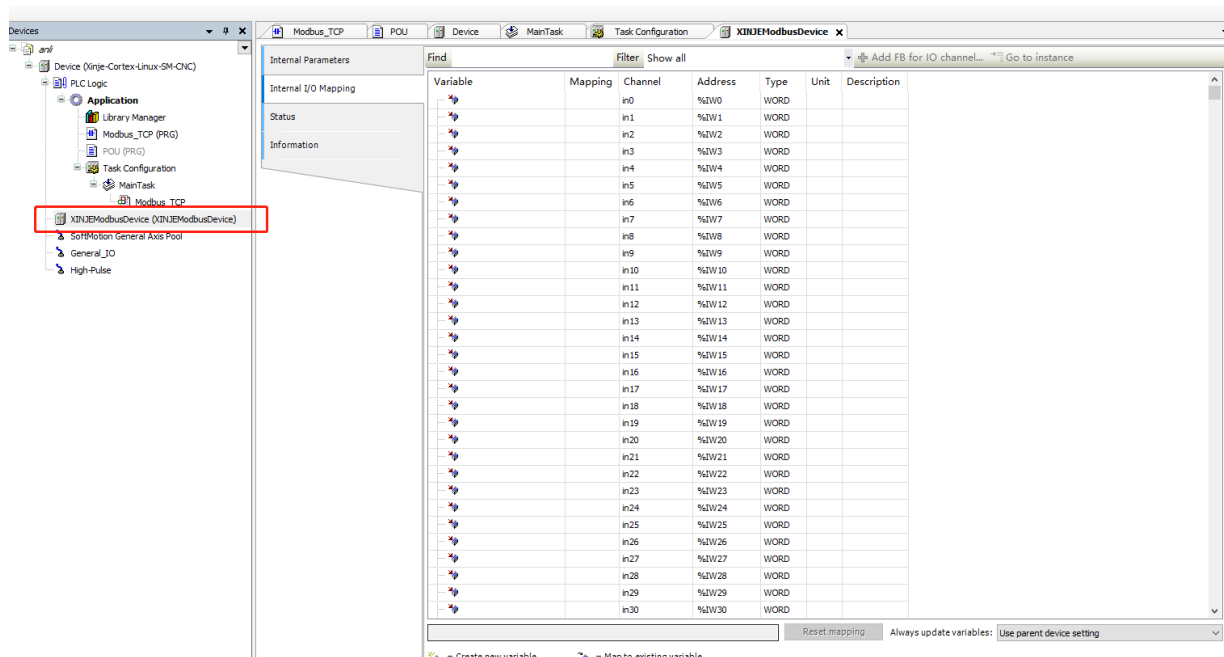
在应用的工程中打开工具设备库，依次点击“Tools→Device Repository”，在弹出的对话框中点击“安装 Install”，选择 XML 文件所在的路径找到 XML 文件，选中后点击打开。如下图所示：



添加成功后可以在左边 Miscellaneous 目录下看到名为“XINJEModbusDevice”的设备描述文件。在左侧选中 Xinje-Cortex-Linux-SM-CNC 右击，点击 Add Device 添加设备。找到刚才添加的“XINJEModbusDevice”设备描述文件后点击“Add Device”完成添加。



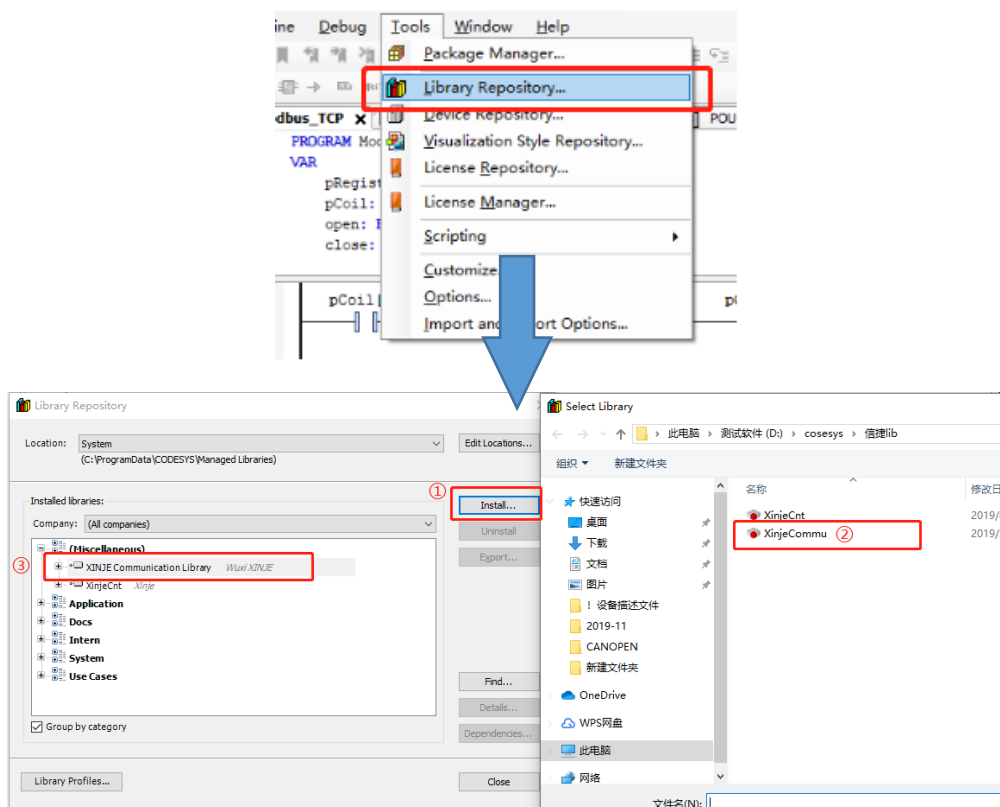
完成添加后可以在左侧看到该设备，如下图所示。点击后可以对该设备的串口参数以及通讯映射地址进行配置。



在 XS3 系列 PLC 进行以太网通讯时，无论作为客户端还是服务器都需要添加该描述文件。以太网通讯的 IP 地址通过 Modbus 小工具设置，具体详见 8.1，无需在“Internal Parameters”选项卡中配置串口参数，在“Internal I/O Mapping”选项卡中可以将设备的输入、输出地址映射到应用程序的变量，这样就创建了所谓的“I/O 映射”。

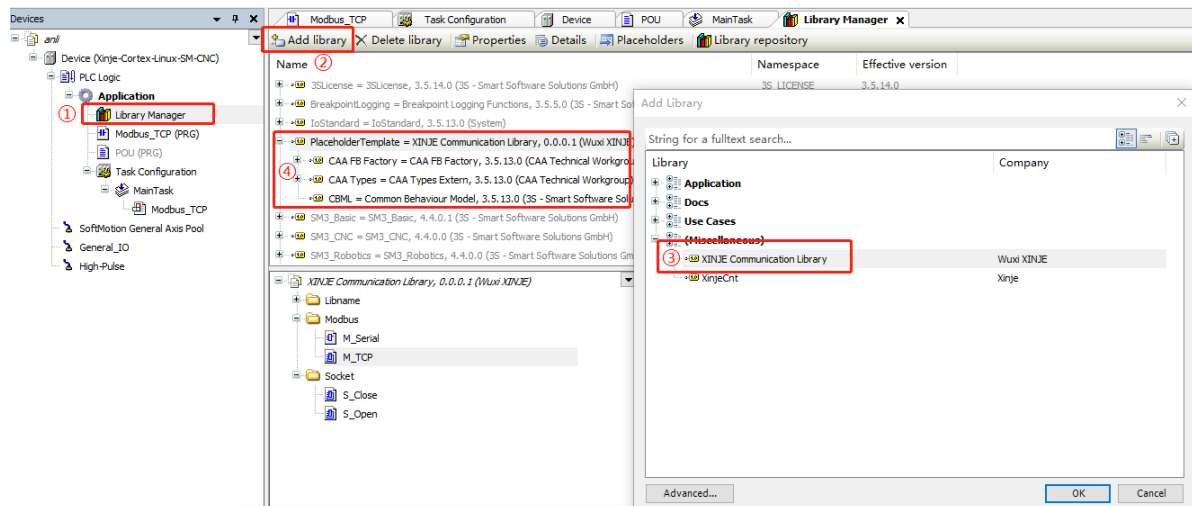
2、添加以太网通讯指令（CODESYS Library）

要使用以太网 ModbusTCP 通讯，还需要添加 ModbusTCP 通讯指令，方法与添加 Modbus 设备描述文件类似。在应用的工程中打开工具设备库，依次点击“Tools→Library Repository”，在弹出的对话框中点击“安装 Install”，选择 CODESYS Library 文件所在的路径找到 CODESYS Library 文件，选中后点击打开。如下图所示：



添加完该 Library 后还需将指令的 Library 添加至工程中。点击左侧 Application ---- Library Manager ---- Add

library 将刚添加的“XinjeCommu” 添加到工程的 Liabary 中，具体操作如下图。



添加完成后即可创建 POU，在 POU 中直接调用以太网通讯指令进行编写程序。

5-6. 以太网通讯案例

例一：XS3 作为客户端，实现上电自动创建 ModbusTCP 连接，并与信捷 XD 系列以太网 PLC 进行线圈、寄存器对象的读写，并能在连接出现异常时自动关闭连接并重新建立。XS3 的 IP 地址为 192.168.6.6，目标通信设备 PLC 的 IP 地址为 192.168.6.7。

程序操作：

- (1) 按照 5-5 中的步骤添加设备描述文件以及以太网通讯指令。
 - (2) 创建 Task 以及 POU 开始编写程序，这里我们以创建梯型图程序为例。
- ① 定义需要用到的变量和功能块，本案例中需要用到的变量定义如下图。

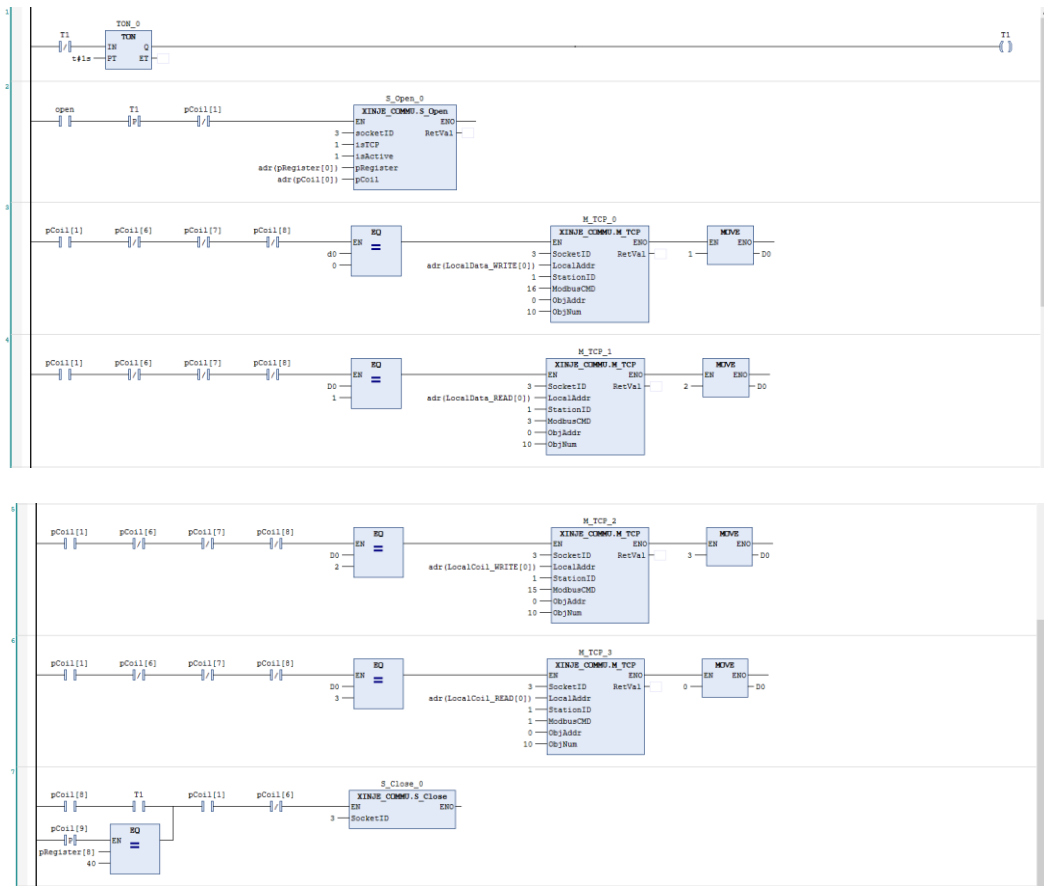
```

1  PROGRAM Modbus_TCP
2  VAR
3      pRegister: ARRAY [0..8] OF WORD;
4      pCoil: ARRAY [0..9] OF BOOL;
5      open: BOOL;
6      close: BOOL;
7      LocalData_WRITE: ARRAY [0..9] OF WORD;
8      LocalData_READ: ARRAY [0..9] OF WORD;
9      LocalCoil_WRITE: ARRAY [0..9] OF BOOL;
10     LocalCoil_READ: ARRAY [0..9] OF BOOL;
11     S_Open_0: XINJE_COMMU.S_Open;
12     M_TCP_0,M_TCP_1,M_TCP_2,M_TCP_3: XINJE_COMMU.M_TCP;
13     S_Close_0:XINJE_COMMU.S_Close;
14     D0: INT;
15     T1: BOOL;
16     TON_0: TON;
17 END_VAR
18

```

② 具体程序如下图所示，这里我们使用 1S 的定时信号自动建立连接，建立成功后不再继续建。成功后进行与 PLC 的 ModbusTCP 通讯，一共有 4 条指令，分别为：字读写和位读写。当连接出现异常或接收超时自动关闭连接。

这里在 4 条 M_TCP 指令之间使用 D0 复制和 pCoil[7]（通讯中标志位）来保证 4 条 M_TCP 指令轮流执行。



(3) 参数块中设置通讯的必要参数，这里我们设置目标 IP 为 192.168.6.7，目标端口为 502，超时时间为 1s。

pRegister[1]中写入 49320，十六进制为 C0A8，即高八位、低八位分别为十六进制的 IP 地址前两位 192.168。pRegister[2]中写入 1543，十六进制为 0607，即 IP 地址的后两位 6.7。pRegister[5]中写入 100，即为 1s。

Device.Application.Modbus_TCP			
Expression	Type	Value	Prepared value
pRegister	ARRAY [0..8] OF W...		
pRegister[0]	WORD	0	
pRegister[1]	WORD	49320	
pRegister[2]	WORD	1543	
pRegister[3]	WORD	502	
pRegister[4]	WORD	0	
pRegister[5]	WORD	100	
pRegister[6]	WORD	0	
pRegister[7]	WORD	0	
pRegister[8]	WORD	0	

写入成功后置 ON 程序中的“open”建立连接，是否建立成功可以在 pCoil 标志位中查看，若建立成功，pCoil[1] 将会置 ON，若连接失败，则 pCoil[9]将会置 ON，此时可以查看 pRegister[8]中的错误代码。

6. 用户程序的组成及执行机制

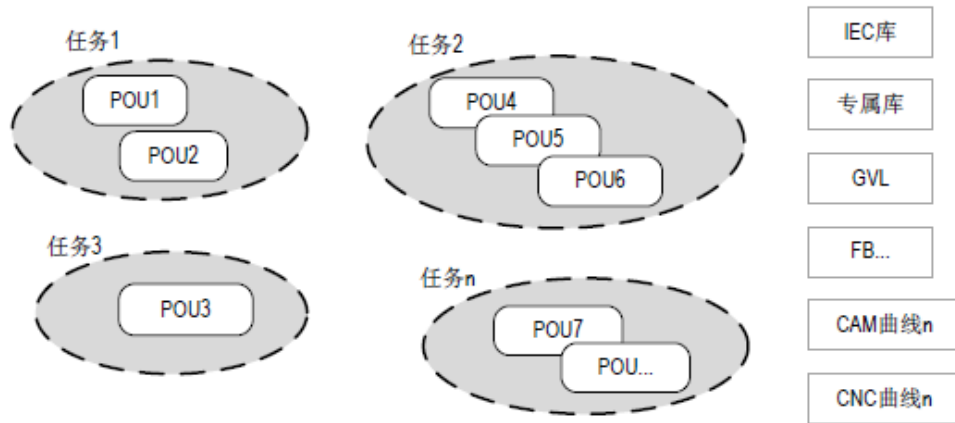
本章主要介绍对象字典区域分配、COE 通信区域、驱动 Profile 区域等内容。

6. 用户程序的组成及执行机制	47
6-1. 用户程序基本结构	48
6-2. 用户程序中的任务与配置	48
6-2-1. 任务类型	48
6-2-2. 任务配置	49
6-3. 用户程序由多个 POU 组成的优势	52

6-1. 用户程序基本结构

XS3是基于多任务操作系统开发的控制器，系统运行的多个功能模块以多任务执行的方式，即可以“同时”执行几个任务，每个任务可以有若干个用户程序组织单元（简称POU），典型的构成举例如下图：

用户工程



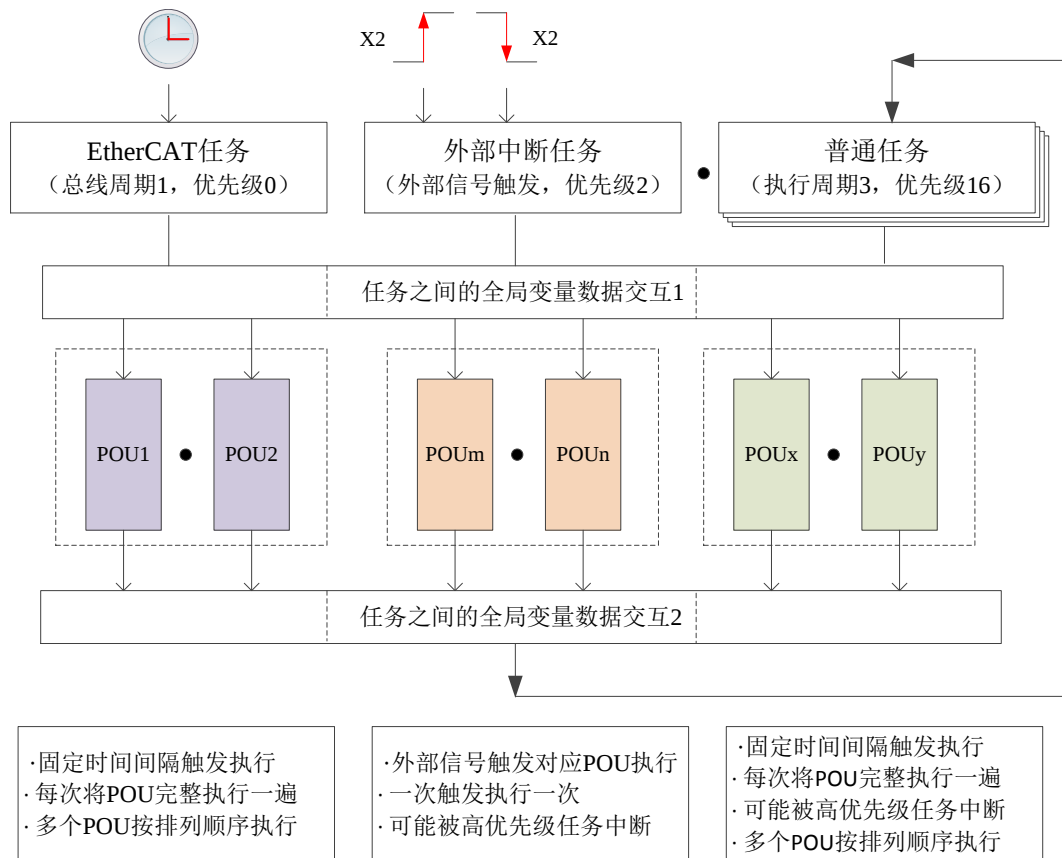
编写XS3的用户程序时，用户可根据应用系统中不同处理的业务类型和紧急程度，分成若干个程序组织单元组成，可将每个任务指定不同的执行触发条件或相应的执行时间间隔（也称执行周期），这样可以让应用系统的控制响应达到最佳状态。没有列入任务配置的POU将不会被执行的。

另外用户工程中，还有一些支撑用户程序的对象，如库函数、全局变量GVL、功能块FB、凸轮定义CAM曲线、多轴插补轨迹定义CNC曲线等等，作为用户程序的组成部分。

6-2. 用户程序中的任务与配置

6-2-1. 任务类型

XS3常见的任务有：EtherCAT任务、外部中断任务、普通任务等。



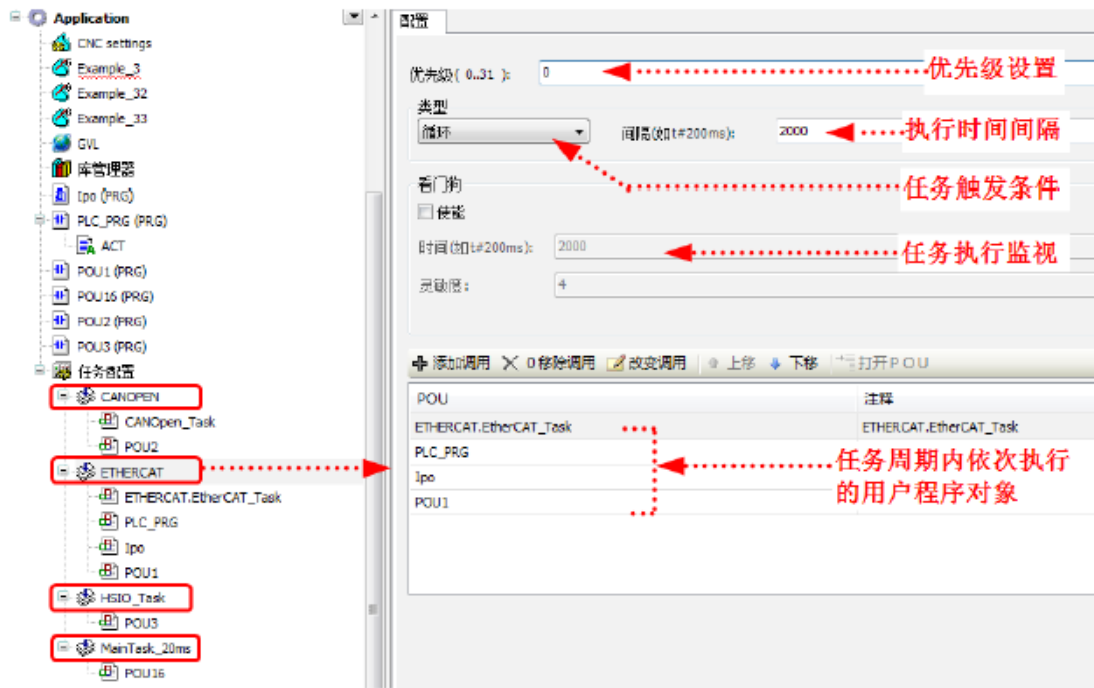
其中EtherCAT任务在XS3中是最为重要的一个任务,运动控制功能的实时处理,都是在这个任务中完成的,它是一个执行时间间隔短、优先级最高的一个时钟中断型的任务,一旦满足时间条件,它可以无条件地中断其他的任务,开始执行EtherCAT任务,直到该任务配置下所有的POU执行完毕,才会退出。

应用系统的同步控制、轨迹控制,往往都有高实时性要求,而逻辑控制的及时性要求则相对较低。在用户程序中,可以将运动控制(MC)的POU放在EtherCAT任务周期中执行,而逻辑控制POU就放在普通任务周期中执行即可。若将特定的程序变量声明为全局变量,就可以在运动控制中,实现与逻辑控制的协调动作。

6-2-2. 任务配置

在一个程序中不同部分的执行通过“任务”来控制。在一个任务配置中可以建立多个任务,而一个任务中,可以调用多个程序组织单元(POU)。每个任务组可以设置不同的执行触发条件、执行时间间隔、优先级等。一旦任务被设置,它就可以控制程序周期执行或者通过特定的事件触发开始执行。

如下图中所示,在CODESYS编程环境的“任务配置”下有4个任务,鼠标选中“任务配置”,点击右键,可以增加新的任务。双击已经存在的任务,右侧窗口可以看到任务的设置参数:



1、任务优先级安排

CODESYS 中可以对任务的优先级进行设置，一共可以设 32 个级别（0~31 之间的一个数字，0 是最高优先级，31 是最低优先级）。

当一个程序在执行时，优先级高的任务优先于优先级任务低的任务，高优先级任务 0 能中断同一资源中较低优先级的程序执行，使较低优先级程序执行被放缓。这样可确保对运动控制等重要任务优先执行，在一些需要高性能运动控制（MC）的应用中，可让控制器的性能得到合理利用。

注意：在任务优先级等级分配时，请勿分配具有相同优先级的任务。如果还存在其他任务视图先于具有相同优先级的任务，则结果可能不确定且不可预知。

默认任务优先级顺序如下：

任务类型	描述
EtherCAT 总线任务	优先级最高，只允许有一个 EtherCAT 任务
外部中断任务	优先级次之，每个外部输入端口事件允许有一个外部中断任务
普通任务	优先级再次之

EtherCAT任务为最高优先级任务，按EtherCAT周期进入该任务，完整执行一遍该任务内的所有POU后才返回较低优先级任务。低优先的任务在执行时，可能被高优先级的任务打断，待高优先级任务执行完成后，返回被打断的任务，继续执行该低优先级任务。

2、任务执行类型

每个任务组配置中需设定任务的执行类型，XS3 支持的任务执行类型如下：

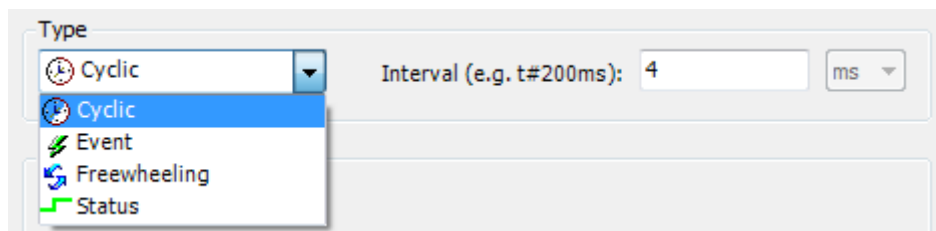
任务执行类型	执行特点	任务举例
循环	在设定的每个时间间隔内执行对应的 POU 一次	EtherCAT 总线任务 普通任务循环
外部事件	可以配置为当 HSIO 状态变化，或高速计数器的读数匹配时，引起对应的 POU 执行一次	HSIO 端口状态中断响应任务 HSIO 端口计数器中断响应任务
惯性滑行（飞轮）	一旦开始执行，即反复循环执行，没有停歇间隔	普通任务循环
事件	在设定的 Bool 型变量状态 0→1 的情况下触发执行一次，而其他的状态组合则不会触发执行	软中断处理 POU
状态	在设定的 Bool 型变量的状态为 1 的情况下，反复循环执行	条件执行任务 POU

3、任务执行周期

“任务执行周期”是指执行该类任务的时间间隔。若任务设为“循环”类型，一般的逻辑控制，普通IO端口变量状态变化比较慢，任务执行周期可以设定的较大一些，比如可设为20ms，若需要及时处理的任务，任务执行周期设定的可以小一些。

CODESYS系统软件中每个“任务”可分配不同的执行周期，有些全局变量可能要在不同的POU之间被访问和修改，于是需要对全局变量进行交互同步，也是在任务的“时间对准点”进行的，因此在设置循环类型任务的周期时，需呈整数倍数的关系。比如，若设EtherCAT周期为1ms、2ms或4ms，则普通循环任务的周期可以为20ms，更低优先级的循环任务周期可以为100ms等等。EtherCAT周期不要设为3ms、6ms、7ms、9ms等，容易造成非整数倍的关系。

如果任务的类型为“Cyclic”，则按照“Interval”中的时间循环执行，具体设置如下图所示：



4、任务执行监视

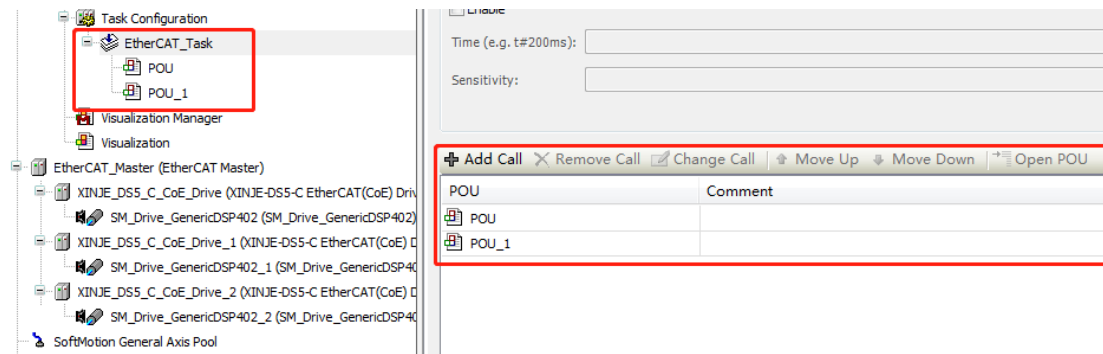
其中 CODESYS 针对每个任务可以配置独立的看门狗，“看门狗”是一种控制器硬件式的计时设备，默认不使用看门狗功能。看门狗的主要功能是监控程序执行时出现的异常或内部时钟发生的故障。如当系统出现死机或当程序进入死循环时，看门狗计时器就会对系统发出重置信号或停止 PLC 当前运行的程序。



“看门狗”触发时间=Time* Sensitivity (ms)，如果程序执行周期超过此触发时间，那么将触发“看门狗”功能并终止当前任务及对应程序。

5、任务配置要点

- EtherCAT 总线通信的任务配置，是一个比较特殊的“循环”类型的任务，具有最高的优先级，其任务周期的设定值，也将是 EtherCAT 总线的通信周期，一般设为 4ms；设定值越小，运动控制的精度越高；当所需控制的轴数越多，则需要设定周期越大，否则容易导致 CPU 的计算负荷过载；
- 有一些任务则在满足某些状态的情况下才开始执行，例如由高速 IO 端口（HSIO）的状态变化（也称 HSIO 中断信号）来引发任务执行，不是按时间间隔来执行；
- 一个任务配置，只能设为一种执行类型、时间间隔及优先级，若要得到不同的执行特性，可以通过添加多个任务配置；
- 每个任务中，都可以指定执行多个用户程序单元（POU），这些 POU 将均以相同的时间间隔执行，执行时按该任务中 POU 的添加顺序执行。如下图所示：



上图中，两个POU的执行顺序依次是POU、POU₁，有全局变量更新操作和判断的情况时，需要注意安排合适的顺序。

6-3. 用户程序由多个 POU 组成的优势

- 不同执行周期的处理程序，应放在不同的 POU 中编写。比如 EtherCAT 周期执行的 POU、外部中断程序 POU、按时间处理的 POU，就必需分成独立的 POU 来编写；
- 为提高程序的可读性，按不同控制工艺段、不同的操作对象、不同的物理结构部件等，分别用不同的 POU 来处理，每个 POU 分别命名为易于理解的名称。如同 C 语言编程，将一个反复调用的处理程序做成一个独立的 POU，方便本工程调用；甚至还方便其他工程沿用；
- 多人合作编程时，每个编程人员各自编写调试自己负责的工艺段的 POU，最后合成为一个用户程序项目；
- CODESYS 编程软件支持 6 种编程语言，根据所需的处理逻辑类型不同，某种的语言可能更方便，而一般情况下，每个 POU 只能用一种编程语言来进行编写，一个项目中若需要同时使用多种编程语言的话，分为多个 POU 来编写也是一个比较好的对策。

7. 用户程序应用编程

本章是基于 XS3 系列 PLC 针对重点功能的应用编程做详细的注解。

7. 用户程序应用编程	53
7-1. 基于 ETHERCAT 运动控制编程	54
7-1-1. XS3 EtherCAT 运控应用的编程要点	54
7-1-2. 单轴控制常用的 MC 功能块	54
7-1-3. MC 指令与 PDO/SDO 配置	55
7-2. 多轴高速定位运动控制编程	56
7-2-1. 高速定位控制功能概述及编程要点	56
7-2-2. 编程示例	58
7-3. 多路高速输入信号采集功能	59
7-3-1. 高速输入采集功能概述及编程要点	59
7-3-2. 编程示例	60
7-4. 扩展模块应用编程	61
7-4-1. 扩展模块应用功能概述及编程要点	61
7-4-2. 编程示例	63
7-5. MODBUS 通讯应用编程	65
7-5-1. MODBUS 通讯功能概述及编程要点	66
7-5-2. 编程示例	67

7-1. 基于 EtherCAT 运动控制编程

7-1-1. XS3 EtherCAT 运控应用的编程要点

XS3 控制器与伺服轴（如信捷 DS5C 系列伺服）配合的运动控制是基于 EtherCAT 总线网络来实现的，与以往的硬件输出的脉冲方式不同，完全是采用软件方式实现，具体来说是在每个很短的 EtherCAT 总线周期中进行一次计算、发布一次控制命令，来实现对伺服的控制。因此，需要注意如下要点：

- 用户MC控制程序，是以EtherCAT任务周期执行的，MC相关的POU应配置在EtherCAT任务下执行，多数MC功能块若放在低优先级的Main任务的POU中，无法正常运行；
- MC功能块的执行，需要通信中的通信数据对象来传递，因此PDO配置表中要有必需的配置项；若遗漏了配置相关的数据对象，伺服可能无法正常运行，且不会有出错报警；
- 控制器可通过SDO的配置对伺服的功能码进行初始化设定操作，设定伺服的运行模式（默认为CSP模式）、伺服电机编码器模式、电子齿轮比等，以确保控制命令与物理运行位置的对应；对伺服的初始化，还可提高设备的调试效率、部件更换后不会出错；
- 对于伺服轴的控制，要遵循轴状态转移的规则和逻辑，根据轴当前的状态和希望的运动，使用合适的MC功能块进行控制；
- 用户程序中使用的是MC功能块的实例，一个MC实例，只能用于一个伺服轴的控制，若同时用于几个伺服轴的控制，会导致控制混乱；
- 一个正在运转的伺服轴，必需有一个MC功能块对其运行进行监控，哪怕是MC_Stop也是一种监控，避免因程序逻辑的跳转导致无MC功能块监控的状态，如果系统会停机报错，这样的错误不容易检查；
- 注意调试的安全处理。若伺服系统采用增量式编码器，正常运行之前需要有归零操作，伺服驱动器的SI信号输入端口，可接入原点位置信号，对于在有限范围内运动（如丝杆），调试前应有极限与安全保护信号。

7-1-2. 单轴控制常用的 MC 功能块

MC 功能块（FB）也称为 MC 指令，准确地讲，用户程序中使用的是 MC 功能块的对象实例，伺服轴通过 MC 对象实例来进行控制。

单轴的控制，一般用于定位的控制，即伺服电机拖动外部机构运动到指定的位置；有时也需要伺服以指定的速度或力矩运行等，在单轴控制中，常用到如下的 MC 功能块：

控制操作	MC 指令	说明
伺服使能	MC_Power	运行该指令，使伺服轴使能，才能进行后续的运行控制
伺服点动运行	MC_Jog	伺服电机的点动运行，常用于低速试车，用于检验设备或调整伺服电机位置
相对定位	MC_MoveRelative	以当前位置为参考，运行指定距离
相对叠加定位	MC_MoveAdditive	命令伺服运行到指定的坐标点
绝对定位	MC_MoveAbsolute	命令伺服运行到指定的坐标点
速度控制	MC_MoveVelocity	命令伺服以指定的速度运行
力矩控制	MC_MoveTorque	命令伺服以指定的力矩运行
伺服暂停	MC_Halt	命令伺服暂停运行，若 MC_Movexxx 再次触发，伺服可以再运行
告警复位	MC_Reset	当伺服出现告警停机后，运行该指令进行复位，
改变伺服运行模式	MC_ControlMode	使用该指令可以让伺服选择“位置”、“速度”或“力矩”模式
伺服原点回归	MC_Home	命令伺服开始原点回归操作，应用系统的原点信号、两侧极限信号等都接在伺服的 SI 端口
控制器原点回归	MC_Homing	控制系统开始原点回归操作，应用系统的原点信号、两侧极限信号等都接在控制器的 SI 端口

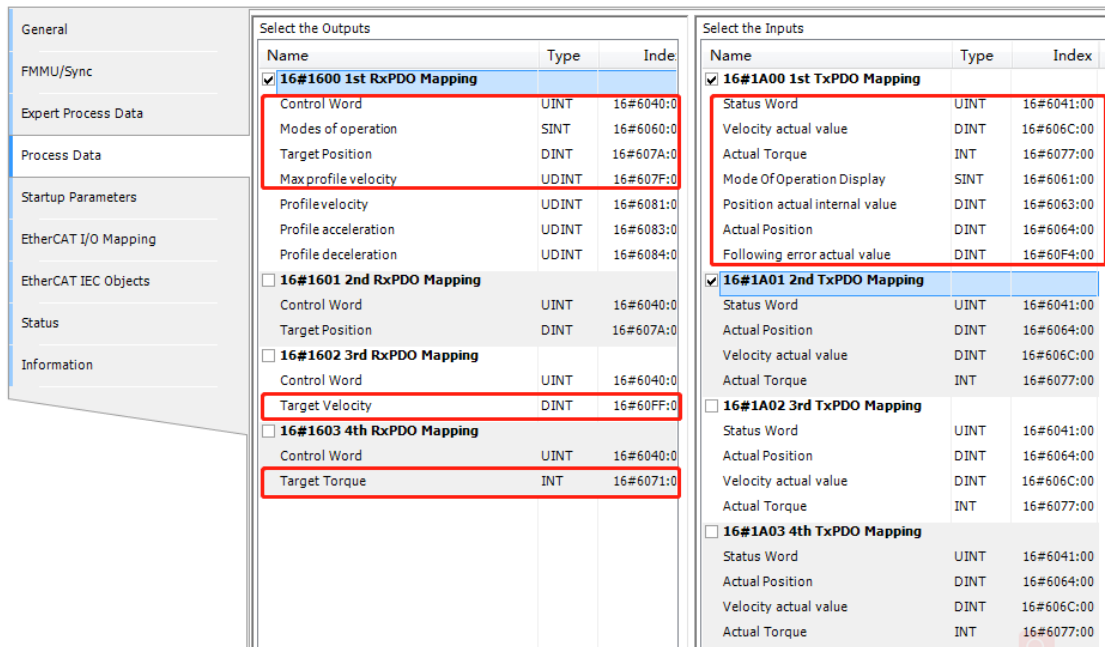
7-1-3. MC 指令与 PDO/SDO 配置

XS3在执行用户程序的伺服轴MC控制命令时，需要将执行MC指令时与伺服交互所需的信息项目添加在通信PDO/SDO配置表中，以便完成所需的控制功能。

MC 指令	所需的 TXPDO 对象	所需的 RXPDO 对象
MC_Power MC_Halt MC_Stop MC_Reset MC_Home MC_Homing	6040h: ControlWord（控制字）	6041h: StatusWord（状态字） 603Fh: Errorcode（错误码）
MC_Jog MC_MoveRelative MC_MoveAdditive MC_MoveAbsolute	607Ah: Target Position（目标位置）	6064h: Position actual value（当前轴位置） 60F4h: Following error actual value（当前跟踪误差）
MC_MoveVelocity	60FFh: Target velocity（目标速度） 607Fh: Max profile velocity（最大内部速度）	606Ch: Velocity actual value（当前实际速度）
MC_MoveTorque	6071h: Target torque（目标力矩）	6077h: Torque actual value（当前实际力矩）
MC_ControlMode	6060h: Modes of operation（运行模式）	6061h: Modes of operation display（运行模式确认）

注：常用的PDO对象说明可见章节4-2-2。

CODESYS配置界面如下图所示：



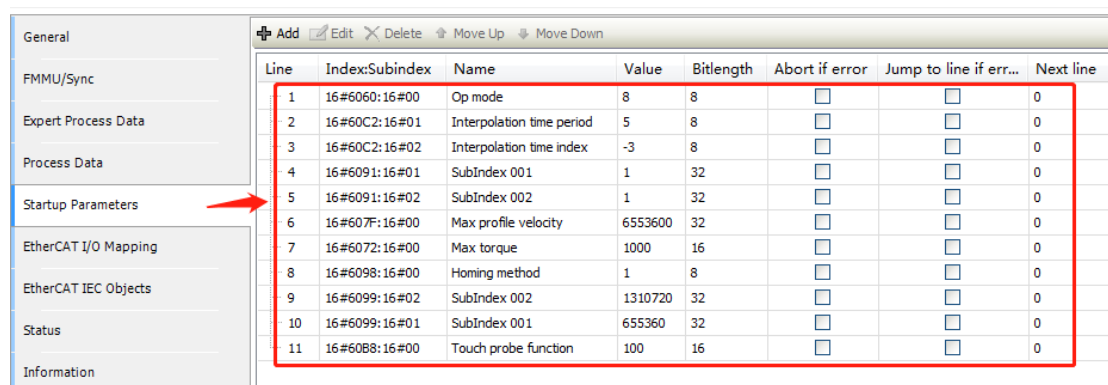
上述TPDO、RPDO 是进行单轴控制所需的基本配置项。

在 MC 控制中，伺服多数情况下为位置模式，尤其是在基于 EtherCAT 总线的应用系统中，为“周期同步位置模式（CSP）”，因此编程时在 SDO 配置中，一般将伺服设置为该运行模式。以信捷 DS5C 系列伺服为例，一般在 SDO 中初始化如下项目：

对伺服的初始化操作	所需的 SDO 对象	说明
运动控制模式	6060h: Modes of operation	按不同模式对应的代码设定
电子齿轮比	6091h : Gear ratio（齿轮比）	--

	6091h-01: Motor revolutions (设定电机旋转数) 6091h-02: Shaft revolutions (设定轴旋转数)	
允许最大转速	607Fh: Max profile velocity (最大内部速度)	适用于速度模式、力矩模式
允许最大力矩	6072h: Max target (最大力矩)	适用于速度模式、力矩模式
回零模式	6098h: Homing method	
回零速度	6099h: Homing speeds 6099h-01: Speed during search 6099h-02: Speed during search for zero	--
探针功能	60B8h: Touchprobe function	--

CODESYS 配置界面如下图所示:



7-2. 多轴高速定位运动控制编程

7-2-1. 高速定位控制功能概述及编程要点

XS3 控制器具有 4 路脉冲输出; 通过使用标准 MC 脉冲指令, 可以进行多种方式脉冲输出, 输出频率最高可达 100KHz。

脉冲输出端口:

脉冲输出端口	脉冲方向端口	最高输出频率	输出方式	输出形式
Y0	Y4	全部0-100KHz	集电极开路方式	脉冲+方向
Y1	Y5	全部0-100KHz	集电极开路方式	脉冲+方向
Y2	Y6	全部0-100KHz	集电极开路方式	脉冲+方向
Y3	Y7	全部0-100KHz	集电极开路方式	脉冲+方向

注:

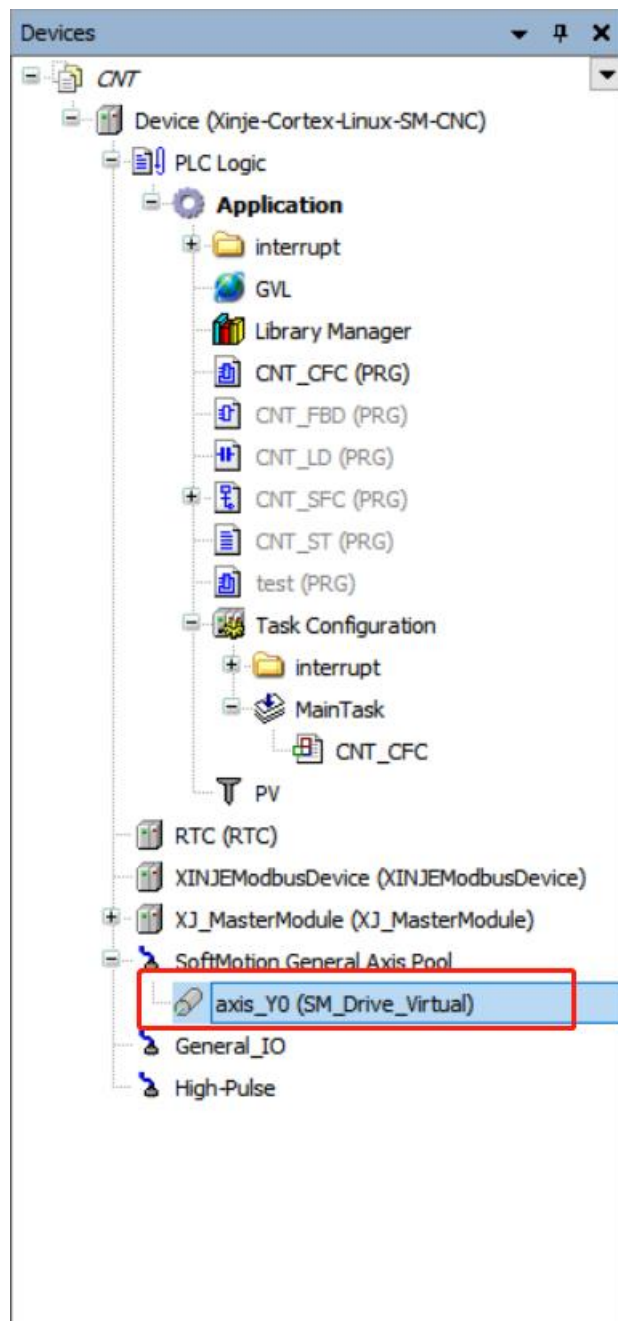
(1) 脉冲输出端子在发脉冲的过程中 LED 指示灯显示微亮, 是由于脉冲输出端子输出脉冲时是占空比为 50% 的方波, 一个脉冲周期一半时间 OFF, 一半时间 ON, 所以脉冲输出的 LED 指示灯也是一半时间亮, 一半时间不亮, 从而显示出微亮。

(2) PLC 可输出 100KHz 到 200KHz 的脉冲, 但无法保证所有伺服正常运行, 请在输出端和 24V 电源之间接入约 500Ω 的电阻。

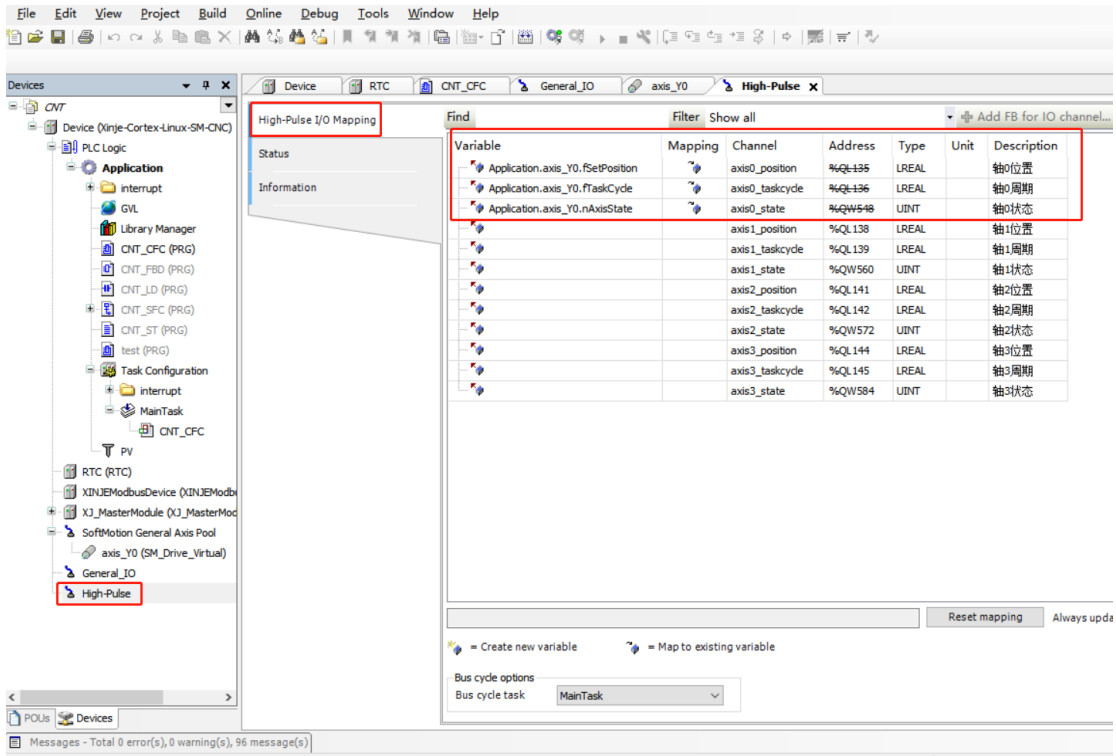
使用 XS3 控制器发脉冲, 需要先进行相关配置:

1、添加虚轴

在“设备窗口”——“SoftMotion General Axis POOL”中添加一个虚轴。

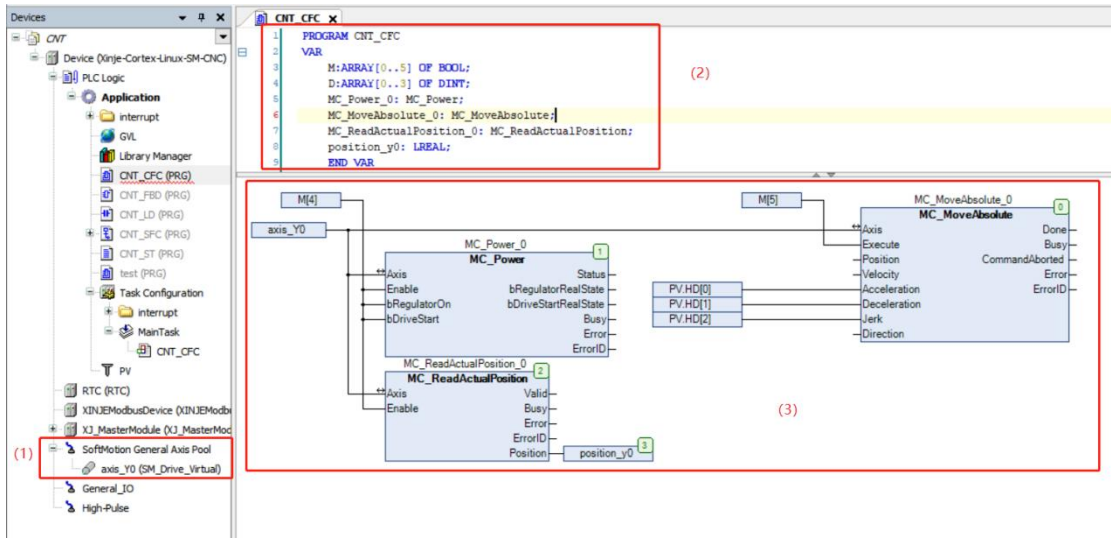


2、在“High-Pulse”中映射变量，“轴0位置”、“轴0周期”、“轴0状态”：



7-2-2. 编程示例

以高速脉冲输出轴 0 按照绝对位置举例：



在 (1) 设备窗口中添加轴 0；

在 (2) 声明编辑器中声明变量；

在 (3) 语言编辑器中写程序：

MC_Power：轴使能指令；

MC_ReadActualPosition：轴实际位置读取指令；

MC_MoveAbsolute：轴绝对位置控制指令；

程序说明：

1、映射到轴

在每个功能块“Axis”中，绑定添加的轴“axis_Y0(SM_Drive_Virtual)”；

2、设定轴运行参数（以MC_MoveAbsolute为例）

在轴运行之前，需要设定轴到达位置(MC_MoveAbsolute.Position)、轴运行速度(MC_MoveAbsolute.Velocity)、加速度(MC_MoveAbsolute.Acceleration)、减速度(MC_MoveAbsolute.Deceleration)、跃度(MC_MoveAbsolute.Jerk)、指令极性(MC_MoveAbsolute.Direction)；

轴运行速度最高 200KHz；

3、轴运行

如上图第（3）部分所示，设定上述轴“轴到达位置”等参数后，先将 M[4]置位，再将 M[5]置位，轴即可按照设定值运行。

7-3. 多路高速输入信号采集功能

7-3-1. 高速输入采集功能概述及编程要点

XS3 控制器具有 4 路高速输入端口；通过使用信捷自研 XJ_Counter 等指令，可以实现高速计数、读高速计数、写高速计数、高速计数中断功能，输入频率最高能接收 200KHz。

高速计数输入端口：

XS3-26T4								
	单相递增模式				AB 相模式			
CounterID	0	1	2	3	0	1	2	3
最高频率	200K	200K	200K	200K	200K	200K	200K	200K
X0+	U+				A+			
X0-	U-				A-			
X1+					B+			
X1-					B-			
X2								
X3+		U+				A+		
X3-		U-				A-		
X4+						B+		
X4-						B-		
X5								
X6+			U+				A+	
X6-			U-				A-	
X7+							B+	
X7-							B-	
X10								
X11+				U+				A+
X11-				U-				A-
X12+								B+
X12-								B-
X13								

使用 XS3 控制器做高速计数时，信捷自研高速计数相关功能块有 4 个分别为：XJ_Counter，XJ_CounterGetValue，XJ_CounterSetValue，XJ_SetCompare，功能块的详细介绍详见第七章。使用上述 4 个功能块前，需要先进行相关配置：

1、设置高速计数端口（CounterID）

每个功能块都要对 CounterID 先进行设置，用以指定高速计数端口，如上文表格所示，CounterID 可设定的值分别为 0、1、2、3，设定好 ID 后，高速输入端口是唯一的。

注：ID 值默认为 0；若同时使用 N 路（ $2 \leq N \leq 4$ ）高速计数，必须在计数使能之前设定 ID 值否则无法正常计数。

2、设置高速计数模式（Mode）

XS3 控制器有两种高速计数模式，分别为单向递增高速计数模式（Mode=TRUE），AB 相高速计数模式（Mode=FALSE），可根据实际情况设定；若不设定，默认为 AB 相高速计数。

注：XS3 控制器高速输入只能接收差分信号（DIFF），不能接收集电极开路信号（OC），务必选用差分信号的编码器或差分板（OC-DIFF）。

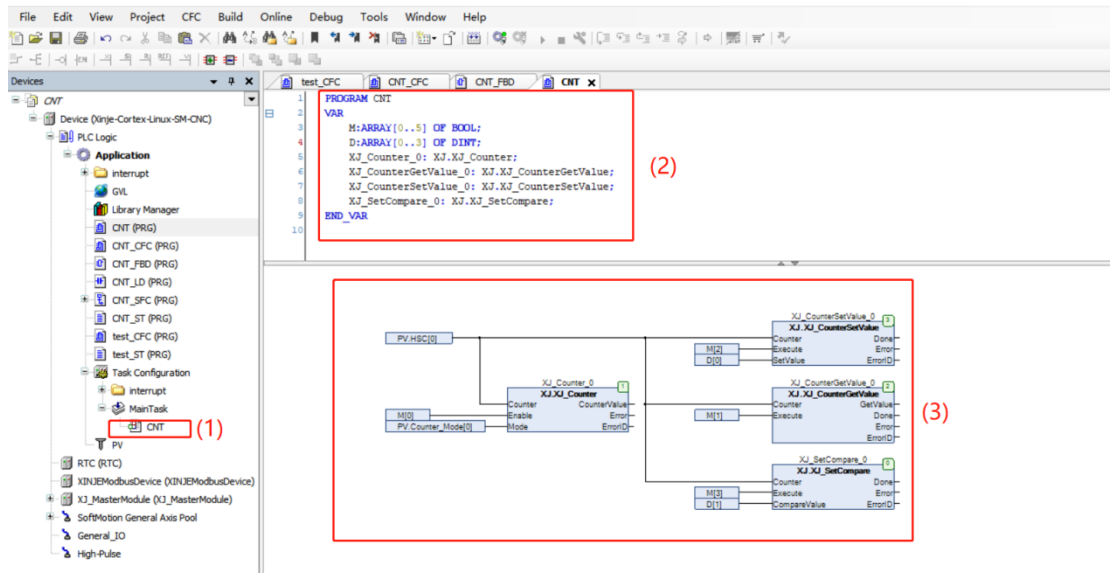
3、其他变量设定

其他变量，如高速计数初值，写高速计数值以及高速计数中断值，可根据实际情况选择设定。

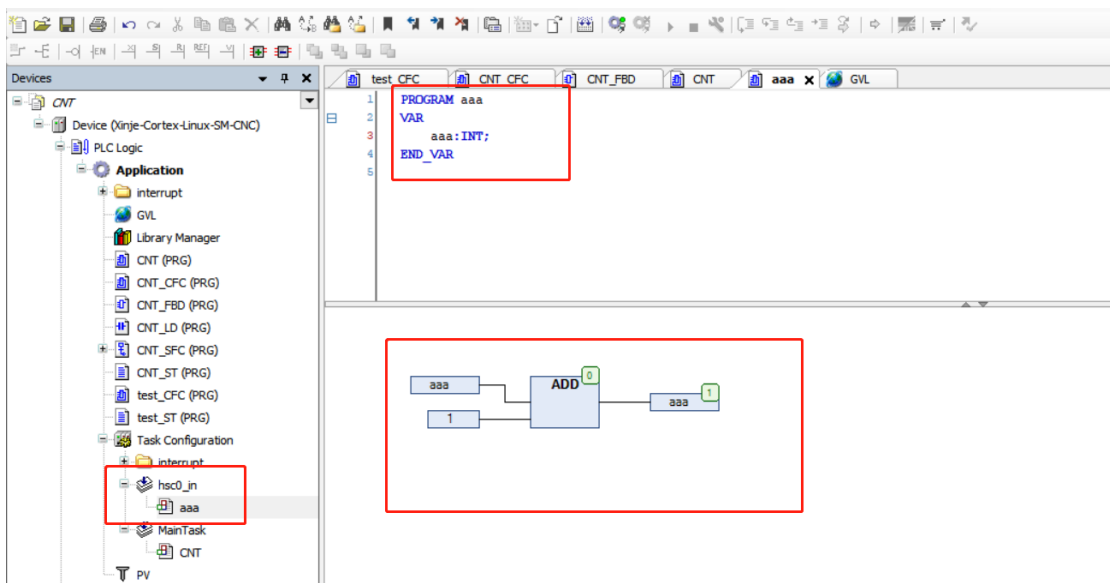
7-3-2. 编程示例

本例中，需用第 1 路高速计数端口进行单相高速计数，并可在计数过程中读取当前计数值，可在计数过程中修改高速计数值，并设置高速计数中断值，使中断 Task 中的程序实现自加 1：

高速计数主程序：



中断程序：



在（1）设备窗口中添加 POU；
 在（2）声明编辑器中声明变量；
 在（3）语言编辑器中写程序；
 XJ_Counter：高速计数；
 XJ_CounterSetValue：写高速计数；
 XJ_CounterGetValue：读高速计数；
 XJ_SetCompare：高速计数中断；

程序说明：

1、设定高速计数端口

4 个功能块的 Counter 输入端都绑定 PV.HSC[0]，设置 CounterID 为 0，即指定第 1 路高速计数端口；设置 CounterValue=0，即设定高速计数初值为 0。

2、设置高速计数模式

本例中要设置 XJ_Counter.Mode=1,设置单相高速计数模式。

3、设置写高速计数值和中断值

将 XJ_CounterSetValue.SetValue 绑定 D[0]，在计数过程中可将 D[0]中的值写入到当前计数值中。

4、设定高速计数中断值

将 XJ_SetCompare.CompareValue 绑定 D[1]，在计数前，将 D[1]中的中断值写入，并关闭使能即可。

5、运行高速计数

设定好中断值之后，开启 M[0]=TRUE,第 1 路高速计数开始计数；

当 M[1]=TRUE 时，读取当前高速计数值，输出到 XJ_CounterGetValue.GetValue；

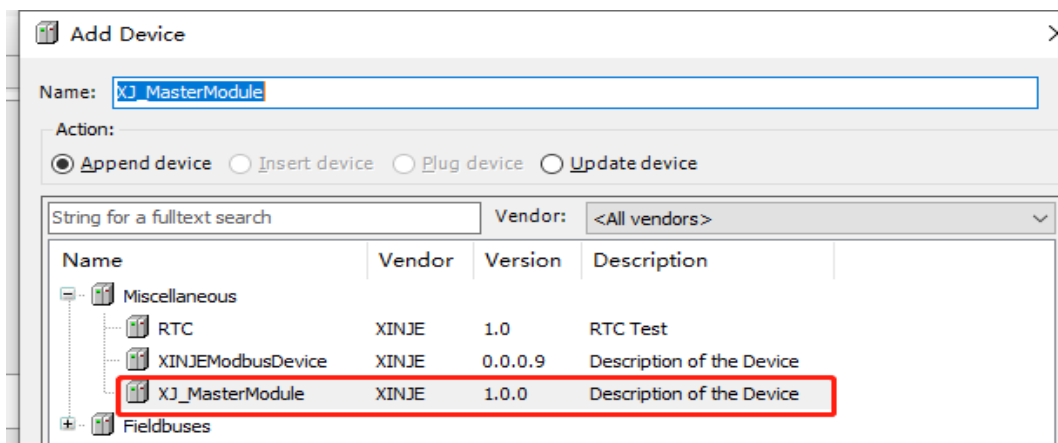
当 M[2]=TRUE 时，将 D[0]中的值写入 XJ_Counter.CounterValue 中；

当 XJ_Counter.CounterValue=D[1]时，中断程序中变量 aaa 会自加 1。

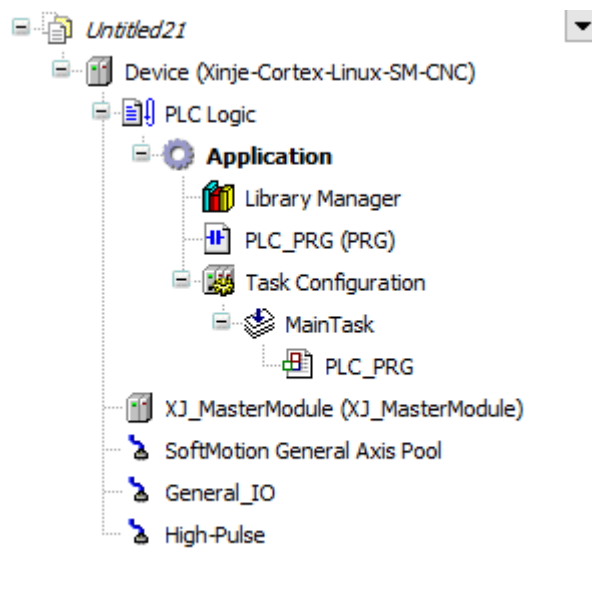
7-4. 扩展模块应用编程

7-4-1. 扩展模块应用功能概述及编程要点

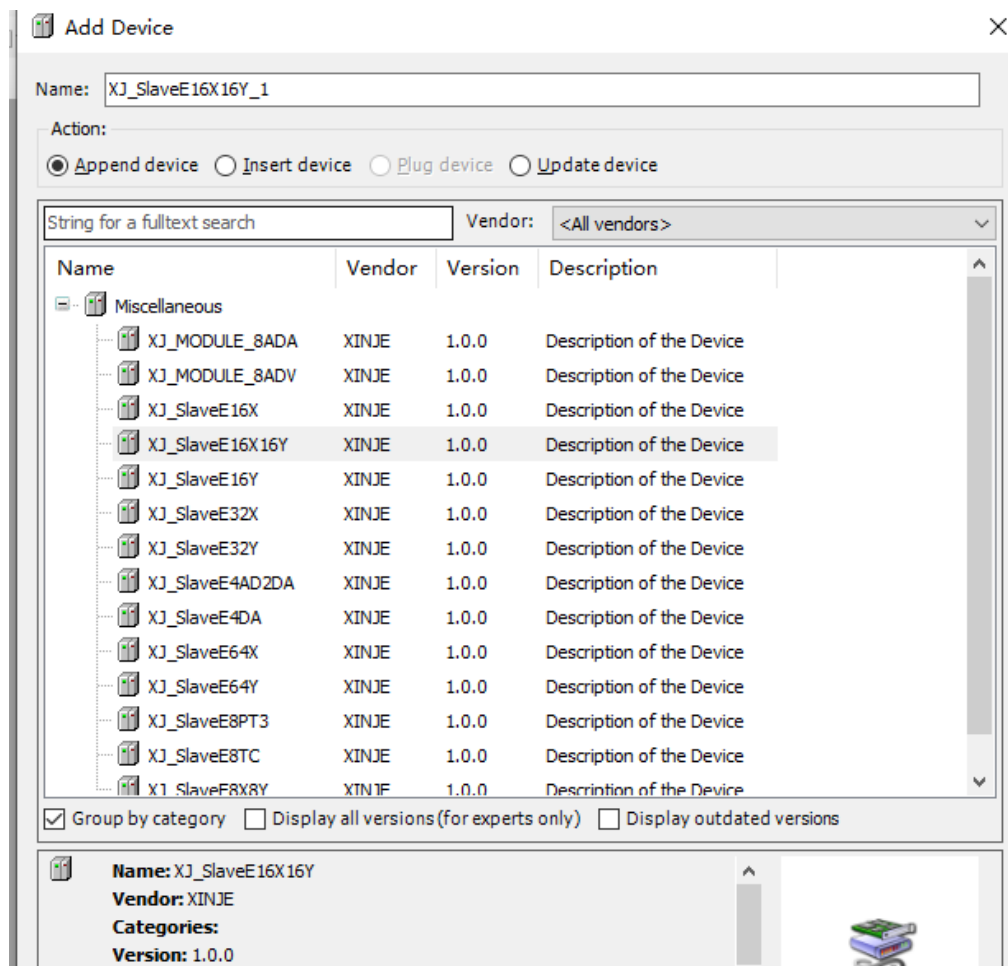
XS3 能匹配 XG 的所有模块，但是如需配置的话首先得先添加信捷的设备描述文件，找到 Device 右键点击 Add Device 会弹出下方窗口点击 XJ_MasterModule 再点击右下角 Add Device 完成添加，如下图：



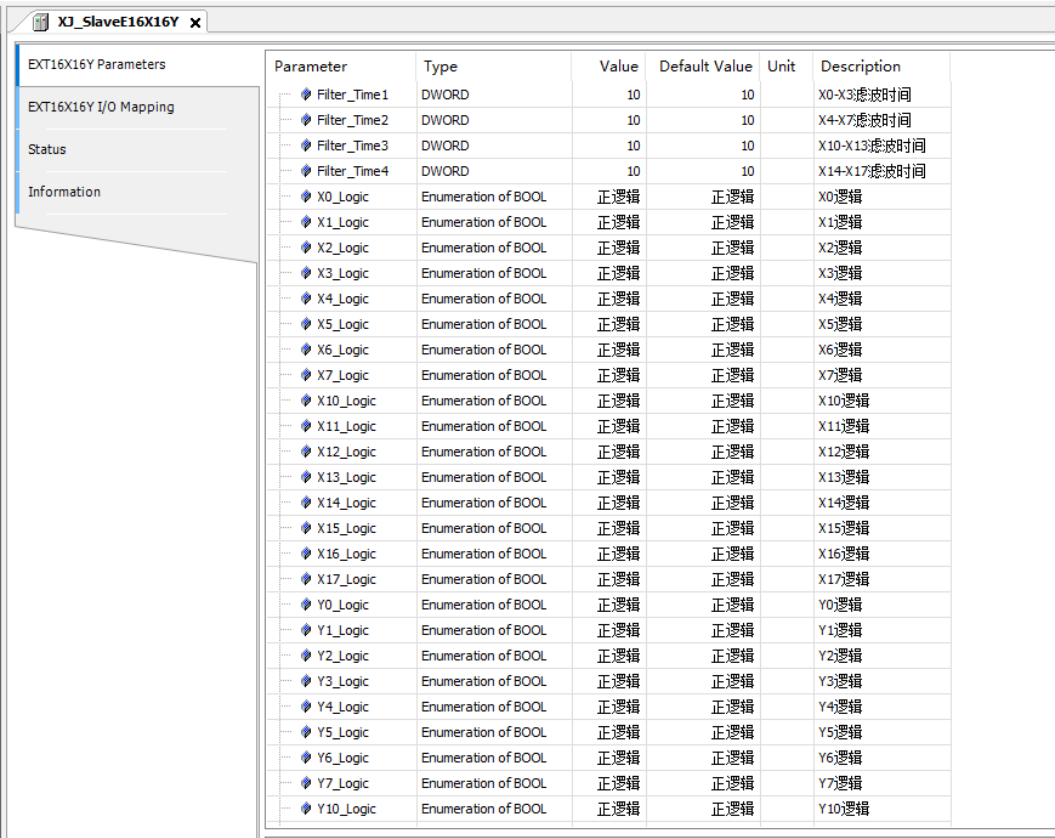
完成后左侧设备窗口如下图所示：



找到 XJ_MasterModule 右键点击 Add Device 就可将 XG 的所有模块添加进模块设备库里：

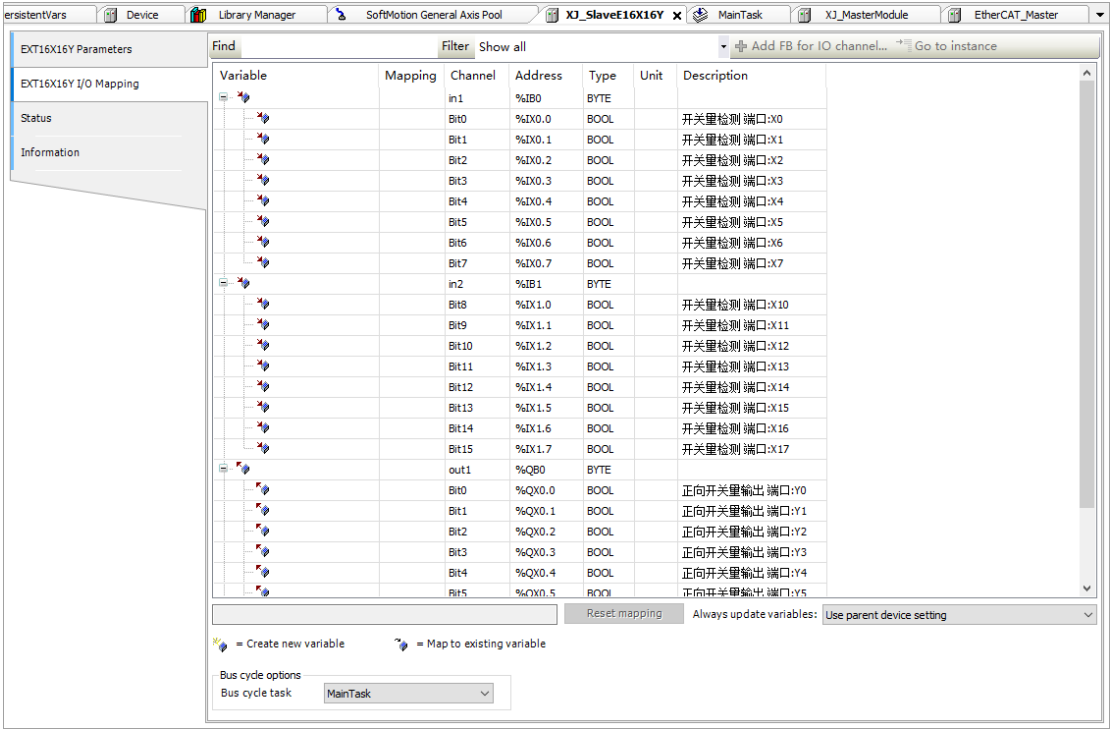


模块以 XG-E16X16Y 为例，添加 XG-E16X16Y 后在左边设备窗口里双击 XJ_SlaveE16X16Y，打开模块配置界面：



Parameter	Type	Value	Default Value	Unit	Description
Filter_Time1	DWORD	10	10		X0-X3滤波时间
Filter_Time2	DWORD	10	10		X4-X7滤波时间
Filter_Time3	DWORD	10	10		X10-X13滤波时间
Filter_Time4	DWORD	10	10		X14-X17滤波时间
X0_Logic	Enumeration of BOOL	正逻辑	正逻辑		X0逻辑
X1_Logic	Enumeration of BOOL	正逻辑	正逻辑		X1逻辑
X2_Logic	Enumeration of BOOL	正逻辑	正逻辑		X2逻辑
X3_Logic	Enumeration of BOOL	正逻辑	正逻辑		X3逻辑
X4_Logic	Enumeration of BOOL	正逻辑	正逻辑		X4逻辑
X5_Logic	Enumeration of BOOL	正逻辑	正逻辑		X5逻辑
X6_Logic	Enumeration of BOOL	正逻辑	正逻辑		X6逻辑
X7_Logic	Enumeration of BOOL	正逻辑	正逻辑		X7逻辑
X10_Logic	Enumeration of BOOL	正逻辑	正逻辑		X10逻辑
X11_Logic	Enumeration of BOOL	正逻辑	正逻辑		X11逻辑
X12_Logic	Enumeration of BOOL	正逻辑	正逻辑		X12逻辑
X13_Logic	Enumeration of BOOL	正逻辑	正逻辑		X13逻辑
X14_Logic	Enumeration of BOOL	正逻辑	正逻辑		X14逻辑
X15_Logic	Enumeration of BOOL	正逻辑	正逻辑		X15逻辑
X16_Logic	Enumeration of BOOL	正逻辑	正逻辑		X16逻辑
X17_Logic	Enumeration of BOOL	正逻辑	正逻辑		X17逻辑
Y0_Logic	Enumeration of BOOL	正逻辑	正逻辑		Y0逻辑
Y1_Logic	Enumeration of BOOL	正逻辑	正逻辑		Y1逻辑
Y2_Logic	Enumeration of BOOL	正逻辑	正逻辑		Y2逻辑
Y3_Logic	Enumeration of BOOL	正逻辑	正逻辑		Y3逻辑
Y4_Logic	Enumeration of BOOL	正逻辑	正逻辑		Y4逻辑
Y5_Logic	Enumeration of BOOL	正逻辑	正逻辑		Y5逻辑
Y6_Logic	Enumeration of BOOL	正逻辑	正逻辑		Y6逻辑
Y7_Logic	Enumeration of BOOL	正逻辑	正逻辑		Y7逻辑
Y10_Logic	Enumeration of BOOL	正逻辑	正逻辑		Y10逻辑

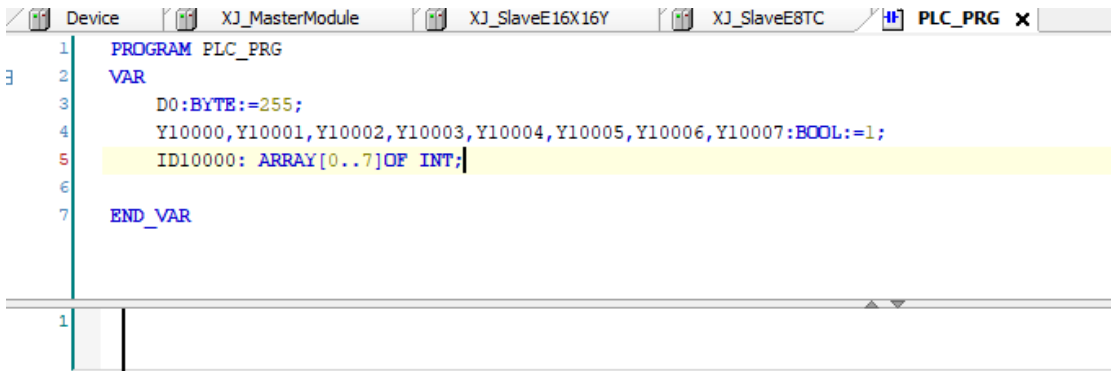
EXT16X16Y Parameters 界面主要是关于模块内部参数的配置，而 EXT16X16Y I/O Mapping 是输入输出映射区，可绑定程序中声明的变量，如下图所示：



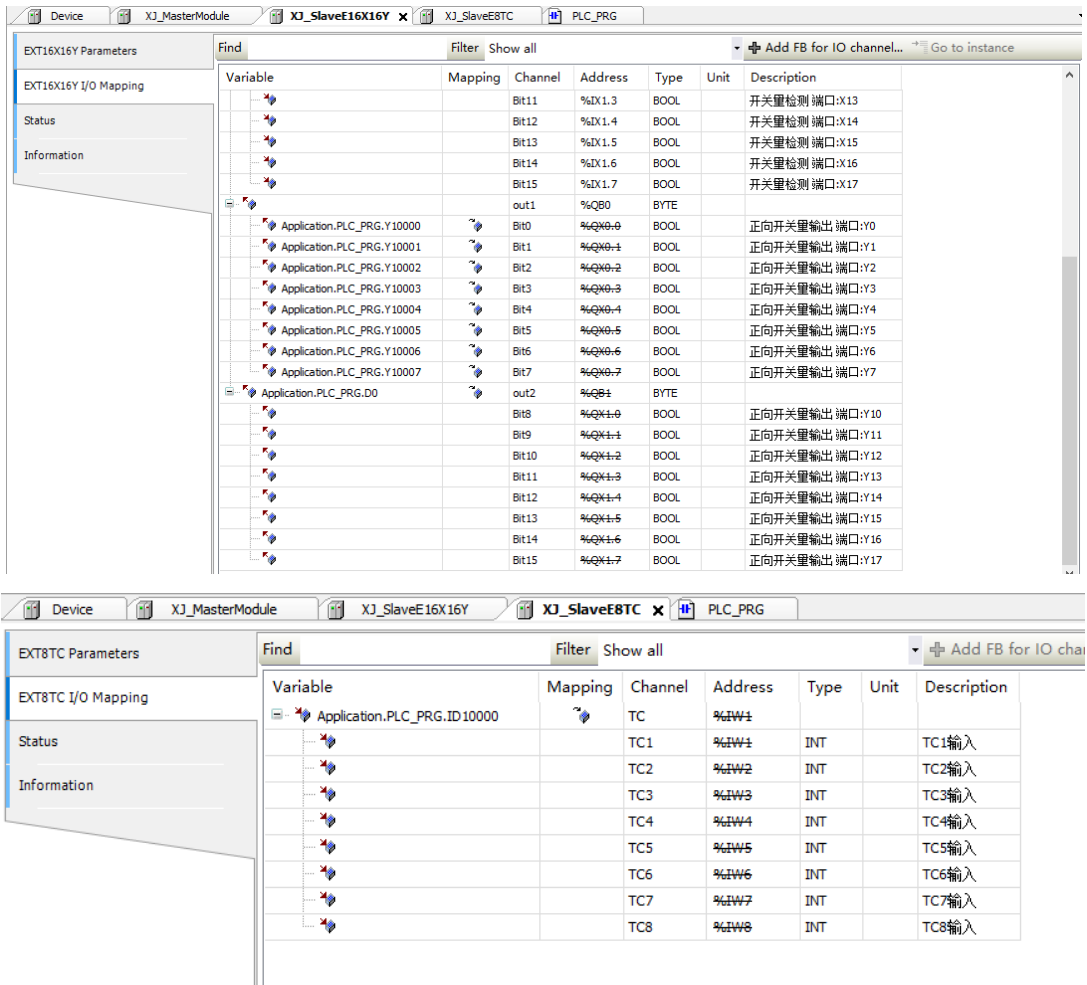
Variable	Mapping	Channel	Address	Type	Unit	Description
		in1	%IB0	BYTE		
		Bit0	%IX0.0	BOOL		开关量检测 端口:X0
		Bit1	%IX0.1	BOOL		开关量检测 端口:X1
		Bit2	%IX0.2	BOOL		开关量检测 端口:X2
		Bit3	%IX0.3	BOOL		开关量检测 端口:X3
		Bit4	%IX0.4	BOOL		开关量检测 端口:X4
		Bit5	%IX0.5	BOOL		开关量检测 端口:X5
		Bit6	%IX0.6	BOOL		开关量检测 端口:X6
		Bit7	%IX0.7	BOOL		开关量检测 端口:X7
		in2	%IB1	BYTE		
		Bit8	%IX1.0	BOOL		开关量检测 端口:X10
		Bit9	%IX1.1	BOOL		开关量检测 端口:X11
		Bit10	%IX1.2	BOOL		开关量检测 端口:X12
		Bit11	%IX1.3	BOOL		开关量检测 端口:X13
		Bit12	%IX1.4	BOOL		开关量检测 端口:X14
		Bit13	%IX1.5	BOOL		开关量检测 端口:X15
		Bit14	%IX1.6	BOOL		开关量检测 端口:X16
		Bit15	%IX1.7	BOOL		开关量检测 端口:X17
		out1	%QB0	BYTE		
		Bit0	%QX0.0	BOOL		正向开关量输出 端口:Y0
		Bit1	%QX0.1	BOOL		正向开关量输出 端口:Y1
		Bit2	%QX0.2	BOOL		正向开关量输出 端口:Y2
		Bit3	%QX0.3	BOOL		正向开关量输出 端口:Y3
		Bit4	%QX0.4	BOOL		正向开关量输出 端口:Y4
		Bit5	%QX0.5	BOOL		正向开关量输出 端口:Y5

7-4-2. 编程示例

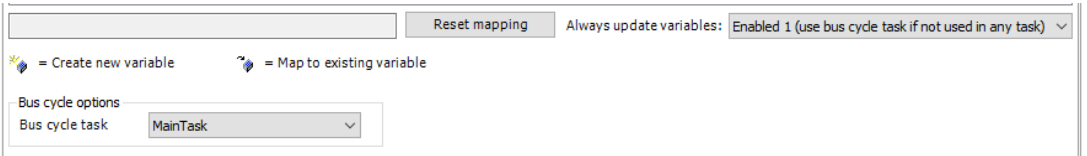
以添加一个 XG-E16X16Y 和一个 XG-8TC-P 模块为例，程序界面如图，定义一个 BYTE 型变量、8 个 BOOL 型变量和一个 INT 数组变量，并给他们赋值。



开关量模块输入输出点可以用 INT、BYTE 或 BOOL 型变量映射：



要使模块运行正常需挂在 MainTask 下：



运行之后程序界面，可以通过更改变量 D0 和 Y 控制 16X16Y 模块输出点，ID10000 数组可以显示 6TC 模块八个通道的温度值。

Expression	Type	Value	Prepared value	Address	Comment
D0	BYTE	255			
Y10000	BOOL	TRUE			
Y10001	BOOL	TRUE			
Y10002	BOOL	TRUE			
Y10003	BOOL	TRUE			
Y10004	BOOL	TRUE			
Y10005	BOOL	TRUE			
Y10006	BOOL	TRUE			
Y10007	BOOL	TRUE			
ID10000	ARRAY [0...7] OF INT				
ID10000[0]	INT	-1			
ID10000[1]	INT	-1			
ID10000[2]	INT	-1			
ID10000[3]	INT	-1			
ID10000[4]	INT	-1			
ID10000[5]	INT	-1			
ID10000[6]	INT	-1			
ID10000[7]	INT	-1			

7-5. MODBUS 通讯应用编程

XS3 可编程控制器本体支持 Modbus 协议通讯主、从机形式。

主站形式:

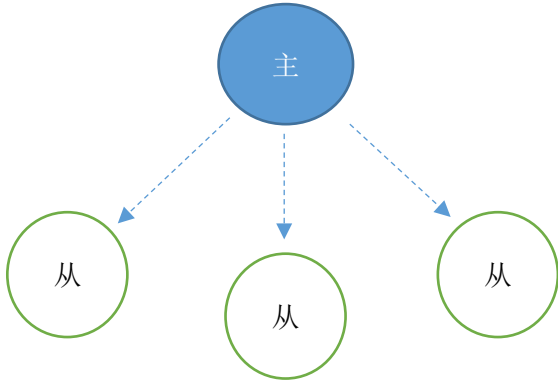
可编程控制器作为主站设备时,通过 Modbus 指令可与其它使用 Modbus-RTU 协议的从机设备通讯;与其他设备进行数据交换。

从站形式:

可编程控制器作为从站设备时,只能对其它主站的要求作出响应。

主从的概念:

在 RS485 网络中,某一时刻,可以有一主多从(如下图),其中主站可以对其中任意从站进行读写操作,从站之间不可直接进行数据交换,主站需编写通讯程序,对其中的某个从站进行读写,从站无需编写通讯程序,只需对主站的读写进行响应即可。(接线方式:所有的 485+连在一起,所有的 485-连在一起。)



在 RS232 网络中(如下图),只能一对一通讯,某一时刻只有一主一从。



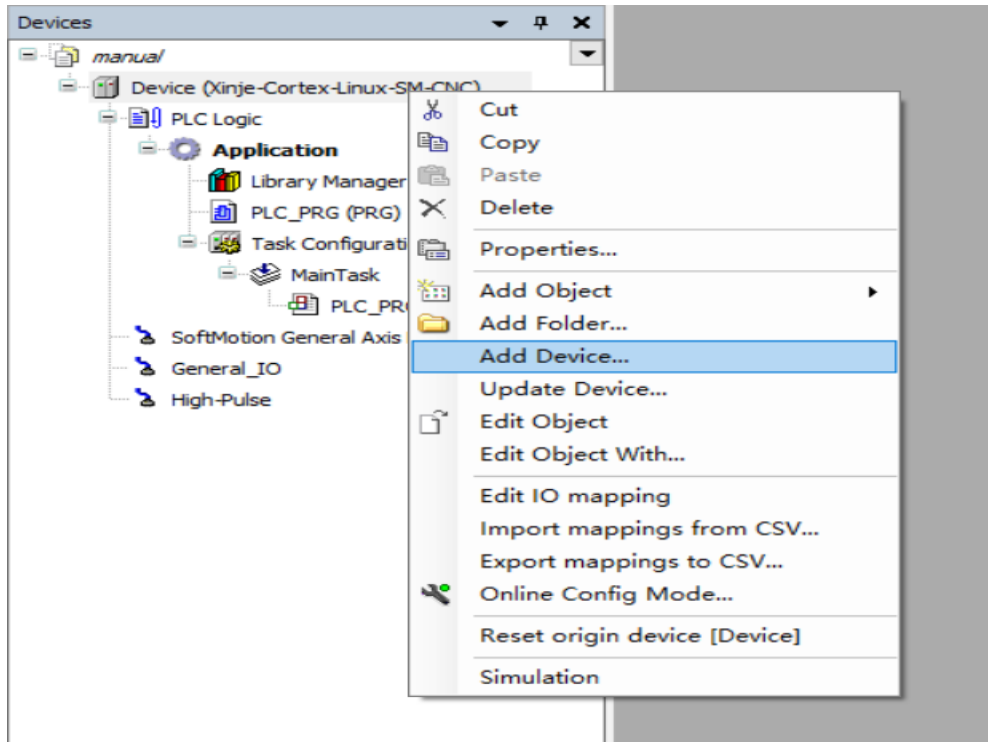
之所以图中有虚线箭头(包括 RS485 网络中),是因为理论上在两个网络中,只要各个 PLC 不发数据,网络中任意 PLC 都可以用来作为主站,其它 PLC 作为从站;但是由于多个 PLC 之间没有一个统一的时钟基准,容易出现在同一时刻有多个 PLC 发送数据,会导致通讯冲突失败,因此不建议这样使用。

7-5-1. MODBUS 通讯功能概述及编程要点

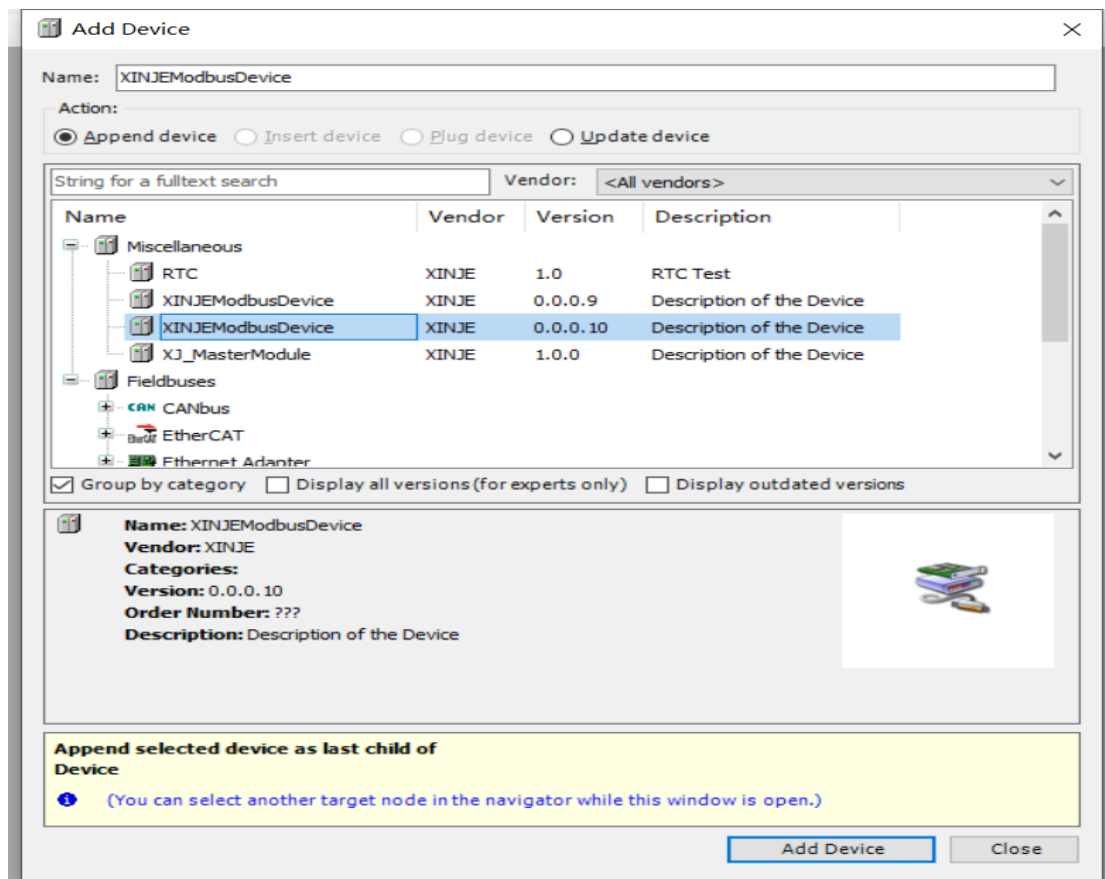
使用 XS3 控制器进行串口通讯时，需要对机器进行一定的配置。配置步骤如下：

串口通讯所需添加配置：

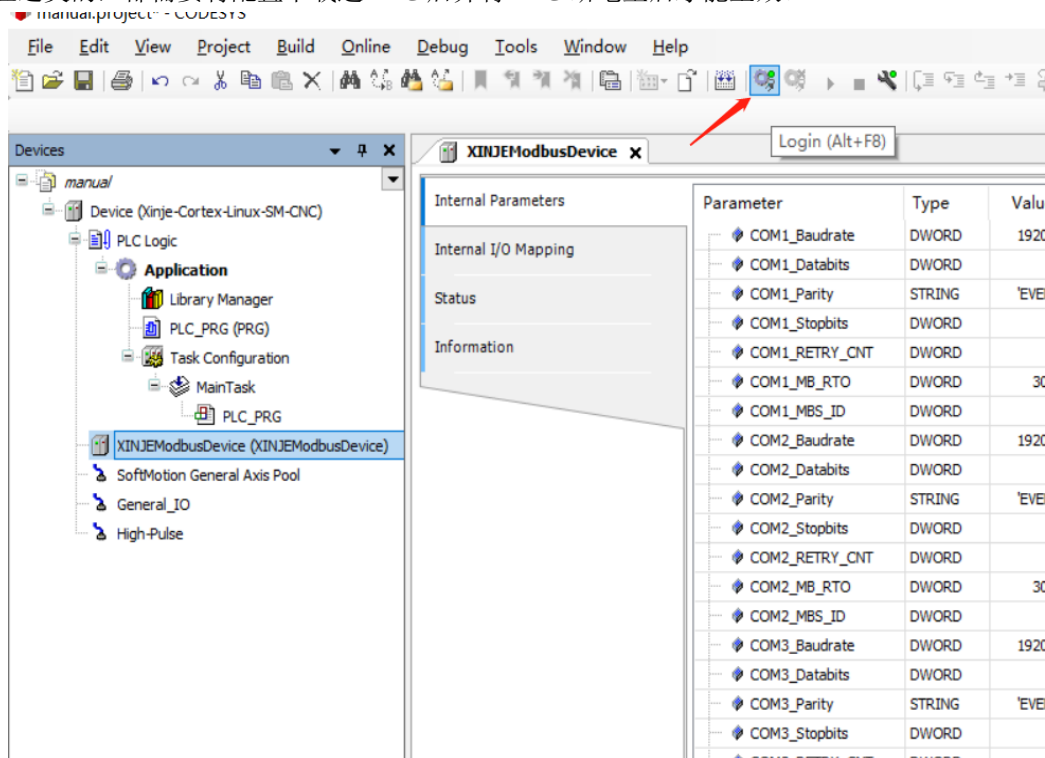
- 1、在总设备处右键后选择添加设备描述文件；



- 2、选择信捷 XINJEModbusDevice 设备描述文件；



3、添加完成串口设备后，配置好串口相关参数（波特率、数据位、站号、校验位、停止位、发送前延时时间、回复超时时间），Login 之后并将 PLC 断电重启后，串口配置即可生效。注意：修改串口配置，类似于波特率、校验位之类的，都需要将配置下载进 PLC 后并将 PLC 断电重启才能生效。



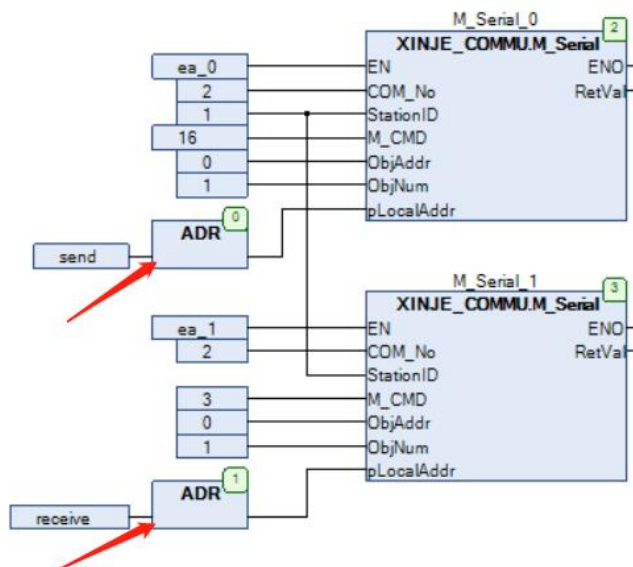
7-5-2. 编程示例

本例用 XS3 控制器作 ModBus 通讯的主站为例：

```

1  PROGRAM com_SFC
2  VAR
3      send:ARRAY[0..49] OF WORD;
4      receive:ARRAY[0..49] OF WORD;
5      ea_0: BOOL;
6      ea_1: BOOL;
7      M_Serial_0: XINJE_COMMU.M_Serial;
8      M_Serial_1: XINJE_COMMU.M_Serial;
9  END_VAR
10

```



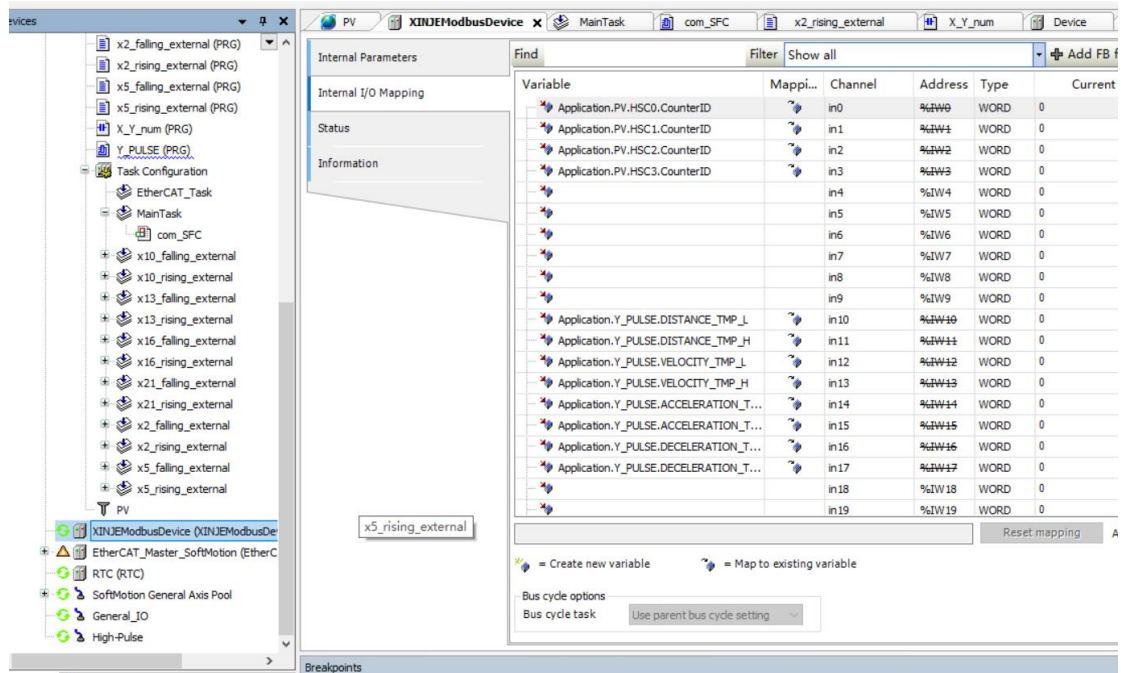
M_Serial_0 实现的功能是：通过串口 2，将 send 数组的第 1 个数据写入 1 号从站以 ModBus 地址为 0 起始的 1 个寄存器。

M_Serial_1 实现的功能是：通过串口 2，读出 1 号从站以 ModBus 地址为 0 起始的 1 个寄存器，并将读出的数据存入 receive 数组的第 1 个元素中。

需要注意的是红标处，需要取出数组的地址后才能传入功能块。

本例用 XS3 控制器作 ModBus 通讯的从站为例：

Codesys 机型做从站时，只需将对应的寄存器、线圈映射到定义的变量，即可在程序中使用，映射如下：



以下为串口配置中寄存器及线圈的 ModBus 地址：

名称	ModBus 地址	类型	描述
In 0~499	0~499	WORD	可读可写
In 500~599	0~99	BOOL	可读可写
Out 0~499	500~999	WORD	只读不可写
Out 500~599	500~599	BOOL	只读不可写

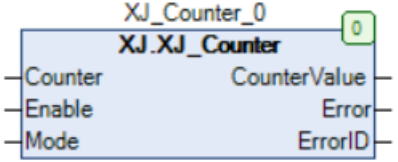
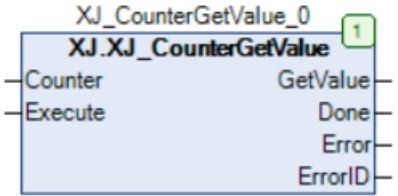
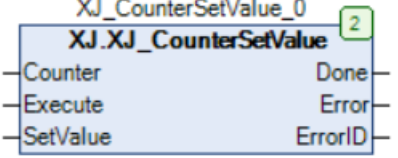
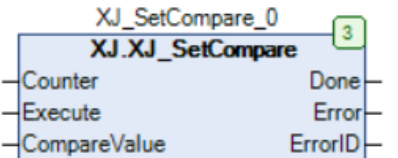
8. 自开发功能块详解

基于 Codesys 平台，信捷开发了相关功能块，便于客户编程使用，本章即是对这些功能块做详细说明。

8. 自开发功能块详解	69
8-1. 高速计数指令	70
8-1-1. 指令格式	70
8-1-2. 相关变量	70
8-1-3. 功能说明	72
8-2. 以太网通讯指令	73
8-2-1. 指令格式	73
8-2-2. 相关变量	73
8-2-3. 功能说明	76
8-3. MODBUS 通讯指令	77
8-3-1. 指令格式	77
8-3-2. 相关变量	77

8-1. 高速计数指令

8-1-1. 指令格式

指令	名称	图形表现	ST表现
XJ_Counter	高速计数		<pre>XJ.XJ_Counter(Counter:= , Enable:= , Mode:= , CounterValue=> , Error=> , ErrorID=>);</pre>
XJ_CounterGetValue	读高速计数		<pre>XJ.XJ_CounterGetValue(Counter:= , Execute:= , GetValue=> , Done=> , Error=> , ErrorID=>);</pre>
XJ_CounterSetValue	写高速计数		<pre>XJ.XJ_CounterSetValue(Counter:= , Execute:= , SetValue:= , Done=> , Error=> , ErrorID=>);</pre>
XJ_SetCompare	高速计数中断		<pre>XJ.XJ_SetCompare(Counter:= , Execute:= , CompareValue:= , Done=> , Error=> , ErrorID=>);</pre>

8-1-2. 相关变量

【XJ_Counter】

① 输入变量

输入变量	名称	数据类型	有效范围	初始值	描述
Counter	计数器	COUNTER_REF	-	-	高速计数器, 规定了高速计数输入端和初值
Enable	使能	BOOL	TRUE,FALSE	FALSE	常开使能进行计数
Mode	计数模式	BOOL	TRUE,FALSE	FALSE	高速计数模式: MODE=FALSE,为AB相高速计数, MODE=TRUE, 为单相高速计数

② 输出变量

输出变量	名称	数据类型	有效范围	初始值	描述
CounterValue	计数值	DINT	数据范围	0	高速计数值
Error	错误标志	BOOL	TRUE,FALSE	FALSE	
ErrorID	错误类型	UINT	-	0	

【XJ_CounterGetValue】

① 输入变量

输入变量	名称	数据类型	有效范围	初始值	描述
Counter	计数器	COUNTER_REF	-	-	高速计数器，规定了高速计数输入端和初值
Execute	使能	BOOL	TRUE,FALSE	FALSE	上升沿触发，读取当前高速计数值

② 输出变量

输出变量	名称	数据类型	有效范围	初始值	描述
GetValue	读取值	DINT	数据范围	0	当前计数值
Done	完成标志	BOOL	TRUE,FALSE	FALSE	完成读取后，标志位TRUE
Error	错误标志	BOOL	TRUE,FALSE	FALSE	
ErrorID	错误类型	UINT	-	0	

【XJ_CounterSetValue】

① 输入变量

输入变量	名称	数据类型	有效范围	初始值	描述
Counter	计数器	COUNTER_REF	-	-	高速计数器，规定了高速计数输入端和初值
Execute	使能	BOOL	TRUE,FALSE	FALSE	上升沿触发，写高速计数值，将 SetValue 的值写入 CounterValue
SetValue	写入值	DINT	数据范围	0	写高速计数设定值

② 输出变量

输出变量	名称	数据类型	有效范围	初始值	描述
Done	完成标志	BOOL	TRUE,FALSE	FALSE	完成写入后，标志位TRUE
Error	错误标志	BOOL	TRUE,FALSE	FALSE	
ErrorID	错误类型	UINT	-	0	

【XJ_SetCompare】

① 输入变量

输入变量	名称	数据类型	有效范围	初始值	描述
Counter	计数器	COUNTER_REF	-	-	高速计数器，规定了高速计数输入端和初值
Execute	使能	BOOL	TRUE,FALSE	FALSE	上升沿触发，设定高速计数中断比较值
CompareValue	比较值	DINT	数据范围	0	高速计数比较值

② 输出变量

输出变量	名称	数据类型	有效范围	初始值	描述
Done	完成标志	BOOL	TRUE,FALSE	FALSE	比较值设置完成，标志位TRUE
Error	错误标志	BOOL	TRUE,FALSE	FALSE	
ErrorID	错误类型	UINT	-	0	

注：使用高速计数比较时，EXecute 使能不能常开，常开会使用高速计数值产生偏差。

8-1-3. 功能说明

1、高速计数功能共有 4 个功能块，分别为高速计数功能块、读高速计数功能块、写高速计数功能块、高速计数比较功能块；XS3 系列高速输入只能接收差分信号（DIFF），不能接收集电极开路信号（OC），请务必选用差分信号的编码器。

2、Counter 为 COUNTER_REF 数据类型：

COUNTER_REF 具体描述如下：

成员	名称	数据类型	有效范围	初始值	描述
CounterID	计数器端口	INT	0,1,2,3	0	选择高速计数器输入端口
CounterValue	计数器初值	DINT	数据范围	0	设定计数器初值

3、XS3 系列高速计数功能有两种模式，分别为单向递增模式和 AB 相模式。

（1）递增模式（Mode=TRUE）

此模式下，计数输入脉冲信号，计数值随着每个脉冲信号的上升沿递增计数。

（2）AB 相模式（Mode=FALSE）

此模式下，高速计数值依照相位差 90° 的脉冲信号（A 相和 B 相）进行递增或递减计数，默认计数模式为 4 倍频。

4、XS3 系列高速计数输入端口分配

XS3-26T4								
	单相递增模式				AB相模式			
CounterID	0	1	2	3	0	1	2	3
最高频率	200K	200K	200K	200K	200K	200K	200K	200K
X0+	U+				A+			
X0-	U-				A-			
X1+					B+			
X1-					B-			
X2								
X3+		U+				A+		
X3-		U-				A-		
X4+						B+		
X4-						B-		
X5								
X6+			U+				A+	
X6-			U-				A-	
X7+							B+	
X7-							B-	
X10								
X11+				U+				A+
X11-				U-				A-
X12+								B+
X12-								B-
X13								

8-2. 以太网通讯指令

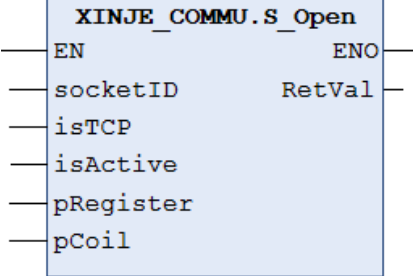
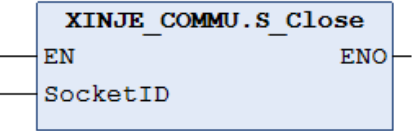
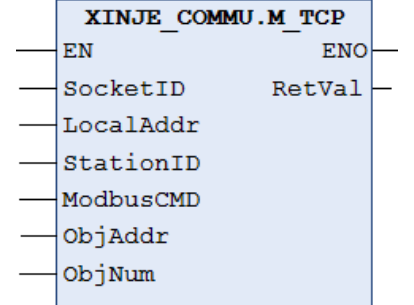
以太网通讯指令包括：通讯任务的开启和关闭、MODBUS TCP 通讯指令。使用以太网指令时，请按照以下步骤进行：

- (1) 开启通讯任务：确认通信协议和通信类型，配置通信参数，创建 TCP 连接端口监听，并绑定套接字 ID。
- (2) 实现数据通信：开启成功的通讯任务，实现以太网自由通信或 MODBUS TCP 数据通讯。
- (3) 关闭通讯任务：当与目标通讯设备通讯完成后，或 TCP 连接出现异常时，需要关闭通讯任务。

8-2-1. 指令格式

(1) 通讯任务创建指令 XINJE_COMMU.S_OPEN，与终止通讯任务指令 XINJE_COMMU.S_Close 配合使用。

(2) PLC 作为客户端时，实现 MODBUS TCP 协议的数据收发指令。与创建通讯任务指令 XINJE_COMMU.S_OPEN、终止通讯指令 XINJE_COMMU.S_CLOSE 指令配合使用。

指令	名称	图形表现	ST表现
XINJE_COMMU.S_Open	创建TCP连接		<pre> XINJE_COMMU.S_Open(socketID:= , isTCP:= , isActive:= , pRegister:= , pCoil:= , RetVal=>); </pre>
XINJE_COMMU.S_Close	TCP关连接指令		<pre> XINJE_COMMU.S_Close(SocketID:=); </pre>
XINJE_COMMU.M_TCP	Modbus TCP 通讯指令		<pre> XINJE_COMMU.M_TCP(SocketID:= , LocalAddr:= , StationID:= , ModbusCMD:= , ObjAddr:= , ObjNum:= , RetVal=>); </pre>

8-2-2. 相关变量

【XINJE_COMMU.S_Open】

输入输出变量

输入输出变量	名称	数据类型	有效范围	描述
pRegister	参数块起始地址	ARRAY OF WORD		共占用连续9个寄存器
pCoil	标志起始地址	ARRAY OF BOOL		共占用连续10个线圈

以定义数组 pRegister: ARRAY [0..8] OF WORD 为例，该数组为参数块的功能定义如下表：

数组地址	名称	说明
pRegister[0]	本机端口	目前只开放Modbus TCP通讯，此参数无需设置。
pRegister[1]	目标IP第一、二段	目标 IP 是指目标通信设备的 IP 地址，和本机在同一个子网内。高字节为第一段，低字节为第二段；例：目标IP为192.168.6.60，将第一、二段转化为16进制填入即为0xC0A8或49320。
pRegister[2]	目标IP第三、四段	高字节为第三段，低字节为第四段；例：目标IP为192.168.6.60，将第三、四段转化为16进制填入即为0x063C或1596。
pRegister[3]	目标端口	目标通信设备的网络端口号。目前只开放 Modbus TCP 通讯，目标端口必须为 502。
pRegister[4]	数据缓冲方式	取值为 0 时，使用 8 位存储方式；为1时，采用 16 位存储方式。 目前只开放Modbus TCP通讯，此参数设置无效。
pRegister[5]	接收超时时间	指PLC产生接收数据请求到该动作终止的总时间。取值范围0-65536，单位是10ms。 设置为0表示不启用接收超时，连续接收数据；设为非0时，启用接收超时。接收超时时间对M_TCP指令有效。 如设置接收超时300ms：请求产生开始等待对方回应300ms，成功接收数据后立即终止，超过300ms未能接收到有效数据，结束当前指令并报接收超时错误。
pRegister[6]	预留	
pRegister[7]	实际接收字节数	目前只开放Modbus TCP通讯，此参数无效。
pRegister[8]	错误码	Modbus TCP 通讯发生异常时的错误信息。详见下表。

Modbus TCP 通讯异常错误码

错误码	错误说明
0	通讯正常
1	需要OPEN的套接字已经建立了连接
2	创建套接字时返回错误
3	绑定到指定的端口失败
4	TCPServerAccept 失败
5	TCPClientConnect 失败
6	调用Send、Recv、Close 时，指定的套接字未建立连接
7	调用Send返回失败
8	调用Recv返回失败
10	指定的发送数据长度大于允许范围
11	指定的接收数据长度大于允许范围
20	UDP通讯时，收到的数据不是来自指定的IP
21	UDP通讯时，收到的数据不是来自指定的Port
30	实际收到的数据长度大于指定长度
31	实际收到的数据长度小于指定长度
40	接收超时
50	指定目标端口号错误，MODBUS TCP不是502端口； 使用端口越界（不在1~60000之间）
100	接收错误
101	接收超时
182	站号错误

错误码	错误说明
183	发送缓存区溢出
400	功能码错误
401	地址错误
402	长度错误
403	数据错误
404	从站忙
405	内存错误（擦写Flash）

以定义数组 pCoil: ARRAY [0..9] OF BOOL 为例，该数组为标志位的功能定义如下表：

数组地址	名称	说明
PCoil[0]	连接中标志	连接建立过程中，PCoil[0] 置ON
PCoil[1]	已连接标志	连接建立完成时，PCoil[1] 置ON
PCoil[2]	发送中标志	数据发送过程中，PCoil[2] 置ON
PCoil[3]	已发送标志	发送数据完成时，PCoil[3] 置ON
PCoil[4]	接收中标志	数据接收过程中，PCoil[4] 置ON
PCoil[5]	已接收标志	接收数据完成时，PCoil[5] 置ON
PCoil[6]	关闭中标志	正在关闭当前连接时，PCoil[6] 置ON
PCoil[7]	MODBUS TCP 通信中标志	正在执行 M_TCP 指令时，PCoil[7] 置ON
PCoil[8]	TCP 异常标志	TCP 连接异常时，PCoil[8] 置ON
PCoil[9]	错误标志	发生通讯错误时，PCoil[9] 置ON

输入变量

输入变量	名称	数据类型	有效范围	描述
EN	使能	BOOL		一次上升沿调用创建一次TCP连接
socketID	套接字ID	WORD	0-63	同时建立的TCP数量不超过 32 个
isTCP	通信类型	WORD	0,1	K0为UDP，K1为TCP； 目前只开放了K1（TCP）
isActive	模式选择	WORD	0,1	K0为服务器，K1为客户端； 目前只开放了K1（客户端）

输出变量

输出变量	名称	数据类型	有效范围	描述
ENO	使能输出	BOOL		当没有发生错误时，ENO = EN
RetVal	返回值			暂未开放

注意：ENO 输出不代表连接建立成功，连接是否成功请查看 pCoil 数据块中的返回值。

【XINJE_COMMU.S_Close】

输入变量

输入变量	名称	数据类型	有效范围	描述
EN	执行	BOOL		上升沿来临时，终止通信任务。
socketID	套接字ID	WORD	0-63	指定要关闭的套接字ID，范围0-63。

【XINJE_COMMU.M_TCP】

输入变量

输入变量	名称	数据类型	有效范围	描述
EN	执行	BOOL		一次上升沿进行一次Modbus TCP 通讯
socketID	套接字ID	WORD	0-63	指定使用的TCP连接，最多建立32个
LocalAddr	本地首地址	ARRAY		指定本地的通讯首地址
StationID	远端通讯站号	WORD	0-247	该功能暂未开放，可以为任意值
ModbusCMD	MODBUS通讯功能码	WORD	1, 2, 3, 4, 5, 6, 15, 16	详见下表
ObjAddr	目标首地址	WORD		此处为从站的 MODBUS 通讯地址
ObjNum	通讯数据个数	WORD		详见下表

输出变量

输出变量	名称	数据类型	有效范围	描述
ENO	使能输出	BOOL		当没有发生错误时，ENO = EN
RetVal	返回值			暂未开放

8-2-3. 功能说明

【XINJE_COMMU.S_OPEN】

通过 XINJE_COMMU.S_Open 指令建立连接，创建功能块时请选择 EN/ENO，指令为 EN 上升沿触发。

一个连接使用一条指令，最多可以建立 32 个连接，连接建立成功后使用 XINJE_COMMU.M_TCP 进行设备间通讯，或使用 XINJE_COMMU.S_Close 指令关闭连接。

执行该功能块时必须先在 pRegister 数据块中设置必要的参数，使能后连接的状态可以查看 pCoil 数据块中的返回值。

ST 语言中的 pCoil 定义类型不能为 ARRAY [0..8] OF BOOL，必须为 ARRAY [0..8] OF BYTE 类型。

XINJE_COMMU.S_Open 该指令无法在 CFC 语言中使用，可以在 ST 语言和梯型图语言中使用。

【XINJE_COMMU.S_Close】

通过 XINJE_COMMU.S_Close 指令关闭连接，创建功能块时请选择 EN/ENO，指令为 EN 上升沿触发。该指令无法单独使用，需和 S_OPEN 指令配合使用。

指令执行后，基于此套接字 ID 的 M_TCP 指令将无法执行。

【XINJE_COMMU.M_TCP】

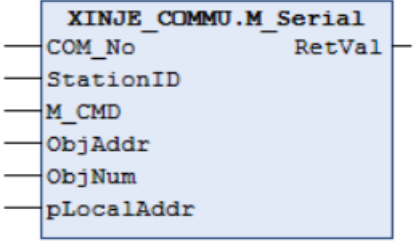
功能码及通讯个数说明：

数值	功能码	说明
K1	读线圈	读取0X类型地址，最大数量2000个
K2	读输入离散量	读取1X类型地址，最大数量2000个
K3	读寄存器	读取4X类型地址，最大数量120个
K4	读输入寄存器	读取3X类型地址，最大数量120个
K5	写单个线圈	写单个0X类型地址
K6	写单个寄存器	写单个4X类型地址
K15	写多个线圈	写0X类型地址，最大数量2000个
K16	写多个寄存器	写4X类型地址，最大数量120个

通过 XINJE_COMMU.M_TCP 指令进行设备间的数据通讯，创建功能块时请选择 EN/ENO。

8-3. MODBUS 通讯指令

8-3-1. 指令格式

指令名称	名称	图形表现	ST表现
M_Serial	串口通讯指令		<pre> XINJE_COMMU.M_Serial(COM_No:= , StationID:= , M_CMD:= , ObjAddr:= , ObjNum:= , pLocalAddr:= , RetVal=>); </pre>

8-3-2. 相关变量

1) 输入变量

输入变量	名称	数据类型	有效范围	初始值
COM_NO	串口	BYTE	1、2、3	0
StationID	从站ID	BYTE		0
M_CMD	功能码	BYTE	标准ModBus功能码	0
ObjAddr	目标地址	WORD		0
ObjNum	操作数	WORD		0
pLocalAddr	本地数组地址	POINTER TO WORD		0

2) 输出变量

输出变量	名称	数据类型	有效范围	初始值
RetVal	暂未使用	暂未使用	暂未使用	暂未使用

3) 变量描述

输入变量:

输入变量	描述
COM_NO	可备选项有: 1、2、3。以此来选择不同的串口来发送数据。
StationID	ModBus通讯中从站所设置的ID。
M_CMD	01H 读线圈、02H 读输入线圈、03H 读寄存器内容、04H 读输入寄存器、05H 写单个线圈、06H 写单个寄存器、10H 写多个寄存器、0FH 写多个线圈
ObjAddr	请查阅相关从站寄存器、线圈的ModBus地址。
ObjNum	读出或者写入寄存器、线圈个数。
pLocalAddr	该参数为数组地址

输出变量:

输出变量	描述
RetVal	暂未使用

4) 指令错误标志:

串口	错误标志	数据类型	默认值	描述
COM1	COMStatus0	BOOL	FALSE	FALSE: 通讯正常。 TURE: 通讯错误, 即COMResult0不为0。

串口	错误标志	数据类型	默认值	描述
	COMResult0	WORD	0	0: 正确 100: 接收错误 101: 接收超时 180: CRC 错误 181: LRC 错误 182: 站号错误 183: 发送缓冲区溢出 400: 功能码错误 401: 地址错误 402: 长度错误 403: 数据错误 404: 从站忙 405: 内存错误 (擦写 FLASH)
COM2	COMStatus10	BOOL	FALSE	FALSE: 通讯正常。 TURE: 通讯错误, 即COMResult10不为0。
	COMResult10	WORD	0	0: 正确 100: 接收错误 101: 接收超时 180: CRC错误 181: LRC错误 182: 站号错误 183: 发送缓冲区溢出 400: 功能码错误 401: 地址错误 402: 长度错误 403: 数据错误 404: 从站忙 405: 内存错误 (擦写 FLASH)
COM3	COMStatus20	BOOL	FALSE	FALSE: 通讯正常。 TURE: 通讯错误, 即COMResult20不为0。
	COMResult20	WORD	0	0: 正确 100: 接收错误 101: 接收超时 180: CRC 错误 181: LRC 错误 182: 站号错误 183: 发送缓冲区溢出 400: 功能码错误 401: 地址错误 402: 长度错误 403: 数据错误 404: 从站忙 405: 内存错误 (擦写 FLASH)

注意: 该通讯指令最多可以同时导通 50 条。最大通讯量为 256 个字节。

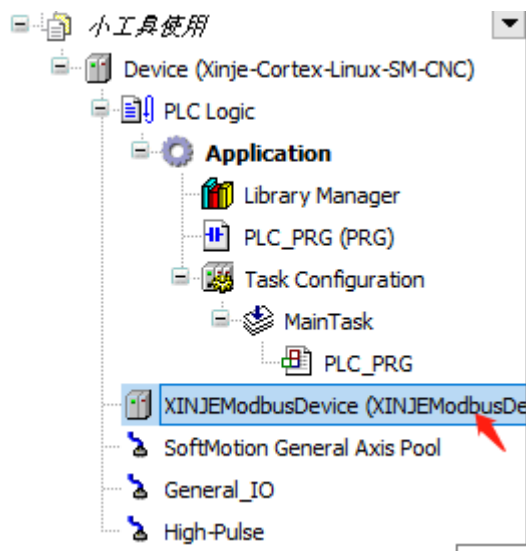
9. 附录

9. 附录	79
9-1. 小工具的使用.....	80

9-1. 小工具的使用

CodeSysPLC 小工具用以读、写 XS3 控制器 IP、读取 XS3 控制器版本信息以及连接模块型号及版本信息，具体使用步骤如下：

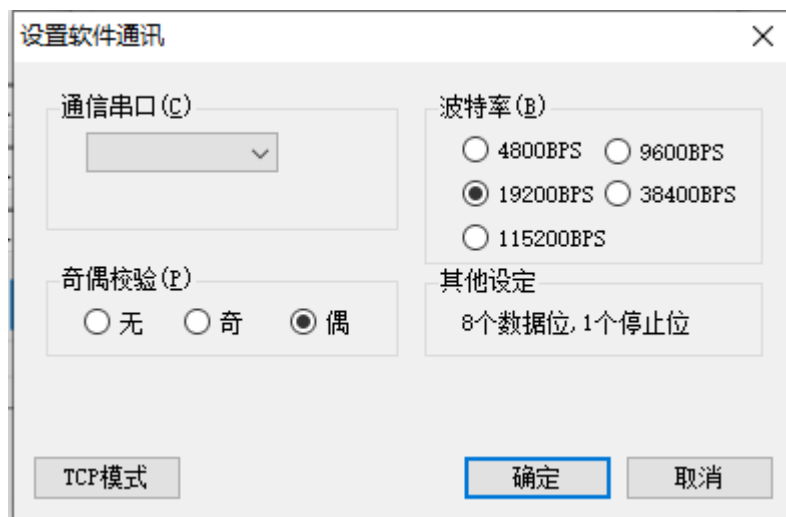
使用小工具读取、更改 IP 或者读取模块信息首先得添加 Modbus 设备库，在左侧设备窗口里找到 Device 右击 Add Device 添加 XINJEModbusDevice,添加之后设备窗口如下图所示：



之后下载程序运行后打开我们的 CodeSysPlc 小工具，界面如下图所示：



如需读取 PLC 和模块信息需要先进行通信配置，点击通信配置，小工具可以通过 Modbus RTU 和 Modbus TCP 两种模式连接 PLC，

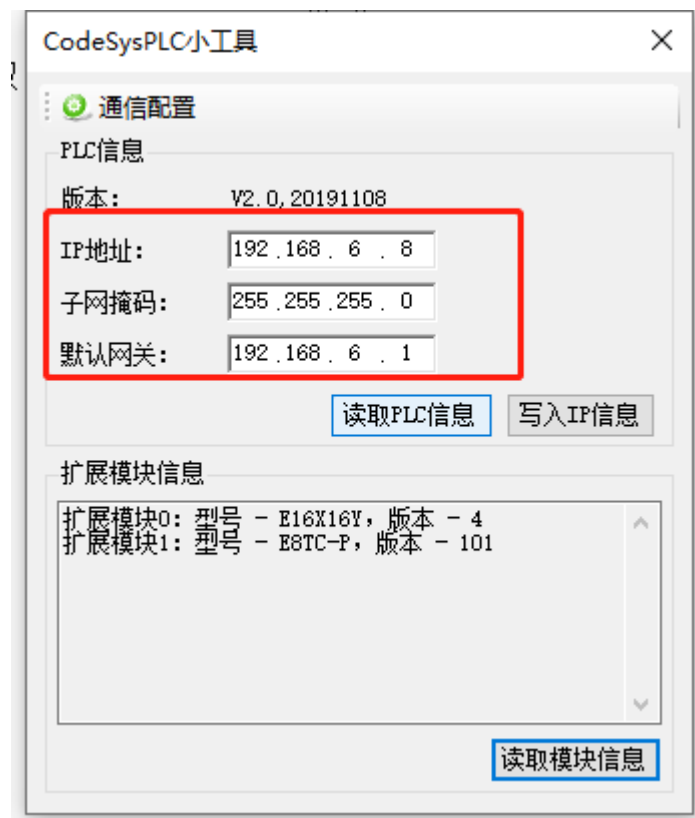


Modbus RTU 模式



Modbus TCP 模式

设置好之后点击确定就可读取 PLC 信息，读取模块信息



更改红色区域点击写入 IP 信息，就可修改 PLC 的 IP，断电上电之后生效。

注：写入 IP 后，不要立即断电，等待 2-3s 再断电。

XINJE



微信扫一扫，关注我们

无锡信捷电气股份有限公司

江苏省无锡市蠡园开发区滴翠路 100 号

创意产业园 7 号楼四楼

邮编：214072

电话：400-885-0136

传真：(0510) 85111290

网址：www.xinje.com

WUXI XINJE ELECTRIC CO., LTD.

4th Floor Building 7, Originality Industry park, Liyuan
Development Zone, Wuxi City, Jiangsu Province

214072

Tel: 400-885-0136

Fax: (510) 85111290