



XS 系列可编程控制器
用户手册【软件篇】(Codesys)

无锡信捷电气股份有限公司

资料编号 PS04 20211125 1.0

	目录	
	Codesys 概述与安装	1
	Codesys 结构	2
XS 系列可编程控制器 用户手册[软件篇] (Codesys)	基础指令	3
	特殊功能	4
	Codesys 工程举例	5
	工业现场总线技术	6
	常见问题及解决办法	7
	手册更新日志	

基本说明

- ◆ 感谢您购买了信捷 XS 系列可编程控制器。
- ◆ 本手册主要介绍 XS 系列可编程控制器的软件的相关使用内容。
- ◆ 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，进行编程。
- ◆ 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部门。
- ◆ 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。
- ◆ 请自行设置后备及安全功能，以避免因本产品故障而可能引发的机器故障或损失。

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。
- ◆ 手册中所叙述的内容如有变动，恕不另行通知。

关联手册

关于 XS 系列 PLC 的硬件相关、高级运动控制指令应用等内容，请查询以下手册。

手册下载网址：www.xinje.com。

- ◆ 《XS 系列可编程控制器用户手册【硬件篇】》
- ◆ 《XS 系列可编程控制器用户手册【高级运动控制篇】》

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。

二〇二一年 十一月

目 录

1. Codesys 概述与安装	1
1-1. Codesys 概述	2
1-2. Codesys 软件架构介绍	2
1-2-1. 开发层	2
1-2-2. 通信层	3
1-2-3. 设备层	3
1-3. Codesys 支持的信捷 PLC	3
1-4. Codesys 的安装与卸载	3
1-4-1. 系统配置要求	3
1-4-2. Codesys 获取	3
1-4-3. Codesys 安装	3
1-4-4. Codesys 版本管理	4
1-4-5. Codesys 卸载	4
1-5. Codesys 帮助	4
2. Codesys 结构	5
2-1. 软件模型	6
2-1-1. 软件模型简介	6
2-1-2. 软件模型的特点	7
2-2. 设备	7
2-2-1. 设备管理	7
2-2-2. 设备编辑器	9
2-3. 应用	10
2-3-1. 任务	10
2-3-2. 库文件	16
2-3-3. 访问路径	17
2-4. 程序组织单元	17
2-4-1. 程序单元结构	18
2-4-2. 函数	19
2-4-3. 功能块	21
2-4-4. 程序	22
2-5. 应用对象	23
2-5-1. 采样跟踪	23
2-5-2. 持续变量	23
2-5-3. 数据单元类型	24
2-5-4. 全局网络变量	24
2-5-5. 配方管理器	25
3. 基础指令	26
3-1. 位逻辑指令	27
3-1-1. 基本位逻辑指令	27
3-1-2. 置位优先与复位优先触发器指令	27
3-1-3. 数据单元类型	27
3-2. 定时器指令	28
3-3. 计数器指令	28
3-4. 数据处理指令	29
3-4-1. 选择操作指令	29

3-4-2. 比较指令	29
3-4-3. 移位指令	30
3-5. 运算指令	30
3-5-1. 赋值指令	30
3-5-2. 算数运算指令	30
3-5-3. 数学运算指令	31
3-5-4. 地址运算指令	31
3-5-5. 数据转换指令	32
4. 特殊功能	33
4-1. 高速计数	34
4-1-1. 功能概述	34
4-1-2. 功能块介绍	34
4-1-3. 参数配置	36
4-1-4. 应用举例	37
4-2. 外部中断	37
4-2-1. 功能概述	37
4-2-2. 应用举例	38
4-3. PLC SHELL	38
4-3-1. 功能概述	38
4-3-2. 指令列表	38
4-3-3. 应用举例	38
4-4. 时钟	43
4-4-1. 功能概述	43
4-4-2. 应用举例	43
5. Codesys 工程举例	45
5-1. 基本编程操作	46
5-2. I/O 映射	48
5-3. 任务配置	49
5-4. 程序下载/读取	51
5-4-1. 编译	51
5-4-2. 登录下载	52
5-4-3. 源代码下载	53
5-4-4. 读取程序	53
5-5. 程序调试	54
5-5-1. 复位	54
5-5-2. 程序调试	54
5-6. 仿真	56
5-7. PLC 脚本功能	56
6. 工业现场总线技术	57
6-1. MODBUS 通信	58
6-1-1. MODBUS 通信概述	58
6-1-2. 参数配置	58
6-2. MODBUS TCP	61
6-2-1. MODBUS TCP 通信概述	61
6-2-2. 参数配置	61
6-3. OPC UA	64
6-3-1. OPC UA 通信概述	64
6-3-2. 参数配置	64

7. 常见问题及解决方法.....	67
7-1. Package	68
7-1-1. Package 命名规则.....	68
7-1-2. Package 的获取.....	68
7-1-3. Package 的安装.....	68
7-2. XS 系列 PLC 固件升级.....	69
7-2-1. 固件命名规则	69
7-2-2. 固件获取	69
7-2-3. 固件安装及其注意事项	69
手册更新日志.....	70

1. Codesys 概述与安装

本章重点说明 XS 系列 PLC 上位机软件 Codesys 的安装系统要求、安装及卸载步骤。

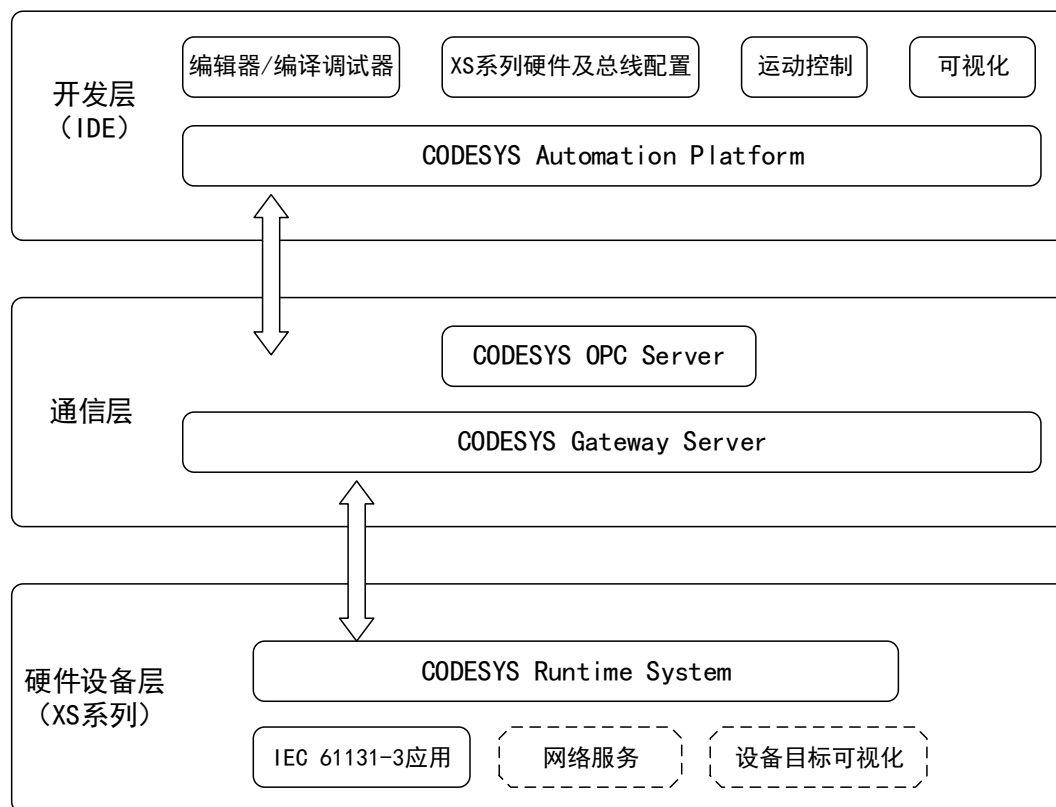
1. Codesys 概述与安装	1
1-1. Codesys 概述	2
1-2. Codesys 软件架构介绍	2
1-2-1. 开发层	2
1-2-2. 通信层	3
1-2-3. 设备层	3
1-3. Codesys 支持的信捷 PLC	3
1-4. Codesys 的安装与卸载	3
1-4-1. 系统配置要求	3
1-4-2. Codesys 获取	3
1-4-3. Codesys 安装	3
1-4-4. Codesys 版本管理	4
1-4-5. Codesys 卸载	4
1-5. Codesys 帮助	4

1-1. Codesys 概述

信捷 XS 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。

1-2. Codesys 软件架构介绍

Codesys 软件功能强大，可靠性高，开放性好，集成了 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。Codesys 软件从架构上可以分为 3 层：应用开发层、通信层和设备层，如图所示：



1-2-1. 开发层

Codesys Development System（具有完善的在线编程和离线编程功能）、编译器及其配件组件、可视化界面编程组件等，同时供用户可选的运动控制模块及安全模块等组件使 Codesys 功能更加完整和强大。

■ 编辑器

Codesys 提供了所有 IEC61131-3 所定义的六种编程语言：功能块图(FBD)、梯形图(LD)、指令表(IL)、结构化文本(ST)、顺序功能图(SFC)、连续功能图(CFC)。

■ 编译器

负责将 Codesys 中的应用程序转换为机器代码并优化可编程控制器的性能。当用户输入了错误的应用程序代码时，立刻会接收到编译器发出的语法错误警告及错误信息，让编程人员可以迅速做出相应纠正。用户不必改变编程方式，就可以使用不同的基于 Codesys 编程的硬件装置（系统）进行工程开发。

■ XS 系列硬件及总线配置

针对 XS 系列硬件设备及不同现场总线协议，该部分负责在 Codesys 中对相应参数进行设定。

■ 可视化界面编程

在 Codesys 中即可实现可视化编程（人机界面 HMI），系统已经集成了可视化编辑器。

■ 运动控制模块

运动控制功能已经集成在 Codesys 中，形成了 SoftMotion(CNC)软件包。基于 PLCopen 的工具包可以实现单轴、多轴运动，电子凸轮传动，电子齿轮传动，复杂多轴 CNC 控制等。

1-2-2. 通信层

应用开发层和硬件设备层之间的通讯是由 Codesys 中的网关服务器来实现的，Codesys 网关服务器中安装了 OPC 服务器。

■ Codesys 网关服务器

作用在应用开发层和硬件设备层之间，可以使用 TCP/IP 协议或通过 CAN 等总线实现远程访问，是 Codesys 开发工具包不可分割的一部分。

■ Codesys OPC 服务器

对基于 Codesys 进行编程的控制器，无需考虑所使用的硬件 CPU，已经集成并实现了 OPC V2.0 规范的多客户端功能，且能同时访问多个控制器。

1-2-3. 设备层

XS 系列 PLC 是系统的硬件设备层，已经安装好 Codesys 的实时核（Codesys Runtime System），能满足工业的实时响应和精确的控制需求。同时，也可以通过使用 Codesys 的可选组件：如 Codesys 目标可视化编程模块或网络可视化编程模块来实现功能上的扩展。

1-3. Codesys 支持的信捷 PLC

XSDH 系列、XS3 系列、M 系列以及视觉工控机。

1-4. Codeys 的安装与卸载

1-4-1. 系统配置要求

硬件及软件的要求：

- ◆ windows 8 或 windows 10 操作系统，推荐 64 位操作系统
- ◆ 内存在 4GB 及以上
- ◆ 硬盘空间 12GB 以上

1-4-2. Codesys 获取

在 Codesys 官方商店下载，网址为 <http://www.Codesys.cn/>。

1-4-3. Codesys 安装

1、硬件及软件的基本要求

由于 Codesys V3.5 软件比较大，处理信息也较多，对 PC 的硬件及软件有一定要求，其要求的最低配置及推荐配置如下表所示：

描述	最低配置	推荐配置
操作系统	Windows 2000 (Windows XP/Windows Vista/Windows 7)	Windows 10
内存	4GB	4GB
硬盘空间	12GB	12GB
处理器	Pentium V, Centrino > 1.8GHz, Pentium M > 1.0GHz	Pentium V, Centrino > 3.0GHz, Pentium M > 1.5GHz

2、安装

以管理员身份运行 Codesys 64 3.5.16.0.exe 安装文件即可进入安装，整个安装过程中安装助手都会引导用户进行安装。

注意：不建议用户将软件安装在 C 盘。

1-4-4. Codesys 版本管理

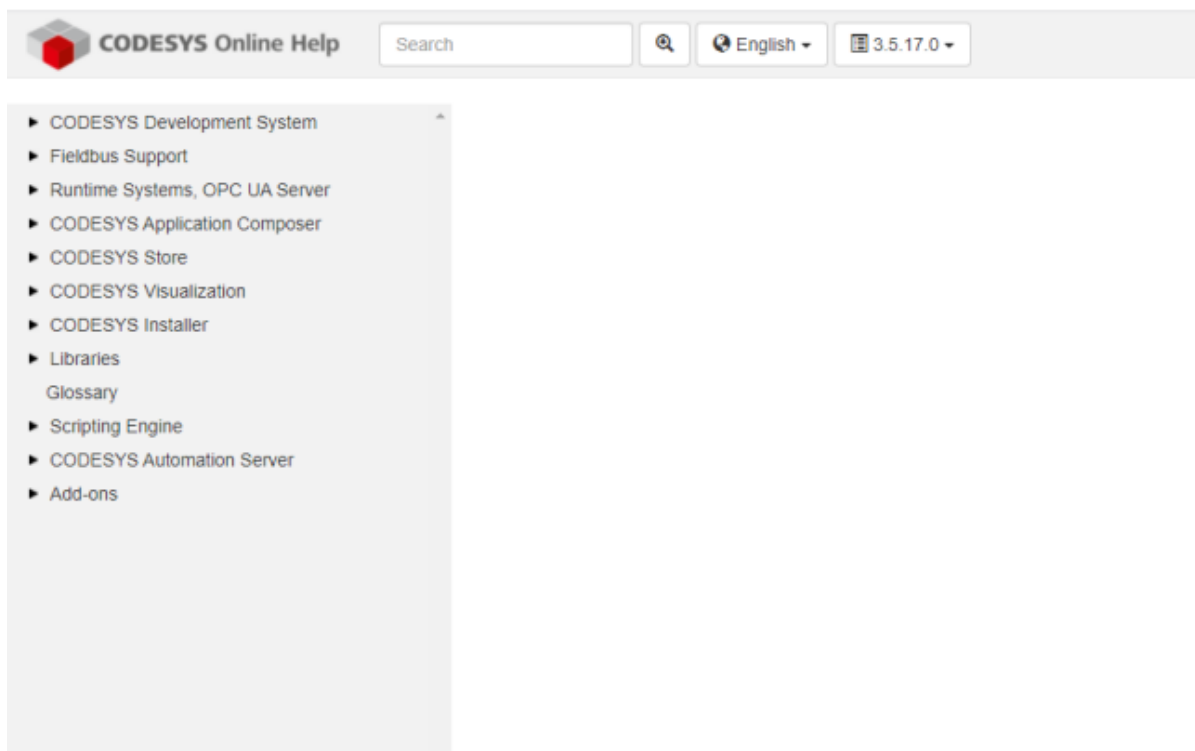
Codesys 上位机支持同时安装多个不同版本，编译器也支持安装使用多个版本，推荐使用版本 3.5.16.40 版本，使用其它版本上位机可能会出现某些功能使用异常的情况。

1-4-5. Codesys 卸载

Codesys 编程软件的卸载可以通过 Windows 的控制面板的添加和删除程序来完成。打开“控制面板”-->“添加/删除程序”，选中“Codesys”，单击删除按钮，根据提示完成卸载。

1-5. Codesys 帮助

用户在打开 Codesys 应用程序后，可以找到帮助（Help）菜单，点击“目录”即可打开在线帮助。用户可以根据索引或者搜索关键字快速找到所需要的内容，如图所示：



2. Codesys 结构

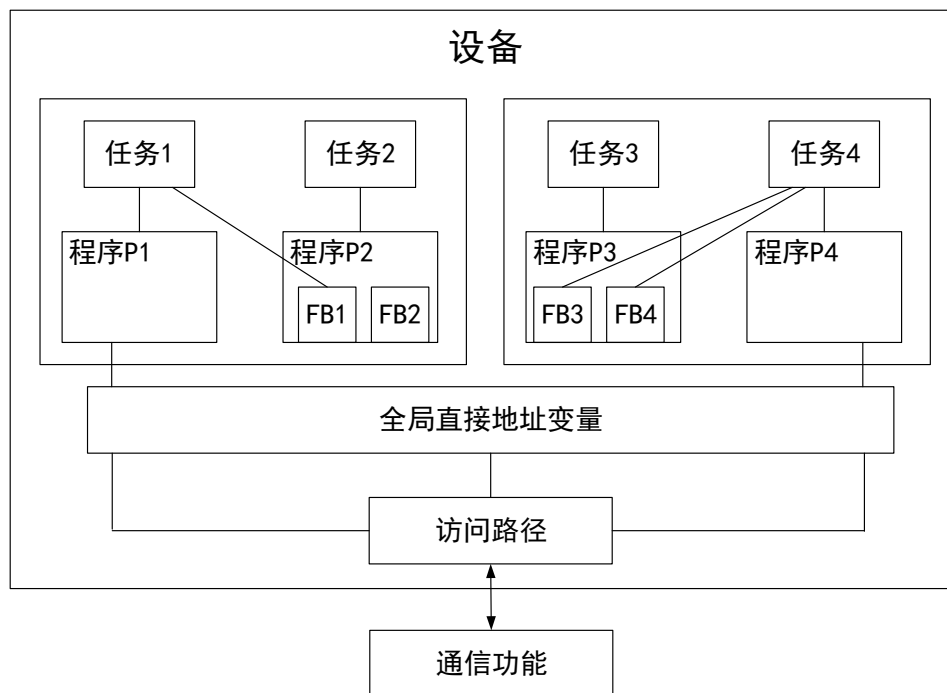
本章主要介绍 PLCopen、IEC61131-3 以及 Codesys 编程平台的基本概念。

2. Codesys 结构	5
2-1. 软件模型	6
2-1-1. 软件模型简介	6
2-1-2. 软件模型的特点	7
2-2. 设备	7
2-2-1. 设备管理	7
2-2-2. 设备编辑器	9
2-3. 应用	10
2-3-1. 任务	10
2-3-2. 库文件	16
2-3-3. 访问路径	17
2-4. 程序组织单元	17
2-4-1. 程序单元结构	18
2-4-2. 函数	19
2-4-3. 功能块	21
2-4-4. 程序	22
2-5. 应用对象	23
2-5-1. 采样跟踪	23
2-5-2. 持续变量	23
2-5-3. 数据单元类型	24
2-5-4. 全局网络变量	24
2-5-5. 配方管理器	25

2-1. 软件模型

2-1-1. 软件模型简介

Codesys 的软件模型描述基本的软件元素及其相互关系，采用分层结构表示，每一层隐含其下面层的许多特性，其内部结构如下图所示。其中，软件元素包含：设备、应用、任务、全局变量、访问路径和应用对象，他们是现代软 PLC 的软件基础，该软件模型与 IEC 61131-3 标准的软件模型保持一致。



该软件模型从原理上描述了如何将一个复杂程序分解为若干小的可管理部分，并在各分解部分之间有清晰和规范的接口方法。软件模型描述了一台可编程控制器如何实现多个独立程序的同时运行，如何实现程序执行的完全控制等。

■ 设备

在模型的最上层是“设备”，“设备”可以等效于一个 PLC 所需的所有软件。针对大型复杂的应用系统，如整个产品线的自动化，可能需要多个 PLC 联机通讯，需要将一个 PLC 与其他多个设备接口实现总线通信。这时，可以将“设备”理解为一个特定类型的控制系统，它包括硬件装置、处理资源、I/O 地址映射和系统内存存储的能力，即等同于一个 PLC。

■ 应用

在 PLC 系统中，设备将所有“应用”结合成组，为“应用”提供数据交换的手段。在每一个设备中，有一个或多个“应用”，应用位于软件模型的第二层。“应用”不仅为运行程序提供了一个支持系统，而且它反映了 PLC 的物理结构，在程序和 PLC 物理 I/O 通道之间提供了一个接口。

应用被分配在一个 PLC 的 CPU 中，因此，可将应用理解为一个 PLC 中的微处理器单元。在应用内定义的全局变量在该应用内部是有效的。应用的主要成员包括全局变量、任务和程序组织单元 (POU) 等。

■ 访问路径

访问路径的主要功能是将全局变量、直接表示变量和程序组织单元的输入/输出变量联系起来，实现信息的存储。它提供在不同应用之间交换数据和信息的方法，每一个应用内的变量可通过其他远程配置来存取。

■ 通讯功能

提供与其他系统、如其他可编程控制器系统、机器人控制器、计算机等装置的通讯，用于实现程序传

输、数据文件传输、监视、诊断等。通常采用符合国际标准的通讯方式（如 RS232、RS485）或工业现场总线如 CANopen、Profibus、EtherCAT、Modbus、Ethernet/IP、DeviceNet 等。

2-1-2. 软件模型的特点

Codesys 软件模型具有特点包括：

- ◆ Codesys 软件模型能够在一台 PLC 中能同时装载、启动和执行多个独立的程序。
- ◆ Codesys 软件模型能够实现对程序执行的完全控制能力。通过标准的任务机制，保证了 PLC 系统对程序执行的完全控制能力。传统 PLC 程序只能顺序扫描执行程序，对某一段程序不能按用户的实际要求定时执行，而该软件模型中任务的机制允许程序的不同部分在不同的时间、以不同的比率并行执行，这大大地扩大了 PLC 的应用范围。
- ◆ Codesys 软件模型是一个国际标准的软件模型，能够适应不同的 PLC 结构。其不是只针对具体的 PLC 系统，而是具有很强的适用性。既能适合小型的 PLC 系统，也可适合较大的分散系统。
- ◆ Codesys 软件模型支持程序组织单元的重用特性：软件的重用性是 Codesys 的重要优点。
- ◆ Codesys 软件模型支持分层设计：一个复杂的软件通常可以通过一层层的分解，最终分解为可管理的程序单元。

2-2. 设备

设备代表了一个具体的目标，即硬件对象，位于 Codesys 软件模型的最上层。该硬件对象可以是控制器，现场总线站点，总线耦合器，驱动器，输入/输出模块或是触摸屏等。每一个设备由一个“设备描述”文件定义，该设备描述文件安装在 Codesys 本机系统中，以供插入到设备树下（这里用“设备树”表示设备窗口中的树状列表）。该设备描述文件确定了设备的相关配置、可编程性、和其他设备的互联性。设备是结构元素，它位于软件模型的最上层，在软件内部是大型的语言元素。

2-2-1. 设备管理

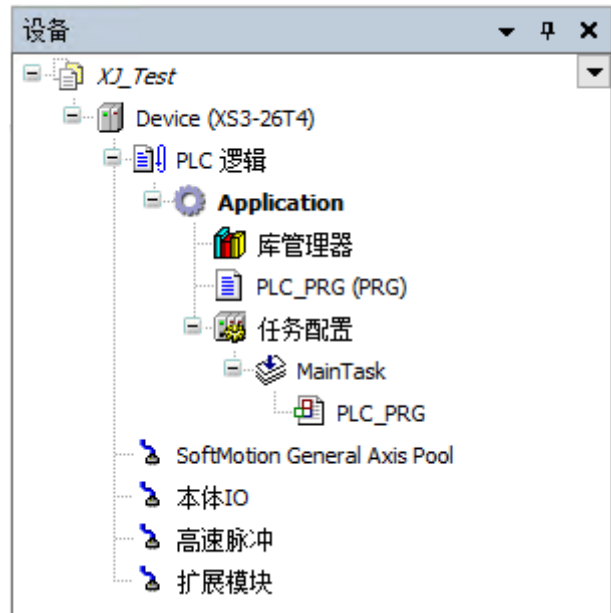
对设备的管理包括对设备的添加，对安装包的管理和对设备库的管理等。

1、添加设备

新建一个工程时，系统会自动弹出对话框，如下图所示。用户可以在模板选项中选择新建一个空工程或标准工程，在选择标准工程时，需要用户选择实际连接的硬件设备。



点击“确定”即可得到如下图所示的设备树。



2、包管理器

所有的“设备”必须事先在“包管理器”中进行安装，包管理器在“工具”菜单中可以选择，用户可以对其进行添加或删除包。

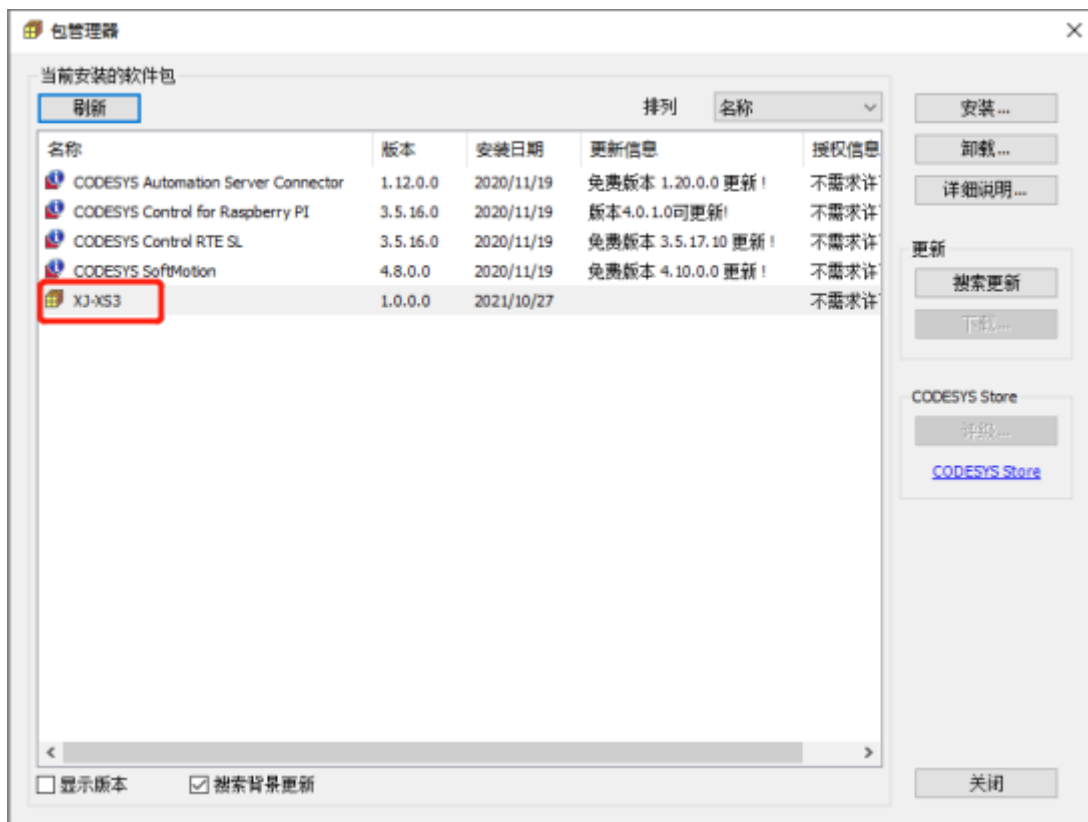
针对不同的硬件设备，需要不同的硬件配置参数，其中必须配置的参数有：代码生成器，内存管理，PLC 功能，I/O 模块配置，另外，必须链接库，网关驱动程序以及用于错误消息的 ini-files 和 PLC 浏览器的相关信息。此外包中整合了特殊功能，包含了相应的库文件，设备描述文件等。

对于本产品的包管理器安装流程如下：

打开“工具”，选择“包管理器”；

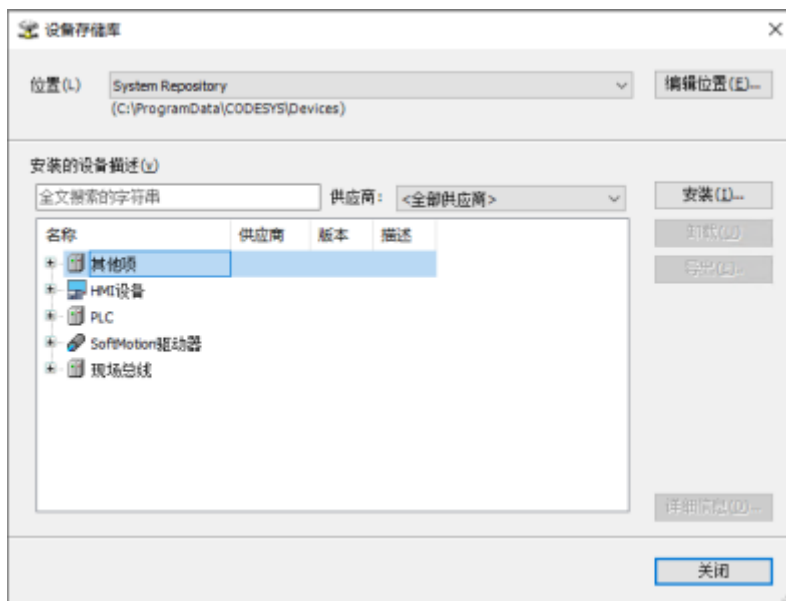
点击“安装”，在目录中找到对应的安装包，本例采用 XJ_XS3.package；

点击“确定”，安装成功，在包管理器中会显示出“XJ_XS3”图标，如下图所示：



3、设备库管理

设备库是当用户添加或删除硬件设备信息时所需要做的操作，设备库是设备的数据库，安装后的所有数据被导入至用户本地系统中，供 Codesys 开发使用。设备库对话框如下图所示：



设备库可用于添加所有的硬件设备，通过在该选项导入相应文件后，使对应数据在本地系统内生成，方便工程中调用。可添加的设备有供应商的 PLC，SoftMotion 运动控制设备（编码器，驱动器等）、现场总线及专用接口等设备。

本产品可添加的设备描述文件包括信捷官方提供的本体和扩展模块的设备描述文件、EtherCAT 的 XML 文件、CANopen 的 EDS 和 DCF 文件、IO-Link 的 IODD 和 Profibus DP 的 GSD 文件等。

2-2-2. 设备编辑器

设备编辑器是用于配置设备的对话框。通过选中设备图标，通过鼠标右键“编辑对象”命令，或者通过在设备窗口中双击设备对象条目打开。

主对话框是根据设备类型，以设备名称来命名的，本产品提供了包含以下子对话框的选项卡，如下表所示：

通讯设置	目标设备和其他可编程设备（PLC）之间连接的相关配置
应用	分别显示设备参数的配置
备份与还原	在 PLC 上备份特定于应用程序的文件
文件	主机和 PLC 之间的文件传输的配置
日志	显示 PLC 的日志文件
PLC 设置	与 I/O 操作相关的应用、停止状态下的 I/O 状态、总线周期选项的配置
PLC 指令	通过 shell 指令可以对 PLC 进行系统配置
用户和组	运行中设备访问相关的用户管理（不要与工程用户管理混同）
访问权限	特殊用户组对运行中的对象和文件访问权限的配置
IEC 对象	通过 IEC 应用程序访问设备的“对象”
时钟 I/O 映射	提供实时时钟
任务部署	显示输入和输出表以及它们对已定义任务的分配
状态	设备的详细状态和诊断信息
信息	设备的基础信息（名称、供应商、版本、序列号等）

2-3. 应用

应用是指在硬件设备（如 PLC）上运行程序时所需要的对象集合。这些对象与硬件设备平台无关，用户可以在程序组织单元（POU）中管理它们。然后在设备窗口中将它们实例化，分配到具体的设备中。这种方法符合面向对象编程的思想。

应用对象包括任务，程序组织单元、任务配置、全局变量、库管理器和采样追踪等。在 Codesys V3.x 中资源对象只能在设备树中进行管理。在设备树中添加对象后，需要按一定的“规则”与被控设备进行映射。对象（如库和全局变量列表等）在工程中的有效范围，会依据设备树中应用和设备对象的层级关系而定，一般来说，一个应用中的对象对其“子应用”也有效，可以被使用。

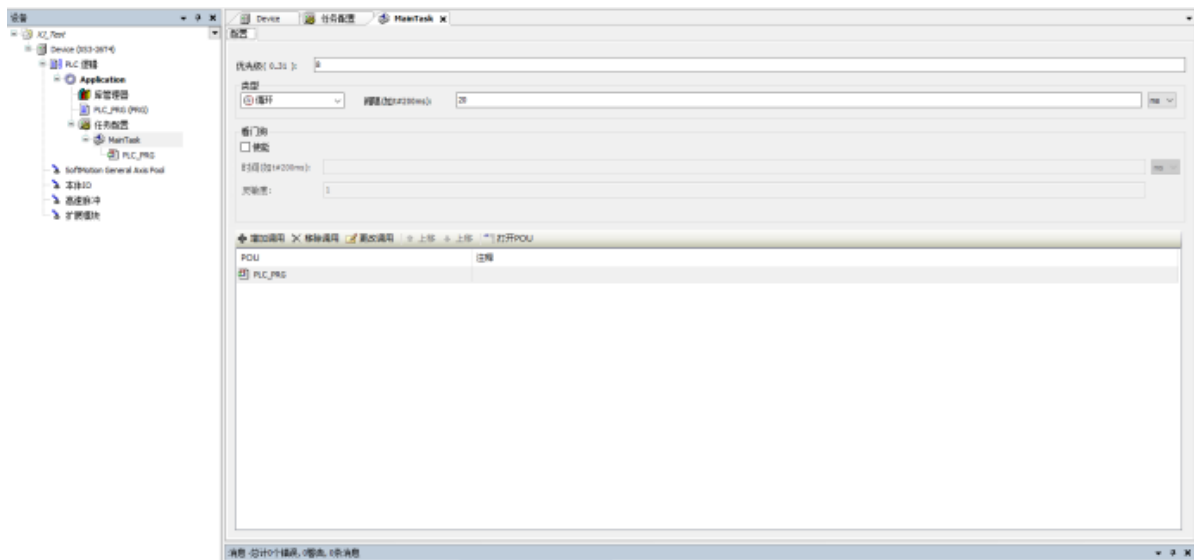
2-3-1. 任务

1、概述

一个程序可以用不同的编程语言来编写。典型的程序由许多互连的功能块组成，各功能块之间可互相交换数据。在一个程序中不同部分的执行通过“任务”来控制。“任务”被配置以后，可以使一系列程序或功能块周期性地执行或由一个特定的事件触发开始执行程序。

在设备树中有“任务管理器”选项卡，使用它除了声明特定的 PLC_PRG 程序外，还可以控制工程内其他子程序的执行处理。任务是用于规定程序组织单元在运行时的属性，它是一个执行控制元素，具有调用的能力。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，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。

在任务配置中，用名称、优先级和任务的启动类型来定义它。这启动类型可以通过时间（周期的，随机的）或通过内部或外部的触发任务时间来定义，例如使用一个布尔型全局变量的上升沿或系统中的某一特定事件。对每个任务，可以设定一串由任务启动的程序。如果在当前周期内执行此任务，那么这些程序会在一个周期的长度内被处理。优先权和条件的结合将决定任务执行的时序，任务设置界面如下图所示：

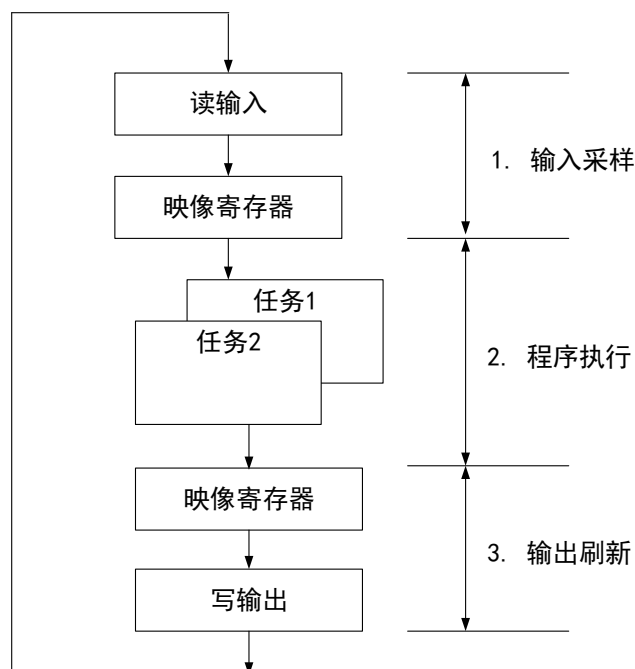


用户在配置任务时，需遵循如下规则：

- (1) 循环任务的最大数为 100；
- (2) 自由运行任务的最大数为 100；
- (3) 事件触发任务的最大数为 100；
- (4) 根据目标系统，PLC_PRG 可能会在任何情况下作为一个自由程序执行，而不用插入任务配置中；
- (5) 处理和调用程序是根据任务编辑器内自上而下的顺序所执行的。

2、PLC 程序执行过程

下图详细地描述了在 PLC 内部执行程序的完整流程，主要有输入采样，程序执行和输出刷新这三个重要的步骤组成。



(1) 输入采样

每次扫描周期开始时，PLC 检测输入设备（开关、按钮等）的状态，将状态写入输入映像寄存区内。在程序执行阶段，运行系统从输入映像区内读取数据进行程序解算。输入映像区的刷新只发生在一个扫描开始阶段，在扫描过程中，即使输出状态改变，输入状态也不会发生变化。

(2) 执行程序

在扫描周期的执行程序阶段，软 PLC 从输入映像区或输出映像区内读取状态和数据，并依照指令进行逻辑运算和算术运算，运算的结果保存在输出映像区相应的单元中。在这一阶段，只有输入映像寄存器的内容保持不变，其他映像寄存器的内容会随着程序的执行而变化。

(3) 输出刷新

输出刷新阶段亦称为写输出阶段，PLC 将输出映像区的状态和数据传送到输出点上，并通过一定的方式隔离和功率放大，驱动外部负载。

PLC 在一个扫描周期内除了完成上述三个阶段的任务外，还要完成内部诊断、通信、公共处理以及输入/输出服务等辅助任务。

由 PLC 的扫描方式可得知，PLC 为了迅速相应输入输出数据的变化，完成控制任务，扫描时间较短，PLC 的扫描时间一般都控制在 ms 数量级，因此需要开发稳定、可靠、响应快的实时系统供 PLC 运行系统所用。

PLC 重复执行上述 (1) 至 (3) 的过程，每重复一次的时间就是一个工作周期（或扫描周期）。

从 PLC 工作过程中可以了解，由于 PLC 采用循环的工作方式，输入信号只会在每个周期的开始阶段进行刷新，输出在每个工作周期的结束阶段进行集中输出，所以必然会产生输出信号相对输入信号的滞后现象。

从 PLC 的输入端有一个信号输入发生到变化到 PLC 的输出端对该输入信号的变化做出反应需要一段时间。滞后时间是设计 PLC 控制系统时应了解的一个重要参数。

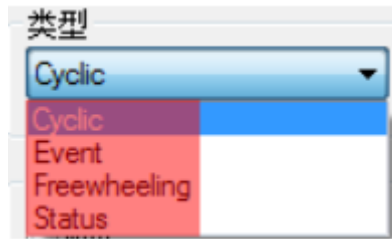
滞后时间的长短和以下因素有关：

- ◆ 输入电路的滤波时间，它由 RC 滤波电路的时间常数决定。改变时间常数可调整输入延迟时间。
- ◆ 输出电路的滞后时间，与输出电路的方式有关系，继电器输出方式的滞后时间一般为 10ms 左右，晶体管输出方式滞后时间小于 1ms。
- ◆ PLC 循环扫描的工作方式。
- ◆ 用户程序中语句的安排。

3、任务的执行类型

在任务配置树的最顶端有条目“任务配置”。下面是当前定义的任务，每个通过任务名代表。特定任务的 POU 调用没有显示在任务配置树中。

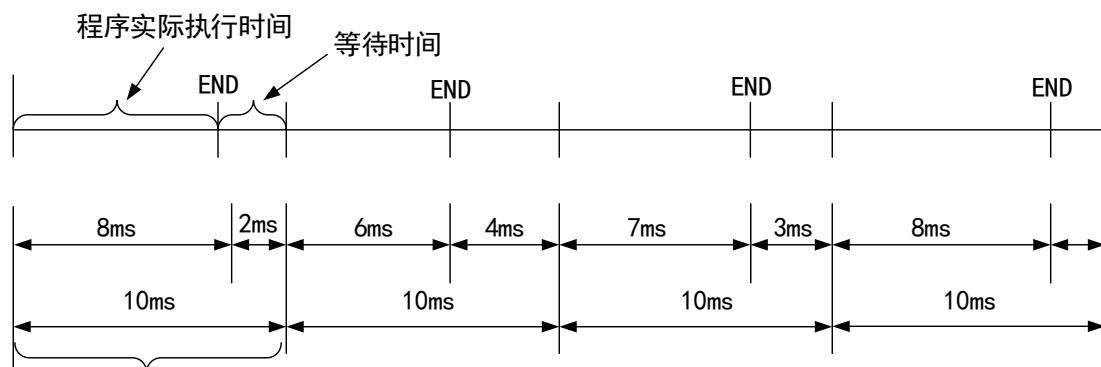
针对每个独立的任务可以对其进行执行的类型编辑及配置。包括固定周期循环、事件触发、自由运行和状态触发 4 种类型。详见下图所示。



(1) 固定周期循环

根据程序中所使用的指令执行与否，程序的处理时间会有所不同，所以实际执行时间在每个扫描周期都发生不同的变化，执行时间有长有短。通过使用固定周期循环方式，能保持一定的循环时间反复执行程序。即使程序的执行时间发生变化，也可以保持一定的刷新间隔时间。在这里，也推荐大家优先选择固定周期循环任务启动方式。

例如，假设将程序对应的任务设定为固定周期循环方式，间隔时间设定为 10ms 时，实际程序执行的时序图如下图所示：

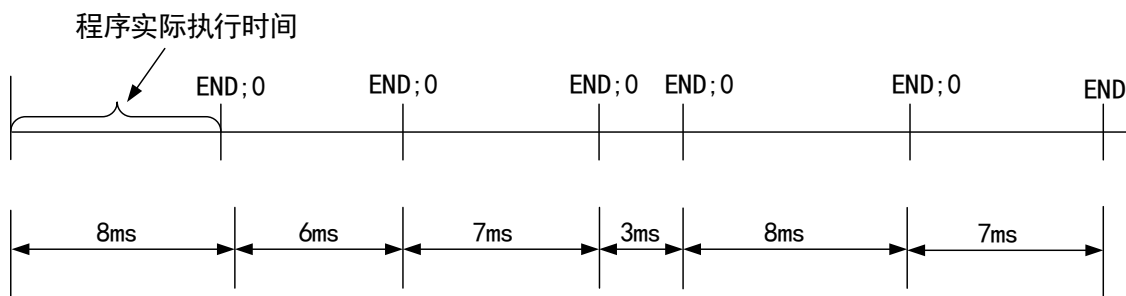


固定周期循环设定时间

如果程序实际执行时间在规定的固定周期循环设定时间内执行完，则空余时间用作等待。如应用中还有优先级较低的任务未被执行，则剩下的等待时间用来执行低优先级的任务。详见任务优先级的说明。

(2) 自由运行

程序一开始运行任务就被处理，一个运行周期结束后任务将在下一个循环中被自动重新启动，如下图所示。不受程序扫描周期（间隔时间）的影响。即确保每次执行完程序的最后一条指令后才进入下一个循环周期。否则不会结束该程序周期。



该执行方式因为没有固定的任务时间，所以每次执行的时间可能都不一样。故不能保证程序的实时性，在实际的应用中选用此方式的场合较少。

(3) 事件触发

如果事件区域的变量得到一个上升沿，任务开始。

(4) 状态触发

如果事件区域的变量为 TRUE，任务开始。

状态触发方式与事件触发功能类似，区别在于状态触发的触发变量只要为 TRUE 程序就执行，为 FALSE 则不执行。而事件触发只采集触发变量的上升沿有效信号。

下图针对事件触发和状态触发分别进行了比较，绿色实线为两种触发方式选择的布尔变量状态，下表为比较的结果。



在采样点 1-4（紫色）不同类型的任务展示了不同的反应。这个具体的事件为 TRUE 完成了状态驱动任务的条件，然而一个事件驱动任务需要事件从 FALSE 变为 TRUE。如果任务计划的采样频率过低，事件的上升沿可能检测不到。

执行点	1	2	3	4
事件触发 (Event)	不执行	执行	执行	执行
状态触发 (Status)	不执行	执行	不执行	不执行

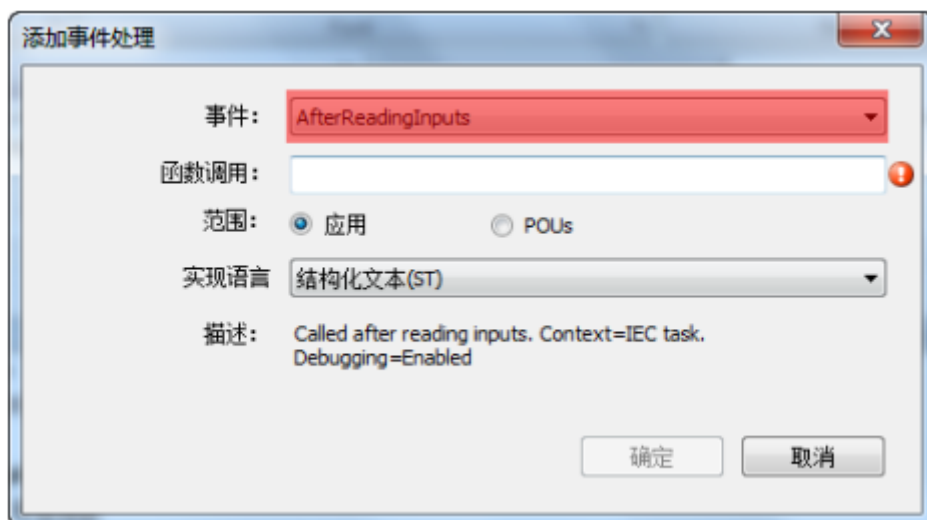
4、系统事件

用户可选择的系统事件是根据实际的目标系统而定的，由目标系统对应的库文件提供相应的系统事件，所以不同的目标硬件设备对应的系统事件可能会不同。常用的系统事件有：停止、开始、登入、改变等。在任务配置中，可以对任务配置中的系统事件进行设置。

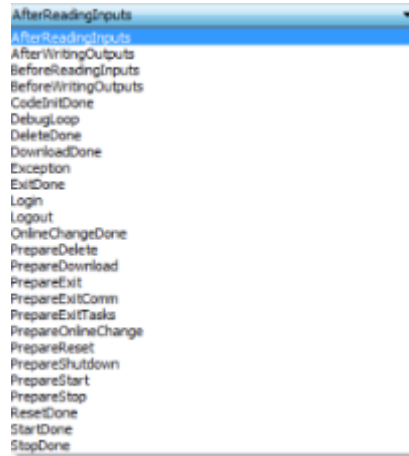
用户可以通过鼠标选择“任务配置”-->“系统事件”进入下图显示的界面。



选择“添加事件处理”按钮可以进行添加系统事件。打开后的界面如下图所示。



可选择的“事件”类型如下图所示。必须在“函数调用”这里必须新建一个函数名，而不能使用 POU 中已经存在的函数。“实现语言”为对应函数的编程语言。设置完点击“确定”。



5、任务优先级

Codesys 软件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。

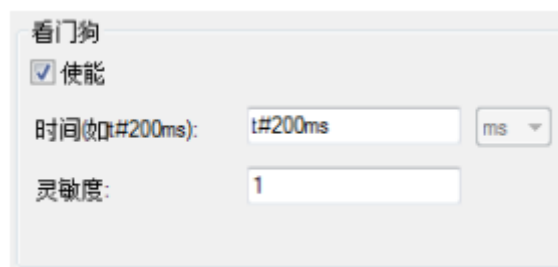
如果任务的类型为“Cyclic”，则按照“间隔”中的时间循环执行，具体设置如下图所示。



6、看门狗

看门狗是一种控制器硬件式的计时设备，Codesys 内可以通过“任务配置”对其进行使能，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。我们可以形象的将它理解为一只小狗需要主人定时的去给它喂食，如果超过规定的时间没有喂，则他马上就会饿。要配置看门狗，必须定义两个参数，时间和灵敏度，看门狗的配置如下图所示。



1) 时间

Codesys 针对每个任务可以配置独立的看门狗。如果目标硬件支持长看门狗时间设置，则可以设置上限和下限。默认的看门狗时间单位为毫秒（ms）。如果程序执行周期超过看门狗触发时间，将激活看门狗功能，并将中止当前任务。

2) 灵敏度

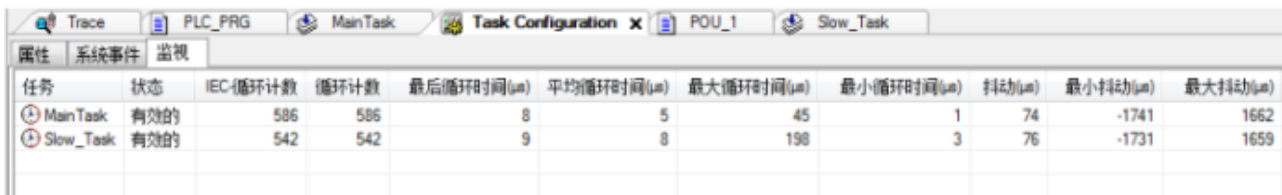
“灵敏度”用于定义必须在控制器检测到应用程序错误之前发生的任务看门狗例外数。默认为 1，可参照下表。

灵敏度	超过设定时间的倍数
0,1	1
2	2
.....
n	n

最终的看门狗触发时间=时间×灵敏度。如果程序实际执行时间超过看门狗触发时间，则激活看门狗。例如，时间为 10ms，灵明度设为 5，则看门狗触发时间为 50ms，只要任务的执行时间超过 50ms，则立即激活看门狗并将任务中止。

7、任务运行状态监视

每个任务可以直接启用或停用，系统会自动配置一个任务监视器，当进入在线模式后，用户可以使用系统自带的监视器对任务的平均/最大/最小循环时间等任务执行相关参数进行监控。如下图所示：



在项目初期阶段，可以使用该功能测试程序最大/最小/平均循环时间，用于测定程序的稳定性及对程序设定任务周期时间的优化。监视窗口中每个参数的具体定义见下表：

参数名	描述	参数名	描述
任务	在任务配置中所定义的任务名	平均循环时间 (µs)	任务平均所需的执行时间，单位为 µs
状态	分别有如下状态： 未创建：程序下载后一致未被建立，当使用时间触发任务可能会出现此状态； 创建：任务已经在实时系统中建立，但还未正式运行； 有效：任务正在被执行； 异常：任务出现异常。	最大/最小循环时间 (µs)	任务最大/最小执行时间，单位为 µs
IEC 循环计数	程序自开始运行至今的循环累积计数。“0”表示不支持该目标系统。	抖动 (µs)	上个周期测量到的抖动值，单位为 µs
循环计数 (µs)	已经运行的周期计数。取决于目标系统，它可以等于 IEC 循环计数，或者更大值，此时即时应用程序没运行，周期也一样被计数。	最小/最大抖动 (µs)	测量得到的最大/最小抖动时间，单位为 µs
最后循环时间 (µs)	上一个周期的任务执行时间，单位为 µs		

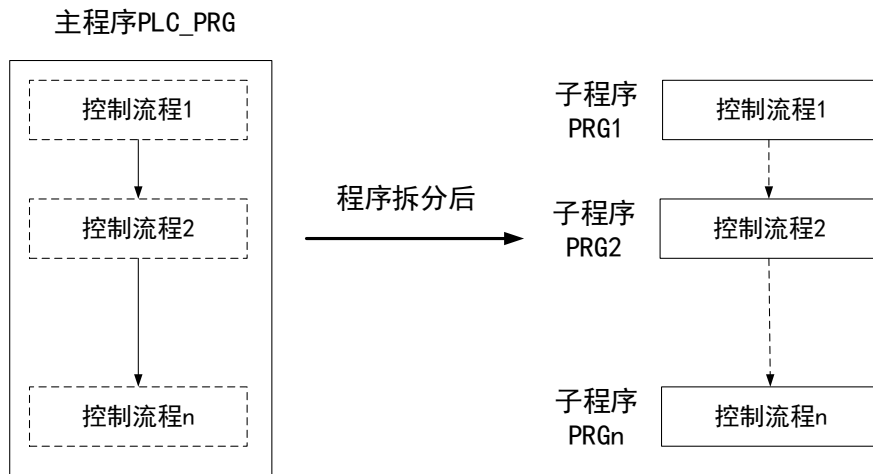
了解了如上各时间的定义后，应遵循如下的时间设定关系，按照此设定方法可以更好的优化程序任务周期及看门狗时间，保障程序的稳定性和程序的实时性。

看门狗触发时间>固定周期循环时间>程序最大循环时间

循环时间比固定周期循环时间长的情况下，CPU 会检测出程序有超出计数，此时，会影响程序的实时性。如果程序循环时间比看门狗时间设定长的情况下，CPU 会检测出看门狗故障，会停止程序的执行。

8、多子程序的运行

在实际的工程项目中，通常可以将程序按控制流程或者按照设备的对象分割成很多子程序，据此，设计人员将可以按各处理单元分别进行编程。如下图所示，以控制流程将主程序拆分为多个不同流程的子程序，拆分的目的是使主程序调理更清晰，并且方便今后的调试。



上图中右半部份是按流程进行分类的各子程序 PRG1, PRG2..PRGn, 图的左半部分为主程序 PLC_PRG, 在主程序中分别调用 PRG1..PRGn 的子程序。

多子程序运行的方式有两种, 第一种在任务配置中添加子程序。第二种方法是在主程序中调用子程序, 也是比较常用及灵活的一种方式。

2-3-2. 库文件

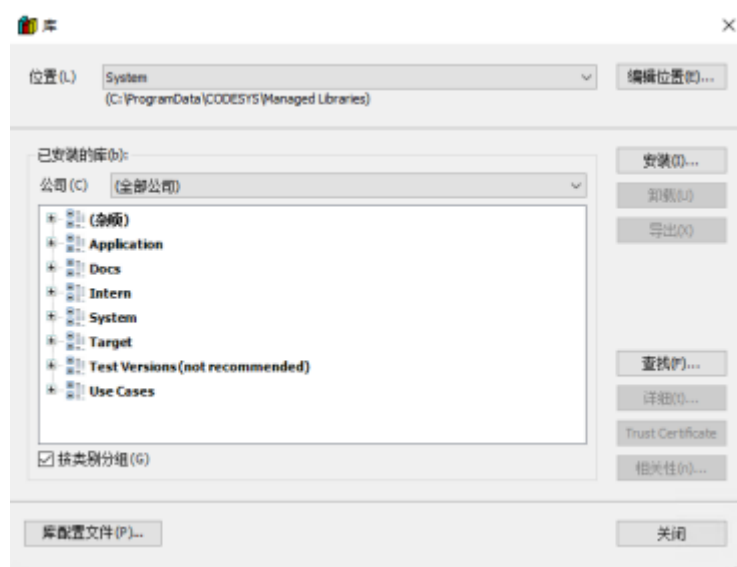
库文件用于存放 Codesys 中可多次使用的程序组织单元 (POU)。Codesys 提供基本库, 用户可以在基础库的基础上构造新库, 并通过加载的方式, 在程序中引用。

库文件是函数、功能块和程序的集合, 其中还包含一些特殊定义的结构体, 枚举类型等。从功能上分可以将库文件分为系统库文件、应用库文件以及厂家自定义库文件。其中, 系统库文件是一个支持 Codesys 软件系统的文件, 它包括对软件结构和语法编写的支持以及标准 I/O 的支持; 应用库文件是支持基本应用的文件库, 包括数据运算功能、定时器、计数器、边沿检测等; 厂商自定义库文件是根据不同厂商对产品规格不同特制的库文件。

1、库文件的管理

库管理器显示与当前项目有关的所有库。库的 POU、数据类型和全局变量, 都可以像用户定义的 POU、数据类。库管理器通过库管理器命令打开, 包括库在内的有关信息和项目一起进行保存。

如需安装计算机上的库文件或供应商所提供的库文件进行调用, 则需要使用到库文件管理。库文件管理是通过使用菜单命令“工具”-->“库”来定义的, 下图为库文件管理视图。



显示的库文件的类别包含应用类、通讯类、控制器类、设备类、系统类等。库文件的使用流程如下：

(1) 库文件的安装。在使用一个库文件之前，必须先在“库”对话框中对其进行“安装”。安装后，才可以在项目中调用该库。

(2) 库文件的调用。安装过库文件后，需要通过库管理器对库文件进行添加，以实现项目对库文件的调用。

2、库文件的属性

库文件需要实现访问的唯一性和安全性。

(1) 访问唯一性。在一个项目中如果有几个模块或变量具有同一个名字，那么访问具有相同名字变量的路径必须是不同的（即“唯一访问”），否则就会发生编译错误。该规则对本地工程、库、被其他库引用的库中的模块或变量都适用。用户可以通过在模块或变量名前加上名字空间来实现唯一访问。

(2) 访问安全性。Codesys 提供库文件加密功能实现对开发者库文件源代码的保护。用户通过在工程设置中将库文件添加许可信息，并保存成“已编译的函数库”，则在下次使用时，需要密码登录才能打开库文件，若密码错误，则不能使用及打开库文件，并触发日志报警。

2-3-3. 访问路径

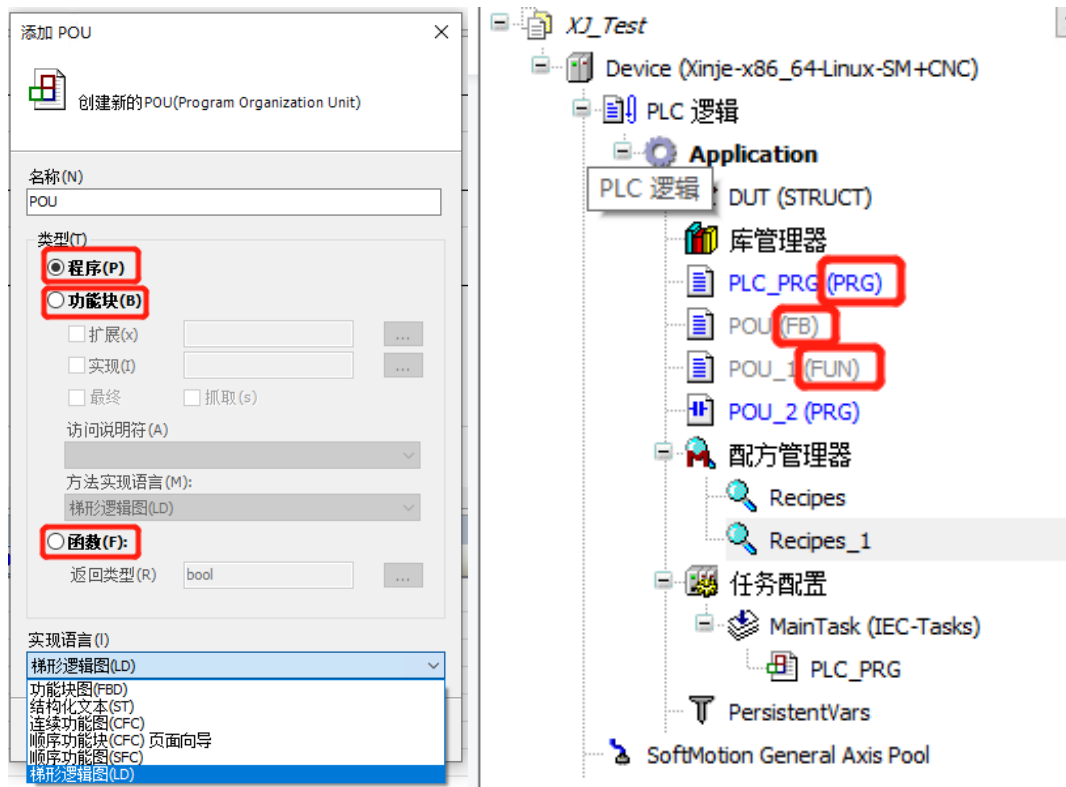
访问路径用于将全局变量、直接表示变量、功能块的输入/输出和局部变量联系起来，实现信息的存储。它提供在不同配置之间交换数据和信息的方法，每一个配置内的许多指定名称的变量可通过其他远程配置来存取。

访问路径功能已经集成在 Codesys 内部，用户不需要对其进行操作，所有的存取操作会在 Codesys 的后台自动进行。

2-4. 程序组织单元

程序组织单元（POU）是用户程序的最小程序单元，由声明区和代码区两部分组成，是全面理解新语言概念的基础，按功能分程序组织单元（POU）可分为函数（FUN）、功能块（FB）和程序（PRG）。

用户可以在项目中使用右键菜单的命令“添加对象”，选择“程序组织单元”，会弹出如图 2.x 所示的对话框，用户可以选择添加程序，功能块或函数，下拉菜单中可以选择对应的编程语言。添加后，可以在左边的项目设备树中查看程序组织单元括号内对应的属性，FB 为功能块，FUN 为函数，PRG 为程序

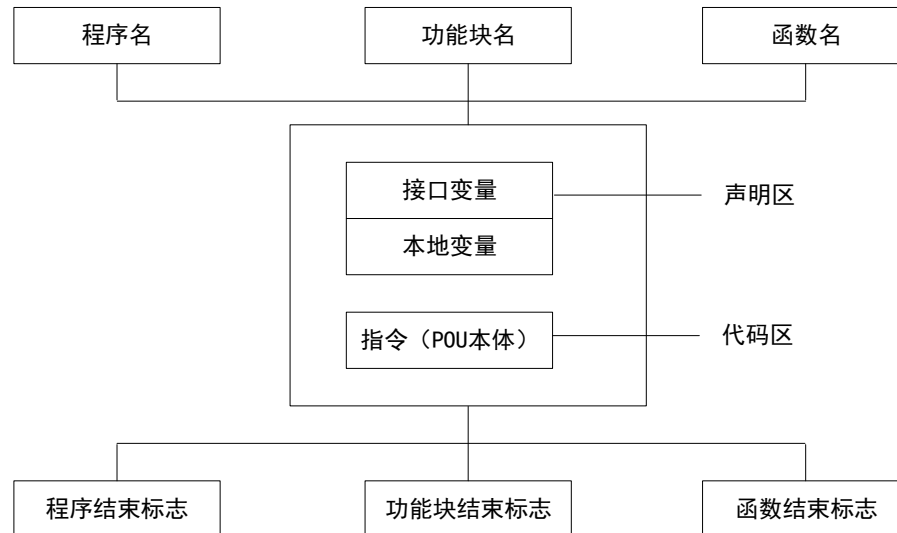


程序组织单元具有如下特点：

- ◆ 可对每个应用领域设置用户的功能块库，便于工程的应用。例如，建立运动控制功能块库等；
- ◆ 可对功能块进行测试和记录；
- ◆ 能够提供全局范围内的库存取功能；
- ◆ 可重复使用，使用的次数无限制；
- ◆ 可改变编程，用于建立功能块网。

2-4-1. 程序单元结构

一个完整的 POU 由 POU 类型及命名、变量声明部分和代码指令部分（POU 主体）等三大部分组成，结构图如下所示：



上图中，从具体功能来看，分别能构成左边的程序（PRG），中间功能块（FB），右边函数（FUN）。从每个功能的结构来看，都可以将其分为声明部分和代码部分。

用户声明的所有变量最终是给程序组织单元所用，变量声明中可声明接口变量和本地变量。

1、声明区

变量声明区是用来指定变量的名称、类型和赋初始值的区域。

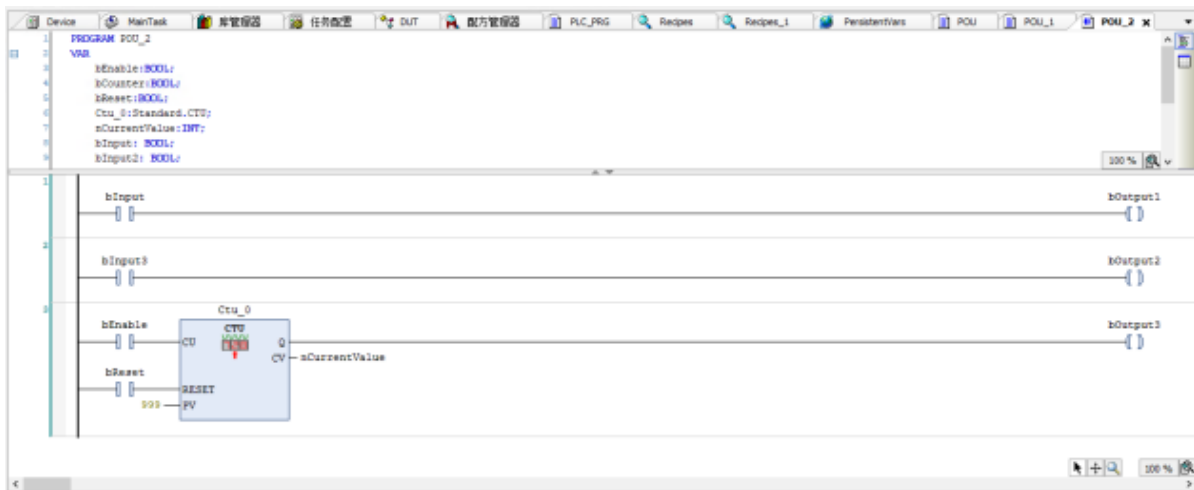
变量声明编辑器用来声明 POU 变量以及声明数据类型。声明部分通常是文本编辑器，也可采用表格编辑器。所有将要在这个 POU 中的使用的变量则在 POU 的声明部分中声明，这些变量包括：输入变量、输出变量、输入/输出变量、本地变量、添加的变量和常量。声明格式是基于 IEC61131-3 标准的。变量的声明采用下面的格式：

<标识符>{AT<Address>}: <数据类型>{: =<初始化>}:

其中{}中的部分是可选择的。

2、代码区

在代码区，Codesys 支持两种文本语言：指令表语言（IL）和结构化文本（ST）。四种图形化语言：功能块图（FBD）、梯形图（LD）、顺序功能图（SFC）以及连续功能图（SFC）。用户可以选择一种或几种语言在主体部分来进行程序设计。主体编辑器界面如下图所示，该图中采用的是梯形图（LD）程序语言。



2-4-2. 函数

为了 PLC 编程语言的应用，函数（FUN）也被定义为一个程序组织单元。函数是一种可以赋予参数，但没有静态变量的程序组织单元。即用相同的输入参数调用某一函数时，该函数总能生成相同的结果作为函数值（返回值）。函数的一个重要特性是它们不能使用内部变量存储数值，这点与功能块完全不同。

函数（FUN）是没有内部状态（没有运行时的内存分配）的基本算法单元。也就是说只要给定相同的输入参数，调用函数必定得到相同的运算结果，绝对没有二义性。我们平时使用的各种数学运算函数，如 $\sin(x)$ 、 \sqrt{x} 等，就是典型的函数类型。

函数是有至少一个输入变量、无私有数据、仅有一个返回值的基本算法单元。

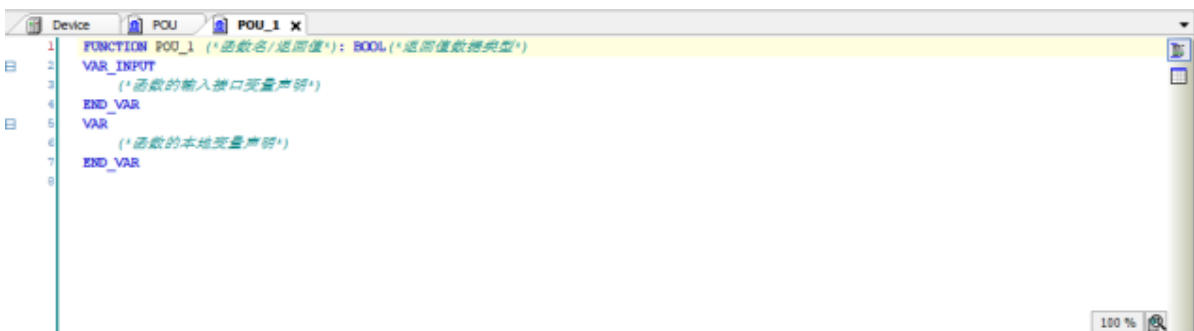
Codesys 的标准库中已经预有标准函数。

函数可以被函数、功能块、程序所使用。

1、函数的表示和声明

1) 自定义函数的表示

函数内部逻辑部分可以使用 6 种编程语言中的任意一种。函数名即是函数的返回值，也可以理解为是函数的输出值，如下图所示：



2) 函数中变量的声明

用户自定义函数时，应注意如下事项：

- ◆ 函数可以拥有很多个输入变量，但只能有一个返回值（输出变量）。但是并没有限制返回值的数据类型，所以可以为一个结构体作为返回值。
- ◆ 函数的重要特征是它们不能在内部变量存储数值，这点与功能块截然不同。
- ◆ 函数没有指定的内存分配，不需要像功能块一样进行实例化。
- ◆ 函数只能调用函数，不能调用功能块。
- ◆ 配置到 VAR_INPUT 的自变量可以是空的、常数、变量或函数调用，在函数调用时，函数是作为实际的自变量被调用的。

2、标准函数

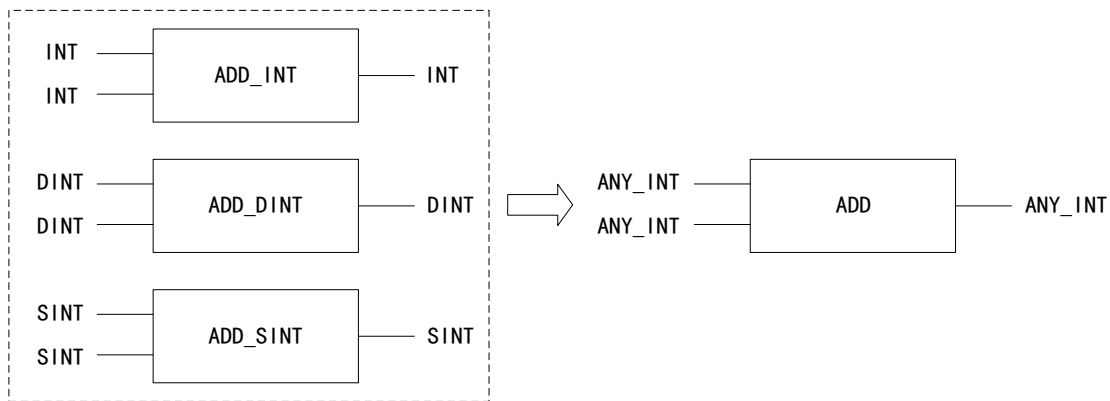
Codesys 支持所有 IEC 的 8 类标准函数。除此之外，还可以使用下列 IEC 标准未规定的函数：ANDN、ORN、XORN、INDEXOF 和 SIZEOF、ADR、BITADR 等。Codesys 共支持如下 11 类函数，具体函数使用及说明在第 6 章会做详细介绍。

3、函数的属性

1) 重载性

对某一个函数来说，如果其输入量以类属数据类型描述，则称为重载函数。这表示该功能的输入量不限于单一的某种数据类型，而是可用于不同的数据类型。Codesys 所有标准函数都具有重载属性，它能够适用于不同的数据类型。如果函数只适用于某数据类型，则需在函数名中给予声明，这称为函数的类型化。

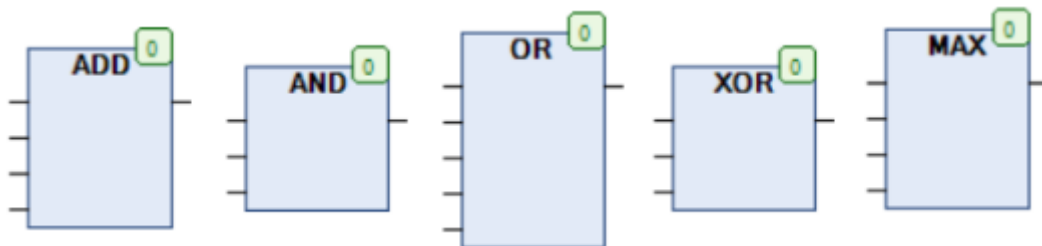
例如一个 PLC 能识别 INT、DINT 和 SINT，则它支持类属数据类型 ANY_INT（包括 BYTE、WORD、DWORD、SINT、USINT、REAL 等）的重载功能 ADD。例如，ADD_INT 是一个限于数据类型的 INT 加法函数，它属于类型化函数，这样看重载功能是独立于类型的。重载函数说明如下图所示：



使用重载函数时，系统会自动选择合适的数据类型。例如，如果调用的 ADD 实参数据类型是 DINT，则系统内部会调用 ADD_DINT 标准功能。

2) 可扩展性

函数的输入变量个数可以扩展的属性称为函数的可扩展属性。例如，ADD 函数的输入变量可以不仅限于两个，它可以实现多个输入变量的加法运算，因此，可以称 ADD 函数具有可扩展属性。并非所有标准函数都具有可扩展属性，该功能的扩展限度受 PLC 所强制的上限、图形编程语言中方框高度限制或函数本身功能定义上的限制，如 DIV 函数就具有该属性。具有可扩展属性的函数可简化程序，降低所需的存储空间。下图是具有可扩展属性的一些函数示例。



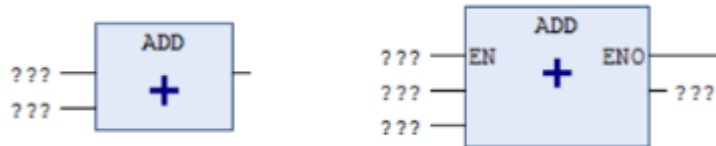
3) EN 和 ENO

只有在梯形图和功能块图编程语言中该属性才有效。EN 和 ENO 分别是函数的输入使能和输出使能。所有的函数都可使用或禁用该属性。

使能输入使能输出的应用原则如下：

- ◆ 当该输入函数被调用时，EN 的值为 False，则该函数体定义的操作不会被程序执行，同时 ENO 的值为 False。
- ◆ EN 为 True 时，该函数被调用，函数体定义的操作被执行，同时 ENO 的值为 True。
- ◆ EN 和 ENO 属性是附加属性，可根据实际需要或使用或禁用该属性。

下图对有EN/ENO的ADD函数和普通的ADD函数进行了比较。



2-4-3. 功能块

功能块 (Function Block) 是把反复使用的部分程序块转换成一种通用部件，他可以在程序中被任何一种编程语言所调用，反复被使用，不仅提高了程序的开发效率，也较少了编程中的错误，从而改善了程序质量。

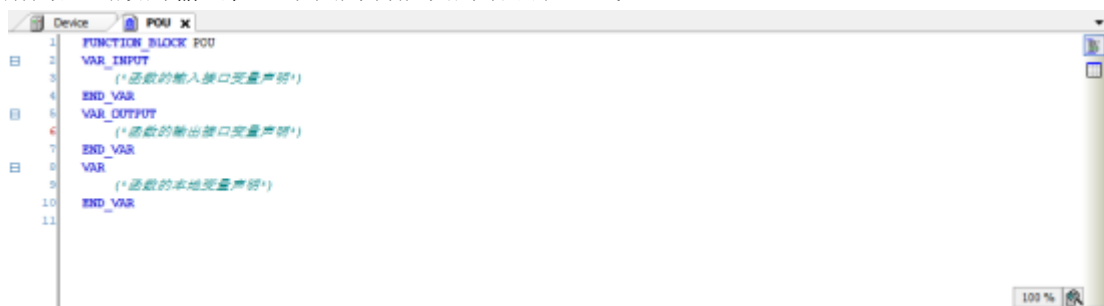
功能块在执行时能够产生一个或多个值的程序组织单元。功能块保留有自己特殊的内部变量，控制器目标执行系统必须给功能块的内部状态变量分配内存，这些内部变量构成自身的状态特征。

功能块的执行逻辑构成了自身的对象行为特征。所以，对于相同参数的输入变量值，由于可能存在不同的内部状态变量，当然就可能得到不同的计算结果。在控制系统中，功能块可以是某种控制算法，例如 PID 功能模块被用于闭环控制，其他功能块可用于计数器，斜坡和滤波等。

1、功能块的表示和声明

1) 自定义功能块的表示

与函数一样，功能块内部逻辑部分可以使用 6 种编程语言中的任意一种。函数名即是函数的返回值，也可以理解为是函数的输出值，下图为功能块的语法表达式。



2) 功能块中变量的声明

功能块中变量声明与函数中变量声明类似，编写时，需注意如下事项：

- ◆ 功能块的内部和输出变量可用限定属性 **RETAIN**，用于表示该变量具有保持功能。而输入变量只能在调用时声明具有保持属性。
- ◆ 一般不允许对功能块输入变量赋值。只有当输入作为功能块的调用部分时，才允许对功能块输入变量赋值。
- ◆ 由于功能块可以调用函数和功能块，所以，也可将调用功能块实例作为其他功能块的实例的变量。如 `DB_FF(S1:=DB_ON.Q, R:=DB_OFF.Q)`。
- ◆ 功能块的输入不赋值表示保持他们的初始值。
- ◆ 为确保功能块不依赖于硬件，功能块的变量声明中不允许将具有固定地址的地址变量（如 `%IX1.1`，`%QD12`）作为局部变量，但在调用时可以给其赋值。
- ◆ 使用 `VAR_INPUT` 和 `VAR_OUTPUT` 会造成占用过多的内存，为此，在功能块编程时，可尽量使用 `VAR_IN_OUT` 替代，减少对存储区的占用。

2、标准功能块

在标准库中已包含双稳态元素、边沿检测、计时器和定时器等功能块。

3、功能块的属性

(1) 实例化

按照 IEC61131-3 的标准，功能块的类型是抽象的结构类型的定义，而不是现实的数据实体，如果不对其进行定义将其实例化，则不能被程序调用和执行。所以功能块是需要实例化后才能被使用。

实例化后的功能块是拥有私有数据、可按照既定逻辑完成特定功能、完全封装的、独立的结构型变量。从而将之前的抽象类型定义转换为数据实体。

(2) 扩展性

Codesys 支持面向对象的编程方式，所以功能块也可以派生出“子”功能块。这样“子”功能块具有“父”功能块的属性，并且可以具有自己附加的特性，可以形象的认为“子”功能块是对“父”功能块的扩展。所以在本文中，把这个叫做“功能块的扩展”。

在声明功能块时加上关键字“EXTENDS”就可以使用扩展功能。也可以通过在“添加对象”对话框添加功能块时，选择“extends”选项来实现扩展。

(3) EN 和 ENO

功能块具有 EN 和 ENO 的附属属性，与函数中 EN 和 ENO 的使用方法类似。

(4) 函数功能块的区别

综上所述，函数和功能块明显的区别如下表总结：

	函数 (FUN)	功能块 (FB)
内存分配	没有指定的内存分配地址	全部数据分配内存地址
输入/输出变量	只允许一个输出变量	多个输出变量或没有输出变量
调用关系	可调用函数，但不能调用功能块	可调用功能块或函数

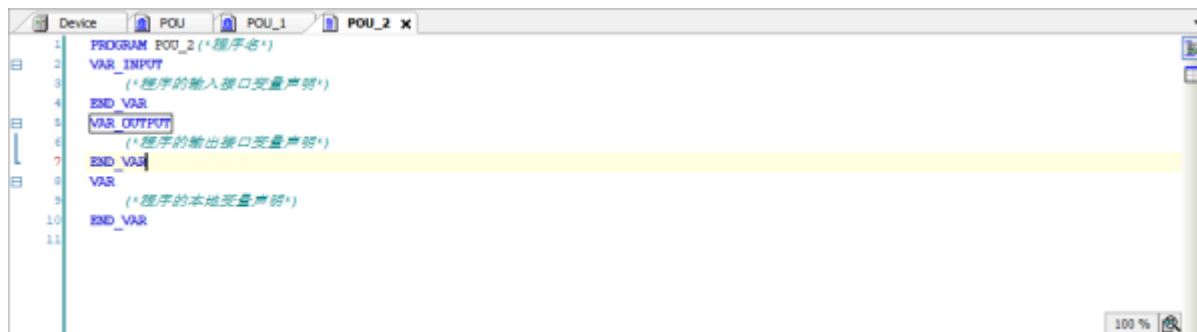
2-4-4. 程序

程序 (Program) 是规划一个任务的主核心，程序拥有最大的调用权，可以调用功能块及函数。

一般而言分为主程序、子程序，广义上讲，也包含硬件配置、任务配置、通讯配置及目标设置信息。一般在程序中定义普通全局变量、映射硬件地址全局变量、局部变量。通过程序间调用实现应用逻辑。

1、程序的表示和声明

程序采用如下的语法表达式表示，程序逻辑部分可以使用 6 种编程语言中的任意一种。



2、程序的性能

(1) 一个程序可包含地址的配置。允许声明存放 PLC 物理地址的直接表示变量，直接表示的地址配置仅用于程序中内部变量的声明。直接表示变量允许分级寻址方式描述，如可有如下的表示。

可以在程序声明中按如下格式填写。

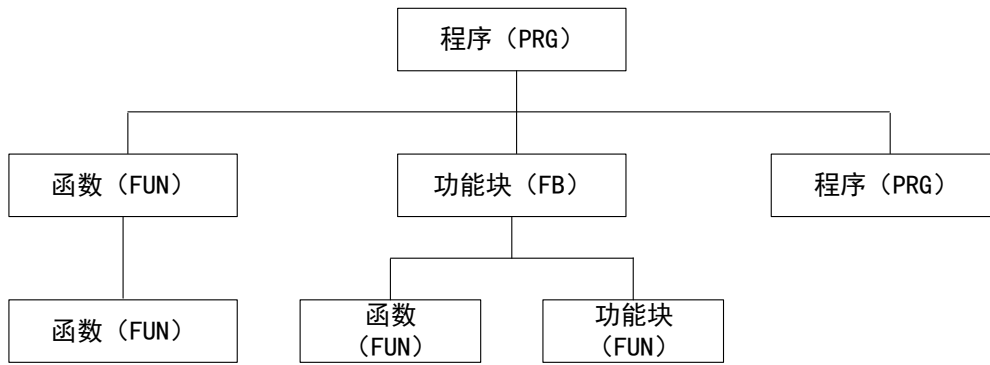
```
bTest AT %QX10.3:BOOL:=TRUE;
```

(2) 程序组织单元不能直接或间接调用其本身，即程序组织单元不能调用由相同类型和相同名称的程序组织单元实例；

(3) 程序仅在资源中实例化。在资源中被声明。程序的实例只需将程序与一个任务结合，否则不会被执行。而功能块仅能在程序或其他功能块中实例化。

3、程序调用关系

在程序中允许调用功能块实例，函数升至调用其他程序，如下图所示：



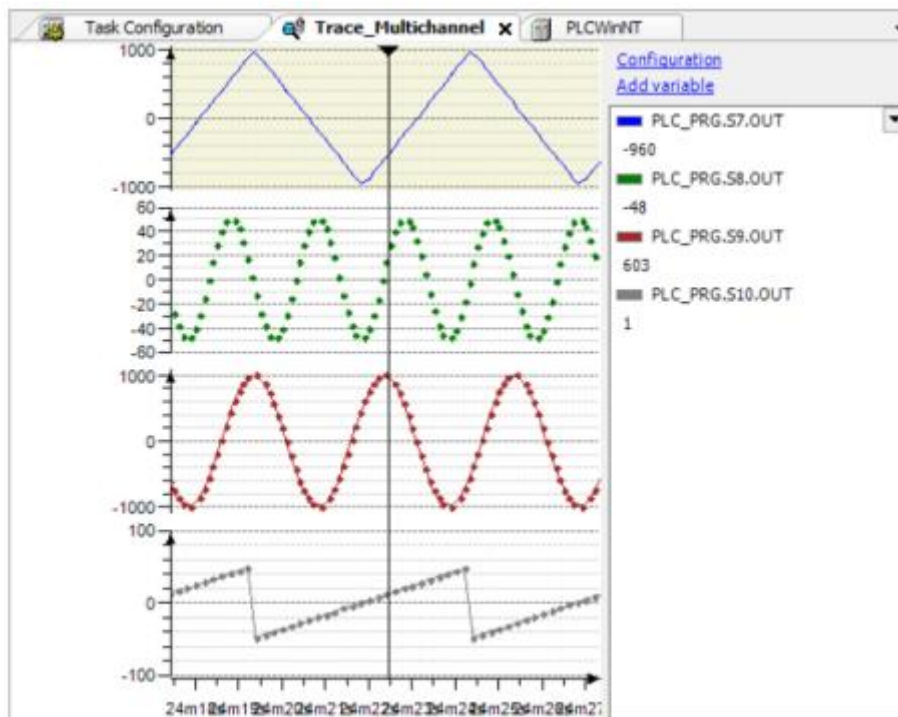
根据上图中的显示，函数和功能块用于构成子程序，程序用于构成用户主程序，因此，程序被认为是全局的。程序是程序组织单元中的最大形式，它可以调用函数，功能块及程序。

功能块允许调用其它功能块及函数。由于函数不存在私有变量，故函数只能调用其他函数，不能调用功能块实例。

2-5. 应用对象

2-5-1. 采样跟踪

采样跟踪的功能是对控制器上变量值历史记录监视跟踪。采样跟踪的工作方式类似于数字采样示波器，是程序调试和诊断过程中非常实用和有效的调试工具。用户可以通过在跟踪管理器中对“跟踪对象”进行添加，对“跟踪配置”进行设置，将程序在执行过程中用到的命令字、状态字、电机运动的速度、位置等参数全程记录下来，用户可以通过对这些参数的观察，了解到程序在控制系统中运行的整个过程。该功能如下图所示：



2-5-2. 持续变量

持续变量的功能是将数据在系统关机或者发生异常中断后保存到存储单元，并在重新上电后调出，并能够继续被程序实用。为适应现场工况，设计 PLC 控制系统时，需考虑掉电或者异常中断后，对数据的保存与恢复问题，用户可通过添加 PersistentVars 列表，将需要掉电保持的数据在列表中进行注册，即可实现持续变量功能。

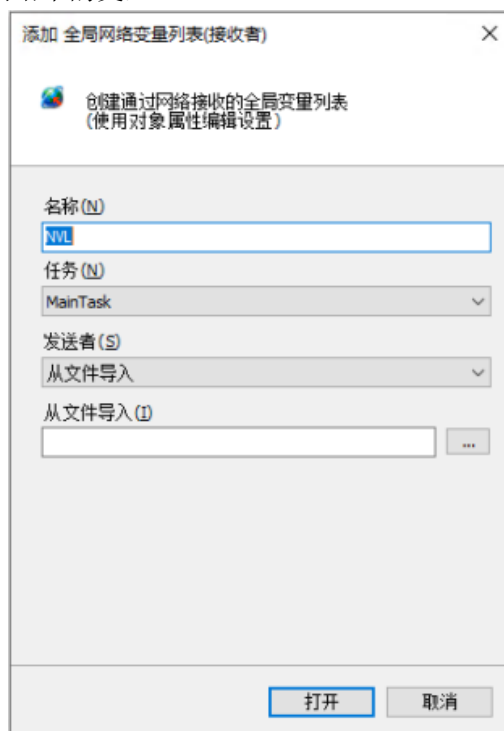
2-5-3. 数据单元类型

数据单元类型（DUT）的功能是给用户提供一个自定义的数据类型，数据类型包括结构体、枚举、别名和联合体等，如下图所示。数据单元类型的使用在用户编程过程中起到规范编程流程、提高编程效率、优化编程格式的功能，实现面向对象的编程方式。



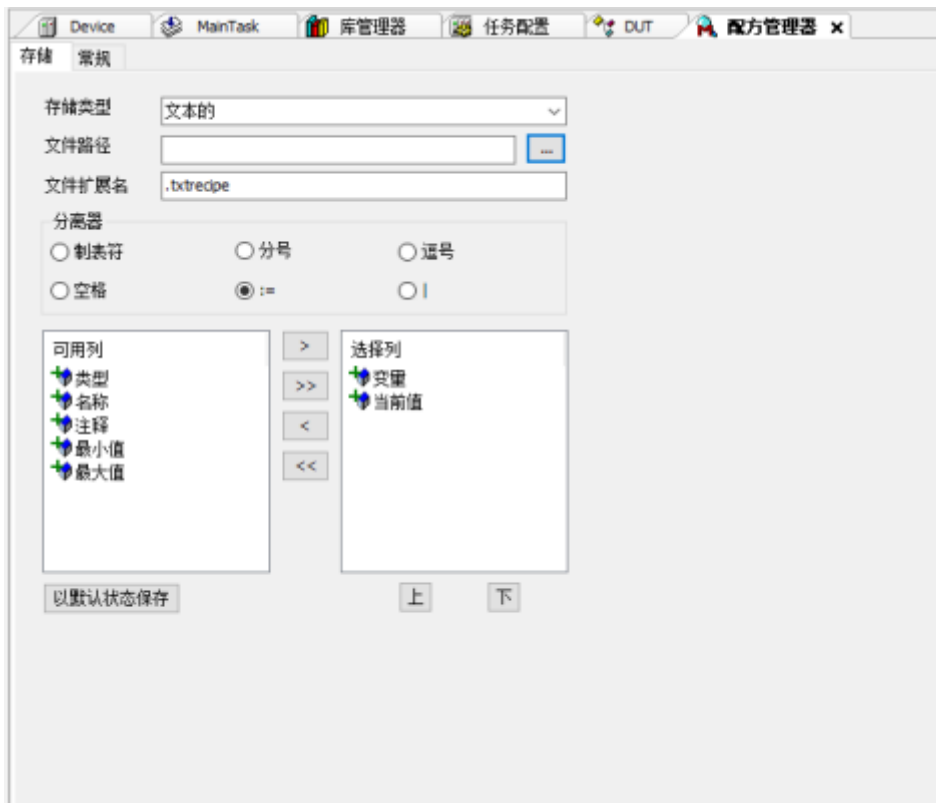
2-5-4. 全局网络变量

全局网络变量列表（GNVL）分为发送方和接收方两种形式。发送方的功能是声明和列出应发送到其他设备或网络项目的网络变量列表（接收方）的全局变量；接收方的功能是列出接收到的网络变量并显示信息（网络、传输信息和发送者等）。用户通过配置全局网络变量编辑器将发送方和接收方的全局网络列表添加到设备树中实现全局变量在网络中的交互。



2-5-5. 配方管理器

配方管理器的功能是提供用于维护用户定义的变量列表（配方定义）。用户通过配方管理器可以对存储位置、存储方式、存储类别进行配置，如下图所示。配方管理器配置成功后，用户即可实现对配方定义的上传与下载。



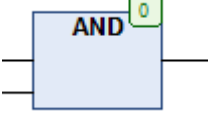
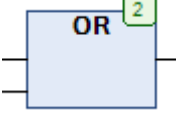
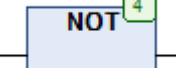
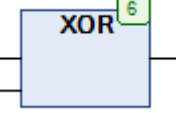
3. 基础指令

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

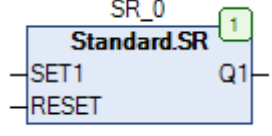
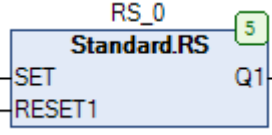
3. 基础指令	26
3-1. 位逻辑指令	27
3-1-1. 基本位逻辑指令	27
3-1-2. 置位优先与复位优先触发器指令	27
3-1-3. 数据单元类型	27
3-2. 定时器指令	28
3-3. 计数器指令	28
3-4. 数据处理指令	29
3-4-1. 选择操作指令	29
3-4-2. 比较指令	29
3-4-3. 移位指令	30
3-5. 运算指令	30
3-5-1. 赋值指令	30
3-5-2. 算数运算指令	30
3-5-3. 数学运算指令	31
3-5-4. 地址运算指令	31
3-5-5. 数据转换指令	32

3-1. 位逻辑指令

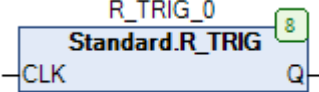
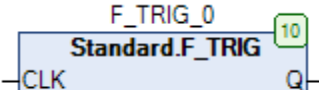
3-1-1. 基本位逻辑指令

指令名称	指令图标	功能介绍
AND		运算符“与”
OR		运算符“或”
NOT		运算符“非”
XOR		运算符“异或”

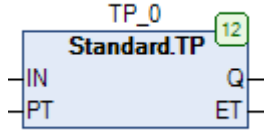
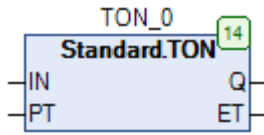
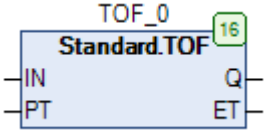
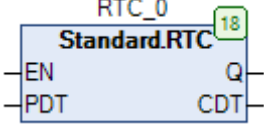
3-1-2. 置位优先与复位优先触发器指令

指令名称	指令图标	功能介绍
SR		置位优先触发器：置位双稳态触发器，置位优先
RS		复位优先触发器：复位双稳态触发器，复位优先

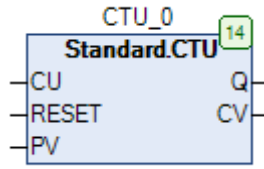
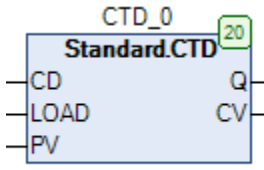
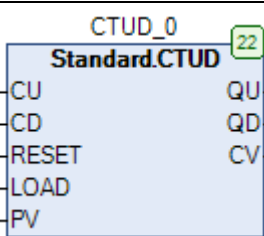
3-1-3. 数据单元类型

指令名称	指令图标	功能介绍
R_TRIG		上升沿触发
F_TRIG		下降沿触发

3-2. 定时器指令

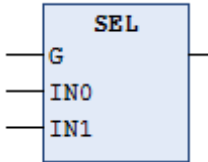
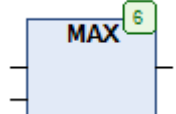

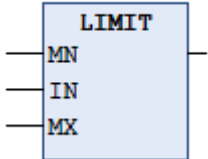
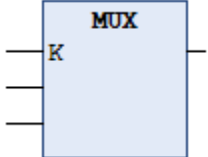
指令名称	指令图标	功能介绍
TP		脉冲定时器：一旦 IN 变为 TRUE，则 Q 为 TRUE，时间将开始在 ET 中以毫秒为单位计数，直到其值等于 PT，则 Q 为 FALSE
TON		通电延时定时器：一旦 IN 变为 TRUE，时间将开始在 ET 中以毫秒为单位计数，直到其值等于 PT，则 Q 为 TRUE
TOF		断电延时定时器：当 IN 为 FALSE 且 ET 等于 PT 时，Q 为 FALSE。否则为真
RTC		实时时钟：在给定时间启动，返回日期和时间

3-3. 计数器指令

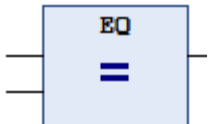
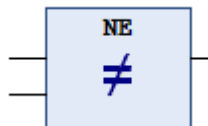
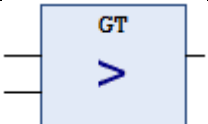
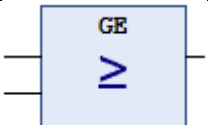
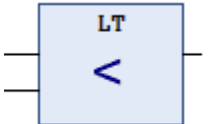
指令名称	指令图标	功能介绍
CTU		增计数器：如果 RESET 为 TRUE，则初始化为 0；CU 上升沿时总是加 1，一旦 CV >= PV，Q 将被设置为 TRUE
CTD		减计数器：如果 LOAD 为 TRUE，CV 将设置为 PV 给定的起始值。CD 上升沿时总是加 1，计数器值(CV)递减 1 直到 0，Q 将被设置为 TRUE
CTUD		增/减双向计数器

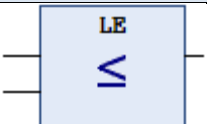
3-4. 数据处理指令

3-4-1. 选择操作指令

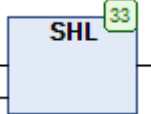
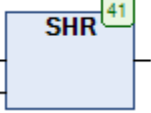
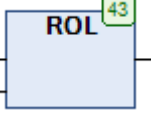

指令名称	指令图标	功能介绍
SEL		二选一指令：选择开关为 FALSE 时，输出为第一个输入数据，选择开关为 TRUE 时，输出为第二个数据
MAX		取最大值
MIN		取最小值
LIMIT		限制值：如果该 IN 值高于 Max 上限，则 LIMIT 产生 Max。如果 IN 的值低于 Min 下限，则结果为 Min
MUX		多选一：MUX 从一组值中选择第 K 个值。第一个值是 K=0。如果 K 大于其他输入的数量(n)，则 Codesys 传递最后一个值

3-4-2. 比较指令

指令名称	指令图标	功能介绍
EQ		等于
NE		不等于
GT		大于
GE		大于或等于
LT		小于

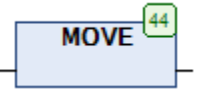
指令名称	指令图标	功能介绍
LE		小于或等于

3-4-3. 移位指令

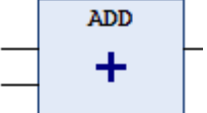
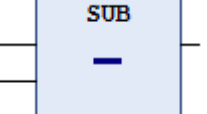

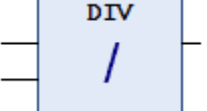
指令名称	指令图标	功能介绍
SHL		按位左移
SHR		按位右移
ROL		循环左移
ROR		循环右移

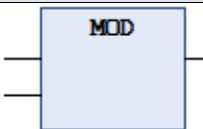
3-5. 运算指令

3-5-1. 赋值指令

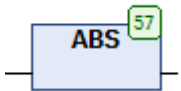
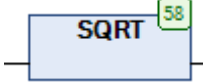
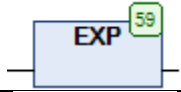
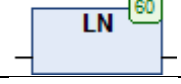

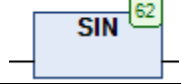
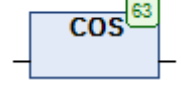
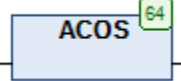
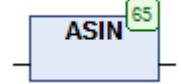
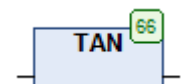
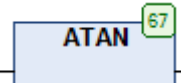
指令名称	指令图标	功能介绍
MOVE		赋值

3-5-2. 算数运算指令

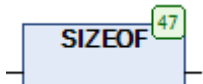


指令名称	指令图标	功能介绍
ADD		相加
SUB		相减
MUL		相乘
DIV		相除

指令名称	指令图标	功能介绍
MOD		取余

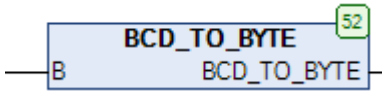
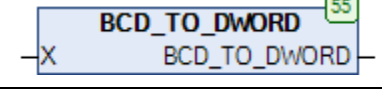
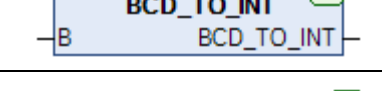
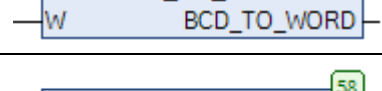
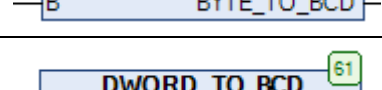
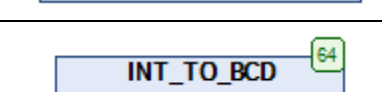
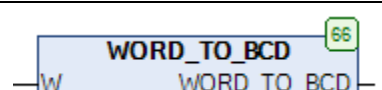

3-5-3. 数学运算指令

指令名称	指令图标	功能介绍
ABS		绝对值指令
SQRT		平方根指令
EXP		指数指令
LN		自然对数指令
LOG		常用对数指令
SIN		正弦指令
COS		余弦指令
ACOS		反余弦指令
ASIN		反正弦指令
TAN		正切指令
ATAN		反正切指令

3-5-4. 地址运算指令

指令名称	指令图标	功能介绍
SIZEOF		数据类型大小
ADR		地址操作符
BITADR		位地址操作符

3-5-5. 数据转换指令

指令名称	指令图标	功能介绍
BCD_TO_BYTE		BCD 转换为 BYTE
BCD_TO_DWORD		BCD 转换为 DWORD
BCD_TO_INT		BCD 转换为 INT
BCD_TO_WORD		BCD 转换为 WORD
BYTE_TO_BCD		BYTE 转换为 BCD
DWORD_TO_BCD		DWORD 转换为 BCD
INT_TO_BCD		INT 转换为 BCD
WORD_TO_BCD		WORD 转换为 BCD

4. 特殊功能

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

4. 特殊功能	33
4-1. 高速计数	34
4-1-1. 功能概述	34
4-1-2. 功能块介绍	34
4-1-3. 参数配置	36
4-1-4. 应用举例	37
4-2. 外部中断	37
4-2-1. 功能概述	37
4-2-2. 应用举例	38
4-3. PLC SHELL	38
4-3-1. 功能概述	38
4-3-2. 指令列表	38
4-3-3. 应用举例	38
4-4. 时钟	43
4-4-1. 功能概述	43
4-4-2. 应用举例	43

4-1. 高速计数

4-1-1. 功能概述

XS 系列 PLC 具有高速计数功能，通过选择不同的计数器来实现针对测量传感器和旋转编码器等高速输入信号的测定，其最高测量频率可达 200kHz。

4-1-2. 功能块介绍

1、指令格式

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

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	Mode	AB_Mode, Single_Mode	FALSE	高速计数模式：MODE=XJ.AB_Mode,为AB相高速计数，MODE=XJ.Single_Mode,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	

注意: ErrorID 若显示的值为 2, 是由于 CounterID 的范围不在 0-3。

3、功能说明

1) 高速计数功能共有 3 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; 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。XSDH 系列高速输入为接收集电极开路信号 (OC)。

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 和 XSDH 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=Single_Mode)

此模式下, 计数输入脉冲信号, 计数值随着每个脉冲信号好的上升沿递增计数。

(2) AB 相模式 (Mode=AB_Mode)

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) XS 系列高速计数输入端口分配

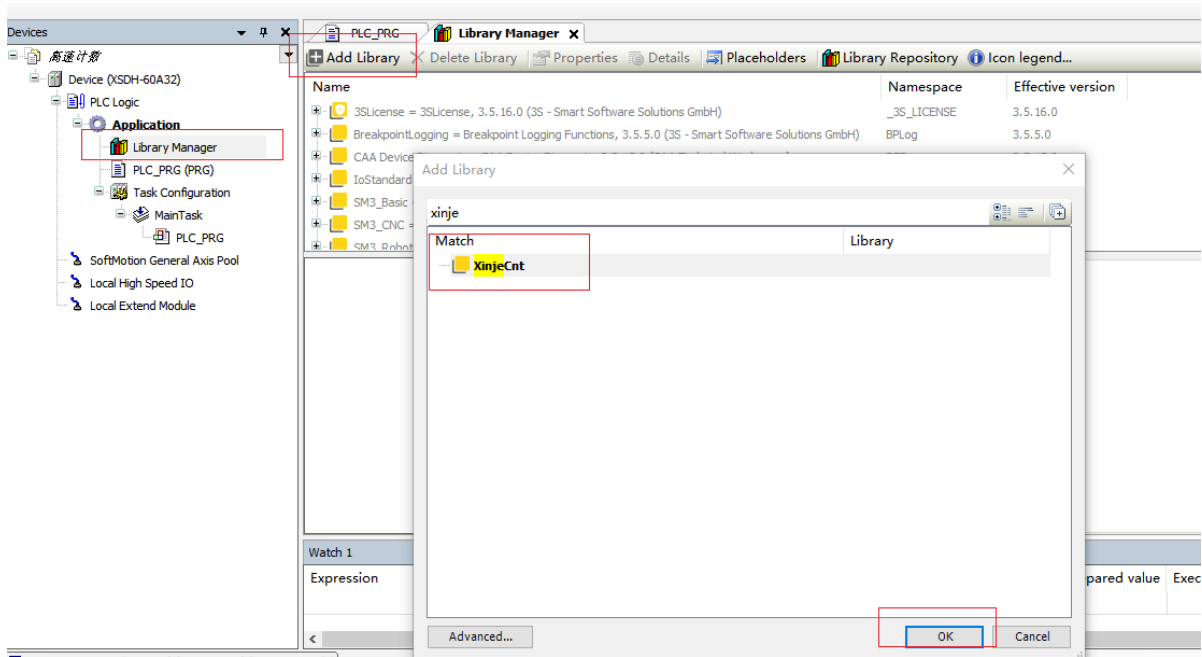
XS3-26T4								
	单相递增模式				AB相模式			
Counter ID	0	1	2	3	0	1	2	3
最高频率	200k	200k	200k	200k	200k	200k	200k	200k
X0+	U+				A+			
X0-	U-				A-			
X1+					B+			
X1-					B-			
X2								
X3+		U+				A+		
X3-		U-				A-		
X4+						B+		
X4-						B-		
X5								
X6+			U+				A+	
X6-			U-				A-	
X7+							B+	
X7-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

XSDH-60A32-E								
	单相递增模式				AB相模式			
Counter ID	0	1	2	3	0	1	2	3
最高频率	200k	200k	200k	200k	100k	100k	100k	100k
X0	U				A			
X1					B			
X2								
X3		U				A		
X4						B		
X5								
X6			U				A	
X7							B	
X10								
X11				U				A
X12								B
X13								

4-1-3. 参数配置

添加库文件：

在“Library Manager”里添加“XinjeCnt”，添加完成可使用高速计数功能。



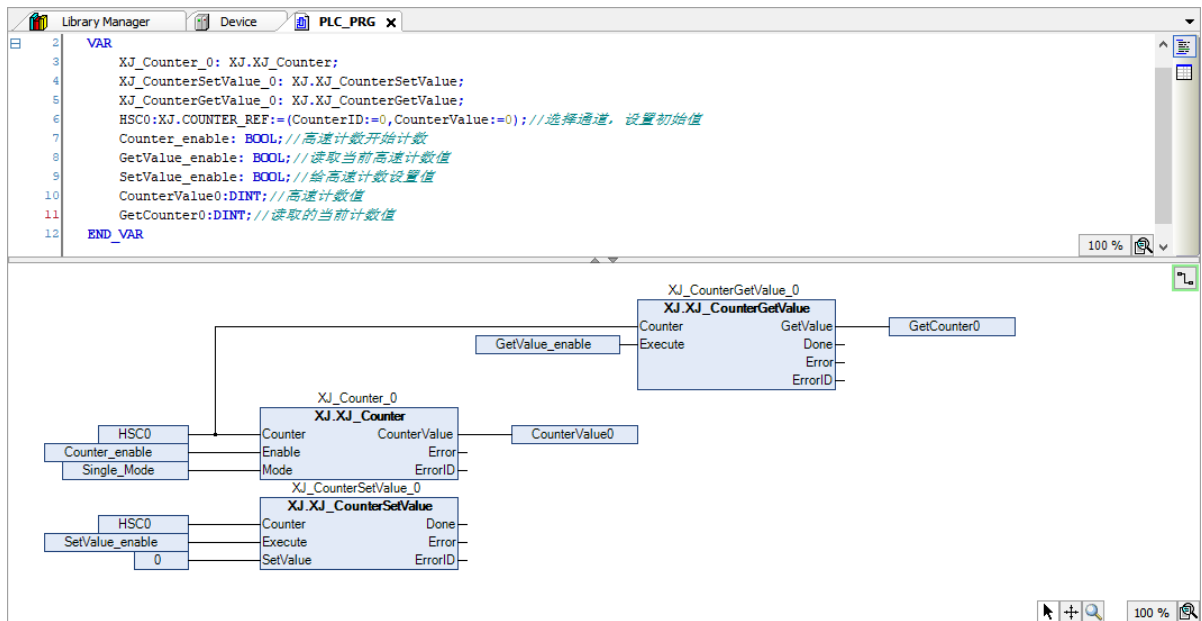
4-1-4. 应用举例

例一：使用第一路高速计数，并可在计数中读取当前计数值，修改当前高速计数值。

程序操作：

- (1) 按照 4-1-3 节中的步骤安装需要使用的库；
- (2) 编写高速计数程序。

编写程序：使用功能块“XJ.XJ_Counter”、“XJ.XJ_CounterGetValue”、“XJ.XJ_CounterSetValue”，可在程序中设置使用的高速计数端口、高速计数模式、设定的高速计数值。



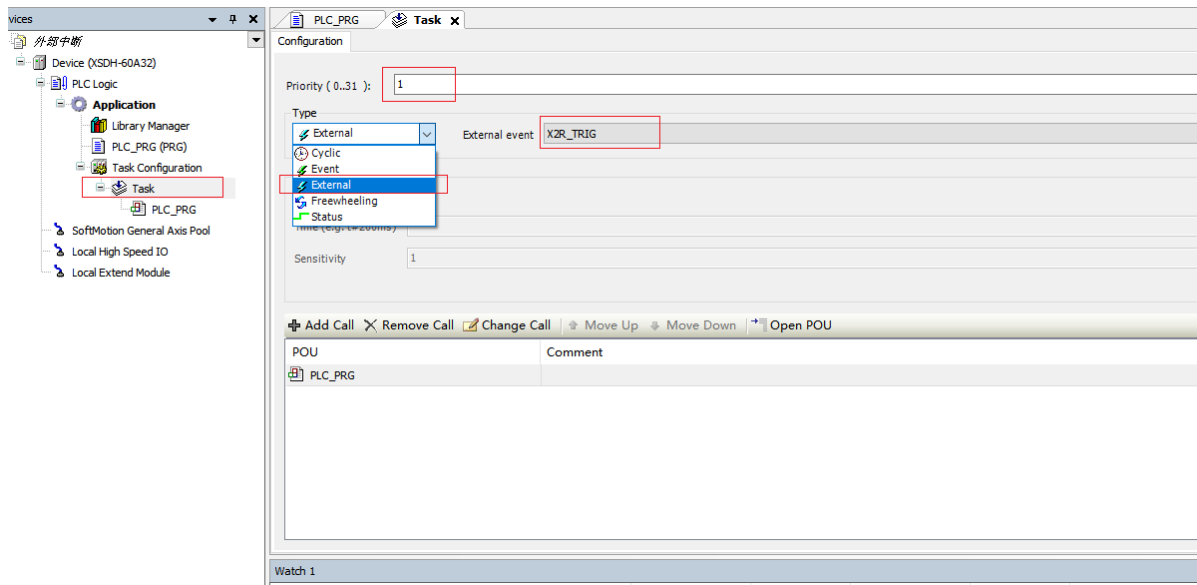
4-2. 外部中断

4-2-1. 功能概述

XS 系列 PLC 支持 X 端子中断，同一个端子支持上升沿和下降沿中断，在 Codesys 中通过任务类型中的外部事件形式来使用中断。如 X2R_TRIG 代表 X2 上升沿中断，X2F_TRIG 代表下降沿中断，各机型支持的中断数量和类型见外部事件“External event”选项。

4-2-2. 应用举例

双击“Task”，在弹出界面中设置为外部事件“External”-外部中断使用端子 X，也可设置外部中断事件的优先级。



4-3. PLC SHELL

4-3-1. 功能概述

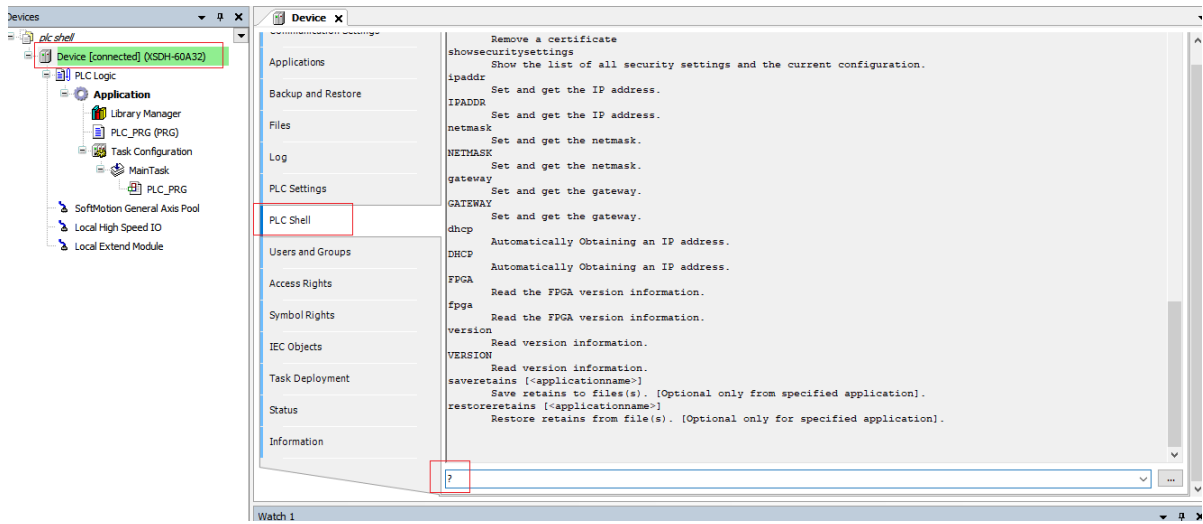
PLC SHELL 功能是一个基于文本的控制监视器，可以用于查询控制器的特定信息，在输入窗口中输入指定命令，结果窗口中接收来自控制器的响应。

4-3-2. 指令列表

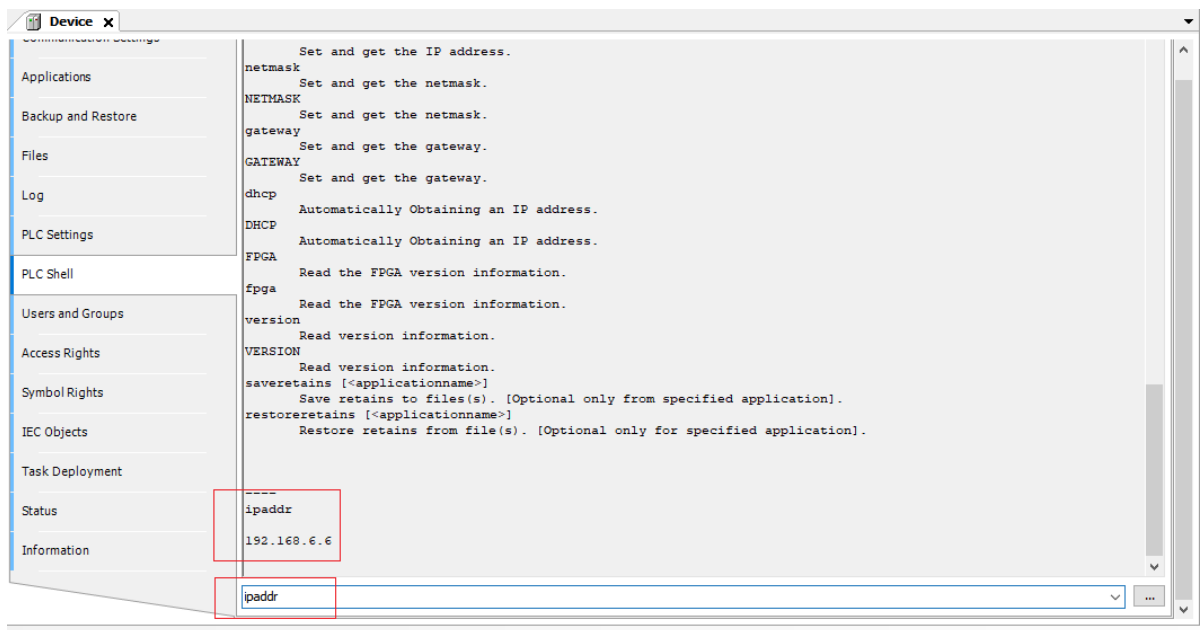
指令名称	功能介绍
ipaddr / IPADDR	获取/设置 PLC 的 IP 地址
netmask / NETMASK	获取/设置 PLC 的子网掩码
gateway / GATEWAY	获取/设置 PLC 的网关
dhcp / DHCP	设置 IP 为自动获取
fpga / FPGA	获取 PLC 的 fpga 版本
version / VERSION	获取 PLC 的固件版本
rtc-get / RTC-GET	获取当前 UTC 时间
rtc-set / RTC-SET	设置 UTC 时间

4-3-3. 应用举例

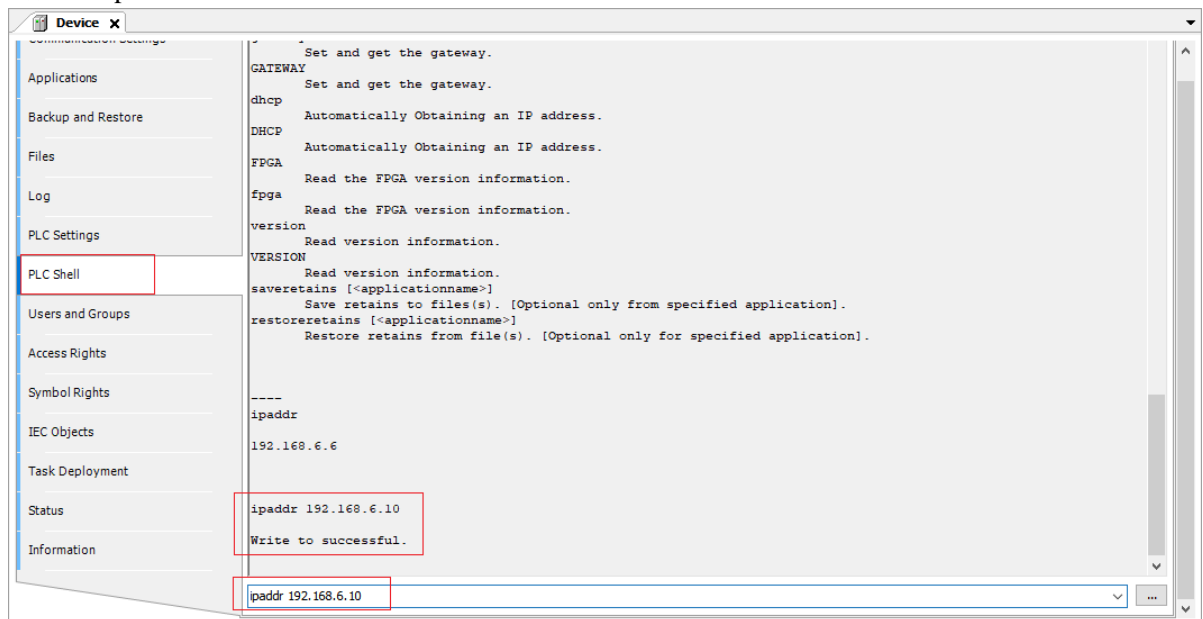
双击“Device”，在“PLC Shell”里输入“?”，可以出来全部功能。可以在这里修改 IP，获取固件版本，设置/读取时钟信息等等。



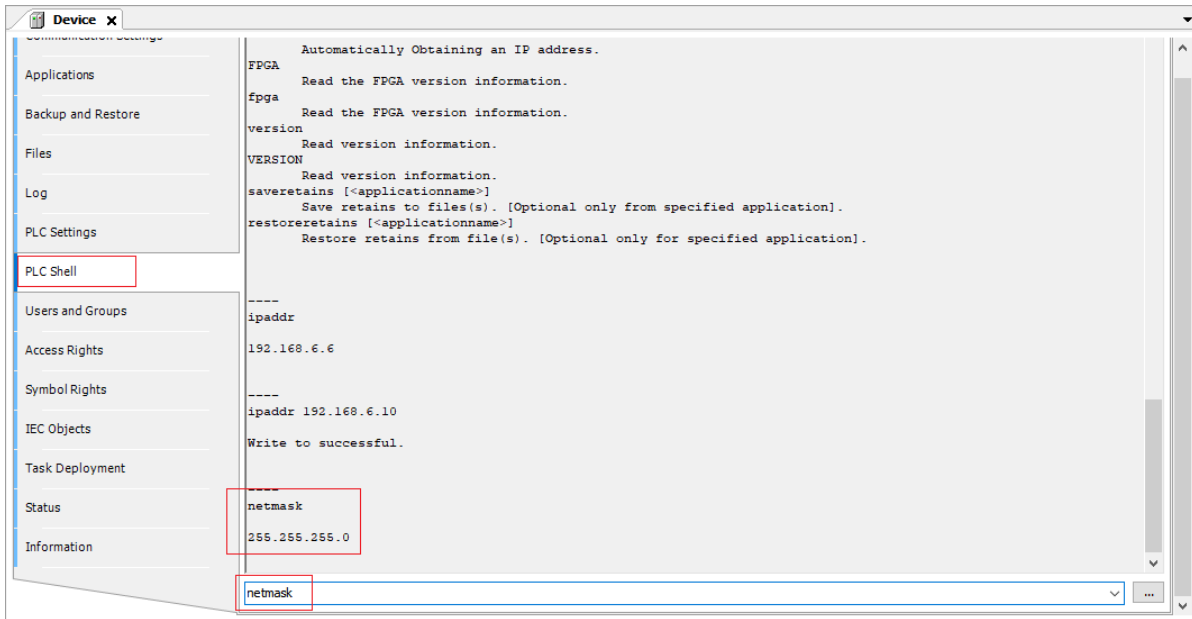
比如：输入“ipaddr”就可以获取到 PLC 的当前的 IP 地址。



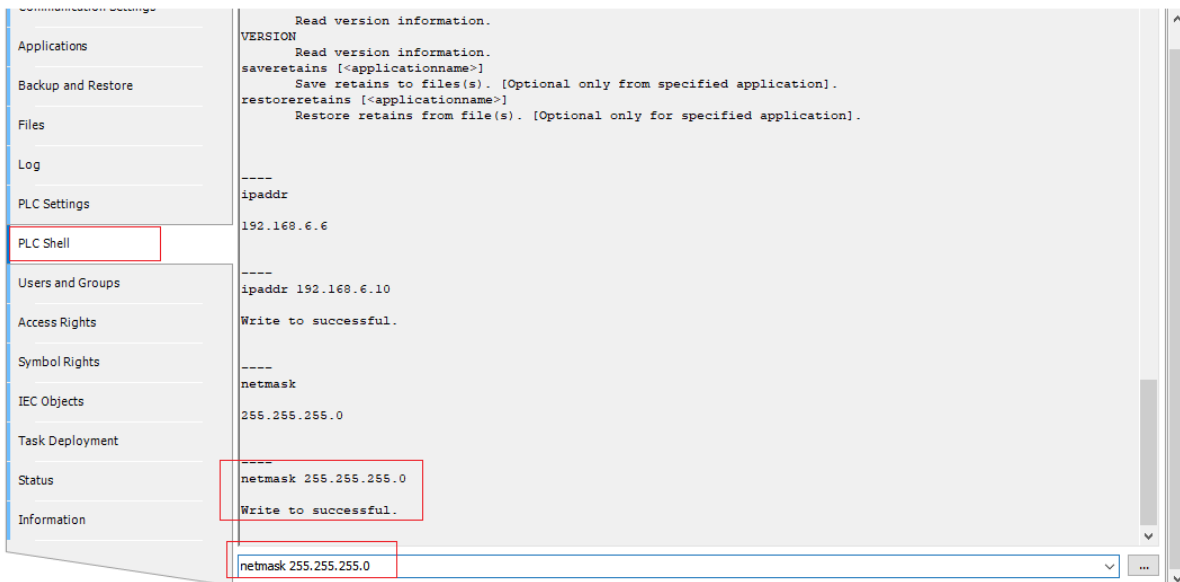
输入“ipaddr 192.168.61.196”，设置 PLC 的 IP 地址，显示“Write to successful”则写入成功。



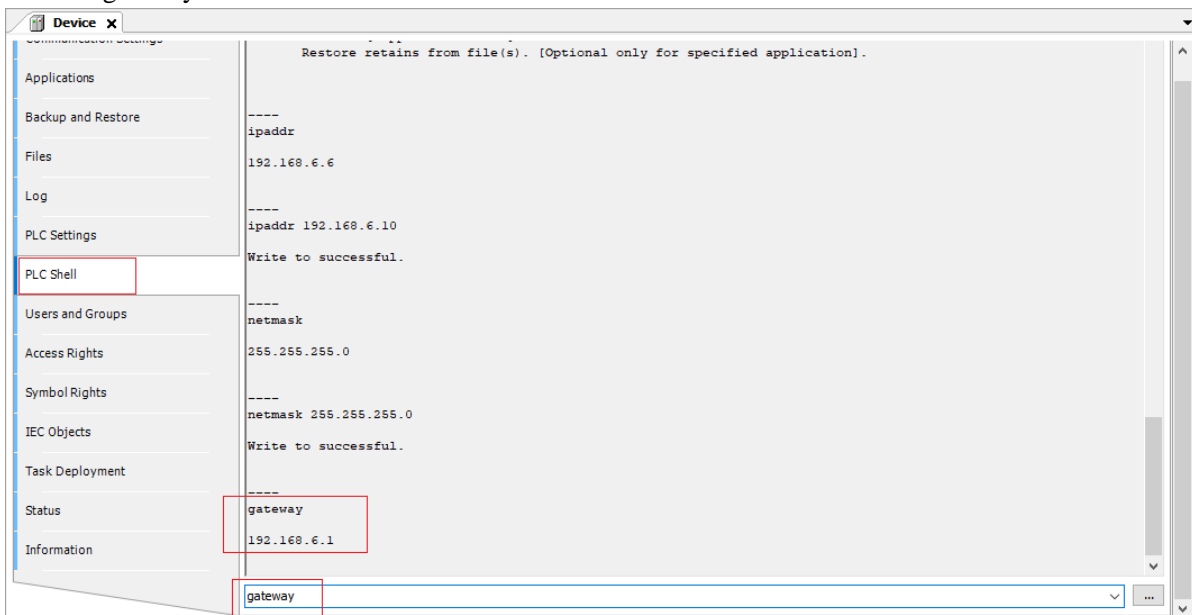
输入“netmask”，就可以获取到 PLC 的当前的子网掩码。



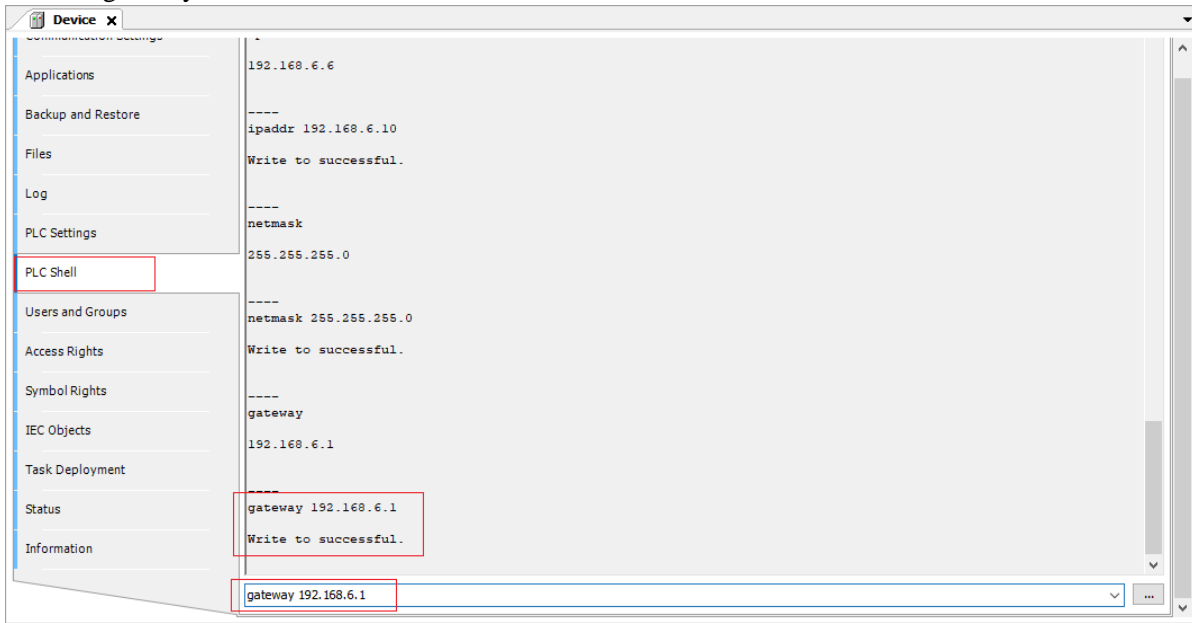
输入“netmask 255.255.254.0”，设置 PLC 的子网掩码，显示“Write to successful”则写入成功。



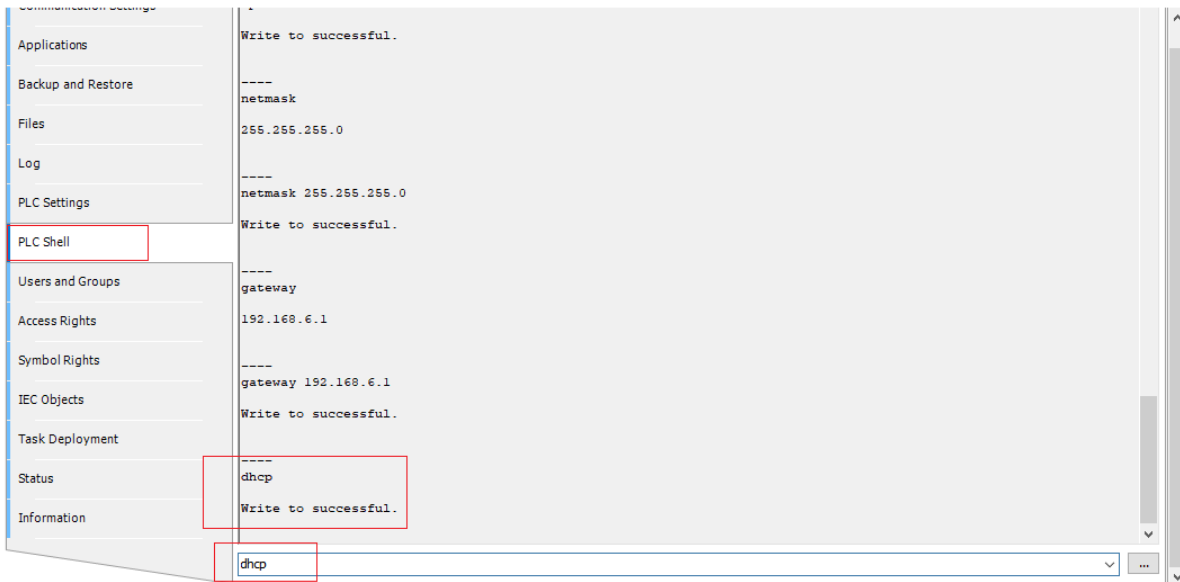
输入“gateway”，就可以获取到 PLC 的当前的默认网关。



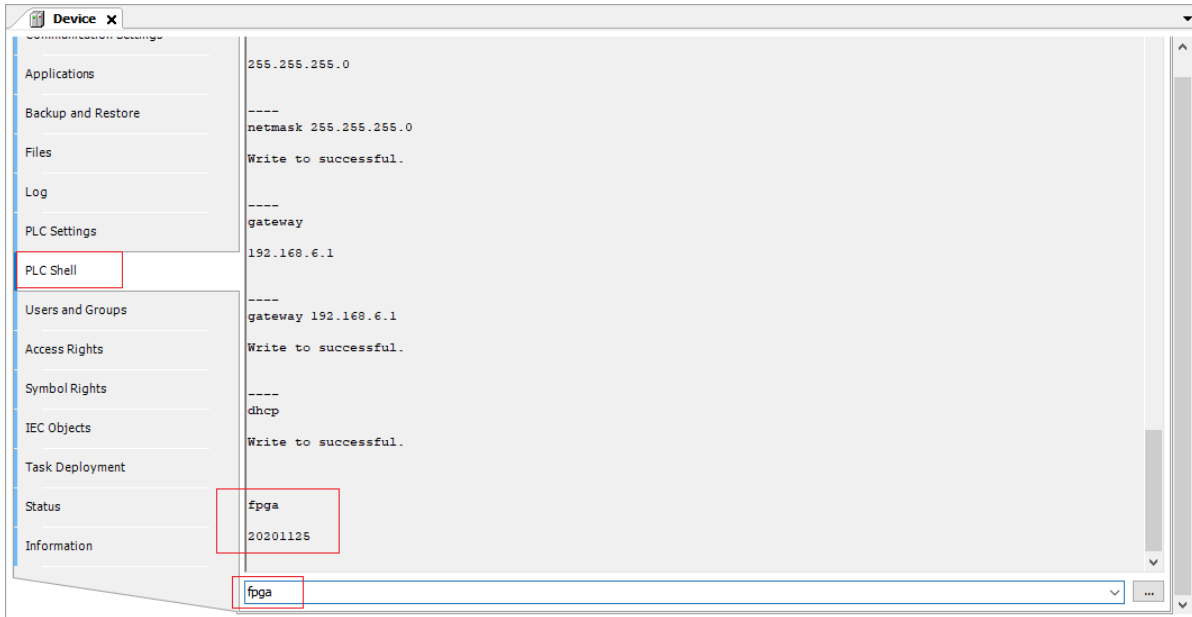
输入“gateway 192.168.60.1”，设置 PLC 的网关，显示“Write to successful”则写入成功。



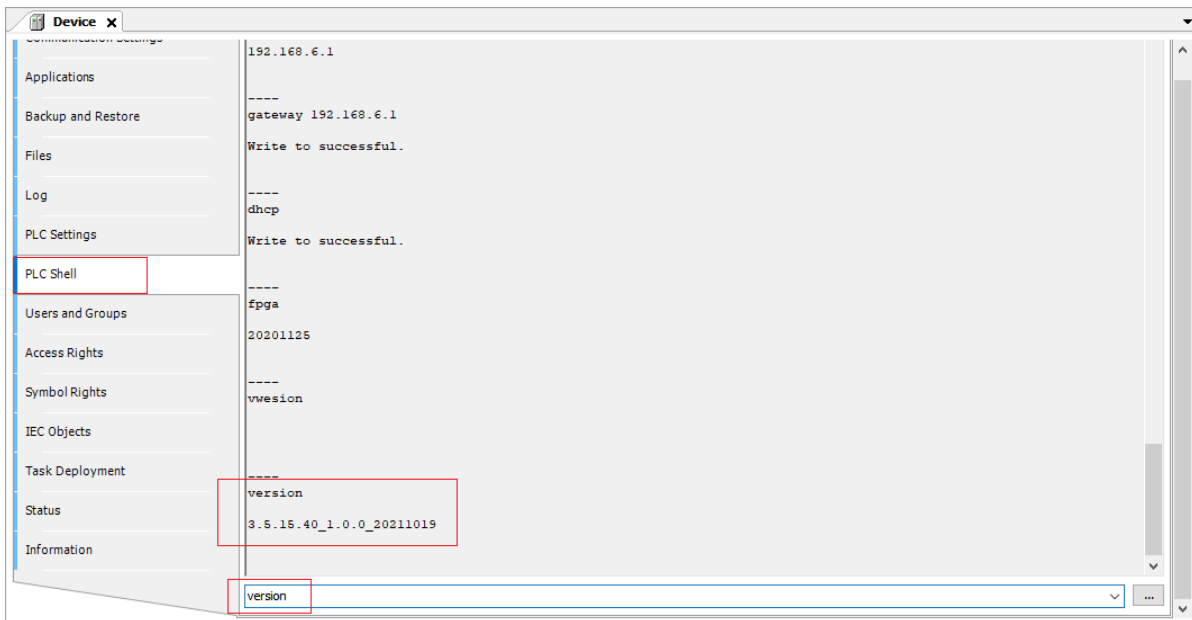
输入“dhcp”，设置 PLC 的 IP 获取方式为自动获取，显示“Write to successful”则写入成功。当 IP 的获取方式为自动获取时，需要保证网络环境良好。



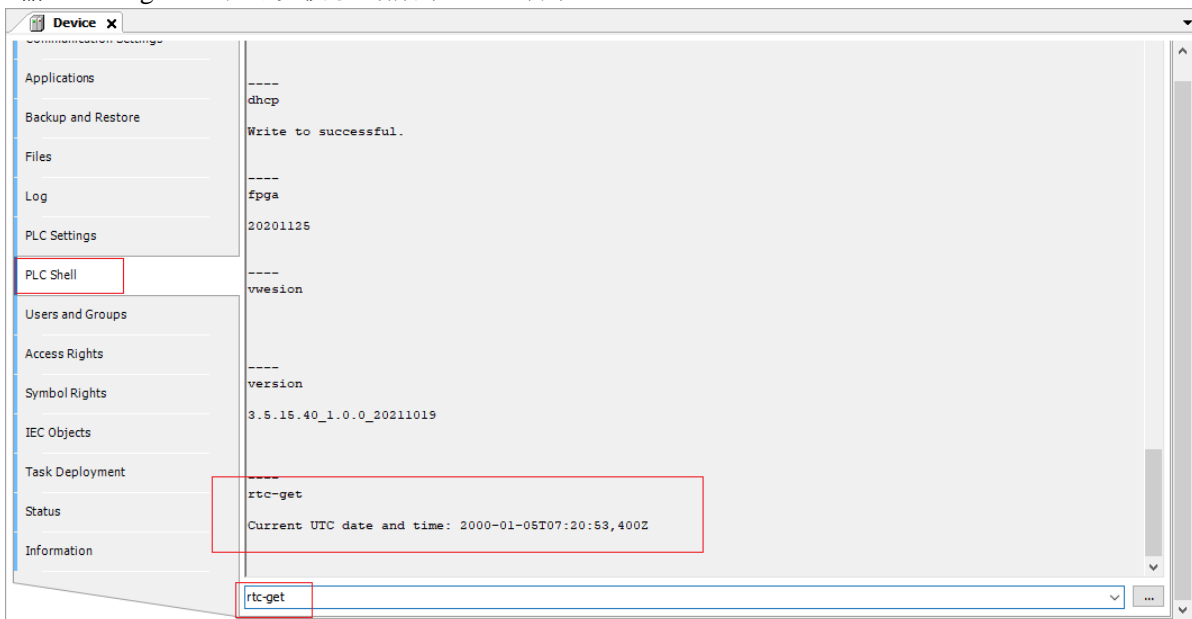
输入“fpga”，就可以获取到 PLC 当前的 FPGA 版本。



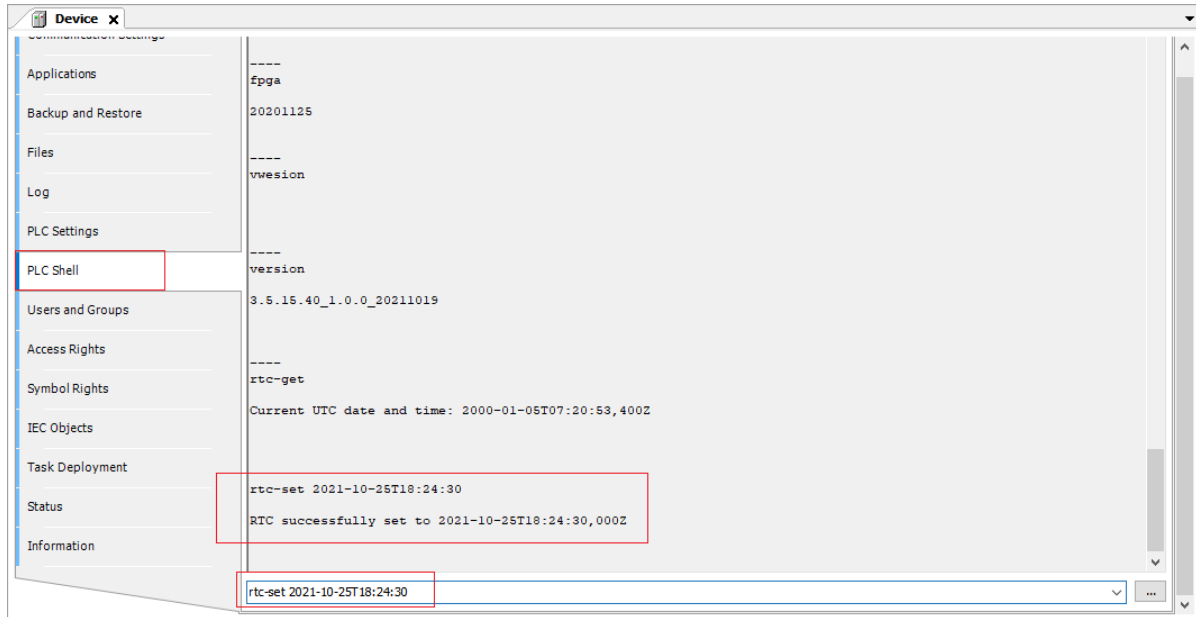
输入“version”，就可以获取到 PLC 当前的固件版本。



输入“rtc-get”，就可以获取当前的 UTC 时间。



输入“rtc-set 2021-10-25T18:24:30”，可设置 UTC 时间。显示“RTC successfully set to 2021-10-25T18:24:30,000Z”则写入成功。其中“000Z”显示不定。



4-4. 时钟

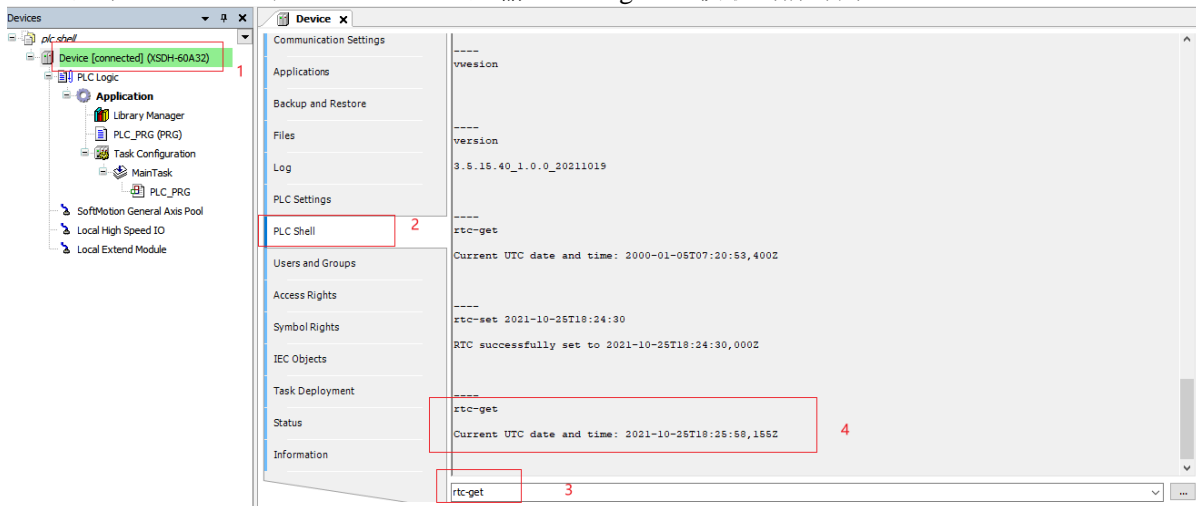
4-4-1. 功能概述

XS 系列 PLC 集成了 RTC，用于记录当前的系统时间，该时钟由电池供电，可保证时间的精确性，同时也支持用户手动修改 RTC 时间。

4-4-2. 应用举例

获取事件的方式：

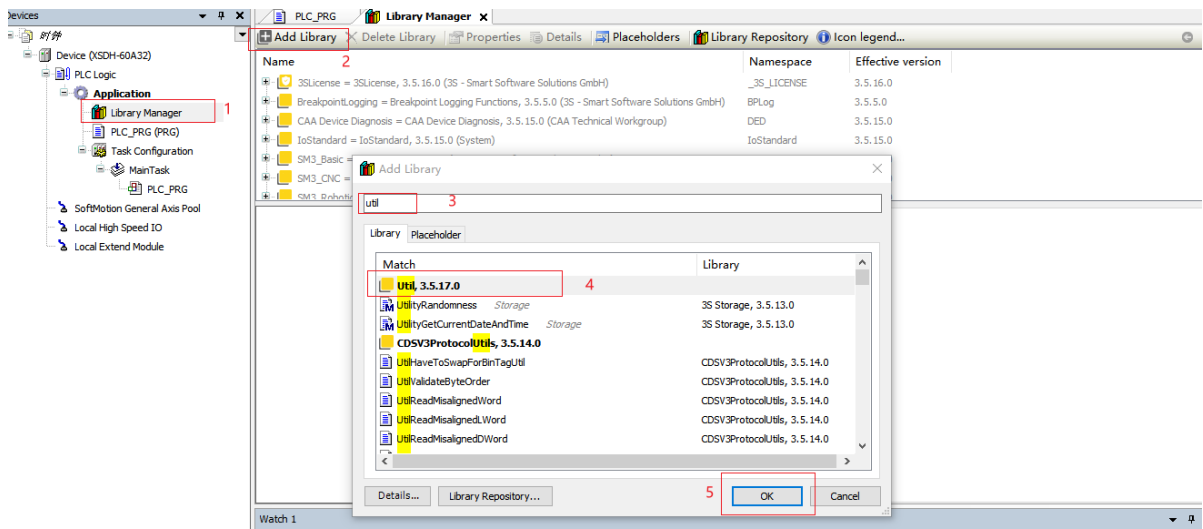
1、双击“Device”，在“PLC Shell”里输入“rtc-get”，获取当前时间。



2、通过时钟指令

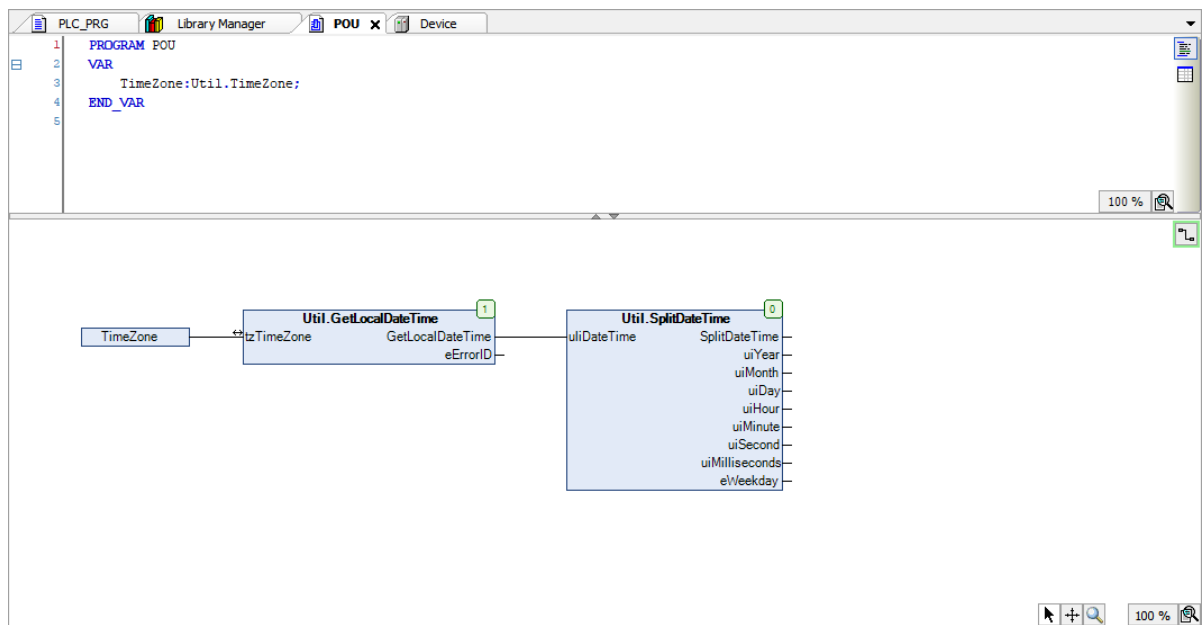
(1) 添加相应的库文件

在“Library Manager”里添加“Util”，添加完成可使用时钟功能。



(2) 编写时钟程序

可使用功能块“Util.GetLocalDateTime”、“Util.SplitDateTime”获取当前时间。在此库中还有其它关于时钟的功能块，可在库“Util”中查看。



5. Codesys 工程举例

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

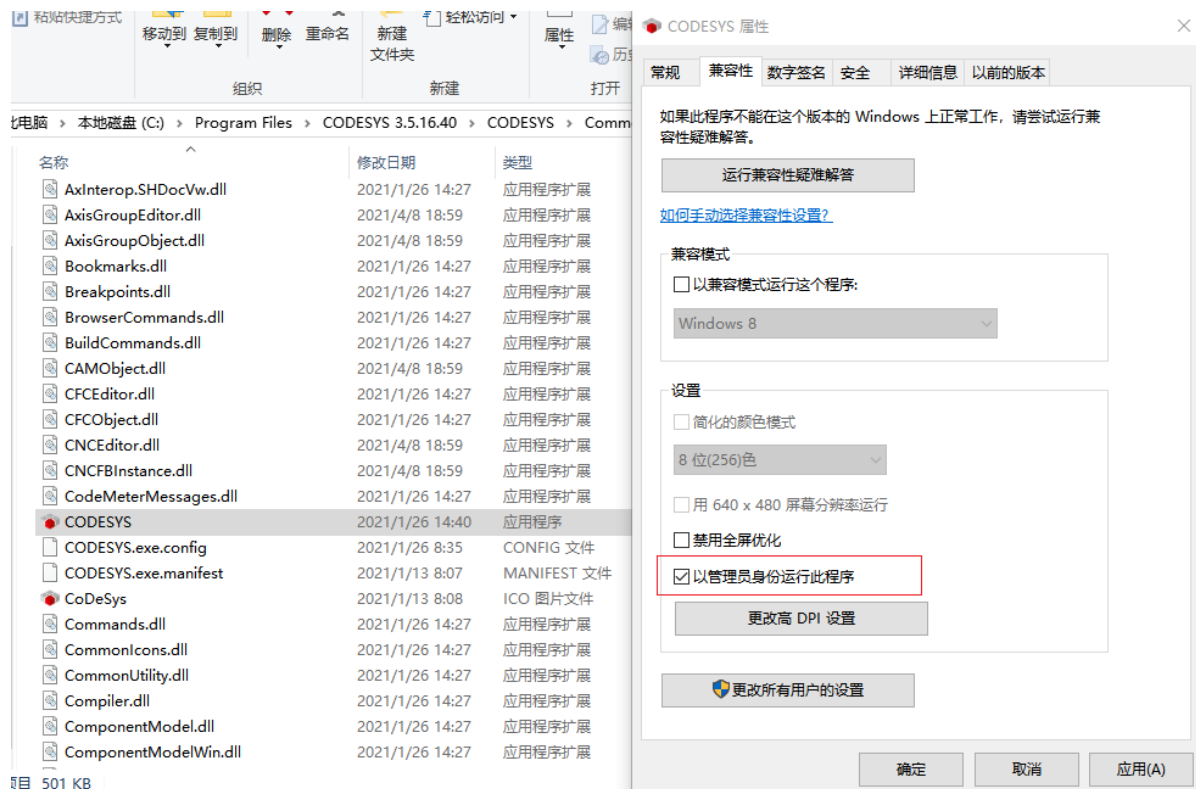
5. Codesys 工程举例	45
5-1. 基本编程操作	46
5-2. I/O 映射	48
5-3. 任务配置	49
5-4. 程序下载/读取	51
5-4-1. 编译	51
5-4-2. 登录下载	52
5-4-3. 源代码下载	53
5-4-4. 读取程序	53
5-5. 程序调试	54
5-5-1. 复位	54
5-5-2. 程序调试	54
5-6. 仿真	56
5-7. PLC 脚本功能	56

5-1. 基本编程操作

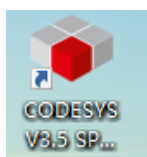
1、启动 Codesys

1) 设置管理员权限

在 Win7 系统下需要以管理员权限打开软件，在 Codesys 的默认安装路径下找到 Codesys.EXE 文件，选中该文件后点击鼠标右键，选择属性，将“Run This Program as an administrator”或者是“以管理员身份运行此程序”的勾选上，点击“OK”确认，如图所示，确认后每次运行 Codesys 系统则会默认以管理员权限自动进入 Codesys。



2) 启动 Codesys



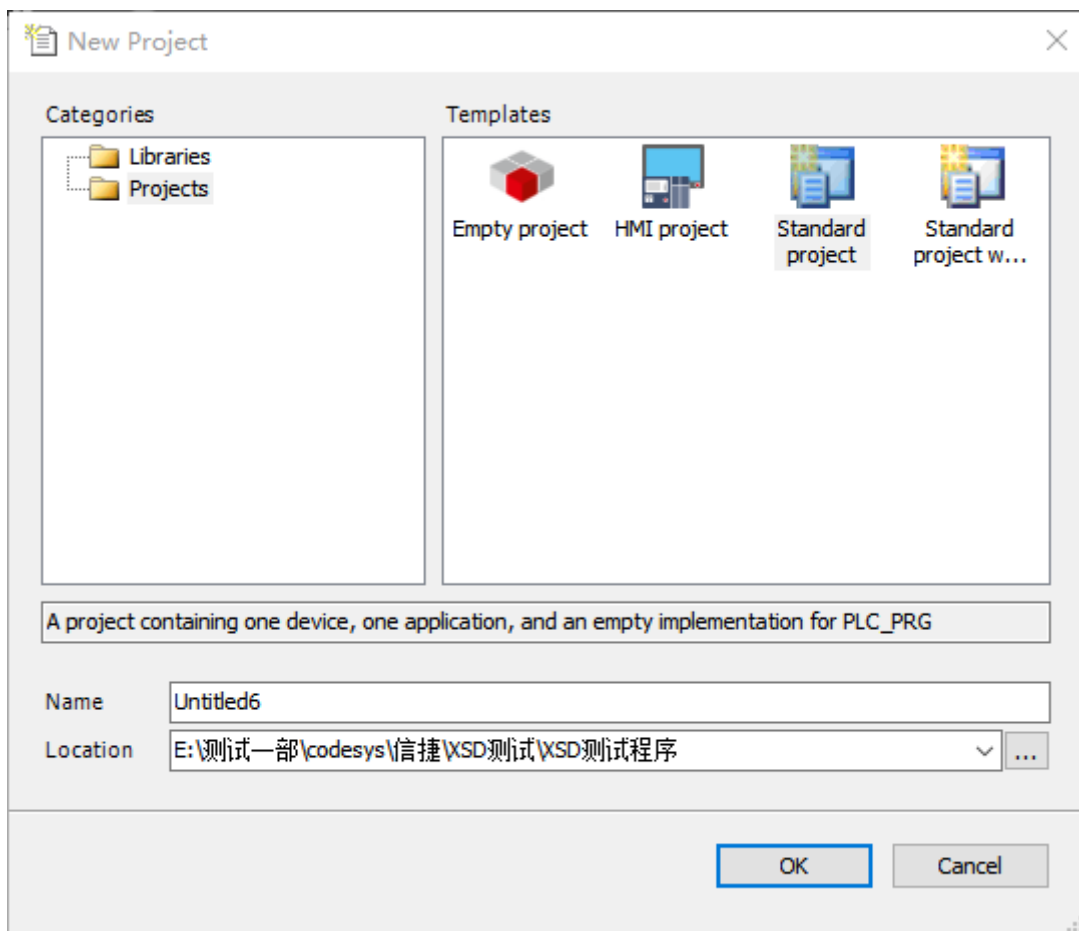
从开始菜单选择> Codesys > Codesys V 3.5 或直接双击桌面上的图标启动 Codesys。

3) 创建工程项目

在文件菜单中选择新建工程命令，创建一个新的工程，如图所示。

(1) 选择项目

用户可以新建工程，空工程或者标准工程。并为工程文件输入名称及路径，点击“确定（OK）”。

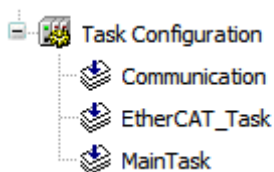


2、PLC 程序文件的建立

PLC 程序文件的建立，是运行结构的运行顺序的建立，也是编程模式的建立，甚至包括了数据区域的分割。在建立程序文件之前，应当对运行结构进行详细的划分，确定连续型、周期型和事件触发型任务，并对周期型和事件触发型任务安排优先级别。新建一个 Codesys 项目后，自动生成一个默认连续性任务，任务下有一个默认的程序及 PLC_PRG。

1) 创建任务

首先，在“任务配置”中管理任务，通常的项目应用中，可以分为主逻辑任务、通讯任务，通讯由于要更新数据源，会将它放在比较高的任务优先等级及较短的循环时间。此外，如项目中涉及到运动控制，也会将其独立出一个任务，并将其放在最高的任务优先等级，如图所示：

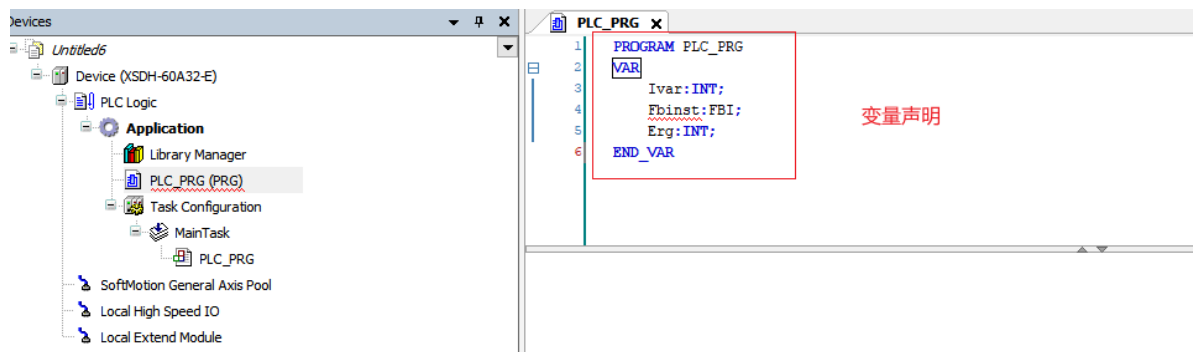


2) POU 的建立

选中“Application”-->右键选择“添加对象”，选择“程序组织单元”进行 POU 的添加。

(1) 声明变量

在设备视窗中，默认 POU 为“PLC_PRG”，双击设备树中“PLC_PRG”，自动在 Codesys 用户界面中部的 ST 语言编辑器中打开。语言编辑器包含声明部分（上部），实现部分（下部），由一个可调的分割线分开。声明部分包括：显示在左侧边框中的行号、POU 类型和名称（如“PROGRAM PLC_PRG”），以及在关键字“VAR”和“END_VAR”之间的变量声明，如下图所示。在编辑器的声明部分，将光标移到 VAR 后，点击回车。插入新的空行，声明 INT 类型变量“Ivar”，INT 类型变量“Erg”，FB1 类型变量“Fbinst”。



(2) 输入程序

在声明区域下的程序编辑区域输入如下代码：

```
1   Ivar:=Ivar+1;
2   Fbinst(in:=11,out =>Erg);
```

(3) 自定义函数/功能块

在变量声明区可以看到，调用了功能块“FB1”，但“FB1”并不是标准功能块，故需要自定义功能块。从工程菜单中选择“添加对象”命令。在“添加对象”对话框左侧选择“程序组织单元”，输入名称：FB1，在类型选项中激活“功能块(B)”选项。实现语言选择“结构化文本(ST)”。点击“打开”按钮确认对象设置。

用于新功能块 FB1 的编辑窗口打开。与 PLC_PRG 的变量声明一样，在此对以下变量声明：

```
FUNCTION_BLOCK FBI
VAR_INPUT
    in:INT;
END_VAR
VAR_OUTPUT
    out:INT;
END_VAR
VAR
    ivar:INT:=2;
END_VAR
```

在程序编辑器实现部分输入以下内容：

```
out:=in+ivar;
```

功能作用是在输入变量“in”的基础上上加“2”赋值给输出“out”。

5-2. I/O 映射

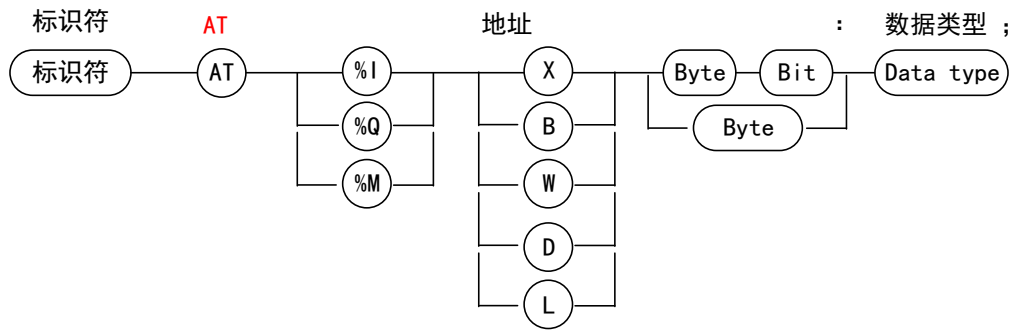
在 Codesys 应用中，当需要和可编程逻辑控制器的 I/O 模块进行变量映射或和外部设备进行网络通讯时，可采取两种方式：

- ① 将 POU 里定义的参数绑定到变量上；
- ② 使用关键字 AT 把变量直接联结到确定地址，直接变量须符合以下规则：

```
AT<地址>:
<标识符> AT <地址>:<数据类型> {:=<初始化值>};
```

{ } 中为可选部分。

使用“%”开始，随后是位置前缀符号和大小前缀符号，如果有分级，则用整数表示分级，并用小数点符号“.”表示，如%IX0.0，%QW0。直接变量声明的具体格式如下图所示：

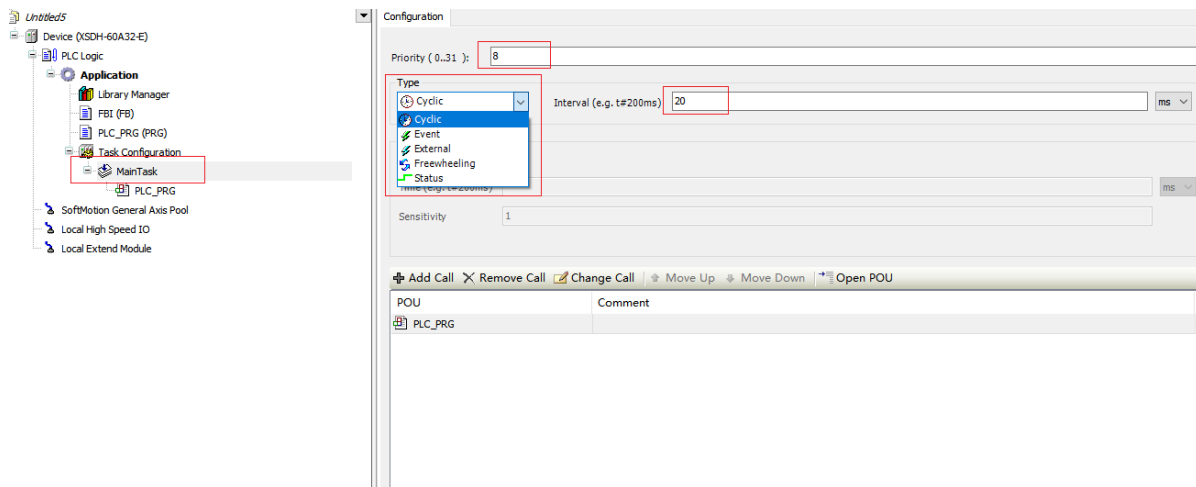


5-3. 任务配置

1、概述

一个程序可以用不同的编程语言来编写。典型的程序由许多互连的功能块组成，各功能块之间可互相交换数据。在一个程序中不同部分的执行通过“任务”来控制。“任务”被配置以后，可以使一系列程序或功能块周期性地执行或由一个特定的事件触发开始执行程序。在设备树中有“任务管理器”选项卡，使用它除了声明特定的 PLC_PRG 程序外，还可以控制工程内其他子程序的执行处理。任务是用于规定程序组织单元在运行时的属性，它是一个执行控制元素，具有调用的能力。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，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。

在任务配置中，用名称、优先级和任务的启动类型来定义它。这启动类型可以通过时间（周期的，随机的）或通过内部或外部的触发任务时间来定义，例如使用一个布尔型全局变量的上升沿或系统中的某一特定事件。对每个任务，可以设定一串由任务启动的程序。如果在当前周期内执行此任务，那么这些程序会在一个周期的长度内被处理。优先权和条件的结合将决定任务执行的时序，任务设置界面如下图所示：



由于 Codesys V3.x 在任务配置时有如下的属性，编程者需遵循如下规则：

- ◆ 循环任务的最大数为 100。
- ◆ 自由运行任务的最大数为 100。
- ◆ 事件触发任务的最大数为 100。
- ◆ 根据目标系统, PLC_PRG 可能会在任何情况下作为一个自由程序执行，而不用插入任务配置中。
- ◆ 处理和调用程序是根据任务编辑器内自上而下的顺序所执行的。

1) 任务优先级 Priority

Codesys 中的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。

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预知。

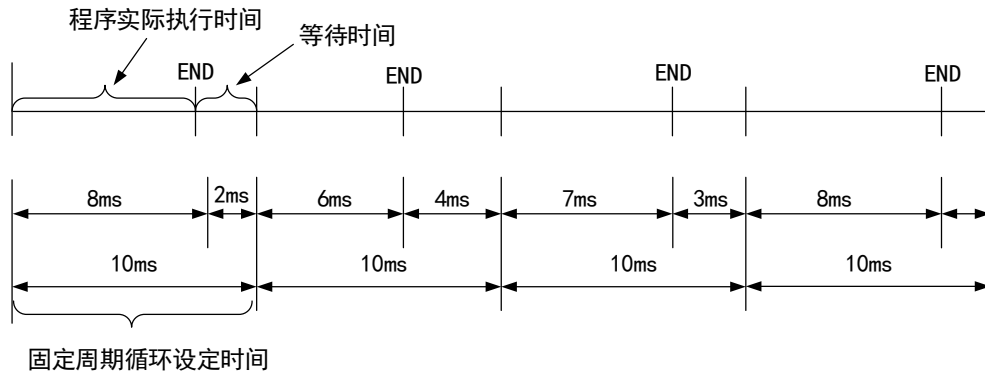
2) 任务的执行类型

针对每个独立的任务可以对其进行执行的类型编辑及配置。包括固定周期循环、事件触发、外部触发、自由运行和状态触发 5 种类型。

(1) 固定周期循环

根据程序中所使用的指令执行与否，程序的处理时间会有所不同，所以实际执行时间在每个扫描周期都发生不同的变化，执行时间有长有短。通过使用固定周期循环方式，能保持一定的循环时间反复执行程序。即使程序的执行时间发生变化，也可以保持一定的刷新闻隔时间。在这里，也推荐大家优先选择固定周期循环任务启动方式。

例如，假设将程序对应的任务设定为固定周期循环方式，间隔时间设定为 10ms 时，实际程序执行的时序图如下图所示。

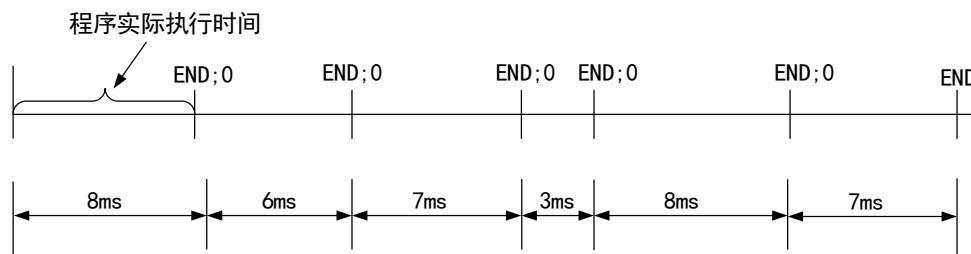


如果程序实际执行时间在规定的固定周期循环设定时间内执行完，则空余时间用作等待。如应用中还有优先级较低的任务未被执行，则剩下的等待时间用来执行低优先级的任务。

(2) 自由运行

程序一开始运行任务就被处理，一个运行周期结束后任务将在下一个循环中被自动重新启动。

不受程序扫描周期（间隔时间）的影响。即确保每次执行完程序的最后一条指令后才进入下一个循环周期。否则不会结束该程序周期。



该执行方式因为没有固定的任务时间，所以每次执行的时间可能都不一样。故不能保证程序的实时性，在实际的应用中选用此方式的场合较少。

(3) 事件触发

如果事件区域的变量得到一个上升沿，任务开始。

(4) 状态触发

如果事件区域的变量为 TRUE，任务开始。

下图中针对事件触发和状态触发分别进行了比较，绿色实线为两种触发方式选择的布尔变量状态，下表为比较的结果。



状态触发方式与事件触发功能类似，区别在于状态触发的触发变量只要为 TRUE 程序就执行，为 FALSE 则不执行。而事件触发只采集触发变量的上升沿有效信号。

在采样点 1-4（紫色）不同类型的任务展示了不同的反应。这个具体的事件为 TRUE 完成了状态驱动任务的条件，然而一个事件驱动任务需要事件从 FALSE 变为 TRUE。如果任务计划的采样频率过低，事件的上升沿可能检测不到。

执行点	1	2	3	4
事件触发 (Event)	不执行	执行	执行	执行
状态触发 (Status)	不执行	执行	不执行	不执行

(5) 外部中断

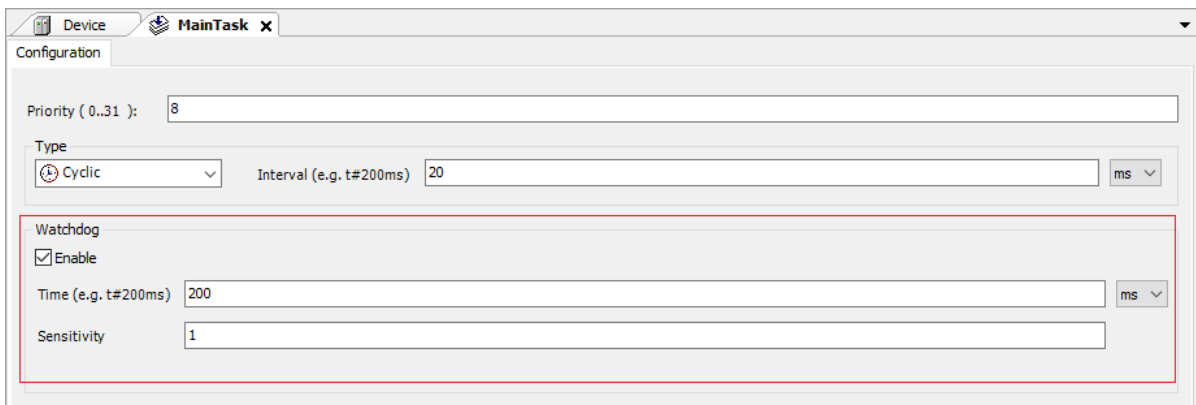
如果事件区域的变量得到一个外部中断信号 X 的上升沿或者下降沿，任务开始。

输入端子 X 可以作为外部中断的输入用，每一输入端对应于一个外部中断，输入的上升沿或者下降沿都可触发中断。

(6) 看门狗

看门狗是一种控制器硬件式的计时设备，Codesys 内可以通过“任务配置”对其进行使能，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。我们可以形象的将它理解为一只有主人定时的去给它喂食，如果超过规定的时间没有喂，则他马上就会饿。要配置看门狗，必须定义两个参数，时间和灵敏度，看门狗的配置如图所示。



① 时间

Codesys 针对每个任务可以配置独立的看门狗。如果目标硬件支持长看门狗时间设置，则可以设置上限和下限。默认的看门狗时间单位为毫秒（ms）。如果程序执行周期超过看门狗触发时间，将激活看门狗功能，并将中止当前任务。

② 灵敏度

“灵敏度”用于定义必须在控制器检测到应用程序错误之前发生的任务看门狗例外数。默认为 1。

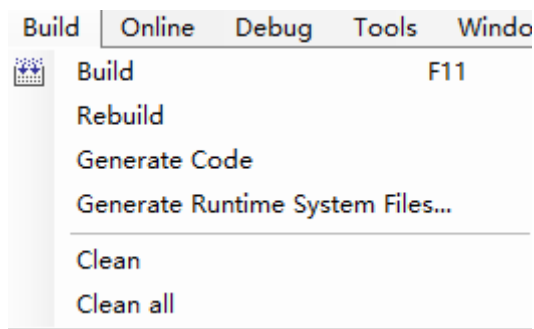
最终的看门狗触发时间=时间×灵敏度。如果程序实际执行时间超过看门狗触发时间，则激活看门狗。例如，时间为 10ms，灵敏度设为 5，则看门狗触发时间为 50ms，只要任务的执行时间超过 50ms，则立即激活看门狗并将任务中止。

5-4. 程序下载/读取

5-4-1. 编译

程序编写完成后，在下载之前需要对程序进行编译。编译命令对编写的程序进行语法检查，并且只编译添加到任务中的程序。如果创建的 POU 没有添加到任务中，编译命令不对该 POU 进行语法检查。

编译指令不生成任何代码，只针对 POU 的语法进行检查。直接执行设备登录命令，系统也将默认执行编译指令（等同于先手动执行编译命令），再编译检查没有语法错误后执行连接登录指令。同样，在编译中不对没有添加到任务中的 POU 进行语法检查。执行登录命令同时会生成代码。



- 1) 编译：对当前的应用进行编译。
- 2) 重新编译：如果需要对已经编译过的应用再次编译，可以通过重新编译进行。
- 3) 生成代码：执行此命令后生成当前应用的机器代码，执行登录命令时，生成代码默认执行。
- 4) 清除：删除当前应用的编译信息，如果再次登录设备时需要重新生成编译信息。
- 5) 清除全部：删除工程中所有的编译信息。

执行编译命令后可以看到，添加到任务里面的“PLC_PRG”显示为蓝色，没有添加到任务里面的则显示为灰色。编译指令不会对灰色的 POU 进行语法检查，因为该程序单元没有处于活动状态，编译指令只针对于处于活动状态的 POU 进行语法检查。如果在编译的过程中发现需要运行的程序单元显示为灰色，可以检查该程序单元是否已经被成功的添加到了所需要运行的任务当中。

编译命令执行完成之后可以在消息栏看到编译生成的信息，其中可以看到编译的程序是否有错误或者警告，以及错误和警告的数量。如果有错误和警告产生可以通过消息窗口进行查看和查找，根据提示信息对程序进行修改。

5-4-2. 登录下载

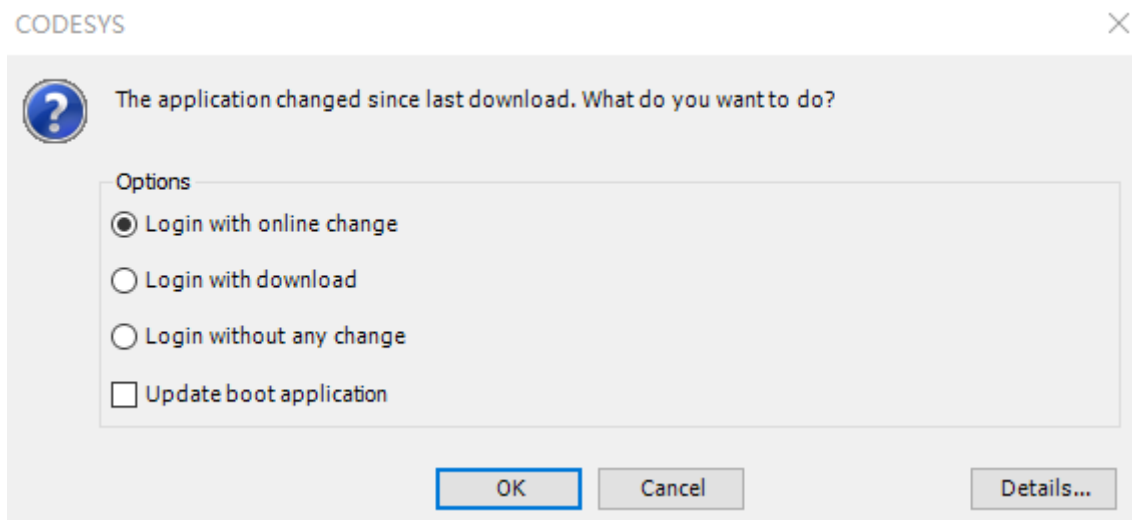
1、登录

登陆使应用程序与目标设备建立起连接，并进入在线状态。能正确登陆的前提条件是要正确配置设备的通讯设置并且应用程序必须是无编译错误的。

对于以当前活动应用登录，生成的代码必须没有错误和设备通信设置必须配置正确。登入后，系统会自动选择程序下载。

2、下载

下载命令，在在线模式下有效。它包括对当前应用程序的编译和生成目标代码两部分。除了语法检查（编译处理）外，还生成应用目标代码并装载到 PLC。



1) 登录-在线下载 (Login with online change)

用户选择此选项后，项目的更改部分被装载到控制器中。使用“在线修改后登入”操作，可以防止控制器进入 STOP 状态。

注意：

① 用户之前至少执行过一次完整的下载。

② 指针数据会更新最近一个周期的数值，如果改变了原变量的数据类型，无法确保数据的准确性，此时，需要重新分配指针数据。

2) 登录并下载 (Login with download)

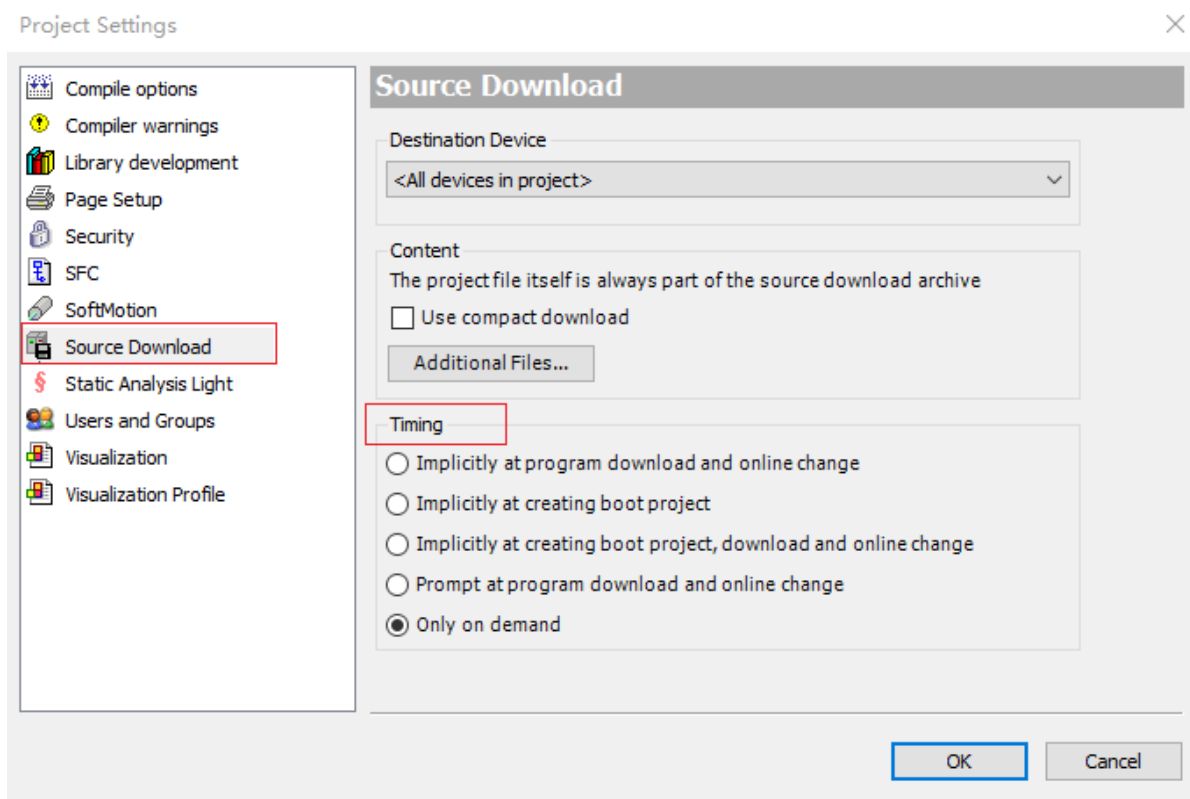
选择“登入并下载”后，将整个项目重新装载到控制器中。与“在线修改后登入”最大的区别是当完成下载后，控制器会停留在 STOP 模式，等待用户发送 RUN 指令，或重启控制器程序才会运行。

3) 登录-不做任何修改

登录时，不更改上次装载到控制器中的程序。

5-4-3. 源代码下载

Codesys 出于对程序员源代码的保护，下载时默认不自动下载源代码，如需下载源代码则需要进行手动设置，点击“在线” --> “下载源代码到连接设备上”。用户也可以在“工程” --> “工程设置” --> “下载源代码” --> “计时”选项中对该属性进行设置。



5-4-4. 读取程序

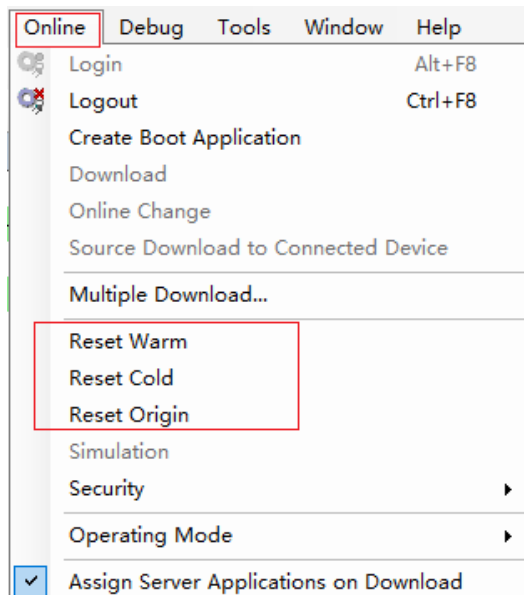
在“文件” > “源上传”命令打开一个设备选择对话框，用户选择连接 PLC 的网络路径，点击“确定”按钮。如果用户选择的路径下已经存在存档文件，给出是否覆盖提示。

这里需要注意的是，在读取程序之前需要确保在前一次的下载过程中已经做过“下载源代码到连接设备上”。否则不能读出控制器中的数据。

5-5. 程序调试

5-5-1. 复位

Codesys 程序复位有如下三种方式，在“在线”菜单中进行选择。



1、热复位

热复位后，除了保持型变量（PERSISTENT 和 RETAIN 变量）外，其它当前的应用的变量都被重新初始化。如果设置了初始值的变量，热复位后这些变量值还原为设定的初始值，否则变量都会被置为标准初始值 0。

2、冷复位

与“热复位”不同的是，冷复位命令不但将普通变量的值设置为当前活动应用程序的初始值，而且将保持型变量（RETAIN 变量）的值也设置为初始值 0。保持型变量（PERSISTENT）保持不变。

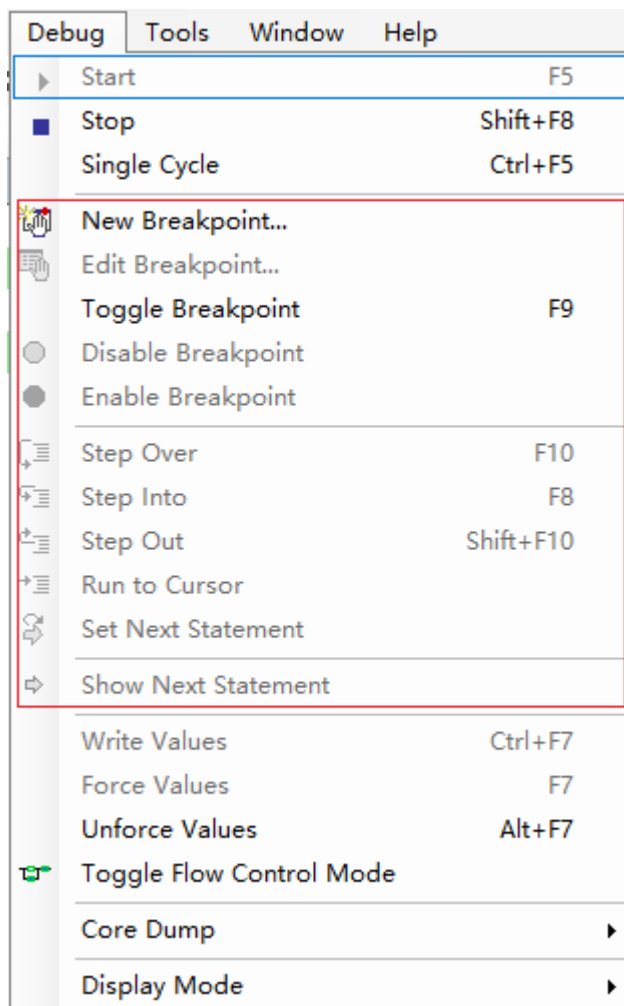
3、初始化复位

当在设备树中选择一个可编程设备时，无论是在离线还是在在线状态下都可以使用此命令。使用此命令将使设备复位到初始状态，即设备中的任何应用、引导工程和剩余变量都将被清除。

由于所有工程信息被清除，重新登入后，需要重新“下载”程序，并“启动”运行。

5-5-2. 程序调试

在 Codesys 中“调试”菜单的视图如图所示。主要的操作涉及断点设置及单循环。



1、断点

断点是程序内处理停止的功能，当程序停止后，程序研发人员可以借此观察程序到断点位置时其变量及 I/O 等相关变量的内容，有助于深入了解程序运行的机制，发现及排除程序故障。

在 Codesys 中所有的编程语言都可以设置断点。在文本编辑器 ST 语言中，断点设置在行上；在 FBD 和 LD 编辑器中设置在网络号上；而在 SFC 中，设置在步上。

2、单步执行

设置断点后，可以进行单步执行程序，该功能可以让程序一步一步的运行，方便编程人员进行调试，以便检查程序中的逻辑错误。

1) 跳过

该命令会执行程序中当前的一条指令，执行完停止。在不调用 POU 时，跳过和跳入命令效果是一样的。但如果是调用 POU，那么跳过不会进入这个 POU，而是把这个 POU 调用当作完整的一步，一次执行完；跳入则会进入该 POU。如果用的是 SFC 语言，那么跳过会把一个动作当作完整的一步，一次执行完。如果想要实现跳到被调用的 POU 中单步调试，就必须用跳入。

2) 跳入

执行时，当前的指令位置由一个黄色箭头表示。如果当前指令没有调用 POU，那么使用该命令和使用跳过命令的效果一样。

3) 跳出

当正在一个 POU 中单步调试时，用跳出会把这个 POU 剩下的指令一次执行完，然后返回到这个 POU 被调用处的下一条指令。所以，如果是一层一层向下调用 POU，那么跳出就会一层一层向上返回，每次返回一层。如果程序中不包含任何 POU 调用，那么跳出无法向上层返回，就会返回到程序的开头处。

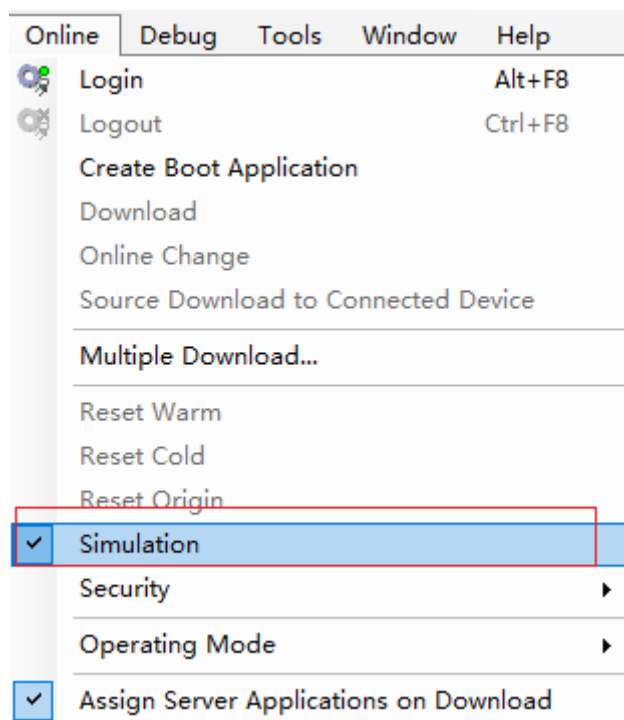
3、单循环

在“调试”→中选择“单循环”，这样程序进行单步运行。即按一次运行，程序执行一个周期即停止并等待下一次运行指令。

5-6. 仿真

离线仿真

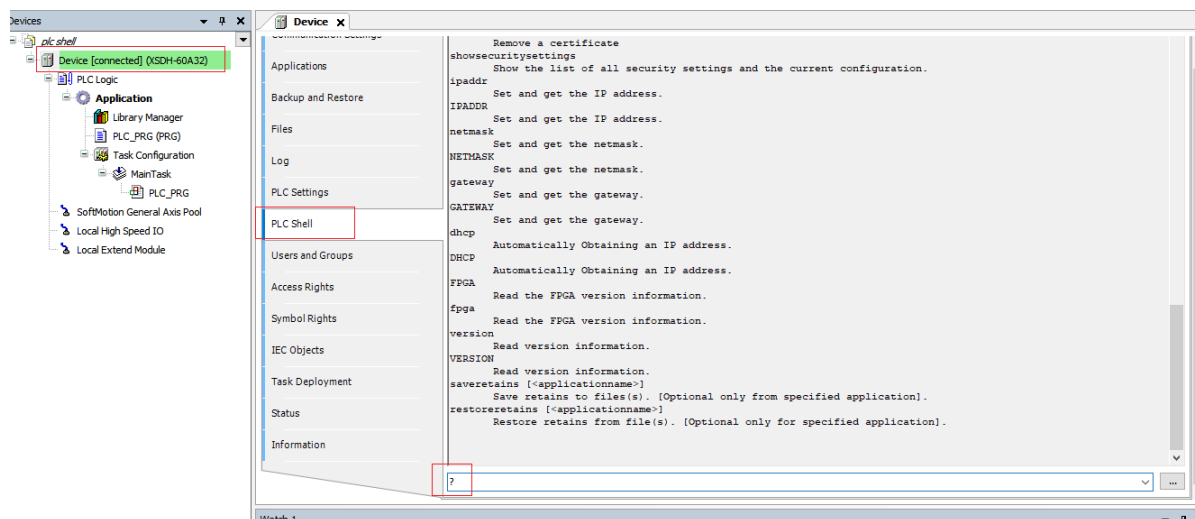
在菜单“在线”中选择“仿真”，便进入了仿真模式下的程序运行过程。确认选项“仿真”前已打上后，编译程序，没有错误后，即可进入仿真模式。



5-7. PLC 脚本功能

该 PLC 脚本是一个基于文本的控制监视（终端）。从控制器中得到的具有特定信息的命令以一个输入行进行输入并且作为一个字符串发送到控制器，返回相关的字符串在浏览窗口中的结果显示，这个功能用于诊断和调试目的。

鼠标双击选中“Device”，在右边视图中找到“PLC shell”，在下方的命令输入框中输入相应指令即可。输入“？”，按回车键即可显示所有该控制器支持的命令。可参考 4-3 节。



注意：如需使用脚本功能必须先登入 PLC 才能使用相应指令。

6. 工业现场总线技术

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

6. 工业现场总线技术	57
6-1. MODBUS 通信	58
6-1-1. MODBUS 通信概述	58
6-1-2. 参数配置	58
6-2. MODBUS TCP	61
6-2-1. MODBUS TCP 通信概述	61
6-2-2. 参数配置	61
6-3. OPC UA	64
6-3-1. OPC UA 通信概述	64
6-3-2. 参数配置	64

6-1. MODBUS 通信

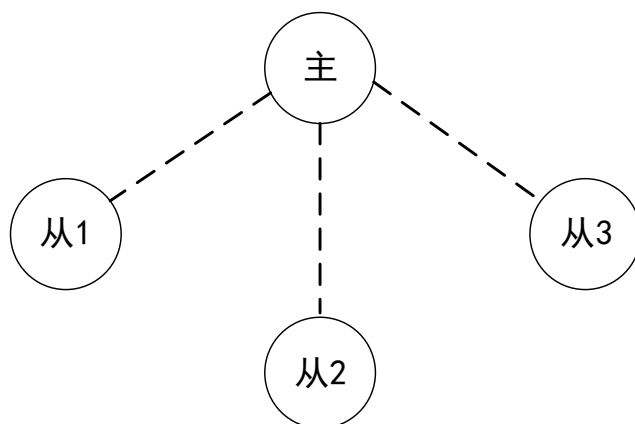
6-1-1. MODBUS 通信概述

XS 系列可编程控制器本体支持 Modbus 协议通讯主、从机形式。

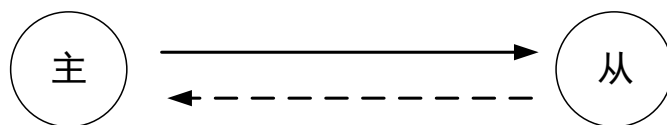
主站形式：可编程控制器作为主站设备时，可与其它使用 Modbus 协议的从机设备通讯；与其他设备进行数据交换。例：信捷 XS 系列 PLC，可以通过通讯来控制变频器。

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。

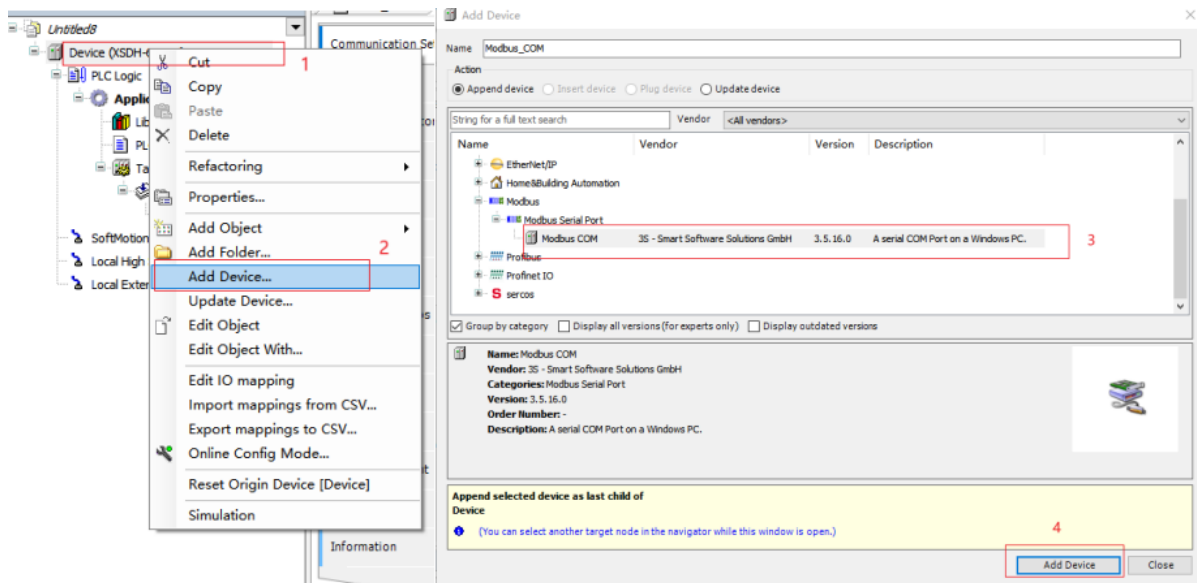


之所以图中有虚线箭头（包括 RS485 网络中），是因为理论上在两个网络中，只要各个 PLC 不发数据，网络中任意 PLC 都可以用来作为主站，其它 PLC 作为从站；但是由于多个 PLC 之间没有一个统一的时钟基准，容易出现在同一时刻有多个 PLC 发送数据，会导致通讯冲突失败，因此不建议这样使用。

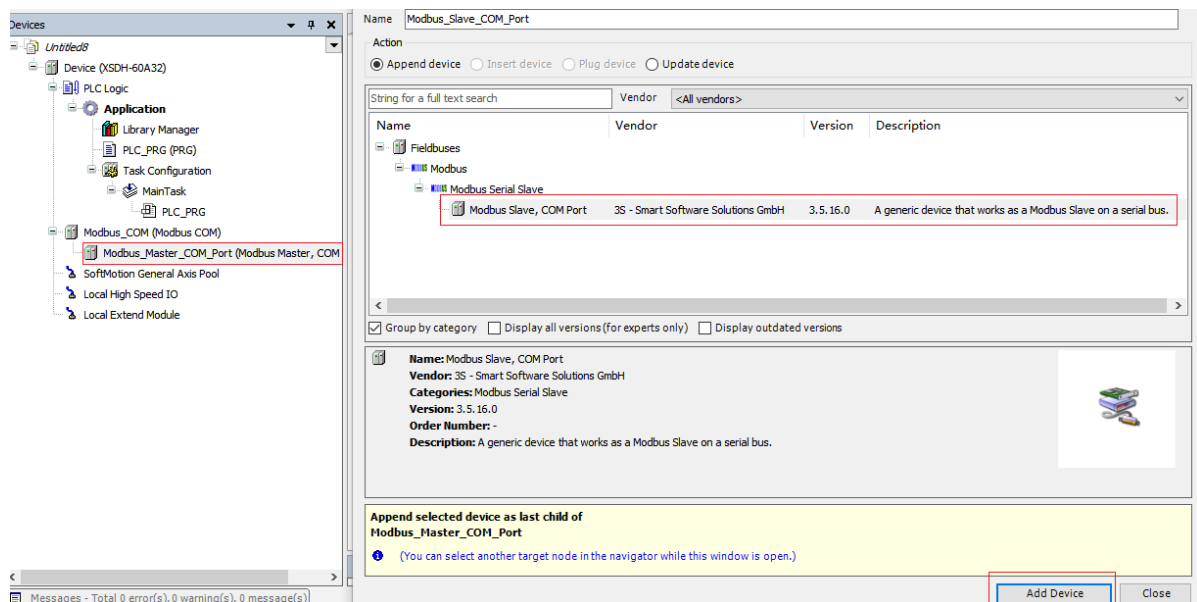
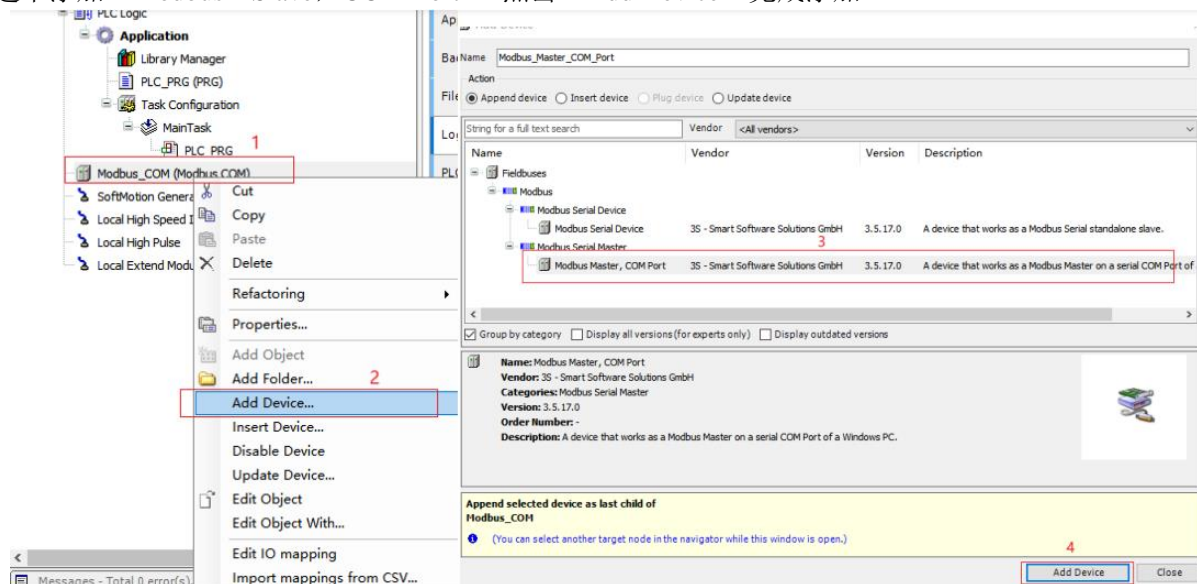
6-1-2. 参数配置

1、Modbus 主站的配置

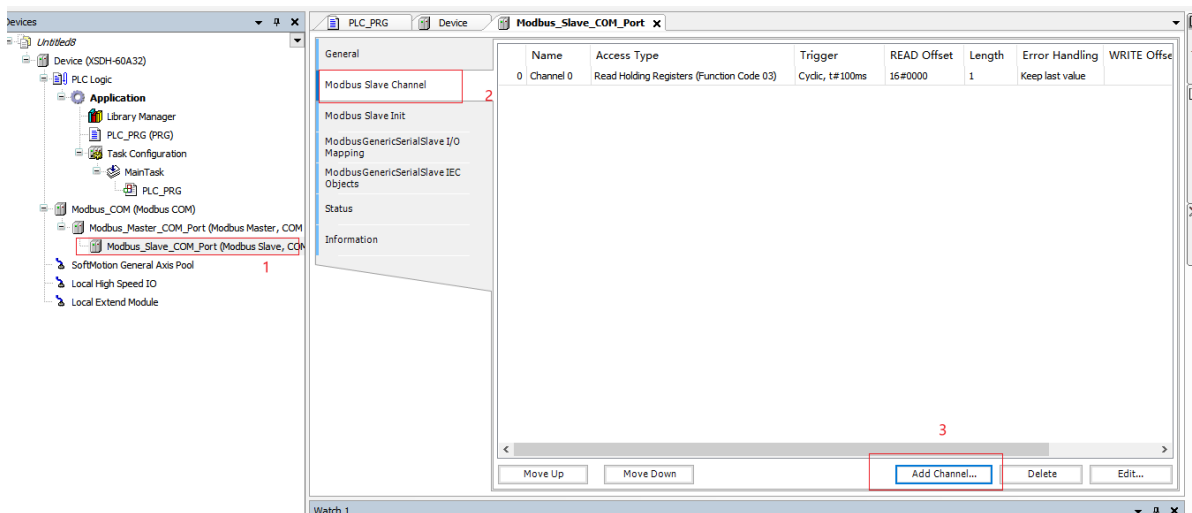
① 在应用的工程中，右击设备“Device”，点击“Add Device”，在弹出的对话框中点击“Modbus COM”进行添加。



② 添加成功后可看到设备下的“Modbus COM”，点击“Add Device”，在弹出的对话框中选中“Modbus Master, COM Port”，点击“Add Device”进行添加。选中“Modbus Master, COM Port”，在弹出的对话框中选中添加“Modbus Slave, COM Port”” 点击“Add Device”完成添加。

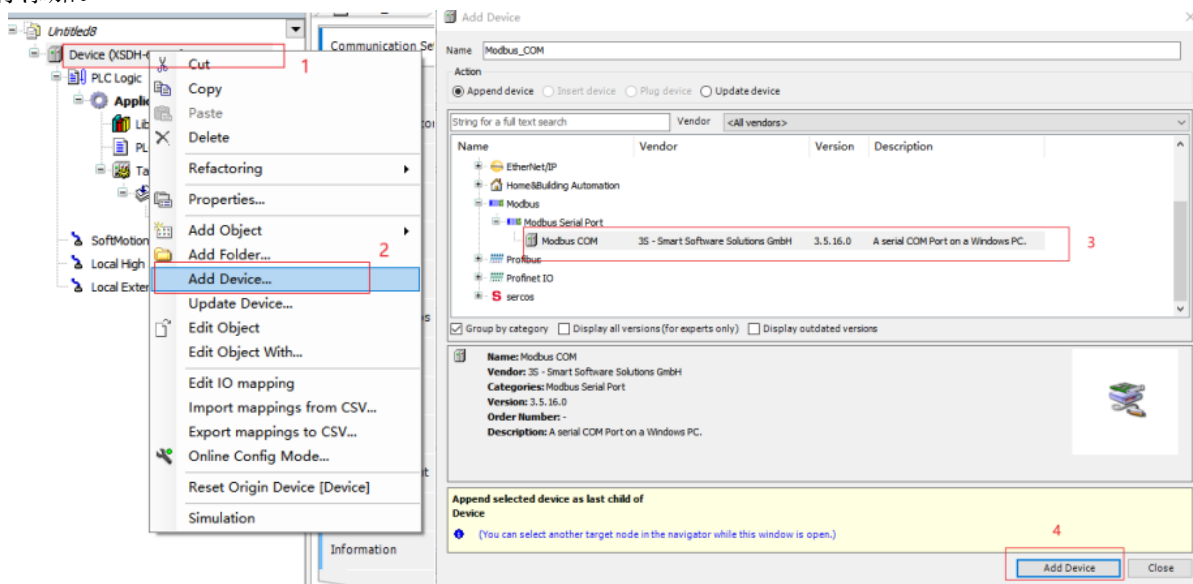


③ 添加后可在左侧设备栏中看到对 Modbus COM 主站的添加，双击“Modbus_Slave_COM_Port”，在右侧“Modbus Slave Channel”里进行读写的配置。

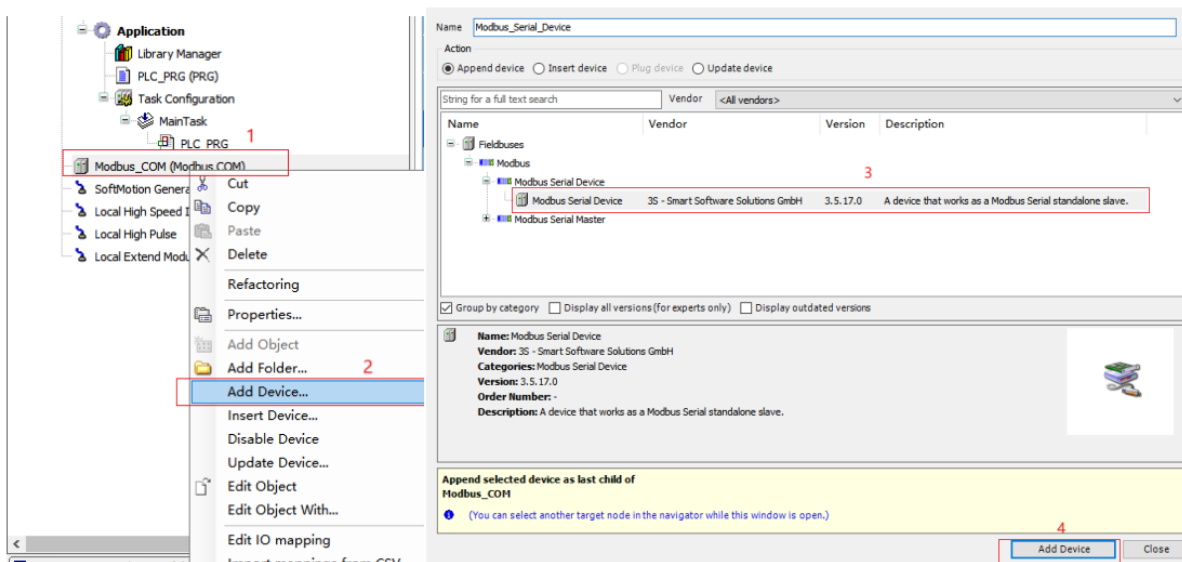


2、Modbus 从站的配置

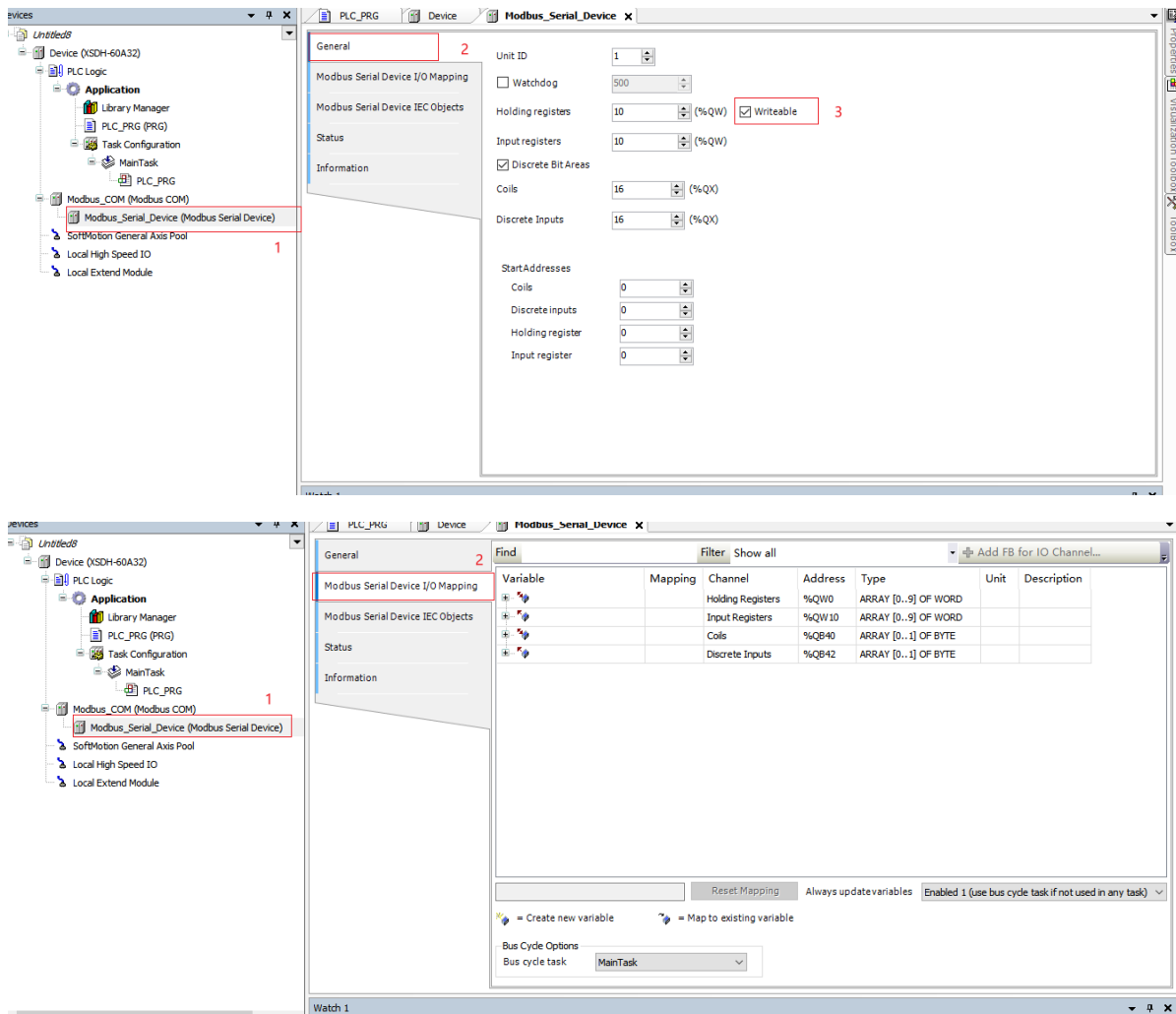
① 在应用的工程中，右击设备“Device”，点击“Add Device”，在弹出的对话框中点击“Modbus COM”进行添加。



② 添加成功后可看到设备下的“Modbus COM”，右击“Add Device”，在弹出的对话框中选中添加“Modbus Serial Device”。如下图所示：



③ 添加后可在左侧设备栏中看到对 Modbus COM 从站的添加，双击“Modbus Serial Device”，在右侧“General”里可配置寄存器和线圈，配置后在“Modbus Serial Device I/O Mapping”里可监控主站对 XS 从站的读写数据。



6-2. MODBUS TCP

6-2-1. MODBUS TCP 通信概述

MODBUS TCP 使用 TCP/IP 在站点间传送 MODBUS 报文，MODBUS TCP 结合了 TCP/IP 协议以及 MODBUS 协议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。与传统的串口方式，MODBUS TCP 插入一个标准的 MODBUS 报文到 TCP 报文中，不再带有数据校验和地址。

XS 系列可编程控制器本体支持 Modbus TCP 协议通讯主、从机形式。

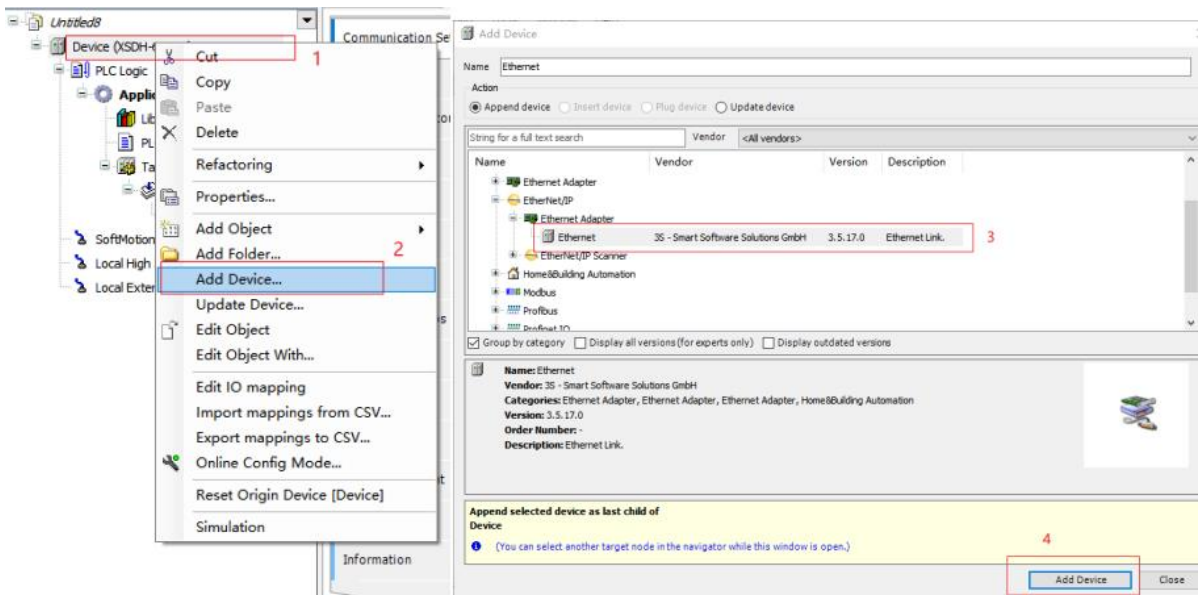
主站形式：可编程控制器作为主站设备时，可与其它使用 Modbus TCP 协议的从机设备通讯。一个主站最多可以连接 64 个从站。

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。

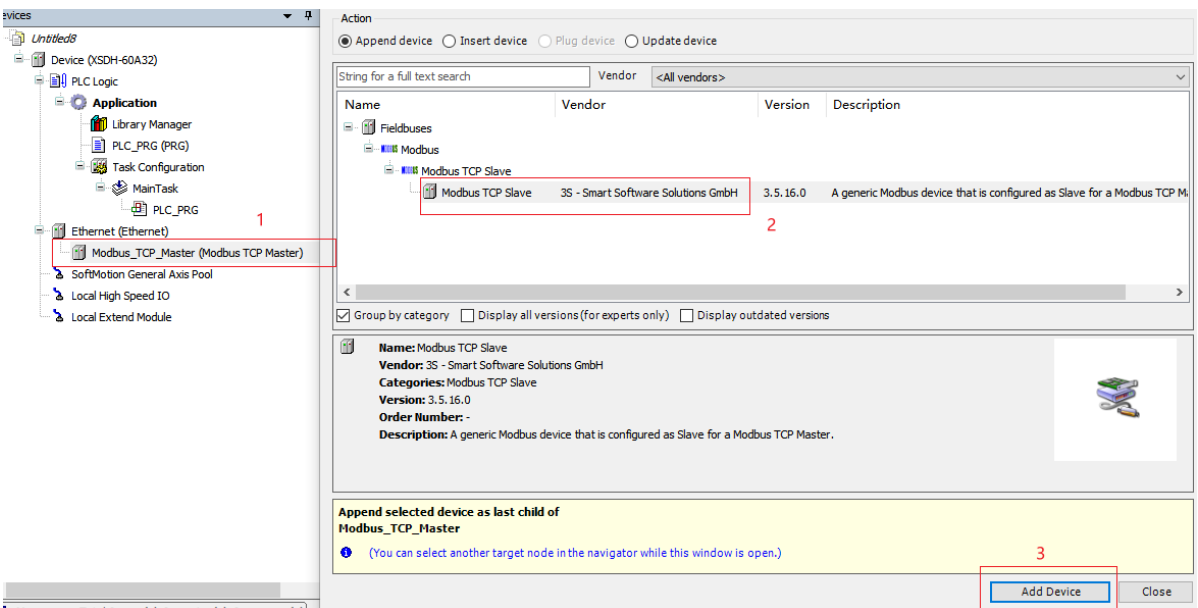
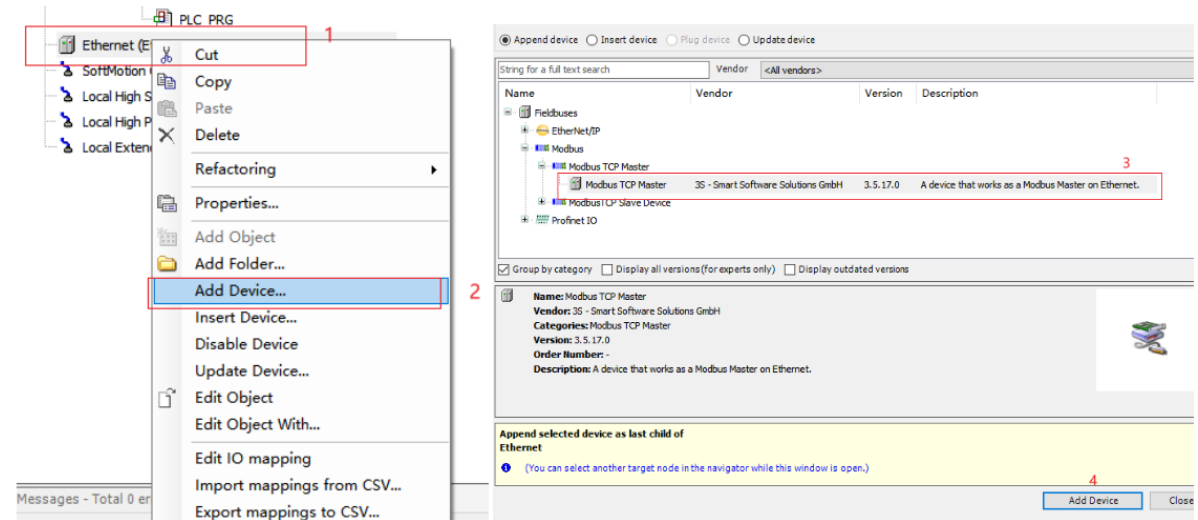
6-2-2. 参数配置

1、Modbus Tcp 客户端的配置

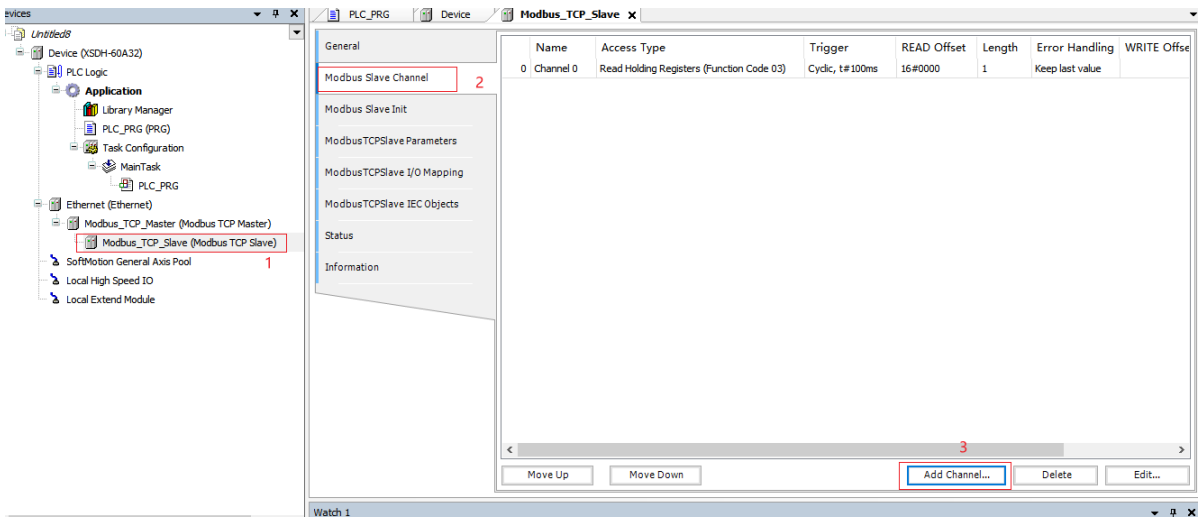
① 在应用的工程中，右击设备“Device”，点击“Add Device”，在弹出的对话框中点击“Ethernet”，如图所示：



② 添加成功后可看到设备下的“Ethernet”，右击“Ethernet”，点击“Add Device”，在弹出的对话框中选中“Modbus Tcp Master”，点击“Add Device”进行添加。选中“Modbus Tcp Master”，在弹出的对话框中选中添加“Modbus Tcp Slave”点击“Add Device”完成添加。

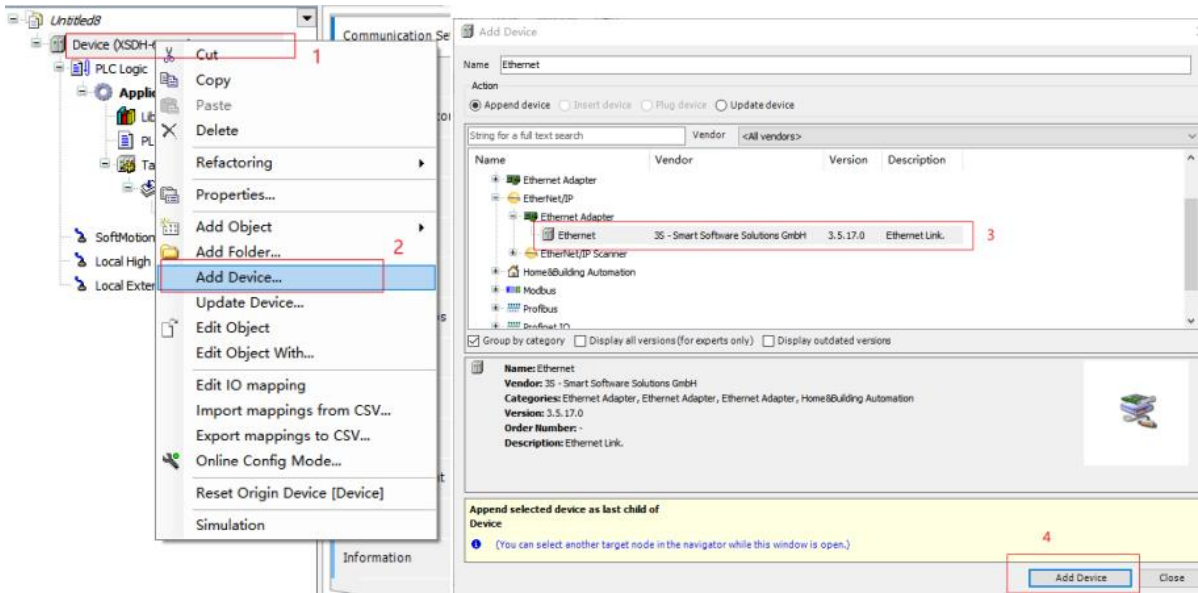


③ 完成添加后可在左侧设备栏中看到对 Modbus Tcp 客户端的添加，双击“Modbus_TCP_Slave”，在右侧“Modbus Slave Channel”里进行读写的配置。

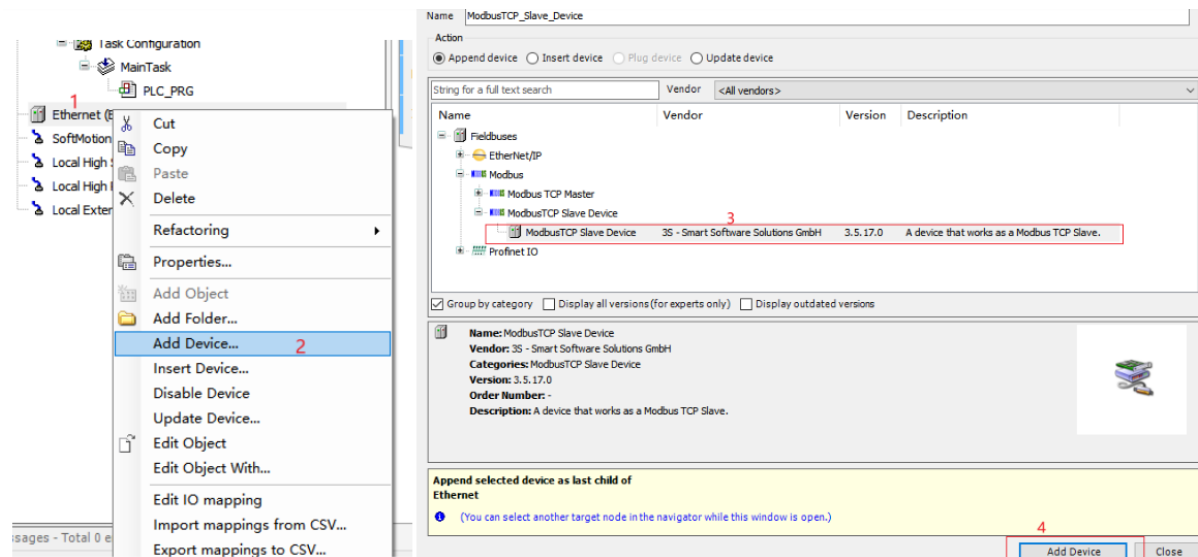


2、Modbus Tcp 服务端的配置

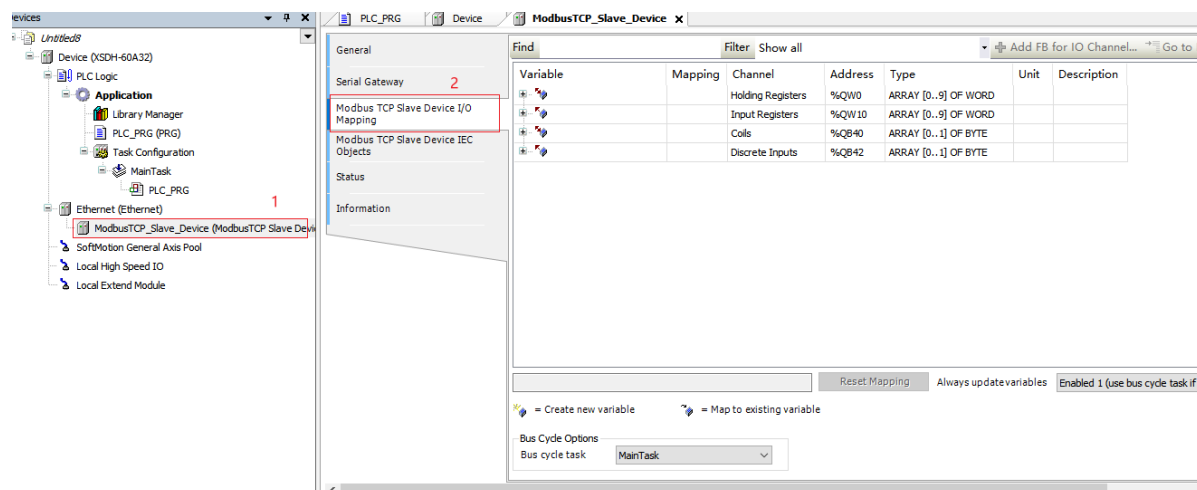
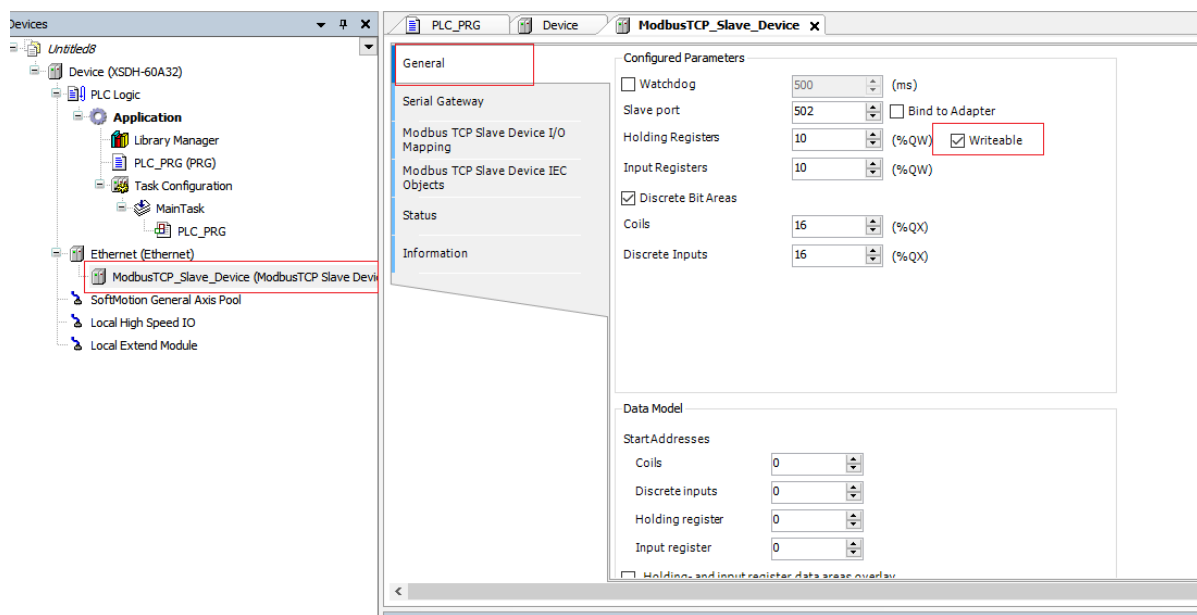
① 在应用的工程中，右击设备“Device”，点击“Add Device”，在弹出的对话框中点击“Ethernet”，如图所示：



② 在添加“Ethernet”后，右击“Add Device”，在弹出的对话框中选中添加“Modbus Tcp Slave Device”。如下图所示：



③完成添加后可在左侧设备栏中看到对 Modbus TcP 服务端的添加，双击“ModbusTCP_Slave_Device”，在右侧“General”里可配置寄存器和线圈，配置后在“Modbus TCP Slave Device I/O Mapping”里可监控客户端对 XS 服务端的读写数据。



6-3. OPC UA

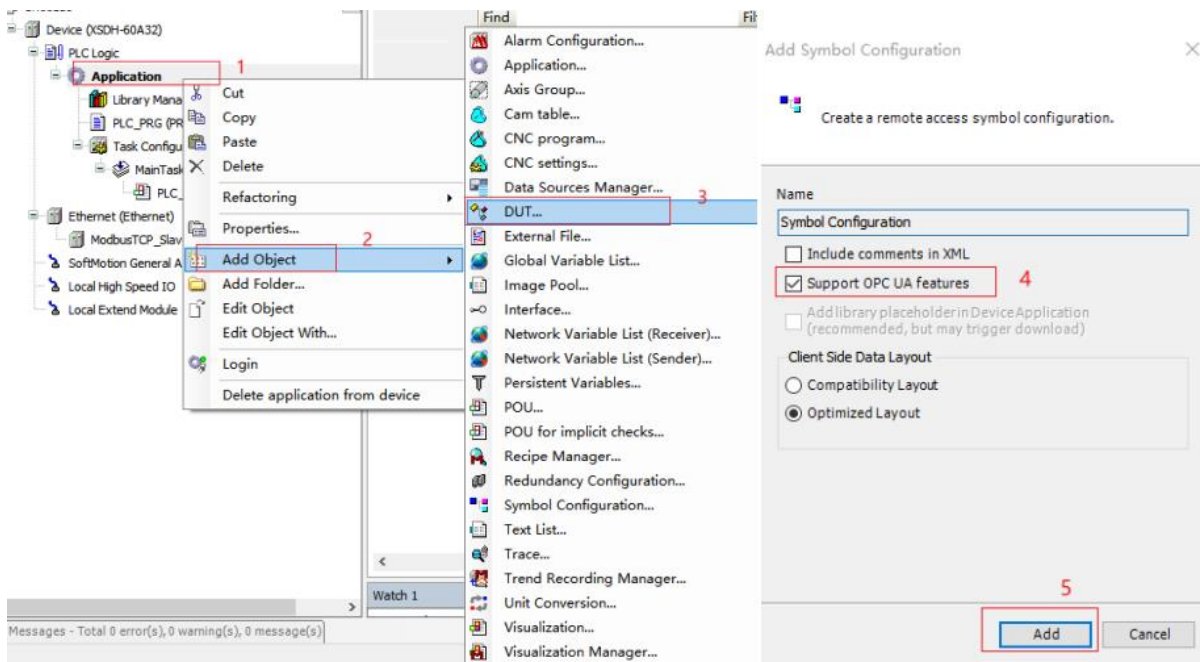
6-3-1. OPC UA 通信概述

OPC UA 于 2008 年发布，是一种于平台无关的面向服务的体系结构，将各个 OPC Classic 规范的所有功能集成到一个可扩展的框架中。

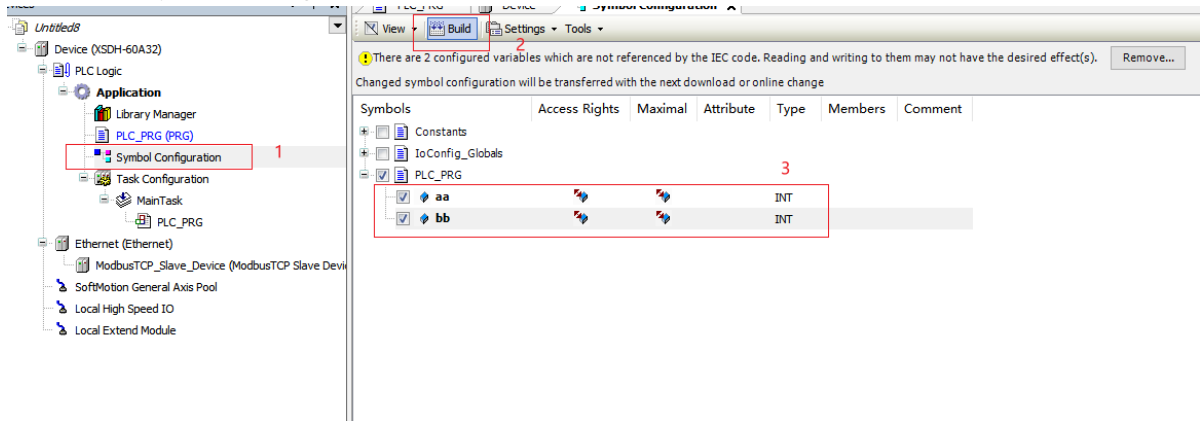
在 XS 系列 PLC 中，集成了 OPC UA 服务器，可以支持用户通过 OPC UA 客户端访问 PLC 中的数据。

6-3-2. 参数配置

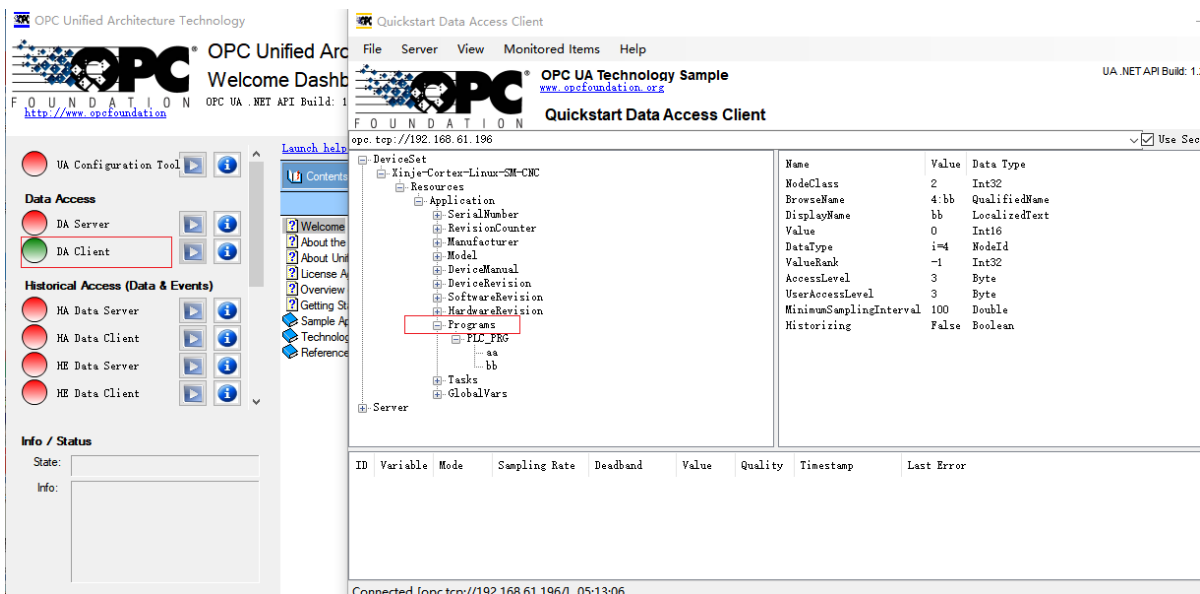
① 在应用的工程中，右击“Application”，选择“Add Object” - “Symbol Configuration..”，在弹出的对话框中勾选“Support OPC UA feature”进行添加，则开启 OPCUA 的功能。

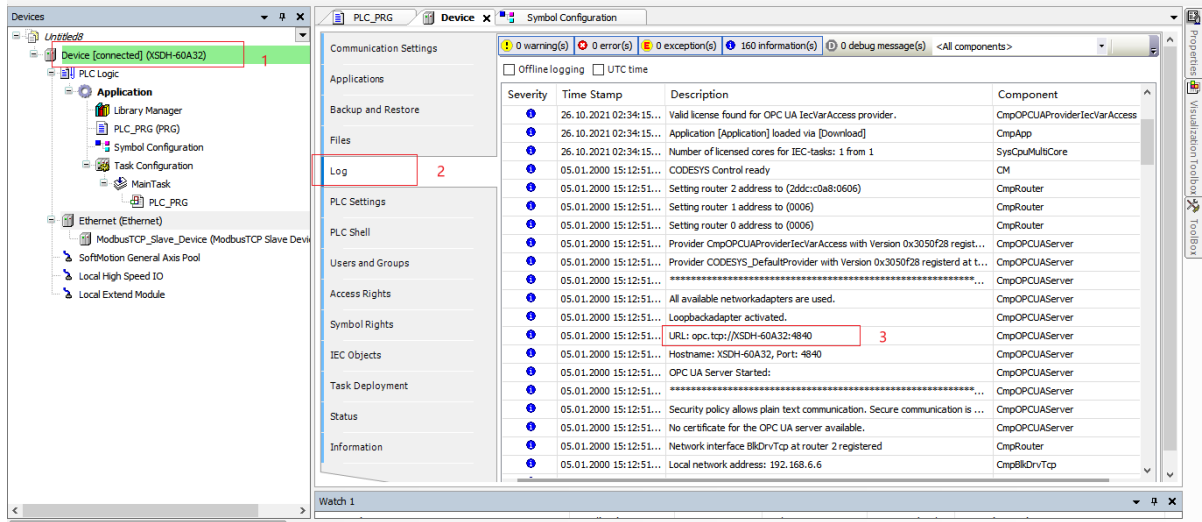


② 双击“Symbol Configuration”，在弹出的界面里点击“Build”后，勾选需要监控的参数。



③ 下载程序后，打开 OPC UA 的小软件，选择“DA Client”，在弹出的界面里输入工控的“IP”地址（比如：opc.tcp://192.168.61.196）或者可输入在“log”里的地址，在“DeviceSet”里找到“programs”。里面就有刚刚勾选的参数。可以在参数上右击--监控--可以读写参数。





7. 常见问题及解决方法

本章主要介绍 Codesys 编程中遇到的常见问题及解决方法。

7. 常见问题及解决方法	67
7-1. Package	68
7-1-1. Package 命名规则	68
7-1-2. Package 的获取	68
7-1-3. Package 的安装	68
7-2. XS 系列 PLC 固件升级	69
7-2-1. 固件命名规则	69
7-2-2. 固件获取	69
7-2-3. 固件安装及其注意事项	69

7-1. Package

7-1-1. Package 命名规则

命名格式: XSDH-60A32_3.5.15.40_1.0.0_P1_20211027

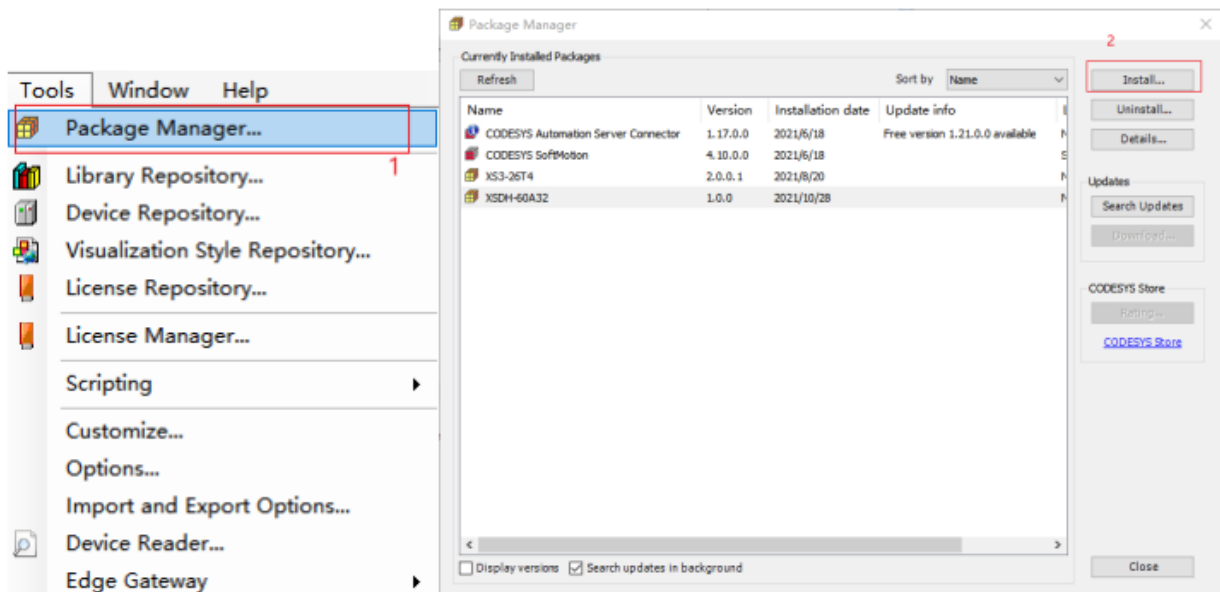
序号	命名	备注
①	XSDH-60A32	PLC 型号
②	3.5.15.40	Runtime 版本
③	1.0.0	Package 投产版本
④	P1	投产后第 1 次在线升级包
⑤	20211027	Package 包升级日期

7-1-2. Package 的获取

请联系技术支持, 电话: 400-885-0136。

7-1-3. Package 的安装

选择“Tools” — “Package Manager..”, 在弹出的界面里安装 Package 包, 选择“Install”, 找到 Package 所在位置进行安。例如: 想要安装 XSDH-60A32 的 Package, 最好先将之前的 Package 进行卸载后再安装新的。



7-2. XS 系列 PLC 固件升级

7-2-1. 固件命名规则

命名格式：XSDH-60A32_3.5.15.40_1.0.0_P1_20211027

序号	命名	备注
①	XSDH-60A32	PLC 型号
②	3.5.15.40	Runtime 版本
③	1.0.0	固件投产版本
④	P1	投产后第 1 次在线固件升级
⑤	20211027	固件升级日期

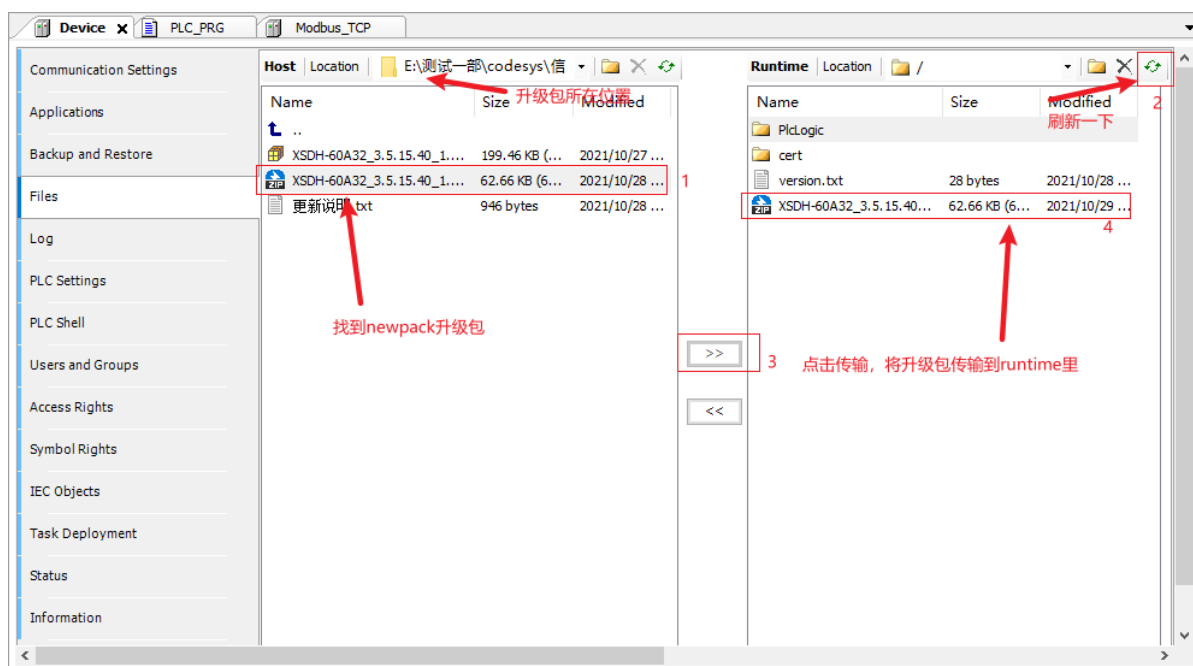
7-2-2. 固件获取

联系技术支持，电话：400-885-0136。

7-2-3. 固件安装及其注意事项

通过 newpack 包升级固件：

创建设备标准工程，连接设备，在主设备目录下选择“文件（Files）”选项，点击右上角的刷新后，将 newpack 升级包传输到 runtime 中，等待传输完成，重启设备，设备在升级时 ERR 灯会常亮，更新结束后 ERR 会熄灭，此时就可以扫到设备。



手册更新日志

本手册的资料编号记载在手册封面的右下角，关于手册改版的信息汇总如下：

序号	资料编号	章节	更新内容
1	PS04 20211125 1.0	-	第一版手册发布

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： (0510) 85134136

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: (510) 85134136

Fax: (510)85111290