

SIEMENS

SIMATIC

TIA Portal STEP 7 Basic V10.5

入门指南

TIA Portal 简介

1

简单实例

2

扩展实例

3

实例“PID 控制”

4

实例“运动”

5




01/2010

A5E02714430-01

法律资讯

警告提示系统

为了您的人身安全以及避免财产损失，必须注意本手册中的提示。人身安全的提示用一个警告三角表示，仅与财产损失有关的提示不带警告三角。警告提示根据危险等级由高到低如下表示。

 危险
表示如果不采取相应的小心措施， 将会 导致死亡或者严重的人身伤害。
 警告
表示如果不采取相应的小心措施， 可能 导致死亡或者严重的人身伤害。
 小心
带有警告三角，表示如果不采取相应的小心措施，可能导致轻微的人身伤害。
小心
不带警告三角，表示如果不采取相应的小心措施，可能导致财产损失。
注意
表示如果不注意相应的提示，可能会出现不希望的结果或状态。


当出现多个危险等级的情况下，每次总是使用最高等级的警告提示。如果在某个警告提示中带有警告可能导致人身伤害的警告三角，则可能在该警告提示中另外还附带有可能导致财产损失的警告。

合格的专业人员

本文件所属的产品/系统只允许由符合各项工作要求的**合格人员**进行操作。其操作必须遵照各自附带的文件说明，特别是其中的安全及警告提示。由于具备相关培训及经验，合格人员可以察觉本产品/系统的风险，并避免可能的危险。

按规定使用 Siemens 产品

请注意下列说明：

 警告
Siemens 产品只允许用于目录和相关技术文件中规定的使用情况。如果要使用其他公司的产品和组件，必须得到 Siemens 推荐和允许。正确的运输、储存、组装、装配、安装、调试、操作和维护是产品安全、正常运行的前提。必须保证允许的环境条件。必须注意相关文件中的提示。

商标

所有带有标记符号 © 的都是西门子股份有限公司的注册商标。标签中的其他符号可能是一些其他商标，这是出于保护所有权利的目的由第三方使用而特别标示的。

责任免除

我们已对印刷品中所述内容与硬件和软件的一致性作过检查。然而不排除存在偏差的可能性，因此我们不保证印刷品中所述内容与硬件和软件完全一致。印刷品中的数据都按规定经过检测，必要的修正值包含在下一版本中。

目录

1	TIA Portal 简介.....	7
1.1	概述.....	7
1.2	TIA Portal 基础.....	9
1.2.1	TIA Portal 概述.....	9
1.2.2	工程组态概念.....	11
1.2.3	数据管理.....	12
1.3	TIA Portal 中的视图.....	13
1.3.1	TIA Portal 中的导航.....	13
1.3.2	Portal 视图.....	14
1.3.3	项目视图.....	15
1.4	实例项目.....	16
1.4.1	概述.....	16
1.4.2	加载项目.....	18
2	简单实例.....	23
2.1	简介.....	23
2.2	创建项目.....	23
2.3	插入并组态 PLC.....	25
2.3.1	插入 PLC.....	25
2.3.2	设备和网络编辑器概述.....	28
2.3.3	组态 PLC.....	31
2.4	创建程序.....	33
2.4.1	什么是组织块?.....	33
2.4.2	打开组织块.....	35
2.4.3	程序编辑器概述.....	37
2.4.4	什么是程序段?.....	39
2.4.5	插入 LAD 指令.....	41
2.4.6	什么是变量?.....	46
2.4.7	定义和互连 PLC 变量.....	48
2.5	测试程序.....	52
2.5.1	将程序加载到目标系统.....	52
2.5.2	使用程序状态测试程序.....	58

2.6	创建 HMI 画面	61
2.6.1	TIA Portal 中的可视化	61
2.6.2	创建带 HMI 画面的 HMI 设备	61
2.6.3	什么是图形对象?	69
2.6.4	创建和组态图形对象	70
2.6.4.1	“设备开/关”按钮	70
2.6.4.2	图形对象“LED”	73
2.6.4.3	图形对象“传送带”	78
2.6.4.4	带有运动仿真的图形对象“瓶子”	81
2.6.4.5	改变运动动画的可见性	87
2.7	测试 HMI 画面	89
2.7.1	将 HMI 画面加载到 HMI 设备	89
2.7.2	仿真运行系统	92
3	扩展实例	95
3.1	简介	95
3.2	扩展程序	97
3.2.1	在 PLC 变量表中定义变量	97
3.2.2	对启动传送带的条件进行编程	101
3.2.2.1	查询机器的状态	101
3.2.2.2	查询瓶子的位置和加热室的状态	103
3.2.2.3	查询巴氏消毒进度	106
3.2.2.4	控制传送带	108
3.2.3	对停止传送带的条件进行编程	111
3.2.4	对加热过程的 PLC 进行编程	116
3.2.5	对加热时段进行编程	118
3.2.6	对状态指示灯进行编程	122
3.3	通过程序状态测试扩展的程序	126
3.4	扩展 HMI 画面	133
3.4.1	图形对象“加热室”	133
3.4.2	图形对象“加热室 LED”	135
3.4.3	图形对象“光栅”	137
3.5	仿真 HMI 画面	142

4	实例“PID 控制”	145
4.1	简介	145
4.2	创建 PID 控制器的组织块	148
4.3	创建工艺对象 PID 控制器	151
4.4	加载仿真块	153
4.5	组态 PID 控制器	157
4.6	更改加热室控制	162
4.7	在控制程序中集成温度比较作为条件	164
4.8	调整 HMI 画面	168
4.9	在线模式下激活 PID 控制器	171
5	实例“运动”	175
5.1	简介	175
5.2	插入工艺对象“轴”	179
5.3	组态工艺对象“轴”	181
5.4	启用轴	183
5.5	相对定位轴	186
5.6	扩展 HMI 画面	190
5.6.1	更改图形对象“传送带”	190
5.6.2	创建另一个图形对象“瓶子”	192
5.6.3	连接 HMI 对象与运动指令	197
5.7	仿真 HMI 画面	200
5.8	启动诊断视图	202
	词汇表	207

TIA Portal 简介

1.1 概述

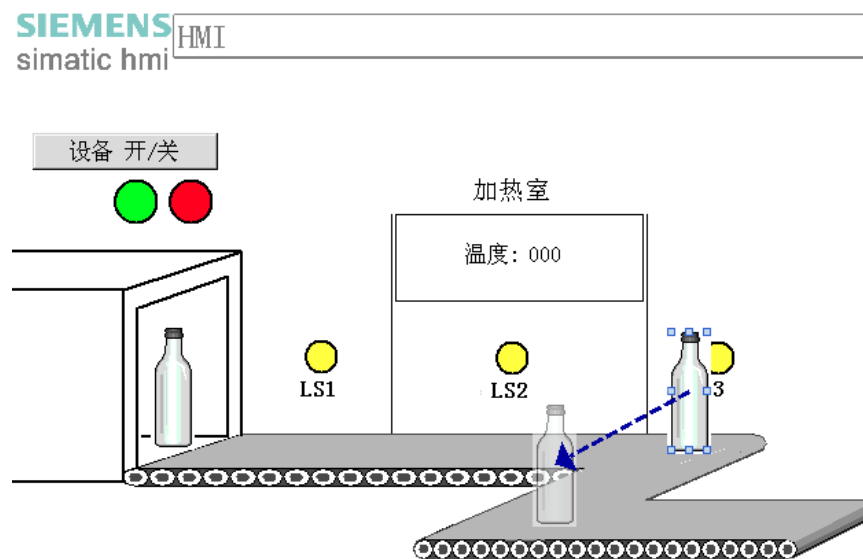
简介

欢迎使用“STEP7 Basic V10.5 入门指南”。

“入门指南”使用一个实例项目向用户介绍如何轻松操作 TIA Portal。

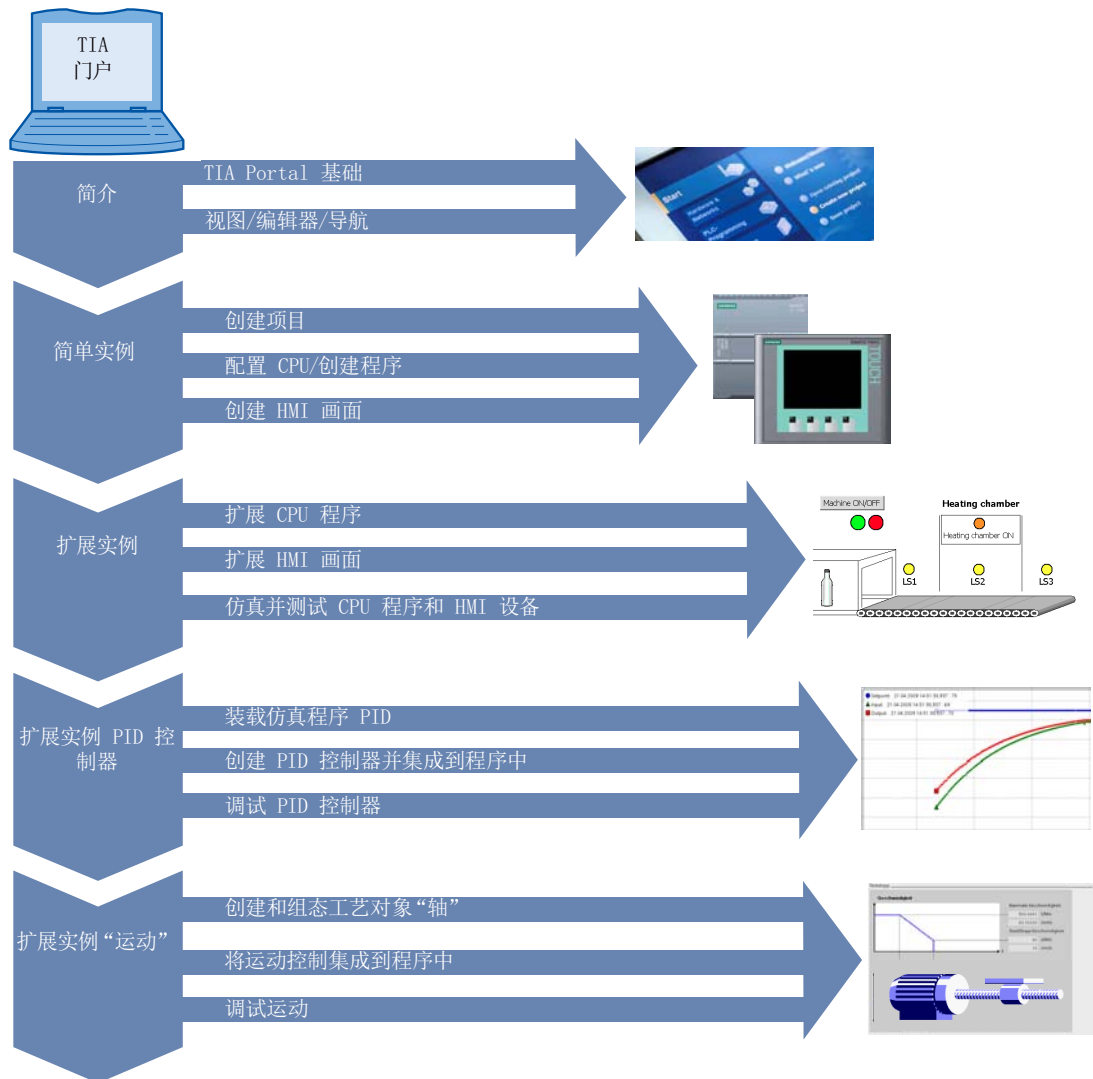
实例项目

本入门指南中创建的实例项目是一个用于在加热室中对牛奶进行巴氏消毒的工作站。简单来说，此过程就是通过传送带将各个瓶子送到加热室中，并在完成加热过程后将其传送到下一个处理阶段。



入门指南的结构

通过每一章的内容来展开介绍该实例项目。首先介绍了一个仅使用 TIA Portal 基本功能的简单项目，之后通过使用越来越复杂的 TIA Portal 功能来逐渐扩展该项目。经验丰富的用户可跳过某些章节；“简单实例”一章为初学者提供有关编程和可视化的附加背景信息。



1.2 TIA Portal 基础

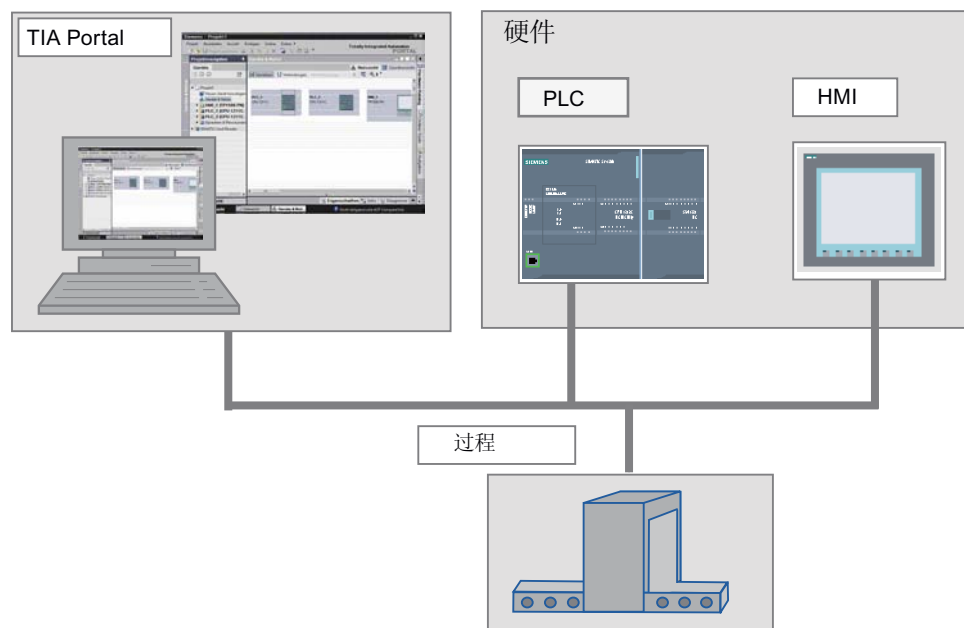
1.2.1 TIA Portal 概述

简介

TIA (Totally Integrated Automation, 全集成自动化) Portal 在一个软件应用程序中集成了各种 SIMATIC 产品, 使用这一软件可以提高生产力和效率。TIA 产品在 TIA Portal 中协同工作, 能够在创建自动化解决方案所需的各个方面为您提供支持。

典型的自动化解决方案包含以下内容:

- 借助程序来控制过程的 PLC。
- 用来操作和可视化过程的 HMI 设备。



任务

TIA Portal 可用来帮助您创建自动化解决方案。关键的组态步骤为：

- 创建项目
- 配置硬件
- 联网设备
- 对 PLC 编程
- 组态可视化
- 加载组态数据
- 使用在线和诊断功能

优点

TIA Portal 有以下优点：

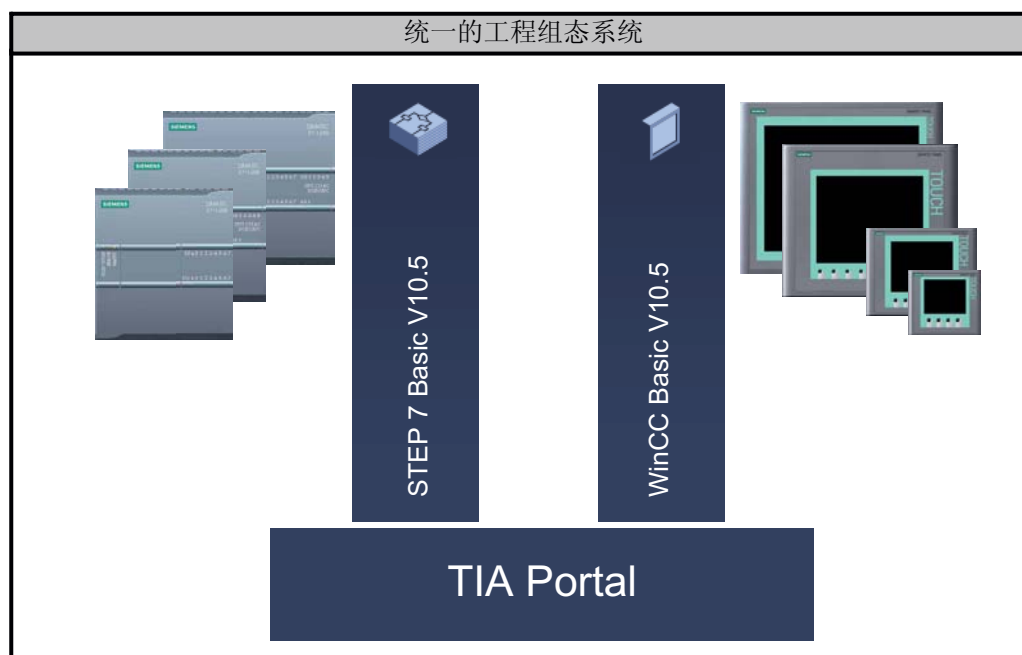
- 公共数据管理
- 易于处理程序、组态数据和可视化数据
- 可使用拖放操作轻松编辑
- 易于将数据加载到设备
- 易于操作
- 支持图形组态和诊断

1.2.2 工程组态概念

工程组态系统

可以使用 TIA Portal 在同一个工程组态系统中组态 PLC 和可视化。所有数据均存储在一个项目中。用于编程 (STEP 7) 和可视化 (WinCC) 的组件不是单独的程序，而是可以访问公共数据库的系统的编辑器。所有数据均存储在一个公共的项目文件中。

始终可以使用一个适合所有任务的公共用户界面来访问所有的编程和可视化功能。



1.2.3 数据管理

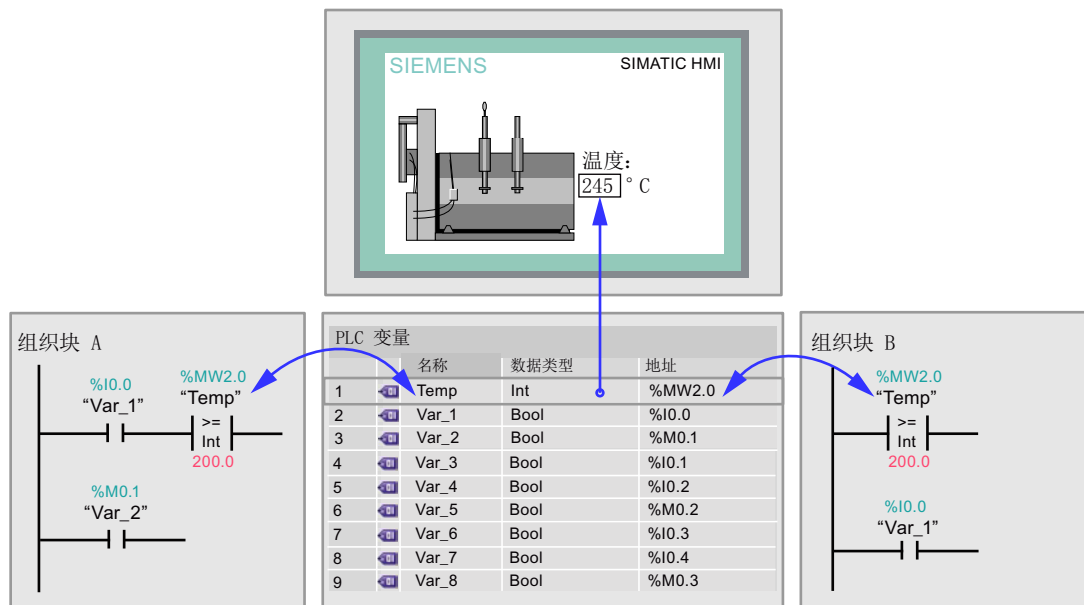
集中数据管理

在 TIA Portal 中，所有数据都存储在一个项目中。修改后的应用程序数据（如变量）会在整个项目内（甚至跨越多台设备）自动更新。

跨项目组成部分的符号寻址

如果在不同 PLC 的多个块中以及 HMI 画面中使用了过程变量，则可以在程序中的任意位置创建或修改该变量。这种情况下，在哪个设备的哪个块中进行修改并不重要。TIA Portal 提供以下用于定义 PLC 变量的选项：

- 在 PLC 变量表中定义
- 在程序编辑器中定义
- 通过 PLC 输入和输出的链接来定义



所有已定义的 PLC 变量都列在 PLC 变量表中，并可在表中进行编辑。变量修改是集中执行且不断进行更新的。一致的数据管理免去了在同一项目内的不同参与者之间（例如，程序员与 HMI 设计者之间）进行同步的必要。

库概念

通过库可以在同一项目和其它项目中再次使用项目组成部分。

- 块、PLC 变量、PLC 变量表、中断、HMI 画面、单个模块或完整站等元素可存储在本地库和全局库中。
- 设备和定义的功能可以重复使用。
- 通过全局库可轻松实现项目之间的数据交换。

1.3 TIA Portal 中的视图

1.3.1 TIA Portal 中的导航

简介

用户在创建项目时将使用不同的视图。以下部分初步概述了 TIA Portal 中的各种视图。

TIA Portal 的视图

对于自动化项目，TIA Portal 提供了两个不同的工作视图，通过它们可快速访问工具箱和各个项目组件：

- **Portal 视图：** Portal 视图支持面向任务的组态。
- **项目视图：** 项目视图支持面向对象的组态。

导航

可以随时使用用户界面左下角的链接在 **Portal** 视图和项目视图之间切换。在组态期间，视图也会根据正在执行的任务类型自动切换。例如，如果要编辑 **Portal** 视图中列出的对象，应用程序会自动切换到项目视图中的相应编辑器。编辑完对象后，可以切换回 **Portal** 视图并继续操作下一个对象或进行下一项活动。

全局保存项目数据

保存项目时，无论打开了哪个视图或编辑器，始终会保存整个项目。

1.3.2 Portal 视图

Portal 视图

Portal 视图提供面向任务的工具箱视图。Portal 视图用于提供一种简单的方式来浏览项目任务和数据。这表明可通过各个 Portal 来访问处理关键任务所需的应用程序功能。下图显示了 Portal 视图的结构：



① 不同任务的 Portal:

Portal 为各个任务区提供了基本功能。在 Portal 视图中提供的 Portal 取决于所安装的产品。

② 所选 Portal 对应的操作:

此处提供了在所选 Portal 中可使用的操作。可在每个 Portal 中调用上下文相关的帮助功能。

③ 为所选操作选择窗口:

所有 Portal 都有选择窗口。该窗口的内容取决于您当前的选择。

④ 切换到项目视图:

可以使用“项目视图”(Project view) 链接切换到项目视图。

⑤ 当前打开的项目的显示区域:

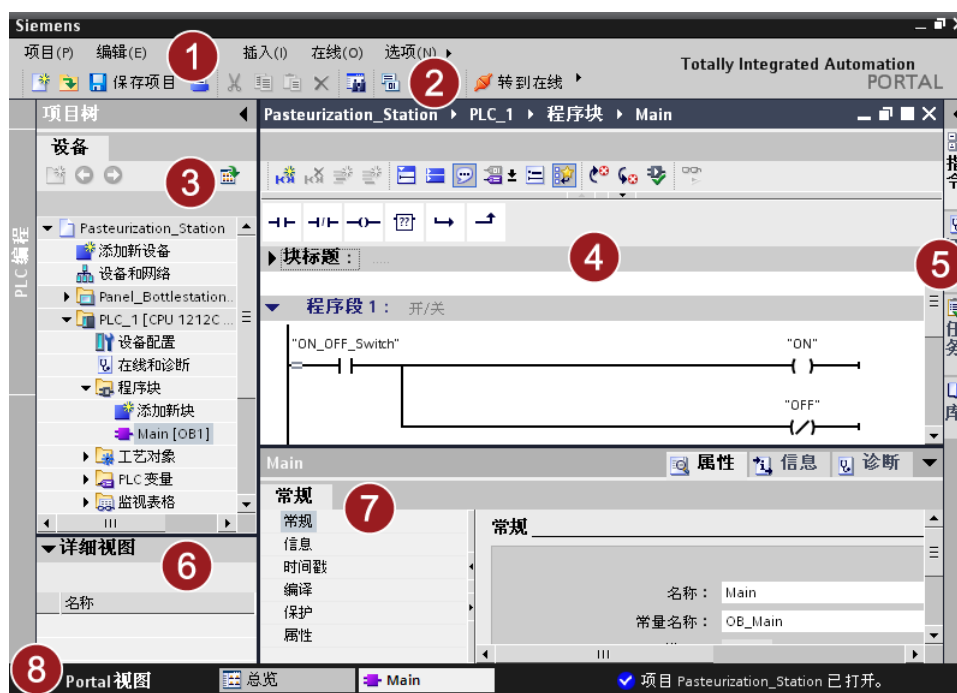
在此处可了解当前打开的是哪个项目。

1.3.3 项目视图

项目视图

项目视图是项目所有组件的结构化视图。项目视图中提供了各种编辑器，可以用来创建和编辑相应的项目组件。

下图显示了项目视图的结构：



- ① 菜单栏：
菜单栏包含您工作所需的全部命令。
- ② 工具栏：
工具栏提供了常用命令的按钮。这提供了一种比菜单更快的命令访问方式。
- ③ 项目树：
通过项目树可以访问所有组件和项目数据。例如，可在项目树中执行以下任务：
 - 添加新组件
 - 编辑现有组件
 - 扫描和修改现有组件的属性
- ④ 工作区：
为进行编辑而打开的对象将显示在工作区内。

1.4 实例项目

- ⑤ 任务卡：
可用的任务卡取决于所编辑或所选择的对象。在屏幕右侧的条形栏中可以找到可用的任务卡。可以随时折叠和重新打开这些任务卡。
- ⑥ 详细视图：
在详细视图中显示所选对象的某些内容。其中可能包含文本列表或变量。
- ⑦ 巡视窗口：
在巡视窗口中显示有关所选对象或所执行动作的附加信息。
- ⑧ 切换到 Portal 视图：
可以使用“Portal 视图”(Portal view) 链接切换到 Portal 视图。

说明

使用组合键“<Ctrl> + 1-5”打开和关闭项目视图的各个窗口。在 TIA Portal 的信息系统中，可以找到关于所有组合键的概述。

1.4 实例项目

1.4.1 概述

简介

入门指南分为四章，每一章都以前一章为基础。每一章末尾的最新状态都会保存在一个项目文件中。要跳过某一章，可以加载上一章的相应状态。

实例项目的内容

下列概述说明了相应章节的目标组。项目以 ZIP 文件形式保存，可从以下地址下载：

<http://support.automation.siemens.com/CN/view/zh/40263542>

请单击“信息”图标查看相关 ZIP 文件。

- 简单实例

这些章节面向没有先验知识的初学者。在简单实例中，将组态 PLC 和一个 HMI 设备，创建一段短程序和一个用于可视化的 HMI 画面。在章节末尾，您将能够使用 HMI 设备打开和关闭机器。

- 扩展实例

如果您具有先验知识，则可以加载简单实例并从这里开始继续操作。在该简单实例扩展中，您将使用 LAD 创建完整的程序以及过程可视化所需的 HMI 元素。

“简单实例”一章之后的项目状态存储在“Simple_Example.ZIP”文件中。

- 扩展实例 PID 控制器

在本章中，您会将工艺对象“PID 控制器”添加到扩展实例。使用 PID 控制器自动控制加热室中的温度。

如果只想使用工艺对象“PID”的功能，您可以通过“Extended_Example.ZIP”文件加载“扩展实例”一章之后的项目状态。

- 扩展实例运动控制

在本章中，您会将工艺对象“轴”添加到扩展实例。使用工艺对象“轴”控制另一条传送带。使用运动控制指令修改瓶子在传送带上的位置。本章之前的项目状态存储在“Extended_Example_PID.ZIP”文件中。

整个入门指南结束时的项目状态存储在“Extended_Example_Motion.ZIP”文件中。

**警告****实例项目只可用于测试**

所提供的实例和说明只用于向您介绍 TIA Portal 的基本功能。

- 请仅在测试环境中使用实例，不可在已投入运行的机器中使用。
- 若将实例程序加载至已投入运行的机器中，可能会导致故障、程序出错以及造成巨大财产损失和人员重伤！

1.4.2 加载项目

简介

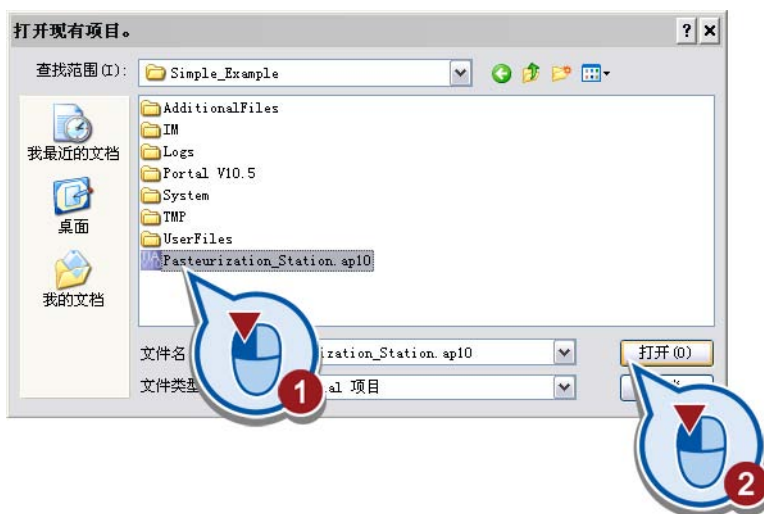
入门指南从章节“简单实例 (页 23)”开始。 如果不想从最开始学习入门指南，以下步骤将向您介绍如何加载项目状态。

步骤

执行下列步骤，加载项目：

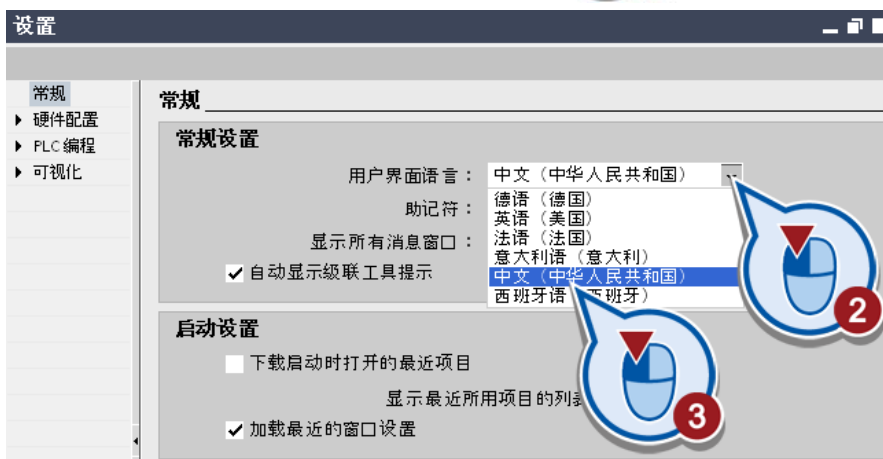
1. 使用 TIA Portal 的 Portal 视图打开相应的项目。



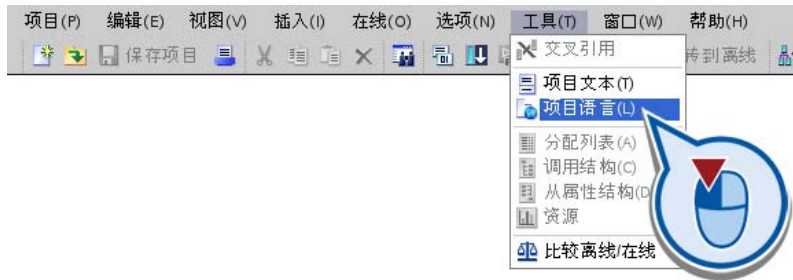


即可完成加载项目。

2. 打开项目视图。
3. 选择用户界面语言。

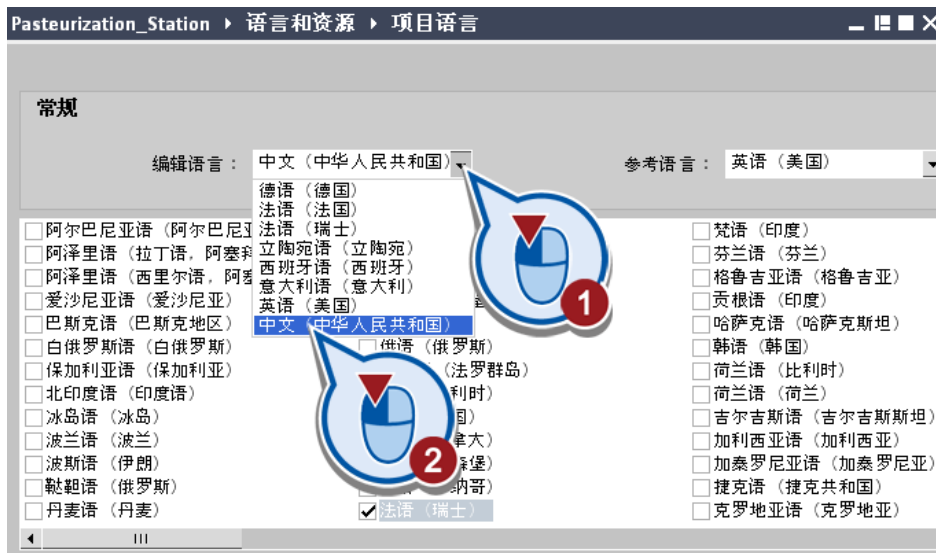


4. 打开项目语言设置窗口。



使用项目语言选择显示项目文本（如文本块或注释）时要使用的语言。

5. 选择项目文本的语言。



使用项目语言选择显示项目文本（如文本块或注释）时要使用的语言。

6. 在项目视图中，启用在运行系统设置中的所需语言。



使用语言设置指定将在 HMI 设备中加载的语言。HMI 设备中可加载的语言数量取决于所用的 HMI 设备。启动过程中，将首先显示序号为“0”的语言。

说明

语言切换

变量名称不会受项目语言切换的影响，因为变量名称必须唯一。

简单实例

2.1 简介

步骤

在入门指南的第一部分，用户将设计一个用于启动和关闭机器的电动按钮。通过按一次按钮启动机器。再次按下该按钮将切断电源，从而关闭机器。

为此，请按下列步骤操作：

- 创建项目
- 组态 PLC
- 创建程序
- 将程序加载到 PLC
- 测试程序
- 创建 HMI 画面

2.2 创建项目

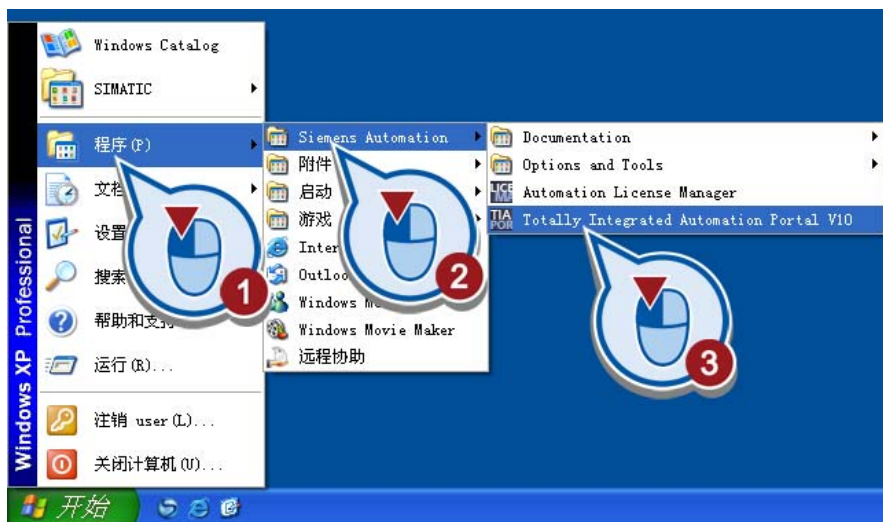
简介

以下步骤介绍了如何创建一个新项目。创建自动化任务时产生的数据和程序会有序地存储在项目中。在本例中打开 TIA Portal 时显示 Portal 视图。“启动”Portal 包含用于创建新项目和打开现有项目的命令。

步骤

要创建新项目，请按以下步骤操作：

1. 启动 TIA Portal。



TIA Portal 在 Portal 视图中打开。

2. 在任意路径下创建项目 "Pasteurization_Station"。



结果

已创建一个新项目。在下一部分中，您将向项目中添加一个新 PLC 并组态其属性。

2.3 插入并组态 PLC

2.3.1 插入 PLC

简介

以下步骤介绍如何在 **Portal** 视图中插入 **PLC** 以及在项目视图中打开其配置。在项目中创建的 **PLC** 的类型必须与所用的硬件一致。

要求

已创建一个项目。

步骤

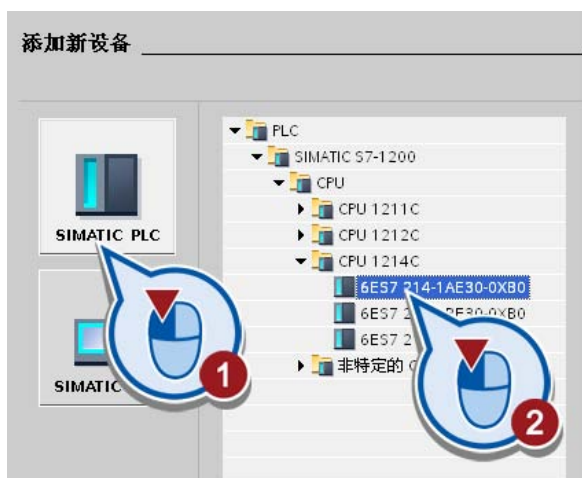
要向项目中添加新设备，请按以下步骤操作：

1. 使用 **Portal** 添加新设备。



2.3 插入并组态 PLC

2. 选择所需的 PLC。



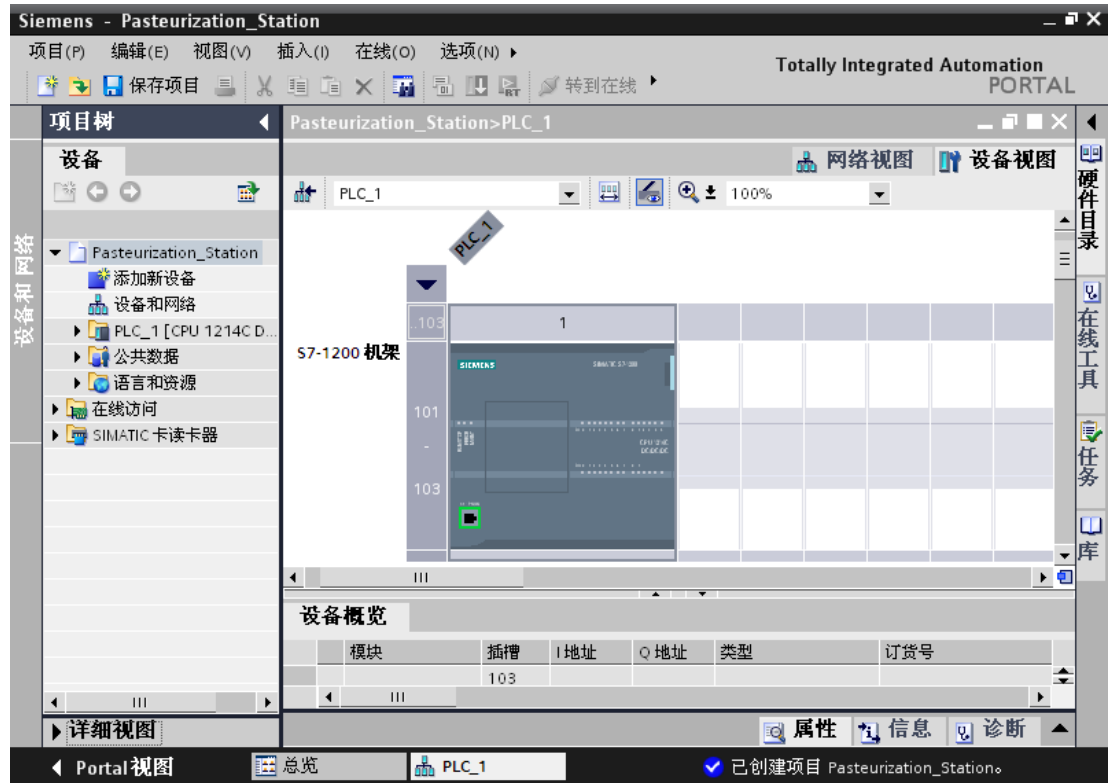
3. 确保启用了“打开设备视图”(Open device view) 选项。
如果未启用该选项，请在选项内单击鼠标左键进行启用。

4. 单击“添加”(Add)。



结果

在项目中创建了一个新 PLC，并在设备和网络编辑器的设备视图中打开了该 PLC。



2.3.2 设备和网络编辑器概述

设备和网络编辑器的功能

设备和网络编辑器是一个集成开发环境，用于对设备和模块进行配置、联网和参数分配。它由网络视图和设备视图组成。可以随时在这两个编辑器之间进行切换。

网络视图

网络视图是设备和网络编辑器的工作区域，在该区域内可以执行以下任务：

- 配置和分配设备参数
- 使设备相互连接

下图显示了网络视图的结构：



① 用于在设备视图与网络视图之间切换的选项卡

② 工具栏：

工具栏中包括用于图形化设备联网、组态连接以及显示地址信息的工具。使用缩放功能可以更改图形区域中的显示。

- ③ 图形区域：

图形区域显示与网络相关的设备、网络、连接和关系。在图形区域中，可以插入硬件目录 (7) 中的设备，并可通过可用接口将这些设备互连。
- ④ 总览导航：

总览导航提供图形区域中所创建对象的概览。按住鼠标按钮，可以快速导航到所需的对象并在图形区域中显示它们。
- ⑤ 表格区域：

表格区域概要说明正在使用的设备、连接以及通信连接。
- ⑥ 巡视窗口：

巡视窗口显示当前所选对象的信息。可以在巡视窗口的“属性”(Properties) 选项卡中编辑所选对象的设置。
- ⑦ “硬件目录”任务卡：

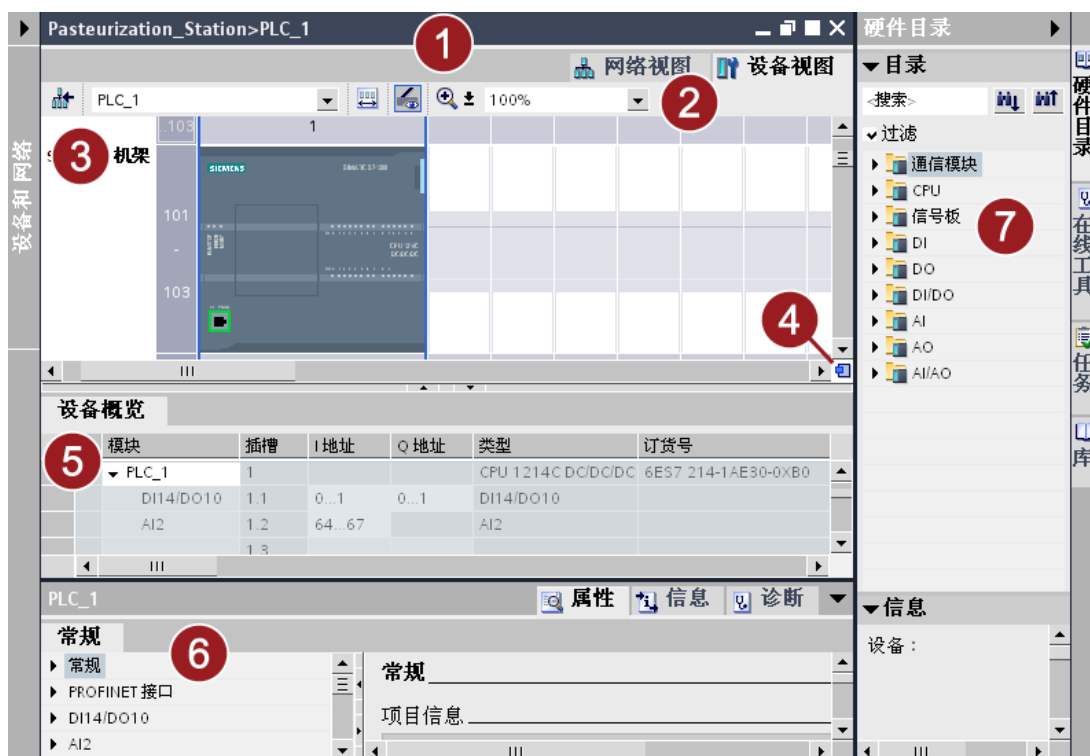
使用“硬件目录”(Hardware catalog) 任务卡可以轻松访问各种硬件组件。将自动化任务所需的设备和模块从硬件目录拖到网络视图的图形区域。

设备视图

设备视图是设备和网络编辑器的工作区域，在该区域内可以执行以下任务：

- 配置和分配设备参数
- 配置和分配模块参数

下图显示了设备视图的结构：



- ① 用于在设备视图与网络视图之间切换的选项卡
- ② 工具栏：
可以使用工具栏在各种设备之间切换以及显示和隐藏某些信息。使用缩放功能可以更改图形区域中的显示。
- ③ 图形区域：
设备视图的图形区域显示设备与相关模块，它们彼此间通过一个或多个机架来分配给对方。在图形区域中，可以将其它硬件对象从硬件目录 (7) 拖到机架的插槽中并对它们进行配置。

- ④ 总览导航：
总览导航提供图形区域中所创建对象的概览。按住鼠标按钮，可以快速导航到所需的对象并在图形区域中显示它们。
- ⑤ 表格区域：
表格区域提供了所用模块以及最重要的技术数据和组织数据的概览。
- ⑥ 巡视窗口：
巡视窗口显示当前所选对象的信息。可以在巡视窗口的“属性”(Properties) 选项卡中编辑所选对象的设置。
- ⑦ “硬件目录”任务卡：
使用“硬件目录”(Hardware catalog) 任务卡可以轻松访问各种硬件组件。将自动化任务所需的设备和模块从硬件目录拖到设备视图的图形区域。

2.3.3 组态 PLC

简介

以下步骤介绍了如何配置所插入 PLC 的 PROFINET 接口。

要求

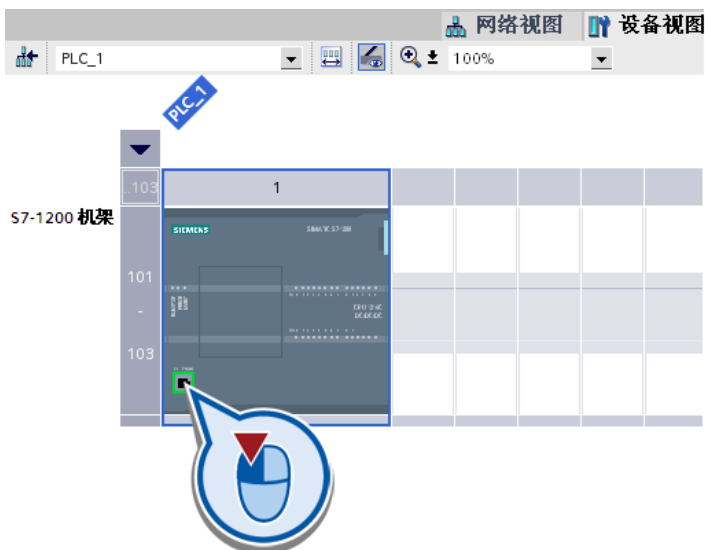
- 已创建项目。
- 已在设备和网络编辑器的设备视图中打开该 PLC。

步骤

要组态 PLC，请按以下步骤操作：

1. 在图形视图中选择 PROFINET 接口。

PROFINET 接口的属性随即显示在巡视窗口中。



2. 在巡视窗口的“以太网地址”(Ethernet addresses) 下面，输入 PLC 的 IP 地址。



3. 单击工具栏上的“保存项目”(Save project) 图标保存项目。
4. 关闭设备和网络编辑器。

结果

通过设置 PROFINET 接口的属性配置了 PLC。

2.4 创建程序

简介

有了 PLC 后，在项目中会自动创建组织块“Main [OB1]”。在下一部分中，您将在该组织块中创建用户程序。

2.4.1 什么是组织块？

用户程序

用户程序可由一个或多个块组成。必须至少使用一个组织块。块包含处理特定自动化任务所需的全部功能。

程序的任务包括：

- 处理过程数据，例如，链接二进制信号，读入并利用模拟量，定义输出的二进制信号以及输出模拟值。
- 中断响应，例如，超出模拟扩展模块测量范围时的诊断错误中断。
- 正常程序执行中的错误处理

组织块

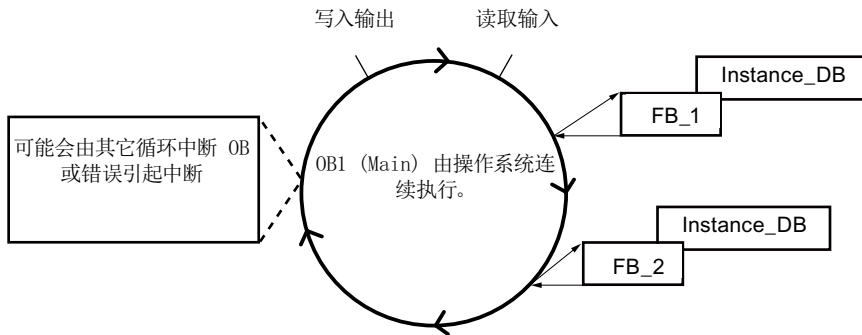
组织块 (OB) 构成 PLC 的操作系统与用户程序之间的接口。组织块由操作系统调用，并控制下列操作：

- 自动化系统的启动行为
- 循环程序执行
- 基于中断的程序执行
- 错误处理

自动化项目中必须至少有一个程序循环 OB。确定 PLC 行为的程序被写入到此程序循环 OB 中。操作系统每个循环调用该 OB 一次，从而开始执行 OB 中包含的程序。每次程序执行结束后，重新开始循环。

可以通过调用其它组织块来中断组织块的程序执行。在执行复杂的自动化任务期间，程序会被构造成在程序循环 OB 中调用并依次执行的若干块。

下图显示了程序循环 OB 的执行过程：



名为“Main [OB1]”的程序循环 OB 是在向项目中插入 PLC 时自动创建的。您在该组织块中创建入门指南项目的程序。

2.4.2 打开组织块

简介

以下步骤介绍如何在程序编辑器中打开组织块。程序编辑器是一个用于创建程序的集成开发环境。

要求

项目中存在 PLC。

步骤

要打开组织块“Main [OB1]”，请按以下步骤操作：

1. 在项目树中打开“程序块”(Program blocks) 文件夹。

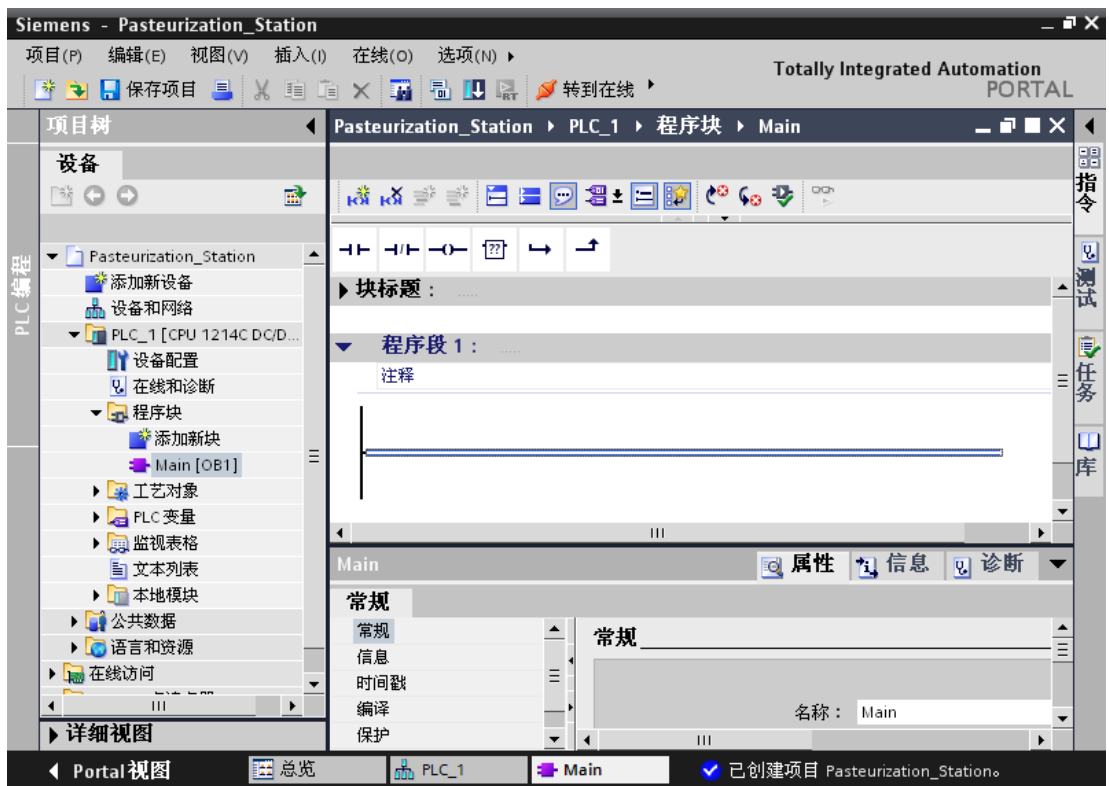


2. 打开组织块“Main [OB1]”。



结果

在程序编辑器中打开了组织块“Main [OB1]”，并可以在此创建程序。



说明

调整工作区

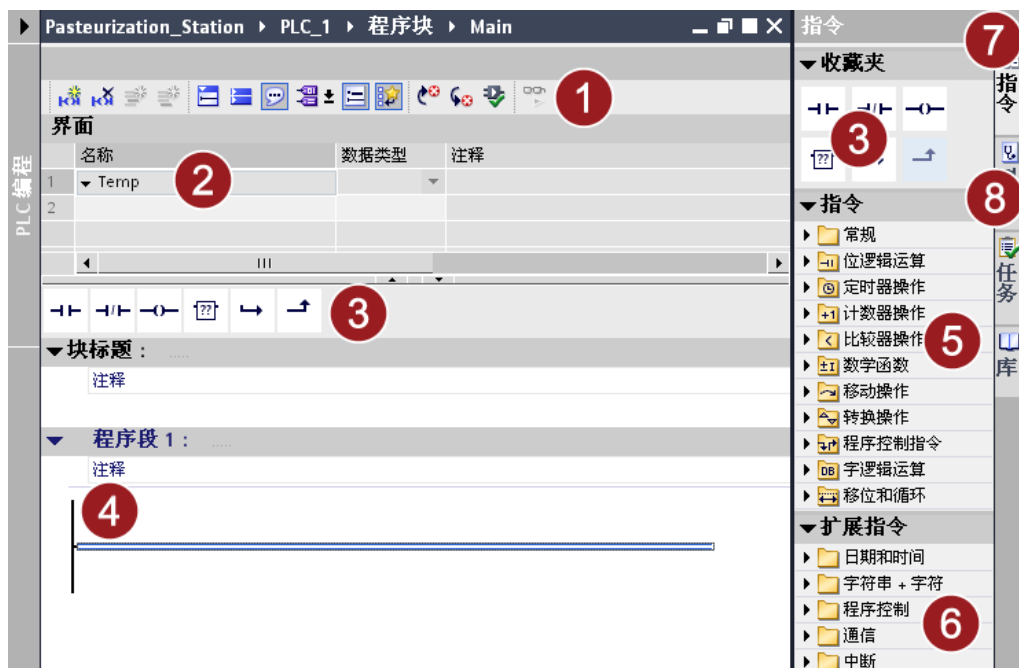
可以移动、分离、水平或垂直拆分工作区中的窗口。TIA Portal 的信息系统提供了更多关于本主题的信息。

2.4.3 程序编辑器概述

程序编辑器的功能

可以使用程序编辑器创建程序包含的块。程序编辑器由若干区域组成，可根据不同功能对各种编程任务的执行提供支持。

下图显示了程序编辑器的结构：



- ① 工具栏：
使用工具栏可以访问程序编辑器的主要功能，例如：
 - 插入、删除、打开和关闭程序段
 - 显示和隐藏绝对操作数
 - 显示和隐藏程序段注释
 - 显示和隐藏收藏夹
 - 显示和隐藏程序状态
- ② 块接口：
通过块接口可以创建和管理局部变量。
- ③ “指令”(Instructions) 任务卡中的“收藏夹”(Favorites) 窗格和程序编辑器中的收藏夹：
通过收藏夹可以快速访问常用的指令。可单独扩展“收藏夹”(Favorites) 窗格以包含更多指令。
- ④ 指令窗口：
指令窗口是程序编辑器的工作区。可在其中执行以下任务：
 - 创建和管理 程序段 (页 39)
 - 输入块和程序段的标题与注释
 - 插入指令并为指令提供变量。
- ⑤ “指令”(Instructions) 任务卡中的“指令”窗格
- ⑥ “指令”(Instructions) 任务卡中的“扩展指令”(Extended Instructions) 窗格
- ⑦ “指令”(Instructions) 任务卡
“指令”(Instructions) 任务卡包含用于创建程序内容的指令。
- ⑧ “测试”(Testing)任务卡

2.4.4 什么是程序段？

简介

组织块程序分为若干程序段。程序段可用于构建程序。每个块最多可以包含 999 个程序段。

在组织块“Main [OB1]”中会自动创建一个程序段。

采用 LAD 编程语言的程序段

可以使用不同编程语言创建组织块的程序。对于实例项目，使用图形编程语言 LAD 编辑组织块“Main [OB1]”。

此编程语言使用基于电路图的表示法，即，块中的每个 LAD 程序被分为若干程序段，每个程序段包含一根电源线和至少一个梯级。

通过添加其它梯级可扩展程序段。可以使用分支在特定梯级中创建并联结构。梯级和程序段按照从上到下、从左到右的顺序执行。

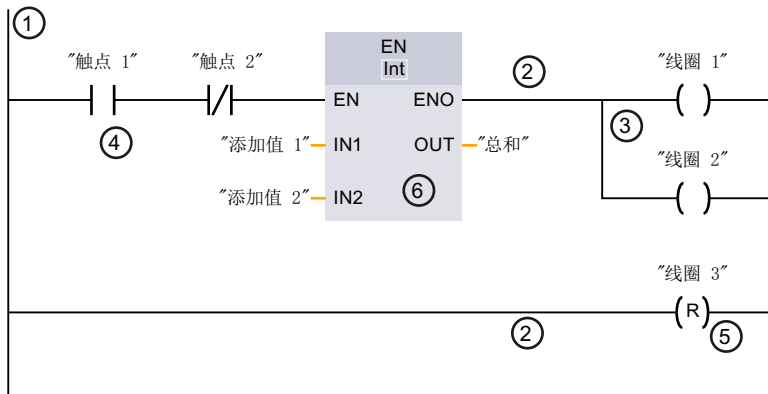
LAD 指令

可以使用用户界面的“指令”任务卡中提供的 LAD 指令创建实际程序内容。有三种不同类型的 LAD 指令：

- 触点：可以使用触点创建或中断两个元素之间的载流连接。在这种情况下，元素可以是 LAD 程序元素或电源线的边沿。电流从左向右传递。可以使用触点查询操作数的信号状态或值，并根据电流的结果对其进行控制。
- 线圈：可以使用线圈修改二进制操作数。线圈可根据逻辑运算结果的信号状态置位或复位二进制操作数。
- 功能框：功能框是具有复杂功能的 LAD 元素。但空功能框除外。可以使用空功能框作为占位符，在其中可以选择所需的运算。

在“指令”任务卡中可找到触点、线圈和功能框的各种变体，这些变体根据其功能被划分到不同的文件夹中。必须给多数 LAD 指令提供变量。

下图显示了已编写的 LAD 程序段实例：



- ① 电源线
- ② 梯级
- ③ 分支
- ④ 触点
- ⑤ 线圈
- ⑥ 功能框

2.4.5 插入 LAD 指令

简介

以下步骤介绍了如何在组织块“Main [OB1]”的第一个程序段中创建实例机器的开关程序。为此，需插入一个分支和下列指令：

- 常开触点
- 输出线圈
- 取反线圈

此外，采用“LAD”编程语言通过分支来设计一个并联结构。

常开触点

下图显示了程序中常开触点的图标：

<操作数>

---|---

常开触点的激活取决于相关变量的信号状态（<操作数>）。

- 如果变量的信号状态为“1”，则常开触点闭合。信号流从左侧电源线通过该常开触点流到右侧电源线，并且该指令输出的信号状态设置为“1”。
- 如果变量的信号状态为“0”，则常开触点不会被激活。到右侧电源线的信号流中断，并且该指令输出的信号状态复位为“0”。

在本项目中，还将用到常闭触点（页 101），它的作用与常开触点基本相同，只是以相反的方式响应变量的信号状态。

输出线圈

下图显示了程序中“输出线圈”指令的图标：

<操作数>

---()---

可以使用“输出线圈”运算置位指定变量（<操作数>）的位。如果线圈输入的信号状态为“1”，则变量位置位为“1”。如果线圈输入的信号状态为“0”，则变量位的信号状态也为“0”。

取反线圈

下图显示了程序中“取反线圈”指令的图标：

<操作数>

---(/)---

“取反线圈”操作对信号状态取反并将相应的位分配给指定变量（<操作数>）。如果线圈输入的信号状态为“1”，则相应位复位为“0”。如果线圈输入的信号状态为“0”，则变量位置位为“1”。

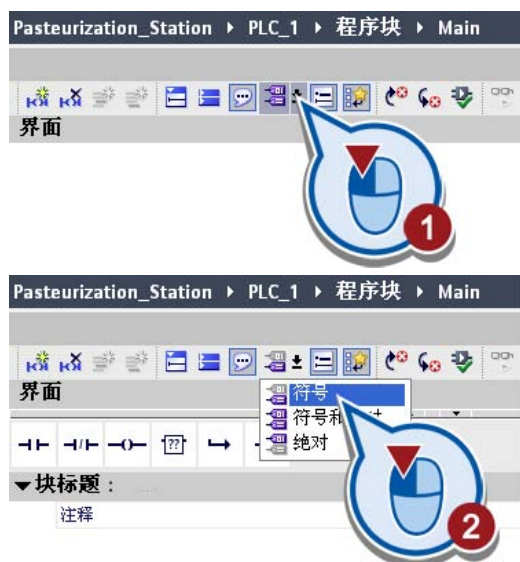
要求

- 已创建项目。
- 已组态 PLC。
- 组织块“Main [OB1]”已打开。

步骤

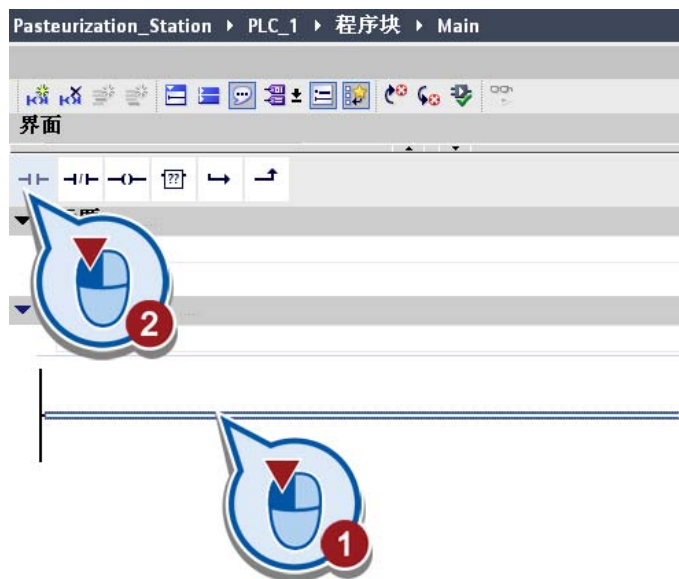
要创建实例机器的开关程序，请按以下步骤操作：

1. 激活变量的符号表示形式。

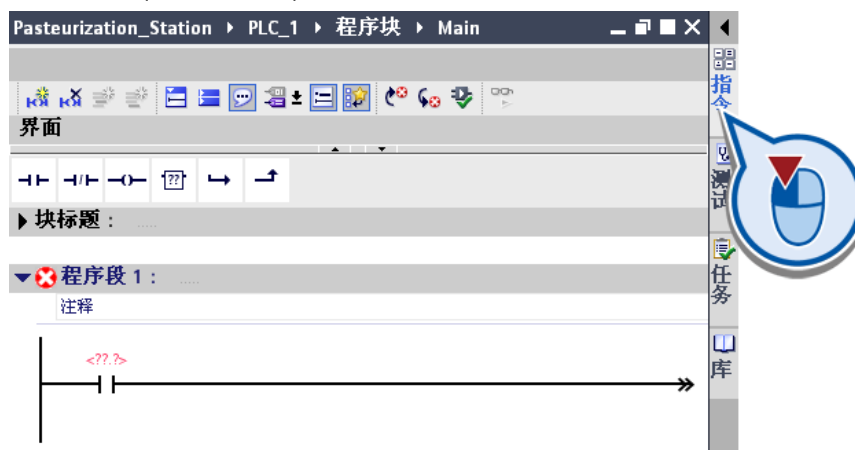


激活变量的符号表示形式后，程序段中将不显示变量地址。

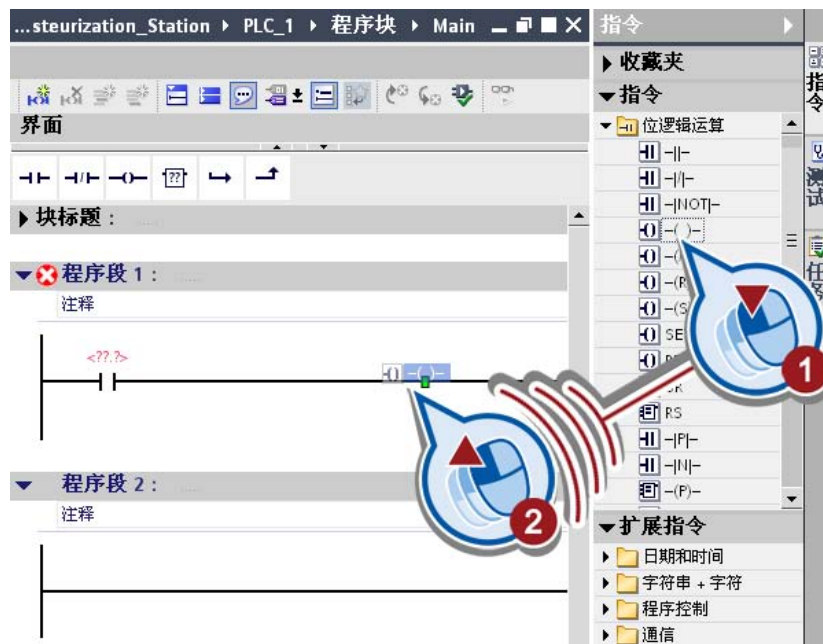
2. 向块中的第一个程序段插入一个常开触点。



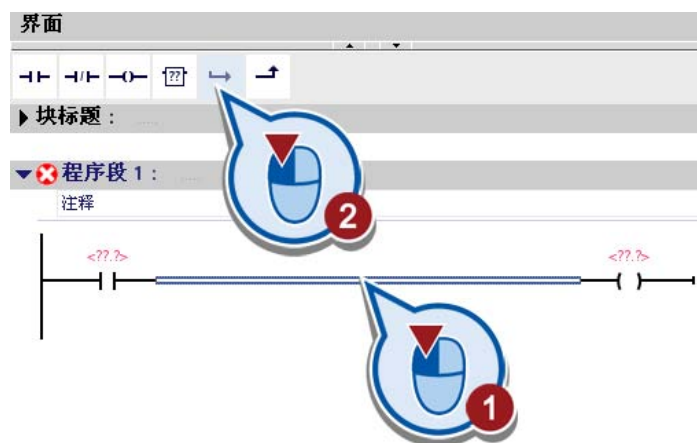
3. 打开“指令”(Instructions) 任务卡。



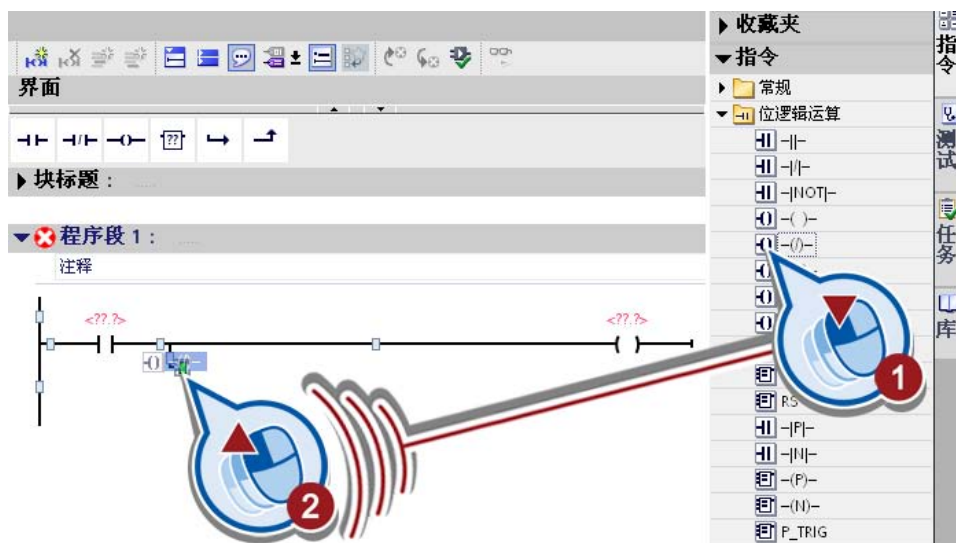
4. 在主梯级的末尾插入“输出线圈”指令。



5. 插入分支。



6. 在分支末尾插入“取反线圈”指令。



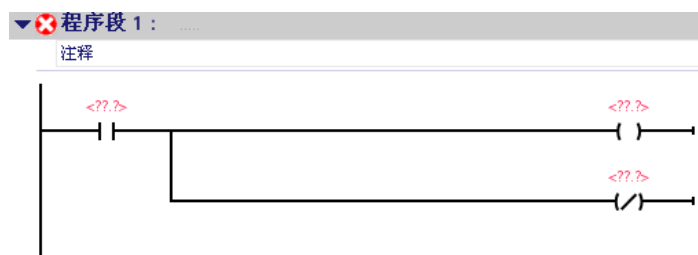
说明

修改指令

如果选择了错误的指令类型，可以通过指令图标直接在程序段中进行修改。通过将鼠标指针移动到指令右上角的黄色三角形上来打开下拉列表选择指令图标。

结果

在组织块“Main [OB1]”的第一个程序段中插入这些 LAD 指令后，即完成创建实例机器开关程序的第一部分。



在下一部分中，将 LAD 指令与 PLC 变量互连，从而完成该程序。

2.4.6 什么是变量？

简介

在上一步中，已向组织块“Main [OB1]”的第一个程序段中插入了三条 LAD 指令。在下一部分中，将把这些指令与 PLC 变量互连。PLC 变量用于为指令提供值，将根据这些值来执行指令。

在下一部分中，您将了解到更多有关用户程序中变量常规功能的信息。

变量

变量是用于程序的，可以有不同值的变量。根据应用范围，变量可分为以下几类：

- 局部变量：局部变量仅适用于定义这些变量的块。
- PLC 变量：PLC 变量适用于整个 PLC。

程序中多数指令都通过变量来操作。为指令分配变量后，即会使用指定变量的值来执行该指令。

变量在 TIA Portal 中集中管理。在程序编辑器中创建 PLC 变量与在 PLC 变量表中创建 PLC 变量没什么区别。如果在程序或 HMI 画面的多个位置使用某个变量，则对该变量所作的更改会立即在所有编辑器中生效。

变量的优点

变量的优点在于可以集中更改程序中使用的寻址方式。若没有变量提供的符号寻址功能，则每次 PLC 输入和输出的组态发生变化时，在用户程序中反复使用的寻址方式必须在程序中的多个位置进行更改。

PLC 变量

PLC 变量由以下部分组成：

- 名称（例如，CONVEYOR_ON）：变量名称只对一个 PLC 有效，并且在整个程序 and 此特定 PLC 中只能出现一次。
- 数据类型（例如，BOOL）：数据类型定义值的表示形式和允许的值范围。例如，通过选择 BOOL 数据类型，即指定变量只接受二进制值“0”和“1”。
- 地址（例如，M 3.1）：变量地址是绝对地址，定义变量读值或写值的存储区。输入、输出和位存储区均为可能的存储区实例。

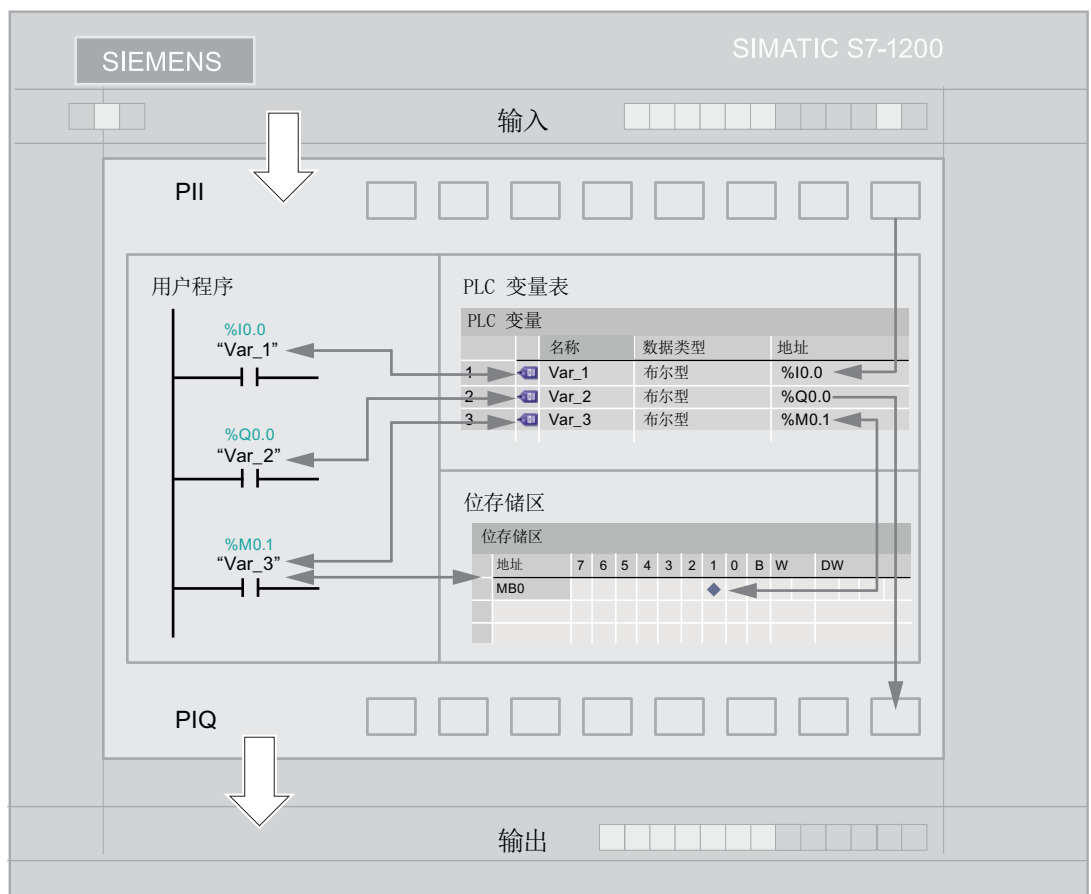
可以使用输入 (I) 和输出 (Q) 存储区对过程映像寻址。

过程映像包含 PLC 模块输入和输出的映像：

- PLC 的操作系统在每个程序循环都会刷新过程映像中输入的信号状态（请参见 什么是组织块？（页 33））。
- 过程映像中输出的信号状态在每个程序循环由 PLC 的操作系统写入到相应输出模块的输出。

机器或过程通常通过输出的信号状态来控制。

下图显示了 PLC 变量表、用户程序中的变量、位存储区以及 PLC 的输入和输出在理论上是如何互相链接的。




位存储区主要用于保存中间结果。在位存储区中寻址的变量值存储在系统存储器中，并且不会传送给模块。变量的数据类型决定变量在存储器中占用的存储空间。例如，**BOOL** 数据类型的变量在存储器中仅占用一位。**INT** 数据类型的变量在存储器中占用 16 位。变量不允许在一个存储区中重叠。变量的地址必须唯一。

PLC 变量表

PLC 变量表包含对于某个 PLC 有效的变量和常量的定义。系统会为项目中创建的每个 PLC 自动创建一个 PLC 变量表。

下表给出了“变量”(Tags) 选项卡中各表格列的含义：

列	说明
	可以单击该符号，以便通过拖放操作将变量移动到程序段中以用作操作数。
名称	为变量定义的且在整个 PLC 中唯一的名称。
数据类型	为变量指定的数据类型。
地址	变量地址。
保持性	保持性变量的值将保留，即使在电源关闭后也是如此。
监视值	PLC 中的当前数据值。 仅当在线连接可用并选择“监视”(Monitoring) 按钮时，此列才会出现。
注释	用于记录变量的注释。

2.4.7 定义和互连 PLC 变量

简介

在 TIA Portal 中，可以在程序段中创建用户程序时直接创建变量。以下步骤介绍了如何定义 PLC 变量以及将插入的 LAD 指令与 PLC 变量互连。LAD 指令根据变量值执行，以此来控制机器的启动和关闭。

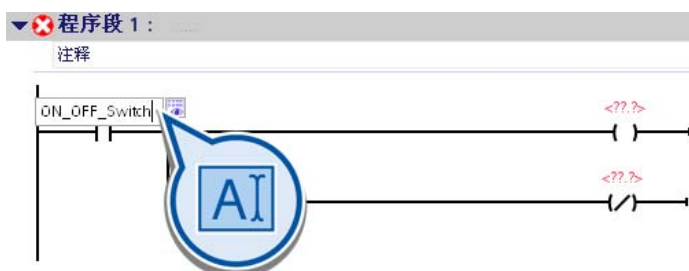
要求

- 已创建项目。
- 已组态 PLC。
- 块“Main [OB1]”已打开。
- LAD 指令“常开触点”、“输出线圈”和“取反线圈”已插入到组织块“Main [OB1]”的第一个程序段。

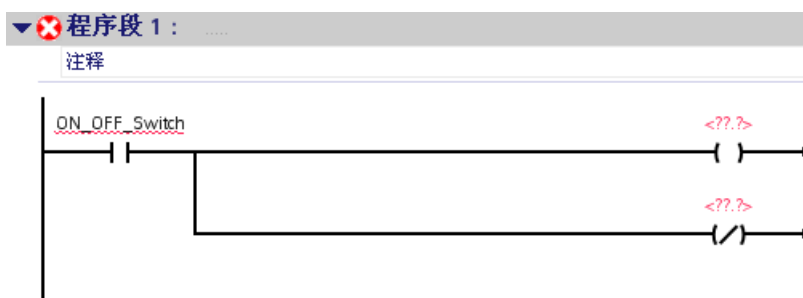
步骤

要将 LAD 指令与 PLC 变量互连，请按以下步骤操作：

1. 打开组织块“Main [OB1]”的第一个程序段。
2. 在常开触点的操作数占位符中输入名称 "ON_OFF_Switch"。



3. 按回车键确认输入内容。



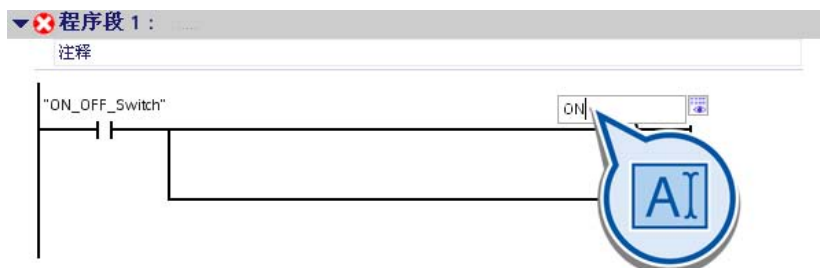
4. 打开“定义变量”(Define tag) 对话框。



5. 定义 "ON_OFF_Switch" 变量。



6. 在“输出线圈”指令的操作数占位符中输入名称 "ON"。



7. 按回车键确认输入内容。

8. 打开“定义变量”(Define tag) 对话框。



1

2

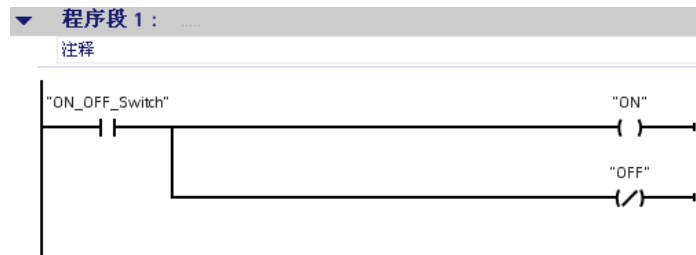
9. 定义“ON”变量。



10. 在“取反线圈”指令的操作数占位符中输入名称 "OFF" 并定义相应变量。
11. 单击工具栏上的“保存”(Save) 按钮以保存该项目。

结果

编写了一个打开和关闭实例机器的按钮开关程序。



操作此按钮开关将产生如下效果：

- 按一次按钮开关，"ON_OFF_Switch" 变量设置为信号状态“1”。
 - 信号流开始传递，并且“输出线圈”指令将 "ON" 变量设置为信号状态“1”。
 - 机器启动。
 - "OFF" 变量的信号状态为“0”，不再起作用。
- 再按一次按钮开关，"ON_OFF_Switch" 变量设置为信号状态“0”。
 - 电流中断，并且“取反线圈”指令将 "OFF" 变量设置为信号状态“1”。
 - 机器关闭。
 - "ON" 变量的信号状态为“0”，不再起作用。

2.5 测试程序

2.5.1 将程序加载到目标系统

简介

以下步骤介绍了如何将程序加载到 PLC 中。加载期间，在编程设备 (PG) 或编程计算机 (PC) 与 PLC 之间会建立在线连接。执行加载时，会将存储在编程设备 (PG) 或编程计算机 (PC) 硬盘中的程序写入到 PLC 的存储器中。期间还会对程序中包含的块进行编译，以便 PLC 能够对其进行处理。编译并加载完程序后，PLC 即能对其进行处理。

说明

在线/离线比较

PLC 不会记录加载过程完成后对编程设备/PC 上的程序所做的更改。您可以使用 TIA Portal 对项目数据进行在线/离线比较以显示所有偏差。在在线模式下，可以借助项目树中的图标来确定编程设备/PC 上的“离线”程序组件与 PLC 上的“在线”编程元素是否完全相同。要刷新 PLC 中程序的状态，必须重新加载程序。

要求

已建立编程设备/PC 和 PLC 之间的连接。

步骤

要将程序加载到 PLC 中，请按以下步骤操作：

1. 启动加载过程。



2. 选择用于连接设备的接口。 激活“显示所有可访问设备”(Display all accessible devices) 复选框。

将在“目标子网中的可访问设备”(Accessible devices in target subnet) 下显示所有可通过所选接口访问的设备。 选择 PLC 并加载用户程序。



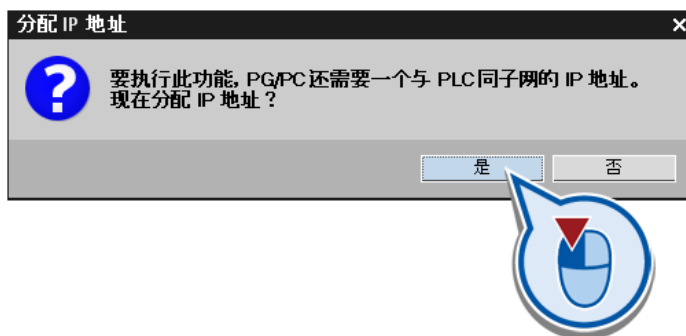
说明

检查在线连接

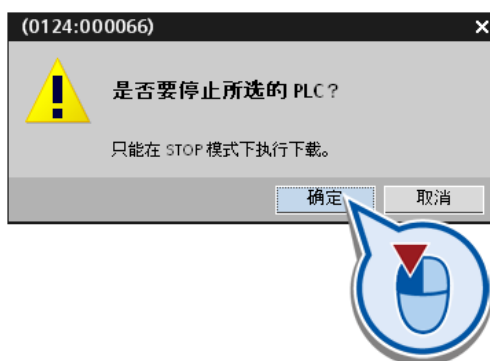
如果“目标子网中的可访问设备”(Accessible devices in target subnet) 列表中未显示任何设备，则可能是如下原因：

- PLC 的硬件连接存在问题。
- 编程设备/PC 上的以太网接口存在问题。
- PLC 的 IP 地址必须与编程设备/PC 的 IP 地址位于同一个子网。

3. 如果尚未分配 IP 地址，请确认分配正确的 IP 地址。



4. 如果 PLC 处于“RUN”模式，请在下面的窗口中将 PLC 设置为“STOP”模式。



“下载预览”(Download preview) 对话框将打开。

5. 如果已配置模块与目标模块之间存在差异，请激活相应复选框接受差异。单击“加载”(Load) 按钮。确保已选中“继续”(Continue) 复选框。

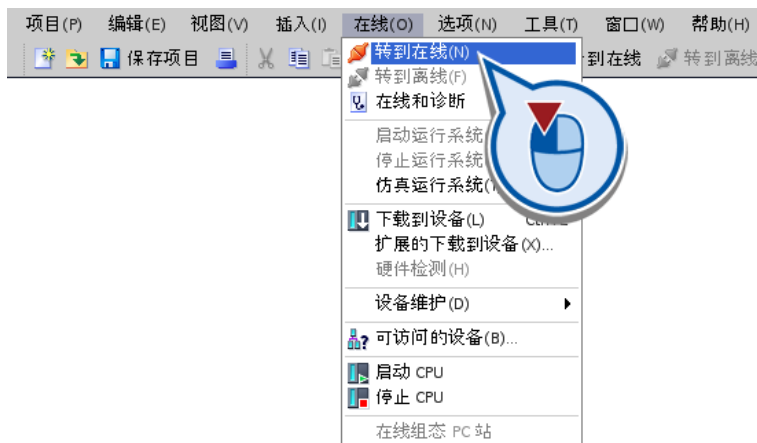


随即会加载程序并显示加载过程中的操作。下载过程完成后，“下载结果”(Download results) 对话框将打开。

6. 启动模块。



7. 激活在线连接。



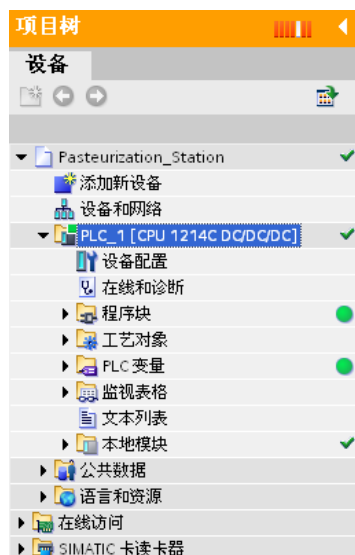
说明

更正编译错误

如果编译过程中检测到程序存在错误，可以在巡视窗口中的“信息 > 编译”(Info > Compile) 下显示这些错误和说明以进行更正。

结果

将程序加载到了 PLC 中。程序组件的状态显示在项目树中。绿色图标表示“离线”和“在线”程序元素相同。在相应的工具提示中可找到其它状态图标的含义。



说明

下载块

从项目树加载块并不是将块传送到 PLC 的唯一方法。可以通过拖放操作将块保存到项目树的可访问设备列表中。如果使用工具栏中的“加载到设备”(Load to device) 按钮，则会加载编辑器中打开的块或项目树中选定的块。

其它信息

可在以下文档中获取有关建立和组态以太网连接的信息：

- “S7-1200 可编程控制器”系统手册
- TIA Portal 信息系统中的“设置以太网接口的参数”一章
- TIA Portal 信息系统中的“组态工业以太网”一章

有关在编程设备/PC 中建立以太网接口的信息，可参见所用操作系统的文档和网卡的文档。

2.5.2 使用程序状态测试程序

简介

以下步骤介绍了如何使用程序状态测试已创建的程序。如果显示程序状态，则可以监视程序的执行。可以从程序中某一特定位置开始启用状态，然后可以获得各变量的值的总览和逻辑运算的结果。利用它可以检查是否正确控制了自动化系统的组件。

程序状态的显示是周期性更新的。它从所选程序段开始。

在程序状态中，可以使用快捷菜单中的“修改”(Modify) 命令执行下列操作之一以将值分配给变量：

- 修改为 1：使用此命令可将 BOOL 数据类型的变量设置为信号状态“1”。
- 修改为 0：使用此命令可将 BOOL 数据类型的变量设置为信号状态“0”。
- 修改值：可以为非 BOOL 数据类型的变量输入修改值。

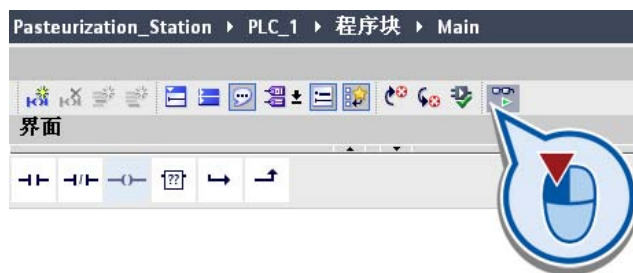
要求

- 已组态 PLC。
- PLC 的输入和输出无电压，因为修改的值会被在线模式下的模块覆盖。
- 组织块“Main [OB1]”已在程序编辑器中打开。

步骤

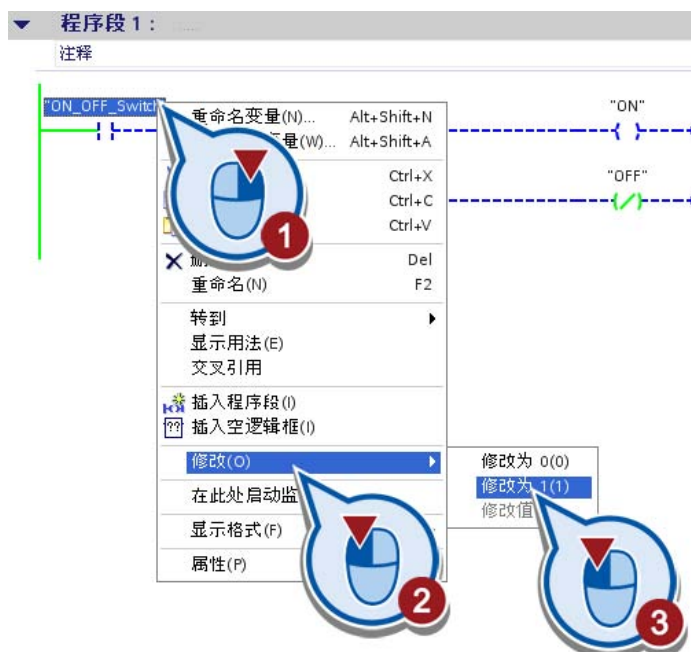
要使用程序状态测试已创建的程序，请按以下步骤操作：

1. 启用监视。

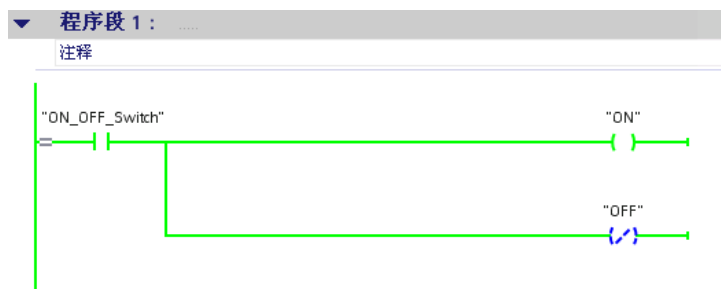


将显示程序状态。

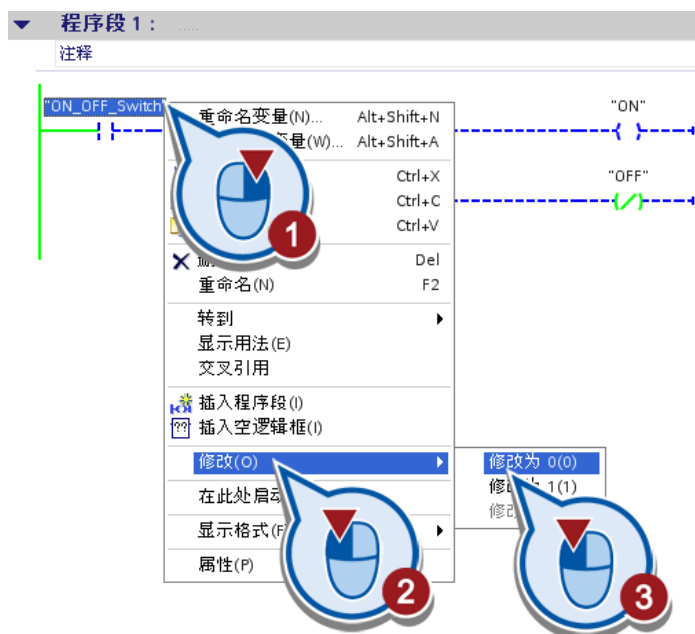
2. 将变量 "ON_OFF_Switch" 修改为“1”。



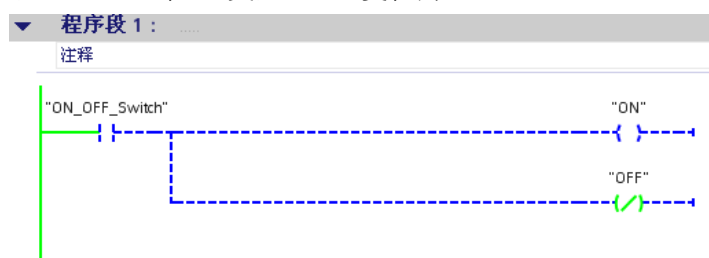
如果变量 "ON_OFF_Switch" 设置为信号状态“1”，则常开触点闭合。电流通过常开触点流到程序段末尾的线圈。电流通过电流路径上的绿色来指示。如果变量 "ON" 置位，实例机器将启动。"OFF" 变量的信号状态为“0”，不再起作用。此状态由蓝色虚线指示。



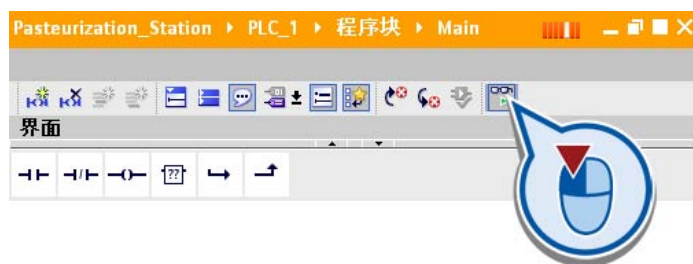
3. 将变量 "ON_OFF_Switch" 修改为“0”。



变量 "ON_OFF_Switch" 复位为信号状态“0”。流向程序段末尾的线圈的电流中断。变量 "OFF" 置位。变量 "ON" 复位为“0”。



4. 禁用监视。



5. 断开在线连接。

结果

已测试程序，并检查了程序是否能够正确执行。

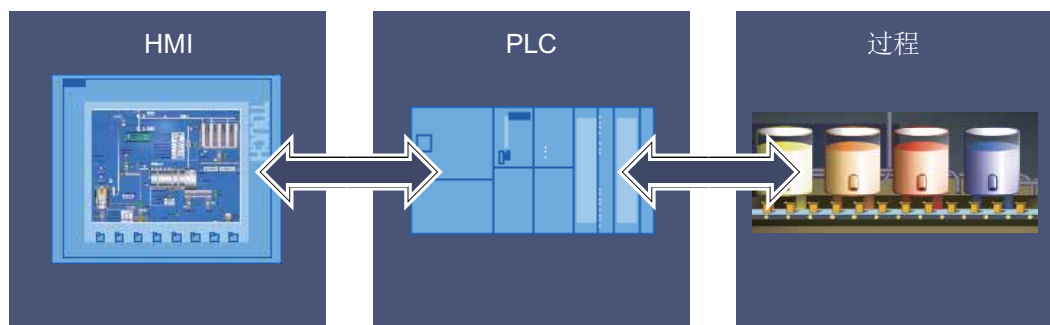
在下一部分中，您将通过编程设计一个 HMI 画面来可视化程序步骤。

2.6 创建 HMI 画面

2.6.1 TIA Portal 中的可视化

HMI - 人机界面

HMI 系统相当于用户和过程之间的接口。过程操作主要由 PLC 控制。用户可以使用 HMI 设备来监视过程或干预正在运行的过程。



以下选项可用于操作和监视机器与工厂：

- 显示过程
- 操作过程
- 输出报警
- 管理过程参数和配方

2.6.2 创建带 HMI 画面的 HMI 设备

简介

以下步骤介绍如何创建新 HMI 设备以及 HMI 画面的模板。

要求

- 已创建程序。
- 已打开项目视图。

添加新的 HMI 设备

要添加 HMI 设备，请按以下步骤操作：

1. 使用项目树添加一个新设备。



2. 指定名称并选择一个 HMI 设备。保留“启动设备向导”(Start device wizard) 复选框为选中状态。

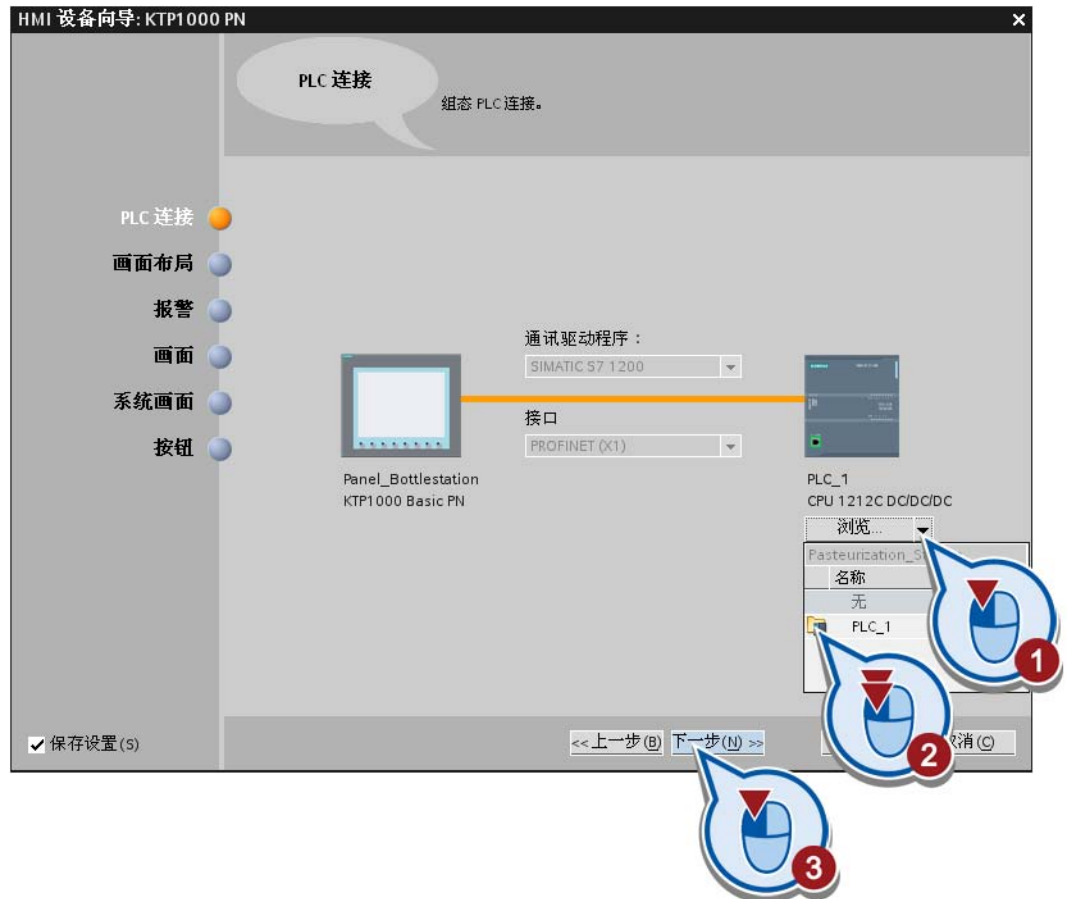


创建 HMI 画面的模板

创建完 HMI 设备后，将打开 HMI 设备向导。HMI 设备向导以“PLC 连接”(PLC connections) 对话框开始。

要创建 HMI 画面的模板，请按以下步骤操作：

1. 组态与 PLC 的连接。

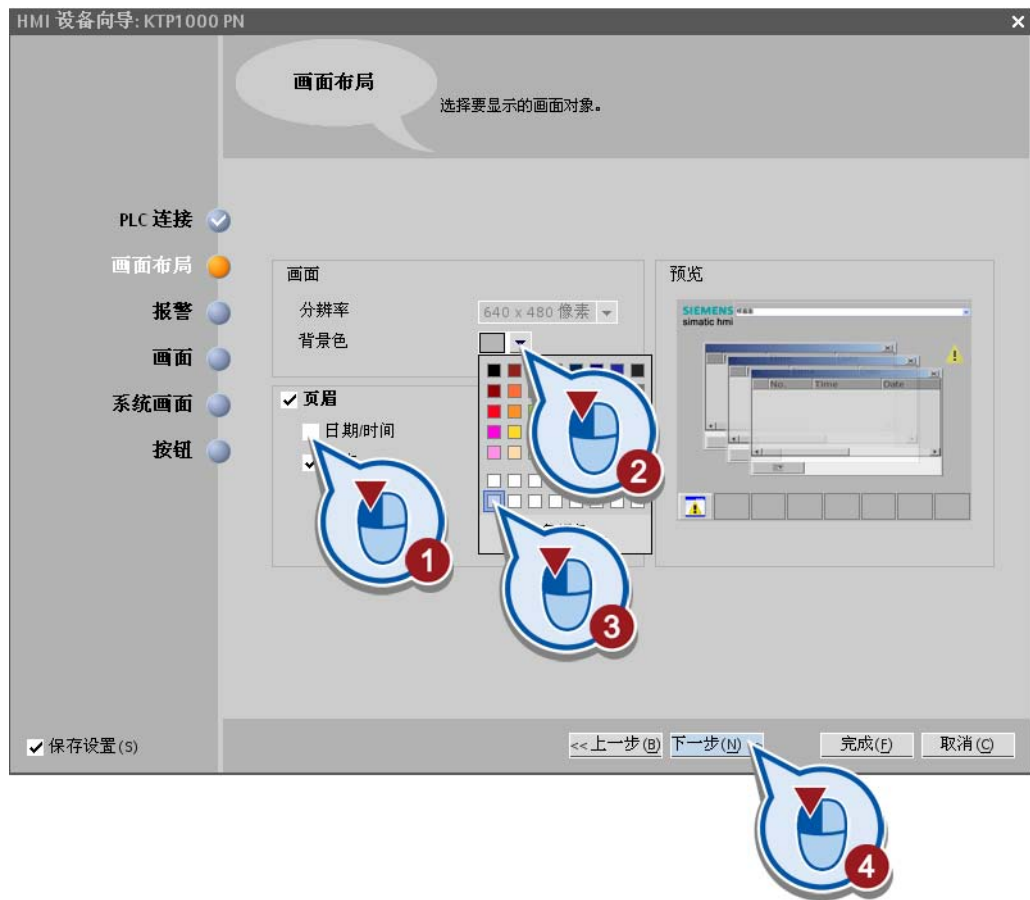


说明

可以随后组态 PLC 的连接

也可以在“设备和网络”(Devices & Networks) 下组态 HMI 设备与 PLC 之间的连接。如果在该对话框中组态连接，将自动建立连接。

2. 选择模板的背景色和页眉的构成元素。

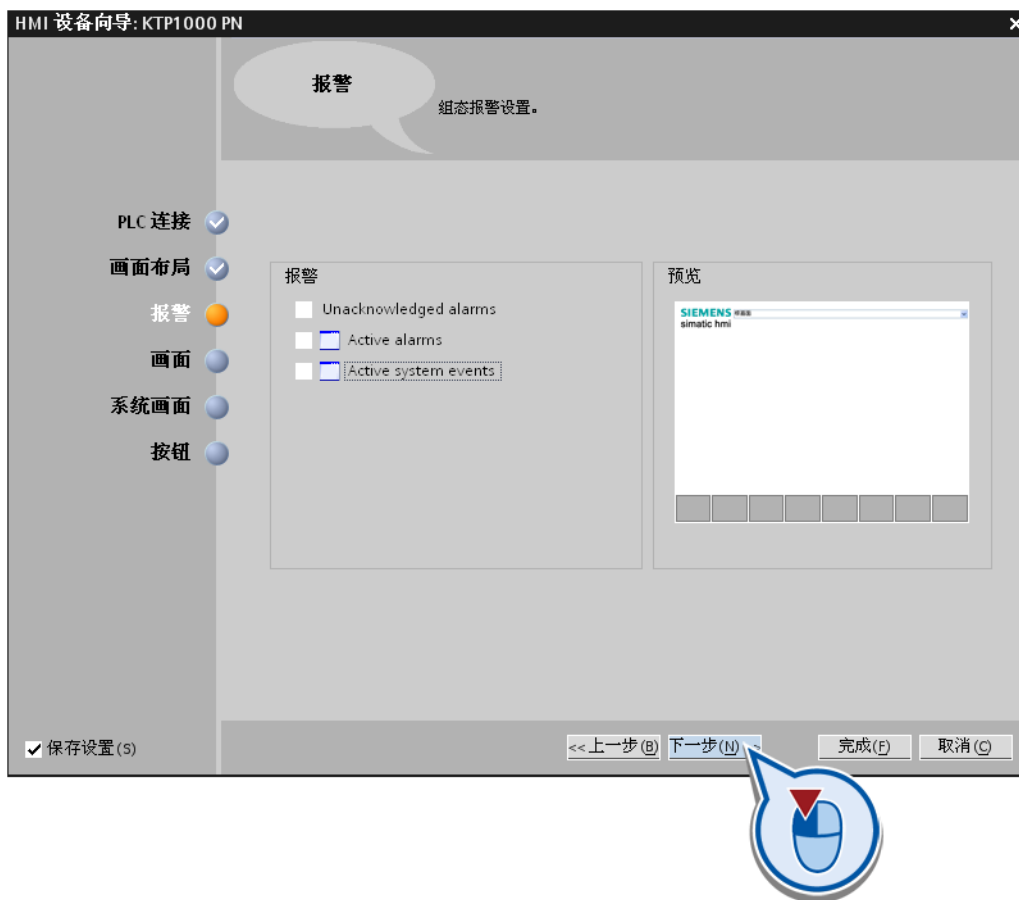


说明

可以随后更改显示

可随后于 HMI 画面的模板中更改在此处所做的显示设置。

3. 禁用报警。对于此实例项目来说，无需使用报警。



说明

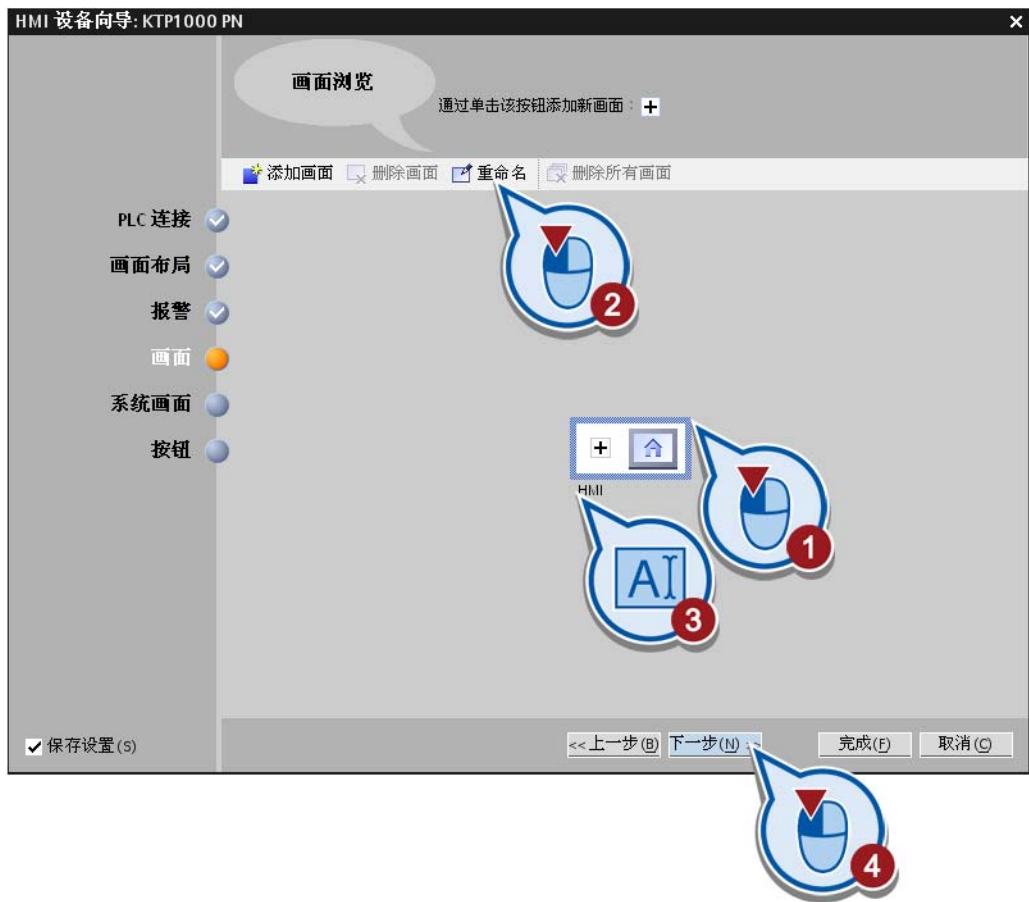
报警

如果通过 HMI 设备向导启用报警，则可以通过 HMI 设备输出报警。在此处所创建的报警窗口将创建在“画面管理”下的全局画面中。报警的用途有很多，例如可在超出限制值时通过 HMI 设备输出警告。再比如，可以通过附加信息对报警内容进行补充，从而可以更容易地定位系统中的故障。用户自定义报警与系统报警之间的基本区别是：

- 用户自定义报警用于监视机器过程。
- 系统报警会被导入到项目中并且包含所用 HMI 设备的状态信息。

TIA Portal 的信息系统中提供了关于报警的附加信息。

- 4. 将以后要在其中创建图形元素的画面重命名为 "HMI"。



说明

画面导航

可以使用该对话框在更广泛的项目中创建画面并建立画面导航。用于在画面之间导航的按钮是自动创建的。

5. 禁用系统画面。对于此实例项目来说，无需使用系统画面。

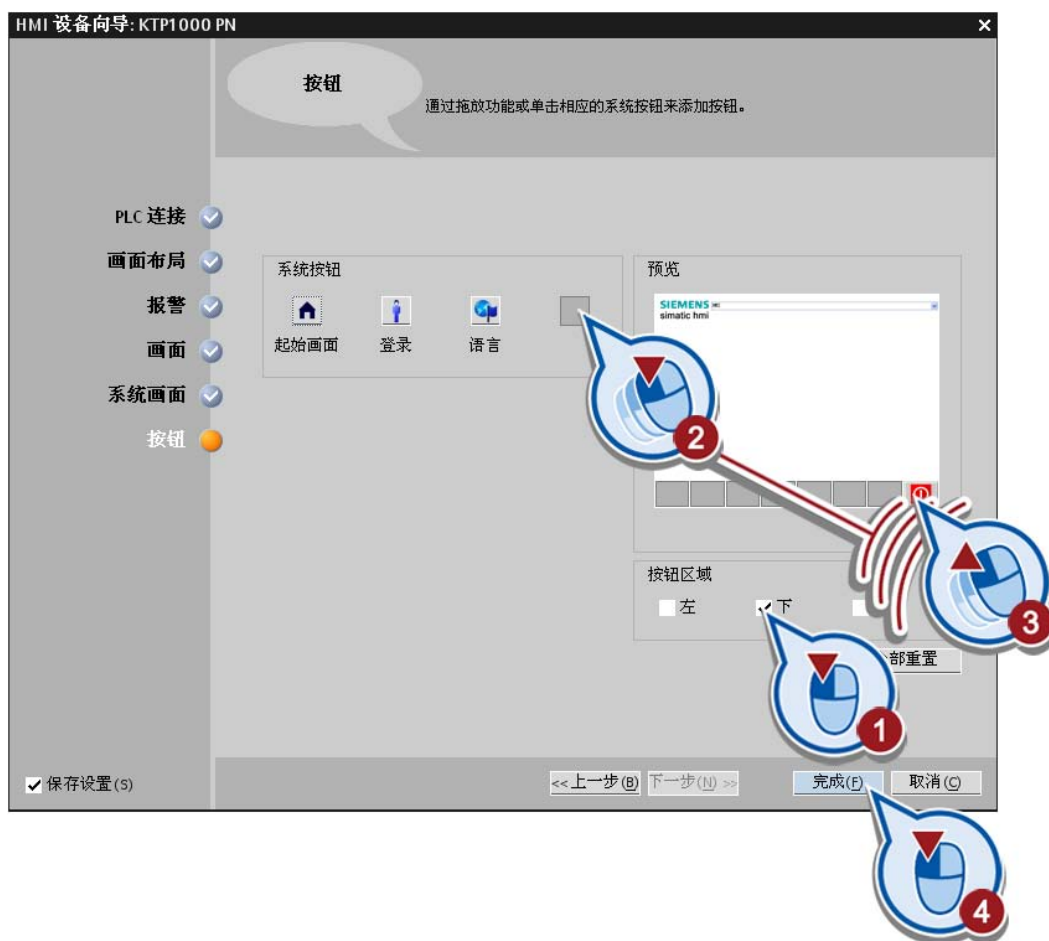


说明

系统画面

可将系统画面作为 HMI 画面使用以设置项目、系统和操作信息以及用户管理。与画面导航一样，用于在主画面和系统画面之间导航的按钮也是自动创建的。

6. 启用下面的按钮区域并插入“退出”(Exit) 按钮。可使用该按钮来终止运行系统。



7. 单击工具栏上的“保存”(Save) 按钮以保存该项目。

结果

您已在项目中创建了一个 HMI 设备并为 HMI 画面创建了一个模板。在项目视图中，创建的 HMI 画面将显示在编辑器中。

在下一部分中，将介绍如何在 HMI 画面中创建画面对象以使编程过程可视化。

2.6.3 什么是图形对象？

简介

可以使用 **TIA Portal** 创建用于操作和监视机器与工厂的画面。预定义的对象可协助您创建这些画面；可以使用这些对象仿真机器、显示过程和定义过程值。HMI 设备的功能决定了 HMI 中的项目可视化和图形对象的功能范围。

图形对象

图形对象是所有可用于 HMI 中项目可视化的元素。例如，这些对象包括：用于可视化机器部件的文本、按钮、图表或图形。

图形对象的用途

图形对象可进行静态可视化或借助变量用作动态对象：

- 运行系统中的静态对象不会发生改变。在“入门指南”项目中，传送带是作为静态对象创建的。
- 动态对象会根据过程改变。用户可通过以下变量来可视化当前过程值：
 - PLC 存储器中的 PLC 变量
 - 以字母数字、趋势图和棒图形式显示的 HMI 设备存储器中的内部变量。

动态对象还包括 HMI 设备中的输入域，用以通过变量在 PLC 和 HMI 设备之间交换过程值和操作员输入值。

2.6.4 创建和组态图形对象

2.6.4.1 “设备开/关”按钮

简介

以下步骤将介绍如何创建“设备开/关”(Machine ON/OFF) 按钮以及如何通过外部 HMI 变量将其连接到 PLC 变量 "ON_OFF_Switch"。可以使用此方法通过 HMI 画面修改 PLC 变量的过程值。

外部 HMI 变量

可使用外部 HMI 变量访问 PLC 地址。例如，这允许您通过 HMI 设备输入过程值或通过按钮直接修改控制程序的过程值。可通过链接到 HMI 设备的 PLC 中的 PLC 变量表来进行寻址。PLC 变量通过符号名称链接到 HMI 变量。这意味着不必在更改 PLC 变量表中的地址时调整 HMI 设备。

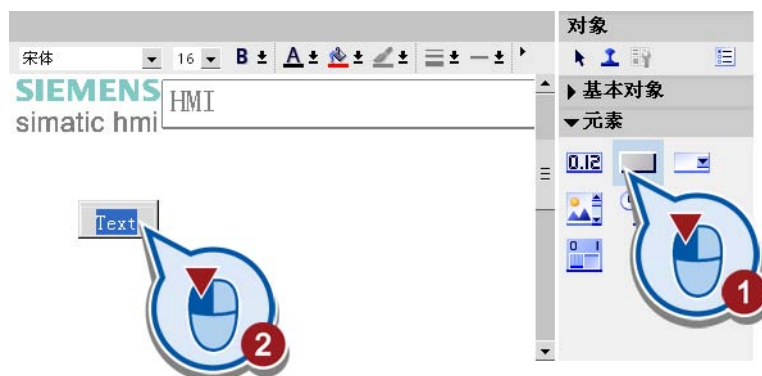
要求

HMI 画面处于打开状态。

步骤

要将“设备开/关”(Machine ON/OFF) 按钮与 PLC 变量 "ON_OFF_Switch" 连接，请按以下步骤操作：

1. 删除 HMI 画面中的标准文本域“欢迎进入...”(Welcome...)。
2. 创建一个按钮。



3. 在巡视窗口中，选择“按内容调整对象大小”(Fit object to contents) 选项，以根据文本长度自动调整按钮的大小。

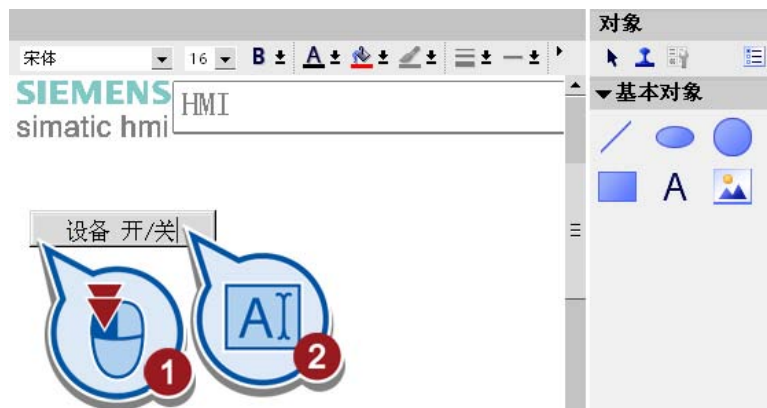


说明

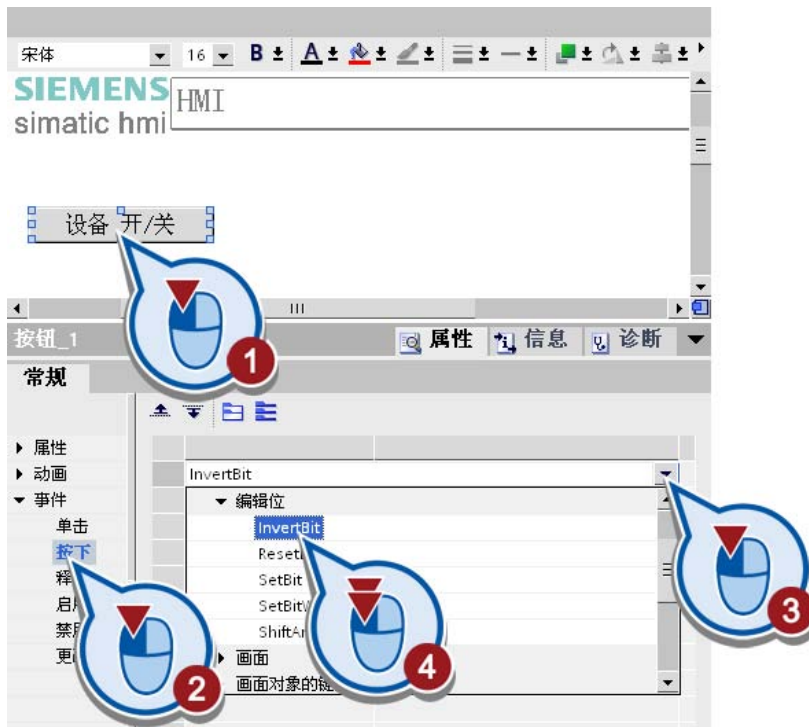
“按内容调整对象大小”功能

特别是以后在带有 HMI 画面语言选择的项目中工作时，可以使用该功能。根据所选择的语言，翻译文本可能会短于或长于原始文本。可使用该功能以确保按钮标签不会被截断。当原始文本中的文本大小发生变化时，按钮的大小会自动调整。

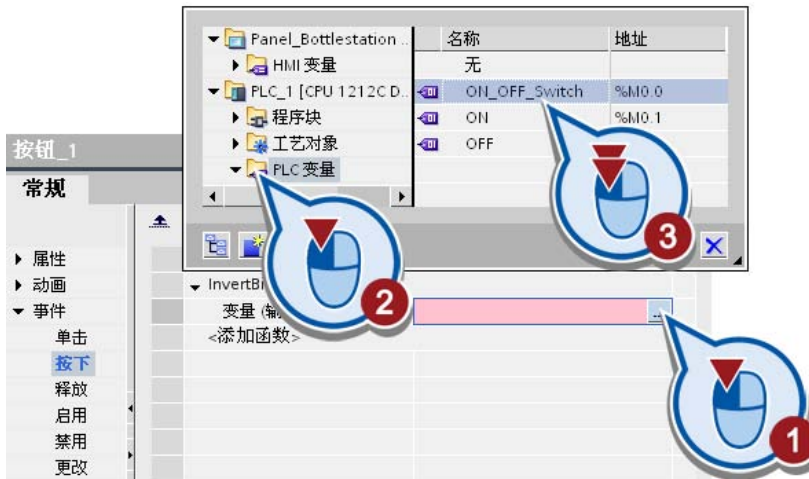
4. 使用文本“设备开/关”(Machine ON/OFF) 来标记该按钮。



- 5. 将“取反位”(InvertBit) 函数分配给该按钮的触发事件“按下”(Pressing)。



- 6. 将“取反位”(InvertBit) 函数与 PLC 变量 "ON_OFF_Switch" 链接。



说明

HMI 连接在 TIA Portal 中是自动创建的

如果未对 HMI 设备和 PLC 之间的连接进行组态，则只要将 PLC 变量链接到 HMI 对象，就会自动建立连接。

结果

已经将“设备开/关”(Machine ON/OFF) 按钮与 PLC 变量 "ON_OFF_Switch" 连接。当您按下 HMI 设备上的该按钮时，PLC 变量的位值将被设置为“1”（设备启动）。当您再次按下该按钮时，PLC 变量的位值将被设置为“0”（设备关闭）。

2.6.4.2 图形对象“LED”

简介

以下步骤将介绍如何使用“圆”对象来设置两种状态 LED（红色/绿色）以及如何根据 PLC 变量“ON_OFF_Switch”的值使其动态化。

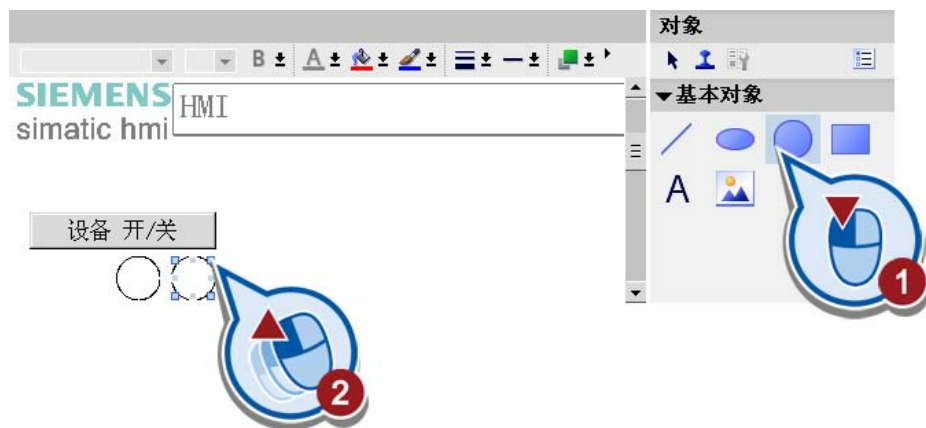
要求

HMI 画面处于打开状态。

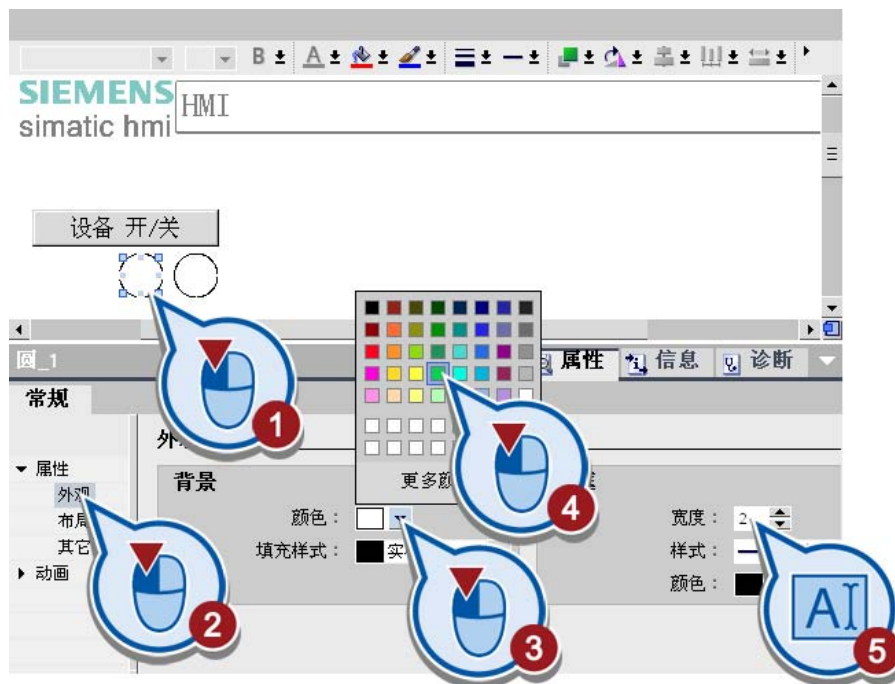
步骤

要创建 LED 并使其动态化，请按以下步骤操作：

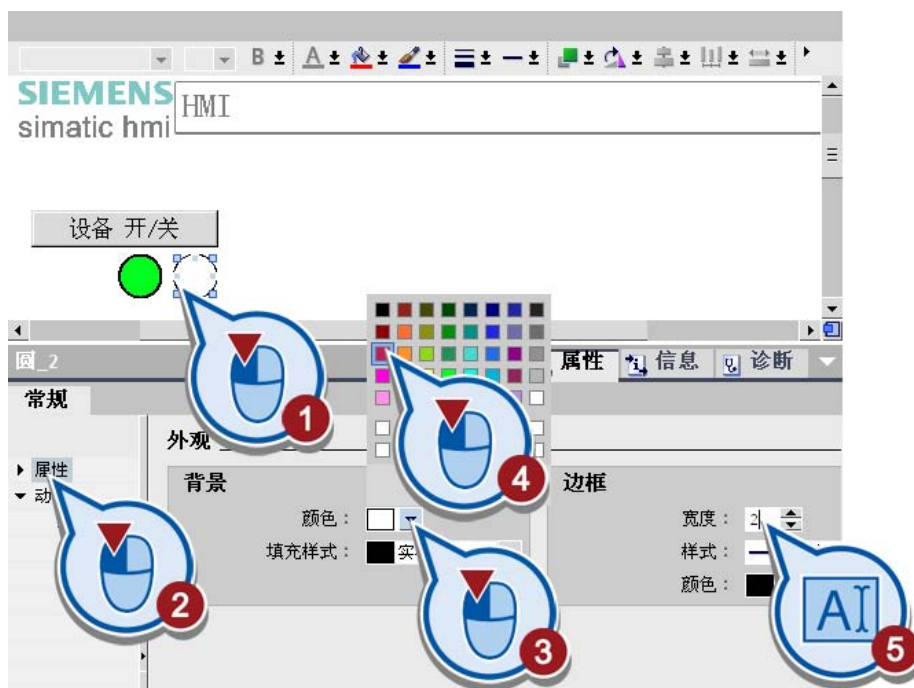
1. 按住 Shift 键，在“设备开/关”(Machine ON/OFF) 按钮的下面绘制两个圆。



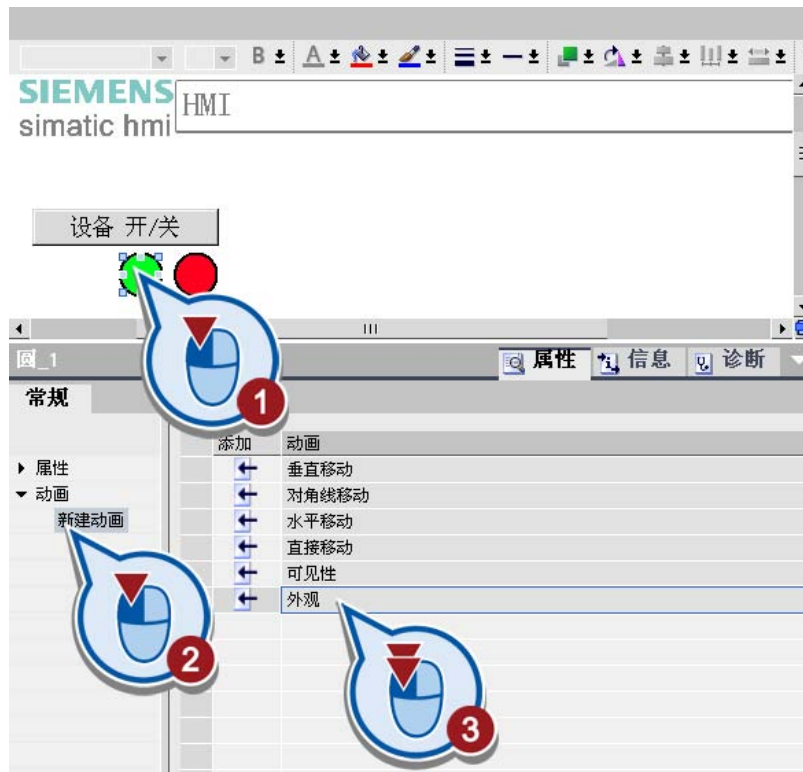
2. 将背景色绿色和宽度为“2”的边框分配给第一个圆。



3. 将背景色红色和同样宽度为“2”的边框分配给第二个圆。



4. 为绿色 LED 创建一个类型为“外观”(Appearance) 的新动画。

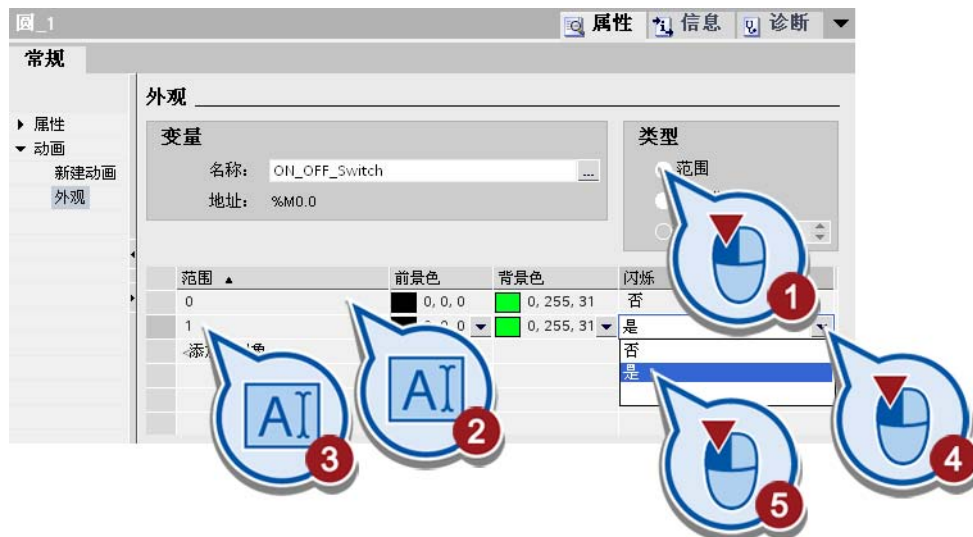


5. 将该动画链接到 PLC 变量“ON_OFF_Switch”。

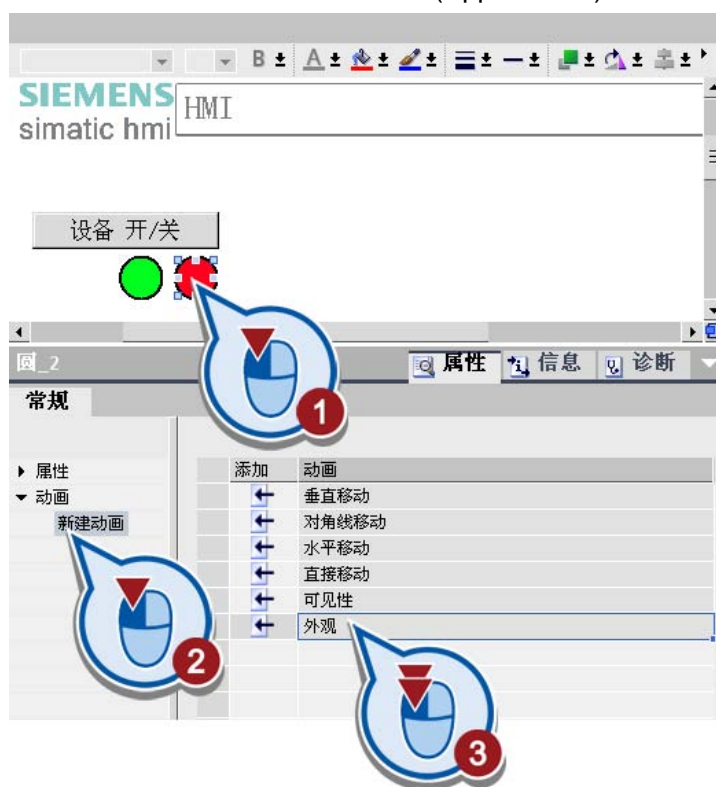


2.6 创建 HMI 画面

6. 改变 LED 的外观以反映该 PLC 变量的状态。只要控制程序将 PLC 变量的位值设置为“1”，LED 就会闪烁。



7. 为红色 LED 创建一个类型为“外观”(Appearance) 的新动画。



8. 同时将该动画链接到 PLC 变量“ON_OFF_Switch”。



9. 改变 LED 的外观以反映该 PLC 变量的状态。只要控制程序将 PLC 变量的位值设置为“0”，LED 就会闪烁。



结果

使用图形对象“圆”创建了状态 LED 并使其动态化。在初始状态下，红色 LED 闪烁。

- 如果通过“设备开/关”(Machine ON/OFF) 按钮启动控制程序，则会将变量“ON_OFF_Switch”的位值设置为“1”并且绿色 LED 闪烁。
- 当再次按下“设备开/关”(Machine ON/OFF) 按钮停止控制程序时，会将变量“ON_OFF_Switch”的位值设置为“0”并且红色 LED 闪烁。

在下一部分，将介绍如何创建图形对象“传送带”。

2.6.4.3 图形对象“传送带”

简介

以下步骤将介绍如何将逻辑运算链接到图形文件夹以导入图形对象。通过逻辑运算导入图形对象“传送带”(Conveyor.Simple.wmf)。

要求

HMI 画面处于打开状态。

步骤

要导入图形对象，请按以下步骤操作：

1. 将位于以下 Internet 地址中的 ZIP 文件“WinCC Graphics”复制到您的本地硬盘并提取该文件。

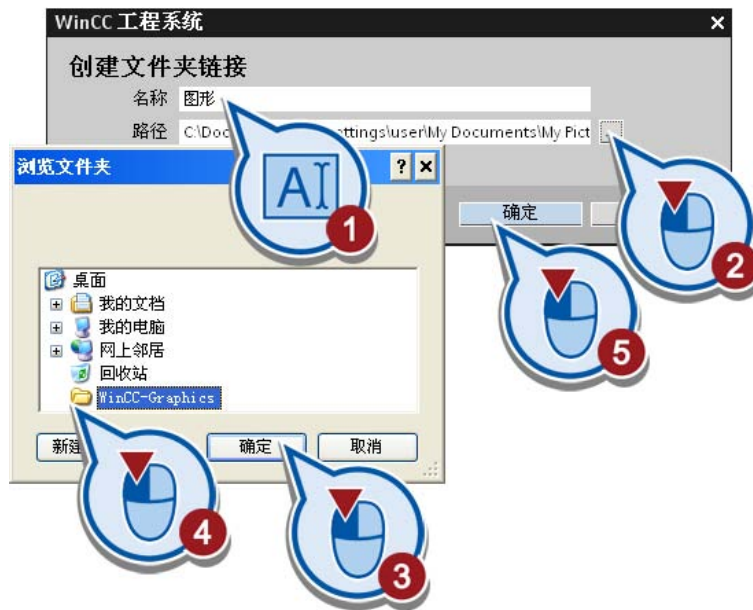
<http://support.automation.siemens.com/CN/view/zh/40263542>

请单击“信息”图标查看相关 ZIP 文件。

2. 打开“工具箱”(Toolbox) 任务卡中的“图形”(Graphics) 窗格并创建一个新的链接。



3. 为该链接给定一个名称，然后选择之前提取的文件夹“WinCC Graphics”。

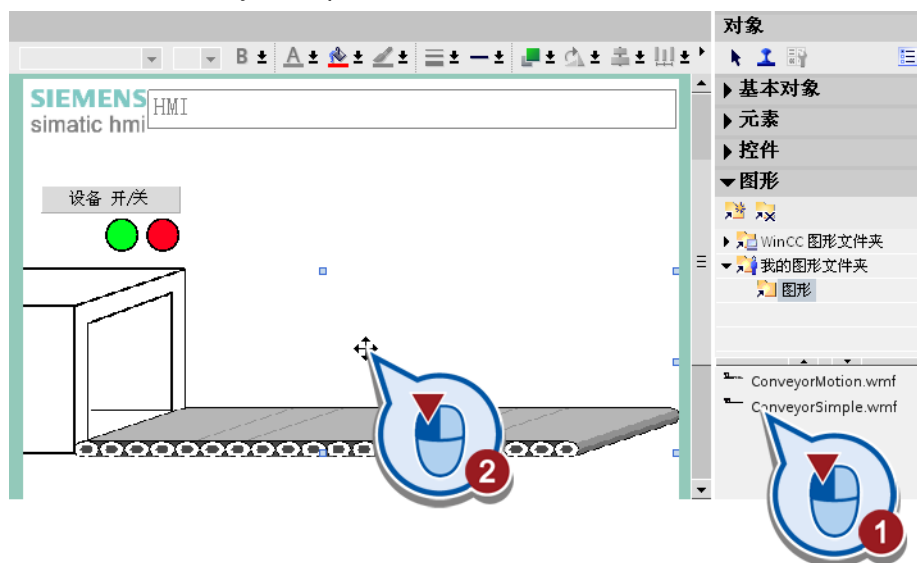


在新创建的链接下将显示两个图形。

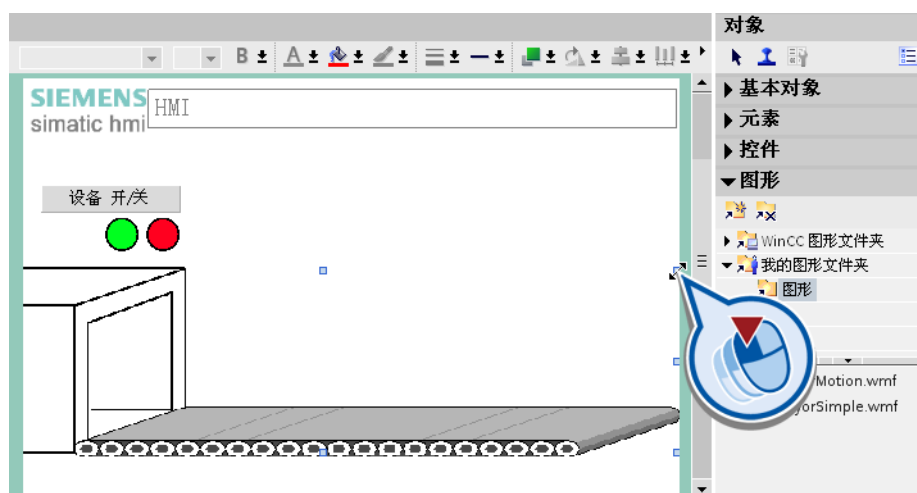
4. 禁用“大图标”(Large icons) 选项。



5. 将图形对象“ConveyorSimple.wmf”放置在 HMI 画面中。



6. 通过缩放调整该图形对象。



结果

已经将静态图形对象“传送带”复制到项目中。移动或删除文件夹“WinCC Graphics”时，只会丢失链接。图形对象仍然保留在项目中。

在下一部分，将介绍如何创建带有运动动画的图形对象“瓶子”。

2.6.4.4 带有运动仿真的图形对象“瓶子”

简介

以下步骤将介绍如何创建带有运动动画的图形对象“瓶子”。在动画中，奶瓶通过传送带从左向右移动。使用一个内部 HMI 变量使对象动态化。

内部 HMI 变量

内部 HMI 变量与 PLC 之间不存在连接。它们存储在 HMI 设备的内存中。只有 HMI 设备能够对这些变量进行读写访问。例如，可通过创建内部 HMI 变量进行独立于控制程序的本地计算。

要求

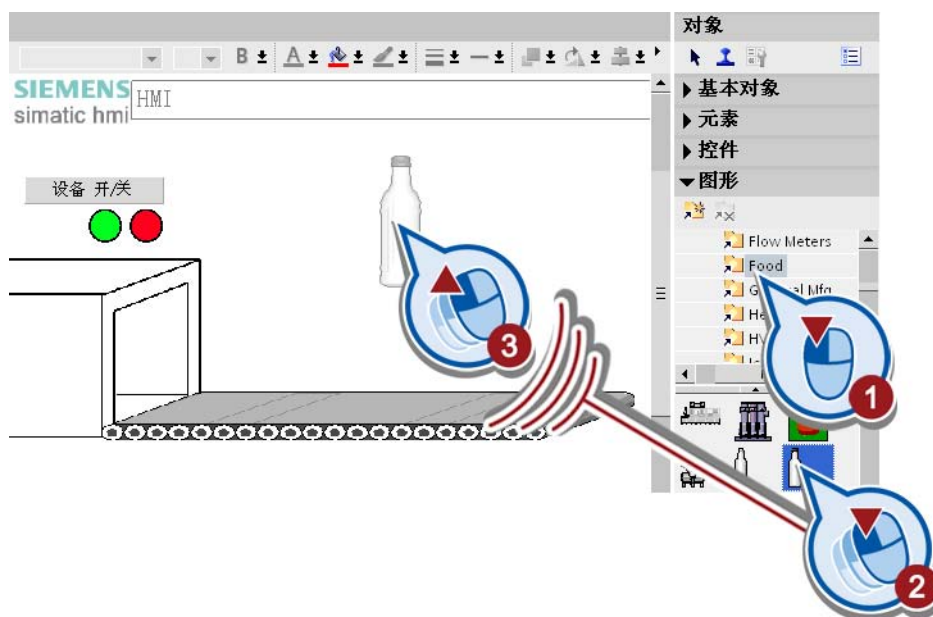
HMI 画面处于打开状态。

步骤

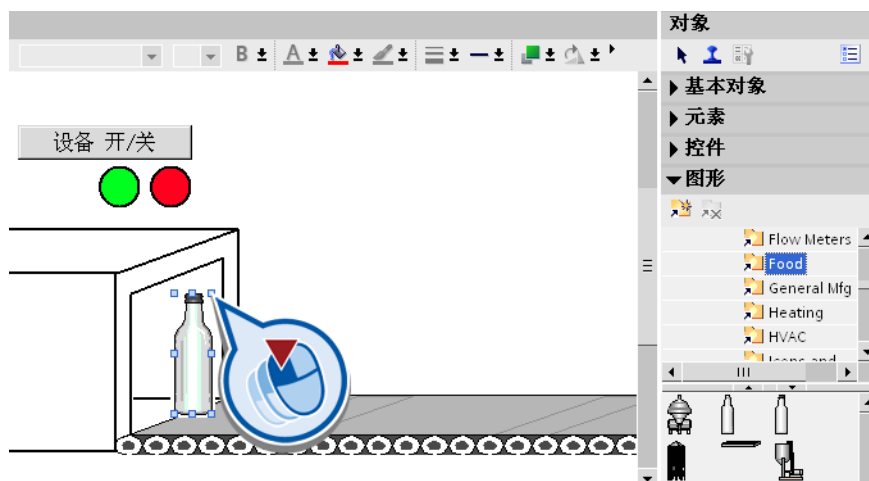
要创建图形对象“瓶子”并组态水平运动，请按以下步骤操作：

1. 通过拖放操作将 WinCC 图形文件夹“符号工厂图形”(Symbol Factory Graphics) >“符号工厂 256 色”(SymbolFactory 256 Colors) >“食品”(Food) 中的图形对象“瓶子”复制到“传送带”对象上方的画面空闲区域。

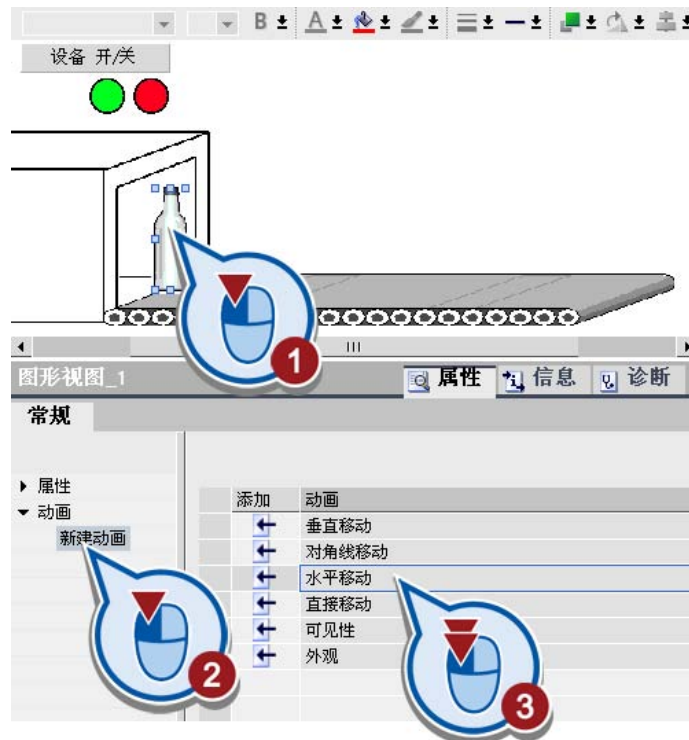
定位瓶子时，请确保将其放在 HMI 画面的空闲区域中。如果直接将瓶子拖动到传送带上，传送带将被瓶子替换。



2. 对瓶子进行缩放以使其高度低于洞的高度。

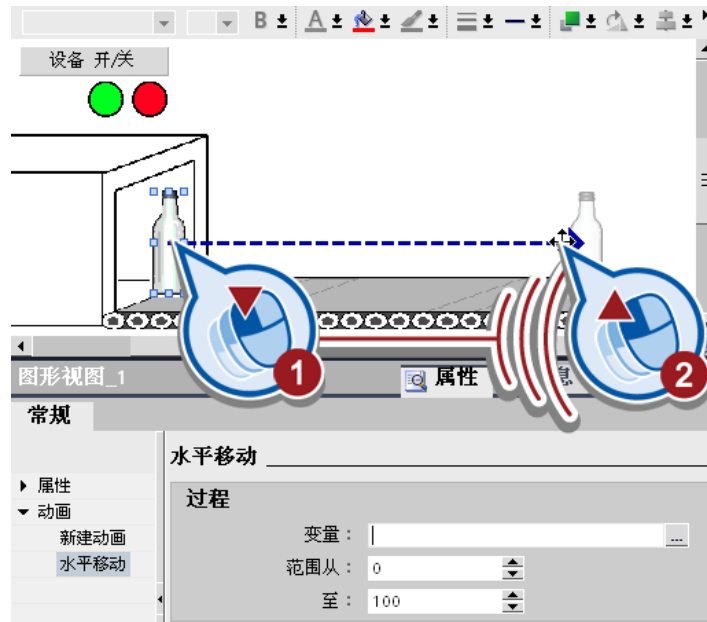


3. 为图形对象“瓶子”创建水平运动动画。



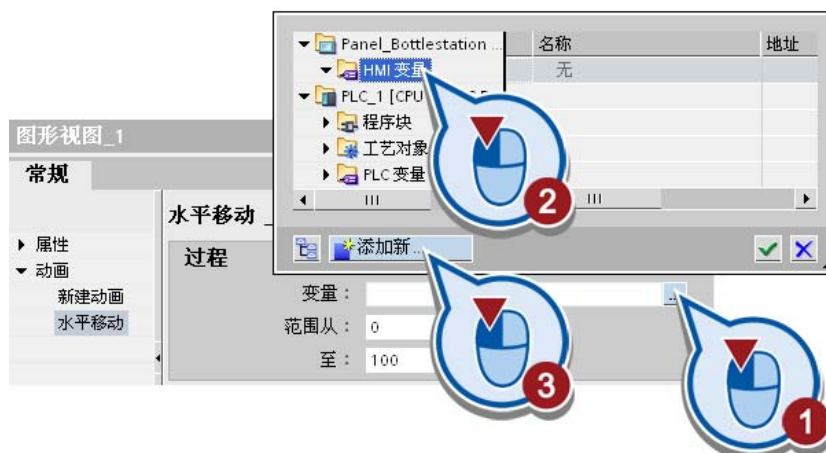
瓶子的透明副本显示在工作区中，该副本通过箭头连接到源对象。

4. 将透明的瓶子移动到传送带的末端。



系统在巡视窗口中自动输入最后位置的像素值。

5. 在巡视窗口中为运动动画创建一个新的 HMI 变量。



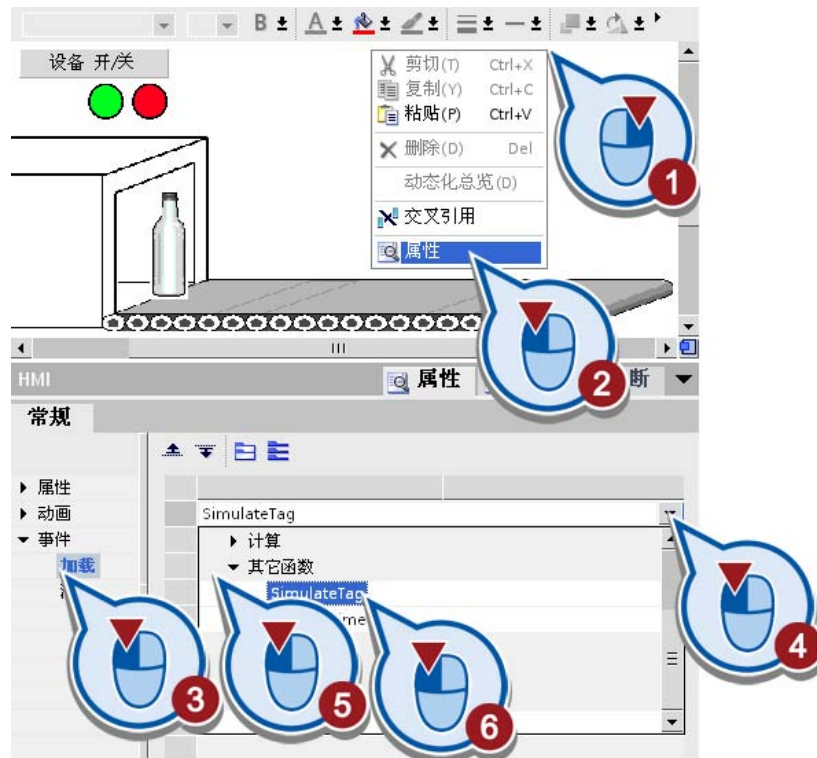
6. 使用“Position_Bottle”作为该变量的名称，使用“Short”作为数据类型。



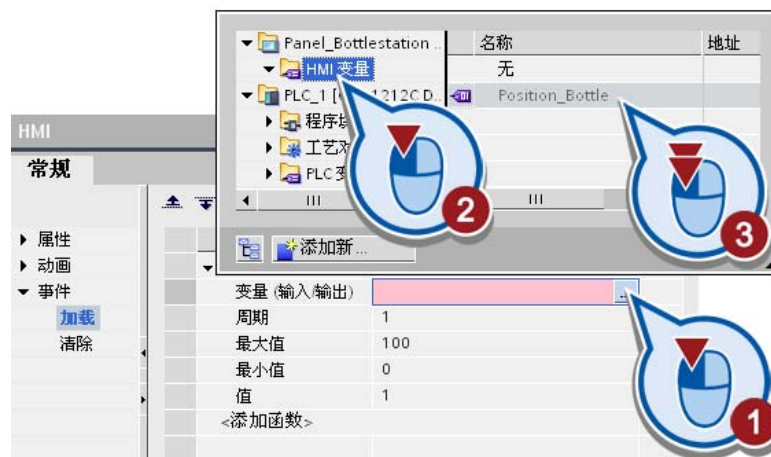
将瓶子的位置链接到该变量。如果当前程序中的变量值发生了改变，瓶子的位置也会随之改变。

要仿真运动，变量“Position_Bottle”的值必须自动更改。该变量的值应该在加载 HMI 画面后自动增加。只要值达到 100，它就将重新从 0 开始。变量的数值更改是通过 HMI 画面的属性进行仿真的。

7. 首先，将“仿真变量”(SimulateTag) 功能添加给 HMI 画面的事件“加载”(Loaded)。



8. 将变量“Position_Bottle”分配给“仿真变量”(Simulate Tag) 函数。



9. 保存项目。

结果

已经创建了带有运动动画的图形对象“瓶子”。在将 HMI 画面加载到 HMI 设备时，变量“Position_Bottle”的值会在每个基本周期 (200 ms) 后加一。当值达到 100 时，会将变量值设置为“0”。瓶子的位置取决于变量值。例如，如果变量值为 50，则瓶子位于传送带的中间。

在下一部分，将介绍如何通过“设备开/关”(Machine ON/OFF) 按钮的变量“ON_OFF_Switch”来修改瓶子的可见性。

2.6.4.5 改变运动动画的可见性

简介

加载 HMI 画面时，会自动启动瓶子的运动动画。以下步骤将介绍如何根据 PLC 变量“ON_OFF_Switch”的值来组态画面中动态化瓶子的可见性。

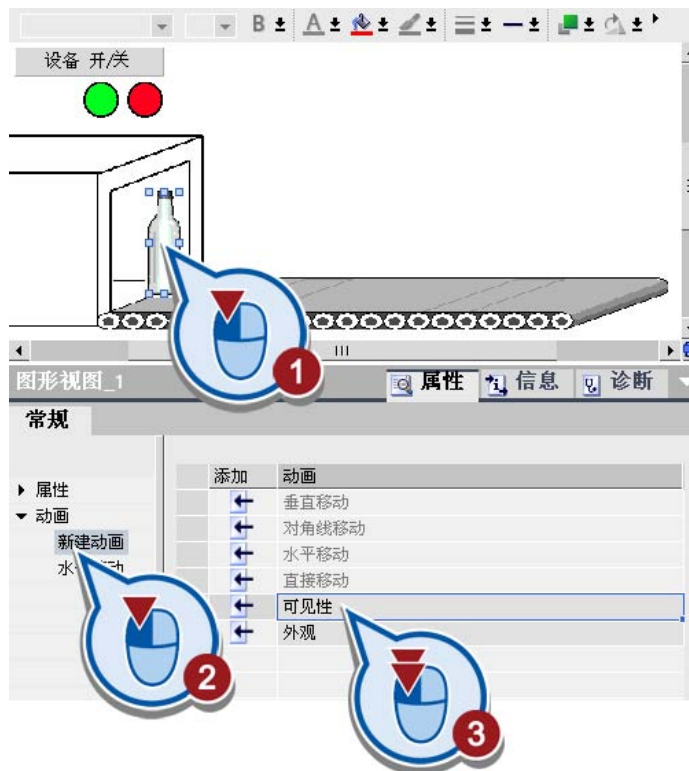
要求

HMI 画面处于打开状态。

步骤

要组态 HMI 画面中瓶子的可见性，请按以下步骤操作：

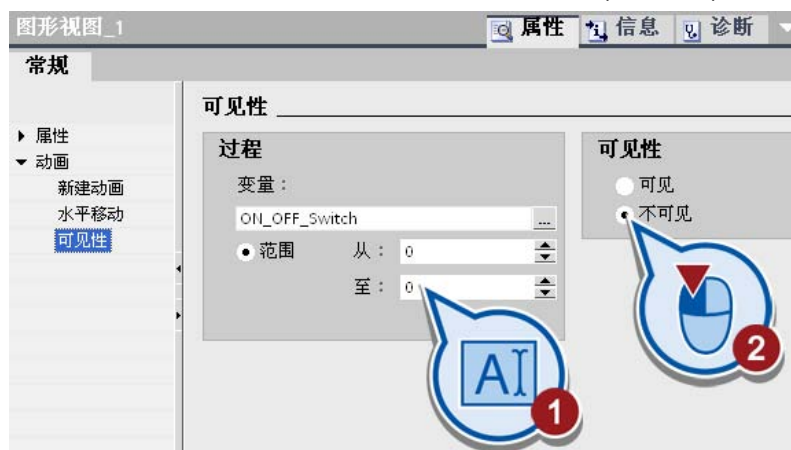
1. 为图形对象“瓶子”创建一个新动画“可见性”。



2. 将 PLC 变量“ON_OFF_Switch”分配给该动画。



3. 将范围为“0”到“0”的变量的可见性切换为“不可见”(Invisible)。



4. 单击工具栏上的“保存”(Save) 按钮以保存该项目。

结果

已经组态了画面中瓶子的可见性。当机器打开并且变量“ON_OFF_Switch”的过程值为“1”时，瓶子是可见的。

2.7 测试 HMI 画面

2.7.1 将 HMI 画面加载到 HMI 设备

简介

可将“入门指南”项目加载到 HMI 设备并在运行系统中执行。为此，组态设备和 HMI 设备之间必须建立连接。如果您没有使用 HMI 设备，则可以在 TIA Portal 中仿真运行系统（请参见 仿真运行系统 (页 92)）。

HMI 设备中的运行系统

HMI 设备用于在过程及生产自动化中操作和监视任务。

如果在“入门指南”项目中使用 HMI 设备，请确保已在 HMI 设备和 PLC 之间建立了连接。

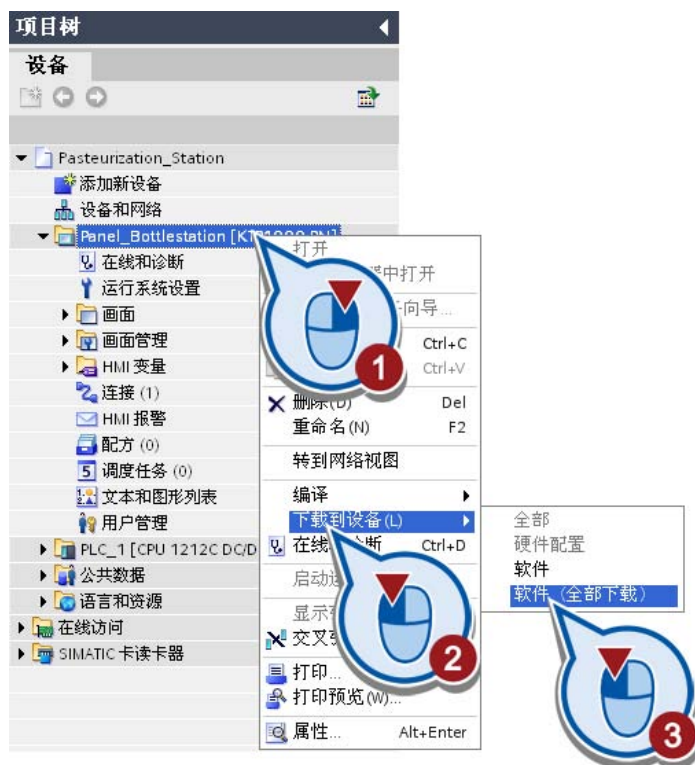
由于“入门指南”项目的 HMI 画面主要使用 PLC 变量，因此只有当 HMI 设备和 PLC 之间已经建立连接时才会执行对象的动画。

要求

- 已经建立了与 HMI 设备的连接。
- 已正确组态 HMI 设备。
- HMI 设备处于传送模式。

步骤

- 1. 启动将软件加载到 HMI 设备的过程。项目在加载过程之前自动编译。



小心

请确保没有任何相关数据已被加载到 HMI 设备。
已加载的软件将被覆盖。

2. 可以覆盖之前加载到 HMI 设备中的软件。



说明

有关 HMI 的信息

有关 HMI 以及如何组态 HMI 设备的更多信息，请参见 TIA Portal 的信息系统和您的 HMI 设备手册。

2.7.2 仿真运行系统

简介

如果您没有使用 HMI 设备，则可以使用运行系统仿真器仿真所有使用的 PLC 变量。

说明

使用变量仿真器启动运行系统仿真

如果在未使用变量仿真器的情况下启动仿真，按钮和控制元素将处于非活动状态。

运行系统仿真器

使用运行系统仿真器仿真独立于程序的已连接 PLC 变量的过程值。可使用运行系统仿真器表选择 PLC 变量并修改它们的值。尽管变量是由运行系统中的 PLC 程序进行设置的，HMI 画面中的对象仍会做出响应。

步骤

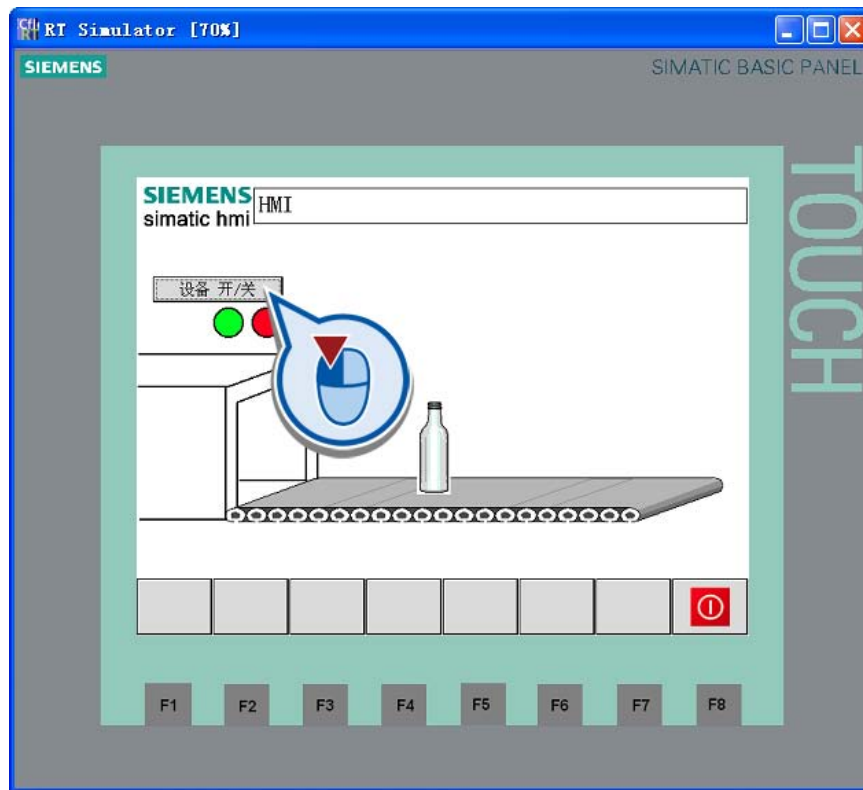
要启动 HMI 画面的仿真，请按以下步骤操作：

1. 通过菜单栏启动运行系统仿真。HMI 窗口必须处于活动状态。如果菜单未激活，则先单击 HMI 画面中的空闲区域。



此时会启动运行系统仿真。启动仿真后，“运行系统仿真器”(RT Simulator) 窗口中将显示 HMI 画面，同时红色 LED 灯会闪烁（机器关闭）。

2. 启动机器。



说明

编译期间的错误消息

如果运行系统仿真由于项目中的错误而无法启动，则相应的错误消息会显示在巡视窗口中的“信息 > 编译”(Info > Compile) 下。双击错误消息时，会自动导航到未正确组态的 HMI 对象。

结果

运动仿真已启动。同时绿色 LED（而不是红色 LED）闪烁。如果再次单击“设备开/关”(Machine ON/OFF) 按钮，瓶子将不再可见，同时红色 LED（而不是绿色 LED）闪烁。要退出运行系统仿真，请关闭窗口或单击“退出运行系统”(Exit runtime) 按钮。

扩展实例

3.1 简介

加载项目

如果跳过了之前的章节，那么可以载入该章节结尾的项目状态（请参见“加载项目 (页 18)”）。本章结尾的项目状态存储在“Simple_Example.ZIP”文件中。

概述

在本实例项目的第二部分，将向您介绍如何创建用于控制简单的牛奶巴氏消毒工作站的附加程序。该工作站由传送带和加热室组成。

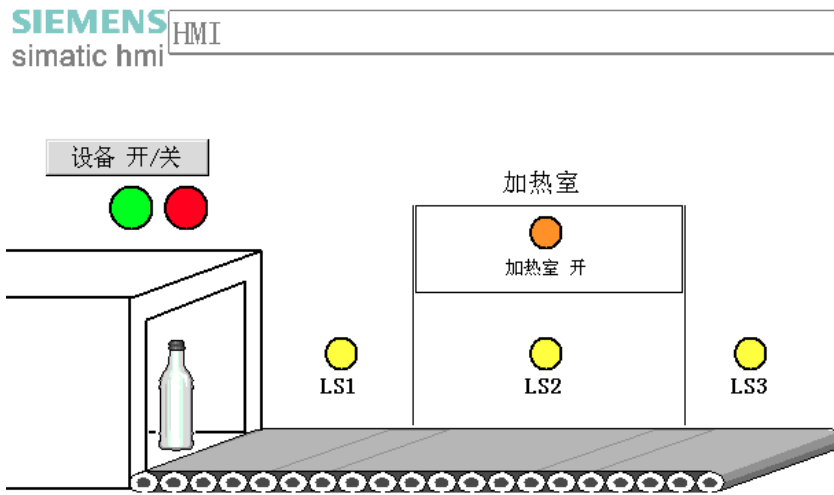
传送带驱动器是由三个光栅进行控制的：

- 当奶瓶位于传送带的起始位置时，会激活第一个光栅（LS1）。此时会启动传送带驱动器并将瓶子移动到加热室中。
- 第二个光栅（LS2）将对加热室中的奶瓶位置进行检测。奶瓶到达预定位置时，它会发出信号使传送带停止并启动加热室中的加热装置。

在加热室中，会以 75°C 的温度对瓶子加热 40 秒。只要正在进行加热，状态指示灯就始终处于点亮状态。加热过程结束时，会重新启动传送带驱动器并将瓶子传送到传送带的末端。

- 当瓶子到达传送带的末端时，第三个光栅（LS3）会被激活并且传送带会停止。随后会由下一个站来收集瓶子。

整个站的启动和关闭都是由在项目的第一部分中编写的按钮来控制的。



步骤

以下工作步骤是为项目的第二部分规划的：

- 扩展程序
通过四个附加程序段来扩展程序。
- 将项目加载到 PLC
将扩展项目加载到 PLC 并在在线视图中监视程序的执行。
- 使用程序状态测试程序
使用程序状态来测试程序的执行。
- 扩展 HMI 画面
要使程序扩展可视化，可将其它图形对象插入现有的 HMI 画面。
- 仿真 HMI 画面
在运行系统模式中执行 HMI 画面。

3.2 扩展程序

3.2.1 在 PLC 变量表中定义变量

简介

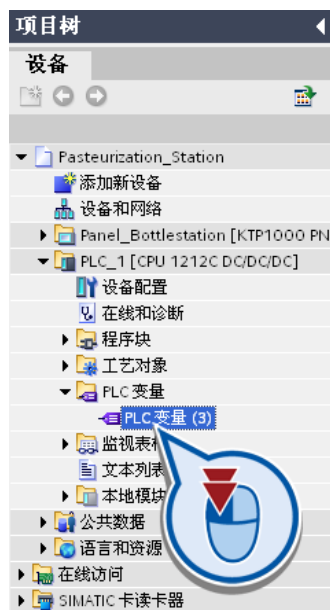
以下步骤将介绍如何定义扩展程序所需的变量。还将介绍如何在 PLC 变量表中创建变量。

有关 PLC 变量和 PLC 变量表的基本信息，请参见“什么是变量？ (页 46)”部分。

步骤

要定义所需的变量，请按以下步骤操作：

1. 打开 PLC 变量表。



PLC 变量表在工作区中打开。变量表中列出在项目的第一部分中所定义的变量。

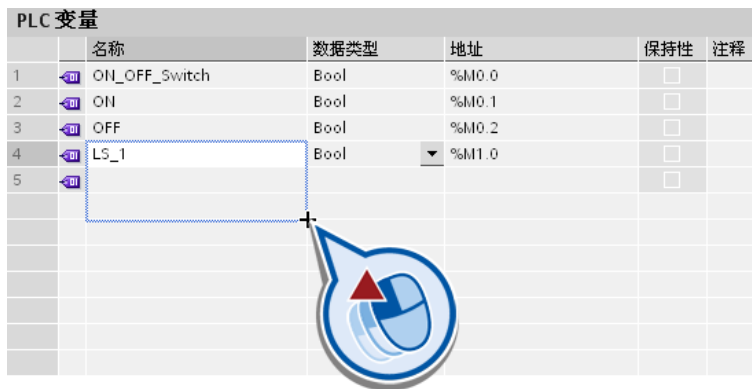
2. 定义变量“LS_1”与地址“M1.0”。



3. 选择“名称”(Name) 为“LS_1”的变量的单元格并单击单元格右下角的填充标记。



4. 将单元格的内容传送给下面的两个单元格。



传送内容时使用连续编号。

5. 定义“conveyor_drive_ON_OFF”变量。

PLC 变量						
	名称	数据类型	地址	保持性	注释	
1	ON_OFF_Switch	Bool	%M0.0	<input type="checkbox"/>		
2	ON	Bool	%M0.1	<input type="checkbox"/>		
3	OFF	Bool	%M0.2	<input type="checkbox"/>		
4	LS_1	Bool	%M1.0	<input type="checkbox"/>		
5	LS_2	Bool	%M1.1	<input type="checkbox"/>		
6	LS_3	Bool	%M1.2	<input type="checkbox"/>		
7	conveyor_drive_ON_OFF	Bool	%M0.3	<input type="checkbox"/>		
8						

6. 定义“chamber_ON_OFF”变量。

PLC 变量						
	名称	数据类型	地址	保持性	注释	
1	ON_OFF_Switch	Bool	%M0.0	<input type="checkbox"/>		
2	ON	Bool	%M0.1	<input type="checkbox"/>		
3	OFF	Bool	%M0.2	<input type="checkbox"/>		
4	LS_1	Bool	%M1.0	<input type="checkbox"/>		
5	LS_2	Bool	%M1.1	<input type="checkbox"/>		
6	LS_3	Bool	%M1.2	<input type="checkbox"/>		
7	conveyor_drive_ON_OFF	Bool	%M0.3	<input type="checkbox"/>		
8	chamber_ON_OFF	Bool	%M0.4	<input type="checkbox"/>		
9						

7. 定义“milk_pasteurized”变量。

PLC 变量						
	名称	数据类型	地址	保持性	注释	
1	ON_OFF_Switch	Bool	%M0.0	<input type="checkbox"/>		
2	ON	Bool	%M0.1	<input type="checkbox"/>		
3	OFF	Bool	%M0.2	<input type="checkbox"/>		
4	LS_1	Bool	%M1.0	<input type="checkbox"/>		
5	LS_2	Bool	%M1.1	<input type="checkbox"/>		
6	LS_3	Bool	%M1.2	<input type="checkbox"/>		
7	conveyor_drive_ON_OFF	Bool	%M0.3	<input type="checkbox"/>		
8	chamber_ON_OFF	Bool	%M0.4	<input type="checkbox"/>		
9	milk_pasteurized	Bool	%M0.5	<input type="checkbox"/>		
10						

结果

已经定义了扩展程序所需的变量。已定义的变量列在了 PLC 变量表中。

下表概要说明了之前定义的所有变量以及各个变量值的含义：

名称	数据类型	地址	功能	变量值的含义
ON_OFF_Switch	BOOL	M0.0	按钮开关	<ul style="list-style-type: none"> • 1: 机器启动。 • 0: 机器关闭。
ON	BOOL	M0.1	启动机器	<ul style="list-style-type: none"> • 1: 机器启动。 • 0: 无效
OFF	BOOL	M0.2	关闭机器	<ul style="list-style-type: none"> • 1: 机器关闭。 • 0: 无效
LS_1	BOOL	M1.0	用于在传送带的起始位置检测瓶子位置的光栅	<ul style="list-style-type: none"> • 1: 光栅 1 已激活。 • 0: 光栅 1 已取消激活。
LS_2	BOOL	M1.1	用于检测加热室中瓶子位置的光栅	<ul style="list-style-type: none"> • 1: 光栅 2 已激活。 • 0: 光栅 2 已取消激活。
LS_3	BOOL	M1.2	用于在传送带的末端检测瓶子位置的光栅	<ul style="list-style-type: none"> • 1: 光栅 3 已激活。 • 0: 光栅 3 已取消激活。
conveyor_drive_ON_OFF	BOOL	M0.3	传送带的工作模式	<ul style="list-style-type: none"> • 1: 传送带正在运转。 • 0: 传送带已停止运转。
chamber_ON_OFF	BOOL	M0.4	加热室的工作模式	<ul style="list-style-type: none"> • 1: 加热室已启动。 • 0: 加热室已关闭。
milk_pasteurized	BOOL	M0.5	巴氏消毒过程的状态	<ul style="list-style-type: none"> • 1: 牛奶已经过巴氏消毒。 • 0: 牛奶尚未经过巴氏消毒。

3.2.2 对启动传送带的条件进行编程

3.2.2.1 查询机器的状态

简介

以下步骤将介绍如何将机器状态作为启动传送带的条件进行编程。第一个条件是传送带只可能在机器已启动时开始运动。要实现这一条件，请使用常开触点(页 41)，以实现与变量“ON”的互连。

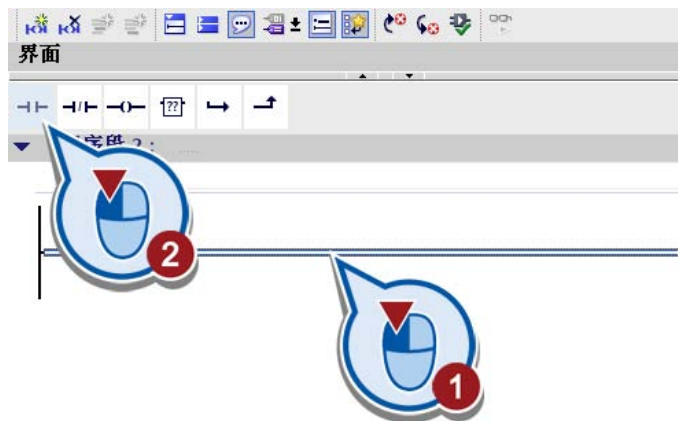
要求

已在组织块“Main [OB1]”的第一个程序段中对项目第一部分中的按钮开关进行了编程。

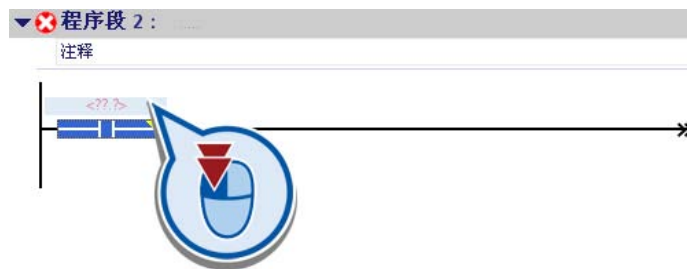
步骤

要将机器状态作为启动传送带的条件来定义，请按以下步骤操作：

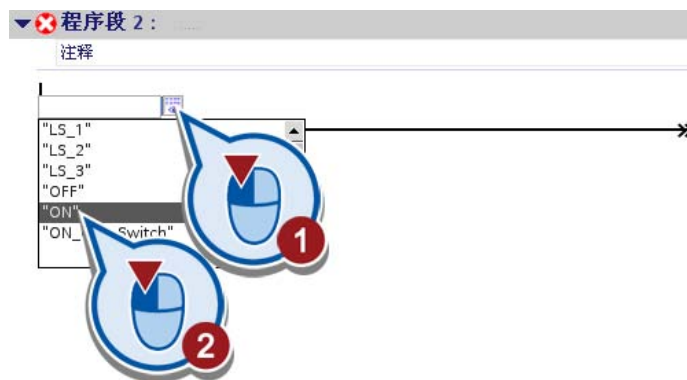
1. 在程序编辑器中打开组织块“Main [OB1]”。
2. 将一个常开触点插入“程序段 2”。



3. 双击常开触点上方的操作数占位符。

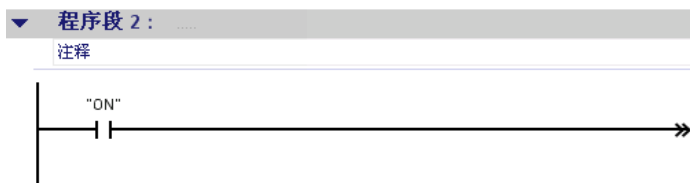


4. 将常开触点与变量“ON”互连。



结果

已经插入一个常开触点并将其与变量“ON”互连。



从而对于启动传送带的第一个条件进行了编程。机器启动时，“ON”变量的信号状态会设置为“1”，常开触点闭合并且电流传递到梯级。在下一部分中，将介绍如何对更多将根据“ON”变量的信号状态执行的条件进行编程。

3.2.2.2 查询瓶子的位置和加热室的状态

简介

以下步骤将介绍如何将瓶子的位置以及加热室的状态作为启动传送带的条件进行编程。
还将介绍如何借助“常闭触点”指令来查询加热室的状态。

常闭触点

下图显示了程序中常闭触点的图标：

<操作数>

---|/|---

常闭触点的激活取决于相关操作数的信号状态。如果操作数的信号状态为“1”，则触点会断开并且流向右侧电源线的电流中断。在这种情况下，指令输出的信号状态为“0”。

当操作数的信号状态为“0”时，常闭触点将闭合。信号流通过该常闭触点流到右侧电源线，并且该指令输出的信号状态设置为“1”。

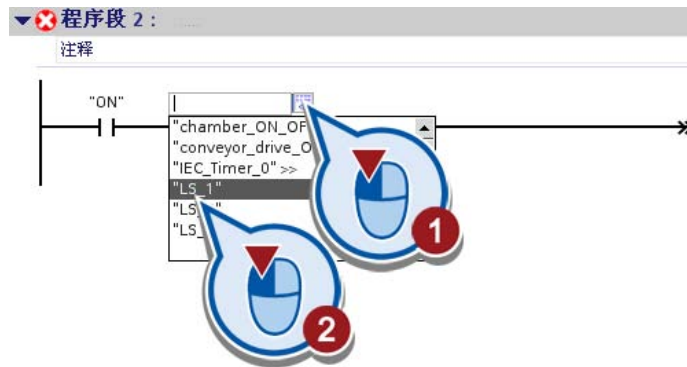
要求

- 组织块“Main [OB1]”已打开。
- 将机器状态作为启动传送带的第一个条件进行编程
(请参见 查询机器的状态 (页 101))

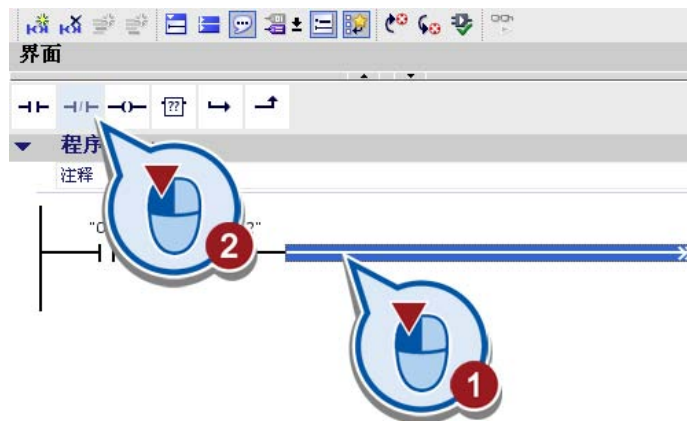
步骤

要将瓶子的位置以及加热室的状态作为启动传送带的条件进行编程，请按以下步骤操作：

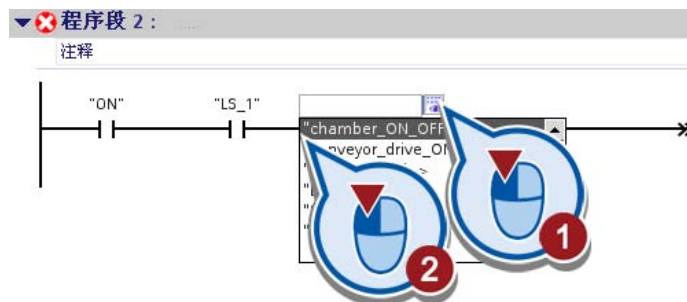
1. 将另一个常开触点插入“程序段 2”。
2. 将常开触点与变量“LS_1”互连。



3. 将一个常闭触点插入“程序段 2”。

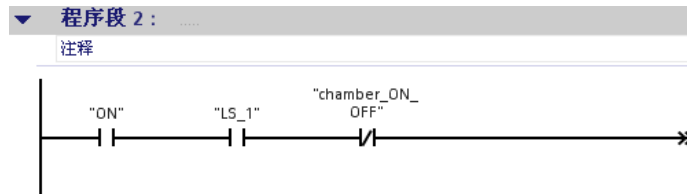


4. 将常闭触点与“chamber_ON_OFF”变量互连。



结果

已经将瓶子的位置以及加热室的状态作为启动传送带的条件进行编程。



3.2.2.3 查询巴氏消毒进度

简介

以下步骤将介绍如何根据巴氏消毒过程的进度来控制传送带。对瓶子中的牛奶进行巴氏消毒后，传送带开始运动并将瓶子传送到传送带的末端。与巴氏消毒过程进度有关的信息存储在“milk_pasteurized”变量中，并可借助常开触点进行查询。

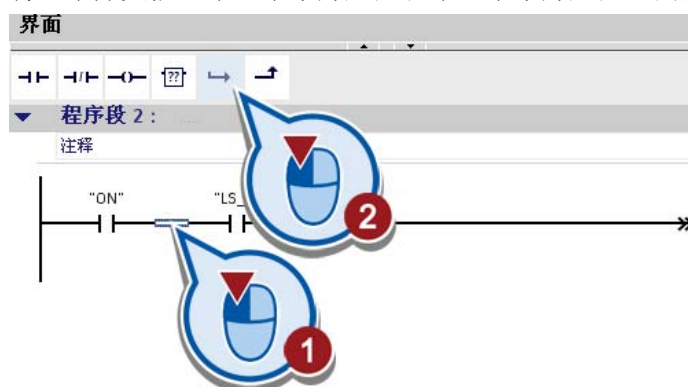
要求

- 组织块“Main [OB1]”已打开。
- 已对“查询机器的状态 (页 101)”和“查询瓶子的位置和加热室的状态 (页 103)”部分中所描述的条件进行了编程。

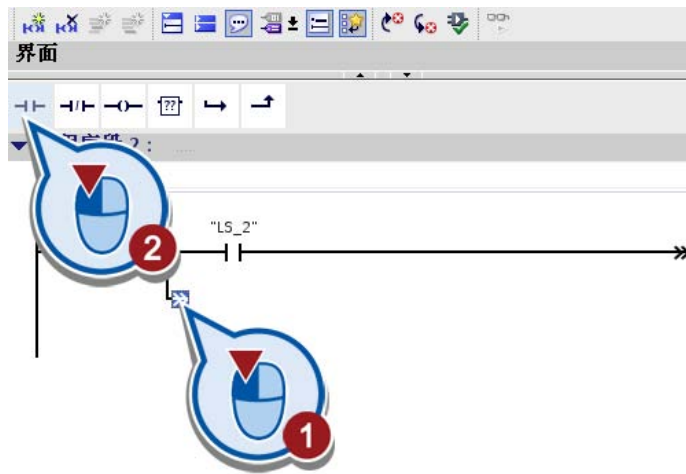
步骤

要根据巴氏消毒过程的进度来控制传送带，请按以下步骤操作：

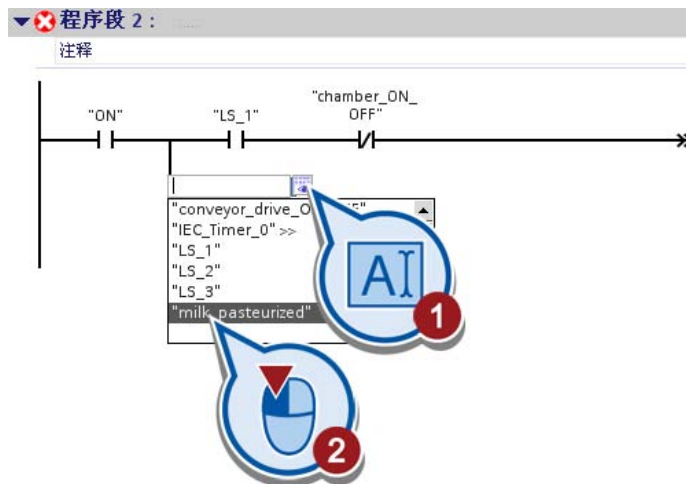
1. 将一个分支插入第一个常开触点和第二个常开触点之间的梯级中。



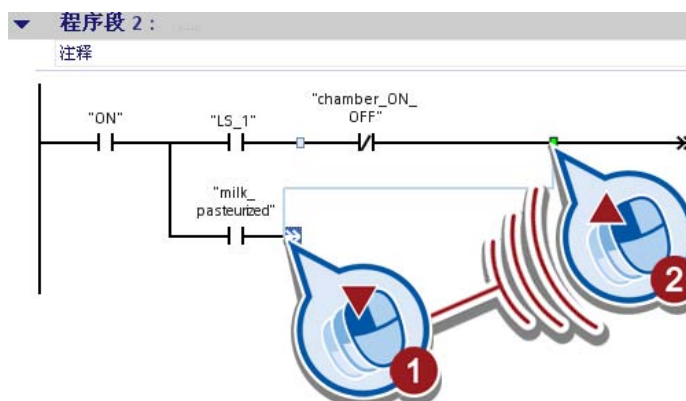
2. 在打开的分支中插入一个常开触点。



3. 将常开触点与“milk_pasteurized”变量互连。

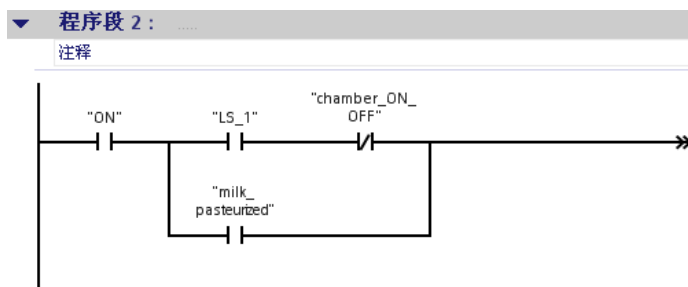


4. 闭合打开的分支。



结果

已经将巴氏消毒进度作为启动传送带的另一个条件进行了编程。



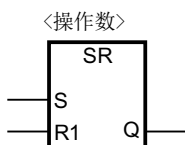
3.2.2.4 控制传送带

简介

以下步骤将介绍如何根据已编程的条件启动传送带。要控制传送带，可使用“置位/复位触发器”指令。

置位/复位触发器

下图显示了“置位/复位触发器”指令的功能框。



可以使用该指令根据指令的 S 和 R1 输入信号状态置位或复位指定操作数的位。

- 如果 S 输入的信号状态为“1”且 R1 输入的信号状态为“0”，则指定的操作数将置位为“1”。
- 如果 S 输入的信号状态为“0”且 R1 输入的信号状态为“1”，则指定的操作数将复位为“0”，因为输入 R1 的优先级高于输入 S。
- 输入 S 和 R1 的信号状态都为“1”时，指定操作数的信号状态将复位为“0”。

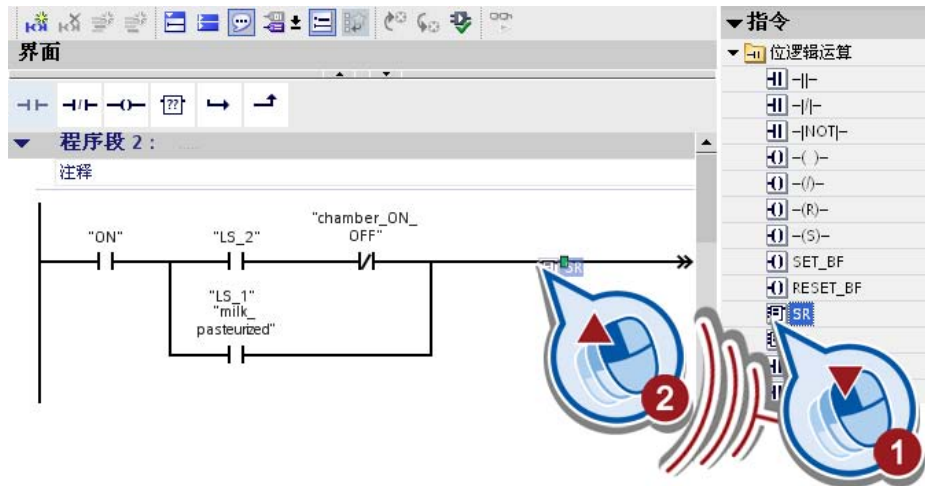
要求

- 组织块“Main [OB1]”已打开。
- 已对“查询机器的状态 (页 101)”、“查询瓶子的位置和加热室的状态 (页 103)”和“查询巴氏消毒进度 (页 106)”部分中所描述的条件进行了编程。

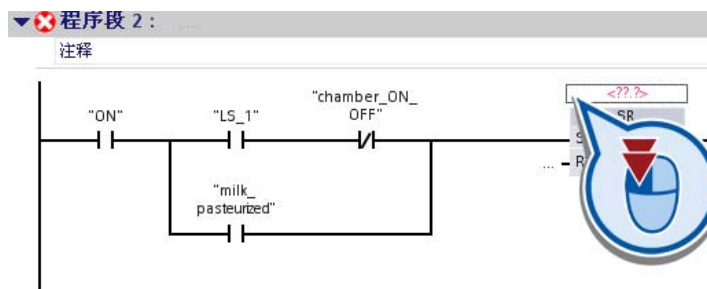
步骤

要对传送带的控制进行编程，请按以下步骤操作：

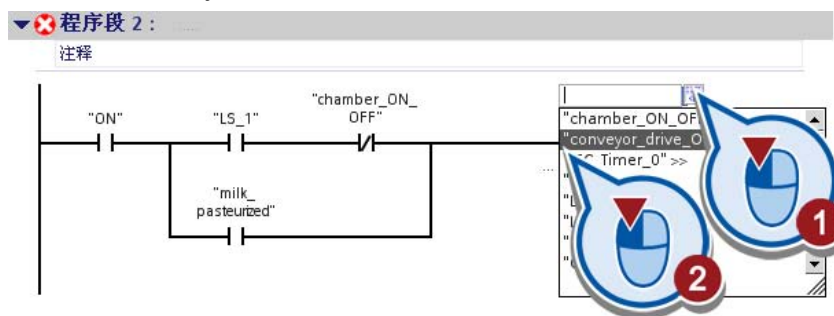
1. 在第二个程序段的末端插入“置位/复位触发器”指令 (SR)。



2. 双击指令上方的操作数占位符。



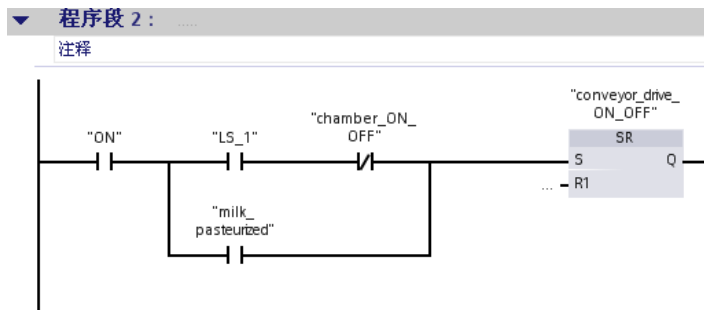
3. 将指令与“conveyor_drive_ON_OFF”变量互连。



4. 单击工具栏上的“保存”(Save) 按钮以保存该项目。

结果

已经对用于启动传送带的条件进行了编程。



当“ON”变量的信号状态为“1”且至少满足以下条件之一时，传送带开始运动：

- 光栅（LS1）在传送带的起始位置检测到瓶子且加热室处于关闭状态。此时，“LS_1”变量的信号状态为“1”，而“chamber_ON_OFF”变量的信号状态为“0”。
- 奶瓶已经过巴氏消毒。此时，变量“milk_pasteurized”的信号状态为“1”。瓶子被传送到传送带的末端。

传送带开始运动时，“conveyor_drive_ON_OFF”变量的信号状态为“1”。

在下一部分，将介绍如何对停止传送带的条件进行编程。

3.2.3 对停止传送带的条件进行编程

简介

以下步骤将介绍如何对停止传送带的条件进行编程。可在“置位/复位触发器”指令的输入 R1 中定义这些内容，“置位/复位触发器”指令已插入到组织块“Main [OB1]”的最后一部分中。

可借助以下指令来对停止传送带的条件进行编程：

- 常开触点
- 常闭触点
- 置位/复位触发器

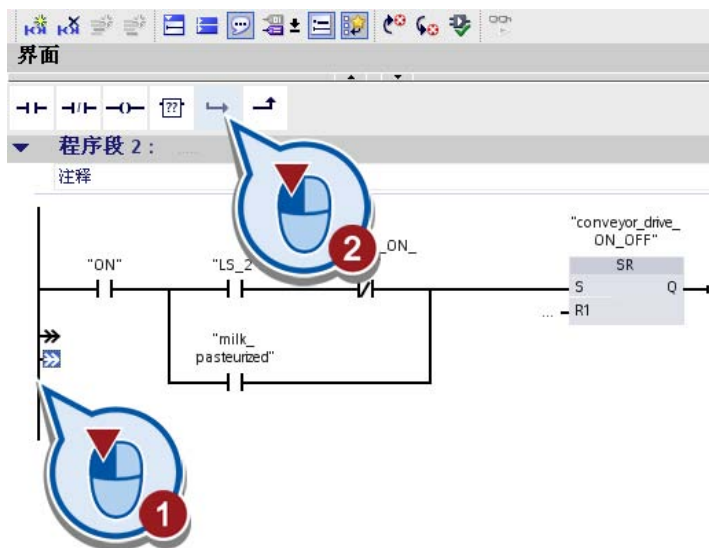
要求

- 已在组织块“Main [OB1]”的第一个程序段中对项目第一部分中的按钮开关进行了编程。
- 已在组织块“Main [OB1]”中对启动传送带的条件进行了编程。
另请参见“对启动传送带的条件进行编程 (页 101)”

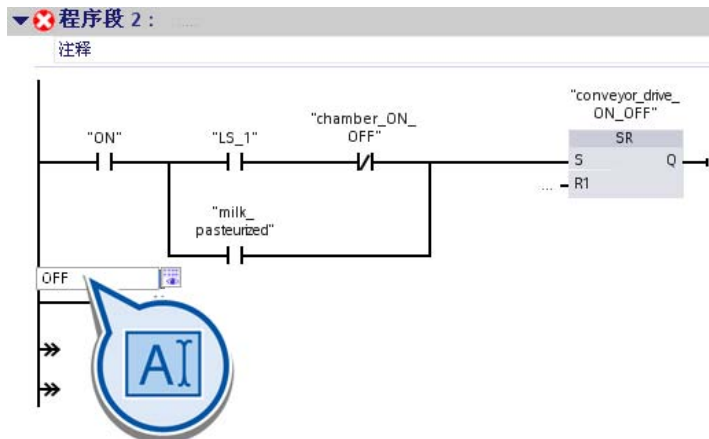
步骤

要对停止传送带的条件进行编程，请按以下步骤操作：

1. 打开组织块“Main [OB1]”中的“程序段 2”。
2. 添加三个新的梯级。



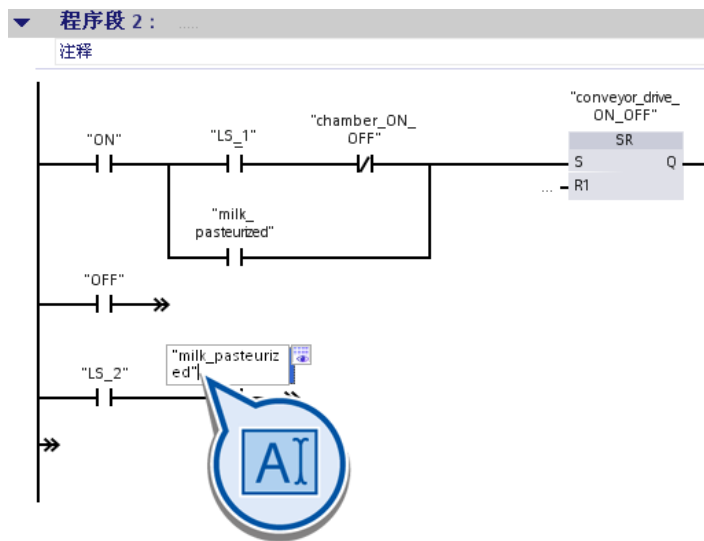
3. 将一个常开触点插入到第一个新添加的梯级中并将其与变量“OFF”互连。



通过互连变量“OFF”可定义如下条件，即机器关闭时停止传送带。

4. 将一个常开触点插入到第二个新添加的梯级中并将其与变量“LS_2”互连。

5. 在第二个梯级的末尾插入一个常闭触点并将其与变量“milk_pasteurized”互连。

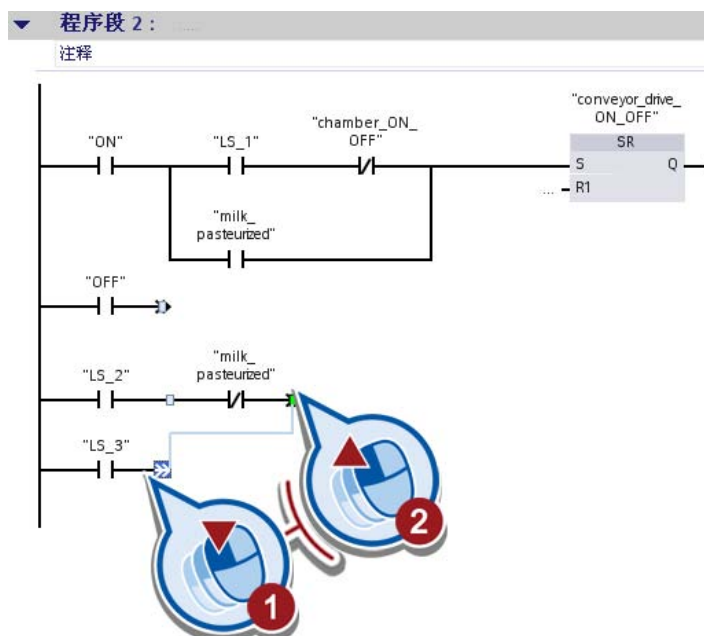


通过互连变量“LS_2”和“milk_pasteurized”可定义如下条件，即瓶子已到达加热室但尚未对牛奶进行巴氏消毒时停止传送带。

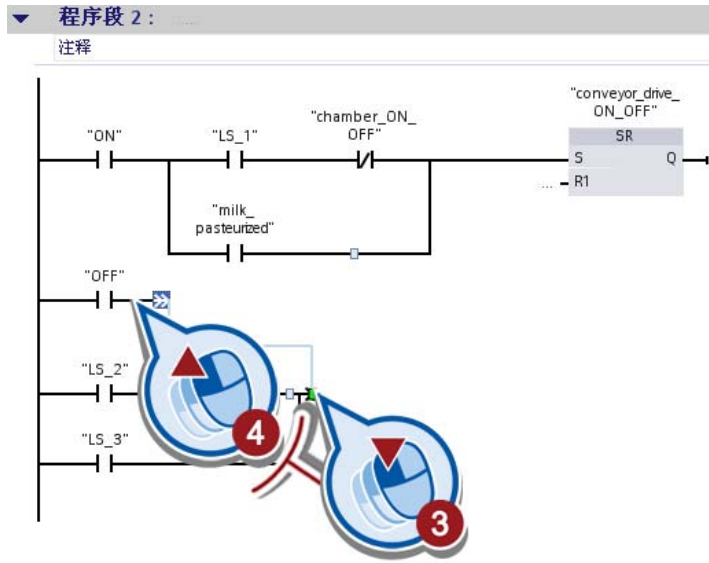
6. 将一个常开触点插入到第三个新添加的梯级中并将其与变量“LS_3”互连。

通过互连变量“LS_3”来定义如下条件，即瓶子已到达传送带的末端时停止传送带。

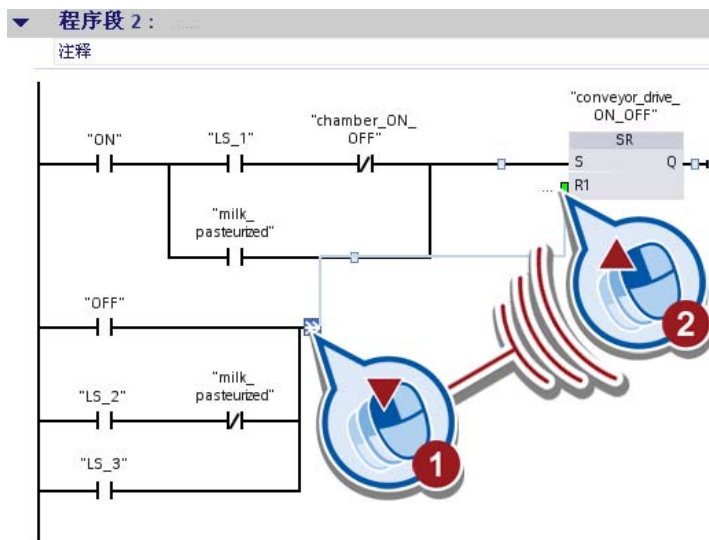
7. 以并联的方式互连插入的常开触点。



3.2 扩展程序

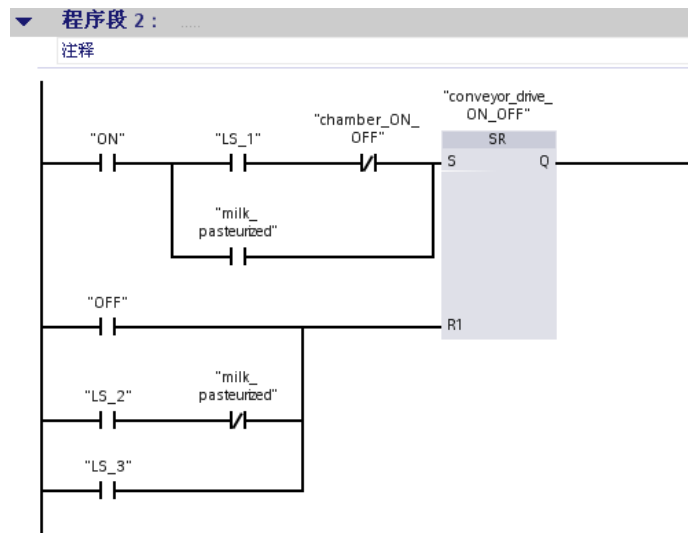


8. 将并联结构与指令“置位/复位触发器”的输入 R1 互连。



结果

已对传送带驱动器的 PLC 进行了编程。将根据奶瓶在传送带上的位置驱动或停止传送带。奶瓶的位置是借助光栅“LS_1”、“LS_2”和“LS_3”进行检测的。



当“OFF”变量的信号状态为“1”或至少满足下列条件之一时，将停止传送带：

- 光栅（LS2）在加热室中检测到瓶子且尚未对牛奶进行巴氏消毒。此时，“LS_2”变量的信号状态为“1”，而“milk_pasteurized”变量的信号状态为“0”。
- 光栅（LS3）在传送带的末端检测到瓶子。此时，变量“LS_3”的信号状态为“1”。

传送带停止运转时，变量“conveyor_drive_ON_OFF”的信号状态为“0”。

在下一部分中，将介绍如何对加热室的启动和关闭进行编程。

3.2.4 对加热过程的 PLC 进行编程

简介

以下步骤将介绍如何对加热过程的 PLC 进行编程。加热过程通过加热室的启动和关闭来控制。在实例项目中，为了简化起见，将假设加热室启动时里面的温度已经达到了 75°C。

可借助以下指令在组织块“Main [OB1]”的第三个程序段中对加热室的 PLC 进行编程：

- 常开触点
- 置位/复位触发器

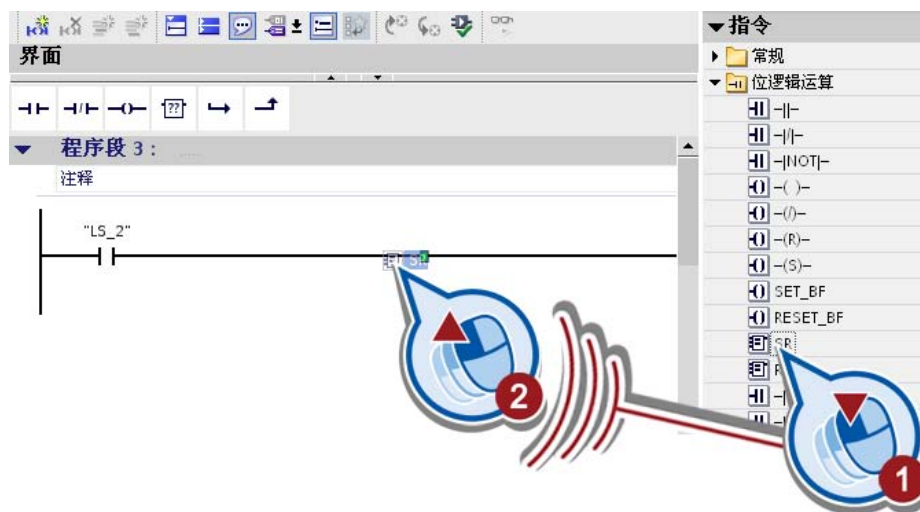
要求

- 组织块“Main [OB1]”已打开。
- 已对组织块“Main [OB1]”的程序段 1 和 2 进行了编程。

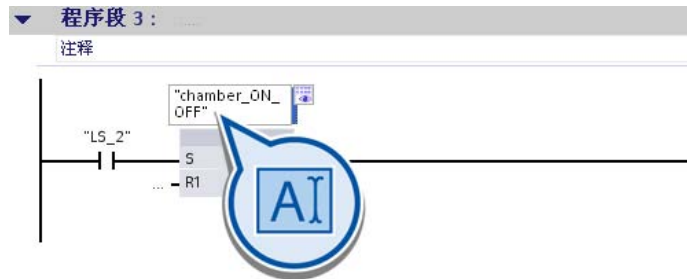
步骤

要对加热过程的 PLC 进行编程，请按以下步骤操作：

1. 打开组织块“Main [OB1]”的第三个程序段。
2. 插入一个常开触点。
3. 将常开触点与变量“LS_2”互连。
4. 插入指令“置位/复位触发器”。

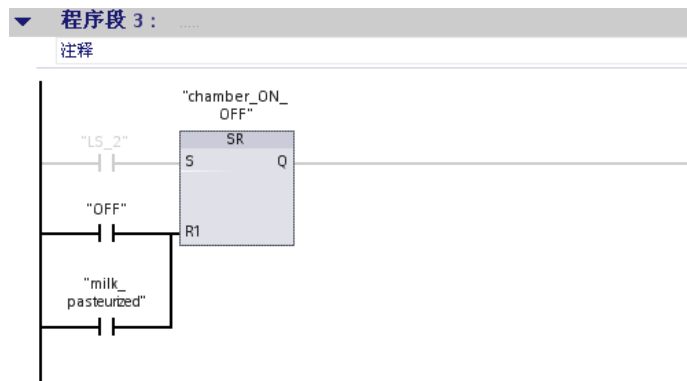


5. 在操作数占位符中，单击指令“置位/复位触发器”的上方并将该指令与变量“chamber_ON_OFF”互连。



当光栅（LS2）检测到瓶子时，变量“LS_2”在指令“置位/复位触发器”的输入处具有信号状态“1”。从而置位变量 chamber_ON_OFF 并启动加热室。

6. 在指令的输入 R1 处对关闭加热室的以下条件进行编程：

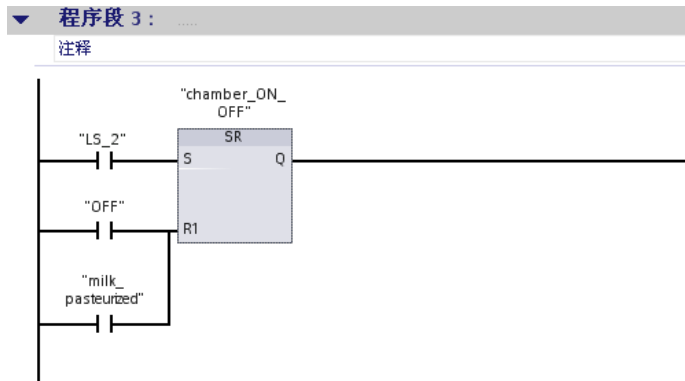


如果满足下列条件之一，将复位变量“chamber_ON_OFF”并关闭加热室：

- 机器已关闭。此时，变量“OFF”的信号状态为“1”。
- 牛奶已经过巴氏消毒。此时，变量“milk_pasteurized”的信号状态为“1”。

结果

已对加热过程的 PLC 进行了编程。



当光栅“LS2”检测到奶瓶时，会启动加热室。超过指定的加热时段且已对牛奶进行了巴氏消毒时，将关闭加热室并把瓶子传送到传送带的末端。

在下一部分中，将介绍如何对加热时段的设置进行编程。

3.2.5 对加热时段进行编程

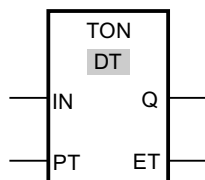
简介

以下步骤将介绍如何对加热过程的持续时间进行编程。要对牛奶进行巴氏消毒，奶瓶需要在加热室中加热 40 秒。可借助以下指令在组织块“Main [OB1]”的第四个程序段中对加热时段的设置进行编程：

- 常开触点
- 接通延迟
- 输出线圈

接通延迟

下图显示了“接通延迟”指令的功能框。



可以使用该指令将带有时间延迟的指定操作数的信号状态置位为“1”。当该指令的输入 IN 的信号状态从“0”变为“1”（上升沿）时，将执行该指令。当启动该指令时，将开始延迟时段 (PT)。超过延迟时间时，输出 Q 的信号状态为“1”。

输出线圈

下图显示了程序中“输出线圈”指令的图标：

<操作数>

---()---

可以使用“输出线圈”指令置位指定操作数的位。线圈输入端的逻辑运算结果 (RLO) 为“1”时，指定操作数的信号状态将被置位为“1”。线圈输入端的信号状态为“0”时，指定操作数的位被复位为“0”。

要求

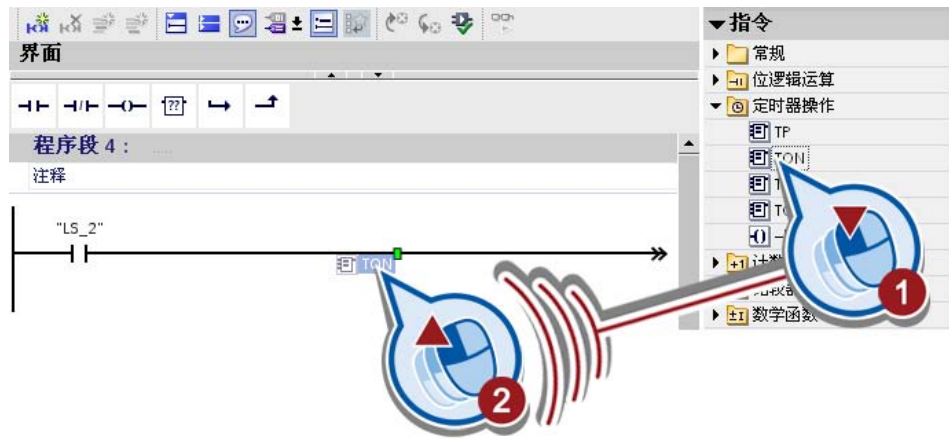
- 组织块“Main [OB1]”已打开。
- 已对组织块“Main [OB1]”的程序段 1 到 3 进行编程。

步骤

要对加热过程的持续时间进行编程，请按以下步骤操作：

1. 打开组织块“Main [OB1]”的第四个程序段。
2. 插入一个常开触点。
3. 将常开触点与变量“LS_2”互连。

4. 插入指令“接通延迟”(TON)。



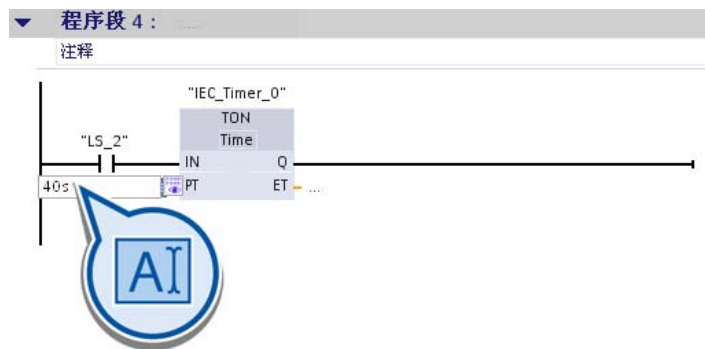
此时将打开用于创建数据块的“调用选项”(Call options) 对话框。

5. 创建数据块“IEC_Timer_0”。



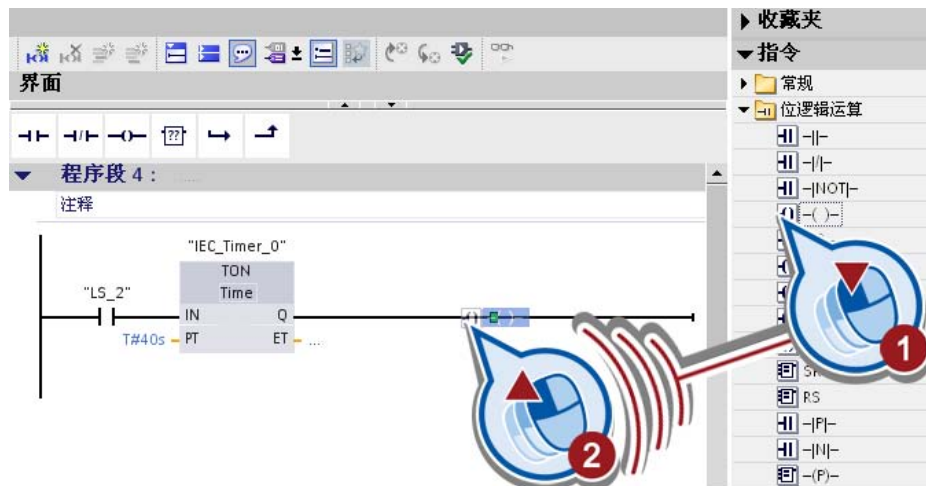
数据块“IEC_Timer_0”创建在“程序块”文件夹中并被分配给指令“接通延迟”。插入的时间指令的数据将被存储在该数据块中。

6. 在时间指令的输入 PT 处输入加热时段 40 秒。

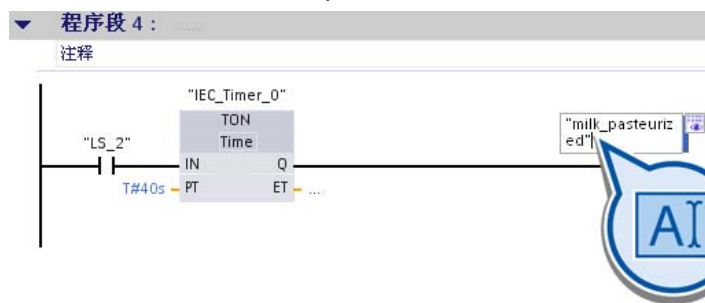


当加热室中的光栅检测到瓶子时，变量“LS_2”的信号状态会从“0”变为“1”。此时在时间指令的输入 PT 中所指定的时段将开始。

7. 在梯级末尾插入“输出线圈”指令。



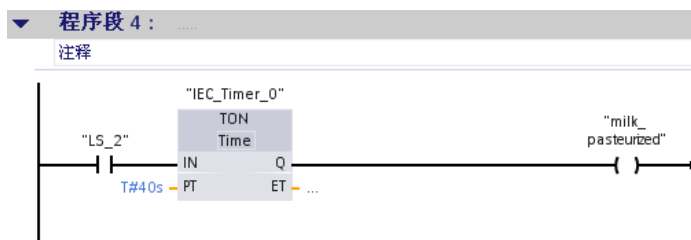
8. 将指令“输出线圈”与“milk_pasteurized”变量互连。



在超过输入 PT 中所指定的时段后，变量“milk_pasteurized”的信号状态将置位为“1”。

结果

已对巴氏消毒过程的持续时间进行了编程。



当奶瓶激活光栅“LS_2”时，将启动加热室并开始巴氏消毒时段。在巴氏消毒时段结束后，变量“milk_pasteurized”的信号状态会置位为“1”。传送带开始运转并将奶瓶传送到传送带的末端。

在下一部分中，将介绍如何对指示加热室工作模式的状态指示灯进行编程。

3.2.6 对状态指示灯进行编程

简介

以下步骤将介绍如何对状态指示灯进行编程。状态指示灯指示加热室的工作模式。当加热室处于工作状态时，变量 LED 的信号状态设置为“1”并开启 HMI 设备上的显示。当加热室处于关闭状态时，变量 LED 的信号状态为“0”，HMI 设备上的显示也会关闭。在下一部分中，将对 HMI 设备上的显示进行编程。

借助以下指令在组织块“Main [OB1]”的第五个程序段中对状态指示灯进行编程：

- 常开触点
- 输出线圈

要求

- 组织块“Main [OB1]”已打开。
- 已对组织块“Main [OB1]”的程序段 1 到 4 进行编程。

步骤

要对状态指示灯进行编程以使其指示加热室的工作模式，请按以下步骤操作：

1. 将一个常开触点插入“Main [OB1]”组织块的第五个程序段。
2. 将常开触点与变量“chamber_ON_OFF”互连。



3. 在梯级末尾插入“输出线圈”指令。
4. 在“输出线圈”指令的操作数占位符中输入名称 LED 并按回车键确认输入。
5. 打开设备和网络编辑器。

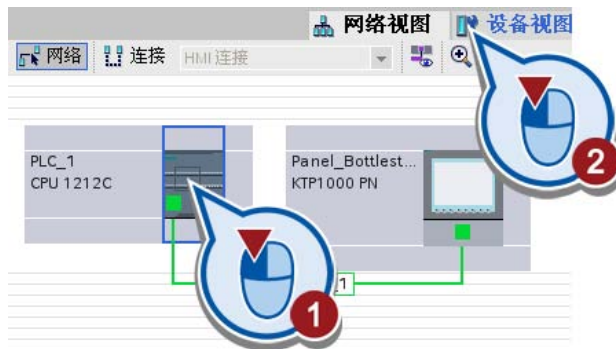


说明

程序段不完整

程序段 5 中显示了一个包含白色叉字的红色圆圈图标。该图标说明此程序段中的互连或变量定义尚未完成。

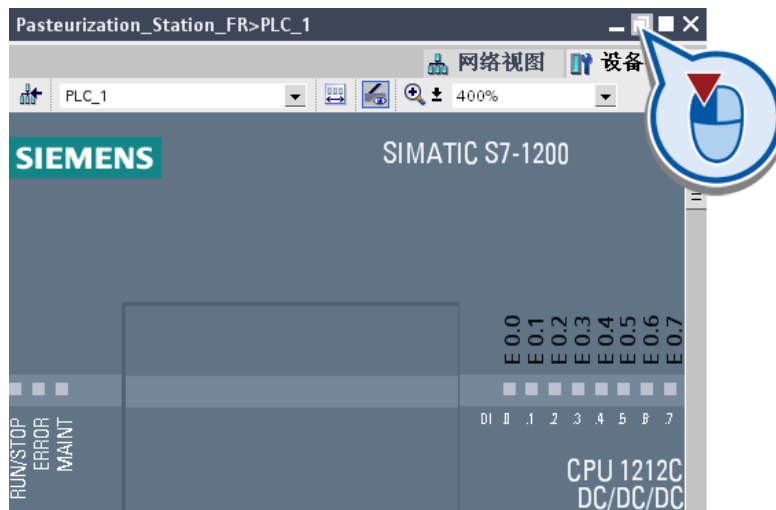
6. 切换到设备视图。



7. 放大设备视图，直到可以清晰地看到 PLC 的输入和输出为止。

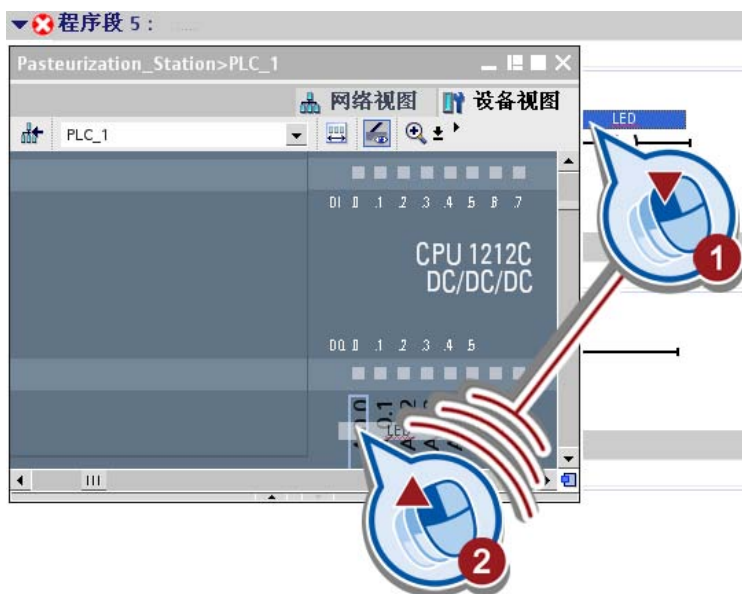


8. 分离设备窗口和程序段编辑器的窗口并将其放置在程序编辑器旁。



9. 切换到组织块“Main [OB1]”的第五个程序段。

10. 将“LED”变量与 PLC 的输出“Q 0.0”互连。



“LED”变量即被定义并显示在 PLC 的输出“Q 0.0”。先前在 PLC 变量表中定义的变量将显示在 PLC 的输入处。

说明

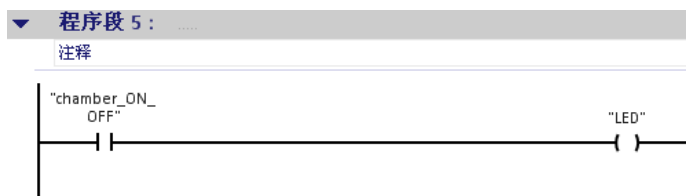
定义变量

无论在设备视图还是在 PLC 变量表或程序段中定义输入和输出的变量，都没有任何区别。已定义的变量还会显示在设备视图中。

11. 单击工具栏上的“保存”(Save) 按钮以保存该项目。

结果

已对指示加热室工作模式的状态指示灯进行了编程。



加热室启动时，“chamber_ON_OFF”变量的信号状态为“1”。通过“输出线圈”指令将“LED”变量的信号状态设置为“1”。只要启动加热室，“LED”变量的信号状态就为“1”。关闭加热室时，变量“chamber_ON_OFF”和“LED”的值会变为“0”

在下面的部分中，将介绍如何通过程序状态测试已创建的程序。

3.3 通过程序状态测试扩展的程序

简介

以下步骤介绍了如何使用程序状态测试已创建的程序。

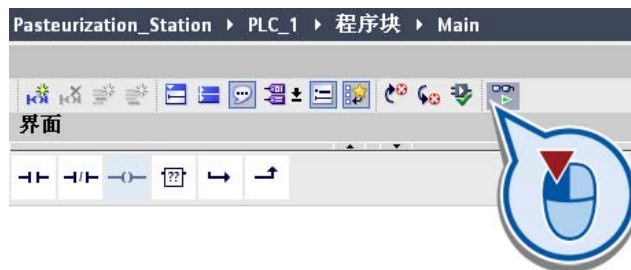
要求

- 已组态 PLC。
- PLC 的输入和输出无电压，因为修改的值会被在线模式下的模块覆盖。
- 组织块“Main [OB1]”已在程序编辑器中打开。

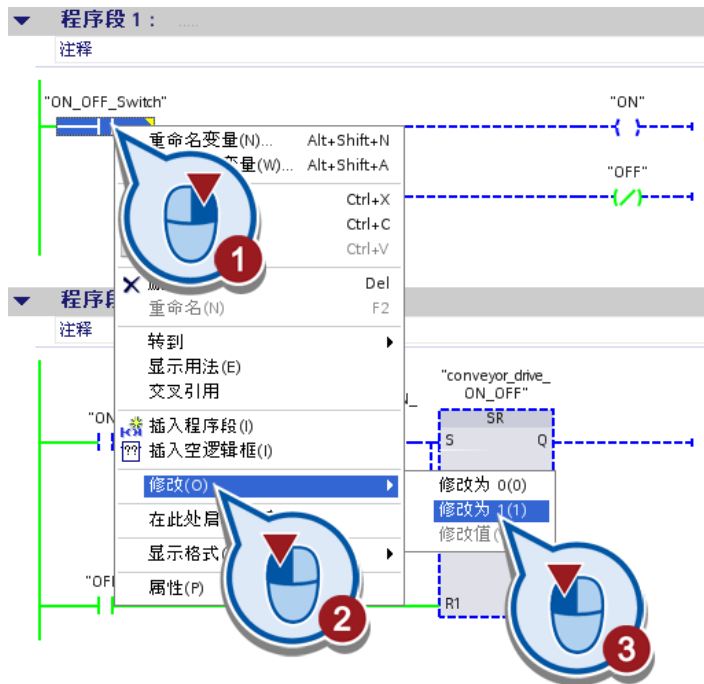
步骤

要使用程序状态测试已创建的程序，请按以下步骤操作：

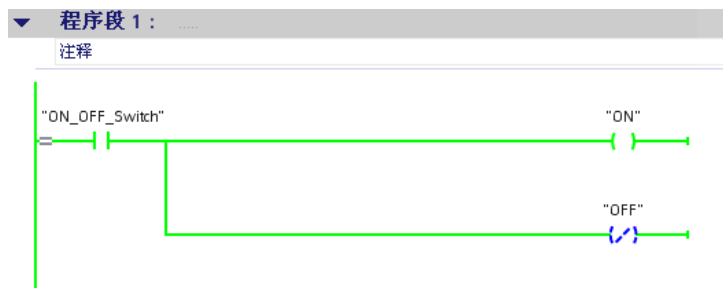
1. 将程序加载到 PLC 中并激活在线连接。更多信息，请参见“将程序加载到目标系统 (页 52)”部分。
2. 单击程序编辑器中工具栏上的“启用/禁用监视”(Monitoring on/off) 按钮。



3. 在程序段 1 中，将变量“ON_OFF_Switch”修改为“1”。



将变量“ON_OFF_Switch”的信号状态设置为“1”。电流通过常开触点流到程序段末尾的线圈。置位变量“ON”，从而启动实例机器。OFF 变量保持复位为“0”，没有任何其它作用。

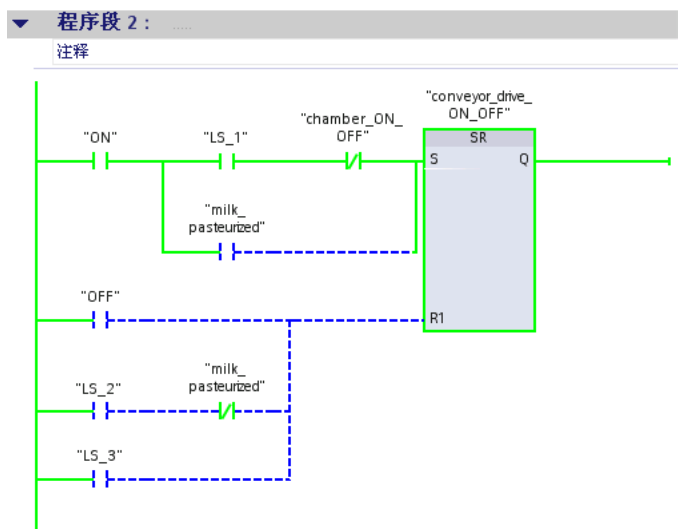


3.3 通过程序状态测试扩展的程序

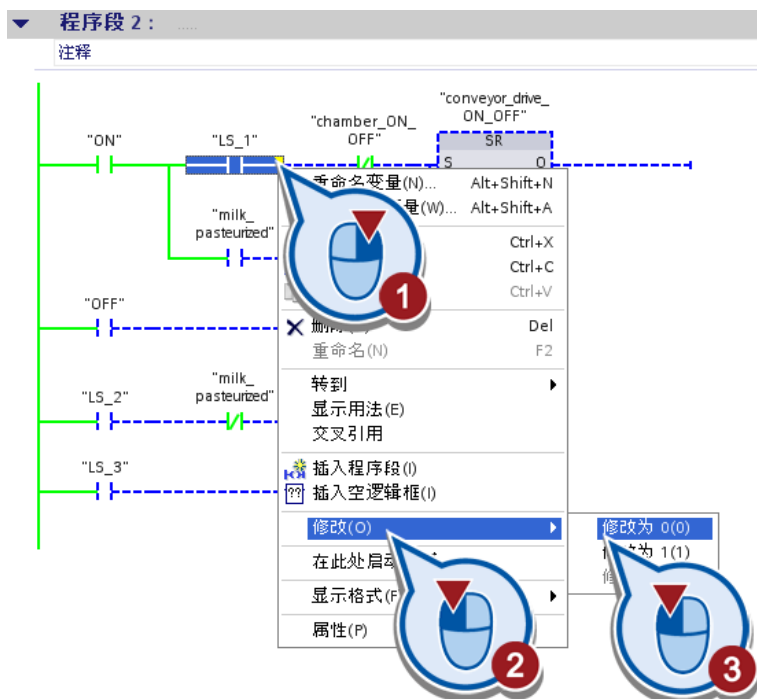
4. 在程序段 2 中，将变量“LS_1”修改为“1”。

将变量“LS_1”的信号状态设置为“1”。这仿真了在传送带的起始位置通过光栅“LS1”检测到瓶子这一过程。由于加热室在仿真过程中处于关闭状态，因此“chamber_ON_OFF”变量的信号状态为“0”。

信号流过主梯级的各个触点，使得“置位/复位触发器”指令的输入 S 的信号状态为“1”。从而将变量“conveyor_drive_ON_OFF”的信号状态设置为“1”，并且驱动传送带。



5. 在程序段 2 中，将变量“LS_1”修改为“0”。



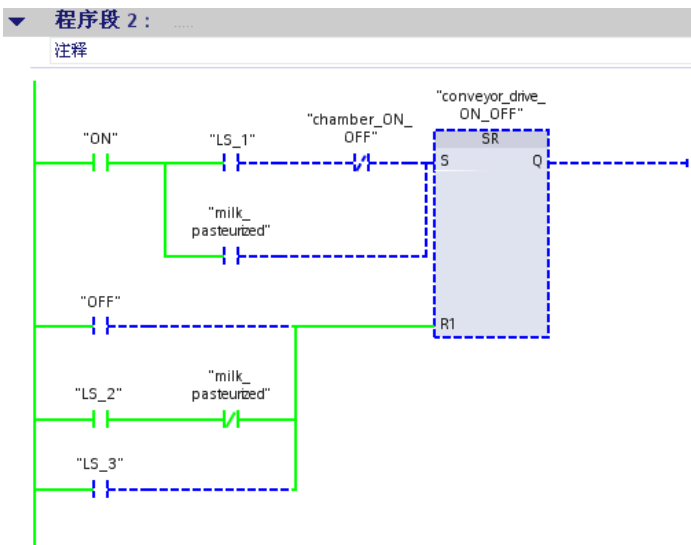
变量 LS_1 的信号状态复位为“0”。这将仿真取消激活光栅“LS1”这一过程。

6. 在程序段 2 中，将变量“LS_2”修改为“1”。

在程序段 2 中：

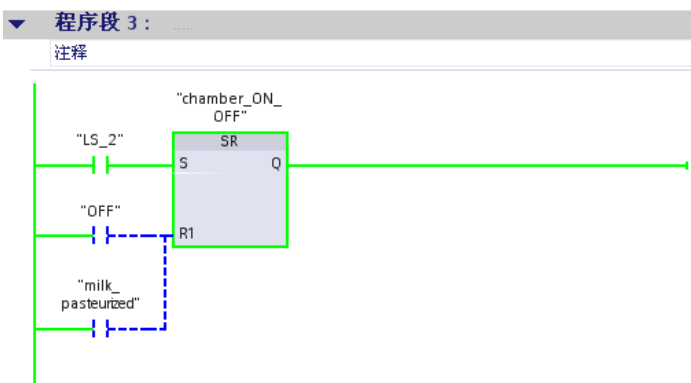
将变量“LS_2”的信号状态设置为“1”。此设置将仿真瓶子被传送到加热室并且光栅“LS2”被激活这一过程。

电流将改变路线流向“置位/复位触发器”指令的输入 R1。从而复位变量 conveyor_drive_ON_OFF 并停止传送带。



在程序段 3 中：

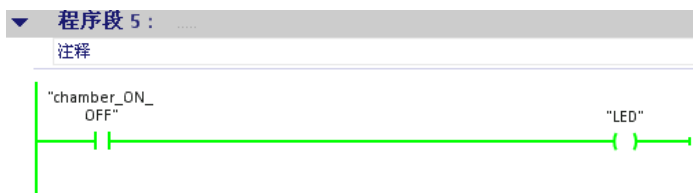
电流通过主梯级的触点流向“置位/复位触发器”指令的输入 S。从而将变量 chamber_ON_OFF 的信号状态设置为“1”并启动加热室。



3.3 通过程序状态测试扩展的程序

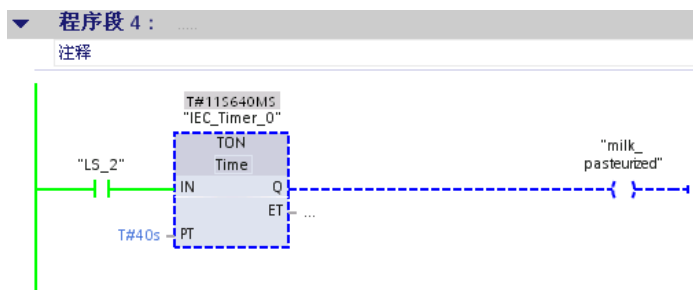
在程序段 5 中：

只要启动加热室（程序段 3），变量 LED 的信号状态就会被设置为“1”且状态指示灯将被激活。

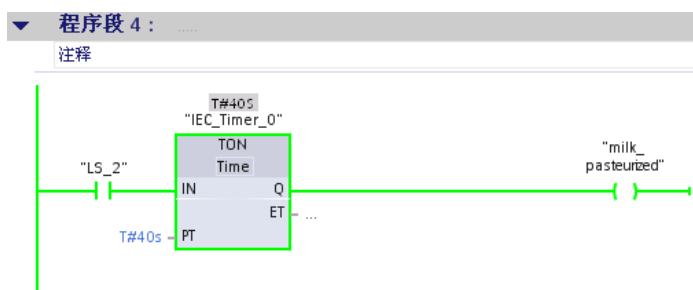


在程序段 4 中：

在“接通延迟”时间运算的 IN 输入检测到一个上升沿并且巴氏消毒时段已开始。

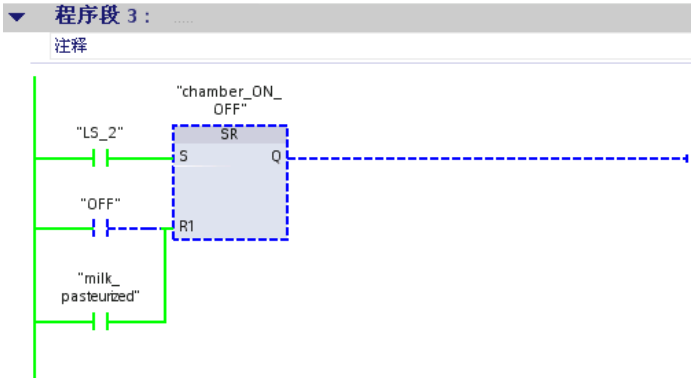


在该时段结束时，牛奶已经过巴氏消毒，并且变量“milk_pasteurized”的信号状态将被设置为“1”。



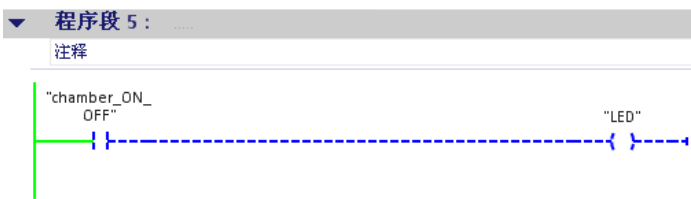
在程序段 3 中：

置位“milk_pasteurized”变量时关闭加热室。



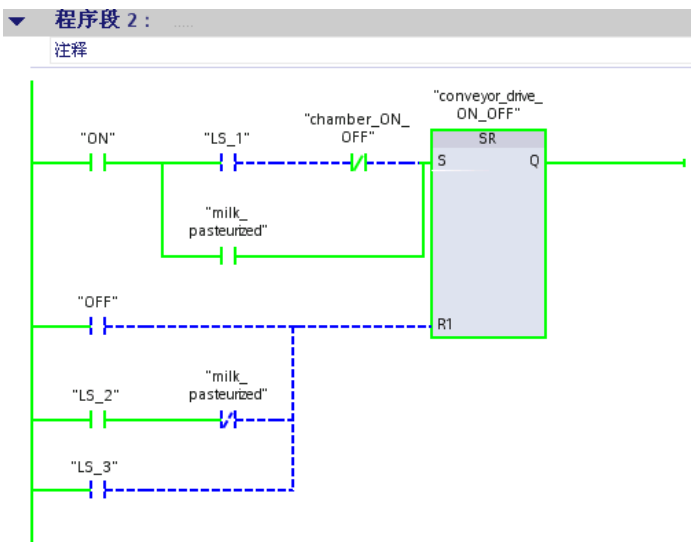
在程序段 5 中：

关闭加热室时，状态指示灯将取消激活。



在程序段 2 中：

传送带再次开始运动。



7. 在程序段 2 中，将变量“LS_2”修改为“0”。

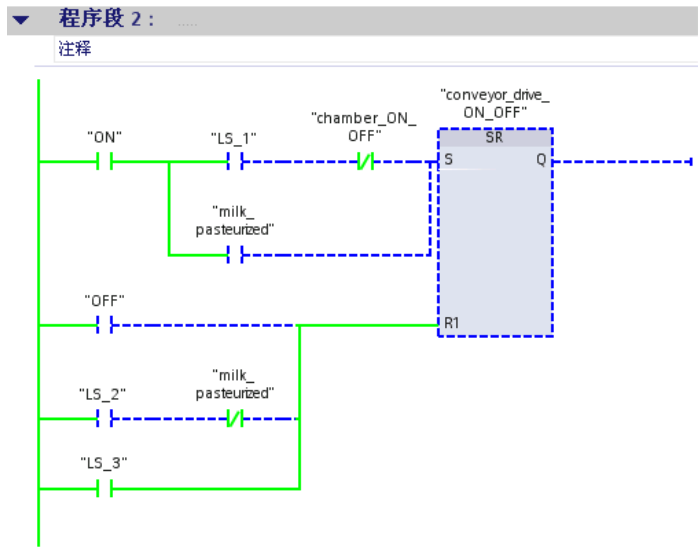
变量 LS_2 的信号状态复位为“0”。此状态将仿真瓶子已离开加热室这一过程。

3.3 通过程序状态测试扩展的程序

8. 在程序段 2 中，将变量“LS_3”修改为“1”。

将变量“LS_3”的信号状态设置为“1”。此状态将仿真瓶子已传送到传送带的末端并且被光栅“LS3”检测到这一过程。

变量 conveyor_drive_ON_OFF 的信号状态复位为“0”，传送带将停止。



9. 终止在线连接。

结果

已测试程序，并检查了程序的执行情况。

在下一部分中，将介绍如何将更多图形对象添加到 HMI 画面。这样能可视化扩展程序的流程。

3.4 扩展 HMI 画面

3.4.1 图形对象“加热室”

简介

以下步骤介绍如何将图形形式的加热室添加到 HMI 画面。使用简单的静态图形对象显示加热室。

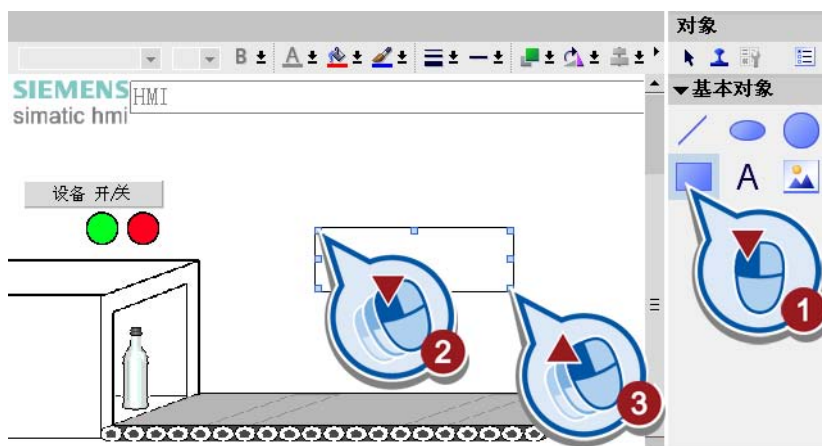
要求

- “简单实例”一章中的 HMI 画面已创建，或者“简单实例 (页 18)”项目已加载。
- 在“Panel_Bottlestation”下创建的 HMI 画面处于打开状态。

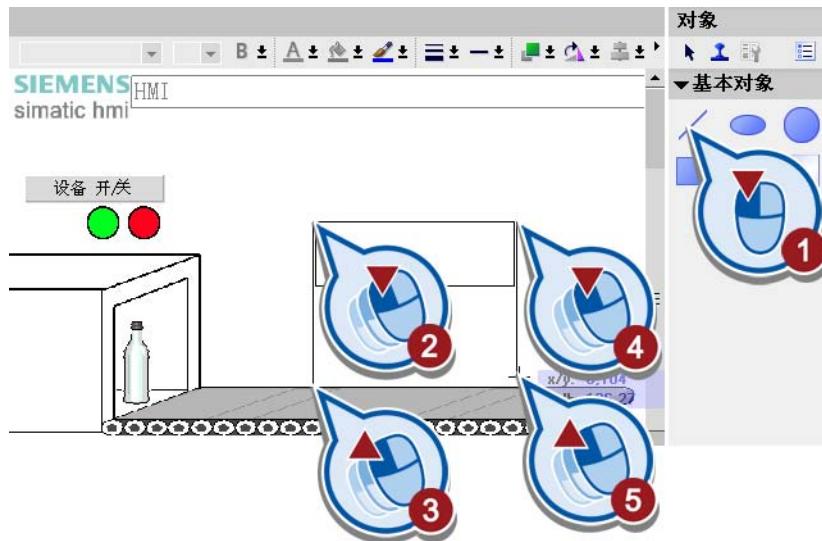
步骤

要将图形形式的加热室添加到 HMI 画面，请按以下步骤操作：

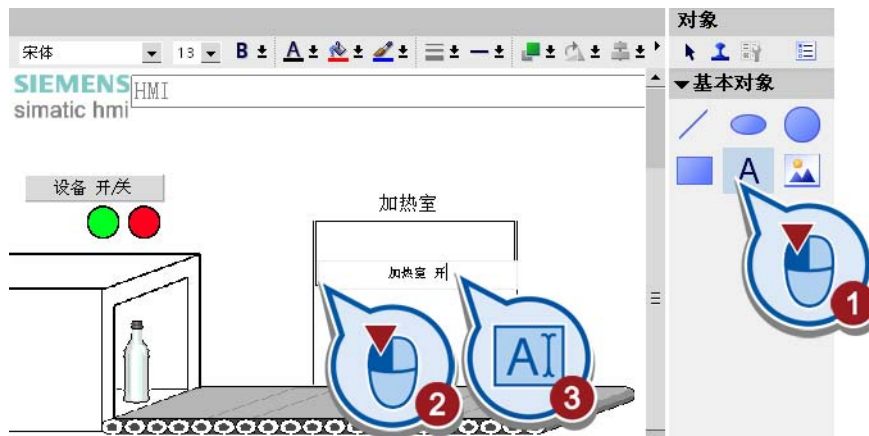
1. 在传送带的上方，画面的中间绘制一个矩形。



2. 在矩形左右两侧各绘制一条垂直线。



3. 创建两个文本块，“加热室”和“加热室 开”。



4. 在巡视窗口中的“常规”(General) 和“外观”(Appearance) 下，调整文本格式。

结果

将加热室的静态画面元素添加到 HMI 画面。

3.4.2 图形对象“加热室 LED”

简介

以下步骤介绍如何将图形形式的 LED 添加到 HMI 画面并通过 PLC 变量“LED”使其动态化。

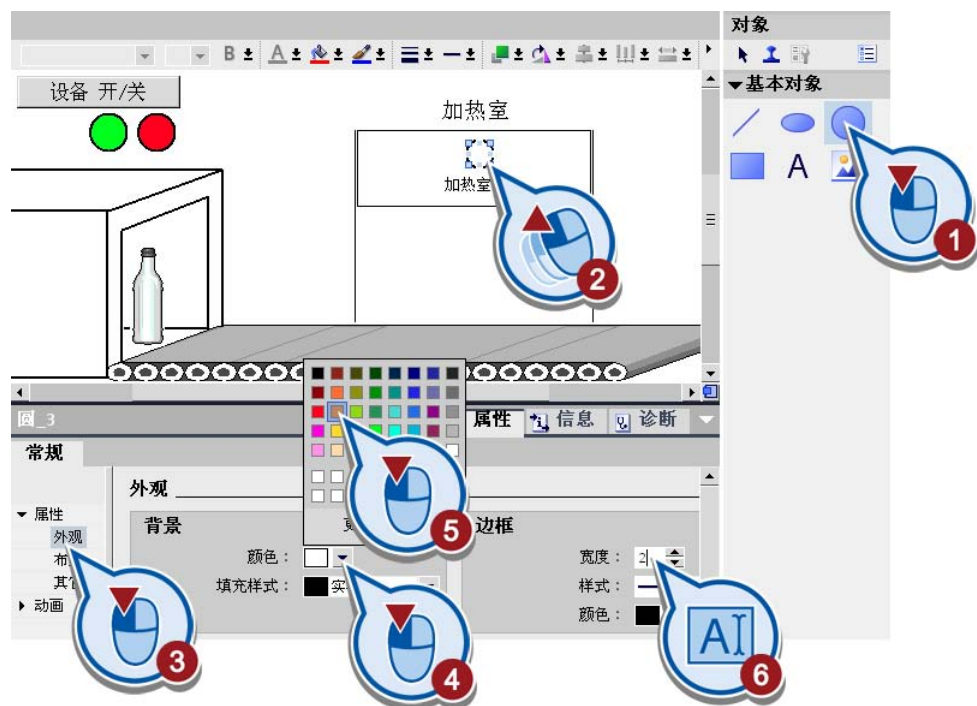
要求

- 已创建程序。
- HMI 画面处于打开状态。

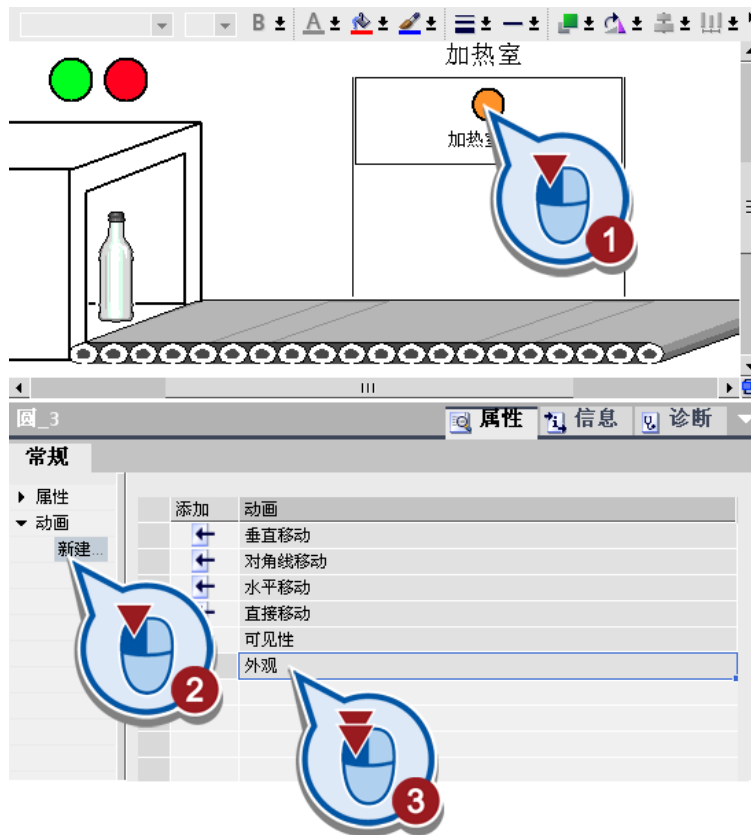
步骤

要为加热室创建 LED 并使其动态化，请按以下步骤操作：

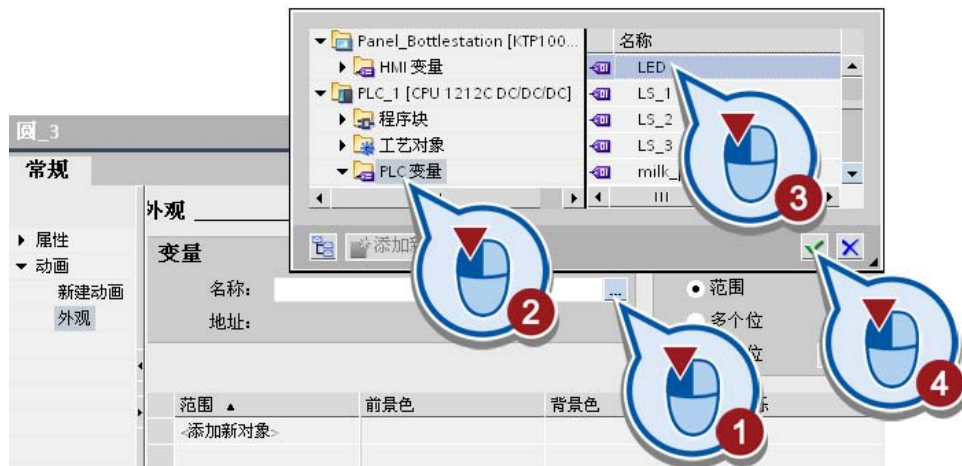
1. 在矩形中文本块“加热室 开”上方的居中位置，插入一个橙色的圆，代表加热室 LED。



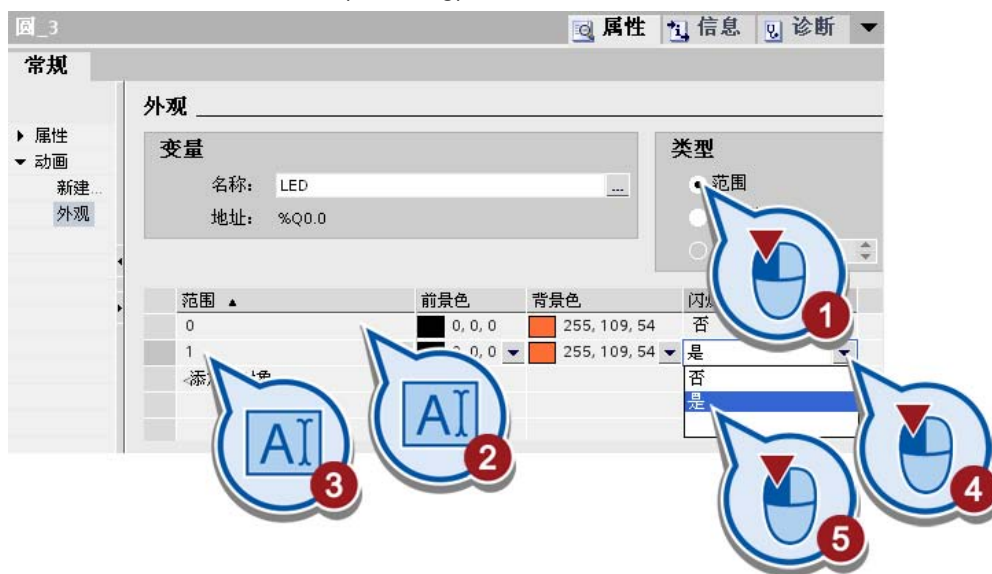
2. 为加热室 LED 创建新动画。



3. 将动画链接到 LED 变量。



4. 将变量的值“1”设置为“闪烁”(Flashing)。



结果

将一个 LED 添加到 HMI 画面。如果“LED”变量的值为“1”，则在 HMI 画面中通过所创建 LED 的闪烁来表示激活加热室。

在下一部分中，将介绍如何将图形形式的光栅添加到 HMI 画面。

3.4.3 图形对象“光栅”

简介

以下步骤介绍如何将光栅“LS1”、“LS2”和“LS3”添加到 HMI 画面。使用基本对象“圆”表示光栅。将已创建的光栅链接到程序的变量并使图形表示动态化。

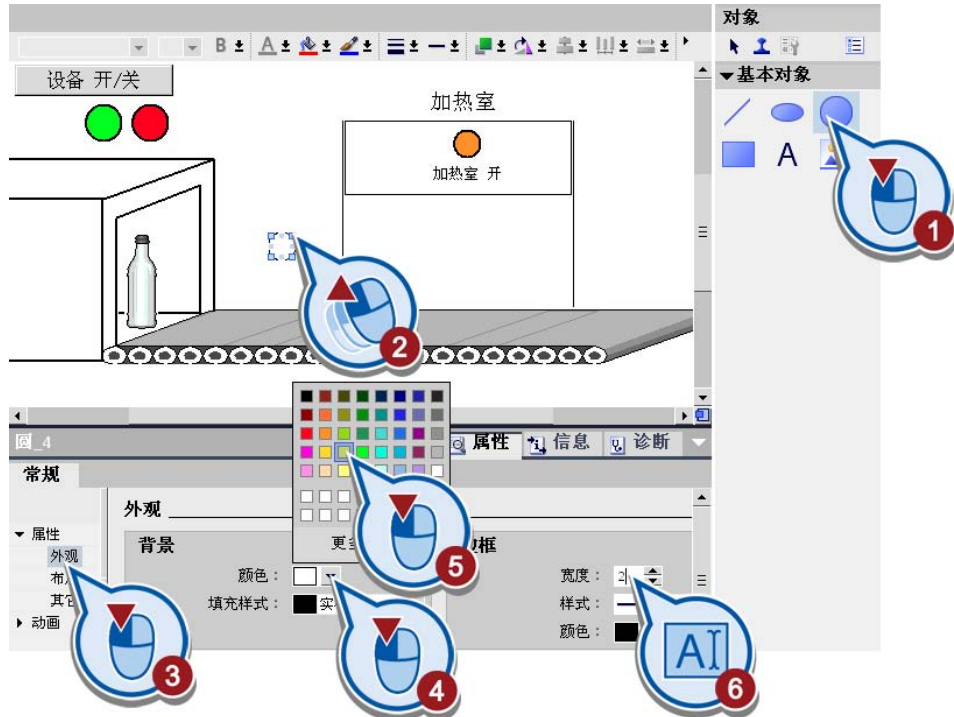
要求

- 已创建控制程序。
- HMI 画面处于打开状态。

步骤

要在 HMI 画面中插入图形形式的光栅，请按以下步骤操作：

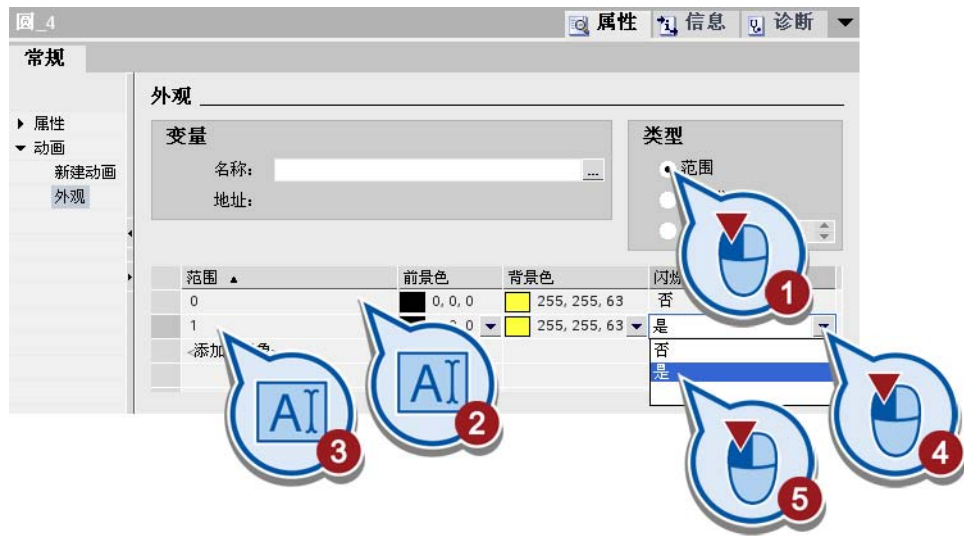
1. 插入一个对象“圆”表示光栅，并将背景色设置为黄色，边框的宽度为“2”。



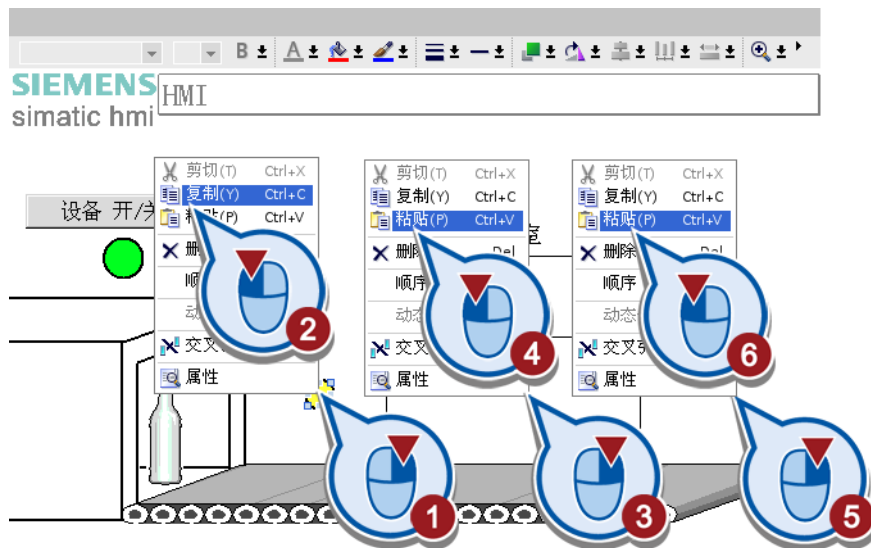
2. 为该光栅创建“外观”(Appearance) 动画。



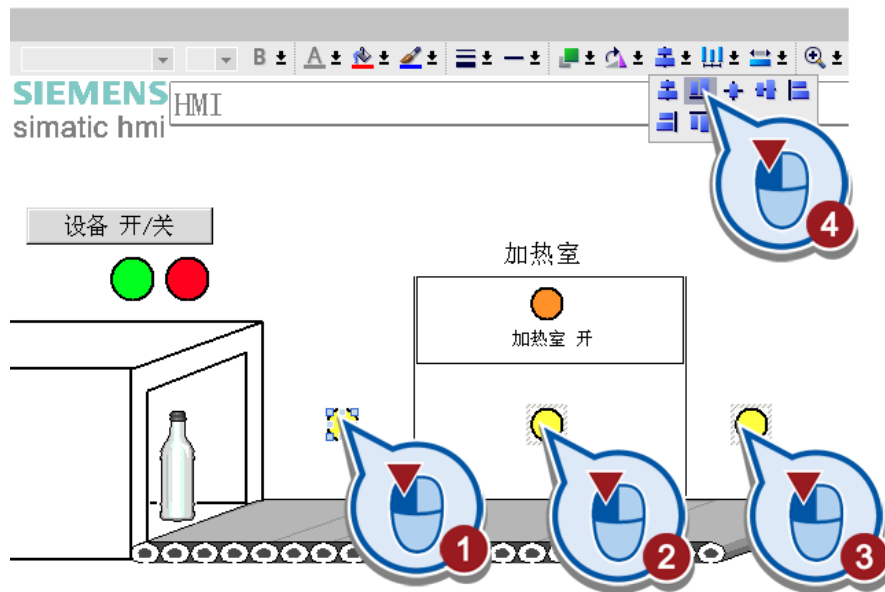
3. 为变量范围“1”激活“闪烁”(Flashing) 功能，无需预先定义变量。



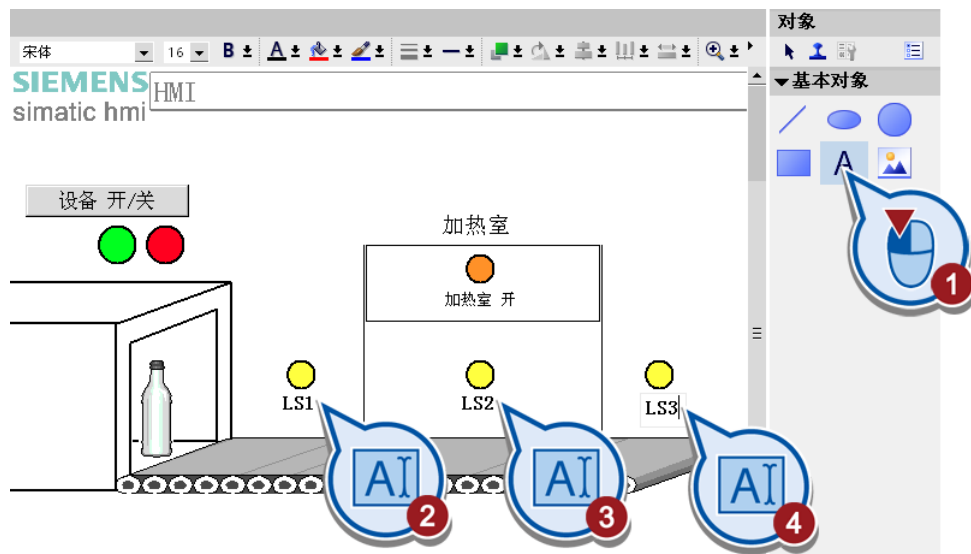
4. 使用复制和粘贴功能再创建两个光栅。



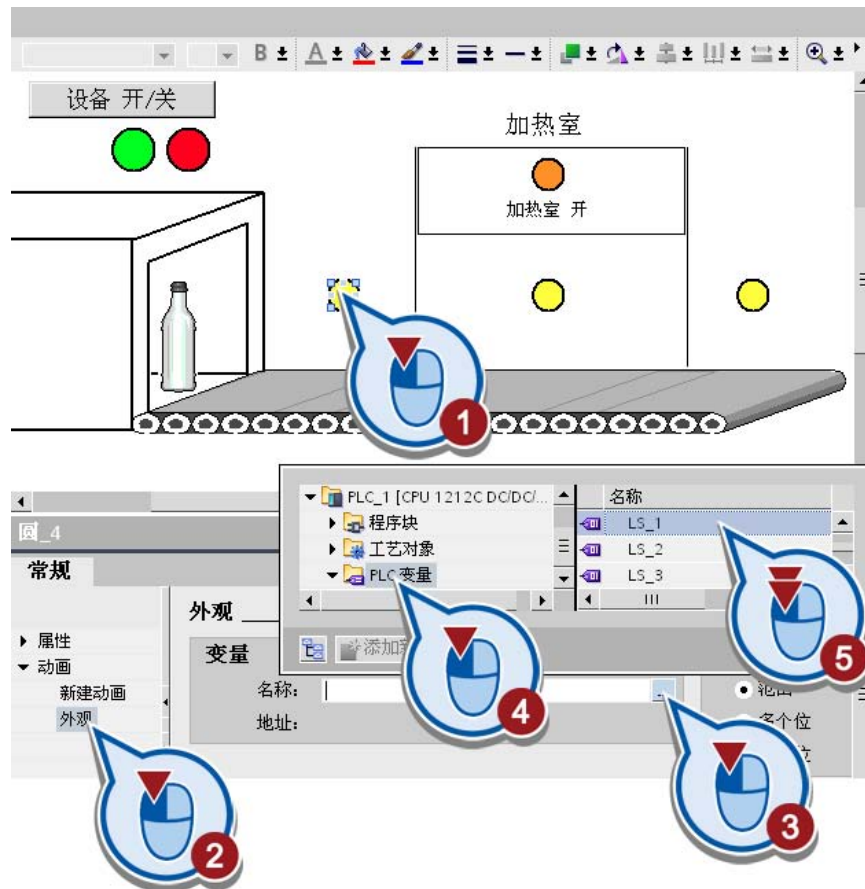
5. 按住 Shift 键，选择全部三个光栅并将它们对齐。



6. 将这些光栅分别标记为“LS1”、“LS2”和“LS3”。



7. 将第一个光栅的外观动画链接到 PLC 变量“LS_1”。



8. 将第二和第三个光栅的外观动画分别链接到 PLC 变量“LS_2”和“LS_3”。

9. 单击工具栏上的“保存”(Save) 按钮以保存该项目。

结果

将光栅“LS1”、“LS2”和“LS3”添加到 HMI 画面。且已将这些光栅与变量链接并使其动态化。

在下一部分中，将介绍如何通过仿真画面来测试动态对象的行为。

3.5 仿真 HMI 画面

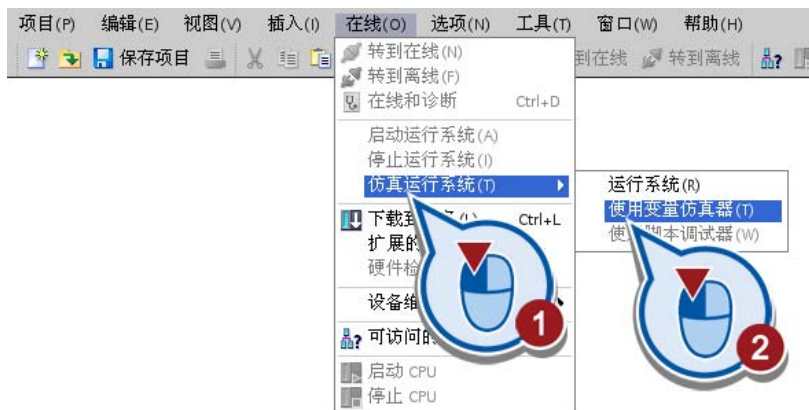
简介

以下步骤介绍如何使用运行系统仿真器测试所创建的 HMI 画面。使用运行系统仿真器可仿真光栅“LS1”对应的 PLC 输入的激活。有关运行系统仿真器的更多信息，请参见“仿真运行系统 (页 92)”部分。

步骤

要启动已创建的 HMI 画面的仿真，请按以下步骤进行操作：

1. 通过菜单栏启动运行系统仿真。HMI 设备窗口必须已激活。如果菜单未激活，则先单击 HMI 画面中的空闲区域。



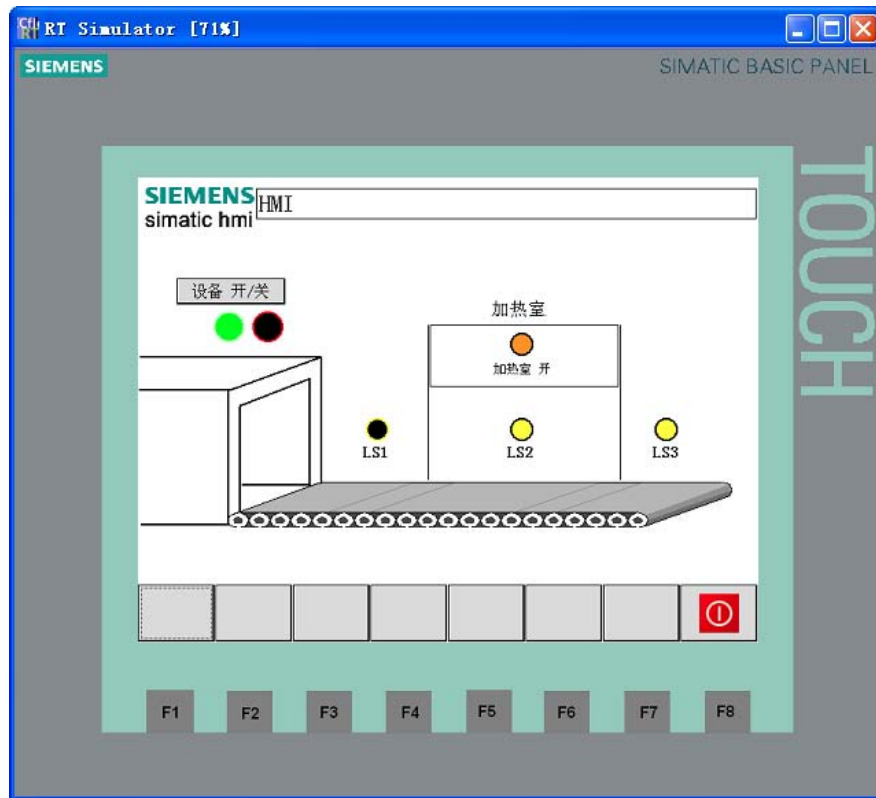
2. 将“LS_1”变量的值设置为“1”。按回车确认设置的值。



3. 切换到“运行系统仿真器”(RT Simulator) 窗口。

结果

仿真了 PLC 变量“LS_1”的值。光栅“LS1”闪烁。



通过在“设置数值”(Set value) 列中指定值可以仿真所有使用的 PLC 变量的值。所有 PLC 变量均为布尔值，这意味着它们的值只能为“0”或“1”。

说明

运行系统仿真器中的变量

HMI 变量显示在运行系统仿真器的变量选项中。在“变量”(Tag) 表列的选项中不会列出与任何事件或 HMI 画面的动画都未链接的 PLC 变量。

在 HMI 设备中测试

可以选择在 HMI 设备上测试 HMI 画面。有关详细信息，请参见“将 HMI 画面加载到 HMI 设备 (页 89)”部分。

实例“PID 控制”

4.1 简介

加载项目

如果跳过了之前的章节，则可以加载上一章结尾处的项目状态（请参见“加载项目 (页 18)”）。上一章结尾的项目状态存储在“Extended_Example.ZIP”文件中。

简介

如果某个物理值（例如，温度、压力或速度）在过程中必须具有特定值，并且该值会根据无法预见的外部条件而变化，则必须使用控制器。

定义 PID 控制器

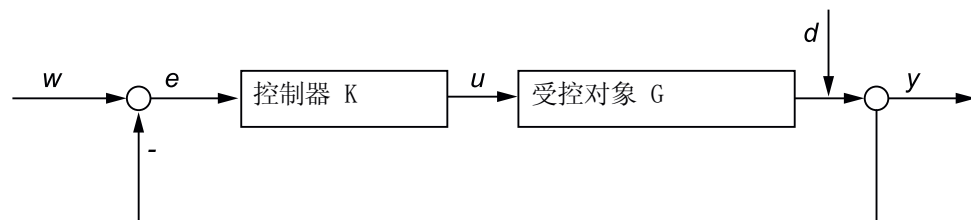
PID 控制器是由比例、积分和微分单元组成。它在控制回路中连续检测受控变量的实际测量值，并将其与期望设定值进行比较。PID 使用所生成的控制偏差来计算控制器的输出，以便尽可能快速平稳地将受控变量调整到设定值。

定义控制回路

控制回路是由受控对象、控制器、测量元件（传感器）和控制元件组成。

- 在本例中，使用具有 PID 控制的 SIMATIC S7-1200 作为控制器。
- 本例中的测量元件是传感器，用于测量加热室内的温度。
- 控制元件是由 PLC 直接控制的加热器。

下面的接线图包含了一个典型的控制回路：



设定值“ w ”已预先定义。在下面的实例中，设定值是加热室中的期望温度 75°C 。可通过设定值 (w) 和实际值 (y) 来计算控制偏差 (e)。控制器 (K) 可将控制偏差转换为受控变量

4.1 简介

(u)。受控变量通过受控对象 (G) 来更改实际值 (y)。本例中的受控对象 (G) 为加热室中的温度调节，可以通过增加或减少能量输入进行控制。

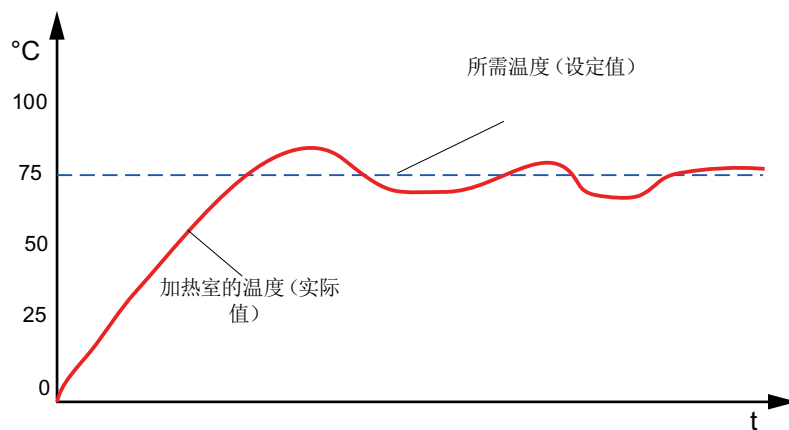
除受控对象 (G) 外，也可以通过干扰变量 (d) 改变实际值 (y)。本例中的干扰变量可能是加热室中意外的温度变化。例如，由室外温度变化引起的温度变化。

使用 PID 控制器

在实例项目中，使用 PID 控制器尽可能快地达到所需的 75°C 温度并尽可能保持设定值不变。

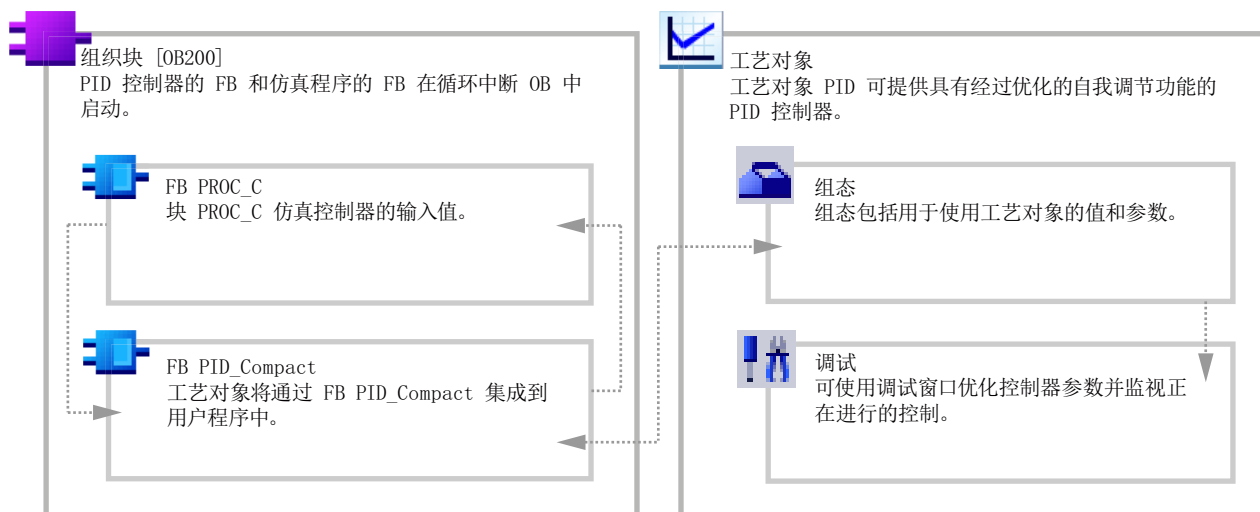
在本例中，由于加热元件在关闭后继续发热，因此将超出设定值。该效应称为“过调”；如果实际值的控制和测量之间存在延时，则会发生过调。

下图显示了首次打开设备后可能的温度特征曲线：



步骤

下图显示了要创建的所有对象的总览：



请按以下步骤操作以创建这些对象：

- 创建第二个组织块 [OB200]，在其中将调用 PID 控制器的块。
- 创建工艺对象“PID_Compact”。
- 将仿真块“PROC_C”加载到组织块 [OB200]。如果使用仿真块，无需使用 PLC 之外的其它硬件。
- 组态工艺对象“PID_Compact”。
 - 选择控制器的类型。
 - 输入控制器的设定值。
 - 将工艺对象“PID_Compact”的实际值和受控变量与仿真块“PROC_C”互连。
- 在工艺窗口的调试窗口中加载用户程序并执行控制器优化。

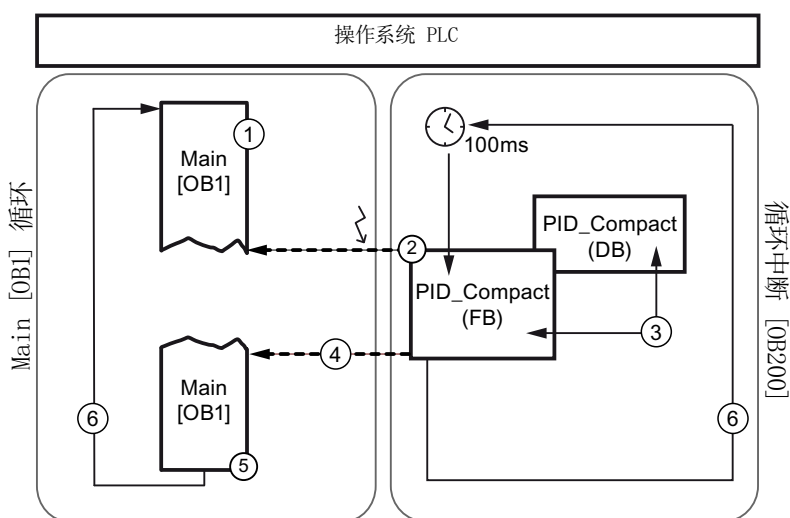
4.2 创建 PID 控制器的组织块

简介

在新的组织块中创建 PID 控制器的块。当前所创建的循环中断组织块将用作新的组织块。

循环中断组织块可用于以周期性时间间隔启动程序，而与循环程序执行情况无关。循环中断 OB 将中断循环程序的执行并会在中断结束后继续执行。

下图显示了带有循环中断 OB 的程序执行：



- ① 程序从 Main [OB1] 开始执行。
- ② 循环中断每 100 ms 触发一次，它会在任何时间（例如，在执行 Main [OB1] 期间）中断程序并执行循环中断 OB 中的程序。
在本例中，程序包含功能块 PID_Compact。
- ③ 执行 PID_Compact 并将值写入数据块 PID_Compact (DB)。
- ④ 执行循环中断 OB 后，Main [OB1] 将从中断点继续执行。相关值将保留不变。
- ⑤ Main [OB1] 操作完成。
- ⑥ 将重新开始该程序循环。

在实例项目中，使用循环中断 OB 调用工艺对象“PID_Compact”。工艺对象“PID_Compact”是 PID 控制器在软件中的映像。可以使用该工艺对象组态 PID 控制器，然后激活该控制器并控制其执行状态。

步骤

要创建 PID 控制器的循环中断 OB，请按以下步骤操作：

1. 打开 Portal 视图。
2. 向现有 PLC 中添加新块。



4.2 创建 PID 控制器的组织块

3. 创建一个名为“PID”的循环中断 OB。请确保已选中“添加新对象并打开”(Add new and open) 复选框。



结果

在项目视图的程序编辑器中打开所创建的循环中断 OB。如果该块没有自动打开，则说明在对话框中没有选中“添加新对象并打开”(Add new and open) 复选框。在这种情况下，请切换到项目视图并打开项目树中的程序块。

在下一部分中，将介绍如何在所创建的块中调用工艺对象“PID_Compact”。

4.3 创建工艺对象 PID 控制器

简介

以下步骤将介绍如何在循环中断 OB“PID [OB200]”中调用工艺对象“PID_Compact”。

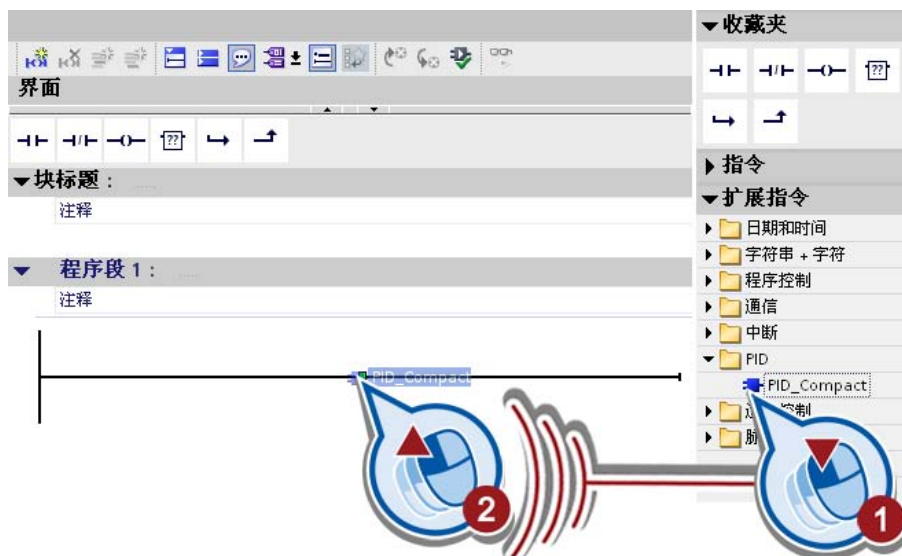
要求

- 已创建带有 PLC S7-1200 的项目。
- 已创建一个循环中断 OB 并在项目视图中将其打开。

步骤

要在循环中断 OB“PID [OB200]”中调用工艺对象“PID_Compact”，请按以下步骤操作：

1. 在组织块“PID [OB200]”的第一个程序段中，创建工艺对象“PID_Compact”。



2. 确定为工艺对象“PID_Compact”创建数据块。



结果

已通过编程设定了在循环中断 OB“PID [OB200]”中调用工艺对象“PID_Compact”并且已创建数据块“PID_Compact_DB”。

在下一部分中，将介绍如何在程序中加载仿真块以便仿真 PID 控制器的输入和输出值。

4.4 加载仿真块

简介

以下步骤介绍了如何在实例项目中加载块“PROC_C”。该块将仿真 PID 控制器的输入和输出值。要使用这些值，请在实例项目中载入库并在第二个程序段中创建该块。

要求

组织块“PID [OB200]”已在项目视图中打开。

步骤

要打开库并复制块，请按以下步骤操作：

1. 将位于以下 Internet 地址中的文件“Simulation Program PID.ZIP”复制到本地硬盘并解压缩该文件。

<http://support.automation.siemens.com/CN/view/zh/40263542>

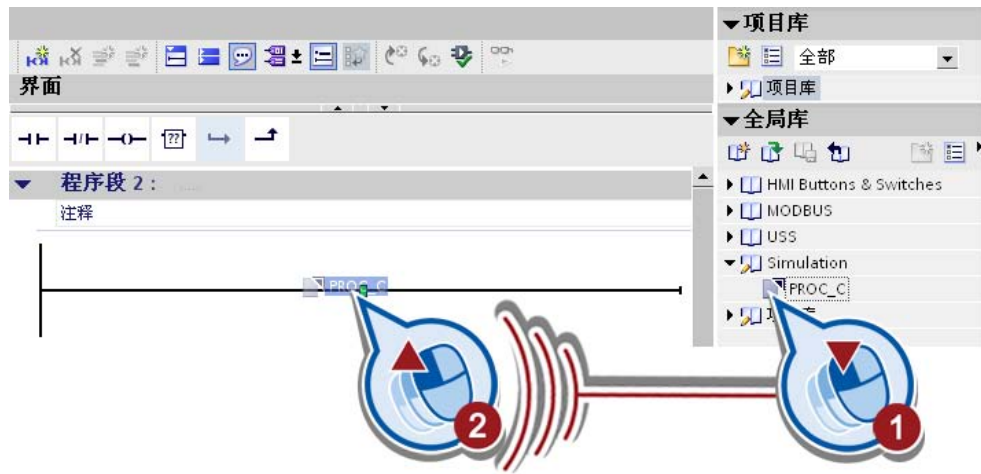
请单击“信息”图标查看相关 ZIP 文件。

2. 解压缩文件“Simulation Program PID.ZIP”。
3. 使用“库”(Libraries) 任务卡打开已解压缩文件目录中的全局库“Simulation”。



该库即被加载。

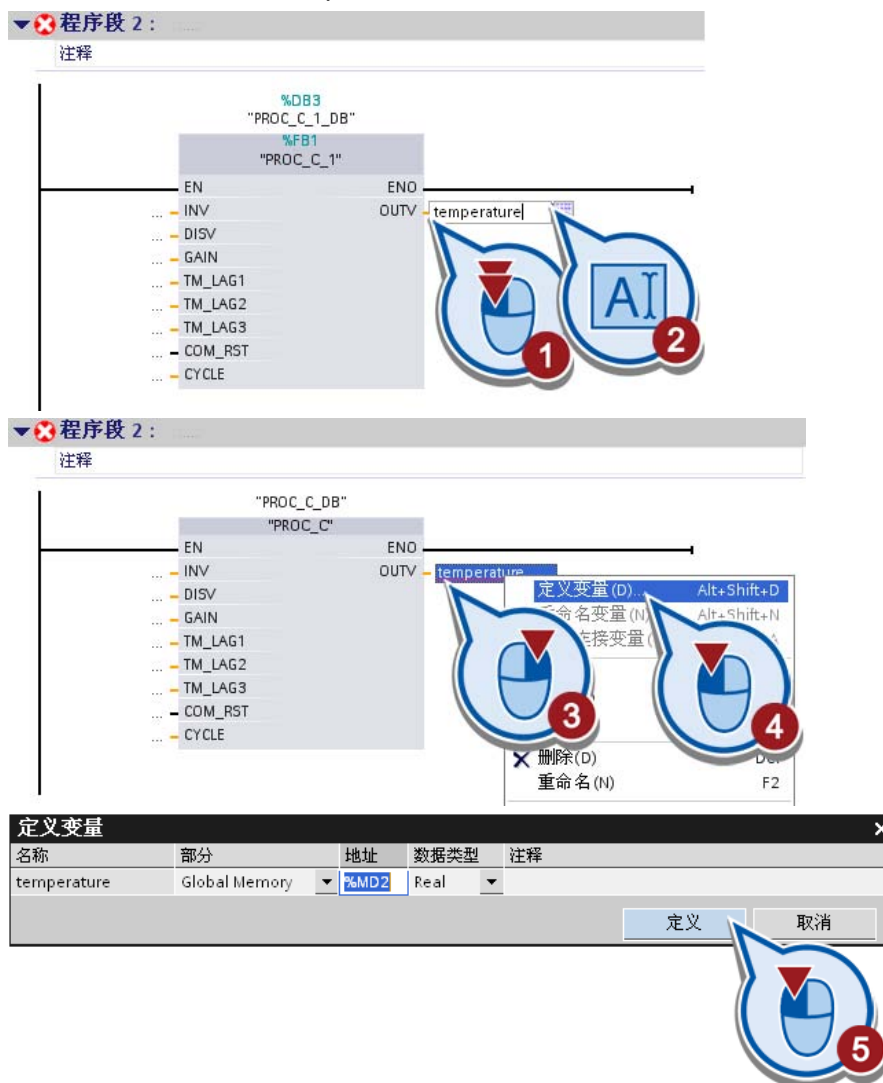
4. 将仿真块“PROC_C”复制到组织块“PID [OB200]”的第二个程序段中。



5. 确定为仿真块“PROC_C”创建数据块。



6. 在 OUTV 参数中定义“temperature”变量。

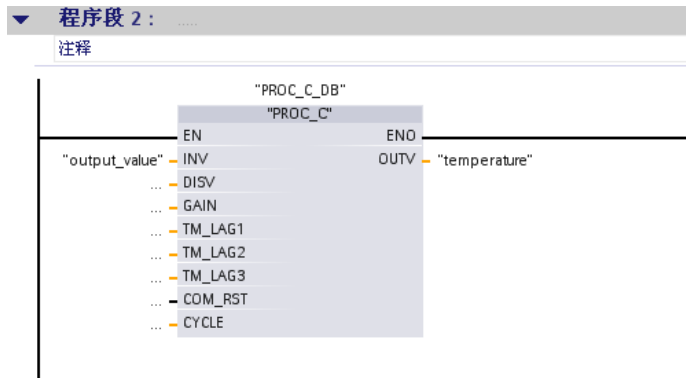


参数 OUTV 的值存储在“temperature”变量中。该参数的值是“PROC_C”块仿真的温度值。

7. 以相同的方式，在参数 INV 中定义变量“output_value”。

结果

在实例项目中加载了用于仿真 PID 控制器的输入和输出值的块。



执行循环中断 OB“PID (OB200)”后，“PROC_C”块会仿真输入和输出值并将这些值加载到背景数据块“PROC_C_DB”中。执行程序时，参数 INV 和 OUTV 的值会映射到变量中。

在下一部分中，将使用工艺对象“PID_Compact”组态 PID 控制器并将其输入和输出链接到仿真块的相应值。

4.5 组态 PID 控制器

简介

以下步骤将介绍如何使用工艺对象“PID_Compact”组态 PID 控制器。

PID 控制器组态的设置

- 控制器类型

控制器类型用于预先选择需控制值的单位。在本例中，将单位为“°C”的“温度”(Temperature) 用作控制器类型。

- 输入/输出参数

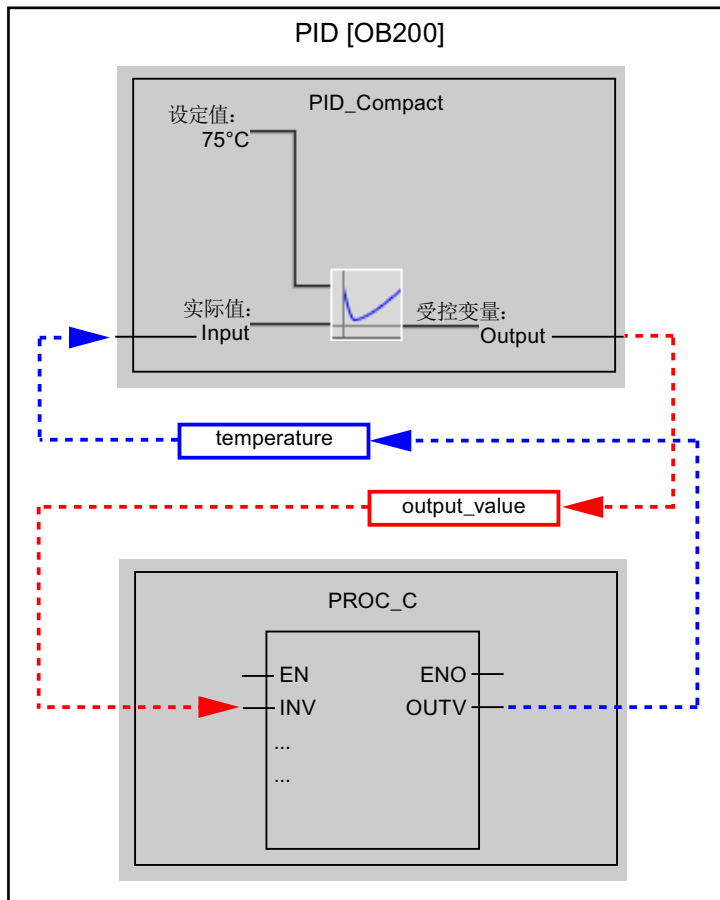
在该区域中，为设定值、实际值和工艺对象“PID_Compact”的受控变量提供输入和输出参数。要在没有其它硬件的情况下使用 PID 控制器，请将“PID_Compact”的输入和输出参数链接到与仿真块“PROC_C”互连的“output_value”和“temperature”变量：

- 实际值由“PROC_C”仿真并用作“PID_Compact”的输入。

在本例中，实际值为映射到“temperature”变量中的加热室中的测量温度。

- 受控变量由工艺对象“PID_Compact”计算，是该块的输出参数。受控变量映射在“output_value”变量中并用作“PROC_C”的输入值。

下图显示了工艺对象“PID_Compact”和仿真块“PROC_C”的互连方式。



要求

- 循环中断 OB“PID [OB200]”处于打开状态。
- 已在组织块“PID [OB200]”中调用了“PID_Compact”块。
- 已在组织块“PID [OB200]”中调用了“PROC_C”仿真块。

步骤

要组态工艺对象“PID_Compact”并将其与仿真块“PROC_C”互连，请按以下步骤操作：

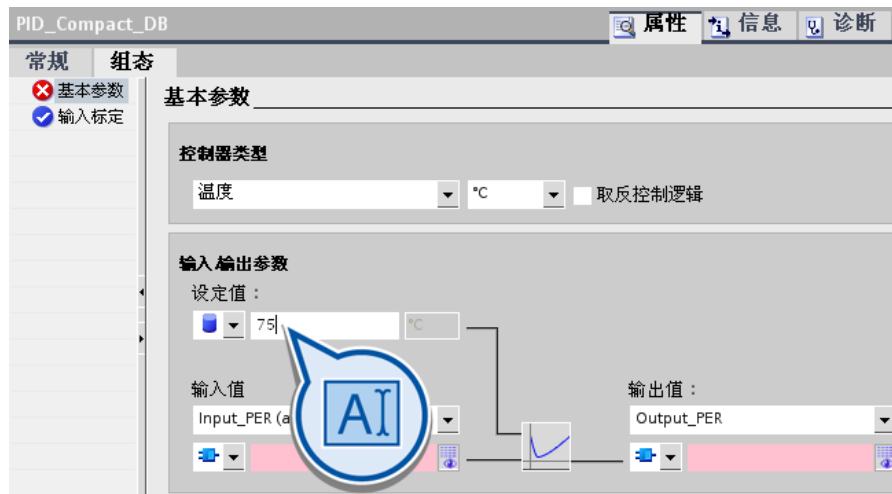
1. 在巡视窗口中打开 PID 控制器的组态。



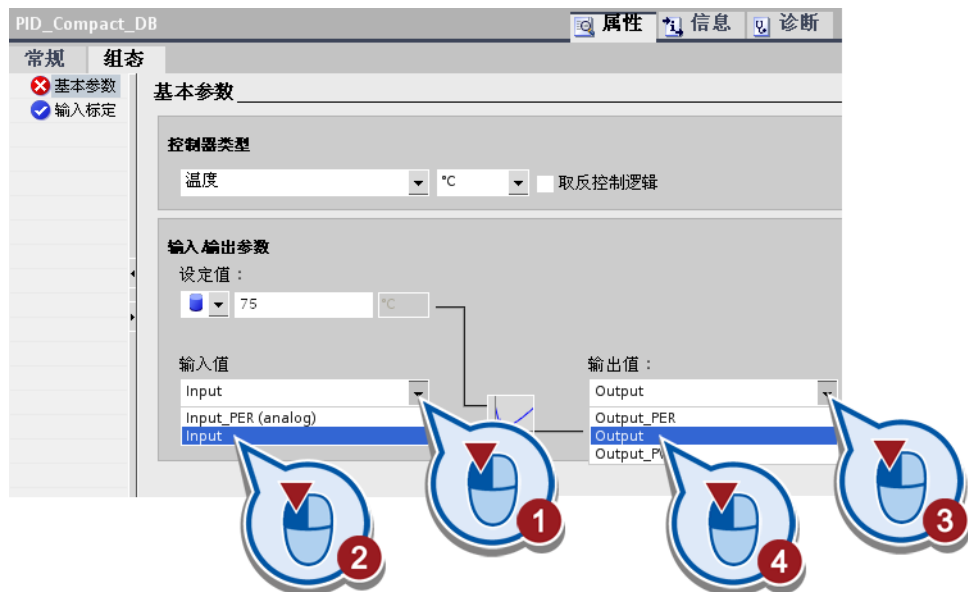
2. 选择控制器的类型。



3. 输入控制器的设定值。



4. 分别为实际值和受控变量选择“输入”(input) 和“输出”(output)。从而指定将使用用户程序的某个变量中的值。

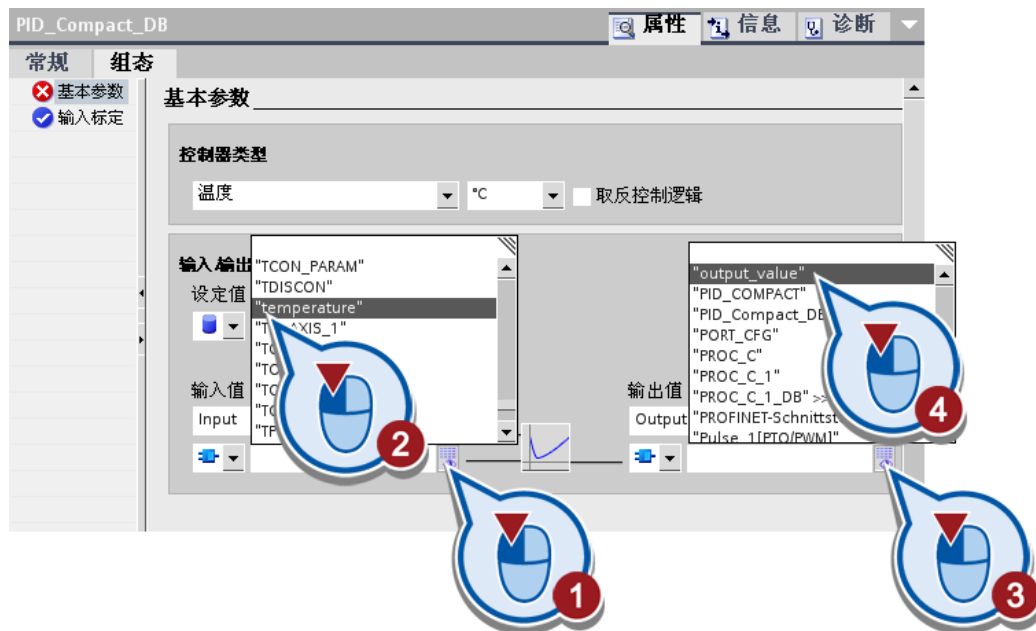


说明

Input(_PER) - Output(_PER)

使用输入和输出为输入或输出参数提供用户程序的实际值。使用 Input_PER 和 Output_PER 可将模拟量输入用作实际值或将模拟量输出用作受控值输出。使用 Output_PWM 来使用通过脉宽调制控制的数字开关输出。在这种情况下，受控变量由变量的开/关次数形成。

5. 将“temperature”变量与实际值互连，并将“output_value”变量与受控变量互连。输入变量的前几个字母时，Intellisense® 将进行相应地过滤。



说明

将块与参数直接互连

下面的部分中将继续使用“temperature”变量。此外，也可以将“PID_Compact”块与仿真块的参数直接互连。参数的寻址方式如下：“块名称”.参数。

如果实际值将是与功能块或 PID_Compact 指令的背景数据块的当前值互连的变量或参数，则可从实际值输入域左侧的下拉列表中选择。

结果

已将 PID 控制器 (PID_Compact) 与仿真块“PROC_C”互连。启动仿真后，PID 控制器会在每次调用组织块“PID [OB200]”时收到新的实际值。

4.6 更改加热室控制

简介

以下步骤将介绍如何根据设备操作对加热室的开/关控制进行编程。设备启动时，将开启加热室。在本部分实例项目中，加热室温度不是恒定温度，而是由创建的 PID 控制器控制。

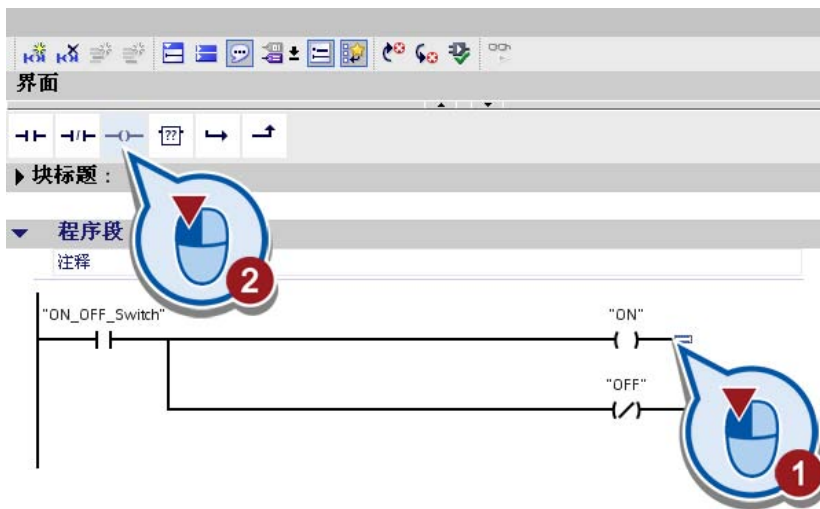
要求

- 组织块“Main [OB1]”已打开。
- 已对组织块“Main [OB1]”的程序段进行编程。

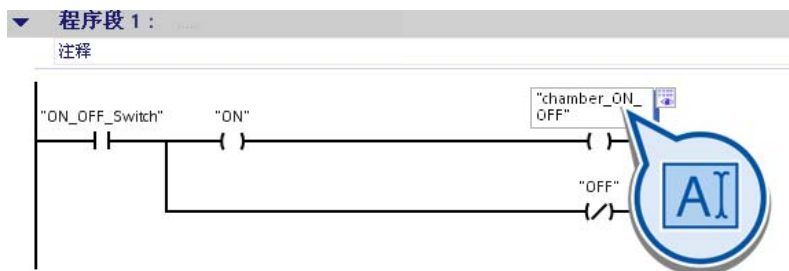
步骤

要更改用于开启或关闭加热室的控制，请按以下步骤操作：

1. 打开组织块“Main [OB1]”的第一个程序段。
2. 在主梯级的末尾插入“输出线圈”指令。



3. 将插入的指令与“chamber_ON_OFF”变量互连。



4. 删除组织块“Main [OB1]”的第三个程序段。

由于加热室的开/关控制是通过组织块的第一个程序段中用于控制设备的程序来进行的，因此不再需要该程序段。



5. 删除第四个程序段，该程序段中对加热室的状态指示灯进行了编程。由于加热室的操作与设备的操作并行进行，因此不再需要状态指示灯。



结果

对加热室的开/关控制进行了编程，可并行操作加热室与设备。

在下一部分中，将对温度值的查询进行编程并将其集成到程序中作为传送带的控制条件。

4.7 在控制程序中集成温度比较作为条件

简介

以下步骤将介绍如何计算 PID 控制器的温度值并集成这些值作为设备的控制条件。巴氏消毒过程由以下组织块控制：

- Main [OB1]
根据 PID 控制器的温度值对设备进行控制的程序在“Main [OB1]”组织块中执行。
- PID [OB200]
PID 控制器的工艺对象“PID_Compact”和仿真块“PROC_C”在组织块“PID [OB200]”中执行。

PID 控制器和控制程序间的连接

按以下方式将 PID 控制器的温度监视集成到组织块“Main [OB1]”的程序中：

“temperature”变量的实际温度值（用作 PID 控制器的输入）应由组织块“Main (OB1)”中的程序读取，并与最小值和最大值进行比较。

- 如果实际温度值在最小值 73°C 和最大值 77°C 之间（设定值 +/- 2°C），则驱动传送带。
- 如果尚未达到最小值 73°C 或已超出最大值 77°C，则复位“conveyor_drive_ON_OFF”变量。

要求

- 组织块“Main [OB1]”已打开。

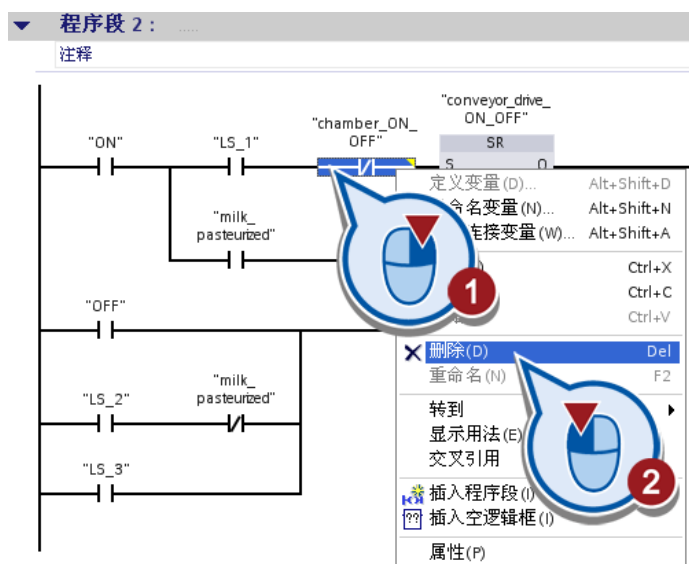
步骤

要插入“temperature”变量值的比较结果作为驱动传送带的条件，请按以下步骤操作：

1. 将指令“大于或等于 (CMP>=)”和“小于或等于 (CMP<=)”插入收藏夹列表。

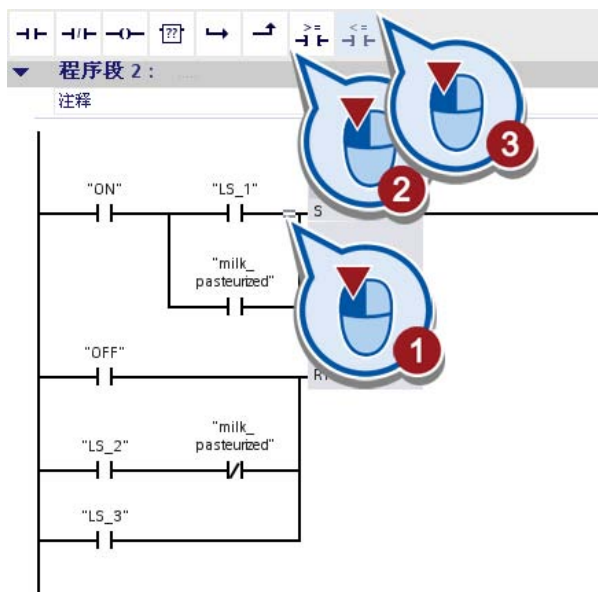


2. 打开组织块“Main [OB1]”的第二个程序段。
3. 删除带有“chamber_ON_OFF”变量的常闭触点。

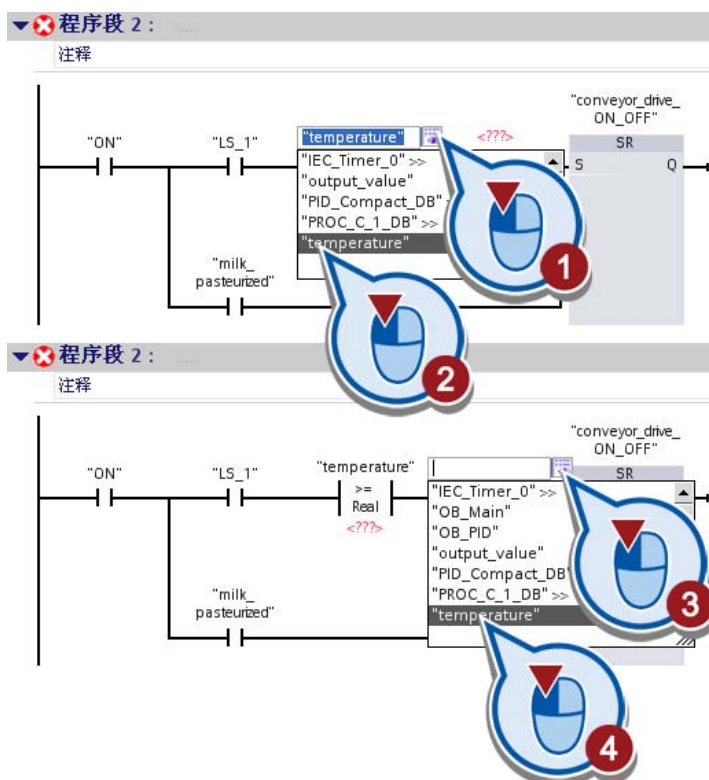


4.7 在控制程序中集成温度比较作为条件

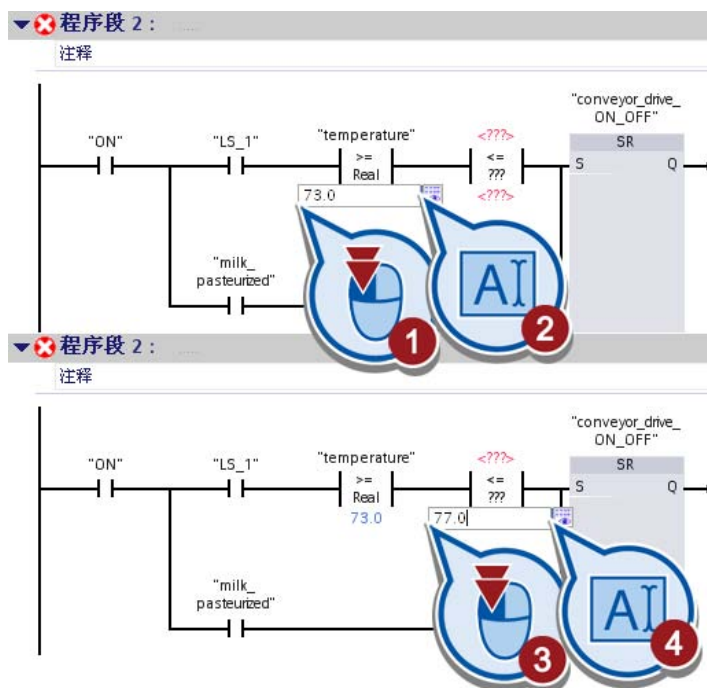
4. 插入收藏夹列表中的指令“大于或等于 (CMP>=)”和“小于或等于 (CMP<=)”。



5. 分别将指令“大于或等于 (CMP>=)”和“小于或等于 (CMP<=)”与“temperature”变量互连。



6. 在指令“大于或等于”下方的操作数占位符中输入值“73.0”并在指令“小于或等于”下方的操作数占位符中输入值“77.0”。



7. 单击工具栏上的“保存”(Save) 按钮以保存该项目。

结果

已对温度值进行编程，使其作为驱动传送带的条件。当设备处于运行状态并且至少满足下列条件之一时，传送带开始运动：

- 光栅“LS1”激活并且加热室中的温度介于 73°C 和 77°C 之间。这意味着：
 - 变量“LS_1”的信号状态为“1”。
 - 变量“temperature”的值介于“73”和“77”之间。
- 牛奶已经过巴氏消毒。此时，变量“milk_pasteurized”的信号状态为“1”。

在下一部分中，将介绍如何在 HMI 画面中调整状态指示灯的连接。

4.8 调整 HMI 画面

简介

以下步骤将介绍如何将加热室的状态指示灯替换为 I/O 域。

在扩展实例中，如果变量“chamber_ON_OFF”的值为“1”，加热室将启动。由于现在通过 PID 控制器控制温度，因此不再使用该 PLC 变量。当前温度现在应显示在 I/O 域中，而不是通过状态指示灯指示。

I/O 域

可以通过 I/O 域在 HMI 画面中输入或显示过程值。

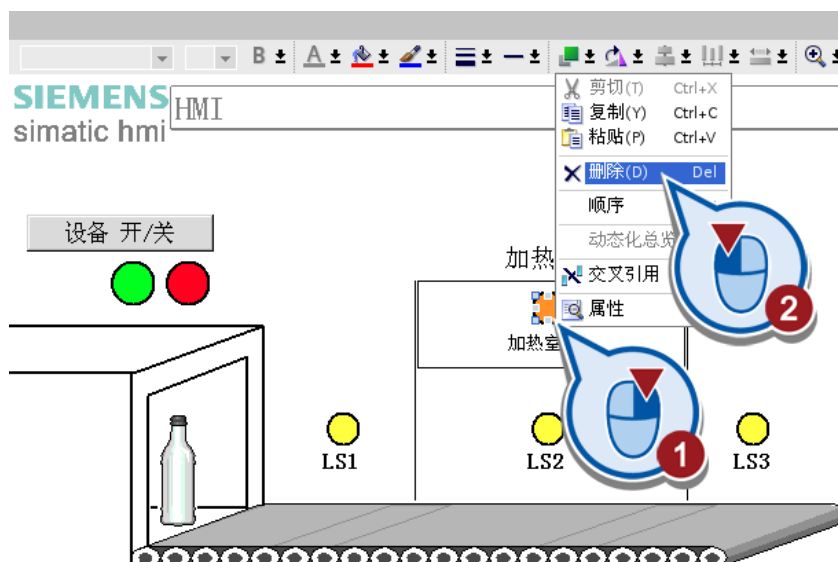
要求

- 已创建程序。
- HMI 画面已在编辑器中打开。

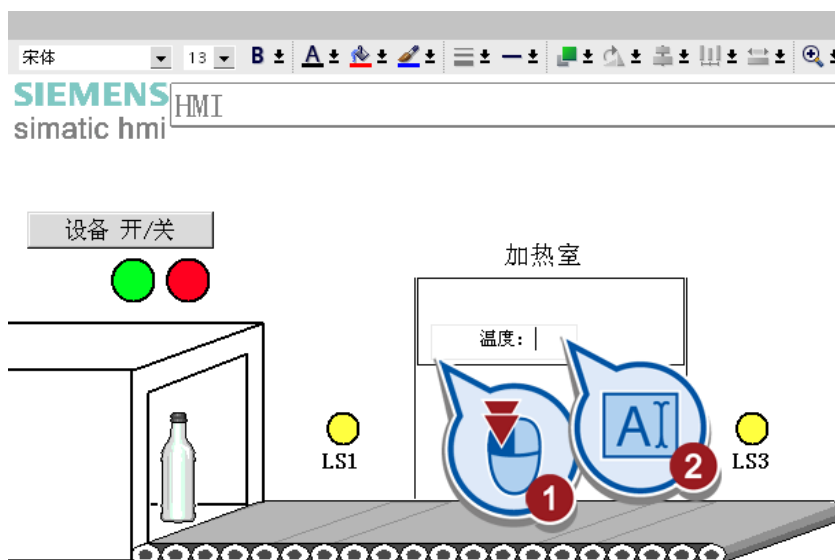
步骤

要在 HMI 画面中显示温度值，请按以下步骤操作：

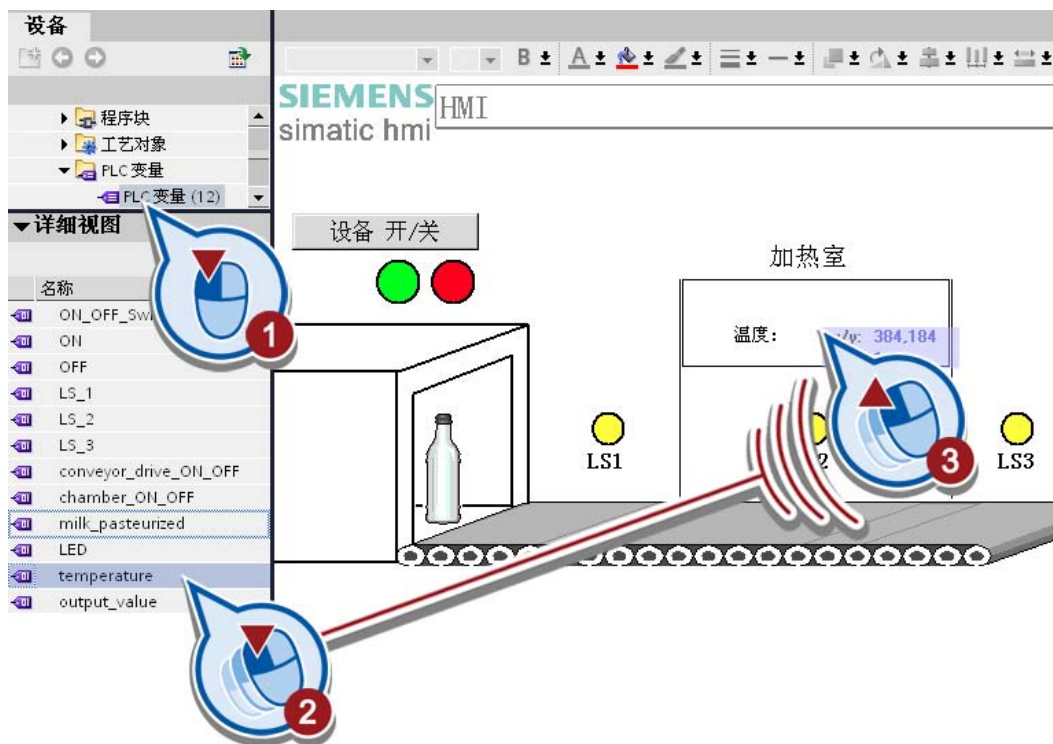
1. 删除橙色的圆。



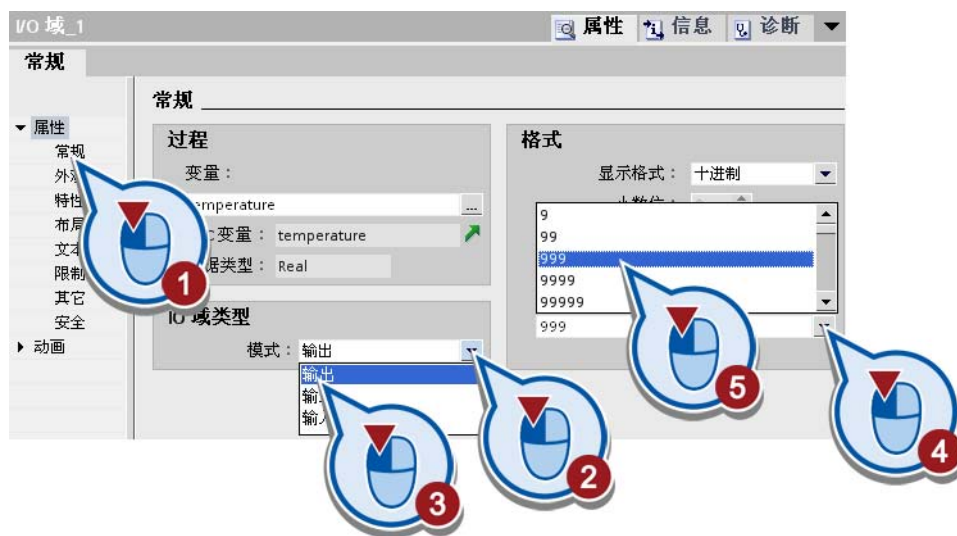
2. 将文本域的内容更改为“温度:”。



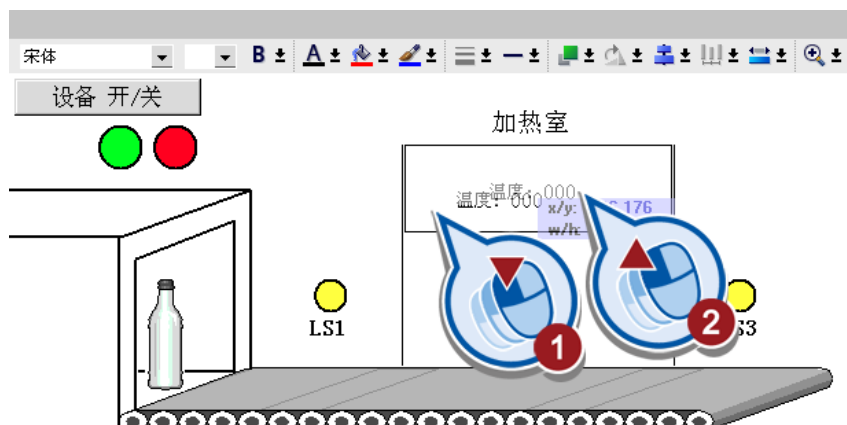
3. 通过 PLC 变量的详细视图在 HMI 画面中插入 I/O 域。



4. 更改 I/O 域的域类型和格式。



5. 调整文本和 I/O 域的大小及位置。



6. 单击工具栏上的“保存”(Save) 按钮以保存该项目。

结果

为“temperature”变量的值创建了输出域。执行用户程序时，变量的当前值将显示在 HMI 设备中。

4.9 在线模式下激活 PID 控制器

简介

以下步骤介绍如何仿真 PID 控制器的功能。为此，首先将程序加载到 PLC 中并在在线模式下激活 PID 控制器。优化控制器后在趋势窗口中观察进行中的控制。

优化控制器

使用优化步骤调整控制器以适应受控对象。可选择以下两种方式来优化控制器：

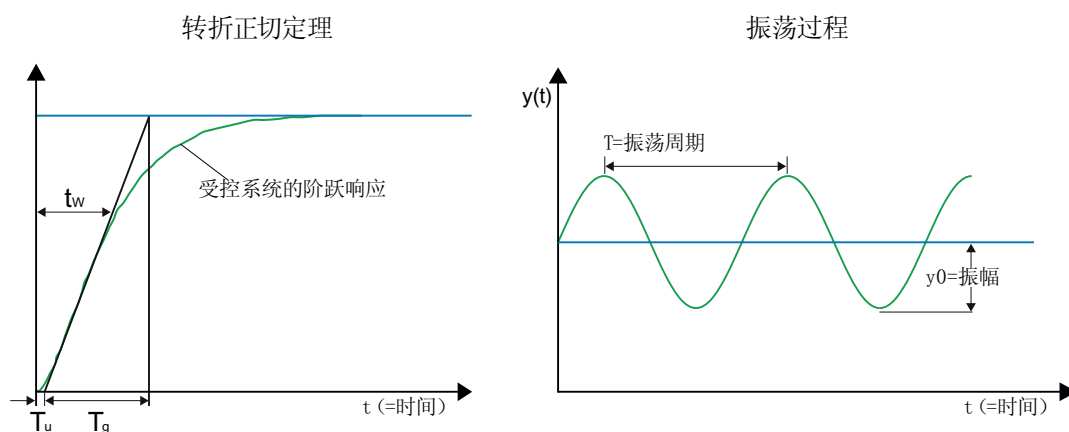
- 启动调节

在启动调节中将应用转折正切定理。该定理用于确定阶跃响应的时间常数。受控对象的阶跃响应中存在一个转折点。对该转折点应用切线；使用该切线可确定过程参数延迟时间 (T_u) 和恢复时间 (T_g)。根据这些过程参数将确定优化的控制器参数。设定值与实际值之间必须至少相差 30%，才能使用转折正切定理确定参数。否则，将通过振荡过程和功能“运行中调节”自动确定控制器参数。

- 运行中调节

运行中调节使用振荡过程来优化控制器参数。通过该过程可间接确定受控对象的行为。增益因子将增大，直到其达到稳定限制且受控变量均匀振荡。控制器参数将基于振荡周期进行计算。

下图显示了分别采用转折正切定理（启动调节）和振荡过程（运行中调节）时受控对象的阶跃响应：



步骤

要激活 PID 控制器并启动仿真，请按以下步骤进行操作：

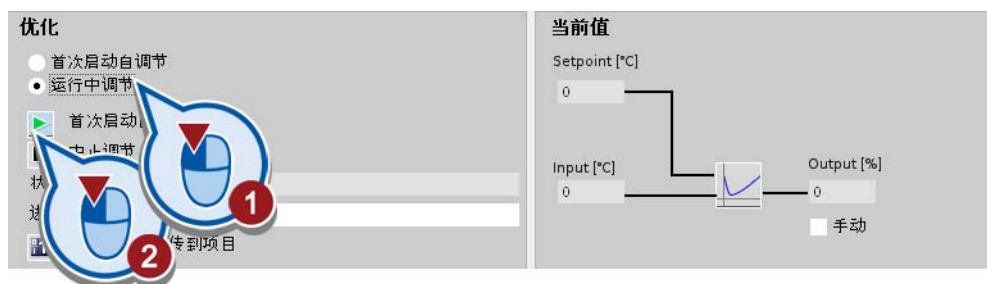
1. 将程序加载到 PLC 中并激活在线连接。更多信息，请参见“将程序加载到目标系统 (页 52)”部分。
2. 从项目树中启动调试。



3. 启用调试窗口的功能。

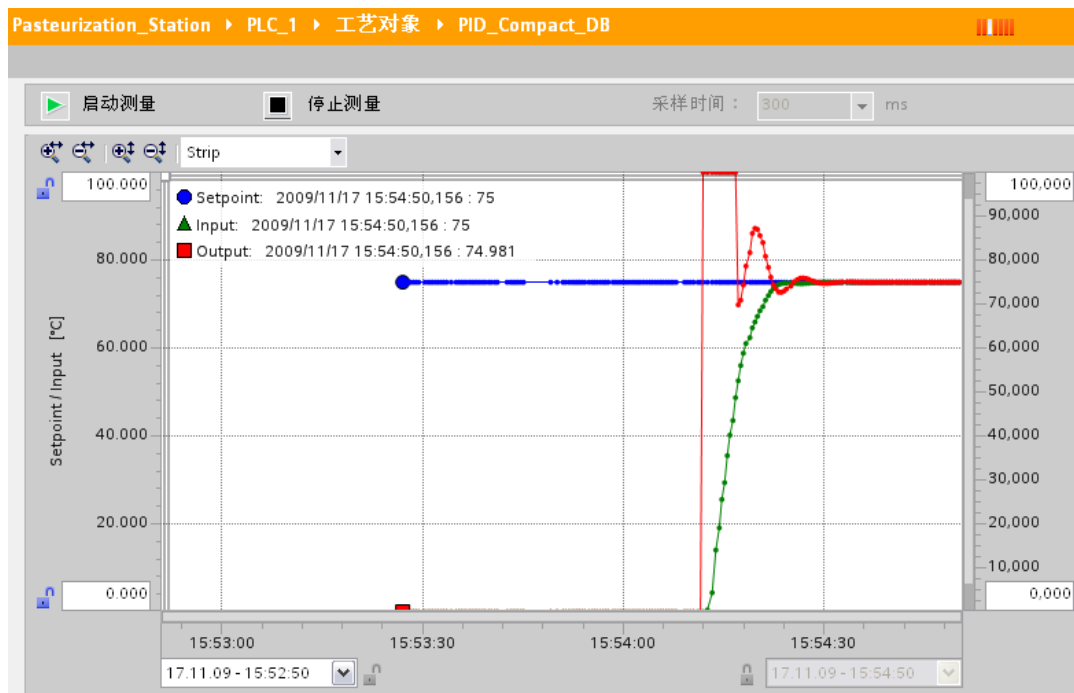


4. 通过启动运行中调节，对控制器优化进行微调。



启动自调节功能。当前工作步骤和发生的任何错误都将显示在“状态”(Status) 域中。进度条将显示当前工作步骤的进度。优化控制器需要分多步执行。每个步骤的进度都从“0”开始。如果自调节完成且没有错误消息，则完成 PID 参数的优化。PID 控制器会切换到自动模式，并使用经过优化的参数。

5. 监视“调试”(Commissioning) 窗口顶部的曲线形状。



说明

曲线的形状

曲线的形状最终会越来越接近设定值。要监视曲线的完整形状，请使用图形显示底部区域中的垂直滚动条。

控制器切换到自动模式后，控制器通过自动调整受控变量来响应由“PROC_C”块仿真的温度变化。

结果

已在在线模式下激活 PID 控制器。在 CPU 通电和重新启动期间，将保留自动模式启动前优化的 PID 参数。如果在将项目数据再次加载到 CPU 时，希望重复使用已在 CPU 中优化过的 PID 参数，可以把 PID 参数保存在项目中。

实例“运动”

5.1 简介

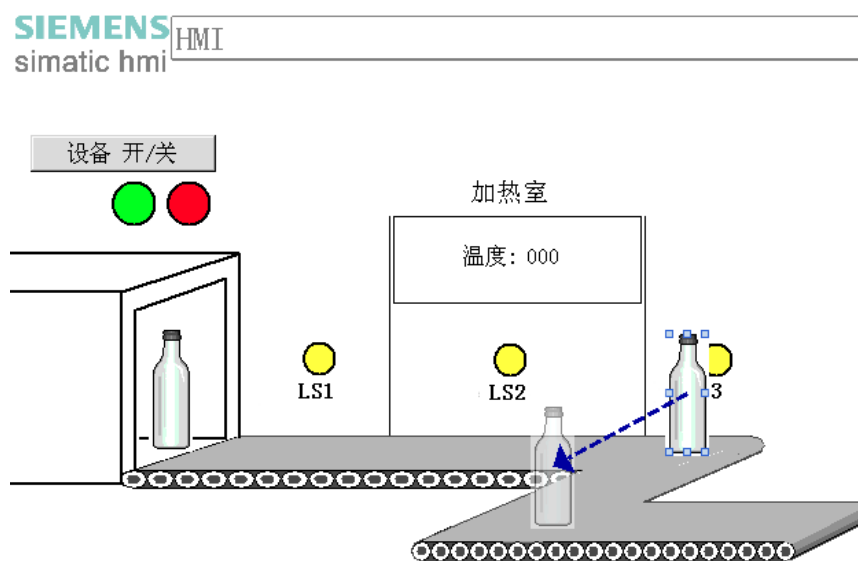
加载项目

如果跳过了之前的章节，则可通过打开项目“Extendet_Example_PID”加载最后一章结尾的项目状态。本章结尾的结果也另存为一个名为“Extendet_Example_Motion”的项目。

有关加载项目的更多信息，请参见“加载项目 (页 18)”部分。

运动的使用

在本章中，将在实例中添加传送带。从光栅“LS3”的位置开始，瓶子通过连接片传送到另一条传送带。



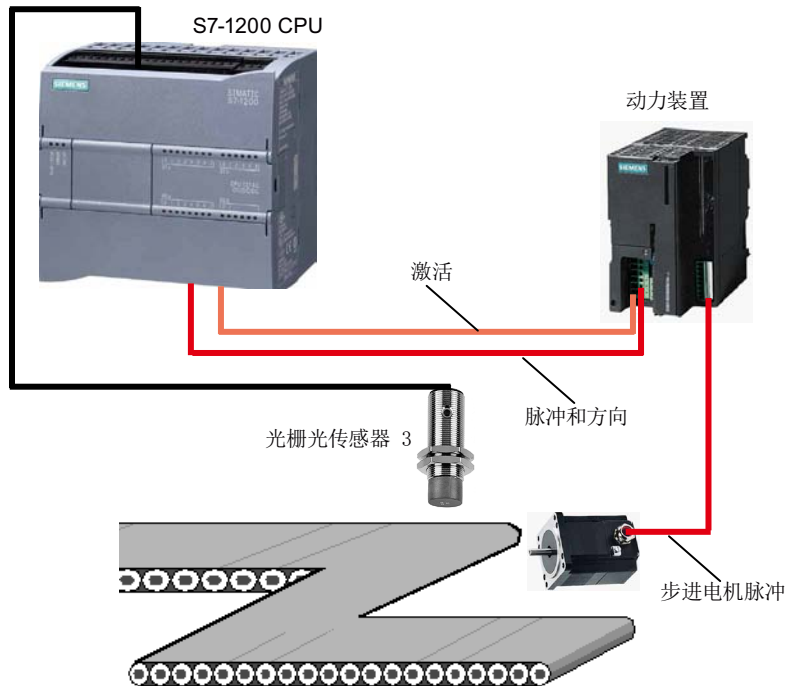
使用工艺对象“轴”控制两条传送带间的连接。PLC 按以下方式集成到项目中：

- 如果机器启动，则在连接片上启用用于激活步进电机的轴。
- 激活光栅“LS3”时，将触发电机的轴。
- 瓶子相对于光栅“LS3”的位置移动到第二条传送带。
- 到达位置时，将激活第二条传送带。

硬件配置

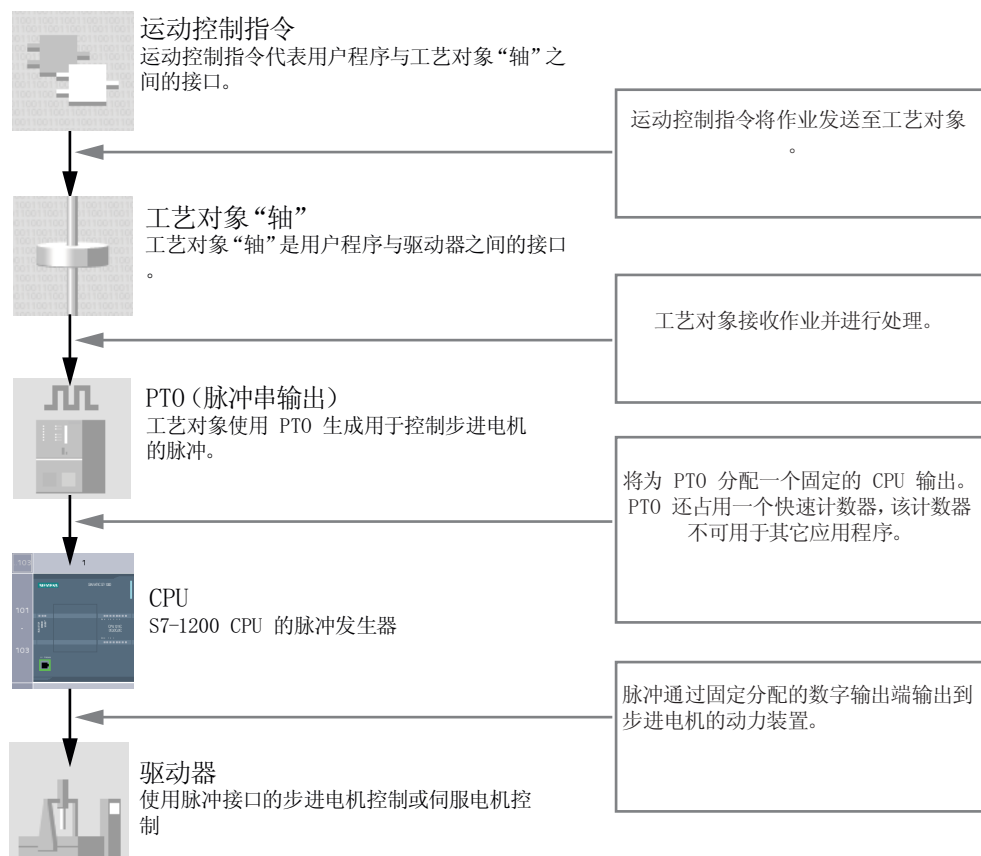
除 PLC 之外无需其它硬件即可执行该实例。 如果使用具有继电器输出的 PLC，则必须将信号板用于 PTO 的输出。 PTO（脉冲串输出，Pulse Train Output）通过输出生成快速脉冲串，这些脉冲串用来控制运动控制操作。

下图针对本例显示了可能的硬件配置外观：



脉冲生成顺序

下图显示了通过运动控制指令生成脉冲的顺序。



用于控制步进电机的脉冲通过输出生成并输出到步进电机的动力装置后, 这些脉冲将由步进电机的动力装置转化为轴向运动。 传送带间的连接由轴进行驱动。

步骤

在本章中，将执行以下步骤：

- 创建工艺对象“轴”。
- 通过以下方式组态工艺对象“轴”：
 - 将脉冲发生器分配给工艺对象。
 - 将脉冲发生器组态为 PTO（脉冲串输出，Pulse Train Output）。
 - 将高速计数器分配给脉冲发生器（自动选择）。
- 创建两个运动控制指令：
 - 一个用于启用轴 (MC_Power)。
 - 一个用于相对定位光栅“LS3”的位置 (MC_MoveRelative)。

5.2 插入工艺对象“轴”

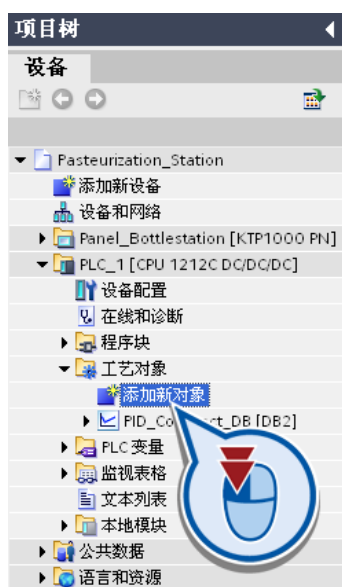
简介

以下步骤介绍了如何创建新的工艺对象“轴”。

步骤

要创建工艺对象“轴”，请按以下步骤操作：

1. 在项目树中创建新的工艺对象。



2. 创建新对象“轴”。



在工作区中启动工艺对象的组态。

结果

创建了工艺对象“轴”。该对象存储在项目树的“工艺对象”(Technology objects) 文件夹中。

在下一部分，将介绍如何组态所创建的工艺对象。

5.3 组态工艺对象“轴”

简介

以下步骤介绍了如何组态工艺对象“轴”。

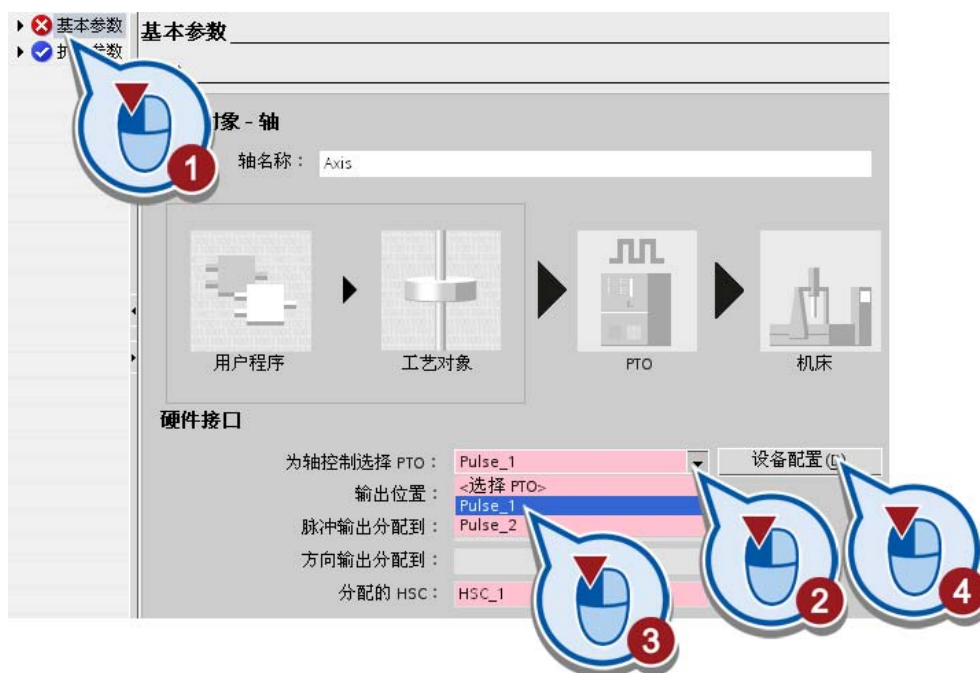
要求

- 已创建工艺对象“轴”。
 - 已启动工艺对象的组态。
- 如果组态没有自动打开，则从项目树启动组态。

步骤

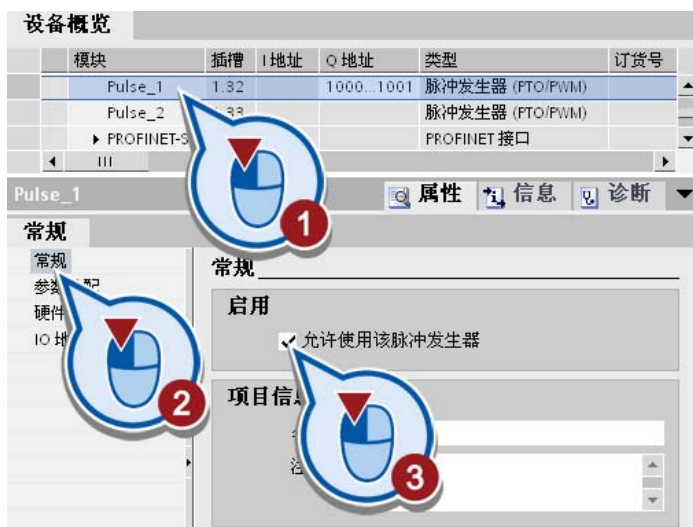
要组态驱动器控制，请按以下步骤操作：

1. 选择 PTO 驱动器控制的脉冲输出“Pulse_1”并切换到设备组态。



选择“Pulse_1”时，将自动分配高速计数器“HSC_1”。设备视图随即打开。

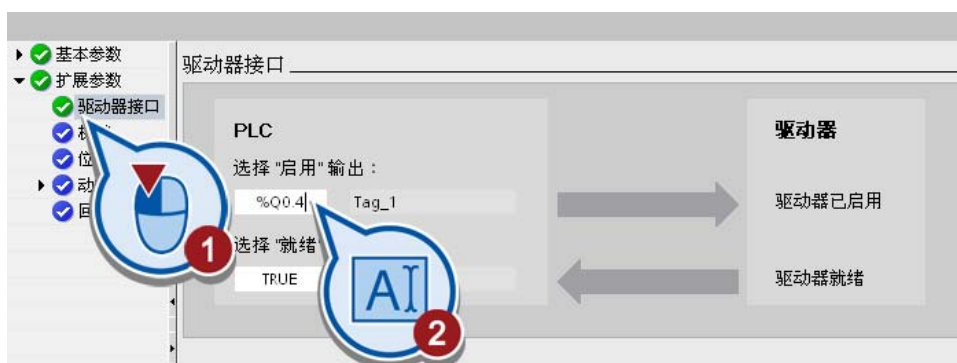
2. 激活脉冲发生器。



3. 使用参数分配将脉冲发生器组态为 PTO。



4. 指定启用驱动器所使用的 PLC 输出。



5. 单击工具栏上的“保存”(Save) 按钮以保存该项目。

结果

已将创建的工艺对象“轴”分配给脉冲发生器“Pulse_1”并将其参数化为 PTO。高速计数器“HSC 1”已自动激活。通过脉冲发生器输出的脉冲将借助高速计数器进行计数。

如果已正确组态工艺对象，则“基本参数”(Basic parameters) 区域的状态和“扩展参数”(Extended parameters) 区域的状态将在组态窗口中以绿色显示。

5.4 启用轴

简介

以下步骤介绍了如何启用传送带的电机轴。

要启用或禁用轴，请使用电机控制指令“MC_Power”。必须在程序中为每个轴调用一次该指令。

使用运动控制指令“MC_Power”可集中启用或禁用轴。

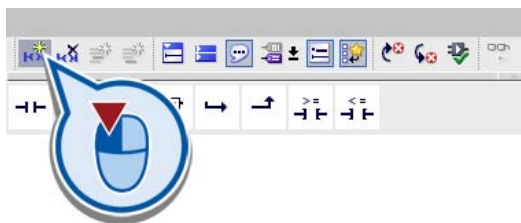
- 如果启用了轴，则分配给此轴的所有运动控制指令都将被启用。
- 如果禁用了轴，则用于此轴的所有运动控制指令都将无效。将中断当前的所有作业。

要求

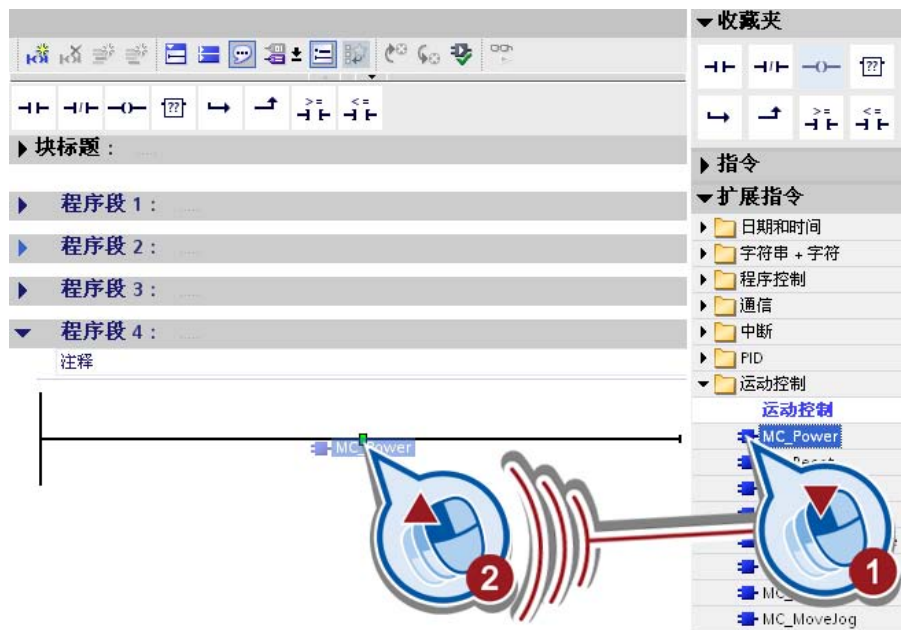
- 已正确组态工艺对象。
- 组织块“Main [OB1]”已打开。

步骤

1. 在组织块“Main [OB1]”中创建新程序段。



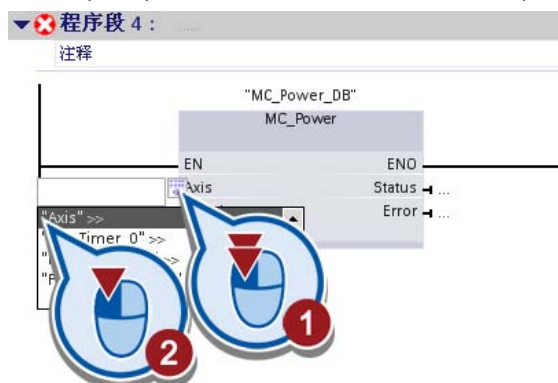
2. 在新程序段中创建运动控制块“MC_Power”。



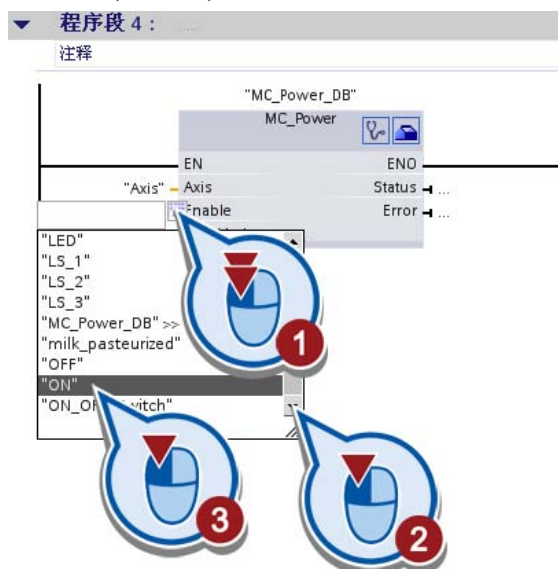
3. 对创建新数据块进行确认。



4. 在“轴”(Axis) 输入端，选择先前组态的“轴”(Axis)。



5. 在“使能”(Enable) 输入端，选择“ON”变量。



6. 保存项目。

结果

在程序中插入了用于启用轴的指令“MC_Power”并将其分配给了工艺对象“轴”。

是否启用轴取决于“使能”(Enable) 输入中“ON”变量的值：

- “ON”变量位的值为“0”（机器关闭）时，禁用轴。
- “ON”变量位的值为“1”（机器接通）时，启用轴。

在下一部分，将介绍如何对传送带相对于起始点的运动进行编程。

5.5 相对定位轴

简介

以下步骤介绍了如何使用运动控制指令“MC_MoveRelative”对第二条传送带相对于起始位置的运动进行编程。

运动的定义如下：

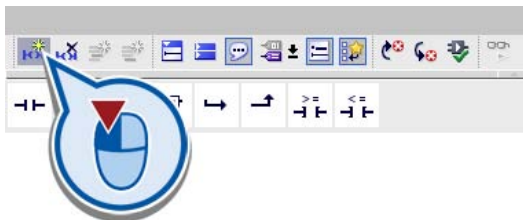
- 运动的起始位置是光栅“LS3”的位置。
- 结束位置在第一条传送带和第二条传送带之间轴的正方向上 0.5 米处。
- 到达结束位置时，将激活第二条传送带。

要求

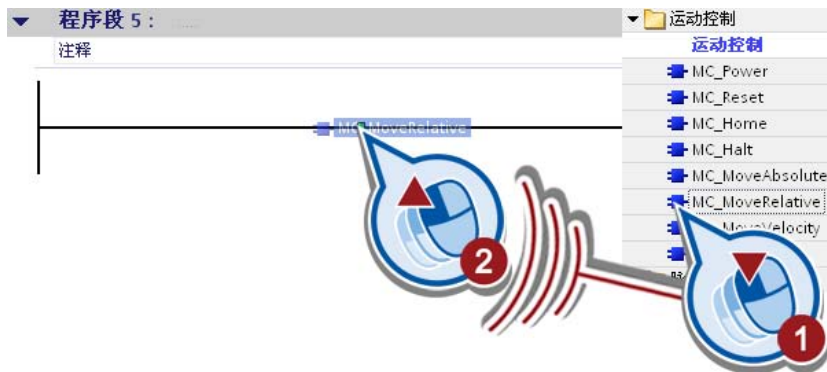
- 已创建工艺对象“轴”并已正确组态。
- 已在组织块“Main [OB1]”中创建了运动控制指令“MC_Power”。
- 组织块“Main [OB1]”已打开。

步骤

1. 在组织块“Main [OB1]”中创建新程序段。



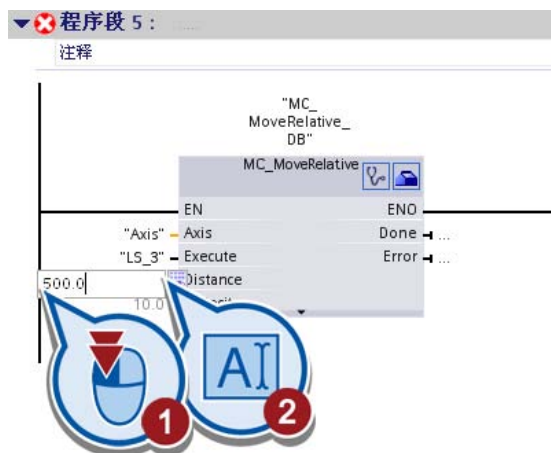
2. 在新程序段中创建块“MC_MoveRelative”。



3. 在“调用选项”(Call options) 对话框中，单击“确定”(OK) 确认创建新数据块。



4. 在“轴”(Axis) 输入端，选中先前组态的“轴”(Axis) 对应的框。
5. 在“执行”(Execute) 输入端，选中“LS_3”变量的框。
6. 在“距离”(Distance) 输入端，输入值“500.0”，并按回车键进行确认。



“距离”(Distance) 测量单位的标准设置是毫米，并且之前已在组态中采用。

7. 在“完成”(Done) 输出端，创建新变量“start_conveyor_2”。

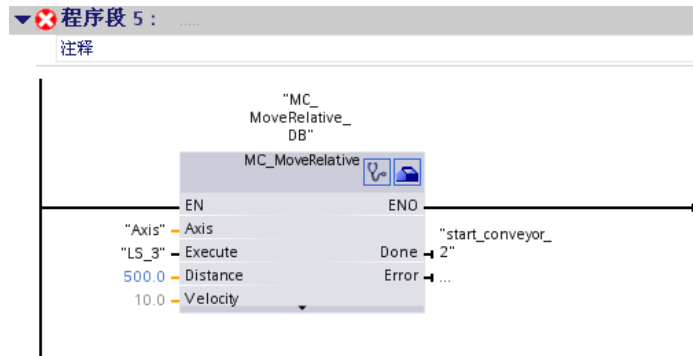
The image shows two screenshots of the SIMATIC Manager interface. The top screenshot shows the 'MC_MoveRelative' block with its 'Done' output connected to the variable 'start_conveyor_2'. A callout bubble with a red '1' points to the 'Done' output, and another with a red '2' points to the variable name. The bottom screenshot shows the same block, but with a context menu open over the variable name. Callout bubbles with red numbers 3, 4, and 5 point to the 'Done' output, the context menu, and the '定义变量(D)...' dialog box, respectively. The dialog box shows the variable name 'start_conveyor_2', the memory address '%M0.6', and the data type 'Bool'.

名称	部分	地址	数据类型	注释
start_conveyor_2	Global Memory	%M0.6	Bool	

8. 单击工具栏上的“保存”(Save) 按钮以保存该项目。

结果

对传送带相对于光栅“LS3”位置的运动进行了编程。



激活光栅“LS3”时，运动开始。到达目标位置时，将在“完成”(Done) 输出端置位“start_conveyor_2”变量的位，从而使第二条传送带开始运动。

在下一部分，将介绍如何向 HMI 画面添加更多的元素以可视化编程过程的执行情况。

5.6 扩展 HMI 画面

5.6.1 更改图形对象“传送带”

简介

以下步骤介绍了如何在 HMI 画面中使用图形“ConveyorMotion.wmf”替换现有图形对象“传送带”。该图形包含两条传送带而不是一条；两条传送带彼此相连。

要求

已解压缩 ZIP 文件“WinCC Graphics”并存储在本地目录中。已在“工具箱”(Toolbox) 任务卡的“图形”(Graphics) 窗格中为该目录创建了链接。请参见“图形对象“传送带”(页 78)”部分。

步骤

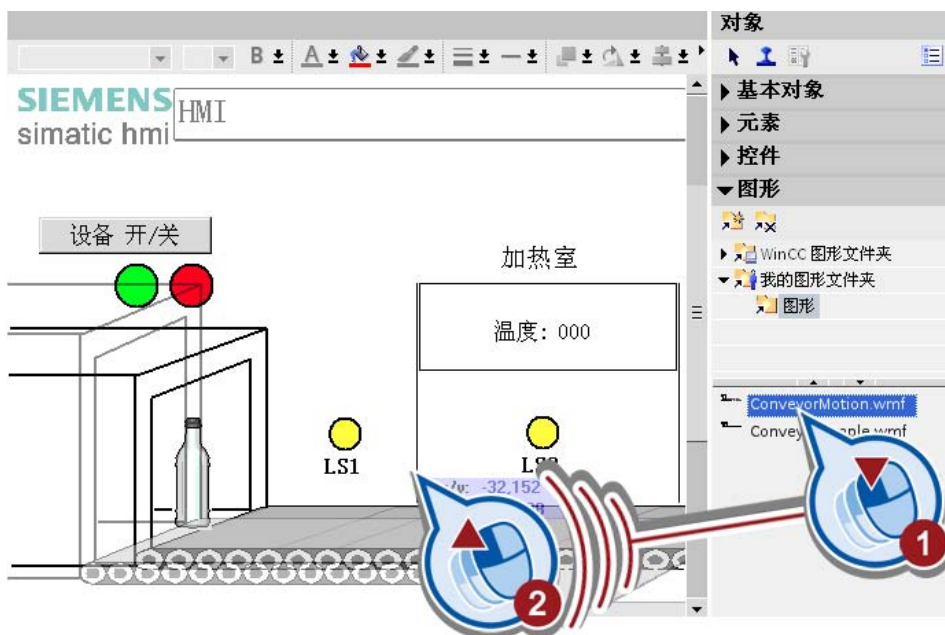
要扩展 HMI 画面，请按以下步骤操作：

1. 打开 HMI 画面。
2. 在“工具箱”(Toolbox) 任务卡的“图形”(Graphics) 窗格中，打开指向本地文件夹“WinCC 图形”(WinCC Graphics) 的链接。
3. 禁用“大图标”(Large icons) 选项。

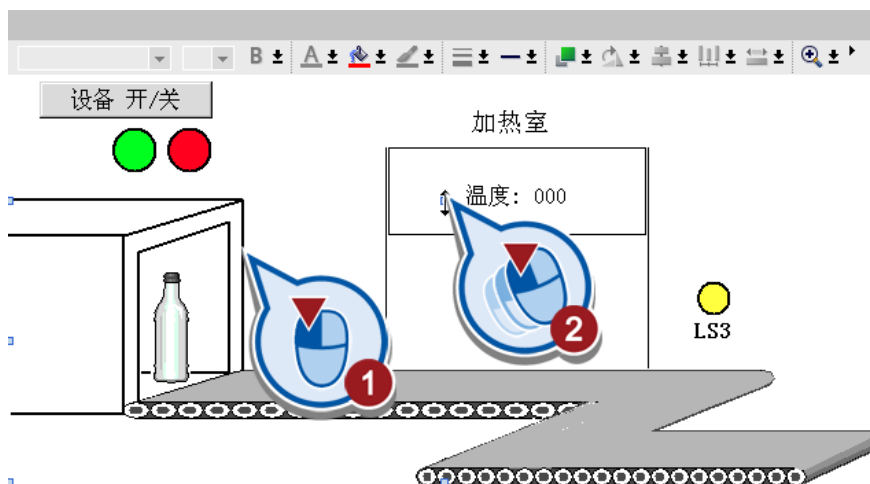


4. 通过拖放操作将图形对象“ConveyorMotion.wmf”拖动到编辑器区域以替换现有传送带。

确保在插入对象时光标旁显示箭头符号。如果光标与加号一起显示，则将添加而非替换图形。



5. 如必要，可通过缩放功能调整该图形对象。



结果

替换了图形对象“传送带”。

在下一部分，将介绍如何向 HMI 画面添加另一个动态化图形对象。

5.6.2 创建另一个图形对象“瓶子”

简介

在下一部分，将创建另一个瓶子及其从第一条传送带到第二条传送带的运动动画。

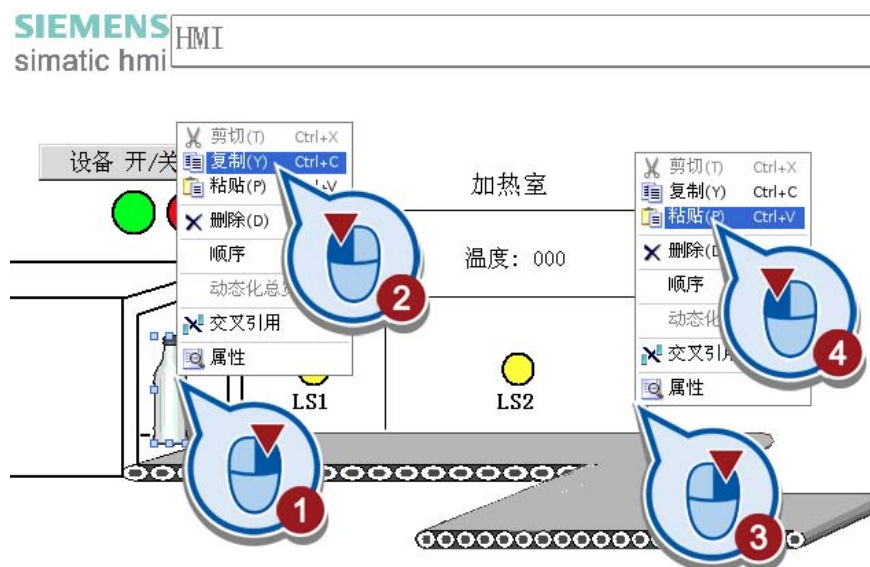
要求

HMI 画面处于打开状态。

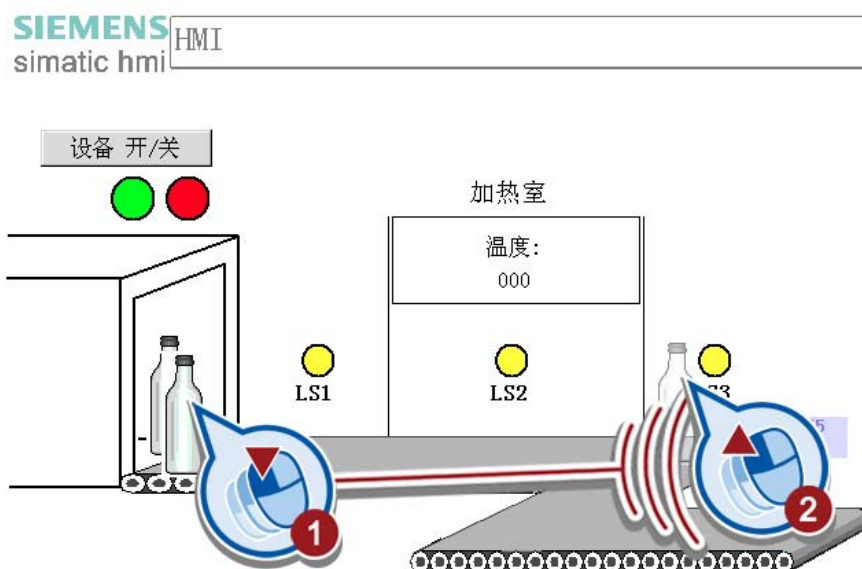
步骤

要在 HMI 画面中使第二个瓶子动态化，请按以下步骤操作：

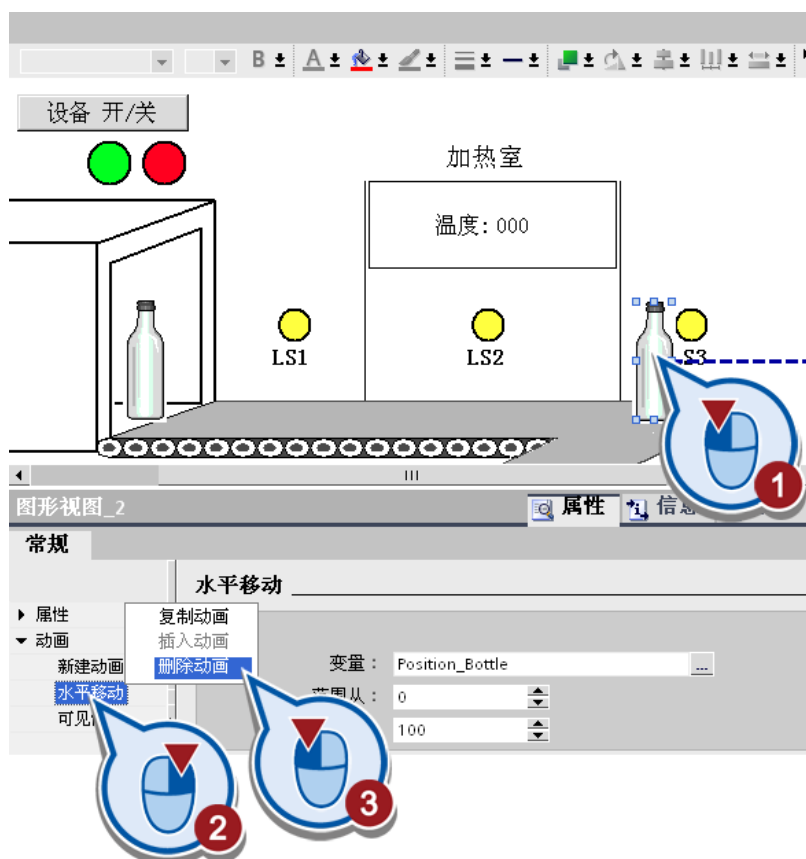
1. 复制现有瓶子。



2. 将复制的瓶子放置在第一个瓶子的运动动画结束处，即光栅“LS3”的前面。



3. 删除所复制瓶子的“水平运动”动画。

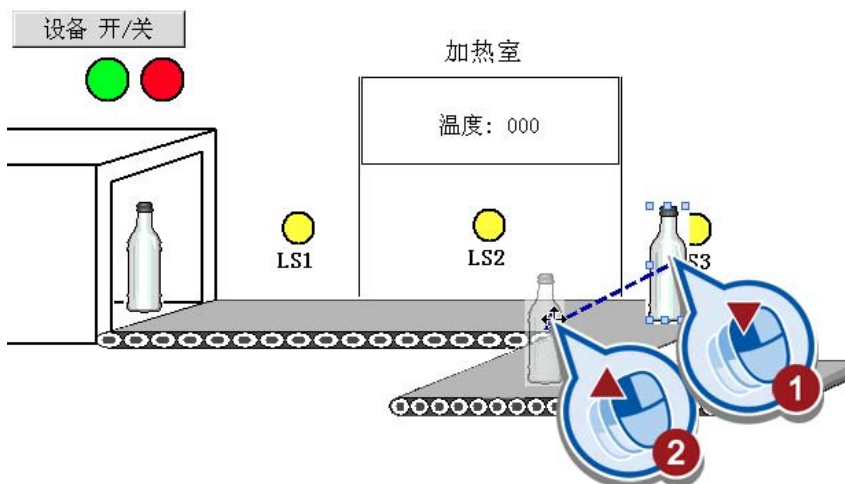


4. 创建“对角线移动”(Diagonal movement) 动画。

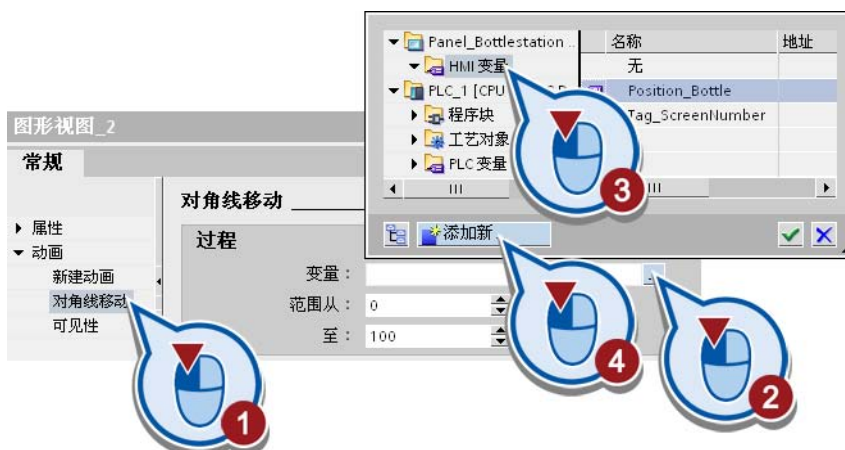


运动动画以蓝色箭头显示。

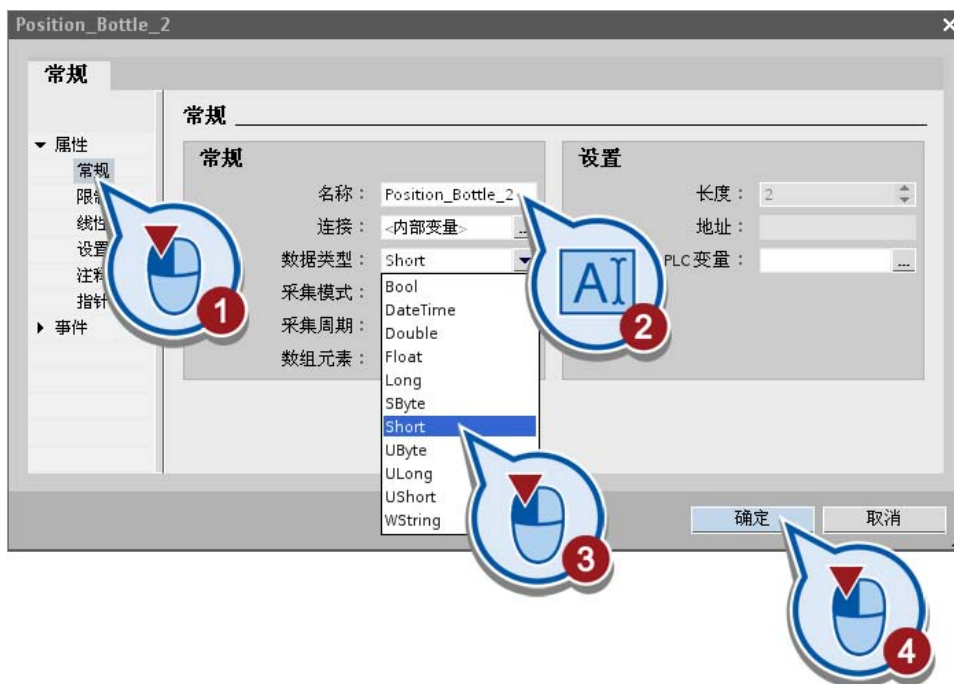
5. 沿着箭头提示将以图形形式表示的透明瓶子拖动到第二条传送带。



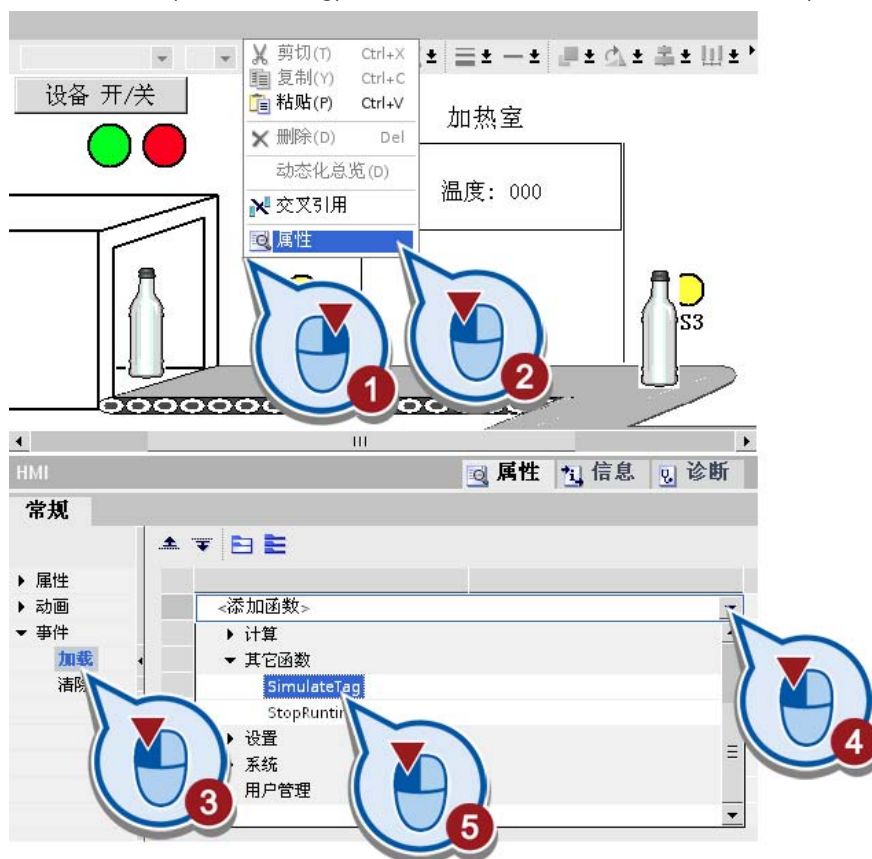
6. 为对角线运动动画创建新的 HMI 变量。



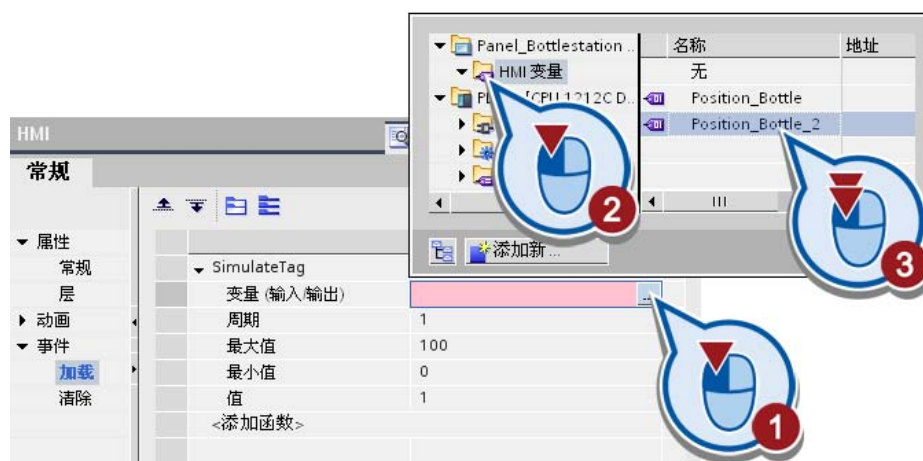
7. 使用“Position_Bottle_2”作为名称，“Short”作为数据类型。



8. 将“仿真变量”(SimulateTag) 功能添加到 HMI 画面的事件“已加载”(Loaded)。



9. 将“Position_Bottle_2”变量分配给“仿真变量”(Simulate Tag) 功能。



结果

创建了用来显示第二个瓶子从第一条传送带运动到第二条传送带的动画。

在下一部分，将更改传送带上瓶子的可见性设置。

5.6.3 连接 HMI 对象与运动指令

简介

以下步骤介绍了如何根据程序进度动态化瓶子在 HMI 画面中的可见性。

可使用“MC_MoveRelative”指令中“Busy”参数的信号状态设置可见性。该参数包含有关是否在该时刻执行运动控制指令的信息。

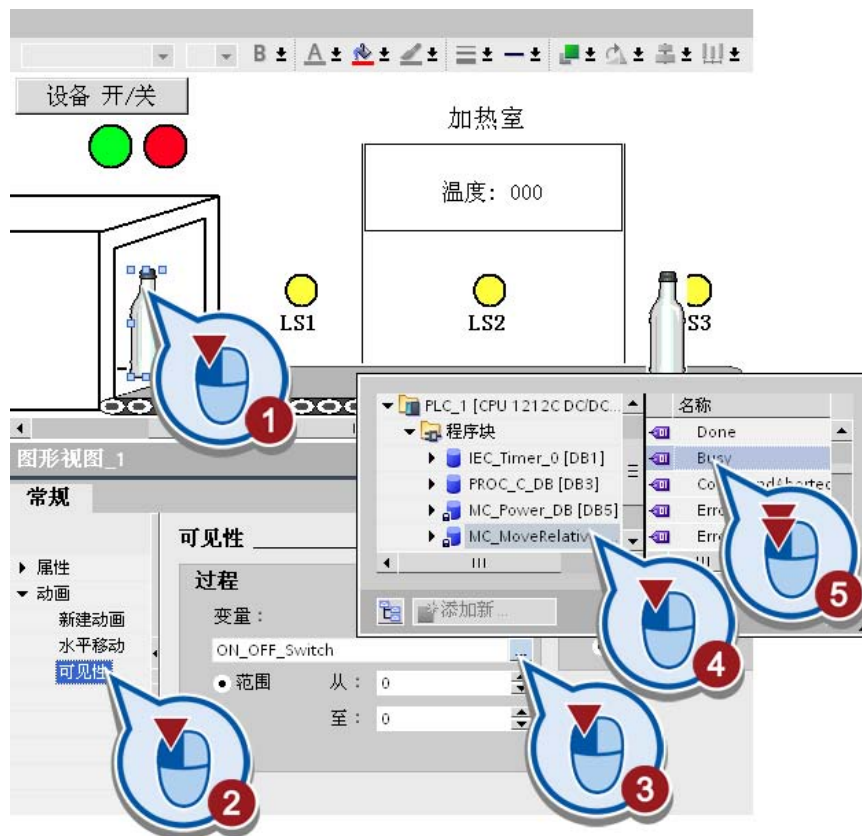
要求

- 已在组织块“Main [OB1]”中插入“MC_MoveRelative”指令。
- HMI 画面处于打开状态。
- 已为第二个瓶子创建运动动画。

步骤

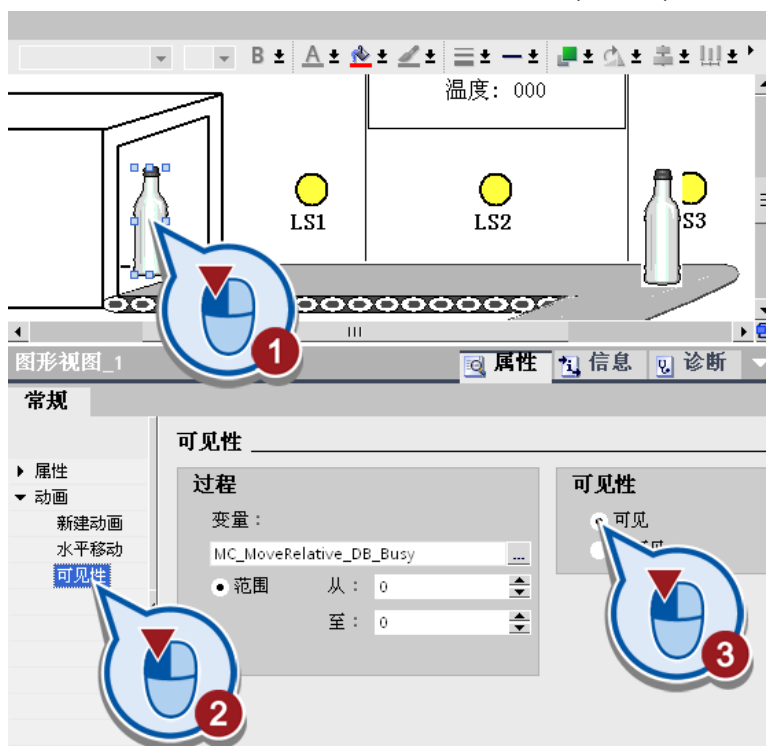
要组态瓶子在 HMI 画面中的可见性，请按以下步骤操作：

1. 将第一个瓶子的“可见性”(Visibility) 动画与数据块“MC_Move Relative”的“Busy”参数链接。



2. 将第二个瓶子的“可见性”(Visibility) 动画与数据块“MC_Move Relative”的“Busy”参数链接。

3. 将第一个瓶子的可见范围“0”到“0”更改为“可见”(Visible)。



保留第二个瓶子的可见范围设置“0”到“0”，即“不可见”(Invisible)。

4. 单击工具栏上的“保存”(Save) 按钮以保存该项目。

结果

已根据“MC_MoveRelative”指令中“Busy”参数的信号状态设置了瓶子在 HMI 画面中的可见性。

执行指令“MC_MoveRelative”时，第二条传送带开始运动。此时，指令中“Busy”参数的信号状态为“1”并且对 HMI 画面产生以下影响：

- 第一个瓶子变为不可见。
- 第二个瓶子变为可见，并从第三个光栅的位置移动到第二条传送带。

如果没有执行指令“MC_MoveRelative”，则该指令中“Busy”参数的信号状态为“0”并且对 HMI 画面产生以下影响：

- 第一个瓶子变为可见，并从第一个光栅的位置移动到第三个光栅的位置。
- 第二个瓶子不再可见。

5.7 仿真 HMI 画面

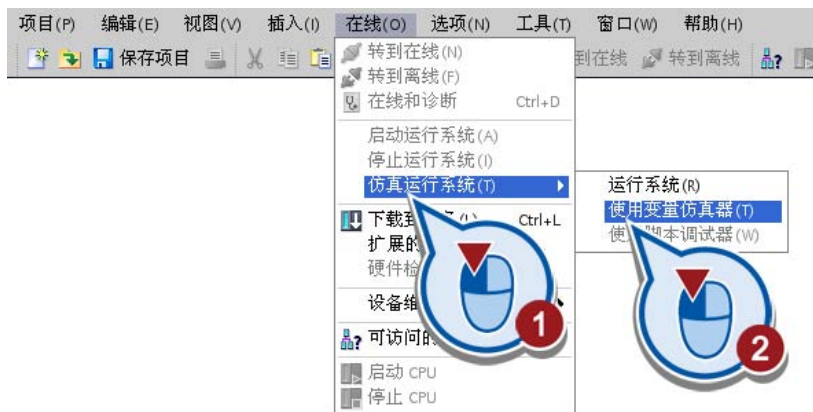
简介

以下步骤介绍如何使用运行系统仿真器测试所创建的 HMI 画面。使用运行系统仿真器可仿真光栅“LS3”对应的 PLC 输入的激活。

步骤

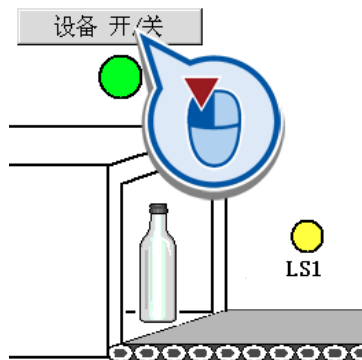
要启动已创建的 HMI 画面的仿真，请按以下步骤进行操作：

1. 打开 HMI 画面。
2. 通过菜单栏启动运行系统仿真。



随即启动运行系统的仿真。启动仿真后，“运行系统仿真器”(RT Simulator) 窗口中将显示 HMI 画面，同时红色 LED 灯会闪烁（机器关闭）。

3. 启动机器。



绿色 LED 灯会闪烁。仍可以看到第一个瓶子，这是因为此动画当前采用自动存储为 HMI 变量的“MC_MoveRelative_DB_Busy”参数的位值。

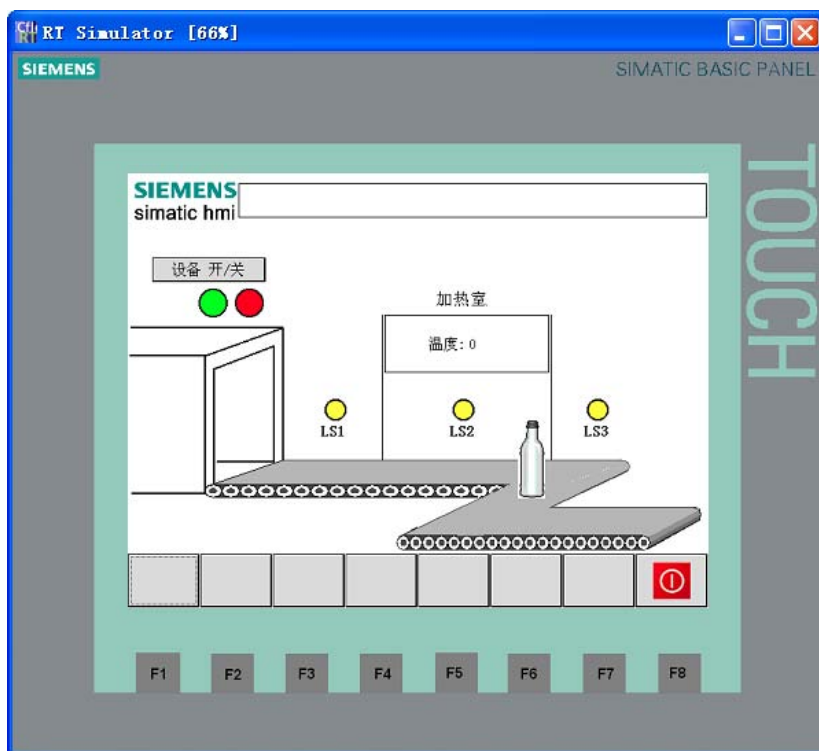
- 将 HMI 变量“MC_MoveRelative_DB_Busy”的值设置为“1”。



- 切换到“运行系统仿真器”(RT Simulator) 窗口。

结果

仿真了变量“MC_MoveRelative_DB_Busy”的值。瓶子在传送带上向光栅“LS3”的位置进行移动。



5.8 启动诊断视图

简介

以下步骤介绍了如何启动工艺对象“轴”的诊断视图。可以使用诊断功能监视电机轴的运动作业和最重要的状态及错误消息。

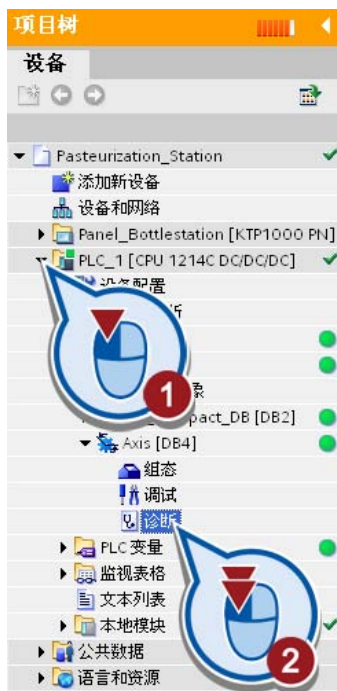
要求

- 已创建工艺对象“轴”并已正确组态。
- 已在组织块“Main [OB1]”中创建了运动控制指令“MC_Power”。
- 已在组织块“Main [OB1]”中创建了运动控制指令“MC_MoveRelativ”。

步骤

要启动诊断功能，请按以下步骤操作：

1. 将程序加载到 PLC 中并激活在线连接。更多信息，请参见“将程序加载到目标系统 (页 52)”部分。
2. 打开工艺对象“轴”的诊断窗口。

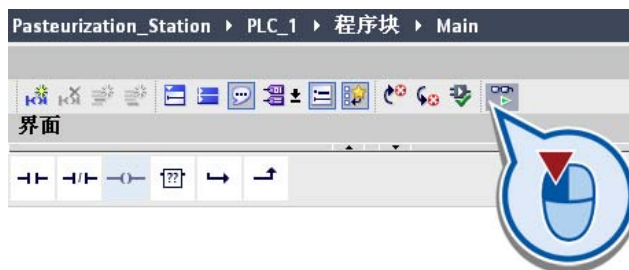


诊断窗口随即打开。

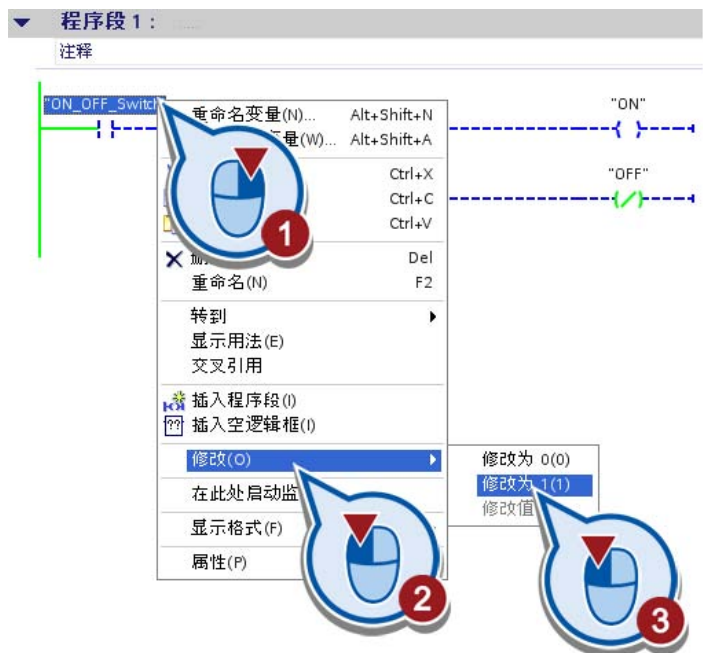


3. 打开组织块“Main [OB1]”。

4. 单击程序编辑器中工具栏上的“启用/禁用监视”(Monitoring on/off) 按钮。



5. 在程序段 1 中，将变量“ON_OFF_Switch”修改为“1”。



在程序段 1 中，将变量“ON_OFF_Switch”的信号状态设置为“1”。电流通过常开触点流到程序段末尾的线圈。置位变量“ON”，从而启动实例机器。

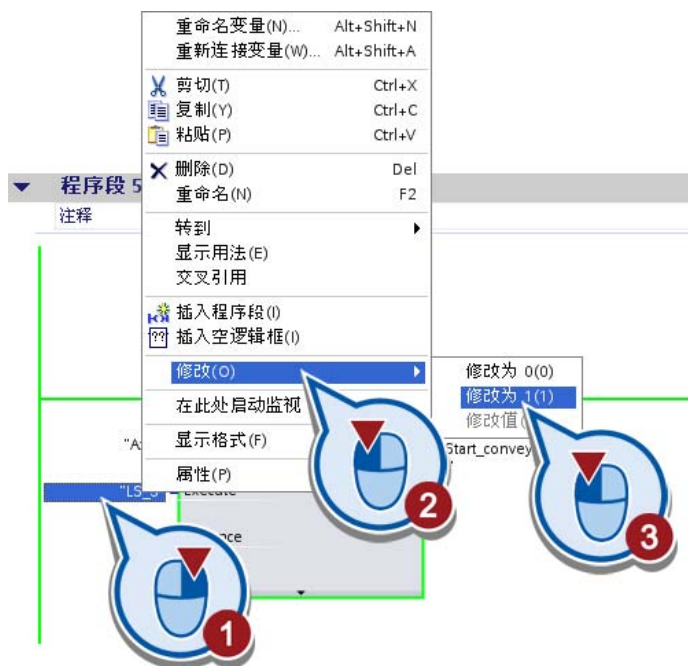
在程序段 4 中，执行指令“MC_Power”，启动传送带的电机轴。

6. 切换到工艺对象“轴”的诊断。

将在诊断窗口的“状态”(Status) 区域显示轴的启用情况。



7. 在程序段 5 中，将变量“LS_3”修改为“1”。



执行指令“MC_MoveRelative”，驱动传送带的电机轴开始运动。

8. 切换到工艺对象“轴”的诊断。

将在诊断窗口的“运动”(Motion) 区域显示电机轴正恒速转动。



9. 显示运动状态。



将在“当前位置”(Current position) 域显示电机轴的当前位置。到达距起始位置 500 mm 的目标位置时，轴会设置为停止。

结果

使用诊断功能对电机轴是否正常运行进行了检查。

词汇表

CPU

在自动化系统的中央处理单元 (CPU, Central Processing Unit) 中存储和执行用户程序。它包含操作系统、处理单元和通信接口。

CPU 操作系统

操作系统将与特定控制任务不相关的所有 CPU 功能和流程组织在一起。

HMI 设备

用于显示状态、过程进展和操作用户程序的屏幕设备。

I/O 域

I/O 域指用来显示和更改变量值的输入和输出域。

PID 控制器

PID 控制器在控制回路中连续检测受控变量的实际测量值，并将其与期望的设定值进行比较。PID 使用所生成的控制偏差计算控制器输出，以便尽可能快速平稳地将受控变量（即，要控制的变量）调整到设定值。

PLC

可编程逻辑控制器 (PLC, Programmable Logic Controller) 是指其功能作为程序存储在控制设备中的电子控制器。因此，设备的安装和接线与 PLC 的功能无关。PLC 至少由一个电源模块、一个 CPU 以及输入和输出模块构成。

PLC 变量表

用于定义在整个 CPU 中都有效的变量的表格。

PTO

“脉冲串输出”的缩写。有些 CPU，如 S7-1200，可通过输出生成快速脉冲串，这些脉冲串可用于控制运动控制操作。

位存储区

CPU 系统存储器中的存储区。可以进行写入和读取（以位、字节、字和双字的形式）。用户使用位存储器来保存中间结果。

修改变量

使用“修改变量”功能，可在执行用户程序期间在指定点修改用户程序变量，并为单个变量分配确定的值。

功能块 (FB)

根据 IEC 1131-3，功能块是指带有静态数据的代码块。可以通过功能块在用户程序中传递参数。因此，功能块非常适合于对反复出现的复杂功能（如闭环控制、工作模式选择）进行编程。因为功能块带有存储器（背景数据块），所以可随时在用户程序中的任何点访问其参数。

功能框

功能框是具有复杂功能的程序元素。但空功能框除外。可以使用空功能框作为占位符，在其中可以选择所需的运算。

变量

变量由地址和符号名构成，通常会在项目中多次使用。地址（例如，输入或位存储区的地址）用来与自动化系统进行通信。变量用于集中执行地址更改（例如，为输入更改地址），而不是在整个用户程序中更改。

地址

CPU 的输入、输出或位存储区中特定地址的名称。

块

在独立的部分中构建用户程序。部分用户程序可划分为块，这些块可在各个位置重复使用，或者使用户程序的结构更加直观。

块参数

可多次使用并在调用相关块时为其提供当前值的块内占位符。

子网

子网包括所有未通过网关互连的网络设备。它可以包含中继器。

寻址

在用户程序中分配地址。可为地址分配特定操作数或操作数范围。实例：输入 I12.1；存储器字 MW25。

库

可多次使用的元素集合。

循环中断

循环中断 OB 以周期性时间间隔启动程序，与循环程序执行无关。循环中断 OB 的启动时间通过时间基数和相位偏移量来指定。

循环时间

循环时间是 CPU 执行一次用户程序所需的时间。

控制

控制是指结果（输出变量）通过反馈回路影响受控变量的过程。

数据块 (DB)

用户程序中用于存储值或字符串的块。数据块分为可由所有代码块访问的全局数据块，以及分配给特定功能块调用的背景数据块。

数据类型

指定在用户程序中如何使用变量值或常量。例如，BOOL 数据类型的变量只使用值 1 或 0。

监视表格

监视表格的作用是集中用户程序中要监视、修改和/或强制的变量。

目标系统

在其中运行用户程序的自动化系统。

程序

程序可解决独立的控制任务。可将其分配给可编程模块，并在较小的单元（例如，块）中进行构建。

程序段

块的程序分为若干程序段。程序段可用来构建程序。

线圈

可以使用线圈修改二进制操作数。线圈可根据逻辑运算结果的信号状态置位或复位二进制操作数。

组态

“组态”意指在设备或网络视图中对设备和模块进行布置、设置和联网。机架用符号表示。和“实际的”机架一样，可以在其上插入规定数量的模块。

组织块

组织块 (OB) 是 CPU 操作系统与用户程序之间的接口。在组织块中指定用户程序的执行顺序。

编程设备

专为工业应用设计的紧凑型个人计算机。编程设备 (PG) 完全是为针对 SIMATIC 自动化系统进行编程而配备的。

编程语言

编程语言可用于创建用户程序并以图形或文本指令的形式提供用于创建用户程序的规定语言子集。用户使用编辑器输入这些指令，并将其编译为可执行用户程序。

背景数据块

背景数据块存储功能块的形参和静态数据。可以将背景数据块分配给功能块调用或功能块的调用层级。

自动化系统

自动化系统是由中央控制器 (CPU) 和各种输入/输出模块组成的可编程逻辑控制器 (PLC, Programmable Logic Controller)。

触点

可以使用触点创建或中断两个元素之间的载流连接。电流从左向右传递。可以使用触点查询操作数的信号状态或值，并根据电流的结果对其进行控制。

输入

CPU 系统存储器内的存储区（过程映像输入）或与输入模块的连接。

输出

CPU 系统存储器内的存储区（输出的过程映像）或与输出模块的连接。

过程映像

将数字量输入和输出模块的信号状态存储到 CPU 上的过程映像中。它分为过程映像输入 (PII, Process Image Input) 和过程映像输出 (PIQ, Process Image Output)。

在执行用户程序和读取过程映像输入之前，操作系统将过程映像输出 (PIQ) 传送到输出模块。

在启动用户程序之前，操作系统从输入模块中读取过程映像输入 (PII)。

运动控制

用于驱动控制的软件组件。运动控制指令控制工艺对象“轴”。工艺对象“轴”对应 PLC 中的轴，适用于通过脉冲接口激活步进电机和伺服电机。

运行系统

运行系统软件在过程模式下执行项目，可用于对过程进行操作和监视。

