

# SIEMENS

## SIMATIC

## S7-1500

### 入门指南

欢迎

---

自动化任务

---

1

硬件部分

---

2

软件部分

---

3

安全性


---

4

## 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。

 <b>危险</b>
表示如果不采取相应的小心措施， <b>将会</b> 导致死亡或者严重的人身伤害。
 <b>警告</b>
表示如果不采取相应的小心措施， <b>可能</b> 导致死亡或者严重的人身伤害。
 <b>小心</b>
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。
<b>注意</b>
表示如果不采取相应的小心措施，可能导致财产损失。

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：

 <b>警告</b>
<b>Siemens</b> 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 <b>Siemens</b> 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。本印刷品中的其他符号可能是一些其他商标。若第三方出于自身目的使用这些商标，将侵害其所有者的权利。

### 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。

# 欢迎

## 欢迎

欢迎使用“TIA Portal V13”入门指南。

在本“入门指南”中，提供了一个示例，用于说明如何通过 TIA Portal 使用 CPU SIMATIC S7-1500 来创建“颜色混合设备”的自动化解决方案。

视频文件将说明如何创建自动化任务的解决方案。

在第一部分中，将对硬件进行装配并准备对 PC 进行组态。

在第二部分中，将通过颜色混合设备示例来组态 CPU 和 HMI 可视化。

此外，您还可以找到自动化解决方案的选项及扩展。

## 长版

西门子为其产品及解决方案提供工业安全功能，以支持工厂、解决方案、机器、设备和/或网络的安全运行。这些功能是整个工业安全机制的重要组成部分。

有鉴于此，西门子不断对产品和解决方案进行开发和完善。

西门子强烈建议您定期检查产品的更新和升级信息。

要确保西门子产品和解决方案的安全操作，还须采取适当的预防措施（例如：设备单元保护机制），并将每个组件纳入全面且先进的工业安全保护机制中。

此外，还需考虑到可能使用的所有第三方产品。更多有关工业安全的信息，请访问 Internet (<http://www.siemens.com/industrialsecurity>)。

要及时了解有关产品的更新和升级信息，请订阅相关产品的实事信息。

更多相关信息，请访问 Internet (<http://support.automation.siemens.com>)。

# 目录

欢迎.....	3
<b>1 自动化任务 .....</b>	<b>7</b>
1.1 简介.....	7
1.2 样本项目.....	9
<b>2 硬件部分.....</b>	<b>16</b>
2.1 简介.....	16
2.1.1 要求.....	16
2.1.2 更多信息.....	17
2.2 安装组合件.....	18
2.2.1 概述.....	18
2.2.2 安装组合件.....	18
2.3 接线.....	23
2.3.1 概述.....	23
2.3.2 接线规则.....	24
2.3.3 对电源连接插头接线.....	25
2.3.4 将负载电流电源 (PM) 接线到 CPU.....	27
2.3.5 电位桥电路.....	29
2.3.6 对数字量输入模块接线.....	29
2.3.7 对数字量输出模块接线.....	32
2.3.8 为前连接器接线.....	34
2.4 通电.....	36
2.4.1 概述.....	36
2.4.2 通电.....	36
2.4.3 通过显示屏分配 IP 地址.....	38
<b>3 软件部分.....</b>	<b>39</b>
3.1 创建项目和硬件.....	39
3.1.1 TIA Portal 简介.....	39
3.1.2 创建项目.....	40
3.1.3 创建 S7-1500 CPU.....	43
3.1.4 运行硬件检测.....	46
3.1.5 创建 ET 200 接口模块.....	48
3.1.6 联网 ET 200 接口模块.....	50
3.1.7 为 ET 200SP 创建输入和输出模块及服务模块.....	51
3.1.8 为 ET 200MP 创建输入和输出模块.....	53



3.1.9	为 ET 200 指定名称 .....	54
3.2	创建程序 .....	55
3.2.1	加载块库 .....	55
3.2.2	删除程序块 Main [OB1] .....	58
3.2.3	复制程序块 .....	60
3.2.4	循环中断 OB .....	61
3.2.4.1	循环中断 OB - 循环时间和相位 .....	61
3.2.4.2	更改循环时间 .....	62
3.2.5	复制变量表 .....	63
3.2.6	编译项目 .....	64
3.2.7	将项目加载到 CPU .....	66
3.2.8	优化块访问 .....	67
3.2.8.1	简介 .....	67
3.2.8.2	扩展并重新加载经优化的“填充”数据块 .....	69
3.2.9	块的版本控制 .....	73
3.2.10	设置保持性 .....	77
3.2.11	激活 EN/ENO 机制 .....	80
3.2.12	使用注释功能 .....	82
3.2.13	局部错误处理 .....	83
3.2.13.1	在块内处理错误 .....	83
3.2.13.2	加载用于局部错误处理的块 .....	85
3.2.13.3	不进行局部错误处理时生成错误 .....	87
3.2.13.4	进行局部错误处理时生成错误 .....	88
3.3	组态可视化 .....	73
3.3.1	现有样本项目 .....	90
3.3.2	HMI 组态 .....	90
3.3.2.1	概述 .....	90
3.3.2.2	SIMATIC HMI 精智面板 .....	91
3.3.2.3	HMI 画面 .....	92
3.3.2.4	其它控制元件 .....	93
3.3.2.5	配方 .....	94
3.3.2.6	归档 .....	95
3.3.2.7	用户自定义函数 .....	96
3.3.2.8	用户管理 .....	98
3.3.2.9	多语言性 .....	99
3.3.2.10	报表 .....	101
3.3.3	插入库中的 HMI 设备 .....	103
3.3.3.1	存储库中的对象 .....	103
3.3.4	组态 HMI 连接 .....	104
3.3.4.1	设备间的通信 .....	104
3.3.4.2	组态 HMI 连接 .....	105
3.3.4.3	连接 HMI 变量 .....	107

3.3.5	组态系统诊断 .....	109
3.3.5.1	系统诊断的基础知识 .....	109
3.3.5.2	系统诊断视图 .....	110
3.3.5.3	组态系统诊断视图 .....	113
3.3.6	模拟 HMI 设备 .....	115
3.3.6.1	仿真基本知识 .....	115
3.3.6.2	以仿真状态运行设备 .....	116
3.4	将项目加载到编程设备 .....	120
3.4.1	将 CPU 加载到项目 .....	120
3.5	通过“项目间工程组态”进行团队工程组态 .....	122
3.5.1	“项目间工程组态”基本知识 .....	122
3.5.2	创建 IPE 文件 .....	123
3.5.3	导入 IPE 文件 .....	124
<b>4</b>	<b>安全性 .....</b>	<b>127</b>
4.1	CPU 保护功能的概述 .....	127
4.2	使用显示屏组态其它访问保护 .....	128
4.3	专有技术保护 .....	129
4.4	防拷贝保护 .....	132
4.5	通过锁定 CPU 进行保护 .....	133
4.6	组态 CPU 的访问保护 .....	134
4.7	组态 HMI 连接保护 .....	137
	索引 .....	138

# 自动化任务

## 1.1 简介

### 简介

在以下部分中将介绍自动化任务。

您将了解有关应用示例、硬件配置和样本项目组件的更多信息。

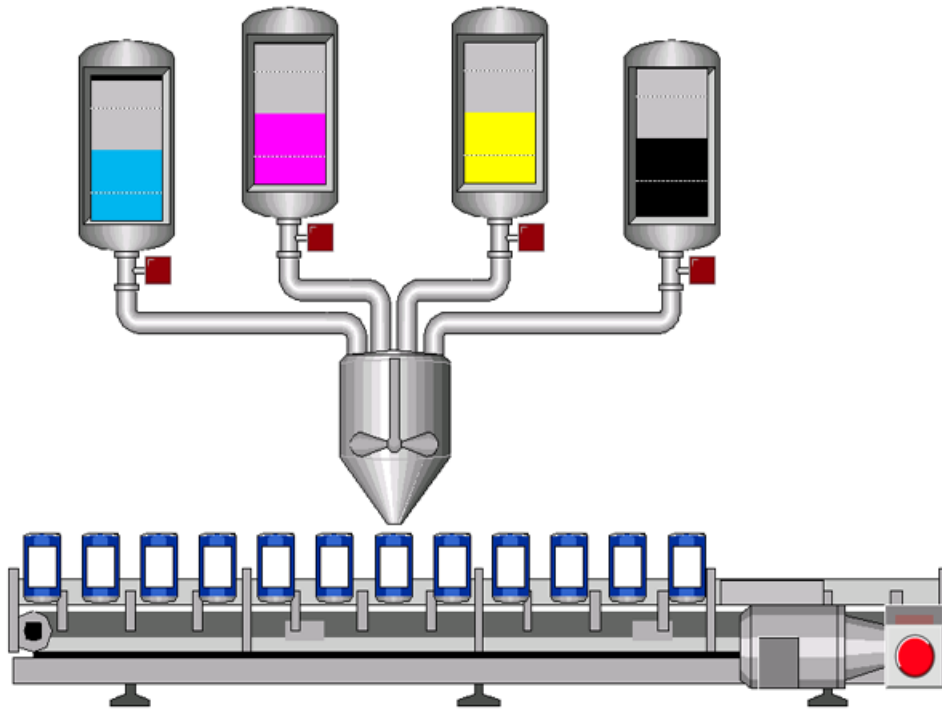
### 应用示例

本“入门指南”应用示例是混合和填充先前所选颜色配方的颜色混合系统。

配方包含四种颜色成分：青色、紫红色、黄色和黑色，即 **CMYK** 颜色空间中的颜色。

填充分四个阶段完成。

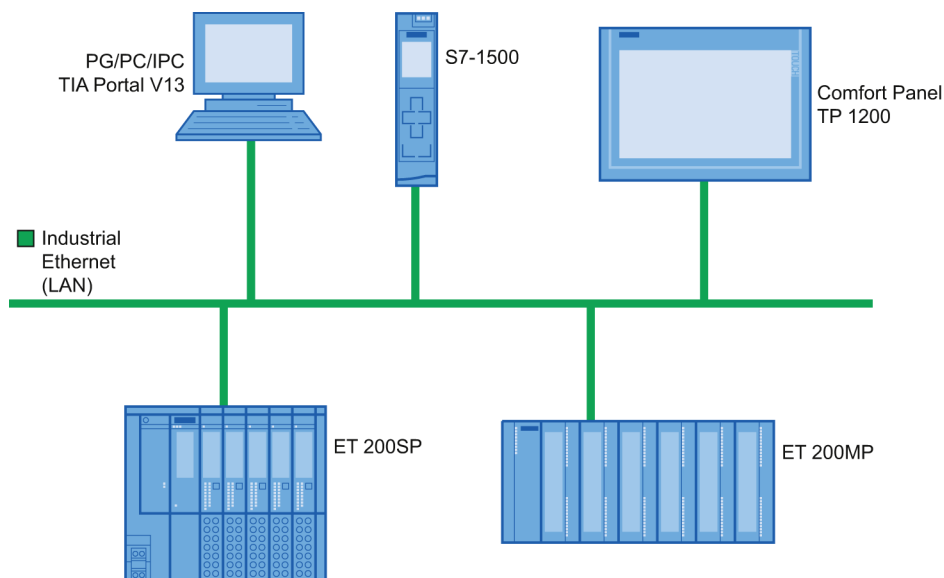
- 使用 **HMI** 配方函数选择颜色混合物。
- 通过打开各罐的阀填充配方成分或四种基本颜色。
- 混合颜色。
- 通过传送带将完成的颜色混合物填充到锡罐并进行传送。



## 硬件配置设计

硬件配置包含下列设备：

- 带有 S7-1500 负载电流源、数字量输入模块和数字量输出模块的 CPU 1511-1 PN。
- 可通过 TIA Portal 进行仿真的 HMI 面板 TP1200 精智型。
- 带有 IM 155-5 PN ST 接口模块以及数字量输入和输出模块的分布式 ET 200MP I/O 系统。
- 带有 IM 155-6 PN ST 接口模块、数字量输入模块、数字量输出模块以及服务模块的分布式 ET 200SP I/O 系统。



## 1.2 样本项目

### 应用样本项目

为了通过 TIA Portal 组态颜色混合系统，应创建样本项目“Color\_Filling\_Station”。

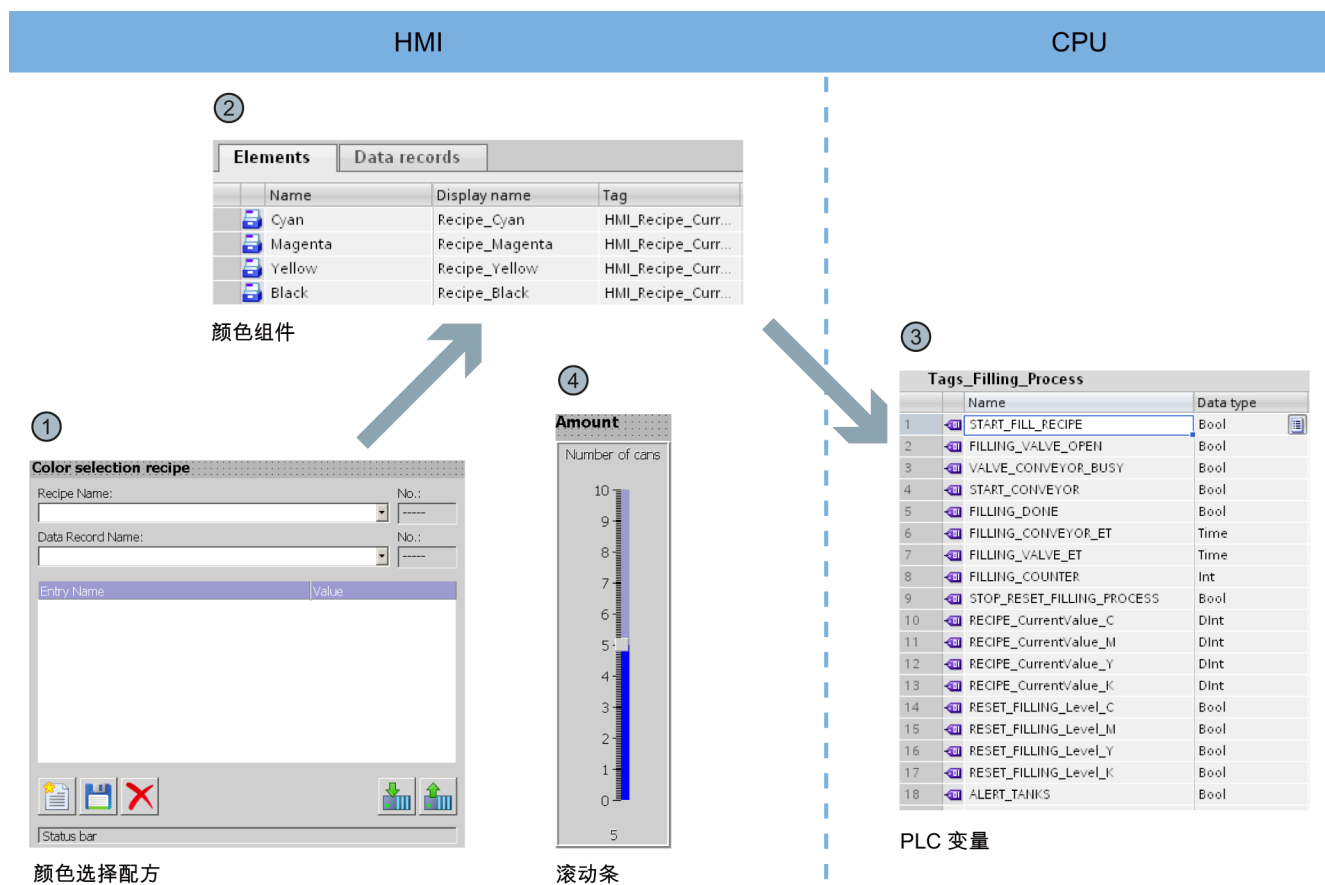
样本项目中已存在下列项目组件：

- CPU 程序块
- 精智面板上的 HMI 可视化

在本部分中，将详细说明样本项目中各项目组件间的关系。

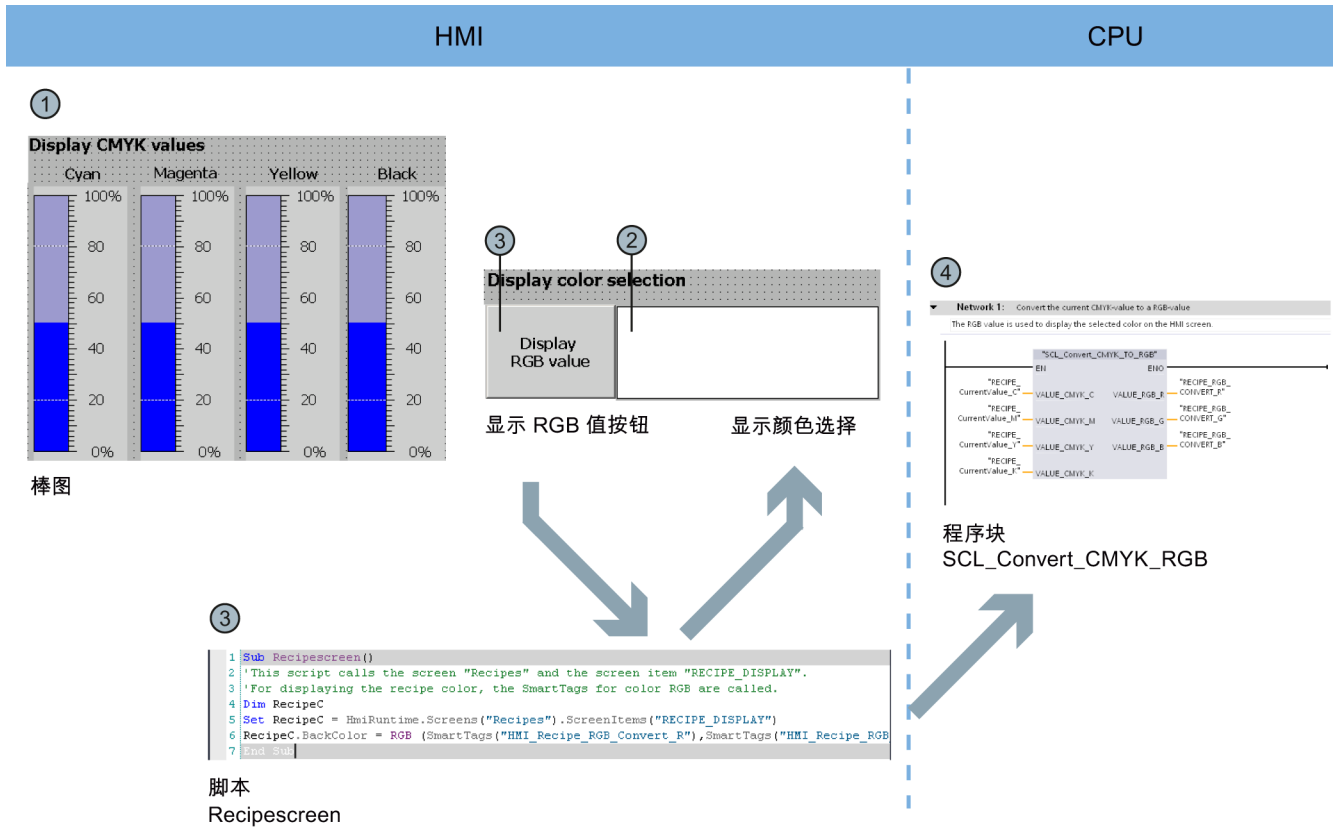
以后用户可自行执行必需的组态步骤。

选择配方



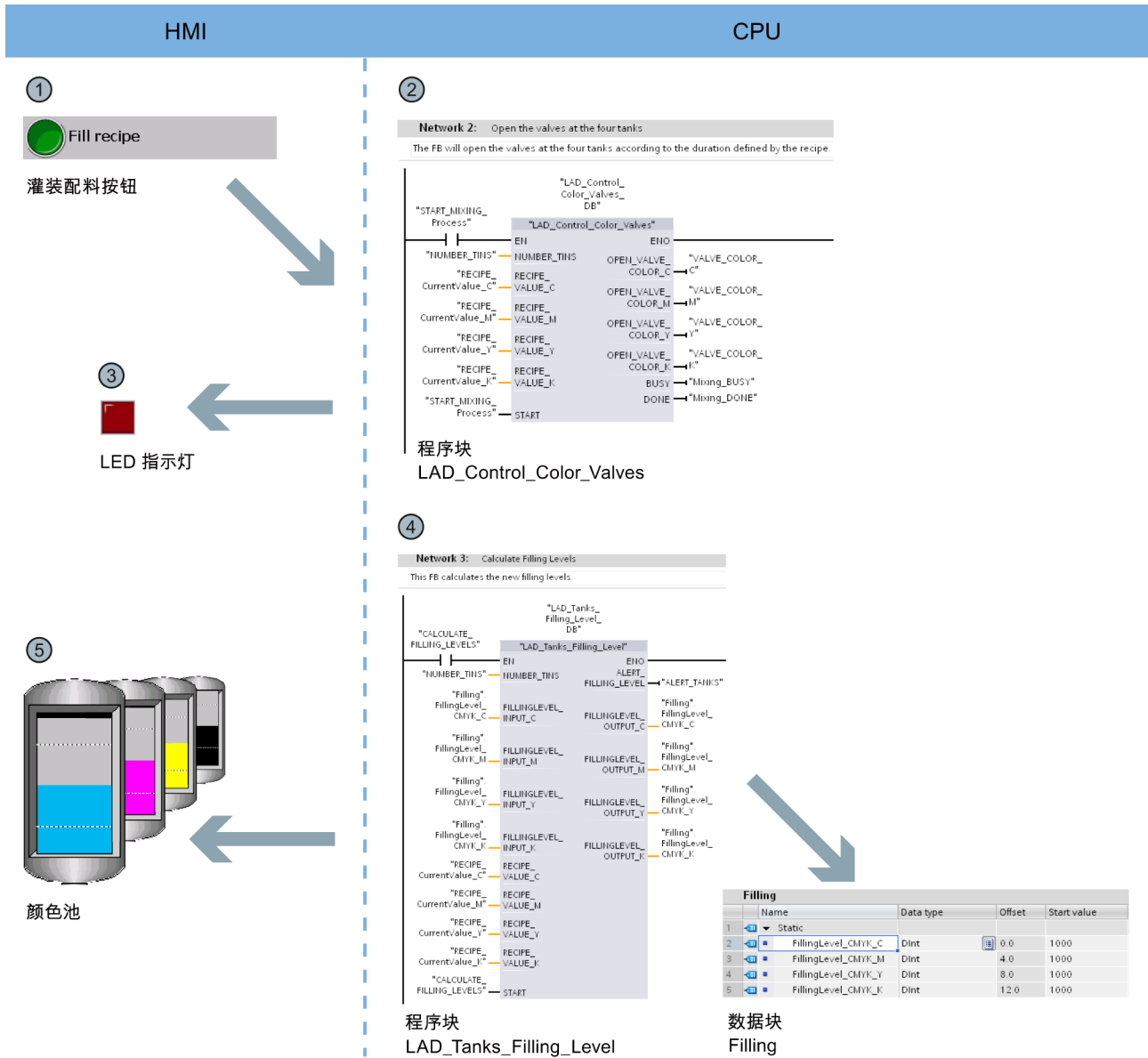
- ① HMI 画面“Recipes”包括“颜色选择配方”(Color Selection Recipe)。这是 TIA Portal 库中的预制对象。通过此对象可选择数据记录并创建新的数据记录。
- ② 数据记录（颜色混合物）和元素（颜色成分）存储在“配方”HMI 编辑器中。每种颜色混合物包含四种颜色成分：青色 (C)、洋红色 (M)、黄色 (Y) 和黑色 (K)。颜色混合物中这四种颜色成分的比例存储在“配方”编辑器中。
- ③ 各颜色成分值在装入颜色混合物时写入到 PLC 变量中。PLC 变量存储在 CPU 的“Tags\_Filling\_Process”变量表中。
- ④ HMI 画面“Recipes”还包含一个滚动条。通过此滚动条指定要填充的锡罐数量。

## 显示 CMYK 和 RGB 值



- ① 当所需颜色混合物显示在 HMI 画面“Recipes”中时，值将以棒图形式显示在 CMYK 颜色空间。
- ② 颜色混合物可以其它形式显示。这需要运行“Recipescreen”脚本。
- ③ 通过单击“显示 RGB 值”(Display RGB Value) 按钮可运行“Recipescreen”脚本。此脚本指定显示分配给 CMYK 值的 RGB 值，这是因为 CMYK 值无法直接输出到画面。
- ④ 所需 RGB 值通过“SCL\_Convert\_CMYK\_TO\_RGB”程序块来计算。

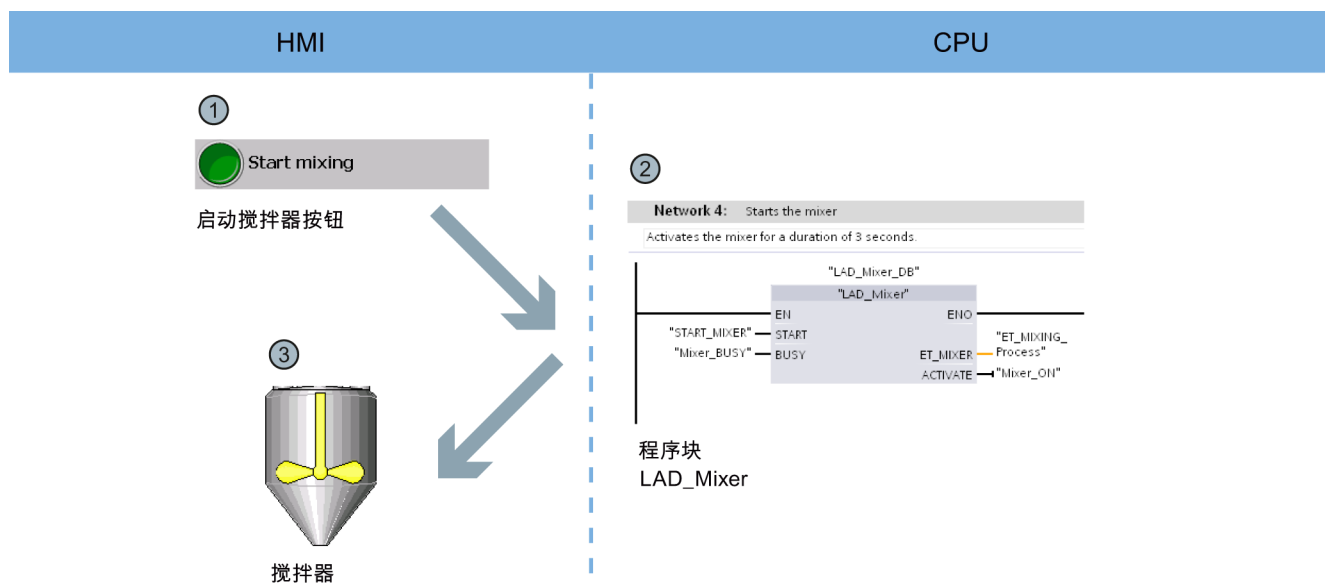
填充配方



- ① HMI 画面“Start screen”中的“填充配方”(Fill recipe) 按钮启动颜色成分填充。此按钮用于激活“LAD\_Control\_Color\_Valves”程序块。
- ② 程序块将根据指定配方和需填充的锡罐数计算，相对于颜色混合物而言这四个阀分别需要保持打开状态的时间。
- ③ 罐下的 LED 指示灯指示该阀是否处于打开状态。
- ④ “LAD\_Tanks\_Filling\_Level”程序块与填充操作同时执行。此程序块计算罐中剩余的填充量。罐中的填充量存储在全局数据块“填充”中。
- ⑤ HMI 画面中的填充量指示器直接与全局数据块相连，并在每个运行系统采集周期更新。

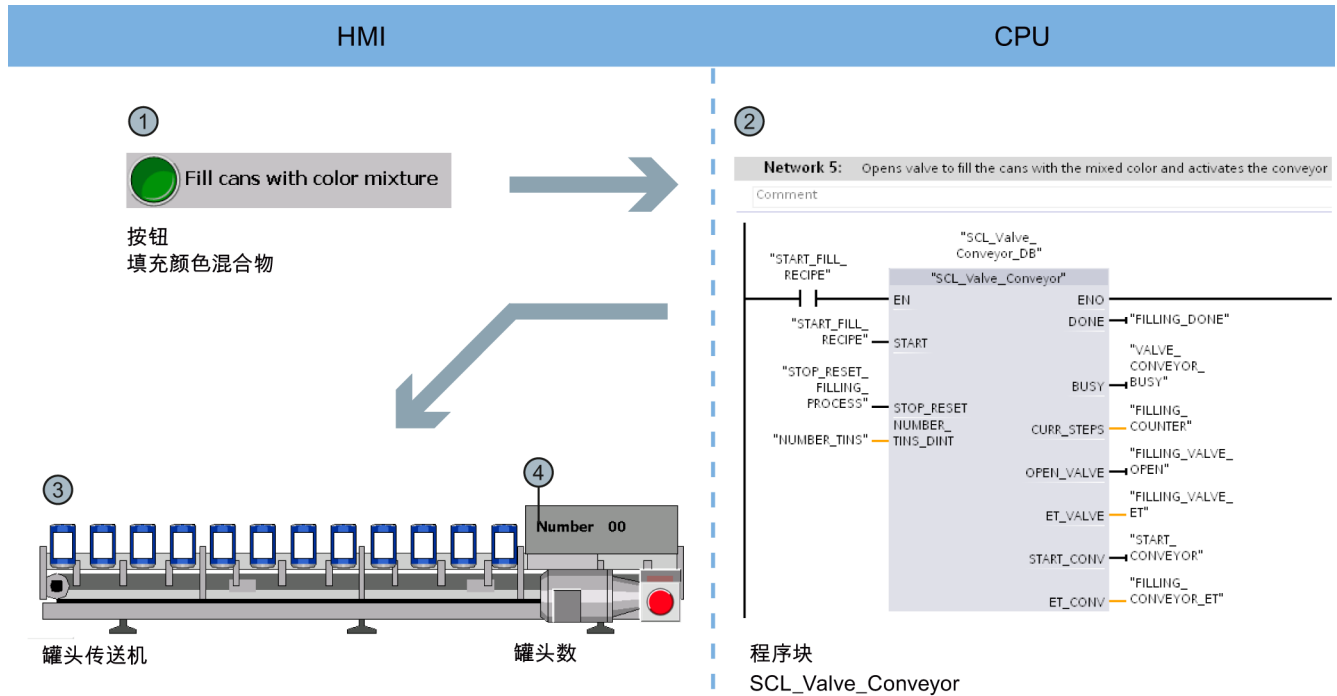


## 开始混合过程



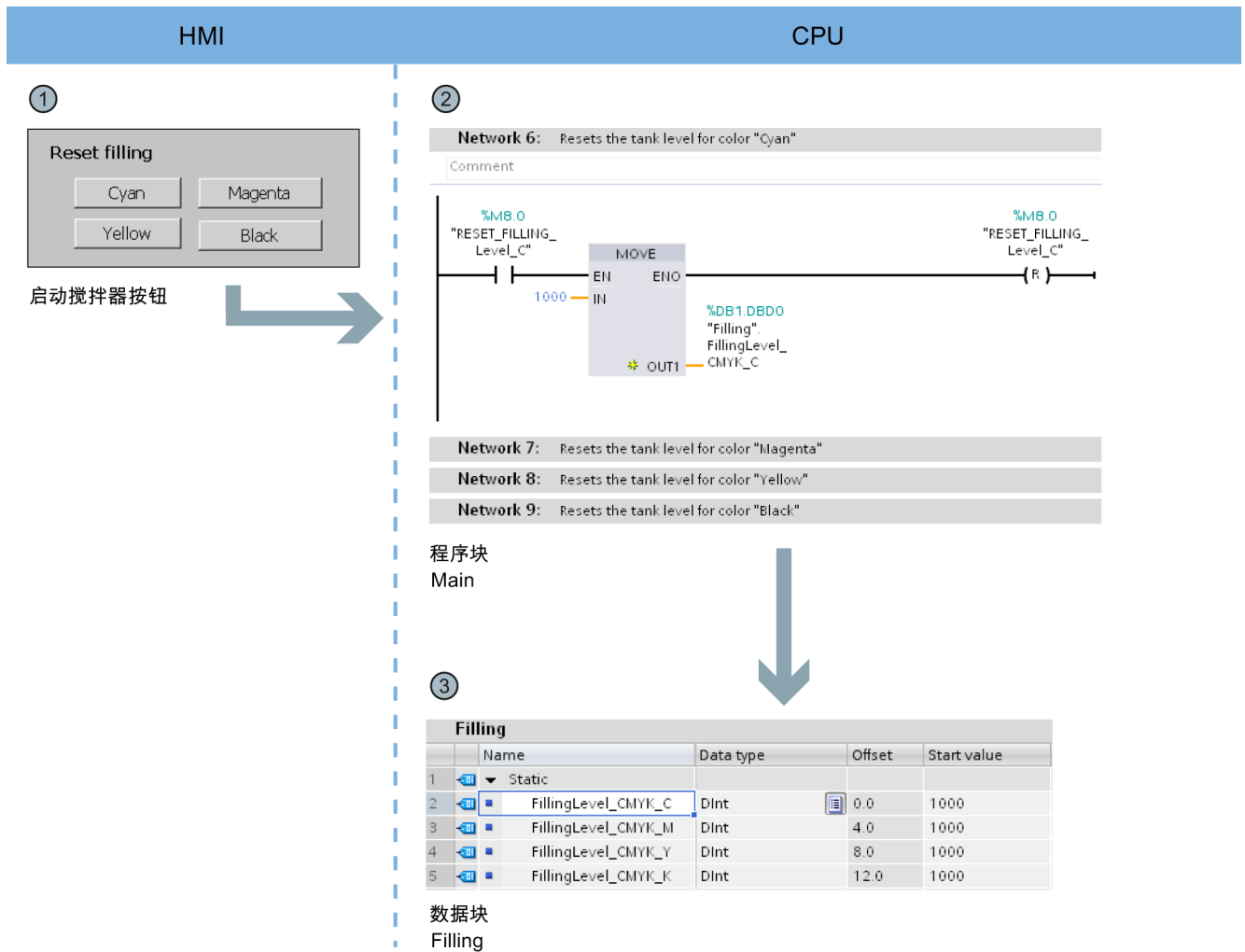
- ① HMI 画面“Start screen”中的“启动混合过程”(Start mixing process) 按钮启动颜色混合系统的搅拌机。
- ② 为此，应在 CPU 端调用“LAD\_Mixer”程序块。它将激活搅拌机 3 秒。
- ③ 通过 HMI 画面上的闪烁指示激活了搅拌机。

填充颜色混合物



- ① HMI 画面“Start screen”中的“填充颜色混合物”(Fill Color Mixture)按钮启动锡罐填充。
- ② 为此，应在 CPU 端激活“SCL\_Valve\_Conveyor”程序块；该程序块控制阀和传送带。
- ③ 此锡罐根据 HMI 画面中传送带的移动显示为动画。
- ④ 计数器指示已填充的锡罐数。

### 重置填充量



- ① HMI 画面“Start screen”中有一个用于重置四个颜色罐的填充量按钮。
- ② 在“Main”程序块的程序段 6 到 9 中已实现了各填充量的重置。
- ③ 程序段 6 至 9 将值重置为全局数据块“填充”中的起始值。

## 硬件部分

### 2.1 简介

带有 TIA Portal 的新型 SIMATIC S7-1500

控制器系列为您提供了大量的选项，用于进一步提高机器的生产力并使工程组态流程更加高效。本入门指南介绍了这些选项。

在前面的基本步骤中，您将更清楚地了解新硬件。同时还展示了如何组态和编程带有 SIMATIC STEP 7 V13 (TIA Portal) 的 SIMATIC S7-1500。通过连接安装有 SIMATIC WinCC Advanced V13 (TIA Portal) 或 SIMATIC WinCC Professional V13 (TIA Portal) 的 SIMATIC HMI 精智面板，可完成一些基本操作。

#### 2.1.1 要求

##### 硬件要求

要实现本入门指南的硬件部分，您将需要：

- 1 × CPU 1511-1 PN (6ES7511-1AK00-0AB0)
- 1 个 S7-1500 负载电流电源 PM 70W 120/230VAC (6EP1332-4BA00)
- 1 根安装导轨 (6ES7590-1AB60-0AA0)
- 1 个数字量输入模块 DI 16x24VDC SRC BA (6ES7521-1BH50-0AA0)
- 1 个数字量输出模块 DQ 16x24VDC/0.5A ST (6ES7522-1BH00-0AB0)
- 2 个前端连接器 (6ES7592-1AM00-0XB0)
- 1 个 4 MB 的 SIMATIC 存储卡（如 6ES7954-8LBxx-0AA0）
- 1 根以太网电缆

以上硬件组件包含在以下启动包中：

S7-1500 的启动包中包含有以下软件： 6ES7511-1AK00-4YB5

## 软件要求

要实现本入门指南的软件部分，您将需要：

- SIMATIC STEP 7 Professional V13
- SIMATIC WinCC Advanced V13 或 SIMATIC WinCC Professional V13



### 可能会导致严重人员伤害

根据相关的应用领域，设备或系统中的 S7-1500 自动化系统需遵守特定的标准和规定。请遵循适用的安全和事故预防规定，例如 IEC 60204-1（通用机器安全要求）。

如果不遵循这些规定，可能会造成严重的人身伤害并可能损坏机器和设施。

## 2.1.2 更多信息

下面提供了关于所使用的硬件的详细信息：

- CPU 1511-1 PN (6ES7511-1AK00-0AB0)  
(<http://support.automation.siemens.com/WW/view/zh/68020492>)
- S7-1500 负载电流电源 PM 70W 120/230VAC (6EP1332-4BA00)  
(<http://support.automation.siemens.com/WW/view/zh/68036174>)
- DI 16x24 VDC SRC BA 数字量输入模块 (6ES7521-1BH50-0AA0)  
(<http://support.automation.siemens.com/WW/view/zh/59191844/>)
- DQ 16x24 VDC/0.5A ST 数字量输出模块 (6ES7522-1BH00-0AB0)  
(<http://support.automation.siemens.com/WW/view/zh/59193401>)

## 2.2 安装组合件

### 2.2.1 概述

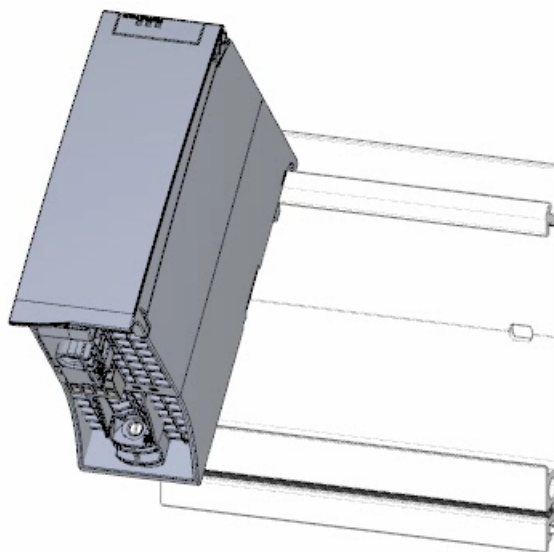
#### 安装组合件

在本部分中将安装此结构。

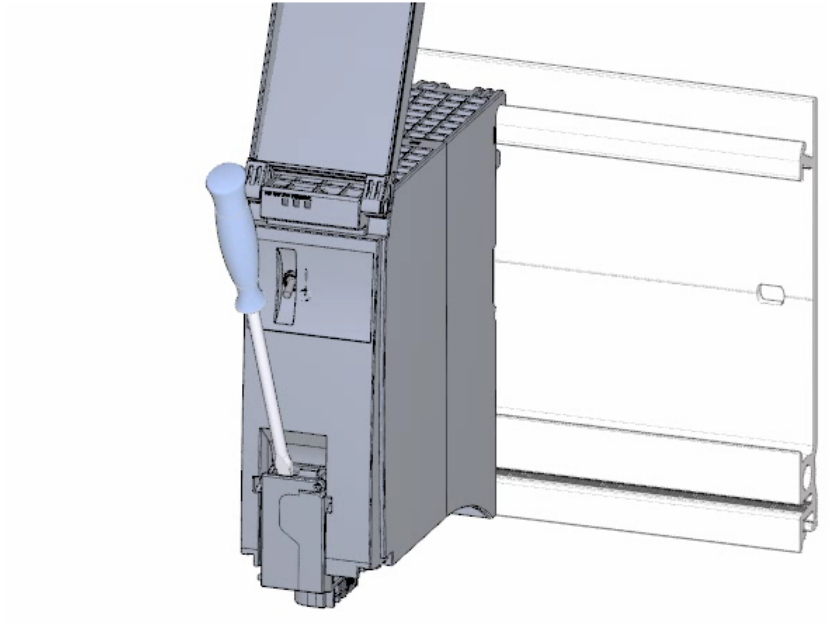
### 2.2.2 安装组合件

#### 步骤

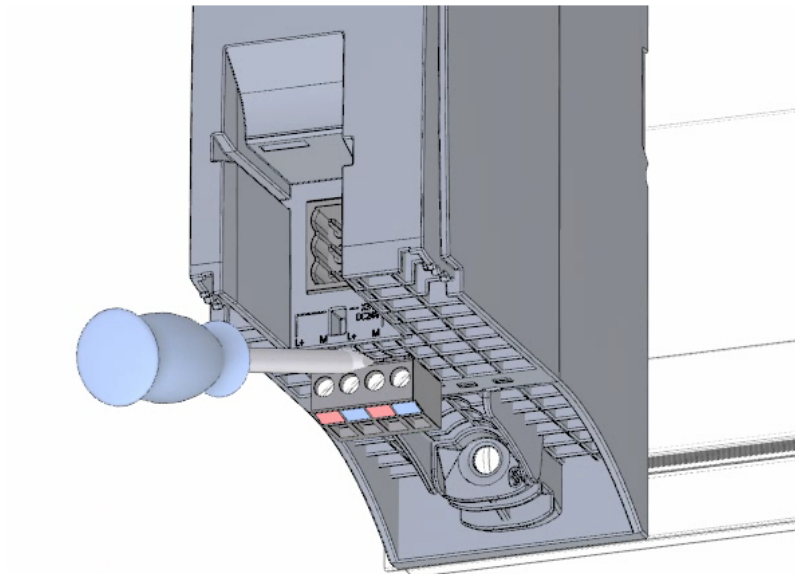
1. 在安装导轨上安装负载电流电源 (PM)。



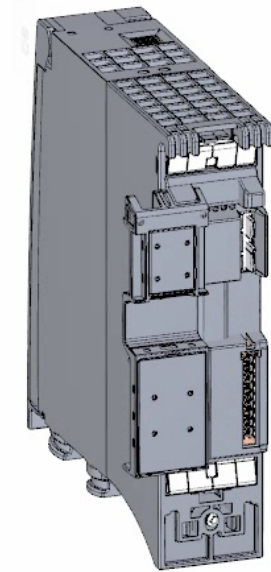
2. 打开前盖并拔出电源连接插头。



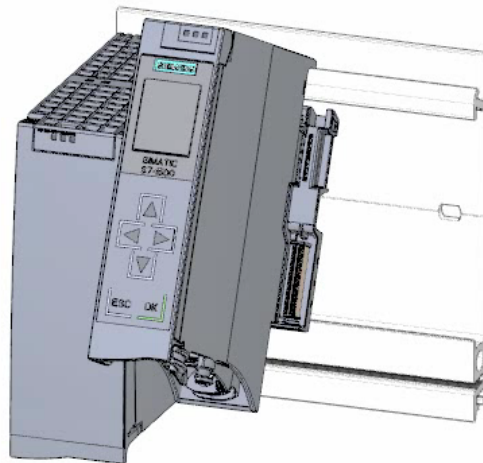
3. 拔出 4 孔连接插头并拧紧负载电流电源 (PM)。



4. 将 U 型连接器插入 CPU 后部。

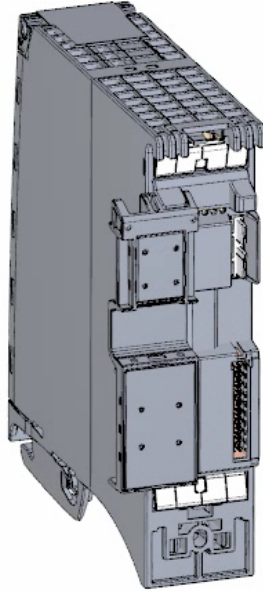


5. 在安装导轨上安装 CPU 并将其拧紧。

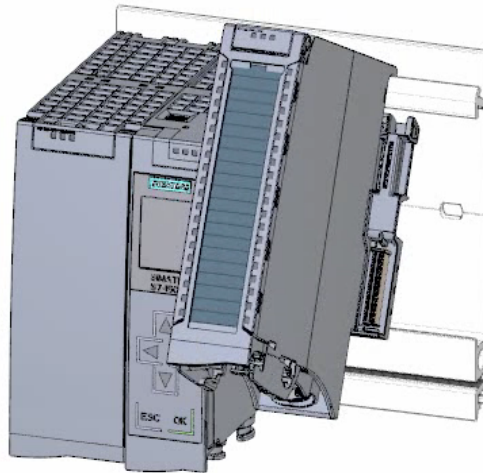




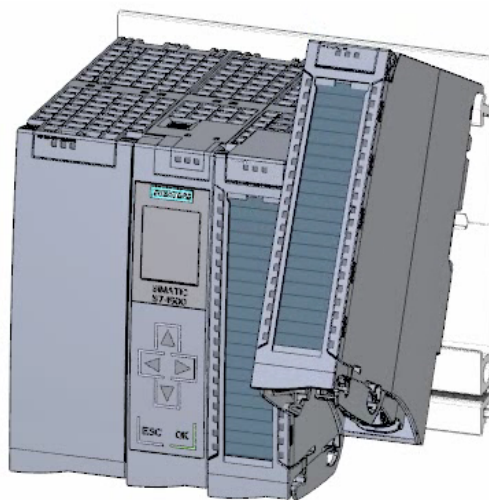
6. 将 U 型连接器插入数字量输入模块后部。



7. 将数字量输入模块连接到安装导轨并将其拧紧。



8. 将数字量输出模块连接到安装导轨并将其拧紧。



## 结果

已安装该组合件。

## 2.3 接线

### 2.3.1 概述

#### 接线组合件

在本部分中将安装此组合件。



在接线期间不能将负载电流电源线连接到电源。

### 2.3.2 接线规则

根据相关的应用领域，设备或系统中 S7-1500 CPU 的运行由特定规则 and 规定集定义。

有关 S7-1500 操作的基本规则 and 规定，请参见“S7-1500 系统说明 (<http://support.automation.siemens.com/WW/view/zh/59191792>)”。

#### CPU 的接线规则

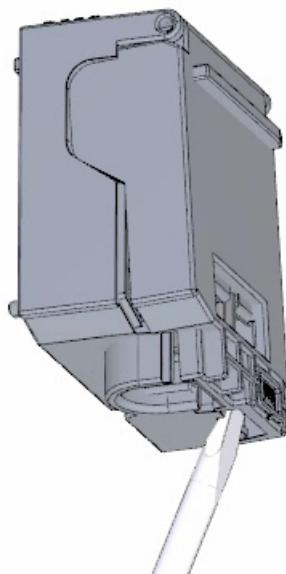
接线规则...		CPU	40 针前端连接器（螺钉 型连接）	负载电源
刚性导线的可连接导线横截面		—	最大 0.25 mm <sup>2</sup>	—
		—	AWG*: 24	—
软绞线的可连接导线横截面	不带导线端头	0.25 至 2.5 mm <sup>2</sup>	0.25 至 1.5 mm <sup>2</sup>	1.5 mm <sup>2</sup>
		AWG*: 24 到 16	AWG*: 24 到 16	AWG*: 16
	带导线端头	0.25 至 2.5 mm <sup>2</sup>	0.25 至 1.5 mm <sup>2</sup>	1.5 mm <sup>2</sup>
		AWG*: 24 到 16	AWG*: 24 到 16	AWG*: 16
每个连接的接线数		1	同一根导线端头中，1 根导线或 2 根导线组合的最大总横截面为 1.5 mm <sup>2</sup>	1
剥去外皮的导线的长度		10 至 11 mm	10 至 11 mm	7 至 8 mm
符合 DIN 46228 的末端套管	不带塑料套管	设计 A, 长 10 mm	设计 A, 长 10 mm 和 12 mm	设计 A, 长 7 mm
	带塑料套管, 0.25 至 1.5 mm <sup>2</sup>	设计 E, 长 10 mm	设计 E, 长 10 mm 和 12 mm	设计 A, 长 7 mm
护套直径		—	—	8.5 mm
工具		3 到 3.5 mm 十字螺丝刀, 锥型设计	3 到 3.5 mm 十字螺丝刀, 锥型设计	3 到 3.5 mm 十字螺丝刀, 锥型设计
连接方法		推入式端子	螺钉型端子	螺钉型端子
拧紧扭矩		—	从 0.4 Nm 到 0.7 Nm	从 0.5 Nm 到 0.6 Nm

\* AWG: 美国线缆规格

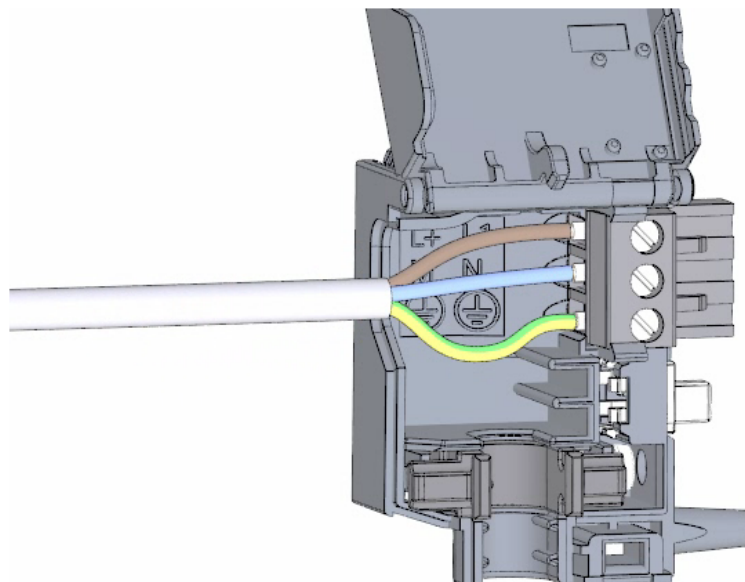
### 2.3.3 对电源连接插头接线

#### 步骤

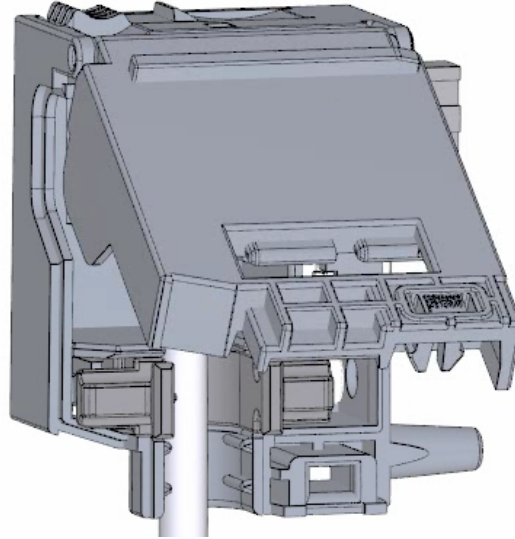
1. 使用适用工具拔出连接器外盖。



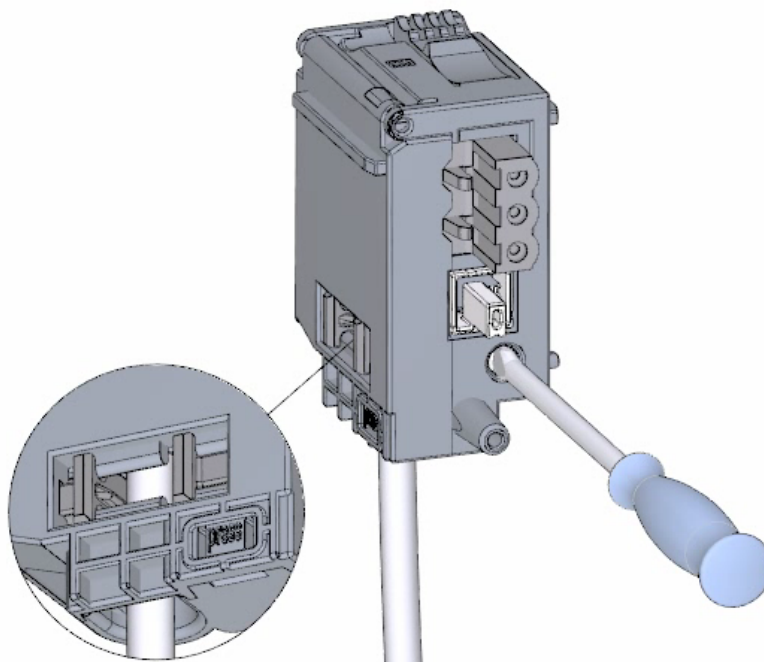
2. 根据接线图将电源线连接到插头上。  
在插头的另一侧，标有该插头认证后可使用的电压信息。  
根据插头背面的信息，通过插入编码元件选择相应电压。



3. 合上外盖。



4. 拧紧电源连接插头上的螺钉。



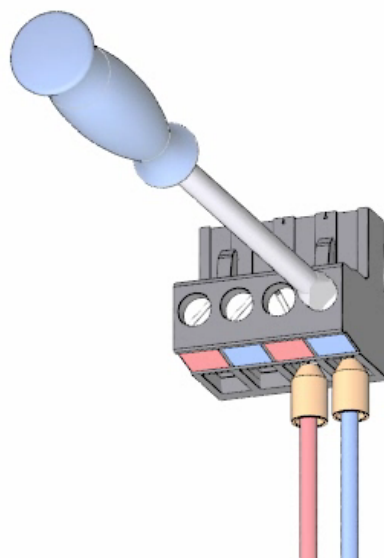
**结果**

电源连接插头现在已接线。

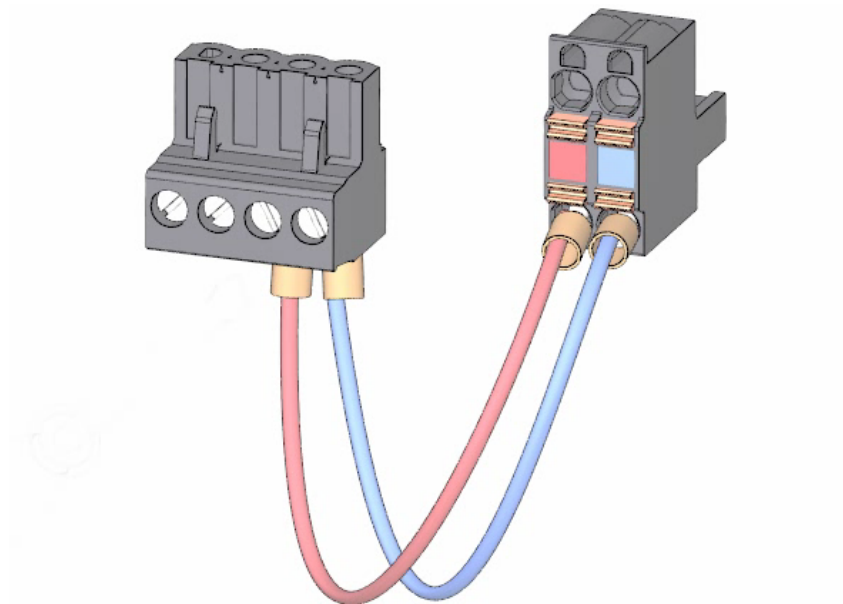
### 2.3.4 将负载电流电源 (PM) 接线到 CPU

#### 步骤

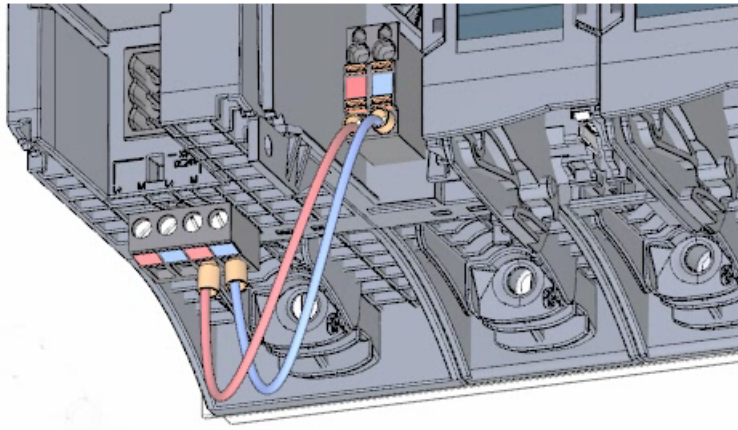
1. 对负载电流电源 (PM) 的 4 孔连接器插头接线。



2. 将 4 孔连接器插头接线到 CPU 的 4 孔电源连接插头。



3. 将负载电流电源 (PM) 连接到 CPU



结果

负载电流电源现在已接线到 CPU。



### 2.3.5 电位桥电路

#### 电位桥电路的应用

如果想要为负载组提供相同的电位（非隔离），则使用为前端连接器提供的电位电路桥。这表明您无需使用两根导线对固线装置接线。

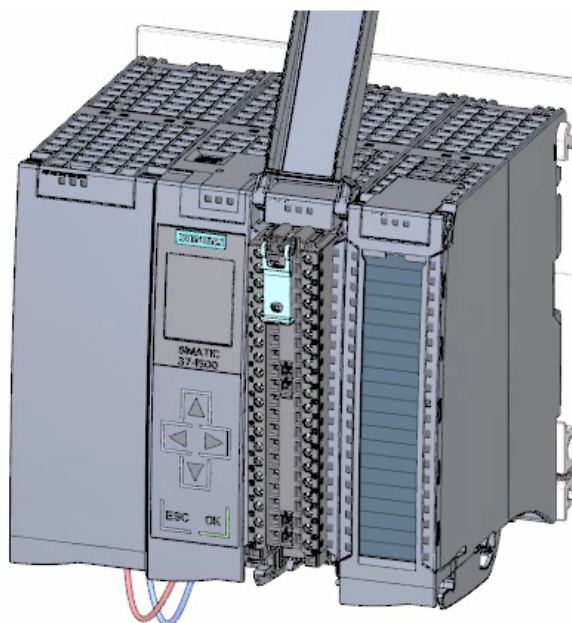
#### 提示

使用前端连接器上的端子 40 (M) 和 39 (L+) 将电位以环型连结到下一个模块。

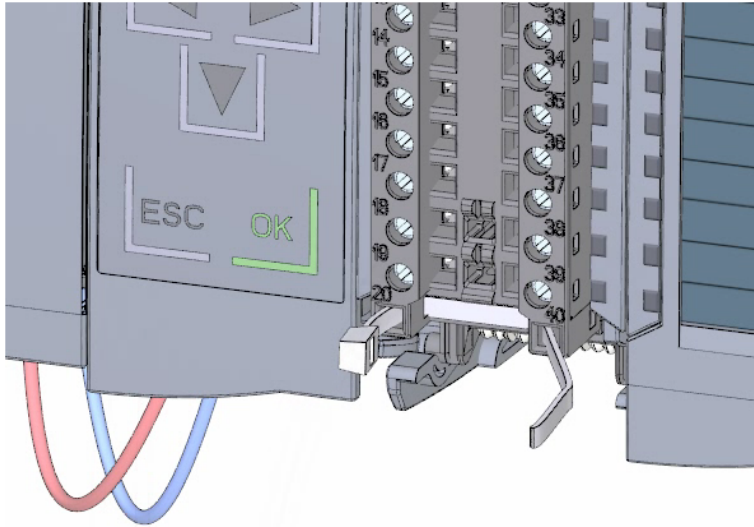
### 2.3.6 对数字量输入模块接线

#### 步骤

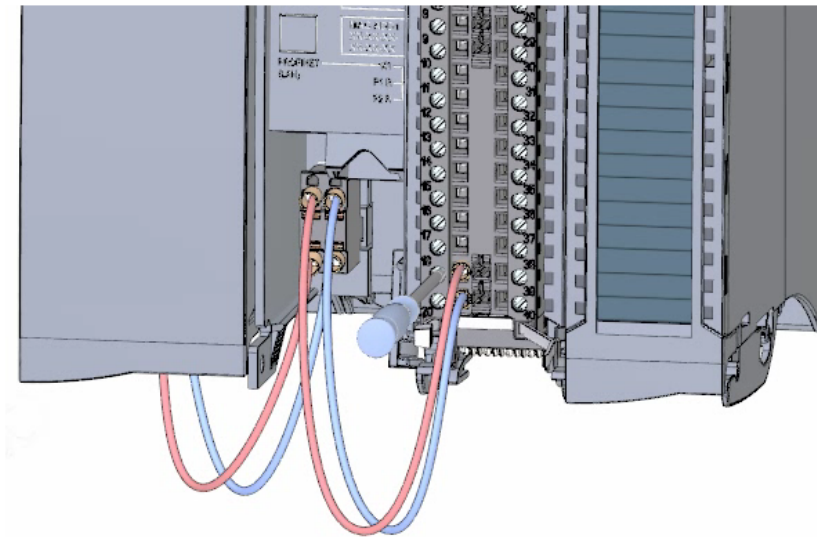
1. 将前端连接器插入预接线位置。在预接线处，前连接器与模块间未进行电气连接。



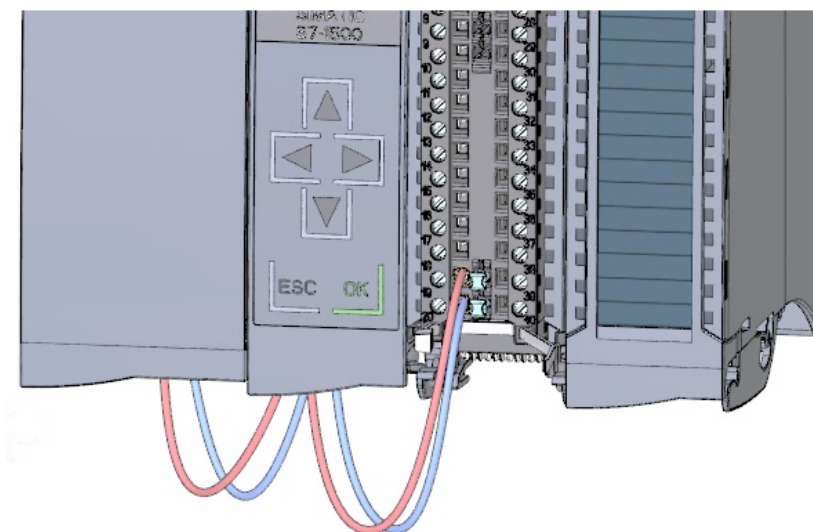
2. 用电缆扎带固定电缆。



3. 将负载电压 24 V DC 连接到端子 20 (M) 和 19 (L+)。



4. 在两个底部端子之间插入电位电路桥。



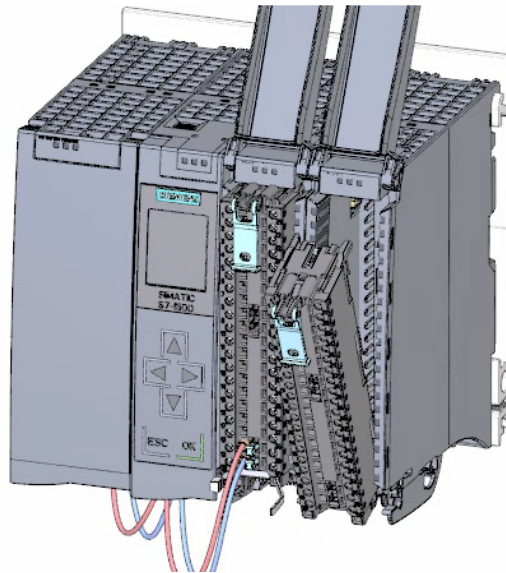
## 结果

数字量输入模块现在已经接线。

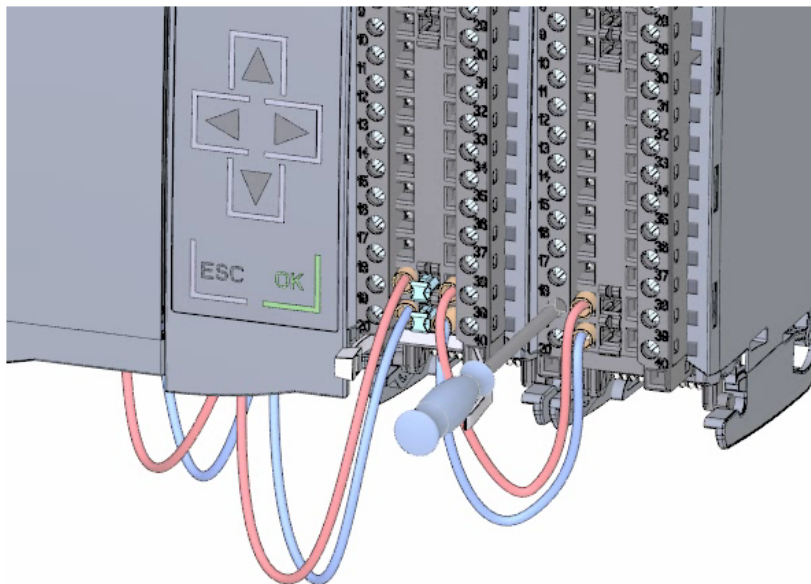
### 2.3.7 对数字量输出模块接线

#### 步骤

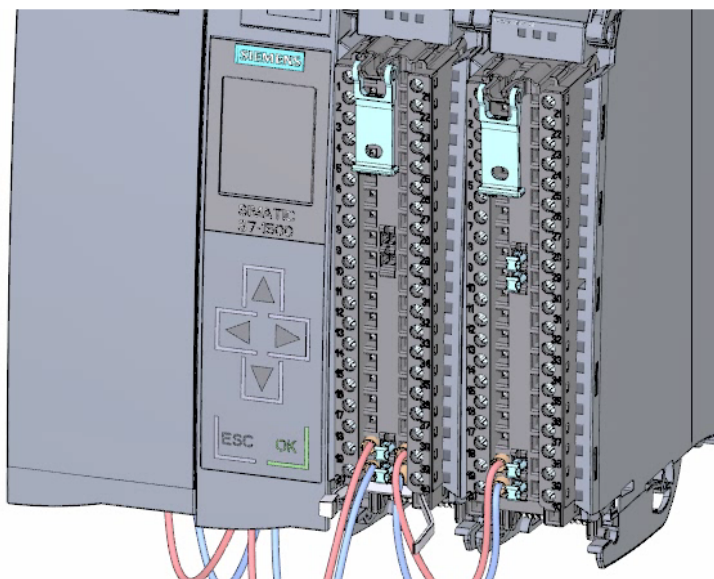
1. 将前端连接器插入预接线位置。



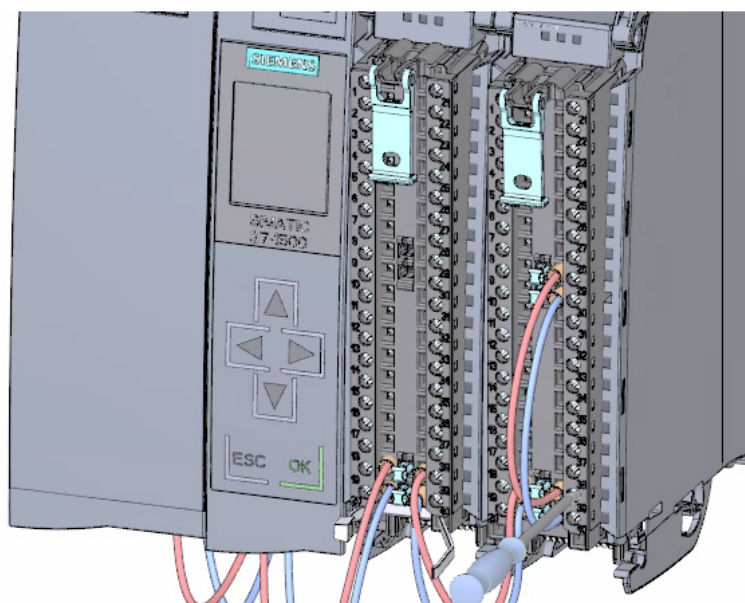
2. 通过数字量输入模块上的端子 40 (M) 和 39 (L+), 从数字量输入模块为的端子 20 (M) 和 19 (L+) 提供 DC 24 V 供电电压。



3. 连接四个电位电路桥。



4. 连接端子 30 和 40，以及 29 和 39。



## 结果

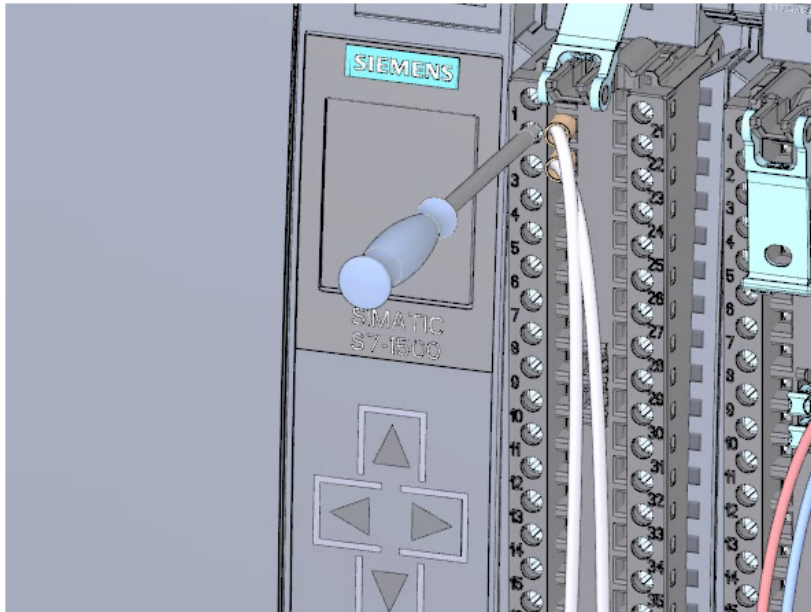
数字量输出模块现在已经接线。



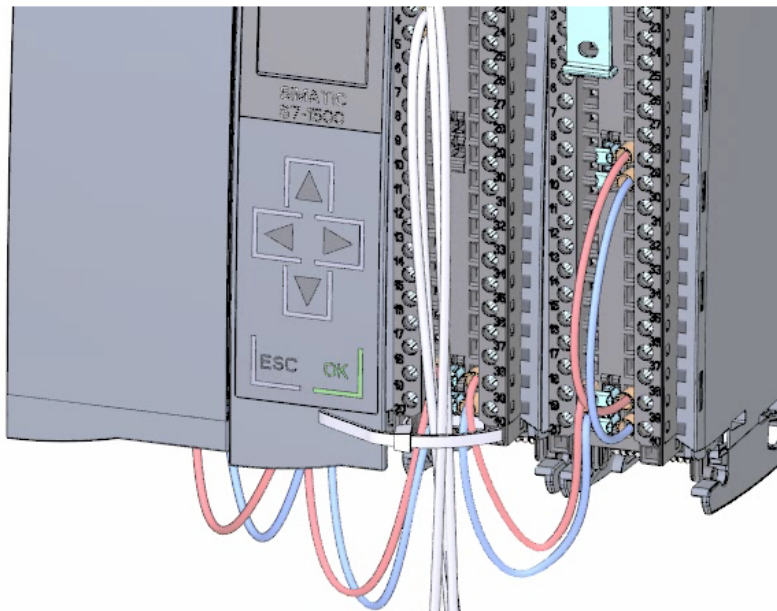
### 2.3.8 为前连接器接线

#### 步骤

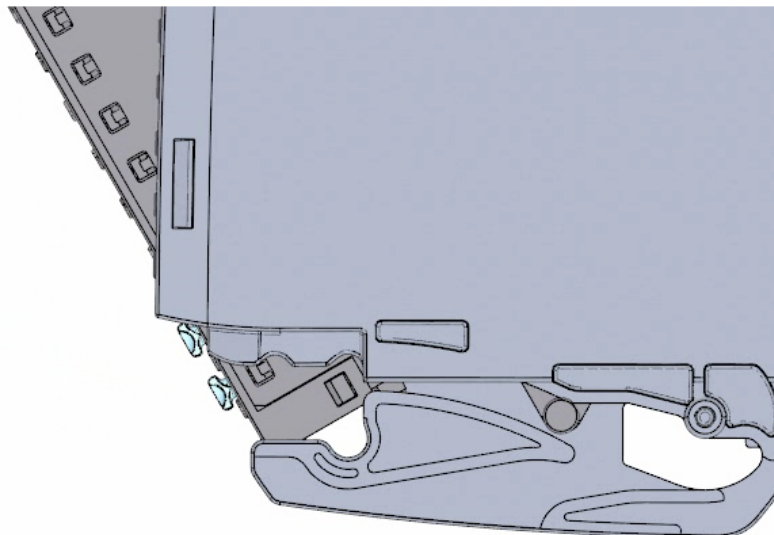
1. 根据端子前盖内侧的接线图，连接各个导线并将其拧紧。



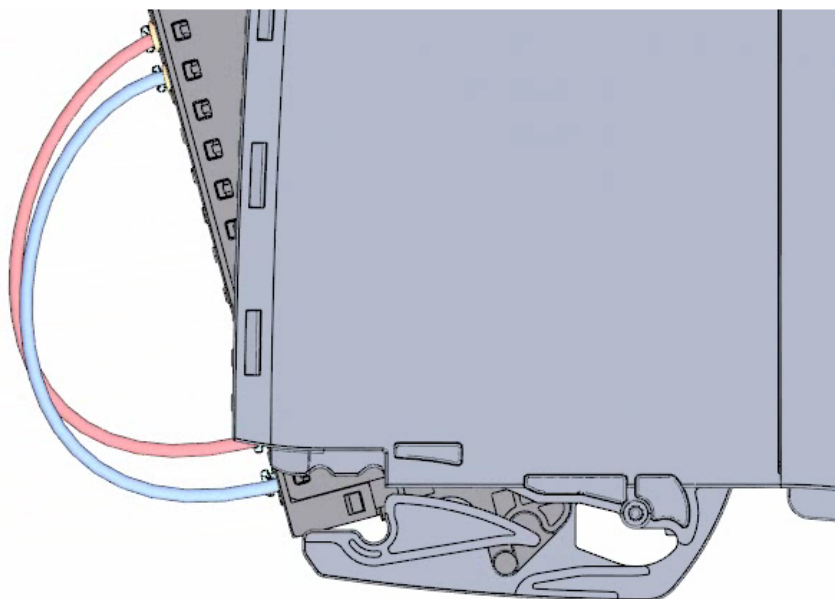
2. 为了消除张力，请使用电缆扎带固定电缆并拉紧。



3. 将前端连接器从预接线位置移到其最终位置。  
至此，已建立了前端连接器和模块间的电气连接。



4. **提示：** 可以直接插入预接线的前端连接器，例如，用于更换模块。



## 结果

前端连接器现在已经接线。

## 2.4 通电

### 2.4.1 概述

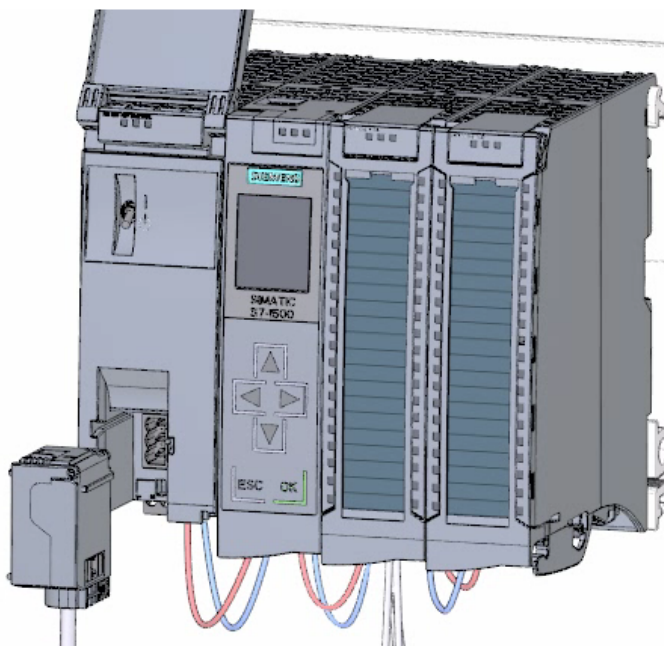
#### 首次启动 CPU

在本部分中将首次启动 CPU。

### 2.4.2 通电

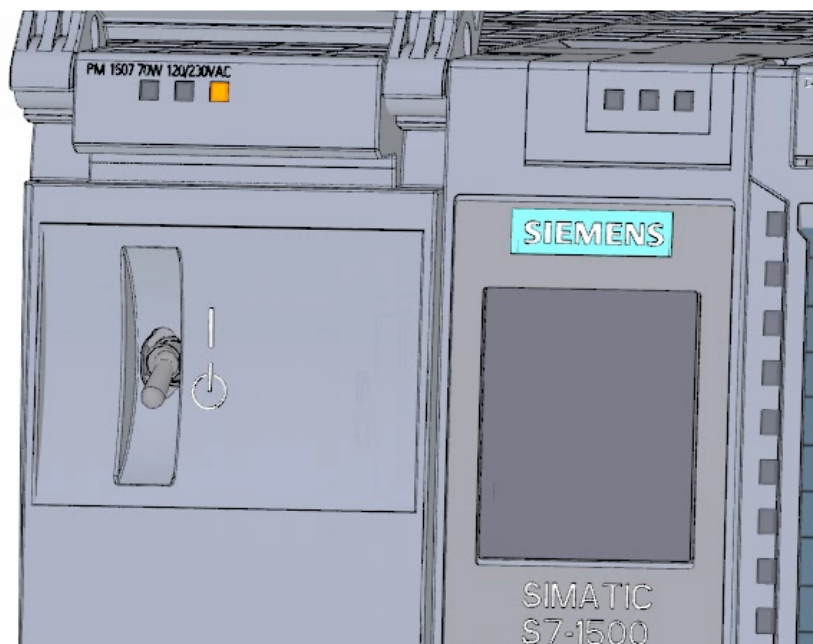
#### 步骤

1. 插入负载电流电源 (PM) 的电源连接插头。

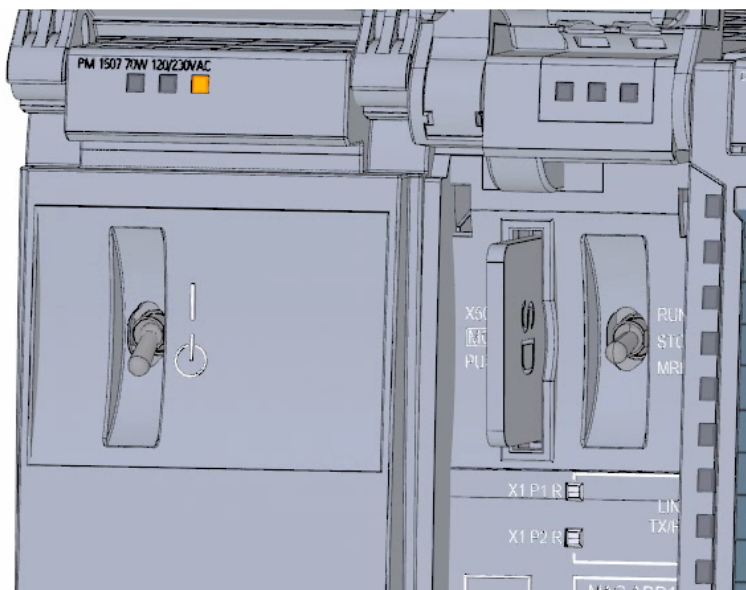




2. 将电源连接插头连接到电源。

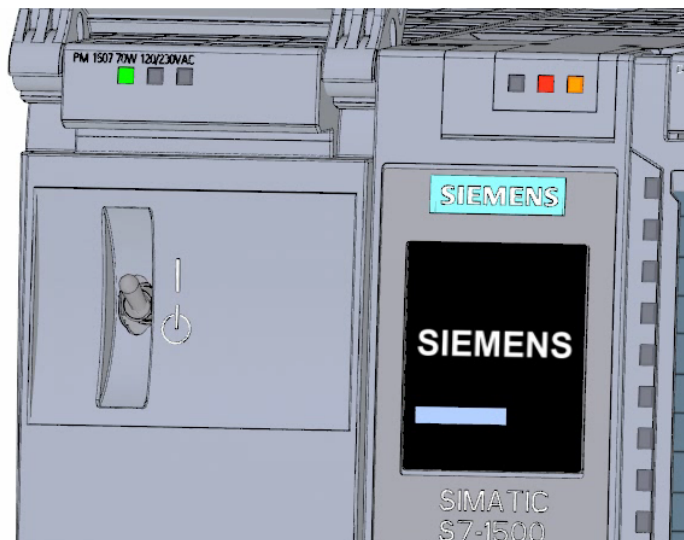


3. 将空的 SIMATIC 内存卡插入 CPU 中。



4. 将负载电源 (PM) 上的开关切换到位置 RUN 处。

将启动 CPU。



### 结果

CPU 启动并处于 STOP 模式。

### 2.4.3 通过显示屏分配 IP 地址

在这一步中，将设置 CPU 的 IP 地址和子网掩码。

### 步骤

1. 浏览到“设置”(Settings)。
2. 选择“地址”(Addresses)。
3. 选择接口“X1 (IE/PN)”。
4. 选择菜单项“IP 地址”(IP Addresses)。
5. 设置 IP 地址 192.168.0.10。
6. 按下模块上的“右”箭头键。
7. 设置子网掩码 255.255.255.0。
8. 按下模块上的“下”箭头键选择菜单项“应用”(Apply)，然后单击“确定”(OK) 确认设置。

### 结果

现在已经设置接口“X1 (IE/PN)”的 IP 地址和子网掩码。

## 软件部分

### 3.1 创建项目和硬件

#### 3.1.1 TIA Portal 简介

##### 简介

Totally Integrated Automation Portal（以下称为 TIA Portal）在单个跨软件平台中提供了实现自动化任务所需的所有功能。

TIA Portal 作为首个用于集成工程组态的共享工作环境，在单一的框架中提供了各种 SIMATIC 系统。因此，TIA Portal 还首次支持可靠且方便的跨系统协作。

所有必需的软件包，包括从硬件组态和编程到过程可视化，都集成在一个综合的工程组态框架中。



### 3.1 创建项目和硬件

#### TIA Portal 的优势

在使用 TIA Portal 时，以下功能在实现自动化解决方案期间提供高效支持。

- **使用统一操作概念的集成工程组态**  
过程自动化和过程可视化“齐头并进”。
- **通过功能强大的编辑器和通用符号实现一致的集中数据管理**  
数据一旦创建，就在所有编辑器中都可用。  
更改及纠正内容将自动应用和更新到整个项目中。
- **完整的库概念**  
可以反复使用现成的指令及项目的现有部分。
- **多种编程语言**  
可以使用五种不同的编程语言来实现自动化任务。

#### 3.1.2 创建项目

##### 简介

在下面步骤中，将创建一个新项目。

所有在创建自动化解决方案期间生成的数据都保存在项目文件中。

数据将以对象形式存储。在项目中，对象按树形结构（项目层次结构）排列。

项目层次结构基于设备和站以及它们的组态数据和程序。

##### 要求

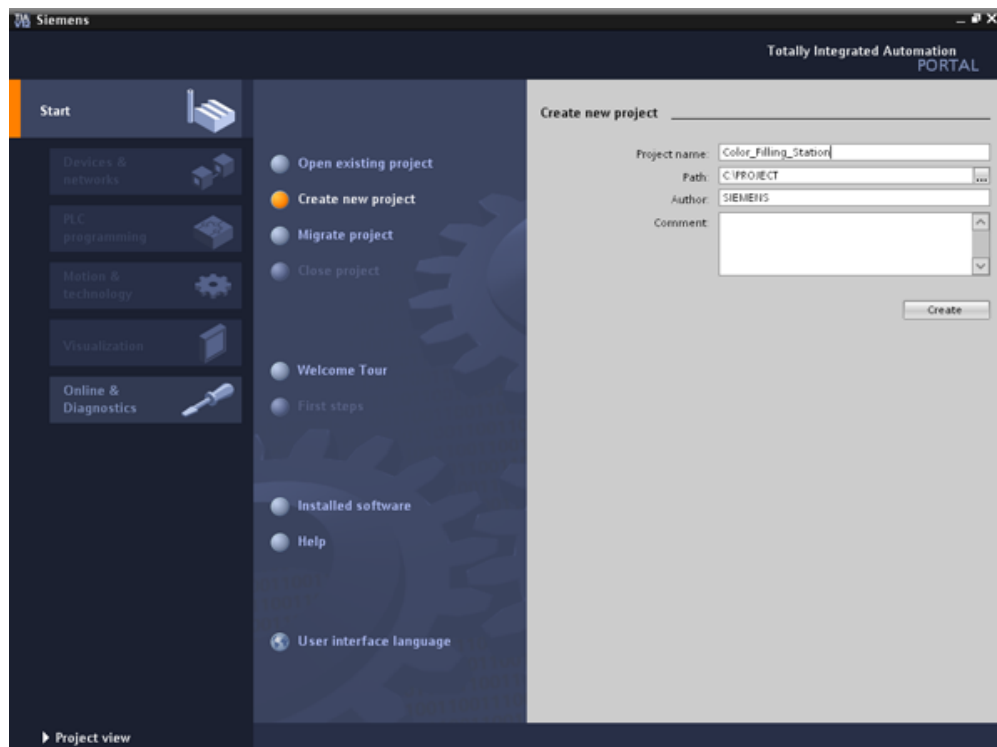
需要以下硬件和软件设备来创建项目：

- **硬件：**
  - 在“入门指南”的硬件部分中进行安装和接线的 CPU 1511-1 PN。
  - 到 PG/PC 的以太网连接。
- **软件：**  
在 PG/PC 上必须已安装以下软件包，且可执行：
  - SIMATIC STEP 7 Professional V13
  - SIMATIC WinCC Advanced V13 或 SIMATIC WinCC Professional V13

## 创建一个新项目

要创建新项目，请按以下步骤操作：

1. 单击“创建新项目”(Create new project)。
2. 输入项目名称。

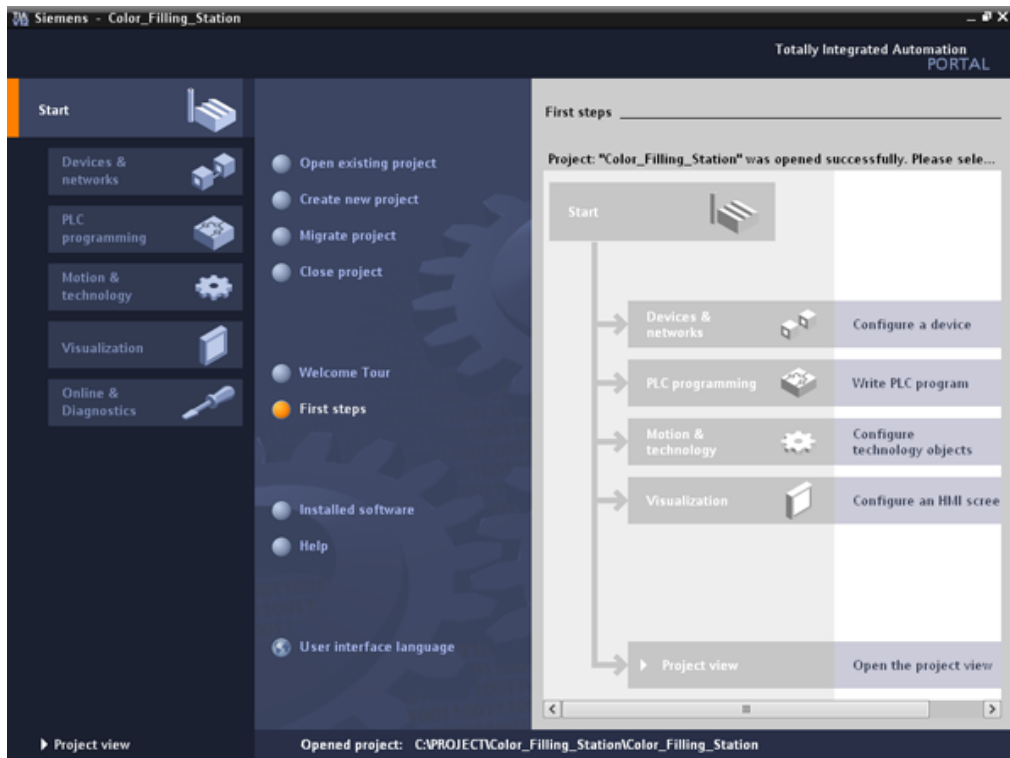


3. 单击“创建”(Create) 以创建新项目。

### 3.1 创建项目和硬件

#### 结果

创建了项目。所有的数据（如硬件配置数据、HMI 中的 CPU 编程和可视化数据）都保存在项目中。



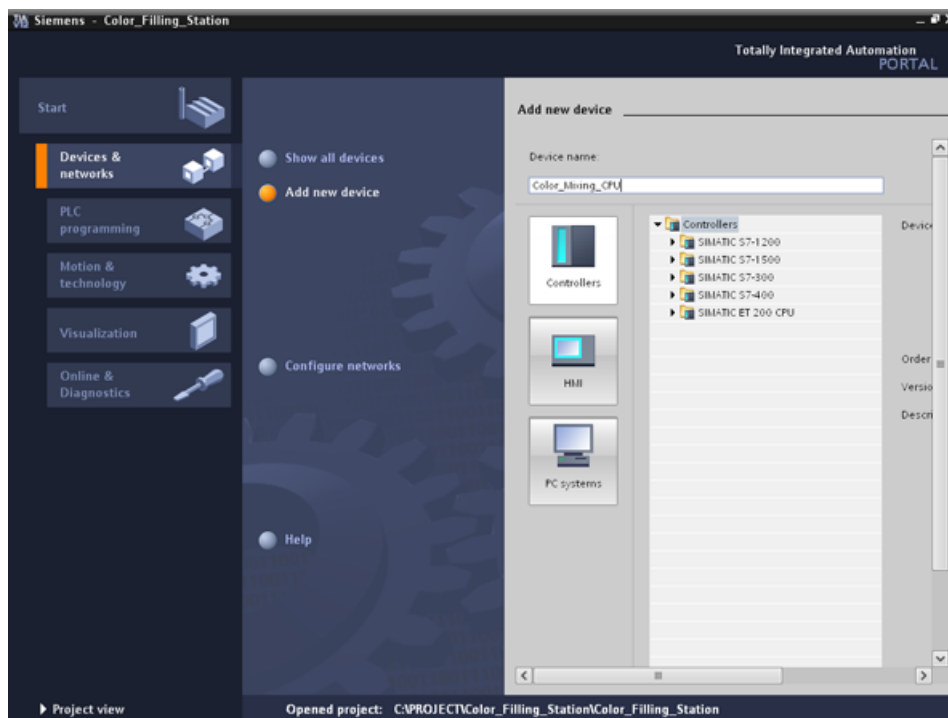
### 3.1.3 创建 S7-1500 CPU

#### 简介

在下面步骤中，将创建一个未指定的 CPU。未指定的 CPU 是稍后将定义的硬件目录中的特定 CPU 的占位符。

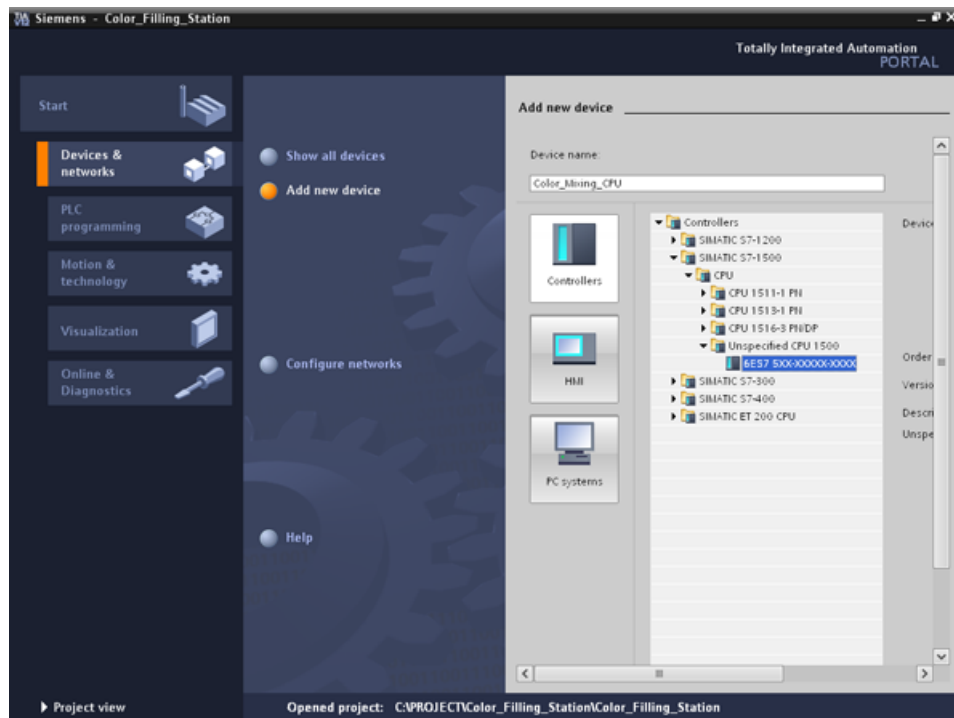
#### 步骤

1. 打开“设备与网络”Portal。
2. 插入新设备。
3. 输入“Color\_Mixing\_CPU”作为 CPU 名称。



4. 打开“SIMATIC S7-1500”文件夹。

5. 选择尚未指定的 CPU。

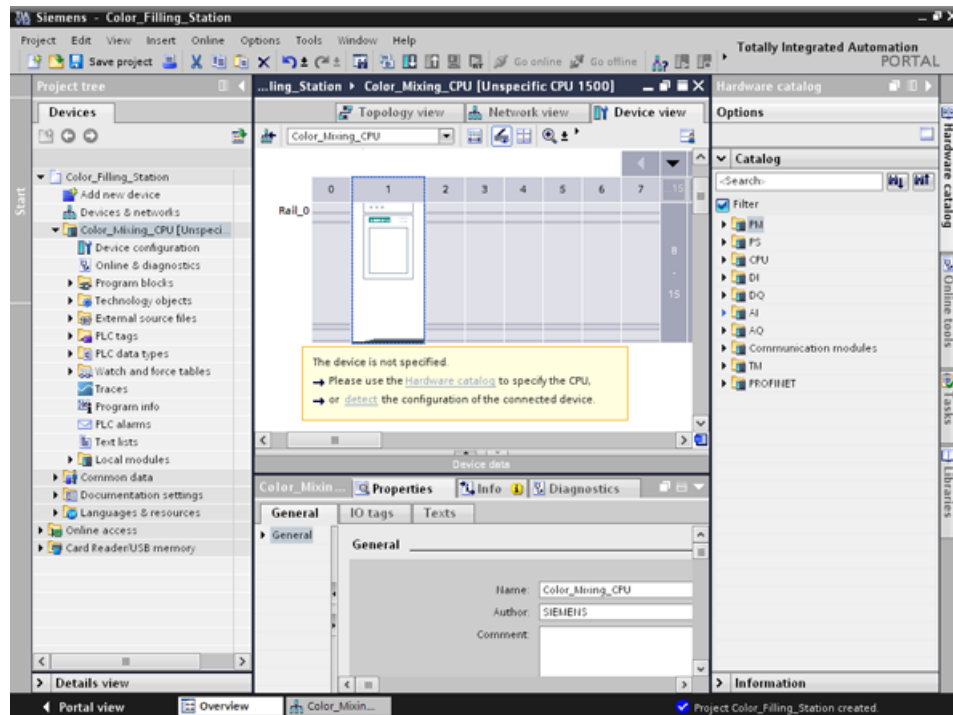


6. 双击创建 CPU。



## 结果

在项目文件中创建了未指定的 CPU。可在此为 CPU 创建用户程序内容。



### 3.1 创建项目和硬件

#### 3.1.4 运行硬件检测

##### 简介

在以下部分中，将使用硬件检测功能读取 CPU 类型。

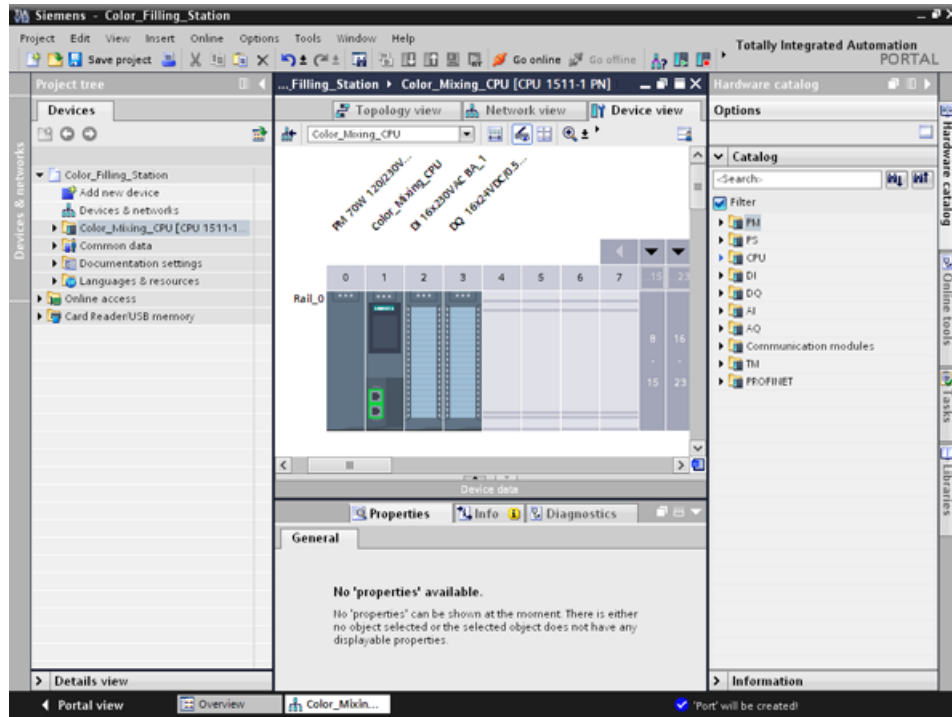
在硬件检测期间运行 LED 闪烁测试。LED 闪烁测试激活了所检测设备上的 LED 指示灯。还可使用此功能验证是否在包含多个设备的硬件配置中选择了正确的设备。

##### 步骤

1. 在项目树中选择未指定的 CPU。
2. 从“在线”(Online) 菜单中选择“硬件检测”(Hardware detection) 功能。  
选项 2: 在设备视图中单击黄色框报警。
3. 选择“PN/IE”条目作为 PG/PC 接口类型。
4. 选择 PG/PC 接口。
5. 单击“显示所有兼容设备”(Show all compatible devices) 选项。
6. 从子网中的兼容设备中选择 CPU。
7. 选择“闪烁 LED”(Flash LED) 复选框以运行闪烁测试。
8. 单击“检测”(Detect) 将未指定的 CPU 替换为所需 CPU 类型。

## 结果

读出 CPU 类型。为项目树中的 CPU 名称附加了用括号括起的正确设备名。  
所用 CPU 和模块显示在硬件配置中。



### 3.1.5 创建 ET 200 接口模块

#### 简介

在以下部分中，将在硬件配置中创建 2 个分布式 I/O 系统：

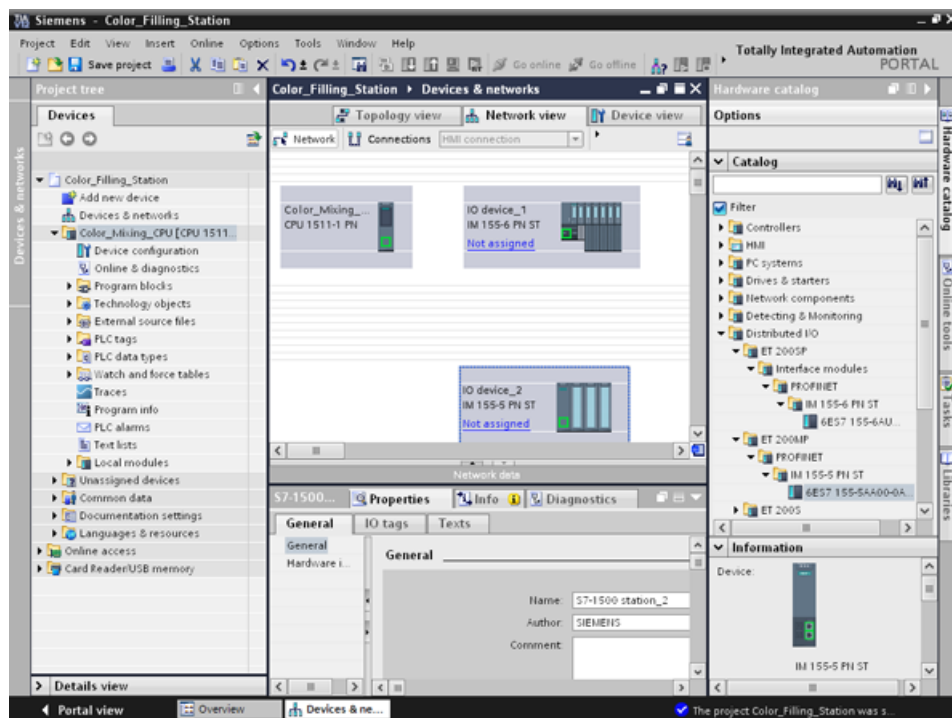
- ET 200SP 分布式 I/O 系统，主要包含下列组件：
  - 与 CPU 通信的接口模块。
  - 最多 32 个可以任意组合插入的模块。
  - 用于完成组态的服务模块。
- ET 200MP 分布式 I/O 系统，包含下列组件：
  - 与 CPU 通信的接口模块。
  - 最多 30 个模块，每个均提供多达 32 个通道。

#### 步骤

1. 打开“硬件目录”(Hardware catalog)。
2. 切换到“网络视图”(Network view)。
3. 打开“分布式 I/O”(Distributed I/O) 和“ET 200SP”文件夹。
4. 打开“IM 155-6 PN ST”文件夹。
5. 将“6ES7 155-6AU00-0BN0”接口模块拖放到网络视图中。
6. 打开“ET 200MP”文件夹。
7. 打开“IM 155-5 PN ST”文件夹。
8. 将“6ES7 155-5AA00-0AB0”接口模块拖放到网络视图中。

## 结果

已在硬件配置中创建了 I/O 系统，但尚未分配给 CPU 1511-1 PN。它们均显示在项目视图的“未分配设备”(Unassigned devices) 下。



## 更多信息

SIMATIC ET 200 产品系列提供了各种可扩展 I/O 系统以适合您的特定应用。

有关 SIMATIC ET 200 分布式 I/O 的更多信息，请访问下列 Internet 网址 Hotspot-Text (<http://www.automation.siemens.com/mcms/distributed-io/en/>)。

### 3.1.6 联网 ET 200 接口模块

#### 简介

在以下部分中，将创建一个 PROFINET I/O 系统。

PROFINET I/O 系统包含 PROFINET IO 控制器及其分配的 PROFINET IO 设备：

- 将已创建的 CPU 1511-1 PN 用作 PROFINET IO 控制器。
- 将这两个分布式 I/O 系统用作 PROFINET IO 设备。

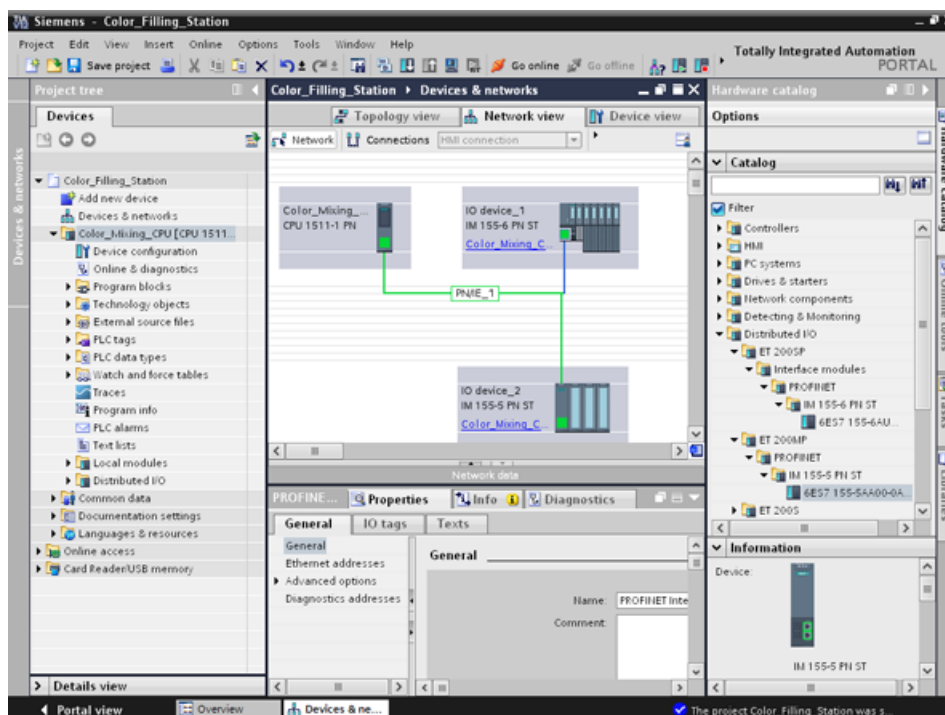
#### 步骤

1. 将连接从 IM 155-5 PN ST 接口模块接口拖放到 CPU 接口。
2. 在 IM 155-6 PN ST 接口模块与 CPU 之间建立另一个连接。

#### 结果

接口模块作为 IO 设备分配给 CPU。两个分布式 IO 系统显示在项目树的“分布式 I/O”(Distributed I/O) 文件夹中的 CPU 下方。

PROFINET I/O 系统在联网过程中自动创建，其属性显示在网络视图中。



### 3.1.7 为 ET 200SP 创建输入和输出模块及服务模块

#### 简介

在以下部分中，将为 ET 200SP 创建输入和输出模块。

#### 说明

需要服务模块来运行输入和输出模块。缺少服务模块将导致这些模块发生故障。

#### 每个电位组的最大组态

各电位组可使用的 I/O 模块数取决于下列因素：

1. 此电位组上运行的所有 I/O 模块的电源总需求
2. 从外部连接到此电位组上的所有负载的电源总需求

根据 1.和 2.计算得到的电源总和不得超出所用的 BaseUnit 和负载电源电压的载流能力。

如下设置模块的“电位组”参数：

参数	取值范围	用途
电位组	使用左侧模块的电位组（默认设置）	如果左侧所有模块总功耗与模块的功耗之和低于 BaseUnit 的载流能力
	启用新的电位组	如果左侧所有模块总功耗与模块的功耗之和高于 BaseUnit 的载流能力

有关电位组的更多信息，请参见 SIMATIC ET 200SP DI 8x24VDC HF 数字量输入模块 (<http://support.automation.siemens.com/DE/view/en/66912542>)的模块手册。

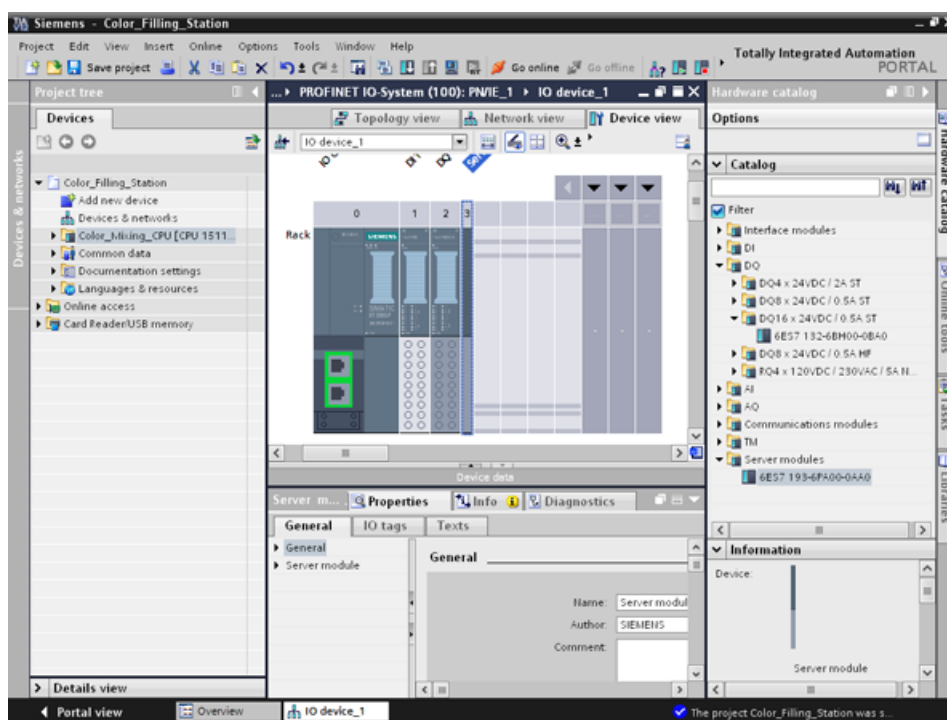
### 3.1 创建项目和硬件

#### 步骤

1. 打开 ET 200SP 设备视图。
2. 打开硬件目录中的“DI”和“DI16 x DC24V ST”文件夹。
3. 将输入模块“6ES7 131-6BH00-0BA0”拖放到导轨的插槽 1 中。
4. 打开“DQ”和“DQ16 x DC24V / 0.5A ST”文件夹。
5. 将输出模块“6ES7 132-6BH00-0BA0”拖放到导轨的插槽 2 中。
6. 打开“服务模块”(Server modules) 文件夹。
7. 将服务模块“6ES7 193-6PA00-0AA0”拖放到导轨的插槽 3 中。

#### 结果

创建了输入和输出模块以及服务模块。





### 3.1.8 为 ET 200MP 创建输入和输出模块

#### 简介

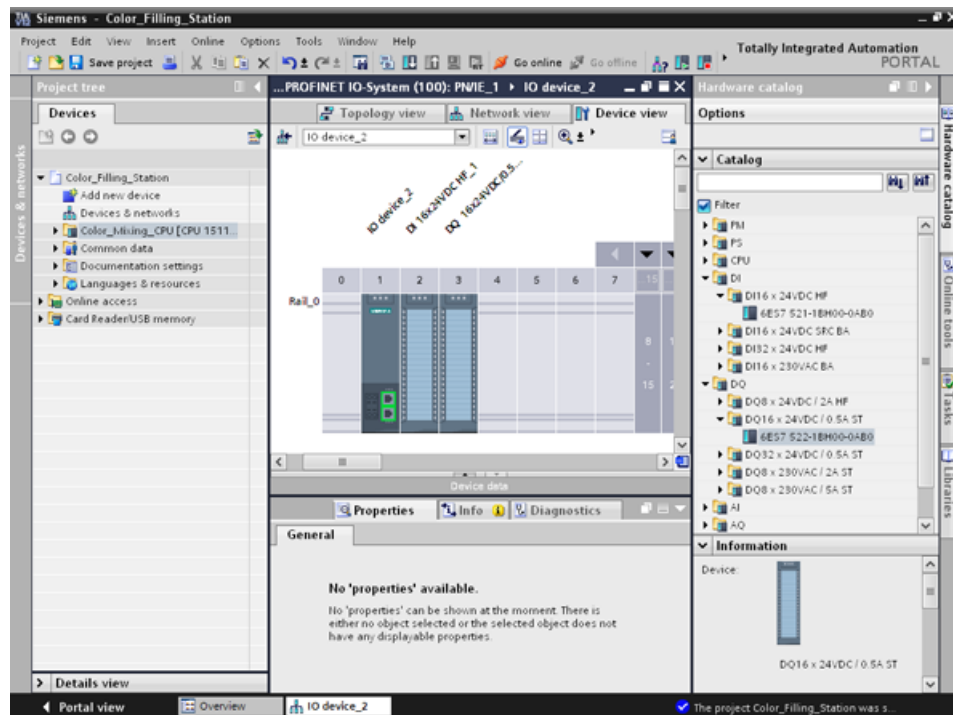
在以下部分中，将为 ET 200MP 创建输入和输出模块。

#### 步骤

1. 打开 ET 200MP 设备视图。
2. 打开硬件目录中的“DI”和“DI16 x DC24V HF”文件夹。
3. 将输入模块“6ES7 521-1BH00-0AB0”拖放到导轨的插槽 2 中。
4. 打开“DO”和“DO16 x DC24V / 0.5A ST”文件夹。
5. 将输出模块“6ES7 522-1BH00-0AB0”拖放到导轨的插槽 3 中。

#### 结果

创建了输入和输出模块。



### 3.1 创建项目和硬件

#### 3.1.9 为 ET 200 指定名称

##### 简介

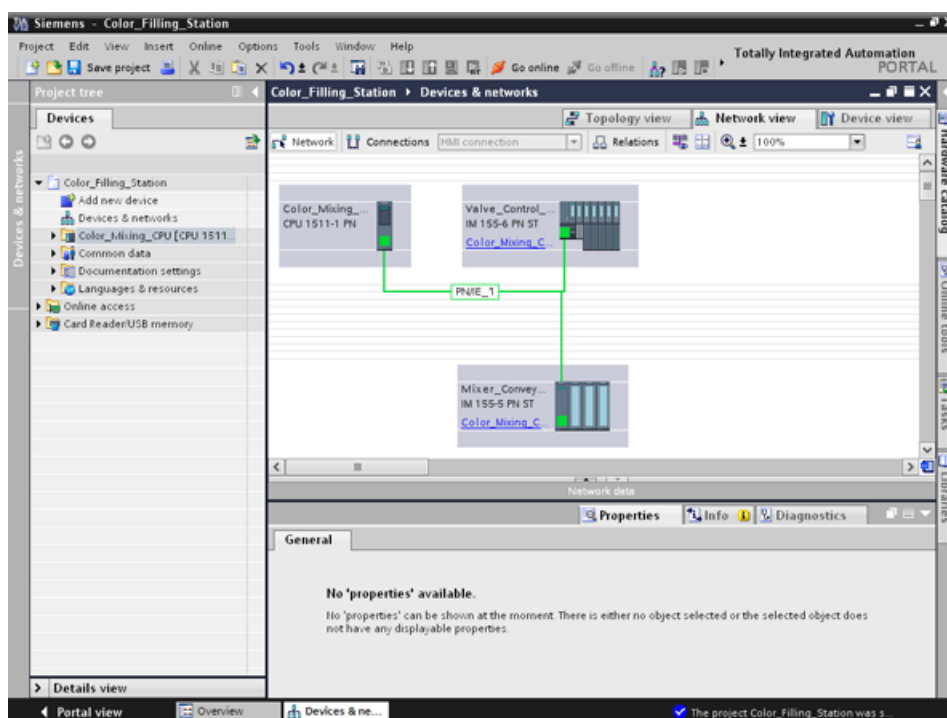
在以下部分中，将为分布式 I/O 指定项目特定的名称。

##### 步骤

1. 选择 ET 200SP。
2. 在巡视窗口的属性 > 常规(Properties > General) 中的“名称”(Name) 域中，输入名称“Valve\_Control\_Unit”。
3. 选择 ET 200MP，并输入新名称“Mixer\_Conveyor\_Control\_Unit”。

##### 结果

指定了项目特定的名称。



## 3.2 创建程序

### 3.2.1 加载块库

#### 简介

在以下部分中，将加载全局库“ProgLib\_ColorFillingStation”。  
该库中的块和变量表是示例项目所需的。该库在“S7-1500/TIA V13 入门指南  
([http://www.automation.siemens.com/salesmaterial-as/interactive-manuals/getting-started\\_simatic-s7-1500/project/color\\_filling\\_station.zip](http://www.automation.siemens.com/salesmaterial-as/interactive-manuals/getting-started_simatic-s7-1500/project/color_filling_station.zip))”中以 ZIP 文件形式提供。  
此库需要先解压缩，然后才能导入到项目中。

#### 全局库

全局库用于存储要在其它项目中重复使用的元素。必须明确创建全局库。  
标准包中提供了下列库：

- "Buttons and Switches"

这些库提供了大量的开关和按钮。通过文件夹将开关和按钮按类分组。  
例如，可在“DiagnosticsButtons”文件夹中找到“系统诊断指示器”对象。  
使用“系统诊断指示器”对象对工厂进行系统诊断。

- "Monitoring and Control objects"

这样就提供了适用于各种设计的复杂操作员控制和显示对象以及合适的控制灯、按钮和开关。

---

#### 说明

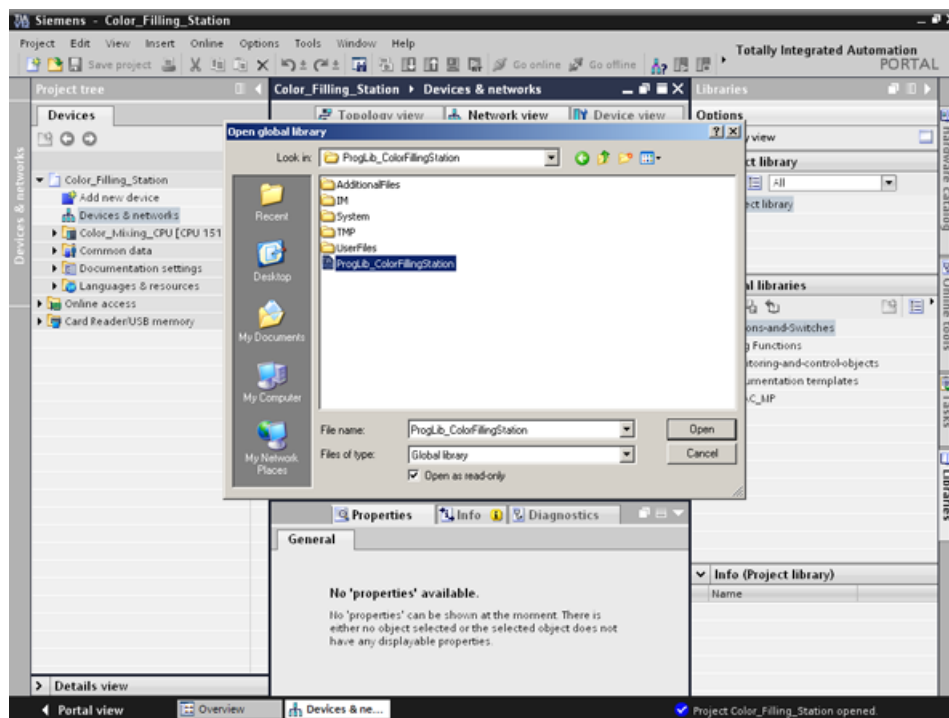
##### 库具有写保护

默认情况下，激活了“打开全局库”(Open global library)  
对话框中的“以只读方式打开”(Open read-only) 选项。  
选中此复选框以在无写保护的情况下打开库。

---

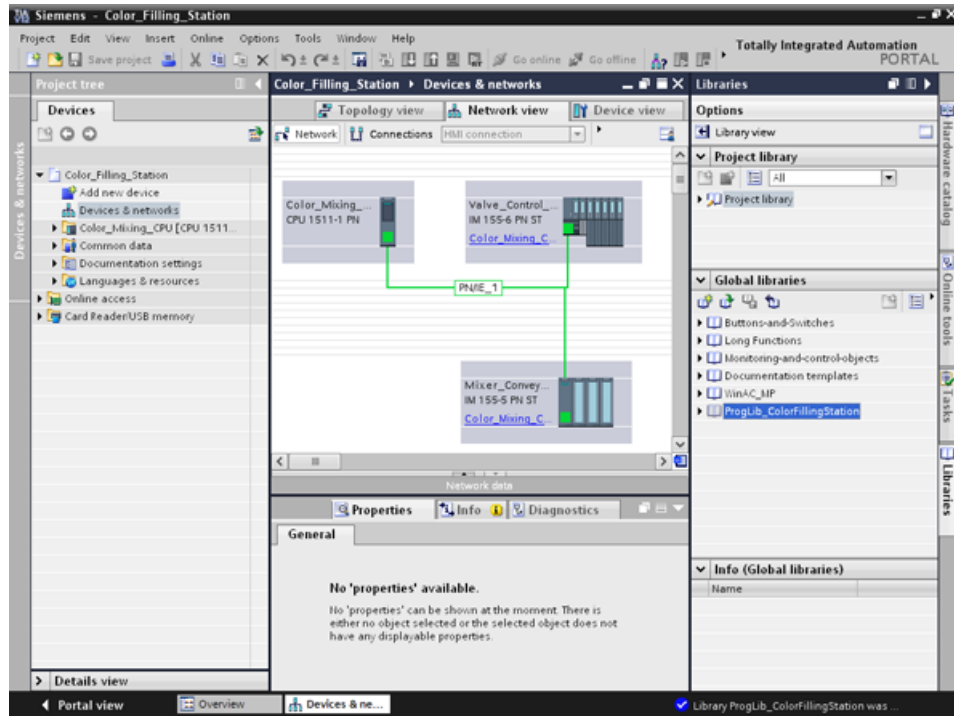
步骤

1. 单击“库”(Libraries) 选项卡。
2. 单击“打开全局库”(Open global library)。
3. 在包含已解压库文件夹的目录中选择“ProgLib\_ColorFillingStation”文件，然后单击“打开”(Open)。



结果

打开“ProgLib\_ColorFillingStation”全局库。

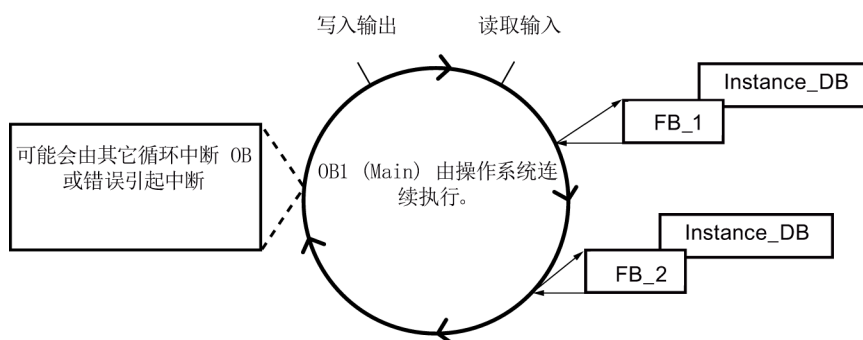


### 3.2.2 删除程序块 Main [OB1]

#### 简介

在以下部分中，将删除项目文件夹中自动生成的“Main [OB1]”程序块。“Main [OB1]”程序块包含在示例项目程序块中。

组织块 (OB) 是 CPU 操作系统和用户程序之间的接口。这些块由操作系统调用。自动化项目中必须至少存在一个循环 OB。

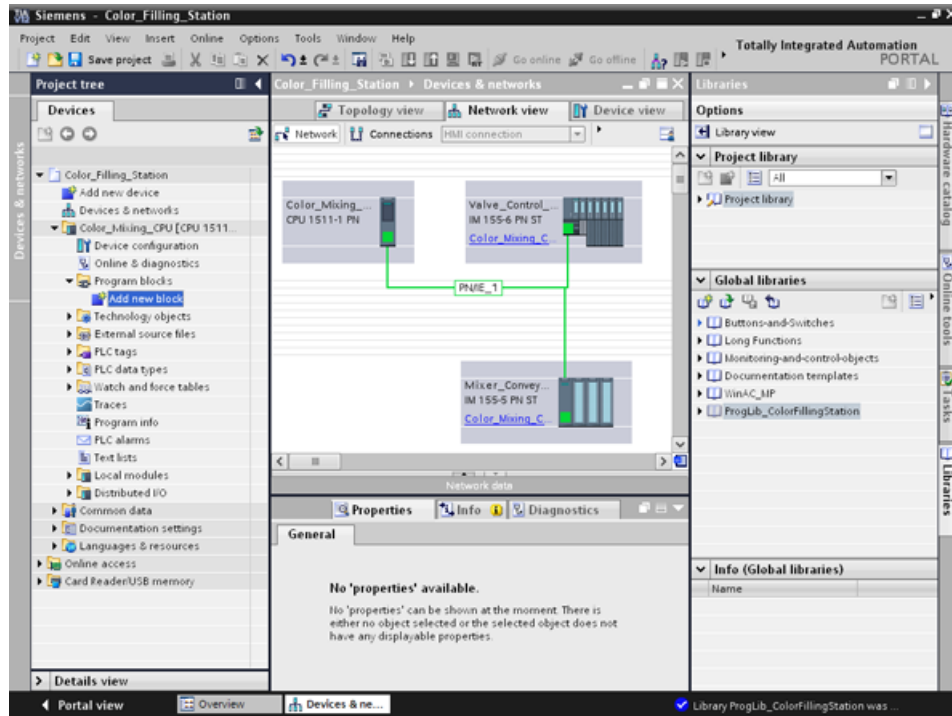


#### 步骤

1. 打开项目树中的“程序块”(Program blocks) 文件夹，然后单击“Main [OB1]”程序块。
2. 右键单击以打开快捷菜单并单击“删除”(Delete)。
3. 单击“是”(Yes) 确认删除块。

## 结果

删除自动生成的“Main [OB1]”程序块。



### 3.2.3 复制程序块

#### 简介

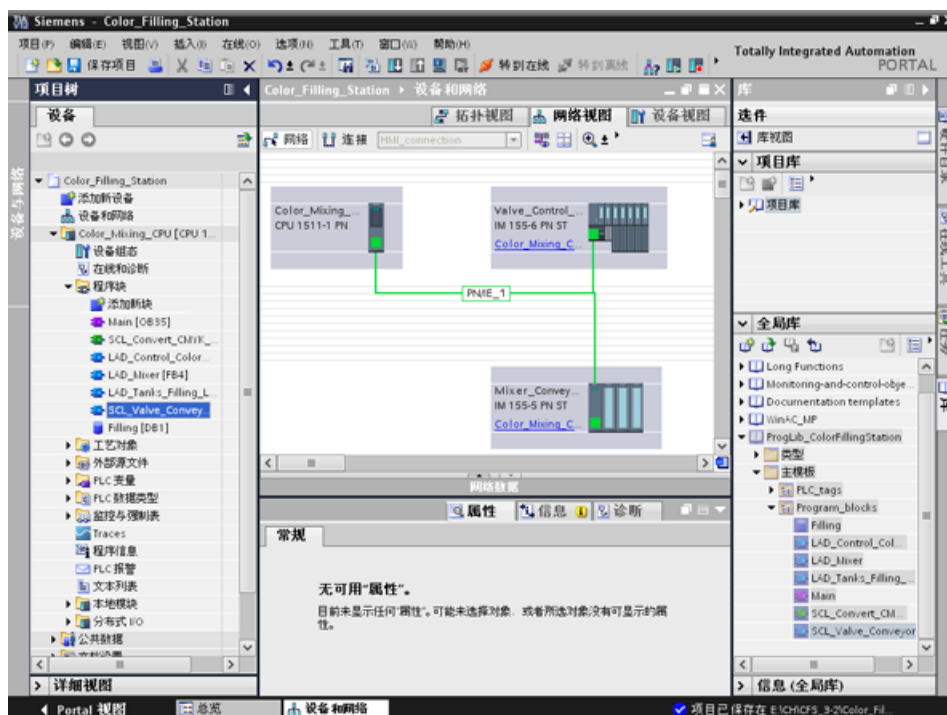
在以下部分中，将“ProgLib\_ColorFillingStation”全局库中的程序块插入到项目中。

#### 步骤

1. 单击全局库“ProgLib\_ColorFillingStation”。
2. 依次单击“主模板”(Master copies) 文件夹和“Programm\_blocks”。
3. 将要导入的程序块从全局库拖放到“程序块”(Program blocks) 文件夹。
4. 对于其它块，请按步 2 和 3 中的描述进行操作。

#### 结果

在同名项目文件夹中插入程序块。





### 3.2.4 循环中断 OB

#### 3.2.4.1 循环中断 OB - 循环时间和相位

##### 可更改循环时间和相位偏移量

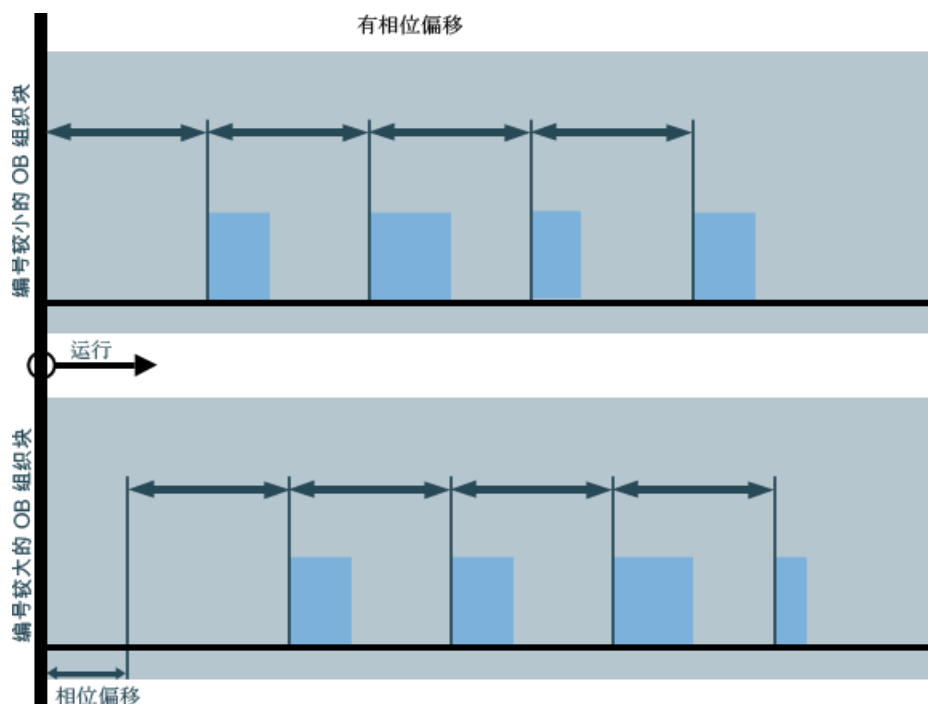
Main [OB35] 位于插入在项目中的程序块下方。Main [OB35] 是一个循环中断组织块（循环中断 OB）。循环中断 OB 将以固定时间间隔启动程序，而与循环程序的执行无关。循环中断 OB 的启动时间可通过循环时间和相位偏移量进行指定。

##### 循环时间

循环时间将决定调用 OB 的时间间隔。默认情况下，循环中断 OB 的循环时间为 100000  $\mu\text{s}$ 。

##### 相位偏移

相位偏移量用于提高周期中断程序的处理时间间隔的准确性。如果 OB 与另一个 OB 具有相同或公用一个多时钟脉冲，则可通过相位偏移量以精确的间隔执行这两个 OB。



### 3.2.4.2 更改循环时间

#### 简介

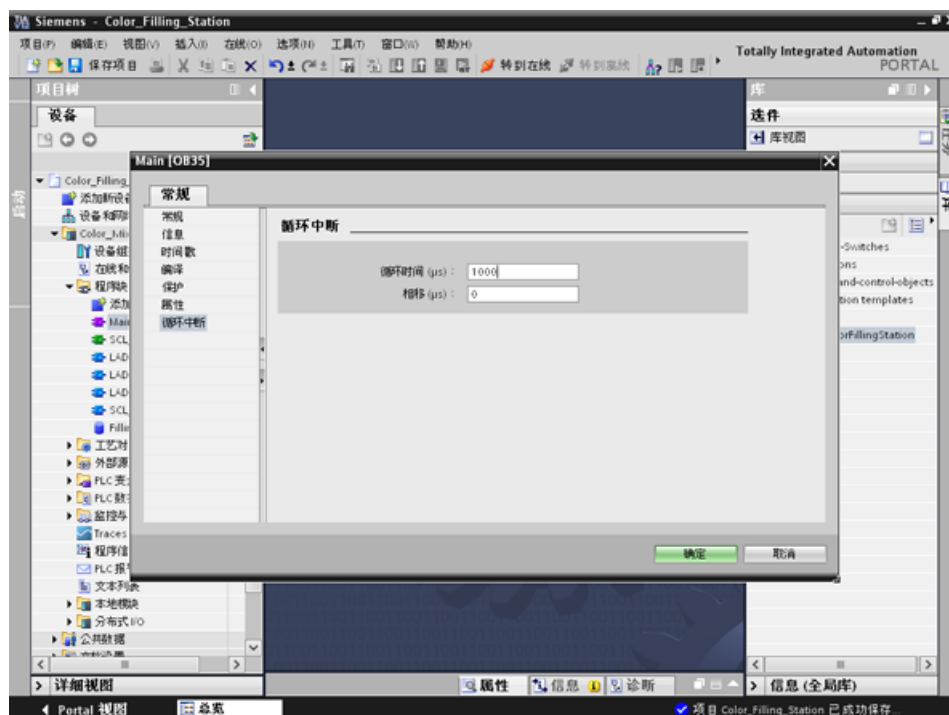
在以下部分中，将更改“Main”程序块的循环时间。

#### 要求

- 程序块“Main”[OB35] 包含在库中
- 存在 FB/FC 调用

#### 步骤

1. 打开“Main”程序块的属性。
2. 请在“常规”(General) 下选择“循环中断”(Cyclic interrupt) 选项。
3. 在“循环时间”(Cycle time) 中输入一个新值，然后单击“确定”(OK)。



#### 结果

循环时间已更改。

### 3.2.5 复制变量表

#### 简介

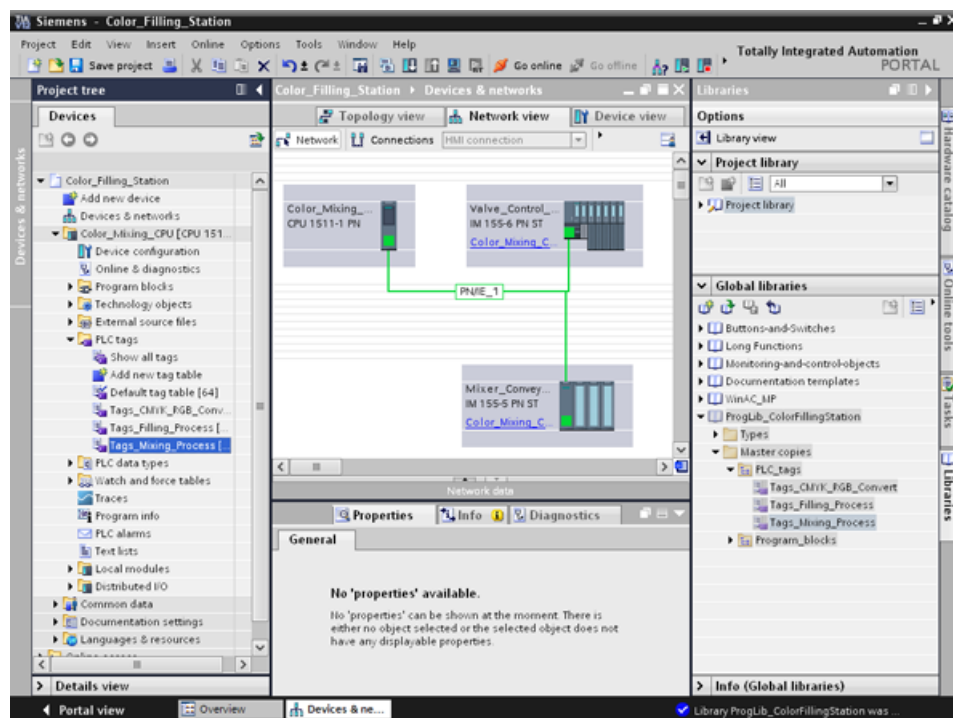
在以下部分中，将“ProgLib\_ColorFillingStation”全局库中的变量表插入到项目中。

#### 步骤

1. 在项目导航中打开“PLC 变量”(PLC tags) 文件夹。
2. 打开“PLC\_tags”文件夹。
3. 将要导入的变量表从全局库拖放到“PLC 变量”(PLC tags) 文件夹。
4. 对于其它变量表，请按步 3 中的描述进行操作。

#### 结果

在同名项目文件夹中插入变量表。



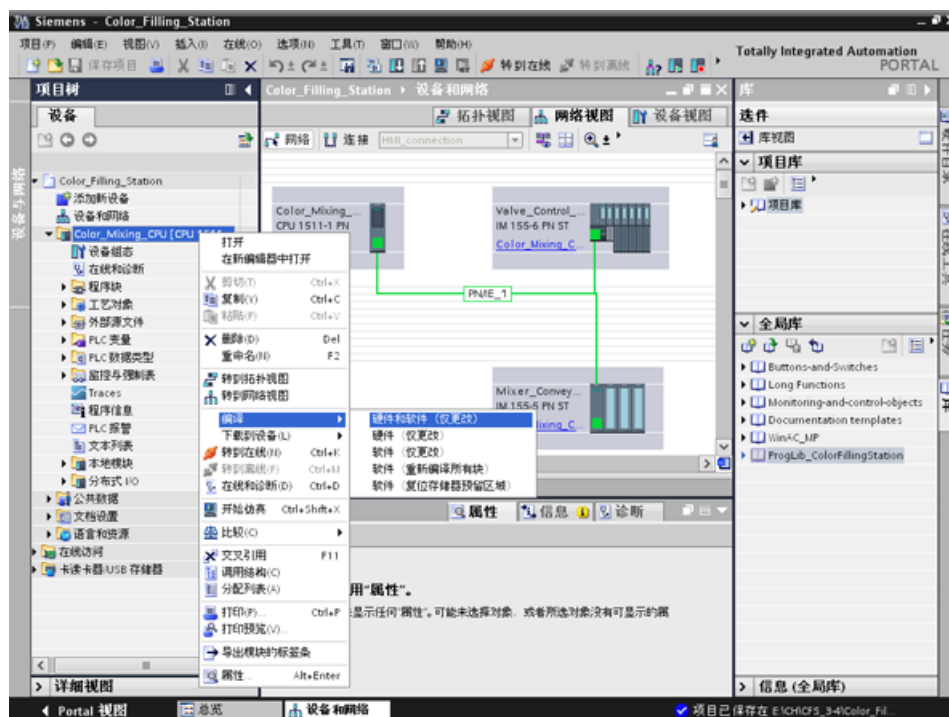
### 3.2.6 编译项目

#### 简介

在以下部分中，将编译“Color\_Filling\_Station”项目。

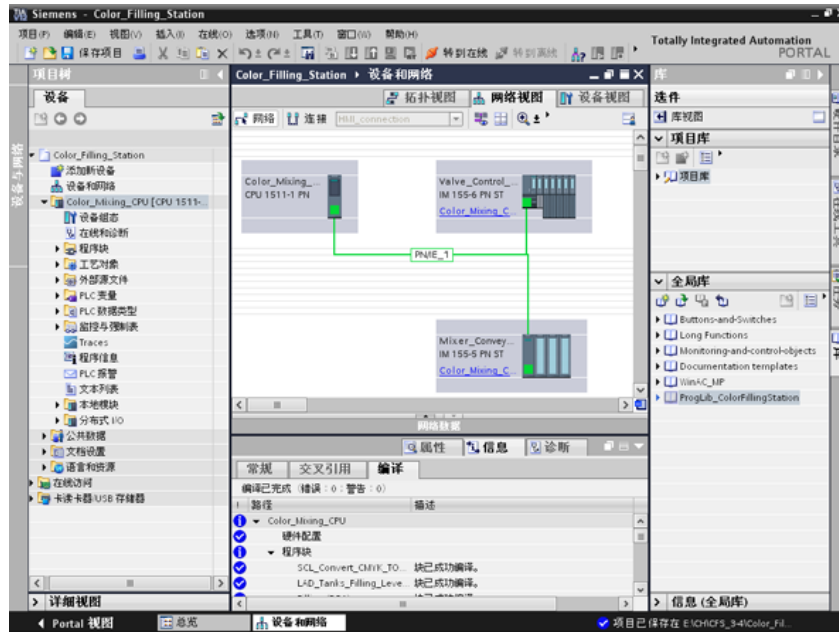
#### 步骤

1. 在项目树(project tree) 中选择“Color\_Mixing\_CPU”CPU。
2. 右键单击以打开快捷菜单，然后选择“编译” > “硬件和软件（仅更改）”(Compile > Hardware and software (only changes))。



## 结果

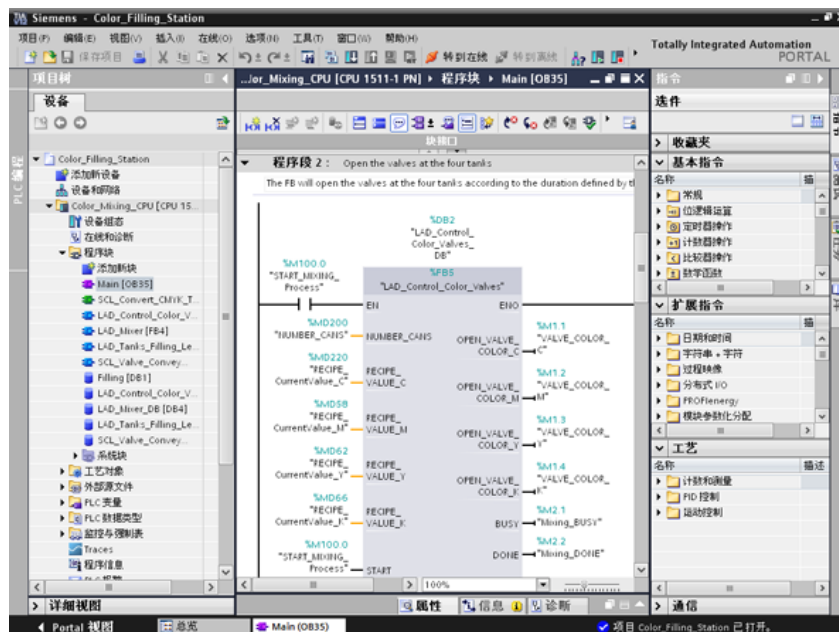
项目已编译并可下载。



## 说明

更新了“Main”程序块。

编译后打开“Main”程序块。已创建了所有背景数据块，并更新了数据库。



## 3.2.7 将项目加载到 CPU

### 简介

在以下部分中，将“Color\_Filling\_Station”项目下载到 CPU。

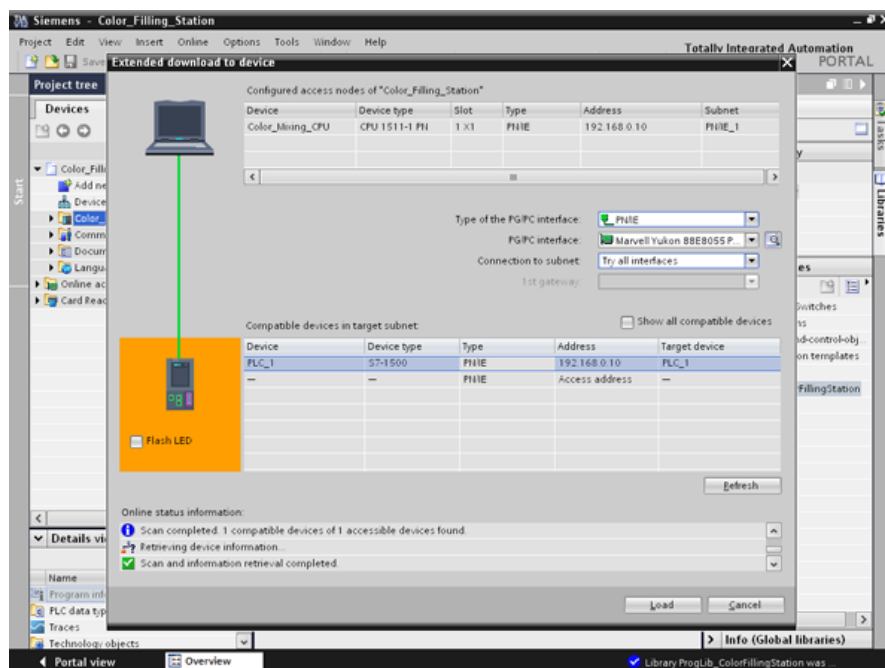
### 说明

#### 显示所有兼容设备

如果在“扩展下载到设备”(Extended download to device) 对话框中进行设置后未显示所需 CPU，请单击选项“显示所有兼容设备”(Show all compatible devices)。

### 步骤

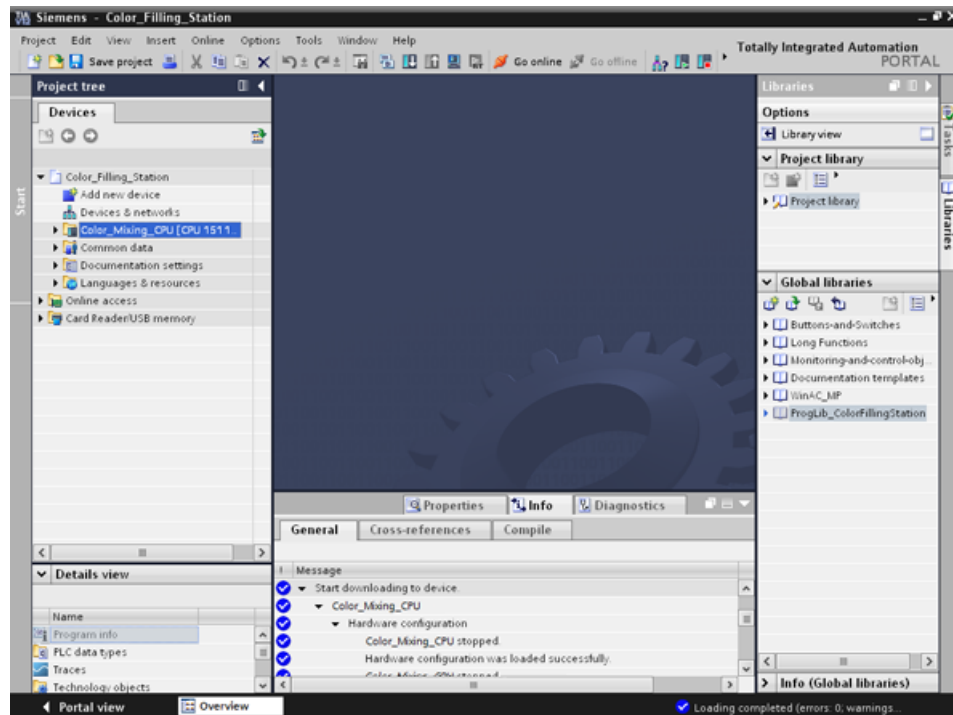
1. 打开 CPU 快捷菜单，选择“下载到设备”>“硬件和软件（仅更改）”(“Download to device”>“Hardware and software (only changes)”)。
2. 从下拉列表中，选择 PG/PC 接口类型、接口以及与子网的连接。
3. 从子网中的兼容设备中选择 CPU，并单击“加载”(Load)。



4. 通过单击“是”(Yes) 和“确定”(OK) 确认这两个“指定 IP 地址”(Assign IP address) 对话框。
5. 对于所有设置为“无操作”(No action) 的条目，在“加载预览”(Load preview) 对话框的下拉列表中选择替代条目，并确认打开选项。
6. 单击“加载”(Load)。
7. 确认“全部启动”(Start all) 选项并单击“完成”(Finish)。

## 结果

项目下载到 CPU 中。



## 3.2.8 优化块访问

### 3.2.8.1 简介

#### 工作原理

S7-1500 系列的 CPU 的“优化数据块”在性能方面得到优化，仅使用符号进行编程。通过使用优化数据块，可以使程序更加高效，这是因为将为声明的变量指定符号名称，而不向其提供固定地址。

您可以创建具有任意结构的数据块，而无需在意各个数据元素的物理排列方式。由于数据的存储方式已优化并由系统进行管理，因此可快速访问经优化的数据。

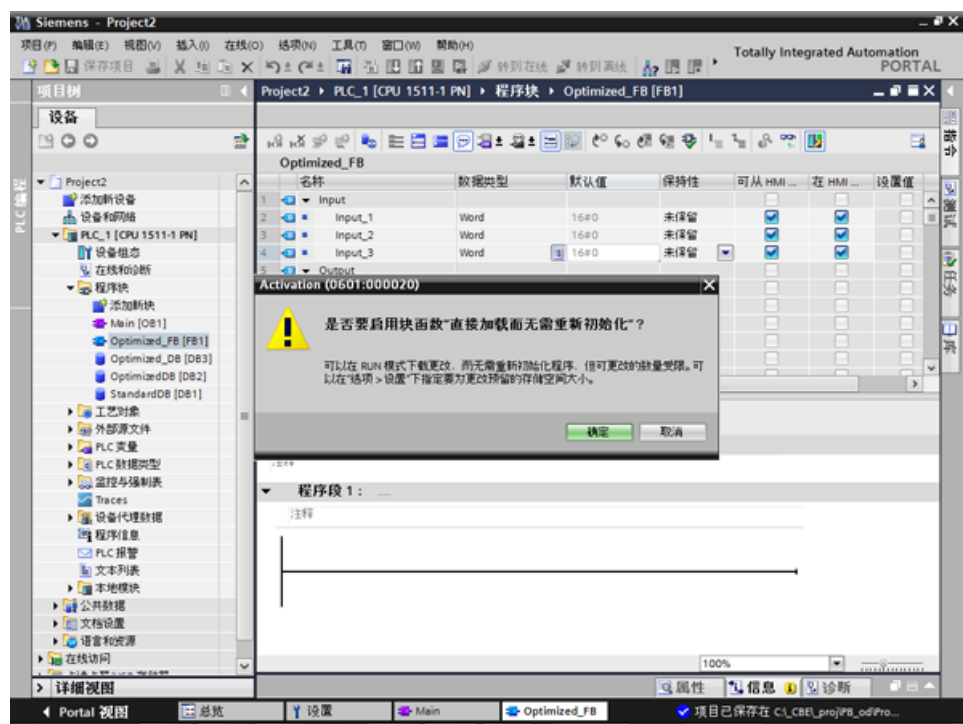
更改数据类型会增加标准块中出现错误的风险。

在优化块中，进行的更改会导致数据存储的重新安排。寻址保持唯一。

3.2 创建程序

为了能够随后编辑已在 CPU 中运行的用户程序，S7-1500 CPU 支持在运行期间扩展函数或函数块的接口。无需将 CPU 设置为 STOP 模式，即可下载已修改的块，此时也不会影响已加载变量的实际值。

另外：可以在数据块中定义 CPU 中的哪些值对于 HMI 设备是只读的（“在 HMI 中可见”(Accessible from HMI)）或者可以写入哪些值（“可从 HMI 访问”(Accessible from HMI)）。





### 3.2.8.2 扩展并重新加载经优化的“填充”数据块

#### 简介

在以下部分中，将使用上一次的填充日期和时间对“Filling”数据块进行补充并重新加载该数据块。

为此，创建一个用于记录日期和时间的块并启用“下载而不重新初始化”(Download without reinitialization) 功能。

**注：**“下载而不重新初始化”(Download without reinitialization) 功能可防止在下载 CPU 期间覆盖该数据块的实际参数。

#### 符号寻址的优势：

在整个项目中使用统一应用且有意义的符号可以使程序代码更易于阅读和理解。这种方法具有以下优点：

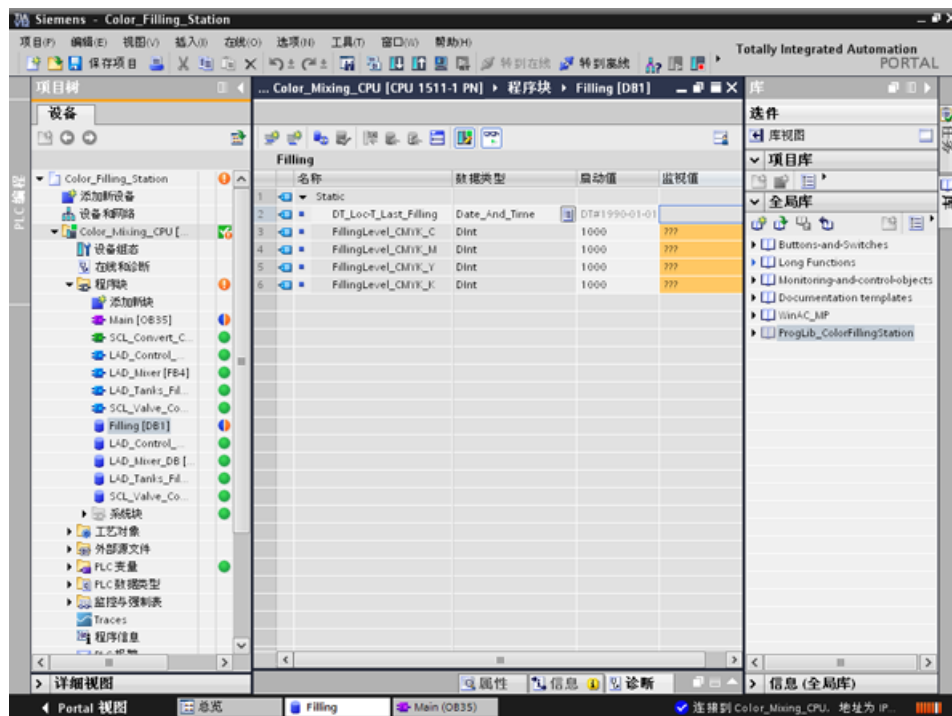
- 无需编写详细的注释。
- 数据访问速度更快。
- 访问数据时不会出错。
- 无需再使用绝对地址。
- 符号到存储器地址的分配由 STEP 7 监视，这意味着在变量的名称或地址更改时，所有使用点都会自动更新。

#### 要求

- 已加载库
- 已编译项目并将其加载到 CPU 中

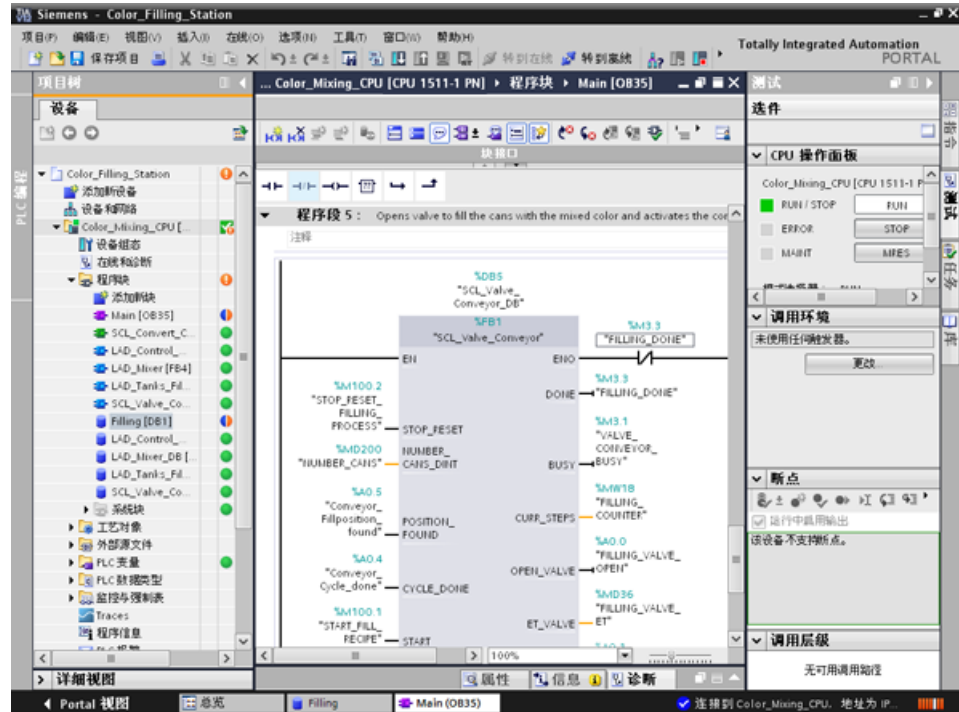
## 步骤

1. 打开“Filling”数据块和“Main”程序块。
2. 为“Main”程序块启用“启用/禁用监视”(Monitoring on/off) 功能。
3. 在“Main”程序块中，在 3 程序段中单击右键打开“FILLING' FillingLevel\_CMYK\_C”的快捷菜单，并选择“修改 > 修改操作数”(Modify > Modify operand)。
4. 输入一个新值并单击“确定”(OK)。
5. 在数据块“Filling”数据块中启用“无需重新初始化的下载”(Download without reinitialization) 功能和“全部监视”(Monitor all) 功能。
6. 创建一个名为“DT\_Loc-T\_Last\_Filling”的新参数并选择“Date\_And\_Time”数据类型。



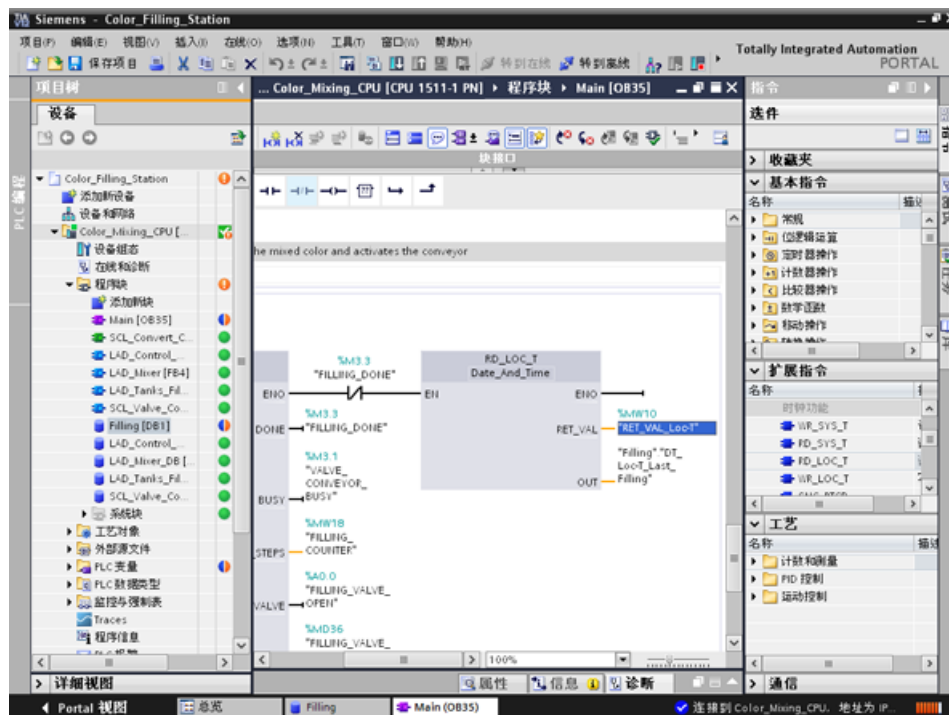
7. 在 5

程序段的“Main”程序块中插入一个常闭触点，并将其与“FILLING\_DONE”参数互连。



8. 从“指令”(Instructions) 选项卡中打开“日期和时间”(Date & time) 文件夹，并在“Main”程序块中插入块“RD\_Loc\_T”。

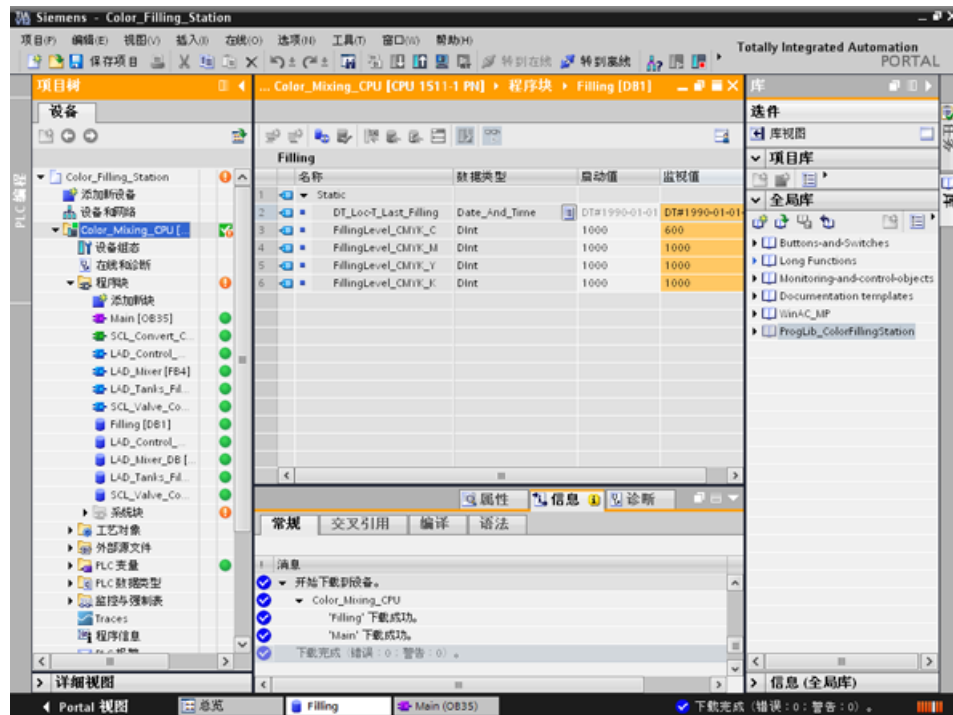
- 将“OUT”输出与“DT\_Loc-T\_Last\_Filling”参数互连，并将“RED\_VAL”输出与新创建的“RED\_VAL\_Loc-T”参数互连。将“LAD\_Tanks\_Filling\_Process”数据块用作“RED\_VAL\_Loc-T”参数的存储位置。



- 编译并下载该项目。

## 结果

重新加载了上一次填充的日期和时间。未覆盖数据块“Filling”的实际参数。



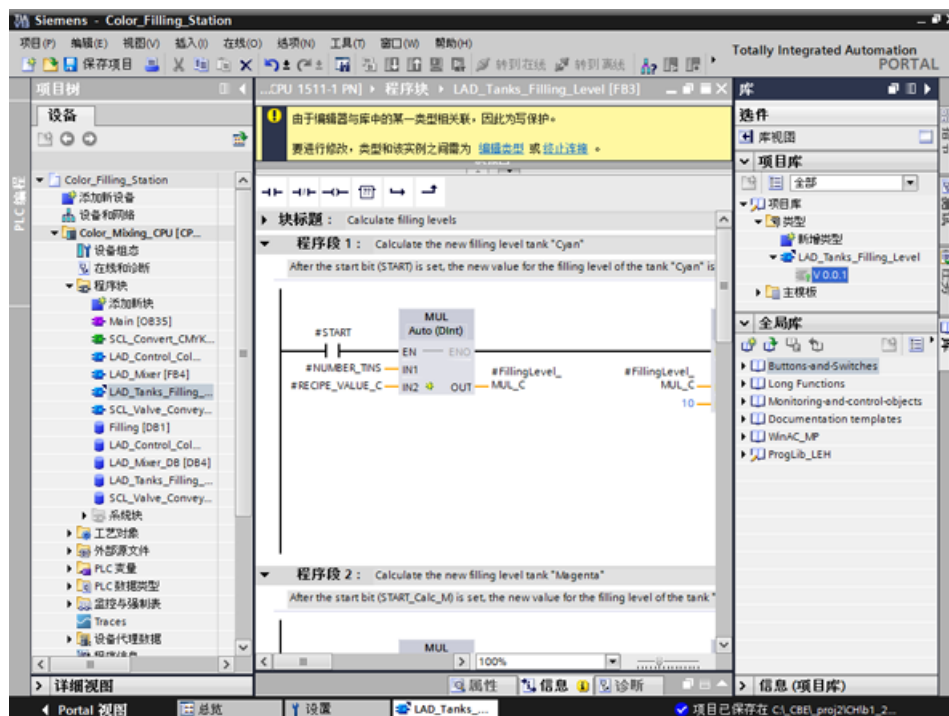
## 3.2.9 块的版本控制

### 简介

块类型的使用可以确保项目中具有较高标准化程度。您可以方便地将对块类型的功能扩展到现有项目中。通过版本控制，可确保进行更改跟踪。在本例中，我们将创建一个“LAD\_Tanks\_Filling”块以作为项目库中的类型。作为一项功能扩展，请将用于液位计算的三个指令替换为 **CalculateBox**，后者将执行所有算术功能。这种优化意味着所需的临时变量更少，从而不必在采用不同编程语言编写的块之间进行切换。

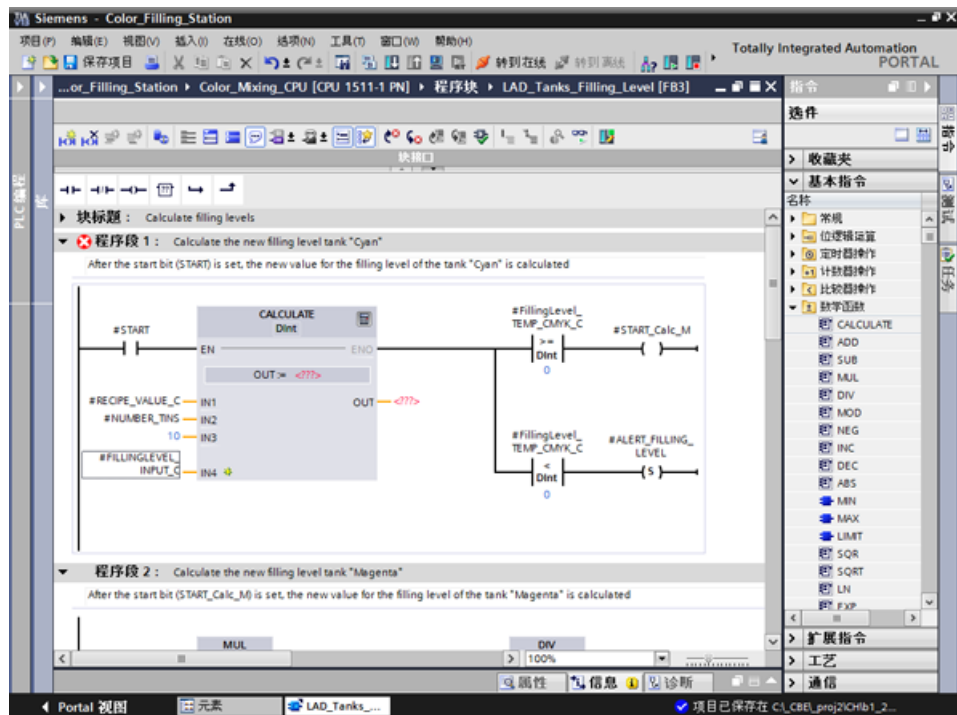
步骤

11. 编译“LAD\_Tanks\_Filling”块，然后将其插到项目库中“Types”的下面。
12. 使用“编辑类型”(Edit type) 创建新的块版本。

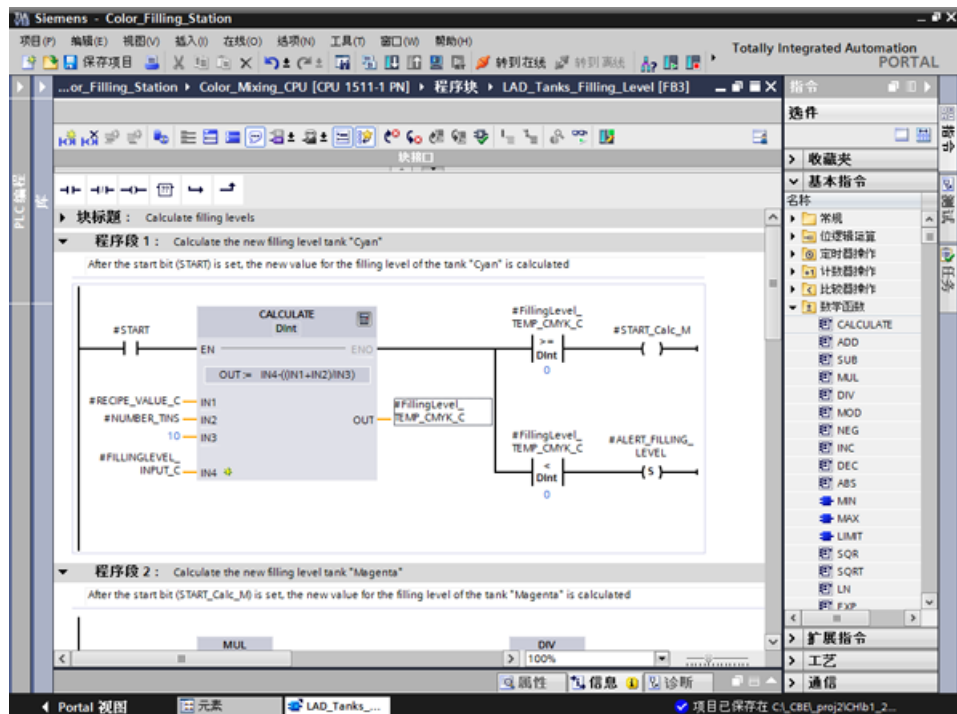


13. 从“基本指令 > 算术函数”(Basic instructions > Mathematical functions) 插入 CALCULATE 指令。
14. 从该块删除 MUL、DIV 和 SUB 指令。

15. 将两个输入插入到 CALCULATE 指令中，然后将这两个输入互连。



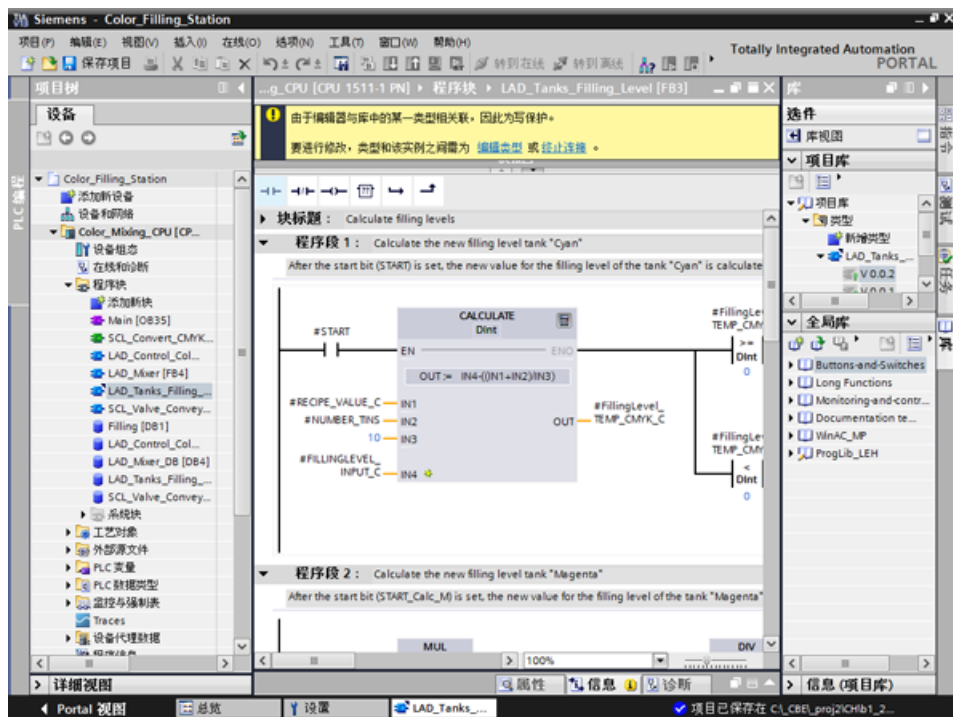
16. 定义计算公式，然后互连输出。



17. 发布块版本。

### 结果

该块类型的修订版本以新版本号保存在库中。





### 3.2.10 设置保持性

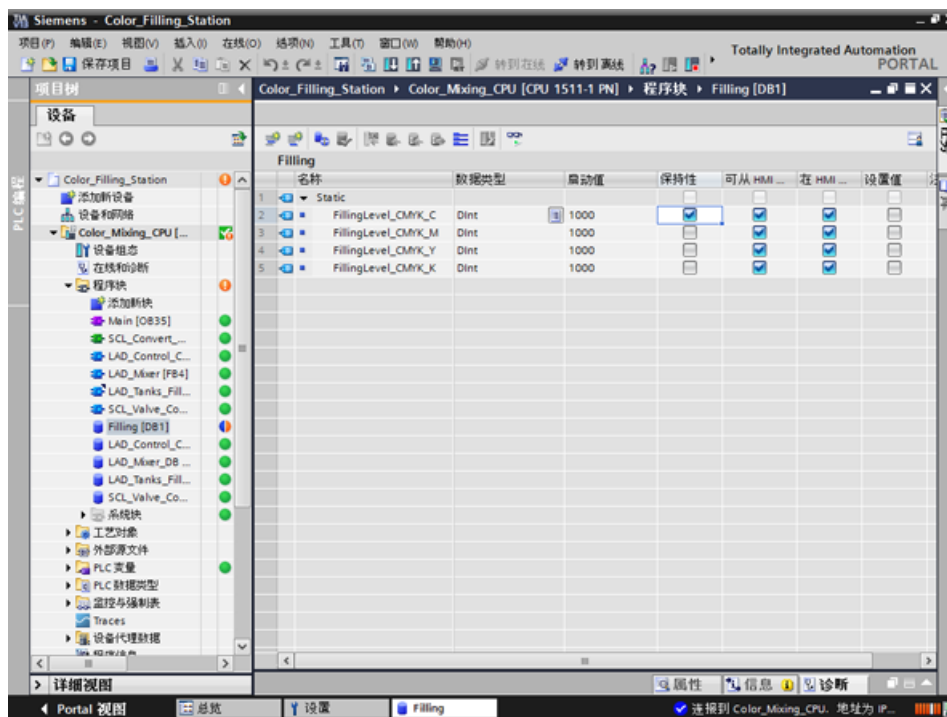
#### 简介

所有变量都使用组态的起始值，在 CPU 启动期间（例如，在发生电源故障后）进行初始化。中断之前变量所具有的最新值将被初始值覆盖。为避免这种情况，请将变量定义为保持性变量。保持性变量即使在重启后也会保留其值。

在本例中，油漆储罐的液位保留在 CPU 的保持性存储器区域中。

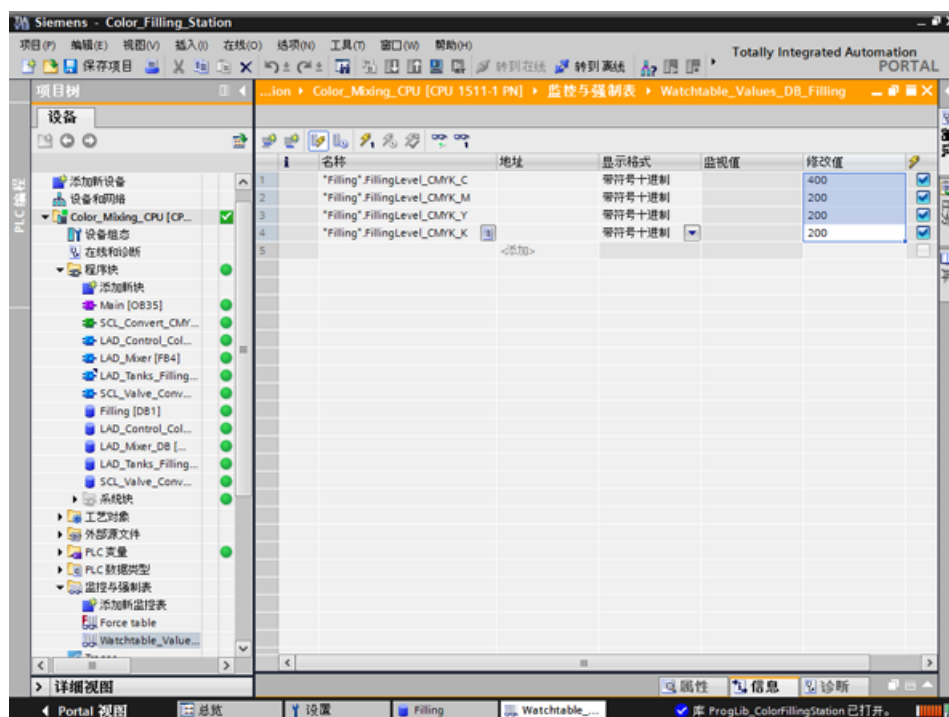
#### 步骤

1. 在线连接到 CPU。
2. 启用“Filling”数据块中的“Cyan”条目。



3. 将更改内容加载到 CPU。
4. 将“Watchtable”对象从库拖到项目中。  
该对象包含灌注液位变量（包括一个控制值）。

5. 使用“立即修改”(Modify now) 将控制值传输至 CPU。



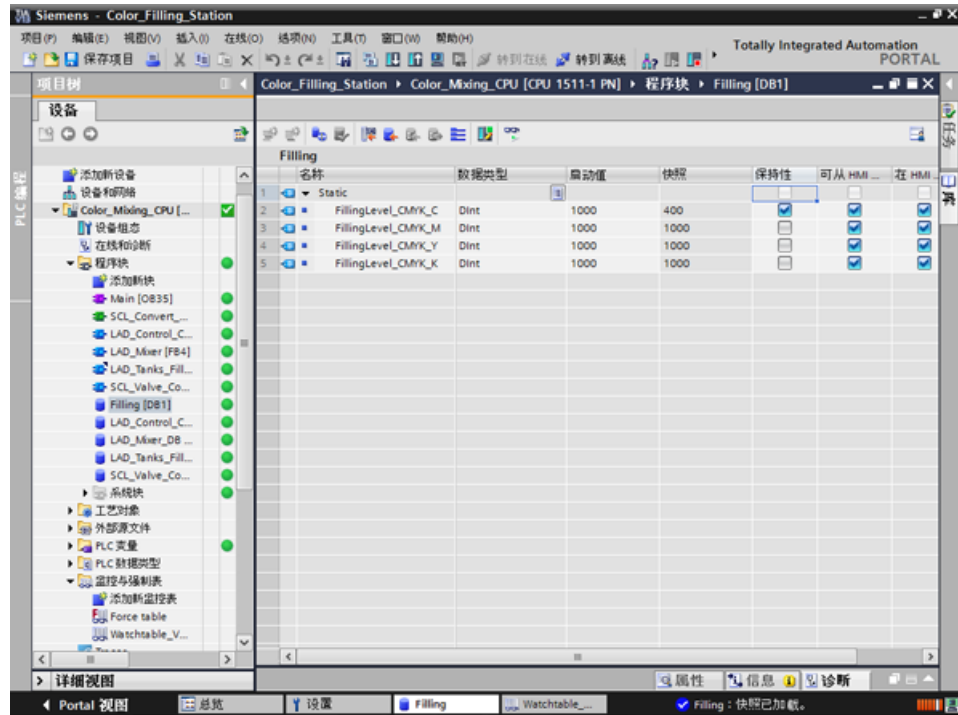
6. 断开与 CPU 的在线连接。若要仿真电源故障，请断开 CPU 的电源。

7. 重新连接电源，然后与 CPU 进行在线连接。

针对“Filling”数据块，启用“全部监视”(Monitor all)。

## 结果

从保持性存储器区域读取“Cyan”的灌注液位。  
所有其它灌注液位都使用其初始值进行初始化。



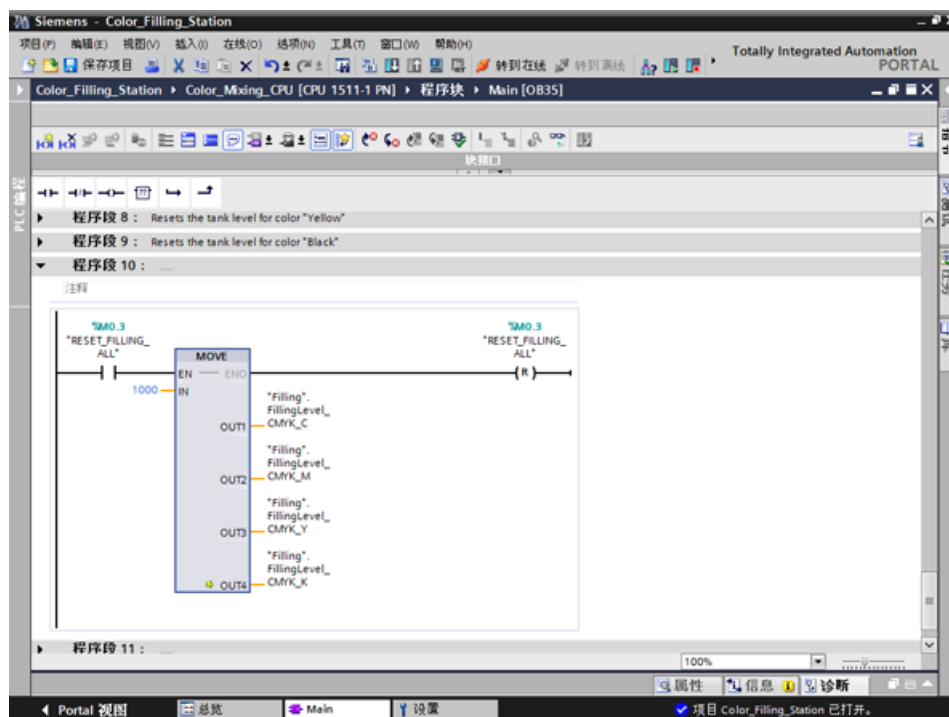
### 3.2.11 激活 EN/ENO 机制

#### 简介

通过各种指令中的 EN/ENO 机制，可以检测运行错误，并避免程序崩溃。默认情况下，新插入的 ENO 指令已被禁用。随后可以激活 ENO 使能输出。在已将所有油漆储罐的灌装液位复位到起始值 (1000) 的新程序段中，可以同时使用此输出。

#### 步骤

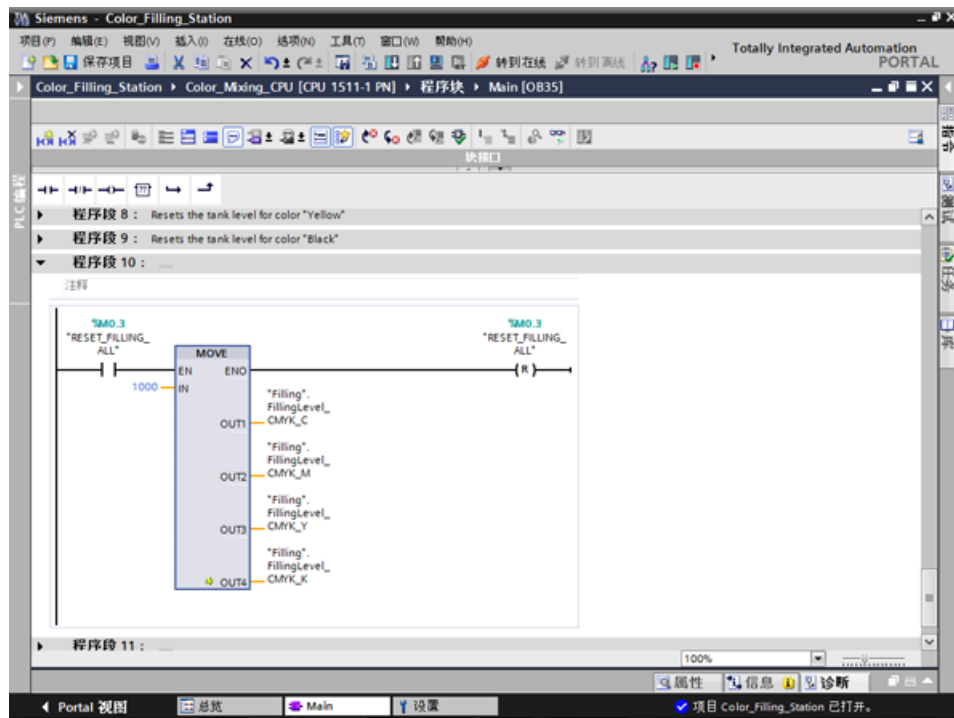
1. 打开 Main[OB35] 程序块，然后将 MOVE 指令插到程序段 10 中。
2. 将该指令扩展到总共四个输出。
3. 在 MOVE 指令的前面插入一个常开触点。
4. 在 MOVE 指令的后面插入一个复位线圈。
5. 将 MOVE 指令的输入和输出互连。



6. 使用 ENO 快捷菜单生成该指令。

## 结果

为该块互连了 EN/ENO 机制。如果执行期间没有任何错误，则 ENO 使能输出的信号状态将为“1”。如果执行期间发生错误，则 ENO 使能输出的信号状态将为“0”。



### 3.2.12 使用注释功能

#### 简介

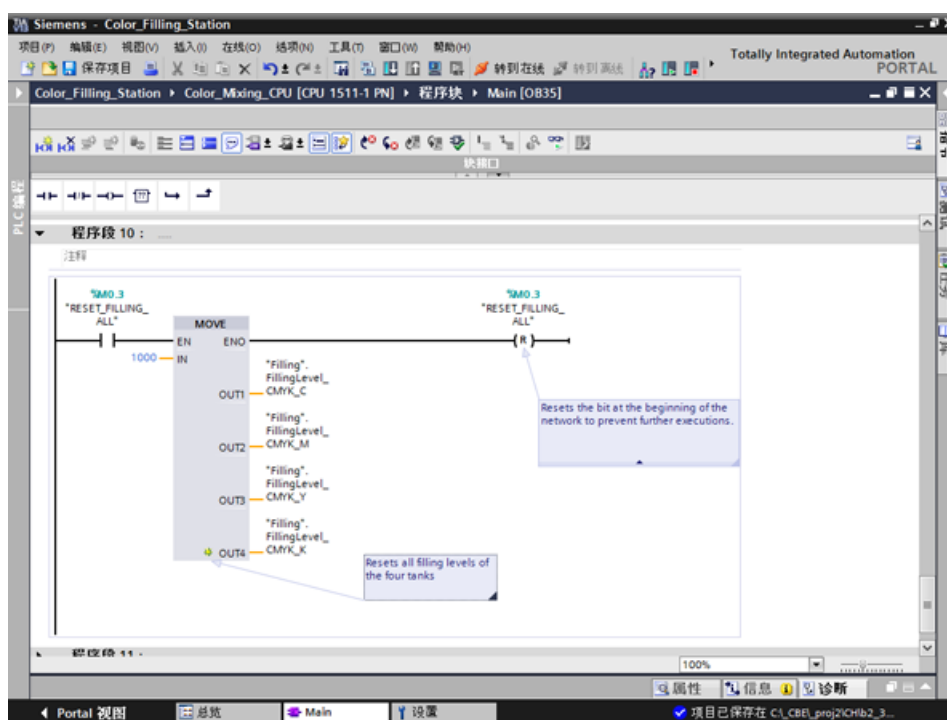
MOVE 和 Reset 指令可通过详细注释来扩展。

#### 步骤

1. 使用快捷菜单插入注释。
2. 输入注释文本。

#### 结果

输入了指令和线圈的注释。

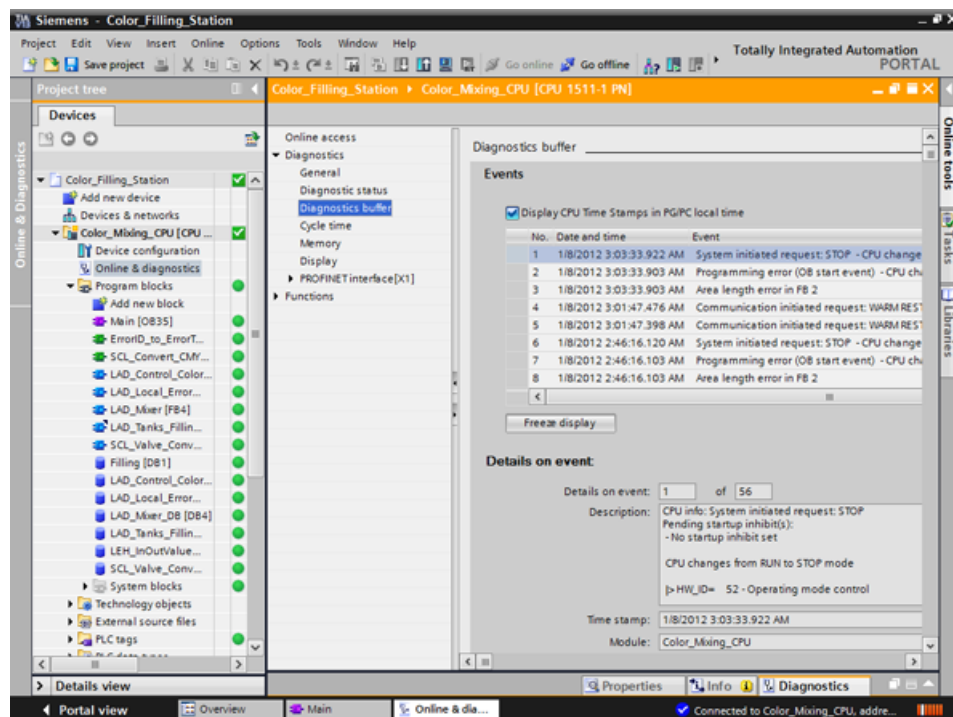


### 3.2.13 局部错误处理

#### 3.2.13.1 在块内处理错误

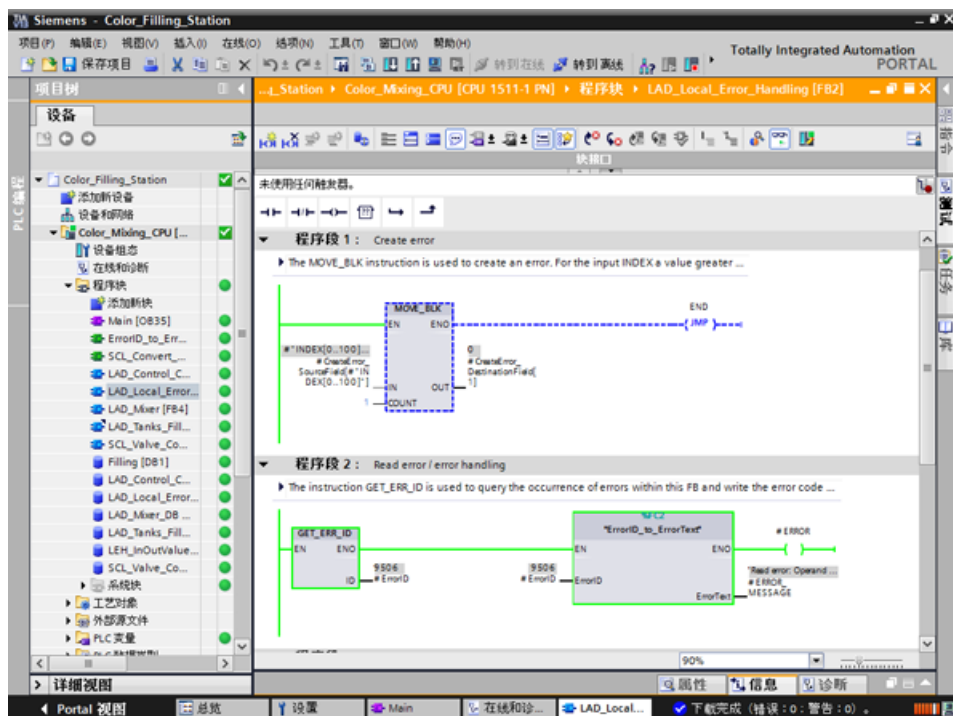
##### 步骤

与 S7-300/400 的 CPU 不同，S7-1500 的 CPU 会在发生非常罕见的错误时转入 STOP 模式。如果发生了错误，就会将错误输入到 CPU 的诊断缓冲区中。通过在每个块上使用局部错误处理，可以避免 CPU 停止。最好在用户程序的开发过程中启用局部错误处理。



### 3.2 创建程序

这样就可以对信息进行精确评估，例如，使用 STL/FBD/LAD 和 SCL 程序对块中的错误处理进行编程。块会生成由“GET\_ERROR\_ID”指令进行评估的错误 ID。可以在 MAIN 块中以及在函数块中调用“GET\_ERROR\_ID”指令。CPU 保留在 RUN 模式。





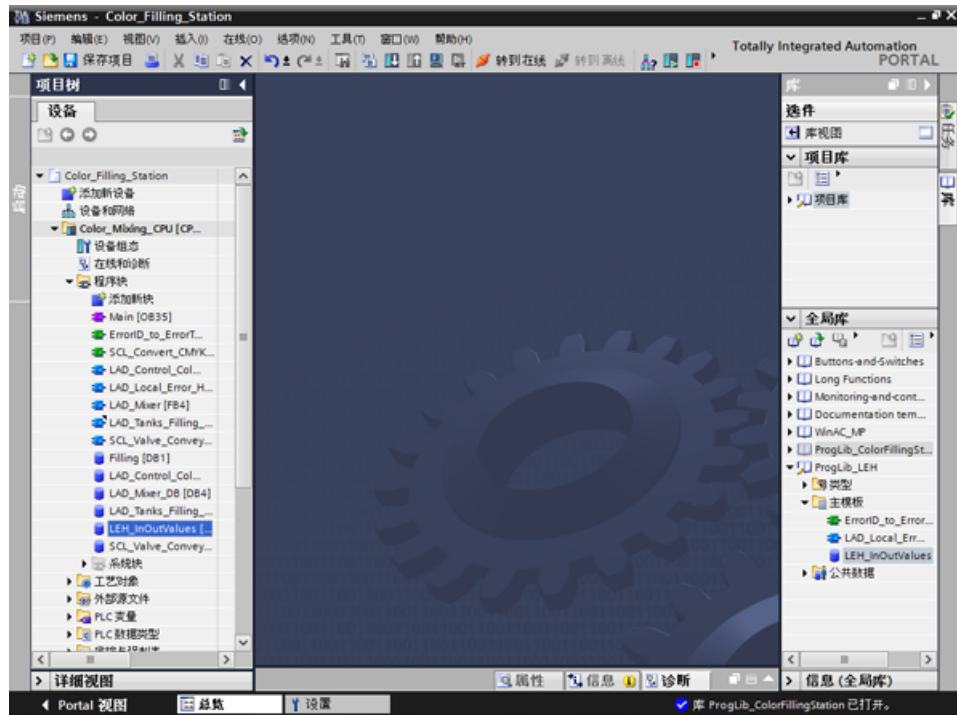
### 3.2.13.2 加载用于局部错误处理的块

#### 简介

为了说明局部错误处理，可在项目中加载“ProgLib\_LEH”库的块。这些块仅用于说明局部错误处理，不在项目中另外使用。

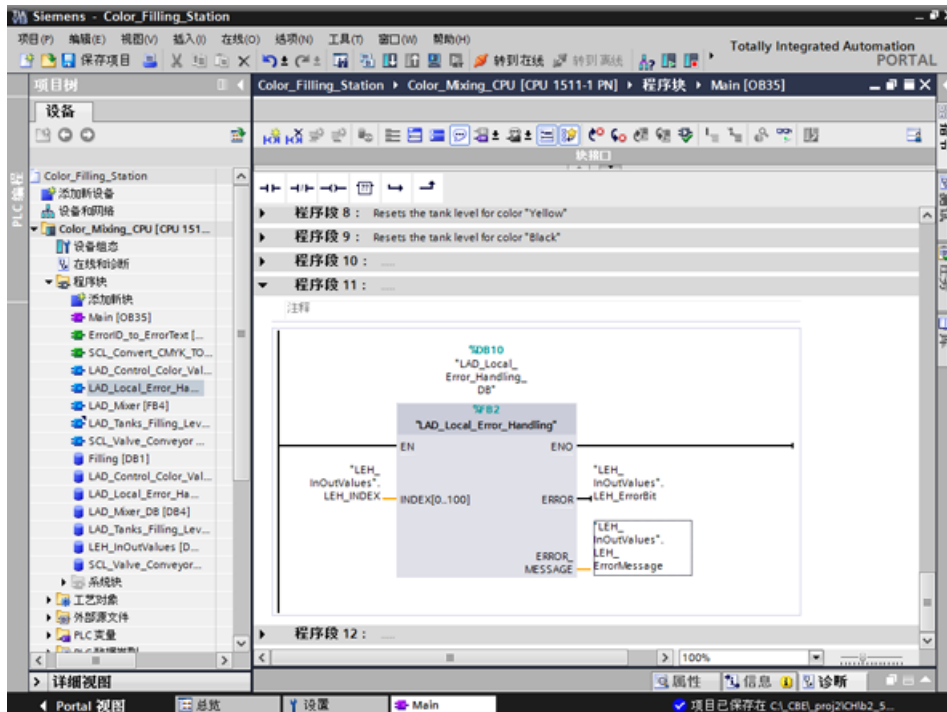
#### 步骤

1. 打开全局库“ProgLib\_LEH”。
2. 将主模板中的块复制到项目中。



3. 在“Main”块的一个空程序段中调用“LAD\_Local\_Error\_Handling”函数块。

4. 将“LAD\_Local\_Error\_Handling”函数块的参数与“LEH\_InOutValues”数据块的变量互连



5. 在线连接到 CPU。

6. 对更改内容进行编译并加载到 CPU。

## 结果

使用“INDEX[0..100]”输入参数处的“LEH\_INDEX”变量随后触发一个编程错误。例如，如果将该输入参数设置为“101”，则会通过输出参数报告错误。

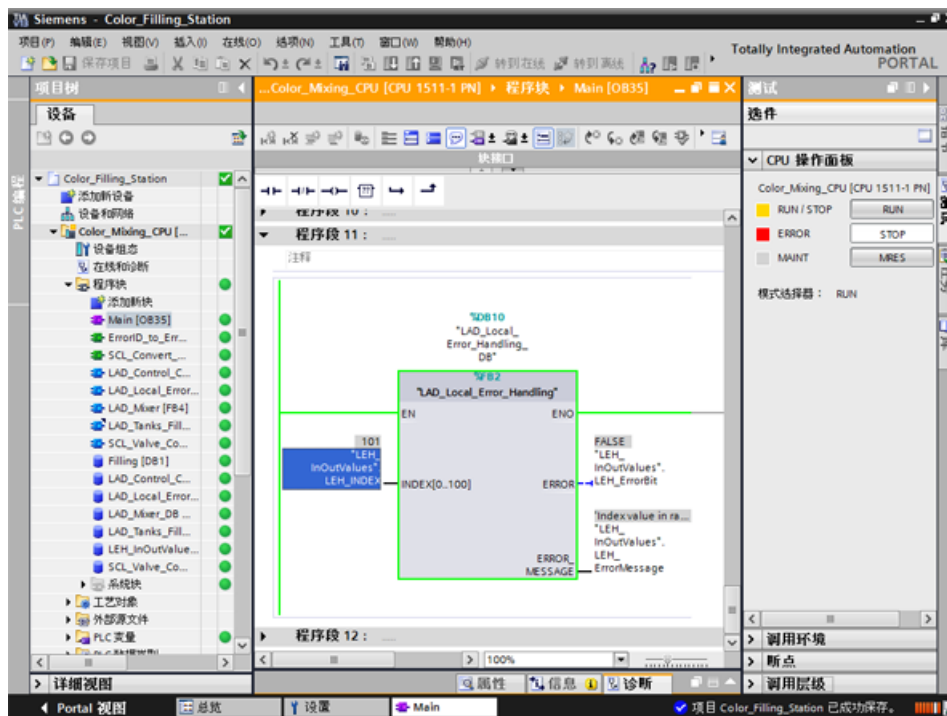
### 3.2.13.3 不进行局部错误处理时生成错误

#### 简介

为了在不使用局部错误处理或不创建相应 OB 的情况下触发编程错误，请执行以下步骤。

#### 步骤

1. 激活“监视”(Monitor) 功能。
2. 将“LEH\_INDEX”变量的值设置为一个无效的值，如“101”。在“测试”(Testing) 对话框中，ERROR LED 指示灯短暂闪烁，CPU 从 RUN 转为 STOP。



3. 切换到诊断缓冲区。错误和错误响应显示在诊断缓冲区中。
4. 将 CPU 设置回 RUN。

#### 结果

从 RUN 状态转换到 STOP 状态会将“LEH\_INDEX”变量复位到起始值“0”。这会 自动将问题解决。

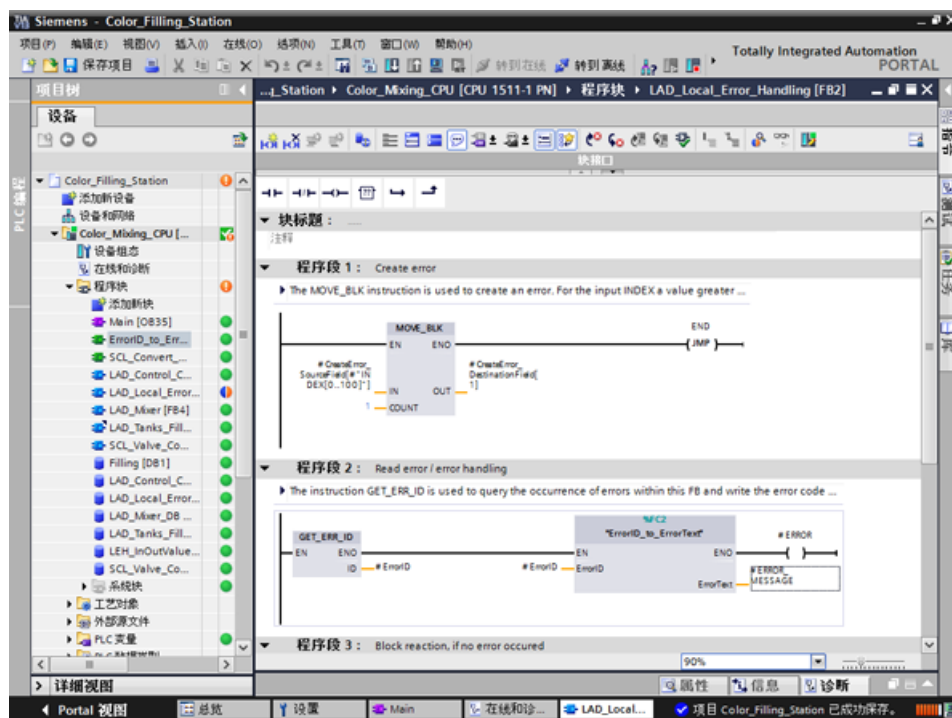
## 3.2.13.4 进行局部错误处理时生成错误

## 简介

为了通过错误消息对错误做出响应，请执行以下步骤将“GET\_ERR\_ID”指令及其 ENO 位用于局部错误处理。这意味着 CPU 保持在 RUN 模式。

## 步骤

1. 打开“LAD\_Local\_Error\_Handling”函数块。
2. 在第二个程序段中插入“GET\_ERR\_ID”指令，然后连接“ID”输出。

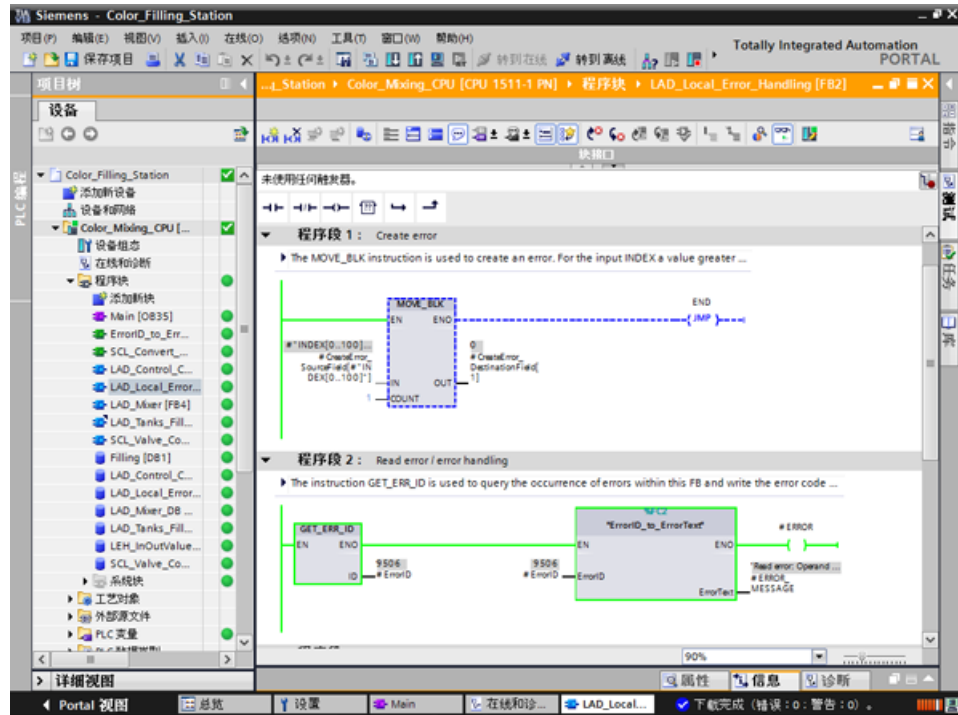


3. 从项目树调用“ErrorID\_to\_ErrorText”函数。
4. 将“ErrorID\_to\_ErrorText”函数的参数进行互连，以便它们可以将错误代码转换为错误消息。
5. 将更改内容加载到 CPU。
6. 通过输入一个无效的值（如“101”），在“Main”组织块中触发错误。通过“ERROR\_MESSAGE”参数输出一个错误消息。

## 结果

只要未将错误纠正，就会输出错误消息。

若要纠正该错误，请为“LEH\_INDEX”变量分配一个有效的值，或重新启动 CPU。



## 3.3 组态可视化

### 3.3.1 现有样本项目

#### 应用样本项目

为了通过 TIA Portal 组态颜色混合系统，应创建样本项目“Color\_Filling\_Station”。样本项目中已存在下列项目组件：CPU 用户程序的程序块和变量表及带有 HMI 画面、HMI 变量和脚本的已组态精智面板。

在本部分中，将详细说明样本项目中各项目组件间的关系。以后用户可自行执行必需的组态步骤。

### 3.3.2 HMI 组态

#### 3.3.2.1 概述

#### HMI 组态简介

所提供项目的“全局库”中包含已编程 CPU 和预组态 HMI 设备。

#### HMI 组态

本部分将介绍 HMI 设备和 HMI 组态。

#### 更多信息

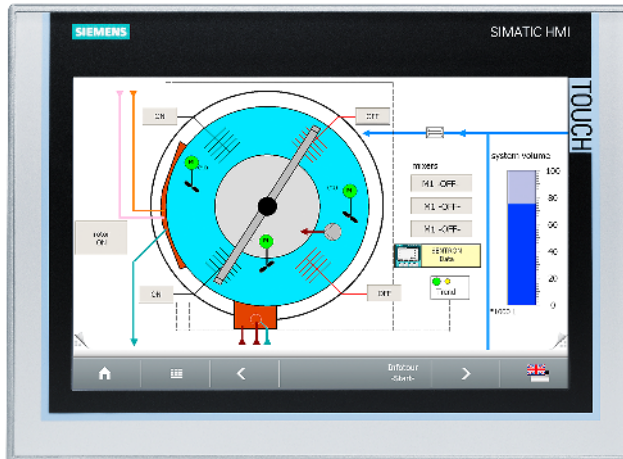
有关 HMI 组态的详细指示信息，请参见：

WinCC V13 精智面板入门指南和 Runtime Advanced

### 3.3.2.2 SIMATIC HMI 精智面板

#### SIMATIC HMI 精智面板

在此，将使用精智面板系列的 TP1200 Comfort HMI 设备对颜料混合系统进行控制。



精智面板非常适用于执行 PROFINET 和 PROFIBUS 系统中较为复杂的 HMI 任务，并具有以下特性：

- 外壳坚固并具有大量接口
- 工业级的宽屏显示不但具有较大的显示视角，而且画面显示高度稳定同时亮度也达到最高。
- 不但可以进行水平安装，也可以进行垂直安装
- 集成有系统诊断查看器，可进行精确诊断

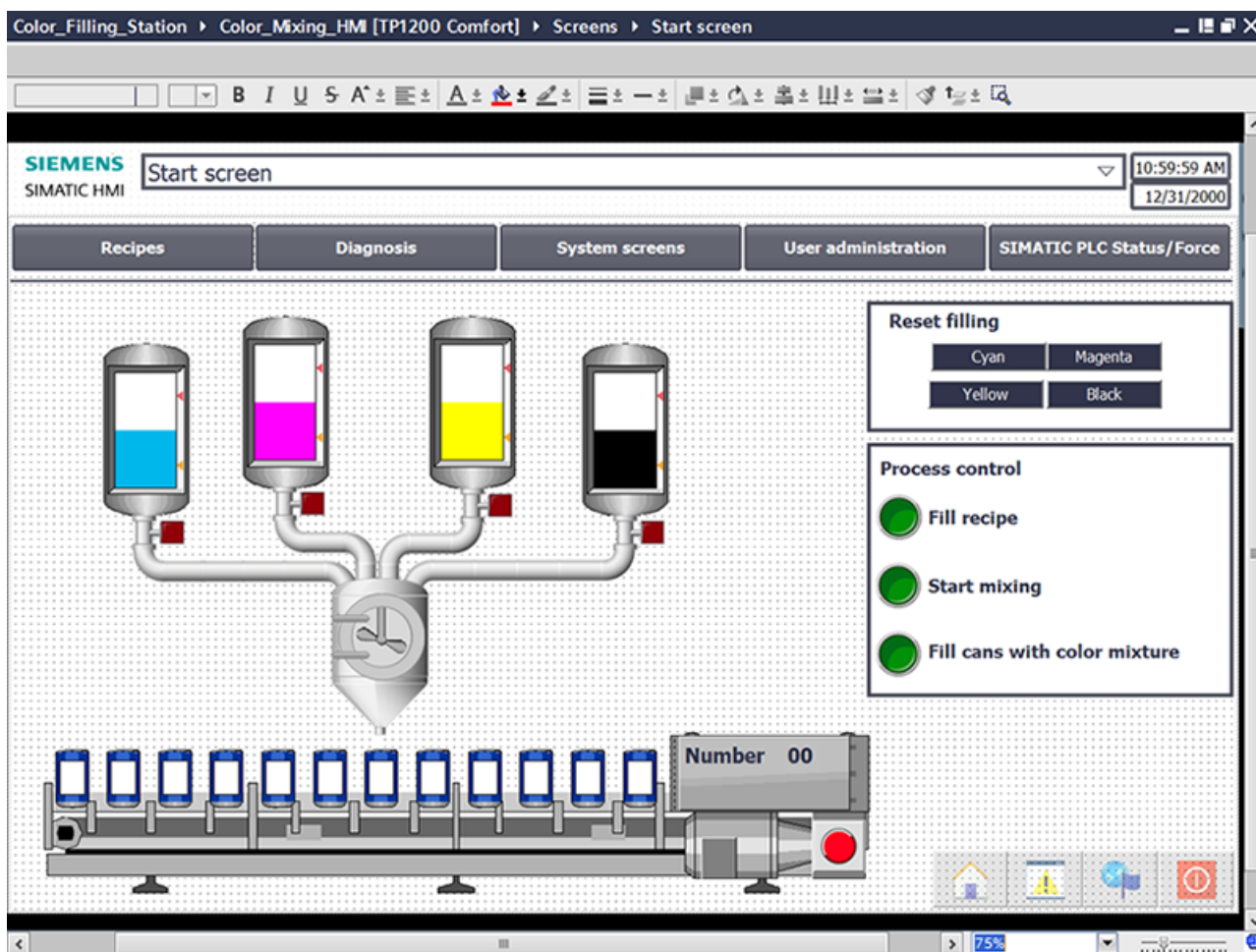
### 3.3.2.3 HMI 画面

#### HMI 画面

使用载入各 HMI 设备的画面运行和监视运行系统中的机器和设备。

可在 WinCC 项目导航的“画面”(Screens) 下管理此画面。

HMI 设备的启动画面用来可视化颜色混合系统以及最重要的状态信息及键图。



颜色混合系统包含下列元素：

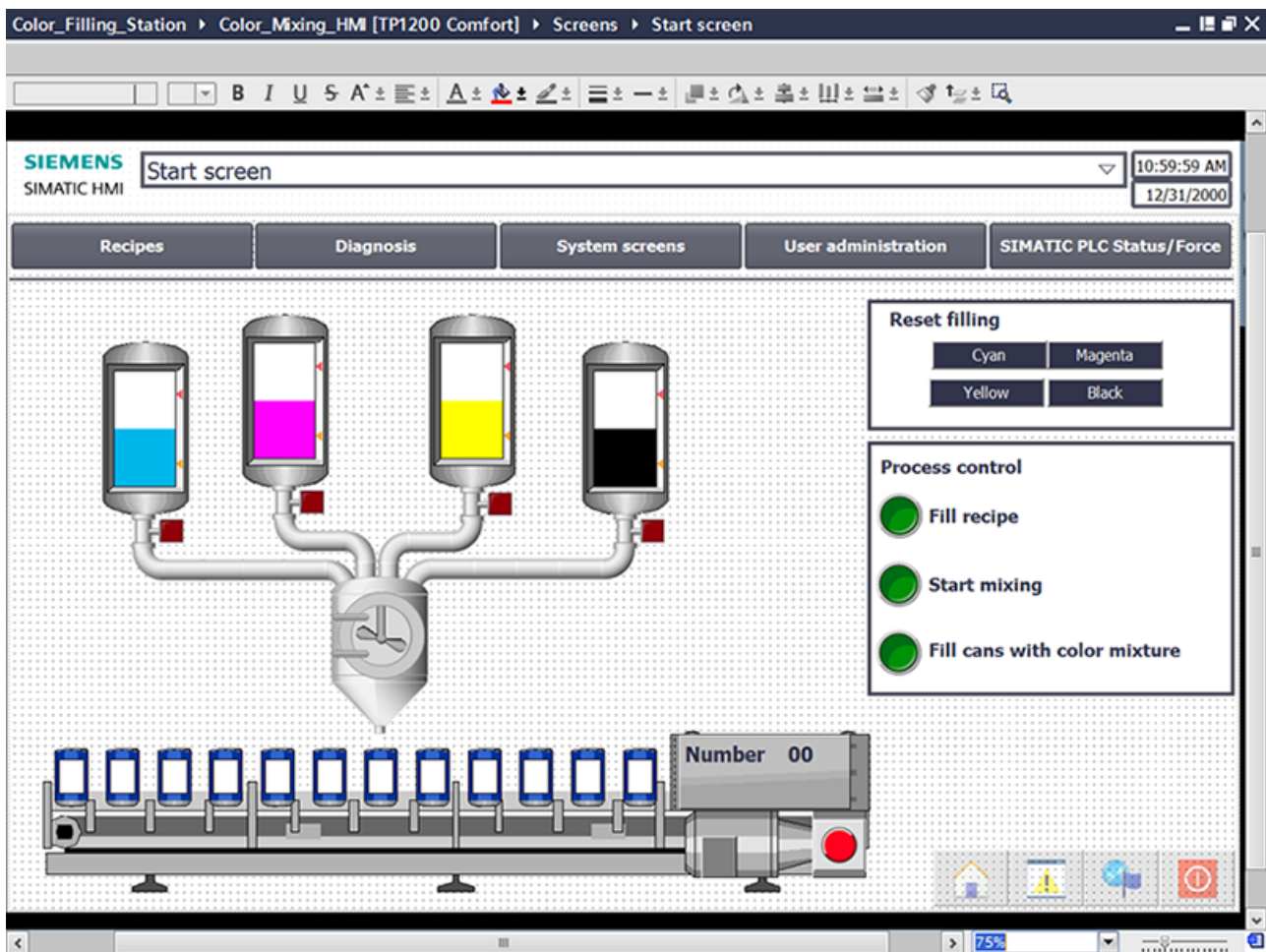
- 带填充量显示的每种打印颜色的颜色存贮罐
- 搅拌器
- 搅拌器的进料器管道
- 带紧急停止开关的传送带



### 3.3.2.4 其它控制元件

#### 其它控制元件

过程步骤“混合颜色”和“填充颜色”将以带动态可视对象的动画形式显示。



示例项目的启动画面包含其它控制对象：

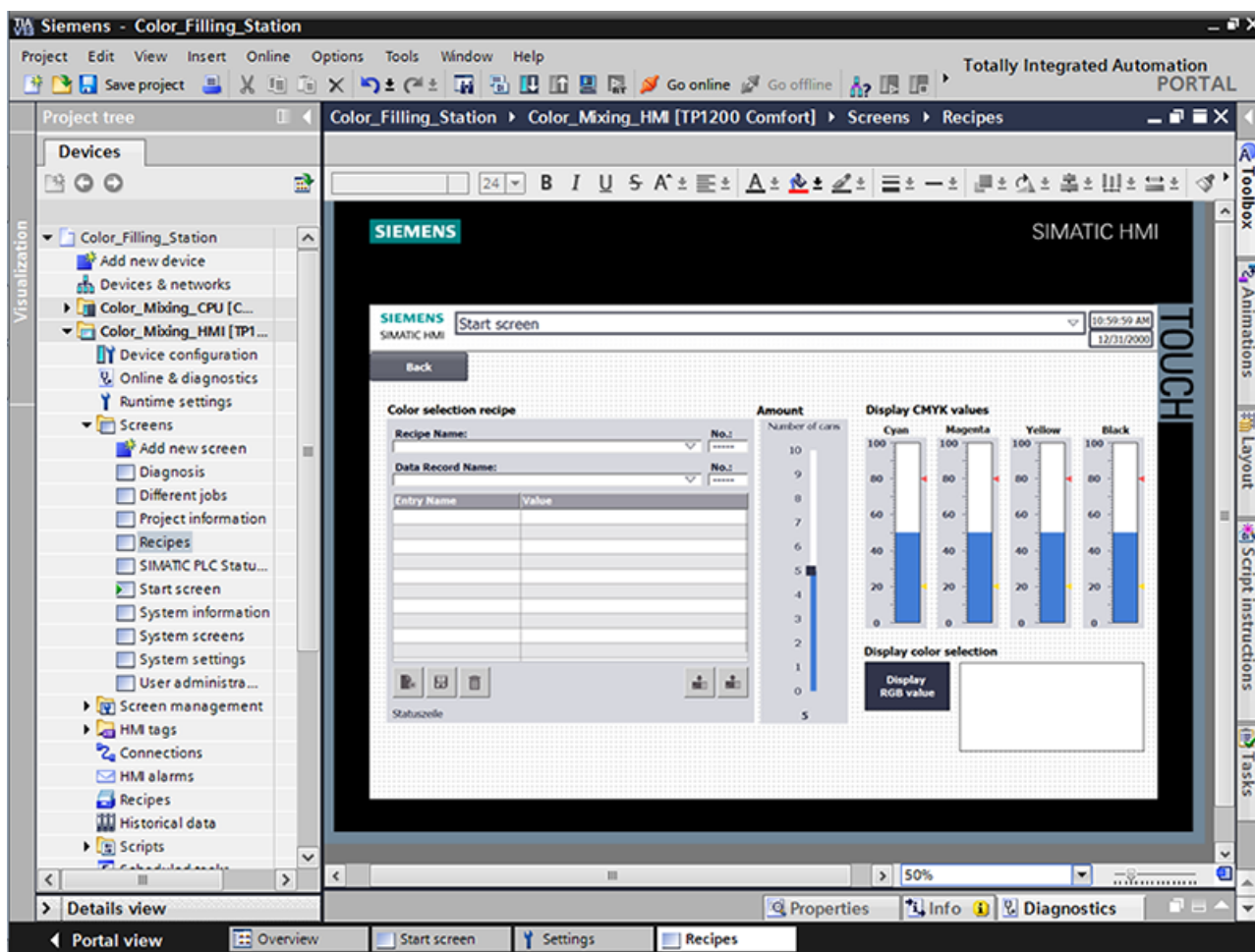
- 画面更改按钮
- 重置填充量按钮
- 用于运行和监视系统的按钮：填充配方、启动混合过程和填充颜色混合物

### 3.3.2.5 配方

#### 配方

配方包含相关生产参数，例如混合比率。

例如，可通过一个步骤将所需的混合比率从 HMI 设备传送到颜色混合系统，以将生产从深橙色切换为黄色。



颜色混合系统可生成混合颜色“橙色”、“琥珀色”、“绿色”和“红色”。

为每种颜色创建一个配方数据记录。

配方数据记录包含产生相应混合颜色的基本色百分比。

配方包含相关参数和用于存储各种色调混合比率的配方数据记录。

### 3.3.2.6 归档

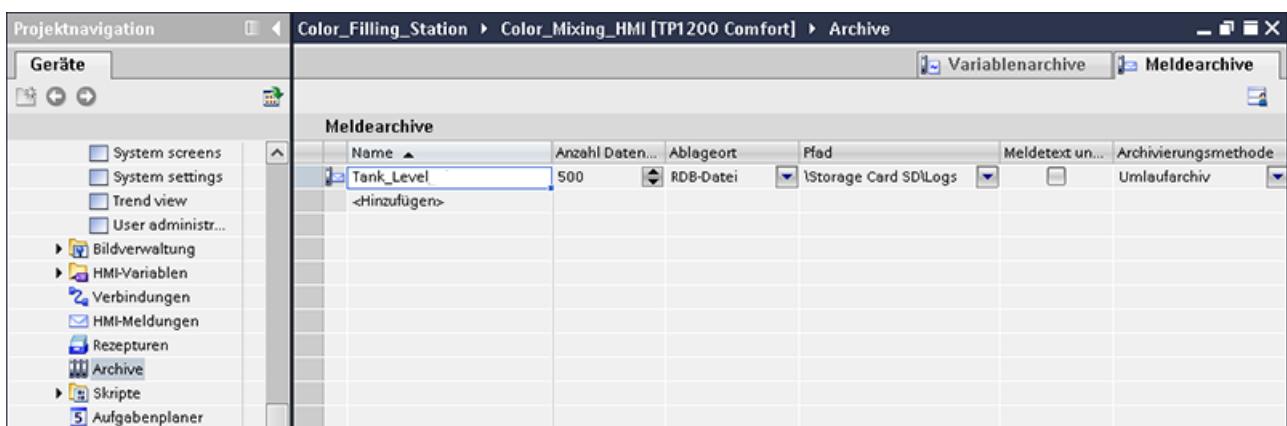
#### 归档

要记录系统的运行事件，应将生产期间生成的报警和过程值保存到日志中。

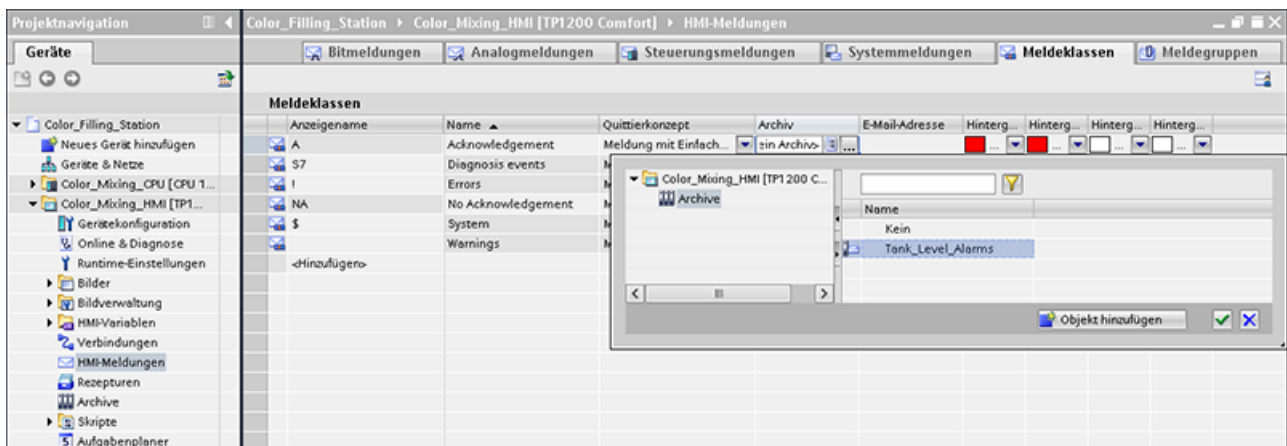
然后评估报警和过程数据日志。

记录颜色混合系统的颜色存储罐填充量。

为此，组态了报警日志“Tank\_Level”。



此日志存储轮班期间填充量过低和填充量过高的报警。



### 3.3.2.7 用户自定义函数

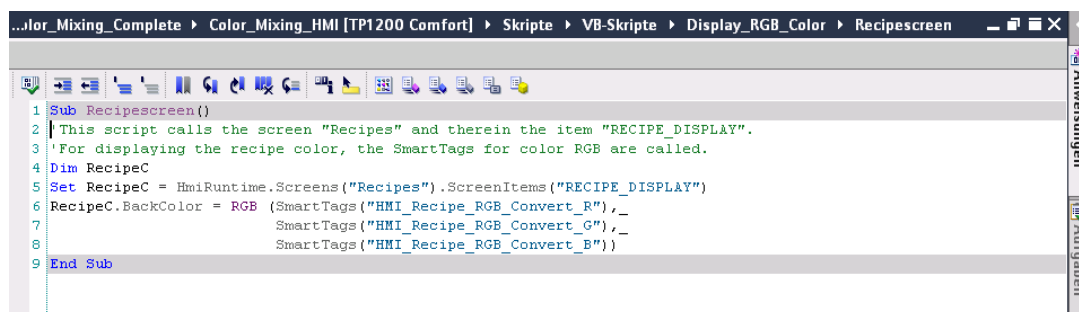
#### 脚本

使用用户自定义函数编程 HMI 设备的其它功能。

WinCC 提供 VBS 编程接口来创建用户自定义函数。

此样本项目使用两个用户自定义函数在监视器的不同画面中显示混合打印颜色。

- “Recipescreen”在“配方”(Recipes) 画面中以所选颜色显示一个矩形。



```
1 Sub Recipescreen()  
2 'This script calls the screen "Recipes" and therein the item "RECIPE_DISPLAY".  
3 'For displaying the recipe color, the SmartTags for color RGB are called.  
4 Dim RecipeC  
5 Set RecipeC = HmiRuntime.Screens("Recipes").ScreenItems("RECIPE_DISPLAY")  
6 RecipeC.BackColor = RGB (SmartTags("HMI_Recipe_RGB_Convert_R"), _  
7 SmartTags("HMI_Recipe_RGB_Convert_G"), _  
8 SmartTags("HMI_Recipe_RGB_Convert_B"))  
9 End Sub
```

- “Startscreen”在启动画面的系统概述中以当前混合色显示已填充容器的标签。

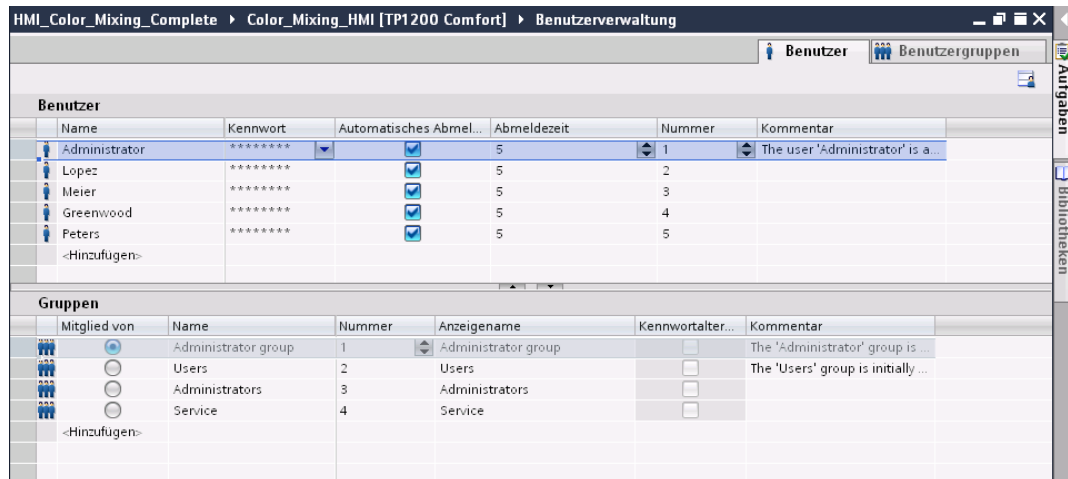
```
...Color_Mixing_Complete > Color_Mixing_HMI [TP1200 Comfort] > Skripte > VB-Skripte > Display_RGB_Color > Startscreen  
1 Sub Startscreen()  
2 |This Script calls screens and their items.  
3 |For displaying the recipe color, the SmartTags for color RGB are called.  
4  
5 Dim Label_0  
6 Set Label_0 = HmiRuntime.Screens("Start screen").ScreenItems("Filling")  
7 Label_0.BackColor = RGB (SmartTags("HMI_Recipe_RGB_Convert_R"),SmartTags("HMI_Recipe_RGB_Convert_G"),SmartTags("HMI_Recipe_RGB_Convert_B"))  
8  
9 Dim Label_7  
10 Set Label_7 = HmiRuntime.Screens("Start screen").ScreenItems("TIN_LABEL_25")  
11 Label_7.BackColor = RGB (SmartTags("HMI_Recipe_RGB_Convert_R"),SmartTags("HMI_Recipe_RGB_Convert_G"),SmartTags("HMI_Recipe_RGB_Convert_B"))  
12  
13 Dim Label_8  
14 Set Label_8 = HmiRuntime.Screens("Start screen").ScreenItems("TIN_LABEL_27")  
15 Label_8.BackColor = RGB (SmartTags("HMI_Recipe_RGB_Convert_R"),SmartTags("HMI_Recipe_RGB_Convert_G"),SmartTags("HMI_Recipe_RGB_Convert_B"))  
16  
17 Dim Label_9  
18 Set Label_9 = HmiRuntime.Screens("Start screen").ScreenItems("TIN_LABEL_29")  
19 Label_9.BackColor = RGB (SmartTags("HMI_Recipe_RGB_Convert_R"),SmartTags("HMI_Recipe_RGB_Convert_G"),SmartTags("HMI_Recipe_RGB_Convert_B"))  
20  
21 Dim Label_10  
22 Set Label_10 = HmiRuntime.Screens("Start screen").ScreenItems("TIN_LABEL_31")  
23 Label_10.BackColor = RGB (SmartTags("HMI_Recipe_RGB_Convert_R"),SmartTags("HMI_Recipe_RGB_Convert_G"),SmartTags("HMI_Recipe_RGB_Convert_B"))  
24  
25 Dim Label_11  
26 Set Label_11 = HmiRuntime.Screens("Start screen").ScreenItems("TIN_LABEL_33")  
27 Label_11.BackColor = RGB (SmartTags("HMI_Recipe_RGB_Convert_R"),SmartTags("HMI_Recipe_RGB_Convert_G"),SmartTags("HMI_Recipe_RGB_Convert_B"))  
28  
29 Dim Label_12  
30 Set Label_12 = HmiRuntime.Screens("Start screen").ScreenItems("TIN_LABEL_35")  
31 Label_12.BackColor = RGB (SmartTags("HMI_Recipe_RGB_Convert_R"),SmartTags("HMI_Recipe_RGB_Convert_G"),SmartTags("HMI_Recipe_RGB_Convert_B"))  
32  
33 Dim Label_19  
34 Set Label_19 = HmiRuntime.Screens("Start screen").ScreenItems("TIN_LABEL_24")  
35 Label_19.BackColor = RGB (SmartTags("HMI_Recipe_RGB_Convert_R"),SmartTags("HMI_Recipe_RGB_Convert_G"),SmartTags("HMI_Recipe_RGB_Convert_B"))  
36  
37 Dim Label_20  
38 Set Label_20 = HmiRuntime.Screens("Start screen").ScreenItems("TIN_LABEL_26")  
39 Label_20.BackColor = RGB (SmartTags("HMI_Recipe_RGB_Convert_R"),SmartTags("HMI_Recipe_RGB_Convert_G"),SmartTags("HMI_Recipe_RGB_Convert_B"))  
40  
41 Dim Label_21  
42 Set Label_21 = HmiRuntime.Screens("Start screen").ScreenItems("TIN_LABEL_28")  
43 Label_21.BackColor = RGB (SmartTags("HMI_Recipe_RGB_Convert_R"),SmartTags("HMI_Recipe_RGB_Convert_G"),SmartTags("HMI_Recipe_RGB_Convert_B"))  
44  
45 Dim Label_22
```

### 3.3.2.8 用户管理

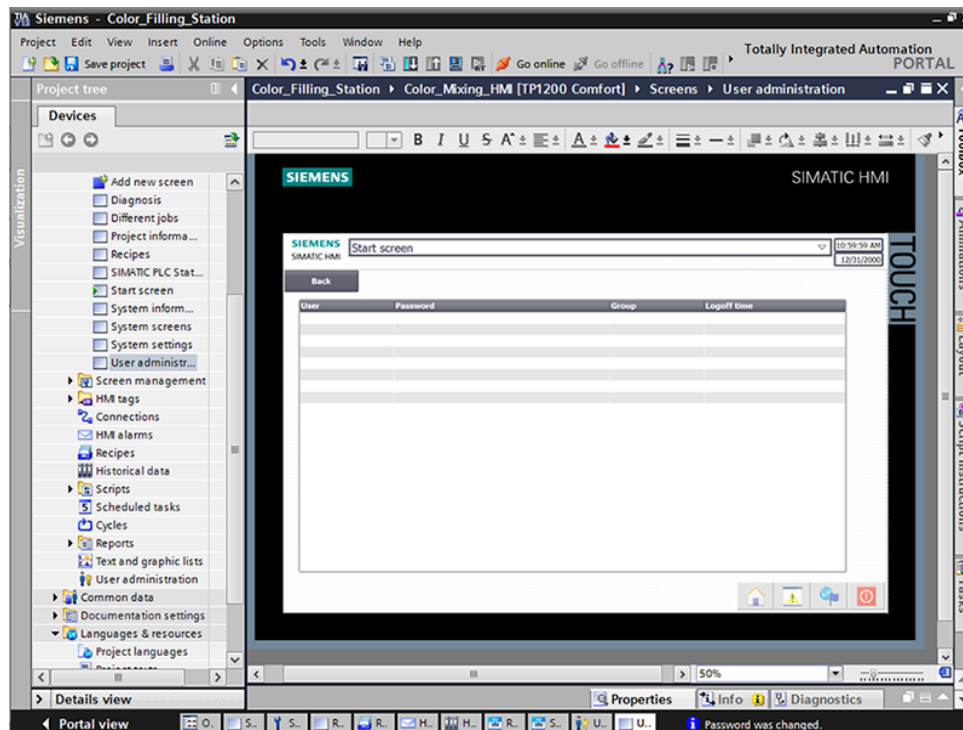
#### 用户管理

##### WinCC

允许用户将安全相关操作限定为特定用户组，并防止数据和函数在运行系统中未经授权的访问。



“用户视图”对象提供对 HMI 设备上的用户和密码的管理功能。



具有用户管理权限的用户可以访问用户视图中的所有功能。他们可以创建和删除用户并更改其密码或其它用户的密码。



### 3.3.2.9 多语言性

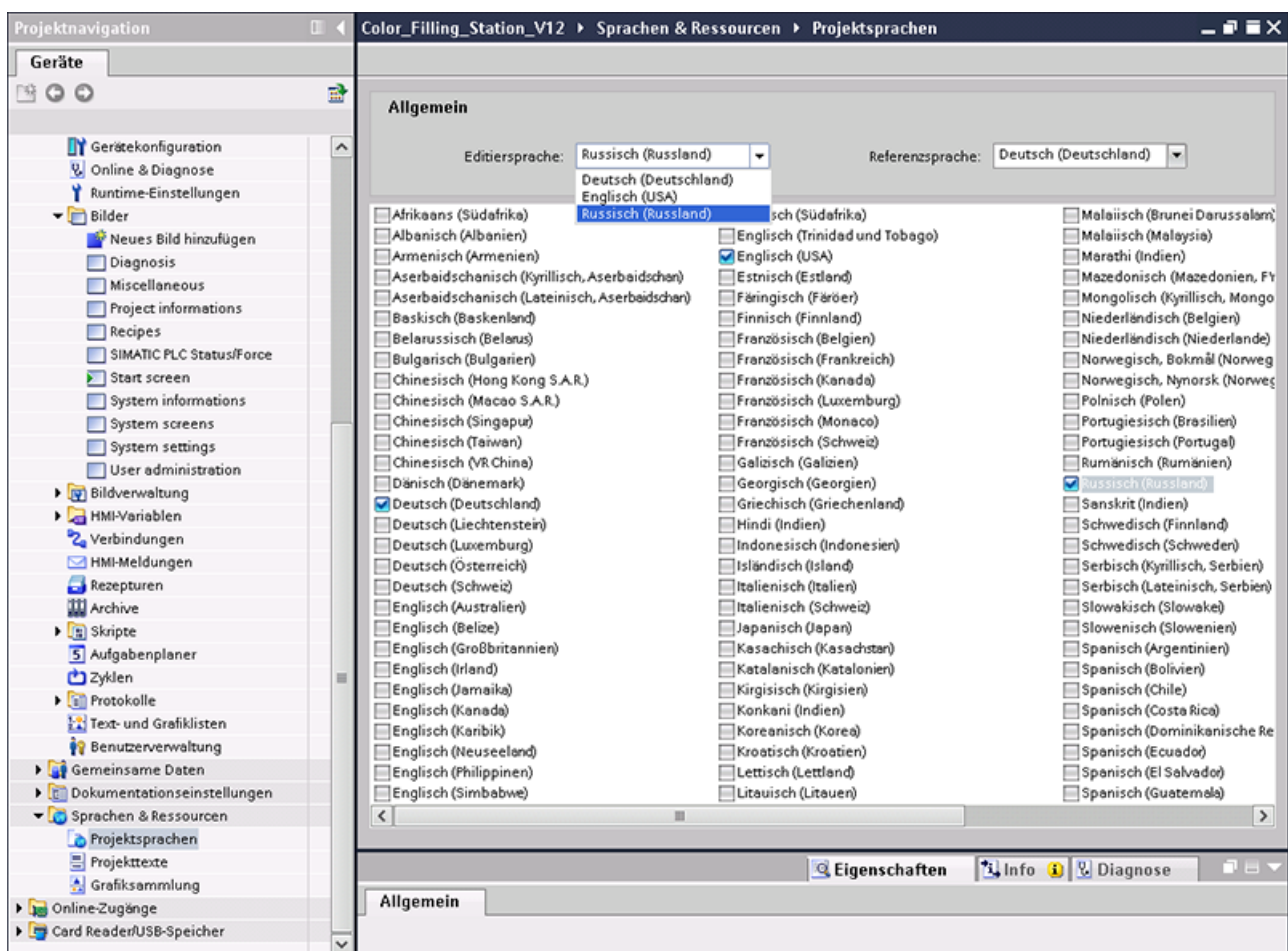
#### 多语言性

WinCC 支持多语言用户界面。

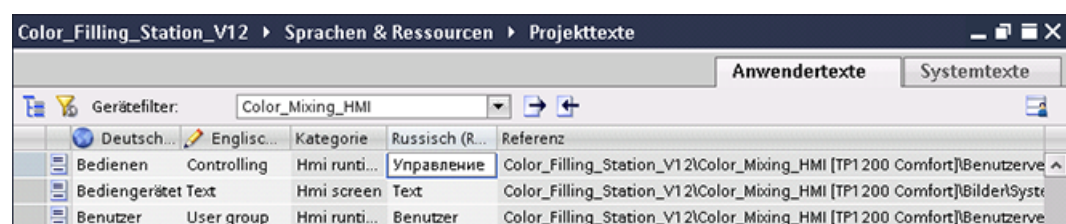
颜色混合系统在位于俄罗斯的新子公司中运作。

维护和维修人员需要俄语用户界面。

为此，该示例项目通过另一语言进行了扩展。

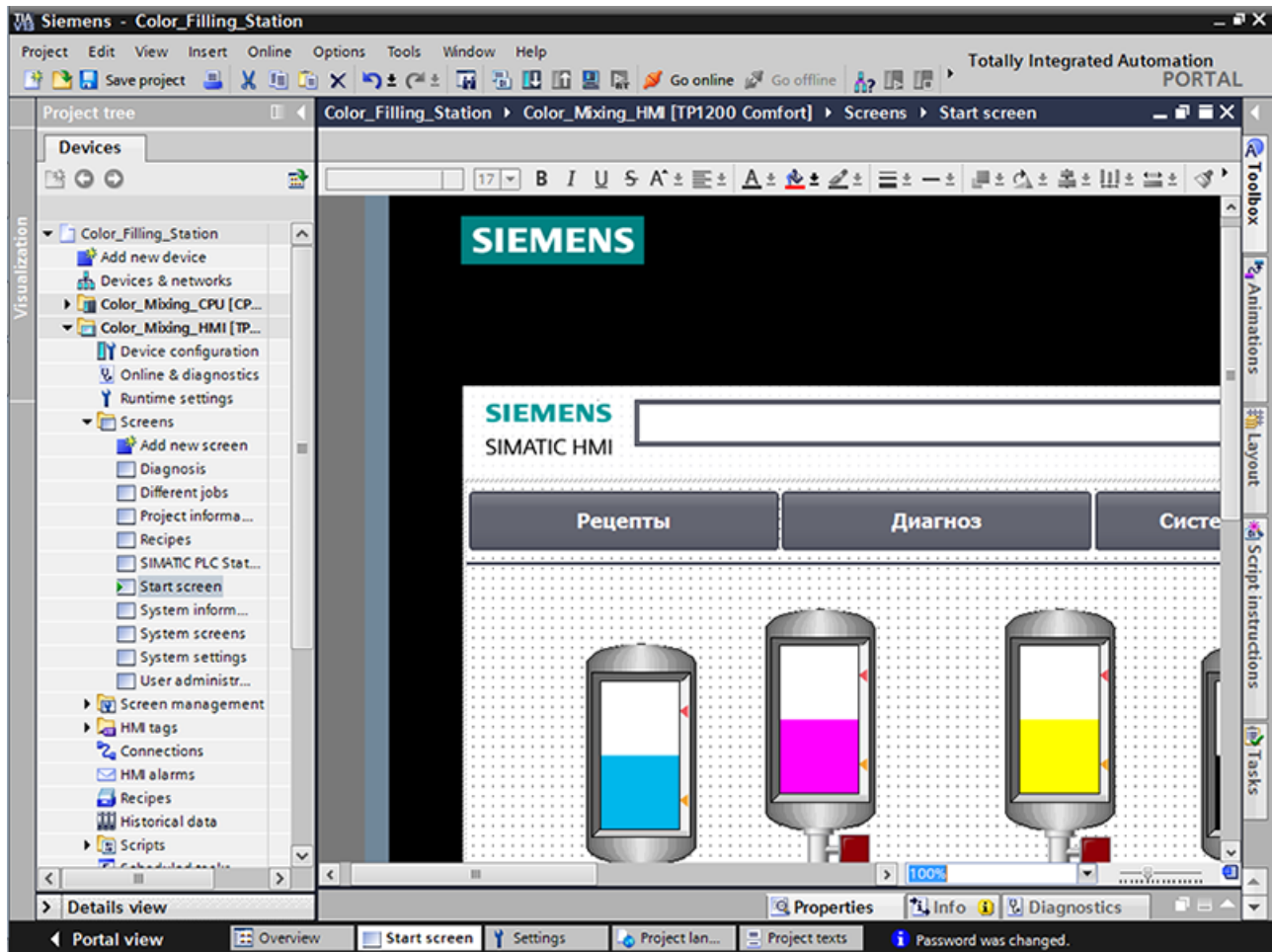


文本在导出并译为俄语后，再次导入。



### 3.3 组态可视化

对于选择的语言，在运行系统中将显示俄语文本。



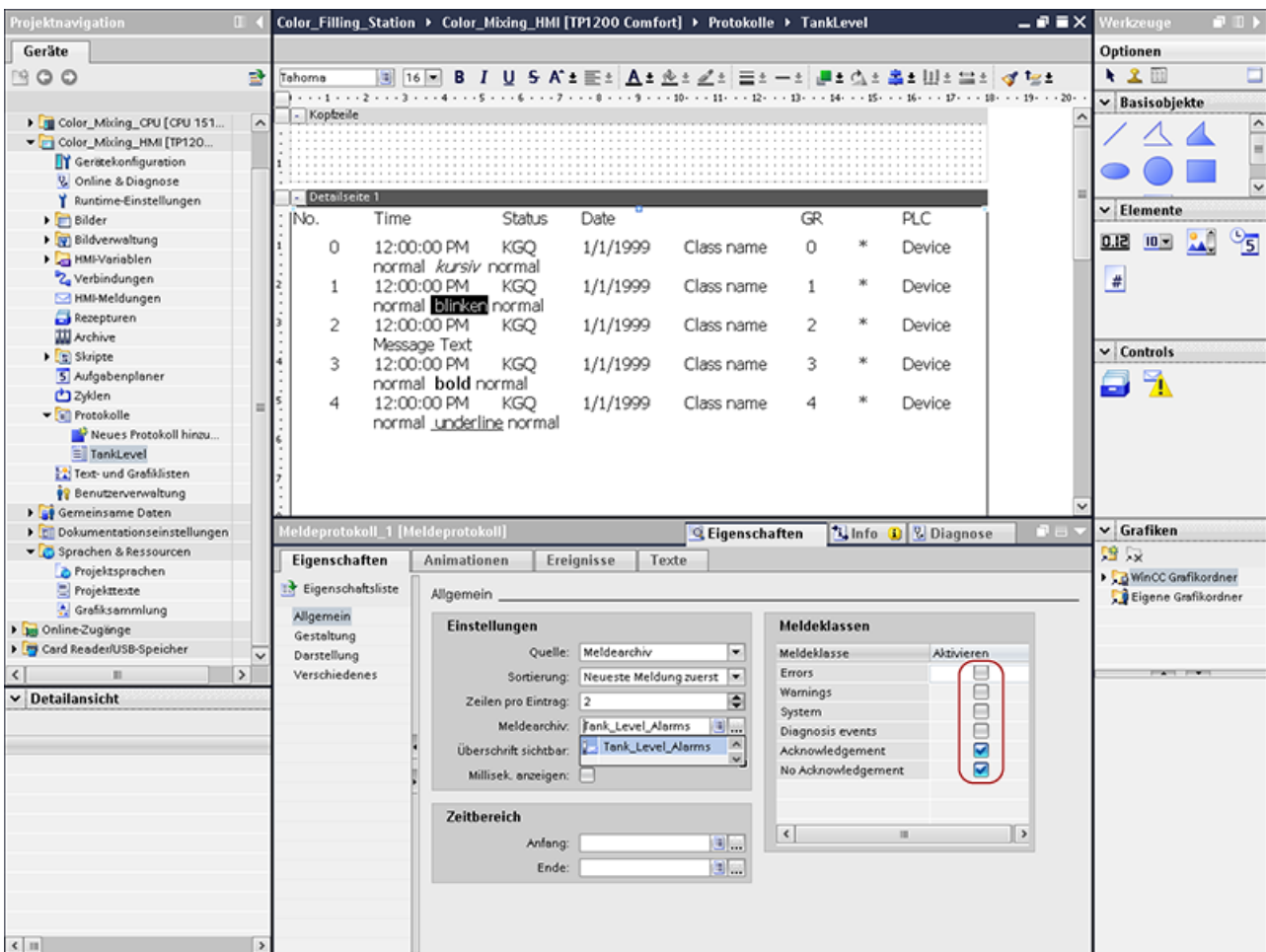


### 3.3.2.10 报表

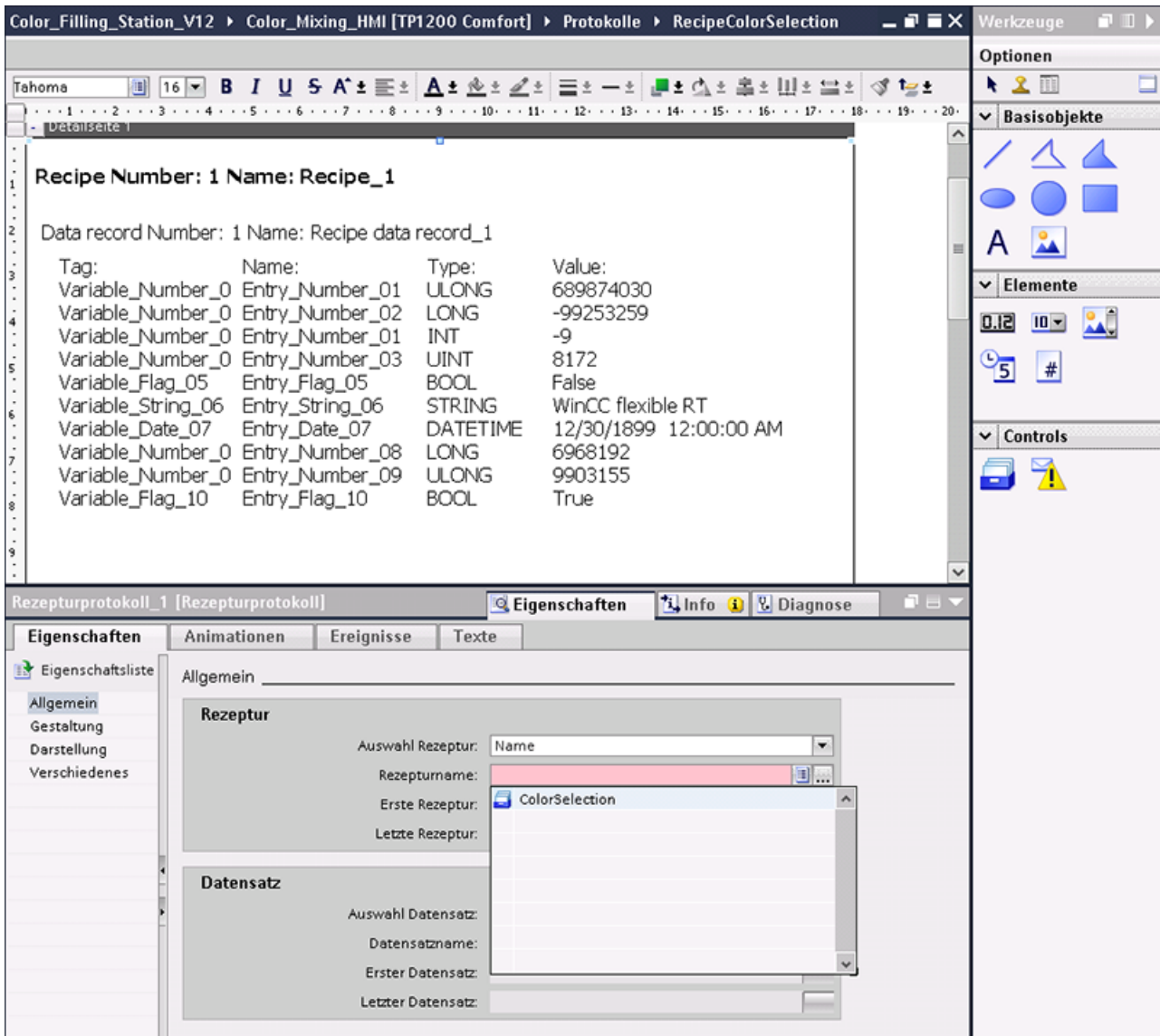
#### 报表

报表用于在生产过程中记录事件以进行产品测试和质量控制。为此，报警和配方数据以固定间隔采用轮班报表形式输出。

在 WinCC 中已针对具有填充量报警的“Tank\_Level”日志创建了报表。



同时也在此项目中创建了配方报表。



连接到 HMI 设备的打印机每天输出报表。

可通过调度器来创建循环输出。

### 3.3.3 插入库中的 HMI 设备

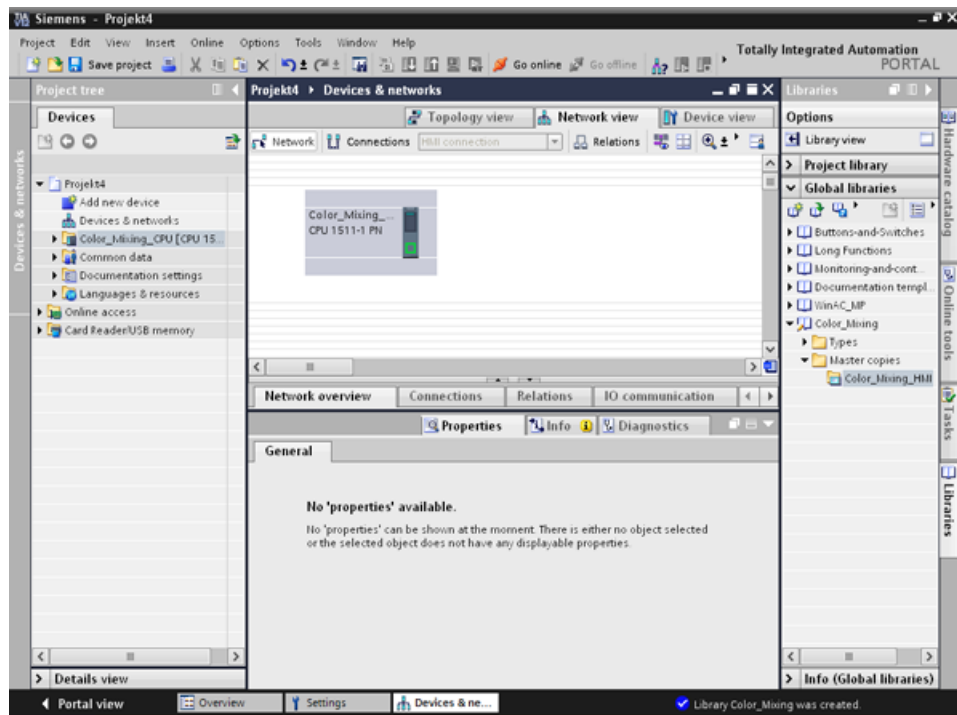
#### 3.3.3.1 存储库中的对象

##### 简介

全局库包含预组态的 HMI 设备。

##### 步骤

1. 打开全局库。
1. 将 HMI 设备“Color\_Mixing\_HMI”拖放到“设备与网络”编辑器。



2. 鼠标指针更改为带有所连对象符号的十字准线。

##### 结果

预组态 HMI 设备已创建并连接到 CPU。

### 3.3.4 组态 HMI 连接

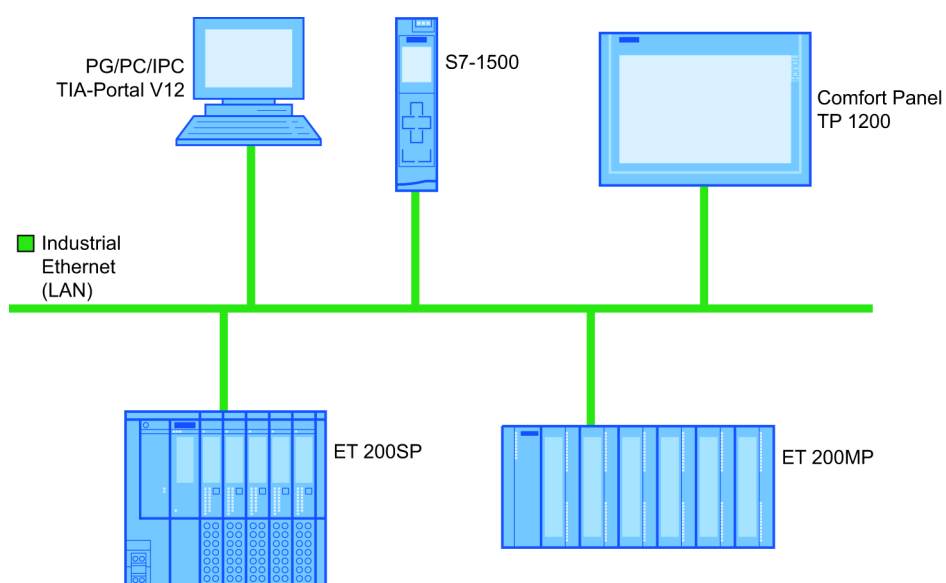
#### 3.3.4.1 设备间的通信

##### 通信

两个设备之间的数据交换称为通信。

设备可以直接互连也可以通过网络互连。

我们将通信中的互连设备认为是通信伙伴。



通信伙伴间传送的数据可用于不同用途：

- 显示过程
- 操作过程
- 输出报警
- 归档过程值和报警
- 记录过程值和报警
- 管理过程参数和机器参数

### 适用于所有通信的基本信息

所有类型通信的基础都是网络组态。

在网络组态中，用户指定两个已组态设备之间的连接。

通过网络组态，还可以确保通信的必要前提条件，即：

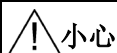
- 网络中的每个设备均分配有唯一地址。
- 设备执行传输特性一致的通信。

#### 3.3.4.2 组态 HMI 连接

##### 简介

在“设备与网络”编辑器中通过 PROFINET 组态精智面板 TP1200 和 CPU 之间的 HMI 连接。

在网络视图中以图形形式显示项目中可用的通信伙伴。



##### 通过以太网通信

在基于以太网的通信中，最终用户将负责数据网络的安全。  
定向攻击可能会导致设备过载并干扰其正常工作。

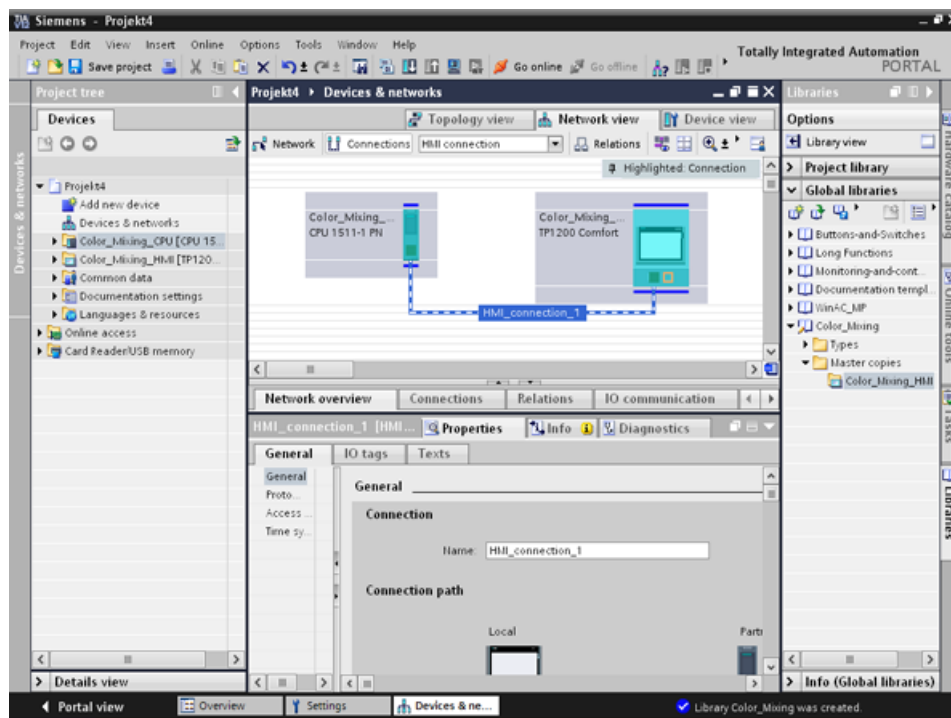
##### 要求

在“设备和网络”编辑器中创建下列通信伙伴：

- HMI 设备：SIMATIC 精智面板
- CPU：SIMATIC S7-1500

### 步骤

1. 单击“连接”(Connections) 按钮并选择“HMI 连接”作为连接类型。  
将以高亮颜色显示可用连接的设备。
2. 单击 CPU 的 PROFINET 接口并使用拖放操作建立到 HMI 设备 PROFINET 接口的连接。



3. 在“网络视图”中单击通信伙伴，并在巡视窗口中根据项目要求更改 PROFINET 参数。

### 说明

创建的 HMI 连接也会显示在编辑器表格区域的“连接”(Connections) 选项卡上。  
检查表中的连接参数。

只能在表格中更改连接的本地名称。

### 结果

在 HMI 设备与 CPU 之间建立了连接。

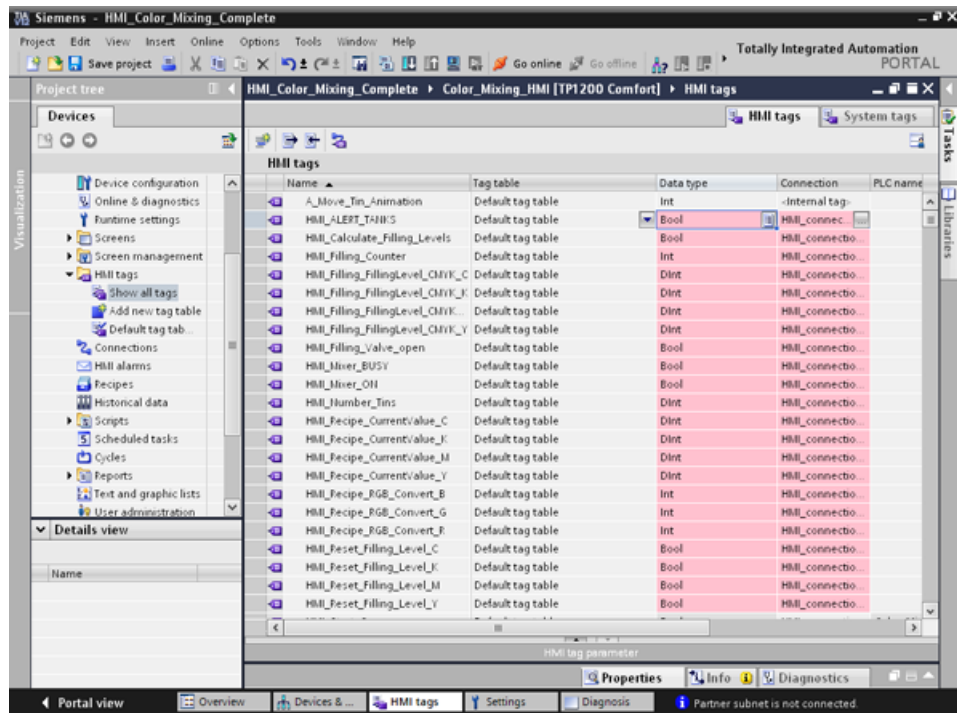
### 3.3.4.3 连接 HMI 变量

#### 简介

在 CPU 和 HMI 设备间建立连接后，便连接了这两个设备的变量。

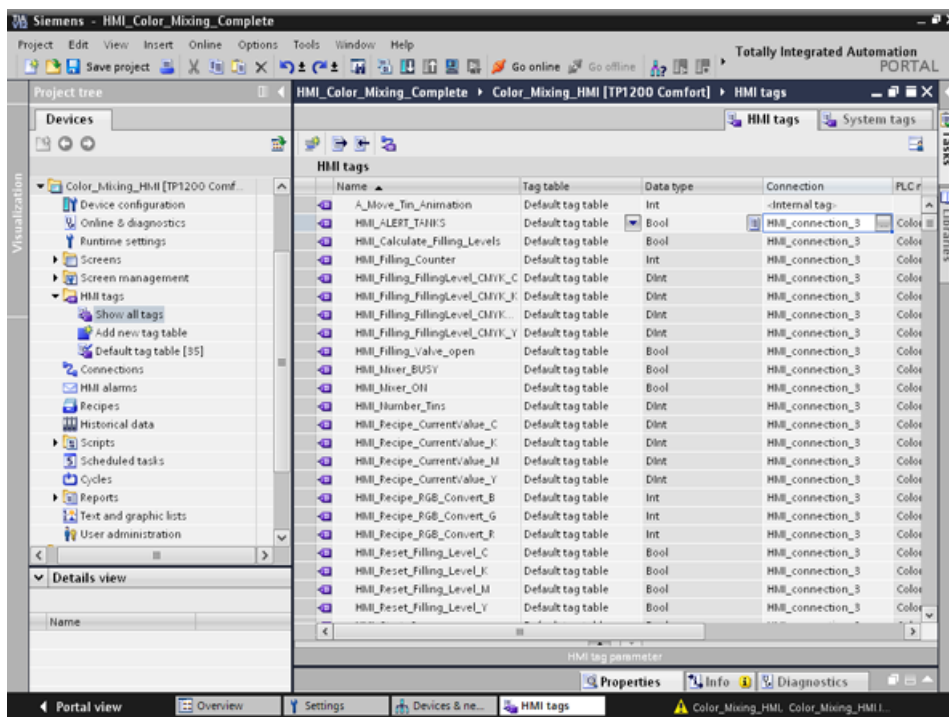
#### 步骤

1. 打开 HMI 变量编辑器。



2. 选择刚在“连接”(Connections) 列中组态的 HMI 连接。

3. 对于以红色突出显示的所有条目，重复此过程。



### 结果

已为 CPU 和 HMI 设备中已组态的变量创建了 HMI 连接。

恢复了此 HMI 连接。



### 3.3.5 组态系统诊断

#### 3.3.5.1 系统诊断的基础知识

##### 简介

系统诊断用于在设备的任何部分中检测问题和错误。WinCC 提供了两个用于快速定位错误的显示和操作元素。

##### 系统诊断视图

报警视图显示 CPU 的状态，而系统诊断视图提供系统内所有可用设备的概览。可直接浏览至错误的原因以及相关设备。您可访问在“设备与网络”编辑器中组态的所有支持诊断功能的设备。

##### 系统诊断窗口

系统诊断窗口是只能在全局画面中使用的操作和显示元素。

系统诊断窗口的功能与系统诊断视图的完全相同。

由于在全局画面中组态了系统诊断窗口，因此，您还可以执行一些操作，例如，指定该对象能否在运行系统中关闭。

### 3.3 组态可视化

#### 3.3.5.2 系统诊断视图

##### 简介

系统诊断显示和系统诊断窗口中提供了四种不同的视图。

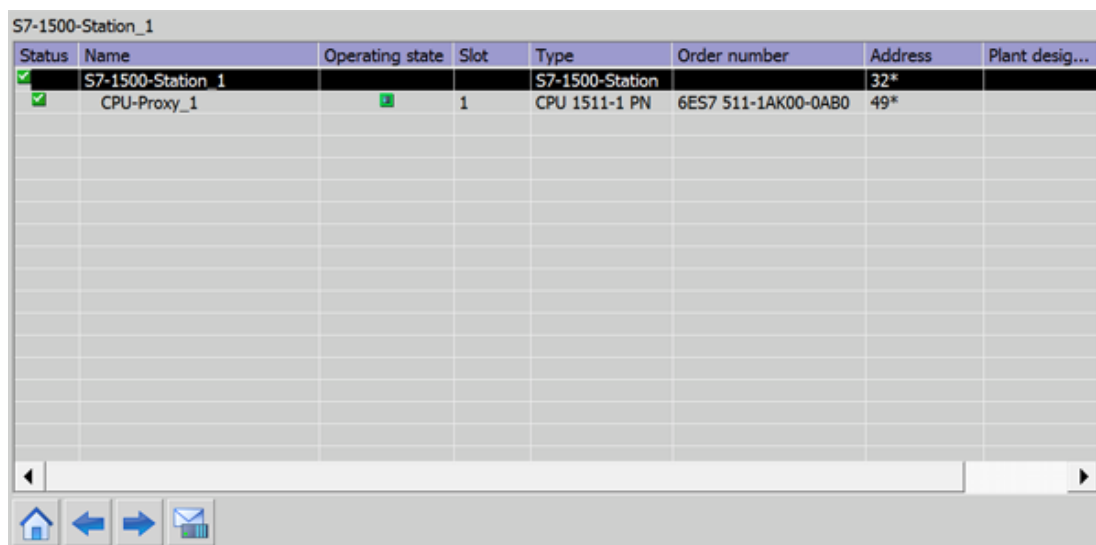
- 设备视图
- 诊断缓冲区视图
- 详细视图
- 矩阵视图（仅针对主系统、PROFIBUS、PROFINET）

##### 设备视图

设备视图以表格形式显示某一层的所有可用设备。

双击某个设备可打开子设备或详细视图。

第一列中的符号提供有关设备当前状态的信息。



Status	Name	Operating state	Slot	Type	Order number	Address	Plant desig...
✓	S7-1500-Station_1			S7-1500-Station		32*	
✓	CPU-Proxy_1	■	1	CPU 1511-1 PN	6ES7 511-1AK00-0AB0	49*	

## 诊断缓冲区视图

诊断缓冲区中的当前数据显示在诊断缓冲区视图中。

N.	Date	Time	Event
1	1/11/2012	10:49:59 PM	Communication initiated request: WARM RESTART - CPU changes from STARTUP to RUN ...
2	1/11/2012	10:49:59 PM	Communication initiated request: WARM RESTART - CPU changes from STOP to STARTUP...
3	1/11/2012	10:49:56 PM	Communication initiated request: STOP - CPU changes from RUN to STOP mode
4	1/11/2012	7:32:36 AM	Communication initiated request: WARM RESTART - CPU changes from STARTUP to RUN ...
5	1/11/2012	7:32:36 AM	Communication initiated request: WARM RESTART - CPU changes from STOP to STARTUP...
6	1/11/2012	7:32:34 AM	Communication initiated request: STOP - CPU changes from RUN to STOP mode
7	1/11/2012	7:30:53 AM	Communication initiated request: WARM RESTART - CPU changes from STARTUP to RUN ...
8	1/11/2012	7:30:53 AM	Communication initiated request: WARM RESTART - CPU changes from STOP to STARTUP...
9	1/11/2012	7:21:03 AM	Communication initiated request: STOP - CPU changes from RUN to STOP mode
...	1/11/2012	4:33:55 AM	Communication initiated request: WARM RESTART - CPU changes from STARTUP to RUN ...
...	1/11/2012	4:33:55 AM	Communication initiated request: WARM RESTART - CPU changes from STOP to STARTUP...
...	1/11/2012	4:22:31 AM	Follow-on operating mode change - CPU changes from STOP (initialization) to STOP mode
...	1/11/2012	4:22:31 AM	Power on - CPU changes from NO POWER to STOP (initialization) mode
...	1/11/2012	4:21:59 AM	Power off - CPU changes from STOP to NO POWER mode
...	1/11/2012	4:19:51 AM	System initiated request: STOP - CPU changes from STOP (firmware update) to STOP mode
...	1/11/2012	4:19:51 AM	Device firmware update finished - CPU changes to STOP mode, new startup inhibit set: .
...	1/11/2012	4:19:03 AM	System initiated request: Firmware update - CPU changes from STOP (initialization) to ST...

## 详细视图

详细视图显示有关所选设备和任何未决错误的详细信息。

在详细视图中检查数据是否正确。在详细视图中不能对错误文本进行排序。

Property	Value
Status	[Error Icon]
Name	DI32
Betriebszustand	
Baugruppenträger	0
Steckplatz	3
Typ	DI32
Bestellnummer	6ES7 521-1BL00-0AB0
Adresse	259*
Anlagenbezeichnung	
Positionskennung	
Hersteller ID	SIEMENS AG
Hardware Version	97
Profil ID	
spezifische Profildaten	0003
I&M Datenversion	1.1
Fehlertext	Drahtbruch
Hilfe: Kontrollieren Sie den Zustand der Verbindungsleitungen.	






### 矩阵视图

矩阵视图仅可用于主系统。矩阵视图显示主系统子设备的状态。

- 在 PROFIBUS 中，可将由 Profibus 分配的编号用作标识（DP 站号）。
- 在 PROFINET 中，IO 设备从 1 开始连续编号。



### 浏览按钮

按钮	功能
	打开子设备或详细视图（如果没有子设备）。
	打开父设备或设备视图（如果没有父设备）。
	打开设备视图。
	打开诊断缓冲区视图。仅在设备视图中可见。
	刷新视图。 已组态的软键，例如 F2

### 3.3.5.3 组态系统诊断视图

#### 简介

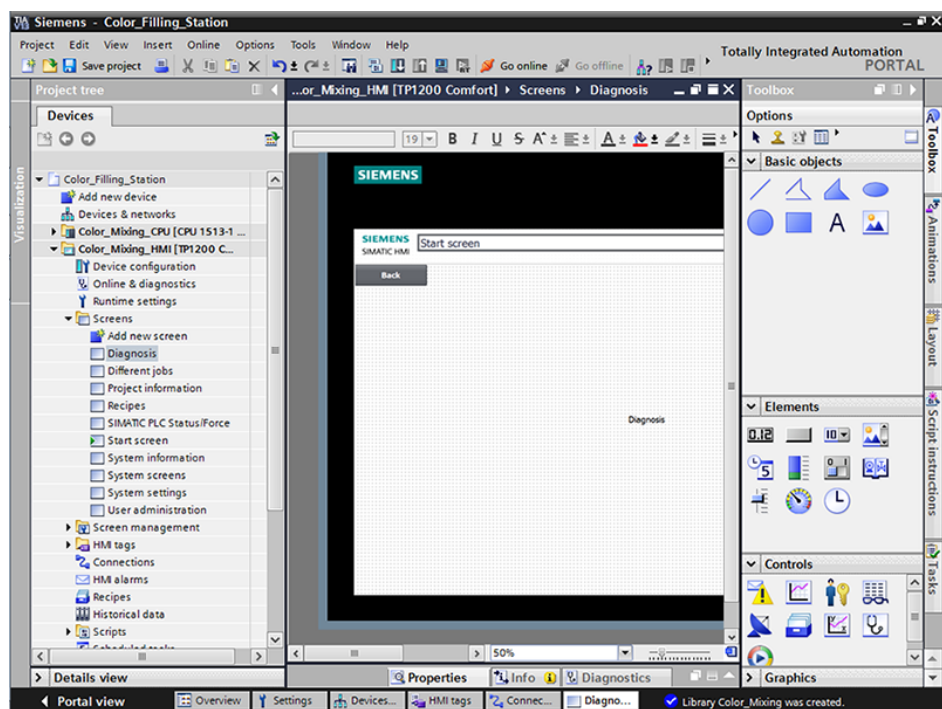
您将一个系统诊断视图添加到自己的项目中，以获取工厂内所有可用设备的概览。

#### 要求

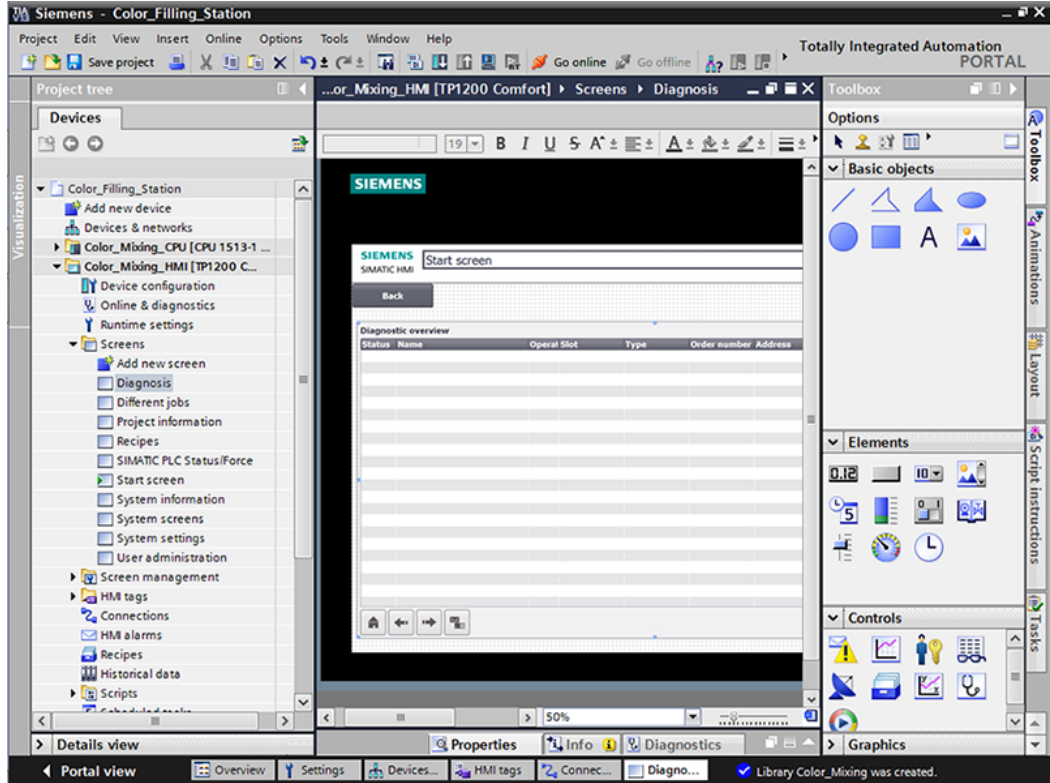
- 已创建 CPU。
- 巡视窗口已打开。

#### 步骤

1. 双击“Diagnostics”HMI 画面。



2. 双击“工具”(Tools) 任务卡中的“系统诊断视图”(System diagnostics view) 对象。  
该对象已添加到该画面中。



3. 在巡视窗口中选择“属性 > 属性 > 列 > 设备/详细视图”(Properties > Properties > Columns > Devices/Detail view)。
4. 在运行系统的设备视图中，启用所需的列，例如状态、名称和插槽。
5. 在运行系统的详细视图中，启用所需的列，例如状态、名称和更高级别的名称。
6. 在诊断缓冲区视图中，启用所需的列，例如状态、名称和机架。
7. 如有必要，调整列标题。
8. 启用“属性 > 属性 > 布局 > 列设置 > 可移动列”(Properties > Properties > Layout > Column settings > Columns moveable) 以在运行系统中移动列。
9. 可以根据需要在“属性 > 属性 > 列标题”(Properties > Properties > Column headers) 下更改列标题。

## 结果

系统诊断视图已添加到“诊断”(Diagnostics) 画面中。

运行系统的系统诊断视图中现在显示了整个设备的错误消息。

### 3.3.6 模拟 HMI 设备

#### 3.3.6.1 仿真基本知识

##### 简介

可以使用仿真器来测试组态 PC 上组态的性能。  
这使您可以在进行生产操作之前快速找到任何逻辑上的组态错误。

可如下所示启动仿真器：

- 在 HMI 设备或画面的快捷菜单中：“启动仿真”(Start simulation)
- 菜单命令“在线 > 仿真 [启动|使用变量仿真器|使用脚本调试器]”(Online > Simulation > [Start|With tag simulator|With script debugger])
- 在 Portal 视图的“可视化 > 仿真设备”(Visualization > Simulate device) 下

##### 要求

组态 PC 上已安装仿真/运行系统组件。

##### 应用领域

可使用仿真器来测试 HMI 系统的以下功能，例如：

- 检查限制级别和报警输出
- 中断的一致性
- 组态的中断仿真
- 组态的警告
- 组态的错误消息
- 检查状态显示

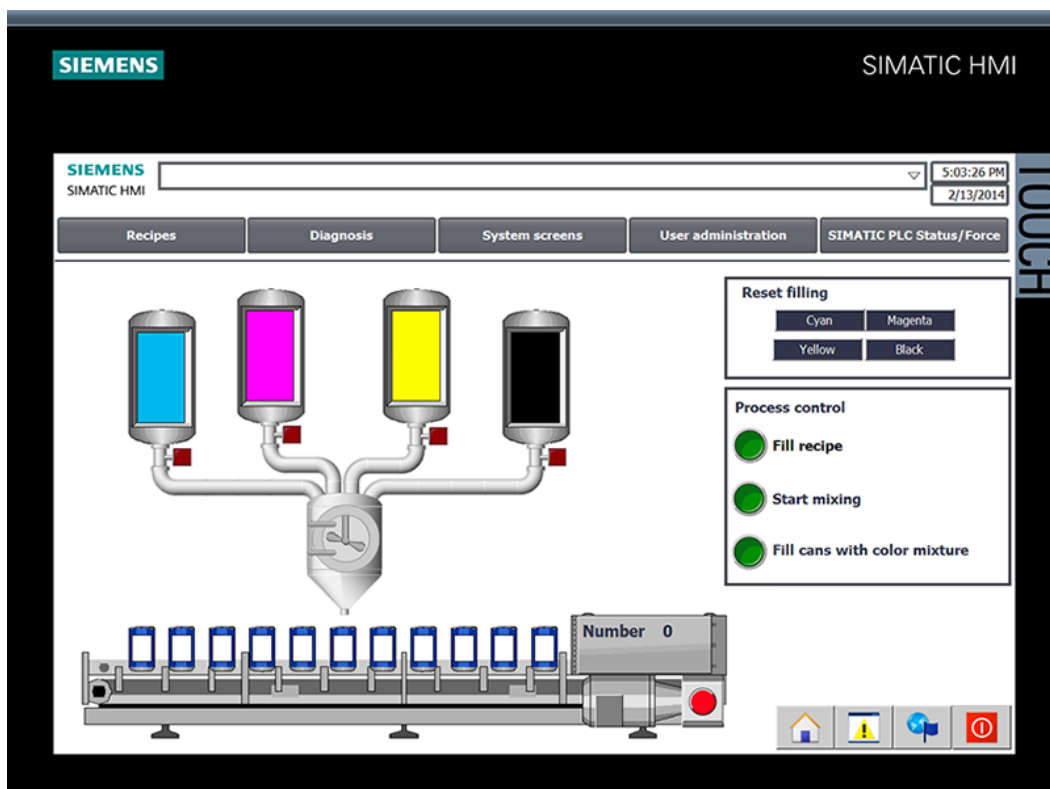
### 3.3.6.2 以仿真状态运行设备

#### 简介

在计算机上仿真 HMI 项目。

#### 步骤

1. 启动 HMI 设备仿真。

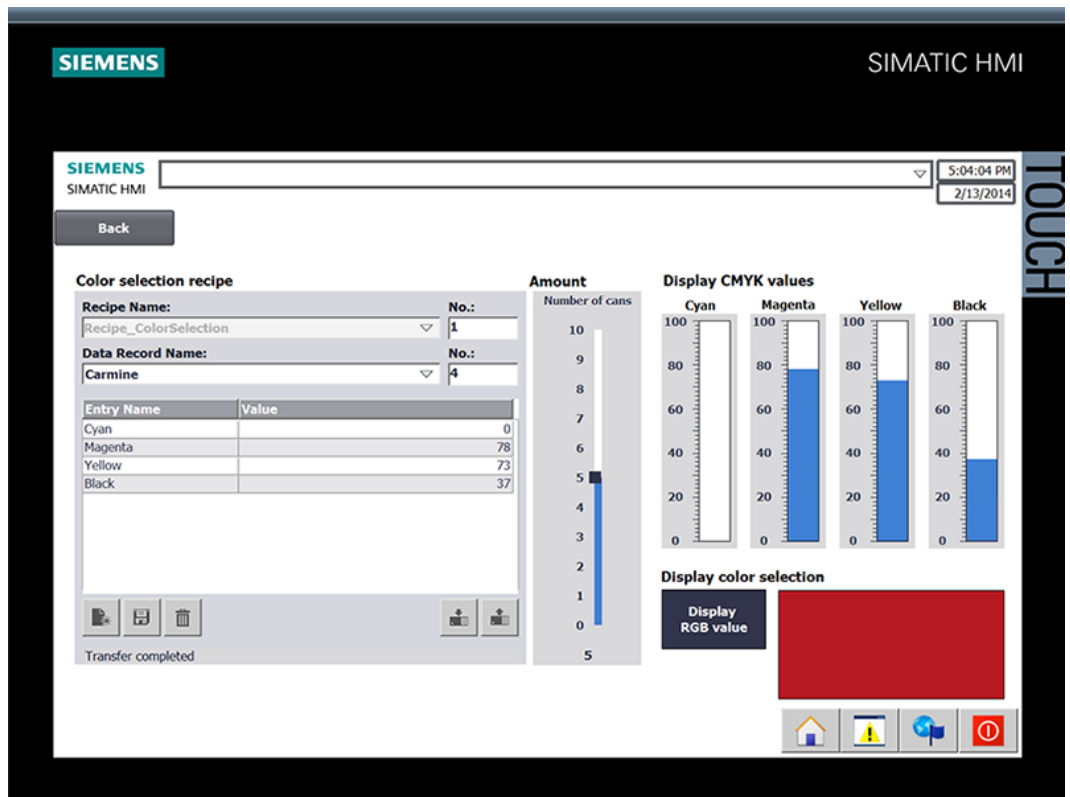


与 CPU 建立连接并显示仿真颜色混合系统。

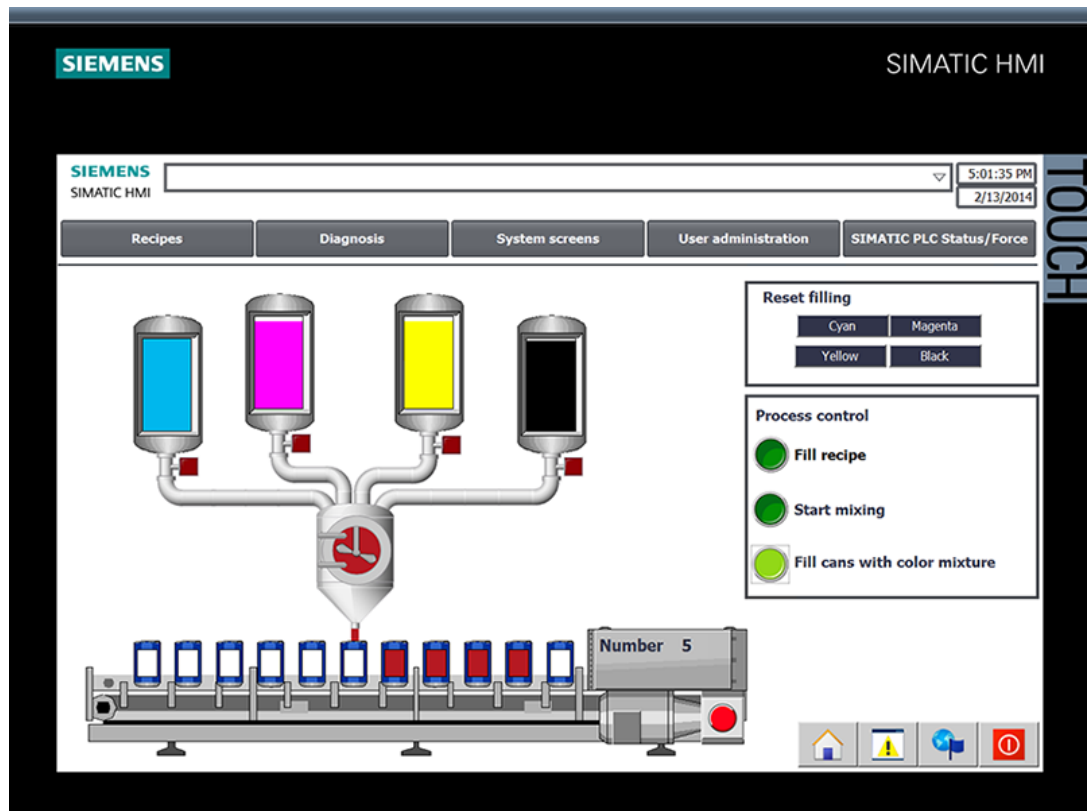
2. 打开“配方”(Recipes) 画面并选择颜色。



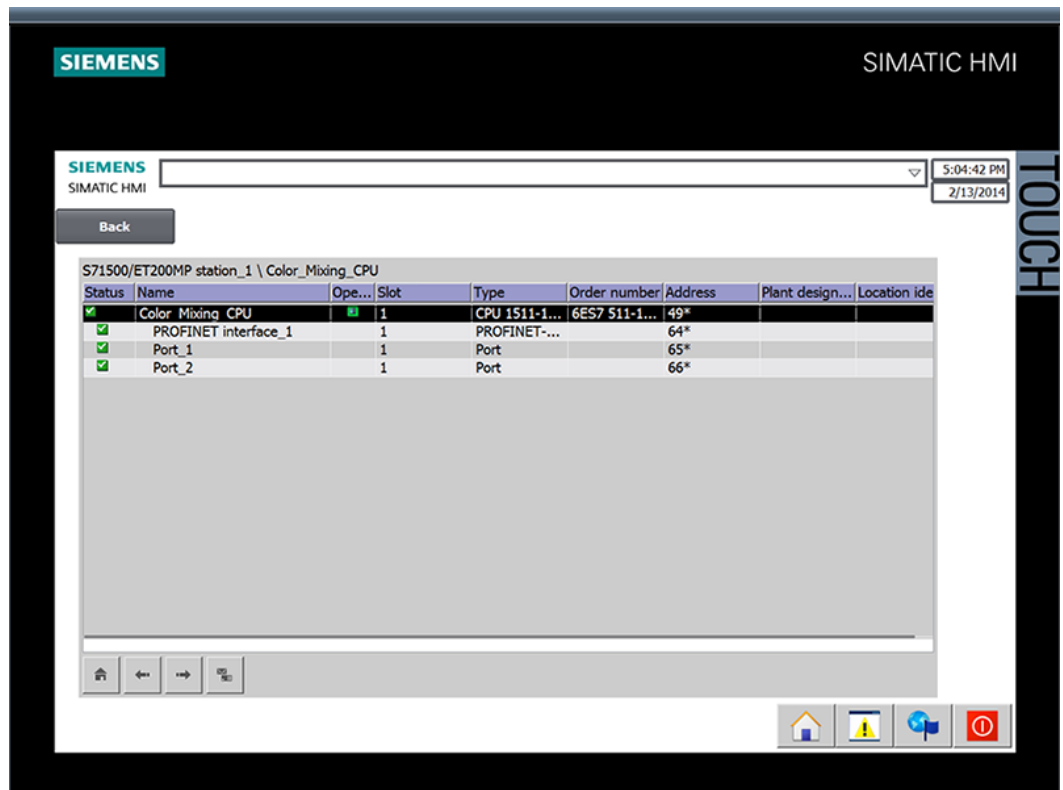
3. 指定容器数并查看所选颜色。



4. 返回至启动画面和启动生产。



5. 可在“诊断”(Diagnostics) 画面中查询当前 CPU 状态。



### 3.4 将项目加载到编程设备

## 3.4 将项目加载到编程设备

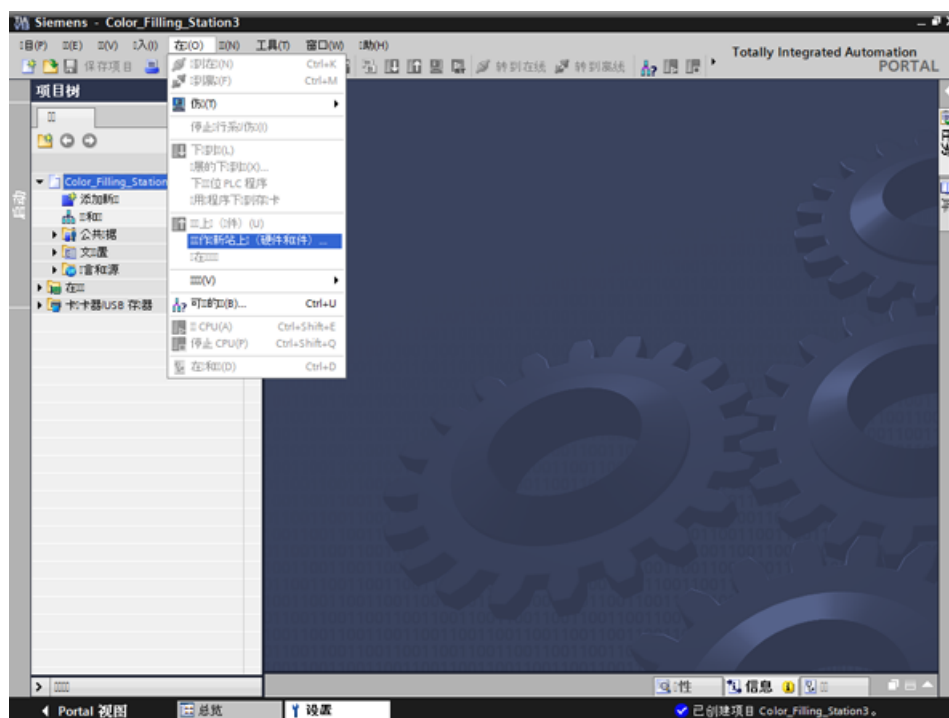
### 3.4.1 将 CPU 加载到项目

#### 简介

可以创建一个新站，其中包含来自硬件配置和用户程序的实用值。

#### 步骤

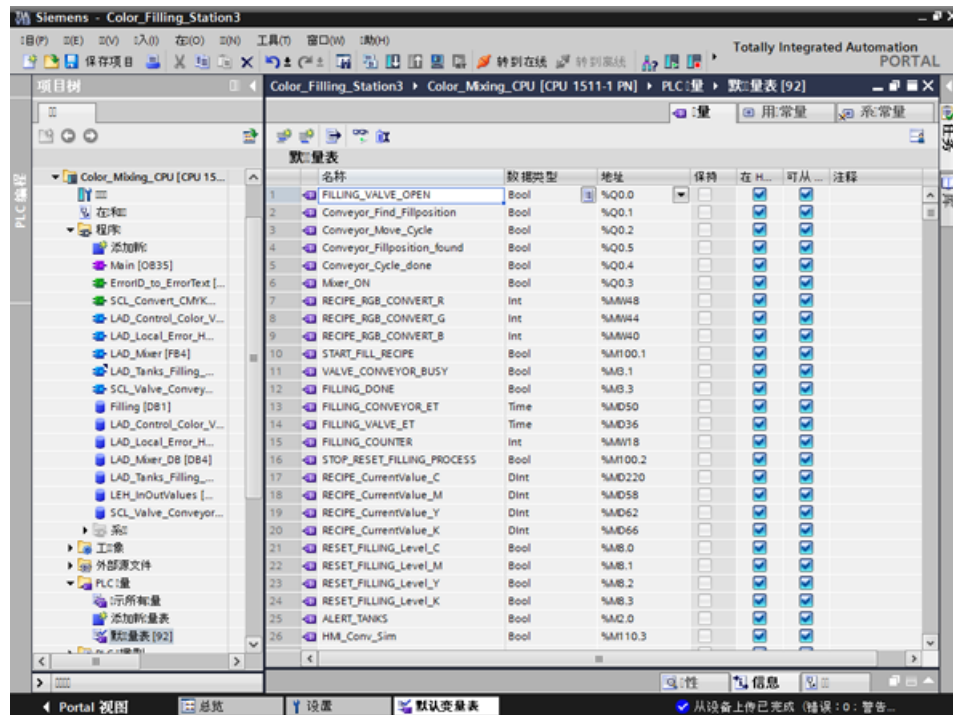
1. 打开用于从 CPU 加载数据的对话框。



2. 选择用于将编程设备连接到 CPU 的接口。自动开始搜索可访问的节点。
3. 在项目中加载 CPU。

## 结果

CPU 的硬件和软件组态加载到项目中。例如，项目中现在包含程序块和变量。



### 3.5 通过“项目间工程组态”进行团队工程组态

## 3.5 通过“项目间工程组态”进行团队工程组态

### 3.5.1 “项目间工程组态”基本知识

#### 简介

本节介绍团队工程组态的优点以及如何生成 HMI 项目工程师所需的 CPU 数据。HMI 项目工程师可以了解如何在项目中使用的这种 CPU 数据。

#### 分布式组态

使用“Inter Project Engineering”，可以在几个不同地点来并行开发用户程序用户界面。HMI 项目工程师不需要任何 CPU 用户程序，也不需要安装 STEP 7。

为了将 HMI 设备连接到 CPU，只需使用 CPU 接口的变量、块、消息和地址信息。程序员可方便地将这种数据导出到一个 IPE 文件中，然后 HMI 开发人员将该文件导入项目。通过传输新的 IPE 文件，可以随时进行更新。

数据在加载到 CPU 和 HMI 设备后保持一致。在 HMI 组态中创建的和 CPU 之间的连接保持最新状态。

---

#### 说明

从 V5.4 SP3 起，通过导入到 TIA Portal 中，可将新的第二代精简系列面板和精智面板导入集成到 STEP 7 项目中。

---

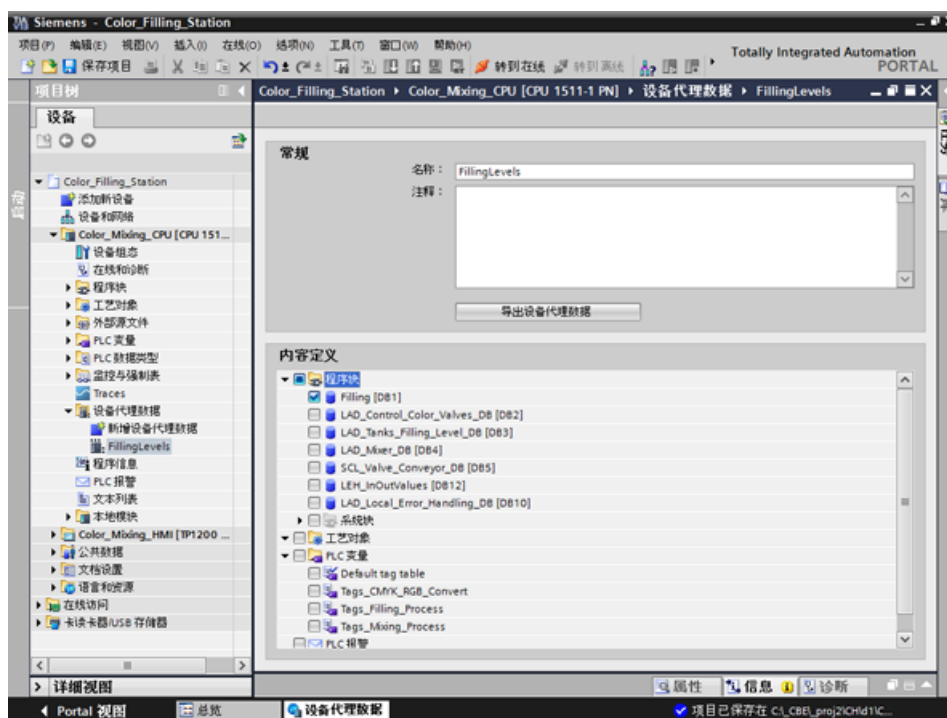
## 3.5.2 创建 IPE 文件

### 简介

要直接在油漆混合装置处显示灌装液位，需要使用一种紧凑型 HMI 设备。您可以委托一个工程公司来实现这种可视化，并以 IPE 文件的形式提供所需的 CPU 数据。

### 步骤

1. 为 CPU 添加新的代理数据。
2. 输入名称并选择所需的 CPU 数据。



3. 导出代理数据。

### 结果

创建了 IPE 文件。例如，可以通过电子邮件将该 IPE 文件发送给工程公司。

### 3.5 通过“项目间工程组态”进行团队工程组态

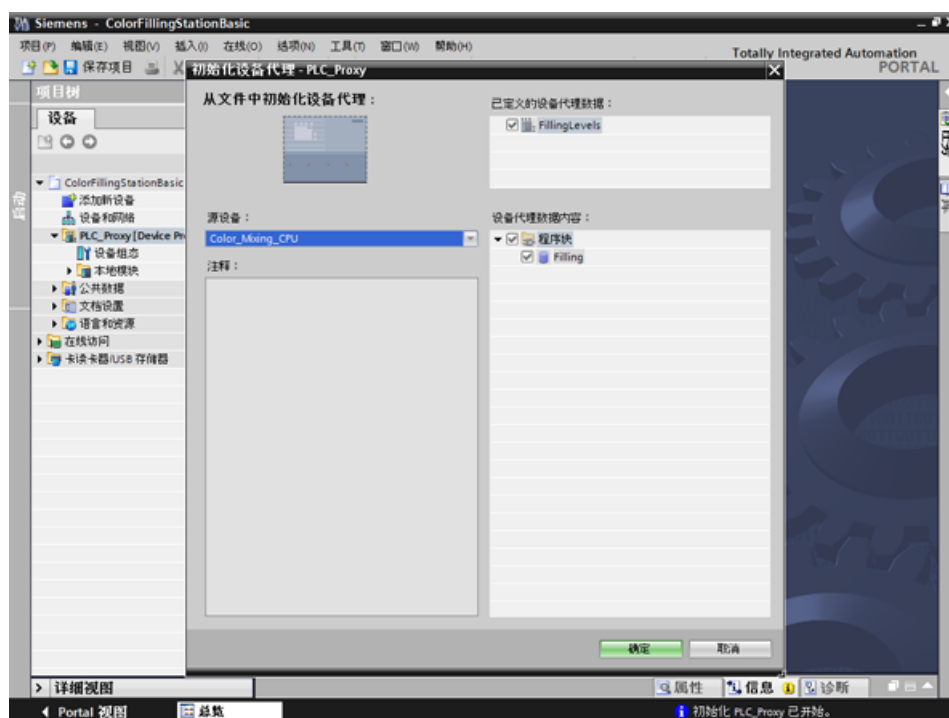
#### 3.5.3 导入 IPE 文件

##### 简介

在工程办公室中，项目工程师创建新项目中的设备代理，并用 IPE 文件中的 CPU 数据将其初始化。项目工程师针对 IPE 文件的每次更新，都重复这种初始化。

##### 步骤

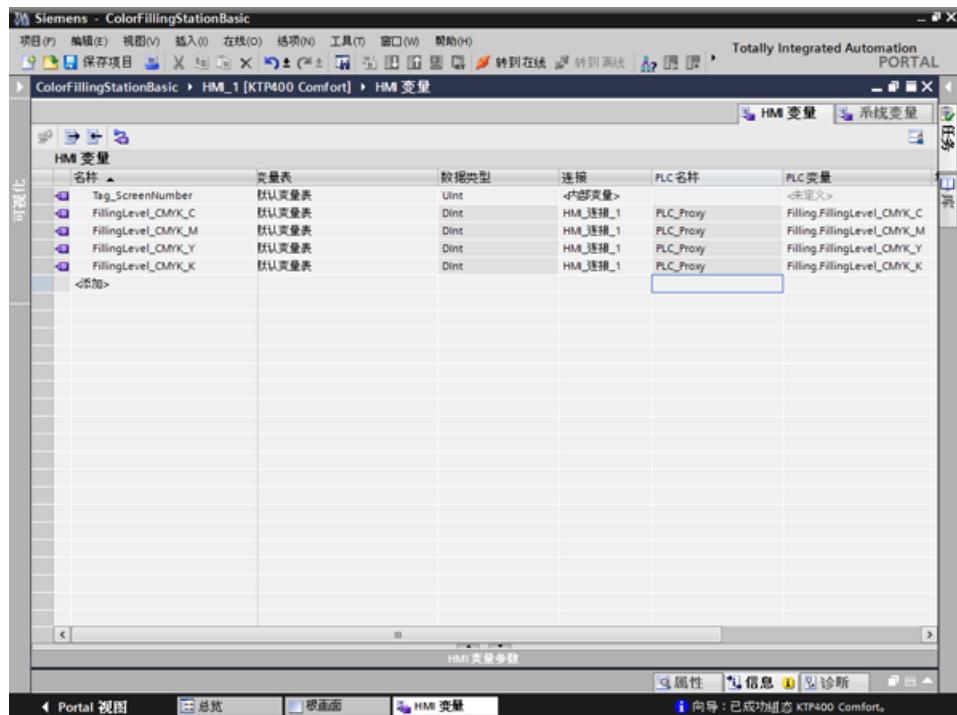
1. 为新项目中的 CPU 创建设备代理，然后将其初始化。



2. 使用设备向导来插入一个精简系列面板。
3. 创建“Cyan”颜色的灌注液位的 HMI 变量，然后选择该 PLC 变量。



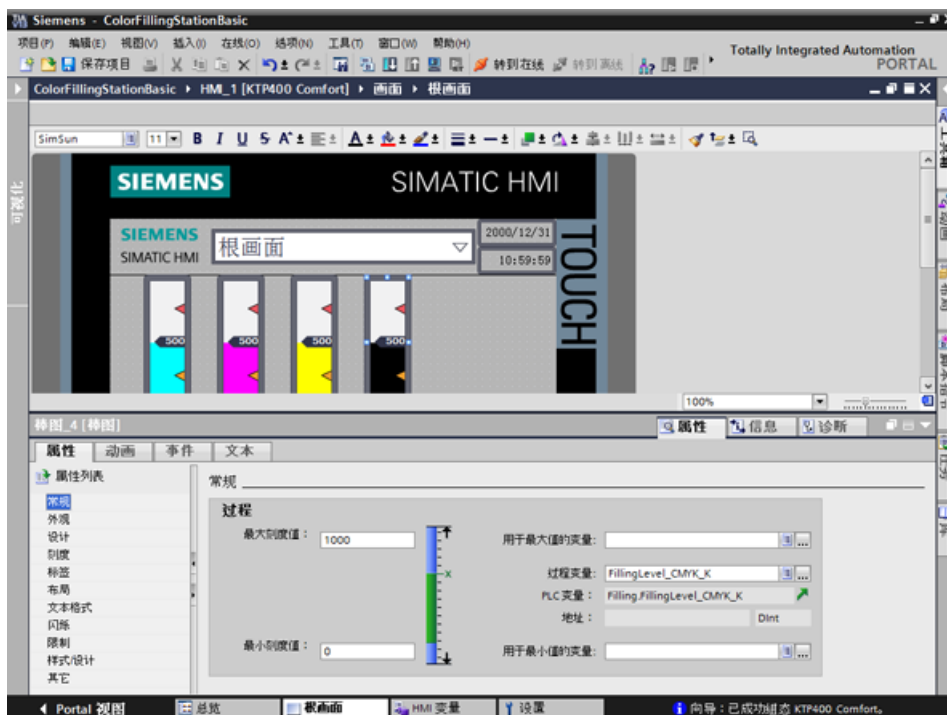
4. 以相同方式创建其它 HMI 变量。



5. 组态一个用于显示“Cyan”颜色的灌注液位的棒图。

3.5 通过“项目间工程组态”进行团队工程组态

6. 以相同方式创建其它灌注液位的棒图。



7. 编译项目。

结果

此时可将该项目从调试工程师加载到 HMI 设备。由于使用 IPE 文件中的 CPU 数据，与 CPU 的通信保持最新状态。

## 安全性

### 4.1 CPU 保护功能的概述

#### 简介

本章描述了下列用于防止对 S7-1500 自动化系统进行未经授权的访问的功能：

- 访问保护
- 专有技术保护
- 防拷贝保护
- 通过锁定 CPU 进行保护

#### 保护 CPU 的其它措施

下列措施进一步防止了从外部源和网络对 S7-1500 CPU 的功能和数据进行未经授权的访问。

- 禁用 Web 服务器
- 禁用通过 NTP 服务器的时间同步
- 禁用 PUT/GET 通信

使用 Web 服务器时，通过在用户管理中设置特定用户的密码保护访问权，可防止对 S7-1500 自动化系统进行未经授权的访问。

## 4.2 使用显示屏组态其它访问保护

### 简介

在 S7-1500 的显示屏上，可防止对受密码保护的 CPU 进行访问（本地锁定）。

仅当操作模式开关处于 RUN 位置时，此访问锁定才生效。

是否应用访问锁定与密码保护无关。即，如果某人通过连接的编程设备访问了 CPU 并且已输入正确的密码，同样会阻止该用户访问 CPU。

可以分别为显示屏上的每个访问级别设置访问阻止，这样就可以在本地允许读取访问，但是不允许写入访问。

### 操作步骤

如果在 STEP 7 中组态了使用密码的访问级别，则可使用显示屏来阻止访问。

按照以下操作在显示屏上设置 S7-1500 CPU 的本地访问保护：

1. 在显示屏上，选择“设置 > 保护”(Settings > Protection) 菜单。
2. 单击“确定”(OK) 进行确认选择，并组态每种访问级别中是否允许在 RUN 模式选择开关下访问：

允许 (Allow)：如果输入 STEP 7 中指定的密码，则可以访问 CPU。

在 RUN 模式下禁用：当操作模式开关位于 RUN

位置时，即便其他用户知道该访问级别的权限密码，也无法登录该 CPU。在 STOP 模式下，可通过输入密码进行访问。

### 显示屏访问保护

可以在 STEP 7 中的 CPU

属性中为显示屏组态密码，这样本地访问保护就可由本地密码进行保护。

## 4.3 专有技术保护

可以使用专有技术保护来保护程序中一个或多个 OB、FB、FC 类型的块以及全局数据块，防止受未经授权的访问。可以输入密码限制对块的访问。密码保护可防止在未经授权的情况下读取或更改块。

如果没有密码，则只能读取有关块的以下数据：

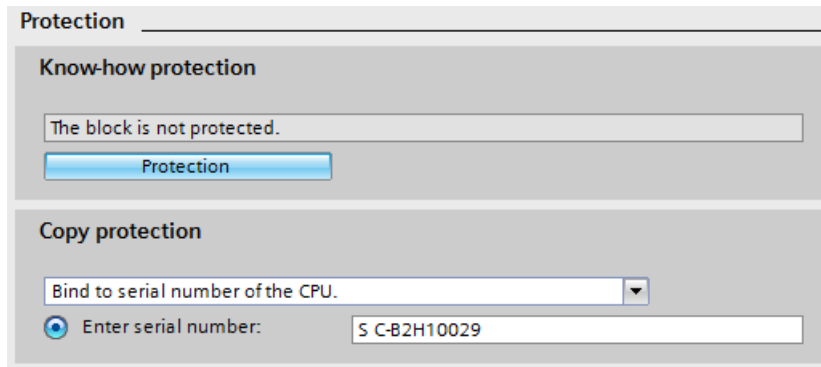
- 块标题、注释和块属性
- 块参数（INPUT、OUTPUT、IN、OUT、RETURN）
- 程序调用结构
- 不带使用点信息的全局变量

对于受到专有技术保护的块，可执行以下进一步操作：

- 复制和删除
- 在程序中调用
- 在线/离线比较
- 加载

## 设置块的专有技术保护

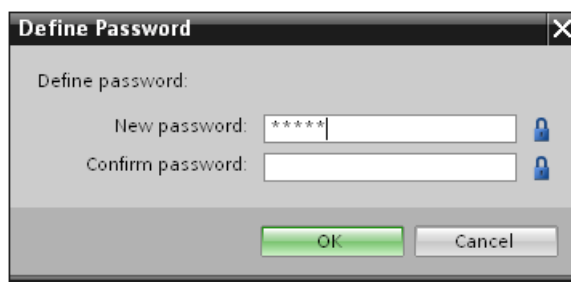
1. 打开相应块的属性。
2. 请在“常规”(General) 下选择“保护”(Protection) 选项。



3. 单击“保护”(Protection) 按钮，显示“专有技术保护”(Know-how protection) 对话框。



4. 单击“定义”(Define) 按钮，打开“定义密码”(Define password) 对话框。



5. 在“新密码”(New password) 域中输入新密码。在“确认密码”(Confirm password) 域中输入相同的密码。
6. 单击“确定”(OK)，确认输入。
7. 单击“确定”(OK)，关闭“专有技术保护”(Know-how protection) 对话框。

结果：所选块将受到专有技术保护。

在项目树中，受专有技术保护的块将标记为一个锁形标识。

输入的密码将应用于所选的所有块。

### 打开受到专有技术保护的块

1. 双击块，将打开“访问保护”(Access protection) 对话框。
2. 输入受专有技术保护块的密码。
3. 单击“确定”(OK)，确认输入。

结果：将打开受专有技术保护的块。

打开该块之后，只要该块或 TIA Portal 打开，就可以编辑该块的程序代码和块接口。

下次打开块时，必须重新输入密码。如果使用“取消”(Cancel)

按钮关闭“访问保护”(Access protection)

对话框，则块虽然可以打开，但不显示块代码也不能进行编辑。

如果复制块或将其添加到库中，不会删除块的专业技术保护。

这些副本块同样也受专有技术保护。

### 删除块的专有技术保护

1. 选择要删除专有技术保护的块。不能在程序编辑器中打开受保护的块。
2. 在“编辑”(Edit) 菜单中，选择“专有技术保护”(Know-how protection) 命令以打开“专有技术保护”(Know-how protection) 对话框。
3. 禁用复选框“隐藏代码(专有技术保护)”(Hide code (know-how protection))。



4. 输入密码。



5. 单击“确定”(OK)，确认输入。

结果：将删除所选块的专有技术保护。

## 4.4 防拷贝保护

防拷贝保护则需将程序或块与一个特定的 SIMATIC 存储卡或 CPU 进行绑定。通过链接 SIMATIC 存储卡或 CPU 的序列号，该程序或块只能与 SIMATIC 存储卡或 CPU 一起使用。

使用这一功能，可通过电子方式（例如，通过电子邮件）或通过发送存储器模块的方式来发送程序或块。

为块设置此类防拷贝保护时，还为其分配了专有技术保护。

未设置专有技术保护时，任何人都可以复位该防拷贝保护。

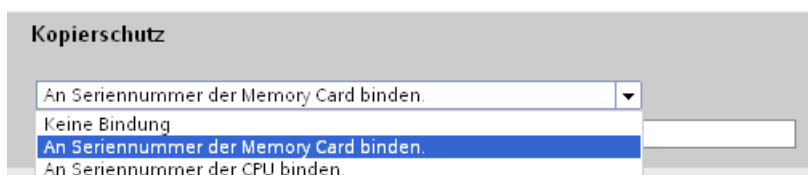
然而当该块已设置为受专有技术保护时，则首先应该将防拷贝保护设置为只读。

### 设置防拷贝保护

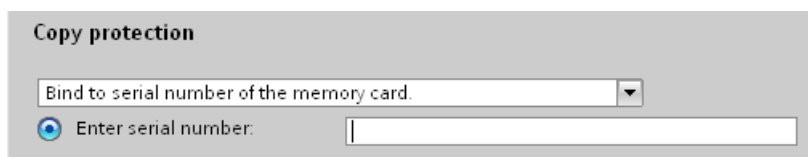
1. 打开相应块的属性。
2. 请在“常规”(General) 下选择“保护”(Protection) 选项。



3. 在“防拷贝保护”（Copy protection）区域中，从下拉列表中选择“绑定 CPU 的序列号”（Bind to serial number of the CPU）条目或“绑定存储卡的序列号”（Bind to serial number of the memory card）条目。



4. 输入 CPU 或 SIMATIC 存储卡的序列号。



5. 现在，可以在“专有技术保护”(Know-how protection) 区域中设置块的专有技术保护。

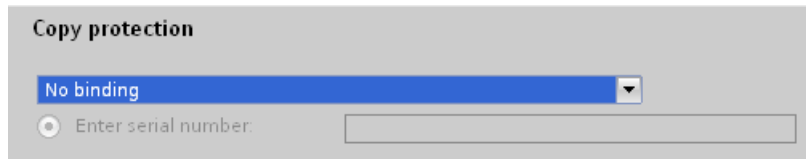
### 说明

如果将受防拷贝保护的块下载到与特定序列号不匹配的设备中，则将拒绝执行整个下载操作。这意味着，也不会下载不带防拷贝保护的块。



## 取消防拷贝保护

1. 取消现有专有技术保护。
2. 打开相应块的属性。
3. 请在“常规”(General) 下选择“保护”(Protection) 选项。
4. 在“防拷贝保护” (Copy protection) 区域中，从下拉列表中选择“不绑定” (No binding) 条目。

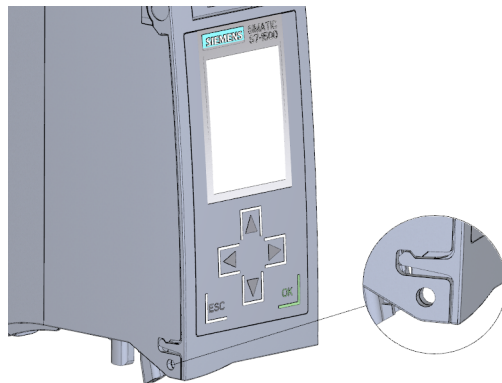


## 4.5 通过锁定 CPU 进行保护

可以使用坚固的前盖板，防止 CPU 免到受未经授权的访问。

通过 CU 外盖上的锁具，可进行如下选择：

- 加盖印章
- 使用锁具锁定前盖板（锁孔直径：3 mm）



## 4.6 组态 CPU 的访问保护

### 简介

CPU 中共有四种访问级别，用于限制对特定功能的访问。

设置 CPU 的访问等级和密码后，则需输入密码才能访问功能和存储区。将在 CPU 的对象属性中指定各种访问级别以及相关的密码条目。

### CPU 的访问级别

访问级别	访问限制
完全访问权（无保护）	所有用户都可以对硬件配置和块进行读取和更改操作。
读访问权	在这一级访问中，可以不输入密码对硬件配置和块进行只读访问。即可将硬件配置和块加载到编程设备中。还可以进行 HMI 访问和诊断数据访问。  但不输入密码，无法将任何块或硬件配置加载到 CPU 中。此外，如果没有密码，也 <b>无法</b> 进行以下操作： 写入的功能测试、更改操作模式 (RUN/STOP) 以及固件更新（在线）。
HMI 访问权	在这一级访问中，不输入密码只能访问 HMI 和诊断数据。  如果不输入密码，既不能将块和硬件配置加载到 CPU 中，也无法从 CPU 中将块和硬件配置加载到编程设备中。  此外，如果没有密码，也 <b>无法</b> 进行以下操作： 写入的功能测试、更改操作模式 (RUN/STOP) 以及固件更新（在线）。
无访问权（完全保护）	对 CPU 进行完全保护时，无法对硬件配置和块进行读写访问。同样无法进行 HMI 访问。PUT/GET 通信的服务器功能在该访问级别中被禁用（无法更改）。  必须通过密码验证，才能提供 CPU 的完全访问权。

无论是哪一种访问级别，都可以无限制地访问某些功能而无需输入密码。例如，使用“可访问的设备”(Accessible devices) 功能进行识别。

CPU 的默认设置为“无限制”(No restriction) 和“无密码保护”(No password protection)。要保护对 CPU 的访问，必须编辑 CPU 的属性并设置密码。

除非禁用了 PUT/GET 通信，否则 CPU 之间的通信（通过块中的通信功能）不受 CPU 的保护等级限制。

权限密码条目允许访问对应级别中允许的所有功能。

## 说明

### 组态一个访问级别并不能取代专有技术保护

通过限制下载权限，组态访问级别可防止对 CPU 进行未经授权的更改。但不会对 SIMATIC 存储卡上的块设置受读写保护。而使用专有技术保护则可以保护 SIMATIC 存储卡上的代码块。

## 对过程进行访问级别参数设置

请按以下步骤组态 S7-1500 CPU 的访问级别：

1. 在巡视窗口中，打开 S7-1500 CPU 的属性。
2. 在区域导航中打开“保护”(Protection) 条目。

将在巡视窗口中显示一张列有各种访问级别的表格。

Protection level	Access			Access permission	
	HMI	Read	Write	Password	Confirmation
<input checked="" type="radio"/> Full access (no protection)	✓	✓	✓		
<input type="radio"/> Read access	✓	✓			
<input type="radio"/> HMI access	✓				
<input type="radio"/> No access (complete protection)					

3. 激活表格第一列中所需的保护等级。

此列中相应访问级别右侧的绿色复选标记将指示如不输入密码仍可执行的操作。

4. 在“密码”(Password) 列中，为所选访问级别指定一个密码。在“确认”(Confirmation) 列中，再次所选输入密码以免输入错误。

确保密码足够安全，即，不要按照机器可识别的模式来设置密码。

必须在第 1 行中输入密码（“完全访问权”访问级别）。

知道该密码的用户就可以不受限制地访问 CPU，而无需考虑所选保护等级。

5. 如果所选访问级别允许的话，可以根据需要将额外的密码分配到其它访问级别。
6. 将硬件配置下载到 CPU，以使访问级别生效。

#### 4.6 组态 CPU 的访问保护

##### 操作期间受密码保护的 CPU 的行为

CPU 保护在将设置下载到 CPU 之后生效。

在执行在线功能之前，需检查所需的权限，必要时提示用户输入密码。

在任何时刻，只能在一个 PG/PC 执行受密码保护的功能。其它 PG/PC 无法登录。

保护数据的访问授权在在线连接时间内有效，或在通过“在线 > 删除访问权限”(Online > Delete access rights) 手动取消访问授权之前保持有效。

在 RUN 模式下，对受密码保护的 CPU

进行的访问仅限于在本地显示屏中进行，同时也不能进行使用密码的访问。

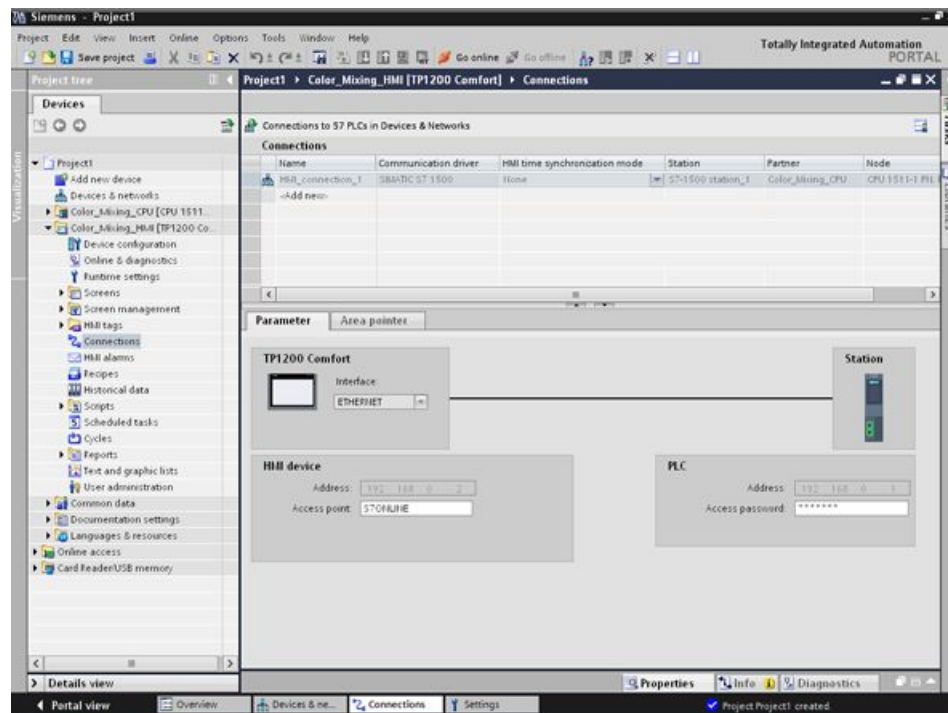
## 4.7 组态 HMI 连接保护

### 简介

如果 CPU 设置了保护等级“完全保护”，HMI 设备只能访问已存储密码的 CPU。仅可在 SIEMENS 的 HMI 设备上使用此功能。

### 步骤

1. 在项目树中打开“连接”(Connections) 编辑器。
2. 选择集成的连接。
3. 在“密码”(Password) 区域中输入 CPU 密码。



### 结果

HMI 设备现在可以与 CPU 通信并交换数据。