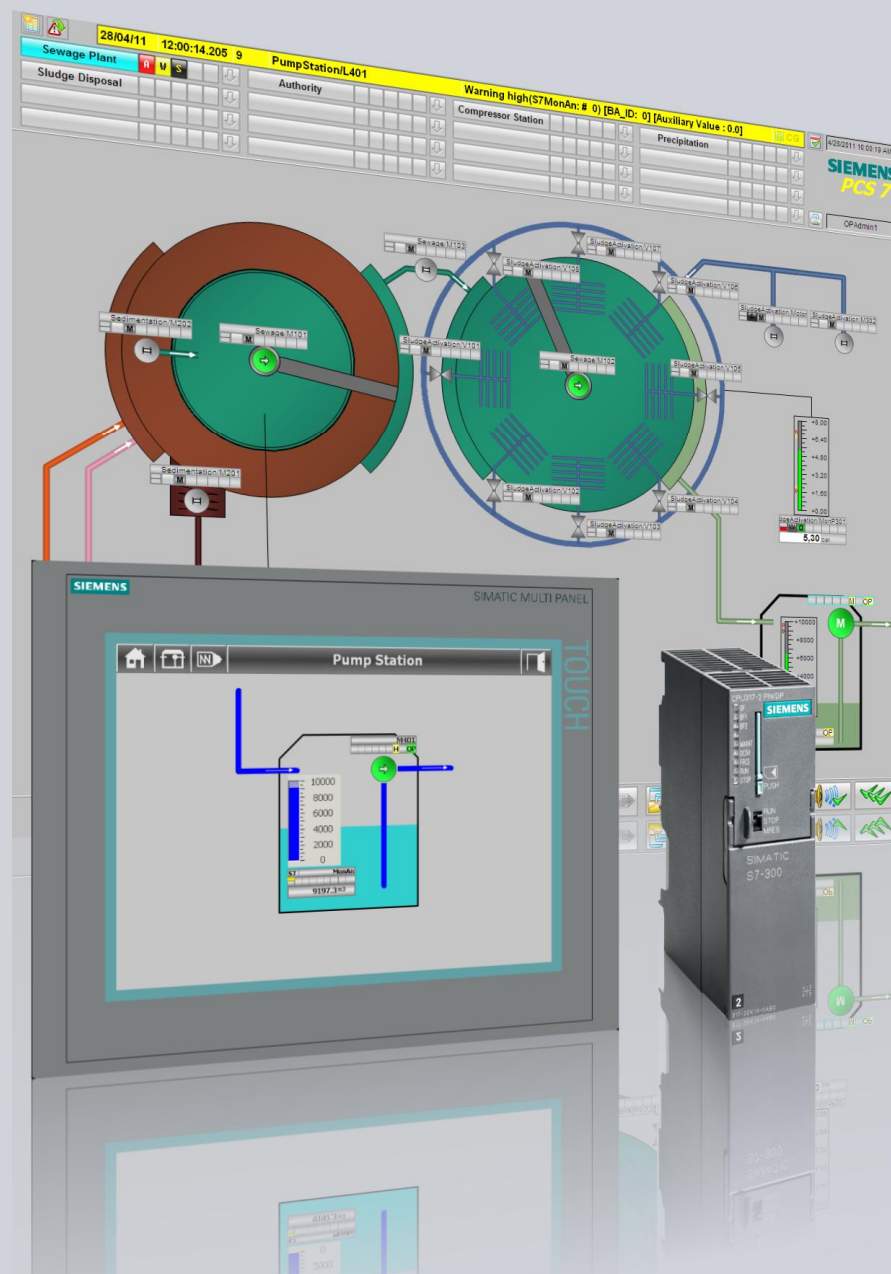


# 将 S7-300 整套设备集成到带有 PCS 7 行业库的 SIMATIC PCS 7 中

SIMATIC PCS 7

技术应用文章 • 2012 年 7 月



## 应用和工具

知其道，用其妙！

**SIEMENS**

## 西门子工业业务领域业务领域在线支持

本文档来源于西门子工业在线支持。以下链接直接去往本文档的下载页面：

<http://support.automation.siemens.com/WW/view/en/50708061>

如果您对本文档存在任何疑问，请发送电子邮件至：

[online-support.automation@siemens.com](mailto:online-support.automation@siemens.com)

# SIEMENES

“手册目录”将会插入到此页中。

## SIMATIC PCS 7

### 将 S7 - 300 整套设备集成到 PCS 7 系统中

应用示例

自动化任务

1

自动化解决方案

2

基本元素

3

基本工作步骤

4

S7 – 400 CPU 中的通信管理

5

使用 PCS 7 行业库组态 S7-300 CPU

6

链接与文献

7

版本历史

8

## 保证与责任

### 备注

本应用示例并不完备，也不局限于组态组态、设备以及任何突发事件。这些应用示例并不代表着特定于用户的解决方案；它们仅用于为典型应用提供支持。您对所述产品的正确使用负有全权责任。这些应用示例并不会免除您安全而专业地使用、安装、操作以及维修本设备的责任。当利用这些应用示例时，您应当意识到，西门子并不会对超出本责任条款的任何损害/索赔承担责任。我们保留随时对本应用示例作出更改而不作事先通知的权力。如果这些应用示例中所提供的建议与西门子的其它出版物（比如产品目录）出现偏差的话，以其它出版物中的内容为准。

我们不对本文档中所包含的信息承担任何责任。

无论根据任何合法原因，对于本应用示例中的例子、信息、程序、设计以及性能数据等的使用而引起的索赔，我们一概不予接受。此类除外责任不适用于强制责任，比如德国产品责任法的约束，在故意、重大过失的情况，或者导致生命、身体与健康受损，产品的质量保证，欺诈性隐瞒缺陷，以及违反合同根基的情况。违反实质性合同义务所造成的损害受限于可预见的损害，除了故意或者重大过失而导致的生命、人身和健康伤害之外，还包括合同中指明的典型损害。上述规定并不意味着您的损害举证负担发生变化。

在事先未经西门子工业部门书面授权的情况下，不得传播或者复制这些应用示例或者示例的摘录内容。

# 目录

保证与责任 .....	4
目录 .....	5
1 自动化任务 .....	9
2 自动化解决方案 .....	10
2.1 S7-400 CPU 中的通信管理 .....	11
2.2 利用行业库组态 S7-300 CPU .....	12
3 基本元素 .....	13
3.1 行业库 .....	13
3.2 时间同步 .....	16
3.3 多用户操作 .....	17
3.4 用户权限和用户说明 .....	18
3.5 信号概念 .....	19
4 基本工作步骤 .....	20
4.1 时间同步的组态 .....	20
4.1.a SIMATIC 环境中的时间同步 .....	21
4.1.b 操作员面板的时间同步 .....	22
4.2 多用户操作的组态 .....	24
5 S7 - 400 CPU 的通信管理 .....	28
5.1 核心功能的描述 .....	28
5.2 所使用的硬件以及软件组件 .....	30
5.3 S7-400 单 CPU 的组态 .....	31
5.3.a 组态未指定的 S7-连接 .....	31
5.3.b 创建数据块 .....	33
5.3.c 组态通信函数块 .....	35
5.3.d 程序逻辑与可视化 .....	36
5.4 S7-400H CPU 的组态 .....	38
5.4.a 在 S7-400H CPU 中组态未指定的连接 .....	39
5.4.b 创建数据块 .....	39
5.4.c 组态 S7-400H CPU 中的通信函数块 .....	40
5.4.d 程序逻辑与可视化 .....	41
6 组态带有 PCS 7 行业库的 S7- 300 CPU .....	42
6.1 核心功能的描述 .....	42
6.2 硬件与软件组件 .....	44
6.3 组态与参数赋值 .....	45
6.3.a 为整套设备创建一个新的项目 .....	45
6.3.b 组态工艺层次 .....	46
6.3.c 创建 S7 程序 .....	47

6.3.d 组态 OS.....	49
6.3.e 组态操作员面板 .....	52
7 链接与文献.....	55
8 版本历史.....	56

---







# 1 自动化任务

## 简介

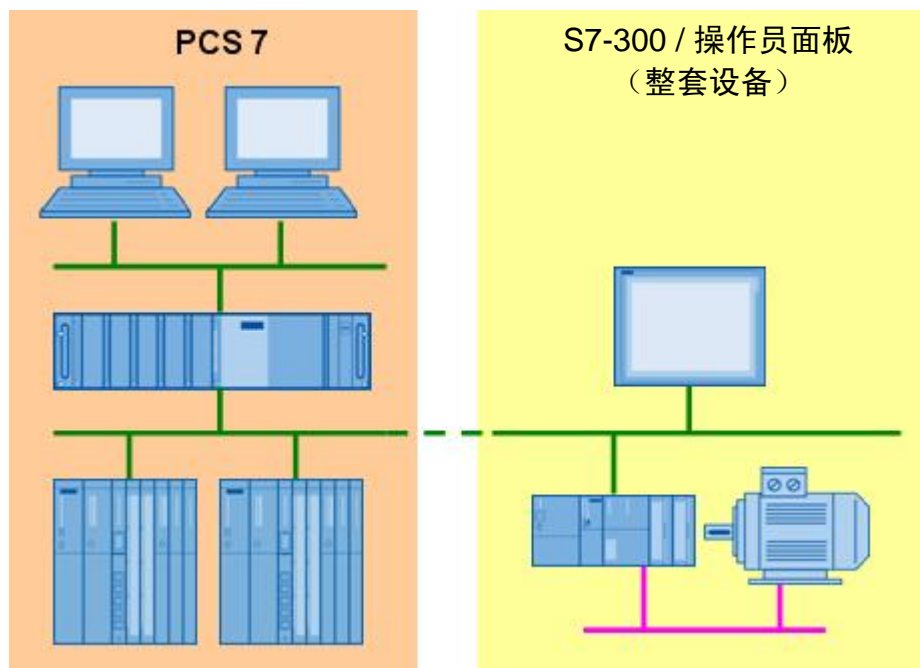
除了与工程过程直接相关的组件以外，许多生产工厂中包含有需要控制的其它机器和组合设备。在许多情况下，这些所谓的‘整套设备’(PU) 都是适用于特定生产任务的子模块，或者是进一步配备了非标准 PCS 7 系统组成部分的工厂组件，比如 S7-300 CPU 或者操作员面板等。举例来说，整套设备可以是离心机、干燥剂或者研磨工作站。

而当将这些整套设备与控制系统相连接时，可能会出现一些问题。各种类型的操作、显示以及处理方案以及所采用的不同警报系统，使得将整套设备集成到更高级别的过程控制系统变得更加复杂。

## 自动化任务的描述

本文档提供了多种选项用于将 S7-300 自动化系统以及操作员面板（HMI 装置）集成到 PCS 7 项目当中，同时组态设定应尽可能满足 PCS 7 标准。

图 1-1



## 2 自动化解决方案

### PCS 7 行业库 (IL) 的使用

“IL for S7” 行业库为 S7-300 CPU 和操作员面板提供了标准化的集成过程。行业库中所包含的工艺块适用于执行器和传感器的处理，这些执行器和传感器也适用于在 S7-300 自动化系统中运行。

这些工艺块支持消息发送功能 (Alarm\_DQ)，并且可通过 OS 进行操作和监控。操作员面板上的可视化可借助相关的界面块和 WinCC flexible 面板进行组态。

### PCS 7 行业库解决方案的优点

- 为过程任务提供综合且一致的解决方案，通过降低误操作风险从而让全过程实现最优运作
- 减少特定于用户的功能总数，从而节约了整个生命周期上的成本
- 在培训和专业知识转移上产生协同效应

### 连接 S7-300 整套设备 (PU) 的场景

本文档描绘了如下两种不同的连接场景：

- 通过利用 “S7Get” 和 “S7Put” 函数块在 S7-400 和 S7-300 之间进行通信（参见第五章“S7-400 CPU 中的通信管理”）
- 对包括面板在内的 S7-300 整套设备进行组态（参见第六章“利用 PCS 7 行业库对 S7-300 CPU 进行组态”）

最佳解决方案的选择取决于需要集成的 PU 是否固定完整的设备以及 PU 的控制程序是否可以修改或者在某种程度上进行补充。

### 限定

以下主题不在本文档的考虑范围之内：

- 与 S7-200、S7-1200、TIA Portal 的连接
- 来自第三方提供者的控制器连接。  
欲了解更多关于本自动化任务的详细信息，请参考以下文档，其中包含有可能的解决方案：  
<http://support.automation.siemens.com/WW/view/en/49740087>
- **PROFIBUS 连接**  
本文档仅涉及以太网连接 PROFIBUS 可以作为替代方案，仅在连接组态上有所差别。
- S7 函数块的编程
- 在 OS 和操作员面板上创建面板。  
预了解更多关于该主题的信息，请参考相关的 PCS 7 以及 WinCC flexible 文档。

## 2 Error! Style not defined.

### 2.1 Error! Style not defined.

#### 预备知识

要求基本了解在 PCS7 中进行系统组态以及在 WinCC flexible 中进行组态的方法。

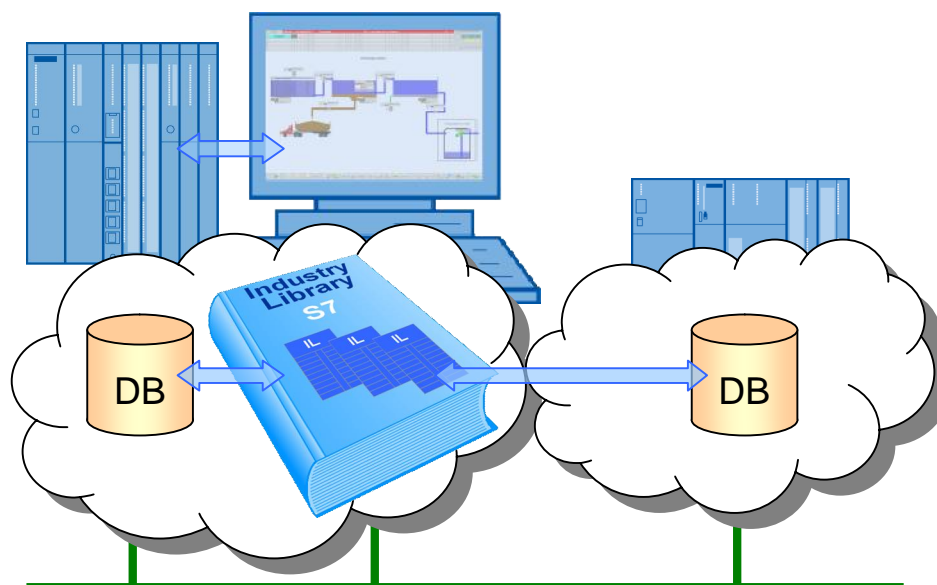
#### 替代方案

作为 PCS 7 行业库的替代方案，您还可以利用“S7 标准库”中的函数块以及您自己的函数块对 S7-300 程序进行组态。然而，这种方法需要花费更多的精力来准备控制程序以及 OS 和操作员面板上的可视化。此外，还无法保证其与 PCS 7 标准的兼容性，这可能会导致更多问题。

## 2.1 S7-400 CPU 中的通信管理

函数块“S7Get”用于从 S7-300 CPU 读取数值以及将这些数值传输至 PCS 7 系统中的 S7-400 CPU 数据块当中。这些数据会在 S7-400 CPU 中进行处理，然后重新传输回至 S7-300 设备数据块。数据在 S7-400 CPU 中的处理使其能够轻易集成到 PCS 7 OS 当中。

图 2-1



由于“S7Get”函数块和“S7Put”函数块均在 S7-400 CPU 中进行处理，因此该解决方案不需要对 S7-300 控制程序中的组态进行任何干预。如果有需要的话，可以将一些已定义的界面以数据块的形式进行集成。

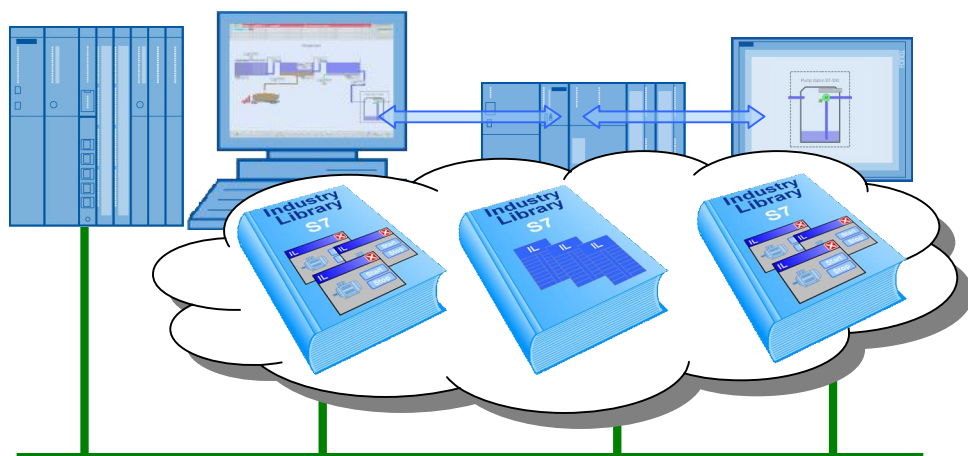
2 Error! Style not defined.

2.2 Error! Style not defined.

## 2.2 利用行业库组态 S7-300 CPU

在这一场景中，整套设备程序仅仅在 S7-300 CPU 中进行处理。借助工艺块以及 IL for S7 接口函数块，可在 PCS 7 OS 以及操作员面板上实现可视化。这种方法将会利用 IL 元素重新创建 PU 的控制程序。

图 2-2



### 3 Error! Style not defined.

#### 3.1 Error! Style not defined.

## 3 基本元素

### 3.1 行业库

本应用中所采用的 IL for S7 函数块库包括通信函数块、监视函数块、工艺函数块、操作函数块以及仿真函数块。这些函数块可以通过相关的面板进行操作和监控。

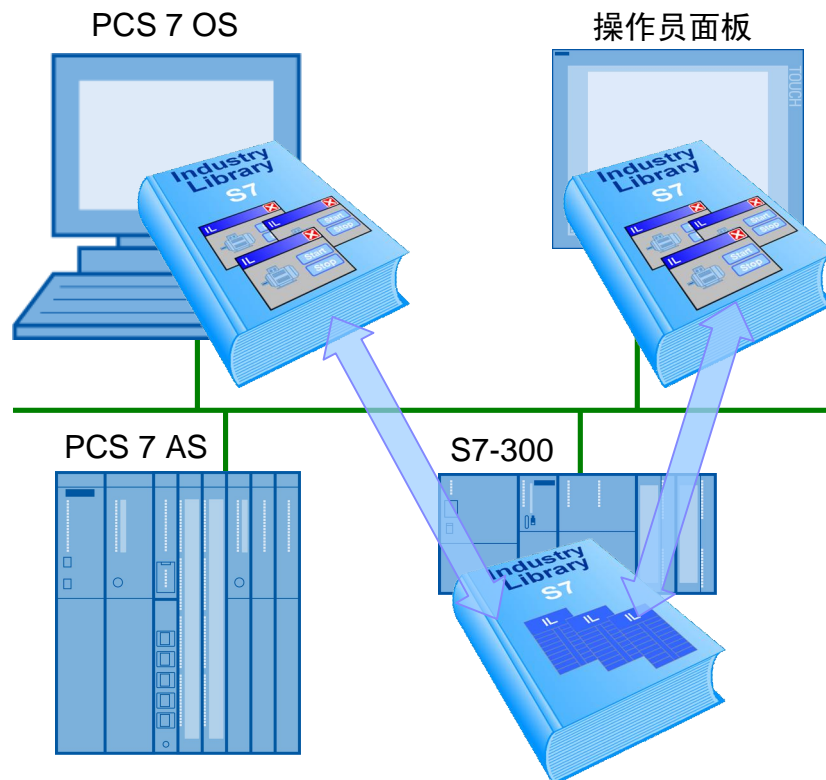
IL for S7 警报概念是基于系统函数块“ALARM\_DQ”。为了在操作员面板上实现可视化，可利用界面函数块来补充工艺功能。

借助于以下的库，可将 S7-300 CPU 和操作员面板集成到 PCS 7 环境当中。

表 3-1

库	说明
PCS 7 行业库 V8.0 (适用于 S7 的 IL)	使用在 S7-300 CPU 和面板中的工艺函数块，用于实现 OS 中的可视化。 WinCC flexible 面板库用于操作员面板上的显示和操作。

图 3-1



### 3 Error! Style not defined.

#### 3.1 Error! Style not defined.

##### 使用行业库的原因

- 通过标准化的方式将风险降至最低
- ‘外观和感觉’与 PCS 7 APL 相一致
- 易于集成 S7-300 以及 操作员面板
- 直接从 PCS 7 服务器访问 S7-300（无子网）
- 减少开发时间和成本
- 可随时升级至更高的 PCS 7 版本

##### 使用行业库的系统要求

PCS 7 行业库 V8.0 兼容以下组态软件：

表 3-2

库	组态软件
IL for S7	SIMATIC STEP 7 V5.5 SIMATIC S7 CFC SIMATIC WinCC V7.0 SP3 或者 SIMATIC PCS 7 V8.0 以及 WinCC flexible Advanced 2008 SP3
IL for PCS 7	SIMATIC PCS 7 V8.0 SIMATIC PCS 7 APL V8.0 WinCC flexible Advanced 2008 SP3

应使用下述硬件或者相关版本：

表 3-3

库	硬件
IL for S7	Ab CPU-314C-2 PN/DP 固件版本 $\geq$ V3.1 oder IM 151-8PN
IL for PCS 7	与 PCS 7 V8.0 的系统要求相同
IL for S7 / IL for PCS 7 (WinCC flexible)	MP 277 或 MP 377（显示尺寸 $\geq$ 10 英寸） SIMATIC IPC277D（显示尺寸 $\geq$ 12 英寸）

### 3 Error! Style not defined.

---

#### 3.1 Error! Style not defined.

带有行业库的 CPU-315 PN/DP 的计算建议

CPU-315 PN/DP 提供以下的相关资源:

- 主存储器: 384kB
- 同步消息: 330

(参见设备手册 “ CPU 31xC 与 CPU 31x: 技术数据”  
<http://support.automation.siemens.com/MW/view/en/12996906>)

---

假定混合配制中的下述程序部件均由 CPU-315 PN/DP 进行控制, 利用程度接近 60%。

- 20 个过程标签
- 20 个驱动器
- 2 台组合设备
- 2 个控制器

### 3 Error! Style not defined.

#### 3.2 Error! Style not defined.

## 3.2 时间同步

在 PCS 7 系统中，包括 PC 站点、自动化系统以及其它外围设备在内的所有组件的时钟时间均须实现同步。这是非常重要的一点，用以确保按照正确的时间顺序执行适当的过程序列以及消息存档。

### 步骤

时间同步可通过多种方式来执行，比如可定义域服务器或者中央系统时钟 (SICLOCK) 作为主时间。

欲了解更多关于时间同步的详细信息，请参考“[SIMATIC 过程控制系统——PCS 7 时间同步](#)”手册。

### 操作员面板的集成

为了避免时间上的不一致，操作员面板也应当进行同步。然而，操作员面板无法通过 SIMATIC 或者 NTP 步骤进行同步。

要实现操作员面板的同步，须先定义区域指针，用于实现面板系统时间与控制器系统时间之间的同步。控制程序会使用“READ\_CLK”功能来提供带有当前系统时间的区域指针。

本文档的后续部分将会进一步描述时间同步的组态设定。

### 请注意

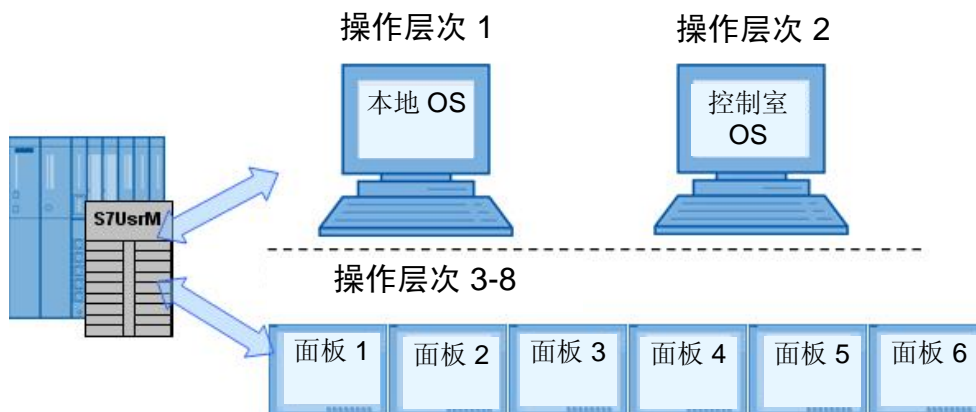
带有集成以太网接口的自动化系统要求使用 NTP 步骤（NTP = Network Time Protocol（网络时间协议））进行同步。



### 3.3 多用户操作

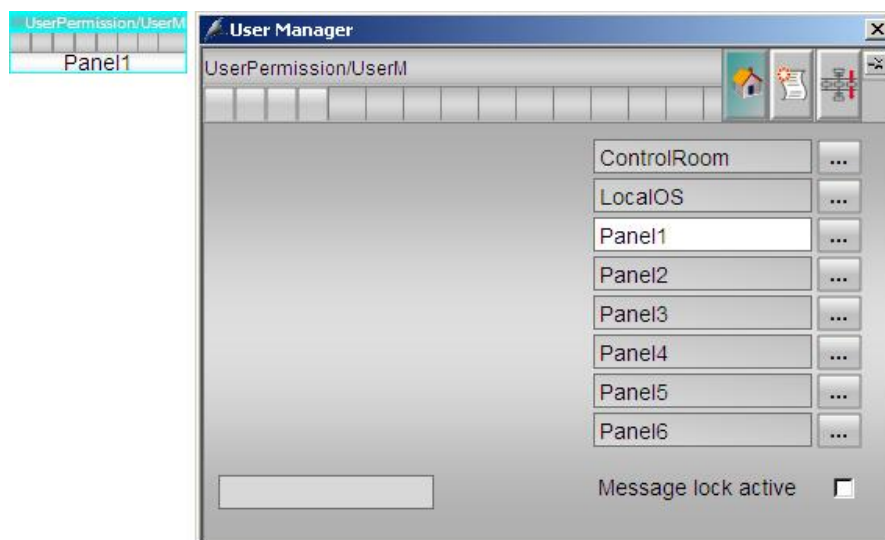
为了避免由于多个站点进行控制操作而造成的不一致，必须引入多用户操作功能。除了其它方面之外，多用户操作的概念包含了 2 个阶段的层次操作。层次 1 和层次 2 的操作由 OS 在控制室中完成。层次 3 至层次 8 的操作需要通过工厂中的操作员面板来完成。如果有需要的话，也可以对这 8 个操作层次单独进行组态。

图 3-2



不同的操作层次由集成于控制程序当中的“S7usrM”函数块进行管理，并通过工艺函数块和接口函数块互相连接以及连接至操作员面板。通过 OS 中的面板或者连接至输入“KS\_DEVICE”，操作层次的选择即可生效。激活了操作员管理功能之后，仅可由已登录且获得“较高级别过程控制”操作授权的用户来操作操作员面板。无法在操作员面板上选择操作层次。

图 3-3



在 OS 中，操作层次由内部标签“@Permission”进行定义。操作员面板的操作层次在接口函数块的“OP\_PERMIS”输入中进行组态。

本文档的后续部分会对多用户操作的组态设定进行描述。

### 3 Error! Style not defined.

#### 3.4 Error! Style not defined.

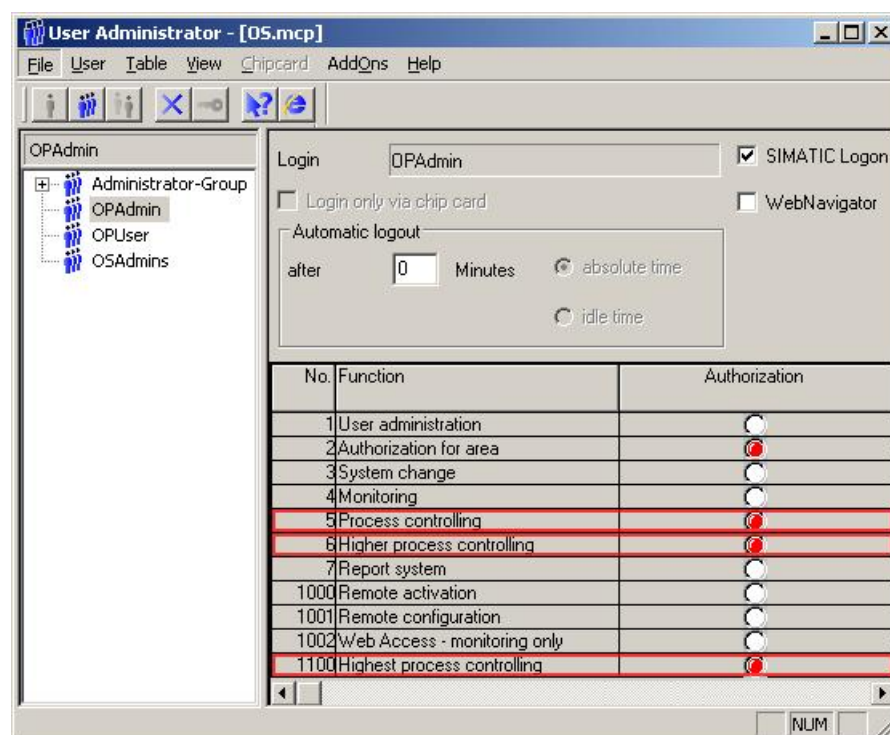
## 3.4 用户权限和用户说明

用于过程控制的 PCS 7 面板提供如下三种标准权限层次：

- 层次 5：过程控制  
允许执行常规控制操作，比如在手动和自动模式之间进行切换。
- 层次 6：较高级别过程控制  
允许对过程造成重大影响的控制操作，比如调节控制器的极限值。
- 层次 1100：最高级别过程控制  
允许对过程数值进行仿真以及出于维护目的而释放运作中的设备。

层次 1101 和 1102 的操作无须具备特定于项目的操作员权限。

图 3-4



欲了解 PCS 7 用户层次结构方面的更多详细信息，请参考“PCS 7 OS 过程控制”手册。

<http://support.automation.siemens.com/WW/view/en/36195920>

### 3 Error! Style not defined.

#### 3.5 Error! Style not defined.

操作员面板上的面板组态仅允许层次 5 以下的操作，更高或者最高级别的过程控制操作仅可在 OS 中执行。

通过指定 WinCC flexible Runtime 的用户管理权限，可以为操作员面板上的 IL 面板访问定义更多限制。某些设备为用户管理提供了 SIMATIC 登录的选项。

行业库所建议使用的下述设备也支持 SIMATIC 登录：

- MP 277, MP 377, SIMATIC IPC
- 带有 WinCC flexible advanced 2008 Runtime 的 PC 平台

对于其它所有不含 SIMATIC 登录的操作员面板，需要对本地用户管理系统进行定义。用户管理系统的组态过程在关于 PCS 7, WinCC flexible 以及 SIMATIC 登录的文档中均有描述。

## 3.5 信号概念

IL for S7 库中的工艺函数块采用系统函数块“Alarm\_DQ”来将组出错信息发送至 OS 以及操作员面板。警报消息可以在这两个站定中进行确认。

与 PCS 7 标准类型的警报信号函数块“Alarm\_8P”相比，“Alarm\_DQ”信号功能还可以在 S7-300 CPU 以及基于 WinCC flexible 的 HMI 设备上使用。但是，每次调用“Alarm\_DQ”仅会生成一条消息。

警报和消息的类型和大小：

表 3-4

	带有 S7-400 的 PCS 7	带有 S7-300 的 PCS 7
警报信号函数块	ALARM_8P / ALARM_DQ	ALARM_DQ
消息数量	高达 1000	高达 300

欲了解更多关于警报信号函数块的信息，请参考“S7-300/400 系统的系统软件以及标准功能”手册。

<http://support.automation.siemens.com/WW/view/en/44240604>

#### 请注意

目前尚不支持通过操作员面板创建操作员消息。如果对这一功能有要求，则必须单独进行组态，比如借助警报信号函数块登记来自操作员面板接口函数块的开关信号，然后发布一条操作员消息。

## 4 基本工作步骤

**请注意** 本章中所描述的工作步骤与 S7-400 CPU 中的通信组态无关。如有需要，请继续阅读第 5 章“S7-400 CPU 中的通信管理”。

### 4.1 时间同步的组态

**请注意** 以下的章节内容仅适用于在 SIMATIC 环境中的时间同步。请注意，并非所有组件均支持 SIMATIC 环境中的时间同步。带有集成以太网接口的自动化系统仅可通过 NTP 步骤实现同步。由于操作员面板不支持这两种步骤，因此必须借助区域指针对它们进行同步，同时需要控制器提供当前系统时间。

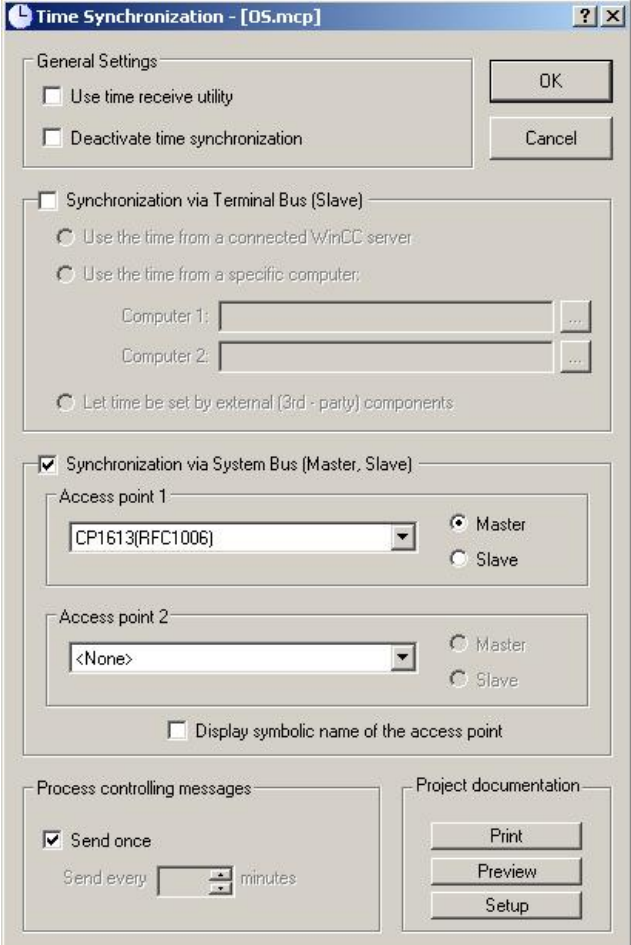
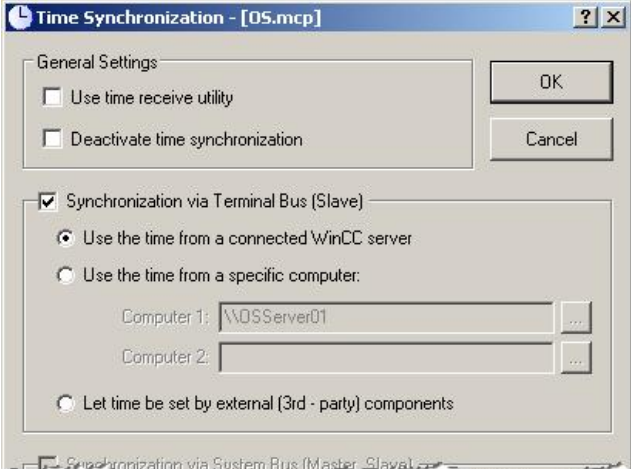
OS 服务器定义为主时间。OS 客户端以及自动化系统均定义为从属时间。通过区域指针，操作员面板会与 AS 实现同步。整个系统的时钟时间应设为 UTC（协调通用时间）。

## 4 Error! Style not defined.

### 4.1 Error! Style not defined.

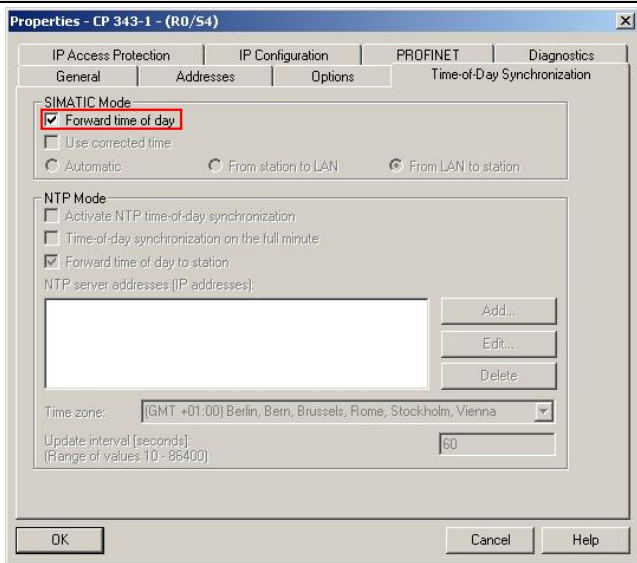
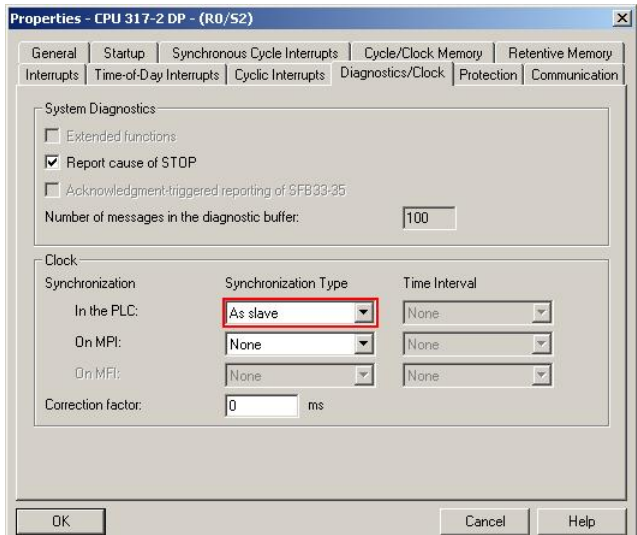
#### 4.1.a SIMATIC 环境中的时间同步

表 4-1

序号	操作	显示
1.	<p>设定主时间</p> <ul style="list-style-type: none"> <li>打开服务器中需要组态为时间主站的 OS 项目。</li> <li>打开“Time Synchronization (时间同步)”编辑器</li> <li>选择“Synchronization via System Bus (通过系统总线进行同步)”选项。</li> <li>选择一个访问点并将其定义为“Master (主时间)”。选择系统总线的 CP。</li> <li>如果有需要的话,您可以再将一个访问点定义为“主时间”。</li> <li>保存更改,然后下载该 OS。</li> </ul>	
2.	<p>组态 OS 客户端的时间同步</p> <ul style="list-style-type: none"> <li>逐个打开所有 OS 客户端的项目。</li> <li>打开“Time Synchronization (时间同步) 编辑器”</li> <li>选择“Synchronization via Terminal Bus (通过终端总线进行同步) 选项。”</li> <li>选择“Use the time from a connected WinCC server (使用来自所连接的 WinCC 服务器的时间)”。</li> <li>保存更改,然后下载该 OS。</li> </ul>	

## 4 Error! Style not defined.

### 4.1 Error! Style not defined.

序号	操作	显示
3.	<p>自动化系统中的时间同步</p> <ul style="list-style-type: none"><li>打开需要组态的 AS 的硬件组态窗口。</li></ul> <p>CP 设定:</p> <ul style="list-style-type: none"><li>打开该 CP 的属性对话框，然后选择“Time-of-Day Synchronization（日期时间同步）”选项卡。</li><li>选择 SIMATIC Mode（SIMATIC 模式）字段中的“Forward time of day（转发日期时间）”选项。</li></ul> <p>CPU 设定:</p> <ul style="list-style-type: none"><li>打开该 CPU 的属性对话框，然后选择“Diagnostics/Clock（诊断/时钟）”选项卡。</li><li>选择同步类型为“<b>As slave</b>（作为从属时间）”。</li><li>保存并编译所作的更改，然后下载硬件组态。</li></ul>	 <p>The screenshot shows the 'Properties - CP 343-1 - (R0/S4)' dialog box. The 'Time-of-Day Synchronization' tab is active. Under 'SIMATIC Mode', the 'Forward time of day' checkbox is checked and highlighted with a red box. Other options include 'Use corrected time' (unchecked), 'Automatic' (selected radio button), 'From station to LAN' (unselected), and 'From LAN to station' (unselected). Under 'NTP Mode', 'Activate NTP time-of-day synchronization' and 'Time-of-day synchronization on the full minute' are unchecked, while 'Forward time of day to station' is checked. There is a text field for 'NTP server addresses (IP addresses)' with 'Add...', 'Edit...', and 'Delete' buttons. The 'Time zone' is set to '(GMT +01:00) Berlin, Bern, Brussels, Rome, Stockholm, Vienna'. The 'Update interval [seconds]' is set to 60.</p>  <p>The screenshot shows the 'Properties - CPU 317-2 DP - (R0/S2)' dialog box. The 'Diagnostics/Clock' tab is active. Under 'System Diagnostics', 'Report cause of STOP' is checked. The 'Number of messages in the diagnostic buffer' is set to 100. Under 'Clock', the 'Synchronization Type' for 'In the PLC' is set to 'As slave' and highlighted with a red box. Other synchronization types for 'On MPI' and 'On MFI' are set to 'None'. The 'Correction factor' is set to 0 ms.</p>

#### 4.1.b 操作员面板的时间同步

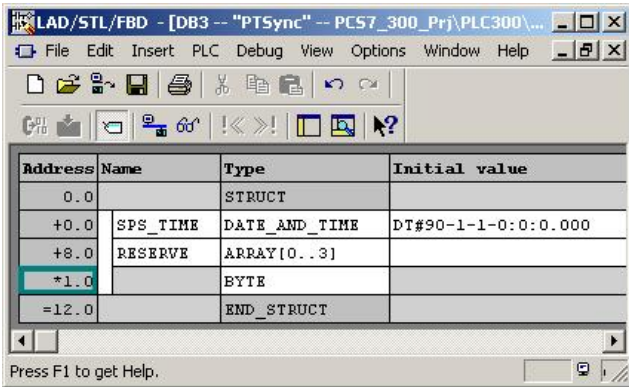
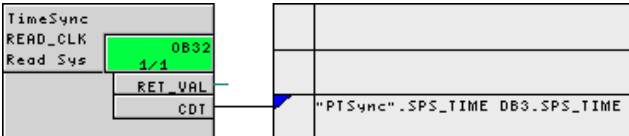
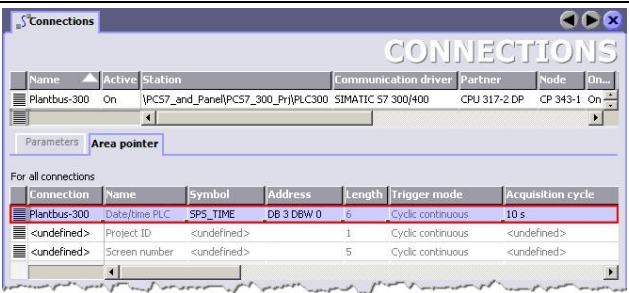
操作员面板的时间同步要求:

- 系统函数块“READ\_CLK”
- 一个 12 字节的数据块
- 适用于操作员面板的“Date/time PLC（日期/时间 PLC）”区域指针

## 4 Error! Style not defined.

### 4.1 Error! Style not defined.

表 4-2

序号	操作	显示
1.	<p>创建一个数据块</p> <p>面板区域指针要求一个 12 字节的数据区。利用以下参数创建一个数据块:</p> <ul style="list-style-type: none"> <li>• 一个类型为 “DATE_AND_TIME” 的参数</li> <li>• 4 个保留字节</li> </ul>	
2.	<p>读取时钟时间</p> <p>借助 SFC “READ_CLK” 读取时钟时间并写入至 DB 参数 “SPS_TIME”。</p> <ul style="list-style-type: none"> <li>• 创建一个新的 CFC 图表。</li> <li>• 将 SFC “READ_CLK” 添加至图表。</li> <li>• 将 “CDT” 输出连接至 DB 的 “SPS_TIME” 参数。</li> <li>• 编译并下载该控制程序。</li> </ul> <p>请注意 该块的调用间隔设为 1 秒便已足够 (OB32)。</p>	
3.	<p>组态区域指针</p> <ul style="list-style-type: none"> <li>• 打开 WinCC flexible 项目中的 Communication (通信) 文件夹, 然后选择需要用于同步的 CPU 连接。</li> <li>• 打开 “Area pointer (区域指针)” 选项卡。</li> <li>• 将区域指针 “Date/time PLC (日期/时间 PLC)” 设定为先前所创建的数据块的连接和地址。</li> <li>• 传输 WinCC flexible 项目。</li> </ul>	

## 4 Error! Style not defined.

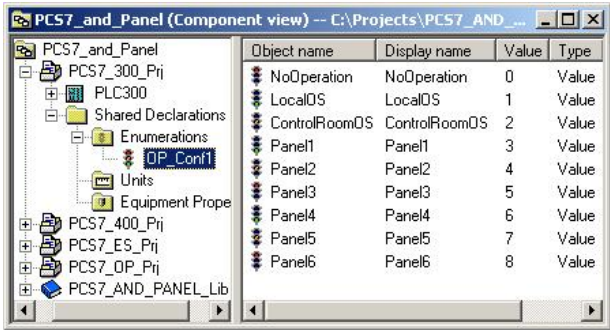
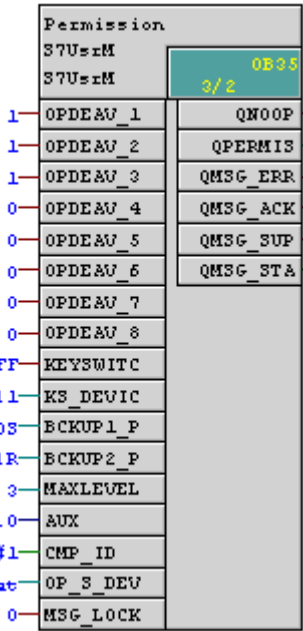
### 4.2 Error! Style not defined.

## 4.2 多用户操作的组态

本章描述组态多用户操作功能所需的全部步骤。

组态过程如下：

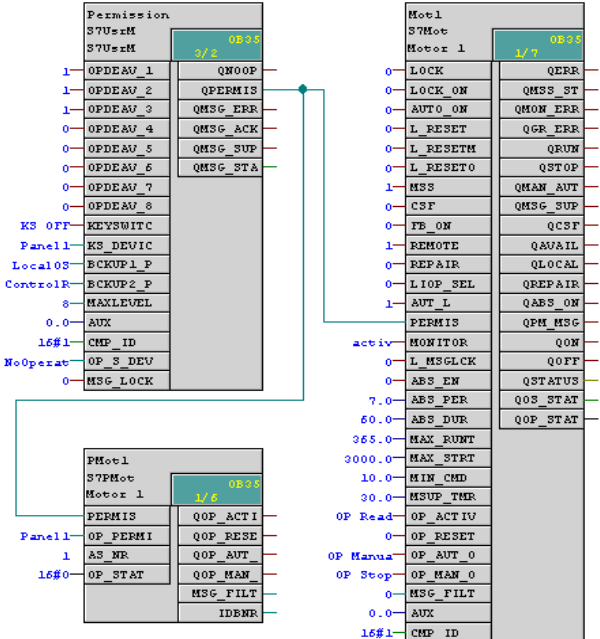
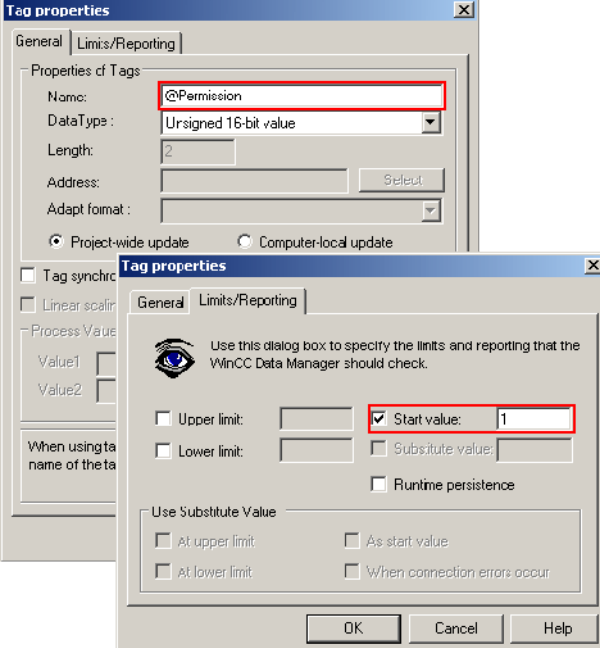
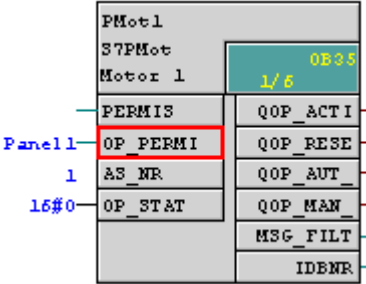
表 4-3

序号	操作	显示																																								
1.	<p>创建枚举列表</p> <p>在操作层次连接中，IL S7 库的函数块已经预先组态在枚举列表“OP_Conf1”中。在此，CFC 图表中的函数块连接不仅显示为枚举数值，而且还显示了操作层次的名称。此外，还生成了 OS 的文本应用并以符号标示在面板中。可以自由选择操作层次的名称。</p> <ul style="list-style-type: none"> <li>创建枚举列表“OP_Conf1”。</li> <li>组态列表对象的值为 0-8。</li> <li>规定数值“0”为“not operated（不操作）”。</li> <li>利用数值 1-8 来定义操作层次名称。</li> </ul>	 <table border="1"> <thead> <tr> <th>Object name</th> <th>Display name</th> <th>Value</th> <th>Type</th> </tr> </thead> <tbody> <tr><td>NoOperation</td><td>NoOperation</td><td>0</td><td>Value</td></tr> <tr><td>LocalOS</td><td>LocalOS</td><td>1</td><td>Value</td></tr> <tr><td>ControlRoomOS</td><td>ControlRoomOS</td><td>2</td><td>Value</td></tr> <tr><td>Panel1</td><td>Panel1</td><td>3</td><td>Value</td></tr> <tr><td>Panel2</td><td>Panel2</td><td>4</td><td>Value</td></tr> <tr><td>Panel3</td><td>Panel3</td><td>5</td><td>Value</td></tr> <tr><td>Panel4</td><td>Panel4</td><td>6</td><td>Value</td></tr> <tr><td>Panel5</td><td>Panel5</td><td>7</td><td>Value</td></tr> <tr><td>Panel6</td><td>Panel6</td><td>8</td><td>Value</td></tr> </tbody> </table>	Object name	Display name	Value	Type	NoOperation	NoOperation	0	Value	LocalOS	LocalOS	1	Value	ControlRoomOS	ControlRoomOS	2	Value	Panel1	Panel1	3	Value	Panel2	Panel2	4	Value	Panel3	Panel3	5	Value	Panel4	Panel4	6	Value	Panel5	Panel5	7	Value	Panel6	Panel6	8	Value
Object name	Display name	Value	Type																																							
NoOperation	NoOperation	0	Value																																							
LocalOS	LocalOS	1	Value																																							
ControlRoomOS	ControlRoomOS	2	Value																																							
Panel1	Panel1	3	Value																																							
Panel2	Panel2	4	Value																																							
Panel3	Panel3	5	Value																																							
Panel4	Panel4	6	Value																																							
Panel5	Panel5	7	Value																																							
Panel6	Panel6	8	Value																																							
2.	<p>组态“S7UsrM”</p> <ul style="list-style-type: none"> <li>将函数块“S7UsrM”添加至新的或者先前的 CFC 图表。</li> <li>定义是否允许在输入“OPDEAV_1”至“OPDEAV_8”中选择层次。</li> <li>利用输入“KEYSWITCH”来规定应在 OS 处选择操作层次还是在函数块输入“KS_DEVICE”处预先进行定义。</li> <li>利用输入“MAXLEVEL”来定义上述已激活的层次数量。</li> </ul>	 <table border="1"> <thead> <tr> <th>Input</th> <th>Value</th> </tr> </thead> <tbody> <tr><td>37UsrM</td><td>OB 3 5</td></tr> <tr><td>37UsrM</td><td>3/2</td></tr> <tr><td>OPDEAV_1</td><td>QN00P</td></tr> <tr><td>OPDEAV_2</td><td>QPERMIS</td></tr> <tr><td>OPDEAV_3</td><td>QMSG_ERR</td></tr> <tr><td>OPDEAV_4</td><td>QMSG_ACK</td></tr> <tr><td>OPDEAV_5</td><td>QMSG_SUP</td></tr> <tr><td>OPDEAV_6</td><td>QMSG_STA</td></tr> <tr><td>OPDEAV_7</td><td></td></tr> <tr><td>OPDEAV_8</td><td></td></tr> <tr><td>KEYSWITC</td><td></td></tr> <tr><td>KS_DEVICE</td><td></td></tr> <tr><td>BCKUP1_P</td><td></td></tr> <tr><td>BCKUP2_P</td><td></td></tr> <tr><td>MAXLEVEL</td><td>3</td></tr> <tr><td>AUX</td><td>0.0</td></tr> <tr><td>CMP_ID</td><td>15#1</td></tr> <tr><td>OP_S_DEV</td><td>NoOperat</td></tr> <tr><td>MSG_LOCK</td><td>0</td></tr> </tbody> </table>	Input	Value	37UsrM	OB 3 5	37UsrM	3/2	OPDEAV_1	QN00P	OPDEAV_2	QPERMIS	OPDEAV_3	QMSG_ERR	OPDEAV_4	QMSG_ACK	OPDEAV_5	QMSG_SUP	OPDEAV_6	QMSG_STA	OPDEAV_7		OPDEAV_8		KEYSWITC		KS_DEVICE		BCKUP1_P		BCKUP2_P		MAXLEVEL	3	AUX	0.0	CMP_ID	15#1	OP_S_DEV	NoOperat	MSG_LOCK	0
Input	Value																																									
37UsrM	OB 3 5																																									
37UsrM	3/2																																									
OPDEAV_1	QN00P																																									
OPDEAV_2	QPERMIS																																									
OPDEAV_3	QMSG_ERR																																									
OPDEAV_4	QMSG_ACK																																									
OPDEAV_5	QMSG_SUP																																									
OPDEAV_6	QMSG_STA																																									
OPDEAV_7																																										
OPDEAV_8																																										
KEYSWITC																																										
KS_DEVICE																																										
BCKUP1_P																																										
BCKUP2_P																																										
MAXLEVEL	3																																									
AUX	0.0																																									
CMP_ID	15#1																																									
OP_S_DEV	NoOperat																																									
MSG_LOCK	0																																									



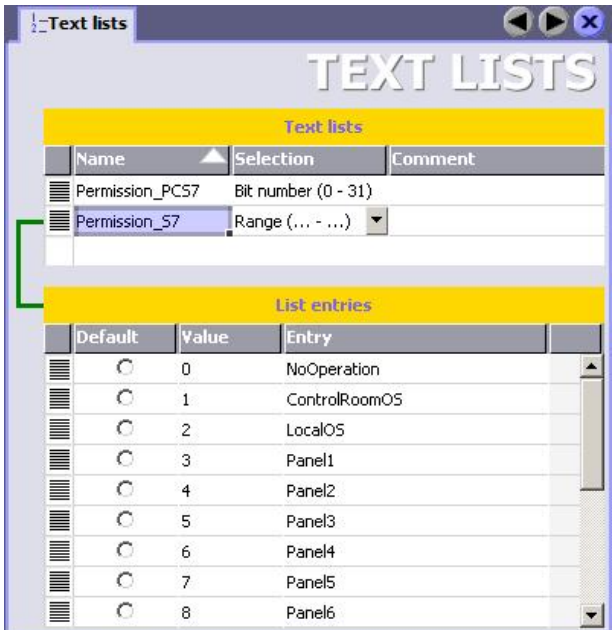
## 4 Error! Style not defined.

### 4.2 Error! Style not defined.

序号	操作	显示
3.	<p><b>S7UsrM 的互相连接</b></p> <p>将输出“QPERMIS”与面板接口函数块的输入“PERMIS”以及工艺函数块的输入“PERMIS”相连。</p> <p>为组态中所用到的每个函数块重复这一步骤。</p>	
4.	<p><b>定义操作层次</b></p> <p>OS 操作层次由内部标签“@Permission”进行定义。操作员面板的操作层次在相关操作员面板的接口函数块中进行定义。</p> <ul style="list-style-type: none"> <li>为每个 OS 创建一个“@Permission”标签，并将相关操作层次的数值定义为起始值。</li> <li>在接口函数块的输入“OP_PERMIS”中指定面板的操作层次。</li> </ul> <p><b>请注意</b></p> <p>如果工艺函数块中的“PERMIS”值与标签值“@Permission”相符，则可以在 OS 中对面板进行操作。</p> <p>如果输入“PERMIS”中所显示的数值与接口函数块中的“OP_PERMIS”数值相同，则可以在操作员面板中对面板进行操作。</p>	 

#### 4 Error! Style not defined.

#### 4.2 Error! Style not defined.

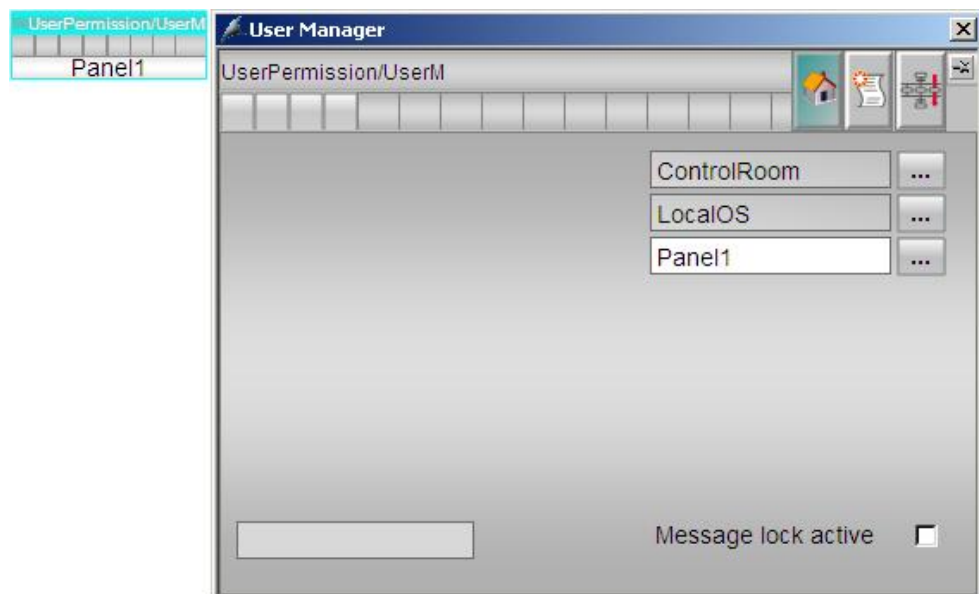
序号	操作	显示
5.	<p>组态操作员面板上的操作层次文本</p> <p>创建文本列表的同时会插入 IL 面板。这些文本中显示了面板中当前已选择的操作层次。如果已经为操作层次赋予了不同的名称，则您可以相应地重新组态这些文本列表。</p> <ul style="list-style-type: none"> <li>• 打开保存在“Text and Graphics Lists (文本与图形列表)”文件夹中的 WinCC-flexible 文本列表。</li> <li>• 选择文本列表“Permission_S7”。</li> <li>• 按照相应的数值重新命名操作层次。</li> </ul>	

#### 运行时操作

在运行时的情况下，您可以选择 OS 的操作层次。如果激活了用户管理功能，则仅可由具备“最高级别过程控制”权限的注册用户来执行这一操作。

仅可选择在相关函数块中激活的层次（“OPDEAV\_1”至“OPDEAV\_8”的输入）。

图 4-1



## 4 Error! Style not defined.

### 4.2 Error! Style not defined.

#### 面板中的显示

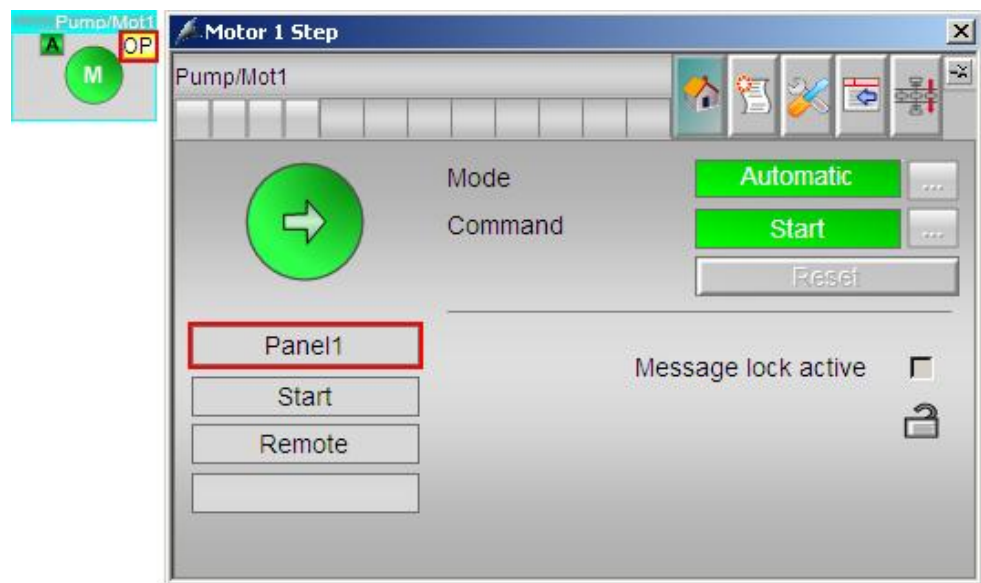
OS 中的函数块图标以及面板显示了站点及其当前优先级。

函数块图标可表示以下信息：

- 层次 1：不显示 – 本地 OS
- 层次 2：CR – 控制室
- 层次 3-8：OP – 操作员面板

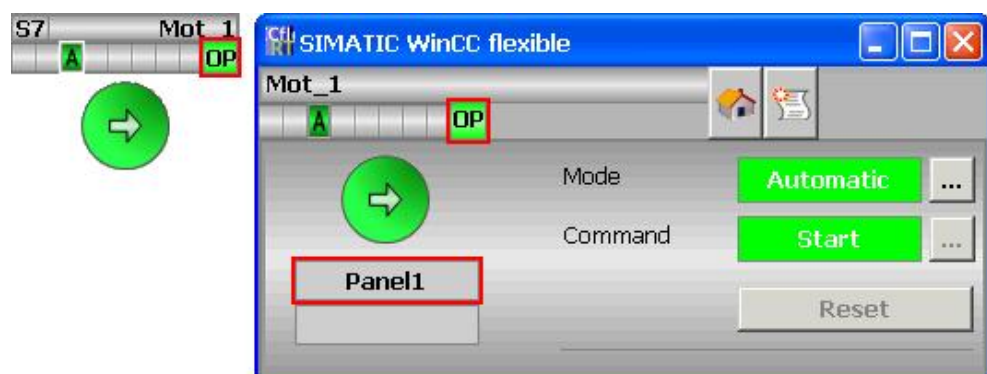
面板中显示了枚举列表中的文本。

图 4-2



所选择的权限层次还会按照 OS 的显示方式显示在操作员面板的面板中。

图 4-3



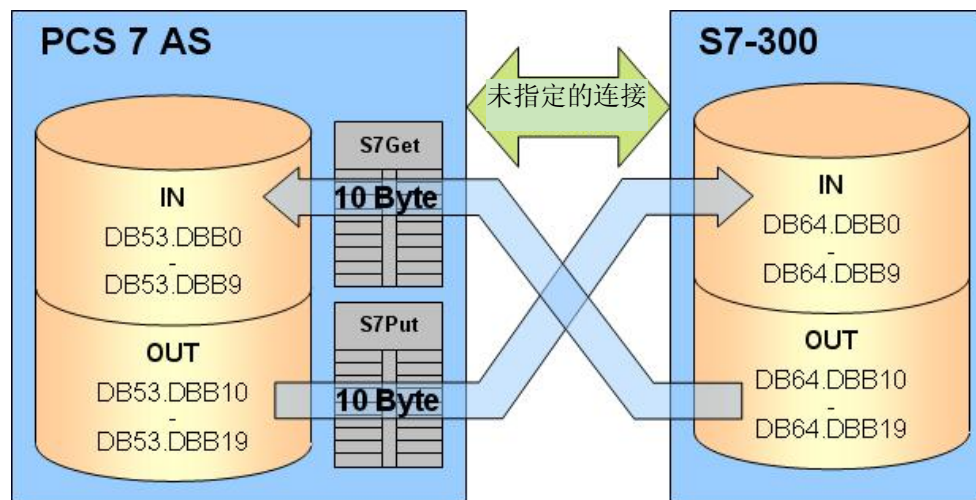
## 5 S7 – 400 CPU 的通信管理

### 5.1 核心功能的描述

本例中使用 IL S7 库中的“S7Put”以及“S7Get”通信函数块。这些函数块集成在自动化系统 (AS) 的 S7 程序中，为 AS 中的 DB 提供 S7-300 CPU 所需的全部数据。这些数据随后会分开单独处理。本章内容并不涉及时间同步的组态以及多用户操作。

借助“S7Get”函数块，可以读取来自 S7-300 CPU 的数据块数值，然后将其传输至集成在 PCS 7 系统中的 S7-400 CPU 数据块当中。这些数据会在 S7-400 CPU 中进行处理，然后通过函数块“S7Put”重新传输回至 S7-300 CPU 的数据块。

图 5-1



“S7Put”和“S7Get”均是消息类型的数据块，用于将关于连接状态的信息传输至 OS。由于数据在 AS 中进行处理，所以部分处理单元可以轻松实现可视化。

## 5 Error! Style not defined.

### 5.1 Error! Style not defined.

#### 核心功能的原理

表 5-1

序号	操作	显示
1.	借助函数块“S7Get”，S7-300 CPU 数据块中的数值会被传输至自动化系统的数据块当中。	
2.	所接收到的过程数值会在 PCS 7 的自动化系统中进行处理，然后可在 OS 中用于显示和操作。	
3.	函数块“S7Put”用于将当前数值重新传输至 S7-300 CPU。	

## 5 Error! Style not defined.

### 5.2 Error! Style not defined.

#### 该解决方案的优点

部分处理单元并不需要重新进行组态。仅必须知悉包含有待处理数据的 S7-300 CPU 数据块。无须在组态计算机上为 S7-300 CPU 安装特殊的函数块库。

## 5.2 所使用的硬件以及软件组件

本应用的创建采用了下述组件：

#### 硬件组件

表 5-2

组件	数量	MLFB/订单号	备注
CPU 416-3 PN/DP	1	6ES7 416-3ER05-0AB0	
CPU 417-4H	1	6ES7 417-4HT14-0AB0	
CPU 317-2 DP	1	6ES7-317-2AJ10-0AB0	

#### 标准软件组件

表 5-3

组件	数量	MLFB/订单号	备注
PCS 7 V7.1 SP2	1	6ES7-658-5AC17-0YA5	
PCS 7 行业库 V8.0	1	6DL5-410-8AA08-0YA0	

## 5 Error! Style not defined.

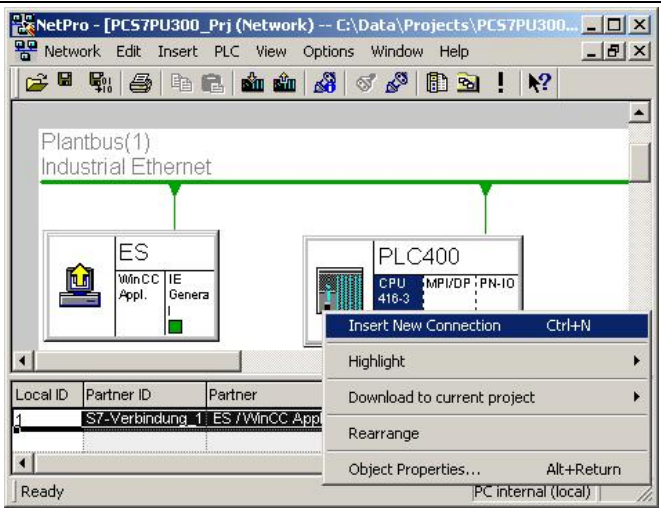
### 5.3 Error! Style not defined.

## 5.3 S7-400 单 CPU 的组态

### 5.3.a 组态未指定的 S7-连接

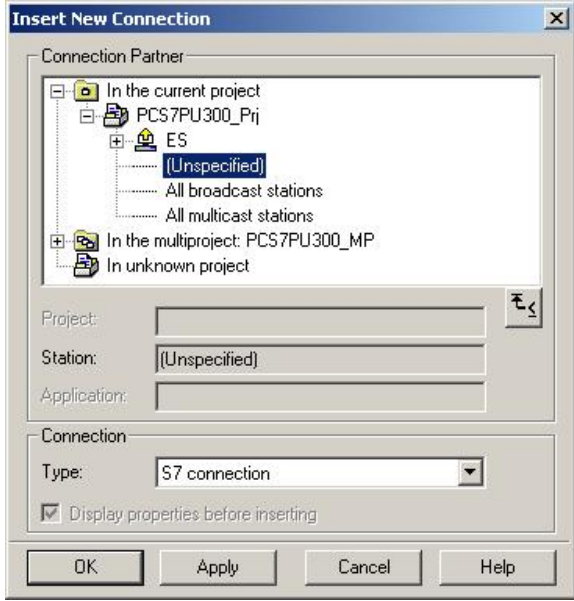
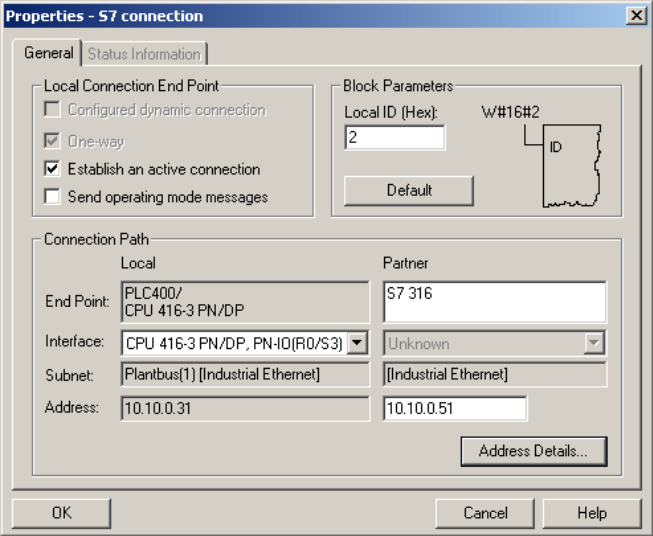
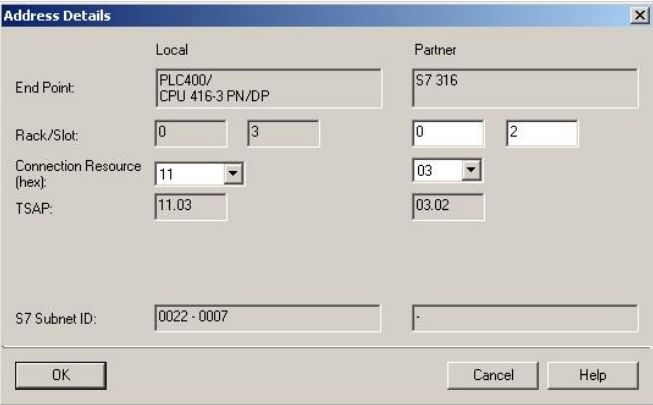
开始进行组态之前，请保证已将两个系统网络进行物理连接，或者这两个 CPU 已经相应地集成到同一个系统网络当中。连接的类型并不重要（现场总线或者工业以太网的方式均可），因为仅在所选择的接口以及地址上有所不同。按照以下步骤进行操作，在自动化系统与 S7-300 CPU 之间建立连接。

表 5-4

序号	操作	注释
1.	打开当前 PCS-7 项目中的“NetPro”编辑器。	点击 SIMATIC 管理器工具栏上的“Configure network（组态网络）”图标。
2.	指定需要与 S7-300 CPU 相连的 CPU，然后在右键菜单中选择“Insert New Connection（插入新的连接）”。	

## 5 Error! Style not defined.

### 5.3 Error! Style not defined.

序号	操作	注释
3.	<p>在 PCS-7 项目中选择“(Unspecified)”条目作为连接对象，然后选择“S7 connection (S7 连接)”作为连接类型。点击“OK”以确认设定。</p>	
4.	<p>打开“Properties – S7 connection (属性 – S7 连接)”对话框，然后设定连接对象的名称和网络地址。</p> <p>本地 ID 会自动设置，而无须进行编辑。该 ID 稍后将在 S7 程序中使用，以指定通信函数块。</p> <p>点击“Address Details... (地址详细信息...)”按钮。</p>	
5.	<p>在“Address Details (地址详细信息)”对话框中指定对象 CPU 的机架号码和插槽。</p> <p>连接资源默认为“03”，代表一端连接至一个未指定的连接对象。</p> <p>在保存/编译以及下载该连接之前，请先点击 OK 以确认所作的设定。</p>	



## 5 Error! Style not defined.

### 5.3 Error! Style not defined.

#### 5.3.b 创建数据块

从数据库中读取的数据会在自动化系统的 S7 程序中进行处理，处理结果会被重新传回原来的数据块中。为了确保 S7-300 CPU 中的数据传输正确性，需要在自动化系统中使用一个由用户自定义的数据块，该数据块必须兼容于 S7-300 CPU 中的读取/写入数据区域。

注意

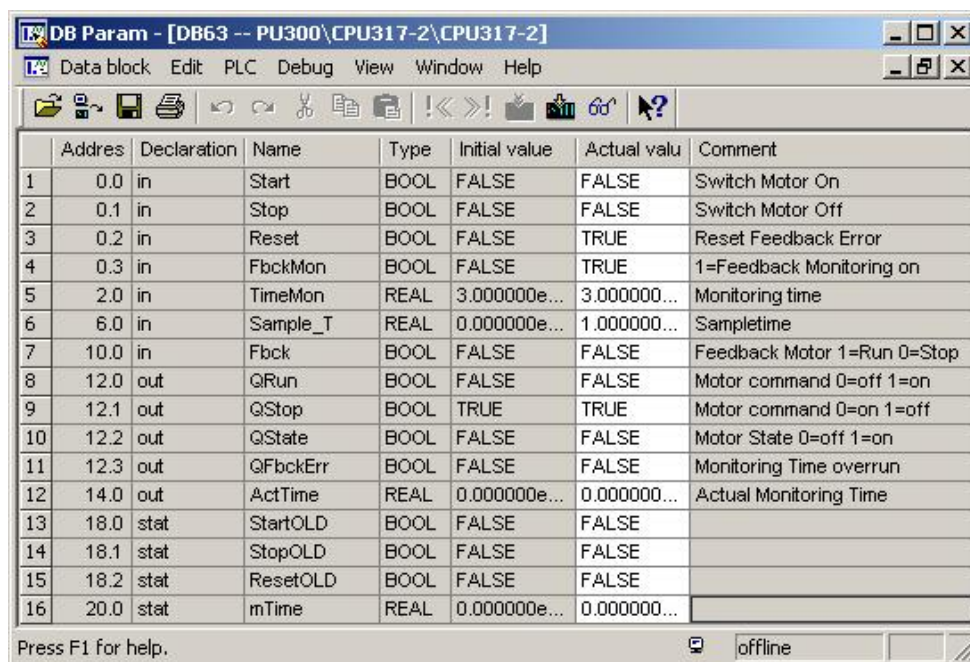
在使用函数块“S7Put”时，所有数据都会在不确认的情况下写入到 S7-300 CPU 的数据块当中。组态程序的时候请谨记这一点，因为这可能导致不希望出现的系统情况。

另外，如果目标数据块被互相连接，还可能会出现数值被覆盖的情况。互连数据块的输入和输出不应被 S7Put 所覆盖。

示例

一个用于控制简单电机的函数块已经集成到 S7-300 CPU 的 S7 程序当中。下图示出了相关数据块的参数。

图 5-2



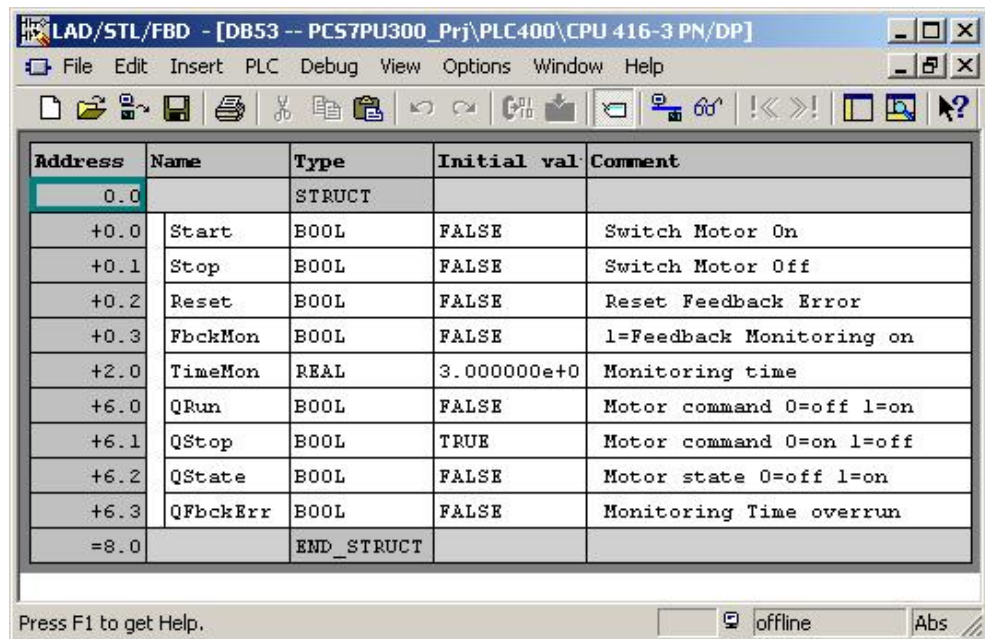
Address	Declaration	Name	Type	Initial value	Actual value	Comment
1	0.0 in	Start	BOOL	FALSE	FALSE	Switch Motor On
2	0.1 in	Stop	BOOL	FALSE	FALSE	Switch Motor Off
3	0.2 in	Reset	BOOL	FALSE	TRUE	Reset Feedback Error
4	0.3 in	FbckMon	BOOL	FALSE	TRUE	1=Feedback Monitoring on
5	2.0 in	TimeMon	REAL	3.000000e...	3.000000...	Monitoring time
6	6.0 in	Sample_T	REAL	0.000000e...	1.000000...	Sampletime
7	10.0 in	Fbck	BOOL	FALSE	FALSE	Feedback Motor 1=Run 0=Stop
8	12.0 out	QRun	BOOL	FALSE	FALSE	Motor command 0=off 1=on
9	12.1 out	QStop	BOOL	TRUE	TRUE	Motor command 0=on 1=off
10	12.2 out	QState	BOOL	FALSE	FALSE	Motor State 0=off 1=on
11	12.3 out	QFbckErr	BOOL	FALSE	FALSE	Monitoring Time overrun
12	14.0 out	ActTime	REAL	0.000000e...	0.000000...	Actual Monitoring Time
13	18.0 stat	StartOLD	BOOL	FALSE	FALSE	
14	18.1 stat	StopOLD	BOOL	FALSE	FALSE	
15	18.2 stat	ResetOLD	BOOL	FALSE	FALSE	
16	20.0 stat	mTime	REAL	0.000000e...	0.000000...	

在这种情况下，必须在自动化系统的 S7 程序中创建带有必要参数的数据块。以下是在自动化系统程序中所创建的数据块：

## 5 Error! Style not defined.

### 5.3 Error! Style not defined.

图 5-3



The screenshot shows the SIMATIC Manager interface with a table defining a structure. The table has five columns: Address, Name, Type, Initial val, and Comment. The rows define variables for a motor control system, including start, stop, reset, and monitoring parameters.

Address	Name	Type	Initial val	Comment
0.0		STRUCT		
+0.0	Start	BOOL	FALSE	Switch Motor On
+0.1	Stop	BOOL	FALSE	Switch Motor Off
+0.2	Reset	BOOL	FALSE	Reset Feedback Error
+0.3	FbckMon	BOOL	FALSE	1=Feedback Monitoring on
+2.0	TimeMon	REAL	3.000000e+0	Monitoring time
+6.0	QRun	BOOL	FALSE	Motor command 0=off 1=on
+6.1	QStop	BOOL	TRUE	Motor command 0=on 1=off
+6.2	QState	BOOL	FALSE	Motor state 0=off 1=on
+6.3	QFbckErr	BOOL	FALSE	Monitoring Time overrun
=8.0		END_STRUCT		

为了通过自动化系统实现电机控制，将借助“S7Get”函数块来捕捉 QRun、QStop、QState 和 QFbckErr 等参数，并采用函数块“S7Put”将参数 Start、Stop、Reset、FbckMon 以及 TimeMon 写入到 S7-300 CPU 中。

### 5.3.c 组态通信函数块

本例将通过利用行业库中的“S7Put”和“S7Get”函数块，使 CPU 之间的通信生效。以下步骤描述了如何将这函数块集成以及组态到自动化系统的 S7 程序当中。

表 5-5

序号	操作	注释
1.	<p>打开现有的图表或者新建一个 CFC 图表。</p> <p>将函数块“S7Put”和“S7Get”添加到该图表中。</p> <p>为了将网络中的通信负载降到最低，这些函数块应当利用周期时间较长的循环中断 OB 来进行调用（比如周期 500ms 的 OB33）。</p>	
2.	<p>确定未指定的 S7 连接的本地 ID 以及对对象站点的数据块编号。</p>	<p>本地 ID 会显示在“Properties – S7 connections (属性 – S7 连接)”对话框的“Block parameter (函数块参数)”当中。打开 S7-300 CPU 的项目以确定该数据块。</p>
3.	<p>利用已确定的数值来组态“S7Get”。</p>	<p>本例中已经确定了以下数值：</p> <ul style="list-style-type: none"> <li>• KOM_ID = 2 (连接 ID)</li> <li>• DBNO_SRC = 63 (源数据块的数量)</li> <li>• ADDR_SRC = 16#C (起始于第 12 字节的数据区域)</li> <li>• DBNO_DST = 53 (目标数据块的数量)</li> <li>• ADDR_DST = 16#6 (起始于第 6 字节的数据区域)</li> <li>• LENGTH = 1 (仅传输 1 个字节)</li> </ul>
4.	<p>利用已确定的数值来组态“S7Put”。</p>	<p>本例中已经确定了以下数值：</p> <ul style="list-style-type: none"> <li>• KOM_ID = 2 (连接 ID)</li> <li>• DBNO_SRC = 53 (源数据块的数量)</li> <li>• ADDR_SRC = 16#0 (起始于第 0 字节的数据区域)</li> <li>• DBNO_DST = 63 (目标数据块的数量)</li> <li>• ADDR_DST = 16#0 (起始于第 0 字节的数据区域)</li> <li>• LENGTH = 6 (传输 6 个字节)</li> </ul>

## 5 Error! Style not defined.

### 5.3 Error! Style not defined.

#### 请注意

通信函数块“S7Put”和“S7Get”仅可按字节来执行数据传输，也就是说源数据位和目标数据位总是以至少 8 位为单位进行传输。

在指定数据块参数的过程中，新的数据类型总是会起始于字节边界（字节，布尔或者字符）或者字边界（所有其它数据类型）。如果数据传输不得包含数据块中的所有布尔参数，仅须在两者之间创建一个不同类型的参数即可。

#### 示例：

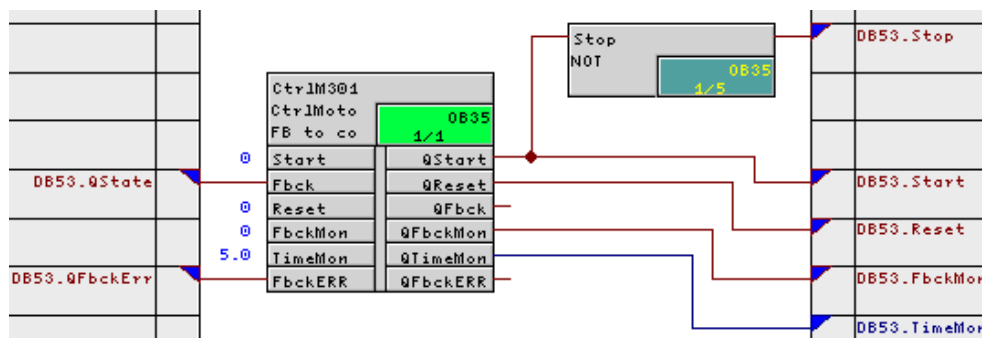
首先，创建两个其实地址分别为 0.0 和 0.1 的布尔参数，然后创建一个起始地址为 1.0 的“字节”类型参数。在地址 2.0 和 2.1 处再创建两个“布尔”类型的参数。那么数据库的总长度便为 4 个字节。

#### 5.3.d 程序逻辑与可视化

可以根据需要对自动化系统控制程序中的数据块参数进行处理。为了按照 PCS 7 标准进行操作，应当利用相应的 OS 面板创建一个适当的函数块。关于该操作过程的信息，请参考 PCS 7 文档中的“函数块编程指令”手册。

本应用中创建了一个简单的函数块，用于在 OS 中创建显示和控制变量，以及在出现错误的情况中发布一条消息（反馈）。该函数块会与用户自定义的数据块参数直接相连。将其纳入到 CFC 图表之后，函数块便如下所示：

图 5-4

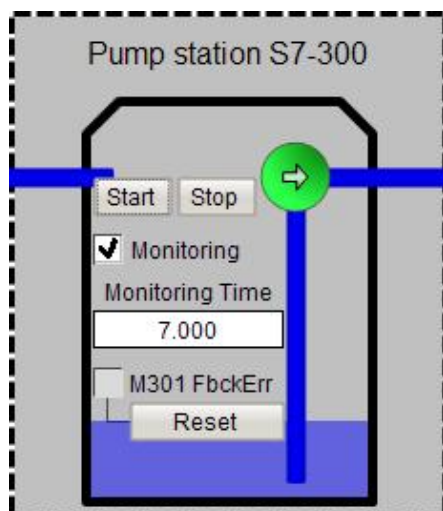


在所提供的 OS 过程映像中，会显示 S7-300 CPU 的电机区域。该区域将如下所示：

## 5 Error! Style not defined.

### 5.3 Error! Style not defined.

图 5-5



作为一种替代方案，还可以在程序逻辑中采用 PCS 7 标准函数块（比如，MOTOR – FB66）但是，该函数块的不足之处在于并不会以输出形式提供所有必须的控制信号（比如用于出错复位的信号）。因此，OS 面板的“RESET（复位）”按钮无法用于对 S7-300 CPU 中的错误进行复位。该功能必须通过其它方式才能实现。

## 5 Error! Style not defined.

### 5.4 Error! Style not defined.

#### 5.4 S7-400H CPU 的组态

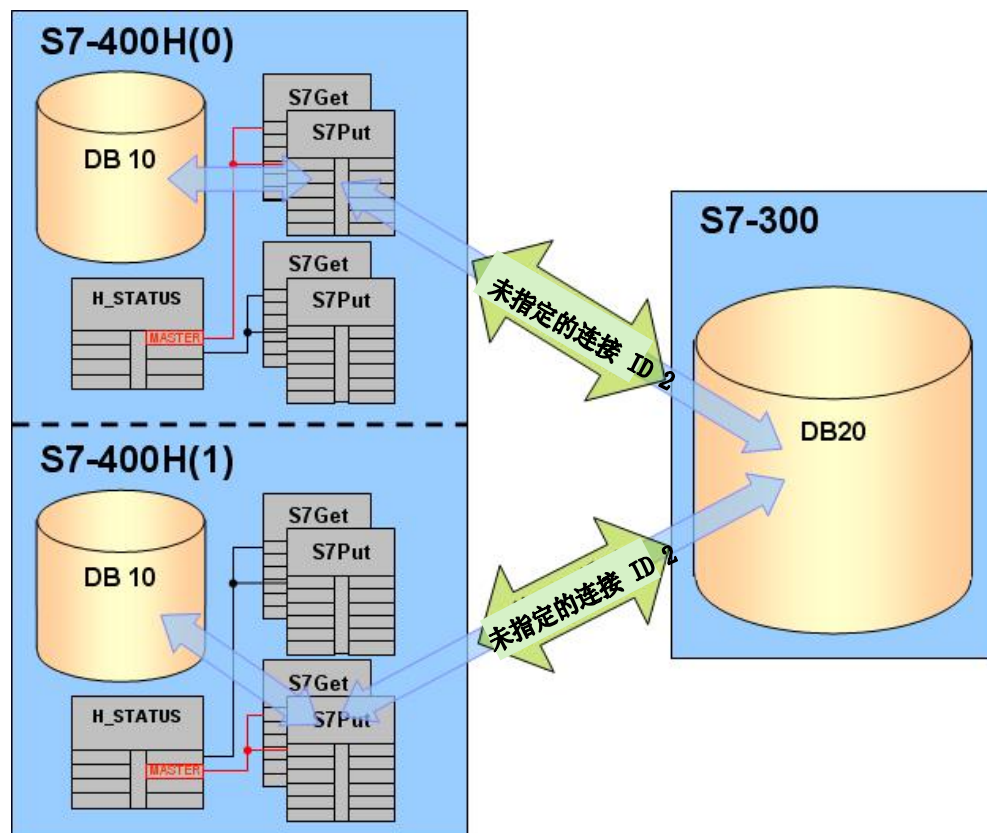
H 系统要求具备两个未指定连接，因为未指定连接在 SIMATIC 管理器中的组态并不提供任何高度可用连接。函数块 S7Put 和 S7Get 均双重集成在了 S7 程序当中。

数据传输会一直由当前定义的主站点来执行。通过“H\_STATUS”的方式可以在用于数据传输的 S7Put 和 S7Get 函数块之间实现切换。该函数块可从以下地址下载得到：

<http://support.automation.siemens.com/WW/view/en/19537149>

其功能原理如以下简图所示。

图 5-6



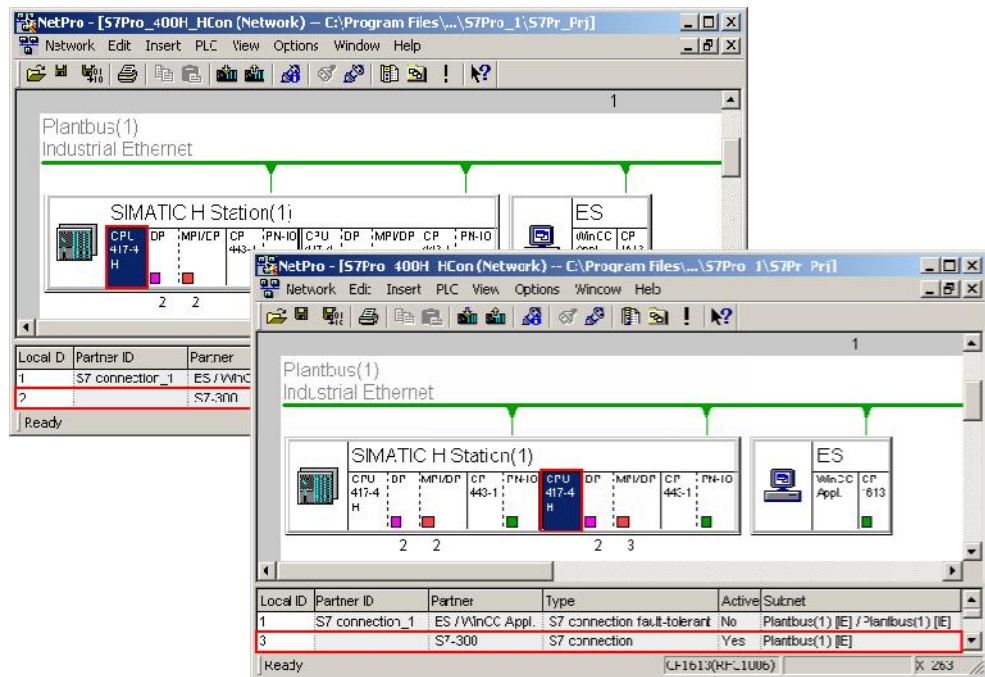
## 5 Error! Style not defined.

### 5.4 Error! Style not defined.

#### 5.4.a 在 S7-400H CPU 中组态未指定的连接

按照章节 5.3.1 “组态未指定的 S7 连接”所述，对这两个 S7-400H CPU 的连接逐一进行组态。这两个连接中的目标站点组态均相同。H-CPU 的每个连接均会分配一个单独的连接 ID。

图 5-7



#### 5.4.b 创建数据块

数据块的组态方式与章节 5.3.2 “创建数据块”中所述的但 CPU 组态步骤相同。

5 Error! Style not defined.

5.4 Error! Style not defined.

5.4.c 组态 S7-400H CPU 中的通信函数块

由于在运行过程中无法更改函数块“S7Put”和“S7Get”连接 ID 中的“KOM\_ID”参数，因此需要为 H-CPU 提供两套这样的函数块。

对于这两个连接，“S7Put”和“S7Get”均按照同样的方式进行组态，仅是“KOM\_ID”的参数值有所不同。

函数块“H\_STATUS”用于使发送和接收块之间的切换生效。该函数块用于输出 H 系统中的运行状态 RUN/STOP 以及 MASTER/RESERVE。

表 5-6

操作	注释
1. 将函数块 H_STATUS、S7Put 以及 S7Get 添加到 CFC 图表中。	<p>The diagram illustrates the configuration of three function blocks: H-State, S7Put, and S7Get. The H-State block (ID 0B35) has two outputs: R0_MSTR and R1_MSTR. The S7Put block (ID 0B35) has an EN input connected to R0_MSTR. The S7Get block (ID 0B35) has an EN input connected to R1_MSTR. The KOM_ID parameter for both S7Put and S7Get is highlighted in red and green, indicating its importance in the configuration.</p>
2. 取消隐藏 S7Put 和 S7Get 函数块中的“EN”输入。	
3. 按照章节 5.3.3 “组态通信函数块”所述对函数块进行组态。	
4. 然后复制函数块，并在参数“KOM_ID”中设定正确的连接 ID。	
5. 对于组态在机架 0 上的 CPU 连接，将状态函数块的输出“R0_MSTR”连接至通信函数块的输入“EN”。	
6. 对于组态在机架 1 上的 CPU 连接，将状态函数块的输出“R1_MSTR”连接至通信函数块的输入“EN”。	

这种组态保证了当前定义为主站的 CPU 会使通信生效。



## 5 Error! Style not defined.

---

### 5.4 Error! Style not defined.

#### 5.4.d 程序逻辑与可视化

余下的控制程序和 OS 组态均与单 CPU 的情况相同。请参考章节 5.3.4 “程序逻辑与可视化”。

## 6 组态带有 PCS 7 行业库的 S7- 300 CPU

### 6.1 核心功能的描述

该整套设备作为 PCS 7 多项目中的一个子项目进行集成。在行业库的协助之下，完成了 S7-300 CPU 的组态，而且过程数据也显示在 OS 以及所连接的操作员面板上。S7-400 CPU 上的实际 PCS 7 程序仍然保持不变。

S7-300 CPU 的程序创建需要 IL S7 库提供帮助。为此，CFC 图表中包含了应用于工艺功能的函数块（比如电机、阀门、测量点.....等等），并且已经相互连接。操作员面板上的可视化可通过将相关的接口函数块集成到 CFC 图表中来实现，该 CFC 图标随后会与相关的函数块相互连接。

此外，还实现了多用户操作的功能。该功能使得操作员可以在 OS 或者 操作员面板上进行操作。

工艺函数块采用了 ALARM\_DQ (SFC 107) 来发布组出错消息。这些可以在 OS 以及操作员面板上显示进行显示和确认。

**请注意**

举例来说，多用户操作功能遵循了层次化的两阶段站点控制理念，并划分为 8 个层次。层次 1 和层次 2 用于在 OS（控制站点）中执行操作，层次 3 至层次 8 用于通过工厂内的操作员面板进行操作。如果有需要的话，也可以对这 8 个操作层次单独进行组态。

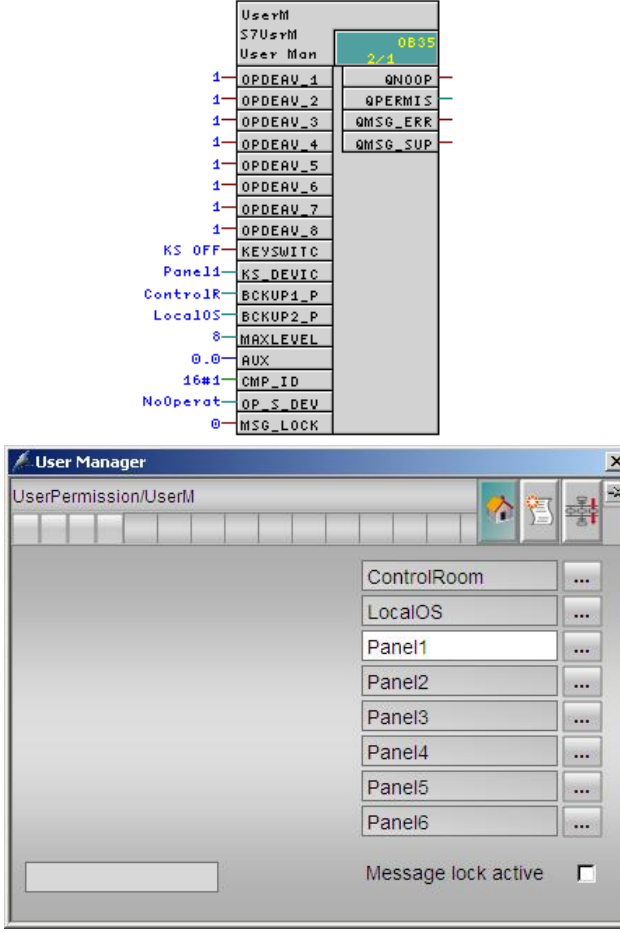
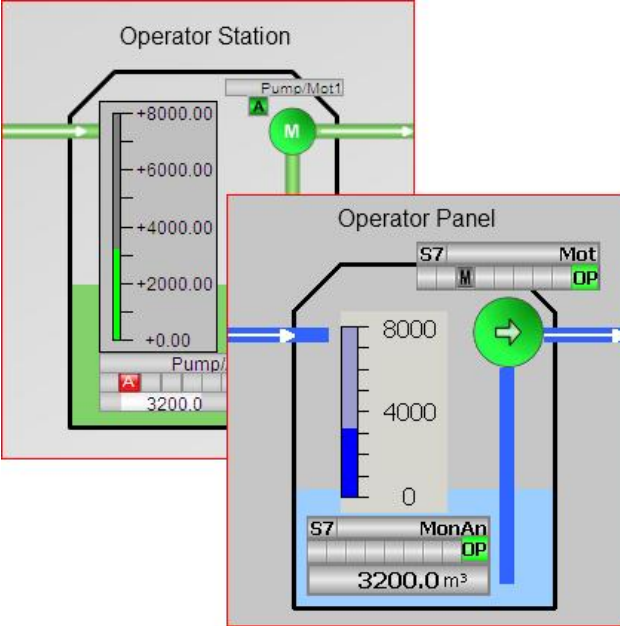
#### 核心功能的原理

表 6-1

序号	操作	显示
1.	整套设备中的执行器和传感器均由 S7-300 CPU 的程序进行控制。该程序的创建借助于 IL S7 库。工艺函数块（比如 S7Mot）被分配到接口函数块（比如 S7PMot）当中用于数据传输，然后这些接口函数块会集成到操作员面板中，并且互相连接。	

## 6 Error! Style not defined.

### 6.1 Error! Style not defined.

序号	操作	显示
2.	<p>操作层次由用户管理器函数块“S7UsrM”进行选择，该函数块与工艺函数块以及接口函数块相连接。然后它会在 OS 中运行，或者由函数块输入“KS_DEVICE”来运行。该函数块无法通过操作员面板进行操作。</p>	 <p>The top part of the screenshot shows a list of user permissions for 'UserM'. The list includes:</p> <ul style="list-style-type: none"> <li>S7UsrM</li> <li>User Man (value: 0635)</li> <li>OPDEAV_1 (value: 2/4)</li> <li>OPDEAV_2 (value: 0N00P)</li> <li>OPDEAV_3 (value: 0PERMIS)</li> <li>OPDEAV_4 (value: 0MSG_ERR)</li> <li>OPDEAV_5 (value: 0MSG_SUP)</li> <li>OPDEAV_6</li> <li>OPDEAV_7</li> <li>OPDEAV_8</li> <li>KS OFF (value: KEYSWITC)</li> <li>Panel1 (value: KS_DEVICE)</li> <li>ControlR (value: BCKUP1_P)</li> <li>LocalOS (value: BCKUP2_P)</li> <li>8 (value: MAXLEVEL)</li> <li>0.0 (value: AUX)</li> <li>16#1 (value: CMP_ID)</li> <li>NoOperat (value: OP_S_DEV)</li> <li>0 (value: MSG_LOCK)</li> </ul> <p>The bottom part of the screenshot shows the 'User Manager' window with the 'UserPermission/UserM' configuration. It lists several panels: ControlRoom, LocalOS, Panel1, Panel2, Panel3, Panel4, Panel5, and Panel6. A 'Message lock active' checkbox is visible at the bottom right.</p>
3.	<p>已提供了 IL S7 库的工艺函数块，且配备了一个适用于 PCS 7 OS 的面板。借助包含在 WinCC flexible 中的面板库，可以对操作员面板上用于可视化的整套设备进行组态。</p>	 <p>The diagram illustrates the configuration of an Operator Station and an Operator Panel for a pump/motor system. The Operator Station shows a tank level indicator with a scale from 0.00 to +8000.00. The Operator Panel shows a similar tank level indicator with a scale from 0 to 8000. The panel also includes a motor status indicator (Mot) and a pump status indicator (Pump/Mot1). The panel is configured to use the S7 library function blocks (S7, MonAn, OP) for visualization.</p>

## 6 Error! Style not defined.

### 6.2 Error! Style not defined.

## 6.2 硬件与软件组件

本应用的创建采用了下述组件：

硬件组件：

表 6-2

组件	数量	MLFB/订单号	备注
CPU 416-3 PN/DP	1	6ES7 416-3ER05-0AB0	
CPU 317-2 DP	1	6ES7-317-2AJ10-0AB0	
多面板 MP 377	1	6AV6-644-0AB01-2AX0	

软件组件：

表 6-3

组件	数量	MLFB/订单号	备注
PCS 7 V8.0 Upd1	1	6ES7-658-1AF08-0YA6	
WinCC flexible 2008 SP3	1	6AV6-613-0AA51-3CA5	
PCS 7 行业库 V8.0	1	6DL5-410-8AA08-0YA0	

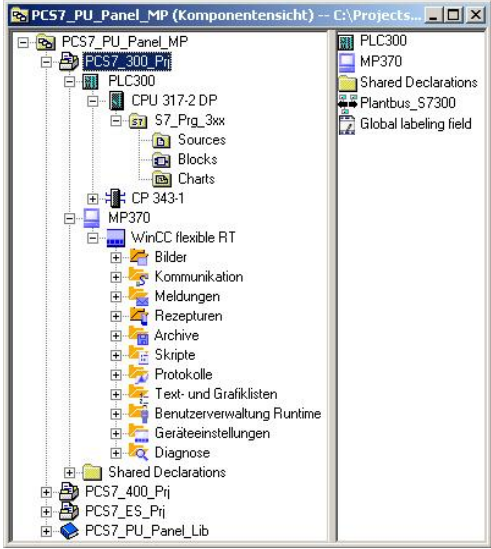
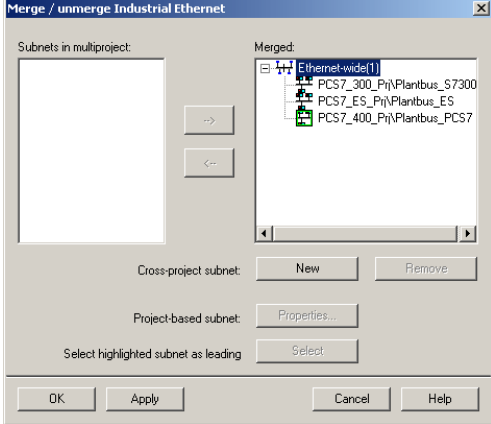
## 6 Error! Style not defined.

### 6.3 Error! Style not defined.

## 6.3 组态与参数赋值

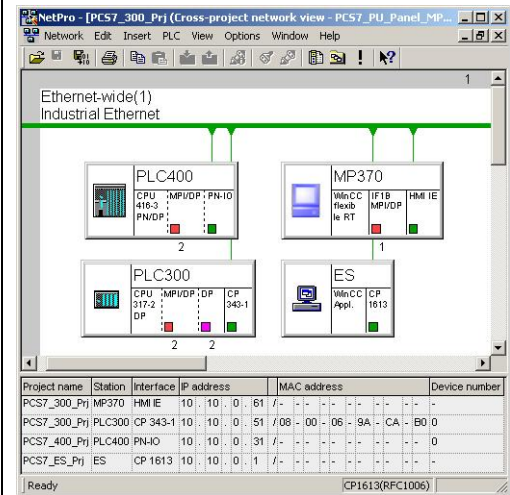
### 6.3.a 为整套设备创建一个新的项目

表 6-4

序号	操作	显示
1.	<p>打开 PCS 7 multi-project。利用菜单命令“File（文件）&gt; Multiproject（多项目）&gt; Add to Multiproject...（添加至多项目.....）”来创建一个包含硬件以及整套设备程序的项目。</p> <p>根据自动化系统的要求，对 SIMATIC 300 以及操作员面板进行组态。</p> <p><b>请注意</b></p> <p>在 multi-project 中指定操作员面板之前，请确保已将 WinCC flexible 安装在 ES 上。作为可选方案，操作员面板还可以组态在一台已安装 WinCC flexible 的单独计算机上。</p>	
2.	<p>要对 multi-project 中的子网进行组合，请打开 NetPro 并选择菜单命令“Edit（编辑）&gt; Merge / Unmerge Subnets（融合/拆分子网）&gt; ...”。</p>	

## 6 Error! Style not defined.

### 6.3 Error! Style not defined.

序号	操作	显示																														
3.	<p>子网结合完毕之后，选择菜单命令“View（查看）&gt; Cross-Project Network View（交叉项目网络视图）”以显示交叉项目网络视图。</p> <p><b>请注意</b> S7-300 CPU 的连接无法通过“PC 内部（本地）”接口来建立。仅当设置了对应的 PC/PC 接口之后，才可以加载该接口。在组态 OS 连接的时候，也应当将此考虑在内。</p>	 <table border="1"> <thead> <tr> <th>Project name</th> <th>Station</th> <th>Interface</th> <th>IP address</th> <th>MAC address</th> <th>Device number</th> </tr> </thead> <tbody> <tr> <td>PCS7_300_Prj</td> <td>MP370</td> <td>HMI IE</td> <td>10.10.0.61</td> <td>-</td> <td>-</td> </tr> <tr> <td>PCS7_300_Prj</td> <td>PLC300</td> <td>CP 343-1</td> <td>10.10.0.51</td> <td>7/08-00-06-9A-CA-B0</td> <td>0</td> </tr> <tr> <td>PCS7_400_Prj</td> <td>PLC400</td> <td>PN-IO</td> <td>10.10.0.31</td> <td>-</td> <td>0</td> </tr> <tr> <td>PCS7_ES_Prj</td> <td>ES</td> <td>CP 1613</td> <td>10.10.0.1</td> <td>-</td> <td>-</td> </tr> </tbody> </table>	Project name	Station	Interface	IP address	MAC address	Device number	PCS7_300_Prj	MP370	HMI IE	10.10.0.61	-	-	PCS7_300_Prj	PLC300	CP 343-1	10.10.0.51	7/08-00-06-9A-CA-B0	0	PCS7_400_Prj	PLC400	PN-IO	10.10.0.31	-	0	PCS7_ES_Prj	ES	CP 1613	10.10.0.1	-	-
Project name	Station	Interface	IP address	MAC address	Device number																											
PCS7_300_Prj	MP370	HMI IE	10.10.0.61	-	-																											
PCS7_300_Prj	PLC300	CP 343-1	10.10.0.51	7/08-00-06-9A-CA-B0	0																											
PCS7_400_Prj	PLC400	PN-IO	10.10.0.31	-	0																											
PCS7_ES_Prj	ES	CP 1613	10.10.0.1	-	-																											

### 6.3.b 组态工艺层次

PCS 7 多项目中各个子项目的工艺层次 (TH) 会默认设置。通常来说，TH 会建立在 OS 的项目中，然后通过利用“Update in Multiproject（更新多项目）”功能将其传输到自动化系统。

本章节将会描述在分开的 OS 区域中整套设备的多用户操作功能。整套设备的实际功能会显示在先前所定义的区域当中。

表 6-5

序号	操作	显示
1.	<p>在 OS 区域的层次中添加一个新的层次文件夹。在该文件夹中创建一个新的 OS 映像。利用“Update in Multiproject（更新多项目）...”功能将工艺层次传输至自动化系统项目。</p>	

## 6 Error! Style not defined.

### 6.3 Error! Style not defined.

#### 6.3.c 创建 S7 程序

以下步骤描述了如何使用 IL S7 库来组态电机 (S7Mot)，使得该电机可以显示在 OS 以及操作员面板上。可以根据需要对外围设备进行互相连接或者对更多信号处理的设定进行组态。

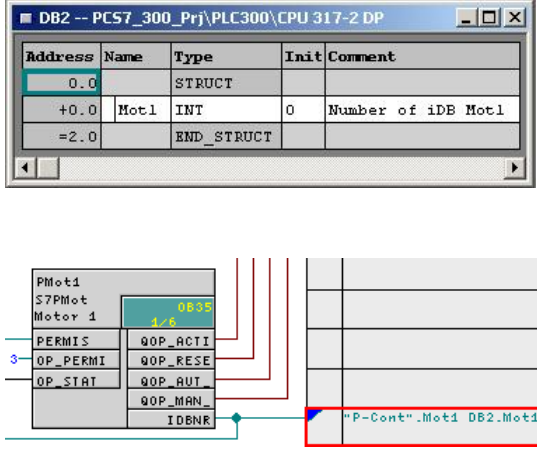
电机函数块可以通过函数块“S7UsrM”进行互相连接，这样可以选择操作层次（操作员面板或者 OS）。

表 6-6

序号	操作	显示
1.	<p>在 S7-300 CPU 程序中创建多用户操作功能。</p> <ul style="list-style-type: none"> <li>打开工艺视图，然后选择 S7-300 CPU 项目。</li> <li>在操作员权限文件夹中创建一个新的 CFC 图表（本例中为“Authority”）。</li> <li>将函数块“S7UsrM”添加至 CFC 图表。该函数块稍后将用于工艺函数块的互相连接。</li> </ul> <p><b>请注意</b> 设置输入“OPDEAV_1”至“OPDEAV_8”，以激活后续可在 OS 中选择的操作层次。此外，操作层次的数量必须在输入“MAXLEVEL”中指定。</p>	
2.	<p>在 S7-300 CPU 程序中创建电机控制功能。</p> <ul style="list-style-type: none"> <li>在 TH 的相关文件夹中创建 CFC 图表。</li> <li>将函数块“S7Mot”和“S7PMot”添加到该图表中。</li> <li>按照右图所示对函数块进行相互连接。</li> <li>将“S7PMot”的输入“OP_PERMIS”设置为操作层次 3。</li> </ul> <p><b>请注意</b> 该函数块的“PERMIS”输入与“S7UsrM”函数块的“QPERMIS”输出相连接。函数块“S7PMot”的输出“IDBNR”与一个附加的数据块参数相连（参见步骤 3）。</p>	

## 6 Error! Style not defined.

### 6.3 Error! Style not defined.

序号	操作	显示
3.	<p>在 S7-300 程序中创建一个数据块。该数据块将作为一个参数容器，包含有所采用的函数块的实例数据块编号。</p> <ul style="list-style-type: none"> <li>切换至 S7-300 程序的“Blocks”文件夹。</li> <li>创建一个新的数据块。确保为其分配一个不会被 CFC 编译器用到的唯一编号。保留区域以及其它应用中的设定可以在“Options (选项) &gt; Settings (设定) &gt; Compile/Download (编译/下载) ..”的 CFC 编辑器中查到。</li> <li>设定参数类型为“INT”。该参数将会连同对应块的数据块编号一同提供。</li> <li>将面板接口块的输出“IDBNR”与数据块的参数相连。</li> </ul> <p><b>请注意</b> 仅需要包含有所用面板函数块 IDB 编号的所有参数的单一数据块即可。</p>	
4.	编译该 S7 程序，并将其下载至控制器。	

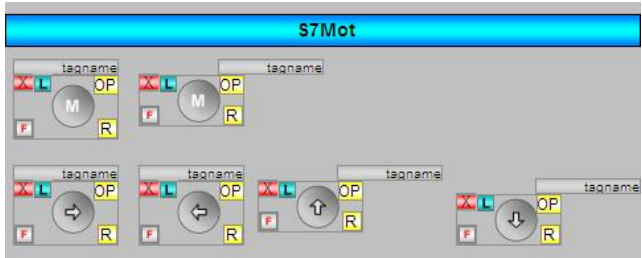
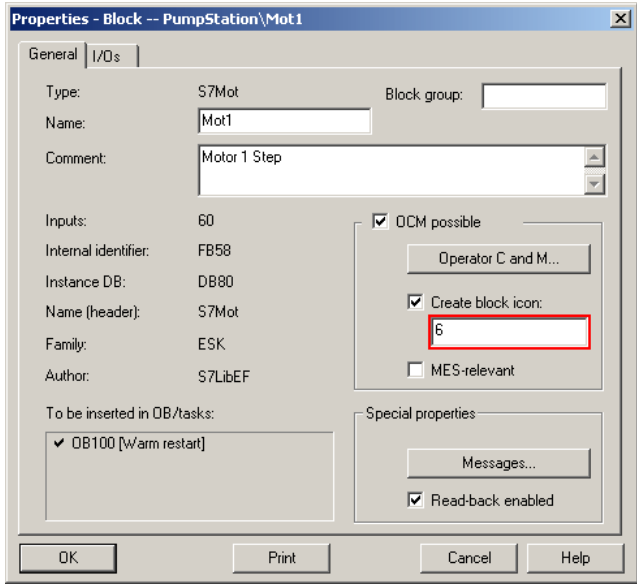


## 6 Error! Style not defined.

### 6.3 Error! Style not defined.

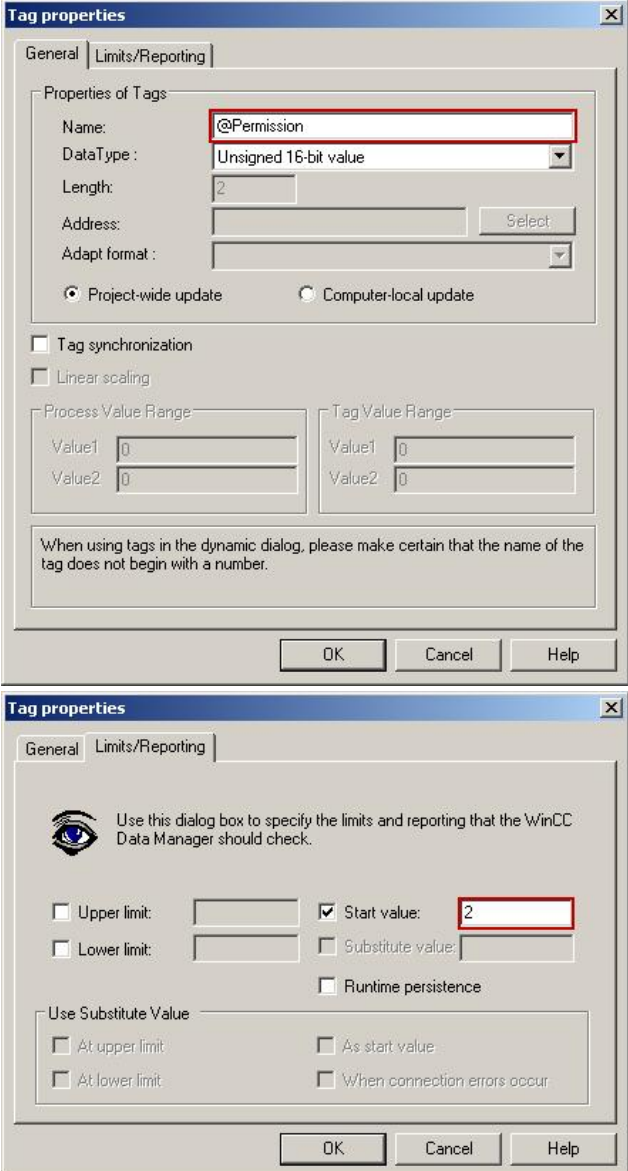
#### 6.3.d 组态 OS

表 6-7

序号	操作	显示
1.	<p>在 OS 映像中，“S7Mot” 函数块的图标 “@S7Mot/1” 将作为标准条目。</p> <p>然而，您还可以从保存在 OS “@PCS7Typicals_IL_S7” 下的一系列函数块图标中选择其它图标。打开相关的 OS 映像以确定所需函数块图标的 ID，然后查看对应的属性。</p> <p>作为一个替代方案，您还可以将鼠标指针移到图标上方，直到出现工具提示。</p>	
2.	<p>如果想要使用其它的块图标，请打开 CFC 图表中相关工艺函数块（比如“S7Mot”）的属性对话框，然后在“Create block icon（创建块图标）”字段中输入所需图标的 ID。如果未进行指定，将采用 “@S7Mot/1” 图标。</p>	
3.	<p>选择“Compile OS（编译 OS）”功能，然后打开 OS 项目。</p>	

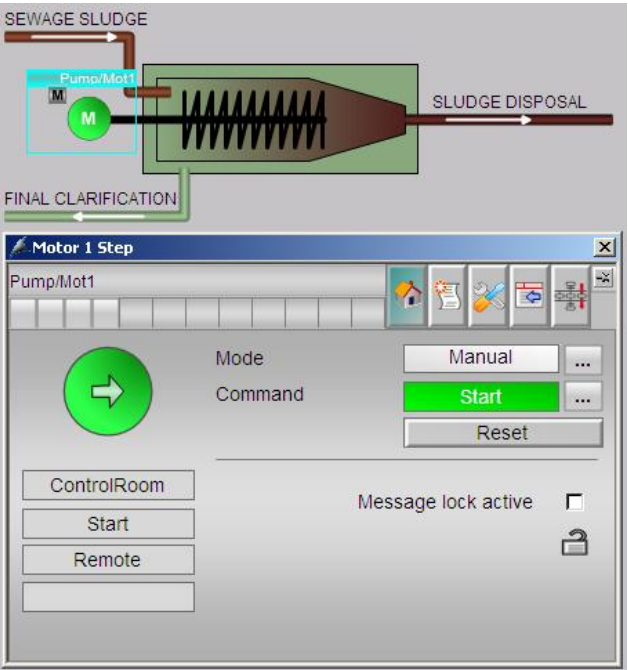
## 6 Error! Style not defined.

### 6.3 Error! Style not defined.

序号	操作	显示
4.	<p>使用多站点控制功能的时候，每个 OS 服务器都必须分配一个特定的操作层次。通过内部标签“@Permission” 可以进行设置。该标签中的层次默认设为起始值。如果所选择的操作层次与“@Permission” 值相符，便可通过该 OS 对 IL 函数块的互相连接进行操作。</p> <ul style="list-style-type: none"> <li>新建一个名为“@Permission”的内部标签，然后将数据类型设为“Unsigned 16-bit value（无符号 16 位数值）”。</li> <li>激活“Start value（起始值）”选项，然后输入需要在此 OS 中使用的操作层次。</li> </ul>	 <p>The image displays two screenshots of the 'Tag properties' dialog box. The top screenshot shows the 'General' tab with the following settings: Name: @Permission, DataType: Unsigned 16-bit value, Length: 2, and Project-wide update selected. The bottom screenshot shows the 'Limits/Reporting' tab with the 'Start value' checkbox checked and the value '2' entered in the text field.</p>

6 Error! Style not defined.

6.3 Error! Style not defined.

序号	操作	显示
5.	<p>打开包含所创建函数块的工厂映像。                      根据要求放置函数块图标并组态工厂影像。                      工厂映像中经过组态的电机如右图                      的例子所示。</p>	 <p>The screenshot displays a control interface for a motor. At the top, a process diagram shows 'SEWAGE SLUDGE' entering a 'Pump/Mot1' block, which is connected to a 'SLUDGE DISPOSAL' line. Below this, a 'Motor 1 Step' control panel is shown. It features a green circular button with a right-pointing arrow, a 'Mode' dropdown menu set to 'Manual', and a prominent green 'Start' button. Other buttons include 'ControlRoom', 'Start', 'Remote', and 'Reset'. A 'Message lock active' checkbox is also visible.</p>

## 6 Error! Style not defined.

### 6.3 Error! Style not defined.

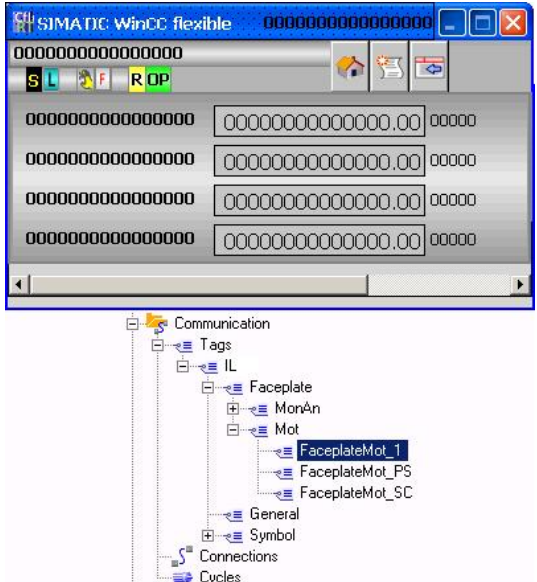
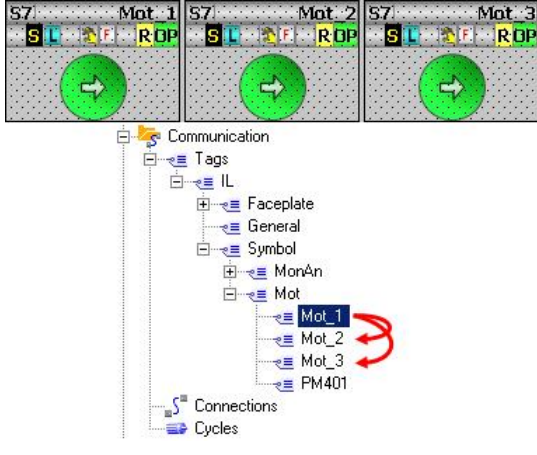
#### 6.3.e 组态操作员面板

WinCC flexible 的行业库中包含了一系列已经预组态的面板。所提供的块图标和面板已经具备所有必须的标签和功能。将其放置在工厂映像中之后，不仅图形对象，就连相关的标签、连接、图形和文本列表也会自动创建。仅须将这些元素适应于指定项目的风格即可。请注意，每个面板以及每个块图标均要求提供单独的标签文件夹。

多个类型相同的工艺函数块的过程数值可以在单一的屏幕画面中显示。点击相应的块图标，即可定义所示面板中的过程数值。

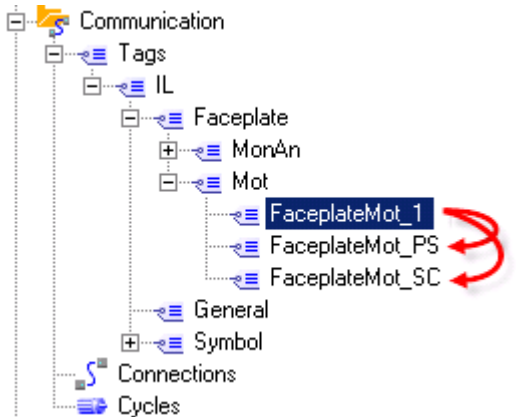
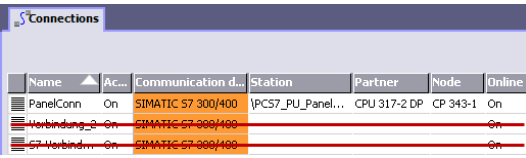
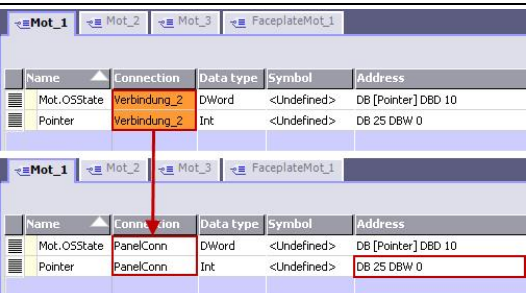
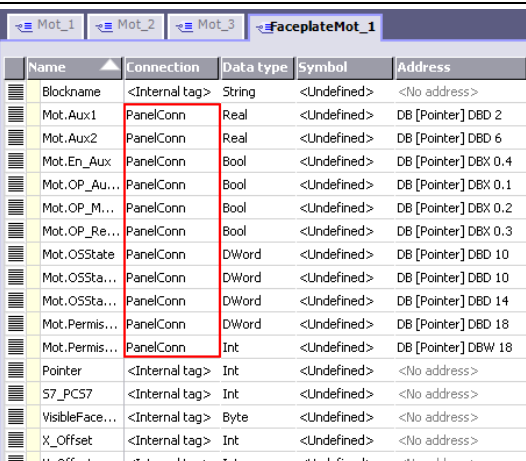
如果想要组态相同类型的多个块图标和面板，请按照下表中的步骤进行操作。

表 6-8

序号	操作	显示
1.	<p>从库中添加所需的面板（比如 MotL）至映像。</p> <p>将会自动生成标签文件夹“IL &gt; Faceplate（面板）&gt; Mot &gt; FaceplateMot_1”。</p> <p><b>请注意</b></p> <p>当重新分配相同的面板时，先前的标签文件夹会被覆盖。而且不会生成更多的标签。</p>	 <p>The screenshot shows the WinCC flexible interface. At the top, there's a title bar 'SIMATIC WinCC flexible' and a status bar '0000000000000000'. Below that, there are four numerical displays, each with a value of '00000000000000.00' and a unit '000000'. The bottom part of the screenshot shows a tree view with the following structure: Communication &gt; Tags &gt; IL &gt; Faceplate &gt; Mot &gt; FaceplateMot_1 (highlighted in blue). Other items in the tree include MonAn, FaceplateMot_PS, FaceplateMot_SC, General, Symbol, Connections, and Cycles.</p>
2.	<p>从库中添加相应的块图标（比如 MotIcon）至映像。</p> <p>将会自动生成标签文件夹“IL &gt; Symbol（符号）&gt; Mot &gt; Mot_1”。</p> <p>重新命名标签文件夹（比如 Motor_1），以避免所创建的标签被下一个块图标所覆盖。</p> <p>所有相同类型的块图标重复执行这一步骤，在同一映像显示中应当标出其过程数值。</p> <p><b>请注意</b></p> <p>块图标所采用的某些标签保存在面板容器当中。每个面板仅需要用到一个标签，当图标添加至标签文件夹时，这些标签会被覆盖掉。</p>	 <p>The screenshot shows three motor icons labeled 'Mot_1', 'Mot_2', and 'Mot_3'. Each icon has a green arrow pointing right. Below the icons, the tree view shows: Communication &gt; Tags &gt; IL &gt; Symbol &gt; MonAn &gt; Mot &gt; Mot_1 (highlighted in blue), Mot_2 (highlighted in blue), Mot_3 (highlighted in blue), and PM401. A red arrow points from the text '应当标出其过程数值' to the 'Mot_1' entry in the tree view.</p>

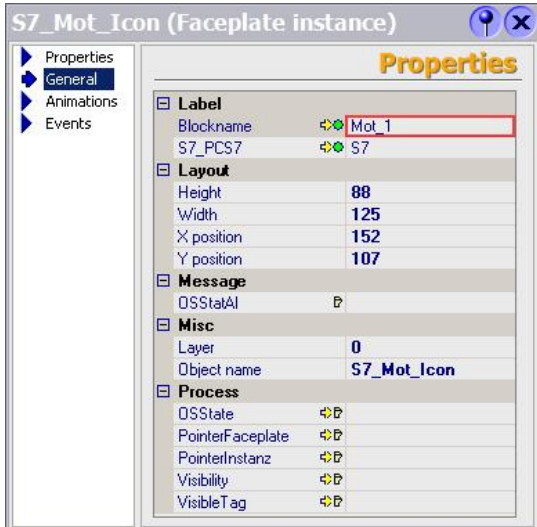
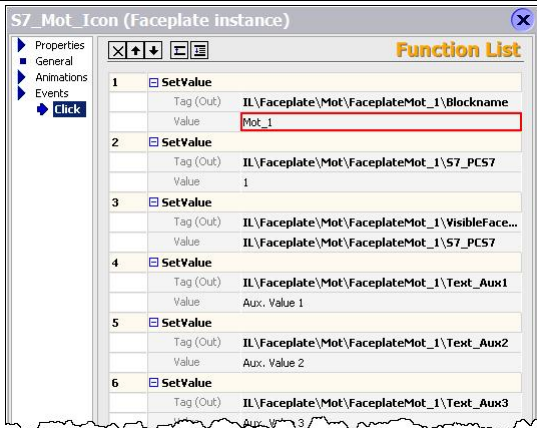
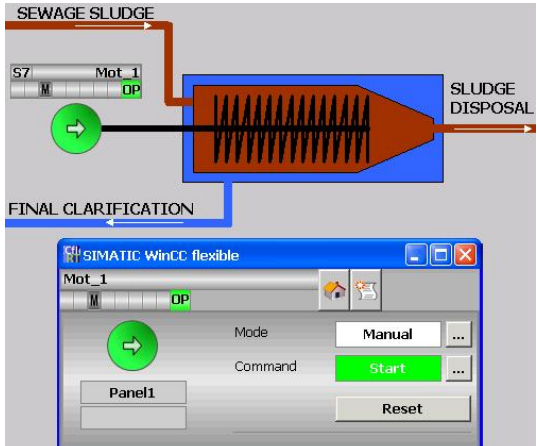
## 6 Error! Style not defined.

### 6.3 Error! Style not defined.

序号	操作	显示																																																																																					
3.	重命名步骤 1 中为面板所创建的标签文件夹。 为每一个添加的面板重复步骤 1 至步骤 3 的操作。																																																																																						
4.	打开项目视图中的 WinCC flexible，然后切换至文件夹“Communication（通信）> Connections（连接）”。 删除全部在添加面板和块图标时所创建的连接，当然前提是不再需要用到这些连接。	 <table border="1"> <thead> <tr> <th>Name</th> <th>Ac...</th> <th>Communication d...</th> <th>Station</th> <th>Partner</th> <th>Node</th> <th>Online</th> </tr> </thead> <tbody> <tr> <td>PanelConn</td> <td>On</td> <td>SIMATIC S7 300/400</td> <td>IPC57_PU_Panel...</td> <td>CPU 317-2 DP</td> <td>CP 343-1</td> <td>On</td> </tr> <tr> <td>Verbindung_2</td> <td>On</td> <td>SIMATIC S7 300/400</td> <td></td> <td></td> <td></td> <td>On</td> </tr> <tr> <td>CP7_Verbind...</td> <td>On</td> <td>SIMATIC S7 300/400</td> <td></td> <td></td> <td></td> <td>On</td> </tr> </tbody> </table>	Name	Ac...	Communication d...	Station	Partner	Node	Online	PanelConn	On	SIMATIC S7 300/400	IPC57_PU_Panel...	CPU 317-2 DP	CP 343-1	On	Verbindung_2	On	SIMATIC S7 300/400				On	CP7_Verbind...	On	SIMATIC S7 300/400				On																																																									
Name	Ac...	Communication d...	Station	Partner	Node	Online																																																																																	
PanelConn	On	SIMATIC S7 300/400	IPC57_PU_Panel...	CPU 317-2 DP	CP 343-1	On																																																																																	
Verbindung_2	On	SIMATIC S7 300/400				On																																																																																	
CP7_Verbind...	On	SIMATIC S7 300/400				On																																																																																	
5.	逐一打开所有新近创建的块图标标签文件夹。 <ul style="list-style-type: none"> <li>更改所有无效的连接。使用选择列表来指定与 CPU 相连的正确连接。</li> <li>寻址“Pointer”标签，使其与新近创建的数据块参数相符。</li> </ul>	 <table border="1"> <thead> <tr> <th>Name</th> <th>Connection</th> <th>Data type</th> <th>Symbol</th> <th>Address</th> </tr> </thead> <tbody> <tr> <td>Mot.OSState</td> <td>Verbindung_2</td> <td>DWord</td> <td>&lt;Undefined&gt;</td> <td>DB [Pointer] DBD 10</td> </tr> <tr> <td>Pointer</td> <td>Verbindung_2</td> <td>Int</td> <td>&lt;Undefined&gt;</td> <td>DB 25 DBW 0</td> </tr> </tbody> </table> <table border="1"> <thead> <tr> <th>Name</th> <th>Connection</th> <th>Data type</th> <th>Symbol</th> <th>Address</th> </tr> </thead> <tbody> <tr> <td>Mot.OSState</td> <td>PanelConn</td> <td>DWord</td> <td>&lt;Undefined&gt;</td> <td>DB [Pointer] DBD 10</td> </tr> <tr> <td>Pointer</td> <td>PanelConn</td> <td>Int</td> <td>&lt;Undefined&gt;</td> <td>DB 25 DBW 0</td> </tr> </tbody> </table>	Name	Connection	Data type	Symbol	Address	Mot.OSState	Verbindung_2	DWord	<Undefined>	DB [Pointer] DBD 10	Pointer	Verbindung_2	Int	<Undefined>	DB 25 DBW 0	Name	Connection	Data type	Symbol	Address	Mot.OSState	PanelConn	DWord	<Undefined>	DB [Pointer] DBD 10	Pointer	PanelConn	Int	<Undefined>	DB 25 DBW 0																																																							
Name	Connection	Data type	Symbol	Address																																																																																			
Mot.OSState	Verbindung_2	DWord	<Undefined>	DB [Pointer] DBD 10																																																																																			
Pointer	Verbindung_2	Int	<Undefined>	DB 25 DBW 0																																																																																			
Name	Connection	Data type	Symbol	Address																																																																																			
Mot.OSState	PanelConn	DWord	<Undefined>	DB [Pointer] DBD 10																																																																																			
Pointer	PanelConn	Int	<Undefined>	DB 25 DBW 0																																																																																			
6.	逐一打开所有新近创建的面板标签文件夹。 <ul style="list-style-type: none"> <li>更改所有无效的连接。使用选择列表来指定与 CPU 相连的正确连接。</li> </ul> <p><b>请注意</b> 面板标签无须进行寻址。过程标签可通过“Pointer”标签进行寻址。点击块图标之后，内部标签值即可用。</p>	 <table border="1"> <thead> <tr> <th>Name</th> <th>Connection</th> <th>Data type</th> <th>Symbol</th> <th>Address</th> </tr> </thead> <tbody> <tr> <td>Blockname</td> <td>&lt;Internal tag&gt;</td> <td>String</td> <td>&lt;Undefined&gt;</td> <td>&lt;No address&gt;</td> </tr> <tr> <td>Mot_Aux1</td> <td>PanelConn</td> <td>Real</td> <td>&lt;Undefined&gt;</td> <td>DB [Pointer] DBD 2</td> </tr> <tr> <td>Mot_Aux2</td> <td>PanelConn</td> <td>Real</td> <td>&lt;Undefined&gt;</td> <td>DB [Pointer] DBD 6</td> </tr> <tr> <td>Mot.En_Aux</td> <td>PanelConn</td> <td>Bool</td> <td>&lt;Undefined&gt;</td> <td>DB [Pointer] DBX 0.4</td> </tr> <tr> <td>Mot.OP_Au...</td> <td>PanelConn</td> <td>Bool</td> <td>&lt;Undefined&gt;</td> <td>DB [Pointer] DBX 0.1</td> </tr> <tr> <td>Mot.OP_M...</td> <td>PanelConn</td> <td>Bool</td> <td>&lt;Undefined&gt;</td> <td>DB [Pointer] DBX 0.2</td> </tr> <tr> <td>Mot.OP_Re...</td> <td>PanelConn</td> <td>Bool</td> <td>&lt;Undefined&gt;</td> <td>DB [Pointer] DBX 0.3</td> </tr> <tr> <td>Mot.OSSta...</td> <td>PanelConn</td> <td>DWord</td> <td>&lt;Undefined&gt;</td> <td>DB [Pointer] DBD 10</td> </tr> <tr> <td>Mot.OSSta...</td> <td>PanelConn</td> <td>DWord</td> <td>&lt;Undefined&gt;</td> <td>DB [Pointer] DBD 14</td> </tr> <tr> <td>Mot.Permis...</td> <td>PanelConn</td> <td>DWord</td> <td>&lt;Undefined&gt;</td> <td>DB [Pointer] DBD 18</td> </tr> <tr> <td>Mot.Permis...</td> <td>PanelConn</td> <td>Int</td> <td>&lt;Undefined&gt;</td> <td>DB [Pointer] DBW 18</td> </tr> <tr> <td>Pointer</td> <td>&lt;Internal tag&gt;</td> <td>Int</td> <td>&lt;Undefined&gt;</td> <td>&lt;No address&gt;</td> </tr> <tr> <td>S7_PCS7</td> <td>&lt;Internal tag&gt;</td> <td>Int</td> <td>&lt;Undefined&gt;</td> <td>&lt;No address&gt;</td> </tr> <tr> <td>VisibleFace...</td> <td>&lt;Internal tag&gt;</td> <td>Byte</td> <td>&lt;Undefined&gt;</td> <td>&lt;No address&gt;</td> </tr> <tr> <td>X_Offset</td> <td>&lt;Internal tag&gt;</td> <td>Int</td> <td>&lt;Undefined&gt;</td> <td>&lt;No address&gt;</td> </tr> <tr> <td>Y_Offset</td> <td>&lt;Internal tag&gt;</td> <td>Int</td> <td>&lt;Undefined&gt;</td> <td>&lt;No address&gt;</td> </tr> </tbody> </table>	Name	Connection	Data type	Symbol	Address	Blockname	<Internal tag>	String	<Undefined>	<No address>	Mot_Aux1	PanelConn	Real	<Undefined>	DB [Pointer] DBD 2	Mot_Aux2	PanelConn	Real	<Undefined>	DB [Pointer] DBD 6	Mot.En_Aux	PanelConn	Bool	<Undefined>	DB [Pointer] DBX 0.4	Mot.OP_Au...	PanelConn	Bool	<Undefined>	DB [Pointer] DBX 0.1	Mot.OP_M...	PanelConn	Bool	<Undefined>	DB [Pointer] DBX 0.2	Mot.OP_Re...	PanelConn	Bool	<Undefined>	DB [Pointer] DBX 0.3	Mot.OSSta...	PanelConn	DWord	<Undefined>	DB [Pointer] DBD 10	Mot.OSSta...	PanelConn	DWord	<Undefined>	DB [Pointer] DBD 14	Mot.Permis...	PanelConn	DWord	<Undefined>	DB [Pointer] DBD 18	Mot.Permis...	PanelConn	Int	<Undefined>	DB [Pointer] DBW 18	Pointer	<Internal tag>	Int	<Undefined>	<No address>	S7_PCS7	<Internal tag>	Int	<Undefined>	<No address>	VisibleFace...	<Internal tag>	Byte	<Undefined>	<No address>	X_Offset	<Internal tag>	Int	<Undefined>	<No address>	Y_Offset	<Internal tag>	Int	<Undefined>	<No address>
Name	Connection	Data type	Symbol	Address																																																																																			
Blockname	<Internal tag>	String	<Undefined>	<No address>																																																																																			
Mot_Aux1	PanelConn	Real	<Undefined>	DB [Pointer] DBD 2																																																																																			
Mot_Aux2	PanelConn	Real	<Undefined>	DB [Pointer] DBD 6																																																																																			
Mot.En_Aux	PanelConn	Bool	<Undefined>	DB [Pointer] DBX 0.4																																																																																			
Mot.OP_Au...	PanelConn	Bool	<Undefined>	DB [Pointer] DBX 0.1																																																																																			
Mot.OP_M...	PanelConn	Bool	<Undefined>	DB [Pointer] DBX 0.2																																																																																			
Mot.OP_Re...	PanelConn	Bool	<Undefined>	DB [Pointer] DBX 0.3																																																																																			
Mot.OSSta...	PanelConn	DWord	<Undefined>	DB [Pointer] DBD 10																																																																																			
Mot.OSSta...	PanelConn	DWord	<Undefined>	DB [Pointer] DBD 14																																																																																			
Mot.Permis...	PanelConn	DWord	<Undefined>	DB [Pointer] DBD 18																																																																																			
Mot.Permis...	PanelConn	Int	<Undefined>	DB [Pointer] DBW 18																																																																																			
Pointer	<Internal tag>	Int	<Undefined>	<No address>																																																																																			
S7_PCS7	<Internal tag>	Int	<Undefined>	<No address>																																																																																			
VisibleFace...	<Internal tag>	Byte	<Undefined>	<No address>																																																																																			
X_Offset	<Internal tag>	Int	<Undefined>	<No address>																																																																																			
Y_Offset	<Internal tag>	Int	<Undefined>	<No address>																																																																																			

## 6 Error! Style not defined.

### 6.3 Error! Style not defined.

序号	操作	显示
7.	<p>为了确保块图标显示清晰且唯一的名称，可在图标属性对话框中进行修改。</p> <ul style="list-style-type: none"> <li>在过程映像中选择需要编辑的图标。</li> <li>打开属性对话框然后选择 "General（常规）" 文件夹。</li> <li>在 "Block name（块名称）" 字段中可以定义特征名称（比如过程标签的名称）。</li> <li>"S7_PCS7" 字段中显示 "S7" 即表示 S7MotL 块已连接。</li> </ul>	
8.	<p>在图标的属性对话框中可以设置是否在面板中显示块图标的名称。</p> <ul style="list-style-type: none"> <li>在过程映像中标记需要编辑的图标。</li> <li>打开属性对话框然后选择 "Events（事件） &gt; Click（点击）" 文件夹。</li> <li>在面板 "block name（块名称）" 标签的事件 "SetValue（设置值）" 行中输入与上述相同的名称。</li> </ul> <p><b>请注意</b></p> <p>如果您已经按照上述的顺序执行组态，这些事件将已包含正确的面板标签。如果想要使用另一个面板，那么需要设置的标签也应在此处进行更改。</p>	
9.	<p>根据需要在过程视图中放置块图标和面板。根据您的要求组态工厂映像。</p> <p>右图示出了经过组态的电机示例。</p>	

## 7 链接与文献

### Internet 链接

本列表所列出的文献内容并不完备，仅提供部分相关信息以供参考。

表 7-1

	主题	标题
\1\	本文档的参考	<a href="http://support.automation.siemens.com/WW/view/en/50708061">http://support.automation.siemens.com/WW/view/en/50708061</a>
\2\	西门子工业在线支持	<a href="http://support.automation.siemens.com">http://support.automation.siemens.com</a>
\3\	销售/交付发布 SIMATIC PCS 7 行业 库 V8.0	<a href="http://support.automation.siemens.com/WW/view/en/60982306">http://support.automation.siemens.com/WW/view/en/60982306</a>
\4\	利用 SIMATIC PCS 7 / OPEN OS 集成第三方 系统	<a href="http://support.automation.siemens.com/WW/view/en/49740087">http://support.automation.siemens.com/WW/view/en/49740087</a>
\5\	“PCS 7 OS 过程控制” 手册	<a href="http://support.automation.siemens.com/WW/view/en/36195920">http://support.automation.siemens.com/WW/view/en/36195920</a>
\6\	如何读取 H 系统的状 态	<a href="http://support.automation.siemens.com/WW/view/en/19537149">http://support.automation.siemens.com/WW/view/en/19537149</a>

## 8 版本历史

表 8-1

版本	日期	修订
V1.0	12.08.2011	出版（内部网络）
V2.0	22.06.2012	更改至第二版（内部网络）