

操作指南 • 08 月 2015 年

S7-1200 组态控制

<https://support.industry.siemens.com/cs/cn/zh/view/109479370>

Unrestricted

目录

1	组态控制的优势和应用	3
2	组态控制功能组态	6
2.1	组态控制功能启用.....	6
2.2	创建新数据类型.....	6
2.3	创建控制数据块.....	7
2.4	编程传送控制数据记录.....	8
2.5	控制数据记录.....	9
3	组态控制示例	11
3.1	示例 1——已组态但未使用模块.....	11
3.2	示例 2——模块位置颠倒.....	12

1 组态控制的优势和应用

组态控制功能可以用来创建一个要在多种不同安装中使用的自动化解方案（机器），各种硬件组态可以保存在 CPU 中，通过用户程序改变硬件组态与实际安装对应。

下面通过一个实际案例说明一下组态控制的优势和应用。

比如一个蛋糕生产线的 OEM 设备提供商，有些工厂需要烘焙的生产线，那么该设备提供商需要组态一个烘焙项目，需要 S7-1200 CPU 和 I/O 模块 A 来负责蛋糕加热的工艺。如图 1-1 所示。

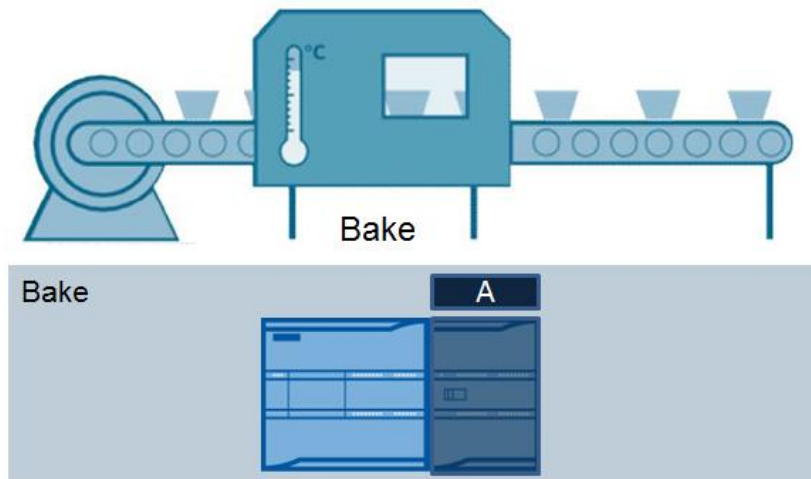


图 1-1 烘焙生产线

同时，有的工厂在烘焙的基础上，还需要添加包装工艺，如果没有组态控制功能，那么该设备提供商需要建立另外一个烘焙与包装项目，在原先 S7-1200 CPU 和 I/O 模块 A 的基础上添加 I/O 模块 B 来负责包装工艺，如图 1-2。

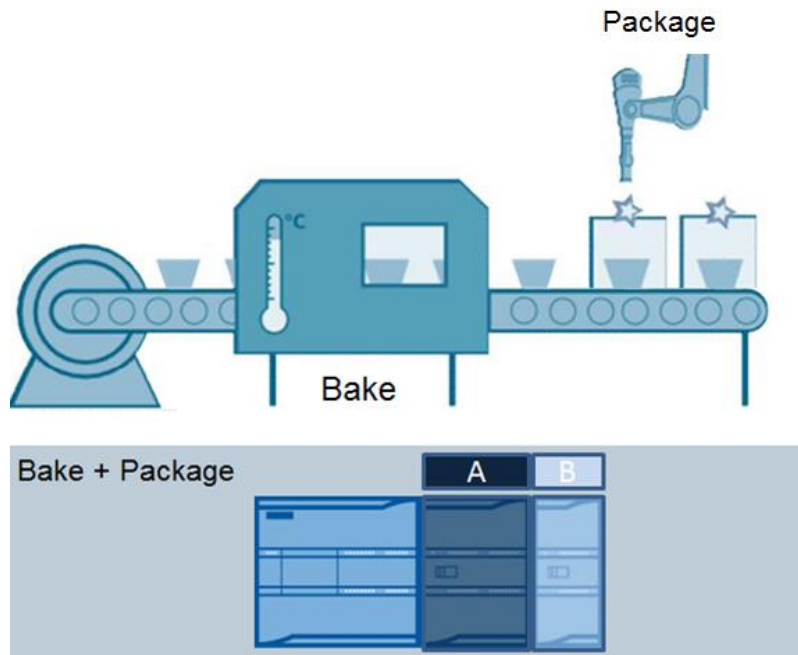


图 1-2 烘焙与包装生产线

另外，这个设备提供商的有些客户还需要蛋糕装饰工艺，在原先烘焙和包装基础上添加例如裱花或打奶油工艺，同样如果没有组态控制功能，该设备提供商还需要针对此类客户创建一个新项目，再增加一个 I/O 模块 C 来负责装饰工艺环节，如图 1-3。

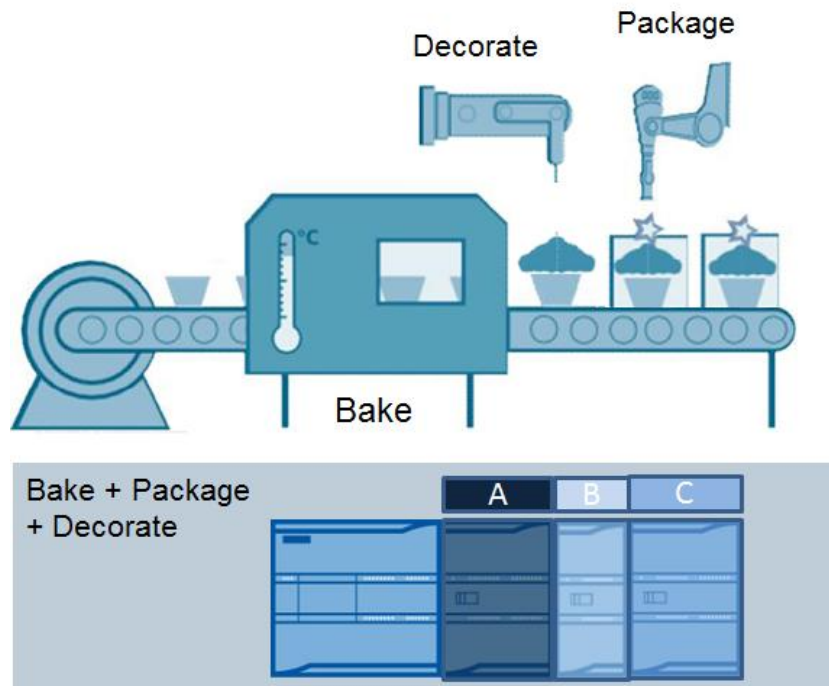


图 1-3 烘焙、包装与装饰生产线

由此可见，在没有组态控制之前，设备提供商需要针对不同客户的不同需求配置多个项目，但是其实这些项目都是基于同一个烘焙设备上添加了不同的组件。利用现在的组态控制功能就可以组态一个最全的配置，然后根据不同需要，通过用户程序灵活调整组态。

2 组态控制功能组态

S7-1200 CPU 从固件版本 V4.1 起开始支持组态控制功能，其组态过程如下所述。

2.1 组态控制功能启用

使用组态控制功能之前，需要在 CPU 设备组态的属性里选择“允许通过用户程序重新组态设备”复选框来启动组态控制功能。如图 2-1。

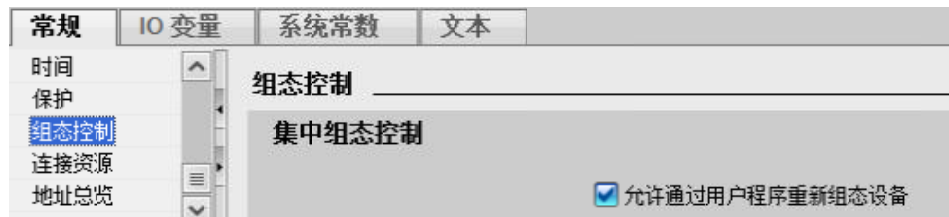


图 2-1 启动组态控制功能

2.2 创建新数据类型

在博途的 PLC 数据类型中添加新的数据类型，创建一个用于包含控制数据记录的 PLC 数据类型，其中前 4 个 USInt 变量用于存储组态控制信息，后面 12 个 USInt 变量对应于 S7-1200 最大组态时的各个插槽的实际状态，具体如 2-2 所示：

ControlDataRecord				
	名称	数据类型	启动值	注释
1	Static			
2	ConfigControl	Struct		
3	Block_length	USInt	16	Length of control data record, including header
4	Block_ID	USInt	196	Data record number
5	Version	USInt	5	
6	Subversion	USInt	0	
7	Slot_1	USInt	255	Assignment for CPU annex card/ Actual annex card
8	Slot_2	USInt	255	Configured slot 2 / Assigned "real" slot
9	Slot_3	USInt	255	Configured slot 3 / Assigned "real" slot
10	Slot_4	USInt	255	Configured slot 4 / Assigned "real" slot
11	Slot_5	USInt	255	Configured slot 5 / Assigned "real" slot
12	Slot_6	USInt	255	Configured slot 6 / Assigned "real" slot
13	Slot_7	USInt	255	Configured slot 7 / Assigned "real" slot
14	Slot_8	USInt	255	Configured slot 8 / Assigned "real" slot
15	Slot_9	USInt	255	Configured slot 9 / Assigned "real" slot
16	Slot_101	USInt	255	Configured slot 101 / Assigned "real" slot
17	Slot_102	USInt	255	Configured slot 102 / Assigned "real" slot
18	Slot_103	USInt	255	Configured slot 103 / Assigned "real" slot

图 2-2 新数据类型

2.3 创建控制数据块

用刚刚创建的数据类型创建一个包含控制数据的数据块，如图 2-3。

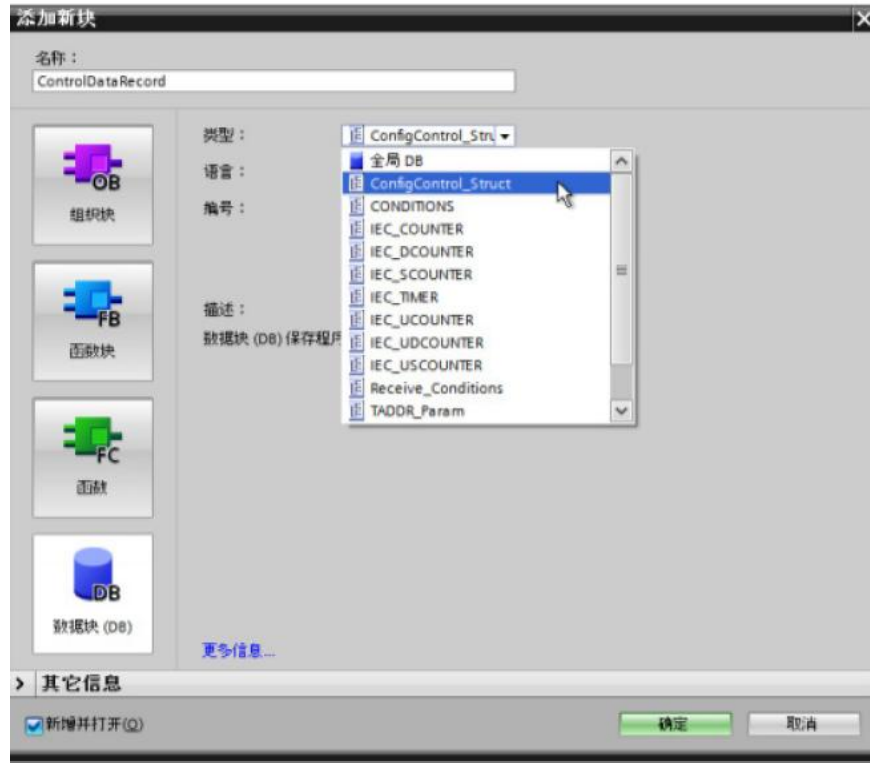


图 2-3 添加新的数据块

在新生成的数据块中，按图 2-4 所示组态 Block_length、Block_ID、版本以及次版本。

根据是否存在插槽以及其在实际安装中的位置组态插槽的值：

- 0：实际组态中不存在已组态的模块。（插槽为空。）
- 1 到 9，101 到 103：已组态插槽的实际插槽位置
- 255：STEP 7 设备组态在此插槽中不包含模块。

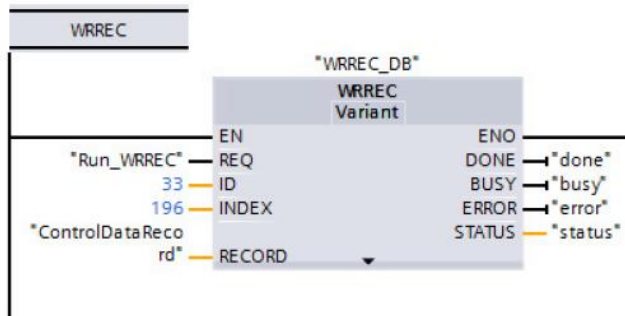
ControlDataRecord				
	名称	数据类型	启动值	注释
1	Static			
2	ConfigControl	Struct		
3	Block_length	USInt	16	Length of control data record, including header
4	Block_ID	USInt	196	Data record number
5	Version	USInt	5	
6	Subversion	USInt	0	
7	Slot_1	USInt	255	Assignment for CPU annex card/ Actual annex card
8	Slot_2	USInt	255	Configured slot 2 / Assigned "real" slot
9	Slot_3	USInt	255	Configured slot 3 / Assigned "real" slot
10	Slot_4	USInt	255	Configured slot 4 / Assigned "real" slot
11	Slot_5	USInt	255	Configured slot 5 / Assigned "real" slot
12	Slot_6	USInt	255	Configured slot 6 / Assigned "real" slot
13	Slot_7	USInt	255	Configured slot 7 / Assigned "real" slot
14	Slot_8	USInt	255	Configured slot 8 / Assigned "real" slot
15	Slot_9	USInt	255	Configured slot 9 / Assigned "real" slot
16	Slot_101	USInt	255	Configured slot 101 / Assigned "real" slot
17	Slot_102	USInt	255	Configured slot 102 / Assigned "real" slot
18	Slot_103	USInt	255	Configured slot 103 / Assigned "real" slot

图 2-4 数据块组态

2.4 编程传送控制数据记录

在启动 OB 中，调用扩展的 WRREC（写入数据记录）指令，将创建的控制数据记录传送到硬件 ID 33 的索引 196。使用标签和 JMP（跳转）指令等待 WRREC 指令完成。

程序段 1:



程序段 2:

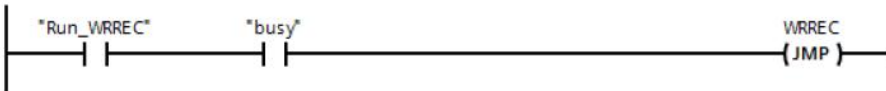


图 2-5 启动 OB 中程序

注意，WRREC 指令在启动 OB 中传送完控制数据记录后组态控制才会生效。如果已启用组态控制但 CPU 不具有控制数据记录，则在退出 STARTUP 模式时会

转到 STOP 模式。所以需要确保启动 OB 中包含传输控制数据记录的程序。

2.5 控制数据记录

图 2-6 中列出了插槽号分配情况：

插槽	模块
1	信号板或通信板（CPU 附件卡）
2 到 9	信号模块
101 到 103	通信模块

图 2-6 插槽号分配

控制数据记录 196 包含插槽分配并表示实际组态，如下所示：

Byte	元素	值	说明	
0	块长度	16	Header	
1	块 ID	196		
2	版本	5		
3	次版本	0		
4	CPU 附件卡的分配	实际附件卡, 0 或 255*	控制元素	
5	组态的插槽 2 的分配	实际插槽, 0 或 255*		说明已将设备中的哪个实际插槽分配给每个单元中组态的插槽。
...		
12	组态的插槽 9 的分配	实际插槽, 0 或 255*	与信号模块不同, 实际存在的通信模块的实际插槽必须与已组态的插槽相同。	
13	组态的插槽 101 的分配	实际插槽或 255*		
14	组态的插槽 102 的分配	实际插槽或 255*		
15	组态的插槽 103 的分配	实际插槽或 255*		

***插槽值：**

0: 实际组态中不存在已组态的模块。（插槽为空。）

1 到 9, 101 到 103: 已组态插槽的实际插槽位置

255: STEP 7 设备组态在此插槽中不包含模块。

图 2-7 数据记录 196 结构

修改控制数据记录时请遵守以下准则：

- 组态控制不支持通信模块的位置更改。插槽 101 到 103 的控制数据记录插槽位置必须与实际安装对应。如果没有为插槽配置模块，请为该插槽位置输入 255。

- 在已填充（已使用）的插槽之间不能有嵌入式空（未使用）插槽。

例如，如果实际组态在插槽 4 中有一个模块，则实际组态在插槽 2 和 3 中也必须有模块。相应地，如果实际组态在插槽 102 中有一个通信模块，则实际组态在插槽 101 中也必须有一个模块。

-
- 如果已启用组态控制，却没有控制数据记录，则 CPU 仍未做好运行准备。如果启动 OB 未传送一个有效的控制数据记录，则 CPU 从启动模式返回到 STOP 模式。CPU 在这种情况下不会初始化集中式 I/O，并将在诊断缓冲区中输入转到 STOP 模式的原因。
 - CPU 将成功传送的控制数据记录保存在保持性存储器中，也就是说，在不更改组态的情况下重启时无需重新写入控制数据记录 196。
 - 每个实际插槽只能在控制数据记录中出现一次。
 - 只能将一个实际插槽分配给一个已组态插槽。

3 组态控制示例

本示例介绍了由一个 CPU 和三个 I/O 模块组成的配置。在第一次实际安装中，插槽 3 处的模块并不存在，因此可使用组态控制将其“隐藏”。

第二次安装时，应用将包括最初隐藏的模块，但现在该模块位于最后一个插槽中。

修改后的控制数据记录可提供有关模块插槽分配的信息。

3.1 示例 1——已组态但未使用模块

设备组态包含实际安装中可能存在的所有模块（最大组态）。如图 3-1 所示。

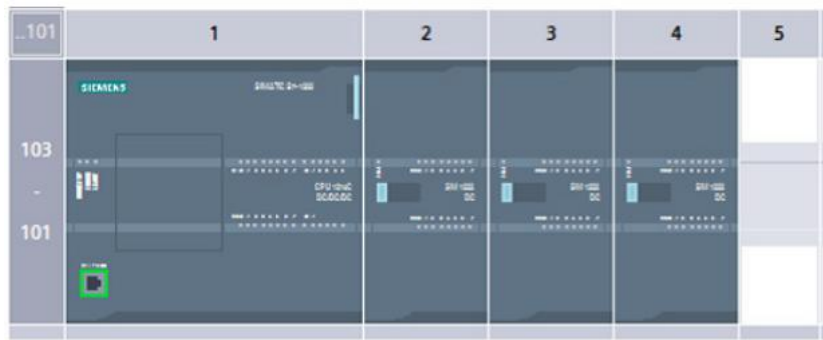


图 3-1 最大安装的设备组态

实际情况下，在设备组态中应位于插槽 3 中的模块不存在，而插槽 4 组态的模块实际位于插槽 3 中。如图 3-2 所示。



图 3-2 实际设备安装

要指示实际安装模块的不存在，必须在控制数据记录中使用 0 组态插槽 3，即 $\text{Slot}_3 = 0$ ，而插槽 4 组态的模块实际位于插槽 3 中，所以 $\text{Slot}_4 = 3$ 。如图 3-3 所示。

ControlDataRecord				
	名称	数据类型	启动值	注释
1	Static			
2	ConfigControl	Struct		
3	Block_length	USInt	16	Length of control data record, including header
4	Block_ID	USInt	196	Data record number
5	Version	USInt	5	
6	Subversion	USInt	0	
7	Slot_1	USInt	255	Assignment for CPU annex card/Actual annex card
8	Slot_2	USInt	2	Configured slot 2 / Assigned "real" slot
9	Slot_3	USInt	0	Configured slot 3 / Assigned "real" slot
10	Slot_4	USInt	3	Configured slot 4 / Assigned "real" slot
11	Slot_5	USInt	255	Configured slot 5 / Assigned "real" slot
12	Slot_6	USInt	255	Configured slot 6 / Assigned "real" slot
13	Slot_7	USInt	255	Configured slot 7 / Assigned "real" slot
14	Slot_8	USInt	255	Configured slot 8 / Assigned "real" slot
15	Slot_9	USInt	255	Configured slot 9 / Assigned "real" slot
16	Slot_101	USInt	255	Configured slot 101 / Assigned "real" slot
17	Slot_102	USInt	255	Configured slot 102 / Assigned "real" slot
18	Slot_103	USInt	255	Configured slot 103 / Assigned "real" slot

图 3-3 示例 1 的数据记录

3.2 示例 2——模块位置颠倒

项目最大组态仍如示例 1 中图 3-1 所示，但在实际安装时，原本位于插槽 3 的模块与位于插槽 4 的模块颠倒了位置。如图 3-4 所示。



图 3-4 实际安装时模块位置颠倒

要将设备组态与实际安装关联，可编辑控制数据记录 Slot_3=4，将模块分配到正确的插槽位置。

ControlDataRecord				
	名称	数据类型	启动值	注释
1	Static			
2	ConfigControl	Struct		
3	Block_length	USInt	16	Length of control data record, including header
4	Block_ID	USInt	196	Data record number
5	Version	USInt	5	
6	Subversion	USInt	0	
7	Slot_1	USInt	255	Assignment for CPU annex card/Actual annex ...
8	Slot_2	USInt	2	Configured slot 2 / Assigned "real" slot
9	Slot_3	USInt	4	Configured slot 3 / Assigned "real" slot
10	Slot_4	USInt	3	Configured slot 4 / Assigned "real" slot
11	Slot_5	USInt	255	Configured slot 5 / Assigned "real" slot
12	Slot_6	USInt	255	Configured slot 6 / Assigned "real" slot
13	Slot_7	USInt	255	Configured slot 7 / Assigned "real" slot
14	Slot_8	USInt	255	Configured slot 8 / Assigned "real" slot
15	Slot_9	USInt	255	Configured slot 9 / Assigned "real" slot
16	Slot_101	USInt	255	Configured slot 101 / Assigned "real" slot
17	Slot_102	USInt	255	Configured slot 102 / Assigned "real" slot
18	Slot_103	USInt	255	Confaired slot 103 / Assianed "real" slot

图 3-5 示例 2 的数据记录