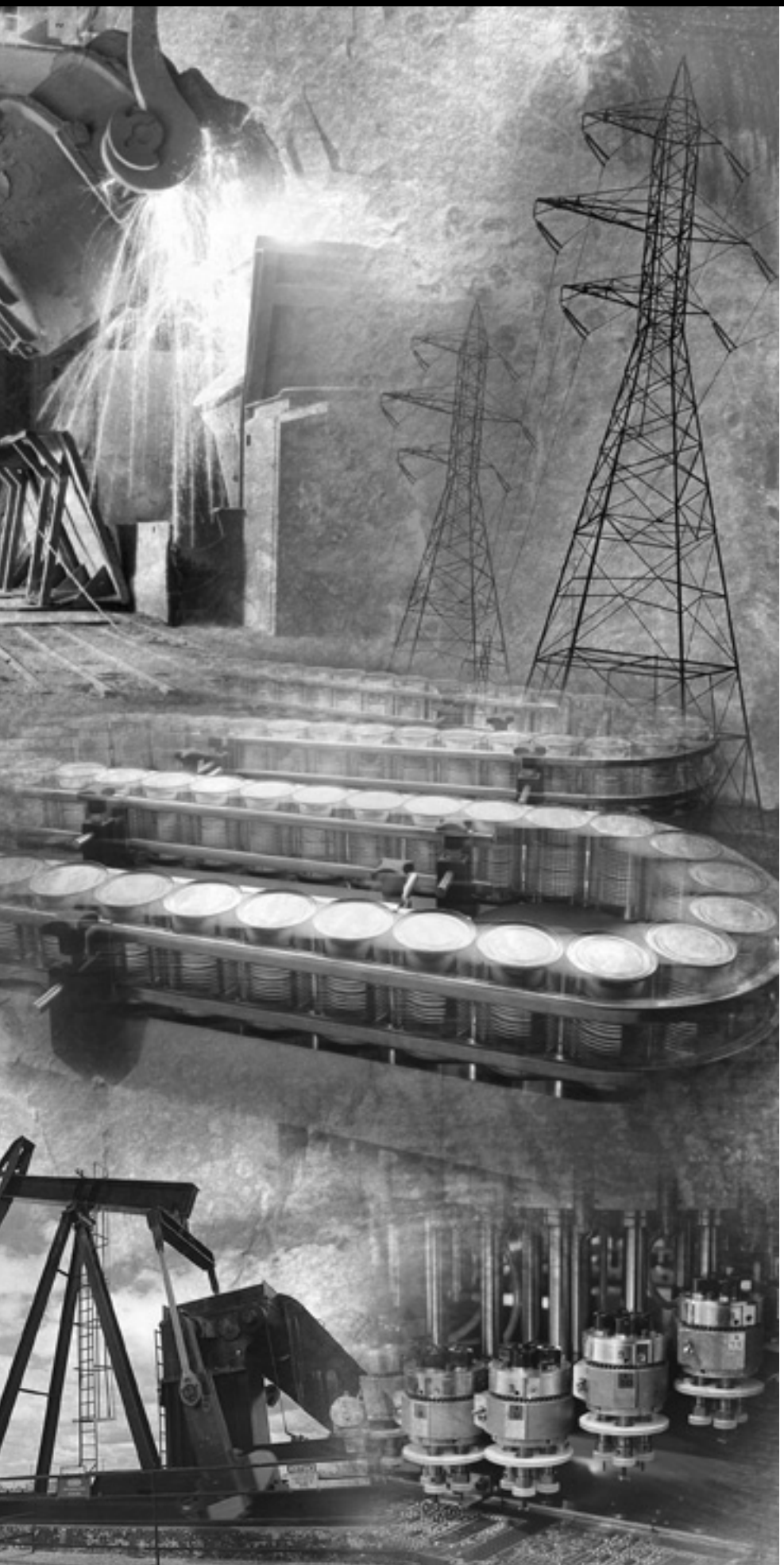


Allen-Bradley

PhaseManager™

1756 ControlLogix®,
1769 CompactLogix™,
1789 SoftLogix™,
1794 FlexLogix™,
20D PowerFlex® 700S with DriveLogix™

用户手册

Rockwell Automation

重要用户信息

固态设备与机电设备具有不同的运行特性。*Safety Guidelines for the Application, Installation and Maintenance of Solid State Controls*（《固态控制的应用、安装和维护安全准则》，出版号 SGI-1.1，可从当地 罗克韦尔自动化销售处或者从 http://www.ab.com/manuals/gi 联机获得）说明了固态设备与硬连接机电设备之间的重要差别。由于存在这些区别，同时由于固态设备的广泛应用，负责应用此设备的所有人员都必须确保仅以可接受的方式应用此设备。

在任何情况下，对于因使用或应用此设备而导致的任何直接或间接的损害，罗克韦尔自动化, Inc. 均不承担任何责任。

本手册中的示例和图表仅供说明之用。由于任何特定的安装都存在很多差异和要求，罗克韦尔自动化 , Inc. 对于依据这些示例和图表所进行的实际应用不承担任何责任和义务。

对于本手册中所述信息、电路、设备或软件的使用， 罗克韦尔自动化, Inc. 不承担专利责任。

未经罗克韦尔自动化 , Inc. 书面许可，任何单位或个人不得复制本手册的全部或部分内容。

在整本手册中，我们在必要的地方做出了说明，以告知您安全注意事项。

警告



指明在危险环境下可能导致爆炸进而造成人身伤害或死亡、财产损坏或经济损失的行为或情况的信息。

重要事项

指明成功应用和理解产品的关键信息。

注意



指明可能造成人身伤害或死亡、财产损坏或经济损失的行为或情况的信息。“注意”部分可帮助您：

- 确定危险情况
- 避免发生危险
- 了解可能的后果

有电危险



标签可贴放在设备上或者设备内 （如驱动器或电机），以警告他人可能存在危险电压。

高温危险



标签可贴放在设备上或者设备内 （如驱动器或电机），以警告他人表面温度可能很高，应注意避免危险。

何时使用本手册

本手册是 Logix5000 系列手册之一。

	如果需要：	请参见：
	初步了解 Logix5000 控制器	*Logix5000 Controllers Quick Start*（Logix5000 控制器快速入门），出版号 1756-QS001
	编写 Logix5000 控制器 — 详细信息和综合信息	*Logix5000 Controllers Common Procedures*（Logix5000 控制器通用程序），出版号 1756-PM001
您的位置	• 使用设备阶段 • 设置设备的状态模型 • 以与 S88 和 PackML 模型相似的方式进行编程	*PhaseManager User Manual*（PhaseManager 用户手册），出版号 LOGIX-UM001
	编写特定的 Logix5000 程序指令	• *Logix5000 Controllers General Instructions Reference Manual*（Logix5000 控制器指令集参考手册），出版号 1756-RM003 • *Logix5000 Controllers Process and Drives Instructions Reference Manual*（Logix5000 控制器过程和驱动指令集参考手册），出版号 1756-RM006 • *Logix5000 Controllers Motion Instructions Reference Manual*（Logix5000 控制器运动指令参考手册），出版号 1756-RM007
	从文本文件导入 Logix5000 项目或标记，或者将这些项目或标记导出到文本文件中	*Logix5000 Controllers Import/Export Reference Manual*（Logix5000 控制器导入 / 导出参考手册），出版号 1756-RM084
	将 PLC-5 或 SLC 500 应用程序转换为 Logix5000 项目	*Logix5550 Controller Converting PLC-5 or SLC 500 Logic to Logix5550 Logic Reference Manual*（Logix5550 控制器 PLC-5 或 SLC 500 逻辑到 Logix5550 逻辑转换参考手册），出版号 1756-6.8.5

如果需要：	请参见：
使用特定的 Logix5000 控制器	• *CompactLogix Controllers User Manual*（CompactLogix 控制器用户手册），出版号 1769-UM007 • *ControlLogix System User Manual*（ControlLogix 系统用户手册），出版号 1756-UM001 • *DriveLogix System 5720 User Manual*（DriveLogix System 5720 用户手册），出版号 20D-UM002 • *DriveLogix5730 Controller for PowerFlex 700S Drives with Phase II Control User Manual*（用于 PowerFlex 700S 驱动器 （阶段 II 控制）的 DriveLogix5730 控制器用户手册），出版号 20D-UM003 • *FlexLogix Controllers User Manual*（FlexLogix 控制器用户手册），出版号 1794-UM001 • *SoftLogix5800 System User Manual*（SoftLogix5800 System 用户手册），出版号 1789-UM002
通过 EtherNet/IP 网络控制设备	*EtherNet/IP Modules in Logix5000 Control Systems User Manual*（Logix5000 控制系统中的 EtherNet/IP 模块用户手册），出版号 ENET-UM001
通过 ControlNet™ 网络控制设备	*ControlNet Modules in Logix5000 Control Systems User Manual*（Logix5000 控制系统中的 ControlNet 模块用户手册），出版号 CNET-UM001
通过 DeviceNet™ 网络控制设备	*DeviceNet Modules in Logix5000 Control Systems User Manual*（Logix5000 控制系统中的 DeviceNet 模块用户手册），出版号 DNET-UM004

本手册的目的

本手册介绍如何设置和编写 Logix5000™ 控制器以使用设备阶段。其中提供的指导和示例可用于：

- 在包含设备阶段的部分中部署代码
- 设置设备的状态模型
- 对设备进行编程，以通过状态模型运行
- 使用设备阶段指令转换到不同的状态，处理故障，设置断点等等。

Logix5000 控制器指以下任何一种控制器：

- 1756 ControlLogix® 控制器
- 1769 CompactLogix™ 控制器
- 1789 SoftLogix5800™ 控制器
- 1794 FlexLogix™ 控制器
- 20D PoweFlex®700S with DriveLogix™ 控制器

本手册的目标读者

本手册面向的是编写或维护工业自动化系统的人员。

要使用本手册，您必须具备以下方面的相关经验：

- 可编程控制器
- 工业自动化系统
- 个人计算机

如何使用本手册

使用本手册时，您会看到一些与其他文本具有不同格式的文本：

文本格式：	标识内容：	示例：	解释：
《斜体》	您在屏幕或示例中看到的项的实际名称	右键单击 *User-Defined*（用户自定义）…	右键单击名为 “User-Defined”（用户自定义）的项。
courier	您需要根据应用程序提供的信息 （一个变量）	右键单击 *name_of_program* …	您必须标识应用程序中的特定程序。标识文本一般是您定义过的名称或变量。
放在中括号中	键盘上的键	按 [Enter]。	按 Enter 键。

说明：

目录

第 1 章

介绍

PhaseManager 简介 1-1
PhaseManager 带来的优势 1-3
什么是状态模型？ 1-4
如何将状态模型应用于设备？ 1-5
设备如何更改状态？ 1-6
可以手动更改状态吗？ 1-7
所属权指什么？ 1-7
如果我的设备不适于使用此状态模型，该如何？ 1-8
PhaseManager 和其他状态模型相比，情形如何？ 1-9
如何开始使用？ 1-9

第 2 章

PhaseManager 快速入门

本章目的 2-1
本章的适用情形 2-1
如何使用本章 2-1
设备 2-1
创建设备阶段 2-2
创建状态例程 2-2
手动逐一进入各状态 2-3
　准备工作 2-3
　操作 2-3
配置设备阶段的初始状态 2-6

第 3 章

准则

本章目的 3-1
本章适用情况 3-1
如何使用本章 3-1
对每个活动使用一个设备阶段 3-2
　示例 1：罐. 3-3
　示例 2：智能传送带. 3-3
填充各设备阶段的状态模型 3-4
　状态模型工作表 3-6
　状态模型工作表 3-7
　状态模型工作表 3-8
将设备阶段代码从设备代码中分离出来 3-9
　示例 1：向罐中注水. 3-10
　示例 2：智能传送带. 3-11
区分正常执行和异常 3-12
　示例 1：向罐中注水. 3-16
　示例 2：智能传送带. 3-17
使用 PCMD 指令转换为其他状态 3-18
　示例 1：罐. 3-21
　示例 2：智能传送带. 3-22
　示例 3：阻塞检测. 3-23

使用 PSC 指令在状态完成时给出信号 3-24
示例 1：向罐中注水. 3-25
示例 2：智能传送带. 3-25
创建设备接口标记 3-26
示例 1：向罐中注水. 3-28
示例 2：智能传送带. 3-29
示例 2：智能传送带（续） 3-30
使用别名标记 3-31
示例 3-31

附录 A

设备阶段指令（PSC、 PCMD、 POVR、 PFL、PCLF、PXRQ、PRNP、PPD、 PATT、 PDET）

本附录目的 A-1
约定和相关术语 A-2
设置和清除 A-2
梯形图梯级条件 A-2
例程预扫描 A-3
选择设备阶段指令 A-4
阶段状态完成 (PSC). A-5
设备阶段命令 (PCMD). A-8
设备阶段重写命令 (POVR) A-13
设备阶段失效 (PFL). A-17
设备阶段清除失效 (PCLF) A-21
设备阶段外部请求 (PXRQ) A-23
设备阶段新参数 (PRNP) A-34
设备阶段暂停 (PPD). A-37
附加到设备阶段 (PATT) A-42
脱离设备阶段 (PDET) A-47

附录 B

PHASE 数据类型

PHASE 数据类型的使用 B-1
设置和清除设备阶段标记值 B-1
PHASE 数据类型 B-2

附录 C

配置设备阶段

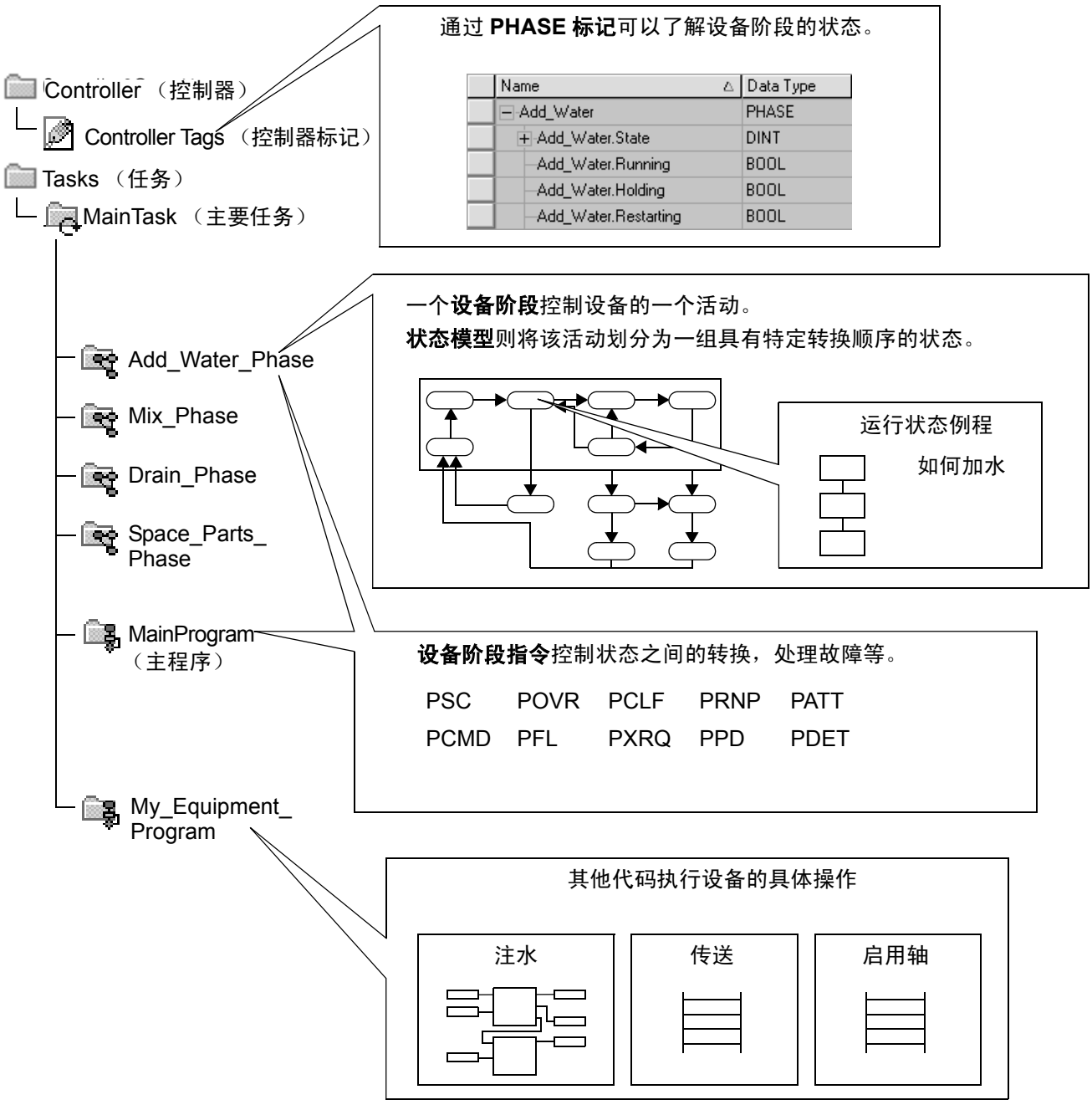
目的 C-1
适用情形 C-1
打开设备阶段的配置 C-1
配置设备阶段 C-2

第 1 章

介绍

PhaseManager 简介

PhaseManager™ 允许您向控制器添加设备阶段。设备阶段可让您更加轻松地对机器或设备的代码进行编写、使用和管理。



下面列出一些 PhaseManager 术语：

术语	说明
设备阶段	设备阶段与程序有诸多类似之处： • 设备阶段在任务中运行。 • 您要为设备阶段提供一组例程和标记。 设备阶段在以下方面有别于程序： • 设备阶段采用状态模型。 • 一个设备阶段只用于执行设备的一个活动。
状态模型	状态模型将设备的操作周期划分为一组状态。每个状态都是设备操作中的一个瞬态。它是设备在某个给定时刻的动作或条件。 设备阶段的状态模型与以下状态模型类似： • 美国标准 ISA S88.01-1995 及其 IEC 等效标准 IEC 61512-1-1998，通常称为 S88 标准 • Pack*ML*，以前受 OMAC 管理，现在是 ISA 中的工作组
状态机器	控制器有针对设备阶段的嵌入式状态机器。这大大简化了状态模型的使用。该状态机器具有以下功能： • 为动作状态调用主例程 （状态例程） • 以最少的编码对状态间的转换进行管理 您为转换条件逐一编码。当条件成立时，状态机器则将设备转换到下一所需状态。 • 确保设备沿允许的路径从一个状态进入另一个状态 例如，如果设备处于 Complete 或 Stopped 状态，设备阶段将确保其只能进入 Resetting 状态。这样，即减少了必须执行的互锁操作。
设备阶段指令	指定用于控制设备阶段的指令。请参见附录 A。
PHASE 标记	每添加一个设备阶段， RSLogix 5000 软件即为该设备阶段生成一个标记。该标记采用 PHASE 数据类型。此标记可用来： • 查看设备阶段处于哪一状态 • 保存设备阶段的失效代码 • 保存步骤索引 • 保存单元 ID • 查看对 RSBizWare Batch 软件的外部请求的状态 • 查看 RSBizWare Batch 软件是否有用于设备阶段的新参数 • 设置 producing 和 standby 状态 若要进一步了解有关 PHASE 数据类型的详细信息，请参见附录 B。

PhaseManager 带来的优势

PhaseManager 帮助您以结构化方式编写设备代码。这样一来，工厂内的所有设备即可保持协调一致。

具体来说， PhaseManager 可帮助您解决如下问题：

问	答
如何尽可能获取最佳的设备性能？	要提高设备性能，您必须先对其进行测量。状态模型提供了一种对设备状态进行测量的方式。有了测量数据，即可计算出所需的效率和性能指标。 如果在整个工厂范围内使用 PhaseManager，那么各个设备间的数据即保持一致。
如何削减将设备集成到工厂所需的开支？	清晰的结构和统一的标记，使您可以轻而易举地在工厂内添加设备，并立即建立通讯。生产线上下的设备采用相同的标记名称来共享数据，所有设备都以相同的方式与更高一级的系统进行通讯。
如何简化代码的维护？	状态模型帮助您部署常用的设备功能。我们发现，最出色的程序员往往会将状态模型用作编码的核心部分。状态模型发挥着代码地图的作用。由于结构清晰，您可以准确地找到所需的代码段。
如何为操作员提供清晰、直观的 HMI？	状态模型让您可以确保所有设备行为一致。这样， HMI 将在整个工厂范围显示一致的设备状况。当 HMI 显示设备处于 idle、 running 或 holding 状态时，操作员即可准确知晓其含义。

什么是状态模型？

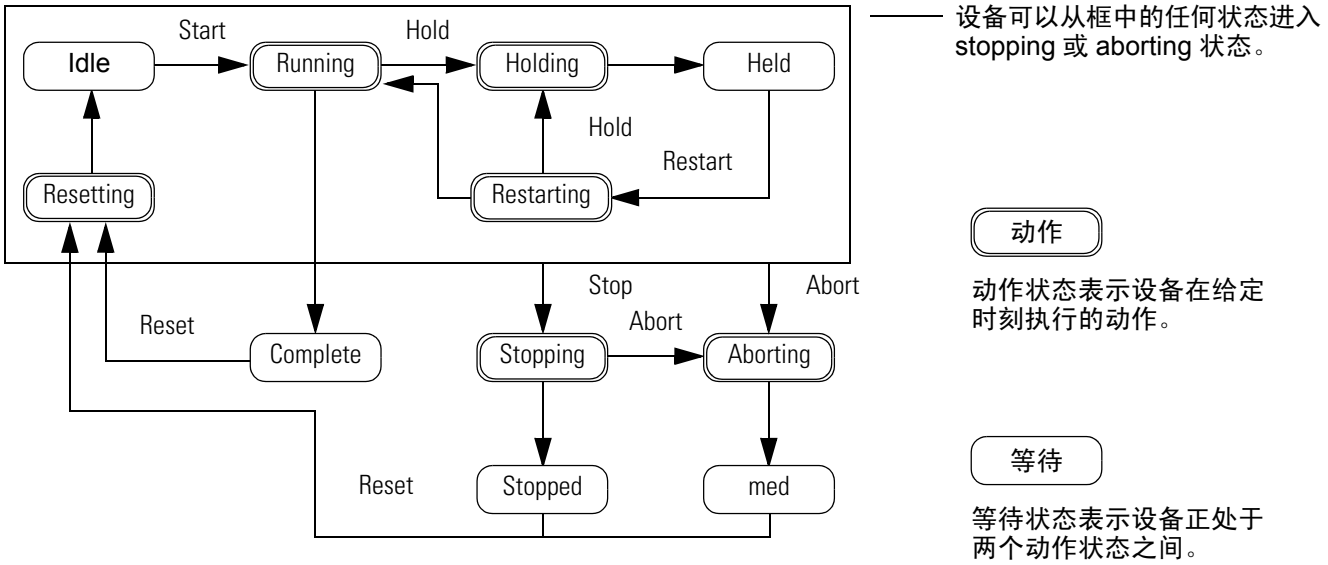
状态模型将设备的操作周期划分为一系列状态。每个状态都是设备操作中的一个瞬态。它是设备在给定时刻的动作或状况。

在状态模型中，您可以定义设备在不同条件下的行为，如运行、保持、停止等。您无须使用设备的所有状态，而只需使用所需的状态即可。

状态有 2 种类型：

状态类型	说明
动作	在某段时间内或特定条件满足之前执行某个动作或某些动作。一个动作状态可以运行一次或多次。
等待	表明已满足特定条件，设备正在等待信号以进入下一个状态。

PhaseManager 采用以下状态：



如何将状态模型应用于设备？

对程序员来讲，采用状态模型看起来似乎意味着重大改变，但实际上，这不过是以一种不同的方式来看待相同的控制问题。

通过状态模型，您可以对设备行为进行定义并将其纳入一个简明的功能规范。这样，您即可显示所发生的情形以及何时发生。

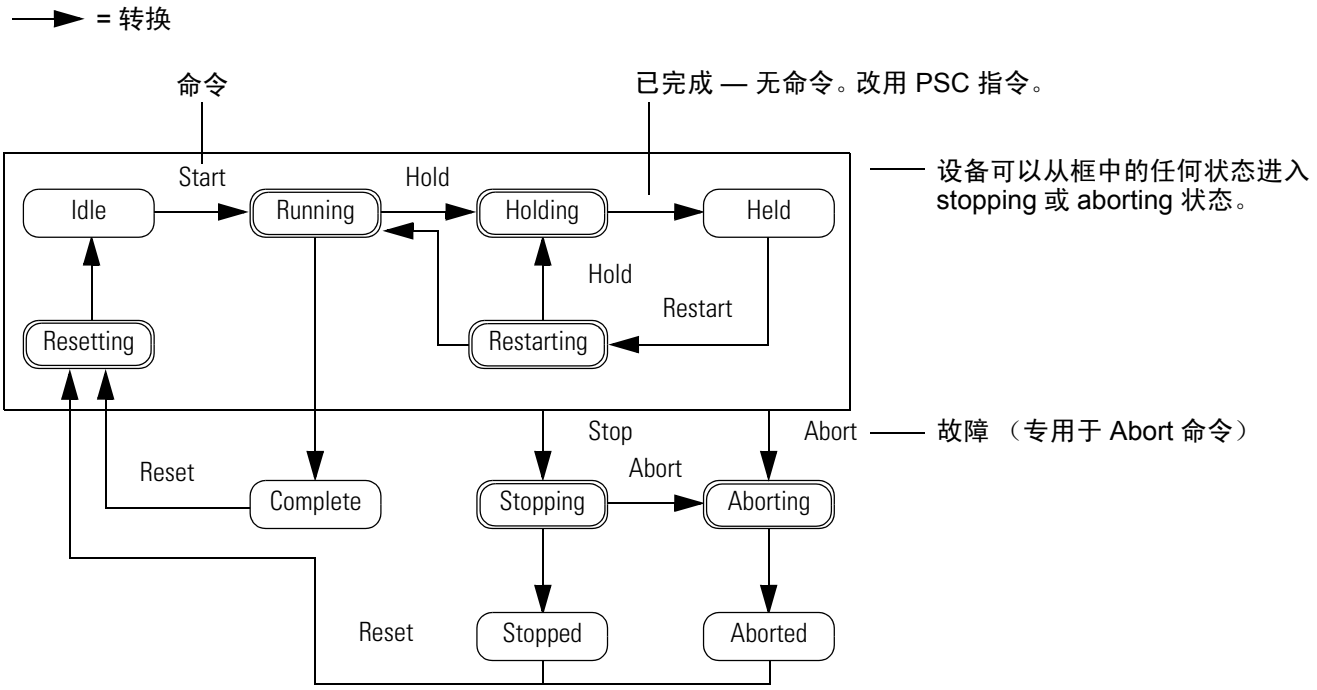
对于如下状态：	问题：
Stopped	接通电源时会发生什么情形？
Resetting	设备如何为运行做好准备？
Idle	如何判断设备已准备好，可以运行？
Running	设备生产产品要执行什么操作？
Holding	设备如何 《**临时**》停止生产，同时又不产生废料？
Held	如何判断设备是否处于安全的 holding 状态？
Restarting	holding 状态之后，设备如何恢复生产？
Complete	如何判断设备何时完成需要执行的操作？
Stopping	正常关机的过程中会发生什么？
Aborting	如果出现故障或操作失败，设备如何关机？
Aborted	如何判断设备是否安全关机？

设备如何更改状态？

状态模型中的箭头指示设备从当前所处状态能进入哪些状态。

- 每个箭头称为一次转换。
- 状态模型使设备只能执行特定的转换。这样，使用相同模型的设备就会具有相同的行为。

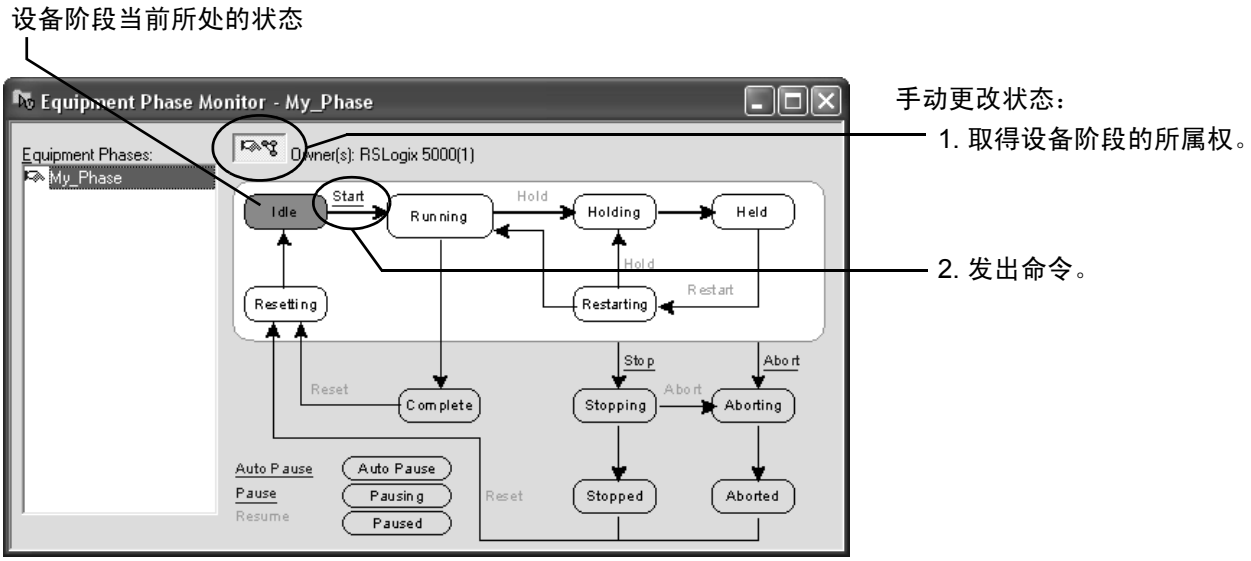
PhaseManager 采用以下转换：



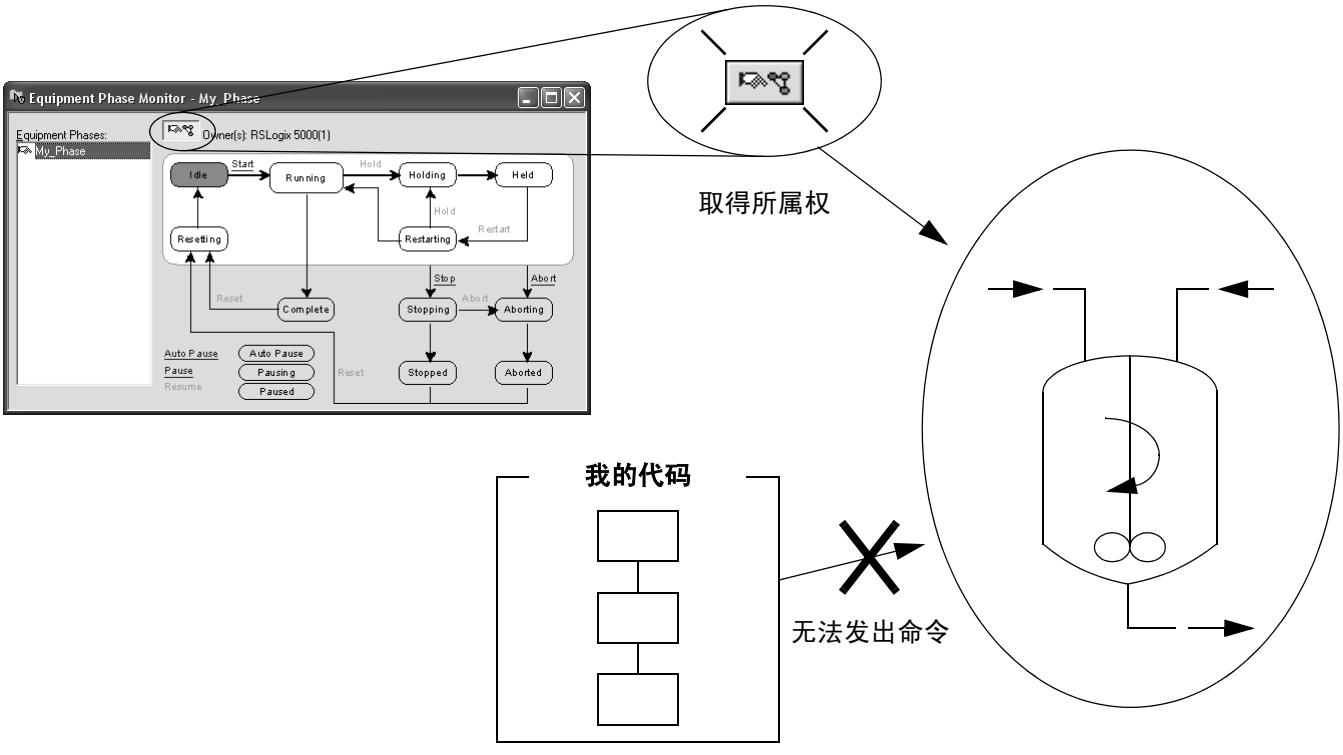
转换类型	说明
命令	命令用来指示设备开始执行某个动作或执行不同的动作。例如，操作员按启动按钮以开始生产，或按停止按钮以关机。 PhaseManager 采用以下命令： reset　stop　restart start　hold　abort
完成	设备完成当前操作后进入对应的等待状态。您不向设备发出命令，而是创建用来在设备执行完当前动作后发出信号的代码。等待状态显示设备已执行完动作。
故障	故障用来提示出现非正常状况。您可以通过代码查找故障并在找到后执行特定的动作。假定您希望设备在发生特定故障时尽快关机，在此情况下，即可通过代码来查找该故障，并在找到后执行 abort 命令。

可以手动更改状态吗？

是。RSLogix 5000 软件通过一个窗口来监视设备阶段并向设备阶段发出命令。



所属权指什么？



所属权用来对程序或 RSBizWare Batch 软件进行锁定，使其无法向设备阶段发出命令。

如果设备阶段的所属权属于	那么
RSLogix 5000 软件	定序器无法向设备阶段发出命令。 这包括： • 内部定序器 — 控制器中的程序 • 外部定序器 — RSBizWare Batch 软件
内部定序器 — 控制器中的程序	其他定序器无法向设备阶段发出命令。
外部定序器 — RSBizWare Batch 软件	其他定序器无法向设备阶段发出命令。

例外情形：使用设备阶段重写命令 (POVR) 指令可发出 hold、 stop 或 abort 命令，而不管所属权属于哪一方。

请参见下列指令：

- 设备阶段命令 (PCMD) 指令，第 A-8 页。
- 设备阶段重写命令 (POVR) 指令，第 A-13 页。
- 附加到设备阶段 (PATT) 指令，第 A-42 页。

如果我的设备不适于使用此状态模型，该如何？

状态模型的一个共有缺陷是它并非适用于所有设备。您可能会听到以下说法，或是产生这样的想法：“我的设备非常复杂，有很多同步动作，而且许多环节都要并行运行。”

请切记，状态模型是从一个非常常规的层面来看待您的设备的。不同的设备执行不同的任务，而对于所执行的具体任务则需要编写特定代码。状态模型只是为您的代码给出了一个较高层次的框架：

- 状态模型对设备的常规行为、命令和状态进行定义。
- 您则可以在这个框架内为设备编写具体代码。

PhaseManager 和其他状态模型相比，情形如何？

下表将 PhaseManager 状态模型与其他常用状态模型进行了比较：

S88	Pack*ML*	PhaseManager
Idle	Starting ⇒ Ready	Resetting ⇒ Idle
Running ⇒ Complete	Producing	Running ⇒ Complete
Pausing ⇒ Paused	Standby	子例程、断点或二者兼有。
Holding ⇒ Held	Holding ⇒ Held	Holding ⇒ Held
Restarting	无	Restarting
Stopping ⇒ Stopped	Stopping ⇒ Stopped	Stopping ⇒ Stopped
Aborting ⇒ Aborted	Aborting ⇒ Aborted	Aborting ⇒ Aborted

如何开始使用？

要开始使用 PhaseManager，请按以下步骤操作：

1. 请安装 RSLogix 5000 15.0 软件或其更高版本。
2. 请参见本手册的其他部分。

若要了解以下信息	请参见
试用 PhaseManager	第 2 章
PhaseManager 项目编程指南	第 3 章
设备阶段指令的详细参考资料	附录 A
PHASE 数据类型的成员说明	附录 B

说明：

第 2 章

PhaseManager 快速入门

本章目的

此快速入门可用来：

- 通过简介了解设备阶段的运行方式
- 监视设备阶段
- 通过手动方式让设备阶段进入其他状态

本章的适用情形

下列情形下可使用本快速入门：

- 第一次试用 PhaseManager 时
- 手动逐一进入各状态，以测试设备阶段

如何使用本章

要使用此快速入门，请按以下步骤操作：

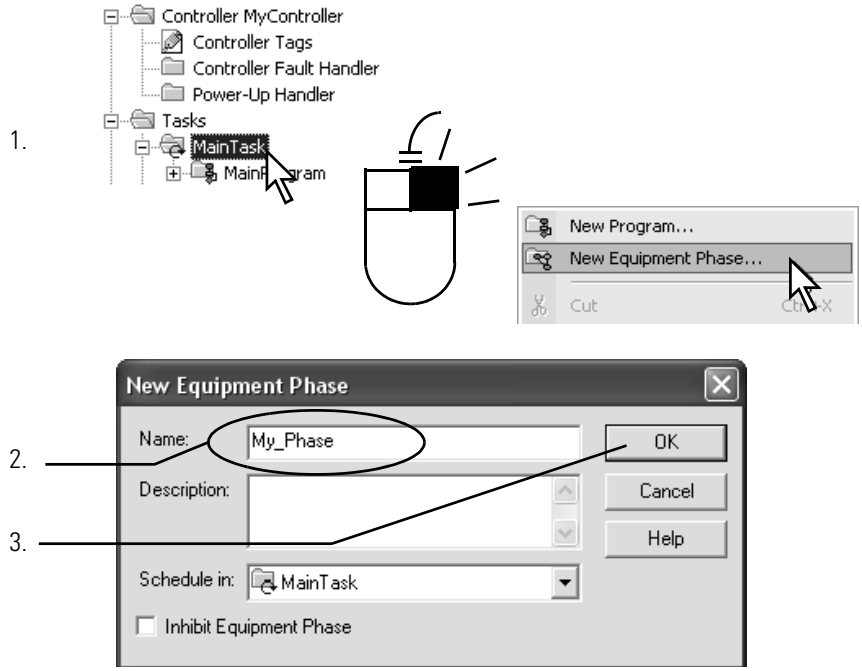
操作	页码
创建设备阶段	2-2
创建状态例程	2-2
手动逐一进入各状态	2-3
配置设备阶段的初始状态	2-6

设备

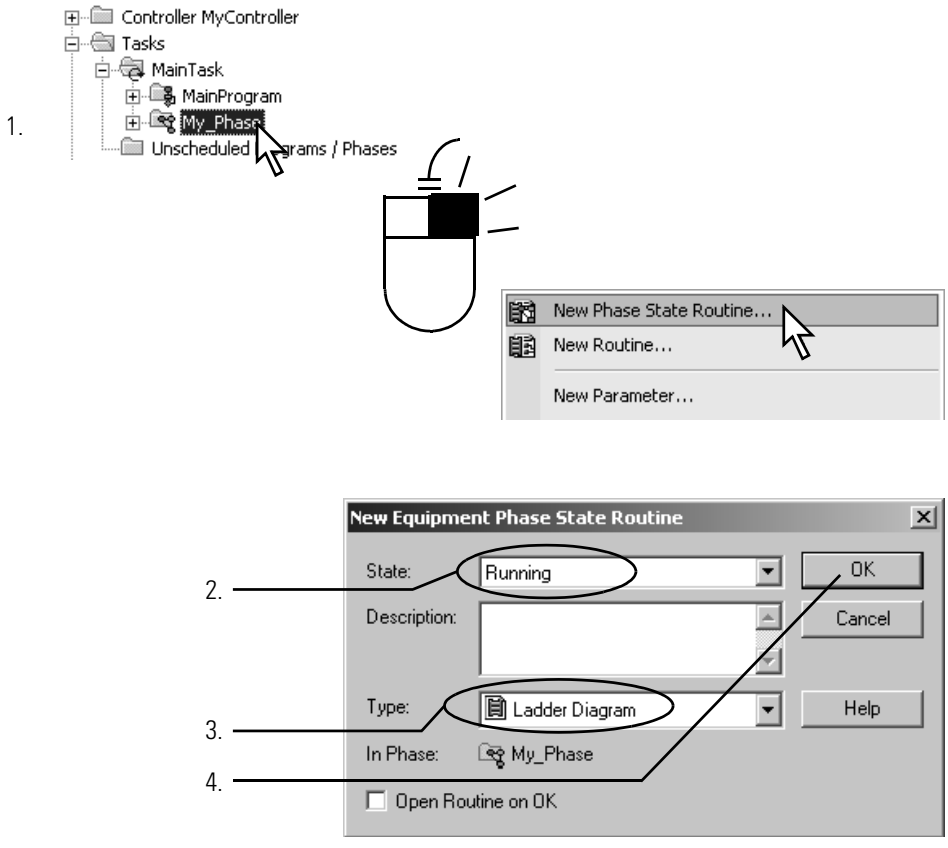
要使用本快速入门，您需要：

- Logix5000 控制器。如果您不能确定哪些控制器是 Logix5000 控制器，请参见前言。
- 用于此控制器的 Firmware 15.0 或更高版本
- 用于此控制器的电源
- 与控制器相连接的通讯路径
  - 通讯卡或内置端口
  - 相应的通讯电缆
- RSLogix 5000 15.0 或更高版本软件

创建设备阶段



创建状态例程



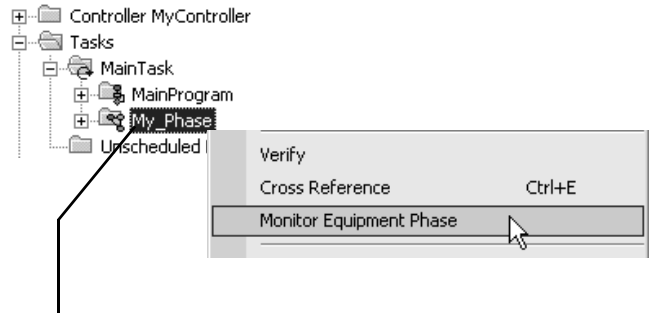
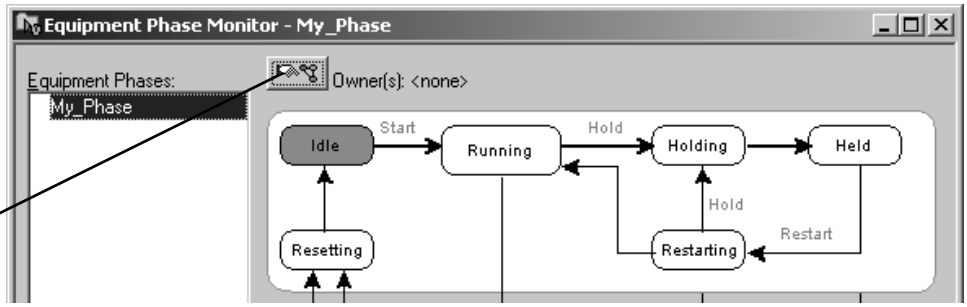
手动逐一进入各状态

准备工作

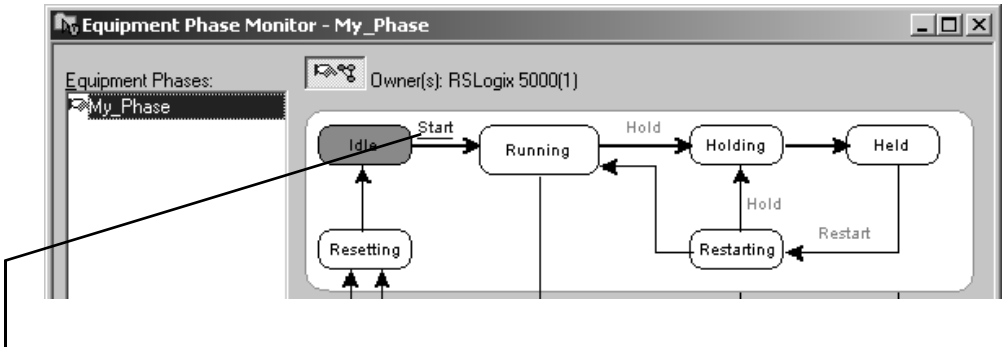
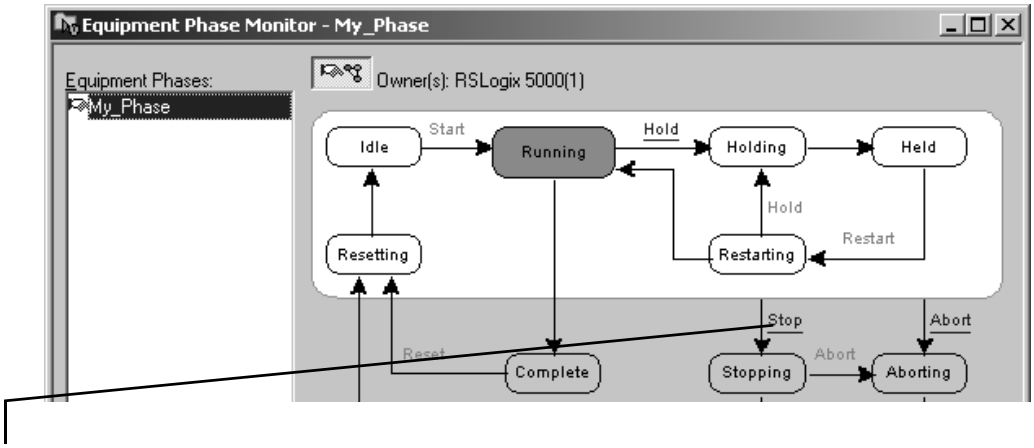
开始执行此程序前，需要执行以下操作：

- 将项目下载到控制器。
- 将控制器置于运行或远程运行模式。

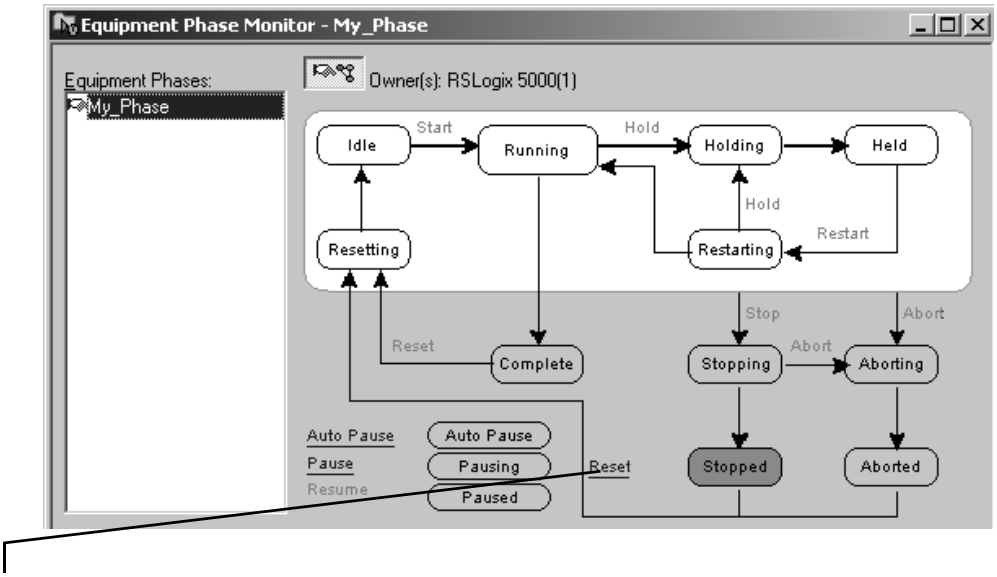
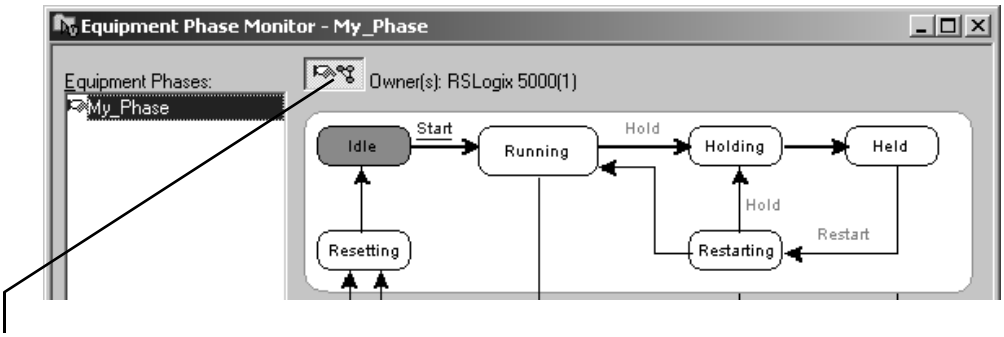
操作

步骤	说明
 1. 右键单击设备阶段并选择 *Monitor Equipment Phase*（监视设备阶段）。	
 2. 单击所属权按钮，然后选择 *Yes—take ownership*（是 -- 取得所属权）。	这种方式可让您通过此窗口逐一进入各个状态。

《转下页》

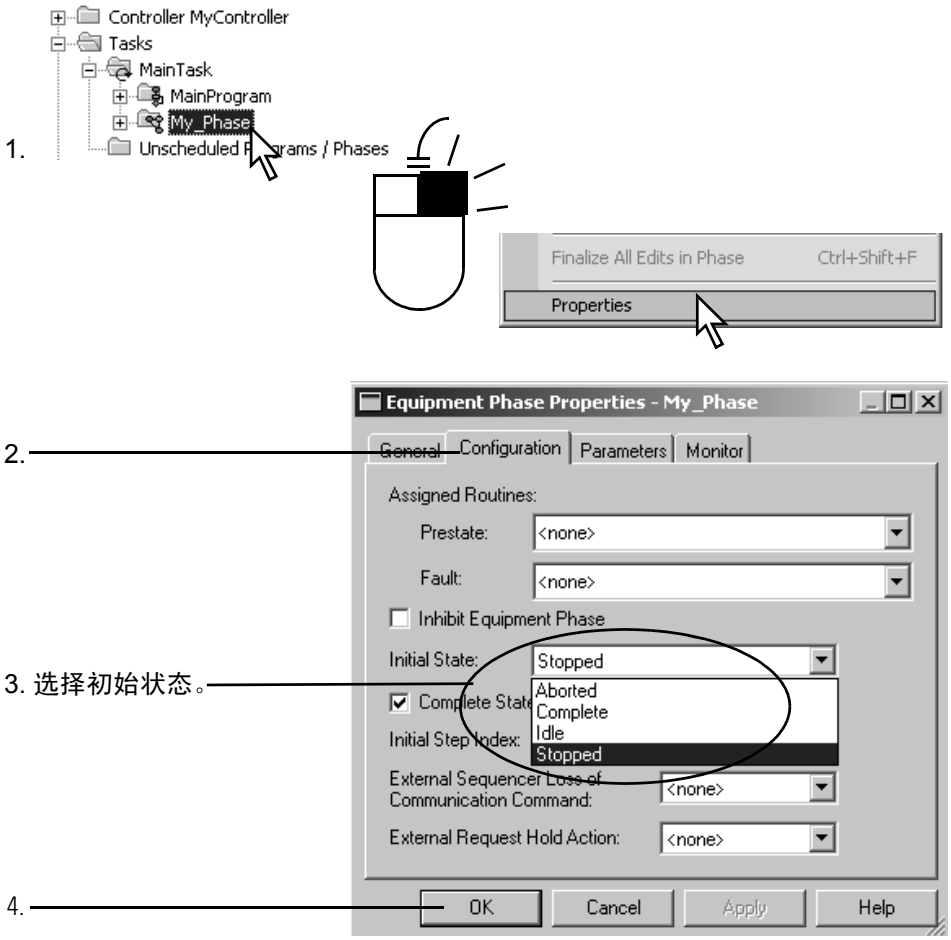
步骤	说明
 3. 单击 *Start*（启动）。	• 设备阶段进入 Running 状态。 • Running 状态例程中如有任何代码，即开始运行。这里的代码是为设备的标准生产顺序编写的代码。
 4. 单击 *Stop*（停止）。	• 设备阶段进入 Stopped 状态。 • Running 状态例程停止运行。 • Stopping 状态例程为可选项。如果没有该例程，设备阶段将直接进入 Stopped 状态。

《转下页》

步骤	说明
	
5. 单击 *Reset*（复位）。	• 设备阶段进入 Idle 状态。 • Resetting 状态例程为可选项。如果没有该例程，设备阶段将直接进入 Idle 状态。
	
6. 单击所属权按钮。	此窗口即释放对该设备阶段的控制权。

配置设备阶段的初始状态

初始状态是通电后设备阶段进入的第一个状态。



第 3 章

准则

本章目的

就如何开发使用设备阶段的 Logix5000 项目以及进行相应的编程提供指导。

本章适用情况

本章适用于以下情况：

- 在对 Logix5000 项目的设备阶段进行布局之前
- 在对项目编程时作为参考

如何使用本章

设计项目前请仔细阅读以下准则并在需要时进行参考：

准则	页码
对每个活动使用一个设备阶段	3-2
填充各设备阶段的状态模型	3-4
将设备阶段代码从设备代码中分离出来	3-9
区分正常执行和异常	3-12
使用 PSC 指令在状态完成时给出信号	3-24
使用 PCMD 指令转换为其他状态	3-18
创建设备接口标记	3-26
使用别名标记	3-31

对每个活动使用一个设备阶段

每个设备阶段都是设备所进行的一项特定活动。设备阶段通知设备该执行什么活动以及何时执行该活动。

确定要使用多少设备阶段时请遵循以下准则：

<table>
<tr><th>准则</th><th>说明</th></tr>
<tr><td>1. 确保每个设备阶段都进行一项独立的活动。</td><td>确保每个设备阶段都进行一项独立 （相对独立）于其他设备的活动。设备阶段控制共同进行该特定活动的所有设备。

示例
<table><tr><th>这可能是设备阶段</th><th>这可能不是设备阶段</th></tr><tr><td>• 将产品装瓶。
• 将瓶装箱。
• 向罐中注水。
• 混合罐中原料。</td><td>• 加快填料轴转速 （过小）
• 运行装瓶线 （过大）
• 打开水阀 （过小）
• 调配原料 （过大）</td></tr></table></td></tr>
<tr><td>2. 保持设备阶段和程序的数目不超过下面的限制。</td><td><table><tr><th>对于此控制器：</th><th>设备阶段和程序的最大数目为：</th></tr><tr><td>ControlLogix</td><td>每个任务 100 个程序和设备阶段</td></tr><tr><td>SoftLogix</td><td>每个任务 100 个程序和设备阶段</td></tr><tr><td>FlexLogix</td><td>每个任务 32 个程序和设备阶段</td></tr><tr><td>CompactLogix</td><td>每个任务 32 个程序和设备阶段</td></tr></table></td></tr>
<tr><td>3. 列出随每个设备阶段运行的设备。</td><td>示例
<table><tr><th>设备阶段</th><th>相关设备</th></tr><tr><td>Add_Water</td><td>水泵
水阀
限位开关</td></tr><tr><td>Smart_Belt</td><td>粗带轴
精带轴
出口带轴</td></tr></table></td></tr>
</table>

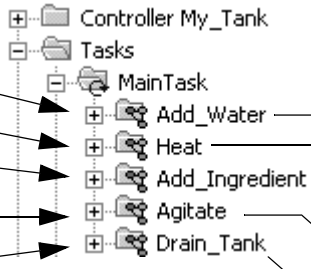
示例 1：罐

下面的示例显示了制作原料的罐的设备阶段。

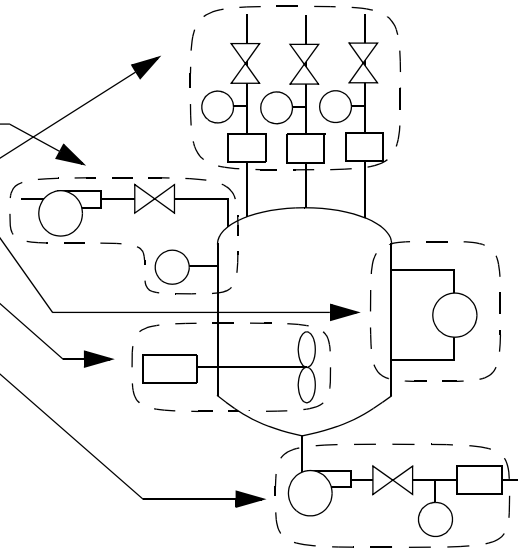
要制作原料，罐需要进行以下操作：

1. 注水。
2. 加热水。
3. 添加其他原料。
4. 混合所有原料。
5. 输出成品。

组成阶段：

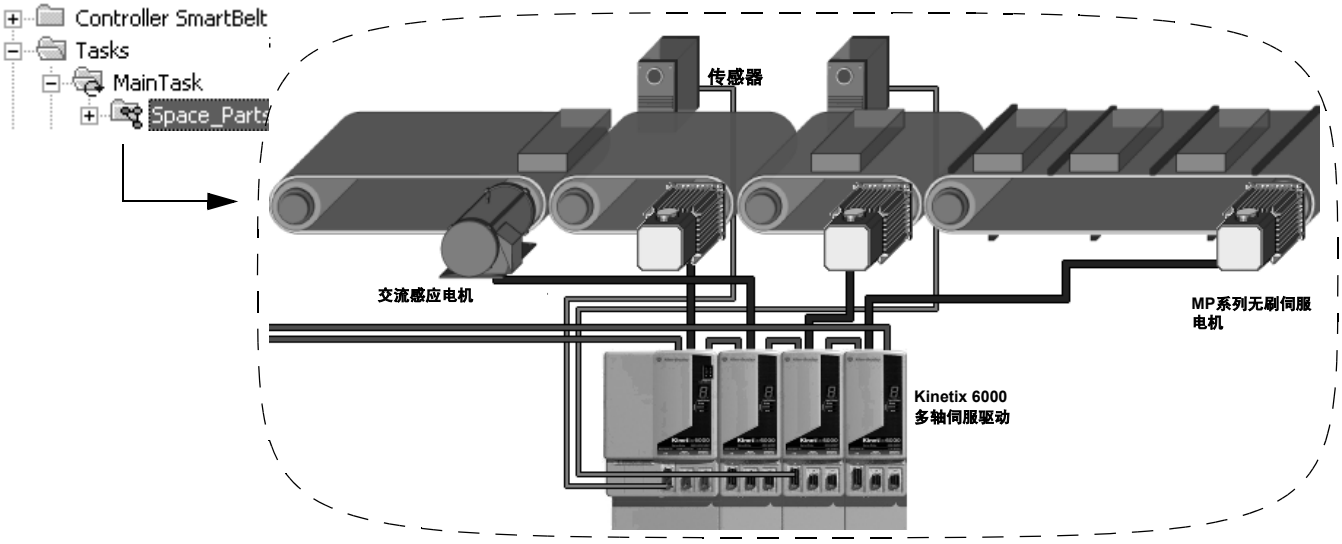


控制设备：



示例 2：智能传送带

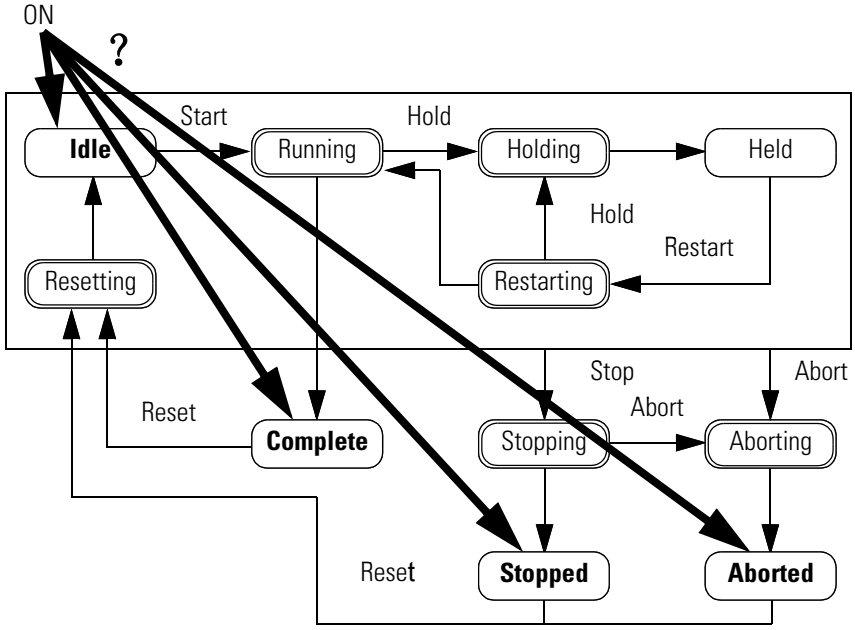
下面的示例显示了一个智能传送带。该智能传送带只进行一项活动。即将产品在出口带上均匀分开。因为它只进行一项活动，所以只需要 1 个设备阶段。



填充各设备阶段的状态模型

状态模型将设备的操作周期划分为一系列状态。每个状态是设备操作中的一个瞬态。它是设备在给定时间的动作和状况。

填充设备阶段的状态模型时请遵循以下准则：

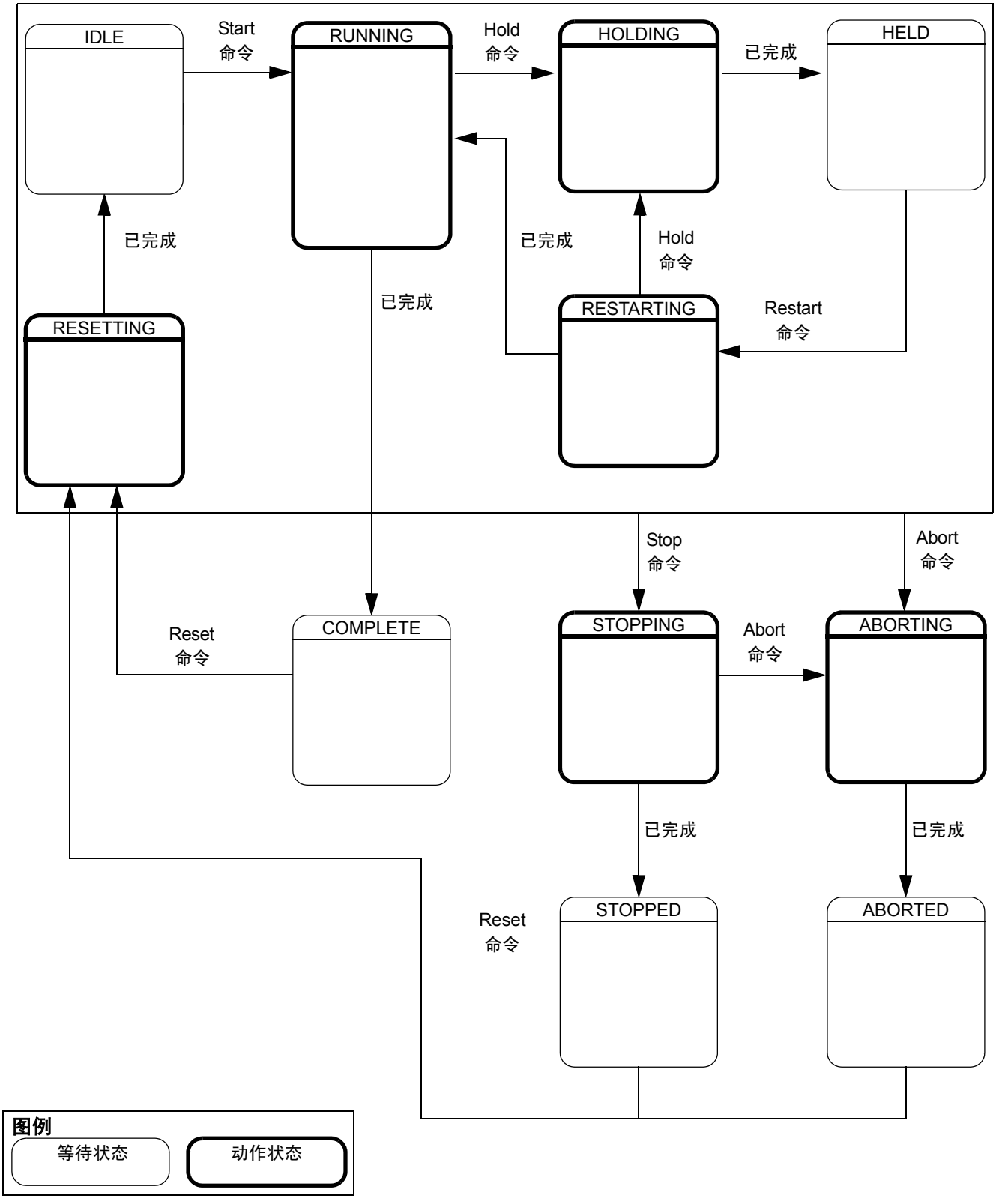
准则	说明
1. 为每个阶段填充 1 个状态模型。	各阶段运行自己的一组状态。为每个阶段填充 1 个状态模型工作表。
2. 确定通电后要用作初始状态的状态。	打开电源时要使设备阶段处于哪个状态？  打开电源时设备阶段进入其初始状态。建议您使用以下状态之一作为初始状态： • idle （默认） • Complete • Stopped 选择能最好地表示设备在通电后等待执行的动作 （重置、运行，等等）的初始状态。

<table>
<tr><th>准则</th><th>说明</th></tr>
<tr><td>3. 从初始状态开始并完成模型。</td><td>从初始状态开始。然后从这一点开始继续执行。下面的问题可帮助您继续操作：
<table>
<tr><th>状态：</th><th>问题：</th></tr>
<tr><td>Stopped</td><td>打开电源时发生了什么？</td></tr>
<tr><td>Resetting</td><td>如何使设备准备好可以运行？</td></tr>
<tr><td>Idle</td><td>如何得知设备已准备好可以运行？</td></tr>
<tr><td>Running</td><td>设备如何进行生产？</td></tr>
<tr><td>Holding</td><td>设备如何暂停才能不产生废料？</td></tr>
<tr><td>Held</td><td>如何得知设备是否已安全暂停？</td></tr>
<tr><td>Restarting</td><td>设备暂停后如何恢复生产？</td></tr>
<tr><td>Complete</td><td>如何得知设备完成了它必须执行的任务？</td></tr>
<tr><td>Stopping</td><td>正常关闭时发生了什么？</td></tr>
<tr><td>Aborting</td><td>发生故障或失效时设备如何关闭？</td></tr>
<tr><td>Aborted</td><td>如何得知设备是否安全关闭？</td></tr>
</table></td></tr>
<tr><td>4. 仅使用所需的状态。</td><td>仅定义适用于设备的状态。“无”需使用所有状态。设备阶段只是跳过所有未添加的状态。</td></tr>
<tr><td>5. 对于 producing 和 standby 状态，使用子例程。</td><td>如果要为设备定义 producing 和 standby 状态，请使用子例程。
A. 为 producing 状态和 standby 状态分别创建一个例程。
B. 在 running 状态中，检查生产与备用条件。设置设备阶段标记的 Producing 位或 Standby 位。
C. 使用 Producing 和 Standby 位作为条件调用相应的例程。

请参见附录 B。</td></tr>
</table>

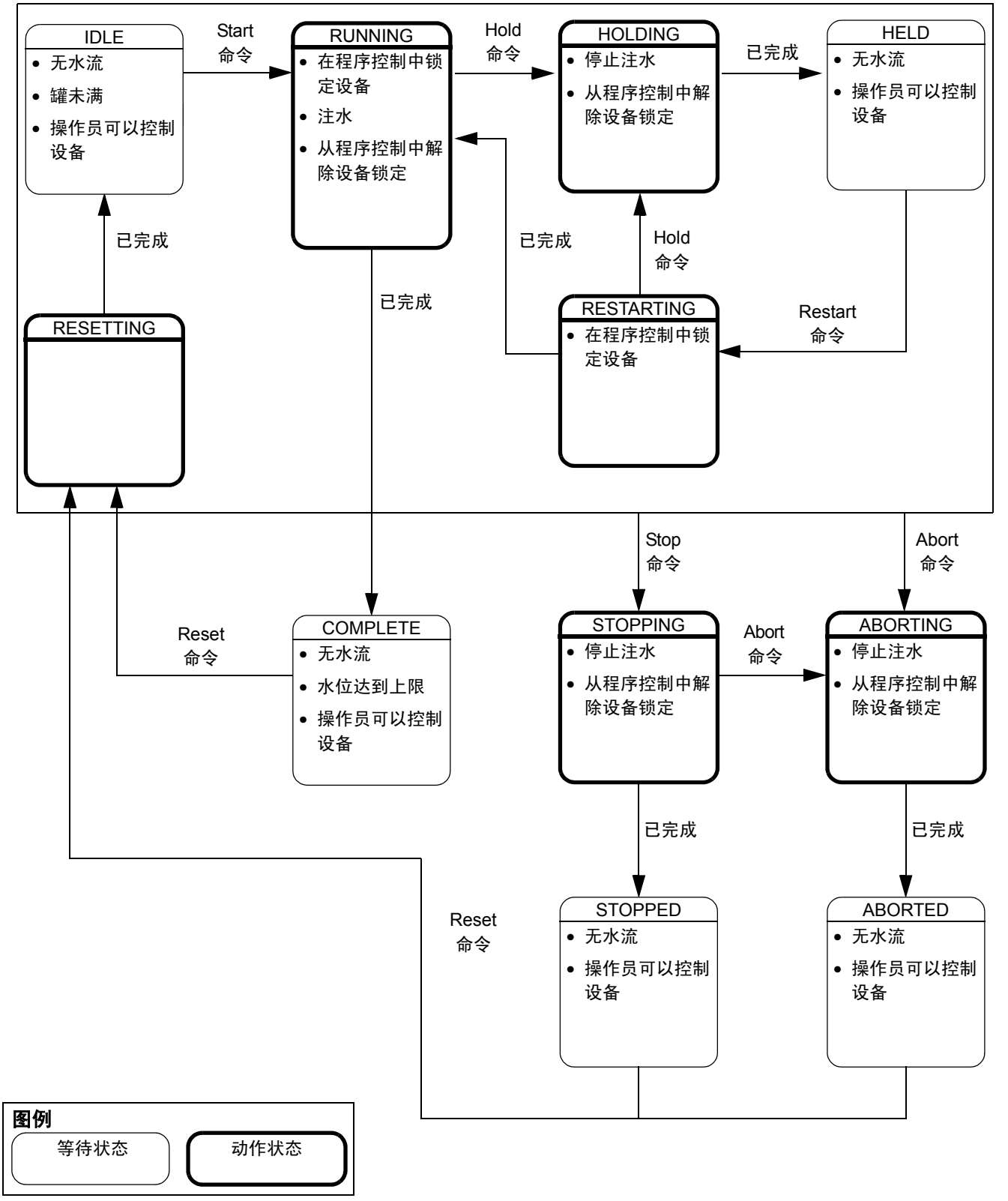
状态模型工作表

设备阶段：



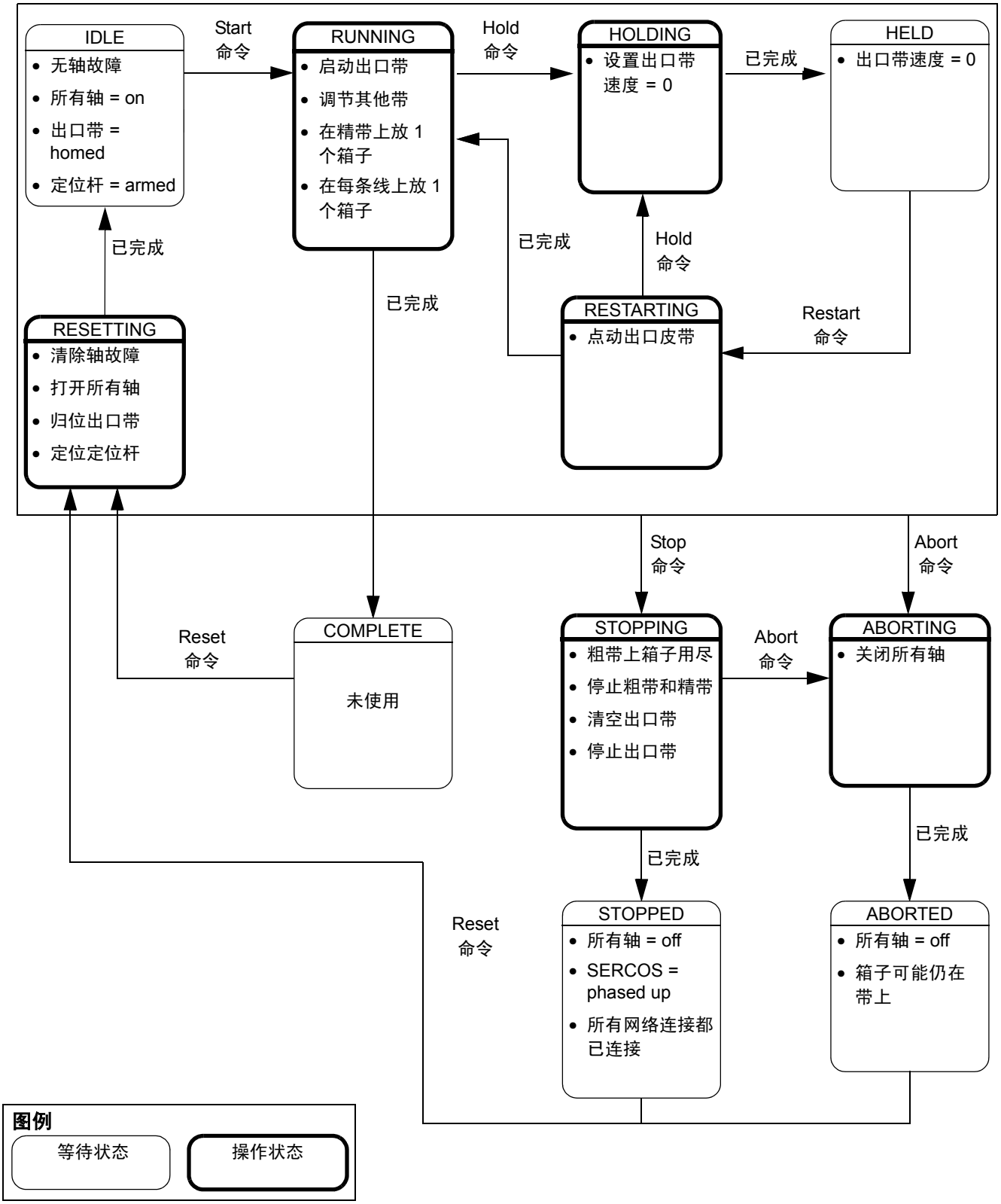
状态模型工作表

设备阶段：《注水》



状态模型工作表

设备阶段：《分隔部件》



将设备阶段代码从设备代码中分离出来

设备阶段的一个优点是，使用它可以将制造产品的过程 （步骤）与对制造产品的设备的控制分离。这大大简化了使用同一设备执行不同过程来生产不同产品的过程。

设备阶段

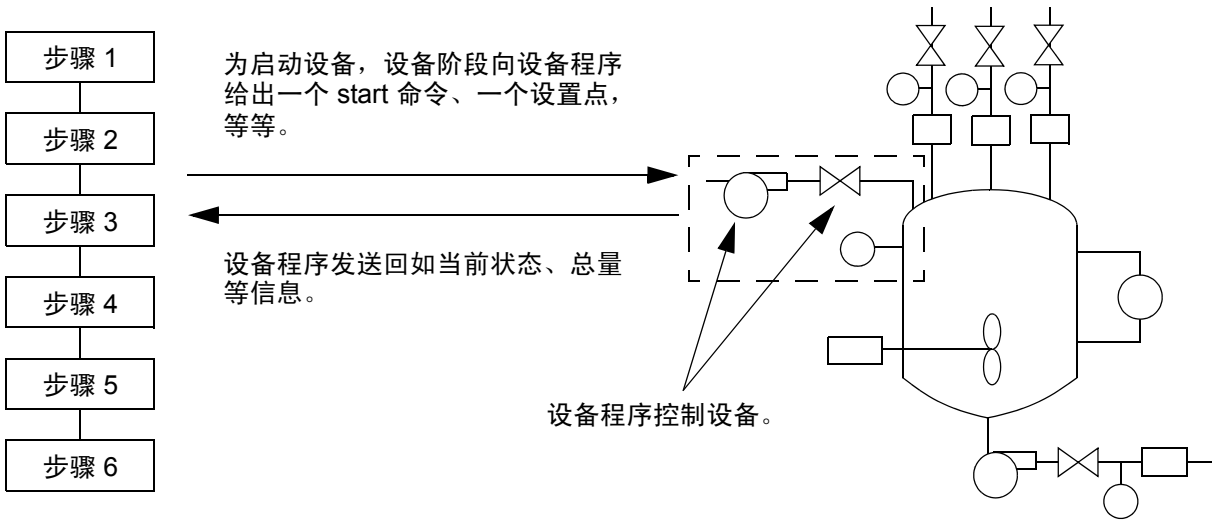
控制设备的操作 （做什么以及何时做）

- 生产产品
- 停止生产产品
- 向罐中注水
- 等待操作员进行操作

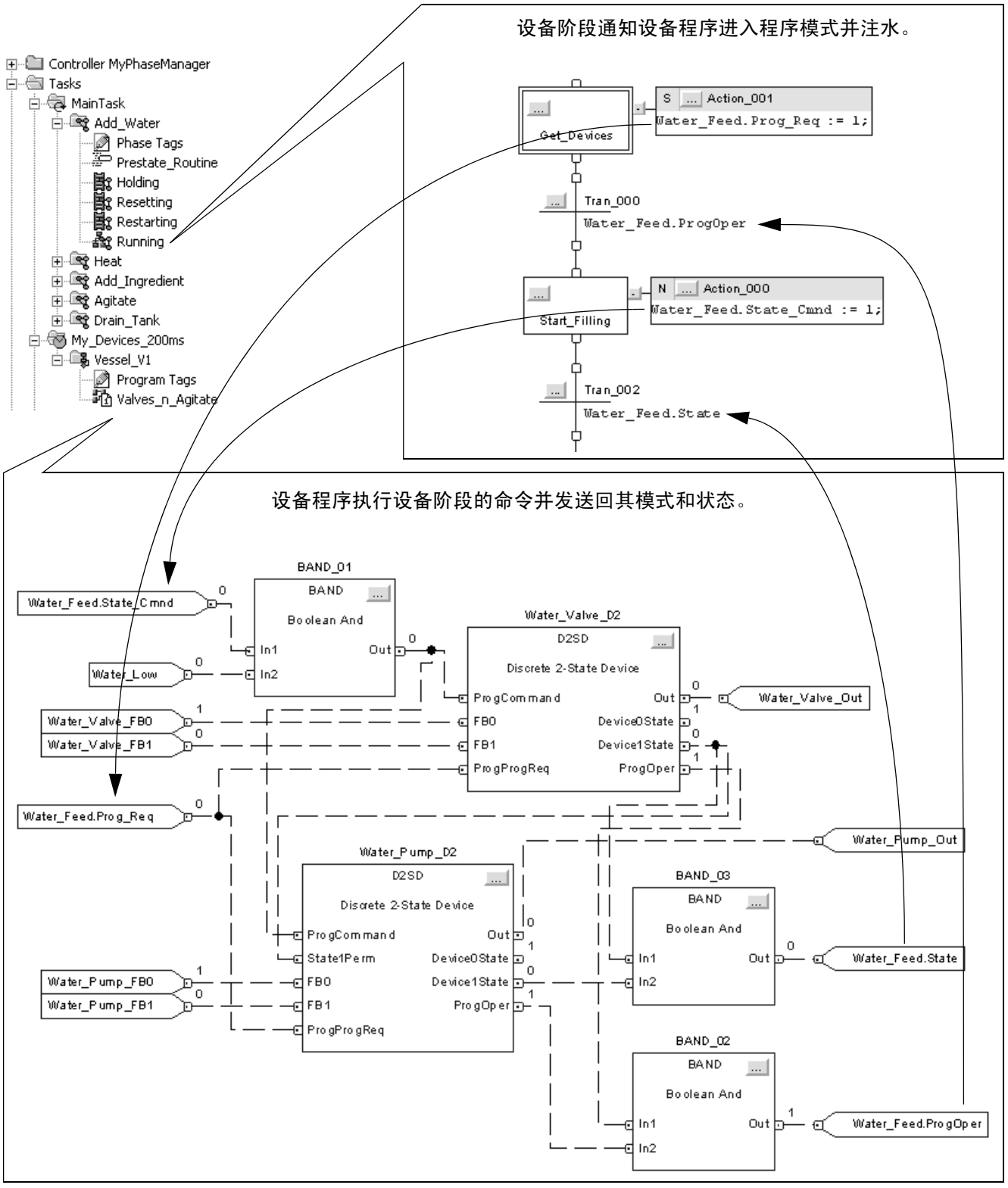
设备程序

执行一组特定装置的动作（执行）

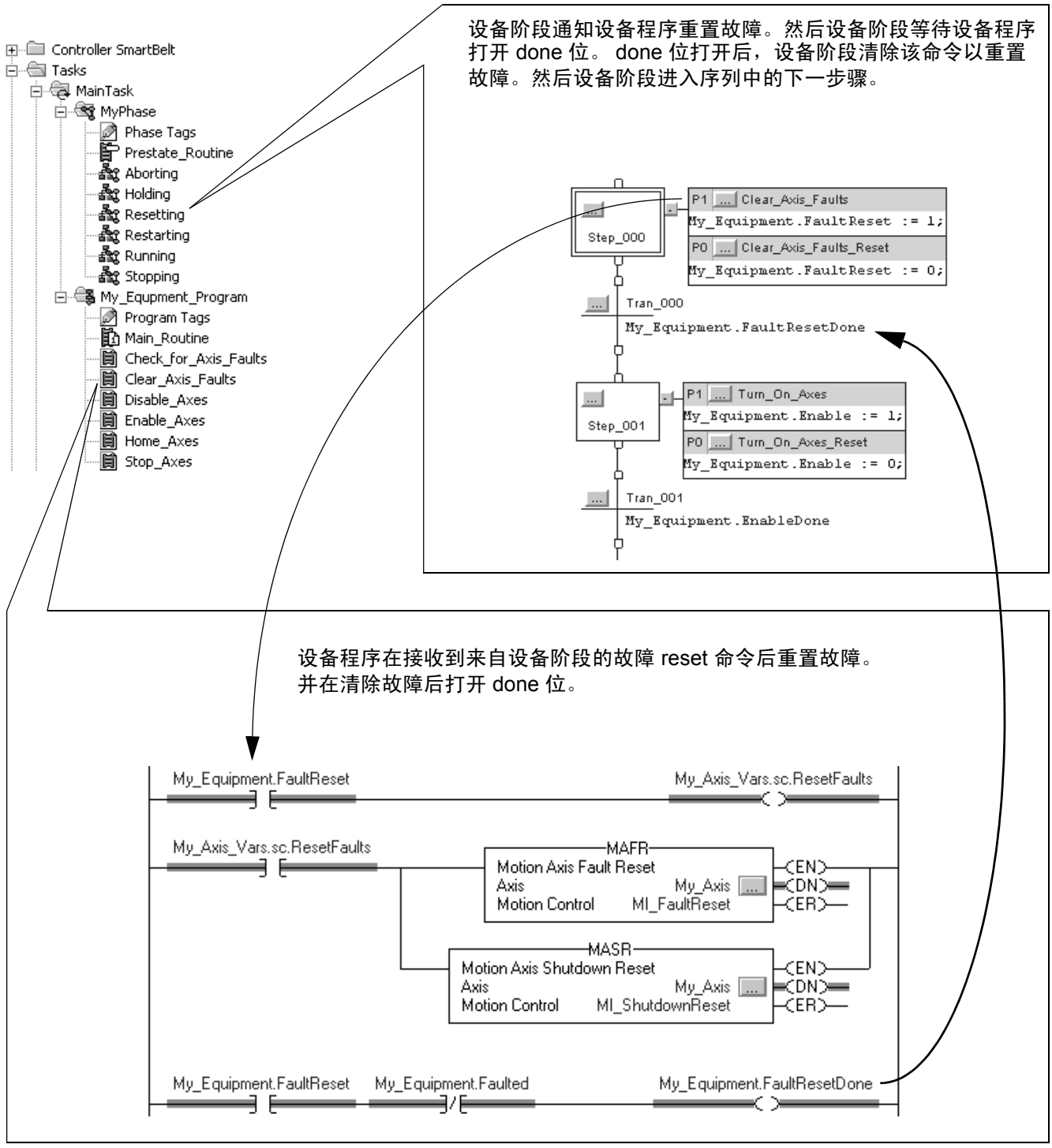
- 推动轴
- 运行泵
- 打开阀门
- 计算控制变量



示例 1：向罐中注水


设备阶段通知设备程序进入程序模式并注水。
Controller MyPhaseManager
Tasks
MainTask
Add_Water
Phase Tags
Prestate_Routine
Holding
Resetting
Restarting
Running
Heat
Add_Ingredient
Agitate
Drain_Tank
My_Devices_200ms
Vessel_V1
Program Tags
Valves_n_Agitate
Get_Devices
S Action_001
Water_Feed.Prog_Req := 1;
Tran_000
Water_Feed.ProgOper
Start_Filling
N Action_000
Water_Feed.State_Cmnd := 1;
Tran_002
Water_Feed.State
设备程序执行设备阶段的命令并发送回其模式和状态。
BAND_01
BAND
Boolean And
In1
In2
Out
Water_Feed.State_Cmnd
Water_Low
Water_Valve_D2
D2SD
Discrete 2-State Device
ProgCommand
FB0
FB1
ProgProgReq
Out
DeviceOState
Device1State
ProgOper
Water_Valve_Out
Water_Valve_FB0
Water_Valve_FB1
Water_Feed.Prog_Req
Water_Pump_Out
Water_Pump_D2
State1Perm
Water_Pump_FB0
Water_Pump_FB1
BAND_03
Water_Feed.State
BAND_02
Water_Feed.ProgOper

示例 2：智能传送带

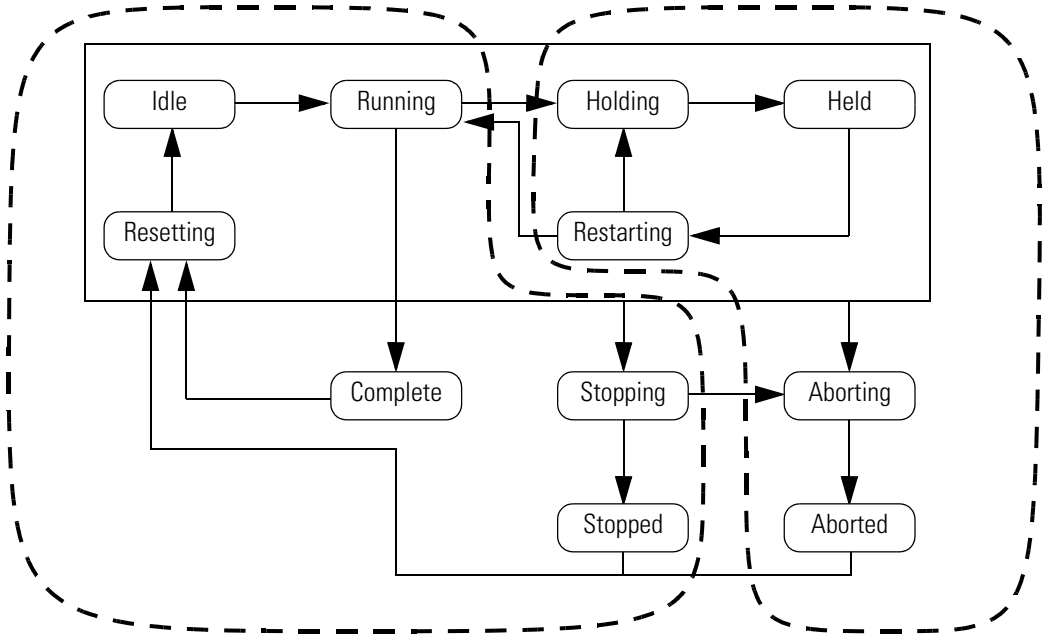

Controller SmartBelt
Tasks
MainTask
MyPhase
Phase Tags
Prestate_Routine
Aborting
Holding
Resetting
Restarting
Running
Stopping
My_Equpment_Program
Program Tags
Main_Routine
Check_for_Axis_Faults
Clear_Axis_Faults
Disable_Axes
Enable_Axes
Home_Axes
Stop_Axes
设备阶段通知设备程序重置故障。然后设备阶段等待设备程序打开 done 位。 done 位打开后，设备阶段清除该命令以重置故障。然后设备阶段进入序列中的下一步骤。
Step_000
P1 Clear_Axis_Faults
My_Equipment.FaultReset := 1;
P0 Clear_Axis_Faults_Reset
My_Equipment.FaultReset := 0;
Tran_000
My_Equipment.FaultResetDone
Step_001
P1 Turn_On_Axes
My_Equipment.Enable := 1;
P0 Turn_On_Axes_Reset
My_Equipment.Enable := 0;
Tran_001
My_Equipment.EnableDone
设备程序在接收到来自设备阶段的故障 reset 命令后重置故障。并在清除故障后打开 done 位。
My_Equipment.FaultReset
My_Axis_Vars.sc.ResetFaults
My_Axis_Vars.sc.ResetFaults
MAFR
Motion Axis Fault Reset
Axis
My_Axis
Motion Control
MI_FaultReset
EN
DN
ER
MASR
Motion Axis Shutdown Reset
Axis
My_Axis
Motion Control
MI_ShutdownReset
EN
DN
ER
My_Equipment.FaultReset
My_Equipment.Faulted
My_Equipment.FaultResetDone

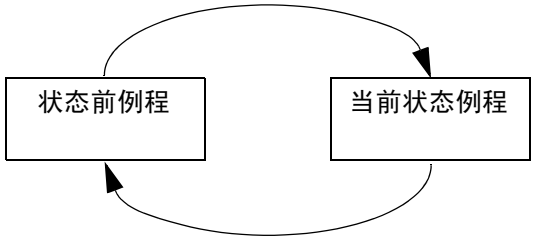
区分正常执行和异常

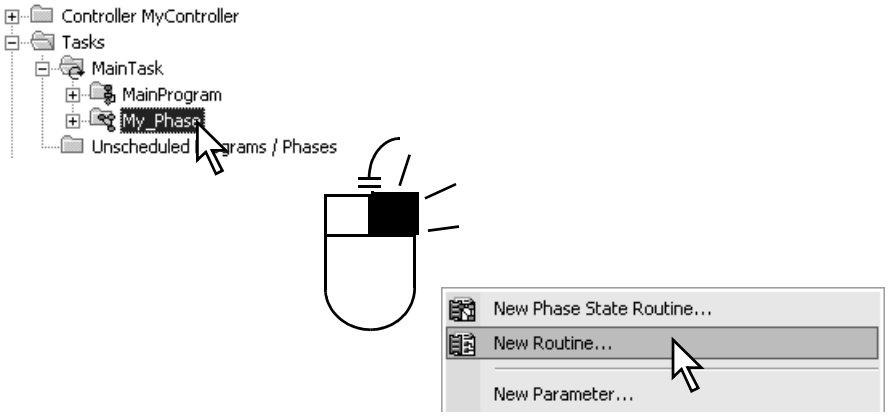
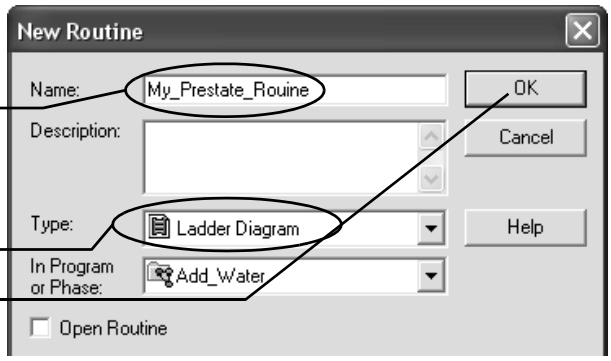
状态模型大大简化了区分设备的正常执行与任何异常 （故障、失效、非正常状况）的过程。

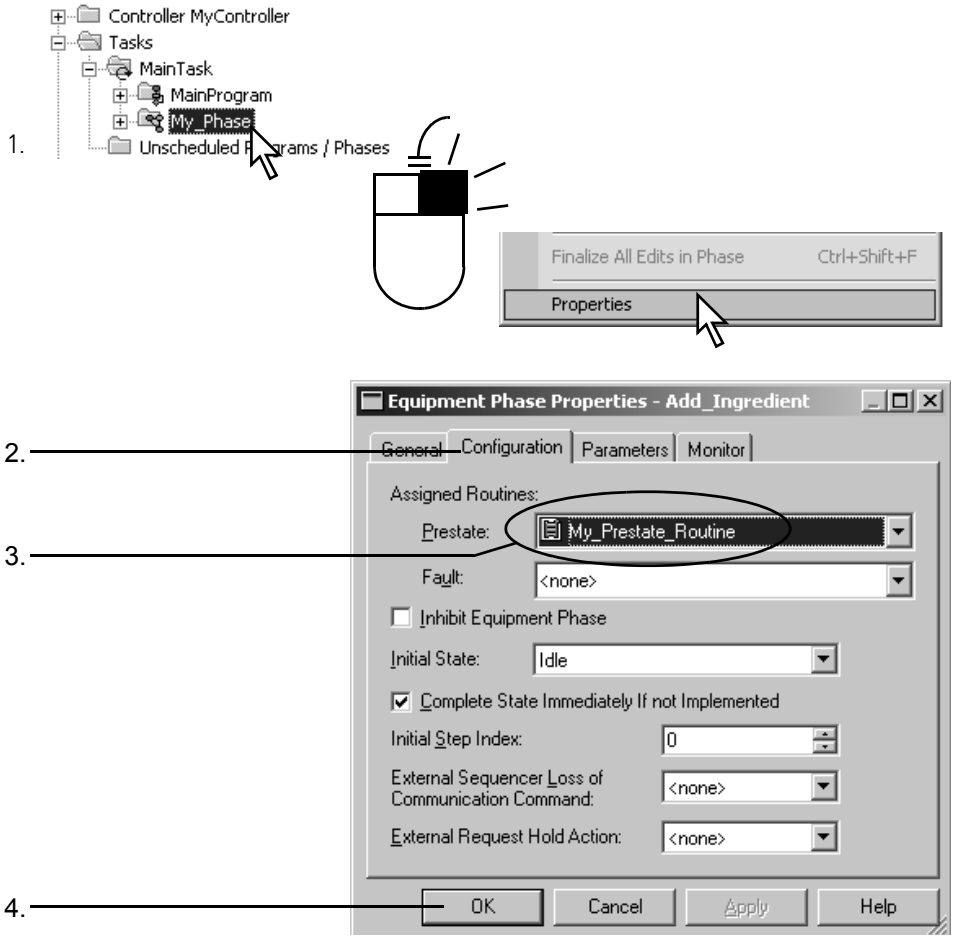
对设备的正常执行使用 resetting、running 和 stopping 状态。

使用 holding、 restarting 和 aborting 状态处理异常 （故障、失效、非正常状况）。



准则	说明
1. 使用状态前例程监视故障	 在始终要监视的情况 （诸如监视故障位）下使用状态前例程。状态前例程： • 始终运行 • 在 每次状态扫描前运行 • 即使在等待状态（idle、 held、 complete、 stopped 或 aborted）时也运行

准则	说明
2. 像创建程序例程一样创建状态前例程。状态前例程不是阶段状态例程。	1.  2. 3. 选择任意语言。 4. 

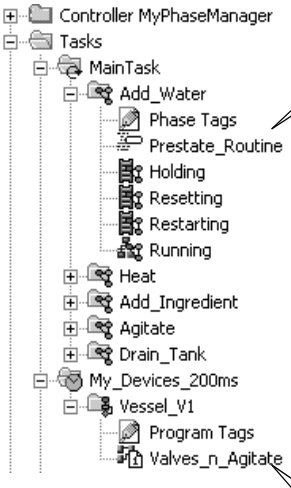
准则	说明
3. 指定状态前例程。	
4. 使用状态位将代码限制为某一特定状态	RSLogix 5000 软件自动为每个设备阶段设置一个标记。通过标记中的位可以得知设备阶段的状态。 • 标记范围为控制器。 • 标记使用 PHASE 数据类型。 • 对要限制为特定状态的代码使用标记位。 **示例** 假定设备阶段名称为 *My_Phase*。并且有一些只能在设备阶段处于 running 状态时运行的代码。在这种情况下，检查 on (1) 的 *My_Phase.Running* 位： `If My_Phase.Running then…` 有关更多信息，请参见附录 B。

准则	说明

5. 使用 PFL 指令发出故障信号	设备阶段失效 (PFL) 指令为设备阶段设置失效代码。使用此代码可发出信号指示出现特定失效，如特定设备故障。
	• PFL 指令为设备阶段向失效成员写入代码。
	• 要查看设备阶段的失效代码，请查看 *phase_name.Failure* 标记。
	• 在发生以下情况中的任意一种之前，失效代码一直存在：
	  • PFL 指令将失效代码设置为更大的数目。
	  • 设备阶段发生 resetting 状态 ？idle 状态的转换。
	  • PCLF 指令清除失效代码。
	  • RSBizWare Batch 软件清除失效代码。
	有关更多信息，请参见第 A-17 页。

6. 使用 PCLF 指令清除失效代码	设备阶段清除失效 (PCLF) 指令清除设备阶段的失效代码。 • CLR 指令、 MOV 指令或赋值 (:=) 《不会》更改设备阶段的失效代码。 • 测试 PCLF 指令时，确保 RSLogix 5000 软件 《不》拥有设备阶段。PCLF 指令在 RSLogix 5000 软件指令拥有设备阶段时 《不》起作用。 有关更多信息，请参见第 A-21 页。

示例 1：向罐中注水



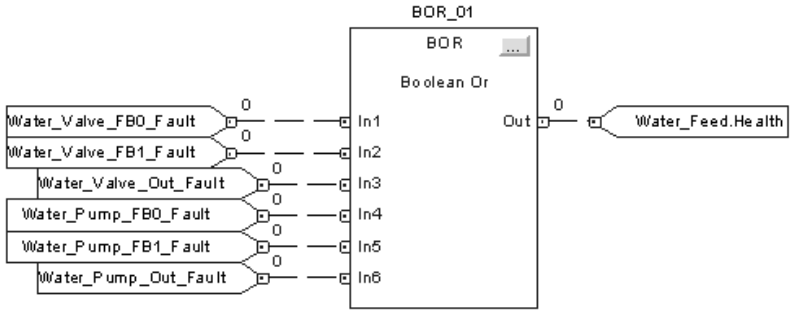
状态前例程在设备阶段处于 running 状态 (*Add_Water.Running* = 1) 时监视设备故障。如果 *Water_Feed.Health* = 1，则发生了故障。发生故障时，设备阶段设置失效代码 202。

```
If Add_Water.Running And Water_Feed.Health Then

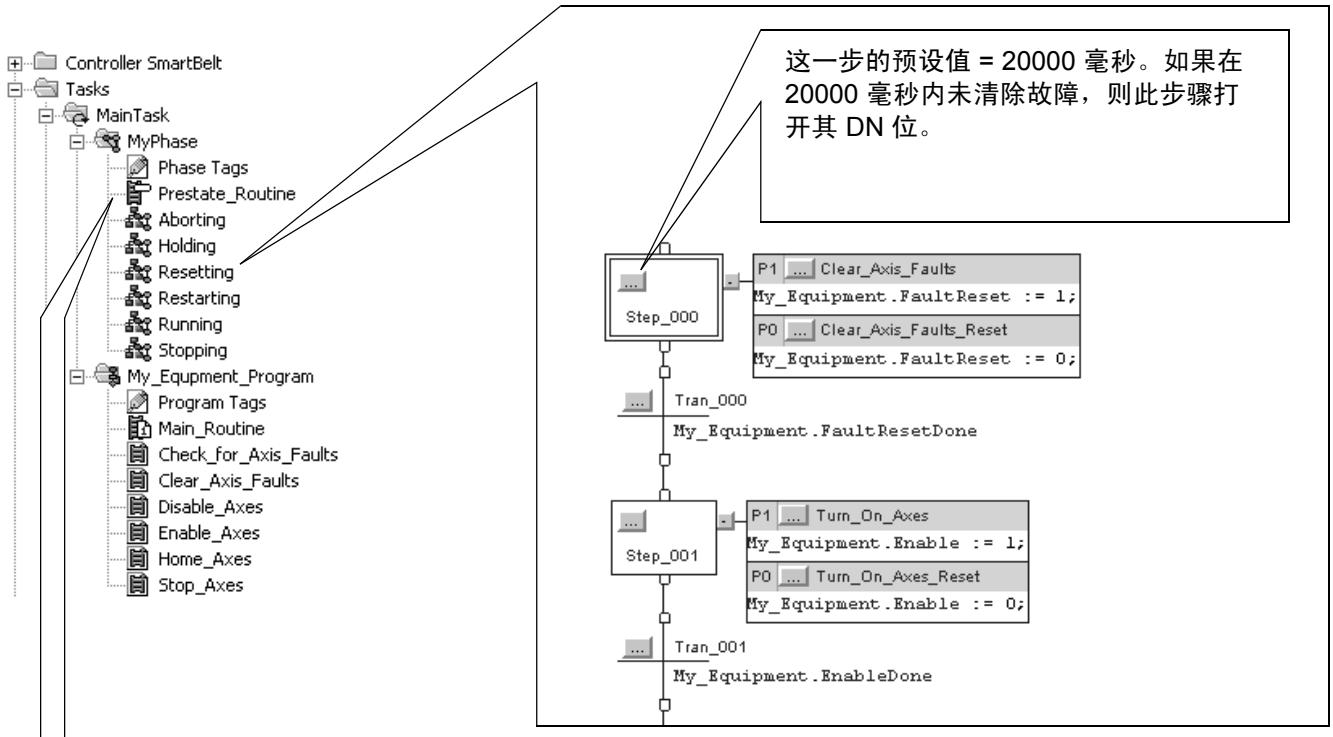
     PFL(202);

End_If;
```

设备程序监视阀门、泵及其反馈装置的故障位。任一设备发生故障时，设备程序将打开 *Water_Feed.Health* 位。

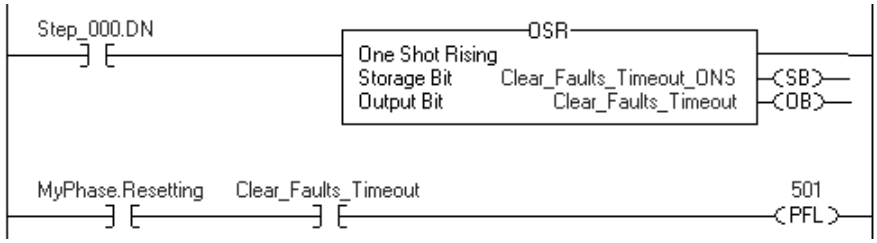


示例 2：智能传送带



如果 Step_000.DN = on，则发生了超时。超时发生时， OSR 指令打开 Clear_Faults_Timeout 位以进行一次扫描。

如果 MyPhase 处于 resetting 状态且打开了 Clear_Faults_Timeout，则 PFL 指令发出信号，指示失效。 PFL 指令设置失效代码 = 501。

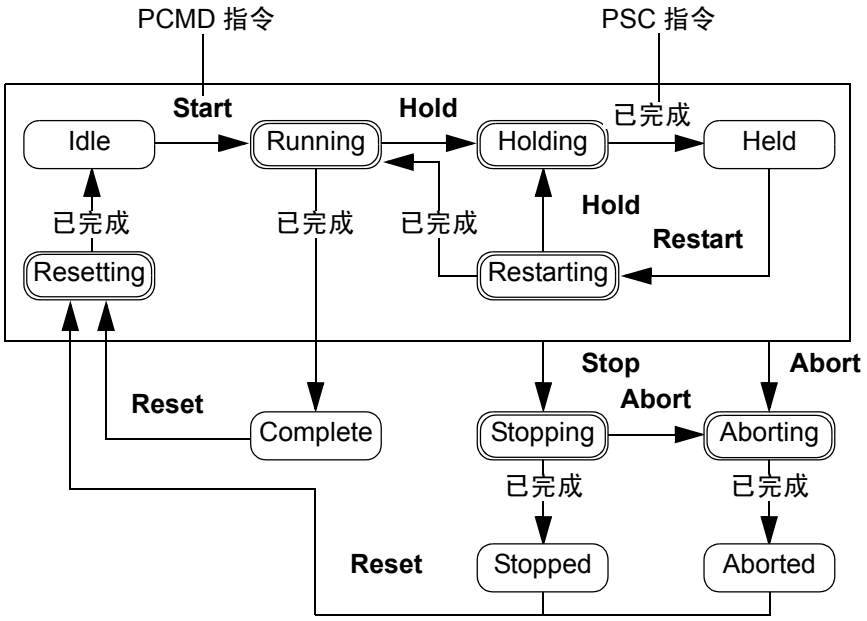


使用 PCMD 指令转换为其他状态

要开始某一动作状态，通常必须向设备阶段发出一个命令。该命令通知设备阶段及其设备开始执行某些操作或执行其他操作。使用设备阶段命令 (PCMD) 指令向某一设备阶段发出一个命令。

可选：您也可以使用 RSBizWare Batch 软件替代 PCMD 指令来触发转换。

使用状态模型查看哪些转换需要 PCMD 指令。



转换类型	说明	指令
命令	命令通知设备开始执行某些操作或执行其他操作。例如，操作员按下开始按钮开始生产，按下停止按钮关机。 PhaseManager 使用以下命令： reset　stop　restart start　hold　abort	PCMD 使用设备阶段命令 (PCMD) 指令发出命令。或使用 RSLogix 5000 软件。 有关更多信息，请参见第 A-8 页。
已完成	设备在完成其工作时进入 waiting 状态。编写代码以在设备完成其工作时发出信号。 waiting 状态指示设备已完成其工作。 **异常**：重启完成后从 restarting 状态进入 running 状态。	PSC 使用阶段状态完成 (PSC) 指令在状态完成后发出信号。有关更多信息，请参见第 A-5 页。

<table>
<tr><th>准则</th><th>说明</th></tr>
<tr><td>1. PCMD 指令会使转换立即进行。</td><td>PCMD 指令使设备阶段进入所要求的状态。设备阶段在完成当前扫描后立即更改状态。甚至会在当前状态未完成时就更改状态。</td></tr>
<tr><td>2. 查看是否需要重置所离开的状态。</td><td>要离开某一动作状态 （如 running、 holding）吗？
是 — 考虑重置所离开的状态的代码。
否 — 可能不需要重置任何代码。
设备阶段在进入不同状态时停止运行当前状态的代码。这使得输出保留为其最近的值，除非新状态控制这些输出。这还会使 SFC 停留在设备阶段更改状态时所在的步骤。

示例 1：不需要重置

假定设备阶段处于 idle 状态。在这种情况下，它不运行任何状态代码。因此在进入如 running、 stopping 等不同状态时可能不需要重置任何状态。

示例 2：不需要重置

假定设备阶段处于 running 状态而您进入 holding 状态。再次回到 running 状态时，您可能会希望回到离开时的位置。在这种情况下，可能不需要重置 running 状态的代码。

示例 3：必须重置

假定在设备阶段在 resetting 状态进行了一半时，您给出 stop 命令。并假定您要在回到该状态时运行整个 resetting 序列。在这种情况下，您可能需要重置 resetting 状态的代码。如果 resetting 状态使用 SFC，则使用该 SFR 指令将其重置到第一步。</td></tr>
<tr><td>3. 使用 SFR 指令重置状态例程的 SFC。</td><td>可以使用 SFC Reset (SFR) 指令重置 SFC。在某些情况下，需要从多个其他状态例程重置 SFC：<table><tr><th>重置此状态的 SFC：</th><th>在此状态例程中放置一条 SFR 指令：</th></tr><tr><td>Running</td><td>Resetting</td></tr><tr><td>Holding</td><td>Holding 讶 — SFC 在最后一步重置其本身。</td></tr><tr><td>Restarting</td><td>在这些例程中同时重置 restarting 例程：
Holding— 谕瓠 … restarting 前返回 holding 时。
Restarting 讶 — SFC 在最后一步重置其本身。</td></tr></table></td></tr>
<tr><td>4. 使用 PCMD 指令进入允许的下一个状态。</td><td>PhaseManager 确保设备阶段符合状态模型。因此设备阶段仅从其当前状态进入特定状态。

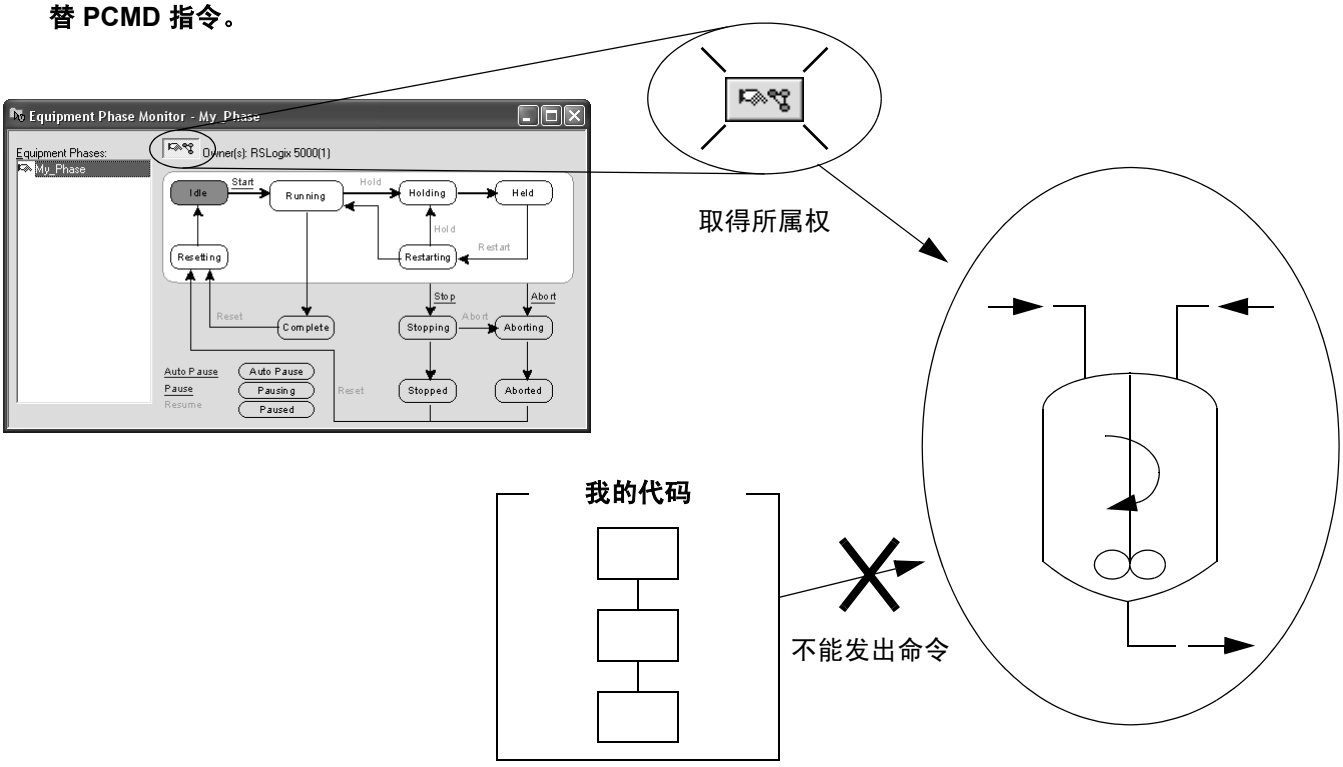
示例 1：允许转换

假定设备阶段处于 running 状态，您向它发出一个 hold 命令。在这种情况下，因为允许转换，所以设备阶段进入 holding 状态。

示例 2：不允许转换

假定设备阶段处于 running 状态，您向它发出一个 reset 命令。在这种情况下，设备阶段保留为 running 状态。要进入 resetting 状态，首先需要停止或中止设备阶段。</td></tr>
</table>

准则	说明
5. 查看是否必须使用 POVR 指令代替 PCMD 指令。	



A. 您要给出 hold、 stop 或 abort 命令吗？

- 否 — 使用 PCMD 指令。
- 是 — 转到步骤 B。

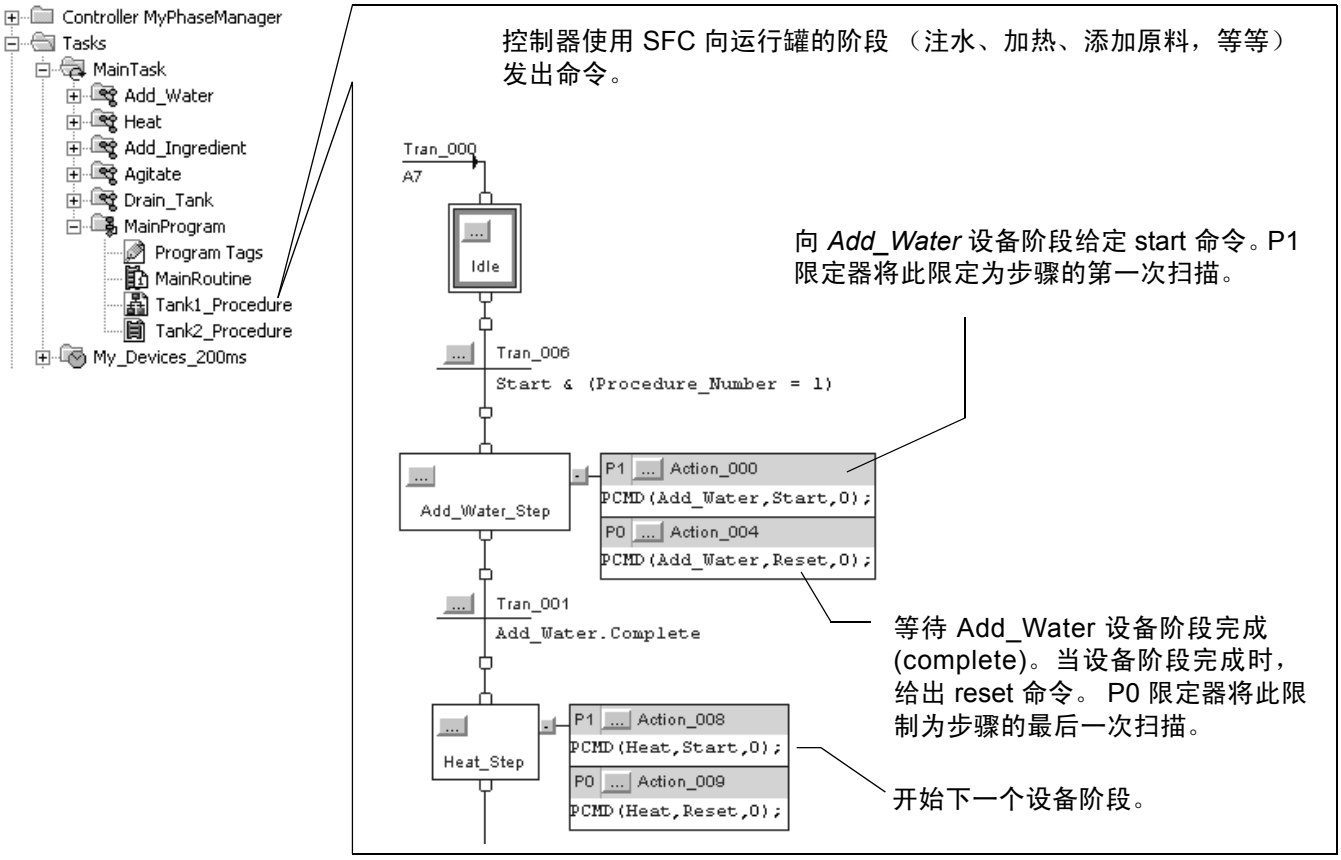
B. 即使在能够通过 RSLogix 5000 软件手动控制设备阶段时命令也必须起作用吗？

- 是 — 改为使用 POVR 指令。请参见第 A-13 页。
- 否 — 转到步骤 C。

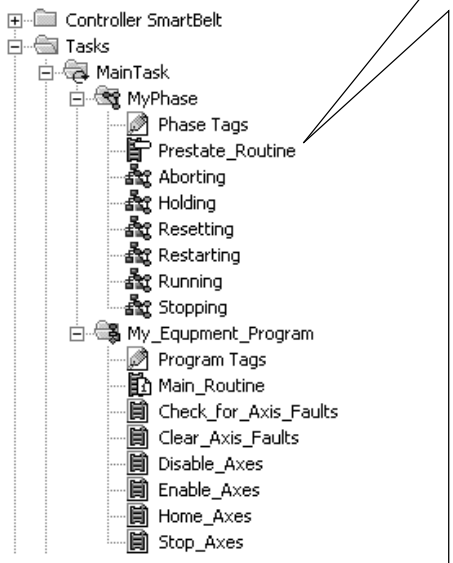
C. 即使在 RSBizWare Batch 软件或其他软件拥有设备阶段时命令也必须起作用吗？

- 是 — 改为使用 POVR 指令。请参见第 A-13 页。
- 否 — 使用 PCMD 指令。

示例 1：罐


Controller MyPhaseManager
Tasks
MainTask
Add_Water
Heat
Add_Ingredient
Agitate
Drain_Tank
MainProgram
Program Tags
MainRoutine
Tank1_Procedure
Tank2_Procedure
My_Devices_200ms
控制器使用 SFC 向运行罐的阶段 （注水、加热、添加原料，等等）发出命令。
Tran_000
A7
Idle
向 Add_Water 设备阶段给定 start 命令。P1 限定器将此限定为步骤的第一次扫描。
Tran_006
Start & (Procedure_Number = 1)
Add_Water_Step
P1 Action_000
PCMD(Add_Water,Start,0);
P0 Action_004
PCMD(Add_Water,Reset,0);
Tran_001
Add_Water.Complete
等待 Add_Water 设备阶段完成 (complete)。当设备阶段完成时，给出 reset 命令。 P0 限定器将此限制为步骤的最后一次扫描。
Heat_Step
P1 Action_008
PCMD(Heat,Start,0);
P0 Action_009
PCMD(Heat,Reset,0);
开始下一个设备阶段。

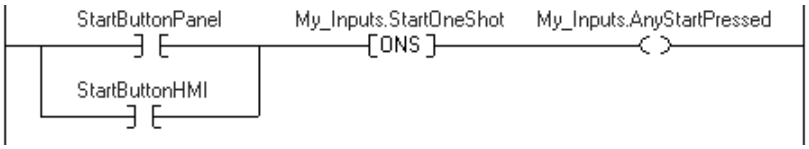
示例 2：智能传送带



如果操作员按下机器或 HMI 上的开始按钮，那么

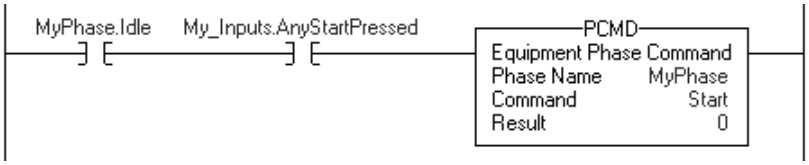
My_Inputs.AnyStartPressed = on，并进行 1 次扫描。

ONS 指令确保 *My_Inputs.AnyStartPressed* 仅在开始按钮进行关 ⇒ 开变化时打开。

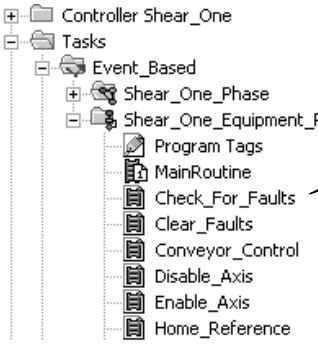


如果设备阶段处于 idle 状态且 *My_Inputs.AnyStartPressed* = on，则

PCMD 指令向 *MyPhase* 给出 start 命令。



示例 3：阻塞检测

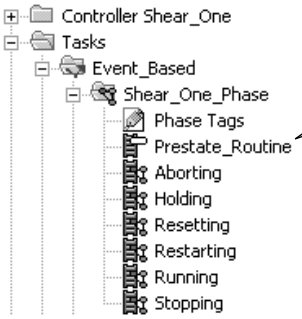


设备程序监视以下故障：

- 故障轴
- 阻塞的材料

如果发生故障，则

Local_Interface.Equipment_Faults_Cleared = 0。此标记为控制器范围的标记 *Shear_1* 的别名。



设备阶段的状态前例程监视设备程序以报告故障。

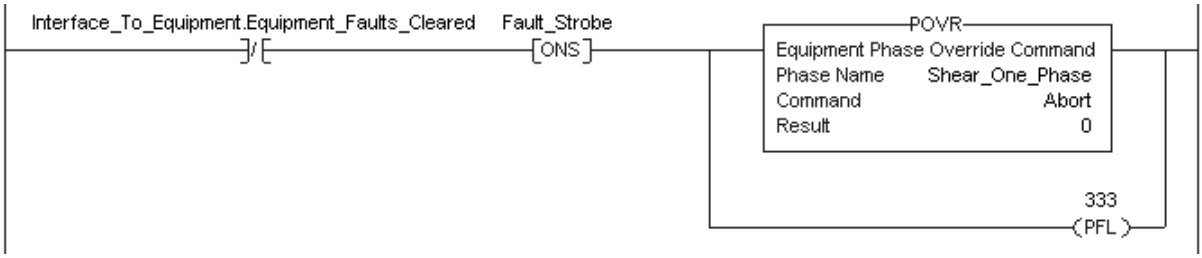
- 如果 *Interface_To_Equipment.Equipment_Faults_Cleared* = 0，则存在故障。
- *Interface_To_Equipment* 和 *Local_Interface* 都是 *Shear_1* 的别名，因此它们的值相同。

如果存在故障，则

向 *Shear_One_Phase* 设备阶段给出 abort 命令。 POVR 指令确保即使有人可以通过 RSLogix 5000 软件手动控制设备阶段该命令也起作用。

PFL 指令设置失效代码 *Shear_One_Phase* = 333。

Fault_Strobe 在一次扫描中执行这些动作。



使用 PSC 指令在状态完成时给出信号

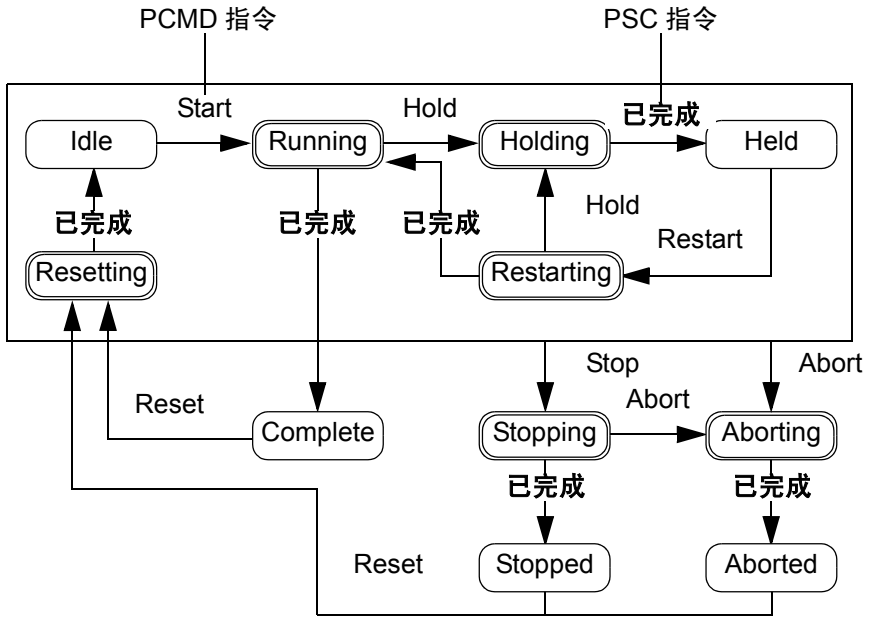
要离开某一动作状态，通常是发出信号指示该状态已完成其必需执行的任务。在状态完成时使用阶段状态完成 (PSC) 指令发出信号。

重要事项

PSC 指令不会 停止例程的当前扫描。

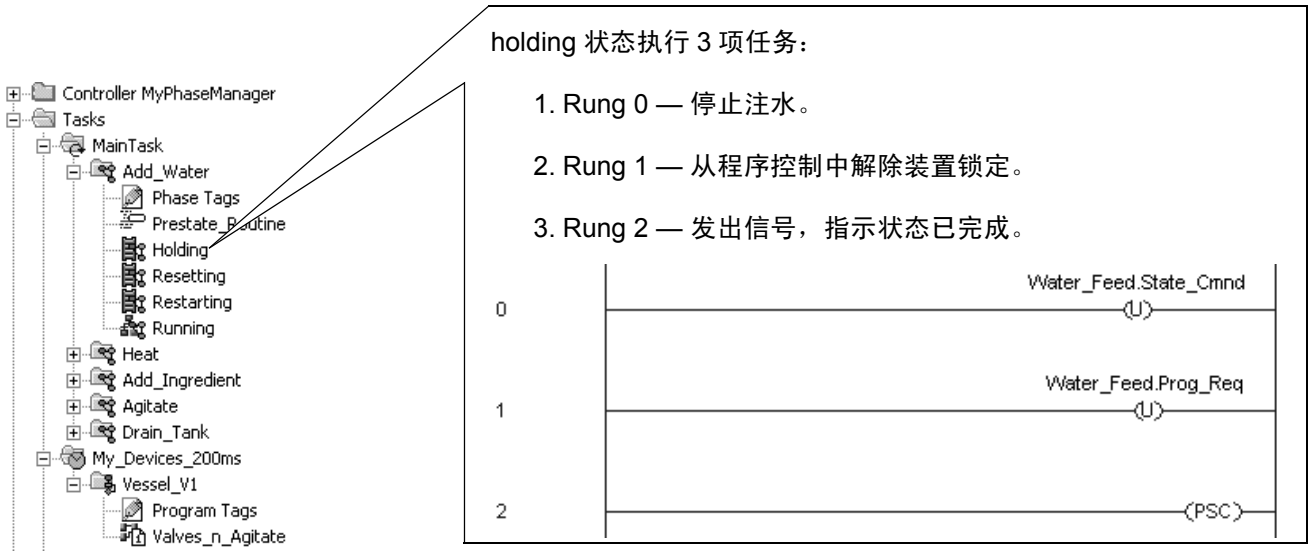
PSC 指令执行时，控制器扫描例程的其余部分，然后将设备阶段转换为下一个状态。 PSC 指令不会 · 终止例程的执行。

使用状态模型查看哪些转换需要 PSC 指令。

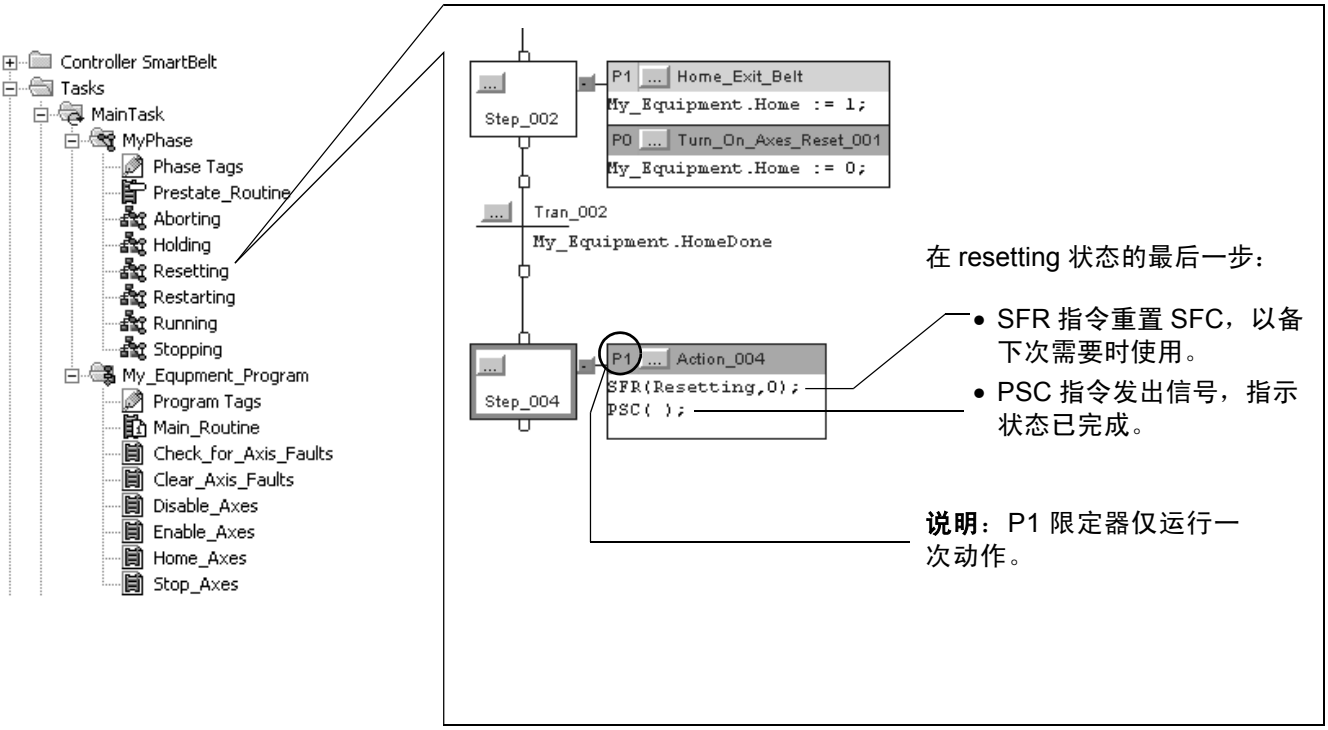


转换类型	说明	指令
命令	命令通知设备开始执行某些操作或执行其他操作。例如，操作员按下开始按钮开始生产，按下停止按钮关机。 PhaseManager 使用以下命令： reset stop restart start hold abort	PCMD 使用设备阶段命令 (PCMD) 指令给出命令。或使用 RSLogix 5000 软件。
已完成	设备在完成正在执行的工作后进入 waiting 状态。编写代码以在设备完成其工作时发出信号。waiting 状态指示设备已完成其工作。 **异常**：重启完成后从 restarting 状态进入 running 状态。	PSC 在状态完成时使用阶段状态完成 (PSC) 指令发出信号。有关更多信息，请参见第 A-5 页。

示例 1：向罐中注水



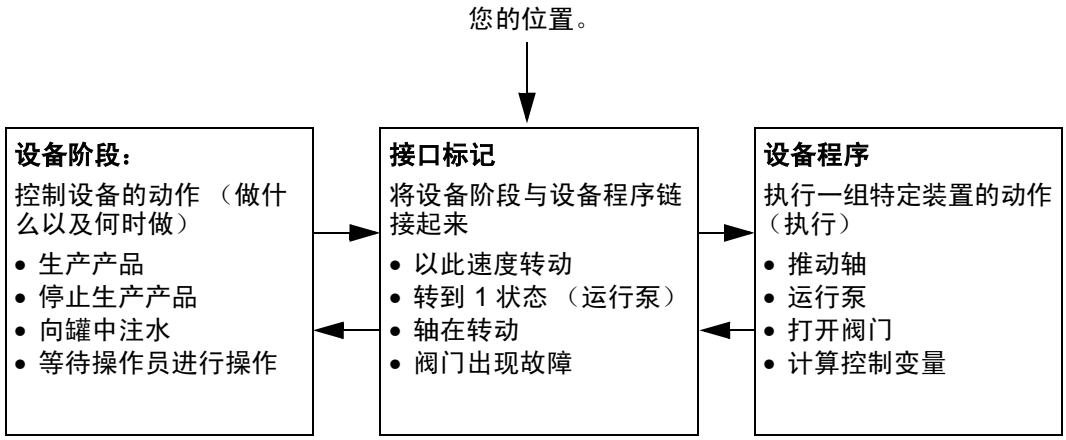
示例 2：智能传送带



创建设备接口标记

设备接口标记将设备阶段与设备程序链接起来。

- 设备阶段使用该标记配置和命令设备程序。
- 设备程序使用该标记报告其状态或状况。



准则	说明：
1. 列出设备阶段必须向设备程序提供或从设备程序获取的值。	将这些值作为设备程序的面板。设备阶段使用这些值控制并监视设备程序。请*勿*包括 I/O 数据。 **设备程序输入**：• 模式请求 • 设置点 • 诸如 on、 off、 start、 stop、 reset 等命令 • 权限 • 重写 **设备程序输出**：• 模式状态 • 控制值 • 已完成或完成 • 警告 • 故障 • 状况指示 • 总值或累计值
2. 创建用户定义数据类型	使用用户定义数据类型可以制作数据模板。使用它可以将相关数据组合为一个数据类型。然后可以使用该数据类型在同一数据布局中进行标记。 如果有多个设备阶段，则可以设计数据类型，以便简化多个设备阶段的使用。考虑以下几点： • 包括一定范围的数据以使数据类型更通用。 • 尽可能使用通用名称。 **示例**：名称 *State_Cmnd* 可用于以两种状态（诸如 on/off、running/not running、 pumping/not pumping 等）运行的所有设备。它比诸如 *Open* 或 *Close* 之类的名称更易于再次使用。这些名称适用于阀门，但不适用于泵或电机。

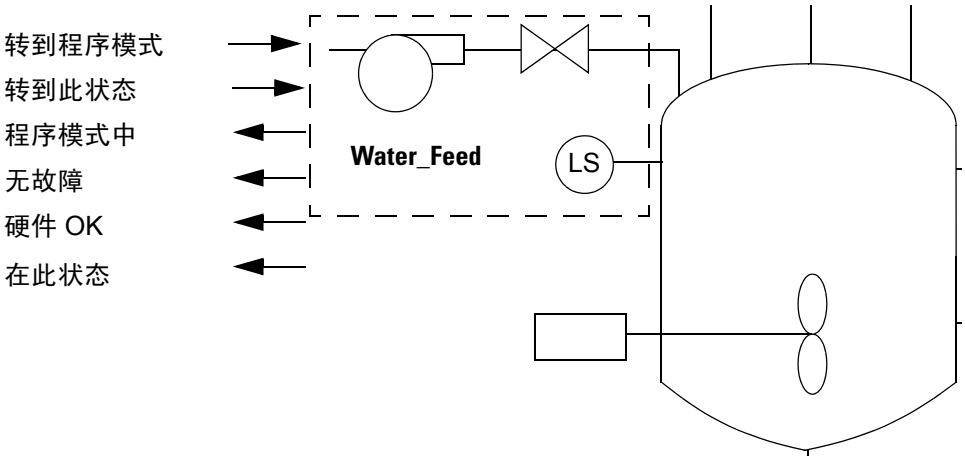
准则	说明：
3. 为每个设备阶段创建一个标记	为每个设备阶段的接口数据创建标记。 • 为每个设备阶段创建一个标记。 • 使用准则 2 中的数据类型。 • 在控制器范围创建该标记。设备阶段和设备程序都必须能够获得该标记。 • 考虑使用别名标记。请参见第 第 3-31 页 页的 《使用别名标记》。

《有关更多信息…》

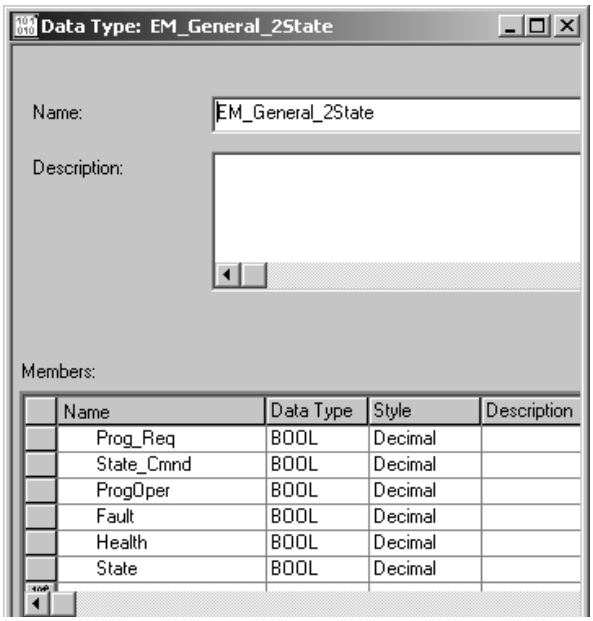
有关如下信息：	请参见出版物：
有关下面内容的指南和注意事项： • 用户定义数据类型 • 别名标记	*Logix5000 Controllers Design Considerationsl*（Logix5000 控制器设计注意事项）， 1756-RM094
执行下面任务的逐步过程： • 创建用户定义数据类型 • 指定别名标记	*Logix5000 Controllers Common Procedures*（Logix5000 控制器通用编程步骤），出版号 1756-PM001

示例 1：向罐中注水

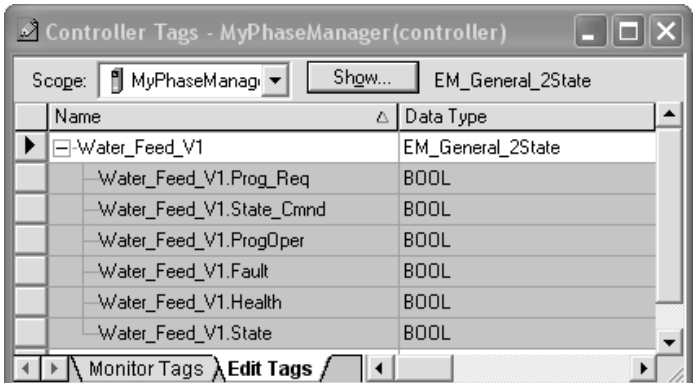
1. 设备阶段和设备程序共享此数据。



2. 用户定义数据类型创建一个数据模板。



3. 标记存储由设备阶段和设备程序共享的数据。该标记使用来自步骤 2 的用户定义数据类型。



示例 2：智能传送带

1. 设备阶段和设备程序共享此数据。

设备程序接口

命令		状况或状态	
Enable	Abort	FaultScroll	EnableCyclingDone
Disable	FaultReset	Faulted	DisableCyclingDone
Home	Stop	EnableDone	AbortingDone
ActivateRun	ArmRegistration	DisableDone	FaultResetDone
EnableProduct		HomeDone	StoppingDone
DisableProduct		ActivateRunDone	Selected
EnableCycling		EnableProductDone	RegistrationArmed
DisableCycling		DisableProductDone	

2. 单个用户定义数据类型保存每个轴的数据。

轴接口

命令		状况或状态		
Enable	Abort	State	NoMotion	MoveActive
Disable	Stop	On	Homed	HomeDone
Home	ActivateRun	Ok	AxisSelected	RunDone
AutoRun		Auto	GearActive	
ResetFaults		Jogging	CamActive	

3. 每个轴都有一个接口标记，整个机器也有一个接口标记。

一个标记存储由设备阶段和设备程序共享的数据。其他标记分别存储各个轴的数据。

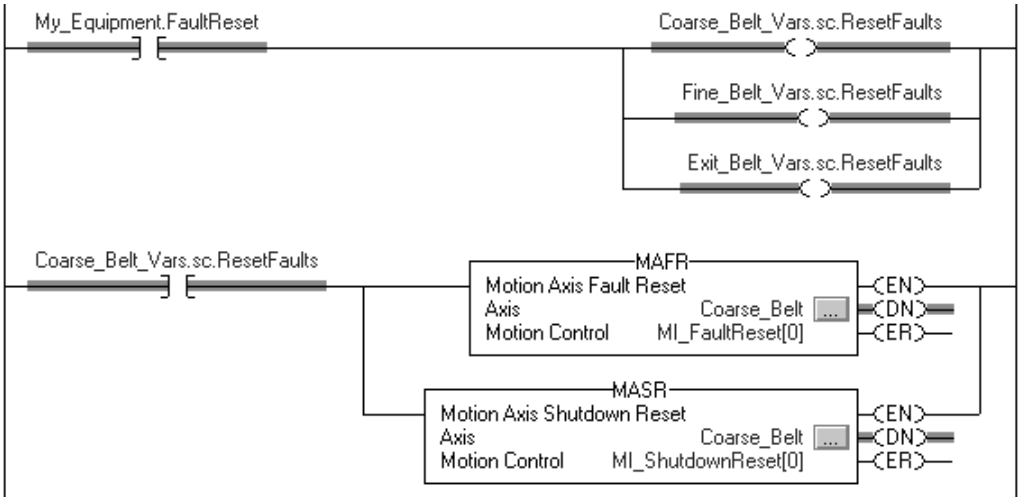


示例 2：智能传送带（续）

4. 设备程序从设备阶段获取命令，然后将其传递给各轴。

设备程序的例程

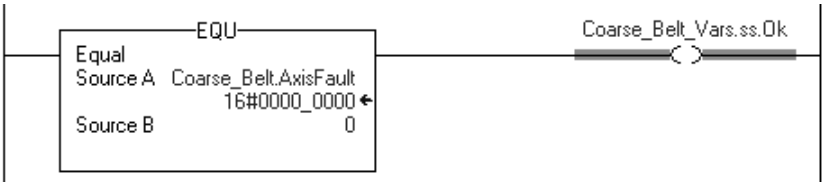
此标记	是下面两者间的接口
My_Equipment	设备阶段和设备程序
Coarse_Belt_Vars	设备程序和轴



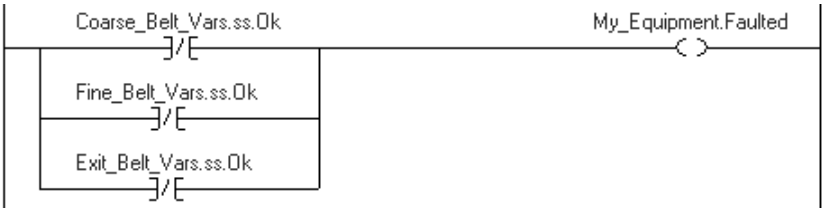
5. 设备程序收集各轴的故障状态，然后将其传递回设备阶段。

设备程序的例程

设备程序检查各轴的故障代码。如果轴没有故障，则打开该轴的 OK 位。



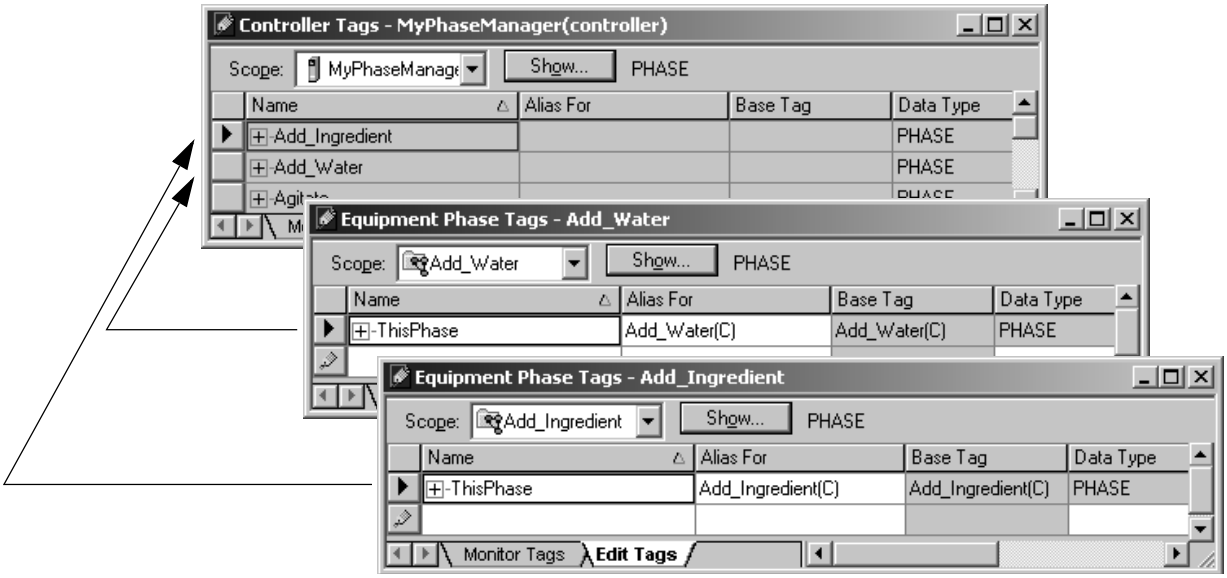
设备程序收集各轴的 OK 状态。如果各轴的 OK 位均为 on，则 *My_Equipment.Faulted* = off （无故障）。



使用别名标记

程序范围的标记和阶段范围的标记简化了代码的重复使用过程。在控制器范围为标记创建标记别名。再次使用设备阶段 （如复制 / 粘贴）时，只需将阶段范围的标记指向控制器范围的新标记即可。这简化了在代码中确定地址的过程。

示例



控制器自动为每个设备阶段创建一个标记。此标记是控制器范围的（控制器标记）。假定您计划对罐的不同部件再次使用某一设备阶段。在这种情况下：

1. 为第一个设备阶段创建一个别名标记。使该标记成为阶段范围的标记，然后将其指向该设备阶段的控制器标记。
2. 在该设备阶段 (ThisPhase) 的代码中使用该别名标记。
3. 制作该设备阶段的一个副本。
4. 将该副本的别名标记指向其控制器标记。

《有关更多信息…》

有关如下信息：	请参见出版物：
别名标记指南和注意事项	*Logix5000 Controllers Design Considerationsl*（Logix5000 控制器设计注意事项）， 1756-RM094
指定别名标记的步骤	*Logix5000 Controllers Common Procedures*（Logix5000 控制器通用编程步骤），出版号 1756-PM001

说明：

设备阶段指令（PSC、PCMD、POVR、PFL、PCLF、PXRQ、PRNP、PPD、PATT、PDET）

本附录目的

本附录以如下格式介绍每个设备阶段指令。

本节：	提供此类信息：
指令名称	标识指令
操作数	列出指令的所有操作数
指令结构	列出指令的控制状态位和值 （如果有）
说明	说明指令的用法定义启用和禁用指令之间的所有差异 （如果需要）
算术状态标志	定义指令是否影响算术状态标志
故障条件	定义指令是否产生轻微或严重故障。如果是，则定义故障类型和代码
执行	定义指令的具体执行方式
示例	至少用每种可用编程语言提供一个编程示例包括每个示例的说明

下面的图标有助于识别语言特定的信息：

图标：	指示编程语言：
	梯形图
	结构化文本

约定和相关术语

设置和清除

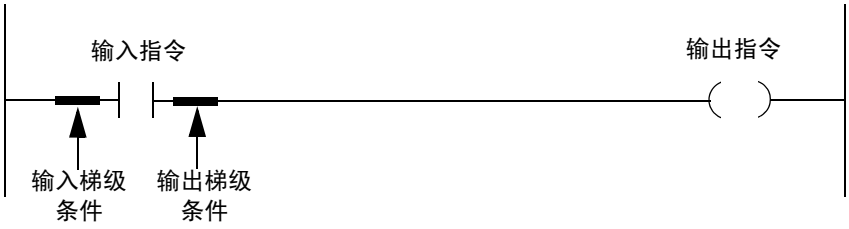
本手册使用 “设置”和 “清除”来定义位的状态 （布尔值）和值（非布尔值）：

术语：	含义：
设置	位被设置为 1 (ON) 值被设置为任何非零数字
清除	位被清除为 0 (OFF) 值中的所有位都被清除为 0

如果操作数或参数支持多个数据类型，则**粗体**数据类型表示最佳数据类型。如果指令的所有操作数都使用相同的最佳数据类型 （通常为 DINT 或 REAL），则指令的执行速度更快，需要的内存更少。

梯形图梯级条件

控制器根据指令前的梯级条件 （输入梯级条件）计算梯形指令。根据输入梯级条件和指令，控制器设置指令后梯级条件 （输出梯级条件），进而影响后续指令。



如果输入指令的输入梯级条件为 true，则控制器计算指令并根据指令的结果设置输出梯级条件。如果指令计算结果为 true，则输出梯级条件为 true ；如果指令计算结果为 false，则输出梯级条件为 false。

例程预扫描

控制器还预扫描例程。预扫描是在控制器中对所有例程进行的特殊扫描。在预扫描过程中，控制器执行以下操作：

- 扫描所有主例程
- 扫描设备阶段的所有状态例程
- 扫描设备阶段的所有状态前例程
- 扫描一次程序和设备阶段的所有子例程

  控制器预扫描某个子例程后，则不 会在预扫描过程中再次扫描该子例程。
- 扫描由梯形图例程的 FOR 指令调用的所有例程
- 忽略可能跳过指令执行的跳转
- 以预扫描模式执行所有指令

  有关特定指令在预扫描过程中如何操作的详细信息，请参见相应指令的 《执行 》一节。
- 将所有非保留赋值 ([:=]) 重置为 0
- 不 更新输入值
- 不 写入输出值

以下情况会导致预扫描：

- 从程序模式切换到运行模式
- 从通电状态自动进入运行模式

在以下情况下，程序或设备阶段不进行预扫描：

- 控制器在运行时程序或设备阶段进行了调度。
- 控制器进入运行模式时程序或设备阶段未进行调度。

选择设备阶段指令

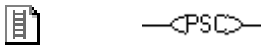
如果需要：	使用指令：	提供支持的语言：	参见页：
发出信号通知设备阶段，状态例程已完成因而进入下一状态	阶段状态完成 (PSC)	梯形图 结构化文本	A-5
更改设备阶段的状态或子状态	设备阶段命令 (PCMD)	梯形图 结构化文本	A-8
无论所属权如何，向设备阶段提供 hold、stop 或 abort 命令	设备阶段重写命令 (POVR)	梯形图 结构化文本	A-13
发出信号指示设备阶段失效	设备阶段失效 (PFL)	梯形图 结构化文本	A-17
清除设备阶段的失效代码	设备阶段清除失效 (PCLF)	梯形图 结构化文本	A-21
开始与 RSBizWare Batch 软件的通讯	设备阶段外部请求 (PXRQ)	梯形图 结构化文本	A-23
清除设备阶段的 NewInputParameters 位	设备阶段新参数 (PRNP)	梯形图 结构化文本	A-34
在设备阶段的逻辑中设置断点	设备阶段暂停 (PPD)	梯形图 结构化文本	A-37
获取设备阶段的所属权，以便实现下面任一操作： • 防止其他程序或 RSBizWare Batch 软件命令设备阶段 • 确认其他程序或 RSBizWare Batch 软件尚 《未》 拥有设备阶段	附加到设备阶段 (PATT)	梯形图 结构化文本	A-42
释放对设备阶段的所属权	脱离设备阶段 (PDET)	梯形图 结构化文本	A-47

阶段状态完成 (PSC)

使用 PSC 指令向设备阶段发出信号，指示状态例程已完成因而进入下一状态。

操作数：

梯形图



无

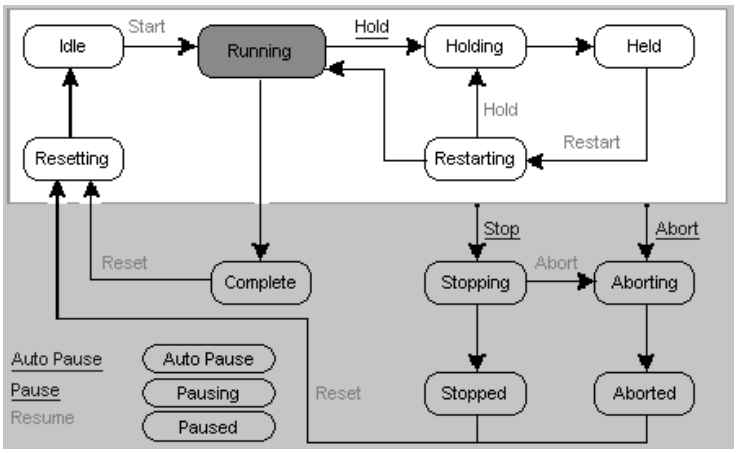
结构化文本

```
PSC( );
```

无

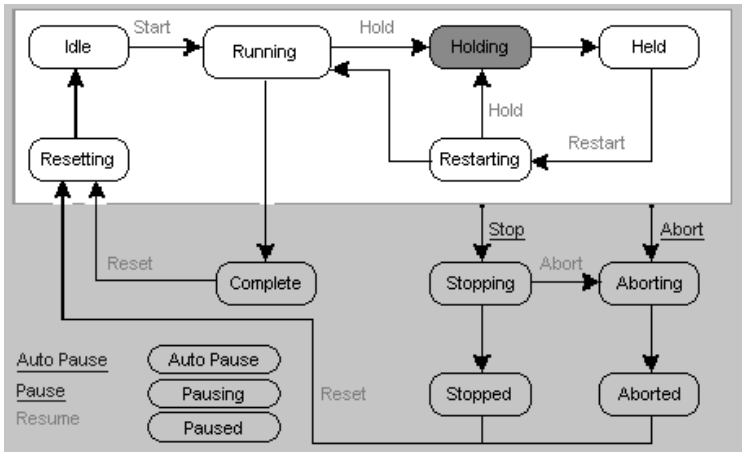
在指令助记码之后必须输入括号 ( )，即使没有操作数也是如此。

说明： PSC 指令发出信号，指示阶段状态例程已完成。



在运行的状态例程中，使用 PSC 指令使设备阶段转换为 complete 状态。

PSC 指令的使用准则

准则：	说明：
❑ **在添加到设备阶段的 《每个》 阶段状态例程中使用 PSC 指令。**	如果不使用 PSC 指令，则设备阶段保持其状态，《不》进入下一状态。 • 将 PSC 指令放置为阶段状态例程中的最后步骤。 • 状态完成时，执行 PSC 指令。
	 在 holding 状态例程中，使用 PSC 指令使设备阶段进入 Held 状态。
❑ **请注意：PSC 指令 《不》 会停止当前的例程扫描。**	PSC 指令执行时，控制器扫描例程的其余部分，然后使设备阶段转换为下一状态。 PSC 指令 《不》 会终止例程的执行。
❑ **《不要》 在预状态例程中使用 PSC 指令。**	仅将 PSC 指令用于通知从一个状态到另一个状态的转换。

算术状态标志： 不受影响

故障条件： 无

执行：

条件：	梯形图操作：	结构化文本操作：
预扫描	输出梯级条件设置为 false。	不采取任何操作。
输入梯级条件为 false	输出梯级条件设置为 false。	不可用
输入梯级条件为 true	• 指令执行。 • 输出梯级条件设置为 true。	不可用
结构化文本的扫描	不可用	在结构化文本中，指令在每次扫描时都会执行。要限制指令的扫描，请使用 SFC 操作的限定符和 / 或结构化文本结构。
指令执行	指令发出信号，指示状态已完成。	指令发出信号，指示状态已完成。
后扫描	输出梯级条件设置为 false。	不采取任何操作。

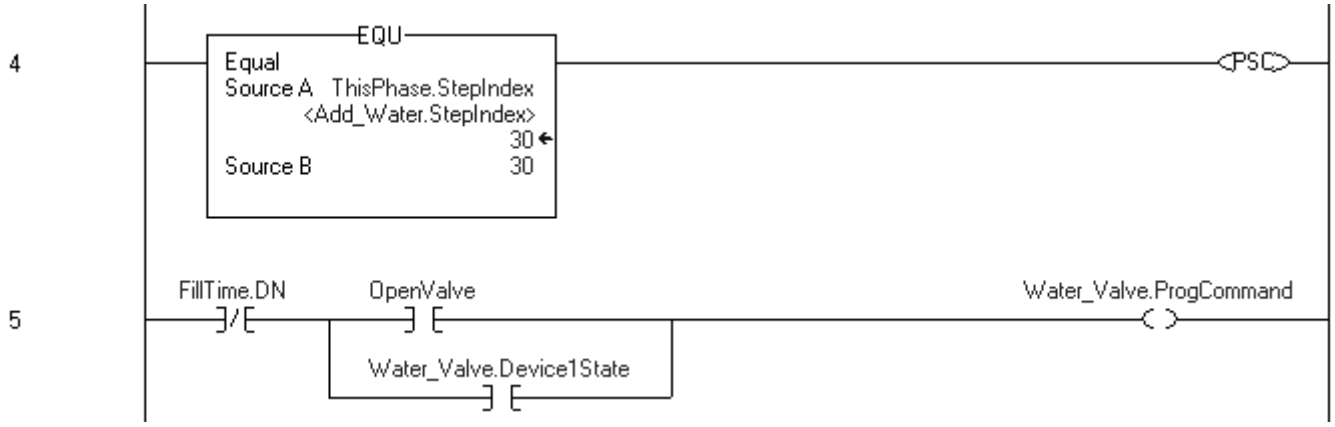
示例：

梯形图

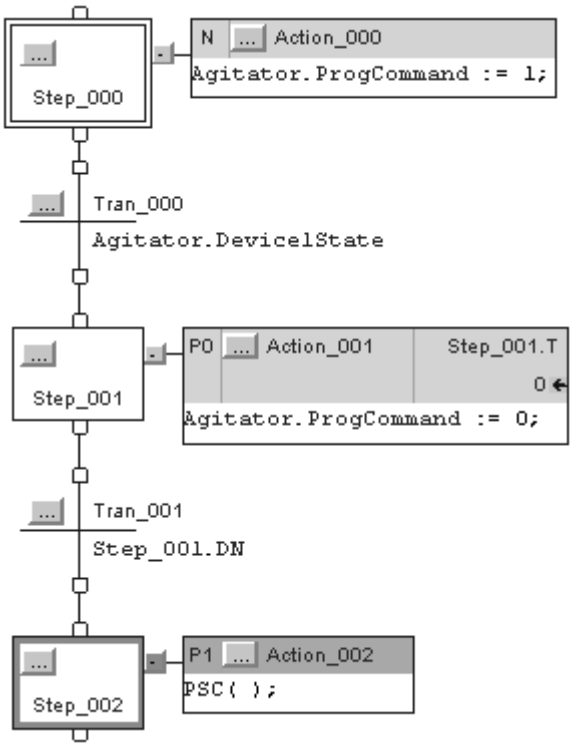
如果 ThisPhase.StepIndex = 30 （例程位于步骤 30），

则 PSC 指令发出信号，指示状态为 done （完成）。

控制器扫描例程 （梯级 5、梯级 6 等）其余部分之后，设备阶段进入下一状态。



结构化文本



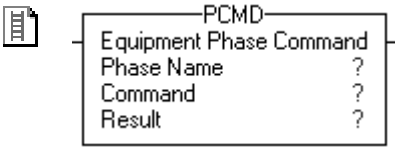
SFC 执行到 Step_002 时， PSC 指令发出信号，指示状态为 done （完成）。

设备阶段命令 (PCMD)

使用 PCMD 指令更改设备阶段的状态或子状态。

操作数：

梯形图



操作数：	类型：	格式：	说明：
阶段名称	阶段	设备阶段的名称	要更改为其他状态的设备阶段
命令	命令	命令的名称	要发送到设备阶段以更改其状态的命令有关可用的命令，请参见图 A.1。
结果	DINT	直接标记	要使指令返回其成功 / 失败的代码，请输入一个 DINT 标记，该标记存储结果代码。 否则，输入 0。

结构化文本

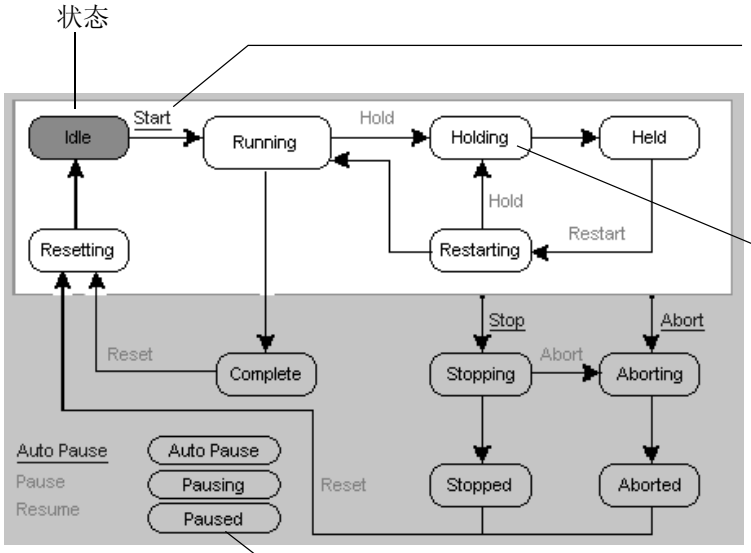


```
PCMD(PhaseName,
Command, Result);
```

操作数与梯形图 PCMD 指令的操作数一样。

说明： PCMD 指令使设备阶段转换为下一状态或子状态。

图 A.1 设备阶段的状态、子状态和命令



命令

某些状态需要命令才能进入下一状态。如果设备阶段处于 idle 状态， start 命令可将设备阶段转换为 running 状态。只要处于 running 状态，设备阶段就执行其 running 状态例程。

其他状态使用阶段状态完成 (PSC) 指令进入下一状态。如果设备阶段处于 holding 状态， PSC 命令可将设备阶段转换为 held 状态。处于 held 状态时，设备阶段需要使用 restart 命令才能进入 restarting 状态。

子状态

使用 auto pause、 pausing 和 paused 子状态测试和调试状态例程。这些子状态需要设备阶段暂停 (PPD) 指令在逻辑中创建断点。使用 auto pause、 pause 和 resume 命令可单步执行这些断点。

PCMD 指令的使用准则

准则：	说明：
❑ **将 PCMD 指令的执行限制于单次扫描。**	将 PCMD 指令的执行限制于单次扫描。每个命令都仅应用于一个或多个特定状态。只要设备阶段的状态发生了更改，该命令就 《不再》有效。要限制执行，请使用如下方法： • 在 P1 脉冲 （上升边缘）或 P0 脉冲 （下降边缘）操作中执行 PCMD 指令 • 在 PCMD 指令前放置一个 One Shot 指令 • 执行 PCMD 指令后前进到下一步骤
❑ **确定是否需要获取设备阶段的所属权。**	作为一个选项，程序可以拥有一个设备阶段。这样可防止其他程序或 RSBizWare Batch 软件也命令该设备阶段。

如果：	则：
将 RSBizWare Batch 软件也用于在此控制器中运行过程 （配方）	在使用 PCMD 指令之前，使用附加到设备阶段 (PATT) 指令获取设备阶段的所属权。请参见第 A-42 页。
使用多个程序命令同一设备阶段	
以上皆否	没有必要拥有设备阶段。

❑ **查看是否必须使用 POVR 指令而不是 PCMD 指令。**	A. 是否将提供 hold、 stop 或 abort 命令？ • 否 — 使用 PCMD 指令。 • 是 — 转到步骤 B。 B. 即使可以通过 RSLogix 5000 软件手动控制设备阶段，是否也必须使用该命令？ • 是 — 换用 POVR 指令。请参见第 A-13 页。 • 否 — 转到步骤 C。 C. 即使 RSBizWare Batch 软件或其他程序拥有该设备阶段，是否也必须使用该命令？ • 是 — 转为使用 POVR 指令。请参见第 A-13 页。 • 否 — 使用 PCMD 指令。 示例：假定设备检查堵塞材料。如果存在堵塞，您始终希望设备中止。这种情况下，使用 POVR 指令。这样，即使可以通过 RSLogix 5000 软件手动控制，设备也会中止。
❑ **确定是否需要结果代码。**	使用 Result 操作数获取显示 PCMD 指令成功 / 失败的代码。

如果：	则在 Result 操作数中输入：
希望出现所属权冲突或其他可能的错误	DINT 标记，其中可以存储指令执行结果的代码。
《不》希望出现所属权冲突或其他错误	0

要解释结果代码，请参见第 A-10 页的 “PCMD 结果代码”。

PCMD 结果代码

如果指定一个标记来存储 PCMD 指令的结果，则该指令在执行时返回以下代码之一：

代码 (Dec)：	说明：
0	命令成功。
24577	命令 无效。
24578	对于设备阶段的当前状态，命令 无效。例如，如果设备阶段处于 running 状态，则 start 命令 无效。
24579	您 《不能》命令设备阶段。下面某个软件或程序已拥有该设备阶段。 • RSLogix 5000 软件 • 外部定序程序 （RSBizWare Batch 软件） • 控制器中的其他程序
24594	设备阶段未调度、被禁止或处于被禁止的任务中。

算术状态标志： 不受影响

故障条件： 无

执行：

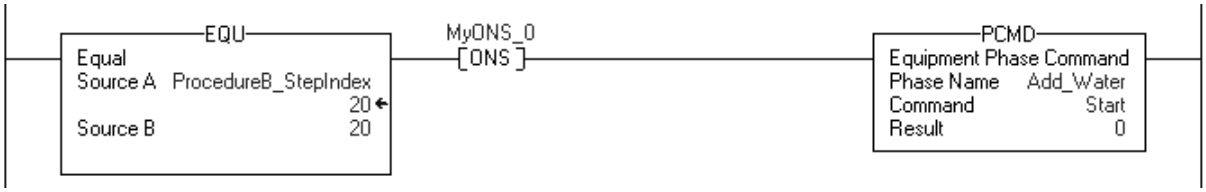
条件：	梯形图操作：	结构化文本操作：
预扫描	输出梯级条件设置为 false。	不采取任何操作。
输入梯级条件为 false	输出梯级条件设置为 false。	不可用
输入梯级条件为 true	• 指令执行。 • 输出梯级条件设置为 true。	不可用
结构化文本的扫描	不可用	在结构化文本中，指令在每次扫描时都会执行。要限制指令的扫描，请使用 SFC 操作的限定符和 / 或结构化文本结构。
指令执行	指令命令设备阶段进入指定的状态。	指令命令设备阶段进入指定的状态。
后扫描	输出梯级条件设置为 false。	不采取任何操作。

示例 1：

梯形图

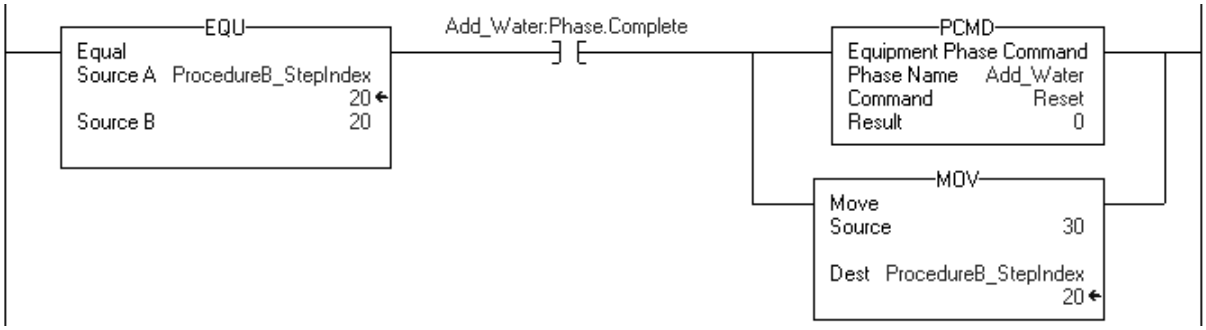
如果 *ProcedureB_StepIndex* = 20 （例程位于步骤 20）。
此操作转换到步骤 20 （ONS 指令发出信号，指示 EQU 指令从 false 变为了 true），
则

通过 start 命令将 *Add_Water* 设备阶段的状态更改为 running。

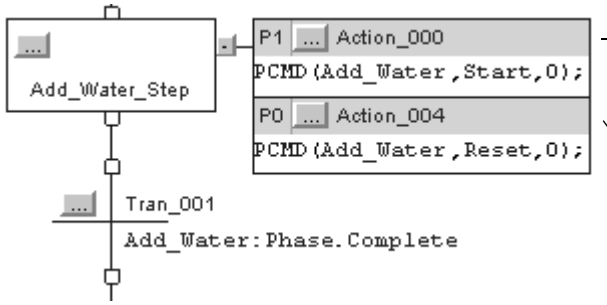


如果 *ProcedureB_StepIndex* = 20 （例程位于步骤 20）。
并且 *Add_Water* 设备阶段已完成 (*Add_Water:Phase.Complete* = 1)
则

通过 reset 命令将 *Add_Water* 设备阶段的状态更改为 resetting。
前进到步骤 30。



结构化文本



SFC 进入 *Add_Water_Step* 时，通过 start 命令将 *Add_Water* 设备阶段更改为 running 状态。P1 限定符将此操作限制于步骤的第一次扫描。

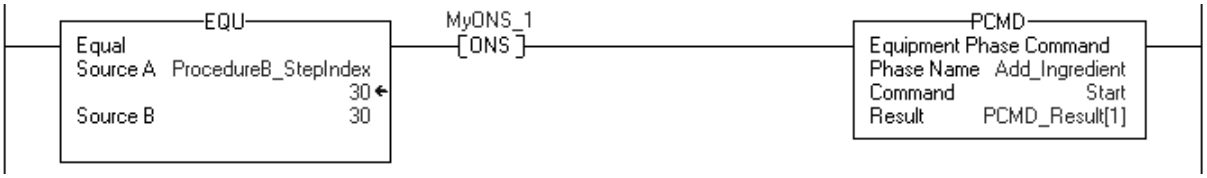
SFC 结束 *Add_Water_Step* (*Add_Water:Phase.Complete* = 1) 之前，通过 reset 命令将 *Add_Water* 设备阶段更改为 resetting 状态。P0 限定符将此操作限制于步骤的最后一次扫描。

示例 2：

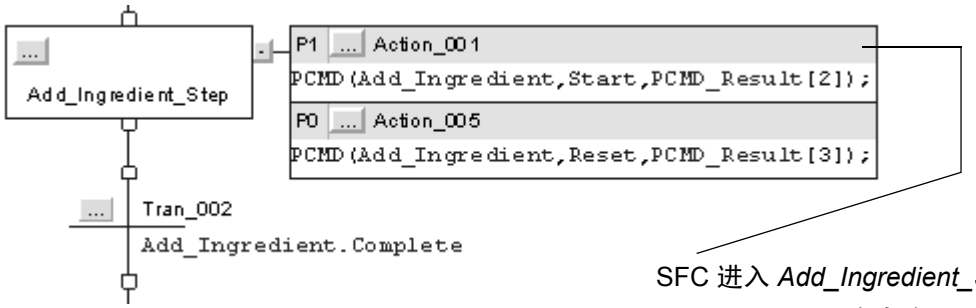
梯形图

如果 *ProcedureB_StepIndex* = 30 （例程位于步骤 30），
此操作转换到步骤 30 （ONS 指令发出信号，指示 EQU 指令从 false 变为了 true），
则

通过 start 命令将 *Add_Ingredient* 设备阶段更改为 running 状态。
确认命令是否成功，将结果代码存储在 *PCMD_Result[1]*（DINT 标记）中。



结构化文本



SFC 进入 *Add_Ingredient_Step* 时：

- 通过 start 命令将 *Add_Ingredient* 设备阶段更改为 running 状态。
- 确认命令是否成功，将结果代码存储在 *PCMD_Result[2]*（DINT 标记）中。

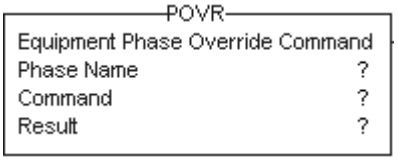
P1 限定符将此操作限制于步骤的第一次扫描。

设备阶段重写命令 (POVR)

无论所属权如何，使用 POVR 指令都可向设备阶段给出 hold、stop 或 abort 命令。

操作数：





梯形图

操作数：	类型：	格式：	说明：
阶段名称	阶段	设备阶段的名称	要更改为其他状态的设备阶段
命令	命令	命令的名称	设备阶段的以下命令之一： • hold • stop • abort
结果	DINT	直接标记	要使指令返回其成功 / 失败的代码，请输入一个 DINT 标记，该标记存储结果代码。 否则，输入 0。



POVR(*PhaseName, Command, Result*);

结构化文本

操作数与梯形图 POVR 指令的操作数一样。

说明： POVR 指令：

- 对设备阶段使用 hold、 stop 或 abort 命令。
- 重写设备阶段的全部所有者。即使 RSLogix 5000 软件、RSBizWare Batch 软件或其他程序已拥有设备阶段，该命令也会执行。

POVR 指令的使用准则

准则：	说明：
❑ **确保需要重写其他所有者。**	即使可通过 RSLogix 5000 软件手动控制设备阶段，是否也希望设备 hold、stop 或 abort？ • 是 — 使用 POVR 指令。 • 否 — 使用 PCMD 指令。 这也适用于 RSBizWare Batch 软件或其他程序。仅当无论所属权如何都必须 hold、 stop 或 abort 时才使用 POVR。 示例：假定设备检查堵塞材料。如果存在堵塞，您始终希望设备中止。这种情况下，使用 POVR 指令。这样，即使可以通过 RSLogix 5000 软件手动控制，设备也会中止。
❑ **将 POVR 的执行限制于单次扫描。**	将 POVR 的执行限制于单次扫描。每个命令都仅应用于一个或多个特定状态。只要设备阶段更改了状态，该命令就 《不再》 有效。要限制执行，请使用如下方法： • 在 P1 脉冲 （上升边缘）或 P0 脉冲 （下降边缘）操作中执行 POVR 指令 • 在 POVR 指令前放置一个 One Shot 指令 • 执行 POVR 指令后前进到下一步骤

POVR 结果代码

如果指定一个标记来存储 POVR 指令的结果，则该指令在执行时返回以下代码之一：

代码 (Dec)：	说明：
0	命令成功。
24577	命令 无效。
24578	对于设备阶段的当前状态，命令 无效。例如，如果设备阶段处于 stopping 状态，则 hold 命令 无效。
24594	设备阶段未调度、被禁止或处于被禁止的任务中。

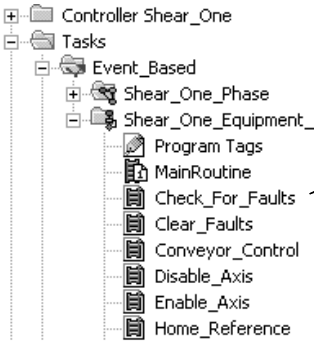
算术状态标志： 不受影响

故障条件： 无

执行：

条件：	梯形图操作：	结构化文本操作：
预扫描	输出梯级条件设置为 false。	不采取任何操作。
输入梯级条件为 false	输出梯级条件设置为 false。	不可用
输入梯级条件为 true	• 指令执行。 • 输出梯级条件设置为 true。	不可用
结构化文本的扫描	不可用	在结构化文本中，指令在每次扫描时都会执行。要限制指令的扫描，请使用 SFC 操作的限定符和 / 或结构化文本结构。
指令执行	指令命令设备阶段进入指定的状态。	指令命令设备阶段进入指定的状态。
后扫描	输出梯级条件设置为 false。	不采取任何操作。

示例：

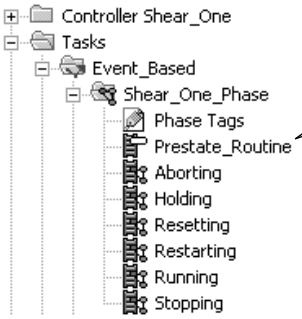


设备程序监视如下故障：

- 故障轴
- 堵塞的材料

如果出现故障，则

Local_Interface.Equipment_Faults_Cleared = 0。此标记是控制器范围的标记 *Shear_1* 的别名。



设备阶段的状态前例程查看设备程序是否发出信号指示出现了故障。

- 如果 *Interface_To_Equipment.Equipment_Faults_Cleared* = 0，则存在故障。
- *Interface_To_Equipment* 和 *Local_Interface* 都是 *Shear_1* 的别名，因此它们的值相同。

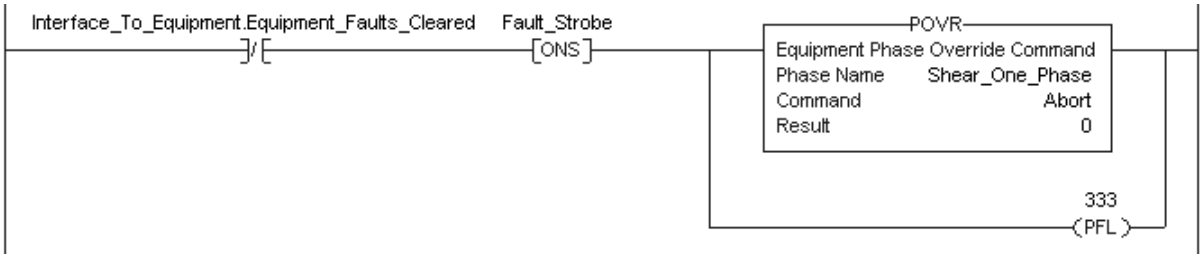
如果出现故障，则

对 *Shear_One_Phase* 设备阶段执行 abort 命令。即使有人可通过 RSLogix 5000 软件手动控制设备阶段，POVR 指令也确保该命令会执行。

PFL 指令设置 *Shear_One_Phase* = 333 的失效代码。

Fault_Strobe 使这些操作限制于单次扫描。

梯形图



结构化文本

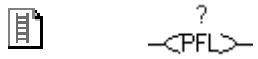
```
If NOT
Interface_To_Equipment.Equipment_Faults_Cleared
And NOT Fault_Strobe Then
    POVR(Shear_One_Phase,Abort,0);
    PFL(333);
End_If;

Fault_Strobe := NOT
Interface_To_Equipment.Equipment_Faults_Cleared;
```

设备阶段失效 (PFL)

将 PFL 指令用作可选方法，发出信号指示设备阶段失效。

操作数：



梯形图

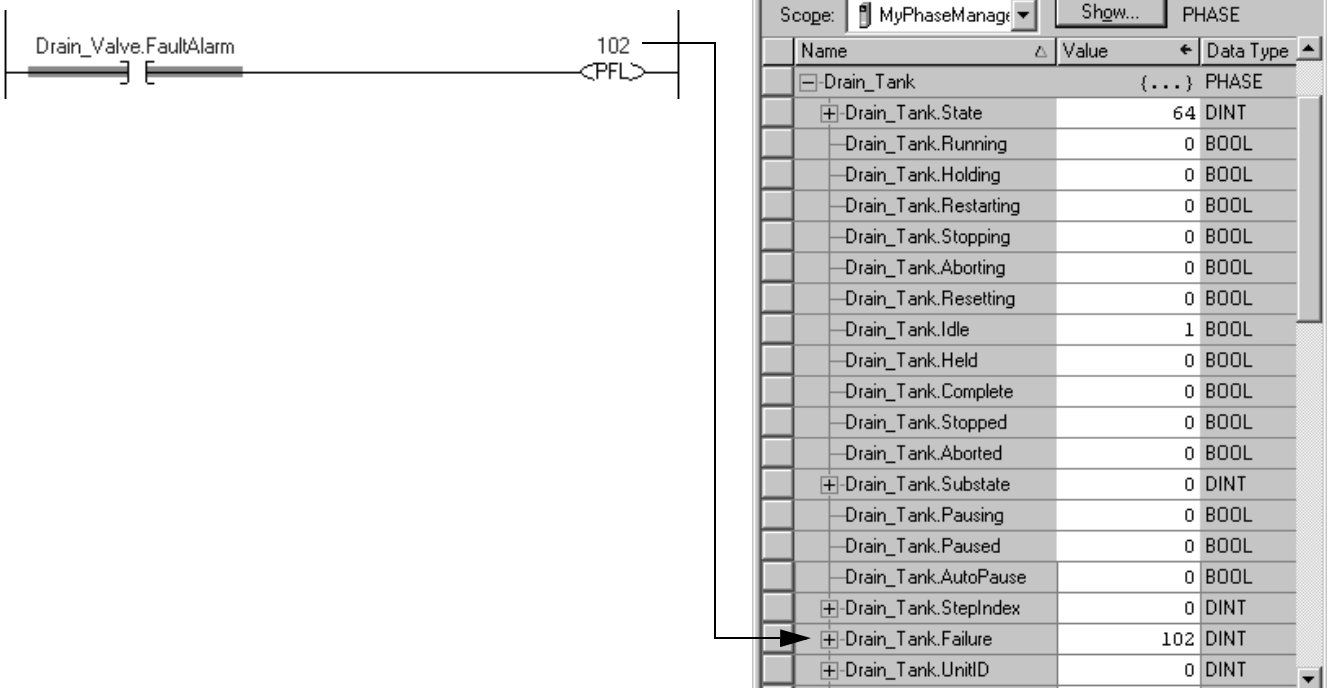
操作数：	类型：	格式：	说明：
Failure_Code	DINT	直接标记	希望向其设置设备阶段失效代码的值

```
PFL(Failure_Code);
```

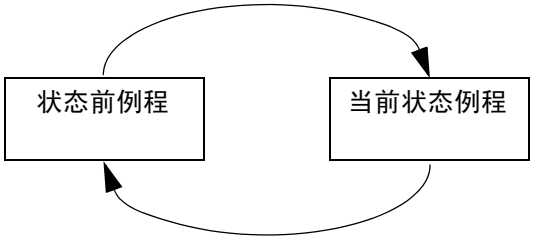
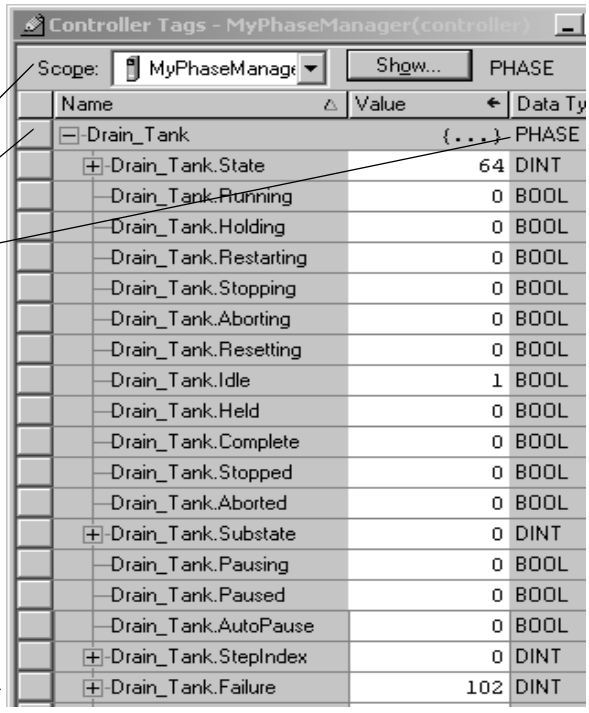
结构化文本

操作数与梯形图 PFL 令的操作数一样。

说明： PFL 指令设置设备阶段失效代码的值。使用此指令可发出信号指示设备阶段的特定失效，如特定设备出现了故障。



PFL 指令的使用准则

准则：	说明：
❑ **将 PFL 指令放入设备阶段。**	PFL 指令设置其所在的设备阶段的失效代码。《没有》任何操作数可标识特定设备阶段。 通常，将 PFL 指令放入设备阶段的状态前例程中。 • 控制器总是扫描状态前例程，即使设备阶段处于 idle 状态也是如此。 • 控制器在 每次状态扫描之前都扫描状态前例程。  • 在处理设备阶段各状态时，使用状态前例程继续监视其运行状况。
❑ **确定失效代码的优先级。**	PFL 指令仅将失效代码设置为大于当前值的值。 • 例如，如果 PFL 指令将失效代码设置为 102，则其他 PFL 指令只能将失效代码设置为大于 102 的值。 • 确保向需要较高优先级处理的异常赋较大的值。否则，较低优先级异常可能会改写更为重要的异常。
❑ **要在出现失效时采取操作，请监视 PHASE 标记的 Failure 成员。**	PFL 指令将其值写入设备阶段的 PHASE 标记的 Failure 成员。 如果创建一个设备阶段，RSLogix 5000 软件则为设备阶段的状态创建一个标记。 控制器范围 名称 = *phase_name* PHASE 数据类型 PFL 指令将其值写入设备阶段的 Failure 成员。 

准则：	说明：
❑ **要清除失效代码，请使用 PCLF 指令。**	必须使用 PCLF 指令才能清除设备阶段的失效代码。指令 （如 CLR 或 MOV）《不》更改失效代码。请参见第 A-21 页的 “设备阶段清除失效 (PCLF)” 指令。

算术状态标志： 不受影响

故障条件： 无

执行：

条件：	梯形图操作：	结构化文本操作：
预扫描	输出梯级条件设置为 false。	不采取任何操作。
输入梯级条件为 false	输出梯级条件设置为 false。	不可用
输入梯级条件为 true	• 指令执行。 • 输出梯级条件设置为 true。	不可用
结构化文本的扫描	不可用	在结构化文本中，指令在每次扫描时都会执行。要限制指令的扫描，请使用 SFC 操作的限定符和 / 或结构化文本结构。
指令执行	该指令设置设备阶段失效代码的值。	该指令设置设备阶段失效代码的值。
后扫描	输出梯级条件设置为 false。	不采取任何操作。

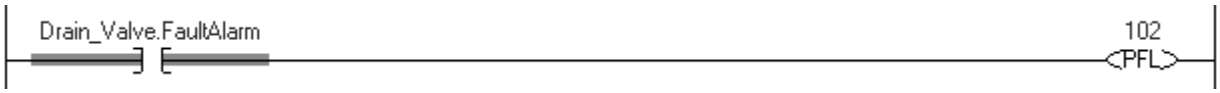
示例：

梯形图

在设备阶段的状态前例程中…

如果 *Drain_Valve.FaultAlarm* = 1 （阀未变为 commanded 状态），则

设备阶段的失效代码为 102。

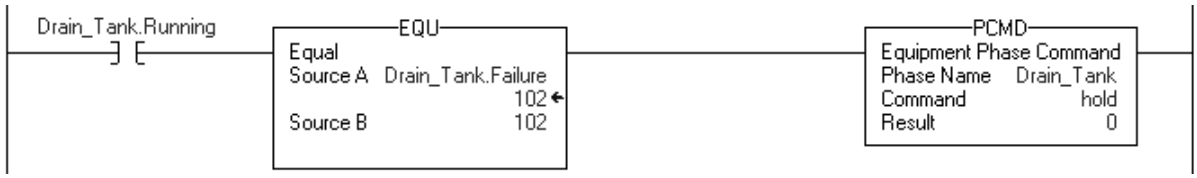


如果 *Drain_Tank.Running* = 1 （Drain_Tank 设备阶段处于 running 状态），

并且 *Drain_Tank.Failure* = 102 （设备阶段的失效代码）

则

通过 hold 命令将 *Drain_Tank* 设备阶段的状态更改为 holding。



结构化文本

在设备阶段的状态前例程中…

```
(* 如果排水阀不进入 commanded 状态，则将此设备阶段的失效代码设置为 102。 *)
If Drain_Valve.FaultAlarm Then
    PFL(102);
End_If;

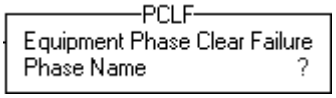
(* 如果 Drain_Tank 设备阶段的状态为 running，并且其失效代码为 102，则执行 hold 命令，并使设备阶段变为 holding 状态。 *)
If Drain_Tank.Running And (Drain_Tank.Failure = 102) Then
    PCMD(Drain_Tank,hold,0);
End_If;
```

设备阶段清除失效 (PCLF)

使用 PCLF 指令清除设备阶段的失效代码。

操作数：





梯形图

操作数：	类型：	格式：	说明：
阶段名称	阶段	设备阶段的名称	要清除其失效代码的设备阶段



```
PCLF(Phase_Name);
```

结构化文本

操作数与梯形图 PCLF 指令的操作数一样。

说明： PCLF 指令清除设备阶段的失效代码。

- 必须使用 PCLF 指令才能清除设备阶段的失效代码。
- CLR 指令、 MOV 指令或赋值 (:=) 不会更改设备阶段的失效代码。
- 确保在使用 PCLF 指令时设备阶段没有其他所有者。如果 RSLogix 5000 软件、 RSBizWare Batch 软件或其他程序拥有设备阶段， PCLF 指令不会清除失效代码。

算术状态标志： 不受影响

故障条件： 无

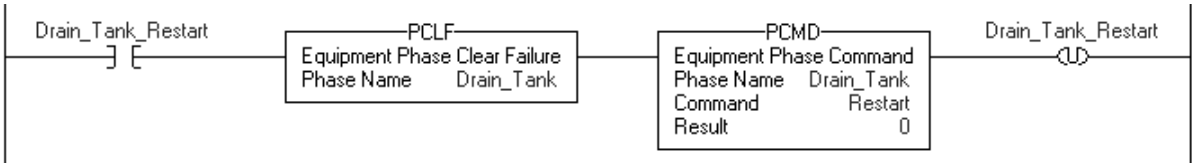
执行：

条件：	梯形图操作：	结构化文本操作：
预扫描	输出梯级条件设置为 false。	不采取任何操作。
输入梯级条件为 false	输出梯级条件设置为 false。	不可用
输入梯级条件为 true	• 指令执行。 • 输出梯级条件设置为 true。	不可用
结构化文本的扫描	不可用	在结构化文本中，指令在每次扫描时都会执行。要限制指令的扫描，请使用 SFC 操作的限定符和 / 或结构化文本结构。
指令执行	该指令清除指定的设备阶段失效代码的值。	该指令清除指定的设备阶段失效代码的值。
后扫描	输出梯级条件设置为 false。	不采取任何操作。

示例：

梯形图

如果 *Drain_Tank_Restart* = 1 （重新启动 *Drain_Tank* 设备阶段），则
清除 *Drain_Tank* 设备阶段的失效代码
通过 restart 命令将 *Drain_Tank* 设备阶段的状态更改为 restarting 状态。
Drain_Tank_Restart = 0。



结构化文本

```
(* 如果 Drain_Tank_Restart = on，则：
     清除 Drain_Tank 设备阶段的失效代码。
     重新启动 Drain_Tank 设备阶段。
     关闭 Drain_Tank_Restart。 *)

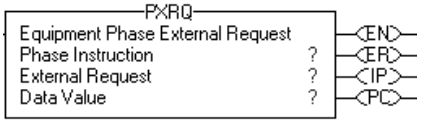
If Drain_Tank_Restart Then
     PCLF(Drain_Tank);
     PCMD(Drain_Tank,Restart,0);
     Drain_Tank_Restart := 0;
End_If;
```

设备阶段外部请求 (PXRQ)

使用 PXRQ 指令开始与 RSBizWare Batch 软件的通讯。

操作数：

梯形图



操作数：	类型：	格式：	说明：
阶段指令	PHASE_INSTRUCTION	标记	控制操作的标记
外部请求	请求	名称	请求的类型 有关可用的请求，请参见第 A-26 页。
数据值	DINT	数组标记	请求的参数 有关数组大小和数据值，请参见第 A-26 页。

结构化文本

```
PXRQ(Phase_Instruction,
External_Request,
Data_Value);
```

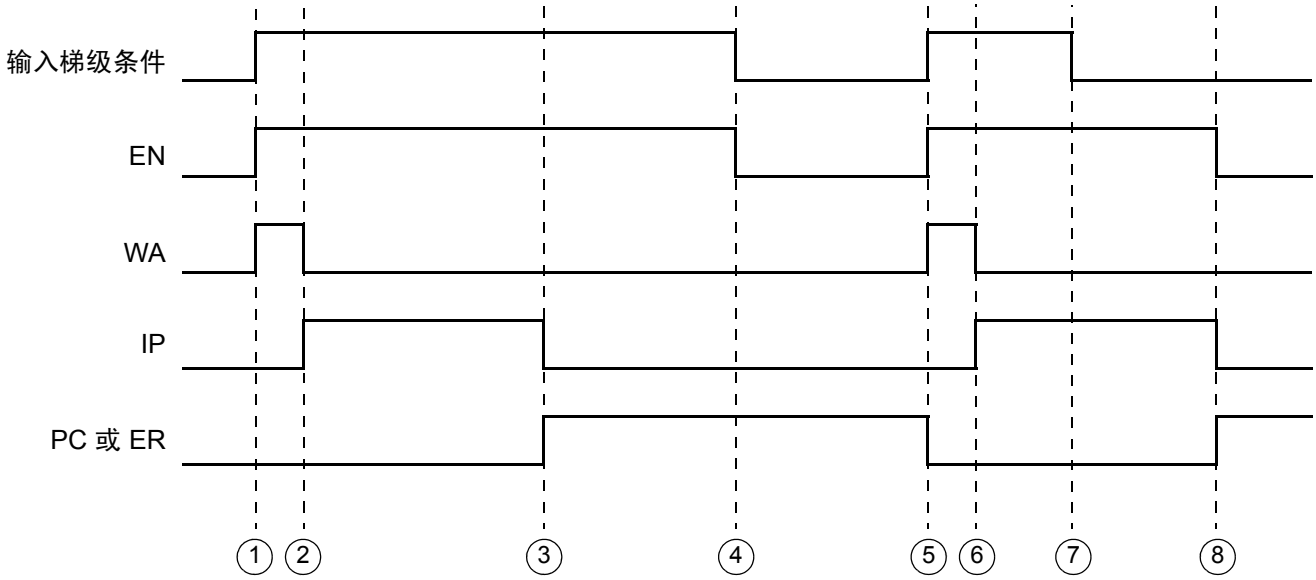
操作数与梯形图 PXRQ 指令的操作数一样。

PHASE_INSTRUCTION 数据类型

如果需要：	则检查或设置如下成员：	数据类型：	说明：
确定从 false 到 true 的转换是否导致指令执行	EN	BOOL	参见第 A-25 页的图 A.2 。
确定请求是否失败	ER	BOOL	参见第 A-25 页的图 A.2 。 要诊断错误，请参见 ERR 和 EXERR 值。
确定 RSBizWare Batch 软件是否已完成其请求处理	PC	BOOL	参见第 A-25 页的图 A.2 。
确定 RSBizWare Batch 软件是否正在处理请求	IP	BOOL	参见第 A-25 页的图 A.2 。
确定指令是否已发送请求但 RSBizWare Batch 软件还未确认	WA	BOOL	参见第 A-25 页的图 A.2 。 满足如下条件之一， WA 也等于 0： • 连接超时 • 出现网络错误 • ABORT = 1
取消请求	abort	BOOL	要中止 （取消）请求，请将 ABORT 位设置为 1。这时控制器会中止指令： • ER = 1 • ERR 显示中止的结果

如果需要：	则检查或设置如下成员：	数据类型：	说明：
• 诊断错误原因 • 编写逻辑响应特定错误	ERR	INT	如果 ER = 1，则错误代码会提供诊断信息。要解释错误代码，请参见第 A-31 页的“PXRQ 错误代码”。
	EXERR	INT	如果 ER = 1，则扩展错误代码提供某些错误的更多诊断信息。要解释扩展错误代码，请参见第 A-31 页的 “PXRQ 错误代码”。
对标记的不同状态位使用一个成员	Status	DINT	**对于成员：** **使用位：** EN 31 ER 28 PC 27 IP 26 WA 25 abort 24

图 A.2 PXRQ 指令的时序图



说明：		说明：	
①	输入梯级条件变为 true。 指令向 RSBizWare Batch 软件发送请求。	⑤	输入梯级条件变为 true。 指令向 RSBizWare Batch 软件发送请求。
②	RSBizWare Batch 软件开始处理请求。	⑥	RSBizWare Batch 软件开始处理请求。
③	出现以下情况之一： • RSBizWare Batch 软件完成其请求处理。(PC = 1)。 • 出现错误。 (ER = 1)。	⑦	输入梯级条件变为 false。 • EN 保持为 1。 • RSBizWare Batch 软件继续处理请求。(IP = 1)。
④	输入梯级条件变为 false。	⑧	出现以下情况之一： • RSBizWare Batch 软件完成其请求处理。(PC = 1)。 • 出现错误。 (ER = 1)。 由于输入梯级条件为 false， EN = 0。

说明： PXRQ 指令向 RSBizWare Batch 软件发送请求。

PXRQ 指令的使用准则

准则：	说明：
❑ **确保对数据值操作数使用数组。**	数据值操作数需要 DINT 数组，即使该数组只包含一个元素 （即数据类型为 DINT[1]）也是如此。
❑ **在梯形图中，设定指令在转换时执行的条件。**	这是转换指令。每次需要执行此指令时，将梯级输入条件由 false 切换为 true。
❑ **在结构化文本中，使用一个结构制约指令的执行。**	在结构化文本中编写 PXRQ 指令时，应考虑如下内容： • 在结构化文本中，指令在 每次 扫描时都会执行。 • PXRQ 指令 《仅》在被扫描时更新其状态位。 要防止指令重复执行同时确保更新状态位，请将指令包含在一个结构中，该结构： • 《仅》在转换 （条件发生更改）时开始指令的执行 • 在 PC = 1 或 ER = 1 之前保持为 true

配置 PXRQ 指令

有关 PXRQ 请求的更多信息，请参见 *RSBizWare Batch PhaseManager User's Guide* （RSBizWare Batch PhaseManager 用户指南），出版号 BATCHXUMA008-EN-P。

如果需要：	则配置 PXRQ 指令，如下所示：		
	外部请求：	数据值数组元素：	值：
下载所有输入参数	下载输入参数	DINT[0]	0
下载单个输入参数	下载输入参数	DINT[0]	参数 ID
下载某个范围的输入参数	下载输入参数	DINT[0]	第一个参数的参数 ID
		DINT[1]	值的编号
在启动或转移控制时下载为自动下载配置的输入参数	下载输入参数子集	DINT[0]	启动时为 1 控制转移时为 2
下载所有输出参数限制	下载输出参数限制	DINT[0]	0
下载单个输出参数限制	下载输出参数限制	DINT[0]	参数 ID
上载所有报告	上载输出参数	DINT[0]	0
上载单个报告	上载输出参数	DINT[0]	报告 ID
上载某个范围的报告	上载输出参数	DINT[0]	第一个报告的报告 ID
		DINT[1]	值的编号
在 terminal 状态或控制转移时上载为自动上载配置的输出参数	上载输出参数子集	DINT[0]	terminal 状态时为 1 控制转移时为 2

如果需要：	**则配置 PXRQ 指令，如下所示：**		
	外部请求：	**数据值数组元素：**	**值：**
向操作员发送消息	向操作员发送消息	DINT[0]	消息 ID
清除来自操作员的一条消息	清除操作员的消息	DINT[0]	0
需要一个资源	需要资源	DINT[0]	设备 ID
需要多个资源	需要资源	DINT[0]	设备 ID
		DINT[1]	设备 ID
		…	…
释放单个资源	释放资源	DINT[0]	设备 ID
释放多个资源	释放资源	DINT[0]	设备 ID
		DINT[1]	设备 ID
		…	…
释放所有资源	释放资源	DINT[0]	0
向其他阶段发送消息（及可选数据）	向链接的阶段发送消息	DINT[0]	消息 ID
		DINT[1]	接收器的编号
		DINT[2]	值 1
		DINT[3]	值 2
		…	…
向其他阶段发送一条消息（及可选数据）并等待阶段接收该消息	向链接的阶段发送消息并等待	DINT[0]	消息 ID
		DINT[1]	接收器的编号
		DINT[2]	值 1
		DINT[3]	值 2
		…	…
等待来自其他阶段的消息	接收来自链接的阶段的消息	DINT[0]	消息 ID
		DINT[1]	值 1
		DINT[2]	值 2
		…	…
取消到其他阶段的消息	取消到链接的阶段的消息	DINT[0]	消息 ID
取消到其他阶段的所有消息	取消到链接的阶段的消息	DINT[0]	0
下载客户的批 ID	下载批数据	DINT[0]	1
		DINT[1]	要在其中存储该值的参数 ID
下载唯一的批 ID	下载批数据	DINT[0]	2
		DINT[1]	要在其中存储该值的参数 ID
下载阶段 ID	下载批数据	DINT[0]	3
		DINT[1]	要在其中存储该值的参数 ID

如果需要：	则配置 PXRQ 指令，如下所示：		
	外部请求：	数据值数组元素：	值：
下载配方控制或手动阶段控制	下载批数据	DINT[0]	4
		DINT[1]	要在其中存储该值的参数 ID 在结果值中： 0 为配方控制 1 为手动阶段控制
下载阶段的当前模式	下载批数据	DINT[0]	5
		DINT[1]	要在其中存储该值的参数 ID 在结果值中： 0 为 P - 自动模式 1 为 O - 自动模式
下载输入参数的下限	下载批数据	DINT[0]	6 输入参数标记存储下限。
下载输入参数的上限	下载批数据	DINT[0]	7 输入参数标记存储上限。
下载与当前使用的容器相关的数据。	下载正在使用的材料轨道数据容器	DINT[0]	1
		DINT[1]	属性 ID
		DINT[2]	要在其中存储该值的参数 ID
下载与当前使用的容器内的当前材料相关的数据。	下载正在使用的材料轨道数据容器	DINT[0]	2
		DINT[1]	属性 ID
		DINT[2]	要在其中存储该值的参数 ID
下载与当前使用的容器内的当前批相关的数据。	下载正在使用的材料轨道数据容器	DINT[0]	3
		DINT[1]	属性 ID
		DINT[2]	要在其中存储该值的参数 ID
上载与当前使用的容器相关的数据	上载正在使用的材料轨道数据容器	DINT[0]	1
		DINT[1]	属性 ID
		DINT[2]	具有该值的参数 ID
上载与当前使用的容器内的当前材料相关的数据。	上载正在使用的材料轨道数据容器	DINT[0]	2
		DINT[1]	属性 ID
		DINT[2]	具有该值的参数 ID
上载与当前使用的容器内的当前批相关的数据。	上载正在使用的材料轨道数据容器	DINT[0]	3
		DINT[1]	属性 ID
		DINT[2]	具有该值的参数 ID
下载当前绑定的容器的优先级	下载容器绑定优先级	DINT[0]	要在其中存储该值的参数 ID
上载当前绑定的新容器优先级	上载容器绑定优先级	DINT[0]	具有该值的参数 ID

如果需要：	**则配置 PXRQ 指令，如下所示：**		
	外部请求：	**数据值数组元素：**	**值：**
下载有关材料是否充足的信息	下载充足的材料	DINT[0]	要在其中存储该值的参数 ID 在结果值中： 0 为材料不足 1 为材料充足
生成签名	生成电子签名	DINT[0]	签名模板的 ID
		DINT[1]	定义签名是否可取消： 不能取消为 0 可以取消为非 0
下载材料属性	下载材料轨道数据库数据	DINT[0]	0
		DINT[1]	要在其中存储该值的参数 ID
		DINT[2]	控制器 ID
		DINT[3]	属性 ID
下载批属性	下载材料轨道数据库数据	DINT[0]	1
		DINT[1]	要在其中存储该值的参数 ID
		DINT[2]	控制器 ID
		DINT[3]	属性 ID
下载容器属性	下载材料轨道数据库数据	DINT[0]	3
		DINT[1]	要在其中存储该值的参数 ID
		DINT[2]	控制器 ID
		DINT[3]	属性 ID
下载容器优先级赋值	下载材料轨道数据库数据	DINT[0]	4
		DINT[1]	要在其中存储该值的参数 ID
		DINT[2]	材料 ID
		DINT[3]	容器 ID
		DINT[4]	进料类型： 1 为添加到容器 2 为从容器分发
上载材料属性	上载材料轨道数据库数据	DINT[0]	5
		DINT[1]	具有该值的报告 ID
		DINT[2]	控制器 ID
		DINT[3]	属性 ID
上载批属性	上载材料轨道数据库数据	DINT[0]	6
		DINT[1]	具有该值的报告 ID
		DINT[2]	控制器 ID
		DINT[3]	属性 ID

如果需要：	则配置 PXRQ 指令，如下所示：		
	外部请求：	数据值数组元素：	值：
上载容器属性	上载材料轨道数据库数据	DINT[0]	8
		DINT[1]	具有该值的报告 ID
		DINT[2]	控制器 ID
		DINT[3]	属性 ID
上载容器优先级赋值	上载材料轨道数据库数据	DINT[0]	9
		DINT[1]	具有该值的报告 ID
		DINT[2]	材料 ID
		DINT[3]	容器 ID
		DINT[4]	进料类型： 添加到容器为 1 从容器分发为 2

PXRQ 错误代码

ERR （十六进制）	EXERR （十六进制）	说明	建议操作
00	0000	PXRQ 指令在将请求发送给 RSBizWare Batch 软件之前被中止。	无
01	0000	PXRQ 指令在将请求发送给 RSBizWare Batch 软件之后被中止。	无
02	0000	两个或更多 PXRQ 指令同时使用同样的请求类型执行	同一时间只限执行一个 PXRQ 指令。
03	0110	通讯错误。请求未发送，原因是该阶段没有任何订阅者。	确保 RSBizWare Batch 软件已连接并正在运行。
	0210	通讯错误。请求未发送，原因是该 Notify 对象没有任何连接。	确保 RSBizWare Batch 软件已连接并正在运行。
	0410	通讯错误。发送失败。	确保到 RSBizWare Batch 软件有连接和通讯路径。
	1010	通讯错误。请求未发送，原因是 RSBizWare Batch 软件未订阅接收外部请求。	确保 RSBizWare Batch 软件已连接并正在运行。
	1020	RSBizWare Batch 软件未附加到该阶段。	确保 RSBizWare Batch 软件附加到该阶段。
04	0002	RSBizWare Batch 软件在处理请求时遇到错误。	确保到 RSBizWare Batch 软件有连接和通讯路径。
	0003	PXRQ 指令包含无效值。	确保到 RSBizWare Batch 软件有连接和通讯路径。
	0004	RSBizWare Batch 软件处理请求的状态不正确。	确保到 RSBizWare Batch 软件有连接和通讯路径。
	0005	两个或更多 PXRQ 指令同时使用不同的请求类型执行	同一时间只限执行一个 PXRQ 指令。
	0006	请求处理结束后存储到参数标记时出现错误。	确保到 RSBizWare Batch 软件有连接和通讯路径。
05	0000	RSBizWare Batch 软件接收到请求，但传递回一个无效的 Cookie。	确保到 RSBizWare Batch 软件有连接和通讯路径。
06	0000	PXRQ 指令向 RSBizWare Batch 软件发送一个无效的参数。	确保到 RSBizWare Batch 软件有连接和通讯路径。

算术状态标志： 不受影响

故障条件： 无

执行：

条件：	梯形图操作：	结构化文本操作：
预扫描	输出梯级条件设置为 false。	不采取任何操作。
输入梯级条件为 false	输出梯级条件设置为 false。	不可用
输入梯级条件为 true	• 输入梯级条件由 false 变为 true 时，指令执行一次。 • 输出梯级条件设置为 true。	不可用
结构化文本的扫描	不可用	在结构化文本中，指令在每次扫描时都会执行。要限制指令的扫描，请使用 SFC 操作的限定符和 / 或结构化文本结构。
指令执行	指令向 RSBizWare Batch 软件发送指定的请求。	指令向 RSBizWare Batch 软件发送指定的请求。
后扫描	输出梯级条件设置为 false。	不采取任何操作。

示例：

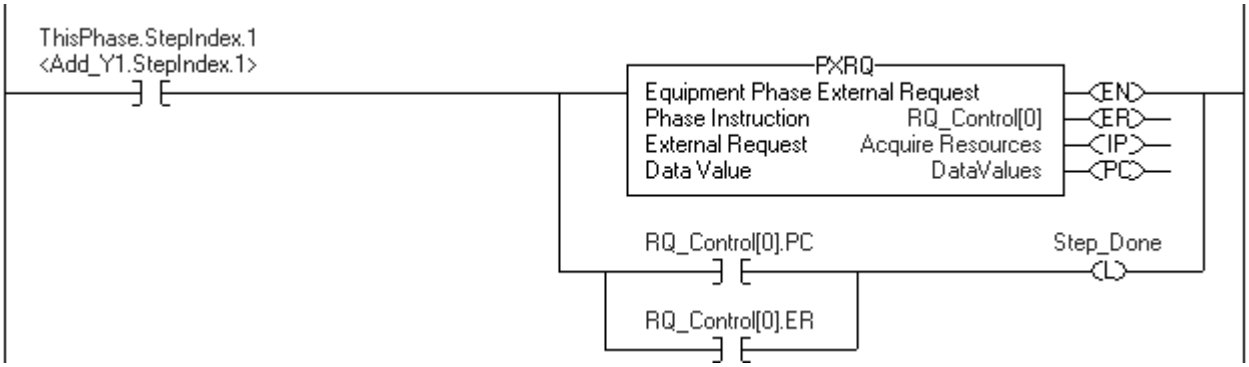
梯形图

如果 *ThisPhase.StepIndex.1* = 1 （例程位于步骤 1），则

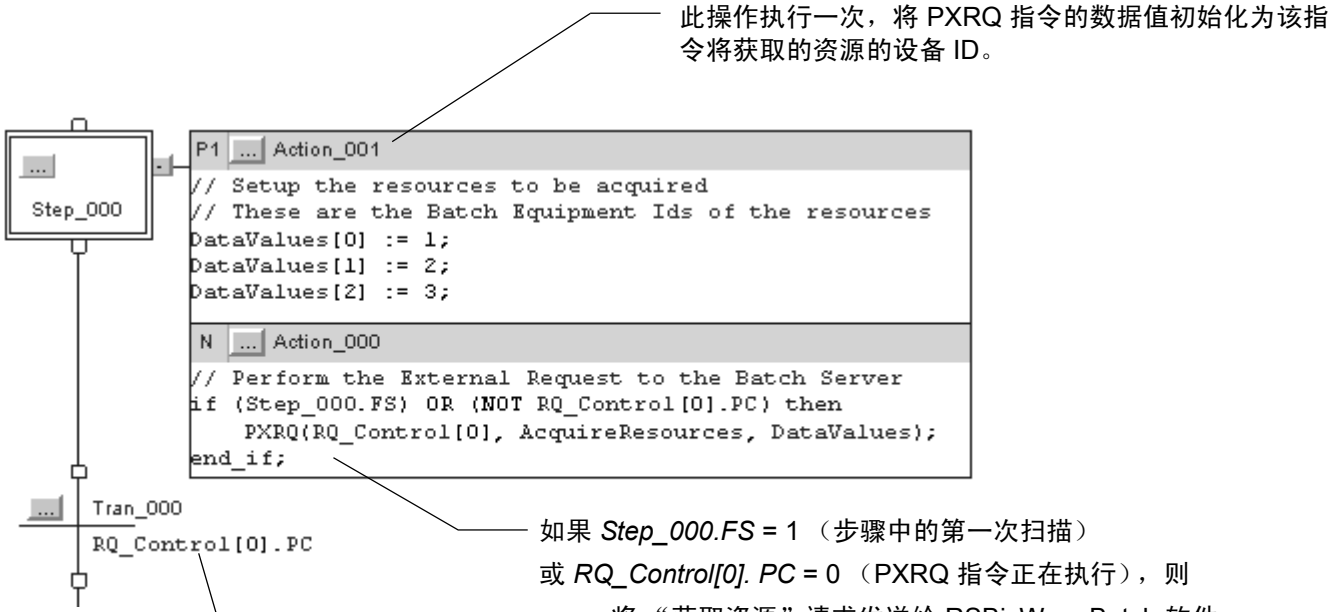
将 “获取资源”请求发送给 RSBizWare Batch 软件。 *DataValues* 标记是一个 DINT 数组，其中包含资源的设备 ID。

如果 *RQ_Control[0].PC* = 1 或 *RQ_Control[0].ER* = 1 （请求完成或失败），则

Done = 1 （此值通知序列进入下一步骤）。



结构化文本

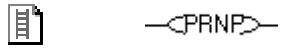


设备阶段新参数 (PRNP)

使用 PRNP 指令清除设备阶段的 NewInputParameters 位。

操作数：

梯形图



无

结构化文本

```
PRNP( );
```

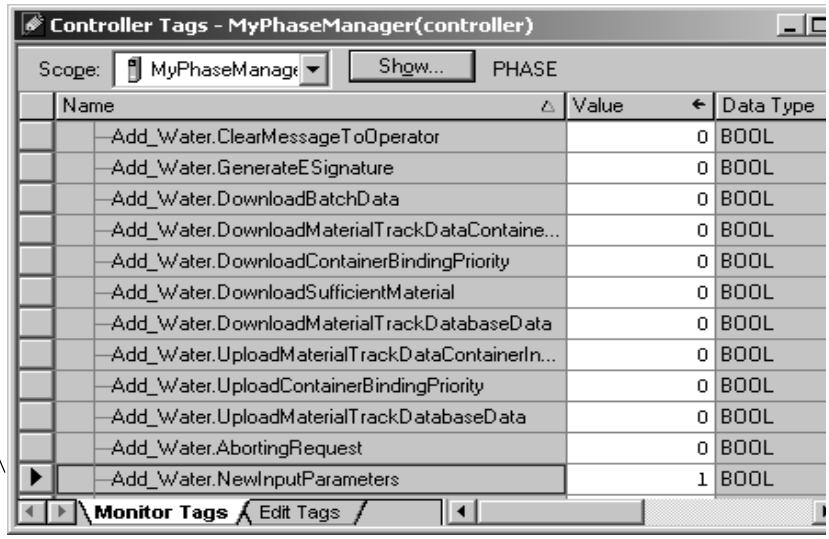
无

在指令助记码之后必须输入括号 ( )，即使没有操作数也如此。

说明： PRNP 指令清除设备阶段的 NewInputParameters 位。

如果 RSBizWare Batch 软件具有某个设备阶段的新参数，则会设置该阶段的 NewInputParameters 位。

下载参数之后，使用 PRNP 指令将该位清除。



算术状态标志： 不受影响

故障条件： 无

执行：

条件：	梯形图操作：	结构化文本操作：
预扫描	输出梯级条件设置为 false。	不采取任何操作。
输入梯级条件为 false	输出梯级条件设置为 false。	不可用
输入梯级条件为 true	• 指令执行。 • 输出梯级条件设置为 true。	不可用
结构化文本的扫描	不可用	在结构化文本中，指令在每次扫描时都会执行。要限制指令的扫描，请使用 SFC 操作的限定符和 / 或结构化文本结构。
指令执行	指令清除设备阶段的 NewInputParameters 位。	指令清除设备阶段的 NewInputParameters 位。
后扫描	输出梯级条件设置为 false。	不采取任何操作。

示例：

梯形图

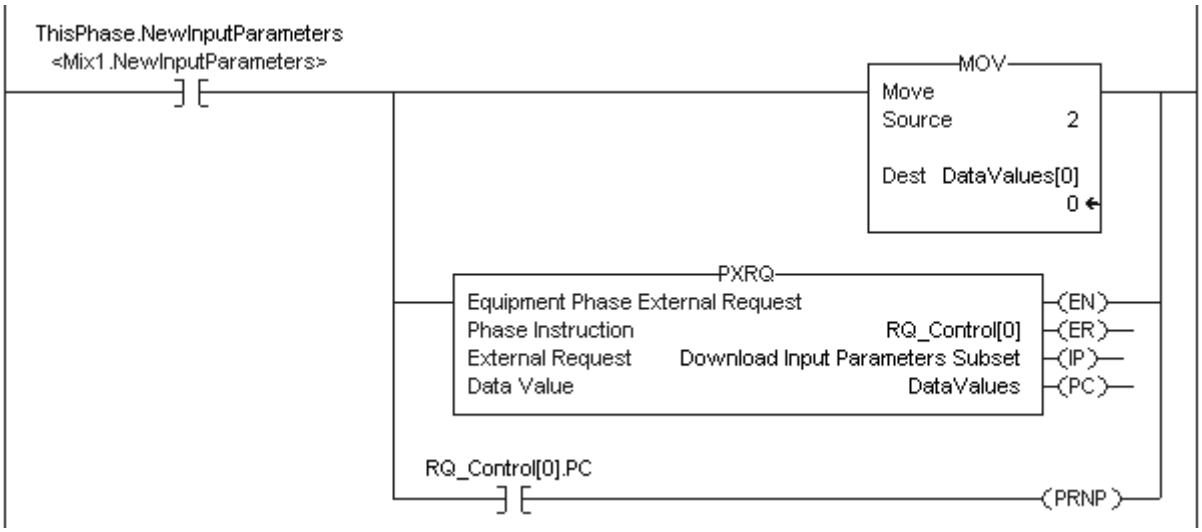
如果 *ThisPhase.NewInputParameters* = 1 （RSBizWare Batch 软件具有设备阶段的新输入参数），则

DataValues[0] = 2。这将 PXRQ 指令设置为用于控制转移。

将 “下载输入参数子集”请求发送给 RSBizWare Batch 软件。由于 *DataValues[0]* = 2，该指令被设置为用于控制转移。

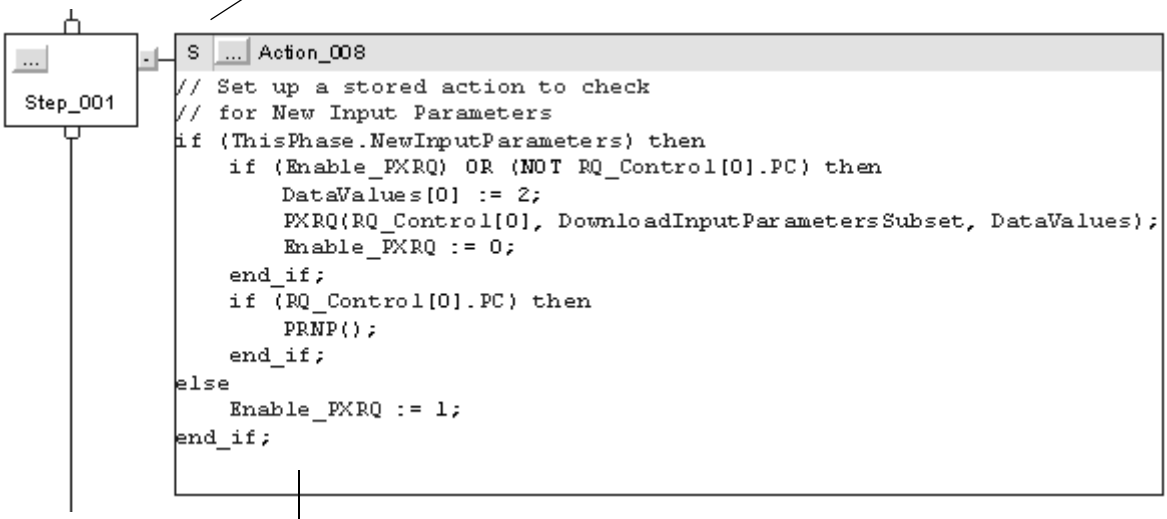
如果 *RQ_Control[0].PC* = 1 （PXRQ 指令已完成），则

通过 PRNP 指令使 *ThisPhase.NewInputParameters* = 0。



结构化文本

此为存储操作。即使 SFC 执行完 Step_001，它也会继续执行。



如果 *ThisPhase.NewInputParameters* = 1 （RSBizWare Batch 软件具有设备阶段的新输入参数），则

如果 *Enable_PXRQ* = 1 （使 PXRQ 指令执行）

或 *RQ_Control[0]. PC* = 0 （PXRQ 指令正在执行），则

DataValues[0] = 2。这将 PXRQ 指令设置为用于控制转移。

将 “下载输入参数子集”请求发送给 RSBizWare Batch 软件。由于 *DataValues[0]* = 2，该指令被设置为用于控制转移。

Enable_PXRQ = 0 （请求完成后，《不要》重新启动 PXRQ 指令）。

如果 *RQ_Control[0]. PC* = 1 （请求完成），则

通过 PRNP 指令使 *ThisPhase.NewInputParameters* = 0。

否则

Enable_PXRQ = 1 （新输入参数下次可用时，执行 PXRQ 指令）。

设备阶段暂停 (PPD)

使用 PPD 指令在设备阶段逻辑中设置断点。

操作数：



-[PPD]-

梯形图

无



PPD( );

结构化文本

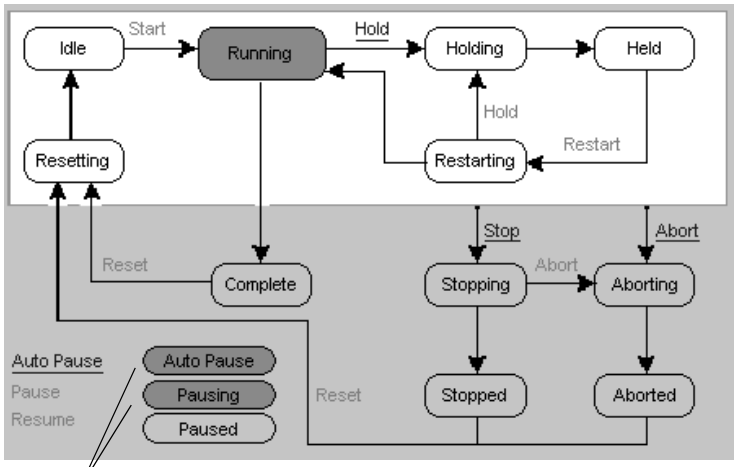
无

在指令助记码之后必须输入括号 ( )，即使没有操作数也如此。

说明： 使用 PPD 指令可以在特定步骤 （断点）处停止执行，以测试和解决逻辑问题。如果设备阶段处于 pausing 子状态，控制器执行一条 PPD 指令，则控制器：

- 将 PHASE 标记的 Paused 位设置为 1
- 使梯级的其余部分为 false (RLL)

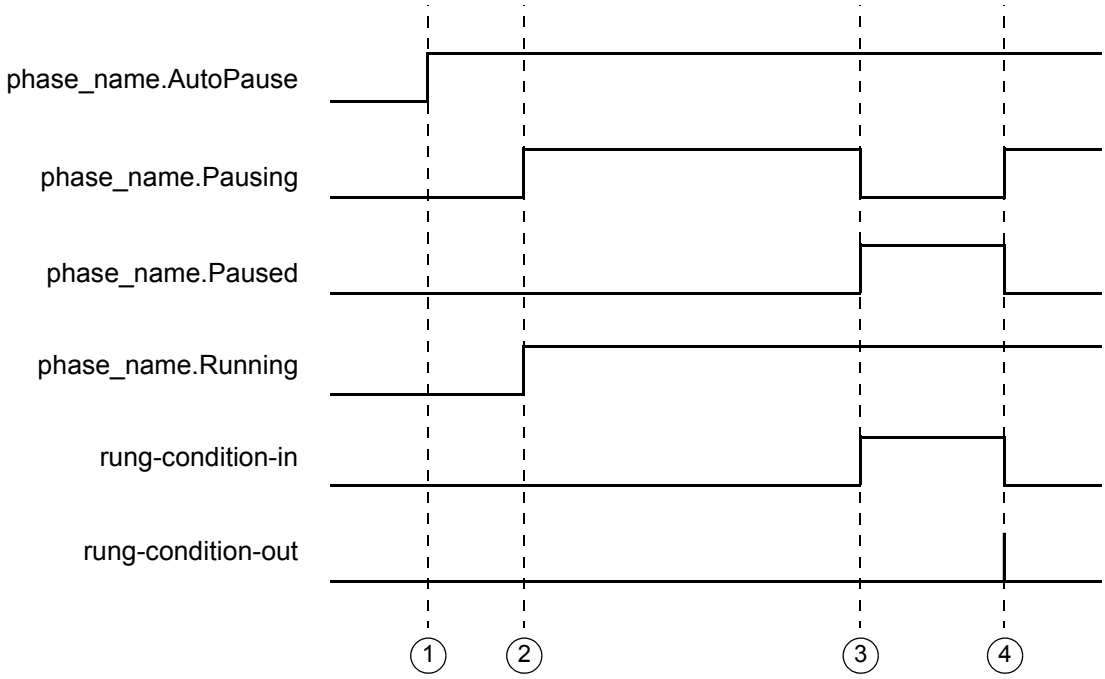
将 PPD 指令放入设备阶段后，您可以使用或忽略这些指令。



使用 auto pause 和 pausing 子状态可以控制设备阶段是否使用断点。

使用 auto pause、 pause 和 resume 命令可以更改子状态和单步执行断点。

下面的时序图说明 PPD 指令如何影响 PHASE 标记的子状态位。



说明：	
①	设备阶段获取 auto pause 命令。
②	设备阶段获取 start 命令。 设备阶段一旦启动，即进入 pausing 子状态。
③	PPD 指令的输入条件变为 true。 设备阶段由 pausing 子状态更改为 paused 子状态
④	设备阶段获取 resume 命令。 逻辑进入下一步骤，因此输入条件变为 false 并且不再执行输出。 由于 auto pause 已经开启，因此设备阶段自动回到 pausing 子状态。

断点的使用准则

准则：	说明：
❑ **将逻辑组织为一系列步骤。**	如果逻辑逐步执行定义的步骤 （如状态机或 SFC），则 PPD 指令 （断点）是最容易使用的。 • 《仅》在满足特定条件时，断点才会进行通知。《它不》会停止设备阶段的执行。 • 要在断点处使逻辑实际中断 (pause)，请对逻辑进行组织，以便逻辑停在断点处，直到使用 resume 命令为止。 请参见第 A-41 页中的示例。
❑ **《不要》将 PPD 指令用作例程的临时结尾。**	设备阶段的状态即使为 paused，也会继续执行其所有逻辑。 • PPD 指令执行时，它只设置设备阶段的 Paused 位。 • 如果在 RLL 中使用 PPD 指令，它只禁用其梯级的其余逻辑。它 《不》会终止或挂起例程的执行。 • 根据 auto pause 和 pause 命令，将 PPD 指令视为可以应用或忽略的条件。
❑ **将 PPD 的执行限制于单次扫描。**	在 pausing 子状态下，设备阶段在执行条件为 true 的 《第一个》PPD 指令时变为 paused 状态。如果 PPD 指令在几次扫描过程中执行，设备阶段可能在同一断点处继续保持为 pause 状态。（这与仅在 false 转换为 true 时执行的 One Shot (ONS) 指令不同。）
❑ **确保同一时间只有一个 PPD 指令为 true。**	PPD 指令 《没有》控制标记可指示其是否已执行。 • 只要条件为 true （且设备阶段处于 pausing 子状态）， PPD 指令就充当断点 （设备阶段设置为 paused 状态）。 • 将逻辑一次限制到一个可能的断点可确保在所需断点暂停。
❑ **选择正确的子状态。**	《仅当》设备阶段处于 pausing 子状态时， PPD 指令 （断点）才会执行：

如果暂停位置为：	使用命令：	说明：
每个实际断点	auto pause	• 设备阶段启动后，则进入 pausing 子状态。 • 在进入 pause 状态后使用 resume 命令，则设备阶段保持处于 pausing 子状态。 • 要关闭 auto pause 状态，请再次使用 auto pause 命令。
第一个实际断点	pause	• 启动设备阶段后使用 pause 命令。 • 要在其他断点处暂停，请使用 resume 命令，然后使用 pause 命令。

算术状态标志： 不受影响

故障条件： 无

执行：

条件：	梯形图操作：		结构化文本操作：	
预扫描	输出梯级条件设置为 false。		不采取任何操作。	
输入梯级条件为 false	输出梯级条件设置为 false。		不可用	
输入梯级条件为 true	指令执行。		不可用	
结构化文本的扫描	不可用		在结构化文本中，指令在每次扫描时都会执行。要限制指令的扫描，请使用 SFC 操作的限定符和 / 或结构化文本结构。	
指令执行	**子状态：**	**动作：**	**子状态：**	**动作：**
	pausing	• 子状态为 paused。 • PHASE 标记的 Paused 位为 1。 • rung-condition-out 为 false。	pausing	• 子状态为 paused。 • PHASE 标记的 Paused 位为 1。
	《非》 pausing	rung-condition-out 为 true。	《非》pausing	不采取任何操作。
后扫描	输出梯级条件设置为 false。		不采取任何操作。	

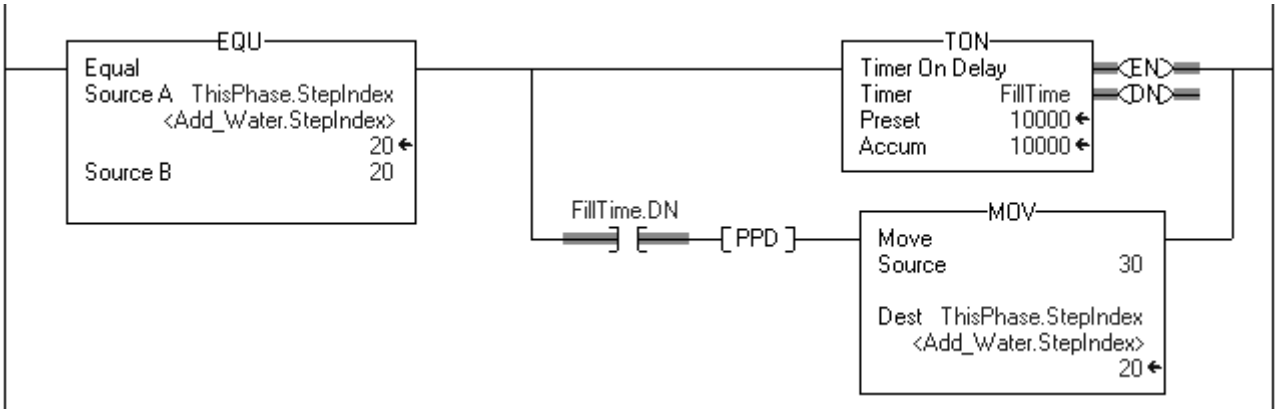
示例：

梯形图

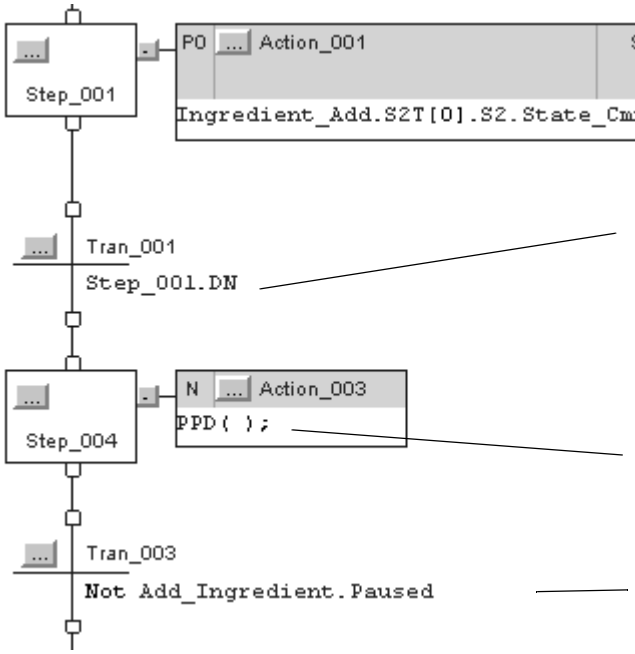
如果设备阶段处于 pausing 子状态
ThisPhase.StepIndex = 20 （例程位于步骤 20）
FillTime.DN = On

则 PPD 指令可防止 MOV 指令在设备阶段获得 resume 命令之前将例程移到步骤 30 （例程仍然位于步骤 20）。

如果设备阶段获得 resume 命令，则 PPD 指令允许 MOV 指令执行，因此例程进入步骤 30。



结构化文本



如果 *Step_001.DN* = 1 （此步骤在预设置时运行），则
Ingredient_Add…State_Cmnd = 0 （停止添加原料）
SFC 进入 *Step_004*

SFC 进入 *Step_004* 时
PPD 指令将 *Add_Ingredient.Paused* 设置为 1
如果 *Add_Ingredient.Paused* = 1，则 SFC 仍然停留在 *Step_004*。

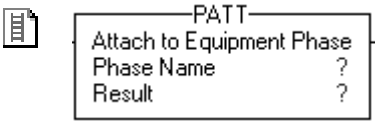
设备阶段获取 resume 命令时
Add_Ingredient.Paused = 0
SFC 进入下一步骤

附加到设备阶段 (PATT)

使用 PATT 指令获取设备阶段的所属权，以实现以下任一目的：

- 防止其他程序或 RSBizWare Batch 软件命令设备阶段
- 确认其他程序或 RSBizWare Batch 软件还未 拥有设备阶段

操作数：



梯形图

操作数：	类型：	格式：	说明：
阶段名称	阶段	设备阶段的名称	希望拥有的设备阶段
结果	DINT	直接标记	要使指令返回其成功 / 失败的代码，请输入一个 DINT 标记，该标记存储结果代码。 否则，输入 0。

```
PATT(Phase_Name,
Result);
```

结构化文本

操作数与梯形图 PATT 指令的操作数一样。

说明： 使用 PATT 指令可使程序获得设备阶段的所属权。

- 所属权是可选的。只要设备阶段还没有所有者，任何定序程序（控制器中的程序、 RSBizWare Batch 软件）都可以命令设备阶段。
- RSBizWare Batch 软件始终具有设备阶段的所属权。
- 定序程序拥有某个设备阶段时，其他定序程序都不能命令该设备阶段。

PATT 指令的使用准则

<table>
<tr><th>准则：</th><th>说明：</th></tr>
<tr><td>❑ 如果多个定序程序使用某个公用设备阶段，则考虑使用所属权。</td><td>所属权可确保某个程序可以命令其所有所需设备阶段，并禁止任何其他定序程序命令这些设备阶段。
<table><tr><th>如果：</th><th>则：</th></tr><tr><td>将 RSBizWare Batch 软件也用于在此控制器中运行序列</td><td rowspan="2">执行序列 （过程）之前，获取序列使用的所有设备阶段的所属权。</td></tr><tr><td>使用多个程序命令同一设备阶段</td></tr><tr><td>以上皆否</td><td>没有必要拥有设备阶段。</td></tr></table></td></tr>
<tr><td>❑ 请注意， RSLogix 5000 软件会重写控制器。</td><td>无论程序或 RSBizWare Batch 软件是否拥有某个设备阶段，您都可以使用 RSLogix 5000 软件重写所属权并命令该设备阶段变为其他状态。
<table><tr><th>软件或程序：</th><th>重写：</th></tr><tr><td>RSLogix 5000 软件</td><td>控制器 （内部定序程序）
RSBizWare Batch 软件 （外部定序程序）</td></tr><tr><td>控制器 （内部定序程序）</td><td>无</td></tr><tr><td>外部定序程序 （RSBizWare Batch 软件）</td><td>无</td></tr></table></td></tr>
<tr><td>❑ 使用 Result 操作数验证所属权。</td><td>使用 Result 操作数获取显示 PATT 指令成功 / 失败的代码。要解释结果代码，请参见第 A-44 页的 “PATT 结果代码”</td></tr>
<tr><td>❑ 避免或应对 24582 结果代码。</td><td>每次执行时， PATT 指令都尝试获取设备阶段的所属权。程序一旦获得设备阶段，再执行 PATT 指令就会生成一个结果代码 24582。使用 PATT 指令时，请执行以下任一操作：
• 将指令的执行限制于单次扫描，以避免生成 24582 结果代码。
• 在所属权条件中包含结果代码 24582。请参见第 A-45 页的示例。</td></tr>
<tr><td>❑ 序列完成后，释放所属权。</td><td>要释放所属权，请使用脱离设备阶段 (PDET) 指令。请参见第 A-47 页。</td></tr>
</table>

PATT 结果代码

如果指定一个标记来存储 PATT 指令的结果，则该指令在执行时返回以下代码之一：

代码 (Dec)：	说明：
0	命令成功。
24579	RSLogix 5000 软件已拥有设备阶段。 • 此程序现在也拥有设备阶段。 • 由于 RSLogix5000 软件的优先级比程序高，因此程序遣荒命令设备阶段。
24582	程序已拥有设备阶段。
24593	下面某个软件或程序已拥有该设备阶段。 • 外部定序程序 （RSBizWare Batch 软件） • 控制器中的其他程序
24594	设备阶段被禁止、未调度或处于被禁止的任务中。

算术状态标志： 不受影响

故障条件： 无

执行：

条件：	梯形图操作：	结构化文本操作：
预扫描	输出梯级条件设置为 false。	不采取任何操作。
输入梯级条件为 false	输出梯级条件设置为 false。	不可用
输入梯级条件为 true	• 指令执行。 • 输出梯级条件设置为 true。	不可用
结构化文本的扫描	不可用	在结构化文本中，指令在每次扫描时都会执行。要限制指令的扫描，请使用 SFC 操作的限定符和 / 或结构化文本结构。
指令执行	指令尝试获取指定的设备阶段的所属权。	指令尝试获取指定的设备阶段的所属权。
后扫描	输出梯级条件设置为 false。	不采取任何操作。

示例： 梯形图

如果 *Step.1* = 1 （序列的第一步），则

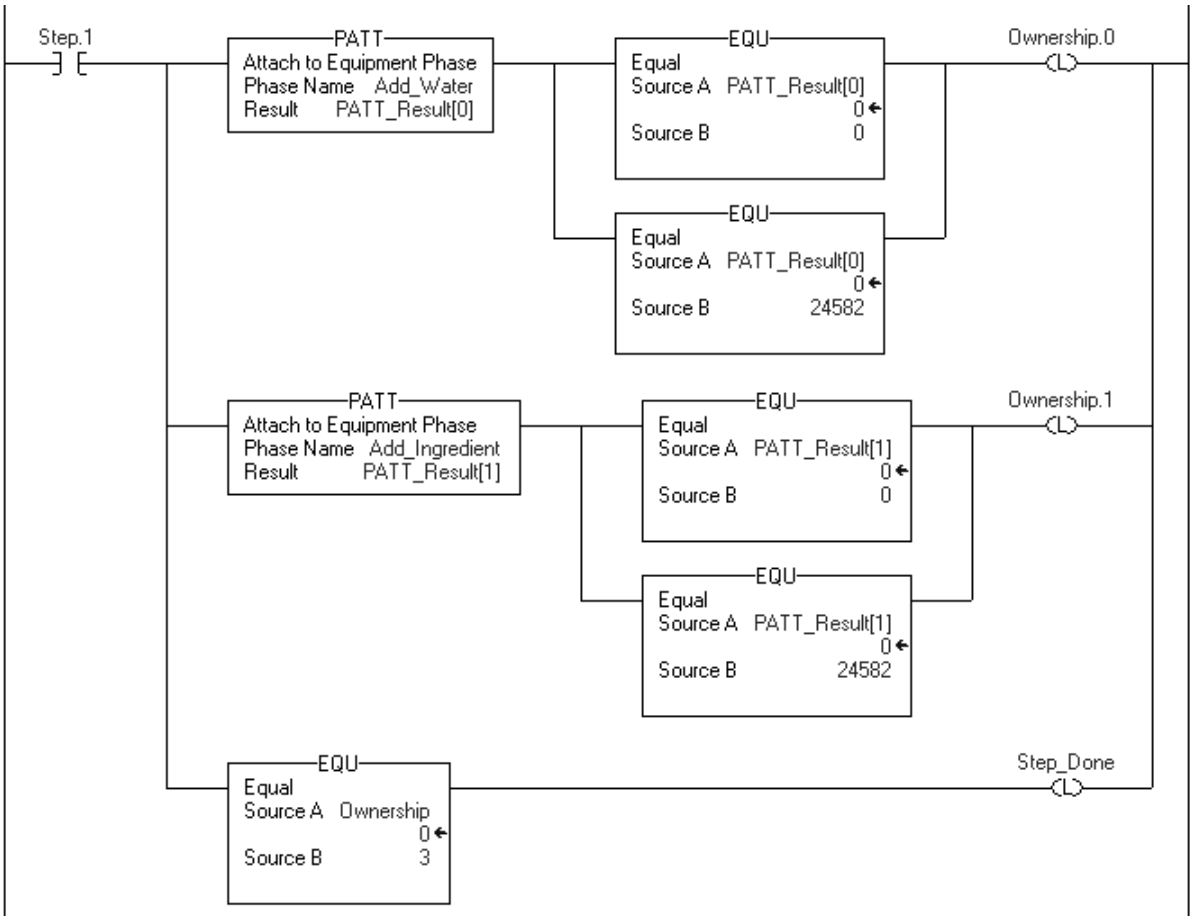
每个 PATT 指令都尝试获取某个设备阶段的所属权。

如果 PATT 指令的执行结果为 0 或 24582 （程序拥有该设备阶段），则

Ownership 标记中的一位为 1。（在 *Ownership* 标记中，每个设备阶段都有一个赋了值的位。）

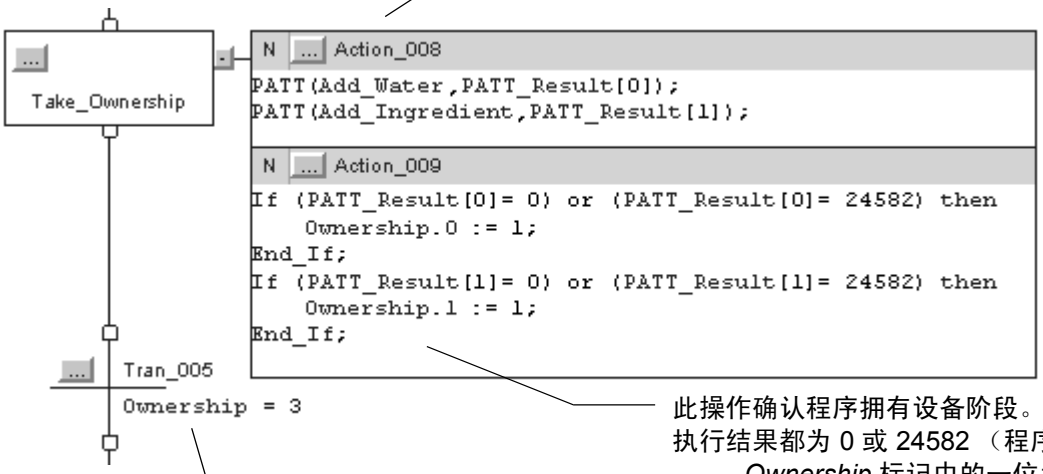
如果 *Ownership* = 3 （程序拥有位 0 和 1 所对应的设备阶段），则：

Done = 1 （此值通知序列进入下一步骤）。



结构化文本

在序列的第一步，此操作尝试获取序列使用的两个设备阶段的所属权。



此操作确认程序拥有设备阶段。如果每个 PATT 指令的执行结果都为 0 或 24582 （程序拥有该设备阶段），则 *Ownership* 标记中的一位为 1。（在 *Ownership* 标记中，每个设备阶段都有一个赋了值的位。）

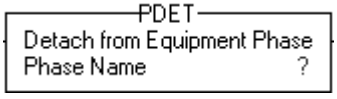
如果 *Ownership* = 3（程序拥有位 0 和 1 对应的设备阶段），则 SFC 进入下一步。

脱离设备阶段 (PDET)

使用 PDET 指令释放设备阶段的所属权。

操作数：





梯形图

操作数：	类型：	格式：	说明：
阶段名称	阶段	设备阶段的名称	希望 《不再 》拥有的设备阶段



```
PDET(Phase_Name);
```

结构化文本

操作数与梯形图 PDET 指令的操作数一样。

说明： 程序在执行 PDET 指令之后不再 拥有设备阶段。这样可释放设备阶段，以供其他程序或 RSBizWare Batch 软件拥有。仅当程序先前是通过附加到设备阶段 (PATT) 指令取得设备阶段的所属权时，才使用 PDET 指令。

算术状态标志： 不受影响

故障条件： 无

执行：

条件：	梯形图操作：	结构化文本操作：
预扫描	输出梯级条件设置为 false。	不采取任何操作。
输入梯级条件为 false	输出梯级条件设置为 false。	不可用
输入梯级条件为 true	• 指令执行。 • 输出梯级条件设置为 true。	不可用
结构化文本的扫描	不可用	在结构化文本中，指令在每次扫描时都会执行。要限制指令的扫描，请使用 SFC 操作的限定符和 / 或结构化文本结构。
指令执行	指令尝试释放指定的设备阶段的所属权。	指令尝试释放指定的设备阶段的所属权。
后扫描	输出梯级条件设置为 false。	不采取任何操作。

示例：

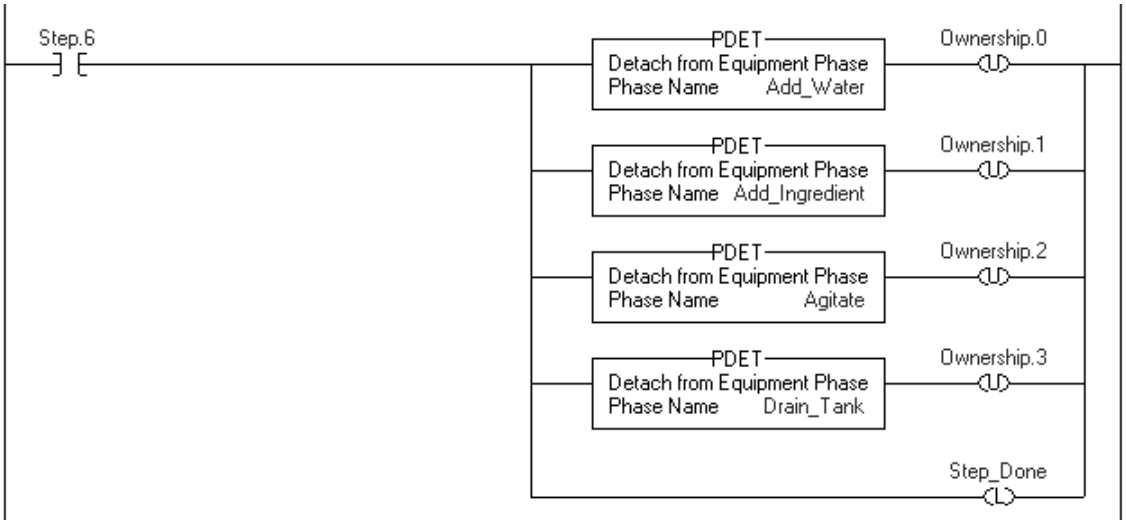
梯形图

如果 *Step.6* = 1 （序列的步骤 6），则

每个 PDET 指令都释放序列拥有的设备阶段的所属权。

每个 *Ownership* 位都为 0。（在 *Ownership* 标记中，每个设备阶段都有一个赋了值的位。）

Done = 1 （此值通知序列进入下一步）。

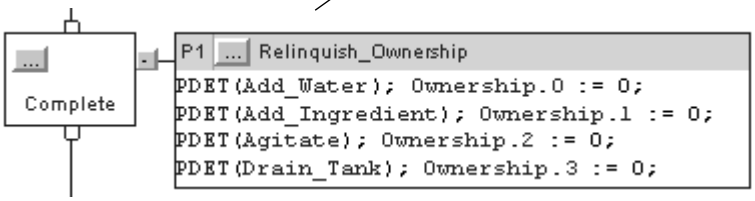


结构化文本

序列完成后，此操作：

- 释放设备阶段的所属权
- 清除所属权标志 （SFC 获得设备阶段时设置的位）

P1 限定符将此操作限制于步骤的第一次扫描。



PHASE 数据类型

PHASE 数据类型的使用

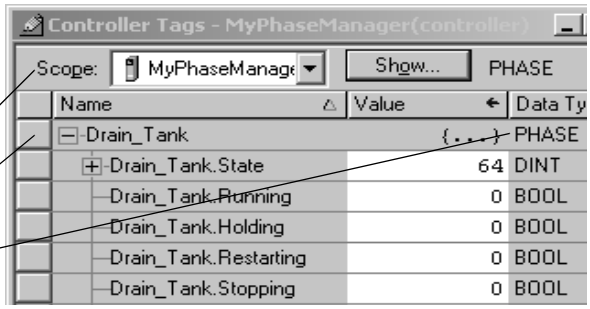
PHASE 数据类型提供有关设备阶段的状态信息。

创建设备阶段时， RSLogix 5000 软件会针对设备阶段的状态创建一个标记。

控制器范围

name = *phase_name*

PHASE 数据类型

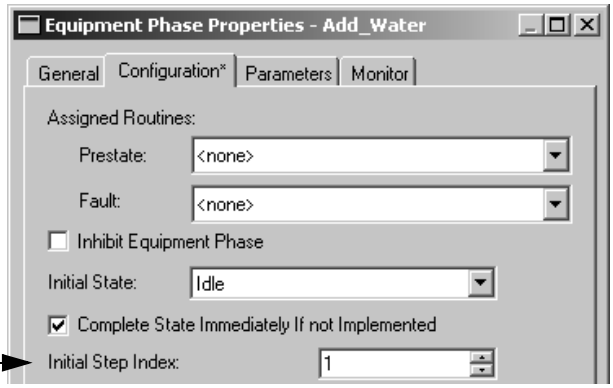


设置和清除设备阶段标记值

对于大多数 PHASE 数据类型的成员，您对它们的值只能进行监视。您只能控制下列成员：

成员	控制方法
StepIndex	如果在梯形图或结构化文本中以步骤序列的形式编写设备阶段，请使用 StepIndex 值作为步骤编号或位值。（SFC 自动按顺序执行步骤。） • 要初始化 StepIndex 值，请使用设备阶段的配置属性：

当设备阶段发生 idle → running 转换时，StepIndex = Initial Step



• 要前进到下一步，请编写用来递增 StepIndex 值的逻辑 （例如 MOV、 MUL、 OTL、 :=）

成员	控制方法	
Failure	**如果需要：**	**请使用指令：**
	设置失效值	设备阶段失效 (PFL)
	清除失效值	设备阶段清除失效 (PCLF)
NewInputParameters	要清除 NewInputParameters 位，请使用“设备阶段新参数”(PRNP) 指令。	
Producing	请使用位级指令或赋值指令设置或清除此位 （例如 OTE、 :=）。	
Standby	请使用位级指令或赋值指令设置或清除此位 （例如 OTE、 :=）。	

PHASE 数据类型

如果需要：	检查成员：	数据类型：	说明：	
使用一个成员监视设备阶段的状态	State	DINT	只读	
			对于以下状态：	**对应的位：**
			Running	0
			Holding	1
			Restarting	2
			Stopping	3
			Aborting	4
			Resetting	5
			Idle	6
			Held	7
			Complete	8
			Stopped	9
			Aborted	10
查看设备阶段是否处于 running 状态	Running	BOOL	只读	
查看设备阶段是否处于 holding 状态	Holding	BOOL	只读	
查看设备阶段是否处于 restarting 状态	Restarting	BOOL	只读	
查看设备阶段是否处于 stopping 状态	Stopping	BOOL	只读	
查看设备阶段是否处于 aborting 状态	Aborting	BOOL	只读	
查看设备阶段是否处于 resetting 状态	Resetting	BOOL	只读	
查看设备阶段是否处于 idle 状态	Idle	BOOL	只读	
查看设备阶段是否处于 held 状态	Held	BOOL	只读	
查看设备阶段是否处于 complete 状态	Complete	BOOL	只读	
查看设备阶段是否处于 stopped 状态	Stopped	BOOL	只读	

如果需要：	检查成员：	数据类型：	说明：
查看设备阶段是否处于 aborted 状态	Aborted	BOOL	只读
使用一个成员监视设备阶段的子状态	Substate	DINT	只读 **对于以下子状态：** **对应的位：** Pausing 0 Paused 1 AutoPause 2
查看设备阶段是否处于 pausing 子状态	Pausing	BOOL	只读
查看设备阶段是否处于 paused 子状态	Paused	BOOL	只读
查看设备阶段是否处于 auto pause 子状态	AutoPause	BOOL	只读
使用整数值或整数的位顺序执行一系列步骤	StepIndex	DINT	• 要初始化 StepIndex 值，请使用设备阶段的配置属性。 • 要前进到下一步，请使用诸如 MOV、MUL 或 := 等逻辑来递增 StepIndex 值。
标记设备阶段的特定异常 （故障、失效、非正常状况等）	Failure	DINT	**如果需要：** **请使用：** 设置失效值 PFL 指令 清除失效值 PCLF 指令
查找设备阶段的单元 ID	UnitID	DINT	RSBizWare Batch 软件会设置此值。
监视设备阶段的所属权	Owner	DINT	只读
查看是否正在通过 PXRQ 指令处理外部请求	PendingRequest	DINT	• 只读 • 从第 0 位开始，每一位都表示某特定请求的状态。位顺序遵循以下所示的与特定请求对应的成员的顺序。
查看是否正在通过 PXRQ 指令处理“下载输入参数”请求	DownloadInputParameters	BOOL	只读
查看是否正在通过 PXRQ 指令处理“下载输入参数子集”请求	DownloadInputParametersSubset	BOOL	只读
查看是否正在通过 PXRQ 指令处理“上载输出参数”请求	UploadOutputParameters	BOOL	只读
查看是否正在通过 PXRQ 指令处理“上载输出参数子集”请求	UploadOutputParameters Subset	BOOL	只读
查看是否正在通过 PXRQ 指令处理“下载输出参数限制”请求	DownloadOutput ParameterLimits	BOOL	只读
查看是否正在通过 PXRQ 指令处理“获取资源”请求	AcquireResources	BOOL	只读
查看是否正在通过 PXRQ 指令处理“释放资源”请求	ReleaseResources	BOOL	只读

如果需要：	检查成员：	数据类型：	说明：
查看是否正在通过 PXRQ 指令处理“向链接阶段发送消息”请求	SendMessageToLinkedPhase	BOOL	只读
查看是否正在通过 PXRQ 指令处理“向链接阶段发送消息并等待”请求	SendMessageToLinkedPhaseAndWait	BOOL	只读
查看是否正在通过 PXRQ 指令处理“从链接阶段接收消息”请求	ReceiveMessageFromLinkedPhase	BOOL	只读
查看是否正在通过 PXRQ 指令处理“取消向链接阶段发送消息”请求	CancelMessageToLinked Phase	BOOL	只读
查看是否正在通过 PXRQ 指令处理“向操作员发送消息”请求	SendMessageToOperator	BOOL	只读
查看是否正在通过 PXRQ 指令处理“清除向操作员发送的消息”请求	ClearMessageToOperator	BOOL	只读
查看是否正在通过 PXRQ 指令处理“生成电子签名”请求	GenerateESignature	BOOL	只读
查看是否正在通过 PXRQ 指令处理“下载批数据”请求	DownloadBatchData	BOOL	只读
查看是否正在通过 PXRQ 指令处理“下载使用的材料轨道数据容器”请求	DownloadMaterialTrackDataContainerInUse	BOOL	只读
查看是否正在通过 PXRQ 指令处理“下载容器绑定优先级”请求	DownloadContainerBindingPriority	BOOL	只读
查看是否正在通过 PXRQ 指令处理“下载充足材料”请求	DownloadSufficientMaterial	BOOL	只读
查看是否正在通过 PXRQ 指令处理“下载材料轨道数据库数据”请求	DownloadMaterialTrackDatabaseData	BOOL	只读
查看是否正在通过 PXRQ 指令处理“上载使用的材料轨道数据容器”请求	UploadMaterialTrackDataContainerInUse	BOOL	只读
查看是否正在通过 PXRQ 指令处理“上载容器绑定优先级”请求	UploadContainderBindingPriority	BOOL	只读
查看是否正在通过 PXRQ 指令处理“上载材料轨道数据库数据”请求	UploadMaterialTrackDatabaseData	BOOL	只读
查看逻辑是否已中止 PXRQ 指令	AbortingRequest	BOOL	只读
查看 RSBizWare Batch 软件是否有用于设备阶段的新参数	NewInputParameters	BOOL	• 只读 • RSBizWare Batch 软件存在用于设备阶段的新参数时会设置此位。 • 要清除 NewInputParameters 位，请使用 PRNP 指令。

如果需要：	检查成员：	数据类型：	说明：
初始化 producing 状态	Producing	BOOL	Logix5000 设备阶段没有 producing 状态。要创建 producing 状态，请使用 Producing 位。
初始化 standby 状态	Standby	BOOL	Logix5000 设备阶段没有 standby 状态。要创建 standby 状态，请使用 Standby 位。

说明：

配置设备阶段

目的

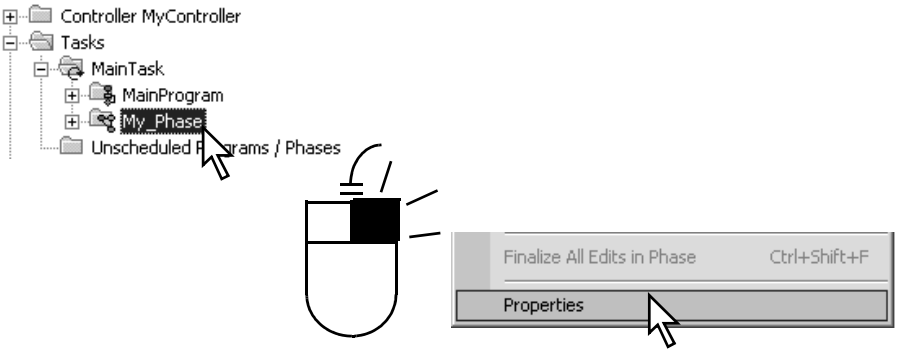
本附录用来引导您完成配置设备阶段设置的步骤。

适用情形

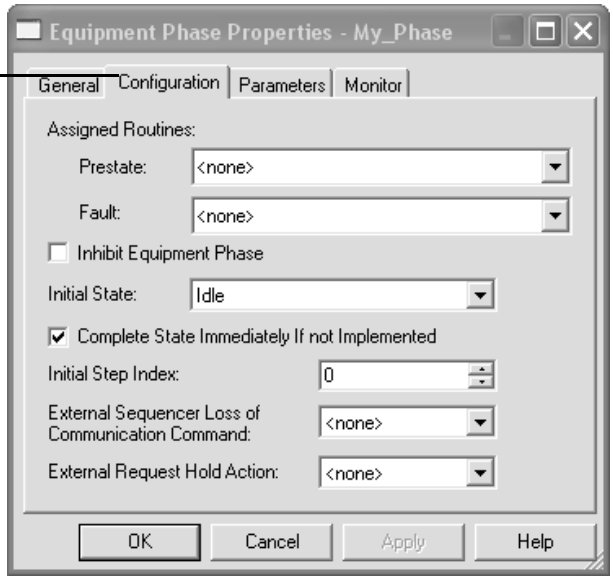
当您要更改设备阶段的默认设置时，请使用本附录。

打开设备阶段的配置

1.

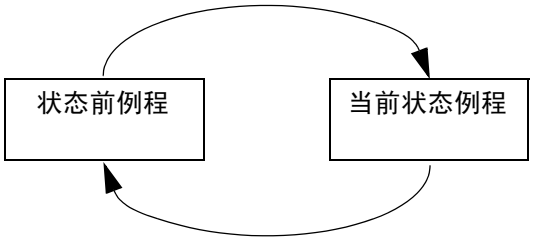


2.



配置设备阶段

请使用以下设置配置设备阶段。

设置	选择
1. Prestate （状态前）	 状态前例程始终处于运行中，即便在设备阶段处于 idle 状态时也不例外。它于 *每次* 状态扫描前运行。 是否要运行状态前例程？ • 是 — 选择要运行的例程。 • 否 — 将此框保留为 <none>
2. Fault （故障）	故障例程可用来清除由指令导致的重大故障。 是否要为此设备阶段中的指令设置故障例程？ • 是 — 选择要作为故障例程的例程。 • 否 — 将此框保留为 <none>
3. Inhibit Equipment Phase （禁止设备阶段）	是否让控制器运行此设备阶段？ • 是 — 将此框保留为非选中状态，或者取消选中此框。 • 否 — 选中此框。
4. Initial State （初始状态）	打开控制器时希望设备阶段进入何种状态？ • idle • complete • stopped • aborted
5. Complete State Immediately If not Implemented （立即完成没有实现的状态）	是否让设备阶段跳过所有未使用的状态？ • 是 — 将此框保留为选中状态，或者选中此框。 • 否 — 取消选中此框。
6. 初始步骤索引	A. 梯形图或结构化文本中是否存在任何状态例程？ • 否 — 跳过此框。 • 是 — 转到步骤 B。 B. 是否有任何状态例程使用步骤编号？ • 是 — 键入每个状态的第一个步骤的编号。 • 否 — 跳过此框。 设备阶段的标记有一个 StepIndex 编号。在设备阶段每次更改状态时，控制器都重置 StepIndex。控制器将 StepIndex 重置为您在 “Initial Step Index”（初始步骤索引）框中输入的编号。

设置	选择
7. External Sequencer Loss of Communication Command （通讯命令丢失外部定序程序）	A. 您是否使用 RSBizWare Batch 软件来对此设备阶段发出命令？ • 否 — 跳过此框。 • 是 — 转到步骤 B。 B. 如果控制器丢失与 RSBizWare Batch 软件的通讯，您希望设备阶段执行何种操作？ • 继续保留在当前状态下 — 选择 <none>。 • 进入 aborting 状态 — 选择 Abort。 • 进入 holding 状态 — 选择 Hold。 • 进入 stopping 状态 — 选择 Stop。 设备阶段必须依然遵循状态模型。例如，当通讯失败时，它只有处于 running 或 restarting 状态时才能进入 holding 状态。
8. External Request Hold Action （外部请求保持动作）	A. 您是否使用了任何 PXRQ 指令？ • 否 — 跳过此框。 • 是 — 转到步骤 B。 B. 如果设备阶段在 PXRQ 指令正在执行时进入 holding 状态，您希望执行何种操作？ • 不执行任何操作 — 选择 <none>。 • 停止请求 — 选择 Clear （清除）。

说明：

A

aborted 状态
使用 1-5
aborting 状态
使用 1-5, 3-12

C

complete 状态
使用 1-5

H

held 状态
使用 1-5
holding 状态
使用 1-5, 3-12

I

idle 状态
使用 1-5

P

PATT 指令 A-42
PCLF 指令 A-21
PCMD 指令 A-8
PDET 指令 A-47
PFL 指令 A-17
使用 3-15
PHASE 数据类型
成员 B-2
设置或清除值 B-1
使用状态位 3-14
POVR 指令 A-13
PPD 指令 A-37
PRNP 指令 A-34
producing 状态
创建 3-4
PSC 指令 A-5
PXRQ 指令 A-23
保持动作 C-3
缺失的通讯 C-3

R

resetting 状态
使用 1-5
restarting 状态
使用 1-5, 3-12
RSBizWare Batch 软件
报告 A-23
外部请求 A-23
RSLogix 5000 软件
给出命令 2-3
监视设备阶段 1-7, 2-3
所属权 1-7
running 状态
使用 1-5

S

standby 状态
创建 3-4
stopped 状态
使用 1-5
stopping 状态
使用 1-5

报

报告
发送 A-23

参

参数
下载或上载 A-23, A-34

测

测试
设备阶段 2-3

程

程序
设备阶段 3-9

初

初始步骤索引
设置 C-2
初始状态
设置 2-6
选择 3-4

创

创建
阶段状态例程 2-2
设备阶段 2-2, 3-2
转换 3-18
状态 3-4

定

定序程序
示例 3-21

断

断点
请参见 PPD 指令

附

附加到设备阶段指令 A-42

故

故障
处理 3-12
示例 3-16, 3-17, 3-23

监

监视
设备阶段 2-3

建

建立
设备接口标记 3-26

阶

阶段
请参见设备阶段
阶段状态例程
添加 2-2
阶段状态完成指令 A-5

禁

禁止
设备阶段 C-2

控

控制转移 A-35

例

例程
添加阶段状态例程 2-2

命

命令
给出 1-6
使用 A-8
示例 3-21, 3-22
通过 RSLogix 5000 软件提供 2-3
用 PCMD 指令发出 3-18

默

默认
设置失效代码 3-15

配

配置
设备阶段 C-1

清

清除
PHASE 标记值 B-1

设

设备程序
创建代码 3-9
建立数据 3-26
接口标记 3-26
使用 3-9
设备阶段
start 3-21, 3-22
编号 3-2
参数 A-23, A-34
测试状态 2-3
初始状态 2-6
处理故障 3-12
创建 2-2, 3-2
创建代码 3-9
创建阶段状态例程 2-2
定义状态 3-4
概述 1-1
故障 3-12
获取所属权 A-42
监视 2-3
监视器 1-7
阶段状态例程 2-2
禁止 C-2
配置 C-1
设计 3-2
设置初始步骤索引 C-2
设置初始状态 2-6
设置断点 A-37

设置或清除标记值 B-1
设置失效代码 3-15
设置状态前例程 3-14
使用 3-2
使用一个命令 A-8
释放所属权 A-47
数据类型 B-2
所属权 A-42, A-47
添加状态前例程 3-13
下载或上载参数 A-23
新输入参数位 A-34
与 PackML 相比 1-9
与 S88 相比 1-9
暂停 A-37
指令 1-1
重写命令 3-20
状态 1-4
设备阶段命令指令 A-8
设备阶段清除失效指令 A-21
设备阶段失效指令 A-17
设备阶段外部请求指令 A-23
设备阶段新参数指令 A-34
设备阶段暂停指令 A-37
设备阶段指令
PATT A-42
PCLF A-21
PCMD A-8
PDET A-47
PFL A-17
POVR A-13
PPD A-37
PRNP A-34
PSC A-5
PXRQ A-23
概述 1-1
设备阶段重写命令指令 A-13
设备模型
请参见设备程序
设置
PHASE 标记值 B-1
PXRQ 指令的保持动作 C-3
初始步骤索引 C-2

失

失效代码
清除 A-21
设置 A-17

示

示例 A-35
处理超时 3-17
处理堵塞 A-16
处理故障 A-20
处理装置故障 3-16
处理阻塞 3-23
定序设备阶段 3-21
分离罐的代码 3-10
分离机器的代码 3-11
给出一个命令 A-11
罐的过程 3-21
罐的接口标记 3-28
罐的设备阶段 3-3
罐的状态模型 3-7
获取几个阶段的所属权 A-45
获取结果代码 A-12
机器的接口标记 3-29
机器的设备阶段 3-3
机器的状态模型 3-8
机器已完成 resetting 3-25
启动机器 3-22
清除失效代码 A-22
设置断点 A-41
使用 PXRQ 指令获取资源 A-33
使用失效代码 A-20
释放所属权 A-48
通知状态为 done A-7
向罐中注水已完成 3-25
重写所有者 A-16

所

所属权
采用 RSLogix 5000 软件 1-7
概述 1-7

添

添加
阶段状态例程 2-2
设备阶段 2-2

脱

脱离设备阶段指令 A-47

外

外部请求
保持动作 C-3
响应缺失的通讯 C-3

异

异常
处理 3-12

重

重写命令
示例 3-23

转

转换
完成时 3-24
逐一进入 2-3

状

状态
标记为已完成 3-24
处理异常 3-12
概述 1-4
设置初始状态 2-6
设置转换 3-18
使用 1-5, 3-4
使用状态位 3-14
与 PackML 相比 1-9
与 S88 相比 1-9
在完成时转换 3-24
逐步进行 1-7
逐一进入 2-3
状态间转换 3-18
状态转换 1-6
状态例程
参见阶段状态例程
状态模型
请参见状态
状态前例程
概述 3-12
使用 3-12
示例 3-16, 3-17, 3-22, 3-23
添加 3-13
指定 3-14

How Are We Doing?

Your comments on our technical publications will help us serve you better in the future. Thank you for taking the time to provide us feedback.

You can complete this form and mail (or fax) it back to us or email us at RADocumentComments@ra.rockwell.com

Pub. Title/Type PhaseManager

Cat. No. _____ Pub. No. LOGIX-UM001A-ZH-P Pub. Date 5 '¬ 2005ƒÍ Part No. 123456-04

Please complete the sections below. Where applicable, rank the feature (1=needs improvement, 2=satisfactory, and 3=outstanding).

Overall Usefulness	1	2	3	How can we make this publication more useful for you?
Completeness (all necessary information is provided)	1	2	3	Can we add more information to help you? procedure/step illustration feature example guideline other explanation definition
Technical Accuracy (all provided information is correct)	1	2	3	Can we be more accurate? text illustration
Clarity (all provided information is easy to understand)	1	2	3	How can we make things clearer?
Other Comments				You can add additional comments on the back of this form.

Your Name _____

Your Title/Function _____

Location/Phone _____

Would you like us to contact you regarding your comments?

___No, there is no need to contact me

___Yes, please call me

___Yes, please email me at _____

___Yes, please contact me via _____

Return this form to: Rockwell Automation Technical Communications, 1 Allen-Bradley Dr., Mayfield Hts., OH 44124-9705

Fax: 440-646-3525 Email: RADocumentComments@ra.rockwell.com

PLEASE FASTEN HERE (DO NOT STAPLE)

Other Comments

PLEASE FOLD HERE

PLEASE REMOVE

NO POSTAGE
NECESSARY
IF MAILED
IN THE
UNITED STATES

BUSINESS REPLY MAIL
FIRST-CLASS MAIL PERMIT NO. 18235 CLEVELAND OH

POSTAGE WILL BE PAID BY THE ADDRESSEE

Allen-Bradley
RELIANCE ELECTRIC
DODGE
ROCKWELL SOFTWARE
Rockwell Automation

1 ALLEN-BRADLEY DR
MAYFIELD HEIGHTS OH 44124-9705

罗克韦尔自动化支持

罗克韦尔自动化在网站上提供可帮助您使用其产品的技术信息。您可以在 http://support.rockwellautomation.com 上找到技术手册、常见问题解答知识库、技术和应用程序说明、示例代码以及指向软件服务包的链接，另外还可以利用 MySupport功能执行自定义设置，从而充分利用这些工具。

为了提供有关安装、配置和故障排除的更高一级的电话支持，我们提供了 TechConnect Support方案。有关更多信息，请联系当地分销商或 罗克韦尔自动化代表，或者访问http://support.rockwellautomation.com。

安装帮助

如果您在安装后的 24小时之内遇到硬件模块问题，请查看本手册中提供的信息。您也可以拨打入门帮助客户支持专线，以修复模块问题并使之运行：：

美国	1.440.646.3223 周一 – 周五， 8am – 5pm EST
美国以外的国家 / 地区	有关任何技术支持问题，请联系当地 罗克韦尔自动化代表。

新产品退货

罗克韦尔自动化会对其所有产品进行测试，以保证其产品在出厂时能够全面正常运行。但是，如果您因产品不能使用而需要退货：

美国	联系当地分销商。您必须向分销商提供客户支持帐户编号 （参见上面的电话号码来获取编号），以便完成退货程序。
美国以外的国家 / 地区	请联系当地 罗克韦尔自动化代表，了解退货手续。

售后技术支持：1080 0610 0327

www.rockwellautomation.com.cn

动力、控制与信息解决方案

Americas: Rockwell Automation, 1201 South Second Street, Milwaukee, WI 53204-2496 USA, Tel: (1)414 382.2000, Fax: (1)414 382.4444

亚太地区 – 香港数码港道100号数码港3座F区14楼 电话：(852)28874788 传真：(852)25109436

北京 – 北京市建国门内大街18号恒基中心办公楼1座4层 邮编：100005 电话：(8610)65182535 传真：(8610)65182536
上海 – 上海市仙霞路319号远东国际广场A幢7楼 邮编：200051 电话：(8621)61206007 传真：(8621)62351099
厦门 – 厦门市湖里工业区悦华路38号 邮编：361006 电话：(86592)6022084 传真：(86592)6021832
沈阳 – 沈阳市沈河区青年大街219号华新国际大厦15-F 单元 邮编：110015 电话：(8624)23961518 传真：(8624)23963539
武汉 – 武汉市建设大道568号新世界国贸大厦I座2202室 邮编：430022 电话：(8627)68850233 传真：(8627)68850232
广州 – 广州市环市东路362号好世界广场2703-04室 邮编：510060 电话：(8620)83849977 传真：(8620)83849989
重庆 – 重庆市渝中区邹容路68号大都会商厦3112-13室 邮编：400010 电话：(8623)63702668 传真：(8623)63702558
大连 – 大连市西岗区中山路147号森茂大厦2305层 邮编：116011 电话：(86411)83687799 传真：(86411)83679970
西安 – 西安市南大街30号中大国际大厦712室 邮编：710002 电话：(8629)87203577 传真：(8629)87203123
深圳 – 深圳市深南东路5047号深圳发展银行大厦15L 邮编：518001 电话：(86755)25847099 传真：(86755)25870900
南京 – 南京市中山南路49号商茂世纪广场44楼A3-A4 座 邮编：210005 电话：(8625)86890445 传真：(8625)86890142
青岛 – 青岛市香港中路36号新世界数码港招银大厦1006室 邮编：266071 电话：(86532)86678338 传真：(86532)86678339
成都 – 成都市总府路2号时代广场A座906室 邮编：610016 电话：(8628)86726886 传真：(8628)68726887

Publication LOGIX-UM001A March 2006

Copyright 2006 Rockwell Automation Inc. Printed in CHINA.

Allen-Bradley

PhaseManager

用户手册