

# SIEMENS

前言

---

使用 S7-1200 运动控制

1

## SIMATIC

S7-1200 运动控制

---

2

## STEP 7 S7-1200 Motion Control V13 SP1

功能手册




11/2014

A5E03790555-AD

## 法律资讯

### 警告提示系统

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

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

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


### 合格的专业人员

本文件所属的产品/系统只允许由符合各项工作要求的**合格人员**进行操作。其操作必须遵照各自附带的文件说明，特别是其中的安全及警告提示。

由于具备相关培训及经验，合格人员可以察觉本产品/系统的风险，并避免可能的危险。

### 按规定使用Siemens 产品

请注意下列说明：

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

### 商标

所有带有标记符号®的都是西门子股份有限公司的注册商标。本印刷品中的其他符号可能是一些其他商标。若第三方出于自身目的使用这些商标，将侵害其所有者的权利。

### 责任免除

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

# 前言

## 本手册用途

此文档可提供有关 S7-1200 运动控制的详细信息。此文档在内容和结构方面与 STEP 7 V13 SP1 在线帮助一致。能够与 STEP 7 进行交互是理解本文档中大部分内容的前提条件。

本文档适用于 STEP 7 程序员和运动控制应用自动化系统的组态、调试和维修领域的人员。

## 需要的基本知识

要理解本文档，需具备自动化工程与运动控制领域的常识。

也需要熟悉在 Windows 操作系统下使用计算机或编程设备的知识。

因为 S7-1200 运动控制是以 STEP 7 为基础的，所以您需要具备使用 STEP 7 的知识。

## 本手册适用范围

本手册适用于 STEP 7 V13 SP1。

## 约定

本文档中包含所述设备的相关图片。这些图片可能与实际提供的设备略有不同。

请同时遵循以下所标注的注意事项：

---

### 说明

这些注意事项包含有关本文档中所述产品、产品操作或应特别关注的文档部分的重要信息。

---

## 其它支持

如果您对本手册中所述的产品有任何疑问，而在本文档中未找到答案，请与我们联系当地办事处的 Siemens 合作伙伴联系。

可以在 Internet (<http://www.automation.siemens.com/mcms/aspa-db/en/automation-technology/Pages/default.aspx>) 上找到相应的联系人。

可在 Internet (<http://www.siemens.com/simatic-tech-doku-portal>) 上找到各种 SIMATIC 产品和系统的技术文档向导。

在互联网 (<https://mall.industry.siemens.com>) 上有在线的产品目录以及订货系统。

## 培训中心

我们提供了一系列课程来帮助您开始使用 S7 可编程控制器。

请联系您当地的培训中心或中央培训中心

(<http://support.automation.siemens.com/CN/view/zh/24486113>)。

## 安全信息

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

有鉴于此，西门子不断对产品和解决方案进行开发和完善。

西门子强烈建议您定期检查产品的更新和升级信息。

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

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

要及时了解有关产品的更新和升级信息，请订阅相关产品的实事信息。

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

# 目录

前言 .....	3
<b>1 使用 S7-1200 运动控制 .....</b>	<b>10</b>
1.1 简介 .....	10
1.1.1 CPU S7-1200 的运动功能 .....	10
1.1.2 用于运动控制的硬件组件 .....	11
1.2 使用 S7-1200 运动控制的基础知识 .....	15
1.2.1 通过 PTO 连接驱动器 .....	15
1.2.1.1 与运动控制相关的 CPU 输出 .....	15
1.2.1.2 脉冲接口的工作原理 .....	19
1.2.1.3 信号类型与行进方向之间的关系 .....	20
1.2.2 PROFIdrive/模拟驱动器接口 .....	24
1.2.2.1 驱动器和编码器连接 .....	24
1.2.2.2 PROFIdrive .....	25
1.2.2.3 闭环控制 .....	26
1.2.2.4 过程响应 .....	27
1.2.3 硬件和软件限位开关 .....	32
1.2.4 冲击限制 .....	33
1.2.5 回原点 .....	34
1.3 运动控制使用指南 .....	35
1.4 使用版本 .....	36
1.4.1 版本概述 .....	36
1.4.2 更改工艺版本 .....	39
1.4.3 变量兼容性列表 .....	40
1.4.4 限位开关状态 .....	45
1.5 定位轴工艺对象 .....	46
1.5.1 集成定位轴工艺对象 .....	46
1.5.2 定位轴工艺对象的工具 .....	50
1.5.3 添加一个定位轴工艺对象 .....	53
1.5.4 组态定位轴工艺对象 .....	54
1.5.4.1 使用组态对话框 .....	54
1.5.4.2 比较值 .....	55
1.5.4.3 基本参数 .....	56
1.5.4.4 扩展参数 .....	74
1.5.4.5 参数视图 .....	103

1.6	工艺对象命令表 .....	127
1.6.1	使用命令表工艺对象 .....	127
1.6.2	命令表工艺对象工具 .....	128
1.6.3	添加工艺对象命令表 .....	129
1.6.4	组态命令表工艺对象 .....	130
1.6.4.1	使用组态对话框 .....	130
1.6.4.2	比较值 .....	131
1.6.4.3	基本参数 .....	132
1.6.4.4	扩展参数 .....	150
1.7	下载到 CPU .....	153
1.8	调试 .....	155
1.8.1	轴控制面板 .....	155
1.8.2	调节 .....	159
1.9	编程 .....	162
1.9.1	运动控制语句概述 .....	162
1.9.2	创建用户程序 .....	163
1.9.3	编程注意事项 .....	166
1.9.4	断电和重新启动后运动控制命令的行为 .....	168
1.9.5	监视激活的命令 .....	169
1.9.5.1	监视激活的命令 .....	169
1.9.5.2	具有输出参数“Done”的运动控制指令 .....	169
1.9.5.3	运动控制指令 MC_MoveVelocity .....	173
1.9.5.4	运动控制指令 MC_MoveJog .....	177
1.9.6	运动控制语句的错误显示 .....	181
1.9.7	重新启动工艺对象 .....	182
1.10	轴 - 诊断 .....	183
1.10.1	状态和错误位（自工艺对象 V4 起） .....	183
1.10.2	运动状态 .....	187
1.10.3	动态设置 .....	188
1.10.4	PROFIdrive 帧 .....	189

1.11	附录 .....	190
1.11.1	将多个轴与同一个 PTO 一起使用 .....	190
1.11.2	将多个驱动器与同一个 PTO 一起使用 .....	193
1.11.3	在更高优先级等级（执行级别）中跟踪作业 .....	194
1.11.4	使用通过 PTO 的驱动器连接时的软限位开关时的特殊情况 .....	197
1.11.4.1	软件限位开关与回原点操作结合使用 .....	197
1.11.4.2	软件限位开关和软件限位开关的位置变化。 .....	201
1.11.4.3	软件限位开关与动态更改结合使用 .....	202
1.11.5	降低短时间定位的速度 .....	204
1.11.6	启动/停止速度的动态调整 .....	204
1.11.7	ErrorID 和 ErrorInfo 列表（自工艺对象 V4 起） .....	205
1.11.8	自 V4 起的定位轴工艺对象的变量 .....	235
1.11.8.1	自 V4 起的位置变量 .....	235
1.11.8.2	自 V4 起的速度变量 .....	236
1.11.8.3	自 V5 起的 ActualPosition 变量 .....	237
1.11.8.4	自 V5 起的 ActualVelocity 变量 .....	238
1.11.8.5	自 V4 起的 Actor 变量 .....	239
1.11.8.6	Sensor[1] 变量 .....	243
1.11.8.7	自 V4 起的单位变量 .....	255
1.11.8.8	自 V4 起的机械变量 .....	256
1.11.8.9	Modulo 变量 V5 及更高版本 .....	257
1.11.8.10	自 V4 起的 DynamicLimits 变量 .....	259
1.11.8.11	自 V4 起的 DynamicDefaults 变量 .....	260
1.11.8.12	自 V4 起的 PositionLimitsSW 变量 .....	262
1.11.8.13	自 V4 起的 PositionLimitsHW 变量 .....	264
1.11.8.14	自 V4 起的回原点变量 .....	267
1.11.8.15	PositionControl 变量 V5 及更高版本 .....	269
1.11.8.16	FollowingError 变量 V5 及更高版本 .....	270
1.11.8.17	PositionMonitoring 变量 V5 及更高版本 .....	272
1.11.8.18	StandstillSignal 变量 V5 及更高版本 .....	274
1.11.8.19	自 V4 起的 StatusPositioning 变量 .....	275
1.11.8.20	StatusDrive 变量 V5 及更高版本 .....	277
1.11.8.21	StatusSensor 变量 V5 及更高版本 .....	278
1.11.8.22	自 V4 起的 StatusBits 变量 .....	279
1.11.8.23	自 V4 起的 ErrorBits 变量 .....	285
1.11.8.24	自 V4 起的 ControlPanel 变量 .....	287
1.11.8.25	自 V4 起的内部变量 .....	287
1.11.8.26	工艺对象变量的更新 .....	287
1.11.9	自 V4 起的命令表工艺对象变量 .....	288
1.11.9.1	自 V4 起的 Command[1...32] 变量 .....	288

1.11.10	版本 V1...4.....	290
1.11.10.1	与运动控制相关的 CPU 输出（工艺版本 V1...3） .....	290
1.11.10.2	组态对话框 .....	295
1.11.10.3	诊断 - 状态和错误位（“轴”工艺对象 V1...3） .....	316
1.11.10.4	ErrorID 和 ErrorInfo .....	320
1.11.10.5	轴工艺对象 V1...3 的变量 .....	342
1.11.10.6	命令表工艺对象 V1...3 的变量 .....	366
<b>2</b>	<b>S7-1200 运动控制.....</b>	<b>368</b>
2.1	S7-1200 运动控制 V4 及更高版本 .....	368
2.1.1	MC_Power.....	368
2.1.1.1	MC_Power: 启用、禁用轴 V4 及更高版本 .....	368
2.1.1.2	MC_Power: V4 的功能图 .....	373
2.1.2	MC_Reset .....	374
2.1.2.1	MC_Reset: 确认故障 V4 及更高版本 .....	374
2.1.3	MC_Home .....	377
2.1.3.1	MC_Home: 使轴归位, 设置参考点 V4 及更高版本.....	377
2.1.4	MC_Halt .....	382
2.1.4.1	MC_Halt: 停止轴 V4 及更高版本.....	382
2.1.4.2	MC_Halt: V4 的功能图 .....	385
2.1.5	MC_MoveAbsolute .....	386
2.1.5.1	MC_MoveAbsolute: 轴的绝对定位 V4 及更高版本.....	386
2.1.5.2	MC_MoveAbsolute: V4 的功能图.....	389
2.1.6	MC_MoveRelative .....	391
2.1.6.1	MC_MoveRelative: 轴的相对定位 V4 及更高版本.....	391
2.1.6.2	MC_MoveRelative: V4 的功能图.....	394
2.1.7	MC_MoveVelocity.....	396
2.1.7.1	MC_MoveVelocity: 以预设的旋转速度移动轴 V4 及更高版本 .....	396
2.1.7.2	MC_MoveVelocity: V4 的功能图 .....	400
2.1.8	MC_MoveJog .....	402
2.1.8.1	MC_MoveJog: 在点动模式下移动轴 V4 及更高版本 .....	402
2.1.8.2	MC_MoveJog: V4 的功能图 .....	405
2.1.9	MC_CommandTable .....	406
2.1.9.1	MC_CommandTable: 按照运动顺序运行轴命令 V4 及更高版本 .....	406
2.1.10	MC_ChangeDynamic .....	409
2.1.10.1	MC_ChangeDynamic: 更改轴的动态设置 V4 及更高版本.....	409
2.1.11	MC_ReadParam.....	413
2.1.11.1	MC_ReadParam: 连续读取定位轴的运动数据 V4 及更高版本.....	413
2.1.12	MC_WriteParam.....	416
2.1.12.1	MC_WriteParam: 写入定位轴的变量 V4 及更高版本.....	416



2.2	S7-1200 运动控制 V1...3 .....	419
2.2.1	MC_Power.....	419
2.2.1.1	MC_Power: 启用、禁用轴 V1...3 .....	419
2.2.1.2	MC_Power: 功能图 V1...3 .....	424
2.2.2	MC_Reset.....	425
2.2.2.1	MC_Reset: 应答故障 V1...3 .....	425
2.2.3	MC_Home .....	428
2.2.3.1	MC_Home: 使轴回原点, 设置参考点 V1...3 .....	428
2.2.4	MC_Halt.....	433
2.2.4.1	MC_Halt: 停止轴 V1...3 .....	433
2.2.4.2	MC_Halt: 功能图 V1...3 .....	435
2.2.5	MC_MoveAbsolute .....	436
2.2.5.1	MC_MoveAbsolute: 轴的绝对定位 V1...3 .....	436
2.2.5.2	MC_MoveAbsolute: 功能图 V1...3.....	439
2.2.6	MC_MoveRelative .....	441
2.2.6.1	MC_MoveRelative: 轴的相对定位 V1...3 .....	441
2.2.6.2	MC_MoveRelative: 功能图 V1...3.....	444
2.2.7	MC_MoveVelocity.....	446
2.2.7.1	MC_MoveVelocity: 以预设的旋转速度移动轴 V1...3.....	446
2.2.7.2	MC_MoveVelocity: 功能图 V1...3 .....	449
2.2.8	MC_MoveJog .....	451
2.2.8.1	MC_MoveJog: 在点动模式下移动轴 V1...3 .....	451
2.2.8.2	MC_MoveJog: 功能图 V1...3.....	454
2.2.9	MC_CommandTable.....	455
2.2.9.1	MC_CommandTable: 按照运动顺序运行轴命令 V2...3 .....	455
2.2.10	MC_ChangeDynamic.....	458
2.2.10.1	MC_ChangeDynamic: 更改轴的动态设置 V2...3 .....	458
索引	.....	462

# 使用 S7-1200 运动控制

## 1.1 简介

### 1.1.1 CPU S7-1200 的运动功能

TIA Portal 结合 CPU S7-1200 的运动控制功能，可帮助用户控制步进电机和伺服电机：

- 在 TIA Portal 中对定位轴和命令表工艺对象进行组态。  
CPU S7-1200 使用这些工艺对象来控制用于控制驱动器的输出。
- 在用户程序中，可以通过运动控制指令来控制轴，也可以启动驱动器的运动命令。

更多信息，请访问 Internet

( <http://www.automation.siemens.com/mcms/topics/en/simatic/simatic-technology/integrated-functions/simatic-s7-1200/Pages/Default.aspx>) 上的多媒体说明文件。

#### 参见

用于运动控制的硬件组件 (页 11)

集成定位轴工艺对象 (页 46)

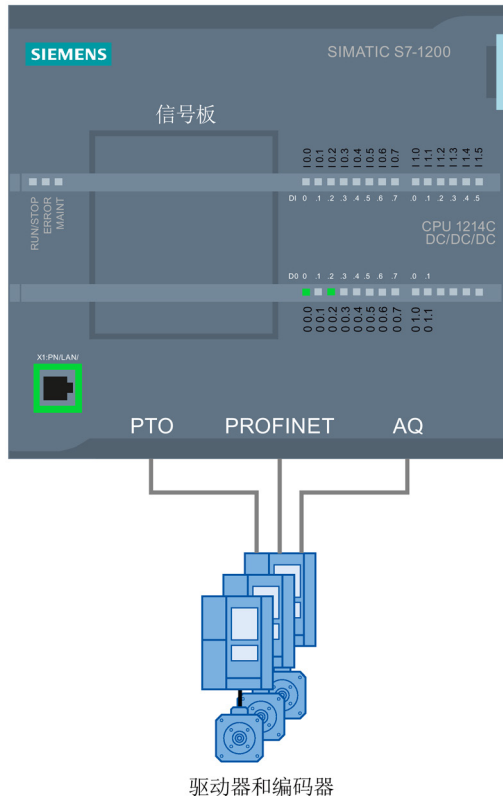
定位轴工艺对象的工具 (页 50)

使用命令表工艺对象 (页 127)

命令表工艺对象工具 (页 128)

## 1.1.2 用于运动控制的硬件组件

下图显示了使用 CPU S7-1200 进行运动控制应用的基本硬件配置。



### CPU S7-1200

CPU S7-1200 兼具可编程逻辑控制器的功能和用于控制驱动器运行的运动控制功能。运动控制功能负责对驱动器进行监控。

### 信号板

可以使用信号板为 CPU 添加其它输入和输出。

如果需要，还可将数字量输出用作控制驱动器的脉冲发生器输出。

对于具有继电器输出的 CPU，

由于继电器不支持所需的开关频率，因此无法通过板载输出来输出脉冲信号。

如果要在这些 CPU 中使用 PTO (Pulse Train Output)，

必须使用具有数字量输出的信号板。

如果需要，还可使用模拟量输出来控制所连接的模拟量驱动器。

## PROFINET

PROFINET 接口用于在 CPU S7-1200 与编程设备之间建立在线连接。

除了 CPU 的在线功能外，附加的调试和诊断功能也可用于运动控制。

PROFINET 仍然支持用于连接 PROFIdrive 驱动器的 PROFIdrive 配置文件。

## 驱动器和编码器

驱动器用于控制轴的运动。编码器提供轴的闭环位置控制的实际位置。

下表显示了驱动器和编码器可能的连接方式：

驱动器连接	轴的闭环/开环控制	编码器连接
PTO (Pulse Train Output) (带有脉冲接口的步进电机和伺服电机)	速度控制	-
模拟量输出 (AQ)	位置控制	<ul style="list-style-type: none"> <li>• 高速计数器 (HSC) 上的编码器</li> <li>• 工艺模块 (TM) 上的编码器</li> <li>• PROFIdrive 编码器 (位于 PROFINET)</li> </ul>
PROFIdrive	位置控制	<ul style="list-style-type: none"> <li>• 驱动器上的编码器</li> <li>• 高速计数器 (HSC) 上的编码器</li> <li>• 工艺模块 (TM) 上的编码器</li> <li>• PROFIdrive 编码器 (位于 PROFINET)</li> </ul>

**CPU 固件 V4.1 的订购信息**

下列订购信息适用于 TIA Portal  
中的当前已安装产品系列（不包括所有已安装的硬件支持包）。

名称	MLFB - 产品编号
CPU 1211C DC/DC/DC	6ES7211-1AE40-0XB0
CPU 1211C AC/DC/RLY	6ES7211-1BE40-0XB0
CPU 1211C DC/DC/RLY	6ES7211-1HE40-0XB0
CPU 1212C DC/DC/DC	6ES7212-1AE40-0XB0
CPU 1212C AC/DC/RLY	6ES7212-1BE40-0XB0
CPU 1212C DC/DC/RLY	6ES7212-1HE40-0XB0
CPU 1214C DC/DC/DC	6ES7214-1AG40-0XB0
CPU 1214C AC/DC/RLY	6ES7214-1BG40-0XB0
CPU 1214C DC/DC/RLY	6ES7214-1HG40-0XB0
CPU 1214FC DC/DC/DC	6ES7214-1AF40-0XB0
CPU 1214FC DC/DC/RLY	6ES7214-1HF40-0XB0
CPU 1215C DC/DC/DC	6ES7215-1AG40-0XB0
CPU 1215C AC/DC/RLY	6ES7215-1BG40-0XB0
CPU 1215C DC/DC/RLY	6ES7215-1HG40-0XB0
CPU 1215FC DC/DC/DC	6ES7215-1AF40-0XB0
CPU 1215FC DC/DC/RLY	6ES7215-1HF40-0XB0
CPU 1217C DC/DC/DC	6ES7217-1AG40-0XB0
信号板 DI4 x DC 24 V (200 kHz)	6ES7221-3BD30-0XB0
信号板 DI4 x DC 5 V (200 kHz)	6ES7 221-3AD30-0XB0
信号板 DQ4 x DC 24 V (200 kHz)	6ES7222-1BD30-0XB0
信号板 DQ4 x DC 5 V (200 kHz)	6ES7222-1AD30-0XB0
信号板 DI2/DQ2 x DC 24 V (20 kHz)	6ES7223-0BD30-0XB0
信号板 DI2/DQ2 x DC 24 V (200 kHz)	6ES7223-3BD30-0XB0
信号板 DI2/DQ2 x DC 5 V (200 kHz)	6ES7223-3AD30-0XB0
信号板 AQ1 x 12 位 (±10 V, 0 到 20 mA)	6ES7 232-4HA30-0XB0

## 1.1 简介

可以使用硬件支持包 (HSP) 安装新的硬件组件。  
安装后，硬件组件随即显示在硬件目录中。

### 参见

CPU S7-1200 的运动功能 (页 10)

与运动控制相关的 CPU 输出 (页 15)

## 1.2 使用 S7-1200 运动控制的基础知识

### 1.2.1 通过 PTO 连接驱动器

#### 1.2.1.1 与运动控制相关的 CPU 输出

可用驱动器的数目取决于 PTO（脉冲串输出）数目以及可用的脉冲发生器输出数目。

下表提供了有关相关依赖性的信息：

#### 最大 PTO 数

每个配备工艺版本 V4 的 CPU 都可使用 4 个 PTO。

也就是说最多可以控制 4 个驱动器。

#### PTO 的信号类型

根据 PTO 的信号类型，每个 PTO（驱动器）需要 1 到 2 个脉冲发生器输出：

信号类型	脉冲发生器输出数目
脉冲 A 和 方向 B（禁用方向输出）*）	1
脉冲 A 和 方向 B *）	2
加计数 A 和减计数 B	2
A/B 相移	2
A/B 相移 - 四倍频	2

\*）方向输出必须是板载输出或者位于信号板上。

## 可用的脉冲发生器输出和频率范围

继电器型 CPU 仅可访问信号板的脉冲发生器输出。

根据 CPU 和信号板，以下各脉冲信号发生器输出可与下列频率范围配用：

板载	Q0.0	Q0.1	Q0.2	Q0.3	Q0.4	Q0.5	Q0.6	Q0.7	Q1.0	Q1.1
CPU 1211 (DC/DC/DC)	100 kHz	100 kHz	100 kHz	100 kHz	-	-	-	-	-	-
CPU 1212 (DC/DC/DC)	100 kHz	100 kHz	100 kHz	100 kHz	30 kHz	30 kHz	-	-	-	-
CPU 1214(F) (DC/DC/DC)	100 kHz	100 kHz	100 kHz	100 kHz	30 kHz	30 kHz	30 kHz	30 kHz	30 kHz	30 kHz
CPU 1215(F) (DC/DC/DC)	100 kHz	100 kHz	100 kHz	100 kHz	30 kHz	30 kHz	30 kHz	30 kHz	30 kHz	30 kHz
CPU 1217 (DC/DC/DC)	1 MHz	1 MHz	1 MHz	1 MHz	100 kHz	100 kHz	100 kHz	100 kHz	100 kHz	100 kHz
信号板	Qx.0	Qx.1	Qx.2	Qx.3	-	-	-	-	-	-
信号板 DI2/DO2 x DC24V 20kHz	20 kHz	20 kHz	-	-	-	-	-	-	-	-
信号板 DI2/DO2 x DC24V 200kHz	200 kHz	200 kHz	-	-	-	-	-	-	-	-
信号板 DO4 x DC24V 200kHz	200 kHz	200 kHz	200 kHz	200 kHz	-	-	-	-	-	-
信号板 DI2/DO2 x DC5V 200kHz	200 kHz	200 kHz	-	-	-	-	-	-	-	-
信号板 DO4 x DC5V 200kHz	200 kHz	200 kHz	200 kHz	200 kHz	-	-	-	-	-	-

最低频率始终为 1Hz。

可将脉冲发生器输出自由分配给 PTO。



**说明**

如果按照信号类型使用具有不同频率范围的脉冲发生器输出，那么每种情况中都要使用最低频率。

但信号类型“脉冲 A 和方向 B”除外。

采用此类信号类型时，始终使用脉冲发生器输出的频率范围。

**说明****通过过程映像访问脉冲发生器输出**

如果已选择 PTO (Pulse Train Output)

并将其分配给某个轴，固件将通过相应的脉冲发生器和方向输出接管控制。

在实现上述控制功能接管后，将断开过程映像和 I/O 输出间的连接。

虽然用户可通过用户程序或监视表写入脉冲发生器和方向输出的过程映像，但所写入的内容不会传送到 I/O 输出。因此通过用户程序或监视表格无法监视 I/O 输出。

读取的信息反映过程映像中的值，与 I/O 输出的实际状态不一致。

对于 CPU 固件非永久使用的其它所有 CPU 输出，通常可以通过过程映像监控 I/O 输出的状态。

**驱动器信号输出**

对于运动控制，可以选择将驱动器接口设置为“驱动器启用”或“驱动器准备就绪”。

使用驱动器接口时，可以随意选择用于驱动器启用的数字量输出和用于“驱动器准备就绪”的数字量输入。

**加速度/减速度限值**

以下限值适用于加速度和减速度：

加速度/减速度	值
最小加速度/减速度	5.0E-3 个脉冲/秒 <sup>2</sup>
最大加速度/减速度	9.5E+9 个脉冲/秒 <sup>2</sup>

### 加加速度限值

以下限值适用于加加速度：

加加速度	值
最小加加速度	4.0E-3 个脉冲/秒 <sup>3</sup>
最大加加速度	1.0E+10 个脉冲/秒 <sup>3</sup>

### 参见

与运动控制相关的 CPU 输出（工艺版本 V1...3） (页 290)

脉冲接口的工作原理 (页 19)

信号类型与行进方向之间的关系 (页 20)

硬件和软件限位开关 (页 32)

冲击限制 (页 33)

回原点 (页 34)

用于运动控制的硬件组件 (页 11)

集成定位轴工艺对象 (页 46)

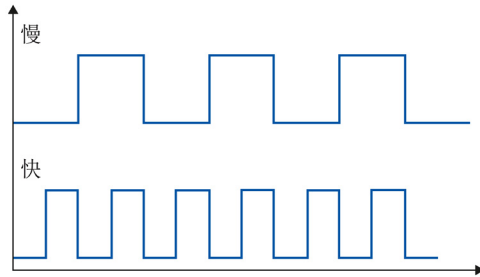
定位轴工艺对象的工具 (页 50)

### 1.2.1.2 脉冲接口的工作原理

根据步进电机的设置，每个脉冲会使步进电机移动特定角度。

例如，如果将步进电机设置为每转 1000 个脉冲，则每个脉冲电机移动  $0.36^\circ$ 。

步进电机的速度通过每单位时间的脉冲数来确定。



（此处所做的说明同样适用于带脉冲接口的伺服电机。）

### 参见

与运动控制相关的 CPU 输出 (页 15)

信号类型与行进方向之间的关系 (页 20)

硬件和软件限位开关 (页 32)

冲击限制 (页 33)

回原点 (页 34)

集成定位轴工艺对象 (页 46)

定位轴工艺对象的工具 (页 50)

### 1.2.1.3 信号类型与行进方向之间的关系

CPU 通过两个输出来输出速度和行进方向。

组态与行进方向之间的关系会因所选信号类型的不同而异。

可在轴组态的“基本参数 > 常规”(Basic parameters > General) 下组态以下信号类型：

- “PTO（脉冲 A 和方向 B）”
- “PTO（向上计数 A，向下计数 B）”（自版本 V4 起）
- “PTO（A/B 相移）”（自版本 V4 起）
- “PTO（A/B 相移 - 四倍频）”（自版本 V4 起）

可在轴组态的“扩展参数 > 机械”(Extended Parameters > Mechanics) 下组态方向。

如果选择“反转方向”(Invert direction) 选项，则相应信号类型的以下方向逻辑会反转。

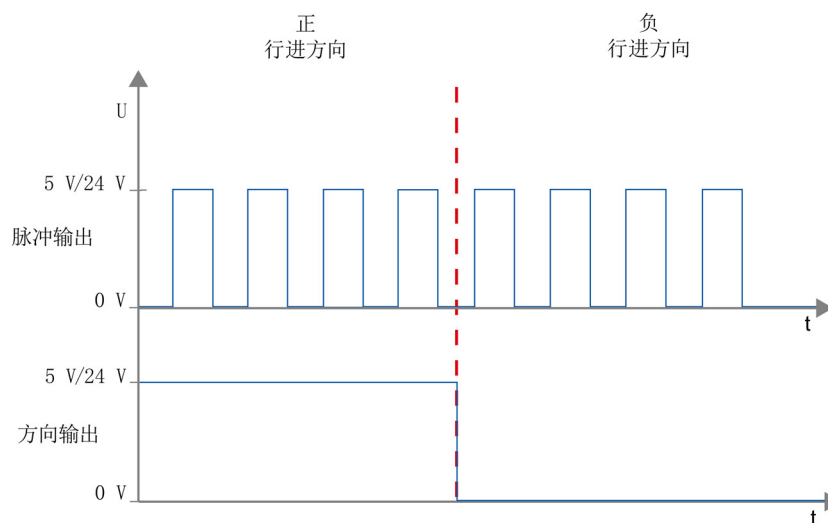
#### PTO（脉冲 A 和方向 B）

为该信号类型评估脉冲输出脉冲和方向输出电平。

脉冲通过 CPU 的脉冲输出进行输出。CPU 的方向输出指定驱动器的旋转方向：

- 方向输出上输出 5 V/24 V  $\Rightarrow$  正向旋转
- 方向输出上输出 0 V  $\Rightarrow$  反向旋转

指定的电压取决于所使用的硬件。各指示值并不适用于 CPU 1217 的差分输出。

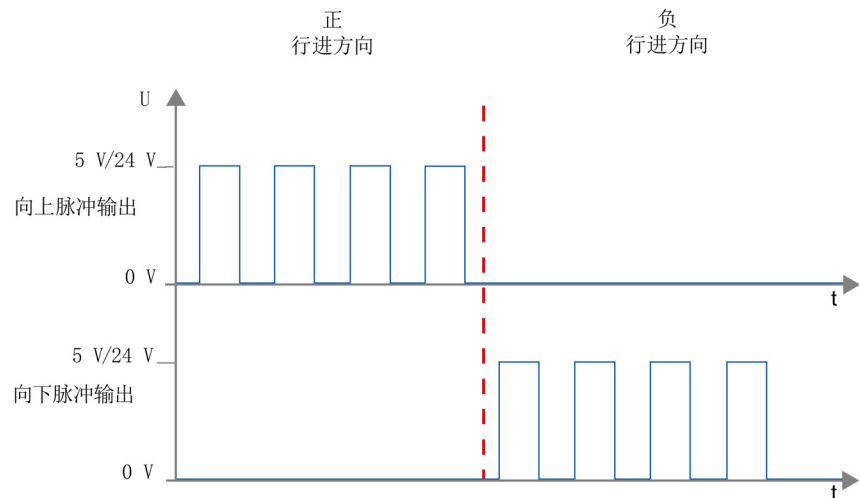


**PTO (向上计数 A, 向下计数 B) (自版本 V4 起)**

针对该信号类型评估一路输出的脉冲。

正转脉冲通过“脉冲输出加计数”输出；反转脉冲通过“脉冲输出减计数”输出。

指定的电压取决于所使用的硬件。各指示值并不适用于 CPU 1217 的差分输出。

**PTO (A/B 相移) (自版本 V4 起)**

为该信号类型评估每种情况下一个输出的上升沿。

脉冲通过“信号 A”输出，相移通过“信号 B”输出。输出之间的相移定义了旋转方向：

- 信号 A 超前信号 B  $90^\circ \Rightarrow$  正转
- 信号 B 超前信号 A  $90^\circ \Rightarrow$  反转

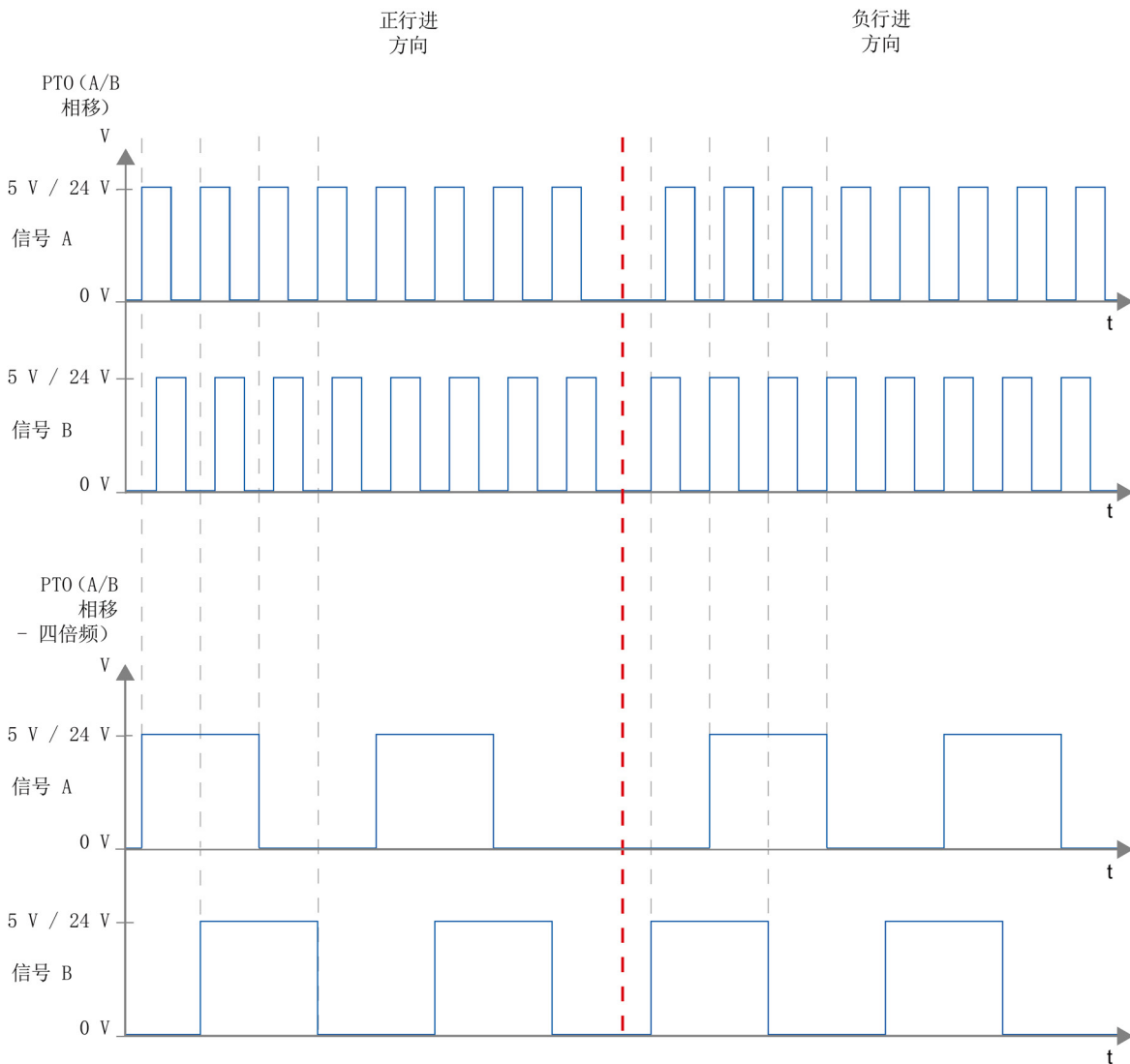
**PTO (A/B 相移 - 四倍频) (自版本 V4 起)**

为该信号类型评估两个输出的上升沿和下降沿。一个脉冲周期有四沿两相 (A 和 B)。因此, 输出中的脉冲频率会减小到四分之一。

脉冲通过“信号 A”输出, 相移通过“信号 B”输出。输出之间的相移定义了旋转方向:

- 信号 A 超前信号 B 90° ⇒ 正转
- 信号 B 超前信号 A 90° ⇒ 反转

指定的电压取决于所使用的硬件。各指示值并不适用于 CPU 1217 的差分输出。



## 反转方向

如果选择“反转旋转信号”(Invert rotation signal) 选项，则会反转方向逻辑：

- **PTO (脉冲 A 和方向 B)**

- 方向输出上输出 0 V (低电平) ⇒ 正向旋转
- 方向输出上输出 5 V/24 V (高电平) ⇒ 负向旋转

指定的电压取决于所使用的硬件。各指示电压并不适用于 CPU 1217 的差分输出。

- **PTO (向上计数 A, 向下计数 B)**

“向下脉冲输出”和“向上脉冲输出”这两个输出相互交换。

- **PTO (A/B 相移)**

“信号 A”和“信号 B”输出相互交换。

- **“PTO (A/B 相移 - 四倍频)**

“信号 A”和“信号 B”输出相互交换。

## 参见

与运动控制相关的 CPU 输出 (页 15)

脉冲接口的工作原理 (页 19)

硬件和软件限位开关 (页 32)

冲击限制 (页 33)

回原点 (页 34)

集成定位轴工艺对象 (页 46)

定位轴工艺对象的工具 (页 50)

## 1.2.2 PROFIdrive/模拟驱动器接口

### 1.2.2.1 驱动器和编码器连接

驱动器和编码器分配给通过 PROFIdrive/模拟驱动器接口与驱动器进行连接的定位轴。

具有 PROFIdrive 功能的驱动器可通过 PROFIdrive 报文进行连接。

带模拟量设定值接口的驱动器可使用模拟量输出与可选启用信号连接。

可通过 PROFIdrive 报文或通过模拟量输出指定驱动器中的设定值。

编码器值通过 PROFIdrive 报文或 HSC 接口进行传输。

### 连接选项

具有 PROFIdrive 功能的驱动器可通过 CPU 的 PROFINET 接口进行连接。

具有模拟量设定值接口的驱动器通过以下某种连接与 CPU 进行连接：

- 通过信号板的模拟量输出
- 通过模拟量输出模块的模拟量输出

可通过以下方式连接编码器：

- 驱动器上的编码器
- 高速计数器 (HSC) 上的编码器
- 工艺模块上的编码器
- PROFIdrive 编码器（直接在 PROFINET IO

### 最大轴数量

根据所用的控制器，可以通过 PROFIdrive 或模拟驱动器接口控制最多 8 个驱动器。



### 1.2.2.2 PROFIdrive

PROFIdrive 是通过 PROFINET IO 连接驱动器和编码器的标准化驱动技术配置文件。支持 PROFIdrive 配置文件的驱动器都可根据 PROFIdrive 标准进行连接。

控制器和驱动器/编码器之间通过各种 PROFIdrive 报文进行通信。每个报文均有一个标准化的结构。可根据具体应用，选择相应的报文。通过 PROFIdrive 报文，可传输控制字、状态字、设定值和实际值。

#### PROFIdrive 报文

定位轴的设定值通过 PROFIdrive 报文 1、2 或 3 传输到驱动器。编码器值既可与设定值（报文 3）一起通过报文传输，也可以通过单独的编码器报文（报文 81 或报文 83）传输。

下表显示了支持用于分配驱动器和编码器的 PROFIdrive 报文：

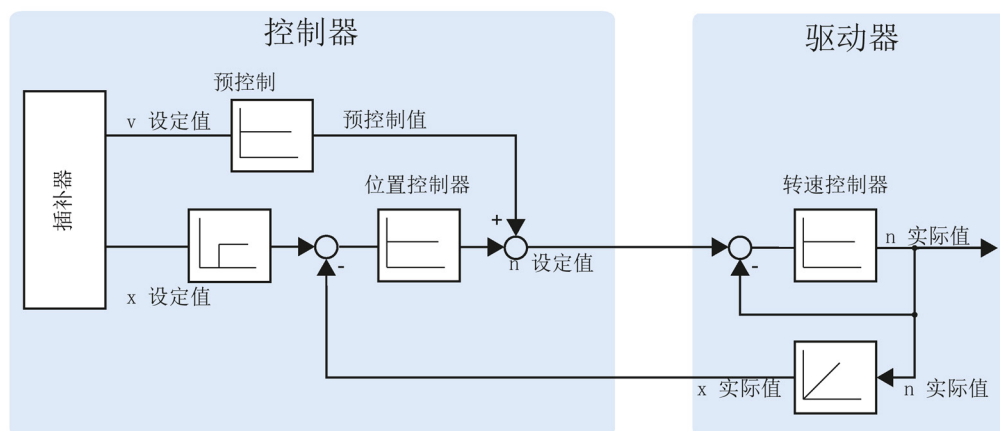
报文	简要描述
<b>标准报文</b>	
1	<ul style="list-style-type: none"> <li>• 16 位速度设定值 (NSET)</li> <li>• 16 位实际速度 (NACT)</li> </ul>
2	<ul style="list-style-type: none"> <li>• 32 位速度设定值 (NSET)</li> <li>• 32 位实际速度 (NACT)</li> <li>• 状态符号</li> </ul>
3	<ul style="list-style-type: none"> <li>• 32 位速度设定值 (NSET)</li> <li>• 32 位实际速度 (NACT)</li> <li>• 编码器值</li> <li>• 状态符号</li> </ul>
<b>标准报文编码器</b>	
81	<ul style="list-style-type: none"> <li>• 编码器值</li> <li>• 状态符号</li> </ul>
83	<ul style="list-style-type: none"> <li>• 32 位实际速度 (NACT)</li> <li>• 编码器值</li> <li>• 状态符号</li> </ul>

### 1.2.2.3 闭环控制

如果通过 PROFIdrive 或模拟量设定值接口连接驱动器，则轴的所有运动都通过位置进行控制。位置控制器是使用预控制速度的 P 控制器。

#### 控制器结构

下图显示了 S7-1200 运动控制的控制器结构：



MC-Interpolator [OB92] 计算轴的位置设定值。位置设定值与实际位置之间的差乘以位置控制器的值。得到的值将添加到预控制值，并作为驱动器的速度设定值进行输出。

编码器记录轴的实际位置，并将其通过 PROFIdrive 报文或 HSC（高速计数器）接口返回到控制器。

### 1.2.2.4 过程响应

#### 用于运动控制的组织块

#### 说明

创建工艺对象时，会自动地创建用于执行工艺对象的组织块。

工艺对象的运动控制功能可创建自己的执行级别，并根据运动控制应用循环进行调用。

创建下列块：

- **MC-Servo [OB91]**

位置控制器的计算

- **MC-Interpolator [OB92]**

评估运动控制指令、生成设定值和监控功能

组织块保护（专有知识保护）。无法查看或更改程序代码。

2 个组织块彼此之间出现的频率关系始终为 1:1。

MC-Servo [OB91] 总是在 MC-Interpolator [OB92] 之前执行。

可以根据控制质量和系统负载的需求，设定组织块的应用循环和优先级。

#### 应用循环

可以在组织块属性中的“常规 > 循环时间”(General > Cycle Time) 下设置应用循环，以在其中调用 MC-Servo [OB91]：

MC-Servo [OB91] 以指定的应用循环周期性调用。

所选的应用循环时间必须足够长，才能在一个循环中处理工艺对象。

如果工艺对象的处理时间长于应用循环，则将发生溢出 (页 28)。

可使用扩展指令“RT\_INFO”检查 MC-Servo [OB91] 和 MC-Interpolator [OB92] 的运行时间。

## 优先级

在组织块的属性中“常规 > 属性 > 优先级”

(General > Properties > Priority) 中，可以按需设定组织块的优先级：

- **MC-Servo [OB91]**

优先级 17 至 26（默认值 25）

- **MC-Interpolator [OB92]**

优先级 17 至 26（默认值 24）

MC-Servo [OB91] 的优先级必须至少比 MC-Interpolator [OB92] 的优先级高 1 级。

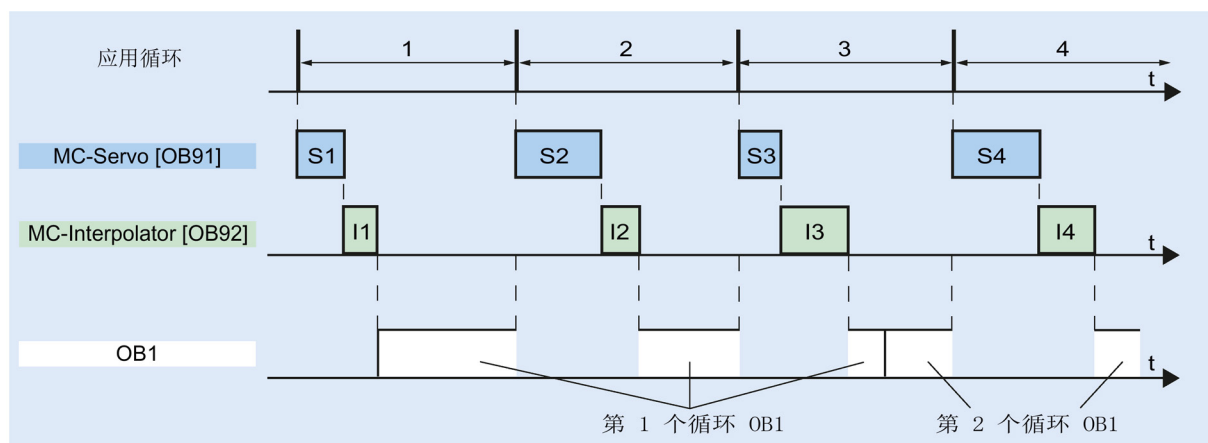
## 操作顺序和超时

处理运动控制功能时，每个应用循环都调用并处理组织块 MC-Servo [OB91] 和 MC-Interpolator [OB92]。余下的循环时间用于处理用户程序。

要实现无错程序执行，应遵循下列规则：

- 在每个应用循环中，都必须启动并完全执行 MC-Servo [OB91]。
- 在每个应用循环中，都必须至少启动相关的 MC-Interpolator [OB92]。

下图示例描述了用于运行组织块 OB1 的无故障操作顺序：

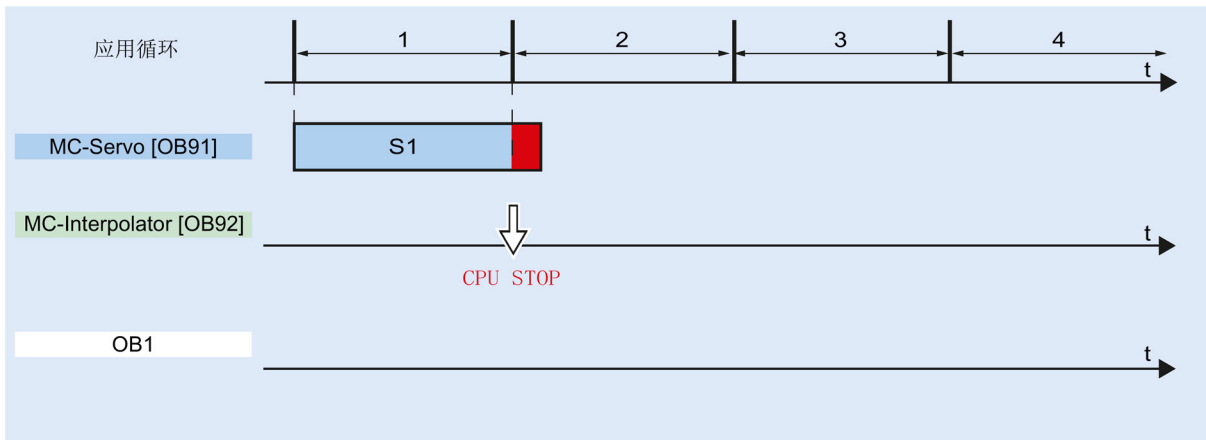


## 溢出

如果未遵循设置的应用循环（例如由于应用循环过短），则可能发生溢出。

CPU 不会容许发生 MC-Servo [OB91] 溢出。溢出将导致 CPU 切换至 STOP 模式。

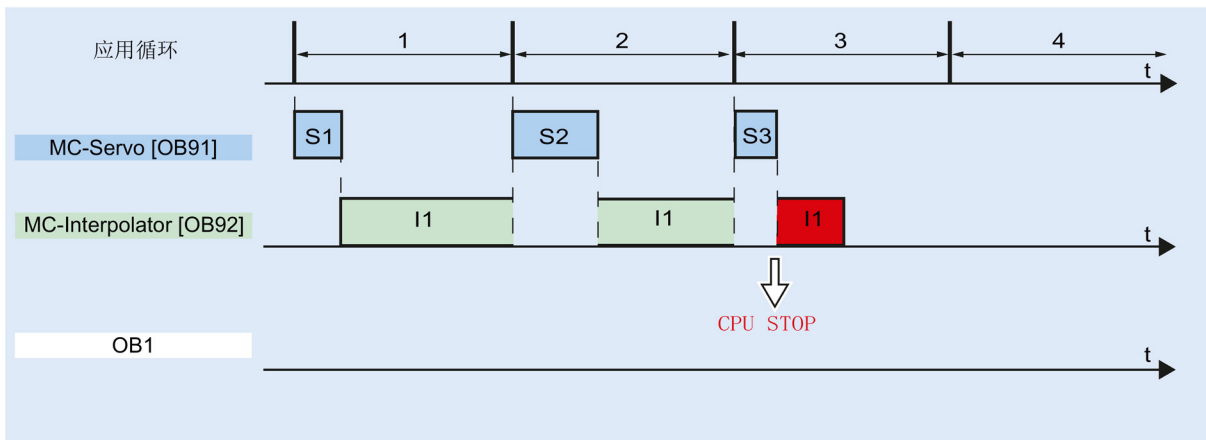
下图显示了发生 MC-Servo [OB91] 溢出时的操作顺序：



MC-Interpolator [OB92] 的执行只能由 MC-Servo [OB91] 调用中断。

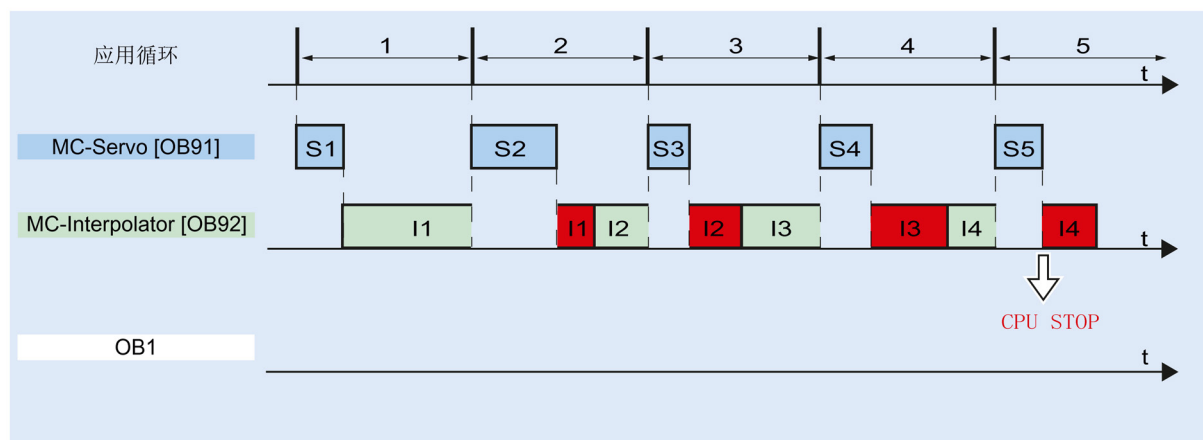
如果发生多次中断，CPU 将切换到 STOP 模式。

下图显示了当 MC-Interpolator [OB92] 在 2 个时间片上被中断时的顺序：



CPU 容许 MC-Interpolator [OB92] 最多连续溢出三次。如果发生多次溢出，CPU 将切换到 STOP 模式。

下图显示了 MC-Interpolator [OB92] 连续溢出 4 次时的顺序：



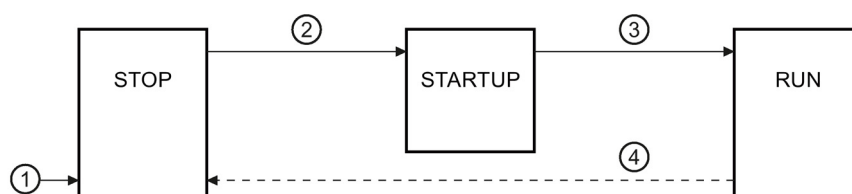
### 操作模式

本节描述各个操作模式下以及模式转换期间的运动控制特性。有关操作模式的一般性描述，请参见《系统手册 S7-1200》。

### 操作模式和转换

CPU 有 3 个操作模式：STOP、STARTUP 和 RUN。

下图描述了这些操作模式和操作模式转换：



### STOP 模式

在 STOP 模式下，不执行用户程序，所有过程输出均被禁用。因此，不执行任何运动控制作业。

工艺对象数据块被更新。

## STARTUP 模式

CPU 开始循环地执行用户程序之前，会运行启动 OB 一次。

在启动 (STARTUP) 模式下，过程输出被禁用。运动控制作业被拒绝。

工艺对象数据块被更新。

## RUN 模式

用户程序在 RUN 模式执行。

在 RUN 模式下，循环地调用、执行已经编程的运动控制作业。

工艺对象数据块被更新。

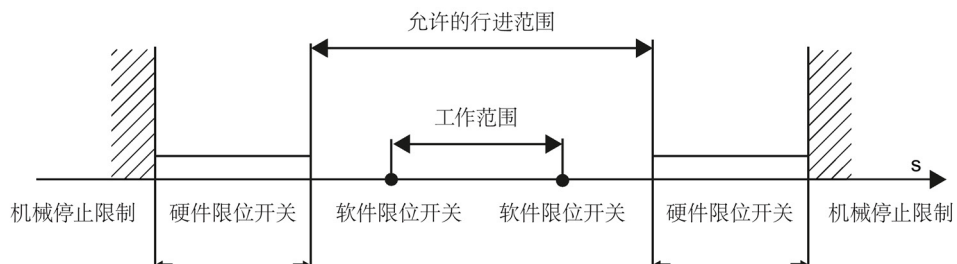
## 操作模式的转换

下表列出了操作模式转换过程中的运动控制特性。

编号	操作模式转换	特性
①	POWER ON → STOP	CPU 重启工艺对象。工艺对象使用装载存储器中的值重新初始化。
②	STOP → STARTUP	与运动控制无关。
③	STARTUP → RUN	过程输出被启用。
④	RUN → STOP	CPU 从 RUN 模式变为 STOP 模式时，将根据错误响应“取消启用”，禁用所有的工艺对象。运行中的运动控制作业被终止。

### 1.2.3 硬件和软件限位开关

硬限位开关和软限位开关用于限制定位轴工艺对象的“允许行进范围”和“工作范围”。这两者的相互关系如下图所示：



硬限位开关是限制轴的最大“允许行进范围”的限位开关。

硬限位开关是物理开关元件，必须与 CPU 中具有中断功能的输入相连接。

软限位开关将限制轴的“工作范围”。它们应位于限制行进范围的相关硬限位开关的内侧。由于软限位开关的位置可以灵活设置，因此可根据当前的运行轨迹和具体要求调整轴的工作范围。

与硬限位开关不同，软限位开关只通过软件来实现，而无需借助自身的开关元件。

在组态中或用户程序中使用硬件和软限位开关之前，必须先事先将其激活。

只有在轴回原点之后，才可以激活软限位开关。

#### 参见

与运动控制相关的 CPU 输出 (页 15)

脉冲接口的工作原理 (页 19)

信号类型与行进方向之间的关系 (页 20)

冲击限制 (页 33)

回原点 (页 34)

集成定位轴工艺对象 (页 46)

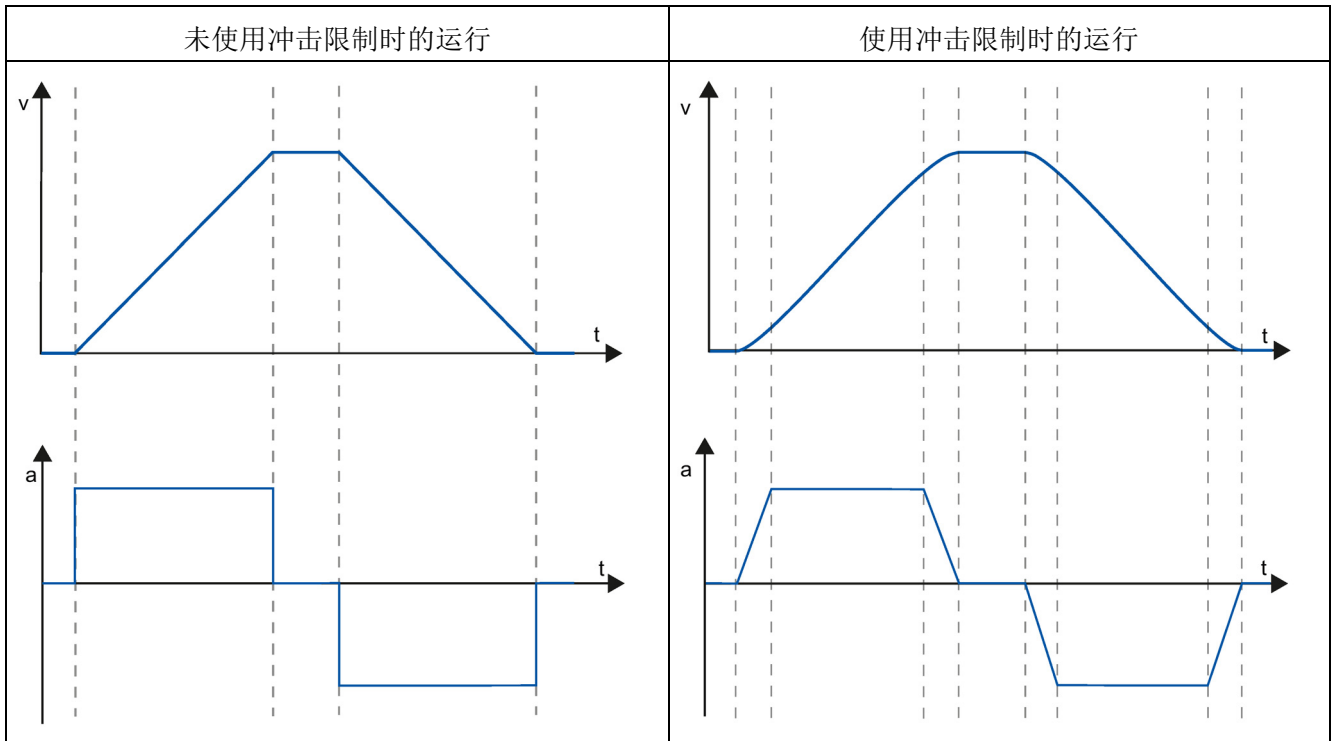
定位轴工艺对象的工具 (页 50)

位置限制 (页 77)



### 1.2.4 冲击限制

利用冲击限制，可以降低在加速和减速斜坡运行期间施加到机械上的应力。当冲击限制器处于激活状态时，加速度和减速度的值不会突然改变，而是逐渐增大和减小的。下图显示了不使用和使用冲击限制时的速度和加速度曲线。



使用冲击限制可以产生“平滑”的轴运动速度轨迹。例如，这可以确保传送带的软启动和软制动。

#### 参见

- 使用冲击限制时轴的行为 (页 87)
- 与运动控制相关的 CPU 输出 (页 15)
- 脉冲接口的工作原理 (页 19)
- 信号类型与行进方向之间的关系 (页 20)
- 硬件和软件限位开关 (页 32)
- 回原点 (页 34)
- 集成定位轴工艺对象 (页 46)
- 定位轴工艺对象的工具 (页 50)

## 1.2.5 回原点

回原点是指使工艺对象的轴坐标与驱动器的实际物理位置相匹配。

对于位置控制的轴，位置的输入与显示完全参考轴的坐标。

因此，轴坐标必需与实际情形相一致。

如果要确保通过驱动器也能准确到达轴的绝对目标位置，上述步骤必不可缺。

在 S7-1200 CPU 中，使用运动控制指令“MC\_Home”执行轴回原点。回原点模式有：

### 回原点模式

- **主动归位**

在主动归位模式下，运动控制指令“MC\_Home”将执行所需要的参考点逼近。

检测到回原点开关时，将根据组态使轴回原点。同时终止当前的行进运动。

- **被动归位**

被动归位期间，运动控制指令“MC\_Home”不会执行任何归位运动。

用户必须通过其它运动控制指令，执行这一步骤中所需的往返运动。

检测到回原点开关时，将根据组态使轴回原点。

被动回原点启动时，不会中止当前的行进运动。

- **绝对式直接回原点**

轴位置的设置与回原点开关无关。同时终止当前的行进运动。

立即将运动控制指令“MC\_Home”中输入参数“Position”的值，设置为轴的参考点。

- **相对式直接回原点**

轴位置的设置与回原点开关无关。同时终止当前的行进运动。

以下语句适用于回到原点后的定位：

新的轴位置 = 当前轴位置 + 指令“MC\_Home”中“Position”参数的值。

## 参见

- 与运动控制相关的 CPU 输出 (页 15)
- 脉冲接口的工作原理 (页 19)
- 信号类型与行进方向之间的关系 (页 20)
- 硬件和软件限位开关 (页 32)
- 冲击限制 (页 33)
- 集成定位轴工艺对象 (页 46)
- 定位轴工艺对象的工具 (页 50)
- 回原点（自定位轴工艺对象 V2 起） (页 91)

## 1.3 运动控制使用指南

该指南介绍了使用 CPU S7-1200 进行运动控制所需的基本步骤。

### 要求

要使用定位轴工艺对象，必须创建一个含有 CPU S7-1200 的项目。

### 步骤

要使用 CPU S7-1200 进行运动控制，并按指定的顺序执行以下步骤。为此，请点击以下链接：

1. 添加一个定位轴工艺对象 (页 53)
2. 使用组态对话框 (页 54)
3. 下载到 CPU (页 153)
4. 在调试窗口中对轴执行功能测试 (页 155)
5. 编程 (页 162)
6. 对轴控制执行诊断 (页 183)

## 1.4 使用版本

### 1.4 使用版本

#### 1.4.1 版本概述

S7-1200 运动控制相关版本之间的关系可在下表中找到：

#### 工艺版本

可以在“指令”(Instructions) 任务卡中的“工艺 > 运动控制 > S7-1200 运动控制”(Technology > Motion Control > S7-1200 Motion Control) 文件夹和“添加新对象”(Add new object) 对话框中检查当前选定的工艺版本。

在“指令”(Instructions) 任务卡的“工艺 > 运动控制 > S7-1200 运动控制”(Technology > Motion Control > S7-1200 Motion Control) 文件夹中选择工艺版本。

如果将相关版本的工艺对象添加到“添加新对象”(Add new object) 对话框，则工艺版本也会更改。

---

#### 说明

相关工艺版本的选择也会影响运动控制指令版本（任务卡）。

工艺对象和运动控制指令仅能在编译时或“下载到设备”时才会转换为选定版本。

---

#### 工艺对象版本

可以在巡视窗口的“属性 > 常规 > 信息”(Properties > General > Information) 下的“版本”(Version) 框中检查工艺对象的版本。

#### 运动控制指令版本

可以在巡视窗口的“属性 > 常规 > 信息”(Properties > General > Information) 下的“版本”(Version) 框中检查运动控制指令版本。

如果使用的运动控制指令版本与下面的兼容性列表不符，相关的运动控制指令将会突出显示在程序编辑器中。

## 兼容性列表

工艺		CPU	工艺对象	运动控制指令
V1.0		V1.0、V2.0、 V2.1、V2.2、 V3.0	轴 V1.0	MC_Power V1.0 MC_Reset V1.0 MC_Home V1.0 MC_Halt V1.0 MC_MoveAbsolute V1.0 MC_MoveRelative V1.0 MC_MoveVelocity V1.0 MC_MoveJog V1.0
V2.0	创新： <ul style="list-style-type: none"> <li>• 加加速度限值</li> <li>• 命令表</li> <li>• MC_Change-Dynamic</li> </ul>	V2.1、V2.2、 V3.0	轴 V2.0、命令表 V2.0	MC_Power V2.0 MC_Reset V2.0 MC_Home V2.0 MC_Halt V2.0 MC_MoveAbsolute V2.0 MC_MoveRelative V2.0 MC_MoveVelocity V2.0 MC_MoveJog V2.0 MC_CommandTable V2.0 MC_ChangeDynamic V2.0
V3.0	创新： 在 RUN 模式下加载	V2.2、V3.0、 V4.0	轴 V3.0、命令表 V3.0	MC_Power V3.0 MC_Reset V3.0 MC_Home V3.0 MC_Stop V3.0 MC_MoveAbsolute V3.0 MC_MoveRelative V3.0 MC_MoveVelocity V3.0 MC_MoveJog V3.0 MC_CommandTable V3.0 MC_ChangeDynamic V3.0

## 1.4 使用版本

工艺		CPU	工艺对象	运动控制指令
V4.0	创新： <ul style="list-style-type: none"> <li>• MC_ReadParam</li> <li>• MC_WriteParam</li> <li>• S7-1200 和 S7-1500 运动控制工艺数据块的标准化。</li> </ul>	V4.0	定位轴 V4.0, 命令表 V4.0	MC_Power V4.0 MC_Reset V4.0 MC_Home V4.0 MC_Halt V4.0 MC_MoveAbsolute V4.0 MC_MoveRelative V4.0 MC_MoveVelocity V4.0 MC_MoveJog V4.0 MC_CommandTable V4.0 MC_ChangeDynamic V4.0 MC_ReadParam V4.0 MC_WriteParam V4.0
V5.0	创新： <ul style="list-style-type: none"> <li>• 通过 PROFIdrive 连接驱动器</li> <li>• 模拟驱动器接口</li> <li>• PROFIdrive/模拟驱动器接口的位置控制</li> <li>• PROFIdrive/模拟驱动器接口的定位监控</li> <li>• MC-Servo [OB91]</li> <li>• MC-Interpolator [OB92]</li> </ul>	V4.1	定位轴 V5.0, 命令表 V5.0	MC_Power V5.0 MC_Reset V5.0 MC_Home V5.0 MC_Halt V5.0 MC_MoveAbsolute V5.0 MC_MoveRelative V5.0 MC_MoveVelocity V5.0 MC_MoveJog V5.0 MC_CommandTable V5.0 MC_ChangeDynamic V5.0 MC_ReadParam V5.0 MC_WriteParam V5.0

## 参见

更改工艺版本 (页 39)

变量兼容性列表 (页 40)

限位开关状态 (页 45)

## 1.4.2 更改工艺版本

在获得新工艺版本的所有优势前，可能需要设置/修改现有项目的工艺版本。

---

### 说明

#### 工艺对象变量的兼容性

当您在 V1...3 和  $\geq$  V4 之间切换时，如果正在用户程序、监控表等中使用工艺对象的变量，请参见兼容性列表 (页 40)。

---

### 设置/更改工艺版本

要设置或更改工艺版本，请执行以下步骤：

1. 打开程序编辑器（例如，通过打开 OB1）。
2. 在“指令”(Instructions) 任务卡的“工艺 > 运动控制 > S7-1200 运动控制”(Technology > Motion Control > S7-1200 Motion Control) 文件夹中打开所需的工艺版本。
3. 保存并编译项目。 请注意编译期间显示的任何错误信息。 处理指示的错误原因。
4. 检查工艺对象的组态。
5. 如有必要，请调整以下对象中的变量名称以与兼容性列表一致。
  - 用户程序
  - 监控表
  - 强制表
  - HMI 组态
  - 跟踪组态

### 参见

版本概述 (页 36)

限位开关状态 (页 45)

### 1.4.3 变量兼容性列表

在 V4 工艺框架中，S7-1200 运动控制和 S7-1500 运动控制的工艺数据块已经过标准化处理。对于 V4 及更高版本，这会生成新的定位轴和命令表工艺对象变量和变量名称。

如果用户程序中使用了工艺对象的变量，并且想要将项目从 V1...3 转换为 V4 或更高版本（或者反向转换），请注意下表中的信息。

编译项目时，“自动转换 V1...3 到 V4”列中列出的变量会自动转换。不会转换监视和强制表中的变量名称或者 HMI 或跟踪组态。



用户程序、监控表等中已新增以下变量或做出了调整，也可能还需要进行更正：

### Config 变量（定位轴）

V1.0 至 V3.0 的变量名称	V4.0 及更高版本的变量名称	自动转换 V1..3 到 ≥ V4
<轴名称>.Config.DynamicDefaults. Acceleration	<轴名称>.DynamicDefaults.Acceleration	√
<轴名称>.Config.DynamicDefaults. Deceleration	<轴名称>.DynamicDefaults.Deceleration	√
<轴名称>.Config.DynamicDefaults. EmergencyDeceleration	<轴名称>.DynamicDefaults. EmergencyDeceleration	√
<轴名称>.Config.DynamicDefaults. Jerk	<轴名称>.DynamicDefaults.Jerk	√
<轴名称>.Config.DynamicDefaults. JerkActive	不可用 在组态的加加速度 > 0.004 个脉冲/s <sup>3</sup> 时，将激活加加速度。	-
<轴名称>.Config.DynamicLimits. MaxVelocity	<轴名称>.DynamicLimits.MaxVelocity	√
<轴名称>.Config.DynamicLimits. MinVelocity	<轴名称>.DynamicLimits.MinVelocity	√
<轴名称>.Config.General.LengthUnit	<轴名称>.Units.LengthUnit	√
<轴名称>.Config.Homing. AutoReversal	<轴名称>.Homing.AutoReversal	√
<轴名称>.Config.Homing.Direction	<轴名称>.Homing.ApproachDirection	√
<轴名称>.Config.Homing.FastVelocity	<轴名称>.Homing.ApproachVelocity	√
<轴名称>.Config.Homing.Offset	<轴名称>.Sensor[1].ActiveHoming.HomePositi onOffset	√
<轴名称>.Config.Homing. SideActiveHoming	<轴名称>.Sensor[1].ActiveHoming.SideInput	√
<轴名称>.Config.Homing. SidePassiveHoming	<轴名称>.Sensor[1].PassiveHoming.SideInput	√
<轴名称>.Config.Homing.SlowVelocity	<轴名称>.Homing.ReferencingVelocity	√

## 1.4 使用版本

V1.0 至 V3.0 的变量名称	V4.0 及更高版本的变量名称	自动转换 V1..3 到 ≥ V4
<轴名称>.Config.Homing. SwitchedLevel	<轴名称>.Sensor[1].ActiveHoming. SwitchLevel <轴名称>.Sensor[1].PassiveHoming. SwitchLevel	-
<轴名称>.Config.Mechanics. InverseDirection	<轴名称>.Actor.InverseDirection	√
<轴名称>.Config.Mechanics. LeadScrew	<轴名称>.Mechanics.LeadScrew	√
<轴名称>.Config.Mechanics. PulsesPerDriveRevolution	<轴名称>.Actor.DriveParameter. PulsesPerDriveRevolution	√
<轴名称>.Config.PositionLimits_HW. Active	<轴名称>.PositionLimitsHW.Active	√
<轴名称>.Config.PositionLimits_HW. MaxSwitchedLevel	<轴名称>.PositionLimitsHW.MaxSwitchLevel	√
<轴名称>.Config.PositionLimits_HW. MinSwitchedLevel	<轴名称>.PositionLimitsHW.MinSwitchLevel	√
<轴名称>.Config.PositionLimits_SW. Active	<轴名称>.PositionLimitsSW.Active	√
<轴名称>.Config.PositionLimits_SW. MaxPosition	<轴名称>.PositionLimitsSW.MaxPosition	√
<轴名称>.Config.PositionLimits_SW. MinPosition	<轴名称>.PositionLimitsSW.MinPosition	√
不可用	<轴名称>.Actor.DirectionMode	-
不可用	<轴名称>.Actor.Type	-
不可用	<轴名称>.Sensor[1].ActiveHoming.Mode	-
不可用	<轴名称>.Sensor[1].PassiveHoming.Mode	-

**ErrorBits 变量（定位轴）**

V1.0 至 V3.0 的变量名称	V4.0 及更高版本的变量名称	自动转换 V1..3 到 ≥ V4
<轴名称>.ErrorBits.HwLimitMax	<轴名称>.ErrorBits.HWLimit	-
<轴名称>.ErrorBits.HwLimitMin	（注意新的状态位和限位开关状态 (页 45)部分。)	
<轴名称>.ErrorBits.SwLimitMaxExceeded	<轴名称>.ErrorBits.SWLimit	-
<轴名称>.ErrorBits.SwLimitMaxReached	（注意新的状态位和限位开关状态 (页 45)部分。)	
<轴名称>.ErrorBits.SwLimitMinExceeded		
<轴名称>.ErrorBits.SwLimitMinReached		
不可用	<轴名称>.ErrorBits.DirectionFault	-

**MotionStatus 变量（定位轴）**

V1.0 至 V3.0 的变量名称	V4.0 及更高版本的变量名称	自动转换 V1..3 到 ≥ V4
<轴名称>.MotionStatus.Distance	<轴名称>.StatusPositioning.Distance	√
<轴名称>.MotionStatus.Position	<轴名称>.Position	√
<轴名称>.MotionStatus.TargetPosition	<轴名称>.StatusPositioning.TargetPosition	√
<轴名称>.MotionStatus.Velocity	<轴名称>.Velocity	√

**StatusBits 变量（定位轴）**

V1.0 至 V3.0 的变量名称	V4.0 及更高版本的变量名称	自动转换 V1..3 到 ≥ V4
<轴名称>.StatusBits.Homing	<轴名称>.StatusBits.HomingCommand	√
<轴名称>.StatusBits.SpeedCommand	<轴名称>.StatusBits.VelocityCommand	√
不可用	<轴名称>.StatusBits.HWLimitMaxActive	-
不可用	<轴名称>.StatusBits.HWLimitMinActive	-
不可用	<轴名称>.StatusBits.SWLimitMaxActive	-
不可用	<轴名称>.StatusBits.SWLimitMinActive	-

**变量（命令表）**

V1.0 至 V3.0 的变量名称	V4.0 及更高版本的变量名称	自动转换 V1..3 到 ≥ V4
<命令表>.Config.Command[n].Position	<命令表>.Command[n].Position	√
<命令表>.Config.Command[n].Velocity	<命令表>.Command[n].Velocity	√
<命令表>.Config.Command[n].Duration	<命令表>.Command[n].Duration	√
<命令表>.Config.Command[n].NextStep	<命令表>.Command[n].NextStep	√
<命令表>.Config.Command[n].StepCode	<命令表>.Command[n].StepCode	√

**参见**

版本概述 (页 36)

更改工艺版本 (页 39)

自 V4 起的定位轴工艺对象的变量 (页 235)

### 1.4.4 限位开关状态

在版本 V4 中，已对到达的限位开关显示的状态和错误位进行了调整。

为了复制版本 V1...3 的错误位行为，可使用以下逻辑运算符：

V1...3	V4 或更高版本
<轴名称>.ErrorBits.HwLimitMin	<轴名称>.ErrorBits.HWLimit AND <轴名称>.StatusBits.HWLLimitMinActive
<轴名称>.ErrorBits.HwLimitMax	<轴名称>.ErrorBits.HWLimit AND <轴名称>.StatusBits.HWLLimitMaxActive
<轴名称>.ErrorBits.SwLimitMinReached	<轴名称>.ErrorBits.SWLimit AND (<轴名称>.Position = <轴名称>.PositioningLimits_SW.MinPosition)
<轴名称>.ErrorBits.SwLimitMinExceeded	<轴名称>.ErrorBits.SWLimit AND (<轴名称>.Position < <轴名称>.PositioningLimits_SW.MinPosition)
<轴名称>.ErrorBits.SwLimitMaxReached	<轴名称>.ErrorBits.SWLimit AND (<轴名称>.Position = <轴名称>.PositioningLimits_SW.MaxPosition)
<轴名称>.ErrorBits.SwLimitMaxExceeded	<轴名称>.ErrorBits.SWLimitAND (<轴名称>.Position > <轴名称>.PositioningLimits_SW.MaxPosition)

#### 参见

版本概述 (页 36)

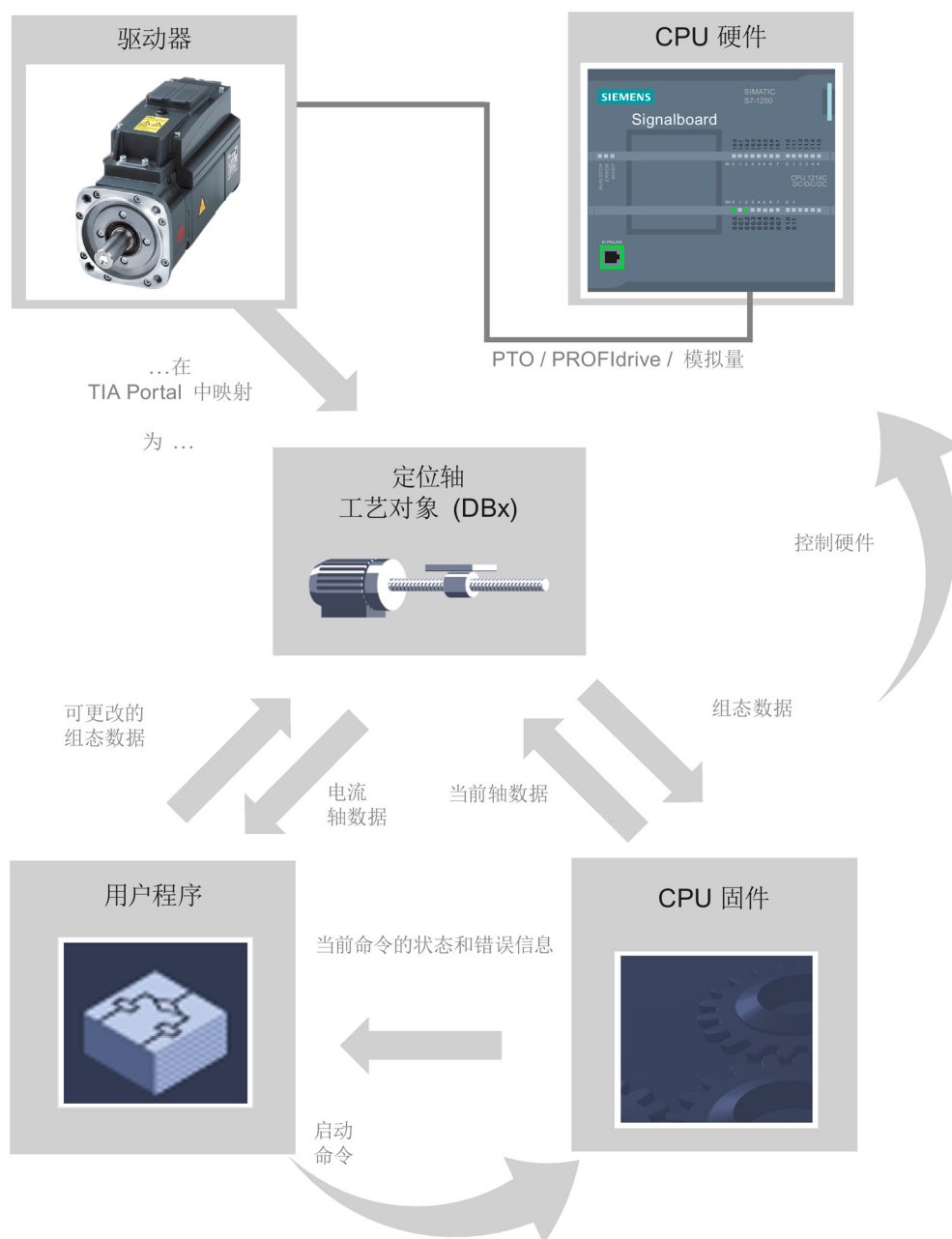
更改工艺版本 (页 39)

变量兼容性列表 (页 40)

## 1.5 定位轴工艺对象

### 1.5.1 集成定位轴工艺对象

下图显示了使用定位轴工艺对象时所采用的硬件和软件组件间的关系：



## CPU 硬件

通过 CPU 硬件对物理驱动器进行监控。

## 驱动器

驱动器是由动力装置和电机组成的单元。可以使用带有脉冲、PROFIdrive 或模拟接口的步进电机和伺服电机。

## 定位轴工艺对象

包含机械的物理驱动器在 TIA Portal 中映射为定位轴工艺对象。为此，使用以下参数组态定位轴工艺对象：

- 要使用的 PTO (Pulse Train Output)/PROFIdrive 驱动器/模拟量输出的选择选项和驱动器接口的组态
- 机械参数和驱动器（机器或系统）的传动比参数
- 位置限制和定位监控的参数
- 动态和归位的参数
- 控制回路的参数

定位轴工艺对象的组态保存在该工艺对象（数据块）中。该数据块也将作为用户程序和 CPU 固件间的接口。用户程序运行期间，当前的轴数据保存在该工艺对象的数据块中。

## 用户程序

可以使用用户程序启动 CPU 固件中的运动控制指令作业。包括以下用于控制轴的作业：

- 启用和禁用轴
- 绝对定位轴
- 相对定位轴
- 以设定的速度移动轴
- 按运动顺序运行轴命令（自 V2 工艺版本起，仅限 PTO）
- 在点动模式下移动轴
- 停止轴
- 参考轴；设置参考点
- 更改轴的动态设置
- 连续读取轴的运动数据
- 写入轴变量
- 确认错误

可以通过运动控制指令的输入参数和轴组态，确定命令参数。  
该指令的输出参数将提供有关状态和所有命令错误的最新信息。

启动轴命令之前，必须使用运动控制指令“MC\_Power”启用轴。

可以使用工艺对象的变量读取组态数据和当前轴数据。

可以通过用户程序更改工艺对象的单个可更改变量（例如，当前加速度）。

也可以通过运动控制指令“MC\_ChangeDynamic”更改轴的动态设置，  
通过“MC\_WriteParam”写入其它组态数据。

通过运动控制指令“MC\_ReadParam”读取轴的当前运动状态。



## CPU 固件

用户程序中启动的运动控制作业在 CPU 固件中进行处理。

使用轴控制面板时，可以通过操作轴控制面板来触发运动控制作业。

CPU 固件执行以下作业，具体取决于组态：

- 计算运动作业的精确运动轨迹和紧急停止情况
- 通过 PROFIdrive/模拟驱动器接口的驱动器连接位置控制
- 通过 PTO 控制驱动器接口的脉冲和方向信号
- 控制驱动器启用
- 监视驱动器，以及硬限位开关和软限位开关
- 将最新状态和错误信息反馈给用户程序中的运动控制指令
- 将当前的轴数据写入到该工艺对象的数据块中

## 参见

自 V4 起的定位轴工艺对象的变量 (页 235)

与运动控制相关的 CPU 输出 (页 15)

信号类型与行进方向之间的关系 (页 20)

定位轴工艺对象的工具 (页 50)

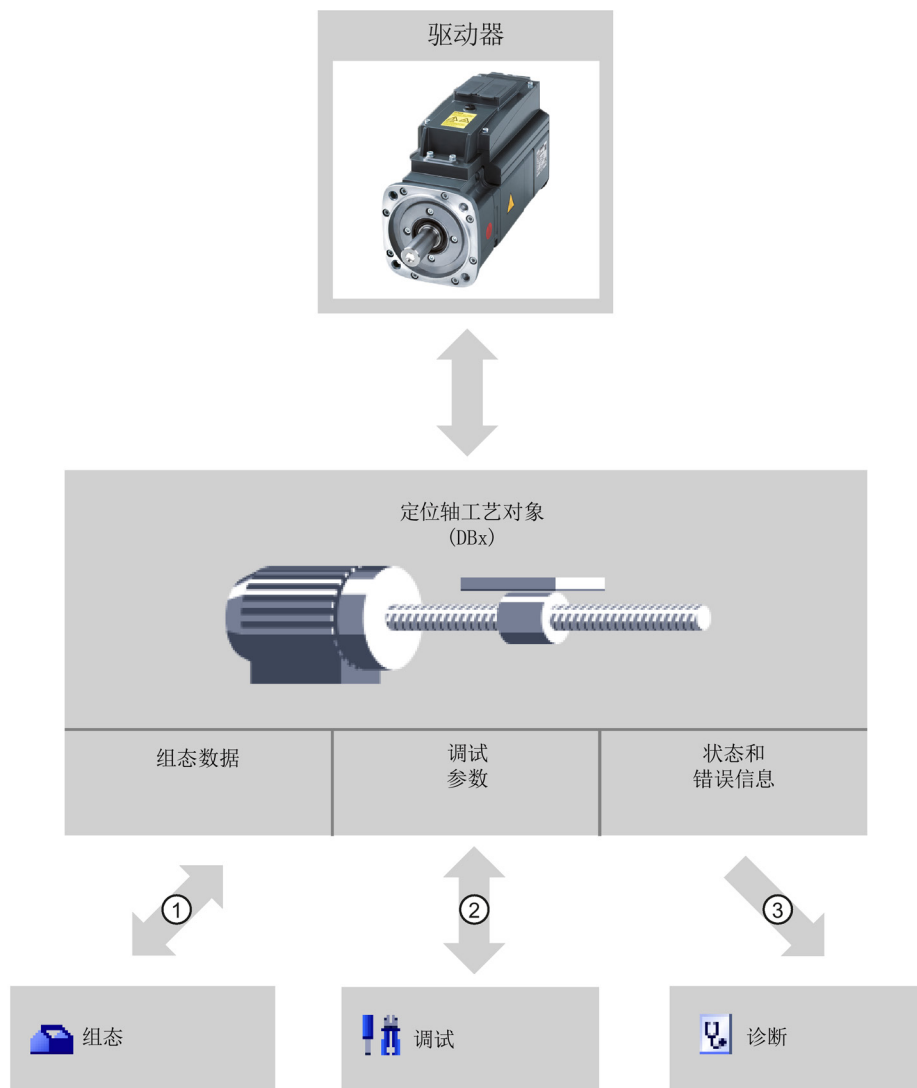
硬件和软件限位开关 (页 32)

回原点 (页 34)

1.5 定位轴工艺对象

1.5.2 定位轴工艺对象的工具

TIA Portal 中将为定位轴工艺对象提供“组态”(Configuration)、“调试”(Commissioning)和“诊断”(Diagnostics) 工具。 下图显示了这三种工具与工艺对象和驱动器的相互关系：



①	读取和写入工艺对象的组态数据
②	通过工艺对象的驱动器控制。 读取要在控制面板上显示的轴状态
③	读取工艺对象的当前状态和错误信息

## 组态

使用“组态”(Configuration) 工具可以组态定位轴工艺对象的以下属性:

- 要使用的 PTO (Pulse Train Output)/PROFIdrive 驱动器/模拟量输出的选择选项和驱动器接口的组态
- 机械装置的属性和驱动器（机器/设备）的传动比参数
- 位置限制和定位监控的属性
- 动态和归位的属性
- 控制回路的参数

在工艺对象的数据块中保存组态数据。

## 调试

使用“调试”工具即可测试轴的功能，无需创建用户程序。

启动该工具时，将显示轴控制面板。轴控制面板提供了下列命令:

- 启用和禁用轴
- 在点动模式下移动轴
- 以绝对和相对方式定位轴
- 使轴归位
- 确认错误信息

可以为运动命令相应地调整动态值。轴控制面板还显示当前的轴状态。

利用基于 PROFIdrive/模拟量输出的驱动器连接，通过自整定，可确定最佳的控制环增益。

## 1.5 定位轴工艺对象

### 诊断

使用“诊断”工具，可以跟踪轴和驱动器的当前状态和错误信息。

### 参见

- 与运动控制相关的 CPU 输出 (页 15)
- 信号类型与行进方向之间的关系 (页 20)
- 集成定位轴工艺对象 (页 46)
- 硬件和软件限位开关 (页 32)
- 回原点 (页 34)
- 组态定位轴工艺对象 (页 54)
- 轴控制面板 (页 155)
- 轴 - 诊断 (页 183)

### 1.5.3 添加一个定位轴工艺对象

#### 要求

已创建具有 CPU S7-1200 的项目。

#### 步骤

要在项目树中添加定位轴工艺对象，请按以下步骤操作：

1. 在项目树中打开“CPU > 工艺对象”(CPU > Technology objects) 文件夹。
2. 双击“添加新对象”(Add new object) 命令。  
将打开“添加新对象”(Add new object) 对话框。
3. 选择“运动控制”(Motion Control) 工艺。
4. 打开“运动控制 > S7-1200 运动控制”  
(Motion Control > S7-1200 Motion Control) 文件夹。
5. 在“版本”(Version) 列中选择所需的工艺版本。
6. 选择“TO\_PositioningAxis”对象。
7. 在“名称”(Name) 输入框中输入轴名称。
8. 要更改自动分配的数据块编号，请选择“手动”(Manual) 选项。
9. 要显示有关工艺对象的其它信息，请单击“其它信息”(Additional information)。
10. 单击“确定”(OK) 确认输入。

#### 结果

创建了新工艺对象，并保存在项目树中的“工艺对象”(Technology objects) 文件夹中。

组织块 MC-Servo [OB91] 和 MC-Interpolator [OB92] 在“程序块”(Program blocks) 文件夹中自动创建。工艺对象在这些组织块中进行处理。

MC-Servo [OB91] 中计算的位置控制器。

MC-Interpolator [OB92] 对运动控制指令、设定值生成和监视功能进行评估。

#### 参见

运动控制使用指南 (页 35)

## 1.5.4 组态定位轴工艺对象

### 1.5.4.1 使用组态对话框

在组态窗口中，组态工艺对象的属性。

要打开工艺对象的组态窗口，请按以下步骤操作：

1. 在项目树中打开所需工艺对象组。
2. 双击“组态”(Configuration) 对象。

组态分为以下几类：

- **基本参数**





基本参数包括必须为工作轴组态的所有参数。

- **扩展参数**

高级参数包括适合特定驱动器或设备的参数。

### 组态窗口图标

组态的区域导航中的图标显示有关组态情况的详细信息：

	<p><b>组态包含默认值且已完成。</b></p> <p>组态仅包含默认值。使用这些默认值即可使用工艺对象，而无需进行更改。</p>
	<p><b>组态包含用户设置的值且已完成</b></p> <p>组态的所有输入域均包含有效值，且至少一个预设值发生更改。</p>
	<p><b>组态未完成或不正确</b></p> <p>至少一个输入域或下拉列表包含无效值。相应域或下拉列表以红色背景显示。 单击弹出错误消息可显示错误原因。</p>
	<p><b>组态有效但包含警告</b></p> <p>例如，只组态了一个硬限位开关。 根据设备的不同，缺少硬限位开关的组态可能引起危险。相应域或下拉列表以黄色背景显示。</p>

### 参见

运动控制使用指南 (页 35)

基本参数 (页 56)

扩展参数 (页 74)

### 1.5.4.2 比较值

如果存在到 CPU 的在线连接，“比较值”功能将显示在工艺对象的组态中。





“比较值”功能提供了以下选项：

- 将项目中组态的起始值与 CPU 中的起始值和实际值进行比较
- 直接编辑实际值和项目的起始值
- 立即检测并显示输入错误和建议的更正措施
- 备份项目中的实际值
- 将项目的起始值作为实际值传送至 CPU

### 图标和操作员控件

如果存在到 CPU 的在线连接，则参数中将显示实际值。

除参数的实际值外，还会显示下列符号：

图标	说明
	CPU 中的起始值与已组态的项目中起始值相匹配
	CPU 中的起始值与已组态的项目中的起始值不匹配
	无法将 CPU 中的起始值与已组态的项目中的起始值进行比较
	使用按钮显示 CPU 的起始值和各个参数的项目起始值。

可以直接更改项目中的实际值和起始值，然后下载到 CPU。

可直接修改的参数的实际值更改将直接传送到 CPU。

## 1.5 定位轴工艺对象

### 1.5.4.3 基本参数

#### 组态 - 常规

在“常规”(General) 组态窗口中，组态定位轴工艺对象的基本属性。

#### 轴名称

在该域中定义轴名称或定位轴工艺对象的名称。该工艺对象以该名称列出在项目树中。

#### 驱动器

选择驱动器连接的类型：

- **PTO (Pulse Train Output)**

驱动器通过脉冲发生器输出、可选使能输出和可选准备就绪输入进行连接。

- **模拟驱动器接口**

驱动器通过模拟量输出、可选使能输出和可选准备就绪输入进行连接。

轴的所有运动均在位置控制下进行。

- **PROFIdrive**

驱动器通过 PROFINET 进行连接。控制器和驱动器之间通过 PROFIdrive 报文进行通信。

轴的所有运动均在位置控制下进行。

如果选择“模拟驱动器连接”(Analog drive connection) 或“PROFIdrive”，其它元素将添加到组态导航中：

- 编码器
- 模数
- 定位监控
- 控制回路

在其它组态窗口中，可以组态待连接的编码器，以及位置控制和定位监控的结果选项。



## 计量单位

从下拉列表中为轴量纲系统选择所需的单位。

所选单位将用于进一步组态定位轴工艺对象和显示当前轴数据。

运动控制指令的输入参数（例如，Position、Distance、Velocity 等）的值也会采用该单位。

---

### 说明

如果后来更改了该量纲系统，则在该工艺对象的所有组态窗口中可能无法相应地进行正确转换。这种情况下，请检查所有轴参数的组态。

可能需要在用户程序中根据新测量单位调整运动控制指令的输入参数的值。

---

## 参见

与运动控制相关的 CPU 输出 (页 15)

信号类型与行进方向之间的关系 (页 20)

组态 - 常规（“轴”工艺对象 V1...3） (页 295)

## 1.5 定位轴工艺对象

### 组态 - 驱动器

#### 组态 - 驱动器 - PTO（脉冲串输出）

在“驱动器”(Drive) 组态窗口中，组态脉冲发生器和驱动器的使能与反馈。

#### 硬件接口

脉冲通过固定分配的 digital 量输出输出到驱动器的动力装置。

对于具有继电器输出的

CPU，由于继电器不支持所需的开关频率，因此无法通过这些输出来输出脉冲信号。

如果要在这些 CPU 中使用 PTO (Pulse Train Output)，必须使用具有 digital 量输出的信号板。

---

#### 说明

PTO 需要高速计数器 (HSC) 的功能。

为此将使用内部 HSC，不能对此 HSC 的计数进行评估。

---

#### 选择脉冲发生器

在该下拉列表中，选择 PTO

(Pulse Train Output)，通过脉冲接口来控制步进电机或伺服电机。

如果没有在设备组态中的其它地方使用脉冲发生器和高速计数器，则系统会自动组态硬件接口。这种情况下，下拉列表中所选的 PTO 以白色背景显示。

按下“设备组态”(Device configuration) 按钮可转到 CPU 设备组态中的脉冲选项参数分配。

## 信号类型

从下拉列表中选择信号类型。可以使用以下信号类型：

- **PTO（脉冲 A 和方向 B）**

使用一个脉冲输出和一个方向输出控制步进电机。

- **PTO（向上计数 A，向下计数 B）**

分别使用一个正向和负向运动的脉冲输出控制步进电机。

- **PTO（A/B 相移）**

A 相和 B 相的两个脉冲输出在同一频率下运行。

在驱动器步进结束时评估这两个脉冲输出的周期。

A 相和 B 相之间的相位偏移量决定了运动方向。

- **PTO（A/B 相位偏移量 - 四重）**

A 相和 B 相的两个脉冲输出在同一频率下运行。

在驱动器步进结束时评估 A 相和 B 相的所有上升沿和下降沿。

A 相和 B 相之间的相位偏移量决定了运动方向。

## 1.5 定位轴工艺对象

下表列出了要根据信号类型组态的参数：

信号类型/参数	说明
<b>PTO（脉冲 A 和方向 B）</b>	
脉冲输出	在此域中选择用于实现正向运动的脉冲输出。 可以通过符号地址或将其分配给绝对地址来选择输出。
激活方向输出	使用此选项，可以启用或禁用方向输出。 在禁用方向输出时，将会限制运动方向。
方向输出	在此域中选择用作方向输出的输出。 可以通过符号地址或将其分配给绝对地址来选择输出。
<b>PTO（向上计数 A，向下计数 B）</b>	
脉冲正向输出	在此域中选择用于实现正向运动的脉冲输出。 可以通过符号地址或将其分配给绝对地址来选择输出。
脉冲反向输出	在此域中选择用于实现负向运动的脉冲输出。 可以通过符号地址或将其分配给绝对地址来选择输出。
<b>PTO（A/B 相位偏移量）/PTO（A/B 相位偏移量 - 四重）</b>	
信号 A	在此域中为 A 相信号选择脉冲输出。 可以通过符号地址或将其分配给绝对地址来选择输出。
信号 B	在此域中为 B 相信号选择脉冲输出。 可以通过符号地址或将其分配给绝对地址来选择输出。

## 驱动器使能信号与反馈

在此区域中组态驱动器使能信号的输出以及驱动器的“驱动器准备就绪”(Drive ready) 反馈信号的输入。

- **选择使能输出**

在此域中为驱动器使能信号选择使能输出。

- **选择准备就绪输入**

在此域中为驱动器的“驱动器准备就绪”(Drive ready) 反馈选择准备就绪输入。驱动器使能信号由运动控制指令“MC\_Power”控制，可以启用对驱动器的供电。如果驱动器在接收到驱动器使能信号之后准备好开始执行运动，则驱动器会向 CPU 发送“驱动器准备就绪”(Drive ready) 信号。

如果驱动器不包含此类型的任何接口，则无需组态这些参数。这种情况下，为准备就绪输入选择值 TRUE。

## 组态 - 驱动器 - 模拟驱动器接口

在“驱动器”(Drive) 组态窗口中，组态模拟量输出以及驱动器的使能与反馈。

## 硬件接口

参考速度通过永久分配的模拟量输出输出到驱动器的动力装置。

在以下区域中为驱动器的控制组态输入和输出：

- **模拟量输出**

在此域中，选择用于控制驱动器的模拟量输出的 PLC 变量。

在打开自动填充功能时，所有输出地址都以 16 位显示（WORD、INT、UINT）。

也可以输入地址，例如 QW20。

如果地址有效，将生成此地址的名称“Axis\_1\_AnalogOutput”，并将其插入到变量表中。

- **选择使能输出**

在此域中为驱动器使能信号选择使能输出。

- **选择准备就绪输入**

在此域中为驱动器的“驱动器准备就绪”(Drive ready) 反馈选择准备就绪输入

驱动器使能信号由运动控制指令“MC\_Power”控制，可以启用对驱动器的供电。

如果驱动器在接收到驱动器使能信号之后准备好开始执行运动，则驱动器会向 CPU 发送“驱动器准备就绪”(Drive ready) 信号。

如果驱动器不包含此类型的任何接口，则无需组态这些参数。

这种情况下，为准备就绪输入选择值 TRUE。

## 与驱动器之间的数据交换

在该区域中，可以组态速度设定值的缩放：

- **参考速度**

驱动器的参考速度，是指模拟量输出为 100% 输出时驱动器的旋转速度。驱动器中必须组态参考速度，并传输到工艺对象的组态中。

以 100% 输出的模拟值具体取决于模拟量输出的类型。

例如，对于  $\pm 10\text{ V}$  的模拟量输出，值  $10\text{ V}$  是以 100% 的方式输出的。

模拟量输出可以过载约 17%。如果驱动器允许过载，则可以使用它在 -117% 到 117% 的范围内运行模拟量输出。

- **最大速度**

在该域中，将指定驱动器的最大速度。

- **反转驱动装置方向**

如果希望将驱动器的旋转方向改为反转，则选中该复选框。

## 组态 - 驱动器 - PROFIdrive

在“驱动器”(Drive) 组态窗口中，选择 PROFIdrive 驱动器并组态驱动器和控制器之间的数据交换。

## 选择 PROFIdrive 驱动器

在“驱动器”(Drive) 域中，选择一个已经组态的 PROFIdrive 驱动器。

## 与驱动器之间的数据交换

在该区域中，可以组态驱动器和控制器之间的数据交换：

- **报文**

在下拉列表中，选择驱动器的报文。其技术数据必须与驱动器的设备组态相一致。

- **输入/输出地址**

这些域显示报文的符号及绝对输入和输出地址。

- **参考速度**

在该域中，根据制造商的技术数据，组态驱动器的参考速度。  
驱动器速度在 -200% 到 200% 的范围内输出。

- **最大速度**

在该域中，将指定驱动器的最大速度。

- **反转驱动装置方向**

如果希望将驱动器的旋转方向改为反转，则选中该复选框。

## 组态 - 编码器

### 编码器连接

在“编码器”(Encoder) 组态窗口中，根据所选的编码器连接组态各个参数。  
可实现以下编码器连接：

- 驱动器上的编码器 (页 65)
- 高速计数器 (HSC) 上的编码器 (页 67)
- 工艺模块 (TM) 上的编码器 (页 70)
- PROFINET 上的 PROFIdrive 编码器 (页 72)



## 组态 - 编码器 - 驱动器上的编码器

### 与编码器之间的数据交换

在该区域中，可以组态编码器和控制器之间的数据交换：

- **报文**

在下拉列表中，选择编码器的报文。其技术数据必须与设备组态相一致。

- **输入/输出地址**

这些域显示报文的符号及绝对输入和输出地址。

### 编码器类型

在“编码器类型”(Encoder type) 框中选择编码器类型。可选择以下编码器类型：

- 线性增量式
- 线性绝对值式
- 增量式旋转式
- 绝对值旋转式

## 1.5 定位轴工艺对象

根据所选编码器类型，组态各个参数。根据所选编码器类型，组态以下参数：

编码器类型/参数		说明
<b>线性增量式</b>		
	两个增量之间的距离	在该域中，可以组态编码器两步之间的距离。
	高精度 - 增量实际值中的位数 (GN_XIST1)	在该域中，组态增量实际值高精度的位数 (GN_XIST1)。
	反转编码器方向	要反转编码器的实际值，请选中此复选框。
<b>线性绝对值式</b>		
	两个增量之间的距离	在该域中，可以组态编码器两步之间的距离。
	高精度 - 增量实际值中的位数 (GN_XIST1)	在该域中，组态增量实际值高精度的位数 (GN_XIST1)。
	高精度 - 绝对实际值中的位数 (Gn_XIST2)	在该域中，组态高精度绝对值倍增系数的预留位数 (Gn_XIST2)。
	反转编码器方向	要反转编码器的实际值，请选中此复选框。
<b>增量式旋转式</b>		
	单转步数	在该域中，组态编码器每一转可以解析出的步数。
	高精度 - 增量实际值中的位数 (GN_XIST1)	在该域中，组态增量实际值高精度的位数 (GN_XIST1)。
	反转编码器方向	要反转编码器的实际值，请选中此复选框。
<b>绝对值旋转式</b>		
	单转步数	在该域中，组态编码器每一转可以解析出的步数。
	转数	在该域中，组态绝对值编码器可以检测出的转数。
	高精度 - 增量实际值中的位数 (GN_XIST1)	在该域中，组态增量实际值高精度的位数 (GN_XIST1)。
	高精度 - 绝对实际值中的位数 (Gn_XIST2)	在该域中，组态高精度绝对值倍增系数的预留位数 (Gn_XIST2)。
	反转编码器方向	要反转编码器的实际值，请选中此复选框。

## 组态 - 编码器 - 高速计数器 (HSC) 上的编码器

### 选择高速计数器 (HSC)

在“选择高速计数器”(Select high-speed counter) 框中，选择编码器将向其传输实际值的高速计数器。

检查所使用的两个高速计数器数字量输入的过滤时间。  
过滤时间应尽可能短，以便脉冲可以可靠地记录。

### HSC 接口

在“操作模式”(Operating mode) 框中，选择高速计数器的操作模式。

根据操作模式组态各个输入：

操作模式/参数		说明
<b>两阶段</b>		
	正向时钟脉冲发生器	在此域中选择用于加计数的脉冲输出。 可以通过符号地址或将其分配给绝对地址来选择输出。 输入的频率和位置（板载，信号板）显示在地址框的旁边。
	反向时钟脉冲发生器	在此域中选择用于减计数的脉冲输出。 可以通过符号地址或将其分配给绝对地址来选择输出。 输入的频率和位置（板载，信号板）显示在地址框的旁边。
<b>A/B 计数器 / A/B 四重计数器</b>		
	时钟脉冲发生器 A	在此域中为 <b>A</b> 相信号选择脉冲输出。 可以通过符号地址或将其分配给绝对地址来选择输出。 输入的频率和位置（板载，信号板）显示在地址框的旁边。
	时钟脉冲发生器 B	在此域中为 <b>B</b> 相信号选择脉冲输出。 可以通过符号地址或将其分配给绝对地址来选择输出。 输入的频率和位置（板载，信号板）显示在地址框的旁边。

## 编码器类型

在“编码器类型”(Encoder type) 框中选择编码器类型。可选择以下编码器类型：

- 线性增量式
- 增量式旋转式

根据所选编码器类型，组态各个参数。根据所选编码器类型，组态以下参数：

编码器类型/参数	说明
<b>线性增量式</b>	
两个增量之间的距离	在该域中，可以组态编码器两步之间的距离。
高精度 - 增量实际值中的位数 (GN_XIST1)	在该域中，组态增量实际值高精度的位数 (GN_XIST1)。
反转编码器方向	要反转编码器的实际值，请选中此复选框。
<b>增量式旋转式</b>	
单转步数	在该域中，组态编码器每一转可以解析出的步数。
高精度 - 增量实际值中的位数 (GN_XIST1)	在该域中，组态增量实际值高精度的位数 (GN_XIST1)。
反转编码器方向	要反转编码器的实际值，请选中此复选框。

## 组态 - 编码器 - 工艺模块 (TM) 上的编码器

### 选择工艺模块 (TM)

在“工艺模块 (TM)”(Technology module (TM)) 框中，选择要连接编码器的工艺模块。

### 与编码器之间的数据交换

在该区域中，可以组态编码器和控制器之间的数据交换：

- 报文

在下拉列表中，选择编码器的报文。其技术数据必须与设备组态相一致。

- 输入/输出地址

这些域显示报文的符号及绝对输入和输出地址。

### 编码器类型

在“编码器类型”(Encoder type) 框中选择编码器类型。可选择以下编码器类型：

- 线性增量式
- 线性绝对值式
- 增量式旋转式
- 绝对值旋转式

根据所选编码器类型，组态各个参数。根据所选编码器类型，组态以下参数：

编码器类型/参数		说明
<b>线性增量式</b>		
	两个增量之间的距离	在该域中，可以组态编码器两步之间的距离。
	高精度 - 增量实际值中的位数 (GN_XIST1)	在该域中，组态增量实际值高精度的位数 (GN_XIST1)。
	反转编码器方向	要反转编码器的实际值，请选中此复选框。
<b>线性绝对值式</b>		
	两个增量之间的距离	在该域中，可以组态编码器两步之间的距离。
	高精度 - 增量实际值中的位数 (GN_XIST1)	在该域中，组态增量实际值高精度的位数 (GN_XIST1)。
	高精度 - 绝对实际值中的位数 (Gn_XIST2)	在该域中，组态高精度绝对值倍增系数的预留位数 (Gn_XIST2)。
	反转编码器方向	要反转编码器的实际值，请选中此复选框。
<b>增量式旋转式</b>		
	单转步数	在该域中，组态编码器每一转可以解析出的步数。
	高精度 - 增量实际值中的位数 (GN_XIST1)	在该域中，组态增量实际值高精度的位数 (GN_XIST1)。
	反转编码器方向	要反转编码器的实际值，请选中此复选框。
<b>绝对值旋转式</b>		
	单转步数	在该域中，组态编码器每一转可以解析出的步数。
	转数	在该域中，组态绝对值编码器可以检测出的转数。
	高精度 - 增量实际值中的位数 (GN_XIST1)	在该域中，组态增量实际值高精度的位数 (GN_XIST1)。
	高精度 - 绝对实际值中的位数 (Gn_XIST2)	在该域中，组态高精度绝对值倍增系数的预留位数 (Gn_XIST2)。
	反转编码器方向	要反转编码器的实际值，请选中此复选框。

## 组态 - 编码器 - PROFINET 上的 PROFIdrive 编码器

### 选择编码器

在“PROFIdrive 编码器”(PROFIdrive encoder) 框中，选择 PROFINET 上的 PROFIdrive 编码器。

### 与编码器之间的数据交换

在该区域中，可以组态编码器和控制器之间的数据交换：

- 报文

在下拉列表中，选择编码器的报文。其技术数据必须与设备组态相一致。

- 输入/输出地址

这些域显示报文的符号及绝对输入和输出地址。

### 编码器类型

在“编码器类型”(Encoder type) 框中选择编码器类型。可选择以下编码器类型：

- 线性增量式
- 线性绝对值式
- 增量式旋转式
- 绝对值旋转式



根据所选编码器类型，组态各个参数。根据所选编码器类型，组态以下参数：

编码器类型/参数		说明
<b>线性增量式</b>		
	两个增量之间的距离	在该域中，可以组态编码器两步之间的距离。
	高精度 - 增量实际值中的位数 (GN_XIST1)	在该域中，组态增量实际值高精度的位数 (GN_XIST1)。
	反转编码器方向	要反转编码器的实际值，请选中此复选框。
<b>线性绝对值式</b>		
	两个增量之间的距离	在该域中，可以组态编码器两步之间的距离。
	高精度 - 增量实际值中的位数 (GN_XIST1)	在该域中，组态增量实际值高精度的位数 (GN_XIST1)。
	高精度 - 绝对实际值中的位数 (Gn_XIST2)	在该域中，组态高精度绝对值倍增系数的预留位数 (Gn_XIST2)。
	反转编码器方向	要反转编码器的实际值，请选中此复选框。
<b>增量式旋转式</b>		
	单转步数	在该域中，组态编码器每一转可以解析出的步数。
	高精度 - 增量实际值中的位数 (GN_XIST1)	在该域中，组态增量实际值高精度的位数 (GN_XIST1)。
	反转编码器方向	要反转编码器的实际值，请选中此复选框。
<b>绝对值旋转式</b>		
	单转步数	在该域中，组态编码器每一转可以解析出的步数。
	转数	在该域中，组态绝对值编码器可以检测出的转数。
	高精度 - 增量实际值中的位数 (GN_XIST1)	在该域中，组态增量实际值高精度的位数 (GN_XIST1)。
	高精度 - 绝对实际值中的位数 (Gn_XIST2)	在该域中，组态高精度绝对值倍增系数的预留位数 (Gn_XIST2)。
	反转编码器方向	要反转编码器的实际值，请选中此复选框。

## 1.5 定位轴工艺对象

### 1.5.4.4 扩展参数

#### 机械

#### 组态 - 机械 - PTO（脉冲串输出）

在“机械”(Mechanics) 组态窗口中组态驱动器的机械属性。

#### 电机每转的脉冲数

在此框中组态电机每转所需的脉冲数。

限值（与所选测量单位无关）：

- $0 < \text{电机每转的脉冲数} \leq 2147483647$

#### 电机每转的距离 (Distance per motor revolution)

在此框中，组态电机每转带动单元的机械系统行进的负载距离。

限值（与所选测量单位无关）：

- $0.0 < \text{每转距离} \leq 1.0e12$

#### 允许的旋转方向（自 V4 工艺版本起）

通过组态此框可决定系统机械是同时朝两个方向运动，还是只朝正向或负向运动。

如果尚未在“PTO（脉冲 A 和方向 B）”模式下激活脉冲发生器的方向输出，则选择受限于正方向或负方向。

## 反转方向

可使用“反转方向”(Invert direction) 复选框根据驱动器的方向逻辑对控制系统进行调整。

方向逻辑将根据所选脉冲发生器的模式反转：

- **PTO (脉冲 A 和方向 B)**

- 方向输出上输出 0 V ⇒ 正向旋转
- 方向输出上输出 5 V/24 V ⇒ 负向旋转

指定的电压取决于所使用的硬件。各指示值并不适用于 CPU 1217 的差分输出。

- **PTO (向上计数 A, 向下计数 B)**

“向下脉冲输出”和“向上脉冲输出”这两个输出相互交换。

- **PTO (A/B 相移)**

“A 相”和“B 相”输出相互交换。

- **“PTO (A/B 相移 - 四倍频)**

“A 相”和“B 相”输出相互交换。

## 组态 - 机械 - PROFIdrive/模拟驱动器接口

在“机械”(Mechanics) 组态窗口中组态驱动器的机械属性及其编码器。

## 编码器安装方式

在下拉列表中，选择如何将编码器安装在机械机构上。支持下列编码器安装类型：

- 在电机轴上
- 外部测量系统

## 位置参数

根据所选编码器安装类型，组态以下位置参数：

编码器安装类型/位置参数	说明
<b>在电机轴上</b>	
电机每转的负载运动	在该域中，组态电机每转的负载距离。
<b>外部测量系统</b>	
电机每转的负载运动	在该域中，组态电机每转的负载距离。
编码器每转的距离	在该域中，组态编码器每旋转一圈外部测量系统所记录的距离。

## 组态 - 模数 (仅限 PROFIdrive/模拟驱动器接口)

如果轴只沿一个方向移动，则位置值将持续增大。可以使用“模数”(modulo) 设置将位置值限制到递归参考系统。

启用“模数”(modulo) 时，  
将使用一个递归模数范围表示工艺对象的位置值。模数范围由起始值和长度定义。

例如，要将轴的位置值限制为一整圈，可将模数范围定义为起始值 = 0°、  
长度 = 360°。编码器精度为 0.1°/编码器步时，位置值的模数范围是 0.0°到 359.9°。

## 启用模数

选择“启用模数”(Enable modulo) 复选框，  
为轴使用递归参考系统（例如 0.0° 到 359.9°）。

## 模数起始值

在该域中，定义模数运算范围的起始位置（例如 0°）。

## 模数长度

在该域中，定义模数范围的长度（例如 360°）。

## 位置限制

### 对硬件限位开关的要求

仅使用逼近后始终保持切换的硬件限位开关。  
只有在返回到有效行程范围后，才会取消这种切换状态。

### 参见

组态 - 位置限制 (页 77)

触发位置限制时的轴操作 (页 79)

在用户程序中更改位置限制的组态 (页 81)

### 组态 - 位置限制

在“位置限制”(Position limits) 组态窗口中组态轴的硬件和软限位开关。

### 启用硬限位开关

使用此复选框可激活下限和上限硬限位开关的功能。  
硬限位开关可用于在归位过程中反转行进方向。  
有关详细信息，请参见归位的组态说明。

### 下限/上限硬限位开关输入

从下拉列表中为下限或上限硬限位开关选择数字量输入。该输入必须具有中断功能。  
数字量板载 CPU 输入和所插入信号板的数字量输入可以选作硬限位开关的输入。

---

#### 说明

默认情况下，数字量输入的滤波时间设置为 6.4 ms。  
如果将这种数字量输入用作硬限位开关的输入，则可能会发生意外减速情况。  
如果出现这种情况，则需降低相关数字量输入的滤波时间。  
可以在数字量输入设备组态的“输入滤波器”(Input filter) 中设置滤波时间。

---

## 1.5 定位轴工艺对象

### 选择信号电平

在此下拉列表中，可以选择逼近硬限位开关时 CPU 输入端存在的信号电平。

- 选择“低电平”(Low level) (常闭触点)

CPU 输入端电平为 0 V (FALSE) 时表示已逼近硬限位开关

- 选择“高电平”(High level) (常开触点)

CPU 输入端电平为 5 V/24 V (TRUE) =  
已逼近硬限位开关 (实际电压取决于使用的硬件)

### 启用软限位开关

使用此复选框可激活下限和上限软限位开关的功能。

---

#### 说明

启用的软限位开关仅影响已回到原点的轴。

---

### 上限和下限软限位开关

在这些框中可输入下限和上限软限位开关的位置值。

限值 (与所选测量单位无关) :

- $-1.0e12 \leq$  下限软限位开关  $\leq 1.0e12$
- $-1.0e12 \leq$  上限软限位开关  $\leq 1.0e12$

上限软限位开关的值必须大于等于下限软限位开关的值。

### 参见

对硬件限位开关的要求 (页 77)

触发位置限制时的轴操作 (页 79)

在用户程序中更改位置限制的组态 (页 81)

组态 - 回原点 - 主动 (页 91)

## 触发位置限制时的轴操作

### 逼近硬件限位开关时的轴操作

逼近硬件限位开关时，根据驱动器连接，轴的动作会不同：

- 基于 PROFIdrive/模块量输出的驱动器连接：

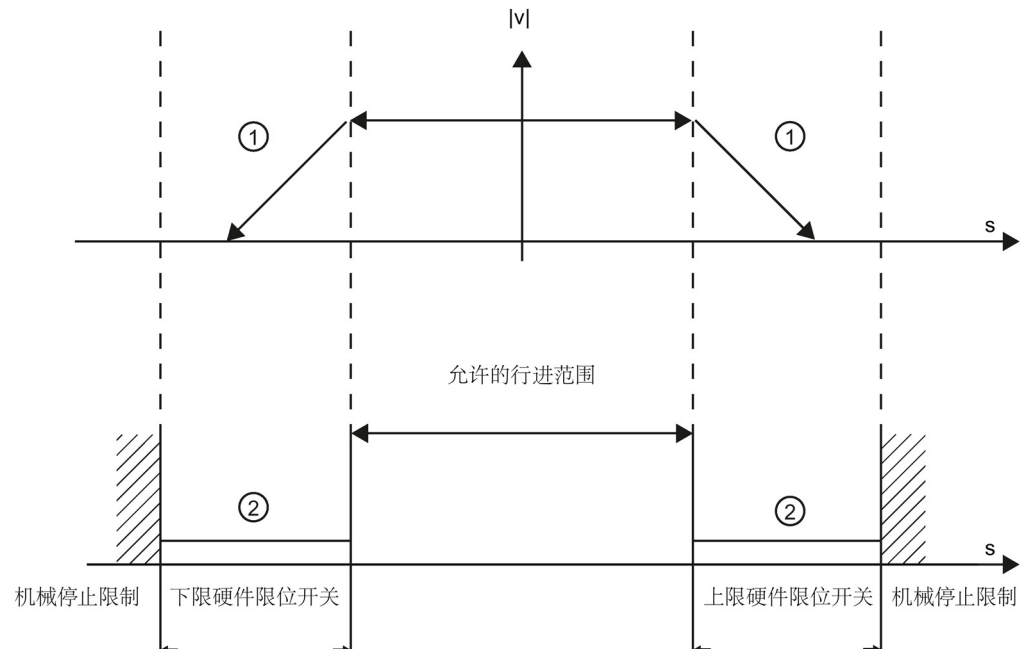
逼近硬限位开关时，轴将禁用，并根据驱动器中的组态进行制动，并转入停止状态。为此，必须选择足够大的驱动器减速度，以使轴在机械挡块前可靠停止。

- 驱动器连接方式 PTO (Pulse Train Output)

逼近硬限位开关时，轴将以所组态的急停减速度制动直到停止。

为此，必须选择足够大的急停减速度，以使轴在机械挡块前可靠停止。

下图显示了轴逼近硬件限位开关后的轴操作：



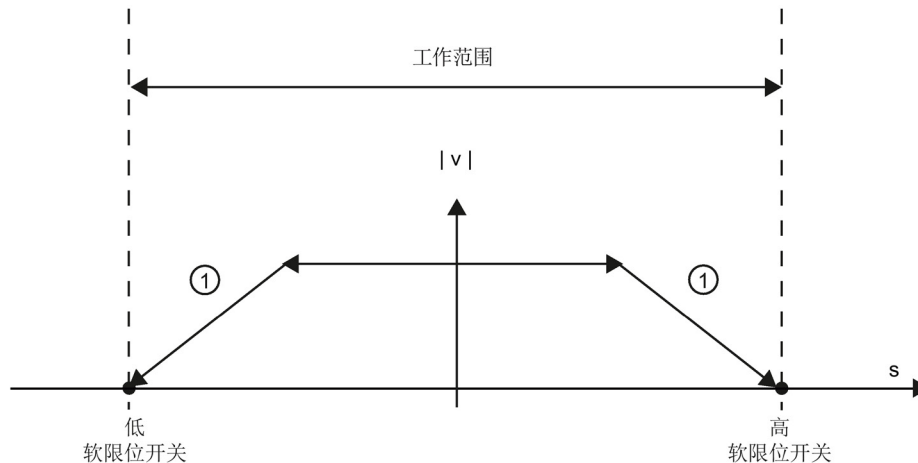
①	轴以所组态的急停减速度制动直到停止。
②	硬件限位开关产生“已逼近”状态信号的范围。

“已逼近硬件限位开关”错误将显示在待启动的运动控制指令、“MC\_Power”和工艺对象变量中。有关用于清除错误的指令，请参见附录中的“ErrorID 和 ErrorInfo 列表”。

## 到达软件限位开关时的轴操作

如果软件限位开关激活，则在软件限位开关所在的位置将停止当前的运动。  
轴将以所组态的减速度制动。

下图显示了轴到达软件限位开关前的轴操作：



① 轴将以所组态的减速度制动直到停止。

“已逼近软限位开关”错误将显示在正在启动的运动控制指令、“MC\_Power”和工艺对象变量中。有关用于清除错误的指令，请参见附录中的“ErrorID 和 ErrorInfo 列表”。

超出软限位开关时，根据驱动器连接，轴的动作会不同：

- **基于 PROFIdrive/模块量输出的驱动器连接：**

超出软限位开关时，轴将禁用，并根据驱动器中的组态进行制动，并转入停止状态。

- **驱动器连接方式 PTO (Pulse Train Output)**

有关超出软限位开关时的轴的动作，请参见“软件限位开关与回原点操作结合使用 (页 197)”和“软件限位开关与动态更改结合使用 (页 202)”部分。

如果机械停止块位于软件限位开关的后面并且有发生机械损坏的风险，则需要使用附加的硬件限位开关。

## 参见

对硬件限位开关的要求 (页 77)

组态 - 位置限制 (页 77)

在用户程序中更改位置限制的组态 (页 81)



## 在用户程序中更改位置限制的组态

可以在 CPU 中运行用户程序期间更改下列组态参数：

### 硬限位开关

在用户程序运行期间，还可以激活和禁用硬限位开关。

要执行该操作，可使用以下工艺对象变量：

- `<轴名称>.PositionLimitsHW.Active`

有关组态参数的更改何时生效的信息，请参见附录中的工艺对象变量 (页 235)说明。

### 软限位开关

在用户程序运行期间，还可以激活和禁用软限位开关并更改其位置值。

要执行该操作，可使用以下工艺对象变量：

- `<轴名称>.PositionLimitsSW.Active`  
用于激活和取消激活软限位开关
- `<轴名称>.PositionLimitsSW.MinPosition`  
用于更改下限软限位开关的位置
- `<轴名称>.PositionLimitsSW.MaxPosition`  
用于更改上限软限位开关的位置

有关组态参数的更改何时生效的信息，请参见附录中的工艺对象变量说明。

## 参见

变量兼容性列表 (页 40)

MC\_ChangeDynamic: 更改轴的动态设置 V4 及更高版本 (页 409)

对硬件限位开关的要求 (页 77)

组态 - 位置限制 (页 77)

触发位置限制时的轴操作 (页 79)

## 动态

### 组态 - 动态 - 常规

可以在“常规动态”(General dynamics) 组态窗口中组态轴的最大速度、启动/停止速度、加速度和减速度以及加加速度限值（自定位轴工艺对象 V2 起）。

### 速度限值的单位

从此下拉列表中可选择要用于设置速度限值的测量单位。这里设置的单位取决于“组态 - 常规”(Configuration - General) 下设置的测量单位，设置此单位仅仅是为了便于输入。

### 最大速度/启动/停止速度

在这些框中定义轴的最大允许速度和启动/停止速度。  
启动/停止速度是轴的最小允许速度，只能组态用于通过 PTO (Pulse Train Output) 的驱动器连接。对于通过 PROFIdrive 或模拟量输出的驱动器连接，启动/停止速度固定为 0。

限值：

下述限值采用“脉冲/秒”作为测量单位：

- **定位轴工艺对象 V4 或更高版本**
  - $1 \leq \text{启动/停止速度} \leq 20000$ （信号板 20 kHz）
    - $1 \leq \text{启动/停止速度} \leq 200000$ （信号板 200 kHz）
    - $1 \leq \text{启动/停止速度} \leq 100000$ （板载 CPU 输出 100 kHz）
    - $1 \leq \text{启动/停止速度} \leq 30000$ （板载 CPU 输出 30 kHz）
    - $1 \leq \text{启动/停止速度} \leq 1000000$ （板载 CPU 输出 1 MHz CPU 1217）
  - $1 \leq \text{最大速度} \leq 20000$ （信号板 20 kHz）
    - $1 \leq \text{最大速度} \leq 200000$ （信号板 200 kHz）
    - $1 \leq \text{最大速度} \leq 100000$ （板载 CPU 输出 100 kHz）
    - $1 \leq \text{最大速度} \leq 30000$ （板载 CPU 输出 30 kHz）
    - $1 \leq \text{最大速度} \leq 1000000$ （板载 CPU 输出 1 MHz CPU 1217）

有关工艺对象定位轴 < V4 的限值，请参见附录与运动控制相关的 CPU 输出（工艺版本 V1...3）（页 290）。

最大速度值必须大于等于启动/停止速度值。

用户必须对其它测量单位所对应的限值进行相应地转换，以便符合指定的机械系统。

### 加速度/减速度 - 加速时间/减速时间

在“加速时间”(Ramp-up time) 或“加速度”(Acceleration) 框中，可在设置所需加速度。所需要的减速度可以在“减速时间”(Ramp-down time) 或“减速度”(Deceleration) 框中设置。

加速时间与加速度、减速时间与减速度之间的关系如下面的方程所示：

$$\text{加速时间} = \frac{\text{最大速度} - \text{启动/停止速度}}{\text{加速度}}$$

$$\text{减速时间} = \frac{\text{最大速度} - \text{启动/停止速度}}{\text{减速度}}$$

用户程序中启动的运动作业将使用所选加速度/减速度执行。

有关通过基于 PTO (Pulse Train Output) 的驱动器连接时的加速度和减速度的限值，请参见 与运动控制相关的 CPU 输出 (页 15) 部分。

---

#### 说明

更改速度限值（“启动/停止速度”和“最大速度”）将影响轴的加速度和减速度值。加速时间和减速时间保持不变。

---

### 启用加加速度限值（自定位轴工艺对象 V2 起）

使用该复选框，可启用加加速度限值。

---

#### 说明

如果发生错误，轴将以组态的急停减速度减速。此时不考虑启用的加加速度限值。

---

## 平滑时间/加加速度（自定位轴工艺对象 V2 起）

在“平滑时间”(Smoothing time) 框或“加加速度”(Jerk) 框中，可以输入加加速度的参数。

- 在“加加速度”(Jerk) 框中，可以为加速和减速斜坡设置所需的加加速度。
- 在“平滑时间”(Smoothing time) 框中，可以为加速斜坡设置所需的平滑时间。

## 说明

## 平滑时间 V2...3

在组态中显示的所设置的平滑时间仅适用于加速斜坡。

如果加速度值和减速度值不同，则根据加速斜坡的冲击计算减速斜坡的平滑时间并使用该平滑时间。（另请参见使用冲击限制时轴的行为 (页 87)）

如下所述调整减速度的平滑时间：

- **加速度 > 减速度**  
减速斜坡使用的平滑时间比加速斜坡使用的平滑时间短。
- **加速度 < 减速度**  
减速斜坡使用的平滑时间比加速斜坡使用的平滑时间长。
- **加速度 = 减速度**  
加速和减速斜坡的平滑时间相等。

平滑时间和冲击之间的关系如下面的方程所示：

$$\text{取整时间 (加速斜坡)} = \frac{\text{加速度}}{\text{步进}}$$

$$\text{取整时间 (减速斜坡)} = \frac{\text{减速度}}{\text{步进}}$$

用户程序中启动的运动作业将使用所选冲击执行。

有关通过基于 PTO (Pulse Train Output) 的驱动器连接时的加加速度限值，请参见 与运动控制相关的 CPU 输出 (页 15) 部分。

## 参见

使用冲击限制时轴的行为 (页 87)

用于运动控制的硬件组件 (页 11)

与运动控制相关的 CPU 输出 (页 15)

组态 - 动态 - 急停 (页 86)

在用户程序中可以更改动态组态 (页 89)

## 组态 - 动态 - 急停

在“动态急停”(Dynamics emergency stop) 组态窗口中，可以组态轴的急停减速度。  
出现错误或者禁用轴时，通过运动控制指令“MC\_Power”  
(输入参数StopMode = 0 或 2) 可以使用该减速度将轴制动至停止状态。

## 速度

“常规动态”(General dynamics) 组态窗口中组态的速度值，将再次显示在该信息区域中。

## 减速度

可在“急停减速度”(Emergency stop deceleration) 或“急停减速时间”(Emergency stop ramp-down time) 字段中设置急停减速度值。

急停减速时间和急停减速度之间的关系如下面的方程所示：

$$\text{急停减速时间} = \frac{\text{最大速度} - \text{启动/停止速度}}{\text{急停减速度}}$$

指定的急停减速度必须足够大，保证可以在出现紧急情况时（例如，在到达机械挡块前逼近硬限位开关时）及时使轴停止。

必须基于组态的最大轴速度选择急停减速度。

限值：

下述限值采用“脉冲/秒<sup>2</sup>”作为测量单位。

- 自 CPU 固件 V3 起  
 $0.005 \leq \text{急停减速度} \leq 9.5E9$
- CPU 固件 V1...2  
 $0.28 \leq \text{急停减速度} \leq 9.5E9$

必须对其它测量单位所对应的限值进行相应地转换，以便符合指定的机械系统。

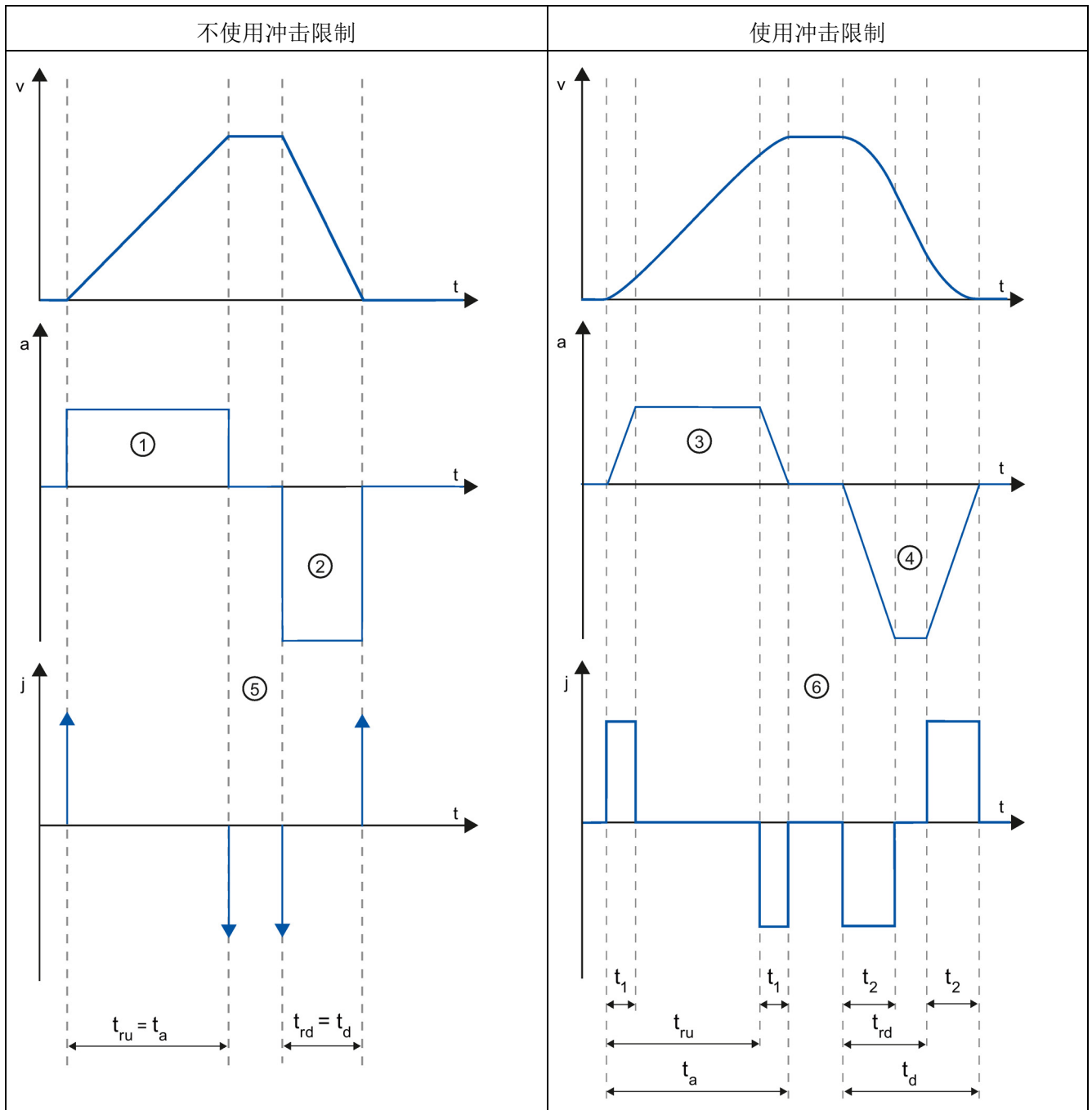
## 参见

组态 - 动态 - 常规 (页 82)

在用户程序中可以更改动态组态 (页 89)

## 使用冲击限制时轴的行为

如果激活了加加速度限值，则不会突然停止轴加速和轴减速，而是根据设置的步进或平滑时间逐渐调整。下图详细显示了在激活和不激活冲击限制的情况下轴的行为：



## 1.5 定位轴工艺对象

t	时间轴
v	速度
a	加速度
j	加加速度
$t_{ru}$	加速时间
$t_a$	轴加速所用时间
$t_{rd}$	减速时间
$t_d$	轴减速所用时间
$t_1$	加速斜坡的平滑时间
$t_2$	减速斜坡的平滑时间

本示例说明减速度值 ② 是加速度值 ① 的二倍的行程。此时，得出的减速时间  $t_{rd}$  仅是加速时间  $t_{ru}$  的一半。

如果没有激活冲击限制，加速度 ① 和减速度 ② 将发生突变。

如果激活了冲击限制器，则加速度 ① 和减速度 ② 将逐渐改变。

由于冲击适用于整个运动，加速度的加速率和减速度的减速率相同。

在不使用冲击限制的情况下进行更改时，步进值 j 将变得无穷大 ⑤。

如果激活了冲击限制，则步进将被限制为组态值 ⑥。

组态中给出的平滑时间  $t_1$  适用于加速斜坡。减速斜坡平滑时间  $t_2$  会根据组态的加加速度值以及组态的减速度进行计算。

## 参见

组态 - 动态 - 常规 (页 82)



## 在用户程序中可以更改动态组态

可以在 CPU 中运行用户程序期间更改下列组态参数：

### 加速度和减速度

在用户程序运行期间，还可以更改加速度和减速度值。  
要执行该操作，可使用以下工艺对象变量：

- `<轴名称>.DynamicDefaults.Acceleration`  
用于更改加速度
- `<轴名称>.DynamicDefaults.Deceleration`  
用于更改减速度

有关组态参数的更改何时生效的信息，请参见附录中的工艺对象变量 (页 235)说明。

### 急停减速度

在用户程序运行期间，还可更改急停减速度值。  
要执行该操作，可使用以下工艺对象变量：

- `<轴名称>.DynamicDefaults.EmergencyDeceleration`

有关组态参数的更改何时生效的信息，请参见附录中的工艺对象变量说明。

---

### 说明

更改该参数后，可能需要调整硬限位开关的位置以及其它安全相关的设置。

---

## 1.5 定位轴工艺对象

### 加加速度限值

还可在用户程序运行时激活和禁用冲击限制和更改冲击值。为此，请使用工艺对象变量 `<轴名称>.DynamicDefaults.Jerk`。

如果为加加速度输入的值  $> 0.004$  个脉冲/s<sup>3</sup>，则加加速度限值由该指定值激活。

如果为加加速度输入的值  $= 0.0$ ，则取消激活加加速度限值。

有关组态参数的更改何时生效的信息，请参见附录中的工艺对象变量说明。

### 参见

在用户程序中更改动态值的组态（“轴”工艺对象 V1...3） (页 305)

变量兼容性列表 (页 40)

MC\_ChangeDynamic: 更改轴的动态设置 V4 及更高版本 (页 409)

组态 - 动态 - 常规 (页 82)

组态 - 动态 - 急停 (页 86)

## 回原点（自定位轴工艺对象 V2 起）

### 组态 - 回原点 - 主动

在“主动归位”(Active homing) 组态窗口中组态主动归位所需的参数。  
运动控制指令“MC\_Home”的输入参数“Mode”= 3 时，会启动主动归位。

### 选择归位模式（仅限通过 PROFIdrive V5 或更高版本的驱动器连接）

执行以下任一归位模式：

- 使用基于 PROFIdrive 报文和接近开关的零位标记
- 使用基于 PROFIdrive 报文的零位标记
- 使用基于数字量输入的归位标记

如果选择了通过 PTO (Pulse Train Output)  
或模拟量输出的驱动器连接，则默认使用基于数字量输入的归位标记。

## 数字量输入

在此区域中，可组态归位开关：

- **归位开关输入**

在此域中为归位开关选择数字量输入。

---

### 说明

默认情况下，数字量输入的滤波时间设置为 6.4 ms。

数字量输入用作归位开关的输入时，可能引起意外减速，从而导致出现误差。

根据归位速度和归位开关的范围，可能检测不到归位位置。

可以在数字量输入设备组态的“输入滤波器”(Input filter) 中设置滤波时间。

指定的滤波时间必须小于归位开关的输入信号持续时间。

---

通过 PTO (Pulse Train Output) 的驱动器连接：

该输入必须具有中断功能。板载 CPU

输入和所插入信号板输入都可选作归位开关的输入。

- **选择信号电平**

在下拉列表中，选择归位时使用的归位开关电平。

- **允许在硬限位开关处自动反向**

激活该复选框可将硬限位开关用作归位过程中的反向凸轮。

只有启用硬限位开关才能实现反向控制（必须至少组态位于逼近方向上的硬限位开关）。

如果在主动归位过程中到达硬限位开关，轴将以组态的减速度（不是以急停减速度）制动，然后反向。然后反向检测归位开关。

如果未激活反向功能且在主动归位过程中轴到达硬限位开关，则将因错误而中止归位过程并以急停减速度对轴进行制动。

---

### 说明

如果可能，采用以下措施之一以确保机器在发生反向时不会行进到机械挡块：

- 保持较低的行进速度。
  - 增加组态的加速度/减速度。
  - 增加硬限位开关和机械挡块之间的距离。
-

## 逼近/归位方向

通过方向选择，可以决定主动归位过程中搜索归位开关的逼近方向以及归位的方向。归位方向指定执行归位操作时轴用于逼近组态的归位开关端的行进方向。

## 归位开关侧

在此处可以选择轴是在归位开关的上侧还是下侧进行归位。

## 逼近速度

在该域中，可以指定归位期间搜索归位开关的速度。

限值（与所选测量单位无关）：

- 启动/停止速度  $\leq$  逼近速度  $\leq$  最大速度

## 归位速度

在该域中，可以指定归位期间逼近归位开关的速度。

限值（与所选测量单位无关）：

- 启动/停止速度  $\leq$  归位速度  $\leq$  最大速度

## 起始位置偏移值

如果指定的归位位置与归位开关的位置存在偏差，则可在该域中指定起始位置偏移量。

如果该值不等于 0，轴在归位开关处归位后将执行以下动作：

1. 以归位速度使轴移动起始位置偏移值指定的一段距离
2. 达到“起始位置偏移值”时，轴处于运动控制指令“MC\_Home”的输入参数“Position”中指定的起始位置处。

限值（与所选测量单位无关）：

- $-1.0e12 \leq$  起始位置偏移值  $\leq 1.0e12$

## 起始位置

将运动控制指令“MC\_Home”中所组态的位置用作起始位置。

## 组态 - 回原点 - 被动

在“归位 - 被动”(Homing - Passive) 组态窗口中，可以组态被动归位所需的参数。

被动归位的移动必须由用户触发（例如，使用轴运动命令）。

运动控制指令“MC\_Home”的输入参数“Mode”= 2 时，会启动被动归位。

## 选择归位模式（仅限通过 PROFIdrive V5 或更高版本的驱动器连接）

执行以下任一归位模式：

- **使用基于 PROFIdrive 报文和接近开关的零位标记**

系统将检查到达接近开关的时间。在到达接近开关并置于指定的归位方向后，可通过 PROFIdrive 报文启用零位标记检测。

在预先选定的方向上到达零位标记后，会将工艺对象的实际位置设置为归位标记位置。

- **使用基于 PROFIdrive 报文的零位标记**

当工艺对象的实际值按照指定的归位方向移动时，系统将立即启用零位标记检测。

在指定的归位方向上到达零位标记后，会将工艺对象的实际位置设置为归位标记位置。

- **使用基于数字量输入的归位标记**

当轴或编码器的实际值在指定的归位方向上移动时，系统将立即检查数字量输入的状态。

在指定的归位方向上到达归位标记（数字量输入的设置）后，会将工艺对象的实际位置设置为归位标记位置。

如果选择了通过 PTO (Pulse Train Output) 的驱动器连接，则默认使用基于数字量输入的归位标记。

## 数字量输入

在此区域中，可组态归位开关：

- **归位开关输入**

在此域中为归位开关选择数字量输入。该输入必须具有中断功能。  
板载 CPU 输入和所插入信号板输入都可选作归位开关的输入。

---

### 说明

默认情况下，数字量输入的滤波时间设置为 6.4 ms。

数字量输入用作归位开关的输入时，可能引起意外减速，从而导致出现误差。

根据归位速度和归位开关的范围，可能检测不到归位位置。

可以在数字量输入设备组态的“输入滤波器”(Input filter) 中设置滤波时间。

指定的滤波时间必须小于归位开关的输入信号持续时间。

---

- **选择信号电平**

在下拉列表中，选择归位时使用的归位开关电平。

## 归位开关侧

在此处可以选择轴是在归位开关的上侧还是下侧进行归位。

## 起始位置

将运动控制指令“MC\_Home”中所组态的位置用作起始位置。

---

### 说明

如果未使用轴运动命令进行被动归位（轴处于停止状态），则将在下一个归位开关的上升沿或下降沿处执行归位操作。

---

## 顺序 - 主动回原点

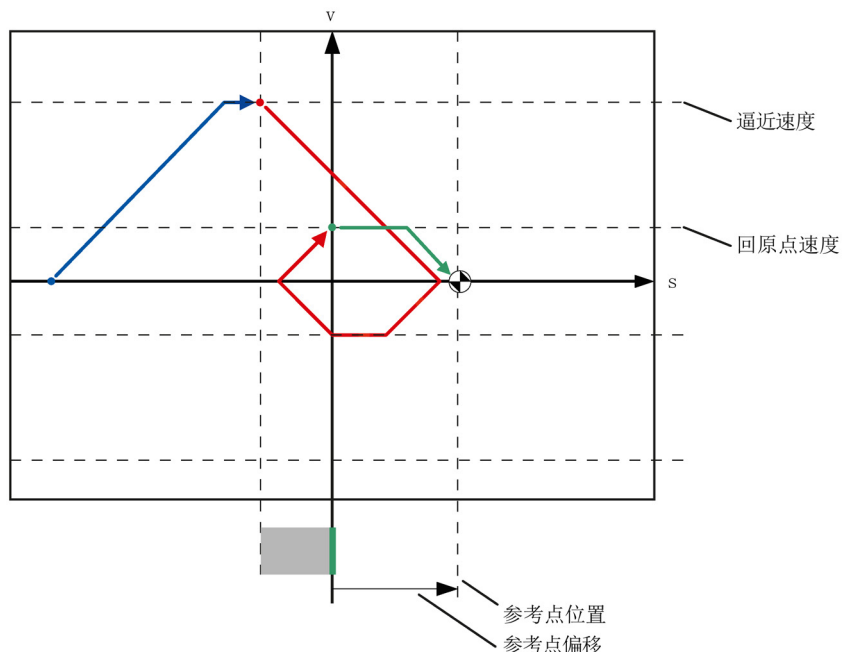
使用运动控制指令“MC\_Home”（输入参数 Mode = 3），可以启动主动归位。

“Position”输入参数指定绝对归位位置。

另外，若要进行测试，也可在轴控制面板中启动主动归位。

下图举例说明了使用以下组态参数时主动起始位置逼近的特征曲线：

- “归位模式”=“使用基于数字量输入的归位标记”
- “逼近/归位方向”=“正方向”
- “归位开关侧”=“上侧”
- “归位位置偏移”值 > 0



### 搜索归位开关（蓝色曲线部分）

主动归位开始后，轴加速到组态的“逼近速度”并以该速度搜索归位开关。

变量 <轴名称>.StatusBits.HomingDone 设置为 FALSE。



### 参考点逼近（红色曲线段）

检测到归位开关时，本示例中的轴将制动并反向，以组态的归位速度向组态的归位开关侧归位。归位运动会使变量 `<轴名称>.StatusBits.HomingDone` 变为 TRUE。

### 行进到归位位置偏移值（绿色曲线段）

归位后，轴以归位速度沿该路径移动到归位位置偏移值。  
此时，轴处于运动控制指令“MC\_Home”的输入参数“Position”中指定的归位位置。

### 参见

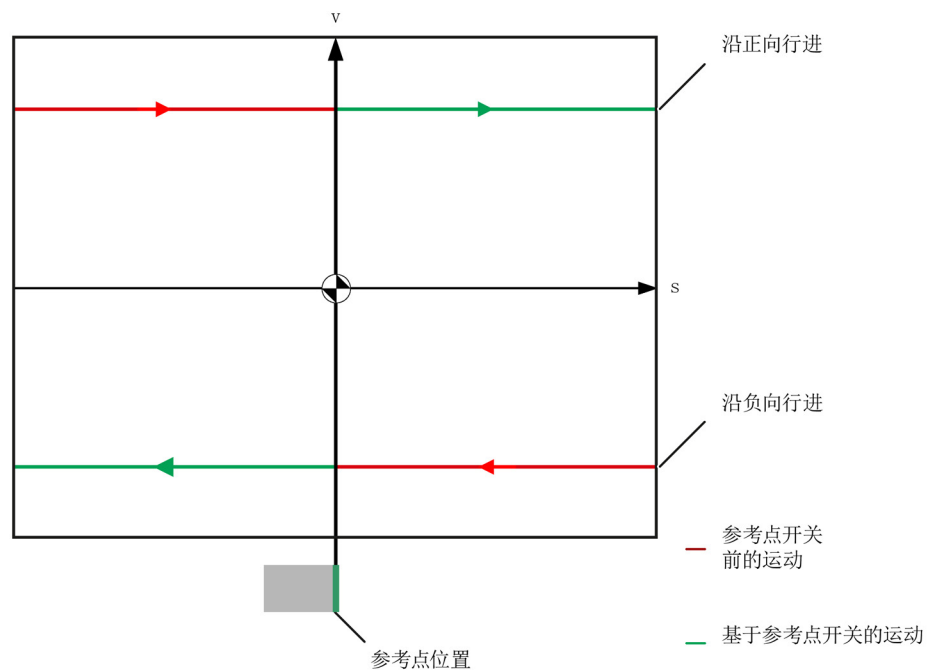
组态 - 回原点 - 常规（轴工艺对象 V2...3）（页 301）

### 顺序 - 被动回原点

通过运动控制指令“MC\_Home”（输入参数 `Mode = 2`）启动被动归位。  
输入参数“Position”指定绝对参考点位置。

下图举例说明了使用以下组态参数时被动归位的特性曲线：

- “归位开关侧”=“上侧”
- “归位模式”=“使用基于数字量输入的归位标记”



### 向归位开关方向移动（曲线的红色部分）

启动被动归位时，运动控制指令“MC\_Home”本身并不执行任何归位运动。必须由用户通过其它运动控制指令（如“MC\_MoveRelative”）来执行到达归位开关所需的行程。如果轴已回到原点，则被动归位过程中，变量 `<轴名称>.StatusBits.HomingDone` 保持为 TRUE。

### 轴归位（由曲线的红色部分过渡到绿色部分）

到达归位开关的组态侧时，轴会归位。将当前的轴位置设置为起始位置。这在运动控制指令“MC\_Home”的“Position”参数中指定。如果之前轴未归位，则会将变量 `<轴名称>.StatusBits.HomingDone` 设置为“TRUE”。之前启动的行程不会取消。

### 移动至超过归位开关（曲线的绿色部分）

在归位开关处归位后，轴继续移动并以正确的轴位置完成之前启动的行程。

### 在用户程序中更改回原点组态

利用版本 V2 及以上版本的定位轴工艺对象，您可以在 CPU 中运行用户程序期间更改下列组态参数：

### 被动归位

可以在用户程序运行期间更改用于被动归位的归位开关端。要执行该操作，可使用以下工艺对象变量：

- `<轴名称>.Sensor[1].PassiveHoming.SideInput`  
用于更改归位开关侧
- `<轴名称>.Sensor[1].PassiveHoming.Mode`  
用于更改归位速度

有关组态参数的更改何时生效的信息，请参见附录中的工艺对象变量 (页 235)说明。

## 主动归位

可以在用户程序运行期间更改主动归位的逼近方向、归位开关侧、逼近速度、归位速度和起始位置偏移量。要执行该操作，可使用以下工艺对象变量：

- `<轴名称>.Homing.AutoReversal`  
用于更改“在硬限位开关处自动反向”
- `<轴名称>.Homing.ApproachDirection`  
用于更改逼近/归位方向
- `<轴名称>.Sensor[1].ActiveHoming.SideInput`  
用于更改归位开关侧
- `<轴名称>.Homing.ApproachVelocity`  
用于更改逼近速度
- `<轴名称>.Homing.ReferencingVelocity`  
用于更改归位速度
- `<轴名称>.Sensor[1].ActiveHoming.HomePositionOffset`  
用于更改起始位置偏移
- `<轴名称>.Sensor[1].ActiveHoming.Mode`  
用于更改归位速度

有关组态参数的更改何时生效的信息，请参见附录中的工艺对象变量说明。

## 参见

变量兼容性列表 (页 40)

MC\_ChangeDynamic: 更改轴的动态设置 V4 及更高版本 (页 409)

## 定位监控

### 组态 - 定位监控 (仅限 PROFIdrive/模拟驱动器接口)

在“定位监控”(Positioning Monitoring) 组态窗口中，组态用于监控目标位置的标准。

定位监控功能将在设定值计算结束时对实际位置的状态进行监控。一旦速度设定值达到零值，则实际位置值必须介于定位窗口的容差时间范围内。

实际值在定位窗口内的停留时间必须超出最短停留时间。

当实际位置在容差时间内到达定位窗口，且在最短停留时间内停留在该窗口，则置位状态位 `<轴名称>.StatusBits.Done`。这样就完成了一个运动命令。

各种设定值插补操作方式中的定位监控都相同。

例如，设定值插补完成的方式如下所示：

- 设定值到达目标位置
- 通过运动控制指令“MC\_Halt”，在运动期间用位置控制停止

在以下情况下，通过定位监控停止轴，并且定位错误 (ErrorID 16#800F) 显示在运动控制指令中：

- 在容差时间内，实际值未到达定位窗口。
- 在最短停留时间内，实际值离开定位窗口。

### 定位窗口

在该域中，组态定位窗口的大小。

### 容差时间

在该域中，组态容差时间。在该容差时间内，位置值必须到达定位窗口。

### 在定位窗口停留的最短时间

在该域中，组态最短停留时间。在最短停留时间内，实际位置值必须位于定位窗口中。

## 组态 - 跟随误差 (仅限 PROFIdrive/模拟驱动器接口)

在“跟随误差”(Following error)

组态窗口中，组态轴的实际位置与位置设定值之间的容许偏差。

跟随误差是轴的位置设定值与实际位置值之间的差值。

计算跟随误差时，会将设定值到驱动器的传输时间、实际位置值到控制器的传输时间计算在内。

根据一个与速度有关的跟随误差限值对跟随误差进行监视。

允许跟随误差取决于速度设定值。

当速度小于一个可调整的速度下限时，将允许跟随误差指定为常数。

而高于该速度下限时，允许跟随误差则随速度设定值按比例增长。

在最大速度下可达到最大允许跟随误差。

超出允许跟随误差时，轴停止，错误 (ErrorID 16#800D) 显示在运动控制指令中。

## 启用跟随误差监控

如果想启用跟随误差监控功能，则选中该复选框。

在启用跟随误差监控时，轴在错误范围（橙色）内停止。

## 最大跟随误差

在该域中，组态最大速度时容许的跟随误差。

## 跟随误差

在该域中，组态低速度时的容许跟随误差（无动态调整）。

## 启动动态调整

在该域中，组态一个速度；超过该速度时，将会动态调整跟随误差。

超过该速度时，将会调整跟随误差，直至达到最大速度时的最大跟随误差。

## 最大速度

该框显示在“动态 > 常规”(Dynamics > General) 下组态的最大速度。

## 1.5 定位轴工艺对象

### 组态 - 停止信号（仅限 PROFIdrive/模拟驱动器接口）

在“停止信号”(Standstill signa) 组态窗口中，组态停止检测标准。

要显示停止 (<轴名称>.StatusBits.StandStill)，  
轴速度必须在停止窗口内保持最短停留时间。

### 停止窗口

在该域中，组态停止窗口的大小。

### 在停止窗口停留的最短时间

在该域中，组态停止窗口中的最短停留时间。

### 组态 - 控制回路（仅限 PROFIdrive/模拟驱动器接口）

在“控制回路”(Control loop) 组态窗口中，组态位置控制回路的预控制和增益 Kv。

Kv 因数影响以下参数：

- 定位精度和停止控制
- 运动的一致性
- 定位时间

轴的机械状态（硬度越高）越好，可以组态的 Kv 因数越大。

这样可以减小跟随误差，实现更快的动态响应。

通过“Tuning (页 159)”(自整定) 功能，可确定控制轴位置的最佳增益。

### 预控制

在此框中，组态位置控制回路的百分比速度预控制。

### 增益 (Kv 因子)

在该域中，组态位置控制回路的增益 Kv。

### 1.5.4.5 参数视图

#### 参数视图简介

参数视图提供了工艺对象中所有相关参数的一般概述。

可获得参数设置的概述，并可在离线和在线模式下轻松地对其进行更改。

功能视图中的名称	在 DB 中的名称	...	项目起始值	数据类型	注释
替代输出值	SavePosition	...	0.0	% Real	请输入替代输出值.
输出值的上限	../OutputUpperL...	...	100.0	% Real	请输入输出值的上限.
输出值的下限	../OutputLowerL...	...	0.0	% Real	请输入输出值的下限.
上端停止位	../UpperPointOut	...	100.0	% Real	请输入上端停止位的输出值.
下端停止位	../LowerPointOut	...	0.0	% Real	请输入下端停止位的输出值.
Feedback_PER 下限	../LowerPointIn	...	0	Real	请输入 Feedback_PER 的下限值.
Feedback_PER 上限	../UpperPointIn	...	27648	Real	请输入 Feedback_PER 的上限值.
警告的上限	../InputUpperWa...	...	3.402822e...	% Real	请输入警告上限.
警告的下限	../InputLowerWa...	...	-3.402822e...	% Real	请输入警告下限.
启用手动输入		...	FALSE		启用手动输入 PID 参数.
比例增益	../Gain	...	1.0	Real	请输入比例增益.
积分作用时间	../Ti	...	20.0	s Real	请输入积分作用时间.
微分作用时间	../Td	...	0.0	Real	请输入微分作用时间.
微分延迟系数	../TdFiltRatio	...	0.2	Real	请输入微分延迟系数.
比例作用权重	../PWeighting	...	1.0	Real	请输入比例作用权重.
微分作用权重	../DWeighting	...	1.0	Real	请输入微分作用权重.
PID 算法采样时间	../Cycle	...	1.0	s Real	请输入 PID 算法的采样时间.
死区宽度	../InputDeadBand	...	0.0	Real	请输入死区宽度.

- ① “参数视图”(Parameter view) 选项卡
- ② 工具栏 (页 106)
- ③ 导航 (页 107)
- ④ 参数表 (页 108)

## 功能范围

提供以下可用于分析工艺对象参数和启用目标性监视与修改的功能。

显示功能：

- 在离线和在线模式下显示参数值
- 显示参数的状态信息
- 显示值偏差和直接连接选项
- 显示组态错误
- 显示由参数引起的值更改
- 显示某参数所有的存储值：PLC 起始值、项目起始值、监视值
- 显示参数存储值的参数比较

操作员控制功能：

- 为在参数之间和参数结构之间进行快速更改而导航。
- 用于更快搜索具体参数的文本过滤器。
- 用于按需自定义参数和参数组顺序的排序功能。
- 用于备份参数视图的结构设置的存储功能。
- 在线监视和修改参数值。
- 为捕获并响应瞬时情况而保存 CPU 参数值快照的功能。
- 用于将参数值快照应用为起始值的功能。
- 将已修改的起始值下载至 CPU。
- 用于比较两个参数值的比较功能。



## 有效性






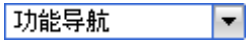
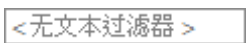


以下工艺对象提供参数视图:

- PID\_Compact
- PID\_3Step
- CONT\_C (仅适用于 S7-1500)
- CONT\_S (仅适用于 S7-1500)
- TCONT\_CP (仅适用于 S7-1500)
- TCONT\_S (仅适用于 S7-1500)
- TO\_Axis\_PTO (S7-1200 运动控制)
- TO\_Positioning\_Axis (S7-1200 运动控制)
- TO\_CommandTable\_PTO (S7-1200 运动控制)
- TO\_CommandTable (S7-1200 运动控制)

## 参数视图结构

## 工具栏

可在参数视图的工具栏中选择以下功能。

图标	功能	说明
	监视全部	在活动的参数视图中启动可见参数监视（在线模式）。
	创建监视值的快照并将该快照的设定值接受为起始值	将当前的监视值应用到“快照”(Snapshot) 列，并更新项目中的起始值。 仅可在 PID_Compact 和 PID_3Step 的在线模式下执行。
	初始化设定值	把在项目中更新过的起始值传送至 CPU。 仅可在 PID_Compact 和 PID_3Step 的在线模式下执行。
	创建监视值的快照	将当前的监视值应用到“快照”(Snapshot) 列。 仅可在在线模式下执行。
	请立即一次性修改全部选定参数	该命令尽快执行一次，而不参考用户程序中的任何特定点。 仅可在在线模式下执行。
	选择导航结构	在功能导航和数据导航之间进行切换。
	文本过滤器...	在输入字符串之后： 显示某一当前可见列中所有包括指定字符串在内的参数。
	选择比较值	在在线模式下，选择要与另一个参数值进行比较的参数值（项目起始值、PLC 起始值、快照） 仅可在在线模式下执行。
	保存窗口设置	为参数视图保存显示设置（例如选择的导航结构和激活的表中的列等）

## 导航

在“参数视图”(Parameter view) 选项卡中，有以下替代导航结构可供选择。

导航	说明
功能导航 <ul style="list-style-type: none"> <li>▼ 所有参数               <ul style="list-style-type: none"> <li>▶ 组态参数</li> <li>▶ 调试参数</li> <li>其它参数</li> </ul> </li> </ul>	在功能导航中，参数结构以组态对话框（“功能视图”(Functional view) 选项卡）、调试对话框和诊断对话框中的结构为基础。最后一个组“其它参数”(Other parameters) 包括工艺对象的所有其它参数。
数据导航 <ul style="list-style-type: none"> <li>▼ 所有参数               <ul style="list-style-type: none"> <li>Input</li> <li>Output</li> <li>InOut</li> <li>▶ Static</li> <li>其它参数</li> </ul> </li> </ul>	在数据导航中，参数结构以背景数据块中的结构为基础。最后一个组“其它参数”(Other parameters) 包括背景数据块中未包括的参数。

可以使用“选择导航结构”(Select navigation structure) 下拉列表来切换导航结构。

## 参数表

下表给出了参数表各列的含义。可以根据需要显示或隐藏列。

- “离线”列 = X: 该列在离线模式下可见。
- “在线”列 = X: 该列在在线模式下可见（在线连接到 CPU）。

列	说明	离线	在线
功能视图中的名称	功能视图中的参数名称。 如果参数未通过工艺对象组态，该显示字段留空。	X	X
DB 中的全称	背景数据块中参数的完整路径。 如果参数未包含在背景数据块中，该显示字段留空。	X	X
在 DB 中的名称	背景数据块中的参数名称。 如果参数是某结构或 UDT 的一部分，则应添加前缀“.”。 如果参数未包含在背景数据块中，该显示字段留空。	X	X
组态的状态	使用状态符号显示组态完整性。 请参见组态的状态（离线）(页 118)	X	
比较结果	“比较值”功能的结果。 如果已在线连接并选择了“监视所有”(Monitor all) 按钮  ，则会显示该列。		X
项目起始值	在项目中组态起始值。 如果输入的值具有语法错误或过程相关错误，则会出现错误指示。	X	X
默认值	为该参数预分配的值。 如果参数未包含在背景数据块中，该显示字段留空。	X	X
快照	CPU 中当前值的快照（监视值）。 如果值具有过程相关错误，则会出现错误指示。	X	X
PLC 起始值	CPU 中的起始值。 如果已在线连接并选择了“监视所有”(Monitor all) 按钮  ，则会显示该列。 如果值具有过程相关错误，则会出现错误指示。		X
监视值	CPU 中的当前值。 如果已在线连接并选择了“监视所有”(Monitor all) 按钮  ，则会显示该列。 如果值具有过程相关错误，则会出现错误指示。		X

列	说明	离线	在线
修改值	用于更改监视值的值。 如果已在线连接并选择了“监视所有”(Monitor all) 按钮  , 则会显示该列。 如果输入的值具有语法错误或过程相关错误, 则会出现错误指示。		X
选择用于传输 	使用“请立即一次性修改全部选定参数”(Modify all selected parameters immediately and once) 按钮选择要传输的修改值。 该列与“修改值”(Modify value) 列一起显示。		X
最小值	参数的最小过程相关值。 如果最小值取决于其它参数, 则将其定义为: <ul style="list-style-type: none"> <li>• 离线: 由项目起始值决定。</li> <li>• 在线: 由监视值决定。</li> </ul>	X	X
最大值	参数的最大过程相关值。 如果最大值取决于其它参数, 则将其定义为: <ul style="list-style-type: none"> <li>• 离线: 由项目起始值决定。</li> <li>• 在线: 由监视值决定。</li> </ul>	X	X
设定值	将参数指定为设定值。可在线初始化这些参数。	X	X
数据类型	变量的数据类型。 如果参数未包含在背景数据块中, 该显示字段留空。	X	X
保持性	将值指定为保持值。 保持性参数的值将保留, 即使在电源关闭后也是如此。	X	X
可从 HMI 访问	指示运行期间 HMI 是否可以访问此参数。	X	X
HMI 中可见	指示 HMI 选择列表中的参数是否默认可见。	X	X
注释	参数的简要描述。	X	X

## 1.5 定位轴工艺对象

### 打开参数视图

#### 要求

工艺对象已添加到项目树中，例如，指令的相关背景数据块已创建。

#### 步骤

1. 在项目树中打开“工艺对象”(Technology objects) 文件夹。
2. 在项目树中打开该工艺对象。
3. 双击“组态”(Configuration) 对象。
4. 选择右上角的“参数视图”(Parameter view) 选项卡。

#### 结果

参数视图将打开。所显示的每个参数都会以参数表中的一行表示。

可显示的参数属性（表的列）根据所使用的参数视图模式（离线或在线）而有所不同。

此外，可以有选择地显示和隐藏表的各个列。

#### 参见

参数视图默认设置 (页 111)

## 参数视图默认设置

### 默认设置

为了让您高效使用参数视图，可以自定义参数显示并保存设置。

可以执行并保存下列自定义项：

- 显示和隐藏列
- 改变列宽
- 改变列的顺序
- 切换导航
- 在导航中选择参数组
- 选择比较值

### 显示和隐藏列

要显示或隐藏参数表中的列，请按以下步骤操作：

1. 将光标放在参数表的标题上。
2. 在快捷菜单中选择“显示/隐藏”(Show/Hide) 命令。  
将显示可选的列。
3. 要显示列，请选中该列的复选框。
4. 要隐藏列，请清除该列的复选框。

或

1. 将光标放在参数表的标题上。
2. 如果要显示所有离线或在线模式的列，  
请在快捷菜单中选择“显示所有列”(Show all columns) 命令。

一些列只能在在线模式下显示： 请参见参数表 (页 108)。

## 改变列宽

要自定义列宽以便阅读所有行中的文本，请按以下步骤操作：

1. 将光标放在参数表标题中要进行自定义的列的右侧，直到光标的形状变为十字。
2. 然后双击此位置。

或

1. 打开参数表标题上的快捷菜单。
2. 点击
  - “优化列宽”(Optimize column width) 或
  - “优化所有列的宽度”(Optimize width of all columns)。

如果列宽设置得过窄，可将光标短暂悬停在相关字段上，即可出现各个字段的完整内容。

## 改变列的顺序

参数表的列可按照任意方式排列。

要更改列的顺序，请按以下步骤操作：

1. 单击列标题并使用拖放操作将其移动至所需位置。  
当松开鼠标时，对象会锚定在新位置上。

## 切换导航

要切换参数的显示格式，请执行以下步骤：

1. 在“选择导航结构”(Select navigation structure) 下拉列表中选择所需的导航。
  - 数据导航
  - 功能导航

另请参见导航 (页 107)。

## 在导航中选择参数组

在所选导航内，可在显示“所有参数”(All parameters) 或显示所选的下级参数组二者中选择其一。

1. 在导航中单击所需的参数组。  
参数表仅显示该参数组的参数。



## 选择比较值（在线）


要为“比较值”(Compare values) 功能设置比较值，请按下列步骤操作：

1. 在“选择比较值”(Selection of compare values) 下拉列表中选择所需的比较值。
  - 项目起始值/PLC 起始值
  - 项目起始值/快照
  - PLC 起始值/快照

默认情况下，会设置“项目起始值/PLC 起始值”(Start value project / Start value PLC) 选项。

## 保存参数视图的默认设置

要保存参数视图的以上自定义项，请按下列步骤操作：

1. 根据需要自定义参数视图。
2. 单击参数视图右上角的“保存窗口设置”(Save window settings) 按钮。

## 使用参数视图

### 概述

如下文所述，下表提供了在线和离线状态下的参数视图功能总览。

- “离线”列 = X：只能在离线模式下使用本功能。
- “在线”列 = X：只能在在线模式下使用本功能。

功能/操作	离线	在线
过滤参数表 (页 114)	X	X
将参数表排序 (页 115)	X	X
将参数数据传送给其它编辑器 (页 116)	X	X
指示错误 (页 117)	X	X
在项目中编辑起始值 (页 117)	X	X
组态的状态（离线） (页 118)	X	
参数视图中的在线监视值 (页 119)		X
创建监视值的快照 (页 120)		X
修改值 (页 121)		X
比较值 (页 122)		X
将来自在线程序的值应用为起始值 (页 124)		X
初始化在线程序中的设定值 (页 125)		X

### 过滤参数表

可通过下列方式过滤参数表中的参数：

- 使用文本过滤器
- 使用导航子组

两种过滤方法可同时使用。

## 使用文本过滤器

可过滤参数表中可见的文本。这表示只可以过滤显示的参数行和参数列中的文本。

1. 在“文本过滤器...”(Text filter...) 输入框中输入所需字符串以进行过滤。

参数表仅显示包含该字符串的参数。

文本过滤器会在下列情况下复位。

- 当选择了导航中的另一个参数组时。
- 当导航从数据导航更改为功能导航，或从功能导航更改为数据导航时。

## 使用导航子组

1. 在导航中单击所需的参数组，例如“静态”(Static)。

参数表仅会显示静态参数。可以为导航的一些组选择其它子组。

2. 如果要再次显示所有参数，请在导航中单击“所有参数”(All parameters)。

## 将参数表排序

参数值按行排列。参数表可以按照任意显示的列进行排序。

- 包含数字值的列会按照数字值的峰值进行排序。
- 文本列会按照字母顺序进行排序。

## 按列排序

1. 将光标放在所需列的标题单元格中。

该单元格的背景会变为蓝色。

2. 单击列标题。

## 结果

整个参数表会按照所选的列进行排序。列标题中会出现一个尖向上的三角形。

再次单击列标题会按以下情况更改排序方式：

- 符号“▲”：参数表按升序排序。
- 符号“▼”：参数表按降序排序。
- 无符号：再次移除排序。参数表采用默认显示。

排序时，忽略“在 DB 中的名称”列的“../”前缀。

## 将参数数据传送给其它编辑器

在选择了参数表的整个参数行之后，可以使用以下操作：

- 拖放
- <Ctrl+C>/<Ctrl+V>
- 通过快捷菜单中的复制/粘贴

将参数传送给 TIA Portal 的以下编辑器：

- 程序编辑器
- 监视表
- 用于跟踪功能的信号表

参数会以全称插入：请参见“DB 中的全称”(Full name in DB) 列中的信息。

## 指示错误

### 错误指示

会导致编译错误（例如超限）的参数分配错误将表示在参数视图中。

每当在参数视图中输入值，就会检查是否有过程相关错误和语法错误，并将结果表示出来。

错误的值会由以下方法表示：

- “组态的状态”（离线模式）或“比较结果”（在线模式，取决于所选的比较类型）列中的红色错误符号

和/或

- 背景为红色的表字段

如果单击错误的字段，会出现弹出错误消息，其中包含有关允许的值范围或所需语法（格式）的信息

### 编译错误

如果参数未显示在组态对话框中，可以从编译器的错误消息处直接打开包含出错参数的参数视图（功能导航）。

### 在项目中编辑起始值

可使用参数视图在离线模式和在线模式下编辑项目中的起始值。

- 可在参数表的“项目起始值”(Start value project) 列中更改值。
- 在参数表的“组态状态”(Status of configuration) 列中，会通过工艺对象组态对话框中相似的状态符号来表示组态进度。

### 约束条件

- 如果其它参数取决于那些起始值发生更改的参数，那么这些相关参数的起始值也会发生调整。
- 如果工艺对象的参数不可编辑，则也无法在参数视图中对其进行编辑。参数是否可以编辑还取决于其它参数值。

## 定义新起始值

要在参数视图中定义参数的起始值，请按下列步骤操作：

1. 打开工艺对象的参数视图。
2. 在“项目起始值”(Start value project) 列中输入所需的起始值。  
该值必须与参数的数据类型相匹配，不能超过参数的值范围。  
“最大值”(Maximum value) 和“最小值”(Minimum value) 列中给出了值范围的限值。

“组态的状态”(Status of configuration) 列用彩色符号表示组态进度。

另请参见 组态的状态（离线）(页 118)

在调整了起始值并将工艺对象下载到 CPU 的情况下，如果未将参数声明为保持（“保留”(Retain) 列），则参数会在启动时采用定义值。

## 错误指示

当输入了起始值时，会检查是否有过程相关错误和语法错误，并将结果表示出来。

错误的起始值会由以下方法表示：

- “组态的状态”（离线模式）或“比较结果”（在线模式，取决于所选的比较类型）列中的红色错误符号

和/或

- “项目起始值”(Start value project) 字段中的红色背景  
如果单击错误的字段，会出现弹出错误消息，其中包含有关允许的值范围或必要的语法（格式）的信息。

## 更正错误起始值

1. 使用来自弹出错误消息的信息更正错误的起始值。

将不再显示红色错误信息、红色字段背景和弹出错误消息。





除非起始值无误，否则项目将无法成功编译。

## 组态的状态（离线）

表示组态状态的图标位于：

- 参数表中的“组态的状态”(Status of configuration) 列中
- 功能导航和数据导航的导航结构中

### “组态的状态”列中的符号

符号	含义
	参数的起始值对应于默认值且有效。用户尚未定义起始值。
	参数的起始值中包含用户定义值。起始值与默认值不同。该起始值无误且有效。
	参数的起始值无效（语法或过程相关错误）。 输入框的背景为红色。单击弹出错误消息，会指出错误原因。
	仅限 S7-1200 运动控制： 参数的起始值有效但包含警告。 输入框的背景为黄色。

### 导航中的符号

导航中的符号指示组态过程的方式与工艺对象组态对话框中的方式相同。

### 参数视图中的在线监视值

可直接在参数视图中监视 CPU 中工艺对象参数当前采用的值（监视值）。

### 要求

- 需要有在线连接。
- 工艺对象已下载到 CPU 中。
- 程序执行处于激活状态（CPU 处于“RUN”模式）。
- 工艺对象的参数视图已打开。

## 步骤

1. 单击  启动监视。

一旦参数视图在线，将额外显示以下各列：

- 比较结果
- PLC 起始值
- 监视值
- 修改值
- 选择用于传输

“监视值”(Monitor value) 列显示了 CPU 上的当前参数值。

各附加列的含义：请参见参数表 (页 108)

2. 再次单击  停止监视。

## 显示


所有仅在线时可用的列以橙色背景显示：

- 浅橙色单元格  中的值可以更改。
- 背景为深橙色  的单元格中的值无法更改。

## 创建监视值的快照

可在 CPU 上备份工艺对象的当前值（监视值）并将其显示在参数视图中。


## 要求

- 需要有在线连接。
- 工艺对象已下载到 CPU 中。
- 程序执行处于激活状态（CPU 处于“RUN”模式）。
- 工艺对象的参数视图已打开。
- “监视所有”(Monitor all) 按钮  已选择。



## 步骤

要显示当前参数值，请按以下步骤操作：

1. 在参数视图中，单击“创建监视值的快照”(Create snapshot of monitor values) 图标 。

## 结果

会向参数表的“快照”(Snapshot) 列传送一次当前监视值。

监视值会在“监视值”(Monitor values) 列中继续更新，此时可分析以此方式“冻结”的值。

## 修改值

可通过参数视图修改 CPU 中工艺对象的值。

可以向参数分配一次值（修改值）并立即对其进行修改。

修改请求会尽快执行，而不参考用户程序中的任何特定点。




修改时存在的危险：

在发生故障或程序错误的情况下，如果在设备运行时更改参数值，则可能会导致严重财产损失和人员重伤。


在使用“修改”功能之前，请确保不会发生危险。

## 要求

- 需要有在线连接。
- 工艺对象已下载到 CPU 中。
- 程序执行处于激活状态（CPU 处于“RUN”模式）。
- 工艺对象的参数视图已打开。
- “监视所有”(Monitor all) 按钮  已选择。
- 参数需可修改（“修改值”(Modify value) 列中的相关字段背景为浅橙色）。

## 步骤

要立即修改参数，请按以下步骤操作：

1. 在参数表的“修改值”(Modify value) 列中输入所需修改值。
2. 检查“选择用于传输”(Select for transmission) 列中用于修改的复选框是否已选中。  
修改值和相关参数的相关复选框会同时自动调整。
3. 单击“请立即一次性修改全部选定参数”  
(Modify all selected parameters immediately and once) 图标 。

选定参数会由指定值立即一次性修改，并可以在“修改值”(Modify values) 列中进行监视。  
“选择用于传输”(Selection for transmission) 列中用于修改的复选框会在修改请求完成之后自动清除。

## 错误指示

当输入了起始值时，会立即检查是否有过程相关错误和语法错误，并将结果表示出来。

错误的起始值会由以下方法表示：

- “修改值”(Modify value) 字段中的红色背景  
并且
- 如果单击错误的字段，会出现弹出错误消息，其中包含有关允许的值范围或必要的语法（格式）的信息

## 错误修改值

- 具有过程相关错误的修改值可以进行传输。
- 具有语法错误的修改值**无法**进行传输。

## 比较值

可以使用比较功能来比较参数的以下存储值：


- 项目起始值
- PLC 起始值
- 快照

## 要求

- 需要有在线连接。
- 工艺对象已下载到 CPU 中。
- 程序执行处于激活状态（CPU 处于“RUN”模式）。
- 工艺对象的参数视图已打开。
- “监视所有”(Monitor all) 按钮  已选择。

## 步骤

要比较不同目标系统中的起始值，请按下列步骤操作：

1. 单击“选择比较值”(Selection of compare values) 图标 。

将打开一个包含比较选项的选择列表：





- 项目起始值 - PLC 起始值（默认设置）
- 项目起始值 - 快照
- PLC 起始值 - 快照

2. 选择所需的比较选项。

所选比较选项的执行方式如下：

- 在选择进行比较的两个列的标题单元格中会出现刻度符号。
- “比较结果”(Compare result) 列中使用的符号用来指示所选列的比较结果。

## “比较结果”列中的符号

符号	含义
	比较值相等且无误。
	比较值不相等但无误。
	两个比较值中至少有一个具有过程相关错误或语法错误。
	无法进行比较。两个比较值中至少有一个不可用（例如快照）。


## 导航中的符号

如果所显示的导航结构下方的参数中至少有一个应用了比较结果，则导航中的符号会以相同方式显示。

## 将来自在线程序的值应用为起始值


为了将来自 CPU 的优化值应用为项目的起始值，应创建监视值的快照。标记为设定值的快照值将应用为项目的起始值。

## 要求

- 工艺对象的类型为“PID\_Compact”或“PID\_3Step”。
- 需要有在线连接。
- 工艺对象已下载到 CPU 中。
- 程序执行处于激活状态（CPU 处于“RUN”模式）。
- 工艺对象的参数视图已打开。
- “监视所有”(Monitor all) 按钮  已选择。

## 步骤

要应用来自 CPU 的优化值，请按下列步骤操作：

1. 单击“创建监视值的快照并将该快照的设定值接受为起始值”(Create snapshot of monitor values and accept setpoints of this snapshot as start values) 图标 。

## 结果

应用到“快照”(Snapshot) 列及其设定值的当前监视值将作为新的起始值复制到“项目起始值”(Start value project) 列。

---

### 说明

#### 应用各个参数的值

还可以将未标记为设定值的各个参数值从“快照”(Snapshot) 列应用到“项目起始值”(Start value project) 列。为此，请使用快捷菜单中的“复制”(Copy) 和“粘贴”(Paste) 命令复制这些值并将其插入“项目起始值”(Start value project) 列中。

---

## 初始化在线程序中的设定值

在 CPU 中，仅通过一个步骤即可将所有在参数视图中标记为“设定值”(Setpoint) 的参数初始化为新值。为此，请将起始值从项目下载到 CPU 中。

CPU 将保持为“RUN”模式。

为避免在冷启动或暖启动期间丢失 CPU 上的数据，请务必将工艺对象也下载到 CPU 中。



**危险**


#### 更改参数值时存在的危险

在发生故障或程序错误的情况下，如果在设备运行时更改参数值，则可能会导致严重财产损失和人员重伤。

在重新初始化设定值之前，请确保不会发生危险。


## 1.5 定位轴工艺对象

### 要求

- 工艺对象的类型为“PID\_Compact”或“PID\_3Step”。
- 需要有在线连接。
- 工艺对象已下载到 CPU 中。
- 程序执行处于激活状态（CPU 处于“RUN”模式）。
- 工艺对象的参数视图已打开。
- “监视所有”(Monitor all) 按钮  已选择。
- 标记为“设定值”(Setpoint) 的参数具有无过程相关错误和语法错误的“项目起始值”(Start value project)。

### 步骤

要初始化所有设定值，请按以下步骤操作：

1. 在“项目起始值”(Start value project) 列中输入所需的值。  
确保起始值没有过程相关错误和语法错误。
2. 单击“初始化设定值”（Initialize setpoints）图标 。

### 结果

CPU 中的设定值初始化为项目的起始值。

## 1.6 工艺对象命令表

### 1.6.1 使用命令表工艺对象

使用工艺对象“命令表”(Command table)，可将多个单独的轴控制命令合并到一个运动序列中。

对于通过 PTO (Pulse Train Output) 连接驱动器的轴，此工艺对象可以用于工艺版本 V2 及更高版本。

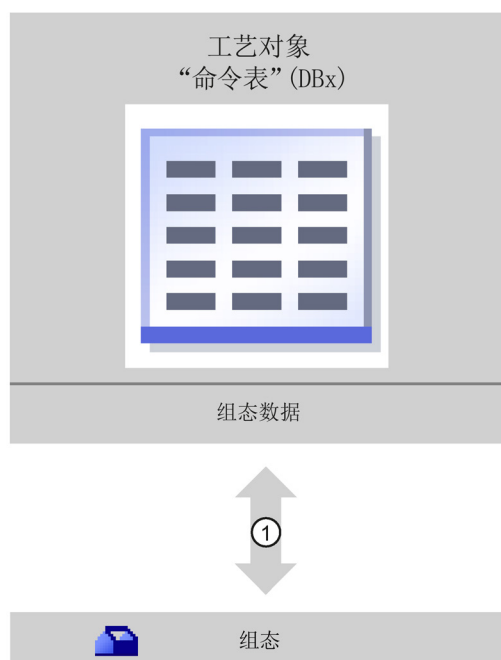
用于在组态对话框中以表格形式组态移动序列。

将项目装载到 CPU 之前，可以在曲线图上检查移动序列的运动轨迹。

然后，所创建的命令表将链接到某个轴并在用户程序中通过“MC\_CommandTable”运动控制指令进行使用。可以处理部分或全部命令表。

## 1.6.2 命令表工艺对象工具

在 TIA Portal 中，为“命令表”工艺对象提供了“组态”工具。  
下图显示了该工具与工艺对象的相互关系：



① 读取和写入工艺对象的组态

### 组态

使用“组态”工具可以组态“命令表”工艺对象的以下属性：

- 可以通过组态单独的作业来创建一个或多个移动序列。
- 可以使用已组态的轴或可组态的默认轴来组态图形显示，以检查移动序列。

移动序列数据保存在工艺对象的数据块中。



### 1.6.3 添加工艺对象命令表

#### 要求

- 已创建具有 CPU S7-1200 的项目。
- CPU 固件版本为 V2.1 或更高版本

#### 步骤

要在项目树中添加“命令表”工艺对象，请按以下步骤操作：

1. 在项目树中打开“CPU > 工艺对象”(CPU > Technology objects) 文件夹。
2. 双击“添加新对象”(Add new object) 命令。  
将打开“添加新对象”(Add new object) 对话框。
3. 选择“运动控制”(Motion Control) 工艺。
4. 打开“运动控制 > S7-1200 运动控制”  
(Motion Control > S7-1200 Motion Control) 文件夹。
5. 在“版本”(Version) 列中选择所需的工艺版本。
6. 选择“TO\_CommandTable”对象。
7. 在“名称”(Name) 输入框中输入命令表名称。
8. 要更改自动分配的数据块编号，请选择“手动”(Manual) 选项。
9. 要显示有关工艺对象的其它信息，请单击“其它信息”(Additional information)。
10. 单击“确定”(OK) 确认输入。

#### 结果

创建了新工艺对象，并保存在项目树中的“工艺对象”(Technology objects) 文件夹中。

## 1.6.4 组态命令表工艺对象

### 1.6.4.1 使用组态对话框

在组态窗口中，组态工艺对象的属性。

要打开工艺对象的组态窗口，请按以下步骤操作：

1. 在项目树中打开所需工艺对象组。
2. 双击“组态”(Configuration) 对象。

组态分为以下几类：

- **基本参数**





基本参数包括必须为功能命令表组态的所有参数。

- **扩展参数**

扩展参数包含默认轴的参数或显示所选轴的参数值。

### 组态窗口图标

组态的区域导航中的图标显示有关组态情况的详细信息：

	<p><b>组态包含默认值且已完成。</b></p> <p>组态仅包含默认值。使用这些默认值即可使用工艺对象，而无需进行更改。</p>
	<p><b>组态包含用户设置的值且已完成</b></p> <p>组态的所有输入域均包含有效值，且至少一个预设值发生更改。</p>
	<p><b>组态未完成或不正确</b></p> <p>至少一个输入域或下拉列表包含无效值。相应域或下拉列表以红色背景显示。 单击弹出错误消息可显示错误原因。</p>
	<p><b>组态包含彼此不兼容的参数值</b></p> <p>组态包含在大小或逻辑上互相矛盾的参数值。相应域或下拉列表以黄色背景显示。</p>

### 参见

运动控制使用指南 (页 35)

基本参数 (页 132)

扩展参数 (页 150)

### 1.6.4.2 比较值

如果存在到 CPU 的在线连接，“比较值”功能将显示在工艺对象的组态中。





“比较值”功能提供了以下选项：

- 将项目中组态的起始值与 CPU 中的起始值和实际值进行比较
- 直接编辑实际值和项目的起始值
- 立即检测并显示输入错误和建议的更正措施
- 备份项目中的实际值
- 将项目的起始值作为实际值传送至 CPU

### 图标和操作员控件

如果存在到 CPU 的在线连接，则参数中将显示实际值。

除参数的实际值外，还会显示下列符号：

图标	说明
	CPU 中的起始值与已组态的项目中起始值相匹配
	CPU 中的起始值与已组态的项目中的起始值不匹配
	无法将 CPU 中的起始值与已组态的项目中的起始值进行比较
	使用按钮显示 CPU 的起始值和各个参数的项目起始值。

可以直接更改项目中的实际值和起始值，然后下载到 CPU。

可直接修改的参数的实际值更改将直接传送到 CPU。

## 1.6 工艺对象命令表

### 1.6.4.3 基本参数

#### 组态 - 常规

在“常规”(General) 组态窗口中组态工艺对象的名称。

#### 名称

在该域中定义命令表的名称或“命令表”工艺对象的名称。  
该工艺对象以该名称列出在项目树中。

#### 参见

组态 - 命令表 (页 132)

快捷菜单命令 - 命令表 (页 138)

使用趋势图 (页 140)

快捷菜单命令 - 曲线图 (页 145)

从“完成命令”过渡到“混合运动” (页 147)

在用户程序中更改命令表组态 (页 149)

#### 组态 - 命令表

在“命令表”(Command Table)

组态窗口中创建所需移动序列，并根据趋势图中的图形视图检查结果。

---

#### 说明

可能在显示的时间行为和趋势位置与轴的实际运动之间存在较小偏差。  
未显示与达到软限位开关对应的移动。

---

## 启用警告

使用该复选框，可在命令表中激活警告显示。

## 使用轴参数

从下拉列表中，选择使用哪些轴参数来选择图形视图以及检查移动序列。

如果尚未将轴添加到“工艺对象”(Technology objects)

文件夹或想要使用未在任何可用轴中组态的值，应选择“默认轴”(Default axis)。

在“高级参数”(Advanced parameters) 下组态默认轴的属性。

“Axis”参数中选择的轴参数，将用于处理用户程序中的命令表。

## 列： 步骤

显示命令步数。

## 1.6 工艺对象命令表

### 列： 命令类型

在该列中选择处理命令表所使用的命令类型。最多可输入 32 个命令。  
将会按顺序处理命令。可以在下列条目和命令类型之间选择：

- **Empty**

该条目用作所有要添加的命令的占位符。处理命令表时会忽略空条目。

- **Halt**

停止轴

（只有在执行“Velocity set point”命令之后该命令才生效）

- **Positioning Relative**

相对定位轴

- **Positioning Absolute**

绝对定位轴

- **Velocity set point**

以设定速度移动轴

- **Wait**

等到给定期结束。Wait 不会停止当前行进。

- **Separator**

将 Separator 行添加到所选行的上方。

Separator 行用作趋势图图形显示的范围限制。

如想处理部分命令表，请使用 Separator 行。

**列： 位置/行进路径**

在此列中可为所选命令输入位置或行进路径：

- **命令“Positioning Relative”**

该命令将按给定的行进路径移动轴。

- **命令“Positioning Absolute”**

该命令将按给定的位置移动轴。

- **Separator**

给定值指定了图形显示的起始位置。

限值（与所选的用户单位无关）：

- $-1.0e12 \leq \text{位置/距离} \leq -1.0e-12$
- $1.0e-12 \leq \text{位置/距离} \leq 1.0e12$
- 位置/行进路径 = 0.0

## 列：速度

在该列中，可以为所选命令输入速度：

- **命令“Positioning Relative”**

该命令将以给定的速度移动轴。

如果选定行进路径不够大，则不会达到给定速度。

- **命令“Positioning Absolute”**

该命令将以给定的速度移动轴。

如果目标位置距起始位置太近，则不会达到给定速度。

- **命令“ Velocity set point”**

该命令将以给定的速度移动轴。

如果选择的运行时间过短，则执行命令期间不会达到给定速度。

限值（与所选的用户单位无关）：

- 对于命令：“Positioning Relative”和“Positioning Absolute”

- $1.0e-12 \leq \text{速度} \leq 1.0e12$

- 对于命令：“Velocity set point”

- $-1.0e12 \leq \text{速度} \leq -1.0e-12$

- $1.0e-12 \leq \text{速度} \leq 1.0e12$

- 速度 = 0.0

## 列：持续时间

在该列中可为所选命令输入持续时间：

- **命令“ Velocity set point”**

该命令将在指定的持续时间内移动轴。持续时间包括加速阶段和匀速行进阶段。

持续时间结束后将会处理下一个命令。

- **命令“Wait”**

等待，直到给定持续时间结束。

限值（与所选的用户单位无关）：

- $0.001s \leq \text{持续时间} \leq 64800s$



## 列： 下一步

从下拉列表中选择过渡到下一步的模式：

- **完成命令**

命令将被完成。 会立即处理下一命令。

- **混合运动**

当前命令的运动将会与下面命令的运动混合。 利用“Positioning Relative”和“Positioning Absolute”命令类型可以进入“混合运动”转换模式。

运动将会与下列命令类型的运动混合：

- Positioning Relative
- Positioning Absolute
- Velocity set point

利用其他命令类型不会发生混合。

有关附加或重叠命令时轴的具体行为，请参见： 从“完成命令”过渡到“混合运动” (页 147)

## 列： 步代码

在此列中可输入数值/位模式， 处理命令时会在“MC\_CommandTable”运动控制指令的“StepCode”输出参数中输出该数值/位模式。

限值：

- $0 \leq \text{代码编号} \leq 65535$

## 参见

组态 - 常规 (页 132)

快捷菜单命令 - 命令表 (页 138)

使用趋势图 (页 140)

快捷菜单命令 - 曲线图 (页 145)

从“完成命令”过渡到“混合运动” (页 147)

在用户程序中更改命令表组态 (页 149)

## 1.6 工艺对象命令表

### 快捷菜单命令 - 命令表

命令表中有以下可用快捷菜单命令：

#### 插入空行 (Insert empty line)

将空行添加到所选行的上方。

只有命令表的结尾有足够多的空行时才会执行快捷菜单命令。

#### 添加空行 (Add empty line)

将空行添加到所选行的下方。

只有命令表的结尾有足够多的空行时才会执行快捷菜单命令。

#### 插入分隔符行 (Insert separator line)

将分隔符行添加到所选行的上方。

无法连续添加两个分隔符行。

#### 添加分隔符行 (Add separator line)

将分隔符行添加到所选行的下方。

无法连续添加两个分隔符行，也无法在命令表的结尾添加分隔符行。

#### 剪切 (Cut)

移除所选行或所选单元格的内容并将其保存到剪切板。

所选行将会删除并且命令表的后续行将会上移。

#### 复制 (Copy)

复制所选行或所选单元格的内容并将其保存到剪切板。

### 粘贴 (Paste)

- 所选行：  
将剪切板中的行粘贴到所选行上方的表中。
- 所选单元格：  
将剪切板的内容粘贴到所选行。  
只有命令表的结尾有足够多的空行时才会执行快捷菜单命令。

### 替换 (Replace)

用剪切板中的行替换所选行。

### 删除 (Delete)

删除所选行。命令表中下方的行将上移。

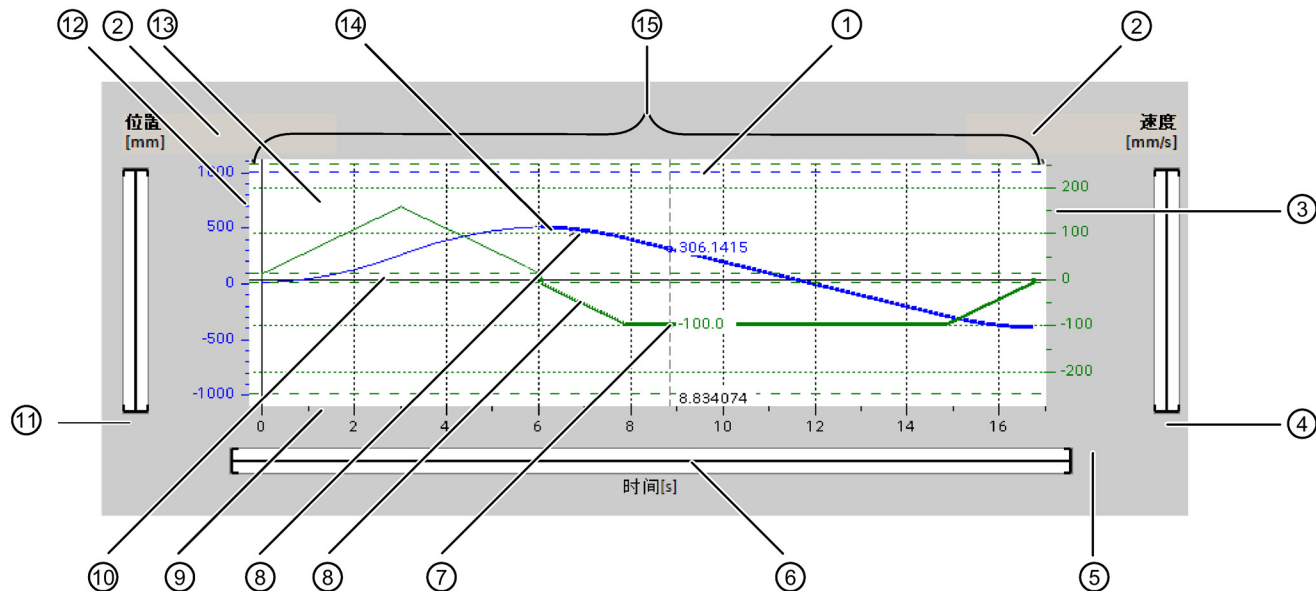
### 参见

- 组态 - 常规 (页 132)
- 组态 - 命令表 (页 132)
- 使用趋势图 (页 140)
- 快捷菜单命令 - 曲线图 (页 145)
- 从“完成命令”过渡到“混合运动” (页 147)
- 在用户程序中更改命令表组态 (页 149)

### 使用趋势图

趋势视图中提供下列工具和和信息：

#### 趋势图和组件



①	比例尺
②	选择网格
③	速度轴刻度范围
④	滚动条，速度轴
⑤	滚动条时间轴
⑥	比例尺位置标记
⑦	速度曲线
⑧	所选命令的曲线部分
⑨	时间轴刻度范围
⑩	启动/停止速度
⑪	滚动条，位置轴
⑫	位置轴刻度范围
⑬	软限位开关位置
⑭	位置曲线
⑮	趋势视图

## 选择分隔符部分

如果命令表包含多个由分隔符分隔的部分，可以在趋势图中选择这些部分，方法是在相应部分中选择一个命令。

## 选择命令

可以在趋势图和命令表中选择命令：

- 单击速度上的某个点或趋势图中的位置曲线。相应的命令会在命令表中突出显示。
- 在命令表中选择一个命令。

曲线的相应部分将会突出显示。

## 选择趋势图的可见范围

按照下面的步骤可以调整要显示的趋势图部分：

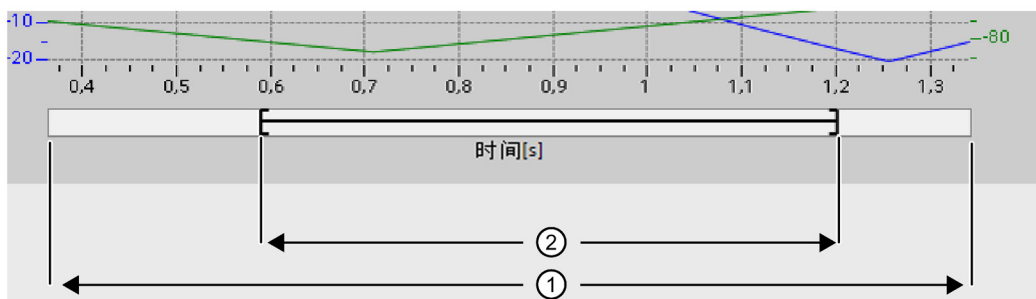
在快捷菜单中选择缩放：

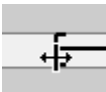

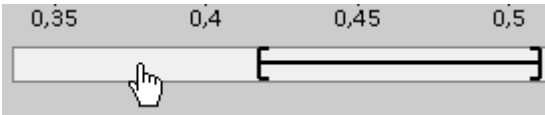
- 缩放到曲线：  
缩放轴，以便位置和速度曲线可见。
- 缩放到曲线和限值：  
缩放轴，以便位置和速度曲线、激活软限位开关的位置和最小最大速度限值可见。

所选图在快捷菜单中将会用勾号标记。

1.6 工艺对象命令表

选择要在范围内显示的部分：

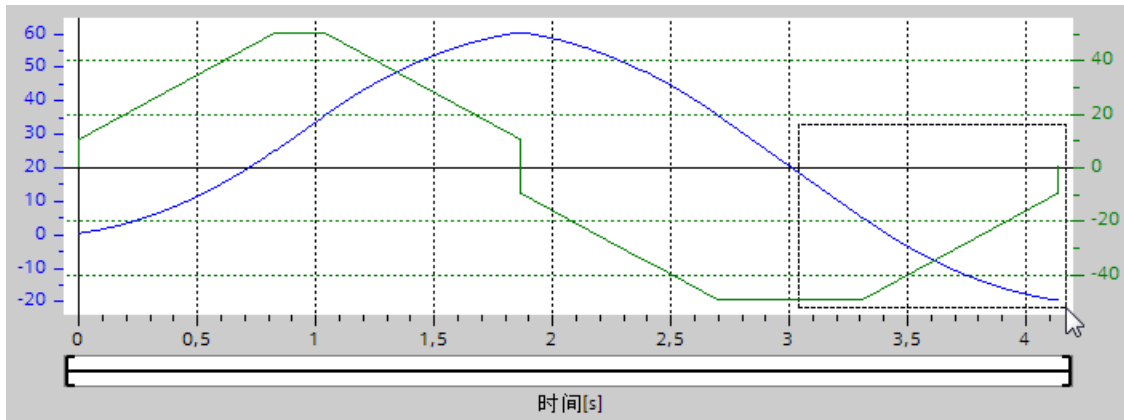


①	曲线值和/或限值所在的范围。（请参见“在快捷菜单中选择”）
②	<p>要在趋势窗口中显示的所选范围。</p> <p>利用空白光标可以设置右边和左边的空白范围。</p>  <p>通过拖动光标在范围 ① 内设置位置。</p>  <p>还可以通过在范围 ① 内单击来定义位置。</p> 

利用鼠标选择要显示的部分：

通过单击和拖动鼠标拖选出趋势图的某一部分。

所选曲线的该部分将会在释放鼠标后立即放大。



撤消对该部分的最后一次更改：

选择快捷命令“撤消缩放”(Undo zoom)，撤消对该部分的最后一次更改。

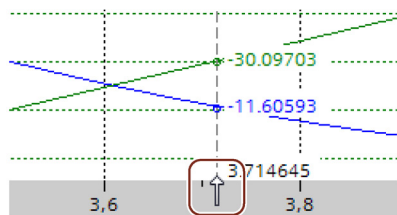
## 同步网格

单击轴刻度，选择是否将网格与位置轴或速度轴同步。

### 从比例尺读取曲线值

使用快捷菜单命令“显示比例尺”(Show ruler) 激活比例尺。

使用比例尺光标可以将比例尺移动至曲线上的任何点。



### 参见

组态 - 常规 (页 132)

组态 - 命令表 (页 132)

快捷菜单命令 - 命令表 (页 138)

快捷菜单命令 - 曲线图 (页 145)

从“完成命令”过渡到“混合运动” (页 147)

在用户程序中更改命令表组态 (页 149)



## 快捷菜单命令 - 曲线图

曲线窗口中有以下可用快捷菜单命令：

### 缩放 100% (Zoom 100%)

选择一个可以显示 100% 曲线值和/或限值的缩放系数。

### 撤消缩放 (Undo zoom)

撤消最后一次缩放更改。

### 趋势缩放

缩放轴，以便位置和速度趋势可见。

### 趋势和限值标度

缩放轴，以便位置和速度趋势、激活软件限位开关的位置和最小最大速度限值可见。

### 显示速度限值

显示速度限值行。

## 1.6 工艺对象命令表

### 显示软件限位开关

显示软件限位开关的行。

### 显示标尺

使标尺渐显/渐隐

在要查看趋势的各个值时可使用标尺。

### 参见

组态 - 常规 (页 132)

组态 - 命令表 (页 132)

快捷菜单命令 - 命令表 (页 138)

使用趋势图 (页 140)

从“完成命令”过渡到“混合运动” (页 147)

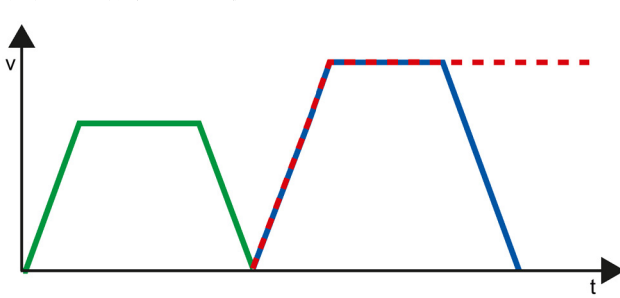
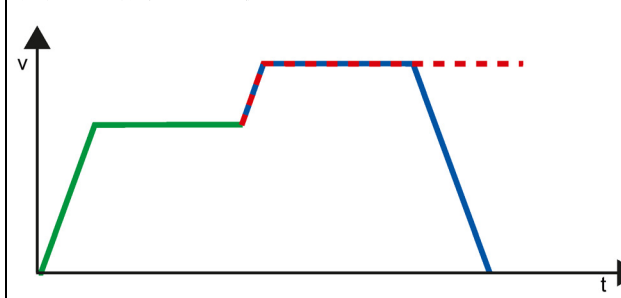
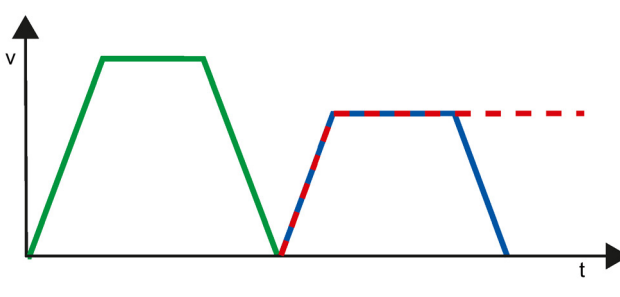
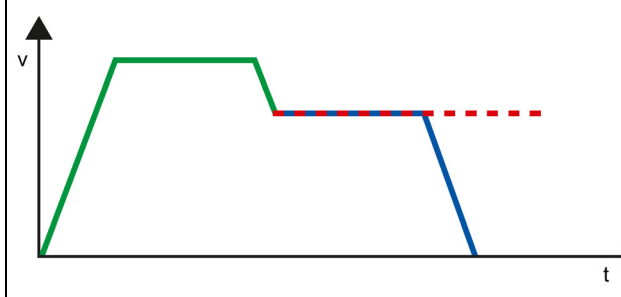
在用户程序中更改命令表组态 (页 149)

## 从“完成命令”过渡到“混合运动”

下图在“下一步”列中显示了不同转换模式移动之间的转换：

### 具有先前定位作业的运动过渡

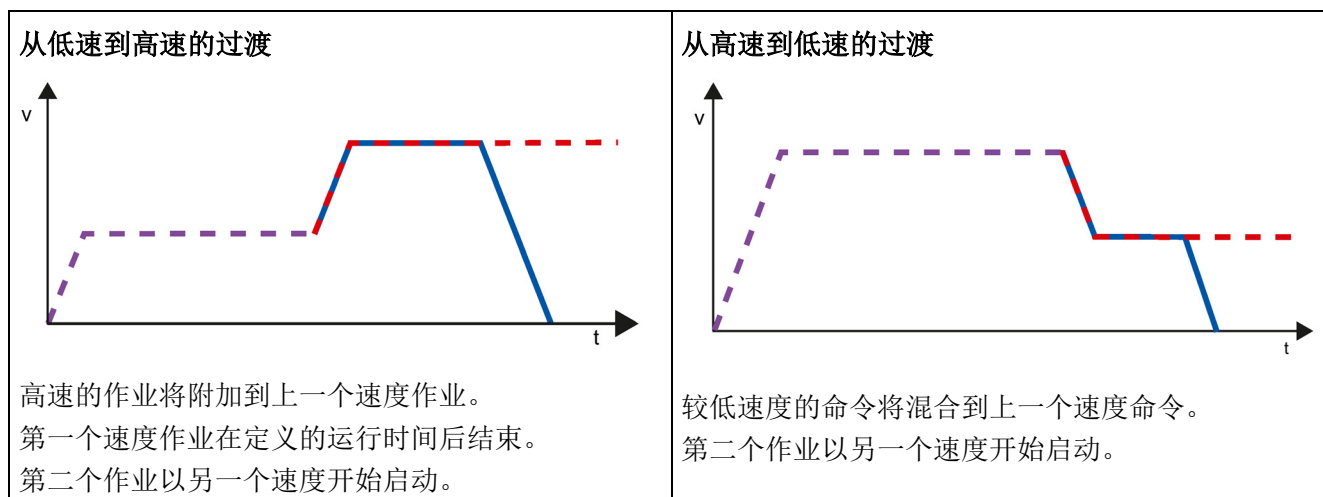
下图显示了两个运动任务的命令顺序。第一个命令用于定位（绿色）。第二个命令用于速度（红色）或定位（蓝色）：

完整作业	混合运动
<p><b>从低速到高速的过渡</b></p>  <p>高速作业会附加到上一定位作业。 定位作业在其目标位置以速度“0”终止。 第二个作业从静止开始启动。</p>	<p><b>从低速到高速的过渡</b></p>  <p>高速作业会与上一定位作业重叠。 第一个定位作业在其目标位置以非静止速度终止。 第二个作业以另一个速度开始启动。</p>
<p><b>从高速到低速的过渡</b></p>  <p>低速作业会附加到上一定位作业。 定位作业在其目标位置以速度“0”终止。 第二个作业从静止开始启动。</p>	<p><b>从高速到低速的过渡</b></p>  <p>低速作业会与上一定位作业重叠。 第一个定位作业在其目标位置以非静止速度终止。 第一个作业以另一个速度开始启动。</p>

—	1. 作业“Positioning Relative”或“Positioning Absolute”
- -	2. 作业“Velocity set point”
—	2. 作业“Positioning Relative”或“Positioning Absolute”

### 具有先前速度作业的运动过渡

下图显示了两个运动任务的命令顺序。第一个命令用于速度（紫色）。  
第二个命令用于速度（红色）或定位（蓝色）：



--	1. 作业“Velocity set point”
- -	2. 作业“Velocity set point”
—	2. 作业“Positioning Relative”或“Positioning Absolute”

### 参见

- 组态 - 常规 (页 132)
- 组态 - 命令表 (页 132)
- 快捷菜单命令 - 命令表 (页 138)
- 使用趋势图 (页 140)
- 快捷菜单命令 - 曲线图 (页 145)
- 在用户程序中更改命令表组态 (页 149)

## 在用户程序中更改命令表组态

可以在 CPU 中运行用户程序期间更改下列组态参数：

## 命令和对应值

还可以在用户程序运行期间更改命令表的参数。

要执行该操作，可使用以下工艺对象变量：

- `<表名称>.Command[1..32].Type`  
用于更改命令类型
- `<表名称>.Command[1..32].Position`  
用于更改位置/行进距离
- `<表名称>.Command[1..32].Velocity`  
用于更改速度
- `<表名称>.Command[1..32].Duration`  
用于更改持续时间
- `<表名称>.Command[1..32].NextStep`  
用于更改参数“下一步”
- `<表名称>.Command[1..32].StepCode`  
用于更改步代码

有关组态参数的更改何时生效的信息，请参见附录中的工艺对象变量 (页 288) 说明。

## 参见

变量兼容性列表 (页 40)

组态 - 常规 (页 132)

组态 - 命令表 (页 132)

快捷菜单命令 - 命令表 (页 138)

使用趋势图 (页 140)

快捷菜单命令 - 曲线图 (页 145)

从“完成命令”过渡到“混合运动” (页 147)

#### 1.6.4.4 扩展参数

##### 组态 - 扩展参数

在“扩展参数”(Extended parameters)  
组态窗口中组态“命令表”工艺对象图表视图的基本属性。

---

##### 说明

如果已在“使用轴参数”(Use axis parameters of) 下选择默认轴，则可以编辑测量单位。  
如果已选择组态轴，将会显示该轴的测量单位。

---

##### 使用轴参数

从下拉列表中，选择使用哪些轴参数来选择图形视图以及检查移动序列。  
如果尚未将轴添加到“工艺对象”(Technology objects)  
文件夹或想要使用未在任何可用轴中组态的值，应选择“默认轴”(Default axis)。  
“轴”(Axis) 参数中选择的轴的轴参数将用于处理用户程序中的命令表。

##### 位置计量单位

在此域中输入默认轴的计量单位。如果已在“使用轴参数”(Use axis parameters of)  
中选择预组态的轴，将会显示这些参数中组态的计量单位。

##### 复制轴参数

选择复制方向及要复制轴参数的轴。  
可将默认轴的轴参数复制到所选轴，或将所选轴的轴参数作为默认轴的轴参数。  
使用“应用组态”(Apply configuration) 按钮，可根据组态复制轴参数。

##### 组态 - 动态

在“动态”(Dynamics) 组态窗口中可以为默认轴组态加速度和减速度以及冲击限制。

---

##### 说明

如果已在“使用轴参数”(Use axis parameters of) 下选择默认轴，则可以编辑下列字段。  
如果已选择组态的轴，将会显示该轴的值。

---

### 加速度/减速度

在“加速度”(Acceleration) 字段中设置默认轴的所需加速度。

在“减速度”(Deceleration) 字段中可以设置所需减速度。

将会用所选加速度/减速度计算命令表中组态的运动作业。

限值:

- $1.0e-12 \leq \text{加速度} \leq 1.0e12$
- $1.0e-12 \leq \text{减速度} \leq 1.0e12$

### 激活冲击限制

使用该复选框，可启用加加速度限值。

### 加加速度

在“加加速度”(Jerk) 字段中，可设置加速和减速的所需加加速度。

将会用所选加加速度计算命令表中组态的运动作业。

限值:

- $1.0e-12 \leq \text{冲击} \leq 1.0e12$

### 组态 - 限值

在“限值”(Limits) 组态窗口中组态最大速度、启动/停止速度和默认轴的软件限位开关。

---

#### 说明

如果已在“使用轴参数”(Use axis parameters of) 下选择默认轴，则可以编辑下列字段。  
如果已选择组态的轴，将会显示该轴的值。

---

### 最大速度/启动/停止速度

在这些框中定义默认轴的最大允许速度和启动/停止速度。  
启动/停止速度是默认轴的最小允许速度。

限值:

- $1.0e-12 \leq \text{启动/停止速度} \leq 1.0e12$

启动/停止速度 = 0.0

- $1.0e-12 \leq \text{最大速度} \leq 1.0e12$

最大速度 = 0.0

最大速度值必须大于等于启动/停止速度值。

### 启用软件限位开关

使用此复选框可激活下限和上限软限位开关的功能。  
未在趋势图中显示与达到软件限位开关对应的移动。

### 下限/上限软件限位开关

在这些框中可输入下限和上限软限位开关的位置值。

限值:

- $-1.0e12 \leq \text{下限软件限位开关} \leq -1.0e-12$

$1.0e-12 \leq \text{下限软件限位开关} \leq 1.0e12$

下限软件限位开关 = 0.0

- $-1.0e12 \leq \text{上限软件限位开关} \leq -1.0e-12$

$1.0e-12 \leq \text{上限软件限位开关} \leq 1.0e12$

上限软件限位开关 = 0.0

上限软限位开关的值必须大于等于下限软限位开关的值。



## 1.7 下载到 CPU

运动控制工艺对象的数据保存在数据块中。

因此，加载新的或已修改的工艺对象时，下载“块”的条件适用。



### 不使用硬件配置加载时轴的可能故障

对轴组态进行以下修改时，会修改硬件配置：

- 修改脉冲发生器 (PTO)
- 修改硬限位开关地址
- 修改归位开关的地址
- 修改 PROFIdrive 报文的地址
- 修改模拟量输出的地址
- 修改使能输出或准备就绪输入的地址

如果使用上下文菜单命令“软件”(Software) 或“软件 (所有块)”(Software (all blocks)) 加载轴的已修改组态而未下载硬件配置，则可能会导致轴发生故障。

确保在下列条件下将当前硬件配置下载到 CPU 中。

## 在 CPU S7-1200 RUN 模式下下载（固件版本 V2.2 及更高版本）

对于固件版本 V2.2 及更高版本的 CPU S7-1200，在 CPU RUN 模式下加载时，会检查是否可在不停止 CPU 的情况下进行加载。

在 RUN 模式下加载数据块时，以下条件适用：

	下载到装载存储器	下载到工作存储器
数据块修改的值	√	-
数据块修改的结构	√（自固件 V4 起）	√（自固件 V4 起） • 下载需重新初始化时 • 下载无需重新初始化的系统备用变量
	-（固件 V2.2...3）	-（固件 V2.2...3）
新数据块	√	√
删除的数据块	√	√

删除数据块以及下载需要重新初始化的数据块时还请注意以下信息：

- 下载定位轴工艺对象时必须将轴禁用。
- 下载命令表工艺对象时，不得激活该命令表的 MC\_CommandTable 命令（参数“Busy”= FALSE）。
- 下载 MC\_Power 背景数据块时，不得激活 MC\_Power 指令（参数“Busy”= FALSE）。

从工艺版本 V3.0 起，运动控制工艺对象（数据块）也可以在 CPU RUN 模式下下载。

V3.0 之前的工艺对象无法通过 CPU RUN 模式下载。

选择下述操作之一，将运动控制工艺对象（版本 V3.0 或更高版本）的已修改版本下载到工作存储器：

- **工艺对象定位轴和命令表**  
将 CPU 操作模式从 STOP 切换到 RUN。
- **工艺对象定位轴**  
将禁用轴并使用运动控制指令“MC\_Reset”执行“Restart”。
- **工艺对象命令表**  
可确保命令表未被使用。  
使用扩展指令“READ\_DBL”将命令表的数据块下载到工作存储器。

## 参见

运动控制使用指南 (页 35)

MC\_Reset: 确认故障 V4 及更高版本 (页 374)

## 1.8 调试

### 1.8.1 轴控制面板

轴控制面板用于在手动模式下移动轴、优化轴设置和测试系统。

只有与 CPU 建立在线连接后，才能使用轴控制面板。

---

#### 说明

##### 轴控制面板的响应时间

轴控制面板操作期间的响应时间取决于 CPU 的通信负载。关闭 TIA Portal 的所有其它在线窗口可将响应时间降至最低。

---

#### “手动控制”按钮

单击“手动控制”(Manual control) 可在手动控制模式下移动轴。

首先在用户程序中使用运动控制指令“MC\_Power”禁用轴。

在“手动控制”(Manual control) 模式下，轴控制面板对轴功能具有优先控制权。

在结束手动控制前，用户程序不能影响轴功能。



##### 自动模式下的其它轴

只能为一根轴激活手动控制。如果自动模式下存在其它轴，则可能引发危险。

如果发生这种情况，请禁用所有其它轴。

## 1.8 调试

### “自动模式”(Automatic mode) 按钮

单击“自动模式”(Automatic mode)，可结束“手动控制”模式。  
轴控制面板交回优先控制权，然后轴可再次由用户程序控制。  
必须在用户程序中重新启用轴。需要时，可使其归位。

请在切换到自动控制前完成所有激活的行进运动；否则，将以急停减速度对轴进行制动。

### “启用”(Enable) 按钮

单击“启用”(Enable) 可在“手动控制”模式下启用轴。  
轴启用时，可以使用轴控制面板功能。

如果因为没有满足某些条件而无法启用轴，请注意“错误消息”(Error message) 框中的错误消息。有关清除错误的信息，请参见附录中的“ErrorID 和 ErrorInfo 列表”。  
纠正错误后，再次启用轴。

### “禁用”(Disable) 按钮

如果要在“手动控制”模式下临时禁用轴，则单击“禁用”(Disable)。

## “命令”(Command) 区域

仅在轴启用后才可以执行“命令”(Command) 区域中的操作。

可以选择以下命令输入之一：

- **点动**

该命令相当于用户程序中的运动控制命令“MC\_MoveJog”。

- **定位**

该命令相当于用户程序中的运动控制命令“MC\_MoveAbsolute”和“MC\_MoveRelative”。必须使轴归位以便进行绝对定位。

- **归位**

该命令相当于用户程序中的运动控制命令“MC\_Home”。

- “设置参考点”(Set reference point) 按钮相当于 Mode = 0（绝对式直接归位）

- “主动归位”(Active homing) 按钮相当于 Mode = 3（主动归位）

对于主动归位，必须在轴组态中组态归位开关。

逼近速度、归位速度和参考位置偏移的值取自尚未更改的轴组态。

根据选择，将显示相关的设定值输入框和命令启动按钮。

使用“激活加加速度”(Activate jerk limit) 按钮可以激活和禁用加加速度限值。

默认情况下，加加速度为组态值的 10%。可根据需要更改该值。

## “轴状态”(Axis status) 区域

如果激活“手动控制”模式，则“轴状态”(Axis status)

区域中将显示当前的轴状态和驱动器状态。“过程值”(Process values) 中显示轴的当前轴位置和速度。

单击“确认”(Acknowledge) 可确认所有清除的错误。

“信息消息”(Info message) 框会显示有关轴状态的高级信息。

## “当前值”(Current values) 区域

此区域将显示轴的当前位置和速度。

## 1.8 调试

### 错误消息

“错误消息”(Error message) 框会显示当前错误。

在“手动控制”模式下，清除错误后，可通过按下“确认”(Acknowledge) 按钮来删除错误条目。

---

#### 说明

##### 速度、加速度/减速度和冲击的初始值

出于安全考虑，轴控制面板激活时，仅使用等于组态值 10% 的值来初始化“速度”(Velocity)、“加速度/减速度”(Acceleration/deceleration) 和“冲击”(Jerk) 参数。“冲击”(Jerk) 参数仅用于 V2.0 及更高版本的“轴”工艺对象。

在组态视图中选择“扩展参数 > 动态 > 常规”

(Extended parameters > Dynamics > General) 后显示的值可用于初始化。

控制面板中的“速度”(Velocity) 参数来自组态中的“最大速度”(Maximum velocity)，“加速度/减速度”(Acceleration/deceleration) 参数则来自组态中的“加速度”(Acceleration)。

在轴控制面板中，可以更改“速度”(Velocity)、“加速度/减速度”(Acceleration/deceleration) 和“冲击”(Jerk) 参数；这不会影响组态中的值。

---

### 参见

运动控制使用指南 (页 35)

## 1.8.2 调节

通过 PROFIdrive/模拟量输出连接驱动器的轴的移动是受位置控制的。

通过“自整定”(Tuning) 功能，可确定用于轴控制回路 (页 102)的最佳增益 (Kv 因子)。为此，可通过“跟踪”(Trace) 功能记录所组态的定位移动过程中轴速度配置文件。然后，可对记录进行评估并对增益进行相应调整。

定位轴工艺对象的“自整定”(Tuning) 功能位于项目树的“工艺对象 > 调试”(Technology object > Commissioning) 中。

“自整定”(Tuning) 对话框中包含有以下区域：

- 主控制
- 轴
- 优化增益设置
- 跟踪

---

### 说明

#### 不传输参数

返回主控制后放弃所组态的参数值。 并根据需要将值传输到组态中。

---


## 主控制

在此区域中，可对工艺对象进行主控制或返回给用户程序：

- **“激活”(Activate) 按钮**

使用“激活”(Activate) 按钮可建立与 CPU 的在线连接，并对所选工艺对象进行主控制。在进行主控制时，请注意以下事项：

- 要进行主控制，必须在用户程序中禁用工艺对象。
- 在返回主控制之前，用户程序不会影响工艺对象的功能。拒绝从用户程序对工艺对象执行运动控制作业，并报告错误 (“ErrorID”= 16#8203)。

 <b>小心</b>
<b>自动模式下的其它轴</b> 仅对选定的工艺对象进行主控制。 如果自动模式下存在其它轴，则可能引发危险。 如果发生这种情况，请禁用所有其它轴。

- **“禁用”(Deactivate) 按钮**

使用“禁用”(Deactivate) 按钮可将主控制返回给用户程序。

## 轴

在此区域中，启用或禁用工艺对象使用轴控制面板/自整定运行：

- **“启用”(Enable) 按钮**

使用“启用”(Enable) 按钮可启用所选的工艺对象。

- **“禁用”(Disable) 按钮**

使用“禁用”(Disable) 按钮可禁用所选的工艺对象。



## 优化增益设置

可以在此区域中设置增益优化：

- **“向前”(Forward) 按钮**

使用“向前”(Forward) 按钮，可朝正方向启动优化测试步。

- **“向后”(Backward) 按钮**

使用“向后”(Backward) 按钮，可朝负方向启动优化测试步。

- **“自定义动态”(Customize dynamics) 复选框**

选择此选项可调整优化的加速度和最大加速度。

- **加速度**

在此域中，组态一个测试步的加速度。

- **最大速度**

在此域中，组态一个测试步的最大速度。

- **测量持续时间**

在此域中，组态一个测试步的持续时间。

- **增益**

在此域中，组态位置控制器的实际增益 (Kv)。

## 跟踪

“跟踪”(Trace) 功能将显示在“自整定”(Tuning) 对话框的下面区域中。

对于每个测试步，都会自动启动所需参数的跟踪记录，并在完成该测试步后显示该记录。返回主控制后，将删除跟踪记录。

有关跟踪功能的完整说明，请参见 TIA Portal 帮助中有关使用跟踪和逻辑分析器功能的部分。

## 1.9 编程

### 1.9.1 运动控制语句概述

在用户程序中，可以使用运动控制指令控制轴。  
这些指令会启动执行所需功能的运动控制命令。

可以从运动控制指令的输出参数中获取运动控制命令的状态及命令执行期间发生的任何错误。适用的运动控制指令有：

- MC\_Power: 启用、禁用轴 V4 及更高版本 (页 368)
- MC\_Reset: 确认故障 V4 及更高版本 (页 374)
- MC\_Home: 使轴归位，设置参考点 V4 及更高版本 (页 377)
- MC\_Halt: 停止轴 V4 及更高版本 (页 382)
- MC\_MoveAbsolute: 轴的绝对定位 V4 及更高版本 (页 386)
- MC\_MoveRelative: 轴的相对定位 V4 及更高版本 (页 391)
- MC\_MoveVelocity: 以预设的旋转速度移动轴 V4 及更高版本 (页 396)
- MC\_MoveJog: 在点动模式下移动轴 V4 及更高版本 (页 402)
- MC\_CommandTable: 按照运动顺序运行轴命令 V4 及更高版本 (页 406)
- MC\_ChangeDynamic: 更改轴的动态设置 V4 及更高版本 (页 409)
- MC\_ReadParam: 连续读取定位轴的运动数据 V4 及更高版本 (页 413)
- MC\_WriteParam: 写入定位轴的变量 V4 及更高版本 (页 416)

#### 参见

创建用户程序 (页 163)

编程注意事项 (页 166)

断电和重新启动后运动控制命令的行为 (页 168)

监视激活的命令 (页 169)

运动控制语句的错误显示 (页 181)

## 1.9.2 创建用户程序

以下部分介绍了如何创建具有用于轴控制的基本组态的用户程序。  
使用待插入的运动控制指令，可以控制所有可用的轴功能。

### 要求

- 已正确创建并组态工艺对象。

在创建和测试用户程序之前，建议使用轴命令表测试轴功能和系统的相应部分。

### 步骤

要创建符合下述原则的用户程序，请按以下步骤操作：

1. 在项目树中，双击代码块（代码块必须在循环程序中调用）。  
代码块在程序编辑器中打开并显示所有可用指令。
2. 打开“工艺”(Technology) 类别以及“运动控制”(Motion Control) 和“S7-1200 运动控制”(S7-1200 Motion Control) 文件夹。
3. 将“MC\_Power”指令拖放到代码块中相应的程序段。  
将打开用于定义背景数据块的对话框。
4. 在下一个对话框中，选择其中一个选项：

#### 单背景

单击“单背景”(Single instance)

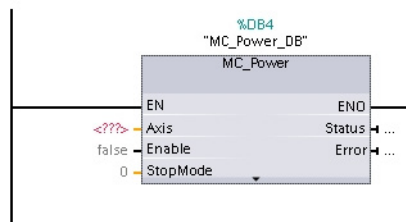
并选择是要自动还是手动定义背景数据块的名称和编号。

#### 多重背景

单击“多重背景”(Multi-instance) 并选择是自动还是手动定义多重背景名称。

5. 单击“确定”(OK)。

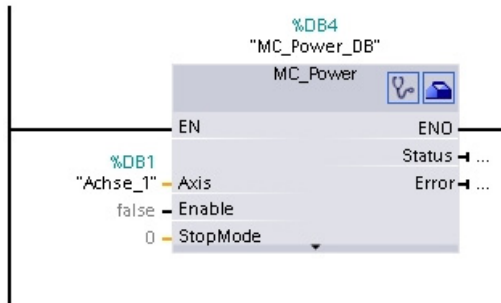
运动控制指令“MC\_Power”将插入到该程序段中。



必须初始化标有“<???”的参数；给所有其它参数分配默认值。

黑体显示的参数是使用运动控制指令时所必需的参数。

6. 在项目树中选择工艺对象并将其拖放到 <??> 上。



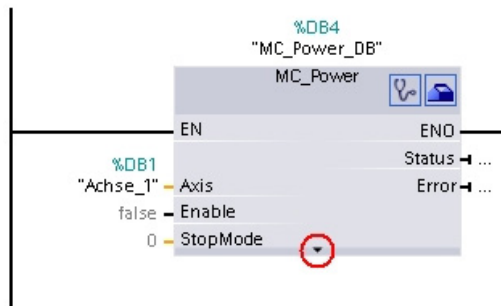
选择工艺对象数据块后，以下按钮将可用：



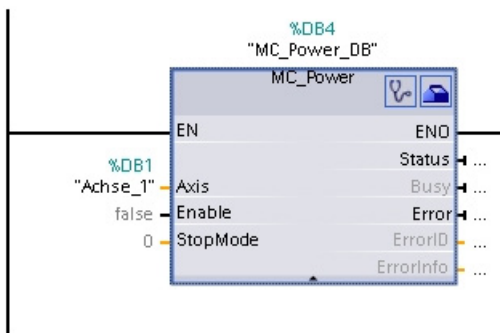
如果想要打开工艺对象的诊断对话框，则单击听诊器图标。



如果想要打开工艺对象的组态视图，则单击工具箱图标。



单击向下箭头图标，可查看运动控制指令的其它参数。



也可使用此时可见的灰显参数。

7. 按照步骤 3 至 6 添加您所选择的运动控制指令。

## 结果

已在用户程序中创建用于轴控制的基本组态。

在用户程序的其它部分中初始化运动控制指令的输入参数，启动需要对“轴”工艺对象执行的作业。

评估运动控制指令的输出参数和数据块中的变量，跟踪启动的作业和轴状态。

有关运动控制指令的参数的详细信息，请参见详细说明。

## 参见

运动控制语句概述 (页 162)

编程注意事项 (页 166)

断电和重新启动后运动控制命令的行为 (页 168)

监视激活的命令 (页 169)

运动控制语句的错误显示 (页 181)

### 1.9.3 编程注意事项

创建用户程序时，请注意以下信息：

- **循环调用所使用的运动控制指令**

命令的当前执行状态通过运动控制指令的输出参数提供。

每次调用运动控制指令时都会更新状态。

因此，确保循环调用所使用的运动控制指令。

- **传送运动控制指令的参数值**

如果调用了该块，则在输入参数“Execute”的上升沿时将传送输入参数的未决参数值。

运动控制命令将以这些参数值启动。

在下次启动运动控制命令前，不会传送运动控制指令随后发生更改的参数值。

运动控制指令“MC\_Power”的 StopMode 输入参数和运动控制指令

“MC\_MoveJog”的 Velocity 输入参数是个例外。

“Enable”= TRUE、“JogForward”和“JogBackward”时，也应用输入参数中的更改。

- **编程时考虑状态信息**

在逐步执行运动控制命令时，确保等到激活的命令执行完成后再启动新命令。

使用运动控制指令的状态消息和工艺对象的“StatusBits”变量，

可以检查激活的命令是否完成。

在下例中，请按照所示顺序执行。如果未遵守此顺序，将显示轴或命令错误。

- **使用运动控制指令“MC\_Power”启用轴**

只有启用轴后，轴才可以执行运动作业。

通过对变量 <轴名称>.StatusBits.Enable = TRUE

与运动控制指令“MC\_Power”的输出参数 Status = TRUE

进行与运算，将检查轴是否已启用。

- **使用运动控制指令“MC\_Reset”确认错误**

启动运动控制命令前，必须使用“MC\_Reset”确认需要确认的错误。

消除错误原因并使用运动控制指令“MC\_Reset”确认错误。

启动新命令前，需检查错误是否已成功确认。为此，需要对变量

<轴名称>.StatusBits.Error = FALSE 和运动控制指令“MC\_Reset”的输出参数

Done = TRUE 进行与运算。

- **使用运动控制指令“MC\_Home”使轴回原点**

使轴回原点后，才可启动 MC\_MoveAbsolute 命令。

通过对变量 <轴名称>.StatusBits.HomingDone = TRUE

与运动控制指令“MC\_Home”的输出参数 Done = TRUE

进行与运算，将检查轴是否已回原点。

- **执行运动控制命令覆盖**

用于移动轴的运动控制作业还可作为覆盖作业执行。

如果在一个运动控制命令处于激活状态时对轴另启动一个新的运动控制命令，则在完全执行现有命令前，新命令将覆盖激活的命令。

被覆盖的命令将通过在运动控制指令中使 `CommandAborted = TRUE` 发出信号。

可以使用 `MC_MoveAbsolute` 命令覆盖激活的 `MC_MoveRelative` 命令。

- **避免多次使用相同的背景**

运动控制命令的所有相关信息都存储在其背景中。

如果要跟踪当前命令的状态，请勿使用该背景启动新命令。

如果要单独跟踪命令，则需使用不同的背景。

如果将同一个背景用于多个运动控制命令，

则各命令的状态和错误信息将会相互覆盖。

- **以不同优先级（运行级别）调用运动控制指令**

在未互锁的情况下，可能无法以不同优先级调用具有相同背景的运动控制指令。

要了解如何调用已锁定的运动控制指令，请参见“跟踪较高优先级（运行级别）的命令 (页 194)”。

## 参见

运动控制语句概述 (页 162)

创建用户程序 (页 163)

断电和重新启动后运动控制命令的行为 (页 168)

监视激活的命令 (页 169)

运动控制语句的错误显示 (页 181)

在更高优先级等级（执行级别）中跟踪作业 (页 194)

#### 1.9.4 断电和重新启动后运动控制命令的行为

断电或 CPU STOP 将中止所有激活的运动控制作业。  
包括脉冲和方向输出在内的所有 CPU 输出都将被重置。

随后上电或 CPU 重新启动 (CPU RUN) 后，将重新初始化工艺对象和运动控制作业。

工艺对象的所有实际数据以及之前激活的运动控制作业的所有状态和错误信息，  
都将重置为各自的初始值。

只有再次使用运动控制指令“MC\_Power”启用轴后，才可重新使用该轴。  
如果需要回原点，则必须再次使用运动控制指令“MC\_Home”使轴回原点。  
使用绝对值编码器时，断电后将保持归位状态。

#### 参见

运动控制语句概述 (页 162)

创建用户程序 (页 163)

编程注意事项 (页 166)

监视激活的命令 (页 169)

运动控制语句的错误显示 (页 181)



## 1.9.5 监视激活的命令

### 1.9.5.1 监视激活的命令

有三个典型组可用于跟踪激活的运动控制命令：

- 具有输出参数“Done”的运动控制指令
- 运动控制指令“MC\_MoveVelocity”
- 运动控制指令“MC\_MoveJog”

### 1.9.5.2 具有输出参数“Done”的运动控制指令

具有输出参数“Done”的运动控制指令通过输入参数“Execute”启动，并且具有明确的结论（例如，对于运动控制指令“MC\_Home”：回原点已成功）。

该命令已完成，轴处于停止状态。

以下运动控制指令的命令具有一个指定功能：

- MC\_Reset
- MC\_Home
- MC\_Halt
- MC\_MoveAbsolute
- MC\_MoveRelative
- MC\_CommandTable（自工艺对象 V2 起）
- MC\_ChangeDynamic（自工艺对象 V2 起）
- MC\_WriteParam（自工艺对象 V4 起）
- MC\_ReadParam（自工艺对象 V4 起）

如果命令已成功完成，则输出参数“Done”的值为 TRUE。

输出参数“Busy”、“CommandAborted”和“Error”分别指示命令仍在处理、已中止或有未决的错误。

运动控制指令“MC\_Reset”不会被中止，因此没有“CommandAborted”输出参数。

运动控制指令“MC\_ChangeDynamic”立即完成，因此没有“Busy”或“CommandAborted”输出参数。

在运动控制命令的执行过程中，输出参数“Busy”的值将为 TRUE。

如果命令已完成、中止或因错误停止，则输出参数“Busy”的值将变为 FALSE。

无论输入参数“Execute”的信号状态是什么，都会发生这种变化。

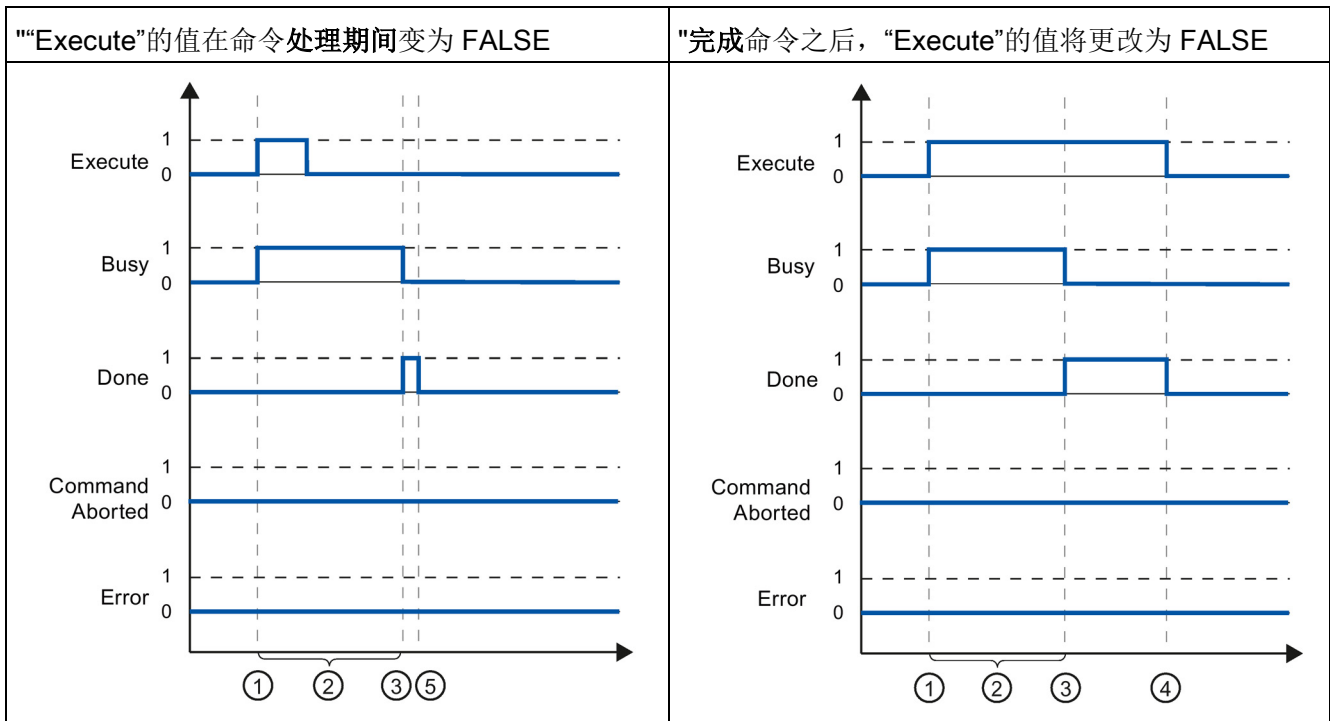
输出参数“Done”、“CommandAborted”和“Error”的值至少在一个周期内都为 TRUE。

当输入参数“Execute”设置为 TRUE 时，将锁存这些状态消息。

下图针对各种示例情况显示了状态位的操作：

### 命令执行完成

如果运动控制命令已在对其下结论前完全执行，则将通过输出参数“Done”的值为 TRUE 的形式对此进行指示。输入参数“Execute”的信号状态影响输出参数“Done”中的显示持续时间：

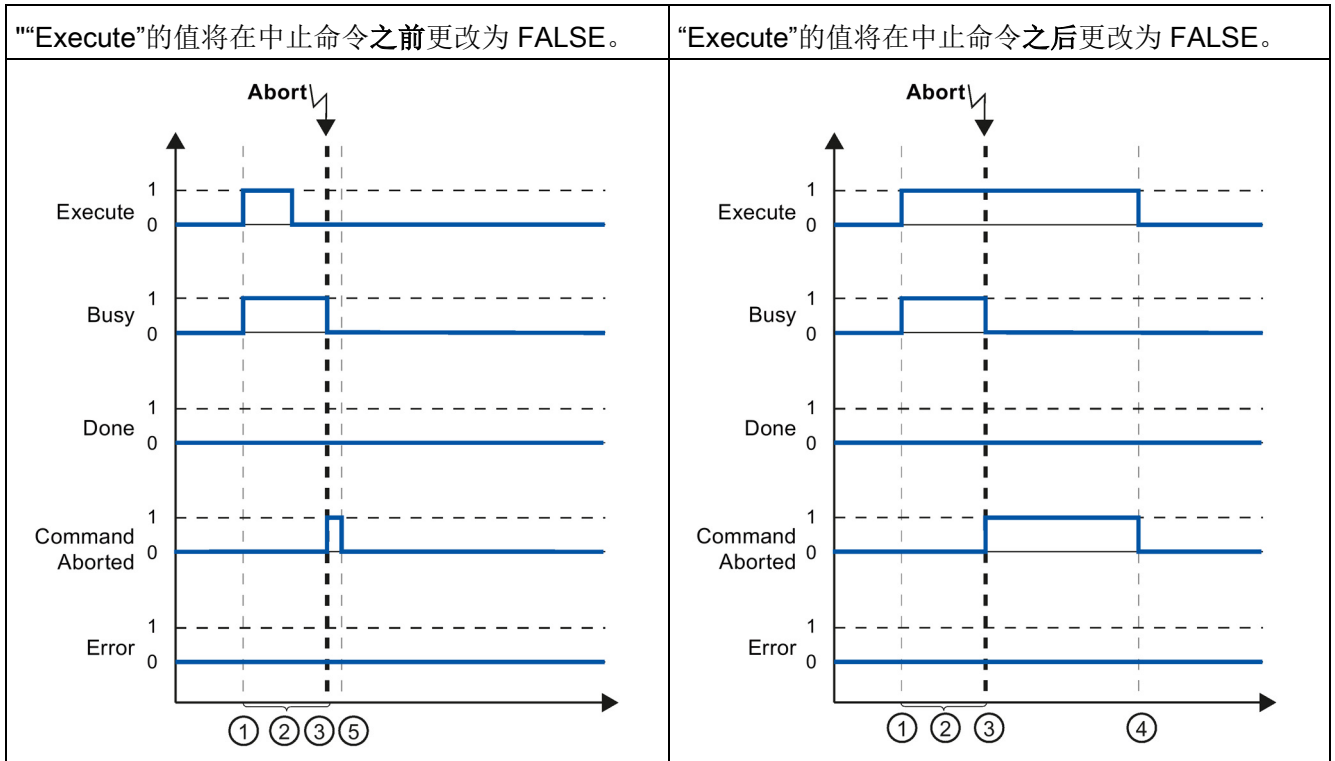


- |   |   |
|---|---|
| ① | 输入参数“Execute”出现上升沿时启动命令。<br>根据编程情况，“Execute”在命令的执行过程中仍然可能复位为值 FALSE，或者保持为值 TRUE，直到命令执行完成为止。 |
| ② | 激活命令时，输出参数“Busy”的值将为 TRUE。  |
| ③ | 命令执行结束后（例如，对于运动控制指令“MC_Home”：回原点已成功），输出参数“Busy”变为 FALSE，“Done”变为 TRUE。                      |
| ④ | 如果“Execute”的值在命令完成之前保持为 TRUE，则“Done”的值也将保持为 TRUE 并且其值随“Execute”一起变为 FALSE。                  |
| ⑤ | 如果“Execute”在命令执行完成之前设置为 FALSE，则“Done”的值仅在一个执行周期内为 TRUE。                                     |

## 中止命令

如果运动控制命令在执行期间中止，则将通过输出参数“CommandAborted”的值为 TRUE 的方式对此进行指示。

输入参数“Execute”的信号状态影响输出参数“CommandAborted”中的显示持续时间：

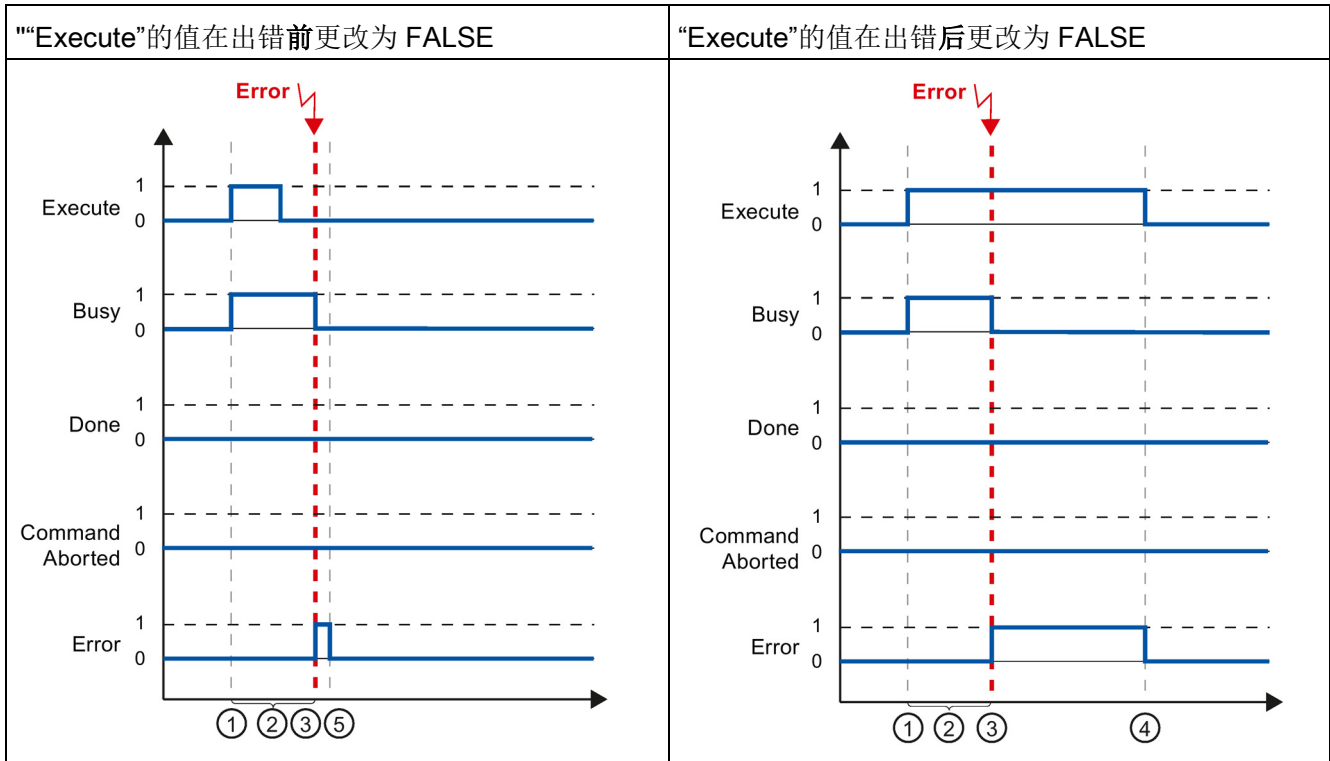


①	输入参数“Execute”出现上升沿时启动命令。 根据编程情况，“Execute”在命令的执行过程中仍然可能复位为值 FALSE，或者保持为值 TRUE，直到命令执行完成为止。
②	激活命令时，输出参数“Busy”的值将为 TRUE。
③	在执行命令的过程中，该命令可由其它运动控制命令中止。 如果中止命令，则输出参数“Busy”的值将变为 FALSE，“CommandAborted”的值将变为 TRUE。
④	如果“Execute”的值在中止命令之前保持为 TRUE， 则“CommandAborted”的值也将保持为 TRUE 并且其值随“Execute”一起变为 FALSE。
⑤	如果“Execute”在中止命令执行之前设置为 FALSE，则“CommandAborted”的值仅在一个执行周期内为 TRUE。

### 命令执行过程中出错

如果在执行运动控制命令过程中出错，则将通过输出参数“Error”的值为 TRUE 的方式对此进行指示。

输入参数“Execute”的信号状态影响输出参数“Error”中的显示持续时间：



①	输入参数“Execute”出现上升沿时启动命令。 根据编程情况，“Execute”在命令的执行过程中仍然可能复位为值 FALSE，或者保持为值 TRUE，直到命令执行完成为止。
②	激活命令时，输出参数“Busy”的值将为 TRUE。
③	执行命令过程中出错。出错时，输出参数“Busy”的值将变为 FALSE 且“Error”的值将变为 TRUE。
④	如果“Execute”的值在出错之前保持为 TRUE，则“Error”的值也将保持为 TRUE 并且其值仅随“Execute”一起变为 FALSE。
⑤	如果“Execute”已在出错之前设置为 FALSE，则“Error”的值仅在一个执行周期内为 TRUE。

### 1.9.5.3 运动控制指令 MC\_MoveVelocity

在参数“Execute”的上升沿启动“MC\_MoveVelocity”命令。

当达到指定的速度且轴以恒定速度移动时，就达到了命令目标。

在达到并保持指定的速度时，将在参数“InVelocity”中用值为 TRUE 进行指示。

例如，可通过“MC\_Halt”命令停止轴运动。

输出参数“Busy”、“CommandAborted”和“Error”分别指示命令仍在处理、已中止或有未决的错误。

在运动控制命令的执行过程中，输出参数“Busy”的值将为 TRUE。

如果命令因另一命令或错误而停止，则输出参数“Busy”的值将变为 FALSE。

无论输入参数“Execute”的信号状态是什么，都会发生这种变化。

输出参数“CommandAborted”和“Error”的值至少在一个周期内保持为 TRUE。

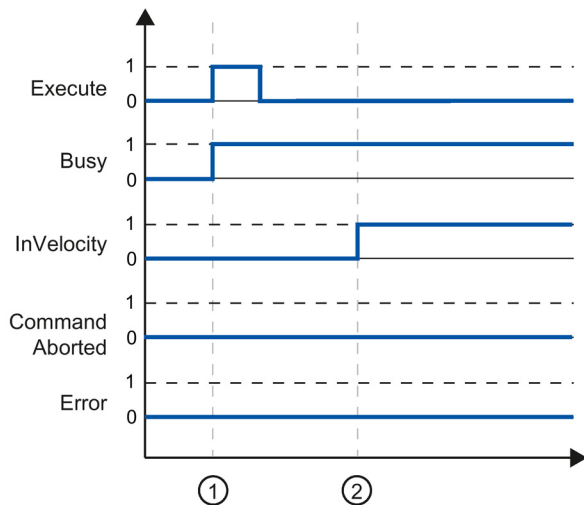
当输入参数“Execute”设置为 TRUE 时，将锁存这些状态消息。

下图针对各种示例情况显示了状态位的操作：

### 达到设置的速度

如果运动控制命令在达到设置的速度前执行完毕，则将通过输出参数“**InVelocity**”的值为 **TRUE** 的方式对此进行指示。

参数“**Execute**”不会影响“**InVelocity**”参数中的指示持续时间。

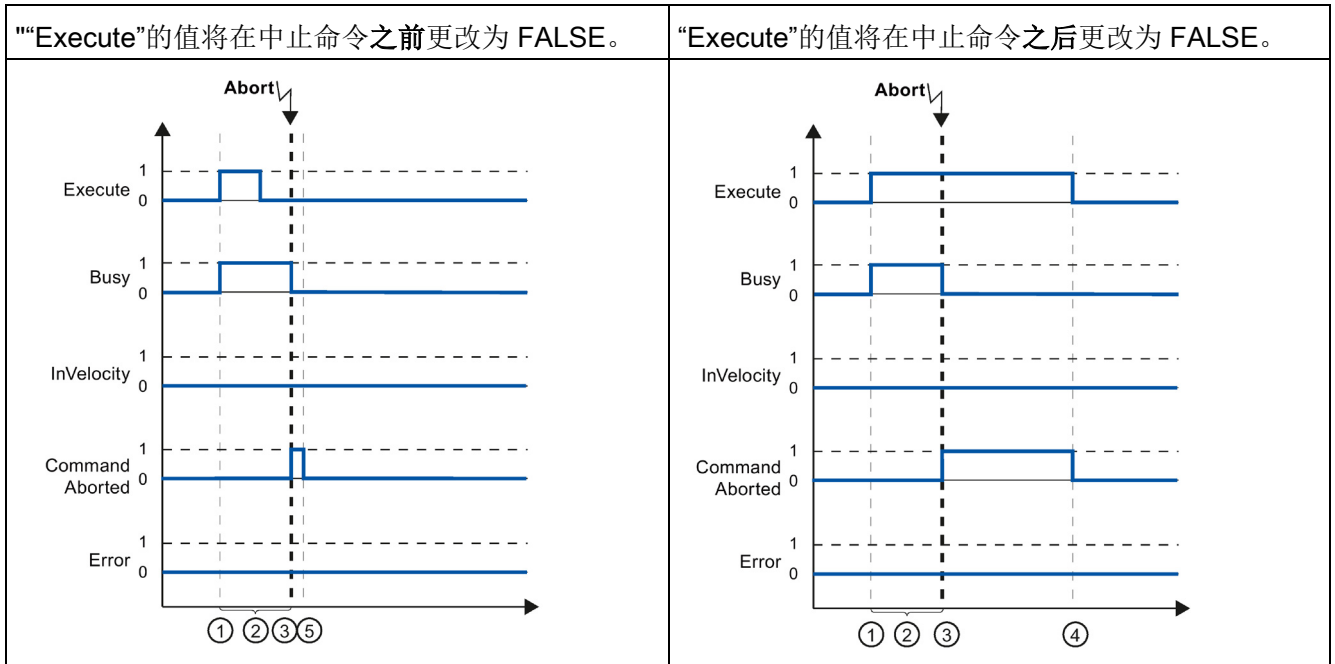


①	<p>将在参数“<b>Execute</b>”的上升沿启动作业。 根据编程，在达到设置的速度之前或之后，可以将“<b>Execute</b>”复位为值 <b>FALSE</b>。在该作业处于激活状态期间，参数“<b>Busy</b>”的值为 <b>TRUE</b>。</p>
②	<p>在达到指定的速度时，参数“<b>InVelocity</b>”的值将变为 <b>TRUE</b>。 在“<b>MC_MoveVelocity</b>”命令由另一运动控制命令覆盖或因错误停止前，“<b>Busy</b>”和“<b>InVelocity</b>”参数的值将一直为 <b>TRUE</b>。</p>

## 命令在达到设置的速度前中止

如果运动控制命令在达到设置的速度前中止，则通过输出参数“CommandAborted”的 TURE 值对此进行指示。

输入参数“Execute”的信号状态影响输出参数“CommandAborted”中的显示持续时间。



①	输入参数“Execute”出现上升沿时启动命令。 根据编程情况，“Execute”在命令执行期间仍然可能被重置为值 FALSE，或者保持为值 TURE，直到命令中止为止。
②	激活命令时，输出参数“Busy”的值将为 TRUE。
③	在执行命令的过程中，该命令可由其它运动控制命令中止。 如果中止命令，则输出参数“Busy”的值将变为 FALSE，“CommandAborted”的值将变为 TRUE。
④	如果“Execute”的值在命令中止之前保持为 TRUE，则“CommandAborted”的值也将保持为 TRUE 并且其状态会随“Execute”一起变为 FALSE。
⑤	如果“Execute”在命令中止之前重置为 FALSE，则“CommandAborted”的值仅在一个执行周期内为 TRUE。

## 说明

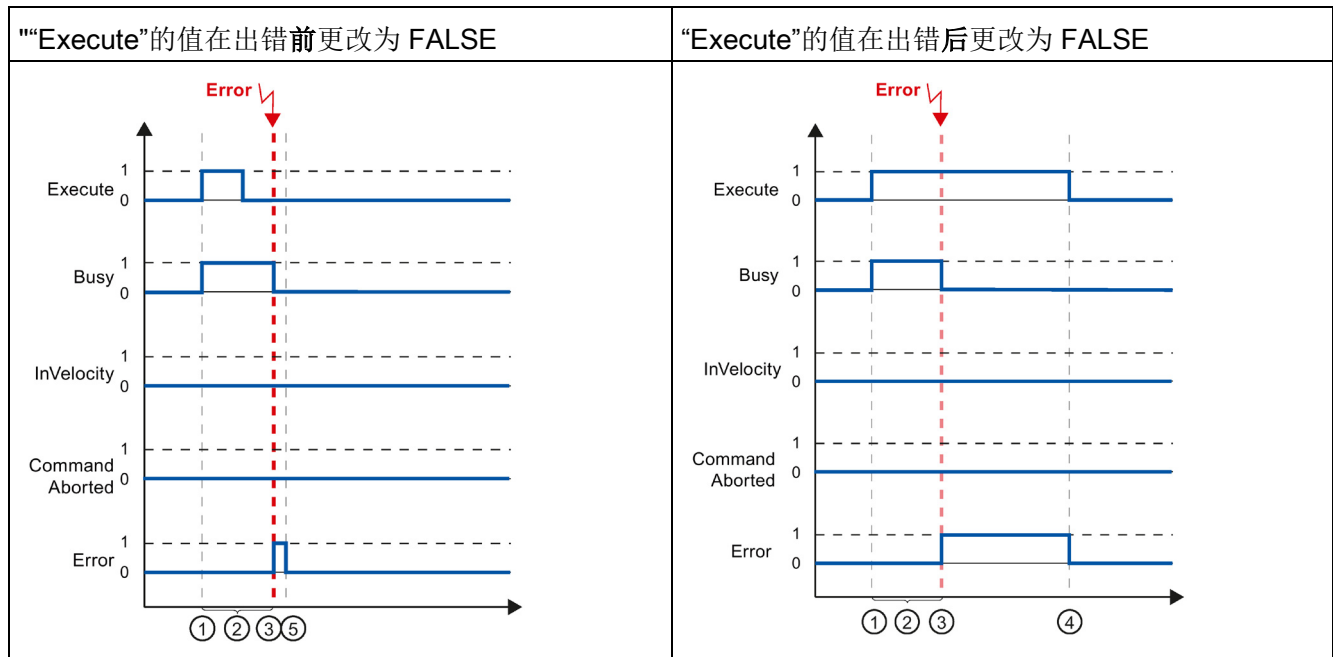
在以下条件下，输出参数“CommandAborted”不指示出现中止：

已达到设置的速度，输入参数“Execute”的值为 FALSE，并且已启动一个新的运动控制命令。

### 达到设置速度前出错

如果在运动控制命令执行期间，达到设置速度之前出错，则通过输出参数“Error”的 TURE 值对此进行指示。

输入参数“Execute”的信号状态影响输出参数“Error”中的显示持续时间：



①	输入参数“Execute”出现上升沿时启动命令。 根据编程情况，“Execute”在命令执行期间仍然可能重置为值 FALSE，或者保持为值 TURE，直到出现错误为止。
②	激活命令时，输出参数“Busy”的值将为 TRUE。
③	执行命令过程中出错。出错时，输出参数“Busy”的值将变为 FALSE 且“Error”的值将变为 TRUE。
④	如果“Execute”的值在出错之前保持为 TRUE，则“Error”的值也将保持为 TRUE 并且其状态仅随“Execute”一起变为 FALSE。
⑤	如果“Execute”已在出错之前重置为 FALSE，则“Error”的值仅在一个执行周期内为 TRUE。

#### 说明

在以下条件下，输出参数“Error”不指示出现错误：

已达到设置的速度，输入参数“Execute”的值为 FALSE，并且发生轴错误（例如，逼近软限位开关）。

轴错误仅在运动控制指令“MC\_Power”中指示。



#### 1.9.5.4 运动控制指令 MC\_MoveJog

运动控制指令“MC\_MoveJog”的命令将实现点动操作。

运动控制命令“MC\_MoveJog”无需定义明确的结束标记。

第一次达到设置的速度且轴恒速运转时，就达到了该命令的目标。

如果达到设置的速度，则通过输出参数“lnVelocity”的 TURE 值对此进行指示。

输入参数“JogForward”或“JogBackward”的值已设置为 FALSE

并且轴已停止时，命令完成。

输出参数“Busy”、“CommandAborted”和“Error”分别指示命令仍在处理、已中止或有未决的错误。

在运动控制命令的处理过程中，输出参数“Busy”的值将为 TRUE。

如果命令已完成、中止或因错误停止，则输出参数“Busy”的值将变为 FALSE。

只要轴在以设置的速度运转，输出参数“lnVelocity”的值就为 TRUE。

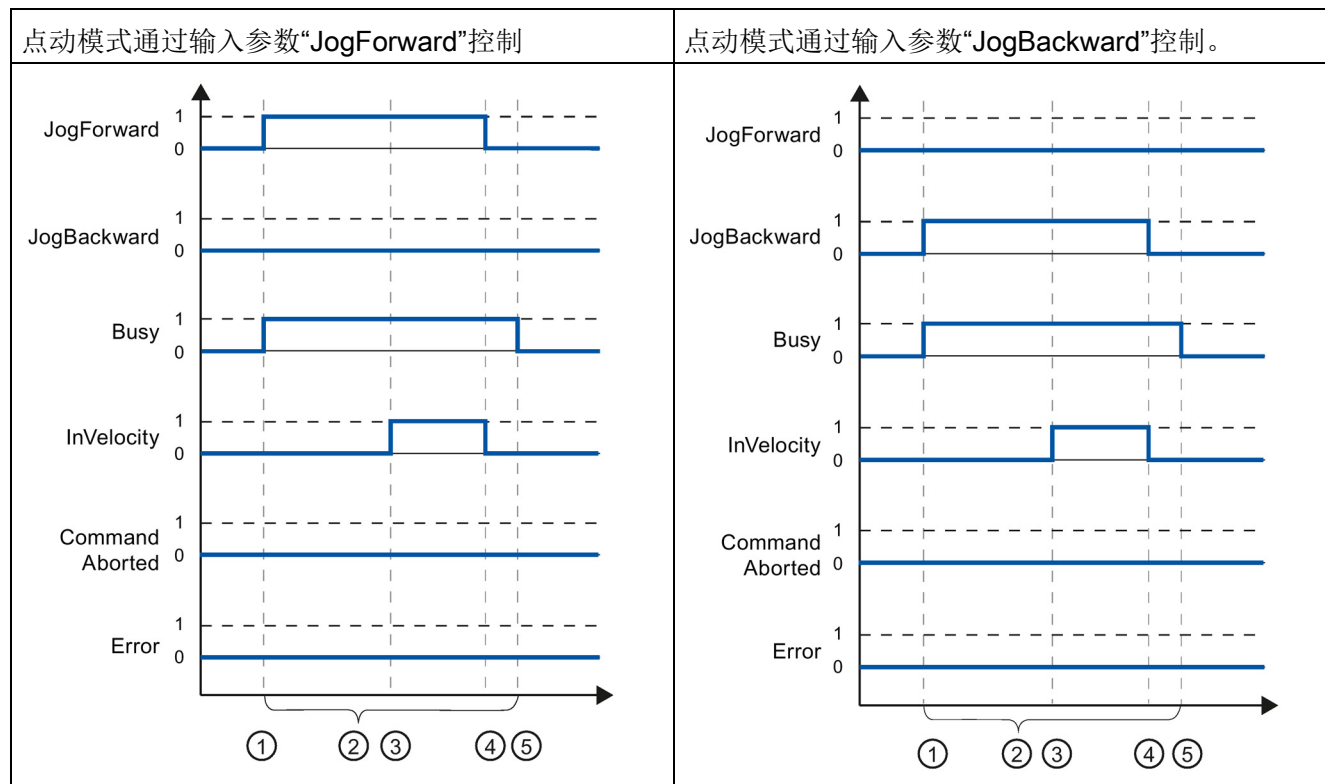
输出参数“CommandAborted”和“Error”保持该状态至少一个周期。

只要输入参数“JogForward”或“JogBackward”设置为 TRUE，就锁存这些状态消息。

下图针对各种示例情况显示了状态位的操作：

### 达到并维持设置的速度

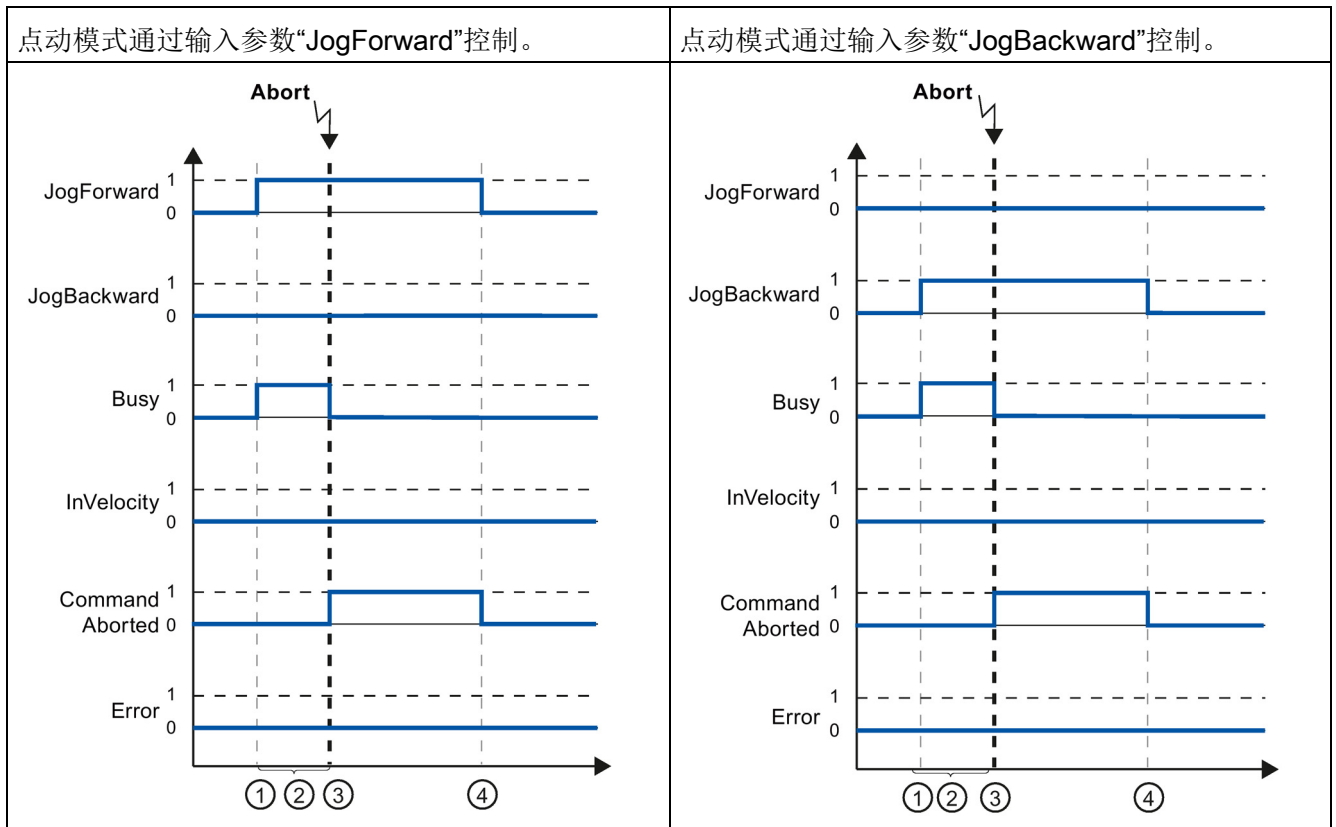
如果运动控制命令在达到设置的速度前执行完毕，则将通过输出参数“**InVelocity**”的值为 **TURE** 的方式对此进行指示。



①	输入参数“JogForward”或“JogBackward”的上升沿时，将启动该命令。
②	激活命令时，输出参数“Busy”的值将为 TRUE。
③	达到设置的速度时，输出参数“ <b>InVelocity</b> ”的值变为 TRUE。
④	输入参数“JogForward”或“JogBackward”的值重置为 FALSE 时，轴运动结束。轴开始减速。结果，轴不再恒速运转并且输出参数“ <b>InVelocity</b> ”的状态变为 FALSE。
⑤	如果轴已停止，则运动控制命令执行完成。同时将输出参数“Busy”的值变为 FALSE。

## 在执行过程中止命令

如果运动控制命令在执行期间中止，则将通过输出参数“CommandAborted”的值为 TRUE 的方式对此进行指示。该行为与是否达到设置的速度无关。



①	输入参数“JogForward”或“JogBackward”的上升沿时，将启动该命令。
②	激活命令时，输出参数“Busy”的值将为 TRUE。
③	在执行命令的过程中，该命令可由其它运动控制命令中止。 如果中止命令，则输出参数“Busy”的值将变为 FALSE，“CommandAborted”的值将变为 TRUE。
④	输入参数“JogForward”或“JogBackward”的值重置为 FALSE 时， 输出参数“CommandAborted”的值将变为 FALSE。

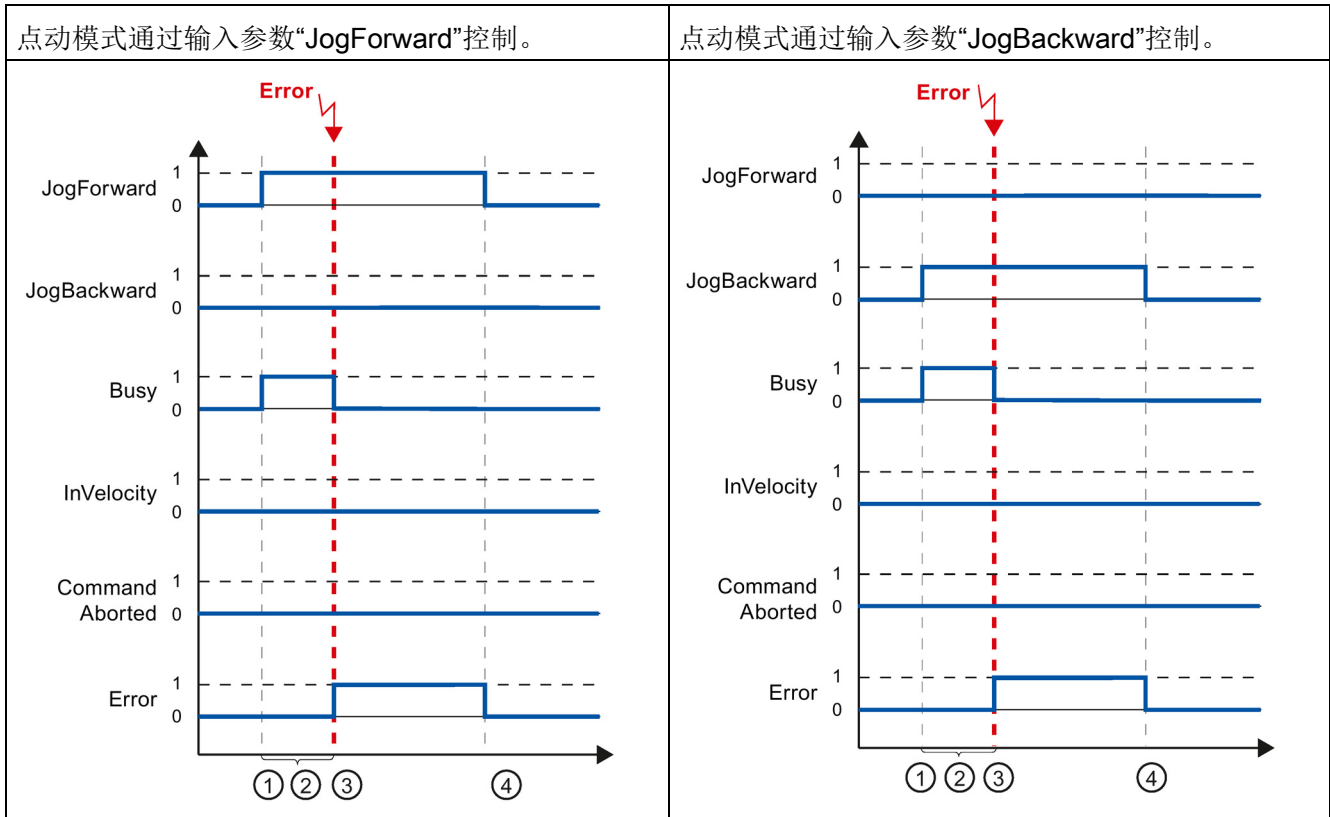
### 说明

如果满足以下所有条件，则将在输出参数“CommandAborted”中指示命令中止并且仅持续一个执行周期：

输入参数“JogForward”和“JogBackward”的值为  
FALSE（但轴仍在减速），并且新的运动控制命令已启动。

在执行命令过程中出错

如果在执行运动控制命令中出错，则将通过输出参数“Error”的值为 TRUE 的方式对此进行指示。该行为与是否达到设置的速度无关。



①	输入参数“JogForward”或“JogBackward”的上升沿时，将启动该命令。
②	激活命令时，输出参数“Busy”的值将为 TRUE。
③	执行命令过程中出错。出错时，输出参数“Busy”的值将变为 FALSE 且“Error”的值将变为 TRUE。
④	输入参数“JogForward”或“JogBackward”的值重置为 FALSE 时，输出参数“Error”的值将变为 FALSE。

说明

如果满足以下所有条件，则在输出参数“Error”中指示出错并且仅持续一个执行周期：  
 输入参数“JogForward”和“JogBackward”的值为 FALSE（但轴仍在减速），并且发生新错误（例如，逼近软限位开关）。

## 1.9.6 运动控制语句的错误显示

运动控制指令将在运动控制指令的输出参数“Error”、“ErrorID”和“ErrorInfo”中指示运动控制命令和工艺对象中的所有错误。

### 在输出参数“Error”、“ErrorID”和“ErrorInfo”中显示错误信息

如果输出参数“Error”的值为 TRUE，则无法执行整个命令或其中各个部分。错误原因通过输出参数“ErrorID”的值进行指示。输出参数“ErrorInfo”的值返回关于错误原因的详细信息。我们区分以下用于错误指示的错误类别：

- **伴随轴停止的运行错误（例如，“已逼近硬限位开关”）**

伴随轴停止的运行错误发生在用户程序运行期间。

如果轴正在运转，则将以组态的减速度或急停减速度使轴停止，具体视错误情况而定。此类错误在错误触发运动控制指令和运动控制指令“MC\_Power”中指示。

- **不伴随轴停止的运行错误（例如，“轴未归位”）**

不伴随轴停止的运行错误发生在用户程序运行期间。

如果轴正在运转，则将继续运转。此类错误仅在触发错误的运动控制指令中指示。

- **运动控制指令中组态出错  
（例如，参数“Velocity”的值错误）**

如果在运动控制指令的输入参数中指定的信息不正确，则将发生参数化错误。

如果轴正在运转，则将继续运转。此类错误仅在触发错误的运动控制指令中指示。

- **“轴”工艺对象的组态错误（例如，“加速度”的值无效）**

如果在轴组态中错误组态一个或多个参数，或者在程序运行期间错误修改可编辑的组态数据，则将发生组态错误。以组态的急停减速度使正在运转的轴停止。

此类错误在错误触发运动控制指令和运动控制指令“MC\_Power”中指示。

- **“命令表”工艺对象的组态错误（例如，“速度”的值无效）**

如果在轴命令表中错误设置一个或多个参数，或者在程序运行期间错误修改可编程的组态数据，则将发生组态错误。如果轴正在运转，则将继续运转。

此类错误仅在运动控制指令“MC\_CommandTable”中指示。

- **内部错误**

发生内部错误时，将使轴停止。此类错误在错误触发运动控制指令中指示，有时也在运动控制指令“MC\_Power”中指示。

有关 ErrorID 和 ErrorInfo 的详细说明及其解决方法，请参见附录 (页 205)。

## 参见

- 运动控制语句概述 (页 162)
- 创建用户程序 (页 163)
- 编程注意事项 (页 166)
- 断电和重新启动后运动控制命令的行为 (页 168)
- 监视激活的命令 (页 169)
- ErrorID 和 ErrorInfo 列表 (自工艺对象 V4 起) (页 205)

## 1.9.7 重新启动工艺对象

### 说明

在接通 CPU 之后或将工艺对象下载到 CPU 之后，系统将使用工艺对象数据块的起始值自动初始化工艺对象。如果在向 CPU 重新进行加载时检测到会引起重新启动的相关变化，则将自动重新启动工艺对象。

如果在 RUN 模式下用户程序更改了与重新启动相关的数据，则用户必须对工艺对象进行重新初始化，以使所做的更改生效。

如果要在重新启动工艺对象之后保留工艺对象数据块中的更改，则必须使用扩展指令“WRIT\_DBL”将更改写入装载存储器中的起始值中。

### 需要重新启动

如果需要重新启动工艺对象，则可在“工艺对象 > 诊断 > 状态和错误位 > 状态消息 > 需要重新启动”(Technology object > Diagnostics > Status and error bits > Status messages > Restart required) 下和工艺对象 <轴名称>.StatusBits.RestartRequired 的变量中指出。

### 重新启动工艺对象

可通过带有参数“Restart”= TRUE 的“MC\_Reset”运动控制指令，触发工艺对象的重新启动。

重新启动将使用增量实际值 (<轴名称>.StatusBits.HomingDone) 重置工艺对象的“已归位”(Homed) 状态。

## 1.10 轴 - 诊断

### 1.10.1 状态和错误位（自工艺对象 V4 起）

在 TIA Portal 中使用诊断功能“状态和错误位”(Status and error bits)

可监视轴的最重要状态和错误消息。

当轴激活时，可以在“手动控制”模式和“自动控制”模式下在线显示诊断功能。

状态错误消息具有下列含义：

#### 状态消息

状态消息 - 轴	说明
已启用	轴已启用且准备就绪，可通过运动控制命令进行控制。 (工艺对象的变量: <轴名称>.StatusBits.Enable)
已归位	轴已归位，可执行运动控制指令“MC_MoveAbsolute”的绝对定位命令。 对于相对定位而言，轴不必归位。特殊情况： <ul style="list-style-type: none"> <li>• 主动归位过程中，该状态为 FALSE。</li> <li>• 如果归位的轴经受被动归位，则在被动归位过程中该状态设置为 TRUE。</li> </ul> (工艺对象的变量: <轴名称>.StatusBits.HomingDone)
轴错误	“轴”工艺对象发生错误。 有关错误的详细信息，请参见自动控制模式下的运动控制指令的 ErrorID 和 ErrorInfo 参数。在手动模式下，轴控制面板的“错误消息”(Error message) 框可显示有关错误原因的详细信息。 (工艺对象的变量: <轴名称>.StatusBits.Error)
控制面板激活	在轴控制面板中启用了“手动控制”(Manual control) 模式。 轴控制面板对“轴”工艺对象具有优先控制权。不能通过用户程序来控制轴。 (工艺对象的变量: <轴名称>.StatusBits.ControlPanelActive)
需要重新启动	已在 CPU RUN 模式下将已修改的轴组态下载到装载存储器。 要将已修改组态下载到工作存储器，需要重新启动轴。 使用运动控制指令 MC_Reset 执行此操作。 (工艺对象的变量: <轴名称>.StatusBits.RestartRequired)

## 1.10 轴 - 诊断

状态消息 - 驱动器	说明
就绪	驱动器准备好运行。 (工艺对象的变量: <轴名称>.StatusBits.DriveReady)
驱动器错误	驱动器因丢失“驱动器就绪”信号而报告错误。 (工艺对象的变量: <轴名称>.ErrorBits.DriveFault)

状态消息 - 运动	说明
停止	轴处于停止状态。 (工艺对象的变量: <轴名称>.StatusBits.StandStill)
加速度	轴在加速。 (工艺对象的变量: <轴名称>.StatusBits.Accelerating)
恒速	轴在恒速运转。 (工艺对象的变量: <轴名称>.StatusBits.ConstantVelocity)
减速度	轴在减速 (速度下降)。 (工艺对象的变量: <轴名称>.StatusBits.Decelerating)

状态消息 - 运动类型	说明
定位	轴执行运动控制指令“MC_MoveAbsolute”、“MC_MoveRelative” 或者轴控制面板的定位命令。 (工艺对象的变量: <轴名称>.StatusBits.PositioningCommand)
以速度设定值行进	轴以运动控制指令“MC_MoveVelocity”、 “MC_MoveJog”或者轴控制面板的速度设定值执行命令。 (工艺对象的变量: <轴名称>.StatusBits.VelocityCommand)
归位	轴执行运动控制指令“MC_Home”或者轴控制面板的归位命令。 (工艺对象的变量: <轴名称>.StatusBits.HomingCommand)
命令表已激活	该轴由运动控制指令“MC_CommandTable”控制。 (工艺对象的变量: <轴名称>.StatusBits.CommandTableActive)



## 限位开关状态消息

限位开关状态消息	说明
已到达下限软限位开关	已到达或超出软限位开关。 (工艺对象的变量: <轴名称>.StatusBits.SWLimitMinActive)
已到达上限软限位开关	已到达或超出硬限位开关。 (工艺对象的变量: <轴名称>.StatusBits.SWLimitMaxActive)
已到达下限硬限位开关	已到达或超过下限硬限位开关。 (工艺对象的变量: <轴名称>.StatusBits.HWLimitMinActive)
已到达上限硬限位开关	已到达或超过上限硬限位开关。 (工艺对象的变量: <轴名称>.StatusBits.HWLimitMaxActive)

## 错误消息

错误消息	说明
已到达软限位开关	已到达或超出软限位开关。 (工艺对象的变量: <轴名称>.ErrorBits.SWLimit)
已到达硬限位开关	已到达或超出硬限位开关。 (工艺对象的变量: <轴名称>.ErrorBits.HWLimit)
无效的运动方向	命令的运动方向与组态的运动方向不符。 (工艺对象的变量: <轴名称>.ErrorBits.DirectionFault)
PTO 已在使用。	另一个轴正在使用此 PTO (Pulse Train Output) 和 HSC (High Speed Counter) 并且该轴已通过“MC_Power”启用。 (工艺对象的变量: <轴名称>.ErrorBits.HWUsed)
编码器	编码器系统中发生错误。 (工艺对象的变量: <轴名称>.ErrorBits.SensorFault)
数据交换	与所连接设备进行通信时发生错误。 (工艺对象的变量: <轴名称>.ErrorBits.CommunicationFault)
定位	在定位运动的末端, 轴定位错误。 (工艺对象的变量: <轴名称>.ErrorBits.PositionigFault)
跟随误差	超出了允许的最大跟随误差。 (工艺对象的变量: <轴名称>.ErrorBits.FollowingErrorFault)

## 1.10 轴 - 诊断

错误消息	说明
编码器值无效	编码器值无效。 (工艺对象的变量: <轴名称>.StatusSensor.State)
组态错误	错误地组态了“轴”工艺对象, 或者在用户程序运行期间错误地修改了可编辑的组态数据。 (工艺对象的变量: <轴名称>.ErrorBits.ConfigFault)
内部错误	发生内部错误。 (工艺对象的变量: <轴名称>.ErrorBits.SystemFault)

下方的输出窗口显示了首次报告且仍未确认的错误。

## 参见

自 V4 起的 StatusBits 变量 (页 279)

自 V4 起的 ErrorBits 变量 (页 285)

诊断 - 状态和错误位 (“轴”工艺对象 V1...3) (页 316)

变量兼容性列表 (页 40)

运动状态 (页 187)

## 1.10.2 运动状态

在 TIA Portal 中使用诊断功能“运动状态”(Motion status) 可监视轴的运动状态。  
当轴激活时，可以在“手动控制”模式和“自动控制”模式下在线显示诊断功能。  
显示的状态信息的含义如下：

状态	说明
当前位置	“当前位置”(Current position) 框指示当前轴位置。 如果轴未回原点，则该值是相对于轴启用位置的位置值。 (工艺对象的变量: <轴名称>.Position)
当前速度	“当前速度”(Current velocity) 框指示轴当前的速度。 (工艺对象的变量: <轴名称>.Velocity)
目标位置	“目标位置”(Target position) 框指示活动定位命令或轴命令表的当前目标位置。 “目标位置”(Target position) 的值仅在定位命令执行期间有效。 (工艺对象的变量: <轴名称>.StatusPositioning.TargetPosition)
剩余行进距离	“剩余行进距离”(Remaining travel distance) 框指示活动定位命令或轴命令表的当前剩余行进距离。 “剩余行进距离”(Remaining travel distance) 值仅在定位命令执行期间有效。 (工艺对象的变量: <轴名称>.StatusPositioning.Distance)

### 参见

- 自 V4 起的位置变量 (页 235)
- 自 V4 起的速度变量 (页 236)
- 自 V4 起的 StatusPositioning 变量 (页 275)
- 变量兼容性列表 (页 40)
- 状态和错误位 (自工艺对象 V4 起) (页 183)

### 1.10.3 动态设置

在 TIA Portal 中使用“动态设置”(Dynamics settings) 诊断功能来监视轴的动态限值。  
当轴激活时，可以在“手动控制”模式和“自动控制”模式下在线显示诊断功能。

显示的状态信息的含义如下：

动态限制	说明
加速度	“加速度”(Acceleration) 框指示当前组态的轴的加速度。 (工艺对象的变量: <轴名称>.DynamicDefaults.Acceleration)
减速度	“减速度”(Deceleration) 框指示当前组态的轴的减速度。 (工艺对象的变量: <轴名称>.DynamicDefaults.Deceleration)
急停减速度	“急停减速度”(Emergency deceleration) 框指示当前组态的轴的急停减速度。 (工艺对象的变量: <轴名称>.DynamicDefaults.EmergencyDeceleration)
加加速度 (自轴工艺对象 V2 起)	“速度”(Velocity) 框指示组态的当前轴步进速度。 (工艺对象的变量: <轴名称>.DynamicDefaults.Jerk)

#### 参见

自 V4 起的 DynamicDefaults 变量 (页 260)

变量兼容性列表 (页 40)

#### 1.10.4 PROFIdrive 帧

在 TIA Portal 中，“工艺对象 > 诊断 > PROFIdrive 报文”(Technology object > Diagnostics > PROFIdrive telegram) 诊断功能用于监视驱动器和编码器返回的 PROFIdrive 报文。在线操作中将显示该诊断功能。

##### “驱动器”(Drive) 区域

在此区域中，将显示驱动器返回给控制器的 PROFIdrive 报文中所包含的以下参数：

- 状态字“SW1”和“SW2”
- 输出到驱动器的速度设定值 (NSET)
- 从驱动器发送信号的实际速度 (NACT)

##### “编码器”(Encoder) 区域

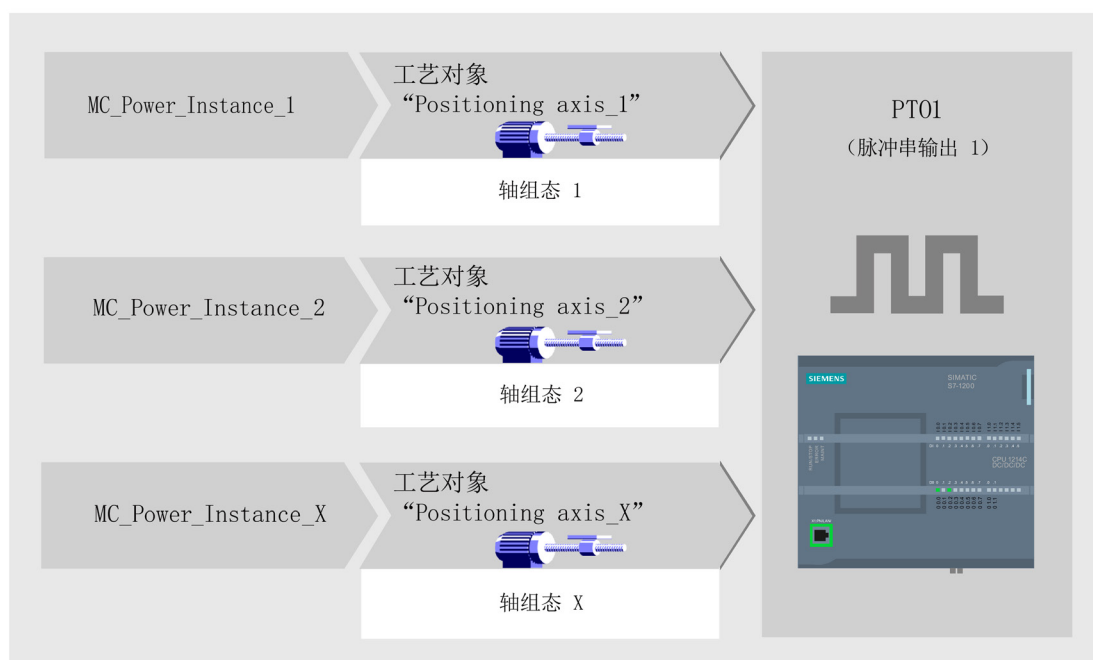
在此区域中，将显示编码器返回给控制器的 PROFIdrive 报文中所包含的以下参数：

- 状态字“G1\_ZSW”
- 实际位置值“G1\_XIST1”（周期性实际编码器值）
- 实际位置值“G1\_XIST2”（编码器的绝对值）

## 1.11 附录

### 1.11.1 将多个轴与同一个 PTO 一起使用

使用 CPU S7-1200 的运动控制功能可通过同一 PTO (Pulse Train Output) 运行多个定位轴工艺对象，因而也可以使用相同的 CPU 输出运行多个定位轴工艺对象。例如，这适用于通过一个 PTO 将不同的轴组态用于不同的生产工序。如下所述，可以根据需要频繁地切换这些轴组态。下图显示了基本的功能关系：

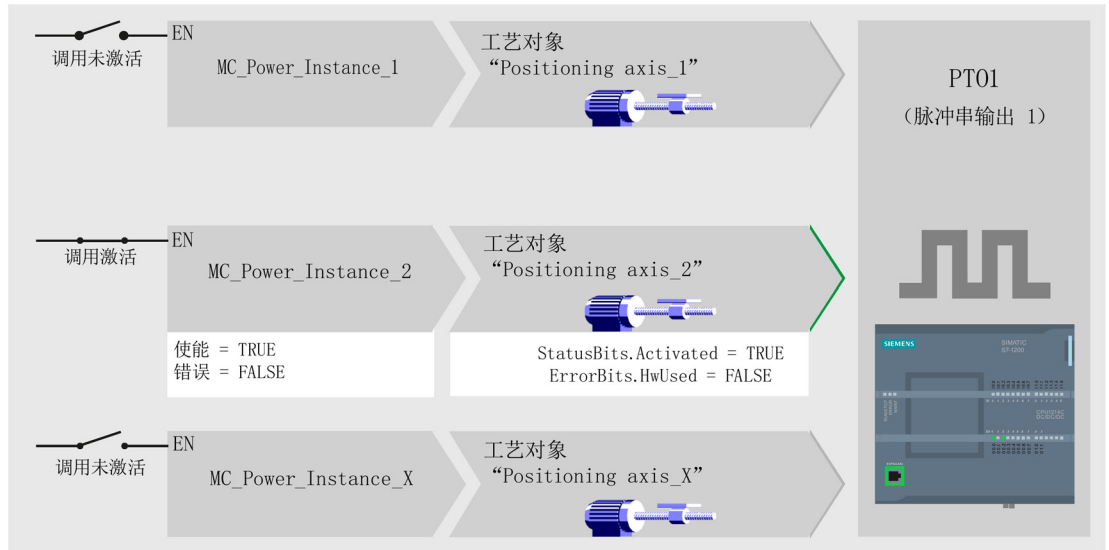


在本例中，多个具有不同轴组态的定位轴工艺对象使用同一 PTO。必须在用户程序中通过单独调用使用独立背景数据块的运动控制指令“MC\_Power”来调用各个轴。每次只能有一个轴使用 PTO。通过变量 `<轴名称>.StatusBits.Activated = TRUE` 指示当前使用 PTO 的轴。

## 切换定位轴工艺对象

下列程序图式说明了如何切换不同的工艺对象，从而切换不同的轴组态。要将一个 PTO 与多个轴一起使用而不会出现错误指示，仅可调用当前所用轴的运动控制指令。

下图使用运动控制指令“MC\_Power”作为示例显示了该原则：



激活轴（此处为“Positioning axis\_2”）的变量在用户程序中通常显示以下内容：

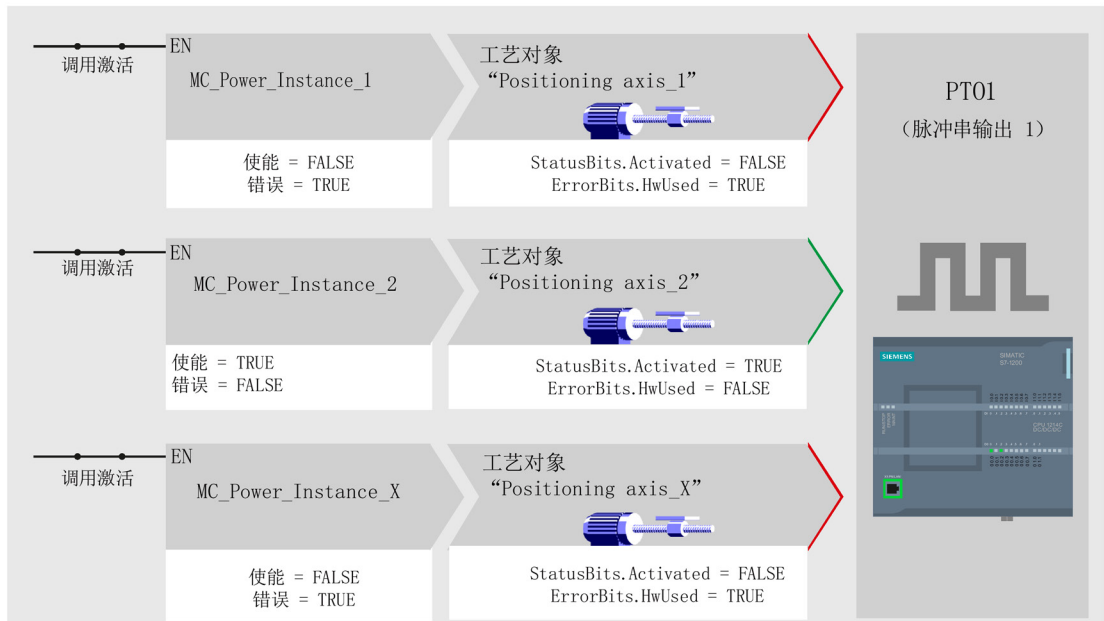
- <轴名称>.StatusBits.Activated = TRUE
- <轴名称>.ErrorBits.HWUsed = FALSE

要从一个定位轴工艺对象切换至另一个，请按以下步骤操作。

在本例中，执行从“Positioning axis\_2”到“Positioning axis\_1”的切换：

1. 结束所激活“Positioning axis\_2”的所有激活行进运动
2. 使用关联的运动控制指令“MC\_Power”（输入参数 Enable = FALSE）禁用“Positioning axis\_2”
3. 要检查“Positioning axis\_2”是否已禁用，需对运动控制指令“MC\_Power”的输出参数 Status = FALSE 与工艺对象变量 <轴名称>.StatusBits.Enable = FALSE 进行“与”运算。
4. 禁用对“Positioning axis\_2”的运动控制指令的有条件调用。
5. 激活对“Positioning axis\_1”的运动控制指令的有条件调用。  
第一次调用相应的运动控制指令“MC\_Power”时，将禁用“Positioning axis\_2”并激活“Positioning axis\_1”。

6. 使用关联的运动控制指令“MC\_Power”（输入参数 Enable = TRUE）启用“Positioning axis\_1”。
  7. 要检查“Positioning axis\_1”是否已启用，请对运动控制指令“MC\_Power”的输出参数 Status = TRUE 与工艺对象变量 <轴名称>.StatusBits.Enable = TRUE 进行“与”运算。
- 也可以循环调用使用单个 PTO 的所有轴的所有运动控制指令。



轴启用（此处为“Positioning axis\_2”）时，该轴进入激活状态。

与有条件调用不同，禁用轴（此处为“Positioning axis\_1”和“Positioning axis\_x”）的运动控制指令将指示错误。这些轴的变量指示状态 <轴名称>.StatusBits.Activated = FALSE 和 <轴名称>.ErrorBits.HWUsed = TRUE。

如果要在执行用户程序的过程中不出现错误提示，则需要对运动控制指令进行有条件调用。

## 参见

自 V4 起的定位轴工艺对象的变量 (页 235)

将多个驱动器与同一个 PTO 一起使用 (页 193)

在更高优先级等级（执行级别）中跟踪作业 (页 194)

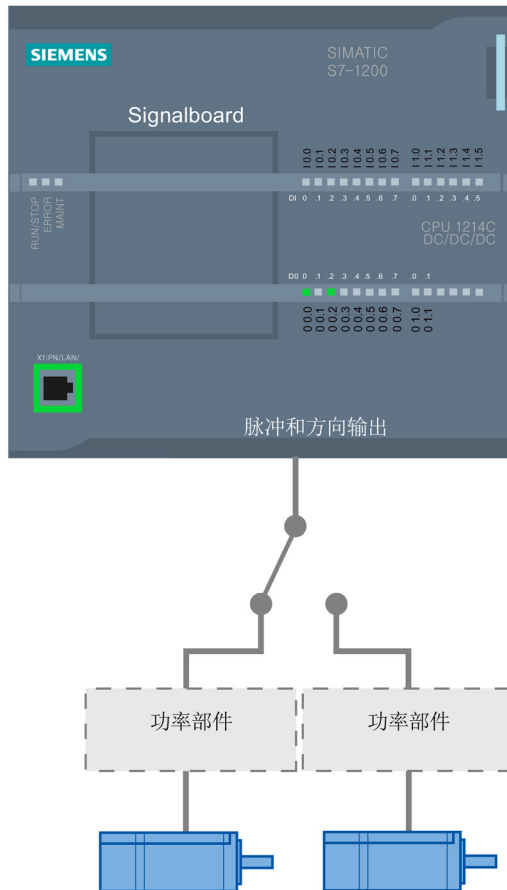
使用通过 PTO 的驱动器连接时的软限位开关时的特殊情况 (页 197)

ErrorID 和 ErrorInfo 列表（自工艺对象 V4 起） (页 205)



### 1.11.2 将多个驱动器与同一个 PTO 一起使用

如果要使用多个驱动器，则可以通过切换的方式使用一个公共 PTO (Pulse Train Output) 来运行这些驱动器。下图显示了基本的电路设计：



如果需要，可由用户程序通过数字量输出来控制驱动器之间的切换。如果各个驱动器需要不同的轴组态，则需要为 PTO 切换这些组态。有关该主题的详细信息，请参见“将多个轴与同一个 PTO 一起使用 (页 190)”。

#### 参见

自 V4 起的定位轴工艺对象的变量 (页 235)

将多个轴与同一个 PTO 一起使用 (页 190)

在更高优先级等级（执行级别）中跟踪作业 (页 194)

使用通过 PTO 的驱动器连接时的软限位开关时的特殊情况 (页 197)

ErrorID 和 ErrorInfo 列表（自工艺对象 V4 起） (页 205)

### 1.11.3 在更高优先级等级（执行级别）中跟踪作业

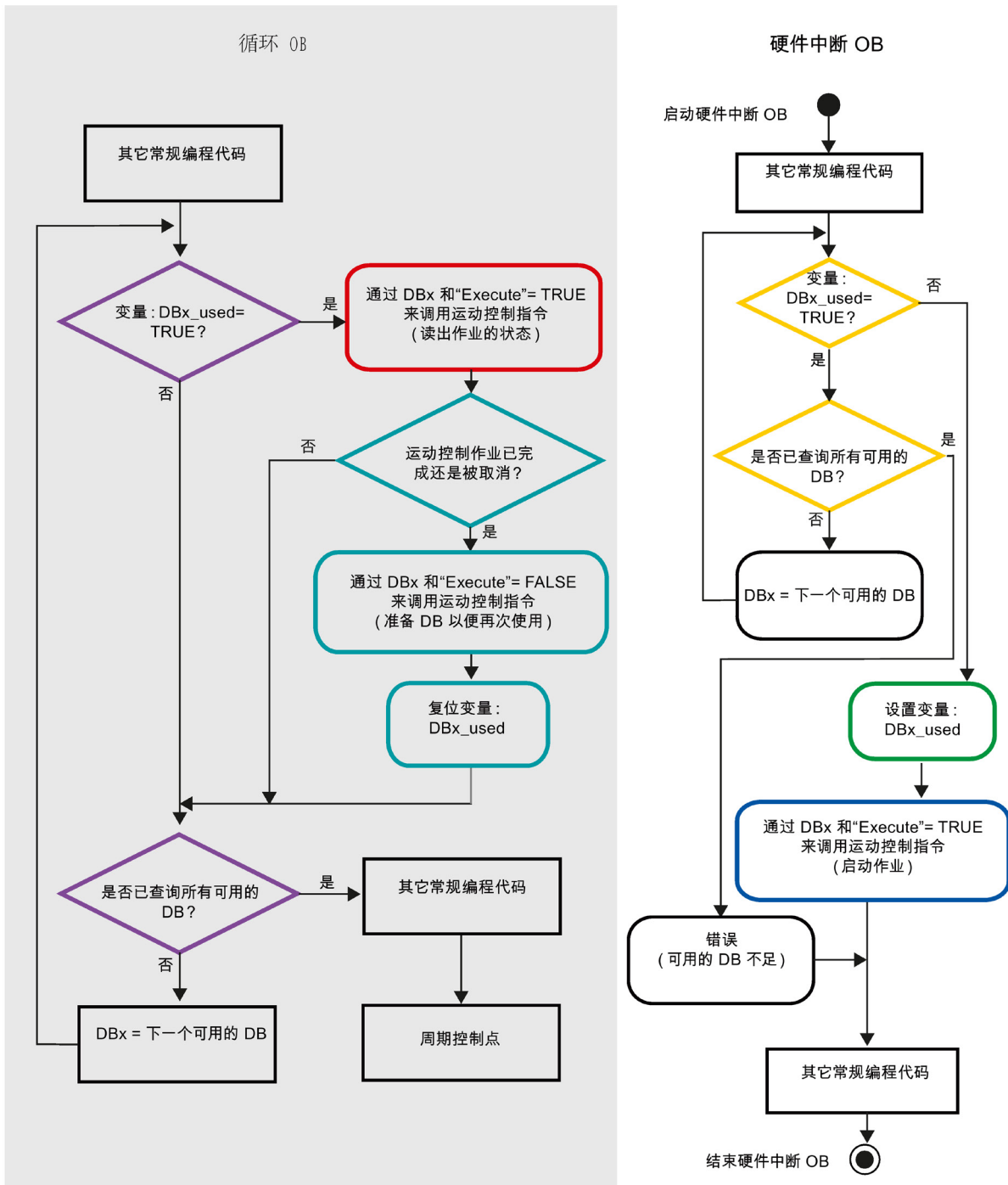
根据应用情况，可能需要在更高优先级等级（执行级别）中启动运动控制命令（例如，中断控制）。

用于状态监控的运动控制指令必须以短周期调用。

如果优先级更高的运动控制命令仅调用一次或调用间隔过大，就不能保证运动控制命令监控的充分性和紧密性。在这些情况之下的跟踪可以在循环 OB 中执行。

当前没有使用的背景数据块对于每个优先级别更高的运动控制命令都必须是可用的。

下面的流程图显示了如何在更高优先级等级（例如，硬件中断 OB）中启动运动控制命令并继续在程序循环 OB 中跟踪：



根据想要启动运动控制命令的频率，必须生成足够数量的背景数据块。用户在 DBx\_used 变量中确定当前使用的背景数据块。

### 在硬件中断 OB 中启动运动控制命令

对 `DBx_used` 变量（橙色）进行二进制查询以查找当前未使用的背景数据块。  
如果找到未被使用的背景数据块，则使用的背景数据块标记为“已使用”（绿色），  
并且将使用该背景数据块（蓝色）启动运动控制命令。

然后执行硬件中断 OB 的所有其它程序部分，接着返回到程序循环 OB。

### 在程序循环 OB 中跟踪启动的运动控制命令

检查循环 OB 中可用的所有背景数据块以确定“`DBx_used`”变量（紫罗兰色）  
当前是否正在使用它们。

如果背景数据块正在使用（正在处理运动控制命令），则调用使用该背景数据块且输入参数 `Execute = TRUE` 的运动控制指令来读出状态消息（红色）。

如果命令完成或已中止，将接着执行以下操作（青绿色）：

- 调用输入参数 `Execute = FALSE` 的运动控制指令
- 重置 `DBx_used` 变量

这随即完成命令跟踪，并且该背景数据块此时可供再次使用。

### 参见

自 V4 起的定位轴工艺对象的变量 (页 235)

将多个轴与同一个 PTO 一起使用 (页 190)

将多个驱动器与同一个 PTO 一起使用 (页 193)

使用通过 PTO 的驱动器连接时的软限位开关时的特殊情况 (页 197)

ErrorID 和 ErrorInfo 列表（自工艺对象 V4 起） (页 205)

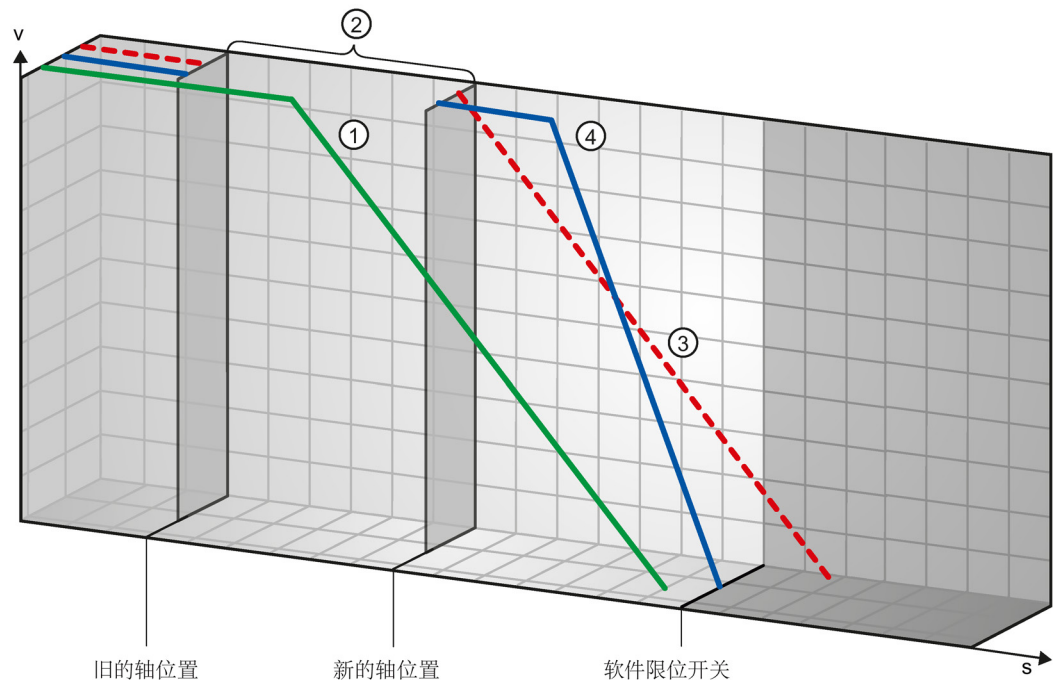
## 1.11.4 使用通过 PTO 的驱动器连接时的软限位开关时的特殊情况

### 1.11.4.1 软件限位开关与回原点操作结合使用

如果回原点作业参数化不当，在软限位开关处可能会影响轴的制动操作。  
开发程序时考虑以下示例。

#### 示例 1:

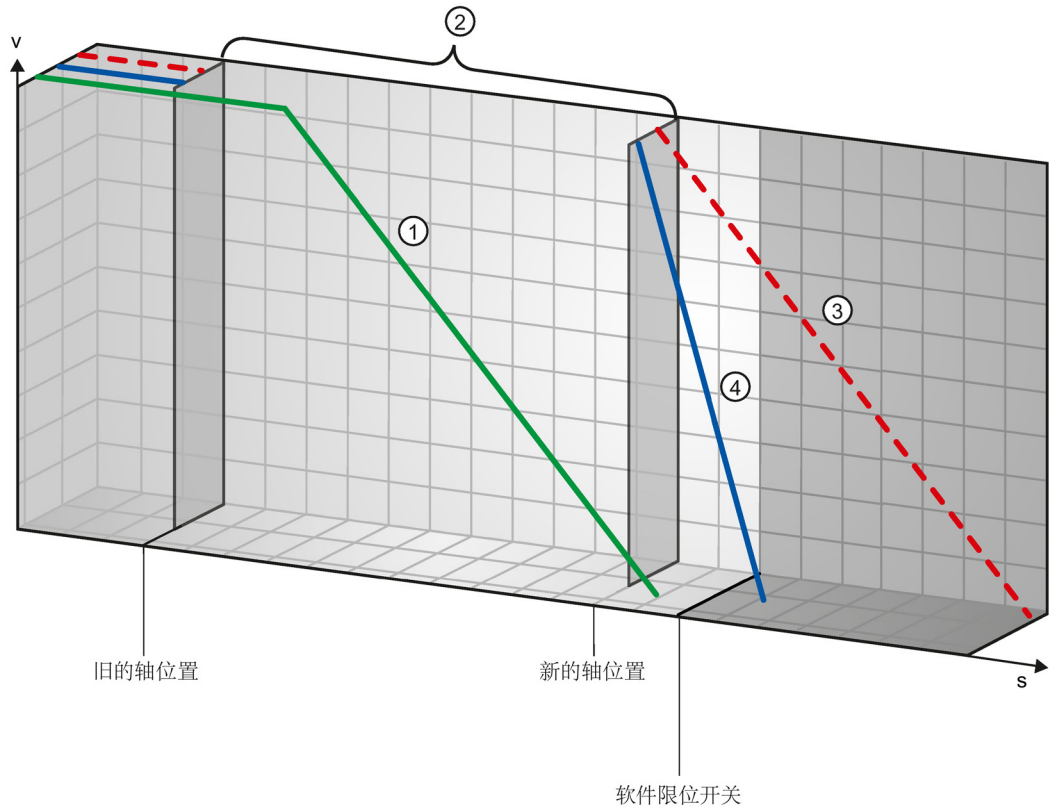
行进命令执行期间，回原点作业（例如，设置参考点）使当前轴位置在软限位开关方向上偏移。仍可在到达软限位开关前使轴停止：



①	绿色曲线显示了无回原点作业时的运动。轴以组态的减速度制动并停止在软限位开关前的某处。
②	由于回原点作业而设置新的轴位置。所以会“跳过”新旧轴位置间的区域。
③	基于新的轴位置，轴理论上可以按组态的减速度制动并停止在软限位开关后的某处（红色曲线）。
④	由于以组态的减速度无法再进行足够的制动，所以轴实际遵从蓝色曲线。 在一段恒速运动后，轴以组态的急停减速度制动并停止在软限位开关的位置处。

示例 2:

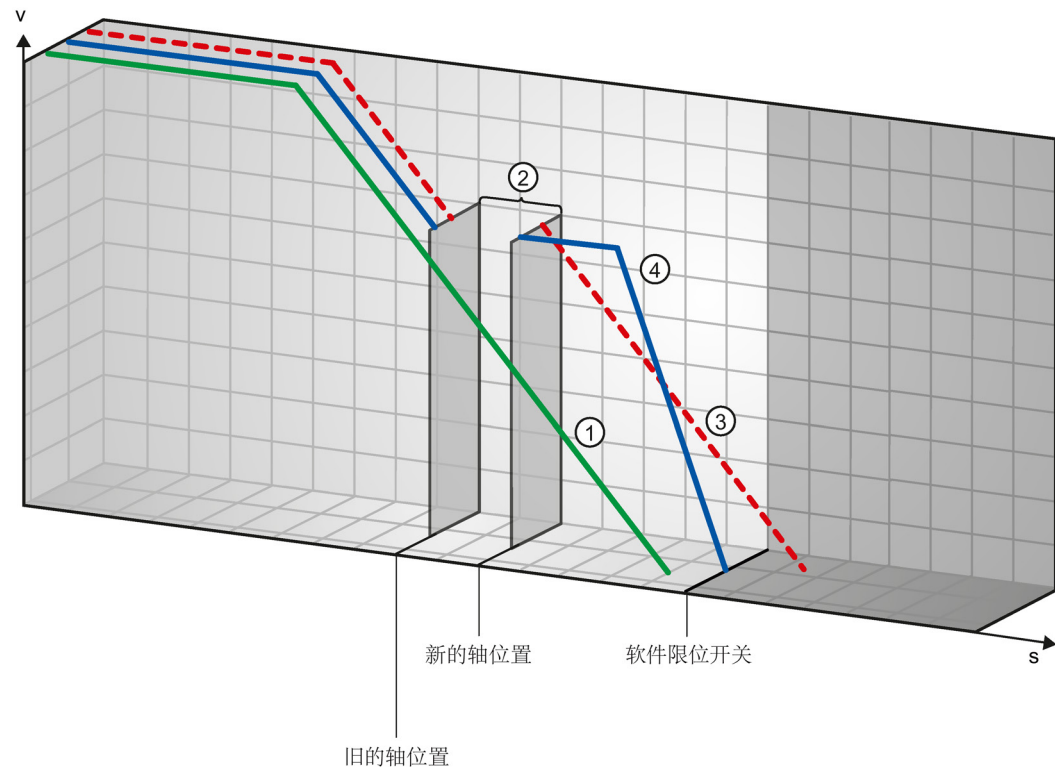
行进命令执行期间，回原点作业（例如，设置参考点）使当前轴位置在软限位开关方向上偏移。与示例 1 不同，无法再在到达软限位开关前使轴停止。  
轴超出软限位开关的位置。



①	绿色曲线显示了无回原点作业时的运动。轴以组态的减速度制动并停止在软限位开关前的某处。
②	由于回原点作业而设置新的轴位置。 所以会“跳过”新旧轴位置间的区域。
③	基于新的轴位置，轴理论上可以按组态的减速度制动并停止在软限位开关后较远的某处（红色曲线）。
④	由于以组态的减速度无法再进行足够的制动，所以轴实际遵从蓝色曲线。 轴以组态的急停减速度制动。 但该急停减速度不足以使轴停止在软限位开关的位置处。 将超出软限位开关的位置。

## 示例 3:

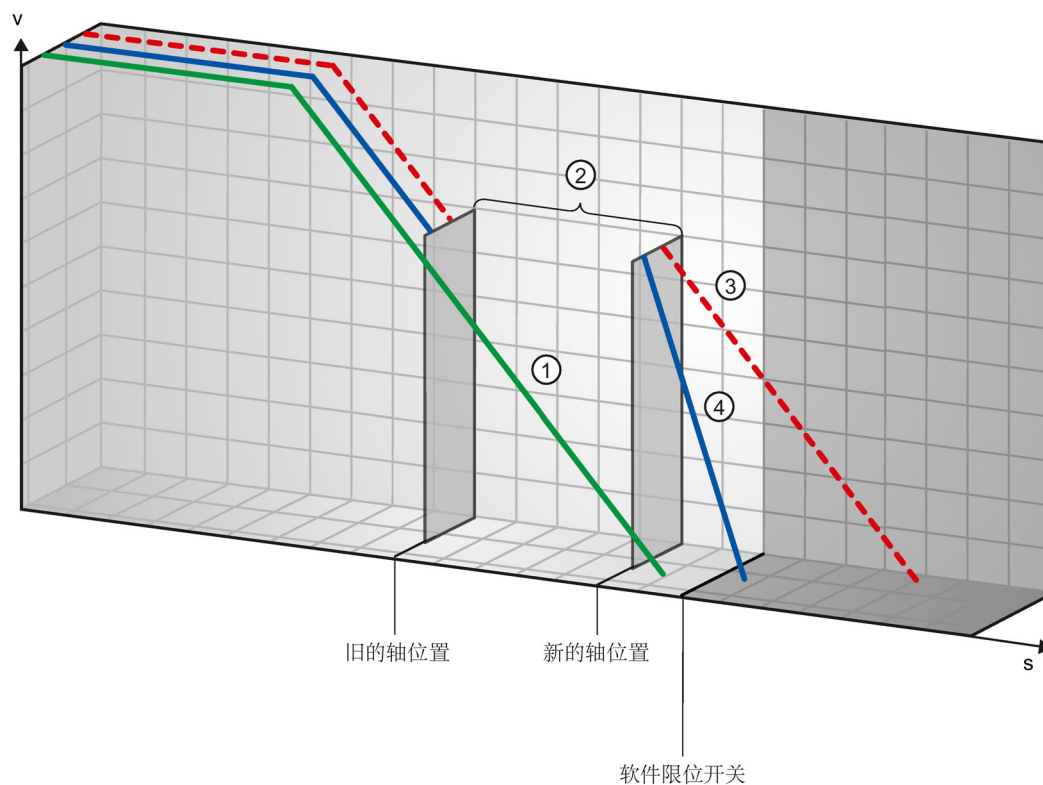
制动操作期间，回原点作业（例如，设置参考点）使当前轴位置在软限位开关方向上偏移。仍可在到达软限位开关前使轴停止：



①	绿色曲线显示了无回原点作业时的运动。轴以组态的减速度制动并停止在软限位开关前的某处。
②	由于回原点作业而设置新的轴位置。所以会“跳过”新旧轴位置间的区域。
③	基于新的轴位置，轴理论上可以按组态的减速度制动并停止在软限位开关后的某处（红色曲线）。
④	由于以组态的减速度无法再进行足够的制动，所以轴实际遵从蓝色曲线。 在一段恒速运动后，轴以组态的急停减速度制动并停止在软限位开关的位置处。

示例 4:

制动操作期间，回原点作业（例如，设置参考点）使当前轴位置在软限位开关方向上偏移。与示例 3 不同，无法再在到达软限位开关前使轴停止。轴超出软限位开关的位置。



①	绿色曲线显示了无回原点作业时的运动。轴以组态的减速度制动并停止在软限位开关前的某处。
②	由于回原点作业而设置新的轴位置。所以会“跳过”新旧轴位置间的区域。
	基于新的轴位置，轴理论上可以按组态的减速度制动并停止在软限位开关后较远的某处（红色曲线）。
④	由于以组态的减速度无法再进行足够的制动，所以轴实际遵从蓝色曲线。轴以组态的急停减速度制动。但该急停减速度不足以使轴停止在软限位开关的位置处。将超出软限位开关的位置。

参见

软件限位开关和软件限位开关的位置变化。(页 201)

软件限位开关与动态更改结合使用(页 202)

触发位置限制时的轴操作(页 79)



#### 1.11.4.2 软件限位开关和软件限位开关的位置变化。

在用户程序运行期间，如果软件限位开关的位置发生异常变化，当前轴位置和软件限位开关的位置之间的距离将会突然减少。

轴的响应与软件限位开关与回原点操作结合使用 (页 197)中所述的内容相似。

#### 参见

软件限位开关与回原点操作结合使用 (页 197)

软件限位开关与动态更改结合使用 (页 202)

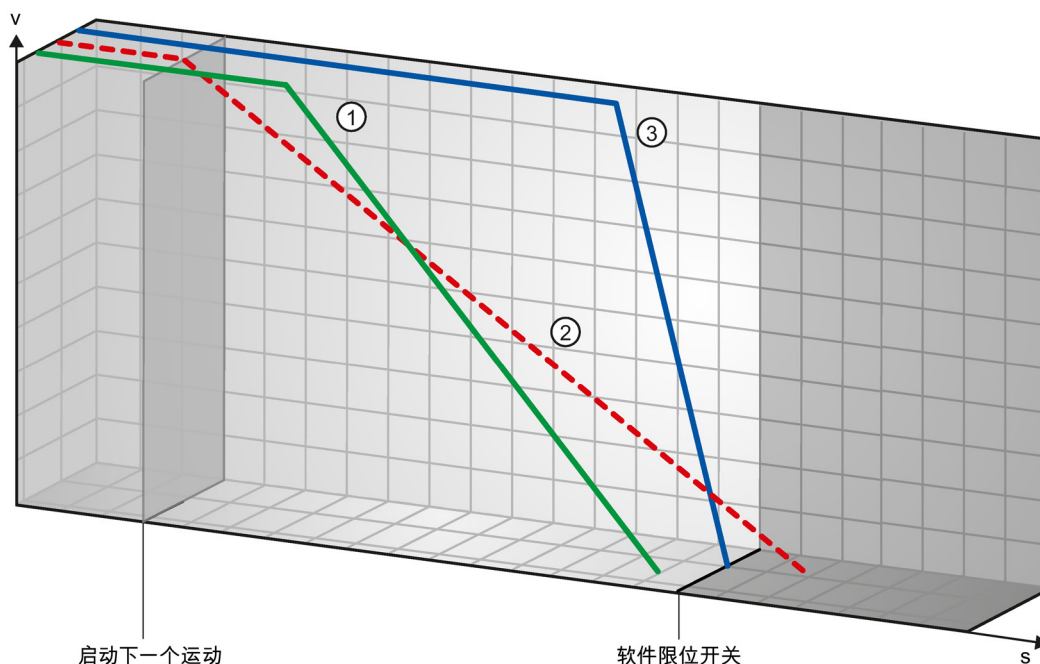
触发位置限制时的轴操作 (页 79)

### 1.11.4.3 软件限位开关与动态更改结合使用

可以结合覆盖运动命令来改变轴在软限位开关区域中的减速度。  
 这适用于启动具有较低减速度（变量“<轴名称>.DynamicDefaults.Deceleration”）  
 的覆盖运动命令的情况。开发程序时考虑以下示例。

#### 示例 1:

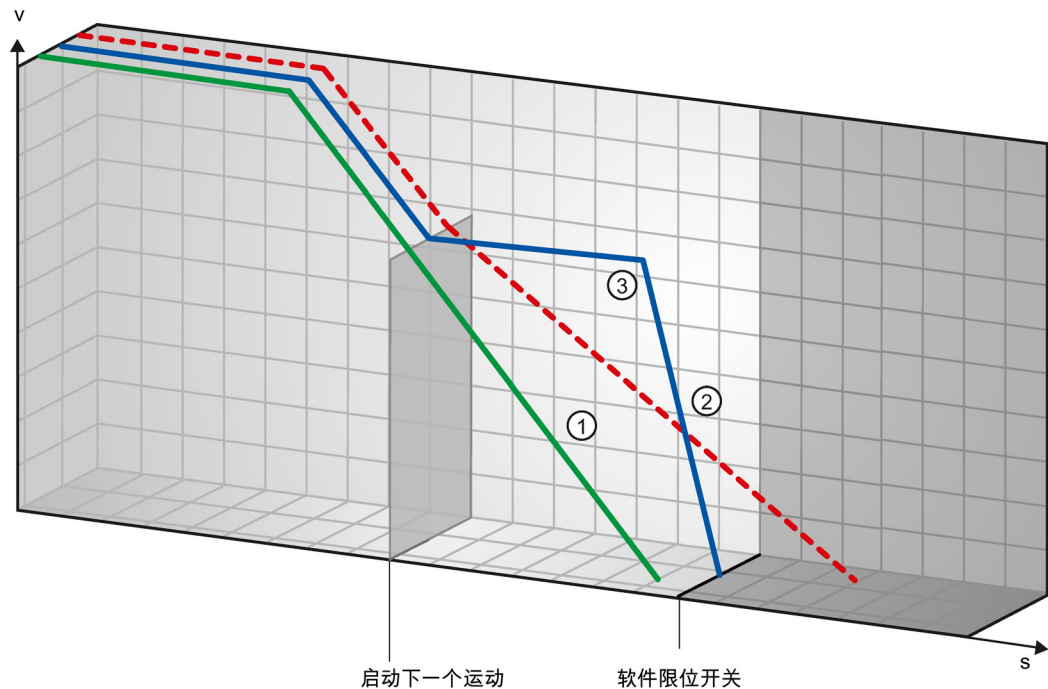
轴运动期间，一个具有较低减速度的运动命令覆盖另一个激活的运动命令：



①	绿色曲线显示了激活的命令未被覆盖时该命令的运动。 轴以组态的减速度制动并停止在软限位开关前的某处。
	基于具有较低减速度的覆盖运动命令，轴理论上会以组态的减速度停止在软限位开关后的某处（红色曲线）。
③	由于以覆盖运动命令的组态减速度无法再进行足够的制动，因此轴实际上遵从蓝色曲线。 在一段恒速运动后，轴以急停减速度制动并停止在软限位开关的位置处。

## 示例 2:

轴制动期间，一个具有较低减速度的运动命令覆盖另一个激活的运动命令：



①	绿色曲线显示了激活的命令 <b>未被覆盖</b> 时该命令的运动。 轴以组态的减速度制动并停止在软限位开关前的某处。
②	基于具有较低减速度的覆盖运动命令，轴理论上会停止在软限位开关后较远的某处（红色曲线）。
③	由于以覆盖运动命令的组态减速度无法再进行足够的制动，因此轴实际上遵从蓝色曲线。 在一段恒速运动后，轴以急停减速度制动并停止在软限位开关的位置处。

## 参见

软件限位开关与回原点操作结合使用 (页 197)

软件限位开关和软件限位开关的位置变化。(页 201)

触发位置限制时的轴操作 (页 79)

### 1.11.5 降低短时间定位的速度

当计划的定位时间小于 2 ms 时，CPU 可以降低定位命令的速度。

然后，命令执行的速度将会在整个持续时间内降低。

降低的速度（以脉冲/秒为单位）计算如下：

- 降低的速度 = 要输出的脉冲数 \* 500Hz

如果计划的定位时间大于 2 ms，则速度不降低。

### 1.11.6 启动/停止速度的动态调整

如果组态速度限值（启动/停止速度和最大速度）、动态值（加速度、减速度和冲击）以及运行命令的目标速度，在某些情况下可能导致 CPU 动态调整启动/停止速度。

例如，由于组态的启动/停止速度较低，第一个脉冲所需的时间将长于整个加速过程可能需要的时间，就属于这种情况。

在这些情况下的第一个脉冲输出速度比组态的启动/停止速度大很多。

后续的脉冲也会动态地调整以确保加速过程可以在指定的时间内完成。

确保在脉冲丢失的情况下，所用的硬件（驱动器）会通过调整适应这种情况，或更改轴的动态设置以避免启动/停止速度的动态调整。

### 1.11.7 ErrorID 和 ErrorInfo 列表（自工艺对象 V4 起）

下表列出了运动控制指令中指示的所有 ErrorID 和 ErrorInfo。  
除错误原因外，还列出了清除错误的解决方法：

根据错误响应，轴在停止时如果发生操作错误，将会停止轴。  
可能会出现以下错误响应：

- **取消启用**

输出设定点 0，并取消启用。轴将根据驱动器中的组态进行制动，并转入停止状态。

- **通过急停斜坡功能进行停止**

将中止处于激活状态的运动命令。轴将通过在“工艺对象 > 扩展参数 > 动态 > 急停斜坡功能”(Technology object > Extended parameters > Dynamics > Emergency stop ramp)

中所组态的急停减速度功能进行制动（没有任何加加速度的限制），  
并转入停止状态。

## 伴随轴停止的运行错误

ErrorID	ErrorInfo	说明	解决方法	错误反应
<b>16#8000</b>		<b>驱动器错误，丢失“驱动器就绪”信号</b>		-
	16#0001	-	使用指令“MC_Reset”确认错误；提供驱动器信号；根据需要重新启动命令	
<b>16#8001</b>		<b>已触发下限软限位开关</b>		通过急停斜坡功能进行停止
	16#000E	已经以当前组态的减速度到达下限软限位开关的位置	使用“MC_Reset”指令确认错误；使用运动命令使轴沿正向移动，超出软限位开关的范围	
	16#000F	已经以急停减速度到达下限软限位开关的位置		
	16#0010	已经以急停减速度超出下限软限位开关的位置		
<b>16#8002</b>		<b>已触发上限软限位开关</b>		通过急停斜坡功能进行停止
	16#000E	已经以当前组态的减速度到达上限软限位开关的位置	使用“MC_Reset”指令确认错误；使用运动命令使轴沿负向移动，超出软限位开关的范围	
	16#000F	已经以急停减速度到达上限软限位开关的位置		
	16#0010	已经以急停减速度超出上限软限位开关的位置		
<b>16#8003</b>		<b>已到达下限硬限位开关</b>		通过 PTO (Pulse Train Output) 的驱动器连接： 通过急停斜坡功能进行停止 通过 PROFIdrive/模拟量输出的驱动器连接： 取消启用
	16#000E	已到达下限硬限位开关。轴以急停减速度停止。 (主动归位期间，未找到归位开关)	使用指令“MC_Reset”确认已启用轴的错误；使用运动命令使轴沿正向移动，超出硬限位开关的范围。	

ErrorID	ErrorInfo	说明	解决方法	错误反应
<b>16#8004</b>		<b>已到达上限硬限位开关</b>		通过 PTO (Pulse Train Output) 的驱动器连接： 通过急停斜坡功能进行停止 通过 PROFIdrive/模拟量输出的驱动器连接： 取消启用
	16#000E	已到达上限硬限位开关。 轴以急停减速度停止。  (主动归位期间，未找到归位开关)	使用指令“MC_Reset”确认已启用轴的错误；使用运动命令使轴沿负向移动，超出硬限位开关的范围。	
<b>16#8005</b>		<b>PTO/HSC 已被另一个轴占用</b>		-
	16#0001	-	<p><b>轴未正确组态：</b> 更正 PTO (Pulse Train Output)/HSC (High Speed Counter) 的组态并将其下载到控制器</p> <p><b>多个轴将通过一个 PTO 运行：</b> 另一个轴正在使用 PTO/HSC。如果要控制当前轴，则必须使用“MC_Power”(Enable = FALSE) 禁用另一个轴。 (另请参见将多个轴与同一个 PTO 一起使用 (页 190))</p>	
<b>16#8006</b>		<b>控制面板中发生通信错误</b>		取消启用
	16#0012	已超时	检查电缆连接，然后再次按下“手动控制”(Manual control) 按钮。	
<b>16#8007</b>		<b>该轴无法启用</b>		-
	16#0025	正在重新启动	请等待，直到轴重新启动完成。	
	16#0026	正在 RUN 模式下执行加载过程	请等待，直到加载过程完成。	

ErrorID	ErrorInfo	说明	解决方法	错误反应
<b>16#8008</b>		<b>无效的运动方向</b>		-
	16#002E	不允许所选的运动方向。	<ul style="list-style-type: none"> <li>调整运动方向，然后重新启动该命令。</li> <li>在工艺对象组态的“扩展参数 &gt; 机械”(Extended parameters &gt; Mechanics) 下可调整允许的旋转方向。 重新启动命令。</li> </ul>	
	16#002F	不能将所选的运动方向反转。		
<b>16#8009</b>		<b>未找到参考开关/编码器零位标记</b>		通过急停斜坡功能进行停止
	16#0033	组态、编码器的硬件或安装、归位开关中出现错误。	<ul style="list-style-type: none"> <li>连接合适的设备。</li> <li>检查设备 (I/O)。</li> <li>比较 HW Config 中的组态和工艺对象。</li> </ul>	
<b>16#800A</b>		<b>编码器发出报警消息</b>		取消启用
	16#0001	-	检查设备的功能、连接和 I/O。	
	16#0034	编码器中的硬件错误		
	16#0035	编码器变脏		
	16#0036	读取编码器绝对值期间出错	将驱动器中的编码器类型或编码器参数 P979 与工艺对象的组态数据进行比较。	
	16#0037	监视编码器的零位标记	编码器报告零位标记监视错误 (Gx_XIST2 中的错误代码 0x0002, 请参见 PROFIdrive 配置文件)。 检查设备的电磁兼容性 (EMC)。	
	16#0038	编码器处于“停止”(Parking) 状态	<ul style="list-style-type: none"> <li>有关错误原因, 请查看所连接的驱动器或编码器。</li> <li>检查错误消息是否可能由驱动器或编码器的调试操作触发。</li> </ul>	
	16#0040	PROFIdrive: 总线上的编码器故障 (站故障)	检查设备的功能、连接和 I/O。	
	16#0041	PROFIdrive: 编码器设备状况错误		



ErrorID	ErrorInfo	说明	解决方法	错误反应
<b>16#800B</b>		<b>位置超出范围</b>		取消启用
	16#0039	正方向超出范围	使轴回到有效的实际值范围。	
	16#003A	负方向超出范围		
	16#003B	位置控制时钟周期中实际位置的改变大于模数长度。	调整所用编码器的模数长度。	
<b>16#800C</b>		<b>驱动器发出报警消息</b>		取消启用
	16#0001	-	检查设备的功能、连接和 I/O。	
	16#003C	PROFIdrive: 驱动器信号“请求的控制”故障		
	16#003D	PROFIdrive: 驱动器已关闭		
	16#003E	PROFIdrive: 总线上的驱动器故障 (站故障)		
	16#003F	PROFIdrive: 驱动器状况信号错误	<ul style="list-style-type: none"> <li>检查设备的功能、连接和 I/O。</li> <li>将 HW Config 的时钟参数 (PROFIBUS 线路、驱动器或编码器的从 OM) 和执行系统的时钟参数进行比较。Tmapc 和伺服必须以相同的时钟周期时间组态。</li> </ul>	
<b>16#800D</b>		<b>超出了允许跟随误差。</b>		取消启用
	16#0001	-	<ul style="list-style-type: none"> <li>检查控制回路的组态。</li> <li>检查编码器的信号方向。</li> <li>检查跟随误差监控的组态。</li> </ul>	

ErrorID	ErrorInfo	说明	解决方法	错误反应
<b>16#800E</b>		<b>硬限位开关处发生错误</b>		取消启用
	16#0042	激活硬限位开关的任意行进方向非法	由于激活了硬限位开关，将禁用编程的运动方向。向相反方向缩回轴。	
	16#0043	硬限位开关极性反转，轴无法释放	检查硬限位开关的机械组态。	
	16#0044	两个硬限位开关均激活，轴无法释放		
<b>16#800F</b>		<b>目标范围发生错误</b>		取消启用
	16#0045	未达到目标范围	在定位容差时间内未达到目标范围。 <ul style="list-style-type: none"> <li>• 检查定位监控的组态。</li> <li>• 检查控制回路的组态。</li> </ul>	
	16#0046	再次离开目标范围	在最短停留时间内已离开目标范围。 <ul style="list-style-type: none"> <li>• 检查定位监控的组态。</li> <li>• 检查控制回路的组态。</li> </ul>	
<b>16#8010</b>		<b>不是模数轴时，下限软限位开关的位置值大于上限软限位开关的位置值</b>		取消启用
	16#0001	-	更改软限位开关的位置。	

## 不伴随轴停止的运行错误

ErrorID	ErrorInfo	说明	解决方法
16#8200		轴未启用	
	16#0001	-	启用轴；重新启动命令
16#8201		轴已由另一个“MC_Power”实例启用	
	16#0001	-	仅通过一个“MC_Power”实例启用轴
16#8202		同时执行的运动控制命令最大数量（对于通过 PTO (Pulse Train Output) 的驱动器连接，最多同时运行 200 条命令，对于通过 PROFIdrive/模拟量输出的驱动器连接，最多同时运行 100 条命令）	
	16#0001	-	减少并行激活命令数；重新启动命令 如果运动控制指令的参数“Busy”= TRUE，则说明命令已激活。
16#8203		轴当前在“手动控制”(Manual control) 模式（轴控制面板）下运行	
	16#0001	-	退出“手动控制”；重新启动命令
16#8204		轴未归位	
	16#0001	-	使用指令“MC_Home”使轴归位； 重新启动命令
16#8205		轴当前由用户程序控制（该错误仅显示在轴控制面板中）	
	16#0013	轴已在用户程序中启用	使用指令“MC_Power”禁用轴或者在轴控制面板中再次选择“手动控制”(Manual control)
16#8206		工艺对象尚未激活	
	16#0001	-	使用指令“MC_Power”Enable = TRUE 启用轴或者在轴控制面板中启用轴。

ErrorID	ErrorInfo	说明	解决方法
<b>16#8207</b>		<b>命令被拒绝</b>	
	16#0016	正在执行主动归位；无法启动另一种归位方法。	等到主动归位完成或通过运动命令（例如，“MC_Halt”）中止主动归位。
	16#0018	当轴正在主动或被动归位时，不能使用命令表进行移动。	一直等到直接地或被动归位完成。
	16#0019	正在处理命令表时，轴不能主动或被动归位。	等到命令表完成或通过运动命令（例如，“MC_Halt”）中止命令表。
	16#0052	指定的位置超出了数值限制。	在运动控制指令中输入有效的位置值。
	16#0053	轴正在沿斜坡上升。	等到轴准备就绪，可运行。
	16#0054	实际值无效	要执行“MC_Home”命令，实际值必须有效。 请检查实际值的状态。 工艺对象 <轴名称>.“StatusSensor.State”变量的值必须为 2（有效值）。
<b>16#8208</b>		<b>最大速度和启动/停止速度的差值无效</b>	
	16#0002	值不是有效数字	更正该值；重新启动命令
	16#000A	值小于或等于 0。	
<b>16#8209</b>		<b>对工艺对象“轴”的加速度无效</b>	
	16#0002	值不是有效数字	更正该值；重新启动命令
	16#000A	值小于或等于 0。	
<b>16#820A</b>		<b>无法重新启动轴</b>	
	16#0013	轴已在用户程序中启用	使用“MC_Power”指令阻止轴；再次重新启动
	16#0027	轴当前在“手动控制”模式（轴控制面板）下运行	退出“手动控制”；再次重新启动
	16#0047	工艺对象未准备好重新启动。	重新下载项目。
	16#0048	不满足工艺对象的重新启动条件。	禁用工艺对象。

ErrorID	ErrorInfo	说明	解决方法
<b>16#820B</b>		无法执行命令表	
	16#0026	正在 RUN 模式下执行加载过程	请等待，直到加载过程完成。
<b>16#820C</b>		无组态可用	
	16#0001	-	内部错误 联系服务热线。

## 块参数错误

ErrorID	ErrorInfo	说明	解决方法
<b>16#8400</b>		<b>运动控制指令的“Position”参数值无效</b>	
	16#0002	值不是有效数字	更正该值；重新启动命令
	16#0005	该值超出值范围（大于 1E+12）	
	16#0006	该值超出值范围（小于 1E+12）	
<b>16#8401</b>		<b>运动控制指令的“Distance”参数值无效</b>	
	16#0002	值不是有效数字	更正该值；重新启动命令
	16#0005	该值超出值范围（大于 1E+12）	
	16#0006	该值超出值范围（小于 1E+12）	
<b>16#8402</b>		<b>运动控制指令的“Velocity”参数值无效</b>	
	16#0002	值不是有效数字	更正该值；重新启动命令
	16#0008	值大于组态的最大速度	
	16#0009	值小于组态的启动/停止速度	
	16#0024	值小于 0	
<b>16#8403</b>		<b>运动控制指令的“Direction”参数值无效</b>	
	16#0011	所选值无效	更正所选值；重新启动命令

ErrorID	ErrorInfo	说明	解决方法
<b>16#8404</b>		<b>运动控制指令的“Mode”参数值无效</b>	
	16#0011	所选值无效	更正所选值；重新启动命令
	16#0015	未组态主动/被动归位	更正组态并将其下载到控制器；启用轴并重新启动命令
	16#0017	虽然硬限位开关已禁用，但还是激活了在硬限位开关处反向	<ul style="list-style-type: none"> <li>使用变量 &lt;轴名称&gt;.PositionLimitsHW.Active = TRUE 激活硬限位开关，并重新启动该命令</li> <li>更正组态并将其下载到控制器；启用轴并重新启动命令</li> </ul>
	16#0055	增量编码器的模式无效	需使用参数“Mode”= 0、1、2、3 启动增量编码器的归位操作。
	16#0056	绝对值编码器的模式无效	绝对值编码器无法执行被动和主动归位操作（“Mode”= 2、3）。 需使用参数“Mode”= 0、1 启动绝对值编码器的归位操作。
<b>16#8405</b>		<b>运动控制指令的“StopMode”参数值无效</b>	
	16#0011	所选值无效	更正所选值；再次启用轴
<b>16#8406</b>		<b>不允许同时正向和反向点动</b>	
	16#0001	-	采取措施确保参数“JogForward”和“JogBackward”的信号状态不会同时为 TRUE；重新启动命令。
<b>16#8407</b>		<b>只有禁用激活的轴后，才允许使用指令“MC_Power”切换到另一个轴。</b>	
	16#0001	-	禁用激活的轴；然后可以切换到其它轴并启用该轴。
<b>16#8408</b>		<b>运动控制指令的“Axis”参数值无效</b>	
	16#001A	指定的值与所需的工艺对象版本不匹配	更正该值；重新启动命令
	16#001B	指定的值与所需的工艺对象类型不匹配	
	16#001C	指定的值不是运动控制工艺数据块	

ErrorID	ErrorInfo	说明	解决方法
<b>16#8409</b>		<b>运动控制指令的“CommandTable”参数值无效</b>	
	16#001A	指定的值与所需的工艺对象版本不匹配	更正该值；重新启动命令
	16#001B	指定的值与所需的工艺对象类型不匹配	
	16#001C	指定的值不是运动控制工艺数据块	
<b>16#840A</b>		<b>运动控制指令的“StartStep”参数值无效</b>	
	16#000A	值小于或等于 0。	更正该值；重新启动命令
	16#001D	开始步进大于结束步进	
	16#001E	值大于 32	
<b>16#840B</b>		<b>运动控制指令的“EndStep”参数值无效</b>	
	16#000A	值小于或等于 0。	更正该值；重新启动命令
	16#001E	值大于 32	
<b>16#840C</b>		<b>运动控制指令的“RampUpTime”参数值无效</b>	
	16#0002	值不是有效数字	更正该值；重新启动命令
	16#000A	值小于或等于 0。	
<b>16#840D</b>		<b>运动控制指令的“RampDownTime”参数值无效</b>	
	16#0002	值不是有效数字	更正该值；重新启动命令
	16#000A	值小于或等于 0。	
<b>16#840E</b>		<b>运动控制指令的“EmergencyRampTime”参数值无效</b>	
	16#0002	值不是有效数字	更正该值；重新启动命令
	16#000A	值小于或等于 0。	
<b>16#840F</b>		<b>运动控制指令的“JerkTime”参数值无效</b>	
	16#0002	值不是有效数字	更正该值；重新启动命令
	16#000A	值小于或等于 0。	



ErrorID	ErrorInfo	说明	解决方法
<b>16#8410</b>		<b>运动控制指令的“Parameter”参数值无效</b>	
	16#0002	值不是有效数字	更正该值；重新启动命令
	16#000B	地址无效	将无错误组态下载到控制器；使用指令“MC_Power”再次启用轴
	16#0028	VARIANT 指针“参数”和“值”的数据类型不匹配。	使用适当的数据类型；重新启动命令。
	16#0029	VARIANT 指针“参数”并未指向工艺对象的数据块。	更正 VARIANT 指针；重新启动命令
	16#002A	无法读取 VARIANT 指针“参数”值。	更正 VARIANT 指针；重新启动命令
	16#002B	无法写入 VARIANT 指针“参数”值。	更正 VARIANT 指针或值；重新启动命令
	16#002C	轴未禁用。	禁用轴；重新启动命令
<b>16#8411</b>		<b>运动控制指令的“Value”参数值无效</b>	
	16#0002	值不是有效数字	更正该值；重新启动命令

## 轴的组态错误

ErrorID	ErrorInfo	说明	解决方法
<b>16#8600</b>		<b>脉冲发生器 (PTO) 的参数分配无效</b>	
	16#000B	地址无效	将无错误组态下载到控制器；使用指令“MC_Power”再次启用轴
	16#0014	所选硬件由另一应用程序使用	
<b>16#8601</b>		<b>高速计数器 (HSC) 的参数分配无效</b>	
	16#000B	地址无效	将无错误组态下载到控制器；使用指令“MC_Power”再次启用轴
	16#0014	所选硬件由另一应用程序使用	
<b>16#8602</b>		<b>“使能输出”(Enable output) 的参数分配无效</b>	
	16#000B	地址无效	将无错误组态下载到控制器；使用指令“MC_Power”再次启用轴
<b>16#8603</b>		<b>“输入就绪”(Ready input) 的参数分配无效</b>	
	16#000B	地址无效	将无错误组态下载到控制器；使用指令“MC_Power”再次启用轴
<b>16#8604</b>		<b>“电机每转的脉冲数”(Pulses per motor revolution) 值无效</b>	
	16#000A	值小于或等于零	将无错误组态下载到控制器；使用指令“MC_Power”再次启用轴
<b>16#8605</b>		<b>“每转的距离”(Distance per revolution) 值无效</b>	
	16#0002	值不是有效数字	将无错误组态下载到控制器；使用指令“MC_Power”再次启用轴
	16#0005	该值超出值范围（大于 1E+12）	
	16#000A	值小于或等于零	
	16#0030	值的数字格式不正确，或者超出有效的数字范围	<ul style="list-style-type: none"> <li>将无错误组态下载到控制器；使用指令“MC_Power”再次启用轴</li> <li>在线更正不正确的值；使用指令“MC_Reset”确认错误并视需要重新启动命令</li> </ul>
<b>16#8606</b>		<b>“启动/停止速度”(Start / stop velocity) 值无效</b>	
	16#0002	值不是有效数字	将无错误组态下载到控制器；使用指令“MC_Power”再次启用轴
	16#0003	值高于硬件上限值	
	16#0004	值小于硬件下限值	
	16#0007	启动/停止速度大于最大速度	

ErrorID	ErrorInfo	说明	解决方法
<b>16#8607</b>		<b>“最大速度”(maximum velocity) 值无效</b>	
	16#0002	值不是有效数字	将无错误组态下载到控制器； 使用指令“MC_Power”再次启用轴
	16#0003	值高于硬件上限值	
	16#0004	值小于硬件下限值	
	16#0030	值的数字格式不正确， 或者超出有效的数字范围	<ul style="list-style-type: none"> <li>将无错误组态下载到控制器； 使用指令“MC_Power”再次启用轴</li> <li>在线更正不正确的值；使用指令“MC_Reset”确认错误并视需要重新启动命令</li> </ul>
<b>16#8608</b>		<b>“加速度”(Acceleration) 值无效</b>	
	16#0002	值不是有效数字	<ul style="list-style-type: none"> <li>将无错误组态下载到控制器； 使用指令“MC_Power”再次启用轴</li> <li>在线更正不正确的值；使用指令“MC_Reset”确认错误并视需要重新启动命令</li> </ul>
	16#0003	值高于硬件上限值	
	16#0004	值小于硬件下限值	
	16#0030	值的数字格式不正确， 或者超出有效的数字范围	<ul style="list-style-type: none"> <li>将无错误组态下载到控制器； 使用指令“MC_Power”再次启用轴</li> <li>在线更正不正确的值；使用指令“MC_Reset”确认错误并视需要重新启动命令</li> </ul>
<b>16#8609</b>		<b>“减速度”(Deceleration) 值无效</b>	
	16#0002	值不是有效数字	<ul style="list-style-type: none"> <li>将无错误组态下载到控制器；使用指令“MC_Power”再次启用轴</li> <li>在线更正不正确的值；使用指令“MC_Reset”确认错误并视需要重新启动命令</li> </ul>
	16#0003	值高于硬件上限值	
	16#0004	值小于硬件下限值	
	16#0030	值的数字格式不正确， 或者超出有效的数字范围	<ul style="list-style-type: none"> <li>将无错误组态下载到控制器； 使用指令“MC_Power”再次启用轴</li> <li>在线更正不正确的值；使用指令“MC_Reset”确认错误并视需要重新启动命令</li> </ul>

ErrorID	ErrorInfo	说明	解决方法
<b>16#860A</b>		<b>“急停减速度”(Emergency stop deceleration) 值无效</b>	
	16#0002	值不是有效数字	<ul style="list-style-type: none"> <li>将无错误组态下载到控制器；使用指令“MC_Power”再次启用轴</li> </ul>
	16#0003	值高于硬件上限值	
	16#0004	值小于硬件下限值	<ul style="list-style-type: none"> <li>在线更正不正确的值；使用指令“MC_Reset”确认错误并视需要重新启动命令</li> </ul>
	16#0030	值的数字格式不正确，或者超出有效的数字范围	<ul style="list-style-type: none"> <li>将无错误组态下载到控制器；使用指令“MC_Power”再次启用轴</li> <li>在线更正不正确的值；使用指令“MC_Reset”确认错误并视需要重新启动命令</li> </ul>
<b>16#860B</b>		<b>下限软限位开关的位置值无效</b>	
	16#0002	值不是有效数字	<ul style="list-style-type: none"> <li>将无错误组态下载到控制器；使用指令“MC_Power”再次启用轴</li> </ul>
	16#0005	该值超出值范围（大于 1E+12）	
	16#0006	该值超出值范围（小于 1E+12）	<ul style="list-style-type: none"> <li>在线更正不正确的值；使用指令“MC_Reset”确认错误并视需要重新启动命令</li> </ul>
	16#0030	下限软限位开关的位置值大于上限软限位开关的位置值	
<b>16#860C</b>		<b>上限软限位开关的位置值无效</b>	
	16#0002	值不是有效数字	<ul style="list-style-type: none"> <li>将无错误组态下载到控制器；使用指令“MC_Power”再次启用轴</li> </ul>
	16#0005	该值超出值范围（大于 1E+12）	
	16#0006	该值超出值范围（小于 1E+12）	<ul style="list-style-type: none"> <li>在线更正不正确的值；使用指令“MC_Reset”确认错误并视需要重新启动命令</li> </ul>
	16#0030	值的数字格式不正确，或者超出有效的数字范围	
<b>16#860D</b>		<b>下限硬限位开关的地址无效</b>	
	16#000B	地址无效	将无错误组态下载到控制器；使用指令“MC_Power”再次启用轴
	16#000C	下降沿的地址无效	
	16#000D	上升沿的地址无效	

ErrorID	ErrorInfo	说明	解决方法
<b>16#860E</b>		<b>上限硬限位开关的地址无效</b>	
	16#000B	地址无效	将无错误组态下载到控制器； 使用指令“MC_Power”再次启用轴
	16#000C	下降沿的地址无效	
	16#000D	上升沿的地址无效	
<b>16#860F</b>		<b>“起始位置偏移值”(home position offset) 无效</b>	
	16#0002	值不是有效数字	<ul style="list-style-type: none"> <li>将无错误组态下载到控制器； 使用指令“MC_Power”再次启用轴</li> <li>在线更正不正确的值；使用指令“MC_Reset”确认错误并视需要重新启动命令</li> </ul>
	16#0005	该值超出值范围（大于 1E+12）	
	16#0006	该值超出值范围（小于 1E+12）	
	16#0030	值的数字格式不正确， 或者超出有效的数字范围	<ul style="list-style-type: none"> <li>将无错误组态下载到控制器； 使用指令“MC_Power”再次启用轴</li> <li>在线更正不正确的值；使用指令“MC_Reset”确认错误并视需要重新启动命令</li> </ul>
<b>16#8610</b>		<b>“逼近速度”(approach velocity) 值无效</b>	
	16#0002	值不是有效数字	<ul style="list-style-type: none"> <li>将无错误组态下载到控制器； 使用指令“MC_Power”再次启用轴</li> <li>在线更正不正确的值；使用指令“MC_Reset”确认错误并视需要重新启动命令</li> </ul>
	16#0008	速度大于最大速度	
	16#0009	速度小于启动/停止速度	
	16#0030	值的数字格式不正确， 或者超出有效的数字范围	<ul style="list-style-type: none"> <li>将无错误组态下载到控制器； 使用指令“MC_Power”再次启用轴</li> <li>在线更正不正确的值；使用指令“MC_Reset”确认错误并视需要重新启动命令</li> </ul>

ErrorID	ErrorInfo	说明	解决方法
<b>16#8611</b>		<b>“归位速度”(Homing velocity) 值无效</b>	
	16#0002	值不是有效数字	<ul style="list-style-type: none"> <li>将无错误组态下载到控制器；使用指令“MC_Power”再次启用轴</li> <li>在线更正不正确的值；使用指令“MC_Reset”确认错误并视需要重新启动命令</li> </ul>
	16#0008	速度大于最大速度	
	16#0009	速度小于启动/停止速度	
	16#0030	值的数字格式不正确，或者超出有效的数字范围	
<b>16#8612</b>		<b>归位开关的地址无效</b>	
	16#000B	地址无效	将无错误组态下载到控制器； 使用指令“MC_Power”再次启用轴
	16#000C	下降沿的地址无效	
	16#000D	上升沿的地址无效	
<b>16#8613</b>		<b>虽然未组态硬限位开关，但还是在主动归位过程中，激活了在该硬限位开关处反向</b>	
	16#0001	-	<ul style="list-style-type: none"> <li>将无错误组态下载到控制器；使用指令“MC_Power”再次启用轴</li> <li>在线更正不正确的值；使用指令“MC_Reset”确认错误并视需要重新启动命令</li> </ul>
<b>16#8614</b>		<b>“冲击”(Jerk) 值无效</b>	
	16#0002	值不是有效数字	<ul style="list-style-type: none"> <li>将无错误组态下载到控制器；使用指令“MC_Power”再次启用轴</li> <li>在线更正不正确的值；使用指令“MC_Reset”确认错误并视需要重新启动命令</li> </ul>
	16#001F	值大于最大冲击值	
	16#0020	值小于最小冲击值	
	16#0030	值的数字格式不正确，或者超出有效的数字范围	<ul style="list-style-type: none"> <li>将无错误组态下载到控制器；使用指令“MC_Power”再次启用轴</li> <li>在线更正不正确的值；使用指令“MC_Reset”确认错误并视需要重新启动命令</li> </ul>

ErrorID	ErrorInfo	说明	解决方法
<b>16#8615</b>		<b>“测量单位”的值无效</b>	
	16#0011	所选值无效	将无错误组态下载到控制器； 使用指令“MC_Power”再次启用轴
<b>16#8616</b>		<b>归位开关的地址无效（自版本 V4 起的被动归位）</b>	
	16#0011	所选值无效	<ul style="list-style-type: none"> <li>将无错误组态下载到控制器； 使用指令“MC_Power”再次启用轴</li> <li>在线更正不正确的值；使用指令“MC_Reset”确认错误并视需要重新启动命令</li> </ul>
	16#0030	值的数字格式不正确，或者超出有效的数字范围	<ul style="list-style-type: none"> <li>将无错误组态下载到控制器； 使用指令“MC_Power”再次启用轴</li> <li>在线更正不正确的值；使用指令“MC_Reset”确认错误并视需要重新启动命令</li> </ul>
<b>16#8617</b>		<b>变量 &lt;轴名称&gt;.Sensor.Sensor[1].ActiveHoming.Mode 的值无效</b>	
	16#0011	所选值无效 (有效值： 2 = 通过数字量输入归位)	<ul style="list-style-type: none"> <li>将无错误组态下载到控制器； 使用指令“MC_Power”再次启用轴</li> <li>在线更正不正确的值；使用指令“MC_Reset”确认错误并视需要重新启动命令</li> </ul>
<b>16#8618</b>		<b>变量 &lt;轴名称&gt;.Sensor.Sensor[1].PassiveHoming.Mode 的值无效</b>	
	16#0011	所选值无效 (有效值： 2 = 通过数字量输入归位)	<ul style="list-style-type: none"> <li>将无错误组态下载到控制器； 使用指令“MC_Power”再次启用轴</li> <li>在线更正不正确的值；使用指令“MC_Reset”确认错误并视需要重新启动命令</li> </ul>
<b>16#8619</b>		<b>变量 &lt;轴名称&gt;.Actor.Type 的值无效</b>	
	16#0011	所选值无效 (有效值： 2 = 通过脉冲接口连接)	<ul style="list-style-type: none"> <li>将无错误组态下载到控制器； 使用指令“MC_Power”再次启用轴</li> <li>在线更正不正确的值；使用指令“MC_Reset”确认错误并视需要重新启动命令</li> </ul>

ErrorID	ErrorInfo	说明	解决方法
<b>16#861A</b>		<b>“允许的旋转方向”(Permitted direction of rotation) 值无效</b>	
	16#0011	所选值无效	<ul style="list-style-type: none"> <li>将无错误组态下载到控制器；使用指令“MC_Power”再次启用轴</li> <li>在线更正不正确的值；使用指令“MC_Reset”确认错误并视需要重新启动命令</li> </ul>
	16#002D	禁用方向输出时不允许“双向”(Both directions) 运动	
<b>16#861B</b>		<b>负载齿轮因数错误</b>	
	16#0031	值无效。	将无错误组态下载到控制器；使用指令“MC_Power”再次启用轴
<b>16#861C</b>		<b>通过增量编码器归位的数据组合非法</b>	
	16#0031	值无效。	<ul style="list-style-type: none"> <li>将无错误组态下载到控制器；使用指令“MC_Power”再次启用轴</li> <li>在线更正不正确的值；使用指令“MC_Reset”确认错误并视需要重新启动命令</li> </ul>
<b>16#861D</b>		<b>设置的编码器安装方式无效。 &lt;轴名称&gt;.Sensor.Sensor[1].MountingMode 中的值无效</b>	
	16#0011	所选值无效	<ul style="list-style-type: none"> <li>将无错误组态下载到控制器；使用指令“MC_Power”再次启用轴</li> <li>在线更正不正确的值；使用指令“MC_Reset”确认错误并视需要重新启动命令</li> </ul>
<b>16#861E</b>		<b>测量编码器轮周长的组态无效。 &lt;轴名称&gt;.Sensor.Sensor[1].Parameter.DistancePerRevolution 中的值无效</b>	
	16#0030	值的数字格式不正确，或者超出有效的数字范围	<ul style="list-style-type: none"> <li>将无错误组态下载到控制器；使用指令“MC_Power”再次启用轴</li> <li>在线更正不正确的值；使用指令“MC_Reset”确认错误并视需要重新启动命令</li> </ul>



ErrorID	ErrorInfo	说明	解决方法
16#861F		组态的线性编码器精度错误。 <轴名称>.Sensor.Sensor[1].Parameter.Resolution 中的值无效	
	16#0030	值的数字格式不正确， 或者超出有效的数字范围	<ul style="list-style-type: none"> <li>将无错误组态下载到控制器； 使用指令“MC_Power”再次启用轴</li> <li>在线更正不正确的值；使用指令 “MC_Reset”确认错误并视需要重新 启动命令</li> </ul>
16#8620		设置的 Gn_XIST1 高精度值无效。 <轴名称>.Sensor.Sensor[1].Parameter.FineResolutionXist1 中的值无效	
	16#0030	值的数字格式不正确， 或者超出有效的数字范围	<ul style="list-style-type: none"> <li>将无错误组态下载到控制器； 使用指令“MC_Power”再次启用轴</li> <li>在线更正不正确的值；使用指令 “MC_Reset”确认错误并视需要重新 启动命令</li> </ul>
16#8621		在 <轴名称>.Sensor.Sensor[1].Parameter.FineResolutionXist1 中设置的 Gn_XIST1 高精度值与 PROFIdrive 参数 P979 中的不一致	
	16#0030	值的数字格式不正确， 或者超出有效的数字范围	<ul style="list-style-type: none"> <li>将无错误组态下载到控制器； 使用指令“MC_Power”再次启用轴</li> <li>在线更正不正确的值；使用指令 “MC_Reset”确认错误并视需要重新 启动命令</li> </ul>
16#8622		组态日期 <轴名称>.Actor.Interface.AddressIn 或 <轴名称>.Actor.Interface.AddressOut 的值无效	
	16#0011	所选值无效	将无错误组态下载到控制器； 使用指令“MC_Power”再次启用轴
16#8623		变量 <轴名称>.Sensor.Sensor[1].Type 中设置的值无效。	
	16#0011	所选值无效	将无错误组态下载到控制器； 使用指令“MC_Power”再次启用轴

ErrorID	ErrorInfo	说明	解决方法
16#8624		设置的编码器系统无效。 <轴名称>.Sensor.Sensor[1].System 中的值无效	
	16#0011	所选值无效	<ul style="list-style-type: none"> <li>将无错误组态下载到控制器；使用指令“MC_Power”再次启用轴</li> <li>在线更正不正确的值；使用指令“MC_Reset”确认错误并视需要重新启动命令</li> </ul>
16#8625		定位监控的参数错误。 <轴名称>.PositioningMonitoring.MinDwellTime 中的值无效	
	16#0030	值的数字格式不正确， 或者超出有效的数字范围	<ul style="list-style-type: none"> <li>将无错误组态下载到控制器；使用指令“MC_Power”再次启用轴</li> <li>在线更正不正确的值；使用指令“MC_Reset”确认错误并视需要重新启动命令</li> </ul>
16#8626		定位监控的参数错误。 <轴名称>.PositioningMonitoring.Window 中的值无效	
	16#0030	值的数字格式不正确， 或者超出有效的数字范围	<ul style="list-style-type: none"> <li>将无错误组态下载到控制器；使用指令“MC_Power”再次启用轴</li> <li>在线更正不正确的值；使用指令“MC_Reset”确认错误并视需要重新启动命令</li> </ul>
16#8627		PROFIdrive 接口组态的实际值错误。 <轴名称>.Sensor.Sensor[1].Interface.AddressIn 或 <轴名称>.Sensor.Sensor[1].Interface.AddressOut 中的值无效	
	16#0011	所选值无效	将无错误组态下载到控制器； 使用指令“MC_Power”再次启用轴
16#8628		控制器数错误	
	16#0030	值的数字格式不正确，或者超出有效的 数字范围	<p>控制回路的增益或预控制值错误。</p> <ul style="list-style-type: none"> <li>将无错误组态下载到控制器；使用指令“MC_Power”再次启用轴</li> <li>在线更正不正确的值；使用指令“MC_Reset”确认错误并视需要重新启动命令 ( &lt;轴名称&gt;.PositionControl.Kv、 &lt;轴名称&gt;.PositionControl.Kpc )</li> </ul>

ErrorID	ErrorInfo	说明	解决方法
<b>16#8629</b>		停止信号的限值错误。 <轴名称>.StandStillSignal.VelocityThreshold 中的值无效	
	16#0030	值的数字格式不正确， 或者超出有效的数字范围	<ul style="list-style-type: none"> <li>将无错误组态下载到控制器； 使用指令“MC_Power”再次启用轴</li> <li>在线更正不正确的值；使用指令 “MC_Reset”确认错误并视需要重新 启动命令</li> </ul>
<b>16#862A</b>		定位监控的参数错误。 <轴名称>.PositioningMonitoring.ToleranceTime 中的值无效	
	16#0030	值的数字格式不正确， 或者超出有效的数字范围	<ul style="list-style-type: none"> <li>将无错误组态下载到控制器； 使用指令“MC_Power”再次启用轴</li> <li>在线更正不正确的值；使用指令 “MC_Reset”确认错误并视需要重新 启动命令</li> </ul>
<b>16#862B</b>		PROFIBUS 参数化不一致；Ti 和 To 的总和大于一个 DP 周期	
	16#0030	值的数字格式不正确， 或者超出有效的数字范围	将无错误组态下载到控制器； 使用指令“MC_Power”再次启用轴
<b>16#862C</b>		停止监视的参数错误。 <轴名称>.StandStillSignal.MinDwellTime 中的值无效	
	16#0030	值的数字格式不正确， 或者超出有效的数字范围	<ul style="list-style-type: none"> <li>将无错误组态下载到控制器； 使用指令“MC_Power”再次启用轴</li> <li>在线更正不正确的值；使用指令 “MC_Reset”确认错误并视需要重新 启动命令</li> </ul>
<b>16#862D</b>		跟随误差监控的参数错误。 <轴名称>.FollowingError.MinValue 中的值无效	
	16#0030	值的数字格式不正确， 或者超出有效的数字范围	<ul style="list-style-type: none"> <li>将无错误组态下载到控制器； 使用指令“MC_Power”再次启用轴</li> <li>在线更正不正确的值；使用指令 “MC_Reset”确认错误并视需要重新 启动命令</li> </ul>

ErrorID	ErrorInfo	说明	解决方法
<b>16#862E</b>		<b>&lt;轴名称&gt;. 中组态日期无效Modulo.Length</b>	
	16#0030	值的数字格式不正确， 或者超出有效的数字范围	<ul style="list-style-type: none"> <li>将无错误组态下载到控制器； 使用指令“MC_Power”再次启用轴</li> <li>在线更正不正确的值；使用指令 “MC_Reset”确认错误并视需要重新 启动命令</li> </ul>
<b>16#862F</b>		<b>&lt;轴名称&gt;. 中组态日期无效Modulo.StartValue</b>	
	16#0030	值的数字格式不正确， 或者超出有效的数字范围	<ul style="list-style-type: none"> <li>将无错误组态下载到控制器； 使用指令“MC_Power”再次启用轴</li> <li>在线更正不正确的值；使用指令 “MC_Reset”确认错误并视需要重新 启动命令</li> </ul>
<b>16#8630</b>		<b>&lt;轴名称&gt;. 中组态日期无效Actor.DriveParameter.ReferenceSpeed</b>	
	16#0030	值的数字格式不正确， 或者超出有效的数字范围	<ul style="list-style-type: none"> <li>将无错误组态下载到控制器； 使用指令“MC_Power”再次启用轴</li> <li>在线更正不正确的值；使用指令 “MC_Reset”确认错误并视需要重新 启动命令</li> </ul>
<b>16#8631</b>		<b>设置的 Gn_XIST2 高精度值无效。 &lt;轴名称&gt;.Sensor.Sensor[1].Parameter.FineResolutionXist2 中的值无效</b>	
	16#0030	值的数字格式不正确， 或者超出有效的数字范围	<ul style="list-style-type: none"> <li>将无错误组态下载到控制器； 使用指令“MC_Power”再次启用轴</li> <li>在线更正不正确的值；使用指令 “MC_Reset”确认错误并视需要重新 启动命令</li> </ul>
<b>16#8632</b>		<b>可确定的编码器精度值无效。 &lt;轴名称&gt;.Sensor.Sensor[1].Parameter.DeterminableRevolutions 中的值无效</b>	
	16#0030	值的数字格式不正确， 或者超出有效的数字范围	<ul style="list-style-type: none"> <li>将无错误组态下载到控制器； 使用指令“MC_Power”再次启用轴</li> <li>在线更正不正确的值；使用指令 “MC_Reset”确认错误并视需要重新 启动命令</li> </ul>

ErrorID	ErrorInfo	说明	解决方法
16#8633		指定的被动归位的归位开关逼近方向无效。 <轴名称>.Sensor.Sensor[1].PassiveHoming.Direction 中的值无效	
16#8634		跟随误差监控的参数错误。 <轴名称>.FollowingError.MaxValue 中的值无效	
	16#0030	值的数字格式不正确， 或者超出有效的数字范围	<ul style="list-style-type: none"> <li>将无错误组态下载到控制器； 使用指令“MC_Power”再次启用轴</li> <li>在线更正不正确的值；使用指令 “MC_Reset”确认错误并视需要重新 启动命令</li> </ul>
16#8635		跟随误差监控的参数错误。 <轴名称>.FollowingError.MinVelocity 中的值无效	
	16#0030	值的数字格式不正确， 或者超出有效的数字范围	<ul style="list-style-type: none"> <li>将无错误组态下载到控制器； 使用指令“MC_Power”再次启用轴</li> <li>在线更正不正确的值；使用指令 “MC_Reset”确认错误并视需要重新 启动命令</li> </ul>
16#8636		控制器因数不正确。 <轴名称>.PositionControl.Kpc 中预控制因数的值无效	
	16#0030	值的数字格式不正确， 或者超出有效的数字范围	<ul style="list-style-type: none"> <li>将无错误组态下载到控制器； 使用指令“MC_Power”再次启用轴</li> <li>在线更正不正确的值；使用指令 “MC_Reset”确认错误并视需要重新 启动命令</li> </ul>
16#8637		<轴名称>. 中组态日期无效Sensor.Sensor[1].Interface.Type	
	16#0011	所选值无效	将无错误组态下载到控制器； 使用指令“MC_Power”再次启用轴
16#8638		<轴名称>. 中组态日期无效Sensor.Sensor[1].Interface.HSC	
	16#0011	所选值无效	将无错误组态下载到控制器； 使用指令“MC_Power”再次启用轴

ErrorID	ErrorInfo	说明	解决方法
<b>16#8639</b>		<b>驱动器上有错误</b>	
	16#0049	设备的组态错误	将无错误组态下载到控制器； 使用指令“MC_Power”再次启用轴
	16#004A	工艺需要一个更小的伺服时钟。	内部系统错误。 检查项目的一致性并将项目重新加载到控制器中。
	16#004B	斜坡上升过程中未初始化设备驱动程序。	要启用工艺对象， 必须完成执行器驱动程序的初始化。 请稍后再执行该命令。
<b>16#863A</b>		<b>与驱动器的通信错误</b>	
	16#004C	设备的组态错误	<ul style="list-style-type: none"> <li>• 连接合适的设备。</li> <li>• 检查设备 (I/O)。</li> <li>• 比较 HW Config 中的组态和工艺对象。</li> </ul>
	16#004D	设备驱动程序需要一个更小的伺服时钟。	<ul style="list-style-type: none"> <li>• 连接合适的设备。</li> <li>• 检查设备 (I/O)。</li> <li>• 比较 HW Config 中的组态和工艺对象。</li> </ul>
	16#004E	与设备进行内部通信时发生错误	检查项目的一致性并将项目重新加载到控制器中。
<b>16#863B</b>		<b>编码器中的错误</b>	
	16#0049	设备的组态错误	将无错误组态下载到控制器； 使用指令“MC_Power”再次启用轴
	16#004A	工艺需要一个更小的伺服时钟。	内部系统错误。 检查项目的一致性并将项目重新加载到控制器中。
	16#004B	斜坡上升过程中未初始化设备驱动程序。	要启用工艺对象，必须完成执行器驱动程序的初始化。 请稍后再执行该命令。

ErrorID	ErrorInfo	说明	解决方法
<b>16#863C</b>		<b>与编码器的通信错误</b>	
	16#004C	设备的组态错误	<ul style="list-style-type: none"> <li>• 连接合适的设备。</li> <li>• 检查设备 (I/O)。</li> <li>• 比较 HW Config 中的组态和工艺对象。</li> </ul>
	16#004D	设备驱动程序需要一个更小的伺服时钟。	<ul style="list-style-type: none"> <li>• 连接合适的设备。</li> <li>• 检查设备 (I/O)。</li> <li>• 比较 HW Config 中的组态和工艺对象。</li> </ul>
	16#004E	与设备进行内部通信时发生错误	检查项目的一致性并将项目重新加载到控制器中。
<b>16#863D</b>		<b>与设备（驱动器或编码器）的通信错误</b>	
	16#0030	值的数字格式不正确，或者超出有效的数字范围	<ul style="list-style-type: none"> <li>• 将无错误组态下载到控制器；使用指令“MC_Power”再次启用轴</li> <li>• 在线更正不正确的值；使用指令“MC_Reset”确认错误并视需要重新启动命令</li> </ul>
	16#0055	请求的逻辑地址无效。	<ul style="list-style-type: none"> <li>• 连接合适的设备。</li> </ul>
	16#0056	请求的逻辑输出地址无效。	<ul style="list-style-type: none"> <li>• 检查设备 (I/O)。</li> </ul>
	16#0057	请求的逻辑输出地址无效。	<ul style="list-style-type: none"> <li>• 检查 HW Config 中的拓扑组态。</li> <li>• 比较 HW Config 中的组态和工艺对象。</li> </ul>
<b>16#863E</b>		<b>变量“ControlPanel.Input.TimeOut”的值无效（轴控制面板）</b>	
	16#0030	值的数字格式不正确，或者超出有效的数字范围	更正工艺对象 <轴名称>.ControlPanel.Input.TimeOut 的变量值。 以毫秒为单位指定该值。

ErrorID	ErrorInfo	说明	解决方法
<b>16#863F</b>		<b>&lt;轴名称&gt;. 中组态日期无效Actor.DriveParameter.MaxSpeed</b>	
	16#0030	值的数字格式不正确，或者超出有效的数字范围	<p>将驱动器和工艺对象组态中的参考值更正为 <code>Actuator.MaxSpeed/2</code>。</p> <p>对于模拟驱动器接口，将驱动器和工艺对象组态中的参考值更正为 <code>Actuator.MaxSpeed/1.17</code>。</p>



## 命令表的组态错误

ErrorID	ErrorInfo	说明	解决方法
<b>16#8700</b>		<b>命令表中“命令类型”(Command type) 的值无效</b>	
	16#0001	-	<ul style="list-style-type: none"> <li>将无错误组态下载到控制器；使用指令“MC_Power”再次启用轴</li> <li>在线更正不正确的值，并视需要重新启动命令</li> </ul>
<b>16#8701</b>		<b>命令表中“位置/行进路径”(Position/travel path) 的值无效</b>	
	16#0002	值不是有效数字	<ul style="list-style-type: none"> <li>将无错误组态下载到控制器；使用指令“MC_Power”再次启用轴</li> <li>在线更正不正确的值，并视需要重新启动命令</li> </ul>
	16#0005	该值超出值范围（大于 1E+12）	
	16#0006	该值超出值范围（小于 1E+12）	
<b>16#8702</b>		<b>命令表中“速度”(Velocity) 的值无效</b>	
	16#0002	值不是有效数字	<ul style="list-style-type: none"> <li>将无错误组态下载到控制器；使用指令“MC_Power”再次启用轴</li> <li>在线更正不正确的值，并视需要重新启动命令</li> </ul>
	16#0008	值大于组态的最大速度	
	16#0009	值小于组态的启动/停止速度	
<b>16#8703</b>		<b>命令表中“持续时间”(Duration) 的值无效</b>	
	16#0002	值不是有效数字	<ul style="list-style-type: none"> <li>将无错误组态下载到控制器；使用指令“MC_Power”再次启用轴</li> <li>在线更正不正确的值，并视需要重新启动命令</li> </ul>
	16#0021	值大于 64800 s	
	16#0022	值小于 0.001 s	
<b>16#8704</b>		<b>命令表中“下一步”(Next step) 的值无效</b>	
	16#0011	所选值无效	<ul style="list-style-type: none"> <li>将无错误组态下载到控制器；使用指令“MC_Power”再次启用轴</li> <li>在线更正不正确的值，并视需要重新启动命令</li> </ul>
	16#0023	该命令不允许命令转换	

内部错误

ErrorID	ErrorInfo	说明	解决方法
16#8FFF		内部错误	
	16#F0**	-	CPU 的断电和上电 如果这样做无效，则联系客户支持。 准备好以下信息： <ul style="list-style-type: none"> <li>• ErrorID</li> <li>• ErrorInfo</li> <li>• 诊断缓冲区条目</li> </ul>

参见

自 V4 起的定位轴工艺对象的变量 (页 235)

ErrorID 和 ErrorInfo (页 320)

运动控制语句的错误显示 (页 181)

将多个轴与同一个 PTO 一起使用 (页 190)

将多个驱动器与同一个 PTO 一起使用 (页 193)

在更高优先级等级（执行级别）中跟踪作业 (页 194)

使用通过 PTO 的驱动器连接时的软限位开关时的特殊情况 (页 197)

## 1.11.8 自 V4 起的定位轴工艺对象的变量

### 1.11.8.1 自 V4 起的位置变量

#### 图例

<b>数据类型</b>	变量的数据类型	
<b>起始值</b>	变量的起始值	
<b>访问</b>	在用户程序中访问变量的方式:	
	RCCP	可在用户程序中读取变量并在每个周期控制点处更新变量。
	RP	可使用运动控制指令“MC_ReadParam”读取该变量。 相应变量的当前值在命令的起始处决定。
<b>有效</b>	指定变量的更改何时生效。	
<b>HMI</b>	该变量可用于 HMI 系统（符合用户程序中的变量访问）。	

<b>&lt;轴名称&gt;.Position</b>				
轴的位置设定值 (用组态的测量单位表示) 如果轴未归位, 则该变量指示的是相对于轴启用位置的位置值。				
<b>数据类型</b>	<b>起始值</b>	<b>访问</b>	<b>有效</b>	<b>HMI</b>
Real	0.0	RCCP、RP	-	√

#### 参见

运动状态 (页 187)

轴工艺对象 V1...3 的变量 (页 342)

1.11.8.2 自 V4 起的速度变量

图例

<b>数据类型</b>	变量的数据类型	
<b>起始值</b>	变量的起始值	
<b>访问</b>	在用户程序中访问变量的方式：	
	RCCP	可在用户程序中读取变量并在每个周期控制点处更新变量。
	RP	可使用运动控制指令“MC_ReadParam”读取该变量。 相应变量的当前值在命令的起始处决定。
<b>有效</b>	指定变量的更改何时生效。	
<b>HMI</b>	该变量可用于 HMI 系统（符合用户程序中的变量访问）。	

<b>&lt;轴名称&gt;.Velocity</b>				
轴的速度设定值 (用组态的测量单位表示)				
<b>数据类型</b>	<b>起始值</b>	<b>访问</b>	<b>有效</b>	<b>HMI</b>
Real	0.0	RCCP、RP	-	√

参见

运动状态 (页 187)

轴工艺对象 V1...3 的变量 (页 342)

## 1.11.8.3 自 V5 起的 ActualPosition 变量

## 图例

<b>数据类型</b>	变量的数据类型	
<b>起始值</b>	变量的起始值	
<b>访问</b>	在用户程序中访问变量的方式:	
	RCCP	可在用户程序中读取变量并在每个周期控制点处更新变量。
	RP	可使用运动控制指令“MC_ReadParam”读取该变量。 相应变量的当前值在命令的起始处决定。
<b>有效</b>	指定变量的更改何时生效。	
<b>HMI</b>	该变量可用于 HMI 系统（符合用户程序中的变量访问）。	

<b>&lt;轴名称&gt;.ActualPosition</b>				
轴的实际位置（用组态的测量单位表示） 如果轴未归位，则该变量指示的是相对于轴启用位置的位置值。				
<b>数据类型</b>	<b>起始值</b>	<b>访问</b>	<b>有效</b>	<b>HMI</b>
Real	0.0	RCCP、RP	-	√

### 1.11.8.4 自 V5 起的 ActualVelocity 变量

#### 图例

<b>数据类型</b>	变量的数据类型	
<b>起始值</b>	变量的起始值	
<b>访问</b>	在用户程序中访问变量的方式：	
	RCCP	可在用户程序中读取变量并在每个周期控制点处更新变量。
	RP	可使用运动控制指令“MC_ReadParam”读取该变量。 相应变量的当前值在命令的起始处决定。
<b>有效</b>	指定变量的更改何时生效。	
<b>HMI</b>	该变量可用于 HMI 系统（符合用户程序中的变量访问）。	

<b>&lt;轴名称&gt;.ActualVelocity</b>				
轴的实际速度 (用组态的测量单位表示)				
<b>数据类型</b>	<b>起始值</b>	<b>访问</b>	<b>有效</b>	<b>HMI</b>
Real	0.0	RCCP、RP	-	√

## 1.11.8.5 自 V4 起的 Actor 变量

## 图例

<b>数据类型</b>	变量的数据类型	
<b>起始值</b>	变量的起始值	
<b>访问</b>	在用户程序中访问变量的方式:	
	R	可以在用户程序中读取该变量。
	WP	如果该轴已禁用 (MC_Power.Status = FALSE), 可使用运动控制指令“MC_WriteParam”对该变量进行写操作。
	-	不能在用户程序中使用该变量。
<b>有效</b>	指定变量的更改何时生效:	
	2	通过 PTO (Pulse Train Output) 的驱动器连接: 轴已启用时
	9	通过 PROFIdrive/模拟量输出的驱动器连接: 在启用轴的情况下重新启动工艺对象后
<b>HMI</b>	该变量可用于 HMI 系统 (符合用户程序中的变量访问)。	

<b>&lt;轴名称&gt;.Actor.Type</b>				
驱动器连接				
<ul style="list-style-type: none"> <li>定位轴工艺对象 V5 及更高版本: <ul style="list-style-type: none"> <li>0 = 驱动器通过模拟量输出连接。轴的所有运动均在位置控制下进行。</li> <li>1 = 驱动器通过 PROFIdrive 报文连接。轴的所有运动均在位置控制下进行。</li> <li>2 = 驱动器通过脉冲接口连接。</li> </ul> </li> <li>定位轴工艺对象 V4: <ul style="list-style-type: none"> <li>驱动器通过脉冲接口连接。</li> </ul> </li> </ul>				
数据类型	起始值	访问	有效	HMI
DInt	2	R	-	√

<b>&lt;轴名称&gt;.Actor.InverseDirection</b>				
反转方向 FALSE = 方向未反转。 TRUE = 方向已反转。				
数据类型	起始值	访问	有效	HMI
Bool	FALSE	R	-	√
		WP	2, 9	-

<b>&lt;轴名称&gt;.Actor.DirectionMode</b>				
允许的旋转方向 0 = 双向 1 = 正向 2 = 负向				
数据类型	起始值	访问	有效	HMI
Int	0	R	-	√
		WP	2, 9	-

<b>&lt;轴名称&gt;.Actor.Interface.AddressIn (定位轴工艺对象 V5 及更高版本)</b>				
PROFIdrive 报文的输入地址 (内部参数)				
数据类型	起始值	访问	有效	HMI
VREF	-	-	-	-

<b>&lt;轴名称&gt;.Actor.Interface.AddressOut (定位轴工艺对象 V5 及更高版本)</b>				
PROFIdrive 报文的输出地址 (内部参数)				
数据类型	起始值	访问	有效	HMI
VREF	-	-	-	-



<b>&lt;轴名称&gt;.Actor.Interface.EnableDriveOutput</b>				
使能输出（内部参数）				
数据类型	起始值	访问	有效	HMI
VREF	-	-	-	-

<b>&lt;轴名称&gt;.Actor.Interface.DriveReadyInput</b>				
输入就绪（内部参数）				
数据类型	起始值	访问	有效	HMI
VREF	-	-	-	-

<b>&lt;轴名称&gt;.Actor.Interface.PTO</b>				
脉冲输出（内部参数）				
数据类型	起始值	访问	有效	HMI
DWord	0	-	-	-

<b>&lt;轴名称&gt;.Actor.DriveParameter.ReferenceSpeed</b> （定位轴工艺对象 V5 及更高版本）				
驱动器速度设定值 (N-set) 的参考值 (100%)				
速度设定值在 PROFIdrive 报文以从 -200% 到 200%“ReferenceSpeed”的标准值的形式来传输。				
通过模拟量输出指定设定值时，只要驱动器允许，模拟量输出就可在从 -117% 到 117% 的范围内运行。				
数据类型	起始值	访问	有效	HMI
Real	3000.0	R	-	√
		WP	2, 9	-

<b>&lt;轴名称&gt;.Actor.DriveParameter.MaxSpeed</b> （定位轴工艺对象 V5 及更高版本）				
驱动器速度设定值 (N-set) 的最大值				
(PROFIdrive: $\text{MaxSpeed} \leq 2 \times \text{ReferenceSpeed}$ )				
模拟量设定值: $\text{MaxSpeed} \leq 1.17 \times \text{ReferenceSpeed}$ )				
数据类型	起始值	访问	有效	HMI
Real	3000.0	R	-	√
		WP	2, 9	-

<b>&lt;轴名称&gt;.Actor.DriveParameter.PulsesPerDriveRevolution</b>				
电机每转的脉冲数				
数据类型	起始值	访问	有效	HMI
DInt	1000.0	R	-	√
		WP	2, 9	-

参见

轴工艺对象 V1...3 的变量 (页 342)

## 1.11.8.6 Sensor[1] 变量

## Sensor[1] 变量 V5 及更高版本

## 图例

<b>数据类型</b>	变量的数据类型		
<b>起始值</b>	变量的起始值 轴组态可以改写起始值。		
<b>访问</b>	在用户程序中访问变量的方式：		
	R	可以在用户程序中读取该变量。	
	WP	如果该轴已禁用 (MC_Power.Status = FALSE), 可使用运动控制指令“MC_WriteParam”对该变量进行写操作。	
	-	不能在用户程序中使用该变量。	
<b>有效</b>	指定变量的更改何时生效：		
	2	通过 PTO (Pulse Train Output) 的驱动器连接： 轴已启用时	
	9	通过 PROFIdrive/模拟量输出的驱动器连接： 在启用轴的情况下重新启动工艺对象后	
<b>HMI</b>	该变量可用于 HMI 系统（符合用户程序中的变量访问）。		

<b>&lt;轴名称&gt;.Sensor[1].Type</b>				
编码器类型（内部参数）				
0 = 增量式				
1 = 绝对值				
<b>数据类型</b>	<b>起始值</b>	<b>访问</b>	<b>有效</b>	<b>HMI</b>
DInt	0	-	-	-

<b>&lt;轴名称&gt;.Sensor[1].InverseDirection</b>				
对实际值取反 FALSE: 实际值未反转 TRUE: 实际值已反转				
数据类型	起始值	访问	有效	HMI
Bool	FALSE	R	-	√
		WP	2, 9	-

<b>&lt;轴名称&gt;.Sensor[1].System</b>				
编码器系统 0 = 线性编码器 1 = 旋转编码器				
数据类型	起始值	访问	有效	HMI
DInt	1	R	-	√
		WP	2, 9	-

<b>&lt;轴名称&gt;.Sensor[1].MountingMode</b>				
编码器的安装类型 0 = 驱动器端 2 = 外部				
数据类型	起始值	访问	有效	HMI
DInt	0	R	-	√
		WP	2, 9	-

**Sensor[1].Interface 变量 V5 及更高版本**

## 图例

<b>数据类型</b>	变量的数据类型
<b>起始值</b>	变量的起始值 轴组态可以改写起始值。
<b>访问</b>	在用户程序中访问变量的方式： - 不能在用户程序中使用该变量。
<b>有效</b>	指定变量的更改何时生效。
<b>HMI</b>	该变量可用于 HMI 系统（符合用户程序中的变量访问）。

<b>&lt;轴名称&gt;.Sensor[1].Interface.Type</b>				
编码器连接（内部参数） 0 = PROFIdrive 编码器（位于 PROFINET） 1 = 工艺模块 (TM) 上的编码器 2 = 驱动器上的编码器 4 = 高速计数器上的编码器				
<b>数据类型</b>	<b>起始值</b>	<b>访问</b>	<b>有效</b>	<b>HMI</b>
DInt	4	-	-	-

<b>&lt;轴名称&gt;.Sensor[1].Interface.AddressIn</b>				
PROFIdrive 报文的输入地址（内部参数）				
<b>数据类型</b>	<b>起始值</b>	<b>访问</b>	<b>有效</b>	<b>HMI</b>
VREF	-	-	-	-

<b>&lt;轴名称&gt;.Sensor[1].Interface.AddressOut</b>				
PROFIdrive 报文的输出地址（内部参数）				
<b>数据类型</b>	<b>起始值</b>	<b>访问</b>	<b>有效</b>	<b>HMI</b>
VREF	-	-	-	-

<b>&lt;轴名称&gt;.Sensor[1].Interface.HSC</b>				
编码器将向其传输实际值的高速计数器（内部参数）				
数据类型	起始值	访问	有效	HMI
DWord	-	-	-	-

## Sensor[1].Parameter 变量 V5 及更高版本

## 图例

<b>数据类型</b>	变量的数据类型	
<b>起始值</b>	变量的起始值 轴组态可以改写起始值。	
<b>访问</b>	在用户程序中访问变量的方式：	
	R	可以在用户程序中读取该变量。
	WP	如果该轴已禁用 (MC_Power.Status = FALSE), 可使用运动控制指令“MC_WriteParam”对该变量进行写操作。
<b>有效</b>	指定变量的更改何时生效：	
	2	通过 PTO (Pulse Train Output) 的驱动器连接： 轴已启用时
	9	通过 PROFIdrive/模拟量输出的驱动器连接： 在启用轴的情况下重新启动工艺对象后
<b>HMI</b>	该变量可用于 HMI 系统（符合用户程序中的变量访问）。	

<b>&lt;轴名称&gt;.Sensor[1].Parameter.Resolution</b>				
线性编码器的精度（两个编码器脉冲之间的偏移值）				
数据类型	起始值	访问	有效	HMI
Real	0.001	R	-	√
		WP	2, 9	-

<b>&lt;轴名称&gt;.Sensor[1].Parameter.StepsPerRevolution</b>				
旋转编码器的每转增量数				
数据类型	起始值	访问	有效	HMI
UDInt	2048	R	-	√
		WP	2, 9	-

<b>&lt;轴名称&gt;.Sensor[1].Parameter.FineResolutionXist1</b>				
高精度的位数 Gn_XIST1（周期性实际编码器值）				
数据类型	起始值	访问	有效	HMI
UDInt	11	R	-	√
		WP	2, 9	-

<b>&lt;轴名称&gt;.Sensor[1].Parameter.FineResolutionXist2</b>				
高精度的位数 Gn_XIST2（编码器的绝对值）				
数据类型	起始值	访问	有效	HMI
UDInt	9	R	-	√
		WP	2, 9	-

<b>&lt;轴名称&gt;.Sensor[1].Parameter.DeterminableRevolutions</b>				
多圈绝对值编码器的差动转数 (对于单转绝对值编码器 = 1; 对于增量编码器 = 0)				
数据类型	起始值	访问	有效	HMI
UDInt	1	R	-	√
		WP	2, 9	-

<b>&lt;轴名称&gt;.Sensor[1].Parameter.DistancePerRevolution</b>				
外部安装的编码器每转的负载距离				
数据类型	起始值	访问	有效	HMI
Real	100.0	R	-	√
		WP	2, 9	-



## Sensor[1].ActiveHoming 变量 V4 及更高版本

## 图例

<b>数据类型</b>	变量的数据类型	
<b>起始值</b>	变量的起始值 轴组态可以改写起始值。	
<b>访问</b>	在用户程序中访问变量的方式:	
	RW	可以在用户程序中读写该变量。 可使用运动控制指令“MC_WriteParam”对该变量进行写操作。
	R	可以在用户程序中读取该变量。
	WP	如果该轴已禁用 (MC_Power.Status = FALSE), 可使用运动控制指令“MC_WriteParam”对该变量进行写操作。
	-	不能在用户程序中使用该变量。
<b>有效</b>	指定变量的更改何时生效:	
	1	通过 PTO (Pulse Train Output) 的驱动器连接: 在轴激活 (<轴名称>.StatusBits.Activated 变量从 FALSE 更改为 TRUE)、 禁用或启用时
	2	通过 PTO (Pulse Train Output) 的驱动器连接: 轴已启用时
	8	通过 PTO (Pulse Train Output) 的驱动器连接: 启动主动归位命令时
	9	通过 PROFIdrive/模拟量输出的驱动器连接: 在启用轴的情况下重新启动工艺对象后
	10	通过 PROFIdrive/模拟量输出的驱动器连接: 下一次调用 MC-Servo [OB91] 时
<b>HMI</b>	该变量可用于 HMI 系统 (符合用户程序中的变量访问)。	

<b>&lt;轴名称&gt;.Sensor[1].ActiveHoming.Mode</b>				
主动归位模式				
<ul style="list-style-type: none"> <li>定位轴工艺对象 V5 及更高版本：                             <ul style="list-style-type: none"> <li>0 = 基于 PROFIdrive 报文的零位标记（不是 PTO）</li> <li>1 = 基于 PROFIdrive 报文和接近开关的零位标记（不是 PTO）</li> <li>2 = 通过数字量输入归位</li> </ul> </li> <li>定位轴工艺对象 V4：                             <ul style="list-style-type: none"> <li>2 = 通过数字量输入归位</li> </ul> </li> </ul>				
数据类型	起始值	访问	有效	HMI
DInt	2	R	-	√
		WP	2, 9	-

<b>&lt;轴名称&gt;.Sensor[1].ActiveHoming.SidelInput</b>				
主动归位期间，轴归位时所到达的归位开关端				
FALSE = 下端				
TRUE = 上端				
数据类型	起始值	访问	有效	HMI
Bool	FALSE	RW	1, 8, 10	√

<b>&lt;轴名称&gt;.Sensor[1].ActiveHoming.DigitalInputAddress</b>				
归位开关的符号输入地址（内部参数）				
数据类型	起始值	访问	有效	HMI
VREF	-	-	-	-

<b>&lt;轴名称&gt;.Sensor[1].ActiveHoming.HomePositionOffset</b>				
起始位置偏移值（主动归位）				
（用组态的测量单位表示）				
数据类型	起始值	访问	有效	HMI
Real	0.0	RW	1, 8, 10	√

<b>&lt;轴名称&gt;.Sensor[1].ActiveHoming.SwitchLevel</b>				
选择到达归位开关时 CPU 输入端存在的信号电平 FALSE = 低电平（低电平有效） TRUE = 高电平（高电平有效）				
数据类型	起始值	访问	有效	HMI
Bool	TRUE	RW	1, 8, 10	√

## 参见

轴工艺对象 V1...3 的变量 (页 342)

## Sensor[1].PassiveHoming 变量 V4 及更高版本

## 图例

<b>数据类型</b>	变量的数据类型	
<b>起始值</b>	变量的起始值 轴组态可以改写起始值。	
<b>访问</b>	在用户程序中访问变量的方式:	
	RW	可以在用户程序中读写该变量。 可使用运动控制指令“MC_WriteParam”对该变量进行写操作。
	R	可以在用户程序中读取该变量。
	WP	如果该轴已禁用 (MC_Power.Status = FALSE), 可使用运动控制指令“MC_WriteParam”对该变量进行写操作。
	-	不能在用户程序中使用该变量。
<b>有效</b>	指定变量的更改何时生效:	
	1	通过 PTO (Pulse Train Output) 的驱动器连接: 在轴激活 (<轴名称>.StatusBits.Activated 变量从 FALSE 更改为 TRUE)、 禁用或启用时
	2	通过 PTO (Pulse Train Output) 的驱动器连接: 轴已启用时
	7	通过 PTO (Pulse Train Output) 的驱动器连接: 启动被动归位命令时
	9	通过 PROFIdrive/模拟量输出的驱动器连接: 在启用轴的情况下重新启动工艺对象后
	10	通过 PROFIdrive/模拟量输出的驱动器连接: 下一次调用 MC-Servo [OB91] 时
<b>HMI</b>	该变量可用于 HMI 系统 (符合用户程序中的变量访问)。	

<b>&lt;轴名称&gt;.Sensor[1].PassiveHoming.Mode</b>				
被动归位模式				
<ul style="list-style-type: none"> <li>定位轴工艺对象 V5 及更高版本：               <ul style="list-style-type: none"> <li>0 = 基于 PROFIdrive 报文的零位标记（不是 PTO）</li> <li>1 = 基于 PROFIdrive 报文和接近开关的零位标记（不是 PTO）</li> <li>2 = 通过数字量输入归位</li> </ul> </li> <li>定位轴工艺对象 V4：               <ul style="list-style-type: none"> <li>2 = 通过数字量输入归位</li> </ul> </li> </ul>				
数据类型	起始值	访问	有效	HMI
DInt	2	R	-	√
		WP	2, 9	-

<b>&lt;轴名称&gt;.Sensor[1].PassiveHoming.SideInput</b>				
被动归位期间，轴归位时所到达的归位开关端				
FALSE = 下端				
TRUE = 上端				
数据类型	起始值	访问	有效	HMI
Bool	FALSE	RW	1, 7, 10	√

<b>&lt;轴名称&gt;.Sensor[1].PassiveHoming.DigitalInputAddress</b>				
归位开关的符号输入地址（内部参数）				
数据类型	起始值	访问	有效	HMI
VREF	-	-	-	-

<b>&lt;轴名称&gt;.Sensor[1].PassiveHoming.SwitchLevel</b>				
选择到达归位开关时 CPU 输入端存在的电平 FALSE = 低电平（低电平有效） TRUE = 高电平（高电平有效）				
数据类型	起始值	访问	有效	HMI
Bool	TRUE	RW	1, 7, 10	√

参见

轴工艺对象 V1...3 的变量 (页 342)

## 1.11.8.7 自 V4 起的单位变量

## 图例

<b>数据类型</b>	变量的数据类型	
<b>起始值</b>	变量的起始值 轴组态可以改写起始值。	
<b>访问</b>	在用户程序中访问变量的方式：	
	R	可以在用户程序中读取该变量。
	WP	如果该轴已禁用 (MC_Power.Status = FALSE), 可使用运动控制指令“MC_WriteParam”对该变量进行写操作。
<b>有效</b>	指定变量的更改何时生效：	
	2	通过 PTO (Pulse Train Output) 的驱动器连接： 轴已启用时
	9	通过 PROFIdrive/模拟量输出的驱动器连接： 在启用轴的情况下重新启动工艺对象后
<b>HMI</b>	该变量可用于 HMI 系统（符合用户程序中的变量访问）。	

<b>&lt;轴名称&gt;.Units.LengthUnit</b>				
参数的已组态测量单位				
-1 =“脉冲”				
1005 =“°”（度）				
1013 =“毫米”				
1010 =: “米”				
1018 =“英尺”				
1019 =“英寸”				
数据类型	起始值	访问	有效	HMI
Int	1013	R	-	√
		WP	2, 9	-

## 参见

轴工艺对象 V1...3 的变量 (页 342)

### 1.11.8.8 自 V4 起的机械变量

#### 图例

<b>数据类型</b>	变量的数据类型	
<b>起始值</b>	变量的起始值 轴组态可以改写起始值。	
<b>访问</b>	在用户程序中访问变量的方式：	
	R	可以在用户程序中读取该变量。
	WP	如果该轴已禁用 (MC_Power.Status = FALSE)， 可使用运动控制指令“MC_WriteParam”对该变量进行写操作。
<b>有效</b>	指定变量的更改何时生效：	
	2	通过 PTO (Pulse Train Output) 的驱动器连接： 轴已启用时
	9	通过 PROFIdrive/模拟量输出的驱动器连接： 在启用轴的情况下重新启动工艺对象后
<b>HMI</b>	该变量可用于 HMI 系统（符合用户程序中的变量访问）。	

<b>&lt;轴名称&gt;.Mechanics.LeadScrew</b>				
每转的距离 (用组态的测量单位表示)				
<b>数据类型</b>	<b>起始值</b>	<b>访问</b>	<b>有效</b>	<b>HMI</b>
Real	10.0	R	-	√
		WP	2, 9	-

#### 参见

轴工艺对象 V1...3 的变量 (页 342)



## 1.11.8.9 Modulo 变量 V5 及更高版本

## 图例

<b>数据类型</b>	变量的数据类型	
<b>起始值</b>	变量的起始值 轴组态可以改写起始值。	
<b>访问</b>	在用户程序中访问变量的方式：	
	R	可以在用户程序中读取该变量。
	WP	如果该轴已禁用 (MC_Power.Status = FALSE), 可使用运动控制指令“MC_WriteParam”对该变量进行写操作。
<b>有效</b>	指定变量的更改何时生效：	
	2	通过 PTO (Pulse Train Output) 的驱动器连接：轴已启用时
	9	通过 PROFIdrive/模拟量输出的驱动器连接： 在启用轴的情况下重新启动工艺对象后
<b>HMI</b>	该变量可用于 HMI 系统（符合用户程序中的变量访问）。	

<b>&lt;轴名称&gt;.Modulo.Enable</b>				
启用模数 FALSE = 模数转换已禁用 TRUE = 模数转换已激活 启用了模数转换后，应针对模数长度 > 0.0 的情况进行检查				
<b>数据类型</b>	<b>起始值</b>	<b>访问</b>	<b>有效</b>	<b>HMI</b>
Bool	FALSE	R	-	√
		WP	2, 9	-

<b>&lt;轴名称&gt;.Modulo.Length</b>				
模数长度				
<b>数据类型</b>	<b>起始值</b>	<b>访问</b>	<b>有效</b>	<b>HMI</b>
Real	360.0	R	-	√
		WP	2, 9	-

<b>&lt;轴名称&gt;.Modulo.StartValue</b>				
模数起始值				
数据类型	起始值	访问	有效	HMI
Real	0.0	R	-	√
		WP	2, 9	-

## 1.11.8.10 自 V4 起的 DynamicLimits 变量

## 图例

<b>数据类型</b>	变量的数据类型	
<b>起始值</b>	变量的起始值 轴组态可以改写起始值。	
<b>访问</b>	在用户程序中访问变量的方式：	
	R	可以在用户程序中读取该变量。
	WP	如果该轴已禁用 (MC_Power.Status = FALSE), 可使用运动控制指令“MC_WriteParam”对该变量进行写操作。
<b>有效</b>	指定变量的更改何时生效：	
	2	通过 PTO (Pulse Train Output) 的驱动器连接：轴已启用时
	9	通过 PROFIdrive/模拟量输出的驱动器连接： 在启用轴的情况下重新启动工艺对象后
<b>HMI</b>	该变量可用于 HMI 系统（符合用户程序中的变量访问）。	

<b>&lt;轴名称&gt;.DynamicLimits.MaxVelocity</b>				
轴的最大速度 (用组态的测量单位表示)				
<b>数据类型</b>	<b>起始值</b>	<b>访问</b>	<b>有效</b>	<b>HMI</b>
Real	250.0	R	-	√
		WP	2, 9	-

<b>&lt;轴名称&gt;.DynamicLimits.MinVelocity</b>				
轴的启动/停止速度 (用组态的测量单位表示)				
<b>数据类型</b>	<b>起始值</b>	<b>访问</b>	<b>有效</b>	<b>HMI</b>
Real	10.0	R	-	√
		WP	2, 9	-

## 参见

轴工艺对象 V1...3 的变量 (页 342)

## 1.11.8.11 自 V4 起的 DynamicDefaults 变量

## 图例

<b>数据类型</b>	变量的数据类型	
<b>起始值</b>	变量的起始值 轴组态可以改写起始值。	
<b>访问</b>	在用户程序中访问变量的方式：	
	<b>RW</b>	可以在用户程序中读写该变量。 可使用运动控制指令“MC_WriteParam”对该变量进行写操作。
<b>有效</b>	指定变量的更改何时生效：	
	<b>1</b>	通过 PTO (Pulse Train Output) 的驱动器连接： 在轴激活（<轴名称>.StatusBits.Activated 变量从 FALSE 更改为 TRUE）、禁用或启用时
	<b>5</b>	通过 PTO (Pulse Train Output) 的驱动器连接： 下一次启动 MC_MoveAbsolute、 MC_MoveRelative、MC_MoveVelocity、MC_MoveJog、MC_Halt、 MC_CommandTable 或激活的 MC_Home 命令时 (Mode = 3)
	<b>6</b>	通过 PTO (Pulse Train Output) 的驱动器连接： MC_MoveJog 命令停止时
	<b>10</b>	通过 PROFIdrive/模拟量输出的驱动器连接： 下一次调用 MC-Servo [OB91] 时
<b>HMI</b>	该变量可用于 HMI 系统（符合用户程序中的变量访问）。	

<b>&lt;轴名称&gt;.DynamicDefaults.Acceleration</b>				
轴的加速度 (用组态的测量单位表示)				
<b>数据类型</b>	<b>起始值</b>	<b>访问</b>	<b>有效</b>	<b>HMI</b>
Real	48.0	RW	1, 5, 6, 10	√

<b>&lt;轴名称&gt;.DynamicDefaults.Deceleration</b>				
轴的减速度 (用组态的测量单位表示)				
数据类型	起始值	访问	有效	HMI
Real	48.0	RW	1, 5, 6, 10	√

<b>&lt;轴名称&gt;.DynamicDefaults.EmergencyDeceleration</b>				
轴的急停减速度 (用组态的测量单位表示)				
数据类型	起始值	访问	有效	HMI
Real	120.0	RW	1, 5, 6, 10	√

<b>&lt;轴名称&gt;.DynamicDefaults.Jerk</b>				
轴加速斜坡和减速斜坡期间的冲击 (用组态的测量单位表示) 在组态的加加速度大于 0.00004 mm/s <sup>2</sup> 时，将激活加加速度。				
数据类型	起始值	访问	有效	HMI
Real	192.0	RW	1, 5, 10	√

## 参见

轴工艺对象 V1...3 的变量 (页 342)

1.11.8.12 自 V4 起的 PositionLimitsSW 变量

图例

<b>数据类型</b>	变量的数据类型	
<b>起始值</b>	变量的起始值 轴组态可以改写起始值。	
<b>访问</b>	在用户程序中访问变量的方式：	
	RW	可以在用户程序中读写该变量。 可使用运动控制指令“MC_WriteParam”对该变量进行写操作。
<b>有效</b>	指定变量的更改何时生效：	
	1	通过 PTO (Pulse Train Output) 的驱动器连接： 在轴激活（<轴名称>.StatusBits.Activated 变量从 FALSE 更改为 TRUE）、禁用或启用时
	5	通过 PTO (Pulse Train Output) 的驱动器连接： 下一次启动 MC_MoveAbsolute、MC_MoveRelative、MC_MoveVelocity、MC_MoveJog、MC_Halt、MC_CommandTable 或激活的 MC_Home 命令时 (Mode = 3)
	6	通过 PTO (Pulse Train Output) 的驱动器连接： MC_MoveJog 命令停止时
	10	通过 PROFIdrive/模拟量输出的驱动器连接： 下一次调用 MC-Servo [OB91] 时
<b>HMI</b>	该变量可用于 HMI 系统（符合用户程序中的变量访问）。	

<b>&lt;轴名称&gt;.PositionLimitsSW.Active</b>				
软限位开关激活 FALSE = 软限位开关取消激活。 TRUE = 软限位开关激活。				
数据类型	起始值	访问	有效	HMI
Bool	FALSE	RW	1, 5, 6, 10	√

<b>&lt;轴名称&gt;.PositionLimitsSW.MinPosition</b>				
软限位开关下限位 (用组态的测量单位表示)				
数据类型	起始值	访问	有效	HMI
Real	-10000.0	RW	1, 5, 6, 10	√

<b>&lt;轴名称&gt;.PositionLimitsSW.MaxPosition</b>				
软限位开关上限位 (用组态的测量单位表示)				
数据类型	起始值	访问	有效	HMI
Real	10000.0	RW	1, 5, 6, 10	√

## 参见

轴工艺对象 V1...3 的变量 (页 342)

## 1.11.8.13 自 V4 起的 PositionLimitsHW 变量

## 图例

<b>数据类型</b>	变量的数据类型	
<b>起始值</b>	变量的起始值 轴组态可以改写起始值。	
<b>访问</b>	在用户程序中访问变量的方式：	
	RW	可以在用户程序中读写该变量。 可使用运动控制指令“MC_WriteParam”对该变量进行写操作。
	R	可以在用户程序中读取该变量。
	WP	如果该轴已禁用 (MC_Power.Status = FALSE)，可使用运动控制指令“MC_WriteParam”对该变量进行写操作。
	-	不能在用户程序中使用该变量。
<b>有效</b>	指定变量的更改何时生效：	
	1	通过 PTO (Pulse Train Output) 的驱动器连接： 在轴激活 (<轴名称>.StatusBits.Activated 变量从 FALSE 更改为 TRUE)、 禁用或启用时
	2	通过 PTO (Pulse Train Output) 的驱动器连接： 轴已启用时
	5	通过 PTO (Pulse Train Output) 的驱动器连接： 下一次启动 MC_MoveAbsolute、MC_MoveRelative、 MC_MoveVelocity、MC_MoveJog、MC_Halt、MC_CommandTable 或激活的 MC_Home 命令时 (Mode = 3)
	6	通过 PTO (Pulse Train Output) 的驱动器连接： MC_MoveJog 命令停止时
	9	通过 PROFIdrive/模拟量输出的驱动器连接： 重新启动工艺对象时
	10	通过 PROFIdrive/模拟量输出的驱动器连接： 下一次调用 MC-Servo [OB91] 时
<b>HMI</b>	该变量可用于 HMI 系统（符合用户程序中的变量访问）。	



<b>&lt;轴名称&gt;.PositionLimitsHW.Active</b>				
硬限位开关激活 FALSE = 硬限位开关取消激活。 TRUE = 硬限位开关激活。				
数据类型	起始值	访问	有效	HMI
Bool	FALSE	RW	1, 5, 6, 10	√

<b>&lt;轴名称&gt;.PositionLimitsHW.MinSwitchLevel</b>				
选择到达下限硬限位开关时 CPU 输入端存在的信号电平 FALSE = 低电平（低电平有效） TRUE = 高电平（高电平有效）				
数据类型	起始值	访问	有效	HMI
Bool	FALSE	R	-	√
		WP	2, 9	-

<b>&lt;轴名称&gt;.PositionLimitsHW.MinSwitchAddress</b>				
下限硬限位开关的符号输入地址（内部参数）				
数据类型	起始值	访问	有效	HMI
VREF	-	-	-	-

<b>&lt;轴名称&gt;.PositionLimitsHW.MaxSwitchLevel</b>				
选择到达上限硬限位开关时 CPU 输入端存在的信号电平 FALSE = 低电平（低电平有效） TRUE = 高电平（高电平有效）				
数据类型	起始值	访问	有效	HMI
Bool	FALSE	R	-	√
		WP	2, 9	-

<b>&lt;轴名称&gt;.PositionLimitsHW.MaxSwitchAddress</b>				
上限硬限位开关的输入地址（内部参数）				
数据类型	起始值	访问	有效	HMI
VREF	-	-	-	-

参见

轴工艺对象 V1...3 的变量 (页 342)

## 1.11.8.14 自 V4 起的回原点变量

## 图例

<b>数据类型</b>	变量的数据类型	
<b>起始值</b>	变量的起始值 轴组态可以改写起始值。	
<b>访问</b>	在用户程序中访问变量的方式：	
	RW	可以在用户程序中读写该变量。 可使用运动控制指令“MC_WriteParam”对该变量进行写操作。
<b>有效</b>	指定变量的更改何时生效：	
	1	通过 PTO (Pulse Train Output) 的驱动器连接： 在轴激活（<轴名称>.StatusBits.Activated 变量从 FALSE 更改为 TRUE）、 禁用或启用时
	8	通过 PTO (Pulse Train Output) 的驱动器连接： 启动主动归位命令时
	10	通过 PROFIdrive/模拟量输出的驱动器连接： 下一次调用 MC-Servo [OB91] 时
<b>HMI</b>	该变量可用于 HMI 系统（符合用户程序中的变量访问）。	

<b>&lt;轴名称&gt;.Homing.AutoReversal</b>				
主动归位期间激活硬限位开关处的自动反向 FALSE = 取消激活硬限位开关处的自动反向。 TRUE= 激活硬限位开关处的自动反向。				
<b>数据类型</b>	<b>起始值</b>	<b>访问</b>	<b>有效</b>	<b>HMI</b>
Bool	FALSE	RW	1, 8, 10	√

<b>&lt;轴名称&gt;.Homing.ApproachDirection</b>				
主动归位期间的逼近方向和归位方向 FALSE = 负逼近方向（用于查找归位开关）和负归位方向 TRUE = 正逼近方向（用于查找归位开关）和正归位方向				
<b>数据类型</b>	<b>起始值</b>	<b>访问</b>	<b>有效</b>	<b>HMI</b>
Bool	TRUE	RW	1, 8, 10	√

<b>&lt;轴名称&gt;.Homing.ApproachVelocity</b>				
主动归位期间轴的逼近速度 (用组态的测量单位表示)				
数据类型	起始值	访问	有效	HMI
Real	200.0	RW	1, 8, 10	√

<b>&lt;轴名称&gt;.Homing.ReferencingVelocity</b>				
主动归位期间轴的归位速度 (用组态的测量单位表示)				
数据类型	起始值	访问	有效	HMI
Real	40.0	RW	1, 8, 10	√

参见

轴工艺对象 V1...3 的变量 (页 342)

## 1.11.8.15 PositionControl 变量 V5 及更高版本

## 图例

<b>数据类型</b>	变量的数据类型	
<b>起始值</b>	变量的起始值 轴组态可以改写起始值。	
<b>访问</b>	在用户程序中访问变量的方式：	
	R	可以在用户程序中读取该变量。
	WP	如果该轴已禁用 (MC_Power.Status = FALSE), 可使用运动控制指令“MC_WriteParam”对该变量进行写操作。
<b>有效</b>	指定变量的更改何时生效：	
	10	通过 PROFIdrive/模拟量输出的驱动器连接： 下一次调用 MC-Servo [OB91] 时
<b>HMI</b>	该变量可用于 HMI 系统（符合用户程序中的变量访问）。	

<b>&lt;轴名称&gt;.PositionControl.Kv</b>				
位置控制中的比例增益 (“Kv”> 0.0)				
<b>数据类型</b>	<b>起始值</b>	<b>访问</b>	<b>有效</b>	<b>HMI</b>
Real	10.0	R	-	√
		WP	10	-

<b>&lt;轴名称&gt;.PositionControl.Kpc</b>				
位置控制的百分比速度预控制				
<b>数据类型</b>	<b>起始值</b>	<b>访问</b>	<b>有效</b>	<b>HMI</b>
Real	100.0	R	-	√
		WP	10	-

1.11.8.16 FollowingError 变量 V5 及更高版本

图例

<b>数据类型</b>	变量的数据类型	
<b>起始值</b>	变量的起始值 轴组态可以改写起始值。	
<b>访问</b>	在用户程序中访问变量的方式：	
	R	可以在用户程序中读取该变量。
	WP	如果该轴已禁用 (MC_Power.Status = FALSE), 可使用运动控制指令“MC_WriteParam”对该变量进行写操作。
<b>有效</b>	指定变量的更改何时生效：	
	9	通过 PROFIdrive/模拟量输出的驱动器连接： 在启用轴的情况下重新启动工艺对象后
	10	通过 PROFIdrive/模拟量输出的驱动器连接： 下一次调用 MC-Servo [OB91] 时
<b>HMI</b>	该变量可用于 HMI 系统（符合用户程序中的变量访问）。	

<b>&lt;轴名称&gt;.FollowingError.EnableMonitoring</b>				
启用跟随误差监控 FALSE = 跟随误差监控已禁用 TRUE = 跟随误差监控已启用				
数据类型	起始值	访问	有效	HMI
Bool	TRUE	R	-	√
		WP	9	-

<b>&lt;轴名称&gt;.FollowingError.MinValue</b>				
在低于“MinVelocity”值的速度下的允许跟随误差。				
数据类型	起始值	访问	有效	HMI
Real	10.0	R	-	√
		WP	10	-

<b>&lt;轴名称&gt;.FollowingError.MaxValue</b>				
在最大速度下可能达到的最大允许跟随误差。				
数据类型	起始值	访问	有效	HMI
Real	100.0	R	-	√
		WP	10	-

<b>&lt;轴名称&gt;.FollowingError.MinVelocity</b>				
"MinValue"允许低于此速度并保持恒定。				
数据类型	起始值	访问	有效	HMI
Real	10.0	R	-	√
		WP	10	-

1.11.8.17 PositionMonitoring 变量 V5 及更高版本

图例

<b>数据类型</b>	变量的数据类型	
<b>起始值</b>	变量的起始值 轴组态可以改写起始值。	
<b>访问</b>	在用户程序中访问变量的方式：	
	R	可以在用户程序中读取该变量。
	WP	如果该轴已禁用 (MC_Power.Status = FALSE), 可使用运动控制指令“MC_WriteParam”对该变量进行写操作。
<b>有效</b>	指定变量的更改何时生效：	
	9	通过 PROFIdrive/模拟量输出的驱动器连接： 在启用轴的情况下重新启动工艺对象后
	10	通过 PROFIdrive/模拟量输出的驱动器连接： 下一次调用 MC-Servo [OB91] 时
<b>HMI</b>	该变量可用于 HMI 系统（符合用户程序中的变量访问）。	

<b>&lt;轴名称&gt;.PositioningMonitoring.ToleranceTime</b>				
容差时间 从达到速度设定值 0 开始直至进入到定位窗口中的最大允许持续时间。				
数据类型	起始值	访问	有效	HMI
Real	1.0	R	-	√
		WP	10	-

<b>&lt;轴名称&gt;.PositioningMonitoring.MinDwellTime</b>				
在定位窗口停留的最短时间				
数据类型	起始值	访问	有效	HMI
Real	0.1	R	-	√
		WP	10	-



<b>&lt;轴名称&gt;.PositioningMonitoring.Window</b>				
定位窗口				
数据类型	起始值	访问	有效	HMI
Real	1.0	R	-	√
		WP	10	-

<b>&lt;轴名称&gt;.PositioningMonitoring.Enable</b>				
启用定位监控				
FALSE = 定位监控已禁用				
TRUE = 定位监控已启用				
数据类型	起始值	访问	有效	HMI
Bool	FALSE	R	-	√
		WP	9	-

### 1.11.8.18 StandstillSignal 变量 V5 及更高版本

#### 图例

<b>数据类型</b>	变量的数据类型	
<b>起始值</b>	变量的起始值 轴组态可以改写起始值。	
<b>访问</b>	在用户程序中访问变量的方式：	
	R	可以在用户程序中读取该变量。
	WP	如果该轴已禁用 (MC_Power.Status = FALSE) ，可使用运动控制指令“MC_WriteParam”对该变量进行写操作。
<b>有效</b>	指定变量的更改何时生效：	
	10	通过 PROFIdrive/模拟量输出的驱动器连接： 下一次调用 MC-Servo [OB91] 时
<b>HMI</b>	该变量可用于 HMI 系统（符合用户程序中的变量访问）。	

<b>&lt;轴名称&gt;.StandstillSignal.VelocityThreshold</b>				
速度阈值 如果速度低于此阈值，则最短停留时间开始。				
数据类型	起始值	访问	有效	HMI
Real	5.0	R	-	√
		WP	10	-

<b>&lt;轴名称&gt;.StandstillSignal..MinDwellTime</b>				
最短停留时间				
数据类型	起始值	访问	有效	HMI
Real	0.01	R	-	√
		WP	10	-

## 1.11.8.19 自 V4 起的 StatusPositioning 变量

## 图例

<b>数据类型</b>	变量的数据类型
<b>起始值</b>	变量的起始值
<b>访问</b>	在用户程序中访问变量的方式:
	RCCP 可在用户程序中读取变量并在每个周期控制点处更新变量。
	RP 可使用运动控制指令“MC_ReadParam”读取该变量。 相应变量的当前值在命令的起始处决定。
<b>有效</b>	指定变量的更改何时生效。
<b>HMI</b>	可以在 HMI 系统中使用该变量。

<b>&lt;轴名称&gt;.StatusPositioning.Distance</b>				
轴距目标位置的当前距离 (用组态的测量单位表示) 该变量值仅在使用“MC_MoveAbsolute”、“MC_MoveRelative”或轴控制面板执行定位命令期间有效。				
<b>数据类型</b>	<b>起始值</b>	<b>访问</b>	<b>有效</b>	<b>HMI</b>
Real	0.0	RCCP、RP	-	√

<b>&lt;轴名称&gt;.StatusPositioning.TargetPosition</b>				
轴的目标位置 (用组态的测量单位表示) 该变量值仅在使用“MC_MoveAbsolute”、“MC_MoveRelative”或轴控制面板执行定位命令期间有效。				
<b>数据类型</b>	<b>起始值</b>	<b>访问</b>	<b>有效</b>	<b>HMI</b>
Real	0.0	RCCP、RP	-	√

<b>&lt;轴名称&gt;.StatusPositioning.FollowingError</b> （定位轴工艺对象 V5 及更高版本）				
轴的当前跟随误差 （用组态的测量单位表示） FollowingError = 0.0，对于通过 PTO (Pulse Train Output) 的驱动器连接。				
数据类型	起始值	访问	有效	HMI
Real	0.0	RCCP、RP	-	√

参见

运动状态 (页 187)

轴工艺对象 V1...3 的变量 (页 342)

## 1.11.8.20 StatusDrive 变量 V5 及更高版本

## 图例

<b>数据类型</b>	变量的数据类型	
<b>起始值</b>	变量的起始值 轴组态可以改写起始值。	
<b>访问</b>	在用户程序中访问变量的方式：	
	RCCP	可在用户程序中读取变量并在每个周期控制点处更新变量。
	RP	可使用运动控制指令“MC_ReadParam”读取该变量。 相应变量的当前值在命令的起始处决定。
<b>有效</b>	指定变量的更改何时生效：	
<b>HMI</b>	该变量可用于 HMI 系统（符合用户程序中的变量访问）。	

<b>&lt;轴名称&gt;.StatusDrive.InOperation</b>				
驱动器的运行状态 FALSE = 驱动器未就绪。将不执行设定值。 TRUE = 驱动器就绪。可以执行设定值。				
<b>数据类型</b>	<b>起始值</b>	<b>访问</b>	<b>有效</b>	<b>HMI</b>
Bool	FALSE	RCCP、RP	-	√

<b>&lt;轴名称&gt;.StatusDrive.CommunicationOK</b>				
控制器与驱动器之间的周期性总线通信 FALSE = 未建立通信 TRUE = 已建立通信				
<b>数据类型</b>	<b>起始值</b>	<b>访问</b>	<b>有效</b>	<b>HMI</b>
Bool	FALSE	RCCP、RP	-	√

## 1.11.8.21 StatusSensor 变量 V5 及更高版本

## 图例

<b>数据类型</b>	变量的数据类型	
<b>起始值</b>	变量的起始值 轴组态可以改写起始值。	
<b>访问</b>	RCCP	可在用户程序中读取变量并在每个周期控制点处更新变量。
	RP	可使用运动控制指令“MC_ReadParam”读取该变量。 相应变量的当前值在命令的起始处决定。
<b>有效</b>	指定变量的更改何时生效：	
<b>HMI</b>	该变量可用于 HMI 系统（符合用户程序中的变量访问）。	

**<轴名称>.StatusSensor.State**

编码器值的状态

0 = 无效

1 = 等待有效状态

2 = 有效

数据类型	起始值	访问	有效	HMI
DInt	0	RCCP、RP	-	√

**<轴名称>.StatusSensor.CommunicationOK**

控制器与编码器之间的周期性总线通信

FALSE = 未建立通信

TRUE= 已建立通信

数据类型	起始值	访问	有效	HMI
Bool	FALSE	RCCP、RP	-	√

**<轴名称>.StatusSensor.AbsEncoderOffset**

到绝对值编码器的值的起始点偏移。

此值将永久性存储在 CPU 中。

数据类型	起始值	访问	有效	HMI
Real	0.0	RCCP、RP	-	√

## 1.11.8.22 自 V4 起的 StatusBits 变量

## 图例

数据类型	变量的数据类型	
起始值	变量的起始值	
访问	在用户程序中访问变量的方式:	
	RCCP	可在用户程序中读取变量并在每个周期控制点处更新变量。
	RP	可使用运动控制指令“MC_ReadParam”读取该变量。 相应变量的当前值在命令的起始处决定。
有效	指定变量的更改何时生效。	
HMI	可以在 HMI 系统中使用该变量。	

<b>&lt;轴名称&gt;.StatusBits.Activated</b>				
轴的激活状态 FALSE = 轴未激活。 TRUE = 轴激活。轴连接到分配的 PTO (Pulse Train Output)。将循环更新工艺数据块中的数据。				
数据类型	起始值	访问	有效	HMI
Bool	FALSE	RCCP、RP	-	√

<b>&lt;轴名称&gt;.StatusBits.Enable</b>				
轴的启用状态 FALSE = 轴未启用。 TRUE = 轴已启用并已准备好接受运动控制命令。				
数据类型	起始值	访问	有效	HMI
Bool	FALSE	RCCP、RP	-	√

<b>&lt;轴名称&gt;.StatusBits.HomingDone</b>				
轴的归位状态 FALSE = 轴未归位。 TRUE = 轴已归位且能够执行绝对定位命令。 对于相对定位而言，轴不必归位。 在主动归位过程中，该状态为 FALSE。 如果轴事先已经归位，则在被动归位期间该状态会保持为 TRUE。				
数据类型	起始值	访问	有效	HMI
Bool	FALSE	RCCP、RP	-	√

<b>&lt;轴名称&gt;.StatusBits.Done</b>				
轴上的命令执行 FALSE = 轴上的运动控制命令已激活。 TRUE = 轴上的运动控制命令未激活。				
数据类型	起始值	访问	有效	HMI
Bool	FALSE	RCCP、RP	-	√

<b>&lt;轴名称&gt;.StatusBits.Error</b>				
轴上的错误状态 FALSE = 轴上没有激活的错误。 TRUE = 轴上发生了错误。 有关错误的详细信息，请参见自动控制模式下的运动控制指令的“ErrorID”和“ErrorInfo”参数。 在手动模式下，轴控制面板的“错误消息”(Error message) 框可显示有关错误原因的详细信息。				
数据类型	起始值	访问	有效	HMI
Bool	FALSE	RCCP、RP	-	√

<b>&lt;轴名称&gt;.StatusBits.StandStill</b>				
轴的停止状态 FALSE = 轴处于运动状态。 TRUE = 轴处于停止状态。				
数据类型	起始值	访问	有效	HMI
Bool	FALSE	RCCP、RP	-	√



<b>&lt;轴名称&gt;.StatusBits.PositioningCommand</b>				
执行定位命令 FALSE = 轴上的定位命令未激活。 TRUE = 轴正在执行“MC_MoveRelative”或“MC_MoveAbsolute”运动控制指令的定位命令。				
数据类型	起始值	访问	有效	HMI
Bool	FALSE	RCCP、RP	-	√

<b>&lt;轴名称&gt;.StatusBits.VelocityCommand</b>				
以速度设定值执行命令 FALSE = 轴上以速度设定值执行的命令未激活。 TRUE = 轴正在以“MC_MoveVelocity”或“MC_MoveJog”运动控制指令的速度设定值执行运动命令。				
数据类型	起始值	访问	有效	HMI
Bool	FALSE	RCCP、RP	-	√

<b>&lt;轴名称&gt;.StatusBits.HomingCommand</b>				
执行归位命令 FALSE = 轴上的归位命令未激活。 TRUE = 轴正在执行“MC_Home”运动控制指令的归位命令。				
数据类型	起始值	访问	有效	HMI
Bool	FALSE	RCCP、RP	-	√

<b>&lt;轴名称&gt;.StatusBits.CommandTableActive</b>				
执行命令表 FALSE = 轴上的命令表未激活。 TRUE = 该轴由“MC_CommandTable”运动控制指令控制。				
数据类型	起始值	访问	有效	HMI
Bool	FALSE	RCCP、RP	-	√

<b>&lt;轴名称&gt;.StatusBits.ConstantVelocity</b>				
恒速 FALSE = 轴处于加速、减速或停止状态。 TRUE = 已达到速度设定值。轴正以恒速运动。				
数据类型	起始值	访问	有效	HMI
Bool	FALSE	RCCP、RP	-	√

<b>&lt;轴名称&gt;.StatusBits.Accelerating</b>				
加速过程 FALSE = 轴正在减速、以恒速运动或处于停止状态。 TRUE = 轴正在加速。				
数据类型	起始值	访问	有效	HMI
Bool	FALSE	RCCP、RP	-	√

<b>&lt;轴名称&gt;.StatusBits.Decelerating</b>				
减速过程 FALSE = 轴正在加速、以恒速运动或处于停止状态。 TRUE = 轴正在减速。				
数据类型	起始值	访问	有效	HMI
Bool	FALSE	RCCP、RP	-	√

<b>&lt;轴名称&gt;.StatusBits.ControlPanelActive</b>				
轴控制面板的激活状态 FALSE = “自动模式”(Automatic mode) 已激活。用户程序对轴具有优先控制权。 TRUE = 在轴控制面板中启用了“手动控制”(Manual control) 模式。轴控制面板对轴具有优先控制权。 不能通过用户程序来控制轴。				
数据类型	起始值	访问	有效	HMI
Bool	FALSE	RCCP、RP	-	√

<b>&lt;轴名称&gt;.StatusBits.DriveReady</b>				
驱动器的运行状态 FALSE = 驱动器未准备就绪。将不执行设定值。 TRUE = 驱动器准备就绪。可以执行设定值。				
数据类型	起始值	访问	有效	HMI
Bool	FALSE	RCCP、RP	-	√

<b>&lt;轴名称&gt;.StatusBits.RestartRequired</b>				
需要重启轴 FALSE = 不需要重启轴。 TRUE = 值已在装载存储器中修改。要在 CPU 处于 RUN 模式时将值下载到工作存储器中，必须重启轴。 使用运动控制指令 MC_Reset 执行此操作。				
数据类型	起始值	访问	有效	HMI
Bool	FALSE	RCCP、RP	-	√

<b>&lt;轴名称&gt;.StatusBits.SWLimitMinActive</b>				
下限软限位开关的状态 FALSE = 轴保持在组态的工作区域内。 TRUE = 已到达或超过下限软限位开关。				
数据类型	起始值	访问	有效	HMI
Bool	FALSE	RCCP、RP	-	√

<b>&lt;轴名称&gt;.StatusBits.SWLimitMaxActive</b>				
上限软限位开关的状态 FALSE = 轴保持在组态的工作区域内。 TRUE = 已到达或超过上限软限位开关。				
数据类型	起始值	访问	有效	HMI
Bool	FALSE	RCCP、RP	-	√

<b>&lt;轴名称&gt;.StatusBits.HWLimitMinActive</b>				
下限硬限位开关的状态 FALSE = 轴保持在组态的允许行进范围内。 TRUE = 已到达或超过下限硬限位开关。				
数据类型	起始值	访问	有效	HMI
Bool	FALSE	RCCP、RP	-	√

<b>&lt;轴名称&gt;.StatusBits.HWLimitMaxActive</b>				
上限硬限位开关的状态 FALSE = 轴保持在组态的允许行进范围内。 TRUE = 已到达或超过上限硬限位开关。				
数据类型	起始值	访问	有效	HMI
Bool	FALSE	RCCP、RP	-	√

**参见**

状态和错误位（自工艺对象 V4 起） (页 183)

轴工艺对象 V1...3 的变量 (页 342)

## 1.11.8.23 自 V4 起的 ErrorBits 变量

## 图例

数据类型	变量的数据类型	
起始值	变量的起始值	
访问	在用户程序中访问变量的方式:	
	RCCP	可在用户程序中读取变量并在每个周期控制点处更新变量。
	RP	可使用运动控制指令“MC_ReadParam”读取该变量。 相应变量的当前值在命令的起始处决定。
有效	指定变量的更改何时生效。	
HMI	可以在 HMI 系统中使用该变量。	

<b>&lt;轴名称&gt;.ErrorBits.SystemFault</b>				
内部系统错误				
数据类型	起始值	访问	有效	HMI
Bool	FALSE	RCCP、RP	-	√

<b>&lt;轴名称&gt;.ErrorBits.ConfigFault</b>				
轴组态错误				
数据类型	起始值	访问	有效	HMI
Bool	FALSE	RCCP、RP	-	√

<b>&lt;轴名称&gt;.ErrorBits.DriveFault</b>				
驱动器中发生错误。“驱动器准备就绪”(Drive ready) 信号丢失。				
数据类型	起始值	访问	有效	HMI
Bool	FALSE	RCCP、RP	-	√

<b>&lt;轴名称&gt;.ErrorBits.SWLimit</b>				
到达或超出软限位开关				
数据类型	起始值	访问	有效	HMI
Bool	FALSE	RCCP、RP	-	√

<b>&lt;轴名称&gt;.ErrorBits.HWLimit</b>				
到达或超出硬限位开关				
数据类型	起始值	访问	有效	HMI
Bool	FALSE	RCCP	-	√

<b>&lt;轴名称&gt;.ErrorBits.DirectionFault</b>				
不允许的运动方向				
数据类型	起始值	访问	有效	HMI
Bool	FALSE	RCCP、RP、RP	-	√

<b>&lt;轴名称&gt;.ErrorBits.HWUsed</b>				
另一个轴正在使用相同的 PTO (Pulse Train Output) 并且该轴已通过“MC_Power”启用。				
数据类型	起始值	访问	有效	HMI
Bool	FALSE	RCCP、RP	-	√

<b>&lt;轴名称&gt;.ErrorBits.SensorFault</b> （定位轴工艺对象 V5 及更高版本）				
编码器系统中发生错误				
数据类型	起始值	访问	有效	HMI
Bool	FALSE	RCCP、RP	-	√

<b>&lt;轴名称&gt;.ErrorBits.CommunicationFault</b> （定位轴工艺对象 V5 及更高版本）				
通信错误 与所连接设备进行通信时发生错误。				
数据类型	起始值	访问	有效	HMI
Bool	FALSE	RCCP、RP	-	√

<b>&lt;轴名称&gt;.ErrorBits.FollowingErrorFault</b> （定位轴工艺对象 V5 及更高版本）				
超出了最大的允许跟随误差				
数据类型	起始值	访问	有效	HMI
Bool	FALSE	RCCP、RP	-	√

<b>&lt;轴名称&gt;.ErrorBits.PositioningFault</b> （定位轴工艺对象 V5 及更高版本）				
定位误差 在定位运动的末端，轴定位错误。				
数据类型	起始值	访问	有效	HMI
Bool	FALSE	RCCP、RP	-	√

**参见**

状态和错误位（自工艺对象 V4 起）（页 183）

轴工艺对象 V1...3 的变量（页 342）

**1.11.8.24 自 V4 起的 ControlPanel 变量**

“ControlPanel”变量中不含任何与用户相关的数据。不能在用户程序中访问这些变量。

**参见**

轴工艺对象 V1...3 的变量（页 342）

**1.11.8.25 自 V4 起的内部变量**

“Internal”变量中不含任何与用户相关的数据。不能在用户程序中访问这些变量。

**参见**

轴工艺对象 V1...3 的变量（页 342）

**1.11.8.26 工艺对象变量的更新**

工艺对象变量中指示的轴的状态和错误信息在各周期控制点处更新。

组态变量值的更改不会立即生效。

有关更改生效条件的信息，请参见相关变量的详细说明。

### 1.11.9 自 V4 起的命令表工艺对象变量

#### 1.11.9.1 自 V4 起的 Command[1...32] 变量

图例

<b>数据类型</b>	变量的数据类型	
<b>起始值</b>	变量的起始值 起始值可由命令表的组态覆盖。	
<b>访问</b>	在用户程序中访问变量的方式:	
	RW	可以在用户程序中读写该变量。
	R	可以在用户程序中读取该变量。
	-	不能在用户程序中使用该变量。
<b>有效</b>	指定变量的更改何时生效。	
<b>HMI</b>	可以在 HMI 系统中使用该变量。	

<b>&lt;命令表&gt;.Command[n].Type</b>				
命令类型				
<ul style="list-style-type: none"> <li>• 0 =“空”命令</li> <li>• 2 =“保持”命令</li> <li>• 5 =“相对定位”命令</li> <li>• 6 =“绝对定位”命令</li> <li>• 7 =“速度设定值”命令</li> <li>• 151 =“等待”命令</li> </ul>				
<b>数据类型</b>	<b>起始值</b>	<b>访问</b>	<b>有效</b>	<b>HMI</b>
Int	0	RW	-	√

<b>&lt;命令表&gt;.Command[n].Position</b>				
命令的目标位置/行进距离				
<b>数据类型</b>	<b>起始值</b>	<b>访问</b>	<b>有效</b>	<b>HMI</b>
Real	0.0	RW	-	√



<b>&lt;轴名称&gt;.Command[n].Velocity</b>				
速度命令				
数据类型	起始值	访问	有效	HMI
Real	0.0	RW	-	√

<b>&lt;命令表&gt;.Command[n].Duration</b>				
持续时间命令				
数据类型	起始值	访问	有效	HMI
Real	0.0	RW	-	√

<b>&lt;命令表&gt;.Command[n].NextStep</b>				
用于切换到下一条命令的模式				
<ul style="list-style-type: none"> <li>• 0 =“完成命令”</li> <li>• 1 =“混合运动”</li> </ul>				
数据类型	起始值	访问	有效	HMI
Int	0	RW	-	√

<b>&lt;命令表&gt;.Command[n].StepCode</b>				
命令步代码				
数据类型	起始值	访问	有效	HMI
Word	0	RW	-	√

## 参见

命令表工艺对象 V1...3 的变量 (页 366)

## 1.11.10 版本 V1...4

### 1.11.10.1 与运动控制相关的 CPU 输出（工艺版本 V1...3）

可用驱动器的数目取决于 CPU、PTO（脉冲串输出）的数目以及可用的脉冲发生器输出的数目。

下表提供了有关相关依赖性的信息：

#### 最大 PTO 数

可控制 PTO（驱动器）的最大数目取决于 CPU 的产品编号：

CPU 产品编号	PTO 数
xxxxxxx-1xx <b>30</b> -xxxx	2
xxxxxxx-1xx <b>31</b> -xxxx	4

无论是否使用信号板，可控制 PTO（驱动器）的最大数目都适用。

#### 可用的脉冲发生器输出

##### CPU

提供一个脉冲输出和一个方向输出，通过脉冲接口对步进电机驱动器或伺服电机驱动器进行控制。脉冲输出为驱动器提供电机运动所需的脉冲。

方向输出则用于控制驱动器的行进方向。

脉冲输出和方向输出彼此互相分配的关系保持不变并形成脉冲发生器输出。板载 CPU 输出或信号板输出可用作脉冲发生器输出。在设备组态期间，可以在“属性”(Properties) 选项卡上的脉冲发生器 (PTO/PWM) 下，选择板载 CPU 输出或信号板输出。

下表列出了每个 CPU 或信号板可用的驱动器的数目：

CPU		板载	信号板				
			DI2/DO2 x DC24V 20kHz	DI2/DO2 x DC24V 200kHz	DO4 x DC24V 200kHz	DI2/DO2 x DC5V 200kHz	DO4 x DC5V 200kHz
CPU 1211C、 CPU 1212C、 CPU 1214C (MLFB - 产品编号 xxxxxxx-1xx30- xxxx)	DC/DC/DC	2	2	2	2	2	2
	AC/DC/RLY	-	1	1	2	1	2
	DC/DC/RLY	-	1	1	2	1	2
CPU 1211C (MLFB - 产品编号 xxxxxxx-1xx31- xxxx)	DC/DC/DC	2	3	3	4	3	4
	AC/DC/RLY	-	1	1	2	1	2
	DC/DC/RLY	-	1	1	2	1	2
CPU 1212C (MLFB - 产品编号 xxxxxxx-1xx31- xxxx)	DC/DC/DC	3	4	4	4	4	4
	AC/DC/RLY	-	1	1	2	1	2
	DC/DC/RLY	-	1	1	2	1	2
CPU 1214C (MLFB - 产品编号 xxxxxxx-1xx31- xxxx)	DC/DC/DC	4	4	4	4	4	4
	AC/DC/RLY	-	1	1	2	1	2
	DC/DC/RLY	-	1	1	2	1	2
CPU 1215C	DC/DC/DC	4	4	4	4	4	4
	AC/DC/RLY	-	1	1	2	1	2
	DC/DC/RLY	-	1	1	2	1	2

下表列出了脉冲输出和方向输出的地址分配：

CPU S7-1200		输出 PTO1 *)		输出 PTO2 **)		输出 PTO3 *)		输出 PTO4 **)	
		脉冲	方向	脉冲	方向	脉冲	方向	脉冲	方向
CPU 1211C、 CPU 1212C、 CPU 1214C、 CPU1215C (DC/DC/DC)	CPU	Ax.0	Ax.1	Ax.2	Ax.3	Ax.4	Ax.5	Ax.6	Ax.7
	信号板	Ay.0	Ay.1	Ay.2	Ay.3	Ay.0	Ay.1	Ay.2	Ay.3
CPU 1211C、 CPU 1212C、 CPU 1214C、 CPU 1215C (AC/DC/RLY)	CPU	-	-	-	-	-	-	-	-
	信号板	Ay.0	Ay.1	Ay.2	Ay.3	Ay.0	Ay.1	Ay.2	Ay.3
CPU 1211C、 CPU 1212C、 CPU 1214C、 CPU 1215C (DC/DC/RLY)	CPU	-	-	-	-	-	-	-	-
	信号板	Ay.0	Ay.1	Ay.2	Ay.3	Ay.0	Ay.1	Ay.2	Ay.3

x = 板载 CPU 输出的初始字节地址（默认值为 0）

y = 信号板输出的初始字节地址（默认值为 4）

\* 如果同时使用 DC/DC/DC 型 CPU 和 DI2/DO2 信号板，则可通过板载 CPU 输出或通过信号板生成 PTO1/3 的信号。

\*\* 如果同时使用 DC/DC/DC 型 CPU 和 DO4 信号板，则可通过板载 CPU 输出或通过信号板生成 PTO1/3 或 PTO2/4 的信号。

PTO3 和 PTO4 仅适用于产品编号为 xxxxxxx-1xx31-xxxx 的 CPU。

## 说明

### 通过过程映像访问脉冲发生器输出

如果已选择 PTO (Pulse Train Output)

并将其分配给某个轴，固件将通过相应的脉冲发生器输出接管控制。

在实现上述控制功能接管后，将断开过程映像和 I/O 输出间的连接。

虽然用户可通过用户程序或监视表写入脉冲发生器输出的过程映像，

但所写入的内容不会传送到 I/O 输出。因此通过用户程序或监视表格无法监视 I/O 输出。

读取的信息反映过程映像中的值，与 I/O 输出的实际状态不一致。

对于 CPU 固件非永久使用的其它所有 CPU 输出，通常可以通过过程映像监控 I/O 输出的状态。

## 脉冲输出的频率范围

脉冲输出的频率范围如下表所示：

脉冲输出	工艺对象定位轴 V1 的频率范围	工艺对象定位轴 V2/V3 的频率范围 (CPU < V3.0)	工艺对象定位轴 V2/V3 的频率范围 (CPU V3.0)
板载 (MLFB - 产品编号 xxxxxxx-1xx30-xxxx)	$2 \text{ Hz} \leq f \leq 100 \text{ kHz}$	$2 \text{ Hz} \leq f \leq 100 \text{ kHz}$	$1 \text{ Hz} \leq f \leq 100 \text{ kHz}$
板载 (MLFB - 产品编号 xxxxxxx-1xx31xxxx)	$2 \text{ Hz} \leq f \leq 100 \text{ kHz}$ (PTO 1+2) $2 \text{ Hz} \leq f \leq 20 \text{ kHz}$ (PTO 3+4)	$2 \text{ Hz} \leq f \leq 100 \text{ kHz}$ (PTO 1+2) $2 \text{ Hz} \leq f \leq 20 \text{ kHz}$ (PTO 3+4)	$1 \text{ Hz} \leq f \leq 100 \text{ kHz}$ (PTO 1+2) $1 \text{ Hz} \leq f \leq 20 \text{ kHz}$ (PTO 3+4)
信号板 DI2/DO2 x DC24V 20kHz	$2 \text{ Hz} \leq f \leq 20 \text{ kHz}$	$2 \text{ Hz} \leq f \leq 20 \text{ kHz}$	$1 \text{ Hz} \leq f \leq 20 \text{ kHz}$
信号板 DI2/DO2 x DC24V 200kHz	$2 \text{ Hz} \leq f \leq 100 \text{ kHz}$	$2 \text{ Hz} \leq f \leq 200 \text{ kHz}$	$1 \text{ Hz} \leq f \leq 200 \text{ kHz}$
信号板 DO4 x DC24V 200kHz	$2 \text{ Hz} \leq f \leq 100 \text{ kHz}$	$2 \text{ Hz} \leq f \leq 200 \text{ kHz}$	$1 \text{ Hz} \leq f \leq 200 \text{ kHz}$
信号板 DI2/DO2 x DC5V 200kHz	$2 \text{ Hz} \leq f \leq 100 \text{ kHz}$	$2 \text{ Hz} \leq f \leq 200 \text{ kHz}$	$1 \text{ Hz} \leq f \leq 200 \text{ kHz}$
信号板 DO4 x DC5V 200kHz	$2 \text{ Hz} \leq f \leq 100 \text{ kHz}$	$2 \text{ Hz} \leq f \leq 200 \text{ kHz}$	$1 \text{ Hz} \leq f \leq 200 \text{ kHz}$

## 驱动器信号

对于运动控制，可以选择将驱动器接口设置为“驱动器启用”或“驱动器准备就绪”。  
使用驱动器接口时，可以随意选择用于驱动器启用的数字量输出和用于“驱动器准备就绪”的数字量输入。

## 加速度/减速度限值

以下限值适用于加速度和减速度：

加速度/减速度	值 (CPU < V3.0)	值 (CPU V3.0)
最小加速度/减速度	2.8E-1 个脉冲/秒 <sup>2</sup>	5.0E-3 个脉冲/秒 <sup>2</sup>
最大加速度/减速度	9.5E+9 个脉冲/秒 <sup>2</sup>	9.5E+9 个脉冲/秒 <sup>2</sup>

## 加加速度限值

以下限值适用于加加速度：

加加速度	值 (CPU < V3.0)	值 (CPU V3.0)
最小加加速度	4.0E-2 个脉冲/秒 <sup>3</sup>	4.0E-3 个脉冲/秒 <sup>3</sup>
最大加加速度	1.5E+8 个脉冲/秒 <sup>3</sup>	1.0E+10 个脉冲/秒 <sup>3</sup>

## 参见

与运动控制相关的 CPU 输出 (页 15)

## 1.11.10.2 组态对话框

### V1...3

#### 组态 - 常规 (“轴”工艺对象 V1...3)

在“常规”(General) 组态窗口中，组态“轴”工艺对象的基本属性。

#### 轴名称:

在该框中定义轴名称，或“轴”工艺对象的名称。  
该工艺对象以该名称列出在项目导航区中。

#### 硬件接口

脉冲通过固定分配的 digital 量输出输出到驱动器的动力装置。

对于具有继电器输出的

CPU，由于继电器不支持所需的开关频率，因此无法通过这些输出来输出脉冲信号。

如果要在这些 CPU 中使用 PTO (Pulse Train Output)，

必须使用具有 digital 量输出的信号板。

---

#### 说明

PTO 需要高速计数器 (HSC) 功能。CPU 版本小于 V3.0 时，可将 HSC 用于此目的。

但随后用户不能再使用该 HSC。CPU 版本大于等于 V3.0 时，将使用内部 HSC。

无法通过其输入地址来评估计数。

PTO 和 HSC 间的分配是固定的。当用户激活 PTO1 时，其连接到 HSC1。

激活 PTO2 时，其与 HSC2 相连。

---

在“脉冲发生器选择”(Pulse generator selection) 下拉列表中，

选择 PTO (Pulse Train Output)，它将提供用于控制采用脉冲接口的步进电机或伺服电机的脉冲。

如果没有在设备组态中的其它地方使用脉冲发生器和高速计数器，则系统会自动组态硬件接口。这种情况下，下拉列表中所选的 PTO 以白色背景显示。

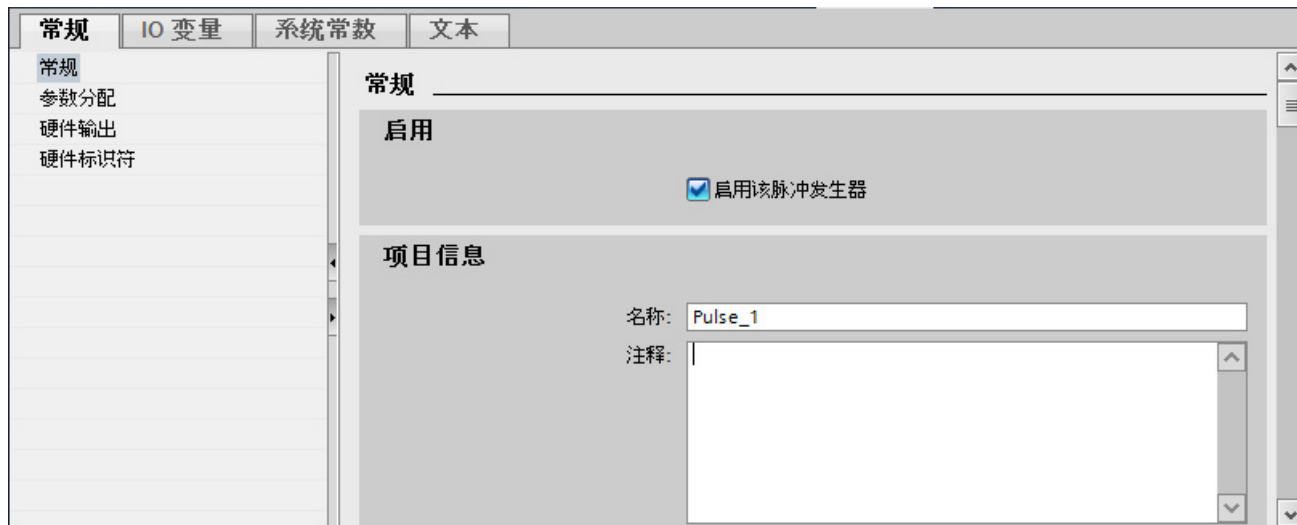
将在“输出源”(Output source)、“脉冲输出”(Pulse output)、“方向输出”(Direction output) 和“已分配高速计数器”(Assigned fast counter) 输出框中列出所用接口。

如果要更换接口或者无法自动组态 PTO (“脉冲发生器选择”(Pulse generator selection) 下拉列表中的条目突出显示为红色)，请按照以下步骤操作：

1. 单击“设备组态”(Device configuration) 按钮。

将打开脉冲发生器设备组态。

如果看不到脉冲发生器的组态，则放大设备组态的属性窗口。



2. 选中“启用该脉冲发生器”(Enable this pulse generator) 复选框。
3. 在区域导航中选择“参数分配”(Parameter assignment) 条目。

将打开“参数分配”(Parameter assignment) 窗口。





4. 在“脉冲发生器用作：”(Pulse generator as:) 下拉列表中选择“PTO”条目。
5. 在“输出源：”(Output source:) 下拉列表中，选择“集成 CPU 输出”(Integrated CPU output) 或“信号板输出”(Signal board output) 条目。只能为 PTO1 或为 PTO1 和 PTO2 选择“信号板输出”(Signal board output) 条目，具体取决于安装的信号板。  
更多详细信息，请参见以下部分：与运动控制相关的 CPU 输出 (页 15)
6. 返回到轴组态。

如果其它任何地方都未使用相应的高速计数器，则“常规”(General) 轴组态中的 PTO 框的背景不会呈现为红色。否则，根据错误消息更正组态。

## 用户单位

从下拉列表中为轴量纲系统选择所需的单位。

所选单位用于进一步组态“轴”工艺对象和显示当前轴数据。

运动控制指令的输入参数（例如，Position、Distance、Velocity 等）的值也会采用该单位。

---

### 说明

如果后来更改了该量纲系统，则在该工艺对象的所有组态窗口中可能无法相应地进行正确转换。这种情况下，请检查所有轴参数的组态。

可能需要在用户程序中根据新测量单位调整运动控制指令的输入参数的值。

---

## 参见

组态 - 常规 (页 56)

## 组态 - 回原点 (“轴”工艺对象 V1)

在“回原点”(Homing) 组态窗口中，组态主动和被动回原点参数。  
可以使用运动控制指令中的“模式”(Mode) 输入参数设置回原点方法。  
其中，Mode = 2 表示被动回原点，Mode = 3 表示主动回原点。

## 参考点开关输入

从下拉列表中为参考点开关选择数字量输入。该输入必须具有中断功能。  
板载 CPU 输入和所插入信号板输入都可以选作参考点开关的输入。

---

### 说明

默认情况下，数字量输入的滤波时间设置为 6.4 ms。  
采用默认时间的数字量输入用作参考点开关的输入时，可能引起意外减速，从而导致出现误差。根据回原点速度和参考点开关的范围，可能检测不到参考点。  
可以在数字量输入设备组态的“输入滤波器”(Input filter) 中设置滤波时间。  
指定的滤波时间必须小于参考点开关的输入信号的持续时间。

---

## 允许在到达硬限位开关后反向 (仅限主动回原点)

激活该复选框可将硬限位开关用作回原点过程中的反向凸轮。  
只有激活硬限位开关才能实现反向控制。如果使用 CPU 固件 V1.0，必须组态两个硬限位开关。如果使用 CPU 固件 V2.0 及更高版本，则仅需组态逼近方向上的硬限位开关。

如果在主动回原点过程中到达硬限位开关，轴将以组态的减速度（不是以急停减速度）制动，然后反向。然后反向检测参考点开关。

如果未激活反向功能且在主动回原点过程中轴到达硬限位开关，则将因错误而中止回原点过程并以急停减速度对轴进行制动。

---

### 说明

采用以下措施之一可确保机器在发生反向时不会行进到机械停止块：

- 保持较低的逼近速度
  - 增大组态的加速度/减速度
  - 增大硬限位开关和机械停止块间的距离
-

### 逼近/回原点方向（主动和被动回原点）

通过方向选择，可以决定主动回原点期间用于搜索参考点开关的“逼近方向”以及回原点的方向。回原点方向指定执行回原点操作时轴用于逼近组态的参考点开关端的行进方向。

有关逼近方向设置对被动回原点的影响，请参见“参考点开关”下的表格。

### 参考点开关侧（主动和被动回原点）

- 主动回原点

在此处可以选择轴在参考点开关的下端还是上端进行回原点。

#### 说明

根据轴的起始位置和组态的回原点参数，回原点过程顺序可能与组态窗口中图示的顺序不同。

- 被动回原点

对于被动回原点，必须由用户通过运动命令来执行回原点的行进运动。

回原点发生在参考点开关的哪一端取决于以下因素：

- “逼近方向”组态
- “参考点开关”组态
- 被动回原点过程中当前的行进方向

下表列出了受这些因素影响的详细信息：

影响因素：			结果：
组态 逼近方向	组态 参考点开关	当前的行进方向	回原点发生在 参考点开关
正方向	“下侧”	正方向	上侧
		负方向	下侧
正方向	“上侧”	正方向	下侧
		负方向	上侧
负方向	“下侧”	正方向	下侧
		负方向	上侧
负方向	“上侧”	正方向	上侧
		负方向	下侧

### 速度（仅限主动回原点）

在该框中，可以指定回原点期间搜索参考点开关的速度。

限值（与所选测量单位无关）：

- 启动/停止速度 ≤ 逼近速度 ≤ 最大速度

### 回原点速度（仅限主动回原点）

在该框中，可以指定回原点期间轴逼近参考点开关的速度。

限值（与所选测量单位无关）：

- 启动/停止速度 ≤ 回原点速度 ≤ 最大速度

### 起始位置偏移值（仅限主动回原点）

如果指定的起始位置与参考点开关的位置存在偏差，则可在此框中指定起始位置偏移值。

如果值不等于 0，轴回到参考点开关位置后将执行以下动作：

1. 以回原点速度使轴移动起始位置偏移值指定的一段距离
2. 达到“起始位置偏移值”时，轴处于运动控制指令“MC\_Home”的输入参数“Position”中指定的起始位置处。

限值（与所选测量单位无关）：

- $-1.0e12 \leq \text{起始位置偏移值} \leq 1.0e12$

### 起始位置

将运动控制指令“MC\_Home”中所组态的位置用作起始位置。

## 组态 - 回原点 (“轴”工艺对象 V2...3)

### 组态 - 回原点 - 常规 (轴工艺对象 V2...3)

在“回原点 - 常规”(Homing - General)

组态窗口中, 可以组态主动和被动回原点的参考点开关输入。

### 参考点开关输入

从该下拉列表中可为回原点开关选择数字量输入。该输入必须具有中断功能。板载 CPU 输入和所插入信号板输入都可选作回原点开关的输入。

---

#### 说明

默认情况下, 数字量输入的滤波时间设置为 6.4 ms。

数字量输入用作回原点开关的输入时, 可能引起意外减速, 从而导致出现误差。

根据回原点速度和回原点开关的范围, 可能检测不到回原点位置。

可以在数字量输入设备组态的“输入滤波器”(Input filter) 中设置滤波时间。

指定的滤波时间必须小于回原点开关的输入信号持续时间。

---

### 有效电平

在下拉列表中, 选择回原点时使用的回原点开关电平。

### 参见

顺序 - 主动回原点 (页 96)

### 组态 - 回原点 - 被动（轴工艺对象 V2...3）

在“归位 - 被动”(Homing - Passive) 组态窗口中，可以组态被动归位所需的参数。

被动归位的移动必须由用户触发（例如，使用轴运动命令）。

运动控制指令“MC\_Home”的输入参数“Mode”= 2 时，会启动被动归位。

### 回原点开关侧

在此处可以选择轴是在回原点开关的下端还是上端进行回原点。

### 起始位置

将运动控制指令“MC\_Home”中所组态的位置用作起始位置。

---

#### 说明

如果未使用轴运动命令进行被动归位（轴处于停止状态），则将在下一个归位开关的上升沿或下降沿处执行归位操作。

---

### 组态 - 回原点 - 主动（轴工艺对象 V2...3）

在“主动回原点”(Active homing) 组态窗口中组态主动回原点所需的参数。

运动控制指令“MC\_Home”的输入参数“Mode”= 3 时，会启动主动回原点。

## 允许在硬限位开关处自动反向

激活该复选框可将硬限位开关用作回原点过程中的反向凸轮。

只有启用硬限位开关才能实现反向控制（必须至少组态位于逼近方向上的硬限位开关）。

如果在主动回原点过程中到达硬限位开关，轴将以组态的减速度（不是以急停减速度）制动，然后反向。然后反向检测回原点开关。

如果未激活反向功能且在主动回原点过程中轴到达硬限位开关，则将因错误而中止回原点过程并以急停减速度对轴进行制动。

---

### 说明

如果可能，采用以下措施之一以确保机器在发生反向时不会行进到机械挡块：

- 保持较低的行进速度。
  - 增加组态的加速度/减速度。
  - 增加硬限位开关和机械挡块之间的距离。
- 

## 逼近/回原点方向

通过方向选择，可以决定主动回原点过程中搜索回原点开关的逼近方向以及回原点的方向。回原点方向指定执行回原点操作时轴用于逼近组态的回原点开关端的行进方向。

## 回原点开关侧

在此处可以选择轴是在回原点开关的下端还是上端进行回原点。

## 速度

在该框中，可以指定回原点期间搜索回原点开关的速度。

限值（与所选测量单位无关）：

- 启动/停止速度  $\leq$  逼近速度  $\leq$  最大速度

## 回原点速度

在该框中，可以指定回原点时逼近参考点开关的速度。

限值（与所选测量单位无关）：

- 启动/停止速度  $\leq$  回原点速度  $\leq$  最大速度

### 起始位置偏移值

如果所需的回原点位置与回原点开关的位置存在偏差，则可在此框中指定回原点位置偏移量。

如果该值不等于 0，轴在回原点开关处回原点后将执行以下动作：

1. 以回原点速度使轴移动起始位置偏移值指定的一段距离
2. 达到“起始位置偏移值”时，轴处于运动控制指令“MC\_Home”的输入参数“Position”中指定的起始位置处。

限值（与所选测量单位无关）：

- $-1.0e12 \leq \text{起始位置偏移值} \leq 1.0e12$

### 起始位置

将运动控制指令“MC\_Home”中所组态的位置用作起始位置。



## 在用户程序中更改动态值的组态 (“轴”工艺对象 V1...3)

可以在 CPU 中运行用户程序期间更改下列组态参数:

### 加速度和减速度

在用户程序运行期间, 还可以更改加速度和减速度值。  
要执行该操作, 可使用以下工艺对象变量:

- <轴名称>.Config.DynamicDefaults.Acceleration  
用于更改加速度
- <轴名称>.Config.DynamicDefaults.Deceleration  
用于更改减速度

有关组态参数的更改何时生效的信息, 请参见附录中的工艺对象变量 (页 342)说明。

### 急停减速度

在用户程序运行期间, 还可更改急停减速度值。  
要执行该操作, 可使用以下工艺对象变量:

- <轴名称>.Config.DynamicDefaults.EmergencyDeceleration

有关组态参数的更改何时生效的信息, 请参见附录中的工艺对象变量说明。

<b>注意</b>
-----------

更改该参数后, 可能需要调整硬限位开关的位置以及其它安全相关的设置。
------------------------------------

### 冲击限制（从 V2.0“轴”工艺对象起）

还可在用户程序运行时激活和禁用冲击限制和更改冲击值。

要执行该操作，可使用以下工艺对象变量：

- `<轴名称>.Config.DynamicDefaults.JerkActive`

用于激活和禁用冲击限制

- `<轴名称>.Config.DynamicDefaults.Jerk`

用于更改冲击

有关组态参数的更改何时生效的信息，请参见附录中的工艺对象变量说明。

### 参见

在用户程序中可以更改动态组态 (页 89)

## V4

### 组态 - 常规（自 V4 起的定位轴工艺对象）

在“常规”(General) 组态窗口中，组态定位轴工艺对象的基本属性。

### 轴名称

在该域中定义轴名称或定位轴工艺对象的名称。该工艺对象以该名称列出在项目树中。

### 硬件接口

脉冲通过固定分配的 digital 量输出输出到驱动器的动力装置。

对于具有继电器输出的

CPU，由于继电器不支持所需的开关频率，因此无法通过这些输出来输出脉冲信号。

如果要在这些 CPU 中使用 PTO (Pulse Train Output)，

必须使用具有 digital 量输出的信号板。

---

#### 说明

PTO 需要高速计数器 (HSC) 的功能。为此将使用内部 HSC，不能对此 HSC 的计数进行评估。

---

### 选择脉冲发生器

在该下拉列表中，选择 PTO (Pulse Train

Output)，它将提供用于控制采用脉冲接口的步进电机或伺服电机的脉冲。

如果没有在设备组态中的其它地方使用脉冲发生器和高速计数器，则系统会自动组态硬件接口。这种情况下，下拉列表中所选的 PTO 以白色背景显示。

### “设备组态”(Device configuration) 按钮

该按钮可转到 CPU 设备组态中的脉冲选项设置。

## 信号类型

从下拉列表中选择所需的信号类型。可以使用以下信号类型：

- **PTO（脉冲 A 和方向 B）**

使用一个脉冲输出和一个方向输出控制步进电机。

- **PTO（向上计数 A，向下计数 B）**

分别使用一个正向和负向运动的脉冲输出控制步进电机。

- **PTO（A/B 相位偏移量）**

A 相和 B 相的两个脉冲输出在同一频率下运行。

在驱动器步进结束时评估这两个脉冲输出的周期。

A 相和 B 相之间的相位偏移量决定了运动方向。

- **PTO（A/B 相位偏移量 - 四重）**

A 相和 B 相的两个脉冲输出在同一频率下运行。

在驱动器步进结束时评估 A 相和 B 相的所有上升沿和下降沿。

A 相和 B 相之间的相位偏移量决定了运动方向。

### 脉冲输出（信号类型“PTO（脉冲 A 和方向 B）”）

在此域中选择需要用作脉冲输出的输出。

可以通过符号地址或将其分配给绝对地址来选择输出。

### 激活方向输出（信号类型“PTO（脉冲 A 和方向 B）”）

在“脉冲和方向”模式下可以取消激活和激活方向输出。利用此选项可以限制运动方向。

### 方向输出（信号类型“PTO（脉冲 A 和方向 B）”）

在此域中选择需要用作方向输出的输出。

可以通过符号地址或将其分配给绝对地址来选择输出。

### 向上脉冲输出（信号类型“PTO（向上计数 A，向下计数 B）”）

在此域中选择用于实现正向运动所需的脉冲输出。

可以通过符号地址或将其分配给绝对地址来选择输出。

**向下脉冲输出（信号类型“PTO（向上计数 A，向下计数 B）”）**

在此域中选择用于实现负向运动所需的脉冲输出。

可以通过符号地址或将其分配给绝对地址来选择输出。

**A 相（信号类型“PTO（A/B 相位偏移量）”和“PTO（A/B 相位偏移量 - 四重）”）**

在此域中为 A 相信号选择所需的脉冲输出。

可以通过符号地址或将其分配给绝对地址来选择输出。

**B 相（信号类型“PTO（A/B 相位偏移量）”和“PTO（A/B 相位偏移量 - 四重）”）**

在此域中为 B 相信号选择所需的脉冲输出。

可以通过符号地址或将其分配给绝对地址来选择输出。

**用户单位**

从下拉列表中为轴量纲系统选择所需的单位。

所选单位将用于进一步组态定位轴工艺对象和显示当前轴数据。

运动控制指令的输入参数（例如，Position、Distance、Velocity 等）的值也会采用该单位。

---

**说明**

如果后来更改了该量纲系统，则在该工艺对象的所有组态窗口中可能无法相应地进行正确转换。 这种情况下，请检查所有轴参数的组态。

可能需要在用户程序中根据新测量单位调整运动控制指令的输入参数的值。

---

### 组态 - 驱动器信号（定位轴工艺对象 V4）

在“驱动器信号”(Drive signals)

组态窗口中组态驱动器使能信号的输出以及驱动器的“驱动器准备就绪”(Drive ready) 反馈信号的输入。

驱动器使能信号由运动控制指令“MC\_Power”控制，可以启用对驱动器的供电。信号通过组态的输出提供给驱动器。

如果驱动器在接收到驱动器使能信号之后准备好开始进行行进，则驱动器会向 CPU 发送“驱动器准备就绪”(Drive ready) 信号。“驱动器准备就绪”信号通过组态的输入传送回 CPU。

如果驱动器不包含任何这一类型的接口，则无需组态这些参数。这种情况下，为准备就绪输入选择值 TRUE。

### 参见

组态 - 机械（定位轴工艺对象 V4）(页 311)

位置限制 (页 77)

动态 (页 82)

回原点（自定位轴工艺对象 V2 起）(页 91)

## 组态 - 机械（定位轴工艺对象 V4）

在“机械”(Mechanics) 组态窗口中组态驱动器的机械属性。

### 电机每转的增量

在此框中组态电机每转所需的脉冲数。

限值（与所选测量单位无关）：

- $0 < \text{电机每转的脉冲数} \leq 2147483647$

### 每转的距离

在此框中，组态电机每转带动单元的机械系统行进的负载距离。

限值（与所选测量单位无关）：

- $0.0 < \text{每转距离} \leq 1.0\text{e}12$

### 允许的旋转方向（自 V4 工艺版本起）

通过组态此框可决定系统机械是同时朝两个方向运动，还是只朝正向或负向运动。

如果尚未在“PTO（脉冲 A 和方向

B）”模式下激活脉冲发生器的方向输出，则选择受限于正方向或负方向。

## 反转方向

可使用“反转方向”(Invert direction) 复选框根据驱动器的方向逻辑对控制系统进行调整。

方向逻辑将根据所选脉冲发生器的模式反转：

- **PTO (脉冲 A 和方向 B)**

- 方向输出上输出 0 V ⇒ 正向旋转
- 方向输出上输出 5 V/24 V ⇒ 负向旋转

指定的电压取决于所使用的硬件。各指示值并不适用于 CPU 1217 的差分输出。

- **PTO (向上计数 A, 向下计数 B)**

“向下脉冲输出”和“向上脉冲输出”这两个输出相互交换。

- **PTO (A/B 相位偏移量)**

“A 相”和“B 相”输出相互交换。

- **PTO (A/B 相位偏移量 - 四重)**

“A 相”和“B 相”输出相互交换。

## 参见

组态 - 驱动器信号 (定位轴工艺对象 V4) (页 310)

位置限制 (页 77)

动态 (页 82)

回原点 (自定位轴工艺对象 V2 起) (页 91)

信号类型与行进方向之间的关系 (页 20)



## 组态 - 归位 - 被动（定位轴工艺对象 V4）

在“归位 - 被动”(Homing - Passive) 组态窗口中，可以组态被动归位所需的参数。

被动归位的移动必须由用户触发（例如，使用轴运动命令）。

运动控制指令“MC\_Home”的输入参数“Mode”= 2 时，会启动被动归位。

## 回原点开关输入

从该下拉列表中可为回原点开关选择数字量输入。该输入必须具有中断功能。板载 CPU 输入和所插入信号板输入都可选作回原点开关的输入。

---

### 说明

默认情况下，数字量输入的滤波时间设置为 6.4 ms。

数字量输入用作回原点开关的输入时，可能引起意外减速，从而导致出现误差。

根据回原点速度和回原点开关的范围，可能检测不到回原点位置。

可以在数字量输入设备组态的“输入滤波器”(Input filter) 中设置滤波时间。

指定的滤波时间必须小于回原点开关的输入信号持续时间。

---

## 有效电平

在下拉列表中，选择回原点时使用的回原点开关电平。

## 回原点开关侧

在此处可以选择轴是在回原点开关的下端还是上端进行回原点。

## 起始位置

将运动控制指令“MC\_Home”中所组态的位置用作起始位置。

---

### 说明

如果未使用轴运动命令进行被动归位（轴处于停止状态），则将在下一个归位开关的上升沿或下降沿处执行归位操作。

---

## 组态 - 归位 - 主动（定位轴工艺对象 V4）

在“主动回原点”(Active homing) 组态窗口中组态主动回原点所需的参数。  
运动控制指令“MC\_Home”的输入参数“Mode”= 3 时，会启动主动回原点。

## 回原点开关输入

从该下拉列表中可为回原点开关选择数字量输入。该输入必须具有中断功能。板载 CPU 输入和所插入信号板输入都可选作回原点开关的输入。

---

### 说明

默认情况下，数字量输入的滤波时间设置为 6.4 ms。

数字量输入用作回原点开关的输入时，可能引起意外减速，从而导致出现误差。根据回原点速度和回原点开关的范围，可能检测不到回原点位置。

可以在数字量输入设备组态的“输入滤波器”(Input filter) 中设置滤波时间。

指定的滤波时间必须小于回原点开关的输入信号持续时间。

---

## 有效电平

在下拉列表中，选择回原点时使用的回原点开关电平。

## 启用硬限位开关处反向

激活该复选框可将硬限位开关用作回原点过程中的反向凸轮。

只有启用硬限位开关才能实现反向控制（必须至少组态位于逼近方向上的硬限位开关）。

如果在主动回原点过程中到达硬限位开关，轴将以组态的减速度（不是以急停减速度）制动，然后反向。然后反向检测回原点开关。

如果未激活反向功能且在主动回原点过程中轴到达硬限位开关，则将因错误而中止回原点过程并以急停减速度对轴进行制动。

---

### 说明

如果可能，采用以下措施之一以确保机器在发生反向时不会行进到机械挡块：

- 保持较低的行进速度。
  - 增加组态的加速度/减速度。
  - 增加硬限位开关和机械挡块之间的距离。
-

## 逼近/回原点方向

通过方向选择，可以决定主动回原点过程中搜索回原点开关的逼近方向以及回原点的方向。回原点方向指定执行回原点操作时轴用于逼近组态的回原点开关端的行进方向。

## 回原点开关侧

在此处可以选择轴是在回原点开关的下端还是上端进行回原点。

## 逼近速度

在该框中，可以指定回原点期间搜索回原点开关的速度。

限值（与所选测量单位无关）：

- 启动/停止速度 ≤ 逼近速度 ≤ 最大速度

## 回原点速度

在该框中，可以指定回原点期间逼近回原点开关的速度。

限值（与所选测量单位无关）：

- 启动/停止速度 ≤ 回原点速度 ≤ 最大速度

## 起始位置偏移值

如果所需的回原点位置与回原点开关的位置存在偏差，则可在该框中指定回原点位置偏移量。

如果该值不等于 0，轴在回原点开关处回原点后将执行以下动作：

1. 以回原点速度使轴移动起始位置偏移值指定的一段距离
2. 达到“起始位置偏移值”时，轴处于运动控制指令“MC\_Home”的输入参数“Position”中指定的起始位置处。

限值（与所选测量单位无关）：

- $-1.0e12 \leq \text{起始位置偏移值} \leq 1.0e12$

## 起始位置

将运动控制指令“MC\_Home”中所组态的位置用作起始位置。

## 1.11.10.3 诊断 - 状态和错误位 (“轴”工艺对象 V1...3)

在 TIA Portal 中使用诊断功能“状态和错误位”(Status and error bits)

可监视轴的最重要状态和错误消息。

当轴激活时，可以在“手动控制”模式和“自动控制”模式下在线显示诊断功能。

状态错误消息具有下列含义：

## 轴的状态

状态	说明
已启用	轴已启用且准备就绪，可通过运动控制命令进行控制。 (工艺对象的变量: <轴名称>.StatusBits.Enable)
已回原点	轴已回原点，可执行运动控制指令“MC_MoveAbsolute”的绝对定位命令。 对于相对定位而言，轴不必回原点。特殊情况： <ul style="list-style-type: none"> <li>主动回原点过程中，该状态为 FALSE。</li> <li>如果回原点的轴经受被动回原点，则在被动回原点过程中该状态设置为 TURE。</li> </ul> (工艺对象的变量: <轴名称>.StatusBits.HomingDone)
轴错误	“轴”工艺对象发生错误。 有关错误的详细信息，请参见自动控制模式下的运动控制指令的 ErrorID 和 ErrorInfo 参数。在手动模式下，轴控制面板的“错误消息”(Error message) 框可显示有关错误原因的详细信息。 (工艺对象的变量: <轴名称>.StatusBits.Error)
轴控制面板已启用	在轴控制面板中启用了“手动控制”(Manual control) 模式。 轴控制面板对“轴”工艺对象具有优先控制权。不能通过用户程序来控制轴。 (工艺对象的变量: <轴名称>.StatusBits.ControlPanelActive)
必须重新启动	已在 CPU RUN 操作模式下将已修改的轴组态下载到装载存储器。 要将已修改组态下载到工作存储器，需要重新启动轴。使用运动控制指令 MC_Reset 执行此操作。

## 驱动器状态

状态	说明
驱动器准备就绪	驱动器准备好运行。 (工艺对象的变量: <轴名称>.StatusBits.DriveReady)
驱动器错误	驱动器因丢失“驱动器就绪”信号而报告错误。 (工艺对象的变量: <轴名称>.ErrorBits.DriveFault)

## 轴运动的状态

状态	说明
停止	轴处于停止状态。 (工艺对象的变量: <轴名称>.StatusBits.StandStill)
加速度	轴在加速。 (工艺对象的变量: <轴名称>.StatusBits.Acceleration)
恒速	轴在恒速运转。 (工艺对象的变量: <轴名称>.StatusBits.ConstantVelocity)
减速度	轴在减速 (速度下降)。 (工艺对象的变量: <轴名称>.StatusBits.Deceleration)

## 运动模式的状态

状态	说明
定位	轴执行运动控制指令“MC_MoveAbsolute”、“MC_MoveRelative”或者轴控制面板的定位命令。 (工艺对象的变量: <轴名称>.StatusBits.PositioningCommand)
以速度设定值行进	轴以运动控制指令“MC_MoveVelocity”、“MC_MoveJog”或者轴控制面板的速度设定值执行命令。 (工艺对象的变量: <轴名称>.StatusBits.SpeedCommand)
回原点	轴执行运动控制指令“MC_Home”或者轴控制面板的回原点命令。 (工艺对象的变量: <轴名称>.StatusBits.Homing)
命令表已激活 (从 V2.0 轴工艺对象起)	该轴由运动控制指令“MC_CommandTable”控制。 (工艺对象的变量: <轴名称>.StatusBits.CommandTableActive)

## 错误消息

错误	说明
已到达下限软限位开关	已到达下限软限位开关。 (工艺对象的变量: <轴名称>.ErrorBits.SwLimitMinReached)
已超出下限软限位开关	已超出下限软限位开关。 (工艺对象的变量: <轴名称>.ErrorBits.SwLimitMinExceeded)
已到达上限软限位开关	已到达上限软限位开关。 (工艺对象的变量: <轴名称>.ErrorBits.SwLimitMaxReached)
已超出上限软限位开关	已超出上限软限位开关。 (工艺对象的变量: <轴名称>.ErrorBits.SwLimitMaxExceeded)
已到达下限硬限位开关	已到达下限硬限位开关。 (工艺对象的变量: <轴名称>.ErrorBits.HwLimitMin)
已到达上限硬限位开关	已到达上限硬限位开关。 (工艺对象的变量: <轴名称>.ErrorBits.HwLimitMax)
PTO 和 HSC 在使用中	另一个轴正在使用此 PTO (Pulse Train Output) 和 HSC (High Speed Counter) 并且该轴已通过“MC_Power”启用。 (工艺对象的变量: <轴名称>.ErrorBits.HwUsed)
组态错误	错误地组态了“轴”工艺对象, 或者在用户程序运行期间错误地修改了可编辑的组态数据。 (工艺对象的变量: <轴名称>.ErrorBits.ConfigFault)
内部错误	发生内部错误。 (工艺对象的变量: <轴名称>.ErrorBits.SystemFault)

## 参见

状态和错误位 (自工艺对象 V4 起) (页 183)

## 1.11.10.4 ErrorID 和 ErrorInfo

## ErrorID 和 ErrorInfo 列表（工艺对象 V2...3）

下表列出了运动控制指令中指示的所有 ErrorID 和 ErrorInfo。  
除错误原因外，还列出了清除错误的解决方法：

## 伴随轴停止的运行错误

ErrorID	ErrorInfo	说明	解决方法
<b>16#8000</b>		<b>驱动器错误，丢失“驱动器就绪”信号</b>	
	16#0001	-	使用指令“MC_Reset”确认错误；提供驱动器信号；根据需要重新启动命令
<b>16#8001</b>		<b>已触发下限软限位开关</b>	
	16#000E	已经以当前组态的减速度到达下限软限位开关的位置	使用“MC_Reset”指令确认错误；使用运动命令使轴沿正向移动，超出软限位开关的范围
	16#000F	已经以急停减速度到达下限软限位开关的位置	
	16#0010	已经以急停减速度超出下限软限位开关的位置	
<b>16#8002</b>		<b>已触发上限软限位开关</b>	
	16#000E	已经以当前组态的减速度到达上限软限位开关的位置	使用“MC_Reset”指令确认错误；使用运动命令使轴沿负向移动，超出软限位开关的范围
	16#000F	已经以急停减速度到达上限软限位开关的位置	
	16#0010	已经以急停减速度超出上限软限位开关的位置	
<b>16#8003</b>		<b>已到达下限硬限位开关</b>	
	16#000E	已到达下限硬限位开关。 轴以急停减速度停止。 (主动回原点期间，未找到参考点开关)	使用指令“MC_Reset”确认已启用轴的错误；使用运动命令使轴沿正向移动，超出硬限位开关的范围。



ErrorID	ErrorInfo	说明	解决方法
<b>16#8004</b>		<b>已到达上限硬限位开关</b>	
	16#000E	已到达上限硬限位开关。 轴以急停减速度停止。 (主动回原点期间, 未找到参考点开关)	使用指令“MC_Reset”确认已启用轴的错误; 使用运动命令使轴沿负向移动, 超出硬限位开关的范围。
<b>16#8005</b>		<b>PTO/HSC 已被另一个轴占用</b>	
	16#0001	-	<b>轴未正确组态:</b> 更正 PTO (Pulse Train Output)/HSC (High Speed Counter) 的组态并将其下载到控制器
			<b>多个轴将通过一个 PTO 运行:</b> 另一个轴正在使用 PTO/HSC。 如果要控制当前轴, 则必须使用“MC_Power”(Enable = FALSE) 禁用另一个轴。 (另请参见将多个轴与同一个 PTO 一起使用 (页 190))
<b>16#8006</b>		<b>控制面板中发生通信错误</b>	
	16#0012	已超时	检查电缆连接, 然后再次按下“手动控制”按钮。
<b>16#8007</b>		<b>该轴无法启用。</b>	
	16#0025	正在重新启动	请等待, 直到轴重新启动完成。
	16#0026	正在 RUN 模式下执行加载过程	请等待, 直到加载过程完成。

## 不伴随轴停止的运行错误

ErrorID	ErrorInfo	说明	解决方法
<b>16#8200</b>		轴未启用	
	16#0001	-	启用轴；重新启动命令
<b>16#8201</b>		轴已由另一个“MC_Power”实例启用	
	16#0001	-	仅通过一个“MC_Power”实例启用轴
<b>16#8202</b>		已超出并行激活的运动控制命令数的最大值（所有运动控制工艺对象最多可具有 200 条命令）	
	16#0001	-	减少并行激活命令数；重新启动命令 如果运动控制指令的参数 “Busy”= TRUE，则说明命令已激活。
<b>16#8203</b>		轴当前在“手动控制”(Manual control) 模式（轴控制面板）下运行	
	16#0001	-	退出“手动控制”；重新启动命令
<b>16#8204</b>		轴未回原点	
	16#0001	-	使用指令“MC_Home”使轴回原点； 重新启动命令
<b>16#8205</b>		轴当前由用户程序控制（该错误仅显示在轴控制面板中）	
	16#0013	轴已在用户程序中启用。	使用指令“MC_Power”禁用轴或者在轴控制面板中再次选择“手动控制”(Manual control)
<b>16#8206</b>		工艺对象尚未激活	
	16#0001	-	使用指令“MC_Power”Enable = TRUE 启用轴或者在轴控制面板中启用轴。
<b>16#8207</b>		命令被拒绝	
	16#0016	正在执行主动回原点；无法启动另一种回原点方法。	等到主动回原点完成或通过运动命令（例如，“MC_Halt”）中止主动回原点。
	16#0018	当轴正在主动或被动回原点时，不能使用命令表进行移动。	一直等到直接地或被动回原点完成。
	16#0019	正在处理命令表时，轴不能主动或被动回原点。	等到命令表完成或通过运动命令（例如，“MC_Halt”）中止命令表。

ErrorID	ErrorInfo	说明	解决方法
<b>16#8208</b>		<b>最大速度和启动/停止速度的差值无效</b>	
	16#0002	值的数字格式无效	更正该值；重新启动命令
	16#000A	值小于或等于 0	
<b>16#8209</b>		<b>对工艺对象“轴”的加速度无效</b>	
	16#0002	值的数字格式无效	更正该值；重新启动命令
	16#000A	值小于或等于 0	
<b>16#820A</b>		<b>无法重新启动轴</b>	
	16#0013	轴已在用户程序中启用。	使用“MC_Power”指令阻止轴；再次重新启动
	16#0027	轴当前在“手动控制”模式（轴控制面板）下运行	退出“手动控制”；再次重新启动
<b>16#820B</b>		<b>无法执行命令表</b>	
	16#0026	正在 RUN 模式下执行加载过程	请等待，直到加载过程完成。

## 块参数错误

ErrorID	ErrorInfo	说明	解决方法
<b>16#8400</b>		<b>运动控制指令的“Position”参数值无效</b>	
	16#0002	值的数字格式无效	更正该值；重新启动命令
	16#0005	该值超出取值范围（大于 $1e^{12}$ ）	
	16#0006	该值超出取值范围（小于 $-1e^{12}$ ）	
<b>16#8401</b>		<b>运动控制指令的“Distance”参数值无效</b>	
	16#0002	值的数字格式无效	更正该值；重新启动命令
	16#0005	该值超出取值范围（大于 $1e^{12}$ ）	
	16#0006	该值超出取值范围（小于 $-1e^{12}$ ）	
<b>16#8402</b>		<b>运动控制指令的“Velocity”参数值无效</b>	
	16#0002	值的数字格式无效	更正该值；重新启动命令
	16#0008	值大于组态的最大速度	
	16#0009	值小于组态的启动/停止速度	
	16#0024	值小于 0	
<b>16#8403</b>		<b>运动控制指令的“Direction”参数值无效</b>	
	16#0011	所选值无效	更正所选值；重新启动命令
<b>16#8404</b>		<b>运动控制指令的“Mode”参数值无效</b>	
	16#0011	所选值无效	更正所选值；重新启动命令
	16#0015	未组态主动/被动回原点	更正组态并将其下载到控制器；启用轴并重新启动命令
	16#0017	虽然硬限位开关已禁用，但还是激活了在硬限位开关处反向	<ul style="list-style-type: none"> <li>使用变量 &lt;轴&gt;.Config.PositionLimits_HW. Active = TRUE 激活硬限位开关，并重新启动该命令</li> <li>更正组态并将其下载到控制器；启用轴并重新启动命令</li> </ul>
<b>16#8405</b>		<b>运动控制指令的“StopMode”参数值无效</b>	
	16#0011	所选值无效	更正所选值；再次启用轴

ErrorID	ErrorInfo	说明	解决方法
<b>16#8406</b>		不允许同时正向和反向点动	
	16#0001	-	采取措施确保参数“JogForward”和“JogBackward”的信号状态不会同时为 TRUE；重新启动命令。
<b>16#8407</b>		只有禁用激活的轴后，才允许使用指令“MC_Power”切换到另一个轴。	
	16#0001	-	禁用激活的轴；然后可以切换到其它轴并启用该轴。
<b>16#8408</b>		运动控制指令的“Axis”参数值无效	
	16#001A	指定的值与所需的工艺对象版本不匹配	更正该值；重新启动命令
	16#001B	指定的值与所需的工艺对象类型不匹配	
	16#001C	指定的值不是运动控制工艺数据块	
<b>16#8409</b>		运动控制指令的“CommandTable”参数值无效	
	16#001A	指定的值与所需的工艺对象版本不匹配	更正该值；重新启动命令
	16#001B	指定的值与所需的工艺对象类型不匹配	
	16#001C	指定的值不是运动控制工艺数据块	
<b>16#840A</b>		运动控制指令的“StartStep”参数值无效	
	16#000A	值小于或等于 0	更正该值；重新启动命令
	16#001D	开始步进大于结束步进	
	16#001E	值大于 32	
<b>16#840B</b>		运动控制指令的“EndStep”参数值无效	
	16#000A	值小于或等于 0	更正该值；重新启动命令
	16#001E	值大于 32	
<b>16#840C</b>		运动控制指令的“RampUpTime”参数值无效	
	16#0002	值的数字格式无效	更正该值；重新启动命令
	16#000A	值小于或等于 0	

ErrorID	ErrorInfo	说明	解决方法
<b>16#840D</b>		运动控制指令的“RampDownTime”参数值无效	
	16#0002	值的数字格式无效	更正该值；重新启动命令
	16#000A	值小于或等于 0	
<b>16#840E</b>		运动控制指令的“EmergencyRampTime”参数值无效	
	16#0002	值的数字格式无效	更正该值；重新启动命令
	16#000A	值小于或等于 0	
<b>16#840F</b>		运动控制指令的“JerkTime”参数值无效	
	16#0002	值的数字格式无效	更正该值；重新启动命令
	16#000A	值小于或等于 0	

## 轴的组态错误

ErrorID	ErrorInfo	说明	解决方法
<b>16#8600</b>		<b>脉冲发生器 (PTO) 的参数分配无效</b>	
	16#000B	地址无效	<ul style="list-style-type: none"> <li>将无错误组态下载到控制器；</li> <li>使用指令“MC_Power”再次启用轴</li> </ul>
	16#0014	所选硬件由另一应用程序使用	
<b>16#8601</b>		<b>高速计数器 (HSC) 的参数分配无效</b>	
	16#000B	地址无效	<ul style="list-style-type: none"> <li>将无错误组态下载到控制器；</li> <li>使用指令“MC_Power”再次启用轴</li> </ul>
	16#0014	所选硬件由另一应用程序使用	
<b>16#8602</b>		<b>“使能输出”(Enable output) 的参数分配无效</b>	
	16#000B	地址无效	<ul style="list-style-type: none"> <li>将无错误组态下载到控制器；</li> <li>使用指令“MC_Power”再次启用轴</li> </ul>
<b>16#8603</b>		<b>“输入就绪”(Ready input) 的参数分配无效</b>	
	16#000B	地址无效	<ul style="list-style-type: none"> <li>将无错误组态下载到控制器；</li> <li>使用指令“MC_Power”再次启用轴</li> </ul>
<b>16#8604</b>		<b>“电机每转的脉冲数”(Pulses per motor revolution) 值无效</b>	
	16#000A	值小于或等于零	<ul style="list-style-type: none"> <li>将无错误组态下载到控制器；</li> <li>使用指令“MC_Power”再次启用轴</li> </ul>
<b>16#8605</b>		<b>“每转的距离”(Distance per revolution) 值无效</b>	
	16#0002	值的数字格式无效	<ul style="list-style-type: none"> <li>将无错误组态下载到控制器；</li> <li>使用指令“MC_Power”再次启用轴</li> </ul>
	16#0005	该值超出取值范围（大于 $1e^{12}$ ）	
	16#000A	值小于或等于零	
<b>16#8606</b>		<b>“启动/停止速度”(Start / stop velocity) 值无效</b>	
	16#0002	值的数字格式无效	<ul style="list-style-type: none"> <li>将无错误组态下载到控制器；</li> <li>使用指令“MC_Power”再次启用轴</li> </ul>
	16#0003	值超出硬限位上限	
	16#0004	值低于硬限位下限	
	16#0007	启动/停止速度大于最大速度	
<b>16#8607</b>		<b>“最大速度”(maximum velocity) 值无效</b>	
	16#0002	值的数字格式无效	<ul style="list-style-type: none"> <li>将无错误组态下载到控制器；</li> <li>使用指令“MC_Power”再次启用轴</li> </ul>
	16#0003	值超出硬限位上限	
	16#0004	值低于硬限位下限	

ErrorID	ErrorInfo	说明	解决方法
<b>16#8608</b>		<b>“加速度”(Acceleration) 值无效</b>	
	16#0002	值的数字格式无效	<ul style="list-style-type: none"> <li>将无错误组态下载到控制器；使用指令“MC_Power”再次启用轴</li> <li>在线更正不正确的值；使用指令“MC_Reset”确认错误并视需要重新启动命令</li> </ul>
	16#0003	值超出硬限位上限	
	16#0004	值低于硬限位下限	
<b>16#8609</b>		<b>“减速度”(Deceleration) 值无效</b>	
	16#0002	值的数字格式无效	<ul style="list-style-type: none"> <li>将无错误组态下载到控制器；使用指令“MC_Power”再次启用轴</li> <li>在线更正不正确的值；使用指令“MC_Reset”确认错误并视需要重新启动命令</li> </ul>
	16#0003	值超出硬限位上限	
	16#0004	值低于硬限位下限	
<b>16#860A</b>		<b>“急停减速度”(Emergency stop deceleration) 值无效</b>	
	16#0002	值的数字格式无效	<ul style="list-style-type: none"> <li>将无错误组态下载到控制器；使用指令“MC_Power”再次启用轴</li> <li>在线更正不正确的值；使用指令“MC_Reset”确认错误并视需要重新启动命令</li> </ul>
	16#0003	值超出硬限位上限	
	16#0004	值低于硬限位下限	
<b>16#860B</b>		<b>下限软限位开关的位置值无效</b>	
	16#0002	值的数字格式无效	<ul style="list-style-type: none"> <li>将无错误组态下载到控制器；使用指令“MC_Power”再次启用轴</li> <li>在线更正不正确的值；使用指令“MC_Reset”确认错误并视需要重新启动命令</li> </ul>
	16#0005	该值超出取值范围（大于 $1e^{12}$ ）	
	16#0006	该值超出取值范围（小于 $-1e^{12}$ ）	
	16#0007	下限软限位开关的位置值大于上限软限位开关的位置值	
<b>16#860C</b>		<b>上限软限位开关的位置值无效</b>	
	16#0002	值的数字格式无效	<ul style="list-style-type: none"> <li>将无错误组态下载到控制器；使用指令“MC_Power”再次启用轴</li> <li>在线更正不正确的值；使用指令“MC_Reset”确认错误并视需要重新启动命令</li> </ul>
	16#0005	该值超出取值范围（大于 $1e^{12}$ ）	
	16#0006	该值超出取值范围（小于 $-1e^{12}$ ）	



ErrorID	ErrorInfo	说明	解决方法
<b>16#860D</b>		<b>下限硬限位开关的地址无效</b>	
	16#000C	下降沿的地址无效	<ul style="list-style-type: none"> <li>将无错误组态下载到控制器；使用指令“MC_Power”再次启用轴</li> </ul>
	16#000D	上升沿的地址无效	
<b>16#860E</b>		<b>上限硬限位开关的地址无效</b>	
	16#000C	下降沿的地址无效	<ul style="list-style-type: none"> <li>将无错误组态下载到控制器；使用指令“MC_Power”再次启用轴</li> </ul>
	16#000D	上升沿的地址无效	
<b>16#860F</b>		<b>“起始位置偏移值”(home position offset) 无效</b>	
	16#0002	值的数字格式无效	<ul style="list-style-type: none"> <li>将无错误组态下载到控制器；使用指令“MC_Power”再次启用轴</li> <li>在线更正不正确的值；使用指令“MC_Reset”确认错误并视需要重新启动命令</li> </ul>
	16#0005	该值超出取值范围（大于 $1e^{12}$ ）	
	16#0006	该值超出取值范围（小于 $-1e^{12}$ ）	
<b>16#8610</b>		<b>“逼近速度”(approach velocity) 值无效</b>	
	16#0002	值的数字格式无效	<ul style="list-style-type: none"> <li>将无错误组态下载到控制器；使用指令“MC_Power”再次启用轴</li> <li>在线更正不正确的值；使用指令“MC_Reset”确认错误并视需要重新启动命令</li> </ul>
	16#0008	速度大于最大速度	
	16#0009	速度小于启动/停止速度	
<b>16#8611</b>		<b>“回原点速度”(Homing velocity) 值无效</b>	
	16#0002	值的数字格式无效	<ul style="list-style-type: none"> <li>将无错误组态下载到控制器；使用指令“MC_Power”再次启用轴</li> <li>在线更正不正确的值；使用指令“MC_Reset”确认错误并视需要重新启动命令</li> </ul>
	16#0008	速度大于最大速度	
	16#0009	速度小于启动/停止速度	
<b>16#8612</b>		<b>参考点开关的地址无效</b>	
	16#000C	下降沿的地址无效	<ul style="list-style-type: none"> <li>将无错误组态下载到控制器；使用指令“MC_Power”再次启用轴</li> </ul>
	16#000D	上升沿的地址无效	

ErrorID	ErrorInfo	说明	解决方法
16#8613		虽然未组态硬限位开关，但还是在主动回原点过程中， 激活了在该硬限位开关处反向	
	16#0001	-	<ul style="list-style-type: none"> <li>将无错误组态下载到控制器； 使用指令“MC_Power”再次启用轴</li> <li>在线更正不正确的值；使用指令 “MC_Reset”确认错误并视需要重新 启动命令</li> </ul>
16#8614		“冲击”(Jerk) 值无效	
	16#0002	值的数字格式无效	<ul style="list-style-type: none"> <li>将无错误组态下载到控制器； 使用指令“MC_Power”再次启用轴</li> <li>在线更正不正确的值；使用指令 “MC_Reset”确认错误并视需要重新 启动命令</li> </ul>
	16#001F	值大于最大冲击值	
	16#0020	值小于最小冲击值	
16#8615		“测量单位”的值无效	
	16#0011	所选值无效	将无错误组态下载到控制器；使用指令 “MC_Power”再次启用轴

## 命令表的组态错误

ErrorID	ErrorInfo	说明	解决方法
<b>16#8700</b>		<b>命令表中“命令类型”(Command type) 的值无效</b>	
	16#0001	-	<ul style="list-style-type: none"> <li>将无错误组态下载到控制器； 使用指令“MC_Power”再次启用轴</li> <li>在线更正不正确的值，并视需要重新启动命令</li> </ul>
<b>16#8701</b>		<b>命令表中“位置/行进路径”(Position/travel path) 的值无效</b>	
	16#0002	值的数字格式无效	<ul style="list-style-type: none"> <li>将无错误组态下载到控制器； 使用指令“MC_Power”再次启用轴</li> <li>在线更正不正确的值，并视需要重新启动命令</li> </ul>
	16#0005	该值超出取值范围（大于 $1e^{12}$ ）	
	16#0006	该值超出取值范围（小于 $-1e^{12}$ ）	
<b>16#8702</b>		<b>命令表中“速度”(Velocity) 的值无效</b>	
	16#0002	值的数字格式无效	<ul style="list-style-type: none"> <li>将无错误组态下载到控制器； 使用指令“MC_Power”再次启用轴</li> <li>在线更正不正确的值，并视需要重新启动命令</li> </ul>
	16#0008	值大于组态的最大速度	
	16#0009	值小于组态的启动/停止速度	
<b>16#8703</b>		<b>命令表中“持续时间”(Duration) 的值无效</b>	
	16#0002	值的数字格式无效	<ul style="list-style-type: none"> <li>将无错误组态下载到控制器； 使用指令“MC_Power”再次启用轴</li> <li>在线更正不正确的值，并视需要重新启动命令</li> </ul>
	16#0021	值大于 64800 s	
	16#0022	值小于 0.001 s	
<b>16#8704</b>		<b>命令表中“下一步”(Next step) 的值无效</b>	
	16#0011	所选值无效	<ul style="list-style-type: none"> <li>将无错误组态下载到控制器； 使用指令“MC_Power”再次启用轴</li> <li>在线更正不正确的值，并视需要重新启动命令</li> </ul>
	16#0023	该命令不允许命令转换	

内部错误

ErrorID	ErrorInfo	说明	解决方法
16#8FFF		内部错误	
	16#F0**	-	CPU 的断电和上电 如果这样做无效，则联系客户支持。 准备好以下信息： <ul style="list-style-type: none"> <li>• ErrorID</li> <li>• ErrorInfo</li> <li>• 诊断缓冲区条目</li> </ul>

参见

ErrorID 和 ErrorInfo 列表（自工艺对象 V4 起）(页 205)

将多个轴与同一个 PTO 一起使用 (页 190)

**ErrorID 和 ErrorInfo 列表（工艺对象 V1）**

下表列出了运动控制指令中指示的所有 ErrorID 和 ErrorInfo。除错误原因外，还列出了清除错误的解决方法：

**伴随轴停止的运行错误**

ErrorID	ErrorInfo	说明	解决方法
<b>16#8000</b>		<b>驱动器错误，丢失“驱动器就绪”信号</b>	
	16#0001	-	使用指令“MC_Reset”确认错误； 提供驱动器信号；根据需要重新启动命令
<b>16#8001</b>		<b>已触发下限软限位开关</b>	
	16#000E	已经以当前组态的减速度到达下限软限位开关的位置	使用“MC_Reset”指令确认错误； 使用运动命令使轴沿正向移动， 超出软限位开关的范围
	16#000F	已经以急停减速度到达下限软限位开关的位置	
	16#0010	已经以急停减速度超出下限软限位开关的位置	
<b>16#8002</b>		<b>已触发上限软限位开关</b>	
	16#000E	已经以当前组态的减速度到达上限软限位开关的位置	使用“MC_Reset”指令确认错误； 使用运动命令使轴沿负向移动， 超出软限位开关的范围
	16#000F	已经以急停减速度到达上限软限位开关的位置	
	16#0010	已经以急停减速度超出上限软限位开关的位置	
<b>16#8003</b>		<b>已到达下限硬限位开关</b>	
	16#000E	已到达下限硬限位开关。轴以急停减速度停止。 (主动回原点期间，未找到参考点开关)	使用指令“MC_Reset”确认已启用轴的 错误；使用运动命令使轴沿正向移动， 超出硬限位开关的范围。

ErrorID	ErrorInfo	说明	解决方法
<b>16#8004</b>		<b>已到达上限硬限位开关</b>	
	16#000E	<p>已到达上限硬限位开关。轴以急停减速度停止。</p> <p>（主动回原点期间，未找到参考点开关）</p>	<p>使用指令“MC_Reset”确认已启用轴的错误；使用运动命令使轴沿负向移动，超出硬限位开关的范围。</p>
<b>16#8005</b>		<b>PTO/HSC 已被另一个轴占用</b>	
	16#0001	-	<p><b>轴未正确组态：</b></p> <p>更正 PTO (Pulse Train Output)/ HSC (High Speed Counter) 的组态并将其下载到控制器</p> <hr/> <p><b>多个轴将通过一个 PTO 运行：</b></p> <p>另一个轴正在使用 PTO/HSC。如果要控制当前轴，则必须使用“MC_Power”(Enable = FALSE) 禁用另一个轴。 （另请参见将多个轴与同一个 PTO 一起使用 (页 190)）</p>

## 不伴随轴停止的运行错误

ErrorID	ErrorInfo	说明	解决方法
<b>16#8200</b>		轴未启用	
	16#0001	-	启用轴；重新启动命令
<b>16#8201</b>		轴已由另一个“MC_Power”实例启用	
	16#0001	-	仅使用一个“MC_Power”指令启用轴
<b>16#8202</b>		超出并行激活的运动控制命令的最大数量（所有运动控制工艺对象最多可执行 200 个命令）	
	16#0001	-	减少并行激活命令数；重新启动命令 如果运动控制指令的参数“Busy”= TRUE，则说明命令已激活。
<b>16#8203</b>		轴当前在“手动控制”(Manual control) 模式（轴控制面板）下运行	
	16#0001	-	退出“手动控制”；重新启动命令
<b>16#8204</b>		轴未回原点	
	16#0001	-	使用指令“MC_Home”使轴回原点； 重新启动命令
<b>16#8205</b>		轴当前由用户程序控制（该错误仅显示在轴控制面板中）	
	16#0001	-	使用指令“MC_Power”禁用轴或者在轴控制面板中再次选择“手动控制”(Manual control)
<b>16#8206</b>		工艺对象“轴”尚未启用	
	16#0001	-	使用指令“MC_Power”Enable = TRUE 启用轴或者在轴控制面板中启用轴。
<b>16#8207</b>		命令被拒绝	
	16#0016	正在执行主动回原点；无法启动另一种回原点方法。	等到主动回原点完成或通过运动命令（例如，“MC_Halt”）中止主动回原点。然后可以启动其它回原点类型。

## 块参数错误

ErrorID	ErrorInfo	说明	解决方法
<b>16#8400</b>		<b>运动控制指令的“Position”参数值无效</b>	
	16#0002	值的数字格式无效	更正“位置”值；重新启动命令
	16#0005	该值超出取值范围（大于 $1e^{12}$ ）	
	16#0006	该值超出取值范围（小于 $-1e^{12}$ ）	
<b>16#8401</b>		<b>运动控制指令的“Distance”参数值无效</b>	
	16#0002	值的数字格式无效	更正“Distance”值；重新启动命令
	16#0005	该值超出取值范围（大于 $1e^{12}$ ）	
	16#0006	该值超出取值范围（小于 $-1e^{12}$ ）	
<b>16#8402</b>		<b>运动控制指令的“Velocity”参数值无效</b>	
	16#0002	值的数字格式无效	更正“Velocity”值；重新启动命令
	16#0008	速度大于最大速度	
	16#0009	速度小于启动/停止速度	
<b>16#8403</b>		<b>运动控制指令的“Direction”参数值无效</b>	
	16#0011	所选值无效	更正所选值；重新启动命令
<b>16#8404</b>		<b>运动控制指令的“Mode”参数值无效</b>	
	16#0011	所选值无效	更正所选值；重新启动命令
	16#0015	未组态主动/被动回原点	更正组态并将其下载到控制器； 启用轴并重新启动命令
	16#0017	虽然已禁用硬限位开关，但还是激活了在该硬限位开关处轴反向	<ul style="list-style-type: none"> <li>使用变量 &lt;轴&gt;.Config.PositionLimits_HW.Active= TRUE 激活硬限位开关，并重新启动该命令</li> <li>更正组态并将其下载到控制器； 启用轴并重新启动命令</li> </ul>
<b>16#8405</b>		<b>运动控制指令的“StopMode”参数值无效</b>	
	16#0011	所选值无效	更正所选值；再次启用轴



ErrorID	ErrorInfo	说明	解决方法
<b>16#8406</b>		不允许同时正向和反向点动	
	16#0001	-	采取措施确保参数“JogForward”和“JogBackward”的信号状态不会同时为TRUE；重新启动命令。
<b>16#8407</b>		只有禁用轴后，才允许使用运动控制指令“MC_Power”开关轴。	
	16#0001	-	禁用激活的轴；然后可以切换到其它轴并启用该轴。

## 组态错误

ErrorID	ErrorInfo	说明	解决方法
<b>16#8600</b>		<b>脉冲发生器 (PTO) 的参数分配无效</b>	
	16#000B	地址无效	更正 PTO (Pulse Train Output) 的组态并将其下载到控制器
<b>16#8601</b>		<b>高速计数器 (HSC) 的参数分配无效</b>	
	16#000B	地址无效	更正 HSC (High Speed Counter) 的组态并将其下载到控制器
<b>16#8602</b>		<b>“使能输出”(Enable output) 的参数分配无效</b>	
	16#000D	地址无效	更正组态并将其下载到控制器
<b>16#8603</b>		<b>“输入就绪”(Ready input) 的参数分配无效</b>	
	16#000D	地址无效	更正组态并将其下载到控制器
<b>16#8604</b>		<b>“电机每转的脉冲数”(Pulses per motor revolution) 值无效</b>	
	16#000A	值小于或等于零	更正组态并将其下载到控制器
<b>16#8605</b>		<b>“每转的距离”(Distance per revolution) 值无效</b>	
	16#0002	值的数字格式无效	更正组态并将其下载到控制器
	16#000A	值小于或等于零	
<b>16#8606</b>		<b>“启动/停止速度”(Start / stop velocity) 值无效</b>	
	16#0002	值的数字格式无效	更正组态并将其下载到控制器
	16#0003	值超出硬件限值	
	16#0004	值小于硬件限值	
	16#0007	启动/停止速度大于最大速度	
<b>16#8607</b>		<b>“最大速度”(maximum velocity) 值无效</b>	
	16#0002	值的数字格式无效	更正组态并将其下载到控制器
	16#0003	值超出硬件限值	
	16#0004	值小于硬件限值	

ErrorID	ErrorInfo	说明	解决方法
<b>16#8608</b>		<b>“加速度”(Acceleration) 值无效</b>	
	16#0002	值的数字格式无效	<ul style="list-style-type: none"> <li>将无错误组态下载到控制器；使用指令“MC_Power”再次启用轴</li> <li>在线更正不正确的值；使用指令“MC_Reset”确认错误并视需要重新启动命令</li> </ul>
	16#0003	值超出硬件限值	
	16#0004	值小于硬件限值	
<b>16#8609</b>		<b>“减速度”(Deceleration) 值无效</b>	
	16#0002	值的数字格式无效	<ul style="list-style-type: none"> <li>将无错误组态下载到控制器；使用指令“MC_Power”再次启用轴</li> <li>在线更正不正确的值；使用指令“MC_Reset”确认错误并视需要重新启动命令</li> </ul>
	16#0003	值超出硬件限值	
	16#0004	值小于硬件限值	
<b>16#860A</b>		<b>“急停减速度”(Emergency stop deceleration) 值无效</b>	
	16#0002	值的数字格式无效	<ul style="list-style-type: none"> <li>将无错误组态下载到控制器；使用指令“MC_Power”再次启用轴</li> <li>在线更正不正确的值；使用指令“MC_Reset”确认错误并视需要重新启动命令</li> </ul>
	16#0003	值超出硬件限值	
	16#0004	值小于硬件限值	
<b>16#860B</b>		<b>下限软限位开关的位置值无效</b>	
	16#0002	值的数字格式无效	<ul style="list-style-type: none"> <li>将无错误组态下载到控制器；使用指令“MC_Power”再次启用轴</li> <li>在线更正不正确的值；使用指令“MC_Reset”确认错误并视需要重新启动命令</li> </ul>
	16#0005	该值超出取值范围（大于 $1e^{12}$ ）	
	16#0006	该值超出取值范围（小于 $-1e^{12}$ ）	
	16#0007	下限软限位开关的位置值大于上限软限位开关的位置值	
<b>16#860C</b>		<b>上限软限位开关的位置值无效</b>	
	16#0002	值的数字格式无效	<ul style="list-style-type: none"> <li>将无错误组态下载到控制器；使用指令“MC_Power”再次启用轴</li> <li>在线更正不正确的值；使用指令“MC_Reset”确认错误并视需要重新启动命令</li> </ul>
	16#0005	该值超出取值范围（大于 $1e^{12}$ ）	
	16#0006	该值超出取值范围（小于 $-1e^{12}$ ）	

ErrorID	ErrorInfo	说明	解决方法
<b>16#860D</b>		<b>下限硬限位开关的地址无效</b>	
	16#000C	下降沿的地址无效	更正组态并将其下载到控制器
	16#000D	上升沿的地址无效	
<b>16#860E</b>		<b>上限硬限位开关的地址无效</b>	
	16#000C	下降沿的地址无效	更正组态并将其下载到控制器
	16#000D	上升沿的地址无效	
<b>16#860F</b>		<b>“起始位置偏移值”(home position offset) 无效</b>	
	16#0002	值的数字格式无效	更正组态并将其下载到控制器
	16#0005	该值超出取值范围 (大于 $1e^{12}$ )	
	16#0006	该值超出取值范围 (小于 $-1e^{12}$ )	
<b>16#8610</b>		<b>“逼近速度”(approach velocity) 值无效</b>	
	16#0002	值的数字格式无效	更正组态并将其下载到控制器
	16#0008	速度大于最大速度	
	16#0009	速度小于启动/停止速度	
<b>16#8611</b>		<b>“回原点速度”(Homing velocity) 值无效</b>	
	16#0002	值的数字格式无效	更正组态并将其下载到控制器
	16#0008	速度大于最大速度	
	16#0009	速度小于启动/停止速度	
<b>16#8612</b>		<b>参考点开关的地址无效</b>	
	16#000C	下降沿的地址无效	更正组态并将其下载到控制器
	16#000D	上升沿的地址无效	
<b>16#8613</b>		<b>虽然未组态硬限位开关，但还是在主动回原点过程中，激活了在该硬限位开关处反向</b>	
	16#0001	-	更正组态并将其下载到控制器

## 内部错误

ErrorID	ErrorInfo	说明	解决方法
16#8FFF		内部错误	
	16#F0**	-	CPU 的断电和上电 如果这样做无效，则联系客户支持。 准备好以下信息： <ul style="list-style-type: none"> <li>• ErrorID</li> <li>• ErrorInfo</li> <li>• 诊断缓冲区条目</li> </ul>

## 参见

ErrorID 和 ErrorInfo 列表（自工艺对象 V4 起）(页 205)

将多个轴与同一个 PTO 一起使用 (页 190)

1.11.10.5 轴工艺对象 V1...3 的变量

Config 变量 V1...3

Config.General 变量 V1...3

图例

数据类型	变量的数据类型		
起始值	变量的起始值 轴组态可以改写初始值。		
访问	在用户程序中对变量的访问：		
	RW	可以在用户程序中读写该变量。	
	R	可以在用户程序中读取该变量。	
	-	不能在用户程序中使用该变量。	
有效	指定变量的更改何时生效。		
HMI	可以在 HMI 系统中使用该变量。		

<轴名称>.Config.General.PTO				
不能在用户程序中评估该变量。				
数据类型	起始值	访问	有效	HMI
DWORD	DW#16#00000000	-	-	-

<轴名称>.Config.General.HSC				
不能在用户程序中评估该变量。				
数据类型	起始值	访问	有效	HMI
DWORD	DW#16#00000000	-	-	-

<b>&lt;轴名称&gt;.Config.General.LengthUnit (从 V2.0“轴”工艺对象起)</b>				
在组态中选择的参数的测量单位:				
<ul style="list-style-type: none"> <li>• 1013 =“毫米”</li> <li>• 1010 =: “米”</li> <li>• 1019 =“英寸”</li> <li>• 1018 =“英尺”</li> <li>• 1005 =“°” (度)</li> <li>• -1 =“脉冲”</li> </ul>				
数据类型	起始值	访问	有效	HMI
Int	1013	R	-	√

### 参见

自 V4 起的定位轴工艺对象的变量 (页 235)

### Config.DriveInterface 变量 V1...3

#### 图例

<b>数据类型</b>	变量的数据类型		
<b>起始值</b>	变量的起始值 轴组态可以改写初始值。		
<b>访问</b>	在用户程序中对变量的访问：		
	RW	可以在用户程序中读写该变量。	
	R	可以在用户程序中读取该变量。	
	-	不能在用户程序中使用该变量。	
<b>有效</b>	指定变量的更改何时生效。		
<b>HMI</b>	可以在 HMI 系统中使用该变量。		

<b>&lt;轴名称&gt;.Config.DriveInterface.EnableOutput</b>				
不能在用户程序中评估该变量。				
数据类型	起始值	访问	有效	HMI
-	-	-	-	-

<b>&lt;轴名称&gt;.Config.DriveInterface.ReadyInput</b>				
不能在用户程序中评估该变量。				
数据类型	起始值	访问	有效	HMI
-	-	-	-	-

#### 参见

自 V4 起的定位轴工艺对象的变量 (页 235)



## Config.Mechanics 变量 V1...3

## 图例

<b>数据类型</b>	变量的数据类型
<b>起始值</b>	变量的起始值 轴组态可以改写初始值。
<b>访问</b>	在用户程序中对变量的访问：
	RW 可以在用户程序中读写该变量。
	R 可以在用户程序中读取该变量。
-	不能在用户程序中使用该变量。
<b>有效</b>	指定变量的更改何时生效。
<b>HMI</b>	可以在 HMI 系统中使用该变量。

<b>&lt;轴名称&gt;.Config.Mechanics.PulsesPerDriveRevolution</b>				
电机每转的增量				
<b>数据类型</b>	<b>起始值</b>	<b>访问</b>	<b>有效</b>	<b>HMI</b>
DInt	L#1000	R	-	√

<b>&lt;轴名称&gt;.Config.Mechanics.LeadScrew</b>				
每转的距离（单位为组态的测量单位）				
<b>数据类型</b>	<b>起始值</b>	<b>访问</b>	<b>有效</b>	<b>HMI</b>
Real	1.0E+001	R	-	√

<b>&lt;轴名称&gt;.Config.Mechanics.InverseDirection</b>				
反转方向				
<b>数据类型</b>	<b>起始值</b>	<b>访问</b>	<b>有效</b>	<b>HMI</b>
Bool	FALSE	R	-	√

## 参见

自 V4 起的定位轴工艺对象的变量 (页 235)

### Config.DynamicLimits 变量 V1...3

#### 图例

<b>数据类型</b>	变量的数据类型		
<b>起始值</b>	变量的起始值 轴组态可以改写初始值。		
<b>访问</b>	在用户程序中对变量的访问：		
	RW	可以在用户程序中读写该变量。	
	R	可以在用户程序中读取该变量。	
	-	不能在用户程序中使用该变量。	
<b>有效</b>	指定变量的更改何时生效。		
<b>HMI</b>	可以在 HMI 系统中使用该变量。		

<b>&lt;轴名称&gt;.Config.DynamicLimits.MinVelocity</b>				
轴的启动/停止速度（单位为组态的测量单位）				
数据类型	起始值	访问	有效	HMI
Real	1.0E+001	R	-	√

<b>&lt;轴名称&gt;.Config.DynamicLimits.MaxVelocity</b>				
轴的最大速度（单位为组态的测量单位）				
数据类型	起始值	访问	有效	HMI
Real	2.5E+002	R	-	√

#### 参见

自 V4 起的定位轴工艺对象的变量 (页 235)

## Config.DynamicDefaults 变量 V1...3

## 图例

<b>数据类型</b>	变量的数据类型	
<b>起始值</b>	变量的起始值 轴组态可以改写初始值。	
<b>访问</b>	在用户程序中对变量的访问：	
	RW	可以在用户程序中读写该变量。
	R	可以在用户程序中读取该变量。
	-	不能在用户程序中使用该变量。
<b>有效</b>	指定变量的更改何时生效。	
	1	在轴激活 (<轴名称>.StatusBits.Activated 变量从 FALSE 更改为 TRUE)、禁用或启用时
	2	轴已启用时
	5	下一次启动 MC_MoveAbsolute、MC_MoveRelative、MC_MoveVelocity、MC_MoveJog、MC_Halt、MC_CommandTable 或激活的 MC_Home 命令时 (Mode = 3)。
	6	MC_MoveJog 命令停止时
<b>HMI</b>	可以在 HMI 系统中使用该变量。	

<轴名称>.Config.DynamicDefaults.Acceleration					
轴的加速度（单位为组态的量纲单位）					
数据类型	起始值	访问	有效		HMI
Real	4.8E+001	RW	5	CPU 固件 V1.0	√
			1, 5, 6	自 V2.0 版起的 CPU 固件	

<轴名称>.Config.DynamicDefaults.Deceleration					
轴的减速度（单位为组态的量纲单位）					
数据类型	起始值	访问	有效		HMI
Real	4.8E+001	RW	5, 6	CPU 固件 V1.0	√
			1, 5, 6	自 V2.0 版起的 CPU 固件	

<b>&lt;轴名称&gt;.Config.DynamicDefaults.EmergencyDeceleration</b>					
轴的急停减速度（单位为组态的量纲单位）					
数据类型	起始值	访问	有效		HMI
Real	1.2E+002	RW	2, 5, 6	CPU 固件 V1.0	√
			1, 5, 6	自 V2.0 版起的 CPU 固件	

<b>&lt;轴名称&gt;.Config.DynamicDefaults.JerkActive</b> （从 V2.0“轴”工艺对象起）				
TRUE = 冲击限制已激活				
数据类型	起始值	访问	有效	HMI
Bool	FALSE	RW	1, 5	√

<b>&lt;轴名称&gt;.Config.DynamicDefaults.Jerk</b> （从 V2.0“轴”工艺对象起）				
轴加速和轴减速斜坡期间的冲击（使用组态的测量单位指定）				
数据类型	起始值	访问	有效	HMI
Real	1.92E+002	RW	1, 5	√

参见

自 V4 起的定位轴工艺对象的变量 (页 235)

## Config.PositionLimits\_SW 变量 V1...3

## 图例

<b>数据类型</b>	变量的数据类型	
<b>起始值</b>	变量的起始值 轴组态可以改写初始值。	
<b>访问</b>	在用户程序中对变量的访问：	
	RW	可以在用户程序中读写该变量。
	R	可以在用户程序中读取该变量。
	-	不能在用户程序中使用该变量。
<b>有效</b>	指定变量的更改何时生效。	
	1	在轴激活（<轴名称>.StatusBits.Activated 变量从 FALSE 更改为 TRUE）、禁用或启用时
	4	在轴停止后的下一次运动控制命令启动时。可以使用变量 <轴名称>.StatusBits.Standstill 检查轴停止状态。 StatusBits.Standstill.
	5	下一次启动 MC_MoveAbsolute、MC_MoveRelative、MC_MoveVelocity、MC_MoveJog、MC_Halt、MC_CommandTable 或激活的 MC_Home 命令时 (Mode = 3)。
<b>HMI</b>	可以在 HMI 系统中使用该变量。	

<b>&lt;轴名称&gt;.Config.PositionLimits_SW.Active</b>					
TRUE = 软限位开关激活					
<b>数据类型</b>	<b>起始值</b>	<b>访问</b>	<b>有效</b>		<b>HMI</b>
Bool	FALSE	RW	4	CPU 固件 V1.0	√
			1, 5, 6	自 V2.0 版起的 CPU 固件	

<b>&lt;轴名称&gt;.Config.PositionLimits_SW.MinPosition</b>					
下限软限位开关的位置（单位为组态的测量单位）					
<b>数据类型</b>	<b>起始值</b>	<b>访问</b>	<b>有效</b>		<b>HMI</b>
Real	-1.0E+004	RW	4	CPU 固件 V1.0	√
			1, 5, 6	自 V2.0 版起的 CPU 固件	

<b>&lt;轴名称&gt;.Config.PositionLimits_SW.MaxPosition</b>					
上限软限位开关的位置（单位为组态的测量单位）					
数据类型	起始值	访问	有效		HMI
Real	1.0E+004	RW	4	CPU 固件 V1.0	√
			1, 5, 6	自 V2.0 版起的 CPU 固件	

参见

自 V4 起的定位轴工艺对象的变量 (页 235)

## Config.PositionLimits\_HW 变量 V1...3

## 图例

<b>数据类型</b>	变量的数据类型	
<b>起始值</b>	变量的起始值 轴组态可以改写初始值。	
<b>访问</b>	在用户程序中对变量的访问：	
	RW	可以在用户程序中读写该变量。
	R	可以在用户程序中读取该变量。
	-	不能在用户程序中使用该变量。
<b>有效</b>	指定变量的更改何时生效。	
	1	在轴激活（<轴名称>.StatusBits.Activated 变量从 FALSE 更改为 TRUE）、禁用或启用时
	3	启用轴后（轴之前必须处于停止状态）。可以使用变量 <轴名称>.StatusBits.Standstill 检查轴停止状态。 StatusBits.Standstill.
	4	在轴停止后的下一次运动控制命令启动时。可以使用变量 <轴名称>.StatusBits.Standstill 检查轴停止状态。 StatusBits.Standstill.
	5	下一次启动 MC_MoveAbsolute、MC_MoveRelative、MC_MoveVelocity、MC_MoveJog、MC_Halt、MC_CommandTable 或激活的 MC_Home 命令时 (Mode = 3)。
<b>HMI</b>	可以在 HMI 系统中使用该变量。	

<b>&lt;轴名称&gt;.Config.PositionLimits_HW.Active</b>					
TRUE = 硬限位开关激活。					
<b>数据类型</b>	<b>起始值</b>	<b>访问</b>	<b>有效</b>		<b>HMI</b>
Bool	FALSE	RW	3, 4	CPU 固件 V1.0	√
			1, 5, 6	自 V2.0 版起的 CPU 固件	

<b>&lt;轴名称&gt;.Config.PositionLimits_HW.MinSwitchedLevel</b>				
TRUE = CPU 输入端电平为 24 V，表示已逼近下限硬限位开关 FALSE = CPU 输入端电平为 0 V，表示已逼近下限硬限位开关				
数据类型	起始值	访问	有效	HMI
Bool	FALSE	R	-	√

<b>&lt;轴名称&gt;.Config.PositionLimits_HW.MinFallingEvent</b>				
不能在用户程序中评估该变量。				
数据类型	起始值	访问	有效	HMI
DWord	DW#16#00000000	-	-	-

<b>&lt;轴名称&gt;.Config.PositionLimits_HW.MinRisingEvent</b>				
不能在用户程序中评估该变量。				
数据类型	起始值	访问	有效	HMI
DWord	DW#16#00000000	-	-	-

<b>&lt;轴名称&gt;.Config.PositionLimits_HW.MaxSwitchedLevel</b>				
TRUE = CPU 输入端电平为 24 V，表示已逼近上限硬限位开关 FALSE = CPU 输入端电平为 0 V，表示已逼近上限硬限位开关				
数据类型	起始值	访问	有效	HMI
Bool	FALSE	R	-	√

<b>&lt;轴名称&gt;.Config.PositionLimits_HW.MaxFallingEvent</b>				
不能在用户程序中评估该变量。				
数据类型	起始值	访问	有效	HMI
DWord	DW#16#00000000	-	-	-



<b>&lt;轴名称&gt;.Config.PositionLimits_HW.MaxRisingEvent</b>				
不能在用户程序中评估该变量。				
数据类型	起始值	访问	有效	HMI
DWord	DW#16#00000000	-	-	-

### 参见

自 V4 起的定位轴工艺对象的变量 (页 235)

### Config.Homing 变量 V1...3

#### 图例

<b>数据类型</b>	变量的数据类型	
<b>起始值</b>	变量的起始值 轴组态可以改写初始值。	
<b>访问</b>	在用户程序中对变量的访问：	
	RW	可以在用户程序中读写该变量。
	R	可以在用户程序中读取该变量。
	-	不能在用户程序中使用该变量。
<b>有效</b>	指定变量的更改何时生效。	
	1	在轴激活 (<轴名称>.StatusBits.Activated 变量从 FALSE 更改为 TRUE)、禁用或启用时
	7	启动被动回原点命令时
	8	启动主动回原点命令时
<b>HMI</b>	可以在 HMI 系统中使用该变量。	

<b>&lt;轴名称&gt;.Config.Homing.AutoReversal</b>					
TRUE = 在启用硬限位开关时反向（主动回原点）					
FALSE = 在禁用硬限位开关时反向（主动回原点）					
数据类型	起始值	访问	有效		HMI
Bool	TRUE	R	-	V1.0“轴”工艺对象	√
		RW	1, 8	V2.0“轴”工艺对象	

<b>&lt;轴名称&gt;.Config.Homing.Direction</b>					
TRUE = 正逼近方向（用于搜索参考点开关）和正回原点方向（主动回原点）					
FALSE = 负逼近方向（用于搜索参考点开关）和正回原点方向（主动回原点）					
数据类型	起始值	访问	有效		HMI
Bool	TRUE	R	-	V1.0“轴”工艺对象	√
		RW	1, 8	V2.0“轴”工艺对象	

<b>&lt;轴名称&gt;.Config.Homing.SideActiveHoming</b> (从 V2.0“轴”工艺对象起)				
TRUE = 在参考点开关的上端回原点 (主动回原点)				
TRUE = 在参考点开关的下端回原点 (主动回原点)				
数据类型	起始值	访问	有效	HMI
Bool	TRUE	RW	1, 8	√

<b>&lt;轴名称&gt;.Config.Homing.SidePassiveHoming</b> (从 V2.0“轴”工艺对象起)				
TRUE = 在参考点开关的上端回原点 (被动回原点)				
TRUE = 在参考点开关的下端回原点 (被动回原点)				
数据类型	起始值	访问	有效	HMI
Bool	TRUE	RW	1, 7	√

<b>&lt;轴名称&gt;.Config.Homing.RisingEdge</b> (从 V1.0“轴”工艺对象起)				
TRUE = 利用参考点开关的信号下降沿回原点 (主动回原点)				
FALSE = 利用参考点开关的信号上升沿回原点 (主动回原点)				
有关该变量对被动回原点的影响的信息, 请参见“组态 - 回原点”中的说明。				
数据类型	起始值	访问	有效	HMI
Bool	FALSE	R	-	√

<b>&lt;轴名称&gt;.Config.Homing.Offset</b>				
回原点位置偏移/单位为组态的测量单位 (主动回原点)				
数据类型	起始值	访问	有效	HMI
Real	0.0	R	-	V1.0“轴”工艺对象
		RW	1, 8	V2.0“轴”工艺对象

<b>&lt;轴名称&gt;.Config.Homing.FastVelocity</b>				
逼近速度/单位为组态的测量单位 (主动回原点)				
数据类型	起始值	访问	有效	HMI
Real	2.0E+002	R	-	V1.0“轴”工艺对象
		RW	1, 8	V2.0“轴”工艺对象

<b>&lt;轴名称&gt;.Config.Homing.SlowVelocity</b>					
回原点速度/单位为组态的测量单位（主动回原点）					
数据类型	起始值	访问	有效		HMI
Real	4.0E+001	R	-	V1.0“轴”工艺对象	√
		RW	1, 8	V2.0“轴”工艺对象	

<b>&lt;轴名称&gt;.Config.Homing.FallingEvent</b>				
不能在用户程序中评估该变量。				
数据类型	起始值	访问	有效	HMI
DWord	DW#16#00000000	-	-	-

<b>&lt;轴名称&gt;.Config.Homing.RisingEvent</b>				
不能在用户程序中评估该变量。				
数据类型	起始值	访问	有效	HMI
DWord	DW#16#00000000	-	-	

参见

自 V4 起的定位轴工艺对象的变量 (页 235)

**MotionStatus 变量 V1...3**

## 图例

<b>数据类型</b>	变量的数据类型	
<b>起始值</b>	变量的起始值	
<b>访问</b>	在用户程序中对变量的访问：	
	<b>RW</b>	可以在用户程序中读写该变量。
	<b>R</b>	可以在用户程序中读取该变量。
	<b>-</b>	不能在用户程序中使用该变量。
<b>有效</b>	指定变量的更改何时生效。	
<b>HMI</b>	可以在 HMI 系统中使用该变量。	

<b>&lt;轴名称&gt;.MotionStatus.Position</b>				
轴的当前位置（单位为组态的测量单位）。如果轴未回原点，则该变量指示相对于轴启用位置的位置值。				
<b>数据类型</b>	<b>起始值</b>	<b>访问</b>	<b>有效</b>	<b>HMI</b>
Real	0.0	R	-	√

<b>&lt;轴名称&gt;.MotionStatus.Velocity</b>				
轴的当前速度（单位为组态的测量单位）				
<b>数据类型</b>	<b>起始值</b>	<b>访问</b>	<b>有效</b>	<b>HMI</b>
Real	0.0	R	-	√

<b>&lt;轴名称&gt;.MotionStatus.Distance</b>				
当前与轴的目标位置的距离（单位为组态的测量单位）。 变量值仅在利用“MC_MoveAbsolute”或“MC_MoveRelative”执行定位命令时或执行轴命令表时有效。				
<b>数据类型</b>	<b>起始值</b>	<b>访问</b>	<b>有效</b>	<b>HMI</b>
Real	0.0	R	-	√

<b>&lt;轴名称&gt;.MotionStatus.TargetPosition</b>				
轴的目标位置（单位为组态的测量单位）。变量值仅在使用“MC_MoveAbsolute”或“MC_MoveRelative”执行定位命令时或执行轴命令表时有效。				
<b>数据类型</b>	<b>起始值</b>	<b>访问</b>	<b>有效</b>	<b>HMI</b>
Real	0.0	R	-	√

参见

运动状态 (页 187)

自 V4 起的定位轴工艺对象的变量 (页 235)

**StatusBits 变量 V1...3**

## 图例

<b>数据类型</b>	变量的数据类型	
<b>起始值</b>	变量的起始值	
<b>访问</b>	在用户程序中对变量的访问:	
	<b>RW</b>	可以在用户程序中读写该变量。
	<b>R</b>	可以在用户程序中读取该变量。
	<b>-</b>	不能在用户程序中使用该变量。
<b>有效</b>	指定变量的更改何时生效。	
<b>HMI</b>	可以在 HMI 系统中使用该变量。	

<b>&lt;轴名称&gt;.StatusBits.Activated</b>				
TRUE = 轴激活。轴连接到分配的 PTO (Pulse Train Output)。将循环更新工艺数据块中的数据。				
<b>数据类型</b>	<b>起始值</b>	<b>访问</b>	<b>有效</b>	<b>HMI</b>
Bool	FALSE	R	-	√

<b>&lt;轴名称&gt;.StatusBits.Enable</b>				
TRUE = 轴已启用且准备好执行运动控制命令。				
<b>数据类型</b>	<b>起始值</b>	<b>访问</b>	<b>有效</b>	<b>HMI</b>
Bool	FALSE	R	-	√

<b>&lt;轴名称&gt;.StatusBits.HomingDone</b>				
TRUE = 轴已回原点且能够执行绝对定位命令。对于相对定位而言，轴不必回原点。 在主动回原点过程中，该状态为 FALSE。如果轴已经回原点， 则在被动回原点过程中该状态将保持为 TRUE。				
<b>数据类型</b>	<b>起始值</b>	<b>访问</b>	<b>有效</b>	<b>HMI</b>
Bool	FALSE	R	-	√

<b>&lt;轴名称&gt;.StatusBits.Done</b>				
TRUE = 该轴没有已激活的运动控制命令。				
数据类型	起始值	访问	有效	HMI
Bool	FALSE	R	-	√

<b>&lt;轴名称&gt;.StatusBits.Error</b>				
TRUE = 轴工艺对象出错。 有关错误的详细信息，请参见自动模式下的运动控制指令的“ErrorID”和“ErrorInfo”参数。 在手动模式下，轴控制面板的“错误消息”(Error message) 框可显示有关错误原因的详细信息。				
数据类型	起始值	访问	有效	HMI
Bool	FALSE	R	-	√

<b>&lt;轴名称&gt;.StatusBits.StandStill</b>				
TRUE = 轴处于停止状态。				
数据类型	起始值	访问	有效	HMI
Bool	FALSE	R	-	√

<b>&lt;轴名称&gt;.StatusBits.PositioningCommand</b>				
TRUE = 该轴正在执行定位命令。				
数据类型	起始值	访问	有效	HMI
Bool	FALSE	R	-	√

<b>&lt;轴名称&gt;.StatusBits.SpeedCommand</b>				
TRUE = 该轴正在以预定义速度执行行进命令。				
数据类型	起始值	访问	有效	HMI
Bool	FALSE	R	-	√

<b>&lt;轴名称&gt;.StatusBits.Homing</b>				
TRUE = 轴正在执行“MC_Home”运动控制指令或轴命令表的回原点命令。				
数据类型	起始值	访问	有效	HMI
Bool	FALSE	R	-	√



<b>&lt;轴名称&gt;.StatusBits.CommandTableActive</b>				
TRUE = 该轴由运动控制指令“MC_CommandTable”控制。				
数据类型	起始值	访问	有效	HMI
Bool	FALSE	R	-	√

<b>&lt;轴名称&gt;.StatusBits.ConstantVelocity</b>				
TRUE = 轴在恒速运转。				
数据类型	起始值	访问	有效	HMI
Bool	FALSE	R	-	√

<b>&lt;轴名称&gt;.StatusBits.Acceleration</b>				
TRUE = 轴在加速。				
数据类型	起始值	访问	有效	HMI
Bool	FALSE	R	-	√

<b>&lt;轴名称&gt;.StatusBits.Deceleration</b>				
TRUE = 轴在减速（速度下降）。				
数据类型	起始值	访问	有效	HMI
Bool	FALSE	R	-	√

<b>&lt;轴名称&gt;.StatusBits.ControlPanelActive</b>				
TRUE = 在轴命令表中启用了“手动控制”模式。轴命令表对“轴”工艺对象具有优先控制权。 不能通过用户程序来控制轴。				
数据类型	起始值	访问	有效	HMI
Bool	FALSE	R	-	√

<b>&lt;轴名称&gt;.StatusBits.DriveReady</b>				
TRUE = 驱动器准备就绪。				
数据类型	起始值	访问	有效	HMI
Bool	FALSE	R	-	√

<b>&lt;轴名称&gt;.StatusBits.RestartRequired</b>				
TRUE = 值已在装载存储器中修改。				
要在 CPU RUN 操作模式下将值下载到工作存储器，需要重新启动轴。使用运动控制指令 MC_Reset 执行此操作。				
数据类型	起始值	访问	有效	HMI
Bool	FALSE	R	-	√

**参见**

状态和错误位（自工艺对象 V4 起） (页 183)

自 V4 起的定位轴工艺对象的变量 (页 235)

**ErrorBits 变量 V1...3**

## 图例

<b>数据类型</b>	变量的数据类型	
<b>起始值</b>	变量的起始值	
<b>访问</b>	在用户程序中对变量的访问:	
	<b>RW</b>	可以在用户程序中读写该变量。
	<b>R</b>	可以在用户程序中读取该变量。
	<b>-</b>	不能在用户程序中使用该变量。
<b>有效</b>	指定变量的更改何时生效。	
<b>HMI</b>	可以在 HMI 系统中使用该变量。	

<b>&lt;轴名称&gt;.ErrorBits.SystemFault</b>				
TRUE = 内部系统错误。				
<b>数据类型</b>	<b>起始值</b>	<b>访问</b>	<b>有效</b>	<b>HMI</b>
Bool	FALSE	R	-	√

<b>&lt;轴名称&gt;.ErrorBits.ConfigFault</b>				
TRUE = 轴组态不正确。				
<b>数据类型</b>	<b>起始值</b>	<b>访问</b>	<b>有效</b>	<b>HMI</b>
Bool	FALSE	R	-	√

<b>&lt;轴名称&gt;.ErrorBits.DriveFault</b>				
TRUE = 驱动器因“驱动器准备就绪”信号丢失而报告了错误。				
<b>数据类型</b>	<b>起始值</b>	<b>访问</b>	<b>有效</b>	<b>HMI</b>
Bool	FALSE	R	-	√

<b>&lt;轴名称&gt;.ErrorBits.SwLimitMinReached</b>				
TRUE = 已到达下限软限位开关。				
<b>数据类型</b>	<b>起始值</b>	<b>访问</b>	<b>有效</b>	<b>HMI</b>
Bool	FALSE	R	-	√

<b>&lt;轴名称&gt;.ErrorBits.SwLimitMinExceeded</b>				
TRUE = 已超出下限软限位开关。				
数据类型	起始值	访问	有效	HMI
Bool	FALSE	R	-	√

<b>&lt;轴名称&gt;.ErrorBits.SwLimitMaxReached</b>				
TRUE = 已到达上限软限位开关。				
数据类型	起始值	访问	有效	HMI
Bool	FALSE	R	-	√

<b>&lt;轴名称&gt;.ErrorBits.SwLimitMaxExceeded</b>				
TRUE = 已超出上限软限位开关。				
数据类型	起始值	访问	有效	HMI
Bool	FALSE	R	-	√

<b>&lt;轴名称&gt;.ErrorBits.HwLimitMin</b>				
TRUE = 已逼近下限硬限位开关。				
数据类型	起始值	访问	有效	HMI
Bool	FALSE	R	-	√

<b>&lt;轴名称&gt;.ErrorBits.HwLimitMax</b>				
TRUE = 已逼近上限硬限位开关。				
数据类型	起始值	访问	有效	HMI
Bool	FALSE	R	-	√

<b>&lt;轴名称&gt;.ErrorBits.HwUsed</b>				
TRUE = 另一个轴正在使用相同的 PTO (Pulse Train Output) 并且该轴已使用“MC_Power”启用。				
数据类型	起始值	访问	有效	HMI
Bool	FALSE	R	-	√

## 参见

状态和错误位（自工艺对象 V4 起） (页 183)

自 V4 起的定位轴工艺对象的变量 (页 235)

## 内部变量 V1...3

“Internal”变量不包含用户相关数据；无法在用户程序中访问这些变量。

## 参见

自 V4 起的定位轴工艺对象的变量 (页 235)

## ControlPanel 变量 V1...3

“ControlPanel”变量不包含用户相关数据；无法在用户程序中访问这些变量。

## 参见

自 V4 起的定位轴工艺对象的变量 (页 235)

## 工艺对象变量的更新

工艺对象变量中指示的轴的状态和错误信息在各周期控制点处更新。

可编辑组态变量值的变化不立即生效。

有关更改生效条件的信息，请参见相关变量的详细说明。

1.11.10.6 命令表工艺对象 V1...3 的变量

Config.Command.Command[1 ... 32] 变量 V1...3

图例

<b>数据类型</b>	变量的数据类型		
<b>起始值</b>	变量的起始值 起始值可由命令表的组态覆盖。		
<b>访问</b>	在用户程序中访问变量的方式:		
	RW	可以在用户程序中读写该变量。	
	R	可以在用户程序中读取该变量。	
	-	不能在用户程序中使用该变量。	
<b>有效</b>	指定变量的更改何时生效。		
<b>HMI</b>	可以在 HMI 系统中使用该变量。		

<b>&lt;命令表&gt;.Config.Command.Command[x].Type</b>				
命令类型				
<ul style="list-style-type: none"> <li>• 0 =“空”命令</li> <li>• 2 =“保持”命令</li> <li>• 5 =“相对定位”命令</li> <li>• 6 =“绝对定位”命令</li> <li>• 7 =“速度设定值”命令</li> <li>• 151 =“等待”命令</li> </ul>				
<b>数据类型</b>	<b>起始值</b>	<b>访问</b>	<b>有效</b>	<b>HMI</b>
Int	0	RW	-	√

<b>&lt;命令表&gt;. Config.Command.Command[x].Position</b>				
目标位置/行进路径命令				
<b>数据类型</b>	<b>起始值</b>	<b>访问</b>	<b>有效</b>	<b>HMI</b>
Real		RW	-	√

<b>&lt;命令表&gt;. Config.Command.Command[x].Velocity</b>				
速度命令				
数据类型	起始值	访问	有效	HMI
Real	0.0	RW	-	√

<b>&lt;命令表&gt;. Config.Command.Command[x].Duration</b>				
持续时间命令				
数据类型	起始值	访问	有效	HMI
Real	0.0	RW	-	√

<b>&lt;命令表&gt;. Config.Command.Command[x].BufferMode</b>				
“下一步”命令的值				
<ul style="list-style-type: none"> <li>• 0 =“完成命令”</li> <li>• 1 =“混合运动”</li> </ul>				
数据类型	起始值	访问	有效	HMI
Int	0	RW	-	√

<b>&lt;命令表&gt;. Config.Command.Command[x].StepCode</b>				
命令步代码				
数据类型	起始值	访问	有效	HMI
Word	0	RW	-	√

## 参见

自 V4 起的命令表工艺对象变量 (页 288)

## S7-1200 运动控制

### 2.1 S7-1200 运动控制 V4 及更高版本

#### 2.1.1 MC\_Power

##### 2.1.1.1 MC\_Power: 启用、禁用轴 V4 及更高版本

#### 说明

“MC\_Power”运动控制指令可启用或禁用轴。

#### 要求

- 定位轴工艺对象已正确组态。
- 没有待决的启用/禁止错误。

#### 超驰响应

运动控制命令无法中止“MC\_Power”的执行。

禁用轴（输入参数“Enable”= FALSE）之后，将根据所选“StopMode”中止相关工艺对象的所有运动控制命令。



## 参数

参数	声明	数据类型	默认值	说明	
Axis	INPUT	TO_Axis	-	轴工艺对象	
Enable	INPUT	BOOL	FALSE	TRUE	轴已启用。
				FALSE	根据组态的“StopMode”中断当前所有作业。停止并禁用轴。
StopMode	INPUT	INT	0	0	<p>紧急停止</p> <p>如果禁用轴的请求处于待决状态，则轴将以组态的急停减速度进行制动。轴在变为静止状态后被禁用。</p>
				1	<p>立即停止</p> <p>如果禁用轴的请求处于待决状态，则会输出该设定值 0，并禁用轴。轴将根据驱动器中的组态进行制动，并转入停止状态。</p> <p>对于通过 PTO (Pulse Train Output) 的驱动器连接： 禁用轴时，将根据基于频率的减速度，停止脉冲输出：</p> <ul style="list-style-type: none"> <li>• 输出频率 <math>\geq 100</math> Hz 减速度：最长 30 ms</li> <li>• 输出频率 <math>&lt; 100</math> Hz 减速度：30 ms；2 Hz 时，最长 1.5 s</li> </ul>
				2	<p>带有加速度变化率控制的紧急停止</p> <p>如果禁用轴的请求处于待决状态，则轴将以组态的急停减速度进行制动。如果激活了加速度变化率控制，会将已组态的加速度变化率考虑在内。轴在变为静止状态后被禁用。</p>

参数	声明	数据类型	默认值	说明	
Status	OUTPUT	BOOL	FALSE	轴的使能状态	
				FALSE	禁用轴。 轴不会执行运动控制命令也不会接受任何新命令（例外：MC_Reset 命令）。 通过 PTO (Pulse Train Output) 的驱动器连接： 轴未回原点。 在禁用轴时，只有在轴停止之后，才会将状态更改为 FALSE。
				TRUE	轴已启用。 轴已就绪，可以执行运动控制命令。 在启用轴时，直到信号“驱动器准备就绪”处于待决状态之后，才会将状态更改为 TRUE。 在轴组态中，如果未组态“驱动器准备就绪”驱动器接口，那么状态将会立即更改为 TRUE。
Busy	OUTPUT	BOOL	FALSE	TRUE “MC_Power”处于活动状态。	
Error	OUTPUT	BOOL	FALSE	TRUE 运动控制指令“MC_Power”或相关工艺对象发生错误。 错误原因，请参见“ErrorID”和“ErrorInfo”的参数说明。	
ErrorID	OUTPUT	WORD	16#0000	参数“Error”的错误 ID (页 205)	
ErrorInfo	OUTPUT	WORD	16#0000	参数“ErrorID”的错误信息 ID (页 205)	

### 说明

如果因出错而停止轴，那么在清除并确认错误之后，会再次自动启用轴。  
这要求输入参数“Enable”的值在该过程中保持为 TRUE。

### 启用带有已组态驱动器接口的轴

要启用轴，请按下列步骤操作：

1. 首先检查是否满足上述要求。
2. 使用所需值对输入参数“StopMode”进行初始化。将输入参数“Enable”设置为 TRUE。

将“启用驱动器”的使能输出更改为 TRUE，以接通驱动器的电源。

CPU 将等待驱动器的“驱动器就绪”信号。

当 CPU 组态完成且输入端出现“驱动器就绪”信号时，将启用轴。

输出参数“Status”和工艺对象变量 <轴名称>.StatusBits.Enable 的值为 TRUE。

### 启用不带已组态驱动器接口的轴

要启用轴，请按下列步骤操作：

1. 首先检查是否满足上述要求。
2. 使用所需值对输入参数“StopMode”进行初始化。将输入参数“Enable”设置为 TRUE。

轴已启用。输出参数“Status”和工艺对象变量 <轴名称>.StatusBits.Enable 的值为 TRUE。

## 禁用轴

要禁用轴，可以按照下列步骤操作：

1. 停止轴。  
可以通过工艺对象变量 `<轴名称>.StatusBits.StandStill` 识别轴何时处于停止状态。
2. 在轴停止后，将输入参数“Enable”设置为 FALSE。
3. 如果输出参数“Busy”和“Status”以及工艺对象变量 `<轴名称>.StatusBits.Enable` 的值均为 FALSE，则说明禁用轴已完成。

## 参见

运动控制语句概述 (页 162)

ErrorID 和 ErrorInfo 列表（自工艺对象 V4 起） (页 205)

MC\_Power: V4 的功能图 (页 373)

MC\_Reset: 确认故障 V4 及更高版本 (页 374)

MC\_Home: 使轴归位，设置参考点 V4 及更高版本 (页 377)

MC\_Halt: 停止轴 V4 及更高版本 (页 382)

MC\_MoveAbsolute: 轴的绝对定位 V4 及更高版本 (页 386)

MC\_MoveRelative: 轴的相对定位 V4 及更高版本 (页 391)

MC\_MoveVelocity: 以预设的旋转速度移动轴 V4 及更高版本 (页 396)

MC\_MoveJog: 在点动模式下移动轴 V4 及更高版本 (页 402)

MC\_ChangeDynamic: 更改轴的动态设置 V4 及更高版本 (页 409)

MC\_CommandTable: 按照运动顺序运行轴命令 V4 及更高版本 (页 406)

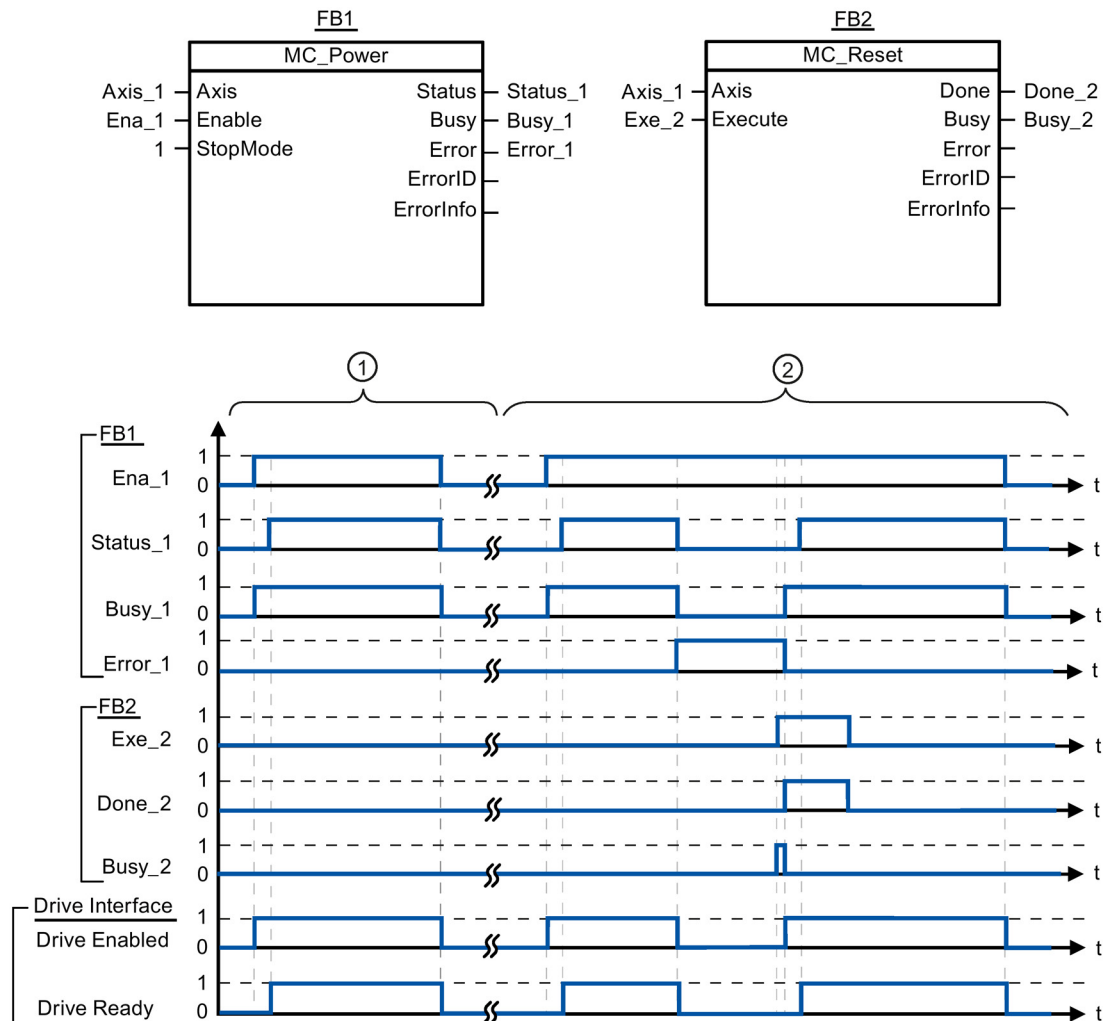
MC\_ReadParam: 连续读取定位轴的运动数据 V4 及更高版本 (页 413)

MC\_WriteParam: 写入定位轴的变量 V4 及更高版本 (页 416)

S7-1200 运动控制 V1...3 (页 419)

## 2.1.1.2 MC\_Power: V4 的功能图

## 功能图



①	启用轴之后再次禁用。驱动器将“驱动器准备就绪”信号发送回 CPU 后，可以通过“Status_1”标识已成功启用轴。
②	在启用轴之后，如果发生错误，则会禁用轴。在消除错误并通过“MC_Reset”确认后，然后该轴再次被启用。

## 参见

MC\_Power: 启用、禁用轴 V4 及更高版本 (页 368)

## 2.1.2 MC\_Reset

### 2.1.2.1 MC\_Reset: 确认故障 V4 及更高版本

#### 说明

运动控制指令“MC\_Reset”可用于确认“伴随轴停止出现的运行错误”和“组态错误”。关于这些需要进行确认的错误，请参见“解决方法”(Remedy) 下的“ErrorID 和 ErrorInfo 列表”。

在 RUN 模式下完成下载后，可将轴组态下载到工作存储器。

#### 要求

- 定位轴工艺对象已正确组态。
- 已经清除了引起这些需确认的待决组态错误的原因（例如，已将定位轴工艺对象中的加速度更改为有效值）。

#### 超驰响应

任何其它运动控制命令均无法中止 MC\_Reset 命令。

新的 MC\_Reset 命令不会中止任何其它激活的运动控制命令。

## 参数

参数	声明	数据类型	默认值	说明	
Axis	INPUT	TO_Axis	-	轴工艺对象	
Execute	INPUT	BOOL	FALSE	上升沿时启动命令	
Restart	INPUT	BOOL	FALSE	TRUE	将轴组态从装载存储器下载到工作存储器。 仅可在禁用轴后，才能执行该命令。 请参见有关下载到 CPU (页 153) 的说明。
				FALSE	确认待决的错误
Done	OUTPUT	BOOL	FALSE	TRUE	错误已确认。
Busy	OUTPUT	BOOL	FALSE	TRUE	命令正在执行
Error	OUTPUT	BOOL	FALSE	TRUE	执行命令期间出错。 错误原因，请参见“ErrorID”和“ErrorInfo”的参数说明。
ErrorID	OUTPUT	WORD	16#0000	参数“Error”的错误 ID (页 205)	
ErrorInfo	OUTPUT	WORD	16#0000	参数“ErrorID”的错误信息 ID (页 205)	

### 可通过 MC\_Reset 参数，对需要进行确认的错误进行确认

要确认错误，请按照下列步骤操作：

1. 首先检查是否满足上述要求。
2. 从输入参数“Execute”的上升沿开始确认错误。
3. 如果输出参数“Done”的值为 TRUE，同时工艺对象变量 <轴名称>.StatusBits.Error 的值为 FALSE，则说明错误已被确认。

### 参见

运动控制语句概述 (页 162)

ErrorID 和 ErrorInfo 列表（自工艺对象 V4 起） (页 205)

下载到 CPU (页 153)

MC\_Power: 启用、禁用轴 V4 及更高版本 (页 368)

MC\_Home: 使轴归位，设置参考点 V4 及更高版本 (页 377)

MC\_Halt: 停止轴 V4 及更高版本 (页 382)

MC\_MoveAbsolute: 轴的绝对定位 V4 及更高版本 (页 386)

MC\_MoveRelative: 轴的相对定位 V4 及更高版本 (页 391)

MC\_MoveVelocity: 以预设的旋转速度移动轴 V4 及更高版本 (页 396)

MC\_MoveJog: 在点动模式下移动轴 V4 及更高版本 (页 402)

MC\_CommandTable: 按照运动顺序运行轴命令 V4 及更高版本 (页 406)

MC\_ChangeDynamic: 更改轴的动态设置 V4 及更高版本 (页 409)

MC\_ReadParam: 连续读取定位轴的运动数据 V4 及更高版本 (页 413)

MC\_WriteParam: 写入定位轴的变量 V4 及更高版本 (页 416)

S7-1200 运动控制 V1...3 (页 419)



## 2.1.3 MC\_Home

### 2.1.3.1 MC\_Home: 使轴归位, 设置参考点 V4 及更高版本

#### 说明

使用“MC\_Home”运动控制指令可将轴坐标与实际物理驱动器位置匹配。轴的绝对定位需要归位。可执行以下类型的归位：

- 主动归位 (Mode = 3)  
自动执行归位步骤。
- 被动归位 (Mode = 2)  
被动归位期间，运动控制指令“MC\_Home”不会执行任何归位运动。用户必须通过其它运动控制指令，执行这一步骤中所需的往返运动。检测到归位开关时，轴即归位。
- 直接绝对归位 (Mode = 0)  
将当前的轴位置设置为参数“Position”的值。
- 直接相对归位 (Mode = 1)  
将当前轴位置的偏移值设置为参数“Position”的值。

#### 要求

- 定位轴工艺对象已正确组态。
- 轴已启用。
- 以 Mode = 0、1 或 2 启动时不会任何激活 MC\_CommandTable 命令。

## 超驰响应

超驰响应取决于所选的模式：

### Mode = 0, 1

任何其它运动控制命令均无法中止 MC\_Home 命令。

MC\_Home 命令不会中止任何激活的运动控制命令。

按照新的归位位置（输入参数“Position”的值）进行归位操作后，将继续执行与位置相关的运动命令。"Position").

### Mode = 2

可通过下列运动控制命令中止 MC\_Home 命令：

- MC\_Home 命令 Mode = 2、3

新的 MC\_Home 命令可中止下列激活的运动控制命令：

- MC\_Home 命令 Mode = 2

按照新的归位位置（输入参数“Position”的值）进行归位操作后，将继续执行与位置相关的运动命令。"Position").

### Mode = 3

可通过下列运动控制命令中止 MC\_Home 命令：

- MC\_Home 命令 Mode = 3
- MC\_Halt 命令
- MC\_MoveAbsolute 命令
- MC\_MoveRelative 命令
- MC\_MoveVelocity 命令
- MC\_MoveJog 命令
- MC\_CommandTable 命令

新的 MC\_Home 命令可中止下列激活的运动控制命令：

- MC\_Home 命令 Mode = 2、3
- MC\_Halt 命令
- MC\_MoveAbsolute 命令
- MC\_MoveRelative 命令
- MC\_MoveVelocity 命令
- MC\_MoveJog 命令
- MC\_CommandTable 命令

## 参数

参数	声明	数据类型	默认值	说明	
Axis	INPUT	TO_Axis	-	轴工艺对象	
Execute	INPUT	BOOL	FALSE	上升沿时启动作业	
Position	INPUT	REAL	0.0	<ul style="list-style-type: none"> <li>Mode = 0、2 和 3 完成归位操作之后，轴的绝对位置</li> <li>Mode = 1 对当前轴位置的修正值</li> </ul> 限值： $-1.0e^{12} \leq \text{Position} \leq 1.0e^{12}$	
Mode	INPUT	INT	0	归位模式	
				0	绝对式直接归位 新的轴位置为参数“Position”位置的 位置的值。
				1	相对式直接归位 新的轴位置等于当前轴位置 + 参数“Position”位置的值。
				2	被动归位 将根据轴组态进行归位。 归位后，将新的轴位置设置为参数 “Position”的值。
3	主动归位 按照轴组态进行归位操作。 归位后，将新的轴位置设置为参数 “Position”的值。				
Done	OUTPUT	BOOL	FALSE	TRUE	命令已完成
Busy	OUTPUT	BOOL	FALSE	TRUE	命令正在执行
Command- Aborted	OUTPUT	BOOL	FALSE	TRUE	作业在执行过程中被另一作业中止。
Error	OUTPUT	BOOL	FALSE	TRUE	执行命令期间出错。 错误原因，请参见“ErrorID”和 “ErrorInfo”的参数说明。

参数	声明	数据类型	默认值	说明
ErrorID	OUTPUT	WORD	16#0000	参数“Error”的错误 ID (页 205)
ErrorInfo	OUTPUT	WORD	16#0000	参数“ErrorID”的错误信息 ID (页 205)

## 复位“归位后”状态

以下条件下复位工艺对象 (<轴名称>.StatusBits.HomingDone) 的“归位后”状态:

- **通过 PTO (Pulse Train Output) 的驱动器连接:**
  - 启动“MC\_Home”命令进行主动归位  
(成功完成归位操作之后,“归位后”状态被复位。)
  - 通过“MC\_Power”运动控制指令禁用轴
  - 在自动模式和手动控制之间切换
  - 关闭之后 -> 启动 CPU
  - CPU 重新启动后 (RUN-STOP -> STOP-RUN)
- **带增量实际值的工艺对象:**
  - 启动“MC\_Home”命令进行主动归位  
(成功完成归位操作之后,“归位后”状态被复位。)
  - 编码器系统内出现错误,或者编码器故障
  - 重新启动工艺对象
  - 在重新启动 CPU 后
  - 存储器复位
  - 修改编码器组态
- **带绝对实际值的工艺对象:**
  - 将 CPU 恢复为出厂设置
  - 修改编码器组态
  - 更换 CPU

## 使轴归位

要使轴归位，请按下列步骤操作：

1. 首先检查是否满足上述要求。
2. 使用这些值提供所需的输入参数，然后在输入参数“Execute”的上升沿开始归位。
3. 如果输出参数“Done”和工艺对象变量 <轴名称>.StatusBits.HomingDone 的值为 TRUE，则说明归位已完成。

## 参见

运动控制语句概述 (页 162)

ErrorID 和 ErrorInfo 列表（自工艺对象 V4 起） (页 205)

MC\_Power: 启用、禁用轴 V4 及更高版本 (页 368)

MC\_Reset: 确认故障 V4 及更高版本 (页 374)

MC\_Halt: 停止轴 V4 及更高版本 (页 382)

MC\_MoveAbsolute: 轴的绝对定位 V4 及更高版本 (页 386)

MC\_MoveRelative: 轴的相对定位 V4 及更高版本 (页 391)

MC\_MoveVelocity: 以预设的旋转速度移动轴 V4 及更高版本 (页 396)

MC\_MoveJog: 在点动模式下移动轴 V4 及更高版本 (页 402)

MC\_CommandTable: 按照运动顺序运行轴命令 V4 及更高版本 (页 406)

MC\_ChangeDynamic: 更改轴的动态设置 V4 及更高版本 (页 409)

MC\_ReadParam: 连续读取定位轴的运动数据 V4 及更高版本 (页 413)

MC\_WriteParam: 写入定位轴的变量 V4 及更高版本 (页 416)

S7-1200 运动控制 V1...3 (页 419)

## 2.1.4 MC\_Halt

### 2.1.4.1 MC\_Halt: 停止轴 V4 及更高版本

#### 说明

通过运动控制指令“MC\_Halt”，可停止所有运动并以组态的减速度停止轴。  
未定义停止位置。

#### 要求

- 定位轴工艺对象已正确组态。
- 轴已启用。

#### 超驰响应

可通过下列运动控制命令中止 MC\_Halt 命令：

- MC\_Home 命令 Mode = 3
- MC\_Halt 命令
- MC\_MoveAbsolute 命令
- MC\_MoveRelative 命令
- MC\_MoveVelocity 命令
- MC\_MoveJog 命令
- MC\_CommandTable 命令

新的 MC\_Halt 命令可中止下列激活的运动控制命令：

- MC\_Home 命令 Mode = 3
- MC\_Halt 命令
- MC\_MoveAbsolute 命令
- MC\_MoveRelative 命令
- MC\_MoveVelocity 命令
- MC\_MoveJog 命令
- MC\_CommandTable 命令

## 参数

参数	声明	数据类型	默认值	说明
Axis	INPUT	TO_SpeedAxis	-	轴工艺对象
Execute	INPUT	BOOL	FALSE	上升沿时启动命令
Done	OUTPUT	BOOL	FALSE	TRUE 速度达到零
Busy	OUTPUT	BOOL	FALSE	TRUE 正在执行命令。
CommandAborted	OUTPUT	BOOL	FALSE	TRUE 命令在执行过程中被另一命令中止。
Error	OUTPUT	BOOL	FALSE	TRUE 执行命令期间出错。 错误原因，请参见“ErrorID”和“ErrorInfo”的参数说明。
ErrorID	OUTPUT	WORD	16#0000	参数“Error”的错误 ID (页 205)
ErrorInfo	OUTPUT	WORD	16#0000	参数“ErrorID”的错误信息 ID (页 205)

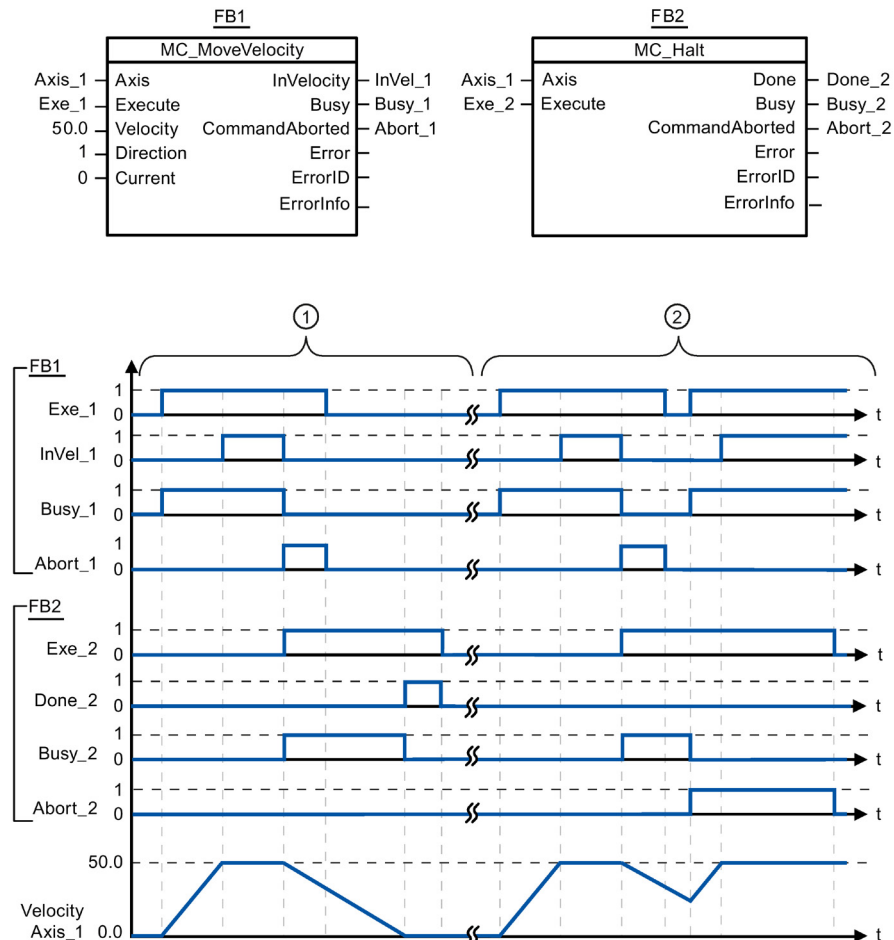
参见

- 运动控制语句概述 (页 162)
- ErrorID 和 ErrorInfo 列表 (自工艺对象 V4 起) (页 205)
- MC\_Halt: V4 的功能图 (页 385)
- MC\_Power: 启用、禁用轴 V4 及更高版本 (页 368)
- MC\_Reset: 确认故障 V4 及更高版本 (页 374)
- MC\_Home: 使轴归位, 设置参考点 V4 及更高版本 (页 377)
- MC\_MoveAbsolute: 轴的绝对定位 V4 及更高版本 (页 386)
- MC\_MoveRelative: 轴的相对定位 V4 及更高版本 (页 391)
- MC\_MoveVelocity: 以预设的旋转速度移动轴 V4 及更高版本 (页 396)
- MC\_MoveJog: 在点动模式下移动轴 V4 及更高版本 (页 402)
- MC\_CommandTable: 按照运动顺序运行轴命令 V4 及更高版本 (页 406)
- MC\_ChangeDynamic: 更改轴的动态设置 V4 及更高版本 (页 409)
- MC\_ReadParam: 连续读取定位轴的运动数据 V4 及更高版本 (页 413)
- MC\_WriteParam: 写入定位轴的变量 V4 及更高版本 (页 416)
- S7-1200 运动控制 V1...3 (页 419)



## 2.1.4.2 MC\_Halt: V4 的功能图

## 功能图



在组态窗口“动态 > 常规”(Dynamics > General) 中，对下列值进行组态：

- 加速度： 10.0
- 减速度： 5.0

①	轴将由 MC_Halt 命令制动，直至停止为止。通过“Done_2”发出轴停止的信号。
②	当 MC_Halt 命令对轴进行制动处理时，另一个运动命令会中止该命令。将通过“Abort_2”发出中止信号。

## 参见

MC\_Halt: 停止轴 V4 及更高版本 (页 382)

## 2.1.5 MC\_MoveAbsolute

### 2.1.5.1 MC\_MoveAbsolute: 轴的绝对定位 V4 及更高版本

#### 说明

运动控制指令“MC\_MoveAbsolute”启动轴定位运动，以将轴移动到某个绝对位置。

#### 要求

- 定位轴工艺对象已正确组态。
- 轴已启用。
- 轴已回原点。

#### 超驰响应

可通过下列运动控制命令中止 MC\_MoveAbsolute 命令：

- MC\_Home 命令 Mode = 3
- MC\_Halt 命令
- MC\_MoveAbsolute 命令
- MC\_MoveRelative 命令
- MC\_MoveVelocity 命令
- MC\_MoveJog 命令
- MC\_CommandTable 命令

新的 MC\_MoveAbsolute 命令可中止下列激活的运动控制命令：

- MC\_Home 命令 Mode = 3
- MC\_Halt 命令
- MC\_MoveAbsolute 命令
- MC\_MoveRelative 命令
- MC\_MoveVelocity 命令
- MC\_MoveJog 命令
- MC\_CommandTable 命令

## 参数

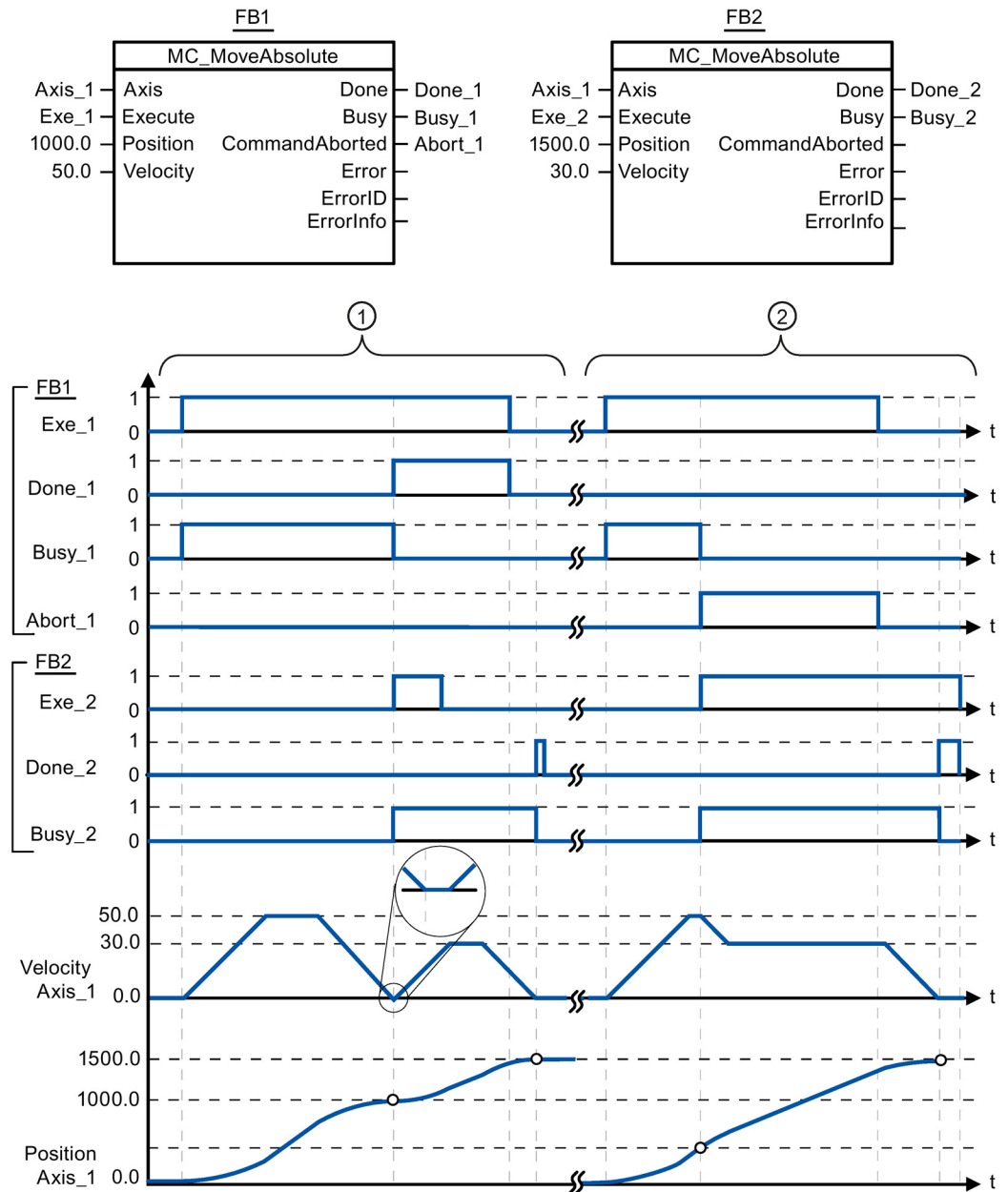
参数	声明	数据类型	默认值	说明
Axis	INPUT	TO_Positioning Axis	-	轴工艺对象
Execute	INPUT	BOOL	FALSE	上升沿时启动命令
Position	INPUT	REAL	0.0	绝对目标位置 限值： $-1.0e^{12} \leq \text{Position} \leq 1.0e^{12}$
Velocity	INPUT	REAL	10.0	轴的速度 由于所组态的加速度和减速度以及待接近的目标位置等原因， 不会始终保持这一速度。 限值： 启动/停止速度 $\leq \text{Velocity} \leq$ 最大速度
Done	OUTPUT	BOOL	FALSE	TRUE   达到绝对目标位置
Busy	OUTPUT	BOOL	FALSE	TRUE   正在执行命令。
CommandAborted	OUTPUT	BOOL	FALSE	TRUE   命令在执行过程中被另一命令中止。
Error	OUTPUT	BOOL	FALSE	TRUE   执行命令期间出错。 错误原因，请参见“ErrorID”和“ErrorInfo”的参数说明。
ErrorID	OUTPUT	WORD	16#0000	参数“Error”的错误 ID (页 205)
ErrorInfo	OUTPUT	WORD	16#0000	参数“ErrorID”的错误信息 ID (页 205)

## 参见

- 运动控制语句概述 (页 162)
- ErrorID 和 ErrorInfo 列表 (自工艺对象 V4 起) (页 205)
- MC\_MoveAbsolute: V4 的功能图 (页 389)
- MC\_Power: 启用、禁用轴 V4 及更高版本 (页 368)
- MC\_Reset: 确认故障 V4 及更高版本 (页 374)
- MC\_Home: 使轴归位, 设置参考点 V4 及更高版本 (页 377)
- MC\_Halt: 停止轴 V4 及更高版本 (页 382)
- MC\_MoveRelative: 轴的相对定位 V4 及更高版本 (页 391)
- MC\_MoveVelocity: 以预设的旋转速度移动轴 V4 及更高版本 (页 396)
- MC\_MoveJog: 在点动模式下移动轴 V4 及更高版本 (页 402)
- MC\_CommandTable: 按照运动顺序运行轴命令 V4 及更高版本 (页 406)
- MC\_ChangeDynamic: 更改轴的动态设置 V4 及更高版本 (页 409)
- MC\_ReadParam: 连续读取定位轴的运动数据 V4 及更高版本 (页 413)
- MC\_WriteParam: 写入定位轴的变量 V4 及更高版本 (页 416)
- S7-1200 运动控制 V1...3 (页 419)

## 2.1.5.2 MC\_MoveAbsolute: V4 的功能图

## 功能图



在组态窗口“动态 > 常规”(Dynamics > General) 中，对下列值进行组态：

- 加速度： 10.0
- 减速度： 10.0

①	<p>通过 MC_MoveAbsolute 命令，可将轴移动到绝对位置 1000.0。</p> <p>当轴达到目标位置后，将通过“Done_1”发出信号。</p> <p>当“Done_1” = TRUE 时，将启动另一个目标位置为 1500.0 的 MC_MoveAbsolute 命令。</p> <p>由于响应时间的原因（如用户程序的循环时间等），轴将暂时停止（参见放大后的详细信息）。</p> <p>当轴达到新的目标位置后，将通过“Done_2”发出信号。</p>
②	<p>可通过另一个 MC_MoveAbsolute 命令中止激活的 MC_MoveAbsolute 命令。</p> <p>将通过“Abort_1”发出中止信号。之后，轴将会以新的速度移动到新的目标位置 1500.0。</p> <p>到达新的目标位置之后，将通过“Done_2”发出信号。</p>

### 参见

MC\_MoveAbsolute: 轴的绝对定位 V4 及更高版本 (页 386)

## 2.1.6 MC\_MoveRelative

### 2.1.6.1 MC\_MoveRelative: 轴的相对定位 V4 及更高版本

#### 说明

通过运动控制指令“MC\_MoveRelative”，启动相对于起始位置的定位运动。

#### 要求

- 定位轴工艺对象已正确组态。
- 轴已启用。

#### 超驰响应

可通过下列运动控制命令中止 MC\_MoveRelative 命令：

- MC\_Home 命令 Mode = 3
- MC\_Halt 命令
- MC\_MoveAbsolute 命令
- MC\_MoveRelative 命令
- MC\_MoveVelocity 命令
- MC\_MoveJog 命令
- MC\_CommandTable 命令

新的 MC\_MoveRelative 命令可中止下列激活的运动控制命令：

- MC\_Home 命令 Mode = 3
- MC\_Halt 命令
- MC\_MoveAbsolute 命令
- MC\_MoveRelative 命令
- MC\_MoveVelocity 命令
- MC\_MoveJog 命令
- MC\_CommandTable 命令

## 参数

参数	声明	数据类型	默认值	说明
Axis	INPUT	TO_Positioning-Axis	-	轴工艺对象
Execute	INPUT	BOOL	FALSE	上升沿时启动命令
Distance	INPUT	REAL	0.0	定位操作的移动距离 限值： $-1.0e^{12} \leq \text{Distance} \leq 1.0e^{12}$
Velocity	INPUT	REAL	10.0	轴的速度 由于所组态的加速度和减速度以及要途经的距离等原因，不会始终保持这一速度。 限值： 启动/停止速度 $\leq$ Velocity $\leq$ 最大速度
Done	OUTPUT	BOOL	FALSE	TRUE   目标位置已到达
Busy	OUTPUT	BOOL	FALSE	TRUE   正在执行命令。
CommandAborted	OUTPUT	BOOL	FALSE	TRUE   命令在执行过程中被另一命令中止。
Error	OUTPUT	BOOL	FALSE	TRUE   执行命令期间出错。 错误原因，请参见“ErrorID”和“ErrorInfo”的参数说明。
ErrorID	OUTPUT	WORD	16#0000	参数“Error”的错误 ID (页 205)
ErrorInfo	OUTPUT	WORD	16#0000	参数“ErrorID”的错误信息 ID (页 205)

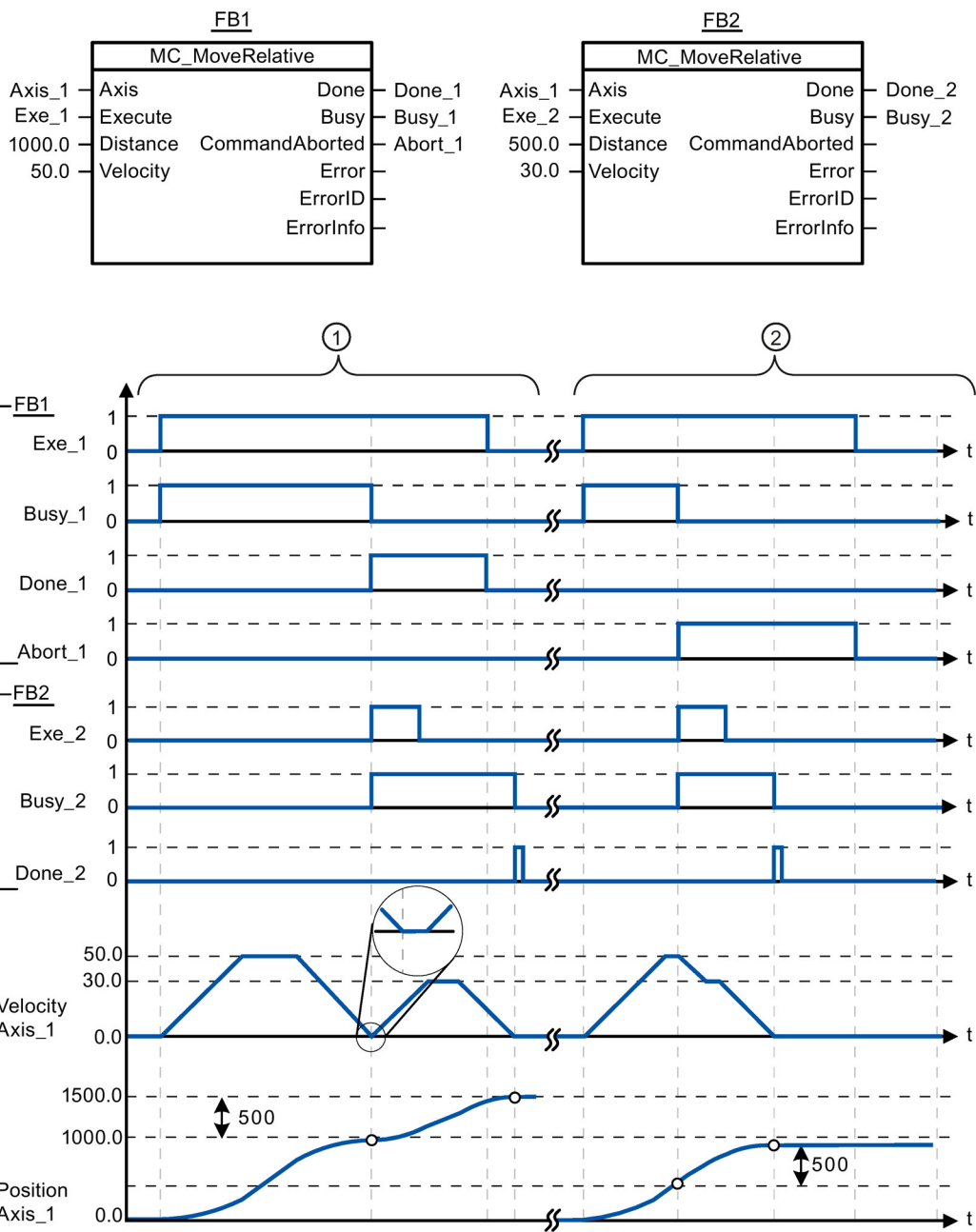


## 参见

- 运动控制语句概述 (页 162)
- ErrorID 和 ErrorInfo 列表 (自工艺对象 V4 起) (页 205)
- MC\_MoveRelative: V4 的功能图 (页 394)
- MC\_Power: 启用、禁用轴 V4 及更高版本 (页 368)
- MC\_Reset: 确认故障 V4 及更高版本 (页 374)
- MC\_Home: 使轴归位, 设置参考点 V4 及更高版本 (页 377)
- MC\_Halt: 停止轴 V4 及更高版本 (页 382)
- MC\_MoveAbsolute: 轴的绝对定位 V4 及更高版本 (页 386)
- MC\_MoveVelocity: 以预设的旋转速度移动轴 V4 及更高版本 (页 396)
- MC\_MoveJog: 在点动模式下移动轴 V4 及更高版本 (页 402)
- MC\_CommandTable: 按照运动顺序运行轴命令 V4 及更高版本 (页 406)
- MC\_ChangeDynamic: 更改轴的动态设置 V4 及更高版本 (页 409)
- MC\_ReadParam: 连续读取定位轴的运动数据 V4 及更高版本 (页 413)
- MC\_WriteParam: 写入定位轴的变量 V4 及更高版本 (页 416)
- S7-1200 运动控制 V1...3 (页 419)

2.1.6.2 MC\_MoveRelative: V4 的功能图

功能图



在组态窗口“动态 > 常规”(Dynamics > General) 中，对下列值进行组态：

- 加速度： 10.0
- 减速度： 10.0

①	<p>轴将通过 MC_MoveRelative 命令移动 1000.0 的距离 (“Distance”)。</p> <p>当轴达到目标位置后，将通过“Done_1”发出信号。“Done_1”= TRUE 时，将启动另一个行进距离为 500.0 的 MC_MoveRelative 命令。</p> <p>由于响应时间的原因（如用户程序的循环时间等），轴将暂时停止（参见放大后的详细信息）。当轴达到新的目标位置后，将通过“Done_2”发出信号。</p>
②	<p>可通过另一个 MC_MoveRelative 命令中止激活的 MC_MoveRelative 命令。</p> <p>将通过“Abort_1”发出中止信号。之后，轴将以新的速度移动 500.0 的距离 (“Distance”)。</p> <p>到达新的目标位置之后，将通过“Done_2”发出信号。</p>

### 参见

MC\_MoveRelative: 轴的相对定位 V4 及更高版本 (页 391)

## 2.1.7 MC\_MoveVelocity

### 2.1.7.1 MC\_MoveVelocity: 以预设的旋转速度移动轴 V4 及更高版本

#### 说明

通过运动控制指令“MC\_MoveVelocity”，以指定的速度连续移动轴。

#### 要求

- 定位轴工艺对象已正确组态。
- 轴已启用。

#### 超驰响应

MC\_MoveVelocity 可由下列运动控制命令中止：

- MC\_Home 命令 Mode = 3
- MC\_Halt 命令
- MC\_MoveAbsolute 命令
- MC\_MoveRelative 命令
- MC\_MoveVelocity 命令
- MC\_MoveJog 命令
- MC\_CommandTable 命令

新的 MC\_MoveVelocity 命令可中止下列激活的运动控制命令：

- MC\_Home 命令 Mode = 3
- MC\_Halt 命令
- MC\_MoveAbsolute 命令
- MC\_MoveRelative 命令
- MC\_MoveVelocity 命令
- MC\_MoveJog 命令
- MC\_CommandTable 命令

## 参数

参数	声明	数据类型	默认值	说明	
Axis	INPUT	TO_SpeedAxis	-	轴工艺对象	
Execute	INPUT	BOOL	FALSE	上升沿时启动命令	
Velocity	INPUT	REAL	10.0	轴运动的指定速度 限值： 启动/停止速度 $\leq$  Velocity  $\leq$ 最大速度 (允许 Velocity = 0.0)	
Direction	INPUT	INT	0	指定方向	
				0	旋转方向取决于参数“Velocity”值的符号
				1	正旋转方向 (将忽略参数“Velocity”值的符号)
				2	负旋转方向 (将忽略参数“Velocity”值的符号)
Current	INPUT	BOOL	FALSE	保持当前速度	
				FALSE	“保持当前速度”已禁用。 将使用参数“Velocity”和“Direction”的值。
				TRUE	“保持当前速度”已启用。 而不考虑参数“Velocity”和“Direction”的值。 当轴继续以当前速度运动时， 参数 "InVelocity" 返回值 TRUE。
InVelocity	OUTPUT	BOOL	FALSE	TRUE <ul style="list-style-type: none"> <li>"Current" = FALSE: 达到参数“Velocity”中指定的速度。</li> <li>"Current" = TRUE: 轴在启动时，以当前速度进行移动。</li> </ul>	
Busy	OUTPUT	BOOL	FALSE	TRUE	正在执行命令。
Command-Aborted	OUTPUT	BOOL	FALSE	TRUE	命令在执行过程中被另一命令中止。

参数	声明	数据类型	默认值	说明
Error	OUTPUT	BOOL	FALSE	TRUE 执行命令期间出错。 错误原因，请参见“ErrorID”和 “ErrorInfo”的参数说明。
ErrorID	OUTPUT	WORD	16#0000	参数“Error”的错误 ID (页 205)
ErrorInfo	OUTPUT	WORD	16#0000	参数“ErrorID”的错误信息 ID (页 205)

### 说明

#### PLCopen 版本 2.0

自 V4 起，运动控制指令“MC\_MoveVelocity”兼容 PLCopen 版本 2.0。

在命令被覆盖或因错误停止前，无论“Execute”参数如何，“InVelocity”和“Busy”参数都会显示其各自的状态。有关更多信息，请参见“跟踪激活命令 (页 173)”部分。

### 速度设定值为零 (Velocity = 0.0) 时的行为

“Velocity”= 0.0 的 MC\_MoveVelocity 命令（如 MC\_Halt 命令）可中止激活的运动命令，并使用所组态的减速度停止轴运动。

当轴停止时，至少在一个程序循环内将输出参数“InVelocity”的值指定为 TRUE。

“Busy”的值在减速过程中为 TRUE，并且随“InVelocity”一起变为 FALSE。

如果将参数“Execute”设置为 TRUE，则“InVelocity”和“Busy”处于锁定状态。

启动“MC\_MoveVelocity”命令时，将置位工艺对象的状态位“SpeedCommand”。

在轴停止时，设置状态位“ConstantVelocity”。

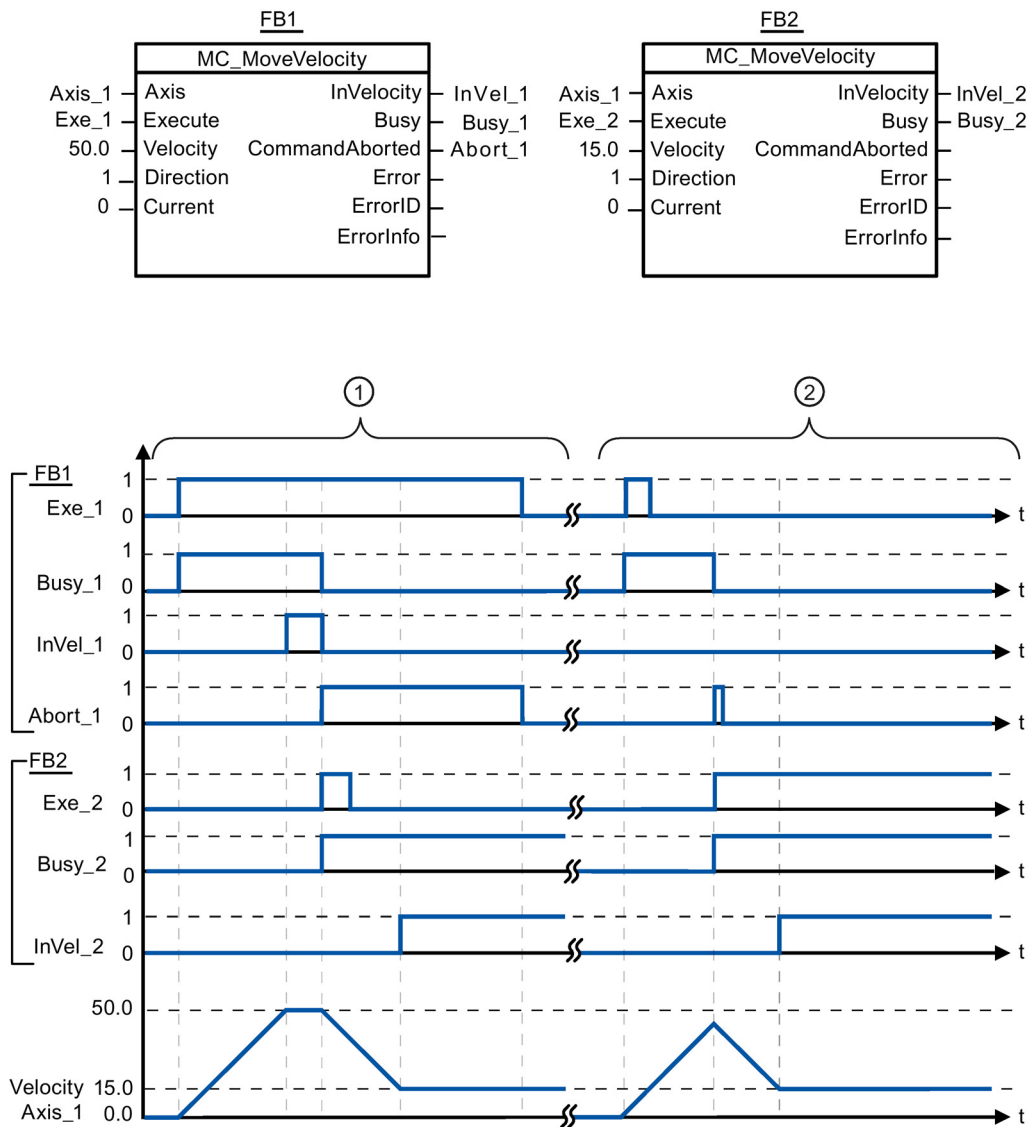
在启动新的运动命令后，将对这两个位进行调整，以适应新的情况。

## 参见

- 运动控制语句概述 (页 162)
- ErrorID 和 ErrorInfo 列表 (自工艺对象 V4 起) (页 205)
- MC\_MoveVelocity: V4 的功能图 (页 400)
- MC\_Power: 启用、禁用轴 V4 及更高版本 (页 368)
- MC\_Reset: 确认故障 V4 及更高版本 (页 374)
- MC\_Home: 使轴归位, 设置参考点 V4 及更高版本 (页 377)
- MC\_Halt: 停止轴 V4 及更高版本 (页 382)
- MC\_MoveAbsolute: 轴的绝对定位 V4 及更高版本 (页 386)
- MC\_MoveRelative: 轴的相对定位 V4 及更高版本 (页 391)
- MC\_MoveJog: 在点动模式下移动轴 V4 及更高版本 (页 402)
- MC\_CommandTable: 按照运动顺序运行轴命令 V4 及更高版本 (页 406)
- MC\_ChangeDynamic: 更改轴的动态设置 V4 及更高版本 (页 409)
- MC\_ReadParam: 连续读取定位轴的运动数据 V4 及更高版本 (页 413)
- MC\_WriteParam: 写入定位轴的变量 V4 及更高版本 (页 416)
- S7-1200 运动控制 V1...3 (页 419)

2.1.7.2 MC\_MoveVelocity: V4 的功能图

功能图



在组态窗口“动态 > 常规”(Dynamics > General) 中，对下列值进行组态：

- 加速度： 10.0
- 减速度： 10.0



①	激活的 MC_MoveVelocity 命令通过“InVel_1”发出信号，说明已达到目标速度。 该命令随后将由另一个 MC_MoveVelocity 命令中止。将通过“Abort_1”发出中止信号。 达到新的目标速度 15.0 之后，将通过“InVel_2”发出信号。之后，轴将以新的恒定速度继续移动。
②	在达到目标速度之前，当前 MC_MoveVelocity 命令将由另一个 MC_MoveVelocity 命令中止。 将通过“Abort_1”发出中止信号。达到新的目标速度 15.0 之后，将通过“InVel_2”发出信号。 之后，轴将以新的恒定速度继续移动。

## 参见

MC\_MoveVelocity: 以预设的旋转速度移动轴 V4 及更高版本 (页 396)

## 2.1.8 MC\_MoveJog

### 2.1.8.1 MC\_MoveJog: 在点动模式下移动轴 V4 及更高版本

#### 说明

通过运动控制指令“MC\_MoveJog”，在点动模式下以指定的速度连续移动轴。  
例如，可以使用该运动控制指令进行测试和调试。

#### 要求

- 定位轴工艺对象已正确组态。
- 轴已启用。

#### 超驰响应

可通过下列运动控制命令中止 MC\_MoveJog 命令：

- MC\_Home 命令 Mode = 3
- MC\_Halt 命令
- MC\_MoveAbsolute 命令
- MC\_MoveRelative 命令
- MC\_MoveVelocity 命令
- MC\_MoveJog 命令
- MC\_CommandTable 命令

新的 MC\_MoveJog 命令可中止下列激活的运动控制命令：

- MC\_Home 命令 Mode = 3
- MC\_Halt 命令
- MC\_MoveAbsolute 命令
- MC\_MoveRelative 命令
- MC\_MoveVelocity 命令
- MC\_MoveJog 命令
- MC\_CommandTable 命令

## 参数

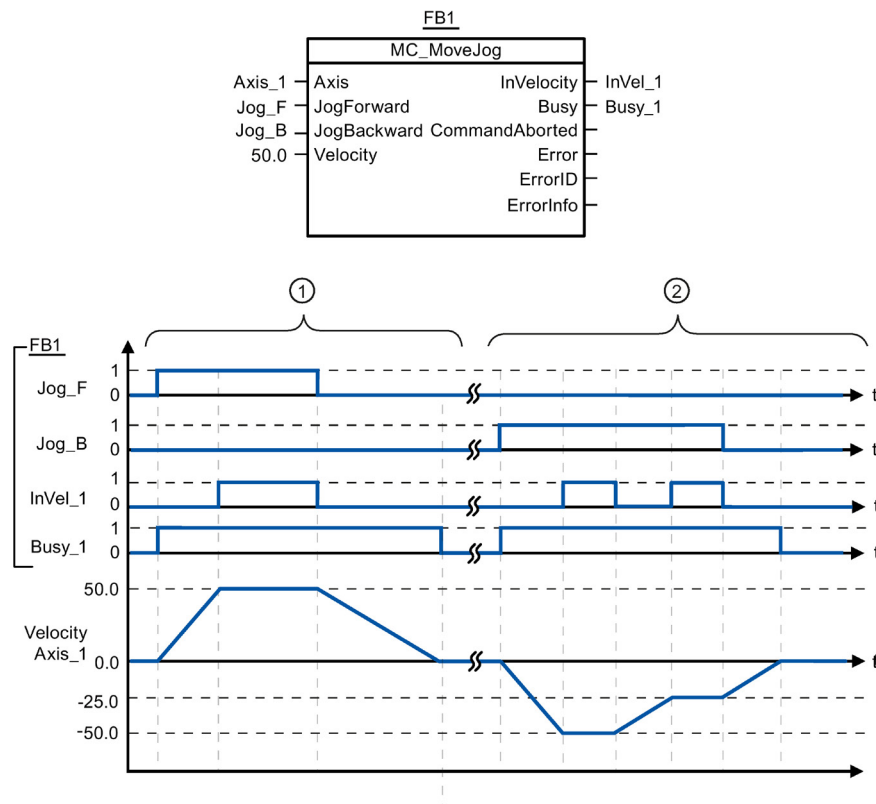
参数	声明	数据类型	默认值	说明
Axis	INPUT	TO_SpeedAxis	-	轴工艺对象
JogForward	INPUT	BOOL	FALSE	如果参数值为 TRUE， 则轴都将按参数“Velocity”中所指定的速度， 正向移动。
JogBackward	INPUT	BOOL	FALSE	如果参数值为 TRUE， 则轴都将按参数“Velocity”中指定的速度， 反向移动。
如果两个参数同时为 TRUE，轴将根据所组态的减速度直至停止。 通过参数“Error”、“ErrorID”和“ErrorInfo”，指出了错误。				
Velocity	INPUT	REAL	10.0	点动模式的预设速度 限值： 启动/停止速度 ≤ 速度 ≤ 最大速度
InVelocity	OUTPUT	BOOL	FALSE	TRUE 达到参数“Velocity”中指定的速度。
Busy	OUTPUT	BOOL	FALSE	TRUE 正在执行命令。
Command- Aborted	OUTPUT	BOOL	FALSE	TRUE 命令在执行过程中被另一 命令中止。
Error	OUTPUT	BOOL	FALSE	TRUE 执行命令期间出错。 错误原因，请参见“ErrorID”和 “ErrorInfo”的参数说明。
ErrorID	OUTPUT	WORD	16#0000	参数“Error”的错误 ID (页 205)
ErrorInfo	OUTPUT	WORD	16#0000	参数“ErrorID”的错误信息 ID (页 205)

## 参见

- 运动控制语句概述 (页 162)
- ErrorID 和 ErrorInfo 列表 (自工艺对象 V4 起) (页 205)
- MC\_MoveJog: V4 的功能图 (页 405)
- MC\_Power: 启用、禁用轴 V4 及更高版本 (页 368)
- MC\_Reset: 确认故障 V4 及更高版本 (页 374)
- MC\_Home: 使轴归位, 设置参考点 V4 及更高版本 (页 377)
- MC\_Halt: 停止轴 V4 及更高版本 (页 382)
- MC\_MoveAbsolute: 轴的绝对定位 V4 及更高版本 (页 386)
- MC\_MoveRelative: 轴的相对定位 V4 及更高版本 (页 391)
- MC\_MoveVelocity: 以预设的旋转速度移动轴 V4 及更高版本 (页 396)
- MC\_CommandTable: 按照运动顺序运行轴命令 V4 及更高版本 (页 406)
- MC\_ChangeDynamic: 更改轴的动态设置 V4 及更高版本 (页 409)
- MC\_ReadParam: 连续读取定位轴的运动数据 V4 及更高版本 (页 413)
- MC\_WriteParam: 写入定位轴的变量 V4 及更高版本 (页 416)
- S7-1200 运动控制 V1...3 (页 419)

## 2.1.8.2 MC\_MoveJog: V4 的功能图

## 功能图



在组态窗口“动态 > 常规”(Dynamics > General) 中，对下列值进行组态：

- 加速度： 10.0
- 减速度： 5.0

①	在点动模式下，轴通过“Jog_F”正向移动。达到目标速度 50.0 之后，将通过“InVel_1”发出信号。复位“Jog_F”之后，将制动轴并逐步停止。
②	在点动模式下，轴通过“Jog_B”反向移动。达到目标速度 -50.0 之后，将通过“InVel_1”发出信号。“Jog_B”置位后，参数“Velocity”的值会更改为 25.0。“InVel_1”复位后，将制动轴。达到新的目标速度 -25.0 之后，将通过“InVel_1”发出信号。复位“Jog_B”之后，将制动轴并逐步停止。

## 参见

MC\_MoveJog: 在点动模式下移动轴 V4 及更高版本 (页 402)

## 2.1.9 MC\_CommandTable

### 2.1.9.1 MC\_CommandTable: 按照运动顺序运行轴命令 V4 及更高版本

#### 说明

运动控制指令“MC\_CommandTable”可将多个单独的轴控制命令组合到一个运动顺序中。“MC\_CommandTable”适用于采用通过 PTO (Pulse Train Output) 的驱动器连接的轴。

#### 要求

- 定位轴工艺对象已插入且已正确组态。
- 驱动器通过 PTO (Pulse Train Output) 进行连接。
- 命令表工艺对象已插入并且组态正确。
- 轴已启用。

#### 超驰响应

可通过下列运动控制命令中止 MC\_CommandTable 命令：

- MC\_Home 命令 Mode = 3
- MC\_Halt 命令
- MC\_MoveAbsolute 命令
- MC\_MoveRelative 命令
- MC\_MoveVelocity 命令
- MC\_MoveJog 命令
- MC\_CommandTable 命令

新的 MC\_CommandTable 命令可中止下列激活的运动控制命令：

- MC\_Home 命令 Mode = 3
- MC\_Halt 命令
- MC\_MoveAbsolute 命令
- MC\_MoveRelative 命令
- MC\_MoveVelocity 命令
- MC\_MoveJog 命令
- MC\_CommandTable 命令

当启动第一个“Positioning Relative”、“Positioning Absolute”、“Velocity set point”或“Halt”命令时，将取消激活的运动控制命令。

### 参数

参数	声明	数据类型	默认值	说明
Axis	INPUT	TO_SpeedAxis	-	轴工艺对象
CommandTable	INPUT	TO_Command-Table	-	命令表工艺对象
Execute	INPUT	BOOL	FALSE	命令表在上升沿时启动
StartStep	INPUT	INT	1	定义命令表应开始执行的步限值： $1 \leq \text{StartStep} \leq \text{EndStep}$
EndStep	INPUT	INT	32	定义命令表应结束执行的步限值： $\text{StartStep} \leq \text{EndStep} \leq 32$
Done	OUTPUT	BOOL	FALSE	TRUE 已成功执行命令表
Busy	OUTPUT	BOOL	FALSE	TRUE 正在执行命令表
Command-Aborted	OUTPUT	BOOL	FALSE	TRUE 已通过另一个命令取消命令表。
Error	OUTPUT	BOOL	FALSE	TRUE 执行命令表期间出错。错误原因，请参见“ErrorID”和“ErrorInfo”的参数说明。
ErrorID	OUTPUT	WORD	16#0000	参数“Error”的错误 ID (页 205)
ErrorInfo	OUTPUT	WORD	16#0000	参数“ErrorID”的错误信息 ID (页 205)
CurrentStep	OUTPUT	INT	0	当前正在执行的命令表中的步
StepCode	OUTPUT	WORD	16#0000	当前正在执行的步的用户定义的数值/位模式

## 参见

- 运动控制语句概述 (页 162)
- ErrorID 和 ErrorInfo 列表 (自工艺对象 V4 起) (页 205)
- MC\_Power: 启用、禁用轴 V4 及更高版本 (页 368)
- MC\_Reset: 确认故障 V4 及更高版本 (页 374)
- MC\_Home: 使轴归位, 设置参考点 V4 及更高版本 (页 377)
- MC\_Halt: 停止轴 V4 及更高版本 (页 382)
- MC\_MoveAbsolute: 轴的绝对定位 V4 及更高版本 (页 386)
- MC\_MoveRelative: 轴的相对定位 V4 及更高版本 (页 391)
- MC\_MoveVelocity: 以预设的旋转速度移动轴 V4 及更高版本 (页 396)
- MC\_MoveJog: 在点动模式下移动轴 V4 及更高版本 (页 402)
- MC\_ChangeDynamic: 更改轴的动态设置 V4 及更高版本 (页 409)
- MC\_ReadParam: 连续读取定位轴的运动数据 V4 及更高版本 (页 413)
- MC\_WriteParam: 写入定位轴的变量 V4 及更高版本 (页 416)
- S7-1200 运动控制 V1...3 (页 419)



## 2.1.10 MC\_ChangeDynamic

### 2.1.10.1 MC\_ChangeDynamic: 更改轴的动态设置 V4 及更高版本

#### 说明

使用运动控制指令“MC\_ChangeDynamic”可以更改轴的下列设置:

- 更改加速时间（加速度）值
- 更改减速时间（减速度）值
- 更改急停减速时间（急停减速度）值
- 更改平滑时间（冲击）值

有关上述更改产生的影响，请参见变量 (页 235) 的说明。

#### 要求

定位轴工艺对象已正确组态。

## 超驰响应

任何其它运动控制命令均无法中止 MC\_ChangeDynamic 命令。

新的 MC\_ChangeDynamic 命令不会中止任何激活的运动控制命令。

## 参数

参数	声明	数据类型	默认值	说明
Axis	INPUT	TO_SpeedAxis	-	轴工艺对象
Execute	INPUT	BOOL	FALSE	上升沿时启动命令
Change RampUp	INPUT	BOOL	FALSE	TRUE 按照输入参数“RampUpTime”更改加速时间
RampUpTime	INPUT	REAL	5.00	不使用冲击限制时，将轴从停止状态加速到组态的最大速度所需的时间（以秒为单位）。 更改将影响变量 <轴名称>。 Config.DynamicDefaults.Acceleration。 有关上述更改产生的影响， 请参见变量的说明。
ChangeRamp Down	INPUT	BOOL	FALSE	TRUE 更改减速时间以与输入参数“RampDownTime”相对应
RampDown Time	INPUT	REAL	5.00	不使用冲击限制器时，将轴从组态的最大速度减速到停止状态所需的时间（以秒为单位）。 更改将影响变量 <轴名称>。 Config.DynamicDefaults.Deceleration。 有关上述更改产生的影响， 请参见变量的说明。
Change Emergency	INPUT	BOOL	FALSE	TRUE 按照输入参数“EmergencyRampTime”更改急停减速时间

参数	声明	数据类型	默认值	说明
Emergency RampTime	INPUT	REAL	2.00	在急停模式下不使用冲击限制器时，将轴从组态的最大速度减速到停止状态所需的时间（以秒为单位）。 更改将影响变量 <轴名称>。 Config.DynamicDefaults. EmergencyDeceleration。 有关上述更改产生的影响，请参见变量的说明。
ChangeJerk Time	INPUT	BOOL	FALSE	TRUE 按照输入参数“JerkTime”更改平滑时间
JerkTime	INPUT	REAL	0.25	用于轴加速斜坡和轴减速斜坡的平滑时间（以秒为单位） 更改将影响变量 <轴名称>。 Config.DynamicDefaults.Jerk。 有关上述更改产生的影响，请参见变量的说明。
Done	OUTPUT	BOOL	FALSE	TRUE 更改的值已写入工艺数据块。 变量的说明将显示更改生效的时间。
Error	OUTPUT	BOOL	FALSE	TRUE 执行命令期间出错。 错误原因，请参见“ErrorID”和“ErrorInfo”的参数说明。
ErrorID	OUTPUT	WORD	16#0000	参数“Error”的错误 ID (页 205)
ErrorInfo	OUTPUT	WORD	16#0000	参数“ErrorID”的错误信息 ID (页 205)

### 说明

在输入参数“RampUpTime”、“RampDownTime”、“EmergencyRampTime”和“JerkTime”中，输入值可能会导致以下生成的参数超出允许的限值范围：

“加速度”(Acceleration)、“减速度”(Deceleration)、  
“急停减速度”(Emergency stop deceleration) 和“冲击”(Jerk)。

请确保输入处于有效范围内，同时注意“动态 (页 82)”部分给出的等式和限值。

## 参见

运动控制语句概述 (页 162)

ErrorID 和 ErrorInfo 列表 (自工艺对象 V4 起) (页 205)

在用户程序中可以更改动态组态 (页 89)

在用户程序中更改回原点组态 (页 98)

自 V4 起的定位轴工艺对象的变量 (页 235)

MC\_Power: 启用、禁用轴 V4 及更高版本 (页 368)

MC\_Reset: 确认故障 V4 及更高版本 (页 374)

MC\_Home: 使轴归位, 设置参考点 V4 及更高版本 (页 377)

MC\_Halt: 停止轴 V4 及更高版本 (页 382)

MC\_MoveAbsolute: 轴的绝对定位 V4 及更高版本 (页 386)

MC\_MoveRelative: 轴的相对定位 V4 及更高版本 (页 391)

MC\_MoveVelocity: 以预设的旋转速度移动轴 V4 及更高版本 (页 396)

MC\_MoveJog: 在点动模式下移动轴 V4 及更高版本 (页 402)

MC\_CommandTable: 按照运动顺序运行轴命令 V4 及更高版本 (页 406)

MC\_ReadParam: 连续读取定位轴的运动数据 V4 及更高版本 (页 413)

MC\_WriteParam: 写入定位轴的变量 V4 及更高版本 (页 416)

S7-1200 运动控制 V1...3 (页 419)

## 2.1.11 MC\_ReadParam

### 2.1.11.1 MC\_ReadParam: 连续读取定位轴的运动数据 V4 及更高版本

#### 说明

“MC\_ReadParam”运动控制指令可连续读取轴的运动数据和状态消息。相应变量的当前值在命令的起始处决定。

可以读取以下运动数据和状态消息：

- 工艺版本 V4 及更高版本：
  - 轴的位置设定值
  - 轴的速度设定值和实际值
  - 轴距目标位置的当前距离
  - 轴的目标位置
- 工艺版本 V5 及更高版本的其它信息：
  - 轴的实际位置
  - 轴的实际速度
  - 当前的跟随误差
  - 驱动器状态
  - 编码器状态
  - 状态位
  - 错误位

#### 要求

定位轴工艺对象已正确组态。

#### 超驰响应

任何其它运动控制命令均无法中止 MC\_ReadParam 命令。

新的 MC\_ReadParam 命令不会中止任何激活的运动控制命令。

## 参数

参数	声明	数据类型	默认值	说明	
Enable	INPUT	BOOL	FALSE	TRUE	读取通过“Parameter”指定的变量并将值存储在通过“Value”指定的目标地址中。
				FALSE	不会更新已分配的运动数据
Parameter	INPUT	VARIANT (REAL)	-	指向要读取的值的 VARIANT 指针。 允许使用下列变量： <ul style="list-style-type: none"> <li>• &lt;轴名称&gt;.Position</li> <li>• &lt;轴名称&gt;.Velocity</li> <li>• &lt;轴名称&gt;.ActualPosition</li> <li>• &lt;轴名称&gt;.ActualVelocity</li> <li>• &lt;轴名称&gt;.StatusPositioning.&lt;变量名称&gt;</li> <li>• &lt;轴名称&gt;.StatusDrive.&lt;变量名称&gt;</li> <li>• &lt;轴名称&gt;.StatusSensor.&lt;变量名称&gt;</li> <li>• &lt;轴名称&gt;.StatusBits.&lt;变量名称&gt;</li> <li>• &lt;轴名称&gt;.ErrorBits.&lt;变量名称&gt;</li> </ul> 有关指定变量的说明和变量结构，请参见附录自 V4 起的定位轴工艺对象的变量 (页 235)。	
Value	INOUT	VARIANT (REAL)	-	指向写入所读取值的目标变量或目标地址的 VARIANT 指针。	
Valid	OUTPUT	BOOL	FALSE	TRUE	读取的值有效。
				FALSE	读取的值无效。
Busy	OUTPUT	BOOL	FALSE	TRUE	命令正在执行
Error	OUTPUT	BOOL	FALSE	TRUE	执行命令期间出错。 错误原因，请参见“ErrorID”和“ErrorInfo”的参数说明。
ErrorID	OUTPUT	WORD	16#0000	参数“Error”的错误 ID (页 205)	
ErrorInfo	OUTPUT	WORD	16#0000	参数“ErrorID”的错误信息 ID (页 205)	

## 参见

- 运动控制语句概述 (页 162)
- ErrorID 和 ErrorInfo 列表 (自工艺对象 V4 起) (页 205)
- S7-1200 运动控制 V1...3 (页 419)
- MC\_Power: 启用、禁用轴 V4 及更高版本 (页 368)
- MC\_Reset: 确认故障 V4 及更高版本 (页 374)
- MC\_Home: 使轴归位, 设置参考点 V4 及更高版本 (页 377)
- MC\_Halt: 停止轴 V4 及更高版本 (页 382)
- MC\_MoveAbsolute: 轴的绝对定位 V4 及更高版本 (页 386)
- MC\_MoveRelative: 轴的相对定位 V4 及更高版本 (页 391)
- MC\_MoveVelocity: 以预设的旋转速度移动轴 V4 及更高版本 (页 396)
- MC\_MoveJog: 在点动模式下移动轴 V4 及更高版本 (页 402)
- MC\_CommandTable: 按照运动顺序运行轴命令 V4 及更高版本 (页 406)
- MC\_ChangeDynamic: 更改轴的动态设置 V4 及更高版本 (页 409)
- MC\_WriteParam: 写入定位轴的变量 V4 及更高版本 (页 416)

## 2.1.12 MC\_WriteParam

### 2.1.12.1 MC\_WriteParam: 写入定位轴的变量 V4 及更高版本

#### 说明

运动控制指令“MC\_WriteParam”可在用户程序中写入定位轴工艺对象的变量。与用户程序中变量的赋值不同的是，“MC\_WriteParam”还可以更改只读变量的值。有关变量、写入变量的条件以及变量生效的时间，请参见工艺对象变量 (页 235)的说明。

使用通过 PROFIdrive/模拟量输出的驱动器连接，部分参数要求使用“MC\_WriteParam”写入后重新启动工艺对象。如果要求重新启动，则工艺对象 <轴名称>.StatusBits.RestartRequired 的变量将会给出相应的指示。在启用工艺对象 (MC\_Power.Status = TRUE) 的情况下重新启动后这些参数的参数值更改将会生效。

#### 要求

- 定位轴工艺对象已正确组态。
- 要在用户程序中写入只读变量，必须禁用轴。

#### 超驰响应

任何其它运动控制命令均无法中止 MC\_WriteParam 命令。  
新的 MC\_WriteParam 命令不会中止任何激活的运动控制命令。



## 参数

参数	声明	数据类型	默认值	说明
Parameter	INPUT	VARIANT (BOOL、INT、 DINT、REAL)	-	指向要写入的工艺对象变量 (页 235)定位轴 (目标地址) 的 VARIANT 指针
Value	INPUT	VARIANT (BOOL、INT、 DINT、REAL)	-	指向要写入的值 (源地址) 的 VARIANT 指针
Execute	INPUT	BOOL	FALSE	上升沿时启动命令
Done	OUTPUT	BOOL	FALSE	TRUE 值已写入
Busy	OUTPUT	BOOL	FALSE	TRUE 命令正在执行
Error	OUTPUT	BOOL	FALSE	TRUE 执行命令期间出错。 错误原因, 请参见“ErrorID” 和“ErrorInfo”的参数 说明。
ErrorID	OUTPUT	WORD	16#0000	参数“Error”的错误 ID (页 205)
ErrorInfo	OUTPUT	WORD	16#0000	参数“ErrorID”的错误信 息 ID (页 205)

参见

运动控制语句概述 (页 162)

ErrorID 和 ErrorInfo 列表 (自工艺对象 V4 起) (页 205)

自 V4 起的定位轴工艺对象的变量 (页 235)

S7-1200 运动控制 V1...3 (页 419)

MC\_Power: 启用、禁用轴 V4 及更高版本 (页 368)

MC\_Reset: 确认故障 V4 及更高版本 (页 374)

MC\_Home: 使轴归位, 设置参考点 V4 及更高版本 (页 377)

MC\_Halt: 停止轴 V4 及更高版本 (页 382)

MC\_MoveAbsolute: 轴的绝对定位 V4 及更高版本 (页 386)

MC\_MoveRelative: 轴的相对定位 V4 及更高版本 (页 391)

MC\_MoveVelocity: 以预设的旋转速度移动轴 V4 及更高版本 (页 396)

MC\_MoveJog: 在点动模式下移动轴 V4 及更高版本 (页 402)

MC\_CommandTable: 按照运动顺序运行轴命令 V4 及更高版本 (页 406)

MC\_ChangeDynamic: 更改轴的动态设置 V4 及更高版本 (页 409)

MC\_ReadParam: 连续读取定位轴的运动数据 V4 及更高版本 (页 413)

## 2.2 S7-1200 运动控制 V1...3

### 2.2.1 MC\_Power

#### 2.2.1.1 MC\_Power: 启用、禁用轴 V1...3

##### 说明

“MC\_Power”运动控制指令可启用或禁用轴。

##### 要求

- 已正确组态轴工艺对象。
- 没有待决的启用/禁止错误。

##### 超驰响应

运动控制命令无法中止“MC\_Power”的执行。

禁用轴（输入参数“Enable”= FALSE）之后，将根据所选“StopMode”中止相关工艺对象的所有运动控制命令。

## 参数

参数	声明	数据类型	默认值	说明	
Axis	INPUT	TO_Axis_1	-	轴工艺对象	
Enable	INPUT	BOOL	FALSE	TRUE	运动控制尝试启用轴。
				FALSE	根据组态的“StopMode”中断当前所有作业。停止并禁用轴。
StopMode	INPUT	INT	0	0	紧急停止 如果禁用轴的请求处于待决状态，则轴将以组态的急停减速度进行制动。 轴在变为静止状态后被禁用。
				1	立即停止 如果禁用轴的请求处于待决状态，则会在不减速的情况下禁用轴。 脉冲输出立即停止。
				2	带有加速度变化率控制的紧急停止 如果禁用轴的请求处于待决状态，则轴将以组态的急停减速度进行制动。 如果激活了加速度变化率控制，会将已组态的加速度变化率考虑在内。 轴在变为静止状态后被禁用。

参数	声明	数据类型	默认值	说明	
Status	OUTPUT	BOOL	FALSE	轴的使能状态	
				FALSE	禁用轴。 轴不会执行运动控制命令也不会接受任何新命令 (例外: MC_Reset 命令)。 轴未回原点。 在禁用轴时, 只有在轴停止之后, 才会将状态更改为 FALSE。
				TRUE	轴已启用。 轴已就绪, 可以执行运动控制命令。 在启用轴时, 直到信号“驱动器准备就绪”处于待决状态之后, 才会将状态更改为 TRUE。 在轴组态中, 如果未组态“驱动器准备就绪”驱动器接口, 那么状态将会立即更改为 TRUE。
Busy	OUTPUT	BOOL	FALSE	TRUE	“MC_Power”处于活动状态。
Error	OUTPUT	BOOL	FALSE	TRUE	运动控制指令“MC_Power”或相关工艺对象发生错误。 错误原因, 请参见“ErrorID”和“ErrorInfo”的参数说明。
ErrorID	OUTPUT	WORD	16#0000	参数“Error”的错误 ID (页 320)	
ErrorInfo	OUTPUT	WORD	16#0000	参数“ErrorID”的错误信息 ID (页 320)	

### 说明

如果因出错而停止轴, 那么在清除并确认错误之后, 会再次自动启用轴。  
这要求输入参数“Enable”的值在该过程中保持为 TRUE。

### 启用带有已组态驱动器接口的轴

要启用轴，请按下列步骤操作：

1. 首先检查是否满足上述要求。
2. 使用所需值对输入参数“StopMode”进行初始化。将输入参数“Enable”设置为 TRUE。

将“启用驱动器”的使能输出更改为 TRUE，以接通驱动器的电源。CPU 将等待驱动器的“驱动器就绪”信号。

当 CPU 组态完成且输入端出现“驱动器就绪”信号时，将启用轴。

输出参数“Status”和工艺对象变量 <轴名称>.StatusBits.Enable 的值为 TRUE。

### 启用不带已组态驱动器接口的轴

要启用轴，请按下列步骤操作：

1. 首先检查是否满足上述要求。
2. 使用所需值对输入参数“StopMode”进行初始化。将输入参数“Enable”设置为 TRUE。轴已启用。输出参数“Status”和工艺对象变量 <轴名称>.StatusBits.Enable 的值为 TRUE。

### 禁用轴

要禁用轴，可以按照下列步骤操作：

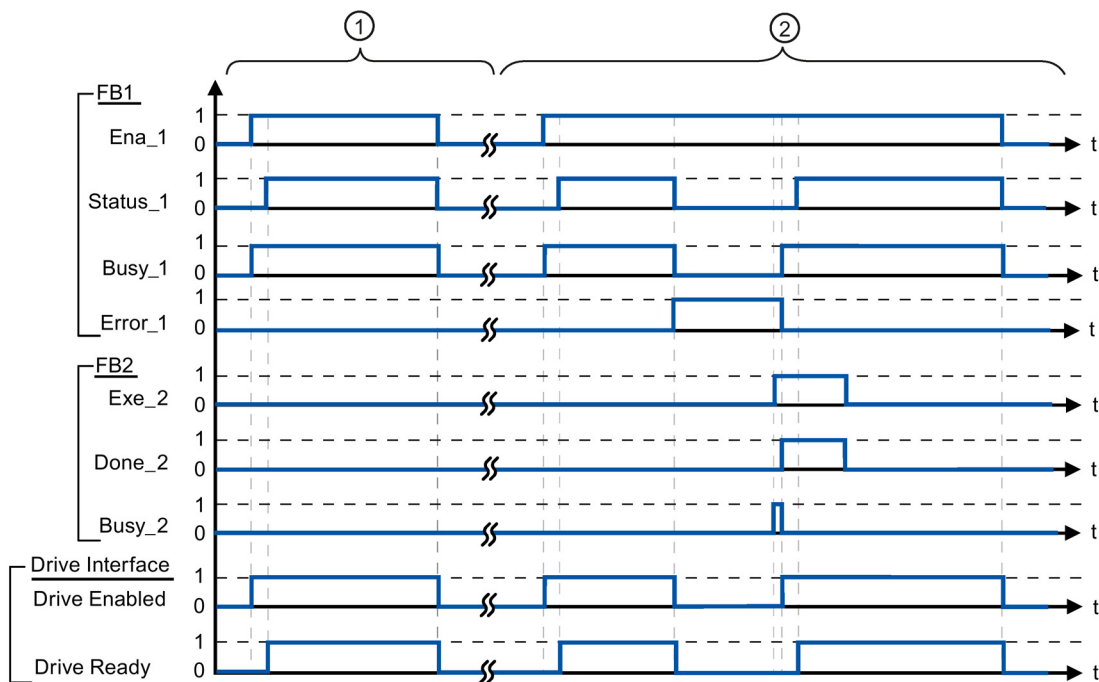
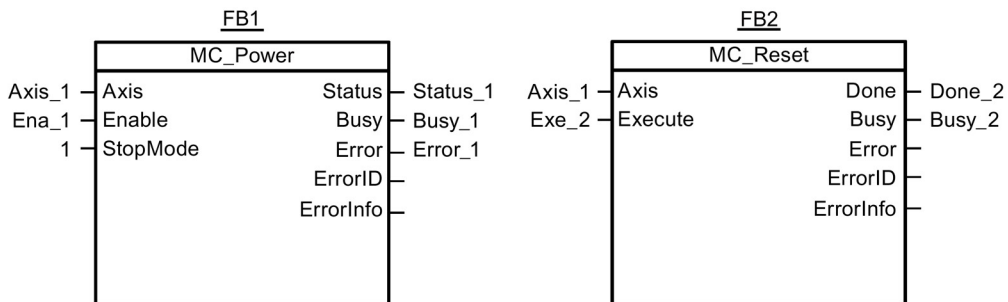
1. 停止轴。  
可以通过工艺对象变量 <轴名称>.StatusBits.StandStill 识别轴何时处于停止状态。
2. 在轴停止后，将输入参数“Enable”设置为 FALSE。
3. 如果输出参数“Busy”和“Status”以及工艺对象变量 <轴名称>.StatusBits.Enable 的值均为 FALSE，则说明禁用轴已完成。

## 参见

- 运动控制语句概述 (页 162)
- S7-1200 运动控制 V4 及更高版本 (页 368)
- ErrorID 和 ErrorInfo (页 320)
- MC\_Power: 功能图 V1...3 (页 424)
- MC\_Reset: 应答故障 V1...3 (页 425)
- MC\_Home: 使轴回原点, 设置参考点 V1...3 (页 428)
- MC\_Halt: 停止轴 V1...3 (页 433)
- MC\_MoveAbsolute: 轴的绝对定位 V1...3 (页 436)
- MC\_MoveRelative: 轴的相对定位 V1...3 (页 441)
- MC\_MoveVelocity: 以预设的旋转速度移动轴 V1...3 (页 446)
- MC\_MoveJog: 在点动模式下移动轴 V1...3 (页 451)
- MC\_CommandTable: 按照运动顺序运行轴命令 V2...3 (页 455)
- MC\_ChangeDynamic: 更改轴的动态设置 V2...3 (页 458)

2.2.1.2 MC\_Power: 功能图 V1...3

功能图



①	启用轴之后再次禁用。驱动器将“驱动器准备就绪”信号发送回 CPU 后，可以通过“Status_1”标识已成功启用轴。
②	在启用轴之后，如果发生错误，则会禁用轴。在消除错误并通过“MC_Reset”确认后，然后该轴再次被启用。

参见

MC\_Power: 启用、禁用轴 V1...3 (页 419)



## 2.2.2 MC\_Reset

### 2.2.2.1 MC\_Reset: 应答故障 V1...3

#### 说明

运动控制指令“MC\_Reset”可用于确认“伴随轴停止出现的运行错误”和“组态错误”。关于这些需要进行确认的错误，请参见“解决方法”下的“ErrorID 和 ErrorInfo 列表”。

从版本 V3.0 开始，可在 RUN 运行模式下将轴组态下载到工作存储器。

#### 要求

- 已正确组态轴工艺对象。
- 已经清除了引起这些需确认的待决组态错误的原因（例如，已将定位轴工艺对象中的加速度更改为有效值）。

#### 超驰响应

任何其它运动控制命令均无法中止 MC\_Reset 命令。

新的 MC\_Reset 命令不会中止任何其它激活的运动控制命令。

## 参数

参数	声明	数据类型	默认值	说明	
Axis	INPUT	TO_Axis_1	-	轴工艺对象	
Execute	INPUT	BOOL	FALSE	上升沿时启动命令	
Restart	INPUT	BOOL	FALSE	(从版本 V3.0 开始)	
				TRUE	将轴组态从装载存储器下载到工作存储器。 仅可在禁用轴后，才能执行该命令。 请参见有关下载到 CPU (页 153) 的说明。
				FALSE	确认待决的错误
Done	OUTPUT	BOOL	FALSE	TRUE 错误已确认。	
Busy	OUTPUT	BOOL	FALSE	TRUE 正在执行命令。	
Error	OUTPUT	BOOL	FALSE	TRUE 执行命令期间出错。 错误原因，请参见“ErrorID”和“ErrorInfo”的参数说明。	
ErrorID	OUTPUT	WORD	16#0000	参数“Error”的错误 ID (页 320)	
ErrorInfo	OUTPUT	WORD	16#0000	参数“ErrorID”的错误信息 ID (页 320)	

## 可通过 MC\_Reset 参数，对需要进行确认的错误进行确认

要确认错误，请按照下列步骤操作：

1. 首先检查是否满足上述要求。
2. 在输入参数“Execute”的上升沿开始确认错误。
3. 如果输出参数“Done”的值为 TRUE，同时工艺对象变量 <轴名称>.StatusBits.Error 的值为 FALSE，则说明错误已被确认。

## 参见

运动控制语句概述 (页 162)

S7-1200 运动控制 V4 及更高版本 (页 368)

ErrorID 和 ErrorInfo (页 320)

下载到 CPU (页 153)

MC\_Power: 启用、禁用轴 V1...3 (页 419)

MC\_Home: 使轴回原点，设置参考点 V1...3 (页 428)

MC\_Halt: 停止轴 V1...3 (页 433)

MC\_MoveAbsolute: 轴的绝对定位 V1...3 (页 436)

MC\_MoveRelative: 轴的相对定位 V1...3 (页 441)

MC\_MoveVelocity: 以预设的旋转速度移动轴 V1...3 (页 446)

MC\_MoveJog: 在点动模式下移动轴 V1...3 (页 451)

MC\_CommandTable: 按照运动顺序运行轴命令 V2...3 (页 455)

MC\_ChangeDynamic: 更改轴的动态设置 V2...3 (页 458)

## 2.2.3 MC\_Home

### 2.2.3.1 MC\_Home: 使轴回原点，设置参考点 V1...3

#### 说明

使用“MC\_Home”运动控制指令可将轴坐标与实际物理驱动器位置匹配。轴的绝对定位需要回原点。可执行以下类型的回原点：

- 主动回原点（Mode = 3）  
自动执行回原点步骤。
- 被动回原点（Mode = 2）  
被动回原点期间，运动控制指令“MC\_Home”不会执行任何回原点运动。用户必须通过其它运动控制指令，执行这一步骤中所需的行进运动。检测到回原点开关时，轴即回原点。
- 直接绝对回原点（Mode = 0）  
将当前的轴位置设置为参数“Position”的值。
- 直接相对回原点（Mode = 1）  
将当前轴位置的偏移值设置为参数“Position”的值。

#### 要求

- 已正确组态轴工艺对象。
- 轴已启用。
- 以 Mode = 0、1 或 2 启动时不会任何激活 MC\_CommandTable 命令。

## 超驰响应

超驰响应取决于所选的模式:

### Mode = 0, 1

任何其它运动控制命令均无法中止 MC\_Home 命令。

MC\_Home 命令不会中止任何激活的运动控制命令。

按照新的回原点位置（输入参数“Position”的值）进行回原点操作后，将继续执行与位置相关的运动命令。

### Mode = 2

可通过下列运动控制命令中止 MC\_Home 命令:

- MC\_Home 命令 Mode = 2、3

新的 MC\_Home 命令可中止下列激活的运动控制命令:

- MC\_Home 命令 Mode = 2

按照新的回原点位置（输入参数“Position”的值）进行回原点操作后，将继续执行与位置相关的运动命令。

### Mode = 3

可通过下列运动控制命令中止 MC\_Home 命令:

- MC\_Home 命令 Mode = 3
- MC\_Halt 命令
- MC\_MoveAbsolute 命令
- MC\_MoveRelative 命令
- MC\_MoveVelocity 命令
- MC\_MoveJog 命令
- MC\_CommandTable 命令

新的 MC\_Home 命令可中止下列激活的运动控制命令:

- MC\_Home 命令 Mode = 2、3
- MC\_Halt 命令
- MC\_MoveAbsolute 命令
- MC\_MoveRelative 命令
- MC\_MoveVelocity 命令
- MC\_MoveJog 命令
- MC\_CommandTable 命令

## 参数

参数	声明	数据类型	默认值	说明	
Axis	INPUT	TO_Axis_1	-	轴工艺对象	
Execute	INPUT	BOOL	FALSE	上升沿时启动命令	
Position	INPUT	REAL	0.0	<ul style="list-style-type: none"> <li>Mode = 0、2 和 3 完成回原点操作之后，轴的绝对位置</li> <li>Mode = 1 对当前轴位置的修正值</li> </ul> 限值： $-1.0e^{12} \leq \text{Position} \leq 1.0e^{12}$	
Mode	INPUT	INT	0	回原点模式	
				0	绝对式直接回原点 新的轴位置为参数“Position”位置的 位置的值。
				1	相对式直接回原点 新的轴位置等于当前轴位置 + 参数“Position”位置的值。
				2	被动回原点 将根据轴组态进行回原点。 回原点后，将新的轴位置设置为参 数“Position”的值。
				3	主动回原点 按照轴组态进行回原点操作。 回原点后，将新的轴位置设置为参 数“Position”的值。
Done	OUTPUT	BOOL	FALSE	TRUE	命令已完成
Busy	OUTPUT	BOOL	FALSE	TRUE	正在执行命令。
Command Aborted	OUTPUT	BOOL	FALSE	TRUE	命令在执行过程中被另一命 令中止。
Error	OUTPUT	BOOL	FALSE	TRUE	执行命令期间出错。 错误原因，请参见“ErrorID”和 “ErrorInfo”的参数说明。

参数	声明	数据类型	默认值	说明
ErrorID	OUTPUT	WORD	16#0000	参数“Error”的错误 ID (页 320)
ErrorInfo	OUTPUT	WORD	16#0000	参数“ErrorID”的错误信息 ID (页 320)

### 说明

在下列情况下轴回原点会丢失：

- 通过“MC\_Power”运动控制指令禁用轴
- 在自动模式和手动控制之间切换
- 在启动主动回原点时。在成功完成回原点操作之后，轴回原点将再次可用。
- 关闭之后 -> 启动 CPU
- CPU 重新启动后 (RUN-STOP -> STOP-RUN)

## 使轴回原点

要使轴回原点，请按下列步骤操作：

1. 首先检查是否满足上述要求。
2. 使用这些值初始化所需的输入参数，然后在输入参数“Execute”的上升沿，开始回原点。  
。
3. 如果输出参数“Done”和工艺对象变量 <轴名称>.StatusBits.HomingDone 的值为 TRUE，则说明回原点已完成。

## 参见

运动控制语句概述 (页 162)

S7-1200 运动控制 V4 及更高版本 (页 368)

ErrorID 和 ErrorInfo (页 320)

MC\_Power: 启用、禁用轴 V1...3 (页 419)

MC\_Reset: 应答故障 V1...3 (页 425)

MC\_Halt: 停止轴 V1...3 (页 433)

MC\_MoveAbsolute: 轴的绝对定位 V1...3 (页 436)

MC\_MoveRelative: 轴的相对定位 V1...3 (页 441)

MC\_MoveVelocity: 以预设的旋转速度移动轴 V1...3 (页 446)

MC\_MoveJog: 在点动模式下移动轴 V1...3 (页 451)

MC\_CommandTable: 按照运动顺序运行轴命令 V2...3 (页 455)

MC\_ChangeDynamic: 更改轴的动态设置 V2...3 (页 458)



## 2.2.4 MC\_Halt

### 2.2.4.1 MC\_Halt: 停止轴 V1...3

#### 说明

通过运动控制指令“MC\_Halt”，可停止所有运动并以组态的减速度停止轴。  
未定义停止位置。

#### 要求

- 已正确组态轴工艺对象。
- 轴已启用。

#### 超驰响应

可通过下列运动控制命令中止 MC\_Halt 命令：

- MC\_Home 命令 Mode = 3
- MC\_Halt 命令
- MC\_MoveAbsolute 命令
- MC\_MoveRelative 命令
- MC\_MoveVelocity 命令
- MC\_MoveJog 命令
- MC\_CommandTable 命令

新的 MC\_Halt 命令可中止下列激活的运动控制命令：

- MC\_Home 命令 Mode = 3
- MC\_Halt 命令
- MC\_MoveAbsolute 命令
- MC\_MoveRelative 命令
- MC\_MoveVelocity 命令
- MC\_MoveJog 命令
- MC\_CommandTable 命令

## 参数

参数	声明	数据类型	默认值	说明
Axis	INPUT	TO_Axis_1	-	轴工艺对象
Execute	INPUT	BOOL	FALSE	上升沿时启动命令
Done	OUTPUT	BOOL	FALSE	TRUE 速度达到零
Busy	OUTPUT	BOOL	FALSE	TRUE 正在执行命令。
Command Aborted	OUTPUT	BOOL	FALSE	TRUE 命令在执行过程中被另一命令中止。
Error	OUTPUT	BOOL	FALSE	TRUE 执行命令期间出错。 错误原因，请参见“ErrorID”和“ErrorInfo”的参数说明。
ErrorID	OUTPUT	WORD	16#0000	参数“Error”的错误 ID (页 320)
ErrorInfo	OUTPUT	WORD	16#0000	参数“ErrorID”的错误信息 ID (页 320)

## 参见

运动控制语句概述 (页 162)

S7-1200 运动控制 V4 及更高版本 (页 368)

ErrorID 和 ErrorInfo (页 320)

MC\_Halt: 功能图 V1...3 (页 435)

MC\_Power: 启用、禁用轴 V1...3 (页 419)

MC\_Reset: 应答故障 V1...3 (页 425)

MC\_Home: 使轴回原点，设置参考点 V1...3 (页 428)

MC\_MoveAbsolute: 轴的绝对定位 V1...3 (页 436)

MC\_MoveRelative: 轴的相对定位 V1...3 (页 441)

MC\_MoveVelocity: 以预设的旋转速度移动轴 V1...3 (页 446)

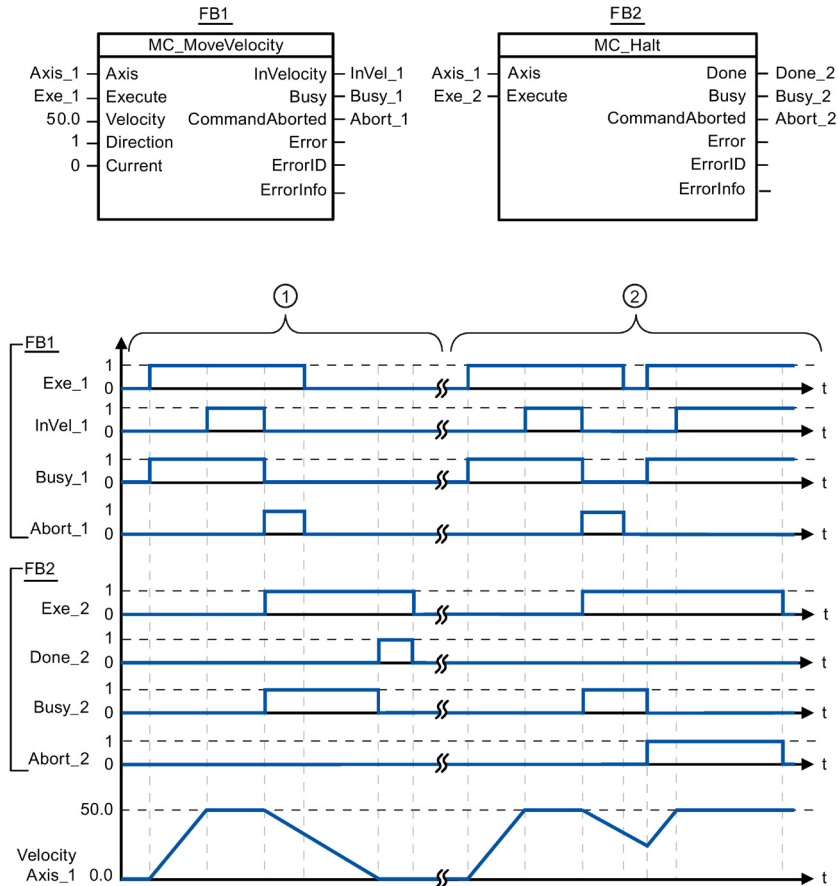
MC\_MoveJog: 在点动模式下移动轴 V1...3 (页 451)

MC\_CommandTable: 按照运动顺序运行轴命令 V2...3 (页 455)

MC\_ChangeDynamic: 更改轴的动态设置 V2...3 (页 458)

### 2.2.4.2 MC\_Halt: 功能图 V1...3

#### 功能图



在组态窗口“动态 > 常规”(Dynamics > General) 中，对下列值进行组态：

- 加速度： 10.0
- 减速度： 5.0

①	轴将由 MC_Halt 命令制动，直至停止为止。通过“Done_2”发出轴停止的信号。
②	当 MC_Halt 命令对轴进行制动处理时，另一个运动命令会中止该命令。将通过“Abort_2”发出中止信号。

#### 参见

MC\_Halt: 停止轴 V1...3 (页 433)

## 2.2.5 MC\_MoveAbsolute

### 2.2.5.1 MC\_MoveAbsolute: 轴的绝对定位 V1...3

#### 说明

运动控制指令“MC\_MoveAbsolute”启动轴定位运动，以将轴移动到某个绝对位置。

#### 要求

- 已正确组态轴工艺对象。
- 轴已启用。
- 轴已回原点。

#### 超驰响应

可通过下列运动控制命令中止 MC\_MoveAbsolute 命令：

- MC\_Home 命令 Mode = 3
- MC\_Halt 命令
- MC\_MoveAbsolute 命令
- MC\_MoveRelative 命令
- MC\_MoveVelocity 命令
- MC\_MoveJog 命令
- MC\_CommandTable 命令

新的 MC\_MoveAbsolute 命令可中止下列激活的运动控制命令：

- MC\_Home 命令 Mode = 3
- MC\_Halt 命令
- MC\_MoveAbsolute 命令
- MC\_MoveRelative 命令
- MC\_MoveVelocity 命令
- MC\_MoveJog 命令
- MC\_CommandTable 命令

## 参数

参数	声明	数据类型	默认值	说明
Axis	INPUT	TO_Axis_1	-	轴工艺对象
Execute	INPUT	BOOL	FALSE	上升沿时启动命令
Position	INPUT	REAL	0.0	绝对目标位置 限值： $-1.0e^{12} \leq \text{Position} \leq 1.0e^{12}$
Velocity	INPUT	REAL	10.0	轴的速度 由于所组态的加速度和减速度以及待接近的目标位置等原因，不会始终保持这一速度。 限值： 启动/停止速度 $\leq \text{Velocity} \leq$ 最大速度
Done	OUTPUT	BOOL	FALSE	TRUE 达到绝对目标位置
Busy	OUTPUT	BOOL	FALSE	TRUE 正在执行命令。
Command Aborted	OUTPUT	BOOL	FALSE	TRUE 命令在执行过程中被另一命令中止。
Error	OUTPUT	BOOL	FALSE	TRUE 执行命令期间出错。 错误原因，请参见“ErrorID”和“ErrorInfo”的参数说明。
ErrorID	OUTPUT	WORD	16#0000	参数“Error”的错误 ID (页 320)
ErrorInfo	OUTPUT	WORD	16#0000	参数“ErrorID”的错误信息 ID (页 320)

参见

运动控制语句概述 (页 162)

S7-1200 运动控制 V4 及更高版本 (页 368)

ErrorID 和 ErrorInfo (页 320)

MC\_MoveAbsolute: 功能图 V1...3 (页 439)

MC\_Power: 启用、禁用轴 V1...3 (页 419)

MC\_Reset: 应答故障 V1...3 (页 425)

MC\_Home: 使轴回原点, 设置参考点 V1...3 (页 428)

MC\_Halt: 停止轴 V1...3 (页 433)

MC\_MoveRelative: 轴的相对定位 V1...3 (页 441)

MC\_MoveVelocity: 以预设的旋转速度移动轴 V1...3 (页 446)

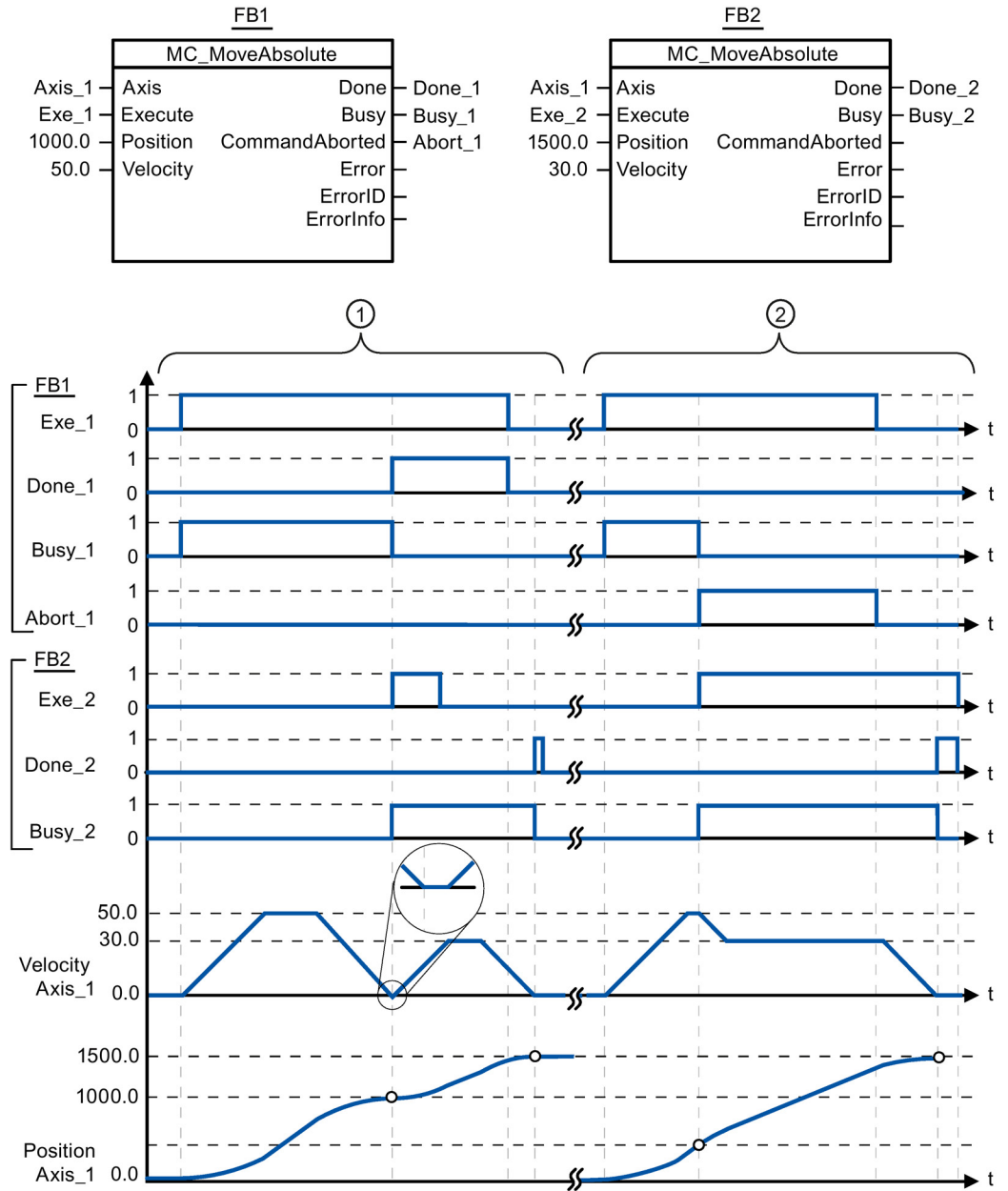
MC\_MoveJog: 在点动模式下移动轴 V1...3 (页 451)

MC\_CommandTable: 按照运动顺序运行轴命令 V2...3 (页 455)

MC\_ChangeDynamic: 更改轴的动态设置 V2...3 (页 458)

### 2.2.5.2 MC\_MoveAbsolute: 功能图 V1...3

功能图



在组态窗口“动态 > 常规”(Dynamics > General) 中，对下列值进行组态：

- 加速度： 10.0
- 减速度： 10.0

①	<p>通过 MC_MoveAbsolute 命令，可将轴移动到绝对位置 1000.0。</p> <p>当轴达到目标位置后，将通过“Done_1”发出信号。当“Done_1” = TRUE 时，将启动另一个目标位置为 1500.0 的 MC_MoveAbsolute 命令。</p> <p>由于响应时间的原因（如用户程序的循环时间等），轴将暂时停止（参见放大后的详细信息）。当轴达到新的目标位置后，将通过“Done_2”发出信号。</p>
②	<p>可通过另一个 MC_MoveAbsolute 命令中止激活的 MC_MoveAbsolute 命令。</p> <p>将通过“Abort_1”发出中止信号。之后，轴将会以新的速度移动到新的目标位置 1500.0。</p> <p>到达新的目标位置之后，将通过“Done_2”发出信号。</p>

### 参见

MC\_MoveAbsolute: 轴的绝对定位 V1...3 (页 436)



## 2.2.6 MC\_MoveRelative

### 2.2.6.1 MC\_MoveRelative: 轴的相对定位 V1...3

#### 说明

通过运动控制指令“MC\_MoveRelative”，启动相对于起始位置的定位运动。

#### 要求

- 已正确组态轴工艺对象。
- 轴已启用。

#### 超驰响应

可通过下列运动控制命令中止 MC\_MoveRelative 命令：

- MC\_Home 命令 Mode = 3
- MC\_Halt 命令
- MC\_MoveAbsolute 命令
- MC\_MoveRelative 命令
- MC\_MoveVelocity 命令
- MC\_MoveJog 命令
- MC\_CommandTable 命令

新的 MC\_MoveRelative 命令可中止下列激活的运动控制命令：

- MC\_Home 命令 Mode = 3
- MC\_Halt 命令
- MC\_MoveAbsolute 命令
- MC\_MoveRelative 命令
- MC\_MoveVelocity 命令
- MC\_MoveJog 命令
- MC\_CommandTable 命令

## 参数

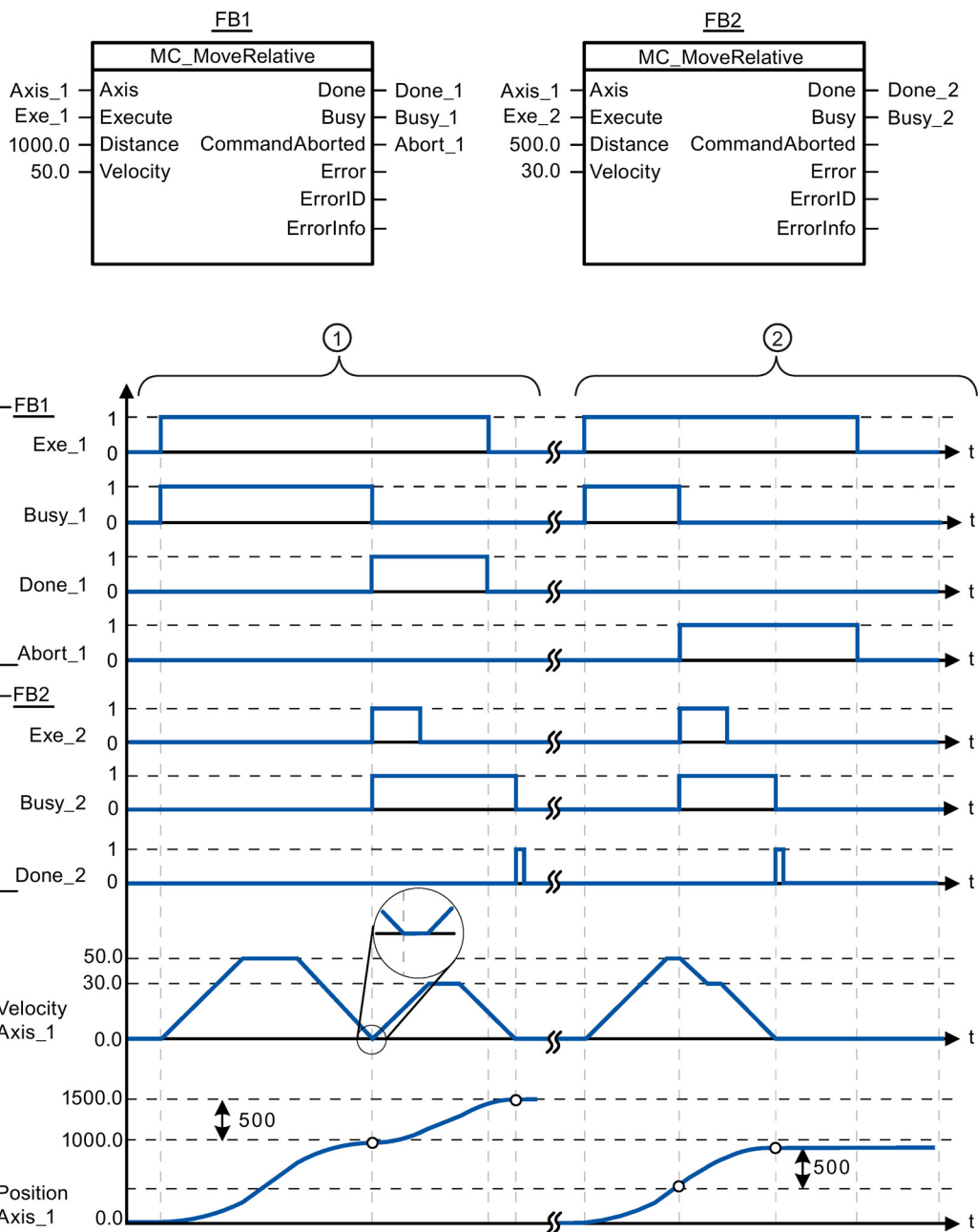
参数	声明	数据类型	默认值	说明
Axis	INPUT	TO_Axis_1	-	轴工艺对象
Execute	INPUT	BOOL	FALSE	上升沿时启动命令
Distance	INPUT	REAL	0.0	定位操作的移动距离 限值： $-1.0e^{12} \leq \text{Distance} \leq 1.0e^{12}$
Velocity	INPUT	REAL	10.0	轴的速度 由于所组态的加速度和减速度以及要途经的距离等原因，不会始终保持这一速度。 限值： 启动/停止速度 $\leq$ Velocity $\leq$ 最大速度
Done	OUTPUT	BOOL	FALSE	TRUE 目标位置已到达
Busy	OUTPUT	BOOL	FALSE	TRUE 正在执行命令。
Command Aborted	OUTPUT	BOOL	FALSE	TRUE 命令在执行过程中被另一命令中止。
Error	OUTPUT	BOOL	FALSE	TRUE 执行命令期间出错。 错误原因，请参见“ErrorID”和“ErrorInfo”的参数说明。
ErrorID	OUTPUT	WORD	16#0000	参数“Error”的错误 ID (页 320)
ErrorInfo	OUTPUT	WORD	16#0000	参数“ErrorID”的错误信息 ID (页 320)

## 参见

- 运动控制语句概述 (页 162)
- S7-1200 运动控制 V4 及更高版本 (页 368)
- ErrorID 和 ErrorInfo (页 320)
- MC\_MoveRelative: 功能图 V1...3 (页 444)
- MC\_Power: 启用、禁用轴 V1...3 (页 419)
- MC\_Reset: 应答故障 V1...3 (页 425)
- MC\_Home: 使轴回原点, 设置参考点 V1...3 (页 428)
- MC\_Halt: 停止轴 V1...3 (页 433)
- MC\_MoveAbsolute: 轴的绝对定位 V1...3 (页 436)
- MC\_MoveVelocity: 以预设的旋转速度移动轴 V1...3 (页 446)
- MC\_MoveJog: 在点动模式下移动轴 V1...3 (页 451)
- MC\_CommandTable: 按照运动顺序运行轴命令 V2...3 (页 455)
- MC\_ChangeDynamic: 更改轴的动态设置 V2...3 (页 458)

2.2.6.2 MC\_MoveRelative: 功能图 V1...3

功能图



在组态窗口“动态 > 常规”(Dynamics > General) 中，对下列值进行组态：

- 加速度： 10.0
- 减速度： 10.0

	<p>轴将通过 MC_MoveRelative 命令移动 1000.0 的距离 (“Distance”)。</p> <p>当轴达到目标位置后，将通过“Done_1”发出信号。“Done_1”= TRUE 时，将启动另一个行进距离为 500.0 的 MC_MoveRelative 命令。</p> <p>由于响应时间的原因（如用户程序的循环时间等），轴将暂时停止（参见放大后的详细信息）。当轴达到新的目标位置后，将通过“Done_2”发出信号。</p>
②	<p>可通过另一个 MC_MoveRelative 命令中止激活的 MC_MoveRelative 命令。</p> <p>将通过“Abort_1”发出中止信号。之后，轴将以新的速度移动 500.0 的距离 (“Distance”)。</p> <p>到达新的目标位置之后，将通过“Done_2”发出信号。</p>

## 参见

MC\_MoveRelative: 轴的相对定位 V1...3 (页 441)

## 2.2.7 MC\_MoveVelocity

### 2.2.7.1 MC\_MoveVelocity: 以预设的旋转速度移动轴 V1...3

#### 说明

通过运动控制指令“MC\_MoveVelocity”，根据指定的速度连续移动轴。

#### 要求

- 已正确组态轴工艺对象。
- 轴已启用。

#### 超驰响应

MC\_MoveVelocity 可由下列运动控制命令中止：

- MC\_Home 命令 Mode = 3
- MC\_Halt 命令
- MC\_MoveAbsolute 命令
- MC\_MoveRelative 命令
- MC\_MoveVelocity 命令
- MC\_MoveJog 命令
- MC\_CommandTable 命令

新的 MC\_MoveVelocity 命令可中止下列激活的运动控制命令：

- MC\_Home 命令 Mode = 3
- MC\_Halt 命令
- MC\_MoveAbsolute 命令
- MC\_MoveRelative 命令
- MC\_MoveVelocity 命令
- MC\_MoveJog 命令
- MC\_CommandTable 命令

## 参数

参数	声明	数据类型	默认值	说明	
Axis	INPUT	TO_Axis_1	-	轴工艺对象	
Execute	INPUT	BOOL	FALSE	上升沿时启动命令	
Velocity	INPUT	REAL	10.0	轴运动的指定速度 限值： 启动/停止速度 $\leq$  Velocity  $\leq$ 最大速度 (允许 Velocity = 0.0)	
Direction	INPUT	INT	0	指定方向	
				0	旋转方向取决于参数“Velocity”值的符号
				1	正旋转方向 (将忽略参数“Velocity”值的符号)
				2	负旋转方向 (将忽略参数“Velocity”值的符号)
Current	INPUT	BOOL	FALSE	保持当前速度	
				FALSE	“保持当前速度”已禁用。 将使用参数“Velocity”和“Direction”的值。
				TRUE	“保持当前速度”已启用。 而不考虑参数“Velocity”和“Direction”的值。 当轴继续以当前速度运动时，参数 "InVelocity 返回值 TRUE。
InVelocity	OUTPUT	BOOL	FALSE	TRUE <ul style="list-style-type: none"> <li>"Current" = FALSE: 达到参数“Velocity”中指定的速度。</li> <li>"Current" = TRUE: 轴在启动时，以当前速度进行移动。</li> </ul>	
Busy	OUTPUT	BOOL	FALSE	TRUE 正在执行命令。	
Command Aborted	OUTPUT	BOOL	FALSE	TRUE 命令在执行过程中被另一命令中止。	
Error	OUTPUT	BOOL	FALSE	TRUE 执行命令期间出错。 错误原因，请参见“ErrorID”和“ErrorInfo”的参数说明。	

参数	声明	数据类型	默认值	说明
ErrorID	OUTPUT	WORD	16#0000	参数“Error”的错误 ID (页 320)
ErrorInfo	OUTPUT	WORD	16#0000	参数“ErrorID”的错误信息 ID (页 320)

### 速度设定值为零 (Velocity = 0.0) 时的行为

“Velocity”= 0.0 的 MC\_MoveVelocity 命令（如 MC\_Halt 命令）可中止激活的运动命令，并使用所组态的减速度停止轴运动。

当轴停止时，至少在一个程序循环内将输出参数“InVelocity”的值指定为 TRUE。

“Busy”的值在减速过程中为 TRUE，并且随“InVelocity”一起变为 FALSE。

如果将参数“Execute”设置为 TRUE，则“InVelocity”和“Busy”处于锁定状态。

启动“MC\_MoveVelocity”命令时，将置位工艺对象的状态位“SpeedCommand”。

在轴停止时，设置状态位“ConstantVelocity”。

在启动新的运动命令后，将对这两个位进行调整，以适应新的情况。

### 参见

运动控制语句概述 (页 162)

S7-1200 运动控制 V4 及更高版本 (页 368)

ErrorID 和 ErrorInfo (页 320)

MC\_MoveVelocity: 功能图 V1...3 (页 449)

MC\_Power: 启用、禁用轴 V1...3 (页 419)

MC\_Reset: 应答故障 V1...3 (页 425)

MC\_Home: 使轴回原点，设置参考点 V1...3 (页 428)

MC\_Halt: 停止轴 V1...3 (页 433)

MC\_MoveAbsolute: 轴的绝对定位 V1...3 (页 436)

MC\_MoveRelative: 轴的相对定位 V1...3 (页 441)

MC\_MoveJog: 在点动模式下移动轴 V1...3 (页 451)

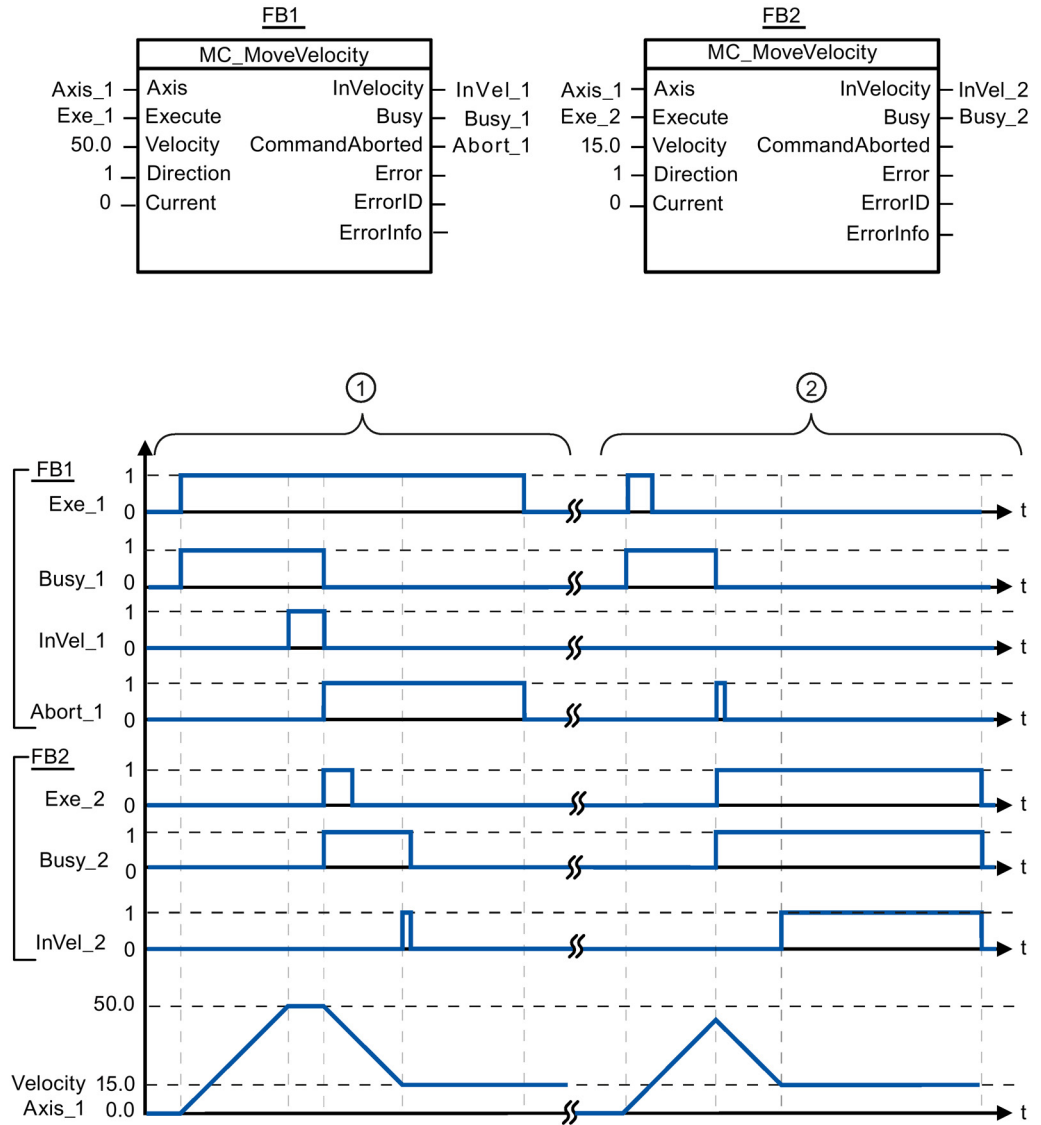
MC\_CommandTable: 按照运动顺序运行轴命令 V2...3 (页 455)

MC\_ChangeDynamic: 更改轴的动态设置 V2...3 (页 458)



### 2.2.7.2 MC\_MoveVelocity: 功能图 V1...3

功能图



在组态窗口“动态 > 常规”(Dynamics > General) 中，对下列值进行组态：

- 加速度：10.0
- 减速度：10.0

①	激活的 MC_MoveVelocity 命令通过“InVel_1”发出信号，说明已达到目标速度。 该命令随后将由另一个 MC_MoveVelocity 命令中止。将通过“Abort_1”发出中止信号。 达到新的目标速度 15.0 之后，将通过“InVel_2”发出信号。之后，轴将以新的恒定速度继续移动。
②	在达到目标速度之前，当前 MC_MoveVelocity 命令将由另一个 MC_MoveVelocity 命令中止。 将通过“Abort_1”发出中止信号。达到新的目标速度 15.0 之后，将通过“InVel_2”发出信号。 之后，轴将以新的恒定速度继续移动。

参见

MC\_MoveVelocity: 以预设的旋转速度移动轴 V1...3 (页 446)

## 2.2.8 MC\_MoveJog

### 2.2.8.1 MC\_MoveJog: 在点动模式下移动轴 V1...3

#### 说明

通过运动控制指令“MC\_MoveJog”，在点动模式下以指定的速度连续移动轴。  
例如，可以使用该运动控制指令进行测试和调试。

#### 要求

- 已正确组态轴工艺对象。
- 轴已启用。

#### 超驰响应

可通过下列运动控制命令中止 MC\_MoveJog 命令：

- MC\_Home 命令 Mode = 3
- MC\_Halt 命令
- MC\_MoveAbsolute 命令
- MC\_MoveRelative 命令
- MC\_MoveVelocity 命令
- MC\_MoveJog 命令
- MC\_CommandTable 命令

新的 MC\_MoveJog 命令可中止下列激活的运动控制命令：

- MC\_Home 命令 Mode = 3
- MC\_Halt 命令
- MC\_MoveAbsolute 命令
- MC\_MoveRelative 命令
- MC\_MoveVelocity 命令
- MC\_MoveJog 命令
- MC\_CommandTable 命令

## 参数

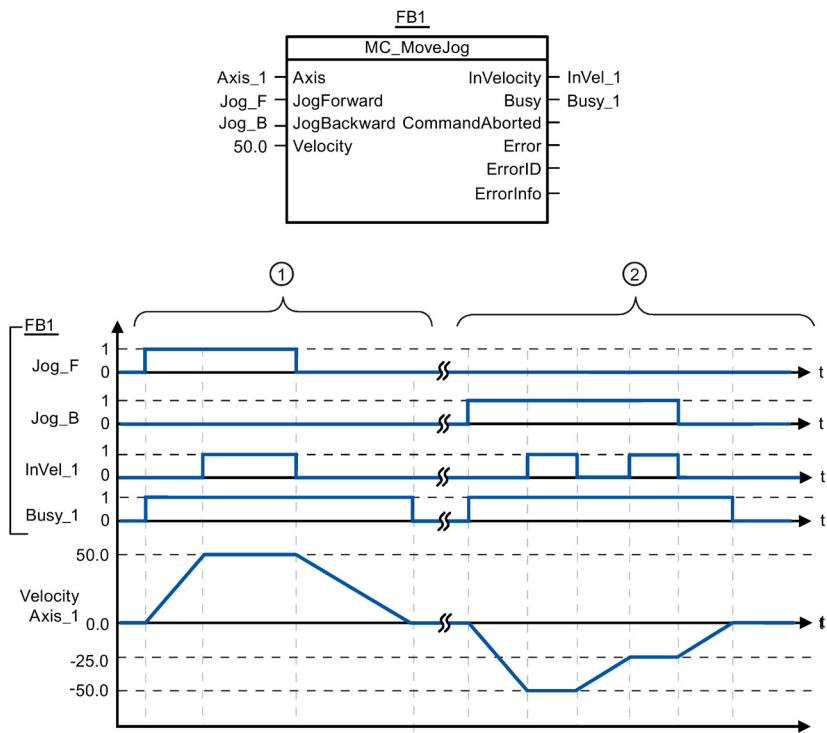
参数	声明	数据类型	默认值	说明
Axis	INPUT	TO_Axis_1	-	轴工艺对象
JogForward	INPUT	BOOL	FALSE	如果参数值为 TRUE，则轴都将按参数“Velocity”中所指定的速度，正向移动。
JogBackward	INPUT	BOOL	FALSE	如果参数值为 TRUE，则轴都将按参数“Velocity”中指定的速度，反向移动。
如果两个参数同时为 TRUE，轴将根据所组态的减速度直至停止。 通过参数“Error”、“ErrorID”和“ErrorInfo”，指出了错误。				
Velocity	INPUT	REAL	10.0	点动模式的预设速度
				限值，指令版本 V1.0: 启动/停止速度 ≤  Velocity  ≤ 最大速度
				限值，指令版本 V2.0: 启动/停止速度 ≤ 速度 ≤ 最大速度
InVelocity	OUTPUT	BOOL	FALSE	TRUE 达到参数“Velocity”中指定的速度。
Busy	OUTPUT	BOOL	FALSE	TRUE 正在执行命令。
CommandAborted	OUTPUT	BOOL	FALSE	TRUE 命令在执行过程中被另一命令中止。
Error	OUTPUT	BOOL	FALSE	TRUE 执行命令期间出错。 错误原因，请参见“ErrorID”和“ErrorInfo”的参数说明。
ErrorID	OUTPUT	WORD	16#0000	参数“Error”的错误 ID (页 320)
ErrorInfo	OUTPUT	WORD	16#0000	参数“ErrorID”的错误信息 ID (页 320)

## 参见

- 运动控制语句概述 (页 162)
- S7-1200 运动控制 V4 及更高版本 (页 368)
- ErrorID 和 ErrorInfo (页 320)
- MC\_MoveJog: 功能图 V1...3 (页 454)
- MC\_Power: 启用、禁用轴 V1...3 (页 419)
- MC\_Reset: 应答故障 V1...3 (页 425)
- MC\_Home: 使轴回原点, 设置参考点 V1...3 (页 428)
- MC\_Halt: 停止轴 V1...3 (页 433)
- MC\_MoveAbsolute: 轴的绝对定位 V1...3 (页 436)
- MC\_MoveRelative: 轴的相对定位 V1...3 (页 441)
- MC\_MoveVelocity: 以预设的旋转速度移动轴 V1...3 (页 446)
- MC\_CommandTable: 按照运动顺序运行轴命令 V2...3 (页 455)
- MC\_ChangeDynamic: 更改轴的动态设置 V2...3 (页 458)

2.2.8.2 MC\_MoveJog: 功能图 V1...3

功能图



在组态窗口“动态 > 常规”(Dynamics > General) 中，对下列值进行组态：

- 加速度： 10.0
- 减速度： 5.0

①	在点动模式下，轴通过“Jog_F”正向移动。达到目标速度 50.0 之后，将通过“InVelo_1”发出信号。复位“Jog_F”之后，轴将再次制动直至停止。
②	在点动模式下，轴通过“Jog_B”反向移动。达到目标速度 50.0 之后，将通过“InVelo_1”发出信号。复位“Jog_B”之后，轴将再次制动直至停止。

参见

MC\_MoveJog: 在点动模式下移动轴 V1...3 (页 451)

## 2.2.9 MC\_CommandTable

### 2.2.9.1 MC\_CommandTable: 按照运动顺序运行轴命令 V2...3

#### 说明

运动控制指令“MC\_CommandTable”可将多个单独的轴控制命令组合到一个运动顺序中。

#### 要求

- 轴工艺对象已插入 V2 且已正确组态。
- 命令表工艺对象已插入并且组态正确。
- 轴已启用。

#### 超驰响应

可通过下列运动控制命令中止 MC\_CommandTable 命令：

- MC\_Home 命令 Mode = 3
- MC\_Halt 命令
- MC\_MoveAbsolute 命令
- MC\_MoveRelative 命令
- MC\_MoveVelocity 命令
- MC\_MoveJog 命令
- MC\_CommandTable 命令

新的 MC\_CommandTable 命令可中止下列激活的运动控制命令：

- MC\_Home 命令 Mode = 3
- MC\_Halt 命令
- MC\_MoveAbsolute 命令
- MC\_MoveRelative 命令
- MC\_MoveVelocity 命令
- MC\_MoveJog 命令
- MC\_CommandTable 命令

当启动第一个“Positioning Relative”、“Positioning Absolute”、“Velocity set point”或“Halt”命令时，将取消激活的运动控制命令。

## 参数

参数	声明	数据类型	默认值	说明
Axis	INPUT	TO_Axis_1	-	轴工艺对象
Command Table	INPUT	TO_Command Table_1	-	命令表工艺对象
Execute	INPUT	BOOL	FALSE	命令表在上升沿时启动
StartStep	INPUT	INT	1	定义命令表应开始执行的步 限值： $1 \leq \text{StartStep} \leq \text{EndStep}$
EndStep	INPUT	INT	32	定义命令表应结束执行的步 限值： $\text{StartStep} \leq \text{EndStep} \leq 32$
Done	OUTPUT	BOOL	FALSE	TRUE 已成功执行命令表
Busy	OUTPUT	BOOL	FALSE	TRUE 正在执行命令表。
Command Aborted	OUTPUT	BOOL	FALSE	TRUE 已通过另一个命令取消命令表。
Error	OUTPUT	BOOL	FALSE	TRUE 执行命令表期间出错。 错误原因，请参见“ErrorID”和“ErrorInfo” 的参数说明。
ErrorID	OUTPUT	WORD	16#0000	参数“Error”的错误 ID (页 320)
ErrorInfo	OUTPUT	WORD	16#0000	参数“ErrorID”的错误信息 ID (页 320)
CurrentStep	OUTPUT	INT	0	当前正在执行的命令表中的步
StepCode	OUTPUT	WORD	16#0000	当前正在执行的步的用户定义的数值/位模式



## 参见

运动控制语句概述 (页 162)

S7-1200 运动控制 V4 及更高版本 (页 368)

ErrorID 和 ErrorInfo (页 320)

MC\_Power: 启用、禁用轴 V1...3 (页 419)

MC\_Reset: 应答故障 V1...3 (页 425)

MC\_Home: 使轴回原点, 设置参考点 V1...3 (页 428)

MC\_Halt: 停止轴 V1...3 (页 433)

MC\_MoveAbsolute: 轴的绝对定位 V1...3 (页 436)

MC\_MoveRelative: 轴的相对定位 V1...3 (页 441)

MC\_MoveVelocity: 以预设的旋转速度移动轴 V1...3 (页 446)

MC\_MoveJog: 在点动模式下移动轴 V1...3 (页 451)

MC\_ChangeDynamic: 更改轴的动态设置 V2...3 (页 458)

## 2.2.10 MC\_ChangeDynamic

### 2.2.10.1 MC\_ChangeDynamic: 更改轴的动态设置 V2...3

#### 说明

使用运动控制指令“MC\_ChangeDynamic”可以更改轴的下列设置:

- 更改加速时间（加速度）值
- 更改减速时间（减速度）值
- 更改急停减速时间（急停减速度）值
- 更改平滑时间（冲击）值

有关上述更改产生的影响，请参见变量 (页 342) 的说明。

#### 要求

- 轴工艺对象已插入版本 V2。
- 已正确组态轴工艺对象。

#### 超驰响应

任何其它运动控制命令均无法中止 MC\_ChangeDynamic 命令。

新的 MC\_ChangeDynamic 命令不会中止任何激活的运动控制命令。

## 参数

参数	声明	数据类型	默认值	说明
Axis	INPUT	TO_Axis_1	-	轴工艺对象
Execute	INPUT	BOOL	FALSE	上升沿时启动命令
ChangeRamp Up	INPUT	BOOL	FALSE	TRUE 按照输入参数“RampUpTime”更改加速时间
RampUpTime	INPUT	REAL	5.00	不使用冲击限制时，将轴从停止状态加速到组态的最大速度所需的时间（以秒为单位）。 更改将影响变量 <轴名称>.Config.DynamicDefaults.Acceleration。 有关上述更改产生的影响，请参见变量的说明。
ChangeRamp Down	INPUT	BOOL	FALSE	TRUE 按照输入参数“RampDownTime”更改减速时间
RampDown Time	INPUT	REAL	5.00	不使用冲击限制器时，将轴从组态的最大速度减速到停止状态所需的时间（以秒为单位）。 更改将影响变量 <轴名称>.Config.DynamicDefaults.Deceleration。 有关上述更改产生的影响，请参见变量的说明。
Change Emergency	INPUT	BOOL	FALSE	TRUE 按照输入参数“EmergencyRampTime”更改急停减速时间
Emergency RampTime	INPUT	REAL	2.00	在急停模式下不使用冲击限制器时，将轴从组态的最大速度减速到停止状态所需的时间（以秒为单位）。 更改将影响变量 <轴名称>.Config.DynamicDefaults.EmergencyDeceleration。 有关上述更改产生的影响，请参见变量的说明。
ChangeJerk Time	INPUT	BOOL	FALSE	TRUE 按照输入参数“JerkTime”更改平滑时间

参数	声明	数据类型	默认值	说明
JerkTime	INPUT	REAL	0.25	用于轴加速斜坡和轴减速斜坡的平滑时间 (以秒为单位) 更改将影响变量 <轴名称>.Config.DynamicDefaults.Jerk。 有关上述更改产生的影响， 请参见变量的说明。
Done	OUTPUT	BOOL	FALSE	TRUE 更改的值已写入工艺数据块。 变量的说明将显示更改生效的时间。
Error	OUTPUT	BOOL	FALSE	TRUE 执行命令期间出错。 错误原因，请参见“ErrorID”和“ErrorInfo” 的参数说明。
ErrorID	OUTPUT	WORD	16#0000	参数“Error”的错误 ID (页 320)
ErrorInfo	OUTPUT	WORD	16#0000	参数“ErrorID”的错误信息 ID (页 320)

### 说明

在输入参数“RampUpTime”、“RampDownTime”、“EmergencyRampTime”和“JerkTime”中，输入值可能会导致以下生成的参数超出允许的限值范围：

“加速度”(Acceleration)、“减速度”(Deceleration)、“急停减速度”(Emergency stop deceleration) 和“冲击”(Jerk)。

请注意“轴工艺对象 -> 组态工艺对象 -> 动态”(Axis technology object -> Configuring the technology object -> Dynamics) 中的等式和限值，确保输入的值处于有效范围内。

## 参见

运动控制语句概述 (页 162)

S7-1200 运动控制 V4 及更高版本 (页 368)

ErrorID 和 ErrorInfo (页 320)

轴工艺对象 V1...3 的变量 (页 342)

MC\_Power: 启用、禁用轴 V1...3 (页 419)

MC\_Reset: 应答故障 V1...3 (页 425)

MC\_Home: 使轴回原点, 设置参考点 V1...3 (页 428)

MC\_Halt: 停止轴 V1...3 (页 433)

MC\_MoveAbsolute: 轴的绝对定位 V1...3 (页 436)

MC\_MoveRelative: 轴的相对定位 V1...3 (页 441)

MC\_MoveVelocity: 以预设的旋转速度移动轴 V1...3 (页 446)

MC\_MoveJog: 在点动模式下移动轴 V1...3 (页 451)

MC\_CommandTable: 按照运动顺序运行轴命令 V2...3 (页 455)

# 索引

## M

MC 插补器 OB, 27, 28

MC 伺服 OB, 27, 28

MC\_ChangeDynamic

参数, 410

指令, 409

MC\_CommandTable

参数, 407

指令, 406

MC\_Halt

参数, 383, 434

功能图, 385

指令, 382

MC\_Home

参数, 379

指令, 377

MC\_MoveAbsolute

参数, 387

功能图, 389

指令, 386

MC\_MoveJog

参数, 403

功能图, 405

指令, 402

MC\_MoveRelative

参数, 392

功能图, 394

指令, 391

MC\_MoveVelocity

功能图, 400

指令, 396

MC\_Power

参数, 369

功能图, 373

指令, 368

MC\_ReadParam

参数, 414

指令, 413

MC\_Reset, 374

MC\_WriteParam

参数, 417

指令, 416

MoveVelocity

参数, 397

## T

TO\_PositioningAxis, 53

## B

闭环控制, 102

步进电机, 11

## C H

插补器 OB, 27, 28

插入分隔符行 (Insert separator line), 138

插入空行 (Insert empty line), 138

## D

定位监控, 100

定位轴工艺对象

Actor 变量, 239

DynamicDefaults 变量, 260

DynamicLimits 变量, 259

ErrorBits 变量, 285

FollowingError 变量, 270

Modulo 变量, 257

PositionControl 变量, 269

PositionLimitsHW 变量, 265

PositionLimitsSW 变量, 262

PositionMonitoring 变量, 272

PTO 和 HSC 组态, 58, 307

Sensor[1] 变量, 243

Sensor[1].ActiveHoming 变量, 250

Sensor[1].Interface 变量, 245

Sensor[1].Parameter 变量, 247

Sensor[1].PassiveHoming 变量, 253

StandstillSignal 变量, 274

StatusBits 变量, 279

StatusDrive 变量, 277

StatusPositioning 变量, 275

StatusSensor 变量, 278

被动归位, 97

变量 ActualPosition, 237

变量 ActualVelocity, 238

常规动态组态, 82

单位变量, 255

更新变量, 287

工具, 50

归位变量, 267

机械变量, 256

机械组态, 74, 75, 311

基本参数, 56, 307

急停减速度组态, 86

计量单位设置, 57

扩展参数, 310

启用冲击限制器时的响应, 87

驱动器使能组态, 310

驱动器信号组态, 61, 310

驱动器准备就绪组态, 310

速度变量, 236

添加新对象, 53

调试概述, 51

位置变量, 235

硬件和软件组件, 46

硬件接口组态, 58, 307

用户单位组态, 309

在用户程序中可以更改动态组态参数, 89

在用户程序中可以更改归位组态参数, 98

诊断概览, 52

轴名称组态, 56, 307

主动归位, 96

组态被动归位, 94, 313

组态逼近/归位方向, 93

组态逼近/回原点方向, 315

组态参考点开关端, 313, 315

组态参考点开关端, 313, 315

组态参考点开关输入, 313, 314

组态参考点位置, 93, 95, 313, 315

组态冲击限制器, 83

组态窗口图标, 54

组态电机每转的距离, 74, 311

组态电机每转的脉冲数, 74, 311

组态反转方向信号, 75, 312

组态概述, 51

组态归位开关端, 93, 95

- 组态归位开关端, 93, 95
- 组态归位开关输入, 92, 95
- 组态归位速度, 93
- 组态回原点速度, 315
- 组态加速度, 83
- 组态加速时间, 83
- 组态减速度, 83
- 组态减速时间, 83
- 组态平滑时间, 84
- 组态起始位置偏移值, 93, 315
- 组态速度限值单位, 82
- 组态允许在硬限位开关处自动反向, 314
- 组态主动归位, 91
- 组态主动回原点, 314
- 组态最大速度/启动/停止速度, 82

## F

- 方向输出和行进方向  
关系, 20

## G

- 跟随误差监控, 101
- 工艺对象命令表: 快捷菜单命令, 138
- 归位
  - 回原点模式, 34

## J

- 基本知识
  - 必需, 3
- 加加速度限值
  - 功能, 33

## M

- 脉冲接口
  - 原理, 19
- 命令表工艺对象
  - Command.Command[1...32] 变量, 288
  - 常规组态, 132
  - 工具, 128
  - 基本参数, 132
  - 扩展参数, 150
  - 命令表组态, 132
  - 添加新对象, 129
  - 用途, 127
  - 组态步代码, 137
  - 组态持续时间, 136
  - 组态窗口图标, 130
  - 组态激活警告, 133
  - 组态命令类型, 134
  - 组态使用轴参数, 133
  - 组态速度, 136
  - 组态位置/行进路径, 135
  - 组态下一步, 137

## S H

- 适用范围
  - 手册, 3
- 手册
  - 适用范围, 3
  - 用途, 3

## C

- 伺服 OB, 27, 28
- 伺服电机, 11



**T**

添加分隔符行 (Add separator line), 138

添加空行 (Add empty line), 138

**D**

调节, 159

**T**

停止信号, 102

**W**

位置控制, 102

**X**

需要的基本知识, 3

**Y**

硬件和软件限位开关

    功能, 32

用途

    手册, 3

优化位置控制器, 159

运动控制 CPU S7-1200

    准则, 35

运动控制 S7-1200 的硬件配置, 11

**Z H**

重新初始化工艺对象, 182

重新启动工艺对象, 182

轴和命令表工艺对象

ErrorID 和 ErrorInfo 列表, 205