Modicon M218 Logic Controller

打包库
M218 打包库指南

05/2011

EIO0000001032. 00

www.schneider-electric.com

Schneider Electric

本文档中提供的信息包含有关此处所涉及产品之性能的一般说明和 / 或技术特性。

本文档并非用于 （也不代替）确定这些产品对于特定用户应用场合的适用性或可靠性。任何此类用户或集成者都有责任就相关特定应用场合或使用方面对产品执行适当且完整的风险分析、评估和测试。 Schneider Electric 或是其任何附属机构或子公司对于误用此处包含的信息而产生的后果概不负责。如果您有关于改进或更正此出版物的任何建议，或者从中发现错误，请通知我们。

未经 Schneider Electric 明确书面许可，不得以任何形式、通过任何电子或机械手段（包括影印）复制本文档的任何部分。

在安装和使用本产品时，必须遵守国家、地区和当地的所有相关的安全法规。出于安全方面的考虑和为了帮助确保符合归档的系统数据，只允许制造商对各个组件进行维修。

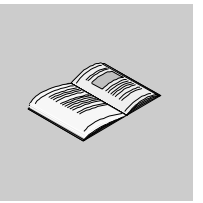
当设备用于具有技术安全要求的应用场合时，必须遵守有关的使用说明。

如果在我们的硬件产品上不正确地使用 Schneider Electric 软件或认可的软件，则可能导致人身伤害、损害或不正确的操作结果。

不遵守此信息可能导致人身伤害或设备损坏。

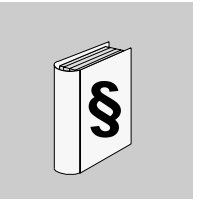
© 2011 Schneider Electric。保留所有权利。

目录



安全信息 5
关于本书 7
章 1 打包库功能块 9
VBagPTOMoveFast 功能块 9
附录 15
附录 A 一般信息 17
专用功能 18
有关管理和运动功能块管理的一般信息 19
附录 B 功能和功能块表示形式 21
功能与功能块之间的差异 22
如何通过 IL 语言使用功能或功能块 23
如何通过 ST 语言使用功能或功能块 26
索引 29

安全信息



重要信息

声明

在尝试安装、操作或维护设备之前，请仔细阅读下述说明并通过查看来熟悉设备。下述特别信息可能会在本文其他地方或设备上出现，提示用户潜在的危险，或者提醒注意有关阐明或简化某一过程的信息。



在“危险”或“警告”安全标签上添加此符号表示存在触电危险，如果不遵守使用说明，将导致人身伤害。



这是提醒注意安全的符号。提醒用户可能存在人身伤害的危险。请遵守所有带此符号的安全注意事项，以避免可能的人身伤害甚至死亡。

⚠ 危险

“危险”表示极可能存在危险，如果不遵守说明，可**导致**严重的人身伤害甚至死亡。

⚠ 警告

“警告”表示可能存在危险，如果不遵守说明，可**导致**严重的人身伤害甚至死亡，或设备损坏。

⚠ 注意
“注意”表示可能存在危险，如果不遵守说明，可导致严重的人身伤害或设备损坏。

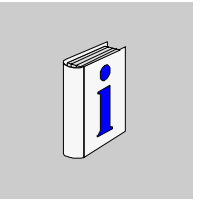
注意
注意（无安全警告符号），表示存在潜在的危险，如果忽视，可能导致设备损坏。

请注意

电气设备的安装、操作、维修和维护工作仅限于合格人员执行。对于使用本资料所引发的任何后果， Schneider Electric 概不负责。

专业人员是指掌握与电气设备的制造和操作相关的技能和知识的人员，他们经过安全培训能够发现和避免相关的危险。

关于本书



概览

文档范围

阅读本文档将使您了解 M218 控制器中提供的打包功能。

要使用本手册，您必须：

- 对 M218 有着透彻理解，包括其设计、功能和在控制系统中的实现。
- 熟练使用下列 IEC 61131-3 PLC 编程语言：
  - 功能块图 (FBD)
  - 梯形图 (LD)
  - 结构化文本 (ST)
  - 指令列表 (IL)
  - 顺序功能图 (SFC)

有效性说明

本文档已随 SoMachine V3.0 的发布进行了更新。

相关的文件

文件名称	参考编号
M218 Logic Controller 编程指南	EIO0000000827（英语）；EIO0000000830（简体中文）
M218 PTOPVM 库指南	EIO0000000839（英语）；EIO0000000840（简体中文）

您可以从我们的网站下载这些技术出版物和其它技术信息，网址是：www.schneider-electric.com。

关于产品的资讯

⚠ 警告

失去控制

- 任何控制方案的设计者都必须考虑到控制路径可能出现故障的情况，并为某些关键控制功能提供一种方法，使其在出现路径故障时，以及出现路径故障后恢复至安全状态。紧急停止和越程停止、断电和重启都属于关键控制功能。
- 对于关键控制功能，必须提供单独或冗余的控制路径。
- 系统控制路径可包括通讯链路。必须对暗含的无法预料的传输延迟或链接失效问题加以考虑。
- 遵守所有事故预防规定和当地的安全指南。[1]
- 为了保证正确运行，在投入使用前，必须对设备的每次执行情况分别进行全面测试。

如果不遵守这些说明，将会导致死亡、严重伤害或设备损坏。

[1] 有关详细信息，请参阅 NEMA ICS 1.1 （最新版）中的 “ 安全指导原则 - 固态控制器的应用、安装和维护 ” 以及 NEMA ICS 7.1 （最新版）中的 “ 结构安全标准及可调速驱动系统的选择、安装与操作指南 ” 或您特定地区的类似规定。

⚠ 警告

意外的设备操作

- 仅使用 Schneider Electric 认可的可与本设备配合使用的软件。
- 每次更改物理硬件配置后，请更新应用程序。

如果不遵守这些说明，将会导致死亡、严重伤害或设备损坏。

用户意见

欢迎对本书提出意见。您可以给我们发邮件，我们的邮件地址是 techcomm@schneider-electric.com。

打包库功能块

1

VBagPTOMoveFast 功能块

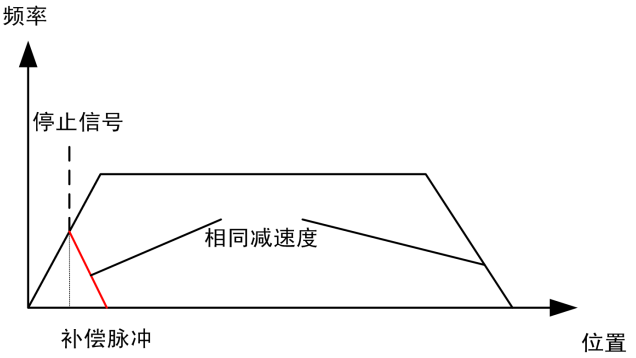
功能描述

此功能块指示以指定速度相对移动。

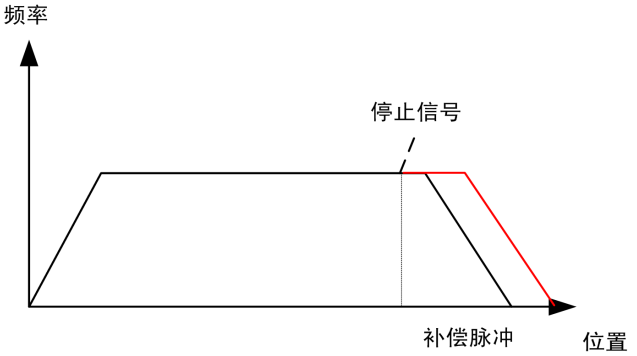
停止信号管理

发生停止信号时可能有 6 种可能的结果：

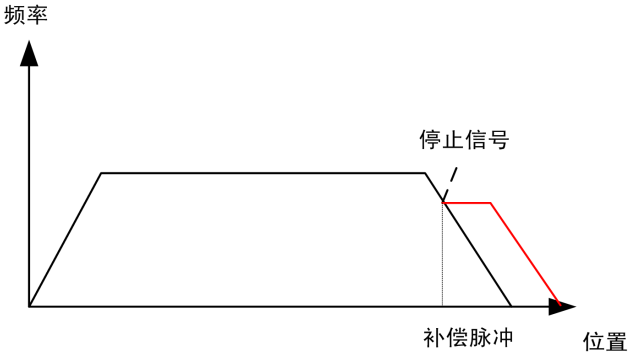
- 在加速期间触发停止信号：



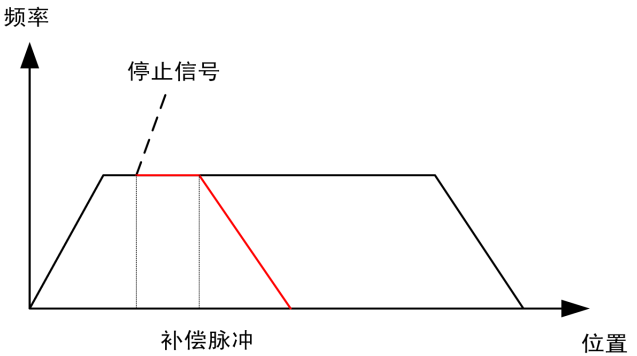
- 在减速前触发停止信号，且来自补偿的脉冲数大于其余 `TargetDistance` 脉冲数。



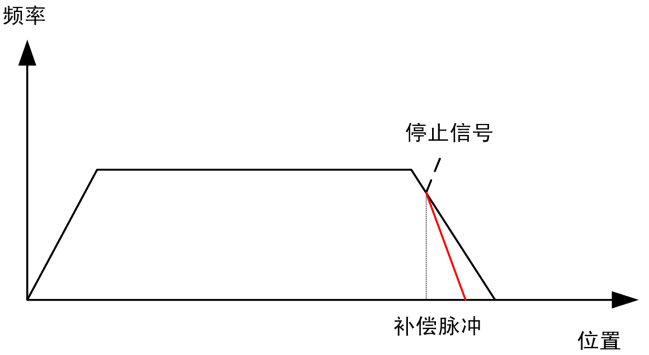
- 在减速期间触发停止信号，且来自补偿的脉冲数大于剩下的 TargetDistance 脉冲数。



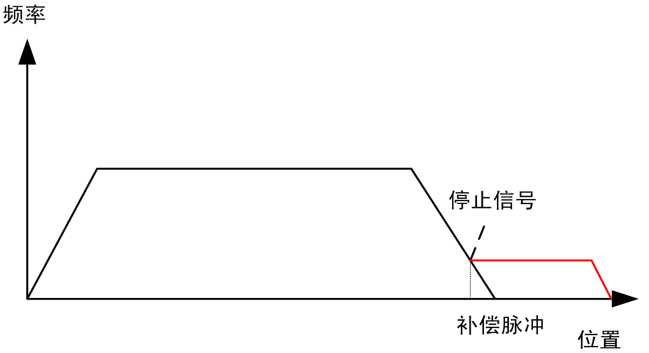
- 在减速前触发停止信号，且来自补偿的脉冲数小于剩下的 TargetDistance 脉冲数。



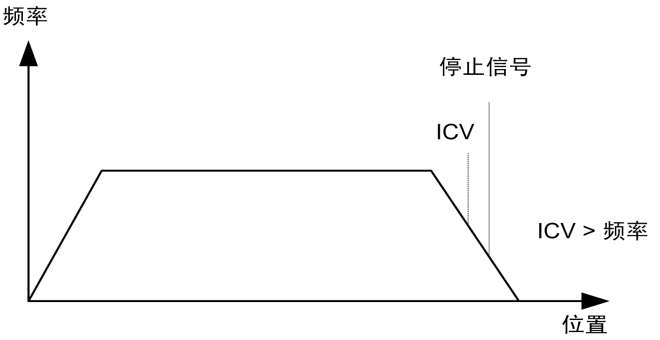
- 在减速期间触发停止信号，且来自补偿的脉冲数小于其余 TargetDistance 脉冲数。



- 如果在减速期间触发停止信号，则速度很低将导致很长的补偿时间。



为了避免长时间补偿，可使用 IgnoreCompVelocity 输入。
如果当前速度小于 IgnoreCompVelocity (ICV)，则将忽略停止信号，且移动将正常完成。



图形表示形式



IL 和 ST 表示形式

若要查看 IL 或 ST 语言的一般表示形式，请参阅*功能和功能块表示形式 (参见 第 21 页)* 一章。

I/O 变量描述

下表介绍输入变量：

输入	类型	注释
PTO_REF_IN	PTO_REF	对 PTO 轴的参考。 连接到 **PTOSimple** 的 PTO_REF，或**管理**或**运动**输出引脚功能块的 PTO_REF_OUT。
Execute	BOOL	在上升沿启动功能块的执行。 如果为 FALSE，则在其执行终结时，复位功能块的输出。
StartInput	FAST_PTO_TRIGGER	启动功能块的物理输入
DelayStart	DWORD	在移动后延迟功能块执行被触发。（0 到 1999 毫秒）
Acceleration	DWORD	以赫兹 / 毫秒或毫秒 （根据配置）表示的加速度。 范围 （毫秒）：**最大加速度** ... **最大软件限制**
方向	PTO_DIRECTION	移动的方向。
速度	DWORD	以赫兹表示的目标速度。 范围：1 到最大频率
TargetDistance	DWORD	要执行的目标距离。 如果达到目标距离，则移动停止。
StopInput	FAST_PTO_TRIGGER	停止功能块的物理输入

输入	类型	注释
补偿	DWORD	停止请求后的其他脉冲数。
减速	DWORD	以赫兹 / 毫秒或毫秒 （根据配置）表示的减速度。 范围 （赫兹 / 毫秒）：1... **减速度最大值** 范围 （毫秒）：**最大减速度** ... **最大软件限制**
IgnoreCompVelocity	DWORD	在除了加速时段的任何时间内，当发生停止信号时，如果当前速度低于此值 （单位为赫兹），则将忽略停止信号，且移动将正常完成。

注意：加速和减速斜坡不能超过 4,294,967,2952,147,483,647 个脉冲。它会以最大频率 100 kHz 将加速 / 减速斜坡的持续时间限制为 80 秒。

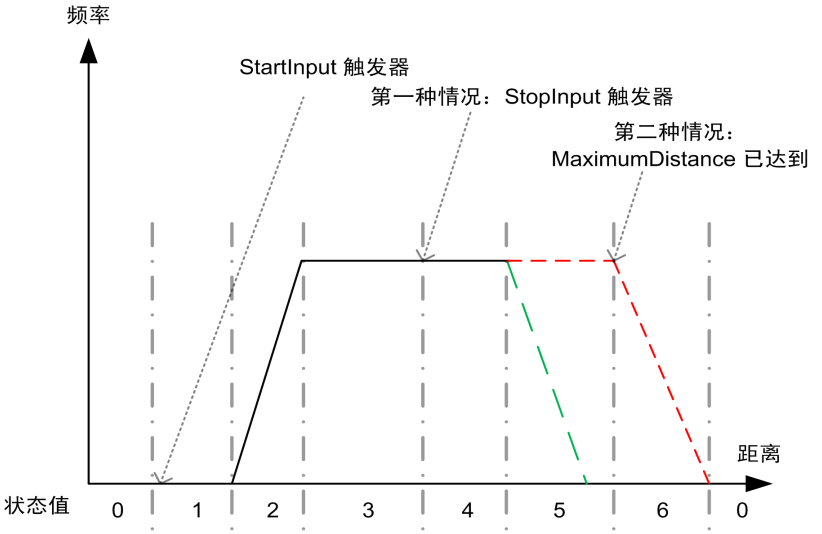
下表介绍输出变量：

PTOPWM_ERR_TYPE

当 Error 为 TRUE 时：检测到的错误的类型。

注意：有关 Done、 Busy、 CommandAborted 和 Execution 引脚的详细信息，请参阅功能块管理的一般信息 *(参见第 19 页)*

下图说明 PTOFastMove 配置文件及其相关的状态：



附录



概述

本附录摘录了部分编程指南，以方便用户获得对库文档的技术理解。

本附录包含了哪些内容？

本附录包含了以下章节：

章	章节标题	页
A	一般信息	17
B	功能和功能块表示形式	21

一般信息

A

概述

本章介绍的信息通用于 PTO 和 HSC 管理和运动功能。

本章包含了哪些内容？

本章包含了以下主题：

主题	页
专用功能	18
有关管理和运动功能块管理的一般信息	19

专用功能

专用输出

脉冲串输出、频率发生器、脉冲宽度调制、高速计数器或编码器使用的输出只能通过功能块访问。不能在应用程序中直接对其进行读写。

使用这些专用功能时，请遵照下列注意事项，以避免这些功能及其控制的设备出现意外设备操作：

- 请勿在不同的程序任务中使用同一个功能块实例。
- 请勿在功能块处于活动状态 （正在执行）时更改功能块参考 (••_REF_IN)。

⚠ 警告

意外的设备操作

- 请勿在多个任务中使用某个功能块的同一实例。
- 请勿在功能块处于活动状态 （正在执行）时修改功能块参考 (••_REF_IN)。

如果不遵守这些说明，将会导致死亡、严重伤害或设备损坏。

有关管理和运动功能块管理的一般信息

输入变量的管理

该功能块在 Execute 输入的上升沿上启动。

此时无需对输入变量进行任何进一步的修改。

按照 IEC 61131-3 标准，如果功能块有任何变量输入缺失 （即断开或未连接），则使用上一次调用功能块实例的值。在此情况下，第一次调用时将应用初始配置值。因此，功能块最好始终带有特定于其输入的已知值，这样有助于消除调试程序的麻烦。对于 HSC 和 PTO 功能块，最好只使用一次实例，且该实例必须位于主任务中。

输出变量的管理

Done、 InVelocity 或 InFrequency 输出与 Busy、 CommandAborted 和 Error 输出相互排斥：在一个功能块上，只能有一个输出为 TRUE。如果 Execute 输入为 TRUE，则其中一个输出为 TRUE。

在 Execute 输入的上升沿，会设置 Busy 输出。在功能块执行过程中，此 Busy 输出保持已设置状态，并在某一其他输出 （Done、 InVelocity、 InFrequency、 CommandAborted 和 Error）的上升沿复位。

当功能块成功执行完毕时，会设置 Done、 InVelocity 或 InFrequency 输出。

当功能块的执行被另一功能块中断时，则改为设置 CommandAborted 输出。

当功能块的执行由于检测到错误而结束时，则会设置 Error 输出，并通过 ErrId 输出给出检测到的错误编号。

Done、 InVelocity、 InFrequency、 Error、 ErrID 和 CommandAborted 输出在 Execute 的下降沿复位。如果 Execute 输入在执行完成之前复位，则在执行结束时，这些输出的设置状态将持续一个任务循环。

当功能块的某个实例在完成之前收到新的 Execute 时，则对于以前的操作，功能块不返回任何反馈，比如 Done。

错误处理

所有功能块都有 2 个输出，可以报告在执行功能块期间检测到的错误。

- 检测到错误时， Error = TRUE。
- ErrID 在 Error = TRUE 时返回检测到的错误 ID。

功能和功能块表示形式

B

概述

每个功能可以使用以下语言表示：

- IL：指令列表
- ST：结构化文本
- LD：梯形图
- FBD：功能块图
- CFC：连续功能图

本章提供功能和功能块表现形式示例，并解释如何将它们用于 IL 和 ST 语言。

本章包含了哪些内容？

本章包含了以下主题：

主题	页
功能与功能块之间的差异	22
如何通过 IL 语言使用功能或功能块	23
如何通过 ST 语言使用功能或功能块	26

功能与功能块之间的差异

功能

功能：

- 是返回一个直接结果的 **POU** （程序组织单元）
- 通过其名称 （而不是通过**实例**）直接调用
- 从一个调用到另一个调用不会保持原有状态
- 可以用作其他表达式中的操作数

示例：布尔操作符 (AND)、计算、转换 (BYTE_TO_INT)

功能块

功能块：

- 是返回一个或多个输出的 **POU** （程序组织单元）
- 始终通过**实例** （具有专用名称和变量的功能块副本）进行调用
- 每个**实例**在从一个调用到另一个调用会保持原有状态 （输出和内部变量）

示例：定时器、计数器

在下面的示例中， `Timer_ON` 是功能块 `TON` 的实例：

```
PROGRAM MyProgram_ST
VAR
    Timer_ON: TON;  // Function Block Instance
    Timer_RunCd: BOOL;
    Timer_PresetValue: TIME := T#5S;
    Timer_Output: BOOL;
    Timer_ElapsedTime: TIME;
END_VAR
```

```
Timer_ON(
    IN:=Timer_RunCd,
    PT:=Timer_PresetValue,
    Q=>Timer_Output,
    ET=>Timer_ElapsedTime);
```

如何通过 IL 语言使用功能或功能块

一般信息

本部分介绍如何使用 IL 语言实现功能和功能块。

我们以功能 IsFirstMastCycle、功能 SetRTCDrift 和功能块 TON 为例来演示实现的过程。

通过 IL 语言使用功能

以下步骤描述如何插入采用 IL 语言的功能：

步骤	操作
1	通过**指令列表**语言打开 POU 或创建新 POU。 **注意：**此处未详细介绍创建 POU 的步骤。有关详细信息，请参阅 SoMachine 全局帮助。
2	创建功能所需的变量。
3	如果功能具有 1 个或多个输入，则使用 LD 指令开始加载第一个输入。
4	在下面插入新行，并执行以下操作： ● 在操作符列 （左侧字段）中键入功能的名称，或 ● 使用**输入助手**选择功能 （在上下文菜单中选择**插入运算块**）。
5	如果功能具有多个输入，则在使用**输入助手**时，会在右侧字段中使用 ??? 自动创建必需的行数。使用与输入顺序对应的适当值或变量来替换 ???。
6	插入新行，以便将功能的结果存储到适当的变量中：在操作符列 （左侧字段）中键入 ST 指令，并在右侧字段中键入变量名。

要阐释该过程，请考虑下面以图形方式表示的功能 IsFirstMastCycle （不带输入参数）和功能 SetRTCDrift （带输入参数）：

功能	图形表示形式
不带输入参数： IsFirstMastCycle	
带输入参数： SetRTCDrift	

在 IL 语言中，功能名称直接用在**操作符列**中：

功能	SoMachine POU IL 编辑器中的表示形式
不带输入参数的功能的 IL 示例： IsFirstMastCycle	1 PROGRAM MyProgram_IL 2 VAR 3 FirstCycle: BOOL; 4 END_VAR 5 1 IsFirstMastCycle ST FirstCycle
带输入参数的功能的 IL 示例： SetRTCDrift	1 PROGRAM MyProgram_IL 2 VAR 3 myDrift: SINT (-29..29) := 5; 4 myDay: DAY_OF_WEEK := SUNDAY; 5 myHour: HOUR := 12; 6 myMinute: MINUTE; 7 myDiag: RTCSETDRIFT_ERROR; 8 END_VAR 9 1 LD myDrift SetRTCDrift myDay myHour myMinute ST myDiag

通过 IL 语言使用功能块

以下步骤描述如何插入采用 IL 语言的功能块：

步骤	操作
1	通过**指令列表**语言打开 POU 或创建新 POU。 **注意：**此处未详细介绍创建 POU 的步骤。有关详细信息，请参阅 SoMachine 全局帮助。
2	创建功能块所需的变量 （包括实例名称）。

步骤	操作
3	使用 CAL 指令调用功能块： ● 使用**输入助手**选择 FB （在上下文菜单中右键单击并选择**插入运算块**）。 ● 会自动创建 CAL 指令和必要的 I/O。 每个参数 (I/O) 都是一条指令： ● 输入的值通过 “:=” 进行设置。 ● 输出的值通过 “=>” 进行设置。
4	在 CAL 右侧字段中，使用实例名称替换 ???。
5	使用适当的变量或立即值替换其他 ???。

要阐释该过程，请考虑下面以图形方式表示的 TON 功能块示例：

功能块	图形表示形式
TON	

在 IL 语言中，功能块名称直接用在**操作符列**中：

功能块	SoMachine POU IL 编辑器中的表示形式

```
1  PROGRAM MyProgram_IL
2  VAR
3      Timer_ON: TON;  // Function Block instance declaration
4      Timer_RunCd: BOOL;
5      Timer_PresetValue: TIME := T#5S;
6      Timer_Output: BOOL;
7      Timer_ElapsedTime: TIME;
8  END_VAR
9
```

```
1  CAL          Timer_ON(
              IN:= Timer_RunCd,
              PT:= Timer_PresetValue,
               Q=> Timer_Output,
              ET=> Timer_ElapsedTime)
```

如何通过 ST 语言使用功能或功能块

一般信息

本部分介绍如何使用 ST 语言实现功能和功能块。

我们以功能 SetRTCDrift 和功能块 TON 为例演示实现的过程。

通过 ST 语言使用功能

以下步骤描述如何插入采用 ST 语言的功能：

步骤	操作
1	通过**结构化文本**语言打开 POU 或创建新 POU。 **注意：**此处未详细介绍创建 POU 的步骤。有关详细信息，请参阅 SoMachine 全局帮助。
2	创建功能所需的变量。
3	在 **POU ST 编辑器**中，使用功能 ST 语言的常规语法。常规语法为： `FunctionResult:= FunctionName(VarInput1, VarInput2,..VarInputx);`

要阐释该过程，请考虑下面以图形方式表示的功能 SetRTCDrift：

功能	图形表示形式
SetRTCDrift	

此功能的 ST 语言如下所示：

```
myRTCAdjust:= SetRTCDrift(myDrift, myDay, myHour, myMinute);
```

通过 ST 语言使用功能块

以下步骤描述如何插入采用 ST 语言的功能块：

步骤	操作
1	通过**结构化文本**语言打开 POU 或创建新 POU。 **注意：**此处未详细介绍创建 POU 的步骤。有关详细信息，请参阅 SoMachine 全局帮助。

步骤	操作
2	创建功能块所需的输入和输出变量以及实例： ● 输入变量是功能块所需的输入参数 ● 输出变量接收功能块返回的值
3	在 **POU ST 编辑器**中，使用功能块 ST 语言的常规语法。常规语法为： `FunctionBlock_InstanceName(Input1:=VarInput1, Input2:=VarInput2,...Ouput1=>VarOutput1, Ouput2=>VarOutput2,...);`

要阐释该过程，请考虑下面以图形方式表示的 TON 功能块示例：

功能块	图形表示形式
TON	

下表显示了采用 ST 语言的功能块调用的示例：

功能块	SoMachine POU ST 编辑器中的表示形式

TON	1 PROGRAM MyProgram_ST
	2 VAR
	3 Timer_ON: TON; // Function Block Instance
	4 Timer_RunCd: BOOL;
	5 Timer_PresetValue: TIME := T#5S;
	6 Timer_Output: BOOL;
	7 Timer_ElapsedTime: TIME;
	8 END_VAR
	
	1 Timer_ON(
	2 IN:=Timer_RunCd,
	3 PT:=Timer_PresetValue,
	4 Q=>Timer_Output,
	5 ET=>Timer_ElapsedTime);

索引



Busy
    状态变量的管理 , *19*
CommandAborted
    状态变量的管理 , *19*
Done
    状态变量的管理 , *19*
ErrID
    状态变量的管理 , *19*
    错误处理 , *19*
Error
    状态变量的管理 , *19*
    错误处理 , *19*
Execute
    状态变量的管理 , *19*
VBagPTOMoveFast
    功能块 , *9*
专用功能 , *18*
功能
    功能与功能块之间的差异 , *22*
    如何通过 IL 语言使用功能或功能块 , *23*
    如何通过 ST 语言使用功能或功能块 , *26*
状态变量的管理
    Busy, *19*
    CommandAborted, *19*
    Done, *19*
    ErrID, *19*
    Error, *19*
    Execute, *19*
运动功能块
    VBagPTOMoveFast , *9*
错误处理
    ErrID, *19*
    Error, *19*