

Modicon M218 Logic Controller

高速计数
M218 HSC 库指南

11/2010

E1000000000838.00

www.schneider-electric.com



本文档中提供的信息包含有关此处所涉及产品的性能的一般说明和 / 或技术特性。本文档并非用于（也不代替）确定这些产品针对特定用户应用的适用性或可靠性。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。

© 2010 Schneider Electric。保留所有权利。

目录

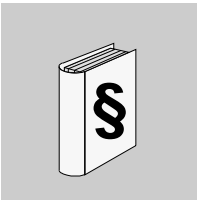


	安全信息	7
	关于本书	9
部分 I	高速计数器概述	11
章 1	M218 内嵌功能	13
	HSC 内嵌功能	14
	Simple 类型概述	16
	Main 类型概述	17
	选择您的 HSC	18
部分 II	一次性模式	21
章 2	一次性模式原理	23
	一次性模式原理介绍	23
章 3	使用 Simple 类型的一次性	25
	概要图	26
	一次性模式下的 simple 类型的配置	27
	Simple 类型编程	29
	调整参数	31
章 4	使用 Main 类型的一次性	33
	概要图	34
	一次性模式下的 Main 类型的配置	35
	Main 类型编程	38
	调整参数	41
部分 III	模数回路模式	43
章 5	模数回路原理	45
	一次性模式原理介绍	45
章 6	使用 Simple 类型的模数回路	47
	概要图	48
	模数回路模式下的 Simple 类型配置	49
	Simple 类型编程	51
	调整参数	53

章 7	使用 Main 类型的模数回路	55
	概要图	56
	模数回路模式下的 Main 类型配置	57
	Main 类型编程	60
	调整参数	63
部分 IV	自由大型模式	65
章 8	自由大型模式原理	67
	自由大型模式原理介绍	67
章 9	使用 Main 类型的自由大型	73
	概要图	74
	自由大型模式下 Main 类型的配置	75
	Main 类型编程	78
	调整参数	81
部分 V	事件计数 模式	83
章 10	事件计数原理	85
	事件计数模式原理介绍	85
章 11	使用 Main 类型的事件计数	87
	概要图	88
	事件计数模式下 Main 类型的配置	89
	Main 类型编程	92
	调整参数	95
部分 VI	频率计 模式	99
章 12	频率计原理	101
	频率计模式原理介绍	101
章 13	使用 Main 类型的频率计	105
	概要图	106
	频率计模式下 Main 类型的配置	107
	Main 类型编程	109
	调整参数	112
部分 VII	可选功能	113
章 14	比较功能	115
14.1	使用 Main 类型的比较	115
	使用 Main 类型的比较原理	116
	对使用 Main 类型的比较进行配置	119
	外部事件配置	120
章 15	捕捉功能	121
15.1	使用 Main 类型的捕捉	121
	使用 Main 类型的捕捉原理	122
	对使用 Main 类型的捕捉进行配置	124

章 16 同步、和启用功能	125
同步功能	126
启用功能	127
附录	129
附录 A 一般信息	131
专用功能	132
有关管理和运动功能块管理的一般信息	133
附录 B 数据类型	135
HSC_ERR_TYPE: HSC 的错误变量的类型	136
HSC_PARAMETER_TYPE: 在 HSC 变量上要获取或设置的参数类型	137
HSC_REF: HSC 参考值	138
HSC_TIMEBASE_TYPE: HSC 时基变量的类型	139
附录 C 功能块	141
HSCGetCapturedValue: 返回捕捉寄存器的内容	142
HSCGetDiag: 提供 HSC 上检测到的错误的详细信息	144
HSCGetParam: 返回 HSC 的参数	146
HSCSetParam: 调整以下参数 HSC	148
HSCMain: HSC Main 功能块	150
HSCSimple: HSC Simple 功能块	154
附录 D 功能和功能块表示形式	157
功能与功能块之间的差异	158
如何通过 IL 语言使用功能或功能块	159
如何通过 ST 语言使用功能或功能块	162
术语表	165
索引	191

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，提示用户潜在的危險，或者提醒注意有关阐明或简化某一过程的信息。



在“危險”或“警告”安全标签上添加此符号表示存在触电危險，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險。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。

⚠ 危險

“危險”表示极可能存在危險，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”表示可能存在危險，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

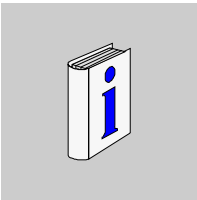
注意(无安全警告符号)，表示存在潜在的危险，如果忽视，可能导致设备损坏。

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

本文档向您介绍 M218 Controller 中提供的高速计数器 (HSC) 功能和变量。
本文档介绍 M218 HSC 库的功能和变量。

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 V2.0 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）； EIO0000000828（简体中文）
Modicon TM2 高速计数器模块硬件指南	EIO0000000022（英语）； EIO0000000023（法语）； EIO0000000024（德语）； EIO0000000025（西班牙语）； EIO0000000026（意大利语）； EIO0000000027（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

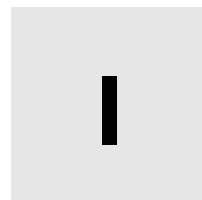
- 仅使用 Schneider Electric 认可的可与本设备配合使用的软件。
- 每次更改物理硬件配置后，请更新应用程序。

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。

高速计数器概述



M218 内嵌功能

1

概述

本章介绍 Modicon M218 Logic Controller 的内嵌功能。

专用于内嵌功能的输入和输出个数取决于控制器参考号 (参见 *Modicon M218 Logic Controller, 编程指南*)。

本章包含了哪些内容？

本章包含了以下主题：

主题	页
HSC 内嵌功能	14
Simple 类型概述	16
Main 类型概述	17
选择您的 HSC	18

HSC 内嵌功能

概述

HSC 功能可以对来自与专用快速输入相连的传感器、编码器、开关等的脉冲进行快速计数

HSC 的类型有 2 种：

- **Simple** 类型：单个输入的计数器 (参见第 16 页)。
- **Main** 类型：最多可使用 2 个快速输入和 2 个反射输出 (参见第 17 页) 的计数器。

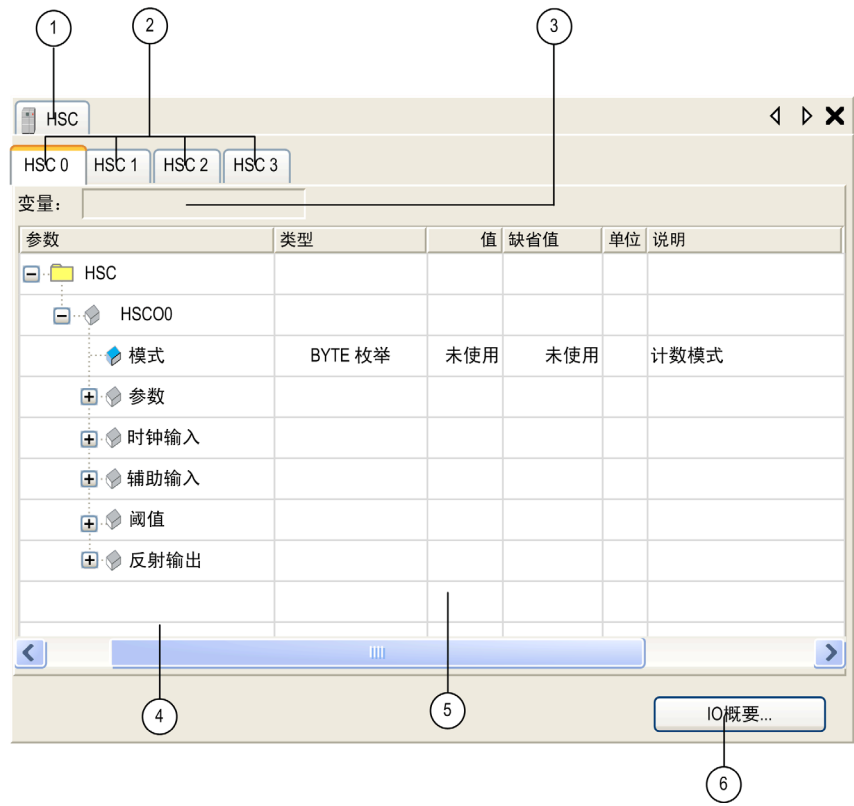
访问配置菜单

按照以下步骤可通过**配置**菜单访问 HSC 内嵌功能配置窗口：

步骤	说明
1	单击 配置 菜单： 
2	双击您需要的控制器。 注意： 还可以右键单击您需要的控制器并选择 编辑参数 。
3	在任务面板中，单击 内嵌功能 → HSC ： 

HSC 配置窗口

下图是用于配置 HSC 的 HSC 配置窗口的示例。



下表介绍 HSC 配置窗口的字段：

标记	操作
1	选择 HSC 选项卡可访问每个 HSC 配置窗口。
2	根据需要配置的 HSC 通道，选择其中一个选项卡。
3	选择需要的 HSC 类型（“Simple”或“Main”）后，使用 变量 字段更改实例。
4	如果参数折叠，则可以通过单击加号进行展开。随后便可访问每个参数的设置。
5	配置窗口，在其中根据使用的模式确定 HSC 参数。
6	单击 IO 概要 按钮后，随即显示“IO 概要”窗口。使用此窗口可检查配置 I/O 映射。

有关配置参数的详细信息，请参阅 M218 HSC 选择矩阵 (参见第 18 页)。

Simple 类型概述

概述

Simple 类型是单个输入计数器。

Simple 类型 HSC 可向 / 从预定义值加计数或减计数。

达到计数后可以对操作进行编程。这些操作在已编程任务（MAST 或 FAST）环境中完成，但没有触发事件。

Simple 类型模式

Simple 类型支持 2 个可配置的计数模式（仅限于单相脉冲）：

One-shot（参见第 25 页）：在此模式下，A 输入上每应用一次脉冲，计数器当前值寄存器就会（从用户定义的值）递减一次，直到计数器达到 0。

模数回路（参见第 47 页）：在此模式下，计数器从 0 计数到用户定义的模数值，然后返回到 0 并重新启动计数，周而复始，反复执行。

性能

Simple 类型的最大频率是 100 kHz。

带 TM218LDA24DRN 的 **Simple** 类型的最大频率是 5 kHz。

Main 类型概述

概述

Main 这种类型的计数器最多可使用 4 个快速输入和 2 个反射输出。

Main 类型模式

Main 类型支持单相（1 个输入）或双相（2 个输入）脉冲上的以下计数模式：

一次性（参见第 33 页）：在此模式下，A 输入上每应用一次脉冲，计数器当前值寄存器就会（从用户定义的值）递减一次，直到计数器达到 0。

模数回路（参见第 55 页）：在此模式下，计数器从 0 计数到用户定义的模数值，然后返回到 0 并重新启动计数，周而复始，反复执行。反过来，计数器从模数值减计数到 0，然后预设为模数值，再重新启动计数。

自由大型（参见第 73 页）：在此模式下，计数器的行为类似于高范围的加和减计数器。可用于单个编码器。

事件计数（参见第 87 页）：在此模式下，计数器对在用户配置的时基期间接收的事件数进行累计。

频率计（参见第 105 页）：在此模式下，计数器测量事件的频率。频率是每秒的事件数 (Hz)。

可选功能

可选功能的配置取决于选择的模式：

- 用于操作计数器（启用、同步）或捕捉当前计数值的硬件输入
- 最多 2 个阈值
- 最多 2 次事件（每个阈值触发 1 次）可与外部任务进行关联
- 最多 2 个反射输出

性能

Main 类型的最大频率是 100 kHz。

选择您的 HSC

概述

本节概述所有 HSC 及其功能，以帮助您选择系统所需的 HSC。

HSC 矩阵

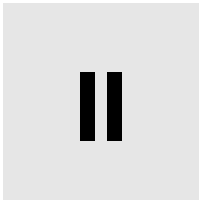
下表概述了所有可用 HSC 及其在所请求模式下的各种规格：

模式	功能	Simple 类型	Main 类型
一次性	计数模式	减计数	减计数
	最大计数频率	100 (5)* kHz	100 kHz
	通过 HSC 物理输入启用	否	是
	通过 HSC 物理输入同步 / 预设	否	是
	比较功能	否	是， 2 个阈值， 2 个输出和事件
	捕捉功能	否	是， 1 个捕捉寄存器
模数回路	计数模式	加计数	单相 加 / 减计数 脉冲 / 方向 积分
	最大计数频率	100 (5)* kHz	100 kHz
	通过 HSC 物理输入启用	否	是， 与第二个计数 输入异或
	通过 HSC 物理输入同步 / 预设	否	是
	比较功能	否	是， 2 个阈值， 2 个输出和事件
	捕捉功能	否	是， 1 个捕捉寄存器

模式	功能	Simple 类型	Main 类型
自由大型	计数模式	-	加 / 减计数 脉冲 / 方向 积分
	最大计数频率	-	100 kHz
	通过 HSC 物理输入启用	-	否
	通过 HSC 物理输入同步 / 预设	-	是
	比较功能	-	是，2 个阈值，2 个输出和事件
	捕捉功能	-	是，1 个捕捉寄存 器
事件	计数模式	-	给定时间段内的脉 冲计数
	最大计数频率	-	100 kHz
	通过 HSC 物理输入启用	-	否
	通过 HSC 物理输入同步 / 预设	-	是
	比较功能	-	否
	捕捉功能	-	否
频率计	计数模式	-	时基内的脉冲计数
	最大计数频率	-	100 kHz
	通过 HSC 物理输入启用	-	否
	通过 HSC 物理输入同步 / 预设	-	否
	比较功能	-	否
	捕捉功能	-	否

*TM218LDA24DRHN 的频率

一次性模式



概述

本部分介绍在**一次性**模式下如何使用 HSC。

本部分包含了哪些内容？

本部分包括以下各章：

章	章节标题	页
2	一次性 模式原理	23
3	使用 Simple 类型的一次性	25
4	使用 Main 类型的一次性	33

一次性模式原理介绍

概述

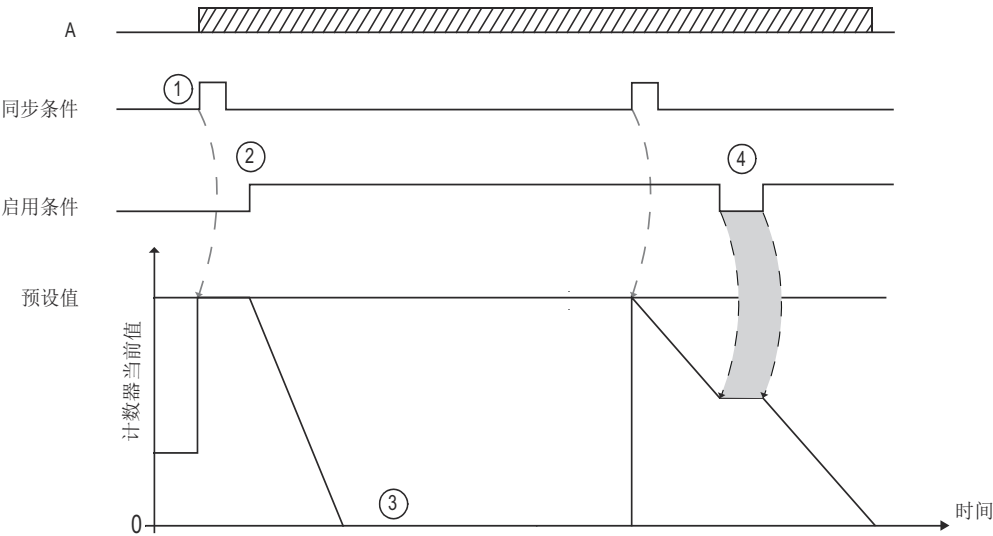
计数器由同步跳变沿激活，同时加载预设值。

启用计数时，每个应用到输入的脉冲都递减当前值。计数器在其当前值达到 0 时停止。

即使新的脉冲应用到输入，计数器值仍然保持为 0。

要再次激活计数器，需要新的同步。

原理图



下表介绍上图的各个阶段：

阶段	操作
1	在同步条件的上升沿上，预设值被加载到计数器（不考虑当前值），并且计数器被激活。
2	当启用条件 = 1 时，当前的计数器值依据输入 A 上的各个脉冲递减，直至达到 0。
3	计数器等待同步条件的下一个上升沿出现。 注： 此时，输入 A 上的脉冲对计数器不起作用。
4	当启用条件 = 0 时，计数器忽略来自输入 A 的脉冲，并保留其当前值，直到启用条件再次 = 1。在启用输入的上升沿，计数器从保留的值开始恢复输入 A 的脉冲的计数。

注意：启用和同步条件取决于配置。启用（参见第 127 页）和同步（参见第 126 页）功能中对这些条件进行了介绍。

使用 Simple 类型的一次性



概述

本章介绍如何在使用 **Simple** 类型的一次性模式下实现高速计数器。

本章包含了哪些内容？

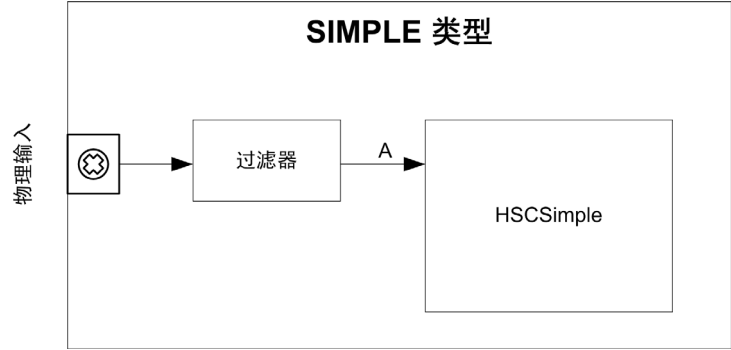
本章包含了以下主题：

主题	页
纲要图	26
一次性模式下的 simple 类型的配置	27
Simple 类型编程	29
调整参数	31

概要图

概要图

下图简要介绍一次性模式下的 Simple 类型：

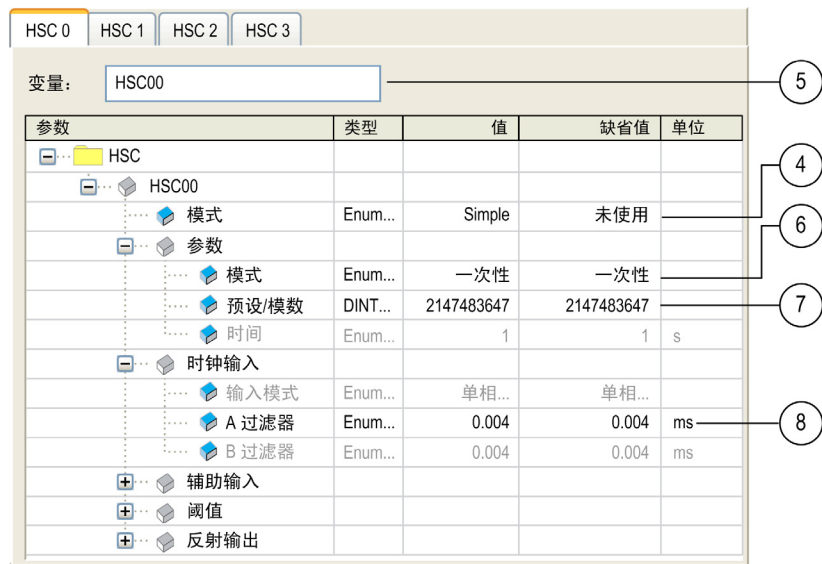


A 是高速计数器的计数输入。

一次性模式下的 simple 类型的配置

配置窗口

下图显示一次性模式下 **Simple** 类型的配置窗口。
项目符号中的数字对应于配置过程表中的数字：



配置过程

按照以下过程配置一次性模式下的 **Simple** 类型：

步骤	操作
1	进入配置屏幕。
2	双击控制器。
3	选择内嵌功能 →HSC 结果：HSC Simple 配置窗口随即打开。
4	通过选择模式 →Simple，从下拉菜单将类型设置为 Simple。
5	随即创建 Simple 类型的实例，您可以通过变量字段重新命名该实例。
6	通过选择参数 →模式 →一次性，从下拉菜单将模式设置为一次性
7	从参数 →预设 / 模数 设置预设值
8	通过选择时钟输入 →A 过滤器，从下拉菜单设置抗跳动过滤器值

IO 摘要

输入 / 输出配置显示在 IO 摘要窗口中，可通过 **IO 摘要**按钮打开：

IO 摘要

输入

通道	配置
I0	HSC 0
I1	HSC 1
I2	HSC 2
I3	HSC 3
I4	
I5	
I6	
I7	
I8	
I9	
I10	
I11	
I12	
I13	

输出

通道	配置
Q0	
Q1	
Q2	
Q3	
Q4	
Q5	
Q6	
Q7	
Q8	
Q9	

OK

请参阅硬件指南以获取接线详细信息。(参见 *Modicon M218 Logic Controller, 硬件指南*)

可编程过滤器

Simple 类型输入的过滤值确定计数器最大频率，如下表所示：

输入	过滤器值	最大计数器频率
A	0.004 毫秒	100 kHz
	0.4 毫秒	1 kHz
	1.2 毫秒	350 Hz
	4 毫秒	100 Hz

Simple 类型编程

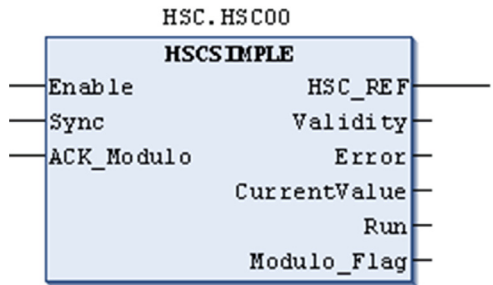
概述

Simple 类型始终由 HSCSimple (参见第 154 页) 功能块进行管理。

注意： 如果使用 HSCSimple 功能块管理其他 HSC 类型，则在编译时显示错误代码。

添加 HSCSimple 功能块

步骤	说明
1	使用插入运算块助手插入 HSCSimple 功能块。可在以下路径找到此功能块： 功能块（库）→M218_HSC →计数器 →HSCSimple
2	输入 Simple 类型实例的名称（在配置中定义，步骤 5），或通过单击以下内容查找功能块实例：  使用输入助手，可以在以下路径选择 HSC 实例： 全局变量 →<MyController> →内嵌功能 →HSC



I/O 变量用法

下表介绍了功能块的不同引脚在一次性模式中的使用方法。

下表介绍了输入变量：

输入	类型	注释
Enable	BOOL	TRUE = 准许对当前计数器值进行更改。
Sync	BOOL	在上升沿预设和启动计数器
ACK_Modulo	BOOL	未使用

下表介绍了输出变量：

输出	类型	注释
HSC_REF	HSC_REF (参见第 138 页)	HSC 的参考。 与管理功能块的 HSC_REF_IN 输入引脚配合使用。
Validity	BOOL	TRUE = 表示功能块上的输出值有效。
Error	BOOL	TRUE = 表示检测到错误。 使用 HSCGetDiag (参见第 144 页) 功能块获得有关此检测到的错误的详细信息。
Run	BOOL	TRUE = 计数器正在运行。 当当前值 达到 0 时切换为 0。要重新启动计数器，则需要 Sync 存在上升沿。
Modulo_Flag	BOOL	不相关
CurrentValue	DWORD	计数器的当前计数值。

调整参数

概述

下表中介绍的参数列表可使用 HSCGetParam (参见第 146 页) 或 HSCSetParam (参见第 148 页) 功能块进行读取或修改。

注意：通过程序设置的参数将覆盖在 HSC 配置窗口中配置的参数值。初始配置参数会在 冷启动或热启动 (参见 *Modicon M218 Logic Controller, 编程指南*) 后恢复。

可调整参数

下表列出的是来自 HSC_PARAMETER_TYPE (参见第 137 页) 的参数，可在程序运行过程中读取或修改这些参数：

参数	描述
HSC_PRESET	获取或设置 HSC 的预设值

使用 Main 类型的一次性



概述

本章介绍如何在使用 **Main** 类型的一次性模式下实现高速计数器。

本章包含了哪些内容？

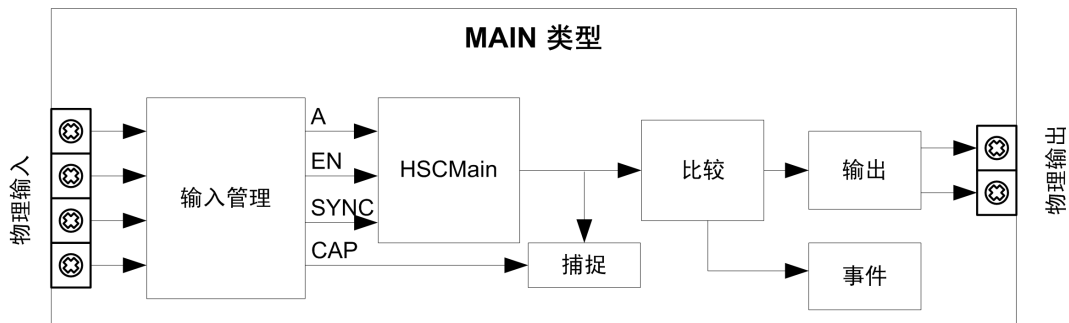
本章包含了以下主题：

主题	页
纲要图	34
一次性模式下的 Main 类型的配置	35
Main 类型编程	38
调整参数	41

纲要图

纲要图

下图简要介绍一次性模式下的 **Main** 类型：



A 是计数器的计数输入。

EN 是计数器的启用输入。

CAP 是计数器的捕捉输入。

SYNC 是计数器的同步输入。

可选功能

除了一次性模式，**Main** 类型还可以提供以下功能：

- 比较 (参见第 115 页)
- 捕捉 (参见第 121 页)
- 通过物理输入启用 (参见第 127 页)
- 通过物理输入同步 (参见第 126 页)

一次性模式下的 Main 类型的配置

配置窗口

下图显示一次性模式下 Main 类型的配置窗口。
项目符号中的数字对应于配置过程表中的数字：

HSC 0HSC 1HSC 2HSC 3

变量：HSC00

参数	类型	值	缺省值	单位
HSC				
HSC00				
模式	Enum...	Main	未使用	
参数				
模式	Enum...	一次性	一次性	
预设/模数	DINT...	2147483647	2147483647	
时间	Enum...	1	1	s
时钟输入				
输入模式	Enum...	单相	单相	
A 过滤器	Enum...	0.004	0.004	ms
B 过滤器	Enum...	0.004	0.004	ms
辅助输入				
SYNC	Enum...	已启用	已禁用	
SYNC 过滤器	Enum...	0.004	0.004	ms
SYNC 跳变沿	Enum...	上升沿	上升沿	
EN	Enum...	已启用	已禁用	
EN 过滤器	Enum...	0.04	0.04	ms
CAP	Enum...	已启用	已禁用	
CAP 过滤器	Enum...	0.04	0.04	ms
CAP 跳变沿	Enum...	上升沿	上升沿	
阈值				
阈值 0	Enum...	已启用	已禁用	
阈值 0 值	DINT...	0	0	
阈值 0 事件	Enum...	否	否	
阈值 1	Enum...	已启用	已禁用	
阈值 1 值	Enum...	1000000000	1000000000	
阈值 1 事件	Enum...	否	否	
反射输出				
反射输出 0	Enum...	是	否	
反射输出 0	Enum...	否	否	
...				
反射输出 1	Enum...	是	否	
反射输出 1	Enum...	否	否	
...				

5

4

6

7

8

9

10

配置过程

按照以下过程配置一次性模式下的 Main 类型：

步骤	操作
1	进入配置窗口
2	双击控制器
3	选择内嵌功能 →HSC 结果：HSC Main 配置窗口随即打开。
4	通过选择模式 →Main，从下拉菜单将类型设置为 Main。
5	随即创建 Main 类型的实例，您可以通过变量字段重新命名该实例。
6	通过选择参数 →模式 →一次性，从下拉菜单将模式设置为一次性
7	从参数 →预设 / 模数设置预设值
8	通过选择时钟输入 →A 过滤器，从下拉菜单设置抗跳动过滤值
9	还可以从下拉菜单启用 SYNC、EN 和 CAP 辅助输入，具体方法是选择辅助输入 →SYNC、EN 或 CAP，以通过物理输入启用同步功能 (参见第 126 页)、启用功能 (参见第 127 页) 和捕捉功能 (参见第 121 页)。
10	或者从下拉菜单启用阈值，具体方法是选择阈值 →阈值 0 →启用 / 禁用，以准许比较功能并配置反射输出 (参见第 115 页)

IO 摘要

输入 / 输出配置显示在 IO 摘要窗口中，可通过 IO 摘要按钮打开：

IO 摘要

输入

通道	配置
I0	HSC 0 - A 过滤器
I1	HSC 0 - EN
I2	HSC 2 - A 过滤器
I3	HSC 2 - EN
I4	HSC 0 - CAP
I5	HSC 0 - SYNC
I6	HSC 2 - CAP
I7	HSC 2 - SYNC
I8	
I9	
I10	
I11	
I12	
I13	

输出

通道	配置
Q0	HSC 0 - 反射输出 0
Q1	HSC 0 - 反射输出 1
Q2	HSC 2 - 反射输出 0
Q3	HSC 2 - 反射输出 1
Q4	
Q5	
Q6	
Q7	
Q8	
Q9	

OK

请参阅硬件指南以获取接线详细信息。(参见 *Modicon M218 Logic Controller, 硬件指南*)

可编程过滤器

Main 类型输入的过滤值确定计数器最大频率，如下表所示：

输入	过滤器值	最大计数器频率
A	0.004 毫秒	100 kHz
	0.4 毫秒	1 kHz
	1.2 毫秒	350 Hz
	4 毫秒	100 Hz

Main 类型编程

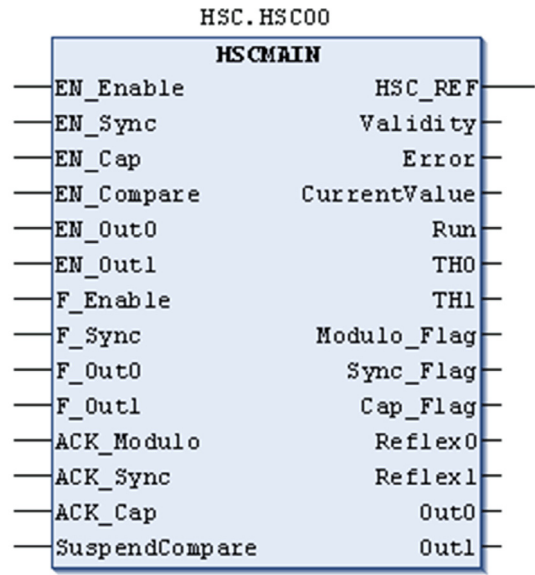
概述

Main 类型始终由 HSCMain 功能块进行管理。

注意：如果使用 HSCMain 功能块管理其他 HSC 类型，则在编译时显示错误代码。

添加 HSCMain 功能块

步骤	说明
1	使用插入运算块助手插入 HSCMain 功能块。可在以下路径找到此功能块：功能块（库）→M218_HSC →计数器 →HSCMain
2	输入 Main 类型实例的名称（在配置中定义，步骤 5），或通过单击以下内容查找功能块实例：  使用输入助手，可以在以下路径选择 HSC 实例：全局变量 →<MyController> →内嵌功能 →HSC



I/O 变量用法

下表介绍了功能块的不同引脚在一次性模式中的使用方法。

下表介绍了输入变量：

输入	类型	描述
EN_Enable	BOOL	配置 EN 输入后：如果为 TRUE ，则准许通过启用输入 (参见第 127 页) 启用计数器。
EN_Sync	BOOL	配置 SYNC 输入后：如果为 TRUE ，则准许通过同步输入 (参见第 126 页) 进行计数器同步并启动。
EN_Cap	BOOL	配置 CAP 输入后：如果为 TRUE ，则启用 捕捉输入 (参见第 121 页)。
EN_Compare	BOOL	TRUE = 启用比较器操作 (参见第 115 页) (使用阈值 0、1)： ● 基本比较 (TH0 、 TH1 输出位) ● 反射 (Reflex0 、 Reflex1 输出位) ● 事件 (在超出阈值时触发外部任务)
EN_Out0	BOOL	TRUE = 启用物理输出 Output0 回显 Reflex0 值 (如果已配置)。
EN_Out1	BOOL	TRUE = 启用物理输出 Output1 回显 Reflex1 值 (如果已配置)。
F_Enable	BOOL	强制实施启用条件 (参见第 127 页)。
F_Sync	BOOL	强制实施同步条件 (参见第 126 页)
F_Out0	BOOL	TRUE = 强制 Output0 为 1 (如果已配置 Reflex0)。
F_Out1	BOOL	TRUE = 强制 Output1 为 1 (如果已配置 Reflex1)。
ACK_Modulo	BOOL	未使用
ACK_Sync	BOOL	在上升沿，复位 Sync_Flag 。
ACK_Cap	BOOL	在上升沿，复位 Cap_Flag 。
SuspendCompare	BOOL	TRUE = 比较结果已挂起： ● 块的 TH0 、 TH1 、 Reflex0 、 Reflex1 、 Out0 、 Out1 输出位保持其上一个值。 ● 物理输出 Output0 和 Output1 保持各自的上一个值。 ● 事件被隐藏。 注意： 在设置 SuspendCompare 的同时，EN_Compare 、 EN_Out0 、 EN_Out1 、 F_Out0 、 F_Out1 保持运行。

下表介绍了输出变量：

输出	类型	注释
HSC_REF	HSC_REF (参见第 138 页)	HSC 的参考。 与管理功能块的 HSC_REF_IN 输入引脚配合使用。
Validity	BOOL	TRUE = 表示功能块上的输出值有效。
Error	BOOL	TRUE = 表示检测到错误。 使用 HSCGetDiag (参见第 144 页) 功能块获得有关此检测到的错误的详细信息。
CurrentValue	DINT	计数器的当前计数值。
Run	BOOL	TRUE = 计数器正在运行。 当当前值达到 0 时切换为 0。要重新启动计数器，则需要 Sync 存在上升沿。
TH0	BOOL	当当前值 > 阈值 0 时，设为 1 (参见第 115 页)。
TH1	BOOL	当当前值 > 阈值 1 时，设为 1 (参见第 115 页)。
Modulo_Flag	BOOL	不相关
Sync_Flag	BOOL	在计数器同步 (参见第 126 页) 时，设为 1。
Cap_Flag	BOOL	有新的捕捉值存储到捕捉寄存器 (参见第 121 页) 后，设为 1。 在进行新的捕捉之前，必须先复位此标志。
Reflex0	BOOL	Reflex0 的状态。(参见第 115 页)
Reflex1	BOOL	Reflex1 的状态。(参见第 115 页)
Out0	BOOL	物理输出 Output0 的状态 (如果已配置 Reflex0)。
Out1	BOOL	物理输出 Output1 的状态 (如果已配置 Reflex1)。

调整参数

概述

下表中介绍的参数列表可使用 HSCGetParam (参见第 146 页) 或 HSCSetParam (参见第 148 页) 功能块进行读取或修改。

注意：通过程序设置的参数将覆盖在 HSC 配置窗口中配置的参数值。初始配置参数会在 冷启动或热启动 (参见 *Modicon M218 Logic Controller, 编程指南*) 后恢复。

可调整参数

下表列出的是来自 HSC_PARAMETER_TYPE (参见第 137 页) 的参数，可在程序运行过程中读取或修改这些参数：

参数	描述
HSC_PRESET	获取或设置 HSC 的预设值
HSC_THRESHOLD0	获取或设置 HSC 的阈值 0 值
HSC_THRESHOLD1	获取或设置 HSC 的阈值 1 值

模数回路模式



概述

本章介绍在**模数回路**模式下如何使用 HSC。

本部分包含了哪些内容？

本部分包括以下各章：

章	章节标题	页
5	模数回路原理	45
6	使用 Simple 类型的 模数回路	47
7	使用 Main 类型的 模数回路	55

模数回路原理

5

一次性模式原理介绍

概述

模数回路模式可用于对一系列移动对象执行的重复操作，如包装和贴标签等应用。

规则

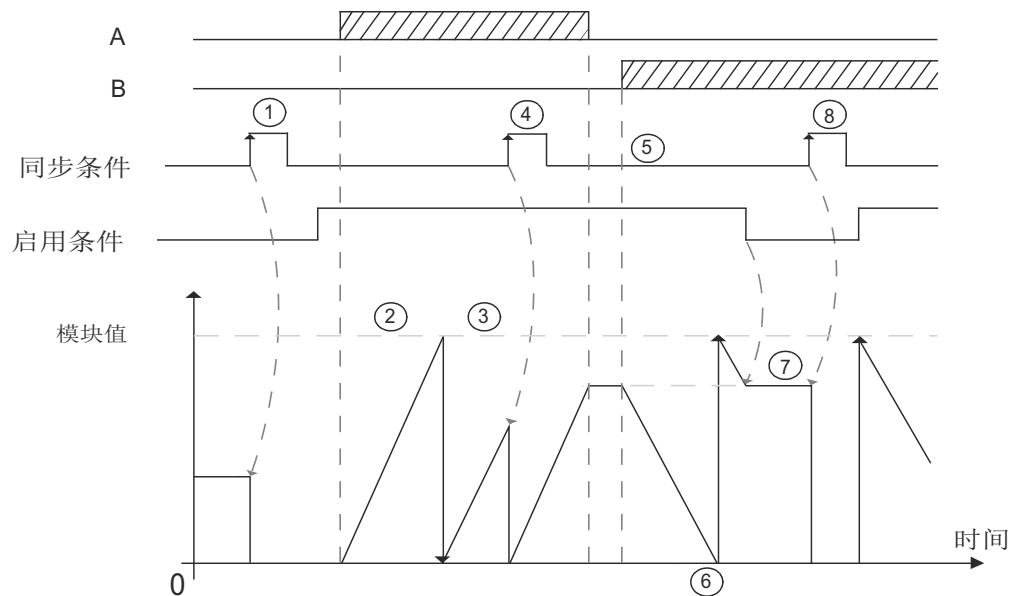
在同步条件 (参见第 126 页) 的上升沿，计数器被激活，并且当前值复位为 0。

启用计数 (参见第 127 页) 后：

递增方向：计数器在达到模数值之前一直递增。在下一个脉冲，计数器复位为 0，模数标志设置为 1 并继续计数。

递减方向：计数器在达到 0 之前一直递减。在下一个脉冲，计数器设置为模数值，模数标志设置为 1 并继续计数。

原理图



阶段	操作
1	在同步条件的上升沿，当前值复位为 0 并激活计数器。
2	当启用条件 = 1 时，A 上的每个脉冲会递增计数器值。
3	计数器达到模数值减 1 时，计数器在下一个脉冲返回 0，并继续计数。 Modulo_Flag 设置为 1。
4	在同步条件的上升沿，当前计数器值复位为 0。
5	当启用条件 = 1 时，B 上的每个脉冲会递减计数器。
6	计数器达到 0 时，计数器在下一个脉冲返回模数值减 1，并继续计数。
7	启用条件 = 0 时，忽略输入上的脉冲。
8	在同步条件的上升沿，当前计数器值复位为 0。

注意：启用和同步条件取决于配置。启用 (参见第 127 页) 和同步 (参见第 126 页) 功能中对这些条件进行了介绍。

使用 Simple 类型的模数回路



概述

本章介绍如何在使用 **Simple** 类型的**模数回路**模式下实现高速计数器。

本章包含了哪些内容？

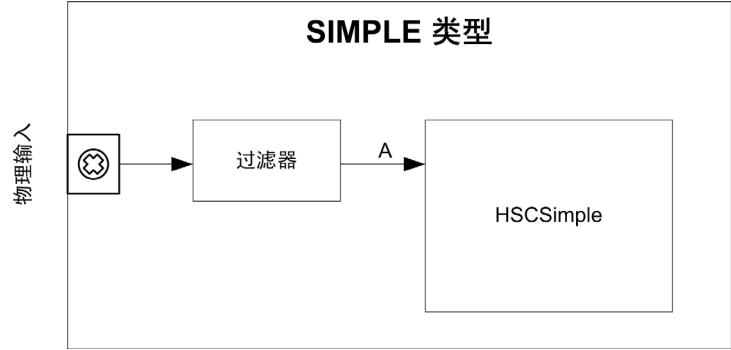
本章包含了以下主题：

主题	页
纲要图	48
模数回路 模式下的 Simple 类型配置	49
Simple 类型编程	51
调整参数	53

概要图

概要图

下图简要介绍模数回路模式下的 Simple 类型：



A 是高速计数器的计数输入。

Simple 类型只能进行加计数。

模数回路模式下的 Simple 类型配置

配置窗口

下图显示的是模数回路模式下 Simple 类型的配置窗口。

项目符号中的数字对应于配置过程表中的数字：



配置过程

按照以下过程配置模数回路模式下的 Simple 类型：

步骤	操作
1	进入配置窗口。
2	双击控制器。
3	选择内嵌功能 →HSC 结果：HSC Simple 配置窗口随即打开。
4	通过选择模式 →Simple，从下拉菜单将类型设置为 Simple。
5	随即创建 Simple 类型的实例，您可以通过变量字段重新命名该实例。
6	通过选择参数 →模式 →模数回路，从下拉菜单将模式设置为模数回路
7	从参数 →预设 / 模数设置模数值
8	通过选择时钟输入 →A 过滤器，从下拉菜单设置抗跳动过滤值

IO 摘要

输入 / 输出配置显示在 IO 摘要窗口中，可通过 **IO 摘要**按钮打开：

IO 摘要

输入

通道	配置
I0	HSC 0
I1	HSC 1
I2	HSC 2
I3	HSC 3
I4	
I5	
I6	
I7	
I8	
I9	
I10	
I11	
I12	
I13	

输出

通道	配置
Q0	
Q1	
Q2	
Q3	
Q4	
Q5	
Q6	
Q7	
Q8	
Q9	

OK

请参阅硬件指南以获取接线详细信息。(参见 *Modicon M218 Logic Controller, 硬件指南*)

可编程过滤器

Simple 类型输入的过滤值确定计数器最大频率，如下表所示：

输入	过滤器值	最大计数器频率
A	0.004 毫秒	100 kHz
	0.4 毫秒	1 kHz
	1.2 毫秒	350 Hz
	4 毫秒	100 Hz

Simple 类型编程

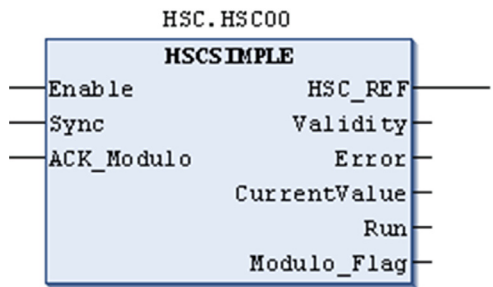
概述

Simple 类型始终由 HSCSimple (参见第 154 页) 功能块进行管理。

注意： 如果使用 HSCSimple 功能块管理其他 HSC 类型，则在编译时显示错误代码。

添加 HSCSimple 功能块

步骤	说明
1	使用插入运算块助手插入 HSCSimple 功能块。可在以下路径找到此功能块： 功能块（库）→M218_HSC →计数器 →HSCSimple
2	输入 Simple 类型实例的名称（在配置中定义，步骤 5），或通过单击以下内容查找功能块实例：  使用输入助手，可以在以下路径选择 HSC 实例： 全局变量 →<MyController> →内嵌功能 →HSC



I/O 变量用法

下表介绍了功能块的不同引脚在**模数回路**模式中的使用方法。

下表介绍了输入变量：

输入	类型	注释
Enable	BOOL	TRUE = 准许对当前计数器值进行更改。
Sync	BOOL	在上升沿预设和启动计数器。
ACK_Modulo	BOOL	在上升沿，复位 Modulo_Flag 。

下表介绍了输出变量：

输出	类型	注释
HSC_REF	HSC_REF (参见第 138 页)	HSC 的参考。 与管理功能块的 HSC_REF_IN 输入引脚配合使用。
Validity	BOOL	TRUE = 表示功能块上的输出值有效。
Error	BOOL	TRUE = 表示检测到错误。 HSCGetDiag (参见第 144 页) 功能块可用于获取有关此检测到的错误的更多信息。
Run	BOOL	TRUE = 表示计数器正在运行。
Modulo_Flag	BOOL	计数器回转模数时，设为 1。
CurrentValue	DWORD	计数器的当前计数值。

调整参数

概述

下表中介绍的参数列表可使用 HSCGetParam (参见第 146 页) 或 HSCSetParam (参见第 148 页) 功能块进行读取或修改。

注意：通过程序设置的参数将覆盖在 HSC 配置窗口中配置的参数值。初始配置参数会在 冷启动或热启动 (参见 *Modicon M218 Logic Controller, 编程指南*) 后恢复。

可调整参数

下表列出的是来自 HSC_PARAMETER_TYPE (参见第 137 页) 的参数，可在程序运行过程中读取或修改这些参数：

参数	描述
HSC_MODULO	获取或设置 HSC 的模数值

使用 Main 类型的模数回路



概述

本章介绍如何在使用 **Main** 类型的**模数回路**模式下实现高速计数器。

本章包含了哪些内容？

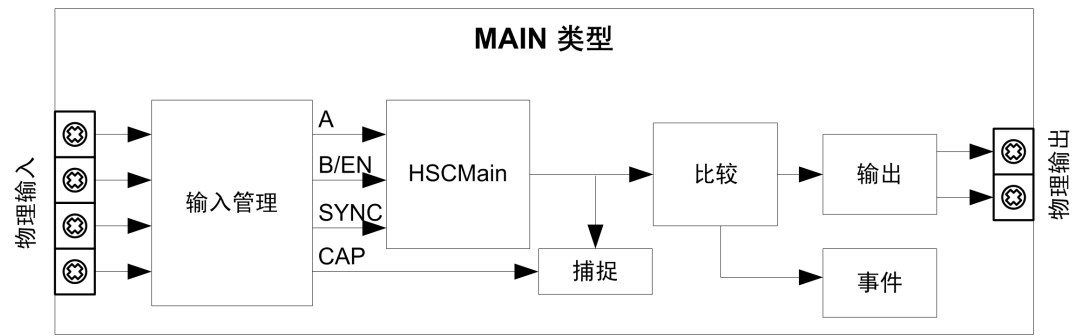
本章包含了以下主题：

主题	页
纲要图	56
模数回路 模式下的 Main 类型配置	57
Main 类型编程	60
调整参数	63

概要图

概要图

下图简要介绍模数回路模式下的 Main 类型：



A 和 B 是计数器的计数输入。

EN 是计数器的启用输入。

CAP 是计数器的捕捉输入。

SYNC 是计数器的同步输入。

可选功能

除了模数回路模式，Main 类型还可以提供以下功能：

- 比较 (参见第 115 页)
- 捕捉 (参见第 121 页)
- 通过物理输入启用 (参见第 127 页)
- 通过物理输入同步 (参见第 126 页)

注意：如果选择的计数模式同时使用输入 A 和 B，则禁用 EN 输入。

模数回路模式下的 Main 类型配置

配置窗口

下图显示的是模数回路模式下 Main 类型的配置窗口。

项目符号中的数字对应于配置过程表中的数字：

HSC 0HSC 1HSC 2HSC 3

变量: HSC00

参数	类型	值	缺省值	单位
HSC				
HSC00				
模式	Enum...	Main	未使用	
参数				
模式	Enum...	模数回路	一次性	
预设/模数	DINT ...	0	2147483647	
时间	Enum...	1	1	s
时钟输入				
输入模式	Enum...	单相	单相	
A 过滤器	Enum...	0.004	0.004	ms
B 过滤器	Enum...	0.004	0.004	ms
辅助输入				
SYNC	Enum...	已启用	已禁用	
SYNC 过滤器	Enum...	0.004	0.004	ms
SYNC 跳变沿	Enum...	上升沿	上升沿	
EN	Enum...	已启用	已禁用	
EN 过滤器	Enum...	0.04	0.04	ms
CAP	Enum...	已启用	已禁用	
CAP 过滤器	Enum...	0.04	0.04	ms
CAP 跳变沿	Enum...	上升沿	上升沿	
阈值				
阈值 0	Enum...	已启用	已禁用	
阈值 0 值	DINT ...	0	0	
阈值 0 事件	Enum...	否	否	
阈值 1	Enum...	已启用	已禁用	
阈值 1 值	DINT ...	1000000000	1000000000	
阈值 1 事件	Enum...	否	否	
反射输出				
反射输出 0	Enum...	是	否	
反射输出 0	Enum...	否	否	
反射输出 0	Enum...	否	否	
反射输出 1	Enum...	是	否	
反射输出 1	Enum...	否	否	
反射输出 1	Enum...	否	否	

5

4

6

7

8

9

10

11

配置过程

请按照以下过程配置 Main 类型：

步骤	操作
1	进入配置窗口
2	双击控制器
3	选择内嵌功能 →HSC 结果：HSC Main 配置窗口随即打开。
4	通过选择模式 →Main，从下拉菜单将类型设置为 Main。
5	随即创建 Main 类型的实例，您可以通过变量字段重新命名该实例。
6	通过选择参数 →模式 →模数回路，从下拉菜单将模式设置为模数回路
7	从参数 →预设 / 模数设置模数值
8	通过选择时钟输入 →输入模式，从下拉菜单选择输入模式值，它决定了第二个输入 B 的操作。
9	通过选择时钟输入 →A 过滤器 （在适用时也可选择 B 过滤器），从下拉菜单设置抗跳动过滤值。
10	还可以从下拉菜单启用 SYNC、EN （仅当输入模式 = 单相时）和 CAP 辅助输入，具体方法是选择辅助输入 →SYNC 或 EN 或 CAP，以通过物理输入启用同步功能 (参见第 126 页)、启用功能 (参见第 127 页) 和捕捉功能 (参见第 121 页)。
11	或者从下拉菜单启用阈值，具体方法是选择阈值 →阈值 0 →启用 / 禁用，以准许比较功能并配置反射输出 (参见第 115 页)。

IO 摘要

输入 / 输出配置显示在 IO 摘要窗口中，可通过 **IO 摘要**按钮打开：

IO 摘要

输入

通道	配置
I0	HSC 0 - A 过滤器
I1	HSC 0 - EN
I2	HSC 2 - A 过滤器
I3	HSC 2 - EN
I4	HSC 0 - CAP
I5	HSC 0 - SYNC
I6	HSC 2 - CAP
I7	HSC 2 - SYNC
I8	
I9	
I10	
I11	
I12	
I13	

输出

通道	配置
Q0	HSC 0 - 反射输出 0
Q1	HSC 0 - 反射输出 1
Q2	HSC 2 - 反射输出 0
Q3	HSC 2 - 反射输出 1
Q4	
Q5	
Q6	
Q7	
Q8	
Q9	

OK

请参阅硬件指南以获取接线详细信息。(参见 *Modicon M218 Logic Controller, 硬件指南*)

可编程过滤器

Main 类型输入的过滤值确定计数器最大频率，如下表所示：

输入	过滤器值	最大计数器频率
A, B	0.004 毫秒	100 kHz
	0.4 毫秒	1 kHz
	1.2 毫秒	350 Hz
	4 毫秒	100 Hz

Main 类型编程

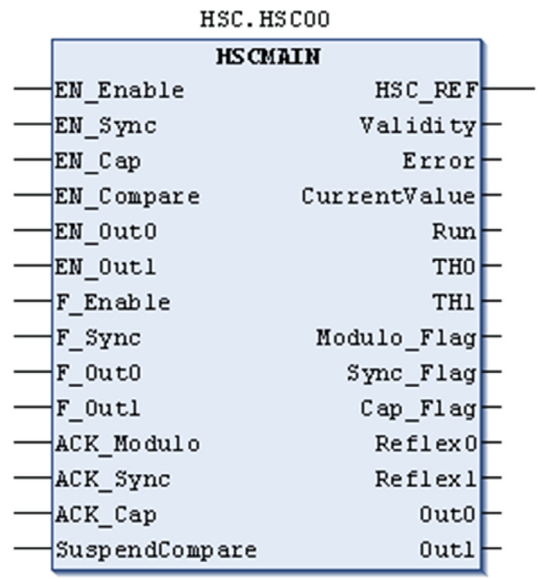
概述

Main 类型始终由 HSCMain 功能块进行管理。

注意：如果使用 HSCMain 功能块管理其他 HSC 类型，则在编译时显示错误代码。

添加 HSCMain 功能块

步骤	说明
1	使用插入运算块助手插入 HSCMain 功能块。可在以下路径找到此功能块：功能块（库）→M218_HSC →计数器 →HSCMain
2	输入 Main 类型实例的名称（在配置中定义，步骤 5），或通过单击以下内容查找功能块实例：  使用输入助手，可以在以下路径选择 HSC 实例：全局变量 →<MyController> →内嵌功能 →HSC



I/O 变量用法

下表介绍了功能块的不同引脚在模数回路模式中的使用方法。

下表介绍了输入变量：

输入	类型	描述
EN_Enable	BOOL	配置 EN 输入后：如果为 TRUE ，则准许通过启用输入 (参见第 127 页) 启用计数器。
EN_Sync	BOOL	配置 SYNC 输入后：如果为 TRUE ，则准许通过同步输入 (参见第 126 页) 进行计数器同步并启动。
EN_Cap	BOOL	配置 CAP 输入后：如果为 TRUE ，则启用 捕捉输入 (参见第 121 页)。
EN_Compare	BOOL	TRUE = 启用比较器操作 (参见第 115 页) (使用阈值 0、1)： ● 基本比较 (TH0 、 TH1 输出位) ● 反射 (Reflex0 、 Reflex1 输出位) ● 事件 (在超出阈值时触发外部任务)
EN_Out0	BOOL	TRUE = 启用物理输出 Output0 回显 Reflex0 值 (如果已配置)。
EN_Out1	BOOL	TRUE = 启用物理输出 Output1 回显 Reflex1 值 (如果已配置)。
F_Enable	BOOL	强制实施启用条件 (参见第 127 页)。
F_Sync	BOOL	强制实施同步条件 (参见第 126 页)
F_Out0	BOOL	TRUE = 强制 Output0 为 1 (如果已配置 Reflex0)。
F_Out1	BOOL	TRUE = 强制 Output1 为 1 (如果已配置 Reflex1)。
ACK_Modulo	BOOL	在上升沿，复位 Modulo_Flag 。
ACK_Sync	BOOL	在上升沿，复位 Sync_Flag 。
ACK_Cap	BOOL	在上升沿，复位 Cap_Flag 。
SuspendCompare	BOOL	TRUE = 比较结果已挂起： ● 块的 TH0 、 TH1 、 Reflex0 、 Reflex1 、 Out0 、 Out1 输出位保持其上一个值。 ● 物理输出 Output0 和 Output1 保持各自的上一个值。 ● 事件被隐藏 注意： 在设置 SuspendCompare 的同时，EN_Compare 、 EN_Out0 、 EN_Out1 、 F_Out0 、 F_Out1 保持运行。

下表介绍了输出变量：

输出	类型	注释
HSC_REF	HSC_REF (参见第 138 页)	HSC 的参考。 与管理功能块的 HSC_REF_IN 输入引脚配合使用。
Validity	BOOL	TRUE = 表示功能块上的输出值有效。
Error	BOOL	TRUE = 表示检测到错误。 使用 HSCGetDiag (参见第 144 页) 功能块获取有关检测到的错误的更多信息。
CurrentValue	DINT	计数器的当前计数值。
Run	BOOL	TRUE = 计数器正在运行。
TH0	BOOL	当当前值 > 阈值 0 时, 设为 1 (参见第 115 页)。
TH1	BOOL	当当前值 > 阈值 1 时, 设为 1 (参见第 115 页)。
Modulo_Flag	BOOL	计数器回转模数或 0 时, 设为 1。
Sync_Flag	BOOL	在计数器同步 (参见第 126 页) 时, 设为 1。
Cap_Flag	BOOL	有新的捕捉值存储到捕捉寄存器 (参见第 121 页) 后, 设为 1。 在进行新的捕捉之前, 必须先复位此标志。
Reflex0	BOOL	Reflex0 的状态。(参见第 115 页)
Reflex1	BOOL	Reflex1 的状态。(参见第 115 页)
Out0	BOOL	物理输出 Output0 的状态 (如果已配置 Reflex0)。
Out1	BOOL	物理输出 Output1 的状态 (如果已配置 Reflex1)。

调整参数

概述

下表中介绍的参数列表可使用 HSCGetParam (参见第 146 页) 或 HSCSetParam (参见第 148 页) 功能块进行读取或修改。

注意：通过程序设置的参数将覆盖在 HSC 配置窗口中配置的参数值。初始配置参数会在 冷启动或热启动 (参见 *Modicon M218 Logic Controller, 编程指南*) 后恢复。

可调整参数

下表列出的是来自 HSC_PARAMETER_TYPE (参见第 137 页) 的参数，可在程序运行过程中读取或修改这些参数：

参数	描述
HSC_MODULO	获取或设置 HSC 的模数值
HSC_THRESHOLD0	获取或设置 HSC 的阈值 0 值
HSC_THRESHOLD1	获取或设置 HSC 的阈值 1 值

自由大型模式

IV

概述

本部分介绍在**自由大型**模式下如何使用 HSC。

本部分包含了哪些内容？

本部分包括以下各章：

章	章节标题	页
8	自由大型模式原理	67
9	使用 Main 类型的 自由大型	73

自由大型模式原理介绍

概述

对于必须了解每个部件传入位置的轴监控或轴贴标情况，可以使用**自由大型**模式。

规则

在**自由大型**模式中，模块在行为上类似于标准的加减计数器。

计数启用 (参见第 127 页) 后，计数器计数方式如下：

递增方向： 计数器递增。

递减方向： 计数器递减。

对于 **Main** 类型，在同步条件 (参见第 126 页) 的上升沿上，计数器被激活，并将当前值设为预设值。

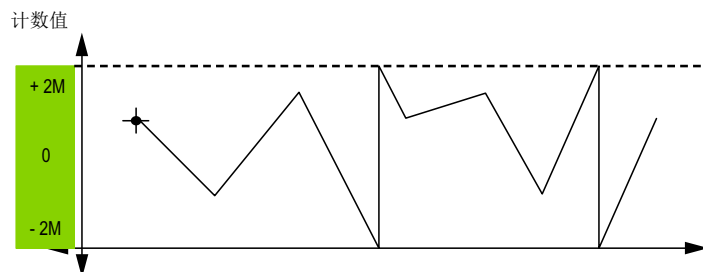
当前计数器通过捕捉 (参见第 121 页) 功能存储在捕捉寄存器。

如果计数器达到计数限制，则计数器会根据 限制管理配置做出反应。

Modulo

在计数器溢出或下溢的情况下，当前计数器值会自动变为相反限制值。

Modulo_Flag 输出设置为 1。



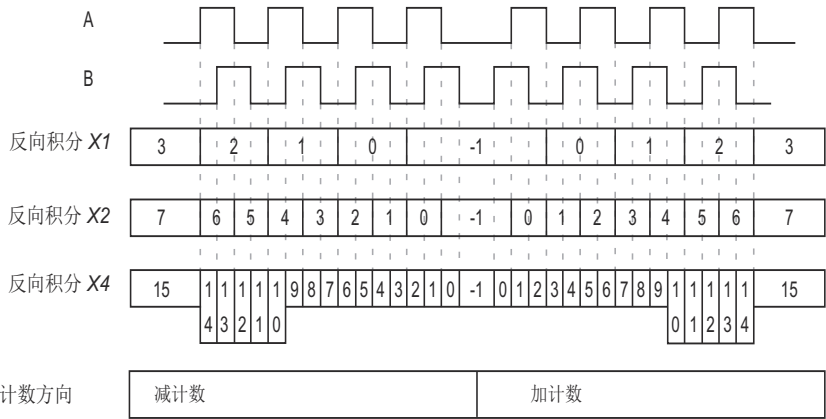
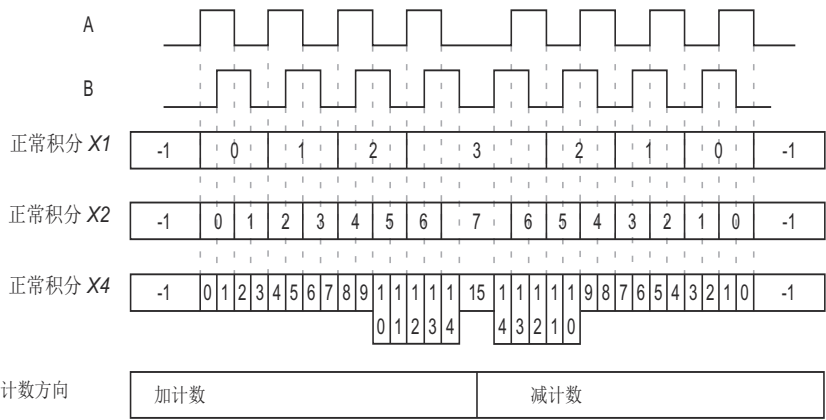
输入模式

下表介绍可用的 8 种输入模式：

输入模式	注释
A = 加计数， B = 减计数	默认模式 计数器在 A 上递增，在 B 上递减。
A = 脉冲， B = 方向	如果 A 上存在上升沿且 B 为 TRUE，则计数器递减。 如果 A 上存在上升沿且 B 为 FALSE，则计数器递增。
正常积分 X1	物理编码器始终提供 2 个信号 90° 移位，首次实现计数器可以计数脉冲并检测方向： ● X1：通过编码器循环进行 1 次计数 ● X2：通过编码器循环进行 2 次计数 ● X4：通过编码器循环进行 4 次计数
正常积分 X2	
正常积分 X4	
反向积分 X1	
反向积分 X2	
反向积分 X4	

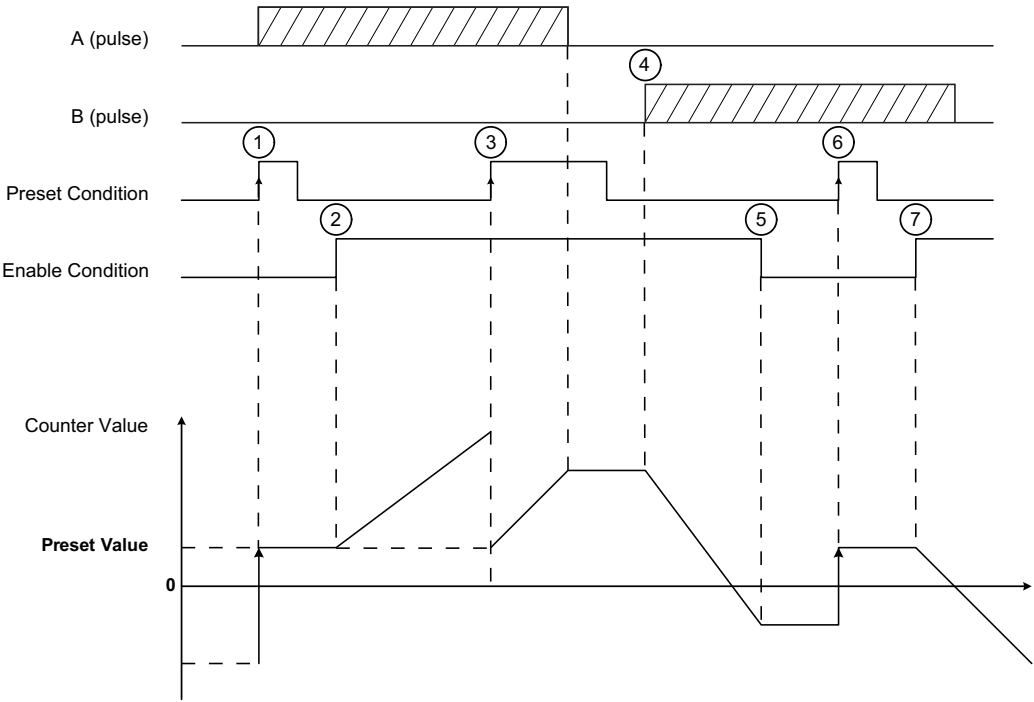
正交原理图

编码器信号根据选定的输入模式进行计数，显示如下：



原理图

下图显示 **A = 加计数**，**B = 减计数**模式：



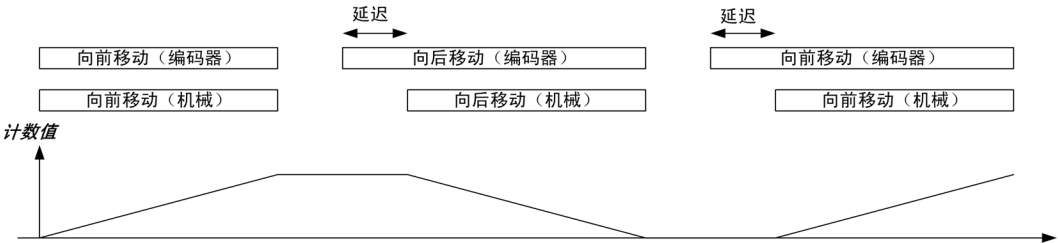
阶段	操作
1	在预设条件的上升沿上，当前值设为预设值，且计数器已激活。
2	当启用条件 = 1 时，A 上的每个脉冲会递增计数器值。
3	在预设条件的上升沿上，当前值设为预设值。
4	当启用条件 = 1 时，B 上的每个脉冲会递减计数器值。
5	当启用条件 = 0，忽略 A 或 B 上的脉冲。
6	在预设条件的上升沿上，当前值设为预设值。
7	当启用条件 = 1 时，B 上的脉冲会递减计数器值。

延迟

如果将旋转反转，计数器会应用滞后修正值。延迟值定义在旋转反转期间计数器没有确认的点数。

这样做的目的是考虑编码器 / 电机轴和机械轴（例如测量垫位置的编码器）之间的延迟。

下图对此行为进行了描述：



使用 Main 类型的自由大型



概述

本章介绍如何在使用 **Main** 类型的**自由大型**模式下实现高速计数器。

本章包含了哪些内容？

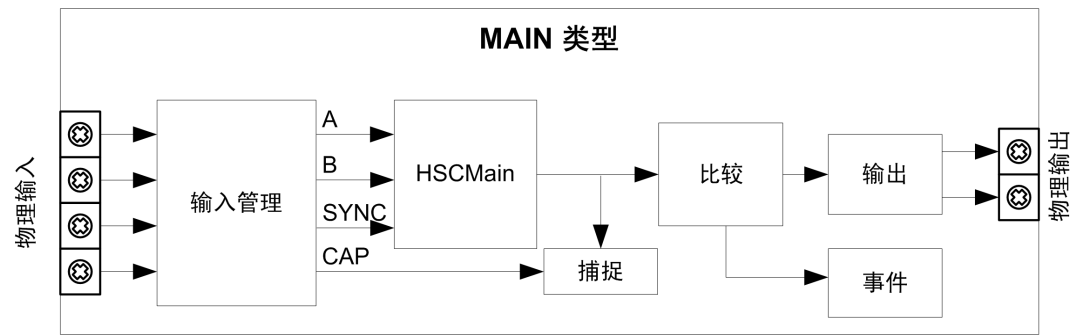
本章包含了以下主题：

主题	页
纲要图	74
自由大型 模式下 Main 类型的配置	75
Main 类型编程	78
调整参数	81

概要图

概要图

下图简要介绍自由大型模式下的 Main 类型：



A 和 B 是计数器的计数输入。

CAP 是计数器的捕捉输入。

SYNC 是计数器的同步输入。

可选功能

除了自由大型模式，Main 类型还可以提供以下功能：

- 比较 (参见第 115 页)
- 捕捉 (参见第 121 页)
- 通过物理输入同步 (参见第 126 页)

自由大型模式下 Main 类型的配置

配置窗口

下图显示的是自由大型模式下 Main 类型的配置窗口。

项目符号中的数字对应于配置过程表中的数字：

HSC 0HSC 1HSC 2HSC 3

变量：

HSC00

参数	类型	值	缺省值	单位
HSC				
HSC00				
模式	Enum...	Main	未使用	
参数				
模式	Enum...	自由大型	一次性	
预设/模数	DINT...	0	2147483647	
时间	Enum...	1	1	s
时钟输入				
输入模式	Enum...	A=向上, B=向下	A=向上, B=向下	
A 过滤器	Enum...	0.004	0.004	ms
B 过滤器	Enum...	0.004	0.004	ms
辅助输入				
SYNC	Enum...	已启用	已禁用	
SYNC 过滤器	Enum...	0.004	0.004	ms
SYNC 跳变沿	Enum...	上升沿	上升沿	
EN	Enum...	已启用	已禁用	
EN 过滤器	Enum...	0.04	0.04	ms
CAP	Enum...	已启用	已禁用	
CAP 过滤器	Enum...	0.04	0.04	ms
CAP 跳变沿	Enum...	上升沿	上升沿	
阈值				
阈值 0	Enum...	已启用	已禁用	
阈值 0 值	DINT...	0	0	
阈值 0 事件	Enum...	否	否	
阈值 1	Enum...	已启用	已禁用	
阈值 1 值	DINT...	1000000000	1000000000	
阈值 1 事件	Enum...	否	否	
反射输出				
反射输出 0	Enum...	是	否	
反射输出 0	Enum...	否	否	
反射输出 0	Enum...	否	否	
反射输出 1	Enum...	是	否	
反射输出 1	Enum...	否	否	
反射输出 1	Enum...	否	否	

5

4

6

7

8

9

10

配置过程

按照以下过程配置自由大型模式下的 Main 类型：

步骤	操作
1	进入配置窗口。
2	双击控制器。
3	选择内嵌功能 →HSC 结果：HSC Main 配置窗口随即打开。
4	通过选择模式 →Main，从下拉菜单将类型设置为 Main。
5	随即创建 Main 类型的实例，您可以通过变量字段重新命名该实例。
6	通过选择参数 →模式 →自由大型，从下拉菜单将模式设置为自由大型
7	从参数 →预设 / 模数 设置预设值
8	通过选择时钟输入 →A 过滤器 （或为相应的 B），从下拉菜单设置抗跳动过滤值。
9	还可以从下拉菜单启用 SYNC 和 CAP 辅助输入，具体方法是选择辅助输入 → SYNC 或 CAP，以通过物理输入启用同步功能 (参见第 126 页)和捕捉功能 (参见第 121 页)。
10	或者从下拉菜单启用阈值，具体方法是选择阈值 →阈值 0 →启用 / 禁用，以准许比较功能并配置反射输出 (参见第 115 页)。

IO 摘要

输入 / 输出配置显示在 IO 摘要窗口中，可通过 IO 摘要按钮打开：

IO 摘要

输入

通道	配置
I0	HSC 0 - A 过滤器
I1	HSC 0 - B 过滤器
I2	HSC 2 - A 过滤器
I3	HSC 2 - B 过滤器
I4	HSC 0 - CAP
I5	HSC 0 - SYNC
I6	HSC 2 - CAP
I7	HSC 2 - SYNC
I8	
I9	
I10	
I11	
I12	
I13	

输出

通道	配置
Q0	HSC 0 - 反射输出 0
Q1	HSC 0 - 反射输出 1
Q2	HSC 2 - 反射输出 0
Q3	HSC 2 - 反射输出 1
Q4	
Q5	
Q6	
Q7	
Q8	
Q9	

OK

请参阅硬件指南以获取接线详细信息。(参见 *Modicon M218 Logic Controller, 硬件指南*)

可编程过滤器

Main 类型输入的过滤值确定计数器最大频率，如下表所示：

输入	过滤器值	最大计数器频率
A, B	0.004 毫秒	100 kHz
	0.4 毫秒	1 kHz
	1.2 毫秒	350 Hz
	4 毫秒	100 Hz

Main 类型编程

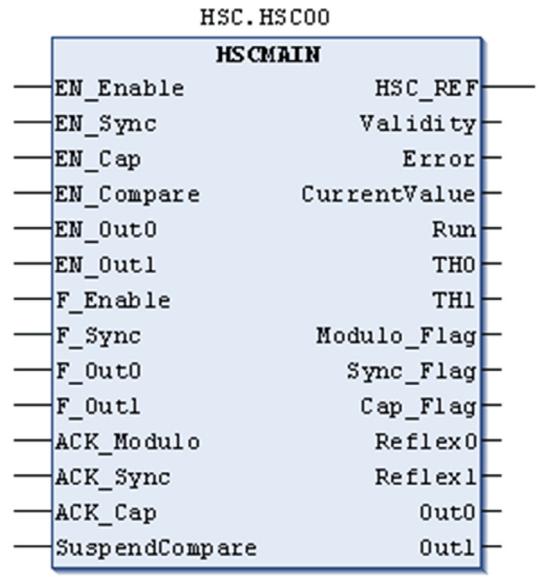
概述

Main 类型始终由 HSCMain 功能块进行管理。

注意：如果使用 HSCMain 功能块管理其他 HSC 类型，则在编译时显示错误代码。

添加 HSCMain 功能块

步骤	说明
1	使用插入运算块助手插入 HSCMain 功能块。可在以下路径找到此功能块：功能块（库）→M218_HSC →计数器 →HSCMain
2	输入 Main 类型实例的名称（在配置中定义，步骤 5），或通过单击以下内容查找功能块实例：  使用输入助手，可以在以下路径选择 HSC 实例：全局变量 →<MyController> →内嵌功能 →HSC



I/O 变量用法

下表介绍了功能块的不同引脚在自由大型模式中的使用方法。

下表介绍了输入变量：

输入	类型	描述
EN_Enable	BOOL	未使用
EN_Sync	BOOL	配置 SYNC 输入后：如果为 TRUE ，则准许通过同步输入 (参见第 126 页) 进行计数器同步并启动。
EN_Cap	BOOL	配置 CAP 输入后：如果为 TRUE ，则启用 捕捉输入 (参见第 121 页)。
EN_Compare	BOOL	TRUE = 启用比较器操作 (参见第 115 页) (使用阈值 0、1)： ● 基本比较 (TH0 、 TH1 输出位) ● 反射 (Reflex0 、 Reflex1 输出位) ● 事件 (在超出阈值时触发外部任务)
EN_Out0	BOOL	TRUE = 启用物理输出 Output0 回显 Reflex0 值 (如果已配置)。
EN_Out1	BOOL	TRUE = 启用物理输出 Output1 回显 Reflex1 值 (如果已配置)。
F_Enable	BOOL	强制实施启用条件 (参见第 127 页)。
F_Sync	BOOL	强制实施同步条件 (参见第 126 页)
F_Out0	BOOL	TRUE = 强制 Output0 为 1 (如果已配置 Reflex0)。
F_Out1	BOOL	TRUE = 强制 Output1 为 1 (如果已配置 Reflex1)。
ACK_Modulo	BOOL	在上升沿，复位 Modulo_Flag 。
ACK_Sync	BOOL	在上升沿，复位 Sync_Flag 。
ACK_Cap	BOOL	在上升沿，复位 Cap_Flag 。
SuspendCompare	BOOL	TRUE = 比较结果已挂起： ● 块的 TH0 、 TH1 、 Reflex0 、 Reflex1 、 Out0 、 Out1 输出位保持其上一个值。 ● 物理输出 Output0 和 Output1 保持各自的上一个值。 ● 事件被隐藏 注意： 在设置 SuspendCompare 的同时，EN_Compare 、 EN_Out0 、 EN_Out1 、 F_Out0 、 F_Out1 保持运行。

下表介绍了输出变量：

输出	类型	注释
HSC_REF	HSC_REF (参见第 138 页)	HSC 的参考。 与管理功能块的 HSC_REF_IN 输入引脚配合使用。
Validity	BOOL	TRUE = 表示功能块上的输出值有效。
HSC_Err	BOOL	TRUE = 表示检测到错误。 使用 HSCGetDiag (参见第 144 页) 功能块获取有关检测到的错误的更多信息。
CurrentValue	DINT	计数器的当前计数值。
Run	BOOL	TRUE = 计数器正在运行。
TH0	BOOL	当当前值 > 阈值 0 时, 设为 1 (参见第 115 页)。
TH1	BOOL	当当前值 > 阈值 1 时, 设为 1 (参见第 115 页)。
Modulo_Flag	BOOL	计数器回转其限制时, 设为 1
Sync_Flag	BOOL	在计数器同步 (参见第 126 页) 时, 设为 1
Cap_Flag	BOOL	有新的捕捉值存储到捕捉寄存器 (参见第 121 页) 后, 设为 1。 在进行新的捕捉之前, 必须先复位此标志。
Reflex0	BOOL	Reflex0 的状态。(参见第 115 页)
Reflex1	BOOL	Reflex1 的状态。(参见第 115 页)
Out0	BOOL	物理输出 Output0 的状态 (如果已配置 Reflex0)。
Out1	BOOL	物理输出 Output1 的状态 (如果已配置 Reflex1)。

调整参数

概述

下表中介绍的参数列表可使用 HSCGetParam (参见第 146 页) 或 HSCSetParam (参见第 148 页) 功能块进行读取或修改。

注意：通过程序设置的参数将覆盖在 HSC 配置窗口中配置的参数值。初始配置参数会在 冷启动或热启动 (参见 *Modicon M218 Logic Controller, 编程指南*) 后恢复。

可调整参数

下表列出的是来自 HSC_PARAMETER_TYPE (参见第 137 页) 的参数，可在程序运行过程中读取或修改这些参数：

参数	描述
HSC_PRESET	获取或设置 HSC 的预设值
HSC_THRESHOLD0	获取或设置 HSC 的阈值 0 值
HSC_THRESHOLD1	获取或设置 HSC 的阈值 1 值

事件计数 模式



概述

本部分介绍在**事件计数** 模式下如何使用 HSC。

本部分包含了哪些内容？

本部分包括以下各章：

章	章节标题	页
10	事件计数 原理	85
11	使用 Main 类型的 事件计数	87

事件计数模式原理介绍

概述

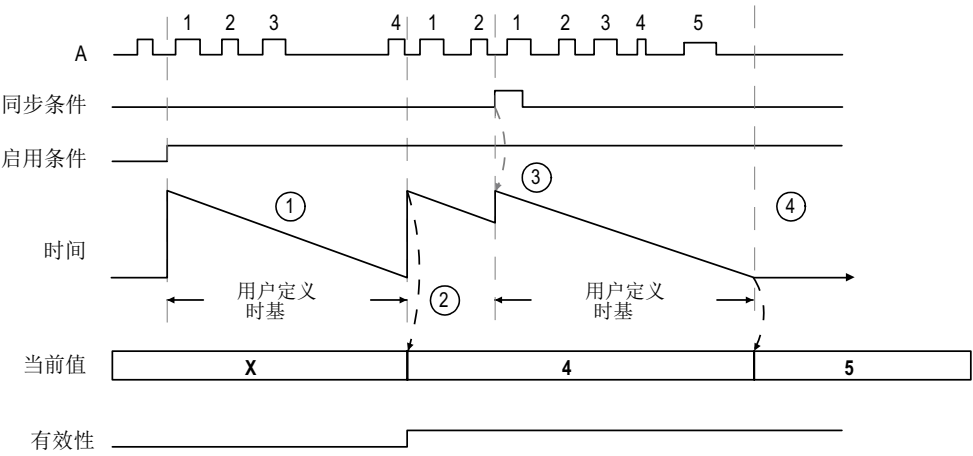
事件计数模式用于计算给定时间段内一系列事件的数量。

规则

计数器评估在预定义时间段内应用到输入的脉冲数。在每个时间段结束时用收到的事件数更新计数寄存器。

在该时间段内可以使用同步。这样将为新的预定义时间段重新启动事件计数。计数在同步条件 (参见第 126 页) 的跳变沿重新启动。

原理图



阶段	操作
1	启用条件 = 1 时，计数器对预定义时间段内物理输入上的事件（脉冲）数进行累计。 如果 Validity = 0，则当前值不相关。
2	第一个时间段过去后，计数器值设置为该时间段内算出的事件数，并且 Validity 设置为 1。 计数针对新时间段重新启动。
3	在同步条件的上升沿： ● 累计值复位为 0 ● 当前值不更新 ● 计数针对新时间段重新启动
4	该时间段过去后，计数器值设置为在该时间段内算出的事件数。 计数针对新时间段重新启动。

注意：

在 **Main** 类型上，当启用条件：

- 设置为 0：中止当前计数，并且当前值 保持为前一个有效值。
- 设置为 1：开始新计数（时基复位为 0）。

使用 Main 类型的事件计数

11

概述

本章介绍如何在使用 **Main** 类型的事件计数模式下实现高速计数器。

本章包含了哪些内容？

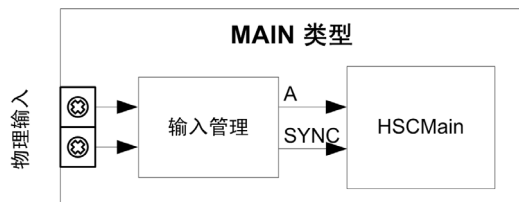
本章包含了以下主题：

主题	页
纲要图	88
事件计数模式下 Main 类型的配置	89
Main 类型编程	92
调整参数	95

概要图

概要图

下图简要介绍事件计数模式下的 **Main** 类型：



A 是计数器的计数输入。

SYNC 是计数器的同步输入。

可选功能

除了事件计数模式，**Main** 类型还可以提供以下功能：

- 通过物理输入同步 (参见第 126 页)

事件计数模式下 Main 类型的配置

配置窗口

下图显示的是事件计数模式下 Main 类型的配置窗口。
项目符号中的数字对应于配置过程表中的数字：

HSC 0HSC 1HSC 2HSC 3

变量：

HSC00

参数	类型	值	缺省值	单位
HSC				
HSC00				
模式	Enum...	Main	未使用	
参数				
模式	Enum...	事件	一次性	
预设/模数	DINT...	2147483647	2147483647	
时间	Enum...	1	1	s
时钟输入				
输入模式	Enum...	单相	单相	
A 过滤器	Enum...	0.004	0.004	ms
B 过滤器	Enum...	0.004	0.004	ms
辅助输入				
SYNC	Enum...	已启用	已禁用	
SYNC 过滤器	Enum...	0.004	0.004	ms
SYNC 跳变沿	Enum...	上升沿	上升沿	
EN	Enum...	已禁用	已禁用	
EN 过滤器	Enum...	0.04	0.04	ms
CAP	Enum...	已禁用	已禁用	
CAP 过滤器	Enum...	0.04	0.04	ms
CAP 跳变沿	Enum...	上升沿	上升沿	
阈值				
反射输出				

5

4

6

7

8

9

配置过程

按照以下过程来配置事件计数模式下的 Main 类型：

步骤	操作
1	进入配置窗口
2	双击控制器
3	选择内嵌功能 →HSC 结果：HSC Main 配置窗口随即打开。

步骤	操作
4	在 模式 （类型）字段中选择 Main 值。
5	随即创建 Main 类型的实例，您可以通过 变量 字段重新命名该实例。
6	通过选择 参数 → 模式 → 事件 ，从下拉菜单将模式设置为 事件计数
7	通过选择 参数 → 时间 ，从下拉菜单定义时间段
8	通过选择 时钟输入 → A 过滤器 ，从下拉菜单设置抗跳动过滤值
9	还可以从下拉菜单启用 SYNC 辅助输入，具体方式是选择 辅助输入 → SYNC ，以通过物理输入启用同步功能 (参见第 126 页)。

IO 摘要

输入 / 输出配置显示在 IO 摘要窗口中，可通过 **IO 摘要**按钮打开：

IO 摘要

输入

通道	配置
I0	HSC 0 - A 过滤器
I1	
I2	HSC 2 - A 过滤器
I3	
I4	
I5	HSC 0 - SYNC
I6	
I7	HSC 2 - SYNC
I8	
I9	
I10	
I11	
I12	
I13	

输出

通道	配置
Q0	
Q1	
Q2	
Q3	
Q4	
Q5	
Q6	
Q7	
Q8	
Q9	

OK

请参阅硬件指南以获取接线详细信息。(参见 *Modicon M218 Logic Controller, 硬件指南*)

可编程过滤器

Main 类型输入的过滤值确定计数器最大频率，如下表所示：

输入	过滤器值	最大计数器频率
A	0.004 毫秒	100 kHz
	0.4 毫秒	1 kHz
	1.2 毫秒	350 Hz
	4 毫秒	100 Hz

Main 类型编程

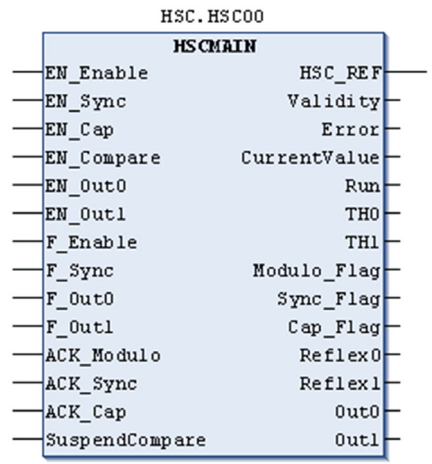
概述

Main 类型始终由 HSCMain 功能块进行管理。

注意：如果使用 HSCMain 功能块管理其他 HSC 类型，则在编译时显示错误代码。

添加 HSCMain 功能块

步骤	说明
1	使用插入运算块助手插入 HSCMain 功能块。可在以下路径找到此功能块：功能块（库）→M218_HSC →计数器 →HSCMain
2	输入 Main 类型实例的名称（在配置中定义，步骤 5），或通过单击以下内容查找功能块实例：  使用输入助手，可以在以下路径选择 HSC 实例：全局变量 →<MyController> →内嵌功能 →HSC



I/O 变量用法

下表介绍了功能块的不同引脚在事件模式中的使用方法。

下表介绍了输入变量：

输入	类型	描述
EN_Enable	BOOL	未使用
EN_Sync	BOOL	配置 SYNC 输入后：如果为 TRUE ，则准许通过同步输入 (参见第 126 页) 进行计数器同步并启动。
EN_Cap	BOOL	未使用
EN_Compare	BOOL	未使用
EN_Out0	BOOL	未使用
EN_Out1	BOOL	未使用
F_Enable	BOOL	强制实施启用条件 (参见第 127 页)。
F_Sync	BOOL	强制实施同步条件 (参见第 126 页)
F_Out0	BOOL	未使用
F_Out1	BOOL	未使用
ACK_Modulo	BOOL	未使用
ACK_Sync	BOOL	在上升沿，复位 Sync_Flag。
ACK_Cap	BOOL	未使用
SuspendCompare	BOOL	未使用

下表介绍了输出变量：

输出	类型	注释
HSC_REF	HSC_REF (参见第 138 页)	HSC 的参考。 与管理功能块的 HSC_REF_IN 输入引脚配合使用。
Validity	BOOL	TRUE = 表示功能块上的输出值有效。
Error	BOOL	TRUE = 表示检测到错误。 HSCGetDiag (参见第 144 页) 功能块可用于获取有关此检测到的错误的更多信息。
CurrentValue	DINT	计数器的当前计数值。
Run	BOOL	TRUE = 计数器正在运行。
TH0	BOOL	不相关
TH1	BOOL	不相关
Modulo_Flag	BOOL	不相关
Sync_Flag	BOOL	在计数器同步 (参见第 126 页) 时，设为 1

输出	类型	注释
Cap_Flag	BOOL	不相关
Reflex0	BOOL	不相关
Reflex1	BOOL	不相关
Out0	BOOL	不相关
Out1	BOOL	不相关

调整参数

概述

下表中介绍的参数列表可使用 HSCGetParam (参见第 146 页) 或 HSCSetParam (参见第 148 页) 功能块进行读取或修改。

注意：通过程序设置的参数将覆盖在 HSC 配置窗口中配置的参数值。初始配置参数会在 冷启动或热启动 (参见 *Modicon M218 Logic Controller, 编程指南*) 后恢复。

可调整参数

下表列出的是来自 HSC_PARAMETER_TYPE (参见第 137 页) 的参数，可在程序运行过程中读取或修改这些参数：

参数	描述
HSC_TIMEBASE	获取或设置 HSC 的时基值

频率计 模式

VI

概述

本部分介绍在**频率计** 模式下如何使用 HSC。

本部分包含了哪些内容？

本部分包括以下各章：

章	章节标题	页
12	频率计原理	101
13	使用 Main 类型的 频率计	105

频率计原理

12

频率计模式原理介绍

概述

频率计模式用于测量事件频率（以 Hz 为单位）。

概述

本章介绍如何在使用 Main 类型的频率计模式中实现高速计数器。

本章包含了哪些内容？

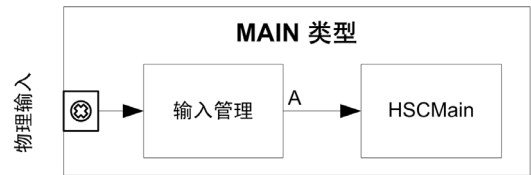
本章包含了以下主题：

主题	页
纲要图	106
频率计模式下 Main 类型的配置	107
Main 类型编程	109
调整参数	112

概要图

概要图

下图简要介绍**频率计**模式下的 Main 类型：

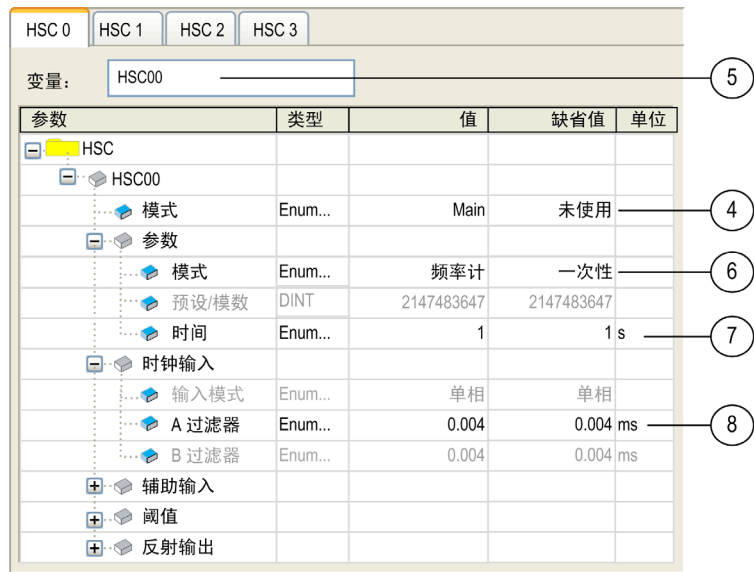


频率计用于计算预定时间段内物理输入 A 上的脉冲数量。该值存储在计数寄存器中（以 Hz 为单位）。

频率计模式下 Main 类型的配置

配置窗口

下图显示的是频率计模式下 Main 类型的配置窗口：
项目符号中的数字对应于配置过程表中的数字：



配置过程

按照以下过程来配置频率计模式下的 Main 类型：

步骤	操作
1	进入配置窗口
2	双击控制器
3	选择内嵌功能 →HSC 结果：HSC Main 配置窗口随即打开。
4	在模式（类型）字段中选择 Main 值。
5	随即创建 Main 类型的实例，您可以通过变量字段重新命名该实例。
6	通过选择参数 →模式 →频率计， 从下拉菜单将模式设置为频率计
7	从参数 →时间设置时基值
8	通过选择时钟输入 →A 过滤器， 从下拉菜单设置抗跳动过滤器值

IO 摘要

输入 / 输出配置显示在 IO 摘要窗口中，可通过 **IO 摘要**按钮打开：

IO 摘要

输入

通道	配置
I0	HSC 0 - A 过滤器
I1	
I2	HSC 2 - A 过滤器
I3	
I4	
I5	
I6	
I7	
I8	
I9	
I10	
I11	
I12	
I13	

输出

通道	配置
Q0	
Q1	
Q2	
Q3	
Q4	
Q5	
Q6	
Q7	
Q8	
Q9	

OK

请参阅硬件指南以获取接线详细信息。

可编程过滤器

Main 类型输入的过滤值确定计数器最大频率，如下表所示：

输入	过滤器值	最大计数器频率
A	0.004 毫秒	100 kHz
	0.4 毫秒	1 kHz
	1.2 毫秒	350 Hz
	4 毫秒	100 Hz

Main 类型编程

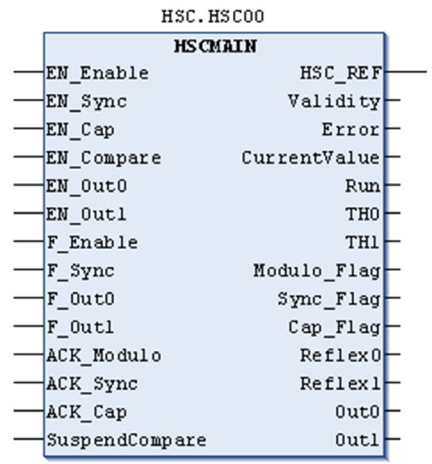
概述

Main 类型始终由 HSCMain 功能块进行管理。

注意：如果使用 HSCMain 功能块管理其他 HSC 类型，则在编译时显示错误代码。

添加 HSCMain 功能块

步骤	说明
1	使用插入运算块助手插入 HSCMain 功能块。可在以下路径找到此功能块：功能块（库）→M218_HSC →计数器 →HSCMain
2	输入 Main 类型实例的名称（在配置中定义，步骤 5），或通过单击以下内容查找功能块实例：  使用输入助手，可以在以下路径选择 HSC 实例：全局变量 →<MyController> →内嵌功能 →HSC



I/O 变量用法

下表介绍了功能块的不同引脚在**频率计**模式中的使用方法。

下表介绍了输入变量：

输入	类型	描述
EN_Enable	BOOL	未使用
EN_Sync	BOOL	未使用
EN_Cap	BOOL	未使用
EN_Compare	BOOL	未使用
EN_Out0	BOOL	未使用
EN_Out1	BOOL	未使用
F_Enable	BOOL	强制实施启用条件 (参见第 127 页)。
F_Sync	BOOL	强制实施同步条件 (参见第 126 页)
F_Out0	BOOL	未使用
F_Out1	BOOL	未使用
ACK_Modulo	BOOL	未使用
ACK_Sync	BOOL	在上升沿，复位 Sync_Flag 。
ACK_Cap	BOOL	未使用
SuspendCompare	BOOL	未使用

下表介绍了输出变量：

输出	类型	注释
HSC_REF	HSC_REF (参见第 138 页)	HSC 的参考。 与管理功能块的 HSC_REF_IN 输入引脚配合使用。
Validity	BOOL	TRUE = 表示功能块上的输出值有效。
Error	BOOL	TRUE = 表示检测到错误。 使用 HSCGetDiag (参见第 144 页) 功能块获得有关此检测到的错误的详细信息。
CurrentValue	DINT	计数器的当前计数值。
Run	BOOL	TRUE = 计数器正在运行。
TH0	BOOL	不相关
TH1	BOOL	不相关
Modulo_Flag	BOOL	不相关

输出	类型	注释
Sync_Flag	BOOL	在计数器同步 时， 设为 1
Cap_Flag	BOOL	不相关
Reflex0	BOOL	不相关
Reflex1	BOOL	不相关
Out0	BOOL	不相关
Out1	BOOL	不相关

调整参数

概述

下表介绍的参数列表可使用 HSCGetParam (参见第 146 页) 或 HSCSetParam (参见第 148 页) 功能块进行读取或修改。

注意：通过程序设置的参数将覆盖在 HSC 配置窗口中配置的参数值。初始配置参数会在 冷启动或热启动 (参见 *Modicon M218 Logic Controller, 编程指南*) 后恢复。

可调整参数

下表列出的是来自 HSC_PARAMETER_TYPE (参见第 137 页) 的参数，在该程序运行过程中可对这些参数进行修改：

参数	说明
HSC_TIMEBASE	获取或设置 HSC 的时间值

可选功能

VII

概述

本部分提供有关 HSC 可选功能的信息。

本部分包含了哪些内容？

本部分包括以下各章：

章	章节标题	页
14	比较功能	115
15	捕捉功能	121
16	同步、和启用功能	125

14.1 使用 Main 类型的比较

概述

本节提供有关使用 **Main** 类型的比较功能的信息。

本节包含了哪些内容？

本节包含了以下主题：

主题	页
使用 Main 类型的比较原理	116
对使用 Main 类型的比较进行配置	119
外部事件配置	120

使用 Main 类型的比较原理

概述

使用 **Main** 类型的比较块用于管理以下模式中的阈值、反射输出和事件：

- One-shot (参见第 25 页)
- Modulo-loop (参见第 43 页)
- Free-Large (参见第 65 页)

激活至少一个阈值后，在配置屏幕 (参见第 119 页) 中对比较进行配置。

比较可以用来触发：

- 对关于阈值的操作进行编程 (参见第 117 页)
- 与外部任务相关联的阈值事件 (参见第 117 页)
- 反射输出 (参见第 117 页)

比较原理

Main 类型或能够管理最多 2 个阈值。

相对于当前的计数值而言，阈值是一个配置值。阈值用来定义区域（至多 3 到个），或对越过的值作出反应。

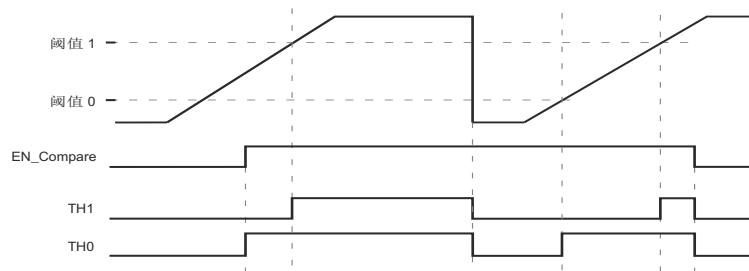
阈值通过配置来定义，也可通过使用 `HSCSetParam` (参见第 146 页) 功能块在应用程序中进行调整。

如果阈值 x ($x = 0, 1$) 已配置，并且已启用比较 (`EN_Compare = 1`)，则 `HSCMain` 功能块的输出引脚 `THx` 发生以下情况：

- 在计数器值 $\geq$ 阈值 x 时设置
- 在计数器值 $<$ 阈值 x 时复位

注意：如果 `HSCMain` 功能块上的 `EN_Compare` 设为 0，则禁用比较功能，包括由阈值事件和反射输出触发的外部任务。

2 个阈值示例：



阈值行为

对于时间常数较低的应用程序来说，使用任务环境中（功能块的 TH0 到 TH1 输出引脚）的可用阈值比较状态是一个理想选择。

例如，可以用它监控水箱中的液体水位。

配置事件

配置越过阈值时的事件可触发外部任务（参见第 120 页）。可以选择在向下、向上或向下 / 向上越过配置的阈值时触发事件。

反射输出行为

配置反射输出可触发物理反射输出。

这些输出在任务环境中不受控制，因此可以将反应时间降至最短。这需要快速执行的操作提供了便利。

高速计数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。

注意：反射输出的状态取决于配置。

更改阈值

在阈值比较正在进行时，请谨慎操作，避免输出或突发性事件任务执行导致的意外结果。如果禁用比较功能，则可以随意修改阈值。但是，如果启用了比较功能，在修改阈值时必须至少暂停阈值比较功能。



警告

意外的设备操作

如果 `EN_Compare = 1`，则在更改阈值时，必须使用 `SuspendCompare` 输入。

如果不遵守这些说明，将会导致死亡、严重伤害或设备损坏。

当 `EN_Compare = 1` 时，比较处于活动状态，需要遵循以下过程：

步骤	操作
1	将 <code>SuspendCompare</code> 设为 1。 在当前值下，比较被冻结： ● 块的 <code>TH0</code>、<code>TH1</code>、<code>Reflex0</code>、<code>Reflex1</code>、<code>Out0</code>、<code>Out1</code> 输出位保持其上一个值。 ● 物理输出 0、1 保持各自上一个值 ● 事件被隐藏 注意：在设置 <code>SuspendCompare</code> 的同时，<code>EN_Compare</code>、<code>EN_Out0</code>、<code>EN_Out1</code>、<code>F_Out0</code>、<code>F_Out1</code> 保持运行。
2	使用 <code>HSCSetParam</code> （参见第 146 页）功能块根据需要修改阈值。
3	将 <code>SuspendCompare</code> 设为 0。 应用新的阈值，并恢复比较。

对使用 Main 类型的比较进行配置

配置窗口

参数	类型	值	缺省值	说明
HSC				
HSC00				
类型	Enum...	Main	未使用	计数器类型
+ 参数				
+ 时钟输入				
+ 辅助输入				
- 阈值				
阈值 0	Enum...	已启用	已禁用	阈值激活
阈值 0 值	DINT...	0	0	阈值 0 值
阈值 0 事件	Enum...	否	否	触发事件
阈值 1	Enum...	已启用	已禁用	阈值激活
阈值 1 值	DINT...	1000000000	1000000000	阈值 1 值
阈值 1 事件	Enum...	否	否	触发事件
- 反射输出				
反射输出 0	Enum...	是	否	设置反射输出
反射输出 0	Enum...	否	否	设置反射输出
反射输出 0	Enum...	否	否	设置反射输出
反射输出 1	Enum...	是	否	设置反射输出
反射输出 1	Enum...	否	否	设置反射输出
反射输出 1	Enum...	否	否	设置反射输出

请按照以下步骤对使用 Main 类型的比较功能进行配置：

步骤	操作
1	进入配置窗口
2	双击控制器
3	选择：内嵌功能 →HSC
4	启用要使用的阈值。
5	提供阈值。
6	也可以选择提供事件条件 (参见第 120 页)。
7	也可以选择配置反射输出行为。(参见第 116 页)

外部事件配置

过程

下面的过程介绍如何配置外部事件 (参见 *Modicon M218 Logic Controller, 编程指南*) 以激活任务：

步骤	操作
1	左键单击任务配置节点，添加任务。
2	在 程序 窗口中，双击要与外部事件关联的任务。
3	在 类型 下拉菜单中，选择 外部 。
4	在 外部事件 下拉菜单中选择要与任务关联的事件。（请参见下表）

外部事件

下表介绍可能会关联任务的外部事件：

事件名称	描述
I0	的输入 I0 设置为 1 时，激活任务。
I1	的输入 I1 设置为 1 时，激活任务。
I2	的输入 I2 设置为 1 时，激活任务。
I3	的输入 I3 设置为 1 时，激活任务。
HSC0_TH0	HSC0 的阈值 TH0 设置为 1 时，激活任务。
HSC0_TH1	HSC0 的阈值 TH1 设置为 1 时，激活任务。
HSC2_TH0	HSC 2 的编码器的阈值 TH0 设置为 1 时，激活任务。
HSC2_TH1	HSC 2 的编码器的阈值 TH1 设置为 1 时，激活任务。

15.1 使用 Main 类型的捕捉

概述

本节提供有关使用 **Main** 类型的捕捉功能的信息。

本节包含了哪些内容？

本节包含了以下主题：

主题	页
使用 Main 类型的捕捉原理	122
对使用 Main 类型的捕捉进行配置	124

使用 Main 类型的捕捉原理

概述

捕捉功能能够根据外部输入信号存储当前的计数器值。

捕捉功能在使用以下模式的 **Main** 类型中可用：

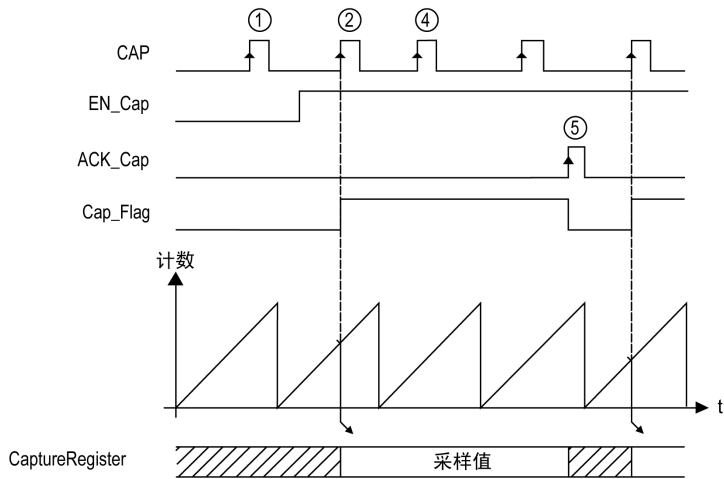
- One-shot (参见第 33 页)
- Modulo-loop (参见第 55 页)
- Free-large (参见第 73 页)

使用此功能，需要：

- 配置可选的捕捉输入：**CAP**
- 使用 HSCGetCapturedValue (参见第 142 页) 功能块检索在应用程序中捕捉到的值。

捕捉原理

下图显示了模数回路模式中的捕捉工作原理：



阶段	操作
1	当 $EN_Cap = 0$ 时，无法执行此功能。
2	当 $EN_Cap = 1$ 时， CAP 上的跳变沿会捕捉当前的计数器值并将其放入捕捉寄存器，然后触发 Cap_Flag 的上升沿。
3	使用 HSCGetCapturedValue (参见第 142 页) 获取存储的值。

阶段	操作
4	当 Cap_Flag = 1 时，忽略物理输入 CAP 上任何新的跳变沿。
5	HSCMain (参见第 150 页) 功能块输入 ACK_Cap 的上升沿会触发下降沿 Cap_Flag 输出。 准许新的捕捉。

对使用 Main 类型的捕捉进行配置

配置窗口

HSC 0HSC 1HSC 2HSC 3

变量: HSC00

参数	类型	值	预设值...	单位	说明
HSC					
HSC00					
模式	Enu...	Main	未使用		类型
参数					
时钟输入					
辅助输入					
SYNC	Enu...	已禁用	已禁用		En
SYNC 过滤器	Enu...	0.004	0.004	ms	过滤器
SYNC 跳变沿	Enu...	上升沿	上升沿		SY
EN	Enu...	已禁用	已禁用		En
EN 过滤器	Enu...	0.04	0.04	ms	过滤器
CAP	Enu...	已启用	已禁用		En.
CAP 过滤器	Enu...	0.04	0.04	ms	过滤器
CAP 跳变沿	Enu...	上升沿	上升沿		CA
阈值					
反射输出					

4

5

6

请按照以下过程对使用 Main 类型的捕捉功能进行配置：

步骤	操作
1	进入配置窗口
2	双击控制器
3	选择：内嵌功能 →HSC
4	启用 CAP 输入。
5	选择 CAP 输入的过滤值。
6	定义 CAP 输入的触发沿。

概述

本章提供有关 HSC 的同步、和启用功能的信息。

本章包含了哪些内容？

本章包含了以下主题：

主题	页
同步功能	126
启用功能	127

同步功能

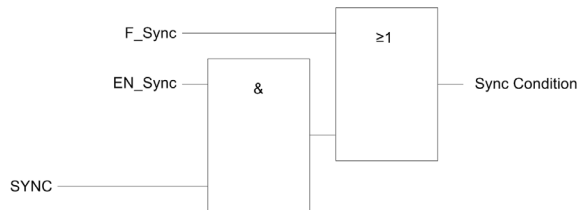
概述

同步功能用于设置 / 复位计数器操作。

说明

此功能用来根据可选 SYNC 物理输入和功能块输入 F_Sync 和 EN_Sync 的状态和配置，对计数器进行同步。

下图显示同步条件：



EN_Sync HSC 功能块的输入

F_Sync HSC 功能块的输入

SYNC 物理输入 SYNC

达到同步条件时，功能块输出 Sync_Flag 设为 1。

同步条件在上升沿上运行。

Simple 类型规格

Simple 类型的同步条件对应于功能块输入 Sync。

同步功能用于以下计数模式：

- 一次性计数器：预设和启动计数器
- 模数回路计数器：复位和启动计数器

Main 类型规格

SYNC 输入可以在配置中启用。

同步功能用于以下计数模式：

- 一次性计数器：预设和启动计数器
- 模数回路计数器：复位和启动计数器
- 自由大型计数器：预设和启动计数器
- Event counting：重新启动与时基对应的内部定时器
- 频率计：重新启动与时基对应的内部定时器

注意：在频率计模式中，同步功能只能通过功能块引脚 F_Sync 激活。

启用功能

概述

启用功能用于准许计数操作。

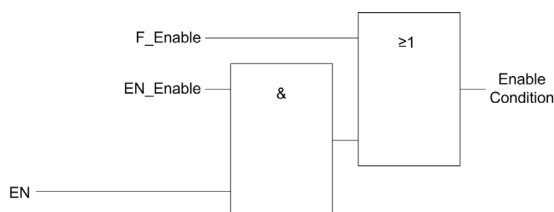
此功能可在下列计数模式中使用：

- 一次性计数器
- 模数回路计数器

描述

此功能用于准许根据可选 **EN** 物理输入以及功能块输入 **F_Enable** 和 **EN_Enable** 的状态对当前计数器值进行更改。

下图介绍了启用条件：



EN_Enable HSC 功能块的输入

F_Enable HSC 功能块的输入

EN 物理输入启用

如果没启用此功能，计数脉冲将被忽略。

注意： **Simple** 类型的启用条件与功能块输入启用 一致。

附录



概述

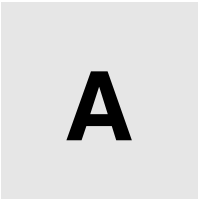
本附录概述了数据类型、功能块，并提供有关所用功能块的一般信息。

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	131
B	数据类型	135
C	功能块	141
D	功能和功能块表示形式	157

一般信息



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	132
有关管理和运动功能块管理的一般信息	133

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)。



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。

数据类型



概述

本章介绍 HSC 库的数据类型。

本章包含了哪些内容？

本章包含了以下主题：

主题	页
HSC_ERR_TYPE: HSC 的错误变量的类型	136
HSC_PARAMETER_TYPE: 在 HSC 变量上要获取或设置的参数类型	137
HSC_REF: HSC 参考值	138
HSC_TIMEBASE_TYPE: HSC 时基变量的类型	139

HSC_ERR_TYPE: HSC 的错误变量的类型

枚举类型描述

枚举数据类型 ENUM 包含检测到的错误的各种类型，这些类型带有下列值：

枚举器	值	描述
HSC_NO_ERROR	00 hex	未检测到错误
HSC_UNKNOWN	01 hex	参考 HSC 不正确或未配置。
HSC_UNKNOWN_PARAMETER	02 hex	参数参考不正确。
HSC_INVALID_PARAMETER	03 hex	参数值不正确。
HSC_COM_ERROR	04 hex	检测到 HSC 模块存在通讯错误。
HSC_CAPTURE_NOT_CONFIGURED	05 hex	未配置捕捉。

HSC_PARAMETER_TYPE: 在 HSC 变量上要获取或设置的参数类型

枚举类型描述

枚举数据类型 ENUM 包含下列值:

枚举器	值	描述
HSC_PRESET	00 hex	获取或设置 HSC 的预设值。
HSC_MODULO	01 hex	获取或设置 HSC 的模数值。
HSC_TIMEBASE	02 hex	获取或设置 HSC 的时基值 (参见第 139 页)。
HSC_SLACK	03 hex	获取或设置 HSC 的延迟值。
HSC_CALIBRATION	04 hex	获取或设置 HSC 的校准值 (仅适用于 TM200HSC206DT/F 模块)。
HSC_THRESHOLD0	05 hex	获取或设置 HSC 的阈值 0 值。
HSC_THRESHOLD1	06 hex	获取或设置 HSC 的阈值 1 值。
HSC_CURRENT	07 hex	获取或设置 HSC 的当前计数值。

HSC_REF: HSC 参考值

数据类型描述

HSC_REF 是用于标识与管理功能块相关的 HSC 功能的字节。

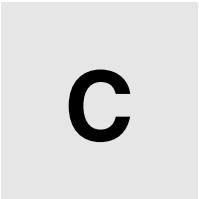
HSC_TIMEBASE_TYPE: HSC 时基变量的类型

枚举类型描述

枚举数据类型 ENUM 包含可用于 HSC 功能块的各种时基值:

名称	值
HSC_100ms	00 hex
HSC_1s	01 hex
HSC_10s	02 hex
HSC_60s	03 hex

功能块



概述

本章介绍 HSC 库的功能和功能块。

本章包含了哪些内容？

本章包含了以下主题：

主题	页
HSCGetCapturedValue: 返回捕捉寄存器的内容	142
HSCGetDiag: 提供 HSC 上检测到的错误的详细信息	144
HSCGetParam: 返回 HSC 的参数	146
HSCSetParam: 调整以下参数 HSC	148
HSCMain: HSC Main 功能块	150
HSCSimple: HSC Simple 功能块	154

HSCGetCapturedValue：返回捕捉寄存器的内容

功能描述

此管理功能块返回捕捉寄存器的内容。

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 157 页）一章。

I/O 变量介绍

下表介绍输入变量：

输入	类型	注释
HSC_REF_IN	HSC_REF（参见第 138 页）	HSC 的参考。 在块执行期间，不得进行更改。
Execute	BOOL	在上升沿上，启动功能块执行。 在下降沿上，功能块执行终止后，将其输出复位。
CaptureNumber	BYTE	捕捉寄存器的索引： - 对于 Main type counter：始终为 0 - 对于专用类型计数器：0 或 1

下表介绍了输出变量：

输出	类型	注释
HSC_REF_OUT	HSC_REF（参见第 138 页）	HSC 的参考。
Done	BOOL	TRUE = 表示 CaptureValue 有效。 功能块执行完成。
Busy	BOOL	TRUE = 表示功能块执行正在进行。

输出	类型	注释
Error	BOOL	TRUE = 表示检测到错误。 功能块执行完成。
ErrID	HSC_ERR_TYPE (参见第 136 页)	当 Error 为 TRUE 时：检测到的错误的类型
CaptureValue	DINT	当 Done 为 TRUE 时：捕捉寄存器值有效。

注意：如果检测到错误，则变量采用上次捕捉的值。

注意：有关 Done 、 Busy 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 133 页)。

添加 HSCGetCapturedValue 功能块

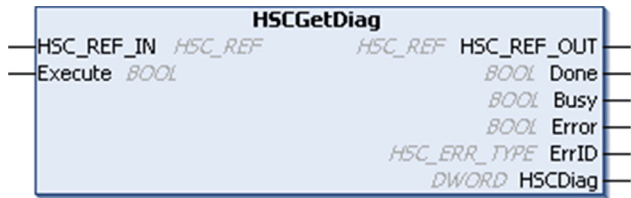
步骤	描述
1	使用插入运算块助手插入 HSCGetCapturedValue 功能块。可在以下路径找到此功能块： 功能块（库）→SEC_HSC→管理
2	将 HSC_REF_IN 输入与 HSC 的 HSC_REF 输出关联起来。

HSCGetDiag: 提供 HSC 上检测到的错误的详细信息

功能描述

此管理功能块返回检测到的 HSC 错误的详细信息。

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 157 页）一章。

I/O 变量介绍

下表介绍了输入变量：

输入	类型	注释
HSC_REF_IN	HSC_REF（参见第 138 页）	HSC 的参考。 在块执行期间，不得进行更改。
Execute	BOOL	在上升沿上，启动功能块执行。 在下降沿上，功能块执行终止后，将其输出复位。

下表介绍了输出变量：

输出	类型	注释
HSC_REF_OUT	HSC_REF（参见第 138 页）	HSC 的参考。
Done	BOOL	TRUE = 表示 HSCDiag 有效。 功能块执行完成。
Busy	BOOL	TRUE = 表示功能块执行正在进行。
Error	BOOL	TRUE = 表示检测到错误。 功能块执行完成。

输出	类型	注释
ErrID	HSC_ERR_TYPE (参见第 136 页)	当 Error 为 TRUE 时：检测到的错误的类型
EXPERTDiag	DWORD	当 Done 为 TRUE 时：诊断值有效，请参考下表。

注意：有关 Done 、 Busy 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 133 页)。

下表指示诊断值：

位	本体 (HSCMain 或 HSCSimple)	TM200HSC206DT/F 模块 (HSCSpecialized)
0	—	检测到输入错误
1	—	检测到输出错误
2	—	—
3	—	—
4	—	自检时检测到内部错误
5	—	—
6	—	检测到总线错误
7	检测到配置错误	检测到配置错误
8	—	—
9	—	配置初始化
10	—	传感器电源电压低
11	—	执行器电源电压低
12	—	检测到 Output0 短路或过流
13	—	检测到 Output1 短路或过流
14	—	—
15	—	—

添加 HSCGetdiag 功能块

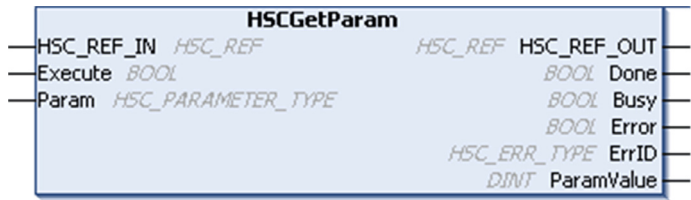
步骤	描述
1	使用插入运算块助手插入 HSCGetDiag 功能块。可在以下路径找到此功能块： 功能块 (库)→SEC_HSC→管理
2	将 HSC_REF_IN 输入与 HSC 的 HSC_REF 输出关联起来。

HSCGetParam: 返回 HSC 的参数

功能描述

此管理功能块返回 HSC 的参数值。

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 157 页）一章。

I/O 变量介绍

下表介绍了输入变量：

输入	类型	注释
HSC_REF_IN	HSC_REF（参见第 138 页）	HSC 的参考。 在块执行期间，不得进行更改。
Execute	BOOL	在上升沿上，启动功能块执行。 在下降沿上，功能块执行终止后， 将其输出复位。
Param	HSC_PARAMETER_T YPE（参见第 137 页）	要读取的参数。

下表介绍了输出变量：

输出	类型	注释
HSC_REF_OUT	HSC_REF（参见第 138 页）	HSC 的参考。
Done	BOOL	TRUE = 表示 ParamValue 有效。 功能块执行完成。
Busy	BOOL	TRUE = 表示功能块执行正在进行。

输出	类型	注释
Error	BOOL	TRUE = 表示检测到错误。 功能块执行完成。
ErrID	HSC_ERR_TYPE (参见第 136 页)	当 Error 为 TRUE 时：检测到的错误的类型
ParamValue	DINT	已读取的参数值。

注意：有关 Done 、 Busy 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 133 页)。

添加 HSCGetParam 功能块

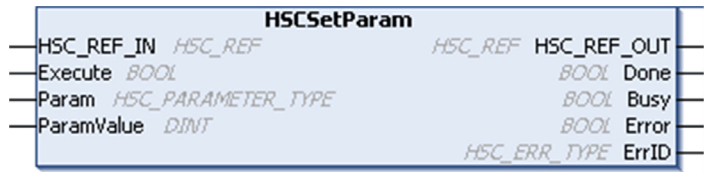
步骤	描述
1	使用插入运算块助手插入 HSCGetParam 功能块。可在以下路径找到此功能块： 功能块（库）→SEC_HSC→管理
2	将 HSC_REF_IN 输入与 HSC 的 HSC_REF 输出关联起来。

HSCSetParam: 调整以下参数 HSC

功能描述

此管理功能块修改 HSC 的参数值。

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 157 页）一章。

I/O 变量介绍

下表介绍了输入变量：

输入	类型	注释
HSC_REF_IN	HSC_REF (参见第 138 页)	HSC 的参考。 在块执行期间，不得进行更改。
Execute	BOOL	在上升沿上，启动功能块执行。 在下降沿上，功能块执行终止后，将其输出复位。
Param	HSC_PARAMETER_TYPE (参见第 137 页)	要读取的参数。
ParamValue	DINT	要写入的参数值。

下表介绍了输出变量：

输出	类型	注释
HSC_REF_OUT	HSC_REF (参见第 138 页)	HSC 的参考。
Done	BOOL	TRUE = 表示参数已成功写入。 功能块执行完成。
Busy	BOOL	TRUE = 表示功能块执行正在进行。

输出	类型	注释
Error	BOOL	TRUE = 表示检测到错误。 功能块执行完成。
ErrID	HSC_ERR_TYPE (参见第 136 页)	当 Error 为 TRUE 时：检测到的错误的类型

注意：有关 Done 、 Busy 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 133 页)。

添加 HSCSetParam 功能块

步骤	描述
1	使用插入运算块助手插入 HSCSetParam 功能块。可在以下路径找到此功能块： 功能块（库）→SEC_HSC→管理
2	将 HSC_REF_IN 输入与 HSC 的 HSC_REF 输出关联起来。

HSCMain: HSC Main 功能块

功能描述

此功能块通过下列功能控制 **Main** 类型计数器：

- 加 / 减计数
- 频率计
- 附加外部 IO
- 阈值
- 事件

使用 **Main** 计数器时，必须使用 HSC Main 功能块。

功能块实例名称必须与配置定义的名称相匹配。此功能块管理的硬件相关信息与 MAST 任务循环同步。



警告

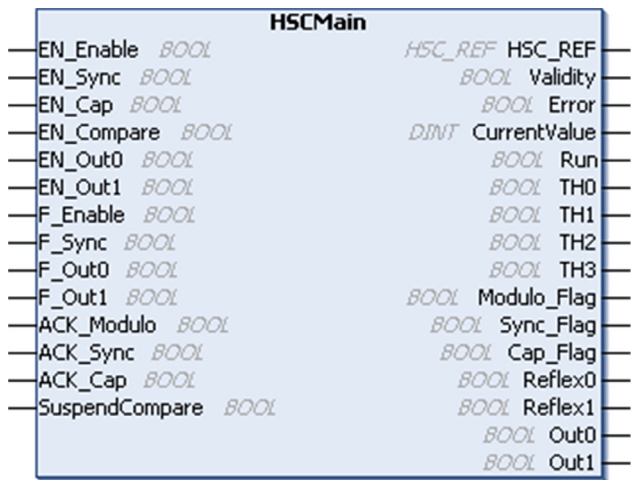
意外的输出值

- 只能在 MAST 任务中使用该功能块实例。
- 请勿在不同的任务中使用同一个功能块实例。

如果不遵守这些说明，将会导致死亡、严重伤害或设备损坏。

注意：SoMachine 允许对 FB 的逻辑输出值进行强制，但是如果此功能处于活动状态（正在执行），则不会影响硬件的相关输出。

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 157 页）一章。

I/O 变量介绍

下表介绍了输入变量：

输入	类型	描述
EN_Enable	BOOL	TRUE = 启用计数器（如果已在 一次性、模数回路 模式中进行了配置）。
EN_Sync	BOOL	TRUE = 准许通过同步输入进行计数器同步并启动。
EN_Cap	BOOL	TRUE = 启用捕捉输入（如果已在 一次性、模数回路、自由大型 模式中进行了配置）。
EN_Compare	BOOL	TRUE = 启用比较器操作（使用阈值 0、1）： ● 基本比较（TH0、TH1 输出位） ● 反射（Reflex0、Reflex1 输出位） ● 事件（在超出阈值时触发外部任务）
EN_Out0	BOOL	TRUE = 启用 Output0 回显 Reflex0 值（如果已在 一次性、模数回路、自由大型 模式中进行了配置）。
EN_Out1	BOOL	TRUE = 启用 Output1 回显 Reflex1 值（如果已在 一次性、模数回路、自由大型 模式中进行了配置）。
F_Enable	BOOL	TRUE = 准许对当前计数器值进行更改。
F_Sync	BOOL	在上升沿时，准许在下列计数模式中同步和启动计数功能： 一次性计数器： 预设和启动计数器 模数回路计数器： 复位和启动计数器 自由大型计数器： 预设和启动计数器 事件计数器： 在开始时重启内部时基 频率计： 在开始时重启内部时基
F_Out0	BOOL	TRUE = 强制 Output0 为 1（如果已在 一次性、模数回路、自由大型 模式中进行了配置）。
F_Out1	BOOL	TRUE = 强制 Output1 为 1（如果已在 一次性、模数回路、自由大型 模式中进行了配置）。
ACK_Modulo	BOOL	在上升沿时，复位 Modulo_Flag（ 模数回路和自由大型 模式）。
ACK_Sync	BOOL	在上升沿，复位 Sync_Flag。

输入	类型	描述
ACK_Cap	BOOL	在上升沿时，复位 Cap_Flag （一次性、模数回路、自由大型模式）。
SuspendCompare	BOOL	TRUE = 比较结果已挂起： - 块的 TH0 、 TH1 、 Reflex0 、 Reflex1 、 Out0 、 Out1 输出位保持其上一个值。 - 物理输出 Output0 和 Output1 保持各自的上一个值。 - 事件被隐藏。 注意：在设置 SuspendCompare 的同时，EN_Compare 、 EN_Out0 、 EN_Out1 、 F_Out0 、 F_Out1 保持运行。

下表介绍了输出变量：

输出	类型	注释
HSC_REF (参见第 138 页)	HSC_REF (参见第 138 页)	HSC 的参考。
Validity	BOOL	TRUE = 表示功能块上的输出值有效。
HSC_Err	BOOL	TRUE = 表示检测到错误。 使用 HSCGetDiag (参见第 144 页) 功能块获得有关此检测到的错误的详细信息。
CurrentValue	DINT	计数器的当前计数值。
Run	BOOL	TRUE = 计数器正在运行。 在一次性模式下，当前值 达到 0 时运行位切换为 0。
TH0	BOOL	TRUE = 当前计数器值 > 阈值 0 （如果已在一次性、模数回路、自由大型模式中进行了配置）。 只有在设置了 EN_Compare 后才处于活动状态。
TH1	BOOL	TRUE = 当前计数器值 > 阈值 1 （如果已在一次性、模数回路、自由大型模式中进行了配置）。 只有在设置了 EN_Compare 后才处于活动状态。
Modulo_Flag	BOOL	进行以下回转时，设为 1： 模数回路计数器：当计数器回转到模数或 0 时 自由大型计数器：当计数器回转其限制时

输出	类型	注释
Sync_Flag	BOOL	进行以下同步时，设为 1： ● 一次性计数器：预设和启动计数器时 ● 模数回路计数器：当计数器复位时 ● 自由大型计数器：当预设计数器时 ● 事件计数器：重新启动与时基对应的内部定时器时 ● 频率计：重新启动与时基对应的内部定时器时
Cap_Flag	BOOL	TRUE = 表示已在捕捉寄存器中锁存了一个值。 在进行新的捕捉之前，必须先复位此标志。
Reflex0	BOOL	Reflex0 的状态（如果已在 一次性、模数回路、自由大型 模式中进行了配置）。 只有在设置了 EN_Compare 后才处于活动状态。
Reflex1	BOOL	Reflex1 的状态（如果已在 一次性、模数回路、自由大型 模式中进行了配置）。 只有在设置了 EN_Compare 后才处于活动状态。
Out0	BOOL	指示 Output0 的状态
Out1	BOOL	指示 Output1 的状态

HSCSimple: HSC Simple 功能块

功能描述

此功能块通过下列简化的功能控制 **Simple** 类型计数器：

- 单向计数
- 无附加外部 IO
- 无阈值

使用 **Simple** 计数器类型时，必需使用 HSCSimple 功能块。

功能块实例名称必须与配置定义的名称相匹配。此功能块管理的硬件相关信息与 MAST 任务循环同步。



警告

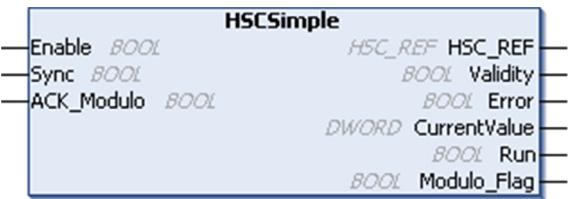
意外的输出值

- 只能在 MAST 任务中使用该功能块实例。
- 请勿在不同的任务中使用同一个功能块实例。

如果不遵守这些说明，将会导致死亡、严重伤害或设备损坏。

注意：SoMachine 允许对 FB 的逻辑输出值进行强制，但是如果此功能处于活动状态（正在执行），则不会影响硬件的相关输出。

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 157 页）一章。

I/O 变量介绍

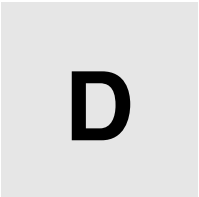
下表介绍了输入变量：

输入	类型	注释
Enable	BOOL	TRUE = 准许对当前计数器值进行更改。
Sync	BOOL	在上升沿预设和启动计数器
ACK_Modulo	BOOL	在上升沿，复位模数标志 Modulo_Flag（在模数回路模式中）。

下表介绍了输出变量：

输出	类型	注释
HSC_REF	HSC_REF（参见第 138 页）	HSC 的参考。
Validity	BOOL	TRUE = 表示功能块上的输出值有效。
HSC_Err	BOOL	TRUE = 表示检测到错误。 使用 HSCGetDiag（参见第 144 页）功能块获取有关检测到的错误的更多信息。
Run	BOOL	TRUE = 计数器正在运行。 在一次性模式中，当当前值达到 0 时切换为 0。 要重新启动计数器，则需要 Sync 存在上升沿。
Modulo_Flag	BOOL	当模数回路计数器执行以下回转时，设为 1： 当计数器回转模数时。
CurrentValue	DWORD	计数器的当前计数值。

功能和功能块表示形式

D

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	158
如何通过 IL 语言使用功能或功能块	159
如何通过 ST 语言使用功能或功能块	162

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：

```
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);
```

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

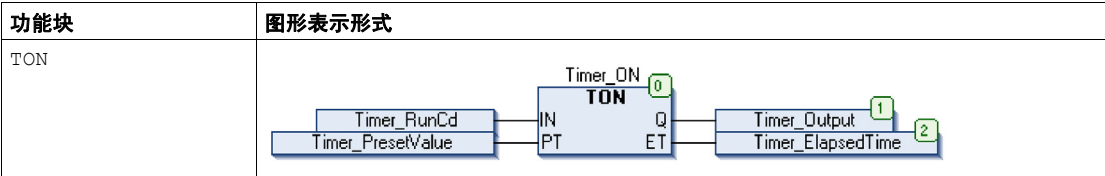
通过 IL 语言使用功能块

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：



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
TON	<pre>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 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128 129 130 131 132 133 134 135 136 137 138 139 140 141 142 143 144 145 146 147 148 149 150 151 152 153 154 155 156 157 158 159 160 161 162 163 164 165 166 167 168 169 170 171 172 173 174 175 176 177 178 179 180 181 182 183 184 185 186 187 188 189 190 191 192 193 194 195 196 197 198 199 200 201 202 203 204 205 206 207 208 209 210 211 212 213 214 215 216 217 218 219 220 221 222 223 224 225 226 227 228 229 230 231 232 233 234 235 236 237 238 239 240 241 242 243 244 245 246 247 248 249 250 251 252 253 254 255 256 257 258 259 260 261 262 263 264 265 266 267 268 269 270 271 272 273 274 275 276 277 278 279 280 281 282 283 284 285 286 287 288 289 290 291 292 293 294 295 296 297 298 299 300 301 302 303 304 305 306 307 308 309 310 311 312 313 314 315 316 317 318 319 320 321 322 323 324 325 326 327 328 329 330 331 332 333 334 335 336 337 338 339 340 341 342 343 344 345 346 347 348 349 350 351 352 353 354 355 356 357 358 359 360 361 362 363 364 365 366 367 368 369 370 371 372 373 374 375 376 377 378 379 380 381 382 383 384 385 386 387 388 389 390 391 392 393 394 395 396 397 398 399 400 401 402 403 404 405 406 407 408 409 410 411 412 413 414 415 416 417 418 419 420 421 422 423 424 425 426 427 428 429 430 431 432 433 434 435 436 437 438 439 440 441 442 443 444 445 446 447 448 449 450 451 452 453 454 455 456 457 458 459 460 461 462 463 464 465 466 467 468 469 470 471 472 473 474 475 476 477 478 479 480 481 482 483 484 485 486 487 488 489 490 491 492 493 494 495 496 497 498 499 500 501 502 503 504 505 506 507 508 509 510 511 512 513 514 515 516 517 518 519 520 521 522 523 524 525 526 527 528 529 530 531 532 533 534 535 536 537 538 539 540 541 542 543 544 545 546 547 548 549 550 551 552 553 554 555 556 557 558 559 560 561 562 563 564 565 566 567 568 569 570 571 572 573 574 575 576 577 578 579 580 581 582 583 584 585 586 587 588 589 590 591 592 593 594 595 596 597 598 599 600 601 602 603 604 605 606 607 608 609 610 611 612 613 614 615 616 617 618 619 620 621 622 623 624 625 626 627 628 629 630 631 632 633 634 635 636 637 638 639 640 641 642 643 644 645 646 647 648 649 650 651 652 653 654 655 656 657 658 659 660 661 662 663 664 665 666 667 668 669 670 671 672 673 674 675 676 677 678 679 680 681 682 683 684 685 686 687 688 689 690 691 692 693 694 695 696 697 698 699 700 701 702 703 704 705 706 707 708 709 710 711 712 713 714 715 716 717 718 719 720 721 722 723 724 725 726 727 728 729 730 731 732 733 734 735 736 737 738 739 740 741 742 743 744 745 746 747 748 749 750 751 752 753 754 755 756 757 758 759 760 761 762 763 764 765 766 767 768 769 770 771 772 773 774 775 776 777 778 779 780 781 782 783 784 785 786 787 788 789 790 791 792 793 794 795 796 797 798 799 800 801 802 803 804 805 806 807 808 809 810 811 812 813 814 815 816 817 818 819 820 821 822 823 824 825 826 827 828 829 830 831 832 833 834 835 836 837 838 839 840 841 842 843 844 845 846 847 848 849 850 851 852 853 854 855 856 857 858 859 860 861 862 863 864 865 866 867 868 869 870 871 872 873 874 875 876 877 878 879 880 881 882 883 884 885 886 887 888 889 890 891 892 893 894 895 896 897 898 899 900 901 902 903 904 905 906 907 908 909 910 911 912 913 914 915 916 917 918 919 920 921 922 923 924 925 926 927 928 929 930 931 932 933 934 935 936 937 938 939 940 941 942 943 944 945 946 947 948 949 950 951 952 953 954 955 956 957 958 959 960 961 962 963 964 965 966 967 968 969 970 971 972 973 974 975 976 977 978 979 980 981 982 983 984 985 986 987 988 989 990 991 992 993 994 995 996 997 998 999 1000</pre>

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	

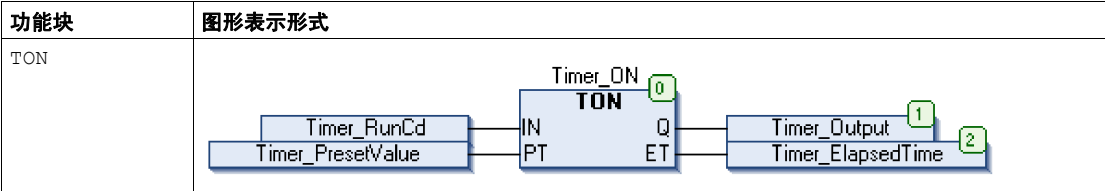
此功能的 ST 语言如下所示：
`myRTCAdjust:= SetRTCDrift(myDrift, myDay, myHour, myMinute);`

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。
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,...);

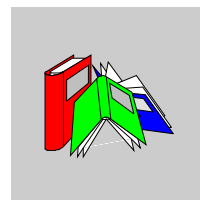
要阐释该过程，请考虑下面以图形方式表示的 TON 功能块示例：



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	<pre>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);</pre>

术语



0-9

%I

根据 IEC 标准，%I 表示输入位（例如，数字量输入类型的语言对象）。

%IW

根据 IEC 标准，%IW 表示输入字寄存器（例如，模拟量输入类型的语言对象）。

%MW

根据 IEC 标准，%MW 表示存储器字寄存器（例如，存储器字类型的语言对象）。

%Q

根据 IEC 标准，%Q 表示输出位（例如，数字量输出类型的语言对象）。

%QW

根据 IEC 标准，%QW 表示输出字寄存器（例如，模拟量输出类型的语言对象）。

1 相位计数器

*1 相位计数器*使用一路硬件输入作为计数器输入。该计数器通常在输入中存在脉冲信号时进行加减计数。

2 相位计数器

*2 相位计数器*使用两个输入计数器信号之间的相位差进行加减计数。

专用 I/O

专用 I/O 是高级特性的专用模块或通道。这些功能通常内嵌于模块中，以便不使用 PLC 控制器的资源，并提供快速响应时间（依功能而定）。就功能而言，它可以算作“独立”模块，因为功能独立于控制器处理循环，它只与控制器 CPU 交换某些信息。

主站 / 从站

在实施了主站 / 从站模型的网络中，控制方向只有一个，即从主站设备或过程到一个或多个从站设备。

以太网

*以太网*是一种用于 LAN 的物理和数据链路层技术，也称为 IEE 802.3。

任务

一组段和子程序，循环或周期性执行 MAST 任务，或周期性执行 FAST 任务。任务具有优先级，并且链接到控制器的输入和输出。这些 I/O 将随之被刷新。一个控制器可以有多个任务。

保留数据

*保留数据*值，用于下一次电源接通或热启动。即使在控制器意外关闭或正常情况下关闭控制器后，该值也仍然保留。

减载

*减载*描述运行规格的降低。对于设备而言，一般是指适当降低标称功率，以利于设备在环境条件较高（如较高的温度或较高的海拔高度）的情况下正常运行。

分配的变量

如果可以获知变量在控制器存储器中的位置，则该变量为“分配的变量”。例如，我们可以说 `water_pressure` 变量通过其与存储器位置 `%MW102.Water_pressure` 的关联进行分配。

功能

功能:

- 是返回 1 个直接结果的 POU
- 直接通过其名称（而不是通过实例）调用
- 不具备从一个调用到下一个调用的持久状态
- 可以用作表达式中的操作数

示例：布尔 (AND) 操作符、计算、转换 (BYTE_TO_INT)

功能块 (FB)

请参见 *FB*。

功能块图 (FBD)

请参见 *FBD*。

协议

*协议*是一种惯例或标准，用于控制和启用两个计算端点之间的连接、通讯和数据传输。

即时寻址

直接对编程指令中被用作操作数和参数的存储器对象（包括物理输入和输出）进行寻址，其方法是使用这些对象的直接地址（例如 `%Iwx` 或 `%QWx`）。

在程序中使用即时寻址虽然可以避免为这些对象创建符号，但也存在缺点。例如，如果通过添加或删除设备、I/O 模块或片段来更改程序配置，则用作编程指令操作数和 / 或参数的即时地址不会自己更新，而必须进行手动更正，这可能需要大量程序修改并导致不正确的编程指令。（请参见 *符号寻址*。）

反射输出

在计数模式下，高速计数器的当前值以其配置阈值为基础测得，以此确定这些专用输出的状态。

后配置

后配置文件包含与机器无关的参数，这包括：

- 机器名
- 设备名或 IP 地址
- Modbus 串行线路地址
- 路由表

周期执行

主任务是循环执行或周期性执行的。在周期模式下，您可以定义必须执行主任务的特定时间（周期）。如果执行时间短于这个时间，则在下一个循环之前将生成等待时间。如果执行时间超过这个时间，则控制系统将指示溢出。如果溢出过高，控制器将停止。

固件

*固件*表示控制器上的操作系统。

子站电缆

*子站电缆*是用于将 TAP 连接到设备的无端接支线。

定位变量

*定位变量*具有地址。（请参见*非定位变量*。）

实时时钟 (RTC)

请参见 RTC

常开

*常开*触点，是一个触点对，在执行器不活动（未通电）时打开，在执行器活动（通电）时关闭。

干线电缆

*干线电缆*是主站电缆，两个物理末端均带有线路端接电阻器。

应用程序源

*应用程序源*文件可以上载到 PC，以重新打开 SoMachine 项目。此源文件可以支持完整的 SoMachine 项目（例如，包含 HMI 应用程序的项目）。

引导应用程序

一些包含与机器相关的参数的文件：

- 机器名
- 设备名或 IP 地址
- Modbus 串行线路地址
- 路由表

循环任务

循环扫描时间具有用户指定的固定持续时间（间隔）。如果当前的扫描时间比循环扫描时间短，则控制器会等到该循环扫描时间过去之后再启动新扫描。

快速 I/O

快速 I/O 是具有某些电子特性（例如，响应时间）的特定 I/O，但对这些通道的处理由控制器 CPU 完成。

总线基板

*总线基板*是一种安装设备，用于将电子模块固定在 DIN 导轨上，并将其连接到 M258 和 LMC058 控制器的 TM5 总线。各个基板总线可扩展 TM5 数据并延伸到电源总线和 24 Vdc I/O 电源段。通过将电子模块插入基板总线可向 TM5 系统添加这些模块。基板总线还为端子块提供关节点。

托盘

*托盘*是一种便携式平板，用来存放或转移货物。

扩展 I/O 模块

扩展输入或输出模块，是将其他 I/O 添加到本体控制器的数字量或模拟量模块。

扩展总线

*扩展总线*是扩展模块和 CPU 之间的电子通讯总线。

扫描

控制器扫描程序执行 3 个基本功能：[1] 读取输入并将这些值放入存储器中；[2] 每次执行应用程序中的 1 个指令并将结果存储在存储器中；[3] 使用这些结果更新输出。

持久性数据

下一次应用程序更改或冷启动时使用的持久性数据的值。仅在重新启动控制器或复位为初始时重新初始化。需要特别指出的是，这些数据下载后它们的值保持不变。

指令列表语言 (IL)

请参见 IL。

控制器

控制器（或称为“可编程逻辑控制器”，或“可编程控制器”）用于工业流程的自动化。

控制器状态输出

*控制器状态输出*是一种特殊功能，用在位于控制器外部负责控制输出设备电源或控制器电源的电路中。

数字量 I/O

数字量输入或输出，它在电子模块上有一个独立的电路连接，与储存该 I/O 电路上的信号值的数据表位直接对应。它可以对 I/O 值进行控制逻辑数字访问。

数据日志

控制器在数据日志中记录与用户应用程序相关的事件。

最短 I/O 更新时间

*最短 I/O 更新时间*是指总线循环关闭的最短时间，以便在每次循环时强制更新 I/O。

机器

*机器*包含若干个 *功能*和 / 或 *设备*，正是这些功能或设备构成了机器。

梯形图语言

请参见 *LD*。

模拟量输入

*模拟量输入*模块包含的电路将模拟量 DC 输入信号转换为可由处理器操作的数字值。言外之意是，模拟量输入通常为直接输入。这表示数据表值将直接反映模拟量信号值。

模拟量输出

*模拟量输出*模块包含的电路将与数字值输入成比例的模拟量 DC 信号从处理器传输到模块。言外之意是，模拟量输出通常为直接输出。这表示数据表值直接控制模拟量信号值。

源极输出

源极输出，是一种接线布局，在这种布局中，输出电子模块向设备提供电流。
+24 Vdc 是源极输出的参考。

漏极输入

*漏极输入*是一种接线布局，在这种布局中，设备向输入电子模块提供电流。0 Vdc 是漏极输入的参考。

热插拔

*热插拔*是在系统保持运行的情况下用相同类型的组件进行组件更换。更换组件安装好之后，便会自动开始运行。

电子模块

在可编程控制器系统中，大多数电子模块直接与机器 / 过程的传感器、执行器和外部设备交互。此类电子模块是安装在总线基板中的组件，用于在控制器和现场设备之间提供电气连接。提供具有多种信号电平和功能的电子模块。（某些电子模块不是 I/O 接口，包括配电模块和发射器 / 接收器模块。）

电源端子

电源连接到这些端子来为控制器供电。

端子块

*端子块*是在电子模块中安装的组件，用于在控制器和现场设备之间提供电气连接。

符号

*符号*是字母数字字符（最多 32 个）组成的字符串，其中第一个字符为字母。它使您可以个性化控制器对象，以促进应用程序的可维护性。

符号寻址

间接对编程指令中被用作操作数和参数的存储器对象（包括物理输入和输出）进行寻址，其具体实现方法是首先使用与编程指令关联的符号为这些对象定义符号。

与即时寻址相比，建议使用此方法，因为如果程序配置更改，则符号会使用其新的即时地址关联自动更新，而用作操作数或参数的即时地址却不会更新。（请参见*即时寻址*。）

系统变量

系统变量结构提供控制器数据和诊断信息，并可以使用它向控制器发送命令。

系统时间

内部时钟，为设备提供系统时间。

结构化文本

以*结构化文本* (ST) 语言编写的程序，包括复杂的语句和嵌套指令（例如：迭代循环、条件执行或功能）。ST 符合 IEC 61131-3。

编码器

*编码器*是用来测量长度或角度的设备（线性或旋转编码器）。

网络

网络包含共享一个公用数据路径和通讯协议的各种互联设备。

节点

*节点*是通讯网络上的可寻址设备。

设备

*设备*是*机器*的组成部分。

输入滤波器

*输入滤波器*是消除输入噪声的特殊功能。此功能可用于消除限位开关中的输入噪声和抖动。所有输入都使用硬件提供一层输入过滤。使用软件的其他滤波器也可通过编程或者配置软件加以配置。

输入端子

*输入端子*位于扩展 I/O 模块前部，用于连接来自输入设备（如传感器、按钮和限位开关）的输入信号。对于某些模块而言，输入端子接受漏极和源极 DC 输入信号。

输出端子

*输出端子*将输出信号连接到输出设备（如机电继电器和电磁阀）。

配置

*配置*包括系统内硬件组件的布局 and 互连以及硬件和软件的选择，这些方面可决定系统的运行特性。

锁定输入

*锁定输入*模块与采用短脉冲传输消息的设备交互。捕捉和记录输入脉冲，用于应用程序以后进行检查。

闪存

*闪存*是可覆盖的非易失性存储器。它存储在一个特殊的可擦除、可重编程的EEPROM上。

阈值输出

*阈值输出*由 HSC 根据配置过程中确定的设置直接控制。

非定位变量

*非定位变量*没有地址。（请参见*定位变量*。）

顺序功能图

请参见 *SFC*。

A**AFB**

应用程序功能块

ARP

地址解析协议，它是将 IP 地址映射到 MAC（硬件）地址的以太网 IP 网络层协议。

ARRAY

ARRAY 是包含单一类型元素的表。语法如下：ARRAY [<limits>] OF <Type>

示例 1: ARRAY [1..2] OF BOOL 是由 2 个 BOOL 类型的元素组成的一维表。

示例 2: ARRAY [1..10, 1..20] OF INT 是由 10x20 个 INT 类型的元素组成的二维表。

ARW

反复位发条

ASCII

*美国信息交换标准码*是用于表示字母数字字符（字母、数字以及某些图形和控制字符）的通讯协议。

ATC

模拟张力控制

ATV

ATV 是 Altivar 驱动器的型号前缀。（例如，“ATV312”指 Altivar 312 变速驱动器。）

AWG

美国接线规格标准，规定了北美地区的接线规格。

B

BCD

二进制编码的十进制格式，利用一个 4 位组（nybble/nibble，也称为半字节）表示 0 到 9 之间的十进制数。在此格式中，用于编码十进制数字的四个位具有部分未使用的组合。例如，数字 2,450 编码为 0010 0100 0101 0000

BOOL

布尔类型，用于计算的基本数据类型。BOOL 变量可为以下值之一：0 (FALSE)，1 (TRUE)。从字中抽取的位为 BOOL 类型，例如：%MW10.4 是编号为 10 的存储器字的第五个位。

BOOTP

引导程序协议，是一种 UDP 网络协议，可由网络客户端用于从服务器自动获取 IP 地址（可能还包括其他数据）。客户端使用客户端 MAC 地址向服务器标识自己。服务器会维护预先配置的客户端设备 MAC 地址及关联 IP 地址表，从而向客户端发送其预先配置的 IP 地址。BOOTP 最初用于使无盘主机能够通过网络远程启动。BOOTP 进程分配一个无限租期的 IP 地址。BOOTP 服务使用 UDP 端口 67 和 68。

bps

每秒位数，传输速率的定义，有时也与乘数千 (kbps) 和兆 (mbps) 一起使用。

BSH

BSH 是 Schneider Electric 的 Lexium 伺服电机。

BYTE

8 位组合在一起称为一个 BYTE 。可以按二进制或八进制模式输入一个 BYTE 。
BYTE 类型以八位的格式编码，其范围为 16#00 到 16#FF （以十六进制表示）。

C

CAN

控制器局域网协议 (ISO 11898)，用于串行总线网络，旨在实现智能系统中智能设备（来自多家制造商）之间的互连，以处理实时的工业应用。CAN 多主站系统可通过实施广播消息传递和先进的诊断机制，确保高度的数据完整性。CAN 最初为汽车行业而开发，现在已应用于多种工业自动控制环境中。

CANmotion

CANmotion 是基于 CANopen 的运动总线，带有可实现 Motion Controller 和驱动器之间同步的其他机制。

CANopen

CANopen 是一种开放工业标准通讯协议和设备配置文件规范。

CFC

连续功能图（IEC61131-3 标准的扩展），是一种图形化编程语言，工作方式与流程图类似。通过添加简单的逻辑块（AND、OR 等等），即可使用此图形格式来表示程序中的每个功能或功能块。每个功能块的输入位于左侧，输出位于右侧。功能块输出可链接到其他功能块的输入，从而创建复合表达式。

CiA

CAN in automation，是一个非赢利的制造商和用户组织，致力于开发和支持基于 CAN 的高层协议。

CIP

在网络应用层实施 *公共工业协议* 后，该协议可以与其他基于 CIP 的网络进行无缝通讯，而无需考虑协议。例如，如果在以太网 TCP/IP 网络的应用层执行 CIP，可创建 EtherNet/IP 环境。同样，如果在 CAN 网络的应用层执行 CIP，可创建 DeviceNet 环境。在这种情况下，EtherNet/IP 网络上的设备可以通过 CIP 桥接器或路由器与 DeviceNet 网络上的设备进行通讯。

CMU

电流测量单位，用于将 TeSys 提供的相对电流值 (%) 转换成真实 ISO 值 (A)。

CPDM

控制器配电模块

CRC

网络消息的*循环冗余校验*字段，它包含产生校验和的少量位。此处的消息由发射器根据消息的内容进行计算。接收节点后，再次计算该字段。一旦两个 CRC 字段存在差异，则说明传输的消息与接收的消息不同。

CSA

加拿大标准协会，定义和维护危险环境中工业电子设备的标准。

CTS

清除发送，是一种数据传输信号，用于确认来自传输站的 RDS 信号。

D

DCE

数据通讯设备，介绍启动、停止和维持网络会话的设备（通常是指调制解调器）。

DHCP

动态主机配置协议，它是 BOOTP 的高级扩展。DHCP 虽然较为高级，但是 DHCP 和 BOOTP 可以通用。（DHCP 可以处理 BOOTP 客户端请求。）

DIN

Deutsches Institut für Normung，是一家制定工程和维度标准的德国机构。

DINT

双精度整数类型，以 32 位格式编码。

DNS

域名系统，是为连接 LAN 或因特网的计算机和设备进行命名的系统。

DSR

数据设置就绪，是一种数据传输信号。

DTM

设备类型管理器，能够显示 SoMachine 中的现场设备，使用它可以通过 SoMachine、控制器和现场总线与每个现场设备进行直接通讯，省却了进行独立电缆连接的麻烦。

DWORD

双字类型，以 32 位格式编码。

E**EDS**

电子数据表，包含诸如设备属性这样的信息，例如驱动器的参数和设置。

EEPROM

电可擦除可编程只读存储器，是一种非易失性存储器，用于存储切断电源时必须保存的数据。

EIA

电子工业联盟，是美国的一个贸易组织，负责制定电气 / 电子和数据通讯标准（包括 RS-232 和 RS-485）。

EIA 机架

电子工业联盟机架，是一种标准化（EIA 310-D、IEC 60297 和 DIN 41494 SC48D）系统，用于在 19 英寸（482.6 毫米）宽的栈或机架中安装各种电子模块。

EN

EN 表示由 CEN（*欧洲标准化委员会*）、CENELEC（*欧洲电工标准化委员会*）或 ETSI（*欧洲电信标准协会*）维护的许多欧洲标准中的某一个标准。

ERC

偏心滚轮传送带

ESD

静电释放

EtherNet/IP

以太网工业协议，是适用于工业系统中自动化解决方案制造的开放式通讯协议。EtherNet/IP 是在其上层执行公共工业协议的网络家族成员。支持组织 (ODVA) 指定 EtherNet/IP 是为了实现全球适应性和介质独立性。

F

FAST 任务

*FAST 任务*是持续时间较短的高优先级周期性任务，通过其编程软件在处理器上运行。此任务运行速度快，不会影响低优先级主 (MAST) 任务的执行。当需要对离散量输入的快速周期性变化进行监控时，FAST 任务就会非常有用。

FB

功能块，执行特定的自动化功能，如速度控制、间隔控制或计数。功能块由配置数据和一组操作参数组成。

FBD

功能块图，是面向图形的编程语言，与 IEC 61131-3 兼容。可用于一系列网络，其中每个网络包含一个框和连接线路的图形结构，该图形结构表示逻辑或算术表达式、功能块的调用、跳转或返回指令。

FDT

现场设备工具，用于现场设备和 SoMachine 之间的标准化通讯。

FE

功能性接地，是指必须进行接地的系统或设备上的接地点，这样有助于防止设备损坏。

FG

频率发生器

FTP

文件传输协议，是一种标准网络协议（以客户端 - 服务器架构为构建基础），用于通过基于 TCP/IP 的网络交换和操作文件。

FWD

前进

G**GVL**

全局变量列表，用于管理每个应用程序 POU 中可用的全局变量。

H**HE10**

用于频率低于 3MHz 的电子信号的矩形连接器，符合 IEC60807-2。

HMI

人机界面，是工业设备采用的一种操作员界面（通常为图形界面）。

HSC

高速计数器

HVAC

*加热通风和空气调节*应用程序，用于监控和控制室内环境。

I**I/O**

输入 / 输出

I/O 扫描

输入 / 输出扫描，持续轮询 I/O 模块，以收集数据位和状态、错误及诊断信息。这一过程用于监控输入和控制输出。

I/O 端子

输入 / 输出端子，位于扩展 I/O 模块前部，用于连接输入和输出信号。

ICMP

因特网控制消息协议，报告错误并提供与数据报处理有关的信息。

IEC

国际电工委员会，是一个非盈利性和非政府性的国际标准组织，负责为所有电器、电子和相关技术制定和发布国际标准。

IEC 61131-3

IEC 61131-3 是工业自动化设备（如控制器）采用的一种 *国际电工委员会* 标准。IEC 61131-3 针对控制器编程语言，并定义了两个图形编程语言和两个文本编程语言标准：

- **图形：**梯形图、功能块图
- **文本：**结构化文本、指令列表

IEEE

电子与电气工程师协会，是一个非盈利性的国际标准和遵从性评估组织，旨在促进电工技术的各个领域的发展。

IEEE 802.3

IEEE 802.3 是 IEEE 标准的一个集合，定义了有线以太网的物理层以及数据链路层的介质访问控制 (MAC) 子层。

IL

以 *指令列表* 语言编写的程序，包括由控制器按顺序执行的一系列指令。每个指令包括一个行号、一个指令代码和一个操作数。（IL 符合 IEC 61131-3。）

INT

单精度 *整数*，以 16 位格式编码。

IP

因特网协议，是 TCP/IP 协议系列中的一部分，用于跟踪设备的因特网地址、对传出消息进行路由并识别传入消息。

IP 20

依据 IEC 60529 制定的 *入口防护等级*，具备 IP20 防护等级的模块可防止进入或接触大于 12.5 毫米的物质。但这类模块不防水。

IP 67

依据 IEC 60529 制定的 *入口防护等级*。具备 IP67 防护等级的模块可全面防止进入和接触尘埃。即使将机体浸入水下 1 米，仍可防止污水的进入。

K**Kd**

微分增益

Ki

积分增益

Kp

比例增益

L**LAN**

局域网，是在家庭、办公室或机构环境中实施的一种短距离通讯网络。

LCD

液晶显示屏

LD

以 *梯形图* 编写的程序，它包括一个控制器程序指令图形表示，其中包含控制器按顺序执行的一系列梯级中的触点、线圈和块符号。符合 IEC 61131-3。

LED

发光二极管，是在通电时发亮的指示灯。

LINT

长整数，是 64 位变量（INT 的四倍或 DINT 的两倍）。

LMC

lexium 运动控制

LRC

纵向冗余校验

LREAL

长实型，是 64 位变量。

LSB

最低有效位（也叫最低有效字节），在传统的十六进制或二进制表示法中，它是数字、地址或字段的一部分，作为最右侧的单值写入。

LWORD

长字类型，以 64 位格式编码。

M

MAC 地址

介质访问控制地址，是与特定硬件设备关联的唯一的 48 位编号。在生产网卡或设备过程中，需要为每个网卡或设备编入一个 MAC 地址。

Magelis

Magelis 是 Schneider Electric 的 HMI 终端系列的商用名称。

MAST

主 (MAST) 任务是一种处理器任务，通过其编程软件运行。MAST 任务有两段：

- **IN:** 在 MAST 任务执行之前，将输入复制到 IN 段。
- **OUT:** 在 MAST 任务执行完后，将输出复制到 OUT 段。

MIB

管理信息库，是一种对象数据库，由类似 SNMP 的网络管理系统监控。SNMP 用于监控由设备的 MIB 所定义的设备。Schneider 已获得了一个专用 MIB：groupeschneider (3833)。

Modbus

Modbus 通信协议允许在连接到同一网络的多个设备之间进行通讯。

Modbus SL

Modbus 串行线路

MSB

最高有效位（也叫 *最高有效字节*），在传统的十六进制或二进制表示法中，是数字、地址或字段的一部分，作为最左侧的单值写入。

N**NAK**

负确认

NC

常闭 触点是当执行器处于非激活状态时（未通电）关闭，处于激活状态时（通电）开启的触点对。

NEC

*美国国家电器规程*规定电气接线和设备的安全安装。

NEMA

美国国家电气制造商协会，负责发布各种类型的电气机箱的性能标准。NEMA 标准涉及防腐蚀、防雨淋和防淹没等性能。对于 IEC 成员国家，IEC 60529 标准还对机箱的入口防护等级进行了分类。

Nibble

Nibble 是半字节（表示一个字节的 4 个位）。

NMT

网络管理协议，提供网络初始化、错误控制和设备状态控制服务。

NMT 状态机

*网络管理状态机*定义各类 CANopen 设备的通讯行为。CANopen NMT 状态机由初始化状态、预操作状态、操作状态和停止状态组成。通电或复位后，设备进入初始化状态。设备初始化完成后，设备自动进入预操作状态，并发送启动消息宣布这种状态转换。发送此消息后，即表示设备可以开始工作了。处于预操作状态的设备可能会启动并传输同步、时间标记或心跳消息。在此状态下，设备无法通过 PDO 进行通讯，而必须通过 SDO 进行通讯。在操作状态下，设备可以使用所有支持的通讯对象。

O

ODVA

开放式 deviceNet 供应商协会，旨在为以 CIP（EtherNet/IP、DeviceNet 和 CompoNet）为基础构建的系列网络技术提供支持。

OS

操作系统。可以用于由用户上传 / 下载的固件。

OSI

开放式系统互连参考模型，是一个 7 层模型，用来描述网络协议通讯。每个抽象层都从其下层接收服务，并向其上层提供服务。

OTB

优化端子块，用于 Advantys I/O 分布式模块环境中

P

PCI

外设组件互连，是用于连接外设的行业标准总线。

PDM

配电模块，向 I/O 模块群集分配 AC 或 DC 现场电源。

PDO

过程数据对象，无需确认的广播消息传输，或在基于 CAN 的网络中从生产者设备发送到消费者设备。来自生产者设备的传输 PDO 具有特定标识符，该标识符与消费者设备的接收 PDO 对应。

PDU

协议数据单元

PE

*保护性接地*是总线上的一种回路，针对控制系统中的传感器或执行器设备生成的故障电流。

PI

比例 - 积分

PID

比例 - 积分 - 微分控制

PLC

可编程逻辑控制器，是工业制造过程的“大脑”。它可以让过程自动化，而不是使用继电器控制系统。PLC 是适合在条件苛刻的工业环境中使用的计算机。

PLCopen

PLCopen 标准通过对工具、库以及模块化软件编程方法进行标准化，为自动化和控制行业带来了效率、灵活性和制造商独立性。

PLI

脉冲锁存输入

POU

程序组织单元，包括源代码变量声明和相应的指令集。POU 有助于简化软件程序、功能和功能块的模块化重用。经过声明后，POU 便可相互使用。SoMachine 编程需要使用 POU。

POU FB

程序组织单元功能块类型，是可以由用户以 ST、IL、LD 或 FBD 语言定义的用户程序。可在应用程序中使用 POU FB 类型实现以下目的：

- 简化程序的设计和输入
- 使程序更便于阅读
- 简化调试
- 减少生成的代码量

Pt100/Pt1000

Platinum 热电阻的特性取决于其在 0° C 温度时的标称电阻 R₀。

- Pt100 (R₀ = 100 欧姆)
- Pt1000 (R₀ = 1 千欧姆)

PTO

脉冲串输出，用于控制，例如，开放回路中的步进器电机。

PWM

脉冲宽度调制，用于调整脉冲信号长度的调节过程（例如，用于温度控制的执行器）。对于此类信号，要使用晶体管输出。

R

RAM

随机存取存储器

REAL

REAL 是数值数据类型。REAL 类型以 32 位格式编码。

RFID

射频识别，是一种自动识别方法，此方法的基础是使用 RFID 标签或转发器存储数据并对数据进行远程检索。

RJ-45

此*标准插座*是一种模块化连接器，通常用于通讯网络。

RPDO

接收 PDO，在基于 CAN 的网络中向设备发送数据。

RPM

每分钟转数

RPS

每秒钟转数

RS-232

RS-232（也称为 EIA RS-232C 或 V.24）是基于三条电线的串行通讯总线的标准类型。

RS-485

RS-485（也称为 EIA RS-485）是基于两条电线的串行通讯总线的标准类型。

RTC

实时时钟选件，在控制器断电后，该选件可以确保一定时间长度内的计时。

RTS

请求发送，是一种数据传输信号，由来自目标节点的 CTS 信号确认。

RTU

远程终端设备，是实际环境中的对象与分布式控制系统或 SCADA 系统之间的交互设备，用来将遥测数据传输到系统和 / 或根据从系统收到的控制消息修改所连接对象的状态。

RxD

接收数据（数据传输信号）

S**SCADA**

监控和数据采集系统，用来监控、管理和控制工业应用程序或过程。

SDO

在基于 CAN 的网络中，*服务数据对象*消息由现场总线主站用于访问（读 / 写）网络节点的对象目录。SDO 类型包括服务 SDO (SSDO) 和客户端 SDO (CSDO)。

SEL-V

安全超低电压，符合 IEC 61140 安全超低电压指令的系统将采用以下方式保护自己：任何 2 个可访问部件之间（或者 1 个可访问部件和 1 类设备的 PE 终端之间）的电压不超过正常情况或单个故障情况下的指定值。

SFC

以*顺序功能图*语言编写的程序，可用于能被拆分为数个步骤的过程。SFC 包括具有关联操作的步骤、具有相关联逻辑条件的转换，以及步骤和转换之间的定向链接。（SFC 标准在 IEC 848 中定义。符合 IEC 61131-3。）

SINT

有符号整数，是 16 位值。

SL

串行线路

SMS

短消息服务，是一种用于电话（或其他设备）的标准通讯服务，它通过移动通讯系统发送简短文本消息。

SNMP

简单网络管理协议，可以通过轮询设备状态、执行安全测试以及查看与数据传输相关的信息来远程控制网络。它还可用于远程管理软件和数据库。该协议还允许执行活动的管理任务，如修改和应用新配置。

SSI

串行同步接口，是用于相对和绝对测量系统（如编码器）的通用接口。

ST

请参见*结构化文本*。

STN

扫描扭曲向列（也称为被动矩阵）

STRING

STRING 变量是一系列 ASCII 字符。

T**TAP**

端子访问点，是连接到干线电缆的接线盒，可以插入子站电缆。

TCP

*传输控制协议*是基于连接的传输层协议，可提供可靠的同步双向数据传输。TCP 是 TCP/IP 协议套件的一部分。

TFT

薄膜传输（也称为主动矩阵）

TP

*触摸探测器*是由快速输入信号（快速传感器）触发的位置捕捉。在触摸探测器输入的上升沿，捕捉编码器的位置。示例：这用于包装机器以捕捉薄膜上的打印标记位置，从而始终在相同位置上进行切割。

TPDO

传输 PDO 在基于 CAN 的系统中从设备读取数据。

TVDA

经过测试、验证和归档的架构

TxD

TxD 表示传输信号。

U

UDINT

无符号双精度整数，以 32 位编码。

UDP

用户数据报协议，是无连接模式协议（由 IETF RFC 768 定义），在该协议下，消息在数据报（数据电报）中传递到 IP 网络上的目标计算机。UDP 协议通常与因特网协议捆绑在一起。UDP/IP 消息不要求获得响应，因此非常适合那些对于丢弃的数据包不需要重新传输（如流视频和需要实时性能的网络）的应用。

UINT

无符号整数，以 16 位编码。

UL

Underwriters Laboratories，美国的一家进行产品测试和安全认证的组织。

UTC

世界协调时间

V

VSD

变速驱动器

W

WORD

WORD 类型以 16 位格式编码。

索引



- Busy
 - 状态变量的管理, 133
- CommandAborted
 - 状态变量的管理, 133
- Done
 - 状态变量的管理, 133
- ErrID
 - 状态变量的管理, 133
 - 错误处理, 133
- Error
 - 状态变量的管理, 133
 - 错误处理, 133
- Execute
 - 状态变量的管理, 133
- HSC Main 类型配置
 - 一次性模式, 35
 - 事件模式, 89
 - 模数回路模式, 57
 - 自由大型模式, 75
 - 频率计模式, 107
- HSC Simple 类型配置
 - 一次性模式, 27
 - 模数回路模式, 49
- HSC_ERR_TYPE
 - 数据类型, 136
- HSC_PARAMETER_TYPE
 - 数据类型, 137
- HSC_REF
 - 数据类型, 138
- HSC_TIMEBASE_TYPE
 - 数据类型, 139
- HSCGetCapturedValue
 - 功能块, 142
- HSCGetDiag
 - 功能块, 144
- HSCGetParam
 - 功能块, 146
- HSCMain
 - 功能块, 150
 - 捕捉, 122
 - 比较, 116
- HSCSetParam
 - 功能块, 148
- HSCSimple
 - 功能块, 154
- 专用功能, 132
- 事件计数
 - 嵌入 HSC 的 HSC 模式, 85
- 内嵌功能配置
 - 内嵌 HSC 配置, 14
- 功能
 - 功能与功能块之间的差异, 158
 - 同步, 126
 - 启用, 127
 - 如何通过 IL 语言使用功能或功能块, 159
 - 如何通过 ST 语言使用功能或功能块, 162
- 功能块
 - HSCGetCapturedValue, 142
 - HSCGetDiag, 144
 - HSCGetParam, 146
 - HSCMain, 150
 - HSCSetParam, 148
 - HSCSimple, 154

同步

- 功能, 126

启用

- 功能, 127

嵌入 HSC 的 HSC 模式

- 事件计数, 85

- 频率计, 101

嵌入式 HSC 的 HSC 模式

- 模数回路, 45

- 自由大型, 67

捕捉

- HSCMain, 122

数据类型

- HSC_ERR_TYPE, 136

- HSC_PARAMETER_TYPE, 137

- HSC_REF, 138

- HSC_TIMEBASE_TYPE, 139

模数回路

- 嵌入式 HSC 的 HSC 模式, 45

比较

- HSCMain, 116

状态变量的管理

- Busy, 133

- CommandAborted, 133

- Done, 133

- ErrID, 133

- Error, 133

- Execute, 133

自由大型

- 嵌入式 HSC 的 HSC 模式, 67

计数器功能块

- HSCMain, 150

- HSCSimple, 154

诊断功能

- HSCGetDiag, 144

调整功能

- HSCGetParam, 146

- HSCSetParam, 148

错误处理

- ErrID, 133

- Error, 133

频率计

- 嵌入 HSC 的 HSC 模式, 101