

A man in a light blue shirt is shown from the side, holding a tablet. The background is a blurred industrial factory floor. Overlaid on the scene are various digital icons and text: a 'NEWS' box with a person icon, a '24/7' icon with a circular arrow, a 'Home' button, and the text 'Industry Online Support'. There are also icons for a folder, a network of people, and a magnifying glass. The overall color scheme is blue and white.

SIEMENS

SMART 的 基本控制库（LBC）的使用文档

STEP 7-Micro/WIN SMART

法律信息

应用实例的使用

应用示例说明了通过文本、图形和/或软件模块形式的几个组件的交互来解决自动化任务。应用示例是西门子（中国）有限公司或其子公司（“西门子”）提供的免费服务。它们不具有约束力，也不要求关于配置和设备的完整性或功能。应用程序示例仅对典型任务提供帮助；它们不构成客户特定的解决方案。您自己有责任按照适用的法规正确和安全操作产品，还必须检查相应应用示例的功能并根据您的系统进行定制。您亦应当遵循警告、安全说明以及任何其他依法使用的信息（如适用），例如通用条件、文档或操作说明。

西门子授予您非排他性的、不可再许可的和不可转让的权利，让经过技术培训的人员使用应用示例。

对应用程序示例的任何更改都由您负责。仅在与您自己的产品结合使用时，与第三方共享应用示例，或复制应用示例或摘录方被允许。该应用实例无须接受收费产品的习惯测试和质量检验；它们可能具有功能和性能缺陷以及错误，其所包含的功能未必能满足您的要求。您有责任据此设计您的使用机制并以恰当的方式使用它们，从而确保可能发生的故障均不会导致环境、财产损失或人身伤害。

免责声明

西门子不基于任何法律原因而承担任何责任，包括但不限于应用示例的可用性、完整性和无缺陷性以及相关信息、配置和性能数据及其造成的任何损害。这不适用于适用法律有强制性规定的情况，或故意、重大过失造成的人身伤害。上述规定并不意味着对您不利的举证责任的任何改变。对于第三方因您使用应用示例而提出的任何索赔，您应向西门子作出赔偿，除非西门子负有法定赔偿责任。

通过使用应用示例，您承认西门子对上述责任条款之外的任何损害不承担责任。

知识产权

应用示例及其所有权利，但不限于其中的专有权利(包括但不限于应用示例中包含的源代码、目标代码、图片、照片、动画、视频、音频、音乐、文本和小程序)、随附材料和每份副本，以及其中的所有知识产权(包括任何版权、专利、商标、商业秘密和公开权)均归西门子、其许可方或关联公司所有。除非本文件明确规定，西门子未就上述知识产权向您明示或默示授予任何权利。您同意，对于任何因您使用应用示例而引发的知识产权侵权索赔或诉讼或与之相关的任何其他损害，应由您(而非西门子)全权负责。

其他信息

西门子保留随时更改应用示例的权利，无需另行通知。如果应用实例中的建议与其他西门子出版物(如目录)之间存在差异，则应优先考虑其他文件的内容。

如您发现应用示例的任何问题或缺陷，请及时与西门子取得联系。西门子会在技术可行和商业合理的范围内，自行决定调查和修复任何问题或缺陷，为您提供支持。

安全信息

西门子提供具有工业安全功能的产品和解决方案，支持工厂、系统、机器和网络的安全运行。

为了保护工厂、系统、机器和网络免受网络威胁，有必要实施——并持续维护——一个整体的、最先进的工业安全概念。西门子的产品和解决方案构成了这一概念的一个元素。

客户有责任防止对其工厂、系统、机器和网络的未经授权的访问。

这些系统、机器和组件只应在必要的情况下连接到企业网络或 Internet，并且只有在适当的安全措施(例如防火墙和/或网络分割)到位的情况下才应连接到这种连接。有关可能实施的工业保安措施的其他资料，请浏览 <https://www.siemens.com/industrialsecurity>。

西门子的产品和解决方案经过不断的发展，使其更加安全。西门子强烈建议，一旦产品更新可用，就立即应用产品更新，并使用最新的产品版本。使用不再受支持的产品版本以及未能应用最新更新可能会增加客户遭受网络威胁的风险。

了解产品更新，请订阅西门子工业安全 RSS <https://www.siemens.com/industrialsecurity>。

西门子已建立接收西门子产品和解决方案安全漏洞信息的平台。您可以通过向 productcert@siemens.com 或 src.cyscn.cn@siemens.com 发送邮件的方式报送您发现或遇到的西门子产品和解决方案的安全漏洞。

西门子将在 <https://www.siemens.com/industrialsecurity> 上不时公布西门子产品和解决方案的安全漏洞和修补措施(如有)。用户应定期访问上述网站并及时采取相关修补措施。西门子强烈建议用户在上述网站登记并订阅 Security Advisory，从而以获取关于最新的安全漏洞和修补措施的及时推送。

目录

1	应用概述.....	5
1.1	通用描述.....	5
1.2	硬件及软件需求	5
2	顺序启动停止.....	6
2.1	应用描述.....	6
2.2	顺序启动顺序停止库程序	6
2.3	顺序启动逆序停止库程序	7
2.4	HMI 参考画面.....	8
3	范围限制.....	9
3.1	应用描述.....	9
3.2	范围限制库程序	9
4	多路复用器	10
4.1	应用描述.....	10
4.2	多路复用库程序	10
5	IO 调整.....	12
5.1	应用描述.....	12
5.2	IO 调整库程序	12
5.3	使用示例.....	13
5.4	参考画面.....	16
6	量程转换.....	17
6.1	应用描述.....	17
6.2	量程转换库程序	17
6.3	功能描述.....	18
6.4	参考画面.....	20
7	数字量信号	21
7.1	应用描述.....	21
7.2	应用程序.....	21
7.3	应用样例.....	23
7.4	参考画面.....	25

1 应用概述

8	三位执行器	26
8.1	应用描述.....	26
8.2	PLC 程序介绍.....	26
8.3	HMI 界面介绍.....	29
9	双手操纵	30
9.1	应用描述.....	30
9.2	双手操作库.....	30
10	两位执行器	32
10.1	应用描述.....	32
10.2	两位执行器库.....	32
10.3	功能描述.....	34
10.4	参考画面.....	35
11	8 路连锁	36
11.1	8 路连锁程序.....	36
11.2	参考画面.....	38
12	温压补偿	39
12.1	温压补偿原理.....	39
12.2	温压补偿程序.....	39
13	正弦发生器	41
13.1	正弦发生器程序.....	41
13.2	使用例程.....	41
14	平均扫描用时	44
14.1	应用描述.....	44
14.2	平均扫描用时库.....	44
14.3	功能描述.....	44
15	实数编码	46
15.1	应用描述.....	46
15.2	实数编码/解码库.....	46
15.3	功能描述.....	47
16	更新日志	48

1 应用概述

1.1 通用描述

SMART 的基本控制库（LBC）提供了基本的控制模块，根据西门子编程风格指南和 PLC Open 为标准化概念编程。

1.2 硬件及软件需求

本应用软硬件的需求

为了使得本应用案例成功运行，必须满足以下硬件和软件需求。

硬件

S7-200 SMART CPU 控制器:

- SIMATIC S7-200 SMART 产品家族

软件

- STEP 7-Micro/WIN SMART

2 顺序启动停止

2.1 应用描述

基于工艺的要求或者电网负荷的要求，有时需要设备按一定的顺序进行启动或者停止。该库可以顺序启动，顺序停止，或者顺序启动，逆序停止，自动按照一定的时间间隔对设备进行启停操作，当某台设备由于不满足启动条件，则自动跳过该设备等功

2.2 顺序启动顺序停止库程序

设备启动时，按照 Active 最低位到最高位顺序依次启动，时间间隔默认 5000ms，若某台设备不满足启动条件或已经处于启动状态，则跳过该设备。

设备停止时，按照 Active 最低位到最高位顺序依次停止，时间间隔默认 5000ms，若某台设备之前未处于启动状态，则跳过该设备。

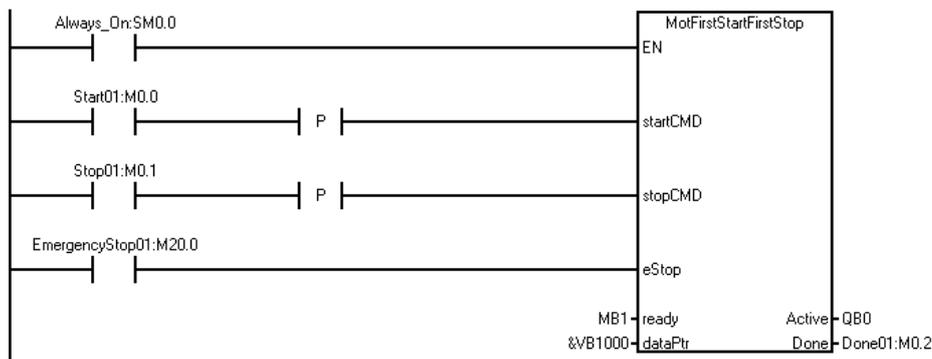


图 1 顺序启停库程序

程序块引脚

参数 & 类型	数据类型	描述
EN	BOOL	程序块使能,使用 SM0.0
startCMD	IN BOOL	设备启动命令,沿触发
stopCMD	IN BOOL	设备停止命令,沿触发
eStop	IN BOOL	设备急停命令
ready	IN BYTE	对应 8 台设备允许启动信号,若无相应的设备或者该设备未准备好,则相应位为 OFF。
dataPtr	IN DWORD	数据指针,用于内部计算的过程变量,占用 14 个字节。

2 顺序启动停止

Active	OUT	BYTE	对应 8 台设备启动输出，位状态为 1，代表对应设备启动，反之，位状态为 0，代表对应设备停止。
Done	OUT	BOOL	启动过程或者停止过程当中，该位为 0

请注意，设备启停之间的时间间隔默认为 5000ms,若需要调整，可将库文件中的常数 5000 修改为其他数值。

在任意时刻，都可以通过急停按钮来停止输出。

2.3 顺序启动逆序停止库程序

设备启动时，按照 Active 最低位到最高位顺序依次启动，时间间隔默认 5000ms，若某台设备不满足启动条件或已经处于启动状态，则跳过该设备。

设备停止时，按照 Active 最高位到最低位顺序依次停止，时间间隔默认 5000ms，若某台设备之前未处于启动状态，则跳过该设备。

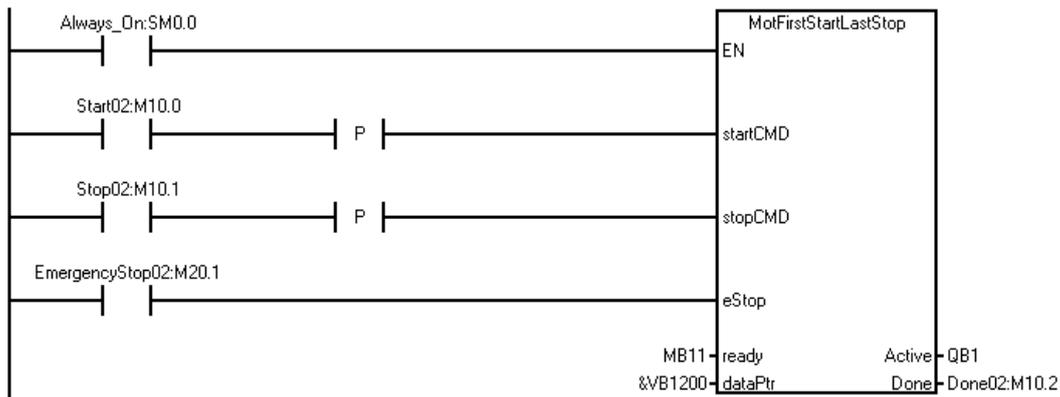


图 2 顺序启动逆序停止库程序

程序块引脚

参数 & 类型	数据类型	描述
EN	BOOL	程序块使能,使用 SM0.0
startCMD	IN BOOL	设备启动命令，沿触发
stopCMD	IN BOOL	设备停止命令，沿触发
eStop	IN BOOL	设备急停命令
ready	IN BYTE	对应 8 台设备允许启动信号，若无相应的设备或者该设备未准备好，则相应位为 OFF。
dataPtr	IN DWORD	数据指针，用于内部计算的过程变量，占用 14 个字节。

2 顺序启动停止

Active	OUT	BYTE	对应 8 台设备启动输出，位状态为 1，代表对应设备启动，反之，位状态为 0，代表对应设备停止。
Done	OUT	BOOL	启动过程或者停止过程当中，该位为 0

请注意，设备启停之间的时间间隔默认为 5000ms,若需要调整，可将库文件中的常数 5000 修改为其他数值。

在任意时刻，都可以通过急停按钮来停止输出。

2.4 HMI 参考画面

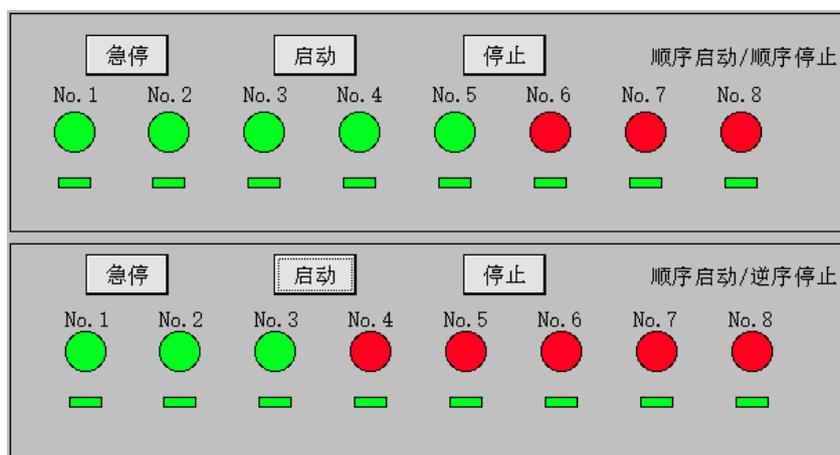


图 5 顺序启停 HMI 参考画面

3 范围限制

3.1 应用描述

范围限制用于根据需求，限制输出值在一定范围之内。

3.2 范围限制库程序

该库程序用于限制输入变量在上限数值与下限数值之间，当输入值小于等于下限值时，输出为下限值，当输入值大于等于上限值时，输出为上限值，当输入值位于上限与下限之间时，输出值等于输入值。

库程序中共包含 4 个块，适用于 4 个数据类型进行范围限制，分别为：
ScopeLimitReal, ScopeLimitDint, ScopeLimitInt, ScopeLimitByte。

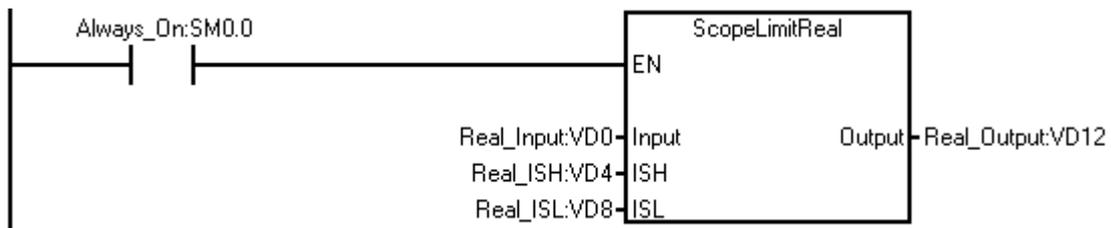


图 3 范围限制库程序

程序块引脚

参数 & 类型		数据类型	描述
EN		BOOL	程序块使能,使用 SM0.0
Input	IN	REAL	输入值
ISH	IN	REAL	输入值上限
ISL	IN	REAL	输入值下限
Output	OUT	REAL	$Input \leq ISL, Output = ISL;$ $ISL < Input < ISH, Output = Input;$ $Input \geq ISH, Output = ISH;$

4 多路复用器

4.1 应用描述

多路复用根据两路输入信号状态组合，输出预设的四路模拟量值。

4.2 多路复用库程序

该库指令根据两个数字量信号的组合选择，输出多达四个预定义的模拟量值或者输出 0。比如控制风机频率运行在 20 赫兹，35 赫兹，40 赫兹，50 赫兹。

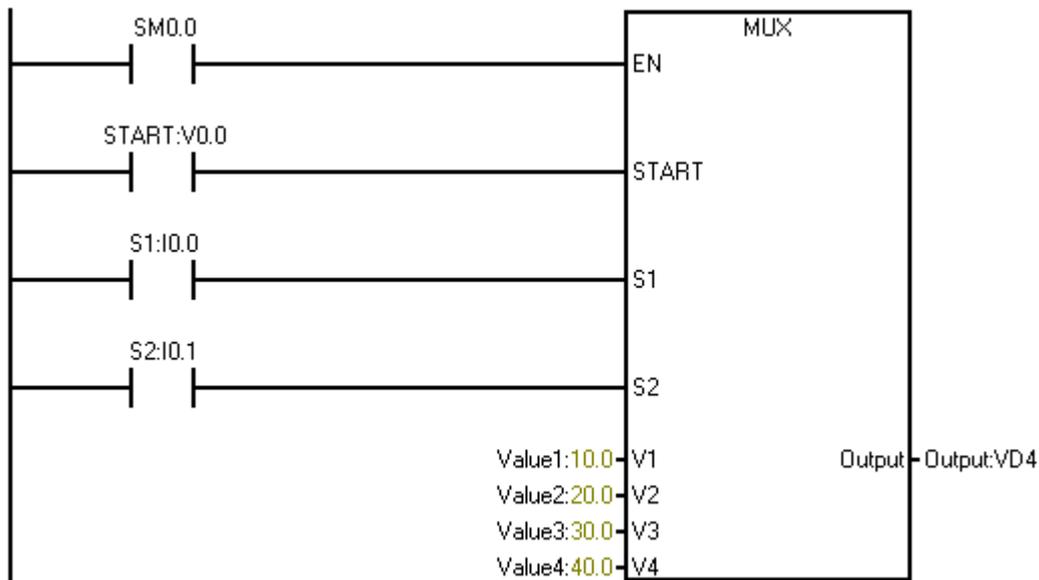


图 4 多路复用器库程序

程序块引脚

参数 & 类型	数据类型	描述
EN	BOOL	程序块使能,使用 SM0.0
START	IN BOOL	高电平时, 根据 S1 和 S2 的状态将预设值输出至 Output。 低电平时, 将 0 输出至 Output。
S1	IN BOOL	S1 和 S2 状态用于选择要发布的输出值 S1=0 和 S2=0: 输出值=V1 S1=0 和 S2=1: 输出值=V2
S2	IN BOOL	S1=1 和 S2=0: 输出值=V3 S1=1 和 S2=1: 输出值=V4
V1	IN REAL	预设值 1

4 多路复用器

V2	IN	REAL	预设值 2
V3	IN	REAL	预设值 3
V4	IN	REAL	预设值 4
Output	OUT	REAL	输出值

5 IO 调整

5.1 应用描述

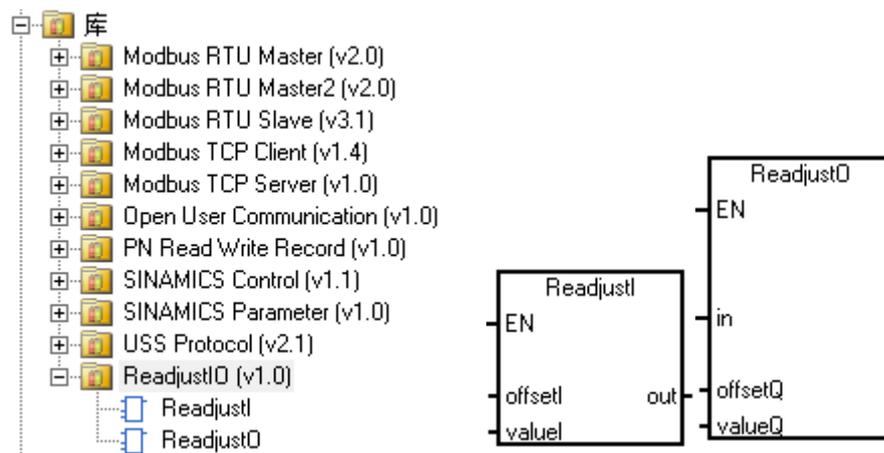
在实际的工程应用中，PLC 系统一般 IO 数量都会有一定冗余设计，这样当其中某一个或者几个 IO 点发生损坏时，可以及时使用备用点，但是这样就需要同时更改程序，对现场维修人员的要求比较高，针对这种工况，开发出调整 IO 的功能库，可以在不更改程序的基础上在 HMI 上直接选择对应 IO 备用点，仅需要用户更改外部接线既可，大大减小了维修难度，同时使用该功能库也可以实现单按钮控制多个输出。

5.2 IO 调整库程序

该程序库主要可以实现如下功能：

1. 不需要更改程序，可以灵活调整 IO 对应功能。
2. 可以实现单按钮控制多个输出。

程序库 IO 调整库



程序块引脚

参数 & 类型		数据类型	描述
EN		BOOL	程序块使能
in	IN	BOOL	输入信号
offsetI	IN	BYTE	输入 valueI 的偏置，比如 ID0，偏置为 1 对应 IO.0，最大 32
offsetQ	IN	BYTE	输出 valueQ 的偏置，比如 QD0，偏置为 8 对应 Q0.7，最大 32
valueI	IN	DWORD	输入 I 地址，如 ID0
valueQ	IN/OUT	DWORD	输出 Q 地址，如 QD0
out	OUT	BOOL	根据 valueI+offsetI 计算出的 I 点输出结果

简单说明

ReadjustI 功能块可以实现不同的输入 I 点选择对应特定的功能。

ReadjustO 功能块可以实现特定功能对应不同的 Q 点选择。

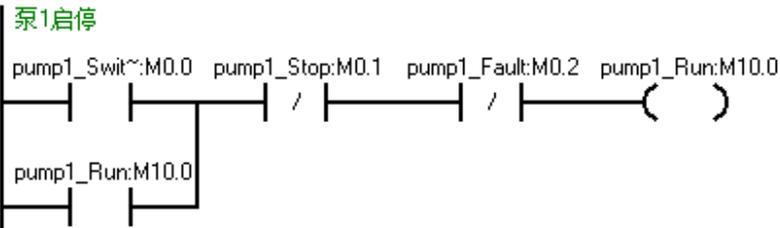
5.3 使用示例

程序调用

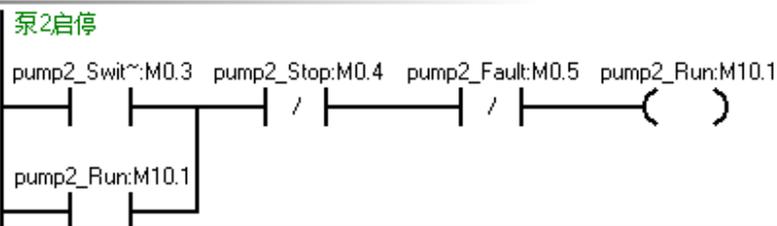
本例程中定义了 2 台水泵的启停操作，其中

M0.0 为泵 1 启动，M0.1 为泵 1 停止，M0.2 为泵 1 故障信号，M10.0 为泵 1 运行

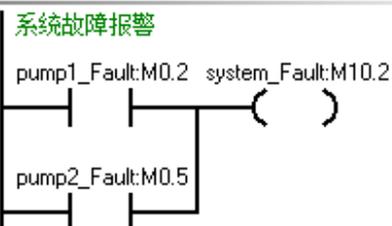
M0.3 为泵 2 启动，M0.4 为泵 2 停止，M0.5 为泵 2 故障信号，M10.1 为泵 2 运行



符号	地址	注释
pump1_Fault	M0.2	泵1故障
pump1_Run	M10.0	泵1运行
pump1_Stop	M0.1	泵1停止
pump1_SwitchOn	M0.0	泵1启动



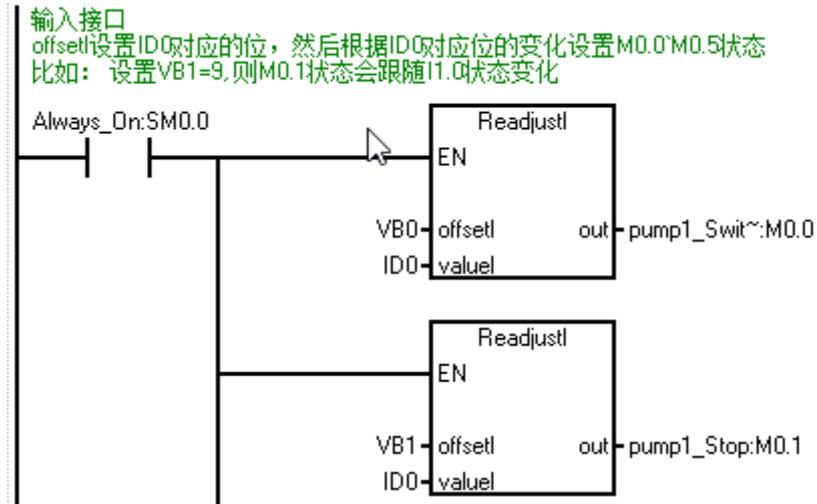
符号	地址	注释
pump2_Fault	M0.5	泵2故障
pump2_Run	M10.1	泵2运行
pump2_Stop	M0.4	泵2停止
pump2_SwitchOn	M0.3	泵2启动



符号	地址	注释
pump1_Fault	M0.2	泵1故障
pump2_Fault	M0.5	泵2故障
system_Fault	M10.2	系统故障

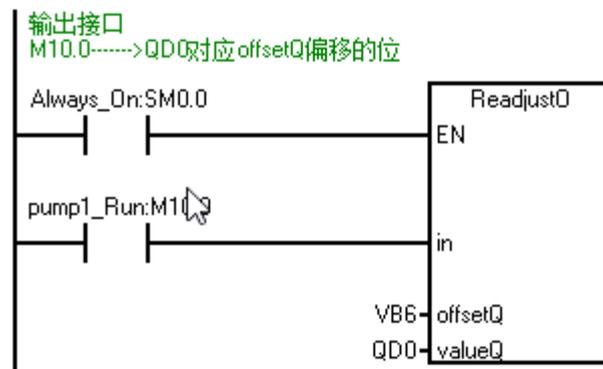
定义输入接口设置

程序段中 M0.0 的状态由 ID0+offset 对应的位决定，如程序中 VB0=1，则 M0.0 的状态由 I0.0 决定，通过更改 offset 可以改变对应关系。

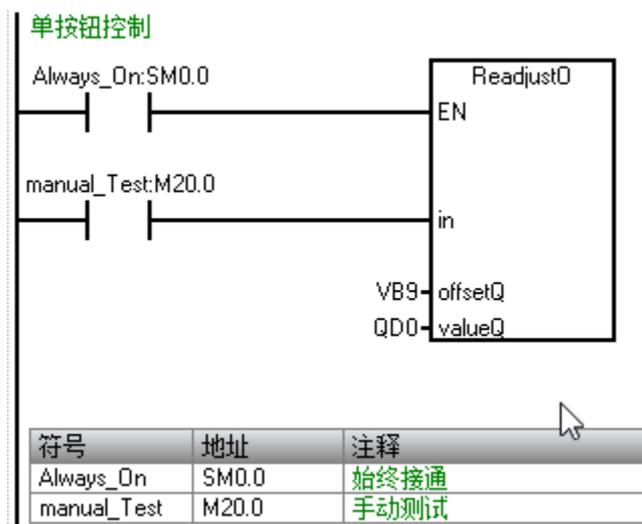


定义输出接口

程序段中 M10.0 的状态决定了 QD0+offset 对应的位，如程序中 VB6=1，则 Q0.0 的状态由 M10.0 决定，更改 offset 的值可以改变对应关系



单按钮控制



5.4 参考画面

调整 IO 触摸屏样例程序使用的是 10 寸精彩触摸屏，通过触摸屏和 PLC 程序通讯，实现灵活调整 IO 对应功能的操作。

画面概览

本触摸屏样例程序画面如下：

可以灵活设置每个功能对应的 IO 硬件，也可以通过设置序号来实现单按钮控制多个输出。

通过如图所示的设置功能如下：

I1---泵 1 启动； I2---泵 1 停止； I3---泵 1 故障；

I4---泵 2 启动； I5---泵 2 停止； I6---泵 2 故障；

Q1---泵 1 运行； Q2---泵 2 运行； Q3---系统故障；

以上功能可以在 HMI 上灵活调整对应关系。



6 量程转换

6.1 应用描述

在工业自动化领域，经常需要对各种参数进行测量和检测。不同的传感器和设备通常具有不同的量程范围。量程转换可以将不同量程范围的数据进行转换和适配，能够准确的基于转换后的数据做出相应的控制和调整。

该库可以用于处理工程量和模拟量的转换。

6.2 量程转换库程序

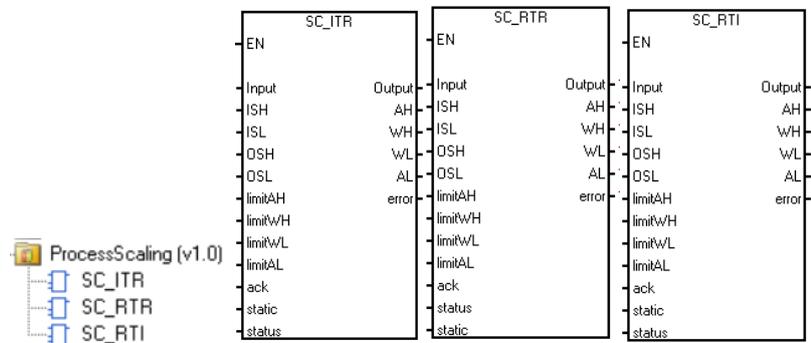
量程转换库程序共包含 3 个功能块：

SC_ITR：实现缩放模拟量输入的值转换为工程量的值。

SC_RTR：实现浮点数到浮点数的转换。

SC_RTI：实现缩放有意义的单位值（温度，高度，压力.....），将其转换为硬件模拟输出值。

程序库-量程转换库



程序块引脚

输入参数

参数 & 类型	数据类型	描述
EN	BOOL	程序块使能
Input	IN	需要缩放的输入值
ISH	IN	输入值的范围上限
ISL	IN	输入值的范围下限
OSH	IN	输出值的范围上限
OSL	IN	输出值的范围下限
LimitAH	IN	输出值的报警上限

6 量程转换

LimitWH	IN	REAL / INT	输出值的警告上限 (LimitAH>LimitWH)
LimitWL	IN	REAL / INT	输出值的警告下限
LimitAL	IN	REAL / INT	输出值的报警下限 (LimitAL<LimitWL)
ack	IN	BOOL	报警确认
stauts	INOUT	BYTE	错误代码
static	INOUT	BYTE	过程变量

输出参数

参数 & 类型		数据类型	描述
Output	OUT	REAL / INT	缩放后的输出值
AH	OUT	BOOL	到达报警上限
WH	OUT	BOOL	到达警告上限
WL	OUT	BOOL	到达警告下限
AL	OUT	BOOL	到达报警下限
error	OUT	BOOL	报错

错误代码说明

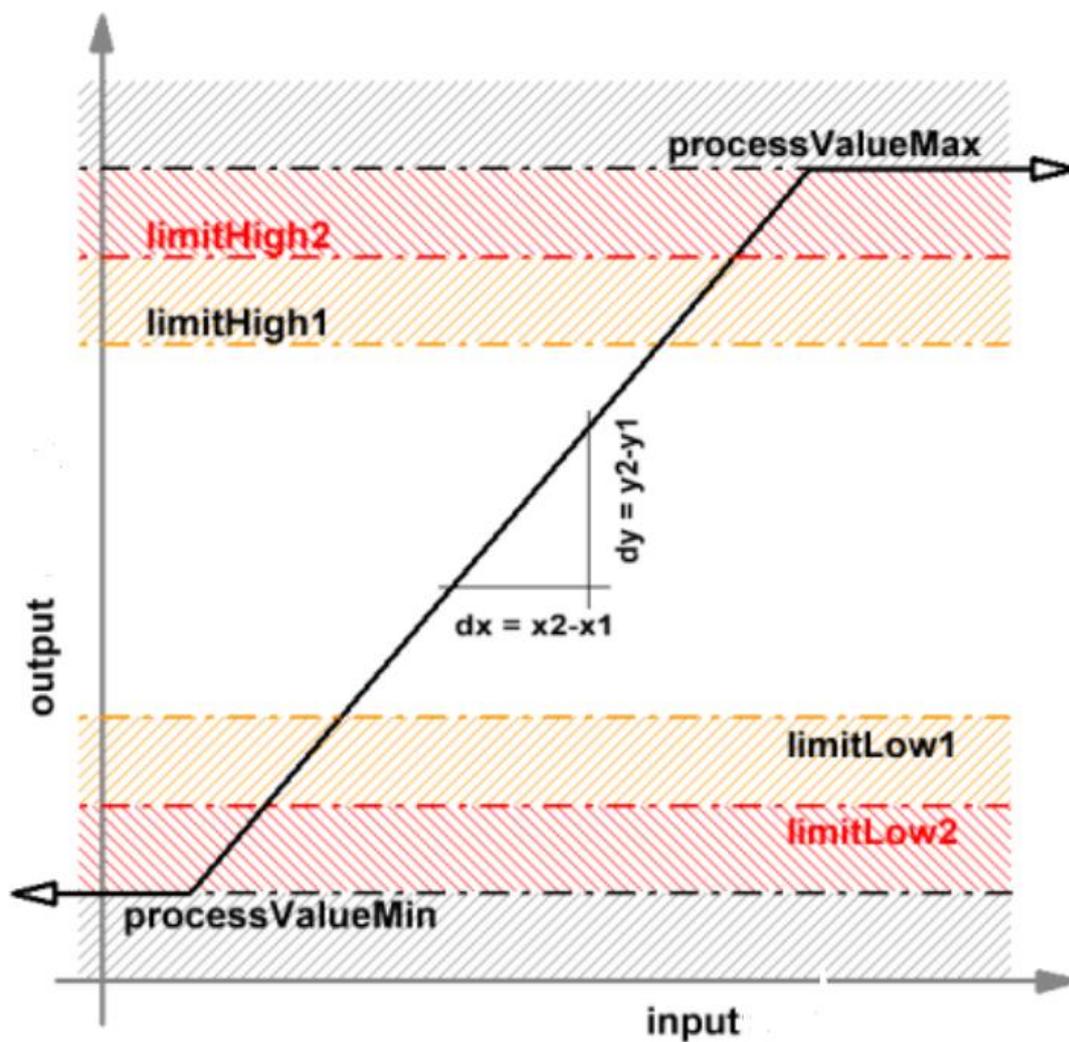
报错状态值	描述
Bit 0	为 1 时表示, 输入值小于输入范围下限 (input<ISL)
Bit 1	为 1 时表示, 输入值大于输入范围上限(input>ISH)
Bit 2	为 1 时表示, 输入范围下限大于输入范围上限(ISL>ISH)
Bit 3	为 1 时表示, 输出范围下限大于等于输出范围上限(OSL≥OSH)
Bit 4	为 1 时表示, 输出值的警告下限大于输出值的警告上 (LimitWL>limitWH)
Bit 5	为 1 时表示, 输出值的报警下限大于输出值的报警上限 (LimitAL>LimitAH)
Bit 6	备用
Bit 7	备用

6.3 功能描述

函数的处理是通过将 True 信号赋值给 enable 来开始的, 输出值根据输入的设置进行缩放。

为确保缩放功能正确工作, 必须要给输入和输出设置最小值和最大值范围。转换关系参考下方转换函数部分内容。

该功能块集成定义高低限值, 当达到某一值时产生报警或警告。



$$dx = x_2 - x_1 = \text{ISH} - \text{ISL}$$

$$dy = y_2 - y_1 = \text{OSH} - \text{OSL}$$

转换函数

$$y = \frac{y_2 - y_1}{x_2 - x_1} \cdot (x - x_1) + y_1$$

x = input

y = output

x_1 = 输入的范围下限 (ISL)

x_2 = 输入的范围上限 (ISH)

y_1 = 输出的范围下限 (OSL)

y_2 = 输出的范围上限 (OSH)

6.4 参考画面

转换后的值通常需要在画面中显示，基于 Smart Line 示例画面集成了该库中所有功能。



1. 显示转换后的输出实时值
2. 指示是否已超过报警或警告的上限值
报警：红色
警告：橙色
3. 指示是否已超过报警或警告的下限值
4. 报警确认按钮
5. 设定输入及输出范围值
6. 设定报警及警告限值
7. 仿真外部输入用

7 数字量信号

7.1 应用描述

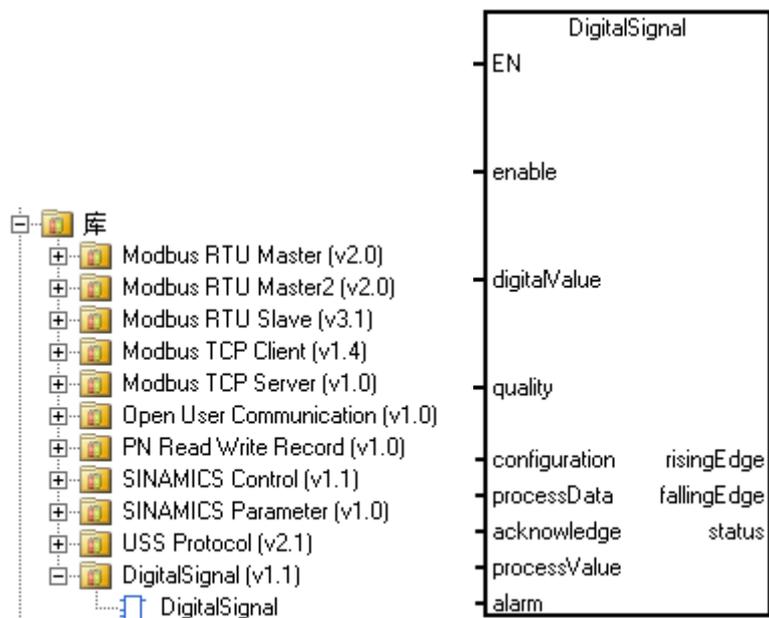
数字量信号是常用的工业信号之一。在实际的工程应用中，获取的数字量信号可能不能作为控制信号直接使用，而需要进一步处理。该功能模块可用于处理原始信号，并结合信号质量，延时与过滤，最终给出有效信号。

该程序库主要可以实现如下功能：

1. 配置信号参数。
2. 输出转换后的有效信号和警报。

7.2 应用程序

程序库-数字量信号库



7 数字量信号

程序块引脚

参数 & 类型		数据类型	描述
EN		BOOL	程序块使能
enable	IN	BOOL	模块使能
digitalValue	IN	BOOL	输入信号
quality	IN	BOOL	信号质量
configuration	IN	DWORD	数字量信号配置地址
processData	IN	DWORD	数字量信号过程地址-临时变量存放区域
acknowledge	IN/OUT	BOOL	确认警报
processValue	IN/OUT	BOOL	输出信号
alarm	IN/OUT	BOOL	警报
risingEdge	OUT	BOOL	输出信号上升沿
fallingEdge	OUT	BOOL	输出信号下降沿
status	OUT	WORD	信号状态

Configuration 指针地址偏移

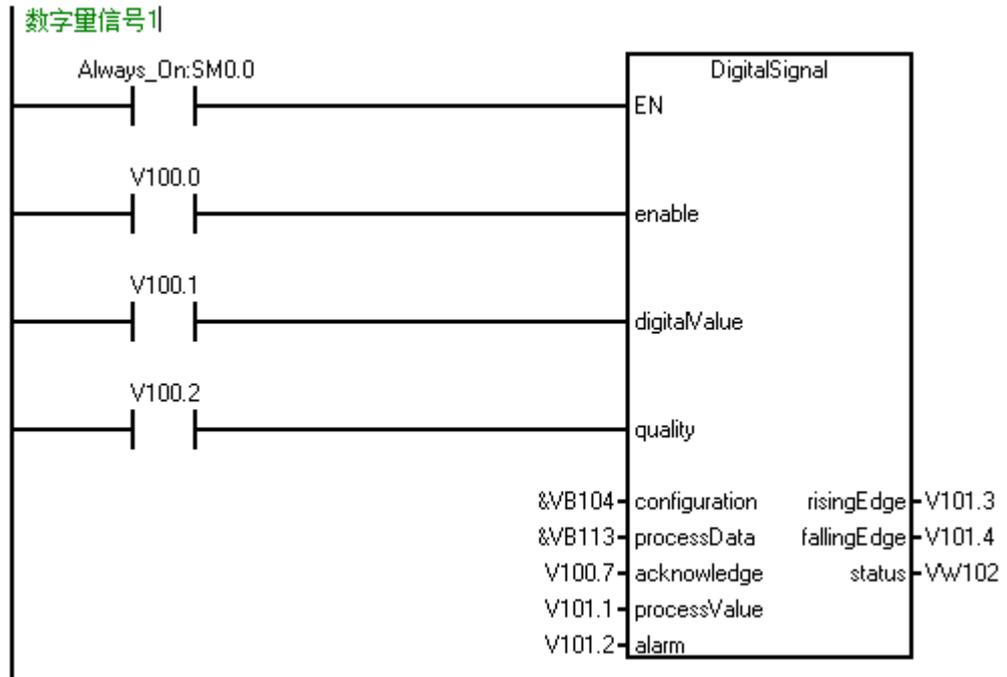
参数 & 类型			数据类型	描述
字节偏移	名称	位号		
0	defaultValue	0	BOOL	默认输出（信号质量差或禁用时）
	invertProcessValue	1	BOOL	反转信号设置
	disableAlarm	2	BOOL	禁用警报设置（0-有警报，1-禁用警报）
1-4	onDelay		DINT	开启延时，单位 ms
5-8	offDelay		DINT	关闭延时，单位 ms

processData 指针地址偏移

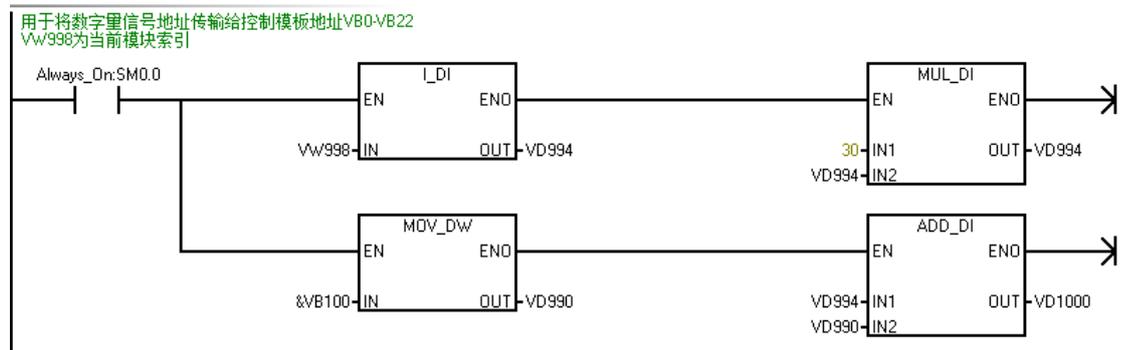
参数 & 类型			数据类型	描述
字节偏移	名称	位号		
0-3	beginTime		DWORD	时间间隔定时器开始时间
4-7	calTime		DWORD	时间间隔定时器时间差
8	timerStartBool	0	BOOL	时间间隔定时器启动标志位

7.3 应用样例

程序调用



本例程中定义了 4 个数字量信号。
 VB100-VB122 为数字量信号 1 地址。
 VB130-VB152 为数字量信号 2 地址。
 VB160-VB182 为数字量信号 3 地址。
 VB190-VB202 为数字量信号 4 地址。
 VB10000-VB10049 为设备名称地址。



VB0-VB22 位 HMI 画面模板数字量信号地址
 VB994-VB1009 为画面所用索引和指针地址

定义接口

以数字量信号 1 为例，其中：

VB100 为输入 (位 0-功能块激活，位 1-数字量输入，位 2-质量，位 7-确认警报)

VB101 为输出 (位 1-输出，位 2-警报，位 3-输出上升沿，位 4-输出下降沿)

VW102 是状态

VB104-VB112 (9 Byte)是配置

其中，VB104 是 bool 数据 (位 0-默认电平，位 1-对调输出，位 2-禁用警报)

VB105 是 DINT 数据-开启延时

VB109 是 DINT 数据-关闭延时

VB113-VB122 (9 Byte)是过程数据

其中，VB105 是 DINT 数据-延时开始时间

VB109 是 DINT 数据-当前延时时间

VB104 是 bool 数据 (位 0-延时开启位)

需注意，配置与过程数据地址应连续，且不可重复使用。

数字量信号 2, 3, 4 结构与数字量信号相同，偏移 30 字节。

7.4 参考画面

数字量信号触摸屏样例程序使用的是 10 寸精彩触摸屏，通过触摸屏和 PLC 程序通讯，实现灵活数字量信号对应功能的操作。

画面概览

本触摸屏样例程序画面如下：

通过左侧按钮，可以灵活监控和设置每个数字量信号，

左侧界面可显示或修改数字量信号状态和参数。

若要修改参数，需要先点击修改按钮解锁参数。



8 三位执行器

8.1 应用描述

基于 SIMATIC S7-200 SMART 开发的三位执行器控制程序，基本功能如下：

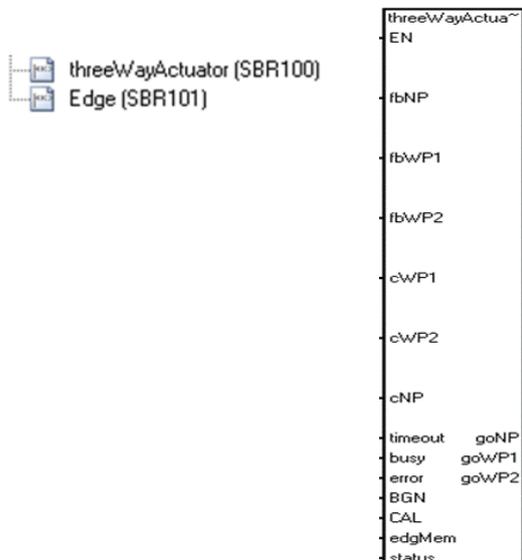
- HMI 界面通过不同颜色显示当前三位执行器所在挡位，通过点击按钮触发执行器动作，通过闪烁来显示执行器动作的目标位置；
- 当执行器正在动作中，即使点击触发按钮，不会触发新的动作；
- PLC 记载执行器最后位置和动作中的目标位置，并传递给 HMI，除闪烁外，同时通过文本域描述执行器当前的动作状态；
- 用户可设置执行器动作超时时间，超时后触发相应报警并显示于 HMI；
- 若有任意两个或以上执行器位置反馈信号同时为正电平，触发位置反馈信号错误报警并显示于 HMI；
- 当 PLC 存在“错误”类型报警时，执行器不会触发任何动作；

8.2 PLC 程序介绍

功能说明：

- 此程序主要用于控制三位执行器的动作，并与 HMI 通信传递数据。
- 程序设置有动作超时时间和执行器位置反馈信号分析，错误时触发相应报警。
- 在执行器动作进行中，或有“错误”类型报警未处理，程序将阻止触发新的动作。

功能块：



引脚说明：

符号	变量类型	数据类型	注释
fbNP	IN	BOOL	中性位反馈信号
fbWP1	IN	BOOL	1号工位反馈信号
fbWP2	IN	BOOL	2号工位反馈信号
cWP1	IN	BOOL	收到至中性位命令
cWP2	IN	BOOL	收到至1号工位命令
cNP	IN	BOOL	收到至2号工位命令
timeout	IN	DINT	超时设置值（毫秒）
goNP	OUT	BOOL	发出至中性位命令
goWP1	OUT	BOOL	发出至一号工位命令
goWP2	OUT	BOOL	发出至二号工位命令
busy	IN_OUT	BOOL	状态忙标志位
error	IN_OUT	BYTE	错位字
status	IN_OUT	BYTE	状态字
BGN	IN_OUT	DINT	计时器缓存开始记录
CAL	IN_OUT	DINT	计时器缓存差值记录
edgMem	IN_OUT	BYTE	上升沿触发缓存

状态位与错误位定义：

State/Error	数值/位号	注释
State	01	执行器正在由中性位移动到 1 号工作位
State	02	执行器正在由中性位移动到 2 号工作位
State	12	执行器正在由 1 号工作位移动到 2 号工作位
State	10	执行器正在由 1 号工作位移动到中性位
State	20	执行器正在由 2 号工作位移动到中性位
State	21	执行器正在由 2 号工作位移动到 1 号工作位
State	30	执行器停留在中性位
State	31	执行器停留在 1 号工作位
State	32	执行器停留在 2 号工作位
Error	L6.1	位置反馈信号错误（有任意两个或以上执行器位置反馈信号同时为正电平）
Error	L6.2	执行器动作超时

8.3 HMI 界面介绍



功能介绍:

- “执行器动作状态”描述执行器正在进行动作的起始位置和目标位置
- “超时报警时长”栏设置执行器超时时长，单位位毫秒
- “中性位”“1号工作位”“2号工作位”按钮常为蓝色表示执行器当前停留于此位，闪烁蓝色表示执行器正朝此位运动
- 报警信息栏显示动作超时报警和位置反馈信号错误报警等

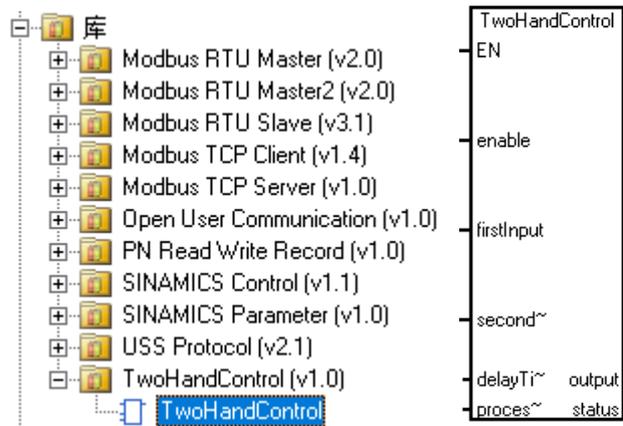
9 双手操纵

9.1 应用描述

监视两个输入状态（例如两个按钮），两个输入必须在一定时间内一起激活，才能将输出设置为 TRUE。

9.2 双手操作库

双手操作库



程序块引脚

参数 & 类型		数据类型	描述
EN		BOOL	程序块使能
enable	IN	BOOL	功能使能信号
firstInput	IN	BOOL	第一个输入信号
secondInput	IN	BOOL	第二个输入信号
delayTime	IN	INT	信号间延时时间设置，单位 ms
processValue	IN/OUT	INT	过程时间值
output	OUT	BOOL	输出结果
status	OUT	WORD	功能块状态 128 功能未激活 0 输入在正常时间内 1 功能块使能，等待输入 2 存在一个输入，开始计时，等待第二个输入 4 超时

简单说明

使用该功能块来监视两个输入信号（例如两个按钮），这两个输入信号必须在一定时间间隔内一起激活，才能将输出设置为 TRUE。当两个输入信号的时间间隔超过了设定的 delayTime，则输出超时信号。（注意，因使用了 SMW22 扫描时间，当扫描

9 双手操纵

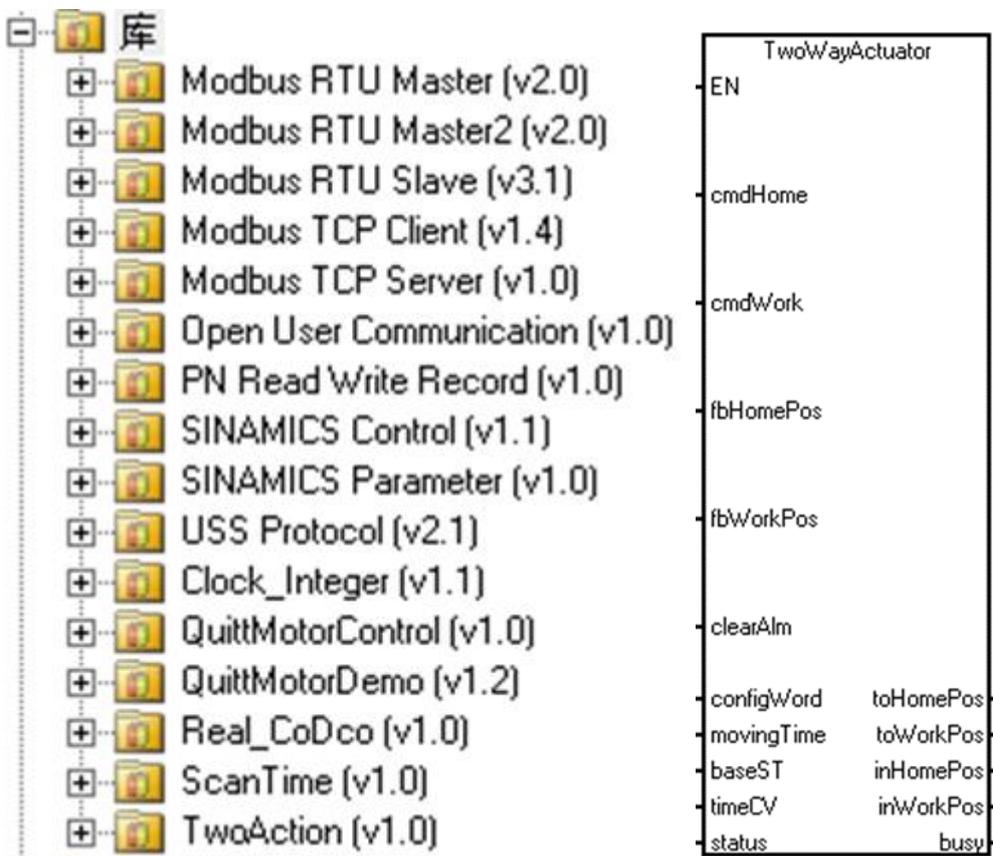
周期比较小时，实际计时会有偏差，用户使用时需要注意，建议扫描周期在 10ms 以上时使用)

10 两位执行器

10.1 应用描述

在应用环境中，存在需要控制到例如气缸、油缸等的场合，需要对控制指令以及反馈状态结合以作出输出控制以及报警等逻辑操作，因此编写两位执行器库以求稳定可靠地处理此类场景。

10.2 两位执行器库



程序引脚

参数 & 类型	数据类型	描述
EN	BOOL	程序块使能
cmdHome	IN BOOL	到 Home 位指令
cmdWork	IN BOOL	到 Work 位指令

10 两位执行器

fbHomePos	IN	BOOL	到 Home 位置反馈
fbWorkPos	IN	BOOL	到 Work 位置反馈
clearAlm	IN	BOOL	复位报警
configWord	IN	WORD	配置字, 详见下表
movingTime	IN	REAL	设定运动所需最大用时, 超出时可触发超时报警, 单位 ms
baseST	IN	REAL	单次扫描时间, 单位 ms
timeCV	IN/OUT	REAL	当前动作已用时间, 单位 ms
status	IN/OUT	WORD	当前状态与报警字, 定义详见下表
toHomePos	OUT	BOOL	运动到 Home 位置的控制位
toWorkPos	OUT	BOOL	运动到 Work 位置的控制位
inHomePos	OUT	BOOL	到达 Home 标志位
inWorkPos	OUT	BOOL	到达 Work 标志位
busy	OUT	BOOL	动作中

configWord 解析如下

位	定义	备注
B0 (对应 L1.0)	Set Output Home Continuously	TRUE: cmdHome 置位时即使 fbHomePos 为 true, toHomePos 仍然输出。 FALSE: cmdHome 置位时若 fbHomePos 为 true, toHomePos 停止输出。
B1 (对应 L1.1)	Set Output Work Continuously	TRUE: cmdWork 置位时即使 fbWorkPos 为 true, toWorkPos 仍然输出。 FALSE: cmdWork 置位时若 fbWorkPos 为 true, toWorkPos 停止输出。
B2 (对应 L1.2)	Disable Home Command	TRUE: 忽略引脚 cmdHome 的输入值, 当 cmdWork 为 false 时, 认为 cmdHome 为 true。 FALSE: 不忽略引脚 cmdHome 的输入值。
B3 (对应 L1.3)	Disable Feedback Home Pos	TRUE: 忽略引脚 fbHomePos 的输入值, 当 fbWorkPos 为 false 时, 认为 fbHomePos 为 true。 FALSE: 不忽略引脚 fbHomePos 的输入值。

10 两位执行器

B4 (对应 L1.4)	Disable Feedback Work Pos	TRUE: 忽略引脚 fbWorkPos 的输入值, 当 fbHomePos 为 false 时, 认为 fbWorkPos 为 true。 FALSE: 不忽略引脚 fbWorkPos 的输入值。
B5 (对应 L1.5)	Disable Alarms	TRUE: 不输出所有报警位。 FALSE: 正常输出报警位。
B6 (对应 L1.6)	Reset Outputs On Error	TRUE: 出现报警时停止输出。 FALSE: 出现报警时仍然输出。
B7 (对应 L1.7)	Enable Signal Conflict Alarm	TRUE: 检测到 cmdHome 与 cmdWork 同时为 true 时输出 CommandConflictAlarm 报警。 FALSE: 不输出 CommandConflictAlarm 报警。

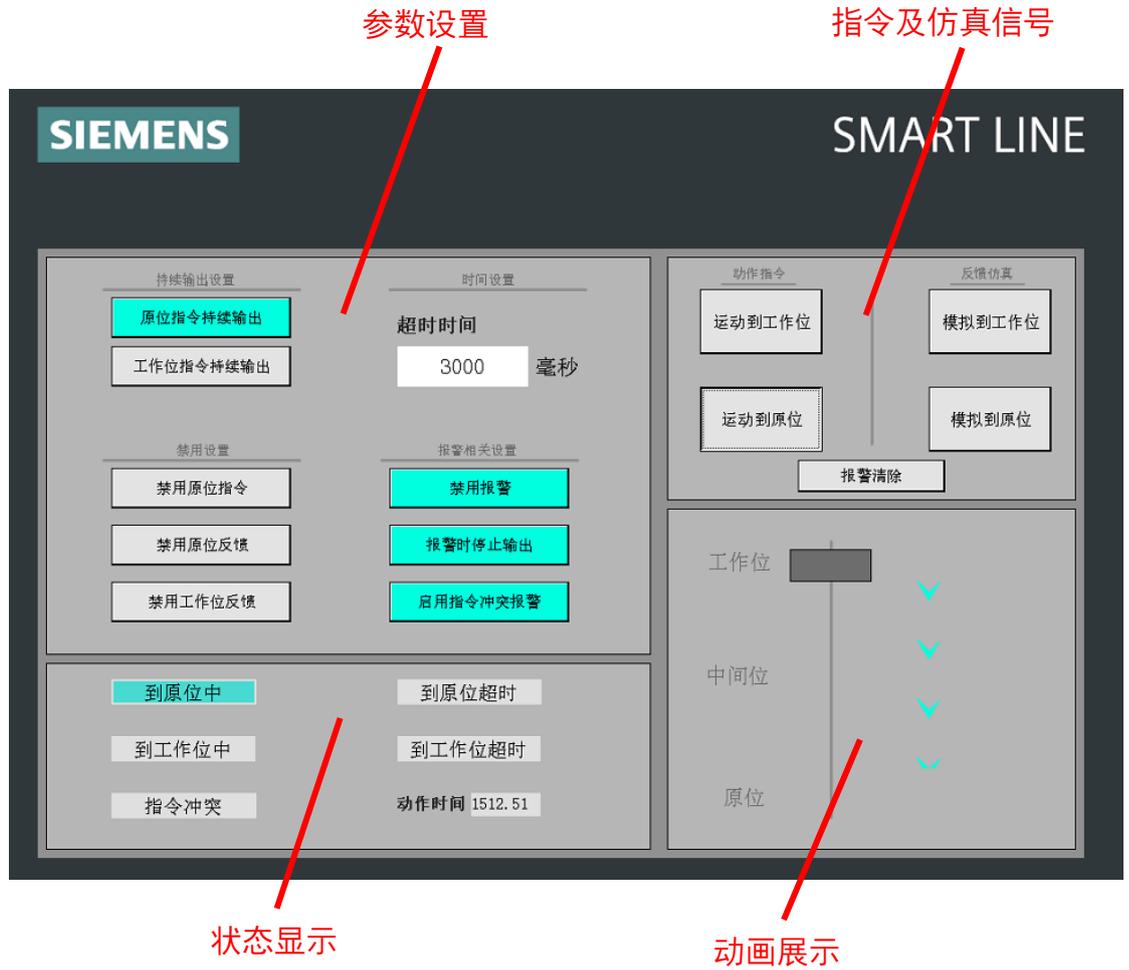
status 字解析

位	定义	备注
B0	GoingToHomePos	状态: 正在前往 Home 位置
B1	GoingToWorkPos	状态: 正在前往 Work 位置
B2	CommandConflictAlarm	报警: 指令冲突, 检测到 cmdHome 与 cmdWork 同时为 true
B3	ToWorkPosTimeoutAlarm	到 Work 位动作超时, 动作时间超出 movingTime 仍未检测到 fbWorkPos。
B4	ToHomePosTimeoutAlarm	到 Home 位动作超时, 动作时间超出 movingTime 仍未检测到 fbHomePos。

10.3 功能描述

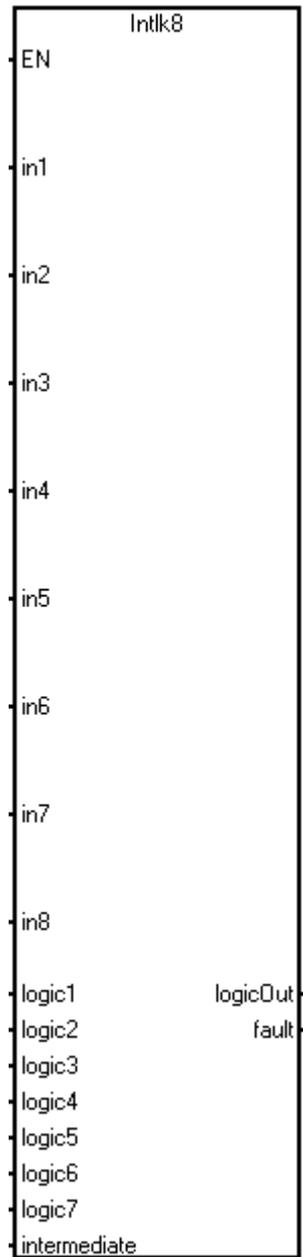
该功能块根据配置字的设置值对输入的指令信号以及反馈信号作综合判断后进行控制输出以及状态输出, 当满足报警条件时, 根据用户设置的配置参数判断是否输出报警或停止控制输出。

10.4 参考画面



11 8 路连锁

11.1 8 路连锁程序



8 路连锁程序块

程序块引脚

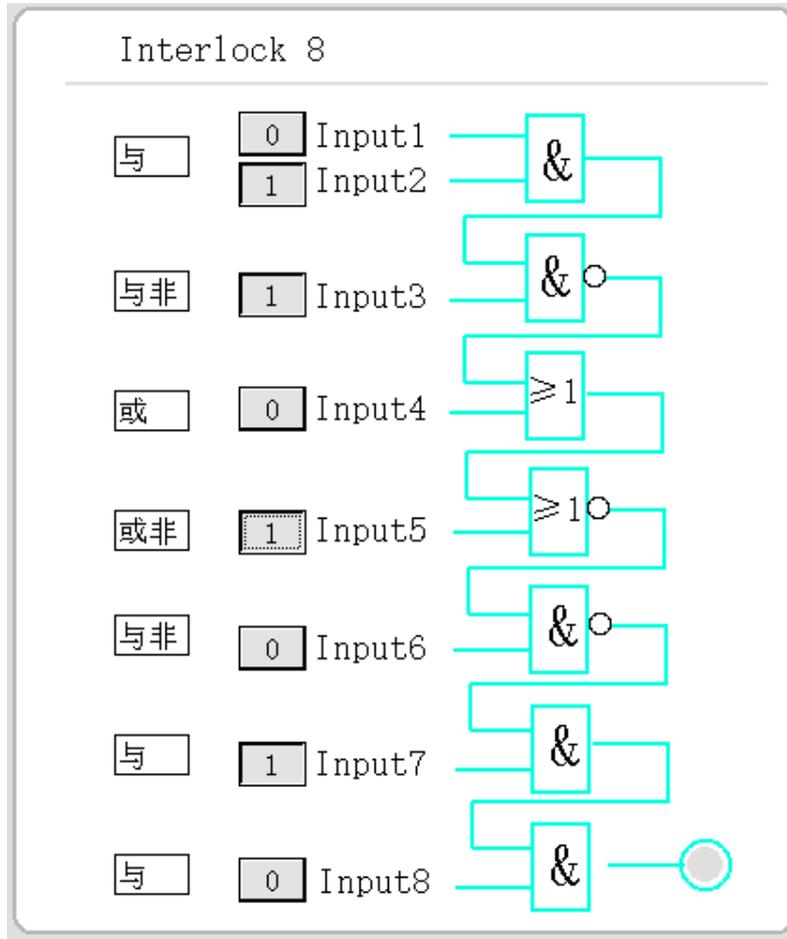
参数 & 类型	数据类型	描述
EN	BOOL	程序块使能, 一般使用 SM0.0
in1	IN BOOL	输入 1

11 8 路连锁

in2	IN	BOOL	输入 2
in3	IN	BOOL	输入 3
in4	IN	BOOL	输入 4
in5	IN	BOOL	输入 5
in6	IN	BOOL	输入 6
in7	IN	BOOL	输入 7
in8	IN	BOOL	输入 8
logic1	IN	BYTE	逻辑 1 (1 与, 2 与非, 3 或非, 4 或)
logic2	IN	BYTE	逻辑 2 (1 与, 2 与非, 3 或非, 4 或)
logic3	IN	BYTE	逻辑 3 (1 与, 2 与非, 3 或非, 4 或)
logic4	IN	BYTE	逻辑 4 (1 与, 2 与非, 3 或非, 4 或)
logic5	IN	BYTE	逻辑 5 (1 与, 2 与非, 3 或非, 4 或)
logic6	IN	BYTE	逻辑 6 (1 与, 2 与非, 3 或非, 4 或)
logic7	IN	BYTE	逻辑 7 (1 与, 2 与非, 3 或非, 4 或)
logicOut	OUT	BOOL	逻辑输出
fault	OUT	BOOL	错误输出

注意：逻辑 1 代表输入 1 和输入 2 之间的逻辑，逻辑 2 代表输入 1 和输入 2 之间逻辑的结果和输入 3 之间的逻辑，依次类推。

11.2 参考画面



12 温压补偿

12.1 温压补偿原理

由于气体的可压缩性，决定了其流量测量的复杂性，气体的实际流量与气体的实时温度和压力有关，所以气体测量流量普遍存在温压补偿问题，其实质就是被测气体的工况下温度，压力与设计数值不符时采取的密度修正措施。本库中使用的标准参比条件为绝对压力等于 101.325kPa 和温度为 20°C。一般气体温压补偿公式如下：

$$F = F_0 \times \sqrt{\frac{P + P_0}{P_b + P_0} \cdot \frac{T_b + T_0}{T + T_0}}$$

其中：

F ：补偿后流量

F_0 ：补偿前流量

P ：实测压力 (kPa)

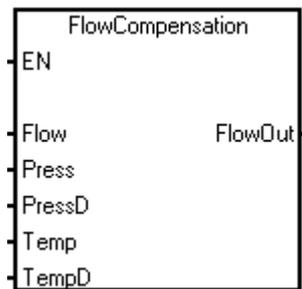
P_b ：设计压力 (kPa)

T ：实测温度 (°C)

T_b ：设计温度 (°C)

T_0 ：标准参比温度 (K)

12.2 温压补偿程序



温压补偿库程序

程序块引脚

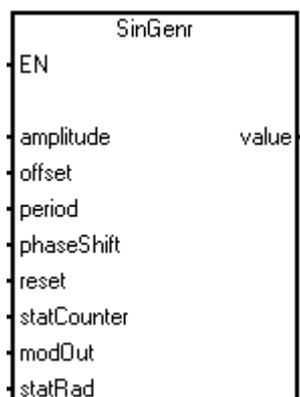
参数 & 类型	数据类型	描述
EN	BOOL	程序块使能, 一般使用 SM0.0
Flow	IN REAL	补偿前流量
Press	IN REAL	实测压力 kPa
PressD	IN REAL	设计压力 kPa

12 温压补偿

Temp	IN	REAL	实测温度°C
TempD	IN	REAL	设计温度°C
FlowOut	OUT	REAL	补偿后流量

13 正弦发生器

13.1 正弦发生器程序



正弦波发生器程序块

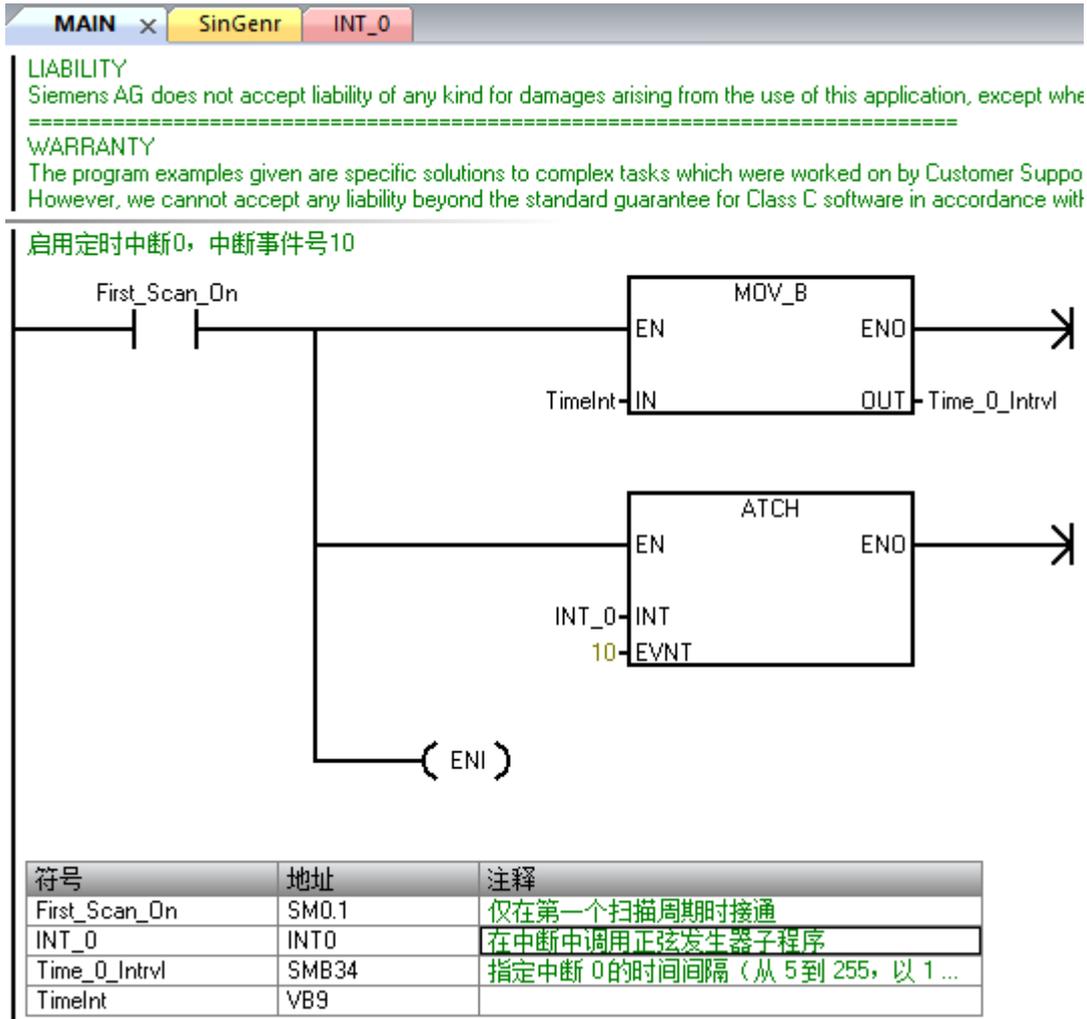
程序块引脚

参数 & 类型	数据类型	描述
EN	BOOL	程序块使能, 一般使用 SM0.0
amplitude	IN REAL	幅值
offset	IN REAL	偏移量
period	IN REAL	周期
phaseShift	IN REAL	相位偏移
reset	IN BOOL	复位
statCounter	INOUT DINT	过程值
modOut	INOUT DINT	过程值
statRad	INOUT DINT	过程值
value	OUT REAL	正弦波输出

注意：必须在定时中断中调用此块，中断时间为 timeInt，本例中调用的时间中断为定时中断 0，若采用定时中断 1，则需要将 SMB34 改为 SMB35。

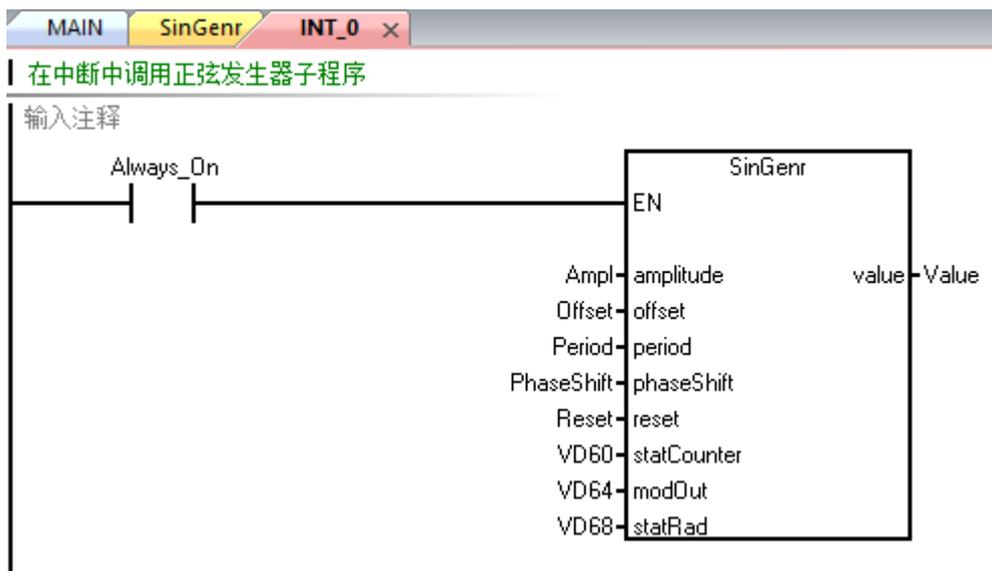
13.2 使用例程

主程序



注意：在主程序中给 SMB34 赋值中断时间

中断程序

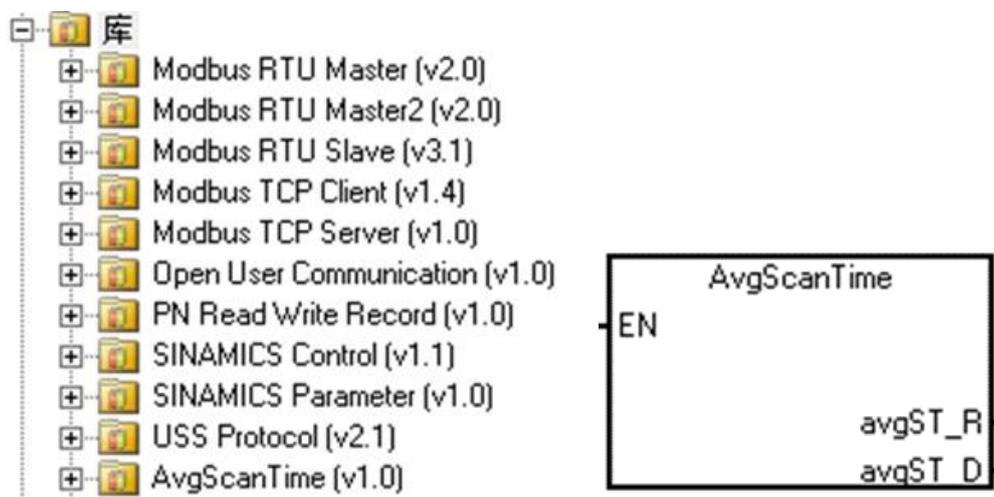


14 平均扫描用时

14.1 应用描述

在应用环境中，经常需要用到更高精度的扫描时间用于计时以节省定时器资源，该库使用平均值求每 0.5s 内的扫描次数以得出平均扫描用时，可用于精度要求较高的计时场合或用于评估 PLC 的计算负载，具有一定的实用性。

14.2 平均扫描用时库



程序块引脚

符号定义	类型	数据类型	含义
avgST_R	Output	Real	每个扫描周期的平均时间，以实数形式输出，单位 ms
avgST_D	Output	Dint	每个扫描周期的平均时间，以整型形式输出，单位 10 μ s（即 avgST_R 乘以 100 后取整）

14.3 功能描述

- 该库占用 13 个 VB 寄存器，其他寄存器不占用。
- 使用时该库仅可使用一次，不可重复调用。
- 建议位于第一段程序中调用该库以保证一致性。

14 平均扫描用时

- 该库依靠求平均值的方式求出 500ms 内每个扫描周期的扫描时间，运行 500ms 后所得的数据方视为有效。
- 实数运算精度参照 S7-200_SMART 系统说明手册。

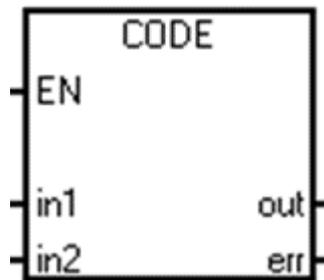
15 实数编码

15.1 应用描述

在使用 SIMATIC S7-200 SMART 编程时，尤其是编程运动控制过程中，经常会遇到需要断电保存浮点数的情况，在某些特定场合中，所需保存的浮点数数值不大且精度要求不高，然由于浮点数保存格式的原因需占用大量宝贵的寄存器资源，因此对该类型实数进行压缩编码可极大节省编程时占用的寄存器空间。

15.2 实数编码/解码库

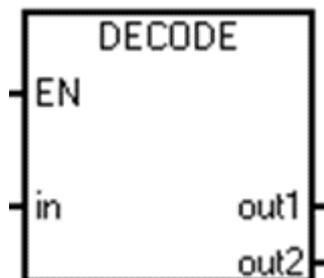
编码功能块



编码程序块引脚

参数 & 类型	数据类型	描述
EN	BOOL	程序块使能
In1	IN REAL	位于编码高位的浮点数， $0 < \text{in1} < 429.48$
In2	IN REAL	位于编码低位的浮点数， $0 < \text{in2} < 999.99$
out	OUT DINT	编码后的双精度整型数，err=0 时有效
err	OUT BOOL	错误输出

解码功能块



解码程序块引脚

参数 & 类型		数据类型	描述
EN		BOOL	程序块使能
In	IN	DINT	编码后的双精度整型数
out1	OUT	REAL	位于编码高位的浮点数, $0 < in1 < 429.48$
out2	OUT	REAL	位于编码低位的浮点数, $0 < in2 < 999.99$

15.3 功能描述

- 使用该功能库时，解码与编码功能需配合使用。
- 使用 CODE 子程序时，需注意编码时输入数范围为：
 - $0 < IN1 < 429.48$
 - $0 < IN2 < 999.99$
 超出此范围时，CODE 子程序的 err 引脚将输出高电平，此时 out1 输出将无效。
- 当浮点数精度高于 0.01 时，CODE 子程序将对 0.01 后的数进行四舍五入操作。使用 CODE 子程序时，若在子程序或中断中调用 CODE，且 out 引脚使用局部变量寄存器，则需注意 out 的数据类型为 DINT。
- 使用 DECODE 子程序时，若在子程序或中断中调用 DECODE，且 in 引脚使用局部变量寄存器，则需注意 in 的数据类型为 DINT。

16 更新日志

版本& 日期	更新描述
V1.0.0 07/2023	