

SoMachine Basic

操作指南

04/2014

EIO0000001359.01

www.schneider-electric.com

Schneider
 **Electric**

本文件中提供的信息包含有关此处所涉及产品之性能的一般说明和 / 或技术特性。本文件并非用于（也不代替）确定这些产品对于特定用户应用场合的适用性或可靠性。任何此类用户或集成者都有责任就相关特定应用场合或使用方面对产品执行适当且完整的风险分析、评估和测试。**Schneider Electric** 或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）复制本文件的任何部分。

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，只有制造商才能对各个组件进行维修。

当设备用于具有技术安全要求的应用场合时，必须遵守有关的使用说明。

未能使用 **Schneider Electric** 软件或认可的软件配合我们的硬件，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。

© 2014 Schneider Electric。保留所有权利。



安全信息	7
关于本书	9
部分 I SoMachine Basic 快速入门.	15
章 1 SoMachine Basic 简介	17
1.1 系统要求和支持的设备	18
系统要求	19
支持的设备	20
支持的编程语言	21
1.2 SoMachine Basic 用户界面基本介绍	22
使用 SoMachine Basic 创建项目	23
使用 SoMachine Basic 开发程序	24
在 SoMachine Basic 中导航	25
操作模式	26
章 2 SoMachine Basic 入门	27
2.1 起始页	28
起始页简介	29
注册 SoMachine Basic 软件	30
项目窗口	31
连接窗口	32
项目模板窗口	34
直接下载应用程序	35
存储器管理	36
部分 II 开发 SoMachine Basic 应用程序	37
章 3 SoMachine Basic 窗口	39
3.1 SoMachine Basic 窗口概述	40
工具栏按钮	41
状态区域	42
系统设置	44
章 4 属性	47
4.1 属性窗口概述	48
属性窗口	49
项目属性	50

章 5	配置	53
5.1	配置窗口概述	54
	配置窗口概述	55
	构建配置	56
章 6	编程	57
6.1	编程工作空间概述	58
	编程工作空间概述	58
6.2	特殊功能	59
	对象	60
	符号寻址	61
	存储器分配	63
	梯形图 / 列表可转换性	64
	如何使用源代码示例	66
6.3	配置程序行为和任务	69
	应用程序行为	70
	任务和扫描模式	72
6.4	管理 POU	75
	POU	76
	使用任务管理 POU	77
	管理梯级	78
	自由 POU	80
6.5	主任务	82
	主任务描述	83
	配置主任务	84
6.6	周期性任务	86
	创建周期性任务	87
	配置周期性任务扫描持续时间	89
6.7	事件任务	90
	事件任务概述	91
	事件源	92
	事件优先级和队列	93
	创建事件任务	94
6.8	使用工具	96
	程序消息	97
	动态数据表	98
	存储器对象	101
	系统对象	103
	I/O 对象	104

	软件对象	105
	搜索并替换	106
	符号列表	108
	梯级模板	110
6.9	梯形图语言编程	113
	梯形图简介	114
	梯形图的编程原理	116
	梯形图图形元素	117
	比较块	122
	操作块	123
	添加注释	124
	编程最佳做法	125
6.10	指令列表编程	128
	指令列表程序概述	129
	列表指令的操作	131
	列表语言指令	132
	使用括号	135
6.11	Grafcet (列表) 编程	138
	Grafcet (列表) 编程的说明	139
	Grafcet 程序结构	140
	如何在 SoMachine Basic 程序中使用 Grafcet 指令	143
6.12	在线模式下调试	145
	修改值	146
	强制值	147
章 7	试运行	149
7.1	试运行窗口概述	150
	试运行窗口概述	150
7.2	管理 Logic Controller 的连接	151
	连接到 Logic Controller.	152
	控制器信息	154
	管理 RTC.	155
7.3	SoMachine Basic 仿真器	156
	SoMachine Basic 模拟器概述	157
	SoMachine Basic 模拟器 I/O 管理器窗口	159
	SoMachine Basic 模拟器时间管理窗口	161
	使用 SoMachine Basic 模拟器修改值.	163
	如何使用 SoMachine Basic 模拟器.	168

7.4	备份和恢复控制器存储器	169
	备份和恢复控制器存储器	169
7.5	下载和上载程序	171
	下载和上载应用程序	172
	控制器更新	173
章 8	保存项目并关闭 SoMachine Basic	175
	保存项目	176
	项目另存为模板	177
	关闭 SoMachine Basic	178
附录	179
附录 A	SoMachine Basic 键盘快捷方式	181
	SoMachine Basic 键盘快捷方式	181
术语表	187
索引	189



重要信息

声明

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



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



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

危险

“危险”表示极可能存在危险，如果不遵守说明，可导致严重的人身伤害甚至死亡。

警告

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

注意

“注意”表示可能存在危险，如果不遵守说明，可导致严重的人身伤害或设备损坏。

注意

“注意”用于表示与人身伤害无关的危害。

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。Schneider Electric 不承担由于使用本资料所引起的任何后果。

专业人员是指掌握与电气设备的制造和操作及其安装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

本指南介绍如何使用 SoMachine Basic 软件为支持的 Logic Controller 配置、编程和试运行应用程序。

有效性说明

本手册中的信息仅适用于 SoMachine Basic 产品。

本文档已随 SoMachine Basic V1.1 的发布进行了更新。

本文档中描述的设备技术特性在网站上也有提供。要在线访问此信息：

步骤	操作
1	访问 Schneider Electric 主页 www.schneider-electric.com 。
2	在 Search 框中键入产品参考号或产品系列名称。 <ul style="list-style-type: none">● 型号 / 产品系列中不得包括空格。● 要获得有关类似模块分组的信息，请使用星号 (*)。
3	如果您输入参考号，则转到 Product datasheets 搜索结果，单击您感兴趣的参考号。 如果您输入产品系列的名称，则转到 Product Ranges 搜索结果，单击您感兴趣的产品系列。
4	如果 Products 搜索结果中出现多个参考号，请单击您感兴趣的参考号。
5	根据屏幕大小，您可能需要向下滚动查看数据表。
6	要将数据表保存为 .pdf 文件或打印数据表，请单击 Download XXX product datasheet 。

本手册中介绍的特性应该与在线显示的那些特性相同。依据我们的持续改进政策，我们将不断修订内容，使其更加清楚了，更加准确。如果您发现手册和在线信息之间存在差异，请以在线信息为准。

相关的文件

文件名称	参考编号
SoMachine Basic 通用功能 - 库指南	EIO0000001474 (英语) EIO0000001475 (法语) EIO0000001476 (德语) EIO0000001477 (西班牙语) EIO0000001478 (意大利语) EIO0000001479 (简体中文) EIO0000001480 (葡萄牙语) EIO0000001481 (土耳其语)
Modicon M221 Logic Controller - 编程指南	EIO0000001360 (英语) EIO0000001361 (法语) EIO0000001362 (德语) EIO0000001363 (西班牙语) EIO0000001364 (意大利语) EIO0000001365 (简体中文) EIO0000001369 (土耳其语) EIO0000001368 (波兰语)
Modicon M221 Logic Controller - 硬件指南	EIO0000001384 (英语) EIO0000001385 (法语) EIO0000001386 (德语) EIO0000001387 (西班牙语) EIO0000001388 (意大利语) EIO0000001389 (简体中文) EIO0000001370 (葡萄牙语) EIO0000001371 (土耳其语)
Modicon TMC2 扩展板 - 编程指南	EIO0000001782 (英语) EIO0000001783 (法语) EIO0000001784 (德语) EIO0000001785 (西班牙语) EIO0000001786 (意大利语) EIO0000001787 (简体中文) EIO0000001788 (波兰语) EIO0000001789 (土耳其语)

文件名称	参考编号
Modicon TMC2 扩展板 - 硬件指南	EIO0000001768 (英语) EIO0000001769 (法语) EIO0000001770 (德语) EIO0000001771 (西班牙语) EIO0000001772 (意大利语) EIO0000001773 (简体中文) EIO0000001775 (土耳其语) EIO0000001774 (波兰语)
Modicon TM3 扩展模块配置 - 编程指南	EIO0000001396 (英语) EIO0000001397 (法语) EIO0000001398 (德语) EIO0000001399 (西班牙语) EIO0000001400 (意大利语) EIO0000001401 (简体中文) EIO0000001374 (葡萄牙语) EIO0000001375 (土耳其语)
Modicon TM3 数字量 I/O 模块 - 硬件指南	EIO0000001408 (英语) EIO0000001409 (法语) EIO0000001410 (德语) EIO0000001411 (西班牙语) EIO0000001412 (意大利语) EIO0000001413 (简体中文) EIO0000001376 (葡萄牙语) EIO0000001377 (土耳其语)
Modicon TM3 模拟量 I/O 模块 - 硬件指南	EIO0000001414 (英语) EIO0000001415 (法语) EIO0000001416 (德语) EIO0000001417 (西班牙语) EIO0000001418 (意大利语) EIO0000001419 (简体中文) EIO0000001378 (波兰语) EIO0000001379 (土耳其语)

文件名称	参考编号
Modicon TM3 专用模块 - 硬件指南	EIO0000001420 (英语) EIO0000001421 (法语) EIO0000001422 (德语) EIO0000001423 (西班牙语) EIO0000001424 (意大利语) EIO0000001425 (简体中文) EIO0000001380 (葡萄牙语) EIO0000001381 (土耳其语)
Modicon TM3 接收器和发射器模块 - 硬件指南	EIO0000001426 (英语) EIO0000001427 (法语) EIO0000001428 (德语) EIO0000001429 (西班牙语) EIO0000001430 (意大利语) EIO0000001431 (简体中文) EIO0000001382 (葡萄牙语) EIO0000001383 (土耳其语)
Modicon TM2 扩展模块配置 - 编程指南	EIO0000000396 (英语) EIO0000000397 (法语) EIO0000000398 (德语) EIO0000000399 (西班牙语) EIO0000000400 (意大利语) EIO0000000401 (简体中文)
Modicon TM2 数字量 I/O 模块 - 硬件指南	EIO0000000028 (英语) EIO0000000029 (法语) EIO0000000030 (德语) EIO0000000031 (西班牙语) EIO0000000032 (意大利语) EIO0000000033 (简体中文)
Modicon TM2 模拟量 I/O 模块 - 硬件指南	EIO0000000034 (英语) EIO0000000035 (法语) EIO0000000036 (德语) EIO0000000037 (西班牙语) EIO0000000038 (意大利语) EIO0000000039 (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。

 **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。这些关键控制功能包括紧急停止、越程停止、断电重启以及类似的安全措施。
- 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路失效问题加以考虑。
- 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。

不遵循上述说明可能导致人员伤亡或设备损坏。

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。

不遵循上述说明可能导致人员伤亡或设备损坏。

部分 I

SoMachine Basic 快速入门

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
1	SoMachine Basic 简介	17
2	SoMachine Basic 入门	27

章 1

SoMachine Basic 简介

本章包含了哪些内容?

本章包含了以下部分:

节	主题	页
1.1	系统要求和支持的设备	18
1.2	SoMachine Basic 用户界面基本介绍	22

节 1.1

系统要求和支持的设备

本节包含了哪些内容？

本节包含了以下主题：

主题	页
系统要求	19
支持的设备	20
支持的编程语言	21

系统要求

概述

SoMachine Basic 的最低系统要求是：

- Intel Core 2 双核处理器或更高配置
- 1 GB RAM
- 以下操作系统之一的 32 位或 64 位版本：
 - Microsoft Windows XP Service Pack 3
 - Microsoft Windows 7

支持的设备

M221 Logic Controller

有关模块配置的详细信息，请参阅以下编程指南和硬件指南：

Logic Controller 类型	硬件指南	编程指南
M221 Logic Controller	Modicon M221 Logic Controller 硬件指南	Modicon M221 Logic Controller 编程指南

TMC2 扩展板

有关扩展板配置的详细信息，请参阅以下编程指南和硬件指南：

扩展板类型	硬件指南	编程指南
TMC2 扩展板	TMC2 扩展板硬件指南	TMC2 扩展板编程指南

TM3 扩展模块

有关模块配置的详细信息，请参阅下面各个扩展模块类型的编程指南和硬件指南：

扩展模块类型	硬件指南	编程指南
TM3 数字量 I/O 扩展模块	TM3 数字量 I/O 扩展模块硬件指南	TM3 扩展模块编程指南
TM3 模拟量 I/O 扩展模块	TM3 模拟量模块硬件指南	
TM3 专用 I/O 扩展模块	TM3 专用 I/O 模块硬件指南	
TM3 发射器和接收器模块	TM3 接收器和发射器模块硬件指南	

TM2 扩展模块

有关模块配置的详细信息，请参阅各个扩展模块类型的编程指南和硬件指南：

扩展模块类型	硬件指南	编程指南
TM2 数字量 I/O 模块	TM2 数字量 I/O 模块硬件指南	TM2 扩展模块编程指南
TM2 模拟量 I/O 模块	TM2 模拟量 I/O 模块硬件指南	

支持的编程语言

概述

可编程 **Logic Controller** 可根据控制程序读取输入、写入输出及解析逻辑。为 **Logic Controller** 创建控制程序包括使用支持的其中一种编程语言编写一系列指令。

SoMachine Basic 支持以下 **IEC 61131 3** 编程语言：

- 梯形图语言
- 指令列表语言
- **Grafcet**（列表）

节 1.2

SoMachine Basic 用户界面基本介绍

本节包含了哪些内容？

本节包含了以下主题：

主题	页
使用 SoMachine Basic 创建项目	23
使用 SoMachine Basic 开发程序	24
在 SoMachine Basic 中导航	25
操作模式	26

使用 SoMachine Basic 创建项目

概述

SoMachine Basic 是一款图形编程工具，可为 Logic Controller 轻松配置、开发和试运行程序。

一些重要术语

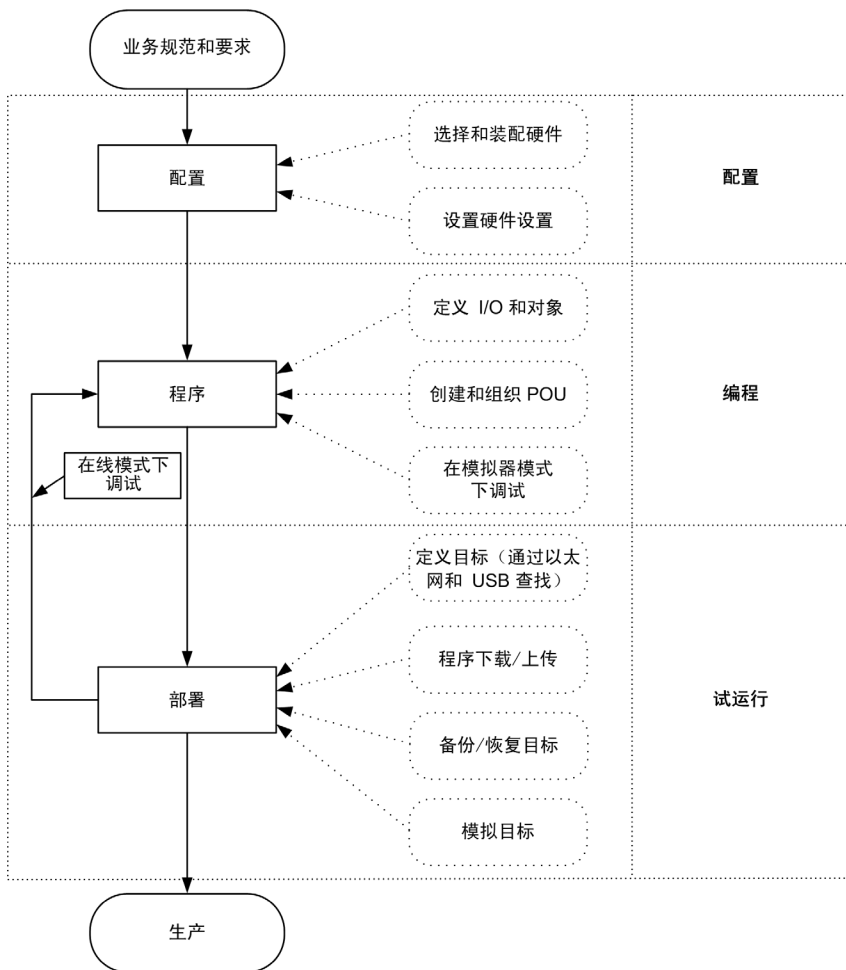
SoMachine Basic 使用以下术语：

- **项目：** SoMachine Basic 项目包含开发者和项目用途的详细信息、项目针对的 Logic Controller 和关联扩展模块的配置、程序的源代码、符号、注释、文档及所有其他相关的信息。
- **应用程序：** 包含下载到 Logic Controller 的项目的所有部分，包括已编译程序、元数据、配置信息，以及符号。
- **程序：** 在 Logic Controller 中运行的已编译源代码。
- **POU**（程序组织单元）：可以重复使用的对象，包括在程序中使用的变量声明以及指令集。

使用 SoMachine Basic 开发程序

简介

下图显示了在 SoMachine Basic 中开发项目的典型阶段（配置、编程和试运行选项卡）：



在 SoMachine Basic 中导航

起始页

启动 SoMachine Basic 时始终显示**起始页**窗口。使用此窗口可注册 SoMachine Basic 软件、管理 Logic Controller 的连接，以及创建或选择要使用的项目。

模块区域

选择要处理的项目后，SoMachine Basic 将显示主窗口。

在主窗口顶部，工具栏 (参见第 41 页) 包含一些图标，可用于执行常见任务，包括返回**起始页**窗口。

工具栏旁的状态栏 (参见第 42 页) 会显示关于 Logic Controller 当前连接状态的信息。

状态栏下方的主窗口分为多个**模块**。每个模块控制开发周期的不同阶段，可以通过单击模块区域顶部的选项卡进行访问。要开发应用程序，请按从左向到右的顺序使用模块：

- **属性** (参见第 47 页)
设置项目属性
- **配置** (参见第 53 页)
定义 Logic Controller 和关联扩展模块的硬件配置
- **编程** (参见第 57 页)
使用其中一种支持的编程语言开发程序
- **试运行** (参见第 149 页)
管理 SoMachine Basic 与 Logic Controller 之间的连接、上传 / 下载应用程序、测试和试运行应用程序。

操作模式

简介

SoMachine Basic 可以在离线或在线模式下运行。满足以下条件时， SoMachine Basic 可以在在线模式下运行：

- PC 已经物理连接到 Logic Controller。
- SoMachine Basic 正在模拟虚拟 Logic Controller（称为模拟器模式）。

离线模式

如果尚未与 Logic Controller 建立物理连接，则 SoMachine Basic 将以离线模式运行。

在离线模式下，您可以配置 SoMachine Basic 以匹配目标硬件组件，然后开发您的应用程序。

在线模式

如果已与 Logic Controller 建立物理连接，则 SoMachine Basic 将以在线模式运行。

处于在线模式后，您可以继续将应用程序下载到 Logic Controller。SoMachine Basic 随后会将 PC 存储器中的应用程序与 Logic Controller 中存储的版本同步，以便调试、监控和修改应用程序。

无法修改处于在线模式中的程序。

注意：在线程序修改受到预定义配置的限制。请参阅存储器管理 (参见第 36 页)。此外，有关详细信息，请参阅在线模式下调试 (参见第 145 页)。

模拟器模式

如果已与模拟的 Logic Controller 建立连接，则 SoMachine Basic 将以模拟器模式运行。在模拟器模式中，不会与 Logic Controller 建立任何物理连接；相反， SoMachine Basic 会模拟到 Logic Controller 和扩展模块的连接，以便运行和测试程序。

有关详细信息，请参阅 SoMachine Basic 模拟器 (参见第 156 页)。

章 2

SoMachine Basic 入门

节 2.1

起始页

本节包含了哪些内容？

本节包含了以下主题：

主题	页
起始页简介	29
注册 SoMachine Basic 软件	30
项目窗口	31
连接窗口	32
项目模板窗口	34
直接下载应用程序	35
存储器管理	36

起始页简介

概述

“起始页”窗口始终是启动 SoMachine Basic 时显示的第一个窗口。

“起始页”窗口具有以下窗口：

- **寄存器** (参见第 30 页)
注册 SoMachine Basic 软件并查看许可证详细信息。
- **项目** (参见第 31 页)
创建新项目或打开现有项目。
- **连接** (参见第 32 页)
连接到 Logic Controller、对控制器下载 / 上传应用程序或者从控制器下载 / 上传应用程序、备份 / 恢复控制器存储器，使控制器的 LED 闪烁。
- **模板** (参见第 34 页)
使用示例项目作为模板创建新项目。
- **帮助**
显示在线帮助。
- **关于**
显示有关 SoMachine Basic 的信息。
- **退出**
退出 SoMachine Basic。

注册 SoMachine Basic 软件

概述

您可以免费使用 SoMachine Basic 软件 30 天，然后就需要注册该软件。注册后，您将收到授权代码以使用该软件。

注册 SoMachine Basic 软件后，您便可以获得技术支持和软件更新。

注册

要注册 SoMachine Basic 软件，请执行以下步骤：

步骤	操作
1	单击 起始页 窗口顶部的 立即注册 按钮。
2	按照“注册向导”上的说明操作。有关详细信息，请单击 帮助 按钮。

要查看 PC 上安装的许可证密钥的详细信息，请单击**起始页**窗口中的**关于**。

项目窗口

概述

使用**项目**窗口可打开要使用的现有 SoMachine Basic、TwidoSoft 或 TwidoSuite 项目。

项目窗口的右侧区域包含其他有用信息的链接。

打开 SoMachine Basic 项目文件

按照以下步骤来打开项目文件：

步骤	操作
1	单击 起始页 窗口中的 项目 。
2	执行下列操作之一： <ul style="list-style-type: none"> ● 单击最近的项目列表中的一个最近项目。 ● 单击创建新项目 ● 单击打开现有项目，然后选择现有 SoMachine Basic 项目文件 (*.smbp) 或示例项目文件 (*.smbe)。 随即会打开所选项目文件并显示 配置 选项卡。

打开 TwidoSuite 或 TwidoSoft 项目文件

SoMachine Basic 可打开为 Twido 可编程控制器创建的应用程序，并将其转换为 SoMachine Basic 项目文件。

按照以下步骤来打开 TwidoSuite 或 TwidoSoft 项目文件：

步骤	操作
1	单击 起始页 窗口中的 项目 。
2	单击 打开现有项目 ，接着在 文件类型 列表中选择以下任意内容，然后通过相应的扩展名浏览并选择现有项目： <ul style="list-style-type: none"> ● TwidoSuite 项目文件 (*.xpr) ● Twido 存档项目文件 (*.xar) ● TwidoSoft 项目文件 (*.twd) 随即会打开所选项目文件并显示 配置 选项卡。

连接窗口

连接的设备

连接窗口显示 2 个 Logic Controller 列表：

1. 本地设备

显示通过 PC 的物理 COM 端口（例如 COM1）、USB 电缆或虚拟化 COM 端口（通过 USB 串行转换器或蓝牙适配器）直接连接到 PC 的所有 Logic Controller。

2. 以太网设备

显示可在与运行 SoMachine Basic 的 PC 相同的以太网子网上访问的所有 Logic Controller。不会列出路由器后面的设备或阻止 UDP 广播的任何设备。


该列表包括 SoMachine Basic 自动检测到的 Logic Controller，以及您手动添加的任何控制器。


手动添加控制器

要将 Logic Controller 添加到以太网设备列表中，请执行以下步骤：

步骤	操作
1	在远程查找字段中，输入要添加的 Logic Controller 的 IP 地址，例如 12.123.134.21
2	单击添加，将该设备添加到以太网设备列表中。

连接到控制器

使用 （刷新按钮）来刷新连接的设备的列表。

使用 （启动闪烁的 LED 按钮）使所选控制器的 LED 闪烁，以便通过该控制器闪烁的 LED 来对其进行物理识别。再次单击此按钮来停止闪烁的 LED。

按照以下步骤来将控制器连接到 SoMachine Basic:

步骤	操作
1	在 本地设备 或 以太网设备 列表选择一个 Logic Controller。
2	单击 登录 按钮来登录到所选的控制器。 如果 Logic Controller 受密码保护, 请键入密码并单击 确定 进行连接。 随即显示状态栏, 其中显示连接进度。
3	成功建立连接后, Logic Controller 的详细信息会显示在窗口的 选择的控制器 区域中, 并且以下按钮可用: <ul style="list-style-type: none"> ● Download application to controller: 将应用程序下载到 Logic Controller, 而且不会在 SoMachine Basic 中将其打开。请参阅直接下载应用程序 (参见第 35 页)。 ● 启动闪烁的 LED: 启动 / 停止连接的控制器的闪烁的 LED。 ● 存储器管理 向 / 从 PC 备份 (参见第 169 页) 或恢复 (参见第 170 页) Logic Controller 存储器。请参阅存储器管理 (参见第 36 页)。 ● Upload application from controller: 从连接的 Logic Controller 上传应用程序, 从而创建新的 SoMachine Basic 项目文件。请参阅上传应用程序 (参见第 172 页)。
4	单击 注销 按钮以注销连接的控制器。

项目模板窗口

概述

您可以使用示例项目作为新 SoMachine Basic 项目的基础。

打开项目模板

按照以下步骤来根据项目模板创建新项目：

步骤	操作
1	选择起始页窗口上的 模板 选项卡。
2	单击 示例项目 列表中的条目，或单击 打开模板 并浏览以选择项目模板文件 (*.smbe)。 结果： 将创建新项目作为所选模板的副本。

直接下载应用程序

概述

您可以将项目文件中包含的应用程序下载到 **Logic Controller**，且无需在 **SoMachine Basic** 中打开项目。此功能适用于以下情况：项目在 **仅下载模式** 中受保护，如果用户没有密码，则无法打开项目。

采用这种方式只能进行下载。要将应用程序从 **Logic Controller** 上载到 **SoMachine Basic**，请参阅上载应用程序 (参见第 172 页)。

直接下载应用程序

要将应用程序直接下载到 **Logic Controller**，请执行以下操作：

步骤	操作
1	使用串行、USB 或以太网电缆，将运行 SoMachine Basic 的 PC 物理连接到 Logic Controller 。
2	选择“起始页”窗口上的 连接 选项卡。
3	在 本地设备 或 以太网设备 列表中选择 Logic Controller ，然后单击 登录 。 结果： SoMachine Basic 建立与 Logic Controller 的连接。
4	单击 Download application to controller 。
5	在 项目文件 字段中，单击“浏览”按钮，选择要下载的 SoMachine Basic 项目文件 (*.smbp)，然后单击 打开 。 有关所选项目文件的信息将显示在窗口的 信息 区域中： <ul style="list-style-type: none"> ● 项目文件是否受密码保护，如果受密码保护，是否同时允许查看和下载，或仅下载。 ● 项目文件中包含的配置信息，例如，检测到的 Logic Controller 系统配置是否与所选项目中包含的配置兼容。
6	SoMachine Basic 会编译所选项目文件中的应用程序。 编译错误 下会列出编译过程中检测到的任何错误。如果检测到编译错误，则 SoMachine Basic 不允许下载应用程序；在 SoMachine Basic 中打开项目，更正错误，然后重试。
7	下载之前，您可以单击下列按钮来控制当前 Logic Controller 状态： <ul style="list-style-type: none"> ● 停止控制器 ● 启动控制器 ● 初始化控制器
8	单击 PC 到控制器（下载） 。 结果： SoMachine Basic 将应用程序下载到连接的 Logic Controller 。

存储器管理

概述

单击**连接**窗口中的**存储器管理**按钮，以备份或恢复 Logic Controller 存储器。

选择要执行的操作：

- 备份到 PC (参见第 169 页)
- 从 PC 恢复 (参见第 170 页)

部分 II

开发 SoMachine Basic 应用程序

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
3	SoMachine Basic 窗口	39
4	属性	47
5	配置	53
6	编程	57
7	试运行	149
8	保存项目并关闭 SoMachine Basic	175

章 3

SoMachine Basic 窗口

节 3.1

SoMachine Basic 窗口概述

本节包含了哪些内容?

本节包含了以下主题:

主题	页
工具栏按钮	41
状态区域	42
系统设置	44

工具栏按钮

简介

工具栏显示在 SoMachine Basic 窗口的顶部，可以轻松访问常用功能。

工具栏

SoMachine Basic 具有以下工具栏：

图标	描述
	创建新项目
	打开现有项目
	保存当前项目。向下箭头可显示包含 保存 和 另存为 选项的菜单。
	剪切
	复制
	粘贴
	显示系统设置 (参见第 44 页) 窗口。
	显示此在线帮助。
	启动 Logic Controller (仅在在线模式下且控制器已不在“运行”状态时可用)
	停止 Logic Controller (仅在在线模式下且控制器处于“运行”状态时可用)
	重新初始化 Logic Controller (仅在在线模式下可用)
	登录到所选的控制器。 注意： 所选的控制器名称会显示在该按钮的左侧。
	启动 SoMachine Basic 模拟器 (参见第 156 页)。

状态区域

概述

主窗口顶部的状态区域显示关于当前系统的状态和信息。

下图显示了 SoMachine Basic 中的状态区域：



1...5 状态区域。

1 程序状态：

显示程序的状态：“程序检测到错误”、“没有检测到任何错误”，或者“程序未完成”。

2 连接状态：

显示 SoMachine Basic 和控制器之间的连接状态，或者显示 SoMachine Basic 和模拟 Logic Controller 之间的连接状态。

3 控制器状态：

显示 Logic Controller 的当前工作状态（“运行”、“停止”、“故障”等）。

4 扫描时间：

显示上次扫描时间。

5 上一个控制器错误：

如果 Logic Controller 处于“已停止”或“故障”状态，则显示最近一次检测到的错误消息。

状态区域消息

以下消息可能显示在状态区域中：

消息类型	消息	描述
程序状态	[该程序发生错误]	没有程序，或者程序检测到错误。
	[无错误]	未检测到错误。
	[建议]	程序未完成。
连接状态	[离线]	SoMachine Basic 正在离线模式下运行。
	[在线]	SoMachine Basic 正在在线模式下运行。
控制器状态	[停止]	Logic Controller 的当前工作状态（“运行”、“停止”、“故障”等）。只有 SoMachine Basic 处于在线模式时才显示。

消息类型	消息	描述
扫描时间	[0 μs]	最近一次扫描时间（以毫秒为单位）。 只有 SoMachine Basic 处于在线模式时才显示。
上一个控制器错误	[未检测到错误]	如果 Logic Controller 处于 HALT (暂停) 状态，则会从系统位和系统字中提取最新的错误消息。 有关系统位和系统字的完整列表，请参阅 Logic Controller 的 <i>编程指南</i> 。 只有 SoMachine Basic 处于在线模式时才显示。

系统设置



概述

此窗口可用于设置 SoMachine Basic 软件的语言、自定义梯形图编辑器，以及选择在创建新项目时，**配置**选项卡上显示的缺省 Logic Controller。

更改用户界面语言

步骤	操作
1	在“系统设置”窗口中选择 系统设置 → 常规 。
2	在 语言 列表中选择要使用语言。
3	单击 应用 。
4	关闭并重新启动 SoMachine Basic 可使用新语言查看用户界面。

自定义梯形图编辑器

步骤	操作
1	在“系统设置”窗口中选择 系统设置 → 梯形图编辑器 。
2	选择梯形图编辑器的 网格线样式 。 <ul style="list-style-type: none"> ● 点（缺省） ● 虚线 ● 线
3	为梯形图编辑器中的单元格设置 列数 (11...30)。 在缺省情况下，编辑器中包含 11 个单元格。 有关详细信息，请参阅梯形图的编程原理 (参见第 116 页)。
4	在 工具选择保护 下，选择： <ul style="list-style-type: none"> ● 保留选择工具：选将所选指令放入梯级后，最近选择的指令会保持选中状态。这样便可以将相同指令重复放入梯级中，而无需重新选择该指令。按 Esc 键或右键单击梯级中的空白单元格，以选择指针工具 。 ● 复位到指针（缺省）：选择并将触点或线圈放入梯级中后，便会自动选择指针工具 。要重新插入相同的触点或线圈元素，请在工具栏中选择它。

步骤	操作
5	<p>选择梯形图编辑器的快捷方式和工具栏样式设置：</p> <ul style="list-style-type: none"> ● SoMachine Basic set（缺省） ● Asian set 1 ● Asian set 2 ● European set ● American set <p>表格将显示所选样式的快捷键列表。</p>
6	单击 应用 并关闭 系统设置 窗口，以查看梯形图编辑器中的更改。

选择缺省 Logic Controller

步骤	操作
1	在“系统设置”窗口中选择 系统设置 → 配置 。
2	单击 首选控制器 ，然后从列表中选择缺省 Logic Controller。
3	单击 应用 。

章 4

属性

节 4.1

属性窗口概述

本节包含了哪些内容？

本节包含了以下主题：

主题	页
属性窗口	49
项目属性	50

属性窗口

概述

属性选项卡用于指定项目的信息以及项目是否受密码保护：

- 有关开发者和项目开发公司的详细信息。
- 项目本身的信息。
- 如果项目受密码保护，则必须正确输入密码才能在 **SoMachine Basic** 中打开该项目。
- 如果存储在 **Logic Controller** 中的应用程序受密码保护，则必须正确输入密码才能将应用程序上载到 **SoMachine Basic** 项目中。

1 左侧区域显示可用的属性列表。

2 右侧区域显示当前在左侧区域中所选项目的属性。

- 1 左侧区域显示可用的属性列表。
- 2 右侧区域显示当前在左侧区域中所选项目的属性。

项目属性

概述

属性窗口用于提供有关 SoMachine Basic 的用户、开发应用程序的公司和项目的详细信息。在此窗口中，还可以在将项目文件和应用程序存储在 **Logic Controller** 中时对其进行密码保护。

指定应用程序开发者属性

要指定应用程序开发者属性，请执行以下步骤：

步骤	操作
1	显示 属性 选项卡，然后单击 项目属性 → 首页 。
2	填写有关信息。
3	单击 应用 。

注意： 右键单击 SoMachine Basic 项目文件时，此信息会显示在 Windows 资源管理器属性窗口中。

指定公司属性

要指定公司属性，请执行以下步骤：

步骤	操作
1	显示 属性 选项卡，然后单击 项目属性 → 公司 。
2	填写有关信息。
3	单击 应用 。

指定项目信息

要指定项目信息，请执行以下步骤：

步骤	操作
1	显示 属性 选项卡，然后单击 项目属性 → 项目信息 。
2	填写有关信息。必填信息项用星号 (*) 标记。 要上载图像（如仪表化机器的照片或 CAD 图像），请单击 更改 ，然后浏览并选择要上载的文件。单击 删除 以删除当前图像。
3	单击 应用 。

使用密码保护项目

可以保护项目文件。如果项目受密码保护，则每次在 SoMachine Basic 中打开该项目时都会提示您输入密码。

步骤	操作
1	显示 属性 选项卡，然后单击 项目属性 → 项目保护 。
2	选择 活动 选项。
3	在 密码 字段中输入要使用密码，然后在 确认 字段中重新输入该密码。
4	选择以下选项之一： <ul style="list-style-type: none"> ● 查看并下载：允许在不知悉密码的情况下查看应用程序的内容并将应用程序下载到 Logic Controller。但是，您必须输入密码才能修改应用程序的内容。 ● 仅下载：您可以在不知悉密码的情况下将应用程序下载到 Logic Controller。通过起始页（参见第 32 页）窗口中的试运行按钮进行下载。在打开项目以查看或修改应用程序时，您必须输入正确的密码。
5	单击 应用 。

从项目中删除密码保护

步骤	操作
1	显示 属性 选项卡，然后单击 项目属性 → 项目保护 。
2	选择 非活动 选项。
3	单击 应用 。

使用密码保护应用程序

SoMachine Basic 允许使用密码来保护存储在 Logic Controller 中的应用程序。此密码可控制将应用程序从 Logic Controller 上载到 SoMachine Basic 项目。

要设置应用程序密码，请执行以下步骤：

步骤	操作
1	显示 属性 选项卡，然后单击 项目属性 → 应用程序保护 。
2	选择应用程序保护级别： <ul style="list-style-type: none"> ● 选中活动并将密码留空，可禁用应用程序从 Logic Controller 到 PC 的所有上载操作。 ● 选中活动并在密码和确认字段中输入相同的密码，将使用密码来保护应用程序。系统提示时，用户必须输入此密码，然后才能将应用程序从 Logic Controller 上载到 PC。 ● 选择非活动将不会保护应用程序。
3	单击 应用 。

章 5

配置

节 5.1

配置窗口概述

本节包含了哪些内容？

本节包含了以下主题：

主题	页
配置窗口概述	55
构建配置	56

配置窗口概述

简介

使用配置窗口可对应用程序对应的 Logic Controller 及其扩展模块重新进行硬件配置。

1 硬件树 -- 当前硬件配置的结构化视图。

2 当前配置 -- Logic Controller 和扩展模块。

3 所有支持的 Logic Controller 和扩展模块硬件组件的目录参考。要将组件添加到当前硬件配置，请将其拖放到当前配置中的插槽。

4 当前配置中所选组件的属性，或硬件树中当前所选项目的属性。

构建配置

替换缺省 Logic Controller

创建新 SoMachine Basic 项目后，Logic Controller 的型号会显示在**配置**窗口的中心区域。

步骤	操作
1	单击 配置 选项卡。
2	如果尚未显示，请在右侧的目录区域中展开 Logic Controller 类别。
3	选择一个 Logic Controller 型号。Logic Controller 物理属性的简短描述会显示在 设备描述 区域中。
4	将 Logic Controller 的型号拖放到窗口中心区域中的现有 Logic Controller 图像上。
5	在系统提示确认替换 Logic Controller 型号时，单击 是 。

注意：在 [**系统设置窗口** 中指定缺省控制器型号。 (参见第 44 页)]

配置 Logic Controller

使用**配置**窗口可配置 Logic Controller。

有关详细信息，请参阅配置中所用 Logic Controller 的**编程指南**。

配置扩展模块

使用**配置**窗口可添加和配置扩展模块。

有关详细信息，请参阅配置中所用扩展模块的**编程指南**。

章 6

编程

本章包含了哪些内容？

本章包含了以下部分：

节	主题	页
6.1	编程工作空间概述	58
6.2	特殊功能	59
6.3	配置程序行为和任务	69
6.4	管理 POU	75
6.5	主任务	82
6.6	周期性任务	86
6.7	事件任务	90
6.8	使用工具	96
6.9	梯形图语言编程	113
6.10	指令列表编程	128
6.11	Grafcet（列表）编程	138
6.12	在线模式下调试	145

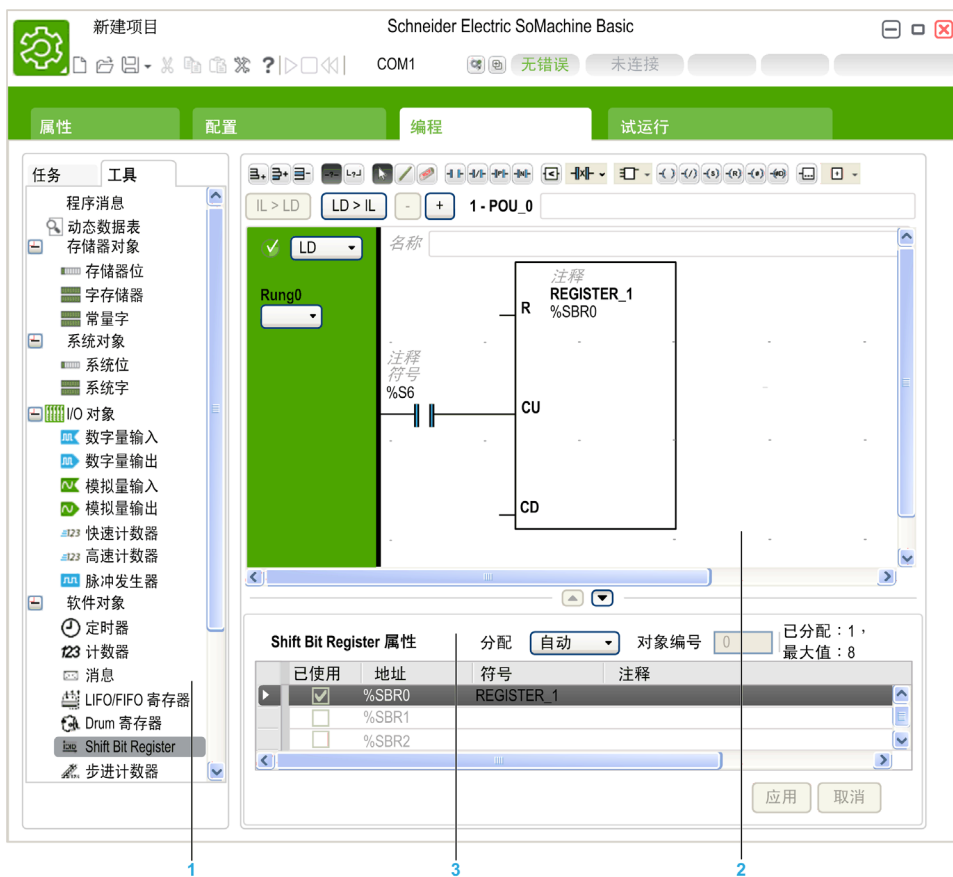
节 6.1

编程工作空间概述

编程工作空间概述

概述

编辑选项卡分为 3 个主要区域：



- 1 编程树用于配置程序、程序对象和功能的属性，以及可用来监控和调试程序的一些工具。
- 2 中心区域上方是编程工作空间，您可以在其中输入程序的源代码。
- 3 中心区域下方用于查看和配置当前在程序或编程树中选择的项目的属性。

节 6.2

特殊功能

本节包含了哪些内容？

本节包含了以下主题：

主题	页
对象	60
符号寻址	61
存储器分配	63
梯形图 / 列表可转换性	64
如何使用源代码示例	66

对象

概述

在 SoMachine Basic 中，*对象*一词用于表示保留供应用程序使用的 Logic Controller 存储器的区域。对象可能是：

- 简单软件变量，例如存储器位和字
- 数字量或模拟量输入和输出的地址
- 控制器内部变量，例如系统字和系统位
- 预定义的系统功能或功能块，例如定时器和计数器。

将为某些对象类型预先分配控制器存储器，或者在应用程序下载到 Logic Controller 时自动分配。

分配存储器后，对象只能由程序进行寻址。使用前缀 % 对对象进行寻址。例如，%MW12 是存储器字的地址，%Q0.3 是嵌入式数字量输出的地址，而 %TMO 是 Timer 功能块的地址。

符号寻址

简介

SoMachine Basic 支持语言对象的符号寻址，即按名称间接寻址对象。使用变量可以快速检查并分析程序逻辑，并且大大简化了应用程序的开发和测试。

示例

例如，WASH_END 符号可用于识别 Timer 功能块的一个实例，表示洗涤循环结束。回想此名称的用意比尝试记住程序地址（如 %TM3）的作用更容易。

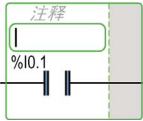
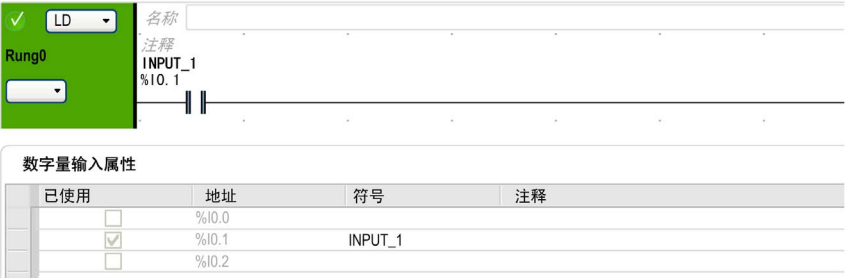
在属性窗口中定义符号

要在属性窗口中定义符号，请执行以下操作：

步骤	操作
1	选择 编程 窗口左侧区域的 工具 选项卡。
2	选择要定义符号的对象类型，例如 I/O 对象 → 数字量输入 ，以显示数字量输入的属性。对象类型的属性窗口便会显示在 编程 窗口的中心区域下方。
3	双击属性表的 符号 列，然后输入要为特定项目定义的符号，例如为输入 %I0.2 定义 Input_1 
4	单击 应用 。

在梯形图编辑器中定义符号

要在梯形图编辑器中定义符号，请执行以下操作：

步骤	操作																				
1	<p>在梯形图编辑器中，单击图形元素的符号行，例如锁存或功能块。随即显示光标：</p> 																				
2	<p>输入要使用的符号，例如 Input_1，然后按 Enter。以下规则适用于符号：</p> <ul style="list-style-type: none"> ● 最多 32 个字符。 ● 字母 (A-Z)、数字 (0-9) 或下划线 ()。 ● 第一个字符必须是字母。不能使用百分比符号 (%)。 ● 符号不区分大小写。例如， Pump1 和 PUMP1 是相同的符号，而且只能用于一个特定的对象，即无法将相同的符号分配给不同的对象。 																				
3	<p>如果图形元素尚未与对象关联，则会显示注释窗口。选择要与符号关联的对象，然后单击确定。或者在提示将符号与对象关联时，单击是。</p>																				
4	<p>双击图形元素的符号或对象，从而在属性窗口的符号列中显示符号：</p>  <table border="1"> <thead> <tr> <th colspan="4">数字量输入属性</th> </tr> <tr> <th>已使用</th> <th>地址</th> <th>符号</th> <th>注释</th> </tr> </thead> <tbody> <tr> <td><input type="checkbox"/></td> <td>%I0.0</td> <td></td> <td></td> </tr> <tr> <td><input checked="" type="checkbox"/></td> <td>%I0.1</td> <td>INPUT_1</td> <td></td> </tr> <tr> <td><input type="checkbox"/></td> <td>%I0.2</td> <td></td> <td></td> </tr> </tbody> </table>	数字量输入属性				已使用	地址	符号	注释	<input type="checkbox"/>	%I0.0			<input checked="" type="checkbox"/>	%I0.1	INPUT_1		<input type="checkbox"/>	%I0.2		
数字量输入属性																					
已使用	地址	符号	注释																		
<input type="checkbox"/>	%I0.0																				
<input checked="" type="checkbox"/>	%I0.1	INPUT_1																			
<input type="checkbox"/>	%I0.2																				

在指令列表代码中显示符号

在显示梯级的指令列表代码时，选中**符号**复选框，可显示代码中任何定义的符号，而不是直接对象引用。

显示所有定义的符号

选择**工具** → **符号列表**以显示所有定义的符号列表 (参见第 108 页)。

存储符号

符号将存储在 Logic Controller 中，作为 SoMachine Basic 应用程序的一部分。

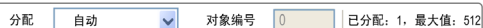
存储器分配

简介

SoMachine Basic 允许您预先分配（保留）Logic Controller 存储器的数据块，以供程序中的特定对象类型使用，包括简单对象（存储器字、常量字）和软件对象（功能块）。

分配模式

在离线模式下，您可以为每个对象类型指定存储器分配模式。在配置这些对象（编程 → 工具）时，以下窗口随之出现在可配置对象列表的上方：



选择要使用的存储器分配模式：

- **自动。** 程序中使用的从偏移 0 到最高存储器地址，或与符号关联的所有对象都会在 Logic Controller 存储器中自动分配。例如：如果程序中使用了存储器字 %MW20，则从 %MW0 至 %MW20（首尾包括在内，共 21 个对象）的所有对象都在存储器中自动分配。如果随后切换到在线模式，则无法分配地址高于进入在线模式之前所用最高地址的新存储器对象。
- **手动。** 在**对象数**框中指定要在存储器中分配的对象数。切换到在线模式后，您可以在程序中添加新触点、线圈或等式（不超过分配的存储器限制），而不需要从 Logic Controller 注销，修改程序，登录，然后重新下载应用程序。

SoMachine Basic 会显示**已分配**存储器对象的总数及 Logic Controller 中可用的**最大**存储器对象数。

梯形图 / 列表可转换性

简介

SoMachine Basic 支持将梯级从梯形图转换为指令列表，以及从指令列表转换回梯形图。这称为 *程序可转换性*。

在 SoMachine Basic 中，您可以根据需要随时在编程语言之间切换梯级。因此，您可以在显示程序时，一些梯级以梯形图显示，而另一些梯级以指令列表显示。

理解可转换性

理解程序可转换性的关键在于检查梯形图梯级与关联的指令列表梯级之间的关系：

- **梯形图梯级**：构成逻辑表达式的梯形图指令集合。
- **列表序列**：与梯形图指令对应并代表相同逻辑表达式的指令列表编程指令集合。

下图显示了常见梯形图梯级及其表示为指令列表指令序列的等效程序逻辑。



等效的指令列表指令：

名称	指令	地址	注释
0000	LD	%I0.5	注释
0001	OR	%I0.4	注释
0002	ST	%Q0.4	注释

无论最初是以梯形图还是以指令列表语言编写程序，程序始终在内部存储为指令列表指令。SoMachine Basic 利用这两种语言之间的程序结构相似性，并使用程序的此内部指令列表映像，将其显示为指令列表程序，或以图形方式显示为梯形图。

可转换性所需指令

指令列表语言中使用功能块需要使用以下指令：

- BLK 标示功能块的开始，并定义梯级的开始以及功能块的输入部分的开始。
- OUT_BLK 标示功能块的输出部分的开始。
- END_BLK 标示功能块和梯级的结束。

并非必须使用这些可转换功能块指令，指令列表程序才能正常运行。

如何使用源代码示例



概述

除非明确提出，否则本手册中包含的源代码示例对梯形图和指令列表编程语言均有效。整个示例可能需要多个梯级。

可转换性过程

本手册中仅显示指令列表源代码。

要获得对应的梯形图源代码，请执行以下操作：

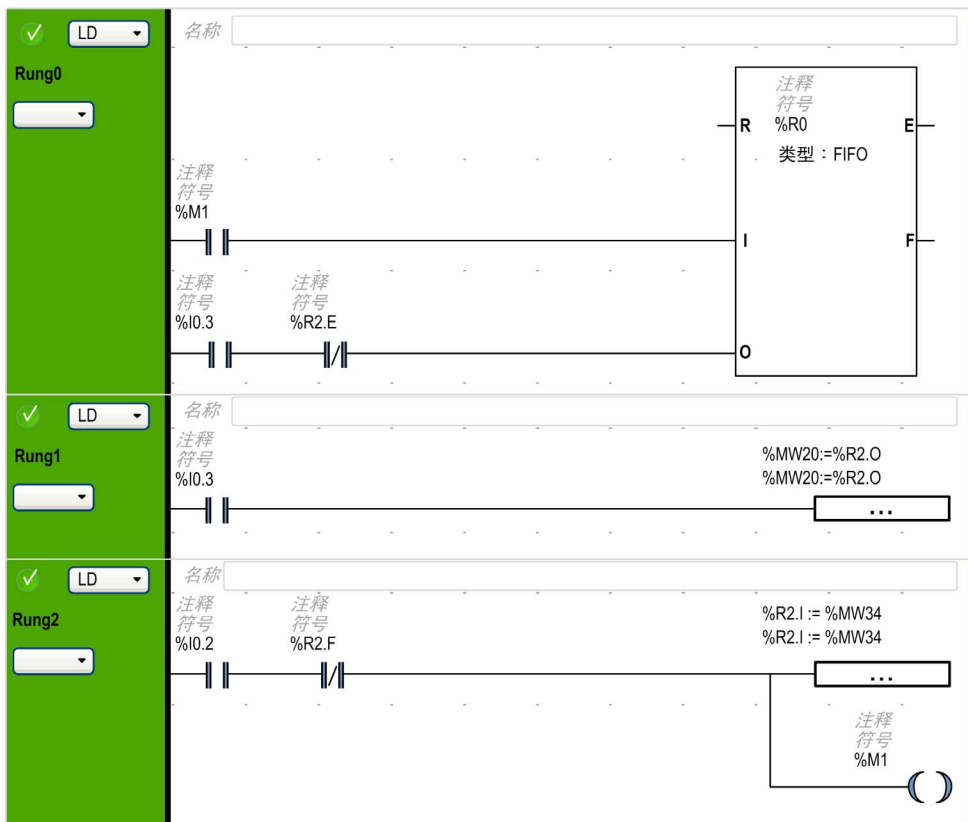
步骤	操作
1	在 SoMachine Basic 中，创建一个包含空梯级的新 POU。
2	在此梯级中，单击 LD > IL 按钮以显示指令列表源代码。
3	选择并复制 (Ctrl+C) 示例程序的第一个梯级的源代码。
4	<p>右键单击第一个指令的行号 0000，然后选择粘贴指令，将源代码粘贴到梯级：</p>  <p>注意： 如果通过在缺省 LD 操作符之前插入行来粘贴指令，那么务必从梯级的最后一行中删除 LD 指令。</p>
5	单击 IL > LD 按钮以显示梯形图源代码。
6	<p>对示例程序中的任何其他梯级重复上述步骤。单击工具栏上的  以添加新梯级。</p>

示例

指令列表程序:

梯级	源代码
0	BLK %R0 LD %M1 I LD %I0.3 ANDN %R2.E O END_BLK
1	LD %I0.3 [%MW20:=%R2.O]
2	LD %I0.2 ANDN %R2.F [%R2.I:=%MW34] ST %M1

相应的梯形图:



节 6.3

配置程序行为和任务

本节包含了哪些内容？

本节包含了以下主题：

主题	页
应用程序行为	70
任务和扫描模式	72

应用程序行为

概述

您可以配置应用程序与 Logic Controller 进行交互的以下方面：

- **功能级别** (参见第 70 页)
- **启动** (参见第 70 页)
- **看门狗** (参见第 71 页)
- **故障预置行为** (参见第 71 页)

配置应用程序行为

步骤	操作
1	选择 编程 窗口左侧区域的 任务 选项卡。
2	选择 行为 项目。 行为 属性便会显示在 编程 窗口的中心区域下方。
3	根据需要修改属性。

功能级别

系统可能包含具有不同固件版本的 Logic Controller，因此具有不同的功能级别。SoMachine Basic 支持功能级别管理，可让您控制应用程序的功能级别。

选择**自动管理**（缺省值），然后选择以下设置之一：

- **最高功能级别**（缺省值）。此设置可让您获得最多的 Logic Controller 功能。
- **最低功能级别**。可让您使用满足您需求的最低级别的功能。

否则，选择**手动管理**，然后选择要使用的固件级别，例如**级别 1.0**。

启动

指定在 Logic Controller 重新启动后程序的行为方式。

- **在先前状态下启动**：程序在其重新启动前所处的执行模式下启动。
- **在“停止”状态下启动**：Logic Controller 不在自动应用程序执行模式下启动。
- **在“运行”状态下启动**（缺省值）：Logic Controller 在自动应用程序执行模式下启动。

使用“自动开始运行”时，控制器将在设备通电时执行程序逻辑。预先了解输出的自动重新激活方式如何影响过程或所控制的机器，这一点至关重要。配置运行 / 停止输入可帮助控制“自动开始运行”功能。此外，运行 / 停止输入旨在提供对远程运行命令的本地控制。如果在控制器由 SoMachine 本地停止后发出的远程运行命令可能具有意外后果，则必须配置和连接运行 / 停止输入，以帮助控制这种情况。

 **警告****机器意外启动**

- 使用“自动开始运行”设置之前，请确认自动重新激活输出未造成意外的后果。
- 使用运行/停止输入有助于避免在“运行”模式下意外重新启动，且可帮助防止从远程位置意外启动。
- 在对运行/停止输入加电之前或在从远程位置发出运行命令之前，请检查机器或过程环境的安全状态。

不遵循上述说明可能导致人员伤亡或设备损坏。

 **警告****机器或过程意外启动**

- 在对运行/停止输入加电之前，请检查机器或过程环境的安全状态。
- 使用运行/停止输入可帮助防止从远程位置意外启动。

不遵循上述说明可能导致人员伤亡或设备损坏。

看门狗

看门狗是特殊的定时器，用于确保程序不会超出其分配的扫描时间。

看门狗定时器的缺省值为 250 毫秒。指定看门狗扫描任务的持续时间。可能的范围为 10...500 毫秒。

故障预置行为

指定 Logic Controller 因任何原因进入“停止”或“例外”状态时要使用的故障预置模式。

有两种故障预置模式：

- 缺省情况下，所有输入设置为在嵌入式 Logic Controller 和扩展模块输出的配置属性中设置的故障预置值。
有关为输出配置故障预置值的信息，请参阅 Logic Controller 或扩展模块的 *编程指南*。
- 选择**维护值**，在 Logic Controller 进入“停止”或“例外”状态时使每个输出保持其当前状态。在此模式下，将忽略为 Logic Controller 和扩展模块输出配置的任何故障预置值。

任务和扫描模式

概述

具有以下扫描模式：

- **正常模式**
连续循环扫描模式（自由运行模式），前一个扫描完成后立即开始新的扫描。
- **周期模式**
周期性循环扫描模式，仅在经过前一个扫描的配置的扫描时间之后开始新的扫描。因此，每次扫描的持续时间都相同。

提供以下任务类型：

- **主任务**：应用程序的主任务。
对于主任务，通过连续循环扫描（在正常扫描模式中）、软件时间（在周期性扫描模式中），或者指定扫描周期 2...150 毫秒（缺省值 100 毫秒）来触发主任务。
- **周期性任务**：定期处理持续时间较短的子程序。
对于周期性任务，通过在周期扫描模式下指定扫描周期 5...255 毫秒（缺省值 255 毫秒），借助因此而配置的软件定时器来触发。
- **事件任务**：持续时间很短的子程序，用于减少应用程序的响应时间。
事件任务由物理输入或 HSC 功能块触发。这些事件与嵌入式数字量输入 (%I0.2...%I0.5)（上升沿、下降沿或两者）或与高速计数器 (%HSC0 和 %HSC1)（当计数达到高速计数器阈值时）相关联。您可以为每个 HSC 功能块配置 2 个事件。

周期性任务和事件在周期性扫描模式中配置。主任务在正常扫描模式或周期性扫描模式中配置。

任务优先级

下表概述了任务类型及其优先级：

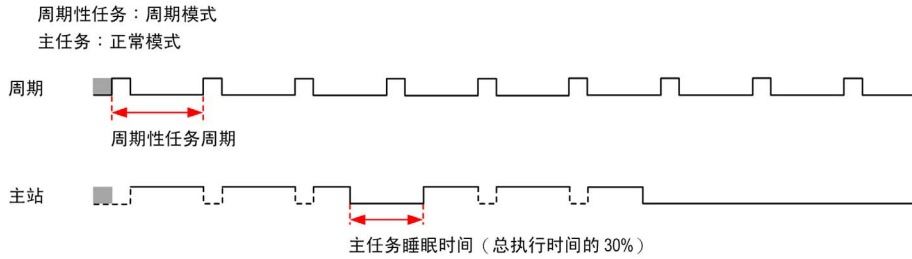
任务类型	扫描模式	触发条件	可配置范围	最大任务数	优先级
主任务	正常	正常	不适用	1	最低
	周期性任务	软件定时器	2...150 毫秒		
周期	周期性任务	软件定时器	5...255 毫秒	1	高于主任务但低于事件任务
事件任务	周期	物理输入	%I0.2...%I0.5	4	最高
		%HSC 功能块	每个 %HSC 对象 2 个事件	4	

事件优先级

请参阅事件优先级和队列（参见第 93 页）。

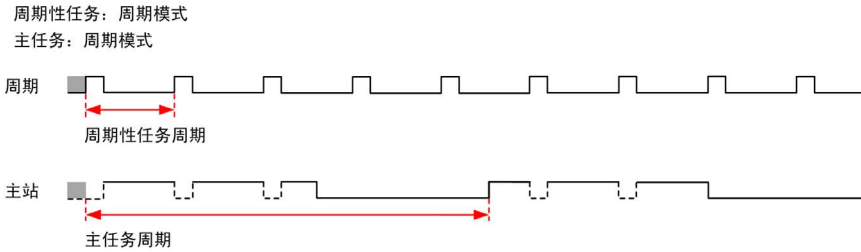
正常扫描模式中的主任务

下图显示了在正常扫描模式中配置主任务时，主任务和周期性任务执行之间的关系：



周期性扫描模式中的主任务

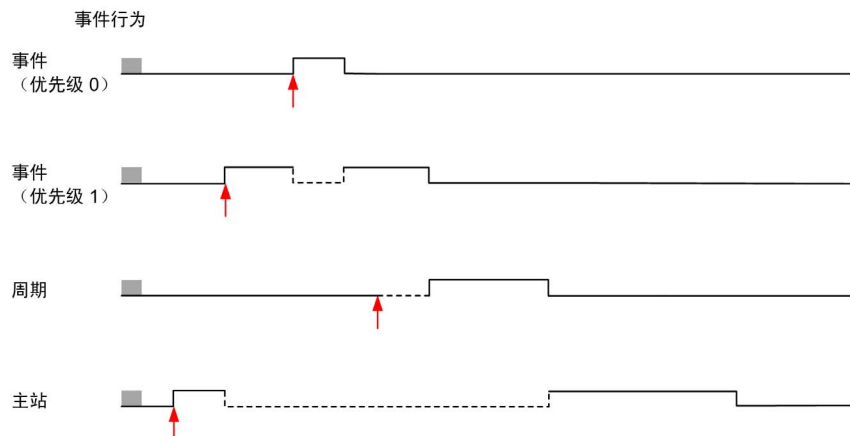
下图显示了在周期性扫描模式中配置主任务时，主任务和周期性任务之间的关系：



主任务与周期性任务的事件优先级

事件优先级控制事件任务、主任务和周期性任务之间的关系。事件任务将中断主任务和周期性任务的执行。

下图显示在周期模式下，事件任务、主任务和周期性任务之间的关系：



事件任务由将任务事件发送到事件任务的硬件中断触发。

节 6.4

管理 POU

本节包含了哪些内容？

本节包含了以下主题：

主题	页
POU	76
使用任务管理 POU	77
管理梯级	78
自由 POU	80

POU

概述

程序组织单元 (POU) 是程序中可重复使用的对象。每个 POU 都包含采用支持的编程语言编写的源代码中的一个变量声明以及一组指令。

始终存在一个 POU，该 POU 链接到程序的主任务。每次启动程序时，便会自动调用此 POU。

您可以创建包含其他对象（例如功能或功能块）的附加 POU。

首次创建时，POU 可能属于以下情形之一：

- 与任务关联 (参见第 77 页)，或者
- 自由 POU (参见第 80 页)。Free POU 不与特定任务或事件关联。例如，自由 POU 可能包含独立于主程序进行维护的库功能。可从程序内部以子程序或跳转的方式调用自由 POU。周期性任务 (参见第 87 页) 是以自由 POU 形式实现的子程序。

使用任务管理 POU

创建与任务关联的新 POU

步骤	操作
1	单击编程窗口左侧区域的 任务 选项卡。
2	右键单击 主任务 项目，然后从显示的右键菜单中选择 添加 POU 。 新的 POU 便会添加到缺省 POU 下面的程序结构中。缺省名称为 n - 新 POU ，其中 n 是整数，随每次创建 POU 而递增。
3	单击“向上”和“向下”按钮，在与任务关联的 POU 列表中上移和下移所选 POU。

重命名 POU


步骤	操作
1	在编程窗口中，使用以下任意方法编辑 POU 名称： <ul style="list-style-type: none"> 在任务选项卡中，右键单击 POU，然后从显示的右键菜单中选择重命名 POU。 在任务选项卡中双击 POU。 在编程工作空间中双击 POU 名称。
2	输入 POU 的新名称，然后按 ENTER。

删除 POU

步骤	操作
1	右键单击 POU。
2	从显示的右键菜单中选择 删除 POU 。
3	单击 是 ，确认删除。

管理梯级

创建梯级

步骤	操作
1	<p>可以使用以下两种方法之一：</p> <ul style="list-style-type: none"> ● 右键单击编程窗口的左侧区域中的任务选项卡中的 POU，然后从显示的右键菜单中选择添加梯级，或者 ● 单击位于任务选项卡顶部的工具栏上的添加梯级按钮 。 <p>随即显示新梯级。</p>
2	系统会为梯级提供序列标识符，例如 Rung0。还可以添加梯级注释，以便通过单击梯级标头来识别梯级。
3	缺省编程语言为 LD（梯形图）。要为此梯级选择其他编程语言，可单击 LD，然后选择其他语言。
4	<p>如果使用 JUMP 指令来调用此梯级，则单击梯级序列标识符 Rungx 下的下拉按钮来为梯级分配标签（其中 x 是 POU 中的梯级编号），并且从列表中选择 %L。</p> <p>结果：梯级标记为 %Ly，其中 y 是标签编号。%L 显示在按钮上，标签编号 y 则以后缀的形式显示在按钮中。</p> <p>注意：定义下一个标签时，标签编号按 1 递增。</p> <p>要修改标签编号，可双击梯级中的标签编号，然后输入新的编号，并且按 ENTER 键。</p>

在现有梯级的上方插入梯级

步骤	操作
1	在编程工作空间中选择一个现有梯级。
2	<p>单击位于 [编程] 窗口顶部的工具栏上的插入新梯级按钮 。</p> <p>新的梯级便会显示在所选梯级的上方。</p>
3	系统会为梯级提供序列标识符，例如 Rung0。还可以添加梯级注释，以便通过单击梯级标头来识别梯级。
4	缺省编程语言为 LD（梯形图）。要为此梯级选择其他编程语言，可单击 LD，然后选择其他语言。
5	<p>如果使用 JUMP 指令来调用此梯级，则单击梯级序列标识符 Rungx 下的下拉按钮来为梯级分配标签（其中 x 是 POU 中的梯级编号），并且从列表中选择 %L。</p> <p>结果：梯级标记为 %Ly，其中 y 是标签编号。%L 显示在按钮上，标签编号 y 则以后缀的形式显示在按钮中。</p> <p>注意：定义下一个标签时，标签编号按 1 递增。</p> <p>要修改标签编号，可双击梯级中的标签编号，然后输入新的编号，并且按 ENTER 键。</p>

复制梯级

步骤	操作
1	右键单击要复制的梯级，然后从显示的右键菜单中选择 复制选定梯级 。
2	右键单击 POU ，然后选择 粘贴梯级 。 结果： 插入没有标签的梯级副本。 注意： 复制梯级时，不会复制梯级的标签。


注意：也可以在**编程**窗口中复制和粘贴梯级：

步骤	操作
1	右键单击相应的梯级，然后选择 复制选定梯级 。
2	右键单击最后一个现有梯级下方的任意位置，然后选择 粘贴梯级 。

重命名梯级

步骤	操作
1	在 编程 窗口中，使用以下任意方法编辑梯级名称： <ul style="list-style-type: none"> 在任务选项卡中，右键单击梯级，然后从显示的右键菜单中选择重命名梯级。 在任务选项卡中双击梯级。 在程序梯级中，双击梯级名称或文本名称。
2	输入梯级的新名称，然后按 ENTER 。

删除梯级

步骤	操作
1	可以使用以下两种方法之一： <ul style="list-style-type: none"> 在任务选项卡中，右键单击梯级，然后从显示的右键菜单中选择删除梯级，或者 在任务选项卡中，选择梯级，并且单击位于编程工作空间顶部的删除梯级按钮 。
2	如果梯级不为空，则会提示您确认删除梯级。单击 是 进行确认，或单击 否 以取消操作。

自由 POU

简介

在 SoMachine Basic 中，自由 POU 是特殊的 POU 类型，它不与任务显式关联：

- [-] 自由 POU
 - [-] 自由 POU_0 (SR2)
 - Rung0
 - Rung1
 - [-] 自由 POU_1 (SR3)
 - Rung0
 - [-] 自由 POU_2 (SR4)
 - Rung0
 - Rung1
 - Rung2

每个自由 POU 都以子程序的形式实现，由使用 SoMachine Basic 支持的任意编程语言编写的 1 个或多个梯级组成。

以下情况使用自由 POU：

- 使用程序梯级内跳转指令对子程序 (SRi) 进行调用
- 配置为周期性任务
- 配置为事件任务，例如高速计数器 (HSC) 功能块 (%HSCi.TH0) 阈值 0 的子程序

当作为周期性任务或事件任务使用时，自由 POU 子程序会自动从任务窗口的自由 POU 区域分别移到该窗口的周期性任务或事件区域。

如果不再作为周期性任务或事件任务使用，则子程序会返回到自由 POU 区域，并供其他任务或事件使用。

创建新的自由 POU

请执行以下步骤以创建新的自由 POU：

步骤	操作
1	选择编程窗口左侧区域的 任务 选项卡。
2	右键单击 自由 POU ，然后选择 添加自由 POU 。 缺省名称为“自由 POU_0”且缺省子程序编号为“SR0”的新 POU 便会显示在 自由 POU 分支的下方，并且新的梯级会显示在 编程 工作空间中。
3	也可以选择右键单击新的 POU 并选择 重命名 POU ，然后输入 POU 的新名称并按 Enter。 自由 POU 的名称也会在 编程 工作空间中显示的梯级中更新。
4	也可以选择输入要与自由 POU 关联的注释 (参见第 124 页)。

步骤	操作
5	选择注释框右侧的 子程序编号 ，并从列表中选择一个子程序编号。 自由 POU 列表中的 POU 描述会随所选的子程序编号（例如“SR11”）进行更新。
6	使用所选的编程语言为自由 POU 创建梯级和源代码。

复制和粘贴现有 POU

请执行以下步骤，复制和粘贴与任务关联的现有 POU，从而创建自由 POU：

步骤	操作
1	选择 编程 窗口左侧区域的 任务 选项卡。
2	右键单击现有 POU，然后选择 复制 POU 。
3	右键单击 自由 POU ，然后选择 粘贴 POU 。 名称为“自由 POU_x”（其中 x 是下一个可用的自由 POU 编号），并且缺省子程序编号为“SRx”（其中 x 是下一个可用的子程序编号）的新的自由 POU 便会显示在 自由 POU 的下方。POU 的所有梯级都会自动与新的自由 POU 子程序编号相关联。

将自由 POU 分配给事件或周期性任务

缺省情况下，自由 POU 和子程序不与任何事件或任务相关联。

有关如何将自由 POU 与周期性任务相关联的信息，请参阅创建周期性任务（参见第 87 页）。

有关如何将自由 POU 与事件相关联的信息，请参阅创建事件任务（参见第 94 页）。

节 6.5

主任务

本节包含了哪些内容？

本节包含了以下主题：

主题	页
主任务描述	83
配置主任务	84

主任务描述

概述

主任务指应用程序的主要任务。主任务是缺省情况下创建的必需任务。主任务由程序组织单元 (POU) 内存在的段和子程序组成。可以使用任何支持的编程语言对主任务的每个 POU 进行编程。

过程

适用于	请参阅
在主任务中创建新的 POU	创建与任务关联的新 POU (参见第 77 页)
在主任务中重命名 POU	重命名 POU (参见第 77 页)
从主任务中删除 POU	删除 POU (参见第 77 页)

配置主任务

过程

按照以下步骤来配置主任务：

步骤	操作
1	选择编程窗口左侧区域的 任务 选项卡。
2	选择 主任务 项目。 主任务 属性便会显示在 SoMachine Basic 窗口的中心区域下方。
3	根据需要修改属性。

主任务属性

扫描模式

选择要用于程序的扫描模式：

- **正常**：如果 Logic Controller 处于正常（自由运行）扫描模式下，则在前一个扫描完成后便会立即启动新扫描。
- **周期性**：在周期性扫描模式下，Logic Controller 会等待配置的扫描时间过去之后再启动新扫描。因此，每次扫描的持续时间都相同。
为周期性扫描模式指定扫描**周期**，范围为 2~150 毫秒。缺省值为 100 毫秒。

缺省扫描模式为“正常”。

控制主任务的系统位和系统字

主任务可以通过系统位 (%S) 和系统字 (%SW) 进行控制：

下表列出了系统位：

系统位	描述
%S11	看门狗溢出
%S19	扫描周期溢出（周期性扫描模式）

下表列出了系统字：

系统字	描述
%SW0	Logic Controller 扫描周期（周期性扫描模式）
%SW27	系统中上次主任务扫描循环期间所用的时间（毫秒）
%SW30	上次扫描时间。显示上次控制器扫描循环的执行时间（毫秒），即主任务扫描循环开始（获得输入）与结束（更新输出）之间所经过的时间。
%SW31	最大扫描时间。显示自 Logic Controller 上次冷启动以来最大控制器扫描周期的执行时间（毫秒）。
%SW32	最小扫描时间。自 Logic Controller 上次冷启动以来最小控制器扫描周期的执行时间（毫秒）。

有关系统位和系统字的完整列表及其含义，请参阅相应硬件平台的 *编程指南*。

节 6.6

周期性任务

本节包含了哪些内容？

本节包含了以下主题：

主题	页
创建周期性任务	87
配置周期性任务扫描持续时间	89

创建周期性任务

概述

周期性任务是由主任务定期处理的子程序，通常持续时间较短。在 SoMachine Basic 中，该子程序作为自由 POU (参见第 80 页) 实现。可以使用 SoMachine Basic 支持的任意编程语言编写该子程序。

注意： 程序中可以存在的最大周期性任务数由 Logic Controller 硬件确定。有关详细信息，请参阅 Logic Controller 的 *编程指南*。

将子程序分配给周期性任务

步骤	操作
1	创建新的自由 POU (参见第 80 页)，其中包含周期性任务子程序。
2	单击编程窗口左侧区域的 任务 选项卡。
3	单击程序树中的 周期性任务 。
4	<p>单击分配自由 POU按钮 。</p> <p>随即显示选择自由 POU窗口：</p>  <p>窗口标题为“选择自由 POU”，包含以下列表项： 周期性任务 (SR0) 自由 POU_0 (SR1)</p> <p>窗口底部有“确定”和“取消”按钮。</p>
5	<p>选择要分配给周期性任务的自由 POU，然后单击确定。</p> <p>所选子程序将从编程树的自由 POU部分移到周期性任务部分。</p> <p>例如，如果将包含子程序 SR4 的自由 POU“自由 POU_0”分配给周期性任务，则自由 POU_0 (%SR4)条目将从程序树的自由 POU部分移到周期性任务部分。</p>
6	<p>要配置周期性任务的持续时间，请在编程树中单击周期性任务，然后在显示的属性窗口中指定周期性任务的周期（持续时间），范围介于 5 至 255 毫秒。</p>

从周期性任务中删除子程序

步骤	操作
1	单击 编程 窗口左侧区域的 任务 选项卡。
2	单击程序树中的 周期性任务 。
3	单击 取消分配自由 POU 按钮  。 所选子程序将从编程树的 周期性任务 部分移到 自由 POU 部分。

配置周期性任务扫描持续时间

过程

按照以下步骤来配置周期性任务的扫描持续时间：

步骤	操作
1	选择编程窗口左侧区域的 任务 选项卡。
2	选择 周期性任务 项目。 周期性任务 属性便会显示在 SoMachine Basic 窗口的中心区域下方。
3	根据需要修改属性。

周期性任务属性

为周期性任务指定扫描**周期**，范围为 5~255 毫秒，然后单击**应用**。缺省值为 255 毫秒。

节 6.7

事件任务

本节包含了哪些内容？

本节包含了以下主题：

主题	页
事件任务概述	91
事件源	92
事件优先级和队列	93
创建事件任务	94

事件任务概述

简介

事件任务：

- 是满足指定条件（事件源）时执行的程序的一部分
- 优先级高于主程序
- 实现快速响应时间，从而减少系统的总响应时间。

事件描述

事件由以下内容组成：

- *事件源*：触发事件后，使程序中断的软件或硬件条件
- *POU*：与事件关联的独立程序实体（子程序）
- *事件队列*：用于在事件执行前存储事件列表
- *优先级*：分配给事件的优先级，以确定事件执行的顺序。

事件源

概述

有 9 个事件源可用：

- 4 个链接到 Logic Controller 的所选物理输入
- 4 个链接到 HSC 功能块阈值（每 %HSC 实例 2 个事件）
- 1 个周期性条件。

事件源会始终连接到单个事件。事件触发后，控制器会立即检测到该事件，然后执行与该事件关联的子程序。

Logic Controller 的物理输入事件

可以将 Logic Controller 的嵌入式数字量输入 %I0.2、%I0.3、%I0.4 和 %I0.5 配置为事件源。

这些事件源可以配置为：

- 在检测到上升沿、下降沿或上升和下降沿时触发事件
- 为事件分配优先级
- 识别与事件关联的子程序。

有关配置输入事件的详细信息，请参阅 Logic Controller 的 *编程指南*。

%HSC 功能块的输出事件

%HSC 功能块的阈值输出 TH0 和 TH1 可以用作事件源。输出 TH0 和 TH1 分别设置为：

- 1，当值大于阈值 S0 和阈值 S1 时，
- 0，当值小于阈值 S0 和阈值 S1 时。

这些输出的上升沿或下降沿可以激活事件处理。

有关配置输出事件的详细信息，请参阅 Logic Controller 的 *编程指南*。

周期性事件

此事件源会定期执行指定的程序段（子程序）。此子程序的优先级高于主任务。但是，周期性事件源的优先级低于所有其他事件源。

有关配置此事件的详细信息，请参阅周期性任务（参见第 86 页）。

事件优先级和队列

事件优先级和队列

事件具有 8 个可能的优先级之一，从 7（最低）到 0（最高）。

每个事件源分配优先级。一次只有一个事件源可能具有优先级 0。因此，其他事件具有较低的优先级，而其执行顺序取决于其相对的优先级和检测到它们的顺序。

事件的优先级控制事件任务执行之间的关系。所有事件任务都会中断主任务和周期性任务的执行。有关详细信息，请参阅主任务与周期性任务的事件优先级（参见第 74 页）。

注意：如果在执行其他任务期间调用事件任务，则写入存储器的全局区域或影响 I/O 值时请务必谨慎。修改以其他方式在其他任务中使用的值可能会对这些任务的逻辑结果造成不利的影



警告

意外的设备操作

在将应用程序投入使用之前，请彻底测试并验证所有任务（主任务、周期性任务和任何事件任务）以及这些任务之间的相互影响。

不遵循上述说明可能导致人员伤亡或设备损坏。

有关配置事件任务优先级的信息，请参阅控制器的 *编程指南*。

事件队列管理

每次出现链接到事件源的中断时，便会启动以下序列：

步骤	描述
1	中断管理： <ul style="list-style-type: none"> ● 识别物理中断， ● 将事件存储在适当的事件队列中， ● 验证没有相同优先级的事件挂起（如果有，则该事件会在队列中保持挂起状态）。
2	保存环境。
3	执行链接到事件编程段的子程序（标记为 SRi: 的子程序）。
4	更新输出
5	恢复环境

重新建立环境之前，必须执行队列中的所有事件。

创建事件任务

概述

您可以查看当前配置的事件源、当前连接到事件的子程序，以及使用系统位和系统字检查事件的当前状态。


要查看当前分配的事件源和分配给事件的子程序（自由 POU），请执行以下操作：

步骤	操作
1	选择 编程 窗口左侧区域的 任务 选项卡。
2	选择 事件 ： <input type="checkbox"/> 事件 %HSC0.TH0 : %HSC0.TH1 : %I0.2 : <input type="checkbox"/> %I0.3 : 自由 POU_0 Rung0 尚未分配给子程序的已配置事件源以红色显示。

注意： 只有嵌入式控制器输入 / 输出才能在事件子程序中使用。


将自由 POU 分配给事件源

请执行以下步骤，将自由 POU 分配给已配置的事件源：

步骤	操作
1	创建新的自由 POU (参见第 80 页)，其中包含用于事件的子程序。
2	在 事件 列表中选择事件源。
3	单击 分配自由 POU 按钮  随即显示 选择自由 POU 窗口。
4	选择要分配给事件源的自由 POU，然后单击 确定 。 所选自由 POU 便会从编程树的 自由 POU 区域移到 事件 区域。

从事件中删除子程序

要删除子程序与事件源之间的关联，请执行以下操作：

步骤	操作
1	在 事件 列表中选择事件源。
2	单击 删除自由 POU 按钮  。 所选子程序将从编程树的 事件 区域移到 自由 POU 区域。

使用系统位和系统字检查事件

以下系统位用于检查事件：

系统位	描述
%S31	用于执行或延迟事件
%S38	用于决定是否将事件放入事件队列中
%S39	用于确定是否丢失事件

以下系统字用于检查事件：

系统字	描述
%SW48	自 Logic Controller 上次冷启动以来执行的事件数（计算除周期性事件外的所有事件）。

冷启动或加载应用程序之后，%S39 和 %SW48 的值复位为 0，系统位 %S31 和 %S38 的值设为其初始状态 1。它们的值在热重启后保持不变。在所有情况下，事件队列都将复位。

节 6.8

使用工具

本节包含了哪些内容？

本节包含了以下主题：

主题	页
程序消息	97
动态数据表	98
存储器对象	101
系统对象	103
I/O 对象	104
软件对象	105
搜索并替换	106
符号列表	108
梯级模板	110

程序消息

概述

SoMachine Basic 会将**编程**选项卡中显示的源代码持续编译成可下载到 Logic Controller 的程序。如果无法编译，则会显示相应的消息。

编译状态

如果梯级编译成功且没有要显示的消息，则会显示绿色勾选符号：

IL		名称		
✓	Rung0	0000	LD	%I0.0 注释
	符号	0001	ST	%Q0.0 注释

如果 SoMachine Basic 无法编译程序，原因是梯级不完整，例如不包含最终指令（例如 END、CALL 或者 Jump），则会显示黄色三角形警告符号：

IL		名称		
⚠	Rung0	0000	LD	%I0.0 注释
	符号	0001	ANDN	%I0.1 注释

如果 SoMachine Basic 检测到的错误阻止成功编译梯级，则会显示红色叉号：

IL		名称		
✗	Rung0	0000	BLK	%SBRO 注释
	符号	0001	LD	%S6 注释
		0002	CU	注释

显示程序消息

如果显示红色叉号，请执行以下步骤以显示检测到的错误的详细信息：

步骤	操作
1	单击编程树中的 工具 选项卡。
2	单击 消息 。 消息列表随即显示在 编程 窗口的中心区域下方。

动态数据表

概述

动态数据表可用于监控对象的值。

您可以手动将对象添加到动态数据表，并查看可能与这些对象关联的符号。在 SoMachine Basic 已连接到 Logic Controller 且程序正在运行（在线模式）时，动态数据表还可用于查看和修改某些对象类型的实时值。

动态数据表是 SoMachine Basic 应用程序的一个组件，因此会随程序一起上载到 Logic Controller。以后从 Logic Controller 下载应用程序时，便可以检索动态数据表中存储的对象和值。

Animation table_0						
					添加	插入
	已使用	地址	符号	值	强制	注释
▶	<input checked="" type="checkbox"/>	%M10		0		
	<input checked="" type="checkbox"/>	%M20		0		
	<input checked="" type="checkbox"/>	%M30		0		
	<input checked="" type="checkbox"/>	%SC0.0		0		
	<input checked="" type="checkbox"/>	%SC0.1		0		
	<input checked="" type="checkbox"/>	%SC0.2		0		
	<input checked="" type="checkbox"/>	%SC0.3		0		

创建动态数据表

步骤	操作
1	选择 编程 窗口左侧区域的 工具 选项卡。
2	右键单击 动态数据表 ，然后从显示的右键菜单中选择 添加新的动态数据表 。 新的动态数据表项目便会显示在 工具 窗口的 动态数据表 区域下方，而属性窗口会显示在窗口的中心区域下方。

将项目添加到动态数据表

步骤	操作
1	选择 编程 窗口左侧区域的 工具 选项卡。
2	选择要在 工具 窗口的 动态数据表 区域中配置动态数据表。 结果： 属性窗口会显示在窗口的中心区域下方。

步骤	操作
3	要将新项目添加到动态数据表的底部，请在文本框中输入对象名称并按 Enter 或单击 添加 。 以下对象可以添加到动态数据表： <ul style="list-style-type: none"> ● I/O 对象 ● 位串（示例：%Mx:L，其中 L 是位计数，应为 8 的倍数） ● 字表（示例：%MWx:L，其中 L 是字计数） ● 字的位（示例：%MWx:X，其中 X 是位的偏移）
4	要在现有对象的正上方添加新对象，请选择动态数据表中的行，并在文本框中输入要添加对象的名称，然后单击 插入 。

动态数据表属性

下表介绍了动态数据表项目的属性：

参数	可编辑	值	描述
已使用	否	True/False	表示对象当前是否正在程序中使用。
地址	否	对象地址	显示对象的地址。
符号	否	有效的符号	与此对象关联的符号名称（如果已定义）。
值	是 (1)	当前值	对象的当前值。 如果对象类型具有读 / 写访问权限且您处于在线模式（参见第 26 页）下，请双击并根据需要输入新对象值。对象的值将在 Logic Controller 中运行的程序中实时更新。 有关详细信息，请参见修改实时值（参见第 146 页）。
强制	是 (1)	强制到 0 强制到 1 取消强制	仅针对数字量输入和数字量输出显示。只有处于在线模式（参见第 26 页）下才可编辑。可用于根据需要输入或输出的值强制为 0 或 1。 选择 未强制 以删除当前应用于地址的任何强制。
注释	否	有效的注释	与此对象关联的注释（如果已定义）。
(1) 取决于对象类型以及是否处于在线模式下。			

配置动态数据表中的项目

要搜索并选择性替换动态数据表中的对象，请右键单击对象，然后选择**搜索并替换**。有关详细信息，请参阅搜索并替换（参见第 106 页）。

要从动态数据表中删除对象，请右键单击对象，然后选择**从动态数据表删除**。

重命名动态数据表

步骤	操作
1	在工具窗口的 动态数据表 区域中，右键单击要重命名的动态数据表，然后单击 重命名动态数据表 。
2	输入动态数据表的新名称，然后按 Enter 。

删除动态数据表

步骤	操作
1	在工具窗口的 动态数据表 区域中，右键单击要删除的动态数据表，然后单击 删除动态数据表 。
2	单击 是 ，确认操作。

存储器对象

概述

存储器对象包括：

- 存储器位
- 字存储器
- 常量字

选择存储器分配模式

查看或更新存储器对象的属性之前，请选择要使用的存储器分配模式（参见第 63 页）。

存储器位属性

下表描述了**存储器位**屏幕中的每个参数：

参数	可编辑	值	缺省值	描述
已使用	否	True/False	False	指示存储器位当前是否正在程序中使用。
地址	否	请参阅位对象	不适用	显示存储器位的地址，其中 x 是 Logic Controller 支持的存储器位数。
符号	是	有效的符号	无	可让您将符号与此存储器位关联。
值	是	请参阅位对象	0	此存储器位的值。
注释	是	有效的注释	无	可让您将注释与此存储器位关联

存储器字属性



请先选择要显示属性的存储器字类型：

- **%MW**。字存储器
- **%MD**。双字
- **%MF**。浮点字

下表描述了**存储器字**的属性：

参数	可编辑	值	缺省值	描述
已使用	否	True/False	False	指示存储器字当前是否正在程序中使用。
使用的等式	否	True/False	False	指示存储器字当前是否正在等式中使用。
地址	否	请参阅字对象	不适用	显示存储器字的地址。
符号	是	有效的符号	无	可让您将符号与此存储器字关联。

参数	可编辑	值	缺省值	描述
值	是	请参阅对象。	0	此存储器字的值。
注释	是	有效的注释	无	可让您将注释与此存储器字关联。

常量字属性



请先选择要显示属性的常量字类型：

- **%KW**。常量字。
- **%KD**。常量双字
- **%KF**。常量浮点字。

下表描述了**常量字**屏幕中的每个参数：

参数	可编辑	值	缺省值	描述
已使用	否	True/False	False	指示常量字当前是否正在程序中使用。
使用的等式	否	True/False	False	指示常量字当前是否正在等式中使用。
地址	否	请参阅对象	不适用	显示常量字的地址。
符号	是	有效的符号	无	可让您将符号与此常量字关联。
值	是	请参阅对象	0	此常量字的值。
注释	是	有效的注释	无	可让您将注释与此常量字关联。

系统对象

概述

系统位和系统字特定于 Logic Controller。有关详细信息，请参阅 Logic Controller 的 *编程指南*。

I/O 对象

概述

以下对象类型由硬件限定且取决于正在使用的 Logic Controller:

- 数字量输入和输出
- 模拟量输入和输出
- 高级功能块，例如快速计数器、高速计数器和脉冲发生器。

有关详细信息，请参阅 Logic Controller 的 *编程指南*。

软件对象

概述

SoMachine Basic 支持以下一般软件对象：

对象	描述
定时器	用于指定在执行操作（例如触发事件）之前的时间段。
计数器	提供事件的加减计数。
消息	允许与外部设备通讯。
LIFO/FIFO 寄存器	存储器功能块，在 FIFO 或 LIFO 模式下分别可存储最多 16 位的 16 个字。
Drum 寄存器	操作原理与机电 Drum Controller 类似，可根据外部事件更改步进。在每次步进中，凸轮的高点会提供一个由 Logic Controller 执行的命令。
Shift Bit Register	提供二进制数据位（0 或 1）向左或向右移位。
步进计数器	提供可分配操作的一系列步进。
排程功能块	用于在预定义的月份、日期和时间控制操作。
PID	用于调整比例 - 积分 - 微分 (PID) 功能。

这些功能块在 *SoMachine Basic 通用功能库指南* 中进行了介绍。

选择存储器分配模式

查看或更新软件对象的属性之前，请选择要使用的存储器分配模式（参见第 63 页）。

搜索并替换

概述

“搜索并替换”可用于查找程序中任意位置使用的某一对象的所有实例，并将它选择性替换为其他对象。

搜索并替换项目

步骤	操作
1	选择 编程 窗口左侧区域的 工具 选项卡。 还可以从 SoMachine Basic 中不同的其他位置调用“搜索并替换”功能，例如，右键单击动态数据表中的项目 (参见第 98 页)，然后选择 搜索并替换 。
2	可以使用以下任意一种方法来显示 搜索并替换 窗口： <ul style="list-style-type: none"> ● 单击编程窗口的工具选项卡中的搜索并替换。 ● 右键单击梯级或梯级中的所选项目，然后在显示的右键菜单中单击搜索并替换。 ● 右键单击任意对象的属性窗口中的行，然后在显示的右键菜单中单击搜索并替换。 <p>下图显示了搜索并替换窗口：</p> 
3	在 查找 框中，输入要查找的对象或符号名称。如果通过右键单击梯级中的所选项目或者对象的属性窗口中的项目来开始搜索，则会预填充 查找 字段。 您可以使用以下通配符： <ul style="list-style-type: none"> ● 星号 (*). 替换搜索术语中的 0 个或多个字符。例如，<code>%MW1*</code> 将查找 <code>%MW1</code> 和 <code>%MW101</code>。 ● 问号 (?). 替代搜索术语中的 1 个字符。例如，输入 <code>COIL?2</code> 将查找 <code>COIL12</code> 而不是 <code>COIL012</code>
4	也可以选择 替换 框中输入替换对象或符号名称。
5	选择 程序 以搜索当前程序的源代码中的项目。 选择 注释 以搜索程序注释中的项目。

步骤	操作									
6	<p>单击搜索或替换。还可以按 ENTER 键来开始搜索。 只有在替换框中提供替换对象或符号名称之后，才会启用替换按钮。 找到的所有项目会在结果列表中列出：</p> <p>结果 <input type="checkbox"/> 显示符号</p> <table border="1"> <thead> <tr> <th>POU</th> <th>梯级</th> <th>代码</th> </tr> </thead> <tbody> <tr> <td>POU_0</td> <td>Rung_0</td> <td>%Q0.0</td> </tr> <tr> <td>POU_0</td> <td>Rung_1</td> <td>LD %Q0.0</td> </tr> </tbody> </table>	POU	梯级	代码	POU_0	Rung_0	%Q0.0	POU_0	Rung_1	LD %Q0.0
POU	梯级	代码								
POU_0	Rung_0	%Q0.0								
POU_0	Rung_1	LD %Q0.0								
7	<p>也可以选中显示符号，改为显示为对象定义的任何符号：</p> <p>结果 <input checked="" type="checkbox"/> 显示符号</p> <table border="1"> <thead> <tr> <th>POU</th> <th>梯级</th> <th>代码</th> </tr> </thead> <tbody> <tr> <td>POU_0</td> <td>Rung_0</td> <td>OUTPUT</td> </tr> <tr> <td>POU_0</td> <td>Rung_1</td> <td>LD OUTPUT</td> </tr> </tbody> </table>	POU	梯级	代码	POU_0	Rung_0	OUTPUT	POU_0	Rung_1	LD OUTPUT
POU	梯级	代码								
POU_0	Rung_0	OUTPUT								
POU_0	Rung_1	LD OUTPUT								
8	<p>单击任意列出的结果，可直接跳转到程序中的代码行。</p>									

符号列表

概述

您可以显示已与程序中对象建立关联的所有符号的列表。会显示具有符号的所有对象，但系统位 (%S) 和系统字 (%SW) 除外。

定义和使用符号 (参见第 61 页) 介绍了如何创建符号并在程序中使用这些符号。

显示符号列表

步骤	操作
1	选择编程窗口左侧区域的工具选项卡。
2	单击 符号列表 。 随即显示“符号列表”窗口。对于每个项目，都会显示以下信息： <ul style="list-style-type: none"> ● 已使用。符号当前是否正在程序中使用。 ● 地址。与符号关联的对象的地址。 ● 符号。符号名称。

导入符号

步骤	操作
1	单击 导入 按钮，或右键单击符号列表中的任意位置并选择 导入符号 。 随即显示 导入符号 窗口。
2	浏览并选择包含要导入符号的逗号分隔值 (CSV) 文件的 文件路径 。
3	也可以选择单击 导入选项 ，为导入的符号配置格式选项： <div style="border: 1px solid #ccc; padding: 5px; margin-top: 5px;"> <input type="checkbox"/> 导入选项 备份 <input checked="" type="checkbox"/> 导入模式 <input checked="" type="radio"/> Delta <input type="radio"/> 完全 分隔符 <input type="text" value="分号"/> </div>

步骤	操作																
4	<p>单击导入。</p> <p>将会创建所选 CSV 文件中的所有符号，并使用指定的格式选项显示在“符号列表”窗口中。如果在导入期间检测到错误，则会显示一个包含这些错误的报告：</p> <div data-bbox="353 293 1211 773" style="border: 1px solid #ccc; padding: 10px;"> <p style="text-align: center; color: green; font-weight: bold;">关于 'symbols_timer_drum.csv' 的报告 ×</p> <hr style="border: 5px solid green;"/> <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th>类型</th> <th>消息</th> <th>线</th> <th>列</th> </tr> </thead> <tbody> <tr> <td>信息</td> <td>备份 'H:\Nigel\SoMachine Basic\Test projects\SymBackup_20130724.csv' 成功</td> <td>0</td> <td>0</td> </tr> <tr> <td>创建</td> <td>符号 'M2' 与 '%M2' 关联</td> <td>2</td> <td>0</td> </tr> <tr> <td>信息</td> <td>导入成功</td> <td>0</td> <td>0</td> </tr> </tbody> </table> <div style="text-align: right; margin-top: 10px;"> <input type="button" value="保存"/> <input type="button" value="关闭"/> </div> </div> <p>单击保存，将报告的内容写入纯文本 (.txt) 文件。</p>	类型	消息	线	列	信息	备份 'H:\Nigel\SoMachine Basic\Test projects\SymBackup_20130724.csv' 成功	0	0	创建	符号 'M2' 与 '%M2' 关联	2	0	信息	导入成功	0	0
类型	消息	线	列														
信息	备份 'H:\Nigel\SoMachine Basic\Test projects\SymBackup_20130724.csv' 成功	0	0														
创建	符号 'M2' 与 '%M2' 关联	2	0														
信息	导入成功	0	0														

导出符号列表

步骤	操作
1	单击 导出 按钮，或右键单击符号列表中的任意位置并选择 导出符号 。此时系统会提示您保存更改。随即显示 导出符号 窗口。
2	浏览并选择要创建的逗号分隔值 (CSV) 文件的 文件路径 和 文件名 。
3	<p>也可以选择单击导出选项，为导出的值配置格式选项：</p> <div data-bbox="353 1149 669 1299" style="border: 1px solid #ccc; padding: 5px;"> <p>∧ 导出选项</p> <p>包含 <input checked="" type="checkbox"/> 标题 <input checked="" type="checkbox"/> 注释</p> <p>分隔符 <input type="text" value="分号"/></p> <p>代码页 <input type="text" value="Unicode"/></p> </div>
4	单击 导出 。随即使用指定的格式选项创建 CSV 文件。

梯级模板

概述

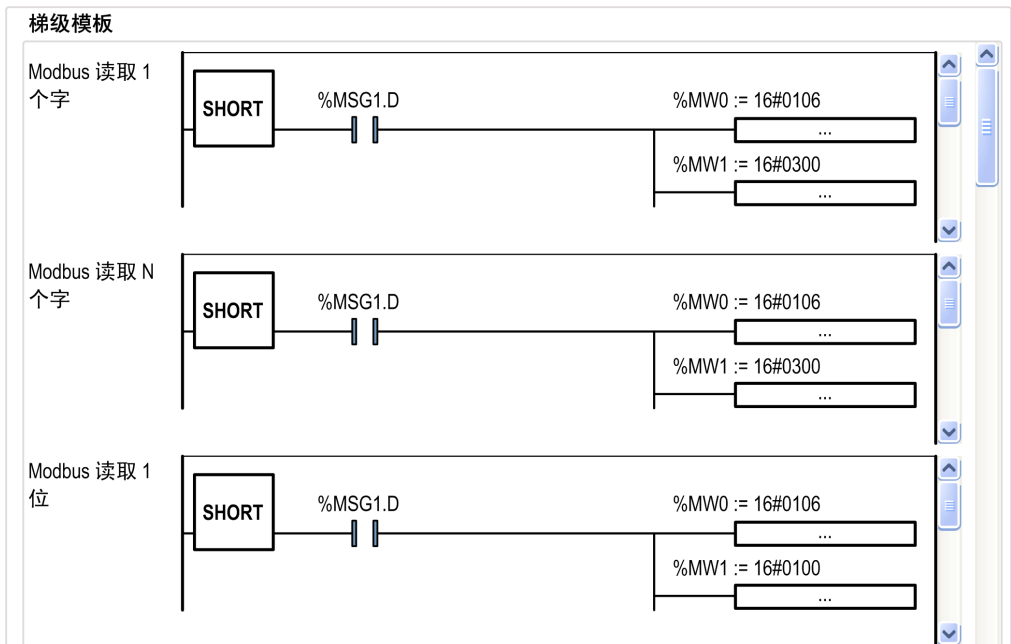
梯级模板是源代码的预先配置部分，您可以将它插入程序，使编程更加快速，同时减少编码错误。SoMachine Basic 会维护梯形图和指令列表梯级模板的单独列表。

将梯级模板插入程序中

按照以下步骤将梯级模板插入程序：

步骤	操作
1	选择 编程 窗口左侧区域的 工具 选项卡。
2	单击 梯级模板 → 梯形图 或 梯级模板 → 指令列表 随即显示梯形图或指令列表格式的当前梯级模板列表。
3	可以通过以下任一方法将梯级模板插入程序： <ul style="list-style-type: none">● 在编程工作空间的程序选择一个梯级，然后双击某梯级模板。● 右键单击某梯级模板，并单击右键菜单中的复制梯级，然后右键单击编程工作空间，并单击右键菜单中的粘贴梯级。 此梯级模板将始终插入 POU 中最后一个梯级之后。使用 编程 窗口 任务 选项卡中的向上和向下箭头按钮重定位您的程序中的梯级。

下图以梯形图语言显示了梯级模板：



下图以指令列表语言显示了梯级模板：

梯级模板	
Modbus 读取 1 个字	0000 LD 1
	0001 AND %MSG1.D
	0002 [%MW0 := 16#0106]
	0003 [%MW1 := 16#0300]
	0004 [%MW2 := 1]
Modbus 读取 N 个字	0000 LD 1
	0001 AND %MSG1.D
	0002 [%MW0 := 16#0106]
	0003 [%MW1 := 16#0300]
	0004 [%MW2 := 1]
Modbus 读取 1 位	0000 LD 1
	0001 AND %MSG1.D
	0002 [%MW0 := 16#0106]
	0003 [%MW1 := 16#0100]
	0004 [%MW2 := 1]

节 6.9

梯形图语言编程

本节包含了哪些内容？



本节包含了以下主题：

主题	页
梯形图简介	114
梯形图的编程原理	116
梯形图图形元素	117
比较块	122
操作块	123
添加注释	124
编程最佳做法	125

梯形图简介

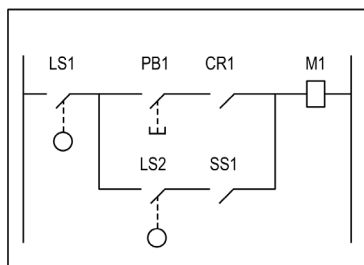
简介

梯形图与表示继电器控制电路的继电器逻辑图类似。两者之间的主要差异在于：继电器逻辑图没有梯形图编程的以下特性：

- 触点符号 () 代表所有输入和二进制逻辑位。
- 线圈符号 () 代表所有输出和二进制逻辑位。
- 数字运算包含在图形梯形图指令集中。

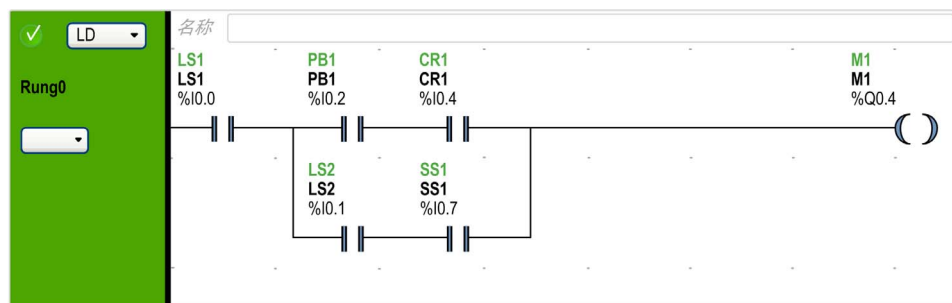
梯形图相当于继电器电路

下图显示了继电器逻辑电路的简化接线图：



Relay logic circuit

相对应的梯形图：



在上图中，继电器逻辑图中与开关设备关联的所有输入在梯形图中显示为触点。继电器逻辑图中的 M1 输出线圈在梯形图中表示为输出线圈符号。梯形图中每个触点 / 线圈符号上方显示的地址编号是指 Logic Controller 的外部输入 / 输出连接的位置。

梯形图梯级

以梯形图语言编写的程序由梯级组成，梯级是 2 个垂直电位棒之间绘制的图形指令集。梯级由 Logic Controller 按顺序执行。

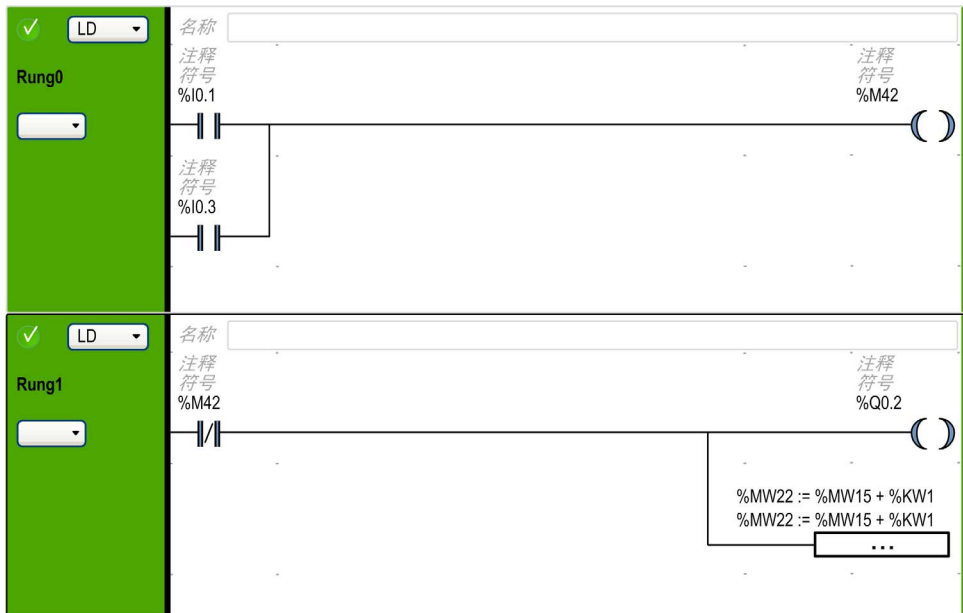
图形指令集代表以下功能：

- 控制器的输入 / 输出（按钮、传感器、继电器、指示灯等）
- 控制器的功能（定时器、计数器等）
- 数学和逻辑运算（加法、除法、AND、XOR 等）
- 比较运算符和其他数值运算（ $A < B$ 、 $A = B$ 、移位、环移等）
- 控制器中的内部变量（位、字等）

这些图形指令通过水平线和竖直线以“与”/“或”形式最终于一个或多个输出线圈连接在一起。梯级无法支持多组已链接的指令。

梯形图梯级示例

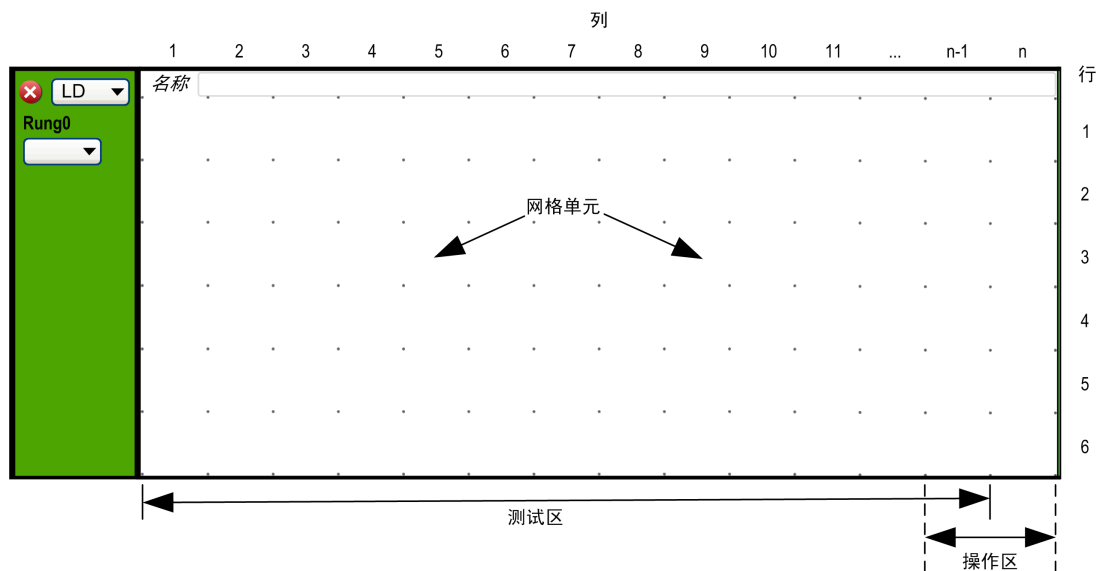
下图是由 2 个梯级组成的梯形图程序的示例。



梯形图的编程原理

编程网格

每个梯形图梯级由最多 1,000 行和 11 到 30 列的网格组成，并分为如下图所示的 2 个区域：



否 已配置的列数 (11...30)。有关配置列数的详细信息，请参阅自定义梯形图编辑器 (参见第 44 页)。

网格单元

单元格用于将图形元素放置在网格中。网格的每个单元格用单元格四角的 4 个点界定。

网格区域

缺省情况下，梯形图编程网格分为 2 个区域：

- 测试区
包含为执行操作而测试的条件。包含列 1 到 n-1 (其中 n 是已配置的列的编号)，并且包含触点、功能块和比较块。
- 操作区
包含将根据测试区中条件的测试结果而执行的输出或操作。包含列 n-1 到 n (其中 n 是已配置的列的编号)，并且包含线圈和操作块。


梯形图图形元素

简介

将图形元素从编程工作空间上方显示的工具栏拖放到网格单元格中，即可将指令插入到梯形图中。

插入图形元素

要将图形元素插入梯级，请执行以下操作：

步骤	操作
1	单击工具栏上要插入的图形元素。如果图形元素为菜单，则会显示菜单中的图形项目；单击要插入的菜单项。
2	将鼠标指针移到梯级中要插入图形元素的位置，然后单击。 注：一些元素必须插入到梯级的测试区或操作区；有关详细信息，请参阅各个图形元素的描述。
3	如果需要，请单击工具栏上的 【选择模式】 图形元素  以复位选择。



梯级

使用以下图形元素来管理程序中的梯级：

图形元素	名称	功能
	创建梯级 (参见第 78 页)	在程序工作空间中的最后一个梯级下方插入新的空梯级。
	插入梯级 (参见第 78 页)	在当前所选梯级的正上方插入新的空梯级。
	删除梯级 (参见第 79 页)	从程序中删除当前所选梯级。如果梯级不为空，则需要您确认要删除梯级的内容。


分支模式

使用以下图形元素来管理梯形图中的分支：

图形元素	名称	功能
	正常模式	用于将编程元素（例如触点、线圈等，功能块除外）与线路放置在一条直线上。
	分支模式	用于将编程元素（例如触点、线圈等，功能块除外）放置在线路的分支上。

选择和线条

使用以下图形元素来选择图形元素和绘制连线：

图形元素	名称	功能
	选择模式	选择模式。
	绘制线条	在 2 个图形元素之间绘制一条线路。
	擦除线条	擦除线路。

触点


使用以下图形元素来插入触点（1 行高乘 1 列宽）：

图形元素	名称	指令列表	功能
	常开触点	LD	在控制位对象处于状态 1 时触点断开。
	常闭触点	LDN	在控制位对象处于状态 0 时触点断开。
	用于检测上升沿的触点	LDR	上升沿：检测控制位对象从 0 到 1 的变化。
	用于检测下降沿的触点	LDF	下降沿：检测控制位对象从 1 到 0 的变化。

比较块

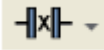
比较块放置于编程网络的测试区。只要指令的全部长度位于测试区中，比较块可以出现在测试区的任何行或列中。

比较块的图形元素占 2 个单元格（1 行高乘 2 列宽）。

图形元素	名称	指令列表	功能
	比较功能块	任何有效的比较表达式	使用比较块图形符号，将指令列表比较表达式（参见第 122 页）插入梯形图梯级。比较表达式比较 2 个操作数；检验结果后，输出将更改为 1。

布尔运算

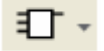
布尔运算的图形元素占 1 个单元格（1 行高乘 1 列宽）。

图形元素	名称	操作符	功能
	XOR 指令	XOR、 XORN、 XORR、 XORF	XOR 指令在操作数和前一个指令的布尔结果之间执行异或运算。 XORN 指令在操作数反转和前一个指令的布尔结果之间执行异或运算。 XORR 指令在操作数上升沿和前一个指令的布尔结果之间执行异或运算。 XORF 指令执行操作数下降沿和前一个指令的布尔结果之间的异或操作。

功能

功能块始终显示在梯形图编程网格的第一行；功能块的上方或下方不能出现梯形指令或连续性的线。梯形测试指令通向功能块的左侧；而测试指令和操作指令从功能块的右侧引出。

功能块的图形元素只能放置在测试区，且需要 2、3 或 4 行乘 2 列的单元格。

图形元素	名称	功能
	定时器、计数器、寄存器等。	每个功能块都有输入和输出，从而链接到其他图形元素。 注意： 功能块的输出不能相互连接（垂直短路）。

线圈

线圈图形元素只能放置在操作区，且占 1 个单元格（1 行高和 1 列宽）。

图形元素	名称	操作符	功能
	直接线圈	ST	相关的位对象采用测试区结果的值。
	反转线圈	STN	相关的位对象采用测试区结果的负值。
	置位线圈	S	当测试区的结果为 1 时，关联的位对象设置为 1。
	复位线圈	R	当测试区的结果为 1 时，关联的位对象设置为 0。


Grafcet 指令

使用以下图形元素来管理梯形图中的分支：

图形元素	名称	操作符	功能
	激活 Grafcet 步骤 / 停用当前步骤	#	停用当前步骤并选择性地激活 Grafcet 程序中的其他步骤。
	停用 Grafcet 步骤	#D	除停用当前步骤外，还停用 Grafcet 程序中的一个步骤。

操作块

操作块图形元素放置在操作区，占据 2 列和 1 行：

图形元素	名称	操作符	功能
	操作块	任何有效的操作符或分配指令	使用操作块图形符号，将指令列表操作和分配指令（参见第 123 页）插入梯形图梯级。






其他项目



其他项目菜单将杂项指令组合在一起。



OPEN 和 SHORT 指令提供了梯形图程序调试和故障排除的简便方法。这些特殊指令通过梯级的短路或开路连接改变梯级的逻辑，如下表中的解释。

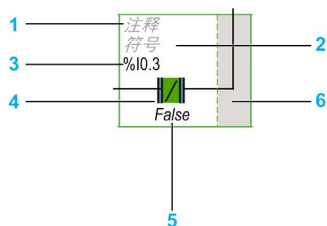
END/JUMP 图形元素放置在操作区，占据 1 个单元格（1 行高和 1 列宽）。

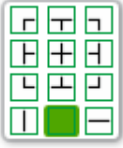
图形元素	名称	操作符	功能
	逻辑非	N	得出其操作数的逆值。
	OPEN	LD 0 AND 0	位于梯级的开始处。 梯级内：在具有连续性的梯形图梯级中创建中断，不论上次逻辑操作的结果如何。
	SHORT	LD 1 OR 1	位于梯级的开始处。 梯级内：允许连续性贯穿梯级，不论上次逻辑操作的结果如何。
	停止程序	END	定义程序的结束。
	有条件停止程序	ENDCN	定义程序的有条件结束。

图形元素	名称	操作符	功能
	跳转或子程序调用	JMP	连接到上部分或下部分已设置标签的梯级。 注意： 使用 IL 语言编程时，连接到上部分或下部分已设置标签的指令行。
	有条件跳转或子程序调用	JMPCN	有条件连接到上部分或下部分已设置标签的梯级。 注意： 使用 IL 语言编程时，连接到上部分或下部分已设置标签的指令行。

触点和线圈

插入单元格后，便会显示与触点和线圈关联的对象的附加信息：

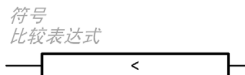


图例	项	描述
1	用户注释	单击以添加注释 (参见第 124 页)。
2	符号	单击以输入要与单元格所含对象关联的符号 (参见第 61 页) 的名称。
3	地址	单击以输入单元格所含对象的地址。
4	图形元素	图形元素。
5	实时值	当处于在线模式 (已连接到逻辑控制器且程序正在运行时)，会显示单元格中对象的实时值。
6	链接区	单击此区域以绘制用于连接此单元格与相邻单元格的连线和接头： 

比较块

将 IL 比较表达式插入梯形图

您可以使用**比较块**图形符号将指令列表比较表达式插入梯形图梯级：



继续执行下列步骤：

步骤	操作
1	单击工具栏上的 比较块  按钮。
2	单击梯级中的任意位置以插入 比较块 。
3	双击 比较表达式 行。
4	输入有效的指令列表比较运算。

获取语法帮助

如果指令列表比较运算的语法不正确，则**比较表达式**框的边框会变为红色。要获取帮助，请执行以下操作之一：

- 将鼠标移到**比较表达式**行上，或
- 选择工具 → **程序消息**。

操作块


将 IL 运算和赋值指令插入梯形图

您可以使用**操作块**图形符号将指令列表运算和赋值指令插入梯形图梯级：

符号
运算表达式



要将操作块插入梯级，请执行以下操作：

步骤	操作
1	单击工具栏上的 操作块  按钮。
2	单击要插入 操作块 的梯级的操作区（最后 2 列）。
3	双击 运算表达式 行。
4	输入有效的指令列表运算或赋值指令。

获取语法帮助

如果指令列表运算或赋值指令的语法不正确，则**运算表达式**框的边框会变为红色。要获取帮助，请执行以下操作之一：

- 将鼠标移到**运算表达式**行上，或
- 选择工具 → 程序消息。

添加注释

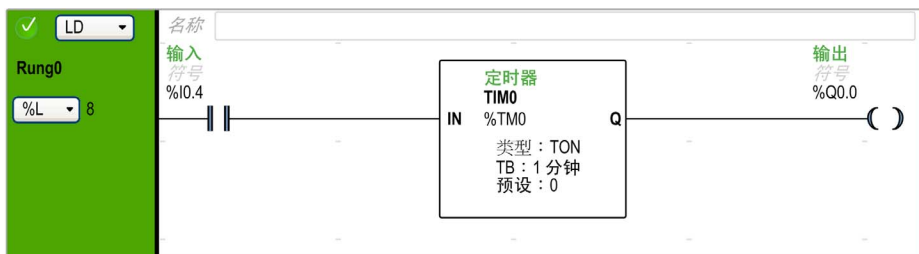
将注释添加到梯形图

要将注释添加到梯形图程序，请执行以下步骤：

步骤	操作
1	将图形元素插入梯级。
2	如果需要，请单击选择指针或按 ESC。
3	单击图形元素顶部的 注释 行。
4	输入注释。
5	单击梯级中的任意位置。

梯形图注释示例

下图显示了梯形图梯级中的注释示例：



编程最佳做法

处理程序跳转

谨慎使用程序跳转，以避免可能增加扫描时间的长循环。避免跳转到位于上部分的指令。

注意：在程序中，上部分指令行出现在跳转之前。在程序中，下部分指令行出现在跳转之后。

输出编程

在程序中，物理输出和逻辑位应该仅修改一次。对于物理输出，在对其进行更新时，仅考虑上次扫描的值。

使用直接连接的紧急停止传感器

直接用于紧急停止的传感器不得由 **Logic Controller** 进行处理。必须将它们直接连接到相应的输出设备，并且在符合当地、国家法规和 / 或国际法规的情况下进行应用。

处理电源恢复

在断电后，通过手动操作有条件地恢复电源。自动重新启动安装可能导致设备的意外操作（使用系统位 %S0、%S1 和 %S9）。



意外的设备操作

请勿在对安全性要求非常高的机器环境中使用由该软件配置及编程的设备。

不遵循上述说明可能导致人员伤亡或设备损坏。

时间和排程功能块管理

应检查系统位 %S51 的状态，该位指示任何检测到的 RTC 错误。

语法验证

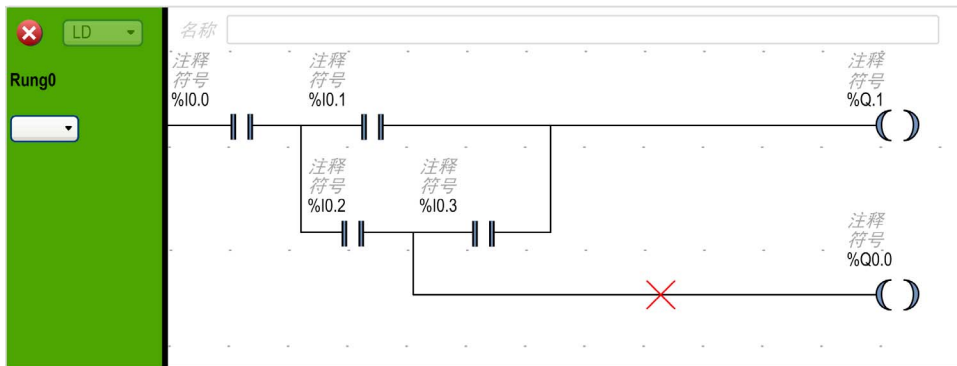
在编程期间，SoMachine Basic 会验证指令、操作数及其关联的语法。

有关使用括号的附加注意事项

请勿将分配指令放置在括号中：

```
LD    %I0.0
MPS
AND  %I0.1
OR(  %I0.2
)
```

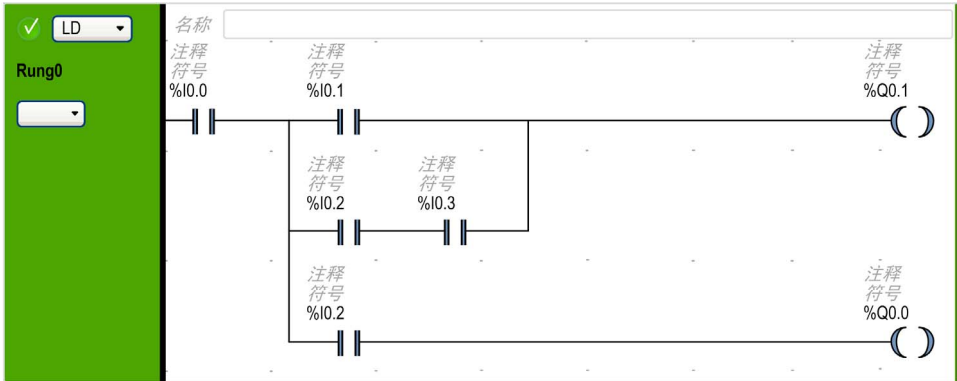
等效梯形图会生成短路错误：



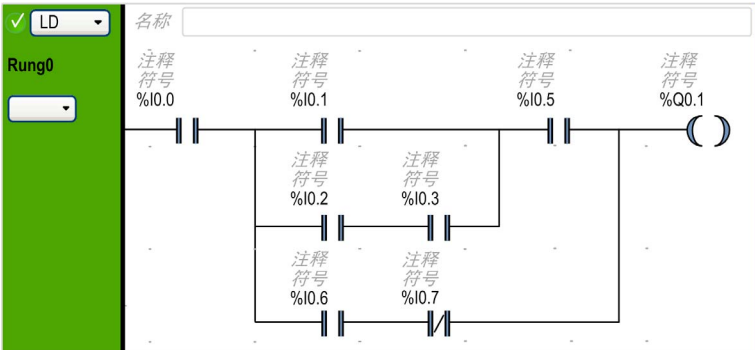
要执行相同的功能，可按照以下方式编写指令：

```
LD    %I0.0
MPS
AND ( %I0.1
OR(  %I0.2
AND  %I0.3
)
)
ST   %Q0.1
MPP
AND  %I0.2
ST   %Q0.0
```

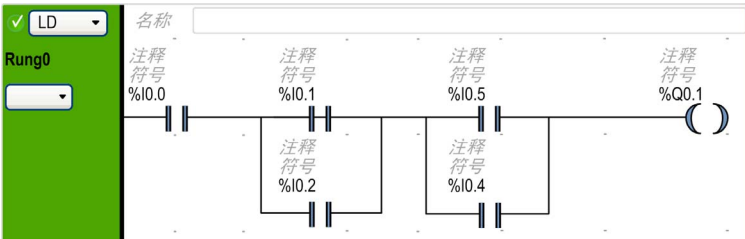
等效梯形图：



如果有多个触点并联，则将它们彼此嵌套起来：



或者，如下所示完全隔开触点。



节 6.10

指令列表编程

本节包含了哪些内容？

本节包含了以下主题：

主题	页
指令列表程序概述	129
列表指令的操作	131
列表语言指令	132
使用括号	135

指令列表程序概述

简介

以指令列表语言编写的程序包括由 Logic Controller 按顺序执行的一系列指令。每条指令由单个程序行表示，并包括以下成分：

- 行号
- 当前值（仅限在线模式下）
- 指令操作符
- 操作数
- 可选注释

指令列表程序示例

下面是指令列表程序的示例。

名称			
0000	LD	%M1	加载位 1
0001	AND (%I0.1	启动分支并加载输入位 1
0002	OR (%I0.2	加载输入位 2
0003	ANDN	%I0.3	加载输入位 3 并反转
0004)		注释
0005)		注释
0006	ST	%Q0.0	设置输出位 0

行号

四位数行号在创建新程序行时生成，并由 SoMachine Basic 自动管理。

当前值

当 SoMachine Basic 处于在线模式（参见第 26 页）（已连接到 Logic Controller 且程序正在运行）时，SoMachine Basic 会在 IL 编辑器窗口中显示对象类型的当前值。

随即更新显示的这些对象的值。

指令操作符

指令操作符是助记符号（称为操作符），用于标识要使用操作数执行的操作。典型操作符可以指定布尔操作和数字操作。

例如，在上述示例程序中，LD 是 LOAD 操作符的助记符号。LOAD 指令可以将操作数 %M1 的值放置（载入）到称为布尔累加器的内部寄存器中。

操作符的类型主要有 2 种：

- 测试操作符
这些指令设置或测试执行操作所需的条件。例如 LOAD (LD) 和 AND。
- 操作操作符
这些指令执行作为之前的逻辑的结果的操作。例如 STORE (ST) 和 RESET (R) 等分配操作符。

操作符与操作数一起构成了指令。

操作数

操作数是对象、地址或符号，表示一条指令中程序可以操纵的值。例如，在上述示例程序中，操作数 %M1 是一个分配了 Logic Controller 的嵌入式输入的地值的地址。一条指令可能有 0 至 3 个操作数，具体取决于指令操作符的类型。

操作数可以代表以下内容：

- 控制器输入和输出（例如传感器、按钮和继电器）。
- 预定义的系统功能（例如定时器和计数器）。
- 算术、逻辑、比较和数字运算。
- 控制器内部变量（例如系统位和系统字）。

注释

将注释添加到指令列表程序

步骤	操作
1	可以选择单击第一行 0000 上方的梯级顶部显示的注释框，然后输入梯级的注释。
2	插入一个指令行。
3	单击指令右侧的 注释 区域。
4	输入注释，然后按 Enter。

列表指令的操作

简介

指令列表二进制指令通常只有一个显式操作数；另一个操作数是隐含的。隐式操作数是布尔累加器中的值。例如，在指令 LD %I0.1 中，%I0.1 是显式操作数。累加器中已加载隐式操作数，而且 %I0.1 的值会覆盖累加器的前一个值。这个值现在会成为后续指令的隐式值。

操作

指令列表指令对累加器的内容和显式操作数执行指定的操作，并用该结果替换累加器的内容。例如，操作 AND %I1.2 在累加器的内容与输入 1.2 之间执行逻辑 AND 运算，并用此结果替换累加器的内容。

所有布尔指令（Load、Store 和 Not 除外）都对 2 个操作数进行运算。2 个操作数的值可以是 True 或 False，指令的程序执行将产生单个值：True 或 False。Load 指令可以将操作数的值放在累加器中，而 Store 指令则可以将累加器中的值传送给操作数。Not 指令没有显式操作数，仅用于反转累加器的状态。

支持的指令列表指令

下表显示了指令列表语言中选择的指令：

指令类型	示例	功能
布尔指令	LD %M10	将内部位 %M10 的值加载到累加器
块指令	IN %TM0	启动定时器 %TM0
字指令	[%MW10 := %MW50+100]	加法运算
程序指令	SR5	调用子程序 #5

列表语言指令

简介




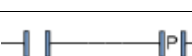
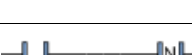
指令列表语言包含以下指令类型或指令块：

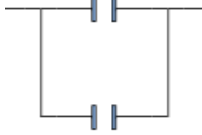
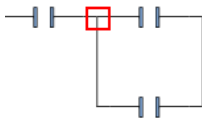
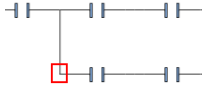

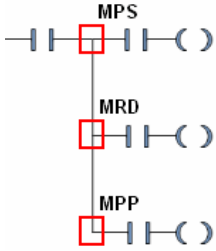

- 测试指令
- 操作指令
- 功能块

本部分标识并介绍列表编程的指令。

测试指令







下表介绍了指令列表语言中的测试指令。

助记符号	名称	对应的图形元素	功能
LD	Load		将操作数的布尔值加载到累加器。
LDN	Load Not		将操作数的布尔值的反值加载到累加器。
LDR	Load Rising		操作数的布尔值从 0 更改为 1 时（上升沿），将该值加载到累加器。因此累加器的值会加载为 0，直至操作数的值下一次从 0 转换为 1。
LDF	Load Falling		操作数的布尔值从 1 更改为 0 时（下降沿），将该值加载到累加器。因此累加器的值会加载为 1，直至操作数的值下一次从 1 转换为 0。
AND	并且		布尔结果等于前一个指令的布尔结果（此结果存储在累加器中）与操作数的状态之间的 AND 逻辑。然后，指令的结果会自行隐式加载到累加器中，并且覆盖之前的值。
ANDN	And Not		布尔结果等于前一个指令的布尔结果（此结果存储在累加器中）与操作数的逆（负）状态之间的 AND 逻辑。然后，指令的结果会自行隐式加载到累加器中，并且覆盖之前的值。
ANDR	And Rising		布尔结果等于前一个指令的布尔结果与检测到操作数的上升沿（1 = 上升沿）之间的与逻辑。然后，指令的结果会自行隐式加载到累加器中，并且覆盖之前的值。
ANDF	And Falling		布尔结果等于前一个指令的布尔结果与检测到操作数的下降沿（1 = 下降沿）之间的与逻辑。然后，指令的结果会自行隐式加载到累加器中，并且覆盖之前的值。

助记符号	名称	对应的图形元素	功能
OR	或		布尔结果等于前一个指令的布尔结果与操作数的状态（此状态存储在累加器中）之间的 OR 逻辑。
AND(And With		逻辑 AND（最多 32 层括号）。括号指定指令之间的中间逻辑结果，然后将该结果与累加器中的值执行逻辑 AND 运算。
OR(Or With		逻辑 OR（最多 32 层括号）。括号指定指令之间的中间逻辑结果，然后将该结果与累加器中的值执行逻辑 OR 运算。
XOR XORN XORR XORF	Ex Or Ex Or Not Ex Or Rising Ex Or Falling		异或
MPS MRD MPP	Memory Push Store Memory Read Memory Pop		输出操作的分支操作符。
N	Not		操作数的值求反。


操作指令

下表介绍了指令列表语言中的操作指令。

助记符号	名称	对应的图形元素	功能
ST	Store		关联的操作数采用测试区结果的值。
STN	Store Not		关联的操作数采用测试区结果的相反值。
S	将		当测试区的结果为 1 时，关联的操作数设置为 1。
R	复位		当测试区的结果为 1 时，关联的操作数设置为 0。
JMP	跳转		无条件连接带有标签的序列（上部分或下部分）。
前一	子程序		在子程序开始处（子程序调用）连接。
END	End		程序结束。
ENDCN	End Conditional		布尔结果为 0 时，按条件结束程序。

功能块

下表介绍了指令列表语言中的功能块。

名称	对应的图形元素	功能
定时器、计数器、寄存器。		<p>对于每个功能块，都有控制该控制块的指令。</p> <p>使用结构化形式连接功能块输入和输出。</p> <p>注意：功能块的输出不能相互连接（垂直短路）。</p> <p>有关详细信息，请参阅软件对象（参见 <i>SoMachine Basic, 通用功能库指南</i>）。</p>

使用括号

简介

通过 AND 和 OR 逻辑操作符，可使用括号来嵌套逻辑指令。这样就可以在梯形图编辑器中指定发散（分支）。括号与指令关联，如下所述：

- 打开括号与 AND 操作符或 OR 操作符关联。
- 关闭括号是每个左括号所必需的指令（不带操作数的操作符）。

使用 AND 指令的示例

以下示例介绍如何在 AND 指令中使用括号：

梯级	指令
0	LD %I0.0 AND %I0.1 OR %I0.2 ST %Q0.0
1	LD %I0.0 AND (%I0.1 OR %I0.2) ST %Q0.1

注意：请参阅可转换性过程（参见第 66 页）以获取等效梯形图。

使用 OR 指令的示例

以下示例介绍如何在 OR 指令中使用括号：

梯级	指令
0	LD %I0.0 AND %I0.1 OR (%I0.2 AND %I0.3) ST %Q0.0

注意：请参阅可转换性过程（参见第 66 页）以获取等效梯形图。

修饰符

下表列出了可以分配给括号的修饰符。

修饰符	功能	示例
N	取反	AND(N 或 OR(N
F	下降沿	AND(F 或 OR(F
R	上升沿	AND(R 或 OR(R
[比较	请参见 比较指令。

注意：“[”修饰符也可以与其他指令搭配使用以作为操作符。有关“[”在其他指令中的更多使用，请参阅数字运算简介。

嵌套括号

可以嵌套最多 32 层括号。

嵌套括号时，请遵循以下规则：

- 每个左括号必须有一个对应的右括号。
- 标签 (%Li:)、子程序 (Sri:)、JMP 指令 (JMP) 和功能块指令不得放在括号之间的表达式里。
- 存储指令 (ST、STN、S 和 R) 不得编在括号之间。
- 堆栈指令 (MPS、MRD 和 MPP) 不能在括号之间使用。

嵌套括号的示例

以下示例介绍如何嵌套括号：

梯级	指令
0	LD %I0.0 AND(%I0.1 OR(N %I0.2 AND %M3))) ST %Q0.0

梯级	指令
1	LD %I0.1 AND (%I0.2 OR (%I0.5 AND %I0.6) AND %I0.4 OR (%I0.7 AND %I0.8)) ST %Q0.0

注意： 请参阅可转换性过程 (参见第 66 页) 以获取等效梯形图。

节 6.11

Grafcet（列表）编程

本节包含了哪些内容？

本节包含了以下主题：

主题	页
Grafcet（列表）编程的说明	139
Grafcet 程序结构	140
如何在 SoMachine Basic 程序中使用 Grafcet 指令	143

Grafcet（列表）编程的说明

简介

通过 SoMachine Basic 中的 Grafcet（列表）编程，可以轻松地将控制序列转换到各个步骤中。可以将控制序列转换到各个 Grafcet 步骤中，然后在采用 Grafcet 指令的程序中使用这些步骤。

Grafcet 步骤的最大数量取决于控制器。活动步数在任意时间仅受总步数的限制。

Grafcet 指令

SoMachine Basic Grafcet 程序具有以下指令：

操作符	操作码	IL 指令	指令名称	相应的图形	描述
=*=	x	=*= x	INITIAL STEP		该指令定义程序中的初始步骤。
=*= POST	不适用	=*= POST	POST PROCESSING (隐式操作数)		该指令定义后期处理和结束顺序处理。
-*-	x	-*- x	STEP		该指令定义程序中用于转换验证的步骤。
#	不适用	#	DEACTIVATE CURRENT STEP (隐式操作数)		该指令停用程序中的当前步骤。
#	x	# x	DEACTIVATE CURRENT STEP and ACTIVATE STEPx		该指令停用程序中的当前步骤并激活步骤 x。
#D	x	#D x	DEACTIVATE CURRENT STEP and STEPx		该指令停用程序中的当前步骤和步骤 x。

x Grafcet 步骤编号（整数，初始值为 1）。

Grafcet 程序结构

简介

SoMachine Basic Grafcet 程序具有以下部分：

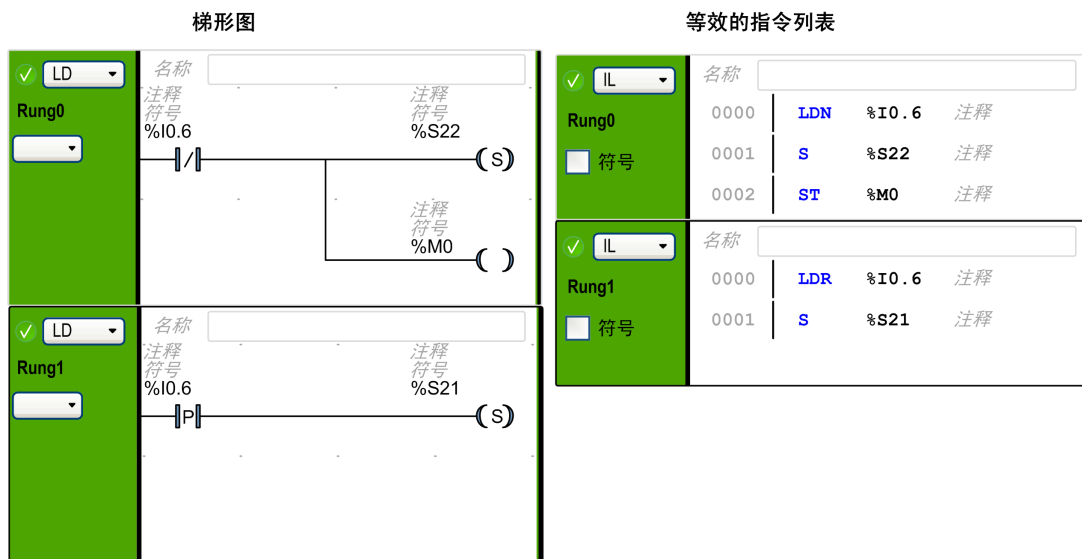
- 预处理
- 顺序处理
- 后处理

预处理

预处理由以下部分组成：

- 电源回流
- 错误管理
- 更改操作模式
- 预置 Grafcet 步骤
- 输入逻辑

在此示例中，系统位 %S21 在输入 %I0.6 的上升沿为 1。此示例禁用了活动步，启用了初始步：



预处理从程序的第一行开始，在第一次出现 **==*** 或 **-*** 指令时结束。

系统位 %S21、%S22 和 %S23 专用于 Grafcet 控制。在预处理过程中，应用程序通常将每个系统位设置为 1（如果需要）。系统在预处理结束时执行相关功能，然后由系统将系统位复位为 0。

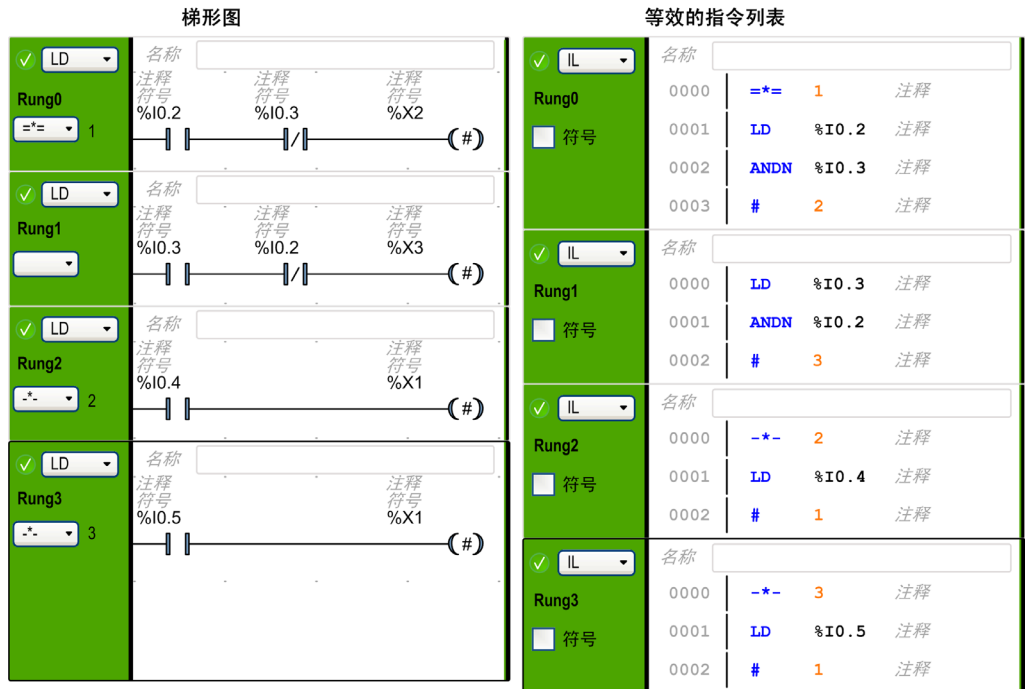
系统位	名称	描述
%S21	Grafcet 初始化	停用所有活动步骤并激活初始步骤。
%S22	Grafcet 重新初始化	停用所有步骤。
%S23	Grafcet 预置	如果在预处理过程中应用程序明确写入了 %Xi 对象，则该位必须设置为 1。如果预处理将该位保持为 1，而不对 %Xi 对象做任何明确的更改，则 Grafcet 冻结（未考虑任何更新）。

序列处理

序列处理在图形中进行（指令表示图形）：

- 步骤
- 与步骤关联的动作
- 转换
- 转换条件

示例：



序列处理在 **POST** 指令执行时或程序结束时结束。

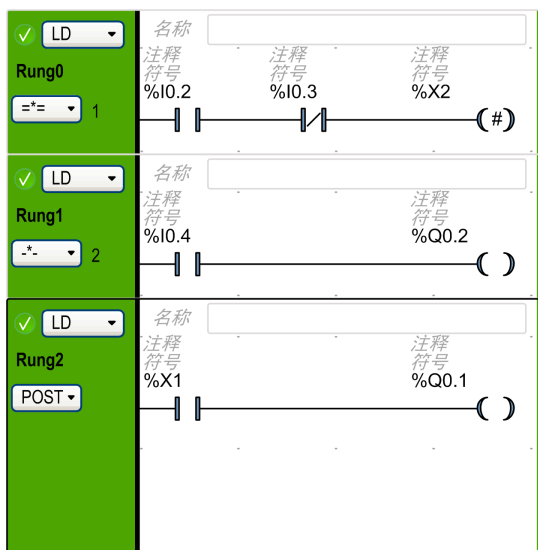
后处理

后处理由以下部分组成：

- 序列处理中用于控制输出的命令
- 针对输出的特定联锁装置

示例：

梯形图



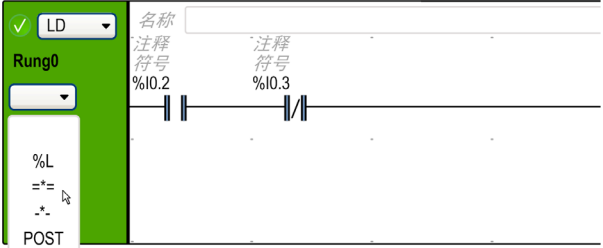
等效的指令列表

名称			
0000	==	1	注释
0001	LD	<code>%I0.2</code>	注释
0002	ANDN	<code>%I0.3</code>	注释
0003	#	2	注释
名称			
0000	-*-	2	注释
0001	LD	<code>%I0.4</code>	注释
0002	ST	<code>%Q0.2</code>	注释
名称			
0000	==	POST	注释
0001	LD	<code>%X1</code>	注释
0002	ST	<code>%Q0.1</code>	注释

如何在 SoMachine Basic 程序中使用 Grafcet 指令

在梯形图中创建 Grafcet 步骤

按照以下步骤在程序中创建 Grafcet 步骤：

步骤	操作
1	<p>在 POU 中，选择某梯级，然后单击梯级顺序标识符 Rungx（其中 <i>x</i> 是 POU 中的梯级号）下的下拉按钮。</p>  <p>结果： 随即出现一个菜单，其中列出 Grafcet 指令。</p>
2	<p>单击列表中的指令，以将梯级定义为初始步、后处理或 Grafcet 程序的特定步骤。</p> <p>结果： 针对 Grafcet 指令设置梯级。按钮中随即出现指令的操作符，按钮后缀中随即出现操作数（步骤数）。</p> <p>注意： 在定义后续 STEP 或 INITIAL STEP 指令时，步骤数以 1 为增量递增。可以在一个程序中仅定义一个 POST 指令；因此 POST 指令没有任何步骤数。</p> <p>要修改步骤数，请双击梯级中的步骤数，并输入新数，然后按 ENTER 键。</p>

在梯形图中激活或停用 Grafcet 步骤

按照以下步骤在程序中激活或停用 Grafcet 步骤：

步骤	操作
1	在 POU 中，选择程序中的某梯级。
2	 单击  （以停用当前步骤并选择性激活指定的步骤）或  （以停用当前步骤并选择性停用指定的步骤），然后将此元素插入梯级的操作区域（请参阅插入图形元素（参见第 117 页））。
3	或者，在梯级中按 ALT+A 以使用 ACTIVATE 指令，或按 ALT+D 以使用 DEACTIVATE 指令。 结果： 此梯级的操作区中随即出现激活或停用梯形图符号。 按 ENTER 键以插入此元素。
4	在此程序梯级中，双击 Grafcet 激活或停用符号上的 地址 字段，然后输入此 Grafcet 位地址（%Xi，其中 i 是步骤编号）。 例如，%x4 是指 Grafcet 程序的步骤 4。如果 %x4 是停用符号的地址，则在使用此符号的梯级的输出为 True 时，将停用步骤 4。 注意： 在每种情况下，都会停用当前步骤。

节 6.12

在线模式下调试

本节包含了哪些内容？

本节包含了以下主题：

主题	页
修改值	146
强制值	147

修改值

简介

当处于在线模式时， **SoMachine Basic** 可让您修改某些对象类型的值。

只有对象具有读 / 写访问权限时，才能在线更新。例如：

- 无法修改模拟量输入的值。
- 可以更新 Timer 功能块的 Preset 参数（%TMO.P 对象）的值。

有关具有读 / 写访问权限的对象类型的信息，请参阅 **SoMachine Basic 通用功能库指南** 或您的硬件平台的 *编程指南* 中的对象描述。

要修改对象的值，请将对象添加到动态数据表（参见第 99 页），并且根据需要设置其属性。

强制值

概述

当处于在线模式时，您可以将数字量输入和数字量输出的值强制为 **False (0)** 或 **True (1)**。此功能可让您将地址设置为特定值，并防止程序逻辑或外部系统更改该值。此功能主要用于调试和微调程序。

要在处于在线模式时强制数字量输入或输出的值，请修改其配置属性或使用动态数据表 (参见第 98 页)。

章 7

试运行

本章包含了哪些内容？

本章包含了以下部分：

节	主题	页
7.1	试运行窗口概述	150
7.2	管理 Logic Controller 的连接	151
7.3	SoMachine Basic 仿真器	156
7.4	备份和恢复控制器存储器	169
7.5	下载和上载程序	171

节 7.1

试运行窗口概述

试运行窗口概述

简介

试运行窗口可用于：

- 登录到逻辑 Logic Controller 或从 Logic Controller 注销。
- 管理 Logic Controller 存储器，例如通过执行备份和恢复操作。
- 管理 Logic Controller 的实时时钟 (RTC)。



- 1 试运行树显示可用的试运行任务。
- 2 右侧区域用于执行试运行任务的操作。

节 7.2

管理 Logic Controller 的连接

本节包含了哪些内容？

本节包含了以下主题：

主题	页
连接到 Logic Controller	152
控制器信息	154
管理 RTC	155

连接到 Logic Controller

概述

单击“试运行”窗口上的**连接**以管理与 Logic Controller 的连接。

连接的控制器

显示 2 个 Logic Controller :

1. 本地设备

显示通过 PC 的物理 COM 端口（例如 COM1）、USB 电缆或虚拟化 COM 端口（通过 USB 串行转换器或蓝牙适配器）连接到 PC 的所有 Logic Controller。

2. 以太网设备


显示可通过以太网（在相同的子网上且不在路由器或任何阻止 UDP 广播的设备下）访问的所有 Logic Controller。此列表包括 SoMachine Basic 自动检测到的 Logic Controller，以及您手动添加的任何控制器。


手动添加以太网控制器

要将 Logic Controller 手动添加到**以太网设备**列表中，请执行以下操作：

步骤	操作
1	在 远程查找 字段中，键入 Logic Controller 的 IP 地址，例如 12.123.134.21
2	单击 添加 ，将该设备添加到 以太网设备 列表中。

连接到 Logic Controller

使用 （刷新按钮）来刷新连接的设备的列表。

使用 （启动闪烁的 LED 按钮）使所选控制器的 LED 闪烁，以便通过该控制器闪烁的 LED 来对其进行物理识别。再次单击此按钮来停止闪烁的 LED。

要登录到 Logic Controller，请执行以下操作：

步骤	操作
1	在 本地设备 或 以太网设备 列表中选择 一个 Logic Controller 。
2	单击 登录 按钮来登录到所选的控制器。 如果 Logic Controller 受密码保护，请键入密码并单击 确定 进行连接。 随即显示状态栏，其中显示连接进度。
3	建立连接后，当前存储在 Logic Controller 中的应用程序保护状态会显示在窗口的 选择的控制器 区域中。 成功建立连接后，Logic Controller 的详细信息会显示在窗口的 选择的控制器 区域中： <ul style="list-style-type: none">● 固件版本● Logic Controller 型号● 已连接到 Logic Controller 的所有扩展模块的型号● SoMachine Basic 与 Logic Controller 之间当前连接状态。
4	SoMachine Basic 检查 Logic Controller 的硬件配置是否兼容当前项目的配置。 如果兼容，则可以将应用程序下载到控制器。将会启用 PC 到控制器（下载） 按钮，您可以继续执行下载应用程序（参见第 172 页）。

控制器信息

概述

单击**试运行**窗口左侧区域中的**控制器信息**，可显示关于 Logic Controller 当前状态的以下信息：

- **可执行 RAM**：如果有有效的应用程序存储在 Logic Controller 的随机存取存储器 (RAM)，则会选中此选项。
- **受保护的 RAM**：如果 Logic Controller RAM 中的应用程序受密码保护，则会选中此选项。
- **强制 I/O**：如果 Logic Controller 中的 1 个或多个数字量输入或输出被强制为特定值 (参见第 99 页)，则会选中此选项。
- **状态**：Logic Controller 的当前状态。
也可以通过测试系统字 %SW6，从程序中获得此信息。有关控制器状态的详细信息，请参阅 Logic Controller 的 *编程指南*。
- **最近一次停止**：Logic Controller 最近一次停止 (“停止”、“暂停”等) 的日期和时间。
也可以通过测试系统字 %SW54 至 %SW57，从程序中获得此信息。
- **最近一次停止的原因**：显示 Logic Controller 最近一次停止的原因。
也可以通过测试系统字 %SW58，从程序中获得此信息。
- **扫描时间 (微秒)**：以下扫描时间：
 - **最小值 (毫秒)**：自 Logic Controller 上次电源接通以来最短的扫描时间。
也可以通过测试系统字 %SW32 (毫秒)，从程序中获得此信息。
 - **当前 (毫秒)**：当前扫描时间。
也可以通过测试系统字 %SW30 (毫秒)，从程序中获得此信息。
 - **最大值 (毫秒)**：自 Logic Controller 上次电源接通以来最长的扫描时间。
也可以通过测试系统字 %SW31 (毫秒)，从程序中获得此信息。
- **控制器时间**：只有 Logic Controller 具有实时时钟 (RTC) 时，才会显示以下信息：
 - **日期 (DD/MM/YYYY)**：Logic Controller 中存储的当前日期。
也可以通过测试系统字 %SW56 和 %SW57，从程序中获得此信息。
 - **时间 (HH:MM:SS)**：Logic Controller 中存储的当前时间。
也可以通过测试系统字 %SW54 和 %SW55，从程序中获得此信息。

日期和时间的格式与为 PC 指定的格式相同。

管理 RTC

概述

RTC 管理窗口用于设置 Logic Controller 的实时时钟 (RTC)。只有当 SoMachine Basic 连接到支持 RTC 的 Logic Controller 时，此功能才可用。

更新 RTC

步骤	操作
1	选择 试运行 窗口左侧区域的 RTC 管理 选项。
2	如果处于在线模式下，则会显示 当前的控制器时间 。 选择用于设置 Logic Controller 时间的模式： <ul style="list-style-type: none">● 手动：此模式会显示日期和时间，允许用户手动选择要在 Logic Controller 中设置的日期和时间。● 自动：此模式会将 Logic Controller 中的时间设置为安装了 SoMachine Basic 的 PC 的当前时间。
3	单击 应用 。

节 7.3

SoMachine Basic 仿真器

本节包含了哪些内容？

本节包含了以下主题：

主题	页
SoMachine Basic 模拟器概述	157
SoMachine Basic 模拟器 I/O 管理器窗口	159
SoMachine Basic 模拟器时间管理窗口	161
使用 SoMachine Basic 模拟器修改值	163
如何使用 SoMachine Basic 模拟器	168

SoMachine Basic 模拟器概述

简介

SoMachine Basic 模拟器可让您：


- 模拟 PC、Logic Controller 与任何扩展模块之间的连接。
- 运行和测试程序，而无需将 Logic Controller 和扩展模块物理连接到 PC。

工作原理是模拟器重复控制器的行为，且实际上是与 SoMachine Basic 连接的虚拟控制器。启动模拟器后，您便可以执行连接、运行、停止及连接到物理控制器时通常可实现的其他相关操作。

访问 SoMachine Basic 模拟器

您可以通过以下任一方法启动模拟器：

- 单击试运行任务区域中的**启动模拟器**按钮。
- 在**试运行**窗口中按 CTRL+B。

- 单击 SoMachine Basic 工具栏中的 （启动模拟器按钮）。

启动模拟器之前，请确保程序有效。否则，模拟器启动将中断，屏幕上将显示一条检测到编译错误的消息。

SoMachine Basic 模拟器窗口

SoMachine Basic 模拟器具有以下 2 个窗口：

- **模拟器时间管理窗口**
可让您控制控制器的 RTC，以便模拟时间推移及其对受 RTC 影响的逻辑结构的影响。
- **模拟器 I/O 管理器窗口**
可让您管理控制器和扩展模块的输入 / 输出状态。

PC 与虚拟 Logic Controller 成功建立连接（请参阅如何使用 SoMachine Basic 模拟器（参见第 168 页））后，屏幕上将显示 SoMachine Basic 模拟器窗口：



- 1 模拟器 I/O 管理器窗口 (参见第 159 页)
- 2 模拟器时间管理窗口 (参见第 161 页)

SoMachine Basic 模拟器 I/O 管理器窗口

概述

模拟器 I/O 管理器窗口具有以下组件：

- LED 状态：
监视模拟控制器的 LED 状态。
- 输入 / 输出状态：
控制程序正在运行时的输入和输出。

模拟器 I/O 管理器窗口

下图显示了模拟器 I/O 管理器窗口：



单击此窗口左上角的固定符号，将此窗口固定在前台或取消固定。

单击此窗口右上角的最小化符号，将此窗口最小化到任务栏中。

LED 状态

PWR、RUN、ERR 和 STAT LED 可在 SoMachine Basic 模拟器 I/O 管理器窗口中进行模拟，因为它们将显示在连接的本体控制器上。

下面是模拟 Logic Controller 的模拟器 I/O 管理器窗口中显示的 LED 状态：

LED	状态信息
PWR	指示模拟 Logic Controller 是否已加电。
RUN	指示模拟 Logic Controller 的 RUN 状态（参阅 LED 显示表）。
ERR	指示模拟 Logic Controller 的 ERR 状态（参阅 LED 显示表）。
STAT	STAT LED 的运行由用户逻辑定义。

输入 / 输出状态

模拟器 I/O 管理器窗口可让您监视和控制程序正在运行时控制器和扩展模块的 I/O。

输入和输出以数字列表形式显示。此列表取决于所选控制器和扩展模块的 I/O。例如，如果控制器具有 n 个数字量输入，则数字列表将显示从 0 开始到 $(n-1)$ 的数，其中每个数与相应输入通道的数字量输入对应。

对于控制器，显示的 I/O 有：

- **IN:** 数字量输入。
- **Out:** 数字量输出。
- **ANA:** 模拟量输入。

对于扩展模块，显示的 I/O 有：

- **IN:** 数字量 / 模拟量输入。
- **OUT:** 数字量 / 模拟量输出。

注意： 模拟量 I/O 与其当前值一起显示在模拟量输入数值右侧。

数字量 I/O 状态通过 I/O 数值的文本颜色标识：

- 绿色：I/O 设置为 1。
- 黑色：I/O 设置为 0。

模拟量 I/O 状态通过以下值标识：

- -（连字符）：未配置 I/O。
- 数：I/O 的当前值。

SoMachine Basic 模拟器时间管理窗口

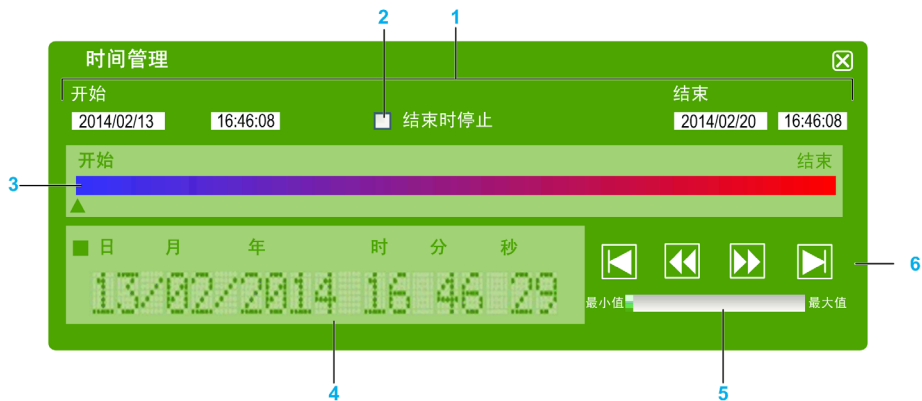
概述

模拟器时间管理窗口包含以下组件：

- 在模拟器中执行程序的日期 / 时间模拟范围：
 - 开始日期和时间
 - 结束日期和时间
 - 结束时停止复选框（达到结束日期和时间时停止执行程序）
- 时间控制滚动条：
 - 手动向前或向后移动来模拟时间推移
- 日期和时间显示：
 - 模拟器的模拟 RTC 的日期和时间
- 控制按钮：
 - 复位、向后跳转、向前跳转或结束与 RTC 关联的时间管理
- 增量条：
 - 调整相对于实际时间的模拟时间推移的速率

模拟器时间管理窗口

此图描述模拟器时间管理窗口：



- 1 日期 / 时间模拟范围（开始 - 结束）
- 2 （日期 / 时间范围）结束时停止复选框
- 3 时间控制滚动条
- 4 RTC 日期和时间
- 5 增量条
- 6 时间推移控制按钮

模拟器日期 / 时间模拟范围

此模拟范围可让您设定和控制模拟器的 RTC。当您将模拟器设置为“运行”状态时，便会使用**开始**日期和时间字段设置 RTC。**结束**日期和时间字段可设定模拟的结束时间。如果选中**结束时停止**复选框，则模拟器将在模拟范围到期时进入“停止”状态。否则，模拟器和 RTC 将继续运行，直到您使用 SoMachine Basic 手动停止模拟器。

时间控制滚动条

时间控制滚动条可让您手动控制设定了模拟范围的日期和时间。单击并按住鼠标右键，同时指向滚动条下方的箭头并向右移动鼠标可前移 RTC 的时间和日期。执行相同的操作并向左移动鼠标可后移 RTC 的时间和日期。

RTC 日期和时间





RTC 日期和时间区域显示 RTC 的值，因为它与持续的模拟相关。当您将模拟器设置为“运行”状态时，便会通过**开始**日期和时间设定 RTC 的初始时间。此后，将使用模拟器中 RTC 的持续计时更新显示。您可以使用时间控制滚动条或时间推移速度控制按钮更改 RTC。

增量条

增量条可让您设定在使用时间推移控制按钮时向前或向后跳转 RTC 值的相对增量。单击增量条可以设置相对于所设定的模拟范围的各种增量。

时间推移控制按钮

您可以使用控制按钮影响 RTC 值，因此控制其对在模拟器中运行的程序的影响，如下所示：

图形元素	命令	描述
	初始化	可让您将日期和时间复位为 开始 时间 / 日期字段中设置的日期和时间。
	向前跳转	可让您将时间和日期从其当前值按增量条设定的增量前移。
	向后跳转	可让您将时间和日期从其当前值按增量条设定的增量后移。
	结束	可让您将日期和时间跳转到 结束 时间 / 日期字段中设置的日期和时间。

使用 SoMachine Basic 模拟器修改值

概述

在在线模式下时，SoMachine Basic 模拟器 I/O 管理器窗口可让您：

- 修改输入的值。
- 跟踪输出。

修改数字量输入的值

按照以下步骤使用单击操作修改数字量输入值：

步骤	操作
1	单击模拟器 I/O 管理器窗口中的数字量输入数，以更改离散量输入值。 结果： 输入数的文本颜色更改。数字量输入值通过文本颜色标识： <ul style="list-style-type: none"> ● 绿色：I/O 设置为 1。 ● 黑色：I/O 设置为 0。
2	再次单击同一输入数可切换该值。


按照以下步骤执行修改数字量输入值的批量操作：

步骤	操作																
1	双击模拟器 I/O 管理器窗口中的数字量输入数。 结果： 屏幕上随即出现列出所有数字量输入的 Set Discrete Inputs 窗口： <div data-bbox="304 873 853 1416" style="border: 1px solid black; padding: 10px; margin: 10px 0;"> <p>设置离散量输入</p> <p>所有离散量输入</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="width: 50%;">输入</th> <th style="width: 50%;">设置为 I/O</th> </tr> </thead> <tbody> <tr> <td>%I0.0</td> <td style="text-align: center;"><input checked="" type="checkbox"/></td> </tr> <tr> <td>%I0.1</td> <td style="text-align: center;"><input checked="" type="checkbox"/></td> </tr> <tr> <td>%I0.2</td> <td style="text-align: center;"><input type="checkbox"/></td> </tr> <tr> <td>%I0.3</td> <td style="text-align: center;"><input type="checkbox"/></td> </tr> <tr> <td>%I0.4</td> <td style="text-align: center;"><input type="checkbox"/></td> </tr> <tr> <td>%I0.5</td> <td style="text-align: center;"><input type="checkbox"/></td> </tr> <tr> <td>%I0.6</td> <td style="text-align: center;"><input type="checkbox"/></td> </tr> </tbody> </table> <p>操作</p> <p style="text-align: center;"> <input type="button" value="全部设置为 0"/> <input type="button" value="全部设置为 1"/> </p> <p style="text-align: center;"> <input type="button" value="帮助"/> <input type="button" value="取消"/> <input type="button" value="正常"/> </p> </div>	输入	设置为 I/O	%I0.0	<input checked="" type="checkbox"/>	%I0.1	<input checked="" type="checkbox"/>	%I0.2	<input type="checkbox"/>	%I0.3	<input type="checkbox"/>	%I0.4	<input type="checkbox"/>	%I0.5	<input type="checkbox"/>	%I0.6	<input type="checkbox"/>
输入	设置为 I/O																
%I0.0	<input checked="" type="checkbox"/>																
%I0.1	<input checked="" type="checkbox"/>																
%I0.2	<input type="checkbox"/>																
%I0.3	<input type="checkbox"/>																
%I0.4	<input type="checkbox"/>																
%I0.5	<input type="checkbox"/>																
%I0.6	<input type="checkbox"/>																

步骤	操作
2	<p>在 Set Discrete Inputs 窗口的操作区域中，单击：</p> <ul style="list-style-type: none"> ● Set all to 0: 将所有输入的值设置为 0。 ● Set all to 1: 将所有输入的值设置为 1。 <p>结果: 如果选中此复选框，则输入值设置为 1。如果不选中，则输入值设置为 0。</p>
3	或者，在 Set Discrete Inputs 窗口的 All Discrete Inputs 区域中，单击与输入对应的复选框，以便单独修改这些值。
4	单击 确定 以保存更改，并从 Set Discrete Inputs 窗口退出。

修改模拟量输入的 I/O 值

按照以下步骤修改模拟量输入值：

步骤	操作
1	<p>双击模拟器 I/O 管理器窗口中的模拟量输入数。</p> <p>结果: 屏幕上随即出现列出所有模拟量输入的 Set Analog Inputs 窗口：</p> 
2	在 Set Analog Inputs 窗口的 All Analog Inputs 区域中，双击 Change Value 列中与要修改的输入对应的值字段。
3	输入 0...1023 范围中的值，然后按 ENTER 键。


步骤	操作
4	或者，在 Set Analog Inputs 窗口中，从 输入 列表选择一个输入，然后移动 Change Value 区域中的滑块，以在 0...1023 之间调整输入值。 在从左到右移动滑块时，值增大，反之亦然。
5	单击 确定 以保存更改，并从 Set Analog Inputs 窗口退出。

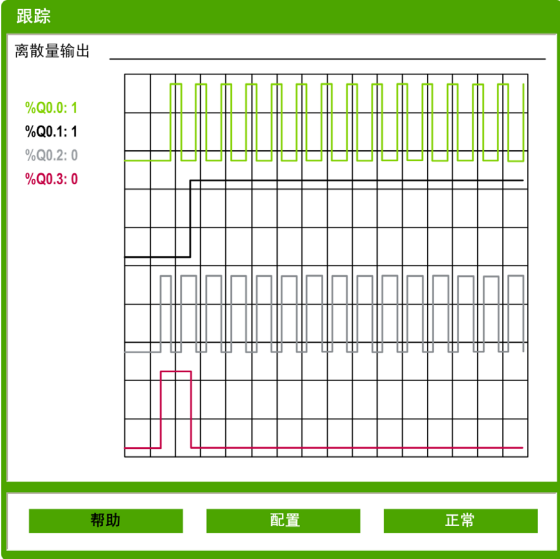
跟踪输出

输出值取决于程序；因此不能修改这些值，但 SoMachine Basic 模拟器可让您跟踪数字量和模拟量输出。

按照以下步骤修改模拟量输入值：

步骤	操作
1	<p>双击模拟器 I/O 管理器窗口中的输出数。 结果： 屏幕上随即出现 Tracing 窗口。</p> 

步骤	操作																
2	<p>单击 Configure 按钮以选择要跟踪的输出。 结果: 屏幕上随即出现 Tracing Configuration 窗口。</p>  <p>跟踪配置</p> <p>选择要跟踪的输出</p> <table border="1" data-bbox="340 337 754 604"> <thead> <tr> <th>选择</th> <th>输出</th> </tr> </thead> <tbody> <tr> <td><input checked="" type="checkbox"/></td> <td>%Q0.0</td> </tr> <tr> <td><input checked="" type="checkbox"/></td> <td>%Q0.1</td> </tr> <tr> <td><input checked="" type="checkbox"/></td> <td>%Q0.2</td> </tr> <tr style="background-color: #92d050;"> <td><input checked="" type="checkbox"/></td> <td>%Q0.3</td> </tr> <tr> <td><input type="checkbox"/></td> <td>%Q0.4</td> </tr> <tr> <td><input type="checkbox"/></td> <td>%Q0.5</td> </tr> <tr> <td><input type="checkbox"/></td> <td>%Q0.6</td> </tr> </tbody> </table> <p>采样间隔 <input type="text" value="1"/> 秒</p> <p>帮助 取消 正常</p>	选择	输出	<input checked="" type="checkbox"/>	%Q0.0	<input checked="" type="checkbox"/>	%Q0.1	<input checked="" type="checkbox"/>	%Q0.2	<input checked="" type="checkbox"/>	%Q0.3	<input type="checkbox"/>	%Q0.4	<input type="checkbox"/>	%Q0.5	<input type="checkbox"/>	%Q0.6
选择	输出																
<input checked="" type="checkbox"/>	%Q0.0																
<input checked="" type="checkbox"/>	%Q0.1																
<input checked="" type="checkbox"/>	%Q0.2																
<input checked="" type="checkbox"/>	%Q0.3																
<input type="checkbox"/>	%Q0.4																
<input type="checkbox"/>	%Q0.5																
<input type="checkbox"/>	%Q0.6																
3	<p>在 Select 复选框列中，单击与要跟踪的输出对应的复选框。</p>																
4	<p>从下拉菜单中选择 Sample Interval 以设置输出跟踪的采样时间间隔：</p> <ul style="list-style-type: none"> ● 1 second ● 5 秒 ● 10 秒 ● 20 秒 																

步骤	操作
5	<p>单击确定以保存，并从 Tracing Configuration 窗口退出。</p> <p>结果：所选输出已添加到 Tracing 窗口，其中显示模拟值输出的跟踪情况：</p>  <p>跟踪</p> <p>离散量输出</p> <p>%Q0.0: 1 %Q0.1: 1 %Q0.2: 0 %Q0.3: 0</p> <p>帮助 配置 正常</p>
6	单击 确定 以从 Tracing 窗口退出。

如何使用 SoMachine Basic 模拟器

过程

按照以下步骤来运行 SoMachine Basic 模拟器以测试程序：

步骤	操作
1	检查状态区域中的状态消息，确保程序是有效的（有关详细信息，请参阅状态区域（参见第 42 页））。程序状态应该是 无错误 。 程序状态为 建议 时，仍然可以运行 SoMachine Basic 模拟器。
2	启动模拟器（参阅访问模拟器（参见第 157 页））。
3	运行控制器。 在 试运行 窗口中，在试运行树中选择 连接 ，然后在试运行任务区域中单击 运行控制器 按钮。
4	使用模拟器主窗口对程序发出命令（参阅控制按钮（参见第 162 页））。
5	在模拟器主窗口中，检查 LED 的状态（参阅 LED 显示）。
6	在模拟器 I/O 管理器窗口中，检查输入 / 输出的状态（参阅输入 / 输出状态（参见第 160 页））。
7	在模拟器 I/O 管理器窗口中，检查 LED 的状态（参阅 LED 状态（参见第 159 页））。
8	根据需要修改 I/O 值（参阅使用模拟器修改值（参见第 163 页））。
9	根据需要跟踪输出（参阅跟踪输出（参见第 165 页））。
10	停止控制器。 在 试运行 窗口中，在试运行树中选择 连接 ，然后在试运行任务区域中单击 停止控制器 按钮。
11	停止模拟器。 在 试运行 窗口中，在试运行树中选择 连接 ，然后在试运行任务区域中单击 停止控制器 按钮，或者按 CTRL+W 退出模拟器。

节 7.4

备份和恢复控制器存储器

备份和恢复控制器存储器

概述

SoMachine Basic 使您能够备份或恢复控制器存储器。通过控制器存储器管理，可以仅备份或恢复存储器位和存储器字。其他对象（例如定时器、计数器，等等）则通过应用程序下载和上传来管理。

只有在在线模式下才能使用备份和恢复选项。

备份到 PC


按照以下步骤来将控制器存储器备份至 PC：

步骤	操作
1	在 试运行 选项卡上，选择 存储器管理 。
2	在 操作 下，选择 从控制器进行备份 。
3	在 目标文件 下，选择 PC 。单击“浏览”按钮，导航到要写入备份文件的文件夹，然后输入备份文件 (*.csv) 的名称。
4	选择 备份存储器变量 以在备份中包括 Logic Controller 存储器。指定要包括在备份中的 第一个存储器位、上一个存储器位、第一个存储器字和上一个存储器字 。
5	单击 从控制器进行备份 按钮以开始备份操作。 随即显示报告窗口，其中显示与存储器备份操作相关的信息或检测到的错误消息的列表。如果操作成功，此列表的最后一行会显示 存储器备份成功 。如果存储器备份操作不成功，那么报告窗口的最后一行中会显示相关的消息，并且会自动删除未完成的文件 (*.csv)。

注意：您可以在控制器处于 **[运行]** 状态时启动备份。但是，根据您指定要包括在备份中的存储器变量的数量，在逻辑扫描之间可能无法完成备份。因此，备份未必前后连贯，这样便可以在两次扫描之间修改存储器变量的值。如果希望变量具有一组一致的值，则可能需要先使控制器进入 **[停止]** 状态。

从 PC 恢复

按照以下步骤来从 PC 恢复控制器存储器：

步骤	操作
1	<p>确保控制器处于“已停止”状态。 如果控制器处于“运行”状态，则执行以下任意一个操作来停止控制器：</p> <ul style="list-style-type: none"> 单击位于 SoMachine Basic 窗口顶部的工具栏中的停止控制器图标 。 在试运行窗口中，在试运行树中选择连接，然后在试运行任务区域中单击停止控制器按钮。
2	在 试运行 选项卡上，选择 存储器管理 。
3	在 操作 列表中，选择 恢复到控制器 。
4	在 源文件 下，选择 PC 以便从 PC 上存储的文件恢复控制器存储器。单击“浏览”按钮，导航到包含该文件的文件夹，然后选择以前备份的文件 (*.csv)。
5	<p>单击恢复到控制器，开始恢复操作。 随即显示报告窗口，其中显示与存储器恢复操作相关的信息或检测到的错误消息的列表。如果操作成功，此列表的最后一行会显示存储器恢复成功。如果存储器备份操作不成功，那么报告窗口的最后一行中会显示相关的消息。</p>

如果在应用程序数据恢复期间出现断电或通讯中断，您的机器可能无法正常工作。如果出现断电或通讯中断，请再次尝试恢复。

注意

设备无法操作

- 恢复开始后，请勿中断应用程序数据的恢复。
- 在恢复成功完成之前，请勿将机器投入使用。

不遵循上述说明可能导致设备损坏。

节 7.5

下载和上载程序

本节包含了哪些内容？

本节包含了以下主题：

主题	页
下载和上载应用程序	172
控制器更新	173

下载和上载应用程序

下载应用程序

按照以下步骤来将当前存储在 SoMachine Basic 中的应用程序下载到 Logic Controller:

步骤	操作
1	在 试运行 窗口的试运行树中，单击 连接 。
2	在 本地设备 或 以太网设备 列表中选择 一个 Logic Controller。
3	单击 登录 按钮来登录到所选的控制器。 如果 Logic Controller 受密码保护，请键入密码并单击 确定 进行连接。
4	单击 PC 到控制器（下载） 。 如果 PC 到控制器（下载） 按钮不可用，请确认是否发生了以下情况： <ul style="list-style-type: none"> ● 存储在 Logic Controller 中的应用程序是否与当前 SoMachine Basic 应用程序完全相同。 ● Logic Controller 系统的硬件配置是否与 SoMachine Basic 应用程序中的当前配置不兼容。
5	如果应用程序已配置为在“ 运行 ”状态下启动，则会显示存在危险的消息，提示用户确认是否像这样来配置应用程序。 单击 确定 以确认下载应用程序，或者单击 取消 并修改配置。
6	单击 确定 以继续传输并覆盖当前 Logic Controller 应用程序。 结果： 随即出现一个状态栏，其中显示了连接状态。
7	要运行已下载的应用程序，请单击 运行控制器 ，然后单击 确定 以确认操作。 如果出现一条消息，通知用户操作模式无法更改，则单击 关闭 ，并且检查 Logic Controller 上的“ 运行 / 停止 ”开关是否处于“ 停止 ”位置；如果已配置运行 / 停止输入，也可能意外导致控制器无法进入“ 运行 ”状态。否则，请参阅 Logic Controller 的 <i>硬件指南</i> 以获取详细信息。

上载应用程序

按照以下步骤来将当前存储在 Logic Controller 中的应用程序上载到 SoMachine Basic:

步骤	操作
1	在 试运行 窗口的试运行树中，单击 连接 。
2	在 本地设备 或 以太网设备 列表中选择 一个 Logic Controller。
3	单击 登录 按钮来登录到所选的控制器。 如果 Logic Controller 受密码保护，请键入密码并单击 确定 进行连接。
4	单击 控制器到 PC（上传） 。 如果 控制器到 PC（上传） 按钮不可用，请确认是否发生了以下情况： <ul style="list-style-type: none"> ● 存储在 Logic Controller 中的应用程序是否与当前 SoMachine Basic 应用程序完全相同。 ● Logic Controller 系统的硬件配置是否与 SoMachine Basic 应用程序中的当前配置不兼容。
5	单击 确定 以确认从 Logic Controller 上传。 结果： 随即出现一个状态栏，其中显示了连接状态。传输完成时，应用程序会从 Logic Controller 上传到 SoMachine Basic。

控制器更新

概述

可以直接从 SoMachine Basic 或使用 SD 卡将固件更新下载到 Logic Controller。

将固件更新发送到 Logic Controller

执行固件更改会删除设备中现有的应用程序，包括闪存中的引导应用程序。

注意

应用程序数据丢失

- 在尝试固件更改之前需备份应用程序，将其备份到 PC 的硬盘。
- 固件更改成功后，恢复设备的应用程序。

不遵循上述说明可能导致设备损坏。

按照以下步骤将固件更新发送到 Logic Controller:

步骤	操作
1	登录 (参见第 152 页) 到 Logic Controller。 如果 Logic Controller 受密码保护，请键入密码并单击 确定 进行连接。
2	单击 试运行 → 固件更新 。
3	单击 选择固件文件 框旁边的“浏览”按钮，然后选择要下载到控制器的固件文件 (*.mfw)。
4	单击 确定 。 结果： 固件下载开始。状态或检测到的错误消息会显示在 详细信息 框中。

如果在应用程序传输或固件更改的过程中出现断电或通讯中断，那么您的设备可能无法正常工作。如果出现断电或通讯中断，请再次尝试传输。

注意

设备无法操作

- 传输一旦开始，不要中断应用程序的传输或固件更改。
- 在传输成功完成之前不要将设备投入使用。

不遵循上述说明可能导致设备损坏。

章 8

保存项目并关闭 SoMachine Basic

本章包含了哪些内容？

本章包含了以下主题：

主题	页
保存项目	176
项目另存为模板	177
关闭 SoMachine Basic	178


保存项目

概述


SoMachine Basic 项目可以另存为本地 PC 上的文件。此文件采用扩展名 *.smbp 且包含：

- 程序选项卡上所含程序的源代码
- 配置选项卡上包含的当前硬件配置
- 在 SoMachine Basic 项目中的设置和首选项。

保存项目

步骤	操作
1	单击工具栏上的 保存  或按 Ctrl-S
2	如果这是首次保存项目，请浏览并选择要存储项目文件的文件夹。
3	输入项目文件的名称，然后单击 保存 。

使用另一个名称保存项目

步骤	操作
1	单击工具栏上 保存按钮  旁的菜单箭头，然后选择 另存为 。
2	浏览并选择要存储项目文件的文件夹。
3	输入项目文件的新名称，然后单击 保存 。

项目另存为模板

概述

SoMachine Basic 项目可以另存为模板。随后项目会在起始页 (参见第 34 页) 的模板选项卡中列出。随后可以使用该项目作为新项目的起点。

项目另存为模板

步骤	操作
1	单击工具栏上的保存按钮  旁的菜单箭头，然后选择另存为模板。
2	如果尚未选择项目，则浏览到 SoMachine Basic 安装文件夹中的示例文件夹。
3	输入项目的名称。
4	选择示例项目文件 (*.smbe) 作为文件类型，然后单击保存。

关闭 SoMachine Basic

概述

要退出 SoMachine Basic，请单击 SoMachine Basic 窗口右上角的**关闭**按钮。
也可以单击**起始页**窗口中的**退出**按钮。

附录



附录 A

SoMachine Basic 键盘快捷方式

SoMachine Basic 键盘快捷方式

键盘快捷方式列表

修饰符	键	命令	查看	条件
CTRL	C	复制	文本框	-
CTRL	V	粘贴	文本框	-
CTRL	X	剪切	文本框	-
CTRL	Q	退出 SoMachine Basic	全部	-
CTRL	S	保存项目	全部	-
CTRL	N	新建项目	全部	-
CTRL	O	打开项目	全部	-
CTRL	Z	撤消	TBD	-
CTRL	Y	重做	TBD	-
CTRL	G	连接	全部	-
CTRL	H	断开连接	全部	-
CTRL	J	下载	试运行	-
CTRL	K	上载	试运行	-
CTRL	L	停止	全部	-
CTRL	M	运行	全部	-
CTRL	W	停止模拟器	全部	-
CTRL	B	运行模拟器	全部	-
CTRL	F	搜索	程序	-
ALT	剩余	转至上一个选项卡	全部	-
ALT	Right	转至下一个选项卡	全部	-
ALT	F4	退出 SoMachine Basic	全部	-
	ALT	显示梯形图快捷方式	程序	-
	F1	显示帮助	全部	-
CTRL	I	在所选梯级之前插入新的梯级	程序	-
CTRL	箭头键	绘制线条	梯形图梯级	已选择绘制工具
CTRL	箭头键	擦除线条	梯形图梯级	已选择擦除工具

修饰符	键	命令	查看	条件
CTRL	箭头键	选择 / 取消选择下一个梯形图单元格 (逐个选择单元格)	梯形图梯级	已选择选择工具
SHIFT	箭头键	选择 / 取消选择下一个梯形图单元格 (按区域选择)	梯形图梯级	已选择选择工具
	Suppr	上一个选项卡	程序	已选择项目
	ESC	将指针复位到选择工具	梯形图梯级	所选的工具不是绘制线条或擦除线条、没有拖动任何项目, 而且没有打开任何弹出窗口
	ESC	取消未决的线条	梯形图梯级	正在绘制
	ESC	取消擦除线条	梯形图梯级	正在擦除
	ESC	取消移动所选的项目 (恢复初始位置)	梯形图梯级	正在拖动梯形图项目
	ESC	关闭建议的列表	梯形图梯级	打开建议的列表 (就像可用于触点的描述符)
	ESC	关闭梯形图工具栏的菜单项	梯形图梯级	打开梯形图工具栏的菜单 (就像功能块)
	ENTER	开始 / 停止移动梯形图元素	梯形图梯级	至少选择一个单元格
	箭头键	移动浮动单元格	梯形图梯级	已开始移动单元格
	箭头键	更改当前单元格	梯形图梯级	缺省
	F5	断开触点	梯形图梯级	Asian set 1 梯形图工具栏
	F6	断开分支	梯形图梯级	Asian set 1 梯形图工具栏
SHIFT	F5	关闭触点	梯形图梯级	Asian set 1 梯形图工具栏
SHIFT	F6	闭合分支	梯形图梯级	Asian set 1 梯形图工具栏
	F7	线圈	梯形图梯级	Asian set 1 梯形图工具栏
CTRL	F7	取反线圈	梯形图梯级	Asian set 1 梯形图工具栏
CTRL	F5	置位线圈	梯形图梯级	Asian set 1 梯形图工具栏
CTRL	F6	复位线圈	梯形图梯级	Asian set 1 梯形图工具栏
	F8	应用程序指令	梯形图梯级	Asian set 1 梯形图工具栏
	F9	绘制水平线	梯形图梯级	Asian set 1 梯形图工具栏
	F10	绘制垂直线	梯形图梯级	Asian set 1 梯形图工具栏
CTRL	F9	删除水平线	梯形图梯级	Asian set 1 梯形图工具栏
CTRL	F10	删除垂直线	梯形图梯级	Asian set 1 梯形图工具栏
SHIFT	F7	上升脉冲断开触点	梯形图梯级	Asian set 1 梯形图工具栏
SHIFT	F8	下降脉冲断开触点	梯形图梯级	Asian set 1 梯形图工具栏
ALT	F7	上升脉冲断开分支	梯形图梯级	Asian set 1 梯形图工具栏

修饰符	键	命令	查看	条件
ALT	F8	下降脉冲断开分支	梯形图梯级	Asian set 1 梯形图工具栏
CTRL+SHIFT	O	比较功能块	梯形图梯级	Asian set 1 梯形图工具栏
	X	XOR 块	梯形图梯级	Asian set 1 梯形图工具栏
	F	功能块	梯形图梯级	Asian set 1 梯形图工具栏
	A	激活步骤	梯形图梯级	Asian set 1 梯形图工具栏
	D	停用步骤	梯形图梯级	Asian set 1 梯形图工具栏
CTRL+ALT	F10	反向操作结果	梯形图梯级	Asian set 1 梯形图工具栏
	O	其他梯级项目	梯形图梯级	Asian set 1 梯形图工具栏
ALT	F10	自由绘制的线	梯形图梯级	Asian set 1 梯形图工具栏
ALT	F9	删除自由绘制的线	梯形图梯级	Asian set 1 梯形图工具栏
	C	新触点	梯形图梯级	Asian set 2 梯形图工具栏
	/	新关闭触点	梯形图梯级	Asian set 2 梯形图工具栏
	W	新触点 OR	梯形图梯级	Asian set 2 梯形图工具栏
	X	新关闭触点 OR	梯形图梯级	Asian set 2 梯形图工具栏
CTRL+SHIFT	F4	上升沿	梯形图梯级	Asian set 2 梯形图工具栏
CTRL+SHIFT	F5	下降沿	梯形图梯级	Asian set 2 梯形图工具栏
CTRL+SHIFT	O	比较功能块	梯形图梯级	Asian set 2 梯形图工具栏
ALT	X	XOR 块	梯形图梯级	Asian set 2 梯形图工具栏
	F10	新垂直线	梯形图梯级	Asian set 2 梯形图工具栏
ALT	L	新水平线	梯形图梯级	Asian set 2 梯形图工具栏
	O	新线圈	梯形图梯级	Asian set 2 梯形图工具栏
	Q	新关闭线圈	梯形图梯级	Asian set 2 梯形图工具栏
CTRL+SHIFT	F9	置位线圈	梯形图梯级	Asian set 2 梯形图工具栏
CTRL+SHIFT	F9	复位线圈	梯形图梯级	Asian set 2 梯形图工具栏
	A	激活步骤	梯形图梯级	Asian set 2 梯形图工具栏
	D	停用步骤	梯形图梯级	Asian set 2 梯形图工具栏
	I	新指令	梯形图梯级	Asian set 2 梯形图工具栏
	F	新功能块	梯形图梯级	Asian set 2 梯形图工具栏
ALT	O	其他梯级项目	梯形图梯级	Asian set 2 梯形图工具栏
	F2	停用分支模式	梯形图梯级	欧洲或美国梯形图工具栏
SHIFT	F2	激活分支模式	梯形图梯级	欧洲或美国梯形图工具栏
SHIFT	F3	常开触点	梯形图梯级	欧洲梯形图工具栏

修饰符	键	命令	查看	条件
SHIFT	F4	正常关闭触点	梯形图梯级	欧洲梯形图工具栏
CTRL+SHIFT	F4	上升沿	梯形图梯级	欧洲梯形图工具栏
CTRL+SHIFT	F5	下降沿	梯形图梯级	欧洲梯形图工具栏
CTRL+SHIFT	O	比较功能块	梯形图梯级	欧洲梯形图工具栏
	X	XOR 块	梯形图梯级	欧洲梯形图工具栏
SHIFT	F7	赋值	梯形图梯级	欧洲梯形图工具栏
CTRL+SHIFT	F9	取反线圈	梯形图梯级	欧洲梯形图工具栏
	F9	置位线圈	梯形图梯级	欧洲梯形图工具栏
SHIFT	F9	复位线圈	梯形图梯级	欧洲梯形图工具栏
	A	激活步骤	梯形图梯级	欧洲梯形图工具栏
	D	停用步骤	梯形图梯级	欧洲梯形图工具栏
SHIFT	F5	Function block (功能块)	梯形图梯级	欧洲梯形图工具栏
CTRL+SHIFT	F6	操作块	梯形图梯级	欧洲梯形图工具栏
	F3	行	梯形图梯级	欧洲梯形图工具栏
	F3	绘制线路	梯形图梯级	欧洲梯形图工具栏
	F4	擦除线路	梯形图梯级	欧洲梯形图工具栏
	O	其他梯级项目	梯形图梯级	欧洲梯形图工具栏
SHIFT	F2	激活分支模式	梯形图梯级	SoMachine 梯形图工具栏
	F2	停用分支模式	梯形图梯级	SoMachine 梯形图工具栏
	F3	绘制线路	梯形图梯级	SoMachine 梯形图工具栏
SHIFT	F3	擦除线路	梯形图梯级	SoMachine 梯形图工具栏
	F4	正常触点	梯形图梯级	SoMachine 梯形图工具栏
SHIFT	F4	取反触点	梯形图梯级	SoMachine 梯形图工具栏
CTRL	F9	线圈	梯形图梯级	SoMachine 梯形图工具栏
CTRL+SHIFT	F9	负线圈	梯形图梯级	SoMachine 梯形图工具栏
	F9	置位线圈	梯形图梯级	SoMachine 梯形图工具栏
SHIFT	F9	复位线圈	梯形图梯级	SoMachine 梯形图工具栏
CTRL+SHIFT	F4	上升沿	梯形图梯级	SoMachine 梯形图工具栏
CTRL+SHIFT	F5	下降沿	梯形图梯级	SoMachine 梯形图工具栏
CTRL+SHIFT	{6, 7, 8, 9}	操作块	梯形图梯级	SoMachine 梯形图工具栏
CTRL+SHIFT	{O, P, Q, R, S, T}	比较块	梯形图梯级	SoMachine 梯形图工具栏
	X 或 ALT+X	XOR 块	梯形图梯级	SoMachine 梯形图工具栏
	O 或 ALT+O	其他梯级项目	梯形图梯级	SoMachine 梯形图工具栏

修饰符	键	命令	查看	条件
	A 或 ALT+A	激活步骤	梯形图梯级	SoMachine 梯形图工具栏
	D 或 ALT+D	停用步骤	梯形图梯级	SoMachine 梯形图工具栏



%S

根据 IEC 标准，%S 表示系统位。

%SW

根据 IEC 标准，%SW 表示系统字。

主任务

通过其编程软件运行的处理器任务。主任务有两个段：

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

动态数据表

用于显示如输入位和存储器字等对象的实时值的软件表。当将 SoMachine Basic 连接到 Logic Controller 时，可以将动态数据表中某些对象类型的值强制指定为特定值。可以将动态数据表作为 SoMachine Basic 应用程序的一部分保存。

应用程序

包括配置数据、符号和文档的程序。

指令列表语言

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基于文本的指令。每个指令均包括一个行号、一个指令代码和一个操作数（请参阅 IEC 61131-3）。

梯形图语言

控制器程序指令的图形表示方法，其中包括控制器按顺序执行的一系列梯级中的触点、线圈和功能块符号（请参阅 IEC 61131-3）。

看门狗

看门狗是一种特殊的定时器，用于确保程序不会超过为它们所分配的扫描时间。通常将看门狗定时器设置为比扫描时间较高的值，并在每个扫描周期结束后重置为零。如果看门狗定时器达到预设值（例如，因为程序陷入了死循环），则表明出现了故障且程序将会停止。

符号

符号是由字母、数字、字符组成的字符串（最多 32 个），其中第一个字符为字母。它使您可以个性化控制器对象，以增加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。

与立即寻址相比，应优先使用此方法，因为如果程序的配置发生更改，则符号会使其新即时地址关联自动更新。相反，用作操作数或参数的任何立即寻址却不会更新（请参阅 [立即寻址](#)）。

配置

一个系统内硬件组件的布局 and 互连以及硬件和软件参数，可决定系统的运行特性。

Free POU

可编程对象单元 (POU) 通常包含用来对独立的主程序任务进行编程和更新的库功能。从程序调用可用的自由 POU 作为子程序或跳转。例如，*周期性任务*是作为自由 POU 实施的子程序。

GRAFCET

以结构和图形格式表示顺序操作的运行。

这是一种将任何顺序控制系统划分为一系列步骤的分析方法，动作、转换和条件均与这些步骤关联。

POU

(*程序组织单元*) 源代码的变量声明和相应的指令集。POUs 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，POUs 便可相互使用。



Grafcet, 139

- 后处理, 142
- 如何使用指令, 143
- 序列处理, 141
- 指令, 139
- 程序结构, 140
- 预处理, 140

Grafcet 指令

- 图形元素, 120

Logic Controller

- 启动时状态, 配置, 70
- 将应用程序直接下载到, 35
- 支持的类型, 20
- 显示当前工作状态, 154
- 更新 RTC, 155
- 替换配置中的当前, 56
- 最近一次停止的日期和时间, 154

Logic Controller 的启动状态, 70

POU

- 复制, 81
- 粘贴, 81
- 自由, 87

RAM

- 可执行文件包含应用程序, 154

RTC

- 使用系统位管理, 125
- 在控制器中更新, 155
- 显示当前日期和时间, 154

TH0, TH1

- 配置为事件源, 92

XOR

- 图形元素, 119

上载

- 使用密码保护, 51

下载

- 固件更新, 173
- 将应用程序直接下载到控制器, 35
- 将用户应用程序下载到控制器, 172

主任务

- 分配 POU, 76
- 系统位和系统字控制, 84
- 配置, 83

事件

- 自上次冷启动以来, 95
- 触发子程序, 92

事件任务

- 概述, 91
- 管理, 94

事件源, 92

- 分配子程序, 94
- 类型, 92

任务, 72

- 事件, 91
- 周期性, 87

任务和扫描模式, 72

- 优先级, 事件, 91
- 使用密码保护应用程序, 49

例外状态

- 故障预置模式行为, 71
- 停止传感器, 接线, 125

停止状态

- 故障预置模式行为, 71

分支模式

- 图形元素, 117

分配模式, 63

列表指令, 132

列表语言

- 概述, 129

创建

- 自由 POU, 80

删除密码保护, 51

功能块

- 图形元素, 119

功能级别, 70

动态数据表, 98

可转换性

- 简介, 64

周期, 扫描, 84

周期性

- 任务, 87
- 扫描周期, 89
- 扫描模式, 84
- 配置持续时间, 89

周期性事件

- 作为事件源, 92

周期性任务

- 将自由 POU 分配给, 81
- 固件, 将更新下载到控制器, 173

图形元素

- 梯形图, 117
- 在控制器中分配存储器, 63
- 在线模式, 63
 - 动态数据表, 98
 - 显示在状态区域中, 42
 - 更新 RTC, 155
 - 概述, 26
 - 编辑动态数据表中的值, 99

复制和粘贴

- POU, 81

子程序

- 与周期性事件关联, 92
- 主任务, 83
- 以自由 POU 形式实现, 76
- 使用事件触发执行, 92
- 分配给任务, 94
- 分配给周期性任务, 87

存储器分配, 63

定时器, 看门狗, 71

密码

- 从项目中删除, 51
- 保护应用程序, 51
- 应用程序是否受保护, 154
- 打开项目文件所需, 51

对象

- 定义, 60
- 实时更新值, 98

寻址

- 符号, 61

导入

- 符号列表, 108

导出

- 符号列表, 109

属性, 49

工作状态

- Logic Controller, 42
- 控制器, 显示, 154

工具栏按钮, 41

布尔

- 累加器, 131

布尔操作符

- 图形元素, 119

常规设置, 44

应用程序

- 下载到控制器, 172
- 使用密码保护, 49, 51
- 定义, 23
- 是否受密码保护, 154
- 行为, 配置, 70

开发程序, 阶段, 24

开发程序的阶段, 25

开发阶段, 25

强制值

- I/O, 154
- 动态数据表中, 98

扩展模块

- 支持的设备, 20

扫描任务, 配置看门狗, 71

扫描时间

- 显示最小值, 最大值, 当前值, 154
- 最小, 显示在状态区域中, 42

扫描模式, 72, 84

括号

- 修饰符, 136
- 在梯形图中使用, 126
- 在程序中使用, 135
- 嵌套, 136

指令

- 上部分 / 下部分, 125

指令列表

- 使用梯级模板, 110
- 显示符号, 62
- 注释, 130

接线停止传感器, 125

操作区, 116

操作块

- 图形元素, 120
- 将赋值指令插入, 123

操作数, 131

- 支持的设备, 20
- 故障预置
 - 模式, 指定, 71
- 数字量输入
 - 配置为事件源, 92
- 替换
 - 配置中的 Logic Controller, 56
- 梯形图
 - 使用括号, 126
 - 使用梯级模板, 110
 - 图形元素, 117
 - 梯级, 115
 - 注释, 124
 - 简介, 114
 - 编程原理, 116
 - 转换为指令列表, 64
- 梯形图编辑器
 - 定义符号, 62
 - 插入后复位指针, 44
 - 自定义, 44
- 梯级
 - 图形元素, 117
- 梯级模板, 110
- 模块区域, 25
- 模式, 离线 / 在线 / 模拟器, 26
- 模拟器, 157
 - I/O 管理器窗口, 159
 - 修改值, 163
 - 修改数字量输入的值, 163
 - 修改模拟量输入的值, 164
 - 如何使用, 168
 - 时间管理窗口, 161
 - 模拟器窗口, 157
 - 访问模拟器, 157
 - 输出跟踪, 165
- 模拟器模式
 - 概述, 26
- 模板
 - 插入梯级, 110
 - 项目, 34
- 正常扫描模式, 84
- 段
 - 主任务, 83
 - 事件中, 91
- 比较块
 - 图形元素, 118
 - 将 IL 比较表达式插入, 122
- 比较表达式
 - 插入梯形图梯级, 122
- 注释
 - 添加到指令列表, 130
 - 添加到梯形图, 124
- 测试区, 116
- 状态
 - 初始 Logic Controller, 配置, 70
- 状态区域, 42
- 用户界面, 设置语言, 44
- 目录, 55
 - 使用型号替换 Logic Controller, 56
- 看门狗定时器, 配置, 71
- 硬件树, 55
- 离线模式
 - 显示在状态区域中, 42
 - 概述, 26
- 程序
 - 定义, 23, 23
 - 跳转, 125
- 程序, 配置故障预置模式, 71
- 程序开发, 阶段, 24
- 程序组织单元 (POU), 76
- 符号
 - 在属性窗口中定义, 61
 - 在指令列表代码中显示, 62
 - 在梯形图编辑器的图形元素中定义, 62
 - 存储在 Logic Controller 中, 62
 - 寻址, 61
 - 已使用的列表, 108
- 符号列表
 - 导入, 108
 - 导出, 109
 - 显示, 108
- 符号寻址, 61

系统位

- %S0, 125*
- %S11, 84*
- %S19, 84*
- %S31, 95*
- %S38, 95*
- %S39, 95*
- %S9, 125*

系统位 / 系统字

- 控制事件, *95*
- 符号列表中, *108*

系统字

- %SW0, 84*
- %SW27, 84*
- %SW30, 84*
- %SW30...%SW32, 154*
- %SW31, 84*
- %SW32, 84*
- %SW48, 95*
- %SW54...%SW57, 154*

累加器, *131*

线圈

- 图形元素, *119*
- 输出的图形表示形式, *114*

线条

- 图形元素, *118*

结束 / 跳转

- 图形元素, *120*

继电器电路, 表示为梯形图, *114*

编程

- 最佳做法, *125*
- 网格, *116*
- 语言, 支持, *21*

网格线, 梯形图编辑器样式, *44*脉冲宽度 (TON), *84*自定义, 梯形图编辑器, *44*

自由 POU

- 分配给事件, *81*
- 分配给事件源, *94*
- 分配给周期性任务, *81*
- 创建, *80*
- 周期性任务, *87*
- 简介, *76*

触点

- 图形元素, *118*
- 输入的图形表示形式, *114*

设置

- 常规, *44*

试运行, *25*语言, 用户界面, *44*调制解调器, 显示状态, *154*

赋值指令

- 插入梯形图梯级, *123*

起始页, *25*

输入

- 修改, *125*
- 配置为事件源, *92*

输出

- 修改, *125*

运算

- 插入梯形图梯级, *123*

选择

- 图形元素, *118*

配置

- 主任务, *83*
- 周期性任务持续时间, *89*
- 应用程序行为, *70*
- 当前, *55*
- 替换 **Logic Controller**, *56*
- 项目属性, *49*

键盘快捷方式, *44*

阈值输出 (%HSC)

- 配置为事件源, *92*

项目

- 使用密码保护, *51*
- 模板, *34*
- 配置属性, *49*