Modicon LMC078

Motion Controller

编程指南

03/2018

EIO0000001914.04

www.schneider-electric.com

本文档中提供的信息包含有关此处所涉及产品之性能的一般说明和/或技术特性。本文档并非用于(也不代替) 确定这些产品对于特定用户应用场合的适用性或可靠性。任何此类用户或设备集成商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。

本手册可用于法律所界定的个人以及非商业用途。 在未获得施耐德电气书面授权的情况下，不得翻印传播本手册全部或部分相关内容、亦不可建立任何有关本手册或其内容的超文本链接。施耐德电气不对个人和非商业机构进行非独占许可以外的授权或许可。 请遵照本手册或其内容原义并自负风险。与此有关的所有其他权利均由施耐德电气保留。

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

当设备用于具有技术安全要求的应用场合时，必须遵守有关的使用说明。

未能使用施耐德电气软件或认可的软件配合我们的硬件，则可能导致人身伤害、设备损坏或不正确的运行结果。

不遵守此信息可能导致人身伤害或设备损坏。

© 2018 Schneider Electric。 保留所有权利。

目录

	安全信息	7
	关于本书	9
第1章	**关于 Modicon LMC078 Motion Controller**	13
	关于 Modicon LMC078 Motion Controller	14
	分布式 I/O 架构	16
第2章	**如何配置控制器**	17
	如何配置控制器	17
第3章	**库**	21
	库	21
第4章	**支持的标准数据类型**	23
	支持的标准数据类型	24
	参数类型	25
第5章	**存储器映射**	27
	控制器存储器结构	28
	RAM 存储器结构	29
	闪存结构	31
	USB 存储盘	32
第6章	**任务**	33
	最大任务数	34
	任务配置屏幕	35
	任务类型	37
	运动任务	40
	系统和任务看门狗	43
	任务优先级	44
	缺省任务配置	47
第7章	**控制器状态和行为**	49
7.1	控制器状态图	50
	控制器状态图	50
7.2	控制器状态介绍	54
	控制器状态介绍	54
7.3	状态转换和系统事件	57
	控制器状态和输出行为	58
	通过命令进行状态转换	61
	错误检测、类型和管理	65
	剩余变量	66

第8章 控制器设备编辑器 67
控制器参数 68
配置参数 70
控制器选择 77
PLC 设置 79
第9章 内嵌输入和输出配置 81
嵌入式 I/O 配置 82
主编码器输入配置 89
第10章 通讯模块 95
10.1 PROFIBUS DP 从站模块配置 96
添加 PROFIBUS DP 从站模块 97
PROFIBUS DP 从站模块配置 99
非循环数据交换 103
10.2 EtherNet/IP 适配器配置 105
EtherNet/IP 适配器配置 106
循环数据交换 109
非循环数据交换 110
10.3 EtherNet/IP 扫描器配置 114
简介 115
支持的设备 116
EtherNet/IP Scanner 配置 118
EtherNet/IP Scanner I/O 映射 120
EtherNet/IP Scanner 状态和诊断 121
目标设备声明 123
目标设置 125
连接配置 127
使用用户参数更换设备 141
EtherNet/IP I/O 映射 144
第11章 以太网配置 147
11.1 以太网服务 148
简介 149
IP 地址配置 151
Modbus TCP 服务器/客户端 156
FTP 服务器 158
FTP 客户端 160
LMC078 Motion Controller 用作 Modbus TCP 上的 IOScanner 从站设备 161

11.2 防火墙配置 166
简介 167
防火墙行为 169
防火墙脚本命令 170
脚本文件 173
第12章 CANopen 配置 175
CANopen 接口配置 176
CANopen 主站配置 177
CANopen 从站配置 179
第13章 Sercos配置 181
Sercos 标准概述 182
Sercos 接口配置 184
Sercos设备 187
设备寻址编辑器 188
Lexium LXM32S 驱动器配置 191
TM5NS31 Sercos Interface Module 194
Sercos 错误代码 195
第14章 串行线路配置 199
串行线路配置 200
ASCII 管理器 202
SoMachine 网络管理器 204
Modbus Serial IOScanner 205
在 Modbus 串行 IOScanner 上添加设备 207
Modbus 管理器 214
将调制解调器添加到管理器 218
第15章 将 Modicon LMC078 Motion Controller 连接到 PC 219
将控制器连接到 PC 219
第16章 固件更新 223
更新 Modicon LMC078 Motion Controller 固件 223
附录 225
附录 A 如何更改控制器的 IP 地址 227
changeIPAddress：更改控制器 IP 地址 227
附录 B 诊断消息 231
消息记录器 232
诊断消息 237

附录 C LMC078 Sercos3 库 247
C.1 数据类型 248
ST_SercosConfiguration 数据类型 249
ST_SercosConfigurationDevice 数据类型 250
ET_Sercos3CmdType 数据类型 252
ET_Sercos3IDNType 数据类型 253
C.2 Sercos功能 254
FC_SercosGetConfiguration 功能 255
FC_SercosReadServiceData 功能 256
FC_SercosReadServiceDataByTopAddr 功能 258
FC_SercosScanConfiguration 功能 260
FC_SercosWriteServiceData 功能 261
FC_SercosWriteServiceDataByTopAddr 功能 263
C.3 异步 Sercos 功能块 265
FB_SercosReadServiceDataAsync：通过 Sercos 接口异步读取数据 . 266
FB_SercosWriteServiceDataAsync：通过 Sercos 接口异步写入数据 . 268
FB_SercosProcedureCommandAsync：通过 Sercos 接口异步发送命令 270
附录 D 用于在用户程序中获取/设置串行线路配置的功能 273
GetSerialConf：获取串行线路配置 274
SetSerialConf：更改串行线路配置 275
SERIAL_CONF：串行线路配置数据类型的结构 277
附录 E 控制器性能 279
处理性能 279
术语表 281
索引 289

安全信息

重要信息

声明

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

在“危险”或“警告”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

危险表示若不加以避免，将**会导致**严重人身伤害甚至死亡的危险情况。

警告

警告表示若不加以避免，可能**会导致**严重人身伤害甚至死亡的危险情况。

小心

小心表示若不加以避免，可能**会导致**轻微或中度人身伤害的危险情况。

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有资质的人员执行。施耐德电气不承担由于使用本资料所引起的任何后果。

有资质的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

本文档的目的是帮助您对 Modicon LMC078 Motion Controller 进行编程和操作。

注意： 在安装、操作、维护或彻底停用 Modicon LMC078 Motion Controller 之前，请阅读并了解本文档和所有相关文档 (参见第 *9* 页)。

Modicon LMC078 Motion Controller 用户应当阅读整个文档，以了解所有功能。

注意： 在本文档中，*实时*是指能够以与接收数据相同的速度处理更新信息。

有效性说明

本文档已随 SoMachine V4.3 TM3TI4D 附加程序的发布进行了更新。

相关的文件

文件名称	参考编号
SoMachine 编程指南	EIO0000000067 (ENG) EIO0000000069 (FRE) EIO0000000068 (GER) EIO0000000071 (SPA) EIO0000000070 (ITA) EIO0000000072 (CHS)
Modicon LMC078 Motion Controller 硬件指南	EIO0000001925 (ENG) EIO0000001926 (FRE) EIO0000001927 (GER) EIO0000001928 (SPA) EIO0000001929 (ITA) EIO0000001930 (CHS) EIO0000001932 (TUR)
Modicon LMC078 Motion Controller 系统功能和变量 PLCSystem 库指南	EIO0000001917 (ENG) EIO0000001918 (FRE) EIO0000001919 (GER) EIO0000001920 (SPA) EIO0000001921 (ITA) EIO0000001922 (CHS) EIO0000001924 (TUR)

文件名称	参考编号
SoMachine - 运动控制库指南	EIO0000002221 (ENG) EIO0000002222 (GER) EIO0000002223 (CHS)
Modicon TM5/TM7 灵活的系统 - 系统计划和安装指南	EIO0000000426 (ENG) EIO0000000427 (FRE) EIO0000000428 (GER) EIO0000000429 (SPA) EIO0000000430 (ITA) EIO0000000431 (CHS)
Modicon TM5 扩展模块配置编程指南	EIO0000000420 (ENG) EIO0000000421 (FRE) EIO0000000422 (GER) EIO0000000423 (SPA) EIO0000000424 (ITA) EIO0000000425 (CHS)
Modicon TM7 扩展块配置编程指南	EIO0000000880 (ENG) EIO0000000881 (FRE) EIO0000000882 (GER) EIO0000000883 (SPA) EIO0000000884 (ITA) EIO0000000885 (CHS)
SoMachine Modbus 和 ASCII 读/写功能 PLCCommunication 库指南	EIO0000000361 (ENG) EIO0000000362 (FRE) EIO0000000363 (GER) EIO0000000364 (SPA) EIO0000000365 (ITA) EIO0000000366 (CHS)
SoMachine 调制解调器功能调制解调器库指南	EIO0000000552 (ENG) EIO0000000491 (FRE) EIO0000000492 (GER) EIO0000000493 (SPA) EIO0000000494 (ITA) EIO0000000495 (CHS)
SoMachine 控制器助手用户指南	EIO0000001671 (ENG) EIO0000001672 (FRE) EIO0000001673 (GER) EIO0000001675 (SPA) EIO0000001674 (ITA) EIO0000001676 (CHS)

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：http://www.schneider-electric.com/en/download

关于产品的资讯

⚠ 警告

失去控制

- 任何控制方案的设计者都必须考虑到控制路径可能出现故障的情况，并为某些关键控制功能提供一种方法，使其在出现路径故障时以及出现路径故障后恢复至安全状态。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。

摘自标准的术语

本手册中的或者出现在产品自身中/上的技术术语、术语、符号和相应描述基本上均源自国际标准的条款或定义。

在功能安全系统、驱动器和一般自动化领域，这可能包括但不限于*安全*、*安全功能*、*安全状态*、*故障*、*故障复位*、*失灵*、*失效*、*错误*、*错误消息*、*危险*等词语。

这些标准包括：

标准	描述
EN 61131-2:2007	编程控制器，第 2 部分：设备要求和测试。
ISO 13849-1:2008	机器安全：控制系统的安全相关部分。 设计通则。
EN 61496-1:2013	机械安全：电子感应式防护设备。 第 1 部分：一般要求和测试。
ISO 12100:2010	机械安全 - 设计的一般原则 - 风险评估和风险抑制
EN 60204-1:2006	机械安全 - 电气机械设备 - 第 1 部分：一般要求

标准	描述
EN 1088:2008 ISO 14119:2013	机械安全 - 与防护设备关联的联锁设备 - 设计和选择原则
ISO 13850:2006	机械安全 - 紧急停止 - 设计原则
EN/IEC 62061:2005	机械安全 - 安全相关的电气、电子和可编程电子控制系统的功能性安全
IEC 61508-1:2010	电气/电子/可编程电子安全相关系统的功能性安全：一般要求。
IEC 61508-2:2010	电气/电子/可编程电子安全相关系统的功能性安全：电气/电子/可编程电子安全相关系统的要求。
IEC 61508-3:2010	电气/电子/可编程电子安全相关系统的功能性安全：软件要求。
IEC 61784-3:2008	用于测量和控制的数字数据通讯：功能性安全现场总线。
2006/42/EC	机械指令
2014/30/EU	电磁兼容性规程
2014/35/EU	低电压规程

此外，本文中所用的名词可能是被无意中使用，因为它们是从其他标准中衍生出来的，如：

标准	描述
IEC 60034 系列	旋转电机
IEC 61800 系列	可调速电力驱动系统
IEC 61158 系列	用于测量和控制的数字数据通讯：用于工业控制系统的现场总线

最后，*操作区*一词可结合特定危险的描述一起使用，其定义相当于 *机器指令 (*) *和 :2010* 中的 *2006/42/EC*风险区*ISO 12100*或*危险区*。

注意： 对于当前文档中引用的特定产品，上述标准可能适用，也可能不适用。若要了解与适用于此处所述产品的各项标准有关的更多信息，请参阅这些产品参考的特性表。

第1章
关于 Modicon LMC078 Motion Controller

简介

本章提供 Modicon LMC078 Motion Controller 和可以使用 SoMachine 配置和编程的设备的相关信息。

本章包含了哪些内容？

本章包含了以下主题：

主题	页
关于 Modicon LMC078 Motion Controller	14
分布式 I/O 架构	16

关于 Modicon LMC078 Motion Controller

概述

Schneider Electric Modicon LMC078 Motion Controller (LMC078CECS20T) 是一款具有多种强大功能的控制器。它可以控制众多应用程序。

Modicon LMC078 Motion Controller 可集中实现控制器和运动功能。Modicon LMC078 Motion Controller 可为最多 24 个轴的机器同步、协调和创建运动功能（最短可在 4 毫秒内同步）。

此控制器因使用 SoMachine 软件平台而成为优化的轴定位解决方案，该平台包含嵌入式自动化功能和人性化的轴配置界面。与 Lexium 32S 伺服驱动器结合使用，您可以轻松设计和试运行应用程序。

有关 Lexium 32S 伺服驱动器的详细信息，请参阅《LXM32S 产品手册》。

SoMachine 编程指南和 LMC078 Motion Controller 编程指南 (参见第 *9* 页)中介绍了软件配置。

主要功能

SoMachine 软件支持以下 IEC61131-3 编程语言用于这些控制器：

- IL：指令列表
- LD：梯形图
- ST：结构化文本
- FBD：功能块图
- SFC：顺序功能图

SoMachine 软件也可用于使用 CFC（连续功能图）语言对这些控制器进行编程。

LMC078 Motion Controller 支持以下现场总线：

- 对于嵌入式通讯接口：
  - CANopen 主站/从站
  - Sercos III
  - Ethernet TCP/IP
  - 串行线路
- 对于可选通讯模块：
  - EtherNet/IP 适配器/扫描器
  - PROFIBUS DP 从站

LMC078 Motion Controller 支持以下 I/O 类型：

- 主编码器输入
- 嵌入式 I/O
  - 数字量 I/O
  - 高级数字量输入（接触式探测器和中断输入）
- CANopen 和 Sercos 现场总线的分布式 I/O（TM5/TM7 模块）

性能

LMC078 Motion Controller 具有以下性能：

- 多达 8 个轴的同步最短可在 1 毫秒内完成
- 多达 16 个轴的同步最短可在 2 毫秒内完成
- 多达 24 个轴的同步最短可在 4 毫秒内完成（对于硬件版本不低于 RS02 的产品）。
- 最短任务循环时间（不适用于运动）：250 微秒

如要显示硬件版本，可执行以下任一一个操作：

1. 显示控制器的配置参数 (参见第 *70* 页)。
2. 检查 HW_Code 参数的前 2 个字符是否分别为“0”和“2”。

或：

1. 查看控制器的 LC Display。
2. 使用菜单按钮显示 HwCode 菜单项。
3. 检查 HwCode 参数的前 2 个字符是否分别为“0”和“2”。

硬件版本RS02 的 HW_Code 或 HwCode 参数示例：

<u>02</u>24013000000000

分布式 I/O 架构

简介

LMC078 Motion Controller 通过以下组件提供创建分布式 I/O 岛的可能性：

- Sercos 现场总线，带 TM5 现场总线接口 (TM5NS31)
- CANopen 现场总线，带 TM5 现场总线接口 (TM5NC31) 或 TM7 现场总线接口 (TM7NCOM•••)

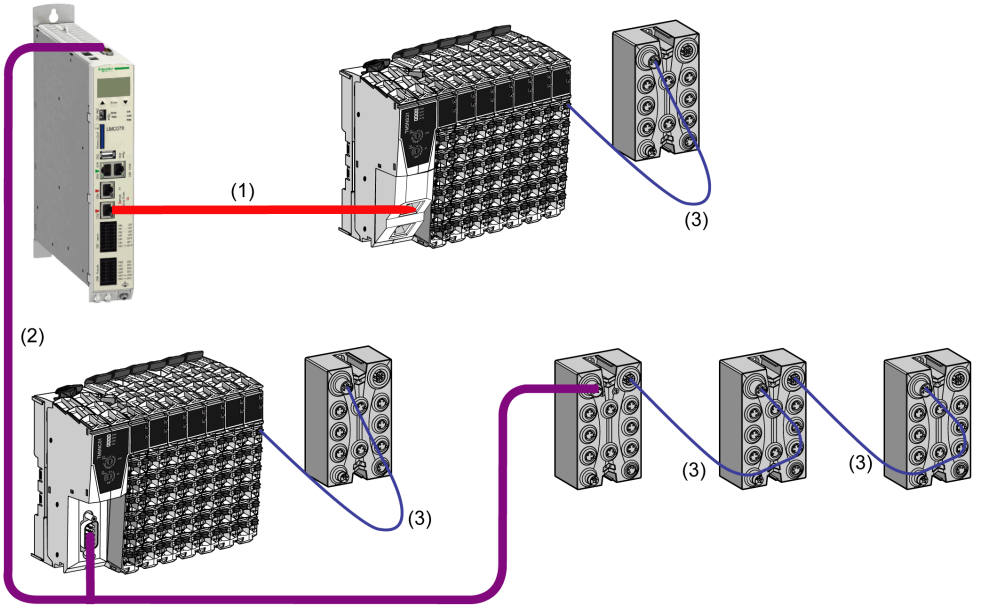
LMC078 Motion Controller 分布式架构

可通过以下各项的关联获得优化远程配置和灵活性：

- LMC078 Motion Controller
- TM5 和/或 TM7 现场总线接口
- TM5 和/或 TM7 扩展模块

由应用程序要求确定 LMC078 Motion Controller 配置的架构。

下图显示了 Sercos 和 CANopen 现场总线的分布式配置：



(1) Sercos 现场总线
(2) CANopen 现场总线
(3) TM5/TM7 扩展总线

有关 TM5 和 TM7 扩展总线的详细信息，请参阅《TM5/TM7 分布式 I/O 架构 *(参见 Modicon TM5 / TM7 Flexible System, System Planning and Installation Guide)*》。

第2章
如何配置控制器

如何配置控制器

简介

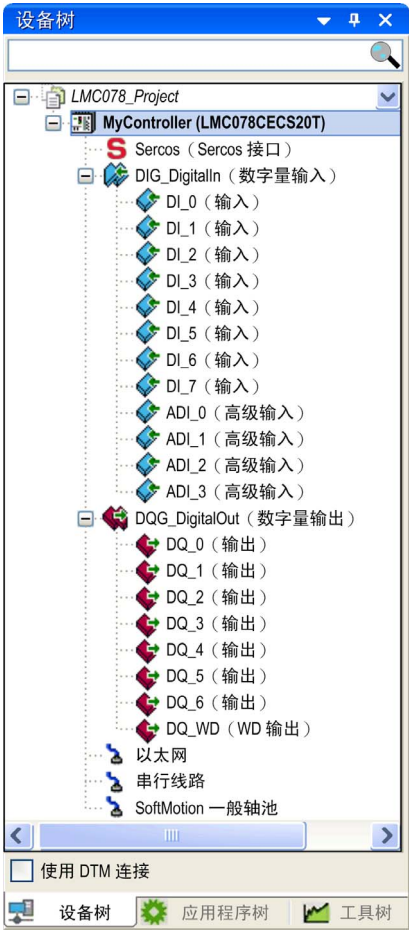
首先，在 SoMachine 软件中创建一个新项目或打开现有项目。

有关如何执行以下操作的信息，请参阅《*SoMachine 编程指南*》：

- 将控制器添加到项目
- 更换现有控制器
- 将控制器转换为不同的兼容设备

设备树

设备树显示了当前硬件配置的结构化视图。当您将控制器添加到项目时，会将许多节点添加到**设备树**，具体取决于控制器提供的功能。



项	用于配置...
Sercos	已嵌入 Sercos III 接口。
DIG_DigitalIn	已嵌入 Motion Controller 的数字量输入。
DQG_DigitalOut	已嵌入 Motion Controller 的数字量输出。
以太网	已嵌入以太网和串行线路通讯接口。
串行线路	
SoftMotion 一般轴池	SoftMotion 设备（虚拟轴配置）。

应用程序树

应用程序树可用于管理项目特定的应用程序以及全局应用程序、POU 和任务。

工具树

工具树可让您：

- 配置项目的 HMI 部分。
- 管理库。
- 访问**设备寻址**工具 (参见第 *188* 页)。
- 访问**消息记录器**工具 (参见第 *232* 页)。
- 添加 CNC 程序。

第3章
库

库

简介

库提供可用于开发您的项目的功能、功能块、数据类型和全局变量。

SoMachine 的**库管理器**提供项目所涉及的库的相关信息，并可帮助您安装新库。有关**库管理器**的详细信息，请参阅 SoMachine - 编程指南。

Modicon LMC078 Motion Controller

为应用程序选择 Modicon LMC078 Motion Controller 后，SoMachine 会加载以下库：

库名称	描述
SystemConfiguration	SoMachine 仅使用此库的内容创建驱动器功能块实例。
LMC078 PLCSystem *(参见 Modicon LMC078 Motion Controller, 系统功能和变量, PLCSystem 库指南)*	包含功能和变量，用于获取信息和向控制器系统发送命令。
SystemConfigurationItf	包含用于管理系统对象（控制器、驱动器和电源）的不同属性的接口。
IoStandard	**CmpIoMgr** 配置类型、**ConfigAccess**、参数和帮助功能：管理应用程序中的 I/O。
Standard	包含为符合 IEC61131-3 所需的功能和功能块，作为 IEC 编程系统的标准 POU。将标准 POU 链接到项目 (standard.library)。
SM3_Basic	包含 SoftMotion 基本管理的功能，有关详细信息，请参阅 SoMachine 在线帮助中的 *CoDeSys 库/SoftMotion 库*一章。
SM3_CNC	包含 SoftMotion CNC 管理的功能，有关详细信息，请参阅 SoMachine 在线帮助中的 *CoDeSys 库/SoftMotion 库*一章。
LMC078 Sercos3 (参见第 *181* 页)	包含通过 Sercos 接口读取/写入数据和发送命令的功能和变量。
Util	包含用于模拟量监视、BCD 转换、位/字节功能、控制器数据类型、功能操控器、数学符号及信号的功能。
CAA Device Diagnosis	该库可提供用于实施简单但高效的诊断功能的功能和接口。 此外，该库还可定义用于访问每个设备和每个现场总线的所需信息的方法。
CDS_MemMan	存储器管理器库

第4章
支持的标准数据类型

简介

本章提供控制器支持的各种 IEC 数据类型。

本章包含了哪些内容？

本章包含了以下主题：

主题	页
支持的标准数据类型	24
参数类型	25

支持的标准数据类型

支持的标准数据类型

控制器支持以下 IEC 数据类型：

数据类型	下限	上限	信息内容
BOOL	FALSE	TRUE	1 位
BYTE	0	255	8 位
WORD	0	65,535	16 位
DWORD	0	4,294,967,295	32 位
LWORD	0	2^{64}-1	64 位
SINT	-128	127	8 位
USINT	0	255	8 位
INT	-32,768	32,767	16 位
UINT	0	65,535	16 位
DINT	-2,147,483,648	2,147,483,647	32 位
UDINT	0	4,294,967,295	32 位
LINT	-2^{63}	2^{63}-1	64 位
ULINT	0	2^{64}-1	64 位
REAL	1.175494351e-38	3.402823466e+38	32 位
STRING	1 个字符	255 个字符	1 个字符 = 1 个字节
WSTRING	1 个字符	255 个字符	1 个字符 = 1 个字
TIME	-	-	32 位

有关 ARRAY、LTIME、DATE、TIME、DATE_AND_TIME 和 TIME_OF_DAY 的有关详细信息，，请参阅《SoMachine 编程指南》。

参数类型

参数类型

下表介绍了控制器参数类型：

类型[1]	可在线编辑	可离线编辑	字体颜色[2]	属性	读取值	写入值
ER	是	是	黑色	用户初始化输入。 仅在复位后传输。	快速存储器访问。	未使用。
ED	是	是	黑色	用户初始化输入。 在更改后直接传输。	快速存储器访问。	快速存储器访问。
ED	是	否	灰色	自动初始化输入为缺省值。 在更改后直接传输。 只能通过 SoMachine 在线更改。	快速存储器访问。	快速存储器访问。
EF	是	是	黑色	用户初始化输入。 在更改后直接传输。	快速存储器访问。	功能访问（需要进行内部计算）。
EF	是	否	灰色	自动初始化输入。 在更改后直接传输。 只能通过 SoMachine 在线更改。	快速存储器访问。	功能访问（需要进行内部计算）。
ES	是	否	灰色	输入。 在更改后直接传输。 只能通过 SoMachine 在线更改。	快速存储器访问。	通过 Sercos 通讯。 调用方延迟（通常为 10...100 毫秒）。
AK	否	否	灰色	输出。 常量值。	快速存储器访问。	不可行。
AD	否	否	灰色	输出。 动态值。	快速存储器访问。	不可行。
AF	否	否	灰色	输出。 动态值。	功能访问（需要进行内部计算）。	不可行。
AS	否	否	灰色	输出。 动态值。	通过 Sercos 通讯。 调用方延迟（通常为 10...100 毫秒）。	不可行。

(1) 参数类型显示在控制器**配置**屏幕的**描述**列（控制器参数 (参见第 *70* 页)、嵌入式 I/O 参数 (参见第 *82* 页)、编码器参数 (参见第 *89* 页)、Sercos 参数 (参见第 *184* 页)）。

(2) 字体颜色是显示在**配置**屏幕上的参数的颜色。如果以黑色字体显示参数，则表示可离线编辑。

Sercos 复位参数

在输入后不会立即接受 Sercos 复位参数，但只有在下一次 Sercos 运行时才会接受（阶段 0 -> 阶段 4）。

下表列出了控制器的 Sercos 复位参数：

参数	分组	验收	参数类型
WorkingMode	标识	阶段 2 -> 阶段 3	EF
IdentificationMode	标识	阶段 2 -> 阶段 3	EF
ConfiguredTopologyAddress	标识	阶段 2 -> 阶段 3	EF
ConfiguredApplicationType	标识	阶段 2 -> 阶段 3	EF
ConfiguredSercosAddress	标识	阶段 2 -> 阶段 3	EF
ConfiguredSerialNumber	标识	阶段 2 -> 阶段 3	EF

第5章
存储器映射

简介

本章介绍 Modicon LMC078 Motion Controller 中不同存储器区域的存储器映射和大小。这些存储器区域用于存储用户程序逻辑、数据和编程库。

本章包含了哪些内容？

本章包含了以下主题：

主题	页
控制器存储器结构	28
RAM 存储器结构	29
闪存结构	31
USB 存储盘	32

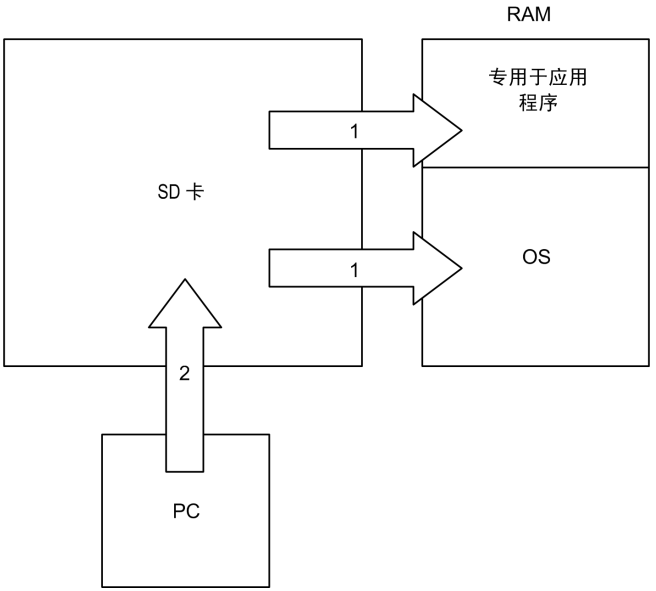
控制器存储器结构

简介

控制器存储器由 3 种类型的物理存储器组成：

- SD 卡 (参见第 *31* 页)存储器包含各种文件（如应用程序、配置文件和 OS）。
- RAM (Random Access Memory) (参见第 *29* 页) 用于执行应用程序和 OS。
- NVRAM 包含保留变量和持久保留变量。

存储器中的文件传输



项	控制器状态	文件传输事件	连接	描述
1	–	在通电和重新启动时自动启动	内部	将 SD 卡中的文件传输到 RAM。 RAM 的内容被覆盖。
2	所有状态	由用户启动	以太网或 USB 编程端口	文件可通过以下途径传输： ● FTP 服务器 (参见第 *158* 页) ● SoMachine

注意： 可以读取、写入和擦除 SD 卡中的文件，具体取决于控制器状态。修改 SD 卡中的文件不会影响正在运行的应用程序。对 SD 卡中的文件的任何更改都将在下次重新启动时生效。

RAM 存储器结构

简介

本节介绍 Modicon LMC078 Motion Controller 不同区域的 RAM (Random Access Memory) 大小。

存储器映射

RAM (512 MB) 包含两个区域：
- 操作系统存储器
- 专用应用程序存储器

NVRAM (128 KB) 包含两个区域：
- 保留变量
- 持久保留变量

NVRAM 包含在断电时或当控制器关闭后由内部电池保留的持久变量和保留变量。

将变量声明为**持久**，会使控制器的循环时间以每 1000 个变量大约 0.2 毫秒的量增加。

持久变量保存在 NVRAM 中，在断电时或当控制器关闭后由内部电池保留。

应为您的应用程序配置必需的最小数量的持久变量，以有助于避免控制器性能降级。

下表介绍专用应用程序存储器：

区域	元素	大小（字节）
系统区域	输入 (%I)	最小：65536
	输出 (%Q)	最小：65536
	存储器 (%M)	最小：65536
用户区域	符号	最小：1048576
	变量	
	应用程序	
	库	

注意： 缺省情况下，在启动阶段过程中分配定义的大小。同样，也可执行动态存储器分配。

下表介绍了 NVRAM 存储器：

区域	大小（字节）
保留变量	1000...84501
保留变量和保留持久性变量	1000...84501

注意： 要验证每个区域的存储器使用情况，右键单击**设备树**中的控制器节点，然后单击**设备存储器信息**。

系统变量

有关系统变量的详细信息，请参阅 *LMC078 PLCSystem 库指南*。

存储器寻址

下表介绍了地址大小 Double word (%MD)、Word (%MW)、Byte (%MB) 和 Bit (%MX) 的存储器寻址：

双字	字	字节	位
%MD0	%MW0	%MB0	%MX0.7...%MX0.0
		%MB1	%MX1.7...%MX1.0
	%MW1	%MB2	%MX2.7...%MX2.0
		%MB3	%MX3.7...%MX3.0
%MD1	%MW2	%MB4	%MX4.7...%MX4.0
		%MB5	%MX5.7...%MX5.0
	%MW3	%MB6	%MX6.7...%MX6.0
		%MB7	%MX7.7...%MX7.0
%MD2	%MW4	%MB8	%MX8.7...%MX8.0
		...	...
	...	...	...
		...	...

存储器范围重叠的示例：

%MD0 包含 %MB0...%MB3，%MW0 包含 %MB0 和 %MB1，%MW1 包含 %MB2 和 %MB3。

闪存结构

简介

SD 卡包含控制器使用的文件系统。

您也可以将 SD 卡用作大容量存储介质用来存储文件。

文件结构

下表列出了 SD 卡的文件结构：

目录	文件	内容
\	Application.app Application.crc	应用程序
\ESystem\	bootc4.sys	引导加载器
	sysc3.sys	VxWorks 内核和固件
	sysc3.cfg	Lzs2 组件配置
\ESystem\FBUSFW\	NETX100-BSL.bin	NetX 引导加载器
	cifXrcX.nxf	NetX 基本固件
	DPS_XC0.nxo	PROFIBUS DP 从站的 NetX 固件
	DPS_XC2.nxo	
	nx100eis.nxo	EtherNet/IP 适配器的 NetX 固件
	nx100eim.nxo	EtherNet/IP 扫描器的 NetX 固件
	nx100ecs.nxo	EtherCAT 从站的 NetX 固件
\ESystem\FirmwareDatabase\D3\	TM5NS31_V245.fw	TM5NS31 Sercos 接口模块的固件
\ESystem\Languages\	english.xml	LCD 语言
\romfs\	Prsnlty.ini	EtherBrick 配置

注意： 您可以使用 **CAA File** 库的功能访问 SD 卡的文件。有关此库的功能块的详细信息，请参阅 SoMachine 在线帮助中的 *CoDeSys 库*主题。

USB 存储盘

简介

可以将 USB 存储盘用作存储文件的大容量介质。它可以通过 FTP 或应用程序进行访问。

注意： 您可以使用 **CAA File** 库的功能访问 USB 存储盘。有关此库的功能块的详细信息，请参阅 SoMachine 在线帮助中的 *CoDeSys 库*主题。

第6章
任务

简介

应用程序树中的**任务配置**节点用于定义一个或多个任务，以控制应用程序的执行。

可用的任务类型有：

- 循环
- 事件
- 外部事件
- 状态

本章包含了哪些内容？

本章包含了以下主题：

主题	页
最大任务数	34
任务配置屏幕	35
任务类型	37
运动任务	40
系统和任务看门狗	43
任务优先级	44
缺省任务配置	47

最大任务数

最大任务数

Modicon LMC078 Motion Controller 最多可支持 1000 个应用程序任务。

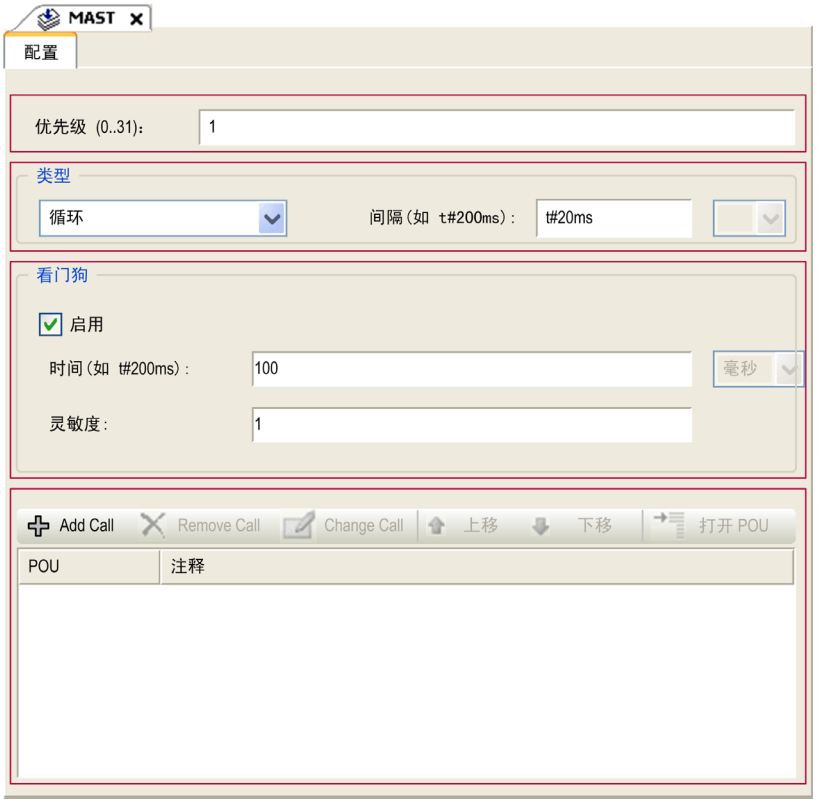
任务配置屏幕

屏幕描述

可通过以下屏幕配置任务。在**应用程序树**中双击要配置的任务可访问此屏幕。

每个配置任务都有自己的参数，这些参数与其他任务无关。

配置窗口由 4 个部分组成：



下表介绍**配置**屏幕的字段：

字段名称	定义
优先级	使用 0 到 31 这些数字配置每个任务的优先级（0 表示最高优先级，31 表示最低优先级）。 一次只能运行一个任务。优先级确定何时运行任务： ● 优先级高的任务先于优先级低的任务执行 ● 具有相同优先级的任务将轮流运行（2 毫秒时间片） **注意：** 请勿分配具有相同优先级的任务。如果还存在其他任务试图先于具有相同优先级的任务执行，则结果可能不确定且不可预知。有关重要的安全信息，请参阅任务优先级 (参见第 *44* 页)。
类型	下列任务类型可用： ● 循环 ● 事件 (参见第 *38* 页) ● 外部 (参见第 *38* 页) ● 状态 (参见第 *38* 页)
看门狗	要配置看门狗 (参见第 *43* 页)，请定义以下 2 个参数： ● **时间**：输入看门狗执行前的超时。 ● **灵敏度**：定义控制器停止程序执行并进入 HALT 状态前的看门狗定时器到期数。
POU	由任务控制的 POU *(参见 SoMachine, 编程指南)*（程序组织单位）列表在任务配置窗口中定义： ● 要添加链接到任务的 POU，请使用命令 **Add Call** 并在**输入助手**编辑器中选择 POU。 ● 要从列表中删除 POU，请使用命令 **Remove Call**。 ● 要将列表的当前所选 POU 替换为其他项，请使用命令 **Change Call**。 ● 按列表中所显示的顺序执行 **POU**。要移动列表中的 **POU**，请选择 **POU** 然后使用命令**上移**或**下移**。 **注意：** 可创建所需数量的 POU。如果应用程序使用多个小型 POU（而不是一个大型 POU），则会延长在线模式下的变量刷新时间。

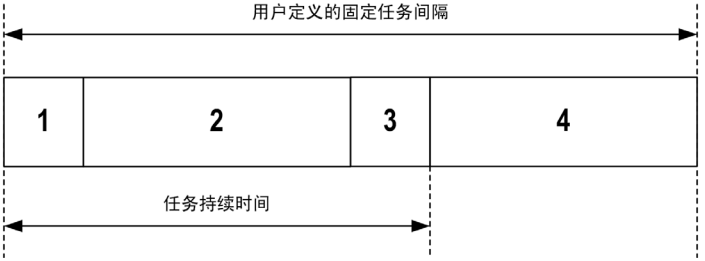
任务类型

简介

以下部分介绍可用于您的程序的各种任务类型，并介绍了任务类型的特性。

循环任务

使用循环任务“配置”子选项卡的“类型”部分中的“间隔”设置向该任务分配固定循环时间。每个循环任务类型的执行方式如下：



1. **读取输入**：将物理输入状态写入 %I 输入存储器变量，并执行其他系统操作。
2. **任务处理**：处理任务中定义的用户代码（POU 等）。%Q 输出存储器变量会根据应用程序指令进行更新。在此操作期间，尚未将分布式 I/O 模块的输出值写入物理输出。嵌入式输出值会立即写入物理输出。
3. **写入分布式 I/O 模块的输出**：%Q 输出存储器变量使用已定义的任何输出强制进行修改；但是，物理输出的写入取决于输出类型和所用指令。
   有关定义总线循环任务的详细信息，请参阅 SoMachine 编程指南和 Modicon LMC078 Motion Controller 设置 (参见第 *79* 页)。
4. **剩余间隔时间**：控制器固件执行系统处理和任何其他较低优先级的任务。

注意： 如果为某个循环任务定义的周期过短，则该任务会重复，而不会执行其他较低优先级的任务或任何系统处理。这将会影响所有任务的执行。

注意： 使用 **GetCurrentTaskCycle** 和 **SetCurrentTaskCycle** 功能通过应用程序获取和设置循环任务的间隔。（有关详细信息，请参阅《Toolbox Advance 库指南》。）

循环任务的最小循环时间为 250 微秒。所配置的循环时间必须为 250 微秒的倍数（例如，500 微秒、750 微秒、1 毫秒等）。

事件任务

此类型的任务由事件驱动，并由程序变量启动。除非有更高优先级的任务先于事件任务执行，否则事件任务在与触发事件关联的布尔变量的上升沿启动。在此情况下，事件任务会根据任务优先级分配的指示启动。

例如，如果您已经定义一个称为 my_Var 的变量并且要将其分配给某个事件，请执行以下步骤：

步骤	操作
1	双击**应用程序树**中的**任务**。
2	从**配置**选项卡的**类型**列表中选择**事件**。
3	单击**事件**字段右侧的**输入助手**按钮 。 **结果**：显示**输入助手**窗口。
4	在**输入助手**对话框的树中导航，查找并分配 my_Var 变量。

注意： 触发事件任务的事件的最大允许频率为 100 Hz。

外部事件任务

此类型的任务由事件驱动，并通过检测到硬件或硬件相关的功能事件而启动。除非有更高优先级的任务先于外部事件任务执行，否则它会在事件发生时启动。在此情况下，外部事件任务会根据任务优先级分配的指示启动。

例如，外部事件任务可能与高级输入上的上升沿关联 (DI8...DI11)。要将 INIRQ1 事件与某个外部事件任务关联，请从**配置**选项卡的**外部事件**下拉列表中选择。

最多有 6 个类型的事件可与外部事件任务关联：

- INIRQx：高级输入的上升沿
- RTP_READ：实时数据读取后的实时进程
- RTP_MENC：主编码器后的实时进程
- RTP_LENC：逻辑编码器后的实时进程
- RTP_AXIS：计算 RefValues 功能块后的实时进程
- MDT_WRITE_ACCESS：用于触发运动任务（对 MDT（Sercos 主站数据报文 (参见第 *182* 页)具有写入权限））

状态任务

此类型的任务由事件驱动，并由程序变量启动。除非有更高优先级的任务先于状态任务执行，否则状态任务在布尔变量与触发事件关联时启动。在此情况下，状态任务会根据任务优先级分配的指示启动。

例如，如果您已经定义一个称为 my_Var 的变量并且要将其分配给某个状态任务，请执行以下步骤：

步骤	操作
1	双击**应用程序树**中的**任务**。
2	从**配置**选项卡的**类型**列表中选择**状态**。
3	单击**事件**字段右侧的**输入助手**按钮 。 **结果**：显示**输入助手**窗口。
4	在**输入助手**对话框的树中导航，查找并分配 my_Var 变量。

运动任务

简介

本节介绍运动任务的特征，并提供有关使用可选已配置运动系统时的可能性能的信息。运动任务通过使用 MDT_WRITE_ACCESS 的**外部**事件名称自动创建。这种机制有利于运动任务与 Sercos 总线的总线循环进行同步。

参数**优先级 (0...31)** 被忽略。任务凭借实时进程的优先级执行（高于 IEC 任务优先级 0）。

SR_Motion POU 自动创建并附加到运动任务。

注意：
适当定义的循环时间应同时满足以下两个要求：

- 运动任务中定义的程序处理必须有足够的时间充分执行。在所有操作条件下测试运动任务的执行时间，以确定此值。
- Sercos **循环时间** (参见第 *184* 页)必须具有足够的持续时间，以便允许在控制器和所有已配置设备之间物理交换所有数据。

如果未配置足够的**循环时间**，则可能会导致系统警戒时钟例外或甚至受控设备同步丢失。例如，不足的**循环时间**可能会导致检测到所有受控设备的 Sercos 主站丢失。在这种情况下，检测到 Sercos 主站丢失的设备将进入其已编程的故障预置状态。在将系统投入使用之前，请务必确认**循环时间**足以允许完全执行运动任务，并完成所有数据的完整物理交换。

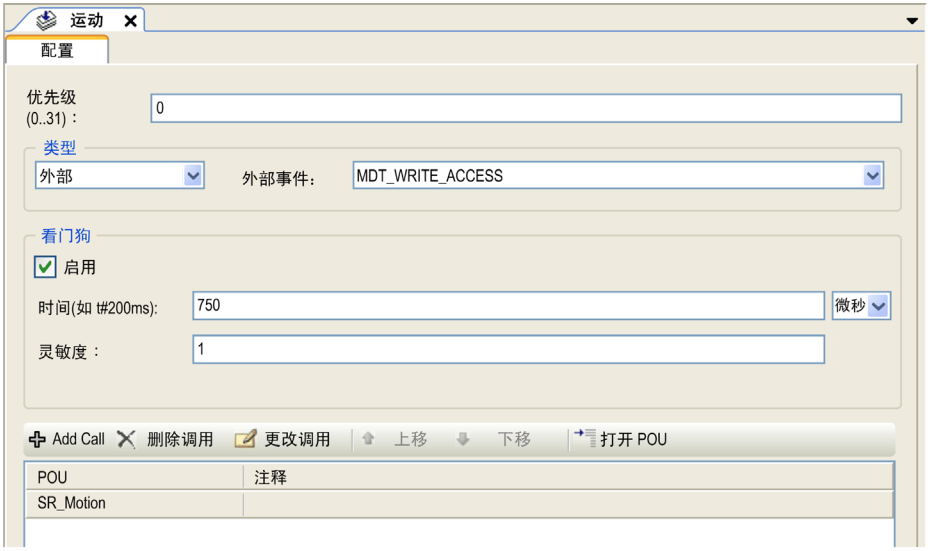
⚠ 警告

意外的设备操作

- 计算任务处理和物理数据交换所需的最短循环时间。
- 为运动任务定义任务（软件）警戒时钟，使其警戒时钟周期稍大于为 Sercos 接口定义的**循环时间**。
- 在将系统投入使用之前，请在正常和异常状态条件下全面测试 Sercos 系统。

不遵循上述说明可能导致人员伤亡或设备损坏。

此图显示运动任务的设置：



注意： 请勿删除运动任务或更改其名称、类型或外部事件属性。否则，在创建应用程序时，SoMachine 便检测不到错误，但一旦您使用应用程序，运动库便会返回错误。

运动任务编程要求

必须使用运动任务来管理与 Sercos 总线及其所连接运动设备（如 Drive Controller）有关的各个编程方面。

这包括：

- 用于获得运动事件的本地输入
- 用于获得运动事件的编码器输入
- 所有运动功能（运动任务 POU 等）的任务处理
- 为响应运动事件而配置的编码器输出
- 为响应运动事件而配置的本地输出

⚠ 警告

意外的设备操作

使用运动任务管理所有与运动有关的输入、输出、任务处理和 Sercos 通讯。

不遵循上述说明可能导致人员伤亡或设备损坏。

运动任务性能

Modicon LMC078 Motion Controller 能够实现高性能。控制器最多可管理：

- 8 个轴，最短同步时间 1 ms
- 16 个轴，最短同步时间 2 ms
- 24 个轴，最短同步时间 4 ms（针对硬件版本高于 RS02 的产品）有关详细信息，请参阅性能（参见第 *15* 页）：

可供使用且仍可让您实现类似性能（通过编写高效的应用程序）的功能子集有：

- 虚拟轴
- 相对和绝对定位
- 速度控制
- 凸轮配置文件
- 电子传动
- 线性和圆形内插通过使用 G 代码

系统和任务看门狗

简介

Modicon LMC078 Motion Controller 实现两种类型的看门狗功能：

- **系统看门狗**：该看门狗在控制器固件中定义并受其管理。它不能由用户进行配置。
- **任务看门狗**：这些看门狗是可为每个任务定义的可选看门狗。这些看门狗由您的应用程序进行管理并可在 SoMachine 中进行配置。

系统看门狗

系统看门狗由控制器固件管理，因此在 SoMachine 在线帮助中有时指的是硬件看门狗。当系统看门狗超过其阈值条件时，会检测到错误并显示在控制器上。

如果在 100 毫秒的间隔中未触发 RTP（实时进程），则会检测到系统看门狗。控制器进入 HALT 状态，需要重新启动控制器以返回 RUNNING 模式。

注意： 用户不能配置系统看门狗。
可将数字量输出 7 (DQ_WD) 配置为由系统看门狗控制的看门狗输出（看门狗输出配置）。

任务看门狗

SoMachine 允许您为应用程序中定义的每个任务配置可选任务看门狗。（在 SoMachine 在线帮助中，任务看门狗有时候也称为软件看门狗或控制定时器）。当您所定义的其中一个任务看门狗达到其阈值条件时，会检测到应用程序错误且控制器进入“暂停”状态（诊断消息 (参见第 *237* 页)）。

在定义任务看门狗时，可使用以下选项：

- **时间**：这定义允许的最长任务执行时间。当任务所用时间超过此值时，控制器会报告任务看门狗异常。
- **灵敏度**：“灵敏度”字段用于定义在控制器检测到应用程序错误之前必须发生的任务看门狗异常次数。

要访问任务看门狗的配置，请双击**应用程序树**中的**任务**。

注意： 有关看门狗的详细信息，请参阅 SoMachine - 编程指南。

任务优先级

任务优先级配置

可以将各个任务的优先级配置为 0 到 31（0 表示最高优先级，31 表示最低优先级）。每个任务的优先级必须唯一。如果向多个任务分配相同的优先级，则这些任务的执行会无法确定且不可预测，这可能会导致意外后果。

⚠ 警告

意外的设备操作

请勿将相同的优先级分配给不同的任务。

不遵循上述说明可能导致人员伤亡或设备损坏。

将 IEC 优先级 (0...31) 映射到系统优先级 (220...251)：

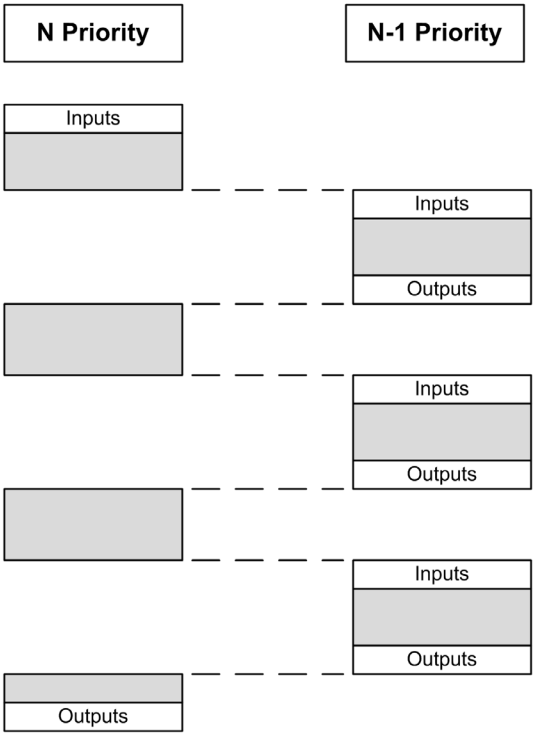
运动任务（附加到事件 MDT_WRITE_ACCESS）是通过实时进程（系统优先级 32）触发的系统优先级 0 创建。

任务优先级建议

- 优先级 0 到 24：控制器任务。将这些优先级分配给具有高可用性要求的任务。
- 优先级 25 到 31：后台任务。将这些优先级分配给具有低可用性要求的任务。

由于任务优先级出现任务抢占

某个任务循环开始后，它可以中断任何优先级较低的任务（任务抢占）。优先级较高的任务循环完成后，被中断的任务将恢复。



注意： 如果在不同任务中使用相同输入，则输入映像可能会在较低优先级任务的任务循环期间发生更改。
为了提高在多任务执行期间正确输出行为的可能性，在不同任务中使用同一字节中的输出时，将显示一条消息。

在将其写入后立即更新嵌入式输出，而不是在循环任务结束时。
在修改其状态后立即更新嵌入式输出，而不是仅在循环任务开始时。

⚠ 警告

意外的设备操作

映射您的输入，从而使任务不会意外更改输入映像。

不遵循上述说明可能导致人员伤亡或设备损坏。

缺省任务配置

缺省任务配置

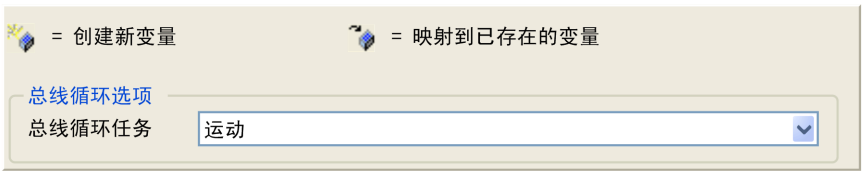
对于 Modicon LMC078 Motion Controller:

- MAST 任务缺省自动创建作为循环任务。其预设优先级为中 (15)，预设时间间隔为 10 毫秒，任务看门狗服务的激活时间为 10 毫秒，灵敏度为 5。有关优先级设置的详细信息，请参阅任务优先级 (参见第 *44* 页)。有关看门狗的详细信息，请参阅系统和任务看门狗 (参见第 *43* 页)。
- 运动任务自动创建。此任务声明为外部事件任务，会将您可为其他操作配置的外部事件任务数减一。运动任务凭借实时进程的优先级执行（优先级参数被忽略）。有关详细信息，请参阅任务优先级 (参见第 *44* 页)。
- SR_Motion POU 自动创建并由运动任务调用。

注意： 请勿删除运动任务或更改其**名称**、**类型**或**外部事件**属性。否则，在创建应用程序时，SoMachine 便检测不到错误，但一旦您使用应用程序，运动库便会返回错误。

缺省**循环时间**为 1 毫秒（Sercos 接口配置 (参见第 *184* 页)）。

必须将运动应用程序控制的所有设备的 **I/O 映射**选项卡中的**总线循环选项**设置为运动任务：



注意： 如果控制器的性能、程序的大小和在程序中执行的功能允许此操作，则您可以考虑放置此任务内的所有代码。

在开发和调试机器过程中，您将需要监控此任务的执行时间。参数 AvailableLoad（控制器对象）、CycleLoad（Sercos 对象）和 RTBWriteRes（Sercos 对象）均可用于估算在实时进程驱动 Sercos 总线时生成代码的负载。如果您不监控执行时间，则可能会导致驱动器和 I/O 输出值的设置点延迟。诊断消息 8507 SERCOS write cycle overflow 表示这种情况。

其他任务

如果需要将代码移动到其他任务，则应创建**循环**类型的其他任务，然后在范围 16 和 31 中选择优先级。

第7章
控制器状态和行为

简介

本章提供与控制器状态、状态转换和对系统事件的响应有关的信息。本章开头详细介绍了控制器状态图并描述了各种状态。接着定义了输出状态与控制器状态的关系，然后解释导致状态转换的命令和事件。最后介绍了与剩余变量有关的信息以及 SoMachine 任务编程选项对系统行为的影响。

本章包含了哪些内容？

本章包含了以下部分：

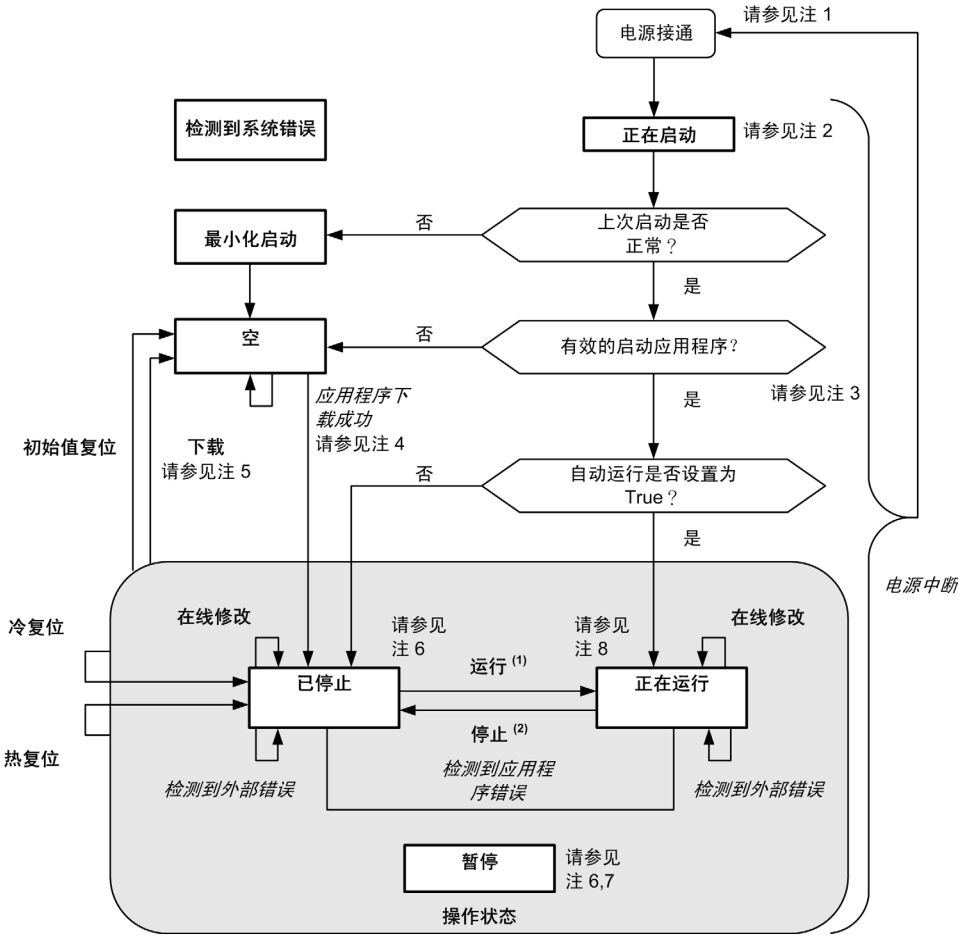
节	主题	页
7.1	控制器状态图	50
7.2	控制器状态介绍	54
7.3	状态转换和系统事件	57

第7.1节
控制器状态图

控制器状态图

控制器状态图

下图描述了控制器的操作模式：



图例：
- 控制器状态以**粗体**表示
- 用户和应用程序命令以**粗体**表示
- 系统事件以*斜体*表示
- 决策、决策结果和一般信息以正常文本表示

(1) 有关“停止”向“运行”状态转换的详细信息，请参阅运行命令 (参见第 *61* 页)。

(2) 有关“运行”向“停止”状态转换的详细信息，请参阅停止命令 (参见第 *61* 页)。

注 1

电源重置（先中断电源，然后接通电源）会删除所有输出强制设置。有关详细信息，请参阅控制器状态和输出行为 (参见第 *58* 页)。

注 2

进入 BOOTING 状态 4 至 5 秒后，LED 才会指示此状态。在正常情况下，启动过程可能最多需要 60 秒。输出会采用其初始化状态。

注 3

应用程序会在验证引导应用程序有效后加载到 RAM 中。

在加载引导应用程序期间，执行检查环境测试以确认剩余变量有效。如果检查环境测试无效，则启动应用程序会加载，但控制器将进入 STOPPED 状态 (参见第 *63* 页)。

注 4

在成功的应用程序下载过程中，会发生以下事件：
- 应用程序直接加载到 RAM 中。
- 缺省情况下，创建引导应用程序，并将其保存到 SD 卡中。

注 5

无论 **AutoRun** 设置或下载前的上一个控制器状态怎样，下载应用程序后的缺省行为是控制器都会进入“已停止”状态。

然而，此时需要考虑两个重要方面：

在线修改： 在控制器处于“运行”状态期间启动的在线修改（部分下载）会在成功后使控制器返回“运行”状态。在使用**登录并看在线变化**选项之前，请在虚拟或非生产环境中测试对应用程序进行的修改，确认控制器和连接的设备是否具备“运行”状态下的预期条件。

> **⚠ 警告**
>
> **意外的设备操作**
>
> 请务必先验证对“运行”状态下的应用程序进行的在线修改是否按预期方式运行，然后再将这些修改下载到控制器。
>
> **不遵循上述说明可能导致人员伤亡或设备损坏。**

注意： 对应用程序的在线修改不会自动写入启动应用，并会在下一次重新启动时由现有启动应用覆盖。如果您希望修改在重新启动之后仍然存在，请通过选择**在线**菜单中的**创建启动应用程序**手动更新启动应用程序（控制器必须处于“已停止”状态才能实现此操作）。

多重下载： SoMachine 具有一项功能，使您可以将完整应用程序下载到网络或现场总线上的多个目标。选择**多重下载...** 命令后的缺省选项之一是**在下载或在线修改后启动全部应用程序**选项，该选项会以“运行”状态重新启动所有下载目标，无论在启动多重下载之前的最后控制器状态为何种状态。如果您不希望所有目标控制器都以“运行”状态重新启动，请取消选择此选项。此外，在使用**多重下载**选项之前，请在虚拟或非生产环境中测试对应用程序进行的修改，确认目标控制器和连接的设备是否具备“运行”状态下的预期条件。

> **⚠ 警告**
>
> **意外的设备操作**
>
> 请务必先验证应用程序是否对于所有目标控制器和设备都按预期方式运行，然后再在选择了“**下装或在线修改之后启动全部应用**”选项的情况下发出“**多重下载...**”命令。
>
> **不遵循上述说明可能导致人员伤亡或设备损坏。**

注意： 与正常下载不同，在多重下载过程中，SoMachine 不提供用于创建启动应用程序的选项。您可以通过选择各个目标控制器上的**在线**菜单中的**创建启动应用程序**随时手动创建启动应用程序（控制器必须处于“已停止”状态才能执行此操作）。

注 6

SoMachine 软件平台提供了多个功能强大的选项，用于在控制器处于“停止”或“暂停”状态期间管理任务执行和输出条件。有关详细信息，请参阅控制器状态描述 (参见第 *54* 页)。

注 7

要退出“暂停”状态，需要发出某个复位命令（热复位、冷复位、初始值复位）、下载应用程序或重置电源。

如果发生不可恢复事件（硬件看门狗或检测到内部错误），则会强制进行电源重置。

注 8

“运行”状态有两种例外情况。

这两种例外情况为：

- 运行但检测到外部错误：此例外情况由 STS 状态 LED 指示，此时显示为绿色常亮且红色闪烁一次。可以通过清除外部错误退出此状态。不需要任何控制器命令。
- 断点运行：此例外情况由 STS 状态 LED 指示，此时显示为绿色闪烁三次。有关详细信息，请参阅控制器状态描述 (参见第 *54* 页)。

第7.2节
控制器状态介绍

控制器状态介绍

简介

本节详细介绍控制器状态。

⚠ 警告

意外的设备操作

- 在没有通过命令进行状态修改、配置控制器选项、上载程序或修改控制器及其所连接设备的物理配置之前，切勿认为控制器已经处于某种控制器状态下了。
- 在执行以上任何操作之前，请考虑对所有所连接设备的影响。
- 在对某个控制器进行操作之前，请始终通过查看其 LED、验证是否存在输出强制并通过 SoMachine 查看控制器状态信息主动确认控制器状态。(1)

不遵循上述说明可能导致人员伤亡或设备损坏。

(1) 可通过使用 LMC078 PLCSystem 库 *(参见 Modicon LMC078 Motion Controller, 系统功能和变量, PLCSystem 库指南)*的 FC_DiagMsgRead 功能或在包含控制器的消息记录器的 SoMachine 中读取控制器状态。

控制器状态表

下表描述了控制器的状态：

控制器状态	描述	STS LED
BOOTING (正在启动)	控制器可执行引导固件及其自身的内部自检。随后它将验证固件和应用程序的校验和。它不执行应用程序，也不进行通讯。	绿色/红色闪烁
在检测到*系统错误*后 BOOTING	此状态与正常的 BOOTING 状态相同，只不过设置了一个标志，使其显示为如同不存在启动应用程序，且 LED 指示不同。	红色快速闪烁
MINIMAL BOOT	SD 卡中不存在有效的固件文件。控制器不执行应用程序。 请参阅升级 Modicon LMC078 Motion Controller 固件 (参见第 *223* 页)。	红色闪烁
EMPTY	不存在应用程序或应用程序无效。	绿色闪烁一次
在检测到*系统错误*后 EMPTY	此状态与正常的 EMPTY 状态相同，只不过设置了一个标志， 使其显示为如同不存在启动应用程序（未加载任何应用程序），且 LED 指示也不同。	红色快速闪烁
RUNNING	控制器正在执行有效应用程序。	绿色

控制器状态	描述	STS LED
断点 RUNNING	此状态与 RUNNING 状态相同，只不过存在以下例外情况： ● 程序的任务处理部分在清除断点之前不会恢复。 ● LED 指示不同。 有关断点管理的详细信息，请参阅 SoMachine 菜单命令在线帮助。	绿色闪烁 3 次
RUNNING 并检测到*外部错误*	此状态与正常的 RUNNING 状态相同，只不过 LED 指示不同。	绿色/闪烁一次红色
STOPPED	控制器中的一个有效应用程序停止。有关此状态下输出和现场总线的行为的说明，请参见STOPPED (已停止) 状态的详细信息 (参见第 *55* 页)。	绿色闪烁
STOPPED 并检测到*外部错误*	此状态与正常的 STOPPED 状态相同，只不过 LED 指示不同。	绿色闪烁/闪烁一次红色
HALT	控制器停止执行应用程序，因为它检测到应用程序错误。 此状态与 STOPPED 状态相同，只不过存在以下例外情况： ● 嵌入式输出采用其初始化值 (参见第 *58* 页)。 ● 当由负责应用程序错误的任务管理时，CAN 总线的行为与未选择**停止时更新 IO** 选项时的情况类似。否则，CAN 总线行为遵循实际设置。 ● LED 指示不同。	红色一次闪烁

STOPPED 状态的详细信息

以下说明适用于 STOPPED 状态：

- 以太网、串行（Modbus、ASCII 等）和 USB 通讯服务保持正常运行，由这些服务写入的命令可以继续影响应用程序、控制器状态和存储器变量。
- 所有输出最初采用其配置的缺省状态（**保持当前值**或**设置所有输出为缺省值**）或输出强制（如果使用）指示的状态。输出的后续状态取决于**停止时更新 IO** 设置以及从远程设备收到的命令。

选择“停止时更新 IO”时的任务和 I/O 行为

如果选择了**停止时更新 IO** 设置：

- ❍ 读取输入操作继续正常执行。会读取物理输入，然后将其写入到 %I 输入存储器变量。
- ❍ 不执行任务处理操作。
- ❍ 写入输出操作继续执行。%Q 输出存储器变量会进行更新以反映**保持当前值**配置或**设置所有输出为默认值**配置，接着针对任何输出强制进行调整，然后写入到物理输出。

  注意： 通过以太网、串行、USB 和 CAN 通讯收到的命令可以继续写入到存储器变量。对 %Q 输出存储器变量进行的修改会写入到物理输出。

选择“停止时更新 IO”时的 CAN 行为

以下情况适用于选中**停止时更新 IO** 设置时的 CAN 总线：

- CAN 总线保持全面正常运行。CAN 总线上的设备可继续识别是否存在正常运行的 CAN 主站。
- TPDO 和 RPDO 继续进行交换。
- 可选的 SDO（如果已配置）继续进行交换。
- 心跳和节点防护功能（如果已配置）继续运行。
- 如果**处于“停止”状态的输出的行为**字段设置为**保持当前值**，则 TPDO 采用上一个实际值继续发出。
- 如果**处于“停止”状态的输出的行为**字段为**将所有输出设置为缺省值**，则上一个实际值会更新为缺省值，后续 TPDO 会采用这些缺省值发出。

未选择“停止时更新 IO”时的任务和 I/O 行为

如果未选择**停止时更新 IO** 设置，控制器会将 I/O 设置为**保持当前值**或**设置所有输出为默认值**条件（根据使用的输出强制进行调整）。在此之后，会出现以下情况：

- 读取输入操作停止。%I 输入存储器变量冻结为其上一个值。
- 不执行任务处理操作。
- 写入输出操作停止。%Q 输出存储器变量可以通过以太网、串行和 USB 连接进行更新。然而，物理输出不受影响，保持配置选项指定的状态。

未选择“停止时更新 IO”时的 CAN 行为

以下情况适用于未选中**停止时更新 IO** 设置时的 CAN 总线：

- CAN 主站停止通讯。CAN 总线上的设备采用其配置的故障预置状态。
- TPDO 和 RPDO 交换停止。
- 可选的 SDO（如果已配置）交换停止。
- 心跳和节点防护功能（如果已配置）停止。
- 在停止 CAN 主站之前，会根据情况将当前或缺省值写入到 TPDO 并发送一次。

第7.3节
状态转换和系统事件

概述

本节开头解释了控制器可能存在的输出状态。然后介绍了用于在控制器状态之间进行转换的系统命令，以及也可以影响这些状态的系统事件。最后解释了剩余变量，以及在状态转换过程中保留不同变量和数据类型的情况。

本节包含了哪些内容？

本节包含了以下主题：

主题	页
控制器状态和输出行为	58
通过命令进行状态转换	61
错误检测、类型和管理	65
剩余变量	66

控制器状态和输出行为

简介

Modicon LMC078 Motion Controller 将输出行为定义为以更加灵活方式响应命令和系统事件。在讨论影响控制器状态的命令和事件之前，有必要先了解此行为。例如，典型控制器仅为停止状态下的输出行为定义 2 个选项：故障预置为缺省值或保持当前值。

可能应用这两个选项的输出行为和控制器状态包括：

- 由**应用程序**管理
- 保持**当前值**
- 将所有**输出设置为缺省值**
- **执行程序**
- 硬件**初始化值**
- 软件**初始化值**
- **输出强制**

由应用程序管理

输出一般情况下由应用程序进行管理。这适用于处于“检出外部错误”状态的 RUNNING 和 RUNNING。

保持当前值

通过在**控制器编辑器**的 **PLC 设置**子选项卡的**处于“停止”状态的输出的行为**下拉菜单中选择**保持当前值**，来选择此选项。要访问控制器编辑器，请在设备树中右键单击控制器，然后选择**编辑对象**。

此输出行为适用于 STOPPED 控制器状态。还适用于处于 HALT 控制器状态的 CAN 总线。输出被设置并保持为其当前状态，尽管输出行为的详细信息会因为**停止时更新 I/O** 选项的设置以及已配置现场总线的命令操作而发生很大变化。有关这些变化的详细信息，请参阅控制器状态描述 (参见第 *54* 页)。

将所有输出设置为缺省值

通过在**控制器编辑器**的 **PLC 设置**子选项卡的**处于“停止”状态的输出的行为**下拉菜单中选择**将所有输出设置为缺省值**，来选择此选项。要访问**控制器编辑器**，请在设备树中右键单击控制器，然后选择**编辑对象**。

当应用程序从 RUN 状态进入 STOPPED 状态时，或者如果应用程序从 RUN 状态进入 HALT 状态，则会采用此输出行为。还适用于处于 HALT 控制器状态的 CAN 总线。输出被设置并保持为其当前状态，尽管输出行为的详细信息会因为**停止时更新 I/O** 选项的设置以及已配置现场总线的命令操作而发生很大变化。有关这些变化的详细信息，请参阅控制器状态描述 (参见第 *54* 页)。

执行程序

您可以通过项目中可用的程序确定输出行为。

通过在**控制器编辑器 PLC 设置**子选项卡的**处于“停止”状态的输出的行为**下拉菜单中选择**执行程序**来选择此选项。

单击按钮 **...** 并通过**输入助手**选择 **POU**。

当控制器处于 STOPPED 状态时执行此程序。

硬件初始化值

此输出状态适应于 BOOTING 和 EMPTY（没有启动应用程序时的电源重置之后或检测到系统错误之后）状态。

在初始化状态时，将输出设置为 0。

软件初始化值

当下载时或复位应用程序时，此输出状态适用。它适用于下载结束或者热复位或冷复位结束时。

软件**初始化值**为输出映像（%I、%Q 或在 %I 或 %Q 上映射的变量）的初始化值。

缺省情况下，它们设置为 0，但是可以映射 GVL 中的 I/O 并向输出分配一个不同于 0 的值。

输出强制

控制器允许您将所选输出的状态强制为定义值，以便于系统测试、试运行和维护。

仅当控制器连接到 SoMachine 时，才能强制输出的值。

为此，请使用**调试**菜单中的**强制值**命令。

输出强制将覆盖对输出执行的所有其他命令，无论正在执行何种任务编程。

如果在定义了输出强制后退出 SoMachine，将向您显示保留输出强制设置的选项。选择此选项后，输出强制会继续控制选定输出的状态，直至您下载应用程序或使用某个复位命令。

如果您的控制器支持，当选中**处于“停止”状态时更新 I/O** 选项时（缺省状态），即使 Logic Controller 处于 STOP 状态，强制的输出也会保持强制值。

输出强制注意事项

您希望强制的输出必须包含在当前正由控制器执行的任务中。未执行任务中或由优先级或事件延迟执行的任务中的强制输出将对该输出不产生影响。但是，一旦延迟的任务得到执行，强制将在此时生效。

根据任务执行情况，强制有可能以您不易察觉的方式影响应用程序。例如，事件任务有可能打开某个输出。之后，您有可能尝试关闭该输出，但此时不触发事件。这样便会明显忽略强制效果。而且，在这之后，该事件可能会在强制生效之时触发该任务。

⚠ 警告

意外的设备操作

- 您必须全面了解强制会对与执行中的任务相关的输出产生怎样的影响。
- 请勿尝试强制包含在您不确定是否会及时执行的任务中的 I/O，除非您打算让强制在下次执行该任务时生效（无论何时）。
- 如果您强制某个输出，但是对物理输出没有产生明显效果，请勿在没有撤销强制的情况下退出 SoMachine。

不遵循上述说明可能导致人员伤亡或设备损坏。

通过命令进行状态转换

运行命令

作用：命令转换到 RUNNING 控制器状态。

启动条件：BOOTING 或 STOPPED 状态。

发出运行命令的方法：

- 在AutoRun配置选项卡中，(参见第 *71* 页) 参数设为 1：在引导后自动启动。
- 在**配置**选项卡 (参见第 *71* 页)中，将 IECProgramStateSet 参数设置为 1。
- SoMachine 在线菜单：选择**启动**命令。
- **登录并看在线变化** 选项：在控制器处于 RUNNING 状态期间启动的在线修改（部分下载）会在成功后使控制器返回 RUNNING 状态。
- **多重下载**命令：如果选择**在下载或在线更改后启动全部应用程序**选项，则将控制器设置为 RUNNING 状态，无论目标控制器最初是处于 RUNNING、STOPPED、HALT 还是 EMPTY 状态。
- 控制器会在某些条件下自动重新启动为 RUNNING 状态。

有关更多详细信息，请参阅控制器状态图 (参见第 *50* 页)。

停止命令

作用：命令转换到 STOPPED 控制器状态。

启动条件：BOOTING、EMPTY 或 RUNNING 状态。

发出停止命令的方法：

- 在**配置**选项卡 (参见第 *71* 页)中，将 IECProgramStateSet 参数设置为 0。
- SoMachine 在线菜单：选择**停止**命令。
- **登录并看在线变化** 选项：在控制器处于 STOPPED 状态期间启动的在线修改（部分下载）会在成功后使控制器返回 STOPPED 状态。
- **下载**命令：将控制器隐式设置为 STOPPED 状态。
- **多重下载**命令：如果未选择**在下载或在线更改后启动全部应用程序**选项，则将控制器设置为 STOPPED 状态，无论目标控制器最初是处于 RUNNING、STOPPED、HALT 还是 EMPTY 状态。
- 控制器会在某些条件下自动重新启动为 STOPPED 状态。

有关更多详细信息，请参阅控制器状态图 (参见第 *50* 页)。

热复位

作用：将所有变量（除了剩余变量）复位为其缺省值。将控制器置于 STOPPED 状态。

启动条件：RUNNING、STOPPED 或 HALT 状态。

发出热复位命令的方法：

- SoMachine 在线菜单：选择**热复位**命令。
- 使用 LMC078 PLCSystem 库 *(参见 Modicon LMC078 Motion Controller, 系统功能和变量, PLCSystem 库指南)*的 `FC_PrgResetAndStart` 功能

热复位命令的效果作用：

1. 应用程序停止。
2. 擦除强制。
3. 复位错误的诊断指示。
4. 保持保留变量的值。
5. 保持保留-持久性变量的值。
6. 所有非定位和非剩余变量都复位为其初始化值。
7. 所有现场总线通讯都停止，然后在完成复位后重新启动。
8. 所有 I/O 都复位为其初始化值。

有关变量的详细信息，请参阅剩余变量 (参见第 *66* 页)。

冷复位

作用：将所有变量（除了保留-持久性类型的剩余变量）都复位为其初始化值。将控制器置于 STOPPED 状态。

启动条件：RUNNING、STOPPED 或 HALT 状态。

发出冷复位命令的方法：

- SoMachine 在线菜单：选择**冷复位**命令。

冷复位命令的作用：

1. 应用程序停止。
2. 擦除强制。
3. 复位错误的诊断指示。
4. 保留变量的值复位为其初始化值。
5. 保持保留-持久性变量的值。
6. 所有非定位和非剩余变量都复位为其初始化值。
7. 所有现场总线通讯都停止，然后在完成复位后重新启动。
8. 所有 I/O 都复位为其初始化值。

有关变量的详细信息，请参阅剩余变量 (参见第 *66* 页)。

初始值复位

作用：将所有变量（包括剩余变量）都复位为其初始化值。擦除控制器上的所有用户文件。将控制器置于 EMPTY 状态。

启动条件：RUNNING、STOPPED 或 HALT 状态。

发出初始值复位命令的方法：

- SoMachine 在线菜单：选择**初始值复位**命令。

初始值复位命令的作用：

1. 应用程序停止。
2. 擦除强制。
3. 擦除所有用户文件（启动应用程序、数据记录、后配置）。
4. 复位错误的诊断指示。
5. 复位保留变量的值。
6. 复位保留-持久性变量的值。
7. 复位所有非定位和非剩余变量。
8. 所有现场总线通讯都停止。
9. 所有 I/O 都复位为其初始化值。

有关变量的详细信息，请参阅剩余变量 (参见第 *66* 页)。

重新启动

作用：命令控制器重新启动。

启动条件：任何状态。

发出重新启动命令的方法：

- 电源重置
- 使用 LMC078 PLCSystem 库 *(参见 Modicon LMC078 Motion Controller, 系统功能和变量, PLCSystem 库指南)*的 FC_SysReset 功能

重新启动的作用：

1. 控制器的状态取决于多种条件：
   a. 在以下条件下，控制器将处于 RUNNING 状态：
      通过电源重置引发重新启动：
      - AutoRun 参数设为 1，控制器在电源重置前未处于 HALT 状态，并且剩余变量有效。
      通过脚本引发重新启动，并且：
      - **启动模式**设置为**在“运行”状态下启动**，已配置“运行/停止”输入或开关并将其设置为 RUN，控制器在电源重置前未处于 HALT 状态，并且剩余变量有效。
   b. 在以下条件下，控制器将处于 STOPPED 状态：
      通过电源重置引发重新启动：
      - AutoRun 参数设为 0。
   c. 在以下条件下，控制器将处于 EMPTY 状态：
      - 不存在启动应用程序或启动应用程序无效，或
      - 由特定系统错误引发重新启动。
   d. 如果不存在有效固件，则控制器状态将为 INVALID_OS。

2. 如果成功加载启动应用程序，则会保持强制。如果未成功加载，则会擦除强制。
3. 复位错误的诊断指示。
4. 如果保存的环境有效，则恢复保留变量的值。
5. 如果保存的环境有效，则恢复保留-持久性变量的值。
6. 所有非定位和非剩余变量都复位为其初始化值。
7. 所有现场总线通讯都会停止，然后在成功加载了启动应用程序之后重新启动。
8. 如果控制器在重新启动之后采用 STOPPED 状态，则所有 I/O 都会复位其初始化值，然后复位为用户配置的缺省值。

有关变量的详细信息，请参阅剩余变量 (参见第 *66* 页)。

注意： 如果应用程序和剩余变量与启动应用程序中的定义相同，则检查环境测试会将环境视作有效。

注意： 如果您在控制器处于 RUNNING 或 STOPPED 状态期间对应用程序进行在线修改，但是未手动更新您的启动应用程序，则控制器会在下一次重新启动时检测到环境中存在差异，剩余变量会按照冷复位命令进行复位，并且控制器会进入 STOPPED 状态。

下载应用程序

作用：将可执行的应用程序加载到 RAM 存储器中。也可以在 SD 卡中创建启动应用程序。

启动条件：RUNNING、STOPPED、HALT 和 EMPTY 状态。

发出下载应用程序命令的方法：

- SoMachine：
  存在 2 个选项用于下载完整应用程序：
  - “下载”命令。
  - “多重下载”命令。

  有关应用程序下载命令的重要信息，请参阅控制器状态图 (参见第 *50* 页)。
- FTP：使用 FTP 将启动应用程序文件加载到 SD 卡。更新的文件会在下一次重新启动时应用。

SoMachine 下载命令的作用：

1. 停止然后现有应用程序，然后将其擦除。
2. 如果有效，则会加载新应用程序，并且控制器采用 STOPPED 状态。
3. 擦除强制。
4. 复位错误的诊断指示。
5. 保留变量的值复位为其初始化值。
6. 保持任何现有保留-持久性变量的值。
7. 所有非定位和非剩余变量都复位为其初始化值。
8. 所有现场总线通讯都会停止，然后在下载完成后启动新应用程序的所有已配置现场总线。
9. 所有 I/O 都复位为其初始化值，然后在下载完成后将其设置为用户配置的新缺省值。

有关变量的详细信息，请参阅剩余变量 (参见第 *66* 页)。

FTP 下载命令的作用：

下一次重新启动后才能看到这些作用的效果。下一次重新启动时，效果与无效环境下的重新启动相同。请参阅重新启动 (参见第 *63* 页)。

错误检测、类型和管理

错误管理

控制器检测和管理三种类型的错误：

- 外部错误
- 应用程序错误
- 系统错误

下表描述了可以检测到的错误类型：

检测到的错误类型	描述	生成的控制器状态
外部错误	外部错误可由处于 RUNNING 或 STOPPED 状态时的系统检测到，但不会影响持续的控制器状态。在以下情况下会检测到外部错误： ● 连接的设备向控制器报告错误。。 ● 控制器检测到外部设备出现错误，例如当外部设备正在通讯但未针对用于控制器而正确配置时。 ● 控制器检测到输出状态存在错误。 ● 控制器检测到与设备的通讯中断。 ● SD 卡中的启动应用程序与 RAM 中应用程序不相同。	RUNNING 并检测到外部错误 或 STOPPED 并检测到外部错误
应用程序错误	遇到错误的编程或超过任务看门狗阈值时，会检测到应用程序错误。	HALT
系统错误	当控制器在运行时期间进入无法管理的条件时，会检测到系统错误。大多数此类状况由固件或硬件例外引起，但有时可能是由于编程不正确而导致检测到系统错误，例如尝试在运行时写入保留的存储器时或发生系统看门狗超时。 **注意：** 一些系统错误可以由运行时管理，因此按照对待应用程序错误的方式进行处理。	BOOTING → EMPTY

注意： 有关诊断的更多详细信息，请参阅《LMC078 PLCSystem 库指南 *(参见 Modicon LMC078 Motion Controller, 系统功能和变量, PLCSystem 库指南)*》。

剩余变量

概述

剩余变量可以在断电、重新启动、复位和应用程序下载时重新初始化或保留其值。剩余变量有多种类型，分别声明为“保留”、“持久”或者组合为“保留持久性”。

注意： 对于此控制器，声明为持久变量的变量行为与声明为保留持久性变量的变量相同。

下表描述了各种情况下剩余变量的行为：

动作	VAR	VAR RETAIN	VAR GLOBAL PERSISTENT RETAIN
对应用程序进行在线修改	X	X	X
在线修改，改动启动应用程序 (1)	–	X	X
停止	X	X	X
电源重置	–	X	X
热复位	–	X(2)	X
冷复位	–	–	X
初始值复位	–	–	–
应用程序下载(3)	–	–	X

X 保持该值。
– 重新初始化该值。
(1) 如果在线修改仅改动启动应用程序的代码部分（例如 a:=a+1; => a:=a+2;），则保持保留变量值。在所有其他情况下，则重新初始化保留变量。
(2) 关于 VAR RETAIN 的详细信息，请参阅热复位命令的作用 (参见第 *62* 页)。
(3) 如果通过 SD 卡下载应用程序，将重新初始化应用程序使用的任何现有持久性变量。但是，如果使用 SoMachine 下载应用程序，现有持久性变量将保持其值。在上述两种情况下，如果下载的应用程序包含与现有应用程序相同的持久性变量，现有保留变量将保持其值。

添加保留持久性变量

在 PersistentVars 窗口中声明保留持久性 (VAR GLOBAL PERSISTENT RETAIN) 符号：

步骤	动作
1	在**应用程序树**中选择**应用程序**节点。
2	单击 。
3	选择**添加其他对象 → 持久性变量**
4	单击**添加**。 **结果**：将显示 PersistentVars 窗口。

第8章
控制器设备编辑器

简介

本章介绍如何配置控制器。

本章包含了哪些内容？

本章包含了以下主题：

主题	页
控制器参数	68
配置参数	70
控制器选择	77
PLC 设置	79

控制器参数

控制器参数

要打开设备编辑器，请双击**设备树**中的 MyController：

选项卡描述

选项卡	描述	限制
配置	访问并配置控制器参数	–
控制器选择 (参见第 *77* 页)	管理 PC 与控制器之间的连接。 ● 帮助您找到网络中的控制器， ● 显示可用控制器列表，以便您可以连接到所选控制器并管理控制器中的应用程序， ● 帮助您从设备编辑器中以物理方式识别控制器， ● 帮助您更改控制器的通讯设置。 基于通讯设置，通过 NetManage 或通过“活动路径”检测控制器列表。要访问**通讯设置**，请单击菜单栏中的**项目 → 项目设置...**。有关详细信息，请参阅 SoMachine 编程指南（*通讯设置*）。	仅在线模式
应用程序	显示正在控制器上运行的应用程序，并可从控制器中删除应用程序。	仅在线模式
文件	PC 与控制器之间的文件管理。	仅在线模式
日志	查看控制器日志文件。	仅在线模式
PLC 设置 (参见第 *79* 页)	以下项目的配置： ● 用于 I/O 处理的应用程序 ● 处于停止模式时的 I/O 行为 ● 总线循环选项	–
任务显示	显示 I/O 及其分配给任务情况的列表。	仅在编译后
用户和组	为支持在线用户管理的设备提供**用户和组**选项卡。它可以设置用户和访问权限组并向他们分配访问权限，以便控制在线模式下对 SoMachine 项目和设备的访问。 有关详细信息，请参阅 SoMachine 编程指南。	–

选项卡	描述	限制
访问权限	为支持在线用户管理的设备提供的**访问权限**选项卡。它用于授予或拒绝当前定义的用户组的特定权限，从而定义运行时用户对于控制器上文件或对象（例如应用程序）的访问权限。 有关详细信息，请参阅 SoMachine 编程指南。	–
信息	显示与设备有关的一般信息（名称、描述、供应商、版本、图像）。	–

配置参数

概述

下图显示了配置选项卡：

我的控制器

配置 | 控制器选择 | PLC 设置 | 文件 | 日志 | 应用程序 | 用户和组 | 访问权限 | 任务显示 | 信息

参数	类型	值	缺省值	单位
常规				
名称	STRING(40)	"	"	
AutoRun	BOOL 枚举	否/0	否/0	
IP_SubNetMask	STRING(15)	'255.255.0.0'	'255.255.0.0'	
IP_Address	STRING(15)	'192.168.100.1'	'192.168.100.1'	
IP_Gateway	STRING(15)	'0.0.0.0'	'0.0.0.0'	
EthernetAdr	STRING(19)	"	"	
MsgFilter	DWORD(0..65535)	16#0000FFFF	16#0000FFFF	
ControllerReset	BOOL 枚举	True/1	True/1	
IOReset	DINT 枚举	下载或编程后复位/2	下载或编程后复位/2	
诊断				
DiagClass	DINT			
DiagCode	DINT			
DiagSource				
DiagMsg	STRING(39)	"	"	
DiagExtMsg	STRING(14)	"	"	
MsgEntries	DINT			
FastTimer	UDINT			微秒
Timer1	DINT			毫秒
Timer10	DINT			10 毫秒
CycleLoad	DINT			%
RTBReadRes	DINT			微秒
RTBWriteRes	DINT			微秒
PowerOK	BOOL 枚举	False/0	False/0	
SetRealTimeClock	DT	0	0	
RealTimeClock	DT	0	0	
BatteryLowWarningDelay	REAL(0..50)	0.0	0.0	h
版本				
FW_Version	STRING(28)	"	"	
ControllerType	STRING(254)	"	"	
ControllerType1	STRING(254)	"	"	
HW_Code	STRING(20)	"	"	
SerialNumber	STRING(20)	"	"	
存储器与磁盘				
RamDiskSize	DINT(128..4096)	1024	1024	KB
RamDiskFree	DINT			KB
Diskfree	DINT			字节
Memoryfree	DINT			字节
系统				
Systemticks	DINT(10..20000)	4000	4000	
EnableLoadEff	BOOL 枚举	关闭/0	关闭/0	
AvailableLoad	DINT	0	0	%
AvailableLoadPeriod	DINT(1..2000)	100	100	毫秒
RemoteCommunicationAccess	DINT 枚举	读取/写入/保存/2	读取/写入/保存/2	
ActivateFatalCrashReaction	BOOL 枚举	是/1	是/1	
IEC 程序				

注意： 有些参数也可在应用程序中访问（如 ObjectName.ParameterName，MyController.AvailableLoad）。

注意： 有些参数也可通过通讯协议进行访问。

参数介绍

下表介绍了诊断和配置的控制器参数：

参数	访问	参数类型	数据类型	值	默认值	描述
常规						
Name	读/写(*)	EF	STRING	“	“	配置对象的符号名称。
AutoRun	读/写	ER	BOOL Enum	no / 0 yes / 1	no / 0	● 0 = Autorun 未激活。 ● 1 = 程序在控制器启动后自动启动。
IP_SubNetMask	R	AF	STRING	255.255.0.0	255.255.0.0	显示 IP 子网掩码。
IP_Address	R	AF	STRING	192.168.100.1	192.168.100.1	显示 IP 地址。
IP_Gateway	R	AF	STRING	0.0.0.0	0.0.0.0	显示网关地址。
EthernetAddr	R	AF	STRING	“	“	显示设备专用以太网地址。
MsgFilter	读/写	EF	DWORD	0...FFFFh	FFFFh	配置消息记录器过滤。 您可以区分 16 种类别的消息： ● 位 0：一般系统消息。 ● 位 1：诊断消息。 ● 位 2：程序系统功能块。 ● 位 3：现场总线特定信息。 ● 位 4...11：未使用。 ● 位 12：扩展系统消息。 ● 位 13-14：未使用。
ControllerReset	读/写(*)	ED	BOOL Enum	FALSE / 0 TRUE / 1	TRUE/1	区分程序复位和控制器复位。 当控制器复位时，会将此参数设置为 1。可使用应用程序将此参数设置为 0。 通过这种方式，您可以区分程序复位和控制器复位。 ● 0 = 程序复位。 ● 1 = 控制器复位。
IOReset	读/写	EF	DINT Enum	0...3	2	定义 I/O 区域的复位模式： ● 0 = 不复位。 ● 1 = 在下载后复位。 ● 2 = 在下载或程序复位后复位。 ● 3 = 在下载、程序复位或程序停止后复位。

参数	访问	参数类型	数据类型	值	默认值	描述
诊断						
DiagClass	R	AD	DINT	-	-	以十进制代码显示诊断类别 (参见第 *237* 页)。
DiagCode	R	AD	DINT	-	-	以十进制代码显示诊断代码 (参见第 *237* 页)。
DiagSource	R	AD	ST_Logical Address	-	-	指定诊断的源 （stLogicalAddress 类型）
DiagMsg	R	AD	STRING	-	-	显示诊断文本。
DiagExtMsg	R	AD	STRING	-	-	显示扩展诊断消息。
MsgEntries	R	AF	DINT	-	-	消息记录器中的条目数。
FastTimer	R	AF	UDINT	-	-	显示定时器（微秒）。定时器派生自 CPU 的 TSC (TimeStampCounter)。TSC 是一个随 CPU 周期运行的 64 位计数器。该计数器从 0 至 2^{31}-1 无休止运行。结束值相当于约 71 分钟。
Timer1	R	AF	DINT	-	-	显示自系统启动后的时间（毫秒）。在达到最大值后（2^{31}-1 毫秒，即约 24.8 天），它从 0 开始向上计数。该计数器在复位后自动启动。Timer1 的精度是 Sercos 驱动器总线的 CycleTime。
Timer10	R	AF	DINT	-	-	显示自系统启动后的时间（步长为 10 毫秒）。
CycleLoad	R	AF	DINT	-	-	表示百分比实时循环的利用率（CycleLoad参数 (参见第 *76* 页)）。
RTBReadRes	R	AD	DINT	-	-	显示实时读取预留（微秒）。
RTBWriteRes	R	AD	DINT	-	-	显示实时写入预留（微秒）。
PowerOK	R	AF	BOOL Enum	FALSE / 0 TRUE / 1	FALSE / 0	表示电源欠压： ● 0 = 电源电压 < 18 V。 ● 1 = 电源电压 > 18 V。
SetRealTimeClock	读/写(*)	EF	DT	-	-	在写入 SetRealTimeClock 参数时设置控制器 RTC。 该参数仅显示设置时钟的时间，或在安装硬件时钟并正常运行后上一次启动的时间。

参数	访问	参数类型	数据类型	值	默认值	描述
RealTimeClock	R	AF	DT	-	-	该参数显示软件时钟的日期和时间，它在控制器通过实时软件时钟启动后自动设置（在控制器关闭后，通过电池供电运行）。
BatteryLow-WarningDelay	读/写	EF	REAL	0...50	0.0	低电量消息延迟（小时）。 如果电池电量低于最小值，则触发诊断消息 8037 Battery low。如果在没有更换电池的情况下确认了诊断消息，则将会在此参数中定义的延迟过后触发诊断消息。当系统复位后，诊断消息的延迟不会持续。
版本						
FW_Version	R	AK	STRING	-	-	显示控制器的固件版本和创建日期。
ControllerType	R	AK	STRING	-	-	显示各种硬件信息： ● 控制器类型：控制器类型的名称。 ● AX：取决于循环时间的最大轴数。 ● RAM：主存储器扩展(MB)。 ● NVRAM：NVRAM 扩展 (KB)。 ● Disks：SD 卡大小 (MB)。
ControllerType1	R	AK	STRING	-	-	显示各种硬件信息： ● PFPGA 版本。 ● CPU 版本。 ● Bios 版本。 ● SFPGA 版本。
HW_Code	R	AK	STRING	-	-	显示控制器的硬件代码。
SerialNumber	R	AK	STRING	-	-	显示控制器的序列号。
存储器与磁盘						
RamDiskSize	读/写	ER	DINT	128...4096	1024	定义 RamDisk 的大小 (KB)。 打开控制器后，会在主存储器中生成带 ram0: 标识符的 RamDisk。读完消息记录器后，系统将 RamDisk 用作数据的临时存储器。
RamDiskFree	R	AF	DINT	-	-	显示 RamDisk 的可用存储器空间。
Diskfree	R	AF	DINT	-	-	显示 SD 卡的可用存储器空间。

参数	访问	参数类型	数据类型	值	默认值	描述
Memoryfree	R	AF	DINT	-	-	显示 RAM 系统存储器的可用存储器空间。
系统						
Systemticks	读/写	ER	DINT	10...20000	4000	此参数可用来设置控制器的系统周期。 **注意：** 如果试图更改此参数，请咨询 Schneider Electric 应用部门。
EnableLoadEff	读/写(*)	EF	BOOL Enum	off / 0 on / 1	off / 0	此参数用于启动和停止有效 CPU 负载测量： ● 0 = 有效 CPU 负载测量已停止。 ● 1 = 有效 CPU 负载测量已启动。
AvailableLoad	R	AD	DINT	0...100	0	显示可用 CPU 时间（百分比）。
AvailableLoadPe-riod	读/写	EF	DINT	1...2000	100	定义有效 CPU 负载测量的测量周期（毫秒）。为此周期计算剩余计算时间。
RemoteCommunica-tionAccess	读/写	ER	DINT Enum	read only / 0 read/write / 1 read/write/ save / 2	read/write/save / 2	您可以使用 NetManage 工具更改控制器的各种通讯参数。 如果不需要，您可以使用参数 RemoteCommunicationAccess 防止它： ● 0 = 值显示在 NetManage 工具中，但控制器不允许对通讯参数进行任何更改。 ● 1 = 控制器临时更改其通讯参数。 ● 2 = 控制器更改其通讯参数，并立即将其保存在 SD 卡上（从而保持此设置，甚至在重新启动时）。保存为可选项，可以在 NetManage 工具中选择。
ActivateFatal-CrashReaction	读/写	ED	BOOL Enum	no / 0 yes / 1	yes / 1	当控制器停止响应时激活/停用特定行为。
IEC 程序						
IECRetainFree	R	AF	DINT	-	-	显示保留区域的可用存储器。
ProjectDate	R	AD	DT	-	-	显示项目日期。
ProjectName	R	AD	STRING	-	-	显示项目名称。

参数	访问	参数类型	数据类型	值	默认值	描述
ProjectTitle	R	AD	STRING	-	-	显示项目标题。
ProjectVersion	R	AD	STRING	-	-	显示项目版本。
ProjectAuthor	R	AD	STRING	-	-	显示项目作者。
ProjectDescrip-tion	R	AD	STRING	-	-	显示项目描述。
ProgrammingSys-tem	R	AD	STRING	-	-	显示编程系统版本 (PSV) 和用于创建程序的设备描述版本 (TSV)。
OnlineChange-Counter	R	AD	DINT	-	0	该值表示自上次下载程序以来已对程序序列的在线更改更新次数。使用此值，可在在线更改后在程序中调用某些初始化步骤。
IECProgramState-Set	读/写	EF	DINT Enum	stop / 0 start / 1	stop / 0	启动/停止 IEC 程序，并显示 IEC 程序的当前状态。
–						
ObjectType	R	AD	STRING	LMCxx8	LMCxx8	对象类型。
stLogicalAddress	R	AD	ST_Logical Address	-	-	控制器参数的逻辑地址。stLogicalAddress = STRUCT (udiType, udiInstance, udiParameterId)
udiType	R	-	UDINT	-	-	
udiInstance	R	-	UDINT	-	-	
udiParameterId	R	-	UDINT	-	-	

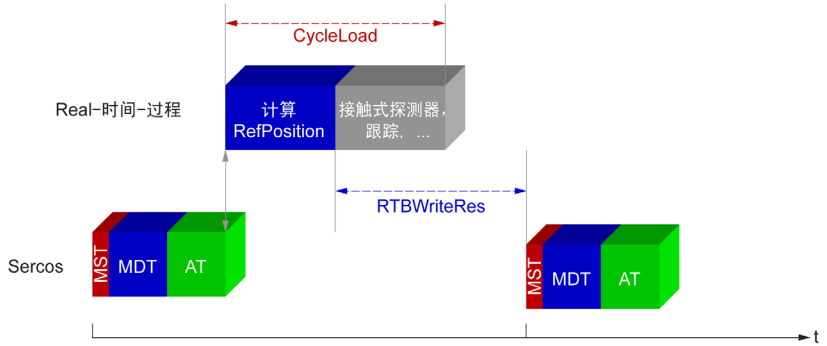
(*) 有关参数访问权限的详细信息，请参阅参数类型 (参见第 *25* 页)。

CycleLoad 参数

参数 CycleLoad 表示实时进程使用的控制器容量 (%)。此参数的值不得超过 40% 至 50% 的平均值（容忍短暂峰值）。然后，剩余 50% 至 60% 可用于其他系统功能，如现场总线服务器和程序。当达到 100% 时发送诊断消息 8511 CPU time overflow。

CycleLoad 参数是用于评估系统负载的简单和清晰的变量。

下图显示了参数 CycleLoad 和 RTBWriteRes 之间的关系：



实时进程 (RTP) 是最重要的系统任务。此任务负责在正确的时间执行所有实时任务。在每个总线循环中，实时处理由 Sercos 实时总线触发。

执行 CycleLoad 包含两个主要步骤：

步骤	描述
1	● 准备循环。 ● 初始化测量变量和监控。 ● 验收由 Sercos 从站的最新驱动器报文 (AT) 提供的实时数据。 ● 处理所有主编码器，如虚拟主站编码器和物理主站编码器。 ● 处理所有从站编码器。 ● 所有轴： ❍ 诊断状态。 ❍ 驱动器的状态机。 ❍ 实时作业准备。 ❍ POS 和 CAM 发生器（主站和从站曲线）。 ❍ 新参考值现在可用。 ❍ 在下一个循环中在主站数据报文 (MDT) 中传输的数据。 ● 在下一个循环中发送要传输的参考值。
2	将会执行其他实时功能，例如： ● 接触式探测器。 ● 跟踪等。

注意： 某些外部事件控制任务以及高优先级任务可能会对实时进程产生负面影响。监控变量 CycleLoad、RTBReadRes 和 RTBWriteRes 可以评估动态行为。

控制器选择

简介

使用此选项卡可管理从 PC 到控制器的连接：

- 帮助您找到网络中的控制器。
- 显示控制器列表，以便您可以连接到所选控制器并管理控制器中的应用程序。
- 帮助您从设备编辑器中以物理方式识别控制器。
- 帮助您处理控制器的通讯设置。

控制器选择工具栏

该工具栏包含下列按钮：

标签	按钮	描述
1	**光信号**	单击该按钮，使所选控制器指示光信号：控制器会快速闪烁控制 LED。如果使用了多个控制器，这可帮助您确定相应控制器。 在单击第二次后该功能会停止，或大约 30 秒后停止。
2	**光信号和声音信号**	不支持。
3	**更新**	单击该按钮刷新控制器的列表。会向网络中的控制器发送请求。响应请求的任何控制器都会连同当前值列出。 会通过每个新请求更新控制器预先存在的条目。 不会删除已经显示在列表中但是没有响应新请求的控制器。通过将红色的十字添加至控制器图标来将它们标记为不活动。 **更新**按钮对应于右键单击列表中的控制器时上下文菜单中提供的**刷新列表**命令。 要刷新所选控制器的信息，可使用上下文菜单提供的命令**刷新此控制器**。该命令从所选控制器请求更多详细信息。 **注意：** **刷新此控制器**命令也可刷新其他控制器的信息。
4	**从列表删除不活动的控制器**	不响应网络扫描的控制器将在列表中被标记为不活动。将通过添加至控制器图标的红色十字对此进行指示。单击此按钮从列表同时删除所有标记为不活动的控制器。 **注意：** 由于网络问题，即使情况并非如此，也可能将控制器标记为不活动状态。 右键单击列表中控制器时打开的上下文菜单中提供另外两个用于删除控制器的命令： ● **从列表中删除选定控制器**命令可让您仅从列表删除所选控制器。 ● **从列表中删除所有控制器**命令可让您从列表同时删除所有控制器。

标签	按钮	描述
5	**新建收藏夹**	可使用**收藏夹**来根据需要调整控制器的选择。这可帮助您保持对网络中许多控制器的跟踪。 **收藏夹**描述通过唯一标识符识别的控制器集合。 单击收藏夹按钮（如**收藏夹 0**）可选择或取消选择收藏夹。如果您尚未选择收藏夹，则会显示所有检测到的控制器。 也可通过上下文菜单访问**收藏夹**。在右键单击列表中的控制器后收藏夹即会打开。 在工具栏中的收藏夹按钮上方移动光标来以工具提示的形式查看相关控制器。
6	**收藏夹 x**	

有关设备编辑器的**控制器选择**视图的详细信息，请参阅 SoMachine 编程指南。

处理通讯设置

使用**处理通讯设置**窗口可以更改以太网通讯设置。为此，请单击**控制器选择**选项卡。此时将显示网络中可用的控制器列表。选择并右键单击所需的行，然后在上下文菜单中单击**处理通讯设置 ...**。

在**处理通讯设置**窗口中，可通过以下 2 种方式配置以太网设置：

- 不使用**永久保存设置**选项：
  配置通讯参数并单击**确定**。这些设置将立即考虑，在控制器复位时不予保留。
- 使用**永久保存设置**选项：
  您也可以在单击**确定**之前激活**永久保存设置**选项。激活此选项后，配置的以太网参数存储在 SD 卡上。控制器复位之后，从 SD 卡配置的以太网参数将处于活动状态。

PLC 设置

概述

下图显示了 **PLC 设置**选项卡：

元素		描述
用于 I/O 处理的应用程序		缺省情况下设置为"应用程序"，因为控制器中只有一个应用程序。
PLC 设置	**处于"停止"状态时更新 IO**	如果已激活此选项，在控制器停止时，输入和输出通道的值也会更新。
	停止时的输出动作	从选择列表中选择下列选项之一，以配置在控制器停止时应如何处理输出通道的值： ● **保持当前值** ● **设置所有输出为默认值** ● **执行程序**
	更新所有设备中的所有变量	如果已激活此选项，那么对于使用当前控制器配置的所有设备，所有 I/O 变量在总线循环任务的每次循环中都会更新。这对应于**一直更新变量**选项（可以在 **I/O 映射**对话框中针对每个设备单独设置该选项）。
总线循环选项	**总线循环任务**	此配置设置是在应用程序设备树中使用的所有**总线循环任务**参数的父级。 具有循环调用的某些设备（如 **CANopen 管理器**）可以附加到特定任务。在设备中，将此设置设为**使用父总线循环设置**时，会使用为控制器配置的设置。 此选择列表显示活动应用程序中当前定义的所有任务。缺省设置为 MAST 任务。 **注意：** **<未指定>** 意味着任务处于"最慢的循环任务"模式下。
附加设置	**生成 IO 映射的强制变量**	未使用。
	启用设备诊断	未使用。

第9章
内嵌输入和输出配置

本章包含了哪些内容？

本章包含了以下主题：

主题	页
嵌入式 I/O 配置	82
主编码器输入配置	89

嵌入式 I/O 配置

简介

该 Modicon LMC078 Motion Controller 可提供：

- 12 个嵌入式输入：
  - 8 路数字量输入：DI_0...DI_7
  - 4 个高级数字量输入（接触式探测器和中断）：ADI_0...ADI_3
- 8 个嵌入式输出：
  - 7 路数字量输出：DQ_0...DQ_6
  - 1 个可配置为看门狗输出的数字量输出：DQ_WD

数字量输入组配置

要配置数字量输入组，双击**设备树**中的 DIG_DigitalIn 节点。

下表介绍了不同参数：

参数	访问	参数类型	数据类型	描述
Name	读/写(*)	EF	STRING(40)	配置对象的符号名称。
Bit0_7	R	AF	USINT	数字量输入 DI_0...DI_7 的值，为每个输入均分配一个位。 位 x = 输入 DI_x 的值
Bit8_11	R	AF	USINT	高级数字量输入 ADI_0...ADI_3 的值，为每个高级输入均分配一个位。 位 x = 输入 ADI_x 的值
ObjectType	R	AD	STRING	对象类型。
stLogicalAddress	R	AD	ST_LogicalAddress	输入组的逻辑地址。

(*) 有关参数访问权限的详细信息，请参阅参数类型 (参见第 *25* 页)。

数字量输入配置

要配置数字量输入，双击**设备树**中的 **DI_x** 节点。

DIG_DigitalIn | DI_0

配置

参数	类型	值	缺省值	单位	描述
名称	STRING(40)	"	"		对象名称 (EF)[0x0018]
值	BOOL 枚举				实际值 (AD)[0x0001]
FilterTime	UDINT(90..4294967)	100	100	微秒	inputfiltertime（步长为 1 微秒）(EF)[0x001A]
ObjectType	STRING	'D_IN5'	'D_IN5'		对象类型 (AD)[0x10000001]
stLogicalAddress					扩展逻辑地址 (AD)[0x10000003]
udiType	UDINT	0	0		
udiInstance	UDINT	0	0		
udiParameterId	UDINT	0	0		

下表介绍输入参数：

参数	访问	参数类型	数据类型	值	默认值	描述
Name	读/写(*)	EF	STRING(40)	“	“	配置对象的符号名称。
Value	R	AD	BOOL Enum	L/0 H/1	-	数字量输入的值。
FilterTime	读/写	EF	UDINT	90...4294967	100	输入的过滤时间（微秒）。
ObjectType	R	AD	STRING	D_IN5	D_IN5	对象类型。
stLogicalAddress	R	AD	ST_LogicalAddress	-	-	输入的逻辑地址。

(*) 有关参数访问权限的详细信息，请参阅参数类型 (参见第 *25* 页)。

高级数字量输入配置

要配置高级数字量输入，双击**设备树**中的 ADI_x 节点。

DIG_DigitalIn | DI_0 | ADI_0

配置

参数	类型	值	缺省值	单位	描述
名称	STRING(40)	"	"		对象名称 (EF)[0x0018]
值	BOOL 枚举				实际值 (AD)[0x0001]
FilterTime	UDINT(90..4294967)	100	100	微秒	inputfiltertime（步长为 1 微秒）(EF)[0x001A]
CaptureState	DINT 枚举	不活动/0	不活动/0		捕捉功能状态 (AD)[0x0005]
SensorDelay	LREAL(-100..100)	0.0	0.0	毫秒	传感器延迟 (EF)[0x0004]
计数器	UDINT	0	0		输入计数器 (EF)[0x000C]
启用	BOOL 枚举	关闭/0	关闭/0		启用输入中断 (EF)[0x000D]
ExtEventEdge	DINT 枚举	正/1	正/1		输入活动边 (EF)[0x001B]
ObjectType	STRING	'D_IN62'	'D_IN62'		对象类型 (AD)[0x10000001]
stLogicalAddress					扩展逻辑地址 (AD)[0x10000003]
udiType	UDINT	0	0		
udiInstance	UDINT	0	0		
udiParameterId	UDINT	0	0		

下表介绍了高级输入参数：

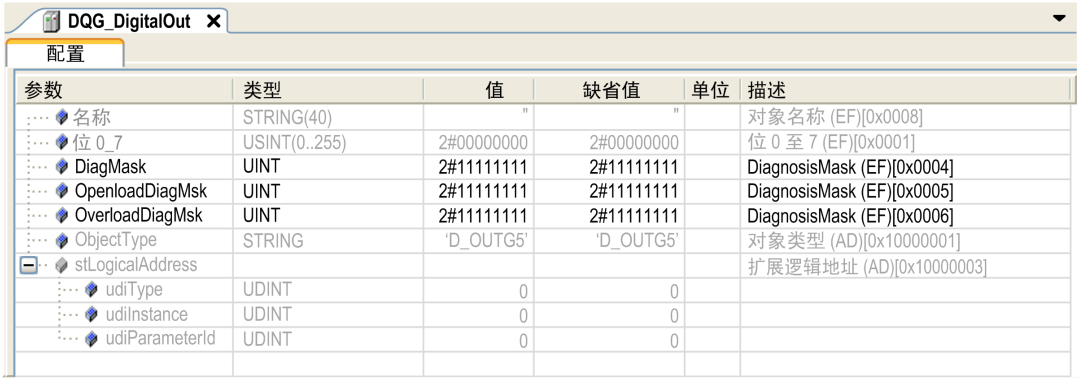
参数	访问	参数类型	数据类型	值	默认值	描述
Name	读/写(*)	EF	STRING(40)	“	“	配置对象的符号名称。
Value	R	AD	BOOL Enum	L / 0 H / 1	-	高级数字量输入的值。
FilterTime	读/写	EF	UDINT	90...4294967	100	输入的过滤时间（微秒）。
CaptureState	R	AD	DINT Enum	inactive / 0 active / 1 captured / 2 overflow / 3 disabled / 4 not ready / 5 virtual / 6	inactive / 0	捕捉功能的状态。
SensorDelay	读/写	EF	LREAL	-100...100	0	传感器延迟（毫秒）。
Counter	R	EF	UDINT	-	0	输入的计数器。
Enable	读/写(*)	EF	BOOL Enum	off / 0 on / 1	关闭/0	启用输入的中断功能。
ExtEventEdge	读/写	EF	DINT Enum	negative / 0 positive / 1 negative and positive / 2	正/1	定义输入的活动边。

参数	访问	参数类型	数据类型	值	默认值	描述
ObjectType	R	AD	STRING	D_IN62	D_IN62	对象类型。
stLogicalAddress	R	AD	ST_LogicalAddress	-	-	高级输入的逻辑地址。

(*) 有关参数访问权限的详细信息，请参阅参数类型 (参见第 *25* 页)。

数字量输出组配置

要配置数字量输出组，双击**设备树**中的 DQG_DigitalOut 节点。



下表介绍了不同参数：

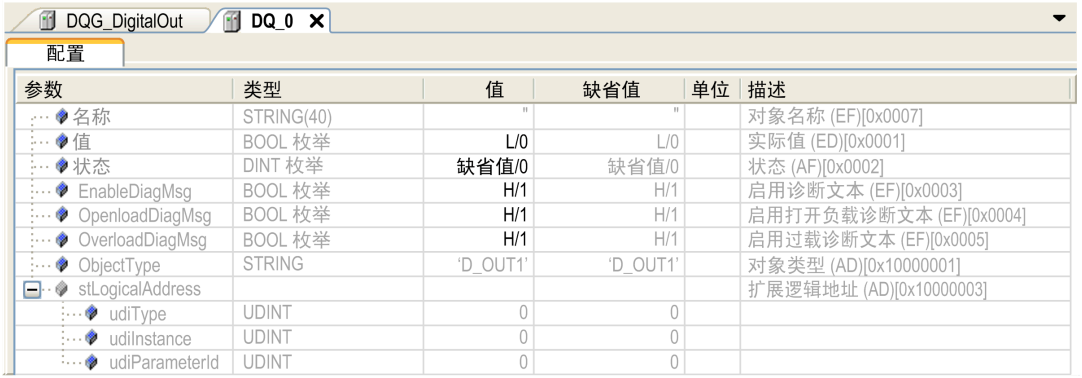
参数	访问	参数类型	数据类型	值	默认值	描述
Name	读/写(*)	EF	STRING(40)	“	“	配置对象的符号名称。
Bit0_7	读/写(*)	EF	USINT	0	0	数字量输出 DQ_0...DQ_7 的值，为每个输出均分配一个位。 位 x = 输出 DQ_x 的值
DiagMask	读/写	EF	UINT	2#11111111	2#11111111	启用每个输出的诊断消息 8788 Wiring error，为每个输出均分配一个位。 位 x = 0，未监控输出 x，且未显示诊断消息。
OpenloadDiagMask	读/写	EF	UINT	2#11111111	2#11111111	启用每个输出的诊断消息 8788 Wiring error / Openload，为每个输出均分配一个位。 位 x = 0，未监控输出 x，且未显示诊断消息。

参数	访问	参数类型	数据类型	值	默认值	描述
OverloadDiagMask	读/写	EF	UINT	2#11111111	2#11111111	启用每个输出的诊断消息 8788 Wiring error / Overload，为每个输出均分配一个位。 位 x = 0，未监控输出 x，且未显示诊断消息。
ObjectType	R	AD	STRING	D_OUTG5	D_OUTG5	对象类型。
stLogicalAddress	R	AD	ST_Logical Address	-	-	输出组的逻辑地址。

(*) 有关参数访问权限的详细信息，请参阅参数类型 (参见第 *25* 页)。

数字量输出配置

要配置数字量输出，双击**设备树**中的 **DQ_x** 节点。



参数	类型	值	缺省值	单位	描述
名称	STRING(40)	"	"		对象名称 (EF)[0x0007]
值	BOOL 枚举	L/0	L/0		实际值 (ED)[0x0001]
状态	DINT 枚举	缺省值/0	缺省值/0		状态 (AF)[0x0002]
EnableDiagMsg	BOOL 枚举	H/1	H/1		启用诊断文本 (EF)[0x0003]
OpenloadDiagMsg	BOOL 枚举	H/1	H/1		启用打开负载诊断文本 (EF)[0x0004]
OverloadDiagMsg	BOOL 枚举	H/1	H/1		启用过载诊断文本 (EF)[0x0005]
ObjectType	STRING	'D_OUT1'	'D_OUT1'		对象类型 (AD)[0x10000001]
stLogicalAddress					扩展逻辑地址 (AD)[0x10000003]
udiType	UDINT	0	0		
udiInstance	UDINT	0	0		
udiParameterId	UDINT	0	0		

下表介绍输出参数：

参数	访问	参数类型	数据类型	值	默认值	描述
Name	读/写(*)	EF	STRING(40)	"	"	配置对象的符号名称。
Value	读/写(*)	ED	BOOL Enum	L/0 H/1	L/0	显示数字量输出的值。
Status	R	AF	DINT Enum	default / 0 openload / 1 overload / 2	default / 0	显示数字量输出的状态： ● 0：缺省状态 ● 1：无加载可用 ● 2：存在短路

参数	访问	参数类型	数据类型	值	默认值	描述
EnableDiagMsg	读/写(*)	EF	BOOL Enum	L / 0 H / 1	H / 1	0 = 输出不报告诊断消息 8788 Wiring error。 1 = 输出报告诊断消息 8788 Wiring error。
OpenloadDiagMsg	读/写(*)	EF	BOOL Enum	L / 0 H / 1	H / 1	0 = 输出不报告诊断消息 8788 Wiring error / Openload。 1 = 输出报告诊断消息 8788 Wiring error / Openload。
OverloadDiagMsg	读/写(*)	EF	BOOL Enum	L / 0 H / 1	H / 1	0 = 输出不报告诊断消息 8788 Wiring error / Overload。 1 = 输出报告诊断消息 8788 Wiring error / Overload。
ObjectType	R	AD	STRING	D_OUT1	D_OUT1	对象类型。
stLogicalAddress	R	AD	ST_Logical Address	-	-	输出的逻辑地址。

(*) 有关参数访问权限的详细信息，请参阅参数类型 (参见第 *25* 页)。

看门狗输出配置

要配置看门狗输出，双击**设备树**中的 **DQ_WD** 节点。

参数	类型	值	缺省值	单位	描述
名称	STRING(40)	"	"		对象名称 (EF)[0x0007]
值	BOOL 枚举	L/0	L/0		实际值 (ED)[0x0001]
状态	DINT 枚举	缺省值/0	缺省值/0		状态 (AF)[0x0002]
EnableDiagMsg	BOOL 枚举	H/1	H/1		启用诊断文本 (EF)[0x0003]
OpenloadDiagMsg	BOOL 枚举	H/1	H/1		启用打开负载诊断文本 (EF)[0x0004]
OverloadDiagMsg	BOOL 枚举	H/1	H/1		启用过载诊断文本 (EF)[0x0005]
WDOutEnable	BOOL 枚举	关闭/0	关闭/0		输出由看门狗控制 (ED)[0x0008]
ObjectType	STRING	'D_OUT2'	'D_OUT2'		对象类型 (AD)[0x10000001]
stLogicalAddress					扩展逻辑地址 (AD)[0x10000003]
udiType	UDINT	0	0		
udiInstance	UDINT	0	0		
udiParameterId	UDINT	0	0		

下表介绍了看门狗参数：

参数	访问	参数类型	数据类型	值	默认值	描述
Name	读/写(*)	EF	STRING(40)	“	“	配置对象的符号名称。
Value	读/写(*)	ED	BOOL Enum	L/0 H/1	L/0	显示看门狗输出的值。
Status	R	AF	DINT Enum	default / 0 openload / 1 overload / 2	default / 0	显示看门狗输出的状态： ● 0：缺省状态 ● 1：无加载可用 ● 2：存在短路
EnableDiagMsg	读/写(*)	EF	BOOL Enum	L / 0 H / 1	H / 1	0 = 输出不报告诊断消息 8788 Wiring error。 1 = 输出报告诊断消息 8788 Wiring error。
OpenloadDiagMsg	读/写(*)	EF	BOOL Enum	L / 0 H / 1	H / 1	0 = 输出不报告诊断消息 8788 Wiring error / Openload。 1 = 输出报告诊断消息 8788 Wiring error / Openload。
OverloadDiagMsg	读/写(*)	EF	BOOL Enum	L / 0 H / 1	H / 1	0 = 输出不报告诊断消息 8788 Wiring error / Overload。 1 = 输出报告诊断消息 8788 Wiring error / Overload。
WDOutEnable	读/写	ED	BOOL Enum	off / 0 on / 1	off / 0	0 = 看门狗输出已停用。 1 = 看门狗输出已激活，且受系统看门狗 (参见第 *43* 页)控制。
ObjectType	R	AD	STRING	D_OUT2	D_OUT2	对象类型。
stLogicalAddress	R	AD	ST_Logical Address	-	-	看门狗输出的逻辑地址。

(*) 有关参数访问权限的详细信息，请参阅参数类型 (参见第 *25* 页)。

主编码器输入配置

简介

控制器具有特定的硬件编码器接口，该接口可以支持：

- 增量编码器 (RS422)
- 绝对编码器 (SinCos Hiperface)

此功能的目的是连接编码器以获取与 Sercos 实时总线同步的位置。然后，将获取的此位置用作 Sercos 上运动驱动器的主轴。

主编码器配置分为两个部分：

- 支持硬件配置的编码器节点。
- 支持比例调整配置的 **SoftMotion_Encoder** 节点。

添加编码器

要将编码器添加到控制器，选择**硬件目录**中的**增量编码器输入**或**SinCos 编码器输入**，然后将其拖放到**设备树**，并将其放置在控制器节点上。

结果：已将编码器添加到控制器作为新子节点，以及添加 **SoftMotion_Encoder** 节点作为编码器的子节点。

有关将设备添加到项目的更多信息，请参阅：

• 使用拖放方法 *(参见 SoMachine, 编程指南)*

• 使用上下文菜单或加号按钮 *(参见 SoMachine, 编程指南)*

增量编码器配置

要配置增量编码器，双击**设备树**中的编码器节点：

IME_EncIn

配置

参数	类型	值	缺省值	单位	描述
名称	STRING(40)	"	"		对象名称 (EF)[0x0014]
启用	BOOL 枚举	开启/1	开启/1		启用 (EF)[0x0001]
州/省	DINT 枚举	未就绪/0	未就绪/0		增量输入状态 (AD)[0x000E]
过滤器	DINT(0..1024)	0	0	毫秒	过滤器 (EF)[0x0004]
CheckOff	BOOL 枚举	否/0	否/0		编码器检查关闭 (EF)[0x0005]
ZeroTrackCheckOff	BOOL 枚举	否/0	否/0		零点轨迹检查关闭 (EF)[0x0013]
ZeroTrackStart	BOOL 枚举	False/0	False/0		零点轨迹检测开始 (EF)[0x0009]
ZeroTrackDetected	BOOL 枚举	False/0	False/0		检测到零点轨迹 (AD)[0x000A]
DiagClass	DINT				诊断类别 (AD)[0x0077]
DiagCode	DINT				诊断代码 (AD)[0x000F]
DiagSource					诊断源 (AD)[0x0078]
DiagMsg	STRING(39)	"	"		诊断文本 (AD)[0x0011]
DiagExtMsg	STRING(14)	"	"		扩展诊断消息 (AD)[0x0079]
ObjectType	STRING	'INC_IN2'	'INC_IN2'		对象类型 (AD)[0x10000001]
stLogicalAddress					扩展逻辑地址 (AD)[0x10000003]

下表介绍了增量编码器参数：

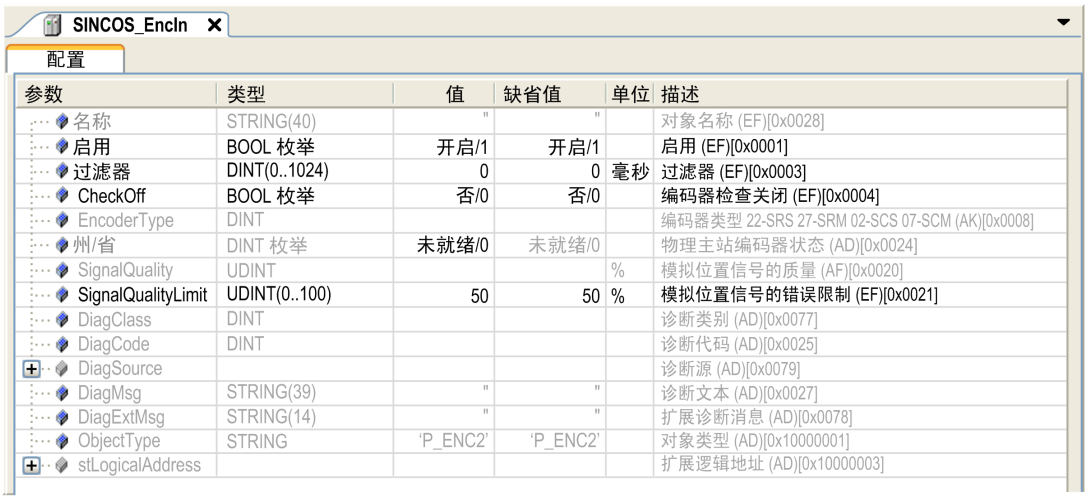
参数	访问	参数类型	数据类型	值	缺省值	描述
Name	读/写(*)	EF	STRING(40)	“	“	配置对象的符号名称。
Enable	读/写	EF	BOOL Enum	off / 0 on / 1	on / 1	启用编码器： ● 0 = 实时进程未处理编码器。 ● 1 = 实时进程已处理编码器。
State	R	AD	DINT Enum	not ready / 0 initialization / 1 no sync / 2 ready / 3	not ready / 0	显示编码器位置数据的可用性和有效性： ● 0 = 编码器或编码器处理未准备就绪。 ● 1 = 编码器处理初始化。 ● 2 = 编码器处理运行与 RTP 不同步。 ● 3 = 编码器的速度和位置值均有效。
Filter	读/写	EF	DINT	0...1024	0	过滤值（毫秒）。 该过滤器是一个低通滤波器，且会影响实际速度。如果将过滤器设置过高，则可能会导致系统出现振荡现象，取决于速度路径。
CheckOff	读/写	EF	BOOL Enum	no / 0 yes / 1	no / 0	禁用编码器监控： ● 0=编码器验证处于活动状态。 ● 1 = 编码器监控处于不活动状态（跟踪和电缆监控）。
ZeroTrackCheck-Off	读/写	EF	BOOL Enum	no / 0 yes / 1	no / 0	禁用编码器的零点轨迹监控： ● 0 = 零点轨迹监控处于活动状态。 ● 1 = 零点轨迹监控处于不活动状态。
ZeroTrackStart	读/写(*)	EF	BOOL Enum	FALSE / 0 TRUE / 1	FALSE / 0	此参数的上升沿可实现零点轨迹检测： ● 0 = 已禁用零点轨迹检测。 ● 1 = 已启用零点轨迹检测。
ZeroTrackDetect-ed	R	AD	BOOL Enum	FALSE / 0 TRUE / 1	FALSE / 0	在启用零点轨迹检测后，此参数表示零点轨迹检测： ● 0 = 未检测到零点轨迹。 ● 1 = 检测到零点轨迹。 检测到零点轨迹后，会将编码器的位置设置为 0。
DiagClass	R	AD	DINT	-	-	以十进制代码显示诊断类别 (参见第 *237* 页)。

参数	访问	参数类型	数据类型	值	缺省值	描述
DiagCode	R	AD	DINT	-	-	以十进制代码显示诊断代码 (参见第 *237* 页)。
DiagSource	R	AD	ST_Logical Address	-	-	诊断源的逻辑地址。
DiagMsg	R	AD	STRING	-	-	诊断消息。
DiagExtMsg	R	AD	STRING	-	-	扩展诊断消息。
ObjectType	R	AD	STRING	INC_IN2	INC_IN2	对象类型。
stLogicalAddress	R	AD	ST_Logical Address	-	-	增量编码器的逻辑地址。

(*) 有关参数访问权限的详细信息，请参阅参数类型 (参见第 *25* 页)。

Hiperface (SinCos) 编码器配置

要配置 Hiperface 编码器，双击**设备树**中的编码器节点：



参数	类型	值	缺省值	单位	描述
名称	STRING(40)	"	"		对象名称 (EF)[0x0028]
启用	BOOL 枚举	开启/1	开启/1		启用 (EF)[0x0001]
过滤器	DINT(0..1024)	0	0	毫秒	过滤器 (EF)[0x0003]
CheckOff	BOOL 枚举	否/0	否/0		编码器检查关闭 (EF)[0x0004]
EncoderType	DINT				编码器类型 22-SRS 27-SRM 02-SCS 07-SCM (AK)[0x0008]
州/省	DINT 枚举	未就绪/0	未就绪/0		物理主站编码器状态 (AD)[0x0024]
SignalQuality	UDINT			%	模拟位置信号的质量 (AF)[0x0020]
SignalQualityLimit	UDINT(0..100)	50	50	%	模拟位置信号的错误限制 (EF)[0x0021]
DiagClass	DINT				诊断类别 (AD)[0x0077]
DiagCode	DINT				诊断代码 (AD)[0x0025]
DiagSource					诊断源 (AD)[0x0079]
DiagMsg	STRING(39)	"	"		诊断文本 (AD)[0x0027]
DiagExtMsg	STRING(14)	"	"		扩展诊断消息 (AD)[0x0078]
ObjectType	STRING	'P_ENC2'	'P_ENC2'		对象类型 (AD)[0x10000001]
stLogicalAddress					扩展逻辑地址 (AD)[0x10000003]

下表介绍了 Hiperface 编码器参数：

参数	访问	参数类型	数据类型	值	缺省值	描述
Name	读/写(*)	EF	STRING(40)	“	“	配置对象的符号名称。
Enable	读/写	EF	BOOL Enum	off / 0 on / 1	on / 1	启用编码器： ● 0 = 实时进程未处理编码器。 ● 1 = 实时进程已处理编码器。
Filter	读/写	EF	DINT	0...1024	0	过滤值（毫秒）。 该过滤器是一个低通滤波器，且会影响实际速度。如果将过滤器设置过高，则可能会导致系统出现振荡现象，取决于速度路径。
CheckOff	读/写	EF	BOOL Enum	no / 0 yes / 1	no / 0	禁用编码器监控： ● 0 = 编码器监控处于活动状态。 ● 1 = 编码器监控处于不活动状态。 已启用诊断消息 8601 Master Encoder signal out of range。
EncoderType	R	AK	DINT	-	-	显示支持的 SinCos Hiperface 编码器类型（在控制器启动时读取）： ● 00h：尚未读取的值 ● 02h：SCS 60/70 ● 07h：SCM 60/70 多转 ● 22h：SRS 50/60/64 和 SCK 25/35/40/45/53 单转 ● 27h：SRM 50/60/64 和电机 25/35/40/45/53 多转 ● 32h：SKS 36 单转 ● 37h：SKM 36 多转 ● 42h：SEK 52 单转
State	R	AD	DINT Enum	not ready / 0 initialization / 1 no sync / 2 get type / 3 ready / 10 read position / 11 write position / 12 read error code / 13	not ready / 0	显示编码器位置数据的可用性和有效性： ● 0：编码器或编码器处理未准备就绪。 ● 1：编码器处理初始化。 ● 2：编码器处理运行与 RTP 不同步。 ● 3：已读取编码器类型。 ● 4...9：状态无效。 ● 10：编码器的速度和位置值均有效。 ● 11：从编码器读取绝对位置。 ● 12：将绝对位置写入编码器。 ● 13：从编码器读取错误代码。
SignalQual-ity	R	AF	UDINT	-	-	说明编码器的模拟量正弦和余弦轨迹的信号质量（百分比）。 正弦和余弦信号必须对应于以下公式： 正弦信号² + 余弦信号² = 1 (100%) 此参数表示该公式规范化到 100%。

参数	访问	参数类型	数据类型	值	缺省值	描述
SignalQualityLimit	R	EF	UDINT	50	50	此参数确定发出诊断代码 8601 Master encoder signal out of the range 的 SignalQuality 的值。
DiagClass	R	AD	DINT	-	-	以十进制代码显示诊断类别 (参见第 *237* 页)。
DiagCode	R	AD	DINT	-	-	以十进制代码显示诊断代码 (参见第 *237* 页)。
DiagSource	R	AD	ST_Logical Address	-	-	诊断源的逻辑地址。
DiagMsg	R	AD	STRING	-	-	诊断消息。
DiagExtMsg	R	AD	STRING	-	-	扩展诊断消息。
ObjectType	R	AD	STRING	P_ENC2	P_ENC2	对象类型。
stLogicalAddress	R	AD	ST_Logical Address	-	-	Hiperface 编码器的逻辑地址。

SoftMotion 编码器配置

要配置 SoftMotion 编码器，双击**设备树**中的 **SoftMotion_Encoder** 节点。

有关 SoftMotion 编码器配置的详细信息，请参阅 SoMachine 在线帮助中的*使用 SoMachine/SoftMotion/SoftMotion 设备编辑器编程*一章。

第10章
通讯模块

简介

本章介绍如何添加和配置通讯模块。

本章包含了哪些内容？

本章包含了以下部分：

节	主题	页
10.1	PROFIBUS DP 从站模块配置	96
10.2	EtherNet/IP 适配器配置	105
10.3	EtherNet/IP 扫描器配置	114

第10.1节
PROFIBUS DP 从站模块配置

简介

本节介绍 VW3E704000000 PROFIBUS DP 模块的配置。

本节包含了哪些内容？

本节包含了以下主题：

主题	页
添加 PROFIBUS DP 从站模块	97
PROFIBUS DP 从站模块配置	99
非循环数据交换	103

添加 PROFIBUS DP 从站模块

概述

根据主站/从站原理，可使用 PROFIBUS 协议进行数据交换。只有主站才能初始化通讯。从站响应来自主站的请求。多个主站可在同一总线上共存。在这种情况下，所有主站都可以读取从站 I/O。但是，单个主站对输出具有写访问权限。交换的数据项数量在配置过程中定义。

该模块支持 2 种类型的交换服务：

- I/O 循环帧交换
- 使用 PROFIBUS DPV1 功能进行非循环数据交换

添加 PROFIBUS DP 从站模块

在**硬件目录**中选择 **PROFIBUS-DPV1-Slave** 模块，然后将其拖放到**设备树**，并将其放置在控制器节点上。

有关将设备添加到项目的更多信息，请参阅：

- 使用拖放方法 *(参见 SoMachine, 编程指南)*
- 使用上下文菜单或加号按钮 *(参见 SoMachine, 编程指南)*

添加虚拟 I/O 设备

下面的 PROFIBUS DP 从站模块可让您添加一个或多个虚拟 I/O 设备。

PROFIBUS DP 模块是 PROFIBUS 主站和控制器之间的媒介，而数据是通过使用您在配置模块时所定义的虚拟 I/O 设备进行交换。虚拟设备不是物理 I/O 模块，它们只是 模块中的逻辑输入和输出对象，您随后可将它们映射到控制器内的存储器。这些输入和输出对象由 PROFIBUS 主站进行读取和写入。相应地，模块会读取该数据并将其写入控制器中的 I/O 存储器位置，使您能够在应用程序中使用该数据。

在**硬件目录**中选择虚拟 I/O 设备，然后将其拖放到**设备树**，并将其放置在 **PROFIBUS-DPV1-Slave** 节点上。

有关将设备添加到项目的更多信息，请参阅：

- 使用拖放方法 *(参见 SoMachine, 编程指南)*
- 使用上下文菜单或加号按钮 *(参见 SoMachine, 编程指南)*

您在模块中定义的虚拟 I/O 设备可以是输入或输出，大小也各不相同，如下表所定义：

名称	I/O 数	格式
X byte input (0x••)	X= 1,2,3,4,8,12,16,20,32 或 64	字节
X byte input con (0x••)	X= 1,2,3,4,8,12,16,20,32 或 64	
X byte output (0x••)	X= 1,2,3,4,8,12,16,20,32 或 64	
X byte output con (0x••)	X= 1,2,3,4,8,12,16,20,32 或 64	
X word input (0x••)	X= 1,2,3,4,8,12,16,20,32 或 64	字
X word input con (0x••)	X= 1,2,3,4,8,12,16,20,32 或 64	
X word output (0x••)	X= 1,2,3,4,8,12,16,20,32 或 64	
X word output con (0x••)	X= 1,2,3,4,8,12,16,20,32 或 64	

在模块中定义这些虚拟输入和/或输出设备后，可以将这些设备映射到控制器内的存储器位置。您将这些虚拟 I/O 设备映射到的存储器对象的类型取决于您在主站和从站之间定义的交换类型。

PROFIBUS DP 从站模块配置

PROFIBUS DP 从站模块配置

在**设备树**中，双击 **PBS_Slave (PROFIBUS-DPV1-Slave)**：

PBS_Slave

PROFIBUS 配置 | PROFIBUS I/O 映射 | 状态 | 信息

参数	类型	值	缺省值	单位
InitParameters				
BusAddr	字节 (2..126)	2	2	
BaudRate	BYTE 枚举	Baudrate_AUTO/15	Baudrate_AUTO/15	
连接器	INT 枚举	前置/1	前置/1	
未使用	BYTE 枚举	NOT_IN_USE	NOT_IN_USE	
WdgEnabled	BOOL	TRUE	TRUE	
WdgTime	UDINT(20..65535)	100	100	
DPV1Enable	BOOL	TRUE	TRUE	
SyncSupported	BOOL	TRUE	TRUE	
FreezeSupported	BOOL	TRUE	TRUE	
FailsafeSupported	BOOL	TRUE	TRUE	
NoAddrChangeSupported	BOOL	TRUE	TRUE	
内部				
信息				
诊断				

黑色参数 读/写访问
灰色参数 只读访问

在 **PROFIBUS Configuration** 选项卡中提供了以下参数：

参数	类型	值	缺省值	描述
InitParameters				
BusAddr	BYTE	2...126	2	PROFIBUS DP 从站地址。
BaudRate	BYTE Enum	0...15	15	PROFIBUS 传输速率： ● 0 = 9600 波特 ● 1 = 19.2 千波特 ● 2 = 93.75 千波特 ● 3 = 187.5 千波特 ● 4 = 500 千波特 ● 6 = 1500 千波特 ● 7 = 3 兆波特 ● 8 = 6 兆波特 ● 9 = 12 兆波特 ● 10 = 31.25 兆波特 ● 11 = 45.45 兆波特 ● 15 = 通过从站识别在主站上设置的自动传输率。

参数	类型	值	缺省值	描述
WdgEnabled	BOOL	TRUE FALSE	TRUE	启用看门狗： ● TRUE = 已激活看门狗。将参数 WdgTime 的值用作看门狗超时。 ● FALSE = 已停用看门狗。忽略参数 WdgTime 的值。
WdgTime	UDINT	20...65535	100	确定看门狗将从站检测为不可通讯之后的时间。已指定该时间（毫秒）。
DPV1Enable	BOOL	TRUE FALSE	TRUE	启用 DPV1 功能进行非循环通讯： ● TRUE = 已激活 DPV1 模式。 ● FALSE = 已禁用 DPV1 模式。
SyncSupported	BOOL	TRUE FALSE	TRUE	启用同步模式以支持同步命令： ● TRUE = 已激活同步模式。 ● FALSE = 已禁用同步模式。
FreezeSupported	BOOL	TRUE FALSE	TRUE	启用冻结模式以支持冻结命令： ● TRUE = 已激活冻结模式。 ● FALSE = 已禁用冻结模式。
FailsafeSup-ported	BOOL	TRUE FALSE	TRUE	启用专有“failsafe”模式（如设备支持）： ● TRUE = 已激活模式。 ● FALSE = 已禁用模式。
Internal				
保留				
Info				
IdentNumber	WORD	3424	3424	显示 PROFIBUS DP 从站模块的标识号。
VendorName	STRING	‘Schneider Electric’	‘Schneider Electric’	显示 PROFIBUS DP 从站模块的供应商名称。
ModelName	STRING	‘PROFIBUS DPV1 slave’	‘PROFIBUS DPV1 slave’	显示 PROFIBUS DP 从站模块的型号名称。
DriverInstance	DWORD	0	0	显示此驱动器实例的标识符。
Diag / ChannelCommonStatusBlock				
CommState	UDINT Enum	UNKNOWN / 0 NOT_CONFIGURED / 1 STOP / 2 IDLE / 3 OPERATE / 4	UNKNOWN / 0	显示通讯通道的当前网络状态： ● 0 = 无法确定。 ● 1 = 未配置。 ● 2 = 已停止。 ● 3 = 空闲。 ● 4 = 正常操作。
CommError	UDINT Enum	诊断代码 (参见第 *101* 页)	SUCCESS / 0x0	显示通讯通道的当前诊断代码。
ErrorCount	UDINT	-	0	显示自上一次启动或最后一次重新启动以来检测到的总错误数。

参数	类型	值	缺省值	描述
Diag / ChannelExtentedStatusBlock				
Baudrate	UDINT Enum	0...15	Baudrate_ AUTO / 15	显示应用的波特率。如果已在从站的初始化参数中设置值 AUTO，则该参数显示主站使用的波特率。 PROFIBUS 传输速率： ● 0 = 9600 波特 ● 1 = 19.2 千波特 ● 2 = 93.75 千波特 ● 3 = 187.5 千波特 ● 4 = 500 千波特 ● 6 = 1500 千波特 ● 7 = 3 兆波特 ● 8 = 6 兆波特 ● 9 = 12 兆波特 ● 10 = 31.25 兆波特 ● 11 = 45.45 兆波特 ● 15 = 主站无法自动连接到从站。

诊断代码

未检测到错误：

值	含义
SUCCESS / 0x0	未检测到错误。

检测到运行时错误：

值	含义
WATCHDOG_TIMEOUT/ 0xC000000C	已超过看门狗时间。

检测到初始化错误：

值	含义
INIT_FAULT/ 0xC0000100	初始化不成功。
DATABASE_ACCESS_FAILED/ 0xC0000101	对数据存储器的访问不成功。

检测到配置错误：

值	含义
NOT_CONFIGURED/ 0xC0000119	模块未配置。
CONFIGURATION_FAULT / 0xC0000120	检测到配置错误。
INCONSISTENT_DATA_SET/ 0xC0000121	检测到不一致的设置数据。
DATA_SET_MISMATCH/ 0xC0000122	检测到设置数据不匹配。
INSUFFICIENT_LICENSE/ 0xC0000123	检测到许可证不足。
PARAMETER_ERROR/ 0xC0000124	检测到参数错误。
INVALID_NETWORK_ADDRESS/ 0xC0000125	网络地址不正确。
SECURITY_MEMORY/ 0xC0000126	安全存储器不可用。

检测到网络错误：

值	含义
COMM_NETWORK_FAULT/ 0xC0000126	检测到网络通讯错误。
COMM_CONNECTION_CLOSED/ 0xC0000141	通讯连接已关闭。
COMM_CONNECTION_TIMEOUT/ 0xC0000142	检测到通讯连接超时。
COMM_DUPLICATE_NODE/ 0xC0000144	检测到重复节点。
COMM_CABLE_DISCONNECT / 0xC0000145	检测到电缆断开。
PROFIBUS_CONNECTION_TIMEOUT/ 0xC009002E	检测到 PROFIBUS 连接超时。

非循环数据交换

注册回调

可以在非循环查询的情况下注册回调。如果 PROFIBUS DPV1 从站驱动器收到主站的非循环查询，则首先会将查询传递到已注册数据区域。

然后，所有已注册用户功能块 (FB) 实例通过使用 IF_AsyncRead 和 IF_AsyncWrite 接口调用已注册到 PROFIBUS DPV1 从站的实例。同时，也会发送自动查询处理的状态，便于解决可能出现的问题。然后，将当前状态写入参数 iq_stError。

要在传入读取查询的情况下调用 AsyncRead 方法，必须将与用来实施 IF_AsyncRead 接口的功能块相同类型的实例注册到 PROFIBUS DPV1 从站。

如果在收到非循环读取查询时再也无法调用此功能块实例的 AsyncRead 方法，则必须删除实例的注册。

注意： 使用方法 IsRegisteredAsyncRead 来验证是否通过使用 PROFIBUS DPV1 从站已将某些功能块实例注册为回调。

下文列出了所有相关方法的列表。

注意： 当您添加模块时，已经将 IoDrvPROFIBUSDPV1Slave 库添加到库管理器。该库包含用于管理 PROFIBUS DP 模块的接口、方法和功能块。

回调注册的相关方法

下表列出了用于注册 PROFIBUS DP 实例的相关方法：

方法	描述
IsRegisteredAsyncAlarmAck	在警报确认到达后，返回是否已经为回调注册的传送功能块。
IsRegisteredAsyncRead	在非循环写入查询到达后，返回是否已经为回调注册的传送功能块。
IsRegisteredAsyncWrite	在非循环读取查询到达后，返回是否已经为回调注册的传送功能块。
RegisterAsyncAlarmAck	在警报确认到达后，将用于回调的传送功能块实例注册到 PROFIBUS 从站。
RegisterAsyncRead	在非循环读取查询到达后，将用于回调的传送功能块实例注册到 PROFIBUS 从站。
RegisterAsyncWrite	在非循环写入查询到达后，将用于回调的传送功能块实例注册到 PROFIBUS 从站。
UnregisterAsyncAlarmAck	在警报确认到达后，取消将用于回调的传送功能块实例注册到 PROFIBUS 从站。
UnregisterAsyncRead	在非循环读取查询到达后，取消将用于回调的传送功能块实例注册到 PROFIBUS 从站。
UnregisterAsyncWrite	在非循环写入查询到达后，取消将用于回调的传送功能块实例注册到 PROFIBUS 从站。

注册非循环数据区域

数据区域可以在 PROFIBUS DPV1 从站中进行注册。然后，PROFIBUS DPV1 从站自动处理传递到已注册数据区域的非循环查询（AsyncRead，AsyncWrite）。需要使用插槽和索引来对非循环数据区域进行寻址。

要创建非循环模块，应使用结构 ST_PROFIBUSDPV1AsyncDataModule。必须在此结构中输入插槽和索引，以及在该地址中输入指向数据区域的指针和数据的长度。

确保指针 pbyData 指向在运行时实际存在的存储器区域，且未删除。例如，为此目的，声明数组作为程序中的变量。数据模块结构无需永久可用，因为在注册数据模块时可复制结构的内容。

下文列出了所有相关方法的列表。

数据区域注册的相关方法

下表列出了用于注册 IF_PROFIBUS_DPV1_Slave 接口的数据区域的相关方法：

方法	描述
IsRegisteredDataModule	验证传送的数据模块是否已注册。 返回值为： ● 0：在某些插槽和索引上未注册任何数据模块。 ● 1：在某些插槽和索引上已注册数据模块。 ● 2：插槽无效。 ● 3：索引无效。
RegisterAsyncDataModule	将传送的数据模块注册到插槽和索引。 返回值为： ● 0：已注册数据模块。 ● 1：插槽无效。 ● 2：索引无效。 ● 3：指针指向为 0 的数据区域。 ● 4：长度无效。 ● 5：数据模块已经存在。
UnregisterAllAsyncDataModules	删除所有已注册数据模块的注册。
UnregisterAsyncDataModule	表明取消注册传送数据模块的原因。 返回值为： ● 0：已取消注册数据模块。 ● 1：插槽无效。 ● 2：索引无效。 ● 3：指针指向为 0 的数据区域。 ● 4：数据模块未注册。
GetAsyncDataModule	读取并返回所定义的注册数据模块的数据。 返回值为： ● 0：已成功读取数据模块。 ● 1：插槽无效。 ● 2：索引无效。 ● 3：没有将任何数据模块注册到此插槽和索引。

第10.2节
EtherNet/IP 适配器配置

简介

本节讲述了如何配置 VW3E704100000 通讯模块的 EtherNet/IP 适配器服务。

本节包含了哪些内容？

本节包含了以下主题：

主题	页
EtherNet/IP 适配器配置	106
循环数据交换	109
非循环数据交换	110

EtherNet/IP 适配器配置

简介

本节介绍 EtherNet/IP 适配器服务的配置。

EtherNet/IP 适配器支持以下交换服务：

- 循环数据交换 (参见第 *109* 页)
- 非循环数据交换 (参见第 *110* 页)

有关 EtherNet/IP (CIP) 的详细信息，请访问 www.odva.org 网站。

添加 EtherNet/IP 适配器

要将 EtherNet/IP 适配器添加到控制器，选择**硬件目录**中的 **EtherNet-IP-Adapter**，然后将其拖放到**设备树**，并将其放置在控制器节点上。

注意： 已将 **InputArea** 和 **OutputArea** 节点添加到 EtherNet/IP 适配器节点。这两个 I/O 模块都用于循环通讯。

以太网/IP 适配器配置

要访问 EtherNet/IP 适配器参数，双击**设备树**中的 **EtherNet-IP-Adapter**：

参数	类型	值	缺省值	单位
内部				
InitParameters				
连接器	INT 枚举	前置/1	前置/1	
未使用	BYTE 枚举	NOT_IN_USE	NOT_IN_USE	
IPAddressConfig	BYTE 枚举	手动/0	手动/0	
IPAddress	STRING(15)	'10.128.234.33'	'10.128.234.33'	
IPSubnetMask	STRING(15)	'255.255.240.0'	'255.255.240.0'	
IPGateway	STRING(15)	'10.128.224.10'	'10.128.224.10'	
EthernetAddress	STRING(13)	''	''	
EthernetConfig	BYTE 枚举	自适应/0	自适应/0	
InputLength	UDINT(1..504)	504	504	
OutputLength	UDINT(1..504)	504	504	
WdgEnabled	BOOL	FALSE	FALSE	
WdgTime	UDINT	100	100	
信息				
诊断				
ChannelCommonStatusBlock				
CommCOS	UDINT	2#0000_0000	2#0000_0000	
CommState	UDINT 枚举	未知/0	未知/0	
CommError	UDINT 枚举		成功/0x0	
版本	UINT	0	0	
WdgTime	UINT	0	0	
Reserved0	ARRAY[0..1] OF UINT	[2(0)]	[2(0)]	
WdgTimeHost	UDINT	0	0	
ErrorCount	UDINT	0	0	

黑色参数 读/写访问
灰色参数 只读访问

EtherNet/IP Configuration 选项卡提供了以下参数：

参数	类型	值	缺省值	描述
Internal				
保留				
InitParameters				
IPAddressConfig	BYTE Enum	manual / 0 enable BOOTP / 1 enable DHCP / 2	manual / 0	定义设置模块的 IP 地址的方式： ● 0 = 使用手动输入的 IP 地址。 ● 1 = 通过引导程序协议确定 IP 地址。 ● 2 = 通过 DHCP 协议确定 IP 地址。
IPAddress	STRING	'10.128.234.33'	'10.128.234.33'	指定 EtherNet/IP 适配器的 IP 地址。
IPSubnetMask	STRING	'255.255.240.0'	'255.255.240.0'	指定 EtherNet/IP 适配器的子网掩码。
IPGateway	STRING	'10.128.224.10'	'10.128.224.10'	指定 EtherNet/IP 适配器的网关地址。

参数	类型	值	缺省值	描述
EthernetAddress	STRING	“	“	显示 EtherNet/IP 适配器的 MAC 地址。
EthernetConfig	BYTE Enum	Auto-negotiation / 0 Full Duplex / 100 Mbit/s / 1 Full Duplex / 10 Mbit/s / 2 Half Duplex / 100 Mbit/s / 3 Half Duplex / 10 Mbit/s / 4	Auto-negotiation / 0	显示 EtherNet/IP 适配器的以太网配置： ● 0 = 如果设置此值，则设备与远程集线器或交换机独立协商连接参数。 ● 1 = 设备以 100 MBit/s 在全双工下工作。 ● 2 = 设备以 10 MBit/s 在全双工下工作。 ● 3 = 设备以 100 MBit/s 在半双工下工作。 ● 4 = 设备以 10 MBit/s 在半双工下工作。
InputLength	UDINT	1...504	504	定义循环数据传输的输入数据大小（字节）（**设备树**中的 **InputArea** 模块）。
OutputLength	UDINT	1...504	504	定义循环数据传输的输出数据大小（字节）（**设备树**中的 **OutputArea** 模块）。
WdgEnabled	BOOL	TRUE FALSE	FALSE	启用看门狗： ● TRUE = 已激活看门狗。将参数 WdgTime 的值用作看门狗超时。 ● FALSE = 已停用看门狗。忽略参数 WdgTime 的值。
WdgTime	UDINT	20...65535	100	确定看门狗将从站检测为不可通讯之后的时间。已指定该时间（毫秒）。
Info				
IdentNumber	WORD	0	0	显示 EtherNet/IP 适配器的标识号。
VendorName	STRING	‘Schneider Electric’	‘Schneider Electric’	显示 EtherNet/IP 适配器的供应商名称。
ModelName	STRING	‘EtherNet/IP-Adapter’	‘EtherNet/IP-Adapter’	显示 EtherNet/IP 适配器的型号名称。
DriverInstance	DWORD	0	0	显示此驱动器实例的标识符。
Diag / ChannelCommonStatusBlock				
CommState	UDINT Enum	UNKNOWN / 0 NOT_CONFIGURED / 1 STOP / 2 IDLE / 3 OPERATE / 4	UNKNOWN / 0	显示通讯通道的当前网络状态： ● 0 = 无法确定。 ● 1 = 未配置。 ● 2 = 已停止。 ● 3 = 空闲。 ● 4 = 正常操作。
CommError	UDINT Enum	诊断代码 (参见第 *101* 页)	SUCCESS / 0x0	显示通讯通道的当前诊断代码。
ErrorCount	UDINT	-	0	显示自上一次启动或最后一次重新启动以来检测到的总错误数。

循环数据交换

输入/输出区域的实例 ID

输入和输出区域拥有以下实例 ID：

元素	实例 ID	大小（字节数）	描述
输入汇编 (InputArea)	101	0...504	主控制器输出的命令字 (%QW)
输出汇编 (OutputArea)	100	0...504	主控制器输入的状态 (%IW)

交换的数据量取决于在 Ethernet/IP Configuration (参见第 *107* 页) 中配置的参数 InputLength 和 OutLength。

注意： 输出表示来自主控制器的“输出”（= 模块的 %IW）。
输入表示来自主控制器的“输入”（= 模块的 %QW）。

EtherNet/IP 适配器的 I/O 模块

在添加模块时，已将 **InputArea** 和 **OutputArea** 节点添加到 EtherNet/IP 适配器节点。这两个 I/O 模块都用于循环通讯。

两个已插入 I/O 模块的数据长度相当于循环通讯的最大数据长度（504 字节）。

例如，如果将 InputLength 设置为 50 并将 OutputLength 设置为 20，则 I/O 模块 **InputArea** 中的前 50 个字节和 I/O 模块 **OutputArea** 中的前 20 个字节与扫描器进行循环交换。

I/O 模块中的剩余 454 (504-50) 个输入字节和 484 (504-20) 个输出字节不使用。

EtherNet/IP 模块 I/O 映射

双击**设备树**中的 **InputArea** 或 **OutputArea** 节点。

可以在 **EthernetIP-Modules I/O 映射**选项卡中定义和命名变量。此选项卡还提供其他信息，例如拓扑寻址。

有关详细的一般描述，请参阅“I/O 映射”选项卡描述 *(参见 SoMachine, 编程指南)*。

非循环数据交换

注册回调

使用 EtherNet/IP 适配器，可以在非循环查询的情况下注册回调。如果 EtherNet/IP 适配器驱动器收到扫描器的非循环查询，则首先会将查询传递到已注册的数据区域。

然后，通过使用接口 IF_EIPEventHandler_AsyncGetAttributeAll、IF_EIPEventHandler_AsyncGetAttributeSingle 和 IF_EIPEventHandler_AsyncSetAttributeSingle 调用已注册到 EtherNet/IP 适配器的所有已注册用户功能块 (FB)。同时，也会发送自动查询处理的状态，便于解决可能出现的问题。然后，将当前状态写入参数 iq_udiError。

注意： 必须调用 RegisterAsyncClass() 方法，以便获取回调方法中某些类别的回调。

注意： 如果已注册非循环数据区域，则此方法在内部自动执行。在这种情况下，您无需手动调用 RegisterAsyncClass() 方法。

下文列出了所有相关方法的列表。

注意： 当您添加模块时，已经将 IoDrvEtherNetIPAdapter 库添加到库管理器。该库包含用于管理 EtherNet/IP 模块的接口、方法和功能块。

回调注册的相关方法

下表列出了用于注册 IF_EtherNetIP_Adapter 接口的数据区域的相关方法：

方法	描述
IsRegisteredAsyncGetAttributeAll	在 GetAttributeAll 查询到达后，返回是否已经为回调注册的传送功能块。
IsRegisteredAsyncGetAttributeSingle	在 GetAttributeSingle 查询到达后，返回是否已经为回调注册的传送功能块。
IsRegisteredAsyncSetAttributeSingle	在 SetAttributeSingle 查询到达后，返回是否已经为回调注册的传送功能块。
RegisterAsyncGetAttributeAll	在 GetAttributeAll 查询情况下，为回调注册传送功能块。
RegisterAsyncGetAttributeSingle	在 GetAttributeSingle 查询情况下，为回调注册传送功能块。
RegisterAsyncSetAttributeSingle	在 SetAttributeSingle 查询情况下，为回调注册传送功能块。
UnregisterAsyncGetAttributeAll	在 GetAttributeAll 查询情况下，为回调取消注册传送功能块。
UnregisterAsyncGetAttributeSingle	在 GetAttributeSingle 查询情况下，为回调取消注册传送功能块。
UnregisterAsyncSetAttributeSingle	在 SetAttributeSingle 查询情况下，为回调取消注册传送功能块。
RegisterAsyncClass	为传送 ClassId 启用回调。只有为某些 ClassId 激活回调后，才会调用为回调注册的代码。 返回值为： ● 0：ClassId 已成功注册。 ● 1：ClassId 已注册。 ● ：ClassId 无效。 ● 3：查询 RegisterAsyncClass 请求无法发送。 ● 4：查询 RegisterAsyncClass 请求不正确。

注册非循环数据区域

可以将非循环查询的数据区域注册到 EtherNet/IP 适配器。然后，EtherNet/IP 适配器自动处理已注册数据区域的非循环查询（Get_Attribute_All、Get_Attribute_Single 和 Set_Attribute_Single）。非循环数据区域通过使用 ClassId、InstanceId 和 AttributeId 进行寻址。

为此目的，必须创建类型 ST_EtherNetIPAttribute 的属性。此属性必须包含以下程序内容：ID、长度和指向数据区域的指针（如数组）。

确保属性指针指向的数据区域在运行期间实际存在且未删除。例如，为此目的，声明数组作为程序中的变量。

然后，通过使用接口 IF_EtherNetIPInstance，将属性注册到功能块 FB_EtherNetIPInstance 的实例。

注意： 确保功能块 FB_EtherNetIPInstance 的实例和属性永远可用，且在运行时不会被意外删除（例如，在方法完成后删除方法的本地变量）。

要将实例注册到 EtherNet/IP 适配器，创建的非循环实例必须包含 ClassId 和 InstanceId。

可以使用方法 SetClassId 和 SetInstanceId 设置这些实例。

通过使用 Ethernet/IP 适配器的 IF_EtherNetIP_Adapter 接口注册实例。

下文列出了所有相关方法的列表。

数据区域注册的相关方法

下表列出了注册 IF_EtherNetIP_Adapter 和 IF_EtherNetIPInstance 接口的数据区域的相关方法：

方法	描述
IF_EtherNetIP_Adapter 接口	
RegisterInstance	将实例注册到 EtherNet/IP 适配器。 返回值为： ● 0：实例注册。 ● 1：实例已注册。 ● 2：实例无效。 ● 3：内部错误。
UnregisterAllInstances	取消注册已注册到 EtherNet/IP 适配器的所有实例。 返回值为： ● 0：已删除所有注册。 ● 1：内部错误。 ● 2：没有注册任何实例。 ● 3：内部错误。
UnregisterInstance	取消注册在 EtherNet/IP 适配器中定义的实例。 返回值为： ● 0：注册已失效。 ● CmpErrors.Errors.ERR_NO_OBJECT：找不到最后一个实例。 ● CmpErrors.Errors.ERR_INVALID_HANDLE：内部错误。

方法	描述
IF_EtherNetIPInterface 接口	
SetClassId	将 ClassId 分配给功能块 FB_EtherNetIPInstance 的实例。 返回值为： ● 0：ClassId 有效且可设置。 ● 1：ClassId 不是介于 Gc_EtherNetIPClass_MinID 和 Gc_EtherNetIPClass_MaxID 之间。
SetInstanceId	将 InstanceId 分配给功能块 FB_EtherNetIPInstance 的实例。 返回值为： ● 0：InstanceId 有效且可设置。 ● 1：ClassId 不是介于 0 和 Gc_EtherNetIPClass_MaxID 之间。
AddAttribute	将传送属性添加到功能块 FB_EtherNetIPInstance 的实例。 返回值为： ● 0：属性已添加。 ● 1：AttributeId 已使用。 ● 2：属性无效。
AddAttributeArray	将数组分割成大小为 i_byAttributeLength 的多个部分，然后添加每个部分作为功能块 FB_EtherNetIPInstance 的实例的属性。属性 ID 为连续编号。o_byStartAttributeId 返回第一个添加属性的 AttributeId。最后一个添加属性的长度可能小于传输参数 i_byAttributeLength。 返回值为： ● 0：已添加所有属性，且未检测到任何错误。 ● 1：在实例中没有足够的可用空间。
RemoveAttribute	删除带有所定义 ID 的属性。 返回值为： ● 0：属性已成功删除。 ● CmpErrors.Errors.ERR_INVALIDID：找不到属性。
RemoveAllAttributes	删除实例中的所有属性，并返回已删除属性的数量。
GetClassId	返回 ClassId。
GetInstanceId	返回 InstanceId。
GetAttribute	返回输入/输出参数 iq_stAttribute 中传送属性 ID 的数据。 返回值 CmpErrors.Errors.ERR_INVALIDID：找不到属性。
GetAttributeCount	返回属性的数量。
GetInstanceFreeSize	返回实例的可用空间大小。
IsValid	表示实例信息是否有效。为此目的，必须设置 ClassID 和 InstanceID。
GetFirstAttribute	返回输入/输出参数 iq_stAttribute 中实例的第一个属性的数据。 返回值为： ● 0：已找到属性。 ● 1：找不到任何属性。

方法	描述
GetNextAttribute	返回输入/输出参数 iq_stAttribute 中实例的下一个属性的数据。 返回值为： ● 0：已找到属性。 ● 1：找不到任何属性。

第10.3节
EtherNet/IP 扫描器配置

简介

本节讲述了如何配置 VW3E704100000 以太网通讯模块的 EtherNet/IP 扫描器服务。

本节包含了哪些内容？

本节包含了以下主题：

主题	页
简介	115
支持的设备	116
EtherNet/IP Scanner 配置	118
EtherNet/IP Scanner I/O 映射	120
EtherNet/IP Scanner 状态和诊断	121
目标设备声明	123
目标设置	125
连接配置	127
使用用户参数更换设备	141
EtherNet/IP I/O 映射	144

简介

EtherNet/IP 概述

EtherNet/IP 是为通过标准以太网实施的基于通用工业协议 (CIP) 的协议赋予的名称。有关 CIP 的详细信息，请访问 *www.odva.org* 网站。

EtherNet/IP 采用“起点/目标”架构进行数据交换：

- 起点是在网络上发起同目标设备进行数据交换的设备。这适用于 I/O 通讯和服务消息传递。这相当于 Modbus 网络中客户端的角色。
- 目标是对起点生成的数据请求作出响应的设备。这适用于 I/O 通讯和服务消息传递。这相当于 Modbus 网络中服务器的角色。

EtherNet/IP 起点与目标之间的通讯通过 EtherNet/IP 连接来实现。

EtherNet/IP 扫描器功能

LMC078 Motion Controller 上的 EtherNet/IP Scanner 是起点设备，它连接到目标设备并与目标设备交换配置信息和输入/输出数据。例如，扫描器可以向目标设备指示目标设备传送输入数据的频率以及扫描器预期返回输出数据的频率。

下表是 LMC078 Motion Controller 的 EtherNet/IP Scanner 服务的功能：

功能	描述
性能	最多有 64 个 EtherNet/IP 目标设备可由控制器管理，并在 10 毫秒时间内受到监控
连接数	1...64
输入字数	0...5712。每个目标设备最多 504 个字
输出字数	0...5712。每个目标设备最多 504 个字
I/O 通讯	EtherNet/IP I/O 扫描器服务 用于通讯和数据传输的功能块
	起点/目标
其他服务	EDS 文件管理

支持的设备

支持的设备

下表列出了 EtherNet/IP Scanner 支持的目标设备：

设备名称		TVDA	主要功能
预定义设备	Altivar 32	X	库、预定义连接、预定义数据交换
	Altivar 320	X	库、预定义连接、预定义数据交换
	Altivar 340	X	库、预定义连接、预定义数据交换
	Altivar 6••	X	库、预定义连接、预定义数据交换
	Altivar 71	X	库、预定义连接、预定义数据交换
	Altivar 9••	X	库、预定义连接、预定义数据交换
	Lexium 32 M	X	库、预定义连接、预定义数据交换
	Lexium ILA	X	库、预定义连接、预定义数据交换
	Lexium ILE	X	库、预定义连接、预定义数据交换
	Lexium ILS	X	库、预定义连接、预定义数据交换
	OsiSense XG	X	预定义连接、预定义数据交换
	OsiSense XUW	X	预定义连接、预定义数据交换
	XPSMCM	X	预定义连接、预定义数据交换
其他设备	具有 EDS 文件的设备(1)	-	用户参数、预定义连接
	通用从站设备(2)	-	用户参数
(1) EDS 文件提供预定义连接以促进网络集成，另外还提供其他信息。 (2) SoMachine 中的一般从站设备用于添加没有预定义连接的 EtherNet/IP 设备，如速度驱动器、传感器或其他控制器。			

主要功能

本表介绍了以下主要功能：

主要功能	描述
库	可供应用程序使用的功能和功能块（专用于该设备）。请参阅运动控制库指南 (参见第 *9* 页)。
预定义连接	用于设置循环数据交换。选择一个包含相关信息的建议连接。有关详细信息，请参阅 连接配置 (参见第 *127* 页)。
预定义数据交换	自动设置循环数据交换：添加设备到项目时，会自动选择一个预定义连接。
用户参数	通电时自动发送到设备中的参数。当更换不支持 FDR 的设备时，这些参数将被采用。

TVDA

一些设备随附有应用程序代码模板（称之为设备模块），通过它们可以在 SoMachine 项目中集成变速驱动器或伺服驱动器等设备。设备模块可在功能模板上实现，它在 SoMachine 中是一种用于记忆预定义应用程序内容的机构。

每一个设备模块都嵌有 SoMachine 应用程序内容来控制现场设备，监控其状态并执行错误管理。它包含一种单独的全局变量定义，后者提供了可跨 SoMachine 自动化项目访问设备功能的接口。

关于更多详情，请参阅 TVDA 设备模块库，功能模板库指南 *(参见 TVDA Device Module Library, Function Template Library Guide)*。

EtherNet/IP Scanner 配置

添加 EtherNet/IP Scanner

要将 EtherNet/IP 扫描仪添加到控制器，选择**硬件目录**中的**以太网 IP 扫描仪**，然后将其拖放到**设备树**，并将其放置在控制器节点上。

有关将设备添加到项目的更多信息，请参阅：

- 使用拖放方法 *(参见 SoMachine, 编程指南)*
- 使用上下文菜单或加号按钮 *(参见 SoMachine, 编程指南)*

EtherNet/IP Scanner 配置

要配置 EtherNet/IP Scanner，双击**设备树**中的 **EtherNet IP Scanner** 节点：

以下参数出现在**扫描仪设置**选项卡中：

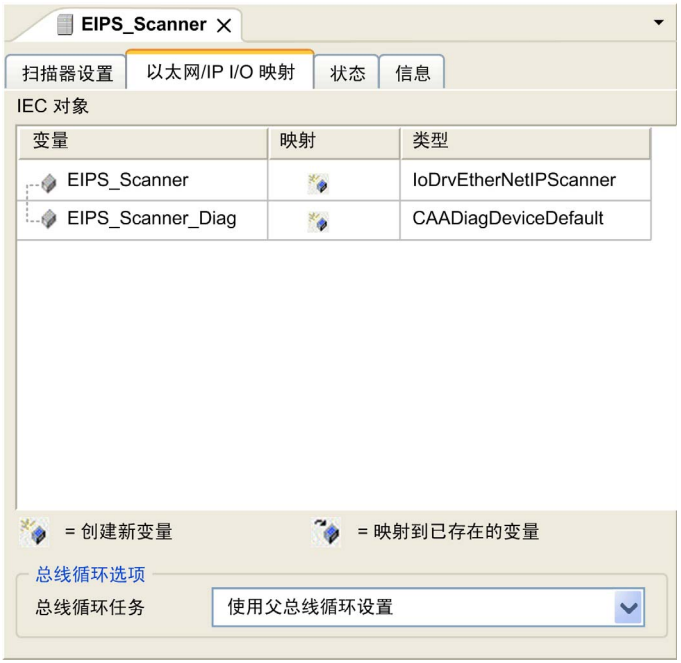
参数	缺省值	描述
IP 地址	192.168.0.1	指定 EtherNet/IP Scanner 的 IP 地址。
子网掩码	255.255.255.0	指定 EtherNet/IP Scanner 的子网掩码。
网关地址	0.0.0.0	指定 EtherNet/IP Scanner 的网关地址。
Speed&Duplex	**自动协商**	选择 EtherNet/IP Scanner 的数据传输方向和速度： **自动协商**。扫描器与远程集线器或交换机独立协商连接参数。 **全双工 /100 Mbit/s**. 扫描器以 100 MBit/s 在全双工下工作。 **全双工 /10 Mbit/s**.扫描器以 10 MBit/s 在全双工下工作。 **半双工 /100 Mbit/s**.扫描器以 100 MBit/s 在半双工下工作。 **半双工 /10 Mbit/s**.扫描器以 10 MBit/s 在半双工下工作。

EtherNet/IP Scanner I/O 映射

配置 EtherNet/IP Scanner I/O 映射

I/O 总线配置让您能够选择用来控制 EtherNet/IP 循环数据交换的任务。

若要配置 EtherNet/IP Scanner I/O 映射，应执行如下操作：

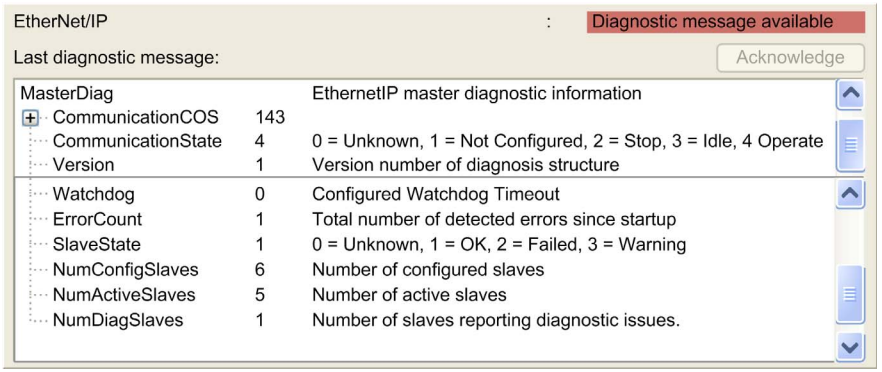
步骤	操作
1	在**设备树**中，双击 **EtherNet IP Scanner**。 **结果：**配置窗口随即显示。
2	选择 **EtherNet/IP I/O Mapping** 选项卡。 
3	**总线循环任务**参数定义 负责 I/O 映像刷新（%QB、 %IB）的任务。这些 I/O 映像与已发送到目标设备的 EtherNet/IP 请求和运行状况位相对应。 选择**总线循环任务**以便使用： ● **使用父总线循环设置**（缺省）。使用在控制器的 PLC 设置 (参见第 *79* 页)中指定的任务。 ● **MAST**. 使用 MAST 任务 (参见第 *47* 页)。 ● **运动**. 使用运动任务 (参见第 *40* 页)。

EtherNet/IP Scanner 状态和诊断

简介

在线模式下，EtherNet/IP Scanner 的**状态**选项卡为 EtherNet/IP Scanner 和连接的设备提供监视和诊断信息。

显示监视和诊断信息

步骤	操作
1	在**设备树**中，双击 **EtherNet IP Scanner**。
2	选择**状态**选项卡。 **结果**：将显示上一次的诊断信息窗口： 

诊断信息

CommunicationCOS 字段为二进制编码十进制值：

位	位设置为 1 时的描述
0	通讯就绪
1	已正确执行配置
2	协议为通讯打开
3	配置已锁定
4	新配置
5	已请求重新开始通讯
6	已启用重新开始通讯
7	已启用直接存储器访问
8...31	未使用

例如，对于 CommunicationCOS 的值 143（操作状态），以下位设置为 1：

- 位 0（通讯就绪）
- 位 1（已正确执行配置）
- 位 2（协议为通讯打开）
- 位 3（配置已锁定）
- 位 7（已启用直接存储器访问）

目标设备声明

概述

本节介绍如何在 EtherNet/IP Scanner 上添加目标设备。

支持的设备 (参见第 *116* 页)中列出了可用的目标设备。

自动设置

在声明每个目标设备时，SoMachine 会自动设置以下各项：

- 按照EtherNet/IP Scanner IP 设置 (参见第 *118* 页)，设置网络设置（IP 地址、子网掩码、网关地址）。
- 对于预定义设备，创建预定义数据交换。

添加目标设备

要在 **Ethernet-IP-Scanner** 节点上添加设备，应在 **硬件目录**中选择设备，将其拖动至**设备树**，然后放置到 **Ethernet-IP-Scanner** 节点上。

有关将设备添加到项目的更多信息，请参阅：

• 使用拖放方法 *(参见 SoMachine, 编程指南)*

• 使用上下文菜单或加号按钮 *(参见 SoMachine, 编程指南)*

根据添加到项目的目标设备，可以自动加载某些库。可用功能块请参见运动控制库指南 (参见第 *9* 页)。

从模板添加目标设备

对于那些不具备主要功能但支持 TVDA 的设备，可以使用模板来声明它们。这样将导入其他元素来促进程序写入。

可将此方法用于 OsiSense XGCS、XUW 和 Preventa XPSMCM 设备。

要从模板添加设备，请执行以下步骤：

步骤	操作
1	在**硬件目录**中，选择**设备模板**复选框。
2	在**硬件目录**中选择设备，将其拖动至**设备树**，然后将其放置到 **Ethernet-IP-Scanner** 节点上。

有关将设备添加到项目的更多信息，请参阅：

• 使用拖放方法 *(参见 SoMachine, 编程指南)*

• 使用上下文菜单或加号按钮 *(参见 SoMachine, 编程指南)*

从 EDS 文件添加目标设备

某些第三方设备在交付时带有 EDS 文件。

要利用 EDS 文件添加设备，请执行以下步骤：

步骤	操作
1	在 SoMachine 菜单中，选择**工具 → 设备库**。
2	单击**安装**以打开**设备描述文件**对话框。
3	在文件类型列表中选择**EDS 和 DCF 文件**。
4	选择 EDS 文件。
5	单击**确定**以关闭对话框。
6	单击**关闭**以关闭**安装设备描述**对话框。
7	选择 **Ethernet-IP-Scanner**，然后单击加号按钮。选择新增的目标设备并单击**添加设备**。关于更多详情，请参阅使用上下文菜单或加号按钮 *(参见 SoMachine, 编程指南)*

目标设置

概述

一旦已将目标设备添加到 EtherNet/IP Scanner 中，便使用设备的**目标设置**选项卡编辑设备的网络设置。

目标设备设置

在**设备树**中，双击 EtherNet/IP 目标设备节点：

地址设置

添加到 EtherNet/IP Scanner 的目标设备必须具有固定 IP 地址。

在 **IP地址**字段中输入设备的 IP 地址。

如果更换设备：

步骤	操作
1	安装新设备.
2	在设备中配置网络设置（IP 地址、子网掩码和网关地址）。
3	直接在新设备中或采用 SoMachine 配置参数。
4	使新设备通电并启动应用程序。

电子键控

电子键控签名用于识别设备。

电子键控是包含在设备固件中的信息（供应商代码、产品代码等）。

当扫描器启动时，它会将每个选中的电子键控值与设备中对应的信息进行对比。

如果设备值与应用程序值不相同，则逻辑控制器将不再与该设备进行通讯。这可在应用范围内通过诊断信息 (参见第 *121* 页)进行监视，以便在您的机器环境范围内采取适当行动。

对于预先配置的设备，您无法修改**电子键控**值。

对于通用 EtherNet/IP 设备，您可以修改**电子键控**值。

对于**电子键控**值，请参阅设备文档中的标识对象 (F1 hex) 说明。

连接配置

连接概述

如要访问 EtherNet/IP 设备，必须配置连接。通过连接，可以交换组合成汇编的数据块。

连接的启动和停止由控制器管理。

汇编

I/O 数据和配置数据可以合并为汇编对象。

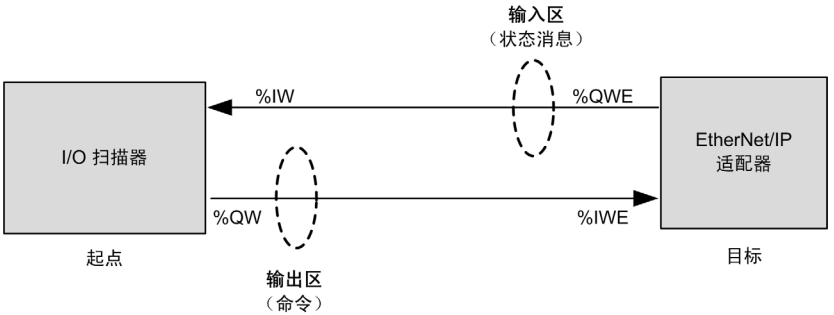
来自不同对象的数据（属性）可以合并成一个单独对象，从而使数据能够在单个连接上进行发送或接收。

汇编对象实例用于为与 I/O 连接有关的输入数据和输出数据汇总数据。

汇编对象在结构上可分为类、实例和属性：

- 类是表现同种系统组件的一组对象。
- 对象实例是某一类范围之内某一特定对象的表现。每个实例都有其自己的一组属性值。
- 属性即对象和/或对象类的特性。通常，属性会提供状态信息或定义对象的操作。

下图显示了 EtherNet/IP 通讯过程中输入汇编和输出汇编的方向性：



EtherNet/IP 配置参数定义如下：

- **实例**：参考汇编的编号。
- **大小**：汇编通道的数量。
  每个通道的内存大小是 2 字节，用于存储 %IBx 或 %QBx 对象的值，其中 x 是通道编号。

例如，如果**输出汇编**的**大小**为 20，这表示有 20 个输入通道（IB0 至 IB19）用于寻址 %IBy…%IB(y+20-1)，其中 y 是该汇编的第一个可用通道。

配置设备连接

如要创建和配置连接，则双击**设备树**中的 EtherNet/IP 目标设备，然后选择**连接**选项卡：

列	注释
连接 N°	连接编号是唯一的。它由 SoMachine 自动分配。
连接名称	缺省连接名称由 SoMachine 自动生成。对于预定义连接 (参见第 *116* 页)，这个名称无法编辑。对于其他连接，缺省连接名称可在编辑连接窗口 (参见第 *137* 页)中编辑。
RPI O --> T（毫秒）	请求包时间间隔：EtherNet/IP Scanner (O --> T) 或目标设备 (T --> O) 请求的循环数据传输之间间隔的时间周期。
RPI T --> O（毫秒）	
O-->T 大小（字节）	将在起点 (O) 和目标 (T) 之间交换的字节的数量。
T-->O 大小（字节）	
Config#1 大小（字节）	第一组配置参数的大小。仅为参数可配置的设备显示。
Config#2 大小（字节）	第二组配置参数的大小。仅为参数可配置的设备显示。
连接路径	其他连接参数的编号转录。

若要创建连接，应单击**添加连接**。参见随后的“添加 EtherNet/IP 连接”。

若要修改连接，应选择连接并单击**编辑连接**，或者双击该连接。

若要删除连接，应选择连接并单击**删除连接**。

添加 EtherNet/IP 连接

如要创建并配置连接，请执行以下步骤：

步骤	操作
1	在**设备树**中，双击 EtherNet/IP 目标设备。
2	选择**连接**选项卡。
3	单击**添加连接**。
4	选择**通用连接（可自由配置）**：

(1) “类 ID”、“实例 ID”和“属性 ID”可在设备文档中找到。

步骤	操作
5	选择**自动产生路径**并配置**组合配置**： ● **类 ID**（缺省值为 4）：类标示符[1] ● **实例 ID**：实例标示符[1] ● **属性 ID**（缺省值为 3）：属性标示符[1]
6	配置**组合消耗 (O --> T)**： ● **类 ID**（缺省值为 4）：类标示符[1] ● **实例 ID**：实例标示符[1] ● **属性 ID**（缺省值为 3）：属性标示符[1]
7	配置**组合生产 (T --> O)**： ● **类 ID**（缺省值为 4）：类标示符[1] ● **实例 ID**：实例标示符[1] ● **属性 ID**（缺省值为 3）：属性标示符[1]
8	配置**通用参数**： ● **连接名称**。 缺省连接名称由 SoMachine 自动生成。对于普通连接，可以编辑缺省名称“普通连接”。 ● **连接路径**. 物理链路对象的编号转录对于普通连接，是可以编辑的。 ● **触发类型**。选择发起数据交换的方式： ❍ **循环：**端点以预定时间间隔定期交换数据。 ❍ **状态改变**：端点仅在数据改变时交换数据。在没有发生改变的情况下，如要防止连接超时，也会以后台循环间隔来交换数据（参见下面的 RPI）。 在选择了**状态改变**的情况下，扫描器至目标以及目标至扫描器连接属性的**禁止时间 (ms)** 会启用。 ❍ **应用程序**。数据交换由应用程序触发。 ● **传输类型**： ❍ **专有所有者**：此为通向输出连接点（通常是汇编对象）的双向连接，该汇编的数据在其中只能由一个扫描器控制。可能存在通向输入汇编的连接；该数据正被发送至扫描器。 ❍ **只监听**：扫描器会从目标设备收到输入数据并向目标设备产生 Heartbeat。不存在输出数据。**只监听**连接只能连接到已有的**专有所有者**或**只输入**连接上。如果此底层连接停止，则**只监听**连接也会停止或超时。 ❍ **只输入**：扫描器会从目标设备收到输入数据并向目标设备产生 Heartbeat。不存在输出数据。 ● **RPI (ms)** 请求包时间间隔。扫描器请求的循环数据传输之间的时间周期。目标设备始终会提供一个最小 RPI，但在控制器中，目标是具有最高 RPI，使得系统不会过载。每次将设备添加到 EtherNet/IP 现场总线时，或每次修改 RPI 值时，确认目标设备所使用的控制器资源的级别。设备 RPI 可以在设备文档中指定，但通常作为 EDS 文件的一部分随附于设备。 ● 选择**超时倍增**：4（缺省值）/8/16/32/64/128/256/512。 将所选择的值乘以 RPI 值，就得到连接超时值。
[1] “类 ID”、“实例 ID”和“属性 ID”可在设备文档中找到。	

步骤	操作
9	配置**扫描器至目标（输出）**特定的参数： ● **O --> T 大小（字节）**：待传输的字节数量。 ● **Config#1 大小（字节）**：第一组配置参数中的参数数量。 ● **Config#2 大小（字节）**：第二组配置参数中的参数数量。 ● **连接类型**。 可使用的连接类型： ❍ **多播**。在扫描器与多个目标设备之间建立连接。 ❍ **点到点**。在扫描器与单个目标设备之间建立连接。 ● **固定/可变**。 所传输的数据量始终相同（**固定**）还是仅传输一定量的缓冲数据（**可变**）。 ● **传输格式**。连接所适用的实时数据格式： ❍ **32 位运行/空闲**。 32 位包头包括运行/空闲通知。 ❍ **纯数据**。 没有运行/空闲通知。 ❍ **Heartbeat**. 没有运行/空闲通知。 ❍ **以零长度空闲**。零长度数据格式指示空闲。 相关详情请参见*OVDA website* ● **禁止时间 (ms)**：2 次数据交换之间的最短时间周期。只有当**触发类型**为**状态改变**时才可访问。该值必须是 2 ms 的倍数。最大值为 **RPI (ms)** 值，最大可能值为 254 ms。
10	配置**目标至扫描器（输入）**特定的参数： ● **T --> O 大小（字节）**：待传输的字节数量。 ● **连接类型**。可使用的连接类型： ❍ **多播**。在目标与多个扫描器之间建立连接。 ❍ **点到点**。在目标与单个扫描器之间建立连接。 ● **固定/可变**。 所传输的数据量始终相同（**固定**）还是仅传输一定量的缓冲数据（**可变**）。 ● **消费汇编 (O --> T)** ● **禁止时间 (ms)**：2 次数据交换之间的最短时间周期。当**触发类型**为**状态的更改**时即可访问。该值必须是 2 ms 的倍数。最大值为 **RPI (ms)** 值，最大可能值为 254 ms。
11	单击**确定**。
(1) “类 ID”、“实例 ID”和“属性 ID”可在设备文档中找到。	

关于受支持汇编的更多详情，请参阅设备的文档。

注意： 考虑到 **O --> T 大小（字节）**和 **T --> O 大小（字节）** 限制因素和扫描器输入/输出字的数量上限 (参见第 *115* 页)，在添加了连接之后，请验证扫描器资源负载。

添加预定义连接

预定义连接（请参见支持的设备 (参见第 *115* 页)）可用于：

- 预定义设备
- 随附有 EDS 文件的其他设备。

根据定义，通用设备并不具备预定义连接。

若要添加预定义连接，应执行以下操作：

步骤	操作
1	在**设备树**中，双击 EtherNet/IP 目标设备。
2	选择**连接**选项卡。
3	单击**添加连接**。
4	选择**预定义连接 (EDS-File)**：

步骤	操作
5	选择某一预定义连接： 新连接 通用连接（可自由配置） 预定义连接（EDS 文件） 选择连接 连接名称 | RPI（毫秒） | O-> T 大小（字节） | T-> O 大小（字节） | 配置 #1 大小（字节） 20-70 专有所有者 | 10 | 4 | 4 20-70 只输入 | 10 | 4 | 4 20-70 只监听 | 10 | 4 | 4 21-71 专有所有者 | 10 | 4 | 4 21-71 只输入 | 10 | 4 | 4 21-71 只监听 | 10 | 4 | 4 22-72 专有所有者 | 10 | 6 | 6 一般参数 连接名称：20-70 只输入 连接路径：20 04 24 06 2C FF 2C 46 触发类型：循环　RPI（毫秒）：16 传输类型：只输入　超时倍增：4 扫描器至目标（输出） O → T 大小（字节）：4 配置 #1 大小（字节）：0 配置 #2 大小（字节）：0 连接类型：点到点 固定/可变：固定 传输格式：32 位运行/空闲 禁止时间 (ms)：0 目标至扫描器（输入） T → O 大小（字节）：4 连接类型：多播 固定/可变：固定 传输格式：纯数据 禁止时间 (ms)：0 确定　取消
6	配置**通用参数**： ● **RPI (ms)** 请求包时间间隔。扫描器请求的循环数据传输之间的时间周期。EDS 中对缺省值进行了定义。 ● 选择**超时倍增**：4（缺省值）/8/16/32/64/128/256/512。将所选择的值乘以 RPI 值，就得到连接超时。
7	配置**扫描器至目标（输出）**： ● **O --> T 大小（字节）**：待传输的字节数量。

步骤	操作
8	配置**目标至扫描器（输入）**： ● **T --> O 大小（字节）**：待传输的字节数量（汇编通道数量） ● 选择**连接类型**：**广播**（缺省值）（如果连接位于扫描器与多个目标设备之间），或者**点到点**（如果连接位于扫描器与单个目标设备之间）。仅对于某些**传输类型**而言是可编辑的。
9	单击**确定**。 **结果**：连接已添加到**连接**选项卡。

配置配置汇编

某些设备支持配置汇编。

配置汇编是单一的请求，在 EtherNet/IP Scanner 启动时发送，它将所有配置参数发送至目标设备。

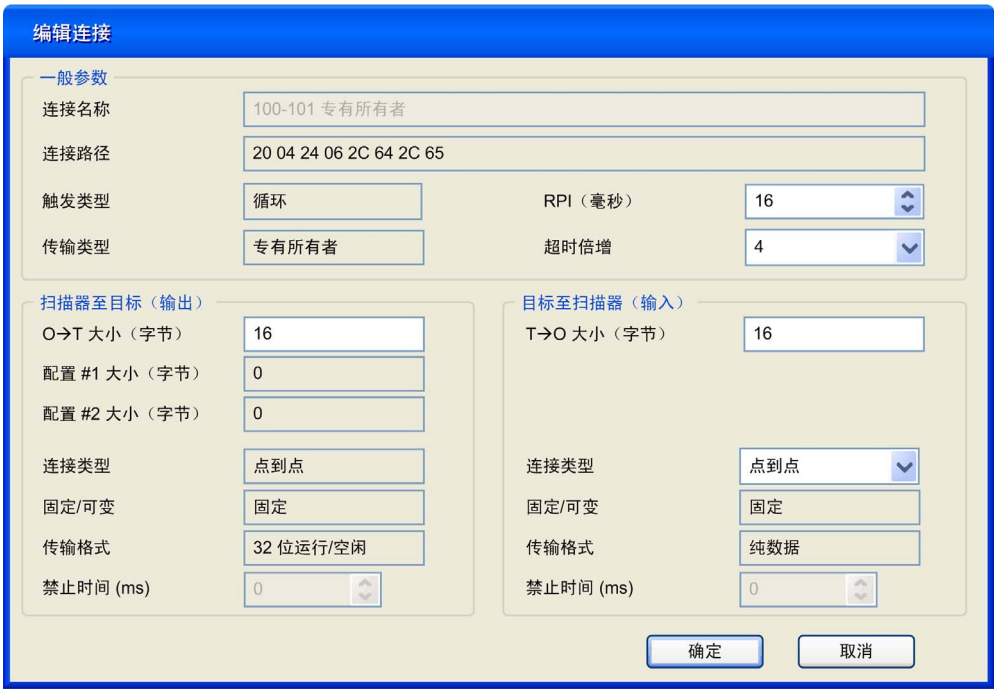
若要配置配置汇编，应执行以下操作：

步骤	操作
1	在**设备树**中，双击 EtherNet/IP 设备。
2	选择**连接**选项卡。
3	选择某一现有的连接，并单击**编辑连接**。
4	选择**通用连接（可自由配置）**。
5	选择**配置汇编**。
6	配置**配置汇编**： ● **类 ID**（缺省值为 4）：类标示符(1) ● **实例 ID**：实例标示符(1) ● **属性 ID**（缺省值为 3）：属性标示符(1)
7	配置**扫描器至目标（输出）**： ● **Config#1 大小（字节）**：第一组配置参数中的参数数量。 ● **Config#2 大小（字节）**：第二组配置参数中的参数数量。

步骤	操作
8	单击**确定**。 **结果：**配置参数显示在**连接**选项卡中：
9	双击**值**列以设置配置参数值。
(1) “类 ID”、“实例 ID”和“属性 ID”可在设备文档中找到。参见如何查找汇编信息 (参见第 *140* 页)。	

编辑预定义连接

如要编辑预定义连接，请选择 **连接**选项卡中的连接，并单击**编辑连接**：

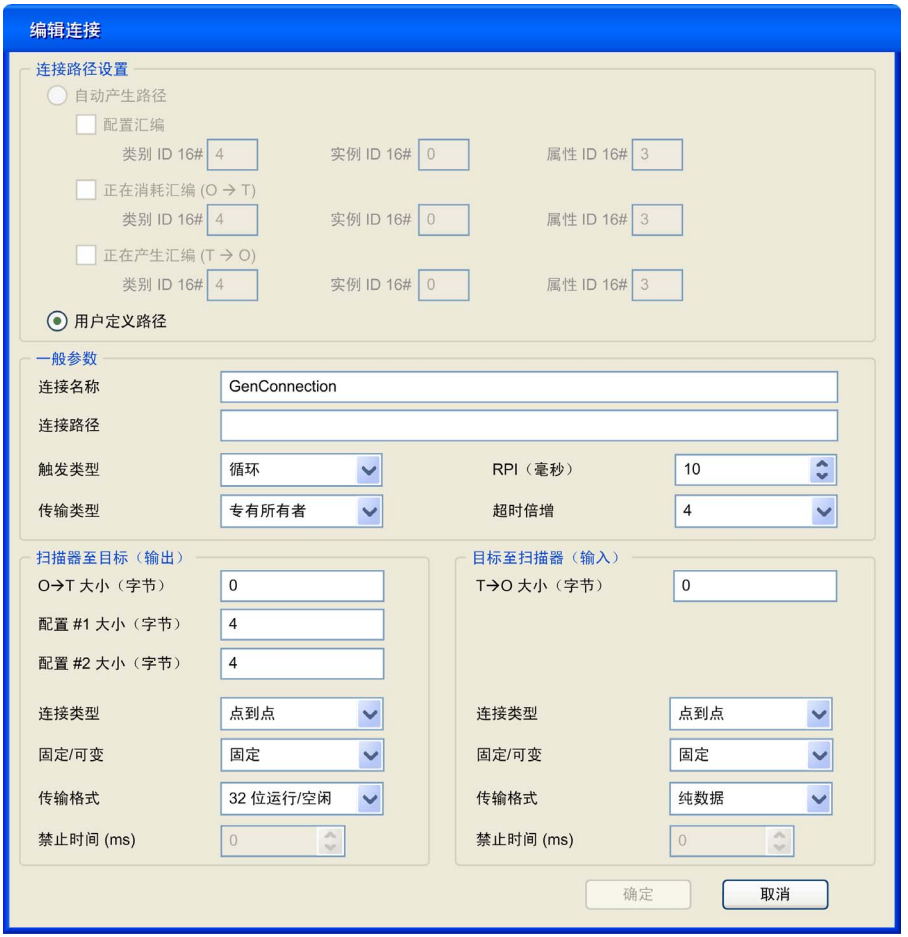


参数	描述
通用参数	
RPI（毫秒）	**RPI (ms)** 请求包时间间隔。扫描器请求的循环数据传输之间的时间周期。设备始终会提供一个最小 RPI，但在控制器中，目标是具有最高 RPI，使得系统不会过载。每次将设备添加到 EtherNet/IP 现场总线时，或每次修改 RPI 值时，确认目标设备所使用资源。设备 RPI 可以在设备文档中指定，但通常作为 EDS 文件的一部分随附于设备。
超时倍增	选择**超时倍增**：4（缺省值）/8/16/32/64/128/256/512。将所选择的值乘以 RPI 值，就得到连接超时值。
扫描器至目标（输出）	
O --> T 大小（字节）	汇编的通道大小。 每个通道的内存大小为 2 个字节，用于存储 %IWx 或 %QWx 对象的值，其中 x 是通道号。
目标至扫描器（输入）	
T --> O 大小（字节）	**T --> O 大小（字节）**：待传输的字节数量（汇编通道数量） 每个通道的内存大小为 2 个字节，用于存储 %IWx 或 %QWx 对象的值，其中 x 是通道号。

参数	描述
连接类型	请求的连接类型： ● **广播**（缺省值）（如果连接位于扫描器与多个目标设备之间） ● **点到点**（如果连接位于扫描器与单个目标设备之间） 仅对于某些**传输类型**而言是可编辑的。
禁止时间 (ms)	2 次数据交换之间的最短周期时间。 当**触发类型**为**状态的更改**时即可访问。禁止时间最大值为 RPI，以 254 ms 为限。

编辑普通连接

如要编辑普通连接，请选择 **连接**选项卡中的连接，并单击**编辑连接**：



继续执行下列步骤：

参数			值	描述
连接路径设置				
自动生成路径			是/否	使您能够配置这些汇编的参数。
	配置汇编		真/假	使您能够配置配置汇编 (参见第 *140* 页)。
		类 ID	2 个字节（缺省值为 04h）	类标示符[1]
		实例 ID	2 个字节（缺省值为 0）	实例标示符[1]
		属性 ID	2 个字节（缺省值为 03h）	属性标示符[1]
	消费汇编 (O --> T)			
		类 ID	2 个字节（缺省值为 04h）	类标示符[1]
		实例 ID	2 个字节（缺省值为 0）	实例标示符[1]
		属性 ID	2 个字节（缺省值为 03h）	属性标示符[1]
	生产汇编 (T --> O)			
		类 ID	2 个字节（缺省值为 04h）	类标示符[1]
		实例 ID	2 个字节（缺省值为 0）	实例标示符[1]
		属性 ID	2 个字节（缺省值为 03h）	属性标示符[1]
自定义路径			是/否	禁用**自动生成路径**区并启用**连接路径**字段。
通用参数				
连接名称			文本字符串	输入普通连接的名称。缺省值为**普通连接**。
连接路径			字节的数组	在选择了**自动产生路径**的情况下的物理链路对象的编号转录。 在选择了**自定义路径**的情况下，是可以编辑的。
触发类型			● **循环**（缺省） ● **状态改变** ● **应用程序**	选择发起数据交换的方式： ● **循环：**端点以预定时间间隔定期交换数据。 ● **状态改变**：端点仅在数据改变时交换数据。在没有发生改变的情况下，如要防止连接超时，也会以后台循环间隔来交换数据（参见下面的 RPI）。 在选择了**状态改变**的情况下，扫描器至目标以及目标至扫描器连接属性的**禁止时间 (ms)** 会启用。 ● **应用程序**。数据交换由应用程序触发。
[1] “类 ID”、“实例 ID”和“属性 ID”可在设备文档中找到。请参阅 如何查找汇编信息 (参见第 *140* 页)。				

参数	值	描述
传输类型	● **专有所有者**（缺省值） ● **冗余所有者**	**专有所有者**：此为通向输出连接点（通常是汇编对象）的双向连接，该汇编的数据在其中只能由一个扫描器控制。可能存在通向输入汇编的连接；该数据正被发送至扫描器。如果输入数据长度为零，则该方向将成为 Heartbeat 连接。 **冗余所有者**。 允许多个独立的起点应用程序各自建立到目标设备传输端的独立且相同的连接。
RPI（毫秒）	以毫秒为单位（缺省值为 10 毫秒）	请求的数据包时间间隔。扫描器请求的循环数据传输之间的时间周期。 设备文档可能指定了设备 RPI。但是，此信息通常是作为随设备交付的 EDS 文件的一部分提供。
超时倍增	4（缺省值） /8/16/32/64/128/256/512	通过将 RPI 乘以超时倍增值来按连接管理扫描器超时。
扫描器至目标（输出）		
O --> T 大小（字节）	0 至 XX => 视具体设备而定	汇编的通道大小。 每个通道的内存大小是 2 字节，用于存储 %IWx 或 %QWx 对象的值，其中 x 是通道编号。
Config#1 大小（字节）	0 至 XX => 视具体设备而定	当连接路径包含了配置汇编时即可访问。 要传输的参数数量（1 个字节）。 配置值将在扫描器启动时发送至设备中。
Config#2 大小（字节）	0 至 XX => 视具体设备而定	
连接类型	点到点	请求的连接类型
固定/可变	固定	请求长度是固定的。
传输格式	● 32 位运行-空闲（缺省值） ● 纯数据 ● Heartbeat	请求的传输格式。有关详细信息，请参阅 *ODVA website*。
禁止时间	0 毫秒	2 次数据交换之间的最短周期时间。
目标至扫描器（输入）		
T --> O 大小（字节）	0 至 XX => 视具体设备而定	汇编的通道大小。 每个通道的内存大小为 2 个字节，用于存储 %IWx 或 %QWx 对象的值，其中 x 是通道号。
连接类型	● 多点传送（缺省值） ● 点到点	请求的连接类型
固定/可变	固定	请求长度是固定的。
传输格式	● 32 位运行空闲 ● 纯数据（缺省值） ● Heartbeat ● 以零长度空闲	请求的传输格式。有关详细信息，请参阅 *ODVA website*。
(1) “类 ID”、“实例 ID”和“属性 ID”可在设备文档中找到。请参阅 如何查找汇编信息 (参见第 *140* 页)。		

参数	值	描述
禁止时间 (ms)	2 ms 的倍数（缺省值为 2 毫秒）	2 次数据交换之间的最短周期时间。 当**触发类型**为**状态的更改**时即可访问。禁止时间最大值为 RPI，以 254 ms 为限。
(1) “类 ID”、“实例 ID”和“属性 ID”可在设备文档中找到。请参阅 如何查找汇编信息 (参见第 *140* 页)。		

如何查找汇编信息

汇编信息在设备文档中提供。它通常是汇编对象说明的一部分。

若要配置汇编，请确定下列信息：

1. 类 ID
   “汇编对象”类 ID 包含在设备文档中，通常等于 04h。
2. 实例 ID
   选择汇编实例，具体取决于应用程序和设备类型。汇编实例的选择会在设备中产生一个专用状态机：
   - **配置汇编**：很少设备支持；通过设备文档验证它支持何种汇编实例。
   - **消费汇编**：在设备文档中有时称为“设备输出”（从设备角度看）。
   - **生产汇编**：在设备文档中有时称为“设备输入”（从设备角度看）。
3. 属性 ID
   搜索要读取的属性。这对应于连接期间交换的数据缓冲。
   “属性”属性对生产汇编必须有写访问权限，对“消费汇编”必须有读访问权限。
   这两个汇编的属性 ID 相同。其值包含在设备文档中，通常等于 03h。它与访问权限为 Get/Set 的属性匹配。名称常常是“数据”，数据类型常常是“字节数组”。

使用用户参数更换设备

概述

您可以在扫描器连接在下列步骤完成之后启动之前配置发送至设备的**用户参数**以加快设备更换：

- 应用程序下载
- 复位热/冷启动
- 手动启动连接

某些 EtherNet/IP 设备具有预定义的**用户参数**。

此外，**用户参数**选项卡还允许您添加和管理其他参数。

关于维护详情，请参阅应用正确的设备配置。

用户参数

在**设备树**中，双击 EtherNet/IP 设备并选择**用户参数**选项卡：

概览 | 目标设置 | 连接 | 用户参数 | 以太网/IP I/O 映射 | 状态 | 信息

新建... 删除... 编辑...

行	名称	类...	实例...	属性...	值	位长度	如果有错误，则中断	如果有错误，则跳行	下一行	注释
1	配置文件	8B	1	2	2	16	☐	☐	0	
2	参考 1 通道	8B	1	E	169	16	☐	☐	0	
3	命令通道 1	8B	1	18	30	16	☐	☐	0	
4	命令切换	8B	1	16	98	16	☐	☐	0	
5	参考 2 切换	8B	1	C	222	16	☐	☐	0	
6	OMA3	AE	1	4	3204	16	☐	☐	0	

列	描述
行	行号。 指示加载到设备中的参数的顺序。
名称	参数名称。
类	该对象对应的类的类 ID(1)。
实例	该对象对应的实例的实例 ID(1)。
属性	该对象对应的属性的属性 ID(1)。
值	参数的值。 双击该值即可对其进行修改。如果适用，包含了可能值的列表将会打开。
位长度	参数位的数量。 依据所选的参数数据类型自动更改。
如果有错误则中断	如果被选中，当检出错误时，参数的传输将中止。
(1) “类 ID”、“实例 ID”和“属性 ID”可在设备文档中找到。请参阅 如何查找用户参数信息 (参见第 *143* 页)。	

列	描述
如果有错误则跳行	如果被选中，当检出错误时，程序会从**下一行**列中规定的行继续执行。因此，可以在初始化时跳过一个功能块，或者定义一次返回。 **注意：** 如果无法写入某个参数，则返回可能会导致无限循环。
下一行	双击以进入将跳转前往的行（如果**如果有错误则跳行**已被选中）。
注释	双击可输入注释。
(1) “类 ID”、“实例 ID”和“属性 ID”可在设备文档中找到。请参阅 如何查找用户参数信息 (参见第 *143* 页)。	

图标	描述
向上移动	在参数列表中将选中的参数上移。
向下移动	在参数列表中将选中的参数下移。
新建	创建新参数。
删除	删除选中的参数.
编辑	编辑选中的参数。

创建或配置用户参数

单击**新建**或选择某一参数并单击**编辑**：

字段	描述
名称	参数名称。
类	该对象类型对应的类的类 ID[(1)]。
实例	一个类实现对应的实例的实例 ID[(1)]。
属性	一个实例特性对应的属性的属性 ID[(1)]。
数据类型	包含可能数据类型的列表。
位长度	参数位的数量。 依据选中的**数据类型**自动更改。
值	参数的值。
[(1)] “类 ID”、“实例 ID”和“属性 ID”可在设备文档中找到。参见如何查找用户参数信息 (参见第 *143* 页)。	

如何查找用户参数信息

可配置的用户参数信息在设备文档中提供。它通常是应用程序对象、显式消息或属于 EtherNet/IP 类别 3 的对象的说明的一部分。

对于用户参数所属的类和/或实例，通常会指定用户参数写访问权限。写操作通常利用称为 `Set_Attribute_Single` 或 `Write one attribute` 的服务来调用。或者，它可能支持服务标示符 0x10（十六进制）或 16（十进制）。

用户参数总是具有如下数值属性：

- 类或类 ID，通常表示为十六进制值
- 实例或实例 ID，通常表示为十六进制值
- 属性或属性 ID，通常表示为十六进制值

用户参数可能还有标示符，其形式为三个十进制数 (xx/yy/zz) 或三个十六进制数 (16#xx/yy/zz)

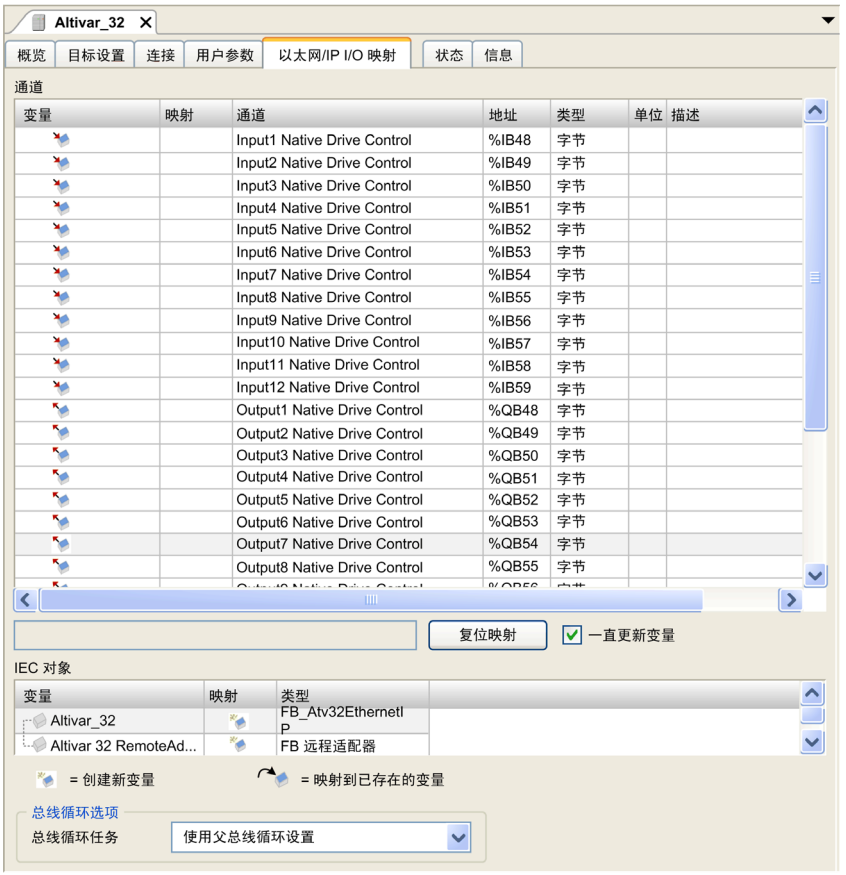
EtherNet/IP I/O 映射

概述

一旦配置了数据交换，您便可以映射将由应用程序使用的各个变量。

配置 EtherNet/IP 目标设备 I/O 映射

若要配置 EtherNet/IP 目标设备 I/O 映射，应执行如下操作：

步骤	操作
1	在**设备树**中，双击 EtherNet/IP 目标设备。 **结果**：其配置窗口随即显示。
2	选择 **EtherNet/IPI/O 映射**选项卡。 

步骤	操作
3	双击**变量**列的单元格，以打开文本字段。 输入变量的名称或单击浏览 **[...]** 按钮，然后使用**输入助手**选择变量。
4	**IEC 对象**。 选项卡的这个部分列出了应用程序可以访问的目标设备的 IEC 对象（比如，为了重启总线或轮询信息）。
5	**总线循环任务**参数定义 负责 I/O 映像刷新（%IB 和 %QB）的任务。这些 I/O 映像与已发送到 EtherNet/IP 目标设备的 EtherNet/IP 请求和运行状况位相对应。 选择列表中的**总线循环任务**： ● **使用父总线循环设置**（缺省）。使用在控制器的 PLC 设置 (参见第 *79* 页)中指定的任务。 ● **MAST**. 使用 MAST 任务 (参见第 *47* 页)。 ● **运动**. 使用运动任务 (参见第 *40* 页)。

第11章
以太网配置

简介

本章介绍如何配置 Modicon LMC078 Motion Controller 的以太网网络接口。

本章包含了哪些内容？

本章包含了以下部分：

节	主题	页
11.1	以太网服务	148
11.2	防火墙配置	166

第11.1节
以太网服务

本节包含了哪些内容？

本节包含了以下主题：

主题	页
简介	149
IP 地址配置	151
Modbus TCP 服务器/客户端	156
FTP 服务器	158
FTP 客户端	160
LMC078 Motion Controller 用作 Modbus TCP 上的 IOScanner 从站设备	161

简介

以太网服务

控制器支持以下服务：

- Modbus TCP 服务器 (参见第 *156* 页)
- Modbus TCP 客户端 (参见第 *156* 页)
- Web 可视化 (参见第 *150* 页)
- FTP 服务器 (参见第 *158* 页)
- FTP 客户端 (参见第 *160* 页)
- 控制器用作 Modbus TCP 上的 IOScanner 从站设备 (参见第 *161* 页)
- IEC VAR ACCESS (参见第 *150* 页)

以太网协议

控制器支持以下协议：

- SoMachine 协议
- IP (Internet Protocol)
- UDP (User Datagram Protocol)
- TCP (Transmission Control Protocol)
- ARP (Address Resolution Protocol)
- ICMP (Internet Control Messaging Protocol)
- IGMP (Internet Group Management Protocol)

连接

下表显示了最大连接数：

连接类型	最大连接数
Modbus 服务器	8
Modbus 客户端	2
以太网/IP 目标	64
FTP 服务器	4
Web 可视化	10

每个基于 TCP 的服务器都管理其各自的一组连接。

如果客户端尝试打开的连接超过了池大小，则控制器会关闭最先打开的连接。

如果客户端尝试打开新连接时所有连接都忙（正在进行数据交换），新连接将被拒绝。

只要控制器继续处于工作状态（RUN、STOP、HALT），所有服务器连接都将保持打开。

退出或进入工作状态（RUN、STOP、HALT）时，所有服务器连接都将关闭，但断电情况除外（因为控制器来不及关闭连接）。

可用服务

借助以太网通讯，控制器可以支持 IEC VAR ACCESS 服务。借助 IEC VAR ACCESS 服务，可以实现控制器和 HMI 之间的数据交换。

控制器也支持**网络变量**服务。通过**网络变量**服务，可以实现控制器之间的数据交换。

注意： 有关详细信息，请参阅《SoMachine - 编程指南》。

Web 可视化

有关 Web 可视化功能的详细信息，请参阅 SoMachine 在线帮助中的*使用 SoMachine/可视化编程*一章。

IP 地址配置

简介

可通过以下各种方法分配控制器的 IP 地址：
- 由 DHCP 服务器分配地址
- 由 BOOTP 服务器分配地址
- 固定 IP 地址

IP 地址可以动态更改：
- 通过 SoMachine 中的控制器选择 *(参见 SoMachine, 编程指南)*选项卡。
- 通过 **changeIPAddress** 功能块 (参见第 *227* 页)。

注意： 如果尝试的寻址方法不成功，控制器将开始使用缺省 IP 地址 (参见第 *154* 页)。

认真管理 IP 地址，因为网络上的每个设备都需要唯一的地址。使多个设备具有相同的 IP 地址会导致网络和相关设备的意外操作。

⚠ 警告

意外的设备操作

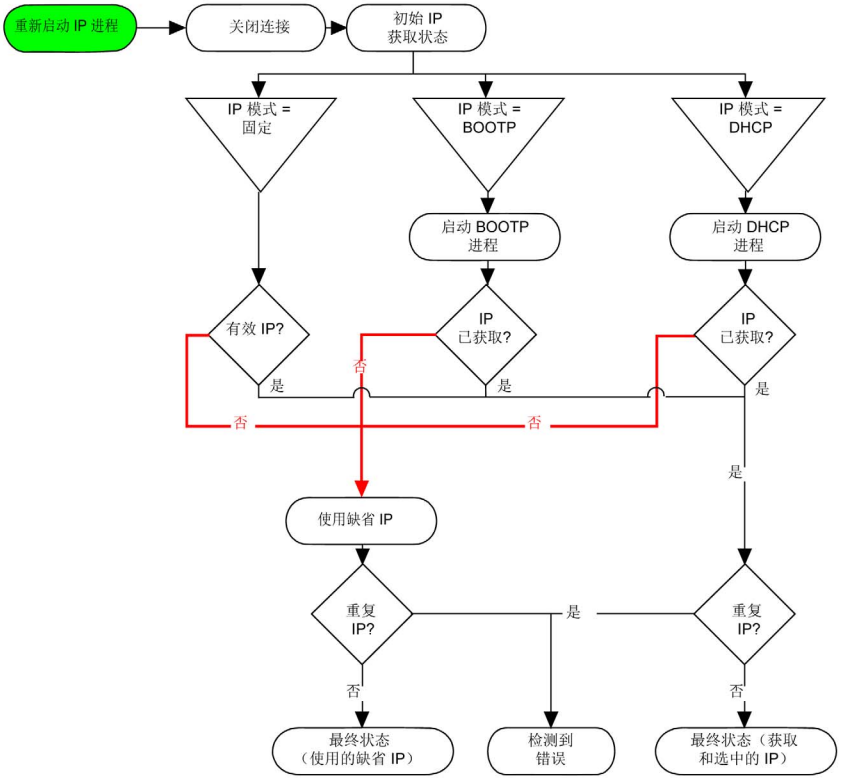
- 确认在网络或远程链路上仅配置了一个主控制器。
- 确认所有设备均有唯一的地址。
- 从系统管理员处获取 IP 地址。
- 在将系统投入使用之前，请确认设备的 IP 地址是唯一的。
- 请不要将同一个 IP 地址分配给网络上的任何其他设备。
- 在克隆包括以太网通讯的任何应用程序后将 IP 地址更新为唯一的地址。

不遵循上述说明可能导致人员伤亡或设备损坏。

注意： 确认系统管理员对网络和子网上所有已分配的 IP 地址留有记录，并向系统管理员通知已执行的所有配置更改。

地址管理

下图显示了控制器的不同地址系统的类型：



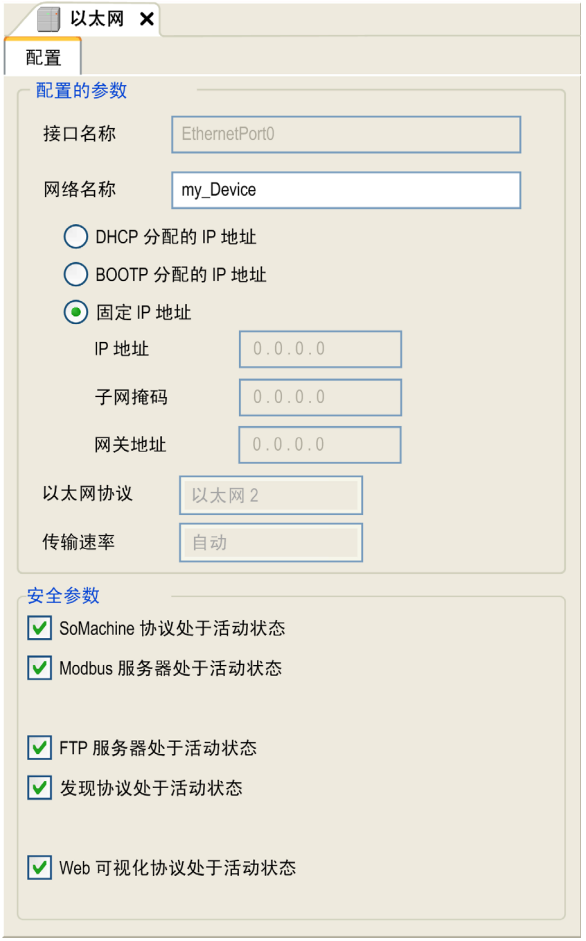
注意： 如果编程为使用 DHCP 或 BOOTP 寻址方法的设备无法联系其相应的服务器，则控制器使用缺省 IP 地址。但是，它将不断地重复此请求。

在以下情况中，IP 进程会重新启动：

- 控制器重新启动
- 以太网电缆重新连接
- 下载应用程序（如果 IP 参数发生更改）
- 在前一次寻址尝试不成功后，检测到 DHCP 或 BOOTP 服务器。

以太网配置

在**设备树**中，双击**以太网**：



配置的参数如下所述：

配置参数	描述
接口名称	网络链路的名称。
网络名称	作为设备名，用于通过 DHCP 检索 IP 地址，最多包含 16 个字符。
DHCP 分配的 IP 地址	IP 地址通过 DHCP 获得。
BOOTP 分配的 IP 地址	IP 地址通过 BOOTP 获得。
固定 IP 地址	要配置固定 IP 地址，可使用**设备编辑器**的控制器选择(参见第 *78* 页)选项卡。
以太网协议	使用的协议类型：Ethernet2
传输速率	总线上的传输速率和方向会自动配置。

缺省 IP 地址

缺省 IP 地址为 190.201.100.100.

缺省子网掩码是 255.255.255.0。

地址类别

IP 地址与以下两项相关联：

- 设备（主机）
- 该设备所连接的网络

IP 地址始终采用 4 字节进行编码。

这些字节在网络地址和设备地址之间的分配可能会发生变化。具体分配方法由地址类别定义。

下表定义了不同的 IP 地址类别：

<table>
<tr><th>地址类别</th><th colspan="5">字节 1</th><th>字节 2</th><th>字节 3</th><th>字节 4</th></tr>
<tr><td>A 类</td><td>0</td><td colspan="4">网络 ID</td><td colspan="3">主机 ID</td></tr>
<tr><td>B 类</td><td>1</td><td>0</td><td colspan="3">网络 ID</td><td colspan="2">主机 ID</td></tr>
<tr><td>C 类</td><td>1</td><td>1</td><td>0</td><td colspan="4">网络 ID</td><td>主机 ID</td></tr>
<tr><td>D 类</td><td>1</td><td>1</td><td>1</td><td>0</td><td colspan="4">多播地址</td></tr>
<tr><td>E 类</td><td>1</td><td>1</td><td>1</td><td>1</td><td>0</td><td colspan="3">保留地址，供以后使用</td></tr>
</table>

子网掩码

子网掩码的作用是在同一个网络地址下为多个物理网络编址。掩码用于划分子网地址和主机 ID 的设备地址。

获取子网地址的方法是：保留 IP 地址中与包含 1 的掩码的位置相对应的位，然后用 0 替换其他位。

反之，获取主机设备子网地址的方法是：保留 IP 地址中与包含 0 的掩码的位置相对应的位，然后用 1 替换其他位。

子网地址的示例：

IP 地址	192 (11000000)	1 (00000001)	17 (00010001)	11 (00001011)
子网掩码	255 (11111111)	255 (11111111)	240 (11110000)	0 (00000000)
子网地址	192 (11000000)	1 (00000001)	16 (00010000)	0 (00000000)

注意： 如果没有网关，设备不在其子网中进行通讯。

网关地址

网关可将消息路由到不在当前网络中的设备。

如果没有网关，则网关地址为 0.0.0.0。

安全参数

安全参数	描述
激活 SoMachine 协议	使用此参数可在以太网接口上停用 SoMachine 协议。停用后，来自每个设备的每个 SoMachine 请求会被拒绝，包括那些来自 UDP 或 TCP 连接的请求。因此，在以太网上将不会与安装 SoMachine 的 PC、希望与此控制器交换变量的 HMI 目标、OPC 服务器或 Controller Assistant 建立任何连接。
激活 Modbus 服务器	使用此参数可以停用 Logic Controller 的 Modbus 服务器。停用后，对 Logic Controller 的每个 Modbus 请求都会被忽略。
激活 FTP 服务器	使用此参数可以停用 Logic Controller 的 FTP 服务器。停用后，FTP 请求会被忽略。
激活 Discovery 协议	使用此参数可停用 Discovery 协议。停用后，Discovery 请求会被忽略。
Web 可视化协议处于活动状态	使用此参数可以停用控制器的 Web 可视化页面。停用后，对 Logic Controller WebVisualisation 协议的 HTTP 请求会被忽略。

节点名称

节点名称用于在网络扫描期间识别设备。网络中的每台设备必须具有唯一的**节点名称**。

LMC078 运动控制器的**节点名称**是其在**设备树**中所显示的控制器名称。

Modbus TCP 服务器/客户端

简介

与 Modbus 串行链路不同，Modbus TCP 不以层次结构为基础，而是基于客户端/服务器模型。

Modicon LMC078 Motion Controller 既可以执行客户端服务，也可以执行服务器服务，因此，它既可以启动与其他控制器和 I/O 设备的通讯，也能响应来自其他控制器、SCADA、HMI 以及其他设备的请求。

无需任何配置，控制器的嵌入式以太网端口即可支持 Modbus 服务器。

Modbus 客户端/服务器包含在固件中，不需用户进行任何编程操作。通过此功能，可以在 RUNNING、STOPPED 和 EMPTY 状态下对其进行访问。

Modbus TCP 客户端

Modbus TCP 客户端无需任何配置，即可支持 PLCCommunication 库中的以下功能块：

- ADDM
- READ_VAR
- SEND_RECV_MSG
- SINGLE_WRITE
- WRITE_READ_VAR
- WRITE_VAR

有关详细信息，请参阅功能块描述 *(参见 SoMachine, Modbus 和 ASCII 读/写功能, PLCCommunication 库指南)*。

Modbus TCP 服务器

Modbus 服务器支持以下 Modbus 请求：

功能代码 十进制（十六进制）	子功能 十进制（十六进制）	功能
1(1)	–	读取数字量输出 (%Q)
2 (2)	–	读取数字量输入 (%I)
3 (3)	–	读取保持寄存器 (%MW)
6 (6)	–	写入单个寄存器 (%MW)
8 (8)	–	诊断
15 (F)	–	写入多个数字量输出 (%Q)
16 (10)	–	写入多个寄存器 (%MW)
23 (17)	–	读取/写入多个寄存器 (%MW)
43 (2B)	14 (E)	读取设备标识

注意： 嵌入式 Modbus 服务器仅确保单字（2 字节）的时间一致性。如果应用程序需要多于 1 个字的时间一致性，须添加和配置 (参见第 *161* 页)添加和配置一个 **Modbus TCP 从站设备**，使得 %IW 和 %QW 缓冲区内容在相关 IEC 任务（默认为 MAST）中保持时间一致性。

诊断请求

下表包含数据选择代码列表：

数据选择代码（十六进制）	描述
00	保留
01	基本网络诊断
02	以太网端口诊断
03	Modbus TCP/端口 502 诊断
04	Modbus TCP/端口 502 连接表
05 - 7E	为其他公共代码保留
7F	数据结构偏移

FTP 服务器

简介

安装在连接到控制器（通过以太网端口）但未安装 SoMachine 的计算机上的任何 FTP 客户端，均可用来在控制器的数据存储区域传输和接收文件。

注意： Schneider Electric 在控制系统的开发和实施过程中严格遵循行业最佳实践。这其中包括一种“深度防御”方法，旨在保护工业控制系统的安全。此方法将控制器置于一个或多个防火墙之后，将访问范围限制为仅经过授权的人员和协议。

⚠ 警告

未经授权访问及其导致的未经授权的机器操作

- 评估环境或机器是否已连接到关键基础结构，如果已连接，请在将自动化系统连接到任何网络之前，基于深度防护采取适当的预防措施。
- 将连接到网络的设备数限制为所需的最小数量。
- 将工业网络与公司内部的其他网络隔离。
- 使用防火墙、VPN 或其他经证实的安全措施，防止意外访问任何网络。
- 监控系统内的活动。
- 防止未经授权方或未经身份验证的操作直接访问或直接链接主体设备。
- 准备恢复计划，包括系统和过程信息的备份。

不遵循上述说明可能导致人员伤亡或设备损坏。

注意： 在当前登录的目标设备的在线用户管理中使用安全相关命令 *(参见 SoMachine, 菜单命令, 联机帮助)*，这些命令提供了用于添加、编辑和删除用户的方法。

即使控制器为空（未启用用户应用程序和用户权限），也可使用 FTP 服务器。

FTP 访问

如果在控制器中启用了用户权限，则通过用户权限控制对 FTP 服务器的访问。有关详细信息，请参阅**用户和组**选项卡描述 (参见第 *68* 页)。

如果在控制器中尚未激活用户权限，则系统会提示您输入用户名和密码。缺省用户名为 USER，缺省密码为 USER。

注意： 最好是使用用户权限来帮助保护整个控制器。

如果用户权限在控制器中未被启用，出于安装安全考虑，您必须在首次登录时立即更改缺省密码。密码可使用 FC_UserChangePassword *(参见 Modicon LMC078 Motion Controller, 系统功能和变量, PLCSystem 库指南)* 功能进行修改。

⚠ 警告

未经授权的数据访问

- 立即将缺省密码更改为新的安全密码。
- 切勿将密码分发给未经授权人员或其他不符合资格的人员。
- 禁用 FTP 可阻止任何意外或未经授权的应用程序数据访问。

不遵循上述说明可能导致人员伤亡或设备损坏。

注意： 安全密码是没有共享或分发给任何未经授权人员的密码，而且不包含任何个人或其他明显易知的信息。而且，混合使用大小写字母和数字可以提供最大安全保护。密码长度应不少于 7 个字符。

注意： 要获取对已启用用户访问权限，且您没有密码的控制器的访问，唯一的方法就是执行“升级固件”操作。此外，您还可以通过执行脚本来清除控制器中的用户权限（有关详细信息，请参阅 SoMachine - 编程指南）。这个方法可以有效地从控制器存储器中删除现有应用程序，但是也恢复了访问该控制器的能力。

如果您未启用用户权限，且已经丢失或忘记密码，则需要使用 SoMachine 直接连接到控制器，并执行初始值复位以重新建立缺省密码。执行此操作后，设置新的安全密码。

文件访问

请参阅文件结构 (参见第 *31* 页)。

FTP 客户端

简介

FtpRemoteFileHandling 库针对远程文件处理提供以下 FTP 客户端功能：

- 读取文件
- 写入文件
- 删除文件
- 列出远程目录的内容
- 添加目录
- 删除目录

注意： Schneider Electric 在控制系统的开发和实施过程中严格遵循行业最佳实践。这其中包括一种“深度防御”方法，旨在保护工业控制系统的安全。此方法将控制器置于一个或多个防火墙之后，将访问范围限制为仅经过授权的人员和协议。

⚠ 警告

未经授权访问及其导致的未经授权的机器操作

- 评估环境或机器是否已连接到关键基础结构，如果已连接，请在将自动化系统连接到任何网络之前，基于深度防护采取适当的预防措施。
- 将连接到网络的设备数限制为所需的最小数量。
- 将工业网络与公司内部的其他网络隔离。
- 使用防火墙、VPN 或其他经证实的安全措施，防止意外访问任何网络。
- 监控系统内的活动。
- 防止未经授权方或未经身份验证的操作直接访问或直接链接主体设备。
- 准备恢复计划，包括系统和过程信息的备份。

不遵循上述说明可能导致人员伤亡或设备损坏。

有关详细信息，请参阅 FtpRemoteFileHandling 库指南。

LMC078 Motion Controller 用作 Modbus TCP 上的 IOScanner 从站设备

概述

本节介绍 LMC078 Motion Controller 用作 **Modbus TCP 从站设备**的配置。

Modbus TCP 从站设备会在控制器上创建一个可通过 Modbus TCP 协议进行访问的特定 I/O 区域。当外部 I/O 扫描器（主站）需要访问控制器的 %IW 和 %QW 对象时会用到该区域。使用 **Modbus TCP 从站设备**的主要优势是控制器对象被集中到一起，可以通过一个 Modbus 请求进行访问。

Modbus 从站设备会将另一个 Modbus 服务器功能添加到控制器。Modbus 客户端应用程序可以使用配置的单元 ID（非 255）访问此服务。从站控制器的嵌入式 Modbus 服务器无需配置，可通过单元 ID = 255 寻址。

可以从从站控制器看到输入/输出：输入由主站写入，输出由主站读取。

TCP 从站设备可以定义特权 Modbus 客户端应用程序，该应用程序的连接不会被强制关闭（嵌入式 Modbus 连接可能会在需要 8 个以上的连接时被关闭）。

有关 Modbus TCP 的详细信息，请参阅 www.odva.org 网站。

添加 Modbus TCP 从站设备

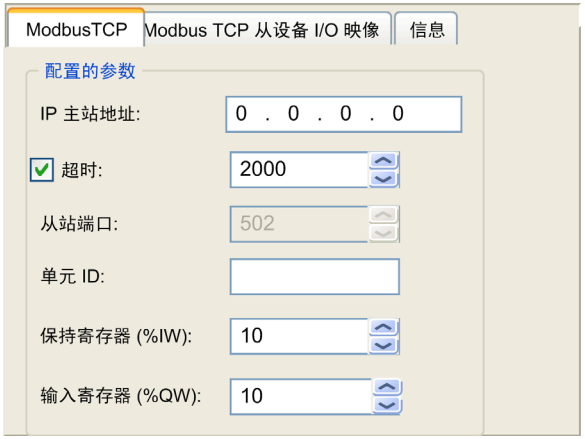
将 LMC078 Motion Controller 配置作为 Modbus TCP 从站设备：

步骤	动作
1	选择 **以太网网络**节点，然后点击加号按钮。 **结果**：将显示**添加设备**对话窗口。
2	选择**Modbus TCP 从站设备**。
3	单击**添加设备**。
4	单击**关闭**。

Modbus TCP 配置

若要配置 **Modbus TCP 从站设备**，请双击**设备树**中的**以太网网络 → Modbus TCP 从站设备**。

将出现以下对话框：



元素	描述
IP 主站地址	Modbus 主站的 IP 地址 此地址上的连接不会关闭。
超时	超时按 500 毫秒递增 **注意：** 超时适用于 IP 主站地址，但该地址为 0.0.0.0 时除外。
从站端口	Modbus 通讯端口 (502)
单元 ID	将请求发送到 Modbus TCP 从站设备 (1...247)，而不是发送到嵌入式 Modbus 服务器 (255)。
保持寄存器 (%IW)	交换要使用的 %IW 寄存器数 (2...40)（每个寄存器为 2 字节）
输入寄存器 (%QW)	交换要使用的 %QW 寄存器数 (2...40)（每个寄存器为 2 字节）

Modbus TCP 从站设备 I/O 映射选项卡

从主站角度看，I/O 映射到 Modbus 寄存器，如下所示：

- %IW 从寄存器 0 映射到 n-1，且可读/写（n = 保持寄存器数量，每个 %IW 寄存器的大小为 2 个字节）。
- %QW 从寄存器 n 映射到 n+m-1，且为只读（m = 输入寄存器数量，每个 %QW 寄存器的大小为 2 个字节）。

在配置 **Modbus TCP 从站设备**后，发送到其单元 ID（Modbus 地址）的 Modbus 命令便会访问控制器的 `%IW` 和 `%QW` 对象，而不是当单元 ID 为 255 时被访问的常规 Modbus 字。这有利于 Modbus TCP IOScanner 应用程序进行读/写操作。

Modbus TCP 从站设备响应 Modbus 命令 目的是与外部 I/O 扫描器交换数据。以下 Modbus 命令受 Modbus TCP 从站设备支持：

功能代码十进制（十六进制）	功能	注释
3 (3)	读取保持寄存器	允许主站读取设备的 %IW 和 %QW 对象
6 (6)	写入单个寄存器	允许主站写入设备的 %IW 对象
16 (10)	写入多个寄存器	允许主站写入设备的 %IW 对象
23 (17)	读取/写入多个寄存器	允许主站读取设备的 %IW 和 %QW 对象，并写入设备的 %IW 对象
其他	不支持	–

注意： 尝试访问 n+m-1 以上的寄存器的 Modbus 请求通过“02 - 非法数据地址”例外代码进行应答。

要将 I/O 对象链接到变量，请选择 **Modbus TCP 从站设备 I/O 映射**选项卡：

Modbus TCP | Modbus TCP 从站设备 I/O 映射 | 信息

通道

变量	映射	通道	地址	类型	缺省值	单位	描述
		输入	%IW0	数组 [0...9]...			Modbus 保持...
Application.POU.tata		Inputs[0]	%IW0	字			
iwModbusTCT_Sla...		Inputs[1]	%IW1	字			
iwModbusTCT_Sla...		Inputs[2]	%IW2	字			
iwModbusTCT_Sla...		Inputs[3]	%IW3	字			
iwModbusTCT_Sla...		Inputs[4]	%IW4	字			
iwModbusTCT_Sla...		Inputs[5]	%IW5	字			
iwModbusTCT_Sla...		Inputs[6]	%IW6	字			
iwModbusTCT_Sla...		Inputs[7]	%IW7	字			
iwModbusTCT_Sla...		Inputs[8]	%IW8	字			
iwModbusTCT_Sla...		Inputs[9]	%IW9	字			
		输出	%QW0	数组 [0...9]...			Modbus 输入...
qwModbusTCP_Sl...		Outputs[0]	%QW0	字			
qwModbusTCP_Sl...		Outputs[1]	%QW1	字			
qwModbusTCP_Sl...		Outputs[2]	%QW2	字			
qwModbusTCP_Sl...		Outputs[3]	%QW3	字			
qwModbusTCP_Sl...		Outputs[4]	%QW4	字			
qwModbusTCP_Sl...		Outputs[5]	%QW5	字			
qwModbusTCP_Sl...		Outputs[6]	%QW6	字			
qwModbusTCP_Sl...		输出[7]	%QW7	字			
qwModbusTCP_Sl...		输出[8]	%QW8	字			
qwModbusTCP_Sl...		输出[9]	%QW9	字			

复位映射　☑ 一直更新变量

IEC 对象

变量	映射	类型
Modbus TCP_Slave_De		IoDrvModbusTCPSlave

= 创建新变量　= 映射到已存在的变量

总线循环选项

总线循环任务　使用父总线循环设置

通道		类型	描述
输入	IW0	WORD	保持寄存器 0
	...	...	...
	IWx	WORD	保持寄存器 x
输出	QW0	WORD	输入寄存器 0
	...	...	...
	QWy	WORD	输入寄存器 y

字数取决于 Modbus TCP 选项卡的**保持寄存器 (%IW)** 和**输入寄存器 (%QW)** 参数。

注意： 输出表示来自起点控制器的 OUTPUT（= 控制器的 %IW）。输入表示来自起点控制器的 INPUT（= 控制器的 %QW）。

注意： **Modbus TCP 从站设备**将 %IW 和 %QW 寄存器当作单个时间一致单元来刷新，与 IEC 任务（默认为 MAST 任务）同步。相比之下，嵌入式 Modbus TCP 服务器仅确保 1 个字（2 字节）的时间一致性。如果应用程序需要多于 1 个字（2 个字节）的时间一致性，须使用 **Modbus TCP 从站设备**。

总线循环选项

选择**总线循环任务**以便使用：

- **使用父总线循环设置**（缺省），
- MAST
- **项目的现有任务**

注意： 在设备的 I/O 映射编辑器中有一个对应的**总线循环任务**，其中包含 **Modbus TCP 从站设备**。此参数定义负责刷新 %IW 和 %QW 寄存器的任务。

第11.2节
防火墙配置

简介

本节介绍如何配置 Modicon LMC078 Motion Controller 的防火墙。

本节包含了哪些内容？

本节包含了以下主题：

主题	页
简介	167
防火墙行为	169
防火墙脚本命令	170
脚本文件	173

简介

防火墙介绍

一般情况下，防火墙通过拦截未授权的访问和允许授权的访问帮助来保护网络安全区周围。防火墙指的是一台设备或一套设备，基于一套规则和其他标准将其配置为允许、拒绝、加密或代理不同安全区之间的流量。

流程控制设备和高速生产机器要求快速的数据吞吐量，并且经常不能容忍控制网络中进攻性安全策略所引入的延迟。因此，通过在网络周围提供保护，防火墙在安全策略中扮演了重要的角色。防火墙是整个系统级别策略的重要组成部分。

注意： Schneider Electric 在控制系统的开发和实施过程中严格遵循行业最佳实践。这其中包括一种“深度防御”方法，旨在保护工业控制系统的安全。此方法将控制器置于一个或多个防火墙之后，将访问范围限制为仅经过授权的人员和协议。

⚠ 警告

未经授权访问及其导致的未经授权的机器操作

- 评估环境或机器是否已连接到关键基础结构，如果已连接，请在将自动化系统连接到任何网络之前，基于深度防护采取适当的预防措施。
- 将连接到网络的设备数限制为所需的最小数量。
- 将工业网络与公司内部的其他网络隔离。
- 使用防火墙、VPN 或其他经证实的安全措施，防止意外访问任何网络。
- 监控系统内的活动。
- 防止未经授权方或未经身份验证的操作直接访问或直接链接主体设备。
- 准备恢复计划，包括系统和过程信息的备份。

不遵循上述说明可能导致人员伤亡或设备损坏。

防火墙配置

可通过以下两种方式管理控制器防火墙配置：

- 静态配置，
- 应用程序设置。

在静态配置中使用的脚本文件。

静态配置

在控制器启动时加载静态配置。

可通过管理位于控制器中的缺省脚本文件来静态配置控制器防火墙。此文件的路径是 */Usr/Cfg/FirewallDefault.cmd*。

应用程序设置

请参阅以太网配置 (参见第 *153* 页)。

防火墙行为

简介

防火墙配置取决于在控制器上执行的操作和初始配置状态。有 4 种可能的初始状态：

- 在控制器中没有缺省脚本文件。
- 存在正确脚本文件。
- 存在不正确的脚本文件。
- 不存在缺省脚本文件，并且应用程序已配置防火墙。

无缺省脚本文件

如果...	则...
启动控制器	不配置防火墙。不激活任何保护。
下载应用程序	根据应用程序设置配置防火墙。

存在缺省脚本文件

如果...	则...
启动控制器	根据缺省脚本文件配置防火墙。
下载应用程序	忽略应用程序的整个配置。 根据缺省脚本文件配置防火墙。

存在不正确的缺省脚本文件

如果...	则...
启动控制器	不配置防火墙。不激活任何保护
下载应用程序	根据应用程序设置配置防火墙。

无缺省脚本文件的应用程序设置

如果...	则...
启动控制器	根据应用程序设置配置防火墙。
下载应用程序	删除上一个应用程序的整个配置。 根据新应用程序设置配置防火墙。

防火墙脚本命令

概述

本节介绍如何编写脚本文件（缺省脚本文件或动态脚本文件），以便在启动控制器期间或触发的特定命令期间执行脚本文件。

脚本文件语法

脚本语法指南 (参见第 *173* 页)中描述脚本文件的语法。

一般防火墙命令

提供下列命令以管理 LMC078 Motion Controller 以太网防火墙：

命令	描述
FireWall Enable	阻止来自 Ethernet 接口的帧。如果未授权指定 IP 地址，将无法在 Ethernet 接口上进行任何通讯。 **注意：** 缺省情况下，在启用防火墙时，将拒绝帧。
FireWall Disable	允许 IP 地址在 Ethernet 接口上访问控制器。
FireWall Eth1 Default Allow	控制器接受帧。
FireWall Eth1 Default Reject	控制器拒绝帧。 **注意：** 缺省情况下，如果不存在此行，则相当于命令 FireWall Eth1 Default Reject。

特定防火墙命令

提供下列命令以配置特定端口和地址的防火墙规则：

命令	范围	描述
Firewall Eth1 Allow IP •.•.•.•	• = 0...255	在所有端口号和端口类型上允许来自指定 IP 地址的帧。
Firewall Eth1 Reject IP •.•.•.•	• = 0...255	在所有端口号和端口类型上拒绝来自指定 IP 地址的帧。
Firewall Eth1 Allow IPs •.•.•.• to •.•.•.•	• = 0...255	所有端口号和端口类型都允许来自指定范围中的 IP 地址的帧。
Firewall Eth1 Reject IPs •.•.•.• to •.•.•.•	• = 0...255	所有端口号和端口类型都拒绝来自指定范围中的 IP 地址的帧。
Firewall Eth1 Allow port_type port Y	Y = (目标端口号 (参见第 *172* 页))	允许带有指定目标端口号的帧。

命令	范围	描述
Firewall Eth1 Reject port_type port Y	Y = (目标端口号 (参见第 *172* 页))	允许带有指定目标端口号的帧。
Firewall Eth1 Allow port_type ports Y1 to Y2	Y = (目标端口号 (参见第 *172* 页))	允许带有指定范围中的目标端口号的帧。
Firewall Eth1 Reject port_type ports Y1 to Y2	Y = (目标端口号 (参见第 *172* 页))	拒绝带有指定范围中的目标端口号的帧。
Firewall Eth1 Allow IP •.•.•.• on port_type port Y	• = 0...255 Y = (目标端口号 (参见第 *172* 页))	允许来自指定 IP 地址并带有指定目标端口号的帧。
Firewall Eth1 Reject IP •.•.•.• on port_type port Y	• = 0...255 Y = (目标端口号 (参见第 *172* 页))	拒绝来自指定 IP 地址并带有指定目标端口号的帧。
Firewall Eth1 Allow IP •.•.•.• on port_type ports Y1 to Y2	• = 0...255 Y = (目标端口号 (参见第 *172* 页))	允许来自指定 IP 地址并带有指定范围中的目标端口号的帧。
Firewall Eth1 Reject IP •.•.•.• on port_type ports Y1 to Y2	• = 0...255 Y = (目标端口号 (参见第 *172* 页))	拒绝来自指定 IP 地址并带有指定范围中的目标端口号的帧。
Firewall Eth1 Allow IPs •1.•1.•1.•1 to •2.•2.•2.•2 on port_type port Y	• = 0...255 Y = (目标端口号 (参见第 *172* 页))	拒绝来自指定范围中的 IP 地址并带有指定目标端口号的帧。
Firewall Eth1 Reject IPs •1.•1.•1.•1 to •2.•2.•2.•2 on port_type port Y	• = 0...255 Y = (目标端口号 (参见第 *172* 页))	拒绝来自指定范围中的 IP 地址并带有指定目标端口号的帧。
Firewall Eth1 Allow IPs •1.•1.•1.•1 to •2.•2.•2.•2 on port_type ports Y1 to Y2	• = 0...255 Y = (目标端口号 (参见第 *172* 页))	允许来自指定范围中的 IP 地址并带有指定范围中的目标端口号的帧。

命令	范围	描述
Firewall Eth1 Reject IPs •1.•1.•1.•1 to •2.•2.•2.•2 on port_type ports Y1 to Y2	• = 0...255 Y = (目标端口号 (参见第 *172* 页))	拒绝来自指定范围中的 IP 地址并带有指定范围中的目标端口号的帧。
Firewall Eth1 Allow MAC ••:••:••:••:••:••	• = 0...F	允许来自指定 MAC 地址 ••:••:••:••:•• 的帧。
Firewall Eth1 Reject MAC ••:••:••:••:••:••	• = 0...F	拒绝带有指定 MAC 地址 ••:••:••:••:•• 的帧。

脚本示例

```
; Enable firewall on Ethernet 1. All frames are rejected;
FireWall Enable;
; Block all Modbus Requests on all IP address
Firewall Eth1 Reject tcp port 502;
; Allow FTP active connection for IP address 85.16.0.17
Firewall Eth1 Allow IP 85.16.0.17 on tcp port 20 to 21;
```

使用的端口

协议	目标端口号
SoMachine	UDP 1740、1741、1742、1743 TCP 1105
FTP	TCP 21、20
HTTP	TCP 80
Modbus	TCP 502
Discovery	UDP 27126、27127
NVL	UDP 缺省值：1202
EtherNet/IP	UDP 2222 TCP 44818

脚本文件

概述

下面描述如何写入脚本文件以配置以太网防火墙 (参见第 *170* 页)。

脚本语法指南

脚本中的每一行命令都以 ";" 为结尾。

如果某行以 ";" 为开头，则表明该行是注释行。

脚本文件中的最大行数为 50。

语法不区分大小写。

如果在脚本文件中不遵守该语法，则不会执行该脚本文件。例如，这表示防火墙配置将保持上一个状态。

注意： 如果脚本文件没有被执行，则会生成一个日志文件。该日志文件在控制器中的位置是 */usr/Syslog/FWLog.txt*。

第12章
CANopen 配置

简介

本章介绍如何配置控制器内提供的 CAN 接口。

本章包含了哪些内容？

本章包含了以下主题：

主题	页
CANopen 接口配置	176
CANopen 主站配置	177
CANopen 从站配置	179

CANopen 接口配置

添加 CAN 总线节点

要将 CAN_Layer 节点添加到控制器，选择**硬件目录**中的 CANbus，然后将其拖放到**设备树**，并将其放置在控制器节点上。

有关将设备添加到项目的更多信息，请参阅：

- 使用拖放方法 *(参见 SoMachine, 编程指南)*
- 使用上下文菜单或加号按钮 *(参见 SoMachine, 编程指南)*

CAN 总线配置

要配置控制器的 CAN 总线，请执行以下操作：

步骤	操作
1	在**设备树**中，双击 CAN_Layer。
2	配置波特率（缺省情况下：250000 位/秒）。

添加 CANopen 管理器

控制器支持以下 CANopen 管理器：

- 将 CAN 端口的 CANopen_Manager 配置为 CANopen 主站
- 将 CAN 端口的 CAN_Local_Device 配置为 CANopen 从站

要将 CANopen 管理器添加到控制器，请在**硬件目录**中选择：

- 对于 CANopen 主站：CANopen_Manager
- 对于 CANopen 从站：CAN_Local_Device

将其拖动到**设备树**并将其放置到高亮显示的节点上。

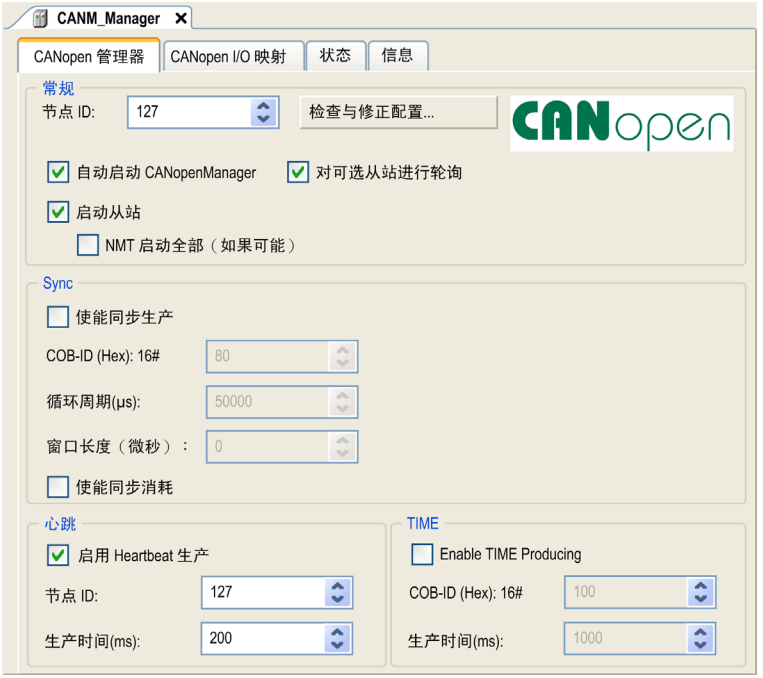
有关将设备添加到项目的更多信息，请参阅：

- 使用拖放方法 *(参见 SoMachine, 编程指南)*
- 使用上下文菜单或加号按钮 *(参见 SoMachine, 编程指南)*

CANopen 主站配置

CANopen 管理器配置

要配置 **CANopen_Manager**，请执行以下步骤：

步骤	操作
1	双击**设备树**中的 **CANopen_Manager**。 **结果：**显示 **CANopen 管理器**配置窗口： 
2	有关 CANopen 管理器配置的详细信息，请参阅 SoMachine 在线帮助中的*使用 SoMachine/设备编辑器/CANopen 管理器编程*一章。

添加 CANopen 设备

有关添加通讯管理器和将从站设备添加到通讯管理器的详细信息，请参阅《SoMachine 编程指南》。

CANopen 操作限制

Modicon LMC078 Motion Controller CANopen 主站有如下操作限制：

从站设备的最大数	63
已接收 PDO (RPDO) 的最大数	252
已传输 PDO (TPDO) 的最大数	252

⚠ 警告

意外的设备操作

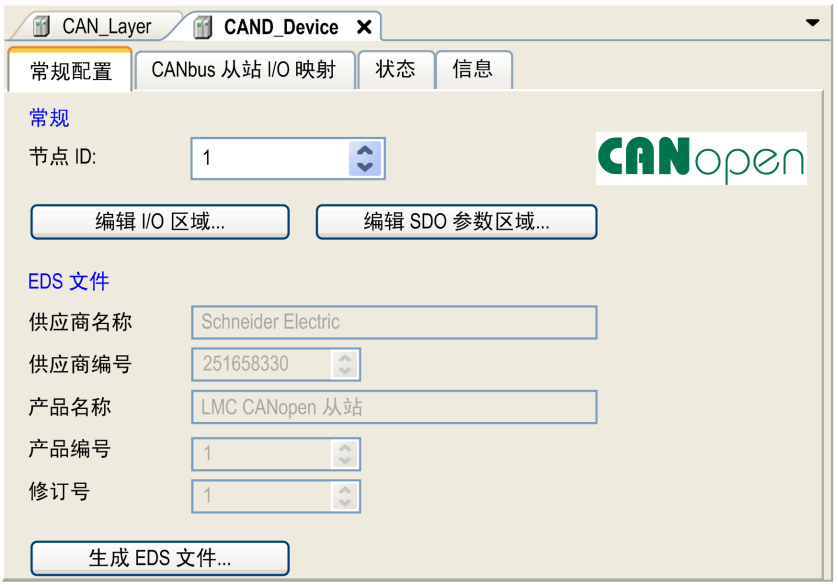
- 连接到控制器的 CANopen 从站设备不能超过 63 个。
- 对应用程序进行编程，以使用 252 个或更少的传输 PDO (TPDO)。
- 对应用程序进行编程，以使用 252 个或更少的接收 PDO (RPDO)。

不遵循上述说明可能导致人员伤亡或设备损坏。

CANopen 从站配置

CANopen 从站配置

要将控制器配置为 CANopen 从站，请执行以下步骤：

步骤	操作
1	选择**设备树**中的控制器节点，单击该节点的绿色加号按钮，或者双击该节点，然后从上下文菜单中选择**添加设备**。 **结果**：**添加设备**对话框随即打开。
2	在**添加设备**对话框中，选择 **CANbus** 并单击**添加设备**按钮。 **结果**：设备已添加到控制器。
3	单击**添加设备**对话框中的**关闭**按钮。
4	选择**设备树**中的 **CAN_Layer** 节点，单击该节点的绿色加号按钮，或者双击该节点，然后从上下文菜单中选择**添加设备**。
5	在**添加设备**对话框中，选择 **CAN Local_Device** 并单击**添加设备**按钮。 **结果**：设备已添加到 **CAN_Layer** 节点。
6	单击**添加设备**对话框中的**关闭**按钮。
7	双击**设备树**中的 **CAND_Device**。 **结果**：将显示 **CAND_Device** 配置窗口： 
8	有关 CANopen 管理器配置的详细信息，请参阅 SoMachine 在线帮助中的*使用 SoMachine/设备编辑器/CANbus 从站设备*一章。

第13章
Sercos配置

简介

本章介绍如何配置 Sercos 的 Modicon LMC078 Motion Controller 接口。

本章包含了哪些内容？

本章包含了以下主题：

主题	页
Sercos 标准概述	182
Sercos 接口配置	184
Sercos设备	187
设备寻址编辑器	188
Lexium LXM32S 驱动器配置	191
TM5NS31 Sercos Interface Module	194
Sercos 错误代码	195

Sercos 标准概述

简介

Sercos 接口是一个标准化接口 (IEC 6149)，用于在控制器、驱动器和 I/O 设备之间进行实时通讯。

本节介绍用于在控制单元和连接在一起形成运动控制系统的相关伺服驱动器之间进行通讯的国际标准化数字接口。此外，还从扭矩、速度或位置接口操作模式方面定义机器可以与多个驱动器操作的运行数据、参数和比例调整的标准化。

Sercos 接口的主要功能如下：

- 环型拓扑结构（冗余）
- 主站/从站系统
- 波特率 100 MBaud
- 最少同步时间 1 ms（8 轴）、2 ms（16 轴）或 4 ms（24 轴）
- 同步（抖动 < 1 微秒）

数据交换

与 Sercos 接口的通讯分成两种类型：

- 与报文的循环通讯：
  循环通讯用于交换实时数据（如位置），并在每个通讯循环中执行一次 (CycleTime)。在每个循环中，将某些指定的数据从控制器传输到所有驱动器，然后从所有驱动器传输到控制器。
- 与 LMC078_Sercos3 库 (参见第 *247* 页)的功能块的非循环通讯。
  非循环通讯用于交换数据（如用于配置通讯的参数、驱动器参数、状态等），其中时间不是关键因素。控制器可以控制非循环通讯。可以使用此通道联系系统中的所有参数，甚至已经循环配置这些参数。

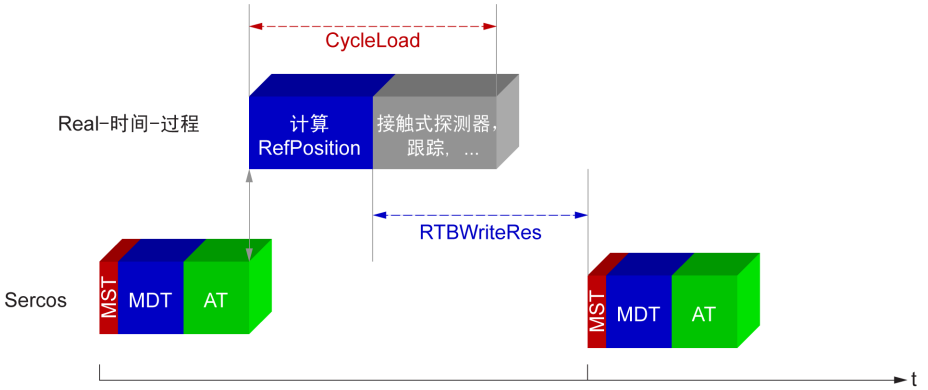
注意： 可以同时使用两种类型的通讯。

循环数据交换

可以通过消息结构（称为报文）实现在运动控制器（Sercos 主站）和伺服驱动器（从站）之间交换信息。已通过 IEC 61491 定义三种报文：

- MST（主站同步报文）：主站会在各个传输循环开始时广播 MST 报文以同步循环计时。
- MDT（主站数据报文）：主站会在各个传输循环中发送一次 MDT 报文，以便将数据（命令值）传输到伺服驱动器（从站）。
- AT（确认报文）：从站会将 AT 报文发送到主站（反馈值）。

下图显示了三种类型的报文：



本主题仅讨论 MDT 和 AT 报文类型。一般报文结构显示如下：

报文 分隔符	地址 字段	可配置数据字段	帧检验 序列	报文 分隔符

报文的管理段包含传输所有报文所需的报文分隔符（开始）、地址字段、帧检验序列和报文分隔符（结束）。

在报文内，在每个通讯周期中，传送可配置数据字段中的实时数据（操作数据）。标识号 (IDN) 提供了此数据的规范。

Sercos 可实现控制器的处理循环与驱动器中的数据交换和控制循环同步。因此，在这些单独的循环之间没有任何干扰，并且控制回路具有最小的恒定空载时间。此外，所有驱动器中的新参考值都将生效，并且所有总线从站都会同时记录转发给控制器作为实际值的测量值。

IDN 描述

IEC 61491 将标识号 (IDN) 分配给 Sercos 驱动器中的所有操作数据。操作数据包括参数、接口程序命令、命令和反馈值。

可以使用两种类别的 IDN：

- 标准 IDN (S)：通过 Sercos 标准 IEC 61491 定义。如果 Sercos 驱动器支持，则标准 IDN 的行为相同，而不考虑驱动器制造商。
- 专有 IDN (P)：它们可以为控制单元和伺服驱动器的制造商所定义的产品特定数据所保留。

注意： 有关在 LXM32S 驱动器中实现的 Sercos IDN 的完整的详细信息，请参阅《Lexium LXM32S Product Manual》。

IDN 通常是 Sercos 驱动器的 16 位或 32 位二进制参数。可以通过以下方法确定 IDN：

- S-0-0047.0.0 标准 IDN：位置命令值
- P-0-3017.0.12 制造商定义的 IDN：当前限制

-0- 指参数设置。许多 Sercos 驱动器（包括 Lexium）不支持参数设置。

Sercos 接口配置

简介

Sercos 配置窗口可让您配置和查看 Sercos 接口参数。

Sercos 接口配置

要访问 Sercos 配置窗口，双击**设备树**中的 **SERCOSIII** 节点。

配置窗口随即显示，如下所示：

Sercos ✕

配置

参数	类型	值	缺省值	单位
通用				
名称	STRING(40)	"	"	
CycleTime	DINT(1000000..4000000)	1000000	1000000	纳秒
拓扑结构	DINT 枚举	无链路/0	无链路/0	
ScannedDevices	DINT			
UsedDevices	DINT			
LastDevicePl	STRING(40)	"	"	
LastDevicePl	STRING(40)	"	"	
阶段控制				
州/省	DINT 枚举	阶段 0/0	阶段 0/0	
PhaseSet	DINT(0..4)	4	4	
ObjectType	STRING	'SERC32'	'SERC32'	
stLogicalAddress				

下表介绍了 Sercos 接口的参数：

参数	访问	参数类型	数据类型	值	缺省值	描述
公共端						
Name	读/写(*)	EF	STRING	"	"	配置对象的符号名称。
CycleTime	读/写	ER	DINT	1000000 2000000 4000000	1000000	定义Sercos总线循环时间（纳秒）。 可以将 CycleTime 设置为 1 毫秒、2 毫秒或 4 毫秒。 如果已更改 CycleTime，则必须复位控制器。

参数	访问	参数类型	数据类型	值	缺省值	描述
Topology	R	AF	DINT Enum	no link / 0 line P1 / 1 line P2 / 2 double line / 3 ring / 4 defect ring / 8	no link / 0	介绍 Sercos 系统的拓扑结构： ● 0 = 未连接 Sercos 设备。 ● 1 = 已将所有 Sercos 设备连接到端口 1。 ● 2 = 已将所有 Sercos 设备连接到端口 2。 ● 3 = 已将 Sercos 设备连接到端口 1 和端口 2。 ● 4 = 从端口 1 连接到端口 2 已关闭（环已闭合）。 ● 8 = 环形拓扑结构无效，拓扑转换尚未完成。
ScannedDevices	R	AD	DINT	-	-	物理扫描的 Sercos 设备数。
UsedDevices	R	AD	DINT	-	-	配置和物理分配的 Sercos 设备数。
LastDeviceP1	R	AD	STRING	“	“	端口 1 上最后一个物理 Sercos 设备的名称。
LastDeviceP2	R	AD	STRING	“	“	端口 2 上最后一个物理 Sercos 设备的名称。
阶段控制						
State	R	AD	DINT Enum	Phase 0 / 0 Phase 1 / 1 Phase 2 / 2 Phase 3 / 3 Phase 4 / 4 Firmware download / 5 Phase 6 / 6 Bus scan / 7 Reinit Sercos / 8 Init / 10 Error / 11 Continuous light / 12 Zero bit stream / 13	Phase 0 / 0	显示 Sercos 系统状态： ● 0 = 配置和启动。 ● 1 = 正常运行，Sercos 阶段 1。 ● 2 = 正常运行，Sercos 阶段 2。 ● 3 = 正常运行，Sercos 阶段 3。 ● 4 = 正常运行，Sercos 阶段 4。 ● 5 = 正在进行固件下载。 ● 6 = 保留 ● 7 = 正在进行总线扫描。 ● 8 = 正在进行 Sercos 重新初始化。 ● 10 = 只会在系统启动过程中短暂发生。 ● 11 = 在运行阶段已检测到 Sercos 错误。 ● 12 = 不适用（对于光连接）。 ● 13 = 不适用（对于光连接）。

参数	访问	参数类型	数据类型	值	缺省值	描述
PhaseSet	读/写	EF	DINT	0...4	4	您可以使用此参数预设 Sercos 总线的所需通讯阶段：(1) ● 0...3 = 程序任务与 Sercos 运行同时启动。 ● 4 = 程序任务启动延迟，直到： ❍ Sercos 运行已到达阶段 4 (State = 4)。 ❍ Sercos 运行因启动错误取消 (State = 11)。
—						
ObjectType	R	AD	STRING	SERC32	SERC32	对象类型。
stLogicalAd-dress	R	AD	ST_Logical Address	-	-	Sercos 参数的逻辑地址。 LogicalAddress = STRUCT (udiType, udiInstance, udiParameterId)

(*) 有关参数访问权限的详细信息，请参阅参数类型 (参见第 *25* 页)。

(1)在“PhaseSet”参数表示实时数据交换处于活动状态后，需要重新初始化轴以实现位置的同步。

注意

位置因 SERCOS 总线阶段更改而丢失

当第一次到达 Sercos 阶段 4 时，可通过编程来确保重新初始化运动系统或使其回归。

不遵循上述说明可能导致设备损坏。

Sercos 运行阶段

下表介绍了 Sercos 运行阶段（阶段 0...4）：

运行阶段	描述
相位 0	验证所连接的设备数是否保持恒定以及所使用的拓扑结构。然后，切换到阶段 1 实现所确定的拓扑结构。
阶段 1	验证是否可以连接从站。为此，配置的所有从站都需要进行简短寻址。如果可以连接环中的所有配置总线从站，则通讯阶段 2 变成活动状态。
阶段 2	主站按顺序与每个从站交换有关一般设备属性的重要的通讯参数和数据。此阶段用于定义和修复循环通道的配置。如果已配置所有总线从站，则通讯阶段 3 变成活动状态。
阶段 3	**参数阶段**：主站与从站交换参数。所有从站可以同时进行寻址。实时数据尚不可用。
阶段 4	**运行阶段**：循环交换实时数据（循环通讯）。使用服务通道可以读取和写入任意数量的参数（非循环通讯）。

Sercos设备

简介

Modicon LMC078 Motion Controller 支持以下 Sercos 设备：

- Lexium LXM32S 驱动器
- TM5NS31 Sercos 接口模块
- 第三方 Sercos 设备

添加 Sercos 设备

可通过两种方法添加 Sercos 设备：

- 使用设备寻址编辑器 (参见第 *188* 页)。
- 使用下述方法。

要添加 Sercos 设备，在硬件目录中选择 **Lexium 32 S** 或 **TM5NS31 Interface**，然后将其拖放到**设备树**，并将其放置到 **SERCOSIII** 节点上。

有关将设备添加到项目的更多信息，请参阅：

• 使用拖放方法 *(参见 SoMachine, 编程指南)*

• 使用上下文菜单或加号按钮 *(参见 SoMachine, 编程指南)*

添加第三方 Sercos 设备

您也可以添加来自第三方供应商的 Sercos 设备 *(参见 SoMachine, 编程指南)*。

注意： 根据不同的第三方供应商，可能必须将 Sercos 设备的参数 `Producer-CycleTime` 设置为与缺省值（1 毫秒）不同的值。有关详细信息，请参见第三方供应商提供的设备文档中的规格说明。

设备寻址编辑器

简介

设备寻址编辑器支持以下功能：

- 扫描已连接到 Sercos 总线的设备
- 新增 Sercos 设备
- 定义 Sercos 设备的标识参数

要访问**设备寻址**工具，选择**工具树**选项卡，然后双击**设备寻址**：

名称	描述
PLC 配置中的设备	编辑器窗口的左侧部分显示 LXM32S 驱动器和控制器项目中 Sercos 配置的 TM5NS31 接口模块。 **注意：** 如果自动分配 Sercos 对象，则会按拓扑地址的顺序列出这些对象，从最低值开始。如果已经列出之前扫描的 Sercos 对象，则会按拓扑地址的顺序将任何进一步的 Sercos 对象添加到列表的末尾，从最低值开始。
操作模式 (参见第 *190* 页)	操作模式用于确定此设备的操作方式。 **注意：** 如果在按住 Shift 键的同时更改此列中的某个值，则会将此列中的所有值设置为该值。
开始 Sercos 扫描	单击此按钮可开始搜索已连接到 Sercos 总线的真正的 LXM32S 驱动器和 TM5NS31 接口模块。 要执行搜索： ● 将 Sercos 总线切换到阶段 0 (参见第 *186* 页)。 ● 停止运行所有应用程序。 ● 必须确认所有诊断消息。

名称	描述
扫描的设备	在执行扫描后，编辑器窗口的右侧部分显示已连接到 Sercos 总线的 LXM32S 驱动器和 TM5NS31 接口模块。列标题列出了扫描设备的数量和自动分配的设备数。 自动分配的所有设备均以非白色高亮显示。您可以稍后通过使用最右侧列中的下拉列表手动更改分配。
<--	单击此按钮可在此行中应用已分配 Sercos 设备的值。
采用所有已分配设备的值	在分配所有 LXM32S 驱动器和 TM5NS31 接口模块后，单击此按钮可在控制器配置的 Sercos 节点中应用所分配对象中已扫描 Sercos 设备的设备数据。
添加	单击此按钮可将一部分新设备添加到控制器配置的 Sercos 节点。

注意： **设备寻址**编辑器的功能仅在离线模式下可用。

行的颜色编码表示是否可以为控制器配置中的特定 Sercos 对象执行登录，如果可以执行登录，则表示登录是否可行。

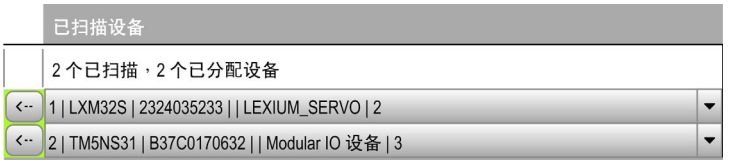
颜色具有以下含义：

颜色	描述
绿色	使用此对象可以执行登录。
红色	使用此对象无法执行登录。
白色	未将任何 Sercos 设备分配给此对象。
黄色	登录到此对象无法确定。所分配 Sercos 设备的值存在偏差。不匹配的值将以**粗体**高亮显示。
粉色	登录到此对象无法确定。但是，在执行 Sercos 扫描（[开始 Sercos 扫描]）后不分配任何 Sercos 设备，即使将**操作模式**设置为**实际**。

Sercos 扫描

在扫描后（**开始 SERCOS 扫描**按钮），程序试图使用拓扑地址将 Sercos 对象从控制器配置分配到已连接到 Sercos 总线的设备。自动分配的所有设备均以非白色高亮显示。

列标题列出了扫描设备的数量和自动分配的设备数：



您可以稍后通过使用最右侧列中的下拉列表更改自动分配。

- 单击想要更改的行中的 ⌄ 按钮。下拉列表显示尚未分配的此类型的所有 Sercos 设备。
- 从列表中选择所需的设备。

注意： 您可以使用列表底部的空行重置分配。

选择列表的每一行均包含 Sercos 设备的简短描述。

值用竖线 ("|") 分隔，并与下列参数相对应：
- TopologyAddress
- ObjectType
- SerialNumber
- ConfiguredApplicationType
- SercosAddress

您可通过两种方法采用参数值：
- 您可以通过单击 **<--** 按钮应用已分配设备的值。
- 您可以通过单击**采用所有已分配设备的值**应用所有已分配设备的值。

当应用这些值时，程序需要将已分配和扫描的 Sercos 设备的值写入控制器配置中的相关 Sercos 对象以便进行调试。

注意： 在应用了值后，相应的行将会以绿色高亮显示。

操作模式

操作模式用于确定 Sercos 设备的操作方式。

在**操作模式**列的下拉列表中选择所需的操作模式：

操作模式	描述
Virtual	Sercos 设备在物理意义上不存在。
Real	Sercos 设备在物理意义上必须存在。
Deactivated	Sercos 设备不在使用中。 它在物理意义上可能存在。
Optional	Sercos 在物理意义上可能存在，但这不是前提条件。

注意： 如果在按住 Shift 键的同时更改此列中的某个值，则会将此列中的所有值设置为该值。

手动添加设备

在**设备寻址**编辑器中，您可以手动将一部分新设备添加到控制器配置。

要手动添加设备，请执行以下步骤：

步骤	操作
1	输入所需的新设备数。
2	从列表中选择所需的设备： ● **LXM32S**：Lexium 32S 驱动器 ● **TM5NS31**：TM5 Sercos 接口模块
3	单击**添加**。 **结果：**已将设备添加到控制器配置。

注意： 如果在添加设备时检测到错误，则无法添加列出的设备，并会在消息窗口中显示相应的解释。

Lexium LXM32S 驱动器配置

描述

要访问设备编辑器屏幕，双击**设备树**中的驱动器节点：

LXM32S 驱动器的设备编辑器屏幕包含以下选项卡：

- **Sercos 循环数据交换**：
  - 驱动器的 Sercos 地址的配置。
  - 驱动器的运行模式 (参见第 *190* 页)的配置。
  - Sercos 隐式交换的配置（MDT 和 AT 报文的 IDN 配置）。
- **I/O 映射**：此选项卡可让您创建 IEC 变量并将其分配给选定的 IDN 以便进行循环交换。
- **配置**：驱动器的配置参数（使用 **Sercos 循环数据交换**选项卡配置驱动器参数）。
- **信息**：此选项卡显示与设备有关的一般信息（名称、描述、供应商、版本、图像）。

注意：
循环交换的缺省配置包含四个 IDN，且不可编辑：

- MDT 报文（控制器到驱动器）：
  - Position command value：S-0-0047.0.0
  - SPDSercos3Control：P-0-3025.0.80
- AT 报文（驱动器到控制器）：
  - Position feedback value：S-0-0051.0.0
  - SPDSercos3Status：P-0-3025.0.81

有关在 LXM32S 设备中实施 IDN 的详细信息，请参阅《Lexium LXM32S Product Manual》。

专用设置配置

专用设置选项可让您修改在控制器和驱动器之间循环交换的 Sercos IDN 的列表。

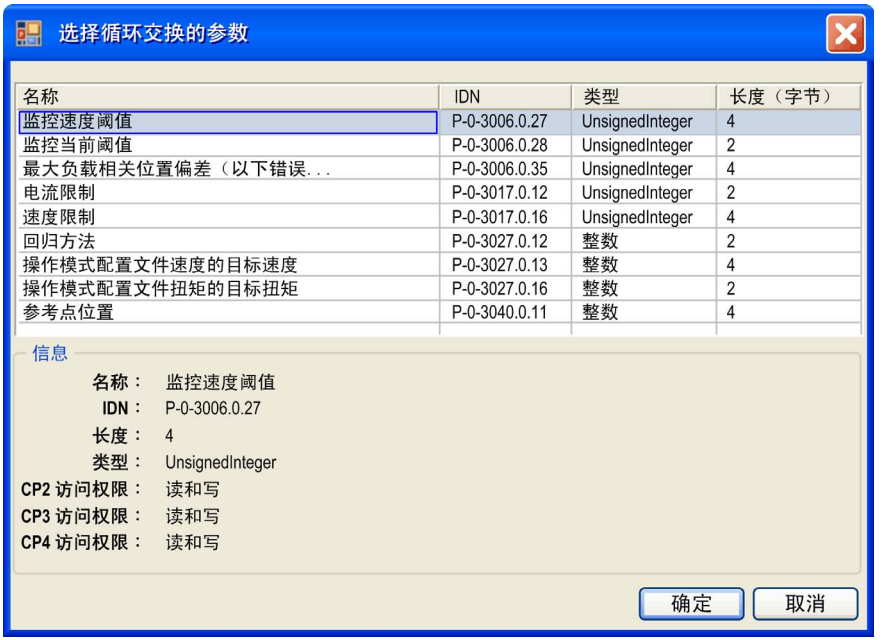
注意： MDT 和 AT 报文的总 IDN 长度限制为 48 个字节。

要激活专用设置，请选中**启用专用设置**复选框：

现在，可以使用下列按钮：

按钮	描述
+	单击此按钮可将 IDN 添加到列表（请参阅下文描述）。
–	选择列表中的 IDN，然后单击此按钮可从列表中删除该 IDN。
向上箭头	选择列表中的 IDN，然后单击此按钮可在列表中向上移动该 IDN。
向下箭头	选择列表中的 IDN，然后单击此按钮可在列表中向下移动该 IDN。

要将 IDN 添加到 MDT 或 AT 报文，请执行以下步骤：

步骤	操作
1	单击 **+**。 **结果：**将显示以下对话框： 
2	在列表中选择要添加的 IDN。
3	单击**确定**。 **结果：**已将 IDN 添加到 MDT 或 AT 报文，并更新所使用的字节数（**实时通道使用率**）。

TM5NS31 Sercos Interface Module

功耗

显示扩展模块的估计功耗：

步骤	操作
1	右键单击**设备树**的 **TM5NS31 Interface** 节点。
2	选择**功耗**。

注意： **功耗**功能表示的电流消耗数字基于假设值，而不是实际的电流测量值。输出的假设值基于典型负载，但可使用每个模块的 I/O 配置选项卡中的 24 Vdc I/O 段外部电流设置进行调整。输入信号的假设基于已知的内部负载，因此无法修改。当需要使用**功耗**功能来测试电源预算时，实际完整的系统测试和试运行是不可替代的。请参阅 TM5 / TM7 系统规划和安装指南。

Sercos 错误代码

Sercos 从站错误消息

通过 4 位数字代码来识别 Sercos 从站向主站报告的错误消息。以下是标准 Sercos 错误代码：

错误代码	描述	注释
0x0nnn	常规错误	–
0x0000	服务通道中没有错误	–
0x0001	服务通道未打开	–
0x0009	非法关闭服务通道	–
0x1nnn	元素 1	标识号
0x1001	不支持 IDN	–
0x1009	元素 1 访问无效	–
0x2nnn	元素 2	名称
0x2001	不支持名称	–
0x2002	名称传输时间过短	主站将“last transmission”设置得过早
0x2003	名称传输时间过短	主站未设置“last transmission”
0x2004	名称无法修改	名称为只读
0x2005	名称受到写保护	–
0x3nnn	元素 3	属性
0x3002	属性传输时间过短	主站将“last transmission”设置得过早
0x3003	属性传输时间过长	主站未设置“last transmission”
0x3004	属性无法修改	属性为只读
0x3005	属性受到写保护	–
0x4nnn	元素 4	单位
0x4001	不支持单位	–
0x4002	单位传输时间过短	主站将“last transmission”设置得过早
0x4003	单位传输时间过长	主站未设置“last transmission”
0x4004	单元无法修改	单位为只读
0x4005	单位受到写保护	–
0x5nnn	元素 5	最小输入值
0x5001	不支持最小输入值	–
0x5002	最小输入值传输时间过短	主站将“last transmission”设置得过早
0x5003	最小输入值传输时间过长	主站未设置“last transmission”
0x5004	最小输入值无法修改	最小输入值为只读
0x5005	最小输入值受到写保护	–

错误代码	描述	注释
0x6nnn	元素 6	最大输入值
0x6001	不支持最大输入值	–
0x6002	最大输入值传输时间过短	主站将“last transmission”设置得过早
0x6003	最大输入值传输时间过长	主站未设置“last transmission”
0x6004	最大输入值无法修改	最大输入值为只读
0x6005	最大输入值受到写保护	–
0x7nnn	元素 7	操作数据
0x7002	操作数据传输时间过短	主站将“last transmission”设置得过早
0x7003	操作数据传输时间过长	主站未设置“last transmission”
0x7004	操作数据无法修改	操作数据为只读
0x7005	操作数据在这个通讯阶段受到写保护	–
0x7006	操作数据小于最小输入值	–
0x7007	操作数据大于最大输入值	–
0x7008	操作数据无效	无效操作数据有可能是不支持的： ● 位号或位组合， ● 值、代码或 ● 已配置的 IDN
0x7009	操作数据受到密码的写保护	–
0x700A	操作数据受到写保护，采用循环配置	IDN 在 MDT 或 AT 中配置。因此，不允许经由服务通道写入
0x700B	间接寻址无效	例如，数据容器、列表处理等
0x700C	操作数据由于其他设置而受到写保护	例如，操作模式、子设备已启用、通讯版本设置等
0x700D	浮点数无效	–
700x0E	操作数据在参数化级别受到写保护	–
700x0F	操作数据在操作级别受到写保护	–
0x7010	程序命令已处于活动状态	–
0x7011	程序命令不可中断	–
0x7012	程序命令不可执行	例如，在这个阶段中，无法激活程序命令
0x7013	程序命令不可执行	相应参数无效
0x7014	所接收的当前列表参数长度与预期不符	–
0x7015	操作数据尚未完全创建	如果创建操作数据的时间较长，则稍后重试
0x71nn	经由 SVC 分段传输列表参数	–
0x7101	S-0-0394 中的 IDN 无效	–
0x7102	S-0-0397 中的空列表不允许写入访问	–
0x7103	所接管的列表段超过 S-0-0394 中列表的最大长度	–

错误代码	描述	注释
0x7104	只读：作为列表索引的列表段的长度超过 S-0-0394 中列表的当前长度	–
0x7105	S-0-0394 中的 IDN 受到写保护	–
0x7106	列表段中的操作数据小于最小输入值	–
0x7107	列表段中的操作数据大于最大输入值	–
0x7108	IDN S-0-0395 中的列表索引无效	–
0x7109	IDN S-0-0394 中的参数不包含变量长度	–
0x710A	IDN S-0-0397 不允许作为 S-0-0394 中的操作数据	–
0x8nnn	为主站内部错误代码保留	错误代码可由控制单元（比如，NC、PLC）的制造商定义
0xAnnn	保留	–
0xBnnn	保留	–
0xCnnn	为从站特定的错误代码保留	用于错误分析和追溯功能（故障排查）
0xDnnn	不经由 SVC 生成和传输错误代码	错误代码由 Sercos 的 TWG 定义
0xD000	没有错误	–
0xD001	服务通道（暂时）不可用	–
0xD002	服务通道被应用程序占用	–
0xD003	服务通道繁忙，从站正处理先前的请求	–
0xD004	Sercos 从站不可达	–
0xD005	服务通道事务已中止	–
0xD006	服务通道不支持写入这个元素	–
0xEnnn	为主站内部错误代码保留	–
0xFnnn	为主站内部错误代码保留	–

注意： 所有其他错误代码都被保留。

Sercos 主站错误消息

如果错误被 Sercos 主站识别，则这些错误会被指定为 5 位十六进制值或负的十进制值。

错误代码	描述
20001	SVC：在内部请求处于活动状态期间出现优先级更高的新请求
20002	SVC：在内部请求处于活动状态期间出现新的内部请求
20003	SVC：传输被优先级更高的另一功能调用取消
20004	SVC：请求了新的传输，但未设置 MBusy
20005	SVC：无效状态：AHS！= MHS（在已设置的 BusyAT 期间）

错误代码	描述
20006	SVC：由于从站未设置 BusyAT 标记而超时
20007	SVC：由于从站设置 BusyAT 标记的时间过长而超时
20008	SVC：不支持所写入的元素（允许 1 或 7）
20009	SVC：写入长度为零
-1	其他错误
-431	服务请求错误（例如，超时）
-445	服务超时
-467	内部状态机错误

第14章
串行线路配置

简介

本章介绍如何配置 Modicon LMC078 Motion Controller 的串行线路通讯。

本章包含了哪些内容？

本章包含了以下主题：

主题	页
串行线路配置	200
ASCII 管理器	202
SoMachine 网络管理器	204
Modbus Serial IOScanner	205
在 Modbus 串行 IOScanner 上添加设备	207
Modbus 管理器	214
将调制解调器添加到管理器	218

串行线路配置

简介

通过串行线路配置窗口可以配置串行线路的物理参数（波特率、奇偶校验等）。

串行线路配置

要配置串行线路，请双击**设备树**中的**串行线路**。

配置窗口随即显示，如下所示：

对于连接到端口的每个串行设备，以下参数必须相同。

元素	描述
波特率	传输速度（位/秒）
奇偶校验	用于错误检测
数据位	用于传输数据的位数
停止位	停止位的数目
物理介质	指定要使用的介质： ● RS485（是否使用极化电阻器） ● RS232
极化电阻器	控制器中集成了极化电阻器。通过此参数可将它们打开或关闭。

当使用新控制器或更新控制器固件时，控制器的串行线路端口在缺省情况下会针对 SoMachine 协议进行配置。SoMachine 协议与其他协议（如 Modbus 串行线路）不兼容。如果将新控制器连接到某个配置了 Modbus 的活动串行线路，或更新连接到该串行线路的控制器的固件，则可能会导致该串行线路上的其他设备停止通讯。在首次下载针对预期协议正确配置了相关端口的有效应用程序之前，请确保控制器未连接到活动 Modbus 串行线路网络。

注意

串行线路中断

在将控制器物理连接到正常运行的 Modbus 串行线路网络之前，请确保应用程序针对 Modbus 正确配置了串行线路端口。

不遵循上述说明可能导致设备损坏。

下表指出了管理器的最大波特率值：

管理器	最大波特率（位/秒）
SoMachine 网络管理器	115200
Modbus 管理器	
ASCII 管理器	
Modbus IOScanner	

ASCII 管理器

简介

ASCII 管理器用于通过简单设备传输和/或接收数据。

添加管理器

要将 ASCII 管理器添加到控制器，请在**硬件目录**中选择 **ASCII 管理器**，将其拖到**设备树**，然后将其放到一个突出显示的节点上。

有关将设备添加到项目的更多信息，请参阅：

- 使用拖放方法 *(参见 SoMachine, 编程指南)*
- 使用上下文菜单或加号按钮 *(参见 SoMachine, 编程指南)*

ASCII 管理器配置

要配置控制器的 ASCII 管理器，请双击**设备树**中的 **ASCII 管理器**。

ASCII 管理器配置窗口随即显示，如下所示：

如下表所述设置参数：

参数	描述
起始字符	如果为 0，则帧中不使用起始字符。否则，将在**接收模式**下使用相应的 ASCII 字符检测帧的开头。在**发送模式**下，此字符将添加到帧的开头。
第一个结束字符	如果为 0，则帧中不使用第一个结束字符。否则，将在**接收模式**下使用相应的 ASCII 字符检测帧的结尾。在**发送模式**下，此字符将添加到帧的结尾。
第二个结束字符	如果为 0，则帧中不使用第二个结束字符。否则，将在**接收模式**下使用相应的 ASCII 字符检测帧的结尾。在**发送模式**下，此字符将添加到帧的结尾。

参数	描述
收到的帧长度	如果为 0，则不使用此参数。此参数使系统可以在控制器收到指定的字符数后推断接收的帧结尾。 **注：**此参数不能与**帧收到超时（毫秒）**同时使用。
帧收到超时（毫秒）	如果为 0，则不使用此参数。使用此参数可以使系统在无收发时间达到指定毫秒数后，推断接收的帧的结束。
串行线路设置	串行线路配置窗口 (参见第 *200* 页)中指定的参数。

注意： 如果使用多个帧终止条件，则第一个为 TRUE 的条件会终止交换。

添加调制解调器

要向 ASCII 管理器添加调制解调器，请参阅将调制解调器添加到管理器 (参见第 *218* 页)。

SoMachine 网络管理器

简介

使用 SoMachine 网络管理器，可通过 SoMachine 软件协议与 XBTGT/XBTGK 高级面板交换变量，或者使用串行线路进行 SoMachine 编程。

添加管理器

要将 SoMachine 网络管理器添加到控制器，请在**硬件目录**中选择 **SoMachine – 网络管理器**，将其拖到**设备树**，然后将其放到一个突出显示的节点上。

有关将设备添加到项目的更多信息，请参阅：

- 使用拖放方法 *(参见 SoMachine, 编程指南)*
- 使用上下文菜单或加号按钮 *(参见 SoMachine, 编程指南)*

配置管理器

SoMachine 网络管理器不需要进行配置。

添加调制解调器

要向 SoMachine 网络管理器添加调制解调器，请参阅将调制解调器添加到管理器 (参见第 *218* 页)。

Modbus Serial IOScanner

简介

Modbus IOScanner 用于简化与 Modbus 从站设备的交换。

添加 Modbus IOScanner

要将 Modbus IOScanner 添加到串行线路，在**硬件目录**中选择 **Modbus_IOScanner**，然后将其拖动到**设备树**，并将其放到高亮显示的节点上。

有关将设备添加到项目的更多信息，请参阅：

- 使用拖放方法 *(参见 SoMachine, 编程指南)*
- 使用上下文菜单或加号按钮 *(参见 SoMachine, 编程指南)*

Modbus IOScanner 配置

要在串行线路上配置 Modbus IOScanner，请双击**设备树**中的 **Modbus IOScanner**。

配置窗口随即显示，如下所示：

如下表所述设置参数：

元素	描述
传输模式	指定要使用的传输模式： ● RTU：使用二进制编码和 CRC 错误校验（8 个数据位） ● ASCII：消息采用 ASCII 格式，LRC 错误校验（7 个数据位） 对于网络上的每个 Modbus 设备，此参数设置必须完全相同。
响应超时（毫秒）	交换中使用的超时。
帧间时间（毫秒）	延迟可减少总线上的数据冲突。 对于网络上的每个 Modbus 设备，此参数设置必须完全相同。

注意： 请勿在配置有 Modbus IOScanner 的串行线路上使用 PLCCommunication 库的功能块。这样会中断 Modbus IOScanner 交换。

总线循环任务选择

Modbus IOScanner 和设备在所选应用程序任务的每个循环时交换数据。

要选择此任务，请选择 **Modbus 主站 IO 映射**选项卡。配置窗口随即显示，如下所示：

总线循环任务参数可让您选择用于管理扫描器的应用程序任务：

- **使用父总线循环设置**：将扫描器与管理控制器的应用程序任务进行关联。
- **MAST**：将扫描器与 MAST 任务进行关联。
- 其他已有任务：您可以选择某一已有任务，并将其与扫描器进行关联。有关应用程序任务的详细信息，请参阅《SoMachine 编程指南》。

与扫描器相关的任务扫描时间不得少于 500 ms。

在 Modbus 串行 IOScanner 上添加设备

简介

本节介绍如何在 Modbus IOScanner 上添加设备。

在 Modbus IOScanner 上添加设备

要在 Modbus IOScanner 上添加设备，请在**硬件目录**中选择**一般 Modbus 从站**，将其拖动到**设备树**，然后将其放到**设备树**的 **Modbus_IOScanner** 节点上。

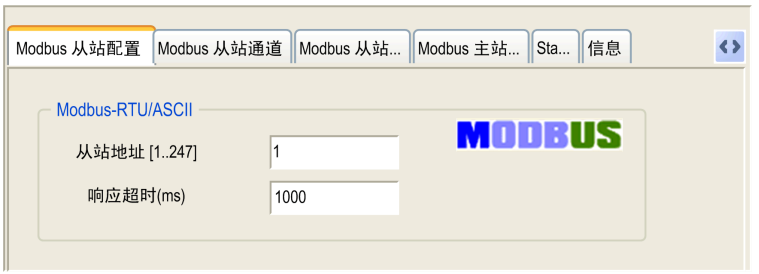
有关将设备添加到项目的更多信息，请参阅：

- 使用拖放方法 *(参见 SoMachine, 编程指南)*
- 使用上下文菜单或加号按钮 *(参见 SoMachine, 编程指南)*

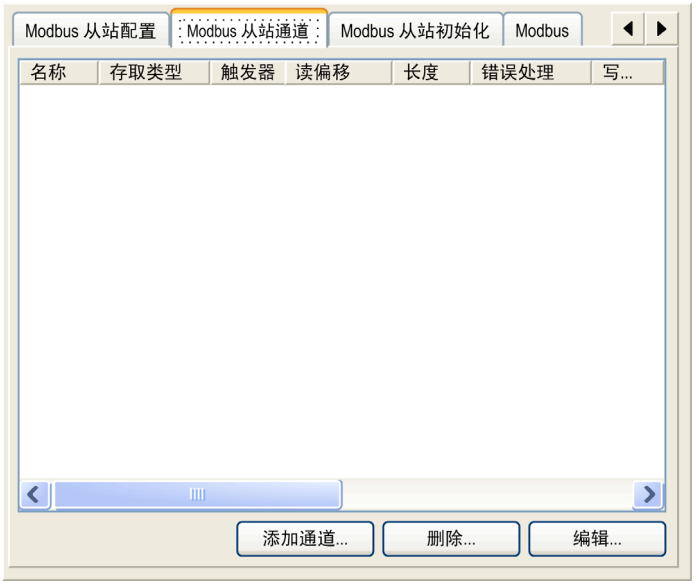
注意： 在 **Modbus 串行主站 I/O 映射** 选项卡的 `%IWx` 和 `%QWx` 中会自动创建用于交换的变量。

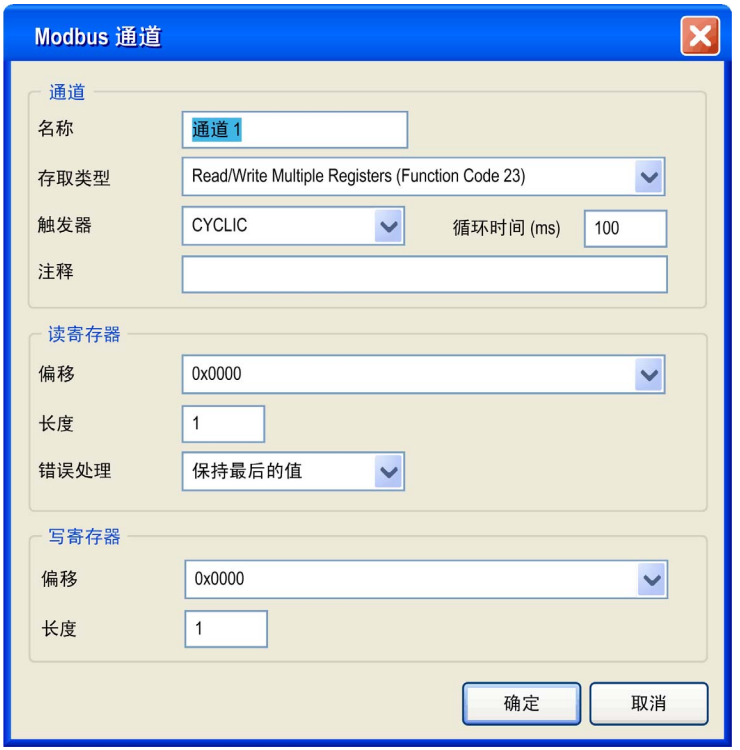
配置 Modbus IOScanner 上添加的设备

要配置 Modbus IOScanner 上添加的设备，请执行以下步骤：

步骤	操作
1	在**设备树**中，双击**一般 Modbus 从站**。 **结果：**将显示配置窗口。 
2	为您的设备输入**从站地址**值（从 1 到 247 中选择一个值）。
3	选择**响应超时**的值（以毫秒为单位）。

要配置 **Modbus 通道**，请执行以下步骤：

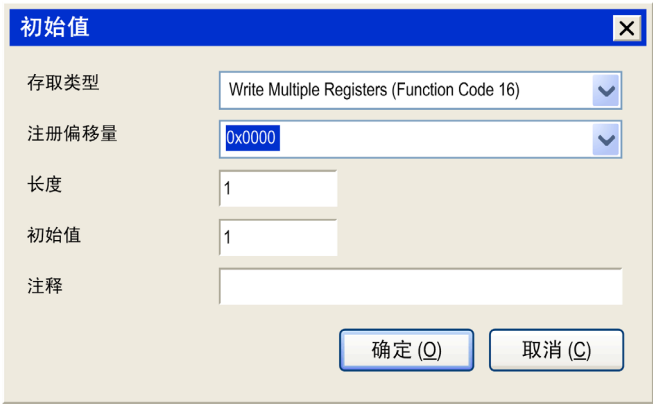
步骤	操作
1	单击 **Modbus 从站通道**选项卡： 

步骤	操作
2	单击**添加通道**按钮： 

步骤	操作
3	配置交换： 在**通道**字段中，可以添加下列值： ● **通道**：输入通道的名称。 ● **存取类型**：选择交换类型：读取、写入或读/写多个寄存器（即 %MW）(参见第 *213* 页)。 ● **触发器**：选择交换的触发器。该触发器可以使用在**循环时间(ms)** 字段中定义的周期进行**循环**，也可以通过某个布尔变量（随后会在 Modbus Master I/O Mapping 选项卡上创建此布尔变量）的**上升沿**来启动。 ● **注释**：添加有关此通道的注释。 在字段**读寄存器**（如果通道是“读取”通道或“读/写”通道）中，可以配置要在 Modbus 从站上读取的 %MW。它们将映射在 %IW 上（参见“Modbus 主站 I/O 映射”选项卡）： ● **偏移**：要读取的 %MW 的偏移。0 表示将读取的第一个对象会是 %MW0。 ● **长度**：要读取的 %MW 数量。例如，如果“偏移”= 2 且“长度”= 3，则通道将读取 %MW2，%MW3 和 %MW4。 ● **错误处理**：选择通讯中断时相关 %IW 的行为。 在字段**写寄存器**（如果通道是“写入”通道或“读/写”通道）中，可以配置要写入到 Modbus 从站的 %MW。它们将映射在 %QW 上（参见“Modbus 主站 I/O 映射”选项卡）： ● **偏移**：要写入的 %MW 的偏移。0 表示将写入的第一个对象会是 %MW0。 ● **长度**：要写入的 %MW 数量。例如，如果“偏移”= 2 且“长度”= 3，则通道将写入 %MW2，%MW3 和 %MW4。
5	单击**确定**确认此通道的配置。 **注意：** 您还可以： ● 单击**删除**按钮删除通道。 ● 单击**编辑**按钮更改通道的参数。

要配置 **Modbus 初始值**，请按照以下步骤操作：

步骤	操作
1	单击 **Modbus 从站初始化**选项卡：

步骤	操作
2	单击**新建**创建新的初始化值：  **初始化值**窗口包含以下参数： ● **访问类型**：选择交换类型：读取、写入或读/写多个寄存器（即 %MW）(参见第 *213* 页)。 ● **寄存器偏移**：要初始化的寄存器的寄存器编号。 ● **长度**：要读取的 %MW 数量。例如，如果“偏移”= 2 且“长度”= 3，则通道将读取 %MW2，%MW3 和 %MW4。 ● **初始值**：用于初始化寄存器的值。 ● **注释**：添加有关此通道的注释。
4	单击**确定**创建新的**初始值**。 **注意：** 您还可以： ● 单击**上移**，更改值在列表中的位置。 ● 单击**删除**，删除列表中的值。 ● 单击**编辑**，更改值的参数。

要配置 Modbus Master I/O Mapping，请按照以下步骤操作：

步骤	操作
1	单击 **Modbus Master I/O Mapping** 选项卡： Generic_Modbus_Slave Modbus 从站配置 | Modbus 从站通道 | Modbus 从站初始化 | Modbus Master I/O Mapping | 状态 | 信息 通道 变量 | 映射 | 通道 | 地址 | 类型 | 缺省值 | 单位 | 说明 通道 1 | %QX2.0 | BIT | 触发器变量 通道 1 | %IW1 | ARRAY [0..0] OF WORD | 读取保持寄存器 通道 2 | %IW2 | ARRAY [0..1] OF WORD | 读取/写入多个寄存器 通道 2 | %QW2 | ARRAY [0..0] OF WORD | 读取/写入多个寄存器
2	双击**变量**列的单元格，以打开文本字段。 输入变量的名称或单击浏览按钮 **[...]**，然后使用**输入助手**选择变量。
3	有关 有关详细信息，I/O 映射的 ，请参阅 SoMachine - 编程指南。

访问类型

下表描述了可用的各种访问类型：

功能	功能代码	可用性
Read Coils	1	Modbus 通道
Read Discrete Inputs	2	Modbus 通道
Read Holding Registers（通道配置的缺省设置）	3	Modbus 通道
Read Input Registers	4	Modbus 通道
Write Single Coil	5	Modbus 通道 初始化值
Write Single Register	6	Modbus 通道 初始化值
Write Multiple Coils	15	Modbus 通道 初始化值
Write Multiple Registers（从站初始化的缺省设置）	16	Modbus 通道 初始化值
Read/Write Multiple Registers	23	Modbus 通道

Modbus 管理器

简介

Modbus 管理器用于主站或从站模式下的 Modbus RTU 或 ASCII 协议。

添加管理器

要将 Modbus 管理器添加到控制器，请在**硬件目录**中选择 **Modbus 管理器**，将其拖到**设备树**，然后将其放到一个突出显示的节点上。

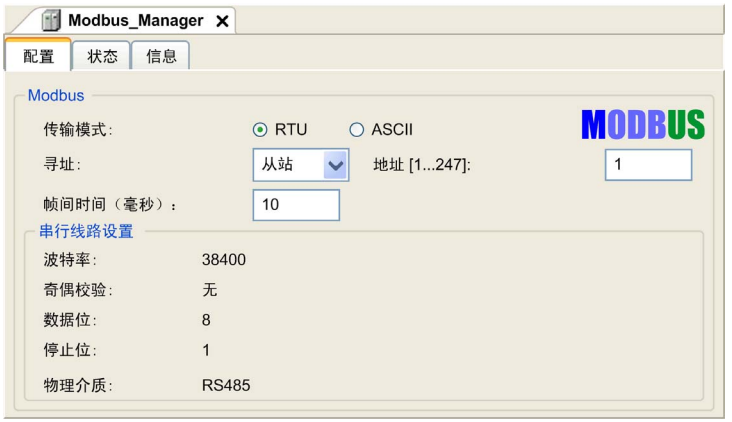
有关将设备添加到项目的更多信息，请参阅：

- 使用拖放方法 *(参见 SoMachine, 编程指南)*
- 使用上下文菜单或加号按钮 *(参见 SoMachine, 编程指南)*

Modbus 管理器配置

要配置控制器的 Modbus 管理器，请双击**设备树**中的 **Modbus 管理器**。

Modbus 管理器配置窗口随即显示，如下所示：



如下表所述设置参数：

元素	描述
传输模式	指定要使用的传输模式： ● RTU：使用二进制编码和 CRC 错误校验（8 个数据位） ● ASCII：消息采用 ASCII 格式，LRC 错误校验（7 个数据位） 对于链路上的每个 Modbus 设备，此参数设置必须完全相同。
寻址	指定设备类型： ● 主任务 ● 从站

元素	描述
地址	设备的 Modbus 地址（选择从站时）。
帧间时间（毫秒）	避免总线冲突的时间。 对于链路上的每个 Modbus 设备，此参数设置必须完全相同。
串行线路设置	串行线路配置窗口中指定的参数。

Modbus 主站

当将控制器配置为 Modbus 主站时，PLCCommunication 库支持以下功能块：

- ADDM
- READ_VAR
- SEND_RECV_MSG
- SINGLE_WRITE
- WRITE_READ_VAR
- WRITE_VAR

有关详细信息，请参阅 *(参见 SoMachine, Modbus 和 ASCII 读/写功能, PLCCommunication 库指南)* 库的PLCCommunicationPLCCommunication。

Modbus 从站

当控制器配置为 Modbus 从站时，支持以下 Modbus 请求：

功能代码 十进制（十六进制）	子功能 十进制（十六进制）	功能
1（十六进制 1）	–	读取数字量输出 (%Q)
2（十六进制 2）	–	读取数字量输入 (%I)
3（十六进制 3）	–	读取多个寄存器 (%MW)
5（十六进制 5）	–	写入单个线圈 (%M)
6（十六进制 6）	–	写入单个寄存器 (%MW)
8（十六进制 8）	–	诊断
15（十六进制的 F）	–	写入多个数字量输出 (%Q)
16（十六进制 10）	–	写入多个寄存器 (%MW)
23（17，十六进制）	–	读取/写入多个寄存器 (%MW)
43（十六进制的 2B）	14（十六进制的 E）	读取设备标识

下表包含诊断 Modbus 请求 08 支持的子功能代码：

子功能代码		功能
十进制	十六进制	
10	0 A	清除计数器和诊断寄存器
11	0B	返回总线消息计数
12	0C	返回总线通讯错误计数
13	0D	返回总线异常错误计数
14	0E	返回从站消息计数
15	0F	返回从站无响应计数
16	10	返回从站 NAK 计数
17	11	返回从站忙计数
18	12	返回总线字符溢出计数

下表列出了可以使用读取设备标识请求（基本标识级别）读取的对象：

对象 ID	对象名称	类型	值
00（十六进制）	供应商代码	ASCII 字符串	Schneider Electric
01（十六进制）	产品代码	ASCII 字符串	控制器参考号 LMC078CECS20T
02（十六进制）	主/次修订号	ASCII 字符串	aa.bb.cc.dd（与设备描述符相同）

下节描述控制器 Modbus 存储器映射与 HMI Modbus 映射之间的差异。如果不对应用程序进行编程以识别映射中的这些差异，则控制器和 HMI 将不会正确通讯。因此，可能会将不正确的值写入到负责输出操作的存储器区域。

⚠ 警告

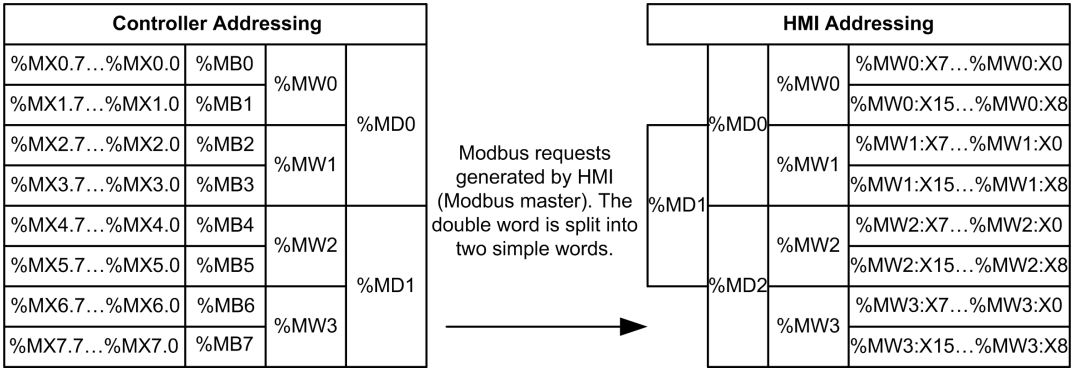
意外的设备操作

请将应用程序编程为可在控制器使用的 Modbus 存储器映射与任意附加 HMI 设备使用的 Modbus 存储器映射之间进行转换。

不遵循上述说明可能导致人员伤亡或设备损坏。

当控制器和 Magelis HMI 通过 Modbus 进行连接时（HMI 是 Modbus 请求的主站），数据交换使用简单字请求。

使用双字时，HMI 存储器的简单字上存在重叠，但是控制器存储器中不存在这种重叠（请参见下图）。为了使 HMI 存储器区域与控制器存储器区域相匹配，HMI 存储器的双字与控制器存储器的双字之比必须为 2。



下面提供了存储器双字匹配的示例：

- HMI 的 %MD2 存储器区域对应于控制器的 %MD1 存储器区域，因为 Modbus 请求使用相同的简单字。
- HMI 的 %MD20 存储器区域对应于控制器的 %MD10 存储器区域，因为 Modbus 请求使用相同的简单字。

下面提供了存储器位匹配的示例：

- HMI 的 %MW0:X9 存储器区域对应于控制器的 %MX1.1 存储器区域，因为控制器存储器中的简单字分为 2 个不同的字节。

添加调制解调器

要向 Modbus 管理器添加调制解调器，请参阅将调制解调器添加到管理器 (参见第 *218* 页)。

将调制解调器添加到管理器

简介

调制解调器可以添加到以下管理器：

- ASCII 管理器
- Modbus 管理器
- SoMachine 网络管理器

注意： 如果您需要使用 SoMachine 网络管理器连接调制解调器，请使用调制解调器 TDW-33（它执行 AT 和 A1 命令）。

将调制解调器添加到管理器

要将调制解调器添加到控制器，请在**硬件目录**中选择所需的调制解调器，将其拖到**设备树**，然后将其放到管理器节点上。

有关将设备添加到项目的更多信息，请参阅：

• 使用拖放方法 *(参见 SoMachine, 编程指南)*

• 使用上下文菜单或加号按钮 *(参见 SoMachine, 编程指南)*

有关详细信息，请参阅调制解调器库 *(参见 调制解调器功能：, 调制解调器库)*。

第15章
将 Modicon LMC078 Motion Controller 连接到 PC

将控制器连接到 PC

概述

要传输、运行和监视应用程序，请使用 USB 电缆或以太网连接将控制器连接到已安装 SoMachine 的计算机。

注意
设备无法操作 务必先将通讯电缆连接到 PC 之后再连接到控制器。 **不遵循上述说明可能导致设备损坏。**

USB Mini-B 端口连接

TCSXCNAMUM3P： 此 USB 电缆适用于持续时间较短的连接，如快速更新或检索数据值。
BMXXCAUSBH045： 接地并屏蔽后，此 USB 电缆适用于持续时间较长的连接。

注意： 在 PC 上同时只能连接 1 台控制器。

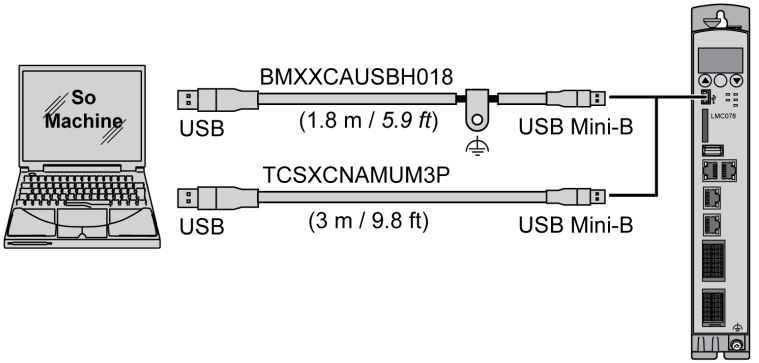
注意： LMC078 Motion Controller 必须在网关管理控制台中选择，可通过双击 Windows 通知区域中的**网关管理控制台**图标 进行访问。此选项在默认情况下未选择。

USB Mini-B 端口是编程端口，可以用于通过 SoMachine 软件将 PC 与 USB 主机端口连接。使用典型的 USB 电缆时，此连接适合用于程序的快速更新或持续时间较短的连接，以执行维护和检查数据值。如果不使用经过特殊调整的电缆以将电磁干扰的影响降到最低，则此连接不适合长期连接（如试运行或监控）。

⚠ 警告
意外的设备操作或设备无法操作 ● 必须使用 USB 屏蔽电缆（如 BMX XCAUSBH0），以稳固连接至系统的功能性接地 (FE) 进行长期连接。 ● 不要使用 USB 连接同时连接多个控制器。 ● 只有在确定工作区域是无危险区域的情况下，才能使用 USB 端口（若配有）。 **不遵循上述说明可能导致人员伤亡或设备损坏。**

首先应将通讯电缆连接到 PC，以最大程度减少影响控制器的静电释放可能性。

下图显示了连接 PC 的 USB 连接：



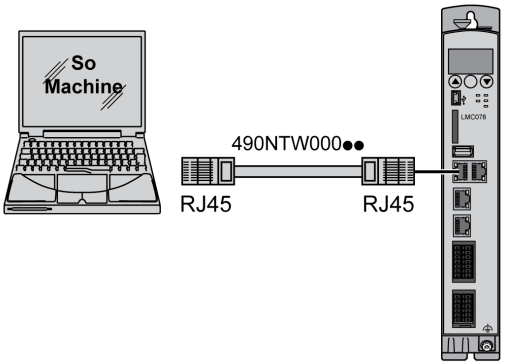
要将 USB 电缆连接到控制器，请执行以下操作：

步骤	操作
1	**1a** 如果使用电缆 BMXXCAUSBH045 或其他具有接地屏蔽连接的电缆建立长期连接，在将电缆连接到控制器和 PC 之前，请将屏蔽连接器牢固地连接到系统的功能性接地 (FE) 或保护性接地 (PE)。 **1b** 如果使用电缆 TCSXCNAMUM3P 或其他非接地 USB 电缆建立短期连接，请继续执行步骤 2。
2	将 USB 电缆连接器连接到 PC。
3	将 USB 电缆的 Mini-B 连接器连接到控制器 USB 连接器。

以太网端口连接

也可以使用以太网电缆将控制器连接到 PC。

下图显示了连接 PC 的以太网连接：



要将控制器连接到 PC，请执行下列操作：

步骤	操作
1	将以太网电缆连接到 PC。
2	将以太网电缆连接到控制器上的以太网端口。

注意： 缺省 IP 地址 (参见第 *154* 页)为：190.201.100.100。

第16章
固件更新

更新 Modicon LMC078 Motion Controller 固件

简介

Modicon LMC078 Motion Controller 的固件更新可从 http://www.schneider-electric.com 网站获取。

固件更新可通过使用**控制器助手**软件执行。

控制器助手提供了两种更新固件的方法：
- 第一种固件更新程序会自动删除控制器中的应用程序。
- 第二种固件更新程序不会删除控制器中的应用程序。

固件更新自动删除应用程序

执行固件更改可删除设备中的当前应用程序，其中包括 SD 卡中的启动应用程序。

注意
应用程序数据丢失 ● 在尝试固件更新之前需备份应用程序，将其备份到 PC 的硬盘。 ● 固件更新成功后，恢复设备的应用程序。 **不遵循上述说明可能导致设备损坏。**

如果移除设备电源，或者在应用程序的数据传输期间出现断电或通讯中断，则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。如果在固件更新过程中出现断电或通讯中断，或者如果使用了无效固件，则设备可能无法正常工作。在这种情况下，使用有效的固件并重新尝试固件更新。

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。 ● 如传输因任何原因中断，则重新开始传输。 ● 在文件传输成功完成之前，不得将设备（逻辑控制器、运动控制器或 HMI 控制器或驱动器）投入运行。 **不遵循上述说明可能导致设备损坏。**

启动 SoMachine Central，然后单击**维护** → **控制器助手**以打开**控制器助手**。

要执行控制器的完整固件更新，请执行以下步骤：

步骤	操作
1	在**主页**对话框上，单击**更新固件...** 按钮。 **结果**：将打开**更新固件**对话框。
2	按照 *SoMachine 控制器助手用户指南*的*更新固件*一章中所述步骤执行操作。

固件更新不会删除应用程序

启动 SoMachine Central，然后单击**维护** → **控制器助手**以打开**控制器助手**。

要执行控制器的完整固件更新而不替换引导应用程序和数据，请执行以下步骤：

步骤	操作
1	在**主页**对话框上，单击**管理映像...** 按钮。 **结果**：**管理映像**对话框打开。
2	单击**读取自...** 控制器按钮。 **结果**：将打开**控制器选择**对话框。
3	选择必需的连接类型和控制器，然后单击**读取**按钮。 **结果**：映像从控制器传输到计算机。 在成功完成此操作后，会将您重定向到**主页**对话框。
4	单击**新建/处理...** 按钮，然后单击**更新固件...**。 **结果**：将打开用于更新固件的对话框。
5	执行个别步骤以更新当前映像中的固件（更改仅在您的计算机上的映像中受影响）。 在最后一步，您可以创建通过控制器读取的映像的备份副本。 **结果**：在更新固件后，将打开**选择下一个操作**对话框。
6	在**选择下一个操作**对话框上，单击**在控制器上写入....** 按钮。 **结果**：将打开**控制器选择**对话框。
7	选择必需的连接类型和控制器，然后单击**写入**按钮。 **结果**：映像从您的计算机传输到控制器。 传输完成后，会将您重定向到**主页**对话框。

有关固件更新和使用固件创建新 SD 卡驱动器的详细信息，请参阅 *SoMachine 控制器助手用户指南*。

附录

概述

本附录列出了在技术上理解 Modicon LMC078 Motion Controller - 编程指南所必需的文档。

本附录包含了哪些内容？

本附录包含了以下章节：

章	章节标题	页
A	如何更改控制器的 IP 地址	227
B	诊断消息	231
C	LMC078 Sercos3 库	247
D	用于在用户程序中获取/设置串行线路配置的功能	273
E	控制器性能	279

附录 A
如何更改控制器的 IP 地址

changeIPAddress：更改控制器 IP 地址

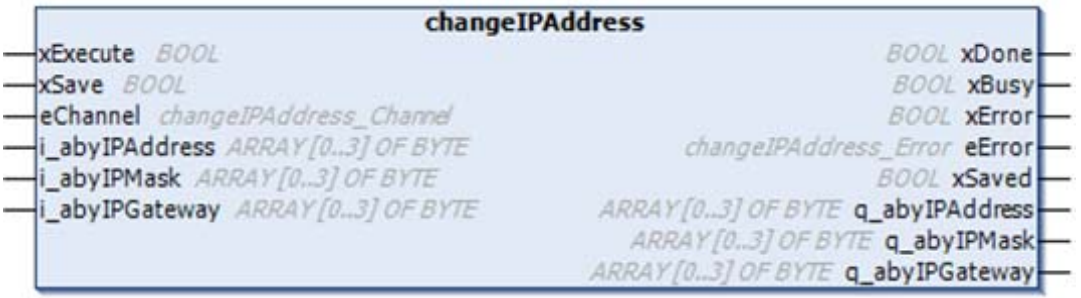
功能块描述

changeIPAddress 功能块提供动态更改控制器 IP 地址及其子网掩码和网关地址的功能。该功能块还能保存 IP 地址，以便用于控制器的后续重启。

注意： 只有将 IP 模式配置为**固定 IP 地址**，才能更改 IP 地址。有关详细信息，请参阅 IP 地址配置 (参见第 *151* 页)。

注意： 有关此功能块的详细信息，请使用 SoMachine 库管理器编辑器的**文档**选项卡。有关此编辑器的用法，请参阅 SoMachine - 编程指南。

图形表示形式



参数描述

输入	类型	注释
xExecute	BOOL	● 上升沿：操作开始。 ● 下降沿：复位输出。如果在功能块完成其操作前即出现下降沿，则输出将按通常方式进行操作，仅当操作完成或检测到错误时才复位。在这种情况下，会在恰好一个循环的时间内在输出上提供对应的输出值（xDone、xError、iError）。
xSave	BOOL	TRUE：保存配置以用于控制器的后续重启。
eChannel	changeIPAddress_Channel	输入 eChannel 是要配置的以太网端口。根据控制器上可用的端口数，它是 changeIPAddress_Channel 中的 2 个值 (参见第 *228* 页)之一（0 或 1）。

输入	类型	注释
i_abyIPAddress	ARRAY[0..3] OF BYTE	要配置的新 IP 地址。格式: 0.0.0.0. **注意：** 如果此输入设置为 0.0.0.0，则配置控制器默认 IP 地址 (参见第 *154* 页)。
i_abyIPMask	ARRAY[0..3] OF BYTE	新子网掩码。格式：0.0.0.0
i_abyIPGateway	ARRAY[0..3] OF BYTE	新网关 IP 地址。格式：0.0.0.0

输出	类型	注释
xDone	BOOL	TRUE：如果已成功配置 IP 地址，或者因为输入 i_abyIPAddress 设置为 0.0.0.0 而成功配置缺省 IP 地址。
xBusy	BOOL	功能块处于活动状态。
xError	BOOL	● TRUE：检测到错误，功能块中止操作。 ● FALSE：未检测到错误。
eError	changeIPAddress_Error	检测到的错误代码 (参见第 *229* 页)。
xSaved	BOOL	保存用于控制器后续重启的配置。
q_abyIPAddress	ARRAY[0..3] OF BYTE	当前控制器的 IP 地址。格式: 0.0.0.0.
q_abyIPMask	ARRAY[0..3] OF BYTE	当前子网掩码。格式: 0.0.0.0.
q_abyIPGateway	ARRAY[0..3] OF BYTE	当前网关 IP 地址。格式: 0.0.0.0.

changeIPAddress_Channel：要配置的以太网端口

changeIPAddress_Channel 枚举数据类型包含以下值：

枚举器	值	描述
CHANNEL_ETHERNET_NETWORK	0	M241、M251MESC、M258、LMC058、LMC078：**以太网端口** M251MESE：**Ethernet_2 端口**
CHANNEL_DEVICE_NETWORK	1	M241：**TM4ES4 以太网端口** M251MESE：**Ethernet_1 端口**

changeIPAddress_Error：错误代码

changeIPAddress_Error 枚举数据类型包含以下值：

枚举器	值	描述
ERR_NO_ERROR	00（十六进制）	未检测到错误。
ERR_UNKNOWN	01（十六进制）	检测到内部错误.
ERR_INVALID_MODE	02（十六进制）	IP 地址未配置为固定 IP 地址。
ERR_INVALID_IP	03（十六进制）	IP 地址无效。
ERR_DUPLICATE_IP	04（十六进制）	已在网络中使用新 IP 地址。
ERR_WRONG_CHANNEL	05（十六进制）	以太网通讯端口错误。
ERR_IP_BEING_SET	06（十六进制）	已经在更改 IP 地址。
ERR_SAVING	07（十六进制）	由于检测到错误或不存在非易失性存储器而未保存 IP 地址。

附录 B
诊断消息

简介

本章介绍 Modicon LMC078 Motion Controller 的诊断消息。

本章包含了哪些内容？

本章包含了以下主题：

主题	页
消息记录器	232
诊断消息	237

消息记录器

概述

消息记录器记录控制器上的重要事件。可以对此信息进行分类、评估并清楚显示。如果检测到错误，则此信息有助于解决问题或找到错误。

要打开**消息记录器**窗口，选择**工具树**，然后双击**消息记录器**节点：

消息记录器	编号	时间戳	类型	对象	实例	诊断代码
LMCxx8 ...	1	Mi 14.Jul 2010 12:06:28				
LMCxx8 ...	2	Mi 14.Jul 2010 12:06:50				
LMCxx8 ...	3	Mi 14.Jul 2010 12:07:01				
LMCxx8 ...	4	Mi 14.Jul 2010 12:07:10				
LMCxx8 ...	5	Mi 14.Jul 2010 12:08:02				

此数据视图显示消息记录器的信息。例如，您可以通过控制器连续调用和处理不同的消息记录器。您也可以添加之前保存的消息记录器文件。

还能够

- 将所显示的消息记录器另存为文件。
- 从列表中删除所显示的消息记录器。

此外，还能够

- 提供每一行及其注释。
- 使用不同颜色标记包含某些属性的所有行。

可通过上下文菜单直接使用这些选项（右键单击）。

所显示的时间与将项目添加到该对话框的时间相对应。按顺序连续编号。

您可以通过单击记录器左侧的加号 (+) 图标打开选定消息记录器。要关闭选定消息记录器，单击减号 (–) 图标：

Message logger

Message logger	No.	Timestamp		Type	Object	Instance	Diag. code	Ext. diagnosis	Message
LMCxx8 r...	1	Wed 02.Apr 2014	08:45:24						
	500	We Apr/02/2014	08:32:57...	1	LMCxx8	MyCon...	8003		Controller boot finished
	499	We Apr/02/2014	08:32:56...	1	LMCxx8	MyCon...	8051		LMC 078/AX=8/RAM=500/
	498	We Apr/02/2014	08:32:56...	1	SERC32	Sercos...	8042	CP4	Sercos phase switched
	497	We Apr/02/2014	08:32:56...	1	SERC32	Sercos...	8042	CP3/1 ms	Sercos phase switched
	496	We Apr/02/2014	08:32:55...	1	LMCxx8	MyCon...	8970	end	Fast Device Replacement
	495	We Apr/02/2014	08:32:55...	1	LMCxx8	MyCon...	8970	start	Fast Device Replacement
	494	We Apr/02/2014	08:32:55...	1	SERC32	Sercos...	8042	CP2/use=0	Sercos phase switched
	493	We Apr/02/2014	08:32:54...	1	SERC32	Sercos...	8042	CP1/scan=2	Sercos phase switched
	492	We Apr/02/2014	08:32:53...	1	SERC32	Sercos...	8042	CP0	Sercos phase switched
	491	We Apr/02/2014	08:32:52...	1	LMCxx8	MyCon...	8002	V01.51.03.03	Controller boot started
	490	We Apr/02/2014	08:32:14...	1	OBJVER		8016		Controller reset
	489	We Apr/02/2014	08:31:52...	1	LMCxx8	MyCon...	8003		Controller boot finished
	488	We Apr/02/2014	08:31:51...	1	LMCxx8	MyCon...	8051		LMC 078/AX=8/RAM=500/
	487	We Apr/02/2014	08:31:51...	1	SERC32	Sercos...	8042	CP4	Sercos phase switched
	486	We Apr/02/2014	08:31:51...	1	SERC32	Sercos...	8042	CP3/1 ms	Sercos phase switched
	485	We Apr/02/2014	08:31:50...	1	LMCxx8	MyCon...	8970	end	Fast Device Replacement
	484	We Apr/02/2014	08:31:50...	1	LMCxx8	MyCon...	8970	start	Fast Device Replacement
	483	We Apr/02/2014	08:31:50...	1	SERC32	Sercos...	8042	CP2/use=0	Sercos phase switched
	482	We Apr/02/2014	08:31:49...	1	SERC32	Sercos...	8042	CP1/scan=2	Sercos phase switched
	481	We Apr/02/2014	08:31:48...	1	SERC32	Sercos...	8042	CP0	Sercos phase switched
	480	We Apr/02/2014	08:31:47...	1	LMCxx8	MyCon...	8002	V01.51.03.03	Controller boot started

有些条目以颜色高亮显示。这有助于您在消息记录器中找到包含可比较属性的行。您可以使用上下文菜单进行调整。在保存或发送，然后打开文件后，将会保留最后选定的颜色设置。

编辑注释

从上下文菜单中选择**编辑注释**。您可以为每个条目添加任何注释。通过这种方式，您可以添加其他信息，用于在服务需要时提供检测到的错误的概述，甚至在很长一段时间后。

复制

从上下文菜单中选择**复制**或按 Ctrl+C 键。您可以将选定行的内容复制到 Windows 剪贴板，然后将其粘贴到任何文本处理应用程序。

从控制器添加消息记录器

从上下文菜单中选择**从控制器添加消息记录器**。该操作将在数据视图中创建新的消息记录器。将创建带当前时间戳的新连续编号。

复位消息记录器

从上下文菜单中选择**复位消息记录器**。该操作在数据视图中本地创建一个空的消息记录器（不包含任何消息），并同时删除控制器上的消息记录器。将删除控制器的所有消息记录器条目。

将消息记录器保存到文件

从上下文菜单中选择**将消息记录器保存到文件**。将打开标准 Windows 对话框。输入文件名，并将选定消息记录器保存到任意目录。

从文件加载消息记录器

从上下文菜单中选择**从文件加载消息记录器**。将打开标准 Windows 对话框。使用此对话框选择保存在任意目录中的所需消息记录器。该操作将在数据视图中创建新的消息记录器。将创建不带当前时间戳的连续编号。

从控制器导入消息记录器

从上下文菜单中选择**从控制器导入消息记录器**。将打开标准 Windows 对话框。使用此对话框选择保存在控制器上的所需消息记录器。该操作将在数据视图中创建新的消息记录器。将创建不带当前时间戳的连续编号。

从列表删除消息记录器

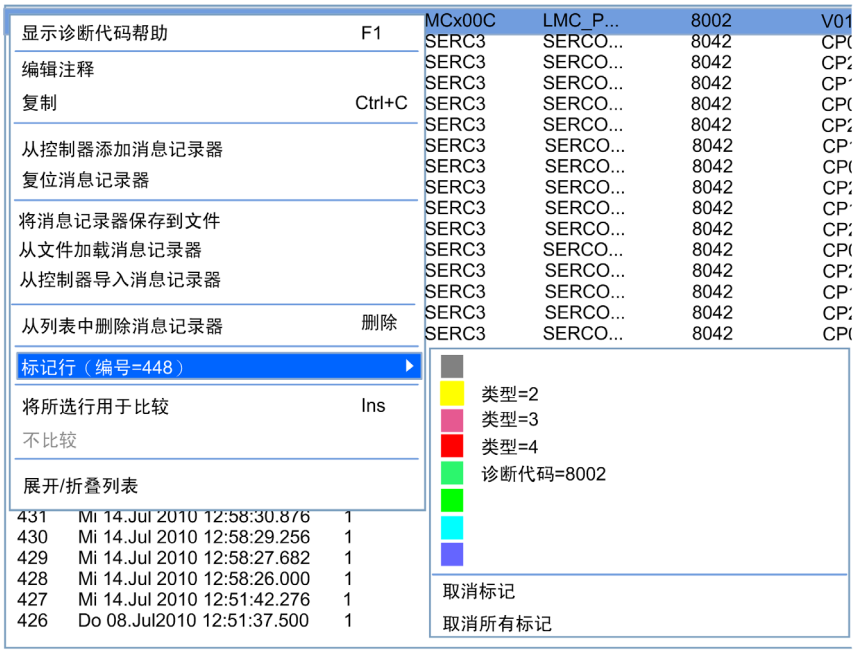
从上下文菜单中选择**从列表删除消息记录器**或按 **Delete** 键。这将会从列表中删除选定消息记录器。

您可以通过按 **Ctrl** 键选择多个消息记录器。

标记行

在消息记录器中，您可以标记包含使用不同颜色表示的某些共同属性的行（高亮显示）。例如，您可以使用粉色高亮显示**类型** 2 的所有行。或者，可以使用浅绿色高亮显示包含**诊断代码** = 8002 的所有行。

下面的上下文菜单列出了在上一段中所述的配置。在颜色高亮显示后，将会列出选择条件。



从上下文菜单中选择**标记行 (xxx=yyy)**。打开的子菜单已经显示了拥有其相应选择文本的一部分颜色条目。如示例所示，包含**类型=2** 和**诊断代码=8002**。

您可以定义自己的选择条件。在调用上下文菜单之前，选定文本由选定的最后一行来确定。继续执行下列步骤：

添加/更改颜色条目：

步骤	操作
1	右键单击包含所需选择条件的列（例如，列标题**诊断代码**和单元格值 8014）。 **结果**：将打开上下文菜单。
2	单击**标记行（诊断代码=8014）**。 **结果**：将打开包含颜色选择条件的子菜单。
3	选择高亮显示颜色，如蓝色。 **结果**：将以蓝色高亮显示包含选择条件（**诊断代码=8014**）的所有行。

删除颜色条目：

步骤	操作
1	右键单击消息记录器中包含想要删除的颜色的行。 **结果：**将打开上下文菜单。
2	单击**标记行**。 **结果：**将打开包含颜色选择条件的子菜单。
3	单击**取消标记**。 **结果：**将取消标记包含此颜色的所有高亮显示行。

删除所有颜色条目：

步骤	操作
1	右键单击消息记录器中包含想要删除的颜色的行。 **结果：**将打开上下文菜单。
2	单击**标记行**。 **结果：**将打开包含颜色选择条件的子菜单。
3	单击**取消所有标记**。 **结果：**将删除所有颜色标记。

注意： 可能会发生其中一行与多个有效的颜色选择条件相匹配的情况。例如，绿色和红色与**诊断代码=8014** 相对应。在这种情况下，所使用的颜色为在上下文菜单的下方列出的颜色。

注意： 如果已将颜色选择条件应用到多种颜色，则在取消标记时将首先删除具有最高优先级的标记。

将所选行用于比较

从上下文菜单中选择**将所选行用于比较**或按 Insert 键。因此，您可以使用**粗体**高亮显示所选行，以便于与其他行进行比较。如果您单击不同的消息，则所标记的此行的时间差将以粗体显示在状态栏中（例如，Time difference: 0.00:03:36.992 of 500(1) with regard to 481(1)）。

不比较

从上下文菜单中选择**不比较**可再次重置初始选择进行比较的行，并删除粗体标记。

展开/折叠列表

从上下文菜单中选择**展开/折叠列表**，可将所有消息记录器交替进行完全折叠或展开。

诊断消息

诊断消息类别

下表介绍了诊断消息类别：

诊断类别	名称	优先级
4	检测到错误导致完全停止	高
3	检测到错误导致单停止（如果该错误由轴触发）	-
2	建议	-
1	消息	低
0	已停用	无

诊断消息

下表列出了诊断消息及其类别：

诊断代码 (DiagCode)	诊断消息 (DiagMsg)	诊断类别 (DiagClass)
8001	Diagnostic acknowledgement	1
8002	Controller boot started	1
8003	Controller boot finished	1
8004	Program started	1
8005	Program automatic start active	1
8006	Program stopped	1
8007	Controller login	1
8008	Controller logout	1
8009	Program reset	1
8010	write file	1
8013	Controller connect to TCP/IP server	1
8014	Controller disconnect from TCP/IP server	1
8015	filesystem <ide0:> repaired	1
8016	Controller reset	1
8017	CANopen emergency message reset	1
8018	CANopen node guarding error resolved	1
8019	CANopen node error info	1
8020	Program cycle check has changed	1
8021	Program cycle check values are changed	1
8022	FC_SetTaskPriority() called	1

诊断代码 (DiagCode)	诊断消息 (DiagMsg)	诊断类别 (DiagClass)
8023	Controller shutdown	1
8027	File write open	1
8028	File write close	1
8029	UPS OK	1
8030	UPS active -no power	1
8031	UPS power supply OK	1
8032	UPS begin saving retain area	1
8033	UPS retain area saved	1
8034	UPS program tasks terminated	1
8035	UPS active -system shutdown started	1
8036	UPS controller rebooting started	1
8037	Battery low	2
8038	NvRam/RTC power outage detected	2
8042	SERCOS phase switched	1
8043	SERCOS detect configuration	1
8044	SERCOS firmware download	1
8045	File write error detected	1
8046	FPGA firmware download	1
8047	PIC firmware download	1
8048	BT-4 firmware download	1
8051	Controller type	1
8052	SERCOS extended diagnostic (MASTER)	1
8053	UPS active overtemperature	1
8054	Controller temperature out of range	2
8055	Controller message HW monitor	1
8056	Controller power supply low	1
8057	Program online change	1
8059	UPS active -IEC-control task running	1
8060	UPS changing state	1
8100	Motor overload	3
8101	Power stage overtemperature	3
8102	Motor overtemperature	3
8104	Control voltage out of range	3
8105	Encoder signal out of range	3

诊断代码 (DiagCode)	诊断消息 (DiagMsg)	诊断类别 (DiagClass)
8106	DC bus controller communication not possible	3
8107	Overcurrent	3
8108	DC bus overvoltage	3
8109	DC bus undervoltage	3
8110	Phase missing	3
8111	Shutdown due to tracking deviation	3
8112	SERCOS telegram invalid	3
8113	Braking resistor error detected	3
8114	Device type plate not readable	3
8116	Commutation error detected	3
8117	Motor type plate not readable	3
8119	Power stage short-circuit /ground error detection	3
8120	Power stage overload	3
8121	Braking resistor - overtemperature	3
8122	Shutdown due to velocity limit	3
8123	Safe Torque Off incorrect	3
8125	Motor load high	2
8126	Power stage temperature high	2
8127	Motor temperature high	2
8129	Power stage load high	2
8130	Temperature of braking resistor high	2
8132	Tracking deviation limit exceeded	2
8133	Speed-dependent current reduction	2
8134	External 24 Vdc low	2
8135	DC bus voltage low	2
8136	Safe Torque Off active	2
8137	Motorless	3
8138	Motor/Drive combination not supported	3
8139	DC bus precharge not possible	3
8140	Motor stop time limit exceeded	3
8142	Control board overtemperature	3
8143	Encoder temperature high	2
8144	DC bus short-circuit or ground error	3
8146	DC bus overload	3

诊断代码 (DiagCode)	诊断消息 (DiagMsg)	诊断类别 (DiagClass)
8153	DC bus discharge not possible	3
8154	Phase L1 missing	2
8155	Phase L2 missing	2
8156	Phase L3 missing	2
8157	DC bus load high	2
8159	DC bus discharge delayed	2
8161	Control board temperature high	2
8163	SERCOS Slave C1D error detected	3
8164	SERCOS C1D man.-specific error detected	3
8165	SERCOS Slave C2D advisory detected	2
8166	SERCOS C2Dman.specific advisory detected	2
8169	SERCOS Slave communication disturbance detected	2
8170	Encoder position not accessible	3
8171	Encoder communication disturbance detected	2
8172	Encoder extended diagnostic error detected	1
8173	Encoder error (track monitoring) detected	1
8177	Power board overtemperature	3
8178	Device internal error detected	3
8179	Braking resistor load high	2
8180	Power board temperature high	2
8181	Fan error detected	2
8182	External 24 Vdc power supply high	1
8183	Device fallback firmware active	3
8184	HW/SW combination not supported	3
8185	Device error detected	3
8186	DC bus voltage high	2
8204	Program cannot be loaded	3
8205	Impermissible parameter value	3
8209	Last boot unsuccessful	3
8300	Program divide by zero	3
8301	coprocessor segment overflow	3
8302	stack error detected	3
8303	general protection error detected	3
8304	coprocessor error detected	3

诊断代码 (DiagCode)	诊断消息 (DiagMsg)	诊断类别 (DiagClass)
8305	memory limit exceeded	3
8306	arithmetic overflow	3
8307	double execution error detected	3
8308	invalid task state segment	3
8309	no memory segment	3
8310	invalid memory segment adjustment	3
8311	coprocessor division error detected	3
8312	Parameter relocation unsuccessful	3
8313	excessive cycle time overrun	3
8316	NvRam data not valid	2
8317	Program cycle time overrun	2
8318	Program calculated profile deleted	3
8320	incorrect array access	3
8321	division by zero	3
8322	exception in IEC task	3
8323	string too long	3
8324	UPS error detected	3
8325	File corrupt	3
8326	Program function not supported	3
8327	CamTrack invalid Position Source	2
8328	CamTrack invalid Destination	2
8329	CamTrack invalid Bit number	2
8330	Program master job not executable	3
8331	Licensing	3
8332	Licensing	3
8333	EncoderNet receiving data not possible	3
8334	EncoderNet receiving data dist.detected	2
8335	EncoderNet synchronization not possible	3
8336	EncoderNet synchronization disturbance detected	2
8337	Parameter DynIECData value too high	3
8338	UPS battery not charged	3
8339	UPS active -system temperature high	3
8340	Data/parameter out of range	3
8341	CamTrack invalid Position type	2

诊断代码 (DiagCode)	诊断消息 (DiagMsg)	诊断类别 (DiagClass)
8400	Program diagnostic message class 0	0
8401	Program diagnostic message class 1	1
8402	Program diagnostic message class 2	2
8403	Program diagnostic message class 3	3
8404	Program diagnostic message class 4	4
8406	IEC diagnostic message class 1	1
8501	SERCOS slave not found	3
8502	SERCOS loop not closed	1
8503	SERCOS service channel error detected	3
8504	SERCOS read cycle overflow	3
8505	SERCOS Master communication disturbance detected	2
8506	SERCOS Master communication not possible	3
8507	SERCOS write cycle overflow	3
8508	SERCOS run-up not possible	3
8509	SERCOS slave SW not supported	3
8510	SERCOS Interrupt lost	3
8511	CPU time overflow	3
8512	SERCOS incorrect device type	3
8517	SERCOS addressing not unique	3
8518	SERCOS too many real slaves	3
8600	Master Encoder communication not possible	3
8601	Master Encoder signal out of range	3
8610	Async FB error	3
8611	Copied Async FB. Use a reference	3
8612	Async FB declared as retain/persistent	3
8613	异步工作超时错误	3
8700	CAN layer 2 driver error detected	3
8701	CAN layer2 initialization error detected	3
8702	CAN layer2 single error detected	3
8703	CAN layer2 errors reach advisory limit	3
8704	CAN layer2 switched passive	3
8705	CAN layer 2 system error detected	3
8706	CAN layer2 errors below advisory limit	1
8707	CAN layer2 switched active	1

诊断代码 (DiagCode)	诊断消息 (DiagMsg)	诊断类别 (DiagClass)
8710	communication error detected	3
8720	no module found	3
8722	no cyclic telegram	3
8723	no PROFIBUS config data	3
8725	firmware of the module was replaced	1
8726	firmware of the module is incorrect	3
8730	incorrect master parameter data	3
8131	automatic bus deactivation	3
8732	slave not responding	3
8733	unrecoverable bus error detected	3
8734	Bus short circuit detected	3
8735	reject bus telegrams	3
8736	no I/O data exchange with slave	3
8737	double IEC address assigned	3
8738	Configuration I/O data > permissible I/O area	3
8739	double PROFIBUS address assigned	3
8750	CANopen node does not exist	3
8751	CANopen node not configured	1
8752	no CANopen EDS file exists	3
8753	initialisation CANopen module unsuccessful	3
8754	CANopen Emergency Message	3
8755	CANopen node guarding error detected	3
8756	CANopen DPM access timeout	3
8757	CANopen configuration error detected	3
8758	Application object Size not supported	3
8759	Application object maximum count limit reached	3
8780	Encoder output frequence > 1MHz	3
8781	Master Encoder no connection	3
8782	Master Encoder signal out of range	3
8785	Hardware component error detected	3
8786	Asynchronous to SERCOS bus	3
8787	configuration error detected	3
8788	Wiring error detected	2
8789	PacNet communication disturbance detected	2

诊断代码 (DiagCode)	诊断消息 (DiagMsg)	诊断类别 (DiagClass)
8790	Module error detected	3
8791	TM5/TM7 module error detected on TM5NS31	3
8800	Insufficient working memory	3
8826	PIC update not possible	4
8827	Controller power-off/hardware monitor	3
8828	Library error detected	1
8903	Software error detected (class 3)	1
8904	Software error detected (class 4)	4
8905	FC_UserRefGeneratorStart not possible	4
8906	ControlMode invalid	3
8907	Encoder interface invalid	3
8908	Unintended motor reaction detected	3
8909	Motor type plate parameter invalid	3
8910	Reference value invalid	3
8910	Name too long	3
8957	SERCOS bus topology changed	2
8958	Encoder communication not possible	3
8959	Mains contactor error detected	3
8960	Invalid mains voltage mode setting	2
8961	Phase missing	2
8963	NRT IPAddr not in IPAddressRangeStatic	2
8964	NRT IPAddressRangeDynamic is insufficient	2
8965	NRT IP parameter read not possible	2
8966	NRT IP parameter write not possible	2
8967	NRT IP parameter device different	2
8968	NRT network overlapping detected	2
8969	Motor cable not connected	3
8970	Fast Device Replacement	1
8971	Fast Device Replacement not successful	2
8972	NRT gateway not in network	2
8973	Program download	1
8974	Brake voltage too low	3
8975	Motor commutation invalid	2
8976	Mains phases wiring not correct	3

诊断代码 (DiagCode)	诊断消息 (DiagMsg)	诊断类别 (DiagClass)
8977	Motor temp. monitoring disabled	2
8978	InverterEnableConfig invalid	3
8979	STO_A and STO_B different levels	3
8980	Braking resistor not connected	3
8981	Bootloader update	1
8982	Device state	1
8983	DC bus precharge active	1
8984	DC bus precharge complete	1
8990	Firmware update not possible	1
8991	Data transfer invalid	1
8992	Braking resistor short circuit	3
8993	Last device on SERCOS port	1
8994	Invalid ProducerCycleTime	3
8995	Update motor type plate	1
8996	Update motor type plate not successful	3
8997	Motor identification invalid	3
8998	TM5/TM7 supply voltage low	3
8999	TM5/TM7 supply voltage advisory	1

附录 C
LMC078 Sercos3 库

简介

本章介绍 LMC078 Sercos3 库。

本章包含了哪些内容？

本章包含了以下部分：

节	主题	页
C.1	数据类型	248
C.2	Sercos功能	254
C.3	异步 Sercos 功能块	265

第C.1节
数据类型

概述

本节介绍 LMC078 Sercos3 库中包含的数据类型。

本节包含了哪些内容？

本节包含了以下主题：

主题	页
ST_SercosConfiguration 数据类型	249
ST_SercosConfigurationDevice 数据类型	250
ET_Sercos3CmdType 数据类型	252
ET_Sercos3IDNType 数据类型	253

ST_SercosConfiguration 数据类型

简介

ST_SercosConfiguration 结构用于存储 Sercos 总线的一般值。

此数据类型可与以下功能结合使用：

- FC_SercosGetConfiguration
- FC_SercosScanConfiguration

数据类型结构

下表介绍了 ST_SercosConfiguration 结构的内容：

枚举	类型	描述
uiNumberOfEntries	UINT	在 Sercos 总线上扫描的 Sercos 设备数。
uiNumberOfPhysicalDevices	UINT	在 Sercos 总线上发现和在 SoMachine 项目中配置的 Sercos 设备数。
uiPhaseRunUpCount	UINT	对开始运行的 Sercos 阶段计数。
iCurrentPhase	INT	Sercos 总线正在运行的当前阶段。
astDevices	ARRAY [0...254]-ST_SercosConfigurationDevice	包含在 Sercos 总线上发现的 Sercos 设备的具体描述的数组。

ST_SercosConfigurationDevice 数据类型

简介

ST_SercosConfigurationDevice 结构用于存储在 Sercos 总线上发现的 Sercos 设备的具体描述。

此数据类型由 ST_SercosConfiguration 数据类型使用。

此数据类型可与以下功能结合使用：

- FC_SercosGetConfiguration
- FC_SercosScanConfiguration

数据类型结构

下表介绍了 ST_SercosConfigurationDevice 结构的内容：

枚举	类型	描述
stLogicalAddress	ST_Logical Address	Sercos 从站的逻辑地址。
uiVendorCode	UINT	设备的制造商代码。
sVendorDeviceId	STRING(40)	Sercos 设备标识符。
sName	STRING(80)	在控制器配置中指定给设备的名称。
sPowerSupply	STRING(40)	当前分配的电源。
udiWorkingMode	UDINT	设备的工作模式。
udiWorkingState	UDINT	当前活动模式。
udiIdentificationMode	UDINT	设备的识别模式。
uiTopologyAddress	UINT	设备在 Sercos 环中的物理位置。
uiConfiguredTopologyAddress	UINT	在 Sercos 总线中发现的拓扑地址，要将该地址分配给设备。
uiSercosAddress	UINT	已分配给设备的 Sercos 地址。 在 Sercos 总线启动时主站自动分配的 Sercos 地址（如果设备不在 IdentificationMode SercosAddress 中）。
uiConfiguredSercosAddress	UINT	配置的 Sercos 地址。
sSerialNumberController	STRING(80)	Sercos 总线中电机控制器的序列号。
sConfiguredSerialNumberController	STRING(80)	在 Sercos 总线中发现的电机控制器的序列号，要将该序列号分配给设备。
sSerialNumberMotor	STRING(40)	伺服电机的序列号。
sConfiguredSerialNumberMotor	STRING(40)	配置伺服电机的序列号。
sApplicationType	STRING(40)	应用类型。
sConfiguredApplicationType	STRING(40)	配置应用程序的类型。
sControllerType	STRING(40)	设备类型。

枚举	类型	描述
sMotorType	STRING(40)	电机的名称。
sFW_Version	STRING(40)	设备的固件版本。 如果控制器配置中的设备为虚拟（非真实）设备，则参数值为 <LMC version>.virtual。
sHW_Version	STRING(40)	设备的硬件版本。 如果控制器配置中的设备为虚拟（非真实）设备，则参数值为 <LMC version>.virtual。
sBootloaderVersion	STRING(40)	设备引导加载器的版本名称。
sFPGA_Version	STRING(40)	设备的 FPGA 版本。
sIPAddress	STRING(15)	设备的定义 IP 地址。
sSubnetmask	STRING(15)	设备的定义子网掩码。
sGateway	STRING(15)	设备的定义网关。
sMACAddress	STRING(17)	设备的 MAC 地址。

ET_Sercos3CmdType 数据类型

简介

ET_Sercos3CmdType 数据与以下功能块结合使用可设置 Sercos 命令：

- FB_SercosReadServiceDataAsync
- FB_SercosWriteServiceDataAsync

读取和写入命令使用条目 0...6，而程序命令则使用条目 7...10。

数据类型结构

下表介绍了 ET_Sercos3CmdType 枚举的内容：

枚举	类型	初始值	描述
ReadWriteName	WORD	0	读取或写入名称。
ReadWriteAttribute	WORD	1	读取或写入属性。
ReadWriteUnit	WORD	2	读取或写入单位。
ReadWriteMinValue	WORD	3	读取或写入最小值。
ReadWriteMaxValue	WORD	4	读取或写入最大值。
ReadWriteUserData	WORD	5	读取或写入用户数据。
ReadWriteVarLength	WORD	6	读取或写入变量长度。
ExecuteCommand	WORD	7	执行命令。
ExecuteCommandStart	WORD	8	开始执行命令。
ExecuteCommandCheck	WORD	9	验证已执行命令的状态。
ExecuteCommandStop	WORD	10	停止执行命令。

ET_Sercos3IDNType 数据类型

简介

ET_Sercos3IDNType 数据类型与以下功能块结合使用可设置 Sercos IDN 类型（标准或专有）：

- FB_SercosProcedureCommandAsync
- FB_SercosReadServiceDataAsync
- FB_SercosWriteServiceDataAsync

数据类型结构

下表介绍了 ET_Sercos3IDNType 枚举的内容：

枚举	类型	初始值	描述
ParameterType_P	WORD	1	专有 IDN (P-x-xxxx.x.x)。
ParameterType_S	WORD	2	标准 IDN (S-x-xxxx.x.x).

第C.2节
Sercos功能

概述

本节介绍 Sercos 功能。

本节包含了哪些内容？

本节包含了以下主题：

主题	页
FC_SercosGetConfiguration 功能	255
FC_SercosReadServiceData 功能	256
FC_SercosReadServiceDataByTopAddr 功能	258
FC_SercosScanConfiguration 功能	260
FC_SercosWriteServiceData 功能	261
FC_SercosWriteServiceDataByTopAddr 功能	263

FC_SercosGetConfiguration 功能

功能描述

FC_SercosGetConfiguration 功能用于获取甚至尚未配置但已连接到 Sercos 总线的所有设备的列表。

注意： Sercos 总线至少必须有一次到达阶段 2。

该功能将返回在更改至阶段 2 结束时所确定的 Sercos 设备。

在结构中返回扫描的设备。如果 Sercos 总线从未到达阶段 2，则该功能返回一个错误。

图形表示形式

IL 和 ST 表示形式

要查看 IL 或 ST 语言的一般表示形式，请参阅*功能和功能块表示形式*。

I/O 变量描述

下表介绍了输入变量：

输入	类型	注释
iq_stSercosConfiguration	ST_SercosConfiguration	用于获取 Sercos 总线配置的结构。

下表描述了输出变量：

输出	类型	注释
FC_SercosGetConfiguration	DINT	请参阅下面的返回值描述表。

下表介绍了返回值：

值	描述
0	该功能已成功执行。
-1	检测到错误。
-2	Sercos 总线从未到达阶段 2。

FC_SercosReadServiceData 功能

功能描述

FC_SercosReadServiceData 功能通过 Sercos 读取用于调试的服务数据。

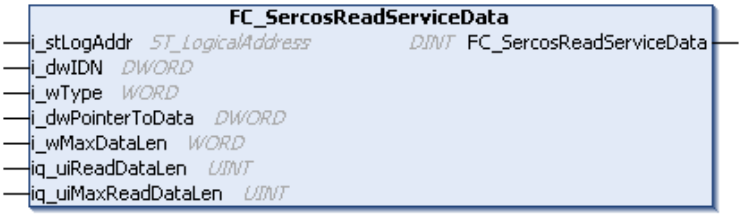
⚠ 警告

意外的设备操作

只有在咨询 Schneider Electric 应用部门后，才可以使用此功能。

不遵循上述说明可能导致人员伤亡或设备损坏。

图形表示形式



IL 和 ST 表示形式

要查看 IL 或 ST 语言的一般表示形式，请参阅*功能和功能块表示形式*。

I/O 变量描述

下表介绍输入变量：

输入	类型	注释
i_stLogAddr	ST_LogicalAddress	Sercos 设备的逻辑地址。
i_dwIDN	DWORD	数据服务 IDN。
i_wType	WORD	数据服务类型： ● 0：读取名称 ● 1：读取属性 ● 2：读取单位 ● 3：读取最小值 ● 4：读取最大值 ● 5：读取用户数据 ● 6：读取用户数据和变量长度 ● 7：执行命令

输入	类型	注释
i_dwPointerToData	DWORD	指向数据存储器的指针，填充数据字节的实际长度。
i_wMaxDataLen	WORD	要读取的数据字节的实际长度。

下表描述了输入/输出变量：

输入	类型	注释
iq_uiReadDataLen	UINT	用于读出 Sercos 参数的数据长度的变量 (uwType=6)。
iq_uiMaxReadDataLen	UINT	用于读出 Sercos 参数的最大数据长度的变量 (uwType=6)。

下表描述了输出变量：

输出	类型	注释
FC_SercosReadServiceData	DINT	请参阅下面的返回值描述表。

下表介绍了返回值：

值	描述
0	该功能已成功执行。
-1	逻辑地址无效。
-431	在服务请求期间检测到错误（如超时）。
-445	服务超时。
-461	在当前阶段，不支持通过服务通道读取参数。
-462	寻址设备不支持 ServiceDataRead 功能。
-464	服务传输无效。

FC_SercosReadServiceDataByTopAddr 功能

功能描述

FC_SercosReadServiceDataByTopAddr 功能通过 Sercos 服务通道从寻址设备读取服务数据。

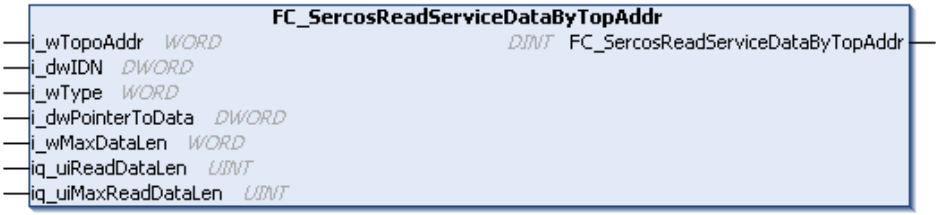
⚠ 警告

意外的设备操作

只有在咨询 Schneider Electric 应用部门后，才可以使用此功能。

不遵循上述说明可能导致人员伤亡或设备损坏。

图形表示形式



IL 和 ST 表示形式

要查看 IL 或 ST 语言的一般表示形式，请参阅*功能和功能块表示形式*。

I/O 变量描述

下表介绍输入变量：

输入	类型	注释
i_wTopoAddr	WORD	Sercos 设备的拓扑地址。
i_dwIDN	DWORD	数据服务 IDN。
i_wType	WORD	数据服务类型： ● 0：读取名称 ● 1：读取属性 ● 2：读取单位 ● 3：读取最小值 ● 4：读取最大值 ● 5：读取用户数据 ● 6：读取用户数据和变量长度 ● 7：执行命令

输入	类型	注释
i_dwPointerToData	DWORD	指向数据存储器的指针，填充数据字节的实际长度。
i_wMaxDataLen	WORD	要读取的数据字节的实际长度。

下表描述了输入/输出变量：

输入	类型	注释
iq_uiReadDataLen	UINT	用于读出 Sercos 参数的数据长度的变量 (uwType=6)。
iq_uiMaxReadDataLen	UINT	用于读出 Sercos 参数的最大数据长度的变量 (uwType=6)。

下表描述了输出变量：

输出	类型	注释
FC_SercosReadServiceDataByTopAddr	DINT	请参阅下面的返回值描述表。

下表介绍了返回值：

值	描述
0	该功能已成功执行。
-1	拓扑地址无效。
-431	在服务请求期间检测到错误（如超时）。
-445	服务超时。
-461	在当前阶段，不支持通过服务通道读取参数。
-462	寻址设备不支持 ServiceDataRead 功能。
-464	服务传输无效。

FC_SercosScanConfiguration 功能

功能描述

FC_SercosScanConfiguration 功能用于获取即使尚未配置但已连接到 Sercos 总线的所有设备的列表。

Sercos 总线通过此功能自动进行扫描。

如果 Sercos 总线位于阶段 2 以下的阶段，则该功能切换至阶段 2。

在阶段 2，对 Sercos 总线进行扫描。在扫描 Sercos 总线时，它可能会返回到初始阶段。

如果 Sercos 总线在阶段 2 或更高阶段，则返回扫描的设备。

在结构中返回扫描的设备。

图形表示形式

IL 和 ST 表示形式

要查看 IL 或 ST 语言的一般表示形式，请参阅*功能和功能块表示形式*。

I/O 变量描述

下表介绍了输入/输出参数：

输入	类型	注释
iq_stSercosConfiguration	ST_SercosConfiguration	用于获取 Sercos 总线配置的结构。

下表描述了输出变量：

输出	类型	注释
FC_SercosScanConfiguration	DINT	请参阅下面的返回值描述表。

下表介绍了返回值：

值	描述
0	该功能已成功执行。
-1	检测到错误。
-2	Sercos 总线无法到达阶段 2。

FC_SercosWriteServiceData 功能

功能描述

FC_SercosWriteServiceData 功能通过 Sercos 写入用于调试的服务数据。

⚠ 警告

意外的设备操作

只有在咨询 Schneider Electric 应用部门后，才可以使用此功能。

不遵循上述说明可能导致人员伤亡或设备损坏。

图形表示形式

IL 和 ST 表示形式

要查看 IL 或 ST 语言的一般表示形式，请参阅*功能和功能块表示形式*。

I/O 变量描述

下表介绍输入变量：

输入	类型	注释
i_stLogAddr	ST_LogicalAddress	Sercos 设备的逻辑地址。
i_dwIDN	DWORD	数据服务 IDN。
i_wType	WORD	数据服务类型： ● 0：写入名称 ● 1：写入属性 ● 2：写入单位 ● 3：写入最小值 ● 4：写入最大值 ● 5：写入用户数据 ● 6：写入用户数据和变量长度 ● 7：执行命令
i_dwPointerToData	DWORD	指向数据存储器的指针，填充数据字节的实际长度。
i_wMaxDataLen	WORD	要写入的数据字节的实际长度。

下表描述了输出变量：

输出	类型	注释
FC_SercosWriteServiceData	DINT	请参阅下面的返回值描述表。

下表介绍了返回值：

值	描述
0	该功能已成功执行。
-1	逻辑地址无效。
-431	在服务请求期间检测到错误（如超时）。
-445	服务超时。
-461	在当前阶段，不支持通过服务通道写入参数。
-462	寻址设备不支持 ServiceDataWrite 功能。
-464	服务传输无效。

FC_SercosWriteServiceDataByTopAddr 功能

功能描述

FC_SercosWriteServiceDataByTopAddr 功能通过 Sercos 服务通道将服务数据写入寻址设备。

⚠ 警告

意外的设备操作

只有在咨询 Schneider Electric 应用部门后，才可以使用此功能。

不遵循上述说明可能导致人员伤亡或设备损坏。

图形表示形式

IL 和 ST 表示形式

要查看 IL 或 ST 语言的一般表示形式，请参阅*功能和功能块表示形式*。

I/O 变量描述

下表介绍输入变量：

输入	类型	注释
i_wTopoAddr	WORD	Sercos 设备的拓扑地址。
i_dwIDN	DWORD	数据服务 IDN。
i_wType	WORD	数据服务类型： ● 0：写入名称 ● 1：写入属性 ● 2：写入单位 ● 3：写入最小值 ● 4：写入最大值 ● 5：写入用户数据 ● 6：写入用户数据和变量长度 ● 7：执行命令
i_dwPointerToData	DWORD	指向数据存储器的指针，填充数据字节的实际长度。
i_wMaxDataLen	WORD	要写入的数据字节的实际长度。

下表描述了输出变量：

输出	类型	注释
FC_SercosWriteServiceDataByTopAddr	DINT	请参阅下面的返回值描述表。

下表介绍了返回值：

值	描述
0	该功能已成功执行。
-1	拓扑地址无效。
-431	在服务请求期间检测到错误（如超时）。
-445	服务超时。
-461	在当前阶段，不支持通过服务通道写入参数。
-462	寻址设备不支持 ServiceDataWrite 功能。
-464	服务传输无效。

第C.3节
异步 Sercos 功能块

概述

本节介绍异步 Sercos 功能块。

这些功能块用于在 Sercos 服务通道上进行非循环通讯。

它们用来通过 Sercos 接口异步读取/写入设备参数并发送命令。

LXM32S 驱动器的所有 IDN 均可通过这些功能块进行访问。

本节包含了哪些内容？

本节包含了以下主题：

主题	页
FB_SercosReadServiceDataAsync：通过 Sercos 接口异步读取数据	266
FB_SercosWriteServiceDataAsync：通过 Sercos 接口异步写入数据	268
FB_SercosProcedureCommandAsync：通过 Sercos 接口异步发送命令	270

FB_SercosReadServiceDataAsync：通过 Sercos 接口异步读取数据

功能块描述

FB_SercosReadServiceDataAsync 功能块通过 Sercos 接口异步读取数据。

图形表示形式

IL 和 ST 表示形式

要查看 IL 或 ST 语言的一般表示形式，请参阅*功能和功能块表示形式*。

I/O 变量描述

下表介绍输入变量：

输入	类型	注释
i_xExecute	BOOL	如果为 TRUE，开始执行功能块。
i_ifSlave	SystemConfiguration.IF_IdentificationMandatory	用于描述设备的接口。它包含设备的名称及其逻辑编号。
i_uwIDN	WORD	设备的 IDN 编号。 如 IDN S-0-1027.0.1： i_uwIDN = 1027
i_etIDNType	ET_Sercos3IDNType (参见第 *253* 页)	IDN 类型（标准或专有）。 如 IDN S-0-1027.0.1： i_etIDNType = ET_Sercos3IDNType.ParameterType_S;
i_uwIDN_SI	WORD	IDN 结构实例。 如 IDN S-0-1027.0.1： i_uwIDN_SI = 0
i_uwIDN_SE	WORD	IDN 结构元素。 如 IDN S-0-1027.0.1： i_uwIDN_SE = 1
i_etCmdType	ET_Sercos3CmdType (参见第 *252* 页)	要执行的命令的类型。 该命令支持命令类型 0...6。
i_pBuffer	POINTER TO BYTE	指向长度 i_szSize 的缓冲器的指针。
i_wSize	WORD	缓冲器的长度。

下表介绍输出变量：

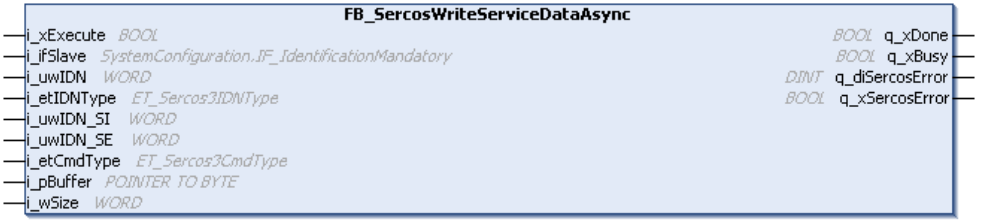
输出	类型	注释
q_xDone	BOOL	如果为 TRUE，表示功能块执行完成，且未检测到任何错误。
q_xBusy	BOOL	如果为 TRUE，表示功能块执行正在进行。
q_diSercosError（参见第 *195* 页）	DINT	同步读取命令返回的 Sercos 错误 (参见第 *195* 页)值。
q_xSercosError	BOOL	如果为 TRUE，表示检测到 Sercos 错误（负值）。
q_uiReadDataLen	WORD	所读取的数据项数。

FB_SercosWriteServiceDataAsync：通过 Sercos 接口异步写入数据

功能块描述

FB_SercosWriteServiceDataAsync 功能块通过 Sercos 接口异步写入数据。

图形表示形式



IL 和 ST 表示形式

要查看 IL 或 ST 语言的一般表示形式，请参阅*功能和功能块表示形式*。

I/O 变量描述

下表介绍输入变量：

输入	类型	注释
i_xExecute	BOOL	如果为 TRUE，开始执行功能块。
i_ifSlave	SystemConfiguration.IF_IdentificationMandatory	用于描述设备的接口。它包含设备的名称及其逻辑编号。
i_uwIDN	WORD	设备的 IDN 编号。 如 IDN S-0-1027.0.1： i_uwIDN = 1027
i_etIDNType	ET_Sercos3IDNType (参见第 *253* 页)	IDN 类型（标准或专有）。 如 IDN S-0-1027.0.1： i_etIDNType = ET_Sercos3IDNType.ParameterType_S;
i_uwIDN_SI	WORD	IDN 结构实例。 如 IDN S-0-1027.0.1： i_uwIDN_SI = 0
i_uwIDN_SE	WORD	IDN 结构元素。 如 IDN S-0-1027.0.1： i_uwIDN_SE = 1
i_etCmdType	ET_Sercos3CmdType (参见第 *252* 页)	要执行的命令的类型。 该命令支持命令类型 0...6。
i_pBuffer	POINTER TO BYTE	指向长度 i_szSize 的缓冲器的指针。
i_wSize	WORD	缓冲器的长度。

下表介绍输出变量：

输出	类型	注释
q_xDone	BOOL	如果为 TRUE，表示功能块执行完成，且未发生任何错误。
q_xBusy	BOOL	如果为 TRUE，表示功能块执行正在进行。
q_diSercosError （参见第 *195* 页）	DINT	同步读取命令返回的 Sercos 错误 (参见第 *195* 页)值。
q_xSercosError	BOOL	如果为 TRUE，表示检测到 Sercos 错误（负值）。

FB_SercosProcedureCommandAsync：通过 Sercos 接口异步发送命令

功能块描述

FB_SercosProcedureCommandAsync 功能块通过 Sercos 接口异步发送命令。

图形表示形式

IL 和 ST 表示形式

要查看 IL 或 ST 语言的一般表示形式，请参阅*功能和功能块表示形式*。

I/O 变量描述

下表介绍输入变量：

输入	类型	注释
i_xExecute	BOOL	如果为 TRUE，开始执行功能块。
i_xAbort	BOOL	如果为 TRUE，中止执行命令。 如果为 FALSE，按计划执行命令。
i_ifSlave	SystemConfiguration.IF_IdentificationMandatory	用于描述设备的接口。它包含设备的名称及其逻辑编号。
i_uwIDN	WORD	设备的 IDN 编号。 如 IDN S-0-1027.0.1： i_uwIDN = 1027
i_etIDNType	ET_Sercos3IDNType (参见第 *253* 页)	IDN 类型（标准或专有）。 如 IDN S-0-1027.0.1： i_etIDNType = ET_Sercos3IDNType.ParameterType_S;
i_uwIDN_SI	WORD	IDN 结构实例。 如 IDN S-0-1027.0.1： i_uwIDN_SI = 0
i_uwIDN_SE	WORD	IDN 结构元素。 如 IDN S-0-1027.0.1： i_uwIDN_SE = 1

下表介绍输出变量：

输出	类型	注释
q_xDone	BOOL	如果为 TRUE，表示功能块执行完成，且未发生任何错误。
q_xBusy	BOOL	如果为 TRUE，表示功能块执行正在进行。
q_diSercosError（参见第 *195* 页）	DINT	同步读取命令返回的 Sercos 错误 (参见第 *195* 页)值。
q_xSercosError	BOOL	如果为 TRUE，表示检测到 Sercos 错误（负值）。
q_uiProcCmdError	UINT	同步读取命令返回的程序命令错误值。
q_xProcCmdError	BOOL	如果为 TRUE，表示检测到程序命令错误。

附录 D
用于在用户程序中获取/设置串行线路配置的功能

概述

本节介绍用于在程序中获取/设置串行线路配置的功能。

要使用这些功能，请添加 **M2xx 通讯**库。

有关添加库的详细信息，请参阅 SoMachine - 编程指南。

本章包含了哪些内容？

本章包含了以下主题：

主题	页
GetSerialConf：获取串行线路配置	274
SetSerialConf：更改串行线路配置	275
SERIAL_CONF：串行线路配置数据类型的结构	277

GetSerialConf：获取串行线路配置

功能描述

GetSerialConf 为特定串行线路的通讯端口返回配置参数。

图形表示形式

参数描述

输入	类型	注释
Link	LinkNumber *(参见 SoMachine, Modbus 和 ASCII 读/写功能, PLCCommunication 库指南)*	Link 是通讯端口号。
PointerToSerialConf	指向 SERIAL_CONF （参见第 *277* 页）的指针	PointerToSerialConf 是配置结构的地址（SERIAL_CONF 类型的变量），配置参数存储在该地址中。ADR 标准功能必须用于定义关联指针。（请参见下面的示例。）

输出	类型	注释
GetSerialConf	WORD	此功能返回： ● 0：返回配置参数 ● 255：未返回配置参数，原因是： ❍ 该功能不成功 ❍ 该功能正在使用中

示例

请参阅 示例。SetSerialConf（参见第 *276* 页）

SetSerialConf：更改串行线路配置

功能描述

SetSerialConf 用于更改串行线路配置。

图形表示形式

注意： 在执行编程时更改串行线路端口的配置，可能会中断与其他连接的设备正在进行的通讯。

⚠ 警告
意外的配置更改导致失去控制 将程序投入使用前，请验证并测试 SetSerialConf 功能的所有参数。 **不遵循上述说明可能导致人员伤亡或设备损坏。**

参数描述

输入	类型	注释
Link	LinkNumber *（参见 SoMachine, Modbus 和 ASCII 读/写功能, PLCCommunication 库指南）*	LinkNumber 是通讯端口号。
PointerToSerialConf	指向 SERIAL_CONF（参见第 *277* 页）的指针	PointerToSerialConf 是配置结构的地址（SERIAL_CONF 类型的变量），新的配置参数存储在该地址中。ADR 标准功能必须用于定义关联指针。（请参见下面的示例。）如果为 0，请将应用程序缺省配置设置为串行线路。

输出	类型	注释
SetSerialConf	WORD	此功能返回： ● 0：新配置已设置 ● 255：拒绝新配置，原因是： ❍ 该功能正在使用中 ❍ 输入参数无效

示例

```
VAR
    MySerialConf：SERIAL_CONF
    结果：WORD；
END_VAR

（*获取串行线路 1 的当前配置*）
GetSerialConf(1, ADR(MySerialConf));

（*更改为 modbus RTU 从站地址 9*）
MySerialConf.Protocol := 0;          （*Modbus RTU/Somachine 协议（在此例中，
CodesysCompliant 选择该协议）*）
MySerialConf.CodesysCompliant := 0; （*Modbus RTU*）
MySerialConf.address := 9;           （*将 modbus 地址设置为 9*）

（*重新配置串行线路 1*）
result := SetSerialConf(1, ADR(MySerialConf));
```

SERIAL_CONF：串行线路配置数据类型的结构

结构描述

结构中包含与串行线路端口有关的配置信息。SERIAL_CONF它包含以下这些变量：

变量	类型	描述
Bauds	DWORD	波特率
InterframeDelay	WORD	Modbus（RTU、ASCII）中 2 个帧之间的最短时间（毫秒）
FrameReceivedTimeout	WORD	在 ASCII 协议中， 使系统经过指定的无收发时间（毫秒）后，在接收时关闭帧结尾。FrameReceivedTimeout如果为 0，则不使用此参数。
FrameLengthReceived	WORD	在 ASCII 协议中， 使系统在控制器已经接收到指定字符数的情况下关闭帧结尾。FrameLengthReceived如果为 0，则不使用此参数。
Protocol	BYTE	Modbus RTU 或 SoMachine（请参见 CodesysCompliant）
		1: Modbus ASCII
		2: ASCII
Address	BYTE	Modbus 地址为 0 到 255（0 用于总站）
Parity	BYTE	0: 无
		1: 奇
		2: 偶
Rs485	BYTE	0: RS232
		1: RS485
ModPol（极化电阻器）	BYTE	0: 否
		1: 是
DataFormat	BYTE	7 位或 8 位
StopBit	BYTE	1: 1 个停止位
		2: 2 个停止位
CharFrameStart	BYTE	在 ASCII 协议中，0 表示帧中没有起始字符。否则，在接收模式下， 将使用相应的 ASCII 字符检测帧的开头。在发送模式下，此字符将添加到用户帧的开头。
CharFrameEnd1	BYTE	在 ASCII 协议中，0 表示帧中没有第二个结束字符。否则，在接收模式下，将使用相应的 ASCII 字符检测帧的结尾。在发送模式下，此字符将添加到用户帧的结尾。
CharFrameEnd2	BYTE	在 ASCII 协议中，0 表示帧中没有第二个结束字符。否则，在接收模式下，将使用相应的 ASCII 字符（和 CharFrameEnd1）检测帧的结尾。在发送模式下，此字符将添加到用户帧的结尾。
CodesysCompliant	BYTE	0: Modbus RTU
		1：SoMachine（当 Protocol = 0 时）
CodesysNetType	BYTE	未使用

附录 E
控制器性能

处理性能

简介

本章提供有关 LMC078 处理性能的信息。

逻辑处理

下表显示了各个逻辑指令的逻辑处理性能：

IL 指令类型	1000 条指令的持续时间
INT 的加/减/乘法	1 微秒
DINT 的加/减/乘法	1 微秒
REAL 的加/减/乘法	3 微秒
REAL 的除法	48 微秒
BOOLEAN 的运算，例如，状态:= 状态 + 值	2 微秒
LD INT + ST INT	1 微秒
LD DINT + ST DINT	1 微秒
LD REAL + ST REAL	9 微秒

通讯和系统处理时间

通讯处理时间因发送/接收的请求数而异。

事件的响应时间

下表中显示的响应时间是指从触发外部任务的输入的信号上升沿到此任务设置的输出跳变沿的时间。事件任务还在设置输出前处理 100 条 IL 指令：

最小值	典型值	最大值
120 微秒	126 微秒	140 微秒

术语

%IW

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

%QW

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

任务

一组段和子程序，MAST 任务为循环或周期性执行，FAST 任务为周期性执行。

任务具有优先级，并且链接到控制器的输入和输出。可以根据任务来刷新这些 I/O。

一个控制器可以有多个任务。

保留数据

在下一次电源接通或热启动时使用的值。该值在因控制器停电关闭或正常要求关闭控制器后仍然保留。

功能块

拥有一个或多个输入并返回一个或多个输出的编程单元。FBs 通过实例（具有专用名称和变量的功能块副本）进行调用，且每个实例在从一个调用到另一个调用会保持原有状态（输出和内部变量）。

示例：定时器、计数器

协议

一种用于控制和启用两个计算端点和设备之间的连接、通讯和数据传输的惯例或标准。

变量

由程序寻址和修改的存储器单元。

固件

表示构成控制器上操作系统的 BIOS、数据参数和编程指令。固件存储在控制器内的非易失性存储器上。

字符串

一系列 ASCII 字符的变量。

应用程序

包括配置数据、符号和文档的程序。

引导应用程序

（*引导应用程序*）包含应用程序的二进制文件。通常它存储在控制器中，使控制器启动用户生成的应用程序。

持久性数据

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

接触式探测器输入

接触式探测器输入是高级数字量输入。这些输入用于测量功能，而测量功能可精确检测与某个测量输入之间的相对位置。一旦激活某个接触式探测器功能，该功能将在系统中独立运行，与 IEC 程序无关。IEC 程序可以使用参数来检测测量功能的状态。此功能由硬件和软件提供支持。

控制器

自动化工业流程（也称为可编程逻辑控制器或可编程控制器）。

毫秒

（*毫秒*）

目标

在 EtherNet/IP 显式信息传递中，对起点设备发送的数据交换请求做出响应的设备。

另参见*起点*

程序

应用程序的组成部分，其中包括可以在逻辑控制器的存储器中安装的经过编译的源代码。

系统变量

用于提供控制器数据和诊断信息，并用来向控制器发送命令的变量。

网络

共享一个公用数据路径和通讯协议的各种互联设备系统。

节点

通讯网络上的可寻址设备。

起点

在 EtherNet/IP 显式消息传递中，发起与目标网络设备进行数据交换的设备，通常为逻辑控制器。

另参见*目标*

运行状况位

用于表示通道的通讯状态的变量。

连续功能图语言

一种基于功能块图语言的图形编程语言（IEC61131-3 标准的扩展），工作原理与流程图类似。但是，不可以使用网络并对图形元素进行任意定位，允许反馈回路。每个功能块的输入位于左侧，输出位于右侧。可以将功能块输出链接到其他功能块的输入以创建复合表达式。

配置

一个系统内硬件组件的布局和互连以及硬件和软件的参数，可决定系统的运行特性。

零跟踪

跟踪递增编码器（用作参考点，每转启用重新初始化），又称为 top 0 或 top Z。

ARP

（*地址解析协议*）用于将 IP 地址映射到 Ethernet（硬件）地址的以太网(MAC) IP 网络层协议。

AT

（*确认电报*）在 Sercos 总线上，通过 AT 电报将数据从从站发送到主站（反馈值）。

BOOL

（*布尔*）用于计算的基本数据类型。BOOL 变量可以是以下两个值之一：0 (FALSE) 或 1 (TRUE)。从 WORD 中抽取的位为 BOOL 类型，例如：%MW10.4 是编号为 10 的存储器 WORD 的五分之一位。

BOOTP

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

CANopen

一种开放式工业标准通讯协议和设备配置文件规范 (EN 50325-4)。

CFC

（*连续功能图*）一种基于功能块图语言的图形编程语言（IEC 61131-3 标准的扩展），工作原理与流程图类似。但是，不可以使用网络并对图形元素进行任意定位，允许反馈回路。每个功能块的输入位于左侧，输出位于右侧。可以将功能块输出链接到其他功能块的输入以创建复合表达式。

CIP

（*公共工业协议*）在网络应用层实施 CIP 后，该协议可以与其他基于 CIP 的网络进行无线通讯，而不考虑协议。例如，如果在 Ethernet TCP/IP 网络的应用层中实施 CIP，则可以创建 EtherNet/IP 环境。同样，如果在 CAN 网络的应用层中实施 CIP，则可以创建 DeviceNet 环境。在这种情况下，EtherNet/IP 网络上的设备可以通过 CIP 桥接器或路由器与 DeviceNet 网络上的设备进行通讯。

CRC

（*循环冗余检验*）用来确定通讯传输的有效性的方法。传输包含构成校验和的位域。发射器根据消息的内容来计算所使用消息的检验和。接收节点后，按照相同的方式再次计算该字段。如果两次 CRC 计算的值存在任何差异，则说明传输的消息与收到的消息不同。

DHCP

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

DINT

（*双精度整数类型*）以 32 位格式进行编码的整数。

DWORD

（*双字*）以 32 位格式进行编码的类型。

EDS

（*电子数据表*）例如，包含设备的属性（如参数和设置）的现场总线设备描述文件。

Ethernet

用于 LANs 的物理和数据链路层技术，也称为 IEEE 802.3。

EtherNet/IP

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

FB

（*功能块*）用于整合一组编程指令以执行特定和规范化操作（如速度控制、间隔控制或计数）的实用编程机制。功能块可以包含配置数据和一组内部或外部操作参数，通常是一个或多个数据输入和输出。

FE

（*功能性接地*）用于增强或以其他方式允许正常操作电敏感设备的公共接地连接（在北美地区也称为功能性接地）。

与保护性接地（保护性接地）相比，功能性接地连接可用于除防震保护以外的任何其他目的，并且通常可以承载电流。使用功能性接地连接的设备示例包括浪涌抑制器和电磁干扰滤波器、某些天线和测量仪器。

FTP

（*文件传输协议*）一种以客户端-服务器架构为构建基础的标准网络协议，用于通过基于 TCP/IP 的网络交换和操作文件，不考虑其大小。

I/O

（*输入/输出*）

ICMP

（*因特网控制消息协议*）报告检测到的错误，并提供与数据报处理有关的信息。

IEC

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

IEC 61131-3

工业自动化设备的 3 部分标准的第 3 部分。IEC 61131-3 针对控制器编程语言，并定义了两个图形编程语言和两个文本编程语言标准。图形编程语言既是梯形图语言又是功能块图语言。文本编程语言包括结构化文本和指令列表。

IL

（*指令列表*）以某种语言编写的程序，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）。

Input Assembly

程序集是网络设备与逻辑控制器之间交换的数据块。Input Assembly 通常包含来自从站或目标设备的状态信息，主站或发起者将读取这些信息。

INT

（*整数*）以 16 位格式进行编码的整数。

IP

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

LD

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

LED

（*发光二极管*）在低电平电荷时亮起的指示灯。

LINT

（*长整数*）以 64 位格式进行编码的整数（INT 的 4 倍或 DINT 的 2 倍）。

LRC

（*纵向冗余校验*）用来确定所传输和存储的数据正确性的错误检测方法。

LWORD

（*长字*）以 64 位格式进行编码的数据类型。

MAC 地址

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

MAST

通过其编程软件运行的处理器任务。MAST 任务有两个段：

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

MDT

（*主站数据电报*）在 Sercos 总线上，主站会在各传输周期中发送一次 MDT 电报，以将数据（命令值）传输到伺服驱动器（从站）。

MST

（*主站同步电报*）在 Sercos 总线上，主站会在各传输周期开始时广播 MST 电报，以同步周期计时。

OS

（*操作系统*）用于管理计算机硬件资源并为计算机程序提供公共服务的软件集合。

Output Assembly

程序集是网络设备与逻辑控制器之间交换的数据块。Output Assembly 通常包含主站或起点发送给从站或目标设备的命令。

PDO

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

PE

（*保护性接地*）通过以接地电位保持设备的任何暴露的导电表面以帮助避免触电危险的公共接地连接。为了避免可能出现电压降，在该导体上不允许电流流过（在北美地区也称为*保护性接地*，或在美国国家电气规范中称为设备接地导体。）

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。

Profibus DP

（*Profibus 分散外设*）一种使用基于屏蔽 2 线线路的电子网络或基于光缆的光纤网络的开放式总线系统。DP 传输允许控制器 CPU 和分布式 I/O 设备之间进行高速的循环式数据交换。

REAL

一种数据类型，可以将它定义为以 32 位格式进行编码的浮点数。

RPDO

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

RPI

*（请求的数据包时间间隔）*扫描器请求的循环数据交换之间的时间周期。EtherNet/IP 设备以扫描器分配给它们的 RPI 指定的速率发布数据，并以等于 RPI 的周期从扫描器接收消息请求。

RTC

（*实时时钟*）由电池供电可连续运转以显示当天时间和日历的时钟，即使在为延长电池使用寿命而未对控制器通电时也一样。

RTP

（*实时处理*）实时处理是最重要的系统任务。此任务负责在正确的时间执行所有实时任务。实时处理由 Sercos 实施总线周期触发。

SDO

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

Sercos

（*串行实时通讯系统*）用于与以下对象互连的数字控制总线：运动控制、驱动器、I/O、传感器和执行器（用于数字控制机器和系统）。这是标准化且开放的控制器到智能数字设备接口，旨在用于标准化闭合回路实时数据的高速串行通讯。

SFC

（*顺序功能图*）一种包括具有关联操作的步骤、具有相关联逻辑条件的转换，以及步骤和转换之间的定向链接的语言。（SFC 标准已在 IEC 848 中定义。符合 IEC 61131-3。）

SINT

（*有符号整数*）15 位值带加号。

ST

（*结构化文本*）一种包括复杂的语句和嵌套指令（如迭代循环、条件执行或功能）的语言。ST 符合 IEC 61131-3

TCP

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

TPDO

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

UDINT

（*无符号双粗度整数*）以 32 位格式进行编码的整数。

UDP

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

UINT

（*无符号整数*）以 16 位格式进行编码的整数。

WORD

一种以 16 位格式进行编码的类型。

索引

ASCII 管理器, *202*
CANopen 接口, *176*
changeIPAddress, *227*
    更改控制器 IP 地址, *227*
ET_Sercos3CmdType
    数据类型, *252*
ET_Sercos3IDNType
    数据类型, *253*
Ethernet
    changeIPAddress 功能块, *227*
EtherNet/IP
    非循环数据交换, *110*
FB_SercosProcedureCommandAsync
    异步发送 Sercos 命令, *270*
FB_SercosReadServiceDataAsync
    异步读取 Sercos 数据, *266*
FB_SercosWriteServiceDataAsync
    异步写入 Sercos 数据, *268*
FC_SercosGetConfiguration
    功能, *255*
FC_SercosReadServiceData
    功能, *256*
FC_SercosReadServiceDataByTopAddr
    功能, *258*
FC_SercosScanConfiguration
    功能, *260*
FC_SercosWriteServiceData
    功能, *261*
FC_SercosWriteServiceDataByTopAddr
    功能, *263*
FTP 客户端, *160*
FTP 服务器
    以太网, *158*
FTPRemoteFileHandling 库, *160*
GetSerialConf
    获取串行线路配置, *274*
IP 地址
    changeIPAddress, *227*
LMC078 Sercos3
    FB_SercosProcedureCommandAsync, *270*
    FB_SercosReadServiceDataAsync, *266*
    FB_SercosWriteServiceDataAsync, *268*
LXM32S 配置, *191*
M2•• 通讯
    GetSerialConf, *274*
    SetSerialConf, *275*
Modbus
    协议, *156*
Modbus Ioscanner, *205*
Modbus TCP 客户端/服务器
    以太网, *156*
Modbus 管理器, *214*
PROFIBUS DP
    非循环数据交换, *103*
Sercos
    FB_SercosProcedureCommandAsync, *270*
    FB_SercosReadServiceDataAsync, *266*
    FB_SercosWriteServiceDataAsync, *268*
    接口配置, *184*
    添加第三方设备, *187*
    添加设备, *187*
Sercos 扫描, *188*
Sercos 标准概述, *182*
SERIAL_CONF, *277*
SetSerialConf, *275*
    设置串行线路配置, *275*
ST_SercosConfiguration
    数据类型, *249*
ST_SercosConfigurationDevice
    数据类型, *250*
下载应用程序, *64*
串行线路
    ASCII 管理器, *202*
    GetSerialConf, *274*
    Modbus 管理器, *214*
    SetSerialConf, *275*

以太网
EtherNet/IP 适配器, *106*
FTP 服务器, *158*
Modbus TCP 从站设备, *161*
Modbus TCP 客户端/服务器, *156*
服务, *149*
任务
事件任务, *38*
外部事件任务, *38*
循环任务, *37*
状态任务, *39*
看门狗, *43*
类型, *37*
停止命令, *61*
冷复位, *62*
初始值复位, *63*
剩余变量, *66*
功能
FC_SercosGetConfiguration, *255*
FC_SercosReadServiceData, *256*
FC_SercosReadServiceDataByTopAddr, *258*
FC_SercosScanConfiguration, *260*
FC_SercosWriteServiceData, *261*
FC_SercosWriteServiceDataByTopAddr, *263*
协议, *149*
IP, *151*
Modbus, *156*
参数类型, *25*
固件更新, *223*
外部事件, *38*
存储器映射, *27*
嵌入式 I/O 配置, *82*
库, *21*
FTPRemoteFileHandling, *160*
控制器配置
PLC 设置, *79*
控制器选择, *77*
数据类型
ET_Sercos3CmdType, *252*
ET_Sercos3IDNType, *253*
ST_SercosConfiguration, *249*
ST_SercosConfigurationDevice, *250*
添加第三方 Sercos 设备, *187*
热复位, *62*
状态图, *50*
硬件初始化值, *59*
编码器配置, *89*
脚本命令
防火墙, *170*
脚本文件
语法规则, *173*
软件初始化值, *59*
输出强制, *59*
输出行为, *59*, *59*, *59*
运动
性能, *42*
编程要求, *41*
运动任务, *40*
运行命令, *61*
配置参数, *70*
重新启动, *63*
防火墙
脚本命令, *170*
配置, *169*
默认脚本文件, *169*