

TeSys Active

TeSys Tera Motor Management System

DTM Library 在线帮助指南

TeSys 为电机启动器提供了创新的互联解决方案。

DOCA0257ZH-01
11/2025



法律声明

本文档中提供的信息包含与产品/解决方案相关的一般说明、技术特性和/或建议。

本文档不应替代详细调研、或运营及场所特定的开发或平面示意图。它不用于判定产品/解决方案对于特定用户应用的适用性或可靠性。任何此类用户都有责任就相关特定应用场合或使用方面，对产品/解决方案执行或者由所选择的任何业内专家（集成师、规格指定者等）对产品/解决方案执行适当且全面的风险分析、评估和测试。

施耐德电气品牌以及本文档中涉及的施耐德电气及其附属公司的任何商标均是施耐德电气或其附属公司的财产。所有其他品牌均为其各自所有者的商标。

本文档及其内容受适用版权法保护，并且仅供参考使用。未经施耐德电气事先书面许可，不得出于任何目的，以任何形式或方式（电子、机械、影印、录制或其他方式）复制或传播本文档的任何部分。

对于将本文档 或其内容用作商业用途的行为，施耐德电气未授予任何权利或许可，但以“原样”为基础进行咨询的非独占个人许可除外。

对于本文档或其内容或其格式，施耐德电气有权随时修改或更新，恕不另行通知。

在适用法律允许的范围内，对于本文档信息内容中的任何错误或遗漏，以及对本文档内容的任何非预期使用或误用，施耐德电气及其附属公司不会承担任何责任或义务。

目录

安全信息.....	5
关于本文档.....	6
安全强化指南.....	9
简介.....	11
TeSys Master Range.....	12
TeSys Tera System.....	13
定义.....	15
TeSys Tera DTM Library 安装的前提条件.....	16
安装 SoMove.....	17
安装 Tesys Tera DTM Library.....	19
将 LTMT Main Unit 连接到 PC.....	20
故障排除.....	21
用户界面.....	22
启动 TesysTera DTM Library.....	23
用户界面描述.....	32
菜单栏.....	33
工具栏.....	38
状态栏.....	40
选项卡区域.....	42
My Device.....	45
参数列表.....	49
General Settings.....	52
Starter Function Setting.....	53
Protection Setting.....	53
Communication Setting.....	57
User Map.....	58
Modified Parameters.....	58
Settings.....	60
General.....	61
Communication.....	62
安全性.....	64
My Dashboard.....	65
Diagnostics.....	67
Monitoring.....	72
Start Curve.....	76
Custom Logic.....	78
FB Diagram.....	79
Logic Simulator.....	80
用户功能.....	81
固件更新.....	82
人机界面 (HMI) 语言更新.....	86
Factory Reset.....	88
Pin 码管理.....	89
Custom Logic Editor.....	92
Custom Logic Editor 介绍.....	93
使用 Custom Logic Editor.....	96
自定义逻辑程序的特性.....	98

自定义逻辑变量的定义	99
LTMT Main Unit 变量定义	100
CALL_EOM 命令描述	103
自定义逻辑语言	113
创建自定义逻辑程序	114
Custom Logic Editor 简介	115
Custom Logic Editor 用户界面	116
逻辑命令	119
逻辑命令	124
程序逻辑命令	125
布尔逻辑命令	127
寄存器逻辑命令	136
定时器逻辑命令	144
锁存逻辑命令	146
计数器逻辑命令	148
数学逻辑命令	150
自定义逻辑程序示例	152
如何检查定时器和乘法命令	153
如何创建真值表	154
功能块图语言	157
FBD 语言概述	158
FBD 编辑器简介	159
FBD 元素	161
启动器类型	162
计算功能块	164
Inputs Blocks	167
功能块	170
逻辑功能块	172
Outputs 功能块	173
使用 FBD 语言进行编程	176
插入 FBD 功能块	177
在功能块之间创建链接	178
FBD 功能块属性	180
FBD 资源管理	181
操作 FBD 功能块	182
选择功能块	183
删除或复制对象	183
FBD 编辑器显示选项	186
显示选项	187
工作区外观和图形选项	188
编译、仿真和传输程序	189
简介	190
LTMT Main Unit 逻辑仿真器	191
初始化和连接	193
在LTMT Main Unit 与 Custom Logic Editor 之间传输逻辑文件	194

安全信息

重要信息

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，提示用户潜在的危險，或者提醒注意有关阐明或简化某一过程的信息。



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



这是提醒注意安全的符号。它用于提醒您注意潜在的人身伤害风险。遵守此符号后面的安全说明，以免发生伤害或死亡事故。

▲ 危險

危險表示若不加以避免，将会导致严重人身伤害甚至死亡的危險情况。

▲ 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險情况。

▲ 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險情况。

注意

注意用于指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險。

关于本文档

文档范围

本在线帮助指南介绍用于TeSys Tera Motor Management System的TeSys™。

本在线帮助指南介绍了成功实现系统的四个关键部分：

- 安装 TeSys Tera DTM Library
- 配置设备参数
- 监控设备状态
- 维护和升级 TeSys Tera system

本在线帮助指南适用于以下 TeSys Tera DTM Library 用户：

- 设计工程师
- 系统集成成员
- 系统操作员
- 维护工程师

有效性说明

本文档适用于：

- SoMove™ 软件版本 v2.10.0
- TeSys Tera DTM Library 版本 V2.0.0

本文档中所述的一些功能的可用性取决于使用的协议和 TeSys Tera system 上安装的物理模块。

一般网络安全说明

近年来，随着联网机器和生产设备的数量日益增多，发生非法访问、数据泄露和运营中断等网络威胁的几率也相应增加。因此，您必须考虑众多可能的网络安全措施，以帮助保护资产和系统免于此类威胁。

为了有助于保持和保护 Schneider Electric 产品的安全，强烈建议您采取 Cybersecurity Best Practices 文档中所述的网络安全最佳做法。

Schneider Electric 还提供其他信息和帮助：

- 订阅 Schneider Electric 安全资讯。
- 访问 Cybersecurity Support Portal 网页，以：
 - 查看安全通知。
 - 报告漏洞和事件。
- 访问 Schneider Electric Cybersecurity and Data Protection Posture 网页，以：
 - 了解网络安全态势。
 - 在网络安全学院中了更详细地了解网络安全。
 - 深入了解 Schneider Electric 的网络安全服务。

产品相关的网络安全信息

请参阅 *TeSys Tera Motor Management System Cybersecurity Guide – DOCA0260EN*。

文档的可用语言

文档提供以下语言版本:

- 英语
- 中文
- 法语
- 德语
- 意大利语
- 韩语
- 西班牙语

相关文档

文档名称	描述	参考编号
TeSys Tera Motor Management System 目录	目录 : <ul style="list-style-type: none"> • 介绍了 TeSys Tera system • 包含 TeSys Tera 技术特性 	LVCATENTER
TeSys Tera Motor Management System 用户指南	它是主要用户指南，其中介绍了完整的 TeSys Tera system。其中还介绍了 LTMT main units、LTMTCT/LTMTCTV Sensor Module、LTMT expansion modules 和 LTMTCUF control operator unit 的主要功能。	DOCA0257ZH
TeSys Tera Motor Management System 安装指南	该指南介绍了 LTMT main units、LTMTCT/LTMTCTV Sensor Module、LTMT expansion modules 和 LTMTCUF control operator unit 的安装、调试和维护。	DOCA0356ZH
TeSys Tera Motor Management System Modbus RTU 通讯指南	该指南介绍了 LTMT main unit 的 Modbus 网络协议通讯。	DOCA0355ZH
TeSys Tera Motor Management System PROFIBUS DP 通讯指南	该指南介绍了 LTMT main unit 的 PROFIBUS DP 网络协议通讯。	DOCA0256ZH
TeSys Tera Motor Management System EtherNet/IP 通讯指南	该指南介绍了 LTMT main unit 的 EtherNet/IP 网络协议通讯。	DOCA0258ZH
TeSys Tera Motor Management System LTMTCUF Control Operator Unit 用户指南	该指南介绍了如何安装、配置和使用 LTMTCUF control operator unit。	DOCA0233ZH
TeSys Tera Motor Management System DTM Library 软件发行说明	该文档提供了有关 TeSys Tera DTM Library 软件的重要信息，并概述了新功能和增强功能。	DOCA0279ZH
TeSys Tera Motor Management System 固件发行说明	该文档提供了有关 TeSys Tera system 固件包版本的重要信息，并概述了新功能和增强功能。	DOCA0276ZH
TeSys Tera Motor Management System 网络安全指南	本指南为 TeSys Tera Motor Management System 提供网络安全方面的信息。本指南介绍如何保护您的运营技术网络或企业串行或 Ethernet 网络。	DOCA0260ZH

要在线查找文档，请访问 Schneider Electric 下载中心 (www.se.com/ww/en/download/)。

有关非包容性或非敏感术语的信息

作为一家负责任、具有包容性的公司，Schneider Electric 不断更新其包含非包容性或非敏感术语的沟通方式和产品。但是，尽管我们做了这些努力，我们的内容仍可能包含某些客户认为不合适的条款。

商标

QR Code 是 DENSO WAVE INCORPORATED 在日本和其他国家或地区的注册商标。

安全强化指南

简介

PC 可以运行各种应用程序，以增强控制环境中的安全。系统具有出厂默认设置，要求重新配置以与 Schneider Electric 的深度防御方法的设备加强建议保持一致。

下面的指南介绍了 Windows 操作系统中的流程。这些仅作为示例提供。您的操作系统和应用程序可能有不同的要求或操作流程。

禁用远程桌面协议

Schneider Electric 的深度防御方法建议包括禁用远程桌面协议 (RDP)，除非您的应用程序需要 RDP。

在 Windows 11 中，使用**设置 > 系统 > 远程桌面 > 禁用远程桌面**（切换到**关**），来禁用远程桌面协议 (RDP)。

更新安全策略

通过命令窗口中的 `gpupdate` 更新您系统中与 PC 相关的安全策略。有关详细信息，请参阅 `gpupdate` 上的 Microsoft 文档。

管理更新

部署前，使用 Microsoft **Windows 更新** 网页上的实用程序，更新所有 PC 操作系统。要在 Windows 中访问此工具，请选择**开始 > 所有程序 > Windows 更新**。

工作站保护

为了降低与工程工作站相关的安全风险，请启用内存利用设置，比如数据执行预防 (DEP) 和地址空间布局随机化 (ASLR)。这些安全设置可以通过使用 Windows 11 操作系统中的系统 Exploit Protection 设置来启用。有关详细信息，请参阅 Microsoft 安全功能网页。

强制使用安全密码

使用满足要求的强密码，包含大写字母、小写字母、数字和特殊字符。启用此功能可通过降低弱密码风险，有助于防止未经授权的访问。

使用非默认端口

更改协议（如 HTTPS、DWPS 和 Modbus TCP）的默认通信端口，可增加一层额外的安全性。

IP允许列表 (IP Allow List)

IP允许列表功能通过仅允许指定的IP地址，限制对系统的访问。这有助于防止未经授权的设备连接到系统，并确保只有受信任的来源能与系统通信。 TeSys Tera system. 要访问IP允许列表功能，请在标准Web服务器 (Standard Web Server)中导航至 **Security > IP Allow List > IP Allow List** 。

简介

此部分内容

TeSys Master Range	12
TeSys Tera System.....	13
定义	15
TeSys Tera DTM Library 安装的前提条件	16
安装 SoMove	17
安装 Tesys Tera DTM Library.....	19
将 LTMT Main Unit 连接到 PC	20
故障排除.....	21

TeSys Master Range

TeSys 是来自全球市场领导者的创新性电机控制、监测与管理一体化解决方案。
TeSys Master TeSys 提供互联互通、高效节能的产品和解决方案，用于电动机及电气负载的通断与保护，并符合全球所有主要电气标准。

TeSys Tera System

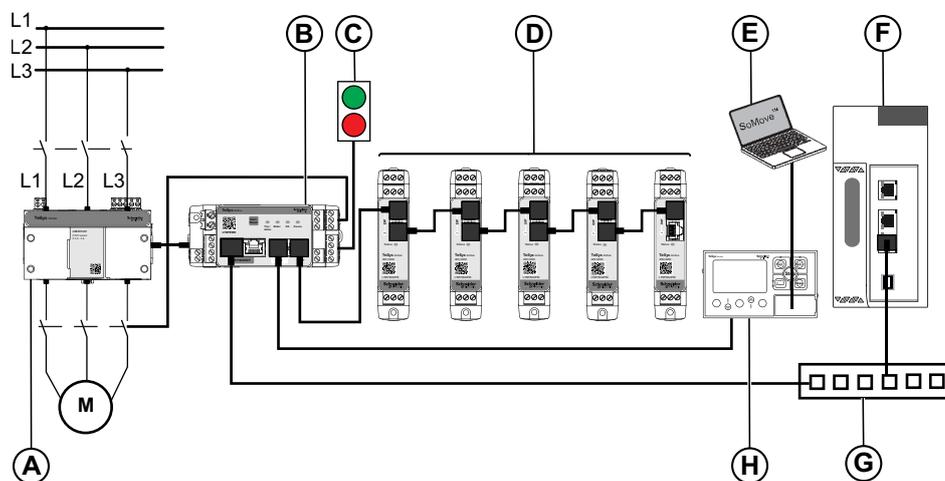
概述

TeSys Tera Motor Management System (或 TeSys Tera system)是 TeSys 智能继电器和电机启动器的Active系列产品的组成部分之一。TeSys Tera system 被设计作为智能电机控制中心 (iMCC) 的可靠基石，为单相或三相 AC 感应电机提供较完整的保护、测量、控制和监视功能。

TeSys Tera system 安装在低压开关设备系统中，通过现场总线网络和电机馈电器连接更高级别的自动化系统。

TeSys Tera system:

- 将 iMCC 馈电器中的常规和先进电机保护、测量和监控集成到带独立 HMI 设备的单一、易于配置且紧凑的通讯模块中。
- 为低压接触器控制的电机启动器馈电器提供保护控制器。
- 为低压应用中的恒速电机提供灵活且模块化的电机管理系统。



- A LTMTCT/LTMTCTV sensor module
- B LTMT main unit
- C 启动/停止命令
- D LTMT expansion modules
- E 运行 SoMoveFDT 容器软件且安装有 TeSys Tera DTM 的 PC。
- F 可编程逻辑控制器 (PLC) 或分布式控制系统 (DCS)
- G Ethernet 交换机
- H LTMTCUF control operator unit

功能特性

TeSys Tera system 管理：

- 带一体式传感器模块的最大电流可达 100 A 且工作电压为 690 V 的单相或三相 AC 感应电机和加热器。
- 带外部电流互感器的最大电流可达 810 A 且工作电压为 690 V 的单相或三相 AC 感应电机和加热器。
- 控制系统与电机馈电器之间的连接提高了工厂可用性。
- 为安装、调试、操作和维护节约了大量成本。
- 配备数字微处理器的控制器，可根据应用和过程要求设置电机参数。

定义

Device Type Manager

Device Type Manager (DTM) 是一个软件模块，托管在特定设备的 FDT container 中。

DTM 的功能包括：

- 安装设备库
- 更新设备库
- 扫描设备的各种现场总线
- 设备功率和电能监测
- 管理设备参数的配置
- 管理项目文件
- 自定义设备参数单位
- 故障排除
- 更新设备固件

SoMove Software 项目文件

SoMove 软件项目文件是预定设备的配置文件，可离线创建并保存，以供将来使用。

项目文件包含以下信息：

- 设备拓扑配置
- 所有参数设置

注：

- 该项目文件未包含定制程序（定制逻辑程序），需手动保存。
- 该文件保存的扩展名为 *.psx。

有关如何创建项目的详细说明，请参阅 *SoMove* 在线帮助。

TeSys Tera DTM Library 安装的前提条件

系统要求

TeSys Tera DTM Library 可安装在 Microsoft® Windows® 11操作系统上。

软件要求

TeSys Tera DTM Library 要求在 PC 上安装有以下软件：

- Microsoft .NET Framework v3.5
- SoMove software v2.10.0
- TeSys Tera DTM Library v2.0.0

硬件要求

安装 TeSys Tera DTM Library 时，建议 PC 满足以下硬件要求：

设备	最低要求	推荐要求
处理器	Pentium 4/Core 2 Duo , 2 GHz	Intel®Core™i3
RAM	2 GB	4 GB
显示	分辨率：1024 x 768、1366 x 768、1600 x 1900 和 1920 x 1080 像素	分辨率：1600 x 1200 和 1920 x 1080像素
	DPI：96 (100%) 和 120 (125%)	DPI：96 (100%) 和 120 (125%)
系统驱动器上的可用硬盘空间	1 GB	2 GB

安装 SoMove

概述

SoMove 软件是基于 Microsoft Windows® 的应用程序，采用开放式 FDT 或 DTM 技术。SoMove 软件包含多个 DTM。

下载 SoMove Software

SoMove 软件可从 Schneider Electric website 下载。

注: 您必须拥有管理员权限，才能在 PC 上下载、安装或卸载此软件。

安装 SoMove Software

以下步骤介绍如何安装 SoMove 软件：

1. 解压下载的文件 *SoMove_FDT*。
2. 该文件夹将包含名为 *SoMove_VX.Y.Z* (其中 X.Y.Z 是版本号) 的 .exe 文件和发行说明。
3. 双击 *SoMove_VX.Y.Z*，以启动安装。
4. 在 **Choose Setup Language** 对话框中，选择安装语言。
5. 选择 **OK**。
6. 在 **Welcome to the Installation Wizard for SoMove** 对话框中，选择 **Next**。
7. 如果出现 **Install Shield Wizard** 对话框，并通知您必须安装 Modbus 驱动程序，请选择 **Install**。

结果： 自动安装 Modbus 驱动程序。

8. 在 **Readme and Release Notes** 对话框中，请选择 **Next**。
9. 在 **License Agreement** 对话框中：
 - a. 仔细阅读许可证协议。
 - b. 选择 **I accept the terms in the license agreement** 选项。
 - c. 选择 **Next**。
10. 在 **Customer Information** 对话框中：
 - a. 在相应字段中输入以下信息：
 - **First name**
 - **Last name**
 - **Company name**
 - b. 选择 **Next**。
11. 在 **Destination Folder** 对话框中：
 - a. 如有必要，通过选择 **Change** 选项，来修改 SoMove 软件目标文件夹。
 - b. 选择 **Next**。
12. 在 **Shortcuts** 对话框中：
 - a. 如要在桌面上和/或快速启动栏中创建快捷方式，请选择相应的选项。
 - b. 选择 **Next**。

13. 在 **Ready to Install the Program** 对话框中，选择 **Install**。

结果：自动安装 SoMove 软件组件及以下库：

- 包含通讯协议的 Modbus 通讯 DTM Library。
- 包含不同目录的 DTM 库。
- SoMove 软件

14. 在 **Installation Wizard Completed** 对话框中，选择 **Finish**。

结果：SoMove 软件被安装到 PC 上。

安装 Tesys Tera DTM Library

概述

在 SoMove 软件中，存在 TeSys Tera system 的特定 DTM。T 必须在安装了 SoMove 软件之后才安装 TeSys Tera DTM Library。

下载 Tesys Tera DTM Library

TeSys Tera DTM Library 可从 Schneider Electric 网站 (www.se.com) 上下载。在 **Search** 字段中输入 **TeSys Tera DTM Library**，以导航到“下载”页面。

注：您必须拥有管理员权限，才能在 PC 上安装或卸载此软件。

安装 Tesys Tera DTM Library

下面的步骤介绍如何安装 TeSys Tera DTM Library：

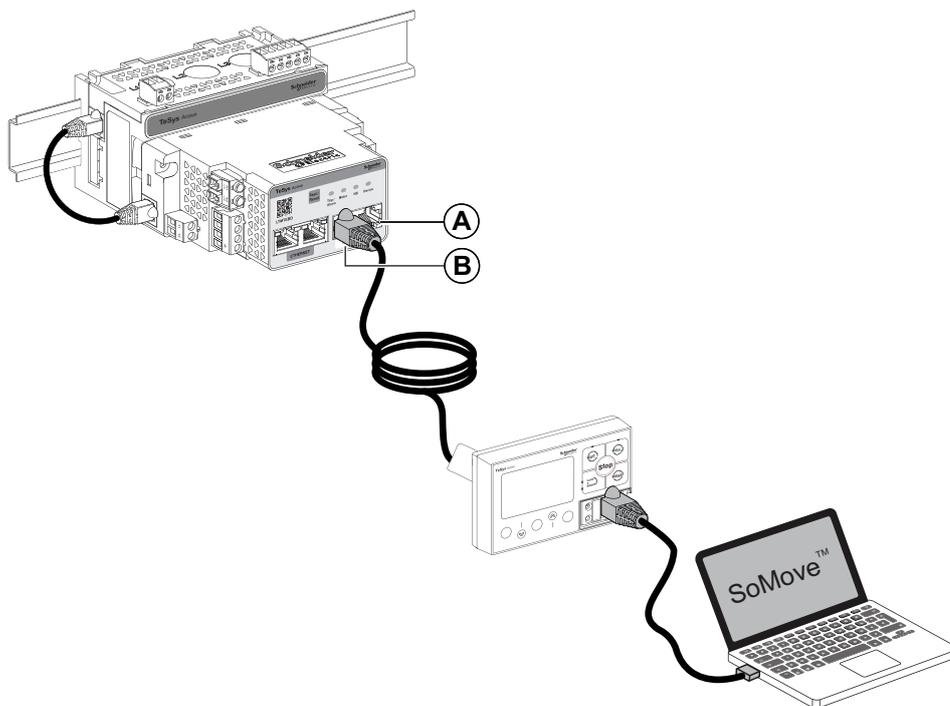
1. 解压下载文件 *TeSysTeraDTMLibrary*。
2. 文件夹将包含 exe 文件 *Schneider_Electric_Tesys_Tera_DTM_Library_V.X.Y.Z* (其中 X.Y.Z 是版本号) 和发行说明 *TeSysTeraDTMLibrary_vx.y.z_ReleaseNotes* (其中 x.y.z 是版本号)。
3. 双击 *Schneider_Electric_Tesys_Tera_DTM_Library_V.X.Y.Z* 以启动安装。
4. 在 **Choose Setup Language** 对话框中，选择安装语言，然后选择 **OK**。
5. 在 **Welcome to the Installation Wizard for Schneider Electric TeSys Tera DTM Library** 对话框中，请选择 **Next**。
6. 在 **Readme and Release Notes** 对话框中，请选择 **Next**。
7. 在 **License Agreement** 对话框中：
 - 仔细阅读许可协议。
 - 在许可协议选项中，选择 **I accept the terms**。
 - 选择 **Next**。
8. 在 **Customer Information** 对话框中：
 - 在相应字段中输入以下信息：
 - First name
 - Last name
 - Company name
 - 选择 **Next**。
9. 在 **Destination Folder** 对话框中：
 - 如有必要，通过选择 **Change** 选项，来修改 TeSys Tera DTM Library 目标文件夹。
 - 选择 **Next**。
10. 在 **Ready to Install the Program** 对话框中，选择 **Install**。
11. 在 **Installation Wizard Completed** 对话框中，选择 **Finish**。

结果：TeSys Tera DTM Library 被安装在您的 PC 上。

将 LTMT Main Unit 连接到 PC

可以通过以下方式就 LTMT main unit 连接到运行 SoMove 软件的 PC :

- 使用 USB 通信线，将运行配置软件的 PC 的 SoMove USB 端口连接到 RJ45 主单元 (Main Unit) 上的 LTMTCUF control operator unit USB 端口，以进行配置和 RJ45 固件更新 (推荐方式)。
- 使用 USB 通信线，将运行配置软件的 PC 的 SoMove USB 端口连接到 HMI 扩展模块 (Expansion Module) 上的 LTMT main unit USB 端口，以进行配置和 RJ45 固件更新。
- 将运行 SoMove 软件的 PC 的 USB 端口连接到现场总线端口以便执行配置。



A 扩展模块端口 (EXP)

B 人机界面端口 (HMI)

有关配置各种通讯协议以及在 TeSys Tera system 与 DTM library 之间建立连接的详细说明，请参阅 SoMove 在线帮助中的“连接”章节。

有关使用 LTMTCUF control operator unit 将 PC 连接到 TeSys Tera system 的详细说明，请参阅 *TeSys Tera Motor Management System LTMTCUF Control Operator Unit User Guide – DOCA0233EN*。

故障排除

问题	原因	解决方法
在 SoMove 软件 v2.10.x 及更高版本中，TeSys Tera DTM Library 不显示在目录窗口中。	应用 DTM 前，未注册 SoMove 软件。	注册 SoMove 软件。 1. 启动 SoMove 软件时，将向用户提示以下消息： 演示版还有 x 天到期。是否要立即注册？ 2. 选择 是 ，确认注册并按照说明进行操作。 注： SoMove 软件的注册是免费的。
	.dll 文件未由 Windows 注册	使用命令提示符窗口来注册 TeSys Tera DTM Library 所需的 .dll 文件。 1. 以管理员身份启动命令提示符窗口，然后输入以下命令： cd C:\Windows\Microsoft.NET\Framework\v2.0.50727 2. 输入以下命令： RegAsm.exe /codebase "C:\Program Files (x86)\Common Files\Schneider Electric Shared\Schneider Electric TeSys Tera DTM Library\TeSysTerraDtm.Kernel.dll 结果： 注册成功后，将显示一条消息。 3. 启动 SoMove 软件。将显示带有进度条的 目录更新 屏幕。 结果： 目录更新完成后，TeSys Tera DTM Library 将显示在 目录 窗口中。

用户界面

此部分内容

启动 TesysTera DTM Library	23
用户界面描述	32
菜单栏	33
工具栏	38
状态栏	40
选项卡区域	42
My Device	45
参数列表	49
Settings	60
My Dashboard	65
Diagnostics	67
Monitoring	72
Start Curve	76
Custom Logic	78
FB Diagram	79
Logic Simulator	80

启动 TesysTera DTM Library

启动 SoMove

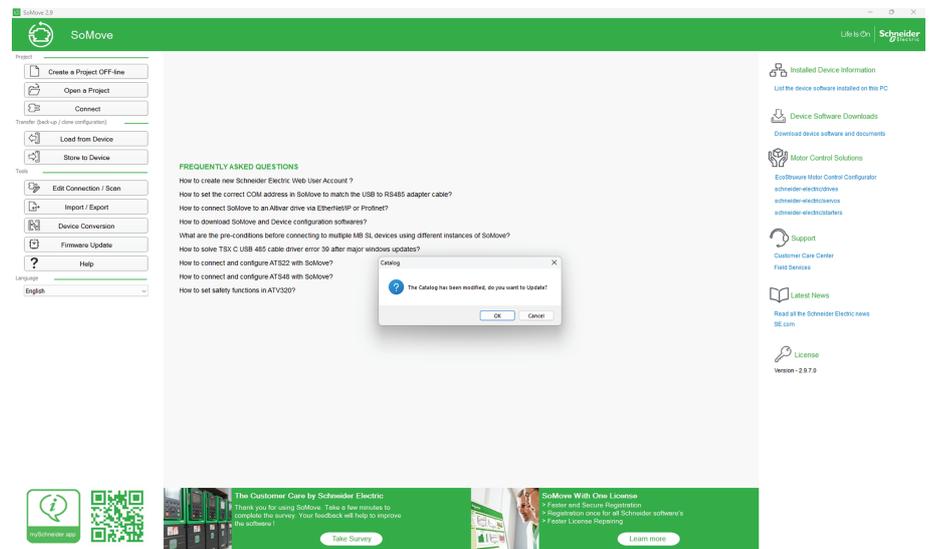
如要创建 TeSys Tera DTM Library 的实例，需要先启动 SoMove 软件。
有关如何启动 SoMove 软件的详细说明，请参阅 SoMove 在线帮助。

创建 TesysTera DTM 实例

以下步骤介绍如何创建 TeSys Tera DTM Library 实例：

1. 打开 SoMove 软件。

结果：显示以下窗口。

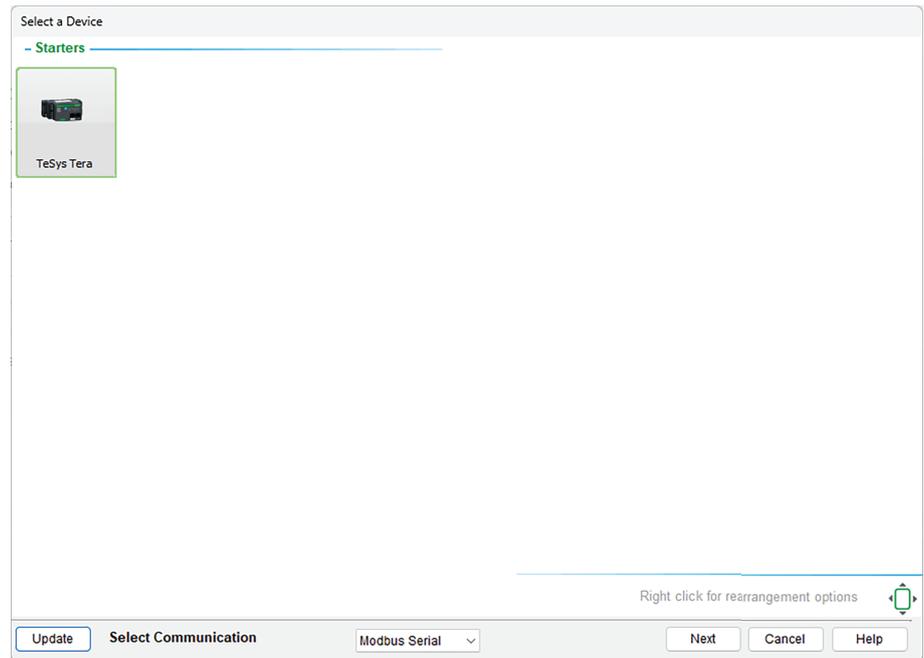


2. 选择 **OK**，更新目录。

注：在首次安装 TeSys Tera DTM Library 并启动 SoMove 软件时，会显示 **Catalog** 窗口。

3. 选择 **Create a Project OFF-line**。

结果： 随即显示 **Select a Device** 窗口。



4. 选择 **TeSys Tera**，然后使用**Select Communication**旁边的下拉列表选择所需的通信方式。单击**下一步**。

结果： 随即打开 TeSys Tera system 工作区域。

注：

- 如果未显示设备，请参阅 故障排除, 21 页。
- 当通信协议选择为 **Modbus TCP**时, 仅限 EtherNet/IP 变体 LTMT main unit 可供配置。

将设备连接到网络

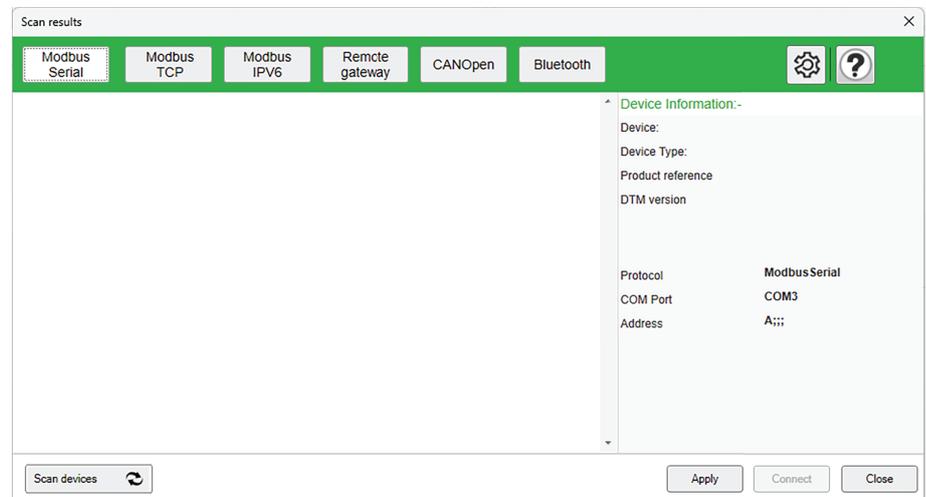
此操作将获取已连接设备的配置。SoMove 软件在会话期间保持连接到此设备。

Modbus 串行

以下过程描述如何通过 **Modbus Serial** 通信连接设备：

1. 打开 SoMove 软件。
2. 通过以下其中一种方式连接到设备：
 - 在 Start 页中，选择 **Edit Connection/Scan**。
 - 在工具栏上，选择  图标。
 - 在菜单栏上，选择 **Communication > Edit Connection/Scan**。

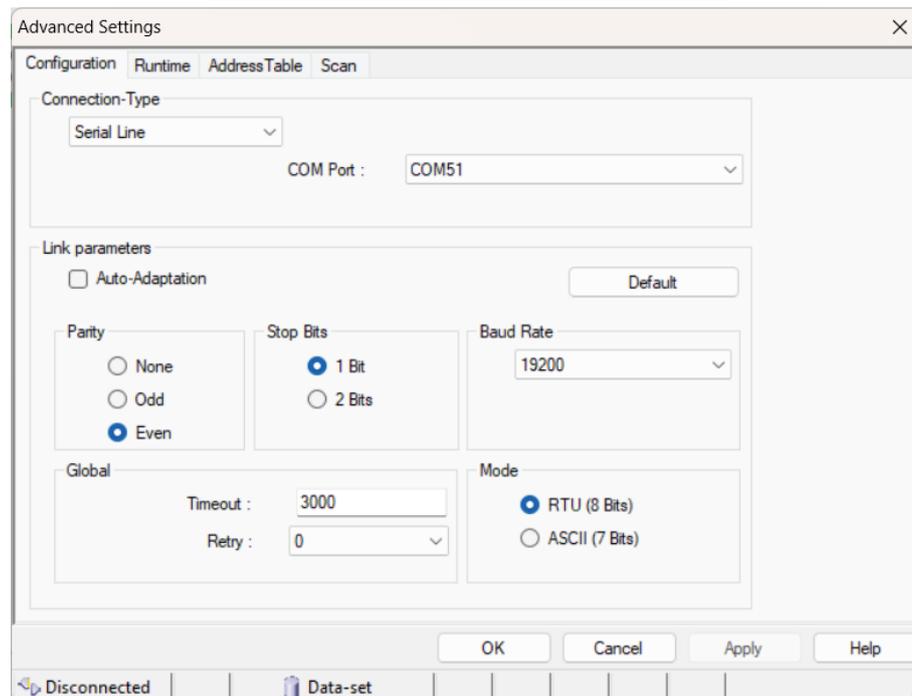
结果： 随即显示 **Scan results** 窗口。



3. 在 **Scan results** 对话框中，选择 **Modbus Serial**。

4. 选择  图标。

结果：随即打开 **Advanced Settings** 窗口。



注： 设备的出厂设置如下（适用于 HMI 或 Modbus RTU 端口）：

- **Parity** : Even
- **Stop Bits** : 1 Bits
- **Baud rate** : 19200

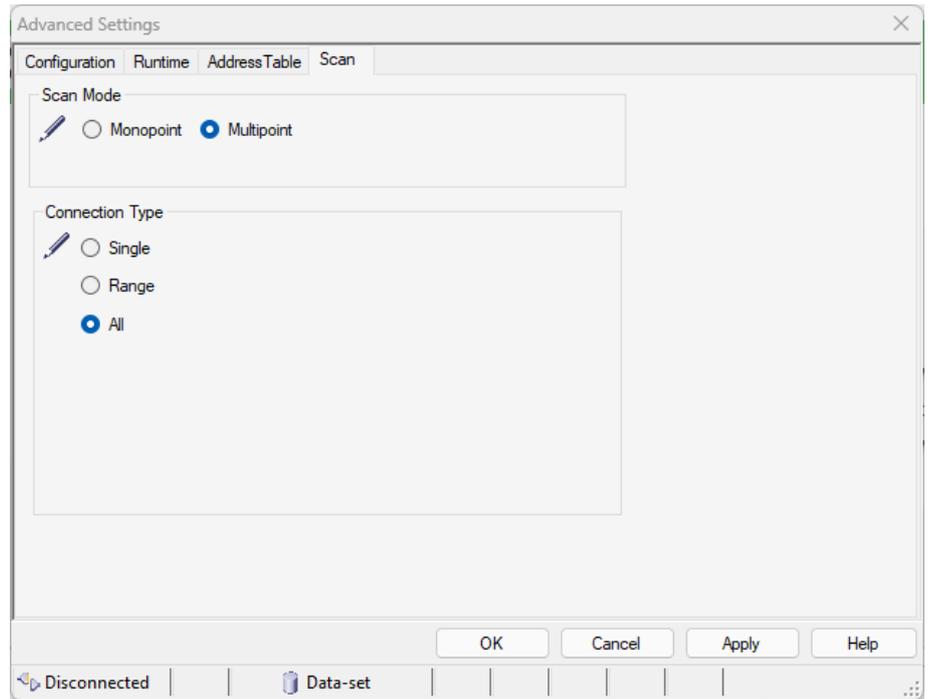
5. 在 **Configuration** 选项卡中，为 Modbus Serial 连接类型输入所需的配置设置。

注： 如果缺省参数设置不可用，请选中 **Auto-Adaptation** 复选框 以发现设备。

有关配置设置的更多信息，请参阅 *SoMove* 在线帮助。

6. 选择 **Scan** 选项卡。

结果：显示 **Advanced Settings** 窗口中的 **Scan** 选项卡。



注：仅支持为 Modbus 连接使用 **Multipoint Scan Mode**。

7. 选择 **Scan Mode** 和 **Connection Type**。

有关更多信息，请参阅 SoMove 在线帮助中的“连接”章节。

8. 选择**OK**。

结果：修改被保存，且 **Advanced Settings** 窗口关闭。将在下一次扫描时应用新值。

注：如果选择 **Cancel**，则 **Advanced Settings** 窗口在不保存任何更改的情况下关闭，下次扫描将应用缺省值。

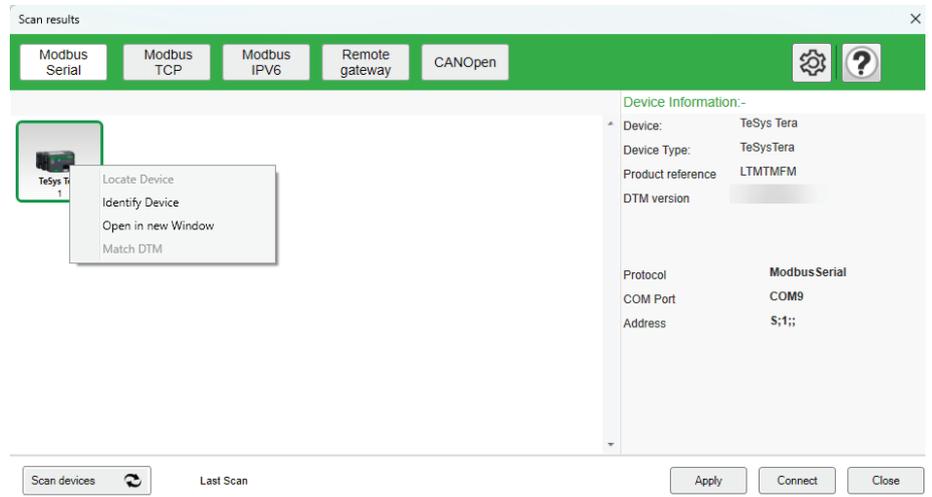
9. 选择 **Scan devices**。

结果：显示网络上具有 Modbus Serial 连接能力的所有设备。

10. 选择适合的 TeSys Tera system，然后选择 **Connect**。

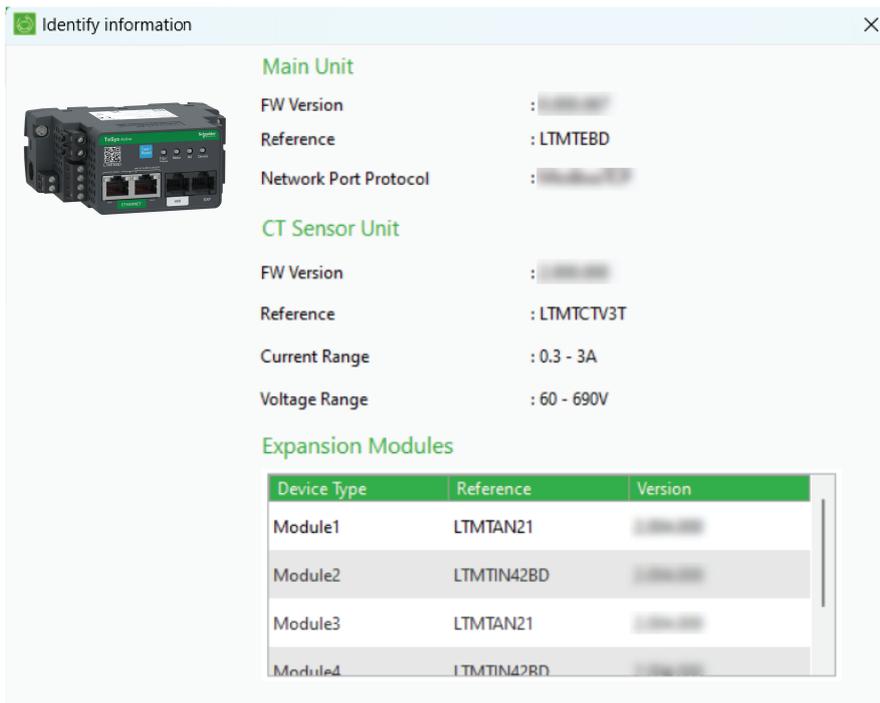
11. 右键单击设备。

结果：随即显示 **Scan results** 窗口。

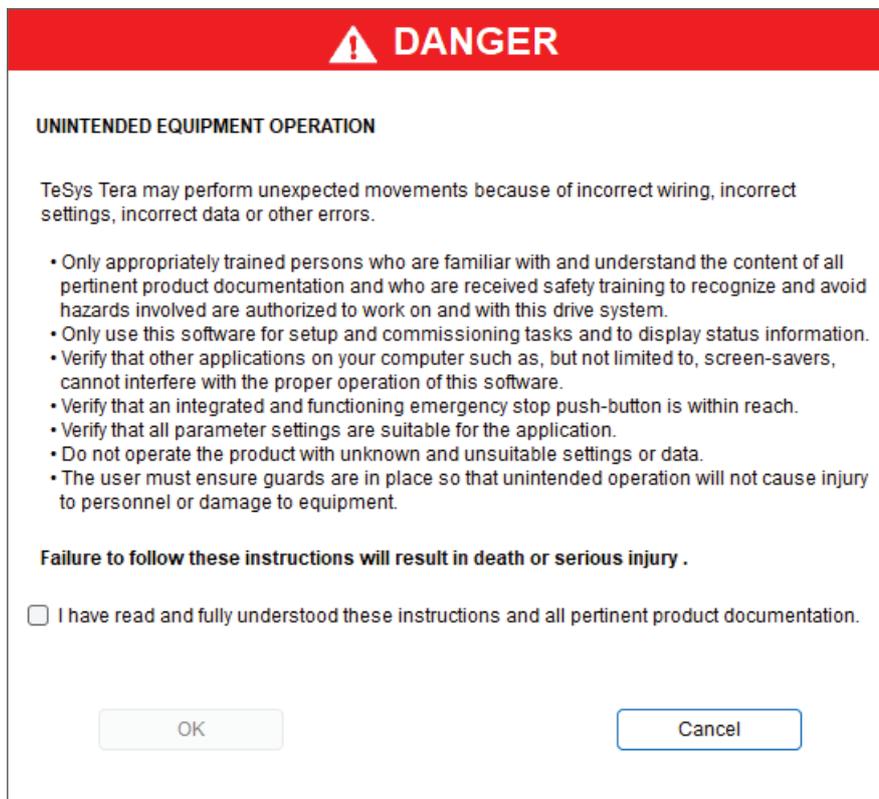


12. 选择 **Identify Device**。

该 **Identify Information** 窗口弹出。该窗口显示已连接设备的详细信息。

13. 选择要连接的已识别设备，然后选择 **Connect**。

14. 危险消息将按以下格式显示。检查是否满足安全消息中的所有要求。选中复选框，然后单击**OK**。



结果：TeSys Tera system 连接到您的 PC。

注:

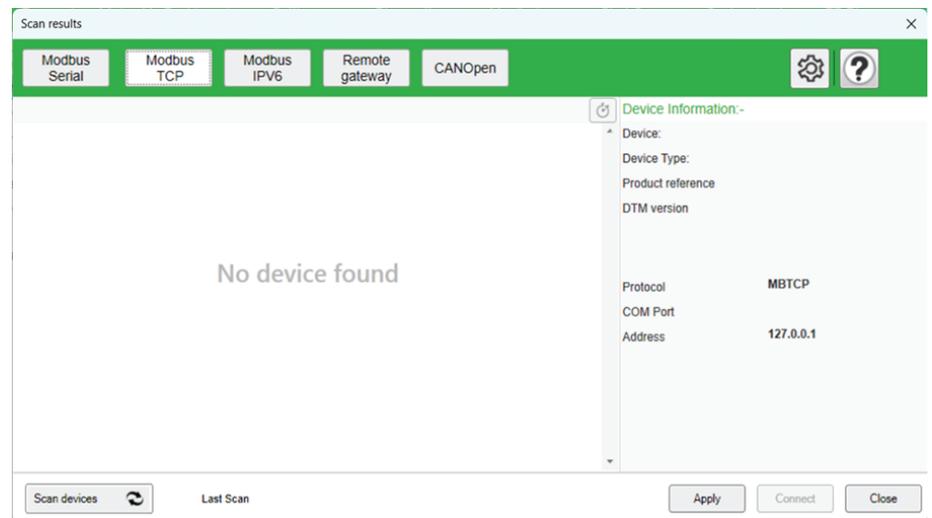
- 确认设备与 PC 之间的电缆正确连接。
- 确认设备已连接到电源。
- 选择 **Advanced Setting**，确认连接参数的定义是正确的。

Modbus TCP

以下过程描述如何通过 **Modbus TCP** 通信连接设备：

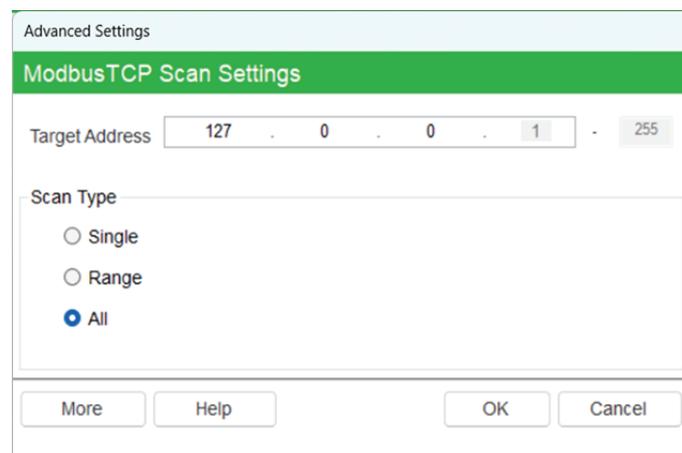
1. 打开 SoMove 软件。
2. 通过以下其中一种方式连接到设备：
 - 在 Start 页中，选择 **Edit Connection/Scan**。
 - 在工具栏上，选择  图标。
 - 在菜单栏上，选择 **Communication > Edit Connection/Scan**。

结果： 随即显示 **Scan results** 窗口。



3. 在 **Scan results** 对话框中，选择 **Modbus TCP**。
4. 选择  图标。

结果： 随即打开 **Advanced Settings** 窗口。



5. 在 **Advanced Settings** 窗口中，输入 **Target Address** 并选择所需的 **Scan Type**。

有关详细信息，请参阅 [SoMove 在线帮助](#)。

6. 选择**OK**。

结果：修改被保存，且 **Advanced Settings** 窗口关闭。将在下一次扫描时应用新值。

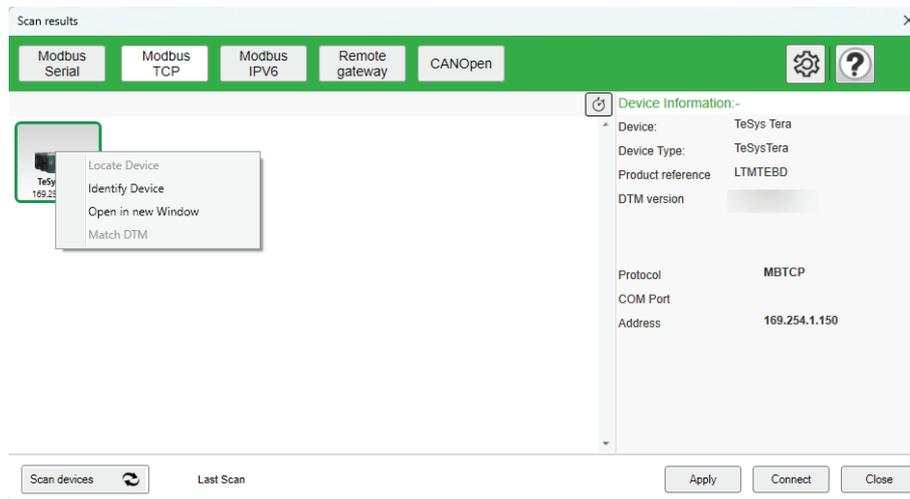
注：如果选择 **Cancel**，则 **Advanced Settings** 窗口在不保存任何更改的情况下关闭，下次扫描将应用缺省值。

7. 选择 **Scan devices**。

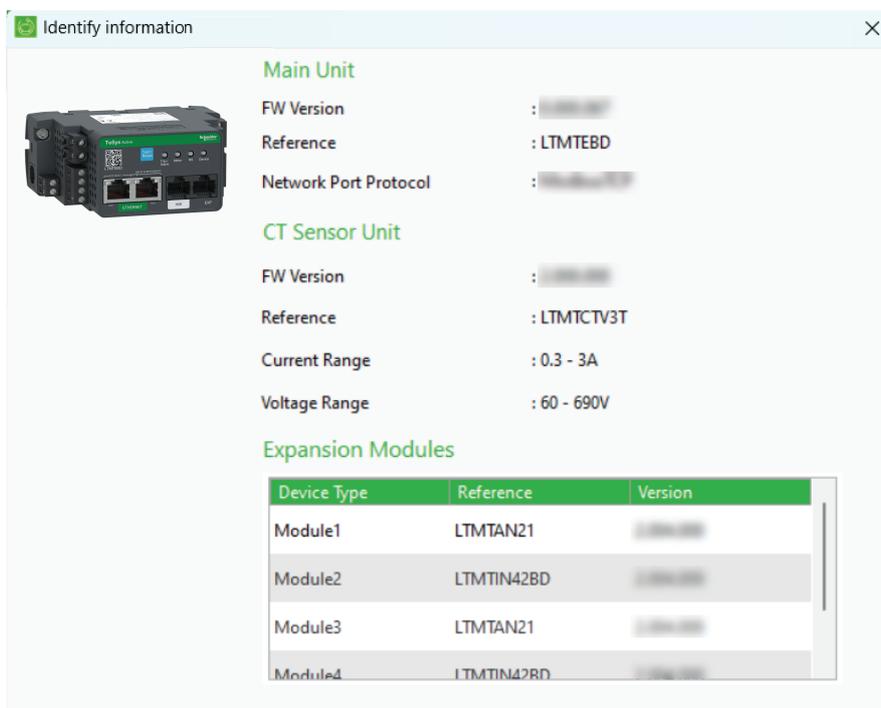
结果：显示网络内所有具备Modbus TCP 连接能力的设备

8. 选择适合的 TeSys Tera system，然后选择 **Connect**。

9. 右键单击设备。

10. 选择 **Identify Device** .

该 **Identify Device** 窗口弹出。该窗口显示已连接设备的详细信息。

11. 选择要连接的已识别设备，然后选择 **Connect**。

12. 危险消息将按以下格式显示。检查是否满足安全消息中的所有要求。选中复选框，然后单击**OK**。

⚠ DANGER

UNINTENDED EQUIPMENT OPERATION

TeSys Tera may perform unexpected movements because of incorrect wiring, incorrect settings, incorrect data or other errors.

- Only appropriately trained persons who are familiar with and understand the content of all pertinent product documentation and who are received safety training to recognize and avoid hazards involved are authorized to work on and with this drive system.
- Only use this software for setup and commissioning tasks and to display status information.
- Verify that other applications on your computer such as, but not limited to, screen-savers, cannot interfere with the proper operation of this software.
- Verify that an integrated and functioning emergency stop push-button is within reach.
- Verify that all parameter settings are suitable for the application.
- Do not operate the product with unknown and unsuitable settings or data.
- The user must ensure guards are in place so that unintended operation will not cause injury to personnel or damage to equipment.

Failure to follow these instructions will result in death or serious injury .

I have read and fully understood these instructions and all pertinent product documentation.

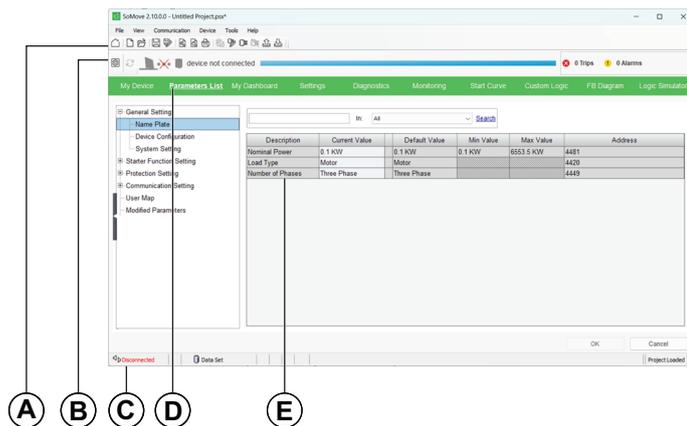
OKCancel

结果： TeSys Tera system 连接到您的 PC。

注：

- 确认设备与 PC 之间的电缆正确连接。
- 确认设备已连接到电源。
- 选择 **Advanced Setting**，确认连接参数的定义是正确的。

用户界面描述



工作区分为以下几个区域：

- A 菜单栏
- B 工具栏
- C 状态栏
- D 选项卡选择栏
- E 选项卡区域（内容取决于所选的选项卡）

菜单栏

描述

菜单栏是 SoMove 软件的一部分。菜单栏位于工作区的顶部，如下所示：



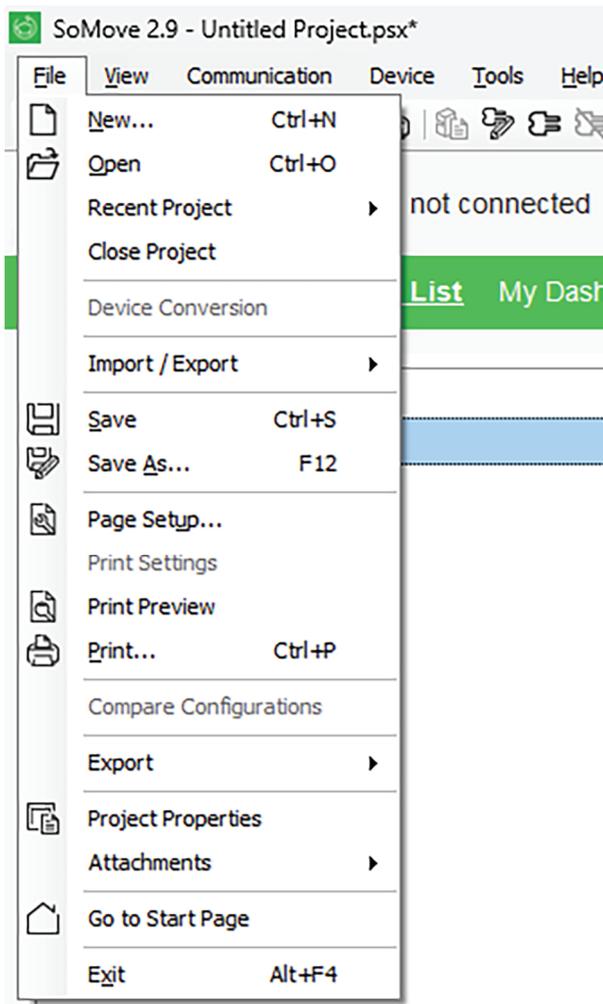
通过菜单栏，用户可通过菜单或图标访问操作功能。

在菜单栏中，灰色命令不可用。

注： SoMove 软件的某些选项不适用于 TeSys Tera DTM Library。不可用命令的详细信息未在描述表中列出。有关 SoMove 软件中可用选项的详细信息，请参阅 SoMove 在线帮助。

File 菜单

可通过单击菜单栏上的 **File** 来访问 **File** 菜单。

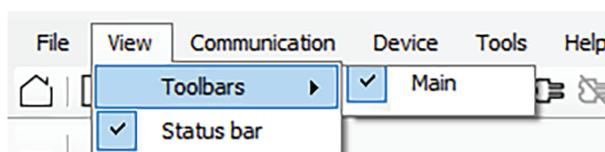


命令	描述
New...	创建新项目。
Open	打开 PC 上所保存的项目。

命令	描述
Recent Project	打开最近打开的项目。
Close Project	通过提示保存，来关闭当前项目。
Save	让您能够保存对现有项目的修改。
Save As	使用名称和/或在新位置保存打开的项目。
Project Properties	显示已打开项目的属性。
Go To Start Page	显示起始页。
Exit	退出 SoMove 软件。

View 菜单

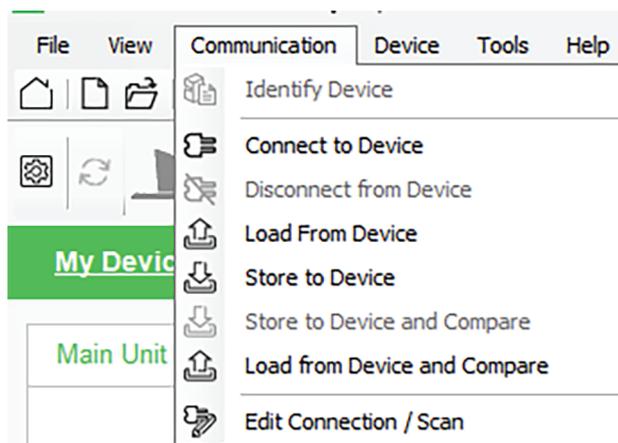
可通过单击菜单栏上的 **View** 来访问 **View** 菜单。



命令	说明
工具栏 > 主工具栏	显示或隐藏主工具栏。
Status bar	显示或隐藏状态栏。

Communication 菜单

可通过单击菜单栏上的 **Communication** 来访问 **Communication** 菜单。

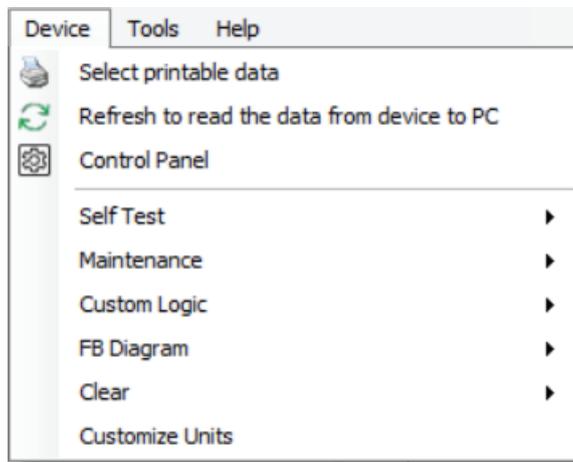


命令	说明
Identify Device	显示以下与已连接的设备有关的信息： <ul style="list-style-type: none"> 设备参考 制造商 固件版本
Connect to Device	获取已连接设备的配置，并在整个会话期间保持连接。
Disconnect from Device	与所连接的设备断开连接。
Load From Device	获取已连接设备的配置。

命令	说明
	注: 进行此操作时, 请确保DTM library 未与设备断开连接, 否则将导致通信中断。
Store to Device	将配置文件从 PC 上所保存的现有项目传输到已连接的设备。 注: 此操作将需要一定时间来下载配置。
Load from Device and Compare	从已连接设备获取配置, 并将其与当前打开的项目配置进行比较。
Edit Connection / Scan	显示连接参数并执行连接测试。

Device 菜单

可通过单击菜单栏上的 **Device** 来访问 **Device** 菜单。

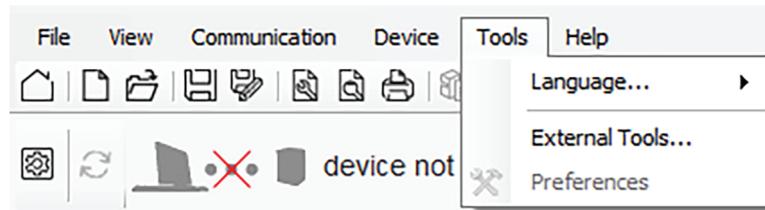


命令	说明
Select printable data	让您能够选择打印所需的数据。
刷新以将数据从设备读取到 PC	刷新参数列表选项卡中的所有参数。
Control Panel	显示或隐藏控制面板。
自检 注: 自检功能仅在线模式下可用。	允许您执行自检。
• (有脱扣) 自检命令	允许您启动带脱扣的设备诊断测试。
• (无脱扣) 自检命令	允许您启动无脱扣的设备诊断测试。
维护 • 固件更新 • 更新日期和时间 注: 更新日期和时间命令仅当 LTMT main unit 通过串行端口连接到 Ethernet 型号变体时可用。	提供设备维护工具的访问权限 访问固件更新命令。 • 允许您设置日期和时间: 提供以下选项: ◦ 系统 ◦ 自定义 注: 如果在运行时更改了系统时区, SoMove 请重新启动 SoMove 以应用更新后的 PC 时区。
自定义逻辑 • New Custom Logic • Open Custom Logic	提供对 Custom Logic Editor 的访问, 以便以自定义逻辑语言创建自定义程序。 • 打开新的自定义逻辑程序文件。 • 打开现有自定义逻辑程序文件。

命令	说明
<ul style="list-style-type: none"> • Save Custom Logic • SaveAs Custom Logic • Close Custom Logic • Compile Custom Logic 	<ul style="list-style-type: none"> • 将自定义逻辑程序文件另存为*.lf文件。 • 将自定义逻辑程序文件以新文件名保存到不同位置。 • 通过提示保存，来关闭自定义逻辑程序文件。 • 根据标准验证在自定义逻辑程序中编写的代码。
FB Diagram <ul style="list-style-type: none"> • New FB Diagram • Open FB Diagram • Save FB Diagram As • Compile FB Diagram to Custom Logic • FBD Editor • View • Tools 	提供对 Custom Logic Editor 的访问，以便以功能块图 (FBD) 语言创建自定义程序。 <ul style="list-style-type: none"> • 创建一个New FB Diagram。 • 打开任何现有的功能块图 (FB Diagram) 文件。 • 将FB图文件保存为 *.gef 文件，并使用新文件名保存在其他位置。 • 编译功能块图，并将其转换为自定义逻辑。 • 允许您编辑功能块图(FB Diagram)。 • 让您您能够使用各种可用的查看选项来查看功能块图。有关更多信息，请参阅 FBD 编辑器显示选项, 186 页。 • 显示用于更改链接或对功能块重新编号的工具。
清除 >清除热内存 (Clear > Clear Thermal Memory) 注: Clear Thermal Memory 仅支持在线模式。	允许您重置设备内存中存储的热数据。
Customize Units	让您能够配置以下显示单位： <ul style="list-style-type: none"> • 温度 (摄氏度或华氏度) • 电机额定值 (KW 或 HP)

Tools 菜单

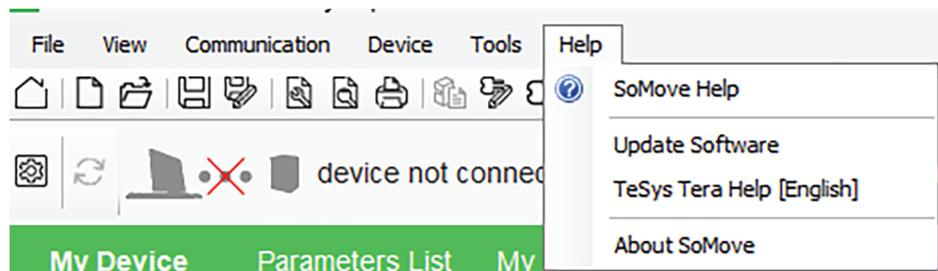
可通过单击菜单栏上的 **Tools** 来访问 **Tools** 菜单。



Tools 菜单中的命令	说明
语言	修改SoMove软件的显示语言。 注: TeSys Tera DTM Library 当前仅支持英语。

Help 菜单

可通过单击菜单栏上的 **Help** 来访问 **Help** 菜单。

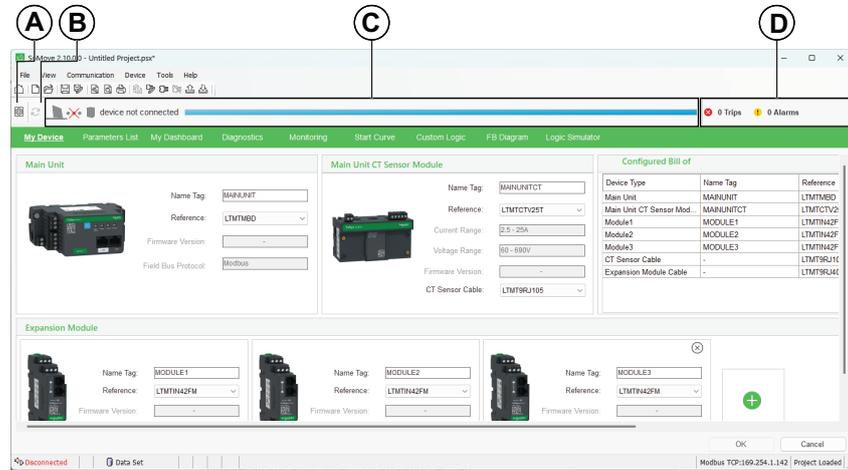


命令	说明
SoMove Help	显示 SoMove 在线帮助。
Update Software	将 SoMove 软件更新至最新版本。
TeSys Tera 帮助 [英语] (Help [English])	打开 TeSys Tera DTM 在线帮助 (Online Help)。
About SoMove	显示有关 SoMove 软件的一般信息。

工具栏

描述

工具栏位于工作区顶部菜单栏的正下方，因具体的 TeSys Tera DTM Library 而异：

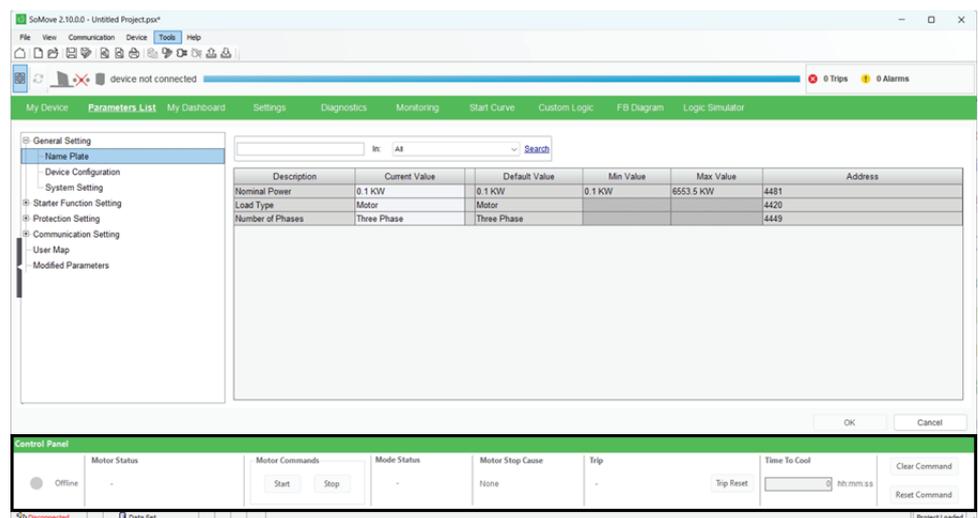


- A 控制面板
- B 刷新
- C 同步数据区
- D 脱扣和报警计数显示区

控制面板

如要访问 **Control Panel**，请在工具栏上选择 .

Control Panel 显示在窗口底部。



Control Panel 显示以下参数：

- 设备状态
- 电机状态
- 模式状态

- 电机停止原因
- 脱扣状态
- 电机停止或脱扣时的冷却时间

Control Panel 让您能够执行以下操作：

- 提供电机命令（启动/停止）
- 脱扣复位
- 清除命令
- 复位命令

Refresh

工具栏上的  选项可刷新 **Parameter List** 选项卡下方的所有参数。

同步数据区

同步数据区显示 LTMT main unit 与 PC 间数据的同步状态。

当 LTMT main unit 处于连接模式时，显示数据自动同步。

模式	图标	描述
断开		LTMT main unit 未与 PC 同步： <ul style="list-style-type: none"> • 参数列表标头和同步数据区为蓝色。 • 未实时从 LTMT main unit 上读取参数。 • 配置模式下所有的参数都可修改。 • 应执行 Store to Device 操作，以将设置写入到设备。若要保存此设置，必须保存项目。
连接		LTMT main unit 与 PC 同步： <ul style="list-style-type: none"> • 参数列表标头和同步数据区为橙色。 • 显示的参数是实时从 LTMT main unit 上读取的 • 已修改的参数在确认时被写入到设备。

脱扣/事件和报警计数显示

此区域显示项目中的脱扣和报警的总数。

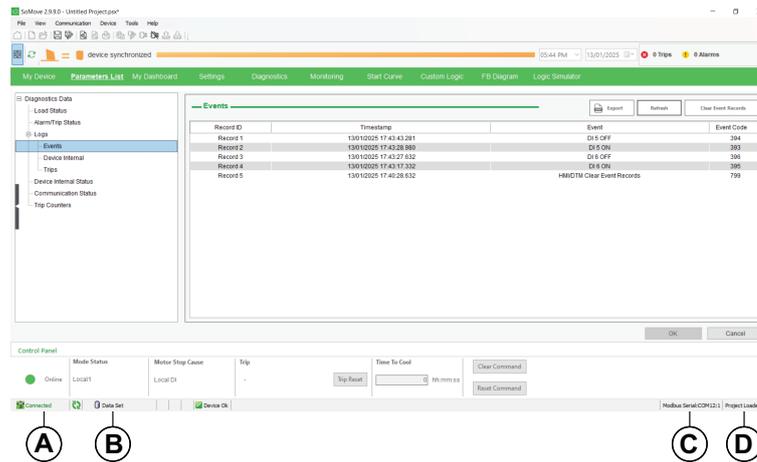


状态栏

目的

状态栏位于工作区底部，显示 LTMT main unit 的当前状态和 SoMove 软件相关信息。有关状态栏的更多信息，请参阅 SoMove 在线帮助。

描述



A

B

C

D

- A 连接状态
- B 数据源
- C 特定消息区
- D 项目状态

连接状态

Connection Status 指示 LTMT main unit 与 PC 之间的连接模式。

模式	图标	描述
断开		LTMT main unit 未连接到 PC。
干扰		LTMT main unit 与 PC 之间的连接丢失。
连接		LTMT main unit 连接到 PC。

数据源

Data Source 指示数据的当前模式。下表介绍了数据源模式：

模式	图标	描述
数据集		数据集中的参数值可以在连接模式下修改。
数据集或设备已锁定		设备在连接模式下受到保护。
设备		设备数据集包含信息。
设备/数据集		所有参数值都存储至两个数据源。

特定消息区

根据用于与设备通讯的现场总线，特定消息区会显示 IP 或设备地址、COM 端口详细信息等。

项目状态

SoMove 软件项目的状态可能是：

- **Project Loaded**：工作区中显示有项目。
- **No Project Open**：项目工作区为空。

选项卡区域

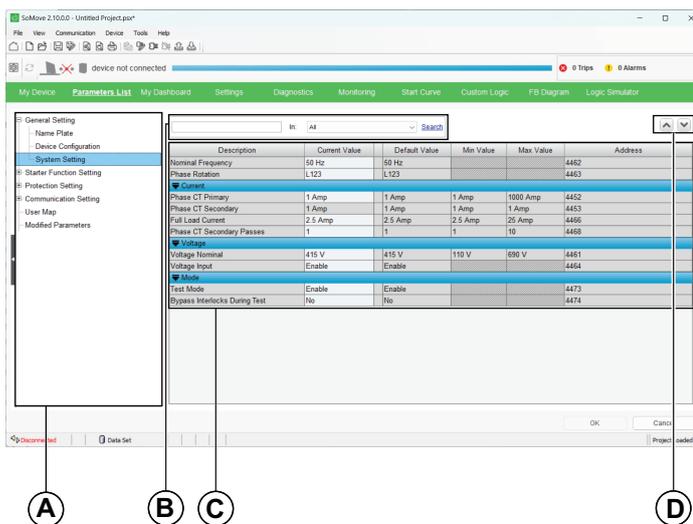
概述

下表列出了各种可用的选项卡。

选项卡名称	描述
My Device	显示设备模块和特性。
Parameters List	显示所有设备模块的可配置参数。
Settings 注: Settings 标签页仅在 My Device 标签页中选择了 Ethernet的LTMT main unit变体时才可用。	显示可配置的 Modbus TCP 和 EtherNet/IP 的设置 LTMT main unit。
My Dashboard	显示所有设备模块的可配置参数和状态参数，可根据要求自定义设置这些参数。
Diagnostic	显示诊断状态数据。
Monitoring	显示设备模块的监控数据。
Start Curve	以图形格式显示电机启动特性。
Custom Logic	提供对 Custom Logic Editor 的访问，以便以自定义逻辑语言创建自定义程序。
FB Diagram	提供使用 FBD 语言编辑自定义程序的自定义逻辑编辑器访问权限。
Logic Simulator	提供对逻辑仿真器的访问，以便以自定义逻辑语言创建自定义程序。

说明

该窗口通过以下选项卡显示常见信息：



- A 树视图
- B 搜索功能
- C 显示区域
- D 展开和折叠按钮

树视图

树视图包含划分为项和子项的参数组。在树视图中选择某个项或子项，以在显示区域中显示参数。

搜索功能

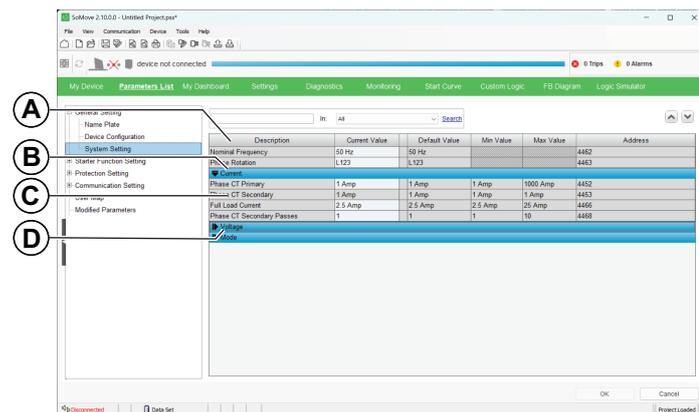
下面的步骤介绍如何在所显示的表中查找特定文本：

1. 在搜索栏的第一个字段中，输入要搜索的字符（部分字、代码或单位）。
2. 从下拉列表中选择要搜索的列。

如果选择 **All** 选项，则会在表格的所有列中执行搜索。

3. 选择 **Search** :
 - 搜索到的第一个匹配文本就会突出显示。
 - 要搜索其他实例，请选择 **Search**。
 - 如果未找到匹配的文本，搜索字段中的字符颜色将变为红色。

网格视图下的显示区域



- A 列标题
- B 参数组
- C 参数：
 - 每行一个参数，参数属性显示在不同的字段中。
 - 白色单元格的内容可进行修改，灰色单元格为只读。
- D 折叠或展开图标：如要折叠或展开参数组，请单击相应组的箭头。

对参数排序

如要根据列中的值对参数排序，请单击标题：

操作	结果	标题示例
单击	<ul style="list-style-type: none"> 参数值按升序排序。 标题出现时箭头朝上。 	
双击	<ul style="list-style-type: none"> 参数值按降序排序。 标题出现时箭头朝下。 	
三击	<ul style="list-style-type: none"> 参数按照原始顺序显示。 标题以其最初的形式出现。 	

修改列顺序

以下步骤介绍如何修改显示区域中的列顺序：

1. 单击列标题。
2. 将列拖动到所需位置。

展开和折叠按钮

显示区域视图可通过显示区域右上角提供的以下按钮进行修改：

按钮	功能	描述
	全部展开	展开所有组以显示所有参数。
	全部折叠	折叠显示区域中的所有组。

My Device

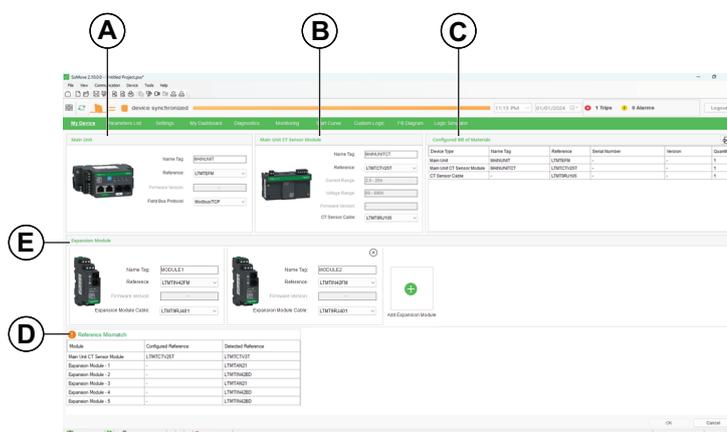
概况

My Device 标签页 用于：

- 在断开模式下配置 TeSys Tera system。
- 在连接模式下显示 TeSys Tera system 特性。

描述

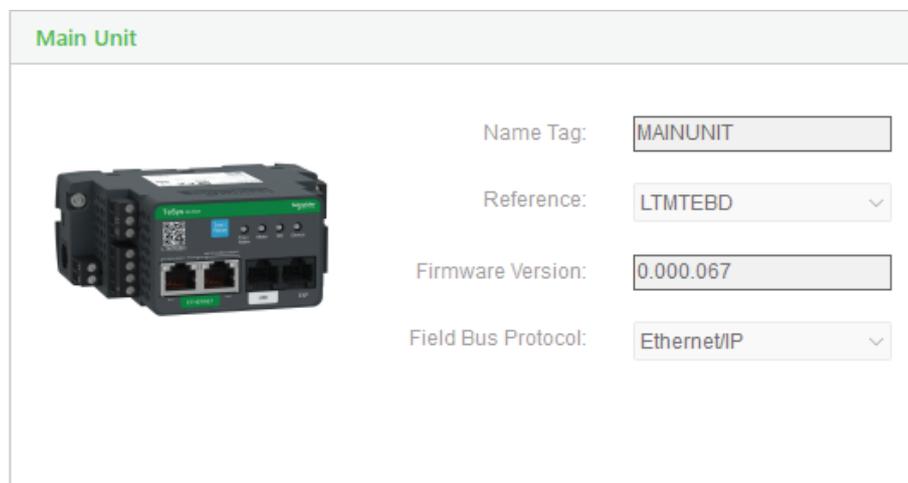
下图显示了 My Device 选项卡的不同部分。



- A Main Unit
- B Main Unit CT Sensor Module
- C Configured Bill of Materials
- D 型号不匹配
- E 扩展模块

Main Unit

Main Unit 区域显示 LTMT main unit 配置的 TeSys Tera system 信息。



Main Unit 部分中提供了以下特性：

特性	描述
Name Tag	输入 LTMT main unit 的名称。 注: <ul style="list-style-type: none"> • Name Tag 字段不能为空。 • Name Tag 字段支持以下字符： <ul style="list-style-type: none"> ◦ 字母：A 到 Z (大写字母) ◦ 数字：1 至 9 ◦ 符号：_、/ 和 - • Name Tag 字段最多支持10个字符。
Reference	选择 LTMT main unit 的参考值。 注: LTMTEFM被选为默认 LTMT main unit 参考值。
Firmware Version	显示所选 LTMT main unit 的固件版本。 注: 只有在设备处于在线模式时，才显示固件版本。
Field Bus Protocol	显示所使用的 LTMT main unit 的网络协议。 以下选项可用于选择 Ethernet 变体 LTMT main unit: <ul style="list-style-type: none"> • Modbus/TCP • EtherNet/IP

注: 如果 DTM library 中配置的 LTMT main unit 型号与实际设备配置不同，将显示错误消息，并且不会建立连接。

Main Unit CT Sensor Module

Main Unit CT Sensor Module 区域显示系统中配置的主单元CT传感器模块的信息。 TeSys Tera system

Main Unit CT Sensor Module



Name Tag:

Reference:

Current Range:

Voltage Range:

Firmware Version:

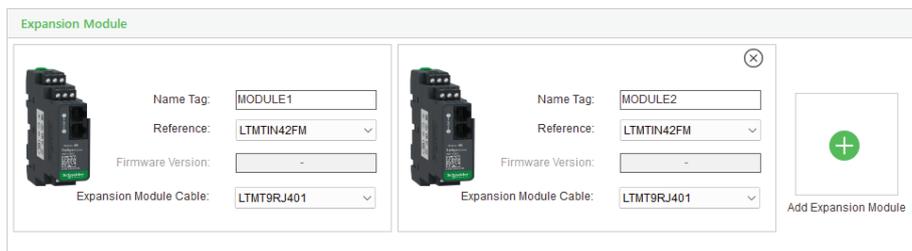
CT Sensor Cable:

Main Unit CT Sensor Module 部分中提供了以下特性：

特性	描述
Name Tag	输入 Main Unit CT Sensor Module 的名称。 注: <ul style="list-style-type: none"> • Name Tag 字段不能为空。 • Name Tag 字段支持以下字符： <ul style="list-style-type: none"> ◦ 字母：A 到 Z (大写字母) ◦ 数字：1 至 9 ◦ 符号：_、/ 和 - • Name Tag 字段最多支持10个字符。
Reference	选择 Main Unit CT Sensor Module 的型号。
Current Range	显示所选传感器模块的电流范围。
Voltage Range	显示所选传感器模块的电压范围。 注: 仅为电压版本显示电压范围。
Firmware Version	显示所选 Main Unit CT Sensor Module 的固件版本。 注: 只有在设备处于在线模式时，才显示固件版本。
CT Sensor Cable	选择所使用的传感器电缆类型。

Expansion Module

Expansion Module 区域显示 TeSys Tera system系统中配置的扩展模块的信息。



Expansion Module 部分中提供了以下特性：

特性	描述
Name Tag	输入 LTMT expansion module 的名称。 注: <ul style="list-style-type: none"> • Name Tag 字段不能为空。 • Name Tag 字段支持以下字符： <ul style="list-style-type: none"> ◦ 字母：A 到 Z (大写字母) ◦ 数字：1 至 9 ◦ 符号：_、/ 和 - • Name Tag 字段最多支持10个字符。
Reference	选择 LTMT expansion module 的型号。
Firmware Version	显示 LTMT expansion module 的固件版本。 注: 设备处于在线模式时，显示固件版本。
Expansion Module Cable	选择用于连接的电缆参考型号 LTMT expansion module 。

选择 **Add Expansion Module** 卡，最多可添加五个 **LTMT expansion modules**。

下表介绍了 TeSys Tera system 中可添加的 **LTMT expansion modules** 的最大数量：

LTMT expansion modules 型号	最大设备数量
LTMTIN42FM/LTMTIN42BD	5
LTMTAN21	2

Reference Mismatch

Reference Mismatch 区域显示已配置设备与识别到的设备之间的差异。

! Reference Mismatch		
Module	Configured Reference	Detected Reference
Main Unit CT Sensor Module	LTMTCTV25T	LTMTCTV3T
Expansion Module - 1	-	LTMTAN21
Expansion Module - 2	-	LTMTIN42BD
Expansion Module - 3	-	LTMTAN21
Expansion Module - 4	-	LTMTIN42BD
Expansion Module - 5	-	LTMTIN42BD

注: Reference Mismatch 区域仅在线模式下显示。

Configured Bill of Materials

Configured Bill of Materials 显示了 TeSys Tera system 中配置的所有设备和附件的列表。

Configured Bill of Materials					
Device Type	Name Tag	Reference	Serial Number	Version	Quantity
Main Unit	MAINUNIT	LTMTBBD	LTMTBBDX900003	0.000.067	1
Main Unit CT Sensor Module	MAINUNITCT	LTMTCTV3T	LTMTCTV3TEX900004	2.000.000	1
Module1	MODULE1	LTMTAN21	LTMTAN21DY900107	2.004.000	1
Module2	MODULE2	LTMTIN42BD	LTMTIN42BDDX900042	2.004.000	1
Module3	MODULE3	LTMTAN21	LTMTAN21DY900006	2.004.000	1
Module4	MODULE4	LTMTIN42BD	LTMTIN42BDDX900049	2.004.000	1
Module5	MODULE5	LTMTIN42BD	LTMTIN42BDDX900012	2.004.000	1
CT Sensor Cable	-	LTMT9RJ105	-	-	1
Expansion Module Cable	-	LTMT9RJ401	-	-	5

基于为 TeSys Tera system 执行的配置，模块列表以表格形式显示。

注: 设备处于在线模式时，显示 **Serial Number** 和 **Version**。

特性	描述
Device Type	显示所连接的设备的类型。
Reference	显示所连接的设备或附件的型号。
Serial Number	显示所连接的设备的序列号。 注: 只有在设备处于在线模式时，才显示序列号。
Version	显示所连接的设备的版本号。 注: 只有在设备处于在线模式时，才显示版本号。
Quantity	显示所使用的设备数、CT 传感器电缆和 Expansion Module 电缆。

通过选择  (**Export Bill of Materials**) 图标，可以将配置的物料清单导出为 CSV 文件。

参数列表

此章节内容

General Settings	52
Starter Function Setting	53
Protection Setting	53
Communication Setting.....	57
User Map.....	58
Modified Parameters	58

概述

Parameters List 选项卡用于定义 TeSys Tera system 的参数设置。

在连接模式和断开模式下，都可以在 **Current Value** 字段中修改参数值。

在连接模式下，可以通过选择 **OK**，将设置更新到设备。

在断开模式下，可以通过执行 **Store to Device** 操作，将配置设置下载到设备。

注: **Parameter List** 选项卡中的值仅在初始加载时更新一次。如要在在线模式下查看 **Parameter List** 选项卡中的更新值，执行刷新操作。

描述

Description	Current Value	Default Value	Min Value	Max Value	Address
Normal Frequency	50 Hz	50 Hz			4482
Phase Rotation	1, 2, 3	1, 2, 3			4483
Phase CT Primary	1 Amp	1 Amp	1 Amp	1000 Amp	4482
Phase CT Secondary	1 Amp	1 Amp	1 Amp	1 Amp	4483
Full Load Current	2.4 Amp	2.4 Amp	2.4 Amp	24 Amp	4486
Phase CT Secondary Phase	1	1	0	0	4489
Voltage Nominal	415 V	415 V	110 V	690 V	4485
Voltage-Hz	Enable	Enable			4488
Test Mode	Enable	Enable			4473
Stop on Wrenches Changing Test	No	No			4474

- A 描述：参数名称或描述
- B Current Value：参数值
- C Wrench Mark：如果当前值与缺省值不同，则显示此标记
- D Default Value：参数的缺省值
- E Min Value：参数的最小值。
- F Max Value：参数的最大值。
- G Address：参数的寄存器地址

在 **Parameters Lists** 选项卡的树视图下方提供了以下类别：

- General Setting
- Starter Function Setting
- Protection Setting
- Communication Setting
- User Map
- Modified Parameters

设置数值

参数的数值可通过两种方式设置：

- 直接输入数值。
- 使用旋钮来选择值。

以下步骤介绍如何通过直接输入来设置数值：

1. 导航至树视图下方的所需部分。
2. 从列表中选择参数。
3. 在 **Current Value** 字段中，输入参数值。
4. 在连接模式下，选择 **OK**，可将值写入到设备，或者在断开模式下，执行此选择，可将值写入到 DTM 离线配置。

以下步骤介绍如何使用旋钮来设置数值：

1. 导航至树视图下方的所需部分。
2. 从列表中选择参数。
3. 在 **Current Value** 字段中，使用旋钮设置参数值。
4. 在连接模式下，选择 **OK**，可将值写入到设备，或者在断开模式下，执行此选择，可将值写入到 DTM 离线配置。

编辑字符串

以下步骤介绍如何设置字符串参数：

1. 导航至树视图下方的所需部分。
2. 从列表中选择参数。
3. 在 **Current Value** 字段中，输入字符串。
4. 在连接模式下，选择 **OK**，可将值写入到设备，或者在断开模式下，执行此选择，可将值写入到 DTM 离线配置。

在列表中选择值

以下步骤介绍如何在列表中选择值：

1. 导航至树视图下方的所需部分。
2. 从列表中选择参数。
3. 在 **Current Value** 字段中，使用下拉列表设置参数值。
4. 在连接模式下，选择 **OK**，可将值写入到设备，或者在断开模式下，执行此选择，可将值写入到 DTM 离线配置。

有关该选项卡的完整描述，请参阅章节选项卡区域描述, 42 页。

注: 在编辑参数列表中的缺省值后，选择 **OK**，则将显示以下警告消息。确保执行了所需措施，然后选择 **OK**。

▲ 警告

设备异常行为

- 更改设备配置可能导致短路或负载电源打开。
- 检查是否根据设备配置完成了适当的接线和配置。
- 更换 CTVT 传感器和启动器时，应确保三相电源已切断。

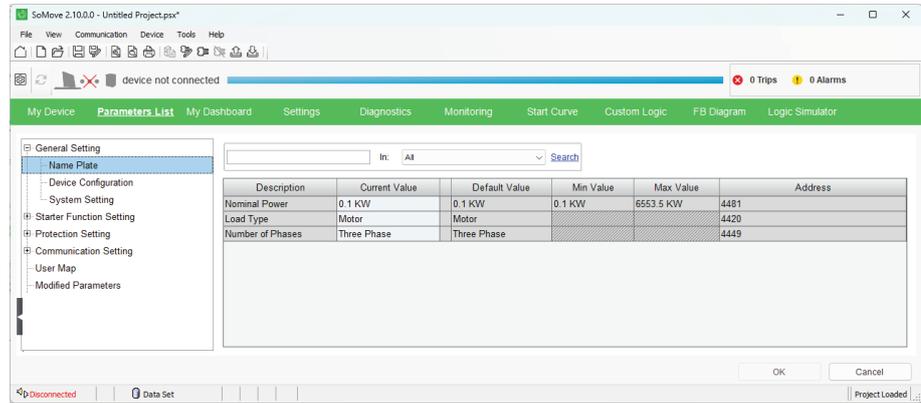
未按说明操作可能导致人身伤亡或设备损坏等严重后果。

重要信息

- 可用于配置的参数基于 TeSys Tera system 配置。
- 如果输入的参数值无效，则会显示错误消息。请阅读此消息并相应地设置值。
- 电机处于运行状态时，无法执行 **Store to Device** 操作。
- 缺省禁用了 **Communication Loss** 保护功能。如果启用了此功能且设备通过 Modbus RTU 通讯端口连接到 SoMove 软件，并且如果您尝试断开 DTM 连接，则将在设备中触发通讯丢失脱扣。
- 在电机处于运行状态时，无法配置所有设置 **Parameters List** 下方的所有设置，但 **User Map** 除外。如要配置这些参数，请停止电机，然后配置设置。
- 如果要在达到吸合级别之前触发报警，则 **Alarm Level** 参数配置应小于或等于 **Trip Level** 参数配置。
- 将系统中的电机启动和停止配置为单个继电器时，DI 启动 **Validation Time** 应大于 DI 停止 **Validation Time**。
- 如要配置 **Fieldbus Protocol Setting**、**Profibus Settings** 和 **HMI Communication** 部分下方的参数，设备应断开与 DTM 的连接。执行“存储到设备”操作以将更改的设置写入到设备后，可以通过使用提供的新参数值执行 **Scan Devices**，来连接到设备。
- 如果 **Main Unit Temperature** 值被设置为 **None**，将无法配置 **Temperature Protection** 设置。
- 如果 LTMTCUF control operator unit 通过 HMI 端口来连接，则 HMI Communication Settings 必须为如下配置：
 - **Node Address** : 1
 - **Baud rate** : 19200 bps
 - **Parity** : Even
 - **Byte Format** : Big Endian
- 在 **Starter Type** 配置为“可逆”的情况下，如果将 DI 选择为 **Forced Start**，那么在电机方向改变时，不会考虑 **Interlocking Time**。
- 有关具体参数的更多信息，请参阅 *TeSys Tera Motor Management System User Guide – DOCA0257EN*。
- 无论配置如何更改，**Default Value** 列中的值保持不变。

General Settings

General Settings 窗口包含 TeSys Tera system 的常规设置。

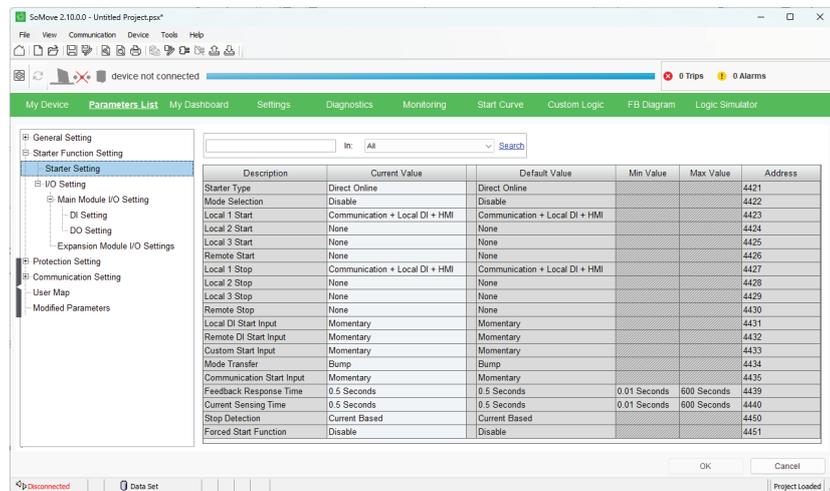


可用的子部分如下：

- **Name Plate**：包含铭牌参数。
 注: 更改 **Load Type** 或 **Number of Phases** 并保存设置将导致设备重启 (device reboot)。
- **Device Configuration**：包含 LTMT main unit 温度设置。
- **System Setting**：包含电流 (current)、电压 (voltage)、相序 (phase rotation)、标称频率 (nominal frequency) 和模式参数 (mode parameters)。

Starter Function Setting

Starter Function Setting 窗口包含可配置参数和选项，它们用于控制电机启动器的运行方式。



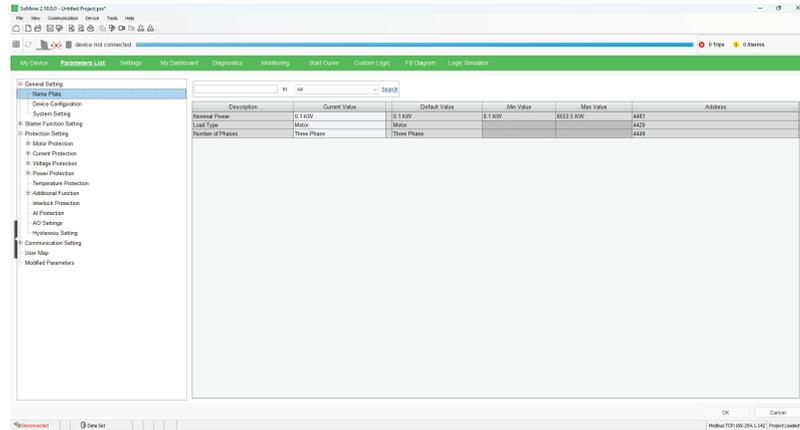
参数分为以下子部分：

- **Starter Setting**：包含与电机启动器关联的可调参数。
 注: 修改启动器设置 **Starter Setting** 下的参数并保存更改将导致设备重启 (device reboot)。
- **I/O Setting**：包含 LTMT main unit 和 LTMT expansion modules 的数字量输入和输出设置。

Protection Setting

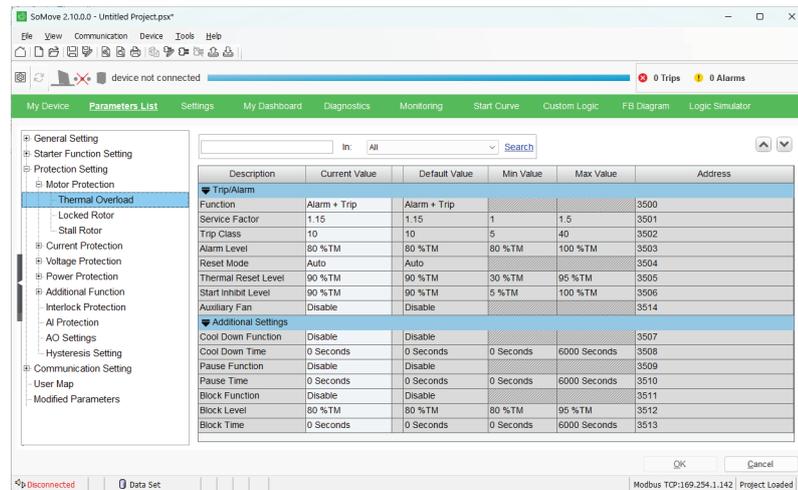
Protection Setting 窗口包含用于保护 TeSys Tera system 的参数。

注: **Protection Setting** 下可用的参数取决于所选的**负载类型** 和 **相数**。



可用的子部分如下：

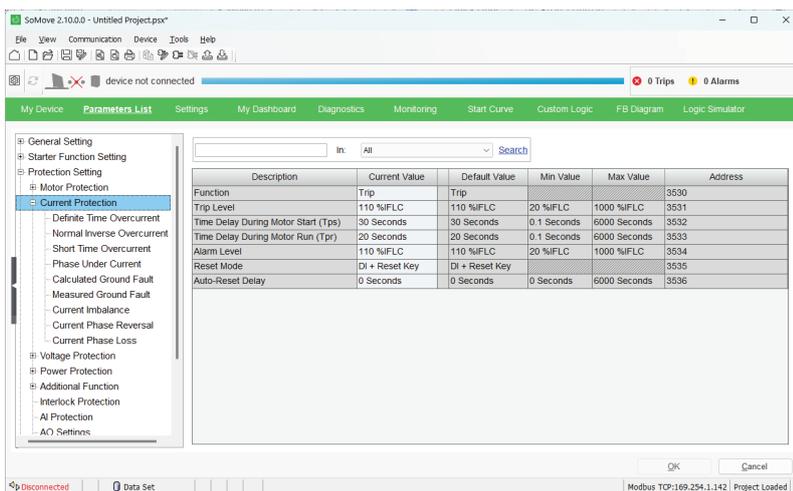
- **Motor Protection** : 包含可配置的电机保护参数。



电机保护参数分为以下几个子部分：

- **Thermal Overload**
- **Locked Rotor**
- **Stall Rotor**

- **Current Protection** : 包含可配置的电流保护参数。

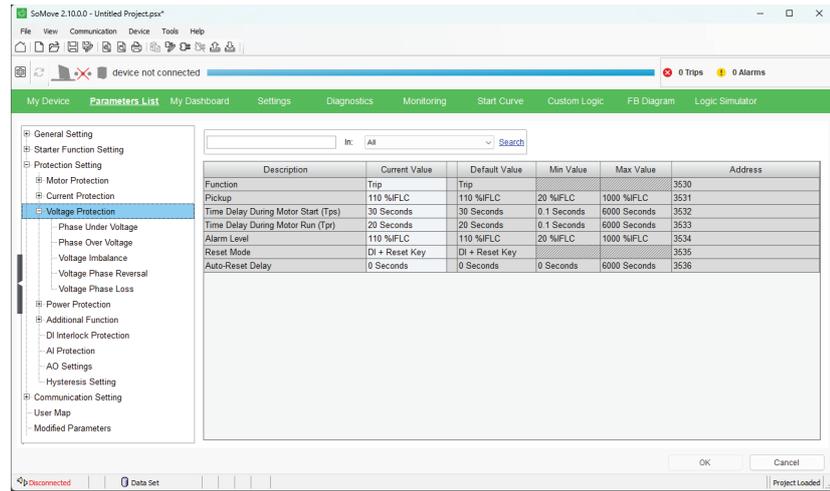


电流保护参数分为以下几个子部分：

- **Definite Time Overcurrent**
- **Normal Inverse Overcurrent**
- **Short Time Overcurrent**
- **Phase Under Current**
- **Calculated Ground Fault**
- **Measured Ground Fault**
- **Current Imbalance**
- **Current Phase Reversal**
- **Current Phase Loss**

注: **Current Protection** 功能下可用的子项取决于传感器模块的类型以及所选的**Number of Phases** 值。

- **Voltage Protection** : 包含可配置的电压保护参数。

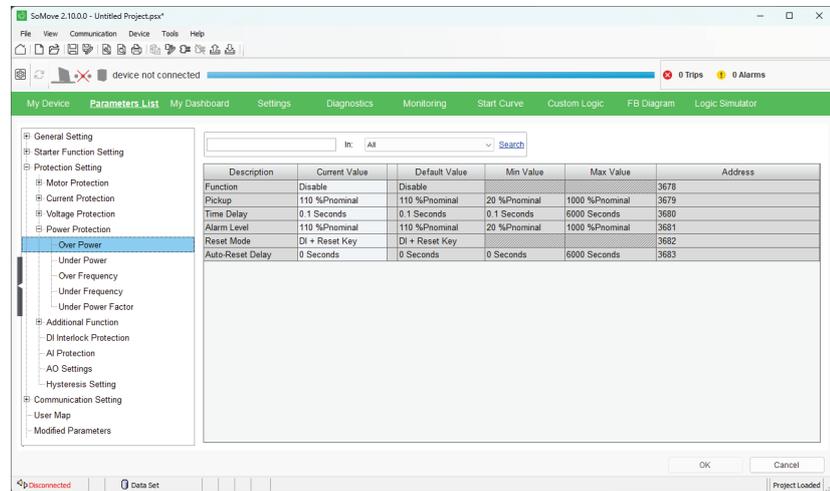


电压保护参数分为以下几个子部分：

- **Phase Under Voltage**
- **Phase Over Voltage**
- **Voltage Imbalance**
- **Voltage Phase Reversal**
- **Voltage Phase Loss**

注:

- **Voltage Protection** 设置仅适用于 LTMTCTV sensor modules.
- 在 **Voltage Protection** 取决于 **Number of Phases** 的数值。
- **Power Protection** : 包含可配置的功率保护参数。



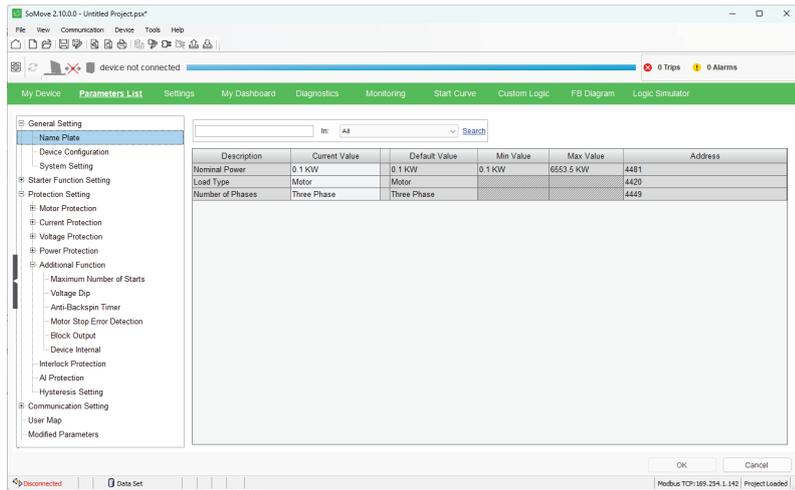
功率保护参数分为以下几个子部分：

- **Over Power**
- **Under Power**
- **Over Frequency**
- **Under Frequency**
- **Under Power Factor**

注:

- **Power Protection** 设置仅适用于 LTMTCTV sensor modules.

- **Additional Function** : 包含可配置的附加功能参数。

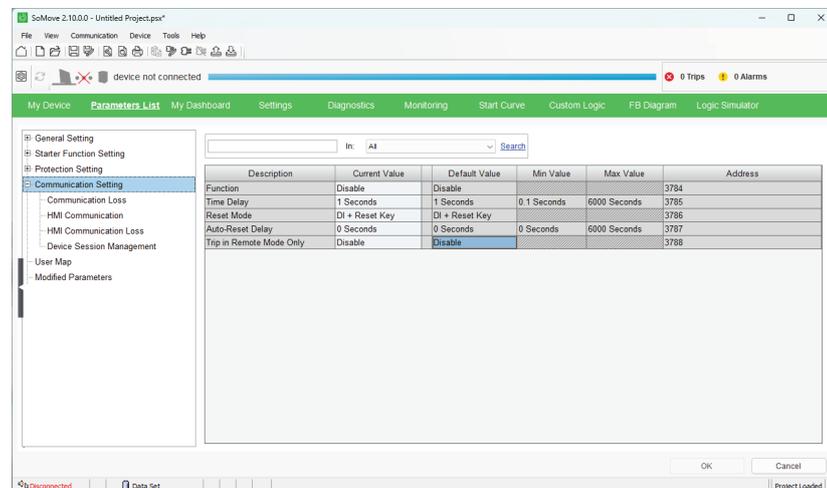


这些参数分为以下类别：

- **Maximum Number of Starts**
- **Voltage Dip**
- **Anti-Backspin Timer**
- **Motor Stop Error Detection**
- **Block Output**
- **Device Internal**
- **DI Interlock Protection** : 包含可配置的数字量输入联锁保护参数。
- **AI Protection** : 包含可配置的模拟量输入保护参数。**AI Protection** 参数仅当在 **My Device** 选项卡中配置了 LTMTAN21 expansion module 时可用。每台设备可配置两个模拟输入。
- **AO Settings** : 包含可配置的模拟量输出保护参数。**AO Settings** 参数仅在 **My Device** 选项卡中配置了 LTMTAN21 expansion module 时可用。每台设备可配置一个模拟输出。
- **Hysteresis Setting** : 包含可配置的滞后参数。

Communication Setting

Communication Setting 窗口包含通讯参数。

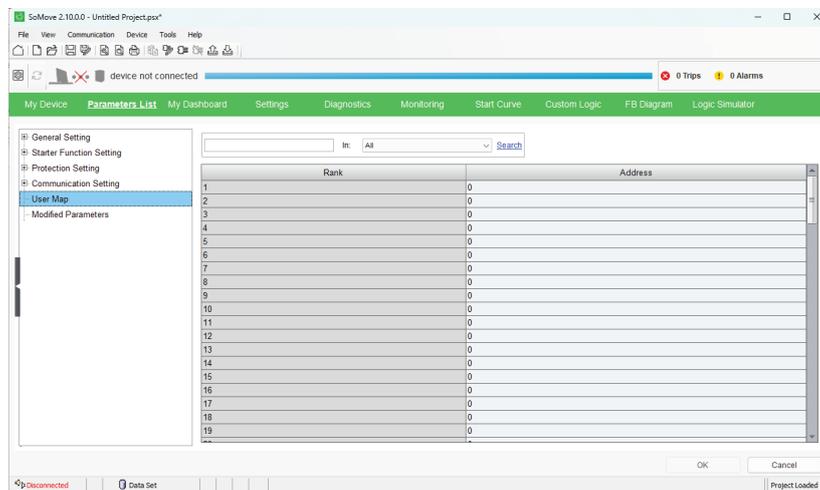


参数分为以下子部分：

- **Fieldbus Protocol Settings** 或 **Profibus Settings**: 此窗口包含 TeSys Tera system 的网络设置参数。
注: 当Field Bus Protocol 分别选择为 Fieldbus Modbus或 Profibus PROFIBUS 时, **Fieldbus Protocol Settings** 或**Profibus Settings** 界面将变为可用。
- **Communication Loss** : 此窗口包含用于配置通讯丢失设置的参数。
- **HMI Communication** : 此窗口包含 LTMTCUF control operator unit 的网络设置参数。
- **HMI Communication Loss** : 此窗口包含用于配置 LTMTCUF control operator unit 通讯丢失设置的参数。
- **Device Session Management** : 此窗口包含用于配置设备会话超时的参数。

User Map

User Map 变量设计用于优化读取和写入寄存器。

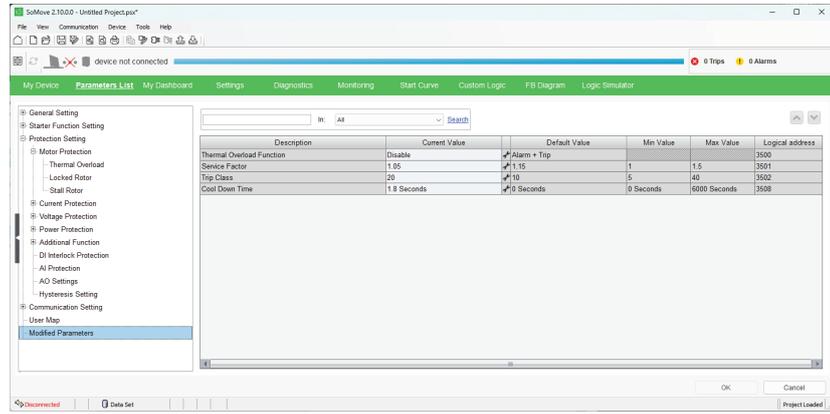


以下步骤介绍如何设置 User Map 地址 :

1. 在树视图中, 选择 **User Map** :
 - 地址排序为 1 至 100。
 - 地址分为四组。
2. 在表中输入一个地址值 :
 - 输入的地址必须为十进制格式, 其范围应为 0 至 9249。
 - 输入地址 0 以从 User Map 中删除地址。
3. 按“输入”确认新地址 :
 - 如果接受该地址, 则该地址将被添加到 User Map。
 - 如果不接受该地址, 则以前接受的地址将保持在 User Map 中。

Modified Parameters

Modified Parameters 窗口显示其值已更改为缺省设置的参数的列表。



Settings

此章节内容

General	61
Communication	62
安全性	64

概况

Settings 选项卡可用于配置 Ethernet 设置 LTMT main unit.

设置值可在连接和断开两种模式下进行修改。

在连接模式下，可以通过选择 **OK**，将设置更新到设备。

在断开模式下，可以通过执行 **Store to Device** 操作，将配置设置下载到设备。

注:

- 仅当通信方式选择为**Modbus TCP** 或选定LTMT main unitEthernet变体时，**Settings**选项卡才可用。
- 当Ethernet 变体通过串行接口连接时，LTMT main unit 设置选项卡下的参数仅可查看，
- **Settings** 选项卡中的值仅更新一次。若要在在线模式下查看更新后的值，请使用刷新  图标执行进行刷新。

说明

在**Settings** 选项卡的树形视图中，包含以下分类：设置值在连接与断开状态下均可修改：

- **General**
- **Communication**
- **Security**

General

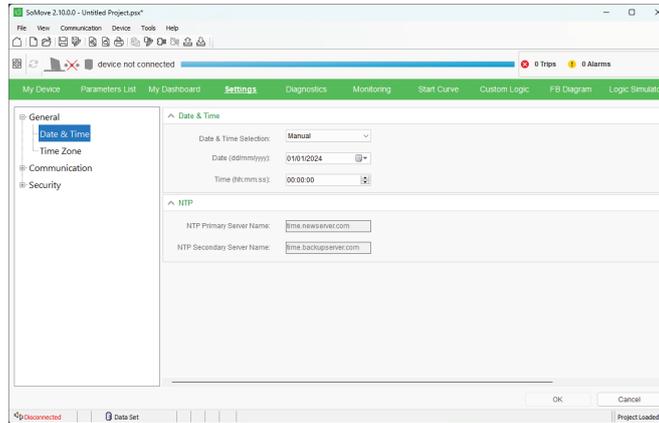
General 窗口包含以下通用设置LTMT main unit:

- **Date & Time**
- **Time Zone**

Date & Time

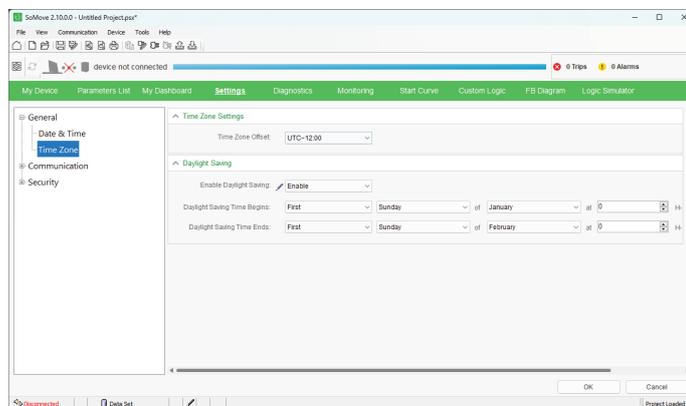
Date & Time 窗口用于配置 LTMT main unit的日期和时间设置。以下选项可在 **Date & Time Selection** 下拉菜单中提供以下选项：

- **Manual** –分别使用**Date (dd/mm/yyyy)** 和 **Time (hh:mm:ss)** 字段手动设定日期和时间。
- **NTP** – 日期和时间是从 **NTP Primary Server Name** 和 **NTP Secondary Server Name** 字段中设置的网络时间协议 (NTP) 服务器获取。
- **Field Bus Protocol** – 日期和时间从已配置的现场总线协议获取。



Time Zone

Time Zone 窗口用于配置LTMT main unit的时区和夏令时设置。



要设置时区，请使用 **Time Zone Offset** 下拉列表中选择所需时区。

要启用夏令时，请在**Enable Daylight Saving**下拉列表中选择**Enable**。在**Daylight Saving Time Begins** 和 **Daylight Saving Time Ends**字段中提供所需的设置。

Communication

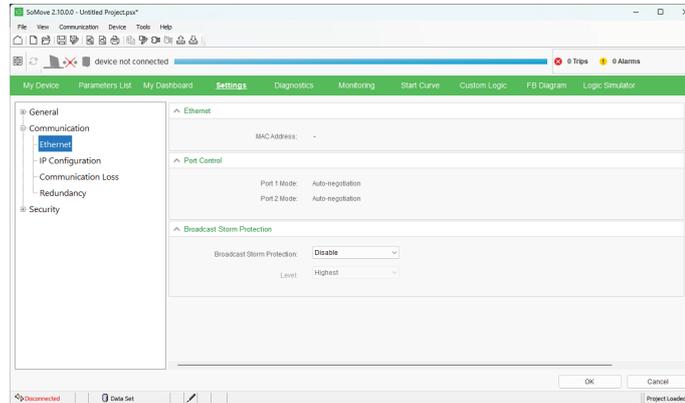
Communication 窗口显示并允许配置通信设置。LTMT main unit可用的子部分如下：

- **Ethernet**
- **IP Configuration**
- **Communication Loss**

Ethernet

Ethernet 窗口将显示以下信息：

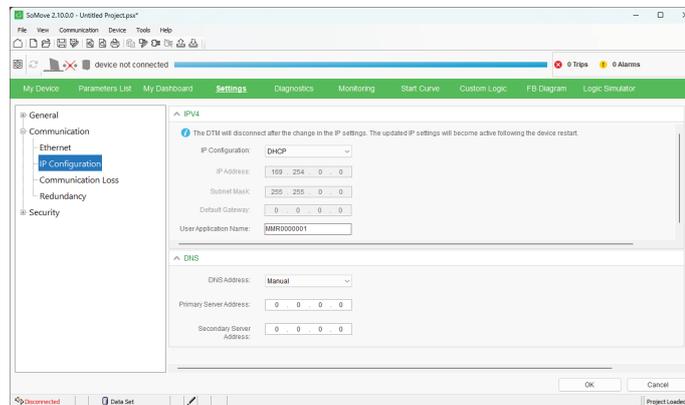
- **MAC Address**
- **Port 1 Mode**
- **Port 2 Mode**



可通过 **Broadcast Storm Protection** 下拉列表启用风暴保护广播功能。使用以下选项选择风暴防护广播的级别：**Level** 下拉列表选择风暴防护广播的级别。

IP 配置

IP Configuration 窗口允许您配置IPV4和DNS LTMT main unit设置。



IP Configuration 可设置为以下选项：

- **DHCP** – 该 **IP Address**, **Subnet Mask**、**Default Gateway** 由动态主机配置协议 (DHCP) 服务器自动分配。
- **Fixed (Static IP)** – 允许您设置 **IP Address**, **Subnet Mask**、**Default Gateway** 设置。

DNS Address 设置可配置为以下内容：

- **Automatic** – **Primary Server Address** 和 **Secondary Server Address** 由 DHCP服务器自动分配。
- **Manual** – 允许您设置 **Primary Server Address** 和 **Secondary Server Address**。

注: 当 **IP Configuration** 设置为 **Fixed (Static IP)**时，**DNS Address** 将位于 **Manual** 模式。

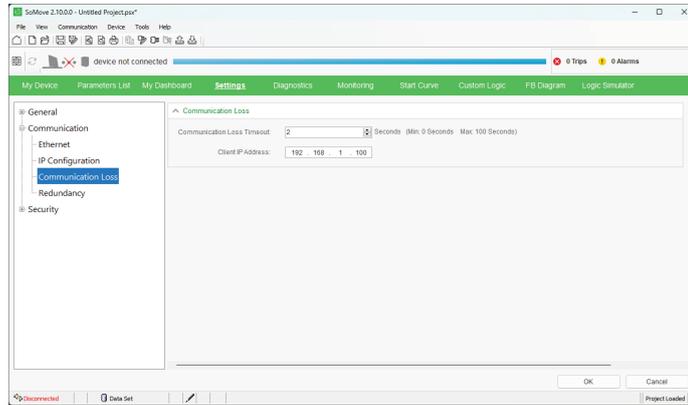
User Application Name 字段允许您为应用程序提供自定义名称。

注: 仅支持字母数字字符。

通讯中断

Communication Loss 窗口允许您配置以下设置：

- **Communication Loss Timeout**
- **Client IP Address**

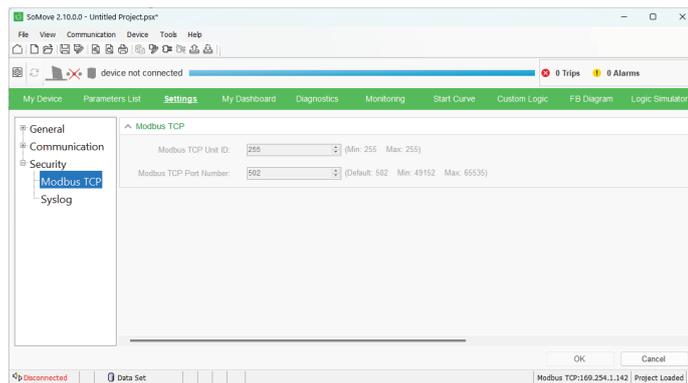


安全性

该 **Security** 设置允许配置系统的 Modbus TCP 安全设置并下载其系统日志。LTMT main unit

Modbus TCP

Modbus TCP 窗口显示 **Modbus TCP Unit ID**，并允许您设置 **Modbus TCP Port Number**。



Syslog

可通过选择 **Export** 导出选项将系统日志信息导出为 CSV 文件。

My Dashboard

概述

My Dashboard 选项卡让您能够根据需求自定义设置要进行配置或监控的数据。它用于：

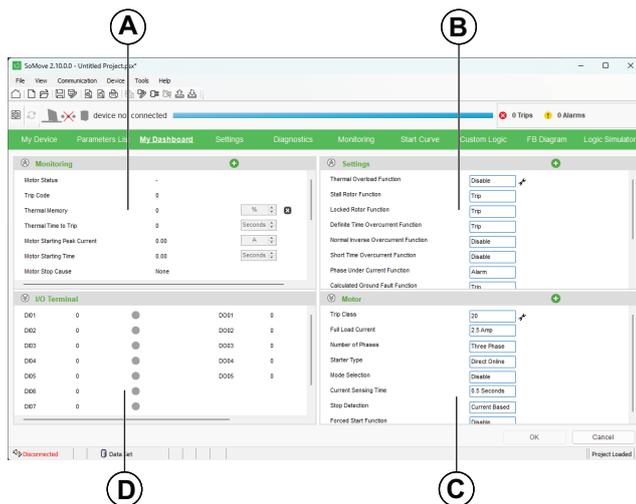
- 显示启动器相关参数的实时值。
- 显示输入/输出端子的信息。
- 配置启动器和电机相关参数。
- 配置保护功能参数。

注: 可以使用**Save Project** 选项来保存 **My Dashboard** 设置。如果保存项目并打开已保存的项目，则会保留这些设置。

描述

工作区分为两个区域：

- **显示区域**：显示启动器参数的实时值以及输入/输出端子的状态以供分析。
- **设置区域**：设置启动器和电机参数。



- A Monitoring
- B Settings
- C Motor
- D I/O terminals

Monitoring

Monitoring 区域让您能够监控所需的参数。

Settings

Settings 区域让您能够根据需求监控和更改保护功能参数值。可通过下拉列表为 **Settings** 区域下的参数分配所需的值。

Motor

Motor 区域让您能够根据需求监控和更改电机参数值。

I/O Terminals

I/O Terminals 区域让您能够监控 I/O 信号的状态。

下表描述了LTMT main unit 的输入/输出的状态以及 LTMT expansion modules 具有数字输入或输出的状态。

输入/输出状态	状态框颜色	描述文本
已激活	绿色	指示 IO 端子已分配且已激活。
未激活	灰色	指示 IO 端子已分配且未激活。

添加参数

以下步骤显示了如何添加参数：

1. 在需要添加参数的区域中，选择  图标。
结果： 随即显示 **Select Parameter** 对话框。
2. 在 **All Parameters** 列中，选择所需的参数，然后选择**向右箭头**。搜索栏可用于查找 **All Parameters** 列下方的可用参数。
结果： 参数被移动到 **Selected Parameters** 列。
3. 如要对参数列表重新排序，请在 **Selected Parameters** 列中选择所需参数，然后选择**向上/向下箭头**。
4. 选择 **OK**。

已添加的参数显示在相应的显示区域中。

删除参数

如要从区域中删除参数，请选择参数前方的  图标。

以下步骤显示了如何删除多个参数：

1. 在需要删除参数的区域中，选择  图标。
结果： 随即显示 **Select Parameter** 对话框。
2. 在 **Selected Parameters** 列中，选择所需的参数，然后选择**向左箭头**。
结果： 参数被移动到 **All Parameters** 列。
3. 选择 **OK**。

这些参数将从相应的显示区域中删除。

注： 只有 **Settings** 和 **Motor** 区域下方的参数列表才可配置。

Diagnostics

概况

Diagnostics选项卡显示了电机的状态、报警或脱扣状态、日志、设备内部状态、通信状态以及 TeSys Tera system脱扣计数器信息。

该选项卡中的数据仅在连接模式下很重要。

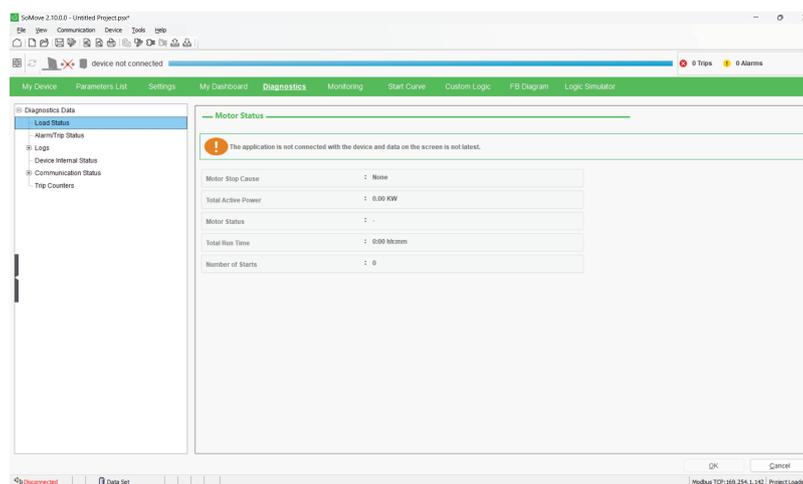
Diagnostics选项卡中的数据是动态更新的，每秒自动刷新一次（**Logs**（日志）节点除外）。

说明

Diagnostics 选项卡中显示了以下参数：

- **Motor or Load Status**
- **Alarm or Trip Status**
- TeSys Tera system 的日志，其中包括：
 - **Events**
 - **Device Internal**
 - **Trips**
- **Device Internal Status**
- **Communication Status**
- **Trip Counters**

Motor or Load Status



Motor Status 或 **Load Status** 窗口分别显示所配置的电机或加热器的参数。

Motor Status 或 **Load Status** 窗口下方显示了以下电机参数：

- **Motor Stop Cause**：停止电机运行的原因。
- **Total Active Power**：电机或加热器运行期间消耗的总功率。
- **Motor Status** 或 **Load Status**：电机或加热器的当前状态。
- **Total Run Hour**：电机或加热器的总运行时间。
- **Number of Starts**：电机或加热器的总启动次数。

Alarm or Trip Status

Alarm or Trip Status 窗口显示与报警或脱扣相关的电机参数状态。

通过在搜索框中添加关键字，在 **In** 下拉列表下方选择所需的选项，并通过选择 **Search**，可以搜索所需参数。

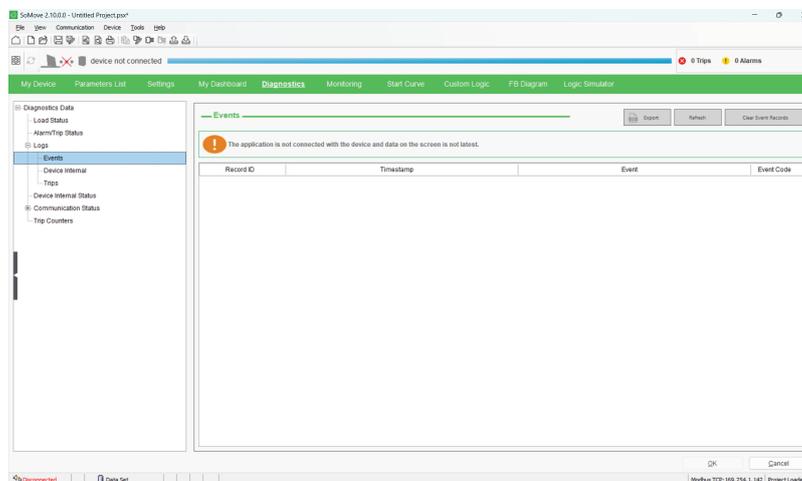
指示灯颜色		状态
灰色（关闭状态）		无脱扣或报警
琥珀色		报警
红色		脱扣

Logs

Logs 窗口显示以下类别的日志：

- **Events**
- **Device Internal**
- **Trips**

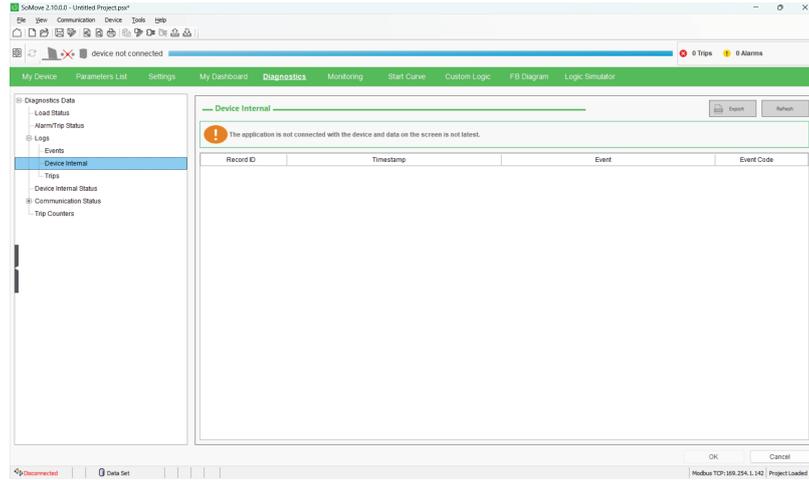
Events



Events 窗口显示 TeSys Tera system 的事件记录列表。

窗口中将显示最多100条事件日志，每条日志包含记录ID、时间戳、事件描述以及事件代码。记录ID 1 表示 最新记录的事件，而记录ID 100 表示最久记录的事件。如果窗口上有 100 个事件日志，且出现了新事件，则会从列表中删除最早的事件，并将新事件添加到列表顶部。

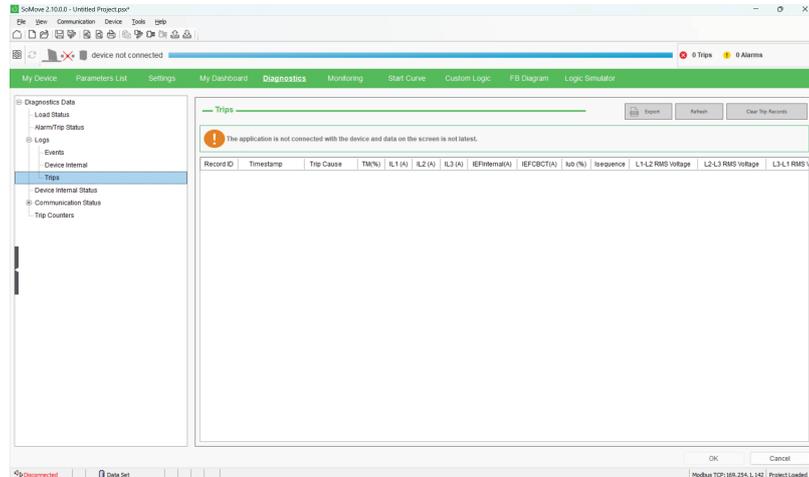
Device Internal



Device Internal 窗口提供 TeSys Tera system 的内部继电器事件列表。

窗口将显示最多20条内部中继日志，每条日志包含记录ID、时间戳、事件描述以及事件代码。记录ID 1是窗口中最新检测到的错误记录，记录ID 20是最旧检测到的错误记录。如果窗口中存在20条错误日志且检测到新错误时，最旧的错误记录将被移除列表并添加新错误记录到列表顶部。

Trips



Trips 窗口提供 TeSys Tera system 的所有脱扣记录。

窗口中最多显示20个脱扣日志，这些日志包含时间戳、脱扣原因、脱扣出现时的重要设置值、检测到脱扣时记录的测量值、以及脱扣代码。记录ID 1表示最新的脱扣记录，而记录ID 20 表示最久的脱扣记录。如果窗口中有 20 个脱扣记录，并且检测到新的脱扣，则将从列表中删除最早的脱扣记录，并将新脱扣记录添加到列表顶部。

Export

Export 功能可将记录列表以CSV格式导出。

如要导出记录列表，请选择 **Export**，选择目标文件夹，然后选择 **Save**。

注: 只有在存在至少一个记录数据时，导出选项才可用。

Refresh

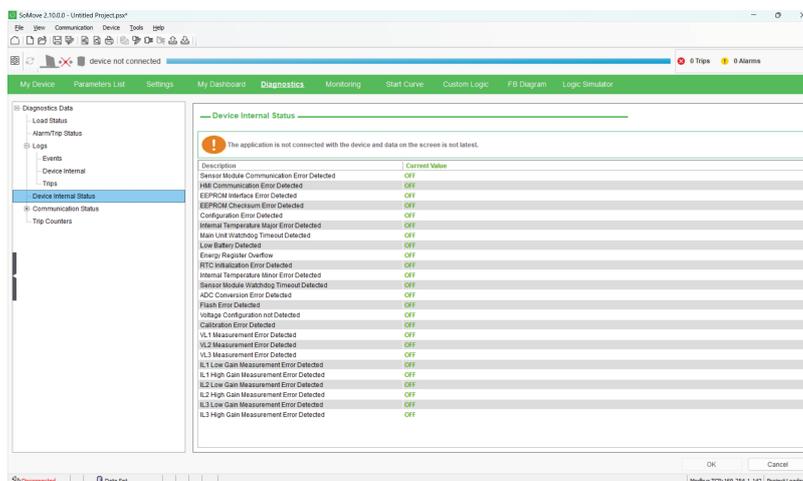
Refresh 功能更新记录列表并显示最新列表。

在连接模式下，可以通过选择 **Refresh** 来更新 Data Record 节点。

Clear Event Logs 或 Clear Trip Logs

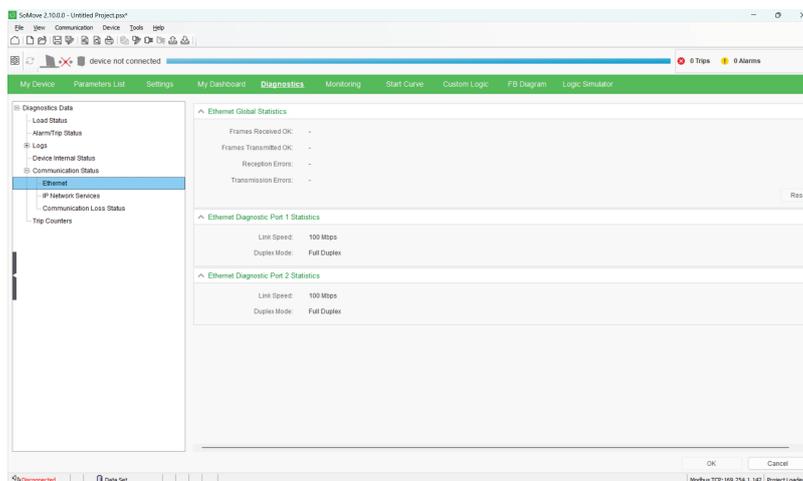
通过在相应的窗口中选择 **Clear Event Logs** 或 **Clear Trip Logs** , **Clear Event Logs** 或 **Clear Trip Logs** 可清除事件或脱扣列表。

Device Internal Status



Device Internal Status 窗口显示 TeSys Tera system 不同模块的通讯和运行状态。

Communication Status

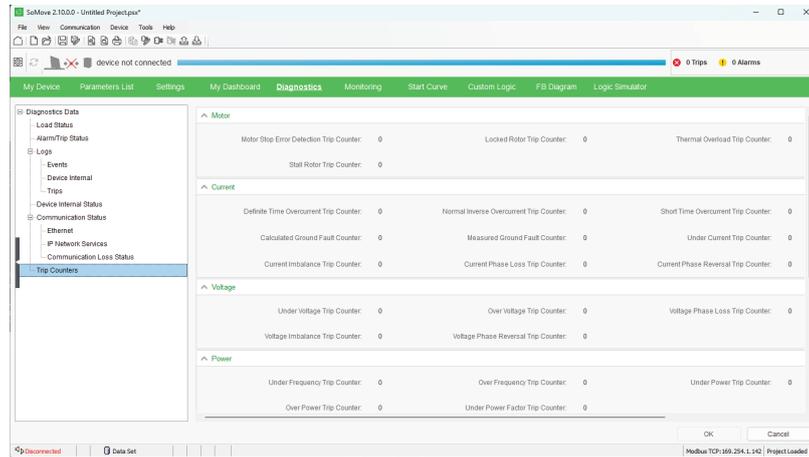


Communication Status 窗口显示 TeSys Tera system 的通讯参数状态。

当在TeSys Tera system使用的LTMT main unit是Ethernet变体时，可使用以下子部分：

- **Ethernet** – 显示 **Ethernet Global Statistics**, **Ethernet Diagnostics Port 1 Statistics** , 以及LTMT main unit的**Ethernet Diagnostics Port 2 Statistics**
- **IP Network Services** – 显示 **Modbus TCP Port** 状态。
- **Communication Loss Status** – 显示 LTMT main unit通信丢失状态。

Trip Counters



Trip Counters 窗口显示 TeSys Tera system 内每个类别的脱扣次数。

Monitoring

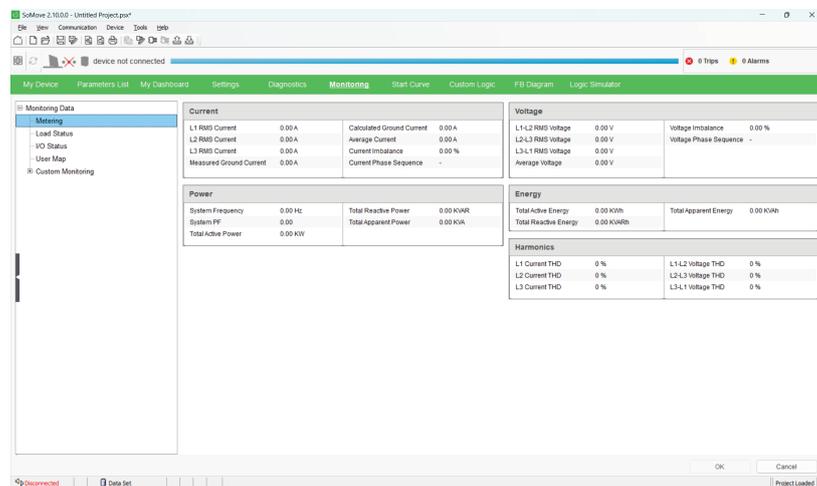
概述

Monitoring 选项卡显示对 TeSys Tera system 相关信息 监控。

显示的监控信息有以下类别：

- **Metering**
- **Load Status**
- **I/O Status**
- **User Map**
- **Custom Monitoring**

Metering



Metering 窗口显示 TeSys Tera system 的测量数据。

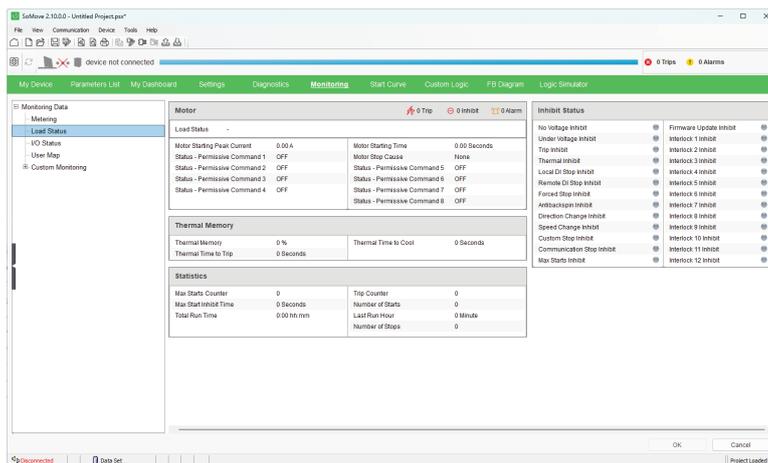
测量信号分为以下类别：

- **电流**
- **电压**
- **功率**
- **电能**
- **温度**
- **Harmonics**

注: **Metering** 窗口中显示的参数取决于以下位置的配置：

- **My Device**选项卡
- **General settings > Device Configuration**
- **General settings > Name Plate > Number of Phases**

Load Status



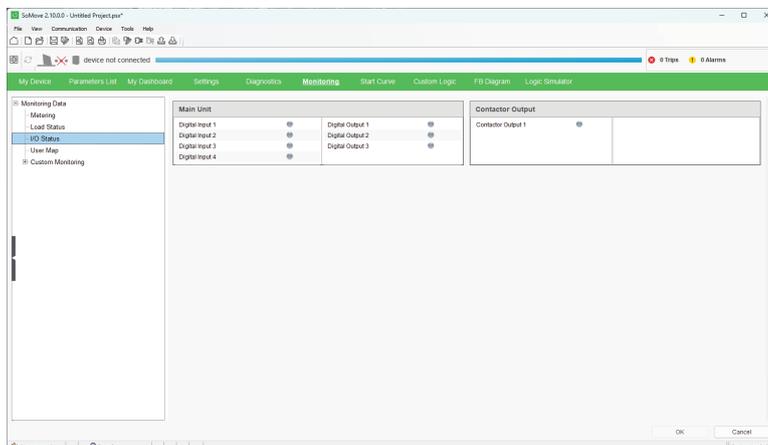
Load Status 窗口显示 TeSys Tera system 的负载状态。

负载状态页面下的参数分为以下类别：

- Motor
- Thermal Memory
- Statistics
- Inhibit Status

Load Status 窗口 还会在 **电机 (Motor)** 分类下显示 脱扣 (trips)、禁止 (inhibits) 和告警 (alarms) 的总次数。

I/O Status

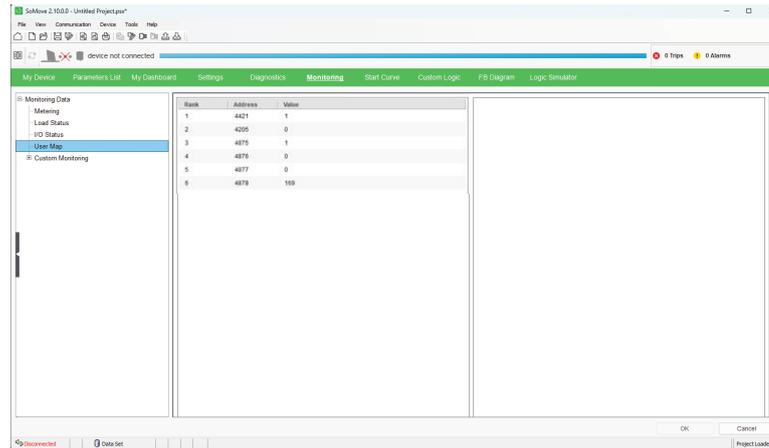


I/O Status 窗口显示设备的模拟量和数字量输入输出状态。 TeSys Tera system

I/O 信号分为以下类别：

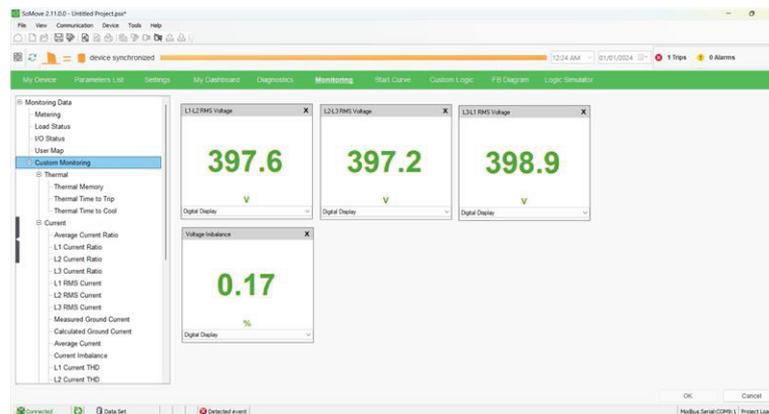
- Main Unit
- Contactor Output
- Expansion Module

User Map



User Map 窗口显示在 **Parameters > User Map** 部分下方配置的参数的实时值。最多显示 100 个参数。

Custom Monitoring



Custom Monitoring 窗口让您能够创建包含小部件的个性化监控窗口。

信号分为以下类别：

- **Thermal**
- **电流**
- **电压**
- **功率**
- **Motor**

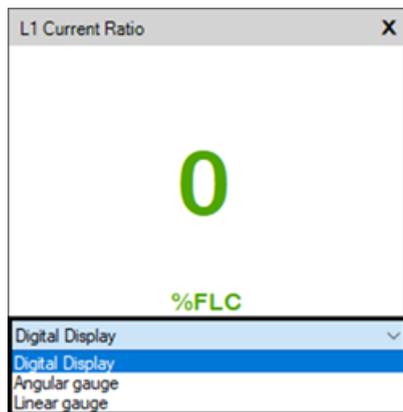
自定义 Custom Monitoring 窗口

以下步骤介绍如何添加小部件：

1. 在可用信号类别下方选择要监控的信号。
2. 单击 **Custom Monitoring** 窗口的显示区域。

结果：所选信号的小部件被添加。

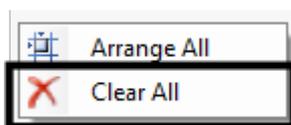
您可以通过从下拉列表中选择选项来修改小部件显示选项。



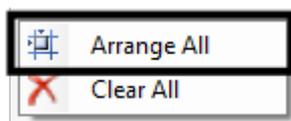
如要从显示区域中删除小部件，请选择  图标。

可以对显示区域执行以下操作。

- 如要删除所有小部件，请右键单击显示区域，然后选择 **Clear All**。



- 如要组织已添加的小部件，请右键单击显示区域，然后选择 **Arrange All**。

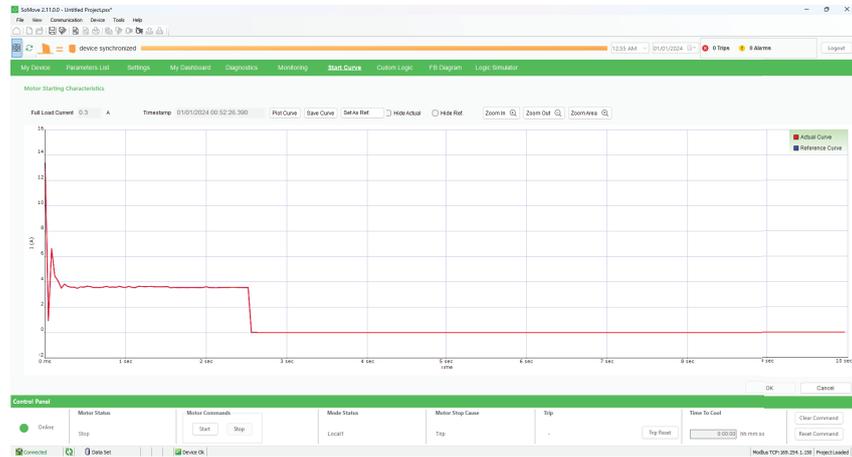


Start Curve

概述

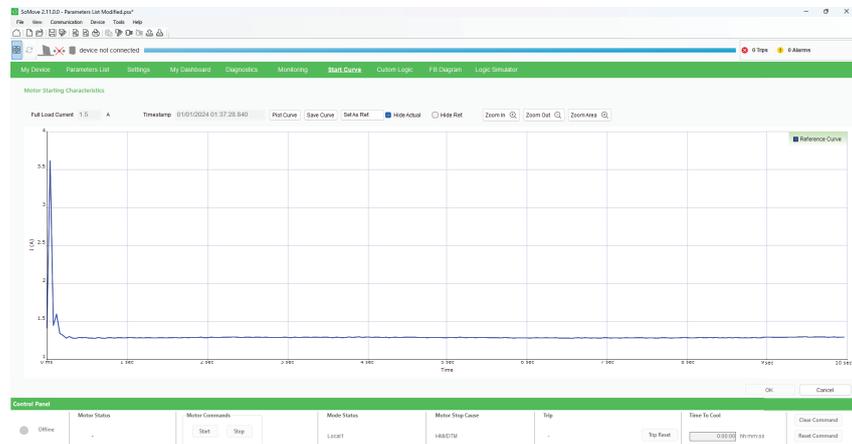
Start Curve 选项卡显示在电机启动时间和采样时间间隔内的电机电流 (I) 图。**Start Curve** 选项卡用于从曲线分析电机启动特性。

执行出厂复位后，将从设备擦除启动曲线数据。



参考曲线

通过单击 **Set as Ref.**，可将绘制的曲线设置为参考曲线。在将新曲线设置作为参考之前，参考曲线将保持相同。蓝色图线表示参考曲线。



缩放显示选项

提供以下选项：

- **Zoom Out:** 用于立即查看程序的更多信息。
- **Zoom In:** 用于重点查看特定程序的细节。
- **Zoom Area:** 用于自定义程序视图。

按键

- **Full Load Current:** 显示满载电流值（单位：安培）。
- **Timestamp:** 显示电机启动时间。
- **Plot:** 读取最新值并刷新电机启动特性曲线。
- **Save:** 捕捉电机启动特性曲线的屏幕截图，并允许您将其保存到 PC。
- **Set As Ref.:** 将实际曲线设置作为参考曲线。

Custom Logic

概述

Custom Logic 选项卡显示 Custom Logic Editor，该编辑器让您能够创建新的自定义逻辑程序文件或者打开并编辑任何现有的自定义逻辑程序文件。

有关 **Custom Logic** 选项卡的更多信息，请参阅 [Custom Logic Editor, 92 页](#)

FB Diagram

概述

FB Diagram 选项卡用于显示功能块图编辑器，该编辑器让您能够创建新的功能块图文件或者打开并编辑任何现有的功能块图文件。

有关 **FB Diagram** 选项卡的更多信息，请参阅 [FBD 编辑器简介](#), 159 页

Logic Simulator

概述

Logic Simulator 选项卡用于显示 LTMT main unit **Logic Simulator**，它让您能够测试自定义逻辑程序的功能。

有关 Logic Simulator 的详细信息，请参阅 LTMT Main Unit 逻辑仿真器, 191 页。

用户功能

此部分内容

固件更新.....	82
人机界面 (HMI) 语言更新	86
Factory Reset	88
Pin 码管理	89

固件更新

概况

TeSys Tera system和HMI设备的固件可通过TeSys Tera DTM。

固件可通过以下模式进行更新：

- TeSys Tera system – 在线模式 (online mode)
- HMI 设备 – 离线模式 (offline mode)

注: 固件更新无法从 SoMove 软件主页执行固件更新。

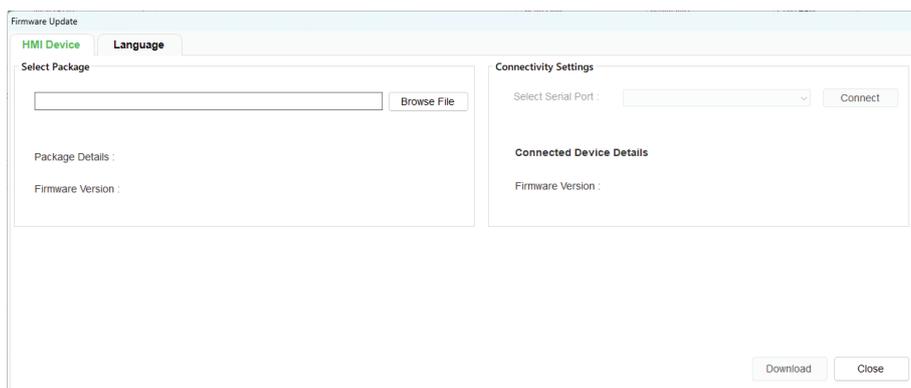
人机界面 (HMI) 设备固件更新

HMI 设备的固件可通过离线模式的DTM进行更新。

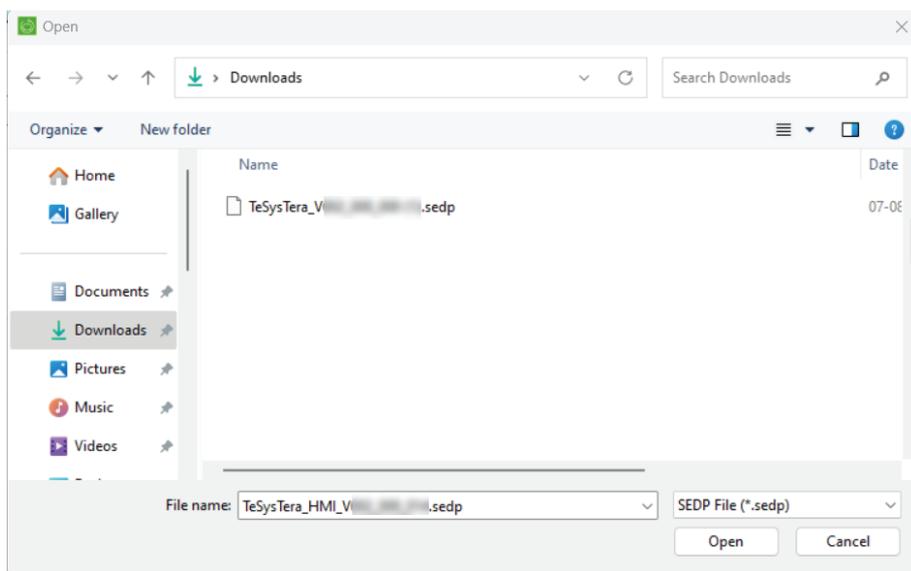
按照以下步骤，从服务器更新HMI设备的固件：

1. 在菜单栏上，选择**Device > Maintenance > Firmware Update**。

结果： 随即显示 **Firmware Update** 窗口。



2. 选择 **Browse File** 浏览并选择电脑中的固件包文件，如下图所示：



3. 选择 **Open**。
4. 使用 **Select Serial Port** 下拉列表选择端口，然后选择 **Connect**。

- 5. 按住 LTMTCUF 上的“Stop”和“Reset”按钮，同时通过连接 LTMT 来为 LTMTCUF 供电。Then press OK 出现。

请按照说明在 HMI 设备上进行操作。

- 6. 执行上述操作后，选择 OK。
- 7. 选择 Download。

注: 操作需要 4 至 12 分钟来更新固件。

- 8. 重启HMI 设备。

结果 : HMI 设备的固件已更新。

TeSys Tera系统固件更新

可以使用以下资源之一更新 TeSys Tera system 的固件 :

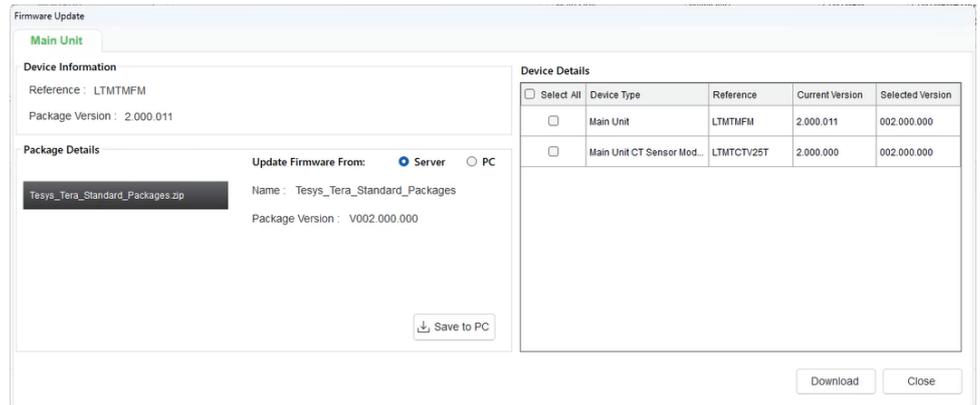
- Server
- PC

注: 确保电脑未连接到 HMI 端口 LTMTCUF control operator unit。

在菜单栏上，选择Device > Maintenance > Firmware Update。

注: TeSys Tera system 固件更新可在线模式下执行。

结果 : 随即显示 Firmware Update 窗口。



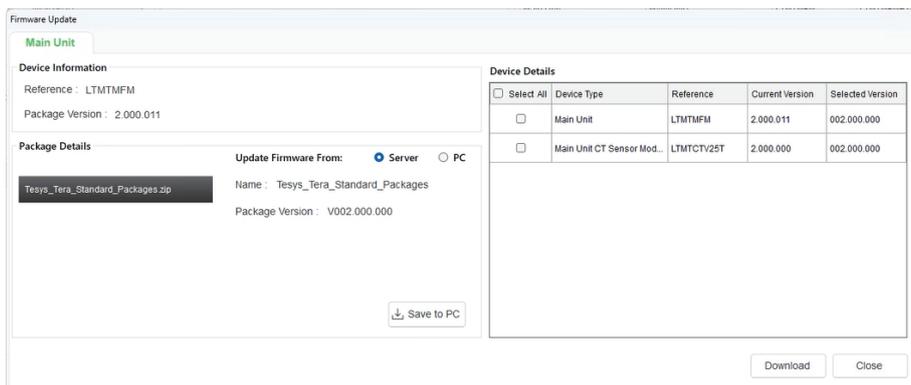
从服务器更新固件

按照以下步骤，从服务器更新固件 :

- 1. 在 Firmware Update 窗口中，选择 Server 在 Update Firmware From 选项中选择

2. 从以下列表中选择固件包：**Firmware Package List** 并选择 **Download**。

注：固件更新适用于 Ethernet 的固件更新 TeSys Tera system 固件更新通过完整包上传方式执行。无法执行单个模块的更新。



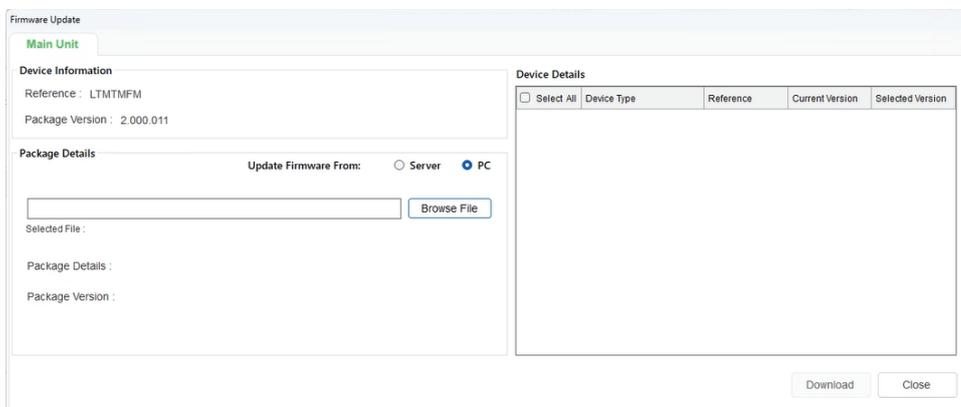
注：为加快固件更新过程，请选择设备支持的最高波特率（115200）。

结果：在 **Firmware Update** 窗口底部显示了更新成功消息。

将固件下载到 PC

选择  图标下载固件包，并将其保存至您的 PC。

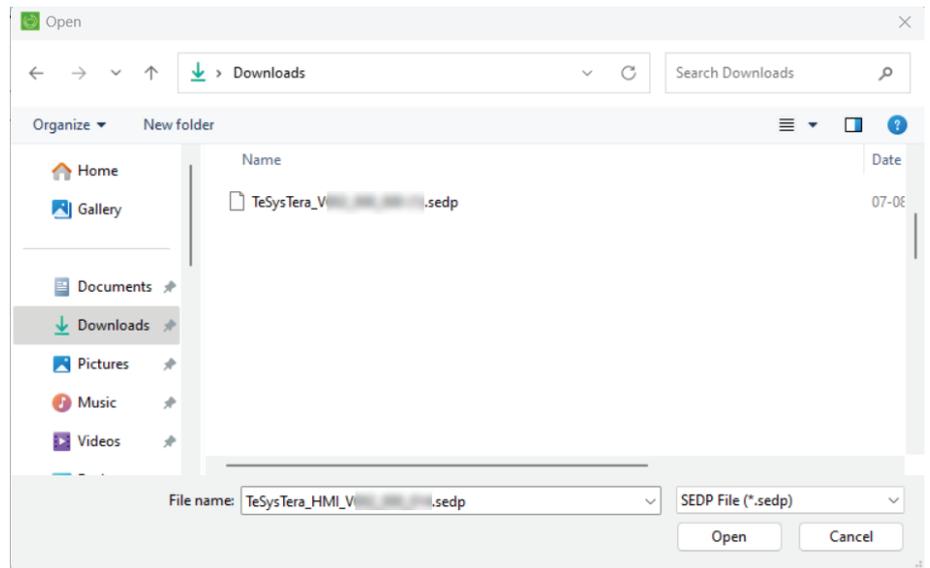
从 PC 更新固件



要从 PC 更新固件，请执行下列步骤：

1. 在 **Firmware Update** 窗口中，选择 **PC** 选项。

2. 选择 **Browse File** 浏览并选择电脑中的固件包文件，如下图所示：



3. 选择 **Open**。

结果：在 **Firmware Update** 窗口底部显示了更新成功消息。

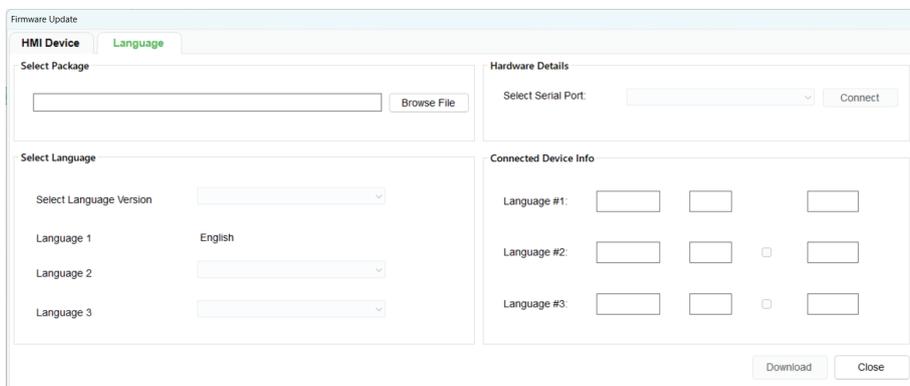
4. 如果更新失败，将显示 **Firmware Update Failed** 对话框。再次执行上述操作程序，以更新固件。

人机界面 (HMI) 语言更新

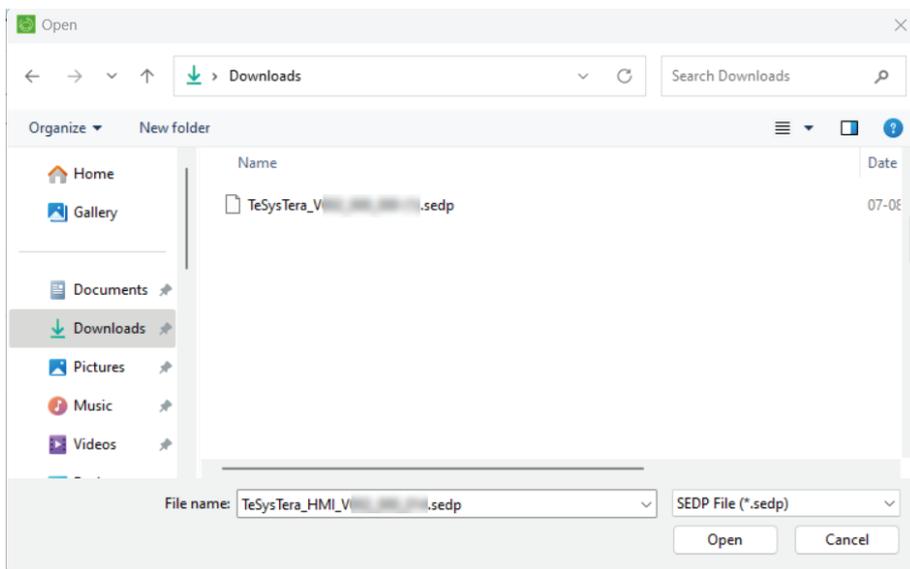
该 HMI 语言更新可通过以下方式执行：TeSys Tera DTM。

以下步骤说明如何更新语言设置：HMI:

1. 在菜单栏上，选择 **Device > Maintenance > Firmware Update**。
2. 导航至 **Language** 选项卡。



3. 点击 **Browse File** 浏览并选择电脑中的固件包文件，如下图所示：



4. 单击 **Open**。

5. 在 **Select Language** 部分，通过相应的下拉菜单选择语言版本及使用语言。

注: 支持下列语言：

- Czech
- 巴西葡萄牙语
- 丹麦语
- German
- 中文
- Spanish
- 波斯语
- Finnish
- French
- 希伯来语
- 印尼语
- 意大利语
- 韩语
- Dutch
- Portuguese
- 俄语
- Chinese

6. 使用 **Select Serial Port** 下拉列表选择端口，然后点击 **Connect**.

7. 当前语言的详细信息 HMI 显示在 **Connected Device Info**.

注: 可通过复选框启用或禁用这些语言。

8. 选择 **Download**.

9. 重新启动 HMI 设备。

结果: 该设备的语言 HMI 设备语言已更新。

Factory Reset

概述

Factory Reset 功能用于将 TeSys Tera system 参数复位为缺省值。

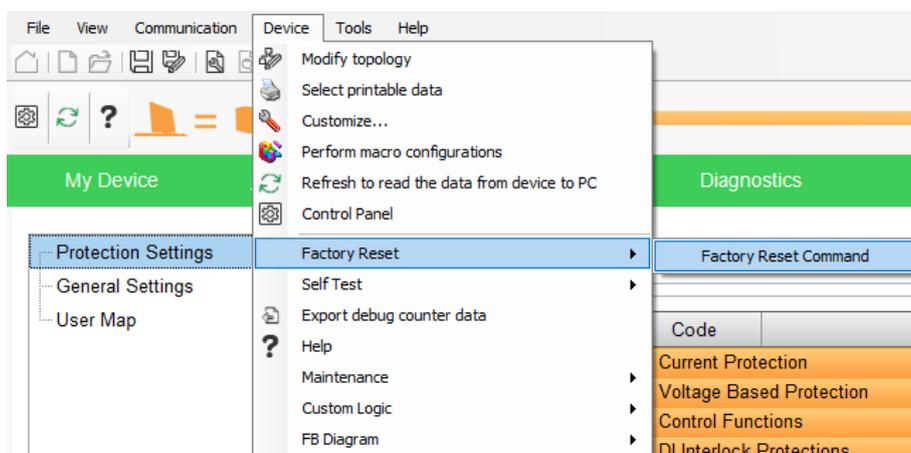
注: 当设备LTMT main unit通过串行接口（HMI端口）连接时，工厂复位功能 TeSys Tera DTM支持以下Ethernet 型号。

执行工厂复位命令

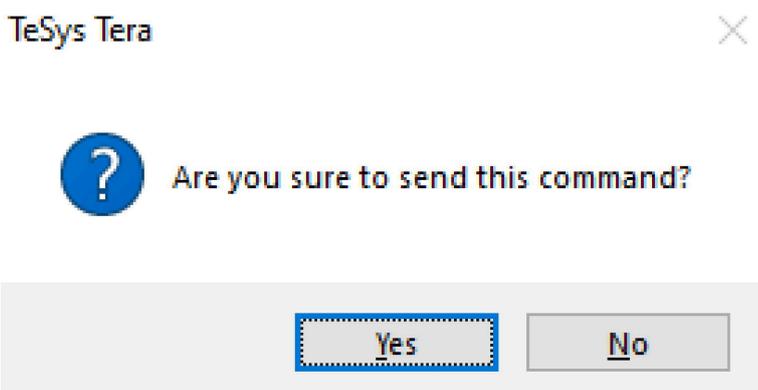
如要执行工厂复位命令，请执行以下步骤：

注: 工厂复位功能仅在连接模式下工作。

1. 选择 **Device > Factory Reset > Factory Rest Command**，如下图所示。



2. 选择该命令后，将出现一个确认弹出消息。



注: 在执行了此命令后，TeSys Tera DTM Library 将断开与 TeSys Tera system 的连接。

在执行了**Factory Reset** 命令后，此功能会将 TeSys Tera system 的配置恢复成出厂设置，然后重启 TeSys Tera system。

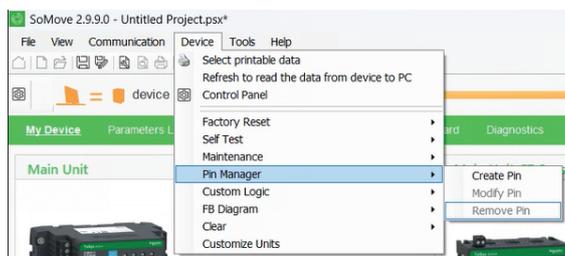
Pin 码管理

Pin Manager 通过允许您在 SoMove 软件上设置 Pin 码，来保证设备的安全。

可以通过导航至 **Device > Pin Manager** 来访问 **Pin Manager**。

注:

- 仅在设备连接到 SoMove 软件时，**Pin Manager** 才可用。
- 对于该设备的不同变体 Ethernet (variants)，如果 LTMT main unit 连接方式为通过串行接口 (HMIPort)，则支持 **Pin Manager** 功能。



在 **Pin Manager** 下方，可以执行以下操作：

- Create Pin
- 修改 Pin 码
- Remove Pin

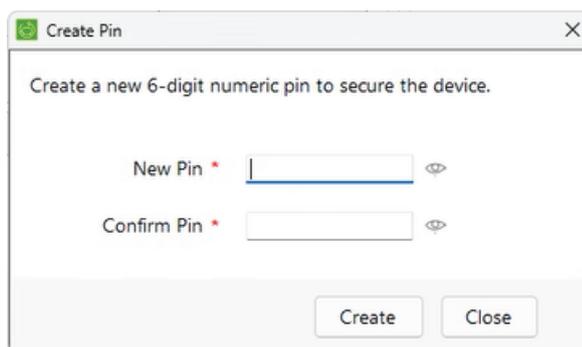
创建 Pin 码

Create Pin 选项让您能够创建新 Pin 码。只有在未设置 Pin 码时，**Create Pin** 选项才可用。

以下步骤显示了如何创建 Pin 码：

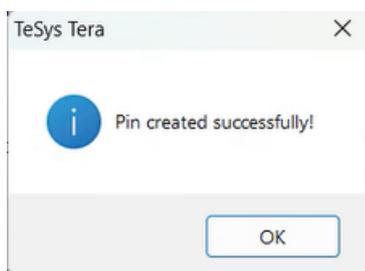
1. 导航至 **Device > Pin Manager > Create Pin**，。

结果： 随即显示 **Create Pin** 对话框。



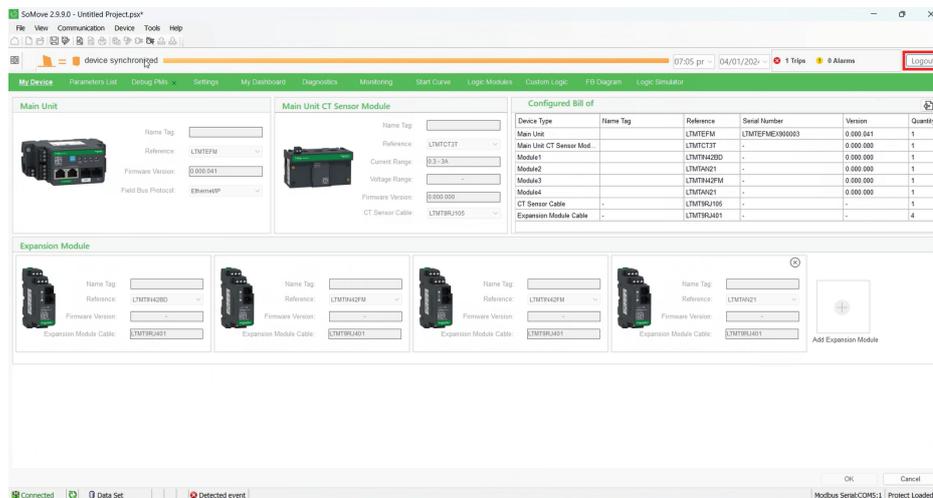
2. 在 **New Pin** 和 **Confirm Pin** 中输入新的六位数字 Pin 码。
3. 选择 **Create**。

结果： Pin 码被创建。



您可以使用 **Parameter List > Device Session Management** 选项卡中的 **Device Session Timeout** 参数来设置会话时间，每个会话最少允许 5 分钟，最多允许 60 分钟。

选择 **Logout**，断开设备与 SoMove 软件的连接。



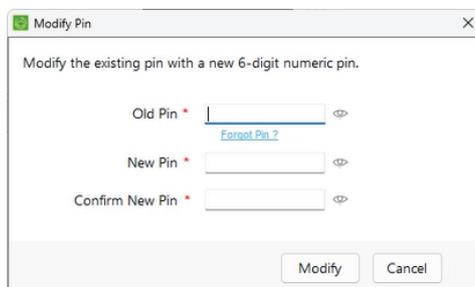
Modify Pin

Modify Pin 选项让您能够修改 Pin 码。只有在设置了引脚后，**Modify Pin** 选项才可用。

以下步骤显示了如何修改 Pin 码：

1. 导航至 **Device > Pin Manager > Modify Pin**

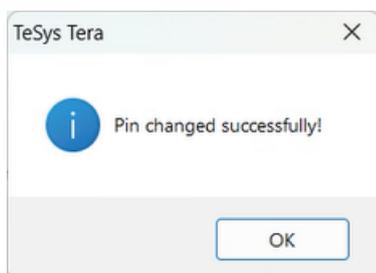
结果： 随即显示 **Modify Pin** 对话框。



2. 分别为 **Old Pin**、**New Pin** 和 **Confirm New Pin** 输入值。

3. 选择 **Modify**。

结果： Pin 码被修改。



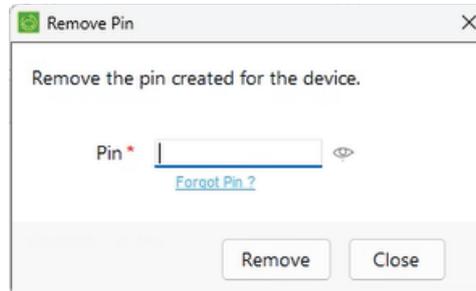
Delete Pin

Remove Pin 选项让您能够删除现有 Pin 码。只有在设置了引脚后，**Remove Pin** 选项才可用。

以下步骤显示了如何删除 Pin 码：

1. 导航至 **Device > Pin Manager > Remove Pin**

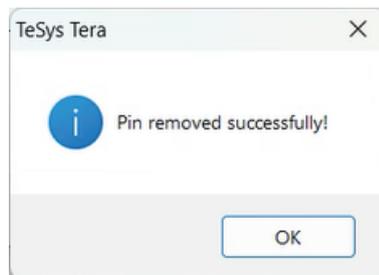
结果：随即显示 **Remove Pin** 对话框。



2. 在 **Pin** 字段中输入现有 Pin 码。

3. 选择 **Remove**。

结果：Pin 码被删除。



Forgot Pin

如果忘记了 Pin 码，则应使用设备上的 **Reset** 按钮执行出厂复位。



注：执行出厂复位时，TeSys Tera DTM 与设备断开连接。

Custom Logic Editor

此部分内容

Custom Logic Editor 介绍.....	93
使用 Custom Logic Editor.....	96
自定义逻辑程序的特性.....	98
自定义逻辑变量的定义.....	99
LTMT Main Unit 变量定义.....	100
CALL_EOM 命令描述.....	103

Custom Logic Editor 介绍

概述

您可以使用 Custom Logic Editor 来自定义设置 LTMT main unit 预定义控制程序。Custom Logic Editor 是一个强大的编程工具，仅适用于包含 TeSys Tera DTM Library 库的 SoMove。为 LTMT main unit 创建自定义控制程序包括：使用其中一种自定义逻辑编程语言来编写一系列指令（逻辑命令）。

Custom Logic Editor 的用途

Custom Logic Editor 的主要用途是，让您能够灵活地修改启动器类型的预定义逻辑或添加附加逻辑功能。

逻辑 ID

每种 LTMT main unit 运行模式（或电机启动器类型）都有一个预定义的控制程序。运行模式（或电机启动器类型）程序通过唯一的逻辑 ID 来标识。预定义的运行模式程序的逻辑 ID 是从 12 到 15 的数字。自定义预定义运行模式程序时，自定义程序的逻辑 ID 必须等于预定义程序的逻辑 ID + 256。

下表按运行模式列出了逻辑 ID：

运行模式/启动器类型	预定义程序的逻辑 ID	自定义程序的逻辑 ID
Reserved	0-11	-
Overload	12	268
Direct Online	13	269
Reverse Direct Online	14	270
Star Delta	15	271
Customized program	-	256-267、279-511

自定义程序

自定义程序是具有特定功能的 LTMT main unit 预定义程序，旨在满足具体应用需求。

在配置有其中一个预定义程序时，LTMT main unit 使用 LTMT main unit 和 PCode 中的固件来管理控制功能。

注：PCode（伪代码）是一组无错误的十六进制指令。

在配置有预定义程序时，LTMT main unit 保留由 LTMT main unit 控制的功能。这些功能包括以下属于父预定义程序的固有特性：

- 限制可写入到地址为 704 (0x02C0) 的寄存器（网络命令寄存器）的内容。
- 在演示模式下显示运行状态（例如：Forward 或 Reverse、Low Speed 或 High Speed）。
- 限制可通过菜单设置的故障预置模式。
- 星三角 (Star Delta) 启动方式中关于启动过程的特定行为特征。

- 限制可通过菜单设置的转换定时器。

预定义程序的结构

在 SoMove 软件上，TeSys Tera DTM Library 库有 11 个预定义程序，针对每种运行模式（或电机启动器类型）提供一个预定义程序。

预定义程序依次执行以下部分：

- 使用逻辑 ID 对程序进行逻辑标识
- 输入管理
- 运行模式执行
- 输出更新

运行模式的执行是嵌入式的，通过功能 `CALL_EOM` 进行调用。

这样，就能够对自定义程序的输入和输出管理进行自定义而无需修改运行模式执行情况。

Custom Logic Editor 编程语言和工具

Custom Logic Editor 提供两种编程语言和工具：

- 自定义逻辑语言 (Custom logic language)：是一种列表指令语言，可通过 Custom Logic Editor 编程工具进行编辑。
- FBD：是一种基于图形对象的编程语言，可通过 FBD 编辑器编程工具进行编辑。

每种编程方法都可实现编程目标。不过，使用 Custom Logic Editor，可以选择喜好的编程方法的形式。

Logic Commands

自定义逻辑语言和 FBD 语言可实现以下命令类型：

- 程序逻辑命令
- 布尔逻辑命令
- 寄存器逻辑命令
- 定时器逻辑命令
- 计数器逻辑命令
- 锁存逻辑命令
- 数学逻辑命令

Custom Logic Editor

下图显示了集成在 TeSys Tera DTM Library 中的 Custom Logic Editor：

使用 Custom Logic Editor

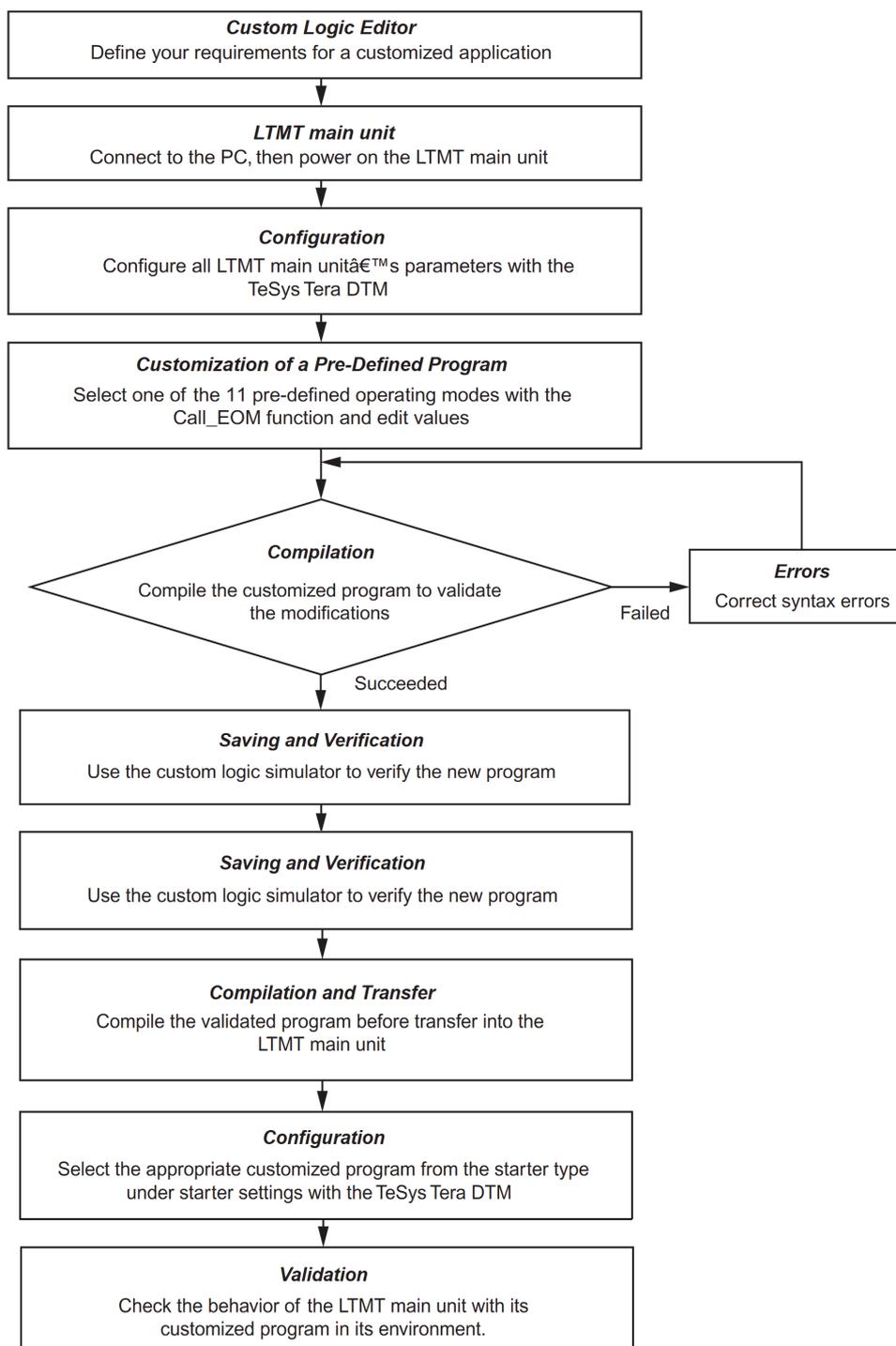
概述

Custom Logic Editor 让您能够创建自定义程序并验证其是否符合需求。这些操作完成后，LTMT main unit 固件会加载并运行您创建的指令。

任务流程图

下图显示了在创建和修改自定义程序期间要执行的所有任务。

注：指定的顺序仅是示例。您所用的顺序取决于自己的工作方法。



自定义逻辑中的自定义方法

如要自定义设置预定义程序文件，请执行以下步骤：

1. 定义与您的应用程序需求相匹配的运行模式。
2. 在 Custom Logic Editor 中打开预定义的运行模式程序文件 (*.rtf)。
3. 在自定义逻辑中编辑预定义程序，按照以下三种方法之一自定义该程序：
 - 预定义的运行模式可满足您的应用需求：但需要附加CALL_EOM功能。
 - 预定义运行模式基本符合您的应用需求，但需要附加功能：使用 CALL_EOM 功能，并在 CALL_EOM 指令后添加附加指令。
 - 预定义的运行模式不符合您的应用需求：需要从头开始创建新程序（不推荐）。
4. 如果需要，编辑自定义程序的输入。
5. 如果需要，编辑自定义程序的输出。
6. 根据 CALL_EOM 和控制模式，更新 Custom Logic Editor 介绍, 93 页。
7. 编译自定义程序。

FBD 中的自定义方法

1. 打开一个空白 FBD 程序页。
2. 创建自定义程序的输入管理。
3. 按照以下三种方法之一创建运行模式执行：
 - 以下操作模式之一符合您的应用需求：但仅使用该CALL_EOM功能。
 - 运行模式之一基本符合您的应用需求但需要附加功能：使用 CALL_EOM 功能，并在 CALL_EOM 指令后添加附加指令。
 - 所有运行模式都不符合您的应用需求：需要从头开始创建新程序（不推荐）。
4. 创建自定义程序的输出管理。
5. 根据 CALL_EOM 和控制模式，更新 Custom Logic Editor 介绍, 93 页。
6. 将 FBD 编译为自定义逻辑。

自定义逻辑程序的特性

简介

传入或传出 LTMT main unit 的数据为 16 位寄存器形式。这些寄存器按数字排序，并通过 16 位寄存器地址 (0-65,535) 进行引用。

自定义程序可以修改以下三类寄存器的值：

- LTMT 寄存器
- 临时寄存器
- 非易失性寄存器

逻辑内存特性

自定义程序的命令列表保存在 LTMT main unit 内部非易失性存储器的某个区域中。

此逻辑内存的格式如下表所示：

内存位置	项目	范围	描述
0	逻辑程序大小 (n)	0-8,191 0 表示未加载任何自定义程序。	16 位字
1	逻辑校验和	0-65,535	偏移 2-n+2 的程序内存之和
2	逻辑 ID	256-511，请参阅	LTMT main unit 中自定义程序的标识符
3	逻辑命令/参数 1	取决于逻辑命令	逻辑功能的一个字
4	逻辑命令/参数 2		
5	逻辑命令/参数 3		
-	-	-	-
n+2	逻辑命令/参数 n	-	逻辑功能的一个字

逻辑内存限制

程序大小取决于逻辑命令的数目。在文本编辑器中，一个命令及其参数将占一行，而在内存中，它占用的内存位置的数目与参数个数相同。

例如，命令 timer 0.1980 将使用四个内存位置。

自定义逻辑变量的定义

简介

使用 Custom Logic Editor，可以在指令 LTMT main unit 的控制程序中实现命令，以：

- 读取或写入到临时寄存器
- 读取或写入到非易失性寄存器
- 读取或写入到 LTMT main unit 寄存器，请参阅 LTMT Main Unit 变量定义, 100 页。

LTMT main unit 通过描述寄存器在自定义逻辑内存空间中的地址的整数来定义每个自定义逻辑寄存器。此整数的值从地址 0 开始，最大地址等于 LTMT main unit 中寄存器可用的内存位置数 - 1。

临时寄存器

LTMT main unit 在临时存储器中提供可通过逻辑命令访问的寄存器。由于这些寄存器位于临时或易失性存储器中，因此对控制器执行电源重置时，它们不会保留值设置。

有 300 个临时寄存器可用，地址范围为 0 至 299。

可用的临时寄存器数量可在地址为 1204 (0x04B4) 的 LTMT main unit 寄存器 (自定义逻辑临时空间) 中读取。

非易失性寄存器

LTMT main unit 在非易失性存储器中提供由逻辑命令使用的寄存器。由于这些寄存器位于非易失性存储器中，因此对控制器执行电源重置时，它们会保留值设置。

有 300 个非易失性寄存器可用，地址范围为 0 至 63。

可用的非易失性寄存器数量可在地址为 1205 (0x04B5) 的 LTMT main unit 寄存器 (自定义逻辑非易失性空间) 中读取。

LTMT Main Unit 变量定义

LTMT Main Unit 寄存器

LTMT main unit 内存包含 9250 个寄存器，这些寄存器的地址范围为 0 至 9249 (0x0000 至 0x2421)。

每个寄存器都是一个 16 位字且为：

- 只读状态，无法编辑其值。
- 读写状态，可以编辑其值。

寄存器编号等于寄存器地址 + 1。

Custom Logic Editor 仅使用寄存器地址。

自定义逻辑命令可用于更改 LTMT main unit 的读写寄存器的值。

访问寄存器

使用 Custom Logic Editor，可以访问 TeSys Tera 通讯指南, 7 页中定义的所有 LTMT main unit 寄存器。

自定义逻辑寄存器

地址为 1200 至 1205 (0x04B0 至 0x04B5) 的寄存器以及地址为 1291 (0x050B) 的寄存器被 TeSys Tera DTM Library 用来访问 LTMT main unit 寄存器。

这些寄存器还是可从通讯端口访问的自定义逻辑寄存器。这些只读寄存器将在以下章节中介绍。

下表列出了这些寄存器：

寄存器地址	定义	范围 (值)
1201 (0x04B1)	自定义逻辑版本	0-65,535
1202 (0x04B2)	自定义逻辑内存空间	
1203 (0x04B3)	已用自定义逻辑内存	
1204 (0x04B4)	自定义逻辑临时空间	
1205 (0x04B5)	自定义逻辑非易失性空间	
1291 (0x050B)	自定义逻辑 DO 输入信息	

地址为 1201 (0x04B1) 的寄存器

地址为 1201 (0x04B1) 的寄存器指示自定义逻辑功能版本。版本号指明了 LTMT main unit 支持的特定逻辑命令组。

地址为 1202 (0x04B2) 的寄存器

地址为 1202 (0x04B2) 的寄存器定义可用的逻辑内存空间。可用于保存逻辑命令的非易失性 LTMT main unit 逻辑内存字 (16 位) 的数量。

地址为 1203 (0x04B3) 的寄存器

地址为 1203 (0x04B3) 的寄存器定义已使用的逻辑内存。这是当前存储在 LTMT main unit 中的逻辑命令所使用的非易失性 LTMT main unit 逻辑内存字 (16 位) 的数量。

地址为 1204 (0x04B4) 的寄存器

地址为 1204 (0x04B4) 的寄存器定义 LTMT main unit 提供的临时寄存器的数量。

地址为 1205 (0x04B5) 的寄存器

地址为 1205 (0x04B5) 的寄存器定义 LTMT main unit 提供的非易失性寄存器的数量。

地址为 1291 (0x050B) 的寄存器

地址为 1291 (0x050B) 的寄存器是自定义逻辑 DO 输入信息寄存器。自定义程序通过它来配置 I/O 分配。

下表描述了此寄存器中的每个位：

寄存器地址	位编号	说明
1291 (0x050B)	0	自定义逻辑数字量输出 1 (DO1) 输入信息
	1	自定义逻辑数字量输出 2 (DO2) 输入信息
	2	自定义逻辑数字量输出 3 (DO3) 输入信息
	3	自定义逻辑数字量输出 4 (DO4) 输入信息
	4	自定义逻辑数字量输出 5 (DO5) 输入信息
	5	自定义逻辑数字量输出 6 (DO6) 输入信息
	6	自定义逻辑数字量输出 7 (DO7) 输入信息
	7	自定义逻辑数字量输出 8 (DO8) 输入信息
	8	自定义逻辑数字量输出 9 (DO9) 输入信息
	9	自定义逻辑数字量输出 10 (DO10) 输入信息
	10	自定义逻辑数字量输出 11 (DO11) 输入信息
	11	自定义逻辑数字量输出 12 (DO12) 输入信息
	12	自定义逻辑数字量输出 13 (DO13) 输入信息
	13	保留
14	保留	

寄存器地址	位编号	说明
	15	保留

地址为 1301 至 1399 (0x0515 至 0x0577) 的寄存器

地址为 1301 至 1399 (0x0515 至 0x0577) 的寄存器是逻辑函数的通用寄存器。它们用于在外部源 (如 PLC) 与自定义逻辑应用程序之间交换信息。

这些易失性寄存器可读写，且可通过自定义逻辑功能或通信端口进行修改。

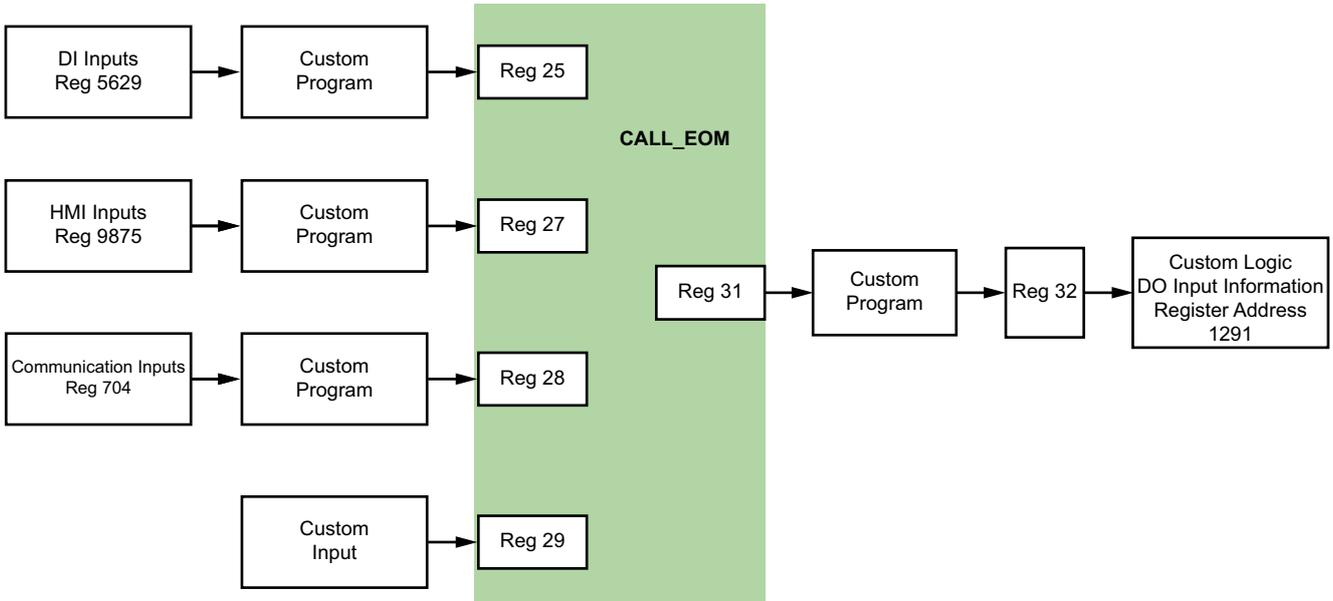
CALL_EOM 命令描述

概述

CALL_EOM 函数允许执行某个运行模式。

为此，该函数使用地址为 0 至 61 的临时寄存器。

为了围绕 CALL_EOM 函数构建自定义计划，必须了解应用程序和 LTMT main unit 的各个寄存器的使用方式：



- 地址为 25 至 29 的临时寄存器是 CALL_EOM 函数的输入寄存器。自定义时，必须按位对它们赋值。
- 地址为 31 的临时寄存器是 CALL_EOM 函数的输出寄存器。其值在执行运行模式后给出。
- 地址为 32 的临时寄存器是一个临时寄存器，用于一次性地设置地址为 1291 (0x050B) 的自定义逻辑 DO 输入信息寄存器。必须使用地址为 32 的临时寄存器来自定义 CALL_EOM 输出。

数字量输入寄存器

寄存器地址	寄存器描述	位	位描述
5629 (0x15FD)	数字量输入寄存器	0	Local-START> DI
		1	Local-STOP DI
		2	Local-START>> DI
		3	模式选择 1
		4	Local-START< DI
		5	Local-START<< DI
		6	Remote-START> DI
		7	Remote-STOP DI
		8	Remote-START>> DI
		9	模式选择 2
		10	Remote-START< DI
		11	Remote-START<< DI
		12	运行 DI
		13	速度变化 DI
		14	保留
15	保留		

HMI 输入寄存器

寄存器地址	寄存器描述	位	位描述
9875 (0x2693)	HMI 命令寄存器 1	0	“电机正向运行”/“高速正向”命令
		1	“电机反向运行”/“高速反向”命令
		2	“本地/远程”模式选择 1
		3	“脱扣复位”命令
		4	“本地/远程”模式选择 2
		5	“自检（无脱扣）”命令
		6	“电机低速正向”命令
		7	“电机低速反向”命令
		8	“复位禁止”命令
		9	“复位启动次数”命令
		10	“复位停止次数”命令
		11	“清除电能”命令
		12	保留
		13	“逻辑测试”命令
		14	“复位总运行时长”命令
15	“自检（带脱扣）”命令		
9876 (0x2694)	HMI 命令寄存器 2	0	保留
		1	保留
		2	“清除热容量水平”命令

		3	保留
		4	保留
		5	“清除脱扣计数器”命令
		6	“出厂复位”命令
		7	“软启动器复位”命令
		8-12	保留
		13	“存储参考起始曲线”命令
		14	“清除脱扣记录”命令
		15	“清除事件记录”命令

通信输入寄存器

寄存器地址	寄存器描述	位	位描述
704 (0x02C0)	通信命令寄存器 1	0	“电机正向运行”/“高速正向”命令
		1	“电机反向运行”/“高速反向”命令
		2	“本地/远程”模式选择 1
		3	“脱扣复位”命令
		4	“本地/远程”模式选择 2
		5	“自检（无脱扣）”命令
		6	“电机低速正向”命令
		7	“电机低速反向”命令
		8	“复位禁止”命令
		9	“复位启动次数”命令
		10	“复位停止次数”命令
		11	“清除电能”命令
		12	“电机停止”命令
		13	“逻辑测试”命令
		14	“复位总运行时长”命令
		15	“自检（带脱扣）”命令
705 (0x02C1)	通信命令寄存器 2	0	保留
		1	保留
		2	“清除热容量水平”命令
		3	保留
		4	保留
		5	“清除脱扣计数器”命令
		6	“出厂复位”命令
		7	“软启动器复位”命令
		8-12	保留
		13	“存储参考起始曲线”命令
		14	“清除脱扣记录”命令
		15	“清除事件记录”命令

CALL_EOM 12 描述

当 CALL_EOM 参数等于 12 时，该函数将执行过载运行模式。

在自定义程序中使用的逻辑 ID 是：

- LOGID_ID 268，用于过载运行模式

临时寄存器的用途如下：

寄存器	寄存器描述	位	位描述
输入寄存器 25	DI 输入的副本	0-15	未使用
输入寄存器 27	输入内容的副本 HMI	0-15	未使用
输入寄存器 28	通信输入副本	0-15	未使用
输出寄存器 31	要分配到 LTMT main unit 输出的 CALL_EOM 指令的输出。	0	自定义逻辑“Run1 Cde”information
		1-15	未使用
输出寄存器 32	要分配到 LTMT main unit 输出的 CALL_EOM 指令的输出。	0	自定义逻辑 DO1 Input information
		1-15	未使用

CALL_EOM 13 描述

当 CALL_EOM 参数等于 13 时，该函数将执行“Direct Online”运行模式。

在自定义程序中使用的逻辑 ID 是：

- LOGID_ID 269，用于“Direct Online”运行模式

临时寄存器的用途如下：

寄存器	寄存器描述	位	位描述
输入寄存器 25	DI 输入的副本	0	Local-START> DI
		1	Local-STOP DI
		2	未使用
		3	DI 模式选择 1
		4-5	未使用
		6	Remote-START> DI
		7	Remote-STOP DI
		8	未使用
		9	DI 模式选择 2
		10-11	未使用
		12	运行 DI
		13-15	未使用
		输入寄存器 27	输入内容的副本 HMI
1	未使用		
2	HMI_STOP		
3-4	未使用		
5	HMI 模式选择 1		
6	HMI 模式选择 2		
7-15	未使用		
输入寄存器 28	通信输入副本	0	COMM 启动 >
		1-3	未使用
		4	远程模式选择 1
		5	远程模式选择 2
		6	COMM 停止
		7-15	未使用
输出寄存器 31	要分配到 LTMT main unit 输出的 CALL_EOM 指令的输出。	0	自定义逻辑“Run1 Cde” information
		1-15	未使用
输出寄存器 32	要分配到 LTMT main unit 输出的 CALL_EOM 指令的输出。	0	自定义逻辑 DO1 Input information
		1-15	未使用

CALL_EOM 14 描述

当 CALL_EOM 参数等于 14 时，该函数将执行“Reverse Direct Online”运行模式。

在自定义程序中使用的逻辑 ID 是：

- LOGID_ID 270，用于“Reverse Direct Online”运行模式

临时寄存器的用途如下：

寄存器	寄存器描述	位	位描述
输入寄存器 25	DI 输入的副本	0	Local-START> DI
		1	Local-STOP DI
		2	未使用
		3	DI 模式选择 1
		4	Local-START< DI
		5	未使用
		6	Remote-START> DI
		7	Remote-STOP DI
		8	未使用
		9	DI 模式选择 2
		10 秒	Remote-START< DI
		11	未使用
		12	运行 DI
		13-15	未使用
输入寄存器 27	输入内容的副本 HMI	0	HMI_START >
		1	未使用
		2	HMI_STOP
		3	HMI_START <
		4	未使用
		5	HMI 模式选择 1
		6	HMI 模式选择 2
		7-15	未使用
输入寄存器 28	通信输入副本	0	COMM 启动 >
		1	未使用
		2	COMM 启动 <
		3	未使用
		4	远程模式选择 1
		5	远程模式选择 2
		6	COMM 停止
		7-15	未使用
输出寄存器 31	要分配到 LTMT main unit 输出的 CALL_EOM 指令的输出。	0	自定义逻辑“Run1 Cde” information
		1	自定义逻辑“Run2 Cde” information
		2-15	未使用
输出寄存器 32	要分配到 LTMT main unit 输出的 CALL_EOM 指令的输出。	0	自定义逻辑 DO1 Input information
		1	自定义逻辑 DO2 Input information
		2-15	未使用

CALL_EOM 15 描述

当 CALL_EOM 参数等于 15 时，该函数将执行“星三角”运行模式。

在自定义程序中使用的逻辑 ID 是：

- LOGID_ID 271，用于“星三角”运行模式

临时寄存器的用途如下：

寄存器	寄存器描述	位	位描述
输入寄存器 25	DI 输入的副本	0	Local-START> DI
		1	Local-STOP DI
		2	未使用
		3	模式选择 1
		4-5	未使用
		6	Remote-START> DI
		7	Remote-STOP DI
		8	未使用
		9	模式选择 2
		10-11	未使用
		12	运行 DI
		13-15	未使用
输入寄存器 27	输入内容的副本 HMI	0	HMI_START >
		1	未使用
		2	HMI_STOP
		3-4	未使用
		5	模式选择 1
		6	模式选择 2
		7-15	未使用
输入寄存器 28	通信输入副本	0	COMM 启动 >
		1-3	未使用
		4	远程模式选择 1
		5	远程模式选择 2
		6	COMM 停止
		7-15	未使用
输出寄存器 31	要分配到 LTMT main unit 输出的 CALL_EOM 指令的输出。	0	“自定义逻辑 Run1”命令信息
		1	“自定义逻辑 Run2”命令信息
		2	“自定义逻辑 Run3”命令信息
		3-15	未使用
输出寄存器 32	要分配到 LTMT main unit 输出的 CALL_EOM 指令的输出。	0	自定义逻辑 DO1 Input information
		1	自定义逻辑 DO2 Input information
		2	自定义逻辑 DO3 Input information
		3-15	未使用

程序示例

```

LOGIC_ID 13
//
// DI Input
LOAD_BIT      5629 0
SET_TMP_BIT   25 0 //Local-START> DI
LOAD_BIT      5629 1
SET_TMP_BIT   25 1 //Local-STOP DI
LOAD_BIT      5629 2
SET_TMP_BIT   25 2 //Local-START>> DI
LOAD_BIT      5629 3
SET_TMP_BIT   25 3 //DI Mode Selection 1
LOAD_BIT      5629 4
SET_TMP_BIT   25 4 //Local-START< DI
LOAD_BIT      5629 5
SET_TMP_BIT   25 5 //Local-START<< DI
LOAD_BIT      5629 6
SET_TMP_BIT   25 6 //Remote-START> DI
LOAD_BIT      5629 7
SET_TMP_BIT   25 7 //Remote-STOP DI
LOAD_BIT      5629 8
SET_TMP_BIT   25 8 //Remote-START>> DI
LOAD_BIT      5629 9
SET_TMP_BIT   25 9 //DI Mode Selection 2
LOAD_BIT      5629 10
SET_TMP_BIT   25 10 //Remote-START< DI
LOAD_BIT      5629 11
SET_TMP_BIT   25 11 //Remote-START<< DI
LOAD_BIT      5629 12
SET_TMP_BIT   25 12 //Run DI
LOAD_BIT      5629 13
SET_TMP_BIT   25 13 //Speed Change
LOAD_BIT      5629 14
SET_TMP_BIT   25 14 //Not Used
LOAD_BIT      5629 15
SET_TMP_BIT   25 15 //Not Used
//
// HMI Input
LOAD_BIT      9875 0
SET_TMP_BIT   27 0 //HMI_START >
LOAD_BIT      9875 6
SET_TMP_BIT   27 1 //HMI_START >>
LOAD_BIT      9875 12

```

```

SET_TMP_BIT 27 2 //HMI_STOP,
LOAD_BIT 9875 1
SET_TMP_BIT 27 3 //HMI_START <
LOAD_BIT 9875 7
SET_TMP_BIT 27 4 //HMI_START <<
LOAD_BIT 9875 2
SET_TMP_BIT 27 5 //HMI Mode Selection 1
LOAD_BIT 9875 4
SET_TMP_BIT 27 6 //HMI Mode Selection 2
//
//
// Remote Input
LOAD_BIT 704 0
SET_TMP_BIT 28 0 //COMM Start >
LOAD_BIT 704 6
SET_TMP_BIT 28 1 //COMM Start >>
LOAD_BIT 704 1
SET_TMP_BIT 28 2 //COMM Start <
LOAD_BIT 704 7
SET_TMP_BIT 28 3 //COMM Start <<
LOAD_BIT 704 2
SET_TMP_BIT 28 4 //Remote Mode Selection 1
LOAD_BIT 704 4
SET_TMP_BIT 28 5 //Remote Mode Selection 2
//
//End customer Zone
//Call Command
//Output
//=====
// Customer Zone: Custom application
// Add specific code for Custom Logic function here"
//
CALL_EOM 13
//=====
// Customer Zone: Outputs management"
//
// Output
LOAD_TMP_BIT 31 0 //CL "Run1 Cde" information
SET_TMP_BIT 32 0 //CL D01 Input information
LOAD_TMP_BIT 31 1 //CL "Run2 Cde" information
SET_TMP_BIT 32 1 //CL D02 Input information
LOAD_TMP_BIT 31 2 //CL "Run3 Cde" information

SET_TMP_BIT 32 2 //CL D03 Input information
LOAD_TMP_BIT 31 3 //CL "Run4 Cde" information
SET_TMP_BIT 32 3 //CL D04 Input information
LOAD_TMP_BIT 31 4 //CL "Run5 Cde" information
SET_TMP_BIT 32 4 //CL D05 Input information
//
//
//
// End Customer Zone
//=====
// Schneider Zone (Do not modify)
//
// CL output in 1291
LOAD_K_BIT 1
SET_NOT_TMP_BIT 0 3
LOAD_TMP_REG 32 //Get image of 1291
ON_SET_REG 1291 0 //Put it into 1291
//

```

自定义逻辑语言

此部分内容

创建自定义逻辑程序	114
逻辑命令	124
自定义逻辑程序示例	152

创建自定义逻辑程序

此章节内容

Custom Logic Editor 简介	115
Custom Logic Editor 用户界面	116
逻辑命令	119

Custom Logic Editor 简介

概述

Custom Logic Editor 是带 TeSys Tera DTM Library 的 SoMove 的组件。使用 Custom Logic Editor，可查看现有的程序文件，或者可使用基于指令的文本语言（而不是基于图形的编程语言）来创建程序文件。

编辑自定义逻辑程序

创建自定义程序的最简单的方法是，从其中一种运行模式（或电机启动器类型）的预定义程序开始。Custom Logic Editor 的安装附带有 11 个预定义程序文件，每种电机启动器类型使用一个这样的文件。

每个程序文件都带有描述性名称（例如：DIRECT_ONLINE）和文件扩展名 *.rtf*。

Custom Logic Editor 用户界面

如要打开 Custom Logic Editor，请选择 **Device > Custom Logic > New Custom Logic**。

无论 TeSys Tera DTM Library 是否处于连接模式，Custom Logic Editor 都可用。不过，只能在连接模式下在 TeSys Tera DTM Library 和设备之间传送程序。

Custom Logic Editor 用户界面

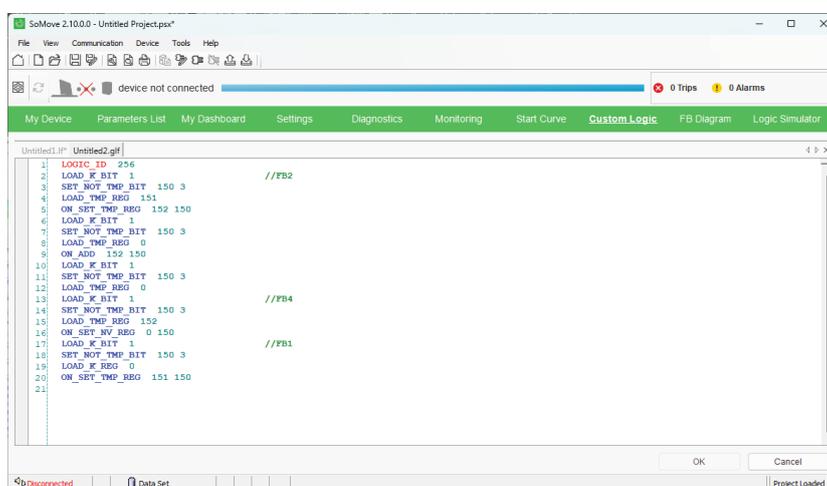
简介

用这种语言编写的程序包含由 LTMT main unit 按顺序执行的一系列指令。每个列表指令由单个程序行表示，并由四个部分组成：

- 行号
- 逻辑命令（记忆码）
- 参数
- 注释

自定义逻辑程序示例

以下窗口是使用 Custom Logic Editor 创建的程序的示例。



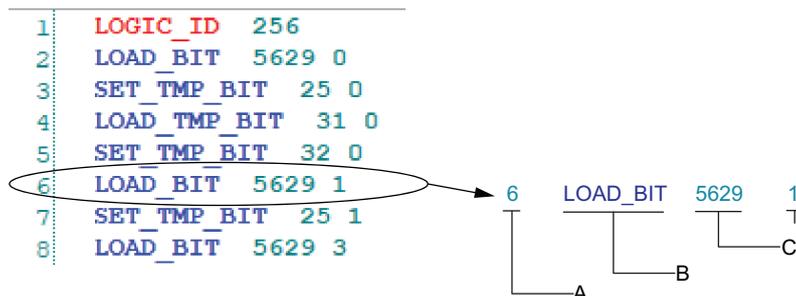
编辑多个程序

您可以同时创建或修改多个自定义程序。单击文件名以在它们之间切换。

例如，在上方的文本视图中，根据要编辑的程序，选择 **DIRECT_ONLINE.rtf** 或 **Untitled.lf**。

指令元素

下图是自定义逻辑程序的示例：



A	行号
B	逻辑命令
C	参数

行号

行号是附加信息：

- 它只能由编辑器定义。
- 它在自定义逻辑功能自身中不具有任何重要性。

逻辑命令

逻辑命令是一条指令，用于使用一个或多个参数标识要执行的操作。在上例中，LOAD_BIT 命令可将参数的值加载到称为 1 位累加器的内部寄存器中。

有两种类型的命令：

- 设置命令
这些命令对必需条件进行设置或测试以执行一项操作（如 LOAD 和 AND 命令）。
- 操作命令
这些命令指示 LTMT main unit 根据设置指令中的信息执行操作（例如，分配命令，如 COMP）。

注：以大写或小写键入一条逻辑命令后，系统将自动识别并以蓝色显示它。

参数

参数是一个数字，表示一个值（寄存器地址、位编号或常量），LTMT main unit 可以在指令中进行操作这个值。逻辑命令可具有 0 至 3 个参数，具体取决于逻辑命令的类型。

例如，在示例程序中，指令 LOAD_BIT 5629 1 包括逻辑命令 LOAD_BIT 和 2 个参数（5629 和 1）。

它指示 LTMT main unit 将地址为 5629 (0x15FD) 的寄存器的位 1 的值加载到累加器中。

通过将指令与命令和参数一起使用，自定义程序可以：

- 读取数字量输入的状态。
- 读取或写入数字量输出的状态。
- 激活定时器和计数器等基本逻辑功能。
- 执行算术运算、逻辑运算、比较运算和数字运算。
- 读取或写入 LTMT main unit 寄存器或这些寄存器中的各个位。

注：输入一个参数时，系统将自动识别它并用分配给参数的颜色显示它。

注释

在 Custom Logic Editor 中，可以在以下位置为程序添加注释：

- 在参数后面的每行末尾处
- 整行中

注:

- 当您输入 // 时，Custom Logic Editor 自动将其后的文本识别为注释，并用绿色显示。
- 无法从 LTMT main unit 获取注释。

语法

在 Custom Logic Editor 中，可以通过以下方式写入指令：

- 在参数之间使用空格、逗号或点
- 使用大写或小写字符

语法检查

输入过程中，文本编辑器将检查指令语法：

- 纠正指令以醒目的蓝色字符显示。
- 错误指令仍显示为黑色，必须在编译前进行纠正。

键盘命令

键盘命令和快捷键与 Windows 操作系统相同（例如：按 **DEL** 或 **DELETE** 可删除字符或行，按 **ENTER** 可转到下一行）。

保存

如要保存编辑或创建的程序，请选择 **Device > Custom Logic**，然后选择 **Save Custom Logic** 或 **SaveAs Custom Logic**。

注: 此文件使用扩展名 *.lf* 进行保存。

逻辑命令

概述

LTMT main unit 项目文件由一系列逻辑命令组成。每个逻辑命令都包含命令本身及最多 3 个参数。

每个逻辑命令所执行的操作都链接到 1 位布尔累加器 (值 0 或 1) 或 16 位无符号累加器 (值范围 0-65,535)。

Custom Logic Editor 提供以下种类的逻辑命令：

- 布尔型
- 寄存器
- 定时器
- 锁存
- 计数器
- 数学

上升沿检测机制

某些逻辑命令在 1 位累加器的上升沿执行。

在当前状态为 1 且其先前状态为 0 时，检测到位的上升沿。该位的先前状态存储在专用历史位中。

注: 如果该历史位发生修改，则可能会扰乱上升沿的检测。

布尔逻辑命令

布尔命令对简单的布尔 (On 或 Off) 值执行求值和控制。布尔命令包括：

命令	参数 1	参数 2	参数 3	描述
LOAD_K_BIT	常量值 (0 或 1)	-	-	将常量值加载到 1 位累加器中。
LOAD_BIT	寄存器地址	寄存器位编号 (0-15)	-	将寄存器位从标识的地址加载到参数 1 中，将参数 2 中标识的位加载到 1 位累加器中。
LOAD_TMP_BIT	临时寄存器地址	寄存器位编号 (0-15)	-	将临时寄存器位加载到 1 位累加器中。
LOAD_NV_BIT	非易失性寄存器地址	寄存器位编号 (0-15)	-	将非易失性寄存器位加载到 1 位累加器中。
LOAD_NOT_BIT	寄存器地址	寄存器位编号 (0-15)	-	将寄存器位的取反布尔值加载到 1 位累加器中。
LOAD_NOT_TMP_BIT	临时寄存器地址	寄存器位编号 (0-15)	-	将临时寄存器位的取反布尔值加载到 1 位累加器中。
LOAD_NOT_NV_BIT	非易失性寄存器地址	寄存器位编号 (0-15)	-	将非易失性寄存器位的取反布尔值加载到 1 位累加器中。
AND_BIT	寄存器地址	寄存器位编号 (0-15)	-	加载寄存器位值和 1 位累加器内容之间的逻辑 AND 链接的结果。结果保存在 1 位累加器中。
AND_TMP_BIT	临时寄存器地址	寄存器位编号 (0-15)	-	加载临时寄存器位值和 1 位累加器内容之间的逻辑 AND 链接的结果。结果保存在 1 位累加器中。
AND_NV_BIT	非易失性寄存器地址	寄存器位编号 (0-15)	-	加载非易失性寄存器位值和 1 位累加器内容之间的逻辑 AND 链接的结果。结果保存在 1 位累加器中。

命令	参数 1	参数 2	参数 3	描述
AND_NOT_BIT	寄存器地址	寄存器位编号 (0-15)	-	加载取反的寄存器位和 1 位累加器之间的逻辑 AND 链接的结果。结果保存在 1 位累加器中。
AND_NOT_TMP_BIT	临时寄存器地址	寄存器位编号 (0-15)	-	加载取反的临时寄存器位和 1 位累加器之间的逻辑 AND 链接的结果。结果保存在 1 位累加器中。
AND_NOT_NV_BIT	非易失性寄存器地址	寄存器位编号 (0-15)	-	加载取反的非易失性寄存器位和 1 位累加器之间的逻辑 AND 链接的结果。结果保存在 1 位累加器中。
OR_BIT	寄存器地址	寄存器位编号 (0-15)	-	在寄存器位值与 1 位累加器内容之间建立逻辑 OR 链接。结果保存在 1 位累加器中。
OR_TMP_BIT	临时寄存器地址	寄存器位编号 (0-15)	-	在临时寄存器位值与 1 位累加器内容之间建立逻辑 OR 链接。结果保存在 1 位累加器中。
OR_NV_BIT	非易失性寄存器地址	寄存器位编号 (0-15)	-	在非易失性寄存器位值与 1 位累加器内容之间建立逻辑 OR 链接。结果保存在 1 位累加器中。
OR_NOT_BIT	寄存器地址	寄存器位编号 (0-15)	-	计算取反的寄存器位和 1 位累加器之间的逻辑 OR 的值。结果保存在 1 位累加器中。
OR_NOT_TMP_BIT	临时寄存器地址	寄存器位编号 (0-15)	-	计算取反的临时寄存器位和 1 位累加器之间的逻辑 OR 的值。结果保存在 1 位累加器中。
OR_NOT_NV_BIT	非易失性寄存器地址	寄存器位编号 (0-15)	-	计算取反的非易失性寄存器位和 1 位累加器之间的逻辑 OR 的值。结果保存在 1 位累加器中。
SET_BIT	寄存器地址	寄存器位编号 (0-15)	-	将 1 位累加器的值设置到寄存器位中。
SET_TMP_BIT	临时寄存器地址	寄存器位编号 (0-15)	-	将 1 位累加器的值设置到临时寄存器位中。
SET_NV_BIT	非易失性寄存器地址	寄存器位编号 (0-15)	-	将 1 位累加器的值设置到非易失性寄存器位中。
SET_NOT_BIT	寄存器地址	寄存器位编号 (0-15)	-	将 1 位累加器的取反值设置到寄存器位中。
SET_NOT_TMP_BIT	临时寄存器地址	寄存器位编号 (0-15)	-	将 1 位累加器的取反值设置到临时寄存器位中。
SET_NOT_NV_BIT	非易失性寄存器地址	寄存器位编号 (0-15)	-	将 1 位累加器的取反值设置到非易失性寄存器位中。
- 参数不适用于逻辑命令。				

寄存器逻辑命令

寄存器命令用于计算和控制 16 位值。寄存器命令包括：

命令	参数 1	参数 2	参数 3	描述
LOAD_K_REG	常量值 (0-65,535)	-	-	将常量值加载到 16 位累加器中。
LOAD_REG	寄存器地址	-	-	将寄存器的副本加载到 16 位累加器中。
LOAD_TMP_REG	临时寄存器地址	-	-	将临时寄存器的副本加载到 16 位累加器中。
LOAD_NV_REG	非易失性寄存器地址	-	-	将非易失性寄存器的副本加载到 16 位累加器中。
COMP_K_REG	常量值 (0-65,535)	临时寄存器地址	-	将参数 1 的内容与 16 位累加器的内容相比较，按以下所述设置状态参数 2 位： BIT 1 ON (如果 16 位累加器 < 参数 1 的内容) BIT 2 ON (如果 16 位累加器 = 参数 1 的内容) BIT 3 ON (如果 16 位累加器 > 参数 1 的内容)

命令	参数 1	参数 2	参数 3	描述
COMP_REG	寄存器地址	临时寄存器地址	-	将参数 1 定义的寄存器的内容与 16 位累加器的内容相比较，按以下所述设置状态参数 2 位： BIT 1 ON (如果 16 位累加器 < 参数 1 定义的寄存器内容) BIT 2 ON (如果 16 位累加器 = 参数 1 定义的寄存器内容) BIT 3 ON (如果 16 位累加器 > 参数 1 定义的寄存器内容)
COMP_TMP_REG	临时寄存器地址	临时寄存器地址	-	将参数 1 定义的寄存器的内容与 16 位累加器的内容相比较，按以下所述设置状态参数 2 位： BIT 1 ON (如果 16 位累加器 < 参数 1 定义的寄存器内容) BIT 2 ON (如果 16 位累加器 = 参数 1 定义的寄存器内容) BIT 3 ON (如果 16 位累加器 > 参数 1 定义的寄存器内容)
COMP_NV_REG	非易失性寄存器地址	临时寄存器地址	-	将参数 1 定义的寄存器的内容与 16 位累加器的内容相比较，按以下所述设置状态参数 2 位： BIT 1 ON (如果 16 位累加器 < 参数 1 定义的寄存器内容) BIT 2 ON (如果 16 位累加器 = 参数 1 定义的寄存器内容) BIT 3 ON (如果 16 位累加器 > 参数 1 定义的寄存器内容)
AND_K	常量值 (0 或 1)	-	-	在常量值与 16 位累加器内容之间建立逻辑 AND 链接。结果保存在 16 位累加器中。
AND_REG	寄存器地址	-	-	在寄存器位值与 16 位累加器内容之间建立逻辑 AND 链接。结果保存在 16 位累加器中。
AND_TMP_REG	临时寄存器地址	-	-	在临时寄存器值与 16 位累加器内容之间建立逻辑 AND 链接。结果保存在 16 位累加器中。
AND_NV_REG	非易失性寄存器地址	-	-	在非易失性寄存器值与 16 位累加器内容之间建立逻辑 AND 链接。结果保存在 16 位累加器中。
OR_K	常量值 (0 或 1)	-	-	在常量值与 16 位累加器内容之间建立逻辑 OR 链接。结果保存在 16 位累加器中。
OR_REG	寄存器地址	-	-	在寄存器位值与 16 位累加器内容之间建立逻辑 OR 链接。结果保存在 16 位累加器中。
OR_TMP_REG	临时寄存器地址	-	-	在临时寄存器值与 16 位累加器内容之间建立逻辑 OR 链接。结果保存在 16 位累加器中。
OR_NV_REG	非易失性寄存器地址	-	-	在非易失性寄存器值与 16 位累加器内容之间建立逻辑 OR 链接。结果保存在 16 位累加器中。
XOR_K	常量值 (0-65,535)	-	-	在常量值与 16 位累加器内容之间建立逻辑异 OR 链接。结果保存在 16 位累加器中。
XOR_REG	寄存器地址	-	-	在寄存器值与 16 位累加器内容之间建立逻辑异 OR 链接。结果保存在 16 位累加器中。
XOR_TMP_REG	临时寄存器地址	-	-	在临时寄存器值与 16 位累加器内容之间建立逻辑异 OR 链接。结果保存在 16 位累加器中。
XOR_NV_REG	非易失性寄存器地址	-	-	在非易失性寄存器值与 16 位累加器内容之间建立逻辑异 OR 链接。结果保存在 16 位累加器中。
ON_SET_REG	寄存器地址	临时寄存器地址	-	在 1 位累加器的上升沿将 16 位累加器内容存储在参数 1 定义的寄存器中。
ON_SET_TMP_REG	临时寄存器地址	临时寄存器地址	-	在 1 位累加器的上升沿将 16 位累加器内容存储在参数 1 定义的临时寄存器中。

命令	参数 1	参数 2	参数 3	描述
ON_SET_NV_REG	非易失性寄存器地址	临时寄存器地址	-	在 1 位累加器的上升沿将 16 位累加器内容存储在参数 1 定义的非易失性寄存器中。
- 参数不适用于逻辑命令。				

定时器逻辑命令

定时器的范围为 0 至 65,535，以秒或十分之几秒为间隔来计量时间：

- 参数 1 指定时间周期
- 参数 2 为计算出的结束时间
- 参数 3 为定时器状态寄存器

定时器命令包括：

命令	参数 1	参数 2	参数 3	描述
TIMER_SEC	临时寄存器（时段）	临时寄存器（计算出的结束时间）	临时寄存器（状态）	按状态寄存器位的描述以秒为单位对参数 1 中的时段输入计数。
TIMER_TENTHS	临时寄存器（时段）	临时寄存器（计算出的结束时间）	临时寄存器（状态）	按状态寄存器位的描述以十分之一秒为单位对参数 1 中的时段输入计数。

锁存逻辑命令

锁存命令包括：

命令	参数 1	参数 2	参数 3	描述
LATCH	临时寄存器 (状态)	-	-	在临时寄存器中记录并保留信号历史。
LATCH_NV	非易失性寄存器 (状态)	-	-	在非易失性寄存器中记录并保留信号历史。
- 参数不适用于逻辑命令。				

计数器逻辑命令

计数器的范围为 0 至 65,535，在计数达到最大值 65,535 时转换为 0。

计数器命令包括：

命令	参数 1	参数 2	参数 3	描述
COUNTER	临时寄存器 (计数器值)	常量值 0-65,535 (预设值)	临时寄存器 (状态)	执行比较计数，将计数和状态都保存到临时寄存器。
COUNTER_NV	非易失性寄存器 (计数器值)	常量值 0-65,535 (预设值)	非易失性寄存器 (状态)	执行比较计数，将计数和状态都保存到非易失性寄存器。

数学逻辑命令

算术命令使用 16 位累加器和临时寄存器执行无符号算术运算。数学命令在 1 位累加器的上升沿执行。数学命令包括：

命令	参数 1	参数 2	参数 3	描述
ON_ADD	临时寄存器 (值)	临时寄存器 (状态)	-	参数 1 = 参数 1 + 16 位累加器。
ON_SUB	临时寄存器 (值)	临时寄存器 (状态)	-	参数 1 = 参数 1 - 16 位累加器。
ON_MUL	临时寄存器 (最高有效字)	临时寄存器 (最低有效字)	临时寄存器 (状态)	参数 1:参数 2 = 16 位累加器 x 参数 2。
ON_DIV	临时寄存器 (最高有效字)	临时寄存器 (最低有效字)	临时寄存器 (状态)	参数1:参数2 = 参数1:参数2 / 16位累加器。
- 参数不适用于逻辑命令。				

逻辑命令

此章节内容

程序逻辑命令	125
布尔逻辑命令	127
寄存器逻辑命令	136
定时器逻辑命令	144
锁存逻辑命令	146
计数器逻辑命令	148
数学逻辑命令	150

程序逻辑命令

概述

程序逻辑命令用于：

- 向 Custom Logic Editor 标识程序文件
- 执行预定义运行模式

可以使用以下命令：

- LOGIC_ID
- CALL_EOM
- NOP

LOGIC_ID

LOGIC_ID 语句充当程序文件的标识符。

LOGIC_ID 值为整数，范围为 256 至 511。

参数个数	表示法
1	LOGIC_ID ID#

输入参数	类型	范围	描述
ID#	UINT	256-511	自定义程序的逻辑 ID

无输出参数。

CALL_EOM

CALL_EOM 将在自定义程序中执行预定义的运行模式。

参数个数	表示法
1	CALL_EOM OP_MODE#

输入参数	类型	范围	表示法
OP_MODE#	INT	12-22	嵌入式运行模式 (EOM) : <ul style="list-style-type: none"> • 12 : Overload • 13 : Direct Online • 14 : Reverse Direct Online • 15 : Star Delta

无输出参数。

NOP

NOP 命令不执行任何操作。

使用 NOP 命令作为程序文件中的占位符，用于替换预先存在的命令，或者用于为未来命令保留空间。

参数个数	表示法
0	NOP

NOP 命令没有参数。

布尔逻辑命令

概述

Custom Logic Editor 使用以下布尔逻辑命令：

- LOAD_K_BIT
- LOAD_BIT
- LOAD_TMP_BIT
- LOAD_NV_BIT
- LOAD_NOT_BIT
- LOAD_NOT_TMP_BIT
- LOAD_NOT_NV_BIT
- AND_BIT
- AND_TMP_BIT
- AND_NV_BIT
- AND_NOT_BIT
- AND_NOT_TMP_BIT
- AND_NOT_NV_BIT
- OR_BIT
- OR_TMP_BIT
- OR_NV_BIT
- OR_NOT_BIT
- OR_NOT_TMP_BIT
- OR_NOT_NV_BIT
- SET_BIT
- SET_TMP_BIT
- SET_NV_BIT
- SET_NOT_BIT
- SET_NOT_TMP_BIT
- SET_NOT_NV_BIT

LOAD_K_BIT

LOAD_K_BIT 命令将常量布尔值 (0 或 1) 加载到 1 位累加器中。

参数个数	表示法
1	LOAD_K_BIT KValue

输入参数	类型	范围	描述
KValue	BOOL	0/1	常量值

无输出参数。

LOAD_BIT

LOAD_BIT 命令将寄存器位的布尔值 (0 或 1) 加载到 1 位累加器中。

参数个数	表示法
2	LOAD_BIT RegAddr BitNo

输入参数	类型	范围	描述
RegAddr	UINT	0-9249	寄存器地址
BitNo	UINT	0-15	位编号

无输出参数。

LOAD_TMP_BIT

LOAD_TMP_BIT 命令将临时寄存器位的布尔值 (0 或 1) 加载到 1 位累加器中。

参数个数	表示法
2	LOAD_TMP_BIT TmpReg BitNo

输入参数	类型	范围	描述
TmpReg	UINT	0-299	临时寄存器编号
BitNo	UINT	0-15	位编号

无输出参数。

LOAD_NV_BIT

LOAD_NV_BIT 命令将非易失性寄存器位的布尔值 (0 或 1) 加载到 1 位累加器中。

参数个数	表示法
2	LOAD_NV_BIT NVReg BitNo

输入参数	类型	范围	描述
NVReg	UINT	0-63	非易失性寄存器编号
BitNo	UINT	0-15	位编号

无输出参数。

LOAD_NOT_BIT

LOAD_NOT_BIT 命令：

- 对指定寄存器位的布尔值 (0 或 1) 取反，然后
- 将取反值加载到 1 位累加器中。

参数个数	表示法
2	LOAD_NOT_BIT RegAddr BitNo

输入参数	类型	范围	描述
RegAddr	UINT	0-9249	寄存器地址
BitNo	UINT	0-15	位编号

无输出参数。

LOAD_NOT_TMP_BIT

LOAD_NOT_TMP_BIT 命令：

- 对指定临时寄存器位的布尔值 (0 或 1) 取反，然后
- 将取反值加载到 1 位累加器中。

参数个数	表示法
2	LOAD_NOT_TMP_BIT TmpReg BitNo

输入参数	类型	范围	描述
TmpReg	UINT	0-299	临时寄存器编号
BitNo	UINT	0-15	位编号

无输出参数。

LOAD_NOT_NV_BIT

LOAD_NOT_NV_BIT 命令：

- 对选定的非易失性寄存器位的布尔值 (0 或 1) 取反，然后
- 将取反值加载到 1 位累加器中。

参数个数	表示法
2	LOAD_NOT_NV_BIT NVReg BitNo

输入参数	类型	范围	描述
NVReg	UINT	0-63	非易失性寄存器编号
BitNo	UINT	0-15	位编号

无输出参数。

AND_BIT

AND_BIT 命令用于在逻辑存储器中的寄存器位值与累加器内容之间建立逻辑 AND 链接：

- 如果 1 位累加器等于 1 且链接的寄存器位等于 1，则 AND 处理的结果也为 1。
- 在所有其他情况下，AND 处理的结果都为 0。

结果保存在 1 位累加器中。

参数个数	表示法
2	AND_BIT RegAddr BitNo

输入参数	类型	范围	描述
RegAddr	UINT	0-9249	寄存器地址
BitNo	UINT	0-15	位编号

无输出参数。

AND_TMP_BIT

AND_TMP_BIT 命令用于在逻辑存储器中的临时寄存器位值与累加器内容之间建立逻辑 AND 链接。

- 如果 1 位累加器等于 1 且链接的临时寄存器位等于 1，则 AND 处理的结果也为 1。
- 在所有其他情况下，AND 处理的结果都为 0。

结果保存在 1 位累加器中。

参数个数	表示法
2	AND_TMP_BIT TmpReg BitNo

输入参数	类型	范围	描述
TmpReg	UINT	0-299	临时寄存器编号
BitNo	UINT	0-15	位编号

无输出参数。

AND_NV_BIT

AND_NV_BIT 命令用于在逻辑存储器中的非易失性寄存器位值与累加器内容之间建立逻辑 AND 链接。

- 如果 1 位累加器等于 1 且链接的非易失性寄存器位等于 1，则 AND 处理的结果也为 1。
- 在所有其他情况下，AND 处理的结果都为 0。

结果保存在 1 位累加器中。

参数个数	表示法
2	AND_NV_BIT NVReg BitNo

输入参数	类型	范围	描述
NVReg	UINT	0-63	非易失性寄存器编号
BitNo	UINT	0-15	位编号

无输出参数。

AND_NOT_BIT

AND_NOT_BIT 命令用于对指定寄存器位的布尔值 (0 或 1) 取反，然后在逻辑存储器中的这个值与累加器内容之间建立逻辑 AND 链接：

- 如果 1 位累加器等于 1 且链接的寄存器位等于 0，则 AND 处理的结果也为 1。
- 在所有其他情况下，AND 处理的结果都为 0。

结果保存在 1 位累加器中。

参数个数	表示法
2	AND_NOT_BIT RegAddr BitNo

输入参数	类型	范围	描述
RegAddr	UINT	0-9249	寄存器地址
BitNo	UINT	0-15	位编号

无输出参数。

AND_NOT_TMP_BIT

AND_NOT_TMP_BIT 命令用于对指定临时寄存器位的布尔值 (0 或 1) 取反，然后在逻辑存储器中的这个值与累加器内容之间建立逻辑 AND 链接：

- 如果 1 位累加器等于 1 且链接的临时寄存器位等于 0，则 AND 处理的结果也为 1。
- 在所有其他情况下，AND 处理的结果都为 0。

结果保存在 1 位累加器中。

参数个数	表示法
2	AND_NOT_TMP_BIT TmpReg BitNo

输入参数	类型	范围	描述
TmpReg	UINT	0-299	临时寄存器编号
BitNo	UINT	0-15	位编号

无输出参数。

AND_NOT_NV_BIT

AND_NOT_NV_BIT 命令用于对选定的非易失性寄存器位的布尔值 (0 或 1) 取反，然后在逻辑存储器中的这个值与累加器内容之间建立逻辑 AND 链接：

- 如果 1 位累加器等于 1 且链接的非易失性寄存器位等于 0，则 AND 处理的结果也为 1。
- 在所有其他情况下，AND 处理的结果都为 0。

结果保存在 1 位累加器中。

参数个数	表示法
2	AND_NOT_NV_BIT NVReg BitNo

输入参数	类型	范围	描述
NVReg	UINT	0-63	非易失性寄存器编号
BitNo	UINT	0-15	位编号

无输出参数。

OR_BIT

OR_BIT 命令用于在逻辑存储器中的寄存器位值与累加器内容之间建立逻辑 OR 链接：

- 如果 1 位累加器或寄存器位的值等于 1，则 OR 处理的结果也为 1。
- 如果进行比较的所有位的值都等于 0，则 OR 处理的结果为 0。

结果保存在 1 位累加器中。

参数个数	表示法
2	OR_BIT RegAddr BitNo

输入参数	类型	范围	描述
RegAddr	UINT	0-9249	寄存器地址
BitNo	UINT	0-15	位编号

无输出参数。

OR_TMP_BIT

OR_TMP_BIT 命令用于在逻辑存储器中的临时寄存器位值与累加器内容之间建立逻辑 OR 链接。

- 如果 1 位累加器或临时寄存器位的值等于 1，则 OR 处理的结果也为 1。
- 如果进行比较的所有位的值都等于 0，则 OR 处理的结果为 0。

结果保存在 1 位累加器中。

参数个数	表示法
2	OR_TMP_BIT TmpReg BitNo

输入参数	类型	范围	描述
TmpReg	UINT	0-299	临时寄存器编号
BitNo	UINT	0-15	位编号

无输出参数。

OR_NV_BIT

OR_NV_BIT 命令用于在逻辑存储器中的非易失性寄存器位值与累加器内容之间建立逻辑 OR 链接。

- 如果 1 位累加器或非易失性寄存器位的值等于 1，则 OR 处理的结果也为 1。
- 如果进行比较的所有位的值都等于 0，则 OR 处理的结果为 0。

结果保存在 1 位累加器中。

参数个数	表示法
2	OR_NV_BIT NVReg BitNo

输入参数	类型	范围	描述
NVReg	UINT	0-63	非易失性寄存器编号
BitNo	UINT	0-15	位编号

无输出参数。

OR_NOT_BIT

OR_NOT_BIT 命令用于对指定寄存器位的布尔值 (0 或 1) 取反，然后在逻辑存储器中的这个值与累加器内容之间建立逻辑 OR 链接：

- 如果 1 位累加器或寄存器位的值等于 0，则 OR 处理的结果也为 1。
- 如果进行比较的所有位的值都等于 0，则 OR 处理的结果为 0。

结果保存在 1 位累加器中。

参数个数	表示法
2	OR_NOT_BIT RegAddr BitNo

输入参数	类型	范围	描述
RegAddr	UINT	0-9249	寄存器地址
BitNo	UINT	0-15	位编号

无输出参数。

OR_NOT_TMP_BIT

OR_NOT_TMP_BIT 命令用于对指定临时寄存器位的布尔值 (0 或 1) 取反，然后在逻辑存储器中的这个值与累加器内容之间建立逻辑 OR 链接：

- 如果 1 位累加器或临时寄存器位的值等于 0，则 OR 处理的结果也为 1。
- 如果进行比较的所有位的值都等于 0，则 OR 处理的结果为 0。

结果保存在 1 位累加器中。

参数个数	表示法
2	OR_NOT_TMP_BIT TmpReg BitNo

输入参数	类型	范围	描述
TmpReg	UINT	0-299	临时寄存器编号
BitNo	UINT	0-15	位编号

无输出参数。

OR_NOT_NV_BIT

OR_NOT_NV_BIT 命令用于对选定的非易失性寄存器位的布尔值 (0 或 1) 取反，然后在逻辑存储器中的这个值与累加器内容之间建立逻辑 OR 链接：

- 如果 1 位累加器或非易失性寄存器位的值等于 0，则 OR 处理的结果也为 1。
- 如果进行比较的所有位的值都等于 0，则 OR 处理的结果为 0。

结果保存在 1 位累加器中。

参数个数	表示法
2	OR_NOT_NV_BIT NVReg BitNo

输入参数	类型	范围	描述
NVReg	UINT	0-63	非易失性寄存器编号
BitNo	UINT	0-15	位编号

无输出参数。

SET_BIT

SET_BIT 命令将 1 位累加器中的值设置到指定的寄存器位。

参数个数	表示法
2	SET_BIT RegAddr BitNo

无输入参数。

输出参数	类型	范围	描述
RegAddr	UINT	0-9249	寄存器地址
BitNo	UINT	0-15	位编号

SET_TMP_BIT

SET_TMP_BIT 命令将 1 位累加器中的值设置到指定的临时寄存器位。

参数个数	表示法
2	SET_TMP_BIT TmpReg BitNo

无输入参数。

输出参数	类型	范围	描述
TmpReg	UINT	0-299	临时寄存器编号
BitNo	UINT	0-15	位编号

SET_NV_BIT

SET_NV_BIT 命令将 1 位累加器中的值设置到指定的非易失性寄存器位。

参数个数	表示法
2	SET_NV_BIT NVReg BitNo

无输入参数。

输出参数	类型	范围	描述
NVReg	UINT	0-63	非易失性寄存器编号
BitNo	UINT	0-15	位编号

SET_NOT_BIT

SET_NOT_BIT 命令将 1 位累加器中反转的值设置到指定的寄存器位。

参数个数	表示法
2	SET_NOT_BIT RegAddr BitNo

无输入参数。

输出参数	类型	范围	描述
RegAddr	UINT	0-9249	寄存器地址
BitNo	UINT	0-15	位编号

SET_NOT_TMP_BIT

SET_NOT_TMP_BIT 命令将 1 位累加器中反转的值设置到指定的临时寄存器位。

参数个数	表示法
2	SET_NOT_TMP_BIT TmpReg BitNo

无输入参数。

输出参数	类型	范围	描述
TmpReg	UINT	0-299	临时寄存器编号
BitNo	UINT	0-15	位编号

SET_NOT_NV_BIT

SET_NOT_NV_BIT 命令将 1 位累加器中反转的值设置到指定的非易失性寄存器位。

参数个数	表示法
2	SET_NOT_NV_BIT NVReg BitNo

无输入参数。

输出参数	类型	范围	描述
NVReg	UINT	0-63	非易失性寄存器编号
BitNo	UINT	0-15	位编号

寄存器逻辑命令

概述

寄存器命令用于计算和控制 16 位值。

Custom Logic Editor 使用以下寄存器命令：

- LOAD_K_REG
- LOAD_REG
- LOAD_TMP_REG
- LOAD_NV_REG
- COMP_K_REG
- COMP_REG
- COMP_TMP_REG
- COMP_NV_REG
- AND_K
- AND_REG
- AND_TMP_REG
- AND_NV_REG
- OR_K
- OR_REG
- OR_TMP_REG
- OR_NV_REG
- XOR_K
- XOR_REG
- XOR_TMP_REG
- XOR_NV_REG
- ON_SET_REG
- ON_SET_TMP_REG
- ON_SET_NV_REG

LOAD_K_REG

LOAD_K_REG 命令可将 16 位常量值加载到逻辑存储器的 16 位累加器中。

参数个数	表示法
1	LOAD_K_REG KValue

输入参数	类型	范围	描述
KValue	UINT	0-65,535	常量值

无输出参数。

LOAD_REG

LOAD_REG 命令可将寄存器的副本加载到逻辑存储器的 16 位累加器中。

参数个数	表示法
1	LOAD_REG RegAddr

输入参数	类型	范围	描述
RegAddr	UINT	0-9249	寄存器地址

无输出参数。

LOAD_TMP_REG

LOAD_TMP_REG 命令将临时寄存器的副本加载到逻辑存储器的 16 位累加器中。

参数个数	表示法
1	LOAD_TMP_REG TmpReg

输入参数	类型	范围	描述
TmpReg	UINT	0-299	临时寄存器编号

无输出参数。

LOAD_NV_REG

LOAD_NV_REG 命令可将非易失性寄存器的副本加载到逻辑存储器的 16 位累加器中。

参数个数	表示法
1	LOAD_NV_REG NVReg

输入参数	类型	范围	描述
NVReg	UINT	0-63	非易失性寄存器编号

无输出参数。

COMP_K_REG

COMP_K_REG 命令可将 16 位累加器的内容与参数 1 常量值进行比较，并通过参数 2 临时寄存器的一位来设置比较结果。

参数个数	表示法
2	COMP_K_REG KValue TmpReg

输入参数	类型	范围/位	描述
KValue	UINT	0-65,535	常量值

输出参数	类型	范围/位	描述
TmpReg	UINT	位 1	16 位累加器 < KValue
		位 2	16 位累加器 = KValue
		位 3	16 位累加器 > KValue

COMP_REG

COMP_REG 命令可将 16 位累加器的内容与参数 1 定义的寄存器的内容进行比较，并通过参数 2 临时寄存器的一位来设置比较结果。

参数个数	表示法
2	COMP_REG RegAddr TmpReg

输入参数	类型	范围/位	描述
RegAddr	UINT	0-9249	寄存器地址

输出参数	类型	范围/位	描述
TmpReg	UINT	位 1	16 位累加器 < RegAddr
		位 2	16 位累加器 = RegAddr
		位 3	16 位累加器 > RegAddr

COMP_TMP_REG

COMP_TMP_REG 命令可将 16 位累加器的内容与参数 1 定义的临时寄存器的内容进行比较，并通过参数 2 临时寄存器的一位来设置比较结果。

参数个数	表示法
2	COMP_TMP_REG TmpReg1 TmpReg2

输入参数	类型	范围/位	描述
TmpReg1	UINT	0-299	临时寄存器编号

输出参数	类型	范围/位	描述
TmpReg2	UINT	位 1	16 位累加器 < TmpReg1
		位 2	16 位累加器 = TmpReg1
		位 3	16 位累加器 > TmpReg1

COMP_NV_REG

COMP_NV_REG 命令可将 16 位累加器的内容与参数 1 定义的非易失性寄存器的内容进行比较，并通过参数 2 临时寄存器的一位来设置比较结果。

参数个数	表示法
2	COMP_NV_REG NVReg TmpReg

输入参数	类型	范围/位	描述
NVReg	UINT	0-63	非易失性寄存器编号

输出参数	类型	范围/位	描述
TmpReg	UINT	位 1	16 位累加器 < NVReg
		位 2	16 位累加器 = NVReg
		位 3	16 位累加器 > NVReg

AND_K

AND_K 命令用于在逻辑存储器中的 16 位常量值与 16 位累加器内容之间建立逻辑 AND 链接。结果保存在 16 位累加器中。

AND 处理将 16 位累加器中的每一位与所链接的 16 位常量值中的相应位进行比较：

- 如果两个位都等于 1，则该位编号的 AND 处理的结果也为 1。
- 在所有其他情况下，该位编号的 AND 处理的结果都为 0。

参数个数	表示法
1	AND_K KValue

输入参数	类型	范围	描述
KValue	UINT	0-65,535	常量值

无输出参数。

AND_REG

AND_REG 命令用于在逻辑存储器中的寄存器值与 16 位累加器内容之间建立逻辑 AND 链接。结果保存在 16 位累加器中。

AND 处理将 16 位累加器中的每一位与所链接的寄存器中的相应位进行比较：

- 如果两个位都等于 1，则该位编号的 AND 处理的结果也为 1。
- 在所有其他情况下，该位编号的 AND 处理的结果都为 0。

参数个数	表示法
1	AND_REG RegAddr

输入参数	类型	范围	描述
RegAddr	UINT	0-9249	寄存器地址

无输出参数。

AND_TMP_REG

AND_TMP_REG 命令用于在逻辑存储器中的临时寄存器值与 16 位累加器内容之间建立逻辑 AND 链接。结果保存在 16 位累加器中。

AND 处理将 16 位累加器中的每一位与所链接的临时寄存器中的相应位进行比较：

- 如果两个位都等于 1，则该位编号的 AND 处理的结果也为 1。
- 在所有其他情况下，该位编号的 AND 处理的结果都为 0。

参数个数	表示法
1	AND_TMP_REG TmpReg

输入参数	类型	范围	描述
TmpReg	UINT	0-299	临时寄存器编号

无输出参数。

AND_NV_REG

AND_NV_REG 命令用于在逻辑存储器中的非易失性寄存器值与 16 位累加器内容之间建立逻辑 AND 链接。结果保存在 16 位累加器中。

AND 处理将 16 位累加器中的每一位与所链接的非易失性寄存器中的相应位进行比较：

- 如果两个位都等于 1，则该位编号的 AND 处理的结果也为 1。
- 在所有其他情况下，该位编号的 AND 处理的结果都为 0。

参数个数	表示法
1	AND_NV_REG NVReg

输入参数	类型	范围	描述
NVReg	UINT	0-63	非易失性寄存器编号

无输出参数。

OR_K

OR_K 命令用于在逻辑存储器中的 16 位常量值与 16 位累加器内容之间建立逻辑 OR 链接。结果保存在 16 位累加器中。

OR 处理将 16 位累加器中的每一位与所链接的 16 位常量值中的相应位进行比较：

- 如果进行比较的任何位都等于 1，则该位编号的 OR 处理的结果也为 1。
- 如果进行比较的所有位都等于 0，则该位编号的 OR 处理的结果为 0。

参数个数	表示法
1	OR_K KValue

输入参数	类型	范围	描述
KValue	UINT	0-65,535	常量值

无输出参数。

OR_REG

OR_REG 命令用于在逻辑存储器中的寄存器值与 16 位累加器内容之间建立逻辑 OR 链接。结果保存在 16 位累加器中。

OR 处理将 16 位累加器中的每一位与所链接的寄存器中的相应位进行比较：

- 如果进行比较的任何位都等于 1，则该位编号的 OR 处理的结果也为 1。
- 如果进行比较的所有位都等于 0，则该位编号的 OR 处理的结果为 0。

参数个数	表示法
1	OR_REG RegAddr

输入参数	类型	范围	描述
RegAddr	UINT	0-9249	寄存器地址

无输出参数。

OR_TMP_REG

OR_TMP_REG 命令用于在逻辑存储器中的临时寄存器值与 16 位累加器内容之间建立逻辑 OR 链接。结果保存在 16 位累加器中。

OR 处理将 16 位累加器中的每一位与所链接的临时寄存器中的相应位进行比较：

- 如果进行比较的任何位都等于 1，则该位编号的 OR 处理的结果也为 1。
- 如果进行比较的所有位都等于 0，则该位编号的 OR 处理的结果为 0。

参数个数	表示法
1	OR_TMP_REG TmpReg

输入参数	类型	范围	描述
TmpReg	UINT	0-299	临时寄存器编号

无输出参数。

OR_NV_REG

OR_NV_REG 命令用于在逻辑存储器中的非易失性寄存器值与 16 位累加器内容之间建立逻辑 OR 链接。结果保存在 16 位累加器中。

OR 处理将 16 位累加器中的每一位与所链接的非易失性寄存器中的相应位进行比较：

- 如果进行比较的任何位都等于 1，则该位编号的 OR 处理的结果也为 1。
- 如果进行比较的所有位都等于 0，则该位编号的 OR 处理的结果为 0。

参数个数	表示法
1	OR_NV_REG NVReg

输入参数	类型	范围	描述
NVReg	UINT	0-63	非易失性寄存器编号

无输出参数。

XOR_K

XOR_K 命令用于在逻辑存储器中的 16 位常量值与 16 位累加器内容之间建立逻辑异 OR 链接。结果保存在 16 位累加器中。

XOR 处理将 16 位累加器中的每一位与所链接的 16 位常量值中的相应位进行比较得出以下结果：

- 如果一个位等于 1 而另一个位等于 0，则 XOR 处理的结果为 1。
- 在所有其他情况下，XOR 处理的结果都为 0。

参数个数	表示法
1	XOR_K KValue

输入参数	类型	范围	描述
KValue	UINT	0-65,535	常量值

无输出参数。

XOR_REG

XOR_REG 命令用于在逻辑存储器中的寄存器值与 16 位累加器内容之间建立逻辑异 OR 链接。结果保存在 16 位累加器中。

XOR 处理将 16 位累加器中的每一位与所链接的寄存器中的相应位进行比较得出以下结果：

- 如果一个位等于 1 而另一个位等于 0，则 XOR 处理的结果为 1。
- 在所有其他情况下，XOR 处理的结果都为 0。

参数个数	表示法
1	XOR_REG RegAddr

输入参数	类型	范围	描述
RegAddr	UINT	0-9249	寄存器地址

无输出参数。

XOR_TMP_REG

XOR_TMP_REG 命令用于在逻辑存储器中的临时寄存器值与 16 位累加器内容之间建立逻辑异 OR 链接。结果保存在 16 位累加器中。

XOR 处理将 16 位累加器中的每一位与所链接的临时寄存器中的相应位进行比较得出以下结果：

- 如果一个位等于 1 而另一个位等于 0，则 XOR 处理的结果为 1。
- 在所有其他情况下，XOR 处理的结果都为 0。

参数个数	表示法
1	XOR_TMP_REG TmpReg

输入参数	类型	范围	描述
TmpReg	UINT	0-299	临时寄存器编号

无输出参数。

XOR_NV_REG

XOR_NV_REG 命令用于在逻辑存储器中的非易失性寄存器值与 16 位累加器内容之间建立逻辑异 OR 链接。结果保存在 16 位累加器中。

XOR 处理将 16 位累加器中的每一位与所链接的非易失性寄存器中的相应位进行比较得出以下结果：

- 如果一个位等于 1 而另一个位等于 0，则 XOR 处理的结果为 1。
- 在所有其他情况下，XOR 处理的结果都为 0。

参数个数	表示法
1	XOR_NV_REG NVReg

输入参数	类型	范围	描述
NVReg	UINT	0-63	非易失性寄存器编号

无输出参数。

ON_SET_REG

ON_SET_REG 命令在 1 位累加器的上升沿将 16 位累加器中的值复制到指定寄存器。

参数个数	表示法
2	ON_SET_REG RegAddr TmpReg

无输入参数。

输出参数	类型	范围/位	描述
RegAddr	UINT	0-9249	寄存器地址
TmpReg	UINT	位 3	1 位累加器历史位

ON_SET_TMP_REG

ON_SET_TMP_REG 命令在 1 位累加器的上升沿将 16 位累加器中的值复制到指定临时寄存器。

参数个数	表示法
2	ON_SET_TMP_REG TmpReg1 TmpReg2

无输入参数。

输出参数	类型	范围/位	描述
TmpReg1	UINT	0-299	临时寄存器编号
TmpReg2	UINT	位 3	1 位累加器历史位

ON_SET_NV_REG

ON_SET_NV_REG 命令在 1 位累加器的上升沿将 16 位累加器中的值复制到指定非易失性寄存器。

参数个数	表示法
1	ON_SET_NV_REG NVReg1 NVReg2

无输入参数。

输出参数	类型	范围/位	描述
NVReg1	UINT	0-63	非易失性寄存器编号
NVReg2	UINT	位 3	1 位累加器历史位

定时器逻辑命令

概述

Custom Logic Editor 使用以下定时器命令：

- TIMER_SEC
- TIMER_TENTHS

注: 将自定义逻辑文件上传到仿真器并启用了定时器后，它将按照预期运行。但是，一旦启用并执行了定时器，便无法再次启用该定时器。这是 TIMER_SEC 和 TIMER_TENTHS 的限制。如要再次启用定时器，应重新上传自定义逻辑文件。在设计和测试使用定时器的自定义逻辑文件时，建议考虑此限制。

TIMER_SEC

TIMER_SEC 命令：

- 以秒计时，最大值为临时寄存器指定的计数值
- 在第二个临时寄存器中计算结束时间
- 由第三个临时寄存器启用，并向该寄存器报告其计数状态

参数个数	表示法
3	TIMER_SEC TmpReg1 TmpReg2 TmpReg3

输入参数	类型	范围/位	描述
TmpReg1	UINT	0-65,535	定时器预设值
TmpReg3	UINT	位 0	<ul style="list-style-type: none"> • 在上升沿启动定时器 • 在下降沿停止定时器

输出参数	类型	范围/位	描述
TmpReg2	UINT	0-65,535	计算出的结束时间
TmpReg3	UINT	位 1	完成定时器操作： <ul style="list-style-type: none"> • 定时器达到 TmpReg2 时设置此位 • 在以下情况下复位此位： <ul style="list-style-type: none"> ◦ TmpReg3.Bit0 已复位 ◦ 重置电源
		位 2	定时器正在执行操作 当定时器达到 TmpReg2 时将位复位
		位 3	TmpReg3.Bit0 历史位
		位 4	保留

TIMER_TENTHS

TIMER_TENTHS 命令：

- 以十分之几秒计时，最大值为临时寄存器指定的计数值。
- 在第二个临时寄存器中计算结束时间。
- 由第三个临时寄存器启用，并向该寄存器报告其计数状态。

注: TIMER_TENTHS 命令的乘数为 10。例如，如果所需值为 6 秒，则输入值应乘以系数 10，表示提供的输入值应为 60 秒。

参数个数	表示法
3	TIMER_TENTHS TmpReg1 TmpReg2 TmpReg3

输入参数	类型	范围/位	描述
TmpReg1	UINT	0-65,535	定时器预设值
TmpReg3	UINT	位 0	<ul style="list-style-type: none"> • 在上升沿启动定时器 • 在下降沿停止定时器

输出参数	类型	范围/位	描述
TmpReg2	UINT	0-65,535	计算出的结束时间
TmpReg3	UINT	位 1	完成定时器操作： <ul style="list-style-type: none"> • 当定时器达到 TmpReg2 时被置位 • 在以下情况下复位此位： <ul style="list-style-type: none"> ◦ TmpReg3.Bit0 已复位 ◦ 重置电源
		位 2	定时器正在执行操作 当定时器达到 TmpReg2 时将位复位
		位 3	TmpReg3.Bit0 历史位
		位 4	保留

锁存逻辑命令

概述

Custom Logic Editor 使用以下锁存命令：

- LATCH
- LATCH_NV

LATCH

LATCH 命令：

- 在临时寄存器中存储布尔值（0 或 1）
- 提供用于设置和复位存储值的方法
- 保存上次扫描的清除和设置状态

参数个数	表示法
1	LATCH TmpReg

输入参数	类型	位	描述
TmpReg	UINT	位 1	在上升沿将 TmpReg.Bit0 设为 1
		位 2	在下降沿，将 TmpReg.Bit1 复位为 0

输出参数	类型	位	描述
TmpReg	UINT	位 0	锁存状态
		位 3	TmpReg.Bit1 历史位
		位 4	TmpReg.Bit2 历史位

LATCH_NV

LATCH_NV 命令：

- 在非易失性寄存器中存储布尔值（0 或 1）
- 提供用于设置和复位存储值的方法
- 保存上次扫描的清除和设置状态

使用 LATCH_NV 命令而不是 LATCH 命令，可在电源重置期间保持锁存状态。

参数个数	表示法
1	LATCH_NV NVReg

输入参数	类型	位	描述
NVReg	UINT	位 1	在上升沿将 TmpReg.Bit0 设为 1
		位 2	在上升沿将 TmpReg.Bit0 复位为 0

输出参数	类型	位	描述
NVReg	UINT	位 0	锁存状态
		位 3	TmpReg.Bit1 历史位
		位 4	TmpReg.Bit2 历史位

计数器逻辑命令

概述

Custom Logic Editor 使用以下计数器逻辑命令：

- COUNTER
- COUNTER_NV

COUNTER

COUNTER 命令：

- 递增或递减计数值
- 提供用于将计数值设置为预设值的方法
- 指示计数值何时等于 0
- 指示计数值与预设值之间的关系 - 等于、大于或小于
- 保存上次扫描的递增、递减和设置状态

参数个数	表示法
3	COUNTER TmpReg1 KValue TmpReg2

输入参数	类型	范围/位	描述
KValue	UINT	0-65,535	计数器预设值
TmpReg2	UINT	位 4	在上升沿上增加计数器当前值。计数器当前值应介于 0 至 65,535 之间。
		位 5	在下降沿上递减计数器当前值。计数器当前值应介于 65,535 至 0 之间。
		位 6	在上升沿将当前计数器值设置为预设值

输出参数	类型	范围/位	描述
TmpReg1	UINT	0-65,535	计数器当前值
TmpReg2	UINT	位 0	计数器当前值为 0 : TmpReg1=0
		位 1	计数器当前值小于预设值 : TmpReg1<KValue
		位 2	计数器当前值等于预设值 : TmpReg1=KValue
		位 3	计数器当前值大于预设值 : TmpReg1>KValue
		位 7	TmpReg2.Bit4 历史位
		位 8	TmpReg2.Bit5 历史位
位 9	TmpReg2.Bit6 历史位		

COUNTER_NV

COUNTER_NV 命令：

- 递增或递减计数值
- 提供用于将计数值设置为预设值的方法
- 指示计数值何时等于 0
- 指示计数值与预设值之间的关系 - 等于、大于或小于
- 保存上次扫描的递增、递减和设置状态

使用 COUNTER_NV 命令而不是 COUNTER 命令，可在电源循环期间保持计数。

参数个数	表示法
3	COUNTER NVReg1 KValue NVReg2

输入参数	类型	范围/位	描述
KValue	UINT	0-65,535	计数器预设值
NVReg2	UINT	位 4	在上升沿上增加计数器当前值。计数器当前值应介于 0 至 65,535 之间。
		位 5	在下降沿上递减计数器当前值。计数器当前值应介于 65,535 至 0 之间。
		位 6	在上升沿将当前计数器值设置为预设值

输出参数	类型	范围/位	描述
NVReg1	UINT	0-65,535	计数器当前值
NVReg2	UINT	位 0	计数器当前值为 0 : NVReg1=0
		位 1	计数器当前值小于预设值 : NVReg1<KValue
		位 2	计数器当前值等于预设值 : NVReg1=KValue
		位 3	计数器当前值大于预设值 : NVReg1>KValue
		位 7	NVReg2.Bit4 历史位
		位 8	NVReg2.Bit5 历史位
		位 9	NVReg2.Bit6 历史位

数学逻辑命令

概述

Custom Logic Editor 使用以下数学命令：

- ON_ADD
- ON_SUB
- ON_MUL
- ON_DIV

ON_ADD

当 1 位累加器从 0 转换为 1 时，ON_ADD 命令执行无符号加法。它将参数 1 中的值与 16 位累加器值相加，然后将结果发回参数 1 中的值。

状态寄存器：

- 如果加法处理的结果超过 65,535，则指示溢出
- 指示上次扫描的 1 位累加器的状态

参数个数	表示法
2	ON_ADD TmpReg1 TmpReg2

输入参数	类型	范围/位	描述
TmpReg1	UINT	0-65,535	要与 16 位累加器相加的值

输出参数	类型	范围/位	描述
TmpReg1	UINT	0-65,535	加法运算的结果
TmpReg2	UINT	位 0	上溢：加法结果大于 65,535。 在此情况下，加法结果等于参数 1 + 65,536。
		位 3	1 位累加器历史位

ON_SUB

当 1 位累加器从 0 转换为 1 时，ON_SUB 命令执行无符号减法。它从参数 1 中的值中减去 16 位累加器值，然后将结果发回参数 1 中的值。

状态寄存器：

- 如果减法处理的结果小于 0，则指示下溢
- 指示上次扫描的 1 位累加器的状态

参数个数	表示法
2	ON_SUB TmpReg1 TmpReg2

输入参数	类型	范围/位	描述
TmpReg1	UINT	0-65,535	要从 16 位累加器中减去的值

输出参数	类型	范围/位	描述
TmpReg1	UINT	0-65,535	减法运算的结果
TmpReg2	UINT	位 0	下溢：减法结果小于 0。 在此情况下，运算的真正结果等于输出到参数 1 - 65,536 的结果。
		位 3	1 位累加器历史位

ON_MUL

当 1 位累加器从 0 转换为 1 时，ON_MUL 命令执行无符号乘法。ON_MUL 过程将来自参数 2 的值与 16 位累加器值相乘，然后将结果发回参数 1 (最高有效字) 和参数 2 (最低有效字)。

状态寄存器指示上次扫描的 1 位累加器的状态。

参数个数	表示法
3	ON_MUL TmpReg1 TmpReg2 TmpReg3

输入参数	类型	范围/位	描述
TmpReg2	UINT	0-65,535	要与 16 位累加器相乘的值

输出参数	类型	范围/位	描述
TmpReg1 和 TmpReg2	UINT	0-65,535	乘法运算的结果： <ul style="list-style-type: none"> • TmpReg1 用于保存最高有效字 • TmpReg2 用于保存最低有效字
TmpReg3	UINT	位 3	1 位累加器历史位

ON_DIV

当 1 位累加器从 0 转换为 1 时，ON_DIV 命令执行无符号除法。ON_DIV 过程将参数 1 和参数 2 的组合值除以 16 位累加器值，然后将结果发回参数 1 (最高有效字) 和参数 2 (最低有效字)。

状态寄存器指示：

- 上溢 (被除数为 0 时)
- 上次扫描的 1 位累加器的状态

参数个数	表示法
3	ON_DIV TmpReg1 TmpReg2 TmpReg3

输入参数	类型	范围/位	描述
TmpReg1 和 TmpReg2	UINT	0-65,535	要除以 16 位累加器的值

输出参数	类型	范围/位	描述
TmpReg1 和 TmpReg2	UINT	0-65,535	除法运算的结果： <ul style="list-style-type: none"> • TmpReg1 用于保存最高有效字 • TmpReg2 用于保存最低有效字
TmpReg3	UINT	位 0	被零除
		位 3	1 位累加器历史位

自定义逻辑程序示例

此章节内容

如何检查定时器和乘法命令	153
如何创建真值表	154

如何检查定时器和乘法命令

概述

自定义应用程序时，您可能需要检查定时器和乘法命令。

使用自定义逻辑程序检查定时器和乘法命令

下图以文本视图形式展示了自定义逻辑程序中如何检查定时器并执行乘法指令的操作说明。

```
LOGIC_ID 356
// A very simple test that checks timers and MUL (multiply command)
// It should switch LO1 and LO2 ON OFF if OK !!
//
LOAD_K_BIT 1
SET_TMP_BIT 115 3
LOAD_TMP_REG 115
ON_SET_TMP_REG 105 111
ON_SET_TMP_REG 108 112
LOAD_NOT_TMP_BIT 110 2 // timer 2 not timing
SET_TMP_BIT 107 0
TIMER_TENTHS 105 106 107
LOAD_NOT_TMP_BIT 107 2 // timer 1 not timing
SET_TMP_BIT 110 0
TIMER_TENTHS 108 109 110
LOAD_TMP_BIT 107 2
SET_BIT 1200 12 // Switch LO1 if timer 1 is working
LOAD_K_REG 50 // Load value of 50
LOAD_K_BIT 1
SET_NOT_TMP_BIT 123 3 // Clear history bit
ON_SET_TMP_REG 122 123 // Save the 50 in temporary register 22
LOAD_K_REG 2 // Load value of 2
SET_NOT_TMP_BIT 123 3
ON_MUL 121 122 123 // Multiply 50x2
LOAD_TMP_REG 122
COMP_K_REG 100 101 // Is result 100?
LOAD_TMP_BIT 110 2 // timer 2 timing
AND_TMP_BIT 101 2 // =100?
SET_BIT 1200 13 // Don't switch LO2 if MUL did not work OK
```

如何创建真值表

概述

自定义应用程序时，您可能需要创建真值表。

使用自定义逻辑程序创建真值表

下图以文本视图形式显示了真值表创建的自定义逻辑程序：

```

LOGIC_ID 444
//
//
// Truth table example
//
//   I1  I2  I3   Output
//   0   0   0     0   (0)
//   0   0   1     1   (1)
//   0   1   0     1   (2)
//   0   1   1     0   (3)
//   1   0   0     1   (4)
//   1   0   1     0   (5)
//   1   1   0     0   (6)
//   1   1   1     0   (7)

LOAD_BIT 457.0           //SET INPUTS
SET_TMP_BIT 1.1
LOAD_BIT 457.1
SET_TMP_BIT 1.2
LOAD_BIT 457.2
SET_TMP_BIT 1.3

//
//**** 3x1 TRUTH TABLE TEMPLATE
//**** Inputs defined as bits 1.1 through 1.3)
//**** Output defined as bit 1.15
//
LOAD_K_BIT 0             //default output OFF
SET_TMP_BIT 1.15        //save partial result

//*****0** Inputs 1-2-3 are OFF OFF OFF
//
LOAD_NOT_TMP_BIT 1.1    //include this SECTION
AND_NOT_TMP_BIT 1.2    //if output is to be ON
AND_NOT_TMP_BIT 1.3    //REMOVE if output to be OFF
SET_TMP_BIT 1.15       //save partial result
//

```

使用自定义逻辑程序创建真值表 (续)

```

LOAD_NOT_TMP_BIT 1.1 //include this SECTION
AND_NOT_TMP_BIT 1.2 //if output is to be ON
AND_TMP_BIT 1.3 //REMOVE if output to be OFF
OR_TMP_BIT 1.15 //include previous result
SET_TMP_BIT 1.15 //save partial result
//
//*****2** Inputs 1-2-3 are OFF ON OFF
//
LOAD_NOT_TMP_BIT 1.1 //include this SECTION
AND_TMP_BIT 1.2 //if output is to be ON
AND_NOT_TMP_BIT 1.3 //REMOVE if output to be OFF
OR_TMP_BIT 1.15 //include previous result
SET_TMP_BIT 1.15 //save partial result
//
//*****3** Inputs 1-2-3 are OFF ON ON
//
LOAD_NOT_TMP_BIT 1.1 //include this SECTION
AND_TMP_BIT 1.2 //if output is to be ON
AND_TMP_BIT 1.3 //REMOVE if output to be OFF
OR_TMP_BIT 1.15 //include previous result
SET_TMP_BIT 1.15 //save partial result
//
//*****4** Inputs 1-2-3 are ON OFF OFF
//
LOAD_TMP_BIT 1.1 //include this SECTION
AND_NOT_TMP_BIT 1.2 //if output is to be ON
AND_NOT_TMP_BIT 1.3 //REMOVE if output to be OFF
OR_TMP_BIT 1.15 //include previous result
SET_TMP_BIT 1.15 //save partial result
//
//*****5** Inputs 1-2-3 are ON OFF ON
//
LOAD_TMP_BIT 1.1 //include this SECTION
AND_NOT_TMP_BIT 1.2 //if output is to be ON
AND_TMP_BIT 1.3 //REMOVE if output to be OFF
OR_TMP_BIT 1.15 //include previous result
SET_TMP_BIT 1.15 //save partial result

```

使用自定义逻辑程序创建真值表 (续)

```
//  
//*****6** Inputs 1-2-3 are ON ON OFF  
//  
LOAD_TMP_BIT 1.1 //include this SECTION  
AND_TMP_BIT 1.2 //if output is to be ON  
AND_NOT_TMP_BIT 1.3 //REMOVE if output to be OFF  
OR_TMP_BIT 1.15 //include previous result  
SET_TMP_BIT 1.15 //save partial result  
//  
//*****7** Inputs 1-2-3 are ON ON ON  
//  
LOAD_TMP_BIT 1.1 //include this SECTION  
AND_TMP_BIT 1.2 //if output is to be ON  
AND_TMP_BIT 1.3 //REMOVE if output to be OFF  
OR_TMP_BIT 1.15 //include previous result  
SET_TMP_BIT 1.15 //save partial result  
  
LOAD_TMP_BIT 1.15 //SET OUTPUT  
SET_BIT 1200.14
```

功能块图语言

此部分内容

FBD 语言概述	158
FBD 元素	161
使用 FBD 语言进行编程	176
操作 FBD 功能块	182
FBD 编辑器显示选项	186

FBD 语言概述

此章节内容

FBD 编辑器简介	159
-----------------	-----

FBD 编辑器简介

概述

FBD 编辑器是 TeSys Tera DTM Library 的组件。使用 FBD 编辑器，可查看现有的 FBD 程序文件，或者可使用 FBD 语言（而不是基于指令的文本编程语言）来创建 FBD 程序文件。

创建 FBD 程序

如要打开 FBD 编辑器，请选择 **Device > FB Diagram > New FB Diagram**，或者选择 **FB Diagram** 选项卡中。FBD 编辑器将出现在主窗口中。

保存 FBD 程序

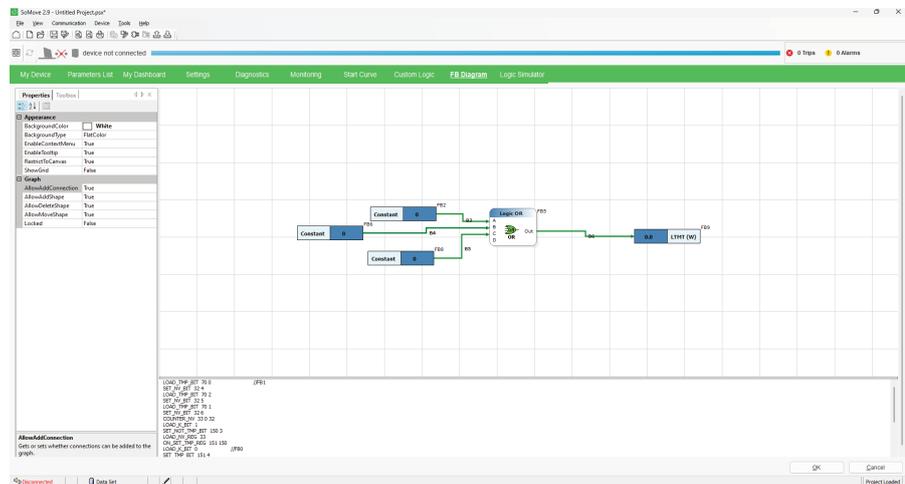
编译 FBD 程序之前，必须保存它。如要保存已创建或编辑的程序，请选择 **Device > FB Diagram > Save FB Diagram as**。

注：文件使用扩展名 *.Gef 进行保存。

FBD 编辑器用户界面

即使 TeSys Tera DTM Library 处于连接模式，也可使用 FBD 编辑器。但是，仅当在 FBD 编辑器中打开 FBD 程序时，才能启用许多菜单项。

下面的窗口显示了在 FBD 编辑器中打开的 FBD 程序：



工作区

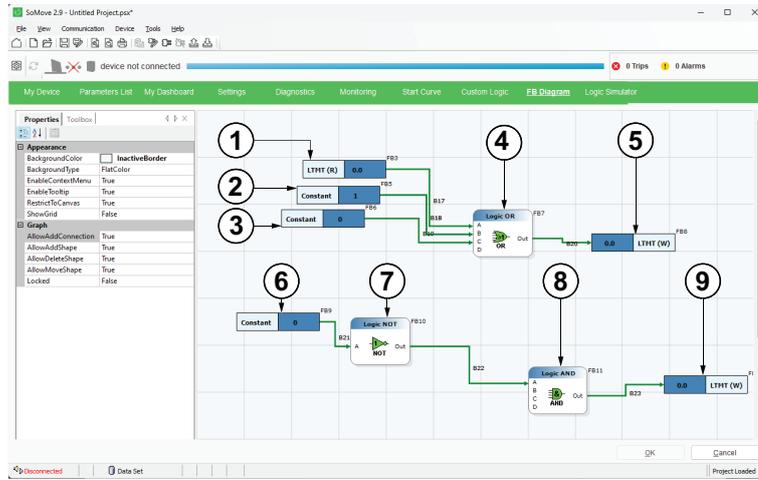
FBD 程序在工作区中进行编辑和创建。

工作区由两个元素组成：

- 功能块
- 链接功能块的线

运行 FBD 程序

FBD 程序逐行、从左至右、从上至下运行。在下面的示例中，按照编号指示的顺序从指令 1 到指令 5 和指令 6 到指令 9 执行指令。



FBD 元素

此章节内容

启动器类型	162
计算功能块	164
Inputs Blocks	167
功能块	170
逻辑功能块	172
Outputs 功能块.....	173

启动器类型

概述

FBD 编辑器允许使用可通过 Toolbox 中的 **Starter Types** 栏访问的启动器功能块：

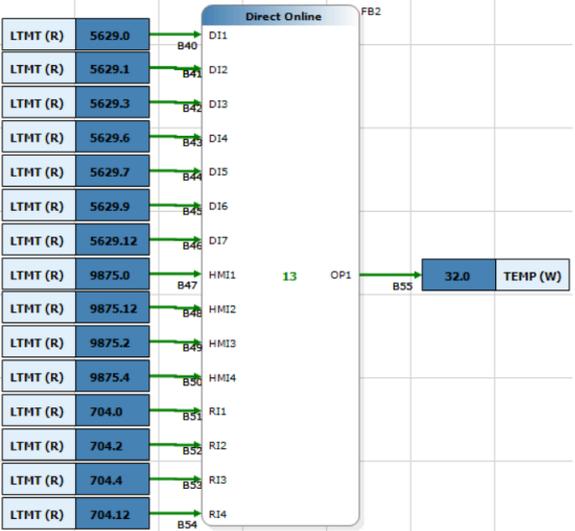
功能块	描述
	Direct Online
	Reversible Direct Online
	Star Delta
	Overload

注: 将光标放在图标上方将会出现定义该图标的工具提示。这将帮助您识别该图标所表示的功能块的类型。

注: 一次只能使用一种类型的启动器。

Direct Online

使用  功能块将启动器类型设置为 Direct Online。

FBD 显示	参数	描述
	数字量输入信号	<ul style="list-style-type: none"> DI1 (5629.0) : Local-START> DI DI2 (5629.1) : Local-STOP DI DI3 (5629.3) : DI 模式选择 1 DI4 (5629.6) : Remote-START> DI DI5 (5629.7) : Remote-STOP DI DI6 (5629.9) : DI 模式选择 2 DI7 (5629.12) : 运行 DI
	HMI 信号	<ul style="list-style-type: none"> HMI1 (9875.0) : HMI_START HMI2 (9875.12) : HMI_STOP HMI3 (9875.2) : HMI 模式选择 1 HMI4 (9875.4) : HMI 模式选择 2
	通讯信号	<ul style="list-style-type: none"> RI1(704.0) : COMM Start RI2 (704.2) : 远程模式选择 1 RI3 (704.4) : 远程模式选择 2 RI4 (704.12) : COMM Stop
	输出信号	<ul style="list-style-type: none"> OP1 (32.0) : CL Run1 Cde information

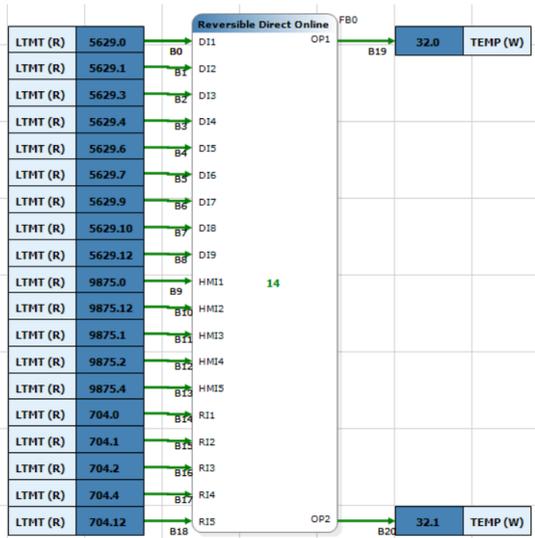
过载

使用  功能块将启动器类型设置为 Overload。

FBD 显示	参数	描述
	OP1	输出信号 1

Reversible Direct Online

使用  功能块将启动器类型设置为 Reversible Direct Online。

FBD 显示	参数	描述
	数字量输入信号	<ul style="list-style-type: none"> DI1 (5629.0) : Local-START> DI DI2 (5629.1) : Local-STOP DI DI3 (5629.3) : DI 模式选择 1 DI4 (5629.4) : Local-START< DI DI5 (5629.6) : Remote-START> DI DI6 (5629.7) : Remote-STOP DI DI7 (5629.9) : DI 模式选择 2 DI8 (5629.10) : Remote-START< DI DI9 (5629.12) : 运行 DI
	HMI 信号	<ul style="list-style-type: none"> HMI1 (9875.0) : HMI_START > HMI2 (9875.12) : HMI_STOP HMI3 (9875.1) : HMI_START < HMI4 (9875.2) : HMI 模式选择 1 HMI5 (9875.4) : HMI 模式选择 2
	通讯信号	<ul style="list-style-type: none"> R11(704.0) : COMM Start > R12 (704.1) : COMM Start < R13 (704.2) : 远程模式选择 1 R14 (704.4) : 远程模式选择 2 R15 (704.12) : COMM Stop
	输出信号	<ul style="list-style-type: none"> OP1 (32.0) : CL Run1 Cde information OP2 (32.1) : CL Run2 Cde information

Star Delta

使用  功能块将启动器类型设置为 Star Delta。

FBD 显示	参数	描述
	数字量输入信号	<ul style="list-style-type: none"> DI1 (5629.0) : Local-START> DI DI2 (5629.1) : Local-STOP DI DI3 (5629.3) : DI 模式选择 1 DI4 (5629.6) : Remote-START> DI DI5 (5629.7) : Remote-STOP DI DI6 (5629.9) : DI 模式选择 2 DI7 (5629.12) : 运行 DI
	HMI 信号	<ul style="list-style-type: none"> HMI1 (9875.0) : HMI_START > HMI2 (9875.12) : HMI_STOP HMI3 (9875.2) : HMI 模式选择 1 HMI4 (9875.4) : HMI 模式选择 2
	通讯信号	<ul style="list-style-type: none"> RI1(704.0) : COMM Start > RI2 (704.2) : 远程模式选择 1 RI3 (704.4) : 远程模式选择 2 RI4 (704.12) : COMM Stop
	输出信号	<ul style="list-style-type: none"> OP1 (32.0) : CL Run1 Cde information OP2 (32.1) : CL Run2 Cde information OP3 (32.2) : CL Run3 Cde information

计算功能块

概述

FBD 编辑器使用可通过 Toolbox 中的 **Computation** 栏访问的以下各种计算功能块：

功能块	描述
	Compare
	Add
	Division
	Multiplication
	Subtraction

注: 将光标放在图标上方将会出现定义该图标的工具提示。这将帮助您识别该图标所表示的功能块的类型。

Compare 功能块



功能块比较两个 16 位寄存器值。

FBD 符号	参数	描述
	Inputs	<ul style="list-style-type: none"> X : 16 位无符号寄存器值 (0 至 65,535) 。 Y : 16 位无符号寄存器值 (0 至 65,535) 。
	Outputs	<ul style="list-style-type: none"> X < Y : 如果值 X 小于值 Y , 则 ON/OFF 临时位的值为 ON。 X = Y : 如果值 X 等于值 Y , 则 ON/OFF 临时位的值为 ON。 X > Y : 如果值 X 大于值 Y , 则 ON/OFF 临时位的值为 ON。

Add 功能块



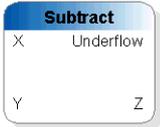
功能块执行两个 16 位寄存器值的无符号加法。

FBD 符号	参数或示例	描述
	Inputs	<ul style="list-style-type: none"> X : 16 位无符号寄存器值 (0 至 65,535) 。 Y : 16 位无符号寄存器值 (0 至 65,535) 。
	Outputs	<ul style="list-style-type: none"> Z : 16 位无符号寄存器结果 (Z = X + Y) 。 Overflow : 值为 ON 或 OFF , 设置为 ON 表示值为 65,536。此值初始设置为 OFF。
	Example	假设 X = 60,000 和 Y = 7,000 , 则上溢值为 ON , 因为 60,000 + 7,000 = 67,000 , 超过了 65,536。结果 Z 即等于 1,464 (1,464 + 65,356 = 67,000) 。

Subtraction 功能块



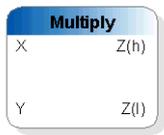
功能块执行两个 16 位寄存器值的无符号减法。

FBD 符号	参数或示例	描述
	Inputs	<ul style="list-style-type: none"> X : 16 位无符号寄存器值 (0 至 65,535) 。 Y : 16 位无符号寄存器值 (0 至 65,535) 。
	Outputs	<ul style="list-style-type: none"> Z : 16 位无符号寄存器结果 (Z = X - Y) 。 Underflow : 值为 ON 或 OFF , 设置为 ON 表示值为负 65,536。此值初始设置为 OFF。
	Example	假设 X = 5 且 Y = 10 , 下溢值为 ON , 因为结果为负值。结果 Z 即等于 65,531 (65,531 - 65,536 = -5)

Multiplication 功能块



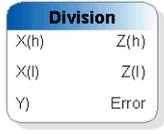
功能块执行两个 16 位寄存器值的无符号乘法。

FBD 符号	参数或示例	描述
 <p>The Multiply symbol shows two input ports labeled X and Y, and two output ports labeled Z(h) and Z(l).</p>	Inputs	<ul style="list-style-type: none"> X : 16 位无符号寄存器值 (0 至 65,535) Y : 16 位无符号寄存器值 (0 至 65,535)
	Outputs	<ul style="list-style-type: none"> Z(h) : 32 位结果的 16 个最高有效位 $Z(h) = (X * Y) / 65,536$ Z(l) : 32 位结果的 16 个最低有效位 $Z(l) = (X * Y) - Z(h) * 65,536$
	Example	假设 X = 20,000 且 Y = 10 , 结果将为 Z(h) = 3 , Z(l) = 3,392 , 因为 $200,000 = 3 * 65,536 + 3,392$

Division 功能块



功能块执行两个 16 位寄存器值的无符号除法。

FBD 符号	参数或示例	描述
 <p>The Division symbol shows two input ports labeled X(h) and X(l), and one input port labeled Y. It has two output ports labeled Z(h) and Z(l), and an output port labeled Error.</p>	Inputs	<ul style="list-style-type: none"> X(h) : 无符号寄存器值的 16 个最高有效位 (0 至 65,535) 。 X(l) : 无符号寄存器值的 16 个最低有效位 (0 至 65,535) 。 Y : 16 位无符号寄存器除数 (0 至 65,535) 。
	Outputs	<ul style="list-style-type: none"> Z(h) : 32 位商的 16 个最高有效位 $Z(h) = (X/Y) / 65,536$ Z(l) : 32 位商的 16 个最低有效位 $Z(l) = (X/Y) - Z(h) * 65,536$ 检测到错误 : 值为 ON 或 OFF , 当出现被零除时 , 设置为 ON。此值初始设置为 OFF。
	Example	假设 X(h) = 3 , X(l) = 3,392 且 Y = 40 , 则结果为 Z(h) = 0 和 Z(l) = 5,000 , 因为 $X(h) * 65,536 + X(l) = 3 * 65,536 + 3,392 = 200,000$ 且 $200,000 / Y = 5,000 = 0 * 65,536 + 5,000$

Inputs Blocks

概述

FBD 编辑器使用可通过 Toolbox 中的 **Inputs** 栏访问的以下各种输入功能块：

功能块	描述
	Constant Bit
	Constant Word
	Register Bit In
	Register Word In
	Register NV Bit In
	Register NV Word In
	Register Temp Bit In
	Register Temp Word In

注：将光标放在图标上方将会出现定义该图标的工具提示。这将帮助您识别该图标所表示的功能块的类型。

Constant Bit 功能块



功能块用于将其他模块的输入设置为0或1。

FBD 符号	参数	描述
	Properties	• 常量位值 0 或 1 (True = 1 且 False = 0)。
	Outputs	• 常量值 0 或 1 (True = 1 且 False = 0)。

Constant Word 功能块



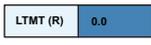
功能块用于设置其他模块的输入值 (范围为0至65,535)。

FBD 符号	参数	描述
	Properties	• 0 至 65,535 的常量值。
	Outputs	• 0 至 65,535 的常量值。

Register Bit In 功能块



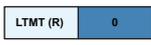
功能块让您能够从 LTMT main unit 寄存器读取寄存器位值并使用该值。

FBD 符号	参数	描述
	Properties	• 任何寄存器地址 (0 至 9249 (0x0000 至 0x2421) 、 9875 (0x2693)、 9876 (0x2694)) x 位编号 (0 至 15) 。
	Outputs	• 值 0 或 1 (True = 1 且 False = 0) 。

Register Word In 功能块



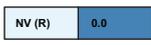
功能块让您能够从 LTMT main unit 寄存器读取寄存器值并使用该值。

FBD 符号	参数	描述
	Properties	• 任何寄存器地址 , 0 至 9249 (0x0000 至 0x2421) 、 9875 (0x2693)、 9876 (0x2694) 。
	Outputs	• 0 至 65,535 的值。

Register NV Bit 功能块



功能块让您能够读取和使用非易失性寄存器位值。

FBD 符号	参数	描述
	Properties	• 任何非易失性寄存器 0 至 63 x 位编号 (0 至 15) 。
	Outputs	• 值 0 或 1 (True = 1 且 False = 0) 。

Register NV Word In 功能块



功能块让您能够读取和使用非易失性寄存器值。

FBD 符号	参数	描述
	Properties	• 0 至 63 的任何非易失性寄存器。
	Outputs	• 0 至 65,535 的值。

Register Temp Bit In 功能块



功能块让您能够读取和使用临时寄存器位值。

FBD 符号	参数	描述
	Properties	<ul style="list-style-type: none"> 任何临时寄存器 0 至 299 x 位编号 (0 至 15)。 .
	Outputs	<ul style="list-style-type: none"> 值 0 或 1 (True = 1 且 False = 0)。

Temp Word In 功能块



功能块让您能够读取和使用临时寄存器值。

FBD 符号	参数	描述
	Properties	<ul style="list-style-type: none"> 0 至 299 的任何临时寄存器。
	Outputs	<ul style="list-style-type: none"> 0 至 65,535 的值。

功能块

概述

FBD 编辑器使用可通过 Toolbox 中的 **Function** 栏访问的以下各种功能块：

功能块	描述
	Counter
	Counter NV
	Volatile Latch
	Non Volatile Latch
	Multiplexer
	TimerSeconds
	TimerTenthSeconds

注：将光标放在图标上方将会出现定义该图标的工具提示。这将帮助您识别该图标所表示的功能块的类型。

Counter 功能块



功能执行比较计数，将计数器的当前值和预设值都保存到临时寄存器。

FBD 符号	参数	描述
	属性	K : 计数器预设值 (UINT 0 至 65,535) 。
	Inputs	<ul style="list-style-type: none"> Inc : 在上升沿上增加计数器当前值。计数器当前值应介于 0 至 65,535 之间。 递减 : 在下降沿上递减计数器当前值。计数器当前值应介于 65,535 至 0 之间。 设置 : 在上升沿将当前计数器值设置为预设值。
	Outputs	<ul style="list-style-type: none"> 计数 : 计数器当前值 (UINT 0 至 65,535) 。此计数值在加电时初始化为零。 <K : 计数器当前值小于预设值 K。 =K : 计数器当前值等于预设值 K。 >K : 计数器当前值大于预设值 K。

注：计数器预设值的范围为 0 至 65,535。如果您需要更大的值或多个预设值，可以使用级联计数器和比较功能。

Counter NV 功能块



功能执行比较计数，将计数器的当前值和预设值都保存到非易失性寄存器。

FBD 符号	参数	描述
	属性	K : 计数器预设值 (UINT 0 至 65,535)。
	Inputs	<ul style="list-style-type: none"> Inc : 在上升沿上增加计数器当前值。计数器当前值应介于 0 至 65,535 之间。 递减 : 在下降沿上递减计数器当前值。计数器当前值应介于 65,535 至 0 之间。 设置 : 在上升沿将当前计数器值设置为预设值。
	Outputs	<ul style="list-style-type: none"> 计数 : 计数器当前值 (UINT 0 至 65,535)。此值保存在非易失性存储器中并在加电时初始化为上一值。 <K : 计数器当前值小于预设值 K。 =K : 计数器当前值等于预设值 K。 >K : 计数器当前值大于预设值 K。

注: 计数器预设值的范围为 0 至 65,535。如果您需要更大的值或多个预设值，可以使用级联计数器和比较功能

Volatile Latch 功能块



功能记录信号历史并将其保留在临时寄存器中。

FBD 符号	参数	描述
	Inputs	<ul style="list-style-type: none"> 设置 : ON/OFF 输入值。当此输入从 OFF 转换为 ON 时，锁存值设置为 ON。 清除 : ON/OFF 输入值。当此输入从 OFF 转换为 ON 时，锁存值设置为 OFF。
	Outputs	<ul style="list-style-type: none"> Q : 表示此锁存的状态的 ON 或 OFF 锁存值。此值保持为 ON/OFF 直到设置或清除的下一上升沿。此值初始设置为 OFF。

Non Volatile Latch 功能块



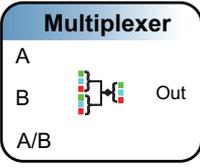
功能记录信号历史并将其保留在非易失性寄存器中。

FBD 符号	参数	描述
	Inputs	<ul style="list-style-type: none"> 设置 : ON/OFF 输入值。当此输入从 OFF 转换为 ON 时，锁存值设置为 ON。 清除 : ON/OFF 输入值。当此输入从 OFF 转换为 ON 时，锁存值设置为 OFF。
	Outputs	<ul style="list-style-type: none"> Q : 表示此锁存的状态的 ON 或 OFF 非易失性寄存器位值。此值保持为 ON/OFF 直到设置或清除的下一上升沿。此值保存在非易失性存储器中并在加电时初始化为上一状态。

Multiplexer 功能块



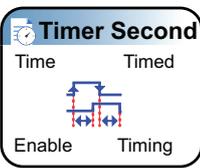
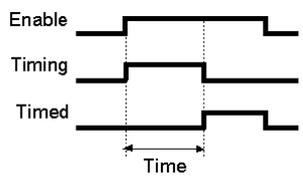
功能使您能够在两个 16 位无符号值之间进行选择。

FBD 符号	参数	描述
	Inputs	<ul style="list-style-type: none"> A : 16 位无符号值 (0 至 65,535)。 B : 16 位无符号值 (0 至 65,535)。 A/B : 选择值 A 或 B 的 ON/OFF 输入值。
	Outputs	<ul style="list-style-type: none"> 输出 : 选定的 16 位值 : <ul style="list-style-type: none"> 如果 A/B 为 ON, 则输出 = A。 如果 A/B 为 OFF, 则输出 = B。

Timer Seconds 功能块



功能以秒为间隔测量时间。

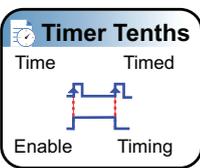
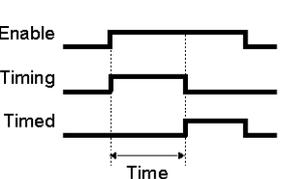
FBD 符号	时序图	参数个数	描述
		Inputs	<ul style="list-style-type: none"> Time : 以秒指定时间周期的 16 位无符号值 (0 至 65,535)。 Enable : ON/OFF 输入值。在 Enable 输入的上升沿加载时段。当 Enable 为 ON 时, 继续测量时间。当 Enable 为 OFF 时停止定时, 输出为 OFF。
		Outputs	<ul style="list-style-type: none"> Timed - 在经过一段时间后变为 ON 的 ON/OFF 值。当 Enable 为 OFF 时, 该值为 OFF Timing - ON/OFF 值, 当 Enable 为 ON 且在测量时间内, 该值为 ON。测量时间结束后, 该值为 OFF。 <p>注: 两个输出不会同时为 ON。</p>

Timer Tenths Seconds 功能块



功能以十分之几秒为间隔来计量时间。

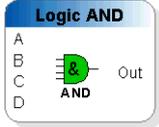
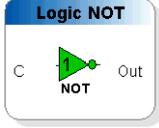
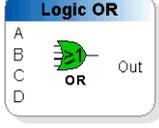
注: TIMER_TENTHS 命令的乘数为 10。例如, 如果所需值为 6 秒, 则输入值应乘以系数 10, 表示提供的输入值应为 60 秒。

FBD 符号	时序图	参数个数	描述
		Inputs	<ul style="list-style-type: none"> Time : 以十分之几秒指定时间周期的 16 位无符号值 (0 至 65,535)。 Enable : ON/OFF 输入值。在 Enable 输入的上升沿加载时段。当 Enable 为 ON 时, 继续测量时间。当 Enable 为 OFF 时停止定时, 输出为 OFF。
		Outputs	<ul style="list-style-type: none"> Timed - 在经过一段时间后变为 ON 的 ON/OFF 值。当 Enable 为 OFF 时, 该值为 OFF Timing - ON/OFF 值, 当 Enable 为 ON 且在测量时间内, 该值为 ON。测量时间结束后, 该值为 OFF。 <p>注: 两个输出不会同时为 ON。</p>

逻辑功能块

概述

FBD 编辑器使用可通过 Toolbox 中的 **Logic blocks** 栏访问的各种逻辑功能块 :

功能	图标	FBD 符号	描述
AND			如果所有输入 (ON 或 OFF 值, 分别代表 1 或 0) 为 ON, 则输出为 ON。 如果至少一个输入为 OFF, 则输出为 OFF。 注: 未连接的输入将假定为 ON。
NOT			如果输入 (ON 或 OFF 值, 分别代表 1 或 0) 为 ON, 则输出为 OFF。 如果输入为 OFF, 则输出为 ON。
OR			如果至少一个输入 (ON 或 OFF 值, 分别代表 1 或 0) 为 ON, 则输出为 ON。 如果所有输入都为 OFF, 则输出为 OFF。 注: 未连接的输入将假定为 OFF。

注: 将光标放在图标上方将会出现定义该图标的工具提示。这将帮助您识别该图标所表示的功能块的类型。

Outputs 功能块

概述

FBD 编辑器使用可通过 Toolbox 中的 **Outputs** 栏访问的以下各种输出功能块：

功能块	描述
	Register Bit Out
	Register Word Out
	Register NV Bit Out
	Register NV Word Out
	Register Temp Bit Out
	Temp Word Out

注: 将光标放在图标上方将会出现定义该图标的工具提示。这将帮助您识别该图标所表示的功能块的类型。

Register Bit Out 功能块



功能块用于将 LTMT main unit 寄存器的 LTMT main unit 寄存器位值设置为 0 或 1。

FBD 符号	参数	描述
	Properties	<ul style="list-style-type: none"> a : 任何寄存器地址, 0 至 9249 (0x0000 至 0x2421)、9875 (0x2693)、9876 (0x2694)。 b : 0 至 15 的位编号。
	Inputs	<ul style="list-style-type: none"> 0 或 1 (ON=1, OFF=0)。

Register Word Out 功能块



功能块用于设置 LTMT main unit 寄存器的 LTMT main unit 寄存器位值。

FBD 符号	参数	描述
	Properties	<ul style="list-style-type: none"> a : 任何寄存器地址, 0 至 9249 (0x0000 至 0x2421)、9875 (0x2693)、9876 (0x2694)。
	Inputs	<ul style="list-style-type: none"> 0 至 65,535 的 16 位无符号值。

Register NV Bit Out 功能块



功能块用于将非易失性寄存器位值设置为 0 或 1。

FBD 符号	参数	描述
	Properties	<ul style="list-style-type: none"> a : 0 至 63 的任何非易失性寄存器。 b : 0 至 15 的位编号。
	Inputs	<ul style="list-style-type: none"> 0 或 1 (ON=1, OFF=0)。

Register NV Word Out 功能块



功能块用于设置非易失性寄存器值。

FBD 符号	参数	描述
	Properties	<ul style="list-style-type: none"> a : 0 至 63 的任何非易失性寄存器。
	Inputs	<ul style="list-style-type: none"> 0 至 65,535 的 16 位无符号值

Register Temp Bit Out 功能块



功能块用于将临时寄存器位值设置为 0 或 1。

FBD 符号	参数	描述
	Properties	<ul style="list-style-type: none"> a : 0 至 299 的任何临时寄存器。 b : 0 至 15 的位编号。
	Inputs	<ul style="list-style-type: none"> 0 或 1 (ON=1 , OFF=0) 。

Temp Word Out 功能块



功能块用于设置临时寄存器值。

FBD 符号	参数	描述
	属性	<ul style="list-style-type: none"> a : 0 至 299 的任何临时寄存器。
	Inputs	<ul style="list-style-type: none"> 0 至 65,535 的 16 位无符号值

使用 FBD 语言进行编程

此章节内容

插入 FBD 功能块	177
在功能块之间创建链接	178
FBD 功能块属性	180
FBD 资源管理	181

插入 FBD 功能块

概述

如要创建 FBD 程序，将功能块插入到工作区中，然后将它们链接到一起。所有类型的功能块都可放到工作区中。

从工具箱插入功能块

以下过程介绍如何将功能块从工具箱插入到工作区中：

1. 选择 **Device > FB Diagram > View > Toolbox**，或者选择左侧的 **Toolbox** 选项卡。
2. 选择要插入的功能块类型：
 - Computation
 - Inputs
 - Function Blocks
 - Logic
 - Starter Types
 - Outputs
3. 左键单击与要插入的功能块相对应的图标。
4. 从工具箱将功能块拖放到工作区。
5. 将功能块放到工作区上的所需位置。
6. 重复步骤 2 至 5 以插入程序所需的所有功能块。

从工作区插入功能块

以下过程介绍如何直接从工作区插入功能块：

1. 在工作区空白处的任何位置，执行右键单击。
结果：显示一个菜单，让您能够选择要插入的功能块类型。
2. 选择要插入的功能块类型：
 - Computation
 - Inputs
 - Function Blocks
 - Logic
 - Starter Types
 - Outputs
3. 左键单击要插入的功能块。
4. 将功能块放到工作区中的所需位置。
5. 重复步骤 1 至 5 以插入程序所需的所有功能块。

在功能块之间创建链接

概述

将功能块放入工作区后，即可将它们链接到一起。为此，请将一个功能块的输出链接到另一个块的输入。您还可以将一个输出回链到相同功能块的输入。

一般规则

放置和连接功能块时，需遵循一些基本规则：

- 连接在一起的一个或多个连接线形成线节点。这在工作区中通过红点来表示。如果若干条线交叉但没有红色的连接点，则表示它们未连接。
- 每个线节只能连接一个输出。
- 禁止连接布尔值和寄存器数据。
- 数据一般从左至右流动。

功能块之间的链接

以下步骤介绍如何将功能块链接在一起：

1. 将鼠标放在第一个功能块的上方。

结果：块边框上将出现一个或多个方块，并指示出输出类型（模拟或布尔）。



2. 单击左鼠标按钮并按住。
3. 保持按住该按钮，将鼠标移至要链接到的块的输入处。

结果：功能块边框上将出现一个或多个方块。如果方块为绿色，则可连接两个功能块。红色方块表示不能进行连接。同时还指示出输出类型（模拟量或布尔）。



注：输入和输出的类型必须相同：布尔输出链接到另一个布尔输出。如果输入或输出不同，FBD 编辑器将显示出一个弹出窗口以指示来源和目标类型不同。

4. 松开鼠标按钮。

结果：两个链接的功能块之间出现一条线和一个数字。

5. 重复步骤 1 至 4，以链接所有功能块。

链接编号

有两种类型的接线：

- 布尔线，编号以 B 开头。
- 寄存器线，编号以 R 开头。

线编号按时间顺序自动递增。

FBD 功能块属性

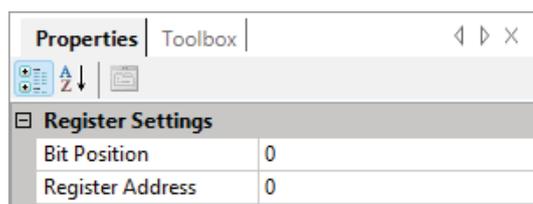
概述

每个块都有一个属性窗口。要显示此窗口，左键单击一个块。

Properties 窗口包含多个选项卡，取决于功能块类型，这些选项卡分为1或2个类别：

- 常规设置，包含块 ID 和注释（对所有类型的块通用。）
- 具体设置项，取决于功能块的类型（例如：寄存器对应寄存器设置，计数器对应计数器设置，等等）。

例如，如果要显示非易失性寄存器的属性，则选择非易失性寄存器功能块并用左键单击它。将显示以下窗口：



注释

在 Comment 区域中，在注释右侧的白框中，可以输入注释。在工作区中选择任何对象或任何空闲位置以保存注释。

设置

大多数块都具有特定的设置选项卡。在此选项卡中，设置功能块的特定设置。这些设置将在每个 FBD 块的帮助中详细介绍。

属性显示

每个功能块的属性都可以通过两种不同的方式来显示：

- 按类别，单击 ，或者
- 按字母顺序，单击 .

FBD 资源管理

概述

LTMT main unit 内存配备有下列资源：

- 9250 个 LTMT 寄存器
- 相当于 8,192 字 (16 位) 的逻辑内存空间大小
- 300 个临时寄存器
- 64 个非易失性寄存器

预留资源

下表列出了所有预留的寄存器及其分配。它还指示出如何控制这些寄存器：

寄存器类型	地址范围	控制者	描述
临时	0-69	用户	临时存储在创建 FBD 程序时分配的位和寄存器。
临时	70-299	FBD 编译器	由编译器使用的预留临时寄存器。
非易失性	0-31	用户	创建 FBD 程序时分配的非易失性位或寄存器。
非易失性	32-63	FBD 编译器	由编译器使用的预留非易失性寄存器。

操作 FBD 功能块

此章节内容

选择功能块	183
删除或复制对象	183

选择功能块

概述

将功能块添加到工作区时，即可选择它们以在工作区中对它们进行重新定位。

选择一个或多个功能块

下表介绍了如何选择一个或多个功能块：

如果您要选择...	则
隔离块	单击块。
多个连续块	通过定义选择区来框住要选择的块。 结果： 所有选定块都由橙色框线高亮显示。 
工作区不同区域中的多个功能块	按下 Shift 键，然后在按住 Shift 键的同时单击要选择的块。 结果： 所有选定块都由橙色框线高亮显示。
包含线的所有对象	选择 Device > FB Diagram > FBD Editor > Select All 注： 也可使用键盘快捷键 CTRL+A 来选择所有对象。

删除或复制对象

概述

有时，可能需要在工作区中删除或复制功能块。

删除功能块

以下步骤介绍如何删除一个或多个功能块：

1. 选择要删除的功能块。

结果：所有选定功能块都由橙色框线突出显示。



2. 按 DELETE 键，或者选择 **Device > FB Diagram > FBD Editor > Delete**。

结果：选定功能块被删除。

1. 选择要删除的功能块。

结果：所有选定功能块都由橙色框线突出显示。



2. 右键单击所选功能块，然后选择 **Delete**。

结果：选定功能块被删除。

剪切、复制或粘贴功能块

以下步骤介绍如何剪切、复制或粘贴一个或多个功能块：

1. 选择要修改的功能块。

结果：所有选定功能块都由橙色框线突出显示。



2. 选择 **Device > FB diagram > FBD editor**，然后选择以下其中一个命令：

- **Copy**
- **Cut**
- **Paste**

结果：**Cut** 删除所选功能块并将它们存储在剪贴板中。**Copy** 将所选功能块复制到剪贴板中，**Copy** 在工作区上复制剪贴板内容。

注：键盘快捷键 **CTRL+C** 和 **CTRL+V** 也可用于复制粘贴所选功能块。

隐藏或显示功能块

以下步骤介绍如何隐藏一个或多个功能块：

1. 选择要隐藏的功能块。

结果：所有选定功能块都由橙色框线突出显示。

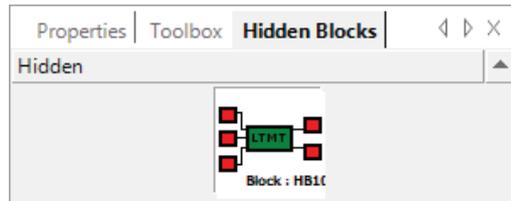


2. 右键单击所选功能块，然后选择 **Hide/Show**。

结果：选定功能块被隐藏并按如下方式显示：



3. 隐藏功能块的列表显示在 **Hidden Blocks** 下方，如下所示：



注：选择了多个功能块时，所有功能块都分组在同一功能块中。

以下步骤介绍如何显示一个或多个已隐藏的功能块：

1. 右键单击已隐藏的功能块，然后选择 **Hide/Show**。

结果：选定功能块被显示。

2. 或者，右键单击 **Hidden Blocks** 部分下方的已隐藏的功能块，然后选择 **Hide/Show**。

结果：选定功能块被显示。

FBD 编辑器显示选项

此章节内容

显示选项.....	187
工作区外观和图形选项.....	188

显示选项

概述

您可自定义以下显示选项以适合自己的需求：

- 缩放
- 链接
- 参数

缩放显示选项

如要访问缩放选项，请选择 **Device > FB Diagram > View**。

提供了以下三种选项：

- 缩小，用于立即查看程序的更多信息。
- 放大，用于重点查看程序的更多细节。
- 缩放至 50%、75%、100%、150%、200% 或 400%，用于自定义程序视图。

链接显示选项

如要访问链接显示选项，请选择 **Device > FB Diagram > Tools**。

提供了三种选项。您可以：

- 对链接重新编号以帮助了解程序执行过程。
- 显示所有链接以查看哪些块链接在一起。
- 隐藏所有链接以获得更好的功能块总体视图。

单击某个链接时，将显示其属性窗口，让您能够自定义出现在该链接旁边的文本。

参数显示选项

以下过程描述了如何访问和更改参数显示选项：

1. 将鼠标放在功能块上方。

结果：功能块边框上将出现一个或多个方块。它还指示出参数是模拟型还是布尔型。



2. 单击此方块。

结果：将出现显示选项。

3. 选择您是否要显示标签以及应出现哪些内容。

工作区外观和图形选项

概述

使用 FBD 编辑器，您可以通过更改其外观和图形选项来自定义工作区。

外观和图形选项

要访问外观和图形选项，左键单击工作区中除对象外的任意位置。

外观选项

下表列出了所有可能的外观自定义选项：

外观选项	描述	可能的选择
Background Color	通过单击显示所需颜色的框来设置工作区的背景色。	在 Custom 、 Web 和 System 选项卡中，在可用的颜色间进行选择。
Background Type	可设置背景类型。	在单色、渐变色或图像背景之间选择。
Enable Context Menu	显示或隐藏上下文菜单。	True 或 False
Enable Tooltip	显示或隐藏工具提示。	True 或 False
Restrict to Canvas	可选择是否应将 FBD 程序保持在画布内。	True 或 False
Show Grid	可选择是否显示准确网格。	True 或 False 注： 不得将此网格与网格线混淆，后者可从顶层视图菜单栏访问。

图形选项

下表列出了所有可能的图形自定义选项：

图形选项	描述	可能的选择
Allow Add Connection	可选择是否能在工作区中添加连接。	True 或 False
Allow Add Shape	可选择是否能在工作区中添加功能块。	True 或 False
Allow Delete Shape	可选择是否能够删除块。	True 或 False
Allow Move Shape	可选择是否能在工作区中移动功能块。	True 或 False
Locked	可选择是否能够编辑 FBD 程序。	True 或 False

Display Gridlines

您可能希望显示网格线。为此，请选择 **Device > FB Diagram > View > Show Gridlines**。

编译、仿真和传输程序

此部分内容

简介	190
LTMT Main Unit 逻辑仿真器	191
初始化和连接	193
在LTMT Main Unit 与 Custom Logic Editor 之间传输逻辑文件	194

简介

编译概述

在将自定义程序下载到 LTMT main unit 之前，必须对其进行编译：

- 使用自定义逻辑语言编写的程序可以直接编译。
- 在编译为自定义逻辑程序之前，必须先将 FBD 语言的程序转换为自定义逻辑语言程序。

编译包括检查程序错误，比如：

- 语法和结构错误
- 无相应地址的符号
- 程序所使用的、但当前不可用的资源
- 程序是否适合可用的 LTMT main unit 内存

将 FB 图转换为自定义逻辑

选择 **Device > FB Diagram > FB Diagram to Custom Logic**，将已创建或编辑的 FBD 编译成自定义逻辑程序。

如果没有检测到错误，程序将自动被复制到 **Custom Logic Editor** 中。

注：在转换 FBD 程序之前，请记住在 FBD 编辑器中保存它，因为无法将自定义逻辑程序文件转换为 FBD 程序文件。

编译自定义逻辑

按照以下步骤将刚创建的自定义逻辑程序编译为 PCode：

1. 选择 **Device > Custom Logic**。
2. 选择 **Compile Custom Logic**。

注：如果未检测到错误，则将显示出 PCode 窗口。否则，将显示 **Detected Error** 窗口。

LTMT Main Unit 逻辑仿真器

概述

带 TeSys Tera DTM Library 的 SoMove 软件随附逻辑仿真器。使用它，在将以自定义逻辑语言创建的自定义程序传送到 LTMT main unit 之前，可以测试该自定义程序的功能。

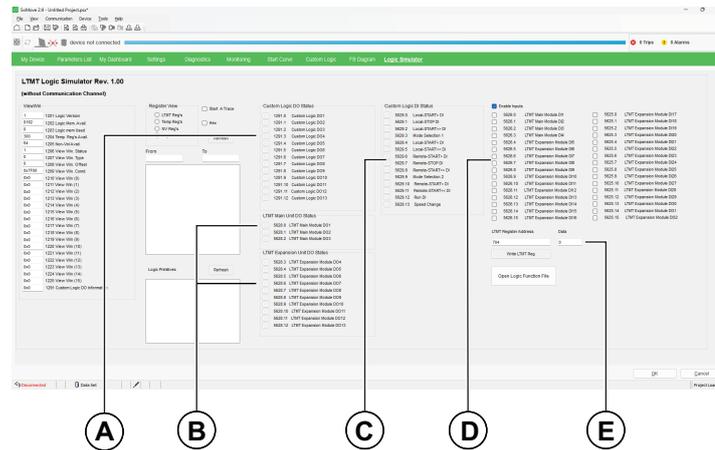
注: 如要仿真 FBD 程序，必须先将其进行转换并保存为扩展名为 *.lf 的自定义逻辑程序。

逻辑仿真器界面

如要打开逻辑仿真器，请选择 **Logic Simulator** 选项卡。然后便会显示逻辑仿真器窗口。在右下角，选择 **Open Logic Function File** 以导入已保存的自定义逻辑程序。

选择自定义程序文件，然后选择 **Open**。

逻辑仿真器 现已加载自定义程序。自定义程序可以在仿真器中仿真，如下所示。



- A 更新自定义逻辑 DO 状态。
- B 更新 LTMT 模块 DO 状态。
- C 更新自定义逻辑 DI 状态。
- D 用于仿真 LTMT 模块 DI 数据。
- E 用于仿真 LTMT 寄存器地址。

使用逻辑仿真器

如要使用逻辑仿真器来仿真自定义逻辑文件，请执行以下步骤：

- 选择 **Logic Simulator** 选项卡。
- 选择 **Open Logic Function File**，并从 PC 中选择已保存的自定义逻辑文件。
- 在 **Register View** 下方，选择寄存器类型，然后单击 **Refresh** 以加载寄存器列表。
- 根据需要，分别修改 **LTMT Register Address** 和 **Data** 字段下方的寄存器和数据，然后单击 **Write LTMT Reg.**

- 选择 **Refresh**，以使用提供的最新值更新寄存器列表。

注: 如要使用提供的新值更新寄存器列表，请单击 **Refresh**。

注: 将自定义逻辑文件上传到仿真器并启用了定时器后，它将按照预期运行。但是，一旦启用并执行了定时器，便无法再次启用该定时器。这是 Timer 和 Tenth Timer 的限制。如要再次启用定时器，应重新上传自定义逻辑文件。在设计和测试使用定时器的自定义逻辑文件时，建议考虑此限制。

初始化和连接

初始化

在将 LTMT main unit 连接到 PC 时，控制器自动初始化。该控制器和 PC 将在初始化过程中交换标识信息。

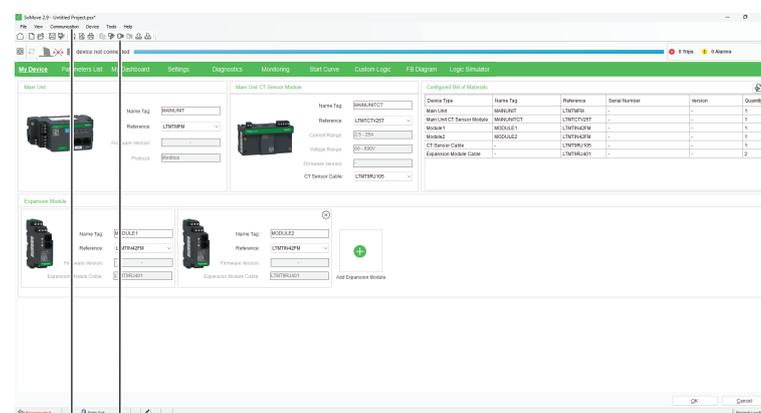
在此过程中，Custom Logic Editor 将指示等待初始化完成。

连接

初始化后，LTMT main unit 应自动连接到 PC。

如要确认是否已连接该控制器，请查看 Custom Logic Editor 中的状态栏。

如果状态栏显示为 **Disconnected**，则选择 **Communication > Connect to Device**，或者选择 **Connect to Device** 图标。



- A 连接状态
- B Communication 菜单
- C Connect to Device 图标

当 PC 连接到控制器时，会短暂地显示一个进度条，当连接过程成功完成时，状态栏中将会显示文字 **Connected**。

在已连接 LTMT main unit 时，您可以：

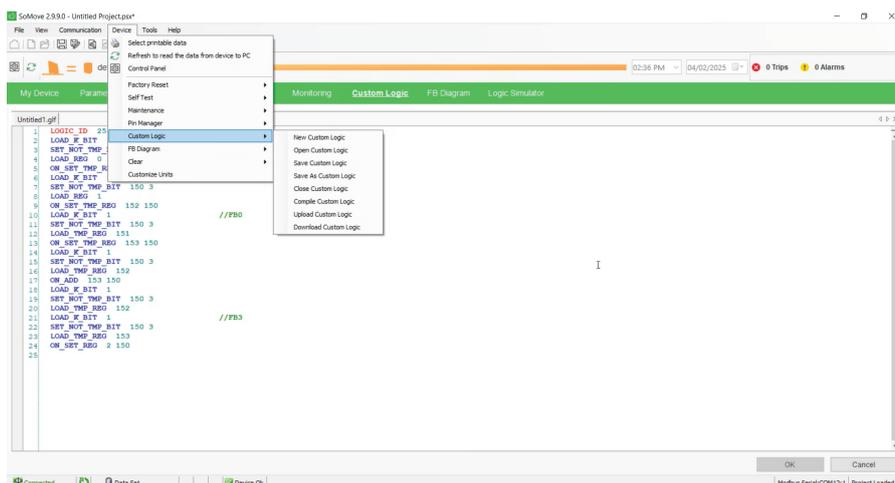
- 将自定义程序文件从控制器上传到带有 TeSys Tera DTM Library 的 SoMove 以供编辑。
- 将自定义程序文件从带有 TeSys Tera DTM Library 的 SoMove 下载到控制器。

在LTMT Main Unit 与 Custom Logic Editor 之间传输逻辑文件

使用设备自定义逻辑将文件传输到 PC

将程序文件从 LTMT main unit 传输到 Custom Logic Editor :

1. 确保 LTMT main unit 已连接到 PC。
2. 选择 **Device > Custom Logic > Upload Custom Logic**，以将程序文件从 LTMT main unit 传输到 Custom Logic Editor。



3. 传输了自定义程序文件之后，可以使用 Custom Logic Editor 将其作为自定义逻辑程序来编辑。

注: 从 LTMT main unit 获取的程序仅处于自定义逻辑中，不带注释。无法从 LTMT main unit 获取 FBD 程序等程序。

4. 程序文件编辑完成后，将工作保存到文件。

选择 **Device > Custom Logic > Save Custom Logic** 或 **Custom Logic Save As Custom Logic**。

使用 PC 自定义逻辑将文件传输到设备的步骤

在编辑并编译了程序文件之后，可以将该文件传输到 LTMT main unit。在带 TeSys Tera DTM Library 的 SoMove 执行此传输之前，必须满足以下条件：

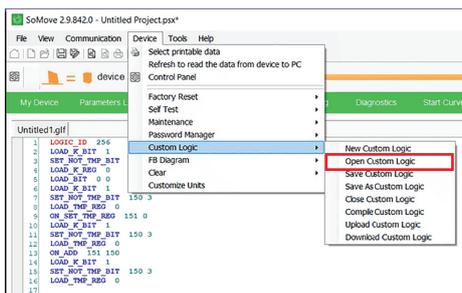
- 要传输的程序文件必须不同于 LTMT main unit 中存在的程序文件，即，软件不传输相同的程序。
- 不得检测电流，即，在线电流必须小于 FLC 的 10%。

如果未满足这些条件，则无法将该文件传送到控制器。

将程序文件从 Custom Logic Editor 传输到 LTMT main unit :

1. 确保 LTMT main unit 已连接到 PC。

2. 确保要传送的文件位于主窗口中。如要打开文件，请从 **Device** 菜单的 **Custom Logic** 子菜单中，选择 **Open Custom Logic**。然后，浏览到正确位置，并选择 **Open**。



3. 选择 **Device > 自定义逻辑 > 编译自定义逻辑**，以编译自定义逻辑程序。
4. 编译程序后，选择 **Device > Custom Logic > Download Custom Logic**，以将程序文件从 Custom Logic Editor 下载到 LTMT main unit。传输现在成功。
5. 随即打开一个新对话框，选择 **Ok** 以将其关闭。

Schneider Electric Industries SAS
35 rue Joseph Monier
92500 Rueil Malmaison
France

www.se.com

由于各种标准、规范和设计不时变更，请索取对本出版物中给出的信息的确认。

© 2025 Schneider Electric. 版权所有。

DOCA0257ZH-01