TeSys Active

TeSys Tera Motor Management System

DTM Library Online Help Guide

TeSys offers innovative and connected solutions for motor starters.

DOCA0275EN-00
03/2025

www.se.com

Legal Information

The information provided in this document contains general descriptions, technical characteristics and/or recommendations related to products/solutions.

This document is not intended as a substitute for a detailed study or operational and site-specific development or schematic plan. It is not to be used for determining suitability or reliability of the products/solutions for specific user applications. It is the duty of any such user to perform or have any professional expert of its choice (integrator, specifier or the like) perform the appropriate and comprehensive risk analysis, evaluation and testing of the products/solutions with respect to the relevant specific application or use thereof.

The Schneider Electric brand and any trademarks of Schneider Electric SE and its subsidiaries referred to in this document are the property of Schneider Electric SE or its subsidiaries. All other brands may be trademarks of their respective owner.

Schneider Electric does not grant any right or license for commercial use of the document or its content, except for a non-exclusive and personal license to consult it on an "as is" basis.

Schneider Electric reserves the right to make changes or updates with respect to or in the content of this document or the format thereof, at any time without notice.

To the extent permitted by applicable law, no responsibility or liability is assumed by Schneider Electric and its subsidiaries for any errors or omissions in the informational content of this document, as well as any non-intended use or misuse of the content thereof.

Table of Contents

Safety Information 5
About the Document 6
Precautions 9
Security Hardening Guidelines 11
Introduction 12
- TeSys Master Range 13
- TeSys Tera System 14
- Definitions 16
- Prerequisites for TeSys Tera DTM Library Installation 17
- Installing SoMove 18
- Installing TeSys Tera DTM Library 20
- Connecting the LTMT Main Unit to PC 21
- Troubleshooting 22

User Interface 23
- Launching the TeSys Tera DTM Library 24
- User Interface Description 30
- Menu Bar 31
- Tool Bar 35
- Status Bar 37
- Tab Zone 39
- My Device 42
- Parameters List 46
  - General Settings 50
  - Starter Function Setting 50
  - Protection Setting 51
  - Communication Setting 54
  - User Map 54
  - Modified Parameters 55
- My Dashboard 56
- Diagnostics 58
- Monitoring 64
- Start Curve 68
- Custom Logic 70
- FB Diagram 71
- Logic Simulator 72

User Functions 73
- Firmware Update 74
- Factory Reset 77
- Pin Management 78

Custom Logic Editor 81
- Presentation of the Custom Logic Editor 82
- Using the Custom Logic Editor 85
- Characteristics of the Custom Logic Program 87
- Definition of the Custom Logic Variables 88
- Definition of LTMT Main Unit Variables 89
- CALL_EOM Command Description 92

Custom Logic Language 100
Creating a Custom Logic Program 101
Introducing Custom Logic Editor 102
Custom Logic Editor User Interface 103
Logic Commands 106
Logic Commands 111
Program Logic Commands 112
Boolean Logic Commands 114
Register Logic Commands 123
Timer Logic Commands 132
Latch Logic Commands 135
Counter Logic Commands 137
Math Logic Commands 139
Custom Logic Program Examples 142
How to Check Timers and Multiply Commands 143
How to Create a Truth Table 144
Function Block Diagram Language 147
Overview of FBD Language 148
Introduction to FBD Editor 149
FBD Elements 151
Starter Types 152
Computation Blocks 154
Inputs Blocks 157
Function Blocks 160
Logic Blocks 163
Outputs Blocks 163
Programming with FBD Language 166
Inserting FBD Blocks 167
Creation of Links between Blocks 168
FBD Blocks Properties 170
FBD Resource Management 171
Manipulating FBD Blocks 172
Selecting Blocks 173
Deleting or Duplicating Objects 173
FBD Editor Display Options 176
Display Options 177
Workspace Appearance and Graph Options 178
Compiling, Simulating, and Transferring a Program 179
Introduction 180
LTMT Main Unit Logic Simulator 181
Initialization and Connection 183
Transferring Logic Files between the LTMT Main Unit and Custom Logic Editor 184

Safety Information

Important Information

Read these instructions carefully, and look at the equipment to become familiar with the device before trying to install, operate, service, or maintain it. The following special messages may appear throughout this documentation or on the equipment to warn of potential hazards or to call attention to information that clarifies or simplifies a procedure.

The addition of this symbol to a “Danger” or “Warning” safety label indicates that an electrical hazard exists which will result in personal injury if the instructions are not followed.

This is the safety alert symbol. It is used to alert you to potential personal injury hazards. Obey all safety messages that follow this symbol to avoid possible injury or death.

⚠ DANGER

DANGER indicates a hazardous situation which, if not avoided, **will result in** death or serious injury.

⚠ WARNING

WARNING indicates a hazardous situation which, if not avoided, **could result in** death or serious injury.

⚠ CAUTION

CAUTION indicates a hazardous situation which, if not avoided, **could result** in minor or moderate injury.

NOTICE

NOTICE is used to address practices not related to physical injury.

Please Note

Electrical equipment should be installed, operated, serviced, and maintained only by qualified personnel. No responsibility is assumed by Schneider Electric for any consequences arising out of the use of this material.

A qualified person is one who has skills and knowledge related to the construction and operation of electrical equipment and its installation, and has received safety training to recognize and avoid the hazards involved.

About the Document

Document Scope

This online help guide describes the TeSys Tera DTM Library for TeSys Tera Motor Management System.

The online help guide describes the four key parts of a successful system implementation:

- Installing the TeSys Tera DTM Library
- Configuring device parameters
- Monitoring the status of the device
- Maintaining and upgrading the TeSys Tera system

The online help guide is intended for the following TeSys Tera DTM Library users:

- Design engineers
- System integrators
- System operators
- Maintenance engineers

NOTE: The product image captured in this guide is for Ethernet variant of the TeSys Tera system, which will be available in the future releases. Please contact your local Schneider Electric representative concerning its availability.

Validity Note

This document is valid for:

- SoMove™ software version V2.9.9
- TeSys Tera DTM Library version V2.0.0

The availability of some functions described in this document depends on the communication protocol used and the physical modules installed on the TeSys Tera system.

General Cybersecurity Information

In recent years, the growing number of networked machines and production plants has seen a corresponding increase in the potential for cyber threats, such as unauthorized access, data breaches, and operational disruptions. You must, therefore, consider all possible cybersecurity measures to help protect assets and systems against such threats.

To help keep your Schneider Electric products secure and protected, it is in your best interest to implement the cybersecurity best practices as described in the Cybersecurity Best Practices document.

Schneider Electric provides additional information and assistance:

- Subscribe to the Schneider Electric security newsletter.
- Visit the Cybersecurity Support Portal web page to:
  - Find Security Notifications.
  - Report vulnerabilities and incidents.
- Visit the Schneider Electric Cybersecurity and Data Protection Posture web page to:
  - Access the cybersecurity posture.
  - Learn more about cybersecurity in the cybersecurity academy.
  - Explore the cybersecurity services from Schneider Electric.

Available Languages of the Document

The document is available in these languages:

- English

Related Documents

Title of documentation	Description	Reference number
TeSys Tera Motor Management System Catalog	The catalog: • Describes the TeSys Tera system • Contains the TeSys Tera technical characteristics	LVCATENTER
TeSys Tera Motor Management System User Guide	This is the main user guide that introduces the complete TeSys Tera system. It describes the main functions of the LTMT main units, LTMTCT/LTMTCTV sensor modules, LTMT expansion units, and LTMTCUF control operator unit.	DOCA0257EN
TeSys Tera Motor Management System Installation Guide	This guide describes the installation, commissioning, and maintenance of the LTMT main units, LTMTCT/ LTMTCTV sensor modules, LTMT expansion modules, and LTMTCUF control operator unit.	DOCA0356EN
TeSys Tera Motor Management System Modbus RTU Communication Guide	This guide describes the Modbus network protocol communication of the LTMT main unit.	DOCA0355EN
TeSys Tera Motor Management System PROFIBUS DP Guide	This guide describes the PROFIBUS DP network protocol communication of the LTMT main unit.	DOCA0256EN
TeSys Tera Motor Management System LTMTCUF Control Operator Unit User Guide	This guide describes how to install, configure, and use the LTMTCUF control operator unit.	DOCA0233EN
TeSys Tera Motor Management System DTM Library Software Release Note	This document provides important information about the TeSys Tera DTM Library software and provides summary of new features and enhancement.	DOCA0279EN
TeSys Tera Motor Management System Firmware Release Note	This document provides information about firmware package versions of the TeSys Tera system and provides summary of new features and enhancement.	DOCA0276EN

To find documents online, visit the Schneider Electric download center (www.se.com/ww/en/download/).

Information on Non-Inclusive or Insensitive Terminology

As a responsible, inclusive company, Schneider Electric is constantly updating its communications and products that contain non-inclusive or insensitive terminology. However, despite these efforts, our content may still contain terms that are deemed inappropriate by some customers.

Trademarks

QR Code is a registered trademark of DENSO WAVE INCORPORATED in Japan and other countries.

Precautions

Read and understand the following precautions before performing any procedures in this guide.

⚠⚠DANGER

HAZARD OF ELECTRIC SHOCK, EXPLOSION, OR ARC FLASH

- This equipment must only be installed and serviced by qualified electrical personnel.
- Turn off all power supplying to this equipment before working on this equipment.
- Use only the specified voltage when operating this equipment and any associated products.
- Always use a properly rated voltage sensing device to confirm power is off.
- Use appropriate interlocks where personnel and/or equipment hazards exist.
- Power line circuits must be wired and protected in compliance with local and national regulatory requirements.
- Apply appropriate personal protective equipment (PPE) and follow safe electrical work practices per NFPA 70E, NOM-029-STPS, or CSA Z462 or local equivalent.

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

⚠WARNING

UNINTENDED EQUIPMENT OPERATION

- Do not disassemble, repair, or modify this equipment. There are no user serviceable parts.
- Install and operate this equipment in an enclosure appropriately rated for its intended application environment.
- Each implementation of this equipment must be individually and thoroughly tested for proper operation before being placed into service.

Failure to follow these instructions can result in death, serious injury, or equipment damage.

California Proposition 65 Warning

⚠

WARNING: This product can expose you to chemicals such as, Humiseal 1A33 Polyurethane, which is known to the State of California to cause cancer and birth defects or other reproductive harm. For more information go to www.P65Warnings.ca.gov.

Qualified Personnel

Only appropriately trained personnel who are familiar with and understand the content of this guide and all other related product documentation are authorized to work on and with this product.

The qualified personnel must be able to detect possible hazards that may arise from modifying parameter values and generally from mechanical, electrical, or electronic equipment. The qualified personnel must be familiar with the standards, provisions, and regulations for the prevention of industrial accidents, which they must observe when designing and implementing the system.

The use and application of the information contained in this guide requires expertise in the design and programming of automated control systems. Only you,

the user, panel builder, or integrator, can be aware of all the conditions and factors present during installation, setup, operation, and maintenance of a process plant or machine, and can therefore determine the automation and associated equipment and the related safeties and interlocks which can be effectively and properly used when selecting automation and control equipment, and any other related equipment or software, for a particular application. You must also consider applicable local, regional, or national standards and/or regulations.

Pay particular attention to conformance with any safety information, electrical requirements, and normative standards that apply to your process plant or machine in the use of this equipment.

Intended Use

The products described in this guide, together with software, accessories, and options, are a part of starters for low-voltage electrical loads, intended for industrial use according to the instructions, directions, examples, and safety information contained in the present document and other supporting documentation.

The product may only be used in compliance with all applicable safety regulations and directives, the specified requirements, and the technical data.

Before using the product, you must perform a risk assessment of the planned application. Based on the results, appropriate safety-related measures must be implemented.

Since the product is used as a component of a process plant or machine, you must ensure the safety of personnel by means of the overall system design.

Operate the product only with the specified cables and accessories. Use only genuine accessories and spare parts.

Any use other than the use explicitly permitted is prohibited and can result in unanticipated hazards.

Security Hardening Guidelines

Introduction

Your PC can run a variety of applications to enhance security in your control environment. The system has factory default settings that require reconfiguration to align with Schneider Electric's device hardening recommendations of the defense-in-depth approach.

The following guidelines describe procedures in a Windows operating system. They are provided as examples only. Your operating system and application may have different requirements or procedures.

Disabling the Remote Desktop Protocol

Schneider Electric’s defense-in-depth approach recommendations include disabling remote desktop protocol (RDP) unless your application requires the RDP.

In Windows 10, remote desktop protocol (RDP) is disabled using **Settings > System > Remote Desktop > Enable Remote Desktop** (toggle to **Off**).

Updating Security Policies

Update the security policies on the PCs in your system by `gpupdate` in a command window. For more information, refer to the Microsoft documentation on `gpupdate`.

Managing Updates

Before deployment, update all PC operating systems using the utilities on Microsoft’s **Windows Update** Web page. To access this tool in Windows, select **Start > All Programs > Windows Update**.

Workstation Protection

To reduce the security risks associated with the engineering workstation, enable the memory exploit settings such as Data Execution Prevention (DEP) and Address Space Layout Randomization (ASLR). These security settings can be enabled by using the system exploit protection settings in Windows 10 operating system. For more information, refer to the Microsoft security features web page.

Introduction

What's in This Part

TeSys Master Range 13
TeSys Tera System 14
Definitions 16
Prerequisites for TeSys Tera DTM Library Installation 17
Installing SoMove 18
Installing TeSys Tera DTM Library 20
Connecting the LTMT Main Unit to PC 21
Troubleshooting 22

TeSys Master Range

TeSys is an innovative motor control and management solution from the global market leader. TeSys offers connected, efficient products and solutions for switching and protection of motors and electrical loads in compliance with all major global electrical standards.

TeSys Tera System

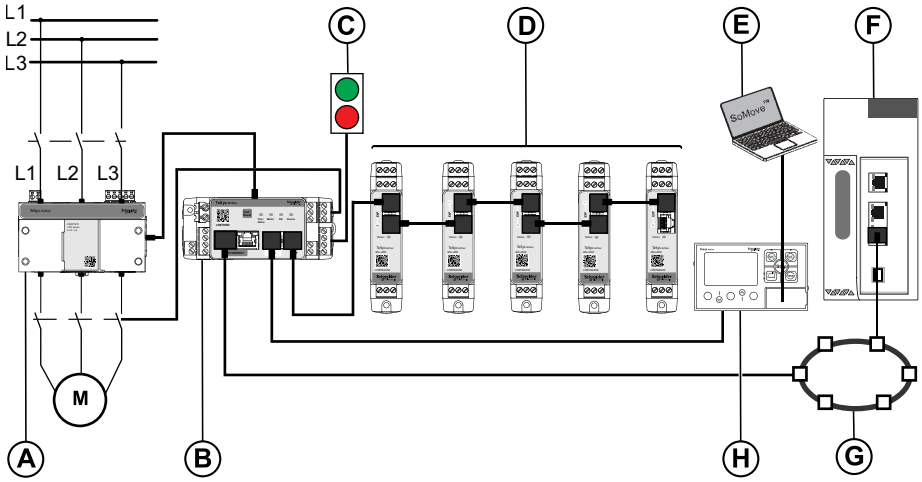
Overview

The TeSys Tera Motor Management System (or TeSys Tera system) is part of the TeSys™ Active range of intelligent relays and motor starters. The TeSys Tera system is designed as a reliable building block for Intelligent Motor Control Centres (iMCCs) to provide complete protection, control, and monitoring capabilities for single-phase or three-phase AC induction motors.

The TeSys Tera system is installed in the low voltage switchgear system and connects the higher level automation system via fieldbus network and the motor feeder.

TeSys Tera system:

- Covers conventional and advanced motor protection, metering, and monitoring in iMCC feeders into single, easy to configure, compact communicating module with a display.
- Provides protection controller for low voltage contactor-controlled motor starter feeders.
- Provides flexible and modular motor management system for motors with constant speeds in low voltage applications.



NOTE: Please contact your local Schneider Electric representative concerning the availability of the EtherNet/IP variant.

A LTMTCT/LTMTCTV sensor module

B LTMT main unit

C Start/Stop commands

D LTMT expansion units

E PC running the TeSys Tera DTM embedded in a FDT container, such as SoMove software

F Programmable Logic Controller (PLC) or Distributed Control System (DCS)

G Communication network

H LTMTCUF control operator unit

Functional Characteristics

The TeSys Tera system manages:

- Single-phase or three-phase AC induction motors up to 100 A with integral sensor module.
- Single-phase or three-phase AC induction motors up to 810 A when using external current transformers.
- The connection between the control system and the motor feeder, increases plant availability.
- Significant savings to the installation, commissioning, operation, and maintenance.
- Numerical microprocessor equipped controller that allows to set parameters of the motor according to the application and process requirements.

Definitions

Device Type Manager

The Device Type Manager (DTM) is a software module hosted in an FDT container for a specific device.

The functions of the DTM include:

- Scanning various field buses for devices
- Device power and energy monitoring
- Management of configuration of device parameters
- Project file management
- Customizing of device parameter units
- Troubleshooting
- Update the firmware of device

SoMove Software Project File

A SoMove software project file is a configuration file for a pre-determined device, that can be created offline and saved for later use.

A project file contains the following information:

- Device topology configuration
- All parameters settings

NOTE:

- The project file does not contain the customized program and should be saved manually.
- This file is saved with the extension **.psx*.

For more information on how to create a project, refer to *SoMove Online Help*.

Prerequisites for TeSys Tera DTM Library Installation

System Requirements

The TeSys Tera DTM Library can be installed on Microsoft Windows 10 and 11 operating system.

It is recommended to install TeSys Tera DTM Library on Microsoft Windows 11 operating system.

Software Requirements

The TeSys Tera DTM Library requires the following software to be installed on the PC:

- Microsoft .NET Framework v3.5
- SoMove software 2.9.9
- TeSys Tera DTM Library v2.0.0

Hardware Requirements

The following hardware requirements are recommended on the PC to install the TeSys Tera DTM Library:

Equipment	Minimum requirement	Recommended requirement
Processor	Pentium 4/Core 2 Duo, 2 GHz	Intel® Core™ i3
RAM	2 GB	4 GB
Display	Resolution: 1024 x 768, 1366 x 768, 1600 x 1900, and 1920 x 1080 pixels	Resolution: 1600 x 1200 and 1920 x 1080 pixels
	DPI: 96 (100%) and 120 (125%)	DPI: 96 (100%) and 120 (125%)
Free hard disk space on system drive	1 GB	2 GB

Installing SoMove

Overview

SoMove software is a Microsoft Windows® based application, using the open FDT or DTM technology. SoMove software contains multiple DTMs.

Downloading SoMove Software

SoMove software can be downloaded from Schneider Electric website.

NOTE: You must have administrator rights to download, install, or uninstall the software on your PC.

Installing SoMove Software

The following steps describe how to install SoMove software:

1. Unzip the downloaded file *SoMove_FDT*.
2. The folder will contain an .exe file named *SoMove _VX.Y.Z* (where X.Y.Z is the version number) and a release note.
3. Double-click *SoMove _VX.Y.Z* to start the installation.
4. In the **Choose Setup Language** dialog, select the installation language.
5. Select **OK**.
6. In the **Welcome to the Installation Wizard for SoMove** dialog, select **Next**.
7. If an **Install Shield Wizard** dialog appears and informs you that you must install Modbus driver, select **Install**.

   Result: Modbus driver is installed automatically.
8. In the **Readme and Release Notes** dialog, select **Next**.
9. In the **License Agreement** dialog:
   - Read carefully the license agreement.
   - Select **I accept the terms in the license agreement** option.
   - Select **Next**.
10. In the **Customer Information** dialog:
    - Enter the following information in the corresponding fields:
      - **First name**
      - **Last name**
      - **Company name**
    - Select **Next**.
11. In the **Destination Folder** dialog:
    - If necessary, modify the SoMove software destination folder by selecting **Change** option.
    - Select **Next**.
12. In the **Shortcuts** dialog:
    - If you want to create a shortcut on the desktop and or in the quick launch bar, select the corresponding options.
    - Select **Next**.

13. In the **Ready to Install the Program** dialog, select **Install**.

    Result: The SoMove software components are installed automatically with:

    - Modbus communication DTM library which contains the communication protocol.
    - DTM libraries which contain different catalogs.
    - SoMove software

14. In the **Installation Wizard Completed** dialog, select **Finish**.

    Result: SoMove software is installed on your PC.

Installing TeSys Tera DTM Library

Overview

In SoMove software, a specific DTM exists for the TeSys Tera system. The TeSys Tera DTM Library must be installed after installing the SoMove software.

Downloading TeSys Tera DTM Library

TeSys Tera DTM Library can be downloaded from the Schneider Electric website (www.se.com)

Enter `TeSys Tera DTM Library` in the **Search** field to navigate to the downloads page.

NOTE: You must have administrator rights to install or uninstall the software on your PC.

Installing TeSys Tera DTM Library

The following steps describe installation of TeSys Tera DTM Library:

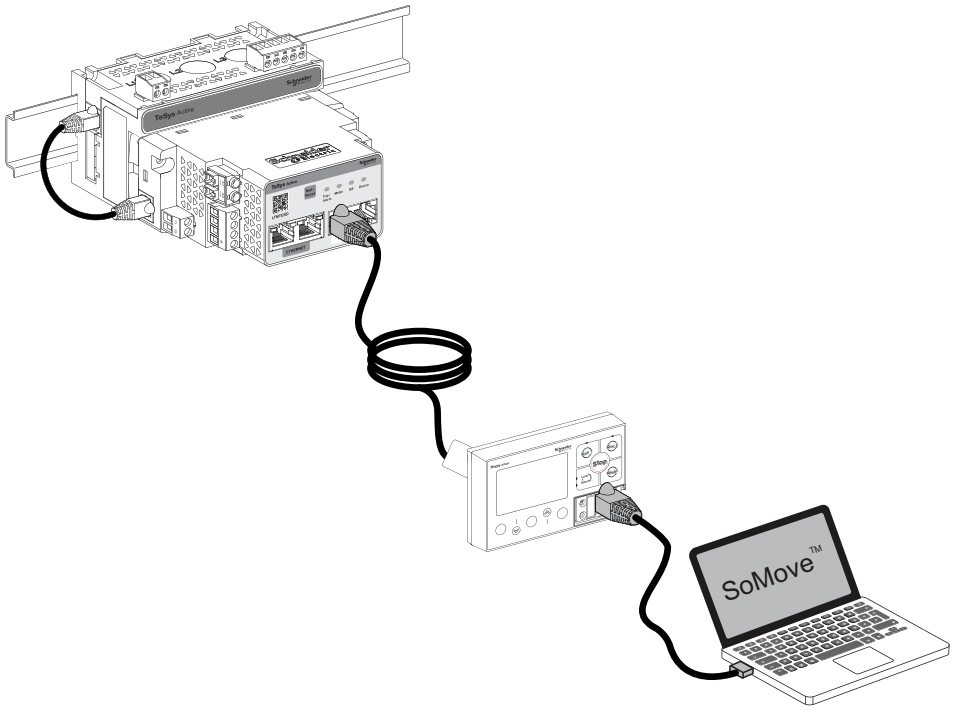
1. Unzip the downloaded file *TeSysTeraDTMLibrary*.
2. The folder will contain an exe file *Schneider_Electric_TeSys_Tera_DTM_Library_V.X.Y.Z* (where X.Y.Z is the version number) and a release note *TeSysTeraDTMLibrary_vx.y.z_ReleaseNotes* (where x.y.z is the version number).
3. Double-click *Schneider_Electric_TeSys_Tera_DTM_Library_V.X.Y.Z* to start the installation.
4. In the **Choose Setup Language** dialog, select the installation language and select **OK**.
5. In the **Welcome to the Installation Wizard for Schneider Electric TeSys Tera DTM Library** dialog, select **Next**.
6. In the **Readme and Release Notes** dialog, select **Next**.
7. In the **License Agreement** dialog:
   - Read the license agreement carefully.
   - Select **I accept the terms** in the license agreement option.
   - Select **Next**.
8. In the **Customer Information** dialog:
   - Enter the following information in the corresponding fields:
     - First name
     - Last name
     - Company name
   - Select **Next**.
9. In the **Destination Folder** dialog:
   - If necessary, modify the TeSys Tera DTM Library destination folder by selecting the **Change** option.
   - Select **Next**.
10. In the **Ready to Install the Program** dialog, select **Install**.
11. In the **Installation Wizard Completed** dialog, select **Finish**.

    Result: The TeSys Tera DTM Library is installed on your PC.

Connecting the LTMT Main Unit to PC

The LTMT main unit can be connected to the PC running SoMove software in the following ways:

- Connecting the USB port of the PC running SoMove software to the RJ45 port on the LTMTCUF control operator unit using the USB/RJ45 cable for configuration and firmware update (recommended).
- Connecting the USB port of the PC running SoMove software to the HMI port on the LTMT main unit using the USB/RJ45 cable for configuration and firmware update.
- Connecting the USB port of the PC running SoMove software to the fieldbus port for configuration.



For more information on configuring the various communication protocol and establishing the connection between the TeSys Tera system and DTM library, refer to section Connection in the *SoMove Online Help*.

For more information on connecting the PC to the TeSys Tera system with LTMTCUF control operator unit, refer to *TeSys Tera Motor Management System LTMTCUF Control Operator Unit User Guide – DOCA0233EN*.

Troubleshooting

Problem	Cause	Solution
TeSys Tera DTM Library does not appear in the **Catalog** window in SoMove software v2.9.x and later.	SoMove software is not registered before applying the DTM.	Register SoMove software. 1. When launching SoMove software, the user will be prompted with the following message: **x Days left for the demo version to expire. Do you want to register now?** 2. Select **Yes** to confirm the registration and follow the instructions. **NOTE:** The registration of SoMove software is free.
	.dll files are not registered by Windows	Use the command prompt window to register .dll file required for the TeSys Tera DTM Library. 1. Launch command prompt window as administrator and enter the following command: **cd C:\Windows\Microsoft.NET \Framework\v2.0.50727** 2. Enter the following command: **RegAsm.exe /codebase "C: \Program Files (x86)\Common Files \Schneider Electric Shared \Schneider Electric TeSys Tera DTM Library\TeSysTerraDtm.Kernel.dll** **Result:** When the registration is successful, a message will be displayed. 3. Launch SoMove software. The **Catalog Update** screen with progress bar will appear. **Result:** After the catalog update is completed, the TeSys Tera DTM Library will be available in the **Catalog** window.

User Interface

What's in This Part

Launching the TeSys Tera DTM Library....24
User Interface Description....30
Menu Bar....31
Tool Bar....35
Status Bar....37
Tab Zone....39
My Device....42
Parameters List....46
My Dashboard....56
Diagnostics....58
Monitoring....64
Start Curve....68
Custom Logic....70
FB Diagram....71
Logic Simulator....72

Launching the TeSys Tera DTM Library

Launching SoMove

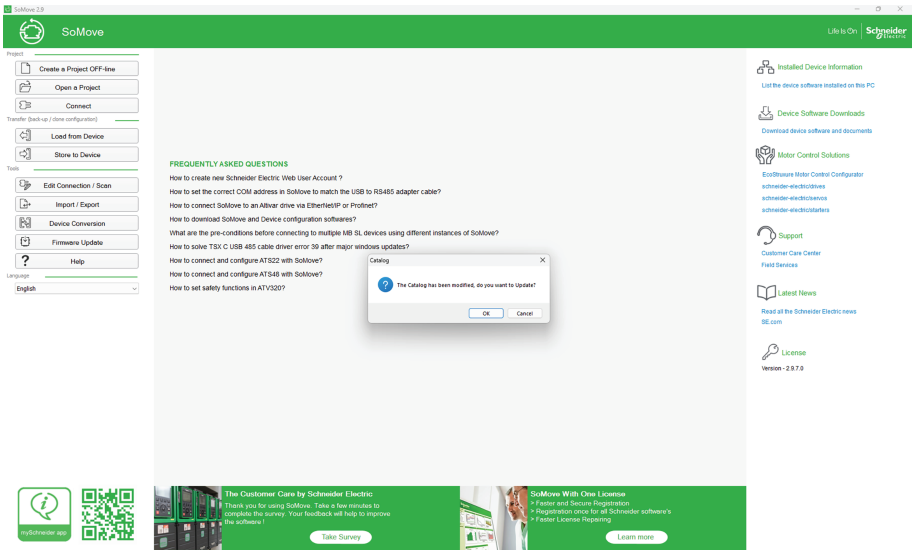
To create an instance of the TeSys Tera DTM Library, you need to first launch SoMove software.
For more information on how to launch the SoMove software, refer to *SoMove Online Help*.

Creating an Instance of the TeSys Tera DTM

The following procedure describes how to create an instance of the TeSys Tera DTM Library:

1. Open SoMove software.

   Result: The following window is displayed.

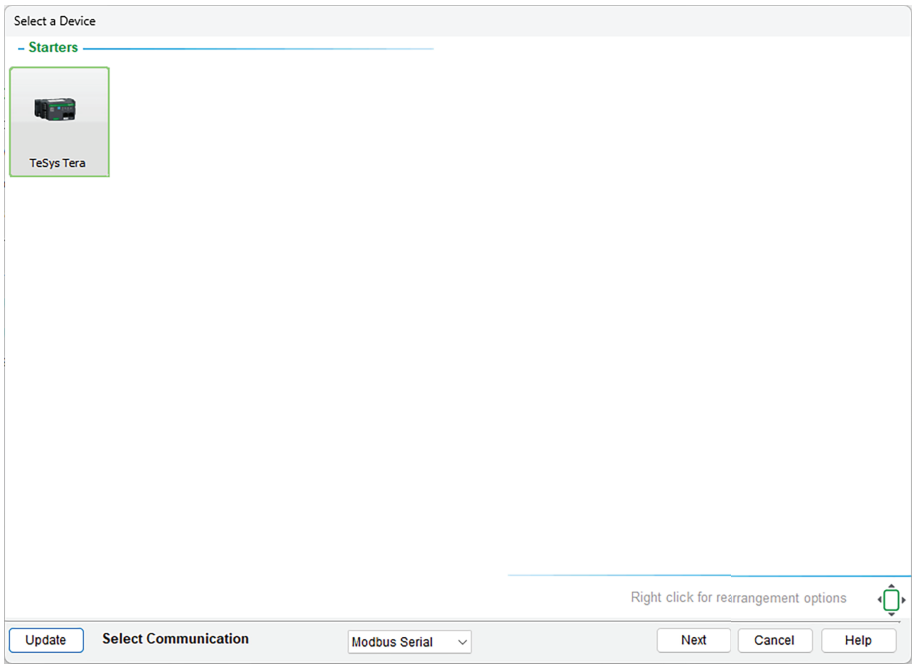


2. Select **OK** to update the catalog.

   NOTE: The **Catalog** window appears for the first time when you install the TeSys Tera DTM Library and launch SoMove software.

3. Select **Create a Project OFF-line**.

   Result: The **Select a Device** window appears.



4. Select **TeSys Tera**, and select **Next**.

   Result: The TeSys Tera system work area opens.

   NOTE:

   - If the device does not appear, then refer to Troubleshooting, page 22.
   - The communication between TeSys Tera system and TeSys Tera DTM Library is through Modbus serial protocol only, irrespective of other fieldbus protocols.

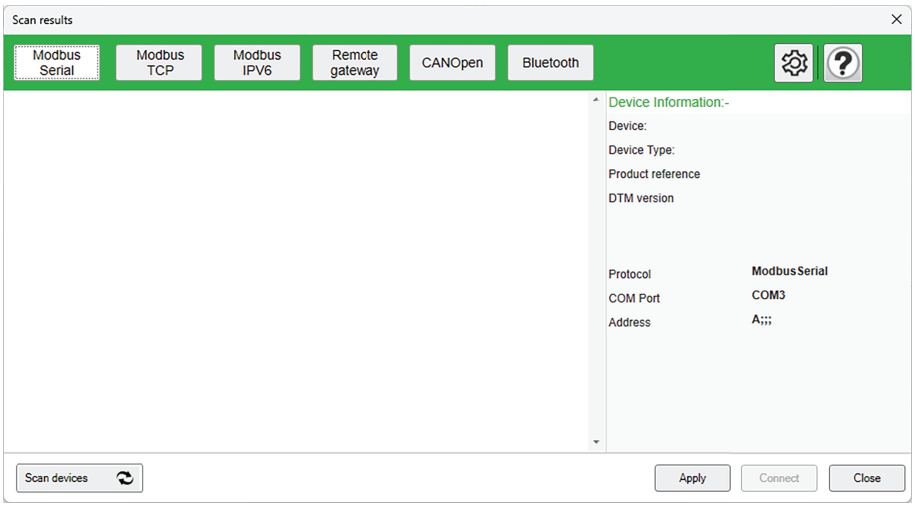
Connecting the Device to Network

This action retrieves the configuration of the connected device. SoMove software remains connected to this device during the session.

The following procedure describes how to connect to a device:

1. Open SoMove software.
2. Connect to the device by one of the following ways:
   - In the Start page, select **Edit Connection/Scan**.
   - On the toolbar, select the icon.
   - On the menu bar, select **Communication > Edit Connection/Scan**.

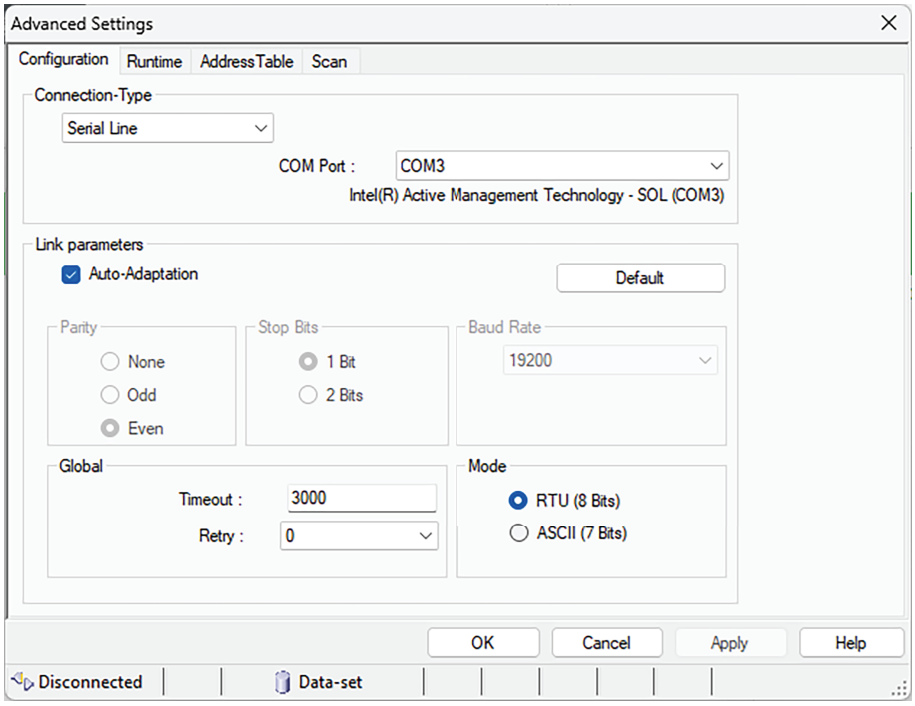
   Result: The **Scan results** window appears.



3. In the **Scan results** dialog, select **Modbus Serial** tab.

4. Select the  icon.

   Result: The **Advanced Settings** window appears.



   NOTE: The factory settings of the device is as follows (applicable for HMI or Modbus RTU port):

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

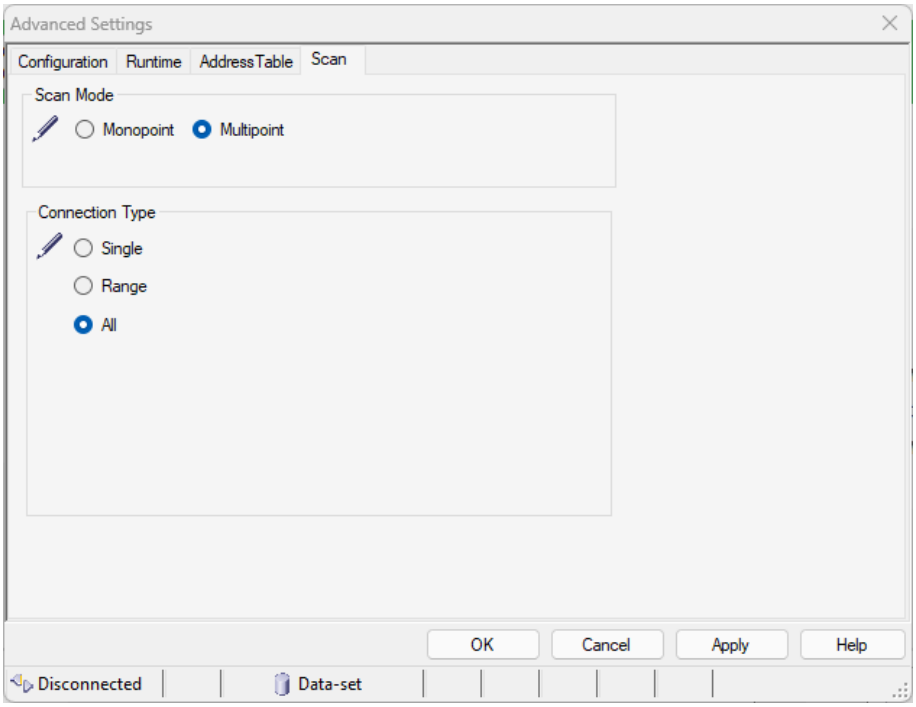
5. In the **Configuration** tab, enter the required configuration settings for the Modbus serial connection type.

   NOTE: If the default parameter settings are not available, select **Auto-Adaptation** check box to discover the device.

   For more information on the configuration settings, refer to Connection section in the *SoMove Online Help*.

6. Select the **Scan** tab.

   Result: The **Scan** tab in **Advanced Settings** window appears.



   NOTE: Only **Multipoint Scan Mode** is supported for Modbus connection.

7. Select the **Scan Mode** and the **Connection Type**.

   For more information, refer to Connection section in the *SoMove Online Help.*

8. Select **OK**.
   Result: The modifications are saved and the **Advanced Settings** window is closed. The new values will be applied in the next scan.

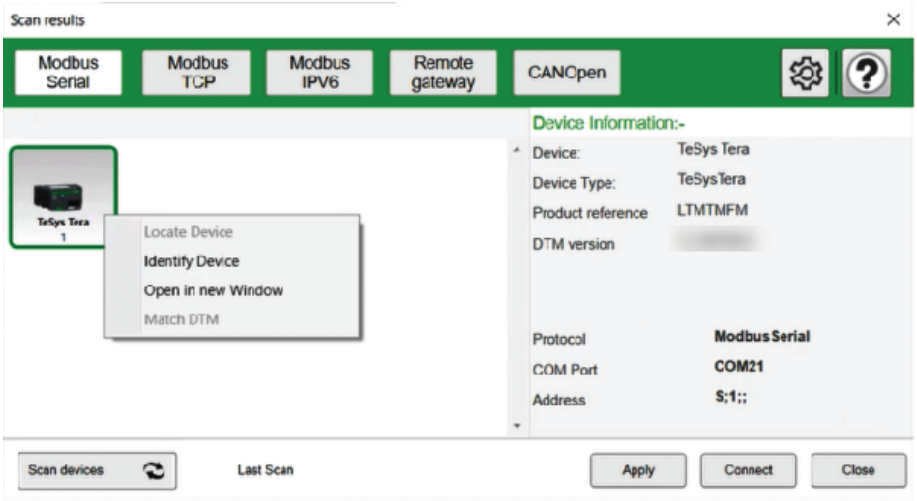
   NOTE: If you select **Cancel**, the **Advanced Settings** window closes without any changes and the default values are applied in the next scan.

9. Select **Scan devices**.
   Result: Displays all the devices on the network with Modbus serial connectivity.

10. Select the suitable TeSys Tera system and select **Connect**.

11. Right-click on the device.

    Result: The **Scan results** window appears.



    Select **Identify Device** if you want to know more about the device.

12. Select the identified device for connection and select **Connect**.

13. Warning message will appear as shown below. Check if all the requirements are met as per the safety message and select **OK**.

⚠WARNING

UNINTENDED EQUIPMENT OPERATION

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

- Only appropriate trained personnel familiar with and understand the content of all pertinent product documentation, who has received safety training to recognize and avoid hazards involved, are authorized to work with this drive system.
- Only use this software for setup and commissioning tasks and to display status information.
- Verify that other applications on your PC such as and not limited to, screen-savers, cannot interfere with the proper operation of the software.
- Verify that an integrated and functioning emergency stop push-button is within reach.
- Verify that all parameter settings are suitable for this application.
- Do not operate the product with unknown and unsuitable settings and data.

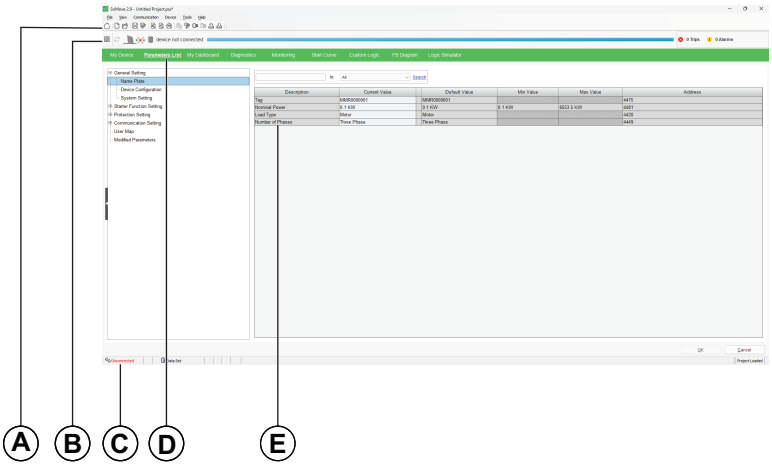
Failure to follow these instructions can result in death, serious injury, or equipment damage.

Result: The TeSys Tera system is connected to your PC.

NOTE:

- Verify that the cable connection between the device and the PC.
- Verify that the device is connected to the power supply.
- Select **Advanced Setting** to verify that the connection parameters are defined correctly.

User Interface Description



The working space is divided into the following zones:

A	Menu bar
B	Tool bar
C	Status bar
D	Tab selection bar
E	Tab zone (content depending on the selected tab)

Menu Bar

Description

The menu bar is a part of SoMove software. The menu bar, at the top of the working space, is represented below:

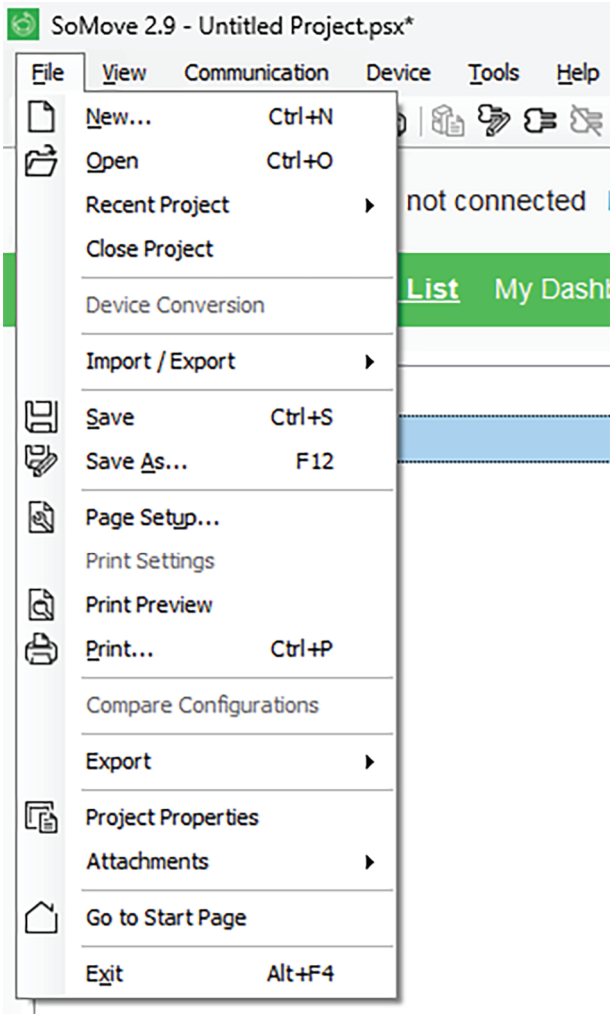
The menu bar gives access to user functions, via menus or icons.

In a menu bar, the commands in grey are unavailable commands.

NOTE: Some options of SoMove software are not applicable for TeSys Tera DTM Library. Details on the unavailable commands are not listed in the description table. For more information about the options available in SoMove software, refer to *SoMove Online Help.*

File Menu

The **File** menu can be accessed by clicking **File** on the menu bar.

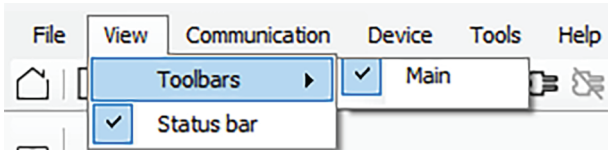


Command	Description
New...	Creates a new project.

Open	Opens a project saved on your PC.
Recent Project	Opens the recently open project.
Close Project	Closes the current project by prompting to save.
Save	Allows you to save modifications to an existing project.
Save As	Saves an open project under a name and or in a new location.
Project Properties	Displays the properties of the open project.
Go To Start Page	Displays the start page.
Exit	Exits **SoMove** software.

View Menu

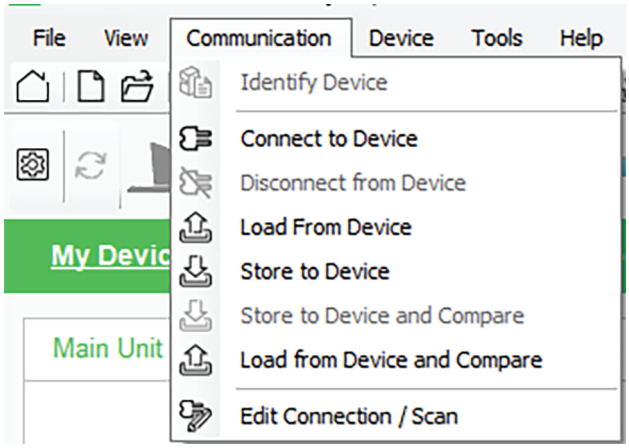
The **View** menu can be accessed by clicking **View** on the menu bar.



Command	Description
Toolbars • Main	Displays or hides the main toolbar.
Status bar	Displays or hides the status bar.

Communication Menu

The **Communication** menu can be accessed by clicking **Communication** on the menu bar.

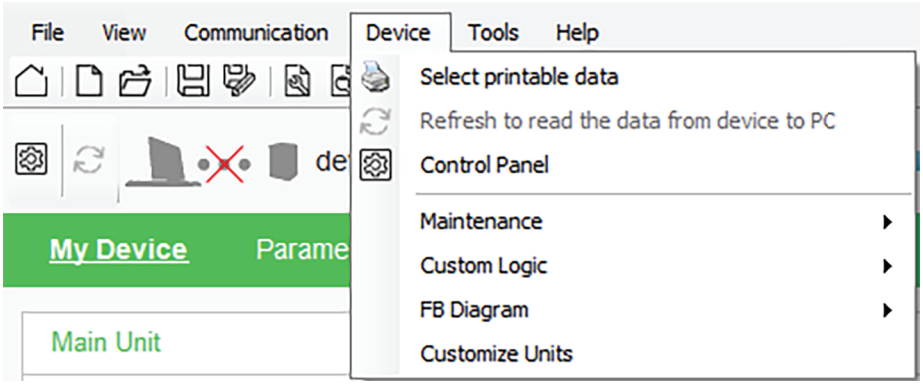


Command	Description
Identify Device	Displays the following information about the connected device: • Name • Manufacturer • Software version
Connect to Device	Retrieves the configuration of the connected device and remains connected during the whole session.

Disconnect from Device	Disconnects from the connected device.
Load From Device	Retrieves the configuration of the connected device. **NOTE:** When this operation is in progress, make sure that DTM library is not disconnected from the device to avoid loss of communication.
Store to Device	Transfers the configuration file from an existing project saved on your PC to the connected device. **NOTE:** This operation will take a minute to download the configuration.
Load from Device and Compare	Retrieves the configuration from the connected device and compares it with the currently opened project configuration.
Edit Connection / Scan	Displays the connection parameters and performs a connection test.

Device Menu

The **Device** menu can be accessed by clicking **Device** on the menu bar.

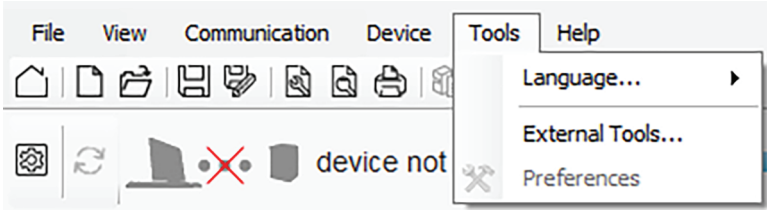


Command	Description
Select printable data	Allows you to select the required data for printing.
Refresh to read the data from device to PC	Refreshes all parameters in the parameter list tab.
Control Panel	Displays or hides the control panel.
Maintenance	Access to firmware update command.
Custom Logic • New Custom Logic • Open Custom Logic • Save Custom Logic • SaveAs Custom Logic • Close Custom Logic • Compile Custom Logic	Provides access to the custom logic editor for customized programs in custom logic language. • Opens a new custom logic program file. • Open an existing custom logic program file. • Saves the custom logic program file. • Allows you to save the custom logic program file in a different location with a new file name. • Closes the custom logic program file by prompting to save. • Validates the code written in the custom logic program with respect to the standards.
FB Diagram • New FB Diagram • Open FB Diagram	Gives access to the custom logic editor of customized programs in Function Block Diagram (FBD) language. • Allows you to create a new FB Diagram. • Allows you to open any existing FB Diagram file.

• Save FB Diagram As • Compile FB Diagram to Custom Logic • FBD Editor • View • Tools	• Allows you to save the FB Diagram file in a different location with a new file name. • Compiles FB Diagram and converts it to Custom Logic. • Allows you to create/edit the FB Diagram. • Allows you to view the FB Diagram with various available viewing options. For more information, refer to FBD Editor Display Options, page 176. • Displays tools to change the links or to reorder the function blocks.
Customize Units	Allows you to configure the following display units: • Temperature (Celsius or Fahrenheit) • Motor Rating (KW or HP)

Tools Menu

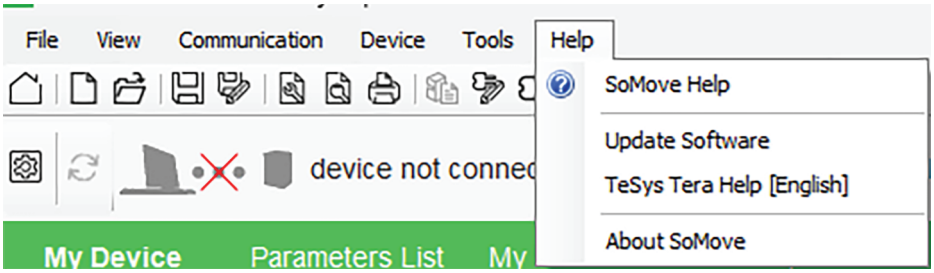
The **Tools** menu can be accessed by clicking **Tools** on the menu bar.



Commands in Tools Menu	Description
Language	Modifies the display language.

Help Menu

The Help menu can be accessed by clicking **Help** on the menu bar.

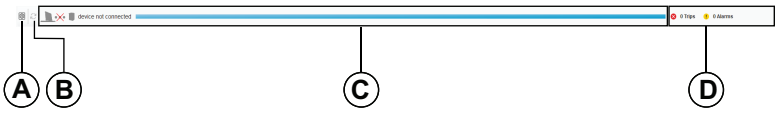


Command	Description
SoMove Help	Displays the *SoMove Online Help*.
Update Software	Updates SoMove software to the latest version.
TeSys Tera Help [English]	Opens the TeSys Tera DTM Online Help.
About SoMove	Displays general information about SoMove software.

Tool Bar

Description

The tool bar, at the top of the working space directly below the menu bar, is specific to the TeSys Tera DTM Library:

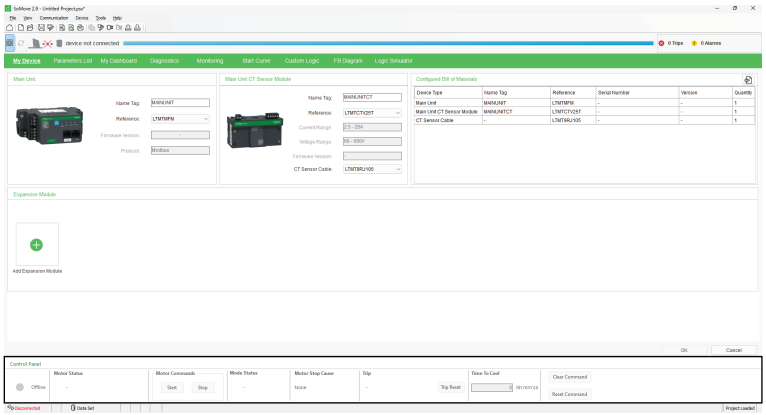


A	Control panel
B	Refresh
C	Synchronization data area
D	Trip/Event and Alarm count display area

Control Panel

To access the **Control Panel**, select on the tool bar. The **Control Panel** appears at the bottom of the window.

The **Control Panel** appears at the bottom of the window.



The **Control Panel** displays the following parameters:

- Device status
- Motor status
- Mode status
- Motor stop cause
- Trip status
- Time to cool when motor is stopped or tripped

The **Control Panel** allows you to perform the following actions:

- Issue motor commands (Start/Stop)
- Trip reset
- Clear command
- Reset command

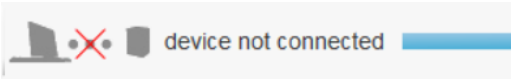
Refresh

The option on the tool bar refreshes all the parameters under the **Parameter List** tab.

Synchronization Data Area

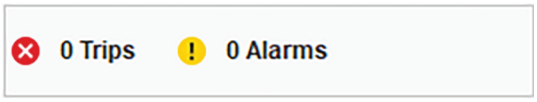
The synchronization data area displays the synchronization status of the data between the LTMT main unit and the PC.

When the LTMT main unit is in the connected mode, displayed data is automatically synchronized.

Mode	Icon	Description
Disconnected		The LTMT main unit is not synchronized with the PC: • Parameters list headers and synchronization data area are blue. • Parameters are not read in real time from the LTMT main unit. • All settings can be modified as in configuration mode. • You should perform **Store to Device** operation to write the settings to the device. To save this setting, you have to save the project.
Connected		The LTMT main unit is synchronized with the PC: • Parameters list headers and synchronization data area are orange. • Parameters displayed are read in real time from the LTMT main unit • Modified parameters are written to device on confirmation.

Trip/Event and Alarm Count Display

This area displays the total number of Trip/Event and Alarm in the project.

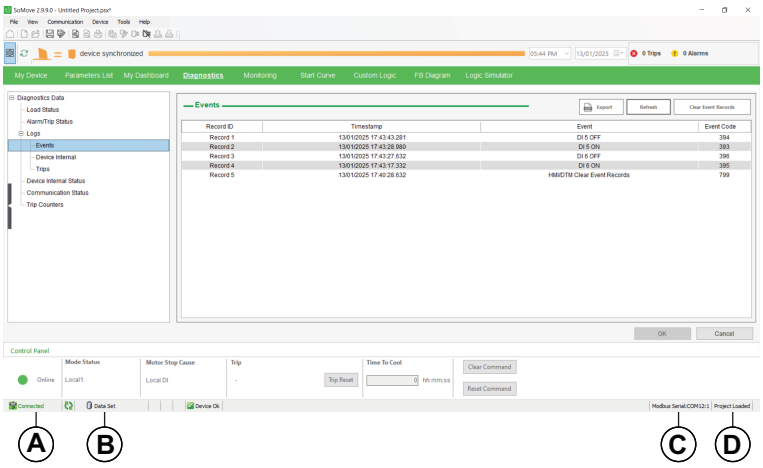


Status Bar

Objective

The status bar, at the bottom of the working space, displays the current status of the LTMT main unit and information related to SoMove software. For more information on the status bar, refer to SoMove *Online Help*.

Description



A	Connection status
B	Data source
C	Specific message zone
D	Project status

Connection Status

The connection status indicates the connection mode between the LTMT main unit and the PC:

Mode	Icon	Description
Disconnected		The LTMT main unit is not connected to the PC.
Disturbed		The connection between the LTMT main unit and the PC is lost.
Connected		The LTMT main unit is connected to the PC.

Data Source

The **Data Source** indicates the current mode of data. The following tables describes the data source modes:

Mode	Icon	Description
Data set		The parameter values in the data set can be modified in connected mode.
Data set or device locked		The device is protected in connected mode.
Device		The device data set contains information.
Device/Data set		All parameter values are stored to both data sources.

Specific Message Zone

Depending on the fieldbus used for communicating with the device, the specific message zone displays the IP or device address, COM port details and so on.

Project Status

The status of the SoMove software project can be:

- **Project Loaded**: Project is displayed in the working space.
- **No Project Open**: Project working space is empty.

Tab Zone

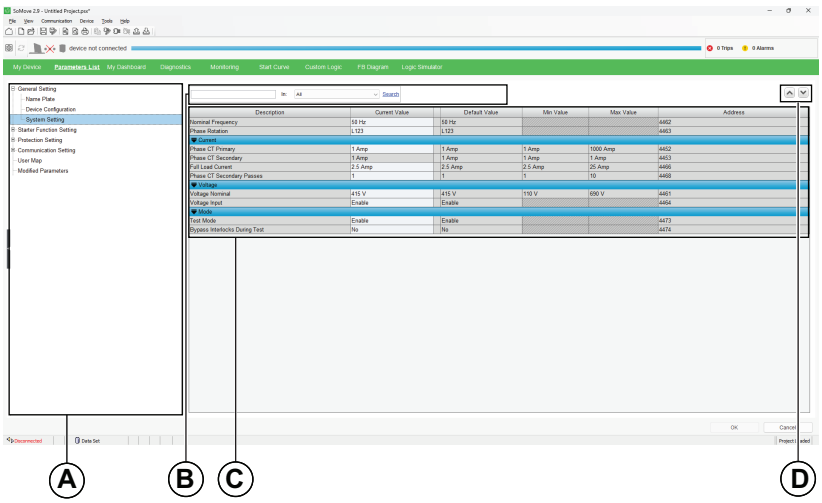
Overview

The table below lists the various tabs available.

Tab name	Description
My Device	Displays the device modules and characteristics.
Parameter List	Displays the configurable parameters of all device modules.
My Dashboard	Displays the configurable and status parameters of all device modules, which can be customize based on the requirement.
Diagnostic	Displays the diagnostic parameter data.
Monitoring	Displays the device modules monitoring data.
Start Curve	Displays the motor starting characteristics in a graphical format.
Custom Logic	Gives access to the custom logic editor of customized programs in custom logic language.
FB Diagram	Gives access to the custom logic editor of customized programs in Function Block Diagram (FBD) language.
Logic Simulator	Gives access to the logic simulator of customized programs in custom logic language.

Description

The window displays the common information in these tabs:



A	Tree view
B	Search function
C	Display area
D	Expand and Collapse buttons

Tree View

The tree view comprises of the parameter groups divided as items and sub-items. Select an item or a sub-item in the tree to display the parameter on the display area.

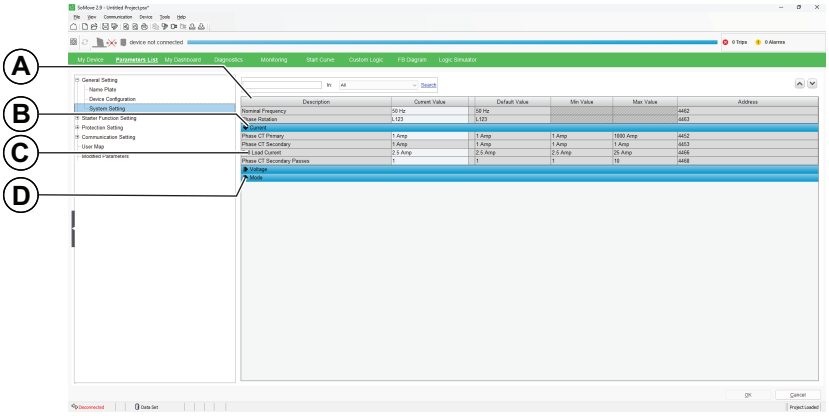
Search Function

The following steps describe how to find a specific text in the displayed table:

1. In the first field of the search bar, enter the characters to search for (part of word, code, or unit).
2. Select the column to search from the drop-down list.

   If you select the **All** option, the search is performed in all columns of the table.
3. Select **Search**:
   - The first matching text found is highlighted.
   - To search for other instances, select **Search**.
   - If no matching text is found, the color of characters in the search field turns red.

Display Area in Grid View



A Column header

B Parameter group

C Parameters:
- One parameter per row with parameter properties displayed in different fields.
- Content of white cells can be modified, gray cells are read-only.

D Collapse or Expand icon: To collapse or expand a parameter group, click the arrow of the corresponding group.

Sorting Parameters

To sort the parameters according to the values in a column, click on the header:

Action	Result	Header Example
Single click	• Parameter values are sorted in ascending order. • Header appears with an arrow pointing upwards.	
Double click	• Parameter values are sorted in descending order. • Header appears with an arrow pointing downwards.	
Triple click	• Parameters are displayed in their initial order. • Header appears according to its initial representation.	

Modifying the Order of Columns

The following steps describe how to modify the order of columns in the display area:

1. Click the header of the column.
2. Drag the column to the desired location.

Expand and Collapse Button

The view of the display area can be modified using the following buttons available on the top right corner of the display area:

Button	Function	Description
	Expand All	Expand all groups to display all parameters.
	Collapse All	Collapse all groups in the display area.

My Device

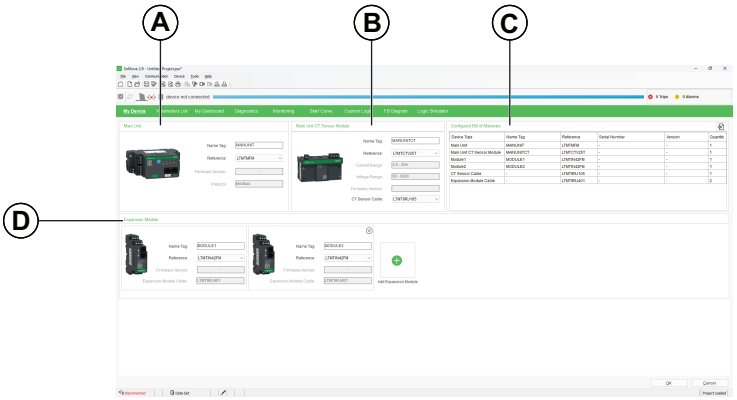
Overview

The **My Device** tab is used to:

- Configure the TeSys Tera system in disconnected mode.
- Display the TeSys Tera system characteristics in connected mode.

Description

This figure presents the different sections of **My Device** tab.



A	Main Unit
B	Main Unit CT Sensor Module
C	Configured Bill of Materials
D	Expansion Modules

Main Unit

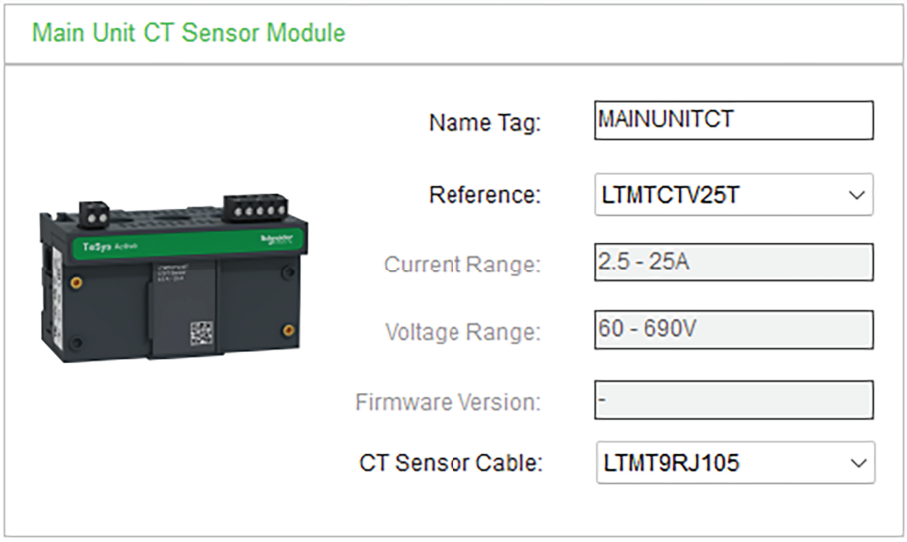
The **Main Unit** section displays the characteristics of the LTMT main unit configured in the TeSys Tera system.

The following characteristics are available in the **Main Unit** section:

Characteristics	Description
Name Tag	Enter the name of the LTMT main unit. **NOTE:** The **Name Tag** field supports the following characters: • Alphabets: A to Z (upper and lower case) • Numbers: 0 to 9 • Symbols: (,), _, /, –, @, #, ' ' and (space)
Reference	Select the reference of the LTMT main unit.
Firmware Version	Displays the firmware version of the selected LTMT main unit. **NOTE:** The firmware version is only displayed when the device is online.
Network Port Protocol	Displays the network protocol of the selected LTMT main unit.

NOTE: If the LTMT main unit reference configured in the DTM library is different from the actual device configuration, an error message will be displayed and the connection will not be established.

Main Unit CT Sensor Module



The **Main Unit CT Sensor Module** section displays the characteristics of the main unit CT sensor module configured in the TeSys Tera system.

The following characteristics are available in the **Main Unit CT Sensor Module** section:

Characteristics	Description
Name Tag	Enter the name of the **Main Unit CT Sensor Module**. **NOTE:** The **Name Tag** field supports the following characters: • Alphabets: A to Z (upper and lower case) • Numbers: 0 to 9 • Symbols: (,), _, /, –, @, #, ' ' and (space)
Reference	Select the reference of the **Main Unit CT Sensor Module**.
Current Range	Displays the current range of the selected sensor module.
Voltage Range	Displays the voltage range of the selected sensor module. **NOTE:** The voltage range is displayed for the voltage variants only.

Characteristics	Description
Firmware Version	Displays the firmware version of the selected main unit CT sensor module. **NOTE:** The firmware version is only displayed when the device is online.
CT Sensor Cable	Select the type of sensor cable used.

Expansion Module

The **Expansion Module** section displays the details of the expansion modules configured in the TeSys Tera system.

The following characteristics are available in the **Expansion Module** section:

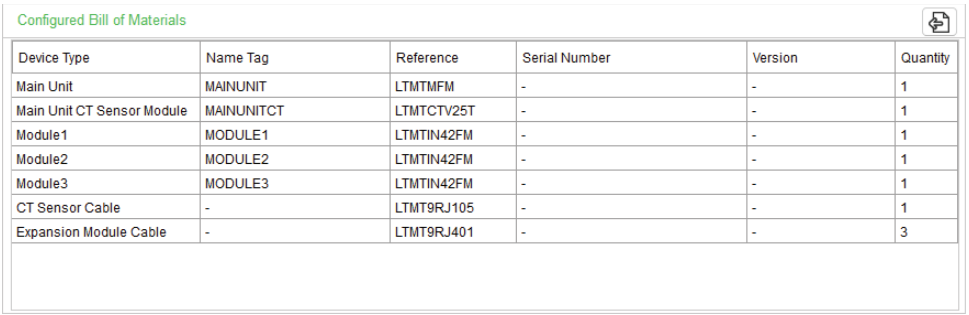
Characteristics	Description
Name Tag	Enter the name of the LTMT expansion unit. **NOTE:** The **Name Tag** field supports the following characters: • Alphabets: A to Z (upper and lower case) • Numbers: 0 to 9 • Symbols: (,), _, /, –, @, #, ' ' and (space)
Reference	Select the reference of the LTMT expansion unit.
Firmware Version	Displays the firmware version of the LTMT expansion unit. **NOTE:** The firmware version is displayed when the device is online.
Expansion Module Cable	Displays the reference of the cable used to connect the LTMT expansion unit.

Select **Add Expansion Module** card to add up to five LTMT expansion units.

NOTE:

A maximum of five LTMTIN42FM or LTMTIN42BD expansion unit can be added in a TeSys Tera system.

Configured Bill of Materials


Configured Bill of Materials

Device Type	Name Tag	Reference	Serial Number	Version	Quantity
Main Unit	MAINUNIT	LTMTMFM	-	-	1
Main Unit CT Sensor Module	MAINUNITCT	LTMTCTV25T	-	-	1
Module1	MODULE1	LTMTIN42FM	-	-	1
Module2	MODULE2	LTMTIN42FM	-	-	1
Module3	MODULE3	LTMTIN42FM	-	-	1
CT Sensor Cable	-	LTMT9RJ105	-	-	1
Expansion Module Cable	-	LTMT9RJ401	-	-	3

The **Configured Bill of Materials** displays the list of all the devices and accessories configured in the TeSys Tera system.

Based on the configuration made for the TeSys Tera system, the list of the modules are displayed in a tabular format.

NOTE: Serial Number and **Version** are displayed when the device is online.

Characteristics	Description
Device Type	Displays the type of device connected.
Reference	Displays the reference of the connected device or accessory.
Serial Number	Displays the serial number of the connected device. **NOTE:** The serial number is only displayed when the device is online.
Version	Displays the version number of the connected device. **NOTE:** The version number is only displayed when the device is online.
Quantity	Displays the number of devices, CT Sensor Cable, and Expansion Module Cable used.

The configured bill of materials can be exported as a CSV file by selecting the (**Export Bill of Materials**) icon .

Parameters List

What's in This Chapter

General Settings 50
Starter Function Setting 50
Protection Setting 51
Communication Setting 54
User Map 54
Modified Parameters 55

Overview

The **Parameters List** tab is used to define the parameter settings of the TeSys Tera system.

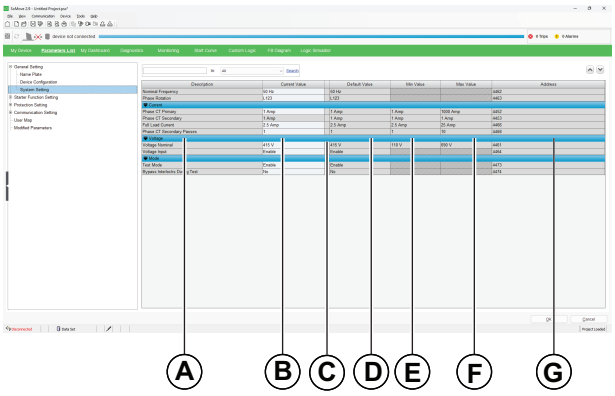
The parameter values can be modified in the **Current Value** field, both in connected and disconnected modes.

In connected mode, the settings can be updated to the device by selecting **OK**.

In disconnected mode, the configuration settings can be downloaded to the device by **Store to Device** operation.

> **NOTE:** The values in the **Parameter List** tab is updated for the first time only. To view the updated values in the **Parameter List** tab in online mode, perform refresh operation.

Description



A	Description: Name or Description of the parameter
B	Current Value: Value of the parameter
C	Wrench Mark: Appears if the current value is different from its default value
D	Default Value: Default value of the parameter
E	Min Value: Minimum value of the parameter
F	Max Value: Maximum value of the parameter
G	Address: Register address of the parameter

The following categories are available under the tree view in the **Parameters Lists** tab:

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

Setting Numerical Values

The numerical value of a parameter can be set in two ways:

- Direct entry of the numerical value.
- Value selection using the spin buttons.

The following steps describe how to set a numerical value by direct entry:

1. Navigate to the required section under the tree view.
2. Select a parameter from the list.
3. Type the parameter value in the **Current Value** field.
4. Select **OK** to write values to device in connected mode or to DTM offline configuration while in disconnected mode.

The following steps describe how to set a numerical value using the spin buttons:

1. Navigate to the required section under the tree view.
2. Select a parameter from the list.
3. In the **Current Value** field, set the parameter value using the spin buttons.
4. Select **OK** to write values to device in connected mode or to DTM offline configuration while in disconnected mode.

Editing a String

The following steps describe how to set a string parameter:

1. Navigate to the required section under the tree view.
2. Select a parameter from the list.
3. Type the string in the **Current Value** field.
4. Select **OK** to write values to device in connected mode or to DTM offline configuration while in disconnected mode.

Selecting Values in a List

The following steps describe how to select a value in a list:

1. Navigate to the required section under the tree view.
2. Select a parameter from the list.
3. In the **Current Value** field, set the parameter value using the drop-down list.
4. Select **OK** to write values to device in connected mode or to DTM offline configuration while in disconnected mode.

For a global description of the tab, refer to section Tab Zone description, page 39.

NOTE: After editing the default values from the parameters list and select **OK**, below warning message will appear. Ensure if the required measures are met, and select **OK**.

⚠WARNING

UNINTENDED MACHINE OPERATION

- Changing the Device Configuration can result in a short-circuit or turn on power supply to the load.
- Check if the appropriate wiring and configuration is done according to the Device Configuration.
- Ensure that the three phase power supply is cut off while changing the CTVT Sensor and Starter.

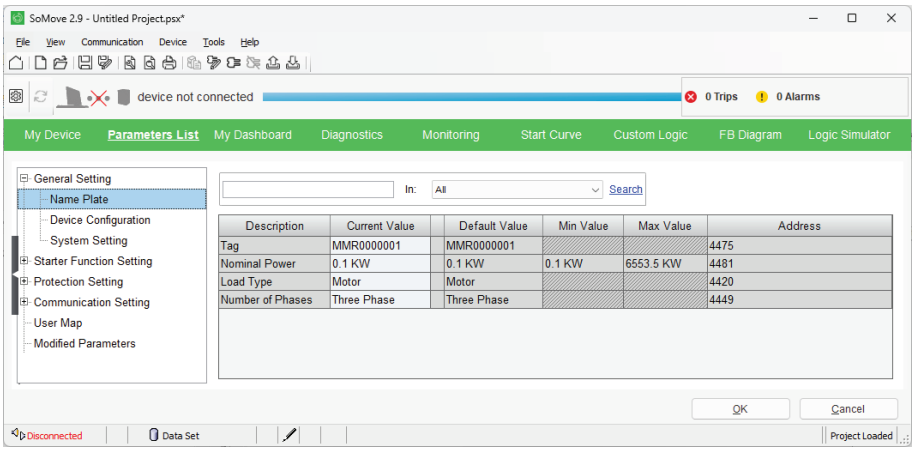
Failure to follow these instructions can result in death, serious injury, or equipment damage.

Important Information

- The parameters available for configuration are based on the TeSys Tera system configuration.
- If the parameter values entered are invalid, an error message will be displayed. Read the message and set the values appropriately.
- **Store to Device** operation cannot be performed when the motor is in run state.
- By default, the **Communication Loss** protection function is disabled. If this function is enabled and the device is connected to the SoMove software through the Modbus RTU communication port, and if you try to disconnect the DTM, the communication loss trip will be triggered in the device.
- Configuration of all setting under **Parameters List**, except for **User Map**, is disabled when the motor is in run state. In order to configure the parameters, stop the motor and then configure the settings.
- For the alarm to be triggered before the pickup level, the **Alarm Level** parameter configuration should be less than or equal to the **Pickup Level** parameter configuration.
- When configuring the start and stop of the motor in a system as a single relay, the DI Start **Validation Time** should be greater than the DI Stop **Validation Time**.
- To configure the parameters under the **Network Setting**, **Modbus Settings**, and **HMI Communication** sections, the device should be disconnected from the DTM. After performing a store to device operation to write the changed settings to the device, you can connect to the device by performing the **Scan Devices** using the new parameter values provided.
- If the **Main Unit Temperature** value is set to **None**, the **Control Panel** operations will not be available for configuration.
- If LTMTCUF control operator unit is connected on HMI port. The HMI Communication Settings must be configured as follows:
  - **Node Address**: 1
  - **Baud rate**: 19200 bps
  - **Parity**: Even
  - **Byte Format**: Big Endian
- If the **Starter Type** is configured as reversible, the **Interlocking Time** is not considered during a change of motor direction if the DI is selected as **Forced Start**.
- For more information on the individual parameters, refer to *TeSys Tera Motor Management System User Guide – DOCA0257EN*.

General Settings

The **General Settings** window contains the general settings of the TeSys Tera system.

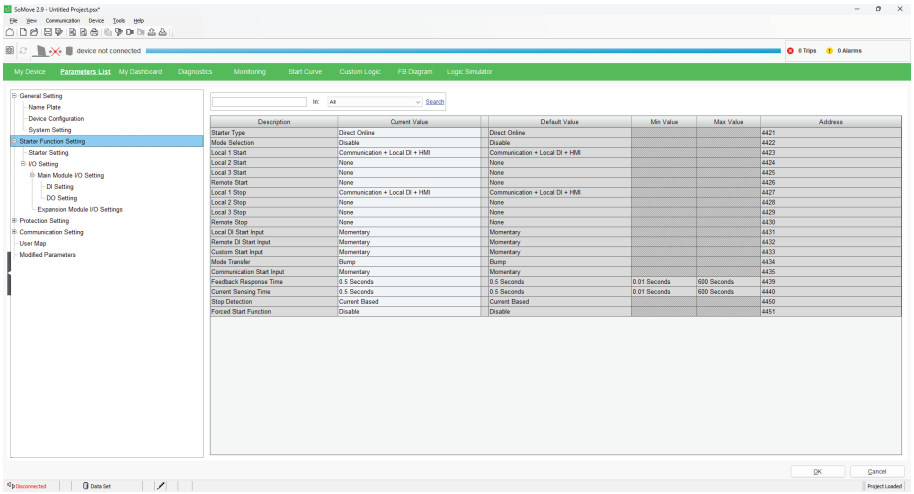


The following sub-sections are available:

- **Name Plate**: Contains the name plate parameters.
- **Device Configuration**: Contains the LTMT main unit temperature settings.
- **System Setting**: Contains the current, voltage, phase rotation, and mode parameters.

Starter Function Setting

The **Starter Function Setting** window contains the configurable parameters and options that control how the motor starter operates.

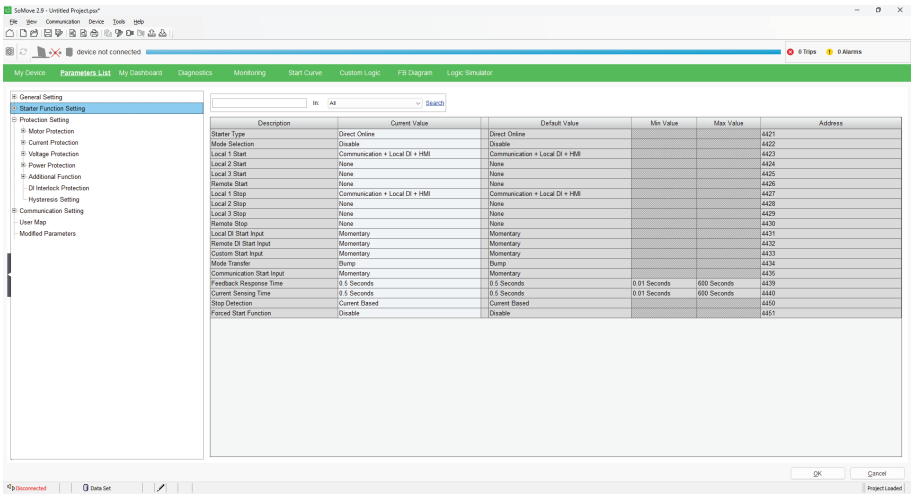


The parameters are divided into the following sub-sections:

- **Starter Setting**: Contains adjustable parameters associated with the motor starter.
- **I/O Setting**: Contains the digital input and output settings of the LTMTCT/ LTMTCTV sensor module and LTMT expansion units.

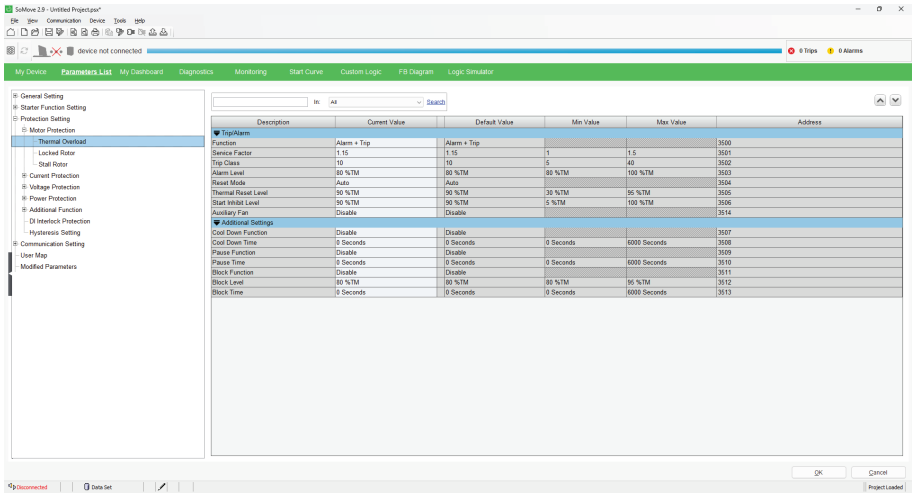
Protection Setting

The **Protection Setting** window contains the parameters to safeguard the TeSys Tera system.



The following sub-sections are available:

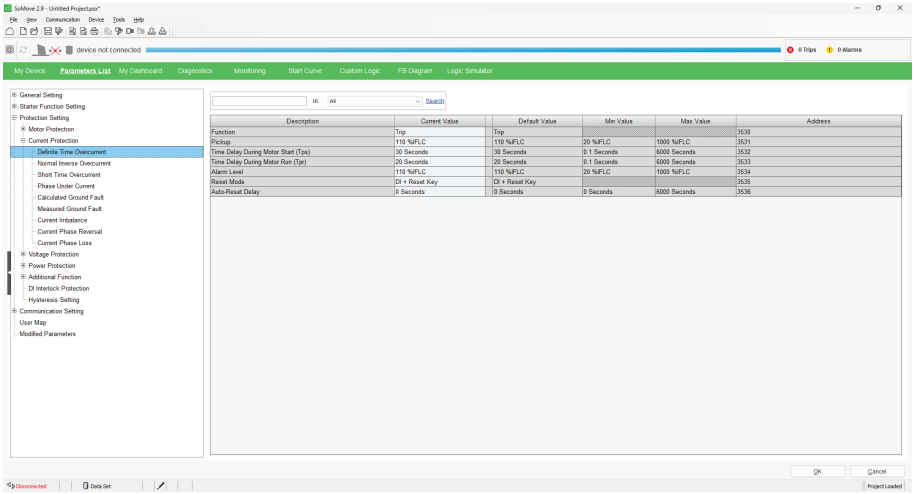
- **Motor Protection**: Contains the configurable motor protection parameters.



The motor protection parameters are classified into the following sub-sections:

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

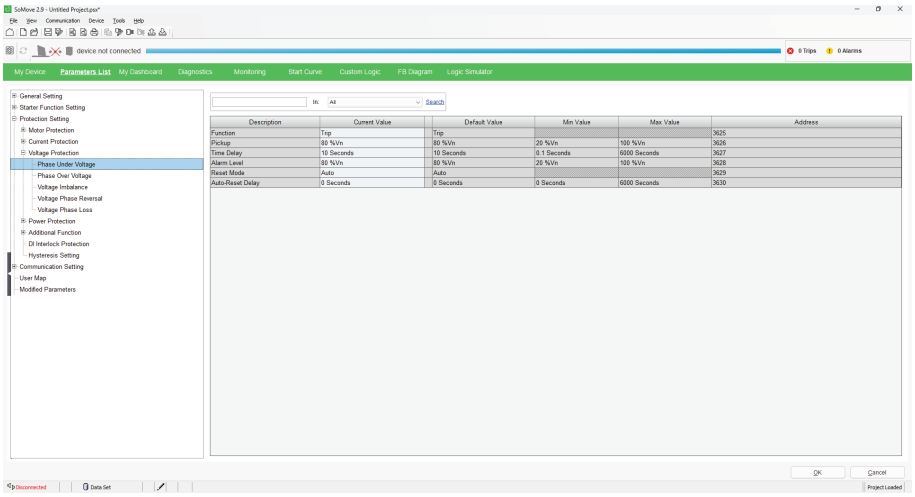
- **Current Protection**: Contains the configurable current protection parameters.



The current protection parameters are classified into the following sub-sections:

  - **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**

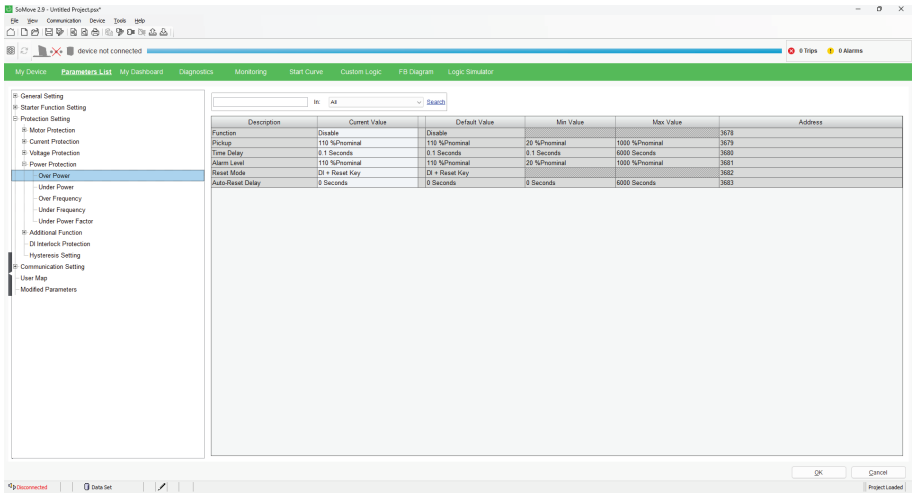
- **Voltage Protection**: Contains the configurable voltage protection parameters.



The voltage protection parameters are classified into the following sub-sections:

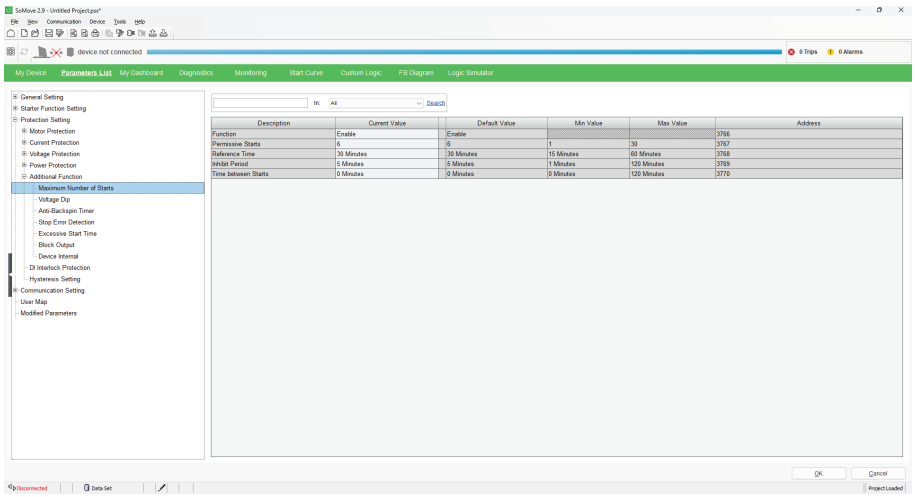
  - **Phase Under Voltage**
  - **Phase Over Voltage**
  - **Voltage Imbalance**
  - **Voltage Phase Reversal**
  - **Voltage Phase Loss**

- **Power Protection**: Contains the configurable power protection parameters.



The power protection parameters are classified into the following sub-sections:

  - **Over Power**
  - **Under Power**
  - **Over Frequency**
  - **Under Frequency**
  - **Under Power Factor**
- **Additional Function**: Contains the configurable additional function parameters.

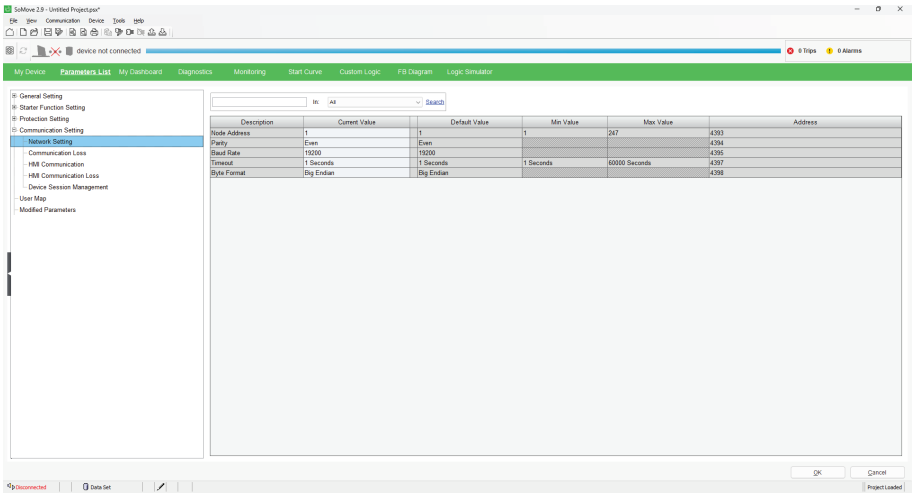


The parameters are classified into the following categories:

  - **Maximum Number of Starts**
  - **Voltage Dip**
  - **Anti-Backspin Timer**
  - **Stop Error Detection**
  - **Excessive Start Timer**
  - **Block Output**
  - **Device Internal**
- **DI Interlock Protection**: Contains the configurable digital input interlock protection parameters.
- **Hysteresis Setting**: Contains the configurable hysteresis parameters.

Communication Setting

The **Communication Setting** window contains the communication parameters.

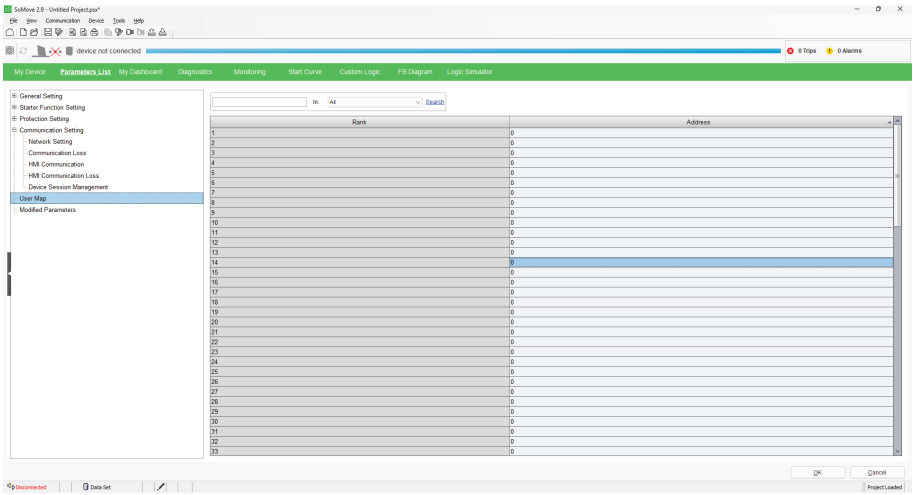


The parameters are divided into the following sub-sections:

- **Network Setting**: This window contains the parameters for the network settings of the TeSys Tera system.
- **Communication Loss**: This window contains parameters to configure the communication loss settings.
- **HMI Communication**: This window contains the parameters for the network settings of the LTMTCUF control operator unit.
- **HMI Communication Loss**: This window contains parameters to configure the communication loss settings of the LTMTCUF control operator unit.
- **Device Session Management**: This window contains parameter to configure the device session time out.

User Map

The **User Map** variables are designed to optimize the read and write registers.



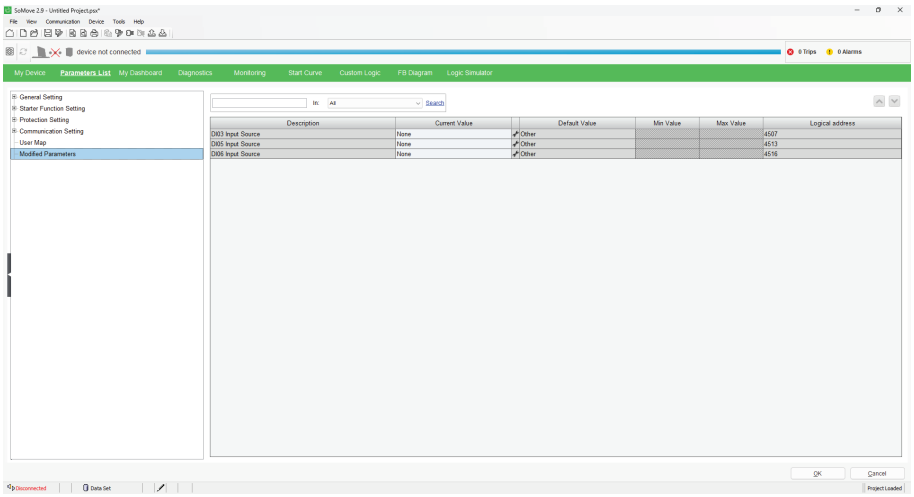
The following procedure describes how to set user map addresses:

1. Select **User Map** in the tree view:
   - Addresses are ranked from 1 to 100.
   - Addresses are divided into four groups.

2. Enter an address value in the table:
   - The entered address must be in the decimal format and should be from 0 to 9249.
   - Enter the address 0 to remove the address from the user map.
3. Press ENTER to validate the new address:
   - If the address is accepted, the address is added to the user map.
   - If the address is not accepted, the previous accepted address is kept in the user map.

Modified Parameters

The **Modified Parameters** window displays a list of the parameters whose values have been changed from the default settings.



My Dashboard

Overview

The **My Dashboard** tab allows you to customize data to configure or monitor based on requirement. It is used to:

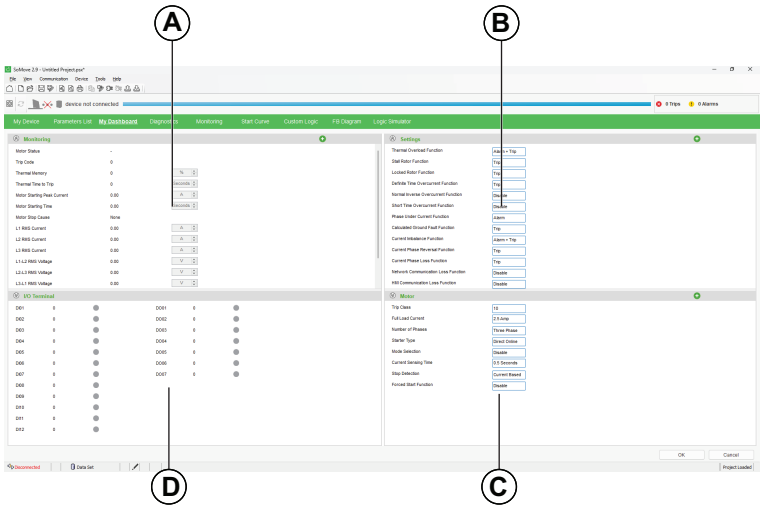
- Display the real-time value of the starter related parameters.
- Display the information of input/output terminals.
- Configure the starter and motor related parameters.
- Configure the protection functions parameters.

NOTE: My Dashboard settings can be saved using **Save Project** option. The settings are retained if you save the project and open the saved project.

Description

The working space is divided into two zones:

- **Display zone**: Displays the real-time value of starter parameters and status of input/output terminals for analysis.
- **Settings zone**: Set the starter and motor parameters.



A	Monitoring
B	Settings
C	Motor
D	I/O terminals

Monitoring

The **Monitoring** area allows you to monitor the required parameters.

Settings

The **Settings** area allows you to monitor and change the protection function parameter values as per requirement.

The parameters under the **Settings** area can be assigned the required values by using the drop-down list.

Motor

The **Motor** area allows you to monitor and change the motor parameter values as per requirement.

I/O Terminals

The **I/O Terminals** area allows you to monitor the status of the I/O signals.

The following table describes the status of the input/output of the LTMT main unit and LTMT expansion modules with digital inputs/outputs.

Status input/output	Color status box	Descriptive text
Active	Green	Indicates that the IO terminal is assigned and active.
Inactive	Grey	Indicates that the IO terminal is assigned and not active.

Adding Parameters

The following procedure shows how to add parameters:

1. Select icon in the area under which parameters need to be added.

   Result: The **Select Parameter** dialog appears.
2. In the **All Parameters** column, select the required parameters and select the **Right arrow**. The search bar can be used to find parameters available under the **All Parameters** column.

   Result: The parameters are moved to the **Selected Parameters** column.
3. To reorder the list of parameters, select the required parameter in the **Selected Parameters** column and select **Up/Down arrow**.
4. Select **OK**.

The parameters added are displayed in the respective display area.

Deleting Parameters

To remove a parameter from an area, select icon in front of the parameter.

The following procedure shows how to remove multiple parameters:

1. Select icon in the area under which parameters need to be removed.

   Result: The **Select Parameter** dialog appears.
2. In the **Selected Parameters** column, select the required parameter and select the **Left Arrow**.

   Result: The parameters are moved to the **All Parameters** column.
3. Select **OK.**

The parameters are removed from the respective display area.

NOTE: The list of parameters is configurable only under **Settings** and **Motor** areas.

Diagnostics

Overview

The **Diagnostics** tab displays the motor status, alarm or trip status, logs, device internal status, communication status and, trip counters information of the TeSys Tera system.

The data in this tab is only significant in the connected mode.

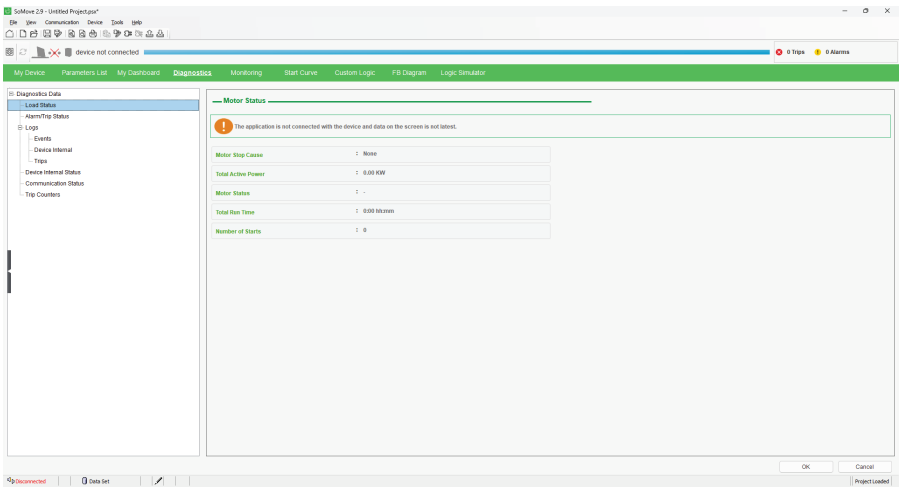
The data in the **Diagnostics** tab is dynamic and reads in every 1 s except the Data Record node.

Description

The following parameters are displayed in the **Diagnostics** tab:

- **Motor or Load Status**
- **Alarm or Trip Status**
- Logs of the TeSys Tera system which includes:
  - **Events**
  - **Device Internal**
  - **Trips**
- **Device Internal Status**
- **Communication Status**
- **Trip Counters**

Motor or Load Status



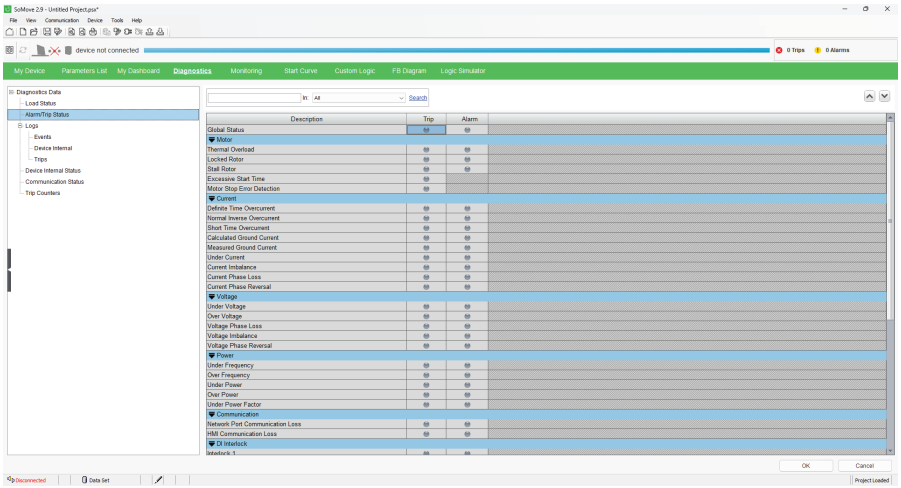
The **Motor Status** or **Load Status** window displays the parameters of the configured motor or heater respectively.

The following motor parameters are displayed under the **Motor Status** or **Load Status** window:

- **Motor Stop Cause**: Reason for stopping the motor operation.
- **Total Active Power**: Total power consumed during the motor or heater operation.
- **Motor Status** or **Load Status**: Current status of the motor or heater.

- **Total Run Hour**: Total operational time of the motor or heater.
- **Number of Starts**: Total number of starts of the motor or heater.

Alarm or Trip Status



The **Alarm or Trip Status** window displays the status of motor parameters with respect to alarm or trip.

The required parameter can be searched by adding keywords in the search box, selecting the required option under the **In** drop-down list and by selecting **Search**.

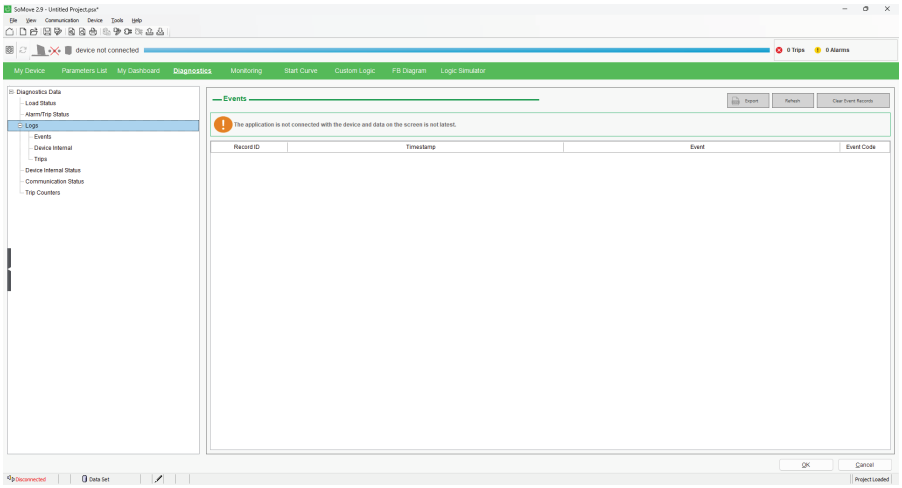
Color of the Indicator		**Status**
Grey		Not Active
Amber		Alarm
Red		Trip

Logs

The **Logs** window displays the logs for the following categories:

- Events
- Device Internal
- Trips

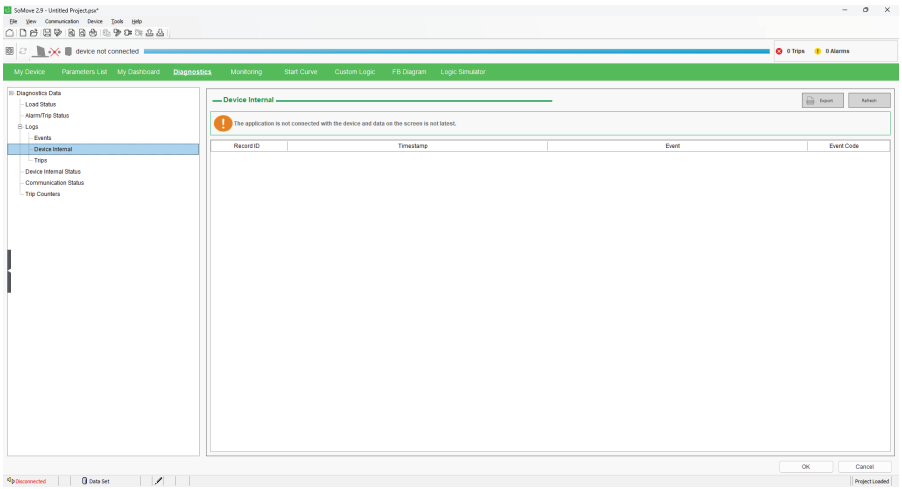
Events



The **Events** window displays the list of event records for the TeSys Tera system.

Up to 100 event logs with timestamp, event description, and event code will be displayed on the window. Serial number 1 is the latest event recorded and serial number 100 is oldest event recorded. If there are 100 event logs on the window and if a new event occurred, then the oldest event will be removed from the list and the new event will be added at the top of the list.

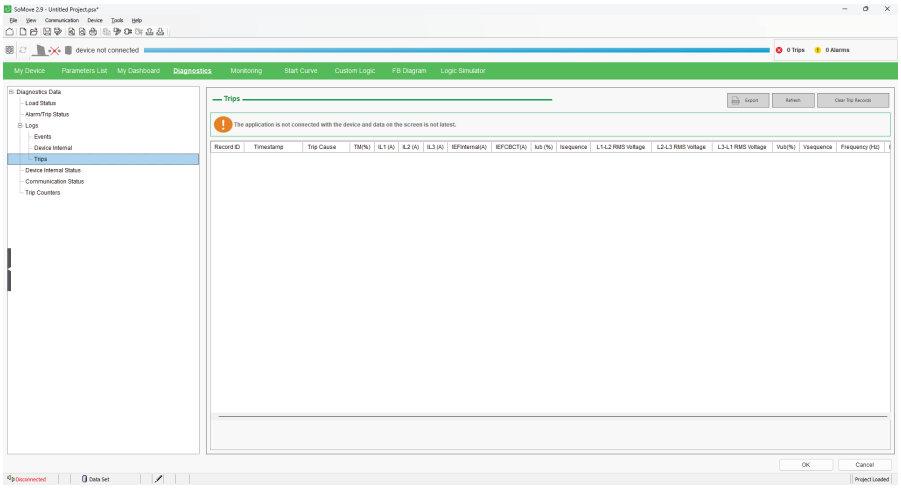
Device Internal



The **Device Internal** window provides the list of internal relay events of the TeSys Tera system.

Up to 20 internal relay logs with timestamp, IRM description, and IRM code will be displayed on the window. Serial number 1 is the latest detected malfunction recorded and Serial number 20 is oldest detected malfunction recorded. If there are 20 malfunction on the window and if a new malfunction is detected, then the oldest detected malfunction will be removed from the list and the newly detected malfunction will be added at the top of the list.

Trips



The **Trips** window provides the list of trip records of the TeSys Tera system.

Up to 20 trip logs with timestamp, trip cause, value of important settings when the trip occurred, value of measurements recorded when the trip was detected and trip code will be displayed on the window. Serial number 1 is the latest trip recorded and serial number 20 is oldest trip recorded. If there are 20 trip records on the window and if a new trip is detected, then the oldest trip record will be removed from the list and the new trip record will be added at the top of the list.

Export

The **Export** function exports the list of records in *.csv* format.

To export the list of records, select **Export**, select the destination folder, and select **Save**.

NOTE: The export option is available only when there is at least one record data.

Refresh

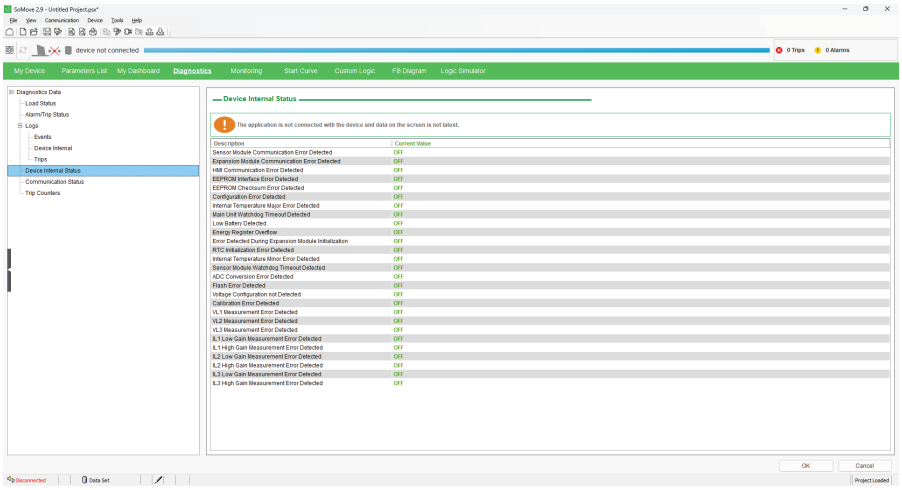
The **Refresh** function updates the list of records and displays the latest list.

The Data Record node can be updated by selecting **Refresh** in the connected mode.

Clear Event Logs or Clear Trip Logs

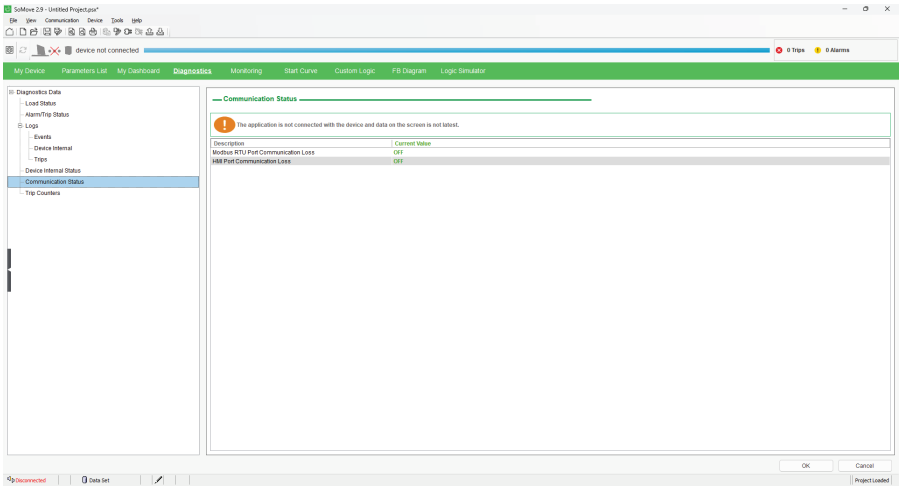
The **Clear Event Logs** or **Clear Trip Logs** clears the list of events or trips by selecting **Clear Event Logs** or **Clear Trip Logs** in the respective windows.

Device Internal Status



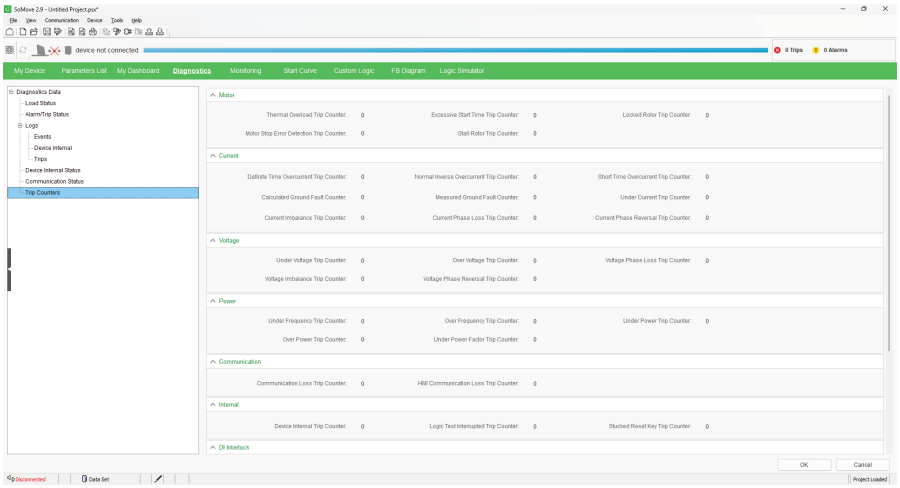
The **Device Internal Status** window displays the communication and operation status of the different modules of the TeSys Tera system.

Communication Status



The **Communication Status** window displays the status of the communication parameter of the TeSys Tera system.

Trip Counters



The **Trip Counters** window displays the number of trips per category within the TeSys Tera system.

Monitoring

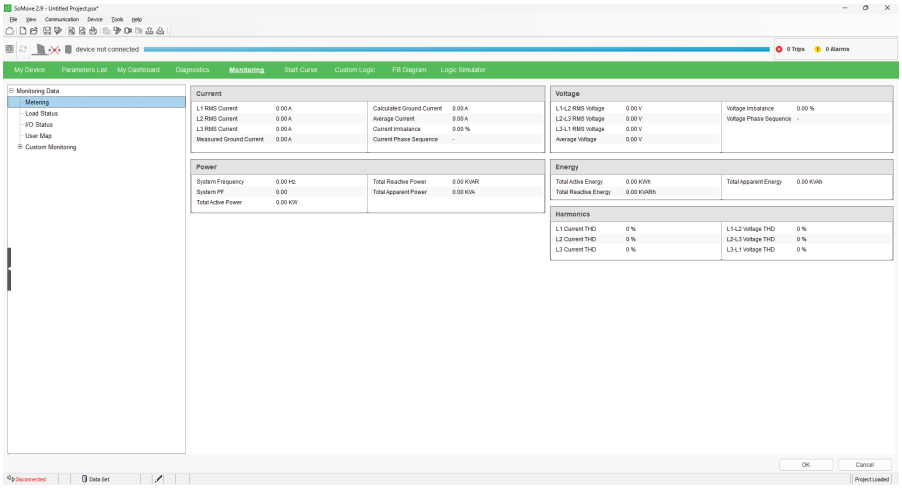
Overview

The **Monitoring** tab displays the monitoring of information related to the TeSys Tera system.

The monitoring information for the following categories are displayed:

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

Metering



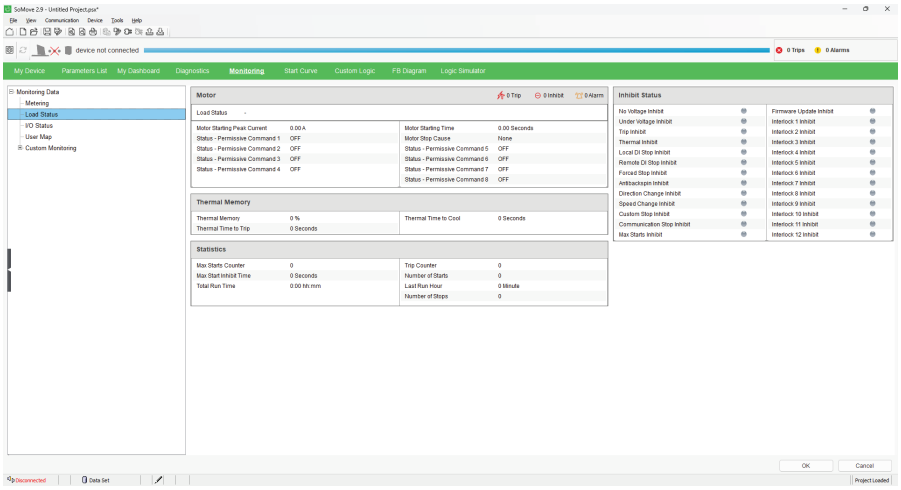
The **Metering** window displays the metering data of the TeSys Tera system.

The metering signals are divided into the following categories:

- Current
- Voltage
- Power
- Energy
- Temperature
- Harmonics

NOTE: The signal values populated in the **Metering** window depends on the configuration made in the **My Device** tab and **General settings** under **Device Configuration**.

Load Status



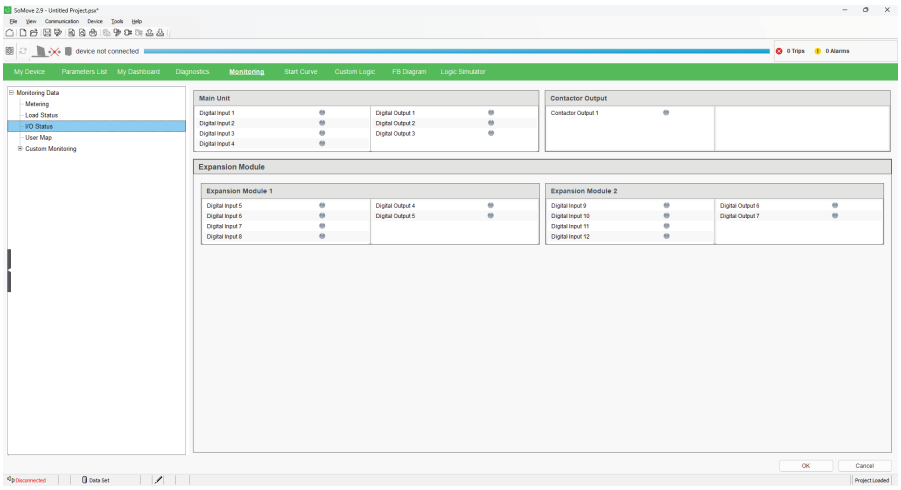
The **Load Status** window displays the load status of the TeSys Tera system.

The parameters under the load status page are divided into the following categories:

- Motor
- Thermal Memory
- Statistics
- Inhibit Status

The **Load Status** window also displays the total number of trips, inhibits and alarms under **Motor**.

I/O Status

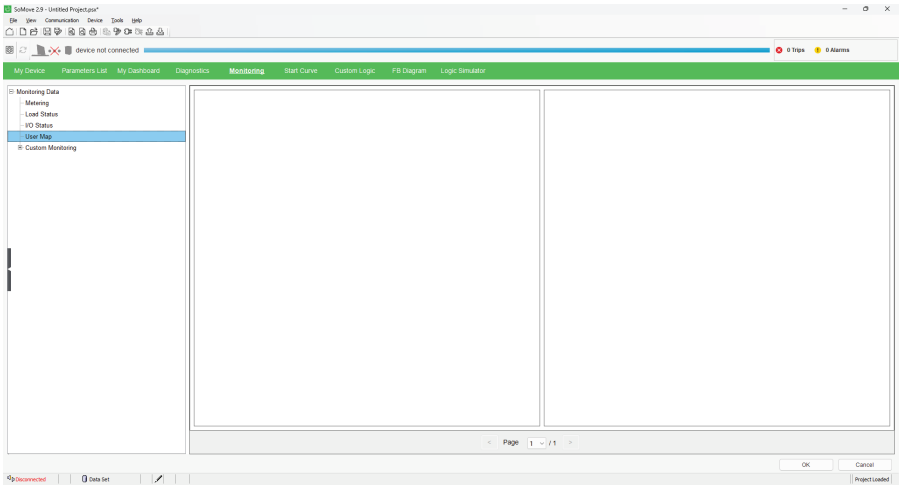


The **I/O Status** window displays the status of the digital inputs and outputs of the TeSys Tera system.

The I/O signals are divided into the following categories:

- Main Unit
- Contactor Output
- Expansion Module

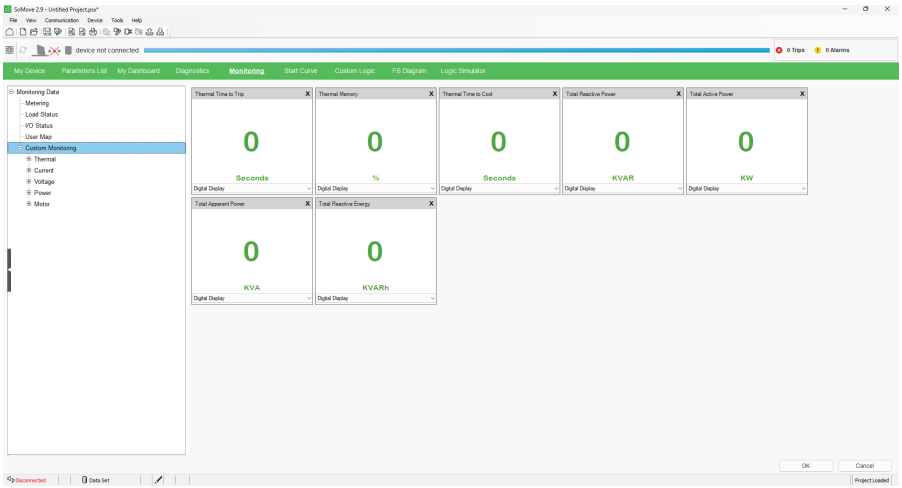
User Map



The **User Map** window displays the live value of the parameters configured under **Parameters > User Map** section.

A maximum of 100 parameters are displayed.

Custom Monitoring



The **Custom Monitoring** window allows you to create a personalized monitoring window with widgets.

The signals are divided into the following categories:

- Thermal
- Current
- Voltage
- Power
- Motor

Customizing the Custom Monitoring Window

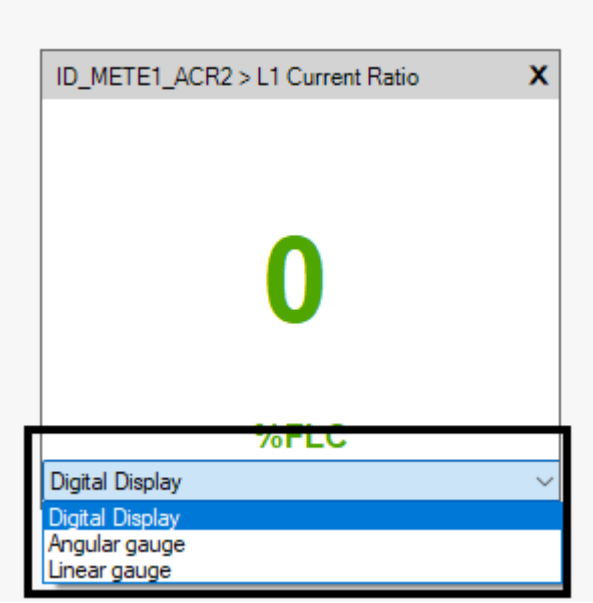
The following procedure describes how to add a widget:

1. Select the signal to be monitored that are available under the available signal categories.

2. Click on the display area of the **Custom Monitoring** window.

   Result: The Widget for the selected signal is added.

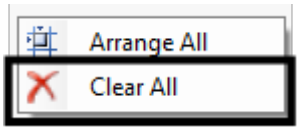
You can modify the widget display options by selecting option from the drop-down list.



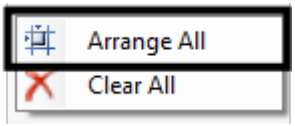
To remove a widget from the display area, select X icon.

The following actions can be performed on the display area:

- To remove all the widgets, right-click on the display area and select **Clear All**.



- To organize the added widgets, right-click on the display area and select **Arrange All.**

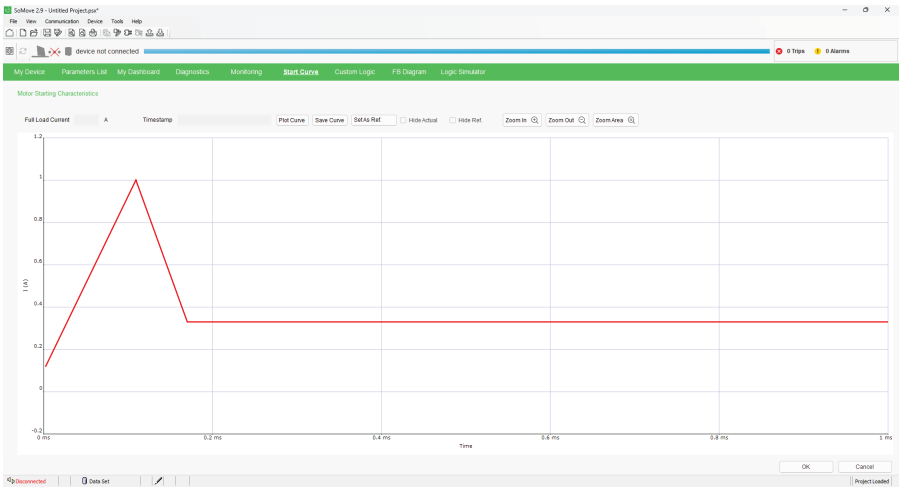


Start Curve

Overview

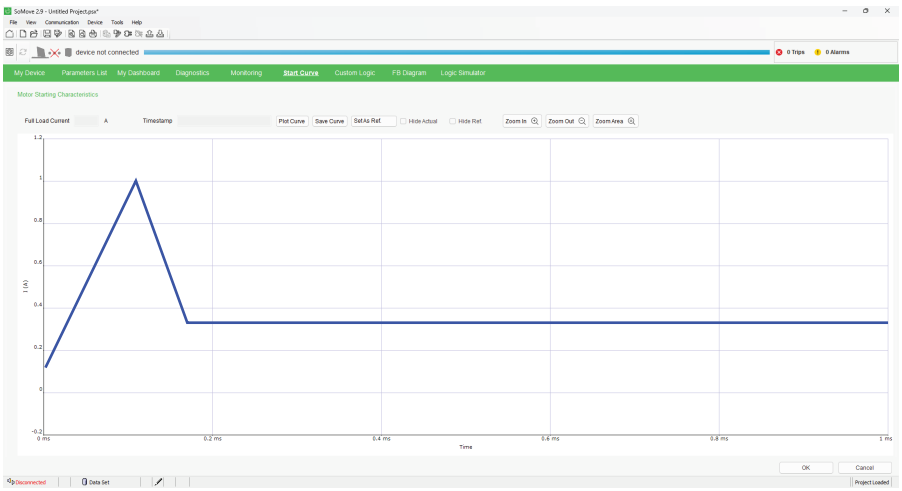
The **Start Curve** tab displays the graph of the motor current (I) during the motor starting time and at sample time intervals. The **Start Curve** tab is used to analyze the motor starting characteristics from the curve.

The start curve data will be erased from the device after performing a factory reset.



Reference Curve

The plotted curve can be set as the reference curve by clicking **Set as Ref.**. The reference curve will remain the same until a new curve is set as reference. The blue graph line represents the Reference curve.



Zoom Display Options

The following options are available:

- **Zoom Out** to see more program at once.
- **Zoom In** to focus on the specific program in detail.

- **Zoom Area** to have a customized view of the program.

Key Buttons

- **Save curve**: Captures the screen shot of the motor start characteristics curve and allows you to save it to the PC.
- **Plot Curve**: Reads the latest values and refreshes the motor start characteristics curve.
- **Time Stamp**: Shows the motor start time.
- **Set As Ref. Curve**: Sets actual curve as reference curve.

Custom Logic

Overview

The **Custom Logic** tab displays the custom logic editor which allows you to create a new custom logic program file or to open and edit any existing custom logic program file.

For more information about **Custom Logic** tab, refer to Custom Logic Editor, page 81

FB Diagram

Overview

The **FB Diagram** tab is used to display the functional block diagram editor which allows you to create a new functional block diagram file or to open and edit any existing functional block diagram files.

For more information about **FB Diagram** tab, refer to Introduction to FBD Editor, page 149

Logic Simulator

Overview

The **Logic Simulator** tab is used to display the LTMT main unit **Logic Simulator** which enables you to test the functioning of a custom logic program.

For more information about the Logic Simulator, refer to LTMT Main Unit Logic Simulator, page 181.

User Functions

What's in This Part

Firmware Update 74
Factory Reset 77
Pin Management 78

Firmware Update

Overview

The firmware of the TeSys Tera system can be updated using one of the following sources:

- **Server**
- **PC**

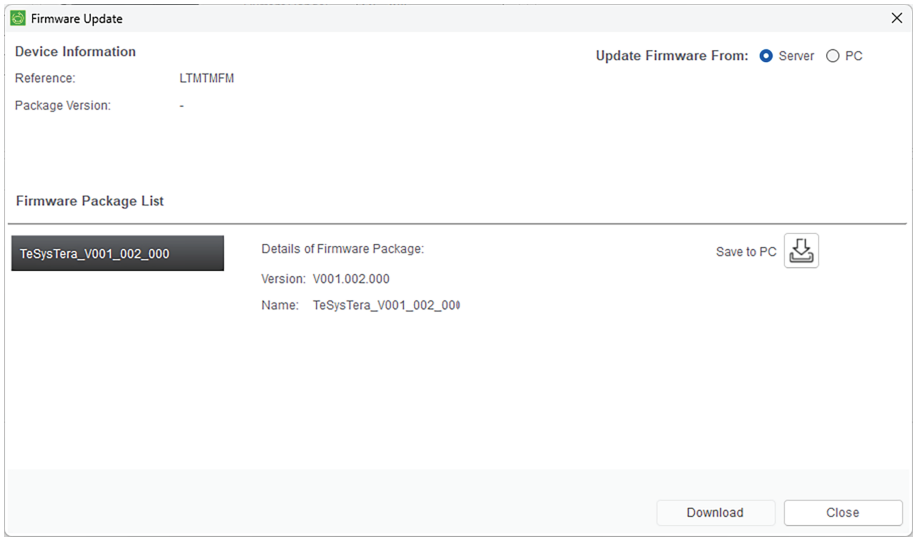
NOTE:

- To update the firmware from the **Server**, you must have an internet connection.
- It is not possible to downgrade the firmware from *TeSysTera_V001_003_000* to *TeSysTera_V001_000_000*.

Accessing the Firmware Update

On the menu bar, select **Device > Maintenance > Firmware Update**.

Result: The **Firmware Update** window appears.



Updating the Firmware from the Server

Perform the following steps to update the firmware from the server:

1. On the **Firmware Update** window, select **Server** in the **Update Firmware From** option.
2. Select the firmware package from the **Firmware Package List** and select **Download**.

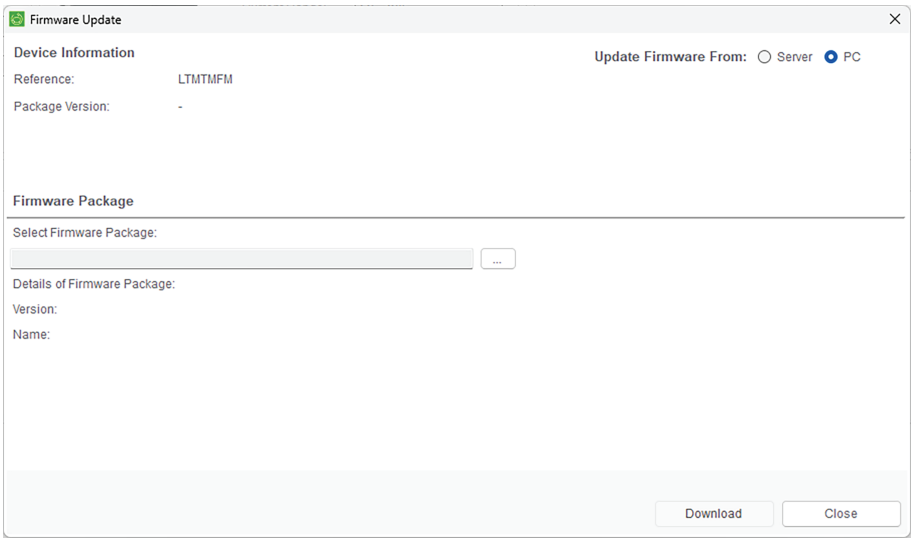
   NOTE: This operation takes 4 to 12 minutes to update the firmware. Select the highest baud rate supported by the device to update the firmware faster.

   Result: An update successful message is displayed at the bottom of the **Firmware Update** window.

Downloading the Firmware to PC

Select icon to download the firmware package and save it to your **PC**.

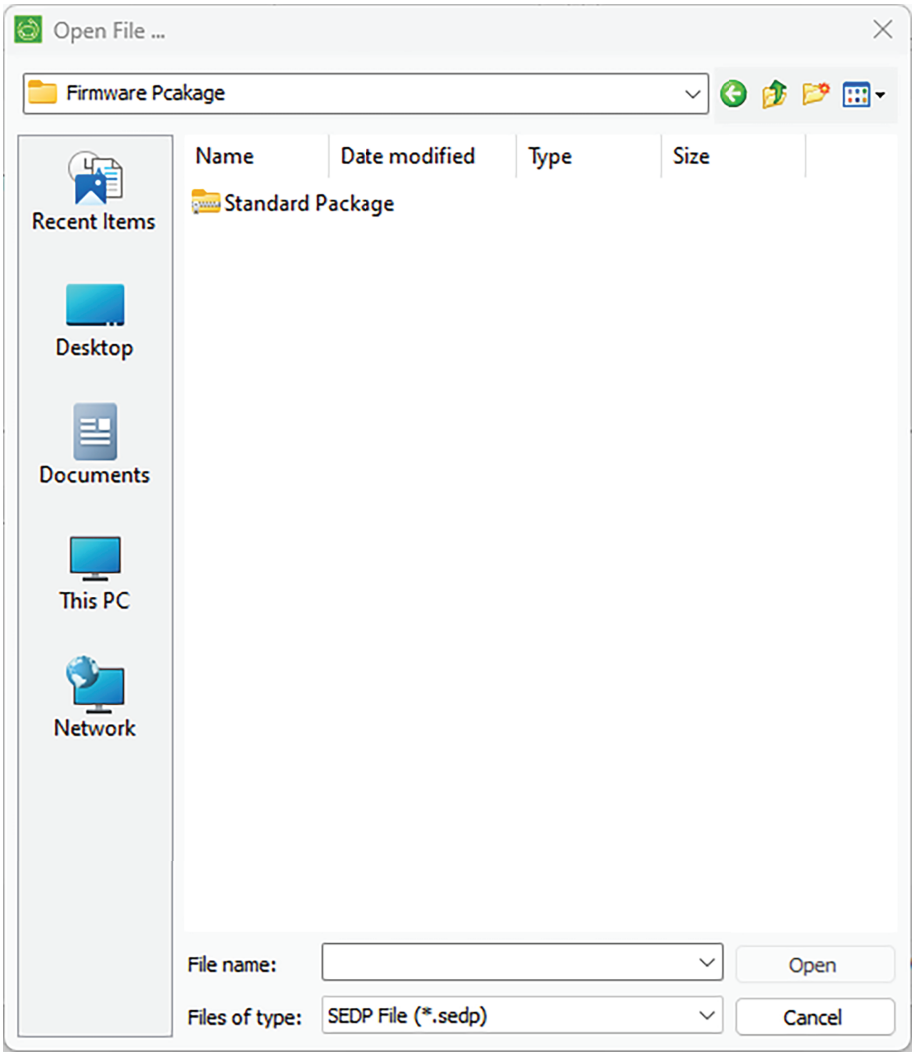
Updating the Firmware from the PC



To update the firmware from **PC**, perform the following steps:

1. On the **Firmware Update** window, select **PC** in the section.

2. Select the ... icon in the **Select Firmware Package** to browse and select the firmware package file from the PC as shown below:



3. Select **Open**.

   Result: An update successful message is displayed at the bottom of the **Firmware Update** window.

4. In the event of a failed update, a **Firmware Update Failed** dialog will appear. Perform the above procedure again to update the firmware.

Factory Reset

Overview

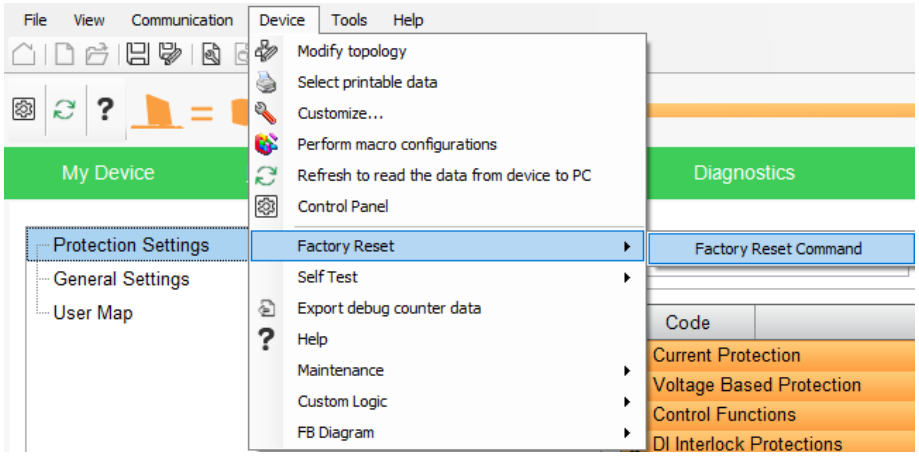
The **Factory Reset** function allows you to reset the TeSys Tera system parameters to its default values.

Performing Factory Reset Command

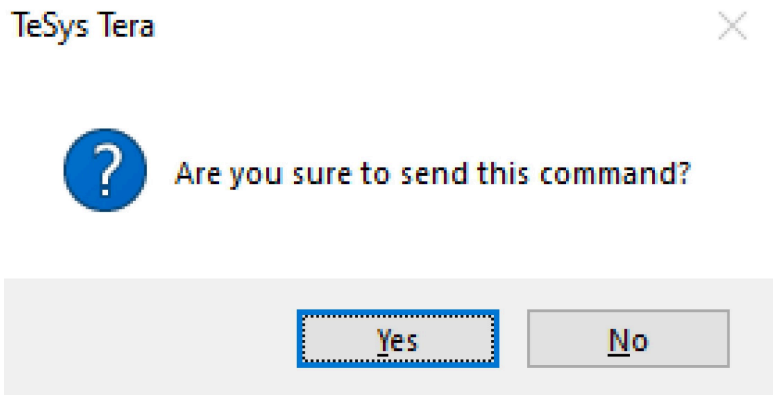
To performing factory reset command, follow the below steps:

NOTE: Factory reset function works only in connected mode.

1. Select **Device > Factory Reset > Factory Rest Command** as shown in the image below.



2. A confirmation pop up will appear after selecting the command.



NOTE: The TeSys Tera DTM Library will be disconnected from the TeSys Tera system after execution of this command.

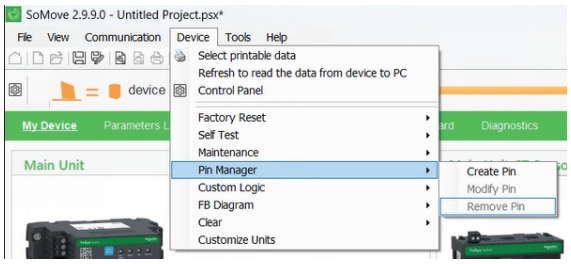
When the **Factory Reset** command is executed, the function restores the configuration of the TeSys Tera system to the factory settings and restarts the TeSys Tera system.

Pin Management

The **Pin Manager** secures the device by allowing you to set a pin on SoMove software.

The **Pin Manager** can be accessed by navigating to **Device > Pin Manager**.

NOTE: The **Pin Manager** is available only when the device is connected to SoMove software.



The following actions can be performed under the **Pin Manager**:

- Create Pin
- Modify Pin
- Remove Pin

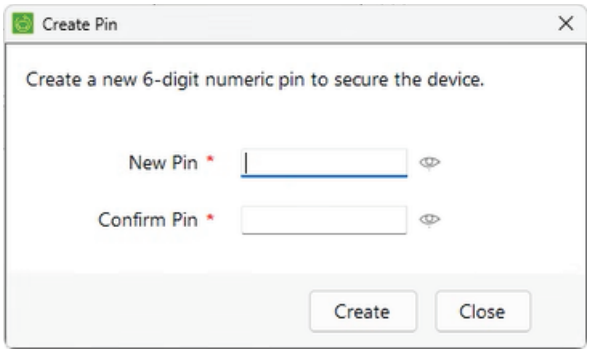
Creating a Pin

The **Create Pin** option allows you to create a new pin. The **Create Pin** option is available only when a pin is not set.

The following procedure shows how to create a pin:

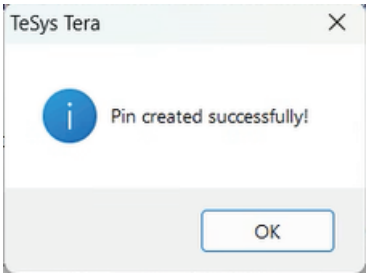
1. Navigate to **Device > Pin Manager > Create Pin** and select **Create Pin**.

   Result: The **Create Pin** dialog appears.



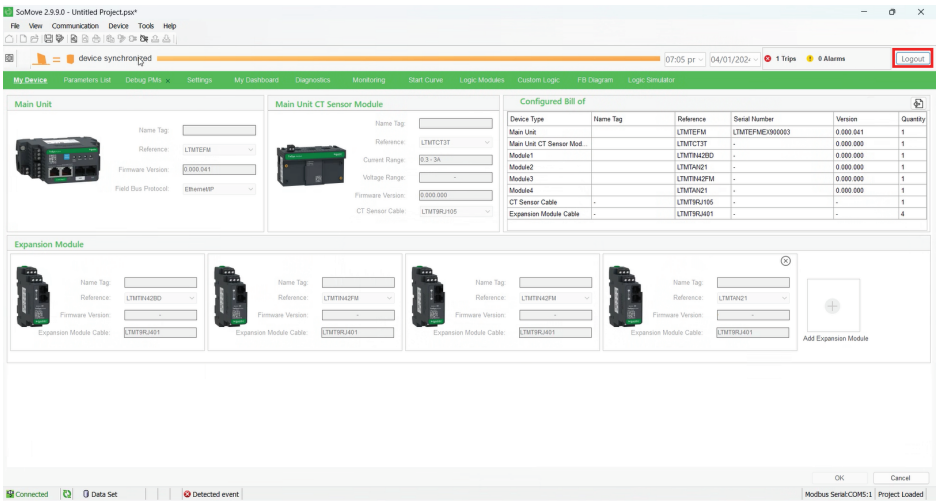
2. Enter a new six digit numeric pin in the **New Pin** and **Confirm Pin** fields.
3. Select **Create**.

   Result: The pin is created.



You can set the session time using the **Device Session Timeout** parameter in the **Parameter List > Device Session Management** tab, a minimum of 5 minutes and a maximum of 60 minutes is allowed for each session.

Select **Logout** to disconnect the device from SoMove software.



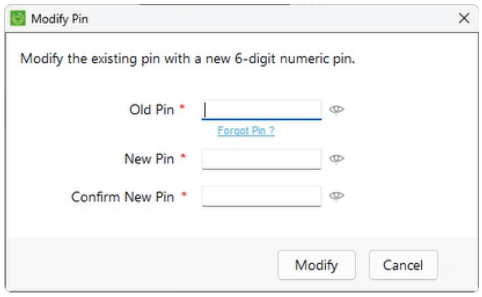
Modify Pin

The **Modify Pin** option allows you to modify the pin. The **Modify Pin** option is available only after a pin is set.

The following procedure shows how to modify the pin:

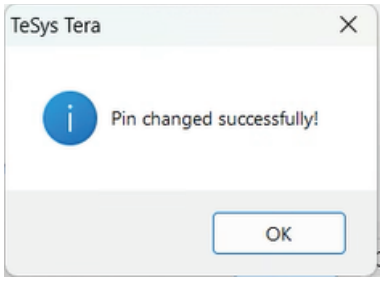
1. Navigate to **Device > Pin Manager > Modify Pin**

   Result: The **Modify Pin** dialog appears.



2. Enter the values for the **Old Pin**, **New Pin**, and **Confirm New Pin** respectively.
3. Select **Modify**.

   Result: The pin is modified.



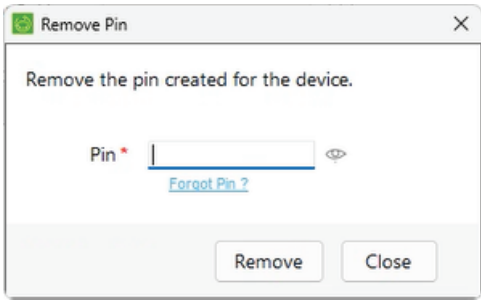
Delete Pin

The **Remove Pin** option allows you to delete the existing pin. The **Remove Pin** option is available only after a pin is set.

The following procedure shows how to remove the pin:

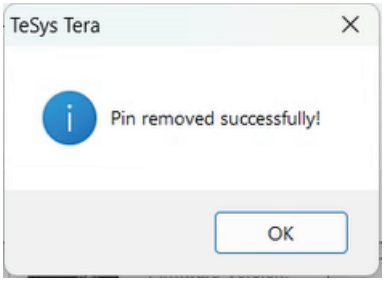
1. Navigate to **Device > Pin Manager > Remove Pin**

   Result: The **Remove Pin** dialog appears.



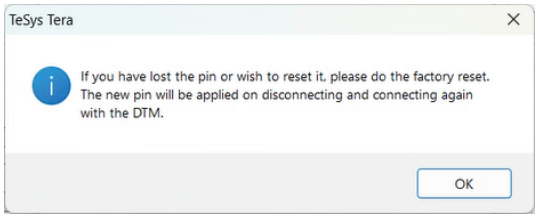
2. Enter the existing pin in the **Pin** field.
3. Select **Remove**.

   Result: The pin is removed.



Forgot Pin

In the event of forgetting the pin, a factory reset should be performed using the **Reset** button available on the device.



NOTE: When performing factory reset, TeSys Tera DTM gets disconnected from the device.

Custom Logic Editor

What's in This Part

Presentation of the Custom Logic Editor....82
Using the Custom Logic Editor....85
Characteristics of the Custom Logic Program....87
Definition of the Custom Logic Variables....88
Definition of LTMT Main Unit Variables....89
CALL_EOM Command Description....92

Presentation of the Custom Logic Editor

Overview

You can customize LTMT main unit pre-defined control programs using the custom logic editor. The custom logic editor is a powerful programming tool that is only available in SoMove software with the TeSys Tera DTM Library. Creating a customized control program for an LTMT main unit consists of writing a series of instructions (logic commands) in one of the custom logic programming languages.

Purpose of Custom Logic Editor

The primary purpose of custom logic editor is to provide you the flexibility to modify the pre-defined logic for the starter types or add additional logic functionality.

Logic ID

There is one pre-defined control program for each LTMT main unit operating mode (or motor starter type). The operating mode (or motor starter type) programs are identified with a unique logic ID. The logic ID of pre-defined operating mode program are numbers from 12 to 22. When a pre-defined operating mode program is customized, the logic ID of the customized program must be equal to the logic ID of the pre-defined program + 256.

This table gives the logic ID according to the operating mode:

Operating mode/Starter type	Logic ID of pre-defined program	Logic ID of customized program
Reserved	0–11	–
Overload	12	268
Direct Online	13	269
Reverse Direct Online	14	270
Star Delta	15	271
Reverse Star Delta	16	272
Dahlander	17	273
Reverse Dahlander	18	274
Pole Changing	19	275
Reverse Pole Changing	20	276
Circuit Breaker Control	21	277
Soft Starter Control	22	278
Reserved	23–255	–
Customized program	–	256–267, 279–511

Customized Programs

A customized program is an LTMT main unit pre-defined program with specific functions to meet individual application needs.

When configured with one of the pre-defined programs, the LTMT main unit manages the control functions using both the firmware in the LTMT main unit and the PCode.

NOTE: PCode (Pseudo Code) is a set of error less hexadecimal instructions.

When configured with a customized program, the LTMT main unit retains the functions controlled by the LTMT main unit. Those functions include the following characteristics that are inherent to the parent pre-defined program:

- Restrictions to what can be written to register at address 704 (0x02C0) (network command register.)
- Display of the operating state in presentation mode (for example: Forward or Reverse, Low Speed or High Speed.)
- Restrictions on the fallback modes that may be set through the menus.
- Specific behaviors regarding the start cycle in Star Delta and Reverse Star Delta mode.
- Restrictions on the transition timer that may be set through the menus.

Pre-Defined Program Structure

There are 11 pre-defined programs available with the TeSys Tera DTM Library on SoMove software, one pre-defined program for each operating mode (or motor starter type).

The pre-defined programs performs the following parts, one after the other:

- Logic identification of the program with the logic ID
- Input management
- Operating mode execution
- Output update

The execution of the operating mode is embedded and called with the function `CALL_EOM`.

This provides the possibility to customize the input and output management of your custom program without modifying the operating mode execution.

Custom Logic Editor Programming Languages and Tools

The custom logic editor provides two programming languages and tools:

- Custom logic language — is a list instruction language editable through the custom logic editor programming tool.
- Function Block Diagram (FBD) — is an object-oriented programming language editable through the FBD editor programming tool.

Each programming method satisfies your programming objectives. However, the custom logic editor allows you to select the style of programming method that you prefer.

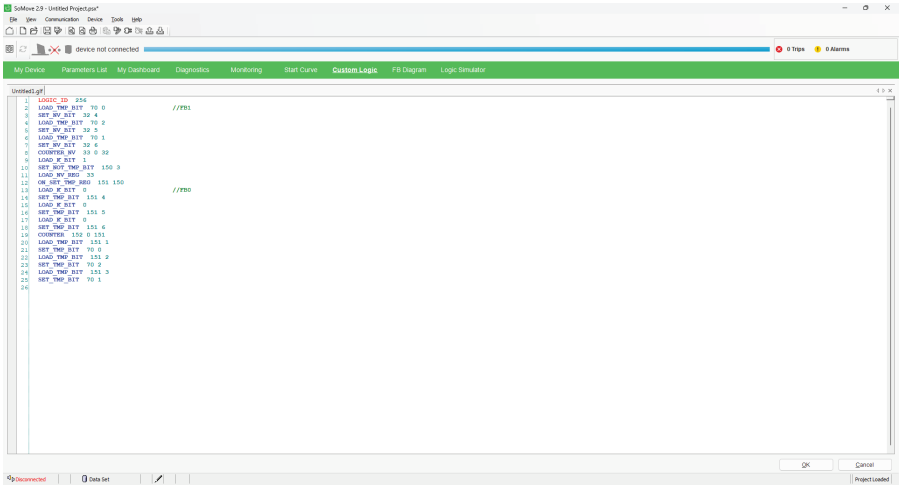
Logic Commands

Both custom logic and FBD languages implement the following types of commands:

- Program logic commands
- Boolean logic commands
- Register logic commands
- Timer logic commands
- Counter logic commands
- Latch logic commands
- Math logic commands

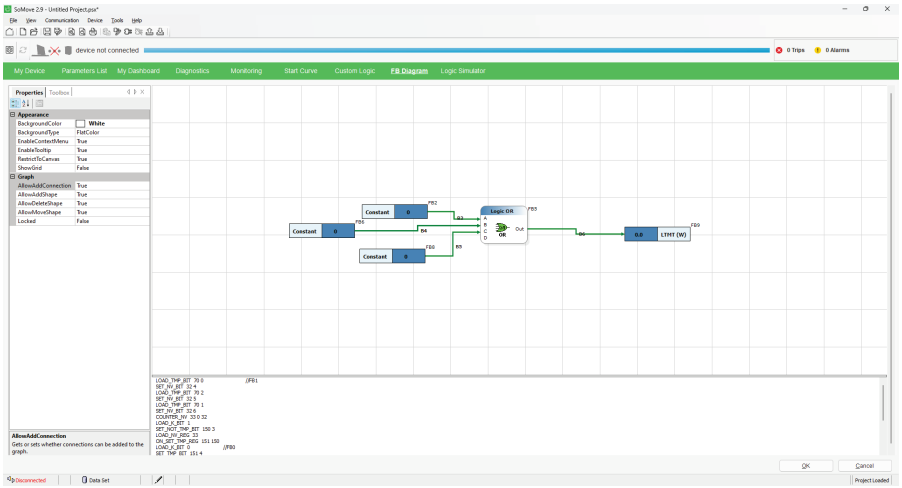
Custom Logic Editor

The following illustration shows the custom logic editor, integrated in the TeSys Tera DTM Library:



Function Block Diagram Editor

The following illustration shows the Function Block Diagram (FBD) editor, integrated in the TeSys Tera DTM Library:



Using the Custom Logic Editor

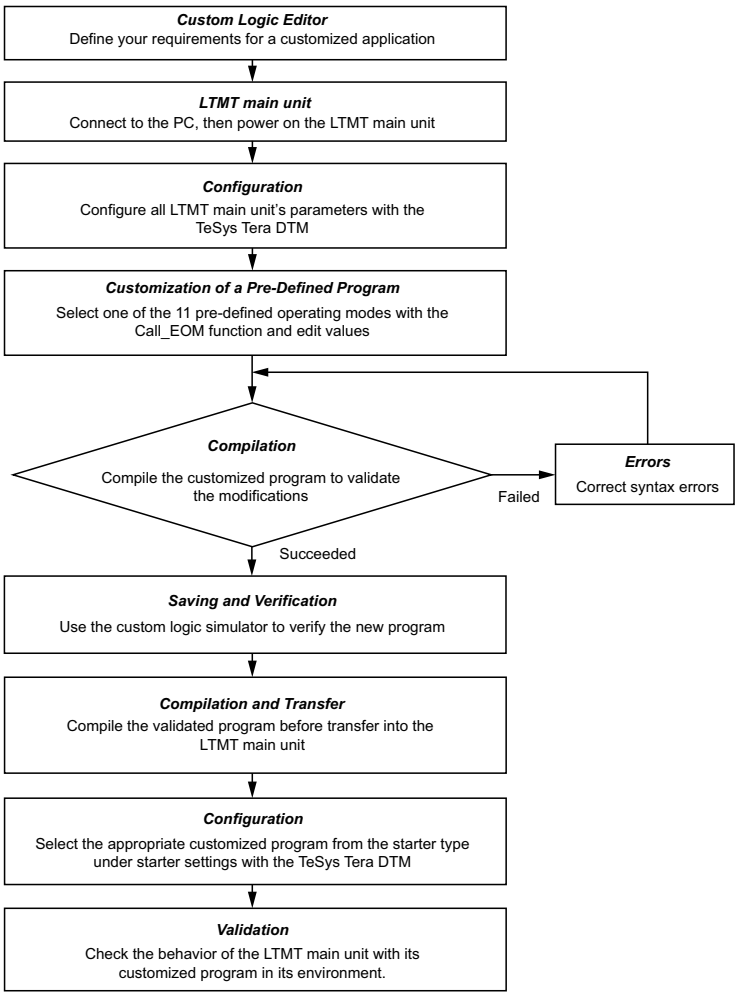
Overview

The custom logic editor enables you to create and validate your own customized program to match with your needs. Once it is made, the LTMT main unit firmware loads and runs instructions you created.

Task Flow Diagram

The following diagram shows all of the tasks to be carried out during the creation and modification of a customized program.

Note: The order defined is provided as an example. The order you use will depend on your own work methods.



Customization Method in Custom Logic

To customize the pre-defined program files, follow below steps:

1. Define the operating modes that matches your application needs.
2. Open the pre-defined operating mode program file (*.rtf) in the custom logic editor.

3. Edit the pre-defined program in custom logic, customize the program following one of the three methods:
   - The pre-defined operating mode matches your application needs: use only the CALL_EOM function.
   - The pre-defined operating mode matches your application needs but additional functions are required: use the CALL_EOM function and add the additional instructions after the CALL_EOM instructions.
   - The pre-defined operating mode does not match your application needs: start a new program from scratch (not recommended).
4. If required, edit the inputs of the customized program.
5. If required, edit the outputs of the customized program.
6. Update the Presentation of the Custom Logic Editor, page 82 according to the CALL_EOM and the control mode.
7. Compile the customized program.

Customization Method in FBD

1. Open a blank FBD program page.
2. Create the input management of the customized program.
3. Create the operating mode execution following one of the three methods:
   - One of the operating mode matches your application needs: use only the CALL_EOM function.
   - One of the operating mode matches your application needs but additional functions are required: use the CALL_EOM function and add the additional instructions after the CALL_EOM instructions.
   - None of the operating mode matches your application needs: create a new program from scratch (not recommended).
4. Create the output management of the customized program.
5. Update the Presentation of the Custom Logic Editor, page 82 according to the CALL_EOM and the control mode.
6. Compile the FBD to custom logic.

Characteristics of the Custom Logic Program

Introduction

The data transferred to or from the LTMT main unit is in the form of 16-bit registers. The registers are numerically ordered and referenced by a 16-bit register address (0–65,535).

The customized program can modify the values of three types of registers:

- LTMT main unit registers
- Temporary registers
- Non-volatile registers

Logic Memory Characteristics

The list of commands for the customized program is saved in an area of the internal non-volatile memory of the LTMT main unit.

The format of this logic memory is illustrated in this table:

Memory location	Item	Range	Description
0	Logic Program Size (n)	0–8,191 0 means that no customized program is loaded.	16-bit word
1	Logic Checksum	0–65,535	Sum of program memory from offset 2–n+2
2	Logic ID	268–511, refer to	Identifier of the customized program within the LTMT main unit
3	Logic Command/Argument 1	Depending on the Logic Commands	One word of logic function
4	Logic Command/Argument 2		
5	Logic Command/Argument 3		
—	—	–	–
n+2	Logic Command/Argument n	–	One word of logic function

Logic Memory Limits

The program size is dependent on the number of logic commands. While in the text editor a command and its arguments will occupy a single line, in the memory, it will occupy as many memory locations as there are arguments.

For example, the command timer 0.1 980 will use four memory locations.

Definition of the Custom Logic Variables

Introduction

The custom logic editor enables you to implement commands in the control program which directs the LTMT main unit:

- To read or write to temporary registers
- To read or write to non-volatile registers
- To read or write to LTMT main unit registers, refer to Definition of LTMT Main Unit Variables, page 89.

The LTMT main unit defines each custom logic register by an integer describing its address in custom logic memory space. The value of this integer begins at address 0 and the maximum address is equal to 1 less than the number of memory locations available for registers in the LTMT main unit.

Temporary Registers

The LTMT main unit provides registers in temporary memory that can be accessed by logic commands. Because these registers exist in temporary or volatile memory, they do not retain their value settings when power to the controller is cycled.

300 temporary registers are available, at addresses ranging from 0 to 299.

The number of temporary registers available can be read in the LTMT main unit register at address 1204 (0x04B4), custom logic temporary space.

Non-Volatile Registers

The LTMT main unit provides registers in non-volatile memory for use by logic commands. Because these registers exist in non-volatile memory, they retain their value settings when power to the controller is cycled.

64 non-volatile registers are available, at addresses ranging from 0 to 63.

The number of non-volatile registers available can be read in the LTMT main unit register at address 1205 (0x04B5), custom logic non-volatile space.

Definition of LTMT Main Unit Variables

LTMT Main Unit Registers

LTMT main unit memory includes 9250 registers at addresses ranging from 0 to 9249 (from 0x0000 to 0x2421).

Each register is a 16-bit word and is either:

- Read-only, with values that cannot be edited.
- Read-write, with values that can be edited.

The register number is equal to the register address + 1.

The custom logic editor uses only register addresses.

Custom logic commands can be used to change the values of read-write registers of the LTMT main unit.

Accessing Registers

Using the custom logic editor, you can access all LTMT main unit registers defined in the TeSys Tera Communication Guides, page 7.

Custom Logic Registers

Registers at addresses from 1200 to 1205 (from 0x04B0 to 0x04B5) and register at address 1291 (0x050B) are used by the TeSys Tera DTM Library to access LTMT main unit registers.

These registers are also the custom logic registers accessible from the communication ports. These read-only registers are described in the following sections.

This table lists these registers:

Register address	Definition	Range (value)
1201 (0x04B1)	Custom logic version	0–65,535
1202 (0x04B2)	Custom logic memory space	
1203 (0x04B3)	Custom logic memory used	
1204 (0x04B4)	Custom logic temporary space	
1205 (0x04B5)	Custom logic non-volatile space	
1291 (0x050B)	Custom logic DO input information	

Register at Address 1201 (0x04B1)

Register at address 1201 (0x04B1) indicates the custom logic capability version. The version number identifies a specific group of logic commands supported by the LTMT main unit.

Register at Address 1202 (0x04B2)

Register at address 1202 (0x04B2) defines the logic memory space available. The number of non-volatile LTMT main unit logic memory words (16 bits) available to save logic commands.

Register at Address 1203 (0x04B3)

Register at address 1203 (0x04B3) defines the logic memory used. This is the number of non-volatile LTMT main unit logic memory words (16 bits) used by logic commands which are currently stored in the LTMT main unit.

Register at Address 1204 (0x04B4)

Register at address 1204 (0x04B4) defines the number of temporary registers provided by the LTMT main unit.

Register at Address 1205 (0x04B5)

Register at address 1205 (0x04B5) defines the number of non-volatile registers provided by the LTMT main unit.

Register at Address 1291 (0x050B)

Register at address 1291 (0x050B) is the custom logic DO input information register. It enables the customized program to configure I/O assignment.

This table describes each bit in this register:

Bit number	Description
0	Custom logic digital output 1 (DO1) input information
1	Custom logic digital output 2 (DO2) input information
2	Custom logic digital output 3 (DO3) input information
3	Custom logic digital output 4 (DO4) input information
4	Custom logic digital output 5 (DO5) input information
5	Custom logic digital output 6 (DO6) input information
6	Custom logic digital output 7 (DO7) input information
7	Custom logic digital output 8 (DO8) input information
8	Custom logic digital output 9 (DO9) input information
9	Custom logic digital output 10 (DO10) input information
10	Custom logic digital output 11 (DO11) input information
11	Custom logic digital output 12 (DO12) input information
12	Custom logic digital output 13 (DO13) input information
13	Reserved
14	Reserved
15	Reserved

Registers at Addresses from 1301 to 1399 (from 0x0515 to 0x0577)

Registers at addresses from 1301 to 1399 (from 0x0515 to 0x0577) are the general purpose registers for logic functions. They are used to exchange information between external sources (such as the PLC) and the custom logic applications.

These volatile registers are read or write and can be edited either by the custom logic functions or via the communication port.

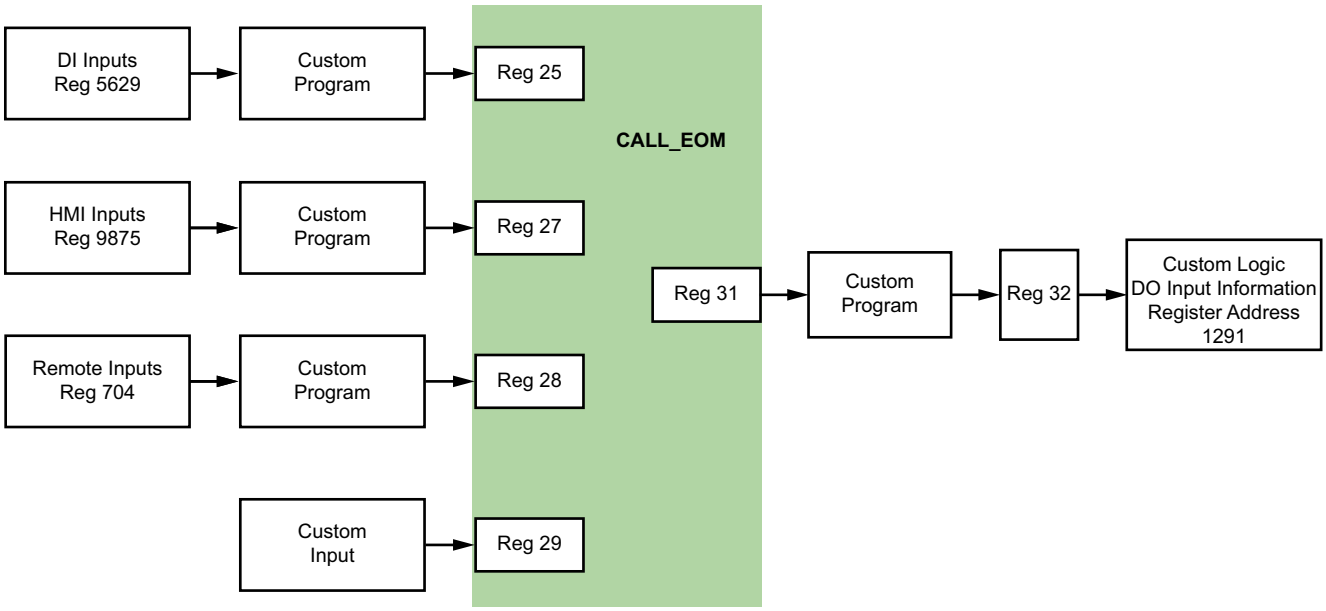
CALL_EOM Command Description

Overview

The CALL_EOM function allows to run an operating mode.

For this purpose, the function uses the temporary registers at addresses from 0 to 61.

To build a customized program around the CALL_EOM function, it is necessary to understand how the different registers of the application and the LTMT main unit are used:



- Temporary registers at addresses from 25 to 29 are the input registers of the CALL_EOM function. When customized, they must be assigned bit per bit.
- Temporary register at address 31 is the output register of the CALL_EOM function. Its value is given after the execution of the operating mode.
- Temporary register at address 32 is a temporary register used to set the custom logic DO input information register at address 1291 (0x050B) in one time. The customization of the CALL_EOM outputs must be done using the temporary register at address 32.

Digital Input Registers

Register address	Register description	Bit	Bit description
5629 (0x15FD)	Digital input register	0	Local-START> DI
		1	Local-STOP DI
		2	Local-START>> DI
		3	Mode selection 1
		4	Local-START< DI
		5	Local-START<< DI
		6	Remote-START> DI
		7	Remote-STOP DI
		8	Remote-START>> DI
		9	Mode selection 2
		10	Remote-START< DI
		11	Remote-START<< DI
		12	Run DI
		13	Speed change DI
		14	Reserved
		15	Reserved

HMI Input Registers

Register address	Register description	Bit	Bit description
9875 (0x2693)	HMI command register 1	0	Motor run forward/High speed forward command
		1	Motor run reverse/High speed reverse command
		2	Local/Remote mode selection 1
		3	Trip reset command
		4	Local/Remote mode selection 2
		5	Self-test (without trip) command
		6	Motor low speed forward command
		7	Motor low speed reverse command
		8	Reset Inhibit command
		9	Reset number of starts command
		10	Reset number of stops command
		11	Clear energy command
		12	Reserved
		13	Logic test command
		14	Reset run hour command
		15	Self-test (with trip) command
9876 (0x2694)	HMI command register 2	0	Reserved
		1	Reserved
		2	Clear thermal capacity level command
		3	Reserved
		4	Reserved

		5	Clear trip counter command
		6	Factory reset command
		7	Soft starter reset command
		8–12	Reserved
		13	Store reference start curve command
		14	Clear trip records command
		15	Clear event records command

Remote Input Registers

Register address	Register description	Bit	Bit description
704 (0x02C0)	Remote command register 1	0	Motor run forward/High speed forward command
		1	Motor run reverse/High speed reverse command
		2	Local/Remote mode selection 1
		3	Trip reset command
		4	Local/Remote mode selection 2
		5	Self-test (without trip) command
		6	Motor low speed forward command
		7	Motor low speed reverse command
		8	Reset Inhibit command
		9	Reset number of starts command
		10	Reset number of stops command
		11	Clear energy command
		12	Reserved
		13	Logic test command
		14	Reset run hour command
		15	Self-test (with trip) command
705 (0x02C1)	Remote command register 2	0	Reserved
		1	Reserved
		2	Clear thermal capacity level command
		3	Reserved
		4	Reserved
		5	Clear trip counter command
		6	Factory reset command
		7	Soft starter reset command
		8–12	Reserved
		13	Store reference start curve command
		14	Clear trip records command
		15	Clear event records command

CALL_EOM 13 Description

When the CALL_EOM argument equals 13, the function executes the Direct Online operating mode.

The logic ID to use in your customized program is:

- LOGID_ID 269 for Direct Online operating mode

The temporary registers are used as follows:

Register	Register description	Bit	Bit description
Input register 25	Copy of the DI inputs	0	Local-START> DI
		1	Local-STOP DI
		2	Not used
		3	Mode Selection 1
		4-5	Not used
		6	Remote-START> DI
		7	Remote-STOP DI
		8	Not used
		9	Mode Selection 2
		10-11	Not used
		12	Run DI
		13-15	Not used
Input register 27	Copy of the HMI inputs	0	HMI_START >
		1	Not used
		2	HMI_STOP
		3-4	Not used
		5	Mode Selection 1
		6	Mode Selection 2
		7-15	Not used
Input register 28	Copy of the remote inputs	0	Remote Start >
		1-3	Not used
		4	Mode Selection 1
		5	Mode Selection 2
		6-15	Not used
Input register 29	Copy of the custom inputs	0	Custom Start >
		1	Not used
		2	Custom Stop
		3-15	Not used
Output register 31	Outputs of the CALL_EOM instruction to assign to LTMT main unit outputs.	0	Custom logic Run1 command information
		1-15	Not used

CALL_EOM 14 Description

When the CALL_EOM argument equals 14, the function executes the Reverse Direct Online operating mode.

The logic ID to use in your customized program is:

- LOGID_ID 270 for Reverse Direct Online operating mode

The temporary registers are used as follows:

Register	Register description	Bit	Bit description
Input register 25	Copy of the DI inputs	0	Local-START> DI
		1	Local-STOP DI
		2	Not used
		3	Mode Selection 1
		4	Local-START< DI
		5	Not used
		6	Remote-START> DI
		7	Remote-STOP DI
		8	Not used
		9	Mode Selection 2
		10	Remote-START< DI
		11	Not used
		12	Run DI
		13-15	Not used
Input register 27	Copy of the HMI inputs	0	HMI_START >
		1	Not used
		2	HMI_STOP
		3	HMI_START <
		4	Not used
		5	Mode Selection 1
		6	Mode Selection 2
		7-15	Not used
Input register 28	Copy of the remote inputs	0	Remote Start >
		1–2	Not used
		3	Remote Start <
		4	Mode Selection 1
		5	Mode Selection 2
		6-15	Not used
Input register 29	Copy of the custom inputs	0	Custom Start >
		1	Not used
		2	Custom Stop
		3	Custom Start <
		4-15	Not used
Output register 31	Outputs of the CALL_EOM instruction to assign to LTMT main unit outputs.	0	Custom logic Run1 command information
		1	Custom logic Run2 command information
		2-15	Not used

CALL_EOM 15 Description

When the CALL_EOM argument equals 15, the function executes the Star Delta operating mode.

The logic ID to use in your customized program is:

- LOGID_ID 271 for Star Delta operating mode

The temporary registers are used as follows:

Register	Register description	Bit	Bit description
Input register 25	Copy of the DI inputs	0	Local-START> DI
		1	Local-STOP DI
		2	Not used
		3	Mode Selection 1
		4-5	Not used
		6	Remote-START> DI
		7	Remote-STOP DI
		8	Not used
		9	Mode Selection 2
		10-11	Not used
		12	Run DI
		13-15	Not used
Input register 27	Copy of the HMI inputs	0	HMI_START >
		1	Not used
		2	HMI_STOP
		3-4	Not used
		5	Mode Selection 1
		6	Mode Selection 2
		7-15	Not used
Input register 28	Copy of the remote inputs	0	Remote Start >
		1-3	Not used
		4	Mode Selection 1
		5	Mode Selection 2
		6-15	Not used
Input register 29	Copy of the custom inputs	0	Custom Start >
		1	Not used
		2	Custom Stop
		3-15	Not used
Output register 31	Outputs of the CALL_EOM instruction to assign to LTMT main unit outputs.	0	Custom logic Run1 command information
		1	Custom logic Run2 command information
		2	Custom logic Run3 command information
		3-15	Not used

Program Example

```
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 DO1 Input information
LOAD_TMP_BIT  31  1                                   //CL “Run2 Cde” information
SET_TMP_BIT   32  1                                   //CL DO2 Input information
LOAD_TMP_BIT  31  2                                   //CL “Run3 Cde” information

SET_TMP_BIT   32  2                                   //CL DO3 Input information
LOAD_TMP_BIT  31  3                                   //CL “Run4 Cde” information
SET_TMP_BIT   32  3                                   //CL DO4 Input information
LOAD_TMP_BIT  31  4                                   //CL “Run5 Cde” information
SET_TMP_BIT   32  4                                   //CL DO5 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
//
```

Custom Logic Language

What's in This Part

Creating a Custom Logic Program 101
Logic Commands 111
Custom Logic Program Examples 142

Creating a Custom Logic Program

What's in This Chapter

Introducing Custom Logic Editor 102
Custom Logic Editor User Interface 103
Logic Commands 106

Introducing Custom Logic Editor

Overview

The custom logic editor is a feature of SoMove software with the TeSys Tera DTM Library. Use the custom logic editor to view an existing program file or to create a program file using an instruction based text language, rather than a graphics based programming language.

Editing Custom Logic Program

The easiest way to create a customized program is to begin with the pre-defined program of one of the operating modes (or motor starter type). The installation of the custom logic editor comes with 11 pre-defined program files, one for each motor starter type.

Each program file bears a descriptive name (e.g. DIRECT_ONLINE) and a file extension of *.rtf*.

Custom Logic Editor User Interface

To open the custom logic editor, select **Device > Custom Logic > New Custom Logic**.

The custom logic editor is available regardless of whether the TeSys Tera DTM Library is in connected mode. However, the transfer of programs between the TeSys Tera DTM Library and the device works only in connected mode.

Custom Logic Editor User Interface

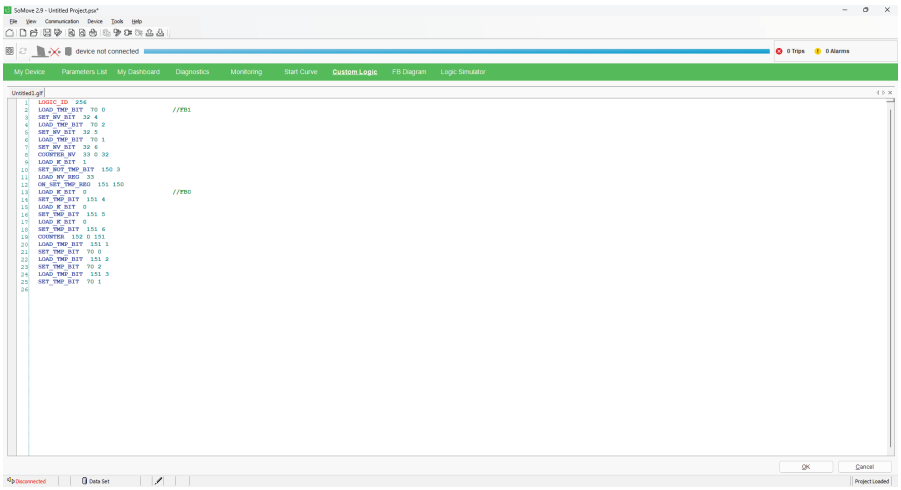
Introduction

A program written in this language consists of a series of instructions executed sequentially by the LTMT main unit. Each list instruction is represented by a single program line and consists of four components:

- Line number
- Logic command (Mnemonics)
- Argument(s)
- Comment(s)

Example of a Custom Logic Program

The following window is an example of a program created with the custom logic editor.



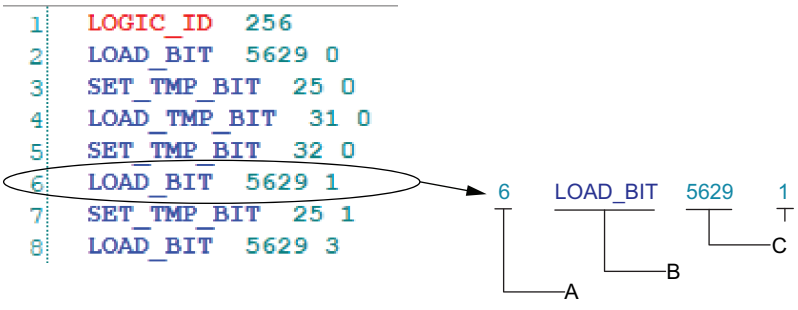
Editing Several Programs

You can create or modify several customized programs at the same time. Click the file name to switch between them.

For instance, in the Text view above, select either **DIRECT_ONLINE.rtf** or **Untitled.lf** depending on the program you want to edit.

Instruction Elements

The following illustration is a sample of custom logic program:



A	Line Number
B	Logic Command
C	Argument(s)

Line Number

The line number is an additional information:

- It is defined only by the editor.
- It does not have any importance in the custom logic function itself.

Logic Command

A logic command is an instruction which identifies the operation to be performed using one or more arguments. In the example, the `LOAD_BIT` command loads the value of the argument into an internal register called the 1-bit accumulator.

There are two types of commands:

- Setup commands

  These set-up or test for the necessary conditions to perform an action (for example, `LOAD` and `AND` commands).

- Actions commands

  These commands direct the LTMT main unit to perform an action based on information in the setup instructions (for example, assignment commands such as `COMP`).

  NOTE: When you type a logic command, either uppercase or lowercase, it is automatically recognized and displayed in blue.

Argument

An argument is a number representing a value (register address, bit number, or constant) that the LTMT main unit can manipulate in an instruction. A logic command can have from 0 to 3 arguments depending on the type of logic command.

For example, in the sample program, the instruction `LOAD_BIT 5629 1` includes a logic command `LOAD_BIT` and 2 arguments, `5629` and `1`.

This instructs the LTMT main unit to load the value of the bit 1 of the register at the address 5629 (0x15FD) into the accumulator.

Using instructions with commands and arguments, the customized program can:

- Read the status of digital inputs.
- Read or write the status of digital outputs.
- Activate basic logic functions such as timers and counters.
- Perform arithmetic, logical, comparisons, and numerical operations.
- Read or write to the LTMT main unit registers or to individual bits in those registers.

  NOTE: When you type an argument, it is automatically recognized and displayed in the color assigned to the arguments.

Comments

In the custom logic editor, it is possible to add comments to the program:

- At the end of each line after the arguments
- In a whole line

NOTE:

- When you type //, the custom logic editor automatically recognizes the text after it as comments and displays it in green.
- Comments can not be retrieved from the LTMT main unit.

Syntax

In the custom logic editor, it is possible to write instructions:

- With blanks, commas, or dots between arguments
- In upper or lower characters

Syntax Check

During typing, the text editor checks the instruction syntax:

- Correct instructions are displayed in bold blue characters.
- Incorrect instructions stay displayed in black color and must be corrected before the compilation.

Keyboard Commands

Keyboard commands and shortcuts are the same as those for Windows operating systems (For example: Press **DEL** or **DELETE** to delete a character or line, press **ENTER** to go to the next line).

Saving

To save the program you edited or created, select **Device > Custom Logic**, then select **Save Custom Logic** or **SaveAs Custom Logic**.

NOTE: This file is saved with the extension **.lf*.

Logic Commands

Overview

The LTMT main unit project files consist of a series of logic commands. Each logic command consists of the command itself, plus up to 3 arguments.

Each logic command performs its operation linked to either a 1-bit Boolean accumulator (value 0 or 1) or a 16-bit unsigned accumulator (value range 0–65,535).

The custom logic editor provides the following kinds of logic commands:

- Boolean
- Register
- Timers
- Latch
- Counters
- Math

Rising Edge Detection Mechanism

Some logic commands work on a rising edge of the 1-bit accumulator.

The rising edge of a bit is detected when its current state is 1 and its previous state was 0. The previous state of the bit is stored in a dedicated history bit.

NOTE: If this history bit is modified, the detection of the rising edge can be disturbed.

Boolean Logic Commands

Boolean commands evaluate and control simple Boolean (On or Off) values. Boolean commands include:

Command	Argument 1	Argument 2	Argument 3	Description
LOAD_K_BIT	Constant value (0 or 1)	–	–	Loads a constant value into the 1-bit accumulator.
LOAD_BIT	Register address	Register bit no. (0–15)	–	Loads a register bit from the address identified in Argument 1, and the bit identified in Argument 2 into the 1-bit accumulator.
LOAD_TMP_BIT	Temporary register address	Register bit no. (0–15)	–	Loads a temporary register bit into the 1-bit accumulator.
LOAD_NV_BIT	Non-volatile register address	Register bit no. (0–15)	–	Loads a non-volatile register bit into the 1-bit accumulator.
LOAD_NOT_BIT	Register address	Register bit no. (0–15)	–	Loads an inverted Boolean value of a register bit into the 1-bit accumulator.
LOAD_NOT_TMP_BIT	Temporary register address	Register bit no. (0–15)	–	Loads an inverted Boolean value of a temporary register bit into the 1-bit accumulator.
LOAD_NOT_NV_BIT	Non-volatile register address	Register bit no. (0–15)	–	Loads an inverted Boolean value of a non-volatile register bit into the 1-bit accumulator.
AND_BIT	Register address	Register bit no. (0–15)	–	Loads the result of a logical AND link between the register bit value and the 1-bit accumulator content. The result is stored in the 1-bit accumulator.
AND_TMP_BIT	Temporary register address	Register bit no. (0–15)	–	Loads the result of a logical AND link between the temporary register bit value and the 1-bit accumulator content. The result is stored in the 1-bit accumulator.

Command	Argument 1	Argument 2	Argument 3	Description
AND_NV_BIT	Non-volatile register address	Register bit no. (0–15)	–	Loads the result of a logical AND link between the non-volatile register bit value and the 1-bit accumulator content. The result is stored in the 1-bit accumulator.
AND_NOT_BIT	Register address	Register bit no. (0–15)	–	Loads the result of a logical AND of the inverted register bit and the 1-bit accumulator. The result is stored in the 1-bit accumulator.
AND_NOT_TMP_BIT	Temporary register address	Register bit no. (0–15)	–	Loads the result of a logical AND of the inverted temporary register bit and the 1-bit accumulator. The result is stored in the 1-bit accumulator.
AND_NOT_NV_BIT	Non-volatile register address	Register bit no. (0–15)	–	Loads the result of a logical AND of the inverted non-volatile register bit and the 1-bit accumulator. The result is stored in the 1-bit accumulator.
OR_BIT	Register address	Register bit no. (0–15)	–	Makes a logical OR link between the register bit value and the 1-bit accumulator content. The result is stored in the 1-bit accumulator.
OR_TMP_BIT	Temporary register address	Register bit no. (0–15)	–	Makes a logical OR link between the temporary register bit value and the 1-bit accumulator content. The result is stored in the 1-bit accumulator.
OR_NV_BIT	Non-volatile register address	Register bit no. (0–15)	–	Makes a logical OR link between the non-volatile register bit value and the 1-bit accumulator content. The result is stored in the 1-bit accumulator.
OR_NOT_BIT	Register address	Register bit no. (0–15)	–	Makes an logical OR of the inverted register bit and the 1-bit accumulator. The result is stored in the 1-bit accumulator.
OR_NOT_TMP_BIT	Temporary register address	Register bit no. (0–15)	–	Makes an logical OR of the inverted temporary register bit and the 1-bit accumulator. The result is stored in the 1-bit accumulator.
OR_NOT_NV_BIT	Non-volatile register address	Register bit no. (0–15)	–	Makes an logical OR of the inverted non-volatile register bit and the 1-bit accumulator. The result is stored in the 1-bit accumulator.
SET_BIT	Register address	Register bit no. (0–15)	–	Sets value of the 1-bit accumulator into a register bit.
SET_TMP_BIT	Temporary register address	Register bit no. (0–15)	–	Sets value of the 1-bit accumulator into a temporary register bit.
SET_NV_BIT	Non-volatile register address	Register bit no. (0–15)	–	Sets value of the 1-bit accumulator into a non-volatile register bit.
SET_NOT_BIT	Register address	Register bit no. (0–15)	–	Sets inverted value of the 1-bit accumulator into a register bit.
SET_NOT_TMP_BIT	Temporary register address	Register bit no. (0–15)	–	Sets inverted value of the 1-bit accumulator into a temporary register bit.
SET_NOT_NV_BIT	Non-volatile register address	Register bit no. (0–15)	–	Sets inverted value of the 1-bit accumulator into a non-volatile register bit.

– Argument not applicable to logic command.

Register Logic Commands

Register commands evaluate and control 16-bit values. Register commands include:

Command	Argument 1	Argument 2	Argument 3	Description
LOAD_K_REG	Constant value (0–65,535)	–	–	Loads a constant value into the 16-bit accumulator.
LOAD_REG	Register address	–	–	Loads a copy of a register into the 16-bit accumulator.
LOAD_TMP_REG	Temporary register address	–	–	Loads a copy of a temporary register into the 16-bit accumulator.

Command	Argument 1	Argument 2	Argument 3	Description
LOAD_NV_REG	Non-volatile register address	–	–	Loads a copy of a non-volatile register into the 16-bit accumulator.
COMP_K_REG	Constant value (0–65,535)	Temporary register address	–	Compares the content of the Argument 1 to the 16-bit accumulator content and sets status Argument 2 bits as follows: BIT 1 ON if 16-bit accumulator < content of the Argument 1 BIT 2 ON if 16-bit accumulator = content of the Argument 1 BIT 3 ON if 16-bit accumulator > content of the Argument 1
COMP_REG	Register address	Temporary register address	–	Compares the content of the register defined by Argument 1 to the 16-bit accumulator content and sets status Argument 2 bits as follows: BIT 1 ON if 16-bit accumulator < content of the register defined by Argument 1 BIT 2 ON if 16-bit accumulator = content of the register defined by Argument 1 BIT 3 ON if 16-bit accumulator > content of the register defined by Argument 1
COMP_TMP_REG	Temporary register address	Temporary register address	–	Compares the content of the register defined by Argument 1 to the 16-bit accumulator content and sets status Argument 2 bits as follows: BIT 1 ON if 16-bit accumulator < content of the register defined by Argument 1 BIT 2 ON if 16-bit accumulator = content of the register defined by Argument 1 BIT 3 ON if 16-bit accumulator > content of the register defined by Argument 1
COMP_NV_REG	Non-volatile register address	Temporary register address	–	Compares the content of the register defined by Argument 1 to the 16-bit accumulator content and sets status Argument 2 bits as follows: BIT 1 ON if 16-bit accumulator < content of the register defined by Argument 1 BIT 2 ON if 16-bit accumulator = content of the register defined by Argument 1 BIT 3 ON if 16-bit accumulator > content of the register defined by Argument 1
AND_K	Constant value (0 or 1)	–	–	Makes a logical AND link between the constant value and the 16-bit accumulator content. The result is stored in the 16-bit accumulator.
AND_REG	Register address	–	–	Makes a logical AND link between the register value and the 16-bit accumulator content. The result is stored in the 16-bit accumulator.
AND_TMP_REG	Temporary register address	–	–	Makes a logical AND link between the temporary register value and the 16-bit accumulator content. The result is stored in the 16-bit accumulator.
AND_NV_REG	Non-volatile register address	–	–	Makes a logical AND link between the non-volatile register value and the 16-bit accumulator content. The result is stored in the 16-bit accumulator.
OR_K	Constant value (0 or 1)	–	–	Makes a logical OR link between the constant value and the 16-bit accumulator content. The result is stored in the 16-bit accumulator.
OR_REG	Register address	–	–	Makes a logical OR link between the register value and the 16-bit accumulator content. The result is stored in the 16-bit accumulator.
OR_TMP_REG	Temporary register address	–	–	Makes a logical OR link between the temporary register value and the 16-bit accumulator content. The result is stored in the 16-bit accumulator.

Command	Argument 1	Argument 2	Argument 3	Description
OR_NV_REG	Non-volatile register address	–	–	Makes a logical exclusive OR link between the non-volatile register value and the 16-bit accumulator content. The result is stored in the 16-bit accumulator.
XOR_K	Constant value (0–65,535)	–	–	Makes a logical exclusive OR link between the constant value and the 16-bit accumulator content. The result is stored in the 16-bit accumulator.
XOR_REG	Register address	–	–	Makes a logical exclusive OR link between the register value and the 16-bit accumulator content. The result is stored in the 16-bit accumulator.
XOR_TMP_REG	Temporary register address	–	–	Makes a logical exclusive OR link between the temporary register value and the 16-bit accumulator content. The result is stored in the 16-bit accumulator.
XOR_NV_REG	Non-volatile register address	–	–	Makes a logical exclusive OR link between the non-volatile register value and the 16-bit accumulator content. The result is stored in the 16-bit accumulator.
ON_SET_REG	Register address	Temporary register address	–	Stores the 16-bit accumulator content into the register defined by Argument 1 on a rising edge of the 1-bit accumulator.
ON_SET_TMP_REG	Temporary register address	Temporary register address	–	Stores the 16-bit accumulator content into the temporary register defined by Argument 1 on a rising edge of the 1-bit accumulator.
ON_SET_NV_REG	Non-volatile register address	Temporary register address	–	Stores the 16-bit accumulator content into the non-volatile register defined by Argument 1 on a rising edge of the 1-bit accumulator.
– Argument not applicable to logic command.				

Timer Logic Commands

Timers have a range from 0 to 65,535 and measure time in intervals of seconds or tenth of seconds:

- Argument 1 specifies the time period
- Argument 2 is a calculated end time
- Argument 3 is the timer status register

Timer commands include:

Command	Argument 1	Argument 2	Argument 3	Description
TIMER_SEC	Temporary register (time period)	Temporary register (calculated end time)	Temporary register (status)	Counts in seconds the time period input in Argument 1 as described by status register bits.
TIMER_TENTHS	Temporary register (time period)	Temporary register (calculated end time)	Temporary register (status)	Counts in tenths of seconds the time period input in Argument 1 as described by status register bits.
TIMER_K_SEC	Constant value 0–65,535 (time period)	Temporary register (calculated end time)	Temporary register (status)	Counts in seconds the time period input in Argument 1 as described by status register bits.
TIMER_K_TENTHS	Constant value 0–65,535 (time period)	Temporary register (calculated end time)	Temporary register (status)	Counts in tenths of seconds the time period input in Argument 1 as described by status register bits.

Latch Logic Commands

Latch commands include:

Command	Argument 1	Argument 2	Argument 3	Description
LATCH	Temporary register (status)	–	–	Records and retains in a temporary register a history of a signal.
LATCH_NV	Non-volatile register (status)	–	–	Records and retains in a non-volatile register a history of a signal.
– Argument not applicable to logic command.				

Counter Logic Commands

Counters have a range from 0 to 65,535 and transition to 0 upon counting to the maximum value of 65,535.

Counter commands include:

Command	Argument 1	Argument 2	Argument 3	Description
COUNTER	Temporary register (counter value)	Constant value 0–65,535 (preset value)	Temporary register (status)	Performs a comparative count, saving both the count and status to temporary registers.
COUNTER_NV	Non-volatile register (counter value)	Constant value 0–65,535 (preset value)	Non-volatile register (status)	Performs a comparative count, saving both the count and status to non-volatile registers.

Math Logic Commands

Math commands perform unsigned math functions using the 16-bit accumulator and temporary registers. Math commands are executed on a rising edge of the 1-bit accumulator. Math commands include:

Command	Argument 1	Argument 2	Argument 3	Description
ON_ADD	Temporary register (value)	Temporary register (status)	–	Argument 1 = Argument 1 + 16-bit accumulator.
ON_SUB	Temporary register (value)	Temporary register (status)	–	Argument 1 = Argument 1 - 16-bit accumulator.
ON_MUL	Temporary register (most significant word)	Temporary register (least significant word)	Temporary register (status)	Argument 1:Argument 2 = 16-bit accumulator x Argument 2.
ON_DIV	Temporary register (most significant word)	Temporary register (least significant word)	Temporary register (status)	Argument 1:Argument 2 = Argument 1:Argument 2 / 16-bit accumulator.
– Argument not applicable to logic command.				

Logic Commands

What's in This Chapter

Program Logic Commands 112
Boolean Logic Commands 114
Register Logic Commands 123
Timer Logic Commands 132
Latch Logic Commands 135
Counter Logic Commands 137
Math Logic Commands 139

Program Logic Commands

Overview

Program logic commands are used to:

- Identify the program file to the custom logic editor
- Execute a pre-defined operating mode

The following commands can be used:

- LOGIC_ID
- CALL_EOM
- NOP

LOGIC_ID

The LOGIC_ID statement acts as an identifier for the program file.

LOGIC_ID values have an integer value range from 256 to 511.

Arguments	Representation
1	LOGIC_ID ID#

Input argument	Type	Range	Description
ID#	UINT	256–511	Logic ID of the customized program

No output arguments.

CALL_EOM

The CALL_EOM executes a pre-defined operating mode in the customized program.

Arguments	Representation
1	CALL_EOM OP_MODE#

Input argument	Type	Range	Representation
OP_MODE#	INT	12–22	Embedded operating mode (EOM): • 13: Direct Online • 14: Reverse Direct Online • 15: Star Delta

No output arguments.

NOP

The NOP command performs no operation.

Use the NOP command as a placeholder in a program file to replace a pre-existing command, or to reserve space for a future command.

Arguments	Representation
0	NOP

The NOP command has no arguments.

Boolean Logic Commands

Overview

The custom logic editor uses the following boolean logic commands:

- 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

The LOAD_K_BIT command loads a constant Boolean value (0 or 1) into the 1-bit accumulator.

Arguments	Representation
1	LOAD_K_BIT KValue

Input arguments	Type	Range	Description
KValue	BOOL	0/1	A constant value

No output arguments.

LOAD_BIT

The LOAD_BIT command loads the Boolean value (0 or 1) of a register bit into the 1-bit accumulator.

Arguments	Representation
2	LOAD_BIT RegAddr BitNo

Input arguments	Type	Range	Description
RegAddr	UINT	0–9249	The register address
BitNo	UINT	0–15	The bit number

No output arguments.

LOAD_TMP_BIT

The LOAD_TMP_BIT command loads the Boolean value (0 or 1) of a temporary register bit into the 1-bit accumulator.

Arguments	Representation
2	LOAD_TMP_BIT TmpReg BitNo

Input arguments	Type	Range	Description
TmpReg	UINT	0–299	The temporary register number
BitNo	UINT	0–15	The bit number

No output arguments.

LOAD_NV_BIT

The LOAD_NV_BIT command loads the Boolean value (0 or 1) of a non-volatile register bit into the 1-bit accumulator.

Arguments	Representation
2	LOAD_NV_BIT NVReg BitNo

Input arguments	Type	Range	Description
NVReg	UINT	0–63	The non-volatile register number
BitNo	UINT	0–15	The bit number

No output arguments.

LOAD_NOT_BIT

The LOAD_NOT_BIT command:

- Inverts the Boolean value (0 or 1) of a specified register bit, then
- Loads the inverted value into the 1-bit accumulator.

Arguments	Representation
2	LOAD_NOT_BIT RegAddr BitNo

Input arguments	Type	Range	Description
RegAddr	UINT	0–9249	The register address
BitNo	UINT	0–15	The bit number

No output arguments.

LOAD_NOT_TMP_BIT

The LOAD_NOT_TMP_BIT command:

- Inverts the Boolean value (0 or 1) of a specified temporary register bit, then
- Loads the inverted value into the 1-bit accumulator.

Arguments	Representation
2	LOAD_NOT_TMP_BIT TmpReg BitNo

Input arguments	Type	Range	Description
TmpReg	UINT	0–299	The temporary register number
BitNo	UINT	0–15	The bit number

No output arguments.

LOAD_NOT_NV_BIT

The LOAD_NOT_NV_BIT command:

- Inverts the Boolean value (0 or 1) of a selected non-volatile register bit, then
- Loads the inverted value into the 1-bit accumulator.

Arguments	Representation
2	LOAD_NOT_NV_BIT NVReg BitNo

Input arguments	Type	Range	Description
NVReg	UINT	0–63	The non-volatile register number
BitNo	UINT	0–15	The bit number

No output arguments.

AND_BIT

The AND_BIT command makes a logical AND link between a register bit value and the accumulator content in logic memory:

- If the 1-bit accumulator equals 1 and the linked register bit equals 1, the result of the AND process is also 1.
- In all other cases the result of the AND process is 0.

The result is saved in the 1-bit accumulator.

Arguments	Representation
2	AND_BIT RegAddr BitNo

Input arguments	Type	Range	Description
RegAddr	UINT	0–9249	The register address
BitNo	UINT	0–15	The bit number

No output arguments.

AND_TMP_BIT

The AND_TMP_BIT command makes a logical AND link between a temporary register bit value and the accumulator content in logic memory.

- If the 1-bit accumulator equals 1 and the linked temporary register bit equals 1, the result of the AND process is also 1.
- In all other cases the result of the AND process is 0.

The result is saved in the 1-bit accumulator.

Arguments	Representation
2	AND_TMP_BIT TmpReg BitNo

Input arguments	Type	Range	Description
TmpReg	UINT	0–299	The temporary register number
BitNo	UINT	0–15	The bit number

No output arguments.

AND_NV_BIT

The AND_NV_BIT command makes a logical AND link between a non-volatile register bit value and the accumulator content in logic memory.

- If the 1-bit accumulator equals 1 and the linked non-volatile register bit equals 1, the result of the AND process is also 1.
- In all other cases the result of the AND process is 0.

The result is saved in the 1-bit accumulator.

Arguments	Representation
2	AND_NV_BIT NVReg BitNo

Input arguments	Type	Range	Description
NVReg	UINT	0–63	The non-volatile register number
BitNo	UINT	0–15	The bit number

No output arguments.

AND_NOT_BIT

The AND_NOT_BIT command inverts the Boolean value (0 or 1) of a specified register bit, then makes a logical AND link between it and the accumulator content in logic memory:

- If the 1-bit accumulator equals 1 and the linked register bit equals 0, the result of the AND process is also 1.
- In all other cases the result of the AND process is 0.

The result is saved in the 1-bit accumulator.

Arguments	Representation
2	AND_NOT_BIT RegAddr BitNo

Input arguments	Type	Range	Description
RegAddr	UINT	0–9249	The register address
BitNo	UINT	0–15	The bit number

No output arguments.

AND_NOT_TMP_BIT

The AND_NOT_TMP_BIT command inverts the Boolean value (0 or 1) of a specified temporary register bit, then makes a logical AND link between it and the accumulator content in logic memory:

- If the 1-bit accumulator equals 1 and the linked temporary register bit equals 0, the result of the AND process is also 1.
- In all other cases the result of the AND process is 0.

The result is saved in the 1-bit accumulator.

Arguments	Representation
2	AND_NOT_TMP_BIT TmpReg BitNo

Input arguments	Type	Range	Description
TmpReg	UINT	0–299	The temporary register number
BitNo	UINT	0–15	The bit number

No output arguments.

AND_NOT_NV_BIT

The AND_NOT_NV_BIT command inverts the Boolean value (0 or 1) of a selected non-volatile register bit, then makes a logical AND link between it and the accumulator content in logic memory:

- If the 1-bit accumulator equals 1 and the linked non-volatile register bit equals 0, the result of the AND process is also 1.
- In all other cases the result of the AND process is 0.

The result is saved in the 1-bit accumulator.

Arguments	Representation
2	AND_NOT_NV_BIT NVReg BitNo

Input arguments	Type	Range	Description
NVReg	UINT	0–63	The non-volatile register number
BitNo	UINT	0–15	The bit number

No output arguments.

OR_BIT

The OR_BIT command makes a logical OR link between a register bit value and the accumulator content in logic memory:

- If the value of either the 1-bit accumulator or the register bit equals 1, the result of the OR process is also 1.
- If the values of all compared bits equal 0, the result of the OR process is 0.

The result is saved in the 1-bit accumulator.

Arguments	Representation
2	OR_BIT RegAddr BitNo

Input arguments	Type	Range	Description
RegAddr	UINT	0–9249	The register address
BitNo	UINT	0–15	The bit number

No output arguments.

OR_TMP_BIT

The OR_TMP_BIT command makes a logical OR link between a temporary register bit value and the accumulator content in logic memory.

- If the value of either the 1-bit accumulator or the temporary register bit equals 1, the result of the OR process is also 1.
- If the values of all compared bits equal 0, the result of the OR process is 0.

The result is saved in the 1-bit accumulator.

Arguments	Representation
2	OR_TMP_BIT TmpReg BitNo

Input arguments	Type	Range	Description
TmpReg	UINT	0–299	The temporary register number
BitNo	UINT	0–15	The bit number

No output arguments.

OR_NV_BIT

The OR_NV_BIT command makes a logical OR link between a non-volatile register bit value and the accumulator content in logic memory.

- If the value of either the 1-bit accumulator or the non-volatile register bit equals 1, the result of the OR process is also 1.
- If the values of all compared bits equal 0, the result of the OR process is 0.

The result is saved in the 1-bit accumulator.

Arguments	Representation
2	OR_NV_BIT NVReg BitNo

Input arguments	Type	Range	Description
NVReg	UINT	0–63	The non-volatile register number
BitNo	UINT	0–15	The bit number

No output arguments.

OR_NOT_BIT

The OR_NOT_BIT command inverts the Boolean value (0 or 1) of a specified register bit, then makes a logical OR link between it and the accumulator content in logic memory:

- If the value of either the 1-bit accumulator or the register bit equals 0, the result of the OR process is also 1.
- If the values of all compared bits equal 0, the result of the OR process is 0.

The result is saved in the 1-bit accumulator.

Arguments	Representation
2	OR_NOT_BIT RegAddr BitNo

Input arguments	Type	Range	Description
RegAddr	UINT	0–9249	The register address
BitNo	UINT	0–15	The bit number

No output arguments.

OR_NOT_TMP_BIT

The OR_NOT_TMP_BIT command inverts the Boolean value (0 or 1) of a specified temporary register bit, then makes a logical OR link between it and the accumulator content in logic memory:

- If the value of either the 1-bit accumulator or the temporary register bit equals 0, the result of the OR process is also 1.
- If the values of all compared bits equal 0, the result of the OR process is 0.

The result is saved in the 1-bit accumulator.

Arguments	Representation
2	OR_NOT_TMP_BIT TmpReg BitNo

Input arguments	Type	Range	Description
TmpReg	UINT	0–299	The temporary register number
BitNo	UINT	0–15	The bit number

No output arguments.

OR_NOT_NV_BIT

The OR_NOT_NV_BIT command inverts the Boolean value (0 or 1) of a selected non-volatile register bit, then makes a logical OR link between it and the accumulator content in logic memory:

- If the value of either the 1-bit accumulator or the non-volatile register bit equals 0, the result of the OR process is also 1.
- If the values of all compared bits equal 0, the result of the OR process is 0.

The result is saved in the 1-bit accumulator.

Arguments	Representation
2	OR_NOT_NV_BIT NVReg BitNo

Input arguments	Type	Range	Description
NVReg	UINT	0–63	The non-volatile register number
BitNo	UINT	0–15	The bit number

No output arguments.

SET_BIT

The SET_BIT command sets the value of the 1-bit accumulator to a specified register bit.

Arguments	Representation
2	SET_BIT RegAddr BitNo

No input arguments.

Output arguments	Type	Range	Description
RegAddr	UINT	0–9249	The register address
BitNo	UINT	0–15	The bit number

SET_TMP_BIT

The SET_TMP_BIT command sets the value of the 1-bit accumulator to a specified temporary register bit.

Arguments	Representation
2	SET_TMP_BIT TmpReg BitNo

No input arguments.

Output arguments	Type	Range	Description
TmpReg	UINT	0–299	The temporary register number
BitNo	UINT	0–15	The bit number

SET_NV_BIT

The SET_NV_BIT command sets the value of the 1-bit accumulator to a specified non-volatile register bit.

Arguments	Representation
2	SET_NV_BIT NVReg BitNo

No input arguments.

Output arguments	Type	Range	Description
NVReg	UINT	0–63	The non-volatile register number
BitNo	UINT	0–15	The bit number

SET_NOT_BIT

The SET_NOT_BIT command sets the inverted value of the 1-bit accumulator to a specified register bit.

Arguments	Representation
2	SET_NOT_BIT RegAddr BitNo

No input arguments.

Output arguments	Type	Range	Description
RegAddr	UINT	0–9249	The register address
BitNo	UINT	0–15	The bit number

SET_NOT_TMP_BIT

The SET_NOT_TMP_BIT command sets the inverted value of the 1-bit accumulator to a specified temporary register bit.

Arguments	Representation
2	SET_NOT_TMP_BIT TmpReg BitNo

No input arguments.

Output arguments	Type	Range	Description
TmpReg	UINT	0–299	The temporary register number
BitNo	UINT	0–15	The bit number

SET_NOT_NV_BIT

The SET_NOT_NV_BIT command sets the inverted value of the 1-bit accumulator to a specified non-volatile register bit.

Arguments	Representation
2	SET_NOT_NV_BIT NVReg BitNo

No input arguments.

Output arguments	Type	Range	Description
NVReg	UINT	0–63	The non-volatile register number
BitNo	UINT	0–15	The bit number

Register Logic Commands

Overview

Register commands evaluate and control 16-bit values.

The custom logic editor uses the following register commands:

- 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

The LOAD_K_REG command loads a 16-bit constant value into the 16-bit accumulator in logic memory.

Arguments	Representation
1	LOAD_K_REG KValue

Input arguments	Type	Range	Description
KValue	UINT	0–65,535	A constant value

No output arguments.

LOAD_REG

The LOAD_REG command loads a copy of a register into the 16-bit accumulator in logic memory.

Arguments	Representation
1	LOAD_REG RegAddr

Input arguments	Type	Range	Description
RegAddr	UINT	0–9249	The register address

No output arguments.

LOAD_TMP_REG

The LOAD_TMP_REG command loads a copy of a temporary register into the 16-bit accumulator in logic memory.

Arguments	Representation
1	LOAD_TMP_REG TmpReg

Input arguments	Type	Range	Description
TmpReg	UINT	0–299	The temporary register number

No output arguments.

LOAD_NV_REG

The LOAD_NV_REG command loads a copy of a non-volatile register into the 16-bit accumulator in logic memory.

Arguments	Representation
1	LOAD_NV_REG NVReg

Input arguments	Type	Range	Description
NVReg	UINT	0–63	The non-volatile register number

No output arguments.

COMP_K_REG

The COMP_K_REG command compares the 16-bit accumulator content to the Argument 1 constant value and sets the result of the comparison in one bit of the Argument 2 temporary register.

Arguments	Representation
2	COMP_K_REG KValue TmpReg

Input arguments	Type	Range/Bit	Description
KValue	UINT	0–65,535	A constant value

Output arguments	Type	Range/Bit	Description
TmpReg	UINT	Bit1	16-bit accumulator < KValue
		Bit2	16-bit accumulator = KValue
		Bit3	16-bit accumulator > KValue

COMP_REG

The COMP_REG command compares the 16-bit accumulator content to the content of the register defined by Argument 1 and sets the result of the comparison in one bit of the Argument 2 temporary register.

Arguments	Representation
2	COMP_REG RegAddr TmpReg

Input arguments	Type	Range/Bit	Description
RegAddr	UINT	0–9249	The register address

Output arguments	Type	Range/Bit	Description
TmpReg	UINT	Bit1	16-bit accumulator < RegAddr
		Bit2	16-bit accumulator = RegAddr
		Bit3	16-bit accumulator > RegAddr

COMP_TMP_REG

The COMP_TMP_REG command compares the 16-bit accumulator content to the content of the temporary register defined by Argument 1 and sets the result of the comparison in one bit of the Argument 2 temporary register.

Arguments	Representation
2	COMP_TMP_REG TmpReg1 TmpReg2

Input arguments	Type	Range/Bit	Description
TmpReg1	UINT	0–299	Temporary register number

Output arguments	Type	Range/Bit	Description
TmpReg2	UINT	Bit1	16-bit accumulator < TmpReg1
		Bit2	16-bit accumulator = TmpReg1
		Bit3	16-bit accumulator > TmpReg1

COMP_NV_REG

The COMP_NV_REG command compares the 16-bit accumulator content to the content of the non-volatile register defined by Argument 1 and sets the result of the comparison in one bit of the Argument 2 temporary register.

Arguments	Representation
2	COMP_NV_REG NVReg TmpReg

Input arguments	Type	Range/Bit	Description
NVReg	UINT	0–63	Non-volatile register number

Output arguments	Type	Range/Bit	Description
TmpReg	UINT	Bit1	16-bit accumulator < NVReg
		Bit2	16-bit accumulator = NVReg
		Bit3	16-bit accumulator > NVReg

AND_K

The AND_K command makes a logical AND link between a 16-bit constant value and the 16-bit accumulator content in logic memory. The result is saved in the 16-bit accumulator.

The AND process compares each bit in the 16-bit accumulator with the corresponding bit in the linked 16-bit constant value:

- If both bits equal 1, the result of the AND process for that bit number is also 1.
- In all other cases the result of the AND process for that bit number is 0.

Arguments	Representation
1	AND_K KValue

Input arguments	Type	Range	Description
KValue	UINT	0–65,535	A constant value

No output arguments.

AND_REG

The AND_REG command makes a logical AND link between the register value and the 16-bit accumulator content in logic memory. The result is saved in the 16-bit accumulator.

The AND process compares each bit in the 16-bit accumulator with the corresponding bit in the linked register:

- If both bits equal 1, the result of the AND process for that bit number is also 1.
- In all other cases the result of the AND process for that bit number is 0.

Arguments	Representation
1	AND_REG RegAddr

Input arguments	Type	Range	Description
RegAddr	UINT	0–9249	The register address

No output arguments.

AND_TMP_REG

The AND_TMP_REG command makes a logical AND link between the temporary register value and the 16-bit accumulator content in logic memory. The result is saved in the 16-bit accumulator.

The AND process compares each bit in the 16-bit accumulator with the corresponding bit in the linked temporary register:

- If both bits equal 1, the result of the AND process for that bit number is also 1.

- In all other cases the result of the AND process for that bit number is 0.

Arguments	Representation
1	AND_TMP_REG TmpReg

Input arguments	Type	Range	Description
TmpReg	UINT	0–299	The temporary register number

No output arguments.

AND_NV_REG

The AND_NV_REG command makes a logical AND link between the non-volatile register value and the 16-bit accumulator content in logic memory. The result is saved in the 16-bit accumulator.

The AND process compares each bit in the 16-bit accumulator with the corresponding bit in the linked non-volatile register:

- If both bits equal 1, the result of the AND process for that bit number is also 1.
- In all other cases the result of the AND process for that bit number is 0.

Arguments	Representation
1	AND_NV_REG NVReg

Input arguments	Type	Range	Description
NVReg	UINT	0–63	The non-volatile register number

No output arguments.

OR_K

The OR_K command makes a logical OR link between a 16-bit constant value and the 16-bit accumulator content in logic memory. The result is saved in the 16-bit accumulator.

The OR process compares each bit in the 16-bit accumulator with the corresponding bit in the linked 16-bit constant value:

- If any compared bit equals 1, the result of the OR process for that bit number is also 1.
- If all compared bits equal 0, the result of the OR process for that bit number is 0.

Arguments	Representation
1	OR_K KValue

Input arguments	Type	Range	Description
KValue	UINT	0–65,535	A constant value

No output arguments.

OR_REG

The OR_REG command makes a logical OR link between the register value and the 16-bit accumulator content in logic memory. The result is saved in the 16-bit accumulator.

The OR process compares each bit in the 16-bit accumulator with the corresponding bit in the linked register:

- If any compared bit equals 1, the result of the OR process for that bit number is also 1.
- If all compared bits equal 0, the result of the OR process for that bit number is 0.

Arguments	Representation
1	OR_REG RegAddr

Input arguments	Type	Range	Description
RegAddr	UINT	0–9249	The register address

No output arguments.

OR_TMP_REG

The OR_TMP_REG command makes a logical OR link between the temporary register value and the 16-bit accumulator content in logic memory. The result is saved in the 16-bit accumulator.

The OR process compares each bit in the 16-bit accumulator with the corresponding bit in the linked temporary register:

- If any compared bit equals 1, the result of the OR process for that bit number is also 1.
- If all compared bits equal 0, the result of the OR process for that bit number is 0.

Arguments	Representation
1	OR_TMP_REG TmpReg

Input arguments	Type	Range	Description
TmpReg	UINT	0–299	The temporary register number

No output arguments.

OR_NV_REG

The OR_NV_REG command makes a logical OR link between the non-volatile register value and the 16-bit accumulator content in logic memory. The result is saved in the 16-bit accumulator.

The OR process compares each bit in the 16-bit accumulator with the corresponding bit in the linked non-volatile register:

- If any compared bit equals 1, the result of the OR process for that bit number is also 1.
- If all compared bits equal 0, the result of the OR process for that bit number is 0.

Arguments	Representation
1	OR_NV_REG NVReg

Input arguments	Type	Range	Description
NVReg	UINT	0–63	The non-volatile register number

No output arguments.

XOR_K

The XOR_K command makes a logical exclusive OR link between a 16-bit constant value and the 16-bit accumulator content in logic memory. The result is saved in the 16-bit accumulator.

The XOR process compares each bit in the 16-bit accumulator with the corresponding bit in the linked 16-bit constant value and yields these results:

- If one bit equals 1 and the other equals 0, the result of the XOR process is 1.
- In all other cases, the result of the XOR process is 0.

Arguments	Representation
1	XOR_K KValue

Input arguments	Type	Range	Description
KValue	UINT	0–65,535	A constant value

No output arguments.

XOR_REG

The XOR_REG command makes a logical exclusive OR link between the register value and the 16-bit accumulator content in logic memory. The result is saved in the 16-bit accumulator.

The XOR process compares each bit in the 16-bit accumulator with the corresponding bit in the linked register and yields these results:

- If one bit equals 1 and the other equals 0, the result of the XOR process is 1.
- In all other cases, the result of the XOR process is 0.

Arguments	Representation
1	XOR_REG RegAddr

Input arguments	Type	Range	Description
RegAddr	UINT	0–9249	The register address

No output arguments.

XOR_TMP_REG

The XOR_TMP_REG command makes a logical exclusive OR link between the temporary register value and the 16-bit accumulator content in logic memory. The result is saved in the 16-bit accumulator.

The XOR process compares each bit in the 16-bit accumulator with the corresponding bit in the linked temporary register and yields these results:

- If one bit equals 1 and the other equals 0, the result of the XOR process is 1.
- In all other cases, the result of the XOR process is 0.

Arguments	Representation
1	XOR_TMP_REG TmpReg

Input arguments	Type	Range	Description
TmpReg	UINT	0–299	The temporary register number

No output arguments.

XOR_NV_REG

The XOR_NV_REG command makes a logical exclusive OR link between the non-volatile register value and the 16-bit accumulator content in logic memory. The result is saved in the 16-bit accumulator.

The XOR process compares each bit in the 16-bit accumulator with the corresponding bit in the linked non-volatile register and yields these results:

- If one bit equals 1 and the other equals 0, the result of the XOR process is 1.
- In all other cases, the result of the XOR process is 0.

Arguments	Representation
1	XOR_NV_REG NVReg

Input arguments	Type	Range	Description
NVReg	UINT	0–63	The non-volatile register number

No output arguments.

ON_SET_REG

The ON_SET_REG command copies the value of the 16-bit accumulator to a specified register on a rising edge of the 1-bit accumulator.

Arguments	Representation
2	ON_SET_REG RegAddr TmpReg

No input argument.

Output arguments	Type	Range/Bit	Description
RegAddr	UINT	0–9249	The register address
TmpReg	UINT	Bit3	1-bit accumulator history bit

ON_SET_TMP_REG

The ON_SET_TMP_REG command copies the value of the 16-bit accumulator to a specified temporary register on a rising edge of the 1-bit accumulator.

Arguments	Representation
2	ON_SET_TMP_REG TmpReg1 TmpReg2

No input argument.

Output arguments	Type	Range/Bit	Description
TmpReg1	UINT	0–299	The temporary register number
TmpReg2	UINT	Bit3	1-bit accumulator history bit

ON_SET_NV_REG

The ON_SET_NV_REG command copies the value of the 16-bit accumulator to a specified non-volatile register on a rising edge of the 1-bit accumulator.

Arguments	Representation
1	ON_SET_NV_REG NVReg1 NVReg2

No input argument.

Output arguments	Type	Range/Bit	Description
NVReg1	UINT	0–63	The non-volatile register number
NVReg2	UINT	Bit3	1-bit accumulator history bit

Timer Logic Commands

Overview

The custom logic editor uses the following timer commands:

- TIMER_SEC
- TIMER_TENTHS
- TIMER_K_SEC
- TIMER_K_TENTHS

NOTE: When a custom logic file is uploaded to the simulator and a timer is enabled, it will function as expected. However, once a timer has been enabled and executed, it cannot be enabled again. This is a limitation of the TIMER_SEC and TIMER_TENTHS. To enable the timer again, the custom logic file should be uploaded again. It is recommend that this limitation is considered when designing and testing the custom logic files when working with timers.

TIMER_SEC

The TIMER_SEC command:

- Counts time in seconds, up to the number of counts specified by a temporary register
- Calculates the end time in a second temporary register
- Is enabled by, and reports its counting status to, a third temporary register

Arguments	Representation
3	TIMER_SEC TmpReg1 TmpReg2 TmpReg3

Input Argument	Type	Range/Bit	Description
TmpReg1	UINT	0–65,535	Timer preset value
TmpReg3	UINT	Bit0	• Starts the timer on a rising edge • Stops the timer on a falling edge

Output Argument	Type	Range/Bit	Description
TmpReg2	UINT	0–65,535	Calculated end time
TmpReg3	UINT	Bit1	Timer done: • Bit set when timer reaches TmpReg2 • Bit reset when: ◦ TmpReg3.Bit0 is reset ◦ Power is cycled
		Bit2	Timer execution in progress Bit reset when timer reaches TmpReg2
		Bit3	TmpReg3.Bit0 history bit
		Bit4	Reserved

TIMER_TENTHS

The TIMER_TENTHS command:

- Counts time in tenths of seconds, up to the number of counts specified by a temporary register
- Calculates the end time in a second temporary register
- Is enabled by, and reports its counting status to, a third temporary register

NOTE: The multiplication factor for the `TIMER_TENTHS` command is 10. For example, if the required value is 6 seconds, the input value should be multiplied by a factor of 10, meaning the input value provided should be 60 seconds.

Arguments	Representation
3	`TIMER_TENTHS TmpReg1 TmpReg2 TmpReg3`

Input Argument	Type	Range/Bit	Description
`TmpReg1`	`UINT`	0–65,535	Timer preset value
`TmpReg3`	`UINT`	Bit0	• Starts the timer on a rising edge • Stops the timer on a falling edge

Output Argument	Type	Range/Bit	Description
`TmpReg2`	`UINT`	0–65,535	Calculated end time
`TmpReg3`	`UINT`	Bit1	Timer done: • Bit set when timer reaches `TmpReg2` • Bit reset when: ◦ `TmpReg3.Bit0` is reset ◦ Power is cycled
		Bit2	Timer execution in progress Bit reset when timer reaches `TmpReg2`
		Bit3	`TmpReg3.Bit0` history bit
		Bit4	Reserved

TIMER_K_SEC

The `TIMER_K_SEC` command:

- Counts time in seconds, up to the number of counts specified by a constant value
- Calculates the end time in a temporary register
- Is enabled by, and reports its counting status to, a second temporary register

Arguments	Representation
3	`TIMER_K_SEC KValue TmpReg1 TmpReg2`

Input Argument	Type	Range/Bit	Description
`KValue`	`UINT`	0–65,535	Timer preset value
`TmpReg2`	`UINT`	Bit0	• Starts the timer on a rising edge • Stops the timer on a falling edge

Output Argument	Type	Range/Bit	Description
TmpReg1	UINT	0–65,535	Calculated end time
TmpReg2	UINT	Bit1	Timer done: • Bit set when timer reaches TmpReg1 • Bit reset when: ◦ TmpReg2.Bit0 is reset ◦ Power is cycled
		Bit2	Timer execution in progress Bit reset when timer reaches TmpReg1
		Bit3	TmpReg2.Bit0 history bit
		Bit4	Reserved

TIMER_K_TENTHS

The TIMER_K_TENTHS command:

- Counts time in tenths of seconds, up to the number of counts specified by a constant value
- Calculates the end time in a temporary register
- Is enabled by, and reports its counting status to, a second temporary register

Arguments	Representation
3	TIMER_K_TENTHS KValue TmpReg1 TmpReg2

Input Argument	Type	Range/Bit	Description
KValue	UINT	0–65,535	Timer preset value
TmpReg2	UINT	Bit0	• Starts the timer on a rising edge • Stops the timer on a falling edge

Output Argument	Type	Range/Bit	Description
TmpReg1	UINT	0–65,535	Calculated end time
TmpReg2	UINT	Bit1	Timer done: • Bit set when timer reaches TmpReg1 • Bit reset when: ◦ TmpReg2.Bit0 is reset ◦ Power is cycled
		Bit2	Timer execution in progress Bit reset when timer reaches TmpReg1
		Bit3	TmpReg2.Bit0 history bit
		Bit4	Reserved

Latch Logic Commands

Overview

The custom logic editor uses the following latch commands:

- LATCH
- LATCH_NV

LATCH

The LATCH command:

- Stores a Boolean value (0 or 1) in a temporary register
- Provides a method for setting and resetting the stored value
- Saves the clear and set status from the previous scan

Arguments	Representation
1	LATCH TmpReg

Input Argument	Type	Bit	Description
TmpReg	UINT	Bit1	Sets the TmpReg.Bit**0** to 1 on a rising edge
		Bit2	Resets the TmpReg.Bit**1** to 0 on a falling edge

Output Argument	Type	Bit	Description
TmpReg	UINT	Bit0	State of the latch
		Bit3	TmpReg.Bit1 history bit
		Bit4	TmpReg.Bit2 history bit

LATCH_NV

The LATCH_NV command:

- Stores a Boolean value (0 or 1) in a non-volatile register
- Provides a method for setting and resetting the stored value
- Saves the clear and set status from the previous scan

Use the LATCH_NV command, instead of the LATCH command, to retain the latch state during a power cycle.

Arguments	Representation
1	LATCH_NV NVReg

Input Argument	Type	Bit	Description
NVReg	UINT	Bit1	Sets the TmpReg.Bit0 to 1 on a rising edge
		Bit2	Resets the TmpReg.Bit0 to 0 on a rising edge

Output Argument	Type	Bit	Description
NVReg	UINT	Bit0	State of the latch
		Bit3	TmpReg.Bit1 history bit
		Bit4	TmpReg.Bit2 history bit

Counter Logic Commands

Overview

The custom logic editor uses the following counter logic commands:

- COUNTER
- COUNTER_NV

COUNTER

The COUNTER command:

- Increments or decrements a count value
- Provides a method for setting the count value to a preset value
- Indicates when the count value equals 0
- Indicates the relationship between the count value and the preset value - equal to, greater than or less than
- Saves the increment, decrement and set status from the previous scan

Argu-ments	Representation
3	COUNTER TmpReg1 KValue TmpReg2

Input Argument	Type	Range/Bit	Description
KValue	UINT	0–65,535	Counter preset value
TmpReg2	UINT	Bit4	Increments the counter current value on a rising edge. Counter current value shall roll over from 0 to 65,535.
		Bit5	Decrements the counter current value on a falling edge. Counter current value shall roll over from 65,535 to 0.
		Bit6	Sets the current counter value to the preset value on a rising edge

Output Argument	Type	Range/Bit	Description
TmpReg1	UINT	0–65,535	Counter current value
TmpReg2	UINT	Bit0	The counter current value is 0: TmpReg1=0
		Bit1	The counter current value is lower than the preset value: TmpReg1<KValue
		Bit2	The counter current value is equal to the preset value: TmpReg1=KValue
		Bit3	The counter current value is greater than the preset value: TmpReg1>KValue
		Bit7	TmpReg2.Bit4 history bit
		Bit8	TmpReg2.Bit5 history bit
		Bit9	TmpReg2.Bit6 history bit

COUNTER_NV

The COUNTER_NV command:

- Increments or decrements a count value

- Provides a method for setting the count value to a preset value
- Indicates when the count value equals 0
- Indicates the relationship between the count value and the preset value - equal to, greater than or less than
- Saves the increment, decrement and set status from the previous scan

Use the COUNTER_NV command, instead of the COUNTER command, to retain the count during a power cycle.

Arguments	Representation
3	COUNTER NVReg1 KValue NVReg2

Input Argument	Type	Range/Bit	Description
KValue	UINT	0–65,535	Counter preset value
NVReg2	UINT	Bit4	Increments the counter current value on a rising edge. Counter current value shall roll over from 0 to 65,535.
		Bit5	Decrements the counter current value on a falling edge. Counter current value shall roll over from 65,535 to 0.
		Bit6	Sets the current counter value to the preset value on a rising edge

Output Argument	Type	Range/Bit	Description
NVReg1	UINT	0–65,535	Counter current value
NVReg2	UINT	Bit0	The counter current value is 0: NVReg1=0
		Bit1	The counter current value is lower than the preset value: NVReg1<KValue
		Bit2	The counter current value is equal to the preset value: NVReg1=KValue
		Bit3	The counter current value is greater than the preset value: NVReg1>KValue
		Bit7	NVReg2.Bit4 history bit
		Bit8	NVReg2.Bit5 history bit
		Bit9	NVReg2.Bit6 history bit

Math Logic Commands

Overview

The custom logic editor uses the following math commands:

- ON_ADD
- ON_SUB
- ON_MUL
- ON_DIV

ON_ADD

The ON_ADD command performs unsigned addition when the 1-bit accumulator transitions from 0 to 1. It adds the value from Argument 1 to the 16-bit accumulator value, then posts the result back to the value in Argument 1.

Status register:

- Indicates an overflow if the result of the addition process exceeds 65,535
- Indicates the status of the 1-bit-accumulator from the previous scan

Arguments	Representation
2	ON_ADD TmpReg1 TmpReg2

Input Argument	Type	Range/Bit	Description
TmpReg1	UINT	0–65,535	Value to add to the 16-bit accumulator

Output Argument	Type	Range/Bit	Description
TmpReg1	UINT	0–65,535	Result of the addition operation
TmpReg2	UINT	Bit0	Overflow: the result of the addition is greater than 65,535. In this case, the result of the addition is equal to Argument 1 + 65,536.
		Bit3	1-bit accumulator history bit

ON_SUB

The ON_SUB command performs unsigned subtraction when the 1-bit accumulator transitions from 0 to 1. It subtracts the 16-bit accumulator value from the value in Argument 1, then posts the result back to the value in Argument 1.

Status register:

- Indicates an underflow if the result of the subtraction process is less than 0
- Indicates the status of the 1-bit-accumulator from the previous scan

Arguments	Representation
2	ON_SUB TmpReg1 TmpReg2

Input Argument	Type	Range/Bit	Description
TmpReg1	UINT	0–65,535	Value to subtract from the 16-bit accumulator

Output Argument	Type	Range/Bit	Description
TmpReg1	UINT	0–65,535	Result of the subtraction operation
TmpReg2	UINT	Bit0	Underflow: the result of the subtraction is less than 0. In this case, the true result of the operation equals the value output to Argument 1 - 65,536.
		Bit3	1-bit accumulator history bit

ON_MUL

The ON_MUL command performs unsigned multiplication when the 1-bit accumulator transitions from 0 to 1. The ON_MUL procedure multiplies the value from Argument 2 against the 16-bit accumulator value, then posts the result back to Argument 1 (most significant word) and Argument 2 (least significant word).

Status register indicates the status of the 1-bit accumulator from the previous scan.

Arguments	Representation
3	ON_MUL TmpReg1 TmpReg2 TmpReg3

Input Argument	Type	Range/Bit	Description
TmpReg2	UINT	0–65,535	Value to be multiply with the 16-bit accumulator

Output Argument	Type	Range/Bit	Description
TmpReg1 and TmpReg2	UINT	0–65,535	Result of the multiplication operation: • TmpReg1 holds the most significant word • TmpReg2 holds the least significant word
TmpReg3	UINT	Bit3	1-bit accumulator history bit

ON_DIV

The ON_DIV command performs unsigned division when the 1-bit accumulator transitions from 0 to 1. The ON_DIV procedure divides the combined value of Argument 1 and Argument 2 by the 16-bit accumulator value, then posts the result back to Argument 1 (most significant word) and Argument 2 (least significant word).

Status register indicates:

- An overflow if division is by 0
- The status of the 1-bit accumulator from the previous scan

Arguments	Representation
3	ON_DIV TmpReg1 TmpReg2 TmpReg3

Input Argument	Type	Range/Bit	Description
TmpReg1 and TmpReg2	UINT	0–65,535	Value to be divided by the 16-bit accumulator

Output Argument	Type	Range/Bit	Description
`TmpReg1` and `TmpReg2`	`UINT`	0–65,535	Result of the division operation: • `TmpReg1` holds the most significant word • `TmpReg2` holds the least significant word
`TmpReg3`	`UINT`	Bit0	Division by 0
		Bit3	1-bit accumulator history bit

Custom Logic Program Examples

What's in This Chapter

How to Check Timers and Multiply Commands 143
How to Create a Truth Table 144

How to Check Timers and Multiply Commands

Overview

When customizing your application you may need to check timers and multiply commands.

Checking Timers and Multiply Commands with a Custom Logic Program

The following diagram gives the custom logic program in Text View of how to check timers and multiply commands:

```
 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
```

How to Create a Truth Table

Overview

When customizing your application, you may need to create a truth table.

Creating a Truth Table with a Custom Logic Program

The following diagram gives the custom logic program in Text View of the creation of a truth table:

```
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
//
```

Creating a Truth Table with a Custom Logic Program (cont'd)

```
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
```

Creating a Truth Table with a Custom Logic Program (cont'd)

```
//
//***************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
```

Function Block Diagram Language

What's in This Part

Overview of FBD Language 148
FBD Elements 151
Programming with FBD Language 166
Manipulating FBD Blocks 172
FBD Editor Display Options 176

Overview of FBD Language

What's in This Chapter

Introduction to FBD Editor 149

Introduction to FBD Editor

Overview

The FBD editor is a feature of the TeSys Tera DTM Library. Use the FBD Editor to view an existing FBD program file or to create an FBD program file using FBD language, rather than an instruction-based text programming language.

Creating an FBD Program

To open the FBD editor, select **Device > FB Diagram > New FB Diagram** or select the **FB Diagram** tab. The FBD editor appears in the main window.

Saving an FBD Program

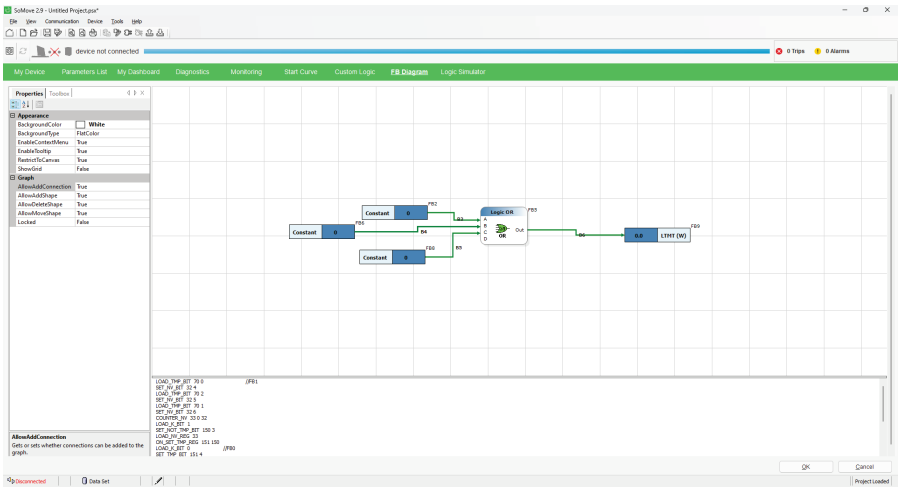
Before compiling the FBD program, you must save it. To save the program you created or edited, select **Device > FB Diagram > Save FB Diagram as**.

NOTE: The file is saved with the extension **.Gef*.

FBD Editor User Interface

The FBD editor is available even when the TeSys Tera DTM Library is in connected mode. However, many of the menu items are enabled only when an FBD program is open in the FBD editor.

The below window shows FBD program open in the FBD editor:



Workspace

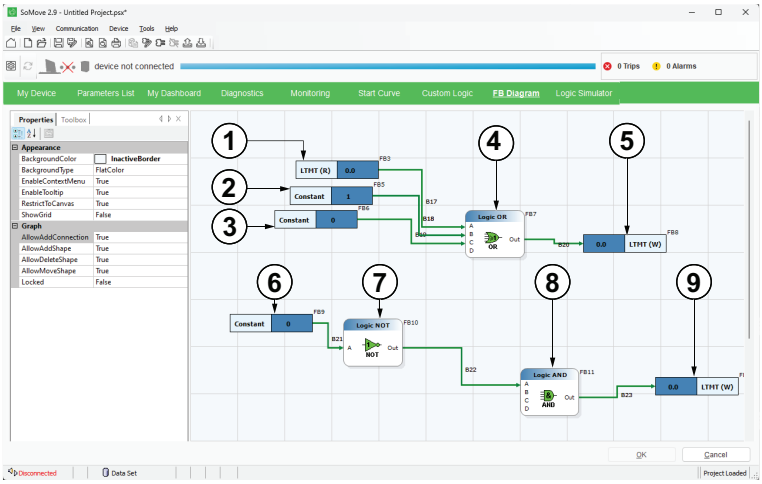
FBD programs are edited and created in the workspace.

The workspace is made up of two elements:

- Blocks
- Wires to link the blocks

Running FBD Programs

FBD programs are run line by line, from the left to the right and from top to bottom. In the example below, the instructions are carried out or performed from instruction 1 to instruction 5 and from instruction 6 to instruction 9, in the order indicated by numbers.



FBD Elements

What's in This Chapter

Starter Types 152
Computation Blocks 154
Inputs Blocks 157
Function Blocks 160
Logic Blocks 163
Outputs Blocks 163

Starter Types

Overview

The FBD editor allows the use of starter block accessible through the **Starter Types** bar in the Toolbox:

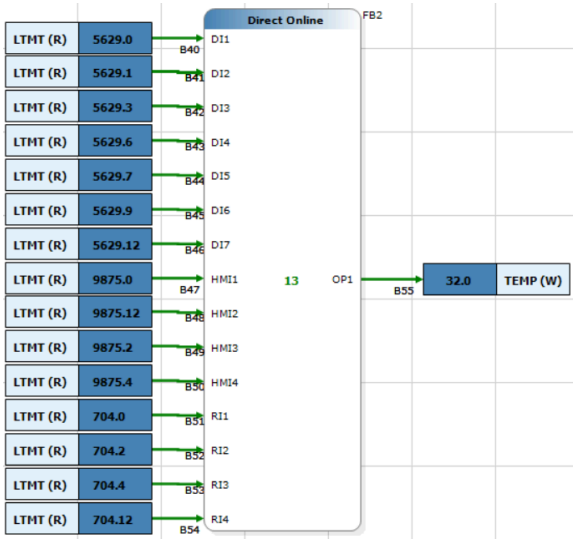
Block	Description
	`Direct Online`
	`Reversible Direct Online`
	`Star Delta`

NOTE: Placing cursor over the icon will reveal a tool tip defining the icon. This will help you distinguish which type of block is represented by that icon.

NOTE: Only one type of starter can be used at a time.

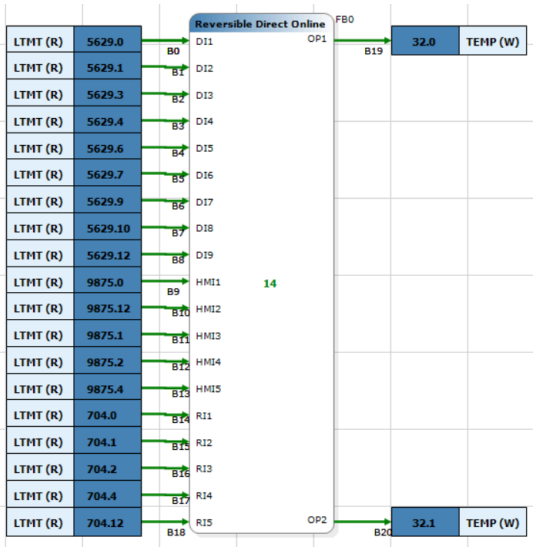
Direct Online

Using the  block sets the starter type as `Direct Online`.

FBD Display	Arguments	Description
	Digital input signal	• DI1 (5629.0): Local-START> DI • DI2 (5629.1): Local-STOP DI • DI3 (5629.3): DI Mode Selection 1 • DI4 (5629.6): Remote-START> DI • DI5 (5629.7): Remote-STOP DI • DI6 (5629.9): DI Mode Selection 2 • DI7 (5629.12): Run DI
	HMI signals	• HMI1 (9875.0): HMI_START • HMI2 (9875.12): HMI_STOP • HMI3 (9875.2): HMI Mode Selection 1 • HMI4 (9875.4): HMI Mode Selection 2
	Communication signals	• RI1(704.0): COMM Start • RI2 (704.2): Remote Mode Selection 1 • RI3 (704.4): Remote Mode Selection 2 • RI4 (704.12): COMM Stop
	Output signal	• OP1 (32.0): CL “Run1 Cde” information

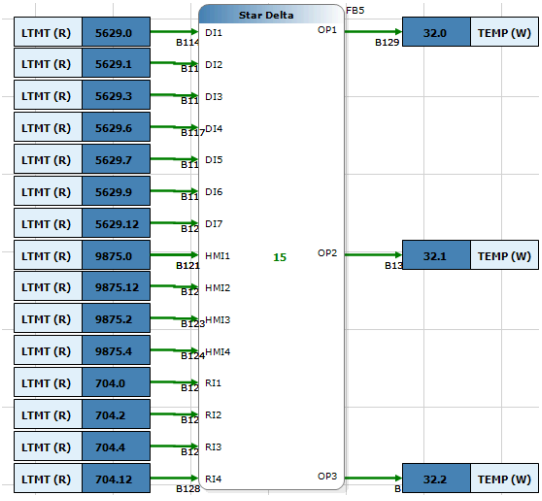
Reversible Direct Online

Using the  block sets the starter type as `Reversible Direct Online`.

FBD Display	Arguments	Description
	Digital input signal	• DI1 (5629.0): Local-START> DI • DI2 (5629.1): Local-STOP DI • DI3 (5629.3): DI Mode Selection 1 • DI4 (5629.4): Local-START< DI • DI5 (5629.6): Remote-START> DI • DI6 (5629.7): Remote-STOP DI • DI7 (5629.9): DI Mode Selection 2 • DI8 (5629.10): Remote-START< DI • DI9 (5629.12): Run DI
	HMI signals	• HMI1 (9875.0): HMI_START > • HMI2 (9875.12): HMI_STOP, • HMI3 (9875.1): HMI_START < • HMI4 (9875.2): HMI Mode Selection 1 • HMI5 (9875.4): HMI Mode Selection 2
	Communication signals	• RI1(704.0): COMM Start > • RI2 (704.1): COMM Start < • RI3 (704.2): Remote Mode Selection 1 • RI4 (704.4): Remote Mode Selection 2 • RI5 (704.12): COMM Stop
	Output signal	• OP1 (32.0): CL “Run1 Cde” information • OP2 (32.1): CL “Run2 Cde” information

Star Delta

Using the  block sets the starter type as `Star Delta`.

FBD Display	Arguments	Description
	Digital input signal	• DI1 (5629.0): Local-START> DI • DI2 (5629.1): Local-STOP DI • DI3 (5629.3): DI Mode Selection 1 • DI4 (5629.6): Remote-START> DI • DI5 (5629.7): Remote-STOP DI • DI6 (5629.9): DI Mode Selection 2 • DI7 (5629.12): Run DI
	HMI signals	• HMI1 (9875.0): HMI_START > • HMI2 (9875.12): HMI_STOP, • HMI3 (9875.2): HMI Mode Selection 1 • HMI4 (9875.4): HMI Mode Selection 2
	Communication signals	• RI1(704.0): COMM Start > • RI2 (704.2): Remote Mode Selection 1 • RI3 (704.4): Remote Mode Selection 2 • RI4 (704.12): COMM Stop
	Output signal	• OP1 (32.0): CL “Run1 Cde” information • OP2 (32.1): CL “Run2 Cde” information • OP3 (32.2): CL “Run3 Cde” information

Computation Blocks

Overview

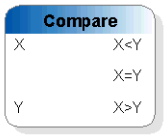
The FBD editor uses various computation blocks accessible through the **Computation** bar in the Toolbox:

Block	Description
	Compare
	Add
	Division
	Multiplication
	Subtraction

NOTE: Placing cursor over the icon will reveal a tool tip defining the icon. This will help you distinguish which type of block is represented by that icon.

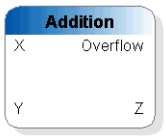
Compare Block

The  block compares two 16-bit register values.

FBD symbol	Arguments	Description
	Inputs	• X: 16-bit unsigned register value (0 to 65,535). • Y: 16-bit unsigned register value (0 to 65,535).
	Outputs	• X < Y: ON/OFF temporary bit that is ON if the value X is less than the value Y. • X = Y: ON/OFF temporary bit that is ON if the value X is equal to the value Y. • X > Y: ON/OFF temporary bit that is ON if the value X is greater than the value Y.

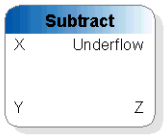
Add Block

The  block performs an unsigned addition of two 16-bit register values.

FBD symbol	Arguments or Example	Description
	Inputs	• X: 16-bit unsigned register value (0 to 65,535). • Y: 16-bit unsigned register value (0 to 65,535).
	Outputs	• Z: 16-bit unsigned register result (Z = X + Y). • Overflow: ON or OFF value which when set ON caries a value of 65,536. The value is initialized to OFF.
	Example	Assuming X = 60,000 and Y = 7,000, the overflow will be ON because 60,000 + 7,000 = 67,000, which is superior to 65,536. The result Z is then equal to 1,464 (1,464 + 65,356 = 67,000).

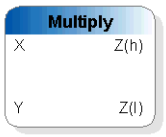
Subtraction Block

The  block performs an unsigned subtraction of two 16-bit register values.

FBD symbol	Arguments or Example	Description
	Inputs	• X: 16-bit unsigned register value (0 to 65,535). • Y: 16-bit unsigned register value (0 to 65,535).
	Outputs	• Z: 16-bit unsigned register result (Z = X – Y). • Underflow: ON or OFF value, which when set ON, caries a value of negative 65,536. The value is initialized to OFF.
	Example	Assuming X = 5 and Y = 10, the underflow will be ON because the result is negative. The result Z is then equal to 65,531 (65,531 – 65,536 = –5)

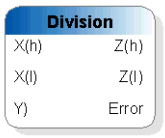
Multiplication Block

The  block performs an unsigned multiplication of two 16-bit register values.

FBD symbol	Arguments or Example	Description
	Inputs	• X: 16-bit unsigned register value (0 to 65,535) • Y: 16-bit unsigned register value (0 to 65,535)
	Outputs	• Z(h): 16 most significant bits of the 32-bit result, Z(h) = (X * Y) / 65,536 • Z(l): 16 least significant bits of the 32-bit result, Z(l) = (X * Y) – Z(h) * 65,536
	Example	Assuming X = 20,000 and Y = 10, the result will be Z(h) = 3 and Z(l) = 3,392 because 200,000 = 3 * 65,536 + 3,392

Division Block

The  block performs an unsigned division of two 16-bit register values.

FBD symbol	Arguments or Example	Description
	Inputs	• X(h): 16 most significant bits of an unsigned register value (0 to 65,535). • X(l): 16 least significant bits of an unsigned register value (0 to 65,535). • Y: 16-bit unsigned register divisor (0–65,535).
	Outputs	• Z(h): 16 most significant bits of the 32-bit quotient, Z(h) = (X / Y) / 65,536 • Z(l): 16 least significant bits of the 32-bit quotient, Z(l) = (X / Y) – Z(h) * 65,536 • Detected Error: ON or OFF value, which is set ON when a division by zero occurs. This value is initialized to OFF.
	Example	Assuming X(h) = 3, X(l) = 3,392 and Y = 40, the result will be Z(h) = 0 and Z(l) = 5,000 because X(h) * 65,536 + X(l) = 3 * 65,536 + 3,392 = 200,000 and 200,000 /Y = 5,000 = 0 * 65,536 + 5,000

Inputs Blocks

Overview

The FBD editor uses various inputs blocks accessible through the **Inputs** bar in the Toolbox:

Block	Description
	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

NOTE: Placing cursor over the icon will reveal a tool tip defining the icon. This will help you distinguish which type of block is represented by that icon.

Constant Bit Block

The  block is used to set other blocks' inputs to 0 or 1.

FBD symbol	Arguments	Description
	Properties	• Constant bit value 0 or 1 (True = 1 and False = 0).
	Outputs	• Constant value 0 or 1 (True = 1 and False = 0).

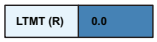
Constant Word Block

The  block is used to set other blocks' inputs values (0 to 65,535).

FBD symbol	Arguments	Description
	Properties	• Constant value from 0 to 65,535.
	Outputs	• Constant value from 0 to 65,535.

Register Bit In Block

The  block enables the reading and use of a register bit value from LTMT main unit registers.

FBD symbol	Arguments	Description
	Properties	• Any register addresses from 0 to 9249 (from 0x0000 to 0x2421), 9875 (0x2693), 9876 (0x2694) x Bit number from 0 to 15.
	Outputs	• Value 0 or 1 (True = 1 and False = 0).

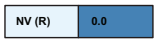
Register Word In Block

The  block enables the reading and use of a register value from LTMT main unit registers.

FBD symbol	Arguments	Description
	Properties	• Any register addresses from 0 to 9249 (from 0x0000 to 0x2421), 9875 (0x2693), 9876 (0x2694).
	Outputs	• Value from 0 to 65,535.

Register NV Bit Block

The  block enables the reading and use of a non-volatile register bit value.

FBD symbol	Arguments	Description
	Properties	• Any non-volatile register from 0 to 63 x Bit number from 0 to 15.
	Outputs	• Value 0 or 1 (True = 1 and False = 0).

Register NV Word In Block

The  block enables the reading and use of a non-volatile register value.

FBD symbol	Arguments	Description
	Properties	• Any non-volatile register from 0 to 63.
	Outputs	• Value from 0 to 65,535.

Register Temp Bit In Block

The 

block enables the reading and use of a temporary register bit value.

FBD symbol	Arguments	Description
	Properties	• Any temporary register from 0 to 299 x Bit number from 0 to 15. •
	Outputs	• Value 0 or 1 (True = 1 and False = 0).

Temp Word In Block

The 

block enables the reading and use of a temporary register value.

FBD symbol	Arguments	Description
	Properties	• Any temporary register from 0 to 299.
	Outputs	• Value from 0 to 65,535.

Function Blocks

Overview

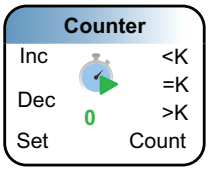
The FBD editor uses various functions blocks accessible through the **Function** bar in the Toolbox:

Block	Description
	Counter
	Counter NV
	Volatile Latch
	Non Volatile Latch
	Multiplexer
	TimerSeconds
	TimerTenthSeconds

NOTE: Placing cursor over the icon will reveal a tool tip defining the icon. This will help you distinguish which type of block is represented by that icon.

Counter Block

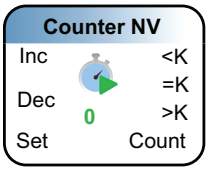
The  function performs a comparative count, saving both the counter current and counter preset values to temporary registers.

FBD symbol	Arguments	Description
	Properties	K: Counter preset value (UINT 0 to 65,535).
	Inputs	• Inc: Increments the counter current value on a rising edge. Counter current value shall roll over from 0 to 65,535. • Dec: Decrements the counter current value on a falling edge. Counter current value shall roll over from 65,535 to 0. • Set: Sets the current counter value to the preset value on a rising edge.
	Outputs	• Count: Counter current value (UINT 0 to 65,535). Count is initialized to zero on power-up. • <K: The counter current value is lower than the preset value K. • =K: The counter current value is equal to the preset value K. • >K: The counter current value is greater than the preset value K.

NOTE: The Counter preset value range is from 0 to 65,535. Cascading counters and compare functions can be used if you need larger values or multiple preset values.

Counter NV Block

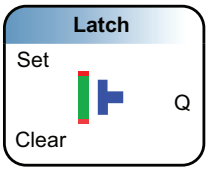
The  function performs a comparative count, saving both the counter current and counter preset values to non-volatile registers.

FBD symbol	Arguments	Description
	Properties	K: Counter preset value (UINT 0 to 65,535).
	Inputs	• Inc: Increments the counter current value on a rising edge. Counter current value shall roll over from 0 to 65,535. • Dec: Decrements the counter current value on a falling edge. Counter current value shall roll over from 65,535 to 0. • Set: Sets the current counter value to the preset value on a rising edge.
	Outputs	• Count: Counter current value (UINT 0 to 65,535). This value is saved in non-volatile memory and initialized to the previous value on power-up. • <K: The counter current value is lower than the preset value K. • =K: The counter current value is equal to the preset value K. • >K: The counter current value is greater than the preset value K.

NOTE: The Counter preset value range is from 0 to 65,535. Cascading counters and compare functions can be used if you need larger values or multiple preset values

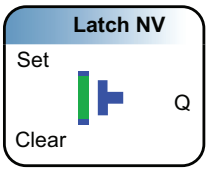
Volatile Latch Block

The  function records and retains signal history in a temporary register.

FBD symbol	Arguments	Description
	Inputs	• Set: ON/OFF input value. The latch value is set ON when this input transitions from OFF to ON. • Clear: ON/OFF input value. The latch value is set OFF when this input transitions from OFF to ON.
	Outputs	• Q: ON or OFF latch value which represents the state of this latch. This value remains ON/OFF until the next rising edge of Set or Clear. This value is initialized to OFF.

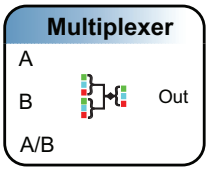
Non Volatile Latch Block

The  function records and retains signal history in a non-volatile register.

FBD symbol	Arguments	Description
	Inputs	• Set: ON/OFF input value. The latch value is set ON when this input transitions from OFF to ON. • Clear: ON/OFF input value. The latch value is set OFF when this input transitions from OFF to ON.
	Outputs	• Q: ON or OFF non-volatile register bit value that represents the state of this latch. This value remains ON/OFF until the next rising edge of Set or Clear. This value is saved in non-volatile memory and initialized to previous state on power-up.

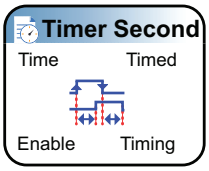
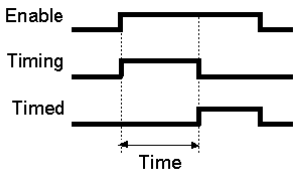
Multiplexer Block

The  function enables you to choose between two 16-bit unsigned values.

FBD symbol	Arguments	Description
	Inputs	• A: 16-bit unsigned value (0 to 65,535). • B: 16-bit unsigned value (0 to 65,535). • A/B: ON/OFF input value that selects value A or B.
	Outputs	• Out: Selected 16-bit value: ◦ If A/B is ON then Out = A. ◦ If A/B is OFF then Out = B.

Timer Seconds Block

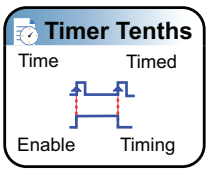
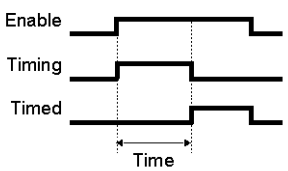
The  function measures time in intervals of seconds.

FBD symbol	Timing diagram	Argu-ments	Description
		Inputs	• Time: 16-bit unsigned value (0 to 65,535) that specifies time period in seconds. • Enable: ON/OFF input value. The time period is loaded on the rising edge of the Enable input. Time measuring continues while Enable is ON. Timing stops and outputs are OFF when Enable is OFF.
		Outputs	• Timed - ON/OFF value which turns ON after time period expires. It is OFF while enable is OFF • Timing - ON/OFF value that is ON while ,Enable is ON & while measuring time. It is OFF after measuring time expires. **NOTE:** Both outputs can never be simultaneously ON.

Timer Tenths Seconds Block

The  function measures time in intervals of tenths of seconds.

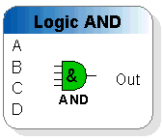
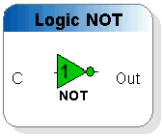
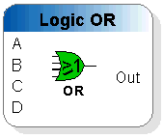
NOTE: The multiplication factor for the `TIMER_TENTHS` command is 10. For example, if the required value is 6 seconds, the input value should be multiplied by a factor of 10, meaning the input value provided should be 60 seconds.

FBD symbol	Timing diagram	Argu-ments	Description
		Inputs	• Time: 16-bit unsigned value (0 to 65,535) that specifies time periods in tenths of seconds. • Enable: ON/OFF input value. The time period is loaded on the rising edge of the Enable input. Time measuring continues while Enable is ON. Timing stops and outputs are OFF when Enable is OFF.
		Outputs	• Timed - ON/OFF value which turns ON after time period expires. It is OFF while enable is OFF • Timing - ON/OFF value that is ON while ,Enable is ON & while measuring time. It is OFF after measuring time expires. **NOTE:** Both outputs can never be simultaneously ON.

Logic Blocks

Overview

The FBD Editor uses various logic blocks accessible through the **Logic blocks** bar in the Toolbox:

Function	Icon	FBD symbol	Description
AND			If all the inputs (ON or OFF values, respectively 1 or 0) are ON, the output is ON. If at least one input is OFF, the output is OFF. **NOTE:** Unconnected inputs are assumed to be ON.
NOT			If the input (ON or OFF values, respectively 1 or 0) is ON, the output is OFF. If the input is OFF, the output is ON.
OR			If at least one input (ON or OFF values, respectively 1 or 0) is ON, the output is ON. If all the inputs are OFF, the output is OFF. **NOTE:** Unconnected inputs are assumed to be OFF.

NOTE: Placing cursor over the icon will reveal a tool tip defining the icon. This will help you distinguish which type of block is represented by that icon.

Outputs Blocks

Overview

The FBD editor uses various outputs blocks accessible through the **Outputs** bar in the Toolbox:

Block	Description
	Register Bit Out
	Register Word Out
	Register NV Bit Out
	Register NV Word Out
	Register Temp Bit Out
	Temp Word Out

NOTE: Placing cursor over the icon will reveal a tool tip defining the icon. This will help you distinguish which type of block is represented by that icon.

Register Bit Out Block

The 

block is used to set an LTMT main unit register bit value to 0 or 1 from the LTMT main unit registers.

FBD symbol	Arguments	Description
	Properties	• a: Any register addresses from 0 to 9249 (from 0x0000 to 0x2421), 9875 (0x2693), 9876 (0x2694). • b: Bit number from 0 to 15.
	Inputs	• 0 or 1 (ON=1 and OFF=0)

Register Word Out Block

The 

block is used to set an LTMT main unit register value from the LTMT main unit registers.

FBD symbol	Arguments	Description
	Properties	• a: Any register addresses from 0 to 9249 (from 0x0000 to 0x2421), 9875 (0x2693), 9876 (0x2694).
	Inputs	• 16-bit unsigned value from 0 to 65,535.

Register NV Bit Out Block

The 

block is used to set a non-volatile register bit value to 0 or 1.

FBD symbol	Arguments	Description
	Properties	• a: Any non-volatile register from 0 to 63. • b: Bit number from 0 to 15.
	Inputs	• 0 or 1 (ON=1 and OFF=0)

Register NV Word Out Block

The 

block is used to set a non-volatile register value.

FBD symbol	Arguments	Description
	Properties	• a: Any non-volatile register from 0 to 63.
	Inputs	• 16-bit unsigned value from 0 to 65,535

Register Temp Bit Out Block

The 

block is used to set a temporary register bit value to 0 or 1.

FBD symbol	Arguments	Description
	Properties	• a: Any temporary register from 0 to 299. • b: Bit number from 0 to 15.
	Inputs	• 0 or 1 (ON=1 and OFF=0)

Temp Word Out Block

The 

block is used to set a temporary register value.

FBD symbol	Arguments	Description
	Properties	• a: Any temporary register from 0 to 299.
	Inputs	• 16-bit unsigned value from 0 to 65,535

Programming with FBD Language

What's in This Chapter

Inserting FBD Blocks 167
Creation of Links between Blocks 168
FBD Blocks Properties 170
FBD Resource Management 171

Inserting FBD Blocks

Overview

To create an FBD program, insert blocks into the workspace, then link them together. All types of blocks can be placed in the workspace.

Inserting Blocks from the Toolbox

The following procedure describes how to insert a block from the toolbox into the workspace:

1. Select **Device > FB Diagram > View > Toolbox** or select the **Toolbox** tab on the left side.
2. Select the type of block to insert:
   - Computation
   - Inputs
   - Function Blocks
   - Logic
   - Starter Types
   - Outputs
3. Left-click on the icon corresponding to the block to insert.
4. Drag and drop the block from the toolbox to the workspace.
5. Position the block in the required location on the workspace.
6. Repeat steps 2 to 5 to insert all the blocks required for the program.

Inserting Blocks from the Workspace

The following procedure describes how to insert a block directly from the workspace:

1. Right-click anywhere on a blank space in the workspace.

   Result: A menu appears and enables you to select the type of block you want to insert.
2. Select the type of block to insert:
   - Computation
   - Inputs
   - Function Blocks
   - Logic
   - Starter Types
   - Outputs
3. Left-click on the block you want to insert.
4. Position the block in the required location in the workspace.
5. Repeat steps 1 to 5 to insert all the blocks required for the program.

Creation of Links between Blocks

Overview

After you have positioned the blocks in the workspace, you can link them together. To do this, you link the output of a block to the input of another block. You can also loop an output back to the input of the same block.

General Rules

There are basic rules that apply when placing and connecting blocks:

- One or more connecting wires attached together form a wire node. This is indicated in the workspace by a red dot. If wires cross without a red connection dot, it means they are not connected.
- Only one output can be attached to each wire node.
- Connections between boolean and register data are prohibited.
- Data typically flows from left to right.

Link Between Blocks

The following procedure describes how to link blocks together:

1. Place the mouse over the first block.

   Result: One or more squares become visible on the block border, and the type of output (analog or boolean) is indicated.

2. Click the left mouse button and hold it down.
3. With the button held down, move the cursor over the input of the block you want to link to.

   Result: One or more squares become visible on the block border. If the square is green, a connection between the two blocks is possible. A red square indicates that a connection is not possible. The type of output (analog or boolean) is also indicated.

   NOTE: Inputs and outputs have to be of the same type: a Boolean output is linked to another Boolean output. If the inputs or outputs are not the same, the FBD editor will display a pop-up window to indicate that origins and destinations are not of the same type.

4. Release the mouse button.

   Result: A line and a number are shown between the two linked blocks.

5. Repeat steps 1 to 4 to link all the blocks.

Link Number

There are two types of wires:

- The Boolean wire, which will have a number beginning with B.
- The Register wire, which will have a number beginning with R.

The wire number is automatically incremented in chronological order.

FBD Blocks Properties

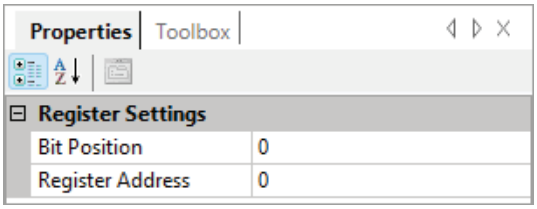
Overview

Each of the blocks has a properties window. To display this window, left-click on a block.

The Properties window consists of several tabs, separated in 1 or 2 categories, depending on the type of block:

- General settings, which contain the block ID and comments (common to all types of blocks.
- Specific settings, depending on the type of block (register settings for registers, counter settings for counters, etc.).

For example, if you want to display non-volatile register properties, select a non-volatile register block and left-click on it. The following window is displayed:



Comments

In the Comment zone, in the white box on the right of comments, you can enter a comment. Select any object or any free location in the workspace to save the comment.

Settings

Most blocks have a specific settings tab. In this tab, set the specific settings of blocks. These settings are described in detail in the help for each of the FBD blocks.

Properties Display

The properties of each block can be displayed in two different ways:

- By category, clicking  or
- By alphabetical order, clicking  .

FBD Resource Management

Overview

The LTMT main unit memory is equipped with the following resources:

- 9250 LTMT registers
- Logic memory space size equal to 8,192 words (16 bits)
- 300 temporary registers
- 64 non-volatile registers

Reserved Resources

The following table lists all reserved registers and their allocation. It also indicates how these registers are controlled:

Register type	Address range	Controlled by	Description
Temporary	0–69	User	Temporary storage of bit and registers assigned by you when creating an FBD program.
Temporary	70–299	FBD compiler	Reserved temporary registers for use by the compiler.
Non-volatile	0–31	User	Non-volatile bits or registers assigned by you when creating an FBD program.
Non-volatile	32–63	FBD compiler	Reserved non-volatile registers for use by the compiler.

Manipulating FBD Blocks

What's in This Chapter

Selecting Blocks..... 173
Deleting or Duplicating Objects..... 173

Selecting Blocks

Overview

When you add blocks to the workspace, you can select them to reposition them within the workspace.

Selecting One or More Blocks

The following table describes how to select one or more blocks:

If you would like to select...	Then
An isolated block	Click block.
Several contiguous blocks	Frame the blocks to be selected by defining a selection zone. **Result**: All of the selected blocks are highlighted with an orange outline. 
Several blocks in different areas of the workspace	Press the SHIFT key, then click the blocks to be selected while continuing to hold down the SHIFT key. **Result**: All of the selected blocks are highlighted with an orange outline.
All objects including wires	Select **Device > FB Diagram > FBD Editor > Select All** **NOTE:** The keyboard shortcut **CTRL+A** can also be used to select all objects.

Deleting or Duplicating Objects

Overview

Sometimes it may be necessary to delete a block or duplicate a block in the workspace.

Deleting Blocks

The following procedures describe how to delete one or more blocks:

1. Select the block(s) to be deleted.

   Result: The selected blocks are highlighted with an orange outline.

2. Press the DELETE key or select **Device > FB Diagram > FBD Editor > Delete**.

   Result: The selected blocks are deleted.

1. Select the block(s) to be deleted.

   Result: The selected blocks are highlighted with an orange outline.

2. Right-click on the selected block and select **Delete**.

   Result: The selected blocks are deleted.

Cut, Copy, or Paste Blocks

The following procedure describes how to cut, copy, or paste one or more blocks :

1. Select the block(s) to be modified.

   Result: The selected blocks are highlighted with an orange outline.

2. Select **Device > FB diagram > FBD editor** and select one of the following commands:
   - **Copy**
   - **Cut**
   - **Paste**

   Result: **Cut** deletes the selected blocks and stores them in the clipboard. **Copy** duplicates the selected blocks in the clipboard and **Paste** duplicates the clipboard contents on the workspace.

   NOTE: The keyboard shortcuts **CTRL+C**, and **CTRL+V** can also be used to copy the selected blocks and paste them.

Hide or Show Blocks

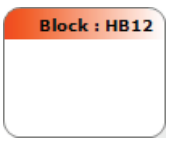
The following procedure describes how to hide one or more blocks :

1. Select the block(s) to be hidden.

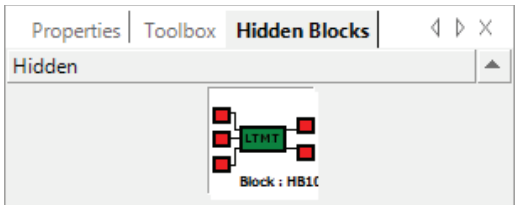
   Result: The selected blocks are highlighted with an orange outline.

2. Right-click on the selected block(s) and select **Hide/Show**.

   Result: The selected blocks are hidden and as displayed as follows:



3. The list of hidden block are displayed under **Hidden Blocks** as shown below:



NOTE: When multiple blocks are selected, all the blocks are grouped in the same block.

The following procedure describes how to show one or more hidden blocks :

1. Right-click on the hidden block(s) and select **Hide/Show**.

   Result: The selected blocks are Shown.

2. Alternatively, right-click on the hidden blocks under **Hidden Blocks** section and select **Hide/Show**.

   Result: The selected blocks are shown.

FBD Editor Display Options

What's in This Chapter

Display Options 177
Workspace Appearance and Graph Options 178

Display Options

Summary

You can customize the following display options to suit your requirements:

- Zoom
- Links
- Arguments

Zoom Display Options

To access zoom options, select **Device > FB Diagram > View**.

Three options are offered:

- Zoom out to see more of the program at once.
- Zoom in to focus on the program in more detail.
- Zoom to 50%, 75%, 100%, 150 %, 200%, or 400% to have a customized view of the program.

Links Display Options

To access links display options, select **Device > FB Diagram > Tools**.

Three options are offered. You can:

- Renumber links, to aid in understanding the execution of the program.
- Show all links, to see which blocks are linked together.
- Hide all links, to have a better overall view of the blocks.

When you click on a link, its properties window appears and enables you to customize the text that appears next to the link.

Arguments Display Options

The following procedure describes how to access and change argument display options:

1. Position the mouse over a block.

   Result: One or more squares become visible on the block border. It also indicates if the argument is analog or boolean.

2. Click on this square.

   Result: The display options appear.

3. Choose if you want the label to be displayed and what text should appear.

Workspace Appearance and Graph Options

Summary

The FBD editor enables you to customize the workspace by changing its appearance and graph options.

Appearance and Graph Options

To access appearance and graph options, left-click anywhere in the workspace, except on an object.

Appearance Options

The following table lists all the possible appearance customization options:

Appearance option	Description	Possible choices
Background Color	Enables you to set the background color of the workspace by clicking the box where the color is displayed.	Choose between the colors available in the **Custom**, **Web**, and **System** tabs.
Background Type	Enables you to set the background type.	Choose between a flat color, gradient, or image background.
Enable Context Menu	Shows or hides the context menu.	True or False
Enable Tooltip	Shows or hides tooltips.	True or False
Restrict to Canvas	Enables you to choose whether the FBD program should be kept inside the canvas.	True or False
Show Grid	Enables you to choose whether the accurate grid is visible.	True or False **NOTE:** This grid must not be confused with the grid line, which is accessed from the top-level **View** menu bar.

Graph Options

The following table lists all the possible graph customization options:

Graph option	Description	Possible choices
Allow Add Connection	Enables you to choose whether connections can be added to the workspace.	True or False
Allow Add Shape	Enables you to choose whether blocks can be added to the workspace.	True or False
Allow Delete Shape	Enables you to choose whether blocks can be deleted.	True or False
Allow Move Shape	Enables you to choose whether blocks can be moved in the workspace.	True or False
Locked	Enables you to choose whether the FBD program can be edited.	True or False

Display Gridlines

You may wish to display the grid lines. In order to do so, select **Device > FB Diagram > View > Show Gridlines**.

Compiling, Simulating, and Transferring a Program

What's in This Part

Introduction 180
LTMT Main Unit Logic Simulator 181
Initialization and Connection 183
Transferring Logic Files between the LTMT Main Unit and Custom Logic Editor 184

Introduction

Compiling Overview

The customized program must be compiled before being downloaded to the LTMT main unit:

- The programs in custom logic language can be compiled directly.
- The programs in FBD language must be first converted in custom logic language programs before compilation as custom logic programs.

Compiling includes a check for program errors, such as:

- Syntax and structure errors
- Symbols without corresponding addresses
- Resources used by the program that are not available
- Whether the program fits in available LTMT main unit memory

Converting FB Diagram to Custom Logic

Select **Device > FB Diagram > FB Diagram to Custom Logic** to compile the created or edited FBD into custom logic program.

The program is automatically copied into the custom logic editor if there is no detected error.

NOTE: Remember to save the FBD program in the FBD editor before converting it, because it is not possible to convert a custom logic program file into an FBD program file.

Compiling Custom Logic

Follow these steps in order to compile the custom logic program just created into PCode:

1. Select **Device > Custom Logic**.
2. Select **Compile Custom Logic**.

   NOTE: If no errors are detected, the PCode window is displayed. Otherwise, the Detected Error window is displayed.

LTMT Main Unit Logic Simulator

Overview

SoMove software with the TeSys Tera DTM Library comes with the logic simulator. It enables to test the functioning of a customized program in custom logic language before transferring it into the LTMT main unit.

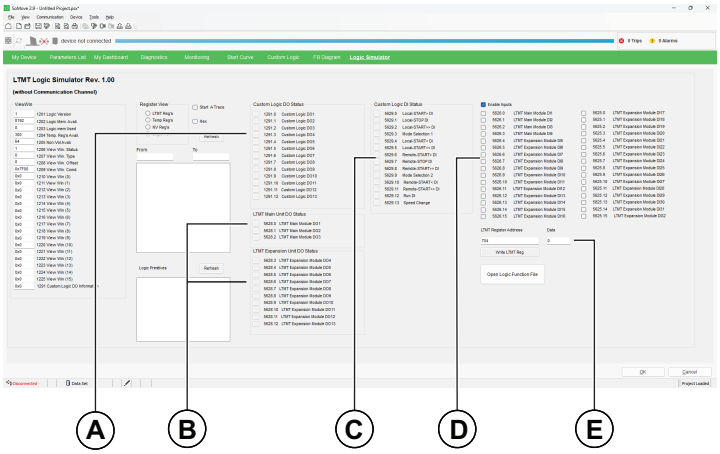
NOTE: To simulate an FBD program, it must be first converted and saved as a custom logic program with the extension **.lf*.

Logic Simulator Interface

To open the logic simulator, select the **Logic Simulator** tab. The logic simulator window is then displayed. In the right bottom corner, select **Open Logic Function File** to import the saved custom logic program.

Select the customized program file and select **Open**.

The logic simulator is now loaded with the customized program. The customized program can be simulated in the simulator as displayed below.



A	Updates the Custom Logic DO status.
B	Updates the LTMT module DO status.
C	Updates the Custom Logic DI status.
D	Used to simulate LTMT module DI data.
E	Used to simulate LTMT register address.

Using the Logic Simulator

To simulate a custom logic file using logic simulator, follow below steps:

- Select **Logic Simulator** Tab
- Select **Open Logic Function File** and select the saved custom logic file from your PC.
- Select the type of register under **Register View** and click **Refresh** to load the register list.
- Modify the register and data under **LTMT Register Address** and **Data** fields respectively as per requirement and click **Write LTMT Reg**.

- Select **Refresh** to update the register list with the latest values provided.

  NOTE: To update the register list with the new values provided, click **Refresh**.

NOTE: When a custom logic file is uploaded to the simulator and a timer is enabled, it will function as expected. However, once a timer has been enabled and executed, it cannot be enabled again. This is a limitation of the Timer and Tenth Timer. To enable the timer again, the custom logic file should be uploaded again. It is recommend that this limitation is considered when designing and testing the custom logic files when working with timers.

Initialization and Connection

Initialization

When you connect the LTMT main unit to the PC, the controller automatically initializes. This initialization process enables the controller and the PC to exchange identification information.

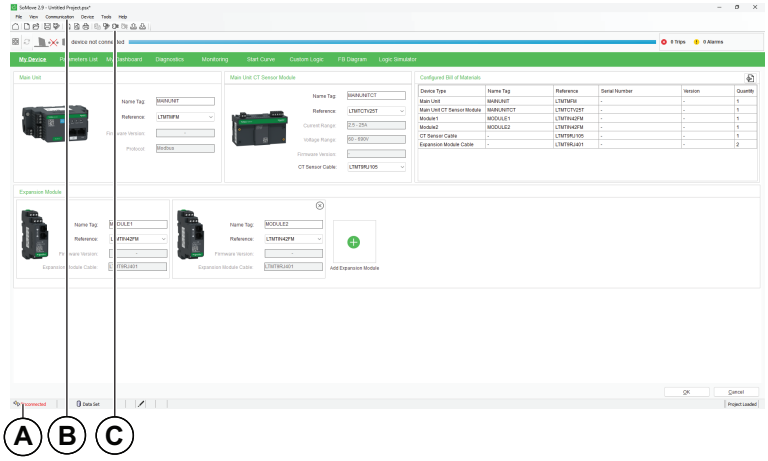
During this process, the custom logic editor indicates wait until initialization is complete.

Connection

After initialization, the LTMT main unit should automatically connect to the PC.

To verify that the controller is connected, check the status bar in the custom logic editor.

If the status bar reads **Disconnected**, then select **Communication > Connect to Device** or select **Connect to Device** icon.



A	Connection Status
B	Communication menu
C	Connect to Device icon

A progress bar briefly appears as your PC connects to the controller, and the word **Connected** appears in the status bar when the connection process successfully completes.

When the LTMT main unit is connected, you can:

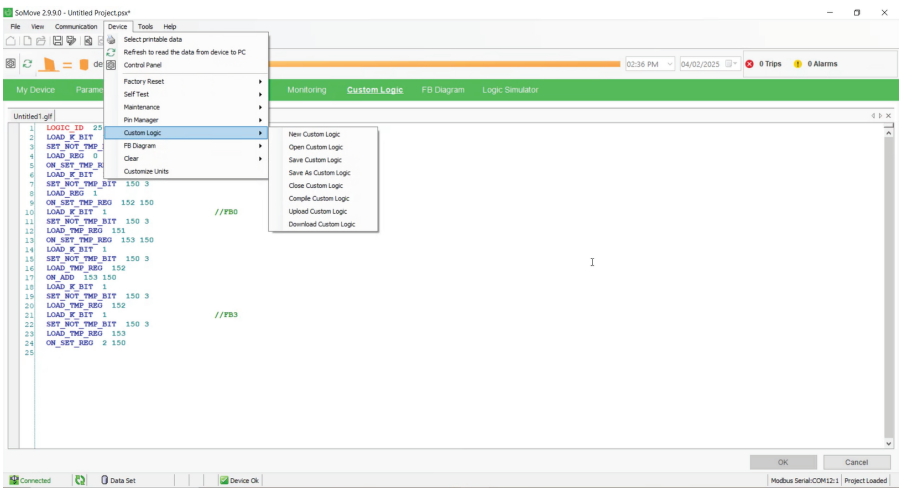
- Upload customized program files from the controller to SoMove software with the TeSys Tera DTM Library for editing.
- Download edited customized program files from SoMove software with the TeSys Tera DTM Library to the controller.

Transferring Logic Files between the LTMT Main Unit and Custom Logic Editor

File Transfer with Device Custom Logic to PC

To transfer the program file from the LTMT main unit to the custom logic editor:

1. Ensure that the LTMT main unit is connected to the PC.
2. Select **Device > Custom Logic > Upload Custom Logic** to transfer the program file from the LTMT main unit to the custom logic editor.



3. When the customized program file has been transferred, you can use the custom logic editor to edit it as a custom logic program.

   NOTE: Programs retrieved from LTMT main unit are in custom logic only without comments. Programs such as FBD programs cannot be retrieved from the LTMT main unit.

4. After your program file edits are complete, save your work to a file.

   Select **Device > Custom Logic > Save Custom Logic** or **Custom LogicSave As Custom Logic**.

File Transfer Procedure with PC Custom Logic to Device

After you have edited and compiled your program file, you can transfer the file to the LTMT main unit. Before SoMove software with the TeSys Tera DTM Library makes this transfer, the following conditions must be met:

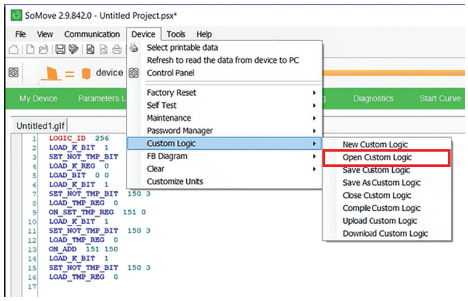
- The program file to transfer must be different than the program file present in the LTMT main unit, that is, the software does not transfer the same program.
- Current must not be detected, that is, online current must be less than 10% of FLC.

If these conditions are not met, the file cannot be transferred to the controller.

To transfer a program file from the custom logic editor to the LTMT main unit:

1. Ensure that the LTMT main unit is connected to the PC.

2. Ensure that the file to be transferred is in the main window. To open a file, select **Open Custom Logic** from the **Custom Logic** submenu of the **Device** menu. Then, browse to the correct location and select **Open**.



3. Select **Device > Custom Logic > Compile Custom Logic** to compile the custom logic program.
4. After the compilation of the program, select **Device > Custom Logic > Download Custom Logic** to download the program file from the custom logic editor to the LTMT main unit. The transfer is now successful.
5. A new dialog opens, select **Ok** to close it.

Schneider Electric
35 rue Joseph Monier
92500 Rueil Malmaison
France

+ 33 (0) 1 41 29 70 00

www.se.com

DOCA0275EN-00