Modicon M262 Logic/Motion Controller

Programming Guide

Original instructions

EIO0000003651.14
04/2026

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.

This document and its content are protected under applicable copyright laws and provided for informative use only. No part of this document may be reproduced or transmitted in any form or by any means (electronic, mechanical, photocopying, recording, or otherwise), for any purpose, without the prior written permission of Schneider Electric.

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 7
About the Document 8
About the Modicon M262 Logic/Motion Controller 14
M262 Logic/Motion Controller Description 14
Modicon M262 Logic Controller 18
Modicon M262 Logic Controller 18
Modicon M262 Motion Controller 20
Modicon M262 Motion Controller 20
Product Related Cybersecurity Information 24
Security Settings Configuration with Cybersecurity Admin Expert Software 24
How to Configure the Controller 34
Configuring the Controller 34
Libraries 36
Libraries 36
Supported Standard Data Types 37
Supported Standard Data Types 37
Memory Mapping 38
Controller Memory Organization 38
Non-Volatile Memory Organization 40
RAM Memory Organization 43
NVRAM Memory Organization 44
Relocation Table 45
Tasks 48
Maximum Number of Tasks 48
Task Types 49
Task Configuration Screen 51
System and Task Watchdogs 53
Task Priorities 54
Default Task Configuration 55
Controller States and Behaviors 56
Controller State Diagram 56
Controller States Description 60
State Transitions and System Events 63
Controller States and Output Behavior 63
Commanding State Transitions 66
Error Detection, Types and Management 72
Remanent Variables 73
Controller Device Editor 75
Controller Parameters 75
Communication Settings 77
PLC Settings 78
Services 79
Ethernet Services 80
Users Rights 86
Predefined **Access Rights** for **Group Persona** 92
Predefined **Access Rights** for **Group Function** 94

Predefined Access Rights Needed by Object and Associated Operations 95
Symbol Configuration Editor 97
Embedded Inputs and Outputs Configuration 102
Configuring the Fast I/Os 102
Embedded I/Os Configuration 102
Hardware Encoder Interface 106
Hardware Encoder Interface 106
Adding an Encoder 107
Encoder Motion Functions 109
Expansion Modules Configuration 111
TM3 I/O Configuration General Description 111
TM3 I/O Bus Configuration 116
TMS Expansion Module Configuration 117
TM3 Expansion Module Configuration 118
Optional I/O Expansion Modules 118
Ethernet Configuration 121
Ethernet Features, Functions and Services 121
Presentation 121
IP Address Configuration 123
Modbus TCP Client/Server 128
FTP Server 129
SNMP 131
Web Server 132
Monitoring Menu 136
Diagnostics Menu 138
Maintenance Menu 144
Machine Assistant Menu 153
Firewall Configuration 153
Introduction 153
Dynamic Changes Procedure 155
Firewall Behavior 156
Firewall Script Commands 157
Used Ports 162
CoDeSys Protocol 162
Industrial Ethernet 164
Industrial Ethernet Presentation 164
DHCP Server 168
Fast Device Replacement 168
Controller as a Target Device on EtherNet/IP 168
Controller as a Slave Device on Modbus TCP 187
Changing the Modbus TCP Port 191
Sercos Configuration 193
Overview of the Sercos Standard 193
Modicon M262 Logic/Motion Controller Sercos Configuration 194
Modicon M262 Motion Controller and Safety Controllers with Sercos 194
Single Wire Architecture 194
Serial Line Configuration 197
Serial Line Configuration 197
Network Manager 198

Modbus Manager 199
ASCII Manager 202
Modbus Serial IOScanner 204
Adding a Device on the Modbus Serial IOScanner 205
ControlChannel: Enables or Disables a Communication Channel 211
Adding a Modem to a Manager 212
SysLog Agent 213
OPC UA 215
OPC UA Overview 215
OPC UA Server Configuration 215
OPC UA Server Overview 215
OPC UA Server Configuration 216
OPC UA Server Symbols Configuration 222
OPC UA Server Performance 224
OPC UA Client Configuration 226
OPC UA Client Overview 226
Programming the OPC UA Client 227
Post Configuration 229
Post Configuration Presentation 229
Post Configuration File Management 230
Post Configuration Example 232
Connecting a Modicon M262 Logic/Motion Controller to a PC 236
Connecting the Controller to a PC 236
Updating Firmware 238
Updating the Controller Firmware by SD Card 238
Updating the Controller Firmware by Controller Assistant 240
Updating TM3 Expansion Modules Firmware 242
Updating TMSES4 Expansion Module Firmware 244
Managing Script Files 246
Creating a Script 246
Generating Scripts and Files 249
Transferring Scripts and Files 250
Cloning a Controller 252
Before Cloning a Controller 252
Cloning a Controller 254
Compatibility 256
Software and Firmware Compatibilities 256
Diagnostic 257
System Diagnostic 257
Diagnostic Messages 257
Machine Assistant 276
Accessing the Web Server Through Industrial Plug and Work 276
Launching the Web Server 276
Using the Machine Assistant 276
Launching the Machine Assistant 276
Managing the Network Scan 277
Managing the Devices Network Settings 278
Backing Up/Restoring Configuration 279
Exporting/Importing .semdt Files 280
Appendices 281

How to Change the IP Address of the Controller 282
changeIPAddress: Change the IP Address of the Controller 282
Functions to Get/Set Serial Line Configuration in User Program 284
GetSerialConf: Get the Serial Line Configuration 284
SetSerialConf: Change the Serial Line Configuration 285
LinkNumber: Communication Port Number 286
SERIAL_CONF: Structure of the Serial Line Configuration Data Type 287
Controller Performance 288
Processing Performance 288
M262 Logic/Motion Controller Event Messages 290
SysLog Messages from M262 Logic/Motion Controller 290
Glossary 293
Index 304

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

The purpose of this document is to help you program and operate your Modicon M262 Logic/Motion Controller with the EcoStruxure Machine Expert software.

NOTE: Read and understand this document and all related documents before installing, operating, or maintaining your Modicon M262 Logic/Motion Controller.

Validity Note

This document has been updated for the release of EcoStruxure™ Machine Expert V2.6.

Product Related Information

⚠ WARNING

LOSS OF CONTROL

- Perform a Failure Mode and Effects Analysis (FMEA), or equivalent risk analysis, of your application, and apply preventive and detective controls before implementation.
- Provide a fallback state for undesired control events or sequences.
- Provide separate or redundant control paths wherever required.
- Supply appropriate parameters, particularly for limits.
- Review the implications of transmission delays and take actions to mitigate them.
- Review the implications of communication link interruptions and take actions to mitigate them.
- Provide independent paths for control functions (for example, emergency stop, over-limit conditions, and error conditions) according to your risk assessment, and applicable codes and regulations.
- Apply local accident prevention and safety regulations and guidelines.[1]
- Test each implementation of a system for proper operation before placing it into service.

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

[1] For additional information, refer to NEMA ICS 1.1 (latest edition), *Safety Guidelines for the Application, Installation, and Maintenance of Solid State Control* and to NEMA ICS 7.1 (latest edition), *Safety Standards for Construction and Guide for Selection, Installation and Operation of Adjustable-Speed Drive Systems* or their equivalent governing your particular location.

⚠ WARNING

UNINTENDED EQUIPMENT OPERATION

- Only use software approved by Schneider Electric for use with this equipment.
- Update your application program every time you change the physical hardware configuration.

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

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 (EIO0000003651)
- French (EIO0000003652)
- German (EIO0000003653)
- Spanish (EIO0000003654)
- Italian (EIO0000003655)
- Chinese (EIO0000003656)

Related Documents

Title of Documentation	Reference Number
EcoStruxure Machine Expert - Programming Guide	EIO0000002854 (ENG) EIO0000002855 (FRE) EIO0000002856 (GER) EIO0000002857 (SPA) EIO0000002858 (ITA) EIO0000002859 (CHS)
Modicon M262 Logic/Motion Controller - Hardware Guide	EIO0000003659 (ENG) EIO0000003660 (FRE) EIO0000003661 (GER) EIO0000003662 (SPA) EIO0000003663 (ITA) EIO0000003664 (CHS)
EcoStruxure Machine Expert - Industrial Ethernet Overview - User Guide	EIO0000003053 (ENG) EIO0000003054 (FRE) EIO0000003055 (GER) EIO0000003056 (SPA) EIO0000003057 (ITA) EIO0000003058 (CHS)
Modicon TM3 Expansion Modules - Programming Guide	EIO0000003119 (ENG) EIO0000003120 (FRE) EIO0000003121 (GER) EIO0000003122 (SPA) EIO0000003123 (ITA) EIO0000003124 (CHS)
Modicon TM3 Expert I/O Modules - HSC Library Guide	EIO0000003683 (ENG) EIO0000003684 (FRE) EIO0000003685 (GER) EIO0000003686 (SPA) EIO0000003687 (ITA) EIO0000003688 (CHS)
Modicon TM5 EtherNet/IP Fieldbus Interface - Programming Guide	EIO0000003707 (ENG) EIO0000003708(FRE) EIO0000003709 (GER) EIO0000003710 (SPA) EIO0000003711 (ITA) EIO0000003712 (CHS)
Modicon TMS Expansion Modules - Programming Guide	EIO0000003691 (ENG) EIO0000003692 (FRE) EIO0000003693 (GER) EIO0000003694 (SPA) EIO0000003695 (ITA) EIO0000003696 (CHS)

Title of Documentation	Reference Number
Modicon M262 Logic/Motion Controller - System Functions and Variables - System Library Guide	EIO0000003667 (ENG) EIO0000003668 (FRE) EIO0000003669 (GER) EIO0000003670 (SPA) EIO0000003671 (ITA) EIO0000003672 (CHS)
Modicon M262 Logic/Motion Controller - Encoder Library Guide	EIO0000003675 (ENG) EIO0000003676(FRE) EIO0000003677(GER) EIO0000003678 (SPA) EIO0000003679 (ITA) EIO0000003680 (CHS)
EcoStruxure Machine Expert - FtpRemoteFileHandling Library Guide	EIO0000002779 (ENG) EIO0000002780 (FRE) EIO0000002781 (GER) EIO0000002782 (SPA) EIO0000002783 (ITA) EIO0000002784 (CHS)
EcoStruxure Machine Expert - SnmpManager Library Guide	EIO0000002797 (ENG) EIO0000002798 (FRE) EIO0000002799 (GER) EIO0000002800 (SPA) EIO0000002801 (ITA) EIO0000002802 (CHS)
EcoStruxure Machine Expert - OpcUaHandling Library Guide	EIO0000004021 (ENG) EIO0000004022 (FRE) EIO0000004023 (GER) EIO0000004025 (SPA) EIO0000004024 (ITA) EIO0000004026 (CHS)
EcoStruxure Machine Expert - SysLog Library Guide	EIO0000004614 (ENG) EIO0000004615 (FRE) EIO0000004616 (GER) EIO0000004617 (ITA) EIO0000004618 (SPA) EIO0000004619 (CHS)
EcoStruxure Machine Expert - Modem Functions - Modem Library Guide	EIO0000000552 (ENG) EIO0000000491 (FRE) EIO0000000492 (GER) EIO0000000493 (SPA) EIO0000000494 (ITA) EIO0000000495 (CHS)

Title of Documentation	Reference Number
Cybersecurity Guidelines for EcoStruxure Machine Expert, Modicon and PacDrive Controllers and Associated Equipment, User Guide	EIO0000004242 (ENG)
Cybersecurity Admin Expert, User Manual	CAE_User_Guide (ENG)

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.

Terminology Derived from Standards

The technical terms, terminology, symbols and the corresponding descriptions in the information contained herein, or that appear in or on the products themselves, are generally derived from the terms or definitions of international standards.

In the area of functional safety systems, drives and general automation, this may include, but is not limited to, terms such as *safety*, *safety function*, *safe state*, *fault*, *fault reset*, *malfunction*, *failure*, *error*, *error message*, *dangerous*, etc.

Among others, these standards include:

Standard	Description
IEC 61131-2:2007	Programmable controllers, part 2: Equipment requirements and tests.
ISO 13849-1:2023	Safety of machinery: Safety related parts of control systems. General principles for design.
EN 61496-1:2020	Safety of machinery: Electro-sensitive protective equipment. Part 1: General requirements and tests.
ISO 12100:2010	Safety of machinery - General principles for design - Risk assessment and risk reduction
EN 60204-1:2006	Safety of machinery - Electrical equipment of machines - Part 1: General requirements
ISO 14119:2013	Safety of machinery - Interlocking devices associated with guards - Principles for design and selection
ISO 13850:2015	Safety of machinery - Emergency stop - Principles for design
IEC 62061:2021	Safety of machinery - Functional safety of safety-related electrical, electronic, and electronic programmable control systems
IEC 61508-1:2010	Functional safety of electrical/electronic/programmable electronic safety-related systems: General requirements.
IEC 61508-2:2010	Functional safety of electrical/electronic/programmable electronic safety-related systems: Requirements for electrical/electronic/programmable electronic safety-related systems.
IEC 61508-3:2010	Functional safety of electrical/electronic/programmable electronic safety-related systems: Software requirements.
IEC 61784-3:2021	Industrial communication networks - Profiles - Part 3: Functional safety fieldbuses - General rules and profile definitions.
2006/42/EC	Machinery Directive

Standard	Description
2014/30/EU	Electromagnetic Compatibility Directive
2014/35/EU	Low Voltage Directive

In addition, terms used in the present document may tangentially be used as they are derived from other standards such as:

Standard	Description
IEC 60034 series	Rotating electrical machines
IEC 61800 series	Adjustable speed electrical power drive systems
IEC 61158 series	Digital data communications for measurement and control – Fieldbus for use in industrial control systems

Finally, the term *zone of operation* may be used in conjunction with the description of specific hazards, and is defined as it is for a *hazard zone* or *danger zone* in the *Machinery Directive* (*2006/42/EC*) and *ISO 12100:2010*.

NOTE: The aforementioned standards may or may not apply to the specific products cited in the present documentation. For more information concerning the individual standards applicable to the products described herein, see the characteristics tables for those product references.

About the Modicon M262 Logic/Motion Controller

Introduction

This chapter provides information about the Modicon M262 Logic/Motion Controller and devices that EcoStruxure Machine Expert can configure and program.

M262 Logic/Motion Controller Description

Overview

The M262 Logic/Motion Controller has various powerful features and can service a wide range of applications.

Programming Languages

The M262 Logic/Motion Controller is configured and programmed with the EcoStruxure Machine Expert software which supports the following IEC 61131-3 programming languages:

- IL: Instruction List
- ST: Structured Text
- FBD: Function Block Diagram
- SFC: Sequential Function Chart
- LD: Ladder Diagram

The EcoStruxure Machine Expert software can also be used to program the M262 Logic/Motion Controller using CFC (Continuous Function Chart) language.

Power Supply

The power supply of the M262 Logic/Motion Controller is 24 Vdc (see Modicon M262 Logic/Motion Controller, Hardware Guide).

Real Time Clock

The M262 Logic/Motion Controller includes a Real Time Clock (RTC) system (see Modicon M262 Logic/Motion Controller, Hardware Guide).

The system time is maintained by capacitors when the power is off. The time is maintained for 1 000 hours when the controller is not supplied.

Run/Stop

The M262 Logic/Motion Controller can be operated by the following:

- A hardware Run/Stop switch (see Modicon M262 Logic/Motion Controller, Hardware Guide)
- A Run/Stop operation by a dedicated digital input, defined in the software configuration. For more information, refer to Configuration of Digital Inputs, page 102.
- A software command
- The system variable PLC_W in a Relocation Table, page 45
- The Web Server Maintenance: Run/Stop Controller Submenu, page 148

Memory

This table describes the different types of memory:

Memory Type	Size	Use
RAM	256 Mbytes, of which 32 Mbytes are available for the application	For the execution of the application and the firmware.
Flash	1 Gbyte	Non-volatile memory dedicated to the retention of the program and data in case of a power interruption.
Non-volatile RAM	512 kbytes	Non-volatile memory dedicated to the retention of the retain-persistent variables, and the diagnostic files and associated information.

Embedded Inputs/Outputs

The following embedded I/O types are available:

- Fast inputs
- Fast source outputs

Encoder for M262 Motion Controller

The following encoder modes are available:

- Incremental mode
- SSI mode

Removable Storage

The M262 Logic/Motion Controller includes an integrated SD card slot (see Modicon M262 Logic/Motion Controller, Hardware Guide).

The main uses of the SD card are:

- Initializing the controller with a new application
- Updating the controller and expansion module firmware, page 238
- Applying post configuration files to the controller, page 229
- Storing recipes files
- Receiving data logging files

Embedded Communication Features

The following types of communication ports are available:

- Serial Line (see Modicon M262 Logic/Motion Controller, Hardware Guide)
- USB Mini-B (see Modicon M262 Logic/Motion Controller, Hardware Guide)
- Ethernet (see Modicon M262 Logic/Motion Controller, Hardware Guide)
- Sercos (Ethernet 1) (see Modicon M262 Logic/Motion Controller, Hardware Guide)

Expansion Module and Bus Coupler Compatibility

Refer to the compatibility tables in EcoStruxure Machine Expert, Compatibility and Migration User Guide.

Compatibility with Modicon Edge I/O NTS Devices

Refer to the compatibility tables in EcoStruxure Machine Expert, Compatibility and Migration User Guide.

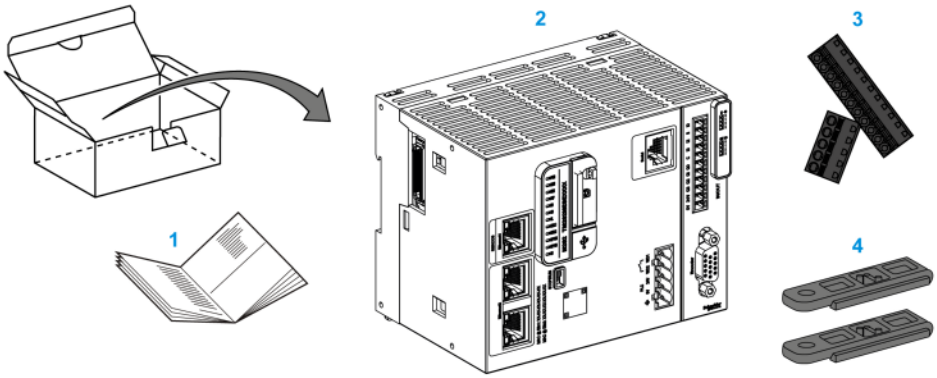
NOTE: To use Modicon Edge I/O NTS devices, you must activate the OPC UA server, page 215.

M262 Logic/Motion Controller References

Reference	Digital I/O	Power supply	Communication Ports	Terminal Type	Encoder
M262 Logic Controller: TM262L•	4 fast inputs 4 fast source outputs	24 Vdc	1 serial line port 1 USB programming port 1 Ethernet port 1 dual port Ethernet switch	Removable spring	–
M262 Motion Controller: TM262M•	4 fast inputs 4 fast source outputs	24 Vdc	1 serial line port 1 USB programming port 1 Ethernet port for fieldbus with Sercos interface 1 dual port Ethernet switch	Removable spring	1 Encoder port
NOTE: You can use the fast inputs/outputs as regular inputs/outputs.					

Delivery Content

The following figure presents the content of the delivery for the M262 Logic/Motion Controller:



1 M262 Logic/Motion Controller Instruction Sheet

2 M262 Logic/Motion Controller

3 Removable spring terminal blocks

4 Attachment parts

Modicon M262 Logic Controller

Modicon M262 Logic Controller

Controller Overview

TM262L• Controller Reference	Digital I/O	Power supply	Communication Ports	Terminal Type	Performance Duration for 1000 instructions	Motion Capabilities
TM262L01MESE8T	4 fast inputs 4 fast source outputs	24 Vdc	1 serial line port 1 USB programming port 1 Ethernet port 1 dual port Ethernet switch	Removable spring	5 µs	Independent Axis on EIP / CANopen
TM262L10MESE8T	4 fast inputs 4 fast source outputs	24 Vdc	1 serial line port 1 USB programming port 1 Ethernet port 1 dual port Ethernet switch	Removable spring	5 µs	Independent Axis on EIP / CANopen
TM262L20MESE8T	4 fast inputs 4 fast source outputs	24 Vdc	1 serial line port 1 USB programming port 1 Ethernet port 1 dual port Ethernet switch	Removable spring	3 µs	Independent Axis on EIP / CANopen

Supported Features

- Cybersecurity access rights, encrypted communication: see Users Rights, page 86
- Web configuration Web server and WebVisualisation: see Web Server, page 132
- MQTT Protocol (signed / encrypted)
- OPC UA Server / Client, (signed / encrypted): see OPC UA Server Overview, page 215 and OPC UA Client Overview, page 226
- 1 IO Scanner: see Modbus Serial IOScanner, page 204
- Supported Services:
  - HTTP (API)
  - DHCP (Client / Server): see DHCP Server, page 168
  - DNS Client
  - POP3 Client
  - RSTP (Eth2 port)
  - SMTP (Client / Agent)
  - SNMP: see SNMP, page 131
  - FTP (Client / Server): see FTP Server, page 129
  - EtherNet IP (Adapter / Scanner): see Controller as a Target Device on EtherNet/IP, page 168
  - Modbus / TCP (Client / Server / NVL): see Modbus TCP Client/Server, page 128
  - Modbus / ASCII – RTU (Master / Slave / IO scanner / modem): see Modbus Manager, page 199
  - CANopen (Master)

Modicon M262 Motion Controller

Modicon M262 Motion Controller

Controller Overview

TM262M• Controller Reference	Digital I/O	Power supply	Communication Ports	Terminal Type	Encoder port	Performance Duration for 1000 instructions	Motion Capabilities
TM262M05MESS8T	4 fast inputs 4 fast source outputs	24 Vdc	1 serial line port 1 USB programming port 1 dual port Ethernet switch 1 Ethernet port for fieldbus with Sercos interface	Removable spring	1 Encoder port	5 µs	Independent Axis on EIP / CANopen Synchronous axis on Sercos (4 axes maximum)
TM262M15MESS8T	4 fast inputs 4 fast source outputs	24 Vdc	1 serial line port 1 USB programming port 1 dual port Ethernet switch 1 Ethernet port for fieldbus with Sercos interface	Removable spring	1 Encoder port	5 µs	Independent Axis on EIP / CANopen Synchronous axis on Sercos (4 axes maximum)
TM262M25MESS8T	4 fast inputs 4 fast source outputs	24 Vdc	1 serial line port 1 USB programming port 1 dual port Ethernet switch 1 Ethernet port for fieldbus with Sercos interface	Removable spring	1 Encoder port	3 µs	Independent Axis on EIP / CANopen Synchronous axis on Sercos (8 axes maximum)
TM262M35MESS8T	4 fast inputs 4 fast source outputs	24 Vdc	1 serial line port 1 USB programming port 1 dual port Ethernet switch 1 Ethernet port for fieldbus with Sercos interface	Removable spring	1 Encoder port	3 µs	Independent Axis on EIP / CANopen Synchronous axis on Sercos (24 axes maximum)

Supported Features

- Cybersecurity access rights, encrypted communication: see Users Rights, page 86
- Web configuration Web server and WebVisualisation: see Web Server, page 132
- MQTT Protocol (signed / encrypted) for TM262M15MESS8T, TM262M25MESS8T and TM262M35MESS8T
- OPC UA Server / Client, (signed / encrypted): see OPC UA Server Overview, page 215 and OPC UA Client Overview, page 226
- 1 IO Scanner: see Modbus Serial IOScanner, page 204
- Supported Services:
  - HTTP (API)
  - DHCP (Client / Server): see DHCP Server, page 168
  - DNS Client
  - POP3 Client
  - RSTP (Eth2 port)
  - SMTP (Client / Agent)
  - SNMP: see SNMP, page 131
  - FTP (Client / Server): see FTP Server, page 129
  - EtherNet IP (Adapter / Scanner): see Controller as a Target Device on EtherNet/IP, page 168
  - Modbus / TCP (Client / Server / NVL): see Modbus TCP Client/Server, page 128
  - Modbus / ASCII – RTU (Master / Slave / IO scanner / modem): see Modbus Manager, page 199
  - CANopen (Master)
  - Sercos (Master): see Sercos Configuration, page 194

Performance Overview

The Modicon TM262M• Motion Controller supports the features available in TM262L• Logic Controller, plus it integrates Motion functionalities.

The TM262M• range of Motion controllers is, without additional devices, ready for motion with the integrated Sercos motion bus. It merges the hard-real-time aspects of the Sercos interface with Ethernet. It is based upon and conforms to the Ethernet standard IEEE 802.3 and ISO/IEC 8802-3 to support the real-time application with high performance. Other features supporting motion functionalities include:

- Synchronous axis Sercos devices, managed by PLCopen libraries, are fully synchronous with the internal Motion task and the Sercos Cycle time, as for example: LXM32S.
- Non-axis Sercos devices are also synchronized with the internal Motion task, for example, TM5NS01 islands or safety-related TM5CSLC100/ TM5CSLC200 controllers.
- External port for Incremental or SSI encoder. The encoder support is synchronized with the Motion application. It can be used like a real axis or a virtual axis.
- Fast inputs. The fast inputs support a touch probe function to capture position. The captured position can be used in the Motion application.
- Motion Kernel is embedded in the TM262M• Motion controller, allowing you to manage the motion functions:
  - Synchronous axis in coordinated movement in which the function blocks are based on the PLCopen Standard to control the position / speed of a single axis.
  - Gearing mode (Master / Slave Function Block).
  - Caming mode, based on recipes, with modifications on the fly. The recipe can be designed with the cam editor included in EcoStruxure Machine Expert.
  - G-code, based on recipes. The recipe can be designed with the CNC editor included in EcoStruxure Machine Expert.

Depending on the Motion controller and the Sercos cycle time, you can configure more or less synchronous axis and non-axis Sercos devices.

A TM5 System island used on Sercos is managed as a non-axis Sercos device. Although there are generally no restriction on the number of I/O in the Sercos configuration, the number of I/O configured increases the load of the Sercos bus and may lead to an overflow. If an overflow occurs, try increasing the Sercos cycle time. If increasing the Sercos cycle time is not compatible with your application, then optimize the application.

The following table indicates the performance capabilities of the Motion application:

Controller Reference	Sercos Cycle Time	Synchronized Axes on Sercos (activated and simulated)	Additional Virtual Axes FB_ControlledAxis	Additional Sercos Devices
TM262M05MESS8T	1 ms	4	1	2
	2 ms	4	1	6
	4 ms	4	1	8
TM262M15MESS8T	1 ms	4	1	4
	2 ms	4	1	12
	4 ms	4	1	12
TM262M25MESS8T	1 ms	4	1	8
	2 ms	8	2	8
	4 ms	8	2	16
TM262M35MESS8T	1 ms	8	2	8
	2 ms	16	4	8
	4 ms	24	16	40

The Motion Sizer is embedded in EcoStruxure Machine Expert to help you define your kinematic architecture. For more information on these features, refer to OneMotionSizer Online Help (see Motion Sizer, Online Help).

Product Related Cybersecurity Information

Introduction

To help keep your Schneider Electric products secure and protected, implement the cybersecurity best practices as indicated in the Cybersecurity Best Practices and Cybersecurity Guidelines for EcoStruxure Machine Expert, Modicon and PacDrive Controllers and Associated Equipment document provided on the Schneider Electric website.

Certificate Management

By default, the following certificates are displayed in the controller Web server in Maintenance: Certificates Submenu, page 149:

- TM262-XX-OPCUA is used for OPC UA.
- TM262-XX is used for HTTP/FTP/WebVisualisation.
- TM262-XX @ AABBCCDDEEFF (nodename) is used for communication with EcoStruxure Machine Expert.

The TM262-XX-OPCUA and TM262-XX certificates can only be managed through the Web server under the Maintenance: Certificates Submenu, page 149. To reset or delete these certificates, the supported methods are:

- Executing a `Format` command using the SD card
- Executing a Reset Origin, page 68 command on the controller

  NOTE: Deleting the TM262-XX-OPCUA or TM262-XX certificates through the Security Screen does not reset, nor regenerate the certificates.

Boot Applications Encryption and Signing

The Modicon M262 Logic/Motion Controller supports boot applications encryption and signing with certificates. For more details, refer to Properties... in the EcoStruxure Machine Expert Menu Commands Online Help.

The Modicon M262 Logic/Motion Controller does not support simple encryption, nor encryption with license management.

Security Settings Configuration with Cybersecurity Admin Expert Software

Introduction

CAE (Cybersecurity Admin Expert) is a software-based tool for building and managing the security configuration and policy of the Operational Technology (OT) within the communication network of control systems. It allows centralized administration of user accounts, roles and permissions for devices such as: network devices (switches, firewalls), PCs and IED/Protection relays. CAE is used for multiple purposes:

- Creating a cybersecurity and security policy
- Configuring the security of devices
- Managing the system definition
- Retrieving security logs of a whole substation, plant or industrial environment

CAE is a Schneider Electric software you can download from https://www.se.com/ww/en/all-products.

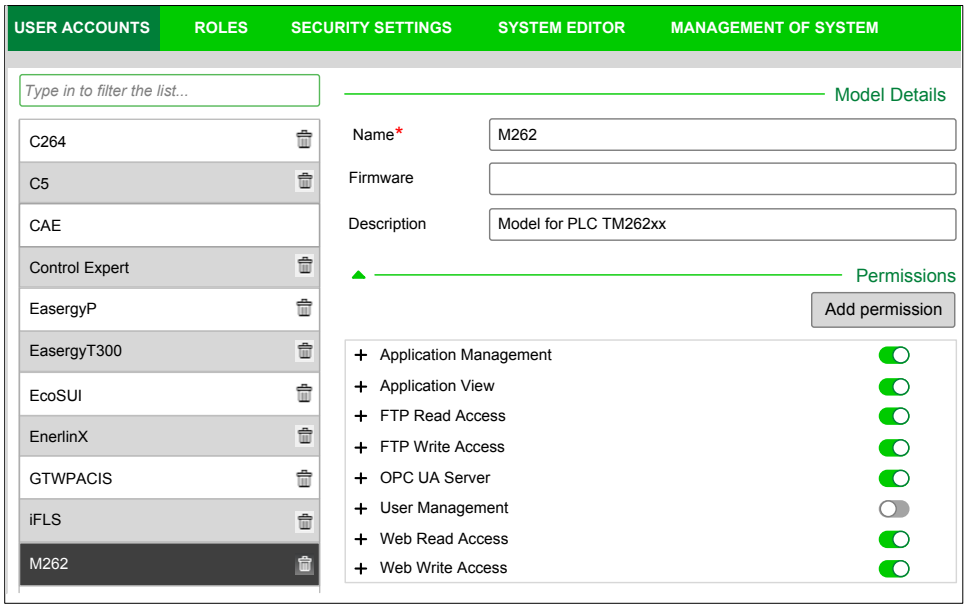
Prior to apply any modification to the CAE settings, refer to the Cybersecurity Admin Expert User Manual.

The M262 Logic/Motion Controller device model encompasses two features:

- Role-Based Access Control (RBAC)
- Device Specific Settings (DSS)

Role-Based Access Control (RBAC)

This feature consists in controlling access to a system's resources based on the roles and permissions of the users. The list of permission covers common use cases, as displayed in the following graphic:



The following table describes each permission, the concerned M262 objects and the corresponding access rights:

CAE Permissions	M262 Object Name	M262 Access Rights
Application Management	Device	USERDB_RIGHT_ALL
	Device.PlcLogic	USERDB_RIGHT_ALL
	Device.PlcLogic.Application	USERDB_RIGHT_ALL
	Device.Settings	USERDB_RIGHT_ALL
	Device.ExternalCmd	USERDB_RIGHT_ALL
	"/"	USERDB_RIGHT_ALL
Application View	Device	USERDB_RIGHT_VIEW
	Device.PlcLogic	USERDB_RIGHT_VIEW
	Device.PlcLogic.Application	USERDB_RIGHT_VIEW
	Device.Settings	USERDB_RIGHT_VIEW
	Device.ExternalCmd	USERDB_RIGHT_VIEW
	"/"	USERDB_RIGHT_VIEW
FTP Read Access	Device.FTP	USERDB_RIGHT_VIEW
FTP Write Access	Device.FTP	USERDB_RIGHT_ALL
OPC UA Server	Device.OPC	USERDB_RIGHT_ALL
User Management	Device.UserManagement	USERDB_RIGHT_ALL
Web Read Access	Device.WEB	USERDB_RIGHT_VIEW
Web Write Access	Device.WEB	USERDB_RIGHT_ALL

Roles and Rights

Up to 20 users are supported by the M262 Logic/Motion Controller. Each user may have several roles. The following table describes the default rights for each user role:

Role	Access Rights
ENGINEER	Application Management Application View FTP Read Access FTP Write Access OPC UA Server Web Read Access Web Write Access
INSTALLER	OPC UA Server Web Read Access
OPERATOR	Application View FTP Read Access FTP Write Access OPC UA Server Web Read Access
SECADM	User Management
VIEWER	Application View FTP Read Access Web Read Access

Device Specific Settings (DSS)

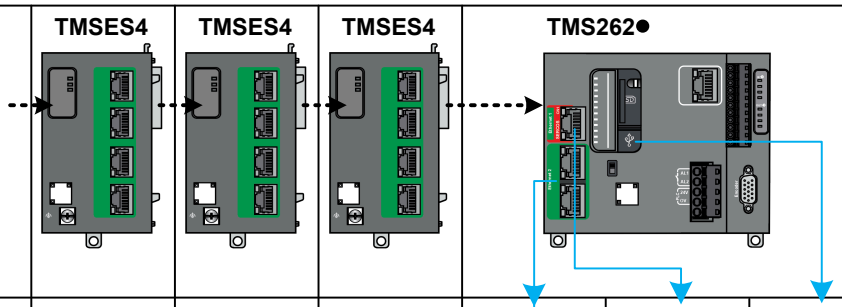
This parameter allows you to configure the device specific settings. This table describes the M262 Logic/Motion Controller **MODELS > Specific Settings** parameters:

Key	Type	Default Value	Description
Discovery Protocols	INTEGER	Decimal: 255 Binary: 1111 1111	Allows you to enable or disable the protocol DPWS and NetManage in each communication port: • TCP port: 5357 • UDP ports: 3702, 5353, 27126, 27127 Bit value: 0 = Disable, 1 = Enable • Bit 0: USB • Bit 1: ETH1 • Bit 2: ETH2 • Bit 3 to 5: TMSES4 1 to 3 • Bit 6 to 7: reserved For further information, refer to the example below (1) . **NOTE:** Disabling these protocols prevents the device to be discovered by the software CAE.
EtherNet/IP	INTEGER	Decimal: 255 Binary: 1111 1111	Allows you to enable or disable EtherNet/IP in each communication port: • TCP port: 44818 • UDP ports: 2222, 44818 Bit value: 0 = Disable, 1 = Enable • Bit 0: USB • Bit 1: ETH1 • Bit 2: ETH2 • Bit 3 to 5: TMSES4 1 to 3 • Bit 6 to 7: reserved For further information, refer to the example below (1) .
FTP Server	INTEGER	Decimal: 255 Binary: 1111 1111	Allows you to enable or disable the protocol in each communication port: • TCP ports: 20, 21 Bit value: 0 = Disable, 1 = Enable • Bit 0: USB • Bit 1: ETH1 • Bit 2: ETH2 • Bit 3 to 5: TMSES4 1 to 3 • Bit 6 to 7: reserved For further information, refer to the example below (1) .
CoDeSys Protocol	INTEGER	Decimal: 255 Binary: 1111 1111	Allows you to enable or disable the protocol in each communication port: • UDP ports: 1740 to 1743 Bit value: 0 = Disable, 1 = Enable • Bit 0: USB • Bit 1: ETH1 • Bit 2: ETH2 • Bit 3 to 5: TMSES4 1 to 3 • Bit 6 to 7: reserved For further information, refer to the example below (1) .

Key	Type	Default Value	Description
Modbus Server	INTEGER	Decimal: 255 Binary: 1111 1111	Allows you to enable or disable the protocol in each communication port: • TCP port: 502 Bit value: 0 = Disable, 1 = Enable • Bit 0: USB • Bit 1: ETH1 • Bit 2: ETH2 • Bit 3 to 5: TMSES4 1 to 3 • Bit 6 to 7: reserved For further information, refer to the example below (1) .
OPC UA Server	INTEGER	Decimal: 255 Binary: 1111 1111	Allows you to enable or disable the protocol in each communication port: • TCP port: 4840 Bit value: 0 = Disable, 1 = Enable • Bit 0: USB • Bit 1: ETH1 • Bit 2: ETH2 • Bit 3 to 5: TMSES4 1 to 3 • Bit 6 to 7: reserved For further information, refer to the example below (1) .
Remote Connection (Fast TCP)	INTEGER	Decimal: 255 Binary: 1111 1111	Allows you to enable or disable the protocol in each communication port: • TCP port: 11740 Bit value: 0 = Disable, 1 = Enable • Bit 0: USB • Bit 1: ETH1 • Bit 2: ETH2 • Bit 3 to 5: TMSES4 1 to 3 • Bit 6 to 7: reserved For further information, refer to the example below (1) .
Clone Application Enabled	BOOL	TRUE	Enable or disable the cloning of the device via the SD card.
SD Card Script Execution Enable	BOOL	TRUE	Enable or disable the execution of scripts via the SD card. Refer to Operation Functions, page 89.
Secure Web Server (HTTPS)	INTEGER	Decimal: 255 Binary: 1111 1111	Allows you to enable or disable the protocol in each communication port: • TCP ports: 80, 443 Bit value: 0 = Disable, 1 = Enable • Bit 0: USB • Bit 1: ETH1 • Bit 2: ETH2 • Bit 3 to 5: TMSES4 1 to 3 • Bit 6 to 7: reserved For further information, refer to the example below (1) . **NOTE:** Disabling this protocol prevents the device to receive data from the software CAE.
SNMP Agent	INTEGER	Decimal: 255 Binary: 1111 1111	Allows you to enable or disable the protocol in each communication port: • UDP ports: 161, 162 Bit value: 0 = Disable, 1 = Enable • Bit 0: USB • Bit 1: ETH1 • Bit 2: ETH2 • Bit 3 to 5: TMSES4 1 to 3 • Bit 6 to 7: reserved For further information, refer to the example below (1) .

Key	Type	Default Value	Description
TFTP Server	INTEGER	Decimal: 255 Binary: 1111 1111	Allows you to enable or disable the protocol in each communication port: • UDP port: 69 Bit value: 0 = Disable, 1 = Enable • Bit 0: USB • Bit 1: ETH1 • Bit 2: ETH2 • Bit 3 to 5: TMSES4 1 to 3 • Bit 6 to 7: reserved For further information, refer to the example below (1) .
WebVisualisation Protocol	INTEGER	Decimal: 255 Binary: 1111 1111	Allows you to enable or disable the protocol in each communication port: • TCP port: 8080, 8089 Bit value: 0 = Disable, 1 = Enable • Bit 0: USB • Bit 1: ETH1 • Bit 2: ETH2 • Bit 3 to 5: TMSES4 1 to 3 • Bit 6 to 7: reserved For further information, refer to the example below (1) .

(1)



Bit	7	6	5	4	3	2	1	0
Interface	Reserved	Reserved	ETH5 (TMSES4_2)	ETH4 (TMSES4_1)	ETH3 (TMSES4)	ETH2 (Ethernet_2)	ETH1 (Ethernet_1)	USB
Example:	0	0	0	0	1	0	1	0

In this example, the chosen protocol is allowed on ETH1 and the first TMSES4. It is blocked on the other interfaces. The related binary value 00001010 corresponds to 10 in decimal, so the related parameter value must be set to 10.

Operating Modes

The control of the device security settings via the CAE is enabled by default on the M262 Logic/Motion Controller. To disable CAE, refer to Post Configuration Presentation, page 229.

Once the connection between the CAE and the controller is accepted, the CAE is allowed to send the RBAC or the DSS configurations. After receiving a valid RBAC configuration, the existing users and groups are deleted, and new groups and users are created based on the RBAC configuration.

If you have used CAE to manage security, and then modify the security settings with EcoStruxure Machine Expert, groups and/or user accounts may be deleted, and inconsistencies may occur.

⚠ WARNING

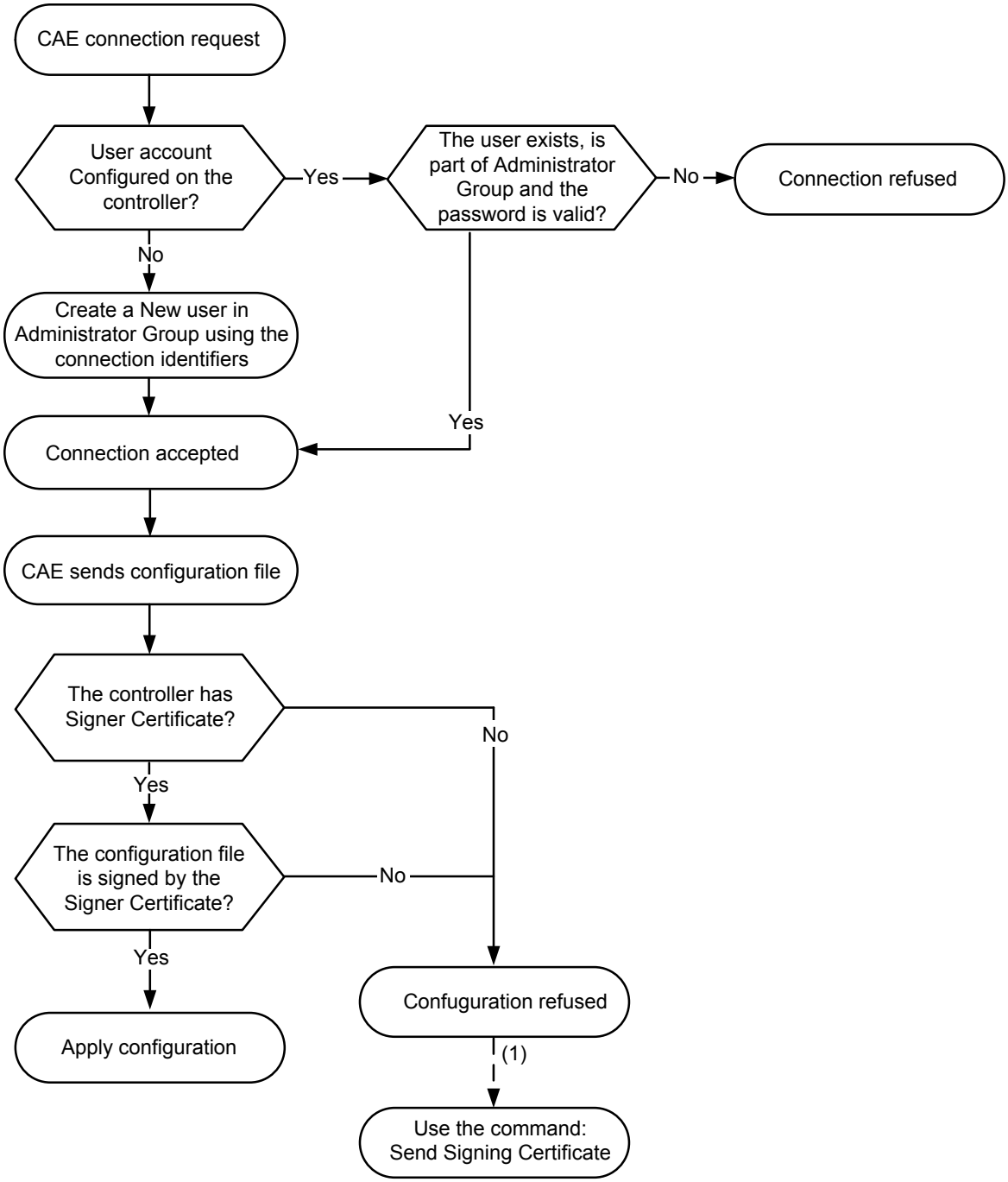
LOSS OF DATA

Do not create the user accounts and groups with EcoStruxure Machine Expert if the security settings are managed by the Cybersecurity Admin Expert (CAE) software.

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

Any DSS configuration received from the CAE is applied immediately.

The following diagram describes the connection between the CAE tool and the controller:



(1) In the case of the first CAE certificate has not been sent yet, use the command **Send Signing Certificate**. Subsequent operation will be proceed automatically.

CAE Options Supported by the M262 Logic/Motion Controller

This table describes the different CAE commands you can use with the M262 Logic/Motion Controller:

Commands	Description
Discover device(s) over the network	Displays the controller in the list of the discovered devices.
Send Signing Certificate	Saves the CAE signing certificate inside the file system. Any configuration received is rejected if the signing certificate is not provisioned. Only one signing certificate is supported.
Send Security Configuration	Sends the RBAC and DSS configuration files and apply the configuration.
Log on	Connects manually the controller when automatic connection did not succeed.
Send DSS	Sends the DSS and to apply the configuration, once it has been validated by the CAE tool, using the CAE signing certificate.
Reset	Resets the security configuration to default. RBAC (users, roles, permissions) and DSS are reset.
Locate	Locates the device by flashing its LED.
Certificate Management > Whitelist	Adds or removes certificate(s) in the whitelist (allowlist).
Certificate Management > Signers	Adds or removes the CAE certificate that can be used to verify the configuration signature. Only one Signer certificate is supported.
Certificate Management > Trusted Chain Certificates	Add a root or intermediate certificate to the controller trusted list. Manages **root CA** and **intermediate CA** certificates, so that the controller can verify the certificates chain of trust.
PKI Management > Get CSR	Allows the controller to generate and send the CSR (certificate sign request) for OPC UA certificate.
PKI Management > Send signed device certificate	Allows you to replace the self signed certificate by the certificate signed by the Certificate Authority (CA) and provided to CAE. The certificate requires a **Reset Cold**, a **Reset Warm** or a **Reboot** of the application to apply.

If a command is not active (in grey in the software), refer to the Cybersecurity Admin Expert User Manual.

This table describes the Public Key Infrastructure (PKI) shared between the M262 Logic/Motion Controller and the CAE. It provides the folder list and their usage.

M262 File System Folders	Description
/usr/pki/cae/castore	Stores working certificate received from CAE.
/usr/pki/cae/csr	Stores the signed certificate request.

How to Configure the Controller

Introduction

This chapter shows the default configuration of a project.

Configuring the Controller

Introduction

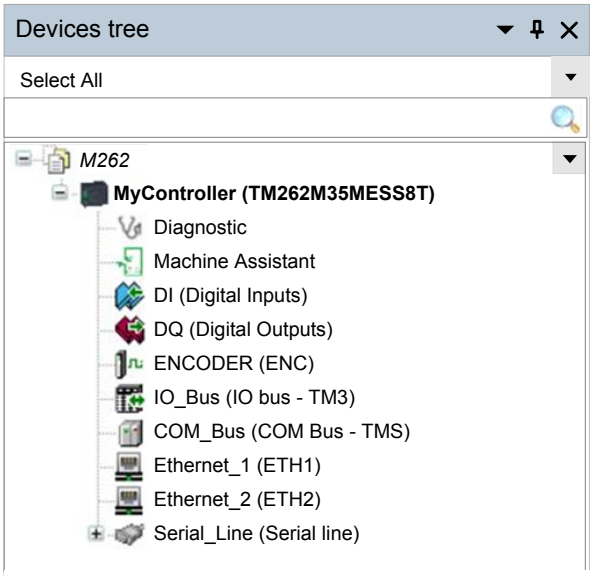
First, create a new project or open an existing project in the EcoStruxure Machine Expert software.

Refer to the EcoStruxure Machine Expert Programming Guide for information on how to:

- Add a controller to your project
- Add expansion modules to your controller
- Replace an existing controller
- Convert a controller to a different but compatible device

Devices Tree

The **Devices tree** presents a structured view of the hardware configuration. When you add a controller to your project, a number of nodes are added to the **Devices tree**, depending on the functions the controller provides.



Item	Use to Configure...
Diagnostic	Diagnostic messages and status.
Machine Assistant	Devices discovery and configuration.
DI	Embedded digital inputs of the controller.
DQ	Embedded digital outputs of the controller.
ENCODER	Incremental or SSI Encoder interface of the controller.
IO_Bus	Expansion modules connected to the controller.
COM_Bus	Communication modules connected to the controller.
Ethernet_1	Embedded Ethernet dedicated to Motion Bus Sercos on TM262M•, dedicated to devices on TM262L•.
Ethernet_2	Embedded Ethernet communication.
Serial_Line	Serial line communication interface.

Applications Tree

The **Applications tree** allows you to manage project-specific applications as well as global applications, POUs, and tasks.

Tools Tree

The **Tools tree** allows you to configure the HMI part of your project and to manage libraries.

The **Tools tree** allows you to:

- Configure the HMI part of your project.
- Access to the **Library Manager** tool.
- Access to the **Message logger** tool, page 147.

Libraries

Introduction

This chapter describes the default libraries of the Modicon M262 Logic/Motion Controller.

Libraries

Overview

Libraries provide functions, function blocks, data types and global variables that can be used to develop your project.

The **Library Manager** provides information about the libraries included in your project and allows you to install new ones. For more information on the **Library Manager**, refer to the Functions and Libraries User Guide. For more information about the libraries compatible with your controller, refer to the EcoStruxure Machine Expert Libraries Overview.

Modicon M262 Logic/Motion Controller

When you select a Modicon M262 Logic/Motion Controller for your application, EcoStruxure Machine Expert automatically loads the following libraries:

Library name	Description
Breakpoint Logging Functions (1)	Provides functions which can be used in Breakpoints for logging.
DeviceAbstractionLayer (1)	Interfaces and parameters of functionalities which are exposed by device objects.
DeviceIntegrationCommon (1)	Common functionalities shared by many devices (reserved for internal use only).
Diagnostic Device Support	Provides function blocks (reserved for internal use only) for devices to provide diagnostic information to the system diagnostic component on Schneider Electric controllers.
IoStandard	Library for the IO-configuration. This library provides the I/O interface for every IEC I/O driver.
M262 PLCSystem	M262 system functions and variables.
M262MotionExtension (1)	Exposes the on-board encoder input and touch-probe functionalities of an M262 to the controller application (reserved for internal use only).
MotionInterface (1)	Low level access for motion control.
PLCCommunication	Management of explicit data exchanges between controller and devices through Modbus or ASCII protocols.
PLCopen MC part 1 (1)	Motion control according to PLCopen Motion Control Part 1 v2.0 (formerly parts 1 and 2).
Relocation Table	Allows you to optimize exchanges between the Modbus client and the controller, by regrouping non-contiguous data into a contiguous table of registers. See Relocation Table, page 45.
SerialLineSystem	Provides serial line diagnostics data.
Standard	Contains IEC programming standard functions and function blocks.
TM3System	Contains functions and function blocks for TM3 I/O bus diagnostic information.
TMSSystem	Contains the function block and enumerated types for TMS I/O bus diagnostic information.
UserFunctionsBase (1)	Core implementation for exposing device functionalities to the controller application.
Util	Programming additional functions and function blocks: Analog Monitors, BCD Conversions, Bit/Byte Functions, Controller Datatypes, Function Manipulators, Mathematical Functions, Signals.
(1) Compatible with TM262M• references only.	

Supported Standard Data Types

Introduction

This chapter provides the different IEC data types supported by the Controller.

Supported Standard Data Types

Overview

The controller supports the following IEC data types:

Data Type	Lower Limit	Upper Limit	Information Content
BOOL	FALSE	TRUE	1 Bit
BYTE	0	255	8 Bit
WORD	0	65,535	16 Bit
DWORD	0	4,294,967,295	32 Bit
LWORD	0	2^{64}-1	64 Bit
SINT	-128	127	8 Bit
USINT	0	255	8 Bit
INT	-32,768	32,767	16 Bit
UINT	0	65,535	16 Bit
DINT	-2,147,483,648	2,147,483,647	32 Bit
UDINT	0	4,294,967,295	32 Bit
LINT	-2^{63}	2^{63}-1	64 Bit
ULINT	0	2^{64}-1	64 Bit
REAL	1.175494351e-38	3.402823466e+38	32 Bit
LREAL	2.2250738585072014e-308	1.7976931348623158e+308	64 Bit
STRING	1 character	–	1 character = 1 byte
WSTRING	1 character	–	1 character = 1 word
TIME	0	4294967295	32 Bit

For more information on `ARRAY`, `LTIME`, `DATE`, `TIME`, `DATE_AND_TIME`, and `TIME_OF_DAY`, refer to the EcoStruxure Machine Expert Programming Guide.

Memory Mapping

Introduction

This chapter describes the memory maps and sizes of the different memory areas in the Modicon M262 Logic/Motion Controller. These memory areas are used to store user program logic, data and the programming libraries.

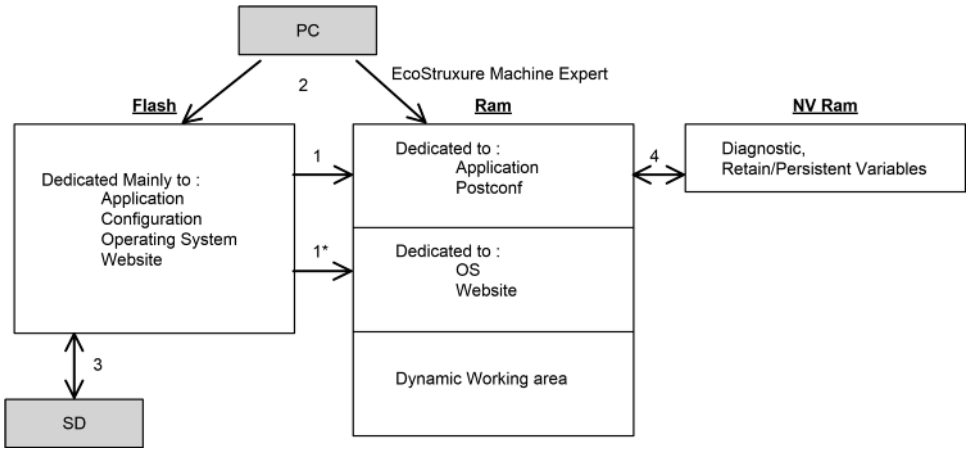
Controller Memory Organization

Introduction

The controller memory is composed of three types of physical memory:

- The Non-Volatile Memory (NVM), page 40 contains files (application, configuration files, and so on).
- The Random Access Memory (RAM), page 43 is used for application execution.
- The Non-Volatile Random Access Memory (NVRAM), page 44 is used to save the retain-persistent variables and diagnostic information.

Files Transfers in Memory



Item	Controller State	File Transfer Events	Connection	Description
1	–	Initiated automatically at Power ON and Reboot	Internal	Files transfer from non-volatile memory to RAM. The content of the RAM is overwritten.
1*	–	Initiated automatically at Power ON and Reboot	Internal	Operating system files transfer.
2	All states except INVALID_OS (1)	Initiated by user	Ethernet or USB programming port	Files can be transferred via: • Web Server, page 132 • FTP server, page 129 • **Controller Assistant** • The software (see the EcoStruxure Machine Expert Programming Guide)
3	All states	Initiated automatically by script (data transfer) or by power cycle (cloning) when an SD card is inserted	SD card	Up/download with SD card(1).
4	All states	Initiated by system	Internal	Save of modified retain-persistent variables and the context on power OFF.
(1) If the controller is in the INVALID_OS state, the only accessible memory is the SD card and only for firmware upgrades.				

NOTE: The modification of files in non-volatile memory does not affect a running application. Any changes to files in non-volatile memory are taken into account at the next reboot, except for the user files directly used by the application.

Non-Volatile Memory Organization

Introduction

The non-volatile memory contains the file system used by the controller.

File Type

The Modicon M262 Logic/Motion Controller manages the following file types:

System function (/sys)	Description
Operating System (OS)	Controller firmware that can be written to non-volatile memory. The firmware file is applied at next reboot of the controller.

User functions (/usr)	Description
Boot application	This file resides in non-volatile memory and contains the compiled binary code of the executable application. Each time the controller is rebooted, the executable application is extracted from the boot application and copied into the controller RAM (1).
Application source	Source file that can be uploaded from non-volatile memory to the PC if the source file is not available on the PC (2).
Post configuration	File that contains Ethernet and serial line parameters. The parameters specified in the file override the parameters in the executable application at each reset.
Firewall parameters	Settings used to configure the firewall of the M262 Logic/Motion Controller. These settings restrict access to authorized personnel and protocols only. See Firewall Configuration, page 153 for more information.
Data logging	Files in which the controller logs events as specified by the application.

(1) The creation of a boot application is optional in EcoStruxure Machine Expert, according to application properties. Default option is to create the boot application on download. When you download an application from the software to the controller, you are transferring only the binary executable application directly to RAM.

(2) EcoStruxure Machine Expert does not support uploading of either the executable application or the boot application to a PC for modification. Program modifications must be made to the application source. When you download your application, you have the option to store the source file to the non-volatile memory.

Log File Type

This table shows the different types of log files that are generated and available in */usr/Syslog* during operation/usage of Modicon M262 Logic/Motion Controller:

Log file type	Buffer behavior
FwLog	Log entries from early boot-up stages. A back-up file is created when the original file reaches 256 kB.
PlcLog	Log entries from supplier (CoDeSys). Once the file reaches 100 kB, a new file is created (PlcLog_0.txt), and so on for a maximum of 3 files.
LoggerFile	Log entries from both Run-time and Build-time. Each file can store a maximum of 500 entries. Once this number is reached, a new file is created (LoggerFile_*.mel), and so on for a maximum of 300 files.

File Organization

This table shows the file organization of the non-volatile memory:

Disk	Directory	File	Content	Up/Downloaded data type
/sys	Pkg	Temporary file	Internal use	N/A
/usr	App	Application.app	Boot application	Application
		Application.crc		–
		Archive.prj (1)	Application source	–
	Cfg	Machine.cfg (1)	Post configuration file, page 229	Configuration
		CodesysLateConf.cfg	Configuration of application by CoDeSys.	Configuration
		FirewallDefault.cmd	Default firewall settings. By default, this file does not exist. It can be added optionally.	Configuration
		ntp.conf	Contains the network time protocol (NTP) configuration.	Configuration
		ntp.drift.	Contains the calculated drift of the system clock compared to UTC time.	Configuration
	Log	*UserDefinedLogName_1*.log	All *.log files created using the data logging functions (see Data Logging Functions, DataLogging Library Guide). You must specify the total number of files created and the names and contents of each log file using the data logging feature.	Log file
		UserDefinedLogName_n.log		–
	pki	–	Certificate store for M262 secured protocols.	–
	Rcp	–	Main directory for recipes.	–
	Syslog	crash.txt (1) LoggerFile_xxx.mel	Record of detected system errors for use by Schneider Electric Technical Support. Refer to Log File Type, page 40.	Log file
	Visu	–	Used for the WebVisualisation feature.	–
	_cnc	UserDefinedName.cnc	Pre-programmed control commands	G-code data
	Alarms	Application.alarmstorage.X.sqlite Application.alarmstorage.X.sqlite.metadata	Database of configured alarms	Alarm manager data
	Trend	Application.TrendRecording.X.sqlite Application.TrendRecording.X.sqlite.metadata	Database of configured trends. Refer to Trend Storage Limits, page 41.	Trend recorder data
/sd0	–	–	SD card. Refer to Managing Script Files, page 246.	–
	–	User files	–	–
(1) If the files had been created due to specific events or customer requirements.				

NOTE: For more information on libraries and available function blocks, refer to Libraries, page 36.

Trend Storage Limits

This table shows the storage limits of the Trend feature:

Element	Limit
Number of variables	255 maximum
Storage size	250 MB maximum (including 1 MB if the Alarms feature is used)

NOTE: Due to the intensive processing of the Trend recording mechanism, reduce the number of variables and set the storage size to a lower limit, for example: 2 MB.

For more information on Trend feature, refer to the EcoStruxure Machine Expert Programming Guide.

Files Redirection

When system, program or certain user activity creates specific file types, the M262 Logic/Motion Controller examines the file extension and automatically moves the file to a corresponding folder in non-volatile memory.

The following table lists the file types that are moved in this way and the destination folder in non-volatile memory:

File extensions	Non-volatile memory folder
*.app, *.ap_, *.err, *.crc, *.frc, *.prj	/usr/App
*.cfg, *.cf_	/usr/Cfg
*.log	/usr/Log
*.rcp, *.rsi	/usr/Rcp
*.urf, *.csv, *.csv_	/usr/URF

Backup Data Logging File

Data logging files can become large to the point of exceeding the space available in the file system. Therefore, you should develop a method to archive the log data periodically on an SD card. You could split the log data into several files, for example `LogMonth1, LogMonth2`, and use the **ExecuteScript** command to copy the first file to an SD card. Afterwards, you may remove it from the internal file system while the second file is accumulating data. If you allow the data logging file to grow and exceed the limits of the file size, you could lose data.

NOTICE

LOSS OF APPLICATION DATA

- Backup SD card data regularly.
- Do not remove power or reset the controller, and do not insert or remove the SD card while it is being accessed.

Failure to follow these instructions can result in equipment damage.

RAM Memory Organization

Introduction

This section describes the Random Access Memory (RAM) of the Modicon M262 Logic/Motion Controller.

Memory Mapping

The RAM size is detailed in Memory, page 15.

The RAM is composed of two areas:

- Dedicated application memory
- OS memory

This table describes the dedicated application memory:

Area	Element
System area	System Area Mappable Addresses %MW0...%MW59999
	System and diagnostic variables (%MW60000...%MW60199) This memory is accessible through Modbus requests only. These must be read-only requests.
	Dynamic Memory Area: Read Relocation Table, page 45 (%MW60200...%MW61999) This memory is accessible through Modbus requests only. These must be read-only requests.
	System and diagnostic variables (%MW62000...%MW62199) This memory is accessible through Modbus requests only. These can be read or write requests.
	Dynamic Memory Area: Write Relocation Table, page 45 (%MW62200...%MW63999) This memory is accessible through Modbus requests only. These can be read or write requests.
User area	Symbols
	Variables
	Libraries
	Application

To display the memory mapping in EcoStruxure Machine Expert, right-click on your controller in the **Devices tree** window and select **Device Memory Info**.

NOTE: If the controller memory usage exceeds 85%, you may be unable to login to EcoStruxure Machine Expert. In that case, restart your controller.

System and Diagnostic Variables

Variables	Description
PLC_R	Structure of controller read-only system variables.
PLC_W	Structure of controller read/write system variables.
ETH_R	Structure of Ethernet read-only system variables (Ethernet counters).
ETH_W	Structure of Ethernet read/write system variables. Allows you to reset Ethernet counters.
SERIAL_R	Structure of Serial Line read-only system variables (Serial Line counters).
SERIAL_W	Structure of Serial Line read-write system variables. Allows you to reset Serial Line counters.
TM3_MODULE_R	Structure of TM3 modules read-only system variables.
TM3_BUS_W	Structure of TM3 bus read-write system variables.
TMS_BUS_DIAG_R	Structure of TMS bus read-only system variables (diagnostic).
TMS_MODULE_DIAG_R	Structure of TMS modules read-only system variables (diagnostic).

For more information on system and diagnostic variables, refer to the Modicon M262 Logic/Motion Controller, System Functions and Variables, System Library Guide.

NVRAM Memory Organization

Introduction

The NVRAM memory contains:

- Files saved for the diagnostics
- Remanent (retain-persistent) variables

NVRAM Size

The following table describes the size of the NVRAM:

User Function	Description	Size
System diagnostics	Contain the controller context saved on power off.	128 kbytes
Remanent (retain-persistent) variables	Modified and saved in the NVRAM. This action impacts the cycle time. Retain: saved after each cycle. Persistent: saved after each modification.	Retain: 64 kbytes Persistent: 64 kbytes

Remanent or retain-persistent variables are saved in the NVRAM. Each subsequent read/write access to these variables requires a NVRAM access. For more information about remanent variables, see Remanent Variables, page 73. For more information about the impact on performance, see Processing Performance, page 288.

NOTE: For an optimum cycle time, only access the retain-persistent variables when necessary. For frequent (Read) access, copy theses variables to a working memory in the RAM.

Relocation Table

Introduction

The **Relocation Table** allows you to organize data to optimize communication between the controller and other equipment by regrouping non-contiguous data into a contiguous table of located registers, accessible through Modbus.

NOTE: A relocation table is considered an object. Only one relocation table object can be added to a controller.

Relocation Table Description

This table describes the **Relocation Table** organization:

Register	Description
60200...61999	Dynamic Memory Area: Read Relocation Table The `%MW` registers are read from the variables at each cycle.
62200...63999	Dynamic Memory Area: Write Relocation Table The `%MW` registers are copied to the variables at each cycle.

For further information, refer to the Modicon M262 Logic/Motion Controller, System Functions and Variables, System Library Guide.

Adding a Relocation Table

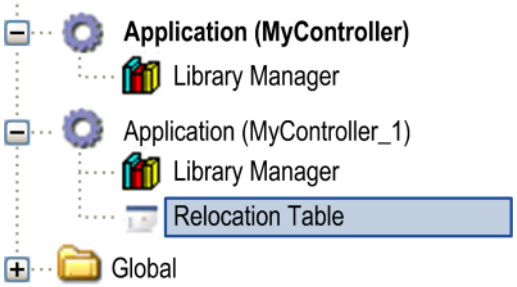
This table describes how to add a **Relocation Table** to your project:

Step	Action
1	In the **Applications tree** tab, select the **Application** node.
2	Click the right mouse button.
3	Click **Objects** > **Relocation Table...**. **Result:** The **Add Relocation Table** window is displayed.
4	Click **Add**. **Result:** The new relocation table is created and initialized. **NOTE:** As a relocation table is unique for a controller, its name is **Relocation Table** and cannot be changed.

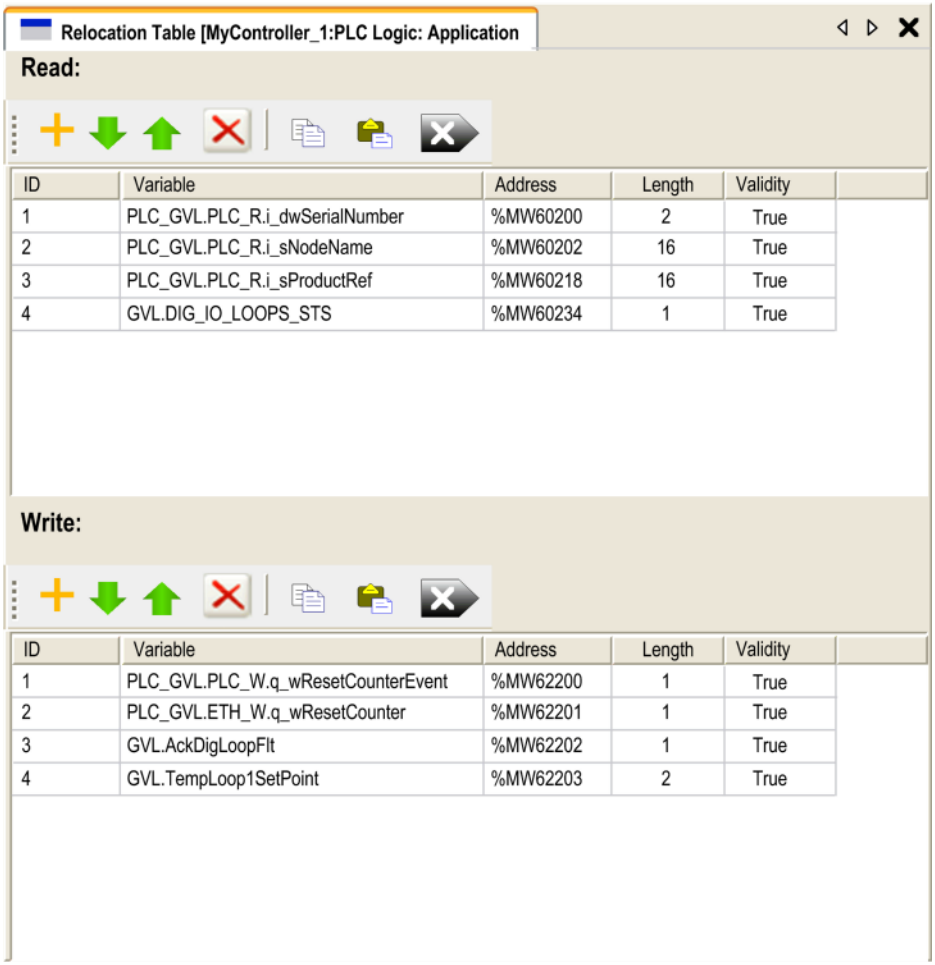
Relocation Table Editor

The relocation table editor allows you to organize your variables in the relocation table.

To access the relocation table editor, double-click the **Relocation Table** node in the **Tools tree** tab:



This picture describes the relocation table editor:



ID	Variable	Address	Length	Validity
1	PLC_GVL.PLC_R.i_dwSerialNumber	%MW60200	2	True
2	PLC_GVL.PLC_R.i_sNodeName	%MW60202	16	True
3	PLC_GVL.PLC_R.i_sProductRef	%MW60218	16	True
4	GVL.DIG_IO_LOOPS_STS	%MW60234	1	True

ID	Variable	Address	Length	Validity
1	PLC_GVL.PLC_W.q_wResetCounterEvent	%MW62200	1	True
2	PLC_GVL.ETH_W.q_wResetCounter	%MW62201	1	True
3	GVL.AckDigLoopFlt	%MW62202	1	True
4	GVL.TempLoop1SetPoint	%MW62203	2	True

Icon	Element	Description
	New Item	Adds an element to the list of mapped variables.
	Move Down	Moves down the selected element of the list.
	Move Up	Moves up the selected element of the list.
	Delete Item	Removes the selected elements of the list.
	Copy	Copies the selected elements of the list.
	Paste	Pastes the elements copied.
	Erase Empty Item	Removes all the elements of the list for which the "Variable" column is empty.
-	ID	Automatic incremental integer (not editable).
-	Variable	The name or the full path of a variable (editable).
-	Address	The address of the system area where the variable is stored (not editable).
-	Length	Variable length in word.
-	Validity	Indicates if the entered variable is valid (not editable).

NOTE: If a variable is undefined after program modifications, the content of the cell is displayed in red, the related **Validity** cell is False, and **Address** is set to -1.

Tasks

Introduction

The **Task Configuration** node in the **Applications tree** allows you to define one or more tasks to control the execution of your application program.

The task types available are:

- Cyclic
- Freewheeling
- Event
- External event

This chapter begins with an explanation of these task types and provides information regarding the maximum number of tasks, the default task configuration, and task prioritization. In addition, this chapter introduces the system and task watchdog functions and explains its relationship to task execution.

Maximum Number of Tasks

Maximum Number of Tasks

The maximum number of tasks you can define for the Modicon M262 Logic/Motion Controller are:

- Total number of tasks = 16
- Cyclic tasks = 8
- Freewheeling tasks = 1
- Cyclic tasks + freewheeling tasks = 8
- Event tasks = 8
- External Event tasks = 8

Special Considerations for Freewheeling

A Freewheeling task, page 50 does not have a fixed duration. In Freewheeling mode, the task scan starts when the previous scan has been completed and after a period of system processing (30% of the total duration of the Freewheeling task). If the system processing period is reduced to less than 15% for more than 3 seconds due to interruptions by other tasks, a system error is detected. For more information, refer to System and Task Watchdogs, page 53.

NOTE: You may wish to avoid using a Freewheeling task in a multi-task application when some high priority and time-consuming tasks are running. Doing so may provoke a task Watchdog Timeout. You should not assign CANopen to a freewheeling task. CANopen should be assigned to a cyclic task.

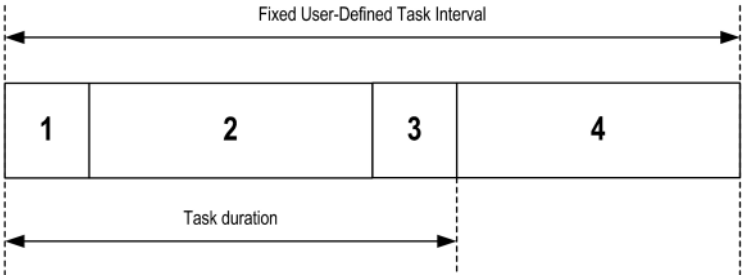
Task Types

Introduction

The following section describes the various task types available for your program, along with a description of the task type characteristics.

Cyclic Task

A Cyclic task is assigned a fixed cycle time using the interval setting in the type section of the configuration subtab for that task. Each Cyclic task type executes as follows:

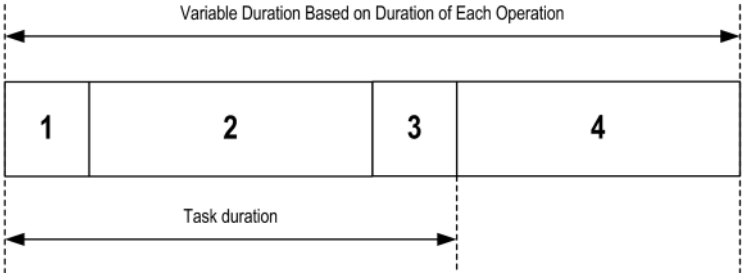


1.	**Read Inputs:** The physical input states are written to the `%I` input memory variables and other system operations are executed.
2.	**Task Processing:** The user code (POU, and so on) defined in the task is processed. The `%Q` output memory variables are updated according to your application program instructions but not yet written to the physical outputs during this operation.
3.	**Write Outputs:** The `%Q` output memory variables are modified with the output forcing that has been defined; however, the writing of the physical outputs depends upon the type of output and instructions used. For more information on defining the bus cycle task, refer to the EcoStruxure Machine Expert Programming Guide and PLC Settings, page 78. For more information on I/O behavior, refer to Controller States Detailed Description, page 60.
4.	**Remaining Interval time:** The controller firmware carries out system processing and other lower priority tasks.

NOTE: If you define too short a period for a cyclic task, it will repeat immediately after the write of the outputs and without executing other lower priority tasks or any system processing. This will affect the execution of all tasks and cause the controller to exceed the system watchdog limits, generating a system watchdog exception.

Freewheeling Task

A Freewheeling task does not have a fixed duration. In Freewheeling mode, each task scan begins when the previous scan has been completed and after a short period of system processing. Each Freewheeling task type executes as follows:



1.	**Read Inputs:** The physical input states are written to the `%I` input memory variables and other system operations are executed.
2.	**Task Processing:** The user code (POU, and so on) defined in the task is processed. The `%Q` output memory variables are updated according to your application program instructions but not yet written to the physical outputs during this operation.
3.	**Write Outputs:** The `%Q` output memory variables are modified with the output forcing that has been defined; however, the writing of the physical outputs depends upon the type of output and instructions used. For more information on defining the bus cycle task, refer to the EcoStruxure Machine Expert Programming Guide and PLC Settings, page 78. For more information on I/O behavior, refer to Controller States Detailed Description, page 60.
4.	**System Processing:** The controller firmware carries out system processing and the other lower priority tasks (for example: HTTP management, Ethernet management, parameters management).

NOTE: If you want to define the task interval, refer to Cyclic Task, page 49.

Event Task

This type of task is event-driven and is initiated by a program variable. It starts at the rising edge of the boolean variable associated to the trigger event unless pre-empted by a higher priority task. In that case, the Event task will start as dictated by the task priority assignments.

For example, if you have defined a variable called `my_Var` and would like to assign it to an Event, proceed as follows:

Step	Action
1	Double-click the **TASK** in the **Applications tree**.
2	Select **Event** from the **Type** list in the **Configuration** tab.
3	Click the **Input Assistant** button to the right of the **Event** field. **Result**: The **Input Assistant** window appears.
4	Navigate in the tree of the **Input Assistant** dialog box to find and assign the `my_Var` variable.

NOTE: When the event task is triggered with too high a frequency, the controller may detect an error and transition to the HALT state (Exception).

The maximum rate of events is:

- 12 events per millisecond for TM262L01MESE8T, TM262L10MESE8T, TM262M05MESS8T and TM262M15MESS8T
- 16 events per millisecond for TM262L20MESE8T, TM262M25MESS8T and TM262M35MESS8T

If the event task is triggered at a higher frequency than this, the message 'ISR Count Exceeded' is logged in the application log page.

External Event Task

This type of task is event-driven and is initiated by the detection of a hardware or hardware-related function event. It starts when the event occurs unless pre-empted by a higher priority task. In that case, the External Event task will start as dictated by the task priority assignments.

For example, an External event task could be associated with an HSC Stop event. To associate the **HSC0_STOP** event to an External event task, select it from the **External event** drop-down list on the **Configuration** tab.

The external event task can be associated with the CAN Sync event. To associate the **CAN_1_SYNC** event to an external event task, select it from the **External event** dropdown list in the **Configuration** tab.

Different types of events can be associated with an External event task:

- HSC thresholds (see Modicon TM3 Expert I/O Modules, HSC Library Guide)
- HSC Stop
- CAN Sync
- AFTER_RTP
- HSC Event Periodmeter
- Event input

NOTE: CAN Sync is a specific event object, depending on the **CANopen manager** configuration.

NOTE: When the external event task is triggered with too high a frequency, the controller may detect an error and transition to the HALT state (Exception).

The maximum rate of events is:

- 12 events per millisecond for TM262L01MESE8T, TM262L10MESE8T, TM262M05MESS8T and TM262M15MESS8T
- 16 events per millisecond for TM262L20MESE8T, TM262M25MESS8T and TM262M35MESS8T

If the event task is triggered at a higher frequency than this, the message 'ISR Count Exceeded' is logged in the application log page.

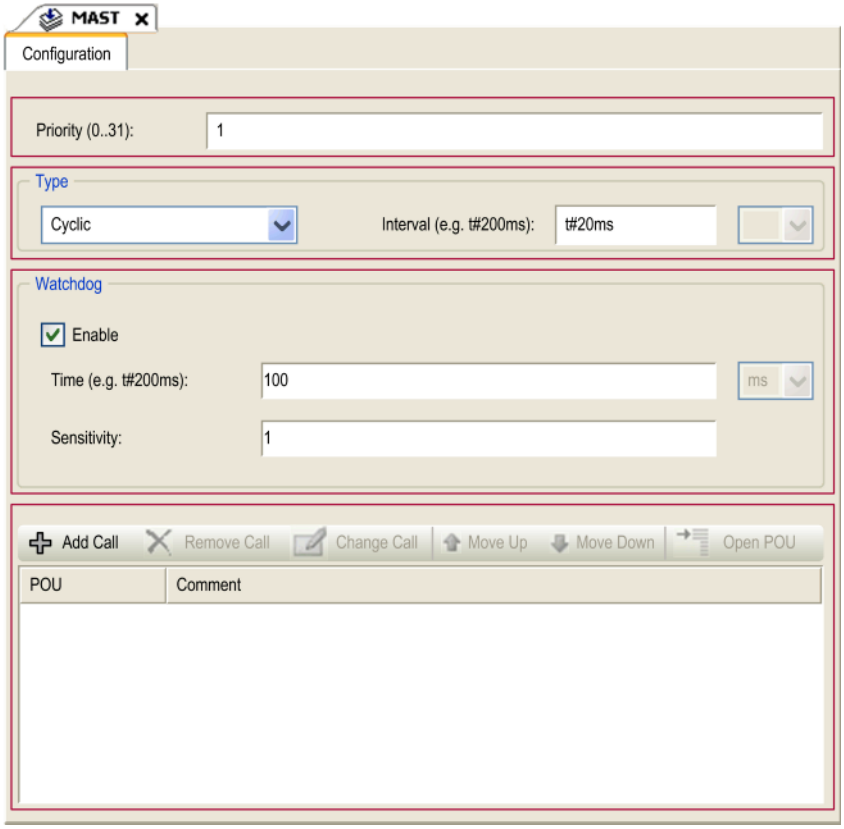
Task Configuration Screen

Screen Description

This screen allows you to configure the tasks. Double-click the task that you want to configure in the **Applications tree** to access this screen.

Each configuration task has its own parameters that are independent of the other tasks.

The **Configuration** window is composed of four parts:



The table describes the fields of the **Configuration** screen:

Field Name	Definition
Priority	Configure the priority of each task with a number from 0 to 31 (0 is the highest priority, 31 is the lowest). Only one task at a time can be running. The priority determines when the task runs: a higher priority task pre-empts a lower priority task. **NOTE:** Do not assign tasks with the same priority. If there are yet other tasks that attempt to pre-empt tasks with the same priority, the result could be indeterminate and unpredictable. For important information, refer to Task Priorities, page 54.
Type	These task types are available: • Cyclic, page 49 • Event, page 50 • External , page 51 • Freewheeling, page 50
Watchdog	To configure the watchdog, page 53, define these two parameters: • **Time**: enter the timeout before watchdog execution. • **Sensitivity**: defines the number of expirations of the watchdog timer before the controller stops program execution and enters a HALT state.
POUs	The list of Programming Organization Units (POUs) controlled by the task is defined in the task configuration window found in the EcoStruxure Machine Expert Programming Guide: • To add a POU linked to the task, use the command **Add Call** and select the POU in the **Input Assistant** editor. • To remove a POU from the list, use the command **Remove Call**. • To replace the selected POU of the list by another one, use the command **Change Call**. • **POU**s are executed in the order shown in the list. To move the **POU**s in the list, select a **POU** and use the command **Move Up** or **Move Down**. **NOTE:** You can create as many POUs as you want. An application with several small POUs, as opposed to one large POU, can improve the refresh time of the variables in online mode.

System and Task Watchdogs

Introduction

Two types of watchdog functionality are implemented for the Modicon M262 Logic/Motion Controller:

- **System Watchdogs**: These watchdogs are managed by the controller firmware. You cannot configure them.
- **Task Watchdogs**: These watchdogs are optional watchdogs that you can define for each task. These are configurable in EcoStruxure Machine Expert.

System Watchdogs

Three system watchdogs are defined for the Modicon M262 Logic/Motion Controller. They are managed by the controller firmware and are therefore sometimes referred to as hardware watchdogs in the online help. When one of the system watchdogs exceeds its threshold conditions, an error is detected.

The threshold conditions for the three system watchdogs are defined as follows:

- If all of the tasks require more than 85% of the processor resources for more than 3 seconds, a system error is detected. The controller enters the HALT state.
- If the total execution time of the tasks with priorities between 0 and 24 reaches 100% of processor resources for more than 1 second, an application error is detected. The controller responds with an automatic reboot into the EMPTY state.
- If the lowest priority task of the system is not executed during an interval of 10 seconds, a system error is detected. The controller responds with an automatic reboot into the EMPTY state.

NOTE: You cannot configure the system watchdogs.

Task Watchdogs

EcoStruxure Machine Expert allows you to configure an optional task watchdog for every task defined in your application program. Task watchdogs are sometimes also referred to as software watchdogs or control timers in the online help. When one of your defined task watchdogs reaches its threshold condition, an application error is detected and the controller enters the HALT state.

When defining a task watchdog, the following options are available:

- **Time**: This defines the maximum execution time for a task. When a task takes longer than this, the controller reports a task watchdog exception.
- **Sensitivity**: The sensitivity field defines the number of task watchdog exceptions that must occur before the controller detects an application error.

To access the configuration of a task watchdog, double-click the **Task** in the **Applications tree**.

NOTE: For more information on watchdogs, refer to watchdog settings found in the EcoStruxure Machine Expert Programming Guide.

Task Priorities

Task Priority Configuration

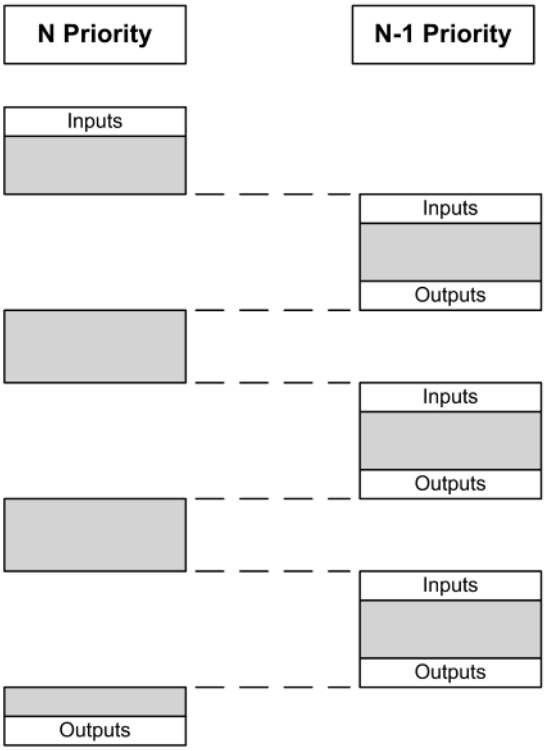
You can configure the priority of each task between 0 and 31 (0 is the highest priority, 31 is the lowest). Each task must have a unique priority. Assigning the same priority to more than one task leads to a build error.

Task Priority Suggestions

- Priority 0 to 24: Controller tasks. Assign these priorities to tasks with a high availability requirement.
- Priority 25 to 31: Background tasks. Assign these priorities to tasks with a low availability requirement.

Task Priorities of Embedded I/Os

When a task cycle starts, it can interrupt any task with lower priority (task preemption). The interrupted task resumes when the higher priority task cycle is finished.



NOTE: If the same input is used in different tasks the input image may change during the task cycle of the lower priority task.

To improve the likelihood of proper output behavior during multitasking, a build error message is displayed if outputs in the same byte are used in different tasks.

⚠WARNING

UNINTENDED EQUIPMENT OPERATION

Map your inputs so that tasks do not alter the input images in an unexpected manner.

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

Task Priorities of TM3 Modules and CANopen I/Os

You can select the task that drives TM3 I/Os and CANopen physical exchanges. In the **PLC settings**, select **Bus cycle task** to define the task for the exchange. By default, the task is set to **MAST**. This definition at the controller level can be overridden by the I/O bus configuration, page 116. During the read and write phases, all physical I/Os are refreshed at the same time. TM3 and CANopen data is copied into a virtual I/O image during a physical exchanges phase, as shown in this figure:

Inputs are read from the I/O image table at the beginning of the task cycle. Outputs are written to the I/O image table at the end of the task.

NOTE: TM3 influence the application execution time. You can configure the **Bus cycle options** using **I/O mapping** tab. Refer to the Modicon TM3 Expansion Modules – Programming Guide.

Default Task Configuration

Default Task Configuration

The MAST task can be configured in Freewheeling or Cyclic mode. The MAST task is automatically created by default in Cyclic mode. Its preset priority is medium (15), its preset interval is 10 ms, and its task watchdog service is activated with a time of 50 ms and a sensitivity of 1. Refer to Task Priorities, page 54 for more information on priority settings. Refer to Task Watchdogs for more information on watchdogs.

Designing an efficient application program is important in systems approaching the maximum number of tasks. In such an application, it can be difficult to keep the resource utilization below the system watchdog threshold. If priority reassignments alone are not sufficient to remain below the threshold, some lower priority tasks can be made to use fewer system resources if the SysTaskWaitSleep function, contained in the SysTask library, is added to those tasks.

NOTE: Do not delete or change the name of the MAST task. Otherwise, the software detects an error when you attempt to build the application, and you are not able to download it to the controller.

Controller States and Behaviors

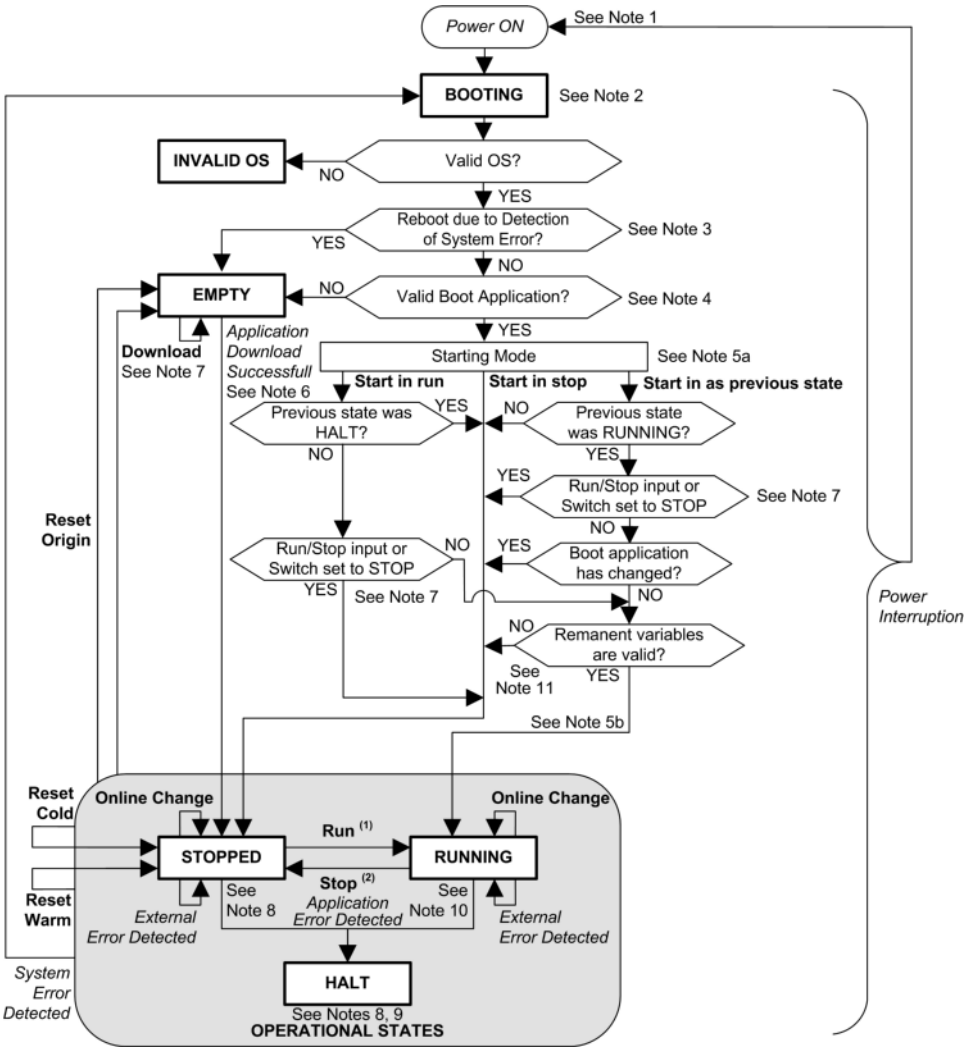
Introduction

This chapter provides information on controller states, state transitions, and behaviors in response to system events. It begins with a detailed controller state diagram and a description of each state. It then defines the relationship of output states to controller states before explaining the commands and events that result in state transitions. It concludes with information about remanent variables and the effect of task programming options on the behavior of your system.

Controller State Diagram

Controller State Diagram

This diagram describes the controller operating mode:



ALL-CAPS BOLD: Controller states

Bold: User and application commands

***Italics*:** System events

Normal text: Decisions, decision results, and general information

(1) For details on STOPPED to RUNNING state transition, refer to Run Command, page 66.

(2) For details on RUNNING to STOPPED state transition, refer to Stop Command, page 66.

Note 1

The alarm relay is open.

Note 2

The outputs assume their hardware initialization states. The encoder power supply is not enabled. The voltage is 0. The alarm relay is closed.

Note 3

In some cases, when a system error is detected, it causes the controller to reboot automatically into the EMPTY state as if no Boot application were present in the non-volatile memory. However, the Boot application is not deleted from the non-volatile memory. In this case, the **ERR LED** (red) flashes fast and regularly.

Note 4

After verification of a valid Boot application the following events occur:

- The application is loaded into RAM.
- The Post Configuration, page 229 file settings (if any) are applied.

During the load of the boot application, a Check context test occurs to verify that the remanent variables are valid. If the Check context test is invalid, the boot application loads but the controller transitions to the STOPPED state, page 70.

Note 5a

The **Starting Mode** is set in the **PLC settings** tab of the **Controller Device Editor**, page 78.

Note 5b

When a power interruption occurs, the controller continues in the RUNNING state for at least 4 ms before shutting down. If you have configured and provide power to the Run/Stop input from the same source as the controller, the loss of power to this input is detected immediately, and the controller behaves as if a STOP command was received.

Therefore, if you provide power to the controller and the Run/Stop input from the same source, your controller reboots normally into the STOPPED state after a power interruption when **Starting Mode** is set to **Start as previous state**.

Note 6

During a successful application download the following events occur:

- The application is loaded directly into RAM.
- By default, the Boot application is created and saved into the non-volatile memory.

Note 7

The default behavior after downloading an application program is for the controller to enter the STOPPED state irrespective of the Run/Stop input setting, the Run/Stop switch position or the last controller state before the download.

However, there are two considerations in this regard:

Online Change	An online change (partial download) initiated while the controller is in the RUNNING state returns the controller to the RUNNING state if successful and provided the Run/Stop input is configured and set to Run or Run/Stop switch is set to Run. Before using the **Login with online change** option, test the changes to your application program in a virtual or non-production environment and confirm that the controller and attached equipment assume their expected conditions in the RUNNING state. **⚠WARNING** **UNINTENDED EQUIPMENT OPERATION** Always verify that online changes to a RUNNING application program operate as expected before downloading them to controllers. **Failure to follow these instructions can result in death, serious injury, or equipment damage.** **NOTE:** Online changes to your program are not automatically written to the Boot application, and are overwritten by the existing Boot application at the next reboot. If you wish your changes to persist through a reboot, manually update the Boot application by selecting **Create boot application** in the online menu (the controller must be in the STOPPED state to achieve this operation).
Multiple Download	EcoStruxure Machine Expert has a feature that allows you to perform a full application download to multiple targets on your network or fieldbus. One of the default options when you select the **Multiple Download...** command is the **Start all applications after download or online change** option, which restarts all download targets in the RUNNING state, provided their respective Run/Stop inputs are commanding the RUNNING state, but irrespective of their last controller state before the multiple download was initiated. Deselect this option if you do not want all targeted controllers to restart in the RUNNING state. In addition, before using the **Multiple Download** option, test the changes to your application program in a virtual or non-production environment and confirm that the targeted controllers and attached equipment assume their expected conditions in the RUNNING state. **⚠WARNING** **UNINTENDED EQUIPMENT OPERATION** Always verify that your application program will operate as expected for all targeted controllers and equipment before issuing the “Multiple Download…” command with the “Start all applications after download or online change” option selected. **Failure to follow these instructions can result in death, serious injury, or equipment damage.** **NOTE:** During a multiple download, unlike a single download, EcoStruxure Machine Expert does not offer the option to create a Boot application. You can manually create a Boot application at any time by selecting **Create boot application** in the **Online menu** on all targeted controllers.

Note 8

The EcoStruxure Machine Expert software platform allows many powerful options for managing task execution and output conditions while the controller is in the

STOPPED or HALT states. Refer to Controller States Description, page 60 for further details.

Note 9

To exit the HALT state it is necessary to issue one of the Reset commands (Reset Warm, Reset Cold, Reset Origin), download an application or cycle power.

In case of non-recoverable event (hardware watchdog or internal error), a power cycle is mandatory.

Note 10

The RUNNING state has two exception conditions:

- RUNNING with External Error: this exception condition is indicated by the I/O LED, which displays solid red. You may exit this state by clearing the external error (probably changing the application configuration). No controller commands are required, but may however include the need of a power cycle of the controller. For more information, refer to I/O Configuration General Description, page 111.
- RUNNING with Breakpoint: this exception condition is indicated by the RUN LED, which displays a single green flash. Refer to Controller States Description, page 60 for further details.

Note 11

The boot application can be different from the application loaded. It can happen when the boot application was downloaded through SD card, FTP, or file transfer or when an online change was performed without creating the boot application.

Controller States Description

Introduction

This section provides a detailed description of the controller states.

⚠ WARNING

UNINTENDED EQUIPMENT OPERATION

- Never assume that your controller is in a certain controller state before commanding a change of state, configuring your controller options, uploading a program, or modifying the physical configuration of the controller and its connected equipment.
- Consider the effect of any of these operations on all connected equipment before performing any of these operations.
- Positively confirm the controller state by viewing its LEDs before acting on the controller.
- Confirm the condition of the Run/Stop input (if so equipped and configured) and/or the Run/Stop switch (if so equipped) before acting on the controller.
- Verify the presence of output forcing before acting on the controller.
- Review the controller status information via EcoStruxure Machine Expert before acting on the controller.(1)

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

(1) The controller states can be read in the PLC_R.i_wStatus system variable of the M262 System library (see Modicon M262 Logic/Motion Controller, System Functions and Variables, System Library Guide).

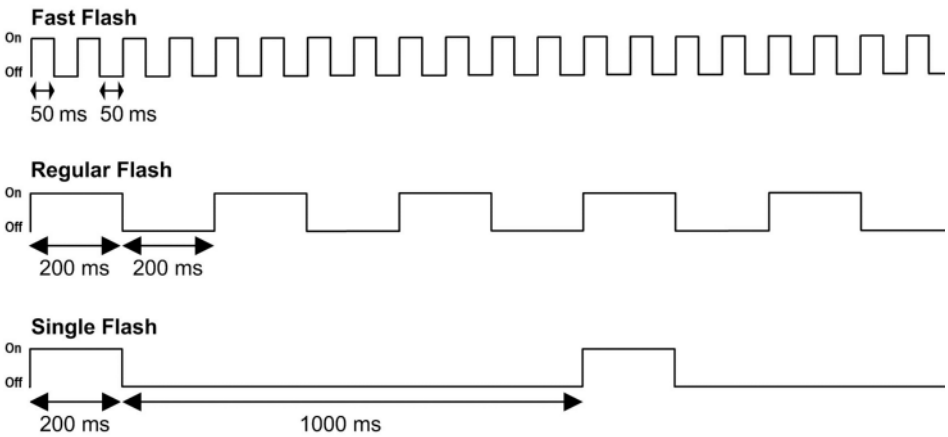
Controller States Table

The following tables describe the controller states:

Controller State	Description	LED Status
BOOTING	The controller executes the boot firmware and its own internal self-tests. It then verifies the checksum of the firmware and user applications.	Each LED, from the PWR LED to the NS or S3 LED, depending on the controller reference, flashes before turning solid green. The boot sequence is complete when all the LEDs are solid green. The LEDs then flash together briefly to indicate that the controller is operational.
INVALID_OS	There is not a valid firmware file present in the non-volatile memory or the firmware is not from Schneider Electric. The controller does not execute the application. Refer to the Updating Firmware, page 238 section to restore a correct state.	FSP LED stays solid red after the boot sequence.

Controller State	Description	LED		
		RUN (Green)	ERR (Red)	I/O (Red)
EMPTY	The controller has no application.	OFF	Single flash	OFF
EMPTY after a system error detected	This state is the same as the other EMPTY state. However, the application is present, and is intentionally not loaded. A reboot (power cycle), or a new application download, restores a correct state.	OFF	Fast flash	OFF
RUNNING	The controller is executing a valid application.	ON	OFF	OFF
RUNNING with breakpoint	This state is same as the RUNNING state with the following exceptions: • The task-processing portion of the program does not resume until the breakpoint is cleared. • The LED indications are different. For more information on breakpoint management, refer to the EcoStruxure Machine Expert Programming Guide.	Single flash	OFF	OFF
RUNNING with external error detected	The controller is executing a valid application and a configuration, TM3, SD card, or other I/O error is detected. When I/O LED is ON, the details about the detected error can be found in *PLC_R.i_lwSystemFault_1* and *PLC_R.i_lwSystemFault_2*. Any of the detected error conditions reported by these variables cause the I/O LED to be ON.	ON	OFF	ON
STOPPED	The controller has a valid application that is stopped. See details of the STOPPED state, page 62 for an explanation of the behavior of outputs and field buses in this state.	Regular flash	OFF	OFF
STOPPED with external error detected	The controller has a valid application that is stopped and a configuration, TM3, SD card, or other I/O error is detected.	Regular flash	OFF	ON
HALT	The controller stops executing the application because it has detected an application error.	Regular flash	ON	–
Boot Application not saved	The controller has an application in memory that differs from the application in non-volatile memory. At next power cycle, the application is changed by the one from non-volatile memory.	ON or regular flash	Single flash	OFF

This figure shows the difference between the fast flash, the regular flash and single flash:



Details of the STOPPED State

The following statements are true for the STOPPED state:

- The input configured as the Run/Stop input remains operational.
- The output configured as the Alarm output remains operational and goes to a value of 0.
- Ethernet, Serial (Modbus, ASCII, and so on), and USB communication services remain operational and commands written by these services can continue to affect the application, the controller state, and the memory variables.
- WebVisualisation is not operational.
- Outputs initially assume their configured default state (**Keep current values** or **Set all outputs to default**) or the state dictated by output forcing if used. The subsequent state of the outputs depends on the value of the **Update IO while in stop** setting and on commands received from remote devices. For more information on the behavior of the TM3 outputs, refer to Modicon TM3 Expansion Modules - Programming Guide.

Task and I/O Behavior When Update IO While In Stop Is Selected	When the **Update IO while in stop** setting is selected: • The Read Inputs operation continues normally. The physical inputs are read and then written to the `%I` input memory variables. • The Task Processing operation is not executed. • The Write Outputs operation continues. The `%Q` output memory variables are updated to reflect either the **Keep current values** configuration or the **Set all outputs to default** configuration, adjusted for any output forcing, and then written to the physical outputs.
CANopen Behavior When Update IO While In Stop Is Selected	The following is true for the CANopen buses when the **Update IO while in stop** setting is selected: • The CANopen bus remains operational. Devices on the CANopen bus continue to perceive the presence of a functional CANopen Master. • TPDO and RPDO continue to be exchanged. • The optional SDO, if configured, continue to be exchanged. • The Heartbeat and Node Guarding functions, if configured, continue to operate. • If the **Behaviour for outputs in Stop** field is set to **Keep current values**, the TPDOs continue to be issued with the last values. • If the **Behaviour for outputs in Stop** field is **Set all outputs to default** the last values are updated to the default values and subsequent TPDOs are issued with these default values.
Task and I/O Behavior When Update IO While In Stop Is Not Selected	When the **Update IO while in stop** setting is not selected, the controller sets the I/O to either the **Keep current values** or **Set all outputs to default** condition (as adjusted for output forcing if used). After this, the following becomes true: • The Read Inputs operation ceases. The `%I` input memory variables are frozen at their last values. • The Task Processing operation is not executed. • The Write Outputs operation ceases. The `%Q` output memory variables can be updated via the Ethernet, Serial, and USB connections. However, the physical outputs are unaffected and retain the state specified by the configuration options.
CANopen Behavior When Update IO While In Stop Is Not Selected	The following is true for the CANopen buses when the **Update IO while in stop** setting is not selected: • The CANopen Master ceases communications. Devices on the CANopen bus assume their configured fallback states. • TPDO and RPDO exchanges cease. • Optional SDO, if configured, exchanges cease. • The Heartbeat and Node Guarding functions, if configured, stop. • The current or default values, as appropriate, are written to the TPDOs and sent once before stopping the CANopen Master.

CANopen Behavior in HALT State

The following table describes the CANopen behavior in HALT state:

HALT state is caused by ProcessorLoadWatchdog	• The CANopen bus does not remain operational. • The CANopen Master ceases communications. Devices on the CANopen bus assume their configured fallback states. • TPDO and RPDO exchanges cease. • Optional SDO, if configured, exchanges cease. • The Heartbeat and Node Guarding functions, if configured, stop.
HALT state is caused by Task Watchdog **The task defined for the Bus cycle task of the CANopen_ Performance is in Exception**	
HALT state is caused by Task Watchdog **The task defined for the Bus cycle task of the CANopen_ Performance is NOT in Exception**	• The CANopen bus remains operational. Devices on the CANopen bus continue to detect the presence of a functional CANopen Master. • TPDO and RPDO continue to be exchanged. • The optional SDO, if configured, continue to be exchanged. • The Heartbeat and Node Guarding functions, if configured, continue to operate. • If the **Behaviour for outputs in Stop** field is set to **Keep current values**, the TPDOs continue to be issued with the last values. • If the **Behaviour for outputs in Stop** field is set to **Set all outputs to default**, the last values are updated to the default values and subsequent TPDOs are issued with these default values.

State Transitions and System Events

Overview

This section begins with an explanation of the output states possible for the controller. It then presents the system commands used to transition between controller states and the system events that can also affect these states. It concludes with an explanation of the Remanent variables, and the circumstances under which different variables and data types are retained through state transitions.

Controller States and Output Behavior

Introduction

The Modicon M262 Logic/Motion Controller defines output behavior in response to commands and system events in a way that allows for greater flexibility. An understanding of this behavior is necessary before discussing the commands and events that affect controller states.

The possible output behaviors and the controller states to which they apply are:

- Managed by **Application Program**
- **Keep current values**
- **Set all outputs to default**
- Hardware **Initialization Values**
- Software **Initialization Values**
- **Output Forcing**

  NOTE: For TM3 **Expert module** reflex output behavior, refer to Modicon TM3 Expansion Modules - Programming Guide.

Managed by Application Program

Your application program manages outputs normally. This applies in the RUNNING and RUNNING with External Error Detected states.

NOTE: An exception to this is if the RUNNING with External Error Detected state has been provoked by a I/O expansion bus error. For more information, refer to I/O Configuration General Description, page 111.

Keep Current Values

Select this option by choosing **Controller Editor > PLC settings > Behavior for outputs in Stop > Keep current values**. To access the **Controller Editor**, right-click on the controller in the **Devices tree** and select **Edit Object**.

This output behavior applies in the STOPPED controller state. It also applies to CAN bus in the HALT controller state, page 63, except if the task defined for the Bus cycle task of the CANopen_Performance is in Exception, or during ProcessorLoadWatchdog.

Outputs maintain their state, although the details of the output behavior vary greatly depending on the setting of the **Update I/O while in stop** option and the actions commanded via configured fieldbuses. Refer to **PLC Settings**, page 78 for more details on these variations.

Set All Outputs to Default

Select this option by choosing **Controller Editor > PLC settings > Behavior for outputs in Stop > Set all outputs to default**. To access the **Controller Editor**, right-click on the controller in the **Devices tree** and select **Edit Object**.

This output behavior applies:

- when the controller is going from RUNNING state to STOPPED state.
- if the controller is going from RUNNING state to HALT state.
- after application download.
- after reset warm/cold command.
- after a reboot.

It also applies to CAN bus in the HALT controller state. Outputs maintain their state, although the details of the output behavior vary greatly depending on the setting of the **Update I/O while in stop** option and the actions commanded via configured fieldbuses. Refer to Controller States Description, page 60 for more details on these variations.

Hardware Initialization Values

This output state applies in the BOOTING, EMPTY (following power cycle with no boot application or after the detection of a system error), and INVALID_OS states.

In the initialization state, analog, transistor, and relay outputs assume the following values:

- For an analog output: Z (high impedance)
- For a fast transistor output: Z (high impedance)
- For a regular transistor output: 0 Vdc
- For a relay output: Open

Software Initialization Values

This output state applies when downloading or when resetting the application. It applies at the end of the download or at the end of a reset warm or cold.

The software **Initialization Values** are the initialization values of outputs images (%I, %Q, or variables mapped on %I or %Q).

By default, they are set to 0 but it is possible to map the I/O in a GVL and assign to the outputs a value different than 0.

Output Forcing

The controller allows you to force the state of selected outputs to a defined value for the purposes of system testing, commissioning, and maintenance.

You are only able to force the value of an output while your controller is connected to EcoStruxure Machine Expert.

To do so, use the **Force values** command in the **Debug** menu.

Output forcing overrides other commands to an output irrespective of the task programming that is being executed.

When you logout of EcoStruxure Machine Expert when output forcing has been defined, you are presented with the option to retain output forcing settings. If you select this option, the output forcing continues to control the state of the selected outputs until you download an application or use one of the Reset commands.

When the option **Update I/O while in stop**, if supported by your controller, is checked (default state), the forced outputs keep the forcing value even when the controller is in STOPPED state.

Output Forcing Considerations

The output you wish to force must be contained in a task that is currently being executed by the controller. Forcing outputs in unexecuted tasks, or in tasks whose execution is delayed either by priorities or events has no effect on the output. However, once the task that had been delayed is executed, the forcing takes effect at that time.

Depending on task execution, the forcing could impact your application in ways that may not be obvious to you. For example, an event task could turn on an output. Later, you may attempt to turn off that output but the event is not being triggered at the time. This would have the effect of the forcing being apparently ignored. Further, at a later time, the event could trigger the task at which point the forcing would take effect.

In case of any forced variable, the FSP LED is flashing red, regular flash.

⚠ WARNING

UNINTENDED EQUIPMENT OPERATION

- You must have a thorough understanding of how forcing will affect the outputs relative to the tasks being executed.
- Do not attempt to force I/O that is contained in tasks that you are not certain will be executed in a timely manner, unless your intent is for the forcing to take affect at the next execution of the task whenever that may be.
- If you force an output and there is no apparent affect on the physical output, do not exit EcoStruxure Machine Expert without removing the forcing.

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

Commanding State Transitions

Run Command

Effect: Commands a transition to the RUNNING controller state.

Starting Conditions: BOOTING or STOPPED state.

Methods for Issuing a Run Command:

- Refer to Run/Stop Input, page 103 for more information.
- Software **Online** menu: Select the **Start** command.
- RUN command from Web server.
- By an external call via Modbus request using the PLC_W.q_wPLCControl and PLC_W.q_uiOpenPLCControl system variables of the M262 System library (see Modicon M262 Logic/Motion Controller, System Functions and Variables, System Library Guide).
- **Login with online change** option: An online change (partial download) initiated while the controller is in the RUNNING state returns the controller to the RUNNING state if successful.
- **Multiple Download** Command: sets the controllers into the RUNNING state if the **Start all applications after download or online change** option is selected, irrespective of whether the targeted controllers were initially in the RUNNING, STOPPED or EMPTY state.
- The controller is restarted into the RUNNING state automatically under certain conditions.

Refer to Controller State Diagram, page 56 for further details.

Stop Command

Effect: Commands a transition to the STOPPED controller state.

Starting Conditions: BOOTING, EMPTY, or RUNNING state.

Methods for Issuing a Stop Command:

- Run/Stop Input: If configured, command a value of 0 to the Run/Stop input. Refer to Run/Stop Input, page 103 for more information.
- Software **Online** menu: Select the **Stop** command.
- STOP command from Web server
- By an internal call by the application or an external call via Modbus request using the PLC_W. q_wPLCControl and PLC_W. q_uiOpenPLCControl system variables of the M262 System library (see Modicon M262 Logic/ Motion Controller, System Functions and Variables, System Library Guide).
- **Login with online change** option: An online change (partial download) initiated while the controller is in the STOPPED state returns the controller to the STOPPED state if successful.
- **Download** Command: implicitly sets the controller into the STOPPED state.
- **Multiple Download** Command: sets the controllers into the STOPPED state if the **Start all applications after download or online change** option is not selected, irrespective of whether the targeted controllers were initially in the RUNNING, STOPPED or EMPTY state.
- REBOOT by Script: The file transfer script on an SD card can issue a REBOOT as its final command. The controller is rebooted into the STOPPED state provided the other conditions of the boot sequence allow this to occur. Refer to Reboot, page 70 for further details.
- The controller is restarted into the STOPPED state automatically under certain conditions.

Refer to Controller State Diagram, page 56 for further details.

Reset Warm

Effect: Resets the variables, except for the remanent variables, to their default values. Places the controller into the STOPPED state.

Starting Conditions: RUNNING, STOPPED, or HALT states.

Methods for Issuing a Reset Warm Command:

- Software **Online** menu: Select the **Reset warm** command.
- By an internal call by the application or an external call via Modbus request using the PLC_W. q_wPLCControl and PLC_W. q_uiOpenPLCControl system variables of the M262 System library (see Modicon M262 Logic/ Motion Controller, System Functions and Variables, System Library Guide).

Effects of the Reset Warm Command:

1. The application stops.
2. Forcing is erased.
3. Diagnostic indications for errors are reset.
4. The values of the retain variables are maintained.
5. The values of the retain-persistent variables are maintained.
6. The non-located and non-remanent variables are reset to their initialization values.
7. The values of the 0...59999 `%MW` registers are reset to 0.
8. The fieldbus communications are stopped and then restarted after the reset is complete.
9. The inputs are reset to their initialization values. The outputs are reset to their software initialization values or their default values if no software initialization values are defined.
10. The Post Configuration file is read, page 229.

For details on variables, refer to Remanent Variables, page 73.

Reset Cold

Effect: Resets the variables, except for the retain-persistent type of remanent variables, to their initialization values. Places the controller into the STOPPED state.

Starting Conditions: RUNNING, STOPPED, or HALT states.

Methods for Issuing a Reset Cold Command:

- Software **Online** menu: Select the **Reset cold** command.
- By an internal call by the application or an external call via Modbus request using the PLC_W. q_wPLCControl and PLC_W. q_uiOpenPLCControl system variables of the M262 System library (see Modicon M262 Logic/ Motion Controller, System Functions and Variables, System Library Guide).

Effects of the Reset Cold Command:

1. The application stops.
2. Forcing is erased.
3. Diagnostic indications for errors are reset.
4. The values of the retain variables are reset to their initialization value.
5. The values of the retain-persistent variables are maintained.
6. The non-located and non-remanent variables are reset to their initialization values.
7. The values of `%MW0` to `%MW59999` registers are reset to 0.
8. The fieldbus communications are stopped and then restarted after the reset is complete.
9. The inputs are reset to their initialization values. The outputs are reset to their software initialization values or their default values if no software initialization values are defined.
10. The Post Configuration file is read, page 229.

For details on variables, refer to Remanent Variables, page 73.

Reset Origin

Effect: Resets all variables, including the remanent variables, to their initialization values. Erases all user files on the controller, including user rights and certificates. Reboots and places the controller into the EMPTY state.

Starting Conditions: RUNNING, STOPPED, or HALT states.

Methods for Issuing a Reset Origin Command:

- Software **Online** menu: Select the **Reset origin** command.

Effects of the Reset Origin Command:

1. The application stops.
2. Forcing is erased.
3. The WebVisualisation files are erased.
4. The user files (Boot application, Post Configuration, App, App/MFW, Cfg) are erased, except CodesysLateConf.cfg.
5. Diagnostic indications for errors are reset.
6. Nodename of the controller is reset to the default value.
7. The values of the retain variables are reset.
8. The values of the retain-persistent variables are reset.
9. The non-located and non-remanent variables are reset.
10. The fieldbus communications are stopped.
11. The other inputs are reset to their initialization values.

    The other outputs are reset to their hardware initialization values.

    Security certificates are erased.
12. The controller reboots.
13. FwLog.txt is maintained and all other System Log files are erased.

For details on variables, refer to Remanent Variables, page 73.

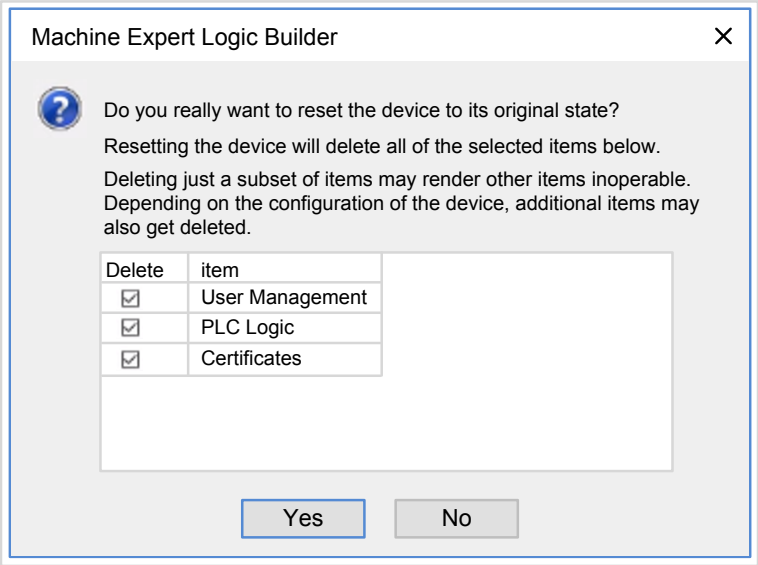
Reset Origin Device

Effect: Resets all variables, including the remanent variables, to their initialization values. Places the controller into the EMPTY state if **PLC Logic** is selected.

Starting Conditions: RUNNING, STOPPED, or HALT states.

Methods for Issuing a Reset Origin Device Command:

- In the software: Right-click **My controller > Reset Origin Device** command. **Result:** a dialog box allows you to select the items to remove:
  - **User Management**
  - **PLC Logic**
  - **Certificates**



When **User Management** is selected, user and groups are reset to default value.

NOTE: If the controller **User rights** are disabled before this command is used, you can connect to the controller without login prompt afterwards. Use the dedicated command in Online menu: **Security > Reset user rights management to default** to enforce again the use of user management.

When **PLC Logic** is selected:

1. The application stops.
2. Forcing is erased.
3. The WebVisualisation files are erased.
4. Diagnostic indications for errors are reset.
5. The values of the retain variables are reset.
6. The values of the retain-persistent variables are reset.
7. The non-located and non-remanent variables are reset.
8. The fieldbus communications are stopped.
9. Embedded Expert I/O are reset to their previous user-configured default values.
10. The other inputs are reset to their initialization values.

    The other outputs are reset to their hardware initialization values.
11. System Logs are maintained.

When **Certificates** is selected:

- Certificate used for encrypted communication is reset.
- Certificates used for Web server, FTP server and OPC UA server/client are not reset.

For details on variables, refer to Remanent Variables, page 73.

Reboot

Effect: Commands a reboot of the controller.

Starting Conditions: Any state.

Methods for Issuing the Reboot Command:

- Power cycle
- REBOOT by Script

Effects of the Reboot:

1. The state of the controller depends on a number of conditions:
   a. The controller state is RUNNING if:

      The Reboot was provoked by a power cycle and:

      - the **Starting Mode** is set to **Start in run**, and if the Run/Stop input is not configured, and if the controller was not in HALT state before the power cycle, and if the remanent variables are valid.

      - the **Starting Mode** is set to **Start in run**, and if the Run/Stop input is configured and set to RUN, and if the controller was not in HALT state before the power cycle, and if the remanent variables are valid.

      - the **Starting Mode** is set to **Start as previous state**, and Controller state was RUNNING before the power cycle, and if the Run/Stop input is not configured and the boot application has not changed and the remanent variables are valid.

      - the **Starting Mode** is set to **Start as previous state**, and Controller state was RUNNING before the power cycle, and if the Run/Stop input is configured and is set to RUN and the remanent variables are valid.

      The Reboot was provoked by a script and:

      - the **Starting Mode** is set to **Start in run**, and if the Run/Stop input is configured and set to RUN, or the switch is set to RUN, and if the controller was not in HALT state before the power cycle, and if the remanent variables are valid.

   b. The controller state is STOPPED if:

      The Reboot was provoked by a power cycle and:

      - the **Starting Mode** is set to **Start in stop**.

      - the **Starting Mode** is set to **Start as previous state** and the controller state was not RUNNING before the power cycle.

      - the **Starting Mode** is set to **Start as previous state** and the controller state was RUNNING before the power cycle, and if the Run/Stop input is not configured, and if the boot application has changed.

      - the **Starting Mode** is set to **Start as previous state** and the controller state was RUNNING before the power cycle, and if the Run/Stop input is not configured, and if the boot application has not changed, and if the remanent variables are not valid.

      - the **Starting Mode** is set to **Start as previous state** and the controller state was RUNNING before the power cycle, and if the Run/Stop input is configured and is set to STOP.

      - the **Starting Mode** is set to **Start in run** and if the controller state was HALT before the power cycle.

      - the **Starting Mode** is set to **Start in run**, and if the controller state was not HALT before the power cycle, and if the Run/Stop input is configured and is set to STOP.

      - the **Starting Mode** is set to **Start as previous state**, and if the Run/Stop input is configured and set to RUN, or the switch is set to RUN, and if the controller was not in HALT state before the power cycle.

      - the **Starting Mode** is set to **Start as previous state**, and if the Run/Stop input is configured and set to RUN, or the switch is set to RUN,HALT state before the power cycle.

c. The controller state is EMPTY if:

- There is no boot application or the boot application is invalid, or
- The reboot was provoked by specific System Errors.

d. The controller state is INVALID_OS if there is no valid firmware.

2. Forcing is maintained if the boot application is loaded successfully. If not, forcing is erased.
3. Diagnostic indications for errors are reset.
4. The values of the retain variables are restored if saved context is valid.
5. The values of the retain-persistent variables are restored if saved context is valid.
6. The non-located and non-remanent variables are reset to their initialization values.
7. The values of `%MW0` to `%MW59999` registers are reset to 0.
8. The fieldbus communications are stopped and restarted after the boot application is loaded successfully.
9. The inputs are reset to their initialization values. The outputs are reset to their hardware initialization values and then to their software initialization values or their default values if no software initialization values are defined.
10. The Post Configuration file is read, page 229.
11. The controller file system is initialized and its resources (sockets, file handles, and so on) are deallocated.

    The performance of the boot-up time of the controller depends on the number of files stored in its file system. Reducing their number as much as possible allows you to obtain better performance.

    The file system employed by the controller needs to be periodically re-established by a power cycle of the controller. If you do not perform regular maintenance of your machine, or if you are using an Uninterruptible Power Supply (UPS), you must force a power cycle (removal and reapplication of power) to the controller at least once a year.

NOTICE
DEGRADATION OF PERFORMANCE Reboot your controller at least once a year by removing and then reapplying power. **Failure to follow these instructions can result in equipment damage.**

For details on variables, refer to Remanent Variables, page 73.

NOTE: The Check context test concludes that the context is valid when the application and the remanent variables are the same as defined in the Boot application.

NOTE: If you provide power to the Run/Stop input from the same source as the controller, the loss of power to this input is detected immediately, and the controller behaves as if a STOP command was received. Therefore, if you provide power to the controller and the Run/Stop input from the same source, your controller reboots normally into the STOPPED state after a power interruption when **Starting Mode** is set to **Start as previous state**.

NOTE: If you make an online change to your application program while your controller is in the RUNNING or STOPPED state but do not manually update your Boot application, the controller detects a difference in context at the next reboot, the remanent variables are reset as per a Reset cold command, and the controller enters the STOPPED state.

Download Application

Effect: Loads your application executable into the RAM memory. Optionally, creates a Boot application in the non-volatile memory.

Starting Conditions: RUNNING, STOPPED, HALT, and EMPTY states.

Methods for Issuing the Download Application Command:

- In the software: 2 options exist for downloading a full application:
  - **Download** command.
  - **Multiple Download** command.

  For important information on the application download commands, refer to Controller State Diagram, page 56.
- FTP: Load Boot application file to the non-volatile memory using FTP. The updated file is applied at the next reboot.
- SD card: Load Boot application file using an SD card in the controller. The updated file is applied at the next reboot. Refer to File Transfer with SD Card for further details.

Effects of the Download Command:

1. The existing application stops and then is erased.
2. If valid, the new application is loaded and the controller assumes a STOPPED state.
3. Forcing is erased.
4. Diagnostic indications for errors are reset.
5. The values of the retain variables are reset to their initialization values.
6. The values of any existing retain-persistent variables are maintained.
7. The non-located and non-remanent variables are reset to their initialization values.
8. The values of `%MW0` to `%MW59999` registers are reset to 0.
9. The fieldbus communications are stopped and then the configured fieldbus of the new application is started after the download is complete.
10. Embedded Expert I/O are reset to their previous user-configured default values and then set to the new user-configured default values after the download is complete.
11. The inputs are reset to their initialization values. The outputs are reset to their hardware initialization values and then to their software initialization values or their default values if no software initialization values are defined, after the download is complete.
12. The Post Configuration file is read, page 229.

For details on variables, refer to Remanent Variables, page 73.

Effects of the FTP or SD Card Download Command:

There are no effects until the next reboot. At the next reboot, the effects are the same as a reboot with an invalid context. Refer to Reboot, page 70.

Error Detection, Types and Management

Error Management

The controller detects and manages three types of errors:

- External errors
- Application errors
- System errors

This table describes the types of errors that may be detected:

Type of Error Detected	Description	Resulting Controller State
External Error	External errors are detected by the system while RUNNING or STOPPED but do not affect the ongoing controller state. An external error is detected in the following cases: • A connected device reports an error to the controller. • The controller detects an error with an external device, for example, when the external device is communicating but not properly configured for use with the controller. • The controller detects an error with an output. • The controller detects a communication interruption with a device. • The controller is configured for an expansion module that is not present or not detected, and has not otherwise been declared as an optional module(1). • The boot application in non-volatile memory is not the same as the one in RAM. • The I/O LED is red ON.	RUNNING with External Error Detected Or STOPPED with External Error Detected
Application Error	An application error is detected when improper programming is encountered or when a task watchdog threshold is exceeded. The ERR LED is red ON.	HALT
System Error	A system error is detected when the controller enters a condition that cannot be managed during runtime. Most such conditions result from firmware or hardware exceptions, but there are some cases when incorrect programming can result in the detection of a system error, for example, when attempting to write to memory that was reserved during runtime, or when a system watchdog occurs. The ERR LED is fast flashing RED. **NOTE:** There are some system errors that can be managed by runtime and are therefore treated like application errors.	BOOTING → EMPTY
(1) Expansion modules may appear to be absent for any number of reasons, even if the absent I/O module is physically present on the bus. For more information, refer to I/O Configuration General Description, page 111.		

NOTE: Refer to the Modicon M262 Logic/Motion Controller, System Functions and Variables, System Library Guide for more detailed information on diagnostics.

Remanent Variables

Overview

Remanent data refers to variables that are defined in Programming Organization Units (POUs) as retain or retain-persistent. In the event of power outages, reboots, resets, and application program downloads, remanent variables can either be reinitialized or retain their values.

This table describes the behavior of remanent variables in each case:

Action	VAR	VAR RETAIN	VAR GLOBAL RETAIN PERSISTENT
Online change to application program	X	X	X
Online change modifying the boot application (1)	–	X	X
Stop	X	X	X
Power cycle	–	X	X
Reset warm	–	X (2)	X
Reset cold	–	–	X
Reset origin	–	–	–
Reset origin device	–	–	–
Download of application program using EcoStruxure Machine Expert (3)	–	–	X
Download of application program using an SD card (3)	–	–	–

X The value is maintained.

– The value is reinitialized.

(1) Retain variable values are maintained if an online change modifies only the code part of the boot application (for example, `a:=a+1; => a:=a+2;`). In all other cases, retain variables are reinitialized.

(2) For more details on VAR RETAIN, refer to Effects of the Reset warm Command, page 67.

(3) If the downloaded application contains the same retain-persistent variables as the existing application, the existing retain variables maintain their values.

Adding Retain-Persistent Variables

Declare retain-persistent (**VAR GLOBAL PERSISTENT RETAIN**) variables in the **PersistentVars** window:

Step	Action
1	In the **Applications tree**, select the **Application** node.
2	Click the right mouse button.
3	Choose **Add Objects** > **Persistent variables**
4	Click **Add**. **Result:** The **PersistentVars** window is displayed.

Retain and Persistent Variables: Performance Impact

Retain or retain-persistent variables are located in a dedicated non-volatile memory. Each time these variables are accessed during Programming Organization Unit (POU) execution, the non-volatile memory is accessed. The access time of these variables is slower than the access time of regular variables, which can impact performance. This is an important fact to take into account when writing performance-sensitive POUs.

For more information about the impact of retain and retain persistent variables on cycle time during POU execution, see Processing Performance, page 288.

Controller Device Editor

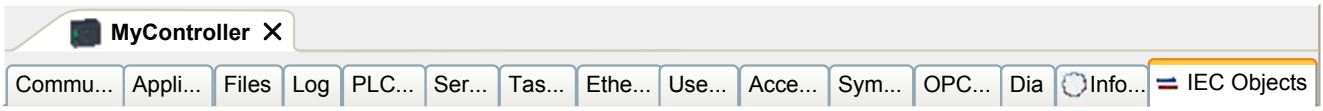
Introduction

This chapter describes how to configure the controller.

Controller Parameters

Controller Parameters

To open the device editor, double-click **MyController** in the **Devices tree**:



Tabs Description

Tab	Description	Restriction
Communication settings, page 77	Manages the connection between the PC and the controller: • Helping you find a controller in a network, • Presenting the list of available controllers, so you can connect to the selected controller and manage the application in the controller, • Helping you physically identify the controller from the device editor, • Helping you change the communication settings of the controller. The controller list is detected through NetManage or through the Active Path based on the communication settings. To access the **Communication settings**, click **Project > Project Settings...** in the menu bar. For more information, refer to the Communication Settings in the EcoStruxure Machine Expert Programming Guide.	–
Applications	Presents the application running on the controller and allows removing the application from the controller. If the state is **EMPTY**, boot application is deleted.	Online mode only
Files, page 40	File management between the PC and the controller. Only one controller disk at a time can be seen through this tab. This tab displays the content of the */usr* directory of the internal non-volatile memory of the controller.	Online mode only
Log	View the controller log file.	Online mode only
PLC settings, page 78	Configuration of: • Starting mode options • I/O behavior in stop • Bus cycle options	–
Services, page 79	Date and time settings, versions.	Online mode only
IEC Objects	Allows you to access to the device from the IEC application through the listed objects. Displays a monitoring view in online mode. For more information, refer to IEC Object in CoDeSys Online Help.	–
Task deployment	Displays a list of I/Os and their assignments to tasks.	After compilation only
Ethernet Services, page 80	**Ethernet_1** and **Ethernet_2** tabs summarize the Ethernet connections. The **IP Routing** tab allows you to configure the routes and the cross network transparency through **IP Forwarding** options. The **Ethernet Resources** tab allows you to calculate the number of connections and channels configured.	–
Users and Groups, page 86	The **Users and Groups** tab is provided for devices supporting online user management. It allows setting up users and access-rights groups and assigning them access rights to control the access on projects and devices in online mode. For more information, refer to the EcoStruxure Machine Expert Programming Guide.	–
Access Rights, page 86	The **Access Rights** tab allows you to define the device access rights of users. For more information, refer to the EcoStruxure Machine Expert Programming Guide.	–
Symbol Rights	Allows the **Administrator** to configure **Users and Groups** access to the symbol sets. For more information, refer to Symbol Configuration in CoDeSys Online Help.	–
OPC UA Server Configuration	Displays the OPC UA Server Configuration, page 216 window.	–
Diagnostic Table	Displays the data of the controller. The data can be accessed using the syntax `NameOfControllerInDeviceTree.NameofParameter`. Example: `MyController.SA_NbPowerOn`.	Online mode only
Information	Displays general information about the device (name, description, provider, version, image).	–

Communication Settings

Introduction

This tab allows you to manage the connection from the PC to the controller:

- Helping you find a controller in a network.
- Presenting the list of controllers, so you can connect to the selected controller and manage the application inside the controller.
- Helping you physically identify the controller from the device editor.
- Helping you change the communication settings of the controller.

You can change the display mode of the **Communication Settings** tab:

- **Simple mode**: refer to the EcoStruxure Machine Expert Programming Guide.
- **Classic mode**: refer to the EcoStruxure Machine Expert Programming Guide.
- **Controller selection mode**: refer to the EcoStruxure Machine Expert Programming Guide.

Editing Communication Settings

The **Edit communication settings** window lets you change the Ethernet communication settings. To do so, click **Communication settings** tab. The list of controllers available in the network appears. Select and right-click the required row and click **Edit communication settings...** in the contextual menu.

You can configure the Ethernet settings in the **Edit communication settings** window in two ways:

- Without the **Save settings permanently** option:

  Configure the communication parameters and click **OK**. These settings are immediately taken into account and are not kept if the controller is reset. For the next resets, the communication parameters configured into the application are taken into account.

- With the **Save settings permanently** option:

  You can also activate the **Save settings permanently** option before you click **OK**. Once this option is activated, the Ethernet parameters configured here are taken into account on reset instead of the Ethernet parameters configured into the application.

For more information on the **Communication settings** view of the device editor, refer to the EcoStruxure Machine Expert Programming Guide.

Editing the communication settings modifies the settings of the Ethernet interface used for the connection.

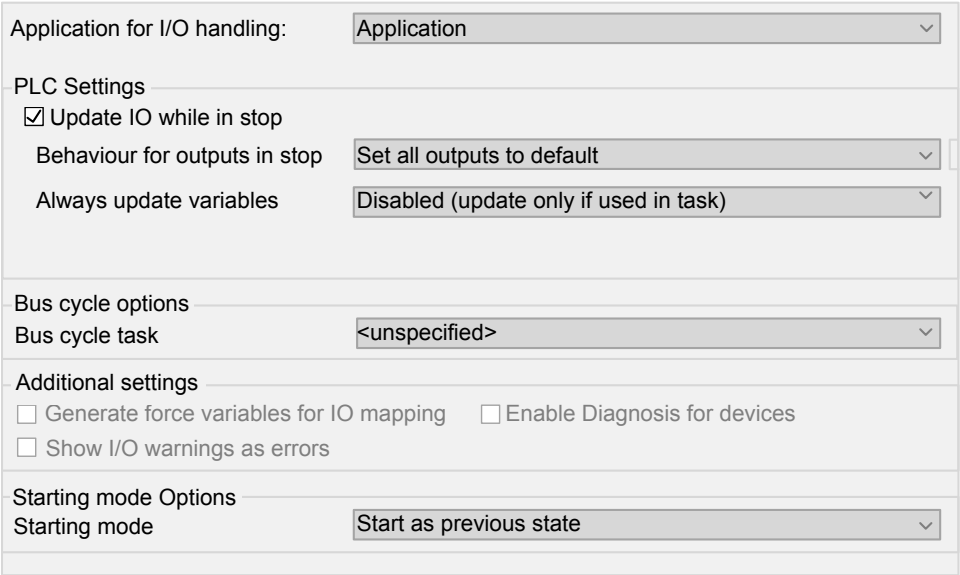
NOTE: If you are connected by USB, the Ethernet_2 settings are modified.

NOTE: Click the update icon to apply the changes.

PLC Settings

Overview

The figure below presents the **PLC Settings** tab:



Element		Description
Application for I/O handling		Select **Application** (as there is only one application in the controller). **NOTE:** If **None** is selected, the application will not be built.
PLC settings	**Update IO while in stop**	If this option is activated (default), the values of the input and output channels are also updated when the controller is stopped.
	Behavior for outputs in Stop	From the selection list, choose one of the following options to configure how the values at the output channels should be handled in case of controller stop: • **Keep current values** • **Set all outputs to default**
	Always update variables	From the selection list, choose one of the following options: • **Disabled (update only if used in task)** • **Enabled 1 (use bus cycle task if not used in any task)** • **Enabled 2 (always in bus cycle task)**
Bus cycle options	**Bus cycle task**	This configuration setting is the parent for all **Bus cycle task** parameters used in the application **Devices tree**. Some devices with cyclic calls, such as a **CANopen manager**, can be attached to a specific task. In the device, when this setting is set to **Use parent bus cycle setting**, the setting set for the controller is used. The selection list offers all tasks currently defined in the active application. The default setting is the MAST task. **NOTE: <unspecified>** means that the task is in "slowest cyclic task" mode.
Additional settings	**Generate force variables for IO mapping**	Not used.
	Enable Diagnosis for devices	Not used.
	Show I/O warnings as errors	Not used.
Starting mode Options	**Starting mode**	This option defines the starting mode on a power-on. For further information, refer to State behavior diagram, page 56. Select with this option one of these starting modes: • **Start as previous state** • **Start in stop** • **Start in run**

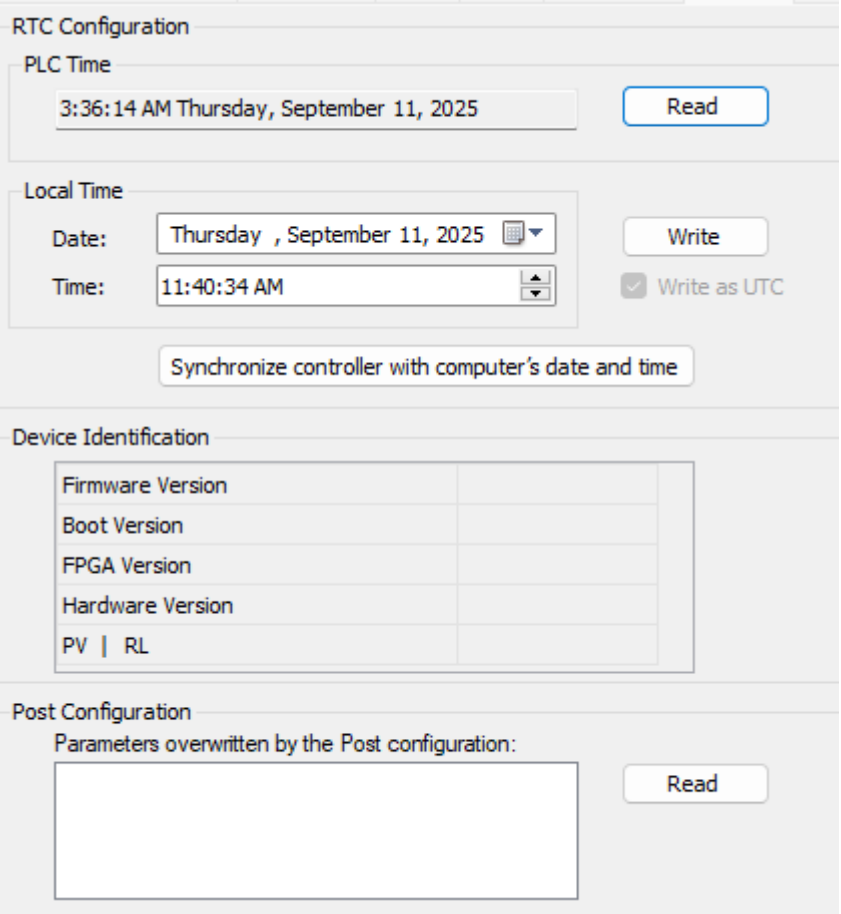
Services

Services Tab

The **Services** tab is divided in three parts:

- RTC Configuration
- Device Identification
- Post Configuration

The figure below shows the **Services** tab:



NOTE: To have controller information, you must be connected to the controller.

NOTE: RTC information can be configured by Web server or using the **SysTimeRtcSet** function block. For more information, refer to the Modicon M262 Logic/Motion Controller, System Functions and Variables, System Library Guide.

Element		Description
RTC Configuration	**PLC Time**	Displays the date and time read from the controller when you click the **Read** button. This read-only field is initially empty. **PLC Time** is returned in controller local time. The timezone of the controller can be found with the Web server.
	Read	Reads the date and time from the controller and displays them in the **PLC Time** field without any conversion.
	Local Time	Defines a date and time that are sent to the controller when you click the **Write** button. If necessary, modify the default values before clicking the **Write** button. A message box informs you about the result of the command. The date and time fields are initially filled with the computer date and time.
	Write	Writes to the controller the date and time of the **Local time** fields. The values are converted to UTC format before being written.
	Synchronize controller with computer's date/time	Writes to the controller the date and time of the computer. The values are converted to UTC format before being written.
Device Identification	**Firmware Version**	Displays the **Firmware Version** of the selected controller, if connected.
	Boot Version	Displays the **Boot Version** of the selected controller, if connected.
	FPGA Version	Displays the **FPGA Version** of the selected controller, if connected.
	Hardware Version	Displays the **Hardware Version** of the selected controller, if connected.
	PV | RL	Displays the **Product Version** (PV) and the **Release Level** (RL) of the selected controller, if connected.
Post Configuration		Displays the application parameters overwritten by the Post configuration, page 229.

Ethernet Services

Presentation

This tab displays the list of Ethernet or Sercos devices which are configured to be controlled by Modicon M262 Logic/Motion Controller.

- **Ethernet_1**
- **Ethernet_2**
- **Ethernet Resources**
- **IP Routing**
- **NTP**

Ethernet_1 and Ethernet_2 Toolbar

The following table describes the toolbar:

Element	Description
Generate IP address	Allows you to generate the configurations of each device configured in the **Devices tree**.
Filter Options	Allows you to display more information on the configured devices.
Discover devices	Start the Machine Assistant which allows you to discover and to configure the devices.

Network Settings

To view the configuration of a device, click the tab above the toolbar. The following information displays:

- **IP Address**
- **Subnet Mask**
- **Gateway**
- **Subnet Address**

Configured Devices in the Project

Element	Description	Restriction
Device Name	Name of the device from the **Devices tree**. Click the device name to access the device configuration.	Cannot be edited.
Device Type	Type of the device.	Cannot be edited.
IP Address	IP Address of the device. Can be left blank for Sercos devices.	–
MAC Address	MAC address of the target device. Can be left blank for Sercos devices.	Can be edited if **IP Address by BOOTP** selected in the configuration of the device.
DHCP Device Name	Hostname of the target device	Can be edited if **IP Address by DHCP** selected in the configuration of the device.
Subnet Mask	Subnet mask of the device	Visible if **Expert Mode** selected in **Filter Options**.
Gateway Address	Gateway address of the device	Visible if **Expert Mode** selected in **Filter Options**.
Identified by	Four identification modes are possible: • **None** • **Fixed** • **BOOTP** • **DHCP**	–
Protocol	Protocol used	Cannot be edited.
Identifier	Identifier of the device	Can be edited for Sercos device.
Identification Mode	Identification mode of the device	Can be edited for Sercos device.
Operating Mode	Three operating modes are possible: • **Activated** • **Simulated** • **Optional**	Can be edited for Sercos device.

Ethernet Resources

The **Ethernet Resources** subtab:

- Displays the number of configured connections and channels.
- Displays the number of input words.
- Displays the number of output words.
- Displays the scanner load.

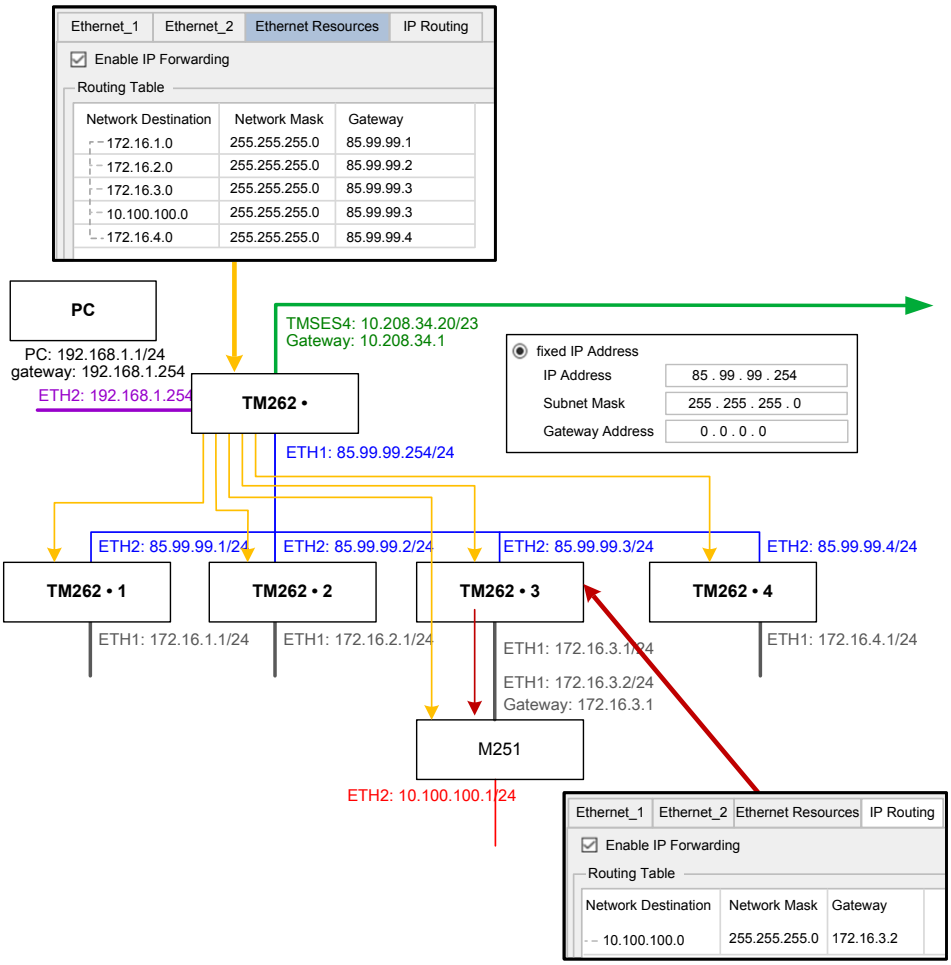
IP Routing

The **IP Routing** subtab allows you to configure the IP routes in the controller. The Modicon M262 Logic/Motion Controller can support up to 30 IP routes.

The parameter **Enable IP forwarding** allows you to deactivate the IP forwarding service of the controller. When deactivated, the communication is not routed from a network to another one. The devices on the device network are no longer accessible from the control network and related features like Web pages access on device or commissioning of device via DTM, EcoStruxure Machine Expert - Safety and so on are not possible.

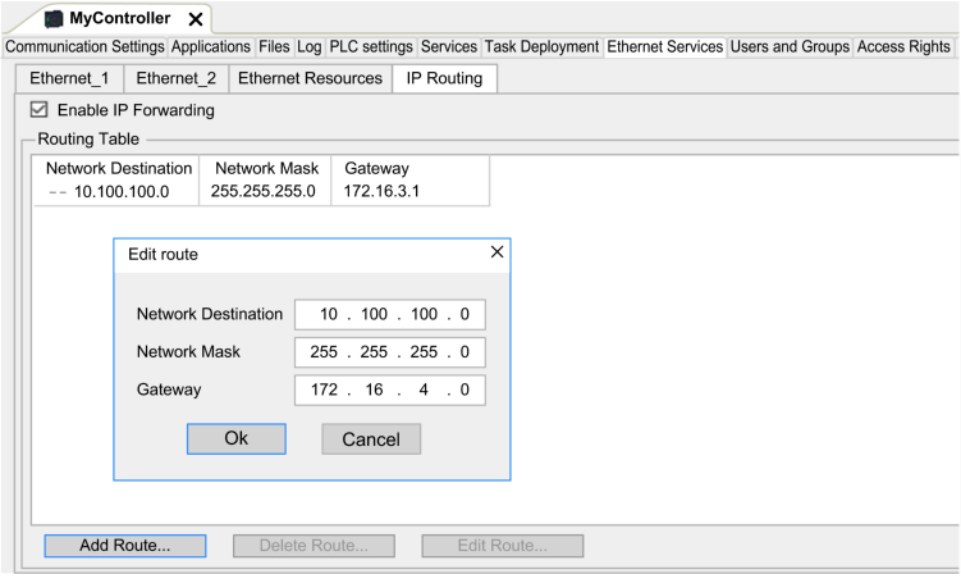
The Modicon M262 Logic/Motion Controller can have up to three Ethernet interfaces. Using a routing table is necessary to communicate with remote networks connected to different Ethernet interfaces. The gateway is the IP address used to connect to the remote network, which needs to be in local network of the controller.

This graphic depicts an example network, in which the last two rows of devices (gray and red) need to be added in the routing table:



Use the routing tables to manage the IP forwarding.

To add a route, double click **My controller** then click **Ethernet Services > IP Routing > Add Route**.



For reasons of network security, TCP/IP forwarding is disabled by default. Therefore, you must manually enable TCP/IP forwarding if you want to access devices through the controller. However, doing so may expose your network to possible cyberattacks if you do not take additional measures to protect your enterprise. In addition, you may be subject to laws and regulations concerning cybersecurity.

⚠ WARNING

UNAUTHENTICATED ACCESS AND SUBSEQUENT NETWORK INTRUSION

- Observe and respect any an all pertinent national, regional and local cybersecurity and/or personal data laws and regulations when enabling TCP/IP forwarding on an industrial network.
- Isolate your industrial network from other networks inside your company.
- Protect any network against unintended access by using firewalls, VPN, or other, proven security measures.

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

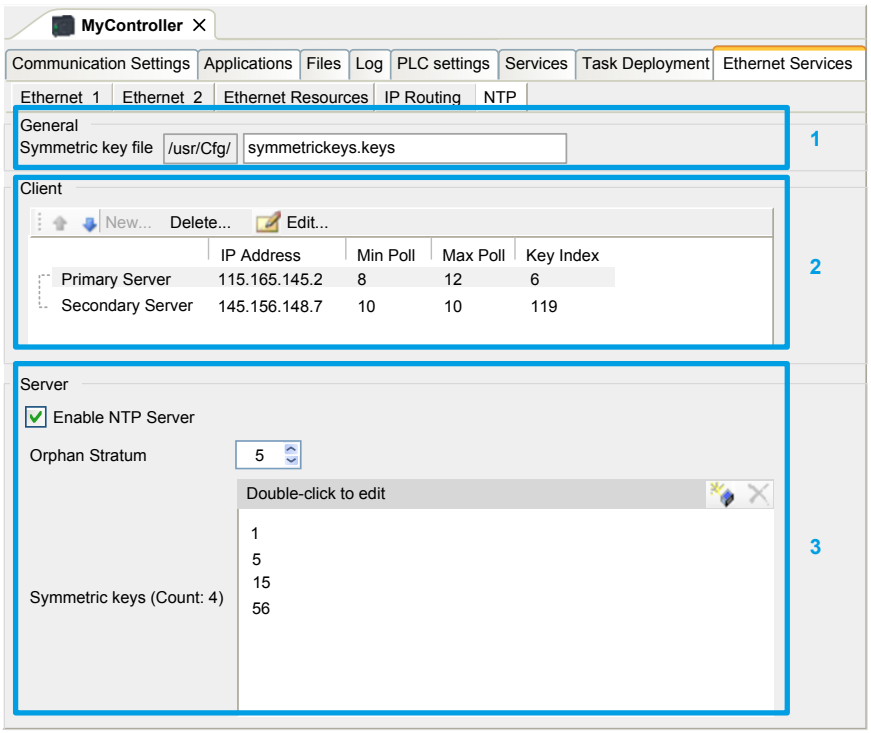
NTP

The **NTP** Protocol synchronizes the clock of device and resists the effects of variable latency (jitter).

The **NTP** subtab is divided in three parts:

- **General** (1)
- **Client** (2)
- **Server** (3)

The figure below shows the **NTP** subtab:



General section

Element	Description
/usr/Cfg*	Folder to which the trusted key file is to be uploaded. Not editable.
Empty*	File name of the **Symmetric keys file**. Editable. Can be left empty if no key index is defined. • Maximum length: 22 characters • File extension: .keys • Allowed characters: a...z, A...Z, 0...9, -, _ **NOTE:** You must enter a valid file name or leave the field empty. **NOTE:** The only authentication method for the key algorithm is MD5 for NTP.

Client section

You can define a maximum of two servers: **Primary Server** and **Secondary Server**. You must specify the following information for each server defined:

Element	Description	Value	Constraint
IP Address	The server IP Address.	Default value: 0.0.0.0	• The address must be used by another server • First byte must be between 1 and 223 • Loopback address is forbidden
Min Poll	The minimum poll value.	Default value: 6 Value range: 3...17 (1)	Minimum poll value must be inferior to maximum poll value.
Max Poll	The maximum poll value.	Default value: 10 Value range: 3...17 (1)	Maximum poll value must be superior to minimum poll value.
Key Index	The key index value.	Default value: 0 Value range: 0...65535	0 means “no key index”.
(1): 3 corresponds to 8 seconds (2^3), 17 corresponds to 131072 seconds (2^{17}).			

Server section

Element	Description	Value	Constraint
Enable NTP Server	Allows you to enable/disable the NTP Server.	Checked/unchecked	You must define stratum for orphan mode or NTP **Client Primary Server** if **NTP Server** is enabled.
Orphan Stratum	The orphan stratum level.	Default value: 0 Value range: 0...15	0 means: no **Orphan Stratum**. See **Orphan Stratum**, page 85.
Symmetric Keys	The list of key indexes.	Value range: 1...65535	Maximum of 32 key indexes, including **Primary Server** and **Secondary Server** key indexes.

NOTE: If you are using the default NTPv3 server of Microsoft Windows, the following configuration should be done on the server: Configuring Systems for High Accuracy.

Orphan Stratum

NTP uses a hierarchical system where each level is called a stratum. These levels are assigned a number starting at 0 for the reference at the top level.

When the controller is both client and server, the stratum is calculated automatically from the NTP server it is connected to. When the **Orphan Stratum** is 0, if the NTP server used by the controller becomes unreachable, the controller indicates to its NTP client that its clock is not synchronized. Otherwise, the value selected is used.

If the controller is only configured as NTP server, it will use the selected value in **Orphan Stratum**. You should select an appropriate stratum value according to the NTP hierarchy of your architecture.

NTP Keys File Syntax Usage

- NTP keys file only supports MD5 hash algorithm.
- The keys file must not have a header.
- No spaces allowed at the beginning line of a key.
- If you insert a comment at the end of a key line, you must add two spaces between the end of the key and the beginning of the comment.

Key file syntax:

```
<key_id> MD5 <password>  #<comment>
```

Examples:

```
1 MD5 key_1MD5  # MD5 hash algorithm
14 MD5 serverPassword  # MD5 hash algorithm
25 MD5 test25Keys  # MD5 key comment
```

NTP Diagnostics

The NTP feature for the Modicon M262 Logic/Motion Controller incorporates diagnostic events, which are represented by three bits within the *PLC_GVL.PLC_R.i_lwSystemFault_1* system variable:

- Bit 23 (NTP Error Detected): This bit is set to zero upon detecting an error during the initialization of the controller as an NTP server or client.
- Bit 25 (NTP Primary Server Error Detected): Indicates the availability status of the primary NTP server for time synchronization. It is set to zero when the primary server is unreachable.
- Bit 26 (NTP Secondary Server Error Detected): Indicates the availability status of the secondary NTP server for time synchronization. It is set to zero when the secondary server is unreachable.

These error indications can be viewed in the Web server under the Diagnostics: Controller Submenu, page 138, **Status > System Fault 1**. The message logger can provide more information on the detected error: refer to Modicon M262 Logic/Motion Controller, System Functions and Variables, System Library Guide.

NOTE: Delays in updating the error bits 25 and 26 may occur under the following conditions:

- Transition from 1 (Server Reachable) to 0 (Server Unreachable) is subject to the minimum and maximum poll settings. Typically, this transition takes an average time ranging from 8 times the minimum poll duration to 8 times the maximum poll duration.
- Transition from 0 (Server Unreachable) to 1 (Server Reachable) occurs, on average, within 5 to 10 minutes when the server is being initialized, for example, during server reboot or initial setup.

Users Rights

Introduction

Users rights contain the following elements: **User**, **Group**, **Object**, **Operation**, **User Rights**, **Access rights**. These elements allow you to manage users accounts and users access rights to control the access on the global projects.

- A **User** is a person or a service with specific **User Rights**.
- A **Group** is a **Persona** or a **Function**. It is predefined or added. Each **Group** provides accesses thanks to **Object**.
- An **Object** is composed by predefined accesses thanks to **Operation**.
- An **Operation** is the elementary action possible.
- **User Rights** are the possible **Access rights**: **VIEW**, **MODIFY**, **EXECUTE** and **ADD_REMOVE** for the dedicated operation.

For more information, refer to the EcoStruxure Machine Expert Programming Guide.

Login and Passwords

Login and password are not set by default. This table describes how to log in:

Server/feature	First connection or connection after reset to default / reset origin / reset origin device	User Rights enabled	Connection after User Rights disabled
EcoStruxure Machine Expert	You must first create your login and your password. **NOTE:** The login and the password that you create during the first connection have administrator privileges. **NOTE:** For information on lost login and passwords, see Troubleshooting, page 92.	**Login**: configured login **Password**: configured password	No login or password required.
Web server	No login possible	**Login**: configured login **Password**: configured password	**Login**: Anonymous **Password**: Anonymous or no password.
FTP server	No login possible	**Login**: configured login **Password**: configured password	**Login**: Anonymous **Password**: Anonymous
OPC UA	No login possible	**Login**: configured login **Password**: configured password	**Login**: Anonymous **Password**: Anonymous
Change Device Name feature	No login possible	**Login**: configured login **Password**: configured password	No login or password required.

⚠ WARNING

UNAUTHORIZED DATA AND/OR APPLICATION ACCESS

- Secure access to the FTP/Web/OPC-UA server(s) using User Rights.
- If you disable User Rights, disable the server(s) to prevent any unwanted or unauthorized access to your application and/or data.

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

NOTE: Anonymous login can be restored by disabling the user rights in **User Management** page of the Web server, page 144.

NOTE: The following characters are supported by the controller:

- login: **a...z A...Z 0...9 – = [ ] \ ; ‘ , . / @ # $ % ^ & * ( ) _ + { } | : “ < > ? ` ~**
- password: **a...z A...Z 0...9 – = [ ] \ ; ‘ , . / @ # $ % ^ & * ( ) _ + { } | : “ < > ? ` ~** and **space**

The length is limited to 60 characters.

Default User Management Settings

By default, the following settings are enabled:

- User management password policy
- User login lock
- Verification of password expiration

These default settings only take effect when no related settings are stored in section CmpUserMgr of CodesysLateConf.cfg file.

Due to password expiration policy, you are prompted to re-enter a different password when there are changes made to the **PLC Time**, page 79 that result in password expiration.

For more information, refer to Change Runtime Password Policy Dialog Box found in the EcoStruxure Machine Expert Programming Guide.

User

The **User** must be defined by the **Administrator** and must be linked with one or several groups.

Default Groups

This table indicates the name and description of the predefined default groups:

Group Name	Group Description
Administrator	• Is created at first connection. • Manages the user rights. • Can define a new **Group**. **NOTE:** The access rights of an **Administrator** cannot be disabled, regardless of explicit allow (+) or deny (-) permissions shown in the Access Rights view (refer to the EcoStruxure Machine Expert Programming Guide).
Persona	
Persona Programmer-Designer	Group dedicated to the design of the application.
Persona Operator	Group dedicated to the usage of the application.
Persona Web Designer	Group dedicated to the management of the Web server.
Persona Communication	Group dedicated to the management of communication features.
Persona Maintenance	Group dedicated to the maintenance of the application.
Function	
Function External Media	Group to allow the usage of External Command (from SD Card).
Function File Access	Group to allow permissions on files tab.
Function FTP	Group to allow usage of FTP.
Function Symbol Configuration	Group to allow access to **Symbol Configuration**.
Function Web Access	Group to allow command on Web server.
Function Monitor	Group to allow monitoring of IEC variables.
Function OPC UA	Group to allow access to OPC UA server.
Function Variable	Group to allow read/write of IEC variables.

Object Names

This table indicates the name and description of the predefined objects:

Object name	Object Description
Device	Object related to the connection of the controller through the software.
ExternalCmd	Object related to script command.
FrmUpdate	Object related to the commands **Update Boot**, **Clone** and **CloneCheck**.
FTP	Object related to FTP access (connection, upload and download on FTP server).
Logger	Object related to the message logger.
OPC_UA	Object related to OPC UA server (connection, read and write variables).
PlcLogic	Object related to the application on the controller.
Settings	Object related to the settings of the controller (nodename…).
UserManagement	Object related to User rights Management.
Web	Object related to the access of the Web server.
FileSystem	Object related to the file access (when accessing through the controller Files tab).

Operation Functions

This list indicates the name of the possible predefined operations:

- **SD card command**
  - Script Command: Reboot
  - Script Command: SET_NODE_NAME
  - Script Command: FIREWALL_INSTALL
  - Script Command: Delete
  - Script Command: Download
  - Script Command: Upload
  - Script Command: UpdateBoot
  - Script Command: CloneCheck (update controller Users Rights from SD card)
  - Clone operation (clone controller contents to empty SD card)
- **FTP server command**
  - Connection to FTP server
  - List Directory
  - Change Directory
  - Create Folder
  - Rename Folder
  - Suppress Folder
  - Create File
  - Rename File
  - Suppress File
  - Download File
  - Upload File

- **OPC UA server command**
  - Connection to OPC UA server
  - Read Variable
  - Write Variable
- **Web server command**
  - Connection to Web server
  - List Variables
  - Read Variable
  - Write Variable
  - Set Time
  - Access to File System
  - Download File
  - Access to Logger
  - Change Password
  - Reject/Trust Certificate (also needs device.settings User Rights Modify)
- **Software command**
  - Reset Origin Device
  - Login
  - Set Node Name
  - Update Logger
  - Create Application
  - Download Application
  - Pass RUN / STOP
  - Reset (Cold / Warm / Origin)
  - Delete Application
  - Create Boot Application
  - Save Retain Variables
  - Restore Retain Variables
  - Add Group
  - Remove Group
  - Add User
  - Remove User
  - Read User Rights
  - Import User Rights
  - Export User Rights

Access Rights

For each **Group** linked with an **Object**, **User Rights** are predefined with specifics **Access Rights**.

This table indicates the **Access Rights**:

Access Rights	Description (1)
VIEW	Allow to read only parameters and applications.
MODIFY	Allow to write, modify and download parameters and applications.
ADD_REMOVE	Allow to add and remove files, scripts and folders.
EXECUTE	Allow to execute and start applications and scripts.
(1) The **Access Rights** description depends on the **Object**. Refer to Predefined Access Rights Needed by Object and Associated Operations, page 95.	

Where to Use Access Rights

EcoStruxure Machine Expert and other third party tools such as a supported web browser can be used to verify access rights.

In EcoStruxure Machine Expert, there are different access points to verify functionalities. You need to enter your username and password to access most functionalities:

Functionality	Software Access Point or Test Tool	Password Needed
ExternalCmd	SD Card script command	No
FrmUpdate		
FTP	FTP client (such as FileZilla)	Yes
Logger	Software: Log tab update button, Web Page	
OPC_UA	OPC UA client (such as UaExpert)	
PlcLogic	Software: Application Login button	
Settings	Software: Device Security Settings button	
UserManagement	Software: User Group/Access Right tab update button	
Web	Web Browser	
FileSystem	Software: Files tab	
FW Update	Controller Assistant	

Import / Export User Rights

Import User Rights and Export User Rights commands allow you to import/export Device user management (*.dum2) files found in the EcoStruxure Machine Expert Programming Guide.

NOTE: The importation of user rights into an M262 Logic/Motion Controller is only possible if these user rights have been exported from an M262 Logic/Motion Controller.

Symbol Rights

The Symbol Rights tab (see Tabs Description, page 76) allows you to configure user group access to the symbol sets. It consists in a customizable set of symbols allowing to separate functions and associate them with a user right. If supported

by the target device, you can combine different symbol sets from the symbols of the application in the symbol configuration editor. The information about the symbol sets is downloaded to the controller. Then you can define the user group that has access to each symbol set.

Troubleshooting

The only way to gain access to a controller that has user access-rights enabled and for which you do not have the password(s) is by performing an Update Firmware operation. This clearing of User Rights can only be accomplished by using a SD card to update the controller firmware, page 238. In addition, you may clear the User Rights in the controller by running a script (refer to Reset the User Rights to Default, page 249). This effectively removes the existing application from the controller memory, but restores the ability to access the controller.

Predefined Access Rights for Group Persona

For each **Group**, several **Objects** are predefined with preset **Access Rights**:

Group: Administrator	
Object name	**Access Rights**
Device	**VIEW / MODIFY / ADD_REMOVE / EXECUTE**
FTP	**VIEW / MODIFY / ADD_REMOVE**
Logger	**VIEW**
OPC_UA	**VIEW / MODIFY**
PlcLogic	**VIEW / MODIFY / ADD_REMOVE / EXECUTE**
Settings	**VIEW / MODIFY**
UserManagement	**VIEW / MODIFY**
Web	**VIEW / MODIFY / EXECUTE**
FileSystem	**VIEW / MODIFY / ADD_REMOVE**

Group: Programmer-Designer persona	
Object name	**Access Rights**
Device	**VIEW / ADD_REMOVE**
FTP	**VIEW / MODIFY / ADD_REMOVE**
Logger	**VIEW**
OPC_UA	**VIEW / MODIFY**
PlcLogic	**VIEW / MODIFY / ADD_REMOVE / EXECUTE**
Settings	**VIEW / MODIFY**
UserManagement	**VIEW**
Web	**VIEW / MODIFY / EXECUTE**
FileSystem	**VIEW / MODIFY / ADD_REMOVE**

Group: Operator persona	
Object name	**Access Rights**
Device	**VIEW**
Logger	**VIEW**
PlcLogic	**VIEW / MODIFY / EXECUTE**
Settings	**VIEW**
UserManagement	**VIEW**
Web	**VIEW / MODIFY / EXECUTE**

Group: Web Designer persona	
Object name	**Access Rights**
Device	**VIEW**
FTP	**VIEW / MODIFY / ADD_REMOVE**
Logger	**VIEW**
OPC_UA	**VIEW**
PlcLogic	**VIEW**
Settings	**VIEW**
UserManagement	**VIEW**
Web	**VIEW / MODIFY / EXECUTE**
FileSystem	**VIEW / MODIFY / ADD_REMOVE**

Group: Communication persona	
Object name	**Access Rights**
Device	**VIEW**
FTP	**VIEW / MODIFY / ADD_REMOVE**
Logger	**VIEW**
OPC_UA	**VIEW / MODIFY**
PlcLogic	**VIEW / MODIFY / EXECUTE**
Settings	**VIEW**
UserManagement	**VIEW**
Web	**VIEW / MODIFY / EXECUTE**
FileSystem	**VIEW / MODIFY / ADD_REMOVE**

Group: Maintenance persona	
Object name	**Access Rights**
Device	**VIEW**
FTP	**VIEW / MODIFY / ADD_REMOVE**
Logger	**VIEW**
OPC_UA	**VIEW**
PlcLogic	**VIEW / EXECUTE**
Settings	**VIEW**
UserManagement	**VIEW**
Web	**VIEW / MODIFY / EXECUTE**
FileSystem	**VIEW / MODIFY / ADD_REMOVE**

Predefined Access Rights for Group Function

For each **Group**, several **Objects** are predefined with predefined **Access Rights**:

Group: Function External Media (1)	
Object name	**Access Rights**
ExternalCmd	**VIEW** / **MODIFY** / **ADD_REMOVE** / **EXECUTE**
FrmUpdate	**VIEW** / **MODIFY** / **ADD_REMOVE** / **EXECUTE**
(1) Enabling the objects in the group External Media will allow the access rights regardless of the user. That is to say, that the rights governing SD cards are global and are not confined to defined users.	

Group: Function File Access	
Object name	**Access Rights**
Logger	**VIEW**
FileSystem	**VIEW** / **MODIFY** / **ADD_REMOVE**

Group: Function FTP Access	
Object name	**Access Rights**
FTP	**VIEW** / **MODIFY** / **ADD_REMOVE**
Logger	**VIEW**

Group: Function Symbol Configuration Access	
Object name	**Access Rights**
Logger	**VIEW**
OPC_UA	**VIEW** / **MODIFY**
PlcLogic	**VIEW** / **MODIFY** / **ADD_REMOVE** / **EXECUTE**
Web	**VIEW** / **MODIFY** / **EXECUTE**

Group: Function Web Access	
Object name	**Access Rights**
Logger	**VIEW**
Web	**VIEW** / **MODIFY** / **EXECUTE**

Group: Function Monitor Access	
Object name	**Access Rights**
Logger	**VIEW**
OPC_UA	**VIEW**
PlcLogic	**VIEW**
Web	**VIEW**

Group: Function OPC UA Access	
Object name	**Access Rights**
Logger	**VIEW**
OPC_UA	**VIEW** / **MODIFY**

Group: Function Variable Access	
Object name	**Access Rights**
Logger	**VIEW**
OPC_UA	**VIEW**
PlcLogic	**VIEW / MODIFY / ADD_REMOVE / EXECUTE**
Web	**VIEW**

Predefined Access Rights Needed by Object and Associated Operations

VIEW access right is used to:

- Log into the controller with EcoStruxure Machine Expert.
- Connect to OPC UA, FTP server and Web server.
- Access User Management page with EcoStruxure Machine Expert.

Object Name	Access Rights			
	ADD_REMOVE	**MODIFY**	**VIEW**	**EXECUTE**
Device	Reset origin device	Set node name	Login	–
ExternalCmd	Delete	Download	Upload	Reboot Set Node Name Firewall install Clone Check
FrmUpdate	Updateboot	–	Clone	Clone Check
FTP	–	Create Folder Rename Folder Suppress Folder Create File Rename File Suppress File Upload File	Connection to FTP Server List Directory Change Directory Download File	–
Logger	–	–	Update logger	–
OPC_UA	–	Connection OPC_UA Read Variable Write Variable	Connection OPC_UA Read Variable	–
PlcLogic	Create application Download application Delete application Create Boot application	Write Variable Restore Retains Var	Read Variable Save retain variables	Pass Run / Stop Reset
Settings	–	Reject / Trust Certificate Set Node Name	–	–
UserManagement	–	Add Group Remove Group Add User Remove User Edit User Rights Import User Rights Reset Origin Device	Read User Rights Export User Rights	–
Web	–	Set Variables Set Time Download File Change Password	Connection to Web server Monitor Variables Access Files System Change Password	Execute Command
FileSystem	–	–	–	–

Symbol Configuration Editor

Overview

The symbol configuration functionality allows you to configure external access to variables. The symbols and the variables can then be monitored in the Web server and be accessed by external applications, such as Vijeo-Designer or OPC server.

To configure symbols for an application, double-click the **Symbol Configuration** node in the **Tools tree**. The **Symbol configuration** editor view opens.

The editor contains a table. Depending on the set filter, it shows the available variables, or just those already selected for the symbol configuration. For this purpose, the concerned info pop-ups with POUs definition or libraries are listed in the **Symbols** column. You can expand them in order to show the corresponding variables.

NOTE: The number of variables you can configure is not limited.

The restrictions depend on the different monitor platforms:

Platform	Maximum Limit
Web server	16 000 bytes
OPC UA	10 000 variables
HMI	Depends on RAM of the reference
CoDeSys protocol	Depends on RAM of the reference

Elements of the Toolbar

Element	Description	
View button	The **View** button allows you to set the following filters to reduce the number of displayed variables:	
	Unconfigured from Project	Even variables not yet added to the symbol configuration, but available for this purpose in the project, are displayed.
	Unconfigured from Libraries	Also variables from libraries, not yet added to the symbol configuration, but available for this purpose in the project, are displayed.
	Symbols exported via attribute	This setting is effective only when the unconfigured variables are displayed (see the two filters described above). It has the effect that also those variables are listed, which are already selected for getting symbols by {`attribute 'symbol' := 'read'`} within their declaration. Such symbols are displayed grayed. The **Attribute** column shows which access right is set for the variable by the pragma. Refer to the following description of the **Access Rights** column found in the EcoStruxure Machine Expert Programming Guide.
Build button	The **Build** button allows you to build the project. The build run refreshes the variables view in the **Symbol Configuration** editor.	
Settings button	The **Settings** button allows you to activate the following options:	
	Support OPC UA features	This function is not supported.
	Include comments in XML	This has the effect that comments assigned to variables are also exported to the symbol file.
	Include node flags in XML	This has the effect that the flags that contain the namespace are also exported to the symbol file. They provide additional information on the origin of the node in the namespace when OPC UA is active.
	Configure comments and attributes...	Opens the **Comments and Attributes** dialog box that allows you to configure the contents of the symbol configuration and the XML file.
	Configure synchronisation with IEC tasks...	Opens the **Options** tab of the **Properties** dialog box of the selected controller. Refer to the description of the **Access variables in sync with IEC tasks** option in the Menu Commands Online Help. **NOTE:** Do not activate the option **Configure synchronisation with IEC tasks...** for motion and real time-critical application because the delayed start of IEC tasks leads to a higher jitter. For further information, refer to the paragraph *Additional Information on the Option Configure synchronisation with IEC tasks...* found in the EcoStruxure Machine Expert Programming Guide.
	Compatibility Layout	Select this option to calculate the data output in the same way as in SoMachine / SoMachine Motion versions prior to V4.3. Do not use this layout together with exported `STRUCT`s that use the attributes *pack_mode* or *relative_offset*. The data layout created for the client is adapted as far as possible to the layout created by the compiler.
	Optimized Layout	Select this option to calculate the data output in an optimized form, independent of the internal compiler layout. The optimization impacts only variables of a structured type and function blocks. No gaps with padding bytes are generated for members that are not published, for example, because they are deactivated in the **Symbol Configuration**. For internal members, for example, function blocks implementing interfaces, no gaps are created either. This option is selected by default. The setting is preserved after a **Project update**.
Tools button	**Save XML Scheme File...**	Opens the dialog box for saving a file in the file system. Allows you to create an XSD (XML Schema Definition) format of the symbol file, for use in external programs.

Description of the Table

Columns of the **Symbol configuration** table:

Column	Description
Symbols	The column shows a list of POUs. You can select variables to be exported. If you select variables of a structured data type, all members of the structure will be exported. You may also select only particular member variables in the dialog box **Symbol Configuration for Data Type**. Click the browse **...** button in the **Members** column to open this dialog box. For further information, refer to the description of the **Members** column.
Access Rights	To modify the access rights for a selected item, click the **Access Rights** column. Each mouse-click switches the symbol within the following definitions: • : read and write • : write-only • : read-only • none
Maximal	Shows the maximum access rights.
Type	Shows the data type of the variable. Variables of an alias data type are displayed as shown in the following example of a variable: `myVar : MY_INT`, , where `MY_INT` is an alias declared as: `TYPE MY_INT : INT; END_TYPE`. In this case, the **Type** column shows `MY_INT : INT`.
Members	Click the **...** button in the **Members** column to open the **Symbol Configuration for Data Type** dialog box. It allows you to select only particular member variables. In case of nested types, this dialog box again provides a button to open another **Symbol Configuration for Data Type** dialog box. This selection applies to all instances of this data type for which symbols are exported. If not all members of a structured type are exported, then an asterisk (*) is displayed in the check boxes of the members to indicate that all exportable members of that type are exported.
Comment	Shows any comments which have been added in the declaration of the variable.

NOTE: With the POU property **Link Always**, an uncompiled POU can be forced to be downloaded to the controller. If this property is set in the **Build** tab of the **Properties** dialog box of the selected POU, then all variables declared in this POU are available, even if those variables are not referenced by other code. Alternatively, you can use the pragma {attribute linkalways} found in the EcoStruxure Machine Expert Programming Guide to make not compiled variables available in the symbol configuration.

Variables which are configured to be exported but which are not valid in the application, for example because their declaration has been removed, are shown in red. This also applies to the concerned POU or library name.

By default, a symbol file is created with a code generation run. This file is transferred to the device with the next download. If you want to create the file without performing a download, use the command **Generate code**, by default available in the **Build** menu.

NOTE: Variables of a global variable list (GVL) will only be available in the symbol configuration if at least one of them is used in the programming code.

Comments and attributes Dialog Box

The **Comments and attributes** dialog box opens upon clicking **Settings > Configure comments and attributes**. It contains the following elements:

Element	Description
Symbol table contents	
Enable extended OPC UA information	This function is not supported.
Include comments	
Include attributes	
Also include comments and attributes for type nodes	
XML symbol file contents	
Include namespace node flags	Namespace node flags provide additional information about the origin of a node in the namespace. Node flags are available in the symbol table when OPC UA is activated. Deselect this option to prevent namespace node flags from being inserted in the XML file if your parser cannot process them.
Include comments	Select this option to save comments in the XML file. In SoMachine / SoMachine Motion versions prior to V4.4, this includes the setting **Prefer docu comments**.
Include attributes	Select this option to save attributes in the XML file.
Also include comments and attributes for type nodes	This option is only available if the option **Include comments** or the option **Include attributes** is activated. If this option is selected, the information for type nodes is also included (user-defined types, such as STRUCT and ENUM elements). If this option is not selected, comments and attributes are only available for directly exported variables.
Select comments These parameters are only available if one of the **Include comments** options is activated.	
Include docu comments: **///They start with triple slash and are usually /// formatted in ReST (library documentation)**	Select the options to determine the kind of comments that are saved in the symbol configuration.
Include normal comments: **(*IEC/ Pascal style comments *) // C++-Style comments with double-slash**	
Always include both types of comments	
Prefer docu comments, fallback to normal ones	
Prefer normal comments, fallback to docu comments	
Filter Attributes (case insensitive) These parameters are only available if one of the **Include attributes** options is activated.	
Include all attributes (“foo”, “bar”, “foo.bar”)	Select the options to determine the attributes that are saved in the symbol configuration.
Match simple identifiers (“foo”, “bar”)	
Include attributes starting with:	
Filter Attributes with regular expression	

Additional Information on Configure synchronization with IEC tasks...

To achieve synchronous, consistent access, the runtime system postpones the processing of read or write request of the symbolic client until no IEC task is executed. As soon as this gap is found, the restart of IEC tasks is postponed until the requested values have been copied to the variables list.

This option is useful for permanently running systems without production clocking, for example, if process values are to be written cyclically in fixed time intervals (such as 60 s).

NOTE: Do not activate the option **Configure synchronization with IEC tasks...** for motion and real time-critical application because the delayed start of IEC tasks leads to a higher jitter.

If you intend to use the **Configure synchronization with IEC tasks...** option, consider the following points when you define the variable lists that are read and written:

- Configure synchronous and consistent access only for those variables for which it is necessary.
- Create separate lists for consistent variables and for variables that may be inconsistent.
- Create several small lists containing consistent variables instead of one large list.
- Define the time intervals for cyclically reading values as large as possible.

The option **Configure synchronization with IEC tasks...** is available at two different locations in EcoStruxure Machine Expert:

- In the **Symbol configuration** editor, as an option of the **Settings** button. (If a symbol configuration is available in the application).
- In the **Options** tab of the **Properties** dialog box of the selected controller.

NOTE: In order for the setting to take effect, perform a **Download** or **Online Change** of the applications on the controller and update the boot applications.

Embedded Inputs and Outputs Configuration

Configuring the Fast I/Os

Embedded I/Os Configuration

Overview

The embedded I/O function allows configuration of the controller inputs and outputs.

The TM262• controllers provide:

- 4 fast inputs
- 4 fast outputs

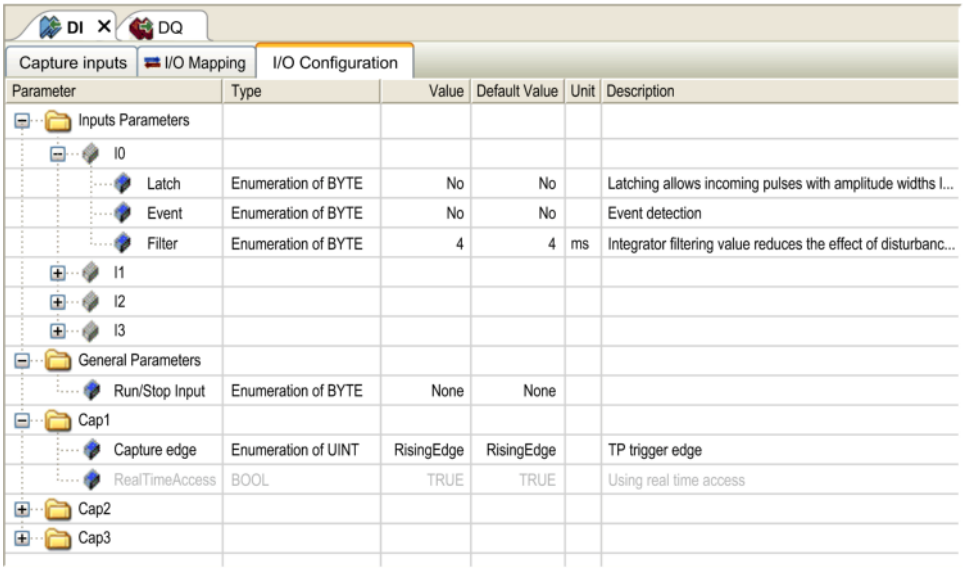
Accessing the I/O Configuration Window

Follow these steps to access the I/O configuration window:

Step	Description
1	Double-click **DI** (digital inputs) or **DQ** (digital outputs) in the **Devices tree**. Refer to Devices tree, page 34.
2	Select the **I/O Configuration** tab.

Configuration of Digital Inputs

This figure shows the **I/O Configuration** tab for digital inputs:



Parameter	Type	Value	Default Value	Unit	Description
Inputs Parameters					
I0					
Latch	Enumeration of BYTE	No	No		Latching allows incoming pulses with amplitude widths l...
Event	Enumeration of BYTE	No	No		Event detection
Filter	Enumeration of BYTE	4	4	ms	Integrator filtering value reduces the effect of disturbanc...
I1					
I2					
I3					
General Parameters					
Run/Stop Input	Enumeration of BYTE	None	None		
Cap1					
Capture edge	Enumeration of UINT	RisingEdge	RisingEdge		TP trigger edge
RealTimeAccess	BOOL	TRUE	TRUE		Using real time access
Cap2					
Cap3					

NOTE: For more information on the **I/O Mapping** tab, refer to the EcoStruxure Machine Expert Programming Guide.

Digital Input Configuration Parameters

For each digital input, you can configure the following parameters:

Parameter	Value	Description	Constraint
Filter	0.000 ms 0.001 ms 0.002 ms 0.005 ms 0.01 ms 0.05 ms 0.08 ms 0.5 ms 1 ms 4 ms* 12 ms	Reduces the effect of noise on a controller input.	Configure **Filter** to 0.000 if you do not want to filter the signal.
Latch	No* Yes	Allows incoming pulses with amplitude widths shorter than the controller scan time to be captured and recorded.	Available if **Event** disabled.
Event	No* Rising edge Falling edge Both edges	Event detection	Available if **Latch** disabled. When **Both edges** is selected, and the input state is TRUE before the controller is powered on, the first falling edge is ignored.
Run/Stop Input	None* I0...I3	The Run/Stop input can be used to run or stop the controller application.	Select one of the inputs to use as the Run/Stop Input.
* Parameter default value			

NOTE: The selection is grey and inactive if the parameter is unavailable.

Run/Stop Input

This table presents the different states:

Input states	Result
State 0	Stops the controller and ignores external Run commands. **FSP** LED is red ON.
A rising edge	From the STOPPED state, initiate a start-up of an application in RUNNING state, if no conflict with Run/Stop switch position.
State 1	The application can be controlled by: • The software (Run/Stop) • A hardware Run/Stop switch • Application (Controller command) • Network command (Run/Stop command) Run/Stop command is available through the Web server command.

NOTE: Run/Stop input is managed even if the option **Update I/O while in stop** is not selected in Controller Device Editor (**PLC settings** tab), page 78.

Inputs assigned to configured expert functions cannot be configured as Run/Stop inputs.

For further details about controller states and states transitions, refer to Controller State Diagram.

⚠WARNING

UNINTENDED MACHINE OR PROCESS START-UP

- Verify the state of security of your machine or process environment before applying power to the Run/Stop input.
- Use the Run/Stop input to help prevent the unintentional start-up from a remote location.

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

Capture Input

Capture Inputs tab allows you to select captures, exclusively for motion applications, and manage them in the **I/O Configuration** tab.

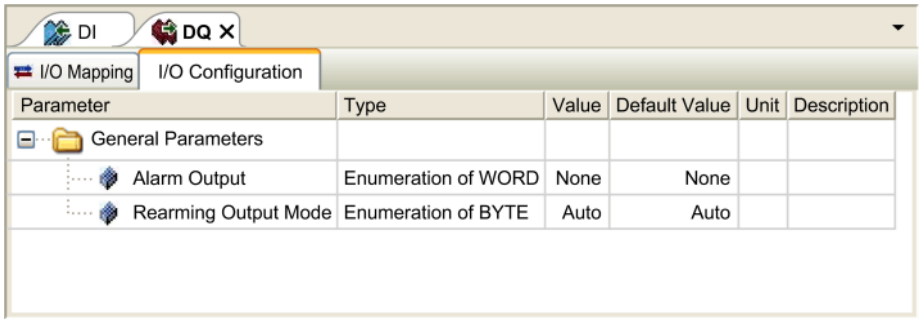
For each capture, you can configure the following parameters:

Parameter	Type	Value	Description	Constraint
Capture edge	**UINT Enum**	**Falling edge** **Rising edge** **Both edges**	Configure the edge on which the encoder position is captured.	Enable the capture positions in **Capture Inputs** tab. Do not use with the function blocks from the **M262 Encoder** Library.
RealTimeAc-cess	**BOOL**	**TRUE**	Using real time access.	Enable the capture positions in **Capture Inputs** tab. Do not use with the function blocks from the **M262 Encoder** Library.

For more information on motion applications and function blocks, such as **MC_TouchProbe** and **MC_AbortTrigger**, refer to M262 Synchronized Motion Control Library Guide.

Configuration of Digital Outputs

This figure shows the **I/O Configuration** tab for digital outputs:



NOTE: For more information on the **I/O Mapping** tab, refer to the EcoStruxure Machine Expert Programming Guide.

Digital Output Configuration Parameters

This table presents the function of the different parameters:

Parameter	Function
General Parameters	
Alarm Output	Select an output to be used as alarm output, page 105.
Rearming Output Mode	Select the rearming output mode, page 105.

NOTE: The selection is grey and inactive if the parameter is unavailable.

Alarm Output

This output is set to logical 1 when the controller is in the RUNNING state and the application program is not stopped at a breakpoint.

The alarm output is set to 0 when a task is stopped at a breakpoint to signal that the controller has stopped executing the application and when the controller is stopped.

NOTE: Outputs assigned to configured expert functions cannot be configured as the alarm output.

Rearming Output Mode

Fast outputs of the Modicon M262 Logic/Motion Controller use push/pull technology. In case of detected error (short-circuit or over temperature), the output is put in the default value and the condition is signaled by status bit and PLC_R_IO_STATUS. This is also signaled by `%IX1.0`.

Two behaviors are possible:

- **Automatic rearming**: as soon as the detected error is corrected, the output is set again according to the current value assigned to it and the diagnostic value is reset.
- **Manual rearming**: when an error is detected, the status is memorized and the output is forced to the default value until user manually clears the status (see I/O mapping channel).

In the case of a short-circuit or current overload, the common group of outputs automatically enters into thermal protection mode (all outputs in the group are set to 0), and are then periodically rearmed (each second) to test the connection state. However, you must be aware of the effect of this rearming on the machine or process being controlled.

⚠ WARNING

UNINTENDED MACHINE START-UP

Inhibit the automatic rearming of outputs if this feature is an undesirable behavior for your machine or process.

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

NOTE: Automatic rearming of outputs can be inhibited through the configuration.

Hardware Encoder Interface

Hardware Encoder Interface

Introduction

The controller has a specific hardware encoder interface that can support:

- Incremental encoder
- SSI absolute encoder

Incremental Mode Principle Description

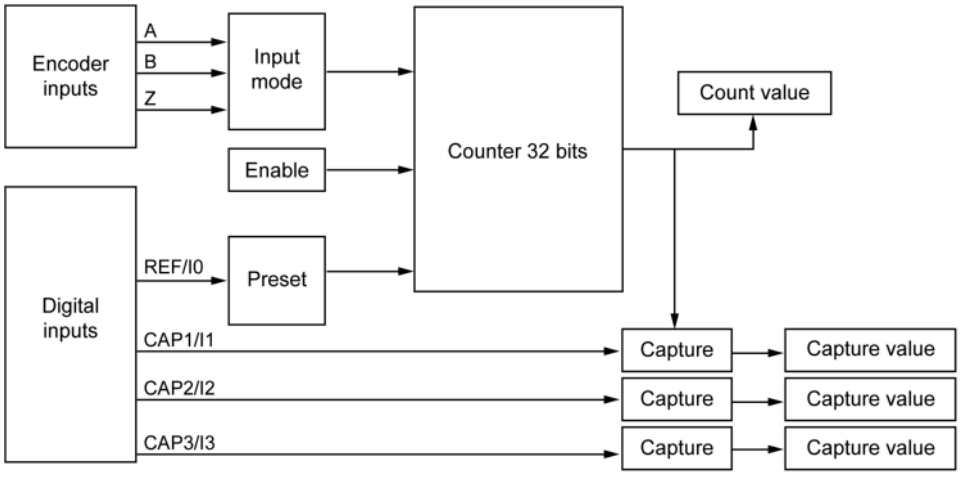
The incremental mode behaves like a standard up/down counter, using pulses and counting these pulses.

Positions must be preset and counting must be initialized to implement and manage the incremental mode.

The counter value can be stored in the capture register by configuring an external event.

Incremental Mode Principle Diagram

The following diagram provides an overview of the encoder in incremental mode:



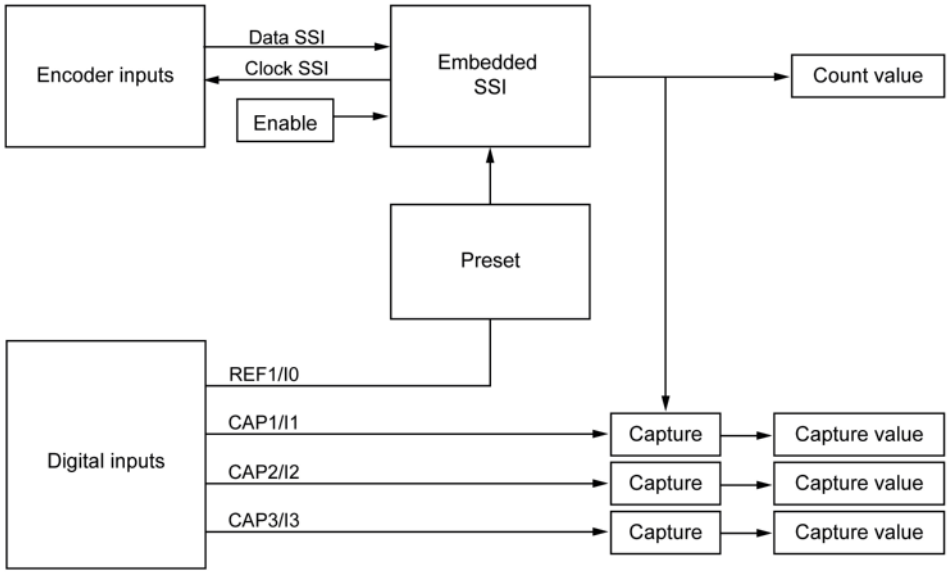
SSI Mode Principle Description

The SSI (Synchronous Serial Interface) mode allows the connection of an absolute encoder.

The position of the absolute encoder is read by an SSI link.

SSI Mode Principle Diagram

The following diagram provides an overview of the encoder in SSI mode:



I/O mapping

This variable is used by the library to identify the encoder, incremental or SSI, to which the function block applies.

Adding an Encoder

Introduction

In order to use the encoder interface, the Modicon M262 Logic/Motion Controller has a specific hardware encoder interface that can support:

- Incremental encoder
- SSI encoder

For more information on usable function blocks, refer to Modicon M262 Logic/Motion Controller - Encoder Library Guide (see Modicon M262 Logic/Motion Controller, Encoder Library Guide).

Adding an Encoder

To add an encoder to your controller, select the encoder in the **Hardware Catalog**. Drag and drop it to the **Devices tree** on one of the highlighted nodes.

For more information on adding a device to your project, refer to:

• Using the Hardware Catalog found in the EcoStruxure Machine Expert Programming Guide

• Using the Contextual Menu or Plus Button found in the EcoStruxure Machine Expert Programming Guide

Incremental Encoder Configuration

To configure the incremental encoder, double-click the encoder node in the **Devices tree**.

This table describes the incremental encoder configuration parameters:

Parameter	Type	Value	Default Value	Unit	Description
Power supply					
Voltage selection	**BYTE Enum**	**None** **5 V** **24 V**	**None**	–	–
Power supply monitor	**BYTE Enum**	**Enabled** **Disabled**	**Disabled**	–	**Enable the power supply monitor**
General					
Input Mode	**BYTE Enum**	**Normal Quadrature x 1** **Normal Quadrature x 2** **Normal Quadrature x 4** **Reverse Quadrature x 1** **Reverse Quadrature x 2** **Reverse Quadrature x 4**	**Normal Quadrature x 1**	–	**Select the period measurement interval**
Counting Inputs					
A Input					
Filter	**BYTE Enum**	**0.000** **0.001** **0.002** **0.005** **0.01** **0.05** **0.08** **0.5** **1** **4** **12**	**4**	**ms**	**Set the filtering value to reduce the bounce effect on the input**
B Input					
Filter	**BYTE Enum**	**0**	**4**	**ms**	**Set the filtering value to reduce the bounce effect on the input**
Preset Input					
Z Input					
Filter	**BYTE Enum**	**0.000** **0.001** **0.002** **0.005** **0.01** **0.05** **0.08** **0.5** **1** **4** **12**	**4**	**ms**	**Set the filtering value to reduce the bounce effect on the input**

SSI Encoder Configuration

To configure the **SSI Encoder**, double-click the encoder node in the **Devices tree**.

This table describes the SSI encoder configuration parameters:

Parameter	Type	Value	Default Value	Unit	Description
Power supply					
Voltage selection	**BYTE Enum**	**None** **5 V** **24 V**	**None**	–	–
Power supply monitor	**BYTE Enum**	**Disabled**	**Disabled**	–	**Enable the power supply monitor**
Synchronous Serial Interface (SSI)					
Transmission Speed	**BYTE Enum**	**100** **250** **500**	**250**	KHz	**Select the speed of data transmission**
Number of bits per frame	**USINT (8...64)**	**8**	**8**	–	**Set the number of bits per frame (header + data bits + status + parity)**
Number of data bits	**USINT (8...32)**	**8**	**8**	–	**Set the number of bits to count turn + bits to count points per turn**
Number of data bits / turn	**USINT (8...16)**	**8**	**8**	–	**Set the number of data bits to count points per turn**
Number of status bits	**USINT (0...4)**	**0**	**0**	–	**Set the number of bits to be reserved for the status**
Parity	**BYTE Enum**	**None**	**None**	–	**Select the parity**
Resolution reduction	**USINT (0...17)**	**0**	**0**	–	**Set the resolution code**
Binary coding	**BYTE Enum**	**Binary**	**Binary**	–	**Select the binary coding mode**

Motion Functions

You can configure specific elements, exclusively for motion applications. For more information, refer to **Motion Functions** tab, page 109.

Encoder Motion Functions

Introduction

The encoder **Motion Functions** tab allows you to configure specific elements, exclusively for motion applications.

NOTE: These **Motion Functions** must not be used with the **M262Encoder** library when check boxes for **Axis**, **Scaling**, **Filter** and/or **DeadTimeCompensation** are enabled.

Configuring the Motion Functions

This table describes the procedure to configure the motion functions

Step	Action
1	Double-click the encoder node in the **Devices tree**.
2	Open the **Motion Functions** tab.
3	Enable check boxes for **Axis**, **Scaling**, **Filter** and/or **DeadTimeCompensation**. **Result:** The configuration parameters are displayed in the **Incremental Encoder configuration** tab or in the **SSI Encoder configuration** tab.

Incremental/SSI Encoder

This table describes the motion functions configuration parameters of the incremental encoder or of the SSI encoder:

Parameter	Type	Value	Default Value	Description
Scaling				
IncrementResolution	**DINT**	1...2,147,483,647	**131072**	**IncrementResolution**
PositionResolution	**LREAL**	1.0...1.7976931348623158e+308	**360.0**	**PositionResolution**
GearIn	**UDINT**	1...4,294,967,295	**1**	**GearIn**
GearOut	**UDINT**	1...4,294,967,295	**1**	**GearOut**
InvertDirection	**BOOL**	**FALSE**	**FALSE**	**Invert movement direction of the axis**
Filter				
AverageDuration	**UDINT**	0...1024	**0**	**Filter duration in Sercos cycles**
DeadTimeCompensation				
Delay	**LREAL**	-100.0...100.0	**0**	**Delay of feedback movement values (position/velocity/acceleration) in milliseconds. This delay will be compensated by the system.**

NOTE: The **DeadTimeCompensation delay**, without a **filter**, results in a very high signal deviation of the feedback velocity and can result in unintended behavior of a coupled slave axis.

⚠WARNING

UNINTENDED EQUIPMENT OPERATION

Use a filter whenever you use a delay for DeadTimeCompensation.

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

Expansion Modules Configuration

Overview

This chapter describes how to configure the TMS and TM3 expansion modules for the Modicon M262 Logic/Motion Controller.

TM3 I/O Configuration General Description

Introduction

In your project, you can add I/O expansion modules to your M262 Logic/Motion Controller to increase the number of digital and analog inputs and outputs over those native to the controller (embedded I/O).

You can add either TM3 I/O expansion modules to the controller, and further expand the number of I/O via TM3 transmitter and receiver modules to create remote I/O configurations. Special rules apply in all cases when creating local and remote I/O expansions (refer to Maximum Hardware Configuration (see Modicon M262 Logic/Motion Controller, Hardware Guide)).

The I/O expansion bus of the M262 Logic/Motion Controller is created when you assemble the I/O expansion modules to the controller.

I/O Expansion Bus Errors

If the controller cannot communicate with one or more I/O expansion modules contained in the program configuration, and those modules are not configured as optional modules (refer to Optional I/O Expansion Modules, page 118), the controller detects it as an I/O expansion bus error. The unsuccessful communication may be detected during the startup of the controller or during runtime, and there may be any number of causes. Causes of communication exceptions on the I/O expansion bus include, among other things, disconnection of or physically missing I/O modules, electromagnetic radiation beyond published environmental specifications, or otherwise inoperative modules.

NOTE: In fallback mode, the TM3 bus waits for controller communications for approximately 200 milliseconds before applying fallback values unless the controller sends out a bus reset, setting the output values to their initialization values after two consecutive bus task cycles. If the two bus task cycles exceeds the TM3 bus timeout, output modules first apply their fallback values and then apply initialization values when the bus reset is sent.

If an I/O expansion bus error is detected:

- The system status LED **I/O** of the controller is illuminated in red indicating an I/O error.
- When EcoStruxure Machine Expert is in online mode, a red triangle appears next to the TM3 expansion module or modules in error and next to the **IO_ Bus** node on the **Devices tree** window.

The following diagnostic information is also available:

- Bit 0 and bit 1 of the *PLC_R.i_lwSystemFault_1* system variable are set to 0.
- The *PLC_R.i_wIOStatus1* and *PLC_R.i_wIOStatus2* system variables are set to *PLC_R_IO_BUS_ERROR*.
- The *TM3_MODULE_R[i].i_wModuleState* system variable, where *[i]* identifies the TM3 expansion module in error, is set to *TM3_BUS_ERROR*.
- The *TM3_GetModuleBusStatus* function block returns the *TM3_ERR_BUS* error code (see Modicon M262 Logic/Motion Controller, System Functions and Variables, System Library Guide).

Refer to PLC_R (see Modicon M262 Logic/Motion Controller, System Functions and Variables, System Library Guide) and TM3_MODULE_R structures for details on system variables.

Active I/O Expansion Bus Error Handling

This figure shows the select mode tab:

Parameter	Type	Value	Default Value	Unit	Description
Optional module	Enumeration of BYTE	No	No		
Functional Mode	Enumeration of BYTE	1	1		1 = Normal Mode 2 = Fallback Mode (TM3 DIO module with SV > = 2.0)

In the TM3 configuration, you can use the **Normal Mode** (**1**) or the **Fallback Mode** (**2**).

The *TM3_BUS_W.q_wIOBusErrPassiv* system variable is set to *ERR_ACTIVE* by default to specify the use of active I/O error handling. The application can set this bit to *ERR_PASSIVE* to use passive I/O error handling instead.

By default, when the controller detects a TM3 module in bus communication error, it sets the bus to a "bus off" condition whereby the TM3 expansion module outputs image value are set to 0 or in Fallback value depending the mode used, and the inputs image value are set to 0. A TM3 expansion module is considered to be in bus communication error when an I/O exchange with the expansion module has been unsuccessful for at least two consecutive bus task cycles. When a bus communication error occurs, the *TM3_MODULE_R[i].i_wModuleState* system variable, where *[i]* is the expansion module number in error, is set to *TM3_BUS_ERROR*. The other bits are set to *TM3_OK*.

> **NOTE:** In **Fallback Mode**, TM3 I/O Bus will wait for controller communications for approximately 200 milliseconds before applying fallback values unless the controller send out a bus reset, setting all output values to their initialization value after two consecutive bus task cycles. If the two bus task cycles exceeds the TM3 I/O Bus timeout, outputs modules will first apply their fallback values and then apply initialization values when the bus reset is sent.

Normal I/O expansion bus operation can only be restored after eliminating the source of the error and performing one of the following:

- Power cycle
- New application download
- Restarting the I/O Bus by setting the *TM3_BUS_W.q_wIOBusRestart* system variable to 1. The bus is restarted only if no expansion modules are in error (*TM3_MODULE_R[i].i_wModuleState* = *TM3_BUS_ERROR*). Refer to Restarting the I/O Expansion Bus, page 114.
- Issuing a **Reset Warm** or **Reset Cold** command with the software, page 66.

This table describes the behavior of the modules connected to the controller, or to the Receiver module, according to their state and type:

Module Type	Modules Without Fallback Management	Modules With Fallback Management
Reset Cold	Default value (1)	
Reset Warm	Default value (1)	
Reset Origin (Empty controller)	Initialization value (2)	
STOP	Default value (1)	
Detected communication error on TM3 bus	Initialization value (2)	
Controller in HALT state	Default value (1)	
TM3 Transmitter-Receiver cable gets cut	Initialization value (2)	
(1): Value set in the software configuration screen. **(2)**: Module I/O value state after receiving a reset command from the bus.		

Passive I/O Expansion Bus Handling

This figure shows the select mode tab:

Parameter	Type	Value	Default Value	Unit	Description
Optional module	Enumeration of BYTE	No	No		
Functional Mode	Enumeration of BYTE	1	1		1 = Normal Mode 2 = Fallback Mode (TM3 DIO module with SV > = 2.0)

In the TM3 configuration, you can use the **Normal Mode** (**1**) or the **Fallback Mode** (**2**).

In **Normal Mode** (**1**), the application can set the system variable *TM3_BUS_W.q_wIOBusErrPassiv* to *ERR_PASSIVE* to use passive I/O error handling. This error handling is provided to afford compatibility with previous firmware versions.

When passive I/O error handling is in use, the controller attempts to continue data bus exchanges with the modules during bus communication errors. While the expansion bus error persists, the controller attempts to re-establish communication on the bus with incommunicative modules, depending on the type of I/O expansion module:

- For TM3 I/O expansion modules, the value of the I/O channels is maintained (**Keep current values**) for approximately 10 seconds while the controller attempts to re-establish communication. If the controller cannot re-establish communications within that time, the affected TM3 I/O expansion outputs are set to 0.

In **Fallback Mode** (**2**), the application can set the system variable *TM3_BUS_W.q_wIOBusErrPassiv* to *ERR_PASSIVE* to use passive I/O error handling. This error handling is provided to afford compatibility with previous firmware versions.

When passive I/O error handling is in use, the controller attempts to continue data bus exchanges with the modules during bus communication errors. While the expansion bus error persists, the controller attempts to re-establish communication on the bus with incommunicative modules, depending on the type of I/O expansion module:

- For TM3 I/O expansion modules, the value of the I/O channels is maintained (**Keep current values**) for approximately 200 milliseconds while the controller attempts to re-establish communication. If the controller cannot re-establish communications within that time, the affected TM3 I/O expansion outputs are set to fallback value.

In either case, the controller continues to solve logic and, if your controller is so equipped, the embedded I/O continues to be managed by the application

(“managed by application program, page 64”) while it attempts to re-establish communication with the incommunicative I/O expansion modules. If the communication is successful, the I/O expansion modules resume to be managed by the application. If communication with the I/O expansion modules is unsuccessful, you must resolve the reason for the unsuccessful communication, and then cycle power on the controller system, or issue a **Reset Warm** or **Reset Cold** command with the software, page 66.

The value of the incommunicative I/O expansion modules input image is maintained and the output image value is set by the application.

Further, if the incommunicative I/O module(s) disturb the communication with unaffected modules, the unaffected modules are also considered to be in error and the *TM3_MODULE_R[i].i_wModuleState* system variable (where *[i]* is the expansion module number) is set to *TM3_BUS_ERROR*. However, with the ongoing data exchanges that characterize the Passive I/O Expansion Bus Error Handling, the unaffected modules apply the data sent, and do not apply the fallback values as for the incommunicative module.

Therefore, you must monitor within your application the state of the bus and the error state of the module(s) on the bus, and take the appropriate action necessary given your particular application.

Refer to Controller States Description, page 60 for more information on the actions taken upon startup of the controller when an I/O expansion bus error is detected.

This table describes the behavior of the modules connected to the controller, or to the Receiver module, according to their state and type:

Module Type	**Modules Without Fallback Management**		**Modules With Fallback Management**	
	Modules Connected to Controller	**Modules Connected to Receiver module**	**Modules Connected to Controller**	**Modules Connected to Receiver module**
Reset Cold	Default value (1)		Default value (1)	
Reset Warm	Default value (1)		Default value (1)	
Reset Origin (Empty controller)	Initialization value (2)		Initialization value (2)	
STOP	Default value (1)		Default value (1)	
Detected communication error on TM3 bus	Maintain during 10 seconds then initialization value (2)		Configured fallback value	
Controller in HALT state	Default value (1)		Default value (1)	
TM3 Transmitter-Receiver cable gets cut	Value controlled by the application	Initialization value (2)	Value controlled by the application	Initialization value (2)
(1): Value set in the software configuration screen. **(2)**: Module I/O value state after receiving a reset command from the bus.				

Restarting the I/O Expansion Bus

When active I/O error handling is being applied, that is, embedded and TM3 outputs set to 0 or fallback value when a bus communication error is detected, the application can request a restart of the I/O expansion bus while the controller is still running (without the need for a Cold Start, Warm Start, power cycle, or application download).

The *TM3_BUS_W. q_wIoBusRestart* system variable is available to request restarts of the I/O expansion bus. The default value of this bit is 0. Provided at least one TM3 expansion module is in error (*TM3_MODULE_R[i].i_wModuleState* set to *TM3_BUS_ERROR*), the application can set *TM3_BUS_W. q_wIoBusRestart* to 1 to request a restart of the I/O expansion bus. On detection of a rising edge of this bit, the controller reconfigures and restarts the I/O expansion bus if all of the following conditions are met:

- The *TM3_BUS_W.q_wIOBusErrPassiv* system variable is set to *ERR_ACTIVE* (that is, I/O expansion bus activity is stopped)
- Bit 0 and bit 1 of the *PLC_R.i_lwSystemFault_1* system variable are set to 0 (I/O expansion bus is in error)
- The *TM3_MODULE_R[i].i_wModuleState* system variable is set to *TM3_BUS_ERROR* (at least one expansion module is in bus communication error)

If the *TM3_BUS_W.q_wIoBusRestart* system variable is set to 1 and any of the above conditions is not met, the controller takes no action.

Match Software and Hardware Configuration

The I/O that may be embedded in your controller is independent of the I/O that you may have added in the form of I/O expansion. It is important that the logical I/O configuration within your program matches the physical I/O configuration of your installation. If you add or remove any physical I/O to or from the I/O expansion bus or, depending on the controller reference, to or from the controller (in the form of cartridges), then you must update your application configuration. This is also true for any field bus devices you may have in your installation. Otherwise, there is the potential that the expansion bus or field bus no longer function while the embedded I/O that may be present in your controller continues to operate.

⚠WARNING

UNINTENDED EQUIPMENT OPERATION

Update the configuration of your program each time you add or delete any type of I/O expansions on your I/O bus, or you add or delete any devices on your field bus.

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

Presentation of the Optional Feature for I/O Expansion Modules

I/O expansion modules can be marked as optional in the configuration. The **Optional module** feature provides a more flexible configuration by the acceptance of the definition of modules that are not physically attached to the controller. Therefore, a single application can support multiple physical configurations of I/O expansion modules, allowing a greater degree of scalability without the necessity of maintaining multiple application files for the same application.

You must be fully aware of the implications and impacts of marking I/O modules as optional in your application, both when those modules are physically absent and present when running your machine or process. Be sure to include this feature in your risk analysis.

⚠WARNING

UNINTENDED EQUIPMENT OPERATION

Include in your risk analysis each of the variations of I/O configurations that can be realized marking I/O expansion modules as optional, and in particular the establishment of TM3 Safety modules (TM3S…) as optional I/O modules, and make a determination whether it is acceptable as it relates to your application.

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

NOTE: For more details about this feature, refer to Optional I/O Expansion Modules, page 118.

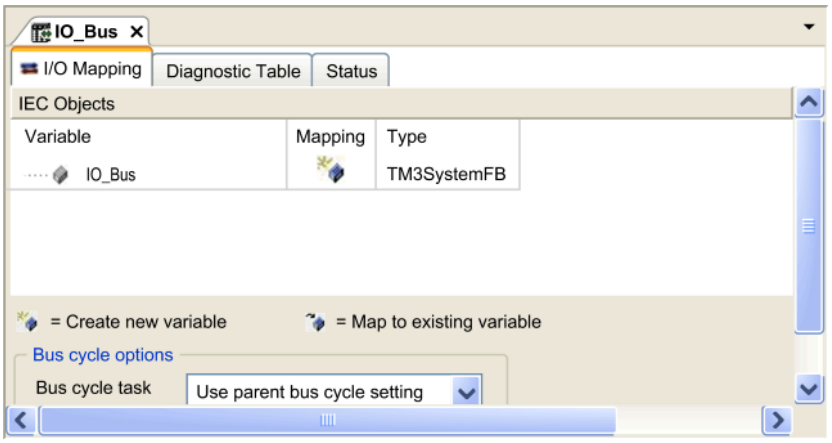
TM3 I/O Bus Configuration

Overview

TM3 I/O bus configuration enables you to select the task that drives TM3 physical exchanges. It can also override the configuration defined in the **PLC settings**, page 78 bus cycle task.

Configuring the I/O Bus

Follow these steps to configure the TM3 I/O bus:

Step	Action
1	In the **Devices tree**, double-click **IO_Bus**. **Result**: The **IO_Bus** editor tab displays: 
2	Set the **Bus cycle task** from the list to either of the following: • **Use parent bus cycle setting** (default) Sets the task for bus exchange as defined in the **PLC settings**. • **MAST** Sets the Master task for bus exchange irrespective of the task defined in the **PLC settings**.

TMS Expansion Module Configuration

Introduction

The Modicon M262 Logic/Motion Controller supports the TMS communication expansion modules.

TMS expansion modules connect to the left side of the controller and are dedicated to Ethernet and CANopen high speed communication. You can configure your TMS expansion modules in the software **Devices Tree**.

NOTE: The TMSES4 expansion module is not a standalone Ethernet switch.

For further information about the TMS expansion modules configuration, refer to the Modicon TMS Expansion Modules Programming Guide.

⚠ WARNING

UNINTENDED EQUIPMENT OPERATION

- Only use software approved by Schneider Electric for use with this equipment.
- Update your application program every time you change the physical hardware configuration.

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

Adding an Expansion Module

To add an expansion module to your controller, select the expansion module in the **Hardware Catalog**, drag it to the **Devices tree**, and drop it on one of the highlighted nodes.

For more information on adding a device to your project, refer to:

• Using the Hardware Catalog found in the EcoStruxure Machine Expert Programming Guide

• Using the Contextual Menu or Plus Button found in the EcoStruxure Machine Expert Programming Guide

Compatibility of the TMS Communication Expansion Modules

You can connect:

- 1 TMSCO1 for TM262L01MESE8T and TM262M05MESS8T
- 3 TMSES4 or 2 TMSES4 and 1 TMSCO1 for the other references

The TMSCO1 must be the leftmost module connected to the controller.

TM3 Expansion Module Configuration

Introduction

The Modicon M262 Logic/Motion Controller supports the following expansion modules:

- TM3 expansion modules:
  - Digital I/O modules
  - Analog I/O modules
  - Expert I/O modules
  - Safety modules
  - Transmitter and receiver modules

For further information about the TM3 expansion modules configuration, refer to the TM3 Expansion Modules Configuration Programming Guide.

⚠ WARNING

UNINTENDED EQUIPMENT OPERATION

- Only use software approved by Schneider Electric for use with this equipment.
- Update your application program every time you change the physical hardware configuration.

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

Adding an Expansion Module

To add an expansion module to your controller, select the expansion module in the **Hardware Catalog**, drag it to the **Devices tree**, and drop it on one of the highlighted nodes.

For more information on adding a device to your project, refer to:

• Using the Hardware Catalog found in the EcoStruxure Machine Expert Programming Guide

• Using the Contextual Menu or Plus Button found in the EcoStruxure Machine Expert Programming Guide

Optional I/O Expansion Modules

Presentation

I/O expansion modules can be marked as optional in the configuration. The **Optional module** feature provides a more flexible configuration by the acceptance of the definition of modules that are not physically attached to the controller. Therefore, a single application can support multiple physical configurations of I/O expansion modules, allowing a greater degree of scalability without the necessity of maintaining multiple application files for the same application.

Without the **Optional module** feature, when the controller starts up the I/O expansion bus (following a power cycle, application download or initialization command), it compares the configuration defined in the application with the physical I/O modules attached to the I/O bus. Among other diagnostics made, if

the controller determines that there are I/O modules defined in the configuration that are not physically present on the I/O bus, an error is detected and the I/O bus does not start.

With the **Optional module** feature, the controller ignores the absent I/O expansion modules that you have marked as optional, which then allows the controller to start the I/O expansion bus.

The controller starts the I/O expansion bus at configuration time (following a power cycle, application download, or initialization command) even if optional expansion modules are not physically connected to the controller.

The TM3 I/O expansion modules can be marked as optional.

NOTE: TM3 Transmitter/Receiver modules (the TM3XTRA1 and the TM3XREC1) cannot be marked as optional.

You must be fully aware of the implications and impacts of marking I/O modules as optional in your application, both when those modules are physically absent and present when running your machine or process. Be sure to include this feature in your risk analysis.

⚠ WARNING

UNINTENDED EQUIPMENT OPERATION

Include in your risk analysis each of the variations of I/O configurations that can be realized marking I/O expansion modules as optional, and in particular the establishment of TM3 Safety modules (TM3S…) as optional I/O modules, and make a determination whether it is acceptable as it relates to your application.

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

Marking an I/O Expansion Module as Optional

To add an expansion module and mark it as optional in the configuration:

Step	Action
1	Add the expansion module to your controller .
2	In the **Devices tree**, double-click the expansion module.
3	Select the **I/O Configuration** tab.
4	In the **Optional module** line, select **Yes** in the **Value** column:

Parameter	Type	Value	Default Value	Unit	Description
Optional module	Enumeration of BYTE	Yes	No		
Outputs					
QW0					
Type	Enumeration of BYTE	Not used	Not used		Range mode
Minimum	INT(-32768...32766)	-32768	-32768		Minimum value
Maximum	INT(-32767...32767)	32767	32767		Maximum value
QW1					
Type	Enumeration of BYTE	Not used	Not used		Range mode
Minimum	INT(-32768...32766)	-32768	-32768		Minimum value
Maximum	INT(-32767...32767)	32767	32767		Maximum value
Diagnostic					
Status Enabled	Enumeration of BYTE	Yes	Yes		

Modifiable by programming = Yes = No

Internal ID Codes

Controllers and bus couplers identify expansion modules by a simple internal ID code. This ID code is not specific to each reference, but identifies the logical structure of the expansion module. Therefore, different references can share the same ID code.

You cannot have two modules with the same internal ID code declared as optional without at least one mandatory module placed between them.

This table shows the internal ID codes of expansion modules:

Modules sharing the same internal ID code	ID code
TM3DI16K, TM3DI16, TM3DI16G	128
TM3DI8, TM3DI8G, TM3DI8A	132
TM3DQ16R, TM3DQ16RG, TM3DQ16T, TM3DQ16TG, TM3DQ16TK, TM3DQ16U, TM3DQ16UG, TM3DQ16UK	129
TM3DQ32TK, TM3DQ32UK	131
TM3DQ8R, TM3DQ8RG, TM3DQ8T, TM3DQ8TG, TM3DQ8U, TM3DQ8UG	133
TM3DM8R, TM3DM8RG	134
TM3DM16R	141
TM3DM24R, TM3DM24RG	135
TM3DM32R	143
TM3SAK6R, TM3SAK6RG	144
TM3SAF5R, TM3SAF5RG	145
TM3SAC5R, TM3SAC5RG	146
TM3SAFL5R, TM3SAFL5RG	147
TM3AI2H, TM3AI2HG	192
TM3AI4, TM3AI4G	193
TM3AI8, TM3AI8G	194
TM3AQ2, TM3AQ2G	195
TM3AQ4, TM3AQ4G	196
TM3AM6, TM3AM6G	197
TM3TM3, TM3TM3G	198
TM3TI4, TM3TI4G	199
TM3TI4D, TM3TI4DG	203
TM3TI8T, TM3TI8TG	200
TM3DI32K	130
TM3XTYS4	136
TM3XFHSC202, TM3XFHSC202G	216
TM3XHSC202, TM3XHSC202G	217

Optional Modules Diagnostic

The following diagnostic information is available: **TM3_MODULE_R[i].i_wModuleState** system variable, where **[i]** identifies the absent TM3 optional expansion module, is set to **TM3_MISSING_OPT_MOD**.

Ethernet Configuration

Introduction

This chapter describes how to configure the Ethernet network interface of the Modicon M262 Logic/Motion Controller.

Ethernet Features, Functions and Services

Presentation

Ethernet Features, Functions and Services

The controller supports the following services:

- Modbus TCP Server and Client, page 128
- DHCP Server, page 168
- Web Server, page 132
- FTP Server, page 129
- SNMP, page 131
- Controller as Target Device On EtherNet/IP, page 168
- Controller as Slave Device On Modbus TCP, page 187
- IEC VAR ACCESS, page 123
- WebVisualisation, page 127
- OPC UA Server, page 215

TM262• Specific Considerations

The TM262• have two different Ethernet networks. Each one gets its own and unique IP and MAC address.

The two Ethernet networks are called Ethernet 1 and Ethernet 2:

- Ethernet 1 is a separate 100 Mbit/s Ethernet port and dedicated to the Sercos communication for the TM262M•.
- Ethernet 2 is a dual 1000 Mbit/s port Ethernet switch.

For example, you can:

- Connect your PC to the Ethernet 1.
- Use a Modbus TCP I/O scanner with the Ethernet 2.

The Network Variables List (NVL) communication works on the Ethernet 1 port and Ethernet 2 port, only if both the Ethernet 1 port and Ethernet 2 port have a valid IP address and are connected to a device.

In addition, the TM262• allows you to connect your computer to the controller with a USB cable and to access the same services as with an Ethernet connection. See Connecting the Controller to a PC, page 236.

Ethernet Protocols

The controller supports the following protocols:

- IP (Internet Protocol) V4, V6
- UDP (User Datagram Protocol)
- TCP (Transmission Control Protocol)
- ARP (Address Resolution Protocol)
- ICMP (Internet Control Messaging Protocol)
- IGMP (Internet Group Management Protocol)

Communication Libraries

The communication libraries can be used in EcoStruxure Machine Expert. Refer to Communication Libraries folder of the online help.

Connections

This table shows the maximum number of connections:

Connection Type	Maximum
Modbus Server	8...24 connections (see the note below)
Modbus Client	8 connections
Modbus TCP I/O Scanner	64 channels
EtherNet/IP Scanner	64 connections
FTP Server	8 connections
Web Server	10 concurrent users
CoDeSys Protocol (software, trace, WebVisualisation, HMI devices)	8 connections
OPC UA Server	4 connections
OPC UA Client	5 connections

NOTE: The Modbus Server allows a maximum of 8 connections by default, but it is possible to increase this maximum to 24 connections.

To change the maximum number of connections, you must add the parameter 1012 in the Post Configuration file Machine.cfg (see Post Configuration File Example, page 232). When this parameter is added, its value is applied upon a reset, an application download or a power cycle.

In case the maximum number of connections is decreased while there are more ongoing connections than this maximum, the Modbus Server closes the exceeding oldest connections.

Each connection based on TCP manages its own set of connections as follows:

1. When a client tries to open a connection that exceeds the poll size, the controller closes the oldest connection.
2. If all connections are busy (exchange in progress) when a client tries to open a new one, the new connection is denied.
3. The server connections stay open as long as the controller stays in operational states (*RUNNING*, *STOPPED*, *HALT*).
4. The server connections are closed when leaving operational states (*RUNNING*, *STOPPED*, *HALT*), except in case of power outage (because the controller does not have time to close the connections).

Connections can be closed when the originator of the connection requests to close the connection it had previously opened.

Services Available

With an Ethernet communication, the **IEC VAR ACCESS** service is supported by the controller. With the **IEC VAR ACCESS** service, data can be exchanged between the controller and an HMI.

The **Network Variables** service is also supported by the controller. With the **Network Variables** service, data can be exchanged between controllers.

NOTE: For more information, refer to the EcoStruxure Machine Expert Programming Guide.

IP Address Configuration

Introduction

There are different ways to assign the IP address to the added Ethernet interface of the controller:

- Address assignment by DHCP server based on the Network Name of the Ethernet interface
- Address assignment by BOOTP server based on the MAC address of the Ethernet interface
- Fixed IP address
- Post configuration file, page 229. If a post configuration file exists, this assignment method has priority over the others.

The IP address can also be changed dynamically through the:

- Communication Settings tab, page 77
- **changeIPAddress** function block, page 282

NOTE: If the attempted addressing method is unsuccessful, the link uses a default IP address, page 126 derived from the MAC address.

Carefully manage the IP addresses because each device on the network requires a unique address. Having multiple devices with the same IP address can cause unintended operation of your network and associated equipment.

⚠ WARNING

UNINTENDED EQUIPMENT OPERATION

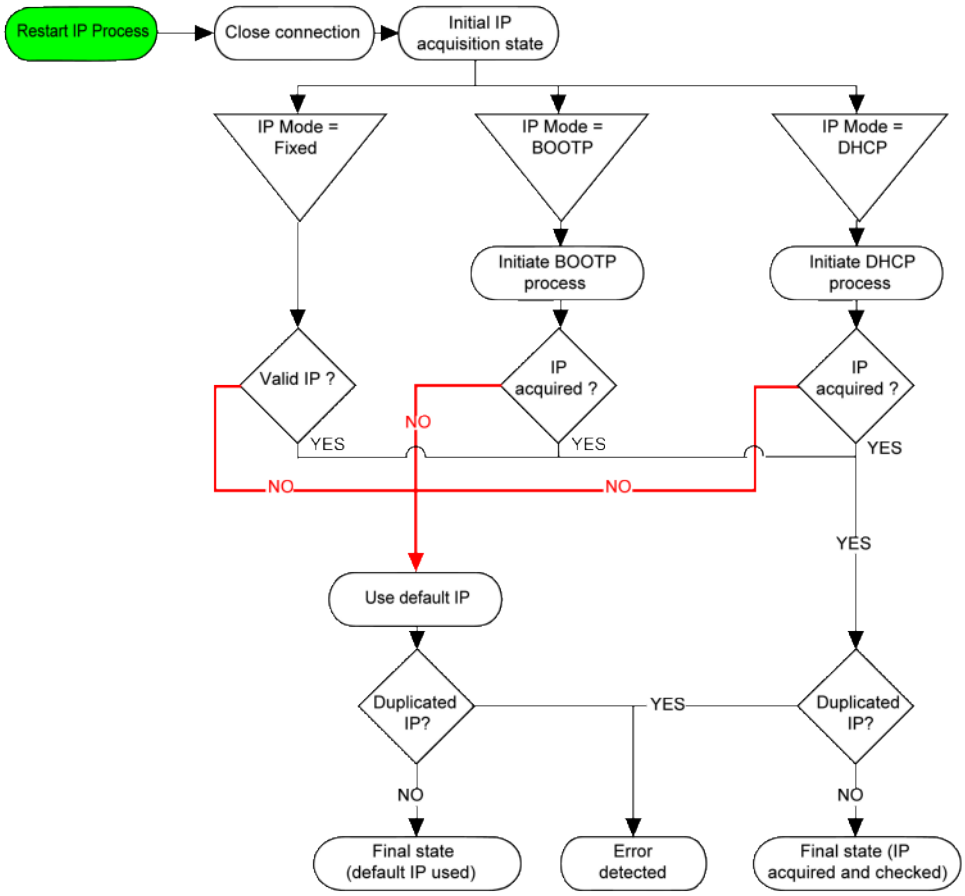
- Verify that there is only one master controller configured on the network or remote link.
- Verify that all devices have unique addresses.
- Obtain your IP address from your system administrator.
- Confirm that the IP address of the device is unique before placing the system into service.
- Do not assign the same IP address to any other equipment on the network.
- Update the IP address after cloning any application that includes Ethernet communications to a unique address.

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

NOTE: Verify that your system administrator maintains a record of assigned IP addresses on the network and subnetwork, and inform the system administrator of any configuration changes performed.

Address Management

This diagram shows the different types of address systems for the controller:



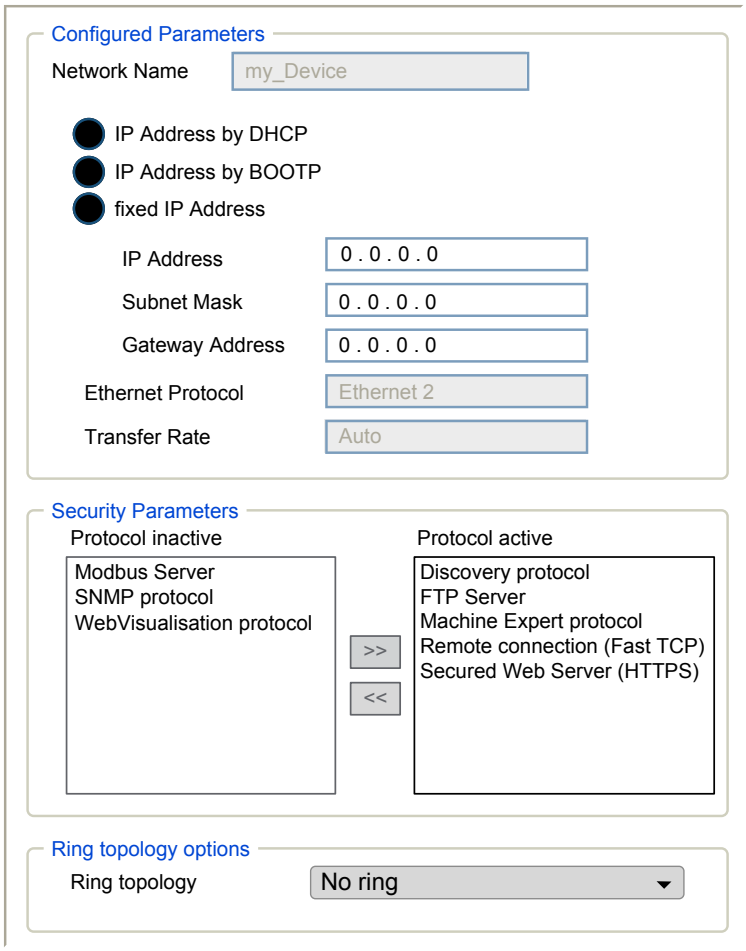
NOTE: If a device programmed to use the DHCP or BOOTP addressing methods is unable to contact its respective server, the controller uses the default IP address. It repeats its request constantly.

The IP process restarts in the following cases:

- Controller reboot
- Ethernet cable reconnection
- Application download (if IP parameters change)
- DHCP or BOOTP server detected after a prior addressing attempt was unsuccessful.

Ethernet Configuration

In the **Devices tree**, double-click **Ethernet_1** or **Ethernet_2**:



NOTE:

- If you are in offline mode, you see the **Configured Parameters** window and, for **Ethernet_2** the **Ring topology options** window. You can edit them.
- If you are in online mode, you see the **Configured Parameters** and **Current Settings** windows. You cannot edit them.

This table describes the configured parameters:

Configured Parameters	Description
Interface Name	Name of the network link. Visible in online mode.
Network Name	Used as device name to retrieve IP address through DHCP, maximum 15 characters. **NOTE:** The network name modification is applied at next power ON.
IP Address by DHCP	IP address is obtained by DHCP server.
IP Address by BOOTP	IP address is obtained by BOOTP server. MAC address is located on the front of the controller.
Fixed IP Address	IP address, Subnet Mask, and Gateway Address are defined by the user.
Ethernet Protocol	Protocol type used: Ethernet 2
Transfer Rate	Speed and Duplex are in auto-negotiation mode.

Default IP Address

The default IP addresses are:

- 10.10.x.y. for Ethernet_1
- 10.11.x.y. for Ethernet_2

When TM262• is not configured, TMSES4 boots and automatically gets its default IP address:

- 10.12.x.z for the first module
- 10.13.x.z for the second module
- 10.14.x.z for the third module

x represents the 5th and y or z represent the 6th bytes of interface MAC address. For example, with a MAC address of 00:80:F4:4E:02:5D, the IP address will be 10.12.2.93

NOTE: The IP addresses must not be in the same IP network.

The MAC address of the Ethernet port can be retrieved on the label placed on the front side of the controller. The MAC address of the TMSES4 port can be calculated with the controller MAC address port Ethernet_2.

NOTE: For EcoStruxure Machine Expert versions prior to V1.2.4, the MAC address is determined by the value on the left side of the controller. See the Compatibility and Migration User Guide.

The default subnet masks are:

- 255.255.0.0 for Ethernet_1
- 255.255.0.0 for Ethernet_2

NOTE: A MAC address is written in hexadecimal format and an IP address in decimal format. Convert the MAC address to decimal format.

Example of conversion:

Port	MAC address	IP address
Ethernet_1	**MAC@Eth1**:00.80.F4.4E.24.10	10.10.36.16
Ethernet_2	**MAC@Eth2**:00.80.F4.4E.24.0B	10.11.36.11
TMS_1	**MAC@TMS**:00.80.F4.50.24.0B	10.12.36.11
TMS_2	**MAC@TMS**:00.80.F4.50.24.0C	10.13.36.12
TMS_3	**MAC@TMS**:00.80.F4.50.24.0D	10.14.36.13

NOTE: The TMSES4 MAC address is calculated as follows: *MAC@TMS_x = MAC@Ethernet2 + 0x20000 + (x-1)*.

Prohibited IP Addresses

USB Network address (192.168.200.0) and TMS Network address (192.168.2.0) are prohibited.

Subnet Mask

The subnet mask is used to address several physical networks with a single network address. The mask is used to separate the subnetwork and the device address in the host ID.

The subnet address is obtained by retaining the bits of the IP address that correspond to the positions of the mask containing 1, and replacing the others with 0.

Conversely, the subnet address of the host device is obtained by retaining the bits of the IP address that correspond to the positions of the mask containing 0, and replacing the others with 1.

Example of a subnet address:

IP address	192 (11000000)	1 (00000001)	17 (00010001)	11 (00001011)
Subnet mask	255 (11111111)	255 (11111111)	240 (11110000)	0 (00000000)
Subnet address	192 (11000000)	1 (00000001)	16 (00010000)	0 (00000000)

NOTE: The device can communicate only on its subnetwork when there is no gateway.

Gateway Address

The gateway allows a message to be routed to a device that is not on the same network.

If there is no gateway, the gateway address is 0.0.0.0.

The default gateway address can be defined on any interface. You can only configure the default gateway on one interface. The traffic to unknown networks is sent through this interface. Please refer to IP Routing, page 81 if you need to configure more than one interface.

Security Parameters

This table describes the different security parameters:

Security Parameters	Description	Default settings
Discovery protocol	This parameter activates/deactivates **Discovery protocol**. When deactivated, Discovery requests are ignored.	Active
FTP Server	This parameter activates/deactivates the FTP server of the controller. When deactivated, FTP requests are ignored.	Active
CoDeSys protocol	This parameter activates/deactivates the CoDeSys protocol on Ethernet interfaces. When deactivated, any programming software request is rejected. Therefore, no connection is possible on Ethernet from a programming PC, from an HMI target that wants to exchange variables with this controller, from an OPC server, or from Controller Assistant.	Active
Modbus Server	This parameter activates/deactivates the Modbus server of the controller. When deactivated, any Modbus request to the controller is ignored.	Inactive
Remote connection (Fast TCP)	This parameter activates/deactivates the remote connection. When deactivated, Fast TCP requests are ignored.	Active
Secured Web Server (HTTPS)	This parameter activates/deactivates the secured Web server of the controller. When deactivated, HTTPS requests to the controller secured Web server are ignored.	Active
SNMP protocol	This parameter activates/deactivates the SNMP server of the controller. When deactivated, SNMP requests are ignored.	Inactive
WebVisualisation protocol	This parameter activates/deactivates the WebVisualisation pages of the controller. When deactivated, HTTP requests to the controller WebVisualisation protocol are ignored.	Inactive

Ring Topology Options

This parameter is only available on the Ethernet_2 network.

This table describes the **Ring topology** options:

Options	Description
No ring	If selected, verify that no ring is wired.
Root	First device of the ring topology.
Participant	One of the devices in the ring topology.

Each device in the ring topology must support the Rapid Spanning Tree Protocol (RSTP).

You can have up to 40 devices in the ring topology.

NOTE: For a network topology that has RSTP enabled, verify that the RPI/timeout combination respects the minimum convergence time of 100 ms that is required for RSTP.

Modbus TCP Client/Server

Introduction

Unlike Modbus serial link, Modbus TCP is not based on a hierarchical structure, but on a client/server model.

The Modicon M262 Logic/Motion Controller implements both client and server services so that it can initiate communications to other controllers and I/O devices, and respond to requests from other controllers, SCADA, HMIs, and other devices.

Without any configuration, the embedded Ethernet port of the controller supports Modbus server.

The Modbus client/server is included in the firmware and does not require any programming action from the user. Due to this feature, it is accessible in RUNNING, STOPPED and EMPTY states.

Modbus TCP Client

The Modbus TCP client supports the following function blocks from the PLCCommunication library without any configuration:

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

For further information, refer to the Function Block Descriptions (see Modbus and ASCII Read/Write Functions, PLCCommunication Library Guide).

Modbus TCP Server

The Modbus server supports the Modbus requests:

Function Code Dec (Hex)	Subfunction Dec (Hex)	Function
1 (1)	–	Read digital outputs (%Q)
2 (2)	–	Read digital inputs (%I)
3 (3)	–	Read holding register (%MW)
6 (6)	–	Write single register (%MW)
8 (8)	–	Diagnostic
15 (F)	–	Write multiple digital outputs (%Q)
16 (10)	–	Write multiple registers (%MW)
23 (17)	–	Read/write multiple registers (%MW)
43 (2B)	14 (E)	Read device identification

Diagnostic Request

This table contains the data selection code list:

Data Selection Code (hex)	Description
00	Reserved
01	Basic Network Diagnostics
02	Ethernet Port Diagnostic
03	Modbus TCP/Port 502 Diagnostics
04	Modbus TCP/Port 502 Connection Table
05 - 7E	Reserved for other public codes
7F	Data Structure Offsets

FTP Server

Introduction

Any FTP client installed on a computer that is connected to the controller (Ethernet port), without EcoStruxure Machine Expert installed, can be used to transfer files to and from the data storage area of the controller.

NOTE: When transferring text files to the controller, switch your FTP client into binary transfer type to avoid conversions of Carriage Return and LineFeed character pairs into a single LineFeed.

NOTE: Schneider Electric adheres to industry best practices in the development and implementation of control systems. This includes a "Defense-in-Depth" approach to secure an Industrial Control System. This approach places the controllers behind one or more firewalls to restrict access to authorized personnel and protocols only.

⚠ WARNING

UNAUTHENTICATED ACCESS AND SUBSEQUENT UNAUTHORIZED MACHINE OPERATION

- Evaluate whether your application environments are connected to your critical infrastructure and, if so, take appropriate steps in terms of prevention, based on Defense-in-Depth, before connecting the automation system to any network.
- Limit the number of devices connected to a network to the minimum necessary.
- Isolate your industrial network from other networks inside your company.
- Protect any network against unintended access by using firewalls, VPN, or other, proven security measures, such as an Intrusion Prevention System or Intrusion Detection System.
- Monitor activities within your systems.
- Prevent subject devices from direct access or direct link by unauthorized parties or unauthenticated actions.
- Install certificates that are issued by publicly known Trusted Certificate Authorities.
- Keep your systems up-to-date and rely only on legitimate sources.
- Prepare a recovery plan including backup of your system and process information.

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

For more information on organizational measures and rules covering access to infrastructures, refer to ISO/IEC 27000 series, Common Criteria for Information Technology Security Evaluation, ISO/IEC 15408, IEC 62351, ISA/IEC 62443, NIST Cybersecurity Framework, Information Security Forum - Standard of Good Practice for Information Security and refer to Cybersecurity Guidelines for EcoStruxure Machine Expert, Modicon and PacDrive Controllers and Associated Equipment.

NOTE: Make use of the security-related commands (see Menu Commands, Online Help) which provide a way to add, edit, and remove a user in the online user management of the target device where you are currently logged in.

FTP Access

Access to the FTP server is controlled by User Rights when they are enabled in the controller. For more information, refer to **Users Rights** description, page 86.

To access the FTP server you must first connect to the controller with EcoStruxure Machine Expert or Controller Assistant, and activate the user rights or create the user for the first login.

NOTE: FTPS (explicit over TLS FTP) is set by default. Simple FTP (non secure) access is not possible at first connection. Set the parameter 1106 to 0 in the post configuration and reboot the controller to allow Simple FTP connection.

FTP Client

The M262 Logic/Motion Controller supports an FTP client library to allow you to query FTP servers. For details, refer to the FtpRemoteFileHandling Library Guide.

Files Access

See File Organization, page 40.

Declaring the Connection to the FTP Server as Trusted

In the context of TLS, certificates can be used to verify the identity of the communication partners. Certificates are sent during the establishment of a connection, referred to as the TLS handshake.

The sending of the certificate is optional for the client, unless the server requests the client certificate. The server sends its certificate for every connection attempt. A connection with the communication partner can be established only if the result of the verification of the certificate is positive.

For secured FTP connections, the FTP server certificate must be provided to the M262 Logic/Motion Controller in the */pki/trusted/certs* folder of the controller file system. The procedures for self-signed and signed certificates are found hereafter.

Procedure for Self-Signed FTP Server Certificates

On the first connection attempt to the FTP server, the self-signed server certificate is stored in the */pki/untrusted* directory of the controller. The following procedure describes how to move the certificate to the */pki/trusted/certs* directory using the **Files** tab of the device editor in the Logic Builder:

Step	Action
1	Login to your controller from your EcoStruxure Machine Expert FTP client application.
2	Select the **Files** tab of the device editor and click the **Refresh** icon.
3	In the right part of the device editor that displays the **Runtime** system, navigate to the */pki/untrusted* directory and select the certificate (.crt) file of your FTP server.
4	In the left part of the device editor that displays the **Host**, select a temporary location and transfer the certificate by clicking the **<<** arrow button in the center of the device editor.
5	In the right part of the device editor, navigate to the */pki/trusted/certs* directory.
6	Select the certificate in the left part of the device editor and transfer it from the temporary location to the */pki/trusted/certs* directory by clicking the **>>** arrow button in the center of the device editor.

Procedure for Signed FTP Server Certificates

If the FTP server certificate is signed by a Certificate Authority (CA), you can save the root or an intermediate CA certificate to the */pki/trusted/certs* directory of the controller before the first connection attempt to the FTP server. Proceed as follows:

Step	Action
1	Login to your controller from your EcoStruxure Machine Expert FTP client application.
2	Select the **Files** tab of the device editor and click the **Refresh** icon.
3	In the right part of the device editor, navigate to the */pki/trusted/certs* directory.
4	Select the CA certificate in the left part of the device editor and transfer it from the temporary location to the */pki/trusted/certs* directory by clicking the **>>** arrow button in the center of the device editor.

SNMP

Introduction

The Simple Network Management Protocol (SNMP) is used to provide the data and services required for managing a network.

The data is stored in a Management Information Base (MIB). The SNMP protocol is used to read or write MIB data. Implementation of the Ethernet SNMP services is minimal, as only the compulsory objects are handled.

SNMP Server

This table presents the supported standard MIB-2 server objects:

Object	Description	Access	Value
sysDescr	Text description of the device	Read	SCHNEIDER M262 Fast Ethernet TCP/IP
sysName	Node administrative name	Read/ Write	Controller reference

The size of these character strings is limited to 50 characters.

The values written are saved to the controller via SNMP client tool software. The Schneider Electric software for this is ConneXview. ConneXview is not supplied with the controller or bus coupler. For more details, refer to www.se.com.

The M262 Logic/Motion Controller supports the following SNMP protocol versions when acting as a server: v1, v2c, and v3 with user-based security model.

To configure an SNMP server, you must enable the SNMP protocol in EcoStruxure Machine Expert for the interface you want to use. By default, the SNMP v3 protocol is selected. To select another version of SNMP protocol, use the Maintenance: SNMP submenu, page 152 of the Web server.

SNMP Client

The M262 Logic/Motion Controller supports an SNMP client library to allow you to query SNMP servers. For details, refer to the SnmpManager Library Guide.

Web Server

Introduction

The controller provides an embedded Web server with a predefined, built-in website. You can use the website for module setup and control, as well as application diagnostics and monitoring. These pages are ready for use with a web browser on Windows PCs or mobile devices. No configuration or programming is required.

The Web server can be accessed by the web browsers listed below:

- Google Chrome (version 138 or greater)
- Mozilla Firefox (version 140 or greater)
- Microsoft Edge (version 138 or greater)

The Web server can be accessed by the mobile device web browsers listed below:

- iOS Safari
- Android Chrome

HTTP (non secured connections) requests are redirected to HTTPS (secure connections).

The Web server is limited to 10 concurrent users, page 122.

The Web server has access to your application for reading and writing data and controlling the state of the controller. By enabling the Web server, you enable these functions. You can disable the Web server on an interface by deselecting the Web server active parameter in the Ethernet Configuration tab, page 125.

If there are cybersecurity concerns over these functions, you must, at a minimum, assign a secure password to the Web server or disable the Web server to help prevent unauthorized access to the application.

The Web server allows you to remotely monitor a controller and its application, and to perform various maintenance activities including modifications to data and configuration parameters, and change the state of the controller. Ensure that the immediate physical environment of the machine and process is in a state that does not present risks to people or property before exercising control remotely.

⚠ WARNING

UNINTENDED EQUIPMENT OPERATION

- Configure and install the RUN/STOP input for the application, if available for your particular controller, so that local control over the starting or stopping of the controller can be maintained regardless of the remote commands sent to the controller.
- Define a secure password for the Web server and do not allow unauthorized or otherwise unqualified personnel to use this feature.
- Ensure that there is a local, competent, and qualified observer present when operating on the controller from a remote location.
- You must have a complete understanding of the application and the machine/process it is controlling before attempting to adjust data, stopping an application that is operating, or starting the controller remotely.
- Take the precautions necessary to assure that you are operating on the intended controller by having clear, identifying documentation within the controller application and its remote connection.

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

Web Server Access

Access to the Web server is controlled by User Rights when they are enabled in the controller. For more information, refer to **Users and Groups**, page 76.

To access the Web server you must first connect to the controller with EcoStruxure Machine Expert or Controller Assistant, and modify the default user password.

⚠ WARNING

UNAUTHORIZED DATA ACCESS

- Secure access to the FTP/Web server using User Rights.
- If you disable User Rights, disable the FTP/Web server to prevent any unwanted or unauthorized access to data in your application.

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

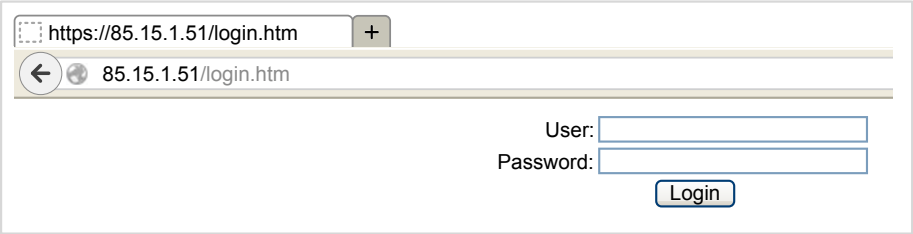
In order to change the password, go to **Users and Groups** tab of the device editor. For more information, refer to the EcoStruxure Machine Expert Programming Guide.

NOTE: The only way to gain access to a controller that has user access-rights enabled and for which you do not have the password(s) is by performing an Update Firmware operation. This clearing of User Rights can only be accomplished by using a SD card to update the controller firmware. In addition, you may clear the User Rights in the controller by running a script (refer to Reset the User Rights to Default, page 249). This effectively removes the existing application from the controller memory, but restores the ability to access the controller.

Home Page Access

To access the website home page, type the IP address of the controller into the browser.

This figure shows the Web server site login page:



This figure shows the home page of the Web server site once you have logged in:

NOTE: Schneider Electric adheres to industry best practices in the development and implementation of control systems. This includes a "Defense-in-Depth" approach to secure an Industrial Control System. This approach places the controllers behind one or more firewalls to restrict access to authorized personnel and protocols only.

⚠ WARNING

UNAUTHENTICATED ACCESS AND SUBSEQUENT UNAUTHORIZED MACHINE OPERATION

- Evaluate whether your application environments are connected to your critical infrastructure and, if so, take appropriate steps in terms of prevention, based on Defense-in-Depth, before connecting the automation system to any network.
- Limit the number of devices connected to a network to the minimum necessary.
- Isolate your industrial network from other networks inside your company.
- Protect any network against unintended access by using firewalls, VPN, or other, proven security measures, such as an Intrusion Prevention System or Intrusion Detection System.
- Monitor activities within your systems.
- Prevent subject devices from direct access or direct link by unauthorized parties or unauthenticated actions.
- Install certificates that are issued by publicly known Trusted Certificate Authorities.
- Keep your systems up-to-date and rely only on legitimate sources.
- Prepare a recovery plan including backup of your system and process information.

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

For more information on organizational measures and rules covering access to infrastructures, refer to ISO/IEC 27000 series, Common Criteria for Information Technology Security Evaluation, ISO/IEC 15408, IEC 62351, ISA/IEC 62443, NIST Cybersecurity Framework, Information Security Forum - Standard of Good Practice for Information Security and refer to Cybersecurity Guidelines for EcoStruxure Machine Expert, Modicon and PacDrive Controllers and Associated Equipment.

Home Page Access Menu

The Home Page Access menu bar allows you to access the main Web server pages.

The Web server contains the following pages:

Menu	Page	Description
Home	Home, page 134	Home page of the controller Web server page. Provides access to the tabs: • Monitoring, page 136 • Diagnostics, page 138 • Maintenance, page 144 • Machine Assistant, page 153

Home page menu descriptions:

Menu	Submenu	Description
Monitoring	Data Parameters, page 136	Allows you to display and modify controller variables.
	IO Viewer, page 137	Shows the module with module I/O values.
	Oscilloscope, page 138	Displays 2 variables in the form of a recorder-type time chart.
Diagnostics	Controller, page 138	Displays controller status.
	Ethernet, page 139	Displays Ethernet diagnostic.
	TM3 Expansion, page 140	Displays expansion module status.
	TMS Expansion, page 141	Displays expansion module status.
	TMSES4, page 142	Displays TMSES4 status.
	Scanner Status, page 142	Displays serial line status.
	EtherNet/IP Status, page 143	Displays Ethernet status.
	CANopen, page 143	Displays CANopen statistics.
Maintenance	Post Conf, page 144	Allows you to access the post configuration file saved on the controller.
	User Management, page 144	Allows you to change user password and customize login message. Possible in secure mode (HTTPS) only. • **Change password (of current user)**: allows you to change user password. • **Users account management**: allows you to remove all passwords from the controller and reset user accounts to their default state. • **Clone management**: allows you to include or exclude user access rights when cloning a controller. • **System use notification**: allows you to customize a message which will be displayed at login.
	Firewall, page 146	Allows you to modify the firewall configuration.
	System Log Files, page 147	Allows you to access log files generated by the controller.
	Message Logger, page 147	Allows you to access controller messages.
	Run/Stop Controller, page 148	Allows you to send Run and Stop commands to the controller.
	SelfAwareness, page 148	Allows you to access temperature, memory usage, processor load and devices information.
	Certificates, page 149	Allows you to customize certificates owned by a Modicon M262 Logic/ Motion Controller.
	Date / Time, page 149	Allows you to set the date, time, time zone and optional daylight saving time.
	SCEP, page 150	Allows you to access the configuration of the SCEP server.
	SNMP, page 152	Allows you to choose the SNMP protocol version and add/delete user accounts.
Machine Assistant	List View	Displays the configuration in list view.
	Graphic View	Displays the configuration in graphic view.
	Scan, page 277	Allows you to scan the devices configured.
	Clear, page 277	Allows you to clear the scan.
	Load .semdt file, page 280	Allows you to upload a .semdt file after scan.
	Export scan results, page 280	Allows you to export the scan results in your local SD Card.

Monitoring Menu

Monitoring: Data Parameters

Monitoring Web Server Variables

To monitor Web server variables, you must select the variables in the Symbol Configuration Editor, page 97.

Monitoring: Data Parameters Submenu

The **Data Parameters** submenu allows you to display and modify variable values:

Element	Description
Add	Adds a list description or a variable
Del	Deletes a list description or a variable
Refresh period	Refreshing period of the variables contained in the list description (in ms)
Refresh	Enables I/O refreshing: • Gray button: refreshing disabled • Orange button: refreshing enabled **NOTE:** Without enabling **Refresh**, when a variable's value is changed in the table the modification is directly sent to the controller.
Load	Loads saved lists from the controller internal Flash to the Web server page
Save	Saves the selected list description in the controller (*/usr/web* directory)

NOTE: The IEC objects (`%MX`, `%IX`, `%QX`) are not directly accessible. To access IEC objects you must first group their contents in located registers (refer to Relocation Table, page 45).

Monitoring: IO Viewer Submenu

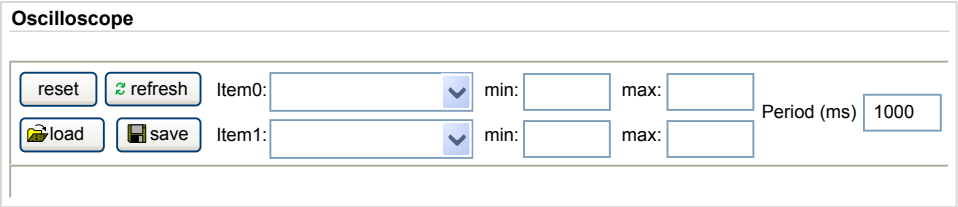
You must add the I/Os in the **Symbol Configuration Editor** to get them visible in the **IO Viewer**. Refer to Symbol Configuration Editor, page 97.

The **IO Viewer** submenu allows you to display the I/O values:

Element	Description
Refresh	Enables I/O refreshing: • Gray button: refreshing disabled • Orange button: refreshing enabled
Period (ms)	I/O refreshing period in ms
<<	Goes to previous I/O list page
>>	Goes to next I/O list page

Monitoring: Oscilloscope Submenu

The **Oscilloscope** submenu can display up to 2 variables in the form of a recorder time chart:

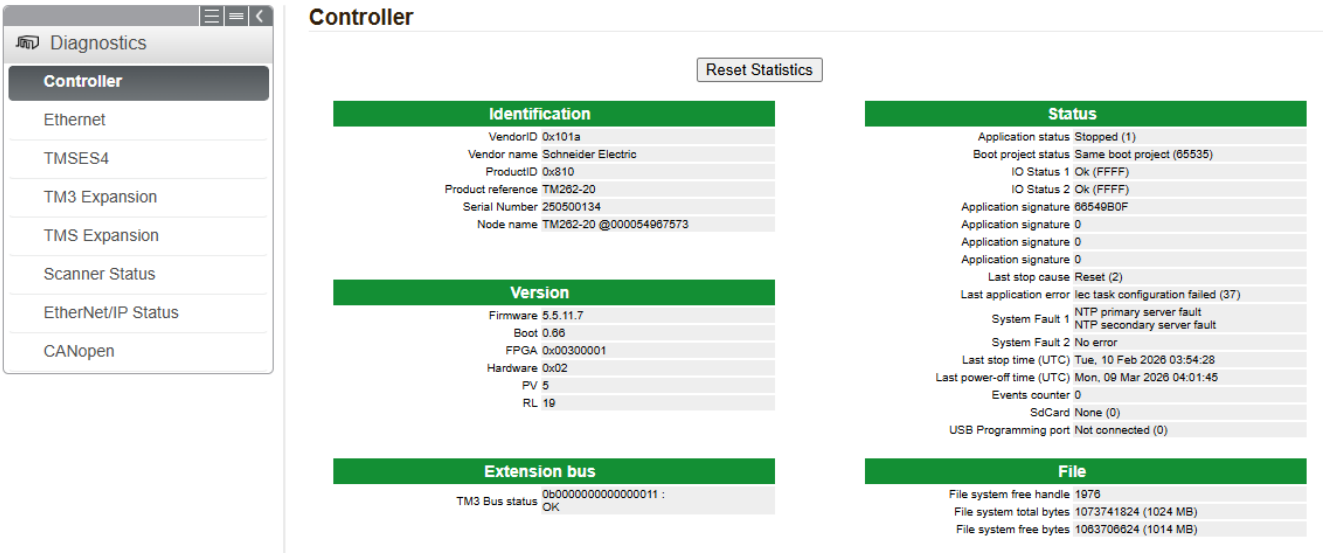


Element	Description
Reset	Erases the memorization
Refresh	Starts/stops refreshing
Load	Loads parameter configuration of Item0 and Item1
Save	Saves parameter configuration of Item0 and Item1 in the controller
Item0	Variable to be displayed
Item1	Variable to be displayed
Min	Minimum value of the variable axis
Max	Maximum value of the variable axis
Period (ms)	Page refresh period in milliseconds

Diagnostics Menu

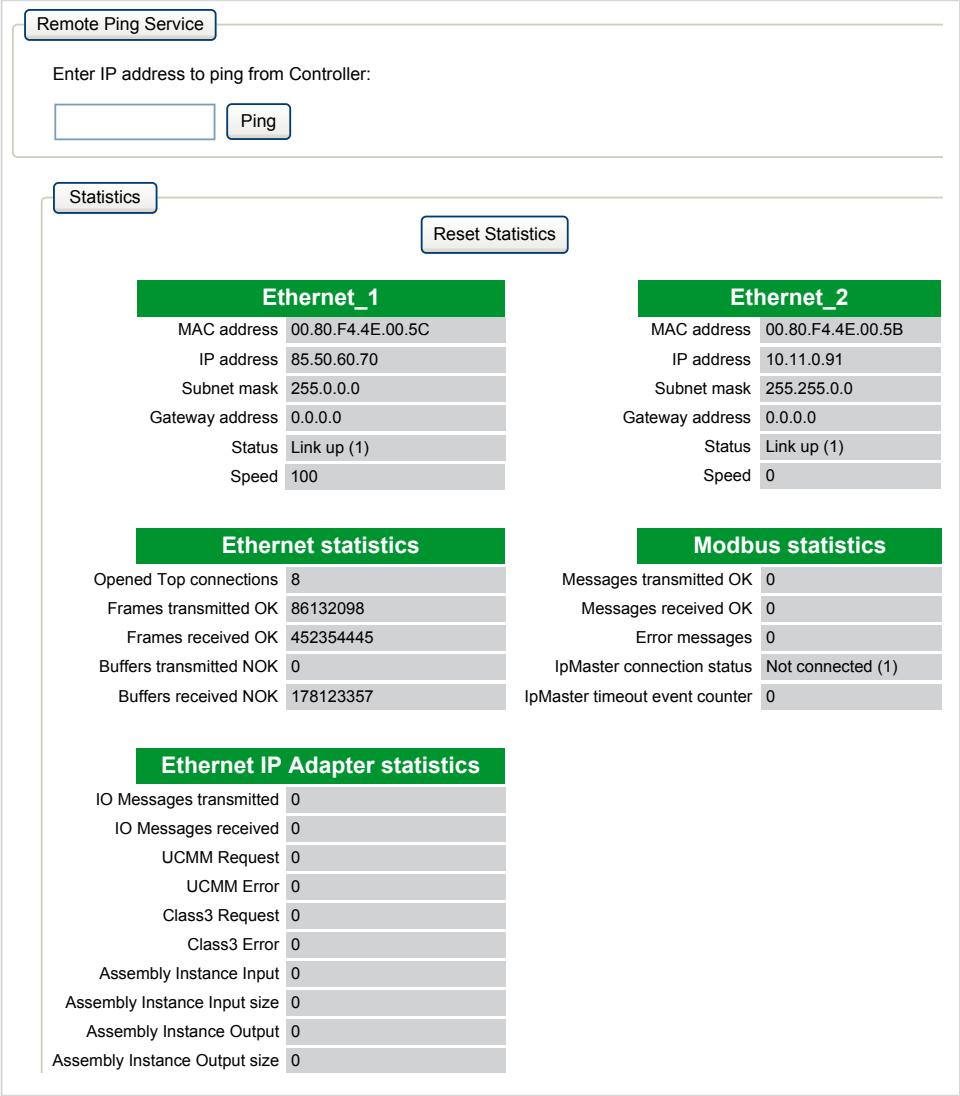
Diagnostics: Controller Submenu

The **Controller** submenu displays information about the controller:



Diagnostics: Ethernet Submenu

The **Ethernet** submenu displays the Ethernet ports status and access to the remote ping service:



Remote Ping Service

Enter IP address to ping from Controller:

Ping

Statistics

Reset Statistics

Ethernet_1	
MAC address	00.80.F4.4E.00.5C
IP address	85.50.60.70
Subnet mask	255.0.0.0
Gateway address	0.0.0.0
Status	Link up (1)
Speed	100

Ethernet_2	
MAC address	00.80.F4.4E.00.5B
IP address	10.11.0.91
Subnet mask	255.255.0.0
Gateway address	0.0.0.0
Status	Link up (1)
Speed	0

Ethernet statistics	
Opened Top connections	8
Frames transmitted OK	86132098
Frames received OK	452354445
Buffers transmitted NOK	0
Buffers received NOK	178123357

Modbus statistics	
Messages transmitted OK	0
Messages received OK	0
Error messages	0
IpMaster connection status	Not connected (1)
IpMaster timeout event counter	0

Ethernet IP Adapter statistics	
IO Messages transmitted	0
IO Messages received	0
UCMM Request	0
UCMM Error	0
Class3 Request	0
Class3 Error	0
Assembly Instance Input	0
Assembly Instance Input size	0
Assembly Instance Output	0
Assembly Instance Output size	0

Diagnostics: TM3 Expansion Submenu

The **TM3 Expansion** submenu shows the TM3 expansion modules status:

Expansion viewer |<< << < 1 - 8 / 14 > >> >>|

Expansion 1	
Module ID	-
Status	Inactive (0)

Expansion 2	
Module ID	-
Status	Inactive (0)

Expansion 3	
Module ID	-
Status	Inactive (0)

Expansion 4	
Module ID	-
Status	Inactive (0)

Expansion 5	
Module ID	-
Status	Inactive (0)

Expansion 6	
Module ID	-
Status	Inactive (0)

Expansion 7	
Module ID	-
Status	Inactive (0)

Expansion 8	
Module ID	-
Status	Inactive (0)

Diagnostics: TMS Expansion Submenu

The **TMS Expansion** submenu shows the TMS expansion modules status:

Expansion viewer |<< << < 1 - 7 / 7 > >> >>|

Expansion 1	
Name	TMSES4
Major type	1
Sub. type	1
Version	1.0.0.3
Module status	Configured (2)
IP status	Ping Success (0)
Pix command status	Disabled (12)

Expansion 2	
Name	
Major type	0
Sub. type	0
Version	
Module status	Discovery (9)
IP status	Not Configured (10)
Pix command status	Disabled (12)

Expansion 3	
Name	
Major type	0
Sub. type	0
Version	
Module status	Discovery (9)
IP status	Not Configured (10)
Pix command status	Disabled (12)

Expansion 4	
Name	
Major type	0
Sub. type	0
Version	
Module status	Discovery (9)
IP status	Not Configured (10)
Pix command status	Disabled (12)

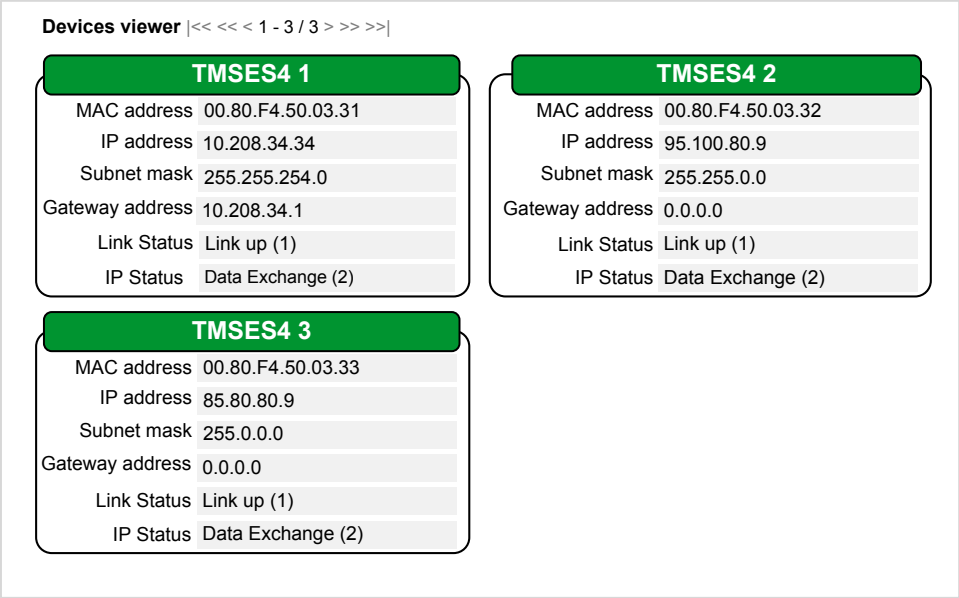
Expansion 5	
Name	
Major type	0
Sub. type	0
Version	
Module status	Discovery (9)
IP status	Not Configured (10)
Pix command status	Disabled (12)

Expansion 6	
Name	
Major type	0
Sub. type	0
Version	
Module status	Discovery (9)
IP status	Not Configured (10)
Pix command status	Disabled (12)

Expansion 7	
Name	
Major type	0
Sub. type	0
Version	
Module status	Discovery (9)
IP status	Not Configured (10)
Pix command status	Disabled (12)

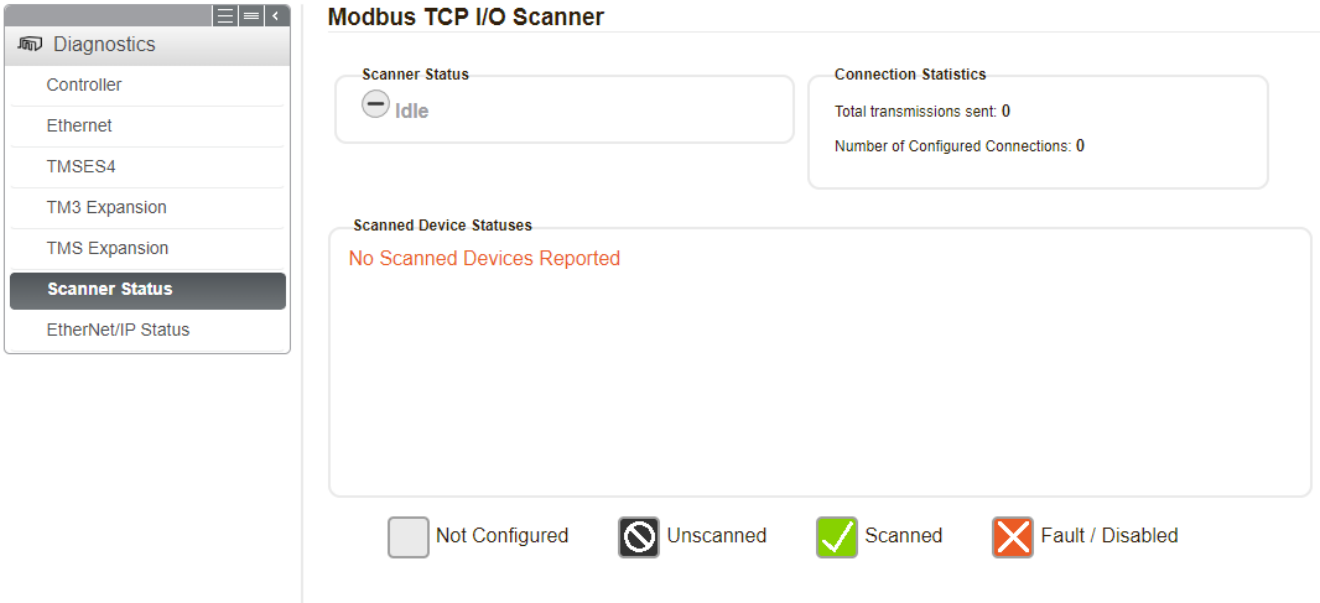
Diagnostics: TMSES4 Submenu

The **TMSES4** submenu shows the status of the TMSES4 Ethernet ports:



Diagnostics: Scanner Status Submenu

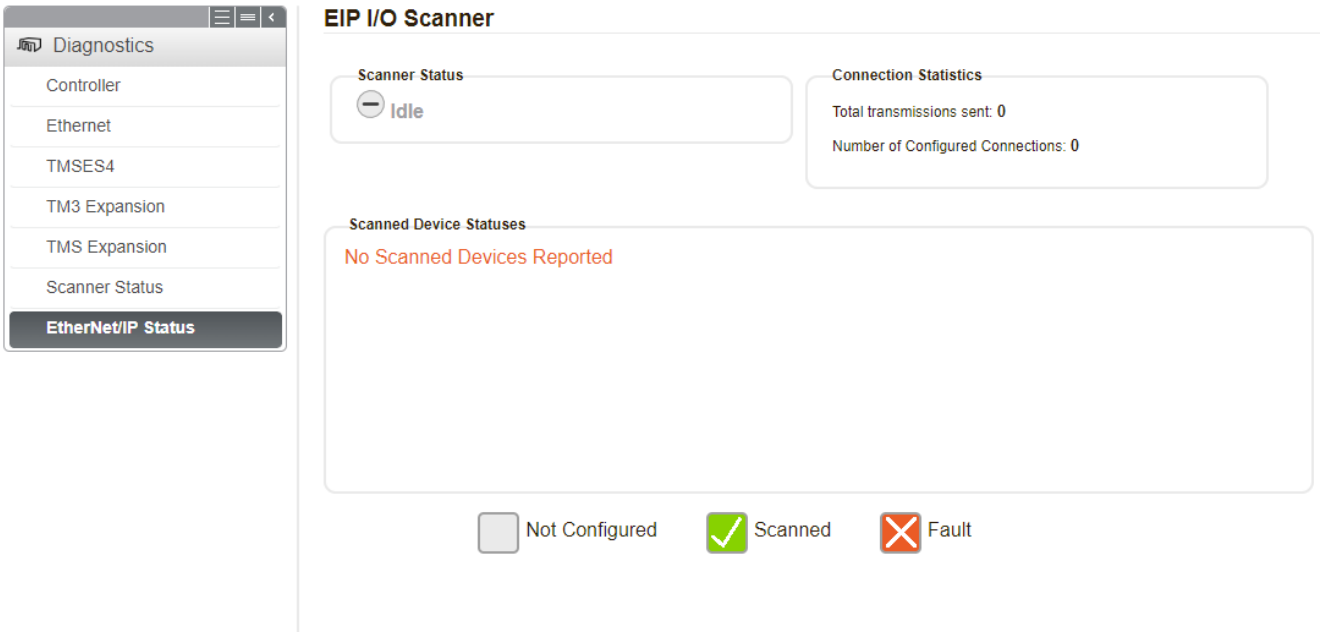
The **Scanner Status** submenu displays status of the Modbus TCP I/O Scanner (IDLE, STOPPED, OPERATIONAL) and the health bit of up to 64 Modbus slave devices:



For more information, refer to Modbus TCP User Guide.

Diagnostics: EtherNet/IP Status Submenu

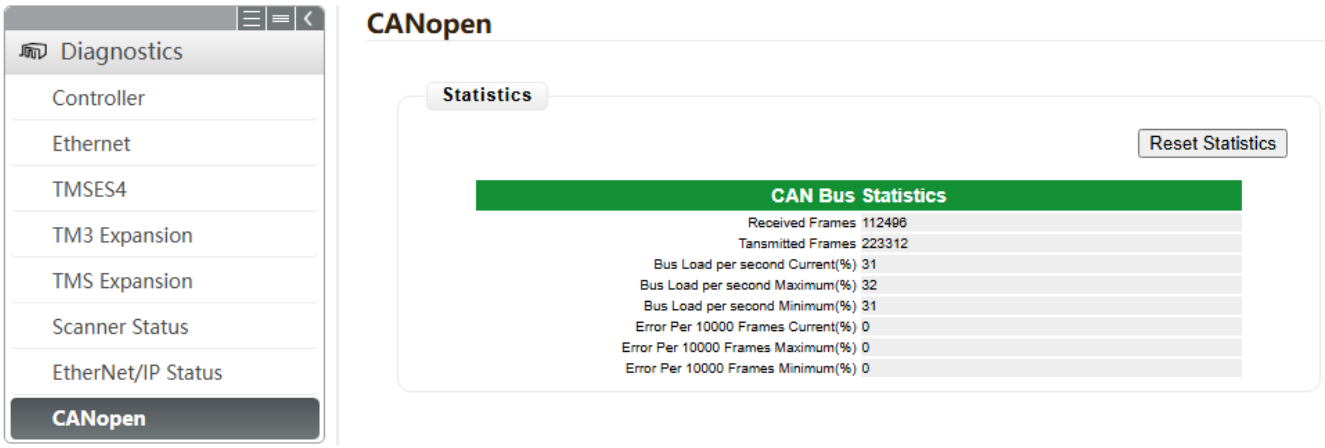
The **EtherNet/IP Status** submenu displays the status of the EtherNet/IP Scanner (IDLE, STOPPED, OPERATIONAL) and the health bit of up to 64 EtherNet/IP target devices:



For more information, refer to EtherNet/IP User Guide.

Diagnostics: CANopen Submenu

The **CANopen** submenu displays information about the CAN bus and the **CANopen** connection:



Element	Description
Received Frames	Number of received CAN frames since CAN connection establishment
Transmitted Frames	Number of transmitted CAN frames since CAN connection establishment
Bus Load per second Current (%)	CAN bus load ratio in percentage during the last second **NOTE:** If the CAN bus load exceeds 50% for at least 10 seconds, a message indicating this event appears in the log messages. This message cannot be triggered again unless the **Reset Statistics** button is clicked.
Bus Load per second Maximum (%)	Maximum CAN bus load ratio in percentage since CAN connection establishment
Bus Load per second Minimum (%)	Minimum CAN bus load ratio in percentage since CAN connection establishment
Error Per 10000 Frames Current (%)	Error frames ratio in percentage during the last 10000 exchanged frames **NOTE:** The value is 0 as long as the number of exchanged frames remains lower than 10000. The value is updated when the number of exchanged frames reaches 10000.
Error Per 10000 Frames Maximum (%)	Maximum error frames ratio per 10000 frames in percentage since CAN connection establishment
Error Per 10000 Frames Minimum (%)	Minimum error frames ratio per 10000 frames in percentage since CAN connection establishment

Maintenance Menu

Introduction

The Maintenance page provides access to the `/usr` folders of the controller non-volatile memory, page 40 and information for device maintenance purposes.

Maintenance: Post Conf Submenu

The **Post Conf** submenu allows you to update the post configuration, page 229 file saved on the controller:

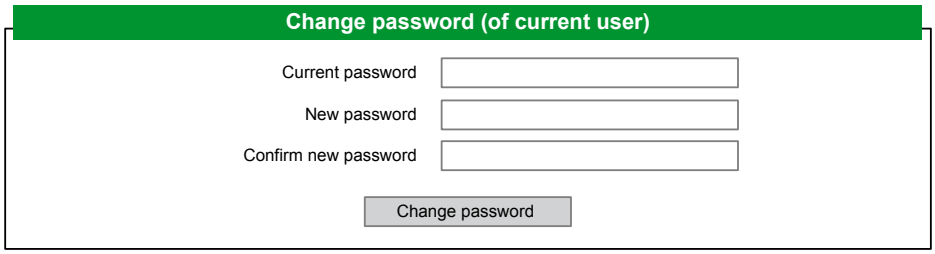
Step	Action
1	Click **Load**.
2	Modify the parameters, page 231.
3	Click **Save**. **NOTE:** The new parameters are considered at the next Post Configuration file reading, page 229.

Maintenance: User Management Submenu

The **User Management** submenu displays a screen that allows you to access four different actions, restricted by using secure protocol (HTTPS):

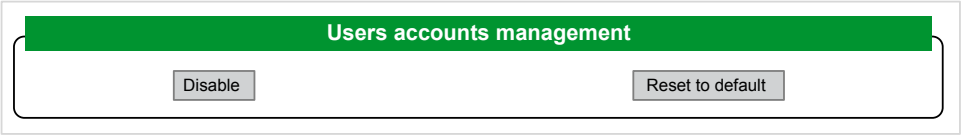
- **Change password (of current user)**:

Allows you to change your password.



- **User accounts management**:

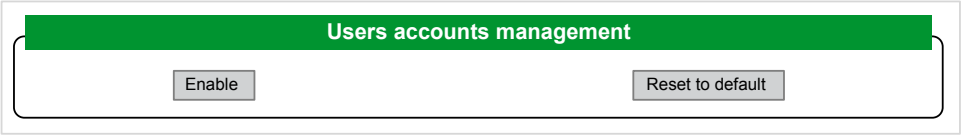
Allows you to manage user accounts management, removing passwords and returning the user accounts on the controller to default settings.



Click **Disable** to deactivate the user rights on the controller. (Passwords are saved and are restored if you click **Enable**). Then, click **OK** on the window that appears to confirm. As a result:

- Users no longer have to set and enter a password to connect to the controller.
- FTP, HTTP, and OPC UA server connections accept anonymous user connections. See Login and passwords table, page 87.

  NOTE: The **Disable** button is only active if the user has administrator privileges.



Click **Enable** to restore the previous user rights saved on the controller. Then, click **OK** on the window that appears to confirm. As a result, users have to enter the password previously set to connect to the controller. See Login and passwords table, page 87.

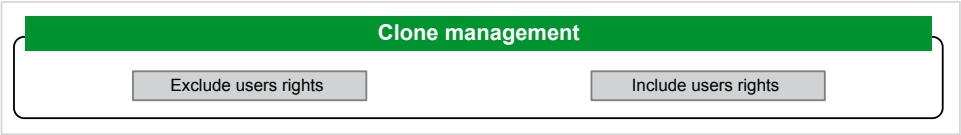
NOTE: The **Enable** button only appears if the user rights are disabled and the user rights backup file is available on the controller.

Click **Reset to default** to return the user accounts on the controller to their default setting state. Then, click **OK** on the window that appears to confirm.

NOTE: Connections to FTP, HTTP, and the OPC UA server are blocked until a new password is set.

- **Clone management**:

Allows you to control whether user rights are copied and applied to the target controller when cloning a controller with an SD card, page 252.



Click **Exclude users rights** to exclude copying user rights to the target controller when cloning a controller.

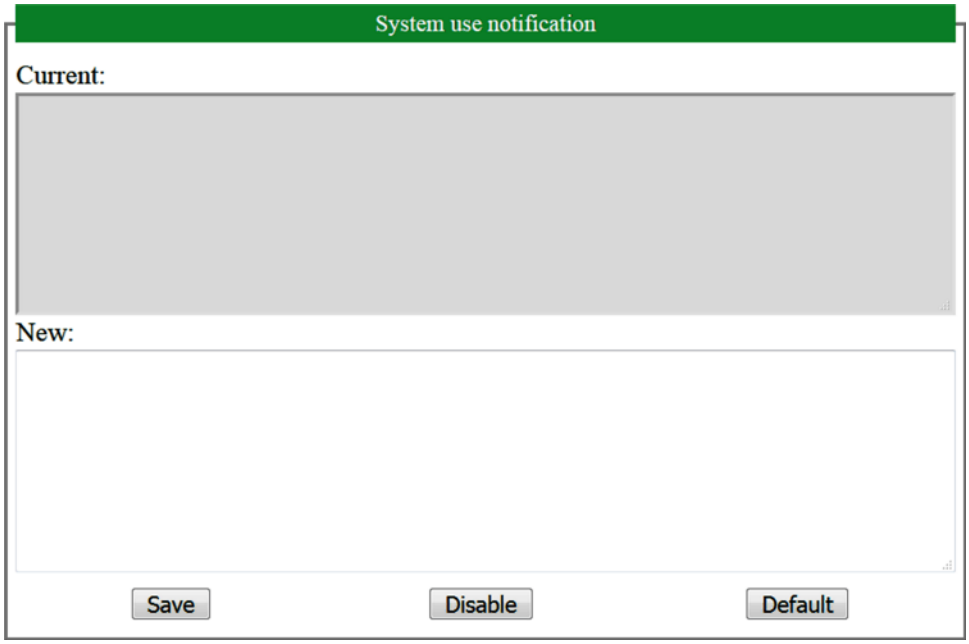
NOTE: By default, the users rights are excluded.

Click **Include users rights** to copy user rights to the target controller when cloning a controller. A popup prompts you to confirm copying the user rights. Click **OK** to continue.

NOTE: The **Exclude users rights** and **Include users rights** buttons are only active if the user is connected to the controller using a secure protocol.

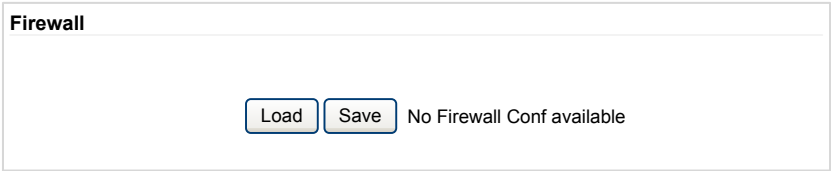
- **System use notification**:

Allows you to customize a message which is displayed at login.



Maintenance: Firewall Submenu

The **Firewall** submenu allows you to modify the default firewall configuration file, page 153:



Maintenance: System Log Files Submenu

The **System Log Files** submenu provides access to log files generated by the controller:

System Log Files

FwLog.txt	8 kb FRI OCT 12 10:51:39 2018
PlcLog_0.txt	104 kB FRI OCT 12 10:46:59 2018
LoggerFile_11-10-2018_02h19m40s.mel	57 kB THU OCT 11 14:19:41 2018
LoggerFile_11-10-2018_04h45m48s.mel	60 kb THU OCT 11 16:45:48 2018
PlcLog_1.txt	104 kB FRI OCT 12 05:12:18 2018
LoggerFile_11-10-2018_04h47m11s.mel	65 kB THU OCT 11 16:47:12 2018
LoggerFile_11-10-2018_06h10m35s.mel	60 kB THU OCT 11 18:10:35 2018
PlcLog_2.txt	104 kB FRI OCT 12 07:27:31 2018
LoggerFile_11-10-2018_07h11m40s.mel	60 kB THU OCT 11 19:11:40 2018
LoggerFile_11-10-2018_09h02m59s.mel	60 kB THU OCT 11 21:02:59 2018
PlcLog.txt	24 kB FRI OCT 12 14:18:56 2018
LoggerFile_11-10-2018_10h14m05s.mel	60 kB THU OCT 11 22:14:05 2018
LoggerFile_12-10-2018_01h28m42s.mel	60 kB FRI OCT 12 01:28:42 2018
LoggerFile_12-10-2018_02h30m44s.mel	60 kB FRI OCT 12 02:30:44 2018
LoggerFile_12-10-2018_05h21m17s.mel	60 kB FRI OCT 12 05:21:17 2018
LoggerFile_12-10-2018_06h23m39s.mel	60 kB FRI OCT 12 06:23:39 2018
LoggerFile_12-10-2018_07h50m10s.mel	60 kB FRI OCT 12 07:50:11 2018
LoggerFile_12-10-2018_08h38m01s.mel	59 kB FRI OCT 12 08:38:01 2018
LoggerFile_12-10-2018_10h36m56s.mel	62 kB FRI OCT 12 10:36:56 2018
LoggerFile_12-10-2018_10h37m19s.mel	64 kB FRI OCT 12 10:37:19 2018
LoggerFile_12-10-2018_10h52m01s.mel	58 kB FRI OCT 12 10:52:01 2018

NOTE: A maximum of 300 log files can be stored in the **Message Logger**. When the maximum log file size is reached, previous logs must be deleted in order to continue saving new diagnostic information.

Maintenance: Message Logger Submenu

The **Message Logger** submenu displays latest controller log messages:

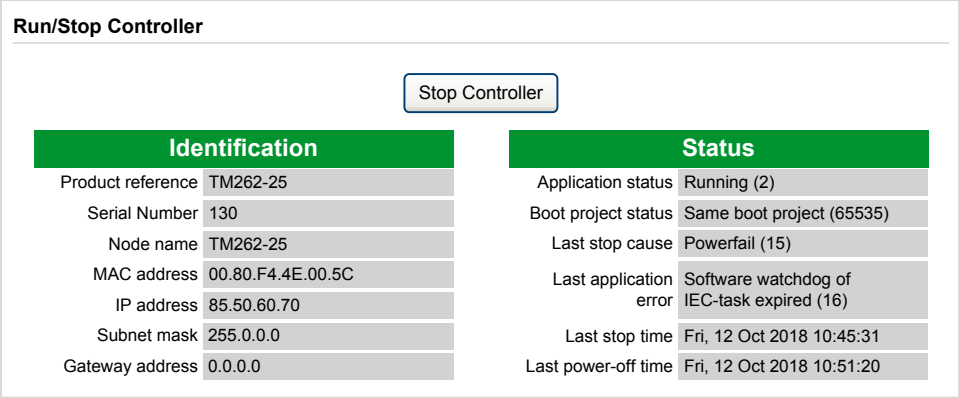
Message Logger

Load Save

No.	Timestamp	Type	Object	Instance	Diag. code	Ext. diagnosis	Message

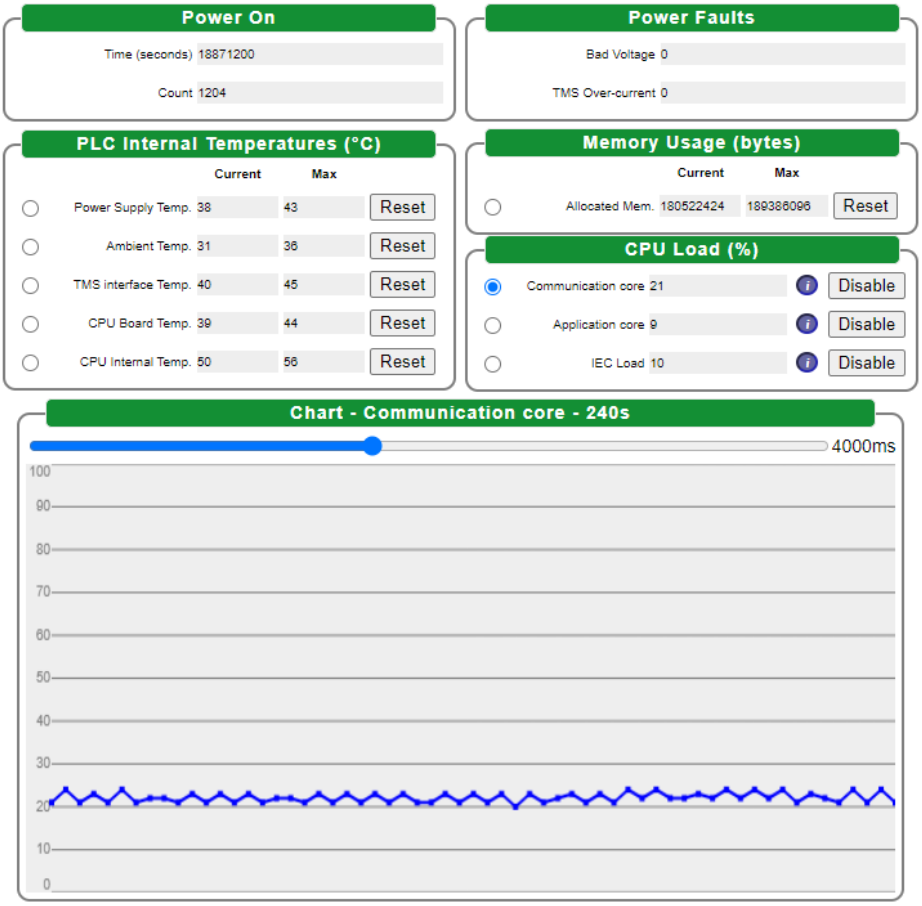
Maintenance: Run/Stop Controller Submenu

The **Run/Stop Controller** submenu allows you to manually stop and restart the controller:



Maintenance: SelfAwareness Submenu

The **SelfAwareness** submenu allows you to access temperature, memory usage, processor load and devices information:

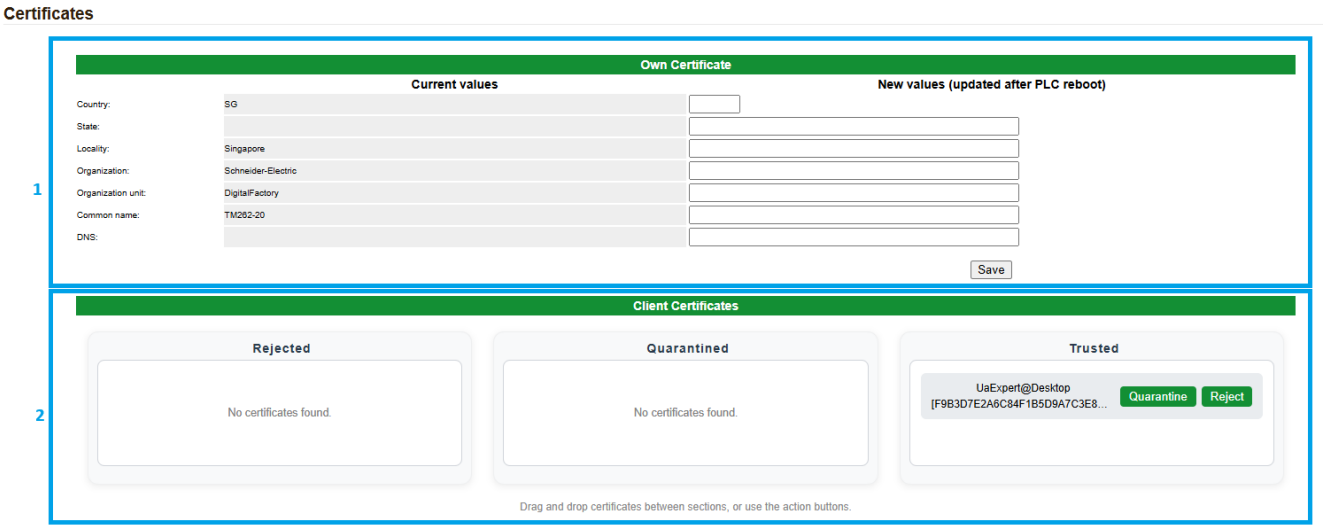


NOTE: The sample rate is set at 4 seconds. Setting under 4 seconds increases the **Communication core** and **CPU Load**.

The controller internal ambient maximum temperature is 100 °C (212 °F). The external ambient maximum temperature can be found in the hardware guide of your controller.

Maintenance: Certificates Submenu

The following graphic shows the **Certificates** submenu:



1: Own Certificate allows you to modify the certificates owned by an M262 Logic/Motion Controller. The optional **DNS** value indicates the domain name for which the certificates is valid (OPC UA or HTTP/FTP).

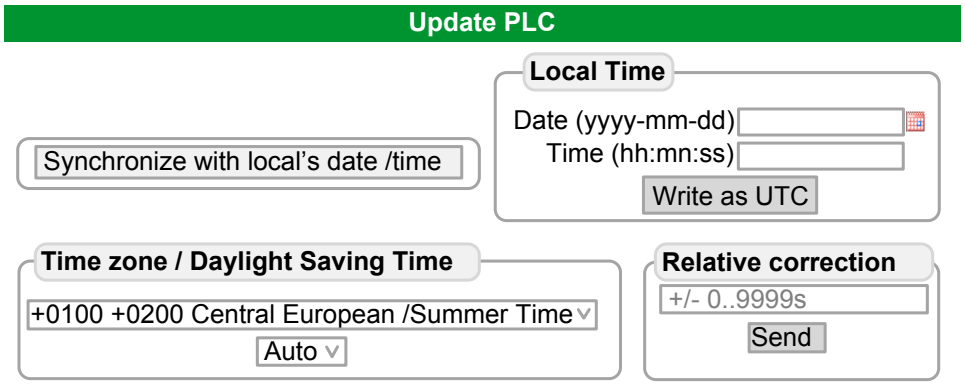
> **NOTE:** Any modification has an impact on OPC UA and HTTP/FTP certificates. See Certificate Management, page 24.
>
> **NOTE:** Any modification overwrites SCEP certificates and requires a new SCEP enrollment. See Maintenance: SCEP Submenu, page 150.

2: Client Certificates allows you to manage certificate trust levels by moving certificates between **Trusted**, **Quarantined** and **Rejected** states.

Maintenance: Date / Time Submenu

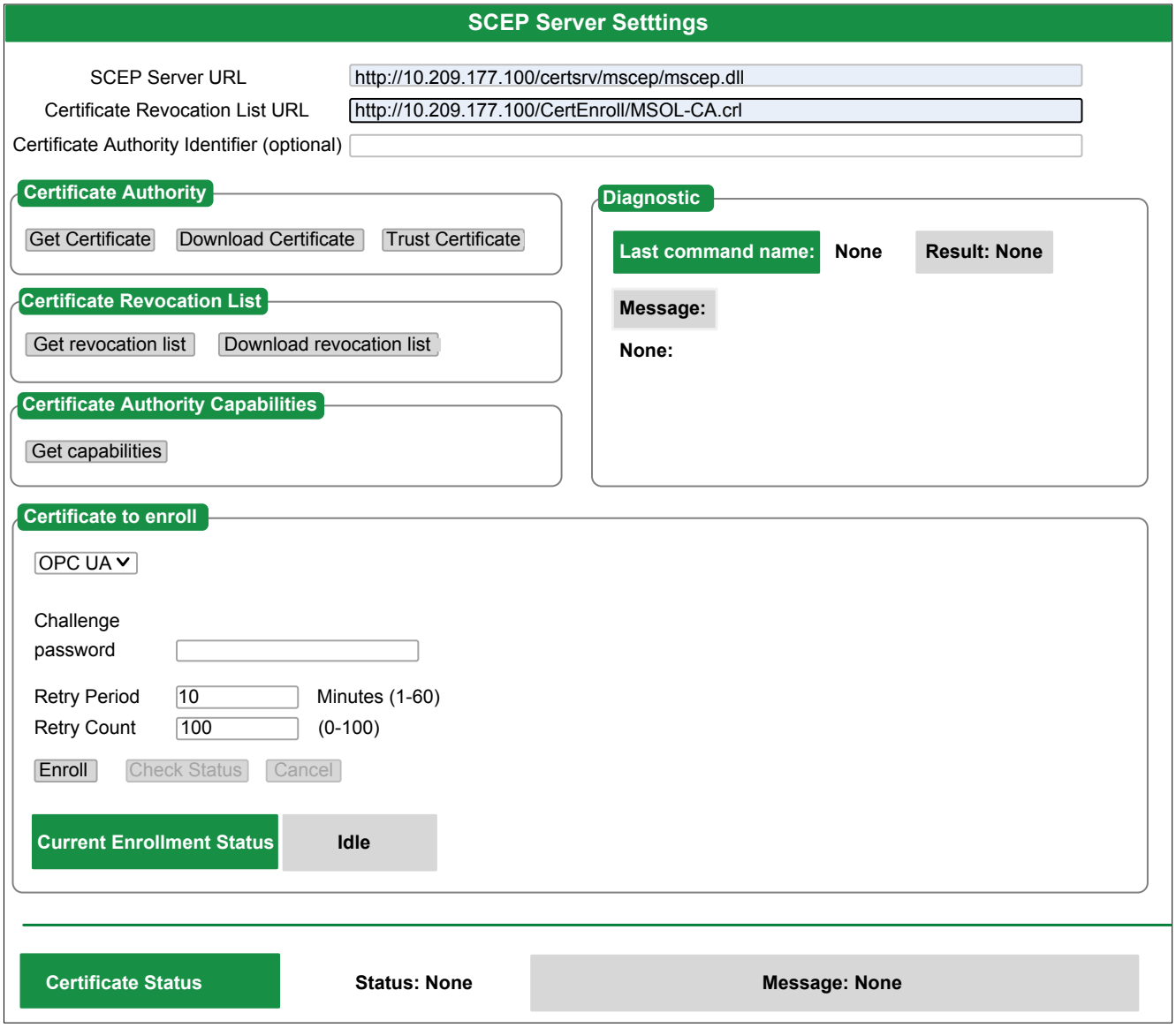
The **Date / Time** submenu displays the date, time, time zone and optional daylight saving time and allows you to manually change them:

Read	
PLC Time	Fri Dec 03 2021 11:10:09 GMT+0100 (CET)
Local Time	Fri Dec 03 2021 11:10:07 GMT+0100 (Romance Standard Time)



Maintenance: SCEP Submenu

The SCEP submenu allows you to communicate with a SCEP server. This section describes how to specify settings that allow the device to obtain certificates from a Certificate Authority (CA) using Simple Certificate Enrollment Protocol (SCEP).



The following table describes the SCEP submenu:

Element	Option	Description
SCEP Server Settings	**SCEP Server URL**	Allows you to specify the URL of the SCEP server to which the device should send certificate requests.
	Certificate Revocation List URL	Allows you to specify the URL of the Certificate Revocation List.
	Certificate Authority Identifier (Optional)	Allows you to choose which certificate is required if a Certificate Authority (CA) has multiple certificates.
Certificate Authority	**Get Certificate**	Allows you to obtain the certificate.
	Download Certificate	Allows you to download the certificate.
	Trust Certificate	Allows you to add the certificate to the trusted list of the device.
Certificate Revocation List	**Get revocation list**	Allows you to obtain the certification revocation list from the Certificate Authority (CA).
	Download revocation list	Displays the content of the Certificate Revocation List (CRL) received.
Certificate Authority Capabilities	**Get capabilities**	Allows you to request which functionality is available from the Certificate Authority (CA).

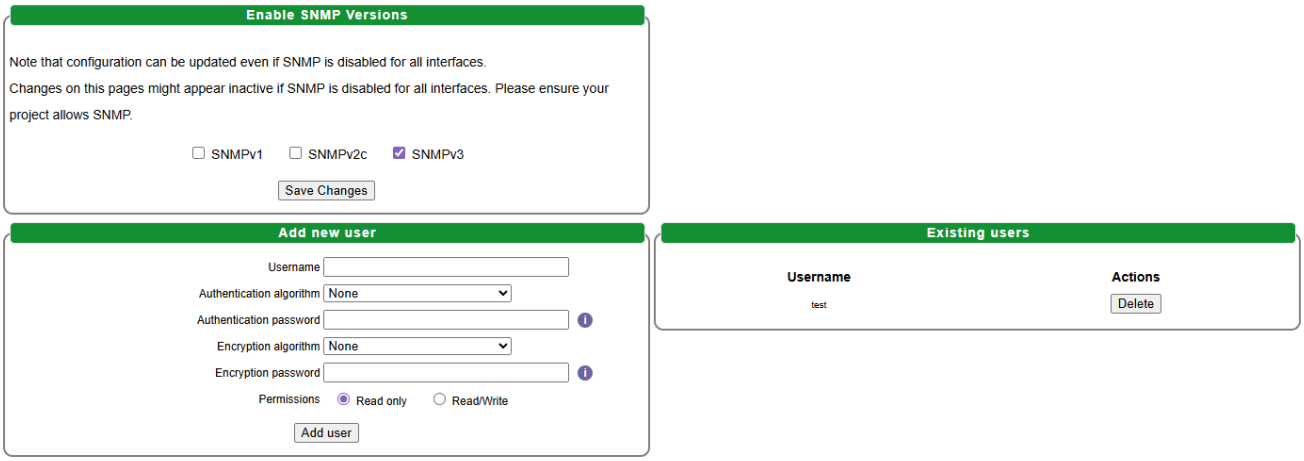
Element	Option	Description	
Diagnostic	**Last command name** **Result** **Message**	Displays the last action executed, its result and a diagnostic message if necessary.	
Certificate to enroll	**Selection list**	From the selection list, select one of the following options to configure the certificate to enroll: • **OPC UA** • **HTTP** (also used for FTP)	
	Challenge password	Password used and provided by the Certificate Authority (CA) for router certificate enrollment and revocation.	
	Retry Period	Specifies the delay, in minutes, between certificate request attempts.	
	Retry Count	Specifies the number of times the device should resubmit a certificate request.	
	Enroll	Allows you to start the enrollment process.	
	Check Status	Allows you to verify the status of the enrollment process.	
	Cancel	Allows you to cancel the enrollment process.	
	Current Enrollment Status	Displays a message about the status of the enrollment process: • **Idle** • **On going**	
	Certificate Status	Displays the status of the certificate and an associated message:	
		Starting	**Enrollment process is starting**
		Success	**Request pending for manual approval**
		Pending	• **Request granted. Certificate will be applied on the next reboot** or • **Request granted. Certificate will be applied on the next reset cold, reset warm or application download**
		Cancel	**Operation cancelled by the user**
		Error	**Request rejected**

This table describes the Public Key Infrastructure (PKI) shared between the M262 Logic/Motion Controller and the SCEP server. It provides the folder list and their usage:

M262 File System Folders	Description
/usr/pki/scep/castore	Stores working certificate received from SCEP server.
/usr/pki/scep/tmp	Stores temporary files.
/usr/pki/scep/csr	Stores the signed certificate request.

Maintenance: SNMP Submenu

The SNMP submenu allows you to choose the Simple Network Management Protocol (SNMP) version and add/remove SNMP user accounts. For more information, refer to SNMP, page 131.



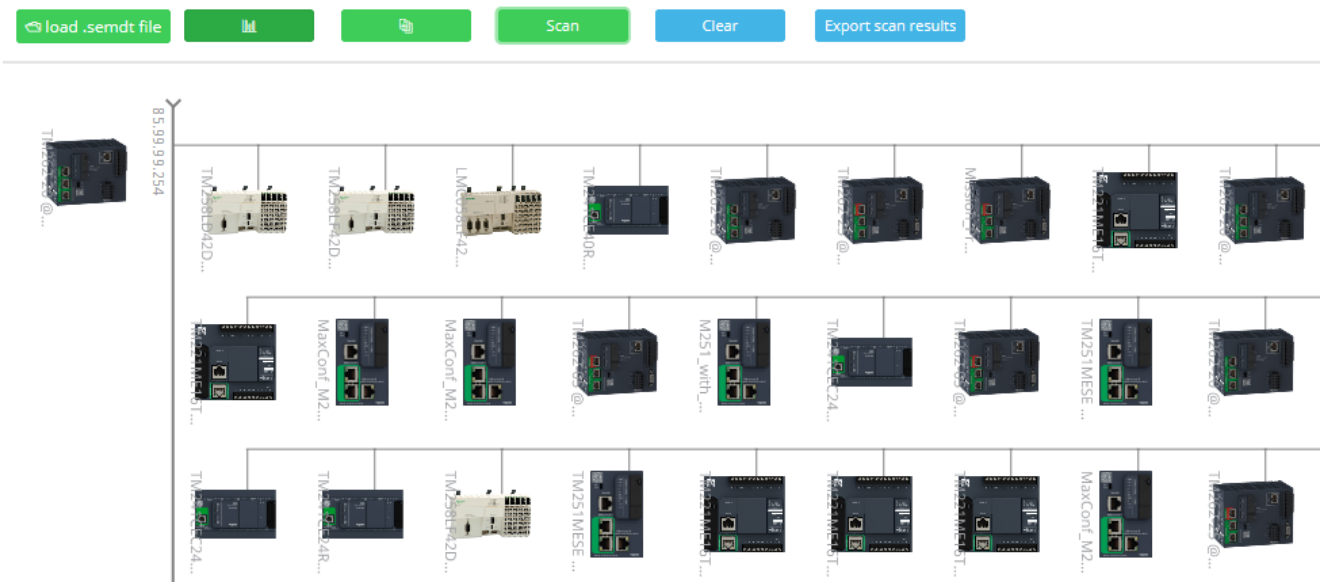
The following table describes the SNMP submenu:

Element	Option	Description
Enable SNMP Versions	• **SNMPv1** • **SNMPv2c** • **SNMPv3**	Select the SNMP protocol version to enable, then confirm your selection by clicking **Save Changes**. This functionality is only available to users with the Device.SNMP.config permission.
Add new user (1)	**Username**	Enter the name of the new user account.
	Authentication algorithm	Select the **Authentication algorithm**: • **None** • **MD5** • **SHA-1** • **SHA-224** • **SHA-256**
	Authentication password	Enter the authentication password of the new user account. It is only required when an **Authentication algorithm** other than **None** is selected.
	Encryption algorithm	Select the **Encryption algorithm**: • **None** • **CBC-DES** • **CFB128-AES-128** • **CFB128-AES-192**
	Encryption password	Enter the encryption password of the new user account. It is only required when an **Encryption algorithm** other than **None** is selected. A user account with encryption also requires authentication.
	Permissions	Select the actions authorized for the new user: • **Read only**: the user can only view the SNMP parameters. • **Read/Write**: the user can view and modify the SNMP parameters.
Existing users (1)	–	Allows you to view and delete the existing user accounts.
(1) This element is only active with **SNMPv3**. It is not supported by **SNMPv1** and **SNMPv2c**.		

Machine Assistant Menu

Overview

The **Machine Assistant** submenu allows you to configure the controller:



For more information on buttons, refer to Industrial Plug and Work, page 276.

Firewall Configuration

Introduction

This section describes how to configure the firewall of the Modicon M262 Logic/ Motion Controller.

Introduction

Firewall Presentation

In general, firewalls help protect network security zone perimeters by blocking unauthorized access and permitting authorized access. A firewall is a device or set of devices configured to permit, deny, encrypt, decrypt, or proxy traffic between different security zones based upon a set of rules and other criteria.

Process control devices and high-speed manufacturing machines require fast data throughput and often cannot tolerate the latency introduced by an aggressive security strategy inside the control network. Firewalls, therefore, play a significant role in a security strategy by providing levels of protection at the perimeters of the network. Firewalls are an important part of an overall, system level strategy.

> **NOTE:** Schneider Electric adheres to industry best practices in the development and implementation of control systems. This includes a "Defense-in-Depth" approach to secure an Industrial Control System. This approach places the controllers behind one or more firewalls to restrict access to authorized personnel and protocols only.

⚠ WARNING

UNAUTHENTICATED ACCESS AND SUBSEQUENT UNAUTHORIZED MACHINE OPERATION

- Evaluate whether your application environments are connected to your critical infrastructure and, if so, take appropriate steps in terms of prevention, based on Defense-in-Depth, before connecting the automation system to any network.
- Limit the number of devices connected to a network to the minimum necessary.
- Isolate your industrial network from other networks inside your company.
- Protect any network against unintended access by using firewalls, VPN, or other, proven security measures, such as an Intrusion Prevention System or Intrusion Detection System.
- Monitor activities within your systems.
- Prevent subject devices from direct access or direct link by unauthorized parties or unauthenticated actions.
- Install certificates that are issued by publicly known Trusted Certificate Authorities.
- Keep your systems up-to-date and rely only on legitimate sources.
- Prepare a recovery plan including backup of your system and process information.

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

For more information on organizational measures and rules covering access to infrastructures, refer to ISO/IEC 27000 series, Common Criteria for Information Technology Security Evaluation, ISO/IEC 15408, IEC 62351, ISA/IEC 62443, NIST Cybersecurity Framework, Information Security Forum - Standard of Good Practice for Information Security and refer to Cybersecurity Guidelines for EcoStruxure Machine Expert, Modicon and PacDrive Controllers and Associated Equipment.

Firewall Configuration

There are three ways to manage the controller firewall configuration:

- Static configuration
- Dynamic changes
- Application settings

Script files are used in the static configuration and for dynamic changes.

Static Configuration

The static configuration is loaded at the controller boot.

The controller firewall can be statically configured by managing a default script file located in the controller. The path to this file is */usr/Cfg/FirewallDefault.cmd*.

NOTE: The file name is case sensitive.

Dynamic Changes

After the controller boot, the controller firewall configuration can be changed by the use of script files.

There are two ways to load these dynamic changes using:

- A physical SD card, page 155.
- A function block, page 155 in the application.

Application Settings

See Ethernet Configuration, page 125.

Dynamic Changes Procedure

Using an SD Card

This table describes the procedure to execute a firewall script from an SD card:

Step	Action
1	Create a valid firewall script, page 157. For example, name the firewall script *FirewallMaintenance.cmd*.
2	Load the firewall script on the SD card. For example, load the firewall script in the *usr/Cfg* folder.

3	In the file *Sys/Cmd/Script.cmd*, add a code line with the command
	`Firewall_install "/pathname/FileName"`
	For example, the code line is
	`Firewall_install "/sd0/usr/Cfg/FirewallMaintenance.cmd"`
	NOTE: The file name is case sensitive.

4	Insert the SD card on the controller.

Using a Function Block in the Application

This table describes the procedure to execute a firewall script from an application:

Step	Action
1	Create a valid firewall script, page 157. For example, name the firewall script *FirewallMaintenance.cmd*.
2	Load the firewall script in the controller memory. For example, load the firewall script in the *usr/Syslog* folder with FTP.
3	Use an `ExecuteScript` function block. For more information, refer to the Modicon M262 Logic/Motion Controller, System Functions and Variables, System Library Guidee (see Modicon M262 Logic/Motion Controller, System Functions and Variables, System Library Guide). For example, the **[SCmd]** input is `'Firewall_install "/usr/Syslog/FirewallMaintenance.cmd"'` **NOTE:** The file name is case sensitive.

Firewall Behavior

Introduction

The firewall configuration depends on the action done on the controller and the initial configuration state. There are five possible initial states:

- There is no default script file in the controller.
- A correct script file is present.
- An incorrect script file is present.
- There is no default script file and the application has configured the firewall.
- A dynamic script file configuration has already been executed.

No Default Script File

If...	Then ...
Boot of the controller	Firewall is not configured. No protection is activated.
Execute dynamic script file	Firewall is configured according to the dynamic script file.
Execute dynamic incorrect script file	Firewall is not configured. No protection is activated.
Download application	Firewall is configured according to the application settings.

Default Script File Present

If...	Then ...
Boot of the controller	Firewall is configured according to the default script file.
Execute dynamic script file	The whole configuration of the default script file is deleted. Firewall is configured according to the dynamic script file.
Execute dynamic incorrect script file	Firewall is configured according to the default script file. The dynamic script file is not taken into account.
Download application	The whole configuration of the application is ignored. Firewall is configured according to the default script file.

Incorrect Default Script File Present

If...	Then ...
Boot of the controller	Firewall is not configured. No protection is activated.
Execute dynamic script file	Firewall is configured according to the dynamic script file.
Download application	Firewall is configured according to the application settings.

Application Settings with No Default Script File

If...	Then ...
Boot of the controller	Firewall is configured according to the application settings.
Execute dynamic script file	The whole configuration of the application settings is deleted. Firewall is configured according to the dynamic script file.
Execute dynamic incorrect script file	Firewall is configured according to the application settings. The dynamic script file is not taken into account.
Download application	The whole configuration of the previous application is deleted. Firewall is configured according to the new application settings.

Execute Dynamic Script File Already Executed

If...	Then ...
Boot of the controller	Firewall is configured according to the dynamic script file configuration (see note).
Execute dynamic script file	The whole configuration of the previous dynamic script file is deleted. Firewall is configured according to the new dynamic script file.
Execute dynamic incorrect script file	Firewall is configured according to the previous dynamic script file configuration. The dynamic incorrect script file is not taken into account.
Download application	The whole configuration of the application is ignored. Firewall is configured according to the dynamic script file.
NOTE: If an SD card containing a cybersecurity script is plugged into the controller, booting is blocked. First remove the SD card to correctly boot the controller.	

Firewall Script Commands

Overview

This section describes how script files (default script files or dynamic script files) are written so that they can be executed during the booting of the controller or during a specific command triggered.

NOTE: The MAC layer rules are managed separately and have more priority over other packet filter rules.

Script File Syntax

The syntax of script files is described in Creating a Script, page 246.

General Firewall Commands

The following commands are available to manage the Ethernet firewall of the M262 Logic/Motion Controller:

Command	Description
`Firewall Enable`	Blocks the frames from the Ethernet interfaces. If no specific IP address or port is authorized, it is not possible to communicate on the Ethernet interfaces. **NOTE:** By default, when the firewall is enabled, the frames are rejected.
`Firewall Disable`	Firewall rules are not applied. Frames are not blocked.
`Firewall Ethx Default Allow` (1)	Frames are accepted by the controller on interface Ethx.
`Firewall Ethx Default Reject` (1)	Frames are rejected by the controller on interface Ethx. **NOTE:** By default, if this line is not present, it corresponds to the command `Firewall Eth1 Default Reject`.

(1) Where Ethx =
- Eth0: USB port
- Eth1: Ethernet_1
- Eth2: Ethernet_2
- Eth3: TMSES4
- Eth4: TMSES4_1
- Eth5: TMSES4_2

Specific Firewall Commands

The following commands are available to configure firewall rules for specific ports and addresses:

Command	Range	Description
`Firewall Ethx Allow IP •.•.•.•`(1)	• = 0...255	Frames from the specified IP address are allowed on all port numbers and port types.
`Firewall Ethx Reject IP •.•.•.•`(1)	• = 0...255	Frames from the specified IP address are rejected on all port numbers and port types.
`Firewall Ethx Allow IPs •.•.•.• to •.•.•.•`(1)	• = 0...255	Frames from the IP addresses in the specified range are allowed for all port numbers and port types. **NOTE:** Rules with specific IP address range will be converted to CIDR format in controller while they are established. Example: “Firewall Eth2 allows IPs 192.168.100.66 to 192.168.100.99 on TCP port 44818” is separated into 7: • 192.168.100.66/31 • 192.168.100.68/30 • 192.168.100.72/29 • 192.168.100.80/28 • 192.168.100.96/27 • 192.168.100.128/26 • 192.168.100.192/29 Using of entire subnet IP ranges avoids firewall rules saturation.
`Firewall Eth1 Reject IPs •.•.•.• to •.•.•.•`(1)	• = 0...255	Frames from the IP addresses in the specified range are rejected for all port numbers and port types.
`Firewall Eth1 Allow port_type port Y`(1)	Y = (destination port numbers)	Frames with the specified destination port number are allowed.
`Firewall Eth1 Reject port_type port Y`(1)	Y = (destination port numbers)	Frames with the specified destination port number are rejected. **NOTE:** When IP forwarding is activated, rules with reject port only filter frames with current controller as destination. They are not applied for the frames routed by the current controller.

Command	Range	Description
`Firewall Eth1 Allow port_type ports Y1 to Y2` (1)	Y = (destination port numbers)	Frames with a destination port number in the specified range are allowed.
`Firewall Eth1 Reject port_type ports Y1 to Y2` (1)	Y = (destination port numbers)	Frames with a destination port number in the specified range are rejected.
`Firewall Eth1 Allow IP •.•.•.• on port_type port Y`(1)	• = 0...255 Y = (destination port numbers)	Frames from the specified IP address and with the specified destination port number are allowed.
`Firewall Ethx Reject IP •.•.•.• on port_type port Y`	• = 0...255 Y = (destination port numbers)	Frames from the specified IP address and with the specified destination port number are rejected.
`Firewall Ethx Allow IP •.•.•.• on port_type ports Y1 to Y2`	• = 0...255 Y = (destination port numbers)	Frames from the specified IP address and with a destination port number in the specified range are allowed.
`Firewall Ethx Reject IP •.•.•.• on port_type ports Y1 to Y2`	• = 0...255 Y = (destination port numbers)	Frames from the specified IP address and with a destination port number in the specified range are rejected.
`Firewall Ethx Allow IPs •1.•1.•1.•1 to •2.•2.•2.•2 on port_type port Y`	• = 0...255 Y = (destination port numbers)	Frames from an IP address in the specified range and with the specified destination port number are allowed.
`Firewall Ethx Reject IPs •1.•1.•1.•1 to •2.•2.•2.•2 on port_type port Y`(1)	• = 0...255 Y = (destination port numbers)	Frames from an IP address in the specified range and with the specified destination port number are rejected.
`Firewall Ethx Allow IPs •1.•1.•1.•1 to •2.•2.•2.•2 on port_type ports Y1 to Y2`(1)	• = 0...255 Y = (destination port numbers)	Frames from an IP address in the specified range and with a destination port number in the specified range are allowed.
`Firewall Ethx Reject IPs •1.•1.•1.•1 to •2.•2.•2.•2 on port_type ports Y1 to Y2`(1)	• = 0...255 Y = (destination port numbers)	Frames from an IP address in the specified range and with a destination port number in the specified range are rejected.
`Firewall Ethx Allow MAC ••:••:••:••:••:••`(1)	• = 0...F	Frames from the specified MAC address ••:••:••:••:•• are allowed. **NOTE:** When the rules to allow the MAC address are applied, only the listed MAC addresses can communicate with the controller, even if other rules are allowed.
`Firewall Ethx Reject MAC ••:••:••:••:••:••`(1)	• = 0...F	Frames with the specified MAC address ••:••:••:••:•• are rejected.
`Firewall Ethx` (1) `Established to port_type port Y`	Y = 0...65535	Frames established from the controller with the protocols TCP/UDP to the specified destination port number are allowed.

(1) Where Ethx =

- Eth0: USB port
- Eth1: Ethernet_1
- Eth2: Ethernet_2
- Eth3: TMSES4
- Eth4: TMSES4_1
- Eth5: TMSES4_2

NOTE: When IP Forwarding is activated, rules with **Reject port** only filter frames with current controller as destination. They are not applied for the frames routed by the current controller.

Script Example

```
; Enable FireWall. All frames are rejected;
FireWall Enable;
; Allow frames on Eth1
FireWall Eth1 Default Allow;
; Block all Modbus Requests on all IP address
Firewall Eth1 Reject tcp port 502;
; Reject frames on Eth2
FireWall Eth2 Default Reject;
; Allow FTP active connection for IP address 85.16.0.17
FireWall Eth2 Allow IP 85.16.0.17 on tcp ports 20 to 21;
```

NOTE: IP addresses are converted to CIDR format.

For example:

"`FireWall Eth2 Allow IPs 192.168.100.66 to 192.168.100.99 on tcp port 44818;`", is separated into 7:

- `192.168.100.66/31`
- `192.168.100.68/30`
- `192.168.100.72/29`
- `192.168.100.80/28`
- `192.168.100.96/27`
- `192.168.100.128/26`
- `192.168.100.192/29`

To prevent a firewall error, use the entire subnet configuration.

The following is an example of a firewall in white list mode. The example has all communication blocked by default and allows only the necessary services.

NOTE: This example is designed to show most of the commands available with the firewall. It should be adapted to your configuration and tested before implementation.

Commands	Comments
`Firewall Enable`	; Enable the firewall.
Eth1 Configuration	
`Firewall Eth1 Default Reject`	; Reject all frames on interface ETH1.
	; In this example, ETH1 is connected to the Industrial Ethernet devices network and therefore can be relatively trusted.
`Firewall Eth1 Allow TCP port 502`	; Allow Modbus TCP server on interface ETH1.
	; There is no authentication on Modbus so this should be allowed only on trusted networks.
`Firewall Eth1 Established to TCP port 502`	; Allow replies to communication established by the controller to TCP port 502.
	; This is necessary when using PlcCommunication library to communicate using Modbus TCP protocol.
`Firewall Eth1 Allow UDP port 2222`	; Allow ETHIP scanner implicit exchanges replies to UDP port 2222 (ETHIP) on interface ETH1.
`Firewall Eth1 Established to TCP port 44818`	; Allow replies to communication established by the controller to TCP port 44818 (ETHIP) on interface ETH1.
	; The last 2 commands allow the EtheNetIP Scanner to communicate with the industrial ethernet devices.
Eth2 Configuration	
`Firewall Eth2 Default Reject`	; Reject all frames on interface ETH2. This interface is connected to a network used mainly for commissioning.
`Firewall Eth2 Allow TCP port 4840`	; Allow OPC UA server on interface ETH2.
`Firewall Eth2 Allow TCP port 443`	; Allow Web server (https) on interface ETH2.
`Firewall Eth2 Allow TCP port 8089`	; Allow WebVisualisation (https) on interface ETH2.
`Firewall Eth2 Allow TCP ports 20 to 21`	; Allow FTP in active mode on interface ETH2.
`Firewall Eth2 Allow IP 192.168.1.1 on UDP ports 27126 to 27127`	; Allow the IP of the commissioning PC to discover and configure the IP address of the controller.
	; This should be allowed only on a trusted network as IP can be changed even if the User Rights are configured.
`Firewall Eth2 Allow IPs 192.168.1.1 to 192.168.1.2 on UDP port 1740`	; Allow the IP of the commissioning PC and an HMI to communicate with the controller using CoDeSys protocol.
`Firewall Eth2 Allow TCP port 11740`	; Allow Fast TCP on interface ETH2. This allow to connect to the controller using TCP.
`Firewall Eth2 Allow TCP port 2222`	; Allow implicit communication with UDP port 2222 (ETHIP) on interface ETH2.
`Firewall Eth2 Allow TCP port 44818`	; Allow explicit communication to TCP port 44818 (ETHIP) on interface ETH2. The last 2 commands allow to use the controller as an EtherNetIP Adapter.
`Firewall Eth2 Allow MAC 4C:CC:6A:A1:09:C8`	; Allow the MAC address of the HMI.
`Firewall Eth2 Allow MAC 00:0C:29:92:43:A8`	; Allow the MAC address of the commissioning PC. Only the MAC addresses allowed can communicate with the controller.
Eth3 Configuration TMSES4	
`Firewall Eth3 Default Reject`	; Reject frames on TMSES4. This interface is connected to the Plant network and can access the web. It should be considered as untrusted.
`Firewall Eth3 Established to TCP port 443`	; Allow http client (for example to connect to Machine Advisor) on interface TMSES4.
`Firewall Eth3 Allow TCP port 11740`	; Allow Fast TCP on interface TMSES4. This allows to connect to the controller remotely. It must not be allowed unless User Rights are activated on the controller.

NOTE: Characters are limited to 200 per line, including comments.

Used Ports

This table lists the port numbers used by the M262 Logic/Motion Controller services:

Service	Port numbers	Out-of-box / Default configuration (1)
Programming software	TCP 11740 UDP 1740	Yes
FTP over TLS	TCP 20, 21	Yes
HTTP (2)	TCP 80	Yes
HTTPS	TCP 443	Yes
Modbus TCP	TCP 502	No
OPC UA	TCP 4840	No
Discovery	UDP 27126, 27127	Yes
Bonjour Discovery Protocol	UDP 5353	Yes
Web Services Dynamical Discovery	TCP 5357 UDP 3702	Yes
SNMP	UDP 161	No
NVL	UDP default value: 1202 (destination port) (Source port is dynamically assigned.)	No
EtherNet/IP	TCP 44818 UDP 2222, 44818	No
WebVisualisation	TCP 8080 (HTTP), 8089 (HTTPS)	No
DHCP/BOOTP	UDP 68	Yes
TFTP	UDP 69	No
NTP	UDP 123	No
Safe Logger (3)	UDP 35021, 45000	Yes
Machine Assistant	UDP 45001, 45003, 45004	Yes

(1) Default configuration is used when a default project is loaded into the controller. The port status is exactly the same as the factory settings.

(2) HTTP is automatically redirected to HTTPS.

(3) Safe Logger ports are only applicable to TM262M• references.

CoDeSys Protocol

Introduction

The CoDeSys protocol is a protocol providing a transparent access to your controllers and HMIs.

The CoDeSys protocol is used for data exchange:

- between the EcoStruxure Machine Expert software (PC) and the runtime systems (controller, HMI) configured by Vijeo-Designer.
- between controllers and integrated HMIs supporting CoDeSys protocol.

CoDeSys Protocol Configuration

The maximum number of clients that can connect simultaneously to CoDeSys protocol is 5.

CoDeSys Protocol Data Types

The following data types are supported by the CoDeSys protocol:

- BOOL
- BYTE
- DINT
- DWORD
- INT
- REAL
- SINT
- UDINT
- UINT
- USINT
- WORD

The maximum number of variables for a data type is 12000.

Industrial Ethernet

Introduction

This chapter describes how to add and configure the Industrial Ethernet.

Industrial Ethernet Presentation

Overview

Industrial Ethernet is the term used to represent the industrial protocols that use the standard Ethernet physical layer and standard Ethernet protocols.

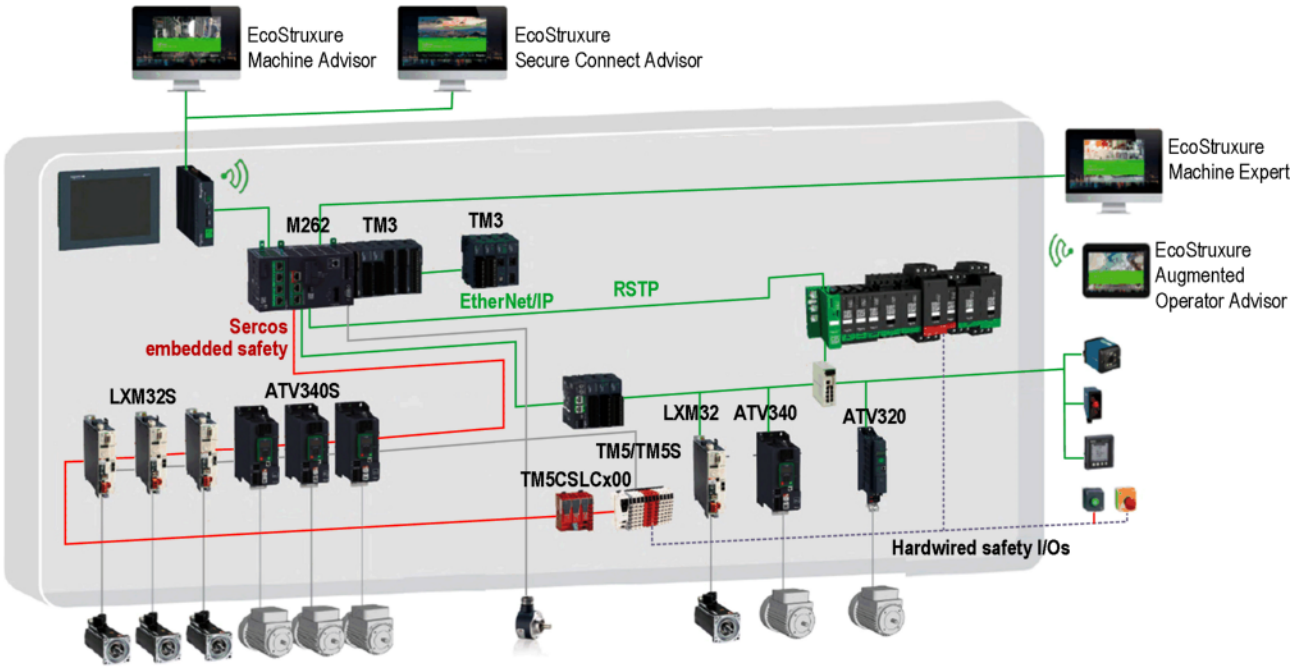
On an Industrial Ethernet network, you can connect:

- industrial devices (industrial protocols)
- non-industrial devices (other Ethernet protocols)

For more information, refer to Industrial Ethernet Overview, User Guide.

Industrial Ethernet Architecture

This figure presents a typical Industrial Ethernet architecture:



This architecture is configurable with EcoStruxure Machine Expert.

Industrial Ethernet Description

M262 Logic/Motion Controller	
Features	**Description**
Topology	Daisy chain and Star via switches
Bandwidth	10/100 Mbit/s for Ethernet 1 port 10/100/1000 Mbit/s for Ethernet 2 port
EtherNet/IP Scanner	
Performance	Up to 64 EtherNet/IP target devices managed by the controller, monitored within a timeslot of: • 40 ms on TM262L01MESE8T, TM262L10MESE8T, TM262M05MESS8T and TM262M15MESS8T • 20 ms on TM262L20MESE8T, TM262M25MESS8T and TM262M35MESS8T
Number of connections	• 0...6 on Ethernet_1 port of TM262M• • 0...64 on Ethernet_2 port of TM262M• and on both Ethernet ports of TM262L•
Number of input words	0...15360
Number of output words	0...15360
I/O communications	EtherNet/IP Scanner service Function block for configuration and data transfer
	Originator/Target
Modbus TCP IOScanner	
Performance	Up to 64 Modbus TCP slave devices managed by the controller, monitored within a timeslot of: • 20 ms on TM262L01MESE8T, TM262L10MESE8T, TM262M05MESS8T and TM262M15MESS8T • 10 ms on TM262L20MESE8T, TM262M25MESS8T and TM262M35MESS8T
Number of connections	• 0...6 on Ethernet_1 port of TM262M• • 0...64 on Ethernet_2 port of TM262M• and on both Ethernet ports of TM262L•
Number of input words	0...8000
Number of output words	0...8000
I/O communications	Modbus TCP IOScanner service Function block for data transfer
	Master/Slave
Sercos	
Performance	Refer to Performance Overview, page 22.

M262 Logic/Motion Controller	
Features	**Description**
Other services	FDT/DTM/EDS management
	FDR (Fast Device Replacement)
	DHCP server
	Security management (refer to Security Parameters, page 127 and Firewall Configuration, page 153)
	Modbus TCP server
	Modbus TCP client
	EtherNet/IP adapter (controller as a target on EtherNet/IP)
	EtherNet/IP Originator
	Modbus TCP server (controller as a slave on Modbus TCP)
	Web server, page 132
	FTP server, page 129
	NTP, page 83
	OPC UA, page 215
	RSTP (ETH2)
	SNMP, page 131
	IEC VAR ACCESS
Additional features	You can mix the Ethernet/IP and Modbus TCP server devices: • 96 (64 EIP and 32 TCP) on TM262L01MESE8T, TM262L10MESE8T, TM262M05MESS8T and TM262M15MESS8T • 128 (64 EIP and 64 TCP) on TM262L20MESE8T, TM262M25MESS8T and TM262M35MESS8T. Devices can be directly accessed for configuration, monitoring, and management purposes. Network transparency between control network and device network (controller can be used as a gateway). **NOTE:** Using the controller as a gateway can impact the performance of the controller.
Single Wire Architecture, page 194	Allows up to 6 Ethernet devices (EtherNet/IP, TCP/IP, and so on) to be added to the end of a cable containing Sercos devices. The last Sercos device acts as a gateway. No additional gateways or switches are required. The Ethernet frames are embedded within the Sercos frames.

EtherNet/IP Overview

EtherNet/IP is the implementation of the CIP protocol over standard Ethernet.

The EtherNet/IP protocol uses an Originator/Target architecture for data exchange.

Originators are devices that initiate data exchanges with Target devices on the network. This applies to both I/O communications and service messaging. This is the equivalent of the role of a client in a Modbus network.

Targets are devices that respond to data requests generated by Originators. This applies to both I/O communications and service messaging. This is the equivalent of the role of a server in a Modbus network.

EtherNet/IP Adapter is an end-device in an EtherNet/IP network. I/O blocks and drives can be EtherNet/IP Adapter devices.

The communication between an EtherNet/IP Originator and Target is accomplished using an EtherNet/IP connection.

Modbus TCP Overview

The Modbus TCP protocol uses a Client/Server architecture for data exchange.

The Modbus TCP explicit (non-cyclic) data exchanges are managed by the application.

Modbus TCP implicit (cyclic) data exchanges are managed by the Modbus TCP IOScanner. The Modbus TCP IOScanner is a service based on Ethernet that polls slave devices continuously to exchange data, status, and diagnostic information. This process monitors inputs and controls outputs of slave devices.

Clients are devices that initiate data exchange with other devices on the network. This applies to both I/O communications and service messaging.

Servers are devices that address any data requests generated by a Client. This applies to both I/O communications and service messaging.

The communication between the Modbus TCP IOScanner and the slave device is accomplished using Modbus TCP channels.

Sercos Overview

For more information on Sercos standard and configuration, refer to Overview of the Sercos Standard, page 193.

Adding the Protocol Manager

The protocol manager must be present on the **Ethernet_1 (ETH1)** or **Ethernet_2 (ETH2)** node of the **Devices tree** to activate these functions and services:

- EtherNet/IP Scanner
- Generic TCP/UDP Manager
- Modbus TCP IO Scanner

When a protocol manager is defined on an interface, this interface address must be **Fixed**. The post-configuration defined for this interface is not applied, if any.

To manually add the protocol manager to the **Ethernet_1 (ETH1)** / **Ethernet_2 (ETH2)**:

Step	Action
1	In the **Devices tree**, select **Ethernet_1 (ETH1)** / **Ethernet_2 (ETH2)** and click the green plus button of the node, or right-click **Ethernet_1 (ETH1)** / **Ethernet_2 (ETH2)** and execute the **Add Device** command from the contextual menu. **Result**: The **Add Device** dialog box opens.
2	In the **Add Device** dialog box, select your protocol manager: • **Protocol Managers > EtherNet/IP Scanner** • **Protocol Managers > ModbusTCP IOScanner** • **Protocol Managers > TCP/UDP Manager**
3	Click the **Add Device** button.
4	Click the **Close** button.

For more information, refer to Industrial Ethernet Manager Configuration (see EtherNet/IP, User Guide), EtherNet/IP Target Settings (see EtherNet/IP, User Guide) and Modbus TCP Settings (see Modbus TCP, User Guide).

Adding the Sercos Master

The Sercos fieldbus must be present on the **Ethernet_1 (ETH1)** to activate the Sercos Master.

To manually add the **Sercos Master** to the **Ethernet_1 (ETH1)**:

Step	Action
1	In the **Devices tree**, select **Ethernet_1 (ETH1)** and click the green plus button of the node, or right-click **Ethernet_1 (ETH1)** and execute the **Add Device** command from the contextual menu. **Result**: The **Add Device** dialog box opens.
2	In the **Add Device** dialog box, select **Protocol Managers** > **Sercos Master**.
3	Click the **Add Device** button.
4	Click the **Close** button.

DHCP Server

Overview

The DHCP server offers addresses to the devices connected on the Ethernet network. The DHCP server only delivers static addresses. A unique identified slave gets a unique address. DHCP slave devices are identified either by their MAC address or their DHCP device name. The DHCP server configuration table defines the relation between addresses and identified slave devices.

The DHCP server addresses are given with an infinite lease time. There is no need for the slave devices to refresh the leased IP address.

The synthesis of the DHCP server configuration is displayed on the **Ethernet Services** tab, page 80.

For more information, refer to IP Addressing Methods (see Modbus TCP, User Guide).

Fast Device Replacement

Overview

The Fast Device Replacement (FDR) helps facilitate replacing and reconfiguring a network device. This function is available on the Ethernet 1 and Ethernet 2 ports of the M262 Logic/Motion Controller.

For more information, refer to Slave Device Replacement with FDR (see Modbus TCP, User Guide).

Controller as a Target Device on EtherNet/IP

Introduction

This section describes the configuration of the M262 Logic/Motion Controller as an EtherNet/IP target device (EtherNet/IP Adapter or EtherNet/IP Scanner).

For further information about EtherNet/IP, refer to the www.odva.org website.

EtherNet/IP Target Configuration

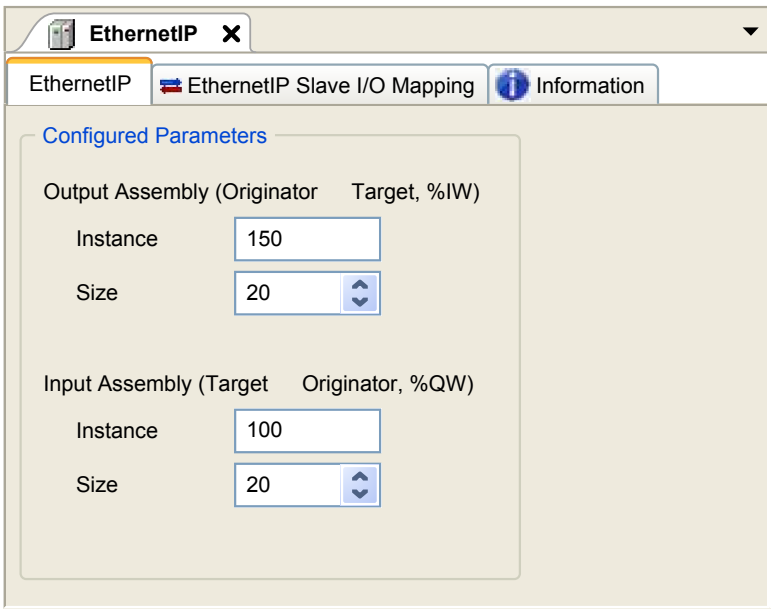
To configure your M262 Logic/Motion Controller as an EtherNet/IP target device, you must:

Step	Action
1	Select **EthernetIP** in the **Hardware Catalog**.
2	Drag and drop it to the **Devices tree** on one of the highlighted nodes. **NOTE:** If the chosen node is **COM_Bus**, a TMSES4 expansion module is automatically added to your configuration. For more information on adding a device to your project, refer to: • Using the Hardware Catalog found in the EcoStruxure Machine Expert Programming Guide • Using the Contextual Menu or Plus Button found in the EcoStruxure Machine Expert Programming Guide

EtherNet/IP Parameters Configuration

To configure the EtherNet/IP parameters, double-click **EthernetIP** in the **Devices tree**.

This dialog box is displayed:



The EtherNet/IP configuration parameters are defined as:

- **Instance**:

  Number referencing the input or output Assembly.

- **Size**:

  Number of channels of an input or output Assembly.

  The memory size of each channel is 2 bytes that stores the value of an %IWx or %QWx object, where x is the channel number.

  For example, if the **Size** of the **Output Assembly** is 20, it represents that there are 20 input channels (IW0...IW19) addressing %IWy...%IW(y+20-1), where y is the first available channel for the Assembly.

Element		Admissible Controller Range	Default Value
Output Assembly	**Instance**	150...189	150
	Size	2...250	20
Input Assembly	**Instance**	100...149	100
	Size	2...250	20

EDS File Generation

You can generate an EDS file to configure EtherNet/IP cyclic data exchanges.

To generate the EDS file:

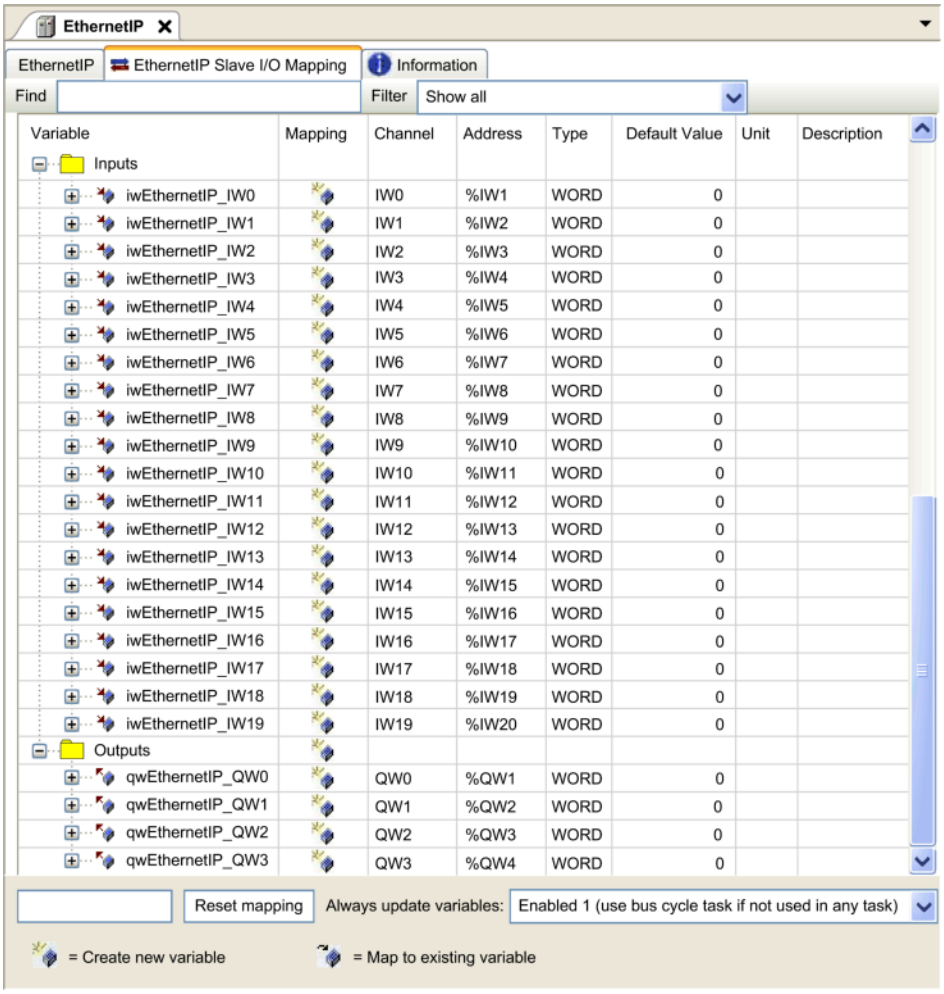
Step	Action
1	In the **Devices tree**, right-click the **EthernetIP** node and choose the **Export as EDS** command from the contextual menu.
2	Modify the default file name and location as required.
3	Click **Save**.

NOTE: The **Major Revision** and **Minor Revision** objects of the EDS file, defined in the file, are used to ensure uniqueness of the EDS file. The values of these objects do not reflect the actual controller revision level.

A generic EDS file for the M262 Logic/Motion Controller is also available on the Schneider Electric website. You must adapt this file to your application by editing it and defining the required Assembly instances and sizes.

EthernetIP Slave I/O Mapping Tab

Variables can be defined and named in the **EthernetIP Slave I/O Mapping** tab. Additional information such as topological addressing is also provided in this tab.



The table below describes the **EthernetIP Slave I/O Mapping** configuration:

Channel		Type	Default Value	Description
Input	IW0	WORD	-	Command word of controller outputs (% QW)
	IWxxx			
Output	QW0	WORD	-	State of controller inputs (%IW)
	QWxxx			

The number of words depends on the size parameter configured in EtherNet/IP Target Configuration, page 168.

Output means OUTPUT from Originator controller (= %IW for the controller).

Input means INPUT from Originator controller (= %QW for the controller).

Connections on EtherNet/IP

To access a target device, an Originator opens a connection which can include several sessions that send requests.

One explicit connection uses one session (a session is a TCP or UDP connection).

One I/O connection uses two sessions.

The following table shows the EtherNet/IP connections limitations:

Characteristic	Maximum
Explicit connections	8 (Class 3)
I/O connections	1 (Class 1)
Connections	8
Sessions	16
Simultaneous requests	32

NOTE: The M262 Logic/Motion Controller supports cyclic connections only. If an Originator opens a connection using a change of state as a trigger, packets are sent at the RPI rate.

NOTE: For a network topology that has RSTP enabled, verify that the RPI/timeout combination respects the minimum convergence time of 100 ms that is required for RSTP.

Profile

The controller supports the following objects:

Object class	Class ID (hex)	Cat.	Number of Instances	Effect on Interface Behavior
Identity Object, page 173	01	1	1	Supports the reset service
Message Router Object, page 175	02	1	1	Explicit message connection
Assembly Object, page 176	04	2	2	Defines I/O data format
Connection Manager Object, page 178	06	–	1	–
TCP/IP Interface Object, page 179	F5	1	1	TCP/IP configuration
Ethernet Link Object, page 180	F6	1	1	Counter and status information
Interface Diagnostic Object, page 181	350	1	1	–
IOScanner Diagnostic Object, page 183	351	1	1	–
Connection Diagnostic Object, page 184	352	1	1	–
Explicit Connection Diagnostic Object , page 186	353	1	1	–
Explicit Connections Diagnostic List Object, page 186	354	1	1	–

Identity Object (Class ID = 01 hex)

The following table describes the class attributes of the Identity Object:

Attribute ID (hex)	Access	Name	Data Type	Value (hex)	Details
1	Get	Revision	UINT	01	Implementation revision of the Identity Object.
2	Get	Max Instance	UINT	01	The largest instance number.
6	Get	Max Class Attribute	UINT	01	The largest class attributes value.
7	Get	Max Instance Attribute	UINT	07	The largest instance attributes value.

The following table describes the Class Services:

Service Code (hex)	Name	Description
01	Get Attribute All	Returns the value of all class attributes.
0E	Get Attribute Single	Returns the value of the specified attribute.

The following table describes the Instance Services:

Service Code (hex)	Name	Description
01	Get Attribute All	Returns the value of all class attributes.
05	Reset (1)	Initializes EtherNet/IP component (controller reboot).
0E	Get Attribute Single	Returns the value of the specified attribute.

(1) Reset Service description:

When the Identity Object receives a Reset request, it:

- determines whether it can provide the type of reset requested
- responds to the request
- attempts to perform the type of reset requested

NOTE: The Reset Service only applies if the corresponding parameter has been activated in post configuration file. See Post Configuration File Example for TM262M•, page 232.

The Reset common service has one specific parameter, Type of Reset (USINT), with the following values:

Value	Type of Reset
0	Reboots the controller **NOTE:** This is the default value if this parameter is omitted.
1	Not supported
2	Not supported
3...99	Reserved
100...199	Vendor specific
200...255	Reserved

The following table describes the Instance attributes:

Attribute ID (hex)	Access	Name	Data Type	Value (hex)	Details
1	Get	Vendor ID	UINT	F3	Schneider Automation ID
2	Get	Device type	UINT	0E	Controller
3	Get	Product code	UINT	4102	Controller product code
4	Get	Revision	Struct of USINT, USINT	–	Product revision number of the controller (1). Equivalent to the 2 low bytes of the controller version.
5	Get	Status	WORD	–	Status word(2)
6	Get	Serial number	UDINT	–	Serial number of the controller: XX + 3 LSB of MAC address.
7	Get	Product name	Struct of USINT, STRING	–	–

(1) Mapped in a WORD:

- MSB: minor revision (second USINT)
- LSB: major revision (first USINT)

Example: 0205 hex means revision V5.2.

(2) Status word (Attribute 5):

Bit	Name	Description
0	Owned	Unused.
1	Reserved	–
2	Configured	TRUE indicates the device application has been reconfigured.
3	Reserved	–
4...7	Extended Device Status	• 0: Self-testing or undetermined • 1: Firmware update in progress • 2: At least one invalid I/O connection detected • 3: No I/O connections established • 4: Non-volatile configuration invalid • 5: Unrecoverable error detected • 6: At least one I/O connection in RUNNING state • 7: At least one I/O connection established, all in idle mode • 8: Reserved • 9...15: Unused
8	Minor Recoverable Fault	TRUE indicates the device detected an error, which, under most circumstances, is recoverable. This type of event does not lead to a change in the device state.
9	Minor Unrecoverable Fault	TRUE indicates the device detected an error, which, under most circumstances, is unrecoverable. This type of event does not lead to a change in the device state.
10	Major Recoverable Fault	TRUE indicates the device detected an error, which requires the device to report an exception and enter into the HALT state. This type of event leads to a change in the device state, but, under most circumstances, is recoverable.
11	Major Unrecoverable Fault	TRUE indicates the device detected an error, which requires the device to report an exception and enter into the HALT state. This type of event leads to a change in the device state, but, under most circumstances, is not recoverable.
12...15	Reserved	–

Message Router Object (Class ID = 02 hex)

The following table describes the class attributes of the Message Router object:

Attribute ID (hex)	Access	Name	Data Type	Value (hex)	Details
1	Get	Revision	UINT	01	Implementation revision number of the Message Router Object.
2	Get	Max Instance	UINT	02	The largest instance number.
3	Get	Number of Instance	UINT	01	The number of object instances.
4	Get	Optional Instance Attribute List	Struct of UINT, UINT [ ]	02	The first 2 bytes contain the number of optional instance attributes. Each following pair of bytes represents the number of other optional instance attributes (from 100 to 119).
5	Get	Optional Service List	UINT	0A	The number and list of any implemented optional services attribute (0: no optional services implemented).
6	Get	Max Class Attribute	UINT	07	The largest class attributes value.
7	Get	Max Instance Attribute	UINT	02	The largest instance attributes value.

The following table describes the Class services:

Service Code (hex)	Name	Description
01	Get_Attribute_All	Returns the value of all class attributes.
0E	Get_Attribute_Single	Returns the value of the specified attribute.

The following table describes the Instance services:

Service Code (hex)	Name	Description
01	Get_Attribute_All	Returns the value of all class attributes.
0E	Get_Attribute_Single	Returns the value of the specified attribute.

The following table describes the Instance attributes:

Attribute ID (hex)	Access	Name	Data Type	Value	Description
1	Get	Implemented Object List	Struct of UINT, UINT [ ]	–	Implemented Object list. The first 2 bytes contain the number of implemented objects. Each 2 bytes that follow represents another implemented class number. This list contains the following objects: • Identity • Message Router • Assembly • Connection Manager • Parameter • File Object • Modbus • Port • TCP/IP • Ethernet Link
2	Get	Number available	UINT	512	Maximum number of concurrent CIP (Class 1 or Class 3) connections supported.
3	Get	Number active	UINT	–	Numbers of connections currently used by system component.

Assembly Object (Class ID = 04 hex)

The following table describes the class attributes of the Assembly object:

Attribute ID (hex)	Access	Name	Data Type	Value (hex)	Details
1	Get	Revision	UINT	02	Implementation revision of the Assembly Object.
2	Get	Max Instance	UINT	BE	The largest instance number.
3	Get	Number of Instances	UINT	03	The number of object instances.
4	Get	Optional Instance Attribute List	Struct of: UINT UINT [ ]	 01 04	The first 2 bytes contain the number of optional instance attributes. Each following pair of bytes represents the number of other optional instance attributes.
5	Get	Optional Service List	UINT	Not suppor-ted	The number and list of any implemented optional services attribute (0: no optional services implemented).
6	Get	Max Class Attribute	UINT	07	The largest class attributes value.
7	Get	Max Instance Attribute	UINT	04	The largest instance attributes value.

The following table describes the Class Services:

Service Code (hex)	Name	Description
0E	Get Attribute Single	Returns the value of the specified attribute.

The following table describes the Instance Services:

Service Code (hex)	Name	Description
0E	Get Attribute Single	Returns the value of the specified attribute.
10	Set Attribute Single	Modifies the value of the specified attribute.

Instances Supported

Output means OUTPUT from Originator controller (= %IW for the controller).

Input means INPUT from Originator controller (= %QW for the controller).

The controller supports 2 Assemblies:

Name	Instance	Data Size
Controller Output (% IW)	Configurable: must be between 100 and 149	2...40 words
Controller Input (% QW)	Configurable: must be between 150 and 189	2...40 words

NOTE: The Assembly object binds together the attributes of multiple objects so that information to or from each object can be communicated over a single connection. Assembly objects are static.

The Assemblies in use can be modified through the parameter access of the network configuration tool (RSNetWorx). The controller needs to recycle power to register a new Assembly assignment.

The following table describes the Instance attributes:

Attribute ID (hex)	Access	Name	Data Type	Value	Description
3	Get/Set	Instance Data	ARRAY of Byte	–	Data Set service only available for Controller output.
4	Get	Instance Data Size	UINT	4...80	Size of data in byte.

Access from an EtherNet/IP Scanner

When an EtherNet/IP Scanner needs to exchange assemblies with a M262 Logic/Motion Controller, it uses the following access parameters (*Connection path*):

- Class 4
- Instance xx where xx is the instance value (example: 2464 hex = instance 100).
- Attribute 3

In addition, a configuration assembly must be defined in the Originator.

For example: Class 4, Instance 3, Attribute 3, the resulting *Connection Path* is:

- 2004 hex
- 2403 hex
- 2c<xx> hex

Connection Manager Object (Class ID = 06 hex)

The following table describes the class attributes of the Assembly Object:

Attribute ID (hex)	Access	Name	Data Type	Value (hex)	Details
1	Get	Revision	UINT	01	Implementation revision of the Connection Manager Object.
2	Get	Max Instance	UINT	01	The largest instance number.
3	Get	Number of Instances	UINT	01	The number of object instances.
4	Get	Optional Instance Attribute List	Struct of: UINT UINT [ ]	–	The number and list of the optional attributes. The first word contains the number of attributes to follow and each following word contains another attribute code. Following optional attributes include: • total number of incoming connection open requests • the number of requests rejected due to non-conforming format of the Forward Open • the number of requests rejected because of insufficient resources • the number of requests rejected due to parameter value sent with the Forward Open • the number of Forward Close requests received • the number of Forward Close requests with an invalid format • the number of Forward Close requests that could not be matched to an active connection • the number of connections that have timed out because the other side stopped producing, or a network disconnection occurred
6	Get	Max Class Attribute	UINT	07	The largest class attributes value.
7	Get	Max Instance Attribute	UINT	08	The largest instance attributes value.

The following table describes the Class Services:

Service Code (hex)	Name	Description
01	Get Attribute All	Returns the value of all class attributes.
0E	Get Attribute Single	Returns the value of the specified attribute.

The following table describes the Instance Services:

Service Code (hex)	Name	Description
01	Get Attribute All	Returns the value of all instance attributes.
0E	Get Attribute Single	Returns the value of the specified attribute.
4E	Forward Close	Closes an existing connection.
52	Unconnected Send	Sends a multi-hop unconnected request.
54	Forward Open	Opens a new connection.

The following table describes the Instance attributes:

Attribute ID (hex)	Access	Name	Data Type	Value	Description
1	Get	Open Requests	UINT	–	Number of Forward Open service requests received.
2	Get	Open Format Rejects	UINT	–	Number of Forward Open service requests which were rejected due to invalid format.
3	Get	Open Resource Rejects	ARRAY of Byte	–	Number of Forward Open service requests which were rejected due to lack of resources.
4	Get	Open Other Rejects	UINT	–	Number of Forward Open service requests which were rejected for reasons other than invalid format or lack of resources.
5	Get	Close Requests	UINT	–	Number of Forward Close service requests received.
6	Get	Close Format Requests	UINT	–	Number of Forward Close service requests which were rejected due to invalid format.
7	Get	Close Other Requests	UINT	–	Number of Forward Close service requests which were rejected for reasons other than invalid format.
8	Get	Connection Timeouts	UINT	–	Total number of connection timeouts that have occurred in connections controlled by this Connection Manager.

TCP/IP Interface Object (Class ID = F5 hex)

This object maintains link specific counters and status information for an Ethernet 802.3 communications interface.

The following table describes the class attributes of the TCP/IP Interface Object:

Attribute ID (hex)	Access	Name	Data Type	Value	Details
1	Get	Revision	UINT	4	Implementation revision of the TCP/IP Interface Object.
2	Get	Max Instance	UINT	2	The largest instance number.
3	Get	Number of Instances	UINT	2	The number of object instances.

The following table describes the Class Services:

Service Code (hex)	Name	Description
01	Get Attribute All	Returns the value of all class attributes.
0E	Get Attribute Single	Returns the value of the specified attribute.

Instance Codes

Only instance 1 is supported.

The following table describes the Instance Services:

Service Code (hex)	Name	Description
01	Get Attribute All	Returns the value of all instance attributes.
0E	Get Attribute Single	Returns the value of the specified instance attribute.

The following table describes the Instance Attributes:

Attribute ID (hex)	Access	Name	Data Type	Value	Description
1	Get	Status	DWORD	Bit level	• 0: The interface configuration attribute has not been configured. • 1: The interface configuration contains a valid configuration. • 2...15: Reserved.
2	Get	Configuration Capability	DWORD	Bit level	• 0: BOOTP Client • 2: DHCP Client • 5: Configurable in the software All other bits are reserved and set to 0.
3	Get	Configuration	DWORD	Bit level	• 0: The interface configuration is valid. • 1: The interface configuration is obtained with BOOTP. • 2: The interface configuration is obtained with DHCP. • 3: Reserved All other bits are reserved and set to 0.
4	Get	Physical Link	UINT	Path size	Number of 16 bits word in the element Path.
			Padded EPATH	Path	Logical segments identifying the physical link object. The path is restricted to one logical class segment and one logical instance segment. The maximum size is 12 bytes.
5	Get	Interface configuration	UDINT	IP Address	–
			UDINT	Network Mask	–
			UDINT	Gateway Address	–
			UDINT	Primary Name	–
			UDINT	Secondary Name	0: No secondary name server address has been configured.
			STRING	Default Domain Name	0: No Domain Name is configured.
6	Get	Host Name	STRING	–	ASCII characters. 0: No host name is configured.

Ethernet Link Object (Class ID = F6 hex)

This object provides the mechanism to configure a TCP/IP network interface device.

The following table describes the class attributes of the Ethernet Link object:

Attribute ID (hex)	Access	Name	Data Type	Value (hex)	Details
1	Get	Revision	UINT	4	Implementation revision of the Ethernet Link Object.
2	Get	Max Instance	UINT	255	The largest instance number.
3	Get	Number of Instances	UINT	4	The number of object instances.

The following table describes the class services:

Service Code (hex)	Name	Description
01	Get Attribute All	Returns the value of all class attributes.
0E	Get Attribute Single	Returns the value of the specified attribute.

Instance Codes

Only instance 1 is supported.

The following table describes the instance services:

Service Code (hex)	Name	Description
01	Get Attribute All	Returns the value of all instance attributes.
0E	Get Attribute Single	Returns the value of the specified instance attribute.

The following table describes the instance attributes:

Attribute ID (hex)	Access	Name	Data Type	Value	Description
1	Get	Interface Speed	UDINT	–	Speed in Mbit/s (10 or 100)
2	Get	Interface Flags	DWORD	Bit level	• 0: Link status • 1: Half/full duplex • 2...4: Negotiation status • 5: Manual setting / requires reset • 6: Local hardware error detected All other bits are reserved and set to 0.
3	Get	Physical Address	ARRAY of 6 USINT	–	This array contains the MAC address of the product. Format: XX-XX-XX-XX-XX-XX

EtherNet/IP Interface Diagnostic Object (Class ID = 350 hex)

The following table describes the class attributes of the EtherNet/IP Interface Diagnostic object:

Attribute ID (hex)	Access	Name	Data Type	Value (hex)	Details
1	Get	Revision	UINT	01	Increased by 1 on each new update of the object.
2	Get	Max Instance	UINT	01	Maximum instance number of the object.

The following table describes the instance attributes of the EtherNet/IP Interface Diagnostic object:

Attribute ID (hex)	Access	Name	Data Type	Details
1	Get	Protocols supported	UINT	Protocol(s) supported (0=not supported, 1=supported): • Bit 0: EtherNet/IP • Bit 1: Modbus TCP • Bits 2...15: Reserved, 0
2	Get	Connection Diag	STRUCT of	
		Max CIP IO Connections opened	UINT	Maximum number of CIP I/O connections opened.
		Current CIP IO Connections	UINT	Number of CIP I/O connections currently opened.
		Max CIP Explicit Connections opened	UINT	Maximum number of CIP explicit connections opened.
		Current CIP Explicit Connections	UINT	Number of CIP explicit connections currently opened
		CIP Connections Opening Errors	UINT	Incremented on each unsuccessful attempt to open a CIP connection.
		CIP Connections Timeout Errors	UINT	Incremented when a CIP connection times out.
		Max EIP TCP Connections opened	UINT	Maximum number of TCP connections opened and used for EtherNet/IP communications.
		Current EIP TCP Connections	UINT	Number of TCP connections currently open and being used for EtherNet/IP communications.
3	Get Clear	IO Messaging Diag	STRUCT of	
		IO Production Counter	UDINT	Incremented each time a Class 0/1 CIP message is sent.
		IO Consumption Counter	UDINT	Incremented each time a Class 0/1 CIP message is received.
		IO Production Send Errors Counter	UINT	Incremented each Time a Class 0/1 message is not sent.
		IO Consumption Receive Errors Counter	UINT	Incremented each time a consumption is received that contains an error.
4	Get Clear	Explicit Messaging Diag	STRUCT of	
		Class3 Msg Send Counter	UDINT	Incremented each time a Class 3 CIP message is sent.
		Class3 Msg Receive Counter	UDINT	Incremented each time a Class 3 CIP message is received.
		UCMM Msg Send Counter	UDINT	Incremented each time a UCMM message is sent.
		UCMM Msg Receive Counter	UDINT	Incremented each time a UCMM message is received.
5	Get	Com Capacity	STRUCT of	
		Max CIP Connections	UINT	Maximum number of supported CIP connections.
		Max TCP Connections	UINT	Maximum number of supported TCP connections.
		Max Urgent priority rate	UINT	Maximum number of CIP transport class 0/1 Urgent priority message packets per second.
		Max Scheduled priority rate	UINT	Maximum number of CIP transport class 0/1 Scheduled priority message packets per second.
		Max High priority rate	UINT	Maximum number of CIP transport class 0/1 High priority message packets per second.
		Max Low priority rate	UINT	Maximum number of CIP transport class 0/1 Low priority message packets per second.
		Max Explicit Messaging rate	UINT	Max CIP transport class 2/3 or other EtherNet/IP messages packets per second

Attribute ID (hex)	Access	Name	Data Type	Details
6	Get	Bandwidth Diag	STRUCT of	
		Current sending Urgent priority rate	UINT	CIP transport class 0/1 Urgent priority message packets sent per second.
		Current reception Urgent priority rate	UINT	CIP transport class 0/1 Urgent priority message packets received per second.
		Current sending Scheduled priority rate	UINT	CIP transport class 0/1 Scheduled priority message packets sent per second.
		Current reception Scheduled priority rate	UINT	CIP transport class 0/1 Scheduled priority message packets received per second.
		Current sending High priority rate	UINT	CIP transport class 0/1 High priority message packets sent per second.
		Current reception High priority rate	UINT	CIP transport class 0/1 High priority message packets received per second.
		Current sending Low priority rate	UINT	CIP transport class 0/1 Low priority message packets sent per second.
		Current reception Low priority rate	UINT	CIP transport class 0/1 Low priority message packets received per second.
		Current sending Explicit Messaging rate	UINT	CIP transport class 2/3 or other EtherNet/IP message packets sent per second.
		Current reception Explicit Messaging rate	UINT	CIP transport class 2/3 or other EtherNet/IP message packets received per second.
7	Get	Modbus Diag	STRUCT of	
		Max. Modbus TCP Connections opened	UINT	Maximum number of TCP connections opened and used for Modbus communications.
		Current Modbus TCP Connections	UINT	Number of TCP connections currently opened and used for Modbus communications.
		Modbus TCP Msg Send Counter	UDINT	Incremented each time a Modbus TCP message is sent.
		Modbus TCP Msg Receive Counter	UDINT	Incremented each time a Modbus TCP message is received.

The following table describes the class services:

Service Code (hex)	Name	Description
01	Get_Attributes_All	Returns the value of all class attributes.
0E	Get_Attribute_Single	Returns the value of a specified attribute.
4C	Get_and_Clear	Gets and clears a specified attribute.

IOScanner Diagnostic Object (Class ID = 351 hex)

The following table describes the class attributes of the IOScanner Diagnostic object:

Attribute ID (hex)	Access	Name	Data Type	Value (hex)	Details
1	Get	Revision	UINT	1	Increased by 1 on each new update of the object.
2	Get	Max Instance	UINT	1	Maximum instance number of the object.

The following table describes the instance attributes of the IOScanner Diagnostic object:

Attribute ID (hex)	Access	Name	Data Type	Details
1	Get	IO Status Table	STRUCT of	
		Size	UINT	Size in bytes of the Status attribute.
		Status	ARRAY of UINT	I/O status. Bit n, where n is instance n of the object, provides the status of the I/O exchanged on the I/O connection: • 0: The input or output status of the I/O connection is in error, or no device. • 1: The input or output status of the I/O connection is correct.

The following table describes the class services:

Service Code (hex)	Name	Description
01	Get_Attributes_All	Returns the value of all class attributes.

IO Connection Diagnostic Object (Class ID = 352 hex)

The following table describes the class attributes of the IO Connection Diagnostic object:

Attribute ID (hex)	Access	Name	Data Type	Value (hex)	Details
1	Get	Revision	UINT	01	Increased by 1 on each new update of the object.
2	Get	Max Instance	UINT	01	Maximum instance number of the object. 0...n where n is the maximum number of CIP I/O connections. **NOTE:** There is an IO Connection Diagnostic object instance for both O->T and T->O paths.

The following table describes the instance attributes of the I/O Connection Diagnostic object:

Attribute ID (hex)	Access	Name	Data Type	Details
1	Get Clear	IO Com Diag	STRUCT of	
		IO Production Counter	UDINT	Incremented each time a production is sent.
		IO Consumption Counter	UDINT	Incremented each time a consumption is received.
		IO Production Send Errors Counter	UINT	Incremented each time a production is not sent due to an error.
		IO Consumption Receive Errors Counter	UINT	Incremented each time a consumption is received that contains an error.
		CIP Connection TimeOut Errors	UINT	Incremented each time a connection times out.
		CIP Connection Opening Errors	UINT	Incremented on each unsuccessful attempt to open a connection.
		CIP Connection State	UINT	State of the CIP IO connection.
		CIP Last Error General Status	UINT	General status of the last error detected on the connection.
		CIP Last Error Extended Status	UINT	Extended status of the last error detected on the connection.
		Input Com Status	UINT	Communication status of the inputs.
		Output Com Status	UINT	Communication status of the outputs.
2	Get	Connection Diag	STRUCT of	
		Production Connection ID	UDINT	Connection ID for production.
		Consumption Connection ID	UDINT	Connection ID for consumption.
		Production RPI	UDINT	Requested Packet Interval (RPI) for productions, in µs.
		Production API	UDINT	Actual Packet Interval (API) for productions.
		Consumption RPI	UDINT	RPI for consumptions.
		Consumption API	UDINT	API for consumptions.
		Production Connection Parameters	UDINT	Connection parameters for productions.
		Consumption Connection Parameters	UDINT	Connection parameters for consumptions.
		Local IP	UDINT	Local IP address for I/O communication.
		Local UDP Port	UINT	Local UDP port number for I/O communication.
		Remote IP	UDINT	Remote IP address for I/O communication.
		Remote UDP Port	UINT	Remote UDP port number for I/O communication.
		Production Multicast IP	UDINT	Multicast IP address for productions, or 0 if multicast is not used.
		Consumption Multicast IP	UDINT	Multicast IP address for consumptions, or 0 if multicast is not used.
		Protocols supported	UINT	Protocol(s) supported (0=not supported, 1= supported): • Bit 0: EtherNet/IP • Bit 1: Modbus TCP • Bit 2: Modbus Serial • Bits 3...15: Reserved, 0

Instance Attributes

The following table describes the class services:

Service Code (hex)	Name	Description
01	Get_Attributes_All	Returns the value of all class attributes.
0E	Get_Attribute_Single	Returns the value of the specified attribute.
4C	Get_and_Clear	Gets and clears a specified attribute.

Explicit Connection Diagnostic Object (Class ID = 353 hex)

The following table describes the class attributes of the Explicit Connection Diagnostic object:

Attribute ID (hex)	Access	Name	Data Type	Value (hex)	Details
1	Get	Revision	UINT	01	Increased by 1 at each new update of the object.
2	Get	Max Instance	UINT	0...n (maximum number of CIP IO connections)	Maximum instance number of the object.

The following table describes the instance attributes of the Explicit Connection Diagnostic object:

Attribute ID (hex)	Access	Name	Data Type	Details
1	Get	Originator Connection ID	UDINT	O to T Connection ID
2	Get	Originator IP	UDINT	–
3	Get	Originator TCP Port	UINT	–
4	Get	Target Connection ID	UDINT	T to O Connection ID
5	Get	Target IP	UDINT	–
6	Get	Target TCP Port	UINT	–
7	Get	Msg Send Counter	UDINT	Incremented each time a Class 3 CIP Message is sent on the connection
8	Get	Msg ReceiveCounter	UDINT	Incremented each time a Class 3 CIP Message is received on the connection.

Explicit Connections Diagnostic List Object (Class ID = 354 hex)

The following table describes the class attributes of the Explicit Connections Diagnostic List object:

Attribute ID (hex)	Access	Name	Data Type	Value (hex)	Details
1	Get	Revision	UINT	01	Increased by 1 at each new update of the object.
2	Get	Max Instance	UINT	0...n	n is the maximum number of concurrent list accesses supported.

The following table describes the instance attributes of the Explicit Connections Diagnostic List object:

Attribute ID (hex)	Access	Name	Data Type	Details
1	Get	Number of Connections	UINT	Total number of open Explicit connections.
2	Get	Explicit Messaging Connections Diagnostic List	ARRAY of STRUCT	Contents of instantiated Explicit Connection Diagnostic objects.
		Originator Connection ID	UDINT	Originator to Target connection ID.
		Originator IP	UDINT	Originator to Target IP address.
		Originator TCP Port	UINT	Originator to Target port number.
		Target Connection ID	UDINT	Target to Originator connection ID.
		Target IP	UDINT	Target to Originator IP address.
		Target TCP Port	UINT	Target to Originator port number.
		Msg Send Counter	UDINT	Incremented each time a Class 3 CIP message is sent on the connection.
		Msg Receive Counter	UDINT	Incremented each time a Class 3 CIP message is sent on the connection.

The following table describes the class services:

Service Code (hex)	Name	Description
08	Create	Creates an instance of the Explicit Connections Diagnostic List object.
09	Delete	Deletes an instance of the Explicit Connections Diagnostic List object.
33	Explicit_Connections_ Diagnostic_Read	Explicit connections diagnostic read object.

Controller as a Slave Device on Modbus TCP

Overview

This section describes the configuration of the M262 Logic/Motion Controller as a **Modbus TCP Slave Device**.

Each M262 Logic/Motion Controller uses a Modbus server which does not need to be configured. The **Modbus TCP Slave Device** adds another Modbus server function to the controller. This server is addressed by the Modbus client application by specifying a configured Unit ID (Modbus address) in the range 1...247. The embedded Modbus server of the slave controller needs no configuration, and is addressed by specifying a Unit ID equal to 255. Refer to Modbus TCP Configuration, page 188.

To configure your M262 Logic/Motion Controller as a **Modbus TCP Slave Device**, you must add **Modbus TCP Slave Device** functionality to your controller (see Adding a Modbus TCP Slave Device thereafter). This functionality creates a specific I/O area in the controller that is accessible with the Modbus TCP protocol.

This I/O area is used whenever an external master needs to access the *%IW* and *%QW* objects of the controller. This **Modbus TCP Slave Device** functionality allows you to furnish to this area the controller I/O objects which can then be accessed with a single Modbus read/write registers request.

Inputs/outputs are seen from the slave controller: inputs are written by the master, and outputs are read by the master.

The **Modbus TCP Slave Device** can define a privileged Modbus client application, whose connection is not forcefully closed (embedded Modbus connections may be closed when more than 8 connections are needed).

The watchdog associated to the privileged connection allows you to verify whether the controller is being polled by the privileged master. If no Modbus request is received within the timeout duration, the diagnostic information *i_byMasterIpLost* is set to 1 (TRUE). For more information, refer to the Ethernet Port Read-Only System Variables (see Modicon M262 Logic/Motion Controller, System Functions and Variables, System Library Guide).

For further information about Modbus TCP, refer to the www.modbus.org website.

Adding a Modbus TCP Slave Device

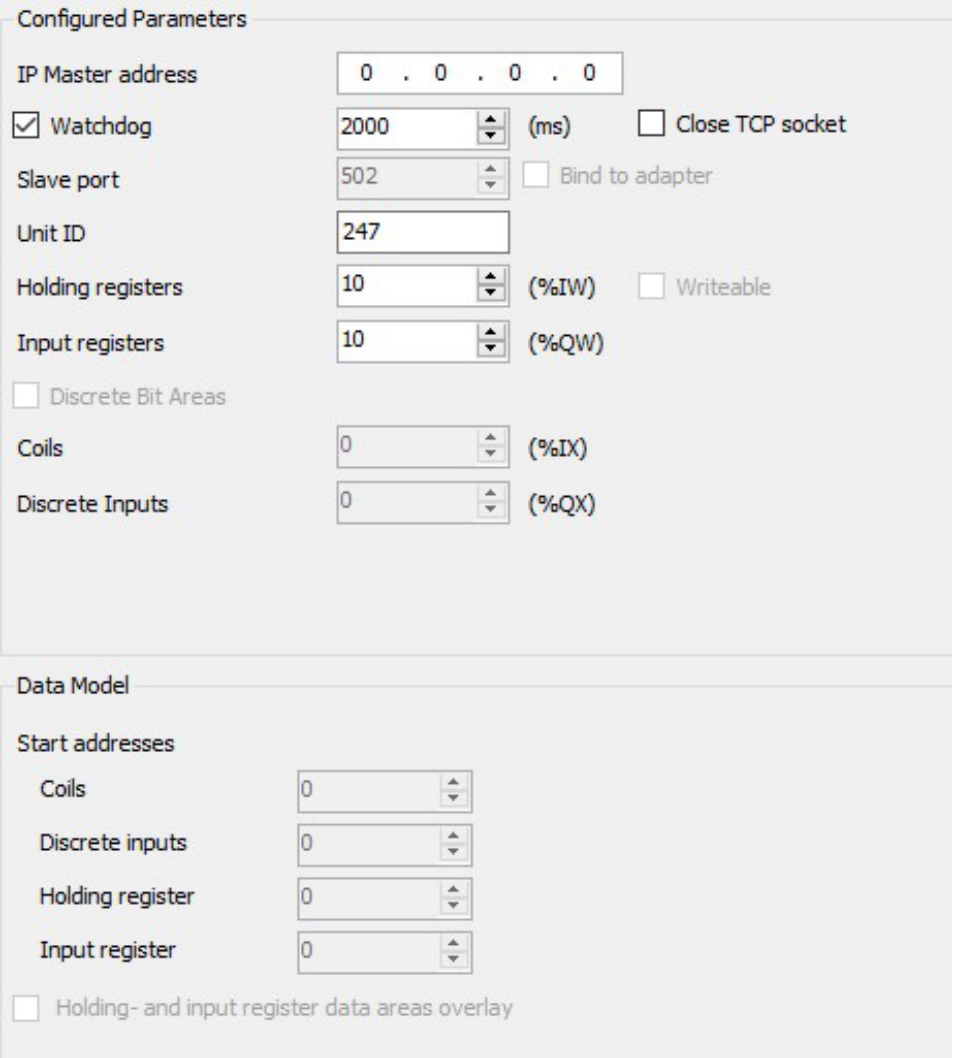
To configure your M262 Logic/Motion Controller as a Modbus TCP slave device, you must:

Step	Action
1	Select **Modbus TCP Slave Device** in the **Hardware Catalog**.
2	Drag and drop it to the **Devices tree** on one of the highlighted nodes. **NOTE:** If the chosen node is **COM_Bus**, a TMSES4 expansion module is automatically added to your configuration. For more information on adding a device to your project, refer to: • Using the Hardware Catalog found in the EcoStruxure Machine Expert Programming Guide • Using the Contextual Menu or Plus Button found in the EcoStruxure Machine Expert Programming Guide

Modbus TCP Configuration

To configure the Modbus TCP slave device, double-click **ModbusTCP_Slave_Device** in the **Devices tree**.

This dialog box appears:



Element	Description
IP Master Address	IP address of the Modbus master The connections are not closed on this address.
Watchdog	Watchdog in 500 ms increments **NOTE:** The watchdog applies to the **IP Master Address** unless the address is 0.0.0.0.
Close TCP socket	When **Close TCP socket** is selected, the TCP socket is closed if the **Watchdog** is enabled and the set time is exceeded.
Slave Port	Modbus communication port (502) **NOTE:** The port number can be modified using the changeModbusPort, page 191, script command.
Unit ID	Sends the requests to the Modbus TCP slave device (1...247), instead of to the embedded Modbus server (255).
Holding Registers (% IW)	Number of %IW registers to be used in the exchange (2...120) (each register is 2 bytes)
Input Registers (% QW)	Number of %QW registers to be used in the exchange (2...120) (each register is 2 bytes)

Modbus TCP Slave Device I/O Mapping Tab

The I/Os are mapped to Modbus registers from the master perspective as follows:

- %IWs are mapped from register 0 to n-1 and are R/W (n = Holding register quantity, each %IW register is 2 bytes).
- %QWs are mapped from register n to n+m -1 and are read only (m = Input registers quantity, each %QW register is 2 bytes).

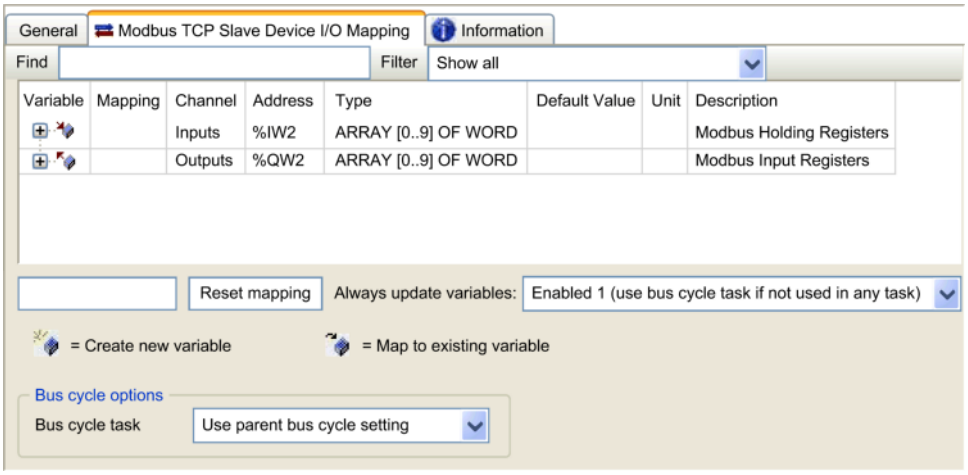
Once a **Modbus TCP Slave Device** has been configured, Modbus commands sent to its Unit ID (Modbus address) access the *%IW* and *%QW* objects of the controller instead of the regular Modbus words (accessed when the Unit ID is 255). This facilitates read/write operations by a Modbus TCP IOScanner application.

The **Modbus TCP Slave Device** responds to a subset of the Modbus commands, but does so in a way that differs from Modbus standards, and with the purpose of exchanging data with the external I/O scanner. The following Modbus commands are supported by the Modbus TCP slave device:

Function Code Dec (Hex)	Function	Comment
3 (3)	Read holding register	Allows the master to read %IW and %QW objects of the device
6 (6)	Write single register	Allows the master to write %IW objects of the device
16 (10)	Write multiple registers	Allows the master to write %IW objects of the device
23 (17)	Read/write multiple registers	Allows the master to read %IW and %QW objects of the device and write %IW objects of the device
Other	Not supported	–

NOTE: Modbus requests that attempt to access registers above n+m-1 are answered by the 02 - ILLEGAL DATA ADDRESS exception code.

To link I/O objects to variables, select the **Modbus TCP Slave Device I/O Mapping** tab:



Channel		Type	Description
Input	IW0	WORD	Holding register 0
	...	...	...
	IWx	WORD	Holding register x
Output	QW0	WORD	Input register 0
	...	...	...
	QWy	WORD	Input register y

The number of words depends on the **Holding Registers (%IW)** and **Input Registers (%QW)** parameters of the **Modbus TCP** tab.

NOTE: Output means OUTPUT from client/master controller (= %IW for the server/slave controller). Input means INPUT from client/master controller (= % QW for the server/slave controller).

For the parameter **Always update variables**, choose one of the following options:

- **Use parent device setting**
- **Enabled 1 (use bus cycle task if not used in any task)** (default setting)
- **Enabled 2 (always in bus cycle task)**

Bus Cycle Options

In the **Modbus TCP Slave Device I/O Mapping** tab, select the **Bus cycle task** to use:

- **Use parent bus cycle setting** (default setting)
- **MAST**
- **An existing task of the project**: you can select an existing task and associate it to the scanner. For more information about the application tasks, refer to the EcoStruxure Machine Expert Programming Guide.

NOTE: There is a corresponding **Bus cycle task** parameter in the I/O mapping editor of the device that contains the **Modbus TCP Slave Device**. This parameter defines the task responsible for refreshing the *%IW* and *%QW* registers.

Changing the Modbus TCP Port

changeModbusPort Command

The *changeModbusPort* command can be used to change the port used for data exchanges with a Modbus TCP master.

The Modbus **Slave Port** is displayed on the Modbus TCP configuration window, page 187.

The default Modbus port number is 502.

Command	Description
`changeModbusPort` *"portnum"*	*portnum* is the new Modbus port number to use and is passed as a string of characters. Before running the command, refer to Used Ports, page 162 to ensure that *portnum* is not being used by another TCP/UDP protocol or process. An error is logged in the */usr/Syslog/FWLog.txt* file if the specified port number is already in use.

To limit the number of open sockets, the *changeModbusPort* command can only be run twice.

A power cycle of the logic controller returns the Modbus port number to the default value (502). The *changeModbusPort* command must therefore be executed after each power cycle.

NOTE: After changing the port number, protocol active selection for the Modbus Server in the **Security Parameters** group on the Ethernet Configuration window, page 123 is no longer valid.

Running the Command from an SD Card Script

Step	Action

1	Create a script file, page 246, for example:
	`; Change Modbus slave port`
	`changeModbusPort "1502";`

2	Name the script file *Script.cmd*.
3	Copy the script file to the SD card.
4	Insert the SD card in the controller.

Running the Command Using ExecuteScript Function Block

The *changeModbusPort* command can be run from within an application using the ExecuteScript function block (see Modicon M262 Logic Controller, System Functions and Variables, System Library Guide).

The following sample code changes the Modbus TCP slave port from the default (502) to 1502.

```
IF (myBExe = FALSE AND (PortNum <> 502)) THEN

   myExecSc( // falling edge for a second change
   xExecute:=FALSE ,
   sCmd:=myCmd ,
   xDone=>myBDone ,
   xBusy=> myBBusy,
   xError=> myBErr,
   eError=> myIerr);
   string1 := 'changeModbusPort "';
   string2 := WORD_TO_STRING(PortNum);
   myCmd := concat(string1,string2);
   myCmd := concat(myCmd,'"');
   myBExe := TRUE;
END_IF

myExecSc(
xExecute:=myBExe ,
sCmd:=myCmd ,
xDone=>myBDone ,
xBusy=> myBBusy,
xError=> myBErr,
eError=> myIerr);
```

Sercos Configuration

Introduction

This chapter describes how to configure the Sercos interface of the Modicon M262 Motion Controller.

Overview of the Sercos Standard

Introduction

The Sercos interface is a standardized interface (IEC 61491) for real-time communication between controllers, drives, servo drives, I/O devices, encoders, and other equipment requiring real-time services.

For motion control, the Sercos standard describes the internationally standardized digital interface for communication between a control unit and associated servo drives. It defines standardization of operating data, parameters, and scaling for machines with multiple drives that can be operated in torque, velocity, or position interface operation modes.

The main features of the Sercos interface are:

- Ring topology (redundancy)
- Master / slave system
- Baud rate 100 MBaud
- Minimum synchronization time of 1 ms (4 axes or 8 axes), 2 ms (16 axes), or 4 ms (24 axes)
- Synchronization (jitter < 1 µs)

Data Exchange

Communication with Sercos interface is divided into two types:

- Cyclical communication:

  The cyclical communication is used for exchanging real-time data (for example, position) and is executed once in every communication cycle (*CycleTime*). Certain specified data are transferred from the controller to all drives and from all drives to the controller in every cycle.

  The exchange of information between the motion controller (Sercos master) and the servo drives (slaves), is accomplished via a message structure known as a telegram. There are three telegrams defined by IEC 61491:

  - MST (Master Synchronization Telegram): An MST telegram is broadcast by the master at the beginning of each transmission cycle to synchronize the timing of the cycle.
  - MDT (Master Data Telegram): An MDT telegram is sent by the master once during each transmission cycle to transmit data (command values) to the servo drives (slaves).
  - AT (Acknowledge Telegram): AT telegrams are sent by the slaves to the master (feedback values).
- Non-cyclical communication with function blocks.

  Non-cyclical communication is used to exchange data such as parameters for configuring communication, the drive parameters, status, and so on, where time is not a critical factor. The controller controls non-cyclical communication. All of the parameters in the system can be contacted using this channel, even parameters that are configured cyclically.

NOTE: The two types of communication can be used simultaneously.

IDN Description

IEC 61491 assigns identification numbers (IDNs) to all the operation data in a Sercos drive. Operation data includes parameters, interface procedure commands, and command and feedback values.

There are two categories of IDNs available:

- Standards IDNs (S): They are defined by the Sercos standard IEC 61491. Standards IDNs, if supported by a Sercos drive, behave the same, irrespective of the drive manufacturer;
- Proprietary IDNs (P): They are reserved for product-specific data that can be defined by the manufacturers of control units and servo drives.

Modicon M262 Logic/Motion Controller Sercos Configuration

Introduction

For more information about Modicon M262 Logic/Motion Controller Sercos configuration, refer to M262 Sercos For Modicon M262 Motion Controller - User Guide (see Sercos For Modicon M262 Motion Controller - User Guide).

Modicon M262 Motion Controller and Safety Controllers with Sercos

Introduction

Sercos fieldbus allows to connect Safety controllers. For more information, refer to the M262 Embedded Safety - Integration Guide (see M262 Embedded Safety - Integration Guide).

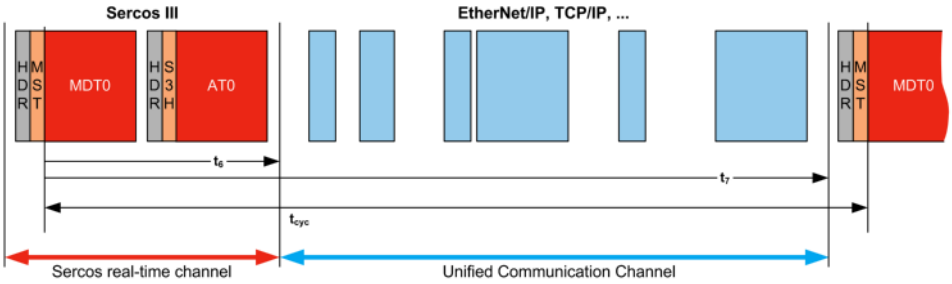
Single Wire Architecture

Overview

In addition to real-time and safety-critical data, the Sercos standard allows the transmission of Ethernet data over a common network infrastructure.

NOTE: The TM262M05MESS8T motion controller does not support the Single Wire Architecture.

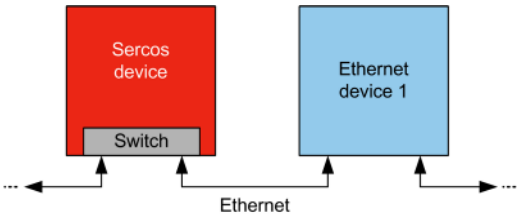
The EtherNet/IP or TCP/IP frames are embedded within the Sercos frame:



This Single Wire Architecture can be implemented using a single network cable connected to the controller. The Ethernet devices are added to the end of the cable after the Sercos devices.

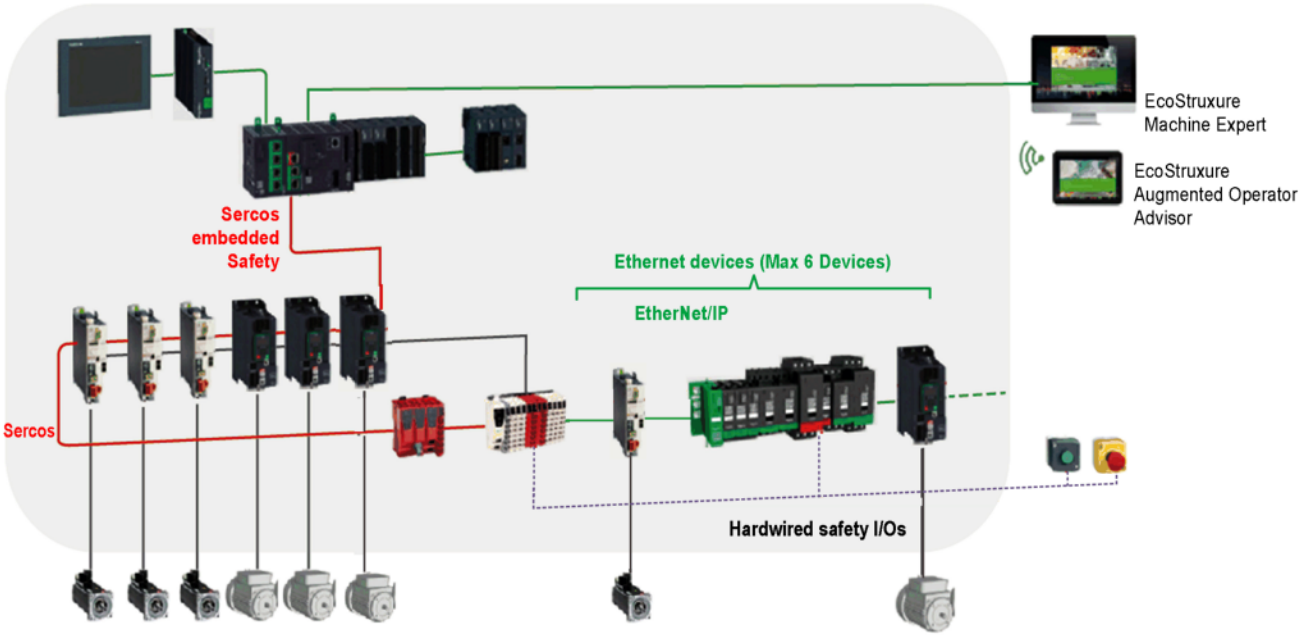
No additional cables or network components (gateways or switches) are required.

The last Sercos device on the cable acts as the gateway. It must have two Sercos connectors; one connected to the upstream Sercos devices, the other to the downstream Ethernet devices:



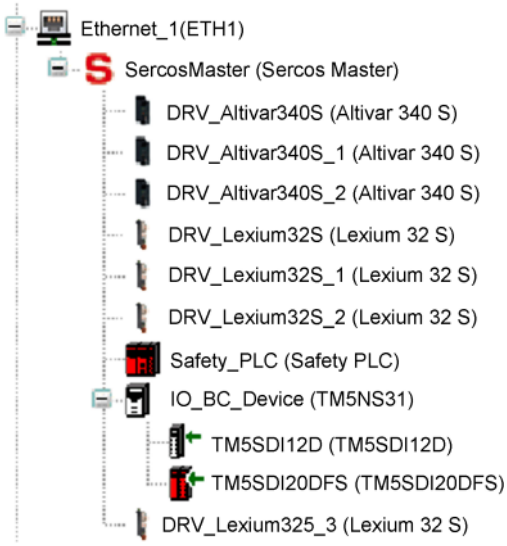
Up to 6 Ethernet devices can be added to the cable.

This figure shows an example Single Wire Architecture:

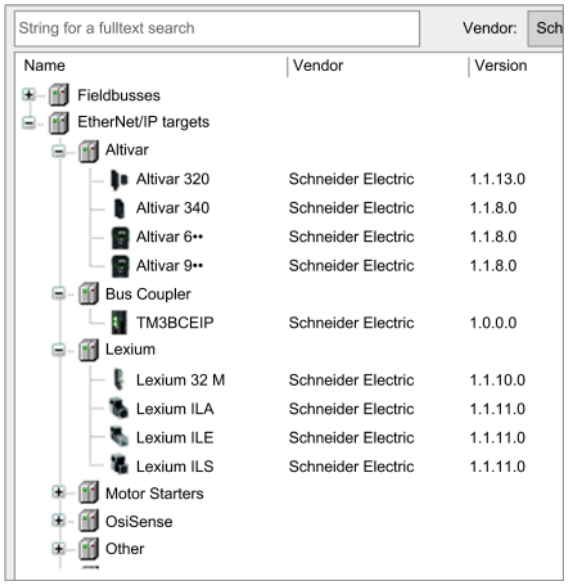


Single Wire Architecture Implementation

This figure shows the implementation of the example Single Wire Architecture in EcoStruxure Machine Expert:



To build this configuration:

Step	Action
1	Add the **Sercos Master** node and Sercos devices in the normal way.
2	Add up to a maximum of 6 Ethernet devices **below** the last Sercos device. Any of the Ethernet target devices available in the **Device editor** window can be added: 
3	Set the Sercos bus to the *Phase 4* state to activate Ethernet communication. When commissioning Sercos devices, it may be necessary to downgrade the Sercos phase, for example, by adjusting the **Communication Cycle Time** parameter in the Sercos device). In this case, the Ethernet devices will enter a fallback state.

Serial Line Configuration

Introduction

This chapter describes how to configure the serial line communication of the Modicon M262 Logic/Motion Controller.

Serial Line Configuration

Introduction

The Serial Line configuration window allows you to configure the physical parameters of a serial line (baud rate, parity, and so on).

Serial Line Configuration

To configure a Serial Line, double-click **Serial line** in the **Devices tree**.

The following parameters must be identical for each serial device connected to the port.

Element	Description
Baud rate	Transmission speed in bits/s
Parity	Used for error detection
Data bits	Number of bits for transmitting data
Stop bits	Number of stop bits
Physical Medium	Specify the medium to use: • RS485 (using polarisation resistor or not) • RS232
Polarization Resistor	Polarization resistors are integrated in the controller. They are switched on or off by this parameter.

The serial line ports of your controller are configured for the CoDeSys protocol by default when new or when you update the controller firmware. The CoDeSys protocol is incompatible with that of other protocols such as Modbus Serial Line. Connecting a new controller to, or updating the firmware of a controller connected to, an active Modbus configured serial line can cause the other devices on the serial line to stop communicating. Make sure that the controller is not connected to an active Modbus serial line network before first downloading a valid application having the concerned port or ports properly configured for the intended protocol.

NOTICE

INTERRUPTION OF SERIAL LINE COMMUNICATIONS

Be sure that your application has the serial line ports properly configured for Modbus before physically connecting the controller to an operational Modbus Serial Line network.

Failure to follow these instructions can result in equipment damage.

This table indicates the maximum baud rate value of the managers:

Manager	Maximum Baud Rate (Bits/s)
Network Manager	115200
Modbus Manager	
ASCII Manager	
Modbus IOScanner	

Serial Line Diagnostic Table

To access the **Serial Line Diagnostic Table**, double click the **Serial Line** node in the **Tools tree** tab. Diagnostic information are accessible with the structure **SERDIAG_W_STRUCT**. For more information, refer to Modicon M262 Logic/Motion Controller, System Functions and Variables, System Library Guide (see Modicon M262 Logic/Motion Controller, System Functions and Variables, System Library Guide).

Network Manager

Introduction

Use the Network Manager to exchange variables with a Magelis Advanced Panel with CoDeSys protocol, or when the Serial Line is used for EcoStruxure Machine Expert programming.

Adding the Manager

To add a Network Manager to your controller, select the **Machine Expert-Network Manager** in the **Hardware Catalog**, drag it to the **Devices tree**, and drop it on one of the highlighted nodes.

For more information on adding a device to your project, refer to:

• Using the Hardware Catalog found in the EcoStruxure Machine Expert Programming Guide

• Using the Contextual Menu or Plus Button found in the EcoStruxure Machine Expert Programming Guide

Configuring the Manager

There is no configuration for the Network Manager.

Adding a Modem

To add a modem to the Network Manager, refer to Adding a Modem to a Manager, page 212.

Modbus Manager

Introduction

The Modbus Manager is used for Modbus RTU or ASCII protocol in master or slave mode.

Adding the Manager

To add a Modbus manager to your controller, select the **Modbus Manager** in the **Hardware Catalog**, drag it to the **Devices tree**, and drop it on one of the highlighted nodes.

For more information on adding a device to your project, refer to:

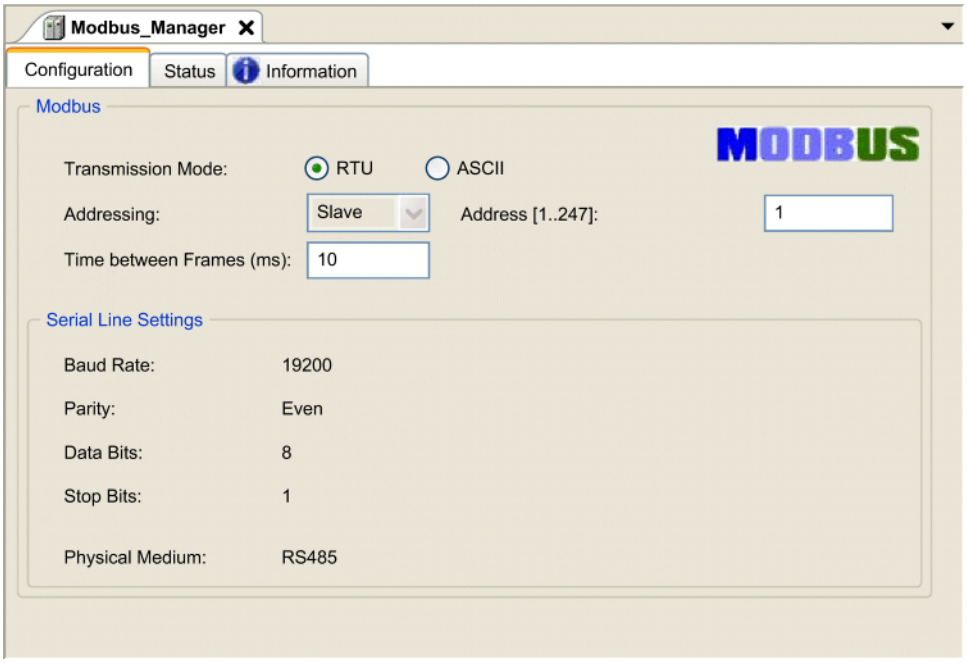
• Using the Hardware Catalog found in the EcoStruxure Machine Expert Programming Guide

• Using the Contextual Menu or Plus Button found in the EcoStruxure Machine Expert Programming Guide

Modbus Manager Configuration

To configure the Modbus Manager of your controller, double-click **Modbus Manager** in the **Devices tree**.

The Modbus Manager configuration window is displayed as below:



Set the parameters as described in this table:

Element	Description
Transmission Mode	Specify the transmission mode to use: • RTU: uses binary coding and CRC error-checking (8 data bits) • ASCII: messages are in ASCII format, LRC error-checking (7 data bits) Set this parameter identical for each Modbus device on the link.
Addressing	Specify the device type: • Master • Slave
Address	Modbus address of the device, when slave is selected.
Time between Frames (ms)	Time to avoid bus-collision. Set this parameter identical for each Modbus device on the link.
Serial Line Settings	Parameters specified in the Serial Line configuration window.

Modbus Master

When the controller is configured as a Modbus Master, the following function blocks are supported from the PLCCommunication Library:

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

For further information, see Function Block Descriptions (see Modbus and ASCII Read/Write Functions, PLCCommunication Library Guide).

Modbus Slave

When the controller is configured as Modbus Slave, the following Modbus requests are supported:

Function Code Dec (Hex)	Sub-Function Dec (Hex)	Function
1 (1 hex)	–	Read digital outputs (%Q)
2 (2 hex)	–	Read digital inputs (%I)
3 (3 hex)	–	Read multiple register (%MW)
6 (6 hex)	–	Write single register (%MW)
8 (8 hex)	–	Diagnostic
15 (F hex)	–	Write multiple digital outputs (%Q)
16 (10 hex)	–	Write multiple registers (%MW)
23 (17 hex)	–	Read/write multiple registers (%MW)
43 (2B hex)	14 (E hex)	Read device identification

This table contains the sub-function codes supported by the diagnostic Modbus request 08:

Sub-Function Code		Function
Dec	**Hex**	
10	0A	Clears Counters and Diagnostic Register
11	0B	Returns Bus Message Count
12	0C	Returns Bus Communication Error Count
13	0D	Returns Bus Exception Error Count
14	0E	Returns Slave Message Count
15	0F	Returns Slave No Response Count
16	10	Returns Slave NAK Count
17	11	Returns Slave Busy Count
18	12	Returns Bus Character Overrun Count

This table lists the objects that can be read with a read device identification request (basic identification level):

Object ID	Object Name	Type	Value
00 hex	Vendor name	ASCII String	Schneider Electric
01 hex	Product code	ASCII String	Controller reference
02 hex	Major / Minor revision	ASCII String	aa.bb.cc.dd (same as device descriptor)

The following section describes the differences between the Modbus memory mapping of the controller and HMI Modbus mapping. If you do not program your application to recognize these differences in mapping, your controller and HMI will not communicate correctly. Thus it will be possible for incorrect values to be written to memory areas responsible for output operations.

⚠ WARNING

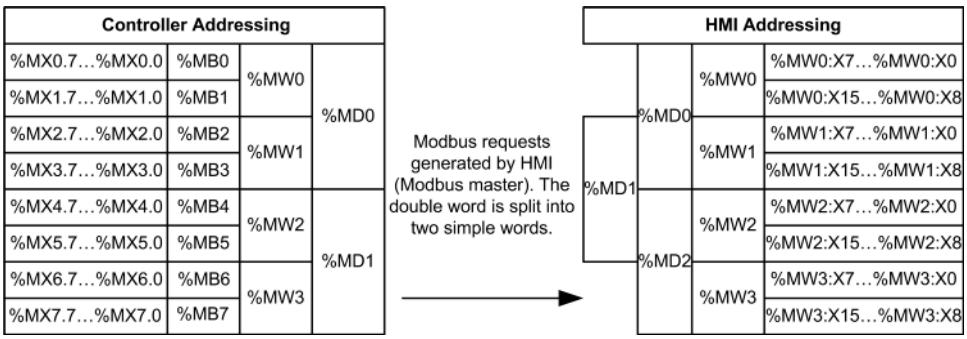
UNINTENDED EQUIPMENT OPERATION

Program your application to translate between the Modbus memory mapping used by the controller and that used by any attached HMI devices.

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

When the controller and the Magelis HMI are connected via Modbus (HMI is master of Modbus requests), the data exchange uses simple word requests.

There is an overlap on simple words of the HMI memory while using double words but not for the controller memory (see following diagram). In order to have a match between the HMI memory area and the controller memory area, the ratio between double words of HMI memory and the double words of controller memory has to be 2.



The following gives examples of memory match for the double words:

- %MD2 memory area of the HMI corresponds to %MD1 memory area of the controller because the same simple words are used by the Modbus request.
- %MD20 memory area of the HMI corresponds to %MD10 memory area of the controller because the same simple words are used by the Modbus request.

The following gives examples of memory match for the bits:

- %MW0:X9 memory area of the HMI corresponds to %MX1.1 memory area of the controller because the simple words are split in 2 distinct bytes in the controller memory.

Adding a Modem

To add a Modem to the Modbus Manager, refer to Adding a Modem to a Manager, page 212.

ASCII Manager

Introduction

The ASCII manager is used on a Serial Line, to transmit and/or receive data with a simple device.

Adding the Manager

To add an ASCII manager to your controller, select the **ASCII Manager** in the **Hardware Catalog**, drag it to the **Devices tree**, and drop it on one of the highlighted nodes.

For more information on adding a device to your project, refer to:

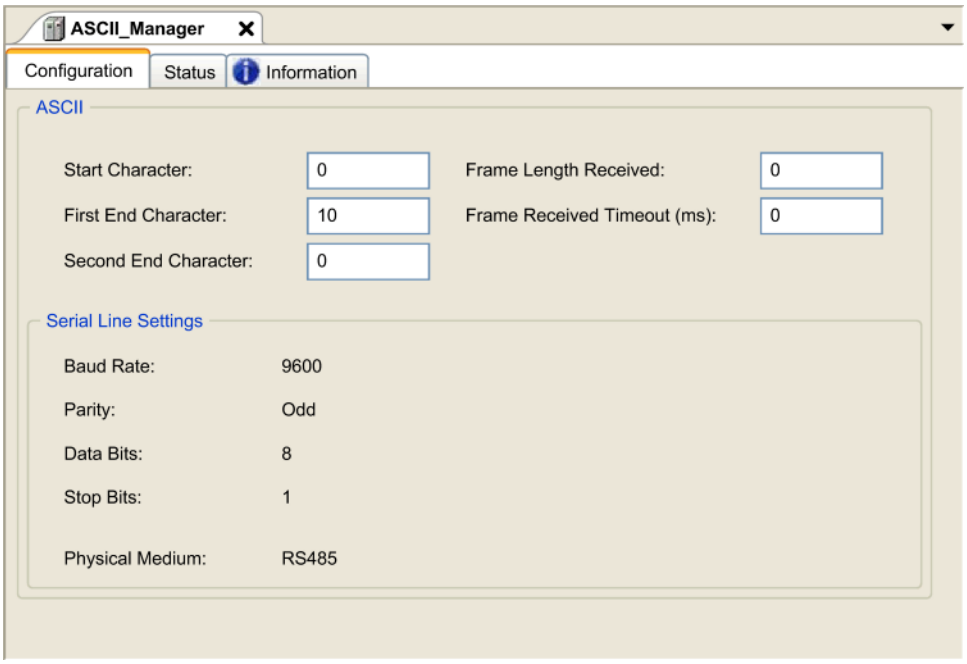
• Using the Hardware Catalog found in the EcoStruxure Machine Expert Programming Guide

• Using the Contextual Menu or Plus Button found in the EcoStruxure Machine Expert Programming Guide

ASCII Manager Configuration

To configure the ASCII manager of your controller, double-click **ASCII Manager** in the **Devices tree**.

The ASCII Manager configuration window is displayed as below:



Set the parameters as described in this table:

Parameter	Description
Start Character	If 0, no start character is used in the frame. Otherwise, in **Receiving Mode**, the corresponding character in ASCII is used to detect the beginning of a frame. In **Sending Mode**, this character is added at the beginning of the frame.
First End Character	If 0, no first end character is used in the frame. Otherwise, in **Receiving Mode**, the corresponding character in ASCII is used to detect the end of a frame. In **Sending Mode**, this character is added at the end of the frame.
Second End Character	If 0, no second end character is used in the frame. Otherwise, in **Receiving Mode**, the corresponding character in ASCII is used to detect the end of a frame. In **Sending Mode**, this character is added at the end of the frame.
Frame Length Received	If 0, this parameter is not used. This parameter allows the system to conclude an end of frame at reception when the controller received the specified number of characters. **Note:** This parameter cannot be used simultaneously with **Frame Received Timeout (ms)**.
Frame Received Timeout (ms)	If 0, this parameter is not used. This parameter allows the system to conclude the end of frame at reception after a silence of the specified number of ms.
Serial Line Settings	Parameters specified in the Serial Line configuration window, page 197.

NOTE: In the case of using several frame termination conditions, the first condition to be TRUE terminates the exchange.

Adding a Modem

To add a Modem to the ASCII manager, refer to Adding a Modem to a Manager, page 212.

Modbus Serial IOScanner

Introduction

The Modbus IOScanner is used to simplify exchanges with Modbus slave devices.

Add a Modbus IOScanner

To add a Modbus IOScanner on a Serial Line, select the **Modbus IOScanner** in the **Hardware Catalog**, drag it to the **Devices tree**, and drop it on one of the highlighted nodes.

For more information on adding a device to your project, refer to:

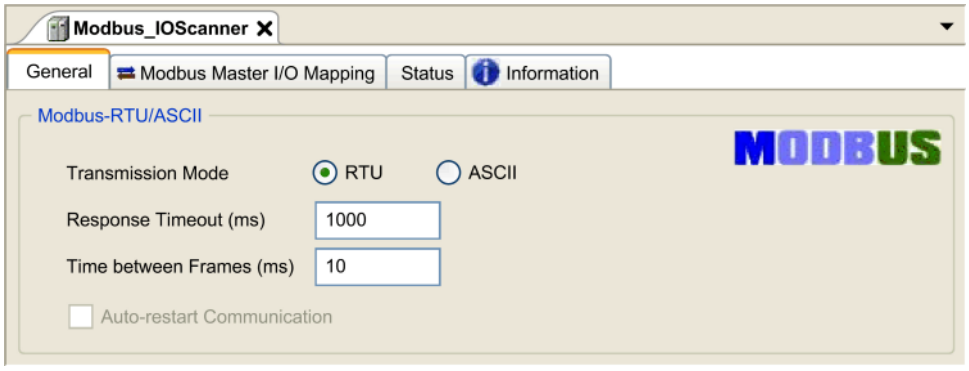
• Using the Hardware Catalog found in the EcoStruxure Machine Expert Programming Guide

• Using the Contextual Menu or Plus Button found in the EcoStruxure Machine Expert Programming Guide

Modbus IOScanner Configuration

To configure a Modbus IOScanner on a Serial Line, double-click **Modbus IOScanner** in the **Devices tree**.

The configuration window is displayed as below:



Set the parameters as described in this table:

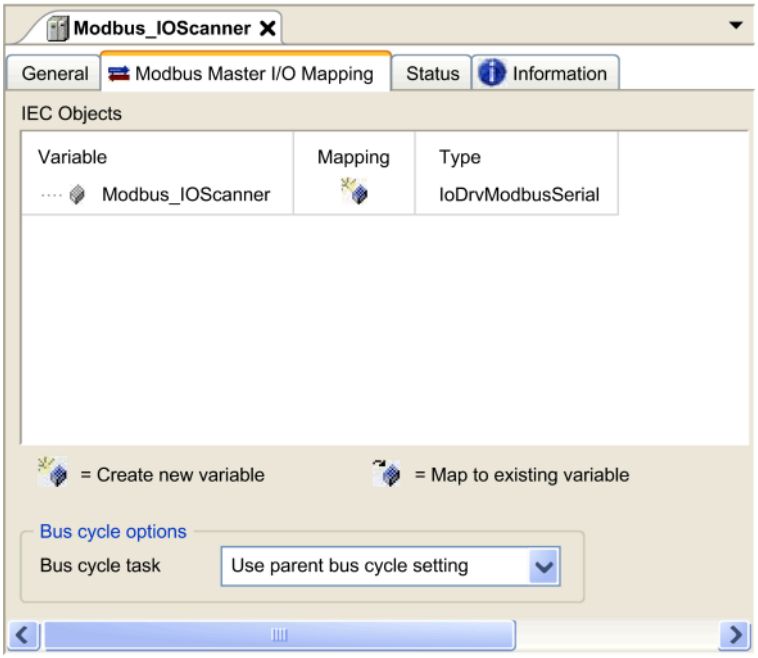
Element	Description
Transmission Mode	Specifies the transmission mode to use: • RTU: uses binary coding and CRC error-checking (8 data bits) • ASCII: messages are in ASCII format, LRC error-checking (7 data bits) Set this parameter identical for each Modbus device on the network.
Response Timeout (ms)	Timeout used in the exchanges.
Time between Frames (ms)	Delay to reduce data collision on the bus. Set this parameter identical for each Modbus device on the network.

NOTE: Do not use function blocks of the PLCCommunication library on a serial line with a Modbus IOScanner configured. This disrupts the Modbus IOScanner exchange.

Bus Cycle Task Selection

The Modbus IOScanner and the devices exchange data at each cycle of the chosen application task.

To select this task, select the **Modbus Master IO Mapping** tab. The configuration window is displayed as below:



The **Bus cycle task** parameter allows you to select the application task that manages the scanner:

- **Use parent bus cycle setting**: associate the scanner with the application task that manages the controller.
- **MAST**: associate the scanner with the MAST task.
- Another existing task: you can select an existing task and associate it to the scanner. For more information about the application tasks, refer to Task Configuration found in the EcoStruxure Machine Expert Programming Guide.

The scan time of the task associated with the scanner must be less than 500 ms.

Adding a Device on the Modbus Serial IOScanner

Introduction

This section describes how to add a device on the Modbus IOScanner.

Adding a Device on the Modbus IOScanner

To add a device on the Modbus IOScanner, select the **Generic Modbus Slave** in the **Hardware Catalog**, drag it to the **Devices tree**, and drop it on the **Modbus_ IOScanner** node of the **Devices tree**.

For more information on adding a device to your project, refer to:

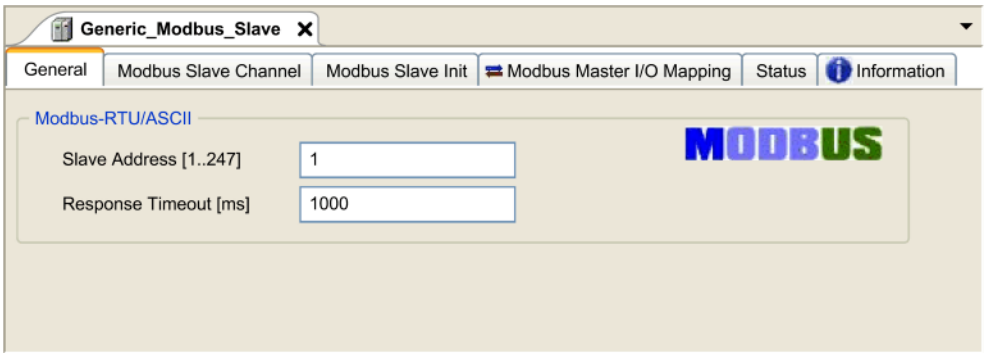
• Using the Hardware Catalog found in the EcoStruxure Machine Expert Programming Guide

• Using the Contextual Menu or Plus Button found in the EcoStruxure Machine Expert Programming Guide

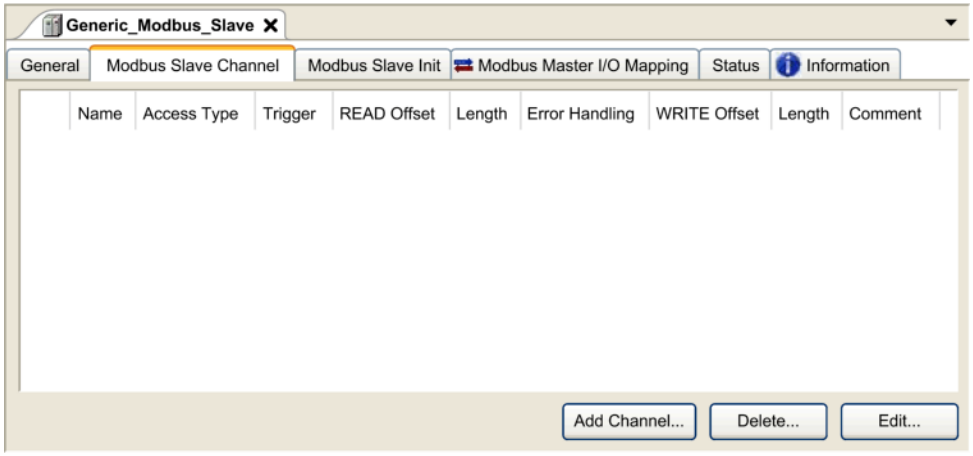
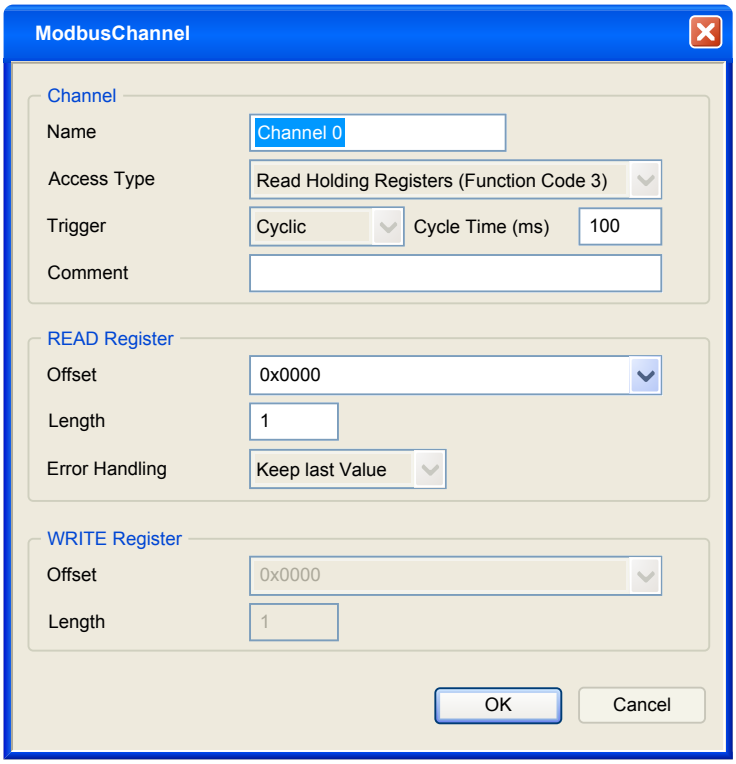
NOTE: The variable for the exchange is automatically created in the `%IWx` and `%QWx` of the **Modbus Serial Master I/O Mapping** tab.

Configuring a Device Added on the Modbus IOScanner

To configure the device added on the Modbus IOScanner, proceed as follows:

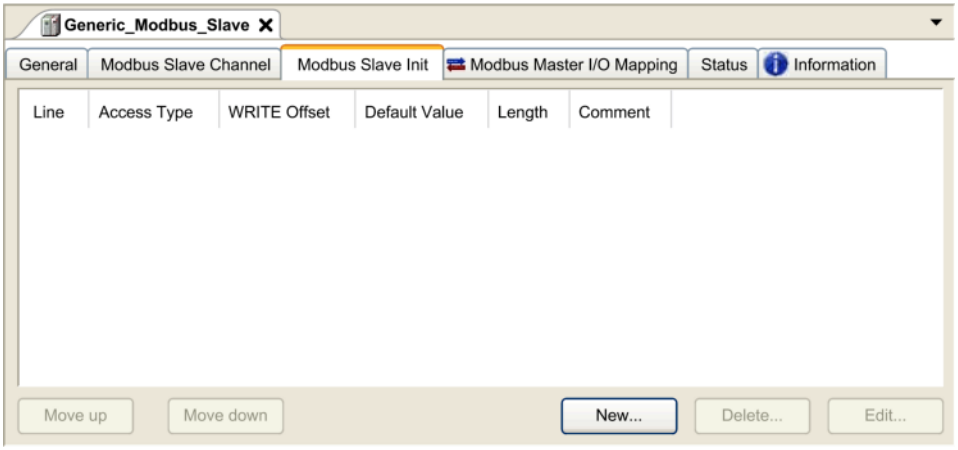
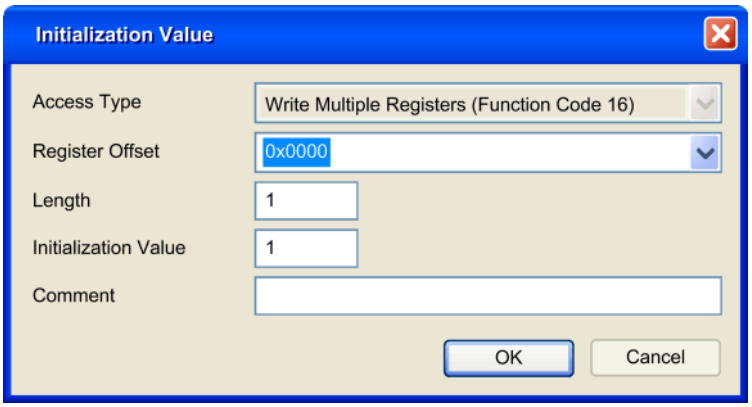
Step	Action
1	In the **Devices tree**, double-click **Generic Modbus Slave**. **Result:** The configuration window is displayed. 
2	Enter a **Slave Address** value for your device (choose a value from 1 to 247).
3	Choose a value for the **Response Timeout** (in ms).

To configure the **Modbus Channels**, proceed as follows:

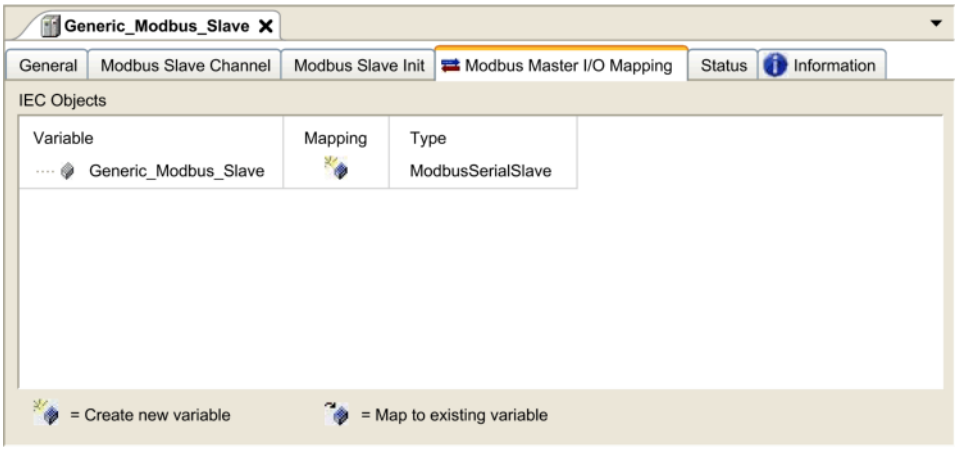
Step	Action
1	Click the **Modbus Slave Channel** tab: 
2	Click the **Add Channel** button: 

Step	Action
3	Configure an exchange: In the area **Channel**, you can add the following values: • **Name**: Enter a name for your channel. • **Access Type**: Choose the exchange type: Read or Write or Read/Write requests. See Access Types, page 210. • **Trigger**: Choose the trigger of the exchange. It can be **CYCLIC** with the period defined in **Cycle Time (ms)** field, started by a **RISING EDGE** on a boolean variable (this boolean variable is then created in the **Modbus Master I/O Mapping** tab), or by the **Application**. • **Comment**: Add a comment about this channel. In the area **READ Register** (if your channel is Read or Read/Write one), you can configure the `%MW` to be read on the Modbus slave. Those are mapped on `%IW` (see Modbus Master I/O Mapping tab): • **Offset**: Offset of the `%MW` to read. 0 means that the first object that is read is `%MW0`. • **Length**: Number of `%MW` to be read. For example, if 'Offset' = 2 and 'Length' = 3, the channel reads `%MW2, %MW3` and `%MW4`. • **Error Handling**: choose the behavior of the related `%IW` in case of loss of communication. In the area **WRITE Register** (if your channel is Write or Read/Write one), you can configure the `%MW` to be written to the Modbus slave. Those are mapped on `%QW` (see Modbus Master I/O Mapping tab): • **Offset**: Offset of the `%MW` to write. 0 means that the first object that is written is `%MW0`. • **Length**: Number of `%MW` to be written. For example, if 'Offset' = 2 and 'Length' = 3, the channel writes `%MW2, %MW3` and `%MW4`.
4	Click **OK** to validate the configuration of this channel. **NOTE:** You can also: • Click the **Delete** button to remove a channel. • Click the **Edit** button to change the parameters of a channel. **Result:** The configured channels are displayed: Library Manager | SR_Main | TM3BC_ModbusTCP | TM3BC_ModbusSL | **Generic_Modbus_Slave** | Modbus_IOScanner General | Modbus Slave Channel | Modbus Slave Init | Modbus Master I/O Mapping | Modbus Master IEC Objects | Status | Information | | Name | Access Type | Trigger | READ Offset | Length | Error Handling | WRITE Offset | Length | Comment | | 0 | Channel 0 | Read Holding Registers (Function Code 03) | Cyclic, t#100ms | 16#0000 | 1 | Keep last value | | | | | 1 | Channel 1 | Write Multiple Registers (Function Code 16) | Cyclic, t#100ms | | | | 16#0000 | 1 | |

To configure your **Modbus Initialization Value**, proceed as follows:

Step	Action
1	Click the **Modbus Slave Init** tab: 
2	Click **New** to create a new initialization value:  The **Initialization Value** window contains the following parameters: • **Access Type**: Enter the exchange type: Write requests , page 210. • **Register Offset**: Register number of register to be initialized. • **Length**: Number of `%MW` to be read. For example, if 'Offset' = 2 and 'Length' = 3, the channel reads `%MW2`, `%MW3` and `%MW4`. • **Initialization Value**: Value the registers are initialized with. • **Comment**: Add a comment about this channel.
3	Click **OK** to create a new **Initialization Value**. **NOTE:** You can also: • Click **Move up** or **Move down** to change the position of a value in the list. • Click **Delete** to remove a value in the list. • Click **Edit** to change the parameters of a value.

To configure your **Modbus Master I/O Mapping**, proceed as follows:

Step	Action
1	Click the **Modbus Master I/O Mapping** tab: 
2	Double-click in a cell of the **Variable** column to open a text field. Enter the name of a variable or click the browse button **[...]** and choose a variable with the **Input Assistant**. For more information on I/O mapping, refer to the EcoStruxure Machine Expert Programming Guide.

Access Types

This table describes the different access types available:

Function	Function Code	Availability
Read Coils	1	**ModbusChannel**
Read Discrete Inputs	2	**ModbusChannel**
Read Holding Registers (default setting for the channel configuration)	3	**ModbusChannel**
Read Input Registers	4	**ModbusChannel**
Write Single Coil	5	**ModbusChannel** **Initialization Value**
Write Single Register	6	**ModbusChannel** **Initialization Value**
Write Multiple Coils	15	**ModbusChannel** **Initialization Value**
Write Multiple Registers (default setting for the slave initialization)	16	**ModbusChannel** **Initialization Value**
Read/Write Multiple Registers	23	**ModbusChannel**

ControlChannel: Enables or Disables a Communication Channel

Function Description

This function allows you to enable or disable a communication channel.

A channel managed by this function is reinitialized to its default value after a reset (cold/warm).

After a stop or after a start, the channel remains disabled if it was disabled before.

On the contrary, after a reset, the channel is enabled even if it was disabled before.

In the case of the TM3BCSL Modbus Serial Line bus coupler, there are multiple, separate and independent communication channels.

⚠WARNING

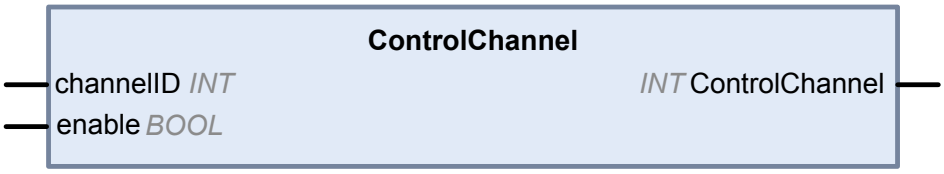
UNINTENDED EQUIPMENT OPERATION

Ensure that the Modbus serial line communication channels of the TM3BCSL bus coupler are set to the same state, either enabled or disabled.

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

NOTE: Use *ChannelID* value -1 to apply the *ControlChannel* on all channels configured on the TM3BCSL Modbus Serial Line bus coupler.

Graphical Representation



I/O Variable Description

This table describes the input variables:

Input	Type	Comment
ControlChannel	INT	Return 0 on success, a negative value on error.
ChannelID	INT	The channel number (visible in the first column of the configuration page). Or -1 to apply the command on all channels of this device.

This table describes the output variable:

Output	Type	Comment
Enable	BOOL	Enable or disable command.

Adding a Modem to a Manager

Introduction

A modem can be added to the following managers:

- ASCII Manager
- Modbus Manager
- Machine Expert Network Manager

NOTE: Use a modem which implements Hayes commands if you need a modem connection with Machine Expert Network Manager.

Adding a Modem to a Manager

To add a modem to your controller, select the modem you want in the **Hardware Catalog**, drag it to the **Devices tree**, and drop it on the manager node.

For more information on adding a device to your project, refer to:

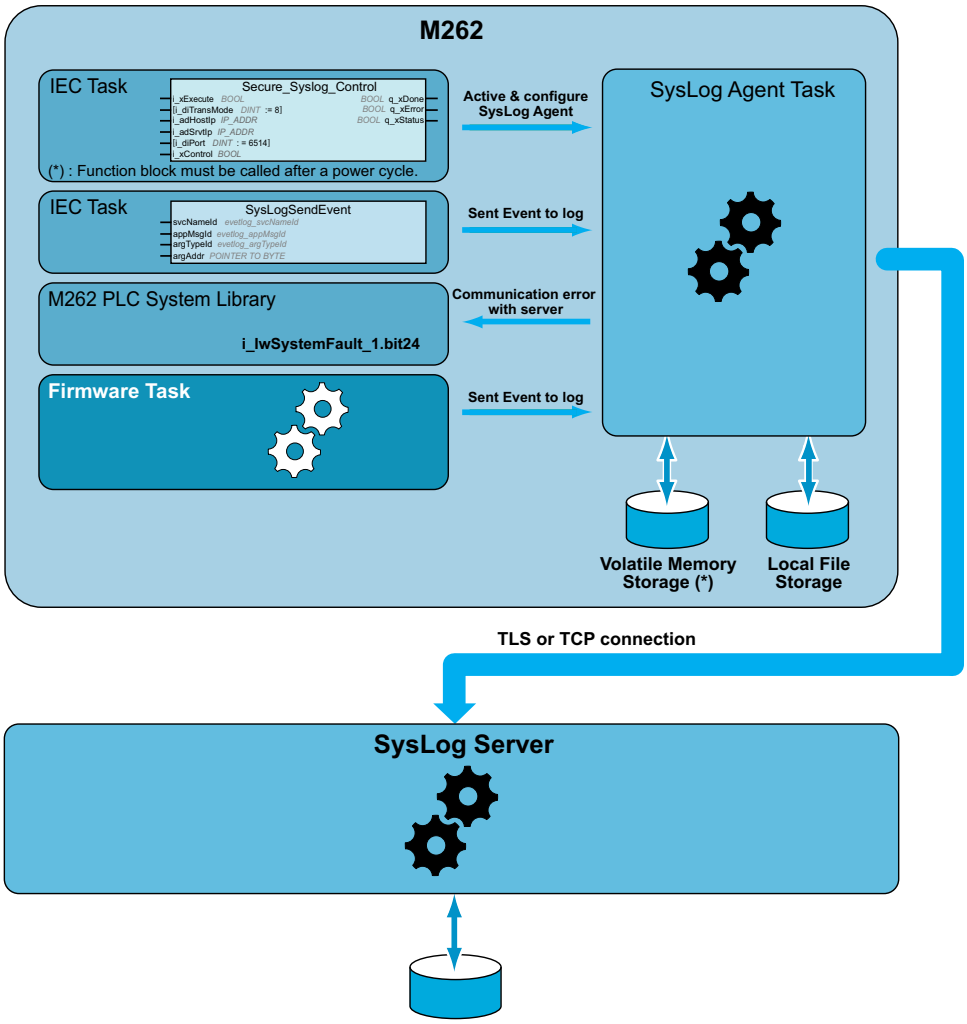
• Using the Hardware Catalog found in the EcoStruxure Machine Expert Programming Guide

• Using the Contextual Menu or Plus Button found in the EcoStruxure Machine Expert Programming Guide

For further information, refer to Modem Functions, Modem Library Guide.

SysLog Agent

System Overview



To activate a SysLog Agent, you have to instantiate a function block in your application. This function block starts and configures a SysLog Agent which operates even if you operate commands to download an application, stop, run or halt your controller.

NOTE: SysLog Agent must be started again after a power cycle.

You can send an Event to Log by using the *SysLogSendEvent* function. Refer to *SysLogSendEvent* function (see Syslog Library Guide).

SysLog Agent stores its configuration to communicate with the SysLog Server in the Volatile Memory Storage. SysLog Agent uses some directories as public key infrastructure (PKI) in Local File Storage to manage certificates (see Syslog Library Guide) of allowed server.

SysLog Agent stores historical information about Event to Log on SysLog Server. These files are useful to restore Events to Log during a period of disconnection. The Modicon M262 Logic/Motion Controller can store at least 2048 Events in these files.

The file access is restricted by the configuration of User Rights on your controller.

Diagnostic of SysLog Agent

A system bit (refer to Modicon M262 Logic/Motion Controller – System Library Guide) is set to 0 when an error is detected. This bit is identified as **PLC_GVL.PLC_R.i_lwSystemFault_1**.

TLS and Controller Compatibility

SysLog Agent is compatible with:

- TLS1.2 and TLS1.3
- TM262 firmware version 5.1.6.1 or later

OPC UA

Overview

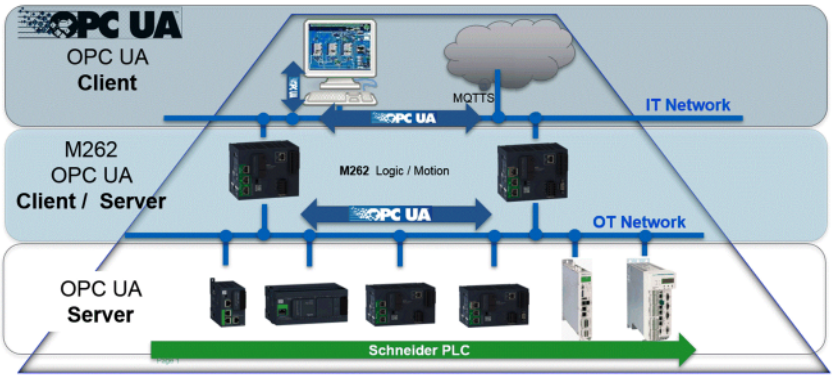
This chapter describes how to configure the OPC UA server and client services of the M262 Logic/Motion Controller.

OPC UA Overview

Introduction

OPC Unified Architecture (OPC UA) is a vendor-independent communication protocol for industrial automation applications.

The M262 Logic/Motion Controller embeds both client and server services:



OPC UA Server Configuration

OPC UA Server Overview

Overview

The OPC Unified Architecture server (OPC UA server) allows the M262 Logic/Motion Controller to exchange data with OPC UA clients. Server and client communicate through sessions.

The items of data (also referred to as symbols) to be shared by the OPC UA server are manually selected from a list of the IEC variables used in the application.

The OPC UA server supports Read and Write access as well as the subscription model. When using the subscription model, the OPC UA server reads the values of symbols from devices at a fixed sampling rate, places the data in a queue, then sends them to clients as notifications at a regular publishing interval. The sampling interval can be shorter than the publishing interval, in which case notifications may be queued until the publishing interval elapses.

Symbols that have not changed value since the previous sample are not re-published. Instead, the OPC UA server sends regular KeepAlive messages to indicate to the client that the connection is still active.

NOTE: The OPC UA server is restarted upon a download, online change or a reset.

User and Group Access Rights

Access to the OPC UA server is controlled by user rights. Refer to User Rights, page 86.

OPC UA Services

The following OPC Foundation Profiles and facets are implemented:

- OPC UA Micro Embedded Profile 2017
  - Core 2017 Server Facet
  - UA-TCP UA-SC UA-Binary
  - Embedded DataChange Subscription
- Security Category
  - SecurityPolicy – Basic256Sha256
  - SecurityPolicy – Basic256

The following functionalities are supported:

- Address Space Model
- Attribute Services
- Base Information
- Discovery Services
- Monitored item Services
- Protocol and Encoding (TCP UA Binary)
- Security
- Session Services
- Subscription Services
- View Services

OPC UA Server Configuration

Introduction

The OPC UA Server Configuration window allows you to configure the OPC UA server. OPC UA server is using encrypted communication by default with maximum security settings set by default.

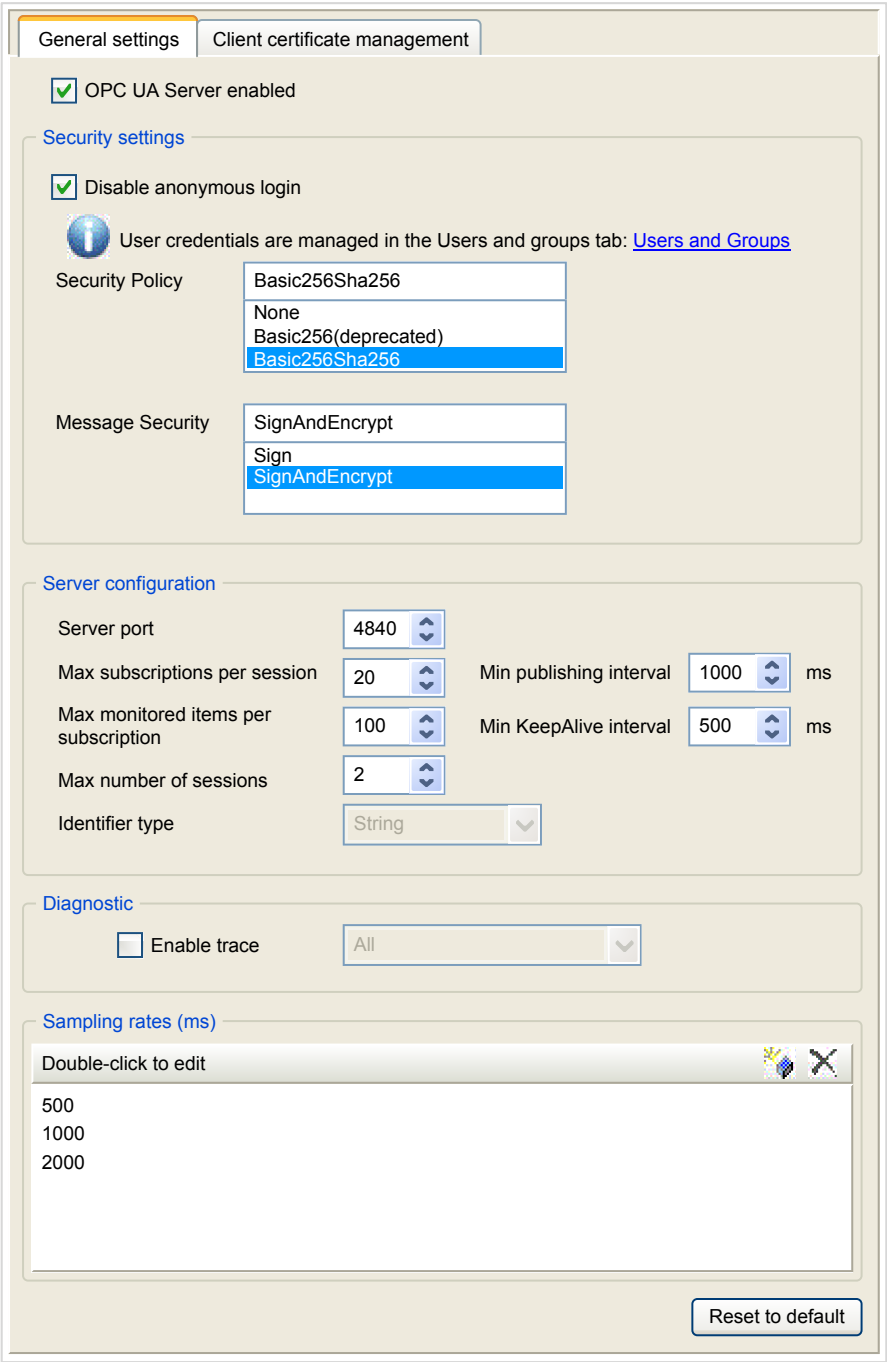
Accessing the OPC UA Server Configuration Tab

To configure the OPC UA Server:

Step	Action
1	In the **Devices tree**, double-click **MyController**.
2	Select the **OPC UA Server Configuration** tab.

OPC UA Server Configuration Tab

The following figure shows the OPC UA Server Configuration window:



OPC UA Server Configuration Description

This table describes the OPC UA Server Configuration parameters:

General Settings

Parameter	Value	Default value	Description
OPC UA Server Enabled	Enabled/ Disabled	Disabled	This checkbox is used to enable or disable the OPC UA Server and Client on the controller.

Security Settings

Parameter	Value	Default value	Description
Disable anonymous login	Enabled/ Disabled	Enabled	Uncheck this checkbox to allow anonymous login on OPC UA server.
Security Policy	None Basic256 (deprecated) (1) Basic256Sha256	Basic256-Sha256	This drop-down menu allows you to secure your exchanges by signing and encrypting the data you send and receive.
Message Security	None Sign SignAndEncrypt	SignAndEncrypt	The messages are related to the **Security Policy** selected.
(1) Security policies marked as deprecated are policies which no longer afford an acceptable level of security.			

Server Configuration

Parameter	Value	Default value	Description
Server port	1...65535	4840	The port number of the OPC UA server. OPC UA clients must append this port number to the TCP URL of the controller to connect to the OPC UA server.
Max. subscriptions per session	1...100 (2)	20	Specify the maximum number of subscriptions allowed within each session.
Min. publishing interval	200...5000	1000	The publishing interval defines how frequently the OPC UA server sends notification packages to clients. Specify the minimum time that must elapse between notifications, in ms.
Max. monitored items per subscription	1...10000 (2)	100	The maximum number of *monitored items* in each subscription that the server assembles into a notification package.
Min. KeepAlive interval	500...5000	500	The OPC UA server only sends notifications when the values of monitored items of data are modified. A *KeepAlive* notification is an empty notification sent by the server to inform the client that although no data has been modified, the subscription is still active. Specify the minimum interval between KeepAlive notifications, in ms.
Max. number of sessions	1...4	2	The maximum number of clients that can connect simultaneously to the OPC UA server.
Identifier type	**String**	**String**	Certain OPC UA clients require a specific format of unique symbol identifier (node ID).
(2) The total count (**Max. subscriptions per session** x **Max. monitored items per subscription**) cannot exceed 10000.			

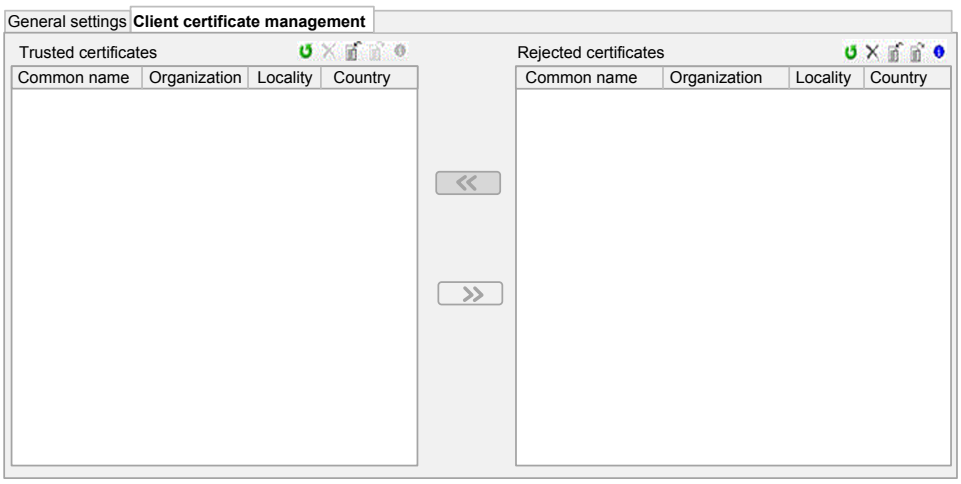
Diagnostic

Parameter	Value	Default value	Description
Enable trace	Enabled/disabled	Enabled	Select this checkbox to include OPC UA diagnostic messages in the controller log file (see the EcoStruxure Machine Expert Programming Guide). Traces are available from the **Log** tab or from the System Log File, page 147 of the Web server. You can select the category of events to write to the log file: • **None** • **Error** • **Warning** • **System** • **Info** • **Debug** • **Content** • **All** (default)
Sampling rates (ms)	**200...5000**	**500** **1000** **2000**	The sampling rate indicates a time interval, in milliseconds (ms). When this interval has elapsed, the server sends the notification package to the client. The sampling rate can be shorter than the publishing interval, in which case notifications are queued until the publishing interval has elapsed. Sampling rates must be in the range 200...5000 (ms). Up to 3 different sampling rates can be configured. Double-click on a sampling rate to edit its value. To add a sampling rate to the list, right-click and choose **Add a new rate**. To remove a sample rate from the list, select the value and click ✕.

Click **Reset to default** to return the configuration parameters on this window to their default values.

Client Certificate Management Tab

This tab allows you to determine which OPC UA client certificates are trusted by the M262 Logic/Motion Controller OPC UA server.



Client Certificate Management Tab, Toolbar

Element	Description
	Both certificate lists are loaded or refreshed.
	Deletes the selected certificates.
	Opens a Windows dialog box (Open) to import a certificate that is uploaded to the selected certificate list (trusted certificates list or rejected certificates list).
	Opens a Windows dialog box (Save as) to export the selected certificates to a selectable path.
	Opens a dialog box containing additional information on the selected certificate. **Certificate properties** Common name: TestCertificate Organization: Schneider Locality: Marktheidenfeld Country: DE Format: X509 Effective date: 12/11/2013 1:38:34 PM Expiration date: 12/10/2018 1:38:34 PM Public key: 3082010A0282010100C2B6EA390... Certificate data: 3082053130820419A00302010202... Serial number: 52A85CCA TestCertificate Close

Trusted Certificates List and Rejected Certificates List

A certificate contains common information about the company that owns the certificate, how long a certificate is valid, and so on. The certificate management provides two list views:

- trusted certificates
- rejected certificates.

Element	Description
Trusted certificates	This list includes the client certificates the server trusts.
Rejected certificated	This list includes the client certificates the server does not trust.
<< >>	Use the << and >> buttons to move a rejected certificate to the Trusted certificates list or the opposite way. During the moving procedure, a progress bar appears and displays the remaining files.

NOTE: OPC UA client and server share the same default PKI folder structure, including the trusted and untrusted (rejected) folders, this means trusting or untrusting (rejecting) a certificate has the same effect for both client and server.

NOTE: The OPC UA self-signed certificate has a limitation when the network interface through which OPC UA communicates is using dynamic IP addresses (DHCP). If you configured DHCP in such interface, make sure your OPC UA peer accepts the M262 Logic/Motion Controller OPC UA self-signed certificate without validation.

OPC UA Certificates Management Actions

This table describes each action concerning OPC UA certificates management and how to achieve it.

Action / Task	EcoStruxure Machine Expert Security Screen (1)	EcoStruxure Machine Expert M262 Files Screen (2)	EcoStruxure Machine Expert M262 OPC UA Server Screen (3)	M262 Webpage Maintenance - Certificates	FTP Protocol (2)
Access to M262 OPC UA PKI folders	YES	YES	YES	NO	YES
Import a certificate	YES	YES	YES	NO	YES
Export a Certificate	YES	YES	YES	NO	YES
Remove a certificate	YES	YES	YES	NO	YES
Trust / Untrust a certificate	NO	YES (4)	YES	YES (5)	YES (4)
Check a certificate information	YES	NO	YES	NO	NO

PKI: Public Key Infrastructure.

(1) Only for M262 own certificates folder.

(2) Except M262 own certificates folder.

(3) Only trusted and untrusted (rejected) certificate folders.

(4) Requires to manually move the certificate from the trusted folder to the untrusted (rejected) folder (and vice-versa).

(5) Requires Administrator access.

OPC UA PKI Folder List and Usage

The table describes the Public Key Infrastructure (PKI) shared between the M262 Logic/Motion Controller OPC UA server and OPC UA client. It provides the folder list and their usage.

M262 File System Folders	Description
/usr/pki	Root folder of the default PKI
/usr/pki/issuer/certs	Contains Certificate Authority (CA) certificates that are required to validate Certification Paths
/usr/pki/issuer/crl	Contains Certificate Revocation Lists (CRL) for CA certificates
/usr/pki/trusted/certs	Contains Trusted certificates
/usr/pki/trusted/crl	Contains Certificate Revocation Lists (CRL) for the Trusted certificates
/usr/pki/untrusted	Contains Untrusted certificates
/usr/pki/quarantine	Not used for M262 OPC UA (legacy for other services)

NOTE: Some of the PKI folders are only available after downloading the application enabling OPC UA (server/client), since some folders are only created in the runtime initialization of OPC UA.

OPC UA Server Symbols Configuration

Introduction

Symbols are the items of data shared with OPC UA clients. Symbols are selected from a list of all the IEC variables used in the application. The selected symbols are then sent to the controller as part of the application download.

Each symbol is assigned a unique identifier. Identifiers are in string format.

This table describes IEC variable Base Types versus OPC UA Data Types:

IEC variable Base Types	OPC UA Data Types
BOOL, BIT	Boolean
BYTE, USINT	Byte
INT	Int16
WORD, UINT	Uint16
DINT, TOD, TIME	Int32
DWORD, UDINT	Uint32
LINT, LTIME	Int64
LWORD, ULINT	Uint64
REAL	Float
LREAL	Double
WSTRING, STRING	Up to 255 characters - String
DATE, DT	Second precision - DateTime
SINT	SByte

Bit memory variables (%MX) cannot be selected. In addition to IEC base data types, the OPC UA server can also expose OPC UA variables from IEC symbols that are composed of the following complex types:

- Arrays and Multi-Dimensional Arrays. These are limited to 3 dimensions.
- Structured data types, and nested structured data types. As long as they are not composed of a UNION field.

  NOTE: For a STRING / WSTRING variable within Structured data types, if the number of characters exceeds specified length, the string is truncated.

Displaying the List of Variables

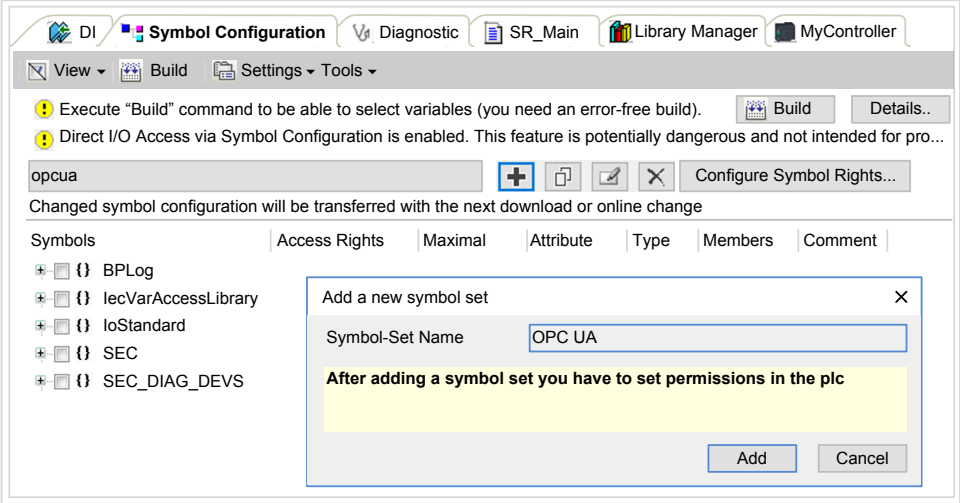
To display the list of variables:

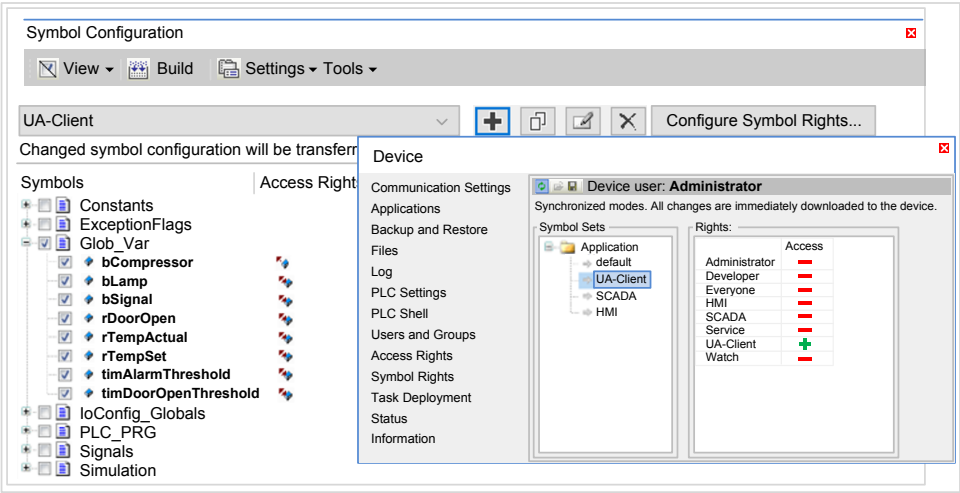
Step	Action
1	On the **Applications tree** tab, right-click **Application** and choose **Add object > Symbol Configuration**. **Result:** The **Add Symbol Configuration** window is displayed. The controller starts the OPC UA server.
2	Click **Add**.

NOTE: The IEC objects `%MX`, `%IX`, `%QX` are not directly accessible. To access IEC objects you must first group their contents in located registers (refer to Relocation Table, page 45).

Selecting OPC UA Server Symbols

The **Symbol Configuration** window displays the variables available for selection as symbols:





Select **IoConfig_Globals_Mapping** to select all the available variables. Otherwise, select individual symbols to share with OPC UA clients.

Each symbol has the following properties:

Name	Description
Symbols	The variable name followed by the address of the variable.
Type	The data type of the variable.
Access type	Click repeatedly to specify the access rights of the symbol: • read-only ( ) (default), • write-only ( ), • or read/write ( ). **NOTE:** Click in the **Access type** column of **IoConfig_Globals_Mapping** to set the access rights of all the symbols at once.
Comment	An optional comment.

Click **Refresh** to update the list of available variables.

OPC UA Server Performance

Overview

As an example, the following provides capacity and performance information for the OPC UA server of the M262 Logic Controller. Design considerations are also provided to help you consider the optimal conditions for the performance of the OPC UA server. Of course, the performance realized by your application depend on many variables and conditions, and may differ from this example.

NOTE: These values are only valid if the OPC UA client function is not used.

System Configurations Used to Evaluate Performance

OPC UA server performance is determined by the system configuration, the number of symbols being published, and the percentage of symbols being refreshed.

The following table presents the number of elements small and medium configurations used for evaluating OPC UA server performance:

Elements	Small	Medium
TM3 expansion modules	0	4
CANopen slave devices	0	0
PTO functions	0	0
HSC functions	0	0
Profibus connections	0	0
Modbus TCP slave devices	0	0
Sercos devices	0	0
Incremental hardware encoders	0	0
Serial lines	0	1
EtherNet/IP adapters	0	1
EtherNet/IP Scanner devices	0	18
Generic TCP/UDP Managers	0	0

This table presents average read/write request times for the sample configurations and for different numbers of symbols:

Average Read/Write Request Times							
Configuration		Number of Symbols					
		50	100	250	400	500	1000
TM262L10MESE8T	Small	6 ms	11 ms	26 ms	41 ms	53 ms	132 ms
TM262L10MESE8T	Medium	16 ms	29 ms	71 ms	117 ms	149 ms	350 ms
TM262L20MESE8T	Small	3 ms	5 ms	12 ms	18 ms	23 ms	56 ms
TM262L20MESE8T	Medium	14 ms	23 ms	51 ms	80 ms	103 ms	123 ms

The following tables present the average time required to refresh a monitored set of symbols using a sampling rate of 200 ms and a publishing interval of 200 ms.

This table presents the average time required to refresh 100% of symbols for each of the sample configurations:

Average Time to Refresh 100% of Symbols				
Configuration		**Number of Symbols**		
		100	**400**	**1000**
TM262L10MESE8T	Small	204 ms	207 ms	218 ms
TM262L10MESE8T	Medium	197 ms	209 ms	680 ms
TM262L20MESE8T	Small	201 ms	203 ms	201 ms
TM262L20MESE8T	Medium	202 ms	205 ms	215 ms

This table presents the average time required to refresh 50% of symbols for each of the sample configurations:

Average Time to Refresh 50% of Symbols				
Configuration		**Number of Symbols**		
		100	**400**	**1000**
TM262L10MESE8T	Small	203 ms	204 ms	208 ms
TM262L10MESE8T	Medium	195 ms	201 ms	623 ms
TM262L20MESE8T	Small	201 ms	202 ms	204 ms
TM262L20MESE8T	Medium	202 ms	203 ms	207 ms

This table presents the average time required to refresh 1% of symbols for each of the sample configurations:

Average Time to Refresh 1% of Symbols				
Configuration		**Number of Symbols**		
		100	**400**	**1000**
TM262L10MESE8T	Small	201 ms	202 ms	202 ms
TM262L10MESE8T	Medium	194 ms	196 ms	285 ms
TM262L20MESE8T	Small	200 ms	201 ms	201 ms
TM262L20MESE8T	Medium	201 ms	202 ms	202 ms

Optimizing OPC UA Server Performance

The OPC UA server functionality is dependent on external communication networks, external device performance, and other external parameters. Data transmitted may be delayed or other possible communication errors may arise that impose practical limits on machine control. Do not use the OPC UA server functionality for safety-related data or other time-dependent purposes.

⚠ WARNING

UNINTENDED EQUIPMENT OPERATION

- Do not allow safety-related data in OPC UA server data exchanges.
- Do not use OPC UA server data exchanges for any critical or time-dependent purposes.
- Do not use OPC UA server data exchanges to change equipment states without having done a risk analysis and implementing appropriate safety-related measures.

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

The above tables can be useful in determining whether OPC UA server performance is within acceptable limits. Be aware, however, that other external factors influence overall system performance, such as the volume of Ethernet traffic.

To optimize OPC UA server performance, consider the following:

- Minimize Ethernet traffic by setting the **Min. publishing interval** to the lowest value that yields an acceptable response time.
- The task cycle time, page 49 configured for the controller must be less than the configured **Min. publishing interval** value.
- Configuring a **Max. number of sessions** (the number of OPC UA clients that can simultaneously connect to the OPC UA server) value of greater than 1 decreases the performance of all sessions.
- The sampling rate determines the frequency at which data is exchanged. Tune the **Sampling rates (ms)** value to product the lowest response time that does not adversely affect the overall performance of the controller.

OPC UA Client Configuration

OPC UA Client Overview

Introduction

The OPC Unified Architecture client (OPC UA client) allows the M262 Logic/Motion Controller to exchange data with remote OPC UA servers.

NOTE: At least one Ethernet interface (**Ethernet_1** or **Ethernet_2**) must be available on the controller and Ethernet traffic not blocked by the firewall, page 153.

The OPC UA client can connect with up to 5 servers concurrently. Each server can exchange 5000 items, with a maximum of 15000 items for all the servers.

Scope of the Security Model

The OPC UA client provides:

- Support of binary message encoding
- Support of message encryption and integrity:
  - **None**, **Basic256** and **Basic256Sha256** security policies.
  - **None**, **Sign** and **Sign and Encrypt** message security modes.
- Support of user authentication:
  - **None** (**Anonymous**) or user name and password.
- Support of user authorization:
  - To read and write value of a node.
  - To browse the information model based on the access rights of the information model, the user or the user's role.

User and Group Access Rights

Access to OPC UA connections and data is controlled by user rights. Refer to User Rights, page 86.

OPC UA Services

The following OPC Foundation Profiles and facets are implemented:

- OPC UA Minimum Client Profile
  - SecurityPolicy – None
  - User Token – Anonymous Facet
  - UA-TCP UA-SC UA-Binary
  - User Token – User Name Password
- Security Category
  - SecurityPolicy – **Basic256Sha256**
  - SecurityPolicy – **Basic256**
- Data Access Facet
  - Attribute Read Client Facet
  - Attribute Write Client Facet
  - DataChange Subscriber Client Facet

The following functionalities are supported:

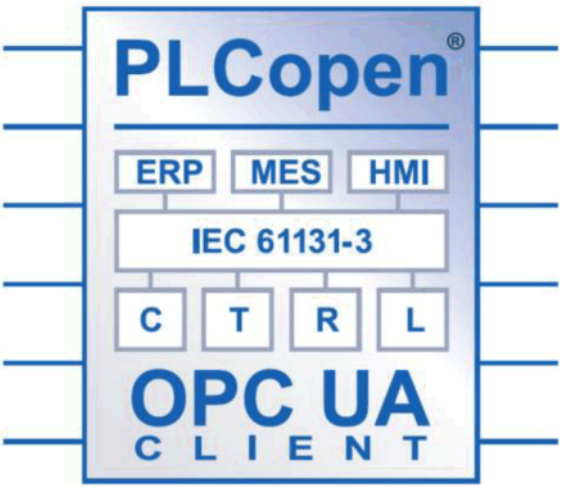
- Attribute Services (for Value attribute only)
- Discovery Services (Configure Endpoint only)
- Monitored Item Services
- Protocol and Encoding (TCP UA Binary)
- Security
- Session Services
- Subscription services
- View Services

Programming the OPC UA Client

Overview

OPC UA client functionality is delivered in the *OpcUaHandling* library.

This library contains IEC 61131-3 standard function blocks to include in your application:



The function blocks allow you to:

- Read/write multiple data items
- Perform diagnostics

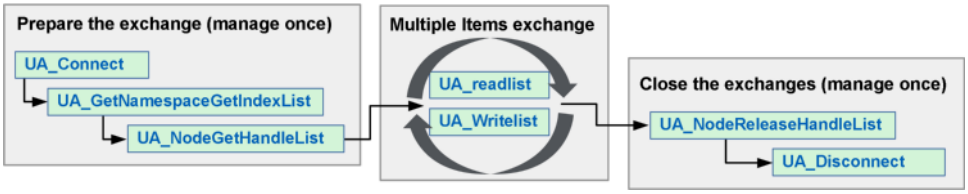
The following function blocks are supported:

- UA_Connect
- UA_ConnectionGetStatus
- UA_Disconnect
- UA_NamespaceGetIndexList
- UA_NodeGetHandleList
- UA_NodeGetInformation
- UA_NodeReleaseHandleList
- UA_ReadList
- UA_WriteList
- UA_Browse
- UA_SubscriptionCreate
- UA_SubscriptionDelete
- UA_SubscriptionProceed
- UA_MonitoredItemAddList
- UA_MonitoredItemOperateList
- UA_MonitoredItemRemoveList
- UA_TranslatePathList
- FB_TimeStamper

For details, refer to the OpcUaHandling Library Guide.

Example: Managing a Read/Write List

This figure shows the function blocks used to read and write items of data managed by a remote OPC UA server:



NOTE: Ensure that **OPC UA Server enabled** is activated to use OPC UA Client. See OPC UA Server Configuration Tab, page 217.

Post Configuration

Introduction

This chapter describes how to generate and configure the post configuration file of the Modicon M262 Logic/Motion Controller.

Post Configuration Presentation

Introduction

Post configuration is an option that allows you to modify some parameters of the application without changing the application. Post configuration parameters are defined in a file called **Machine.cfg**, which is stored in the controller.

By default, all parameters are set in the application. The parameters defined in the Post Configuration file are used instead of the corresponding parameters defined in the application.

Parameters

The Post Configuration file allows you to change network parameters.

Ethernet parameters:

- IP Address
- Subnet Mask
- Gateway Address
- IP Config Mode
- Device Name

Serial Line parameters, for each serial line in the application (embedded port or PCI module):

- Baud rate
- Parity
- Data bits
- Stop bit

FTP Server Encryption:

- FTP Server Encryption setting parameter

WebVisualisation:

- WebVisualisation connection type (encryption enforcing parameter)

CAE Enable:

- Cybersecurity Admin Expert parameter

ODVA Enable:

- ODVA parameter

OPC UA Server CRL check:

- CRL check deactivation

OPC UA Server Uri:

- Customized controller OPC UA Server Application Uri name

  NOTE: "urn:Schneider:M262:Application" will be replaced by "urn:Schneider:M262:{$param}" in case a custom namespace is provided in Machine.cfg.

NOTE: Default Server Uri is applied if the configured value is not valid for the controller. Refer to the message logger logs for troubleshooting.

Parameter updates with a Post Configuration file that impacts parameters used by other devices through a communication port are not updated in the other devices.

For example, if the IP address used by an HMI is updated in the configuration with a Post Configuration file, the HMI uses the previous address. You must update the address used by the HMI independently.

If the OPC UA Server Uri is replaced, the certificate is re-generated and needs to be trusted again by the client devices.

Operating Mode

The Post Configuration file is read after:

- A Reset Warm command, page 67
- A Reset Cold command, page 67
- A reboot, page 70
- An application download, page 72

NOTE: The M262 Logic/Motion Controller reads a maximum of 50 entries from the Post Configuration file. If the file contains more than 50 entries, only the first 50 entries are saved. An error message is displayed in the message logger: `Cannot store all information from lateconf`.

Refer to Controller States and Behaviors, page 56 for further details on controller states and transitions.

NOTE: The post configuration is ignored for applications where a scanner is configured.

Post Configuration File Management

Introduction

The file **Machine.cfg** is located in the directory *`/usr/cfg`*.

Each parameter is specified by a variable type, variable ID, and value. The format is:

```
id[moduleType].pos[param1Id].id[param2Id].param[param3Id].
paramField=value
```

Each parameter is defined on three lines in the Post Configuration file:

- The first line describes the internal 'path' for this parameter.
- The second line is a comment describing the parameter.
- The third line is the definition of the parameter (as described above) with its value.

Parameters

The `pos[param1Id]` parameter (see Introduction, page 230) depends on the controller reference:

TM262L•

- Ethernet 1: `pos[4]`
- Ethernet 2: `pos[5]`
- Serial line: `pos[6]`

TM262M•

- Ethernet 1: `pos[5]`
- Ethernet 2: `pos[6]`
- Serial line: `pos[7]`

Post Configuration File Generation

The Post Configuration file (**Machine.cfg**) is generated by EcoStruxure Machine Expert.

To generate the file, proceed as follows:

Step	Action
1	In the menu bar, choose **Build** > **Post Configuration** > **Generate...** **Result:** An explorer window is displayed.
2	Select the destination folder of the Post Configuration file.
3	Click **OK**.

When you use EcoStruxure Machine Expert to create a Post Configuration file (**Generate**), it reads the value of each parameter assigned in your application program and then writes the values to the **Machine.cfg** Post Configuration file.

After generating a Post Configuration file, review the file and remove any parameter assignments that you wish to remain under the control of your application. Keep only those parameter assignments that you wish changed by the Post Configuration function that are necessary to make your application portable and then modify those values appropriately.

Post Configuration File Transfer

After creating and modifying your Post Configuration file, transfer it to the `/usr/cfg` directory of the controller. The controller does not read the **Machine.cfg** file unless it is in this directory.

You can transfer the Post Configuration file by the following methods:

- SD card, page 250 (with the proper script)
- Download through the FTP server, page 129
- Download with EcoStruxure Machine Expert controller device editor, page 75

Modifying a Post Configuration File

If the Post Configuration file is located in the PC, use a text editor to modify it.

NOTE: Do not change the text file encoding. The default encoding is ANSI.

To modify the Post Configuration file directly in the controller, use the **Setup** menu of the Web server, page 132.

To modify the Post Configuration file in the controller with EcoStruxure Machine Expert in online mode:

Step	Action
1	In the **Devices tree**, click the controller name.
2	Click **Build** > **Post Configuration** > **Edit...** **Result:** The Post Configuration file opens in a text editor.
3	Edit the file.
4	If you want to apply the modifications after saving them, select **Reset device after sending**.
5	Click **Save as**.
6	Click **Close**.

NOTE: If the parameters are invalid, they are ignored.

Deleting the Post Configuration File

You can delete the Post Configuration file by the following methods:

- SD card (with the delete script)
- Through the FTP server, page 129
- Online with EcoStruxure Machine Expert controller device editor, page 75, **Files** tab

For more information on **Files** tab of the Device Editor, refer to the EcoStruxure Machine Expert Programming Guide.

NOTE: The parameters defined in the application are used instead of the corresponding parameters defined in the Post Configuration file after:

- A Reset Warm command, page 67
- A Reset Cold command, page 67
- A reboot, page 70
- An application download, page 72

Post Configuration Example

Post Configuration File Example for TM262M•

```
# [PLC_REF] / RNDIS USB address

# RNDIS USB address

.param[1104] = [192, 168, 200, 1]


# [PLC_REF] / RNDIS USB mask

# RNDIS USB mask
.param[1105] = [255, 255, 255, 0]


# [PLC_REF] / FTP Server Encryption

# 1=encryption enforced, 0=otherwise

.param[1106] = 1
```

```
# [PLC_REF] / WebVisu Connection Type
# 0=Only HTTP connections are supported, 1 = Only HTTPS connections are supported, 2 = HTTP and HTTPS connections are supported, 3 = HTTP connections are redirected to HTTPS
.param[1107] = 3

# [PLC_REF] / CAE Enable
# 1=Enabled, 0=Disabled
.param[1108] = 1

# [PLC_REF] / Advanced ODVA features Enable
# 1=Enabled, 0=Disabled
.param[1109] = 0

# [PLC_REF] / Ethernet_1 / IPAddress
# Ethernet IP address
id[45000].pos[5].id[111].param[0] = [192, 168, 1, 3]]

# [PLC_REF] / Ethernet_1 / SubnetMask
# Ethernet IP mask
id[45000].pos[5].id[111].param[1] = [255, 255, 255, 0]]

# [PLC_REF] / Ethernet_1 / GatewayAddress
# Ethernet IP gateway address
id[45000].pos[5].id[111].param[2] = [0, 0, 0, 0]]

# [PLC_REF] / Ethernet_1 / IPConfigMode
# IP configuration mode: 0:FIXED 1:BOOTP 2:DHCP
id[45000].pos[5].id[111].param[4] = 0

# [PLC_REF] / Ethernet_1 / DeviceName
# Name of the device on the Ethernet network
id[45000].pos[5].id[111].param[5] = 'my_Device'

# [PLC_REF] / Ethernet_2 / IPAddress
# Ethernet IP address
id[45000].pos[6].id[45111].param[0] = [192, 168, 102, 2]

# [PLC_REF] / Ethernet_2 / SubnetMask
# Ethernet IP mask
```

```
id[45000].pos[6].id[45111].param[1] = [255, 255, 255, 0]

# [PLC_REF] / Ethernet_2 / GatewayAddress
# Ethernet IP gateway address
id[45000].pos[6].id[45111].param[2] = [0, 0, 0, 0]

# [PLC_REF] / Ethernet_2 / IPConfigMode
# IP configuration mode: 0:FIXED 1:BOOTP 2:DHCP
id[45000].pos[6].id[45111].param[4] = 0

# [PLC_REF] / Ethernet_2 / DeviceName
# Name of the device on the Ethernet network
id[45000].pos[6].id[45111].param[5] = 'my_Device'

# [PLC_REF] / Serial_Line / Serial Line Configuration / Baudrate
# Serial Line Baud Rate in bit/s
id[45000].pos[7].id[40101].param[10000].Bauds = 19200

# [PLC_REF] / Serial_Line / Serial Line Configuration / Parity
# Serial Line Parity (0=None, 1=Odd, 2=Even)
id[45000].pos[7].id[40101].param[10000].Parity = 2

# [PLC_REF] / Serial_Line / Serial Line Configuration / DataBits
# Serial Line Data bits (7 or 8)
id[45000].pos[7].id[40101].param[10000].DataFormat = 8

# [PLC_REF] / Serial_Line / Serial Line Configuration / StopBits
# Serial Line Stop bits (1 or 2)
id[45000].pos[7].id[40101].param[10000].StopBit = 1

# [PLC_REF] / Modbus Server / Maximum number of connections
# Maximum number of connections value (between 8 and 24)
.param[1012] = 10

# [PLC_REF] / OPCUA ServerUri
# Customize OPCUA ServerUri, only ASCII letters, digits, '-' and
'_', 29 char max. Default value is applied if empty or invalid
.param[1204] = ''

# [PLC_REF] / OPCUA server CRL check
```

```
# 1=CRL check disabled, 0=CRL check enabled
.param[1205] = 0
```

Connecting a Modicon M262 Logic/Motion Controller to a PC

Introduction

This chapter shows how to connect a Modicon M262 Logic/Motion Controller to a PC.

Connecting the Controller to a PC

Overview

To transfer, run, and monitor the applications, you can use either a USB cable or an Ethernet connection to connect the controller to a computer with EcoStruxure Machine Expert installed.

NOTICE

INOPERABLE EQUIPMENT

Always connect the communication cable to the PC before connecting it to the controller.

Failure to follow these instructions can result in equipment damage.

USB Mini-B Port Connection

Cable Reference	Details
BMXXCAUSBH018	Grounded and shielded, this USB cable is suitable for long duration connections.
TCSXCNAMUM3P	This USB cable is suitable for short duration connections such as quick updates or retrieving data values.

NOTE: You can only connect 1 controller or any other device associated with EcoStruxure Machine Expert and its component to the PC at any one time.

The USB Mini-B Port is the programming port you can use to connect a PC with a USB host port using EcoStruxure Machine Expert software. Using a typical USB cable, this connection is suitable for quick updates of the program or short duration connections to perform maintenance and inspect data values. It is not suitable for long-term connections such as commissioning or monitoring without the use of specially adapted cables to help minimize electromagnetic interference.

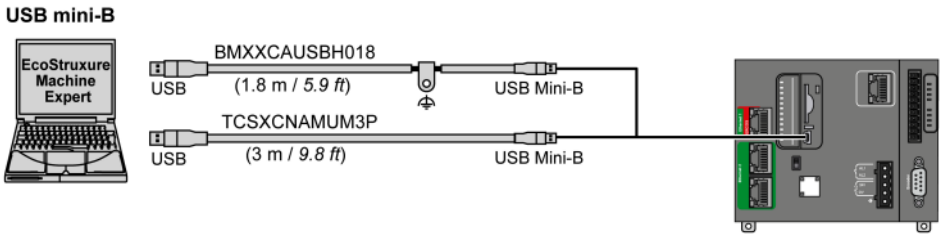
⚠ WARNING

UNINTENDED EQUIPMENT OPERATION OR INOPERABLE EQUIPMENT

- You must use a shielded USB cable such as a BMX XCAUSBH0•• secured to the functional ground (FE) of the system for any long-term connection.
- Do not connect more than one controller or bus coupler at a time using USB connections.
- Do not use the USB port(s), if so equipped, unless the location is known to be non-hazardous.

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

The communication cable should be connected to the PC first to minimize the possibility of electrostatic discharge affecting the controller.

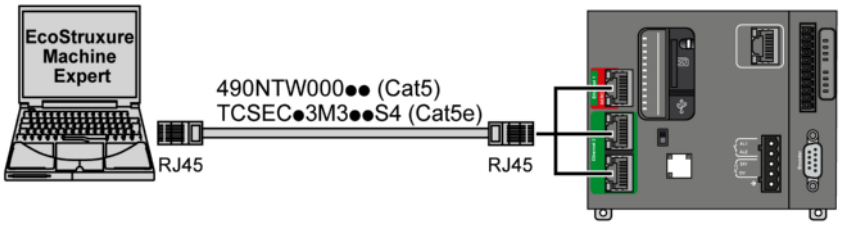


To connect the USB cable to your controller, follow the steps below:

Step	Action
1	**1a**. If making a long-term connection using the cable BMXXCAUSBH018, or other cable with a ground shield connection, be sure to securely connect the shield connector to the functional ground (FE) or protective ground (PE) of your system before connecting the cable to your controller and your PC. **1b**. If making a short-term connection using the cable TCSXCNAMUM3P or other non-grounded USB cable, proceed to step 2.
2	Connect your USB cable to the computer.
3	Open the protective cover for the USB mini-B slot on the controller.
4	Connect the mini-B connector of your USB cable to the controller.

Ethernet Port Connection

You can also connect the controller to a PC using an Ethernet cable.



To connect the controller to the PC, do the following:

Step	Action
1	Connect the Ethernet cable to the PC.
2	Connect the Ethernet cable to any of the Ethernet ports on the controller.

Updating Firmware

Introduction

Updating the controller firmware is possible by using:

- An SD card with a compatible script file.
- The Controller Assistant.

Updating the TM3 and the TMS firmware is possible by using an SD card with a compatible script file.

Performing a firmware update deletes the application program in the device, including the configuration files, the user management, the user rights, optionally the certificates (depending on the update method) and the Boot Application in non-volatile memory.

For more information about the firmware update and creating a new flash disk with firmware, refer to Project Settings - Firmware Update and Non-Volatile Memory Organization, page 40.

Updating the Controller Firmware by SD Card

Before Updating Firmware

The Modicon M262 Logic/Motion Controller accepts only SD cards formatted in FAT or FAT32.

The SD card must have a label. To add a label:

1. Insert the SD card in your PC.
2. Right-click on the drive in Windows Explorer.
3. Choose **Properties**.

⚠ WARNING

UNINTENDED EQUIPMENT OPERATION

- You must have operational knowledge of your machine or process before connecting this device to your controller.
- Ensure that guards are in place so that any potential unintended equipment operation will not cause injury to personnel or damage to equipment.

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

When an SD card is inserted into the SD card slot of the controller, the firmware searches and executes the script contained in the SD card (*/sys/cmd/Script.cmd*).

Performing a firmware update deletes the application program in the device, including the configuration files, the user management, the user rights, the certificates and the Boot Application in non-volatile memory.

NOTICE

LOSS OF APPLICATION DATA

- Perform a backup of the application program to the hard disk of the PC before attempting a firmware update.
- Restore the application program to the device after a successful firmware update.

Failure to follow these instructions can result in equipment damage.

If you remove power to the device, or there is a power outage or communication interruption during the transfer of the application, your device may become inoperative. If a communication interruption or a power outage occurs, reattempt the transfer. If there is a power outage or communication interruption during a firmware update, or if an invalid firmware is used, your device will become inoperative. In this case, use a valid firmware and reattempt the firmware update.

NOTICE

INOPERABLE EQUIPMENT

- Do not interrupt the transfer of the application program or a firmware change once the transfer has begun.
- Re-initiate the transfer if the transfer is interrupted for any reason.
- Do not attempt to place the device into service until the file transfer has completed successfully.

Failure to follow these instructions can result in equipment damage.

The serial line ports of your controller are configured for the CoDeSys protocol by default when new or when you update the controller firmware. The CoDeSys protocol is incompatible with that of other protocols such as Modbus Serial Line. Connecting a new controller to, or updating the firmware of a controller connected to, an active Modbus configured serial line can cause the other devices on the serial line to stop communicating. Make sure that the controller is not connected to an active Modbus serial line network before first downloading a valid application having the concerned port or ports properly configured for the intended protocol.

NOTICE

INTERRUPTION OF SERIAL LINE COMMUNICATIONS

Be sure that your application has the serial line ports properly configured for Modbus before physically connecting the controller to an operational Modbus Serial Line network.

Failure to follow these instructions can result in equipment damage.

Updating Firmware

Follow these steps to update the firmware by an SD card:

Step	Action
1	Download the firmware updates for Modicon M262 Logic/Motion Controller on the Schneider Electric website (in .zip format).
2	Extract the .zip file to the root of the SD card. **NOTE:** The SD card folder \sys\cmd\ contains the download script file.
3	Remove power from the controller.
4	Insert the SD card into the controller.
5	Restore power to the controller. **NOTE:** The SD LED (green) is flashing during the operation.
6	Wait until the end of the download: • If the **SD** LED (green) is ON, the download ended successfully. • If the **SD** LED (yellow) is ON, an error is detected. A script.log file is created in the SD card \sys\cmd\ folder. Contact your Schneider Electric local support.
7	Remove the SD card from the controller. **Result**: If the download ended successfully, the controller restarts automatically with new firmware. The restart is longer than usual.

Updating the Controller Firmware by Controller Assistant

Before Updating Firmware

Performing a firmware update deletes the application program in the controller, including the configuration files, the user management, the user rights and the Boot Application in non-volatile memory. The certificates are not deleted.

NOTICE

LOSS OF APPLICATION DATA

- Perform a backup of the application program to the hard disk of the PC before attempting a firmware update.
- Restore the application program to the device after a successful firmware update.

Failure to follow these instructions can result in equipment damage.

If you remove power to the device, or there is a power outage or communication interruption during the transfer of the application, your device may become inoperative. If a communication interruption or a power outage occurs, reattempt the transfer. If there is a power outage or communication interruption during a firmware update, or if an invalid firmware is used, your device will become inoperative. In this case, use a valid firmware and reattempt the firmware update.

NOTICE

INOPERABLE EQUIPMENT

- Do not interrupt the transfer of the application program or a firmware change once the transfer has begun.
- Re-initiate the transfer if the transfer is interrupted for any reason.
- Do not attempt to place the device into service until the file transfer has completed successfully.

Failure to follow these instructions can result in equipment damage.

The serial line ports of your controller are configured for the CoDeSys protocol by default when new or when you update the controller firmware. The CoDeSys protocol is incompatible with that of other protocols such as Modbus Serial Line. Connecting a new controller to, or updating the firmware of a controller connected to, an active Modbus configured serial line can cause the other devices on the serial line to stop communicating. Make sure that the controller is not connected to an active Modbus serial line network before first downloading a valid application having the concerned port or ports properly configured for the intended protocol.

NOTICE

INTERRUPTION OF SERIAL LINE COMMUNICATIONS

Be sure that your application has the serial line ports properly configured for Modbus before physically connecting the controller to an operational Modbus Serial Line network.

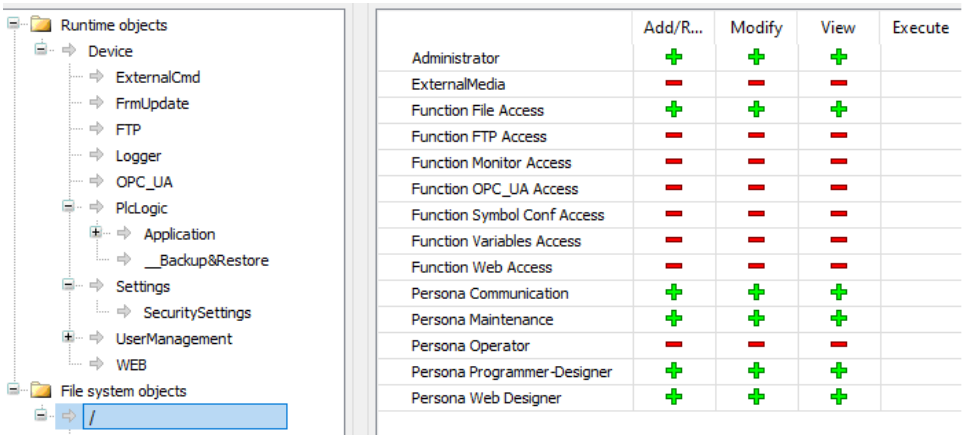
Failure to follow these instructions can result in equipment damage.

User Rights for Firmware Update

To update the firmware by Controller Assistant, you must be part of a user group which has access rights to the **File system objects > /** folder.

In the example below:

- User groups with + symbols have access rights. Users in these groups can update firmware.
- User groups with - symbols do not have access rights. Users in these groups cannot update firmware.



By default:

- **Persona Communication**, **Persona Maintenance**, **Persona Programmer-Designer** and **Persona Web Designer** groups have access rights.
- **Persona Operator** group does not have access rights.

For more information, refer to User Rights, page 86 and Access Rights found in the EcoStruxure Machine Expert Programming Guide.

Updating Firmware

You have two ways to update the firmware by the Controller Assistant:

- Using an SD card
- Writing on the controller

To execute a complete firmware update of a controller, using an SD card, replacing the Boot application and data, in offline mode, proceed as follows:

Step	Action
1	Insert an empty SD card into the PC.
2	Click **Tools > External Tools > Open controller Assistant**.
3	On the **Home** dialog box, click the **Update firmware....** button. **Result**: The **Update firmware (step 1 from 4)** dialog box is displayed.
4	Select the **Controller type** and the **Controller firmware version**.
5	Click the **Next** button. **Result**: The **Update firmware (step 2 from 4)** dialog box is displayed.
6	If needed, change the **Communication settings** and click the **Next** button. **Result**: The **Update firmware (step 3 from 4)** dialog box is displayed.
7	Click the **Write to...** button. **Result**: The **Update firmware (step 4 from 4)** dialog box is displayed.
8	Select your SD card in **Disk drive** and click the **Write** button. At the end of the writing, the **Home** dialog box is displayed.

To execute a complete firmware update of a controller, replacing the Boot application and data, writing on the controller in online mode, proceed as follows:

Step	Action
1	Click **Tools** > **External Tools** > **Open controller Assistant**.
2	On the **Home** dialog box, click the **Update firmware....** button. **Result**: The **Update firmware (step 1 from 4)** dialog box is displayed.
3	Select the **Controller type** and the **Controller firmware version**.
4	Click the **Next** button. **Result**: The **Update firmware (step 2 from 4)** dialog box is displayed.
5	If needed, change the **Communication settings** and click the **Next** button. **Result**: The **Update firmware (step 3 from 4)** dialog box is displayed.
6	Click the **Write on controller...** button. **Result**: The **Update firmware (step 4 from 4)** dialog box is displayed.
7	Select the **Controller** and click the **Connect** button. **Result:** The controller is in `STOPPED` state. At the end of the writing, the **Home** dialog box is displayed. A message is displayed to indicate that you must reboot the controller.

Updating TM3 Expansion Modules Firmware

Overview

The firmware update for the controller and the expansion modules are available on the Schneider Electric website (in .zip format).

Downloading Firmware to TM3 Expansion Modules

The firmware can be updated in:

- TM3X•HSC•
- TM3D• and TM3XTYS4 with firmware version ≥ 28 (SV ≥ 2.0), except TM3DM16R and TM3DM32R
- TM3A• and TM3T• with firmware version ≥ 26 (SV ≥ 1.4)

NOTE: The software version (SV) is found on the packaging and product labels.

Firmware updates are performed if, during a power on, at least one firmware file is present in the `/usr/TM3fwupdate/` directory of controller. You can download the file(s) to the controller using the SD card, an FTP file transfer, or through EcoStruxure Machine Expert.

The controller updates the firmware of the TM3 expansion modules on the I/O bus, including those that are:

- Connected remotely, using a TM3 Transmitter/Receiver module.
- In configurations comprising a mix of TM3 and TM2 expansion modules.

The following table describes how to download firmware to one or more TM3 expansion modules using an SD card:

Step	Action
1	Insert an empty SD card into the PC.
2	Create the folder path `/sys/Cmd` and create a file called *Script.cmd*.
3	Edit the file and insert the following command for each firmware file you wish to transfer to the controller: `Download "/usr/TM3fwupdate/<filename>"`
4	Create the folder path */usr/TM3fwupdate/* in the SD card root directory and copy the firmware files to the *TM3fwupdate* folder.
5	Ensure that power is removed from controller.
6	Remove the SD card from the PC and insert it into the SD card slot of the controller.
7	Restore power to the controller. Wait until the end of the operation (until the **SD** LED is green ON). **Result:** The controller begins transferring the firmware file(s) from the SD card to the */usr/TM3fwupdate* in the controller. During this operation, the **SD** LED on the controller is flashing. A *SCRIPT.log* file is created on the SD card and contains the result of the file transfer. If an error is detected, the **SD** and **ERR** LEDs flash and the detected error is logged in *SCRIPT.log* file.
8	Remove power from the controller.
9	Remove SD card from the controller.
10	Restore power to the controller. **Result:** The controller transfers the firmware file(s) to the appropriate TM3 I/O module (s). **NOTE:** The TM3 update process adds approximately 15 seconds to the controller boot duration.
11	Verify in the message logger of the controller that the firmware is successfully updated: `Your TM3 Module X successfully updated`. X corresponds to the position of the module on the bus. **NOTE:** You can also obtain the logger information in the *PlcLog.txt* file in the */usr/Syslog/* directory of the controller file system. **NOTE:** If the controller encounters an error during the update, the update terminates with that module.
12	If a targeted module was not updated successfully, or there are no message logger messages for all the targeted modules, see the Recovery Procedure, page 243. If all targeted modules were successfully updated, delete the firmware file(s) from */usr/TM3fwupdate/* folder on the controller. You can delete the files directly using EcoStruxure Machine Expert, or by creating and executing a script containing the following command: `Delete "/usr/TM3fwupdate/*"`
13	After the update(s), remove power from the controller (and TM3XREC1 receiver module, if any).
14	Restore power to the controller (and TM3XREC1 receiver module, if any). **Result**: The module(s) is (are) updated.

Recovery Procedure

If you remove power to the device, or there is a power outage or communication interruption during the transfer of the application, your device may become inoperative. If a communication interruption or a power outage occurs, reattempt the transfer. If there is a power outage or communication interruption during a firmware update, or if an invalid firmware is used, your device will become inoperative. In this case, use a valid firmware and reattempt the firmware update.

NOTICE

INOPERABLE EQUIPMENT

- Do not interrupt the transfer of the application program or a firmware change once the transfer has begun.
- Re-initiate the transfer if the transfer is interrupted for any reason.
- Do not attempt to place the device into service until the file transfer has completed successfully.

Failure to follow these instructions can result in equipment damage.

If, during the reattempted firmware update, the update prematurely terminates with an error, it means that the communication interruption or power outage had damaged the firmware of one of your modules in your configuration, and that module must be reinitialized.

NOTE: Once the firmware update process detects an error with the firmware in the destination module, the update process is terminated. After you have reinitialized the damaged module following the recovery procedure, any modules that followed the damaged module remain unchanged and will need to have their firmware updated.

The following table describes how to reinitialize the firmware on TM3 expansion modules:

Step	Action
1	Ensure that the correct firmware is present in the `/usr/TM3fwupdate/` directory of the controller.
2	Remove power from the controller.
3	Disassemble from the controller all TM3 expansion modules that are functioning normally, up to the first module to recover. Refer to the hardware guides of the modules for disassembly instructions.
4	Apply power to the controller. **NOTE:** The TM3 update process adds approximately 15 seconds to the controller boot duration.
5	Verify in the message logger of the controller that the firmware is successfully updated: `Your TM3 Module X successfully updated`. X corresponds to the position of the module on the bus.
6	Remove power from the controller.
7	Reassemble the TM3 expansion module configuration to the controller. Refer to the hardware guides of the modules for assembly instructions.
8	Restore power to the controller. **Result:** The controller transfers the firmware file(s) to the appropriate and yet to be updated TM3 I/O module(s). **NOTE:** The TM3 update process adds approximately 15 seconds to the controller boot duration.
9	Verify in the message logger of the controller that the firmware is successfully updated: `Your TM3 Module X successfully updated`. X corresponds to the position of the module on the bus. **NOTE:** You can also obtain the logger information in the *Sys.log* file in the */usr/Log* directory of the controller file system.
10	Delete the firmware file(s) from */usr/TM3fwupdate/* folder on the controller.

Updating TMSES4 Expansion Module Firmware

Overview

The firmware update for the M262 Logic/Motion Controller are available on the Schneider Electric website (in .zip format).

Updating TMSES4 Module Firmware

The firmware can be updated in TMSES4 module.

Firmware updates are performed using a script file on an SD card.

When the SD card is inserted in the SD card slot of the controller, the controller updates the firmware of the TMSES4 expansion modules on the I/O bus.

Follow these steps to update the firmware by an SD card:

Step	Action
1	Insert an empty SD card into the PC.
2	Create the folder path `/TMS/` in the SD card root directory and copy the two .bin files to the TMS folder. **NOTE:** The smaller file is an information file for checking (reference, version...) which points to the larger file, containing the firmware itself.
3	Remove power from the controller.
4	Remove the SD card from the PC and insert it into the SD card slot of the controller.
5	Restore power to the controller. **Result:** The controller begins transferring the firmware file from the SD card to the updatable expansion modules. During this operation, the **MOD STS** LED on the module is fast flashing green. The firmware update takes until two minutes for each expansion module being updated. Do not remove power from the controller, or remove the SD card, while the operation is in progress. Otherwise, the firmware update may be unsuccessful, and the modules may no longer function correctly.
6	Wait until the end of the download. If the **MOD STS** LED of the module is: • Green ON, the download ended successfully. • Fast flashing red, an error is detected.
7	Verify in the message logger of the controller that the firmware is successfully updated.
8	Remove SD card from the controller.
9	Remove power from the controller, then restore power to the controller. **Result:** The controller restarts automatically with new firmware if the download ended successfully.

If you remove power to the device, or there is a power outage or communication interruption during the transfer of the application, your device may become inoperative. If a communication interruption or a power outage occurs, reattempt the transfer. If there is a power outage or communication interruption during a firmware update, or if an invalid firmware is used, your device will become inoperative. In this case, use a valid firmware and reattempt the firmware update.

NOTICE

INOPERABLE EQUIPMENT

- Do not interrupt the transfer of the application program or a firmware change once the transfer has begun.
- Re-initiate the transfer if the transfer is interrupted for any reason.
- Do not attempt to place the device into service until the file transfer has completed successfully.

Failure to follow these instructions can result in equipment damage.

Managing Script Files

Introduction

The following describes how to write script files (default script file or dynamic script file) to be executed from an SD card or by an application using the ExecuteScript function block (see Modicon M262 Logic/Motion Controller, System Functions and Variables, System Library Guide).

NOTE: If the script file is not executed, a log file is generated. The log file location in the controller is */usr/Syslog/FWLog.txt*.

NOTE: When User Rights are activated on a controller and access rights of the group **ExternalMedia** on objects **ExternalCmd** are denied, scripts used to **Upload**/**Download**/**Delete** files are disabled via SD card scripts (use of the ExecuteScript function block is unaffected by User Rights). For more details about User Rights, refer to the EcoStruxure Machine Expert Programming Guide.

Creating a Script

Introduction

The EcoStruxure Machine Expert script language provides a powerful tool to automatize sequences. You can start single commands or complex command sequences directly from the program environment. For more information on the script, refer to the EcoStruxure Machine Expert Programming Guide.

Before Creating Scripts Using an SD card

The Modicon M262 Logic/Motion Controller accepts only SD cards formatted in FAT or FAT32.

The SD card must have a label. To add a label:

1. Insert the SD card in your PC.
2. Right-click on the drive in Windows Explorer.
3. Choose **Properties**.

⚠WARNING

UNINTENDED EQUIPMENT OPERATION

- You must have operational knowledge of your machine or process before connecting this device to your controller.
- Ensure that guards are in place so that any potential unintended equipment operation will not cause injury to personnel or damage to equipment.

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

When an SD card is inserted into the SD card slot of the controller, the firmware searches and executes the script contained in the SD card (/sys/cmd/Script.cmd).

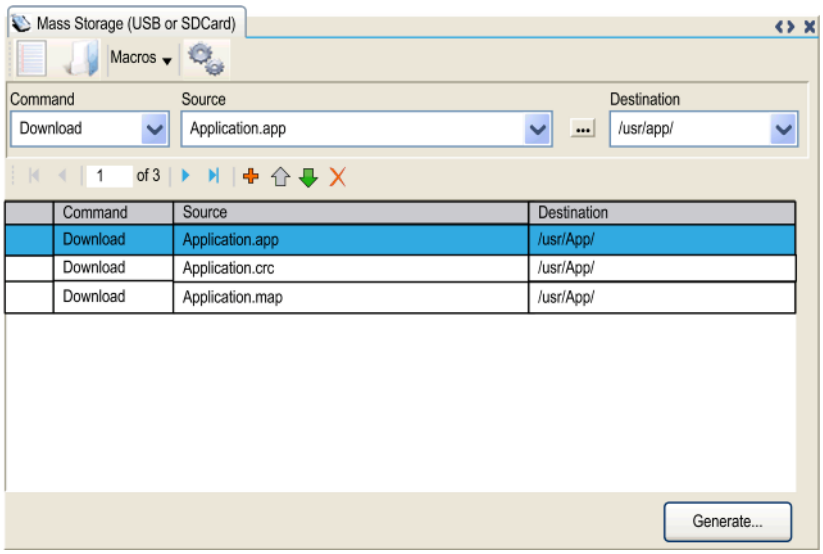
If you remove power to the device, or there is a power outage or communication interruption during the transfer of the application, your device may become inoperative. If a communication interruption or a power outage occurs, reattempt the transfer. If there is a power outage or communication interruption during a firmware update, or if an invalid firmware is used, your device will become inoperative. In this case, use a valid firmware and reattempt the firmware update.

NOTICE
INOPERABLE EQUIPMENT • Do not interrupt the transfer of the application program or a firmware change once the transfer has begun. • Re-initiate the transfer if the transfer is interrupted for any reason. • Do not attempt to place the device into service until the file transfer has completed successfully. **Failure to follow these instructions can result in equipment damage.**

Creating a Script

The following lists the script syntax guidelines:

- If the line begins with a ";", the line is a comment.
- The maximum number of lines in a script file is 50.
- If the syntax is not respected in the script file, the script file is not executed. This means, for example, that the firewall configuration remains in the previous state.

Step	Action
1	Click **Project > Mass Storage (USB or SDCard)** in the main menu. **Result**: The **Mass Storage (USB or SDCard)** tab displays: 
2	Click , then .
3	Select a **Command**.
4	Depending on the selected command, select a **Source** and **Destination**.

Description of the Mass Storage (USB or SDCard) tab

This table describes the **Mass Storage (USB or SDCard)** tab:

Element	Description
New	Create a new script.
Open	Open a script.
Macros	Insert a Macro. A macro is a sequence of unitary commands. A macro helps to perform many common operations such as upload application, download application, and so on.
Generate	Generate the script and all necessary files on the SD card.
Command	Basic instructions.
Source	Source file path on the PC or the controller.
Destination	Destination directory on the PC or the controller.
Add New	Add a script command.
Move Up/ Down	Change the script commands order.
Delete	Delete a script command.

This table describes the commands:

Command	Description	Source	Destination	Syntax
Download	Download a file from the SD card to the controller.	Select the file to download.	Select the controller destination directory.	`'Download "/usr/Cfg/*"'`
SetNode-Name	Sets the node name of the controller.	New node name.	Controller node name	`'SetNodeName "Name_PLC"'`
	Resets the node name of the controller.	Default node name.	Controller node name	`'SetNodeName ""'`
Upload	Upload files contained in a controller directory to the SD card.	Select the directory.	-	`'Upload "/usr/*"'`
Delete	Delete files contained in a controller directory. **NOTE:** Delete "*" does not delete system files.	Select the directory and enter a specific file name. **Important:** By default, all directory files are selected.	-	`'Delete "/usr/SysLog/*"'`
Reboot	Restart the controller (only available at the end of the script).	-	-	`'Reboot'`
changeMod-busPort	Refer to Changing the Modbus TCP Port, page 191.	-	-	`'changeModbu-sPort "portnum"'`

This table describes the macros:

Macros	Description	Directory/Files
Download App	Download the application from the SD card to the controller.	`/usr/App/*.app` `/usr/App/*.crc`
Upload App	Upload the application from the controller to the SD card.	
Download Sources	Download the project archive from the SD card to the controller.	`/usr/App/*.prj`
Upload Sources	Upload the project archive from the controller to the SD card.	
Download Multi-files	Download multiple files from the SD card to a controller directory.	Defined by user
Upload Log	Upload the log files from the controller to the SD card.	`/usr/Log/*.log`

Reset the User Rights to Default

You can manually create a script to remove the user rights, along with the application, from the controller. This script must contain this command:

```
Format "/usr"
Reboot
```

NOTE: This command also removes user application and data.

Step	Action
1	Remove power from the controller.
2	Insert the prepared SD card in the source controller.
3	Restore power to the source controller. **Result:** The copy starts automatically. During the copy, the **PWR** and **I/O** LEDs are ON and the **SD** LED flashes regularly.
4	Wait until the copy is completed. **Result:** The SD LED is ON and the controller reboots with default user rights. If an error was detected, the ERR LED is ON and the controller is in STOPPED state.

Generating Scripts and Files

Generating Existing Scripts and Files

Step	Action
1	Click **Project > Mass Storage (USB or SDCard)...**. **Result:** The **Mass Storage (USB or SDCard)** tab displays.
2	Click **Macros** and select an action from the drop-down list.
3	Select the files to generate.
4	Click **Generate...**.
5	Select the destination folder.

Generating New Scripts and Files

Step	Action
1	Click **Project** > **Mass Storage (USB or SDCard)...**. **Result:** The **Mass Storage (USB or SDCard)** tab displays.
2	Create a script, page 246.
3	Select the files to generate.
4	Click **Generate...**.
5	Select the destination folder.

Transferring Scripts and Files

Before Transferring Scripts and Files

You can transfer scripts and files from and to the controller using an SD card.

The Modicon M262 Logic/Motion Controller accepts only SD cards formatted in FAT or FAT32.

The SD card must have a label. To add a label:

1. Insert the SD card in your PC.
2. Right-click on the drive in Windows Explorer.
3. Choose **Properties**.

⚠WARNING

UNINTENDED EQUIPMENT OPERATION

- You must have operational knowledge of your machine or process before connecting this device to your controller.
- Ensure that guards are in place so that any potential unintended equipment operation will not cause injury to personnel or damage to equipment.

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

When an SD card is inserted into the SD card slot of the controller, the firmware searches and executes the script contained in the SD card (/sys/cmd/Script.cmd).

NOTE: The controller operation is not modified during file transfer.

Performing a firmware update deletes the application program in the device, including the configuration files, the user management, the user rights, the certificates and the Boot Application in non-volatile memory.

NOTICE

LOSS OF APPLICATION DATA

- Perform a backup of the application program to the hard disk of the PC before attempting a firmware update.
- Restore the application program to the device after a successful firmware update.

Failure to follow these instructions can result in equipment damage.

If you remove power to the device, or there is a power outage or communication interruption during the transfer of the application, your device may become inoperative. If a communication interruption or a power outage occurs, reattempt

the transfer. If there is a power outage or communication interruption during a firmware update, or if an invalid firmware is used, your device will become inoperative. In this case, use a valid firmware and reattempt the firmware update.

NOTICE
INOPERABLE EQUIPMENT • Do not interrupt the transfer of the application program or a firmware change once the transfer has begun. • Re-initiate the transfer if the transfer is interrupted for any reason. • Do not attempt to place the device into service until the file transfer has completed successfully. **Failure to follow these instructions can result in equipment damage.**

Transferring

Step	Action
1	Create the script with the **Mass Storage (USB or SDCard)** editor. If necessary, refer to Creating a Script, page 246.
2	Click **Generate...** and select the SD card root directory. **Result**: The script and files are transferred on the SD card.
3	Insert the SD card into the controller. **Result:** The transfer procedure starts and the **SD** LED is flashing during this procedure.
4	Wait until the end of the download: • If the **SD** LED (green) is ON, the download ended successfully. • If the **SD** LED (green) is OFF, and the **ERR** and **I/O** LEDs (red) flash regularly, an error is detected.
5	Remove the SD card from the controller. **NOTE:** Changes are applied after next restart.

When the controller has executed the script, the result is logged on the SD card (file `/sys/cmd/script.log`).

Cloning a Controller

Introduction

The clone function allows you to upload the application from one controller and to download it only to a same controller reference.

This function clones every parameter of the controller (for example applications, firmware, data file, post configuration, remanent variables). Refer to Memory Mapping, page 38.

Cloning the controller is possible by:

- Using an SD Card with a compatible script file
- Using the **FB_ControlClone**
- Using the **Controller Assistant**

When using an SD card, you can also copy the controller firmware and user access rights to the target controller.

NOTE: User access rights can only be copied with an SD Card if the **Include User Rights** button has previously been clicked on the **Maintenance > User Management > Clone Management** subpage of the Web server, page 144.

Before Cloning a Controller

Safety Instructions

If you remove power to the device, or there is a power outage or communication interruption during the transfer of the application, your device may become inoperative. If a communication interruption or a power outage occurs, reattempt the transfer. If there is a power outage or communication interruption during a firmware update, or if an invalid firmware is used, your device will become inoperative. In this case, use a valid firmware and reattempt the firmware update.

NOTICE

INOPERABLE EQUIPMENT

- Do not interrupt the transfer of the application program or a firmware change once the transfer has begun.
- Re-initiate the transfer if the transfer is interrupted for any reason.
- Do not attempt to place the device into service until the file transfer has completed successfully.

Failure to follow these instructions can result in equipment damage.

⚠WARNING

UNINTENDED EQUIPMENT OPERATION

Consult the controller state and behavior diagram in this document to understand the state that will be assumed by the controller after you cycle power.

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

Access Rights

By default, clone is allowed without using the function block **FB_ControlClone**. If you want to restrict access to the clone feature, you can remove the access rights of the `FrmUpdate` object on **ExternalMedia** group, page 88. As a result, cloning will be not allowed without using **FB_ControlClone**.

For more details about this function block, refer to the Modicon M262 Logic/Motion Controller, System Functions and Variables, System Library Guide.

If you wish to control access to the cloned application in the target controller, you must use the **Include Users Rights** button (on the **Clone Management** subpage of the Web server, page 144) of the source controller before doing the clone operation.

For more details about Access Rights, refer to the EcoStruxure Machine Expert Programming Guide.

SD Card Rules

The Modicon M262 Logic/Motion Controller accepts only SD cards formatted in FAT or FAT32.

When an SD card is inserted into the SD card slot of the controller, the firmware searches and executes the script contained in the SD card (/sys/cmd/Script.cmd).

The SD card must have a label. To add a label:

1. Insert the SD card in your PC.
2. Right-click on the drive in Windows Explorer.
3. Choose **Properties**.

⚠ WARNING

UNINTENDED EQUIPMENT OPERATION

- You must have operational knowledge of your machine or process before connecting this device to your controller.
- Ensure that guards are in place so that any potential unintended equipment operation will not cause injury to personnel or damage to equipment.

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

Cloning a Controller

Cloning Procedure

Cloning the controller first removes the existing application from the target controller memory, if user access-rights are being copied to and enabled in the target controller. Refer to the Clone Management, page 144 subpage of the Web server.

Step	Action
1	Erase an SD card and set the card label as follows: **CLONExxx** **NOTE:** The label must begin with '**CLONE**' (not case sensitive), followed by any normal character (a...z, A...Z, 0...9).
2	Select if you want to clone the **Users Rights**. Refer to the Clone Management subpage, page 144 of the Web server.
3	Remove power from the source controller.
4	Insert the prepared SD card in the source controller.
5	Restore power to the source controller. **Result:** The copy starts automatically. During the copy, the **PWR** and **I/O** LEDs are ON and the **SD** LED flashes regularly.
6	Wait until the copy is completed. **Result:** • If the copy is successful, the **SD** LED (green) is ON and turns OFF after the controller starts in normal application mode. • If an error is detected, the **ERR** LED (red) is ON and the controller is in STOPPED state.
7	Remove the SD card from the source controller.
8	Remove power from the target controller.
9	Insert the SD card into the target controller.
10	Restore power to the target controller. **Result:** The paste action starts automatically and the **SD** LED is flashing during this procedure.
11	Wait until the end of the paste action. **Result:** • If the cloning is successful, the **SD** LED (green) is ON. • If an error is detected, the **SD** LED (green) is OFF, and the **ERR** and **I/O** LEDs (red) flash regularly. • If the cloning is done with an error, the **SD** LED (orange) is ON.
12	Remove the SD card to restart the target controller.

NOTE: When they are copied, access rights are only operational after a controller reboot.

Cloned and non cloned directories

For security reasons, not all directories of /usr files are cloned.

This table indicates the cloned and non cloned directories of /usr files:

Directory	Status
App	Cloned
Cfg	Cloned
Dta	Cloned
Fdr	Cloned
Log	Cloned
Other /usr directories	Cloned
pki	Not cloned
Rcp	Cloned
Syslog	Not cloned
Visu	Cloned
Web	Cloned

Compatibility

Software and Firmware Compatibilities

Introduction

Software and Firmware compatibilities are described in the Compatibility and Migration User Guide.

Diagnostic

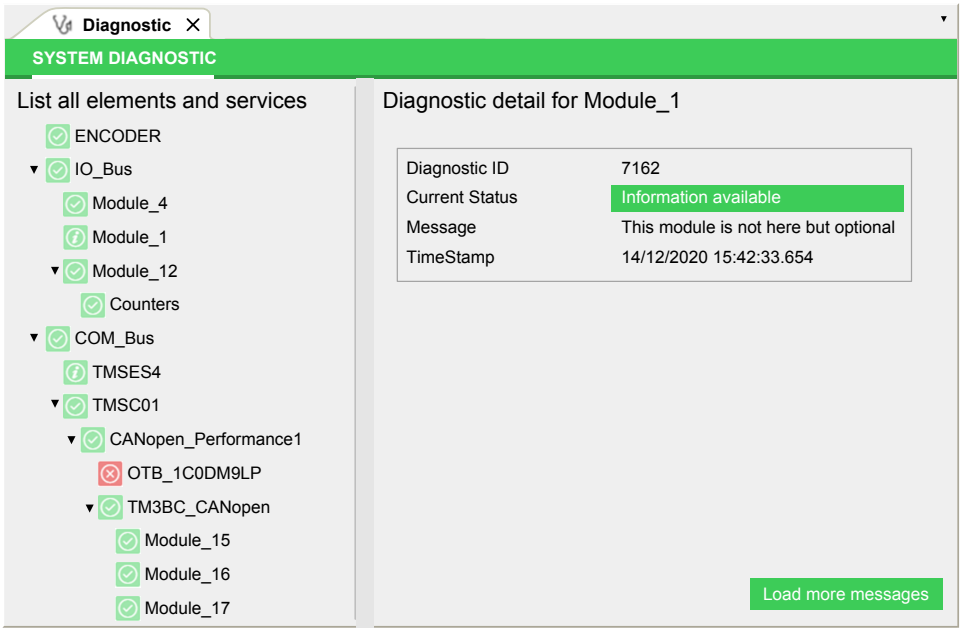
System Diagnostic

Presentation

The **Diagnostic** function displays diagnostic details as messages for the configured elements and services.

System Diagnostic View

To open the diagnostic view, double click **Diagnostic** in the **Devices tree**:



Diagnostic Messages

Diagnostic Messages Objects

Title	Description
Application and SD Card Diagnostic Messages	See M262 Application and SD Card Diagnostic Messages, page 258
Communication Diagnostic Messages	See M262 Communication Diagnostic Messages, page 259
OPC UA Functions Diagnostic Messages	See M262 OPC UA Functions Diagnostic Messages, page 262
M262 Hardware TM3 Expansions Diagnostic Messages	See M262 Hardware TM3 Expansions Diagnostic Messages, page 266
M262 Hardware TMS Expansions Diagnostic Messages	See M262 Hardware TMS Expansions Diagnostic Messages, page 268
M262 Hardware Expansions Diagnostic Messages	See M262 Hardware Expansions Diagnostic Messages, page 269
M262 Synchronized Motion Diagnostic Messages	See M262 Synchronized Motion Diagnostic Messages, page 270
M262 Motion Control Diagnostic Messages	See M262 Motion Control Diagnostic Messages, page 273
M262 Hardware IO Link Diagnostic Messages	See M262 Hardware IO Link Diagnostic Messages, page 275

Generic Diagnostic Messages

Diagnostic ID	Description	Criticality
1	Generic OK message	OK
2	Generic advisory message	Advisory
3	Generic error message	Error
4	Generic test message	Info

M262 Application and SD Card Diagnostic Messages

Diagnostic ID	Description	Causes	Possible Solutions	Criticality
7235	Controller component is running.	-	-	OK
7236	System watchdogs exceeds its threshold condition. The threshold conditions for the three system watchdogs are defined as follows: If all of the tasks require more than 85% of the processor resources for more than 3 seconds, a system error is detected. The controller enters the **HALT** state. If the total execution time of the tasks with priorities between 0 and 24 reaches 100% of processor resources for more than 1 second, an application error is detected. The controller responds with an automatic reboot into the **EMPTY** state. If the lowest priority task of the system is not executed during an interval of 10 seconds, a system error is detected. The controller responds with an automatic reboot into the **EMPTY** state.	-	-	Advisory
7237	There is no application loaded in the controller.	-	-	Info
7238	The controller has a valid application that is stopped.	-	-	Info
7239	The controller is executing a valid application.	-	-	Info
7240	Application is in error, refer to the EcoStruxure Machine Expert Programming Guide to get more details about the error.	-	-	Advisory
7241	-	-	-	OK
7242	Application has to be set in **STOPPED** state, verify the **PLC_R_STOP_CAUSE** data type value to get more details.	-	-	OK
7243	Boot project does not exist in non-volatile memory.	-	-	Info
7244	Boot project in non-volatile memory is the same as the project loaded in memory.	-	-	OK
7245	Boot project in non-volatile memory is different from the project loaded in memory.	-	-	Advisory
7246	Boot project is being created.	-	-	Info
7247	Error detected in the SD card. More details on the error that was detected are written to the file FwLog.txt.	-	-	Advisory
7248	No SD card detected in the slot or the slot is not connected.	-	-	Info
7249	SD card is in read-only mode.	-	-	Info
7250	SD card is in read/write mode.	-	-	Info

M262 Communication Diagnostic Messages

Diagnostic ID	Description	Causes	Possible Solutions	Criticality
7106	The number of Ethernet interfaces to which the DHCP feature is applied is greater than authorized.	Project might be corrupted.	Re-create the project and re-compile.	Error
7107	The Ethernet interface cannot be found for this address.	Project might be corrupted.	Re-create the project and re-compile.	Error
7120	One IP address is set to two network interfaces in the same sub-network.	Incorrect IP address setting.	Verify the IP address setting for each network interface.	Error
7121	The bus coupler TM3 or TM5 is in valid communication state.	-	-	OK
7122	Bus coupler is in communication error.	Incorrect IP address configured or no connection between the controller and the bus coupler.	Verify the connection between controller and bus coupler. Verify the IP address of the bus coupler.	Error
7123	Bus coupler is incorrectly configured.	Configuration of the bus coupler in the software is incorrect.	Verify the configuration of the bus coupler, recompile and re-download the application.	Error
7124	The application was stopped.	-	-	Info
7126	Applied to generic Modbus devices, when the Modbus communication is stopped.	Modbus devices are in stop, no communication.	Run the application.	Info
7127	Applied to generic Modbus devices. The Modbus device is in operational state.	-	-	Info
7133	The module is OK.	-	-	OK
7134	The application is stopped.	-	-	Info
7135	Communication error is detected during the runtime process for slice modules. This module status is managed and sent to the controller by the bus coupler via Ethernet/IP or Modbus protocol.	Several possible causes related to the bus coupler.	Verify the bus coupler status.	Error
7136	The slice module configured in the project is missing physically in the network.	No slice module connected to the bus couple.	Verify the slice module.	Error
7137	Indeterminable status reported by the I/O module.	-	-	Advisory
7138	Ethernet/IP or Modbus I/O Scanner detects communication errors with the bus coupler.	Cable disconnected or disturbances in the network.	Verify cable connection. Verify network configuration: IP address, netmask and gateway address.	Error
7139	Modbus communication is stopped.	-	Application stopped by user.	Advisory
7140	The bus coupler TM3 or TM5 is configured in the project but missing physically.	No bus coupler is connected in the network.	Verify the project or cable connection.	Advisory
7141	Bus coupler is in an error state during the runtime. All sub modules are automatically switched to this state. The application is in **RUNNING** state.	The bus coupler reports an error.	Verify the bus coupler status.	Advisory
7142	Undefined status reported by the bus coupler.	-	-	Advisory
7143	Ethernet/IP device error detected due to incorrect configuration.	Device is incorrectly configured.	Verify the device configuration.	Error
7144	Module is incorrectly configured. This runtime status is managed and sent to the controller by the bus coupler via Ethernet/IP or Modbus protocol.	Configuration of the slice in the software is incorrect.	Verify whether the module in the project is the same as the one present physically.	Error
7701	No memory available for memory allocation.	Memory allocation issue or corrupted project.	Reboot the controller or rebuild the project.	Error
7100	The Ethernet network interface is running without errors.	-	-	OK

Diagnostic ID	Description	Causes	Possible Solutions	Criticality
7101	The Ethernet network interface is not running. **ETH 1** and **ETH 2** correspond to the Ethernet objects as seen in the software. **ETH 3** to **ETH 7** are part of the TMS module which can be added to the controller.	Wiring issue or incorrect IP address setting.	Verify the wiring and verify the network configuration in the software.	Error
7102	The Ethernet network interface IP address duplicated in the network. **ETH 1** and **ETH 2** correspond to the Ethernet objects as seen in the software. **ETH 3** to **ETH 7** are part of the TMS module which can be added to the controller.	IP address duplicated.	Verify whether the IP address is unique in the network.	Error
7103	The Ethernet device is waiting for an IP address from server.	-	-	Info
7104	The Ethernet device is waiting to be configured according to the IP address received from server.	-	-	Info
7105	IP address conflict is detected between two Ethernet interfaces.	Incorrect IP address setting.	Verify IP address configuration for Ethernet interfaces in the project.	Error
7110	When the Ethernet Interface e.g. ETH 1 is in an error state all the sub-modules are set to an error state. No Ethernet communication is possible.	The Ethernet interface is in an error state.	Verify the state of the Ethernet interface	Error
7111	The interface is not allowed to be configured with the specified IP address if the Ethernet scanner or Sercos is set on this interface.	Sercos or Ethernet/IP is configured for this interface.	Verify the setting of the Ethernet interface.	Error
7112	Controller is trying to configure a new gateway different from others inside the same sub-network.	More than one gateway setting detected in the sub-network.	Verify the gateway setting.	Error
7113	No such interface is found.	No such interface is in the project.	Verify the interface setting.	Error
7114	IP address conflict is detected. If the conflict source is NOT from net manager or the task SetIpTask, interface uses the default setting.	Trying to use identical IP address for multiple interfaces.	Verify the IP address setting for each interface.	Error
7115	IP address conflict is detected. If the conflict source is from net manager or the task setIpTask, the setting is stopped and the interface setting keeps previous one.	Trying to use identical IP address for multiple interfaces.	Verify the IP address setting for each interface.	Error
7116	Memory allocation error due to insufficient memory.	Corrupted project.	Reboot controller or rebuild the project.	Error
7117	Network saturation is detected on Ethernet interface.	The saturation could be caused by a network misconfiguration or by an external agent.	Verify your network and security settings.	Error
7118	Network saturation ended.	-	-	Info
7119	The USB Ethernet interface is not working.	USB driver issue.	Verify PC controller-USB driver status on your Operating System. Verify the USB network mask setting.	Error
6100	The USB Ethernet interface setting is incorrect, use the default mask setting (255.255.255.0) instead.	Ethernet network issue: set network mask was unsuccessful.	Verify the USB Ethernet mask setting.	Error

Diagnostic ID	Description	Causes	Possible Solutions	Criticality
6120	The CAN bus communication cannot be established due to an incorrect initialization .	• No TMSCO1/CAN cable connected • Incorrect baudrate • Physical CAN network configuration issue, such as incorrect terminal resistors, incorrect node ID	Connect the TMSCO1 bus, verify the baudrate, the physical connection and the node id.	Error
7800	Modem configuration error or device incommunicative.	The modem might be absent or is not correctly configured.	Verify the wiring or verify the configuration inside the project.	Error

M262 OPC UA Functions Diagnostic Messages

Diagnostic ID	Description	Causes	Possible solutions	Criticality
7905	The system could not allocate memory for this specific element of the Symbol Configuration.	Not enough runtime memory.	Reduce the amount of symbols in the symbol configuration. Try to clean the memory with a reboot.	Error
7903	Name of the symbol in symbol configuration has more than 255 characters.	Symbol name string too long.	Reduce the name of the symbol exposed in the symbol configuration.	Error
7906	The maximum number of symbols to be allocated has been reached. Further symbols will be ignored.	Too many symbols.	Reduce the amount of symbols in the symbol configuration.	Info
7260	Finished creating the OPC UA Server AddressSpace.	-	-	OK
7262	Could not get OPC UA configuration from the application.	Server parameters missing or corrupted.	Make sure OPC UA server configuration is correct. Clear the application from the controller, recompile the application and download the application into the controller again.	Error
7263	Configuration of the OPC UA server in the application is OK.	-	-	OK
7269	Could not allocate memory for a value of an OPC UA node.	Not enough runtime memory.	Try to clean the controller memory. Alternatively try to reduce the size of the array.	Error
7270	Could not allocate memory for a subscription sample value of an OPC UA node.	Not enough runtime memory.	Try to clean the controller memory. Alternatively try to reduce the size of the sample queue of your OPC UA subscription.	Error
7271	Could not allocate memory for a value of an OPC UA node.	Not enough runtime memory.	Try to clean the controller memory. Alternatively try to reduce the size of the array.	Error
7272	Could not allocate memory for a value of an OPC UA node.	Not enough runtime memory.	Try to clean the controller memory. Alternatively try to reduce the size of the string.	Error
7273	Could not get the symbol size from the Symbol Configuration.	Internal error during calculation of the size of the symbol.	Make sure symbol configuration is correct. Clear the application from the controller, recompile the application and symbol configuration, then download the application into the controller again.	Error
7274	Could not allocate memory for a value of an OPC UA node.	Not enough runtime memory.	Try to reduce the number of symbols in the symbol configuration.	Error
7275	Could not instantiate an OPC UA node of this datatype, the datatype is not supported by the OPC UA Server.	Unsupported Symbol DataType.	Change the DataType of the Symbol.	Error
7276	Could not get the symbol corresponding to this OPC UA node.	Interface error.	Make sure symbol configuration is correct. Clear the application from the controller, recompile the application and symbol configuration, then download the application into the controller again.	Error
7277	Could not get the symbol corresponding to this OPC UA node.	Interface error.	Make sure symbol configuration is correct. Clear the application from the controller, recompile the application and symbol configuration, then download the application into the controller again.	Error

Diagnostic ID	Description	Causes	Possible solutions	Criticality
7278	Not enough memory to create symbol list.	Not enough runtime memory.	Try to clean the controller memory. Alternatively try to reduce the amount of symbols in the symbol configuration.	Error
7279	Could not allocate memory for an array of values of an OPC UA node.	Not enough runtime memory.	Try to clean the controller memory. Alternatively try to reduce the size of the array.	Error
7280	Could not get the type description of an array.	Interface error.	Make sure symbol configuration is correct. Clear the application from the controller, recompile the application and symbol configuration, then download the application into the controller again.	Error
7281	The type of symbol is not supported by the OPCU Server.	Unsupported Symbol DataType.	Change the DataType of the Symbol.	Error
7282	The size of the symbol with DataType Wstring is over the limit.	Size of Symbol too big.	Reduce the size of the WSTRING symbol to 126 words or less.	Error
7283	Not enough memory to create symbol list.	Not enough runtime memory.	Try to clean the controller memory. Alternatively try to reduce the amount of symbols in the symbol configuration.	Error
7284	Could not create variable in address space.	Not enough runtime memory.	Try to clean the controller memory. Alternatively try to reduce the amount of symbols in the symbol configuration.	Error
7285	The type of array symbol is not supported by the OPC UA Server.	Unsupported Symbol DataType.	Change the DataType of the Array Symbol.	Error
7286	Could not allocate memory for a value of an OPC UA node.	Not enough runtime memory.	Try to clean the controller memory. Alternatively try to reduce the size of the array.	Error
7287	Could not get the value of an OPC UA node.	Interface error.	Make sure symbol configuration is correct. Clear the application from the controller, recompile the application and symbol configuration, then download the application into the controller again.	Error
7288	Could not allocate a new symbol.	Not enough runtime memory.	Try to clean the controller memory. Alternatively try to reduce the amount of symbols in the symbol configuration.	Error
7289	Could not create the OPC UA server address space.	-	Make sure symbol configuration is correct. Clear the application from the controller, recompile the application and symbol configuration, then download the application into the controller again.	Error
7290	Could not get the OPC UA server configuration from the application.	Interface error.	Make sure OPC UA server configuration is correct. Clear the application from the controller, recompile the application and download the application into the controller again.	Error
7291	Could not allocate memory for the configured OPC UA server endpoints.	Not enough runtime memory.	Try to change the server endpoint in the configuration. Try to adjust the security policy and/or message security.	Error

Diagnostic ID	Description	Causes	Possible solutions	Criticality
7292	Could not initialize the OPC UA stack with the given configuration.	Configuration error.	Make sure OPC UA server configuration is correct. Clear the application from the controller, recompile the application and download the application into the controller again.	Error
7293	Could not create the datatypes table of the OPC UA server.	Configuration error.	Make sure symbol configuration is correct and that exposed datatypes are supported. Clear the application from the controller, recompile the application and symbol configuration, then download the application into the controller again.	Error
7294	Could not add a datatype to the table of datatypes of the OPC UA server.	Configuration error. Not enough runtime memory.	Make sure symbol configuration is correct and that exposed datatypes are supported. Clear the application from the controller, recompile the application and symbol configuration, then download the application into the controller again.	Error
7296	The OPC UA server could not create the folder for the untrusted certificates.	FileSystem error. Not enough filesystem memory.	Clear some space on the controller physical memory.	Advisory
7297	The OPC UA server could not create the folder for the trusted certificates.	FileSystem error. Not enough filesystem memory.	Clear some space on the controller physical memory.	Advisory
7298	The OPC UA server could not create the folder for the revoked certificates list (CRL).	FileSystem error. Not enough filesystem memory.	Clear some space on the controller physical memory.	Advisory
7299	The OPC UA server could not create the folder for the Issuers certificates (other certificates in the certification path).	FileSystem error. Not enough filesystem memory.	Clear some space on the controller physical memory.	Advisory
7900	The OPC UA server could not create the folder for the Issuers certificate revoke list (CRL for other certificates in the certification path).	FileSystem error. Not enough filesystem memory.	Clear some space on the controller physical memory.	Advisory
7901	Could not add the user defined datatype to the table of datatypes of the OPC UA server.	Configuration error.	Make sure symbol configuration is correct and that exposed datatypes are supported. Clear the application from the controller, recompile the application and symbol configuration, then download the application into the controller again.	Error
7902	Finished adding user defined datatypes to the OPC UA server.	-	-	Info
7907	The user has connected to the OPC UA server.	-	-	Info
7908	The session of the user has expired and has timed out. The user has been disconnected.	-	-	Info
7909	You have disconnected from the OPC UA server manually.	-	-	Info
7910	A connection attempt has been made with an incorrect user name or password.	OPC UA Session Authentication.	Verify your authentication credentials in your OPC UA client, and reconnect.	Error
7911	The type of token used for connection authentication is invalid.	OPC UA Session Authentication.	Verify your authentication token in your OPC UA client, make sure it is supported by the server, and reconnect.	Error

Diagnostic ID	Description	Causes	Possible solutions	Criticality
7912	The server has reached a maximum of clients connected simultaneously.	OPC UA Server configuration.	Try to disconnect an unused client, and reconnect with the current one. Try to increase the maximum amount of client connections on the OPC UA server configuration.	Error
7913	The pointed client certificate has expired and is no longer valid for new OPC UA sessions.	-	Try to regenerate the client certificate with a new validity date.	Info
7914	The pointed client certificate is invalid.	-	Ensure that the client certificate respects the OPC UA defined extensions (such as the Alternative Subject OPC UA fields).	Info
7915	The pointed client certificate has been added into the untrusted folder.	-	This usually happens at the client first connection attempt. To accept a connection from this client either trust the certificate or move the certificate to the trusted folder.	Info
7916	The OPC UA server's certificate has been created.	-	-	Info
7917	The pointed client certificate has been trusted (added into the trusted folder).	-	You can now connect to the OPC UA server using this client.	Info
7918	The pointed client certificate has been added into the untrusted folder.	-	This usually happens at the client first connection attempt. To accept a connection from this client either trust the certificate or move the certificate to the trusted folder.	Info
7919	The client certificate did not pass the minimum required verifications.	OPC UA Client Certificate.	Make sure your client certificate is correct. Try to regenerate the client certificate.	Error
7920	The client has disconnected due to a low level transport protocol disconnection.	OPC UA Connection.	Try to reinitialize the client and reconnect. Try to restart the server and reconnect.	Error
7921	A successful connection to the OPC UA server was made from the indicated IP address.	-	-	Info
7922	At least three unsuccessful connections to the OPC UA server were made from the indicated IP address.	OPC UA Session Authentication.	Verify your authentication credentials in your OPC UA client, and reconnect (if it is a known client).	Error
7923	The user has disconnected from the OPC UA server manually (from the given IP address).	-	-	Info

M262 Hardware TM3 Expansions Diagnostic Messages

Diagnostic ID	Description	Causes	Possible Solutions	Criticality
7160	Configuration of I/O expansion bus TM3 done without errors.	-	-	OK
7161	Error configuring the TM3 I/O expansion bus.	The project is misconfigured.	Verify in the project if the configured TM3 slices match with the existing ones.	Error
7162	The optional module is not mandatory. If it is not present, there is no impact on other modules.	The module is missing. As it is an optional module this might be expected.	Verify if the module is missing and if this is intended.	Info
7163	Error detected during the configuration of the module.	The module is missing. The module reference does not correspond to the configured one in the project. The module has the correct reference but the firmware version might be different.	Verify whether the module is connected and that the reference and firmware version are correct.	Error
7164	The reinitialization of the TM3 I/O expansion bus was successful.	-	-	OK
7165	The reinitialization of the TM3 I/O expansion bus was unsuccessful.	The project is misconfigured.	Verify in the project if the configured TM3 slices match with the existing ones.	Error
7166	Indeterminable error during scanning of the internal bus. Identification of the modules was unsuccessful.	A module might be in error.	Verify whether all modules are correctly connected and powered.	Error
7167	Indeterminable error during scanning of the internal bus. Identification of the modules was unsuccessful.	A module might be in error.	Verify whether all modules are correctly connected and powered.	Error
7168	No module found on the bus.	A module might be in error.	Verify whether all modules are correctly connected and powered.	Error
7169	Preparing the update of the module.	-	-	Info
7170	The given firmware file is invalid.	Firmware file is invalid	Verify firmware file and/or download again from Schneider Electric website.	Error
7171	The given firmware file cannot be processed by the controller.	Firmware format is not supported by the controller.	Update the controller firmware version.	Error
7172	Internal bus error during firmware update.	Timeout on the internal bus.	Retry the process.	Error
7173	Internal system error during firmware update.	-	Retry the process.	Error
7174	Module will be updated with the new firmware.	-	-	Info
7175	Module will not be updated. Firmware of the module is already up to date.	-	-	Info
7176	The TM3 firmware for the designated module was updated.	-	-	OK
7177	Internal bus error during firmware update.	A module might be in error.	Retry the process.	Error
7178	Internal system error during firmware update.	A module might be in error.	Retry the process.	Error
7179	Firmware file cannot be processed by the controller.	Firmware format is not recognized by the controller.	Verify that the file is a firmware file. Update the controller firmware version, if necessary.	Error
7180	Module firmware cannot be updated.	Some old I/O modules do not support the firmware update.	Replace the module with a module hardware version that supports firmware updates.	Error

Diagnostic ID	Description	Causes	Possible Solutions	Criticality
7181	Internal error during firmware update.	Indeterminable system error during the process of the firmware update.	Retry the process.	Error
7183	The firmware update process was completed without errors.	-	-	OK
7184	I/O bus is in an error state during the runtime. All sub modules are automatically switched to this state. The application is in **RUNNING** state.	A module might be in error.	-	Advisory
7185	More modules than expected found in the I/O bus.	More modules connected than configured.	Remove extra modules from the bus.	Error

M262 Hardware TMS Expansions Diagnostic Messages

Diagnostic ID	Description	Causes	Possible Solutions	Criticality
6315	Health option requested to the module but no response.	The smart communication module has an internal error or one of its interface is overloaded.	Verify that the network is configured properly on any interface of the given TMS and make sure that it is connected correctly	Error
7210	The configured module is not matching the module found at the current position.	A module might be in error. The configured module is incompatible with the firmware version of the module.	Update the firmware of the module.	Error
7211	The module inserted at this specific position uses an unsupported firmware version.	The firmware version of the module is not supported by the controller firmware version.	Update the controller firmware version. Update the module firmware version.	Error
7212	The configured module is not supported by the controller firmware version.	The configured version of the module is not supported by the controller firmware version.	Update the controller firmware.	Error
7213	The configured module is not matching the module found at the current position.	A module might be in error. The configured module is incompatible with the firmware version of the module.	Update the firmware of the module.	Error
7221	An internal MAC address has been found twice in the configuration.	There is an error with the MAC address of the physical module.	The module in error must be replaced.	Error
7222	An internal MAC address has been found twice in the configuration.	There is an error with the MAC address of the physical module.	The module in error must be replaced.	Error
7223	More than 7 modules found on the bus or in the configuration. This is not supported by the system.	Misconfiguration in the application. There are more than 7 modules inserted on the controller.	Remove the extra module (s) from the configuration or remove them.	Error
7224	No power supply to the module detected.	The wiring of the module might be incorrect. The module might be in error.	Verify the wiring and power supply or change the module.	Error
7225	A module has been configured at this specific place but no module found.	A module is missing.	Change your configuration or insert the missing module on the bus.	Error
7226	A module has been configured at this specific place but no module found.	A module is missing.	Change your configuration or insert the missing module on the bus.	Error
7228	A module has been found on the bus but it is not configured.	An extra module has been found on the bus.	Change your configuration to add it or disconnect the module from the bus.	Error
7229	Internal communication error on the TMS bus.	A module might be in error.	Reboot the controller.	Error
7230	A module has been removed or added on the bus.	Hot swap of modules not supported by the TMS bus	Reboot the controller.	Error
7231	Internal communication error on the TMS while the bus was already stopped.	A module might be in error. Hot swap of modules not supported by the TMS bus.	Reboot the controller.	Error
7232	No module found on the bus.	A module might be in error.	Reboot the controller. Replace the module.	Error
7233	The firmware update is unsuccessful for the given module.	Possible error with the firmware file.	Verify whether the file is the correct one.	Error
7234	Module firmware is not transferred.	Could be that the communication with the TMS was interrupted or trying to transfer an incorrect firmware file.	Verify whether the TMS is correctly connected and that the file is the correct one.	Error
6310	TMS Ethernet discovery error detected.	-	-	Error

Diagnostic ID	Description	Causes	Possible Solutions	Criticality
6311	TMS module configuration error.	-	-	Error
6312	Error assigning the IP address to the TMS.	-	-	Error
6313	VLAN configuration error.	-	-	Error
6314	The Ethernet interface of the TMS is inoperative.	-	-	Error

M262 Hardware Expansions Diagnostic Messages

Diagnostic ID	Description	Causes	Possible Solutions	Criticality
7510	Initialization error detected for I/O devices.	The configuration in the project might be incorrect.	Verify the project.	Error
7511	Configuration error detected for I/O devices.	The configuration in the project might be incorrect.	Verify the project.	Error
7512	Shortcut error detected for I/O devices.	The cabling of the I/O might have errors.	Verify the electrical wiring.	Error
7513	No power supply detected for I/O devices.	The power supply might be disconnected or the cabling might have errors.	Verify the electrical wiring.	Error
7610	No power supply detected for the encoder.	The power supply might be disconnected or the cabling might be incorrect.	Verify the electrical wiring.	Error
7611	No communication with the encoder.	Communication with the encoder is unstable or inoperative.	Verify the wiring.	Error

M262 Synchronized Motion Diagnostic Messages

Diagnostic ID	Description	Causes	Possible Solutions	Criticality
7300	Sercos Phase NRT successfully activated.	-	-	Info
7301	Sercos Phase 0 successfully activated.	-	-	Info
7302	Sercos Phase 1 successfully activated.	-	-	Info
7303	Sercos Phase 2 successfully activated.	-	-	Info
7304	Sercos Phase 3 successfully activated.	-	-	Info
7305	Sercos Phase 4 successfully activated.	-	-	Info
7306	Sercos Simulated Phase 2 successfully activated.	-	-	Info
7307	Sercos Simulated Phase 3 successfully activated.	-	-	Info
7308	Sercos Simulated Phase 4 successfully activated.	-	-	Info
7309	Shows the number of physically connected Sercos devices independently of the device type (I/O or drive).	-	-	Info
7310	Transition to Sercos Phase 0 unsuccessful.	No Sercos devices connected e.g. cable damaged or absent.	-	Error
7311	Transition to Sercos Phase 1 unsuccessful.	-	-	Error
7312	Transition to Sercos Phase 2 unsuccessful.	-	-	Error
7313	Transition to Sercos Phase 3 unsuccessful.	-	-	Error
7314	Transition to Sercos Phase 4 unsuccessful.	-	-	Error
7315	AxisRef was not stopped properly before stopping the controller application.	One axis was running while the controller application has been stopped. An automatic **Errorstop** has been triggered on this axis.	Ensure that all axes have been stopped properly (e.g.: using MC_Stop) before stopping the controller application.	Error
7316	The Sercos master is controlling if the drives are sending a correct connection control in each Sercos cycle. If the master detects an incorrect connection control (usually the NewData-Bit is not toggled correctly). The master creates a Log Message about this. One incorrect connection control is allowed. If a second one follows in the next Sercos cycle the axis connected to the drive will be put to **ErrorStop**.	Connection issue with the Sercos slave.	Verify the wiring for this specific slave.	Advisory
7317	Removing existing Network Address Translation (NAT) rule unsuccessful.	vxWorks rejected NAT rule removal.	Reboot the controller.	Error
7318	Setting NAT rule TCP unsuccessful.	xWorks rejected NAT rule creation for TCP.	Reboot the controller.	Error
7319	Setting NAT rule UDP unsuccessful.	xWorks rejected NAT rule creation for UDP.	Reboot the controller.	Error
7320	The Sercos master netmask, set under Ethernet 1, must be set to 255.255.255.0.	“Sercos master” netmask, set under Ethernet 1, is not set to 255.255.255.0.	The “Sercos master” netmask, set under Ethernet 1, must be set to 255.255.255.0.	Advisory
7321	Two logical devices tried to connect to one physical slave: conflict with Sercos address.	Multiple devices have been configured to the same Sercos address in your application.	Ensure that each device has a unique Sercos address configured in your application.	Error
7322	Two logical devices tried to connect to one physical slave: conflict with topological address.	Multiple devices have been configured to the same Topology address in your application.	Ensure that each device has a unique Topology address configured in your application.	Error

Diagnostic ID	Description	Causes	Possible Solutions	Criticality
7323	Two logical devices tried to connect to one physical slave: conflict between one topological address and a Sercos address.	Multiple devices have been configured to the same Sercos and Topology address in your application.	Ensure that each device has a unique Sercos and Topology address configured in your application.	Error
7324	Sercos slave at a given topological address reports error, Sercos phase change to Phase 2 with duplicate Sercos address.	Multiple devices have the same Sercos address configured in their communication settings.	Ensure that each device has a unique Sercos address configured in their communication settings.	Error
7325	In case that the Sercos address of a device is internally set to 0, or if duplicated Sercos address have been detected. And automatic reassignment of the Sercos address is done by the master.	Settings in the device are blocking the master from reassigning its Sercos address (e.g. a hardware switch is defining the Sercos address).	Reassign manually the device to a free Sercos address.	Info
7326	In case that the Sercos address of a device is internally set to 0, or if duplicated Sercos address have been detected. And automatic reassignment of the Sercos address is done by the master.	Settings in the device are blocking the master from reassigning its Sercos address (e.g. a hardware switch is defining the Sercos address).	Reassign manually the device to a free Sercos address.	Error
7327	IP address already used by Sercos master and configured for another device.	-	Change IP address of Sercos master or Sercos device.	Advisory
7328	IP address already used by another device.	-	Reconfigure IP address of Sercos device.	Advisory
7329	Phase up to phase X is not possible as a CoSeMa error Y is detected.	-	Contact the support.	Error
7330	Phase up to phase X is not possible as a CoSeMa error Y is detected.	-	Contact the support.	Error
7331	Phase up to phase X is not possible as a CoSeMa error Y is detected.	-	Contact the support.	Error
7332	In case that the Sercos address of a device is internally set to 0, or if duplicated Sercos address have been detected. And automatic reassignment of the Sercos address is done by the master.	-	-	Info
7333	In case that the Sercos address of a device is internally set to 0, or if duplicated Sercos address have been detected. And automatic reassignment of the Sercos address is done by the master. Leading to a new phase up initiated by the master.	Duplicated Sercos address identified on the network.	Change the Sercos address assignment in the project Sercos configuration.	Info
7334	No communication with Sercos slave at the given Topological Address.	-	Verify whether the Sercos slave is connected to the master and still operating correctly.	Error
7335	No data received from Sercos slave at the given Topological Address.	The connection control IDN from the slave stopped toggling.	Verify whether the Sercos slave is connected to the master and still operating correctly.	Error
7336	Class 1 error from Sercos slave at Topological Address.	An error is detected on the Sercos slave.	Trigger procedure command S-0-099.	Error
7337	Class 2 error from Sercos slave at Topological Address.	An advisory is detected on the Sercos slave.	-	Advisory
7338	Class 1 error from Sercos slave at Topological Address.	An error is detected on the Sercos slave.	Read IDN S-0-0390.0.0.	Error
7339	Class 2 error from Sercos slave at Topological Address.	An advisory is detected on the Sercos slave.	Read IDN S-0-0390.0.0.	Advisory
7340	Displays the number of devices configured in the application, and how many have been scanned on Sercos by the master.	-	-	Info
7341	Device limit exceeded for CycleTime.	Too many devices have been configured in your application for the cycle time configured.	Increase the cycle time configured or reduce the number of devices configured.	Error

Diagnostic ID	Description	Causes	Possible Solutions	Criticality
7342	Sercos cable disconnected from master.	The Sercos cable has been disconnected from the master.	Ensure that the Sercos cable is connected to the master.	Error
7343	Stack error raised by Sercos master for multiple consecutive cycles.	Sercos slave disconnected from fieldbus or unresponsive.	Ensure that all Sercos slaves are wired correctly and operational.	Error
7344	Stack error raised by Sercos master for multiple consecutive cycles.	Sercos slave did not respond.	Ensure that all Sercos slaves are wired correctly and operational.	Error
7345	RTMP time exceeds the Sercos cycle.	The load of the Real Time Motion.	Reduce the load in your application.	Error
7346	The message 'RTMP time exceeds the Sercos cycle ...' is not displayed further until next Sercos phase up to avoid too many redundant messages.	Process (Motion+Sercos task) have exceeded the maximum allowed load.	Reduce the load in your application.	Error
7347	Displays the Sercos Cycle Time configured in your application in ns.	-	-	Info
7348	There is a logical device configured with Sercos address X, that cannot get mapped to a physical device on Sercos line.	Connection issue with the Sercos slave or an incorrect Sercos address is configured.	Ensure that the configured device is connected to the master, and operational.	Error
7349	There is a logical device configured with topological address X, that cannot get mapped to a physical device on Sercos line.	Connection issue with the Sercos slave or an incorrect topological address is configured.	Ensure that the configured device is connected to the master, and operational.	Error
7350	Phase up to phase X is not possible as a CoSeMa error Y is detected.	No connection to Sercos devices.	Ensure that all Sercos slaves are wired correctly and operational.	Error
7351	Phase up to phase X is not possible as a CoSeMa error Y is detected.	-	-	Error
7352	Phase up to phase X is not possible as a CoSeMa error Y is detected.	Incorrect timing configuration or incorrect process data configuration or incorrect IP address configuration or incorrect device assignment.	Verify your device configuration and application device mapping in the project.	Error
7353	Phase up to phase X is not possible as a CoSeMa error Y is detected.	Maximum limits of real axis reached or duplicated Sercos address.	Reduce the number of physical axes and verify in your project the uniqueness of the Sercos address.	Error
7354	Phase up to phase X is not possible as a CoSeMa error Y is detected.	-	-	Error
7355	Writing data on the given IDN is unsuccessful.	One IDN configuration is unsuccessful.	Re-initiate a new Phase up and ensure that the device is operational.	Error
7356	LXM32S CoplaCommunication module software revision is inferior to that required for correct operation.	-	Update the firmware of the LXM32S Copla module.	Error
7357	LXM32S Software Revision is inferior to that required for correct operation.	-	Update the firmware of the LXM32S.	Error
7358	An undeterminable external exception occurred and terminated either the Lxm32s-Homing or the SercosStateMachine Task.	Indeterminable firmware-generated response.	Reboot the controller.	Error
7359	An undeterminable external exception occurred and terminated the motion task.	Indeterminable firmware-generated response.	Reboot the controller.	Error
7360	An undeterminable internal exception occurred and terminated either the Lxm32s-Homing or the SercosStateMachine Task.	Indeterminable firmware-generated response.	Reboot the controller.	Error
7361	An undeterminable internal exception occurred and terminated the motion task.	Indeterminable firmware-generated response.	Reboot the controller.	Error

M262 Motion Control Diagnostic Messages

Diagnostic ID	Description	Causes	Possible Solutions	Criticality
7400	The axis is blocked by a different function block which cannot be interrupted.	MC_Stop.Execute = TRUE and another motion function block is executed or MC_Home is busy and another motion function block is executed.	Set axis to Standstill.	Error
7401	The power stage must be enabled before the function block can be executed.	MC_Power.Enable = FALSE when executing a motion function block.	Call MCPower with Enable = TRUE.	Error
7402	The function block cannot be repeated as long as the output Busy = TRUE.	A function block is executed again while a previous execution is ongoing.	Ensure that the function block is not busy.	Error
7403	The specified parameter address is not supported by the device.	The parameter address assigned to the MC_ReadParameter or MC_WriteParameter is not supported by the device.	Verify whether the parameter address is correct. Verify whether the parameter you want to access is supported by the device.	Error
7404	The number entered for the signal input is outside of the permissible value range.	Dedicated input is outside the valid value range.	Verify that the value of the input is inside the valid value range.	Error
7405	The number specified for the signal output is outside of the permissible value range.	Dedicated output is outside the valid value range.	Verify that the value of the output is inside the valid value range.	Error
7406	The command is not executed, the device is not ready.	The library is configuring the drive and a function block is executed.	Call the function block MC_ReadAxisInfo and verify that the output ReadyForPowerOn is TRUE.	Error
7407	Communication error detected. The connection to the device has been interrupted.	Incorrect fieldbus settings (address,…) or damaged/incorrect cable.	Verify fieldbus settings (device configuration). Verify wiring (hardware).	Error
7408	The command is not executed within the permissible time delay.	The execution time of the function block exceeds the specified timeout.	Increase the value for the dedicated timeout property.	Error
7409	Value out of range. The value is outside the permissible value range.	Dedicated input is outside the valid value range.	Verify that the value of the input is inside the valid value range.	Error
7410	Buffer full. Internal error detected.	The buffer of the internal FIFO for acyclic data exchange reaches the limit.	Reduce parallel execution of read and write function blocks. Contact your Schneider Electric service representative.	Error
7411	Parameter not supported by device.	The value of the input ParameterNumber assigned to the MC_ReadParameter or MC_WriteParameter is not supported by the device.	Verify whether the value of the input ParameterNumber is correct.	Error
7412	Touch probe number invalid. The specified value for the number of the Touchprobe input is invalid.	Dedicated input is outside the valid value range.	Verify that the value of the input is inside the valid value range.	Error
7413	The specified edge of the Touchprobe input is invalid.	Input TriggerEdge is outside the valid value range or the selected trigger edge is not supported by the drive.	Verify that the value of the input TriggerEdge is inside the valid value range. Verify whether the selected trigger edge is supported by the drive.	Error
7414	Touch probe inactive. An attempt has been made to cancel an inactive Touchprobe.	MC_AbortTrigger is executed for a touch probe which is not active.	Execute MC_AbortTrigger only for active touch probes.	Error
7415	Touch probe active. An attempt has been made to execute an active Touchprobe.	MC_TouchProbe is executed for an already active touch probe.	Execute MC_TouchProbe only for inactive touch probes.	Error

Diagnostic ID	Description	Causes	Possible Solutions	Criticality
7416	The detected error cannot be reset with MC_Reset.	Execution of the function block MC_Reset does not reset the drive error (e.g. STO drive error).	Verify drive state. Restart the device after having remedied the cause of the detected error.	Error
7417	Acceleration out of range. The value for the acceleration is outside of the permissible value range.	Dedicated input is outside the valid value range.	Verify that the value of the input is inside the valid value range.	Error
7418	Deceleration out of range. The value for the deceleration is outside of the permissible value range.	Dedicated input is outside the valid value range.	Verify that the value of the input is inside the valid value range.	Error
7419	Position out of range. The value for the target position is outside of the permissible value range.	Dedicated input is outside the valid value range.	Verify that the value of the input is inside the valid value range.	Error
7420	Velocity out of range. The value for the target velocity is outside of the permissible value range.	Dedicated input is outside the valid value range.	Verify that the value of the input is inside the valid value range.	Error
7421	Torque out of range. The value for the target torque is outside of the permissible value range.	Dedicated input is outside the valid value range.	Verify that the value of the input is inside the valid value range.	Error
7422	Numerator out of range. The value for the numerator is outside of the permissible value range.	Dedicated input is outside the valid value range.	Verify that the value of the input is inside the valid value range.	Error
7423	Denominator out of range. The value for the denominator is outside of the permissible value range.	Dedicated input is outside the valid value range.	Verify that the value of the input is inside the valid value range.	Error
7424	Halt active. The Halt function is active and the command is not executed.	The external halt function of the drive is active.	Verify that the external halt function is not active.	Error
7425	Function block Control_ATV active. The function block cannot be executed as long as the function block Control_ATV is enabled.	A motion function block is executed while the ATV drive is commanded by the Control_ATV function block.	Verify that the Control_ATV function block is not commanding the ATV.	Error
7426	Not ready for applying power. The power stage cannot be enabled in the operating state of the drive.	The drive is not able to apply power (e.g. no main supply).	Verify the drive status.	Error
7427	Incorrect drive type. The function block does not support the linked Axis_Ref type.	The executed function block does not support the drive (e.g. MoveVelocity_LXM32 is executed with ATV axis).	Verify that the executed function block is supported by the drive.	Error
7428	Setpoint source invalid. Invalid value at the input SetpointSource of the function block TorqueControl_ LXM32 or MoveVelocity_LXM32.	The value of the input SetpointSource is outside the valid value range. (Only for the function blocks MoveVelocity_LXM32 and MoveVelocity_SD328A).	Verify that the value of the input SetpointSource is supported by the drive.	Error
7429	The selected homing method is not supported.	The input HomingMode is not supported by the drive.	Verify that the value of the input HomingMode is supported by the drive.	Error
7430	The digital output is set to an incorrect signal output function. Set the signal output function to Freely Available.	Execution of the function block MC_ WriteDigitalOutput for ILX drive and the output is not configured as Freely Available.	Verify that the function of the drive output is Freely Available.	Error
7431	The operating mode is not supported.	The drive does not support the operation mode requested by the executed function block or ATV does not support operation mode profile position or homing.	Verify that the executed function block is supported by the drive.	Error

M262 Hardware IO Link Diagnostic Messages

Diagnostic ID	Description	Causes	Possible Solutions	Criticality
7960	Module is in INACTIVE state.	Configuration choice.	Modify your configuration and download again.	Advisory
7961	Module is in SIO_OUT mode.	Configuration choice.	-	OK
7962	Module is in SIO_IN mode.	Configuration choice.	-	OK
7963	Module is in PREOPERATIONAL state.	Module is in PREOPERATIONAL state.	-	OK
7964	Module is in OPERATIONAL state.	User configuration and startup of the IO-link device.	-	OK
7965	Parameter server data is OK.	User configuration and startup of the IO-link device with the parameter server enabled.	-	OK
7966	Parameter server: Upload ongoing.	Consequence of a user request.	-	OK
7967	Parameter server: Download ongoing.	Consequence of a user request.	-	OK
7968	Parameter server: Undeterminable error.	• Parameter server not supported • Error accessing an object that is managed by the Parameter • Internal Error	Verify the compliance of the IO-link device with the Parameter server needs.	Advisory
7969	Parameter server is locked.	Consequence of a user request.	-	Info
7970	Parameter server is empty.	Parameter server manipulation before filling it with data.	Ensure to download data beforehand.	Info
7971	Parameter server: New serial number recognized.	New device of the same kind of the previous one connected.	Verify that the IO-link device corresponds to the IODD file imported.	Info
7972	Invalid process data.	Invalid process data definition.	Verify the cable and replace if the necessary.	Advisory
7973	No communication.	Incorrect wiring or/and issue in the IO-link device.	Verify that the IO-link device corresponds to the IODD file imported	Error
7974	Device/Vendor ID mismatch.	Incorrect device definition.	Replace with a new IO-link device	Error
7975	Startup error detected.	Issue in the IO-link device during the startup phase.	Verify the hardware configuration relative to the software configuration and the wiring	Error
7976	The IO-link communication module has an issue.	Configuration or wiring error.	-	Advisory

Machine Assistant

Introduction

The Industrial Plug and Work technology supports Machine Assistant. It facilitates the machine configuration through Ethernet network.

Accessing the Web Server Through Industrial Plug and Work

Launching the Web Server

How to Launch the Web Server

This table describes how to launch the Web server:

Step	Action
1	Connect the controller to the PC using an RJ45 cable and open your computer network explorer. **Result**: The controller appears in your computer network explorer.
2	Double-click the controller to access the Web server authentication page.
3	Log in to access the home page of the Web server site., page 132

Using the Machine Assistant

Launching the Machine Assistant

Overview

Machine Assistant is displayed similarly in EcoStruxure Machine Expert and on the controller Web Server. Using this tab, you can monitor the controller and its connected devices.

Launching the Machine Assistant in the Web Server

Launch the Web server, page 276 and log in to access the home page of the Web server site, page 132. Click the **Machine Assistant** tab. The **Machine Assistant** window is displayed.

Launching the Machine Assistant in the Software

Step	Action
1	In EcoStruxure Machine Expert, create a project with a M262 Logic/Motion Controller.
2	Double-click the **Machine Assistant** node in the **Devices tree**. **Result:** The **Machine Assistant** window is displayed.

Managing the Network Scan

Overview

The network scan allows you to detect your controller and all the slave devices connected.

NOTE: EtherNet/IP devices are detected if they are located in the same subnetwork as does the controller.

Scanning the Network in the Web Server

In the **Machine Assistant** window of the Web Server, click the **Scan** button.

Result: The scan is launched and run continuously. All the devices connected to the network are detected.

The scan is stopped when you click **Stop Scan** or close the **Machine Assistant**.

NOTE: The buttons are displayed in the control menu after the run scan has detected devices. Depending on your device, different buttons are displayed.

Scanning the Network in the Software

In EcoStruxure Machine Expert, connect to the controller and click **Launch Scan**.

Result: The scan is launched and run continuously. All the devices connected to the network are detected.

The scan is automatically stopped when you close the **Machine Assistant**.

Scan Status

You must add devices to the project.

This table describes the status of the scan:

Color of the device display	Status
Red	The device exists in the project but is not detected.
Blue	The device is detected but not configured.
Orange	The device is partially detected. The configuration must be updated.

The scan status is automatically updated in EcoStruxure Machine Expert.

NOTE: To view the scan status on the **Machine Assistant** in the Web Server, the relevant .semdt file must be uploaded by clicking the **load .semdt file** button located in the top right corner of the **Machine Assistant** window. Refer to Exporting/Importing .semdt Files, page 280 for more details about . semdt files. The uploaded file is discarded upon navigating to other tabs in the Web Server and must be re-uploaded to view the status.

Updating the Device Configuration

Click **Add/update selected device in project** in EcoStruxure Machine Expert to add or update a device. If a device connected to the controller is not detected, verify that the devices are in the same subnetwork.

Locating a Device

This function allows you to identify your target device. The **Locate** button is displayed in the Web server when a scan is launched and has started to detect devices. When the scan has detected a device, click the **Locate** button to make the LED of the target device flash.

NOTE: the locate service must be supported by your devices. Refer to the devices documentation.

Removing the Network Scan Result

Click the **Clear** button to remove the scan result.

Managing the Devices Network Settings

Setting the IP Address Configuration

You can modify the IPv4 address and subnet mask of your slave device using the command **Set IP Address**:

Step	Action
1	Click the desired device.
2	Click the **locate** button to make the LED of the target device flash.
3	Click the **Set IP Address** command. **Result:** The set IP menu is displayed.
4	Modify the data in the desired fields.
5	Check the **save** box.
6	Click the **Send command** button before closing.

Setting DHCP

You can use DHCP and modify the DHCP name of your slave device using the command **Set DHCP**:

Step	Action
1	Click the desired device.
2	Click the **locate** button to make the LED of the target device flash.
3	Click the **Set DHCP** command. **Result:** The set DHCP menu is displayed.
4	Modify the DHCP network name in the required fields.
5	Check the **save** box.
6	Click the **Send command** button before closing.

NOTE: the network name modification is applied at next power ON.

Setting BOOTP

You can use BOOTP using the command **Set BOOTP**:

Step	Action
1	Click the desired device.
2	Click the **locate** button to make the LED of the target device flash.
3	Click the **Set BOOTP** command. **Result:** The set BOOTP menu is displayed.
4	Check the **save** box.
5	Click the **Send command** button before closing.

Create Link/Delete Link

You can create a network link to a device using the command **Create link**. A link to the device appears and allows users to connect to the device via the Web server. You can delete the link by clicking **Delete http link**. These commands are available by using Machine Assistant in the Web server.

NOTE: You must select the **secure** option to create an operational secured link (HTTPS).

Backing Up/Restoring Configuration

Introduction

You can save and restore the application and firmware of a scanned device.

NOTE: The **Backup** button and the **Restore** button are displayed if a scan has been performed.

Backing Up Configuration

This table describes how to back up the configuration:

Step	Action
1	Insert an SD card in the master controller, page 253.
2	Click the **Locate** button menu to make the LED of the target device flash.
3	Click the **Backup** button under the **commands** menu. **Result:** The backup menu is displayed.
4	Log in (FTP username and password).
5	Click the **Send command** button. **Result:** The saved files are stored in the SD Card.

Restoring Configuration

The **Restore** button is displayed if a backup has been performed.

This table describes how to restore the configuration:

Step	Action
1	Insert the SD card which contains your saved configurations in the source controller, page 253.
2	Click the **Restore** button under the **commands** menu. **Result:** The restore menu is displayed.
3	Log in (FTP username and password).
4	Select the configuration to restore.
5	Click the **Send command** button. **Result:** A message is displayed asking you to reboot the device.
6	Reboot the device and restart the controller.

Exporting/Importing .semdt Files

Introduction

Machine Assistant allows you to export your project when using EcoStruxure Machine Expert, or to export the scan results when using the Web server.

You can import the scan results from the Web server in an empty project in EcoStruxure Machine Expert.

You can also import a project from EcoStruxure Machine Expert to the Web server. You can compare the configured devices to the scanned devices.

Exporting .semdt Files

This table describes how to export an .semdt file from the Web server:

Step	Action
1	Click the **scan** button to scan the connected devices.
2	Click the **Export scan results** button.
3	Save the .semdt file in your PC. **Result:** Your project and the detected devices during the scan are exported.

This table describes how to export an .semdt file from EcoStruxure Machine Expert:

Step	Action
1	Open your project in offline mode.
2	Click the **scan** button to scan your project.
3	Click the **Export configuration as semdt file** button.
4	Save the .semdt file on your PC. **Result:** Your project is exported.

Importing .semdt Files

The **load .semdt file** button allows you to upload a project in EcoStruxure Machine Expert, or scanned devices in the Web server.

Appendices

What's in This Part

How to Change the IP Address of the Controller.....282
Functions to Get/Set Serial Line Configuration in User Program.....284
Controller Performance.....288
M262 Logic/Motion Controller Event Messages 290

Overview

This appendix lists the documents necessary for technical understanding of the Modicon M262 Logic/Motion Controller Programming Guide.

How to Change the IP Address of the Controller

What's in This Chapter

changeIPAddress: Change the IP Address of the Controller 282

changeIPAddress: Change the IP Address of the Controller

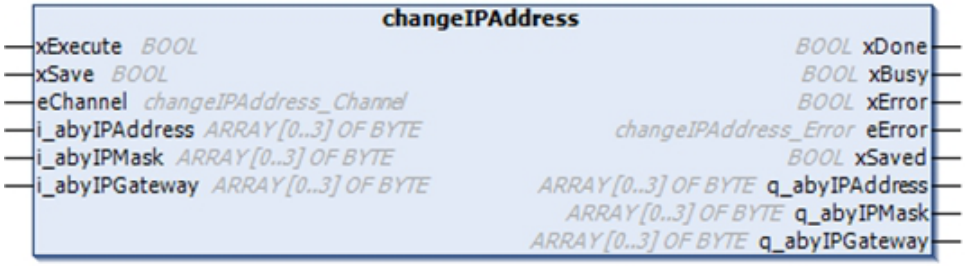
Function Block Description

The *changeIPAddress* function block provides the capability to change dynamically a controller IP address, its subnet mask and its gateway address. The function block can also save the IP address so that it is used in subsequent reboots of the controller.

NOTE: Changing the IP addresses is only possible if the IP mode is configured to **fixed IP address**. For more details, refer to IP Address Configuration, page 123.

NOTE: For more information on the function block, use the **Documentation** tab of the Library Manager Editor. For the use of this editor, refer to the Functions and Libraries User Guide.

Graphical Representation



Parameter Description

Input	Type	Comment
xExecute	BOOL	• Rising edge: action starts. • Falling edge: resets outputs. If a falling edge occurs before the function block has completed its action, the outputs operate in the usual manner and are only reset if either the action is completed or in the event that an error is detected. In this case, the corresponding output values (*xDone*, *xError*, *iError*) are present at the outputs for exactly one cycle.
xSave	BOOL	TRUE: save configuration for subsequent reboots of the controller.
eChannel	changeIPAddress_Channel	The input *eChannel* is the Ethernet port to be configured. Depending on the number of the ports available on the controller, it is one of 5 values, page 283 in *changeIPAddress_Channel* (0 or 1).
i_abyIPAddress	ARRAY[0..3] OF BYTE	The new IP Address to be configured. Format: 0.0.0.0. **NOTE:** If this input is set to 0.0.0.0 then the controller default IP addresses, page 126 is configured.
i_abyIPMask	ARRAY[0..3] OF BYTE	The new subnet mask. Format: 0.0.0.0
i_abyIPGateway	ARRAY[0..3] OF BYTE	The new gateway IP address. Format: 0.0.0.0

Output	Type	Comment
xDone	BOOL	TRUE: if IP Addresses have been successfully configured or if default IP Addresses have been successfully configured because input *i_abyIPAddress* is set to 0.0.0.0.
xBusy	BOOL	Function block active.
xError	BOOL	• TRUE: error detected, function block aborts action. • FALSE: no error has been detected.
eError	changeIPAddress_Error	Error code of the detected error, page 283.
xSaved	BOOL	Configuration saved for the subsequent reboots of the controller.
q_abyIPAddress	ARRAY[0..3] OF BYTE	Current controller IP address. Format: 0.0.0.0.
q_abyIPMask	ARRAY[0..3] OF BYTE	Current subnet mask. Format: 0.0.0.0.
q_abyIPGateway	ARRAY[0..3] OF BYTE	Current gateway IP address. Format: 0.0.0.0.

changeIPAddress_Channel: Ethernet port to be configured

The *changeIPAddress_Channel* enumeration data type contains the following values:

Enumerator	Value	Description
CHANNEL_ETHERNET_NETWORK	0	M241, M251MESC, M258, LMC058, LMC078: **Ethernet port** M251MESE: **Ethernet_2 port**
CHANNEL_DEVICE_NETWORK	1	M241: **TM4ES4 Ethernet port** M251MESE: **Ethernet_1 port**
CHANNEL_M262_ETH1	2	**Ethernet_1 port**
CHANNEL_M262_ETH2	3	**Ethernet_2 port**
CHANNEL_M262_TMS1	4	**1st TMS module**

changeIPAddress_Error: Error Codes

The *changeIPAddress_Error* enumeration data type contains the following values:

Enumerator	Value	Description
ERR_NO_ERROR	00 hex	No error detected.
ERR_UNKNOWN	01 hex	Internal error detected.
ERR_INVALID_MODE	02 hex	IP address is not configured as a fixed IP address.
ERR_INVALID_IP	03 hex	Invalid IP address.
ERR_DUPLICATE_IP	04 hex	The new IP address is already used in the network.
ERR_WRONG_CHANNEL	05 hex	Incorrect Ethernet communication port.
ERR_IP_BEING_SET	06 hex	IP address is already being changed.
ERR_SAVING	07 hex	IP addresses not saved due to a detected error or no non-volatile memory present.
ERR_DHCP_SERVER	08 hex	A DHCP server is configured on this Ethernet communication port.
ERR_IOSCANNER_CONF	09 hex	IO Scanner/Sercos master is configured on this Ethernet communication port.

Functions to Get/Set Serial Line Configuration in User Program

What's in This Chapter

GetSerialConf: Get the Serial Line Configuration 284
SetSerialConf: Change the Serial Line Configuration 285
LinkNumber: Communication Port Number 286
SERIAL_CONF: Structure of the Serial Line Configuration Data Type 287

Overview

This section describes the functions to get/set the serial line configuration in your program.

To use these functions, add the **M2xx Communication** library.

For further information on adding a library, refer to the EcoStruxure Machine Expert Programming Guide.

GetSerialConf: Get the Serial Line Configuration

Function Description

GetSerialConf returns the configuration parameters for a specific serial line communication port.

Graphical Representation

Parameter Description

Input	Type	Comment
Link	*LinkNumber*, page 286	*Link* is the communication port number.
PointerToSerialConf	*PointerToSerialConf*, page 287	*PointerToSerialConf* is the address of the configuration structure (variable of *SERIAL_CONF* type) in which the configuration parameters are stored. The *ADR* standard function must be used to define the associated pointer. Refer to the *SetSerialConf*, page 286 example.

Output	Type	Comment
GetSerialConf	WORD	This function returns: • 0: The configuration parameters are returned • 255: The configuration parameters are not returned because: ◦ the function was not successful ◦ the function is in progress

SetSerialConf: Change the Serial Line Configuration

Function Description

SetSerialConf is used to change the serial line configuration.

Graphical Representation

NOTE: Changing the configuration of the Serial Line(s) port(s) during programming execution can interrupt ongoing communications with other connected devices.

⚠WARNING

LOSS OF CONTROL DUE TO CONFIGURATION CHANGE

Validate and test all the parameters of the *SetSerialConf* function before putting your program into service.

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

Parameter Description

Input	Type	Comment
Link	*LinkNumber*, page 286	*LinkNumber* is the communication port number.
PointerToSerialConf	*PointerToSerialConf*, page 287	*PointerToSerialConf* is the address of the configuration structure (variable of *SERIAL_CONF* type) in which the new configuration parameters are stored. The *ADR* standard function must be used to define the associated pointer. (See the example below.) If 0, set the application default configuration to the serial line.

Output	Type	Comment
SetSerialConf	WORD	This function returns: • 0: The new configuration is set • 255: The new configuration is refused because: ◦ the function is in progress ◦ the input parameters are not valid

Example

```
VAR
  MySerialConf: SERIAL_CONF
  result: WORD;
END_VAR
(*Get current configuration of serial line 1*)
GetSerialConf(1, ADR(MySerialConf));
(*Change to modbus RTU slave address 9*)
MySerialConf.Protocol := 0;          (*Modbus RTU/CoDeSys
protocol (in this case CodesysCompliant selects the
protocol)*)
MySerialConf.CodesysCompliant := 0; (*Modbus RTU*)
MySerialConf.address := 9;          (*Set modbus address to
9*)
(*Reconfigure the serial line 1*)
result := SetSerialConf(1, ADR(MySerialConf));
```

LinkNumber: Communication Port Number

Enumerated Type Description

The *LinkNumber* enumerated data type is a list of the available communication ports. It contains these values:

Enumerator	Value (hex)	Description
USBConsole	00	USB port not available for communication exchanges
COM1	01	Serial COM 1 (embedded serial link)
COM2	02	Serial COM 2
EthEmbed and TMSES4	03	Embedded Ethernet link and TMSES4 expansion modules
CANEmbed	04	Embedded CANopen link
COM3	05	Serial COM 3

If one serial PCI module is installed, then the serial PCI module link is COM2, regardless of the physical PCI slots used.

If two serial PCI modules are installed, then the serial PCI module inserted in the left PCI slot is COM2 and the serial PCI module inserted in the right PCI slot is COM3.

SERIAL_CONF: Structure of the Serial Line Configuration Data Type

Structure Description

The *SERIAL_CONF* structure contains configuration information about the serial line port. It contains these variables:

Variable	Type	Description
Bauds	DWORD	Baud rate
InterframeDelay	WORD	Minimum time (in ms) between 2 frames in Modbus (RTU, ASCII)
FrameReceivedTimeout	WORD	In the ASCII protocol, *FrameReceivedTimeout* allows the system to conclude the end of a frame at reception after a silence of the specified number of ms. If 0 this parameter is not used.
FrameLengthReceived	WORD	In the ASCII protocol, *FrameLengthReceived* allows the system to conclude the end of a frame at reception, when the controller received the specified number of characters. If 0, this parameter is not used.
Protocol	BYTE	0: Modbus RTU or CoDeSys protocol (see *CodesysCompliant* below)
		1: Modbus ASCII
		2: ASCII
Address	BYTE	Modbus address 0 to 255 (0 for Master)
Parity	BYTE	0: none
		1: odd
		2: even
Rs485	BYTE	0: RS232
		1: RS485
ModPol (polarizartion resistor)	BYTE	0: no
		1: yes
DataFormat	BYTE	7 bits or 8 bits
StopBit	BYTE	1: 1 stop bit
		2: 2 stop bits
CharFrameStart	BYTE	In the ASCII protocol, 0 means there is no start character in the frame. Otherwise, the corresponding ASCII character is used to detect the beginning of a frame in receiving mode. In sending mode, this character is added at the beginning of the user frame.
CharFrameEnd1	BYTE	In the ASCII protocol, 0 means there is no second end character in the frame. Otherwise, the corresponding ASCII character is used to detect the end of a frame in receiving mode. In sending mode, this character is added at the end of the user frame.
CharFrameEnd2	BYTE	In the ASCII protocol, 0 means there is no second end character in the frame. Otherwise, the corresponding ASCII character is used (along with *CharFrameEnd1*) to detect the end of a frame in receiving mode. In sending mode, this character is added at the end of the user frame.
CodesysCompliant	BYTE	0: Modbus RTU
		1: CoDeSys protocol (when *Protocol* = 0)
CodesysNetType	BYTE	Not used

Controller Performance

What's in This Chapter

Processing Performance.....288

This chapter provides information about the Modicon M262 Logic/Motion Controller processing performance.

Processing Performance

Introduction

This chapter provides information about the Modicon M262 Logic/Motion Controller processing performance.

Logic Processing

This table presents logic processing performance for various logical instructions:

IL Instruction Type	Duration for 1000 Instructions (µs)	
	TM262L01MESE8T **TM262L10MESE8T** **TM262M05MESS8T** **TM262M15MESS8T**	**TM262L20MESE8T** **TM262M25MESS8T** **TM262M35MESS8T**
Addition/subtraction/multiplication of INT	5	3
Addition/subtraction/multiplication of DINT	5	3
Addition/subtraction of REAL	11	6
Multiplication of REAL	14	7
Division of REAL	39	20
Operation on BOOLEAN (for example: Status:= Status and value)	12	6
LD INT + ST INT	6	3
LD DINT + ST DINT	6	3
LD REAL + ST REAL	6	3

Retain and Retain-Persistent Performance

The retain and retain-persistent variables are saved in a dedicated memory, see NVRAM Memory Organization, page 44. Each read/write access to these variables, impacts the cycle time.

This table presents the performance impact of retain and retain-persistent variables on cycle time during POU execution:

IL Instruction Type	Duration for 1000 variables (µs)	
	TM262L01MESE8T **TM262L10MESE8T** **TM262M05MESS8T** **TM262M15MESS8T**	**TM262L20MESE8T** **TM262M25MESS8T** **TM262M35MESS8T**
READ 1000 INT	434	377
WRITE 1000 INT	418	359
READ 1000 BYTE	434	377
WRITE 1000 BYTE	419	359
READ 1000 DINT	662	685
WRITE 1000 DINT	699	539

Communication and System Processing Time

The communication processing time varies, depending on the number of requests sent and received.

Response Time on Event for Embedded Input

The response time presented in the following table represents the time between a signal rising edge on an input triggering an external task and the edge of an output set by this task:

Minimum	Typical	Maximum
60 µs	80 µs	100 µs

M262 Logic/Motion Controller Event Messages

What's in This Chapter

SysLog Messages from M262 Logic/Motion Controller 290

SysLog Messages from M262 Logic/Motion Controller

Message ID List

This table presents the list of SysLog event messages generated by the M262 Logic/Motion Controller:

ID Code	Message ID
0x001	CONNECTION_SUCCESS, page 290
0x003	CONNECTION_FAILURE, page 291
0x006	DISCONNECTION, page 291
0x20A	CONFIGURATION_CHANGE, page 291
0x403	OPERATING_MODE_CHANGE, page 292
0x406	TAMPERING, page 292
0x501	USERACCOUNT_CHANGE, page 292

CONNECTION_SUCCESS (0x001)

Characteristics	Description
Event Title	Successful connection
Event Description	Successful connections from a user (human or machine) to a machine. It can be through: • Secured protocol • Unsecured protocol if allowed by your security policy • Local interface Local port and local interface are product dependent.
Event Result	Connection OK
Protocols or Service	HTTP FTP CoDeSys Communication OPC UA
Example	<86>1 2018-03-28T23:20:50.52Z "localIpAddr" M262 "Protocol name" CONNECTION_SUCCESS [meta sequenceId='"x"] [authn@3833 itf="localPort" peer="@IpPeer:peerPort" user = "userName"]
Severity	Informational

CONNECTION_FAILURE (0x003)

Characteristics	Description
Event Title	Unsuccessful connection
Event Description	Unsuccessful connections from a user (human or machine) to a machine. It can be through: • Secured protocol • Unsecured protocol if allowed by your security policy • Local interface Standardized reasons are specified in Event Result.
Event Result	Invalid password Indeterminable user Maximum number of connections reached
Protocols or Service	HTTP FTP CoDeSys Communication OPC UA
Example	<85>1 2018-03-28T23:20:50.52Z "localIpAddr" M262 "Protocol name" CONNECTION_FAILURE [meta sequenceId= '"x"][authn@3833 itf="localPort" peer=""peerIpAddr:peerPort" user="userName"] Max connection reached
Severity	Notice

DISCONNECTION (0x006)

Characteristics	Description
Event Title	Disconnection
Event Description	A human or a component disconnected manually or after a time-out due to inactivity. Standardized reasons are specified in Event Result (MSG).
Event Result	Manual logout
Protocols or Service	HTTP CoDeSys Communication OPC UA
Example	<86>1 2018-03-28T23:20:50.52Z "localIpAddr" M262 "Protocol name" DISCONNECTION [meta sequenceId='"x"] [authn@3833 itf="localPort" peer="peerFQDN:peerPort" user="userName"] Manual logout
Severity	Informational

CONFIGURATION_CHANGE (0x20A)

Characteristics	Description
Event Title	Configuration change
Event Description	A new (not cyber-security related) configuration has been successfully uploaded, verified and changed. Standardized objects are Applications, Web Pages.
Event Result	-
Protocols or Service	Configuration
Example	<86>1 2018-03-28T23:20:50.52Z "localIpAddr" M262 Configuration CONFIGURATION_CHANGE [meta sequenceId='"x"] [config@3833 object="Object" value="version"]
Severity	Informational

OPERATING_MODE_CHANGE (0x403)

Characteristics	Description
Event Title	Operating mode change
Event Description	Operating mode change (Run, Stop, Init, …) requested by logged human user. Standardized modes are specified in Event Result (MSG).
Event Result	Init Run Stop
Protocols or Service	System
Example	<85>1 2018-03-28T23:20:50.52Z "localIpAddr" M262 System OPERATING_MODE_CHANGE [meta sequenceId= '"x"] - Init
Severity	Notice

TAMPERING (0x406)

Characteristics	Description
Event Title	Detection of an attack on the system security
Event Description	Detection of hardware tampering (SL3) or detection of flash tampering during secure boot if SysLog supported by bootloader (SL2) or detection of software intrusion (SL4). Standardized intrusions are specified in Event Result (MSG).
Event Result	Network Storm
Protocols or Service	System
Example	<81>1 2018-03-28T23:20:50.52Z "localIpAddr" M262 System TAMPERING [meta sequenceId='"x"] - Physical tamper detection
Severity	Alert

USERACCOUNT_CHANGE (0x501)

Characteristics	Description
Event Title	User account creation, modification or deletion
Event Description	Creation of new ID/password or modification of ID/password or Role Based Access Control (RBAC) levels of authorization. Standardized actions are specified in Event Result (MSG).
Event Result	User account creation User account modification User account deletion Password update
Protocols or Service	Credential
Example	<86>1 2018-03-28T23:20:50.52Z "localIpAddr" M262 Credential USERACCOUNT_CHANGE [meta sequenceId= '"x"][cred@3833 name="UserName"] User account creation
Severity	Informational

Glossary

A

analog input:

Converts received voltage or current levels into numerical values. You can store and process these values within the logic controller.

analog output:

Converts numerical values within the logic controller and sends out proportional voltage or current levels.

application source:

The collection of human-readable controller instructions, configuration data, HMI instructions, symbols, and other program documentation. The application source file is saved on the PC and you can download the application source file to most logic controllers. The application source file is used to build the executable program that runs in the logic controller.

application:

A program including configuration data, symbols, and documentation.

ARP:

(*address resolution protocol*) An IP network layer protocol for Ethernet that maps an IP address to a MAC (hardware) address.

ASIC:

(*application specific integrated circuit*) A silicon processor (chip) custom designed especially for an application.

AT:

(*acknowledge telegram*) On Sercos bus, data are sent by the slaves to the master through AT telegrams (feedback values).

B

BCD:

(*binary coded decimal*) The format that represents decimal numbers between 0 and 9 with a set of 4 bits (a nybble/nibble, also titled as half byte). In this format, the 4 bits used to encode decimal numbers have an unused range of combinations.

For example, the number 2,450 is encoded as `0010 0100 0101 0000`.

BOOL:

(*boolean*) A basic data type in computing. A `BOOL` variable can have one of these values: 0 (`FALSE`), 1 (`TRUE`). A bit that is extracted from a word is of type `BOOL`; for example, `%MW10.4` is a fifth bit of memory word number 10.

Boot application:

(*boot application*) The binary file that contains the application. Usually, it is stored in the controller and allows the controller to boot on the application that the user has generated.

BOOTP:

(*bootstrap protocol*) A UDP network protocol that can be used by a network client to automatically obtain an IP address (and possibly other data) from a server. The client identifies itself to the server using the client MAC address. The server, which maintains a pre-configured table of client device MAC addresses and associated IP addresses, sends the client its pre-configured IP address. BOOTP was originally used as a method that enabled diskless hosts to be remotely booted over a network. The BOOTP process assigns an infinite lease of an IP address. The BOOTP service utilizes UDP ports 67 and 68.

byte:

A type that is encoded in an 8-bit format, ranging from 00 hex to FF hex.

C

CA:

(*Certificate Authority*) An entity that issues digital certificates to certify the ownership of a public key by the named subject of the certificate.

CAE:

(*Cybersecurity Admin Expert*) Schneider Electric software used by Security Administrator to manage substation security.

CFC:

(*continuous function chart*) A graphical programming language (an extension of the IEC 61131-3 standard) based on the function block diagram language that works like a flowchart. However, no networks are used and free positioning of graphic elements is possible, which allows feedback loops. For each block, the inputs are on the left and the outputs on the right. You can link the block outputs to the inputs of other blocks to create complex expressions.

configuration:

The arrangement and interconnection of hardware components within a system and the hardware and software parameters that determine the operating characteristics of the system.

continuous function chart language:

A graphical programming language (an extension of the IEC61131-3 standard) based on the function block diagram language that works like a flowchart. However, no networks are used and free positioning of graphic elements is possible, which allows feedback loops. For each block, the inputs are on the left and the outputs on the right. You can link the block outputs to inputs of other blocks to create complex expressions.

control network:

A network containing logic controllers, SCADA systems, PCs, HMI, switches, ...

Two kinds of topologies are supported:

- flat: all modules and devices in this network belong to same subnet.
- 2 levels: the network is split into an operation network and an inter-controller network.

These two networks can be physically independent, but are generally linked by a routing device.

controller:

Automates industrial processes (also known as programmable logic controller or programmable controller).

CRC:

(*cyclical redundancy check*) A method used to determine the validity of a communication transmission. The transmission contains a bit field that constitutes a checksum. The message is used to calculate the checksum by the transmitter according to the content of the message. Receiving nodes, then recalculate the field in the same manner. Any discrepancy in the value of the 2 CRC calculations indicates that the transmitted message and the received message are different.

CRL:

(*Certificate Revocation List*) A list of digital certificates that have been revoked by the issuing Certificate Authority (CA) before their scheduled expiration date.

D

data log:

The controller logs events relative to the user application in a *data log*.

device network:

A network that contains devices connected to a specific communication port of a logic controller. This controller is seen as a master from the devices point of view.

DHCP:

(*dynamic host configuration protocol*) An advanced extension of BOOTP. DHCP is more advanced, but both DHCP and BOOTP are common. (DHCP can handle BOOTP client requests.)

DINT:

(*double integer type*) Encoded in 32-bit format.

DNS:

(*domain name system*) The naming system for computers and devices connected to a LAN or the Internet.

DWORD:

(*double word*) Encoded in 32-bit format.

E

EDS:

(*electronic data sheet*) A file for fieldbus device description that contains, for example, the properties of a device such as parameters and settings.

equipment:

A part of a machine including sub-assemblies such as conveyors, turntables, and so on.

Ethernet:

A physical and data link layer technology for LANs, also known as IEEE 802.3.

expansion bus:

An electronic communication bus between expansion I/O modules and a controller or bus coupler.

F

FBD:

(*function block diagram*) One of 5 languages for logic or control supported by the standard IEC 61131-3 for control systems. Function block diagram is a graphically oriented programming language. It works with a list of networks, where each network contains a graphical structure of boxes and connection lines, which represents either a logical or arithmetic expression, the call of a function block, a jump, or a return instruction.

FE:

(functional Earth) A common grounding connection to enhance or otherwise allow normal operation of electrically sensitive equipment (also referred to as functional ground in North America).

In contrast to a protective Earth (protective ground), a functional earth connection serves a purpose other than shock protection, and may normally carry current. Examples of devices that use functional earth connections include surge suppressors and electromagnetic interference filters, certain antennas, and measurement instruments.

firmware:

Represents the BIOS, data parameters, and programming instructions that constitute the operating system on a controller. The firmware is stored in non-volatile memory within the controller.

freewheeling:

When a logic controller is in freewheeling scan mode, a new task scan starts as soon as the previous scan has been completed. Contrast with *periodic scan mode*.

FreqGen:

(*frequency generator*) A function that generates a square wave signal with programmable frequency.

FTP:

(*file transfer protocol*) A standard network protocol built on a client-server architecture to exchange and manipulate files over TCP/IP based networks regardless of their size.

G

GRAFCET:

The functioning of a sequential operation in a structured and graphic form.

This is an analytical method that divides any sequential control system into a series of steps, with which actions, transitions, and conditions are associated.

GVL:

(*global variable list*) Manages global variables within a project.

H

HE10:

Rectangular connector for electrical signals with frequencies below 3 MHz, complying with IEC 60807-2.

HSC:

(*high-speed counter*) A function that counts pulses on the controller or on expansion module inputs.

I

I/O:

(*input/output*)

ICMP:

(*Internet control message protocol*) Reports errors detected and provides information related to datagram processing.

IEC 61131-3:

Part 3 of a 3-part IEC standard for industrial automation equipment. IEC 61131-3 is concerned with controller programming languages and defines 2 graphical and 2 textual programming language standards. The graphical programming languages are ladder diagram and function block diagram. The textual programming languages include structured text and instruction list.

IEC:

(*international electrotechnical commission*) A non-profit and non-governmental international standards organization that prepares and publishes international standards for electrical, electronic, and related technologies.

IL:

(*instruction list*) A program written in the language that is composed of a series of text-based instructions executed sequentially by the controller. Each instruction includes a line number, an instruction code, and an operand (refer to IEC 61131-3).

instruction list language:

A program written in the instruction list language that is composed of a series of text-based instructions executed sequentially by the controller. Each instruction includes a line number, an instruction code, and an operand (see IEC 61131-3).

INT:

(*integer*) A whole number encoded in 16 bits.

IP:

(*Internet protocol* Part of the TCP/IP protocol family that tracks the Internet addresses of devices, routes outgoing messages, and recognizes incoming messages.

K

KeepAlive:

Messages sent by the OPC UA server to keep a subscription active. This is necessary when none of the monitored items of data have been updated since the previous publication.

L

ladder diagram language:

A graphical representation of the instructions of a controller program with symbols for contacts, coils, and blocks in a series of rungs executed sequentially by a controller (see IEC 61131-3).

LD:

(*ladder diagram*) A graphical representation of the instructions of a controller program with symbols for contacts, coils, and blocks in a series of rungs executed sequentially by a controller (refer to IEC 61131-3).

LED:

(*light emitting diode*) An indicator that illuminates under a low-level electrical charge.

LINT:

(*long integer*) A whole number encoded in a 64-bit format (4 times `INT` or 2 times `DINT`).

LRC:

(*longitudinal redundancy checking*) An error-detection method for determining the correctness of transmitted and stored data.

LREAL:

(*long real*) A floating-point number encoded in a 64-bit format.

LWORD:

(*long word*) A data type encoded in a 64-bit format.

M

MAC address:

(*media access control address*) A unique 48-bit number associated with a specific piece of hardware. The MAC address is programmed into each network card or device when it is manufactured.

MAST:

A processor task that is run through its programming software. The MAST task has 2 sections:

- **IN:** Inputs are copied to the IN section before execution of the MAST task.
- **OUT:** Outputs are copied to the OUT section after execution of the MAST task.

NOTE:

MDT:

(*master data telegram*) On Sercos bus, an MDT telegram is sent by the master once during each transmission cycle to transmit data (command values) to the servo drives (slaves).

MIB:

(*management information base*) An object database that is monitored by a network management system like SNMP. SNMP monitors devices are defined by their MIBs. Schneider Electric has obtained a private MIB, groupeschneider (3833).

monitored items:

In OPC UA, the items of data (samples) made available by the OPC UA server that clients subscribe to.

MSB:

(*most significant bit/byte* The part of a number, address, or field that is written as the left-most single value in conventional hexadecimal or binary notation.

ms:

(*millisecond*)

MST:

(*master synchronization telegram*) On Sercos bus, an MST telegram is broadcast by the master at the beginning of each transmission cycle to synchronize the timing of the cycle.

N

network:

A system of interconnected devices that share a common data path and protocol for communications.

node:

An addressable device on a communication network.

notifications:

In OPC UA, messages sent by the OPC UA server to inform clients that new items of data are available.

NTP:

(*Network Time Protocol*) is a protocol for synchronizing clocks, within a few milliseconds of Coordinated Universal Time (UTC), of asynchronous computer systems connected over nondeterministic data networks.

NVM:

(*Non-volatile memory*) A non-volatile memory that can be overwritten. It is stored on a special EEPROM that can be erased and reprogrammed.

O

OPC UA:

(*OPC Unified Architecture*) OPC UA is an interoperability standard for the secured and reliable exchange of data in the industrial automation space. It is a platform independent communication protocol using the server/client model. The connection between client and server is commonly based on the reliable transport layer protocol (TCP, Transmission Control Protocol).

For more information about the OPC especially OPC UA refer to the official webpage of the OPC Foundation at https://opcfoundation.org.

OS:

(*operating system*) A collection of software that manages computer hardware resources and provides common services for computer programs.

P

PCI:

(*peripheral component interconnect*) An industry-standard bus for attaching peripherals.

PE:

(*Protective Earth*) A common grounding connection to help avoid the hazard of electric shock by keeping any exposed conductive surface of a device at earth potential. To avoid possible voltage drop, no current is allowed to flow in this conductor (also referred to as *protective ground* in North America or as an equipment grounding conductor in the US national electrical code).

PKI:

(*Public Key Infrastructure*) A system for creating, storing, and distributing digital certificates which are used to verify that a particular public key belongs to a certain entity. The PKI creates digital certificates which map public keys to entities, stores these certificates in a central repository and revokes them if necessary.

post configuration:

(*post configuration*) An option that allows to modify some parameters of the application without changing the application. Post configuration parameters are defined in a file that is stored in the controller. They are overloading the configuration parameters of the application.

POU:

(*program organization unit*) A variable declaration in source code and a corresponding instruction set. POUs facilitate the modular re-use of software programs, functions, and function blocks. Once declared, POUs are available to one another.

program:

The component of an application that consists of compiled source code capable of being installed in the memory of a logic controller.

protocol:

A convention or standard definition that controls or enables the connection, communication, and data transfer between 2 computing system and devices.

PTO:

(*pulse train outputs*) A fast output that oscillates between off and on in a fixed 50-50 duty cycle, producing a square wave form. PTO is especially well suited for applications such as stepper motors, frequency converters, and servo motor control, among others.

publishing interval:

In OPC UA, the frequency at which the OPC_UA server sends notifications to clients informing them that data updates are available.

PWM:

(*pulse width modulation*) A fast output that oscillates between off and on in an adjustable duty cycle, producing a rectangular wave form (though you can adjust it to produce a square wave).

R

REAL:

A data type that is defined as a floating-point number encoded in a 32-bit format.

RJ45:

A standard type of 8-pin connector for network cables defined for Ethernet.

RPDO:

(*receive process data object*) An unconfirmed broadcast message or sent from a producer device to a consumer device in a CAN-based network. The transmit PDO from the producer device has a specific identifier that corresponds to the receive PDO of the consumer devices.

RPI:

(*requested packet interval)* The time period between cyclic data exchanges requested by the scanner. EtherNet/IP devices publish data at the rate specified by the RPI assigned to them by the scanner, and they receive message requests from the scanner with a period equal to RPI.

RSTP:

(*rapid spanning tree protocol*) A high-speed network protocol that builds a loop-free logical topology for Ethernet networks.

RTC:

(*real-time clock*) A battery-backed time-of-day and calender clock that operates continuously, even when the controller is not powered for the life of the battery.

run:

A command that causes the controller to scan the application program, read the physical inputs, and write to the physical outputs according to solution of the logic of the program.

S

sampling rate:

In OPC UA, the frequency at which the OPC UA server reads items of data from connected devices.

scan:

A function that includes:

- reading inputs and placing the values in memory
- executing the application program 1 instruction at a time and storing the results in memory
- using the results to update outputs

SCEP:

(*Simple Certificate Enrollment Protocol*) A certificate management protocol allowing IT administrators to issue certificates automatically by standardizing the exchange with the CA. The certificates can be enrolled on devices on a large-scale.

SDO:

(*service data object*) A message used by the field bus master to access (read/write) the object directories of network nodes in CAN-based networks. SDO types include service SDOs (SSDOs) and client SDOs (CSDOs).

Sercos:

(*serial real-time communications system*) A digital control bus that interconnects, motion controls, drives, I/Os, sensors, and actuators for numerically controlled machines and systems. It is a standardized and open controller-to-intelligent digital device interface, designed for high-speed serial communication of standardized closed-loop real-time data.

SFC:

(*sequential function chart*) A language that is composed of steps with associated actions, transitions with associated logic condition, and directed links between steps and transitions. (The SFC standard is defined in IEC 848. It is IEC 61131-3 compliant.)

SINT:

(*signed integer*) A 15-bit value plus sign.

SNMP:

(*simple network management protocol*) A protocol that can control a network remotely by polling the devices for their status and viewing information related to data transmission. You can also use it to manage software and databases remotely. The protocol also permits active management tasks, such as modifying and applying a new configuration.

STOP:

A command that causes the controller to stop running an application program.

string:

A variable that is a series of ASCII characters.

ST:

(*structured text*) A language that includes complex statements and nested instructions (such as iteration loops, conditional executions, or functions). ST is compliant with IEC 61131-3.

T

task:

A group of sections and subroutines, executed cyclically or periodically for the MAST task or periodically for the FAST task.

A task possesses a level of priority and is linked to inputs and outputs of the controller. These I/O are refreshed in relation to the task.

A controller can have several tasks.

TCP:

(*transmission control protocol*) A connection-based transport layer protocol that provides a simultaneous bi-directional transmission of data. TCP is part of the TCP/IP protocol suite.

terminal block:

(*terminal block*) The component that mounts in an electronic module and provides electrical connections between the controller and the field devices.

TLS:

(*Transport Layer Security*) is a secure protocol technique used to protect information over a computer network.

U

UDINT:

(*unsigned double integer*) Encoded in 32 bits.

UDP:

(*user datagram protocol*) A connectionless mode protocol (defined by IETF RFC 768) in which messages are delivered in a datagram (data telegram) to a destination computer on an IP network. The UDP protocol is typically bundled with the Internet protocol. UDP/IP messages do not expect a response, and are therefore ideal for applications in which dropped packets do not require retransmission (such as streaming video and networks that demand real-time performance).

UINT:

(*unsigned integer*) Encoded in 16 bits.

V

variable:

A memory unit that is addressed and modified by a program.

W

watchdog:

A watchdog is a special timer used to ensure that programs do not overrun their allocated scan time. The watchdog timer is usually set to a higher value than the scan time and reset to 0 at the end of each scan cycle. If the watchdog timer reaches the preset value, for example, because the program is caught in an endless loop, an error is declared and the program stopped.

WORD:

A type encoded in a 16-bit format.

Index

A

Adding an Encoder
  Incremental Encoder 107
  SSI Encoder 107
ASCII Manager 202

C

changeIPAddress 282
  changing the controller IP address 282
changeModbusPort
  command syntax 191
  script example 192
ControlChannel 211
  Enables or disables a communication channel ... 211
controller configuration
  communication settings 77
Controller Configuration
  NTP 83
  PLC Settings 78
  Services 79
cyclic data exchanges, generating EDS file 170

D

Data Types
  LinkNumber 286
DHCP server 168
Download application 72

E

Edge I/O NTS devices 16
EDS file, generating 170
Embedded Functions Configuration
  Embedded I/O Configuration 102
Enables or disables a communication channel
  ControlChannel 211
Ethernet
  changeIPAddress function block 282
  EtherNet/IP device 168
  FTP Server 129
  Modbus TCP Client/Server 128
  Modbus TCP slave device 187
  services 121
  SNMP 131
  Web server 132
EtherNet/IP adapter 169
ExecuteScript example 192
External Event 51

F

Fast Device Replacement 168
features
  key features 14
firewall
  configuration 156
  default script file 156
  script commands 157
firmware
  downloading to TM3 expansion modules 242
  downloading to TMS expansion modules 245
FTP Client 130
FTP Server 129

G

GetSerialConf 284
  getting the serial line configuration 284

H

Hardware Initialization Values 64

I

I/O bus configuration 116
I/O configuration general information
  general practices 111
Industrial Ethernet
  overview 164
Industrial Plug and Work 276
IP address
  changeIPAddress 282

K

KeepAlive (OPC UA) 215
KeepAlive interval (OPC UA) 217

L

libraries 36
LinkNumber 286
  Data Types 286

M

M2•• communication
  GetSerialConf 284
  LinkNumber 286
  SERIAL_CONF 287
  SetSerialConf 285
Machine Assistant 276
Memory Mapping 38
Modbus
  protocols 128
Modbus IOscanner 204
Modbus Manager 199
Modbus TCP Client/Server
  Ethernet 128
Modbus TCP port, changing 191
monitored items (OPC UA) 215

O

OPC UA client 226
OPC UA server
  configuration 216
  KeepAlive interval 217
  overview 215
  publishing interval 217
  sampling interval 217
  selecting symbols 223
  symbols configuration 222
Output Behavior 64–65
Output Forcing 65

overview of the Sercos standard 193

P

Post Configuration 229
baud rate 229
CAE Enable 229
data bits 229
device name 229
example 232
file management 230
FTP 229
gateway address 229
IP address 229
IP configuration mode 229
ODVA Enable 229
parity 229
presentation 229
stop bit 229
subnet mask 229
WebVisualisation 229
programming languages
IL, LD, Grafcet 14
protocols 121
IP 123
Modbus 128
SNMP 131
publishing interval (OPC UA) 215, 217

R

Reboot 70
Remanent variables 73
Reset cold 67
Reset origin 68
Reset origin device 68
Reset warm 67
Run command 66

S

sampling interval (OPC UA) 215, 217
script commands
firewall 157
serial line
ASCII Manager 202
GetSerialConf 284
Modbus Manager 199
SERIAL_CONF 287
SetSerialConf 285
SERIAL_CONF 287
SetSerialConf 285
setting the serial line configuration 285
SNMP
Ethernet 131
protocols 131
Software Initialization Values 65
State diagram 56
Stop command 66
symbols (OPC UA) 222

T

Task
Cyclic task 49
Event task 50
External Event Task 51
Freewheeling task 50
Types 49
Watchdogs 53
TM3 analog I/O modules
downloading firmware to 242
TMS analog I/O modules
downloading firmware to 245
Trend 41

U

updating the firmware of TM3 expansion modules 242
updating the firmware of TMSES4 expansion module 244

W

Web server
Ethernet 132

Schneider Electric
35 rue Joseph Monier
92500 Rueil Malmaison
France

www.se.com

As standards, specifications, and design change from time to time, please ask for confirmation of the information given in this publication.

© 2026 Schneider Electric. All rights reserved.

EIO0000003651.14