

Modicon M262 Logic/Motion Controller Programming Guide

10/2019



The information provided in this documentation contains general descriptions and/or technical characteristics of the performance of the products contained herein. This documentation is not intended as a substitute for and is not to be used for determining suitability or reliability of these products for specific user applications. It is the duty of any such user or integrator to perform the appropriate and complete risk analysis, evaluation and testing of the products with respect to the relevant specific application or use thereof. Neither Schneider Electric nor any of its affiliates or subsidiaries shall be responsible or liable for misuse of the information contained herein. If you have any suggestions for improvements or amendments or have found errors in this publication, please notify us.

You agree not to reproduce, other than for your own personal, noncommercial use, all or part of this document on any medium whatsoever without permission of Schneider Electric, given in writing. You also agree not to establish any hypertext links to this document or its content. Schneider Electric does not grant any right or license for the personal and noncommercial use of the document or its content, except for a non-exclusive license to consult it on an "as is" basis, at your own risk. All other rights are reserved.

All pertinent state, regional, and local safety regulations must be observed when installing and using this product. For reasons of safety and to help ensure compliance with documented system data, only the manufacturer should perform repairs to components.

When devices are used for applications with technical safety requirements, the relevant instructions must be followed.

Failure to use Schneider Electric software or approved software with our hardware products may result in injury, harm, or improper operating results.

Failure to observe this information can result in injury or equipment damage.

© 2019 Schneider Electric. All rights reserved.

Table of Contents



	Safety Information	7
	About the Book	9
Chapter 1	About the Modicon M262 Logic/Motion Controller	15
	M262 Logic/Motion Controller Description	15
Chapter 2	Modicon M262 Motion Controller	19
	Modicon M262 Motion Controller	19
Chapter 3	How to Configure the Controller	21
	Configuring the Controller	21
Chapter 4	Libraries	23
	Libraries	23
Chapter 5	Supported Standard Data Types	25
	Supported Standard Data Types	25
Chapter 6	Memory Mapping	27
	Controller Memory Organization	28
	Flash Memory Organization	30
	RAM Memory Organization	33
	NVRAM Memory Organization	35
	Relocation Table	36
Chapter 7	Tasks	39
	Maximum Number of Tasks	40
	Task Types	41
	Task Configuration Screen	45
	System and Task Watchdogs	47
	Task Priorities	48
	Default Task Configuration	51
Chapter 8	Controller States and Behaviors	53
8.1	Controller State Diagram	54
	Controller State Diagram	55
8.2	Controller States Description	59
	Controller States Description	59
8.3	State Transitions and System Events	63
	Controller States and Output Behavior	64
	Commanding State Transitions	67
	Error Detection, Types, and Management	74
	Remanent Variables	75

Chapter 9	Controller Device Editor	77
	Controller Parameters	78
	Communication Settings	80
	PLC Settings	81
	Services	83
	Ethernet Services	85
	Users Rights	89
Chapter 10	Embedded Inputs and Outputs Configuration	91
10.1	Configuring the Fast I/Os	92
	Embedded I/Os Configuration	92
10.2	Hardware Encoder Interface	98
	Hardware Encoder Interface	99
	Adding an Encoder	101
	Encoder Motion Functions	104
Chapter 11	Expansion Modules Configuration	105
	TM3 I/O Configuration General Description	106
	TM3 I/O Bus Configuration	110
	TMS Expansion Module Configuration	111
	TM3 Expansion Module Configuration	112
	Optional I/O Expansion Modules	113
Chapter 12	Ethernet Configuration	117
12.1	Ethernet Services	118
	Presentation	119
	IP Address Configuration	121
	Modbus TCP Client/Server	127
	Web Server	129
	Symbol Configuration Editor	151
	FTP Server	157
	SNMP	159
	Controller as a Target Device on EtherNet/IP	160
	Controller as a Slave Device on Modbus TCP	184
12.2	Firewall Configuration	189
	Introduction	190
	Dynamic Changes Procedure	192
	Firewall Behavior	193
	Firewall Script Commands	195

Chapter 13	Industrial Ethernet	203
	Industrial Ethernet Presentation	204
	DHCP Server	209
	Fast Device Replacement	210
Chapter 14	Sercos Configuration	211
	Overview of the Sercos Standard	212
	Modicon M262 Logic/Motion Controller Sercos Configuration	214
	Modicon M262 Motion Controller and Safety Controllers with Sercos	215
Chapter 15	Serial Line Configuration	217
	Serial Line Configuration	218
	Machine Expert Network Manager	220
	Modbus Manager	221
	ASCII Manager	225
	Modbus Serial IOScanner	227
	Adding a Device on the Modbus Serial IOScanner	229
	Adding a Modem to a Manager	235
Chapter 16	OPC UA Server Configuration	237
	OPC UA Server Overview	238
	OPC UA Server Configuration	239
	OPC UA Server Symbols Configuration	243
Chapter 17	Post Configuration	245
	Post Configuration Presentation	246
	Post Configuration File Management	247
	Post Configuration Example	249
Chapter 18	Connecting a Modicon M262 Logic/Motion Controller to a PC	251
	Connecting the Controller to a PC	251
Chapter 19	Updating Firmware	253
	Updating the Controller Firmware by SD Card	254
	Updating the Controller Firmware by Controller Assistant	257
	Updating TM3 Expansion Modules Firmware	260
	Updating TMS Expansion Modules Firmware	263
Chapter 20	Managing Script Files	265
	Creating a Script	266
	Generating Scripts and Files	271
	Transferring Scripts and Files	272

Chapter 21	Cloning a Controller	275
	Before Cloning a Controller	276
	Cloning a Controller	278
Chapter 22	Compatibility	281
	Software and Firmware Compatibilities	281
Chapter 23	Industrial Plug and Work	283
23.1	Accessing the Web Server	284
	Launching the Web Server	284
23.2	Using the Machine Assistant	285
	Launching the Machine Assistant	286
	Managing the Network Scan	287
	Managing the Devices Network Settings	289
	Backing Up/Restoring Configuration	290
	Exporting/Importing .semtdt Files	291
Appendices	293
Appendix A	How to Change the IP Address of the Controller	295
	changeIPAddress: Change the IP address of the controller	295
Appendix B	Functions to Get/Set Serial Line Configuration in User Program	299
	GetSerialConf: Get the Serial Line Configuration	300
	SetSerialConf: Change the Serial Line Configuration	301
	SERIAL_CONF: Structure of the Serial Line Configuration Data Type	303
Appendix C	Controller Performance	305
	Processing Performance	305
Glossary	307
Index	317

Safety Information



Important Information

NOTICE

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 Book



At a Glance

Document Scope

The purpose of this document is to help you to 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.

The Modicon M262 Logic/Motion Controller users should read through the entire document to understand all features.

Validity Note

This document has been updated for the release of EcoStruxure™ Machine Expert V1.1 SP1.

The characteristics that are presented in the present document should be the same as those characteristics that appear online. In line with our policy of constant improvement, we may revise content over time to improve clarity and accuracy. If you see a difference between the document and online information, use the online information as your reference.

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) EIO0000003665 (POR) EIO0000003666 (TUR)

Title of Documentation	Reference Number
Modicon TM3 Expansion Modules Configuration - Programming Guide	EIO0000001402 (ENG) EIO0000001403 (FRE) EIO0000001404 (GER) EIO0000001405 (SPA) EIO0000001406 (ITA) EIO0000001407 (CHS)
Modicon TMS Expansion Modules - Programming Guide	EIO0000003691 (ENG) EIO0000003692 (FRE) EIO0000003693 (GER) EIO0000003694 (SPA) EIO0000003695 (ITA) EIO0000003696 (CHS) EIO0000003697 (POR) EIO0000003698 (TUR)
Modicon M262 Logic/Motion Controller - System Library Guide	EIO0000003667 (ENG) EIO0000003668 (FRE) EIO0000003669 (GER) EIO0000003670 (SPA) EIO0000003671 (ITA) EIO0000003672 (CHS) EIO0000003673 (POR) EIO0000003674 (TUR)
Modicon TM3 Expert I/O Modules - HSC Library Guide	EIO0000003683 (ENG) EIO0000003684 (FRE) EIO0000003685 (GER) EIO0000003686 (SPA) EIO0000003687 (ITA) EIO0000003688 (CHS) EIO0000003689 (POR) EIO0000003690 (TUR)
Modicon M262 Logic/Motion Controller - Encoder Library Guide	EIO0000003675 (ENG) EIO0000003676 (FRE) EIO0000003677 (GER) EIO0000003678 (SPA) EIO0000003679 (ITA) EIO0000003680 (CHS) EIO0000003681 (POR) EIO0000003682 (TUR)

Title of Documentation	Reference Number
Embedded Safety for M262 Integration Guide	EIO0000003921 (ENG) EIO0000003923 (FRE) EIO0000003922 (GER) EIO0000003926 (SPA) EIO0000003924 (ITA) EIO0000003925 (CHS)
Sercos for M262 - User Guide	EIO0000003883 (ENG) EIO0000003885 (FRE) EIO0000003884 (GER) EIO0000003888 (SPA) EIO0000003886 (ITA) EIO0000003887 (CHS)
Controller Assistant - User Guide	EIO0000001671 (ENG) EIO0000001672 (FRE) EIO0000001673 (GER) EIO0000001675 (SPA) EIO0000001674 (ITA) EIO0000001676 (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)

You can download these technical publications and other technical information from our website at <https://www.schneider-electric.com/en/download>

Product Related Information

WARNING

LOSS OF CONTROL

- The designer of any control scheme must consider the potential failure modes of control paths and, for certain critical control functions, provide a means to achieve a safe state during and after a path failure. Examples of critical control functions are emergency stop and overtravel stop, power outage and restart.
- Separate or redundant control paths must be provided for critical control functions.
- System control paths may include communication links. Consideration must be given to the implications of unanticipated transmission delays or failures of the link.
- Observe all accident prevention regulations and local safety guidelines.¹
- Each implementation of this equipment must be individually and thoroughly tested for proper operation before being placed into service.

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

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.

Terminology Derived from Standards

The technical terms, terminology, symbols and the corresponding descriptions in this manual, 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 13849-1:2015	Safety of machinery: Safety related parts of control systems. General principles for design.
EN 61496-1:2013	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 62061:2015	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 61784-3:2016	Industrial communication networks - Profiles - Part 3: Functional safety fieldbuses - General rules and profile definitions.
2006/42/EC	Machinery Directive
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.

Chapter 1

About the Modicon M262 Logic/Motion Controller

M262 Logic/Motion Controller Description

Overview

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

Software configuration, programming, and commissioning are accomplished with the EcoStruxure Machine Expert software version 1.1 or later, described in detail in the EcoStruxure Machine Expert Programming Guide as well as the present document.

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

EcoStruxure Machine Expert software can also be used to program these controllers 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.

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 externally 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 (*see page 93*).
- An EcoStruxure Machine Expert software command.
- The system variable PLC_W in a Relocation Table (*see page 36*).
- The Web server (*see page 129*).

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

The following encoder modes are available:

- Incremental mode
- SSI mode

Removable Storage

The M262 Logic/Motion Controllers include 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 (*see page 253*)
- Applying post configuration files to the controller (*see page 246*)
- Storing recipes, files
- Receiving data logging files

Embedded Communication Features

The following types of communication ports are available:

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

Expansion Module and Bus Coupler Compatibility

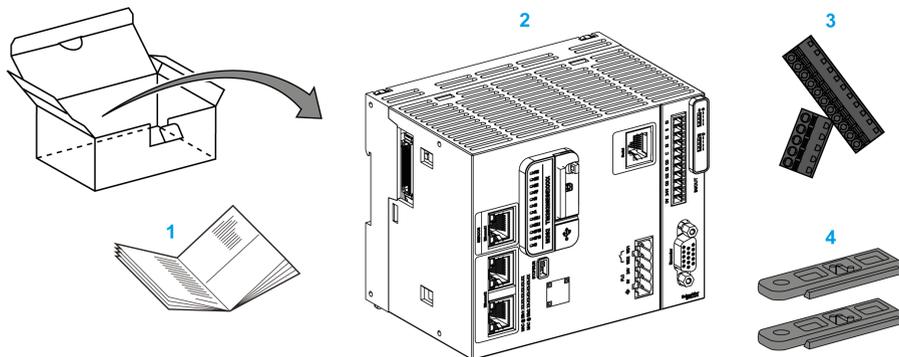
Refer to the compatibility tables in the EcoStruxure Machine Expert - Compatibility and Migration User Guide (see *EcoStruxure Machine Expert Compatibility and Migration, User Guide*).

M262 Logic/Motion Controller

Reference	Digital I/O	Power supply	Communication Ports	Terminal Type	Encoder
M262 Logic Controller: TM262L•	4 fast inputs Source outputs 4 fast 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 Source outputs 4 fast 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

Chapter 2

Modicon M262 Motion Controller

Modicon M262 Motion Controller

Controller Overview

The Schneider Electric Modicon TM262M15MESS8T, TM262M25MESS8T and TM262M35MESS8T are controllers with various powerful features which can control a wide range of motion applications.

The Modicon TM262M• Motion Controller centrally implements the Logic Controller functions and powerful advanced motion functions.

A Modicon TM262M• Motion Controller creates, synchronizes and coordinates the motion functions of a machine for a maximum of 16 axes, synchronized in 2 ms.

These controllers are designed for axis positioning using the EcoStruxure Machine Expert software platform.

Performance Overview

The Modicon TM262M• Motion Controller supports all the features normally available in our 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: LMX32S.
- Non axis Sercos devices are also synchronized with the internal Motion task, for example, TM5NS01 islands or safety-related SLC 100/200 controllers.
- External Encoder
 - External port for Incremental or SSI encoder. The encoder support is fully synchronized with the Motion application. It can be used like a real axis or a virtual axis.
- Fast input
 - 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 move in which the Functions blocks are based on the PLCopen Standard to control easily the position / speed of a single axis.
 - Gearing mode (Master / Slave Function Block).
 - Caming mode, based on recipes, with modification on fly. The recipe can be designed thanks to a cam editor included in EcoStruxure Machine Expert.

Depending 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 non axis Sercos device. These islands are fully configurable. The number of I/O configured increases the load of the Sercos Bus and may lead to an overflow. If an overflow occurs, and assuming your application tolerates it, increase 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
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	16	4	24

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

Chapter 3

How to Configure the Controller

Configuring the Controller

Introduction

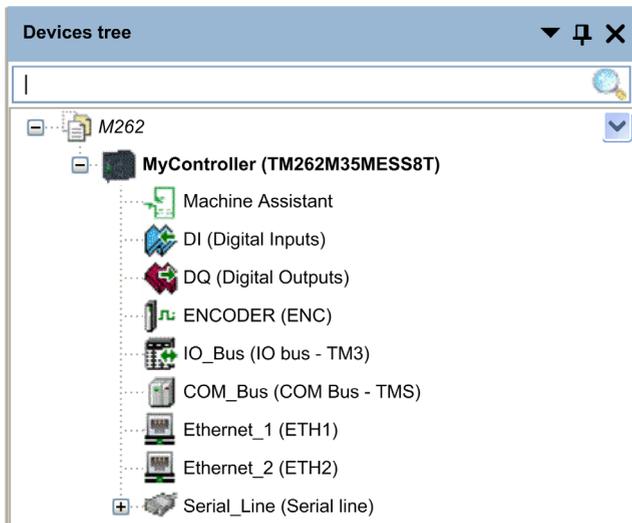
First, create a new project or open an existing project in the EcoStruxure Machine Expert software.

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 current 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...
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 (*see page 146*).

Chapter 4

Libraries

Libraries

Introduction

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

The **Library Manager** of EcoStruxure Machine Expert 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.

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
IoStandard	Standard library for the IO-configuration. This library provides the basic I/O interface for every IEC I/O driver.
Standard	Contains functions and function blocks which are required matching IEC61131-3 as standard POU's for an IEC programming system. Link the standard POU's to the project (standard.library).
Util	Analog Monitors, BCD Conversions, Bit/Byte Functions, Controller Datatypes, Function Manipulators, Mathematical Functions, Signals.
SE_PLCCommunication	Contains communication functions. Most of them are dedicated to Modbus exchange. Communication functions are processed asynchronously with regard to the application task that called the function.
SE_M262 PLCSystem	Contains functions and variables to get diagnostic information and send commands to the controller system.
SE_Relocation Table <i>(see page 36)</i>	Allows you to optimize exchanges between the Modbus client and the controller, by regrouping non-contiguous data into a contiguous table of registers.
M262 Encoder	Contains function blocks and enumerated types for enable, monitor and preset of the encoder, capture of the encoder value in incremental or SSI mode, and read of the active values of the scaling parameters used to compute the unit value.
TM3System	Contains functions and function blocks for TM3 expansion modules.
TMSSystem	Contains the function block and enumerated types for TMS expansion modules.

Chapter 5

Supported Standard Data Types

Supported Standard Data Types

Supported Standard Data Types

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	-	-	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.

Chapter 6

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.

What Is in This Chapter?

This chapter contains the following topics:

Topic	Page
Controller Memory Organization	28
Flash Memory Organization	30
RAM Memory Organization	33
NVRAM Memory Organization	35
Relocation Table	36

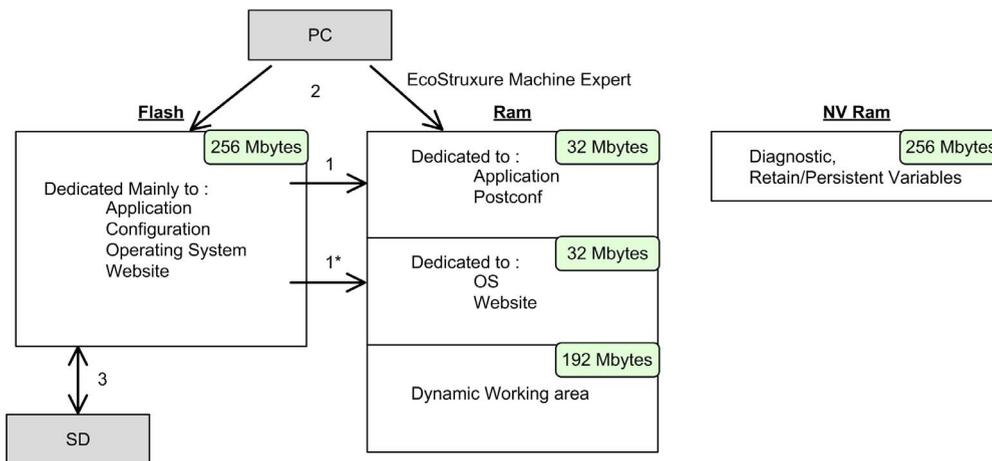
Controller Memory Organization

Introduction

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

- The Flash memory (*see page 30*) contains files (application, configuration files, and so on).
- The Random Access Memory (RAM) is used for application execution.
- The Non-Volatile Random Access Memory (NVRAM) 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 Flash 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: <ul style="list-style-type: none"> • Web server (<i>see page 129</i>) • FTP server (<i>see page 157</i>) • Controller Assistant • EcoStruxure Machine Expert (<i>see EcoStruxure Machine Expert, Programming Guide</i>)
(1) If the controller is in the INVALID_OS state, the only accessible memory is the SD card and only for firmware upgrades.				

Item	Controller State	File Transfer Events	Connection	Description
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 Flash memory does not affect a running application. Any changes to files in Flash memory are taken into account at the next reboot, excepted for the user files directly used by the application.

Flash Memory Organization

Introduction

The Flash 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 Flash memory. The firmware file is applied at next reboot of the controller.

User functions (/usr)	Description
Boot application	This file resides in Flash 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 Flash 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 (<i>see page 189</i>) for more information.
Data logging	Files in which the controller logs events as specified by the application.
<p>(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 EcoStruxure Machine Expert to the controller, you are transferring only the binary executable application directly to RAM.</p> <p>(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 Flash memory.</p>	

File Organization

This table shows the file organization of the flash 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 (<i>see page 245</i>)	Configuration
		CodesysLateConf.cfg	Name of application to launch.	Configuration
		FirewallDefault.cmd	Default firewall settings. By default, this file does not exist. It can be added optionally.	Configuration
/usr	Log	<i>UserDefinedLogName_1.log</i>	All *.log files created using the data logging functions (<i>see EcoStruxure Machine Expert, Data Logging Functions, DataLogging Library Guide</i>). You must specify the total number of files created and the names and contents of each log file using the datalogging feature.	log file
		...		–
		<i>UserDefinedLogName_n.log</i>		–
	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.	Log file
	Visu	–	Used for the Webvisualization feature.	–
/sd0	–	–	SD card. Refer to Managing Script Files (<i>see page 265</i>).	–
	–	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 (*see page 23*).

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 flash memory.

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

File extensions	Flash memory folder
*.app, *.ap_, *.err, *.crc, *.frc, *.prj	/usr/App
*.cfg, *.cf_	/usr/Cfg
*.log	/usr/Log
*.rcp, *.rsi	/usr/Rcp

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) size for different areas of the Modicon M262 Logic/Motion Controller.

Memory Mapping

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 (<i>see page 36</i>) (%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 (<i>see page 36</i>) (%MW62200...%MW63999) This memory is accessible through Modbus requests only. These can be read or write requests.
User area	Symbols
	Variables
	Libraries
	Application

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 modules read-write system variables.
TMS_BUS_DIAG_R	Structure of TMS modules 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 M262 System Library Guide (*see 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. Saved after each modification. This action impacts the cycle time.	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 \(see page 75\)](#). For more information about the impact on performance, see [Processing Performance \(see page 305\)](#).

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 M262 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:

The screenshot shows the 'Relocation Table [MyController_1:PLC Logic: Application]' window. It has a title bar with standard window controls. Below the title bar, there are two sections: 'Read:' and 'Write:'. Each section has a toolbar with icons for adding (+), moving down (green arrow), moving up (green arrow), deleting (red X), copying (document icon), pasting (document icon with arrow), and a right-pointing arrow with an X. Below each toolbar is a table with columns for ID, Variable, Address, Length, and Validity.

Read:

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

Write:

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.AckDigLoopFit	%MW62202	1	True
4	GVL.TempLoop1SetPoint	%MW62203	2	True

Icon	Element	Description
	New Item	Adds an element to the list of system 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.

Chapter 7

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.

What Is in This Chapter?

This chapter contains the following topics:

Topic	Page
Maximum Number of Tasks	40
Task Types	41
Task Configuration Screen	45
System and Task Watchdogs	47
Task Priorities	48
Default Task Configuration	51

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
- Event tasks = 8
- External Event tasks = 8

Special Considerations for Freewheeling

A Freewheeling task (*see page 42*) 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 the System Watchdog (*see Modicon LMC078, Motion Controller, Programming Guide*).

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.

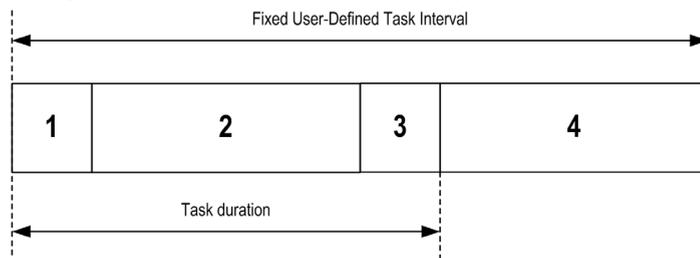
Task Types

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 Configuration subtab for that task. Each Cyclic task type executes as follows:

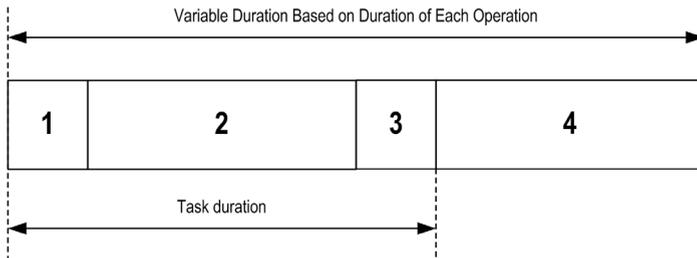


1. **Read Inputs:** The physical input states are written to the $\%I$ input memory variables and other system operations are executed.
2. **Task Processing:** The user code (POU, and so on) defined in the task is processed. The $\%Q$ output memory variables are updated according to your application program instructions but not yet written to the physical outputs during this operation.
3. **Write Outputs:** The $\%Q$ output memory variables are modified with any 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 Modicon M262 Logic/Motion Controller Settings ([see page 81](#)).
For more information on I/O behavior, refer to Controller States Detailed Description ([see page 59](#)).
4. **Remaining Interval time:** The controller firmware carries out system processing and any 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 any 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 Modicon M262 Logic/Motion Controller Settings ([see page 81](#)).
For more information on I/O behavior, refer to Controller States Detailed Description ([see page 59](#)).
- 4. System Processing:** The controller firmware carries out system processing and any 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 ([see page 41](#)).

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 <code>my_Var</code> 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 10 events per millisecond for TM262L10MESE8T and TM262M15MESS8T, and 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
- 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 10 events per millisecond for TM262L10MESE8T and TM262M15MESS8T, and 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 screenshot shows the 'MAST x' Configuration window. It is divided into four main sections:

- Priority (0..31):** A text input field containing the value '1'.
- Type:** A dropdown menu set to 'Cyclic' and an 'Interval (e.g. t#200ms):' field containing '#20ms'.
- Watchdog:** A section with a checked 'Enable' checkbox, a 'Time (e.g. t#200ms):' field containing '100' with a unit dropdown set to 'ms', and a 'Sensitivity:' field containing '1'.
- Table:** A table with two columns: 'POU' and 'Comment'. Above the table is a toolbar with icons for 'Add Call', 'Remove Call', 'Change Call', 'Move Up', 'Move Down', and 'Open POU'.

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

Field Name	Definition
Priority	<p>Configure the priority of each task with a number from 0 to 31 (0 is the highest priority, 31 is the lowest).</p> <p>Only one task at a time can be running. The priority determines when the task runs:</p> <ul style="list-style-type: none"> ● A higher priority task pre-empts a lower priority task ● Tasks with same priority run in turn (2 ms time-slice) <p>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 safety information, refer to Task Priorities (see page 48).</p>
Type	<p>These task types are available:</p> <ul style="list-style-type: none"> ● Cyclic (see page 41) ● Event (see page 43) ● External (see page 44) ● Freewheeling (see page 42)
Watchdog	<p>To configure the watchdog (<i>see Modicon LMC078, Motion Controller, Programming Guide</i>), define these two parameters:</p> <ul style="list-style-type: none"> ● 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	<p>The list of Programming Organization Units (POUs) controlled by the task is defined in the task configuration window (<i>see EcoStruxure Machine Expert, Programming Guide</i>):</p> <ul style="list-style-type: none"> ● 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 currently selected POU of the list by another one, use the command Change Call. ● POUs are executed in the order shown in the list. To move the POUs in the list, select a POU and use the command Move Up or Move Down. <p>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.</p>

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 EcoStruxure Machine Expert 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 EcoStruxure Machine Expert 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 allowable 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 EcoStruxure Machine Expert Programming Guide.

Task Priorities

Task Priority Configuration

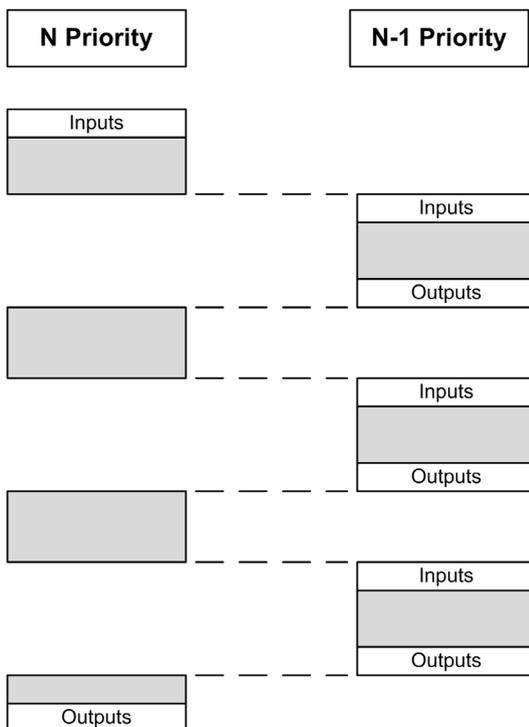
You can configure the priority of each task between 0 and 31 (0 is the highest priority, 31 is the lowest). Each task must have a unique priority. Assigning the same priority to more than one task leads to a build error.

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 (*see page 110*). 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 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 (*see page 48*) 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, EcoStruxure Machine Expert detects an error when you attempt to build the application, and you are not able to download it to the controller.

Chapter 8

Controller States and Behaviors

Introduction

This chapter provides you with 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 EcoStruxure Machine Expert task programming options on the behavior of your system.

What Is in This Chapter?

This chapter contains the following sections:

Section	Topic	Page
8.1	Controller State Diagram	54
8.2	Controller States Description	59
8.3	State Transitions and System Events	63

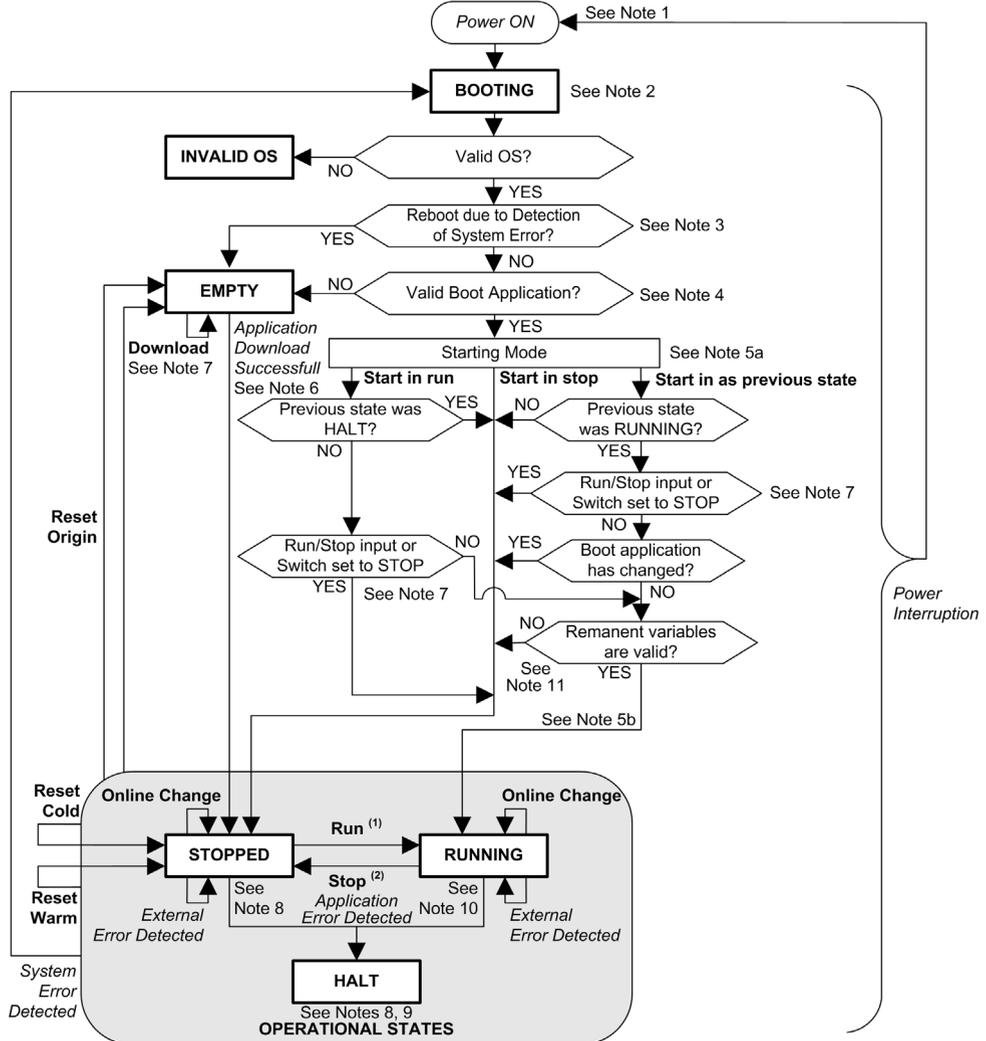
Section 8.1

Controller State Diagram

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 (*see page 67*).
- (2) For details on RUNNING to STOPPED state transition, refer to Stop Command (*see page 67*).

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 Flash memory. However, the Boot application is not deleted from the Flash 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 (*see page 245*) file settings (if any) are applied.

During the load of the boot application, a Check context test occurs to assure that the Remanent variables are valid. If the Check context test is invalid, the boot application loads but the controller assumes STOPPED state (*see page 70*).

Note 5a

The **Starting Mode** is set in the **PLC settings** tab of the **Controller Device Editor** (*see page 81*).

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 Flash 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 normal 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 (*see page 59*) 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 cycle power 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 (*see page 106*).
- RUNNING with Breakpoint: this exception condition is indicated by the RUN LED, which displays a single flash. Refer to Controller States Description (*see page 59*) 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.

Section 8.2

Controller States Description

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.

Controller States Table

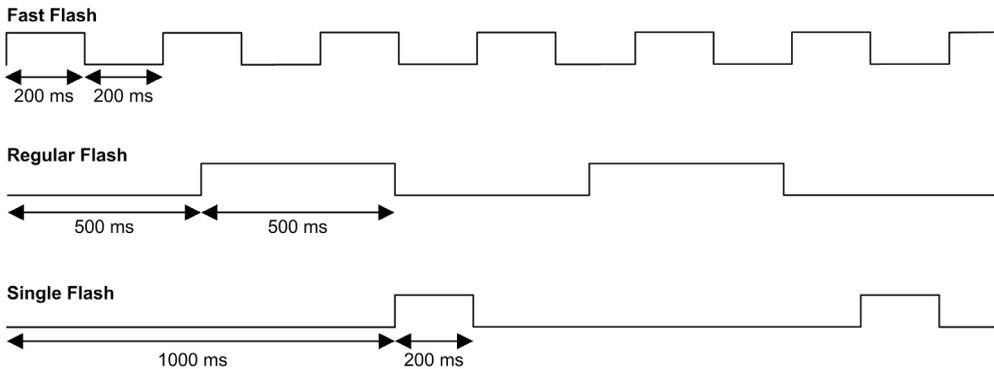
The following tables describe the controller states:

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 flash memory or the firmware is not from Schneider Electric. The controller does not execute the application. Refer to the Updating Firmware (<i>see page 253</i>) 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 normal EMPTY state. But the application is present, and is intentionally not loaded. A next reboot (power cycle), or a new application download, restores 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: <ul style="list-style-type: none"> • 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 EcoStruxure Machine Expert Programming Guide. 	Single flash	OFF	OFF
RUNNING with external error detected	Configuration, TM3, SD card, or other I/O error 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 (<i>see page 61</i>) for an explanation of the behavior of outputs and field buses in this state.	Regular flash	OFF	OFF

Controller State	Description	LED		
		RUN (Green)	ERR (Red)	I/O (Red)
STOPPED with external error detected	Configuration, TM3, SD card, or other I/O error 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 Flash memory. At next power cycle, the application is changed by the one from Flash 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.
- 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 Configuration - Programming Guide (see 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 fully 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 actual values.
- If the **Behaviour for outputs in Stop** field is **Set all outputs to default** the last actual 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.

Section 8.3

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.

What Is in This Section?

This section contains the following topics:

Topic	Page
Controller States and Output Behavior	64
Commanding State Transitions	67
Error Detection, Types, and Management	74
Remanent Variables	75

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. For example, typical controllers define only two options for output behavior in stop: fallback to default value or keep current value.

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 Configuration - Programming Guide (*see 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 (*see page 106*).

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 device 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. Outputs are set to and maintained in their current 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 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 device tree and select **Edit Object**.

This output behavior applies:

- when the controller is going from RUN state to STOPPED state.
- if the controller is going from RUN 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 are set to and maintained in their current 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 (*see page 59*) 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 all other commands (except write immediate) 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 logic controller is in STOP.

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 (*see page 94*) for more information.
- EcoStruxure Machine Expert 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.
- **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, HALT, or EMPTY state.
- The controller is restarted into the RUNNING state automatically under certain conditions.

Refer to Controller State Diagram (*see page 55*) 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 (*see page 94*) for more information.
- EcoStruxure Machine Expert Online Menu: Select the **Stop** command.
- STOP command from WebServer
- 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.
- **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, HALT, 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 (*see page 70*) for further details.
- The controller is restarted into the STOPPED state automatically under certain conditions.

Refer to Controller State Diagram (*see page 55*) for further details.

Reset Warm

Effect: Resets all 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:

- EcoStruxure Machine Expert 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.

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. All 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. All fieldbus communications are stopped and then restarted after the reset is complete.
9. All inputs are reset to their initialization values. All 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 (*see page 245*).

For details on variables, refer to Remanent Variables (*see page 75*).

Reset Cold

Effect: Resets all 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:

- EcoStruxure Machine Expert 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.

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. All 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. All fieldbus communications are stopped and then restarted after the reset is complete.

9. All inputs are reset to their initialization values. All 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 (*see page 245*).
- For details on variables, refer to Remanent Variables (*see page 75*).

Reset Origin

Effect: Resets all variables, including the remanent variables, to their initialization values. Erases all user files on the controller. Places the controller into the EMPTY state.

Starting Conditions: RUNNING, STOPPED, or HALT states.

Methods for Issuing a Reset Origin Command:

- EcoStruxure Machine Expert Online Menu: Select the **Reset origin** command.

Effects of the Reset Origin Command:

1. The application stops.
2. Forcing is erased.
3. The web visu files are erased.
4. The user files (Boot application, Post Configuration, App, App/MFW, Cfg) are erased.
5. Diagnostic indications for errors are reset.
6. The values of the retain variables are reset.
7. The values of the retain-persistent variables are reset.
8. All non-located and non-remanent variables are reset.
9. All fieldbus communications are stopped.
10. Embedded Expert I/O are reset to their previous user-configured default values.
11. All other inputs are reset to their initialization values.
All other outputs are reset to their hardware initialization values.
12. User rights are deleted.
13. Connection with EcoStruxure Machine Expert is maintained online.
14. Security certificates are erased.
15. System Logs are maintained.
16. The controller reboots.

For details on variables, refer to Remanent Variables (*see page 75*).

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 set to 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.

The Reboot was provoked by a script and:

- the **Starting Mode** is set to **Start in run**, and if the Run/Stop input or switch 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.

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 set to 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 set to 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 or switch is configured and 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 or switch is not configured, and if the controller was not in 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. All 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. All fieldbus communications are stopped and restarted after the boot application is loaded successfully.
 9. All inputs are reset to their initialization values. All 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 (*see page 245*).
 11. The controller file system is initialized and its resources (sockets, file handles, and so on) are deallocated.

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 (*see page 75*).

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 Flash memory.

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

Methods for Issuing the Download Application Command:

- EcoStruxure Machine Expert:
 - 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.

- FTP: Load Boot application file to the Flash 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 SD card slot. The updated file is applied at the next reboot. Refer to File Transfer with SD Card for further details.

Effects of the EcoStruxure Machine Expert 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. All 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. All fieldbus communications are stopped and then any 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. All inputs are reset to their initialization values. All 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 (*see page 245*).

For details on variables, refer to Remanent Variables (*see page 75*).

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 (*see 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	<p>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:</p> <ul style="list-style-type: none"> • 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 Flash memory is not the same as the one in RAM. • The I/O LED is red ON. 	<p>RUNNING with External Error Detected Or STOPPED with External Error Detected</p>
Application Error	<p>An application error is detected when improper programming is encountered or when a task watchdog threshold is exceeded. The controller is in STOP state. The ERR LED is red ON.</p>	HALT
System Error	<p>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.</p> <p>NOTE: There are some system errors that can be managed by runtime and are therefore treated like application errors.</p>	BOOTING → EMPTY
<p>(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 (see page 106).</p>		

NOTE: Refer to the M262 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	–	–	–
Download of application program ⁽³⁾	–	–	X

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 ([see page 68](#)).

(3) If the application is downloaded using an SD card, any existing retain-persistent variables used by the application are reinitialized. If the application is downloaded using EcoStruxure Machine Expert, however, existing retain-persistent variables maintain their values. In both cases, 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**) symbols 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 ([see page 305](#)).

Chapter 9

Controller Device Editor

Introduction

This chapter describes how to configure the controller.

What Is in This Chapter?

This chapter contains the following topics:

Topic	Page
Controller Parameters	78
Communication Settings	80
PLC Settings	81
Services	83
Ethernet Services	85
Users Rights	89

Controller Parameters

Controller Parameters

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



Tabs Description

Tab	Description	Restriction
Communication settings <i>(see page 80)</i>	Manages the connection between the PC and the controller: <ul style="list-style-type: none"> ● 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 EcoStruxure Machine Expert Programming Guide (<i>Communication Settings</i>).	–
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 <i>(see page 30)</i>	File management between the PC and the controller. Only one logic controller disk at a time can be seen through this tab. This tab displays the content of the <i>/usr</i> directory of the internal flash memory of the controller.	Online mode only
Log	View the controller log file.	Online mode only
PLC settings <i>(see page 81)</i>	Configuration of: <ul style="list-style-type: none"> ● Starting mode options ● I/O behavior in stop ● Bus cycle options 	–
Services <i>(see page 83)</i>	Date and time settings, versions.	Online mode only
Task deployment	Displays a list of I/Os and their assignments to tasks.	After compilation only

Tab	Description	Restriction
Ethernet Services <i>(see page 85)</i>	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	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 EcoStruxure Machine Expert projects and devices in online mode. For more information, refer to the EcoStruxure Machine Expert Programming Guide.	–
Access Rights <i>(see page 89)</i>	The Access Rights tab allows you to define the device access rights of device users. For more information, refer to the EcoStruxure Machine Expert Programming Guide <i>(see EcoStruxure Machine Expert, Programming Guide)</i> .	–
OPC UA Server Configuration	Displays the OPC UA Server Configuration <i>(see page 239)</i> window.	–
Diagnostic Table	Displays the data of the controller. The self-awareness data can be accessed using the syntax <code>NameOfControllerInDeviceTree.NameofParameter.</code> Example: <code>MyController.SA_NbPowerOn.</code>	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.

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 context 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 EcoStruxure Machine Expert 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	By default, set to Application because there is only one application in the controller.
PLC settings	Update IO while in stop If this option is activated (default), the values of the input and output channels get 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: <ul style="list-style-type: none"> ● Keep current values ● Set all outputs to default
	Always update variables By default, set to Enabled 1 (use bus cycle task if not used in task) and not editable.
Bus cycle options	Bus cycle task This configuration setting is the parent for all Bus cycle task parameters used in the application device 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.

Element		Description
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 (see page 55). Select with this option one of these starting modes: <ul style="list-style-type: none">● 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:

The screenshot shows the Services tab interface with the following components:

- RTC Configuration**
 - PLC Time**: A text input field and a **Read** button.
 - Local Time**:
 - Date**: A dropdown menu showing "Tuesday 6 September 2016".
 - Time**: A dropdown menu showing "16:24:27".
 - Write** button.
 - Write as UTC** checkbox.
 - Synchronize with local's date/time** button.
- Device Identification**
 - Firmware Version**: Text input field.
 - Boot Version**: Text input field.
 - Coprocessor Version**: Text input field.
- Post Configuration**
 - Parameters overwritten by the Post configuration:** Text input field.
 - Read** button.

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 M262 System Library Guide.

Element		Description
RTC Configuration	PLC Time	Displays the date and time read from the controller when the Read button is clicked, with no conversion applied. This read-only field is initially empty. If Write as UTC is selected, PLC Time is in UTC.
	Read	Reads the date and time saved on the controller and displays the values in the PLC Time field.
	Local Time	Lets you define a date and a time that are sent to the controller when the Write button is clicked. 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 current PC settings.
	Write	Writes the date and time defined in the Local time field to the logic controller. A message box informs you of the result of the command. Select the Write as UTC checkbox before running this command to write the values in UTC format.
	Synchronize with local date/time	Lets you directly send the PC settings. A message box informs you of the result of the command. Select Write as UTC before running this command to use UTC format. Use UTC time when using secure communication.
Device Identification		Displays the Firmware Version , the Boot Version , and the Coprocessor Version of the selected controller, if connected.
Post Configuration		Displays the application parameters overwritten by the Post configuration (<i>see page 245</i>).

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

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.

Element	Description	Restriction
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: <ul style="list-style-type: none"> ● 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: <ul style="list-style-type: none"> ● Activated ● Simulated ● Optional 	Can be edited for Sercos device.

Ethernet Resources

The **Ethernet Resources** tab:

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

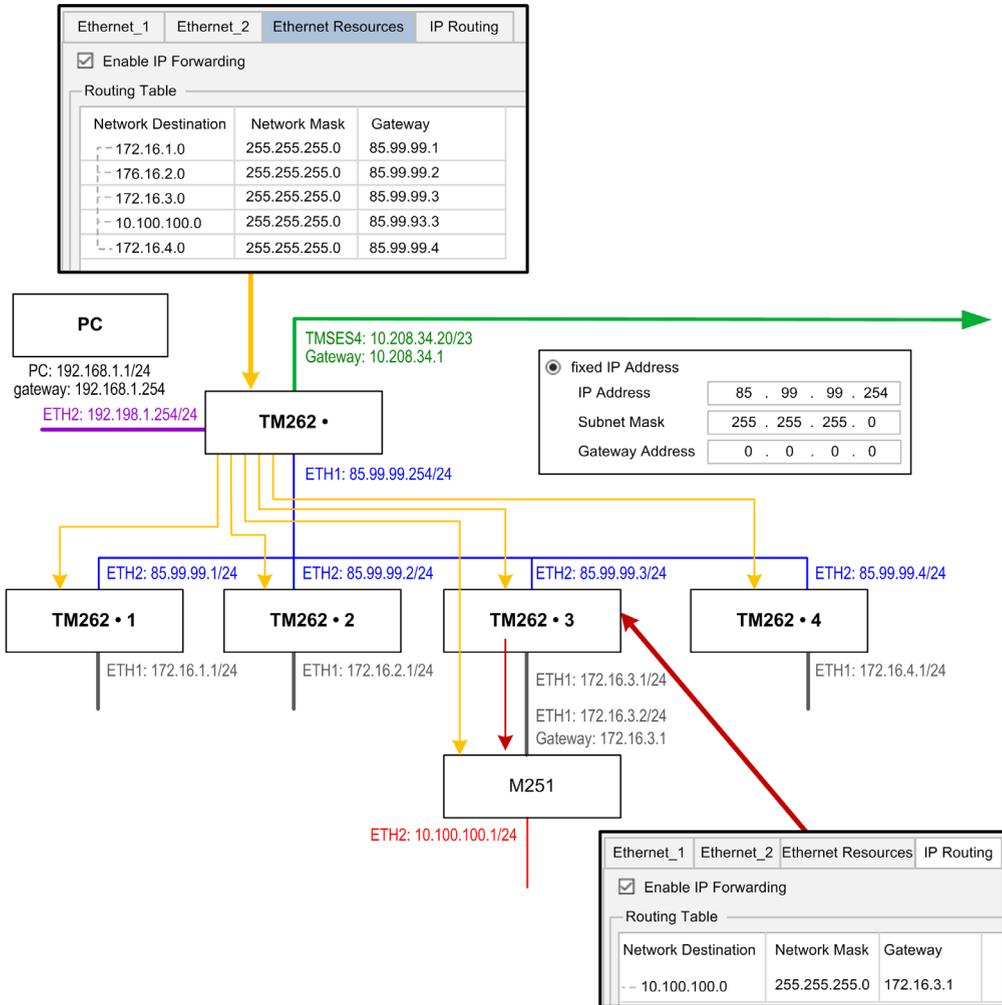
IP Routing

The **IP Routing** tab allows you to configure the IP routes in the controller.

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 (Web pages, DTM, and so on).

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.

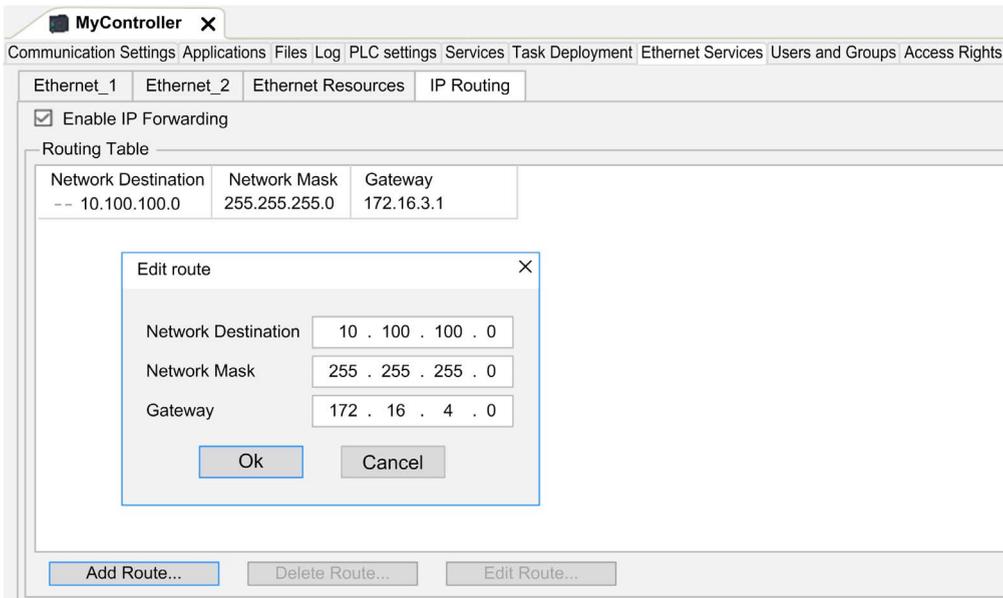
This graphic depicts an example network, in which the last two rows of devices (gray and red) need to be added in the routing table:



Use the routing tables to manage the IP forwarding. The routing tables are settled with:

Network Destination	Network Mask	Gateway
xx.xx.xx.xx	xx.xx.xx.xx	xx.xx.xx.xx

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



Users Rights

Introduction

Users and Groups and **Access Right** tabs (*see page 78*) allows to manage user accounts, user access rights groups and the associated access rights, to control the access on projects. For more informations, refer to the EcoStruxure Machine Expert Programming Guide (*see EcoStruxure Machine Expert, Programming Guide*).

Login and passwords

Login and password are set by default, must be activated and can be reset as origin.

This table describes how to log in:

Server/feature	Default values Login / Password	First user right activation Login / Password	After user right activated Login / Password
EcoStruxure Machine Expert	No login or password required	Administrator / Administrator	Administrator / configured password
Web server	Anonymous / No password required	No login or password required	Administrator / configured password
FTP	Anonymous / Anonymous	No login or password required	Administrator / configured password
OPC-UA	Anonymous / No password required	No login or password required	Administrator / configured password
Change Device Name feature	No login or password required	No login or password required	Administrator / configured password

WARNING

UNAUTHORIZED DATA AND/OR APPLICATION ACCESS

- Secure access to the FTP/Web/OPC-UA server(s) using User Rights.
- If you do not enable 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: After the user rights are activated, anonymous log in is no longer permitted by the controller.

NOTE: Only ASCII characters are supported by the controller.

Default users and groups

One user and two groups are set by default:

- User: **Administrator**
- Groups: **Administrator** and **Everyone**

Access Rights

You can give **Access Rights** to groups.

You can allow the following operations through the access rights:

- **VIEW**
- **MODIFY**
- **EXECUTE**
- **ADD_REMOVE**

Chapter 10

Embedded Inputs and Outputs Configuration

What Is in This Chapter?

This chapter contains the following sections:

Section	Topic	Page
10.1	Configuring the Fast I/Os	92
10.2	Hardware Encoder Interface	98

Section 10.1

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• logic 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 (<i>see page 21</i>).
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 (*see 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 0.1 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 don't 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: <ul style="list-style-type: none"> ● EcoStruxure Machine Expert (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) (*see page 81*).

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 (*see page 54*).

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.

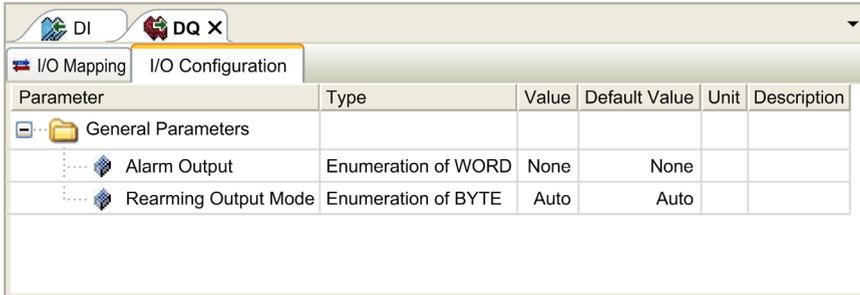
For each capture, you can configure the following parameters:

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 .
RealTimeAccess	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 related, 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 (<i>see page 96</i>).
Rearming Output Mode	Select the rearming output mode (<i>see page 97</i>).

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 tri-state 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 tri-state 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.

Section 10.2

Hardware Encoder Interface

What Is in This Section?

This section contains the following topics:

Topic	Page
Hardware Encoder Interface	99
Adding an Encoder	101
Encoder Motion Functions	104

Hardware Encoder Interface

Introduction

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

- Incremental encoder
- SSI absolute encoder

Incremental Mode Principle Description

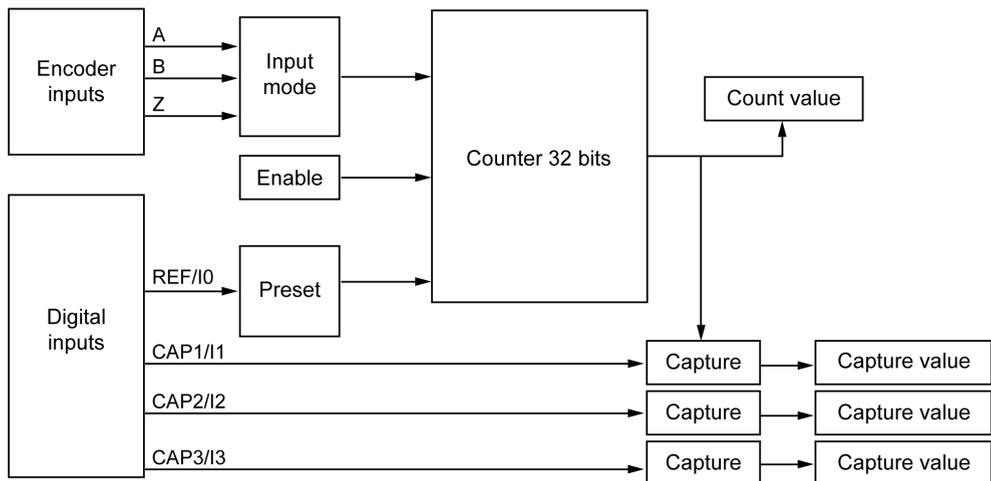
The incremental mode behaves like a standard up/down counter, using pulses and counting these pulses.

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:



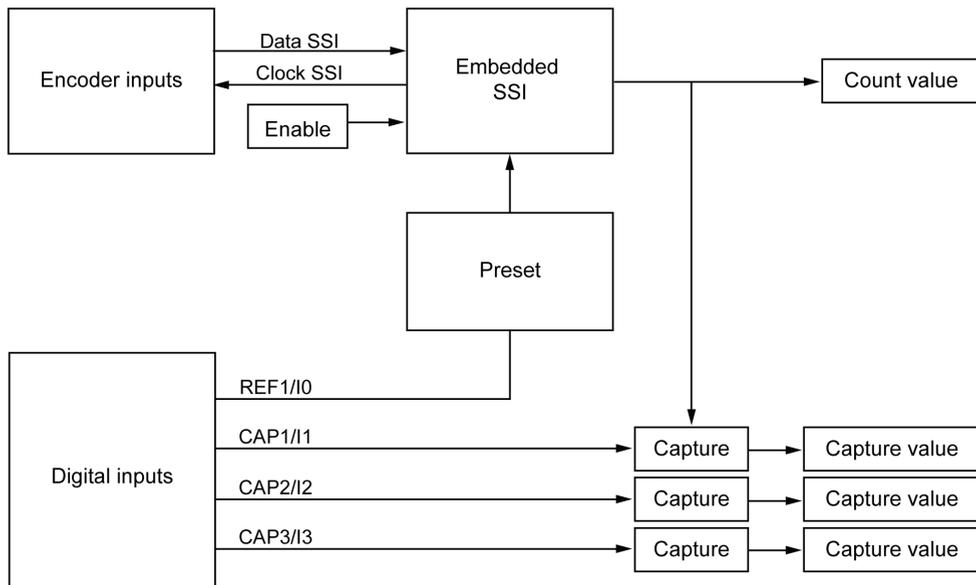
SSI Mode Principle Description

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 Drag-and-drop Method (*see EcoStruxure Machine Expert, Programming Guide*)
- Using the Contextual Menu or Plus Button (*see 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

Parameter	Type	Value	Default Value	Unit	Description
Counting Inputs					
A Input					
Filter	BYTE Enum	0.000 0.001 0.002 0.005 0.05 0.01 0.08 0.5 1 4 12	0.002	ms	Set the filtering value to reduce the bounce effect on the input
B Input					
Filter	BYTE Enum	0	0.002	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 0.05 0.01 0.08 0.5 1 4 12	0.002	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 ([see page 104](#)).

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 **FeedbackAxis** and/or **EncoderScaling** 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 FeedbackAxis and/or EncoderScaling . 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	0	IncrementResolution
PositionResolution	LREAL	1.0...1.7976931348623158e+308	0.0	PositionResolution
GearIn	UDINT	1...4,294,967,295	1	GearIn
GearOut	UDINT	1...4,294,967,295	1	GearOut

Chapter 11

Expansion Modules Configuration

Overview

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

What Is in This Chapter?

This chapter contains the following topics:

Topic	Page
TM3 I/O Configuration General Description	106
TM3 I/O Bus Configuration	110
TMS Expansion Module Configuration	111
TM3 Expansion Module Configuration	112
Optional I/O Expansion Modules	113

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 logic 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 logic controller.

I/O Expansion Bus Errors

If the logic 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 (*see page 113*)), the logic controller considers it as an I/O expansion bus error. The unsuccessful communication may be detected during the startup of the logic 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.

If an I/O expansion bus error is detected:

- The system status LED **I/O** of the logic 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

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 logic controller detects a TM3 module in bus communication error it sets the bus to a "bus off" condition whereby the TM3 expansion module outputs, the input image value and the output 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`. All other bits are set to `TM3_OK`.

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 if at least one expansion module is in error (`TM3_MODULE_R[i].i_wModuleState = TM3_BUS_ERROR`). Refer to Restarting the I/O Expansion Bus ([see page 108](#)).
- Issuing a **Reset Warm** or **Reset Cold** command with EcoStruxure Machine Expert ([see page 67](#)).

Passive I/O Expansion Bus Handling

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 logic controller attempts to continue data bus exchanges with the modules during bus communication errors. While the expansion bus error persists, the logic 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 logic controller attempts to re-establish communication. If the logic controller cannot re-establish communications within that time, all affected TM3 I/O expansion outputs are set to 0.

In either case, the logic 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 ([see 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 logic controller system, or issue a **Reset Warm** or **Reset Cold** command with EcoStruxure Machine Expert ([see page 67](#)).

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.

For more information on the actions taken upon startup of the logic controller when an I/O expansion bus error is detected, refer to Controller States Description ([see page 59](#)).

Restarting the I/O Expansion Bus

When active I/O error handling is being applied, that is, embedded and TM3 outputs set to 0 when a bus communication error is detected, the application can request a restart of the I/O expansion bus while the logic 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 logic 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 logic 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 logic 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 (*see page 113*).

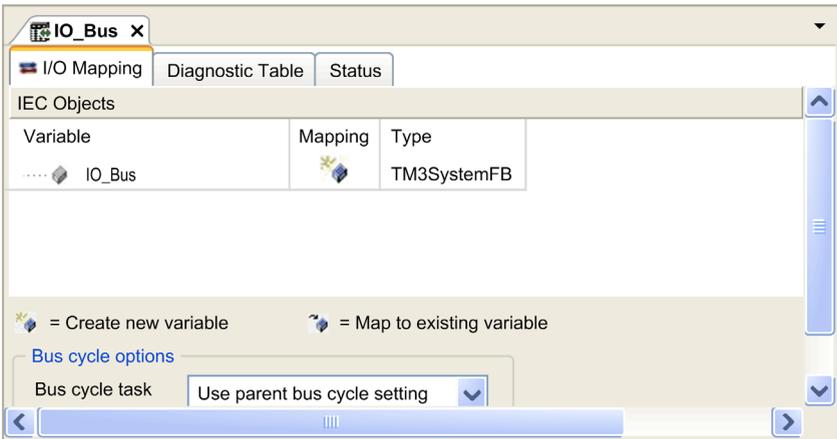
TM3 I/O Bus Configuration

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** (*see page 81*) bus cycle task.

Configuring the I/O Bus

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

Step	Description
1	<p>In the Devices tree, double-click IO_Bus. Result: The IO_Bus editor tab displays:</p> 
2	<p>Set the Bus cycle task from the list to either of the following:</p> <ul style="list-style-type: none"> ● 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 EcoStruxure Machine Expert **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 TMS 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 Drag-and-drop Method (*see EcoStruxure Machine Expert, Programming Guide*)
- Using the Contextual Menu or Plus Button (*see EcoStruxure Machine Expert, Programming Guide*)

Compatibility of the TMS Communication Expansion Modules

You can connect one TMSES4 and one TMSCO1 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 Drag-and-drop Method (*see EcoStruxure Machine Expert, Programming Guide*)
- Using the Contextual Menu or Plus Button (*see 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: <table border="1" data-bbox="326 430 1108 917"> <thead> <tr> <th>Parameter</th> <th>Type</th> <th>Value</th> <th>Default Value</th> <th>Unit</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>Optional module</td> <td>Enumeration of BYTE</td> <td>Yes</td> <td>No</td> <td></td> <td></td> </tr> <tr> <td>Outputs</td> <td></td> <td></td> <td></td> <td></td> <td></td> </tr> <tr> <td> QW0</td> <td></td> <td></td> <td></td> <td></td> <td></td> </tr> <tr> <td> Type</td> <td>Enumeration of BYTE</td> <td>Not used</td> <td>Not used</td> <td></td> <td>Range mode</td> </tr> <tr> <td> Minimum</td> <td>INT(-32768...32766)</td> <td>-32768</td> <td>-32768</td> <td></td> <td>Minimum value</td> </tr> <tr> <td> Maximum</td> <td>INT(-32767...32767)</td> <td>32767</td> <td>32767</td> <td></td> <td>Maximum value</td> </tr> <tr> <td> QW1</td> <td></td> <td></td> <td></td> <td></td> <td></td> </tr> <tr> <td> Type</td> <td>Enumeration of BYTE</td> <td>Not used</td> <td>Not used</td> <td></td> <td>Range mode</td> </tr> <tr> <td> Minimum</td> <td>INT(-32768...32766)</td> <td>-32768</td> <td>-32768</td> <td></td> <td>Minimum value</td> </tr> <tr> <td> Maximum</td> <td>INT(-32767...32767)</td> <td>32767</td> <td>32767</td> <td></td> <td>Maximum value</td> </tr> <tr> <td>Diagnostic</td> <td></td> <td></td> <td></td> <td></td> <td></td> </tr> <tr> <td> Status Enabled</td> <td>Enumeration of BYTE</td> <td>Yes</td> <td>Yes</td> <td></td> <td></td> </tr> </tbody> </table> <p>Modifiable by programming  = Yes  = No</p>	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		
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																																																																												

Shared 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 groups the module references sharing the same internal ID code:

Modules sharing the same internal ID code
TM3DI16K, TM3DI16, TM3DI16G
TM3DQ16R, TM3DQ16RG, TM3DQ16T, TM3DQ16TG, TM3DQ16TK, TM3DQ16U, TM3DQ16UG, TM3DQ16UK
TM3DQ32TK, TM3DQ32UK
TM3DI8, TM3DI8G, TM3DI8A

Modules sharing the same internal ID code
TM3DQ8R, TM3DQ8RG, TM3DQ8T, TM3DQ8TG, TM3DQ8U, TM3DQ8UG
TM3DM8R, TM3DM8RG
TM3DM24R, TM3DM24RG
TM3SAK6R, TM3SAK6RG
TM3SAF5R, TM3SAF5RG
TM3SAC5R, TM3SAC5RG
TM3SAFL5R, TM3SAFL5RG
TM3AI2H, TM3AI2HG
TM3AI4, TM3AI4G
TM3AI8, TM3AI8G
TM3AQ2, TM3AQ2G
TM3AQ4, TM3AQ4G
TM3AM6, TM3AM6G
TM3TM3, TM3TM3G
TM3TI4, TM3TI4G
TM3TI4D, TM3TI4DG
TM3TI8T, TM3TI8TG
TM3XFHSC202, TM3XFHSC202G
TM3XHSC202, TM3XHSC202G

Chapter 12

Ethernet Configuration

Introduction

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

What Is in This Chapter?

This chapter contains the following sections:

Section	Topic	Page
12.1	Ethernet Services	118
12.2	Firewall Configuration	189

Section 12.1

Ethernet Services

What Is in This Section?

This section contains the following topics:

Topic	Page
Presentation	119
IP Address Configuration	121
Modbus TCP Client/Server	127
Web Server	129
Symbol Configuration Editor	151
FTP Server	157
SNMP	159
Controller as a Target Device on EtherNet/IP	160
Controller as a Slave Device on Modbus TCP	184

Presentation

Ethernet Services

The controller supports the following services:

- Modbus TCP Server (*see page 127*)
- Modbus TCP Client (*see page 127*)
- DHCP Server (*see page 209*)
- Web Server (*see page 129*)
- FTP Server (*see page 157*)
- SNMP (*see page 159*)
- Controller as Target Device On EtherNet/IP (*see page 160*)
- Controller as Slave Device On Modbus TCP (*see page 184*)
- IEC VAR ACCESS (*see page 120*)
- Web Visualization (*see page 125*)
- OPCUA Server (*see page 237*)

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 an USB cable and to access the same services as with an Ethernet connection (*see page 251*).

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 EcoStruxure Machine Expert online help.

Connections

This table shows the maximum number of connections:

Connection Type	Maximum
Modbus Server	8 connections
Modbus Client	8 connections
Modbus TCP I/O Scanner	64 channels
EtherNet/IP Scanner	64 connections
FTP Server	4 connections
Web Server	10 concurrent users
Machine Expert Protocol (EcoStruxure Machine Expert software, trace, Web visualization, HMI devices)	8

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. All server connections stay open as long as the controller stays in operational states (`RUNNING`, `STOPPED`, `HALT`).
4. All server connections are closed when leaving or entering 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. The available variables are selected in the **Symbol Configuration**.

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 (*see page 245*). 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 (*see page 80*) tab in EcoStruxure Machine Expert
- **changeIPAddress** function block (*see page 295*)

NOTE: If the attempted addressing method is unsuccessful, the link uses a default IP address (*see page 124*) 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

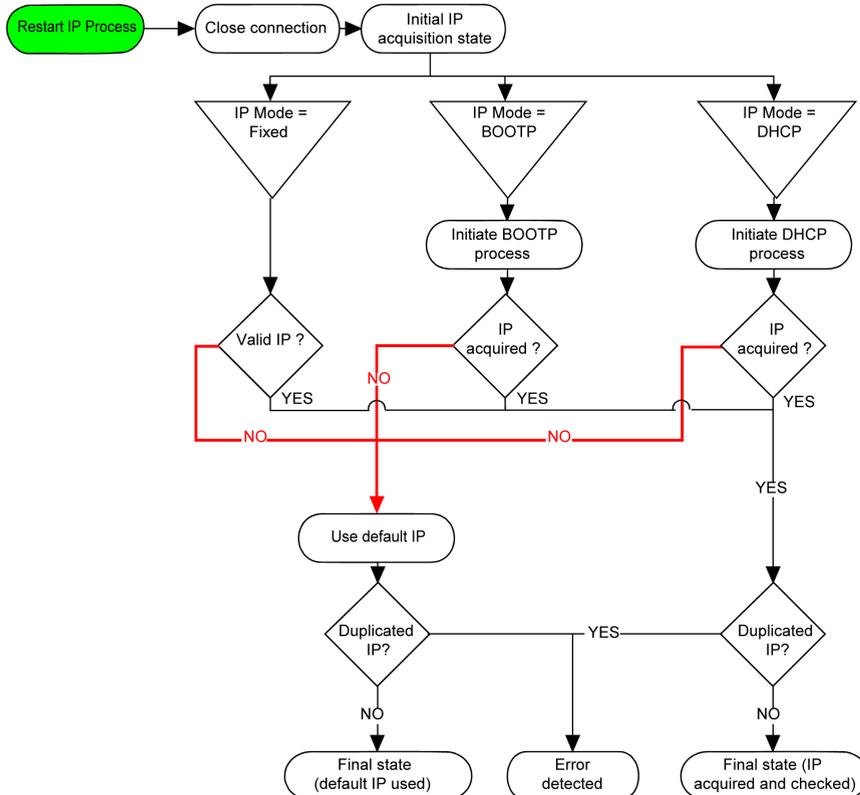
- Verify that there is only one master controller configured on the network or remote link.
- Verify that all devices have unique addresses.
- Obtain your IP address from your system administrator.
- Confirm that the IP address of the device is unique before placing the system into service.
- Do not 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**

The screenshot displays the Ethernet Configuration dialog box, which is divided into several sections:

- Configured Parameters:**
 - Network Name: my_Device
 - IP Address by DHCP:
 - IP Address by BOOTP:
 - fixed IP Address:
 - IP Address: 0 . 0 . 0 . 0
 - Subnet Mask: 0 . 0 . 0 . 0
 - Gateway Address: 0 . 0 . 0 . 0
 - Ethernet Protocol: Ethernet 2
 - Transfer Rate: Auto
- Current Settings:**
 - Interface Name: gem0
 - Network Name: my_Device
 - IP Address by DHCP:
 - IP Address by BOOTP:
 - fixed IP Address:
 - IP Address: 85 . 18 . 1 . 12
 - Subnet Mask: 255 . 0 . 0 . 0
 - Gateway Address: 0 . 0 . 0 . 0
 - Ethernet Protocol: Ethernet 2
 - Transfer Rate: Auto
- Security Parameters (Left):**
 - Protocol inactive: (Empty list)
 - Protocol active:
 - Discovery protocol
 - FTP Server
 - Machine Expert protocol
 - Modbus Server
 - Remote connection (Fast TCP)
 - Secured Web Server (HTTPS)
 - SNMP protocol
 - Web Server (HTTP)
 - Web Visualisation protocol
- Security Parameters (Right):**
 - Protocol inactive: (Empty list)
 - Protocol active:
 - Discovery protocol
 - FTP Server
 - Machine Expert protocol
 - Modbus Server
 - Remote connection (Fast TCP)
 - Secured Web Server (HTTPS)
 - SNMP protocol
 - Web Server (HTTP)
 - Web Visualisation protocol
- Ring topology options:**
 - Ring topology: No ring
- Adapter Status:**
 - MAC Address: 00:80:F4:4E:00:79
 - Network Status: Data Exchanges

Note: If you are in online mode, you see the two windows. You cannot edit them. 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.

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.
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.

Configured Parameters	Description
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.x. for Ethernet_1
- 10.11.x.y. for Ethernet_2

When TM262* is not configured, TMS4 boots and automatically gets its default IP address:

- 10.12.x.y for the first module

x and y 5th and 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 two IP addresses must not be in the same IP network.

The MAC address of the port can be retrieved on the label placed on the front side of the controller.

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: If the MAC address is 00.80.F4.01.80.F2, the default IP address is 10.10.128.242.

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 current network.

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

The gateway address must be defined on Ethernet_1 interface. The traffic to unknown networks is sent through this interface.

Security Parameters

This table describes the different security parameters:

Security Parameters	Description
Discovery protocol	This parameter deactivates Discovery protocol. When deactivated, Discovery requests are ignored.
FTP Server	This parameter deactivates the FTP Server of the controller. When deactivated, FTP requests are ignored.
Machine Expert protocol	This parameter deactivates the Machine Expert protocol on Ethernet interfaces. When deactivated, every Machine Expert request from every device is rejected. Therefore, no connection is possible on Ethernet from a PC with EcoStruxure Machine Expert, from an HMI target that wants to exchange variables with this controller, from an OPC server, or from Controller Assistant.
Modbus Server	This parameter deactivates the Modbus Server of the controller. When deactivated, every Modbus request to the controller is ignored.
Remote connection	This parameter deactivates the remote connection. When deactivated, Fast TCP requests are ignored.
Secured Web Server	This parameter deactivates the Secured Web Server of the controller. When deactivated, HTTPS requests to the controller Secured Web Server are ignored.
SNMP protocol	This parameter deactivates the SNMP server of the controller. When deactivated, SNMP requests are ignored.
Web Server	This parameter deactivates the Web Server of the controller. When deactivated, HTTP requests to the controller Web Server are ignored.
WebVisualisation protocol	This parameter deactivates the Web visualization pages of the controller. When deactivated, HTTP requests to the logic controller WebVisualisation protocol are ignored.

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 RSTP protocol.

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

For further information, refer to the Industrial Ethernet Considerations Modicon M262 Logic/Motion Controller, Hardware Guide.

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 to 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 EcoStruxure Machine Expert, 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

Web Server

Introduction

The Web server is a tool that allows you to remotely monitor a controller and its application, perform various maintenance activities including modifications to data and configuration parameters, and change the state of the controller.

As standard equipment, 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 Windows Web browser or mobile device. No configuration or programming is required.

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

- Google Chrome (version 65.0 or greater)
- Mozilla Firefox (version 54 or greater)
- Microsoft Internet Explorer (version 11 or greater)

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

- iOS Safari
- Android Chrome

You can access the Web server with HTTP (non secured connections) or HTTPS (secure connections). Some actions (**User Management**) are only possible in secure mode.

The Web server is limited to 10 concurrent users (*see page 120*).

The Web server has full 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 (*see page 123*).

If there are security concerns over these functions, you must, at a minimum, assign a secure password to the Web Server or disable the Web server to prevent unauthorized access to the application. Care must be taken to ensure that the immediate physical environment of the machine and process is in a state that will not present safety risks to people or property before exercising control remotely.

WARNING

UNINTENDED EQUIPMENT OPERATION

- 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.

NOTE: The Web server must only be used by authorized and qualified personnel. A qualified person is one who has the skills and knowledge related to the construction and operation of the machine and the process controlled by the application and its installation, and has received safety training to recognize and avoid the hazards involved. No responsibility is assumed by Schneider Electric for any consequences arising out of the use of this feature.

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** Tab Description (*see page 78*).

If User Rights are not enabled in the controller, you are prompted for a user name and password unique to the Web server. The default user name is Anonymous and no password is required.

NOTE: You cannot modify the default user name and password. To secure the Web server functions, you must do so with **Users and Groups**.

⚠ WARNING
<p>UNAUTHORIZED DATA ACCESS</p> <ul style="list-style-type: none"> • Secure access to the FTP/Web server using User Rights. • If you do not enable User Rights, disable the FTP/Web server to prevent any unwanted or unauthorized access to data in your application. <p>Failure to follow these instructions can result in death, serious injury, or equipment damage.</p>

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 or USB key (depending on the support of your particular controller) to update the controller firmware. In addition, you may clear the User Rights in the controller by running a script (for more information, refer to EcoStruxure Machine Expert Programming Guide). 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:

The image shows a browser window with the address bar containing 'http://85.15.1.51/login.htm'. Below the browser, there is a 'User:' label followed by an input field, a 'Password:' label followed by an input field, and a 'Login' button.

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 environment or your machines 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.
- Monitor activities within your systems.
- Prevent subject devices from direct access or direct link by unauthorized parties or unauthenticated actions.
- 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.

Home Page Access

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

The Web Server contains the following pages:

Menu	Page	Description
Home	Home (<i>see page 131</i>)	Home page of the controller Web server page. Provides access to the tabs: <ul style="list-style-type: none"> ● Monitoring ● Diagnostics ● Maintenance ● Machine Assistant

Home page menu descriptions:

Menu	Submenu	Description
Monitoring	Data Parameters (<i>see page 136</i>)	Lets you display and modify controller variables.
	IO Viewer (<i>see page 137</i>)	Shows the module with module I/O values.
	Oscilloscope (<i>see page 137</i>)	Displays 2 variables in the form of a recorder-type time chart.
Diagnostics	Controller (<i>see page 138</i>)	Displays controller status.
	Ethernet (<i>see page 139</i>)	Displays Ethernet diagnostic.
	TM3 Expansion (<i>see page 140</i>)	Displays expansion module status.
	TMS Expansion (<i>see page 141</i>)	Displays expansion module status.
	Scanner Status (<i>see page 142</i>)	Displays serial line status.
	EtherNet/IP Status (<i>see page 143</i>)	Displays Ethernet status.
	Sercos	Displays Sercos diagnostic.

Menu	Submenu	Description
Maintenance	Post configuration <i>(see page 144)</i>	Lets you access the post configuration file saved on the controller.
	User Management <i>(see page 144)</i>	Lets you change actual user password and customize login message. Possible in secure mode (HTTPS) only.
	Firewall <i>(see page 145)</i>	Lets you modify the firewall configuration.
	System Log Files <i>(see page 145)</i>	Lets you access log files generated by the controller.
	Message Logger <i>(see page 146)</i>	Lets you access controller messages.
	Run/Stop Controller <i>(see page 146)</i>	Lets you send Run and Stop commands to the controller.
	SelfAwareness <i>(see page 147)</i>	Lets you access memory usage, temperature and slave devices information.
	Certificates <i>(see page 148)</i>	Lets you customize certificates owned by an M262 controller.
	Date / Time <i>(see page 149)</i>	Lets you set the date and time.
Machine Assistant	List View	Displays the configuration in list view.
	Graphic view	Displays the configuration in graphic view.
	Scan <i>(see page 287)</i>	Lets you scan the devices configured.
	Clear <i>(see page 287)</i>	Lets you clear the scan.
	load .semtd file <i>(see page 291)</i>	Lets you upload a .semtd file after scan.
	Export scan results <i>(see page 291)</i>	Lets you export the scan results in your local SD Card.
	Log out	Lets you log out.

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 will not present safety 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.

NOTE: The Web server must only be used by authorized and qualified personnel. A qualified person is one who has the skills and knowledge related to the construction and operation of the machine and the process controlled by the application and its installation, and has received safety training to recognize and avoid the hazards involved. No responsibility is assumed by Schneider Electric for any consequences arising out of the use of this feature.

Monitoring: Data Parameters

Monitoring Web Server Variables

To monitor Web server variables, you must select the variables in the Symbol Configuration Editor (*see page 151*).

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: <ul style="list-style-type: none"> ● Gray button: refreshing disabled ● Orange button: refreshing enabled <p>NOTE: Without enabling Refresh, when a variable's value is changed in the table the modification is directly sent to the controller.</p>
Load	Loads saved lists from the controller internal Flash to the Web server page
Save	Saves the selected list description in the controller (<i>/usr/web</i> 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 (*see page 36*)).

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 (*see page 151*).

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

IO Viewer

 refresh	Period (ms)	<input type="text" value="1000"/>		
Mapping	Address	Type	Format	Value

Element	Description
Refresh	Enables I/O refreshing: <ul style="list-style-type: none"> ● 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:

Oscilloscope

 reset	 refresh	Item0: <input type="text"/> 	min: <input type="text"/>	max: <input type="text"/>	Period (ms) <input type="text" value="1000"/>
 load	 save	Item1: <input type="text"/> 	min: <input type="text"/>	max: <input type="text"/>	

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

Diagnostic: Controller Submenu

The **Controller** submenu displays information about the current status of the controller:

Controller

Reset Statistics

Identification	
VendorID	0x101a
Vendor name	Schneider Electric
ProductID	0x811
Product reference	TM262-25
Serial Number	130
Node name	TM262-25 Ivan

Version	
Firmware	4.0.0.36
Boot	0.0.0.36
Hardware	0x20202
Coprocessor	0x80280100

Extension bus	
TM3 Bus status	0b0000000000000011 : OK

Status	
Application status	Running (2)
Boot project status	Same boot project (65535)
IO Status 1	Power supply fault (4)
IO Status 2	Ok (FFFF)
Application signature	B9394DC8
Application signature	0
Application signature	0
Application signature	0
Last stop cause	Powerfall (15)
Last application error	Software watchdog of IEC-task expired (16)
System Fault 1	TMS module fault
System Fault 2	No error
Last stop time	Fri, 12 Oct 2018 10:45:31
Last power-off time	Fri, 12 Oct 2018 10:51:20
Events counter	0
SdCard	None (0)
USB Programming port	Not connected (0)

File	
File system free handle	1974
File system total bytes	1073741824 (1024 MB)
File system free bytes	106259744 (1013 MB)

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 viewer** submenu shows the expansion module 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 viewer** submenu shows the expansion module 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: 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:

Modbus TCP I/O Scanner

Scanner Status

 Idle

Connection Statistics

Total transmissions sent: **0**

Number of Configured Connections: **0**

Scanned Device Statuses

No Scanned Devices Reported

- Not Configured Scanned Fault

For more information, refer to EcoStruxure Machine Expert 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:

EIP I/O Scanner

Scanner Status

 Idle

Connection Statistics

Total transmissions sent: **0**

Number of Configured Connections: **0**

Scanned Device Statuses

No Scanned Devices Reported

Not Configured
 Scanned
 Fault

For more information, refer to EcoStruxure Machine Expert EtherNet/IP User guide.

Maintenance Page

The Maintenance page provides access to the `/usr` folders of the controller flash memory (*see page 30*) and various informations that are useful for device maintenance purposes.

Step	Action
1	Click Load .
2	Modify the parameters (<i>see page 248</i>).
3	Click Save . NOTE: The new parameters will be considered at next Post Configuration file reading (<i>see page 246</i>).

Maintenance: Post Conf Submenu

The **Post Conf** submenu allows you to update the post configuration (*see page 245*) file saved on the controller:

Post Conf

No Post Conf available

Maintenance: User Management Submenu

The **User Management** submenu allows you to change the password and customize a message which will be displayed at login:

User Management

<p>Change password (of current user)</p> <p>Current password</p> <input type="text"/> New password	<p>System use notification</p> <p>Current:</p> <div style="border: 1px solid gray; background-color: #e0e0e0; padding: 5px; min-height: 100px;"><void></div>
<input type="text"/> Confirm new password	<p>New:</p> <div style="border: 1px solid gray; background-color: #e0e0e0; padding: 5px; min-height: 100px;"></div>
<p><input type="button" value="Change password"/></p> <p>Caution: Change password is only allowed with a secure protocol.</p>	<p><input type="button" value="Save"/> <input type="button" value="Disable"/></p> <p><input type="button" value="Default"/></p> <p>No system use notification is displayed at login.</p>
<p>Users accounts</p> <p><input type="button" value="Clear all users accounts"/></p>	

NOTE: The option to clear all user accounts is only active if the current user has administrative privileges.

Maintenance: Firewall Submenu

The **Firewall** submenu allows you to modify the default firewall configuration file (*see page 189*):

Firewall

No Firewall Conf available

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

Maintenance: Message Logger Submenu

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

Message Logger

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:

Run/Stop Controller

Identification	
Product reference	TM262-25
Serial Number	130
Node name	TM262-25 Ivan
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	
Application status	Running (2)
Boot project status	Same boot project (65535)
Last stop cause	Powerfail (15)
Last application error	Software watchdog of IEC-task expired (16)
Last stop time	Fri, 12 Oct 2018 10:45:31
Last power-off time	Fri, 12 Oct 2018 10:51:20

Maintenance: SelfAwareness Submenu

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

SelfAwareness

<div style="background-color: #008000; color: white; padding: 2px; text-align: center; font-weight: bold;">Power On</div> <p>Time (seconds) <input type="text" value="1207875"/></p> <p>Count <input type="text" value="60"/></p>	<div style="background-color: #008000; color: white; padding: 2px; text-align: center; font-weight: bold;">PLC Internal Temperatures</div> <p>Power Supply: Current (°C) <input type="text" value="43"/></p> <p>Max (°C) <input type="text" value="43"/></p> <p><input type="button" value="Reset"/></p> <p>Ambient: Current (°C) <input type="text" value="35"/></p> <p>Max (°C) <input type="text" value="36"/></p> <p><input type="button" value="Reset"/></p> <p>TMS Interface: Current (°C) <input type="text" value="44"/></p> <p>Max (°C) <input type="text" value="45"/></p> <p><input type="button" value="Reset"/></p> <p>CPU Board: Current (°C) <input type="text" value="44"/></p> <p>Max (°C) <input type="text" value="45"/></p> <p><input type="button" value="Reset"/></p> <p>CPU Internal: Current (°C) <input type="text" value="56"/></p> <p>Max (°C) <input type="text" value="57"/></p> <p><input type="button" value="Reset"/></p>
<div style="background-color: #008000; color: white; padding: 2px; text-align: center; font-weight: bold;">Power Faults</div> <p>Bad Voltage <input type="text" value="0"/></p> <p>TMS Over-current <input type="text" value="0"/></p>	
<div style="background-color: #008000; color: white; padding: 2px; text-align: center; font-weight: bold;">Memory Usage</div> <p>Current (bytes) <input type="text" value="103120748"/></p> <p>Max (bytes) <input type="text" value="103773108"/></p> <p><input type="button" value="Reset"/></p>	

Devices viewer |<< << < 1 - 6 / 6 > >> >>|

<div style="background-color: #008000; color: white; padding: 2px; text-align: center; font-weight: bold;">Device 1</div> <p>Vendor ID <input type="text"/></p> <p>Product Name <input type="text"/></p> <p>Serial Number <input type="text"/></p> <p>Firmware Version <input type="text"/></p> <p>Product Code <input type="text"/></p>	<div style="background-color: #008000; color: white; padding: 2px; text-align: center; font-weight: bold;">Device 2</div> <p>Vendor ID <input type="text"/></p> <p>Product Name <input type="text"/></p> <p>Serial Number <input type="text"/></p> <p>Firmware Version <input type="text"/></p> <p>Product Code <input type="text"/></p>
--	--

Maintenance: Certificates Submenu

The **Certificates** submenu allows the customization of certificates owned by an M262 controller, and manual setting of client certificates to trust:

Certificates

Own Certificate		
	Current values	New values (updated after PLC reboot)
Country:	FR	--
State:		----
Locality:	Carros	----
Common name:	TM262-25	----
Organization:	Schneider-Electric	----
Organization unit:	MachineSolutions	----
<input type="button" value="Save"/>		

Client Certificates	
Rejected	Trusted
<input type="button" value=">>"/>	<input type="button" value="<<"/>

Maintenance: Date / Time Submenu

The **Date / Time** submenu displays the current date and time and allows setting of the date and time manually:

Date / Time

Read

Local Time	Fri Oct 12 2018 15:00:29 GMT+0200 (heure d'été d'Europe centrale)
PLC Time	Fri, 12 Oct 2018 15:00:29

Update PLC

Date (yyyy-mm-dd)


Time (hh:mm:ss)

Relative correction (+/- 0 .. 9999s)

Caution: Modifying PLC time can cause web session(s) to expire immediately.

Machine Assistant

The **Machine Assistant** submenu allows you to configure the controller:



Web site version: 1.0.0.0 | © 2000-2018 Schneider Electric. All Rights Reserved.

For more information on buttons, refer to Industrial Plug and Work. ([see page 283](#))

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 POU's 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 each model

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 <code>{attribute 'symbol' := 'read'}</code> 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 (<i>see EcoStruxure Machine Expert, Programming Guide</i>).
Build button	The Build button allows you to build the project. The build run refreshes the variables view in the Symbol Configuration editor.

Element	Description	
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 EcoStruxure Machine Expert Menu Commands Online Help (<i>see EcoStruxure Machine Expert, Menu Commands, Online Help</i>). 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 <i>Additional Information on the Option Configure synchronisation with IEC tasks...</i> (<i>see EcoStruxure Machine Expert, Programming Guide</i>).
	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 <code>STRUCTS</code> that use the attributes <code>pack_mode</code> or <code>relative_offset</code> . 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 by default selected for projects with EcoStruxure Machine Expert. 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: <ul style="list-style-type: none"> ●  : 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: <code>myVar : MY_INT,,</code> where <code>MY_INT</code> is an alias declared as: <code>TYPE MY_INT : INT; END_TYPE.</code> In this case, the Type column shows <code>MY_INT : INT</code> .
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} (see *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.

Element	Description
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 synchronisation 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 synchronisation 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 synchronisation 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 synchronisation 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.

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: 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 environment or your machines 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.
- Monitor activities within your systems.
- Prevent subject devices from direct access or direct link by unauthorized parties or unauthenticated actions.
- 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.

NOTE: Make use of the security-related commands (*see EcoStruxure Machine Expert, 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.

The FTP server is available even if the controller is empty (no user application and no User Rights are enabled).

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 and Groups** Tab Description (*see page 78*).

If User Rights are not enabled in the controller, you are prompted for a user name and password unique to the FTP server. The default user name is Anonymous and the default password is also Anonymous.

NOTE: You cannot modify the default user name and password. To secure the FTP server functions, you must do so with **Users and Groups**.

WARNING

UNAUTHORIZED DATA ACCESS

- Secure access to the FTP/Web server using User Rights.
- If you do not enable 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 or USB key (depending on the support of your particular controller) to update the controller firmware. In addition, you may clear the User Rights in the controller by running a script (for more information, refer to EcoStruxure Machine Expert Programming Guide). This effectively removes the existing application from the controller memory, but restores the ability to access the controller.

Files Access

See File Organization (*see page 30*).

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.schneider-electric.com.

SNMP Client

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

For further information about EtherNet/IP, refer to the www.odva.org website.

EtherNet/IP Target Configuration

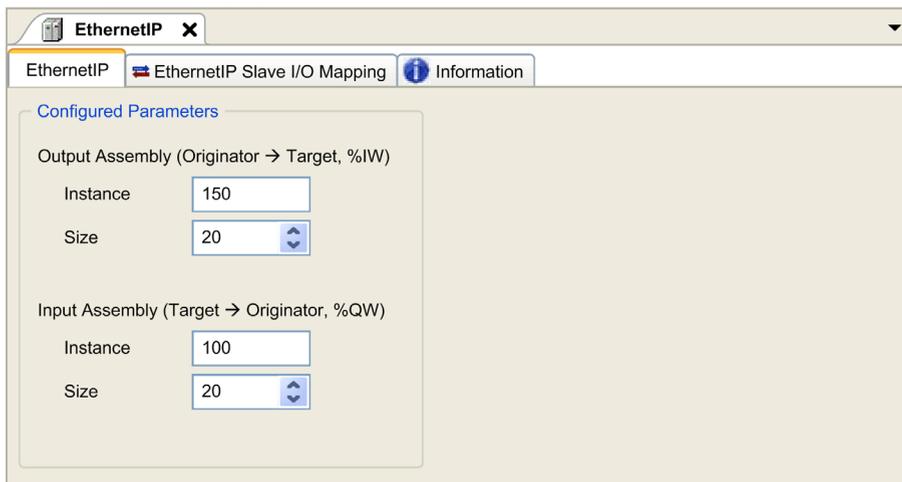
To configure your M262 Logic/Motion Controller as an EtherNet/IP target device, you must:

Step	Action
1	In the Hardware Catalog , select Devices & Modules → Communication → Ethernet IP → EthernetIP .
2	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: <ul style="list-style-type: none"> • Using the Drag-and-drop Method (<i>see EcoStruxure Machine Expert, Programming Guide</i>) • Using the Contextual Menu or Plus Button (<i>see EcoStruxure Machine Expert, Programming Guide</i>)

EtherNet/IP Parameters Configuration

To configure the EtherNet/IP parameters, double-click an Ethernet port → **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	EcoStruxure Machine Expert Default Value
Output Assembly	Instance	150...189	150
	Size	2...120	20
Input Assembly	Instance	100...149	100
	Size	2...120	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 context 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 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.

Variable	Mapping	Channel	Address	Type	Default Value	Unit	Description
Inputs							
iwEthernetP_IW0		IW0	%IW1	WORD	0		
iwEthernetP_IW1		IW1	%IW2	WORD	0		
iwEthernetP_IW2		IW2	%IW3	WORD	0		
iwEthernetP_IW3		IW3	%IW4	WORD	0		
iwEthernetP_IW4		IW4	%IW5	WORD	0		
iwEthernetP_IW5		IW5	%IW6	WORD	0		
iwEthernetP_IW6		IW6	%IW7	WORD	0		
iwEthernetP_IW7		IW7	%IW8	WORD	0		
iwEthernetP_IW8		IW8	%IW9	WORD	0		
iwEthernetP_IW9		IW9	%IW10	WORD	0		
iwEthernetP_IW10		IW10	%IW11	WORD	0		
iwEthernetP_IW11		IW11	%IW12	WORD	0		
iwEthernetP_IW12		IW12	%IW13	WORD	0		
iwEthernetP_IW13		IW13	%IW14	WORD	0		
iwEthernetP_IW14		IW14	%IW15	WORD	0		
iwEthernetP_IW15		IW15	%IW16	WORD	0		
iwEthernetP_IW16		IW16	%IW17	WORD	0		
iwEthernetP_IW17		IW17	%IW18	WORD	0		
iwEthernetP_IW18		IW18	%IW19	WORD	0		
iwEthernetP_IW19		IW19	%IW20	WORD	0		
Outputs							
qwEthernetP_QW0		QW0	%QW1	WORD	0		
qwEthernetP_QW1		QW1	%QW2	WORD	0		
qwEthernetP_QW2		QW2	%QW3	WORD	0		
qwEthernetP_QW3		QW3	%QW4	WORD	0		

Reset mapping Always update variables: Enabled 1 (use bus cycle task if not used in any task)

= Create new variable = Map to existing variable

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 (*see page 160*).

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.

Profile

The controller supports the following objects:

Object class	Class ID (hex)	Cat.	Number of Instances	Effect on Interface Behavior
Identity Object (<i>see page 164</i>)	01	1	1	Supports the reset service
Message Router Object (<i>see page 167</i>)	02	1	1	Explicit message connection
Assembly Object (<i>see page 168</i>)	04	2	2	Defines I/O data format
Connection Manager Object (<i>see page 170</i>)	06	–	1	–
TCP/IP Interface Object (<i>see page 173</i>)	F5	1	1	TCP/IP configuration
Ethernet Link Object (<i>see page 175</i>)	F6	1	1	Counter and status information

Object class	Class ID (hex)	Cat.	Number of Instances	Effect on Interface Behavior
Interface Diagnostic Object (see page 176)	350	1	1	–
IOScanner Diagnostic Object (see page 179)	351	1	1	–
Connection Diagnostic Object (see page 180)	352	1	1	–
Explicit Connection Diagnostic Object (see page 182)	353	1	1	–
Explicit Connections Diagnostic List Object (see page 183)	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 Instances	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

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⁾

Attribute ID (hex)	Access	Name	Data Type	Value (hex)	Details
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	<ul style="list-style-type: none"> • 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.

Bit	Name	Description
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 Instances	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: <ul style="list-style-type: none"> ● 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 Instances	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.

Attribute ID (hex)	Access	Name	Data Type	Value (hex)	Details
5	Get	Optional Service List	UINT	Not supported	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 a EtherNet/IP Scanner

When a 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 Instances	UINT	01	The largest instance number.
3	Get	Number of Instances	UINT	01	The number of object instances.

Attribute ID (hex)	Access	Name	Data Type	Value (hex)	Details
4	Get	Optional Instance Attribute List	Struct of: UINT UINT []	–	<p>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.</p> <p>Following optional attributes include:</p> <ul style="list-style-type: none"> ● 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 Instances	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	<ul style="list-style-type: none"> ● 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	<ul style="list-style-type: none"> ● 0: BOOTP Client ● 1: DNS Client ● 2: DHCP Client ● 5: Configured in EcoStruxure Machine Expert <p>All other bits are reserved and set to 0.</p>

Attribute ID (hex)	Access	Name	Data Type	Value	Description
3	Get	Configuration	DWORD	Bit level	<ul style="list-style-type: none"> ● 0: The interface configuration is valid. ● 1: The interface configuration is obtained with BOOTP. ● 2: The interface configuration is obtained with DHCP. ● 3: reserved ● 4: DNS Enable <p>All other bits are reserved and set to 0.</p>
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 Instances	UINT	3	The largest instance number.
3	Get	Number of Instances	UINT	3	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	<ul style="list-style-type: none"> ● 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): <ul style="list-style-type: none"> ● Bit 0: EtherNet/IP ● Bit 1: Modbus TCP ● Bit 2: Modbus Serial ● Bits 3...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.

Attribute ID (hex)	Access	Name	Data Type	Details
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: <ul style="list-style-type: none"> ● 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.

Attribute ID (hex)	Access	Name	Data Type	Details
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 μ s.
		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): <ul style="list-style-type: none"> ● 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 (*see page 185*).

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.odva.org website.

Adding a Modbus TCP Slave Device

To add a Modbus TCP slave device, select **Modbus TCP Slave Device** in the **Hardware Catalog**. Drag and drop it to the **Devices tree** on one of the highlighted nodes.

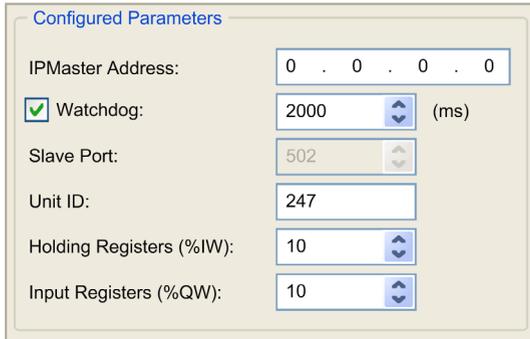
For more information on adding a device to your project, refer to:

- Using the Drag-and-drop Method (*see EcoStruxure Machine Expert, Programming Guide*)
- Using the Contextual Menu or Plus Button (*see EcoStruxure Machine Expert, Programming Guide*)

Modbus TCP Configuration

To configure the Modbus TCP slave device, double-click **Ethernet_2** → **ModbusTCP_Slave_Device** in the **Devices tree**.

This dialog box appears:



The dialog box shows the following configuration parameters:

- IPMaster Address: 0 . 0 . 0 . 0
- Watchdog: 2000 (ms)
- Slave Port: 502
- Unit ID: 247
- Holding Registers (%IW): 10
- Input Registers (%QW): 10

Element	Description
Configured Parameters	
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.
Slave Port	Modbus communication port (502)
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...40) (each register is 2 bytes)
Input Registers (%QW)	Number of %QW registers to be used in the exchange (2...40) (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 I/O Scanner 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:

Variable	Mapping	Channel	Address	Type	Default Value	Unit	Description
		Inputs	%IW2	ARRAY [0..9] OF WORD			Modbus Holding Registers
		Outputs	%QW2	ARRAY [0..9] OF WORD			Modbus Input Registers

Reset mapping Always update variables: Enabled 1 (use bus cycle task if not used in any task)

= Create new variable = Map to existing variable

Bus cycle options
Bus cycle task Use parent bus cycle setting

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).

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** (the default),
- **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 (*see 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.

Section 12.2

Firewall Configuration

Introduction

This section describes how to configure the firewall of the Modicon M262 Logic/Motion Controller.

What Is in This Section?

This section contains the following topics:

Topic	Page
Introduction	190
Dynamic Changes Procedure	192
Firewall Behavior	193
Firewall Script Commands	195

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 environment or your machines 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.
- Monitor activities within your systems.
- Prevent subject devices from direct access or direct link by unauthorized parties or unauthenticated actions.
- 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.

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 (*see page 192*).
- A function block (*see page 192*) in the application.

Application Settings

See Ethernet Configuration (*see page 123*).

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 (<i>see page 195</i>). For example, name the firewall script <i>FirewallMaintenance.cmd</i> .
2	Load the firewall script on the SD card. For example, load the firewall script in the <i>usr/Cfg</i> folder.
3	In the file <i>Sys/Cmd/Script.cmd</i> , add a code line with the command <code>Firewall_install "/pathname/FileName"</code> For example, the code line is <code>Firewall_install "/sd0/usr/Cfg/FirewallMaintenance.cmd"</code> 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 from an application:

Step	Action
1	Create a valid firewall script (<i>see page 195</i>). For example, name the firewall script <i>FirewallMaintenance.cmd</i> .
2	Load the firewall script in the controller memory. For example, load the firewall script in the <i>usr/Syslog</i> folder with FTP.
3	Use an <code>ExecuteScript</code> function block. For more information, refer to the M262 System Library Guide (<i>see Modicon M262 Logic/Motion Controller, System Functions and Variables, System Library Guide</i>). For example, the [SCmd] input is <code>'Firewall_install "/usr/Syslog/FirewallMaintenance.cmd"'</code> 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 \(see page 266\)](#).

General Firewall Commands

The following commands are available to manage the Ethernet firewall of the M262 Logic/Motion Controller:

Command	Description
<code>Firewall Enable</code>	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.
<code>Firewall Disable</code>	IP addresses can access the controller on the Ethernet interfaces.
<code>Firewall Ethx Default Allow⁽¹⁾</code>	Frames are accepted by the controller on interface Ethx.
<code>Firewall Ethx Default Reject⁽¹⁾</code>	Frames are rejected by the controller on interface Ethx. NOTE: By default, if this line is not present, it corresponds to the command <code>Firewall Eth1 Default Reject</code> .
(1) Where Ethx = <ul style="list-style-type: none"> ● Eth0: USB port ● Eth1: Ethernet_1 ● Eth2: Ethernet_2 ● Eth3: TMSSES4 	

Specific Firewall Commands

The following commands are available to configure firewall rules for specific ports and addresses:

Command	Range	Description
Firewall Eth1 Allow IP	• = 0...255	Frames from the specified IP address are allowed on all port numbers and port types.
Firewall Eth1 Reject IP	• = 0...255	Frames from the specified IP address are rejected on all port numbers and port types.
Firewall Eth1 Allow IPs to	• = 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: <ul style="list-style-type: none"> ● 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	• = 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	Y = (destination port numbers (<i>see page 200</i>))	Frames with the specified destination port number are allowed.
Firewall Eth1 Reject port_type port Y	Y = (destination port numbers (<i>see page 200</i>))	Frames with the specified destination port number are rejected.
Firewall Eth1 Allow port_type ports Y1 to Y2	Y = (destination port numbers (<i>see page 200</i>))	Frames with a destination port number in the specified range are allowed.
Firewall Eth1 Reject port_type ports Y1 to Y2	Y = (destination port numbers (<i>see page 200</i>))	Frames with a destination port number in the specified range are rejected.
(1) If: <ul style="list-style-type: none"> ● x=0, USB port. ● x=1, Ethernet 1 port. ● x=2, Ethernet 2 port. ● x=3, Ethernet port of the TMSSES4. 		

Command	Range	Description
Firewall Eth1 Allow IP on port_type port Y	• = 0...255 Y = (destination port numbers (<i>see page 200</i>))	Frames from the specified IP address and with the specified destination port number are allowed.
Firewall Eth1 Reject IP on port_type port Y	• = 0...255 Y = (destination port numbers (<i>see page 200</i>))	Frames from the specified IP address and with the specified destination port number are rejected.
Firewall Eth1 Allow IP on port_type ports Y1 to Y2	• = 0...255 Y = (destination port numbers (<i>see page 200</i>))	Frames from the specified IP address and with a destination port number in the specified range are allowed.
Firewall Eth1 Reject IP on port_type ports Y1 to Y2	• = 0...255 Y = (destination port numbers (<i>see page 200</i>))	Frames from the specified IP address and with a destination port number in the specified range are rejected.
Firewall Eth1 Allow IPs •1.1.1.1 to •2.2.2.2 on port_type port Y	• = 0...255 Y = (destination port numbers (<i>see page 200</i>))	Frames from an IP address in the specified range and with the specified destination port number are allowed.
Firewall Eth1 Reject IPs •1.1.1.1 to •2.2.2.2 on port_type port Y	• = 0...255 Y = (destination port numbers (<i>see page 200</i>))	Frames from an IP address in the specified range and with the specified destination port number are rejected.
Firewall Eth1 Allow IPs •1.1.1.1 to •2.2.2.2 on port_type ports Y1 to Y2	• = 0...255 Y = (destination port numbers (<i>see page 200</i>))	Frames from an IP address in the specified range and with a destination port number in the specified range are allowed.
Firewall Eth1 Reject IPs •1.1.1.1 to •2.2.2.2 on port_type ports Y1 to Y2	• = 0...255 Y = (destination port numbers (<i>see page 200</i>))	Frames from an IP address in the specified range and with a destination port number in the specified range are rejected.
(1) If: <ul style="list-style-type: none"> ● x=0, USB port. ● x=1, Ethernet 1 port. ● x=2, Ethernet 2 port. ● x=3, Ethernet port of the TMSES4. 		

Command	Range	Description
Firewall Eth1 Allow MAC ••:••:••:••:••:••	• = 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 Eth1 Reject MAC ••:~••:~••:~••:~••:~••	• = 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) If: <ul style="list-style-type: none"> ● x=0, USB port. ● x=1, Ethernet 1 port. ● x=2, Ethernet 2 port. ● x=3, Ethernet port of the TMSSES4. 		

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;
```

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

```
Firewall Enable
```

Comments

```
; 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 web visu (https) on interface ETH2.
```

```
Firewall Eth2 Allow TCP port 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 IP 192.168.1.1 to IP  
192.168.1.2 on UDP port 1740
```

```
; Allow the IP of the commissioning PC and an HMI to  
communicate with the controller using Machine Expert  
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 EtherNet/IP 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 TMSSES4

```

Firewall Eth3 Default Reject ; Reject frames on TMSSES4. 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 TMSSES4.

Firewall Eth3 Allow TCP port 11740 ; Allow Fast TCP on interface TMSSES4. This allow 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.

Ports Used

Protocol	Destination Port Numbers
Machine Expert	UDP 1740, 1741, 1742, 1743 TCP 11740
FTP	TCP 21, 20
HTTP	TCP 80
HTTPS	TCP 443
Modbus	TCP 502
Machine Expert Discovery	UDP 27126, 27127
Web Services Dynamic Discovery	UDP 3702 TCP 5357
SNMP	UDP 161, 162
NVL	UDP Default value: 1202
EtherNet/IP	UDP 2222 TCP 44818
Webvisualization	HTTP 8080 HTTPS 8089
TFTP	UDP 69 (used for FDR server only)

Protocol	Destination Port Numbers
SafeLogger	UDP 35021, 45000
Machine Assistant	UDP 45001...45004

Chapter 13

Industrial Ethernet

Introduction

This chapter describes how to add and configure the Industrial Ethernet.

What Is in This Chapter?

This chapter contains the following topics:

Topic	Page
Industrial Ethernet Presentation	204
DHCP Server	209
Fast Device Replacement	210

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.

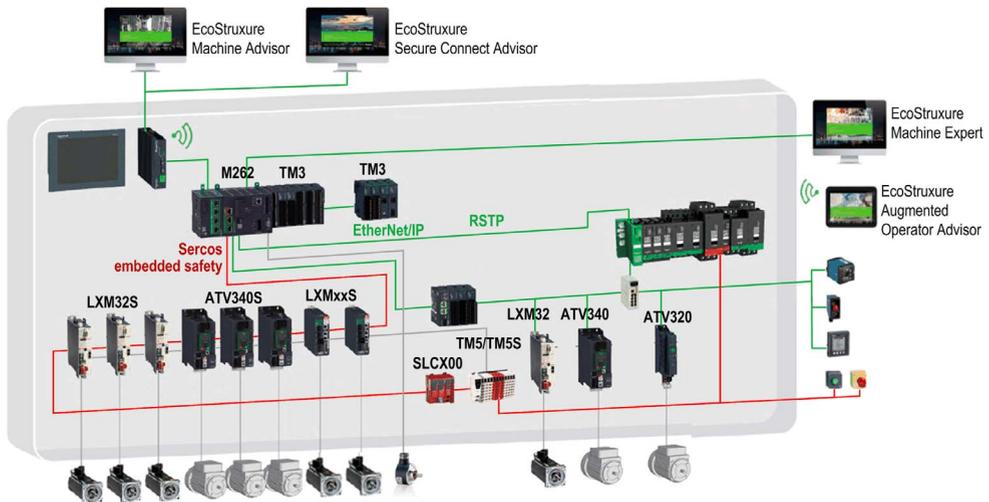
On an Industrial Ethernet network, you can connect:

- industrial devices (industrial protocols)
- non-industrial devices (other Ethernet protocols)

For more information, refer to Industrial Ethernet User Guide (*see EcoStruxure Machine Expert Industrial Ethernet, 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 ⁽¹⁾ EtherNet/IP target devices managed by the logic controller, monitored within a timeslot of 20 ms
Number of connections	0...64 ⁽¹⁾
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 logic controller, monitored within a timeslot of 64 ms.
Number of connections	0...64 ⁽¹⁾
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
(1) 0...6 on Ethernet_1 port of TM262M•.	

M262 Logic/Motion Controller	
Features	Description
Sercos	
Performance	Refer to Performance Overview. (<i>see page 19</i>)
Other services	FDT/DTM/EDS management
	FDR (Fast Device Replacement)
	DHCP server
	Security management (refer to Security Parameters (<i>see page 125</i>) and Firewall Configuration (<i>see page 189</i>))
	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
	FTP server
	SNMP
	IEC VAR ACCESS
Additional features	<p>You can mix the Ethernet/IP and Modbus TCP server devices:</p> <ul style="list-style-type: none"> ● 96 on TM262L10MESE8T and TM262M15MESS8T ● 128 on TM262L20MESE8T, TM262M25MESS8T and TM262M35MESS8T. <p>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).</p> <p>NOTE: Using the controller as a gateway can impact the performance of the controller.</p>
(1) 0...6 on Ethernet_1 port of TM262M•.	

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 ([see page 212](#)).

Adding the Protocol Manager

The protocol manager must be present on the **Ethernet_1 (ETH1)** and **Ethernet_2 (ETH2)** nodes of the device 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.

The protocol manager is available by default under the **Ethernet_1 (ETH1)** and **Ethernet_2 (ETH2)** nodes. It is automatically added when a slave device is added on the **Ethernet_1 (ETH1)** or **Ethernet_2 (ETH2)** node.

To manually add the a function or service to the **Ethernet_1 (ETH1)** or **Ethernet_2 (ETH2)**, select the protocol manager in the **Hardware Catalog** and drag and drop it on one of the highlighted nodes.

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

- Using the Drag-and-drop Method (*see EcoStruxure Machine Expert, Programming Guide*)
- Using the Contextual Menu or Plus Button (*see EcoStruxure Machine Expert, Programming Guide*)

Adding the Sercos Master

The Sercos fieldbus must be present on the **Ethernet_1 (ETH1)** to activate the Sercos Master. It is automatically added when a slave device is added on the **Ethernet_1 (ETH1)** node.

To manually add **Sercos Master** to the **Ethernet_1 (ETH1)**, select **Sercos Master** in the **Hardware Catalog** and drag and drop it on one of the highlighted nodes.

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

- Using the Drag-and-drop Method (*see EcoStruxure Machine Expert, Programming Guide*)
- Using the Contextual Menu or Plus Button (*see EcoStruxure Machine Expert, Programming Guide*)

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 (*see page 85*).

For more information, refer to IP Addressing Methods (*see EcoStruxure Machine Expert 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 EcoStruxure Machine Expert Modbus TCP, User Guide*).

Chapter 14

Sercos Configuration

Introduction

This chapter describes how to configure the Sercos interface of the Modicon M262 Motion Controller.

What Is in This Chapter?

This chapter contains the following topics:

Topic	Page
Overview of the Sercos Standard	212
Modicon M262 Logic/Motion Controller Sercos Configuration	214
Modicon M262 Motion Controller and Safety Controllers with Sercos	215

Overview of the Sercos Standard

Introduction

The Sercos interface is a standardized interface (IEC 6149) 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 (8 axes), 2 ms (16 axes), or 4 ms (24 axes)
- Synchronization (jitter < 1 μ s)

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 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 Embedded Safety for M262 - Integration Guide.

Chapter 15

Serial Line Configuration

Introduction

This chapter describes how to configure the serial line communication of the Modicon M262 Logic/Motion Controller.

What Is in This Chapter?

This chapter contains the following topics:

Topic	Page
Serial Line Configuration	218
Machine Expert Network Manager	220
Modbus Manager	221
ASCII Manager	225
Modbus Serial IOScanner	227
Adding a Device on the Modbus Serial IOScanner	229
Adding a Modem to a Manager	235

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: <ul style="list-style-type: none"> ● 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 Machine Expert protocol by default when new or when you update the controller firmware. The Machine Expert 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)
Machine Expert 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 M262 System Library Guide (*see Modicon M262 Logic/Motion Controller, System Functions and Variables, System Library Guide*).

Machine Expert Network Manager

Introduction

Use the Machine Expert Network Manager to exchange variables with a Magelis Advanced Panel with Machine Expert software protocol, or when the Serial Line is used for EcoStruxure Machine Expert programming.

Adding the Manager

To add a Machine Expert 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 Drag-and-drop Method (*see EcoStruxure Machine Expert, Programming Guide*)
- Using the Contextual Menu or Plus Button (*see EcoStruxure Machine Expert, Programming Guide*)

Configuring the Manager

There is no configuration for Machine Expert Network Manager.

Adding a Modem

To add a modem to the Machine Expert Network Manager, refer to Adding a Modem to a Manager (*see page 235*).

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.

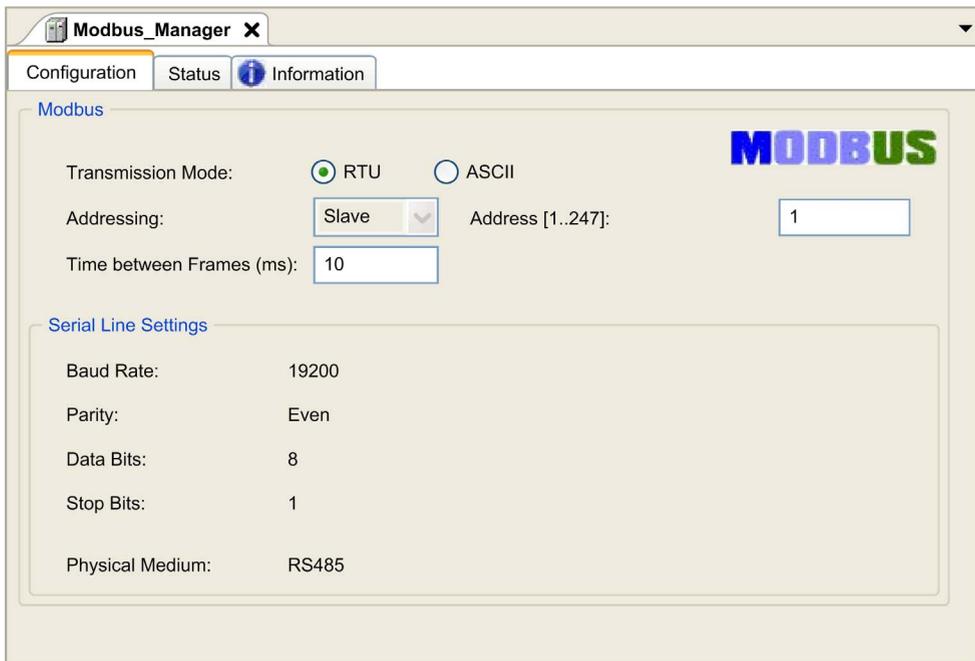
For more information on adding a device to your project, refer to:

- Using the Drag-and-drop Method (*see EcoStruxure Machine Expert, Programming Guide*)
- Using the Contextual Menu or Plus Button (*see 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:



Modbus Manager X

Configuration | Status | Information

Modbus

Transmission Mode: RTU ASCII

Addressing: Slave Address [1..247]: 1

Time between Frames (ms): 10

Serial Line Settings

Baud Rate:	19200
Parity:	Even
Data Bits:	8
Stop Bits:	1
Physical Medium:	RS485

Set the parameters as described in this table:

Element	Description
Transmission Mode	Specify the transmission mode to use: <ul style="list-style-type: none"> ● 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: <ul style="list-style-type: none"> ● 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 EcoStruxure Machine Expert, Modbus and ASCII Read/Write Functions, PLCCommunication Library Guide*) of the PLCCommunication Library.

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)

Function Code Dec (Hex)	Sub-Function Dec (Hex)	Function
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

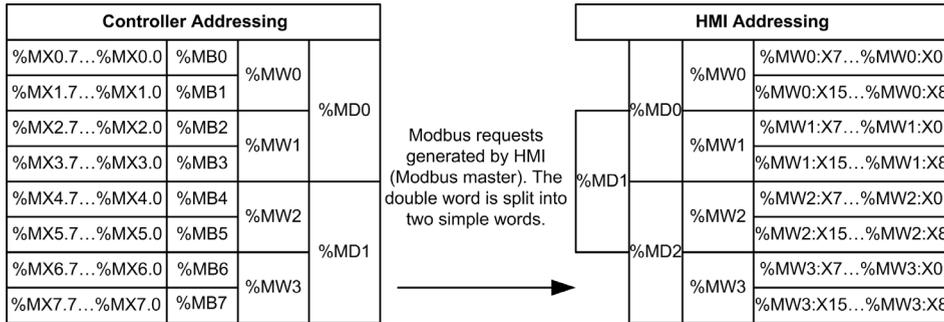
UNINTENDED EQUIPMENT OPERATION

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 ([see page 235](#)).

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.

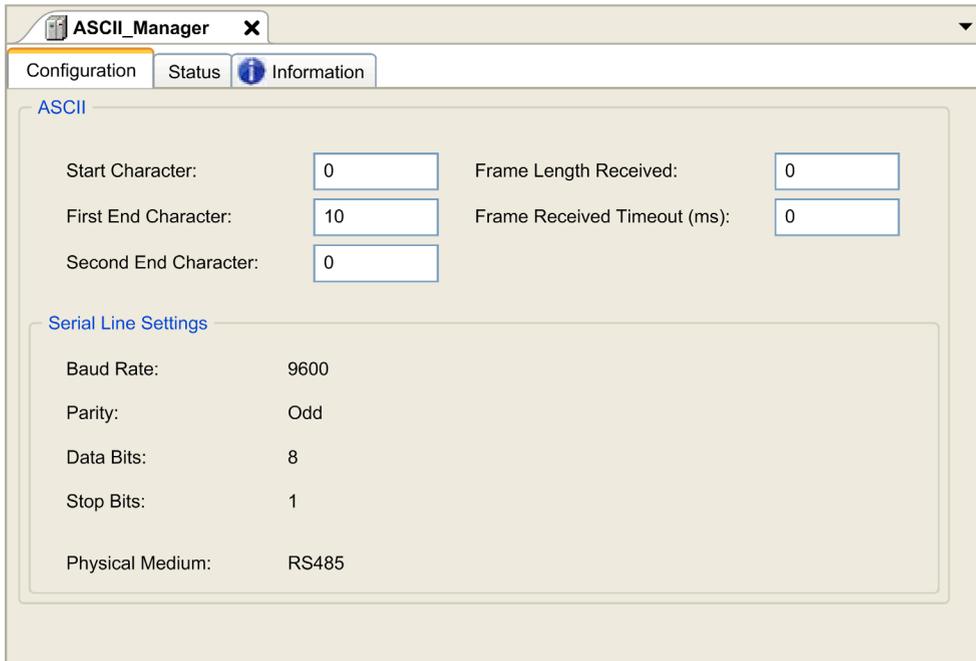
For more information on adding a device to your project, refer to:

- Using the Drag-and-drop Method (*see EcoStruxure Machine Expert, Programming Guide*)
- Using the Contextual Menu or Plus Button (*see 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:



ASCII			
Start Character:	<input type="text" value="0"/>	Frame Length Received:	<input type="text" value="0"/>
First End Character:	<input type="text" value="10"/>	Frame Received Timeout (ms):	<input type="text" value="0"/>
Second End Character:	<input type="text" value="0"/>		
Serial Line Settings			
Baud Rate:	9600		
Parity:	Odd		
Data Bits:	8		
Stop Bits:	1		
Physical Medium:	RS485		

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 (<i>see page 218</i>).

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 (*see page 235*).

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.

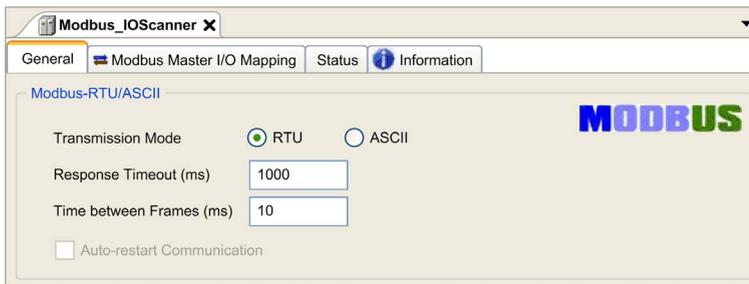
For more information on adding a device to your project, refer to:

- Using the Drag-and-drop Method (*see EcoStruxure Machine Expert, Programming Guide*)
- Using the Contextual Menu or Plus Button (*see 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:

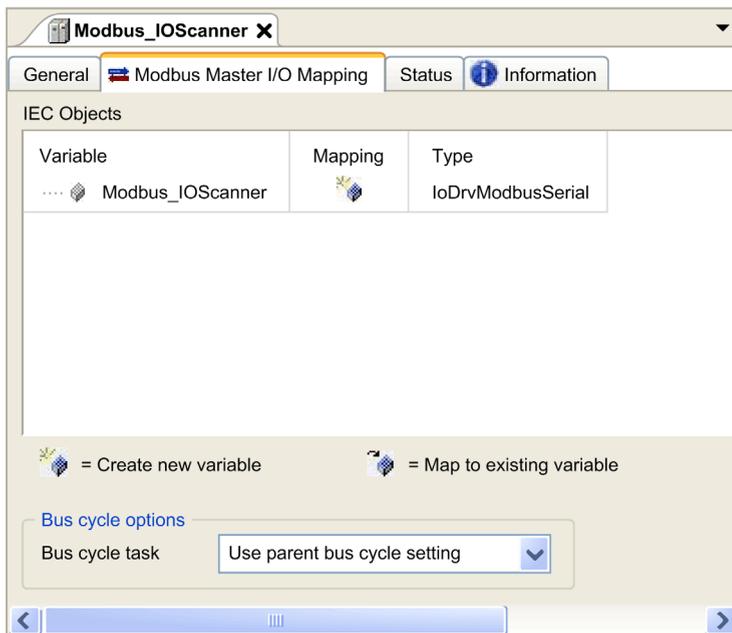
Element	Description
Transmission Mode	Specifies the transmission mode to use: <ul style="list-style-type: none"> • RTU: uses binary coding and CRC error-checking (8 data bits) • ASCII: messages are in ASCII format, LRC error-checking (7 data bits) Set this parameter identical for each Modbus device on 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 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**.

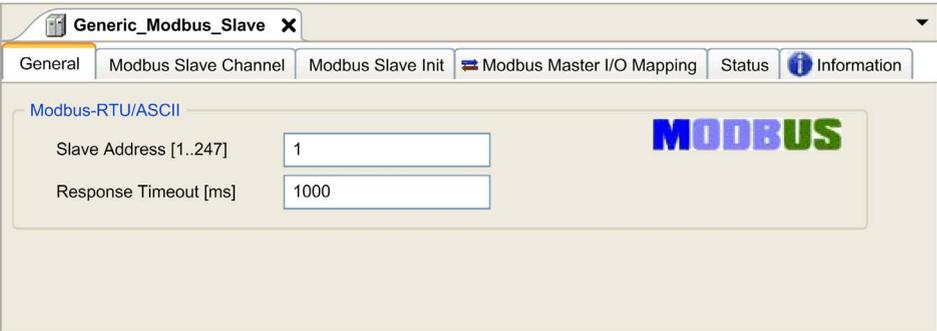
For more information on adding a device to your project, refer to:

- Using the Drag-and-drop Method (*see EcoStruxure Machine Expert, Programming Guide*)
- Using the Contextual Menu or Plus Button (*see EcoStruxure Machine Expert, Programming Guide*)

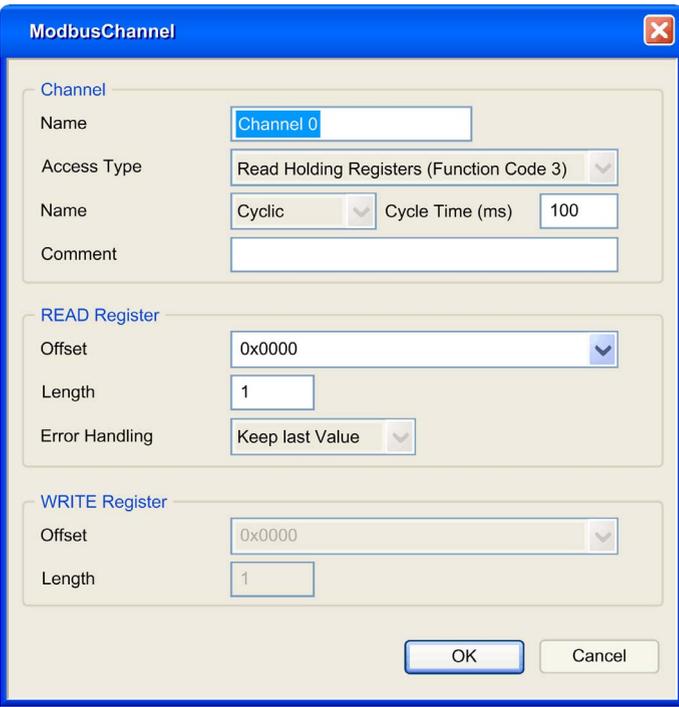
NOTE: The variable for the exchange is automatically created in the **%IWx** and **%QWx** of the **Modbus Serial Master I/O Mapping** tab.

Configuring a Device Added on the Modbus IOScanner

To configure the device added on the Modbus IOScanner, proceed as follows:

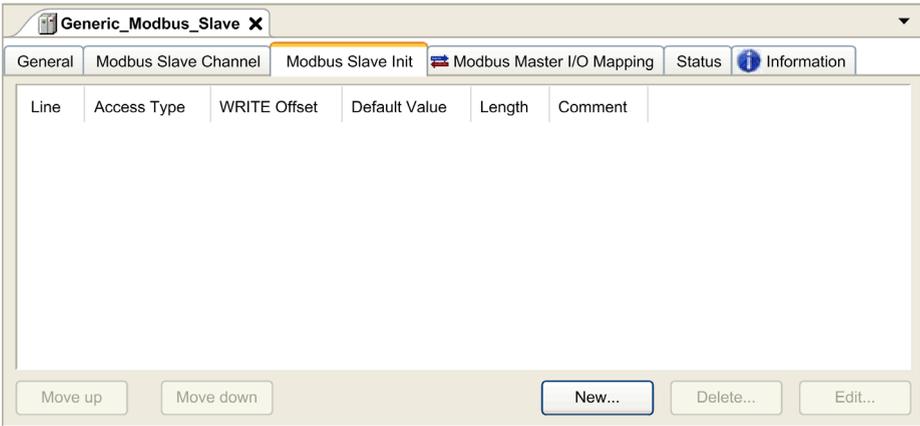
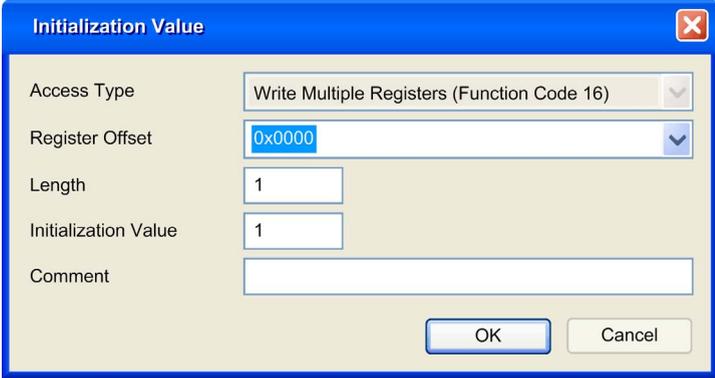
Step	Action
1	<p>In the Devices tree, double-click Generic Modbus Slave. Result: The configuration window is displayed.</p> 
2	Enter a Slave Address value for your device (choose a value from 1 to 247).
3	Choose a value for the Response Timeout (in ms).

To configure the **Modbus Channels**, proceed as follows:

Step	Action
1	<p>Click the Modbus Slave Channel tab:</p> 
2	<p>Click the Add Channel button:</p> 

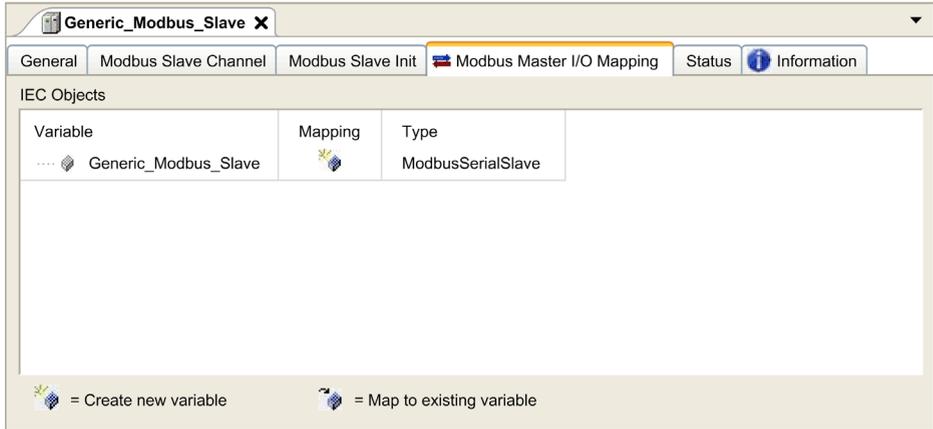
Step	Action
3	<p>Configure an exchange:</p> <p>In the field Channel, you can add the following values:</p> <ul style="list-style-type: none"> ● Channel: Enter a name for your channel. ● Access Type: Choose the exchange type: Read or Write or Read/Write requests (<i>see page 234</i>). ● Trigger: Choose 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. <p>In the field 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):</p> <ul style="list-style-type: none"> ● 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. <p>In the field 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):</p> <ul style="list-style-type: none"> ● 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	<p>Click OK to validate the configuration of this channel.</p> <p>NOTE: You can also:</p> <ul style="list-style-type: none"> ● Click the Delete button to remove a channel. ● Click the Edit button to change the parameters of a channel.

To configure your **Modbus Initialization Value**, proceed as follows:

Step	Action
1	<p>Click the Modbus Slave Init tab:</p> 
2	<p>Click New to create a new initialization value:</p>  <p>The Initialization Value window contains the following parameters:</p> <ul style="list-style-type: none"> ● Access Type: Enter the exchange type: Write requests (<i>see page 234</i>). ● 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.

Step	Action
3	<p>Click OK to create a new Initialization Value.</p> <p>NOTE: You can also:</p> <ul style="list-style-type: none"> • 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	<p>Click the Modbus Master I/O Mapping tab:</p> 
2	<p>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 chose a variable with the Input Assistant.</p>
3	<p>For more information on I/O mapping, refer to EcoStruxure Machine Expert Programming Guide.</p>

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

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 Drag-and-drop Method (*see EcoStruxure Machine Expert, Programming Guide*)
- Using the Contextual Menu or Plus Button (*see EcoStruxure Machine Expert, Programming Guide*)

For further information, refer to Modem Library Guide (*see EcoStruxure Machine Expert, Modem Functions, Modem Library Guide*).

Chapter 16

OPC UA Server Configuration

Introduction

This chapter describes how to configure the OPC UA server of the M262 Logic/Motion Controller.

What Is in This Chapter?

This chapter contains the following topics:

Topic	Page
OPC UA Server Overview	238
OPC UA Server Configuration	239
OPC UA Server Symbols Configuration	243

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 monitored 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.

OPC UA uses a subscription model; clients subscribe to symbols. 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.

User and Group Access Rights

Access to the OPC UA server is controlled by user rights. Refer to Users and Groups in the EcoStruxure Machine Expert Programming Guide.

OPC UA Services

The following table describes the supported OPC UA services:

OPC UA Service	Description
Address Space Model	Yes
Session services	Yes
Attribute services	Yes
Monitored item services	Yes
Queued items	Yes
Subscription services	Yes
Publishing method	Yes

OPC UA Server Configuration

Introduction

The OPC UA Server Configuration window allows you to configure the OPC UA server.

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:

The screenshot displays the OPC UA Server Configuration window with the following sections and settings:

- Tabs:** General settings (selected), Client certificate management.
- General settings:**
 - OPC UA Server enabled
- Security settings:**
 - Disable anonymous login
 - User credentials are managed in the Users and groups tab: [Users and Groups](#)
 - Security Policy: (Dropdown menu open showing: None, Basic256, Basic256Sha256)
 - Message Security: (Dropdown menu open showing: None)
- Server configuration:**
 - Server port: (Spinners)
 - Max subscriptions per session: (Spinners)
 - Max monitored items per subscription: (Spinners)
 - Max number of sessions: (Spinners)
 - Identifier type: (Dropdown)
 - Min publishing interval: ms (Spinners)
 - Min KeepAlive interval: ms (Spinners)
- Diagnostic:**
 - Enable trace (Dropdown)
- Sampling rates (ms):**
 - Double-click to edit (with edit and delete icons)
 - 500
 - 1000
 - 2000
- Buttons:** Reset to default

OPC UA Server Configuration Description

This table describes the OPC UA Server Configuration parameters:

Parameter	Value	Default value	Description
OPC UA Server Enabled	Enabled/ Disabled	Enabled	This checkbox is used to enable or disable the OPC UA Server on the controller.
Security Settings			
Disable anonymous login	Enabled/ Disabled	Disabled	By default, this checkbox is cleared, meaning that OPC UA clients can connect to the server anonymously. Select this checkbox to require that clients provide a valid user name and password to connect to the OPC UA server. NOTE: Anonymous login is disabled when User Rights are activated, regardless of the state of this parameter.
Security Policy	None Basic256 Basic256Sha256	–	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	–	The messages are related to the Security Policy selected.
Server Configuration			
Server port	0...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	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...1000	100	The maximum number of <i>monitored items</i> 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 <i>KeepAlive</i> 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.

Parameter	Value	Default value	Description
Max. number of sessions	1...4	2	The maximum number of clients that can connect simultaneously to the OPC UA server.
Identifier type	Numeric String	Numeric	Certain OPC UA clients require a specific format of unique symbol identifier (node ID). Select the format of the identifiers: <ul style="list-style-type: none"> • Numeric values • Text strings
Diagnostic			
Enable trace	Enabled/disabled	Enabled	Select this checkbox to include OPC UA diagnostic messages in the controller log file (<i>see EcoStruxure Machine Expert, Programming Guide</i>). Traces are available from the Log tab or from the System Log File of the Web Server. You can select the category of events to write to the log file: <ul style="list-style-type: none"> • 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.

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 logic controller as part of the application download.

Each symbol is assigned a unique identifier. As certain client types may require a specific format, identifiers can be configured to be in either string or numeric format.

The OPC UA server supports the following IEC variable types:

- Boolean
- Byte
- Int16, Int32, Int64
- UInt16, UInt32, UInt64
- Float
- Double
- String (255 bytes)
- Sbyte

Bit memory variables (%MX) cannot be selected.

Displaying the List of Variables

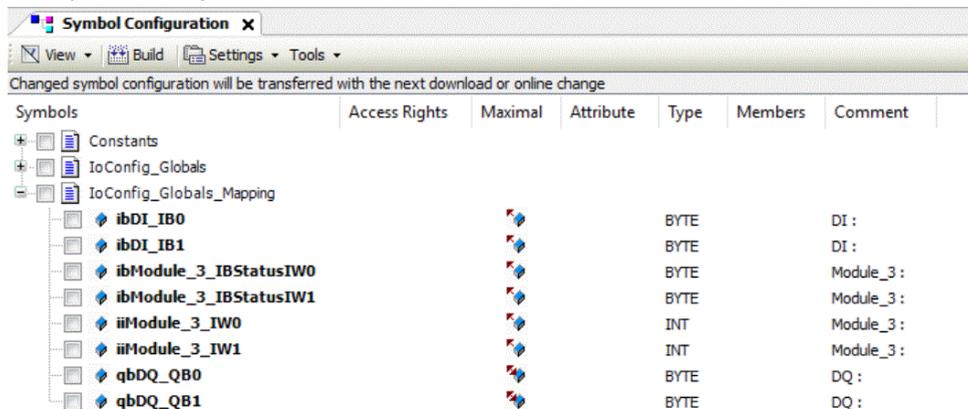
To display the list of variables:

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 ([see page 36](#))).

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.

Chapter 17

Post Configuration

Introduction

This chapter describes how to generate and configure the post configuration file of the Modicon M262 Logic/Motion Controller.

What Is in This Chapter?

This chapter contains the following topics:

Topic	Page
Post Configuration Presentation	246
Post Configuration File Management	247
Post Configuration Example	249

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

NOTE: Parameter updates with a Post Configuration file that impacts parameters used by other devices via 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.

Operating Mode

The Post Configuration file is read after:

- A Reset Warm command (*see page 68*)
- A Reset Cold command (*see page 68*)
- A reboot (*see page 70*)
- An application download (*see page 73*)

Refer to Controller States and Behaviors (*see page 53*) 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
```

where you have to modify only 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 in a comprehensive way.
- The third line is the definition of the parameter (as described above) with its value.

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 → 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 (*see page 272*) (with the proper script)
- Download through the FTP server (*see page 157*)
- Download with EcoStruxure Machine Expert controller device editor (*see page 78*)

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 (*see page 129*).

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 (*see page 157*)
- Online with EcoStruxure Machine Expert controller device editor (*see page 78*), **Files** tab

For more information on **Files** tab of the Device Editor, refer to 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 (*see page 68*)
- A Reset Cold command (*see page 68*)
- A reboot (*see page 70*)
- An application download (*see page 73*)

Post Configuration Example

Post Configuration File Example

```
# TM262M25MESS8T / Ethernet_1 / IPAddress
# Ethernet IP address
id[45000].pos[5].id[111].param[0] = [192, 168, 1, 3]]

# TM262M25MESS8T / Ethernet_1 / SubnetMask
# Ethernet IP mask
id[45000].pos[5].id[111].param[1] = [255, 255, 255, 0]]

# TM262M25MESS8T / Ethernet_1 / GatewayAddress
# Ethernet IP gateway address
id[45000].pos[5].id[111].param[2] = [0, 0, 0, 0]]

# TM262M25MESS8T / Ethernet_1 / IPConfigMode
# IP configuration mode: 0:FIXED 1:BOOTP 2:DHCP
id[45000].pos[5].id[111].param[4] = 0

# TM262M25MESS8T / Ethernet_1 / DeviceName
# Name of the device on the Ethernet network
id[45000].pos[5].id[111].param[5] = 'my_Device'

# TM262M25MESS8T / Ethernet_2 / IPAddress
# Ethernet IP address
id[45000].pos[6].id[45111].param[0] = [192, 168, 102, 2]]

# TM262M25MESS8T / Ethernet_2 / SubnetMask
# Ethernet IP mask
id[45000].pos[6].id[45111].param[1] = [255, 255, 255, 0]]
```

Post Configuration

```
# TM262M25MESS8T / Ethernet_2 / GatewayAddress
# Ethernet IP gateway address
id[45000].pos[6].id[45111].param[2] = [0, 0, 0, 0]

# TM262M25MESS8T / Ethernet_2 / IPConfigMode
# IP configuration mode: 0:FIXED 1:BOOTP 2:DHCP
id[45000].pos[6].id[45111].param[4] = 0

# TM262M25MESS8T / Ethernet_2 / DeviceName
# Name of the device on the Ethernet network
id[45000].pos[6].id[45111].param[5] = 'my_Device'

# TM262M25MESS8T / Serial_Line / Serial Line Configuration / Baudrate
# Serial Line Baud Rate in bit/s
id[45000].pos[7].id[40101].param[10000].Bauds = 19200

# TM262M25MESS8T / 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

# TM262M25MESS8T / Serial_Line / Serial Line Configuration / DataBits
# Serial Line Data bits (7 or 8)
id[45000].pos[7].id[40101].param[10000].DataFormat = 8

# TM262M25MESS8T / Serial_Line / Serial Line Configuration / StopBits
# Serial Line Stop bits (1 or 2)
id[45000].pos[7].id[40101].param[10000].StopBit = 1
```

Chapter 18

Connecting 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

TCSXCNAMUM3P: This USB cable is suitable for short duration connections such as quick updates or retrieving data values.

BMXXCAUSBH018: Grounded and shielded, this USB cable is suitable for long duration connections.

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

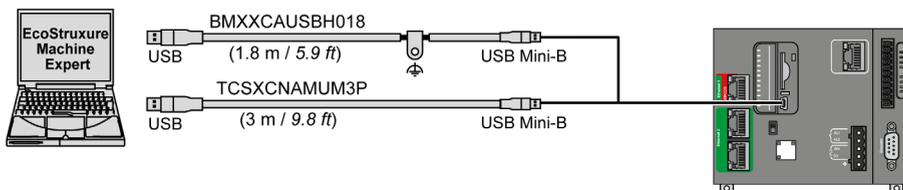
UNINTENDED EQUIPMENT OPERATION OR INOPERABLE EQUIPMENT

- 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.

USB mini-B

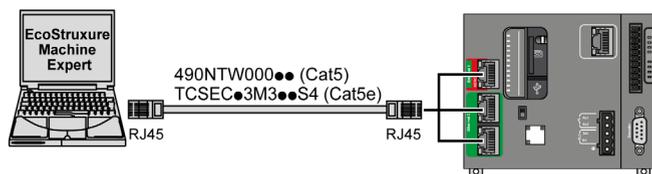


To connect the USB cable to your controller, follow the steps below:

Step	Action
1	<p>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.</p> <p>1b If making a short-term connection using the cable TCSXCNAMUM3P or other non-grounded USB cable, proceed to step 2.</p>
2	Connect your USB cable to the computer.
3	Open the hinged access cover.
4	Connect the Mini connector of your USB cable to the controller USB connector.

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 either of the Ethernet 2 ports on the controller.

Chapter 19

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 firmwares is possible by using an SD card with a compatible script file.

Performing a firmware update deletes the current application program in the device, including the Boot Application in Flash memory.

For more information about the firmware update and creating a new flash disk with firmware, refer to Project Settings - Firmware Update and Flash Memory Organization (*see page 30*).

What Is in This Chapter?

This chapter contains the following topics:

Topic	Page
Updating the Controller Firmware by SD Card	254
Updating the Controller Firmware by Controller Assistant	257
Updating TM3 Expansion Modules Firmware	260
Updating TMS Expansion Modules Firmware	263

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 current application program in the device, including the Boot Application in Flash 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 Machine Expert protocol by default when new or when you update the controller firmware. The Machine Expert 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: <ul style="list-style-type: none"> ● 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.
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

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 Machine Expert protocol by default when new or when you update the controller firmware. The Machine Expert 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

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, click the Update firmware.... button. Result: The Update firmware (step 1 from 4) dialog 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 is displayed.
6	If needed, change the Communication settings and click the Next button. Result: The Update firmware (step 3 from 4) dialog is displayed.
7	Click the Write to... button. Result: The Update firmware (step 4 from 4) dialog is displayed.
8	Select your SD card in Disk drive and click the Write button. At the end of the writing, the Home dialog 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, click the Update firmware.... button. Result: The Update firmware (step 1 from 4) dialog 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 is displayed.
5	If needed, change the Communication settings and click the Next button. Result: The Update firmware (step 3 from 4) dialog is displayed.
6	Click the Write on controller... button. Result: The Update firmware (step 4 from 4) dialog 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 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•
- TM3DI16 and TM3DI16G with firmware version (SV) \geq 2.0
- TM3A• and TM3T• with firmware version (SV) \geq 2.0

NOTE: The firmware 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 <code>/sys/Command</code> and create a file called <code>Script.cmd</code> .
3	Edit the file and insert the following command for each firmware file you wish to transfer to the controller: <code>Download "usr/TM3fwupdate/<filename>"</code>
4	Create the folder path <code>/usr/TM3fwupdate/</code> in the SD card root directory and copy the firmware files to the <code>TM3fwupdate</code> 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 <code>/usr/TM3fwupdate</code> in the controller. During this operation, the SD LED on the controller is flashing. A <code>SCRIPT.log</code> 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 <code>SCRIPT.log</code> file.

Step	Action
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: <i>Your TM3 Module X successfully updated.</i> X corresponds to the position of the module on the bus. NOTE: You can also obtain the logger information in the <code>PlcLog.txt</code> file in the <code>/usr/Syslog/</code> directory of the controller file system. NOTE: If the controller encounters an error during the update, the update terminates with that module.
12	If all targeted modules were successfully updated, delete the firmware file(s) from <code>/usr/TM3fwupdate/</code> 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: <code>Delete "usr/TM3fwupdate/*"</code> NOTE: If a targeted module was not updated successfully, or there are no message logger messages for all the targeted modules, see the Recovery Procedure (see page 261) below.

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 <code>/usr/TM3fwupdate/</code> 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: <code>Your TM3 Module X successfully updated</code> . 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: <code>Your TM3 Module X successfully updated</code> . X corresponds to the position of the module on the bus. NOTE: You can also obtain the logger information in the <code>sys.log</code> file in the <code>/usr/Log</code> directory of the controller file system.
10	Delete the firmware file(s) from <code>/usr/TM3fwupdate/</code> folder on the controller.

Updating TMS Expansion Modules Firmware

Overview

The firmware update for the M262 Logic/Motion Controller are available on the [Schneider Electric](#) website (in .zip format).

Updating TMSES4 and TMSCO1 Modules Firmware

The firmware can be updated in TMSES4 and TMSCO1 modules.

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 TMS 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 (model, 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: <ul style="list-style-type: none"> ● 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.

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.

Chapter 20

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 **Everyone** on objects USB 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.

What Is in This Chapter?

This chapter contains the following topics:

Topic	Page
Creating a Script	266
Generating Scripts and Files	271
Transferring Scripts and Files	272

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 EcoStruxure Machine Expert 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

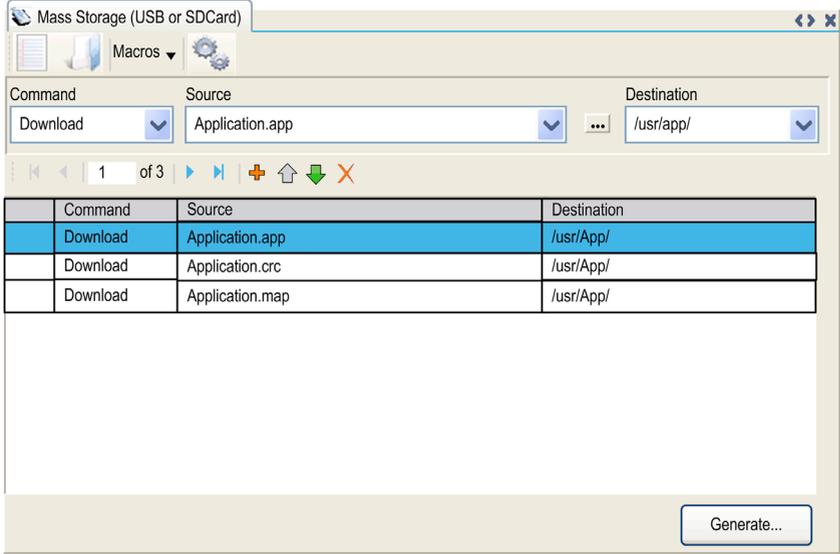
- Do not interrupt the transfer of the application program or a firmware change once the transfer has begun.
- Re-initiate the transfer if the transfer is interrupted for any reason.
- Do not attempt to place the device into service until the file transfer has completed successfully.

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	<p>Click Project → Mass Storage (USB or SDCard) in the main menu. Result: The Mass Storage (USB or SDCard) tab displays:</p> 
2	<p>Click , then .</p>
3	<p>Select a Command.</p>
4	<p>Depending on the selected command, select a Source and Destination.</p>

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/*"'
SetNodeName	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'

This table describes the macros:

Macros	Description	Directory/Files
Download App	Download the application from the SD card to the controller.	/usr/App/*.app
Upload App	Upload the application from the controller to the SD card.	/usr/App/*.crc
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

Removing the User Rights from the Controller

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/*" '`.

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 (<i>see page 266</i>).
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.

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 (see page 266).
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: <ul style="list-style-type: none"> ● 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`).

Chapter 21

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 (*see page 27*).

For security reasons, it does not duplicate the Web Server/FTP password, nor any user access-rights, on any targeted machine.

Cloning the controller is possible by:

- Using an SD Card with a compatible script file
- Using the **Controller Assistant**

What Is in This Chapter?

This chapter contains the following topics:

Topic	Page
Before Cloning a Controller	276
Cloning a Controller	278

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

Ensure access-rights are disabled in the source controller before doing a clone operation. The function block **FB_ControlClone** allows cloning. For more details about this function block, refer to the M262 System Library Guide. For more details about Access Rights, refer to the EcoStruxure Machine Expert Programming Guide.

If you wish to control access to the cloned application in the target controller, you need to enable and establish user access-rights, and any Web Server/FTP passwords, which are controller-specific. 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 controller memory, regardless of any user access-rights that may be enabled in the target controller.

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	Remove power from the source controller.
3	Insert the prepared SD card in the source controller.
4	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.
5	Wait until the copy is completed. Result: The SD LED is ON and the controller starts in normal application mode. If an error was detected, the ERR LED is ON and the controller is in STOPPED state.
6	Remove the SD card from the source controller.
7	Remove power from the target controller.
8	Insert the SD card into the target controller.
9	Restore power to the target controller. Result: The paste starts automatically and the SD LED is flashing during this procedure.
10	Wait until the end of the paste: <ul style="list-style-type: none"> ● If the SD LED (green) is ON, the cloning ended successfully. ● If the SD LED (green) is OFF, and the ERR and I/O LEDs (red) flash regularly, an error is detected. ● If the SD LED (orange) is ON, the cloning is done with an error.
11	Remove the SD card to restart the target controller.

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
pki/own	Not cloned
Rcp	Cloned
Syslog	Not cloned
URF	Not cloned
Visu	Cloned
Web	Cloned

Chapter 22

Compatibility

Software and Firmware Compatibilities

EcoStruxure Machine Expert Compatibility and Migration

Software and Firmware compatibilities are described in the EcoStruxure Machine Expert Compatibility and Migration User Guide.

Chapter 23

Industrial Plug and Work

Introduction

The Industrial Plug and Work function facilitates the machine configuration through Ethernet network.

What Is in This Chapter?

This chapter contains the following sections:

Section	Topic	Page
23.1	Accessing the Web Server	284
23.2	Using the Machine Assistant	285

Section 23.1

Accessing the Web Server

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. (<i>see page 129</i>)

Section 23.2

Using the Machine Assistant

What Is in This Section?

This section contains the following topics:

Topic	Page
Launching the Machine Assistant	286
Managing the Network Scan	287
Managing the Devices Network Settings	289
Backing Up/Restoring Configuration	290
Exporting/Importing .semtd Files	291

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 (*see page 284*) and log in to access the home page of the Web Server site (*see page 129*). Click the **Machine Assistant** tab. The **Machine Assistant** window is displayed.

Launching the Machine Assistant in EcoStruxure Machine Expert

Step	Action
1	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

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 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.

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.

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.

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 (<i>see page 277</i>).
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 (<i>see page 277</i>).
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 .semtdt 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 .semtdt Files

This table describes how to export an .semtdt 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 .semtdt file in your PC. Result: Your project and the detected devices during the scan are exported.

This table describes how to export an .semtdt 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 semtdt file button.
4	Save the .semtdt file on your PC. Result: Your project is exported.

Importing .semtdt Files

The **load .semtdt file** button allows you to upload a project in EcoStruxure Machine Expert or scanned devices in the Web Server.

Appendices



Overview

This appendix lists the documents necessary for technical understanding of the Modicon M262 Logic/Motion Controller Programming Guide.

What Is in This Appendix?

The appendix contains the following chapters:

Chapter	Chapter Name	Page
A	How to Change the IP Address of the Controller	295
B	Functions to Get/Set Serial Line Configuration in User Program	299
C	Controller Performance	305

Appendix A

How to Change the IP Address of the Controller

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 (*see page 121*).

NOTE: For more information on the function block, use the **Documentation** tab of EcoStruxure Machine Expert Library Manager Editor. For the use of this editor, refer EcoStruxure Machine Expert Programming Guide (*see EcoStruxure Machine Expert, Functions and Libraries User Guide*).

Graphical Representation



Parameter Description

Input	Type	Comment
xExecute	BOOL	<ul style="list-style-type: none"> ● 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 (<i>see page 297</i>) 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 (<i>see page 124</i>) 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	<ul style="list-style-type: none"> ● TRUE: error detected, function block aborts action. ● FALSE: no error has been detected.
eError	changeIPAddress_Error	Error code of the detected error (<i>see page 297</i>).
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.

Appendix B

Functions to Get/Set Serial Line Configuration in User Program

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.

What Is in This Chapter?

This chapter contains the following topics:

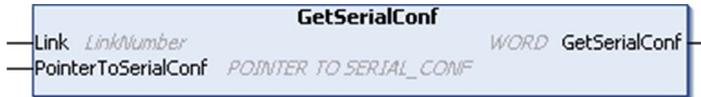
Topic	Page
GetSerialConf: Get the Serial Line Configuration	300
SetSerialConf: Change the Serial Line Configuration	301
SERIAL_CONF: Structure of the Serial Line Configuration Data Type	303

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 (see <i>EcoStruxure Machine Expert, Modbus and ASCII Read/Write Functions, PLCCommunication Library Guide</i>)	Link is the communication port number.
PointerToSerialConf	POINTER TO SERIAL_CONF (see <i>page 303</i>)	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. (See the example below.)

Output	Type	Comment
GetSerialConf	WORD	This function returns: <ul style="list-style-type: none"> ● 0: The configuration parameters are returned ● 255: The configuration parameters are not returned because: <ul style="list-style-type: none"> ○ the function was not successful ○ the function is in progress

Example

Refer to the SetSerialConf (see *page 302*) example.

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 (see <i>EcoStruxure Machine Expert, Modbus and ASCII Read/Write Functions, PLCCommunication Library Guide</i>)	LinkNumber is the communication port number.
PointerToSerialConf	POINTER TO SERIAL_CONF (see page 303)	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: <ul style="list-style-type: none"> ● 0: The new configuration is set ● 255: The new configuration is refused because: <ul style="list-style-type: none"> ○ 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/Machine Expert 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));
    
```

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, <code>FrameReceivedTimeout</code> 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, <code>FrameLengthReceived</code> 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 Machine Expert (see <code>CodesysCompliant</code>)
		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 (polarization 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.

Variable	Type	Description
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: Machine Expert (when Protocol = 0)
CodesysNetType	BYTE	not used

Appendix C

Controller Performance

Processing Performance

Introduction

This chapter provides information about the M262 processing performance.

Logic Processing

This table presents logic processing performance for various logical instructions:

IL Instruction Type	Duration for 1000 Instructions (µs)	
	TM262L10MESE8T 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 (*see page 35*). 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 (μ s)	
	TM262L10MESE8T TM262M15MESS8T	TM262L20MESE8T TM262M25MESS8T TM262M35MESS8T
READ 1000 INT	434	377
WRITE 1000 INT	418	359
READ 1000 BYTE	434	377
WRITE 1000 BYTE	419	359
READ1000 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

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 μ s	80 μ s	100 μ s



A

analog output

Converts numerical values within the logic controller and sends out proportional voltage or current levels.

application

A program including configuration data, symbols, and documentation.

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.

ARP

(*address resolution protocol*) An IP network layer protocol for Ethernet that maps an IP address to a MAC (hardware) address.

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**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.

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.

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.

flash memory

A non-volatile memory that can be overwritten. It is stored on a special EEPROM that can be erased and reprogrammed.

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*.

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

GVL

(global variable list) Manages global variables within an EcoStruxure Machine Expert project.

H

HE10

Rectangular connector for electrical signals with frequencies below 3 MHz, complying with IEC 60807-2.

I

I/O

(input/output)

ICMP

(Internet control message protocol) Reports errors detected and provides information related to datagram processing.

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.

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.

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).

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.

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.

ms

(*millisecond*)

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.

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.

O

OPC UA

OPC Unified Architecture.

P

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).

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. POU's facilitate the modular re-use of software programs, functions, and function blocks. Once declared, POU's 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.

publishing interval

In OPC UA, the frequency at which the OPC-UA server sends notifications to clients informing them that data updates are available.

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 calendar 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

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.

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.

STOP

A command that causes the controller to stop running an application program.

string

A variable that is a series of ASCII characters.

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.

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.



A

- Adding an Encoder
 - Incremental Encoder, *101*
 - SSI Encoder, *101*
- ASCII Manager, *225*

C

- changIPAddress, *295*
 - changing the controller IP address, *295*
- controller configuration
 - communication settings, *80*
- Controller Configuration
 - PLC Settings, *81*
 - Services, *83*
- cyclic data exchanges, generating EDS file for, *161*

D

- DHCP server, *209*
- Download application, *73*

E

- EDS file, generating, *161*
- Embedded Functions Configuration
 - Embedded I/O Configuration, *92*
- Ethernet
 - changIPAddress function block, *295*
- EtherNet
 - EtherNet/IP device, *160*
- Ethernet
 - FTP Server, *157*
 - Modbus TCP Client/Server, *127*
 - Modbus TCP slave device, *184*
 - Services, *119*
 - SNMP, *159*
 - Web server, *129*
- EtherNet/IP Adapter, *160*

- External Event, *44*

F

- Fast Device Replacement, *210*
- features
 - key features, *15*
- firewall
 - configuration, *193*
 - default script file, *193*
 - script commands, *195*
- firmware
 - downloading to TM3 expansion modules, *260*
 - downloading to TMS expansion modules, *263*
- FTP Server
 - Ethernet, *157*

G

- GetSerialConf
 - getting the serial line configuration, *300*

H

- Hardware Initialization Values, *65*

I

- I/O bus configuration, *110*
- I/O configuration general information
 - general practices, *106*
- Industrial Ethernet
 - overview, *204*
- Industrial Plug and Work, *283*
- IP address
 - changIPAddress, *295*

K

KeepAlive (OPC UA), *238*
KeepAlive interval (OPC UA), *241*

L

libraries, *23*

M

M2•• communication
 GetSerialConf, *300*
 SetSerialConf, *301*
Machine Assistant, *285*
Memory Mapping, *27*
Modbus
 Protocols, *127*
Modbus Ioscanner, *227*
Modbus Manager, *221*
Modbus TCP Client/Server
 Ethernet, *127*
monitored items (OPC UA), *238*

O

OPC UA server
 configuration, *239*
 KeepAlive interval, *241*
 overview, *238*
 publishing interval, *241*
 sampling interval, *241*
 selecting symbols, *244*
 symbols configuration, *243*
Output Behavior, *65, 65, 66*
Output Forcing, *66*
overview of the Sercos standard , *212*

P

Post Configuration, *245*
 baud rate, *246*
 data bits, *246*
 device name, *246*
 Example, *249*
 file management, *247*
 gateway address, *246*
 IP address, *246*
 IP configuration mode, *246*
 parity, *246*
 presentation, *246*
 stop bit, *246*
 subnet mask, *246*
programming languages
 IL, LD, Grafcet, *15*
Protocols, *119*
 IP, *121*
 Modbus, *127*
protocols
 SNMP, *159*
publishing interval (OPC UA), *238, 241*

R

Reboot, *70*
Remanent variables, *75*
Reset cold, *68*
Reset origin, *69*
Reset warm, *68*
Run command, *67*

S

sampling interval (OPC UA), *238, 241*
script commands
 firewall, *195*
serial line
 ASCII Manager, *225*
 GetSerialConf, *300*
 Modbus Manager, *221*
 SetSerialConf, *301*
SERIAL_CONF, *303*
SetSerialConf, *301*
 setting the serial line configuration, *301*

SNMP

- Ethernet, *159*
- protocols, *159*
- Software Initialization Values, *65*
- State diagram, *55*
- Stop command, *67*
- symbols (OPC UA), *243*

T

Task

- Cyclic task, *41*
- Event task, *43*
- External Event Task, *44*
- Freewheeling task, *42*
- Types, *41*
- Watchdogs, *47*
- TM3 analog I/O modules
 - downloading firmware to, *260*
- TMS analog I/O modules
 - downloading firmware to, *263*

U

- updating the firmware of TM3 expansion modules, *260*
- updating the firmware of TMS expansion modules, *263*

W

Web server

- Ethernet, *129*

