

Modicon TMS

Expansion Modules

Programming Guide

Original instructions

EIO0000003691.05
12/2024



Legal Information

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

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

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

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

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

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

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

Table of Contents

Safety Information.....	5
About the Book.....	6
TMS Description	10
TMS General Description	10
Configuring the Communication Bus (COM_Bus).....	10
Adding a TMS Expansion Module	12
TMSES4 Ethernet Module.....	14
Ethernet Services	14
Presentation	14
IP Address Configuration	16
Modicon M262 Logic/Motion Controller as a Target Device on EtherNet/IP	20
Modicon M262 Logic/Motion Controller as a Slave Device on Modbus TCP	21
Firewall Configuration	25
Introduction	25
Dynamic Changes Procedure	27
Firewall Behavior	28
Firewall Script Commands	29
Used Ports	33
TMSCO1 CANopen Communications Module	34
Configuring the CANopen Interface.....	34
Glossary	37
Index	40

Safety Information

Important Information

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



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



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

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

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

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

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

Please Note

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

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

About the Book

Document Scope

This document describes the configuration of the TMS expansion modules for EcoStruxure Automation Expert - Motion and EcoStruxure Machine Expert. For further information, refer to the separate documents provided in the online help.

Validity Note

For more information on the validity of the present document, see the online help for the product.

Available Languages of this Document

This document is available in these languages:

- English (EIO0000003691)
- French (EIO0000003692)
- German (EIO0000003693)
- Spanish (EIO0000003694)
- Italian (EIO0000003695)
- Chinese (EIO0000003696)

Related Documents

Title of Documentation	Reference Number
EcoStruxure Automation Expert - Motion, EcoStruxure Machine Expert - Programming Guide	EIO0000002854 (ENG) EIO0000002855 (FRE) EIO0000002856 (GER) EIO0000002858 (SPA) EIO0000002857 (ITA) EIO0000002859 (CHS)
Modicon M262 Logic/Motion Controller - Programming Guide	EIO0000003651 (ENG) EIO0000003652 (FRA) EIO0000003653 (GER) EIO0000003654 (SPA) EIO0000003655 (ITA) EIO0000003656 (CHS)
TMS Expansion Modules - Hardware Guide	EIO0000003699 (ENG) EIO0000003700 (FRA) EIO0000003701 (GER) EIO0000003702 (SPA) EIO0000003703 (ITA) EIO0000003704 (CHS)

Title of Documentation	Reference Number
TMSES4 Expansion Modules - Instruction Sheet	PHA44907
TMSCO1 Expansion Modules - Instruction Sheet	PHA44909

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

Product Related Information

▲ WARNING
<p>LOSS OF CONTROL</p> <ul style="list-style-type: none"> • Perform a Failure Mode and Effects Analysis (FMEA), or equivalent risk analysis, of your application, and apply preventive and detective controls before implementation. • Provide a fallback state for undesired control events or sequences. • Provide separate or redundant control paths wherever required. • Supply appropriate parameters, particularly for limits. • Review the implications of transmission delays and take actions to mitigate them. • Review the implications of communication link interruptions and take actions to mitigate them. • Provide independent paths for control functions (for example, emergency stop, over-limit conditions, and error conditions) according to your risk assessment, and applicable codes and regulations. • Apply local accident prevention and safety regulations and guidelines.¹ • Test each implementation of a system for proper operation before placing it into service. <p>Failure to follow these instructions can result in death, serious injury, or equipment damage.</p>

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
<p>UNINTENDED EQUIPMENT OPERATION</p> <ul style="list-style-type: none"> • Only use software approved by Schneider Electric for use with this equipment. • Update your application program every time you change the physical hardware configuration. <p>Failure to follow these instructions can result in death, serious injury, or equipment damage.</p>

Information on Non-Inclusive or Insensitive Terminology

As a responsible, inclusive company, Schneider Electric is constantly updating its communications and products that contain non-inclusive or insensitive

terminology. However, despite these efforts, our content may still contain terms that are deemed inappropriate by some customers.

Terminology Derived from Standards

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

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

Among others, these standards include:

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

TMS Description

TMS General Description

Introduction

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

TMS Expansion Module Features

The following table describes the TMS expansion module features:

Module reference	Type	Terminal type	Compatibility
TMSES4	Ethernet communication	RJ45	TM262L10MESE8T TM262L20MESE8T TM262M15MESS8T TM262M25MESS8T TM262M35MESS8T
TMSCO1	CANopen master module	SUB-D 9 pin male	TM262L• TM262M•

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

Configuring the Communication Bus (COM_Bus)

Configuring the Communication Bus

To configure the communication bus, proceed as follows:

Step	Action
1	In the Devices tree , double-click COM_Bus . Result: The COM_Bus configuration window is displayed.
2	Click one of the tabs: <ul style="list-style-type: none"> • TMS Bus • I/O Mapping • Diagnostic Table

TMS Bus Tab

The TMS communication bus has an internal IP network architecture. The network address is fixed for general configurations, however, the network address must be entered manually for complex configurations requiring multiple networks and interconnected M262 controllers.

To configure the network address, proceed as follows:

Step	Action
1	Click Network address .
2	Type the new network address. Result: The Subnet mask , Host min , and Host max fields are automatically updated.

I/O Mapping Tab

The **I/O Mapping** tab is fixed and cannot be modified.

Diagnostic Table Tab

The **Diagnostic Table** tab provides a diagnostic status of each connected module.

NOTE: This table is only for TMSES4 modules.

Parameter	Data type	Default value	Value	Description
<i>ConfState</i>	UNIT	0	0: No configuration	Global state of the bus
			1: Configuration not valid	
			2: Reserved	
			3: Configuration valid and applied	
<i>NbModules</i>	UNIT	0	0...3	Number of detected TMS modules
<i>Name</i>	STRING(15)	–	–	TMS module name
<i>MajorType</i>	WORD	0	–	Type code of the TMS module
<i>SubType</i>	WORD	0	–	Subtype code of the TMS module
<i>Version</i>	STRING(15)	–	–	Firmware version of the TMS module ⁽¹⁾
<i>ModuleState</i>	DWORD	TMS_MODULE_POWERED	0	Detection of the TMS module by the controller
		TMS_MODULE_INITIALIZED	1	
		TMS_MODULE_CONFIGURED	2	
		TMS_MODULE_EXCHANGE_FAULT	3	
		MODULE_ERROR	4	
		TMS_MODULE_HEALTH_SEND_FAULT	5	
		TMS_MODULE_HEALTH_RCV_TIMEOUT	6	
		TMS_MODULE_HEALTH_RCV_MISC	7	
		TMS_MODULE_HEALTH_RESP_ERR	8	
		TMS_MODULE_DISCOVERY	9	

Parameter	Data type	Default value	Value	Description
<i>IpState</i>	DWORD	TMS_IP_PING_SUCCESS	0	IP communication between M262 and TMS module
		TMS_IP_CONFIG_CMD_ERROR	1	
		TMS_IP_CONFIG_RESP_WAIT	2	
		TMS_IP_CONFIG_RESP_ERROR	3	
		TMS_IP_CONFIG_RESP_NONE	4	
		TMS_IP_CONFIG_SUCCESS	5	
		TMS_IP_PING_CMD_ERROR	6	
		TMS_IP_PING_RESP_WAIT	7	
		TMS_IP_PING_RESP_ERROR	8	
		TMS_IP_PING_RESP_NONE	9	
		TMS_IP_NOT_CONFIGURED	11	
<i>PixCmdState</i>	Enumeration of DWORD	TMS_PIXCMD_EXCHING	0	TMS module is handling a process image
		TMS_PIXCMD_CONFIG_NONE	1	
		TMS_PIXCMD_CONFIG_CMD_ERROR	2	
		TMS_PIXCMD_CONFIG_RESP_WAIT	3	
		TMS_PIXCMD_CONFIG_RESP_ERROR	4	
		TMS_PIXCMD_CONFIG_ONLY	5	
		TMS_PIXCMD_CONFIG_SUCCESS	6	
		TMS_PIXCMD_ENABLE_CMD_ERROR	7	
		TMS_PIXCMD_ENABLE_RESP_WAIT	8	
		TMS_PIXCMD_ENABLE_RESP_ERROR	9	
		TMS_PIXCMD_EXCH_ERROR	10	
		TMS_PIXCMD_DISABLING	11	
		TMS_PIXCMD_DISABLED	12	

(1) Refer to Updating TMSSES4 Expansion Module Firmware (see Modicon M262 Logic/Motion Controller, Programming Guide) for information on how to update the TMSSES4 expansion module firmware.

Adding a TMS Expansion Module

Adding a TMS Expansion Module

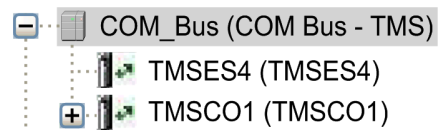
To add a TMS expansion module to your controller, select the expansion module in the **Hardware Catalog**, drag it to the **Devices tree**, and drop it on the **COM_Bus** node.

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

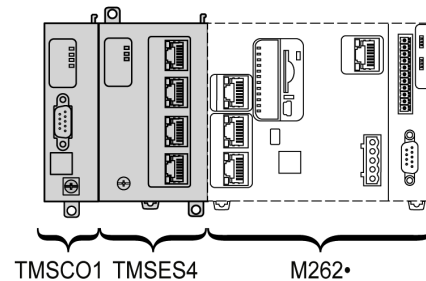
- Using the Drag-and-drop Method found in the EcoStruxure Automation Expert - Motion / EcoStruxure Machine Expert Programming Guide
- Using the Contextual Menu or Plus Button found in the EcoStruxure Automation Expert - Motion / EcoStruxure Machine Expert Programming Guide

Expansion Module Layout

In the software, the module layout is displayed from the top to the bottom:



Physically, the expansion modules are connected from the right to the left:



For more information about the compatibility with the M262 Logic/Motion Controller, refer to [TMS Expansion Module Features](#), page 10.

Configuring an Expansion Module

To configure your TMS expansion module, double-click the expansion module node in the **Devices tree**.

TMSES4 Ethernet Module

Introduction

This chapter describes the configuration of the TMSES4 Ethernet expansion module.

Ethernet Services

Introduction

This section describes how to configure the Ethernet services provided by the TMSES4 expansion module.

Presentation

Ethernet Services

The TMSES4 expansion module provides an Ethernet interface to extend the number of Ethernet ports for a controller.

The module supports the following controller services:

- Modbus TCP Server, page 15
- Web Server (see Modicon M262 Logic/Motion Controller, Programming Guide)
- FTP Server (see Modicon M262 Logic/Motion Controller, Programming Guide)
- SNMP (see Modicon M262 Logic/Motion Controller, Programming Guide)
- M262 Logic/Motion Controller as Target Device on EtherNet/IP, page 20
- M262 Logic/Motion Controller as Slave Device on Modbus TCP, page 21
- IEC VAR access, page 15

NOTE: Network Variable List (NVL) communication requires that the Ethernet port has a valid IP address and that the device is connected.

Ethernet Protocol

The Ethernet module supports the following protocols:

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

TCP Server Connections

This table shows the total of TCP server connections for the controller and the TMSES4 modules:

Connection Type	Maximum Number of Simultaneous Server Connections
Modbus Server	8 simultaneous TCP server connections maximum for TMSES4 and controller or for the controller alone.
EtherNet/IP Device	16
FTP Server	4
Web Server	10

Each server based on TCP manages its own set of connections.

When a client tries to open a Modbus Server connection that exceeds the maximum number of connections, the controller closes the oldest connection. In other cases, the attempt to open a connection is denied.

If all connections are busy (exchange in progress) when a client tries to open a new one, the new connection is denied.

The server connections stay open as long as the controller stays in operational states (*RUN, STOP, HALT*).

The server connections are closed when leaving or entering operational states (*RUN, STOP, HALT*), except in the case of power outage (because the controller does not have time to close the connections).

For more information about the operational states, refer to the controller state diagram (see Modicon M262 Logic/Motion Controller, Programming Guide).

Modbus TCP Server

The Modbus server supports the following Modbus requests:

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

Available Services

With an Ethernet communication, the **IEC VAR ACCESS** service is supported by the controller. The **IEC VAR ACCESS** service allows an exchange of variables between the controller and an HMI.

In addition, the **NetWork variables** service is supported by the controller. The **NetWork variables** service allows an exchange of data between controllers.

NOTE: For more information, refer to the EcoStruxure Automation Expert - Motion / EcoStruxure Machine Expert Programming Guide.

IP Address Configuration

Introduction

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

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

x and z represent the 5th and the 6th bytes of interface MAC address. For example, with a MAC address of 00:80:F4:50:02:5D, the IP address will be 10.12.2.93.

See Ethernet Configuration, page 18 for more information about the MAC address location.

The default subnet mask is 255.255.0.0.

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

- Address assignment by DHCP server
- Address assignment by BOOTP server
- Fixed IP address
- Post configuration file (see Modicon M262 Logic/Motion Controller, Programming Guide). If a post configuration file exists, this assignment method has priority over the others.

The IP address can also be changed dynamically through the:

- **Communication Settings** tab (see Modicon M262 Logic/Motion Controller, Programming Guide)
- **changeIPAddress** function block (see Modicon M262 Logic/Motion Controller, Programming Guide)

NOTE: If the attempted addressing method is unsuccessful, the link uses a default IP address 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.

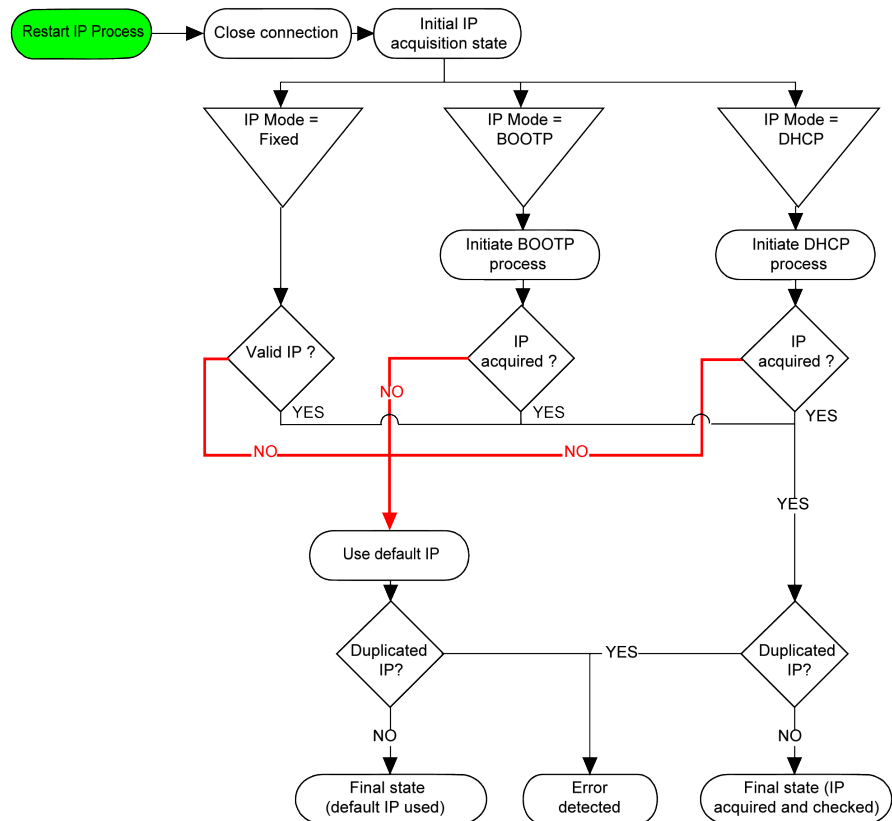
Failure to follow these instructions can result in death, serious injury, or equipment damage.

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.

NOTE: The TMSES4 module must be in a different subnetwork than the controller Ethernet ports.

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. However, the process is repeated until the respective server is reached and an IP address is acquired.

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 **TMSES4**:

NOTE:

- If you are in offline mode, you see the **Configured Parameters** window (displayed above). You can edit the parameters.
- If you are in online mode, you see the **Configured Parameters** and **Current Settings** windows. You cannot edit the parameters.

This table describes the configured parameters:

Configured Parameters	Description
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 left side of the controller.
Fixed IP Address	IP address, Subnet Mask, and Gateway Address are defined by the user.
Ethernet Protocol	Protocol type used: Ethernet 2
Transfer Rate	Speed and Duplex are in auto-negotiation mode.

Default IP Address

The MAC address of the Ethernet port can be retrieved on the label placed on the front side of the M262 controller. The MAC address of the TMSES4 port can be retrieved on the label placed on the left side of the M262 controller.

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

Example of conversion:

Port	MAC address	IP address
TMS_1	00.80.F4.50.03.31	10.12.3.49
TMS_2	00.80.F4.50.03.32	10.13.3.50
TMS_3	00.80.F4.50.03.33	10.14.3.51

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 does not communicate 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 external networks is sent through this interface.

Security Parameters

This table describes the different security parameters:

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

Modicon M262 Logic/Motion 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

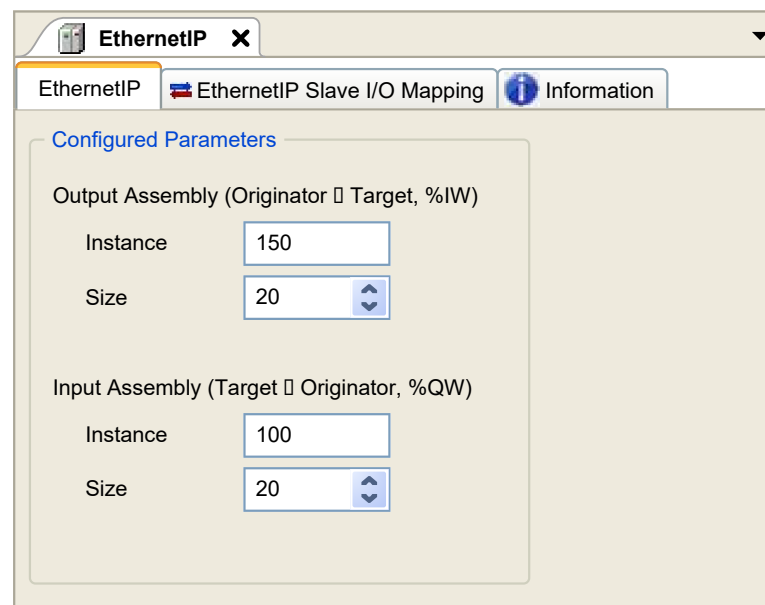
To configure your M262 Logic/Motion Controller as a target device on Ethernet/IP, follow these steps:

Step	Action
1	Select EthernetIP in the Hardware Catalog .
2	<p>Drag and drop it to the Devices tree on one of the highlighted nodes.</p> <p>NOTE: If the chosen node is COM_Bus, a TMSES4 expansion module is automatically added to your configuration.</p> <p>For more information on adding a device to your project, refer to:</p> <ul style="list-style-type: none"> • Using the Drag-and-drop Method found in the EcoStruxure Automation Expert - Motion / EcoStruxure Machine Expert Programming Guide • Using the Contextual Menu or Plus Button found in the EcoStruxure Automation Expert - Motion / EcoStruxure Machine Expert Programming Guide

EtherNet/IP Parameters Configuration

To configure the EtherNet/IP parameters, double-click **EthernetIP** in the **Devices Tree**.

This dialog box is displayed:



The EtherNet/IP I/O configuration parameters are defined as:

- **Instance:**

Number referencing the input or output Assembly.

- **Size:**

Number of channels of an input or output Assembly.

Each channel has a 2-byte memory 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, there are 20 input channels (IW0...IW19) addressing %IWy...%IW(y+20-1), where y is the first available channel for the Assembly.

Element		Admissible Controller Range	Default Value
Output Assembly	Instance	150...189	150
	Size	2...250	20
Input Assembly	Instance	100...149	100
	Size	2...250	20

Refer to the M262 Logic/Motion Controller Programming Guide for more information on the following topics:

- Generating an EDS file
- Configuring I/Os
- Objects supported by the controller

Modicon M262 Logic/Motion 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**.

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).

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 provide the controller I/O objects to this area, which can then be accessed with a single Modbus read/write registers request.

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. See the *Modbus TCP Configuration*, page 22.

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 eight connections are needed).

The timeout duration 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, see the Ethernet Port

Read Only System Variables (see Modicon M262 Logic/Motion Controller, System Functions and Variables, System Library Guide).

For further information about Modbus TCP, refer to the www.modbus.org website.

Adding a Modbus TCP Slave Device

To add Modbus TCP slave device functionality to your M262 Logic/Motion Controller, follow these steps:

Step	Action
1	Select Modbus TCP Slave Device in the Hardware Catalog .
2	Drag and drop it to the Devices tree on one of the highlighted nodes. NOTE: If the chosen node is COM_Bus , a TMSES4 expansion module is automatically added to your configuration. For more information on adding a device to your project, refer to: <ul style="list-style-type: none">• Using the Drag-and-drop Method found in the EcoStruxure Automation Expert - Motion / EcoStruxure Machine Expert Programming Guide• Using the Contextual Menu or Plus Button found in the EcoStruxure Automation Expert - Motion / EcoStruxure Machine Expert Programming Guide

Configuring a Modbus TCP Slave Device

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

This dialog box appears:

Configured Parameters

IP Master address:

Watchdog: (ms) Close TCP socket

Slave port: Bind to adapter

Unit ID:

Holding registers: (%IW) Writeable

Input registers: (%QW)

Discrete Bit Areas

Coils: (%IX)

Discrete Inputs: (%QX)

Data Model

Start addresses

Coils:

Discrete inputs:

Holding register:

Input register:

Holding- and input register data areas overlay

Element	Description
IP Master address	IP address of the Modbus master The connections are not closed on this address.
Watchdog	Watchdog in 500 ms increments NOTE: The watchdog applies to the IP Master address unless the address is 0.0.0.0.
Close TCP socket	When Close TCP socket is selected, the TCP socket is closed if the Watchdog is enabled and the set time is exceeded.
Slave port	Modbus communication port (502)
Unit ID	Sends the requests to the Modbus TCP slave device (1...247), instead of the embedded Modbus server (255).
Holding registers (%IW)	Number of %IW registers to be used in the exchange (2...120) (each register is 2 bytes)
Input registers (%QW)	Number of %QW registers to be used in the exchange (2...120) (each register is 2 bytes)

Modbus TCP Slave Device I/O Mapping Tab

The I/Os are mapped to Modbus registers from the master perspective in the following way:

- %IW's are mapped from register 0 to n-1 and are R/W (n = Holding register quantity, each %IW register is 2 bytes).
- %QW's are mapped from register n to n+m -1 and are read only (m = Input registers quantity, each %QW register is 2 bytes).

When a **Modbus TCP Slave Device** has been configured, Modbus commands sent to its Unit ID (Modbus address) are handled differently than the same command would be when addressed to any other Modbus device on the network. For example, when the Modbus command 3 (3 hex) is sent to a Modbus device, it reads and returns the value of one or more registers. When this same command is sent to the Modbus TCP Slave (see M262 Logic/Motion Controller, Programming Guide), it facilitates a read operation by the external I/O scanner.

When a **Modbus TCP Slave Device** has been configured, Modbus commands sent to its Unit ID (Modbus address) access to the %IW and %QW objects of the controller, linked to the Modbus TCP device, instead of the regular Modbus words (accessed when the Unit ID is 255). This facilitates read/write operations by a Modbus TCP IOScanner application.

The **Modbus TCP Slave Device** responds to a subset of the Modbus commands 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:

The screenshot displays the 'Modbus TCP Slave Device I/O Mapping' configuration window. It features a tabbed interface with 'General', 'Modbus TCP Slave Device I/O Mapping', and 'Information' tabs. A search bar and a 'Filter' dropdown are at the top. The main area contains a table with the following data:

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

Below the table, there are controls for 'Reset mapping', 'Always update variables' (set to 'Enabled 1 (use bus cycle task if not used in any task)'), and 'Bus cycle options' (set to '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 the client/master controller (%IW for the server/slave controller). Input means INPUT from the client/master controller (%QW for the server/slave controller).

For the parameter **Always update variables**, choose one of the following options:

- **Use parent device setting**
- **Enabled 1 (use bus cycle task if not used in any task)** (default setting)
- **Enabled 2 (always in bus cycle task)**

Bus Cycle Options

In the **Modbus TCP Slave Device I/O Mapping** tab, select the **Bus cycle task** to use:

- **Use parent bus cycle setting** (default setting)
- **MAST**
- **An existing task of the project:** you can select an existing task and associate it to the scanner. For more information about the application tasks, refer to the EcoStruxure Automation Expert - Motion / 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.

Firewall Configuration

Introduction

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

Introduction

Firewall Presentation

In general, firewalls help protect network security zone perimeters by blocking unauthorized access and permitting authorized access. A firewall is a device or set of devices configured to permit, deny, encrypt, decrypt, or proxy traffic between different security zones based upon a set of rules and other criteria.

Process control devices and high-speed manufacturing machines require fast data throughput and often cannot tolerate the latency introduced by an aggressive security strategy inside the control network. Firewalls, therefore, play a significant

role in a security strategy by providing levels of protection at the perimeters of the network. Firewalls are an important part of an overall, system level strategy.

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

▲ WARNING

UNAUTHENTICATED ACCESS AND SUBSEQUENT UNAUTHORIZED MACHINE OPERATION

- Evaluate whether your 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, such as an Intrusion Prevention System or Intrusion Detection System.
- Monitor activities within your systems.
- Prevent subject devices from direct access or direct link by unauthorized parties or unauthenticated actions.
- Install certificates that are issued by publicly known Trusted Certificate Authorities.
- Keep your systems up-to-date and rely only on legitimate sources.
- Prepare a recovery plan including backup of your system and process information.

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

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

Firewall Configuration

There are three ways to manage the controller firewall configuration:

- Static configuration
- Dynamic changes
- Application settings

Script files are used in the static configuration and for dynamic changes.

Static Configuration

The static configuration is loaded at the controller boot.

The controller firewall can be statically configured by managing a default script file located in the controller. The path to this file is `/usr/Cfg/FirewallDefault.cmd`.

NOTE: The file name is case sensitive.

Dynamic Changes

After the controller boot, the controller firewall configuration can be changed by the use of script files.

There are two ways to load these dynamic changes using:

- A physical SD card, page 27.
- A function block, page 27 in the application.

Application Settings

See Ethernet Configuration, page 18.

Dynamic Changes Procedure

Using an SD Card

This table describes the procedure to execute a script file from an SD card:

Step	Action
1	Create a valid script file, page 29. For example, name the script file <i>FirewallMaintenance.cmd</i> .
2	Load the script file on the SD card. For example, load the script file 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 script file from an application:

Step	Action
1	Create a valid script file, page 29. For example, name the script file <i>FirewallMaintenance.cmd</i> .
2	Load the script file in the controller memory. For example, load the script file 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 (see Modicon M262 Logic/Motion Controller, System Functions and Variables, System Library Guide). 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.

NOTE: To determine whether the firewall is configured and enabled, consult the message logger.

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.

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 Modicon M262 Logic/Motion Controller, Programming Guide).

General Firewall Commands

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

Command	Description
Firewall Enable	Blocks the frames from the Ethernet interfaces. If no specific IP address is authorized, it is not possible to communicate on the Ethernet interfaces. NOTE: By default, when the firewall is enabled, the frames are rejected.
Firewall Disable	Firewall rules are not applied. Frames are not blocked
Firewall Ethx Default Allow ⁽¹⁾	Frames are accepted by the controller.
Firewall Ethx Default Reject ⁽¹⁾	Frames are rejected by the controller. 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"> • Eth1: Ethernet_1 • Eth2: Ethernet_2 • Eth3: TMSES4 (first Ethernet module from the left) • Eth4: TMSES4 (second Ethernet module from the left) • Eth5: TMSES4 (third Ethernet module from the left) 	

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.
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)	Frames with the specified destination port number are allowed.
Firewall Eth1 Reject port_type port Y	Y = (destination port numbers)	Frames with the specified destination port number are rejected.
Firewall Eth1 Allow port_type ports Y1 to Y2	Y = (destination port numbers)	Frames with a destination port number in the specified range are allowed.
Firewall Eth1 Reject port_type ports Y1 to Y2	Y = (destination port numbers)	Frames with a destination port number in the specified range are rejected.
Firewall Eth1 Allow IP ••••• on port_type port Y	• = 0...255 Y = (destination port numbers)	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)	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)	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)	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)	Frames from an IP address in the specified range and with the specified destination port number are allowed.

Command	Range	Description
Firewall Eth1 Reject IPs •1.1.1.1 to •2.2.2.2 •2 on port_type port Y	• = 0...255 Y = (destination port numbers)	Frames from an IP address in the specified range and with the specified destination port number are rejected.
Firewall Eth1 Allow IPs •1.1.1.1 to •2.2.2.2 •2 on port_type ports Y1 to Y2	• = 0...255 Y = (destination port numbers)	Frames from an IP address in the specified range and with a destination port number in the specified range are allowed.
Firewall Eth1 Reject IPs •1.1.1.1 to •2.2.2.2 •2 on port_type ports Y1 to Y2	• = 0...255 Y = (destination port numbers)	Frames from an IP address in the specified range and with a destination port number in the specified range are rejected.
Firewall 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.
<p>(1) If:</p> <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 (first Ethernet module from the left). • x=4: Ethernet port of the TMSES4 (second Ethernet module from the left). • x=5: Ethernet port of the TMSES4 (third Ethernet module from the left). 		

Script Example

The following is an example of a Firewall in white list mode. The example has all communication blocked by default and allows only the necessary services.

NOTE: This example is designed to show most of the commands available with the firewall. It should be adapted to your configuration and tested before implementation.

Commands	Comments
Firewall Enable	Enable the firewall.
Eth1 Configuration	
Firewall Eth1 Default Reject	Reject all frames on interface ETH1. In this example, ETH1 is connected to the Industrial Ethernet devices network and therefore can be relatively trusted.
Firewall Eth1 Allow TCP port 502	Allow Modbus TCP server on interface ETH1. There is no authentication on Modbus so this should be allowed only on trusted networks.
Firewall Eth1 Established to TCP port 502	Allow replies to communication established by the controller to TCP port 502. This is necessary when using PlcCommunication library to communicate using Modbus TCP protocol.
Firewall Eth1 Allow UDP port 2222	Allow ETHIP scanner implicit exchanges replies to UDP port 2222 (ETHIP) on interface ETH1.
Firewall Eth1 Established to TCP port 44818	Allow replies to communication established by the controller to TCP port 44818 (ETHIP) on interface ETH1. The last 2 commands allow the EtheNetIP Scanner to communicate with the industrial ethernet devices.
Eth2 Configuration	
Firewall Eth2 Default Reject	Reject all frames on interface ETH2. This interface is connected to a network used mainly for commissioning.
Firewall Eth2 Allow TCP port 4840	Allow OPC-UA server on interface ETH2.
Firewall Eth2 Allow TCP port 443	Allow web server (https) on interface ETH2.
Firewall Eth2 Allow TCP port 8089	Allow web visu (https) on interface ETH2.
Firewall Eth2 Allow TCP ports 20 to 21	Allow ftp in active mode on interface ETH2.
Firewall Eth2 Allow IP 192.168.1.1 on UDP ports 27126 to 27127	Allow the IP of the commissioning PC to discover and configure the IP address of the controller. This should be allowed only on a trusted network as IP can be changed even if the User Rights are configured.
Firewall Eth2 Allow IPs 192.168.1.1 to 192.168.1.2 on UDP port 1740	Allow the IP of the commissioning PC and an HMI to communicate with the controller using 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 EtherNetIP Adapter.
Firewall Eth2 Allow MAC 4C:CC:6A:A1:09:C8	Allow the MAC address of the HMI.
Firewall Eth2 Allow MAC 00:0C:29:92:43:A8	Allow the MAC address of the commissioning PC. Only the MAC addresses allowed can communicate with the controller.
Eth3 Configuration TMSES4	
Firewall Eth3 Default Reject	Reject frames on TMSES4. This interface is connected to the Plant network and can access the web. It should be considered as untrusted.
Firewall Eth3 Established to TCP port 443	Allow https client (for example to connect to Machine Advisor) on interface TMSES4.
Firewall Eth3 Allow TCP port 11740	Allow Fast TCP on interface TMSES4. 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.

Used Ports

This table lists the port numbers used by the M262 Logic/Motion Controller services:

Service	Port numbers	Out-of-box / Default configuration ⁽¹⁾
Programming software	TCP 11740 UDP 1740	Yes
FTP over TLS	TCP 20, 21	Yes
HTTP ⁽²⁾	TCP 80	Yes
HTTPS	TCP 443	Yes
Modbus TCP	TCP 502	No
OPC UA	TCP 4840	No
Discovery	UDP 27126, 27127	Yes
Bonjour Discovery Protocol	UDP 5353	Yes
Web Services Dynamical Discovery	TCP 5357 UDP 3702	Yes
SNMP	UDP 161	No
NVL	UDP default value: 1202 (destination port) (Source port is dynamically assigned.)	No
EtherNet/IP	TCP 44818 UDP 2222, 44818	No
WebVisualisation	TCP 8080 (HTTP), 8089 (HTTPS)	No
DHCP/BOOTP	UDP 68	Yes
TFTP	UDP 69	No
NTP	UDP 123	No
Safe Logger ⁽³⁾	UDP 35021, 45000	Yes
Machine Assistant	UDP 45001, 45003, 45004	Yes

(1) Default configuration is used when a default project is loaded into the controller. The port status is exactly the same as the factory settings.

(2) HTTP is automatically redirected to HTTPS.

(3) **Safe Logger** ports are only applicable to TM262M• references.

TMSCO1 CANopen Communications Module

Introduction

This chapter describes the configuration of the TMSCO1 CANopen communications module.

Configuring the CANopen Interface

Introduction

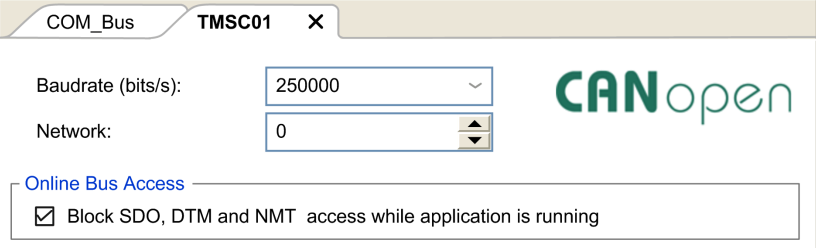
CANopen is an open industry-standard communication protocol and device profile specification (EN 50325-4) that is based on the Controller Area Network (CAN) protocol. The "Layer 7" CAN protocol was developed for embedded networking applications and defines communication and device functions for CAN-based systems.

CANopen supports both cyclic and event driven communication, allowing you to reduce bus load to a minimum but still maintain short reaction times.

You can set up your CANopen communications using a TMSCO1 module. This module connects to the communication bus (**COM_Bus**) on the left side of the controller, using the left bus connector interface. You can connect one TMSCO1 module. It must be the last module on the left side of the controller.

Configuring the CAN Bus

To configure the **CAN** bus of your controller, proceed in the following way:

Step	Action
1	Add a TMSCO1 module.
2	In the Devices tree , double-click TMSCO1 .
3	Configure the baudrate (by default: 250000 bits/s):  <p>NOTE: The Online Bus Access option allows you to block SDO, DTM, and NMT sending through the status screen.</p>

When connecting a DTM to a device using the network, the DTM communicates in parallel with the running application. The overall performance of the system is impacted and may overload the network, and therefore have consequences for the coherency of data across devices under control.

⚠ WARNING

UNINTENDED EQUIPMENT OPERATION

Place your machine or process in a state such that DTM communications does not impact its performance.

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

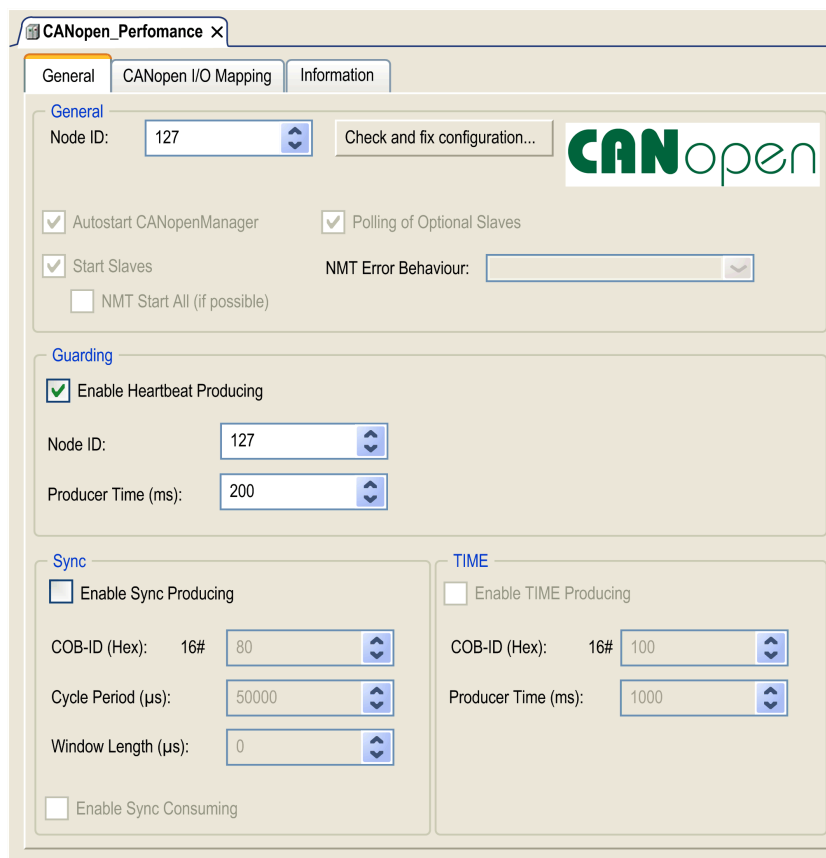
Adding a CANopen Performance Manager

Adding a **TMSCO1** module adds automatically the **CANopen Performance Manager** functionality to your controller.

Configuring a CANopen Performance Manager

To configure **CANopen Performance**, double-click **COM_Bus > TMSO1 > CANopen Performance** in the **Devices tree**.

This dialog box appears:



The **General** tab of the **CANopen_Performance** configuration dialog box is divided into four areas:

- **General:** General information containing node ID and enabled configuration options.
- **Guarding:** If **Enable Heartbeat Producing** is selected, guarding is enabled and the NMT master can verify the state of individual nodes. The heartbeat mechanism allows the network master to detect a loss of communication from the network slaves and the network slaves to react to a loss of communication from the master. The default setting is heartbeat producing at 200 ms.
- **Sync:** If **Enable Sync Producing** is selected, a specific event object is added. The **TMSC01_Sync** task is added to the **Application > Task Configuration** node in the **Applications tree**.

If you deselect the **Enable Sync Producing** option in this dialog box, the **TMSC01_Sync** task is automatically deleted from the **Applications Tree** in your program.

NOTE: Do not delete or change the **Type** or **External event** attributes of **TMSC01_Sync** tasks. If you do so, the software will detect an error when you attempt to build the application, and you will not be able to download it to the controller.

- **TIME:** Not editable.

CANopen Operating Limits

The CANopen master has the following operating limits:

- Maximum number of slave devices: 63
- Maximum number of Receive PDO (RPDO): 252
- Maximum number of Transmit PDO (TPDO): 252

▲ WARNING

UNINTENDED EQUIPMENT OPERATION

- Do not connect more than 63 CANopen slave devices to the controller
- Program your application to use 252 or fewer Transmit PDO (TPDO).
- Program your application to use 252 or fewer Receive PDO (RPDO).

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

CAN Bus Format

The CAN bus format is CAN2.0A for CANopen.

Glossary

A

ARP:

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

B

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.

C

CANopen:

An open industry-standard communication protocol and device profile specification (EN 50325-4).

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.

D

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.)

DNS:

(*domain name system*) The naming system for computers and devices connected to a LAN or the Internet.

DTM:

(*device type manager*) Classified into 2 categories:

- Device DTMs connect to the field device configuration components.
- CommDTMs connect to the software communication components.

The DTM provides a unified structure for accessing device parameters and configuring, operating, and diagnosing the devices. DTMs can range from a simple graphical user interface for setting device parameters to a highly sophisticated application capable of performing complex real-time calculations for diagnosis and maintenance purposes.

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.

EtherNet/IP:

(Ethernet industrial protocol) An open communications protocol for manufacturing automation solutions in industrial systems. EtherNet/IP is in a family of networks that implement the common industrial protocol at its upper layers. The supporting organization (ODVA) specifies EtherNet/IP to accomplish global adaptability and media independence.

Ethernet:

A physical and data link layer technology for LANs, also known as IEEE 802.3.

I

ICMP:

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

IGMP:

(Internet group management protocol) A communications protocol used by hosts and adjacent routers on IPv4 networks to establish multicast group memberships.

IP:

(Internet protocol) Part of the TCP/IP protocol family that tracks the Internet addresses of devices, routes outgoing messages, and recognizes incoming messages.

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.

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.

N

NMT:

(network management) CANopen protocols that provide services for network initialization, detected error control, and device status control.

P

PDO:

(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.

protocol:

A convention or standard definition that controls or enables the connection, communication, and data transfer between 2 computing system and devices.

R

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.

S

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).

T

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.

TPDO:

(transmit 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.

U

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).

Index

E

Ethernet	
EtherNet/IP device	20
Modbus TCP slave device	21
services	14
expansion modules	
adding	12
configuration	13

F

firewall	
configuration	28
default script file	28
script commands	29

P

protocols	14
Protocols	
IP	16

S

script commands	
firewall	29

Schneider Electric
35 rue Joseph Monier
92500 Rueil Malmaison
France

+ 33 (0) 1 41 29 70 00

www.se.com

As standards, specifications, and design change from time to time,
please ask for confirmation of the information given in this publication.

© 2024 Schneider Electric. All rights reserved.

EIO0000003691.05