Machine Automation Controller

NJ/NX-series

CPU Unit Built-in EtherCAT® Port

User's Manual

NX701-1□□□
NX502-1□□□
NX102-1□□□
NX102-90□□
NX1P2-1□□□□□
NX1P2-9□□□□□
NJ501-□□□□
NJ301-1□□□
NJ101-10□□
NJ101-90□□

CPU Unit

always in control

W505-E1-29

NOTE

1. All rights reserved. No part of this publication may be reproduced, stored in a retrieval system, or transmitted, in any form, or by any means, mechanical, electronic, photocopying, recording, or otherwise, without the prior written permission of OMRON.
2. No patent liability is assumed with respect to the use of the information contained herein. Moreover, because OMRON is constantly striving to improve its high-quality products, the information contained in this manual is subject to change without notice.
3. Every precaution has been taken in the preparation of this manual. Nevertheless, OMRON assumes no responsibility for errors or omissions.
   Neither is any liability assumed for damages resulting from the use of the information contained in this publication.

Trademarks

- Sysmac and SYSMAC are trademarks or registered trademarks of OMRON Corporation in Japan and other countries for OMRON factory automation products.
- Microsoft, Windows, Excel, Visual Basic, and Microsoft Edge are either registered trademarks or trademarks of Microsoft Corporation in the United States and other countries.
- EtherCAT® is registered trademark and patented technology, licensed by Beckhoff Automation GmbH, Germany.
- ODVA, CIP, CompoNet, DeviceNet, and EtherNet/IP are trademarks of ODVA.
- The SD and SDHC logos are trademarks of SD-3C, LLC.

Other company names and product names in this document are the trademarks or registered trademarks of their respective companies.

Copyrights

- Microsoft product screen shots used with permission from Microsoft.
- This product incorporates certain third party software. The license and copyright information associated with this software is available at http://www.fa.omron.co.jp/nj_info_e/.

Introduction

Thank you for purchasing an NJ/NX-series CPU Unit.

This manual contains information that is necessary to use the NJ/NX-series CPU Unit. Please read this manual and make sure you understand the functionality and performance of the NJ/NX-series CPU Unit before you attempt to use it in a control system.

Keep this manual in a safe place where it will be available for reference during operation.

Intended Audience

This manual is intended for the following personnel, who must also have knowledge of electrical systems (an electrical engineer or the equivalent).

- Personnel in charge of introducing FA systems.
- Personnel in charge of designing FA systems.
- Personnel in charge of installing and maintaining FA systems.
- Personnel in charge of managing FA systems and facilities.

For programming, this manual is intended for personnel who understand the programming language specifications in international standard IEC 61131-3 or Japanese standard JIS B 3503.

Applicable Products

This manual covers the following products.

- NX-series CPU Units
  - NX701-17□□
  - NX701-16□□
  - NX502-15□□
  - NX502-14□□
  - NX502-13□□
  - NX102-12□□
  - NX102-11□□
  - NX102-10□□
  - NX102-90□□
  - NX1P2-11□□□□
  - NX1P2-11□□□□1
  - NX1P2-10□□□□
  - NX1P2-10□□□□1
  - NX1P2-90□□□□
  - NX1P2-90□□□□1
  - NX1P2-9B□□□□
  - NX1P2-9B□□□□1

- NJ-series CPU Units
  - NJ501-□5□□
  - NJ501-□4□□
  - NJ501-□3□□
  - NJ301-12□□

- NJ301-11□□
- NJ101-10□□
- NJ101-90□□

Part of the specifications and restrictions for the CPU Units are given in other manuals. Refer to *Relevant Manuals* on page 3 and *Related Manuals* on page 22.

Relevant Manuals

The following table provides the relevant manuals for the NJ/NX-series CPU Units. Read all of the manuals that are relevant to your system configuration and application before you use the NJ/NX-series CPU Unit.

Most operations are performed from the Sysmac Studio Automation Software. Refer to the *Sysmac Studio Version 1 Operation Manual (Cat. No. W504)* for information on the Sysmac Studio.

Purpose of use	Manual																			
	Basic information																			
	NX-series CPU Unit Hardware User's Manual	NX-series NX502 CPU Unit Hardware User's Manual	NX-series NX102 CPU Unit Hardware User's Manual	NX-series NX1P2 CPU Unit Hardware User's Manual	NJ-series CPU Unit Hardware User's Manual	NJ/NX-series CPU Unit Software User's Manual	NX-series NX1P2 CPU Unit Built-in I/O and Option Board User's Manual	NJ/NX-series Instructions Reference Manual	NJ/NX-series CPU Unit Motion Control User's Manual	NJ/NX-series Motion Control Instructions Reference Manual	NJ/NX-series CPU Unit Built-in EtherCAT Port User's Manual	NJ/NX-series CPU Unit Built-in EtherNet/IP Port User's Manual	NJ/NX-series CPU Unit OPC UA User's Manual	NX-series CPU Unit FINS User's Manual	NJ/NX-series Database Connection CPU Units User's Manual	NJ-series SECS/GEM CPU Units User's Manual	NJ-series Robot Integrated CPU Unit User's Manual	NJ-series NJ Robotics CPU Unit User's Manual	NJ/NY-series NC Integrated Controller User's Manual	NJ/NX-series Troubleshooting Manual
Introduction to NX701 CPU Units	○																			
Introduction to NX502 CPU Units		○																		
Introduction to NX102 CPU Units			○																	
Introduction to NX1P2 CPU Units				○																
Introduction to NJ-series Controllers					○															
Setting devices and hardware	○	○	○	○	○															
Using motion control									○											
Using EtherCAT											○									
Using EtherNet/IP												○								
Using robot control for OMRON robots																	○			

Purpose of use	Manual																			
	Basic information																			
	NX-series CPU Unit Hardware User's Manual	NX-series NX502 CPU Unit Hardware User's Manual	NX-series NX102 CPU Unit Hardware User's Manual	NX-series NX1P2 CPU Unit Hardware User's Manual	NJ-series CPU Unit Hardware User's Manual	NJ/NX-series CPU Unit Software User's Manual	NX-series NX1P2 CPU Unit Built-in I/O and Option Board User's Manual	NJ/NX-series Instructions Reference Manual	NJ/NX-series CPU Unit Motion Control User's Manual	NJ/NX-series Motion Control Instructions Reference Manual	NJ/NX-series CPU Unit Built-in EtherCAT Port User's Manual	NJ/NX-series CPU Unit Built-in EtherNet/IP Port User's Manual	NJ/NX-series CPU Unit OPC UA User's Manual	NX-series CPU Unit FINS User's Manual	NJ/NX-series Database Connection CPU Units User's Manual	NJ-series SECS/GEM CPU Units User's Manual	NJ-series Robot Integrated CPU Unit User's Manual	NJ-series NJ Robotics CPU Unit User's Manual	NJ/NY-series NC Integrated Controller User's Manual	NJ/NX-series Troubleshooting Manual
Software settings						○														
Using motion control									○											
Using EtherCAT											○									
Using EtherNet/IP												○								
Using OPC UA													○							
Using FINS														○						
Using the database connection service															○					
Using the GEM Services																○				
Using robot control for OMRON robots																	○			
Using robot control by NJ Robotics function																		○		
Using numerical control																			○	
Using the NX1P2 CPU Unit functions							○													
Writing the user program						○		○												
Using motion control									○	○										
Using EtherCAT											○									
Using EtherNet/IP												○								
Using OPC UA													○							
Using FINS														○						
Using the database connection service															○					
Using the GEM Services																○				
Using robot control for OMRON robots																	○			
Using robot control by NJ Robotics function																		○		
Using numerical control																			○	
Programming error processing																				○
Using the NX1P2 CPU Unit functions							○													

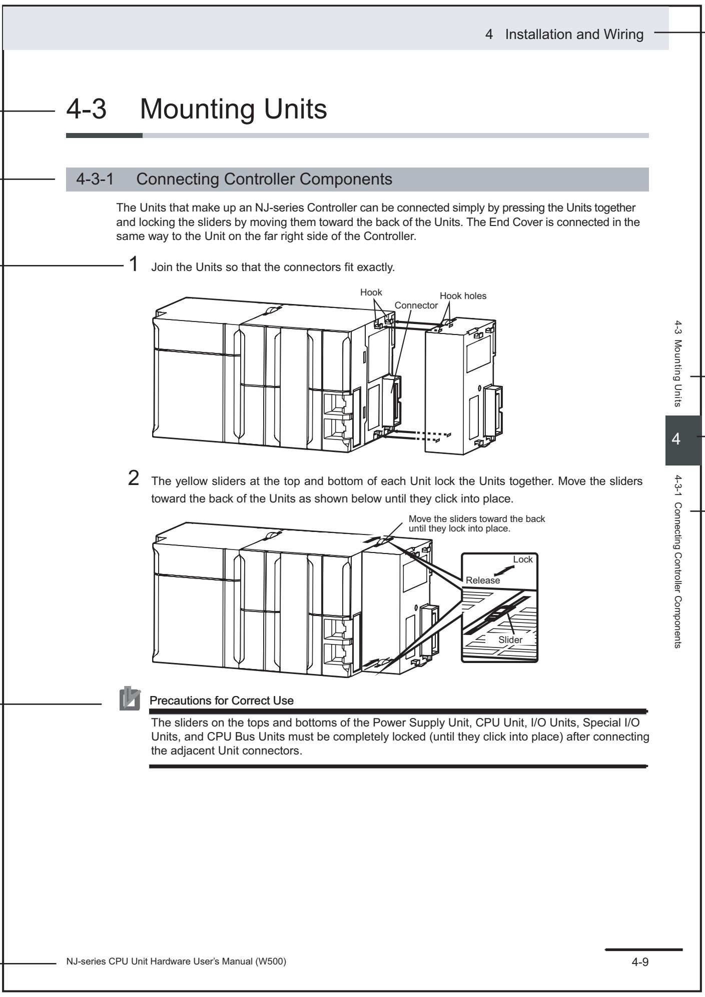
Purpose of use	Manual																			
	Basic information																			
	NX-series CPU Unit Hardware User's Manual	NX-series NX502 CPU Unit Hardware User's Manual	NX-series NX102 CPU Unit Hardware User's Manual	NX-series NX1P2 CPU Unit Hardware User's Manual	NJ-series CPU Unit Hardware User's Manual	NJ/NX-series CPU Unit Software User's Manual	NX-series NX1P2 CPU Unit Built-in I/O and Option Board User's Manual	NJ/NX-series Instructions Reference Manual	NJ/NX-series CPU Unit Motion Control User's Manual	NJ/NX-series Motion Control Instructions Reference Manual	NJ/NX-series CPU Unit Built-in EtherCAT Port User's Manual	NJ/NX-series CPU Unit Built-in EtherNet/IP Port User's Manual	NJ/NX-series CPU Unit OPC UA User's Manual	NX-series CPU Unit FINS User's Manual	NJ/NX-series Database Connection CPU Units User's Manual	NJ-series SECS/GEM CPU Units User's Manual	NJ-series Robot Integrated CPU Unit User's Manual	NJ-series NJ Robotics CPU Unit User's Manual	NJ/NY-series NC Integrated Controller User's Manual	NJ/NX-series Troubleshooting Manual
Testing operation and debugging						○														
Using motion control									○											
Using EtherCAT											○									
Using EtherNet/IP												○								
Using OPC UA													○							
Using FINS														○						
Using the database connection service															○					
Using the GEM Services																○				
Using robot control for OMRON robots																	○			
Using robot control by NJ Robotics function																		○		
Using numerical control																			○	
Using the NX1P2 CPU Unit functions							○													
Learning about error management and corrections[*1]													△	△	△	△	△	△	△	○
Maintenance	○	○	○	○	○															
Using motion control									○											
Using EtherCAT											○									
Using EtherNet/IP												○								

*1. Refer to the *NJ/NX-series Troubleshooting Manual (Cat. No. W503)* for the error management concepts and the error items. However, refer to the manuals that are indicated with triangles for details on errors corresponding to the products with the manuals that are indicated with triangles.

Manual Structure

Page Structure

The following page structure is used in this manual.



4 Installation and Wiring

4-3 Mounting Units

4-3-1 Connecting Controller Components

The Units that make up an NJ-series Controller can be connected simply by pressing the Units together and locking the sliders by moving them toward the back of the Units. The End Cover is connected in the same way to the Unit on the far right side of the Controller.

1 Join the Units so that the connectors fit exactly.

2 The yellow sliders at the top and bottom of each Unit lock the Units together. Move the sliders toward the back of the Units as shown below until they click into place.

Move the sliders toward the back until they lock into place.
Lock
Release
Slider

Precautions for Correct Use

The sliders on the tops and bottoms of the Power Supply Unit, CPU Unit, I/O Units, Special I/O Units, and CPU Bus Units must be completely locked (until they click into place) after connecting the adjacent Unit connectors.

4-3 Mounting Units

4

4-3-1 Connecting Controller Components

NJ-series CPU Unit Hardware User's Manual (W500)

4-9

Level 1 heading

Level 2 heading

Level 3 heading

Gives the current headings.

Page tab

Gives the number of the main section.

Level 2 heading

Level 3 heading

A step in a procedure

Indicates a procedure.

Special information

Icons indicate precautions, additional information, or reference information.

Manual name

This illustration is provided only as a sample. It may not literally appear in this manual.

Special Information

Special information in this manual is classified as follows:

Precautions for Safe Use

Precautions on what to do and what not to do to ensure safe usage of the product.

Precautions for Correct Use

Precautions on what to do and what not to do to ensure proper operation and performance.

Additional Information

Additional information to read as required.
This information is provided to increase understanding or make operation easier.

Version Information

Information on differences in specifications and functionality for Controller with different unit versions and for different versions of the Sysmac Studio is given.

Precaution on Terminology

In this manual, "download" refers to transferring data from the Sysmac Studio to the physical Controller and "upload" refers to transferring data from the physical Controller to the Sysmac Studio.
For the Sysmac Studio, "synchronization" is used to both "upload" and "download" data. Here, "synchronize" means to automatically compare the data for the Sysmac Studio on the computer with the data in the physical Controller and transfer the data in the direction that is specified by the user.

Sections in this Manual

CONTENTS

Section 1 Introduction

Section 2 Part Names and Slave Settings

Section 3 EtherCAT Communications

Section 4 EtherCAT Network Wiring

Section 5 Setting Up EtherCAT Communications with the Sysmac Studio

Section 6 Process Data Communications and SDO Communications

Section 7 System-defined Variables Related to the Built-in EtherCAT Port

Section 8 Example of Operations for EtherCAT Communications

Section 9 Troubleshooting

Appendices

Index

Terms and Conditions Agreement

Warranty, Limitations of Liability

Warranties

- **Exclusive Warranty**

Omron's exclusive warranty is that the Products will be free from defects in materials and workmanship for a period of twelve months from the date of sale by Omron (or such other period expressed in writing by Omron). Omron disclaims all other warranties, express or implied.

- **Limitations**

OMRON MAKES NO WARRANTY OR REPRESENTATION, EXPRESS OR IMPLIED, ABOUT NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE OF THE PRODUCTS. BUYER ACKNOWLEDGES THAT IT ALONE HAS DETERMINED THAT THE PRODUCTS WILL SUITABLY MEET THE REQUIREMENTS OF THEIR INTENDED USE.

Omron further disclaims all warranties and responsibility of any type for claims or expenses based on infringement by the Products or otherwise of any intellectual property right.

- **Buyer Remedy**

Omron's sole obligation hereunder shall be, at Omron's election, to (i) replace (in the form originally shipped with Buyer responsible for labor charges for removal or replacement thereof) the non-complying Product, (ii) repair the non-complying Product, or (iii) repay or credit Buyer an amount equal to the purchase price of the non-complying Product; provided that in no event shall Omron be responsible for warranty, repair, indemnity or any other claims or expenses regarding the Products unless Omron's analysis confirms that the Products were properly handled, stored, installed and maintained and not subject to contamination, abuse, misuse or inappropriate modification. Return of any Products by Buyer must be approved in writing by Omron before shipment. Omron Companies shall not be liable for the suitability or unsuitability or the results from the use of Products in combination with any electrical or electronic components, circuits, system assemblies or any other materials or substances or environments. Any advice, recommendations or information given orally or in writing, are not to be construed as an amendment or addition to the above warranty.

See http://www.omron.com/global/ or contact your Omron representative for published information.

Limitation on Liability; Etc

OMRON COMPANIES SHALL NOT BE LIABLE FOR SPECIAL, INDIRECT, INCIDENTAL, OR CONSEQUENTIAL DAMAGES, LOSS OF PROFITS OR PRODUCTION OR COMMERCIAL LOSS IN ANY

WAY CONNECTED WITH THE PRODUCTS, WHETHER SUCH CLAIM IS BASED IN CONTRACT, WARRANTY, NEGLIGENCE OR STRICT LIABILITY.

Further, in no event shall liability of Omron Companies exceed the individual price of the Product on which liability is asserted.

Application Considerations

Suitability of Use

Omron Companies shall not be responsible for conformity with any standards, codes or regulations which apply to the combination of the Product in the Buyer's application or use of the Product. At Buyer's request, Omron will provide applicable third party certification documents identifying ratings and limitations of use which apply to the Product. This information by itself is not sufficient for a complete determination of the suitability of the Product in combination with the end product, machine, system, or other application or use. Buyer shall be solely responsible for determining appropriateness of the particular Product with respect to Buyer's application, product or system. Buyer shall take application responsibility in all cases.

NEVER USE THE PRODUCT FOR AN APPLICATION INVOLVING SERIOUS RISK TO LIFE OR PROPERTY OR IN LARGE QUANTITIES WITHOUT ENSURING THAT THE SYSTEM AS A WHOLE HAS BEEN DESIGNED TO ADDRESS THE RISKS, AND THAT THE OMRON PRODUCT(S) IS PROPERLY RATED AND INSTALLED FOR THE INTENDED USE WITHIN THE OVERALL EQUIPMENT OR SYSTEM.

Programmable Products

Omron Companies shall not be responsible for the user's programming of a programmable Product, or any consequence thereof.

Disclaimers

Performance Data

Data presented in Omron Company websites, catalogs and other materials is provided as a guide for the user in determining suitability and does not constitute a warranty. It may represent the result of Omron's test conditions, and the user must correlate it to actual application requirements. Actual performance is subject to the Omron's Warranty and Limitations of Liability.

Change in Specifications

Product specifications and accessories may be changed at any time based on improvements and other reasons. It is our practice to change part numbers when published ratings or features are changed, or when significant construction changes are made. However, some specifications of the Product may be changed without any notice. When in doubt, special part numbers may be assigned to fix or establish key specifications for your application. Please consult with your Omron's representative at any time to confirm actual specifications of purchased Product.

Errors and Omissions

Information presented by Omron Companies has been checked and is believed to be accurate; however, no responsibility is assumed for clerical, typographical or proofreading errors or omissions.

Statement of security responsibilities for assumed use cases and against threats

OMRON SHALL NOT BE RESPONSIBLE AND/OR LIABLE FOR ANY LOSS, DAMAGE, OR EXPENSES DIRECTLY OR INDIRECTLY RESULTING FROM THE INFECTION OF OMRON PRODUCTS, ANY SOFTWARE INSTALLED THEREON OR ANY COMPUTER EQUIPMENT, COMPUTER PROGRAMS, NETWORKS, DATABASES OR OTHER PROPRIETARY MATERIAL CONNECTED THERETO BY DISTRIBUTED DENIAL OF SERVICE ATTACK, COMPUTER VIRUSES, OTHER TECHNOLOGICALLY HARMFUL MATERIAL AND/OR UNAUTHORIZED ACCESS.

It shall be the users sole responsibility to determine and use adequate measures and checkpoints to satisfy the users particular requirements for (i) antivirus protection, (ii) data input and output, (iii) maintaining a means for reconstruction of lost data, (iv) preventing Omron Products and/or software installed thereon from being infected with computer viruses and (v) protecting Omron Products from unauthorized access.

Safety Precautions

Refer to the following manuals for safety precautions.

- *NX-series CPU Unit Hardware User's Manual (Cat. No. W535)*
- *NX-series NX502 CPU Unit Hardware User’s Manual (Cat. No. W629)*
- *NX-series NX102 CPU Unit Hardware User's Manual (Cat. No. W593)*
- *NX-series NX1P2 CPU Unit Hardware User's Manual (Cat. No. W578)*
- *NJ-series CPU Unit Hardware User’s Manual (Cat No. W500)*

Precautions for Safe Use

Refer to the following manuals for precautions for safe use.

- *NX-series CPU Unit Hardware User's Manual (Cat. No. W535)*
- *NX-series NX502 CPU Unit Hardware User’s Manual (Cat. No. W629)*
- *NX-series NX102 CPU Unit Hardware User's Manual (Cat. No. W593)*
- *NX-series NX1P2 CPU Unit Hardware User's Manual (Cat. No. W578)*
- *NJ-series CPU Unit Hardware User’s Manual (Cat No. W500)*

Precautions for Correct Use

Refer to the following manuals for precautions for correct use.

- *NX-series CPU Unit Hardware User's Manual (Cat. No. W535)*
- *NX-series NX502 CPU Unit Hardware User’s Manual (Cat. No. W629)*
- *NX-series NX102 CPU Unit Hardware User's Manual (Cat. No. W593)*
- *NX-series NX1P2 CPU Unit Hardware User's Manual (Cat. No. W578)*
- *NJ-series CPU Unit Hardware User’s Manual (Cat No. W500)*

Regulations and Standards

Refer to the following manuals for regulations and standards.

- *NX-series CPU Unit Hardware User's Manual (Cat. No. W535)*
- *NX-series NX502 CPU Unit Hardware User’s Manual (Cat. No. W629)*
- *NX-series NX102 CPU Unit Hardware User's Manual (Cat. No. W593)*
- *NX-series NX1P2 CPU Unit Hardware User's Manual (Cat. No. W578)*
- *NJ-series CPU Unit Hardware User’s Manual (Cat No. W500)*

Versions

Hardware revisions and unit versions are used to manage the hardware and software in NJ/NX-series Units and EtherCAT slaves. The hardware revision or unit version is updated each time there is a change in hardware or software specifications. Even when two Units or EtherCAT slaves have the same model number, they will have functional or performance differences if they have different hardware revisions or unit versions.

Refer to the following manuals for versions.

- *NX-series CPU Unit Hardware User's Manual (Cat. No. W535)*
- *NX-series NX502 CPU Unit Hardware User’s Manual (Cat. No. W629)*
- *NX-series NX102 CPU Unit Hardware User's Manual (Cat. No. W593)*
- *NX-series NX1P2 CPU Unit Hardware User's Manual (Cat. No. W578)*
- *NJ-series CPU Unit Hardware User’s Manual (Cat No. W500)*

Unit Versions of CPU Units and Sysmac Studio Versions

The functions that are supported depend on the unit version of the NJ/NX-series CPU Unit. The version of Sysmac Studio that supports the functions that were added for an upgrade is required to use those functions.
Refer to the *NJ/NX-series CPU Unit Software User’s Manual* (Cat. No. W501) for the relationship between the unit versions of CPU Units and the Sysmac Studio versions, and for the functions that are supported by each unit version.

Related Manuals

The followings are the manuals related to this manual. Use these manuals for reference.

Manual name	Cat. No.	Model numbers	Application	Description
NX-series CPU Unit Hardware User's Manual	W535	NX701-□□□□	Learning the basic specifications of the NX701 CPU Units, including introductory information, designing, installation, and maintenance. Mainly hardware information is provided.	An introduction to the entire NX701 system is provided along with the following information on the CPU Unit. • Features and system configuration • Introduction • Part names and functions • General specifications • Installation and wiring • Maintenance and inspection
NX-series NX502 CPU Unit Hardware User’s Manual	W629	NX502-□□□□	Learning the basic specifications of the NX502 CPU Units, including introductory information, designing, installation, and maintenance. Mainly hardware information is provided.	An introduction to the entire NX502 system is provided along with the following information on the CPU Unit. • Features and system configuration • Introduction • Part names and functions • General specifications • Installation and wiring • Maintenance and inspection
NX-series NX102 CPU Unit Hardware User’s Manual	W593	NX102-□□□□	Learning the basic specifications of the NX102 CPU Units, including introductory information, designing, installation, and maintenance. Mainly hardware information is provided.	An introduction to the entire NX102 system is provided along with the following information on the CPU Unit. • Features and system configuration • Introduction • Part names and functions • General specifications • Installation and wiring • Maintenance and inspection
NX-series NX1P2 CPU Unit Hardware User’s Manual	W578	NX1P2-□□□□	Learning the basic specifications of the NX1P2 CPU Units, including introductory information, designing, installation, and maintenance. Mainly hardware information is provided.	An introduction to the entire NX1P2 system is provided along with the following information on the CPU Unit. • Features and system configuration • Introduction • Part names and functions • General specifications • Installation and wiring • Maintenance and inspection
NJ-series CPU Unit Hardware User's Manual	W500	NJ501-□□□□ NJ301-□□□□ NJ101-□□□□	Learning the basic specifications of the NJ-series CPU Units, including introductory information, designing, installation, and maintenance. Mainly hardware information is provided.	An introduction to the entire NJ-series system is provided along with the following information on the CPU Unit. • Features and system configuration • Introduction • Part names and functions • General specifications • Installation and wiring • Maintenance and inspection

Manual name	Cat. No.	Model numbers	Application	Description
NJ/NX-series CPU Unit Software User's Manual	W501	NX701-□□□□ NX502-□□□□ NX102-□□□□ NX1P2-□□□□ NJ501-□□□□ NJ301-□□□□ NJ101-□□□□	Learning how to program and set up an NJ/NX-series CPU Unit. Mainly software information is provided.	The following information is provided on a Controller built with an NJ/NX-series CPU Unit. • CPU Unit operation • CPU Unit features • Initial settings • Programming based on IEC 61131-3 language specifications
NX-series NX1P2 CPU Unit Built-in I/O and Option Board User's Manual	W579	NX1P2-□□□□	Learning about the details of functions only for an NX-series NX1P2 CPU Unit and an introduction of functions for an NJ/NX-series CPU Unit.	Of the functions for an NX1P2 CPU Unit, the following information is provided. • Built-in I/O • Serial Communications Option Boards • Analog I/O Option Boards An introduction of following functions for an NJ/NX-series CPU Unit is also provided. • Motion control functions • EtherNet/IP communications functions • EtherCAT communications functions
NJ/NX-series Instructions Reference Manual	W502	NX701-□□□□ NX502-□□□□ NX102-□□□□ NX1P2-□□□□ NJ501-□□□□ NJ301-□□□□ NJ101-□□□□	Learning detailed specifications on the basic instructions of an NJ/NX-series CPU Unit.	The instructions in the instruction set (IEC 61131-3 specifications) are described.
NJ/NX-series CPU Unit Motion Control User's Manual	W507	NX701-□□□□ NX502-□□□□ NX102-□□□□ NX1P2-□□□□ NJ501-□□□□ NJ301-□□□□ NJ101-□□□□	Learning about motion control settings and programming concepts.	The settings and operation of the CPU Unit and programming concepts for motion control are described.
NJ/NX-series Motion Control Instructions Reference Manual	W508	NX701-□□□□ NX502-□□□□ NX102-□□□□ NX1P2-□□□□ NJ501-□□□□ NJ301-□□□□ NJ101-□□□□	Learning about the specifications of the motion control instructions.	The motion control instructions are described.
NJ/NX-series CPU Unit Built-in EtherCAT® Port User's Manual	W505	NX701-□□□□ NX502-□□□□ NX102-□□□□ NX1P2-□□□□ NJ501-□□□□ NJ301-□□□□ NJ101-□□□□	Using the built-in EtherCAT port on an NJ/NX-series CPU Unit.	Information on the built-in EtherCAT port is provided. This manual provides an introduction and provides information on the configuration, features, and setup.
NJ/NX-series CPU Unit Built-in EtherNet/IP™ Port User's Manual	W506	NX701-□□□□ NX502-□□□□ NX102-□□□□ NX1P2-□□□□ NJ501-□□□□ NJ301-□□□□ NJ101-□□□□	Using the built-in EtherNet/IP port on an NJ/NX-series CPU Unit.	Information on the built-in EtherNet/IP port is provided. Information is provided on the basic setup, tag data links, and other features.
NJ/NX-series CPU Unit OPC UA User's Manual	W588	NX701-□□□□ NX502-□□□□ NX102-□□□□ NJ501-1□00	Using the OPC UA.	Describes the OPC UA.

Manual name	Cat. No.	Model numbers	Application	Description
NX-series CPU Unit FINS Function User’s Manual	W596	NX701-□□20 NX502-□□□□ NX102-□□□□	Using the FINS function of an NX-series CPU Unit.	Describes the FINS function of an NX-series CPU Unit.
NJ/NX-series Database Connection CPU Units User’s Manual	W527	NX701-□□20 NX502-□□□□ NX102-□□20 NJ501-□□20 NJ101-□□20	Using the database connection service with NJ/NX-series Controllers.	Describes the database connection service.
NJ-series SECS/GEM CPU Units User’s Manual	W528	NJ501-1340	Using the GEM Services with NJ-series Controllers.	Provides information on the GEM Services.
NJ-series Robot Integrated CPU Unit User’s Manual	O037	NJ501-R□□□	Using the NJ-series Robot Integrated CPU Unit.	Describes the settings and operation of the CPU Unit and programming concepts for OMRON robot control.
Sysmac Studio Robot Integrated System Building Function with Robot Integrated CPU Unit Operation Manual	W595	SYSMAC-SE2□□□ SYSMAC-SE200D-64	Learning about the operating procedures and functions of the Sysmac Studio to configure Robot Integrated System using Robot Integrated CPU Unit.	Describes the operating procedures of the Sysmac Studio for Robot Integrated CPU Unit.
Sysmac Studio Robot Integrated System Building Function with IPC Application Controller Operation Manual	W621	SYSMAC-SE2□□□ SYSMAC-SE200D-64	Learning about the operating procedures and functions of the Sysmac Studio to configure Robot Integrated System using IPC Application Controller.	Describes the operating procedures of the Sysmac Studio for IPC Application Controller.
Sysmac Studio 3D Simulation Function Operation Manual	W618	SYSMAC-SE2□□□ SYSMAC-SA4□□□-64	Learning about an outline of the 3D simulation function of the Sysmac Studio and how to use the function.	Describes an outline, execution procedures, and operating procedures for the 3D simulation function of the Sysmac Studio.
NJ-series NJ Robotics CPU Unit User’s Manual	W539	NJ501-4□□□ NJ501-R□□□	Controlling robots with NJ-series CPU Units.	Describes the functionality to control robots.
NJ/NY-series NC Integrated Controller User’s Manual	O030	NJ501-5300 NY532-5400	Performing numerical control with NJ/NY-series Controllers.	Describes the functionality to perform the numerical control.
NJ/NY-series G code Instructions Reference Manual	O031	NJ501-5300 NY532-5400	Learning about the specifications of the G code/M code instructions.	The G code/M code instructions are described.
NJ/NX-series Troubleshooting Manual	W503	NX701-□□□□ NX502-□□□□ NX102-□□□□ NX1P2-□□□□ NJ501-□□□□ NJ301-□□□□ NJ101-□□□□	Learning about the errors that may be detected in an NJ/NX-series Controller.	Concepts on managing errors that may be detected in an NJ/NX-series Controller and information on individual errors are described.
Sysmac Studio Version 1 Operation Manual	W504	SYSMAC-SE2□□□	Learning about the operating procedures and functions of the Sysmac Studio.	Describes the operating procedures of the Sysmac Studio.

Manual name	Cat. No.	Model numbers	Application	Description
CNC Operator Operation Manual	O032	SYSMAC-RTNC0□□□D	Learning an introduction of the CNC Operator and how to use it.	An introduction of the CNC Operator, installation procedures, basic operations, connection operations, and operating procedures for main functions are described.
NX-series EtherCAT® Coupler Unit User's Manual	W519	NX-ECC□□□	Learning how to use the NX-series EtherCAT Coupler Unit and EtherCAT Slave Terminals.	The following items are described: the overall system and configuration methods of an EtherCAT Slave Terminal (which consists of an NX-series EtherCAT Coupler Unit and NX Units), and information on hardware, setup, and functions to set up, control, and monitor NX Units through EtherCAT.
NX-series Safety Control Unit User's Manual	Z930	NX-SL□□□□ NX-SI□□□□ NX-SO□□□□	Learning how to use NX-series Safety Control Units.	Describes the hardware, setup methods, and functions of the NX-series Safety Control Units.

Revision History

A manual revision code appears as a suffix to the catalog number on the front and back covers of the manual.

Cat. No. W505-E1-29

Revision code

Revision code	Date	Revised content
01	July 2011	Original production
02	March 2012	Added information on the NJ301-□□□□ and corrected mistakes.
03	May 2012	• Added information on functional support for unit version 1.02 and later of the CPU Units. • Corrected mistakes.
04	August 2012	• Added information on functional support for unit version 1.03 and later of the CPU Units. • Corrected mistakes.
05	February 2013	• Added information on functional support for unit version 1.04 and later of the CPU Units. • Corrected mistakes.
06	April 2013	• Added information on functional support for unit version 1.05 and later of the CPU Units. • Corrected mistakes.
07	June 2013	• Added information on functional support for unit version 1.06 and later of the CPU Units. • Corrected mistakes.
08	September 2013	• Added information on functional support for unit version 1.07 and later of the CPU Units. • Corrected mistakes.
09	December 2013	• Added information on functional support for unit version 1.08 and later of the CPU Units. • Corrected mistakes.
10	July 2014	• Corrected mistakes.
11	January 2015	• Added information on functional support for unit version 1.10 and later of the CPU Units. • Corrected mistakes.
12	April 2015	• Added information on the NX701-□□□□. • Added information on the NJ101-□□□□. • Corrected mistakes.
13	April 2016	• Added information on functional support for unit version 1.11 and later of the CPU Units. • Corrected mistakes.
14	October 2016	• Added information on the NX1P2-□□□□□□. • Corrected mistakes.
15	April 2017	• Corrected mistakes.

Revision code	Date	Revised content
16	October 2017	• Corrected mistakes.
17	April 2018	• Corrected mistakes.
18	April 2018	• Added information on the NX102-□□□□. • Collected descriptions on event codes and errors to the *NJ/NX-series Troubleshooting Manual*. • Corrected mistakes.
19	July 2018	Corrected mistakes.
20	October 2018	Corrected mistakes.
21	July 2019	• Added information on functional support for unit version 1.40 of the CPU Units. • Corrected mistakes.
22	July 2019	Corrected mistakes.
23	October 2019	• Added information on the NX1P2-9B□□□□. • Corrected mistakes.
24	August 2020	• Added information on the NJ501-R□00. • Corrected mistakes.
25	October 2020	• Added information on functional support for unit version 1.42 of the CPU Units. • Corrected mistakes.
26	July 2021	• Added information on the functions supported by unit version 1.46 of the NX102-□□00, NX1P2-□□□□□, NJ501-1□00, NJ501-R□□□, NJ301-□□□□, and NJ101-□□00. • Added information on the functions supported by unit version 1.26 of the NJ501-□□20, NJ501-1340, NJ501-5300, NJ501-4□□□, NJ101-1□20, and NX701-□□□□. • Added information on the functions supported by unit version 1.37 of the NX102-□□20. • Corrected mistakes.
27	October 2021	• Added information related to the hardware revision A of the NX701-□□□□.
28	April 2022	• Added information to Terms and Conditions Agreement.
29	April 2023	• Added information on the NX502-1□□□. • Corrected mistakes.

1 Introduction

This section provides an overview of EtherCAT communications, describes the system configuration and specifications, and provides operating procedures.

1-1 Introduction to EtherCAT

EtherCAT (Ethernet Control Automation Technology) is a high-performance industrial network system that enables faster and more efficient communications based on Ethernet.
Each node achieves a short communications cycle time by transmitting Ethernet frames at high speed.
Furthermore, even though EtherCAT is a unique protocol, it offers excellent general-purpose applicability. For example, you can use Ethernet cables because EtherCAT utilizes standard Ethernet technology for the physical layer. And the effectiveness of EtherCAT can be fully utilized not only in large control systems that require high processing speeds and system integrity, but also in small and medium control systems.

1-1-1 EtherCAT Features

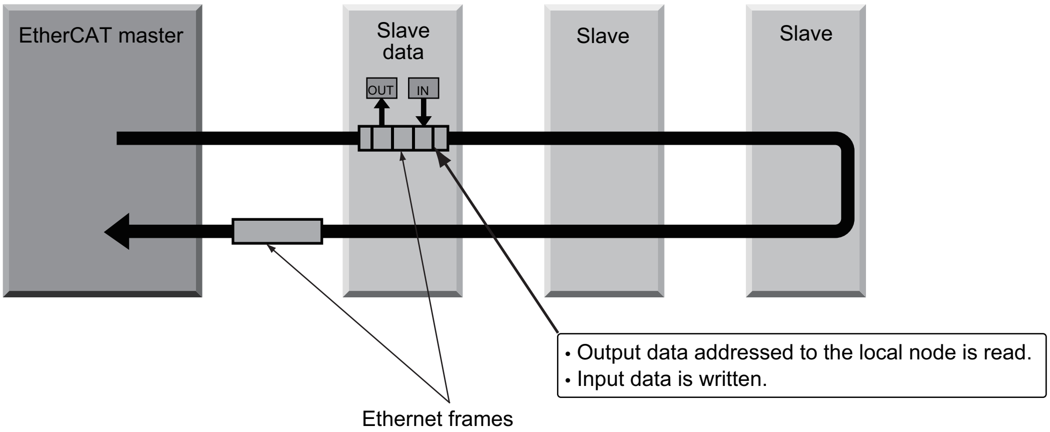
EtherCAT provides the following features.

High-speed Communications at 100 Mbps

The I/O response time from signal input to signal output has been significantly reduced. By fully utilizing the optimized Ethernet frame bandwidth to transmit data using a high-speed repeat method, it is possible to efficiently transmit a wide variety of data.

1-1-2 EtherCAT Communications

EtherCAT does not send data to individual slave nodes on the network, instead, it passes Ethernet frames through all of the slave nodes.
When frame passes through a slave node, the slave node reads and writes data in the areas allocated to it in the frames in a few nanoseconds.
The Ethernet frames transmitted by the EtherCAT master pass through all EtherCAT slaves without stopping. The last slave returns all of the frames, which again pass through all of the slaves before returning to the EtherCAT master.
This mechanism ensures high speed and realtime data transmission.



1-1-3 EtherCAT Features for NJ/NX-series CPU Units

EtherCAT in the NJ/NX-series CPU Units has the following features.

Synchronization of the CPU Unit Processing Period and the EtherCAT Communications Cycle

The period of sequence processing and motion processing in the CPU Unit matches the process data communications cycle of EtherCAT.

This enables high-precision sequence control and motion control with a stable fixed period.

Also, with the NX701 CPU Unit, slaves to be synchronized can be divided into two groups with different process data communications cycles. This means that slaves are separately processed in one group that needs high-speed communications and the other that needs not.

Accessing Data with Device Variables without Considering Addresses

EtherCAT slaves are accessed using device variables that are assigned to the I/O ports of the EtherCAT slaves. Various types of data in Servo Drive and the encoder input slaves are accessed using structure-type Axis Variables prepared in advance.

This enables access to slaves on EtherCAT without regard to addresses.

Optimum Functionality and Ease of Operation Based on Unified Specifications

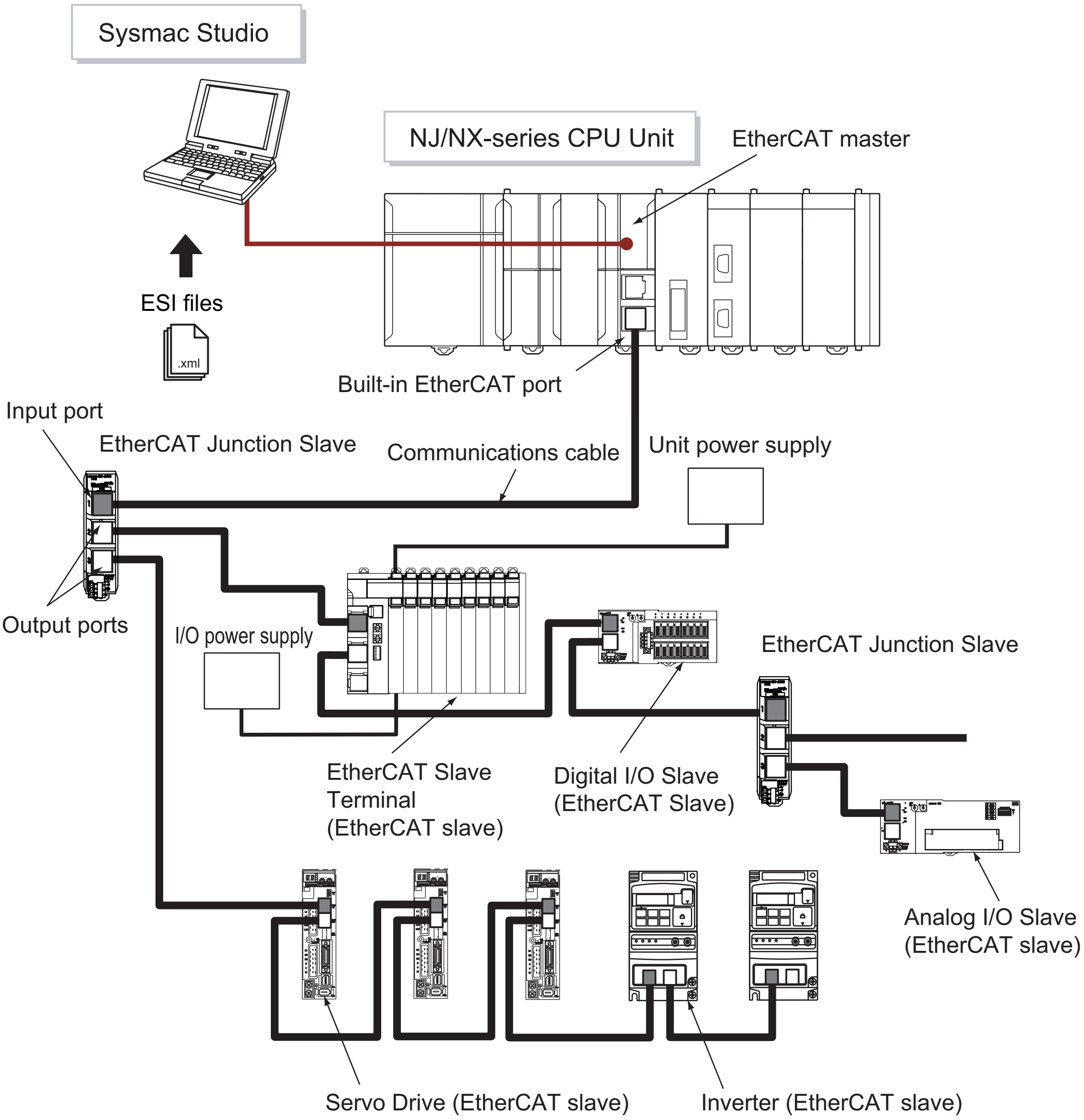
You can use the NJ/NX-series Machine Automation Controllers together with Sysmac devices*1 and the Sysmac Studio Automation Software to achieve optimum functionality and ease of operation.

*1. “Sysmac devices” is a generic name for EtherCAT slaves and other OMRON control components that were designed with the same communications and user interface specifications.

1-2 System Configuration and Configuration Devices

1-2-1 System Configuration

The EtherCAT network configuration and configuration devices are shown below.



Outlines of the configuration devices are given below.

● EtherCAT Master

The EtherCAT master manages the network, monitors the status of slaves, and exchanges I/O data with slaves. There is one output port.

● Output Port

The output port transmits EtherCAT communications data to other devices. When you connect the output port to another device, always connect it to the input port on the other device. Normal communications will not be possible if you connect to the output port on another device.

● Input Port

This port is used to input EtherCAT communications data. Always connect it to the output port on another device. Normal communications will not be possible if you connect to the input port on another device.

● EtherCAT Slaves

Each EtherCAT slave outputs the output data that it received from the EtherCAT master through the EtherCAT network. It also sends input data to the EtherCAT master through the EtherCAT network.

- Each slave has one input port and at least one output port.
- In the EtherCAT network, you can use the salves and Junction Slaves that are listed below in total up to the maximum number of slaves.
- Assign node addresses to the slaves and Junction Slaves. The node address can be any value within the settable node address range, and can be assigned without any distinction between the slaves and Junction Slaves. But any node address should not be used more than once.
- Some slaves can operate as only synced slave or as only non-synced slave. Some slaves can select to operate the synced slave or the non-synced slave.
  - Non-synced Slaves
    For these slaves, the distributed clock setting is disabled or selected **---**.
  - Synced Slaves
    Perform the clock synchronization to communicate between the EtherCAT master and slaves based on the Distributed Clock (DC). This applies for the slaves that the distributed clock setting is enabled.

Note 1. Refer to *1-3 Specifications of Built-in EtherCAT Port* on page 1-10 for details on the maximum number of slaves and the settable node address range.

Note 2. The slaves are synchronized between synced slaves even when multiple non-synced slaves and Junction Slaves are connected.

The following table lists some of the OMRON EtherCAT slaves that are available.

Name	Type	Model	Synced /Non-synced Slaves	Reference Clock*1	Assigning an axis
Digital I/O Slaves	Slaves with screw terminals and 2-tier terminal block	GX-□D16□1/OC1601	Non-synced Slaves	Exist	Not possible
	Slaves with screw terminals and 3-tier terminal block	GX-ID16□2/OD16□2/MD16□2	Non-synced Slaves	Exist	Not possible
	Slaves with e-CON connectors	GX-□D16□8/□D32□8	Non-synced Slaves	Exist	Not possible
Analog I/O Slaves	Slaves with screw terminals and 2-tier terminal block	GX-AD0471/DA0271	Non-synced Slaves	Exist	Not possible

Name	Type	Model	Synced /Non-synced Slaves	Reference Clock[1]	Assigning an axis
NX-series EtherCAT Coupler Unit	NX Series	NX-ECC201[2] NX-ECC202[3] NX-ECC203[4]	Synced or Non-synced Slaves[5]	Exist	Not possible (Some of the connected NX Units can be assigned to an axis.)
Multifunctional, Compact Inverters	MX2 Series	3G3MX2 with EtherCAT Communications Unit 3G3AX-MX2-ECT	Non-synced Slaves	Exist	Not possible
Advanced General-purpose Inverters	RX Series	3G3RX-V1 with EtherCAT Communications Unit 3G3AX-RX-ECT	Non-synced Slaves	Exist	Not possible
AC Servo Drive	G5-series Servo Drive with EtherCAT communications	R88M-K/R88D-KN□-ECT	Synced Slaves	Exist	Possible
Linear Servo Drives	G5-series Linear Servo Drives with EtherCAT communications	R88L-EC/R88D-KN□-ECT-L	Synced Slaves	Exist	Possible
Encoder Input Slaves	Slave with 3-tier terminal block	GX-EC0211/EC0241	Synced Slaves	Exist	Possible
Vision Sensors	FH Series	FH-1□□□ FH-3□□□	Synced Slaves	Exist	Not possible
Specialized Vision Sensors for Positioning	FQ-M-series Sensors with EtherCAT Communications	FQ-MS12□(-□)-ECT	Synced Slaves	Exist	Not possible
	FZM1-series Vision Sensors	FZM1-35□-ECT	Synced Slaves	Exist	Not possible
Digital Sensor Communications Unit	E3NW Series	E3NW-ECT	Synced or Non-synced Slaves[5]	Exist	Not possible
Fiber Sensor Communications Unit	E3X Series	E3X-ECT	Synced or Non-synced Slaves[5]	Exist	Not possible
Confocal Fiber Type Displacement Sensors	ZW Series	ZW-CE1□	Synced Slaves	Exist	Not possible

[1] Shows whether or not the slave provides a reference clock. A slave or a master that provides a reference clock is required for a system configuration that enables synchronization between slaves based on a distributed clock (DC). When you connect a slave that supports the reference clock, you must meet some conditions. Refer to *4-1-4 Precautions for Using Junction Slaves* on page 4-10.

[2] A CPU Unit with unit version 1.05 or later and Sysmac Studio version 1.06 or higher are required.

[3] A CPU Unit with unit version 1.07 or later and Sysmac Studio version 1.08 or higher are required.

[4] A CPU Unit with unit version 1.07 or later and Sysmac Studio version 1.11 or higher are required.

[5] Either can be set in the distributed clock setting for slaves.

● EtherCAT Junction Slave

This is a special slave for branching EtherCAT network wiring. It may be omitted as "Junction Slave" in this manual below.

- Cascade connections are possible within the range of node addresses that the EtherCAT master can handle.

- Each Junction Slave has one input port and more than one output port. The output ports on each Junction Slave can be connected to another Junction Slave or other EtherCAT slaves.
- The Junction Slave supports the reference clock that is required for a system configuration that enables synchronization between slaves based on a distributed clock (DC).
- The following models are examples of some of the OMRON EtherCAT slaves.

Slave type/name	Number of ports	Model
EtherCAT Junction Slave	3 ports	GX-JC03
	6 ports	GX-JC06

Precautions for Safe Use

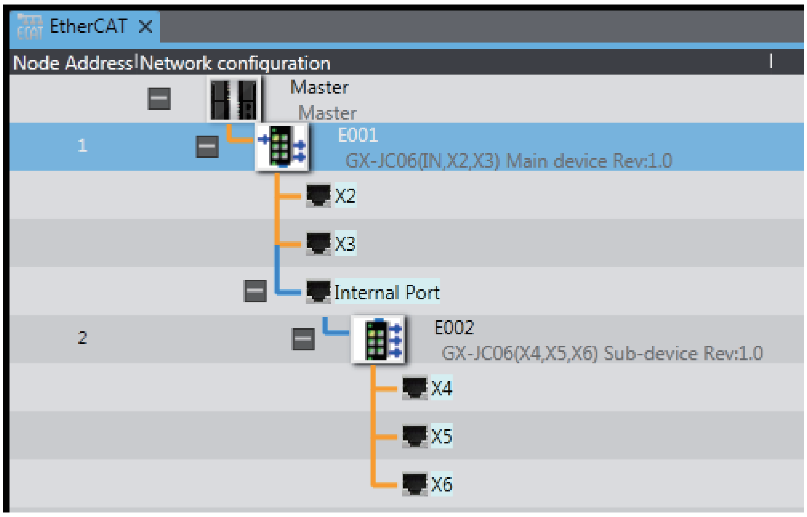
You cannot use standard Ethernet hubs or repeater hubs with EtherCAT communications. If you use one of these, a major fault level error or other error may occur.

Precautions for Correct Use

- The GX-JC03 and GX-JC06 EtherCAT Junction Slaves do not have hardware switches for node address setting. The Sysmac Studio is required to set the node addresses for these slaves. The node addresses are set to 0 by default. If you use these slaves with the default node address, a *Network Configuration Verification Error* (84220000 hex) event will occur.
- The GX-JC03 EtherCAT Junction Slave needs only one node address. The GX-JC06 EtherCAT Junction Slave needs two node addresses.

Additional Information

The GX-JC06 EtherCAT Junction Slave which appears to be one Unit is actually composed of two slaves. The two slaves appear as "Main device" and "Sub-device" in the Sysmac Studio, respectively, as shown in the figure below. The input port of Sub-device appears as "Internal Port" in the Sysmac Studio and it is internally connected to the Main device.



● Sysmac Studio

The Sysmac Studio runs on a personal computer and it is used to configure EtherCAT networks and slaves, and to program, monitor, and debug the Controller.

● Communications Cables

Use a straight, shielded twisted-pair cable (double shielding with aluminum tape and braiding) of Ethernet category 5 (100BASE-TX) or higher.

● ESI (EtherCAT Slave Information) File

The ESI files contain information unique to the EtherCAT slaves in XML format.

You can load an ESI file into the Sysmac Studio, to easily allocate slave process data and make other settings.

● Unit Power Supplies

This power supply is for slave communications and internal operation.

● I/O Power Supply

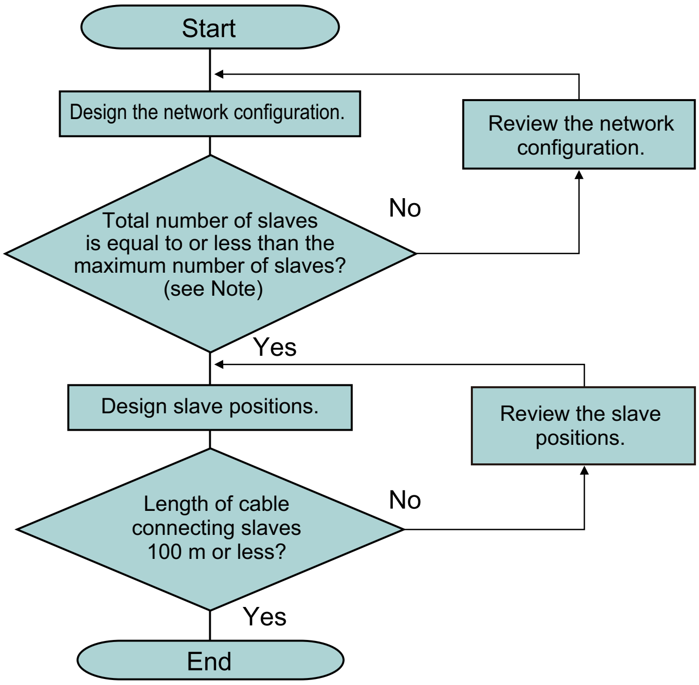
This power supply is for I/O operations of external devices connected to the slaves.

Additional Information

With a CPU Unit with unit version 1.06 or later and Sysmac Studio version 1.07 or higher, you can add NX-series Safety Control Units to the EtherCAT network.
You can use NX-series Safety Control Units on EtherCAT Slave Terminals to build a safety control system on EtherCAT.

1-2-2 Determining the Network Configuration

Determine the type, total number, and positions of slaves in the network. Check the total number of slaves and the cable length between slaves based on the following workflow.


Start
Design the network configuration.
Review the network configuration.
Total number of slaves is equal to or less than the maximum number of slaves? (see Note)
No
Yes
Design slave positions.
Review the slave positions.
Length of cable connecting slaves 100 m or less?
No
Yes
End

Note Refer to *1-3 Specifications of Built-in EtherCAT Port* on page 1-10 for details on the maximum number of slaves.

Precautions for Safe Use

- You cannot use standard Ethernet hubs or repeater hubs with EtherCAT communications. If you use one of these, a major fault level error or other error may occur.
- Make sure that the communications distance, number of devices connected, and method of connection for EtherCAT are within specifications.

1-3 Specifications of Built-in EtherCAT Port

Item	Specification					
	NX701-□□□□	NX502-□□□□	NX102-□□□□	NX1P2-□□□□□□□	NJ501-□□□□/NJ301-□□□□	NJ101-□□□□
Communications protocol	EtherCAT protocol					
Supported services	CoE (process data communications and SDO communications)[*1]					
Synchronization	DC (Distributed Clock)					
Physical layer	100BASE-TX					
Modulation	Baseband					
Baud rate	100 Mbit/s (100BASE-TX)					
Duplex mode[*2]	Auto					
Topology	Line, daisy chain, and branching	Line, daisy chain, branching, and ring[*3]				
Transmission media	Twisted-pair cable of category 5 or higher (Recommended cable: straight, double-shielded cable with aluminum tape and braiding)					
Maximum transmission distance between nodes	100 m					
Maximum number of slaves	512	256	64	16[*4]	192	64
Settable node address range	1 to 512	1 to 256	1 to 192			
Maximum process data size	Input: 11,472 bytes Output: 11,472 bytes *5		Input: 5,736 bytes Output: 5,736 bytes *6	Input: 1,434 bytes Output: 1,434 bytes *7	Input: 5,736 bytes Output: 5,736 bytes *6	
Maximum size per slave	Input: 1,434 bytes Output: 1,434 bytes					
Maximum message size	2,048 bytes					

Item	Specification					
	NX701-□□□□	NX502-□□□□	NX102-□□□□	NX1P2-□□□□□□□	NJ501-□□□□/NJ301-□□□□	NJ101-□□□□
Communications cycle	• Primary periodic task 125 μs, 250 μs to 8 ms (in 250-μs increments) • Priority-5 periodic task 125 μs, 250 μs to 100 ms (in 250-μs increments)[*8]	250 μs to 8 ms (in 250-μs increments)	1,000 μs to 32,000 μs (in 250-μs increments)	2,000 μs to 8,000 μs (in 250-μs increments)[*9]	500 μs[*10], 1,000 μs, 2,000 μs, or 4,000 μs	1,000 μs, 2,000 μs, or 4,000 μs
Sync jitter	1 μs max.					

*1. Refer to *3-1-1 CoE (CAN Application Protocol over EtherCAT)* on page 3-2 for details on CoE.

*2. Connection is possible only in full duplex mode. Half-duplex connections will result in link OFF status.

*3. A ring topology can be used for project unit version 1.40 or later.

*4. The maximum number of slaves is 8 for an NX1P2-9B□□□□□ CPU Unit.

*5. However, the data must be within eight frames.

*6. However, for project unit version earlier than 1.40, the data must be within four frames.

*7. However, for project unit version earlier than 1.40, the data must be within one frame.

*8. Setting is available only for integer multiples of the task period of the primary periodic task.

*9. The communications cycle is 4,000 μs to 8,000 μs (in 250-μs increments) for an NX1P2-9B□□□□□ CPU Unit.

*10. Unit version 1.03 or later of the CPU Unit is required to use this setting on the NJ301-□□□□.

1-4 Overview of Communications

This section provides an overview of the communications functions of the built-in EtherCAT port.

1-4-1 Process Data Communications and SDO Communications

The built-in EtherCAT port performs the following communications method to exchange information with EtherCAT slaves.

- Process data communications
- SDO communications

Process Data Communications

"Process data communications" is a cyclic communications method in which control information is exchanged in a fixed cycle between the EtherCAT master and slaves.
The fixed cycle is called a process data communications cycle. The EtherCAT master can exchange information with EtherCAT slaves in realtime in this process data communications cycle.
The same control period is also used for the process data communications cycle for EtherCAT. This enables precise sequence and motion control in a fixed period with very little deviation.
Refer to *6-1 Process Data Communications (PDO Communications)* on page 6-2 for details on the specifications of process data communications.

SDO communications

"SDO communications" is a communications method in which control information is exchanged in non-cyclic event communications between the EtherCAT master and slaves.
You can use EtherCAT communications instructions to read and write the SDO data in EtherCAT slaves.
Refer to *6-2 SDO Communications* on page 6-19 for details on the specifications of SDO communications.

1-4-2 Other Functions

In addition to process data communications and SDO communications, the built-in EtherCAT port also provides functions related to EtherCAT network configurations and setup, as well as communications control and maintenance during operation or at error occurrence.
Among these functions, some cannot be used depending on the unit version of the CPU Unit, project unit version, the Sysmac Studio version, or their combination. Refer to *A-7 Version Information* on page A-36 for versions that can be used.

Network Configurations and Setup

● Enable/Disable Setting for Slaves

Use this function to select the EtherCAT slaves for communications with from among those registered in the network configuration information.

- You can design a network with future addition of EtherCAT slaves in mind, by setting EtherCAT slaves that you plan to add at a later time to **Disabled** and then registering them in the network configuration information on the EtherCAT master.
- You can change the EtherCAT slaves for communications based on the device configuration during system operation.

You can enable/disable each slave in the Sysmac Studio. Refer to *5-5 EtherCAT Slave Parameter Settings* on page 5-20 for the setting procedure.
To enable/disable slaves during system operation, use the Enable/Disable EtherCAT Slave instruction. Refer to the *NJ/NX-series Instructions Reference Manual (Cat. No. W502)* for details on this instruction.

● Cable Redundancy Function

Use this function to continue the communications with EtherCAT slaves, even if a communications cable is broken in the EtherCAT communications path.
With the NJ/NX-series CPU Units, use Junction Slaves and configure a ring topology in the network configuration. In the ring topology, the EtherCAT master can continue the normal communications with the EtherCAT slaves even if there is only one point of disconnection in the communications cables.
You can set the cable redundancy function from the Sysmac Studio. Refer to *5-2-2 Cable Redundancy Setting* on page 5-5 for the cable redundancy function.
Refer to *4-1-3 Ring Topology* on page 4-4 for the ring topology.

Communications Control during Operation

● Wait Time Setting for Slave Startup

Use this function to set the wait time until all of the EtherCAT slaves are connected to the network.
When you use a slave that takes time to start, use a longer wait time setting to prevent errors.
Set the wait time for slave startup in the Sysmac Studio. Refer to *5-4 EtherCAT Master Parameter Settings* on page 5-14 for the setting procedure.

Communications Control at Error Occurrence

● Fail-soft Operation

Use this function to continue or stop the communications with EtherCAT slaves that are operating normally, if a communications error occurs.
"Fail-soft operation" refers an operation that only normally operating EtherCAT slaves are allowed to operate continuously.
The EtherCAT master can continue the communications with the EtherCAT slaves until the operation is stopped safely through the user program or user operation.

You enable/disable the fail-soft operation in Sysmac Studio. Refer to *5-4 EtherCAT Master Parameter Settings* on page 5-14 for the setting procedure.

Maintenance

● Disconnecting/Reconnecting Slaves

Use this function to temporarily stop and start process data communications with a specified slave. It is useful because, during system operation, you can replace an erroneous EtherCAT slave without interrupting the communications with EtherCAT slaves that are operating normally.
Refer to *9-4 Replacing Slaves during Communications* on page 9-26 for details on how to use this function.

● Diagnosis/Statistics Log

The diagnostic and statistical information provides statistics on the number of communications frames sent and received by the EtherCAT master and EtherCAT slaves as well as the number of frames for which errors were detected.
This function acquires the diagnostic and statistical information at the specified cycle and saves the information as a log file in an SD Memory Card that is mounted on the CPU Unit.
You can use it to diagnose the EtherCAT network line quality based on the diagnostic and statistical information.
Use this function for the following applications.

- Checking the EtherCAT network line quality for predictive monitoring and preventive maintenance
- Finding locations of errors when they occur

Refer to *9-2-4 Diagnostic and Statistical Information* on page 9-3 for details on how to use this function.

1-5 EtherCAT Communications Procedure

1-5-1 Overview

Step		Section
1.	Mounting and Setting Devices and Hardware	*2-2 Setting the Node Addresses of the EtherCAT Slaves* on page 2-9
2.	Laying EtherCAT Communications Cables	*Section 4 EtherCAT Network Wiring* on page 4-1
3.	Creating the EtherCAT Network Configuration	*5-2 Creating the EtherCAT Network Configuration* on page 5-3
4.	Setting EtherCAT Slave Variables and Axes	*5-3 Setting EtherCAT Slave Variables and Axes* on page 5-9
5.	Setting EtherCAT Parameters	*5-4 EtherCAT Master Parameter Settings* on page 5-14 *5-5 EtherCAT Slave Parameter Settings* on page 5-20
6.	Programming	*Section 6 Process Data Communications and SDO Communications* on page 6-1
7.	Turning ON Power and Going Online from the Sysmac Studio	*2-1-3 Connecting the Sysmac Studio* on page 2-7 *3-3 State Transitions for EtherCAT Communications* on page 3-9
8.	Online Debugging	*5-6 Comparing and Merging EtherCAT Network Configurations* on page 5-28
9.	Downloading the Network Configuration Information and the User Program	*5-9 Downloading the Network Configuration Information from the Sysmac Studio* on page 5-45
10.	Checking the EtherCAT Process Data Communications Status	*5-10 Confirming Communications after Completing EtherCAT Configuration and Settings* on page 5-47

1-5-2 Details

Step	Description	Sysmac Studio operation	Section

Step		Description	Sysmac Studio operation	Section
1.	Mounting and Setting Devices and Hardware	• Use the hardware switches on all of the EtherCAT slaves in the network to set the node addresses. (The starting node address and sequence are not specified.) Refer to relevant manuals for each slave for information on the procedure for setting node addresses. **Note** Use the Sysmac Studio to set the node address if there are no hardware switches or the node address is beyond the range that can be set with the hardware switches. • Connect the EtherCAT slaves and external I/O devices.		*2-2 Setting the Node Addresses of the EtherCAT Slaves* on page 2-9
2.	Laying EtherCAT Communications Cables	Connect the EtherCAT slaves to the EtherCAT port of the NJ/NX-series CPU Unit. If there is more than one EtherCAT slave, connect them using a daisy chain/line topology, branching topology, or ring topology.		*Section 4 EtherCAT Network Wiring* on page 4-1
3.	Creating the EtherCAT Network Configuration	• Use the Sysmac Studio to create a new project. a) Go offline and create the EtherCAT network configuration. or b) Go online and automatically create the EtherCAT network configuration from the actual network devices. Do so after making the online connection that is described in step 7.	Create the EtherCAT network configuration with **EtherCAT** under **Configurations and Setup**.	*5-2 Creating the EtherCAT Network Configuration* on page 5-3
4.	Setting EtherCAT Slave Variables and Axes			
	All EtherCAT Slaves	Allocate variables to the I/O ports. • If necessary, change the names of automatically generated device variables for each I/O port to user-defined variable names.	Make the settings with **I/O Map** under **Configurations and Setup**.	*5-3-1 Registering Device Variables for All EtherCAT Slaves* on page 5-9
	Only EtherCAT Servo Drive and Encoder Input Slaves	Set up the axes. • Create axes (axes variables). • Set the axis types for the Axis Variables (to a servo axis, virtual axis, etc.), and set the IDs of the Servo Drives. Axis Variables are automatically registered in the global variable table.	Create and set up the axes from **Axis Settings** under **Configuration and Setup** − **Motion Control Setup** − **Axis Settings** − **Add**.	*5-3-2 Axis Settings for Servo Drives and Encoder Input Slaves* on page 5-13
5.	Setting EtherCAT Parameters			

Step		Description	Sysmac Studio operation	Section
	Setting EtherCAT Master Parameters	Set the EtherCAT master parameters. (Examples: Fail-soft Operation Setting and Wait Time for Slave Startup) The values that are set are reflected in the network configuration information.	Set an EtherCAT master with **EtherCAT** under **Configurations and Setup**.	*5-4 EtherCAT Master Parameter Settings* on page 5-14
	Setting EtherCAT Slave Parameters	Set the EtherCAT slave parameters. (Example: Enable/disable slaves)	Set EtherCAT slaves with **EtherCAT** under **Configurations and Setup**.	*5-5 EtherCAT Slave Parameter Settings* on page 5-20
6.	Programming			
	Process Data Communications	Specify the device variables in the user program. Input conditions include system-defined variables for network error flags and normal or error flags for each slave.	Create the programs with **POUs** under **Programming**.	*6-1 Process Data Communications (PDO Communications)* on page 6-2
		Assign to the task that refreshes I/O for each EtherCAT slave.	Set it for each slave from **I/O Control Task Settings** under **Configurations and Setup** - **Task Settings**.	
	SDO communications	Read and write the following specified data for slaves: Use EC_CoESDORead and EC_CoESDOWrite instructions. • SDO data in slaves (parameters, error information, etc.)	Create the programs with **POUs** under **Programming**.	*6-2 SDO Communications* on page 6-19
7.	Turning ON Power and Going Online from the Sysmac Studio	• Turn ON the power supply to EtherCAT slaves. • Turn ON the I/O power supplies to the slaves. • Turn ON the power supply to NJ/NX-series Controller. • Use the Sysmac Studio to set communications with the NJ/NX-series Controller and go online.	Select **Communications Setup** from the **Controller Menu** and make the settings to go online.	*2-1-3 Connecting the Sysmac Studio* on page 2-7 *3-3 State Transitions for EtherCAT Communications* on page 3-9
8.	Online Debugging	Compare and merge the network configuration that was set on the Sysmac Studio and the actual network configuration.	Right-click the EtherCAT master under **Configurations and Setup** − **EtherCAT** and select **Compare and Merge with Actual Network Configuration**.	*5-6 Comparing and Merging EtherCAT Network Configurations* on page 5-28

Step		Description	Sysmac Studio operation	Section
9.	Downloading the Network Configuration Information and the User Program	Download the network configuration information (EtherCAT configuration, process data information, and parameters). **Note** Use the synchronization operation of the Sysmac Studio to download the project.	Select **Synchronization** from the **Controller Menu** to download and verify the network configuration information.	*5-9 Downloading the Network Configuration Information from the Sysmac Studio* on page 5-45
10.	Checking the EtherCAT Process Data Communications Status	• In the Sysmac Studio, check that process data communications with all EtherCAT slaves are performed normally. • Check the indicators on the NJ/NX-series CPU Unit. a) A flashing yellow EtherCAT LINK/ACT indicator shows that data is being transmitted and received after the link is established. b) A solid green EtherCAT NET RUN indicator shows the device is in the Operational state (normal communications state). c) If the EtherCAT NET ERR indicator is not lit, there is no error.	Display the Controller Status Pane and switch to the detailed view.	*5-10 Confirming Communications after Completing EtherCAT Configuration and Settings* on page 5-47

Part Names and Slave Settings

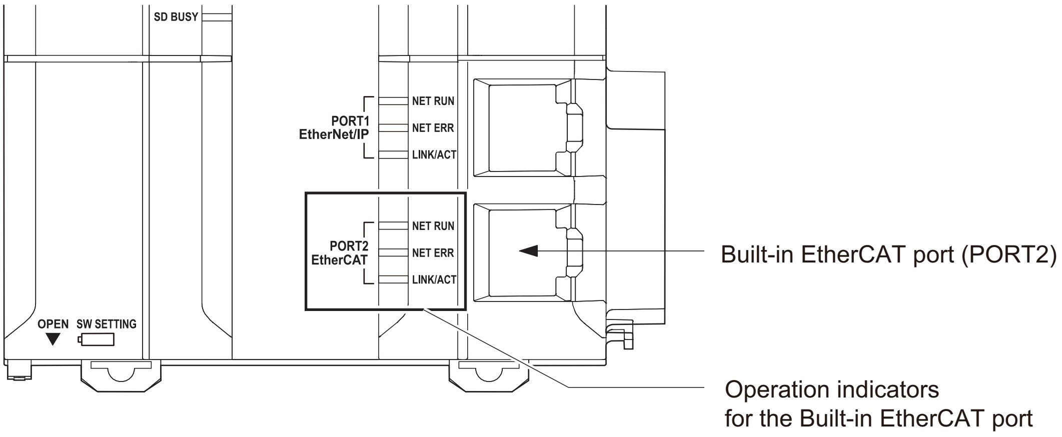
This section provides the part names and describes the slave settings and Sysmac device functions.

2-1 Part Names and Functions

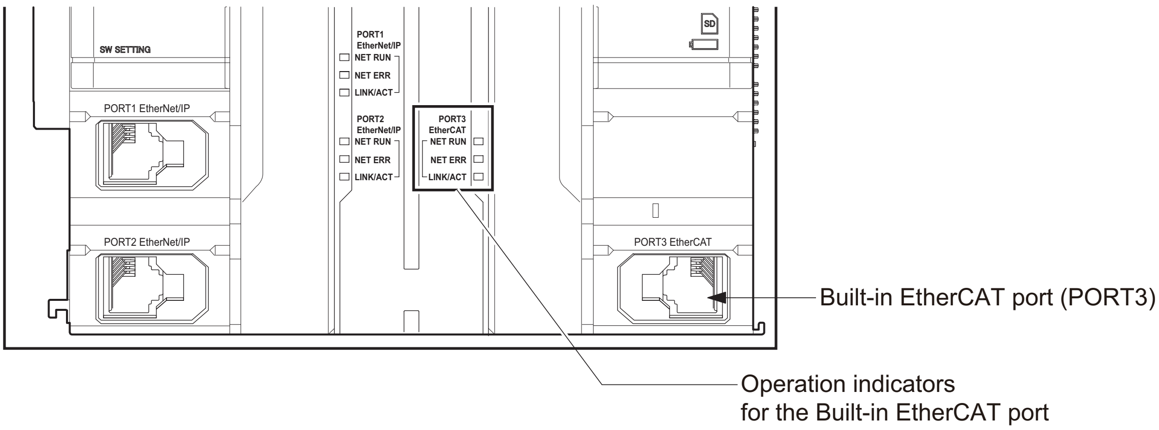
2-1-1 Built-in EtherCAT Port Indicators

The indicators show the operating status of the built-in EtherCAT port on an NJ/NX-series CPU Unit. The locations of indicators of the built-in EtherCAT port on the CPU Unit as well as the colors and statuses of indicators associated with corresponding operating status are shown below.
The locations of the built-in EtherCAT port indicators are shown below.

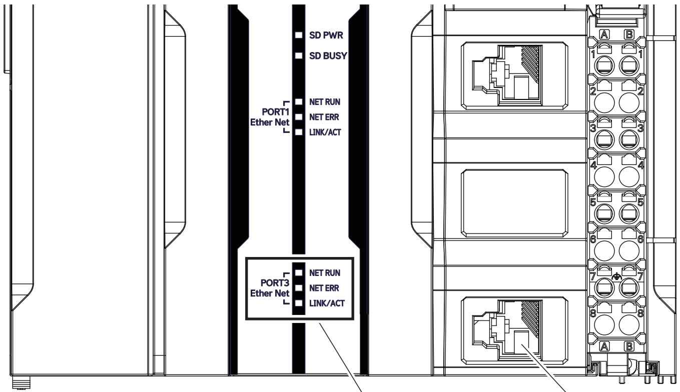
● NJ-series CPU Units



● NX701 CPU Units



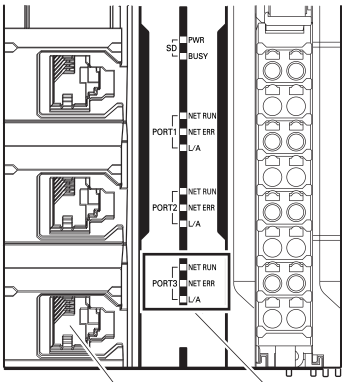
● NX502 CPU Units



Built-in EtherCAT port (PORT3)

Operation indicators for the Built-in EtherCAT port

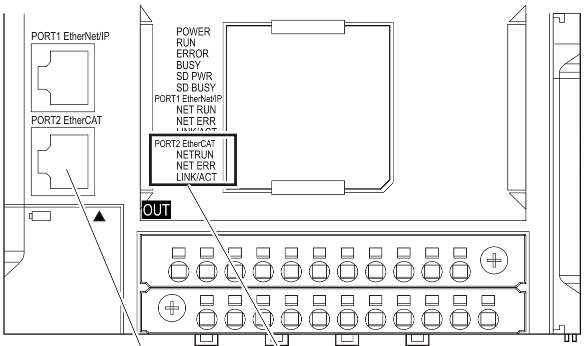
● NX102 CPU Unit



Operation indicators for the Built-in EtherCAT port

Built-in EtherCAT port (PORT3)

● NX1P2 CPU Unit



Operation indicators for the Built-in EtherCAT port

Built-in EtherCAT port (PORT2)

Operation Indicators and Operating Status for the Built-in EtherCAT Port

The operating status corresponding to colors and status of the built-in EtherCAT port indicators are shown below.

Label	Color	Status	Meaning
NET RUN	Gree n	Lit	EtherCAT communications are in progress. • Inputs and outputs for I/O data are in operation.
		Flashing	EtherCAT communications are being established. This indicator shows either of the following conditions. • Only message communications are in operation. • Only message communications and I/O data inputs are in operation.
		Not lit	EtherCAT communications are stopped. • The power supply is OFF or the CPU Unit was reset. • A *MAC Address Error*, *Communications Controller Error*, or other error occurred.
NET ERR	Red	Lit	A hardware error or unrecoverable error occurred, such as for exception processing.
		Flashing	A recoverable error occurred.
		Not lit	There are no errors.
LINK/ACT or L/A	Yel-low	Lit	A link was established.
		Flashing	Data communications are in progress after the link is established. Flashes every time data is sent or received.
		Not lit	The link was not established.

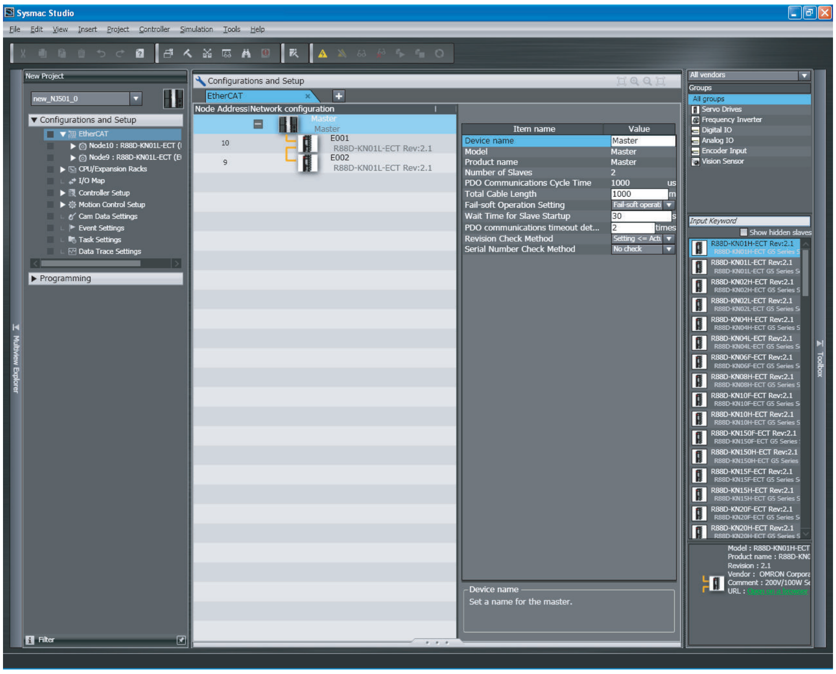
Refer to *3-3-2 Control States for EtherCAT Communications* on page 3-9 for details on the states.

2-1-2 Windows Used in Sysmac Studio

Use the Sysmac Studio to create the EtherCAT network configuration and to make other settings.
The following windows are used.

● Creating the EtherCAT Network Configuration

Use the EtherCAT Configuration Editor offline to register slaves in the EtherCAT slave configuration.
Or, upload the network configuration online.

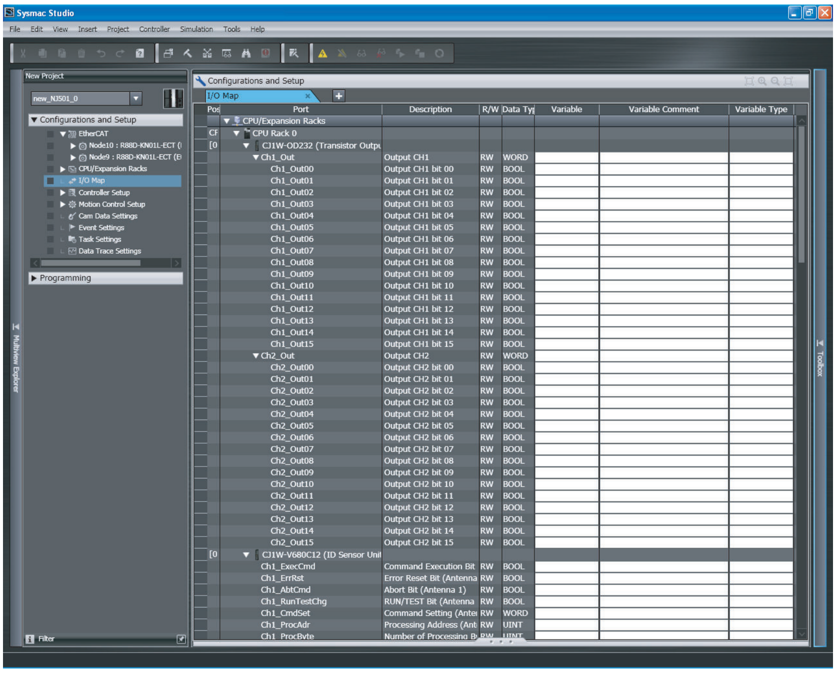


Item name	Value
Device name	Master
Model	Master
Product name	Master
Number of Slaves	2
PDO Communications Cycle Time	1000 us
Total Cable Length	1000 m
Fail-soft Operation Setting	Fail-soft operati
Wait Time for Slave Startup	30 s
PDO communications timeout det...	2 times
Revision Check Method	Setting <= Actı
Serial Number Check Method	No check

Device name
Set a name for the master.

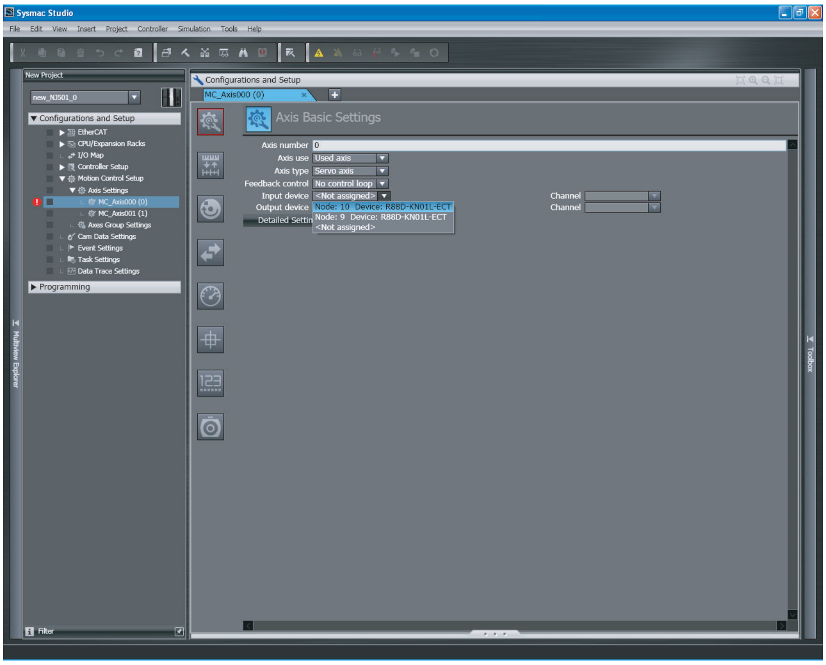
● Setting EtherCAT Slave Variables and Axes

- I/O Map: Used to allocate device variables.
  Use the I/O Map to assign device variables to the I/O ports of the EtherCAT slaves.



Pos	Port	Description	R/W	Data Typ	Variable	Variable Comment	Variable Type
	CPU/Expansion Racks						
CF	CPU Rack 0						
[0	CJ1W-OD232 (Transistor Outpu						
	Ch1_Out	Output CH1	RW	WORD			
	Ch1_Out00	Output CH1 bit 00	RW	BOOL			
	Ch1_Out01	Output CH1 bit 01	RW	BOOL			
	Ch1_Out02	Output CH1 bit 02	RW	BOOL			
	Ch1_Out03	Output CH1 bit 03	RW	BOOL			
	Ch1_Out04	Output CH1 bit 04	RW	BOOL			
	Ch1_Out05	Output CH1 bit 05	RW	BOOL			
	Ch1_Out06	Output CH1 bit 06	RW	BOOL			
	Ch1_Out07	Output CH1 bit 07	RW	BOOL			
	Ch1_Out08	Output CH1 bit 08	RW	BOOL			
	Ch1_Out09	Output CH1 bit 09	RW	BOOL			
	Ch1_Out10	Output CH1 bit 10	RW	BOOL			
	Ch1_Out11	Output CH1 bit 11	RW	BOOL			
	Ch1_Out12	Output CH1 bit 12	RW	BOOL			
	Ch1_Out13	Output CH1 bit 13	RW	BOOL			
	Ch1_Out14	Output CH1 bit 14	RW	BOOL			
	Ch1_Out15	Output CH1 bit 15	RW	BOOL			
	Ch2_Out	Output CH2	RW	WORD			
	Ch2_Out00	Output CH2 bit 00	RW	BOOL			
	Ch2_Out01	Output CH2 bit 01	RW	BOOL			
	Ch2_Out02	Output CH2 bit 02	RW	BOOL			
	Ch2_Out03	Output CH2 bit 03	RW	BOOL			
	Ch2_Out04	Output CH2 bit 04	RW	BOOL			
	Ch2_Out05	Output CH2 bit 05	RW	BOOL			
	Ch2_Out06	Output CH2 bit 06	RW	BOOL			
	Ch2_Out07	Output CH2 bit 07	RW	BOOL			
	Ch2_Out08	Output CH2 bit 08	RW	BOOL			
	Ch2_Out09	Output CH2 bit 09	RW	BOOL			
	Ch2_Out10	Output CH2 bit 10	RW	BOOL			
	Ch2_Out11	Output CH2 bit 11	RW	BOOL			
	Ch2_Out12	Output CH2 bit 12	RW	BOOL			
	Ch2_Out13	Output CH2 bit 13	RW	BOOL			
	Ch2_Out14	Output CH2 bit 14	RW	BOOL			
	Ch2_Out15	Output CH2 bit 15	RW	BOOL			
[0	CJ1W-V680C12 (ID Sensor Unit						
	Ch1_ExecCmd	Command Execution Bit	RW	BOOL			
	Ch1_ErrRst	Error Reset Bit (Antenna	RW	BOOL			
	Ch1_AbtCmd	Abort Bit (Antenna 1)	RW	BOOL			
	Ch1_RunTestChg	RUN/TEST Bit (Antenna	RW	BOOL			
	Ch1_CmdSet	Command Setting (Anter	RW	WORD			
	Ch1_ProcAdr	Processing Address (Ant	RW	UINT			
	Ch1_ProcByte	Number of Processing By	RW	UINT			

- Axis Basic Settings: Used to create Axis Variables and set parameters for Servo Drive and encoder input slaves.
  Use the Axis Basic Settings to assign Axis Variables to the Servo Drive/encoder input slaves.

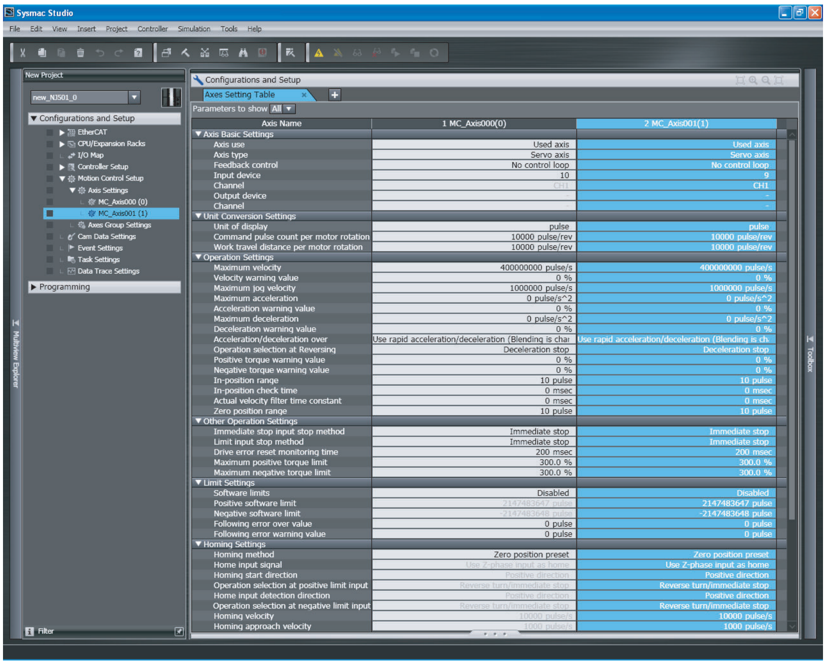


Configurations and Setup

Axis number: 0
Axis use: Used axis
Axis type: Servo axis
Feedback control: No control loop
Input device: <Not assigned>
Output device:
- Node: 10 Device: R88D-KN01L-ECT
- Node: 9 Device: R88D-KN01L-ECT
- <Not assigned>

Channel:
Channel:

- Setting Axis Parameters:
  Use the Axis Setting Table to set axis parameters.



Configurations and Setup

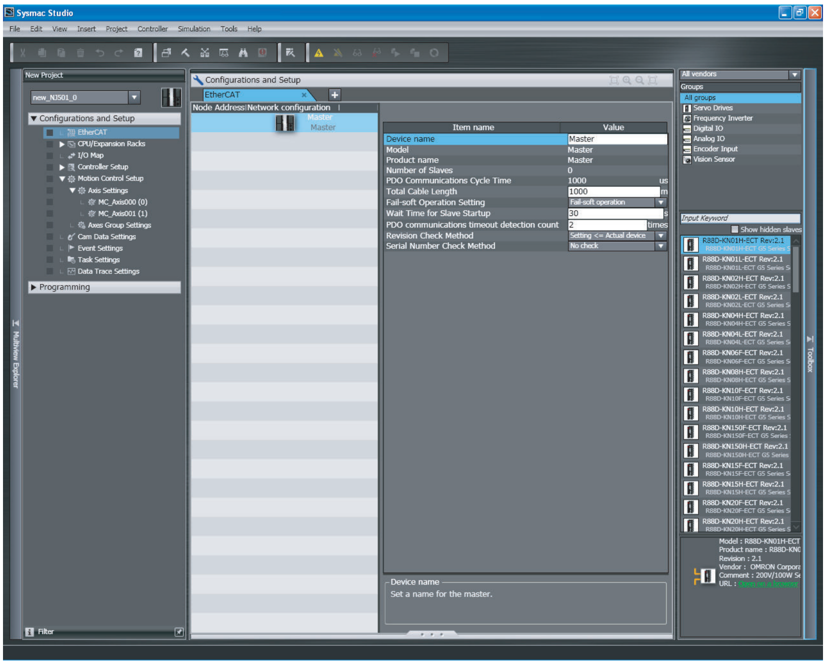
Parameters to show: All

Axis Name	1 MC_Axis000(0)	2 MC_Axis001(1)
▼ Axis Basic Settings		
Axis use	Used axis	Used axis
Axis type	Servo axis	Servo axis
Feedback control	No control loop	No control loop
Input device	10	9
Channel	CH1	CH1
Output device	-	-
Channel	-	-
▼ Unit Conversion Settings		
Unit of display	pulse	pulse
Command pulse count per motor rotation	10000 pulse/rev	10000 pulse/rev
Work travel distance per motor rotation	10000 pulse/rev	10000 pulse/rev
▼ Operation Settings		
Maximum velocity	400000000 pulse/s	400000000 pulse/s
Velocity warning value	0 %	0 %
Maximum jog velocity	1000000 pulse/s	1000000 pulse/s
Maximum acceleration	0 pulse/s^2	0 pulse/s^2
Acceleration warning value	0 %	0 %
Maximum deceleration	0 pulse/s^2	0 pulse/s^2
Deceleration warning value	0 %	0 %
Acceleration/deceleration over	Use rapid acceleration/deceleration (Blending is char	Use rapid acceleration/deceleration (Blending is ch
Operation selection at Reversing	Deceleration stop	Deceleration stop
Positive torque warning value	0 %	0 %
Negative torque warning value	0 %	0 %
In-position range	10 pulse	10 pulse
In-position check time	0 msec	0 msec
Actual velocity filter time constant	0 msec	0 msec
Zero position range	10 pulse	10 pulse
▼ Other Operation Settings		
Immediate stop input stop method	Immediate stop	Immediate stop
Limit input stop method	Immediate stop	Immediate stop
Drive error reset monitoring time	200 msec	200 msec
Maximum positive torque limit	300.0 %	300.0 %
Maximum negative torque limit	300.0 %	300.0 %
▼ Limit Settings		
Software limits	Disabled	Disabled
Positive software limit	2147483647 pulse	2147483647 pulse
Negative software limit	-2147483648 pulse	-2147483648 pulse
Following error over value	0 pulse	0 pulse
Following error warning value	0 pulse	0 pulse
▼ Homing Settings		
Homing method	Zero position preset	Zero position preset
Home input signal	Use Z-phase input as home	Use Z-phase input as home
Homing start direction	Positive direction	Positive direction
Operation selection at positive limit input	Reverse turn/immediate stop	Reverse turn/immediate stop
Home input detection direction	Positive direction	Positive direction
Operation selection at negative limit input	Reverse turn/immediate stop	Reverse turn/immediate stop
Homing velocity	10000 pulse/s	10000 pulse/s
Homing approach velocity	1000 pulse/s	1000 pulse/s

● EtherCAT Master and Slave Parameter Settings

Set the EtherCAT master and slave parameters after selecting a master or slave on the EtherCAT network configuration.

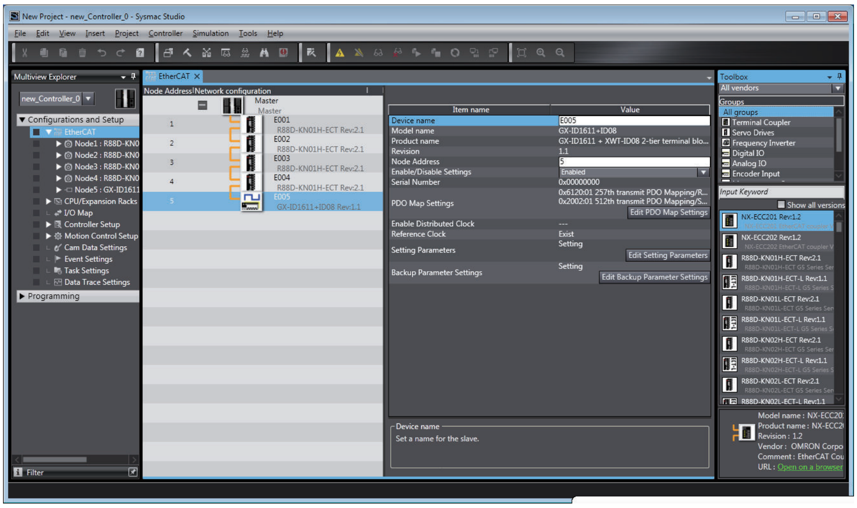
- Parameter Setting for the EtherCAT Master



Item name	Value
Device name	Master
Model	Master
Product name	Master
Number of Slaves	0
PDO Communications Cycle Time	1000 us
Total Cable Length	1000 m
Fail-soft Operation Setting	Fail-soft operation
Wait Time for Slave Startup	30 s
PDO communications timeout detection count	2 times
Revision Check Method	Setting <= Actual device
Serial Number Check Method	No check

Device name
Set a name for the master.

- Parameter Setting for the EtherCAT Slaves



Item name	Value
Device name	E005
Model name	GX-ID1611+ID08
Product name	GX-ID1611 + XWT-ID08 2-tier terminal blo...
Revision	1.1
Node Address	5
Enable/Disable Settings	Enabled
Serial Number	0x00000000
PDO Map Settings	0x6120:01 257th transmit PDO Mapping/R... 0x2002:01 512th transmit PDO Mapping/S... Edit PDO Map Settings
Enable Distributed Clock	---
Reference Clock	Exist
Setting Parameters	Setting Edit Setting Parameters
Backup Parameter Settings	Setting Edit Backup Parameter Settings

Device name
Set a name for the slave.

Refer to *Section 5 Setting Up EtherCAT Communications with the Sysmac Studio* on page 5-1 for the Sysmac Studio procedures.

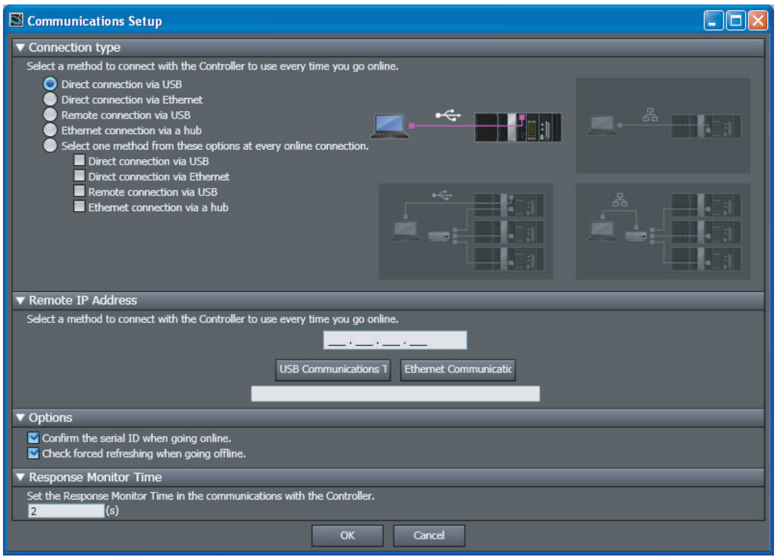
2-1-3 Connecting the Sysmac Studio

You can connect the Sysmac Studio to the NJ/NX-series CPU Unit through the USB or EtherNet/IP port. You must set the connection method, IP address to connect to, and other parameters for communications between the computer and Controller.

Precautions for Correct Use

You cannot connect a computer to an NX701 CPU Unit with hardware revision A or later, NX502 CPU Unit, NX102 CPU Unit, or NX1P2 CPU Unit via USB, because it does not provide a peripheral USB port.

1 Select **Communications Setup** from the **Controller Menu**.
The **Communications Setup** Dialog Box is displayed.



Communications Setup

▼ Connection type

Select a method to connect with the Controller to use every time you go online.

- ● Direct connection via USB
- ○ Direct connection via Ethernet
- ○ Remote connection via USB
- ○ Ethernet connection via a hub
- ○ Select one method from these options at every online connection.
  - □ Direct connection via USB
  - □ Direct connection via Ethernet
  - □ Remote connection via USB
  - □ Ethernet connection via a hub

▼ Remote IP Address

Select a method to connect with the Controller to use every time you go online.

___.___.___.___

▼ Options

☑ Confirm the serial ID when going online.

☑ Check forced refreshing when going offline.

▼ Response Monitor Time

Set the Response Monitor Time in the communications with the Controller.

2 (s)

2 Select the connection method for the connection configuration in the **Connection type** Area. If you select **Remote connection via USB** or **Ethernet connection via a hub**, enter the IP address of the Controller in the **Remote IP Address** Area. Also set **Options** and **Response Monitor Time** if necessary. Refer to *Setting Items on the **Communications Setup** Dialog Box* on page 2-8 described later in this manual for information on the settings.

3 Click the **OK** Button.
This completes the setup.

● Setting Items on the Communications Setup Dialog Box

Item	Description	
Connection type	Specify the connection method to use for online communications. The specified connection method is used when you go online.	
Remote IP Address	If you select **Remote connection via USB** or **Ethernet connection via a hub**, set the IP address of the Controller that you normally connect to.	
	USB Communications Test Button **Ethernet Communications Test** Button	These buttons perform a communication test with the Controller at the specified remote IP address. The result is displayed as follows: • Normal: "Test OK" • Error: "The Controller was not found." or "There is more than one Controller with the specified IP address."
Options	Confirm the serial ID when going online.	When going online, the names and serial IDs are compared between the project and the Controller to make sure that a connection is made to the intended Controller.
	Check forced refreshing when going offline.	Before going offline, a check is made to see if any forced refreshing values are still in effect.
Response Monitor Time	You can set the response monitoring time for communications with the Controller. An error is displayed if a response is not received before this time expires. **Note** The time can be set to between 1 and 3,600 s.	

2-2 Setting the Node Addresses of the EtherCAT Slaves

This section describes the procedure to set the node addresses of EtherCAT slaves.

There are methods to set a node address with hardware switches or the Sysmac Studio.

Additional Information

- The hardware switch setting is read only once when the power is turned ON. Even if the setting is changed after the power supply is turned ON, the new setting will not be used until the next time that power is turned ON.
- If same node address is set for more than one node, a *Slave Node Address Duplicated Error* (24200000 hex) event occurs.
  For the operation of slaves when the event occurs, refer to the description of each event in *Errors in the EtherCAT Master Function Module* in the *NJ/NX-series Troubleshooting Manual (Cat. No. W503)*.
- If you use an OMRON EtherCAT slave, after setting the node address, you need to cycle the power supply to the slave to apply the setting.

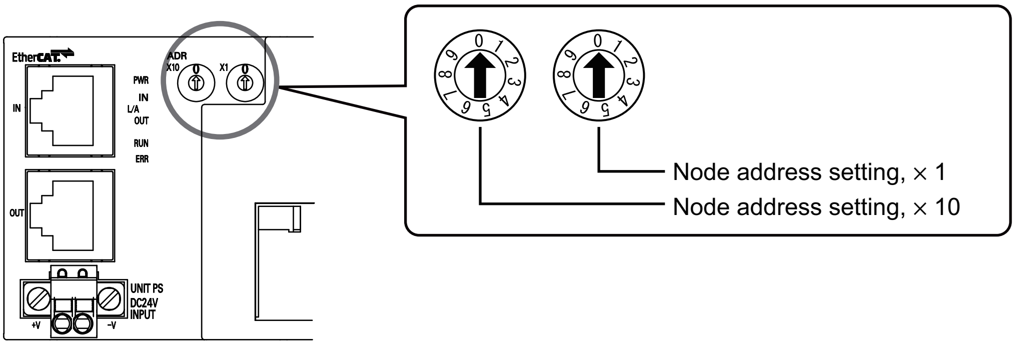
2-2-1 Setting Node Address Using Hardware Switches

For slaves with hardware switches, you can set the node address using only the hardware switches, instead of using the Sysmac Studio.

This section describes the node address setting with two hardware switches using an OMRON GX-series Remote I/O Terminal as an example below.

The 10s digit is set using the left rotary switch and the 1s digit is set using the right rotary switch.

The setting range is from 00 to 99.



The node address setting values are described in the following table.

Switch setting	Node address setting
00	Set with the Sysmac Studio.[*1]
01 to 99	Set with the hardware switches.

[*1] Refer to *1-3 Specifications of Built-in EtherCAT Port* on page 1-10 for details on the settable node address range.

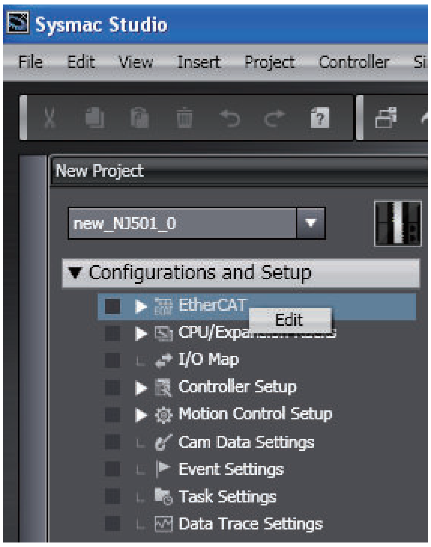
2-2-2 Setting the Node Address from the Sysmac Studio

Use the Sysmac Studio to set the node address if there are no hardware switches or the node address is beyond the range that can be set with the hardware switches.

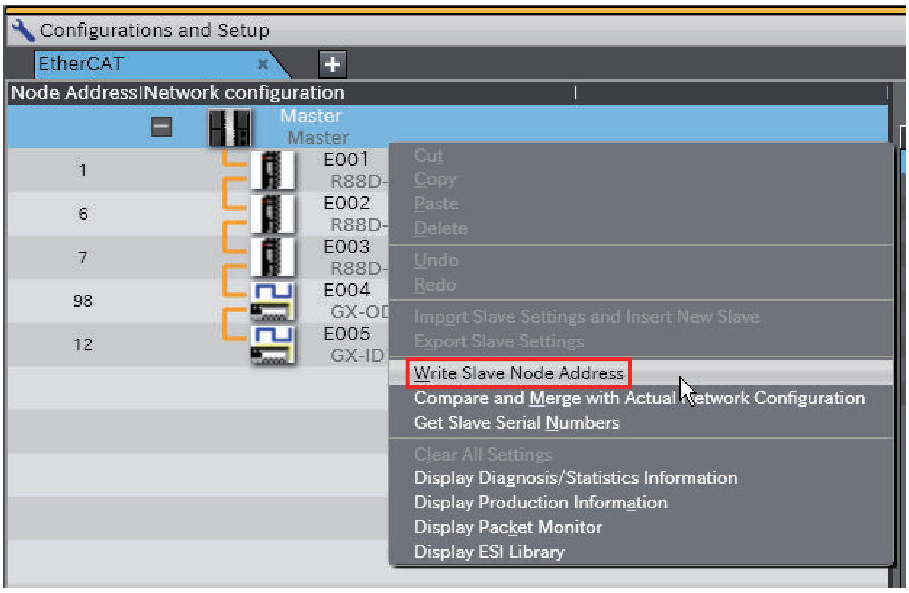
● Setting Procedure

1 Start the Sysmac Studio and go online with the Controller.

2 Double-click **EtherCAT** under **Configurations and Setup** on the Multiview Explorer. Or, right-click **EtherCAT** under **Configurations and Setup** and select **Edit**.



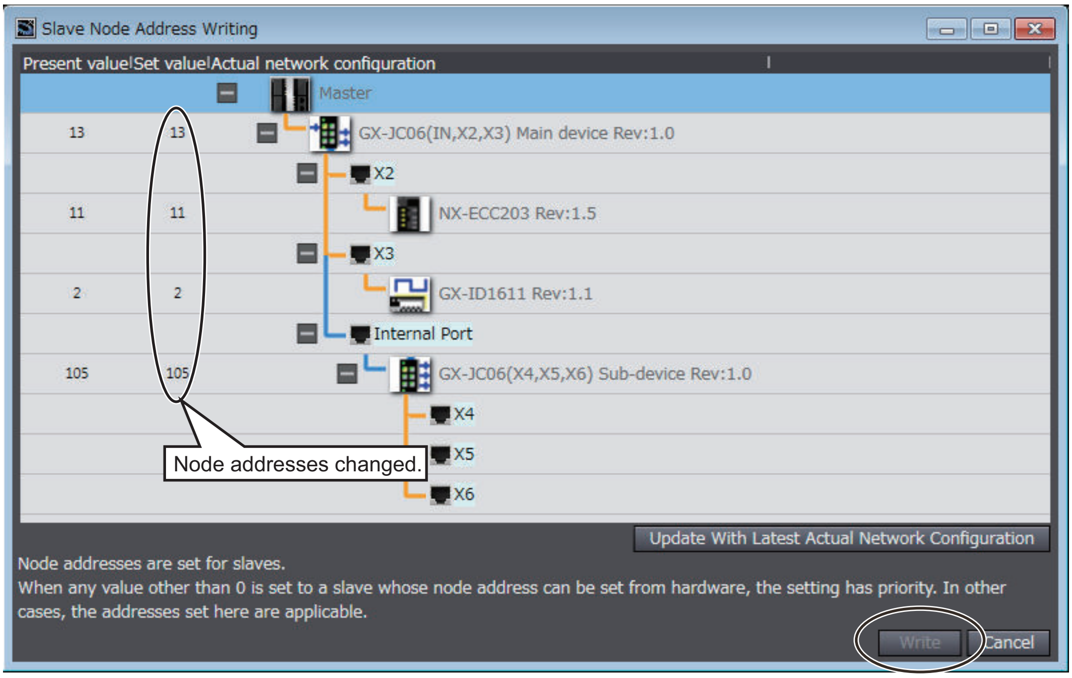
3 Right-click the EtherCAT master that is displayed in the EtherCAT Tab Page and select **Write Slave Node Address**.



The Slave Node Address Writing Dialog Box is displayed.

Refer to *When the Slave Node Address Writing Dialog Box Is Not Displayed* on page 2-11 if the dialog box is not displayed.

4 If there are slaves for which the node address is not set (i.e., for which the **present value** is 0) or if there is more than one slave with the same node address (indicated by ❗), change the **set values** of the slave addresses, and then click the **Write** Button.
(If the correct node addresses are set for all of the slaves, click the **Cancel** Button.)



Slave Node Address Writing

Present value	Set value	Actual network configuration
		Master
13	13	GX-JC06(IN,X2,X3) Main device Rev:1.0
		X2
11	11	NX-ECC203 Rev:1.5
		X3
2	2	GX-ID1611 Rev:1.1
		Internal Port
105	105	GX-JC06(X4,X5,X6) Sub-device Rev:1.0
		X4
		X5
		X6

Node addresses changed.

Node addresses are set for slaves.
When any value other than 0 is set to a slave whose node address can be set from hardware, the setting has priority. In other cases, the addresses set here are applicable.

The node addresses are written to the actual slaves.

5 Cycle the power supply to the slave for which **Set value** of the node address is changed.
The set node address becomes valid.

6 Click the **Update With Latest Actual Network Configuration** Button.
The present values of each slave node address are read and **Present value** of node addresses are updated. Check that **Present value** and **Set value** of the slave node address for which **Set value** was changed show the same values.

Version Information

Sysmac Studio version 1.22 or higher is required to use the **Update With Latest Actual Network Configuration** Button.

● When the Slave Node Address Writing Dialog Box Is Not Displayed

After you execute **Write Slave Node Address** and even several tens of seconds has passed, an error message may be displayed, instead of the **Slave Node Address Writing** Dialog Box.
The following are causes that the error message is displayed.

- The communications cable is not wired correctly.
- More than the maximum number of slaves are connected.

Make suitable corrections according to the displayed message and then write the slave node address again.

Refer to *5-6-3 When* ***Compare and Merge with Actual Network Configuration*** *Dialog Box Is Not Displayed* on page 5-32 for how to handle error messages.

2-3 Features of Sysmac Devices

2-3-1 Sysmac Devices

"Sysmac devices" is a generic name for EtherCAT slaves and other OMRON control components that were designed with the same communications and user interface specifications.
You can use the NJ/NX-series Machine Automation Controllers together with Sysmac devices and the Sysmac Studio Automation Software to achieve optimum functionality and ease of operation.

2-3-2 Sysmac Device Features

You can connect Sysmac device slaves to an NJ/NX-series Machine Automation Controller to use the following features.

Sysmac device features	Description
Troubleshooting	OMRON defines component error status as Sysmac errors for the entire Sysmac Series. You can use Sysmac errors to display errors that occur in slaves on the Sysmac Studio together with corrections for the errors.
Backing up and restoring parameters	An optional feature in the version-1.0.1 ESI specifications is used to enable backing up and restoring slave parameters (called backup parameters).
Saving node address settings	The node address setting for each slave is stored in non-volatile memory within the slave.
Verifying the EtherCAT network configuration using serial numbers	You can verify the EtherCAT network configuration based on serial numbers.

Troubleshooting

OMRON defines component error status as Sysmac errors for the entire Sysmac Series.
You can use Sysmac errors to display errors that occur in slaves on the Sysmac Studio together with corrections for the errors.
This allows you to detect slave errors and isolate the causes of the errors.
If a Sysmac error occurs in a slave, the values of the *_EC_SlavErr* system-defined variable and the element of the *_EC_SlavErrTbl* system-defined variable that corresponds to the slave node address change.
Refer to *7-1-3 EtherCAT Master Function Module, Category Name: _EC* on page 7-9 in the for details on the values.

Additional Information

- This feature is not supported for OMRON slaves that are not Sysmac devices and slaves that are manufactured by other companies. Check the error detection methods for each slave.
- Before you reset an error from the NJ/NX-series EtherCAT master, make sure you confirm the cause of the error in the slave.
- Sysmac errors are not displayed on the Troubleshooting Dialog Box when the slave whose process data communications are not possible, is detected for Sysmac errors. To display the Sysmac errors, you need to change the state in which process data communications are possible for the slave. Therefore, if both the error that process data communications are not possible for the slave and Sysmac error occur at the same time, first remove the error that process data communications are not possible for the slave and then make the suitable correction. Check the Sysmac error for the slave after the process data communications are restarted with the slave. Then, remove the cause of the Sysmac error and reset the error again. For details on the error that process data communications are not possible for the slave, refer to *Errors in the EtherCAT Master Function Module* in the *NJ/NX-series Troubleshooting Manual (Cat. No. W503)*. You can find it in the operation cell for each event.
- If you delete the assignment of the Sysmac error status from the processing data communications data, e.g., to reduce the load on the EtherCAT communications line, you will not be able to use the Sysmac error status in troubleshooting. (To delete the assignment, edit the PDO map settings in the EtherCAT master settings on the Sysmac Studio so that the map object that is mapped to object 2002 hex is not selected.)
  If you do so, you can detect errors that occur in the slave applications other than communications errors only through the transmission of emergency messages or status information that is mapped for other process data communications. Use emergency messages and other status information for which PDOs are mapped to detect slave errors in this case. Transmission of emergency messages is disabled by default for OMRON slaves.

Additional Information

- When the EtherCAT master detects an emergency message from a slave, the *_EC_SlavEmergErr* system-defined variable changes to TRUE. Check the contents of the emergency message from the slave on the Troubleshooting Dialog Box.
- Refer to the *NJ/NX-series Troubleshooting Manual (Cat. No. W503)* for details on checking for errors and corrections.

Backing Up and Restoring Parameters

The Sysmac devices use an option in the version-1.0.1 ESI specifications to enable backing up and restoring the backup parameters that are specified in the ESI files in non-volatile memory in the slaves.

You can back up and restore the backup parameters that are stored from the Sysmac Studio.

This makes it easy to set slaves when they are replaced.

Additional Information

- If you use slaves from other manufacturers, obtain the slave information files that are compliant with the version-1.0.1 ESI specifications from the slave manufacturers.
- Refer to *9-4-3 Backing Up Settings* on page 9-28 for how to back up slave data from the Sysmac Studio.
- Refer to *9-4-4 Restoring Settings* on page 9-30 for how to restore slave data from the Sysmac Studio.
- Refer to *A-3 Multi-vendor Environments* on page A-15 for details on ESI.

Saving Node Address Settings

Each Sysmac device stores its own node address setting in non-volatile memory within the slave.

This allows the master to identify slaves on the network.

The node addresses are set as given below depending on the type of slave.

- Slaves with Hardware Switches

  The set value for the node address is determined depending on the set value for hardware switches.

  00: The node address is set from the Sysmac Studio.

  Not 00: The setting on the hardware switches is used.

- Slaves without Hardware Switches

  The node address is set from the Sysmac Studio.

Precautions for Correct Use

- Set a node address for all slaves so that they can be identified on the EtherCAT network. Do not assign the same node address more than once.
- If you use slaves from other manufacturers with project unit version earlier than 1.40, set the node addresses from the Sysmac Studio. For project unit version 1.40 or later, you can set the node addresses either from the Sysmac Studio or with hardware switches.

Additional Information

- Refer to *2-2 Setting the Node Addresses of the EtherCAT Slaves* on page 2-9 or the relevant manuals for each slave for how to set the node addresses.
- Refer to *5-5 EtherCAT Slave Parameter Settings* on page 5-20 for information on setting node addresses from the Sysmac Studio.

Verifying the EtherCAT Network Configuration Using Serial Numbers

Each Sysmac device slave stores its serial number in non-volatile memory within the slave.

If serial number verification is enabled in the EtherCAT master settings, the EtherCAT network configuration is verified based on the serial numbers when the EtherCAT master is started. The following are enabled by verification of the serial numbers.

- If the EtherCAT network configuration changes, the serial numbers will not match and a *Network Configuration Verification Error* (84220000 hex) or *Network Configuration Verification Error (Mismatched Slave)* (84330004 hex) event occurs. This helps prevent forgetting to set the parameters when a slave is replaced.
- The serial numbers of any of the slaves can be checked from the EtherCAT master.

Additional Information

Refer to *5-4 EtherCAT Master Parameter Settings* on page 5-14 for information on serial number verification.

2-3-3 List of Sysmac Devices

The following table lists the OMRON Sysmac slaves. Refer to relevant manuals for each slave for information on OMRON slaves that are not listed in the following table.

Name	Model	Revision
AC Servo Drives	R88D-KN□□□-ECT	Revision 2.1 or later
Multifunctional, Compact Inverter	3G3AX-MX2-ECT	Revision 1.1 or later
Digital I/O Slaves	GX-ID□□□□ GX-OD□□□□ GX-MD□□□□ GX-OC□□□□	Revision 1.1 or later
Analog I/O Slaves	GX-AD0□71 GX-DA0□71	Revision 1.1 or later
Encoder Input Slaves	GX-EC02□1	Revision 1.1 or later
Vision Sensors	FH-1□□□ FH-3□□□	Revision 1.0 or later
Specialized Vision Sensors for Positioning	FQ-MS12□-ECT FQ-MS12□-M-ECT	Revision 1.0 or later
Smart Fiber Sensor Communications Unit	E3X-ECT	Revision 1.0 or later
EtherCAT Junction Slaves	GX-JC03 GX-JC06	Revision 1.0 or later
NX-series EtherCAT Coupler Units	NX-ECC201[*1]	Revision 1.0 or later
	NX-ECC202[*2]	Revision 1.2 or later
	NX-ECC203[*3]	Revision 1.3 or later
Advanced General-purpose Inverters	3G3RX-V1 with EtherCAT Communications Unit 3G3AX-RX-ECT	Revision 1.0 or later
Linear Servo Drives	R88L-EC R88D-KN□-ECT-L	Revision 1.0 or later
Digital Sensor Communications Unit	E3NW-ECT	Revision 1.0 or later
Fiber Sensor Communications Unit	E3X-ECT	Revision 1.0 or later
Confocal Fiber Type Displacement Sensors	ZW-CE1□	Revision 1.0 or later

[*1] A CPU Unit with unit version 1.05 or later and Sysmac Studio version 1.06 or higher are required.

[*2] A CPU Unit with unit version 1.07 or later and Sysmac Studio version 1.08 or higher are required.

[*3] A CPU Unit with unit version 1.07 or later and Sysmac Studio version 1.11 or higher are required.

2-3-4 Sysmac Device Features and EtherCAT Masters

The following table shows the relationship between Sysmac device features and EtherCAT masters.

OMRON Sysmac Device Slaves

Sysmac device feature	OMRON EtherCAT master		EtherCAT master from another manufacturer
	NJ/NX-series CPU Unit NX701-□□□□ NX502-□□□□ NX102-□□□□ NX1P2-□□□□□□□ NJ501-□□□□ NJ301-□□□□ NJ101-□□□□	CJ-series Position Control Unit CJ1W-NC□8□	
Troubleshooting	Supported.	Not supported.[*1]	Not supported.[*1]

Sysmac device feature	OMRON EtherCAT master		EtherCAT master from another manufacturer
	NJ/NX-series CPU Unit NX701-□□□□ NX502-□□□□ NX102-□□□□ NX1P2-□□□□□□□ NJ501-□□□□ NJ301-□□□□ NJ101-□□□□	CJ-series Position Control Unit CJ1W-NC□8□	
Backing up and restoring parameters	Supported.[*2]	Partially supported.[*3]	Partially supported.[*4]
Saving node address settings	Supported.	Supported.	Partially supported.[*5]
Verifying the EtherCAT network configuration	Supported.	Not supported.	Partially supported.[*6]

*1. Errors are notified with emergency messages.

*2. EtherCAT slaves for which data can be backed up have specific precautions. Refer to *EtherCAT Slaves for Which You Can Back Up Data* in the *NJ/NX-series CPU Unit Software User's Manual (Cat. No. W501)* for details on the precautions.

*3. You cannot specify data to back up and restore.

*4. Available backup and restore functions depend on the EtherCAT master from another manufacturer that you use.

*5. The node address setting method depends on the EtherCAT master from another manufacturer that you use.

*6. Whether the serial number can be checked or not depends on the EtherCAT master from another manufacturer that you use.

OMRON Slaves That Do Not Support Sysmac Devices

Sysmac device feature	OMRON EtherCAT master		EtherCAT master from another manufacturer
	NJ/NX-series CPU Unit NX701-□□□□ NX502-□□□□ NX102-□□□□ NX1P2-□□□□□□□ NJ501-□□□□ NJ301-□□□□ NJ101-□□□□	CJ-series Position Control Unit CJ1W-NC□8□	
Troubleshooting	Not supported.[*1]	Not supported.[*1]	Not supported.[*1]
Backing up and restoring parameters	Supported.[*2]	Partially supported.[*3]	Partially supported.[*4]
Saving node address settings	Partially supported.[*5]	Partially supported.[*5]	Partially supported.[*6]
Verifying the EtherCAT network configuration	Not supported.	Not supported.	Partially supported.[*7]

*1. Errors are notified with emergency messages.

*2. EtherCAT slaves for which data can be backed up have specific precautions. Refer to *EtherCAT Slaves for Which You Can Back Up Data* in the *NJ/NX-series CPU Unit Software User's Manual (Cat. No. W501)* for details on the precautions.

*3. You cannot specify data to back up and restore.

*4. Available backup and restore functions depend on the EtherCAT master from another manufacturer that you use.

*5. Setting is possible on the node address switches. Node addresses set with the software cannot be stored in the slave.

*6. The node address setting method depends on the EtherCAT master from another manufacturer that you use.

*7. Whether the serial number can be checked or not depends on the EtherCAT master from another manufacturer that you use.

Slaves from Other Manufacturers

If you connect slaves from other manufacturers to an OMRON EtherCAT master, functionality is restricted as given below depending on the functionality of the slaves.

- Some items for slave function setting may be missing or the slave functionality may be restricted with an ESI file that is not supported by Sysmac Studio. The slave may not operate properly.
- The node address setting from the Sysmac Studio may not be supported by some slave specifications.
- Verification with serial numbers is not possible for slaves that do not have the serial number in SII.

3

EtherCAT Communications

This section describes the different types of EtherCAT communications, EtherCAT settings, and state transitions.

3-1 EtherCAT Communications Types and Settings

3-1-1 CoE (CAN Application Protocol over EtherCAT)

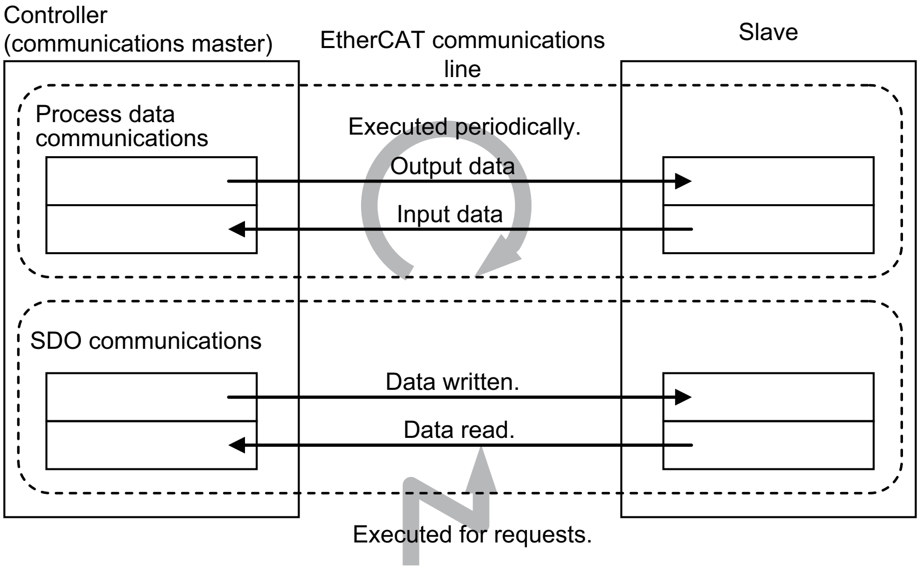
In the built-in EtherCAT port of the NJ/NX-series CPU Unit, CoE (CAN application protocol over EtherCAT) is used as a protocol for exchanging data with the slaves on EtherCAT.
CoE implements message communications of CAN application over an EtherCAT network.
With CoE, the parameters and control information held by the slaves are specified according to data specifications for the object dictionary.

Process Data Communications and SDO Communications

There are two communications that are used to communicate data between the master and slaves. These are described below.

Process Data Communications
Communications using process data objects (PDOs) to exchange information in realtime with a fixed period.

SDO communications
Communications using service data objects (SDOs) for communicating information when required.



The built-in EtherCAT port in the NJ/NX-series CPU Unit uses process data communications for commands to refresh I/O data, such as I/O data for EtherCAT slaves and data for Servomotor position control, on a fixed control period. It uses SDO communications for commands to read or write data at specified times, such as for parameter transfers.

Process data communications are executed each control cycle to refresh data continuously, and SDO communications are executed between PDO communications.

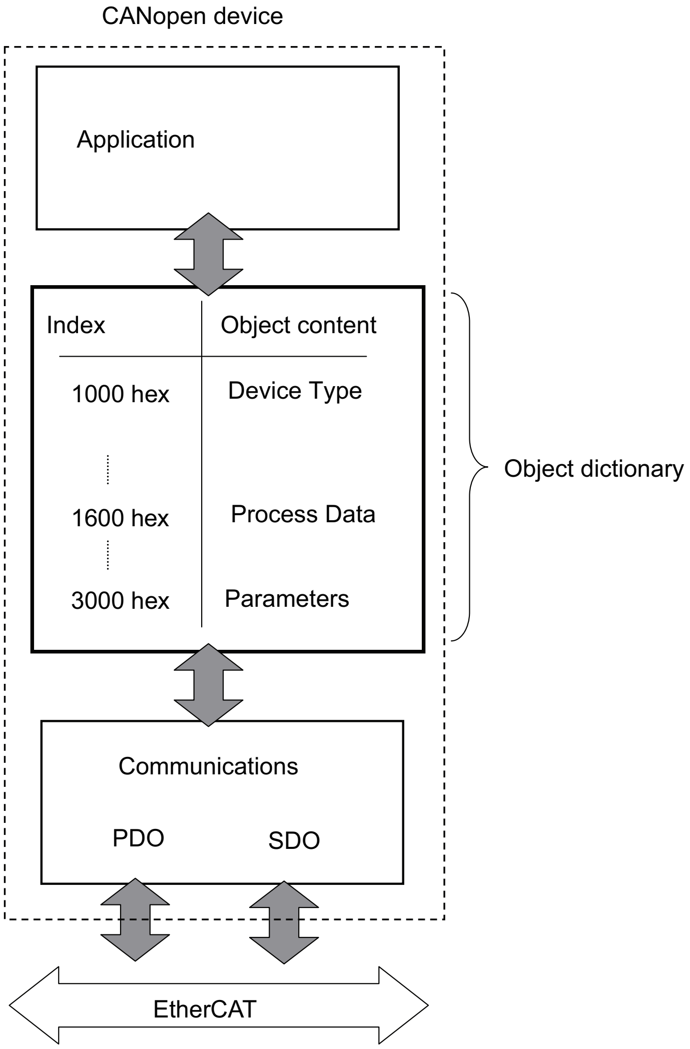
Additional Information

Object Dictionary

The object dictionary is a data table within the device that interfaces the application and communications. It describes the information handled by the device. Each piece of information is called an object.

Each object is assigned a 4-digit hexadecimal index.

Each object is divided into a data type area, a CoE communications area, and a manufacturer-specific area.



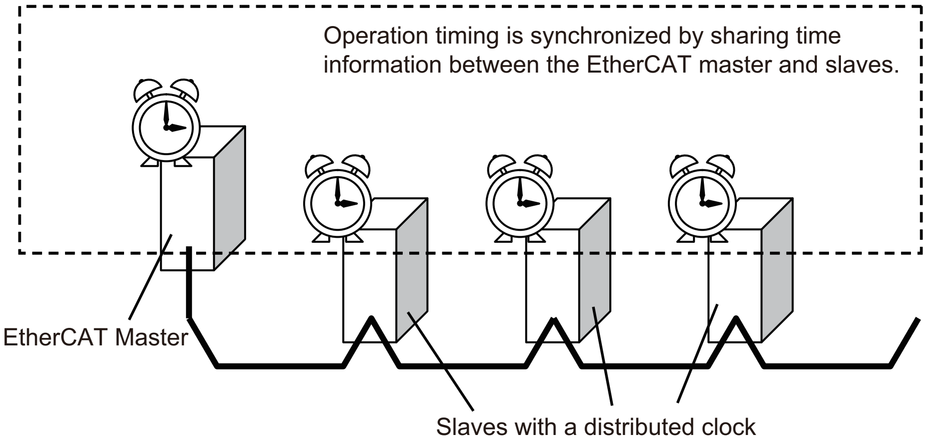
Process data communications is used to read and write the process data in the CoE communications area in these objects.

SDO communications is used to read and write the parameters within the manufacturer-specific area in these objects.

Additional Information

Distributed Clock (DC)

This is a unique EtherCAT feature that enables precisely clock synchronization.
The DC-based clock synchronization enables sharing the same time between the EtherCAT master and the slaves.
This enables the timing of the operation of the EtherCAT master and slaves to be synchronized with the shared time.


Operation timing is synchronized by sharing time information between the EtherCAT master and slaves.
EtherCAT Master
Slaves with a distributed clock

The clock that gives the standard network time is called a "reference clock".
The reference clock is provided either by the master or a slave that supports the function to provide a reference clock, to the other slaves. As described in the table below, whether the master or slave provides a reference clock depends on the used CPU Unit.

When NX-series CPU Unit is Used	When NJ-series CPU Unit is Used
The EtherCAT master of the NX-series CPU Unit	Of the slaves that provide a reference clock, the slave that is the closest to the EtherCAT master on the wiring. In the above figure, the left most slave of the slaves with the DC functionality. *1

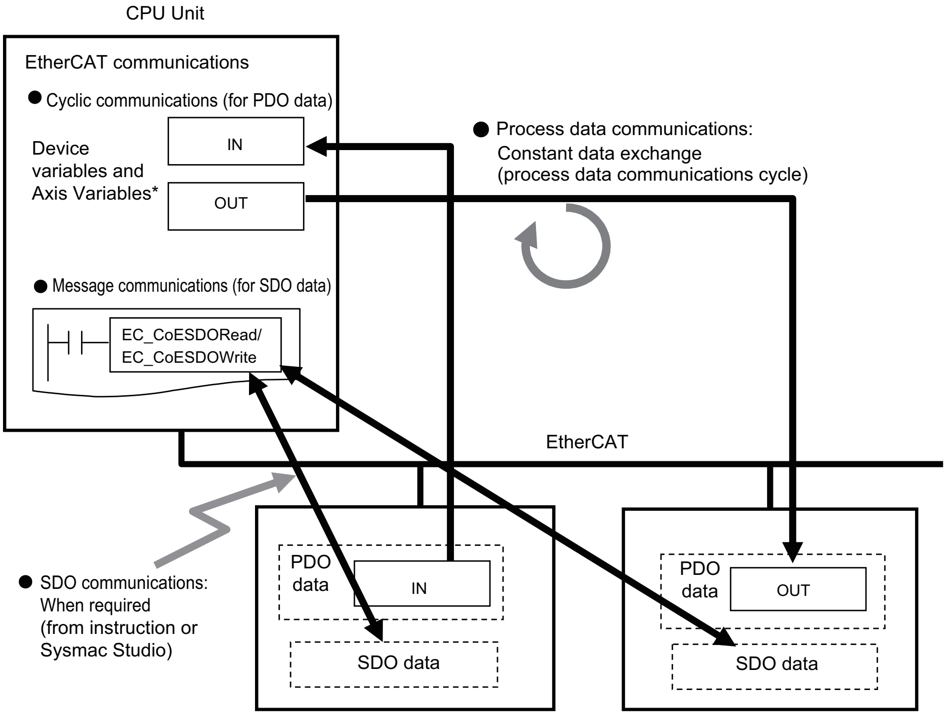
*1. Slaves with the DC functionality can provide a reference clock.

The EtherCAT master uses this reference clock to send frames so that the EtherCAT master and slaves remain synchronized.
The slaves for which the distributed clock setting is enabled (i.e., synchronized slaves) perform advanced time synchronization by performing input and outputs based on the reference clock.

3-1-2 Types of Communications

The following two methods are used to exchange data between master and slaves in EtherCAT communications.

Communication type	Name of communication type	Timing of processing	Type of data
Cyclic communications	Process data communications (PDO communications)	Constant (process data communications cycle)	PDO data
Message communications	SDO communications	When required.	SDO data



* Device variables for EtherCAT slaves
Axis Variables only for Servo Drive and encoder input slaves to which axes are allocated

Process Data Communications (PDO Communications)

PDO communications is used for constant data exchange between the master and slaves. It is called "process data communications".

PDO data (i.e., I/O data that is mapped to PDOs) that is allocated in advance is input and output periodically each EtherCAT process data communications cycle (i.e., the task period of primary periodic task or priority-5 periodic task).

It is accessed from the NJ/NX-series CPU Unit in the following ways.

- With device variables for EtherCAT slave I/O
- With Axis Variables for Servo Drive and encoder input slaves to which an axis is allocated

Additional Information

You can use the priority-5 periodic task only with NX701 CPU Units.

SDO Communications

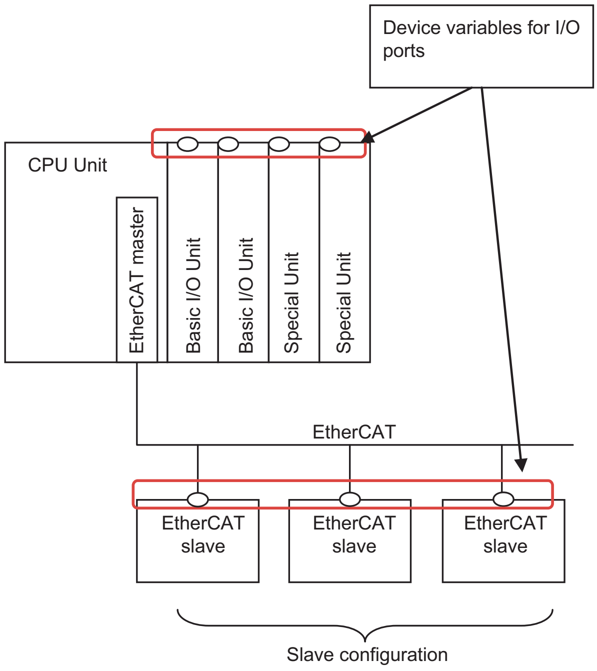
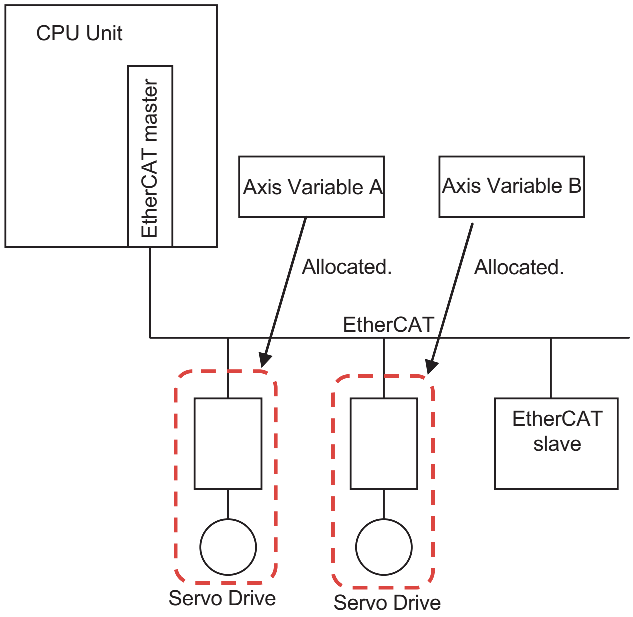
SDO communications are used to read and write specified slave data from the master when required. This is called "SDO communications".

You can read/write the following specified slave data with the EC_CoESDORead (Read CoE SDO) instruction or the EC_CoESDOWrite (Write CoE SDO) instruction.

- SDO data in slaves (parameters, error information, etc.)

3-1-3 Types of EtherCAT Variables

There are three types of EtherCAT variables as listed below.

Variable type		Description
Allocated variables	Device variables	These are common in-out variables for EtherCAT slaves. You can change the names of these variables.  Device variables for I/O ports CPU Unit EtherCAT master Basic I/O Unit Basic I/O Unit Special Unit Special Unit EtherCAT EtherCAT slave EtherCAT slave EtherCAT slave Slave configuration
	Axis Variables	These are structure variables for I/O and parameters for Servo Drive and encoder input slaves. You can change the names of these variables.  CPU Unit EtherCAT master Axis Variable A Axis Variable B Allocated. Allocated. EtherCAT EtherCAT slave Servo Drive Servo Drive

Variable type		Description
System-defined variables	System-defined variables for EtherCAT master	These variables are defined by the system for communications parameters, communication status, and other functions. You cannot change the names of these variables.

3-1-4 Settings Required for EtherCAT Communications

For EtherCAT communications, you must set the following network configuration information from the Sysmac Studio and download it to the CPU Unit.

- Network configuration: Master and slave configuration
- EtherCAT master settings: Parameters settings, such as the Fail-soft Operation Setting or Wait Time for Slave Startup
- Process data information: Allocation information for slave PDO data

Additional Information

EtherCAT communications cannot be performed unless the correct network configuration information is downloaded to the CPU Unit.
Refer to *Section 5 Setting Up EtherCAT Communications with the Sysmac Studio* on page 5-1 for details.

3-2 Programming EtherCAT Communications

The user program in the NJ/NX-series CPU Unit reads/writes EtherCAT slave data and performs motion control for Servo Drive and encoder input slaves.

Instructions and variables are used according to slave types and target data as shown below.

Slave type	Type of data	Timing	Instructions	Variables
Slaves to which you cannot assign axes	**Process data**	Constantly read/ written	Read/write instructions such as LD, OUT and MOV	Device variables
	SDO data	Read/written as required	EtherCAT communications instructions (EC_CoESDORead or EC_CoESDOWrite)	User-defined variables
Slaves to which you can assign axes	**Process data**	Constantly read/ written	Motion control instructions or read/write instructions such as LD, OUT and MOV	Axis Variables or device variables
	SDO data	Read/written as required	EtherCAT communications instructions (EC_CoESDORead or EC_CoESDOWrite)	User-defined variables

3-3 State Transitions for EtherCAT Communications

3-3-1 Self Diagnosis at Startup

The EtherCAT master executes the following self-diagnosis when the power is turned ON.
The results of self-diagnosis are provided in the following events and system-defined variables as EtherCAT master errors if errors are detected.

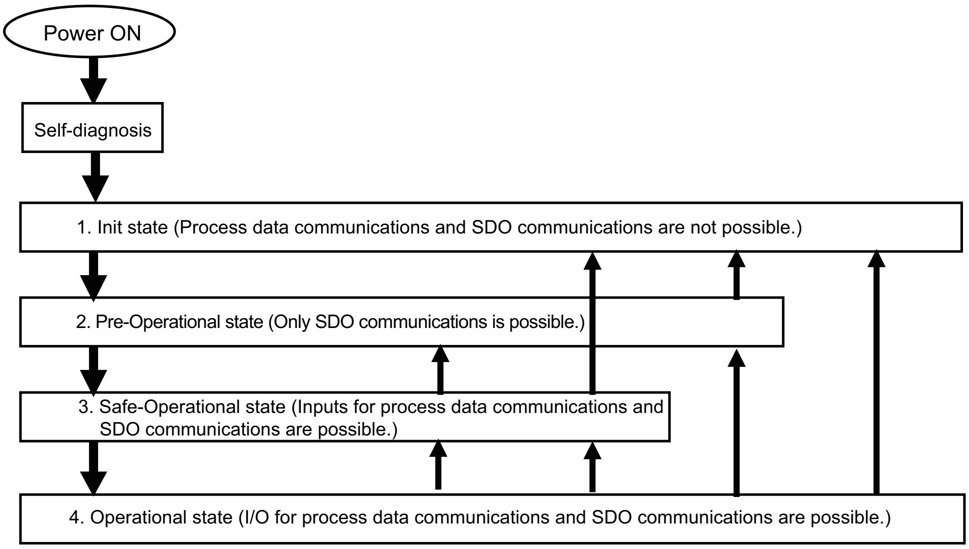
Diagnosis	Notification in case of an error detected	
	Events	System-defined variables
Diagnosis of network configuration information	Network Configuration Information Error (34400000 hex)	_EC_NetCfgErr
Diagnosis of communications port	MAC Address Error (14400000 hex)	_EC_MacAdrErr
	Communications Controller Error (04400000 hex)	_EC_LanHwErr

Refer to *Errors in the EtherCAT Master Function Module* in the *NJ/NX-series Troubleshooting Manual (Cat. No. W503)* for details on events.

3-3-2 Control States for EtherCAT Communications

Control State Machine

EtherCAT communications provides four control states. Communications is controlled by moving between these states.


Power ON
Self-diagnosis
1. Init state (Process data communications and SDO communications are not possible.)
2. Pre-Operational state (Only SDO communications is possible.)
3. Safe-Operational state (Inputs for process data communications and SDO communications are possible.)
4. Operational state (I/O for process data communications and SDO communications are possible.)

After the power is turned ON, the communications master and slaves go from the Init state to the Pre-Operational state, Safe-Operational state, and then Operational state before starting EtherCAT communications.

Afterwards, EtherCAT communications are performed while the state changes automatically between these states according to error occurrence and other conditions.

The current control state can be determined using the RUN indicator on the front panel.

State	Description	Process data communications	SDO communications	RUN indicator
1: Init	Communications are being initialized. Communications are not possible. This state continues if there is not the network configuration information.	Not allowed	Not allowed	Not lit
2: Pre-Operational	Only SDO communications are possible in this state. Communications always enters this state after initialization, and changes to the Safe-Operational state after initial setting of the network is performed.	Not allowed	Possible	Flashing
3: Safe-Operational	In this state, SDO communications and only inputs for process data communications are possible. Communications always enters this state before going into the Operational state.	Only inputs are possible	Possible	Flashing
4: Operational	This is the normal state for communications.	Possible	Possible	Lit

Additional Information

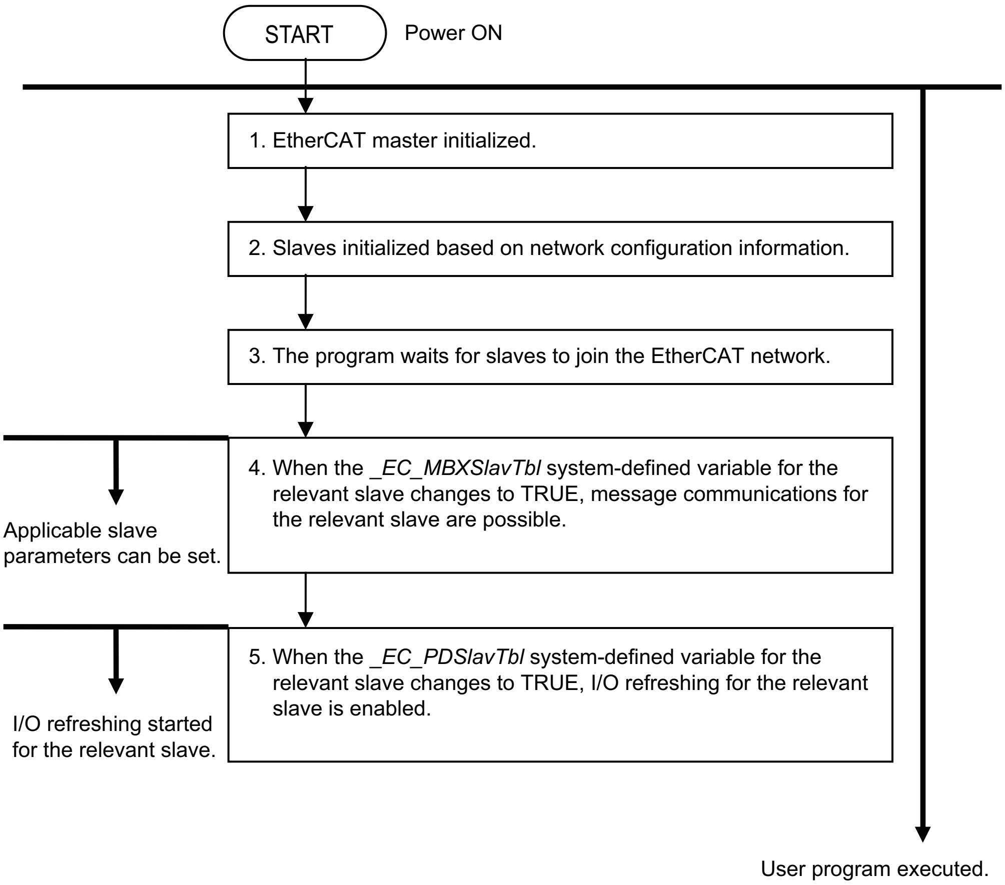
I/O Refreshing

The procedure from startup of the EtherCAT network until process data I/O can be refreshed is shown below.

There is no correlation between the startup of the EtherCAT network and the execution of the user program. Design the user program by adding the system-defined variables* for the relevant slaves to the interlock conditions of the device variables for the slaves.

* _EC_MBXSlavTbl[1..n], _EC_PDSlavTbl[1..n]

Here, n is the maximum value of the settable node address. Refer to *1-3 Specifications of Built-in EtherCAT Port* on page 1-10 for the maximum value of the settable node address.


START
Power ON
1. EtherCAT master initialized.
2. Slaves initialized based on network configuration information.
3. The program waits for slaves to join the EtherCAT network.
4. When the _EC_MBXSlavTbl system-defined variable for the relevant slave changes to TRUE, message communications for the relevant slave are possible.
Applicable slave parameters can be set.
5. When the _EC_PDSlavTbl system-defined variable for the relevant slave changes to TRUE, I/O refreshing for the relevant slave is enabled.
I/O refreshing started for the relevant slave.
User program executed.

3-3-3 EtherCAT Status in Relation to CPU Unit Status

Refer to *A-1 EtherCAT Status in Relation to CPU Unit Status* on page A-2 for details on the following: memory related to the EtherCAT master, the ability to download master settings and slave settings, and the status of slaves according to the CPU Unit operating mode and the status of Controller errors.

4

EtherCAT Network Wiring

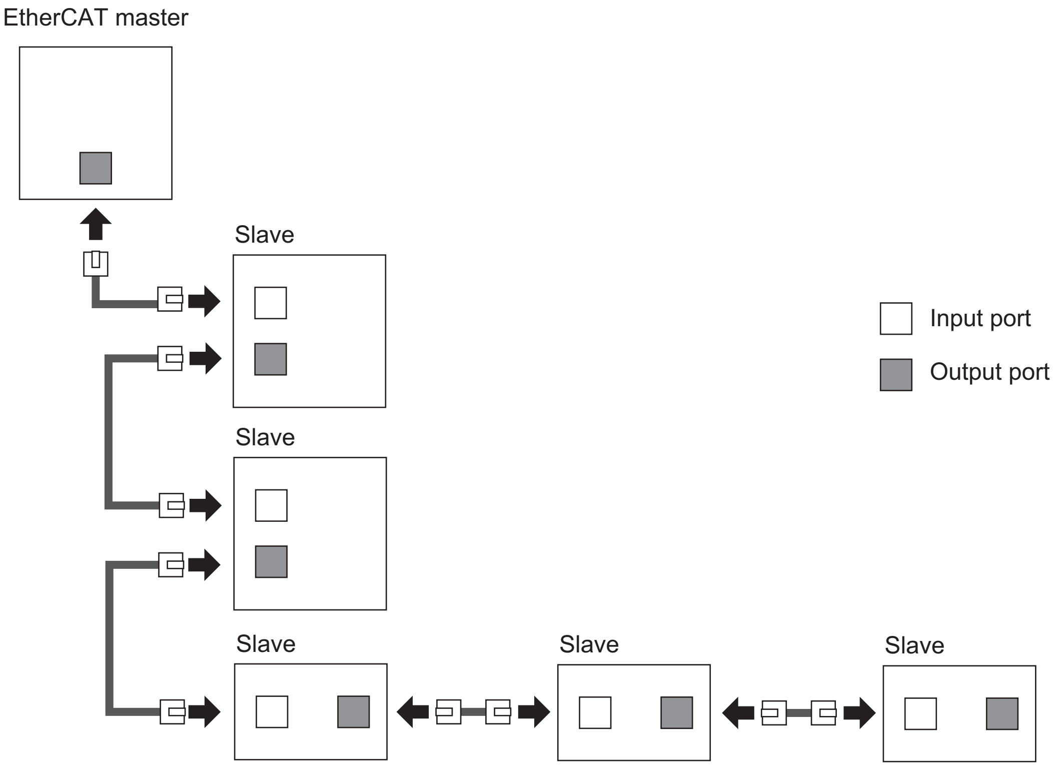
This section describes how to connect and wire an EtherCAT network.

4-1 Supported Network Topologies

The NJ/NX-series CPU Units support line/daisy chain, branching, and ring topologies. You can combine more than one topology.
Each of these topologies is described below.

4-1-1 Line/Daisy Chain Topology

This topology connects the output ports and input ports of slaves one after another in a row.

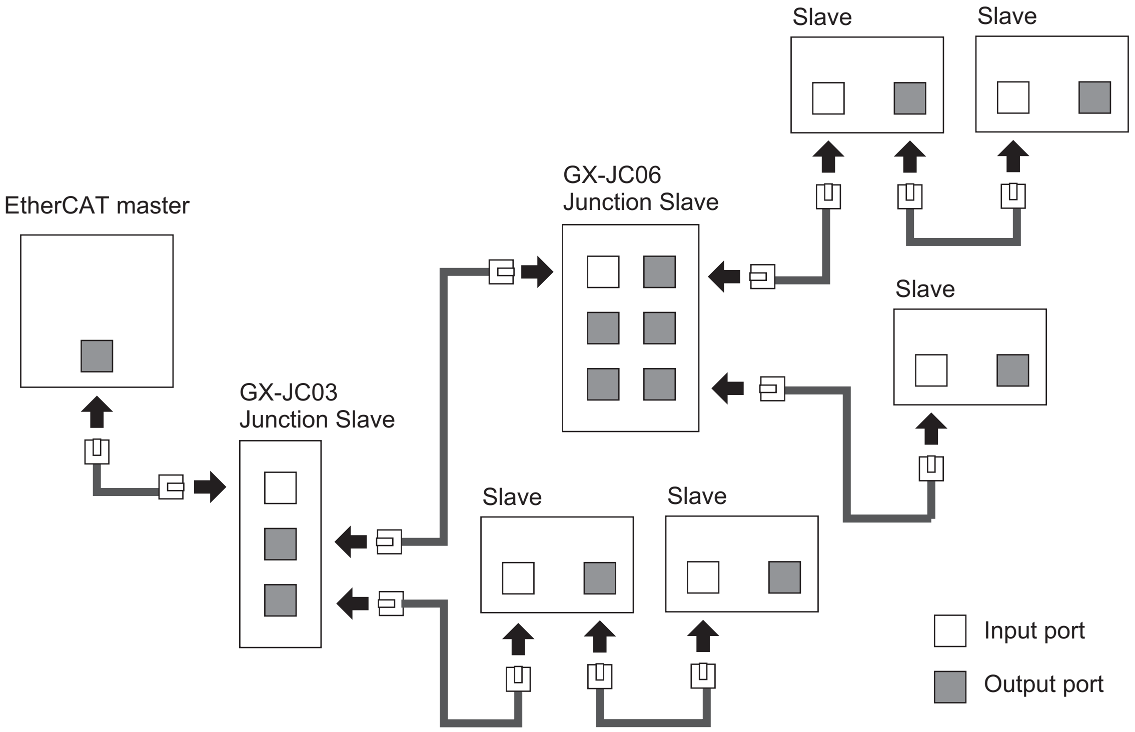


Features of Line/Daisy Chain Topology

The master and slaves can be connected easily. However, if a slave is disconnected due to a broken cable or other reason, all slaves after the disconnected slave are also disconnected from the network.

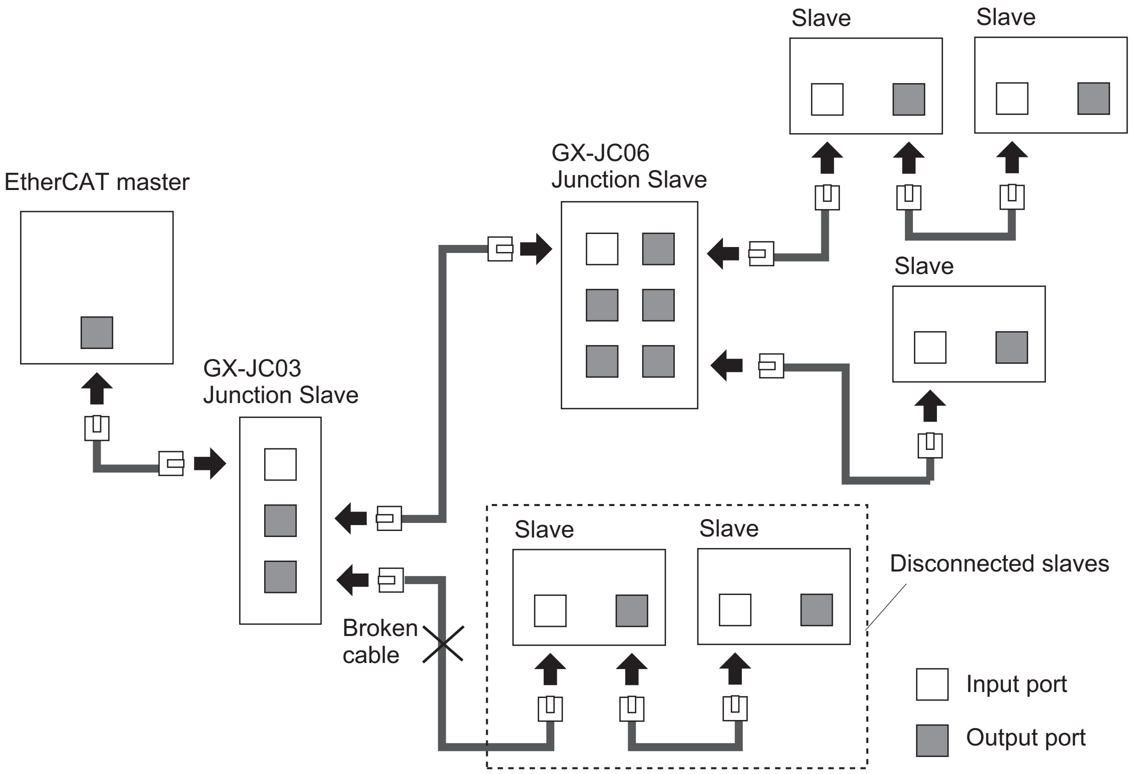
4-1-2 Branching Topology

This topology uses Junction Slaves to configure more than one line/daisy chain.



Features of Branching Topology

In a configuration that consists of only a line/daisy chain topology, if a slave is disconnected due to a broken cable or other reason, all slaves after the disconnected slave are also disconnected from the network. If you configure a network with the branching topology, you can reduce the disconnected slaves to only those in a branched path.



4-1-3 Ring Topology

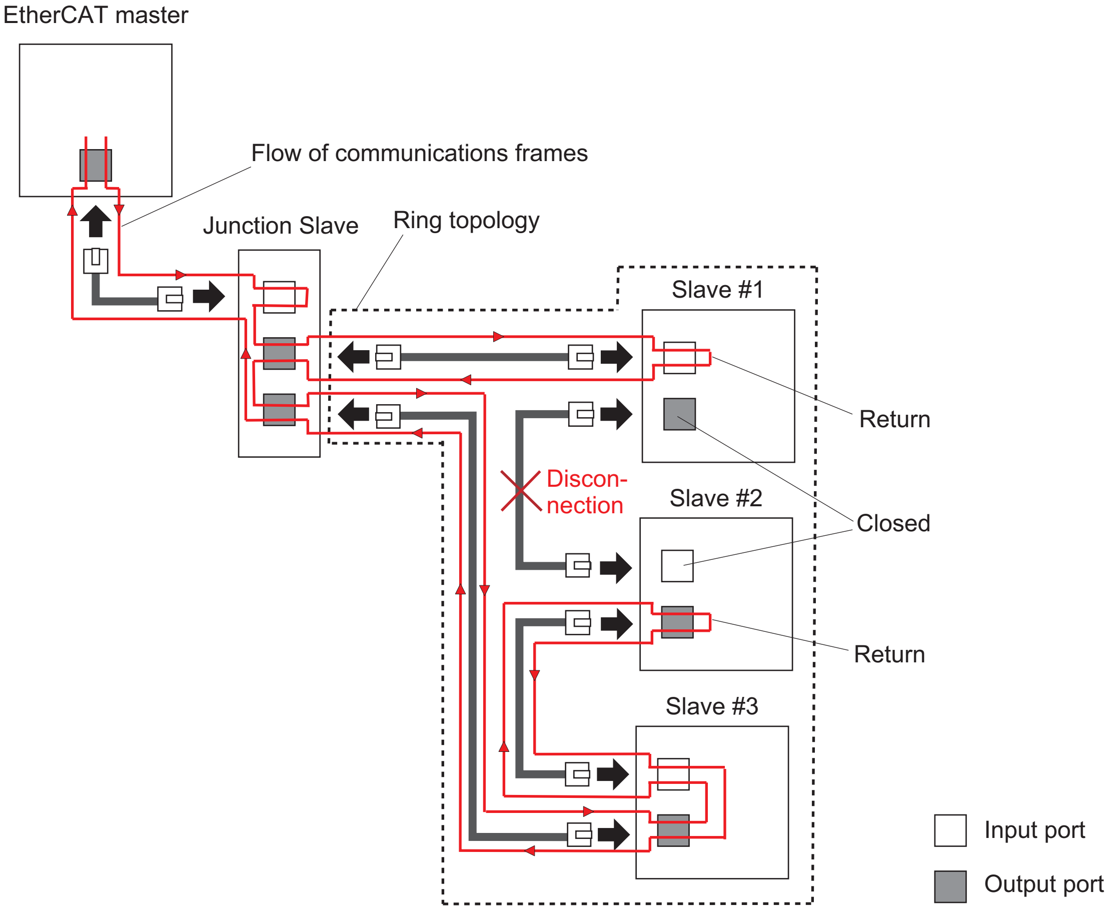
This topology uses Junction Slaves to connect slaves in a ring.
This section describes the features, components, and precautions for the ring topology.

Version Information

Project unit version 1.40 or later and Sysmac Studio version 1.29 or higher are required to use the ring topology.

Features of Ring Topology

In a ring topology, the master can continue the normal communications with all the slaves even if there is only one point of disconnection in the communications cables.
If disconnection occurs at a point between Slave #1 and Slave #2 as shown in the figure below, in Slave #1, the output port is closed, but communications frames return at the input port. The communications frames then flow via the Junction Slave into Slave #3 and return at the output port of Slave #2.
This enable the master to continue the normal communication.



Precautions for Correct Use

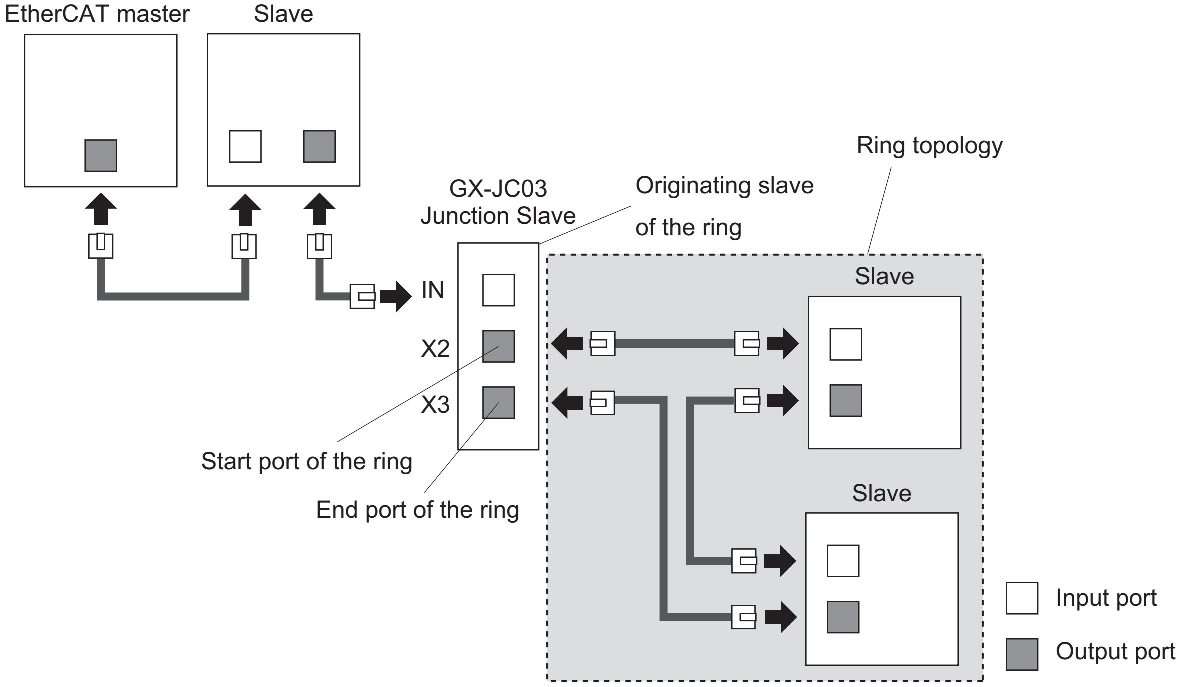
- If there is more than one point of disconnection in the communications cables in the ring topology, the master stops communications with slaves to which the communications paths are broken with the Junction Slave. It continues communications with other slaves. In this case, an *Illegal Slave Disconnection Detected* (84310002 hex) event occurs.
- If a power interruption occurs only at one slave in the ring topology, the master can continue the normal communications with slaves at which the power interruption does not occur in the ring topology. In this case, an *Illegal Slave Disconnection Detected* (84310002 hex) event occurs.
- If a power interruption occurs at more than one slave in the ring topology, the master stops communications with the slaves at which the power interruption occurred and slaves to which the communications paths are broken due to the power interruption. It continues communications with other slaves. In this case, an *Illegal Slave Disconnection Detected* (84310002 hex) event occurs.

Components of the Ring Topology

The components of the ring topology are shown below.

The following table describes the terms and definitions used for the explanation of the components of ring topology.

Term	Definition
Originating slave of the ring	A Junction Slave that is the starting point of the ring topology. This includes the start port of the ring and the end port of the ring.
Start port of the ring	A port that is the start point of the ring topology. The start port of the ring is X2 for the GX-JC03.
End port of the ring	A port that is the end point of the ring topology. The end port of the ring is X3 for the GX-JC03.



● Junction Slave

a. An EtherCAT configuration can have only one Junction Slave that becomes the originating slave of the ring. Use the OMRON GX-JC03 EtherCAT Junction Slave.

Precautions for Correct Use

To use a Junction Slave as the originating slave of the ring, set **Cable Redundancy** to **Enable** from the Sysmac Studio. Refer to *5-2-2 Cable Redundancy Setting* on page 5-5 for how to set **Cable Redundancy** to **Enable**.

b. You can use more than one Junction Slave in a ring topology. Use the OMRON GX-JC03 EtherCAT Junction Slave.

Precautions for Correct Use

Use the OMRON EtherCAT Junction Slaves with unit version 1.1 or later.

● Slave

In a ring topology, use slaves that support the ring topology.

If you use OMRON slaves, refer to the user's manual for each slave for information on topologies, and check that the slaves support the ring topology. Check also that their versions are supported.

Version Information

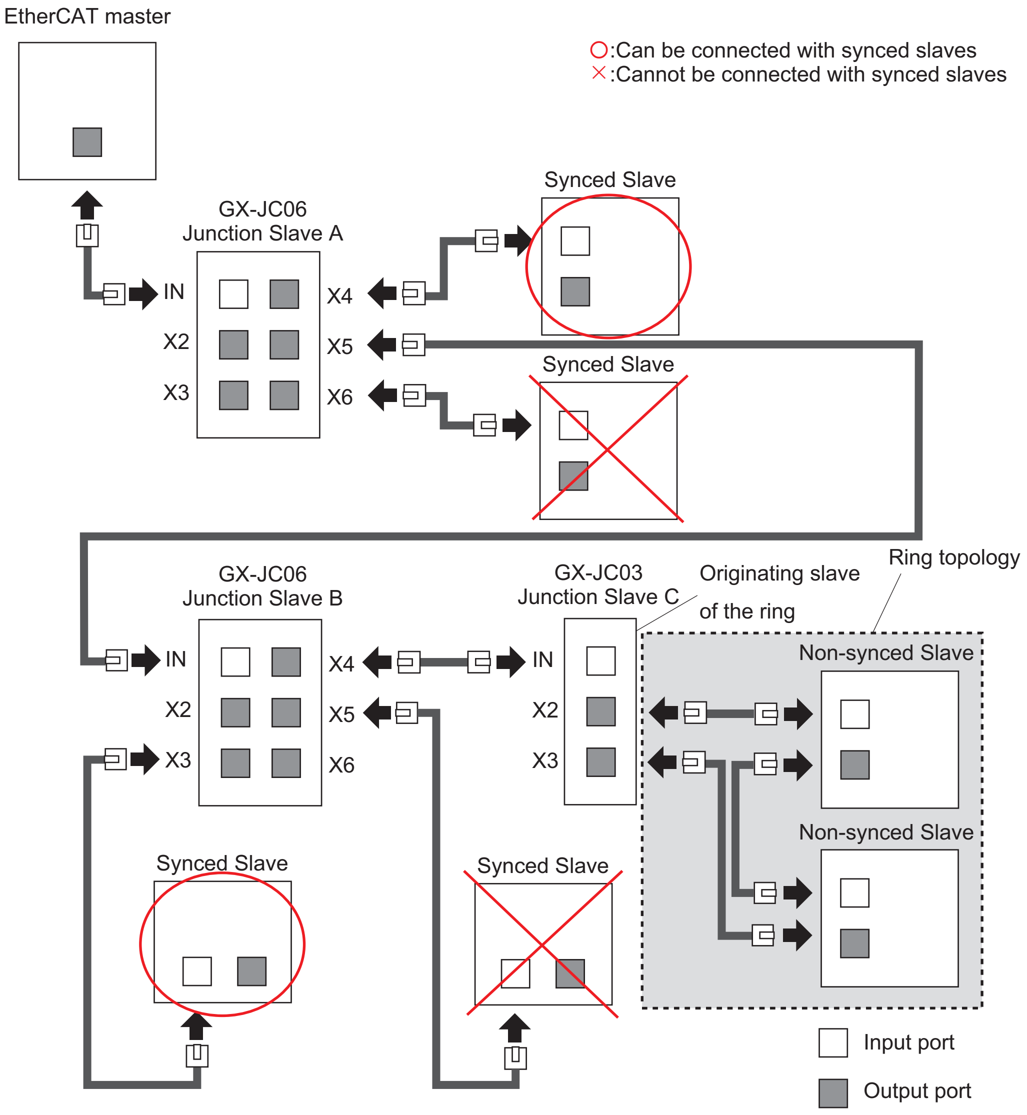
For project unit version earlier than 1.42, if the Junction Slave is connected between the EtherCAT master and the originating slave of the ring, synced slaves cannot be connected to the ports that are not the following output ports of the Junction Slaves.

- Output port numbers smaller than those for the Junction Slave connected to the input port of the originating slave of the ring
- Output port numbers smaller than those for the Junction Slave connected to input ports for the Junction Slave between Junction Slaves if more than one Junction Slave is connected

This section describes an example to which two Junction Slaves are connected.

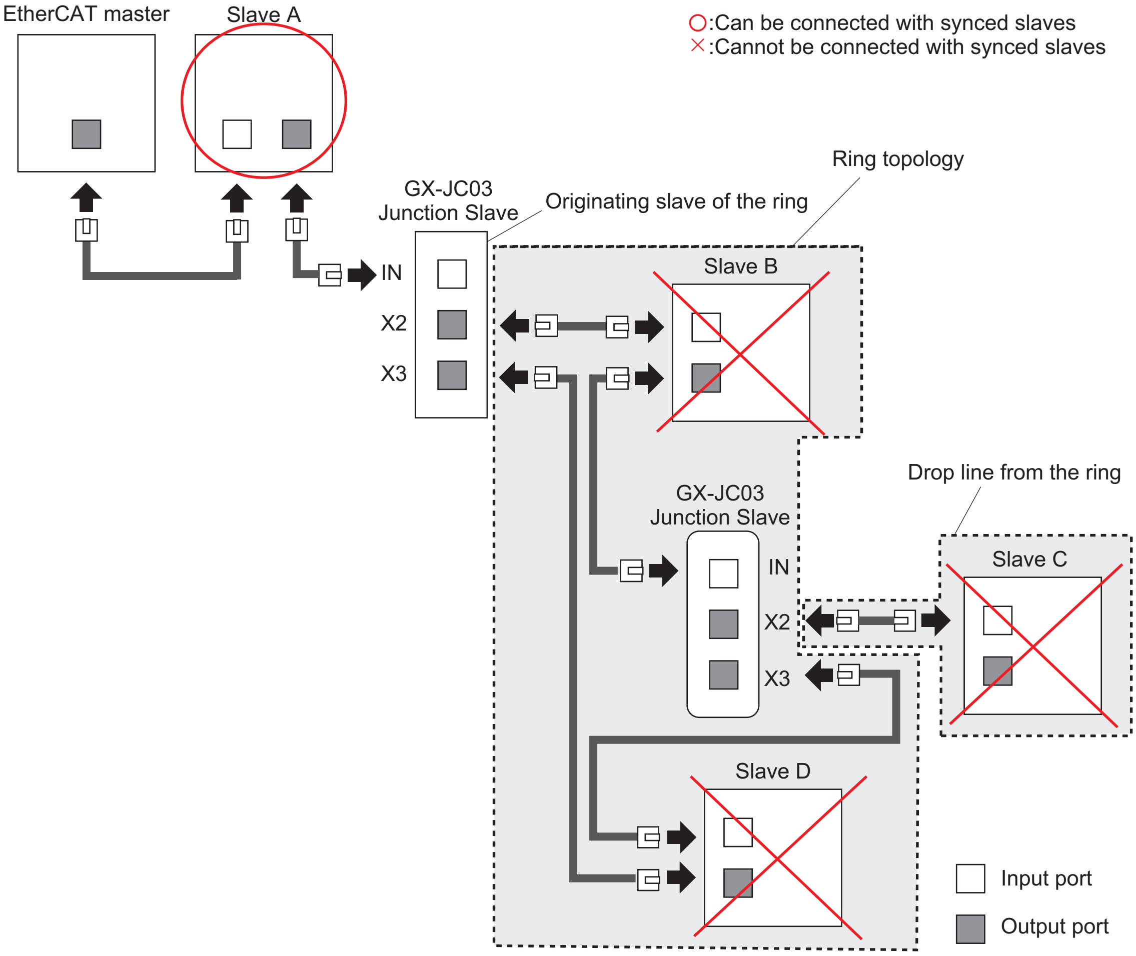
In the following figure, you can connect a synced slave to X3 for Junction Slave B because the input port of the originating slave of the ring is connected to X4 for Junction Slave B. However, you cannot connect a synced slave to X5 for Junction Slave B.

You can also connect a synced slave to X4 for Junction Slave A because the input port of the Junction Slave B is connected to X5 for Junction Slave A. However, you cannot connect a synced slave to X6 for Junction Slave A.



Version Information

For project unit version earlier than 1.42, you cannot connect synced slaves in a ring topology and in a drop line from the ring.
Slave B, C, and D shown in the following figure are "in the ring topology" or "in the drop line from the ring", so you cannot connect the synced slaves.



Precautions for Correct Use

Some slaves can select to operate the synced slave or the non-synced slave. For project unit version earlier than 1.42, if you connect this slave in the ring topology and in the drop line from the ring, set distributed clock setting to **Disabled** on the Sysmac Studio so that the slave will operate as a non-synced slave.
For example, for the EtherCAT Coupler Units, the initial value of the distributed clock setting is **Enabled**, so you must change the value to **Disabled**.

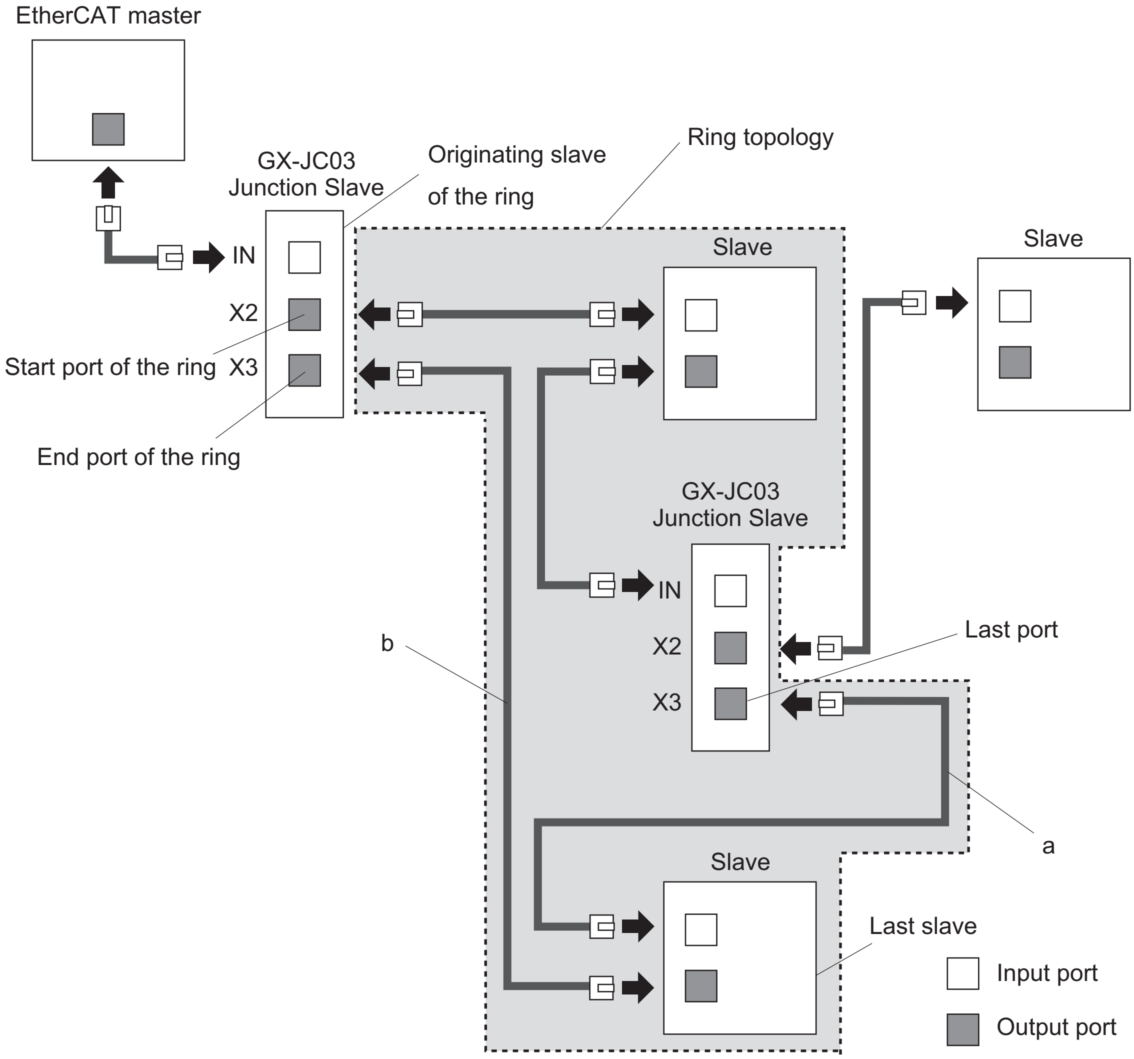
Configuring Ring Topology

The procedures to configure a ring topology are shown below.

a. If you use a Junction Slave in a ring topology, connect the last port of the Junction Slave to the input port of the next slave to connect to in the ring topology.
The last port is the last to receive communications frames among multiple output ports on a Junction Slave. The last port is X3 for the GX-JC03 Junction Slave.

b. In the slaves that are connected in series from the start port of the ring in a ring topology, connect the output port of the last slave in the ring topology to the end port of the ring.

The above examples of a and b are given in the figure below.



Precautions for Using a Ring Topology

The precautions for using a ring topology are described below.

Precautions for Correct Use

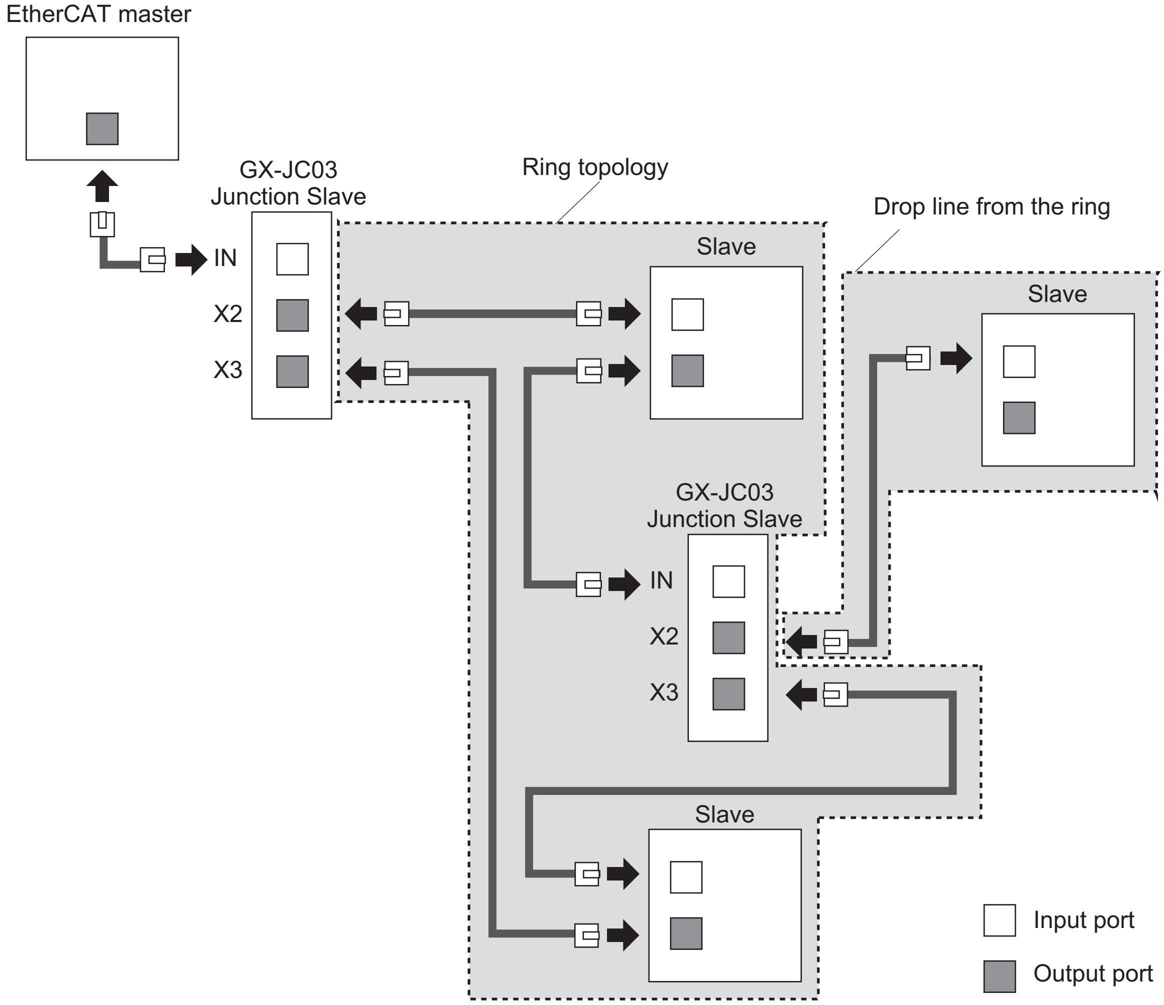
- If a power OFF or disconnection occurs in a slave that is outside the ring topology or an error in the EtherCAT master is reset, when there is a synced slave in the actual network configuration, the cable redundancy status is reset temporarily for DC Synchronous Correction.
- The Communications Error Setting parameters must be set for all of the OMRON synced slaves in the EtherCAT network. Refer to *Precautions for Using Cable Redundancy Function* on page 5-8 for details.
- When the wiring for slaves in the ring topology is incorrect and input ports or output ports are connected each other, an EtherCAT Frame Not Received will occur and message communications and process data communications may stop. Therefore, make sure that the wiring of communications cables for slaves is correct.

Additional Information

- You can configure a ring topology in the EtherCAT network.
- Use the OMRON GX-JC03 EtherCAT Junction Slave for the originating slave of the ring.
- If a Junction Slave is connected in the ring topology, use the OMRON GX-JC03 EtherCAT Junction Slave.
- If the OMRON slaves are connected in the ring topology, refer to the user's manual for each slave for information on topologies, and check that the slaves support the ring topology. Check also that their versions are supported.

Example for Configuring Drop Line from the Ring with Junction Slaves

You can connect Junction Slaves in a ring topology to configure a drop line from the ring.



4-1-4 Precautions for Using Junction Slaves

This section provides precautions for using Junction Slaves in a network configuration based on the branching or ring topology.

Precautions for Safe Use

You cannot use standard Ethernet hubs or repeater hubs with EtherCAT communications. If you use one of these, a major fault level error or other error may occur.

Precautions for Correct Use

When you use a system configuration in which an EtherCAT Junction Slave is connected to the EtherCAT master to synchronize between synced slaves, you need a EtherCAT master or slave that can provide a reference clock.

With the NX-series CPU Unit, the EtherCAT master provides a reference clock.
With the NJ-series CPU Unit, at least one of the following two conditions must be met because the EtherCAT master does not provide a reference clock.

- A slave that provides a reference clock must be connected between the EtherCAT master and the first EtherCAT Junction Slave.
- The EtherCAT Junction Slave must provide a reference clock.

However, the above conditions do not apply when synchronization is not performed between synced slaves in the entire EtherCAT network.

You can determine whether the EtherCAT Junction Slave supports a reference clock from the **Reference Clock** in the slave settings on the Sysmac Studio.
OMRON EtherCAT Junction Slaves support a reference clock.

Precautions for Correct Use

For project unit version 1.40 or later, check that the EtherCAT Junction Slave ports actually connected to slaves agree with the ports of the network configuration information on the Sysmac Studio. If the ports do not agree, a communications error occurs. You can perform the compare and merge operation in the Sysmac Studio to check whether the ports agree.
For project unit version earlier than 1.40, an error does not occur and the CPU Unit can continue the communications even when the EtherCAT Junction Slave ports actually connected to slaves do not agree with the ports of the network configuration information on the Sysmac Studio.

4-2 Laying the EtherCAT Network

This section describes how to install EtherCAT networks.

4-2-1 Installation Precautions

Basic precautions for the installation of EtherCAT networks are provided below.

Precautions when Installing a Network

- When you install an EtherCAT network, take sufficient safety precautions and perform the installation according to standards and specifications. (Refer to *JIS X5252* or to *electrical facility technical references*.)
  An expert well versed in safety measures and the standards and specifications should be asked to perform the installation.
- Do not install EtherCAT network equipment near sources of noise.
  If the network must be installed in an area with noise, take steps to address the noise, such as placing equipment in metal cases.
- When using a shielded cable with the shields on both ends of the cable connected to connector hoods, ground loops induced by improper grounding methods may decrease noise immunity and cause device damage. To prevent ground loops caused by differences in potential between device grounding points, the reference potential between the devices must be stabilized. Design grounding appropriately so that noise current does not flow to ground lines between the devices.
  For grounding methods, refer to the *NJ-series CPU Unit Hardware User’s Manual (Cat. No. W500)*, *NX-series CPU Unit Hardware User's Manual (Cat. No. W535)*, *NX-series NX502 CPU Unit Hardware User’s Manual (Cat. No. W629)*, *NX-series NX102 CPU Unit Hardware User's Manual (Cat. No. W593)*, or *NX-series NX1P2 CPU Unit Hardware User's Manual (Cat. No. W578)*.

Precautions when Installing Communications Cables

- Check the following items on the communications cables that are used in the network.
  - Are there any breaks?
  - Are there any shorts?
  - Are there any connector problems?
- When you connect the cable to the communications connectors on devices, firmly insert the communications cable connector until it locks in place.
- Do not lay the communications cables together with high-voltage lines.
- Do not lay the communications cable near devices that generate noise.
- Do not lay the communications cables in locations subject to high temperatures or high humidity.
- Do not lay the communications cables in locations subject to excessive dirt and dust or to oil mist or other contaminants.
- There are limitations on the bending radius of communications cables. Check the specifications of the communications cable for the bending radius.

4-2-2 Installing EtherCAT Communications Cables

Ethernet communications cables and connectors are used to connect the built-in EtherCAT port with EtherCAT slaves.

Use a straight, shielded twisted-pair cable (double shielding with aluminum tape and braiding) of Ethernet category 5 (100BASE-TX) or higher.

Cable with Connectors

● **Size and Conductor Pairs: AWG26 × 4 Pairs**

Product name	Manufacturer	Cable length [m]	Model	Contact information
Cable with Connectors on Both Ends (RJ45/RJ45) **Standard RJ45 plugs**[*1] **Cable Sheath material: LSZH**[*2] **Cable color: Yellow**[*3] 	OMRON Corporation	0.3	XS6W-6LSZH8SS30CM-Y	OMRON Customer Service Center
		0.5	XS6W-6LSZH8SS50CM-Y	
		1	XS6W-6LSZH8SS100CM-Y	
		2	XS6W-6LSZH8SS200CM-Y	
		3	XS6W-6LSZH8SS300CM-Y	
		5	XS6W-6LSZH8SS500CM-Y	

*1. Cables are available in the following lengths: 0.2, 0.3, 0.5, 1, 1.5, 2, 3, 5, 7.5, 10, 15, and 20 m. Refer to the *Industrial Ethernet Connectors Catalog* (Cat. No. G019) for details.

*2. This is the Low Smoke Zero Halogen cable for in-cabinet use. Although the LSZH cable is single shielded, its communications and noise characteristics meet the standards. PUR cables for out-of-cabinet use are also available.

*3. Cables colors are available in blue, yellow, or green.

● Size and Conductor Pairs: AWG 22 × 2 Pairs

Product name	Manufacturer	Cable length [m]	Model	Contact information
Cable with Connectors on Both Ends (RJ45/RJ45) **Rugged RJ45 plug**[*1] **Cable color: Light blue** 	OMRON Corporation	0.3	XS5W-T421-AMD-K	OMRON Customer Service Center
		0.5	XS5W-T421-BMD-K	
		1	XS5W-T421-CMD-K	
		2	XS5W-T421-DMD-K	
		5	XS5W-T421-GMD-K	
		10	XS5W-T421-JMD-K	
Cable with Connectors on Both Ends (M12 Straight/M12 Straight) **Shield Strengthening cable**[*2] **M12/Smartclick connectors** **Cable color: Black** 	OMRON Corporation	0.5	XS5W-T421-BM2-SS	
		1	XS5W-T421-CM2-SS	
		2	XS5W-T421-DM2-SS	
		3	XS5W-T421-EM2-SS	
		5	XS5W-T421-GM2-SS	
		10	XS5W-T421-JM2-SS	
Cable with Connectors on Both Ends (M12 Straight/RJ45) **Shield Strengthening cable**[*2] **M12/Smartclick connector and rugged RJ45 plug** **Cable color: Black** 	OMRON Corporation	0.5	XS5W-T421-BMC-SS	
		1	XS5W-T421-CMC-SS	
		2	XS5W-T421-DMC-SS	
		3	XS5W-T421-EMC-SS	
		5	XS5W-T421-GMC-SS	
		10	XS5W-T421-JMC-SS	
Cable with Connectors on Both Ends (RJ45/RJ45) **Rugged standard plugs**[*3] **Cable color: Yellow** 	3M Japan Limited	0.25	3RHS4-1100-0.25M	3M Japan Limited
		0.5	3RHS4-1100-0.5M	
		1	3RHS4-1100-1M	
		2	3RHS4-1100-2M	
		5	3RHS4-1100-5M	
		10	3RHS4-1100-10M	

[*1] Cables are available in the following lengths: 0.3, 0.5, 1, 2, 3, 5, 10, and 15 m. Refer to the *Industrial Ethernet Connectors Catalog* (Cat. No. G019) for details.

[*2] For details, contact your OMRON representative.

[*3] Cables are available from 0.25 m to 100 m. Ask the manufacturer for details on the models that are not described in the table.

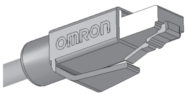
Cables and Connectors

● Size and Conductor Pairs: AWG 24 × 4 Pairs

Part name	Manufacturer	Model	Contact information
Cables	Hitachi Metals, Ltd.	NETSTAR-C5E SAB 0.5 × 4P*	Planning Department, Kanetsu Co., Ltd.
	Kuramo Electric Co., Ltd.	KETH-SB*	Kuramo Electric Co., Ltd.
RJ45 Connectors	Panduit Corporation	MPS588*	Panduit Corporation US Headquarters

* We recommend that you use combinations of the above Cables and Connectors.

● Size and Conductor Pairs: AWG 22 × 2 Pairs

Part name	Manufacturer	Model	Contact information
Cables	Kuramo Electric Co., Ltd.	KETH-PSB-OMR*	Kuramo Electric Co., Ltd.
	JMACS Japan Co., Ltd.	PNET/B*	JMACS Japan Co., Ltd.
RJ45 Assembly Connectors 	OMRON Corporation	XS6G-T421-1*	OMRON Customer Service Center

* We recommend that you use combinations of the above Cables and Connectors.

Part name	Manufacturer	Model	Contact information
Cables	3M Japan Limited	79100-IE4P-F1-YE*	3M Japan Limited
RJ45 Assembly Connectors		3R104-1110-000AM*	

* We recommend that you use combinations of the above Cables and Connectors.

Precautions for Correct Use

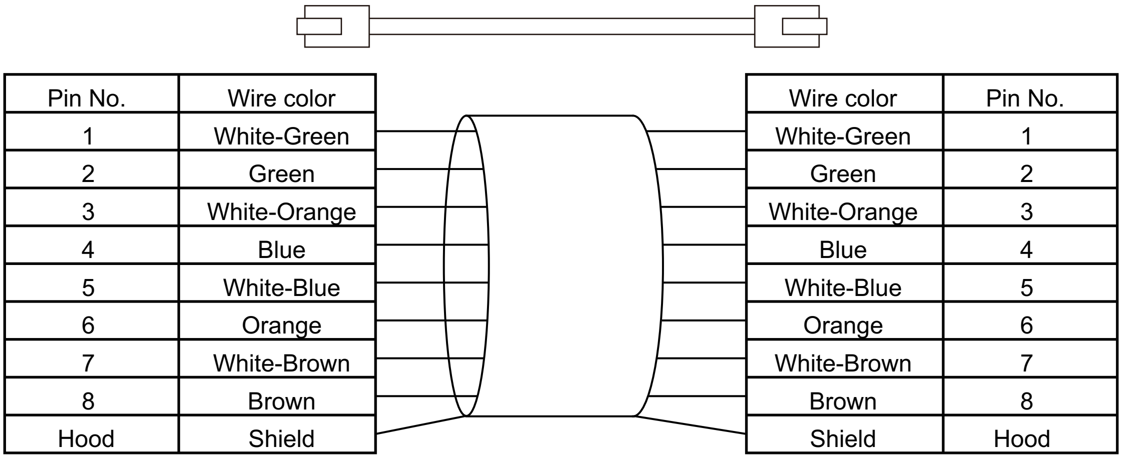
- The maximum length between nodes is 100 m. However, some cables are specified for less than 100 m. Generally speaking, if the conductors are twisted wire rather than solid wire, transmission performance will be lower, and reliable communications may not be possible at 100 m. Confirm details with the cable manufacturer.
- When selecting a connector, confirm that it is applicable to the cable that will be used. Confirm the following items: Conductor size, conductor type (solid wire or twisted wire), number of twisted pairs (2 or 4), outer diameter, etc.

Additional Information

If an Ethernet cable of category 5 or higher is used, communications will be possible even if the cable is not shielded. However, we recommend a cable with double, aluminum tape and braided shielding to ensure sufficient noise immunity.

Attaching the Connectors to the Cable and Pin Assignments

Use straight wiring to attach the connectors to the communications cable.



Pin No.	Wire color	Wire color	Pin No.
1	White-Green	White-Green	1
2	Green	Green	2
3	White-Orange	White-Orange	3
4	Blue	Blue	4
5	White-Blue	White-Blue	5
6	Orange	Orange	6
7	White-Brown	White-Brown	7
8	Brown	Brown	8
Hood	Shield	Shield	Hood

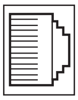
Note 1. Connect the cable shield to the connector hood at both ends of the cable.

Note 2. There are two connection methods for Ethernet: T568A and T568B. The T568A connection method is shown above, but the T568B connection method can also be used.

● Connector Specifications

Specification	Description
Electrical characteristics	Conforms to IEEE 802.3 standards.
Connector structure	RJ45 8-pin modular connector (Conforms to ISO 8877.)

● Pin Assignments

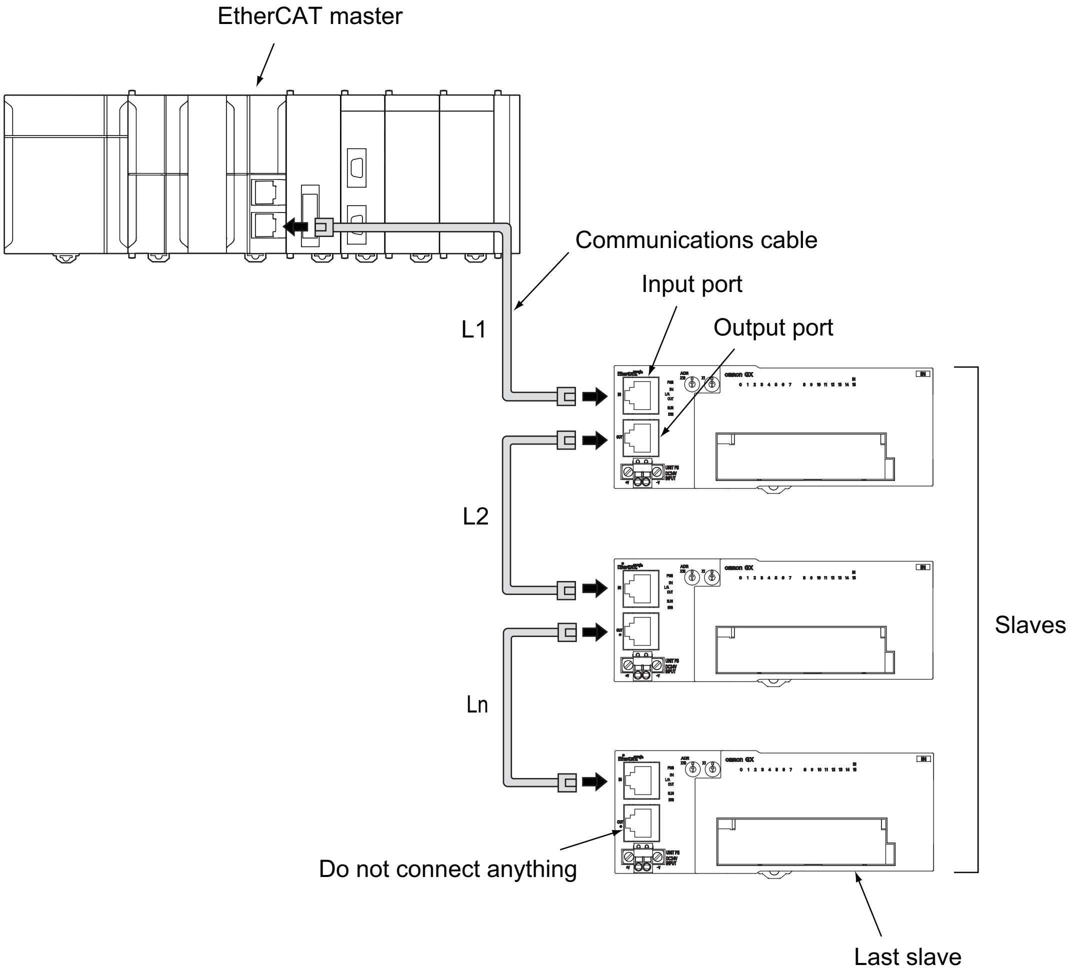
	Pin No.	Signal name	Abbreviation	Signal direction
	1	Transmission data +	TD+	Output
	2	Transmission data −	TD−	Output
	3	Reception data +	RD+	Input
	4	Not used.	---	---
	5	Not used.	---	---
	6	Reception data −	RD-	Input
	7	Not used.	---	---
	8	Not used.	---	---
	Hood	Frame ground	FG	---

4-2-3 Connecting Communications Cables

Cable connections can be made freely in EtherCAT networks.

Connect the communications cable from the EtherCAT master to the input port on the first slave, and then connect the communications cable to the next slave to the output port on the first slave.

Do not connect anything to the output port of the slave at the end of the network.



The cable between the EtherCAT master and a slave and between any two slaves (L1, L2 ... Ln) must be 100 m or less.

Firmly connect the communications cable connector until it clicks into place.

Precautions for Correct Use

Always perform one of the following operations before you disconnect the communications cable to a EtherCAT slave during operation.

- Disconnect the EtherCAT slave from the network.
- Disable the EtherCAT slaves and all slaves connected after it.

Refer to *9-4 Replacing Slaves during Communications* on page 9-26 for information on replacing slaves during operation.

Additional Information

Make sure the cable between each pair of devices connects an output port to an input port. Normal communications are not possible if an output port is connected to another output port or an input port is connected to another input port.

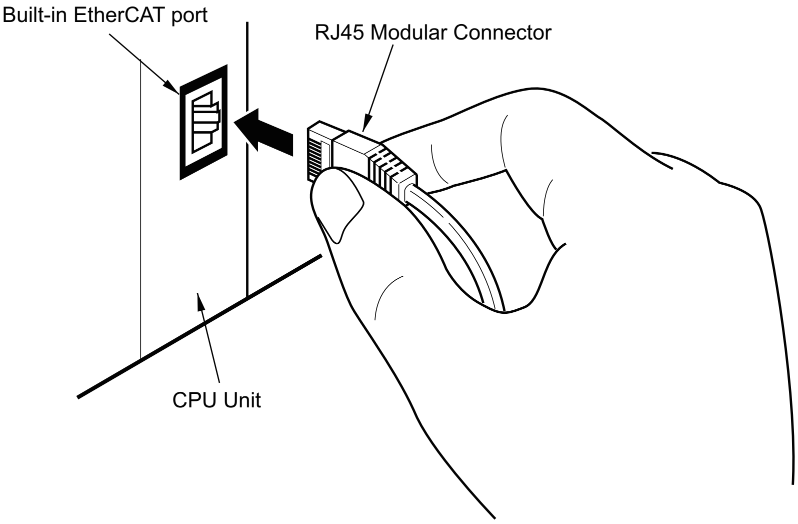
4-2-4 Cable Connection Procedure

Precautions for Correct Use

- Turn OFF the Controller's power supply before connecting or disconnecting Ethernet communications cable.
- Allow extra space for the bending radius of the communications cable.
  The required space depends on the communications cable, connector, and CPU Unit that are used.
  Refer to the *NJ-series CPU Unit Hardware User’s Manual (Cat. No. W500)* for details on the NJ-series CPU Unit.
  Refer to the *NX-series CPU Unit Hardware User's Manual (Cat. No. W535)* for details on the NX701 CPU Unit.
  Refer to the *NX-series NX502 CPU Unit Hardware User’s Manual (Cat. No. W629)* for details on the NX502 CPU Unit.
  Refer to the *NX-series NX102 CPU Unit Hardware User's Manual (Cat. No. W593)* for details on the NX102 CPU Unit.
  Refer to the *NX-series NX1P2 CPU Unit Hardware User's Manual (Cat. No. W578)* for details on the NX1P2 CPU Unit.

1 Lay the Ethernet communications cable.

2 Connect the Ethernet communications cable to the built-in EtherCAT port on the NJ/NX-series CPU Unit. Firmly insert the connector until it locks into place.



5

Setting Up EtherCAT Communications with the Sysmac Studio

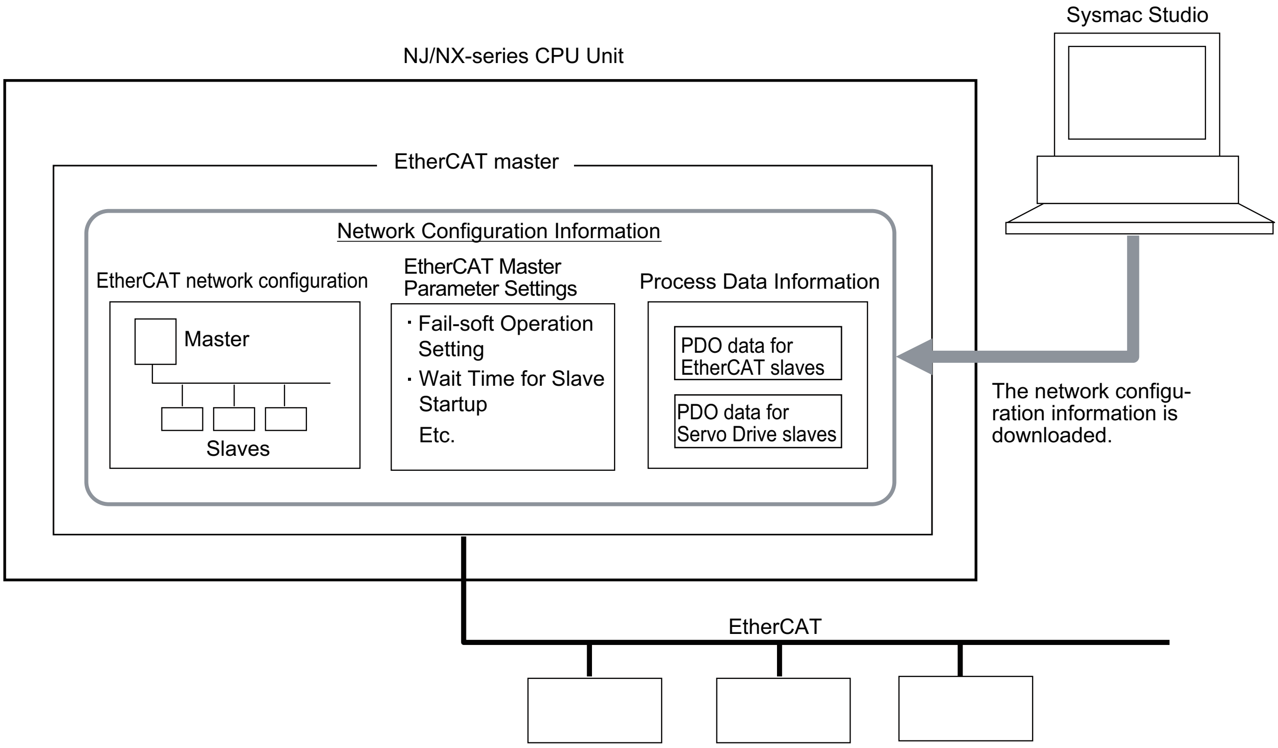
This section describes how to set the network configuration information and how to check EtherCAT communications from the Sysmac Studio.

5-1 Overview of Network Configuration Information

To execute EtherCAT communications, you must create the following three types of data with the Sysmac Studio and download the data to the CPU Unit. When this data is set, the network configuration information is automatically created.
When you download the network configuration information to the CPU Unit, the EtherCAT master initializes slaves and performs process data communications based on the network configuration information.
This section describes the procedures to set the network configuration information and how to check EtherCAT communications.

Network Configuration Information

Network configuration information	• EtherCAT network configuration
	• Process data information This is allocation information for PDO data in the slaves. It is also called PDO mapping data.
	• EtherCAT master and slave parameter settings (Examples: Fail-soft Operation Setting and Wait Time for Slave Startup)


Sysmac Studio
NJ/NX-series CPU Unit
EtherCAT master
Network Configuration Information
EtherCAT network configuration
Master
Slaves
EtherCAT Master Parameter Settings
· Fail-soft Operation Setting
· Wait Time for Slave Startup
Etc.
Process Data Information
PDO data for EtherCAT slaves
PDO data for Servo Drive slaves
The network configuration information is downloaded.
EtherCAT

5-2 Creating the EtherCAT Network Configuration

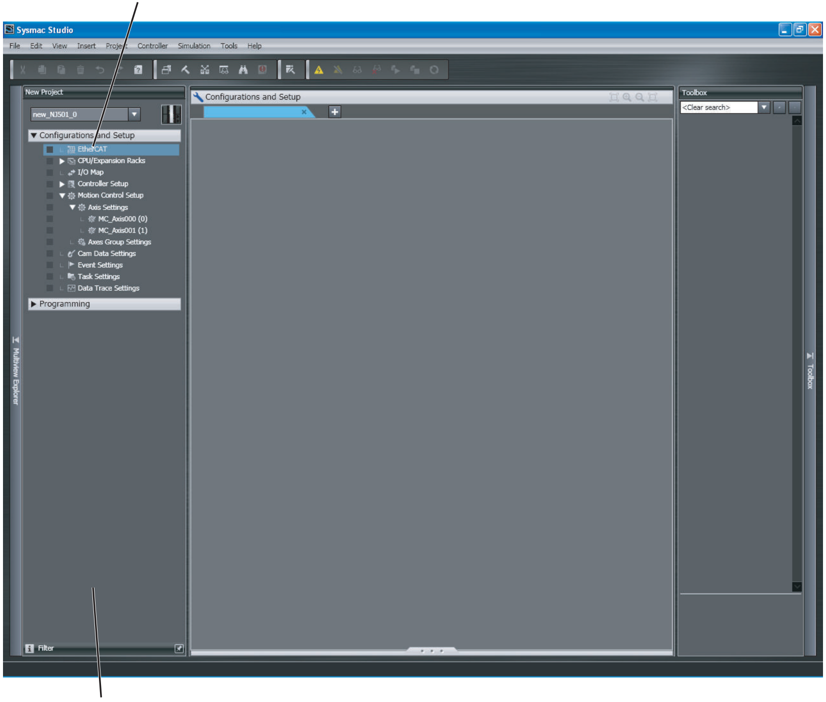
This section describes how to create the EtherCAT network configuration.

5-2-1 Procedure to Create the EtherCAT Network Configuration

1. Start the Sysmac Studio offline.

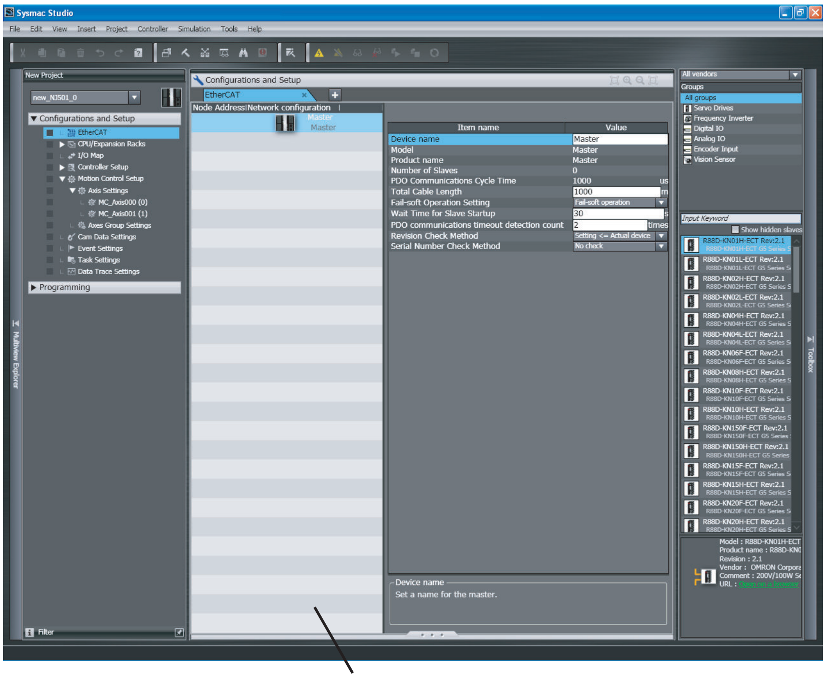
2. Double-click **EtherCAT** under **Configurations and Setup** on the Multiview Explorer. Or, right-click **EtherCAT** under **Configurations and Setup** and select **Edit**.

Double-click **EtherCAT** or rightclick **EtherCAT** and select **Edit**.



Multi-view Explorer

EtherCAT Master will be displayed in the EtherCAT Tab Page.



Item name	Value
Device name	Master
Model	Master
Product name	Master
Number of Slaves	0
PDO Communications Cycle Time	1000 us
Total Cable Length	1000 m
Fail-soft Operation Setting	Fail-soft operation
Wait Time for Slave Startup	30 s
PDO communications timeout detection count	2 times
Revision Check Method	Setting <= Actual device
Serial Number Check Method	No check

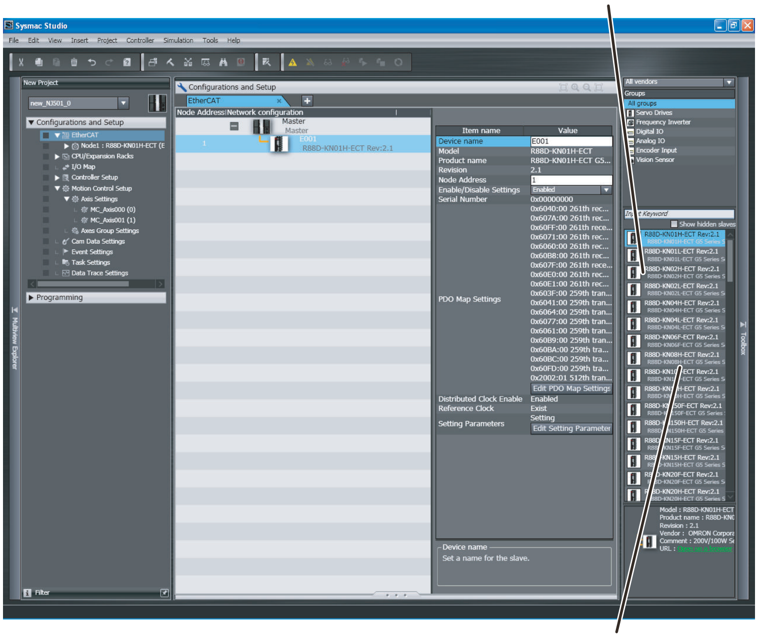
Device name
Set a name for the master.

EtherCAT Tab Page

3 Select a slave in the Toolbox, drag it to the EtherCAT Tab Page, and drop it to the master.

The slave will be added under the master.

Toolbox



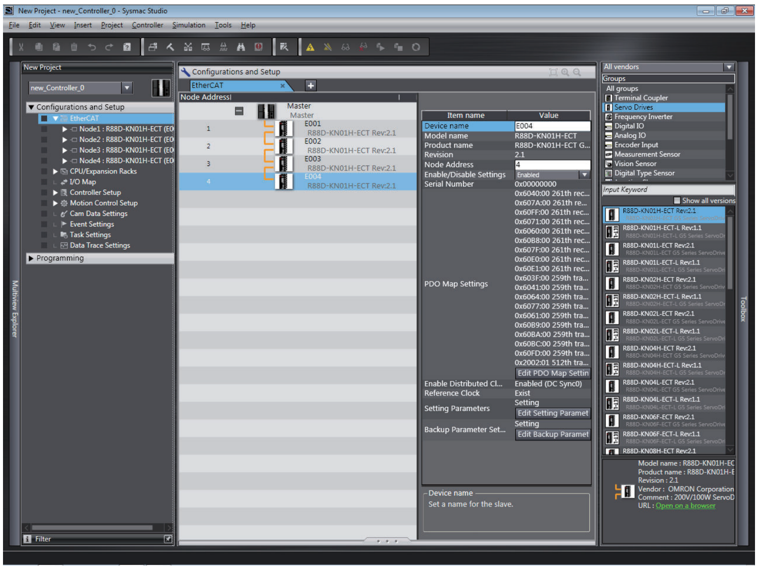
Item name	Value
Device name	E001
Model	R88D-KN01H-ECT
Product name	R88D-KN01H-ECT G5...
Revision	2.1
Node Address	1
Enable/Disable Settings	Enabled
Serial Number	0x00000000
PDO Map Settings	0x6040:00 261th rec... 0x607A:00 261th rec... 0x60FF:00 261th rece... 0x6071:00 261th rec... 0x6060:00 261th rec... 0x60B8:00 261th rec... 0x607F:00 261th rece... 0x60E0:00 261th rec... 0x60E1:00 261th rec... 0x603F:00 259th tran... 0x6041:00 259th tran... 0x6064:00 259th tran... 0x6077:00 259th tran... 0x6061:00 259th tran... 0x60B9:00 259th tran... 0x60BA:00 259th tra... 0x60BC:00 259th tra... 0x60FD:00 259th tra... 0x2002:01 512th tran... Edit PDO Map Settings
Distributed Clock Enable	Enabled
Reference Clock	Exist
Setting Parameters	Setting Edit Setting Parameter

Device name
Set a name for the slave.

Drag the slaves.

4 Select another slave in the Toolbox, drag it to the EtherCAT Tab Page, and drop it under the slave to which you want to connect it to.

The slave will be added under the slave.



Item name	Value
Device name	E004
Model name	R88D-KN01H-ECT
Product name	R88D-KN01H-ECT G...
Revision	2.1
Node Address	4
Enable/Disable Settings	Enabled
Serial Number	0x00000000
PDO Map Settings	0x6040:00 261th rec... 0x607A:00 261th re... 0x60FF:00 261th rec... 0x6071:00 261th rec... 0x6060:00 261th rec... 0x60B8:00 261th rec... 0x607F:00 261th rec... 0x60E0:00 261th rec... 0x60E1:00 261th rec... 0x603F:00 259th tra... 0x6041:00 259th tra... 0x6064:00 259th tra... 0x6077:00 259th tra... 0x6061:00 259th tra... 0x60B9:00 259th tra... 0x60BA:00 259th tra... 0x60BC:00 259th tra... 0x60FD:00 259th tra... 0x2002:01 512th tra... Edit PDO Map Settin
Enable Distributed Cl...	Enabled (DC Sync0)
Reference Clock	Exist
Setting Parameters	Setting Edit Setting Paramet
Backup Parameter Set...	Setting Edit Backup Paramet

Device name
Set a name for the slave.

Model name : R88D-KN01H-EC
Product name : R88D-KN01H-E
Revision : 2.1
Vendor : OMRON Corporation
Comment : 200V/100W ServoD
URL : Open on a browser

- For detailed procedures, such as those for deleting slaves, or copying and pasting slaves, refer to the *Sysmac Studio Version 1 Operation Manual (Cat. No. W504)*.
- Refer to the *NX-series EtherCAT Coupler Units User’s Manual (Cat. No. W519)* and the *Sysmac Studio Version 1 Operation Manual (Cat. No. W504)* for information on creating the EtherCAT Slave Terminal configuration.
- Refer to *MDP Settings for EtherCAT Network Slaves* in the *Sysmac Studio Version 1 Operation Manual (Cat. No. W504)* for information on MDP slaves and editing the configurations of the modules that you can connect to the MDP slaves.

Additional Information

You can go online and read the actual network configuration from the Sysmac Studio to display it and to use it as the network configuration in the project.
For details, refer to *5-7 Automatically Creating the Network Configuration* on page 5-40.

5-2-2 Cable Redundancy Setting

This section describes the settings for configuring a ring topology.
To configure a ring topology, you must enable the cable redundancy function.

Precautions for Correct Use

Refer to *4-1-3 Ring Topology* on page 4-4 for precautions on the specifications and configurations of the ring topology.

The following terms are used to define the states of a ring topology for which the cable redundancy function is enabled.

Term	State of ring topology
Cable redundancy status	A state of the ring topology, in which there is no broken cable and all slaves participate in it, except for disabled slaves.
Ring disconnection status	A state of the ring topology, in which there is only one point of disconnection in the communications cables.

The operation of the cable redundancy function and its setting procedure are described below.

Operation of the Cable Redundancy Function

The operation of the cable redundancy function is described below.

- If there is only one point of disconnection in the communications cables in a ring topology and its state changes from the cable redundancy status to the ring disconnection status, a *Ring Disconnection Detected* (84390000 hex) event occurs and the state of the communications cables and the point of disconnection are provided in the following system-defined variables.
  a) *_EC_RingBreaking* (Ring Disconnection)
  b) *_EC_RingBreakNodeAdr* (Slave Node Address Before Ring Disconnection)

Additional Information

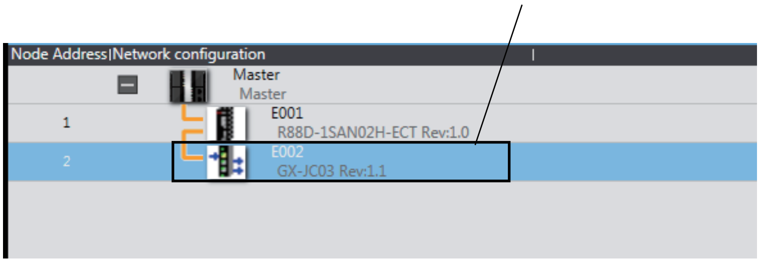
When the observation event level is set for the *Ring Disconnection Detected* event, even the event occurs, the event is not displayed in current errors. To detect a broken cable in a ring topology, monitor the above system-defined variables with the user program.
If the event level is set to the minor fault and the event occurs, it is displayed on the current errors in the Sysmac Studio.

- When you repair the point of disconnection, for example, by replacing the cables in the ring disconnection status, the topology state automatically changes to cable redundancy status.

Procedure to Set the Cable Redundancy Function

Use the Sysmac Studio and enable the cable redundancy function for a part of the EtherCAT network configuration where you want to set a ring topology.
The procedure to set the GX-JC03 Junction Slave as the originating slave of the ring is described below.

1 In the EtherCAT Tab Page, drop the Junction Slave that becomes the originating slave of the ring, to the master or slave.



Originating slave of the ring

Node Address|Network configuration

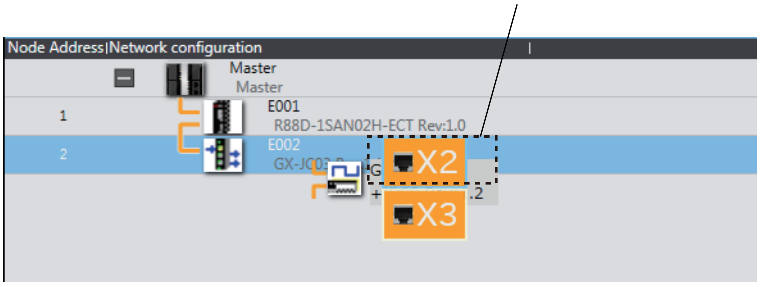
Master
Master

1 E001
R88D-1SAN02H-ECT Rev:1.0

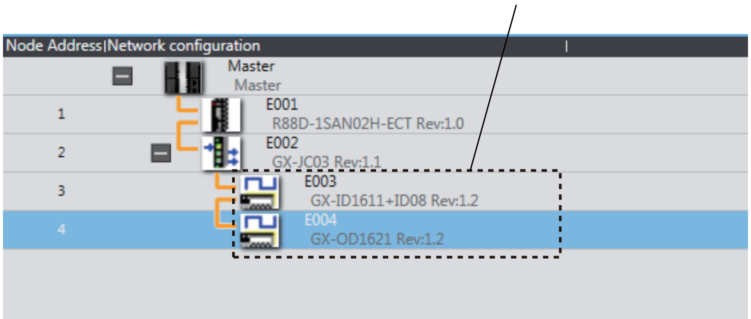
2 E002
GX-JC03 Rev:1.1

2 Drop the slave to the start port of the ring X2 that becomes the start point of the ring topology. The slave is connected to the start port of the ring. If you add more slaves, drop a slave to the connected slave and then connect the slave to the start port of the ring.

Start port of the ring.



Connect the slave to the start port of the ring.

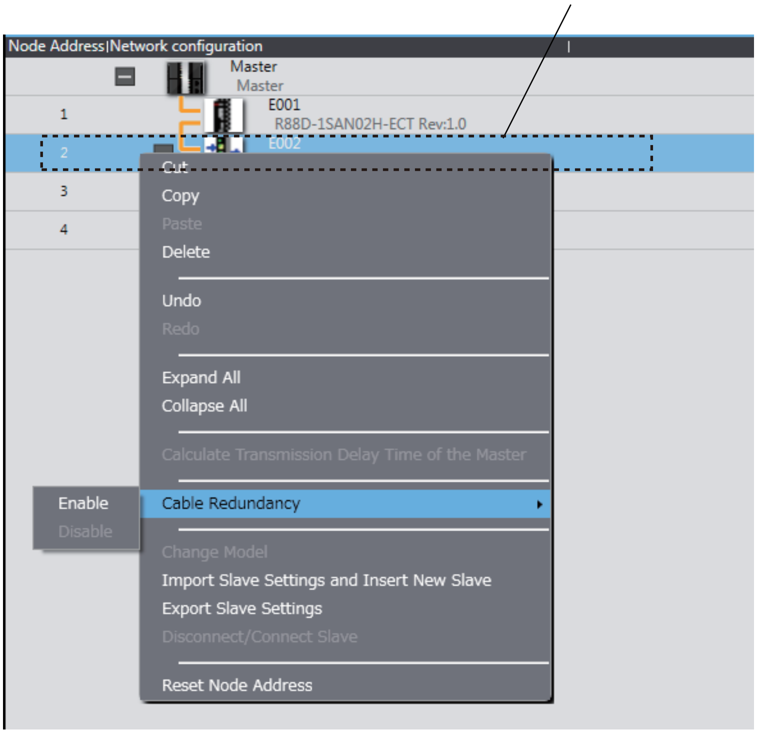


Precautions for Correct Use

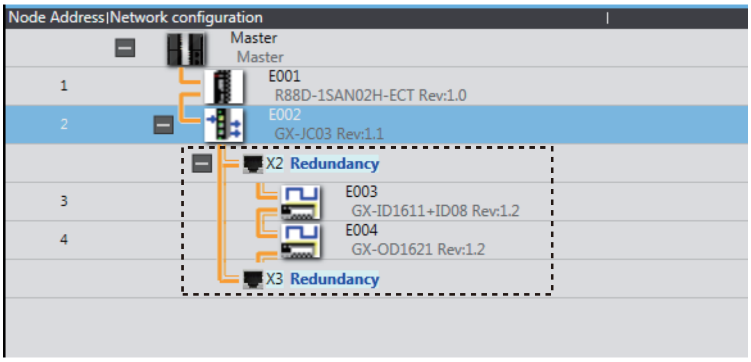
Do not connect the slave to the end port of the ring X3 that becomes the end point of the ring topology. You cannot enable the cable redundancy function.

3 In the EtherCAT Tab Page, right-click the originating slave of the ring and select **Cable Redundancy** to **Enable**.

Originating slave of the ring



As shown below, "Redundancy" is indicated and double lines are displayed on the location where the cable redundancy is enabled.



Precautions for Using Cable Redundancy Function

This section provides precautions for using the cable redundancy function.

Precautions for Correct Use

When the cable redundancy function is used, make the following settings properly.

- Set the **PDO communications timeout detection count** setting in the EtherCAT master to at least 3.
  An error may occur if the communications cable is disconnected or the power supply to the EtherCAT slave is turned OFF while **PDO communications timeout detection count** is set to 2 (default) or less. Refer to *5-4 EtherCAT Master Parameter Settings* on page 5-14 for the setting procedure.
- Set the set value of the Communications Error Setting for all of the OMRON synced slaves in the EtherCAT network to the following value.
  A Communications Synchronization Error will occur and communications may not continue if the communications cable is disconnected or the power supply to the EtherCAT slaves is turned OFF while the set value is set to a value other than the following value.
  Refer to *5-5 EtherCAT Slave Parameter Settings* on page 5-20 or relevant manuals for each slave for details on the setting procedure.
  For project unit version 1.42 or later: *2* or higher
  For project unit version earlier than 1.42: *1* or higher

5-3 Setting EtherCAT Slave Variables and Axes

Device variables are used to read and write process data for EtherCAT slaves. Axis variables are used to manipulate slaves to which axes are assigned from the Motion Control Function Module. This section describes how to register device variables and set the axes.

5-3-1 Registering Device Variables for All EtherCAT Slaves

You use the I/O Map in the Sysmac Studio to assign device variables to the I/O ports. The device variables that you create are registered in the global variable table.
Use one of the following three methods.

- Selecting variables that were previously registered in a variable table
- Inputting new device variable names.
- Automatically creating device variable names

Selecting Variables That Were Previously Registered in a Variable Table

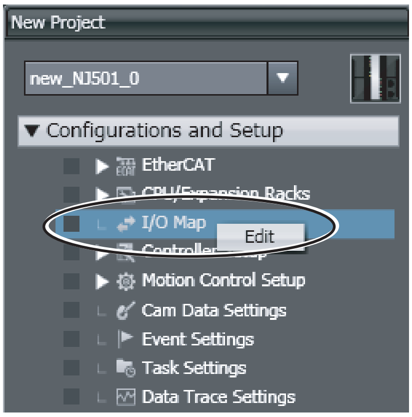
You can select variables that are already registered on the menu on the I/O Map Tab Page.

For example, this method can be used to register device variables in the following cases.

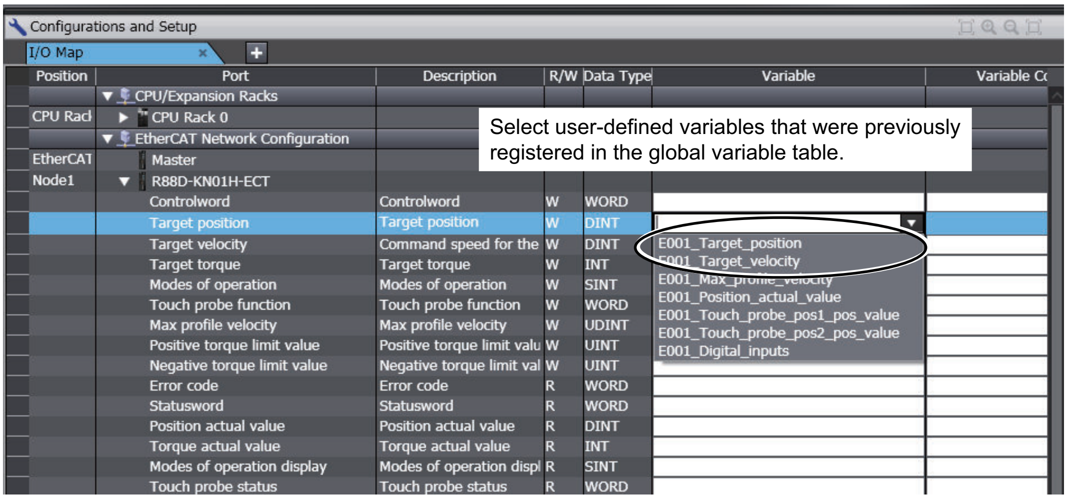
- To write the program before the slave configuration information is created.
- To reuse programs from another project.

Use the following procedure.

1. Register the variables in advance in the global variable table or the local variable table of one of the POUs.
2. Program using those variables.
3. Create the slave configuration information.
4. Double-click **I/O Map** under **Configurations and Setup** on the Multiview Explorer. Or, right-click **I/O Map** under **Configurations and Setup** and select **Edit**.



5 Select the variables from the pull-down list in the I/O Map Tab Page to assign them to I/O ports.


Configurations and Setup
I/O Map
E001_Target_position
E001_Target_velocity
E001_Max_profile_velocity
E001_Position_actual_value
E001_Touch_probe_pos1_pos_value
E001_Touch_probe_pos2_pos_value
E001_Digital_inputs

Position	Port	Description	R/W	Data Type	Variable	Variable Cc
	CPU/Expansion Racks					
CPU Racl	CPU Rack 0					
	EtherCAT Network Configuration					
EtherCAT	Master					
Node1	R88D-KN01H-ECT					
	Controlword	Controlword	W	WORD		
	Target position	Target position	W	DINT		
	Target velocity	Command speed for the	W	DINT		
	Target torque	Target torque	W	INT		
	Modes of operation	Modes of operation	W	SINT		
	Touch probe function	Touch probe function	W	WORD		
	Max profile velocity	Max profile velocity	W	UDINT		
	Positive torque limit value	Positive torque limit valu	W	UINT		
	Negative torque limit value	Negative torque limit val	W	UINT		
	Error code	Error code	R	WORD		
	Statusword	Statusword	R	WORD		
	Position actual value	Position actual value	R	DINT		
	Torque actual value	Torque actual value	R	INT		
	Modes of operation display	Modes of operation displ	R	SINT		
	Touch probe status	Touch probe status	R	WORD		

Select user-defined variables that were previously registered in the global variable table.

Inputting New Device Variable Names

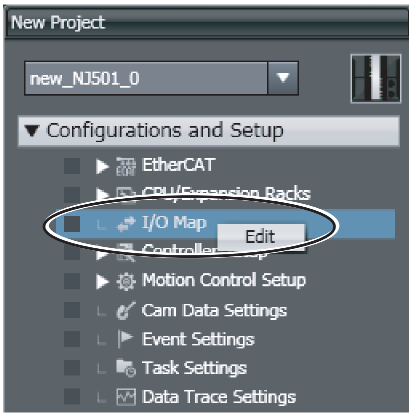
You can input the required device variable names.

For example, this method can be used to register device variables in the following cases.

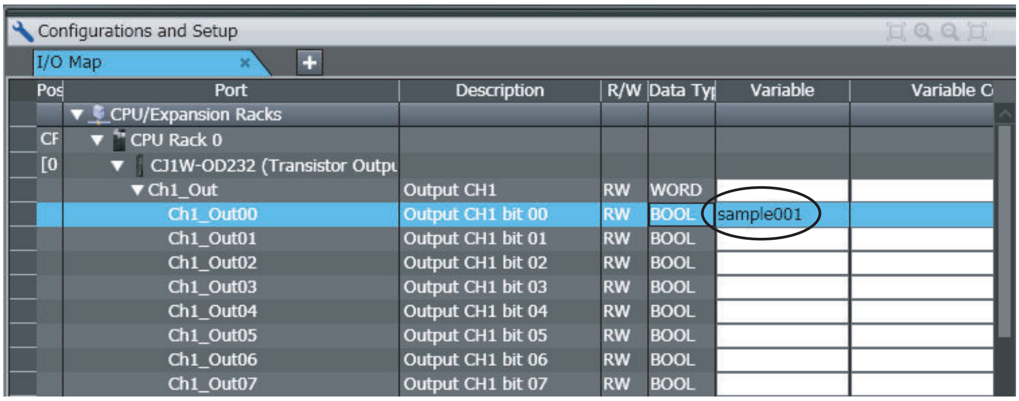
- To give suitable names to device variables that are assigned to I/O for I/O slaves.

Use the following procedure.

1 Double-click **I/O Map** under **Configurations and Setup** on the Multiview Explorer. Or, right-click **I/O Map** under **Configurations and Setup** and select **Edit**.



2 Select the I/O port on the I/O Map Tag Page and enter the variable name in the **Variable** Column.

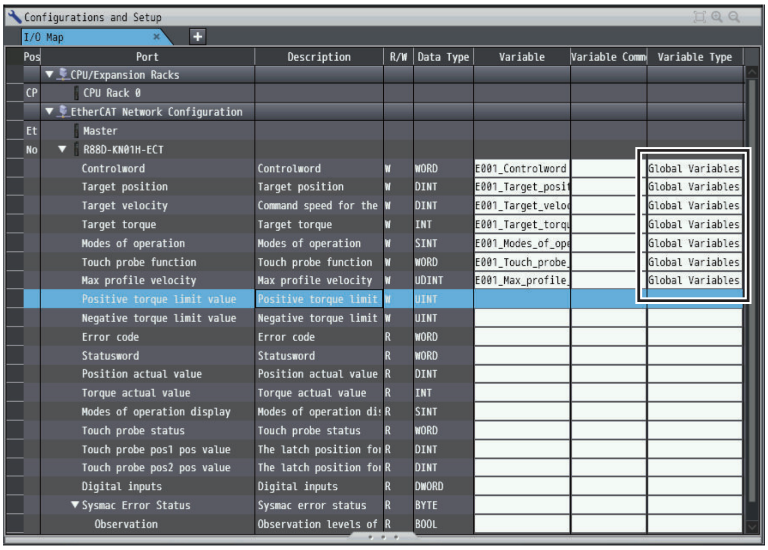

Configurations and Setup

I/O Map

Pos	Port	Description	R/W	Data Typ	Variable	Variable C
	▼ CPU/Expansion Racks					
CF	▼ CPU Rack 0					
[0	▼ CJ1W-OD232 (Transistor Outpu					
	▼ Ch1_Out	Output CH1	RW	WORD		
	Ch1_Out00	Output CH1 bit 00	RW	BOOL	sample001	
	Ch1_Out01	Output CH1 bit 01	RW	BOOL		
	Ch1_Out02	Output CH1 bit 02	RW	BOOL		
	Ch1_Out03	Output CH1 bit 03	RW	BOOL		
	Ch1_Out04	Output CH1 bit 04	RW	BOOL		
	Ch1_Out05	Output CH1 bit 05	RW	BOOL		
	Ch1_Out06	Output CH1 bit 06	RW	BOOL		
	Ch1_Out07	Output CH1 bit 07	RW	BOOL		

Device variables are assigned to the I/O ports of the slaves.

3 To specify a variable table for the scope, specify the **Variable Type**.


Configurations and Setup

I/O Map

Pos	Port	Description	R/W	Data Type	Variable	Variable Comm	Variable Type
	▼ CPU/Expansion Racks						
CP	CPU Rack 0						
	▼ EtherCAT Network Configuration						
Et	Master						
No	▼ R88D-KN01H-ECT						
	Controlword	Controlword	W	WORD	E001_Controlword		Global Variables
	Target position	Target position	W	DINT	E001_Target_posit		Global Variables
	Target velocity	Command speed for the	W	DINT	E001_Target_veloc		Global Variables
	Target torque	Target torque	W	INT	E001_Target_torqu		Global Variables
	Modes of operation	Modes of operation	W	SINT	E001_Modes_of_ope		Global Variables
	Touch probe function	Touch probe function	W	WORD	E001_Touch_probe_		Global Variables
	Max profile velocity	Max profile velocity	W	UDINT	E001_Max_profile_		Global Variables
	Positive torque limit value	Positive torque limit	W	UINT			
	Negative torque limit value	Negative torque limit	W	UINT			
	Error code	Error code	R	WORD			
	Statusword	Statusword	R	WORD			
	Position actual value	Position actual value	R	DINT			
	Torque actual value	Torque actual value	R	INT			
	Modes of operation display	Modes of operation di:	R	SINT			
	Touch probe status	Touch probe status	R	WORD			
	Touch probe pos1 pos value	The latch position foı	R	DINT			
	Touch probe pos2 pos value	The latch position foı	R	DINT			
	Digital inputs	Digital inputs	R	DWORD			
	▼ Sysmac Error Status	Sysmac error status	R	BYTE			
	Observation	Observation levels of	R	BOOL			

Automatically Creating Device Variable Names

The device variable names can be created by combining the device name and the I/O port name. The device names are set in the slave parameters.

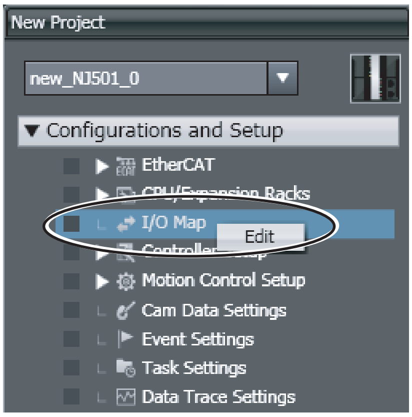
The default device names are E followed by a serial number that starts from 001.

For example, this method can be used to register device variables in the following cases.

- To eliminate the work that is involved in obtaining the device variable names.

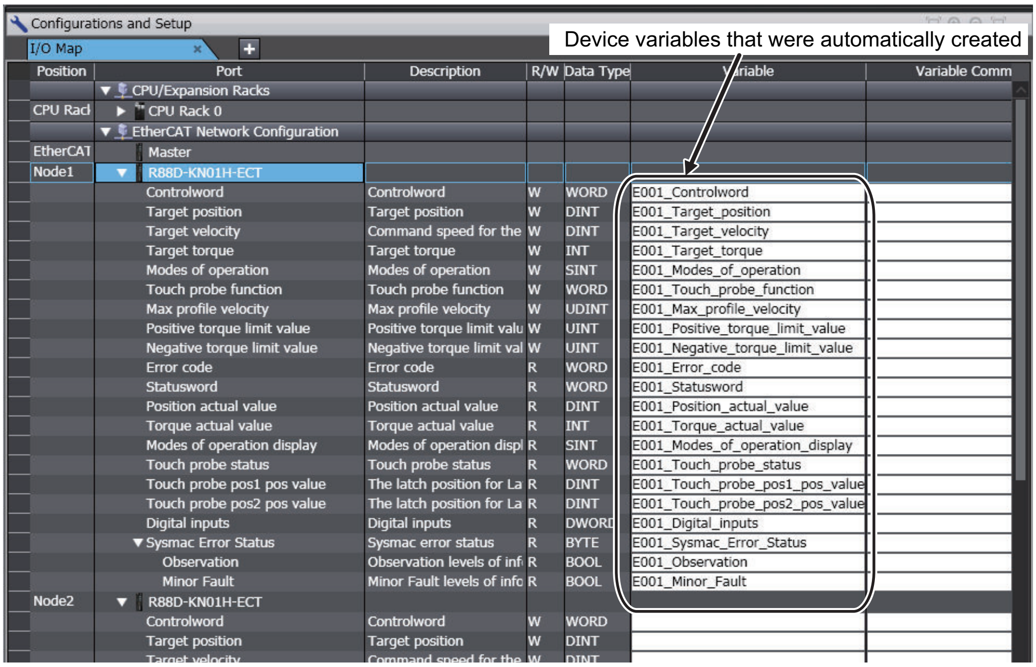
Use the following procedure.

1 Double-click **I/O Map** under **Configurations and Setup** on the Multiview Explorer. Or, right-click **I/O Map** under **Configurations and Setup** and select **Edit**.



The I/O Map Tab Page will be displayed.

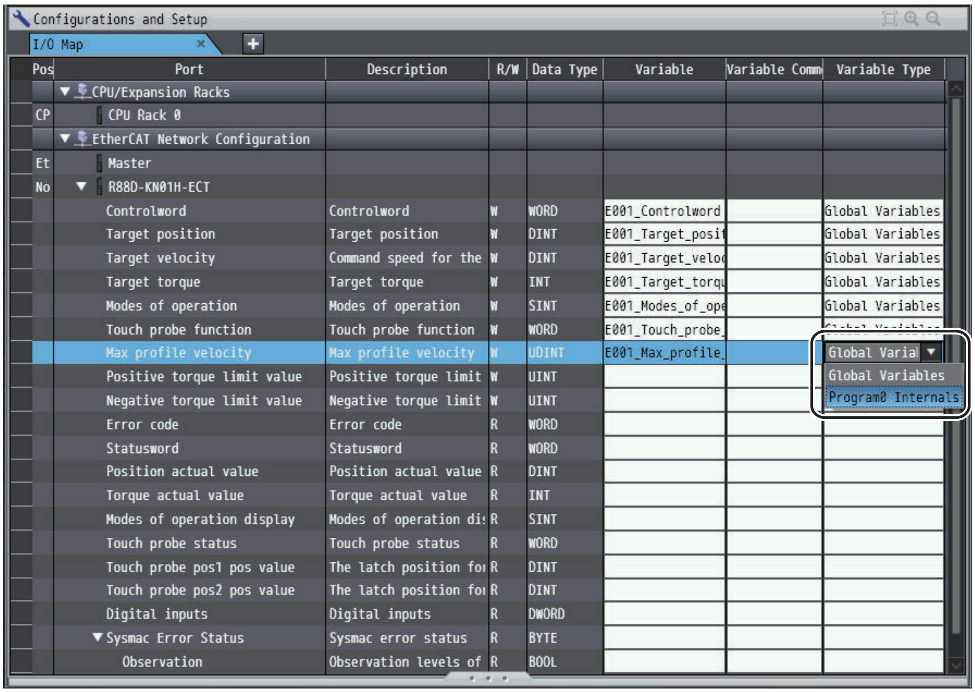
2 Select a slave or I/O ports on the I/O Map Tab Page, right-click, and select **Create Device Variable**.



Device variables that were automatically created

Position	Port	Description	R/W	Data Type	Variable	Variable Comm
	▼ CPU/Expansion Racks					
CPU Racl	▶ CPU Rack 0					
	▼ EtherCAT Network Configuration					
EtherCAT	Master					
Node1	▼ R88D-KN01H-ECT					
	Controlword	Controlword	W	WORD	E001_Controlword	
	Target position	Target position	W	DINT	E001_Target_position	
	Target velocity	Command speed for the	W	DINT	E001_Target_velocity	
	Target torque	Target torque	W	INT	E001_Target_torque	
	Modes of operation	Modes of operation	W	SINT	E001_Modes_of_operation	
	Touch probe function	Touch probe function	W	WORD	E001_Touch_probe_function	
	Max profile velocity	Max profile velocity	W	UDINT	E001_Max_profile_velocity	
	Positive torque limit value	Positive torque limit valu	W	UINT	E001_Positive_torque_limit_value	
	Negative torque limit value	Negative torque limit val	W	UINT	E001_Negative_torque_limit_value	
	Error code	Error code	R	WORD	E001_Error_code	
	Statusword	Statusword	R	WORD	E001_Statusword	
	Position actual value	Position actual value	R	DINT	E001_Position_actual_value	
	Torque actual value	Torque actual value	R	INT	E001_Torque_actual_value	
	Modes of operation display	Modes of operation displ	R	SINT	E001_Modes_of_operation_display	
	Touch probe status	Touch probe status	R	WORD	E001_Touch_probe_status	
	Touch probe pos1 pos value	The latch position for La	R	DINT	E001_Touch_probe_pos1_pos_value	
	Touch probe pos2 pos value	The latch position for La	R	DINT	E001_Touch_probe_pos2_pos_value	
	Digital inputs	Digital inputs	R	DWORD	E001_Digital_inputs	
	▼ Sysmac Error Status	Sysmac error status	R	BYTE	E001_Sysmac_Error_Status	
	Observation	Observation levels of inf	R	BOOL	E001_Observation	
	Minor Fault	Minor Fault levels of info	R	BOOL	E001_Minor_Fault	
Node2	▼ R88D-KN01H-ECT					
	Controlword	Controlword	W	WORD		
	Target position	Target position	W	DINT		
	Target velocity	Command speed for the	W	DINT		

3 If you specify a variable table for the scope, specify the **Variable Type** afterward.



Configurations and Setup

I/O Map

Pos	Port	Description	R/W	Data Type	Variable	Variable Comm	Variable Type
	▼ CPU/Expansion Racks						
CP	CPU Rack 0						
	▼ EtherCAT Network Configuration						
Et	Master						
No	▼ R88D-KN01H-ECT						
	Controlword	Controlword	W	WORD	E001_Controlword		Global Variables
	Target position	Target position	W	DINT	E001_Target_posit		Global Variables
	Target velocity	Command speed for the	W	DINT	E001_Target_veloc		Global Variables
	Target torque	Target torque	W	INT	E001_Target_torqu		Global Variables
	Modes of operation	Modes of operation	W	SINT	E001_Modes_of_ope		Global Variables
	Touch probe function	Touch probe function	W	WORD	E001_Touch_probe_		Global Variables
	Max profile velocity	Max profile velocity	W	UDINT	E001_Max_profile_		Global Varial ▼
	Positive torque limit value	Positive torque limit	W	UINT			
	Negative torque limit value	Negative torque limit	W	UINT			
	Error code	Error code	R	WORD			
	Statusword	Statusword	R	WORD			
	Position actual value	Position actual value	R	DINT			
	Torque actual value	Torque actual value	R	INT			
	Modes of operation display	Modes of operation di	R	SINT			
	Touch probe status	Touch probe status	R	WORD			
	Touch probe pos1 pos value	The latch position fo	R	DINT			
	Touch probe pos2 pos value	The latch position fo	R	DINT			
	Digital inputs	Digital inputs	R	DWORD			
	▼ Sysmac Error Status	Sysmac error status	R	BYTE			
	Observation	Observation levels of	R	BOOL			

Device variables are automatically assigned to the I/O ports of the slaves and registered in the variable table that is specified by the **Variable Type**.

Additional Information

- We recommend that you set the device names.
- To delete the mapping for the variable that is assigned to a port, delete the contents of the cell in the **Variable** Column or right-click the cell and select **Reset Assignment**.
  The mapping of the device variable is deleted. The variable is not deleted from the variable table.
  Refer to the *NJ/NX-series CPU Unit Software User's Manual (Cat. No. W501)* for details on devices variables.
- If you do not specify anything in the **Variable Type** Column when you register a variable or create device variables, the variables are automatically registered in the global variable table.

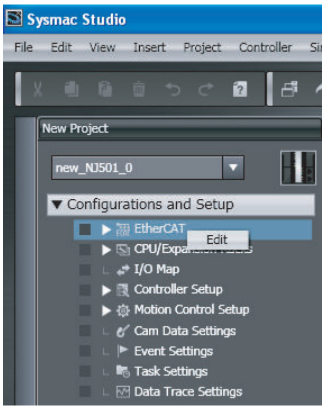
5-3-2 Axis Settings for Servo Drives and Encoder Input Slaves

To operate slaves that can be assigned to axes in the Motion Control Function Module, you have to make axes settings.

Refer to the *NJ/NX-series CPU Unit Motion Control User's Manual (Cat. No. W507)* for details on the items and procedures of the axes setting.

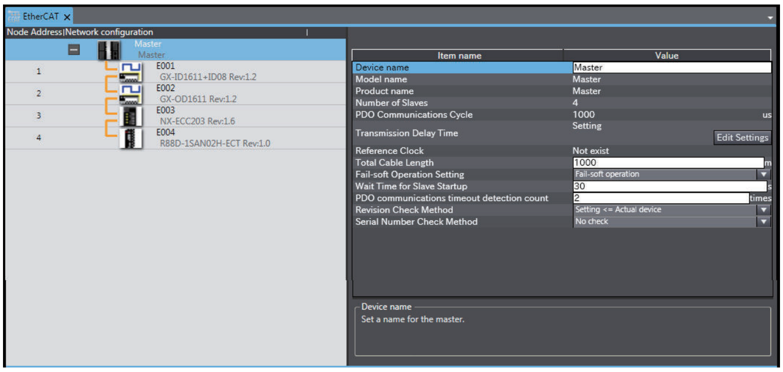
5-4 EtherCAT Master Parameter Settings

1 Double-click **EtherCAT** under **Configurations and Setup** on the Multiview Explorer. Or, right-click **EtherCAT** under **Configurations and Setup** and select **Edit**.



2 Select the EtherCAT master that is displayed in the Edit Pane.

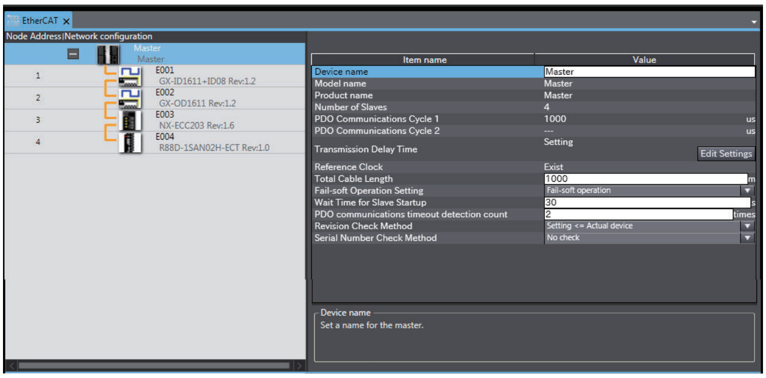
- For NJ-series CPU Units


EtherCAT
Node Address|Network configuration
Master
Master
1 E001 GX-ID1611+ID08 Rev:1.2
2 E002 GX-OD1611 Rev:1.2
3 E003 NX-ECC203 Rev:1.6
4 E004 R88D-1SAN02H-ECT Rev:1.0
Edit Settings

Item name	Value
Device name	Master
Model name	Master
Product name	Master
Number of Slaves	4
PDO Communications Cycle	1000 us
Transmission Delay Time	Setting
Reference Clock	Not exist
Total Cable Length	1000 m
Fail-soft Operation Setting	Fail-soft operation
Wait Time for Slave Startup	30 s
PDO communications timeout detection count	2 times
Revision Check Method	Setting <= Actual device
Serial Number Check Method	No check

Device name
Set a name for the master.

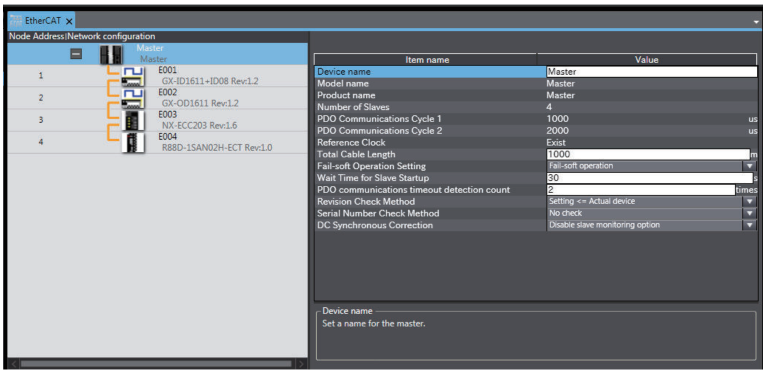
- For NX-series CPU Units


EtherCAT
Node Address|Network configuration
Master
Master
1 E001 GX-ID1611+ID08 Rev:1.2
2 E002 GX-OD1611 Rev:1.2
3 E003 NX-ECC203 Rev:1.6
4 E004 R88D-1SAN02H-ECT Rev:1.0
Edit Settings

Item name	Value
Device name	Master
Model name	Master
Product name	Master
Number of Slaves	4
PDO Communications Cycle 1	1000 us
PDO Communications Cycle 2	--- us
Transmission Delay Time	Setting
Reference Clock	Exist
Total Cable Length	1000 m
Fail-soft Operation Setting	Fail-soft operation
Wait Time for Slave Startup	30 s
PDO communications timeout detection count	2 times
Revision Check Method	Setting <= Actual device
Serial Number Check Method	No check

Device name
Set a name for the master.

The above screen is for NX102 CPU Units.



Node Address	Network configuration
	Master Master
1	E001 GX-ID1611+ID08 Rev:1.2
2	E002 GX-OD1611 Rev:1.2
3	E003 NX-ECC203 Rev:1.6
4	E004 R88D-1SAN02H-ECT Rev:1.0

Item name	Value	
Device name	Master	
Model name	Master	
Product name	Master	
Number of Slaves	4	
PDO Communications Cycle 1	1000	us
PDO Communications Cycle 2	2000	us
Reference Clock	Exist	
Total Cable Length	1000	m
Fail-soft Operation Setting	Fail-soft operation	
Wait Time for Slave Startup	30	s
PDO communications timeout detection count	2	times
Revision Check Method	Setting <= Actual device	
Serial Number Check Method	No check	
DC Synchronous Correction	Disable slave monitoring option	

Device name
Set a name for the master.

The above screen is for NX701 CPU Units.

The EtherCAT master settings are listed below.

Name	Editing	Remarks
Device name	OK	Displays the name of the EtherCAT master. Default setting: Master
Model name	Not allowed.	Always “Master.”
Product name	Not allowed.	Always “Master.”
Number of Slaves	Not allowed.	The number of slaves is automatically calculated based on the topology. Display range: 0 to the maximum number of slaves*1.
PDO Communications Cycle*2	Not allowed.	Displays the time of Process Data Communications Cycle. This is automatically input based on the task period of the primary periodic task. Display range: 500 to 4,000 μs Default setting: 1,000 μs
PDO Communications Cycle 1*3	Not allowed.	Displays the time of Process Data Communications Cycle 1. This is automatically input based on the task period of the primary periodic task. Display range: 125, 250 to 32,000 μs. Default setting:1,000 μs*4
PDO Communications Cycle 2*3	Not allowed.	Displays the time of Process Data Communications Cycle 2. This is automatically input based on the task period of the priority-5 periodic task.*5 Display range: 125, 250 to 100,000 μs. Default setting: 2,000 μs

Name	Editing	Remarks
Transmission Delay Time[*6]	OK	Sets the transmission delay time that is a parameter used to calculate the I/O refresh time. Selects the transmission delay time for each PDO communications cycle from the following values. • Estimated result from network configuration setting on Sysmac Studio • Calculation result from the measured value in the actual network configuration[*7] Default setting: Estimated result from network configuration setting on Sysmac Studio
Reference Clock	Not allowed.	Displays whether the master provides a reference clock.[*8] Exist: The master provides a reference clock. Not Exist: The master does not provides a reference clock.
Total Cable Length[*9]	OK	This is the total cable length, including the cable between the EtherCAT master and a slave and cables between slaves. Setting range: 1 to (100 × Maximum number of slaves)[*10] Default setting: 1,000 m
Fail-soft Operation Setting	OK	Selects **Stop** or **Fail-soft operation** for communications with all slaves on the network when the master detects a communications error (cable disconnection, slave malfunction, etc.). Default setting: Fail-soft operation For the operating specifications of the master and slaves with the Fail-soft Operation Setting when a communications error occurs, refer to *Errors in the EtherCAT Master Function Module* in the *NJ/NX-series Troubleshooting Manual (Cat. No. W503)*. For events that are affected by the Fail-soft Operation Setting, the operating specifications are specified for each set value.
Wait Time for Slave Startup	OK	Sets the time to wait from the detection of the link on the EtherCAT port until all slaves connect the network. Setting range: 3 to 200 s Default setting: 30 s Process data communications start if all slaves are present within the wait time. If all of the slaves are not present after the wait time elapses, a *Network Configuration Verification Error* (84220000 hex), *Network Configuration Verification Error (Slave Unconnected)* (84380000 hex), *Network Configuration Verification Error (Unnecessary Slave Connected)* (84320003 hex), *Network Configuration Verification Error (Mismatched Slave)* (84330004 hex), or *Network Configuration Verification Error (Incorrect Ring Wiring)* (843A0000 hex) event occurs. When you use a slave that takes time to start, use a longer wait time setting to prevent errors.
PDO Communications Timeout Detection Count	OK	A *Process Data Reception Timeout* (842B0000 hex) event occurs if process data (PDO) communications timeouts occur continuously more often than the specified number of times. Set the value to 2 or higher for a slave that needs to be replaced. Set the value to 3 or higher if the cable redundancy function is used. Setting range: 1 to 50 times[*11] Default setting: 2 errors

Name	Editing	Remarks
Revision Check Method	OK	Specifies the method to use to verify the revision numbers that are stored in the network configuration information (setting) against the actual revision numbers of the slaves (actual device) at the start of communications. Communications will not start if there are unverified slaves, and a *Network Configuration Verification Error* (84220000 hex) or *Network Configuration Verification Error (Mismatched Slave)* (84330004 hex) event will occur. Setting values: Setting <= Actual device, Setting = Actual device, or No check[*12][*13] Default setting: Setting <= Actual device
Serial Number Check Method[*14]	OK	Specifies whether to verify the slave serial numbers that are stored in the network configuration information against the serial numbers that are set in the actual slaves when initiating communications. Communications will not start if there are unverified slaves, and a *Network Configuration Verification Error* (84220000 hex) or *Network Configuration Verification Error (Mismatched Slave)* (84330004 hex) event will occur. Setting values: Setting = Actual device or No check Default setting: No check
DC Synchronous Correction[*15]	OK	Specifies whether to use the slave monitoring option during DC Synchronous Correction which ensures sync jitter accuracy of the DC time. If *Enable slave monitoring option* is selected, the master executes DC Synchronous Correction while monitoring the DC time of the slave. Use this setting if a *Slave Application Error* (84280000 hex) event occurs when the slaves are started. The error can be prevented if you select *Enable slave monitoring option* and reconnect the slave. If the error does not occur any more, that means a longer time is required for the startup of all slaves and slave reconnection. Setting range: Enable, or Disable slave monitoring option Default setting: Disable slave monitoring option

*1. Refer to *1-3 Specifications of Built-in EtherCAT Port* on page 1-10 for details on the maximum number of slaves.

*2. Displayed only with the NJ-series CPU Unit.

*3. Displayed only with the NX-series CPU Unit.

*4. For the NX102 and NX1P2 CPU Units, the default setting is 2,000 µs.

*5. When the priority-5 periodic task is not used, "---" is displayed.

*6. This can be used for project unit version 1.40 or later.

*7. Calculate the transmission delay time from the measured value in the actual network configuration if the *Process Data Reception Timeout* (842B0000 hex) or *EtherCAT Frame Not Received* (842E0000 hex) event occurs. Refer to *A-4 Setting Transmission Delay Time by Actual Measurement* on page A-31 for how to set the transmission delay time from the measured value in the actual network configuration.

*8. **Exist** is displayed with the NX-series CPU Unit, **Not Exist** is displayed with the NJ-series CPU Unit.

*9. Only when the total of cable lengths exceeds the default (1,000 m), set an appropriate value to the total cable length.

*10. For project unit version 1.40 or later, this value is 1 to 51,200 m.

*11. The range is 1 to 8 for the following CPU Units.

- NX102-□□00, NX1P2-□□□□□□, NJ501-1□00, NJ501-R□□□, NJ301-□□□□, and NJ101-□□00 with project unit version earlier than 1.40
- NX701-□□□□, NJ501-□□20, NJ501-1340, NJ501-5300, NJ501-4□□□, and NJ101-1□20 with project unit version earlier than 1.26
- NX102-□□20 with project unit version earlier than 1.37

*12. Refer to *Communications with Slaves Based on the Result of Revision Number Check* on page 5-18 for the operation of communications with slaves that are determined by the result of revision number check.

*13. If you enable the cable redundancy in the network configuration, set a value other than **No check**.

*14. If the set value is **Setting = Actual device**, communications are not started with a slave, in a network, that is replaced. If it is necessary to replace a slave while the set value is **Setting = Actual device**, you must correct the network configuration information and transfer it to the EtherCAT master again. Set this parameter to **Setting = Actual device** only when strict management of the equipment configuration is required.

*15. A CPU Unit with unit version 1.10 or later and Sysmac Studio version 1.12 or higher are required to use this function. However, for project unit version 1.40 or later, this setting is not supported. Select **Enable slave monitoring option** to use DC Synchronous Correction.

Precautions for Safe Use

- If the **Fail-soft Operation Setting** parameter is set to **Stop**, process data communications will stop for all slaves when an EtherCAT communications error is detected in a slave.
  The Servo Drive will operate according to the Servo Drive specifications.
  Make sure that the Fail-soft Operation parameter setting results in safe operation when a device error occurs.
- If the **Revision Check Method** is set to **No check**, parameters are also transferred to slaves with different EtherCAT revisions.
  If an incompatible revision of a slave is connected, incorrect parameters may be set and operation may not be correct.
  If you set the **Revision Check Method** to **No check**, make sure that only compatible slaves are connected before transferring the parameters.

Precautions for Correct Use

If you use an NX-series Safety Control Unit, set the *safety task period* and *FSoE Watchdog Timer* according to **PDO Communications Timeout Detection Count**. If they are not set, a timeout for safety process data communications may occur. Refer to the *NX-series Safety Control Unit User's Manual (Cat. No. Z930)* for details on the setting procedures.

● Communications with Slaves Based on the Result of Revision Number Check

Setting value of *Revision Check Method*	Result of revision number check	Communications with slaves
Setting <= Actual device	Setting > Actual device	Communications are not possible.
	Setting = Actual device	Communications are possible.
	Setting < Actual device	Communications are possible.
Setting = Actual device	Setting > Actual device	Communications are not possible.
	Setting = Actual device	Communications are possible.
	Setting < Actual device	Communications are not possible.
No check	Setting > Actual device	Communications are possible.
	Setting = Actual device	Communications are possible.
	Setting < Actual device	Communications are possible.

Additional Information

Network Configuration Verification

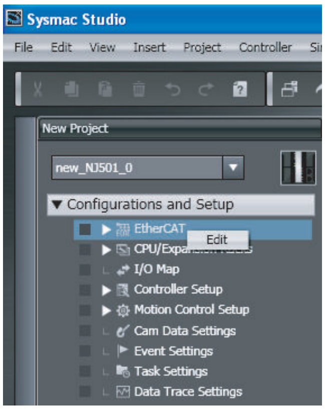
The network configuration information is verified against the actual network when the EtherCAT master is started. If an inconsistency is found during verification, a *Network Configuration Verification Error* (84220000 hex) or *Network Configuration Verification Error (Mismatched Slave)* (84330004 hex) event will occur.

Verified information	Required/optional	Description
Number of slaves connected	Required	Number of slaves in the network
Vendor ID	Required	Vendor ID of each slave
Product code	Required	Product code of each slave
Revision	Optional[*1]	Revision number of each slave
Serial number	Optional[*1]	Serial numbers of all slaves

[*1] Set this by the **Revision Check Method** and **Serial Number Check Method** in the EtherCAT master settings. You cannot set this setting for each slave.

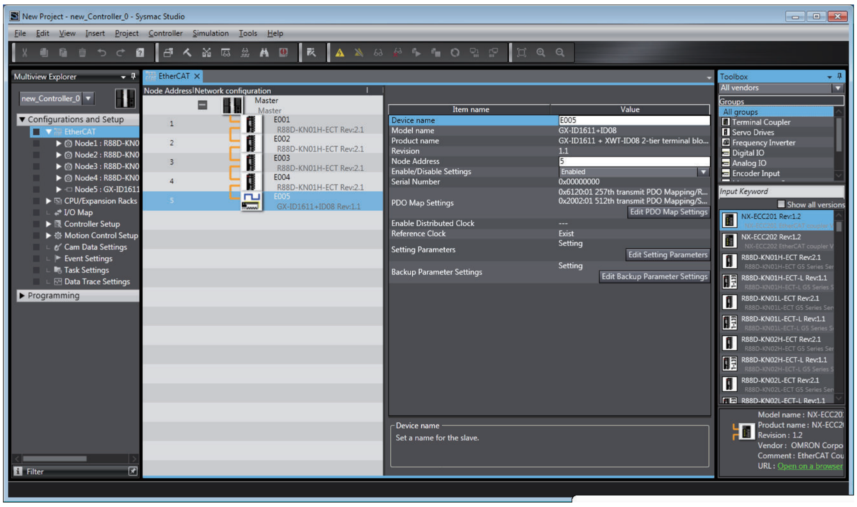
5-5 EtherCAT Slave Parameter Settings

1 Double-click **EtherCAT** under **Configurations and Setup** on the Multiview Explorer. Or, right-click **EtherCAT** under **Configurations and Setup** and select **Edit**.



2 Select an EtherCAT slave that is displayed in the EtherCAT Tab Page.

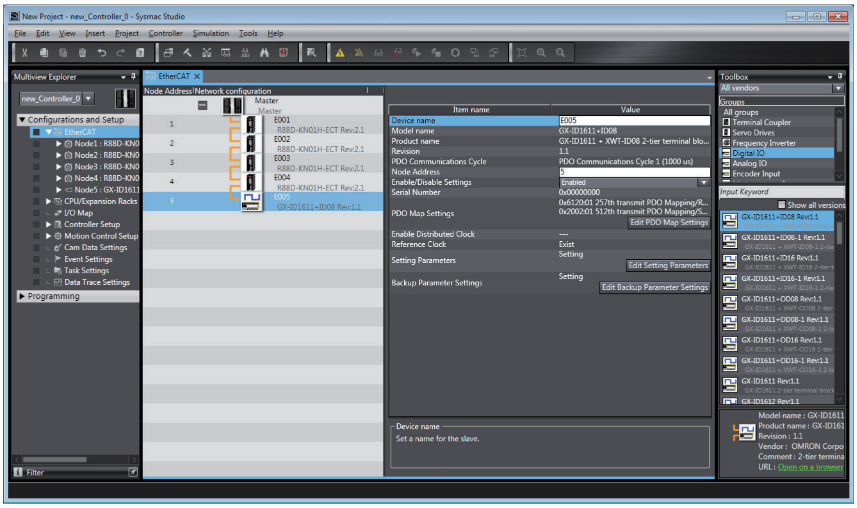
- For NJ-series CPU Units



Item name	Value
Device name	E005
Model name	GX-ID1611+ID08
Product name	GX-ID1611 + XWT-ID08 2-tier terminal blo...
Revision	1.1
Node Address	5
Enable/Disable Settings	Enabled
Serial Number	0x00000000
PDO Map Settings	0x6120:01 257th transmit PDO Mapping/R... 0x2002:01 512th transmit PDO Mapping/S... Edit PDO Map Settings
Enable Distributed Clock	---
Reference Clock	Exist
Setting Parameters	Setting Edit Setting Parameters
Backup Parameter Settings	Setting Edit Backup Parameter Settings

Device name
Set a name for the slave.

- For NX-series CPU Units



Item name	Value
Device name	E005
Model name	GX-ID1611+ID08
Product name	GX-ID1611 + XWT-ID08 2-tier terminal blo...
Revision	1.1
PDO Communications Cycle	PDO Communications Cycle 1 (1000 us)
Node Address	5
Enable/Disable Settings	Enabled
Serial Number	0x00000000
PDO Map Settings	0x6120:01 257th transmit PDO Mapping/R... 0x2002:01 512th transmit PDO Mapping/S... Edit PDO Map Settings
Enable Distributed Clock	---
Reference Clock	Exist
Setting Parameters	Setting Edit Setting Parameters
Backup Parameter Settings	Setting Edit Backup Parameter Settings

The EtherCAT slave settings are listed below.

Name	Edit-ing	Remarks
Device name	OK	Displays the name of the slave. Default setting: E*** (where * is a serial number starting from 001) The default value is automatically generated based on the node address.
Model name	Not allowed.	Automatically displays the name of the slave model.
Product name	Not allowed.	Displays the product name of the slave.
Revision	Not allowed.	Displays the revision of the slave.
PDO Communications Cycle[*1]	Not allowed.	Displays the process data communications cycle assigned for the slave. Either of the following values is displayed. • PDO Communications Cycle 1 Communications cycle is the task period of the primary periodic task. • PDO Communications Cycle 2 Communications cycle is the task period of the priority-5 periodic task.
Node Address	OK	Sets the node address. The default value is set automatically when a slave is added. Setting range: Settable node address range[*2]

Name	Editing	Remarks
Enable/Disable Settings	OK	Enables or disables the slave as a communications target. • Enabled: The slave will operate. • Disabled: The slave will not operate.[*3] Set the following slaves as disabled slaves: slaves that are not installed on the physical EtherCAT network but are scheduled for addition at a later date or slaves that not used due to changes in the device configuration during system operation. Even for disabled slaves, process data allocations can be set and used as system-defined variables and device variables in the user program. Default setting: Enabled
Serial Number	Not allowed.	Displays the serial number of the slave. Default: 0x00000000 (when offline) The value is updated to the serial number of the physical slave when you select **Get Slave Serial Numbers** from the menu for the master. A *Network Configuration Verification Error* (84220000 hex) or *Network Configuration Verification Error (Mismatched Slave)* (84330004 hex) event occurs if the serial number on the Sysmac Studio and the serial number of the physical slave do not agree when the **Serial Number Check Method** in the master settings is set to **Setting = Actual device**. Refer to *5-8 Using the Sysmac Studio to Obtain Serial Numbers from the Actual Network Configuration* on page 5-43 for how to access the serial number of the physical slave.
PDO Map Settings	OK	Default allocations of process data for slaves are provided by the Sysmac Studio. When the network configuration is created, device variables are automatically created in the initial process data allocations. You can read and write these process data allocations as device variables from the user program. When valid PDOs are present, a list of them is displayed. When valid PDOs are not present, "---" is displayed. If editing the process data allocations is required, click the **Edit PDO Map Settings** Button at the bottom of the list. Refer to *5-5-1 Changing the PDO Map Settings* on page 5-23 for details.
Enable Distributed Clock	OK	If a slave provides a distributed clock, **Enabled** is displayed. If a slave provides a distribute clock which allows disable setting, either **Enabled** or **Disabled** is displayed. You can select enabling or disabling the clock. If the distributed clock is enabled, you can select the synchronization timing (Sync0 or Sync1) for some slaves. If Sync0 is selected, **Enabled (DC Sync0)** is displayed. If a slave does not provide a distributed clock, "---" is displayed.
Reference Clock	Not allowed.	Displays whether the slave provides a reference clock.[*4] **Exist** is displayed if the slave provides a reference clock. **Not exist** is displayed if the slave does not provide a reference clock.

Name	Editing	Remarks
Setting Parameters	OK	If a slave has an initial parameter setting function, **Setting** is displayed. If a slave does not have an initial parameter setting function, "---" is displayed. To edit the settings, click the **Edit Setting Parameters** Button.
Backup Parameter Settings	OK	If a slave has a backup parameter setting function, **Setting** is displayed. If a slave does not have a backup parameter setting function, "---" is displayed. To edit the settings, click the **Edit Backup Parameter Settings** Button.

*1. Displayed only with the NX-series CPU Unit.

*2. Refer to *1-3 Specifications of Built-in EtherCAT Port* on page 1-10 for details on the settable node address range.

*3. Network errors do not occur even if disabled slaves are not present on the EtherCAT network. You can use them to prepare for future system expansions or operate without them for the current system. You can execute a special instruction to enable a disabled slave during system operation. You can also execute the special instruction to disable a slave during system operation.

*4. With the NJ-series CPU Unit, if you configure the system to connect the EtherCAT Junction Slave and to synchronize the slaves that have an enabled distributed clock, there must be a slave that provides a reference clock between the Master and the first EtherCAT Junction Slave. Or, the first EtherCAT Junction Slave must provide a reference clock.

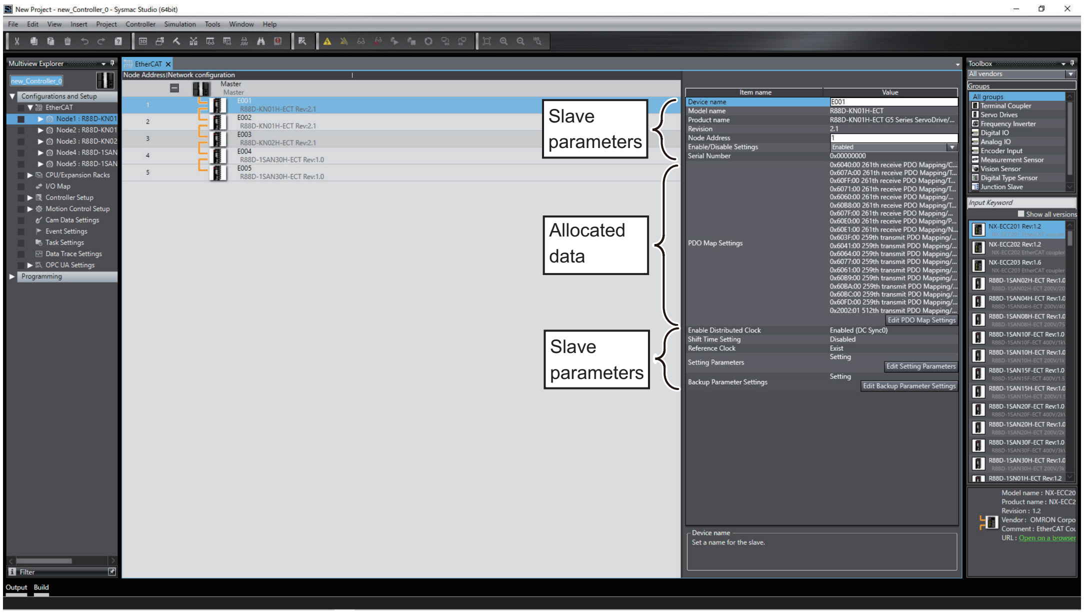
5-5-1 Changing the PDO Map Settings

Default allocations of process data for slaves are defined in the ESI files.

Depending on the slave, you may have to change the process data assignment as required by the application.

The operations on the Sysmac Studio is shown as follows.

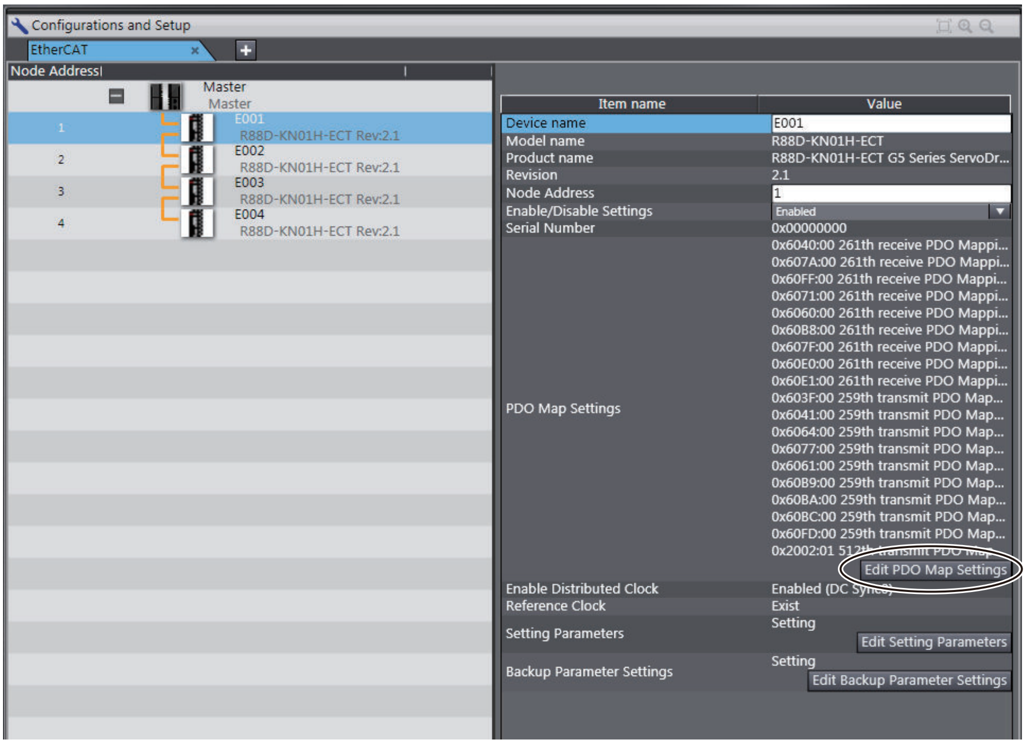
1 Select the slave for which to change the allocated data. The parameters and allocated data for the slave will appear.



Item name	Value
Device name	E001
Model name	R88D-KN01H-ECT
Product name	R88D-KN01H-ECT G5 Series ServoDrive/...
Revision	2.1
Node Address	1
Enable/Disable Settings	Enabled
Serial Number	0x00000000
PDO Map Settings	0x6040:00 261th receive PDO Mapping/C... 0x607A:00 261th receive PDO Mapping/T... 0x60FF:00 261th receive PDO Mapping/T... 0x6071:00 261th receive PDO Mapping/T... 0x6060:00 261th receive PDO Mapping/... 0x60B8:00 261th receive PDO Mapping/T... 0x607F:00 261th receive PDO Mapping/... 0x60E0:00 261th receive PDO Mapping/P... 0x60E1:00 261th receive PDO Mapping/N... 0x603F:00 259th transmit PDO Mapping/... 0x6041:00 259th transmit PDO Mapping/... 0x6064:00 259th transmit PDO Mapping/... 0x6077:00 259th transmit PDO Mapping/... 0x6061:00 259th transmit PDO Mapping/... 0x60B9:00 259th transmit PDO Mapping/... 0x60BA:00 259th transmit PDO Mapping/... 0x60BC:00 259th transmit PDO Mapping/... 0x60FD:00 259th transmit PDO Mapping/... 0x2002:01 512th transmit PDO Mapping/... Edit PDO Map Settings
Enable Distributed Clock	Enabled (DC Sync0)
Shift Time Setting	Disabled
Reference Clock	Exist
Setting Parameters	Setting Edit Setting Parameters
Backup Parameter Settings	Setting Edit Backup Parameter Settings

Device name
Set a name for the slave.

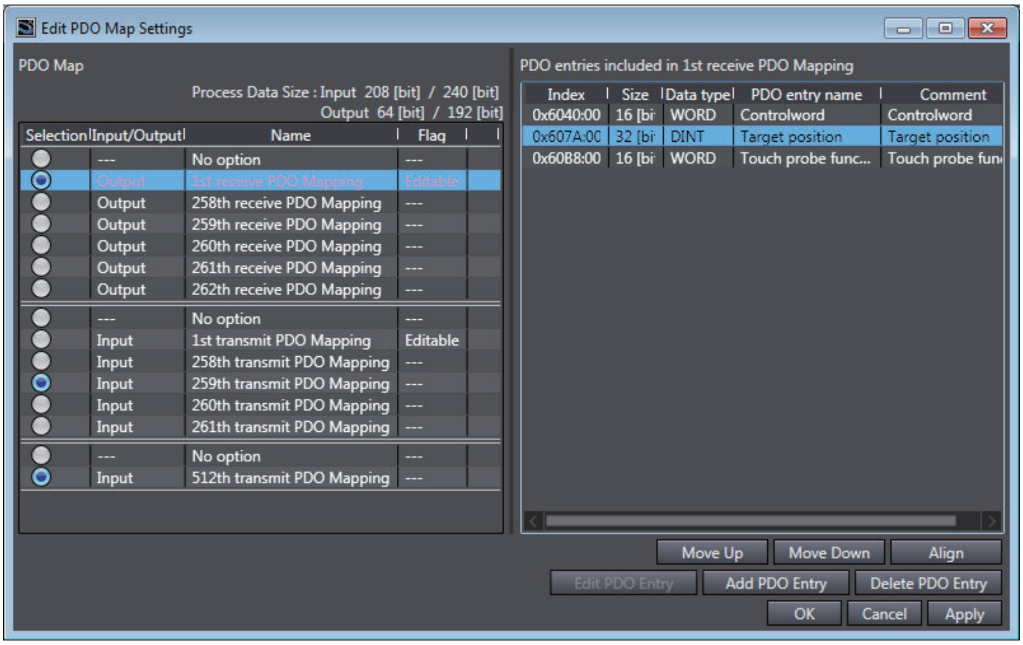
2 Click the **Edit PDO Map Settings** Button at the lower right of the allocated data in the pane.



Node Address	
	Master Master
1	E001 R88D-KN01H-ECT Rev:2.1
2	E002 R88D-KN01H-ECT Rev:2.1
3	E003 R88D-KN01H-ECT Rev:2.1
4	E004 R88D-KN01H-ECT Rev:2.1

Item name	Value
Device name	E001
Model name	R88D-KN01H-ECT
Product name	R88D-KN01H-ECT G5 Series ServoDr...
Revision	2.1
Node Address	1
Enable/Disable Settings	Enabled
Serial Number	0x00000000
PDO Map Settings	0x6040:00 261th receive PDO Mappi... 0x607A:00 261th receive PDO Mappi... 0x60FF:00 261th receive PDO Mappi... 0x6071:00 261th receive PDO Mappi... 0x6060:00 261th receive PDO Mappi... 0x60B8:00 261th receive PDO Mappi... 0x607F:00 261th receive PDO Mappi... 0x60E0:00 261th receive PDO Mappi... 0x60E1:00 261th receive PDO Mappi... 0x603F:00 259th transmit PDO Map... 0x6041:00 259th transmit PDO Map... 0x6064:00 259th transmit PDO Map... 0x6077:00 259th transmit PDO Map... 0x6061:00 259th transmit PDO Map... 0x60B9:00 259th transmit PDO Map... 0x60BA:00 259th transmit PDO Map... 0x60BC:00 259th transmit PDO Map... 0x60FD:00 259th transmit PDO Map... 0x2002:01 512th transmit PDO Map...
Enable Distributed Clock	Enabled (DC Synch)
Reference Clock	Exist
Setting Parameters	Setting
Backup Parameter Settings	Setting

The **Edit PDO Map Settings** Window will appear. Edit the allocated data as required.



PDO Map

Process Data Size : Input 208 [bit] / 240 [bit]
Output 64 [bit] / 192 [bit]

Selection	Input/Output	Name	Flag
○	---	No option	---
●	Output	1st receive PDO Mapping	Editable
○	Output	258th receive PDO Mapping	---
○	Output	259th receive PDO Mapping	---
○	Output	260th receive PDO Mapping	---
○	Output	261th receive PDO Mapping	---
○	Output	262th receive PDO Mapping	---
○	---	No option	---
○	Input	1st transmit PDO Mapping	Editable
○	Input	258th transmit PDO Mapping	---
●	Input	259th transmit PDO Mapping	---
○	Input	260th transmit PDO Mapping	---
○	Input	261th transmit PDO Mapping	---
○	---	No option	---
●	Input	512th transmit PDO Mapping	---

PDO entries included in 1st receive PDO Mapping

Index	Size	Data type	PDO entry name	Comment
0x6040:00	16 [bi	WORD	Controlword	Controlword
0x607A:00	32 [bi	DINT	Target position	Target position
0x60B8:00	16 [bi	WORD	Touch probe func...	Touch probe fun

3 Select the output data (RxPDO) and input data (TxPDO) in the **PDO Map**.
You can add or delete the PDOs with **Editable** Flags as entries for the objects to use for the slaves. Refer to *A-3-4 Editing PDO Entry Tables* on page A-21 for details.

4 Click the **OK** Button.

5-5-2 EtherCAT Slave Enable/Disable Settings

You can temporarily stop and start process data communications with a specified EtherCAT slave without stopping the entire communications system. When an EtherCAT slave is disabled, only message communications can be performed. Use the Sysmac Studio or the instruction in the user program to enable or disable EtherCAT slaves.

Version Information

A CPU Unit with unit version 1.04 or later and Sysmac Studio version 1.05 or higher are required to disable EtherCAT slaves with the special instruction.

Precautions for Correct Use

- If an EtherCAT slave is enabled with the EC_ChangeEnableSetting (Enable/Disable EtherCAT Slave) instruction, check the element of the EtherCAT slave that will be enabled is TRUE in the *_EC_EntrySlavTbl[]* (Network Connected Slave Table) system-defined variable.
- If the cable redundancy setting is enabled, you cannot use the EC_ChangeEnableSetting (Enable/Disable EtherCAT Slave) instruction to enable or disable the EtherCAT slave in the ring topology. You can set to enable or disable EtherCAT slaves from the Sysmac Studio.

● Relation between EtherCAT Slave Status and Enable/Disable Instruction Settings

The following table gives the results after instruction execution with the EC_ChangeEnableSetting (Enable/Disable EtherCAT Slave) instruction for the EtherCAT slave status.

EtherCAT slave status before instruction execution	Instruction setting	Result after instruction execution	
		Instruction execution result	EtherCAT slave enabled/disabled status
Enable	Enable	Normal end	Enabled (not changed)
	Disable		Disabled
Disabled with physical slaves	Enable		Enabled
	Disable		Disabled (not changed)
Disabled with no physical slave	Enable	Error end	Disabled (not changed)
	Disable		
Enable with disconnected *1	Enable		Enabled (not changed)
	Disable		

*1. You cannot use the instruction to disable or enable the slave if it is disconnected or in an error state. Execute a reconnection command or an error reset to the slave before you execute the instruction.

Precautions for Correct Use

If you use the special instruction to change the enable/disable setting of a slave, turning OFF the power supply to the master will discard the change and revert the setting to those set from Sysmac Studio. To retain the changed setting, you must change the setting from the Sysmac Studio and then transfer the network configuration information to the CPU Unit.

● Relation between Changes in the Enable/Disable Setting and I/O Operations

The I/O operations that occur for changes in the enable/disable settings of an EtherCAT slave are given in the following table.

Change in status	Inputs		Outputs	
	Input refresh values	Device variable values	Device variable values	Output refresh values
Disabled slaves after turning ON the power supply or transferring the EtherCAT network configuration information	Input refreshing is performed but not applied to the device variables.	0 or FALSE	Depends on results of user program execution.	Output refreshing is not performed.
Slaves changed from disabled to enabled	Input refreshing is performed.	Input refresh values are applied.	Depends on results of user program execution.	The values of the device variables are output.
Slaves changed from enabled to disabled	Input refreshing is performed but not applied to the device variables.	The values from before the status of the slave was changed are retained.	Depends on results of user program execution.	Depends on the settings of the slave and on slave models. With GX-series EtherCAT Slaves, a value is output according to the *Error Mode Output* setting on each slave.

Precautions for Correct Use

To access the I/O devices variable values of a slave after you disable or enable the slave, you need to check the validity of the process data. Refer to *6-1-3 Checking Validity of Process Data* on page 6-6 for how to check the validity of the process data.

● Relation between Operation and the Slave Information at a Disabled Slave

If the actual network configuration has a slave in a position of a slave set to **Disable** (or disabled slave), the slave operation depends on the slave information. The operation differs by the result of comparison between the slave information of the disabled slave and the slave information of the slave on the actual network configuration. The table below gives the operation of the disabled slave.

Slave information		Operation for disabled slave			
Node address	Vendor ID/ Product code	Message communications	Process data communications	Error	Backup, restore, and compare operations
Same	Same	Performed	Not performed	Normal operation (no error)	Performed
	Different	Not performed		A *Network Configuration Verification Error* (84220000 hex) or *Network Configuration Verification Error (Mismatched Slave)* (84330004 hex) event occurs.	Not performed
Different	Same				
	Different				

5-6 Comparing and Merging EtherCAT Network Configurations

5-6-1 Introduction for Comparing and Merging EtherCAT Network Configurations

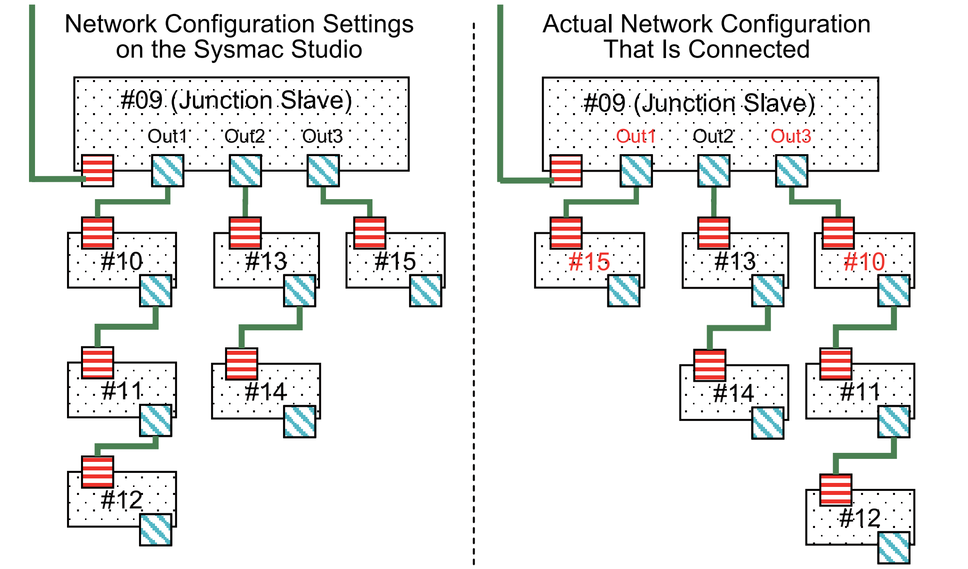
Use the Sysmac Studio to compare the network configuration information settings in the Sysmac Studio with the actual network configuration that is connected, including branch locations of a branching topology.

The following items are compared.

- Node Addresses
- Vendor IDs
- Product codes
- Revisions
- Connection ports

If the settings in the Sysmac Studio and the actual network configuration match completely, it is possible to identify the locations of errors when they occur.

If differences are shown in the comparison results, merge the configurations from the Sysmac Studio.



The output port numbers for the branches do not match the node addresses for the connected slaves. In this example, the settings on the Sysmac Studio and the slaves connected to Out1 and Out3 on node #09 in the actual network configuration do not match.

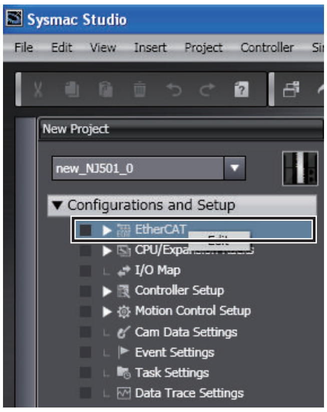
🟥 Input port ▨ Output port "#" indicates a node address.

5-6-2 Operation Procedure for Comparing and Merging EtherCAT Network Configurations

Use the following procedure to compare and merge the network configuration settings on the Sysmac Studio and the actual network configuration.

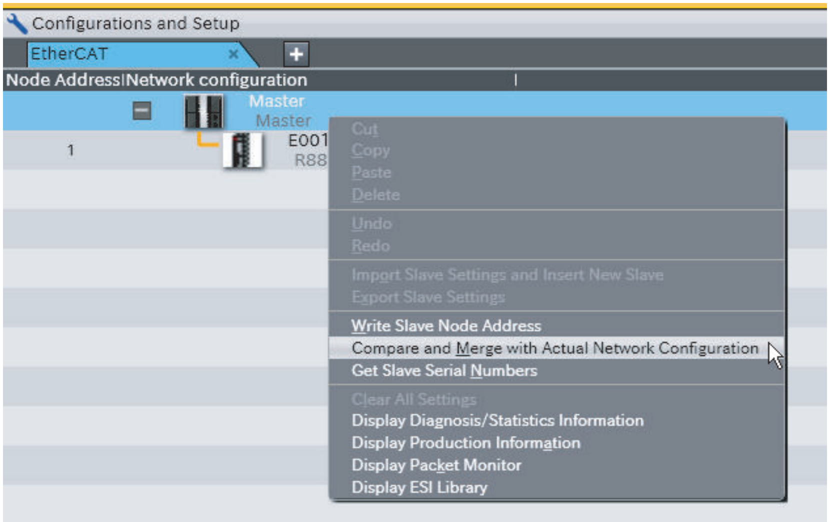
1 Start the Sysmac Studio and go online with the Controller.

2 Double-click **EtherCAT** under **Configurations and Setup** on the Multiview Explorer. Or, right-click **EtherCAT** under **Configurations and Setup** and select **Edit**.

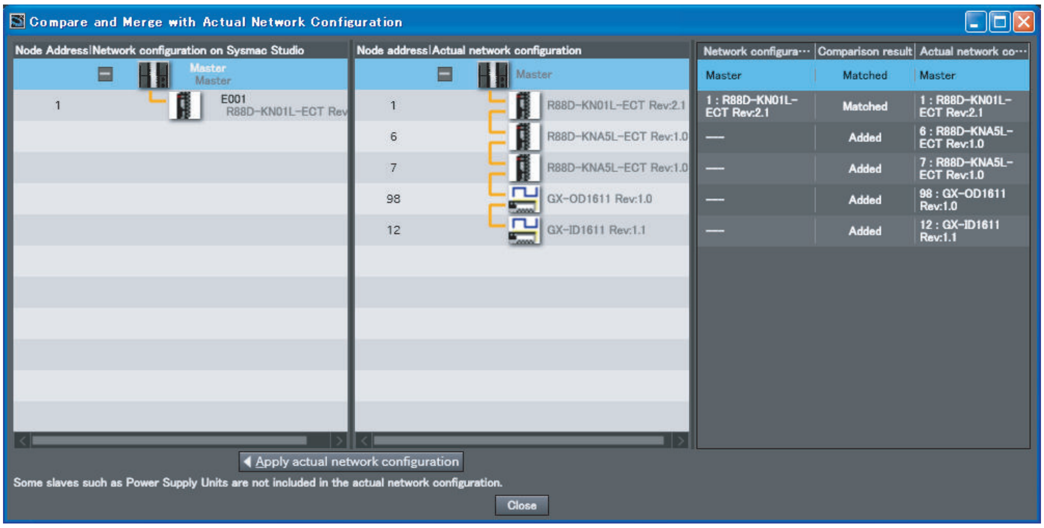


The EtherCAT Tab Page is displayed.

3 Right-click the EtherCAT master that is displayed in the EtherCAT Tab Page and select **Compare and Merge with Actual Network Configuration**.



The **Compare and Merge with Actual Network Configuration** Dialog Box is displayed. The results of comparing the settings on the Sysmac Studio with the actual network configuration are displayed in the **Comparison results** Column.


Compare and Merge with Actual Network Configuration

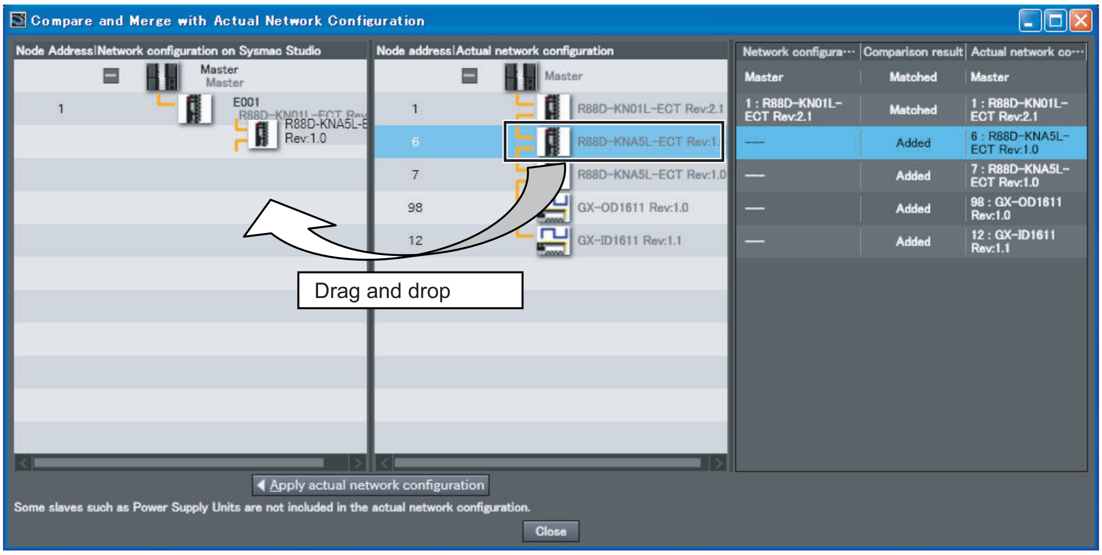
Node Address	Network configuration on Sysmac Studio
	Master
1	E001 R88D-KN01L-ECT Rev

Node address	Actual network configuration
	Master
1	R88D-KN01L-ECT Rev:2.1
6	R88D-KNA5L-ECT Rev:1.0
7	R88D-KNA5L-ECT Rev:1.0
98	GX-OD1611 Rev:1.0
12	GX-ID1611 Rev:1.1

Network configura···	Comparison result	Actual network co···
Master	Matched	Master
1 : R88D-KN01L-ECT Rev:2.1	Matched	1 : R88D-KN01L-ECT Rev:2.1
—	Added	6 : R88D-KNA5L-ECT Rev:1.0
—	Added	7 : R88D-KNA5L-ECT Rev:1.0
—	Added	98 : GX-OD1611 Rev:1.0
—	Added	12 : GX-ID1611 Rev:1.1

Some slaves such as Power Supply Units are not included in the actual network configuration.

4 If there is a slave that exists only on the actual network configuration, **Added** is displayed in the **Comparison results** Column. Drag the slave in the **Actual network configuration** to the **Network configuration on Sysmac Studio** and drop it.


Compare and Merge with Actual Network Configuration

Node Address	Network configuration on Sysmac Studio
	Master
1	E001 R88D-KN01L-ECT Rev R88D-KNA5L-E Rev:1.0

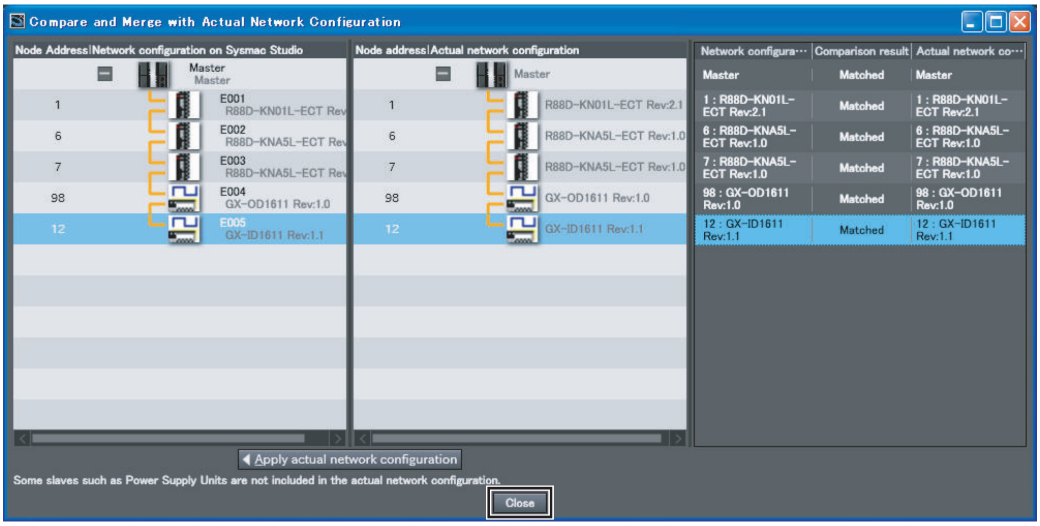
Node address	Actual network configuration
	Master
1	R88D-KN01L-ECT Rev:2.1
6	R88D-KNA5L-ECT Rev:1.
7	R88D-KNA5L-ECT Rev:1.0
98	GX-OD1611 Rev:1.0
12	GX-ID1611 Rev:1.1

Network configura···	Comparison result	Actual network co···
Master	Matched	Master
1 : R88D-KN01L-ECT Rev:2.1	Matched	1 : R88D-KN01L-ECT Rev:2.1
—	Added	6 : R88D-KNA5L-ECT Rev:1.0
—	Added	7 : R88D-KNA5L-ECT Rev:1.0
—	Added	98 : GX-OD1611 Rev:1.0
—	Added	12 : GX-ID1611 Rev:1.1

Some slaves such as Power Supply Units are not included in the actual network configuration.

After you drag and drop the missing slaves and add them to the **Network configuration on Sysmac Studio**, the **Comparison results** Column will show that everything in the configurations matches.

5 Click the **Close** Button.



Compare and Merge with Actual Network Configuration

Node Address	Network configuration on Sysmac Studio
	Master Master
1	E001 R88D-KN01L-ECT Rev
6	E002 R88D-KNA5L-ECT Rev
7	E003 R88D-KNA5L-ECT Rev
98	E004 GX-OD1611 Rev:1.0
12	E005 GX-ID1611 Rev:1.1

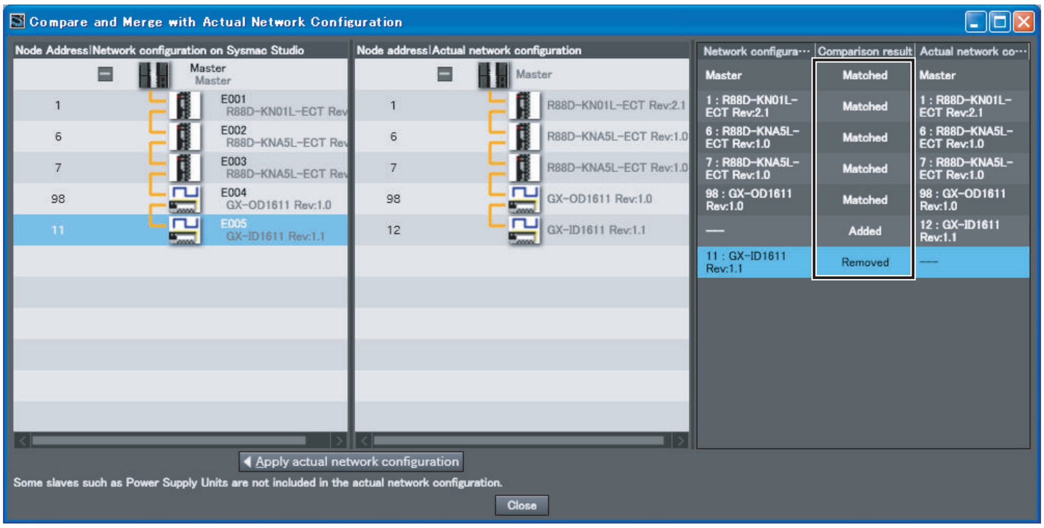
Node address	Actual network configuration
	Master
1	R88D-KN01L-ECT Rev:2.1
6	R88D-KNA5L-ECT Rev:1.0
7	R88D-KNA5L-ECT Rev:1.0
98	GX-OD1611 Rev:1.0
12	GX-ID1611 Rev:1.1

Network configura···	Comparison result	Actual network co···
Master	Matched	Master
1 : R88D-KN01L-ECT Rev:2.1	Matched	1 : R88D-KN01L-ECT Rev:2.1
6 : R88D-KNA5L-ECT Rev:1.0	Matched	6 : R88D-KNA5L-ECT Rev:1.0
7 : R88D-KNA5L-ECT Rev:1.0	Matched	7 : R88D-KNA5L-ECT Rev:1.0
98 : GX-OD1611 Rev:1.0	Matched	98 : GX-OD1611 Rev:1.0
12 : GX-ID1611 Rev:1.1	Matched	12 : GX-ID1611 Rev:1.1

Some slaves such as Power Supply Units are not included in the actual network configuration.

The Sysmac Studio returns to the EtherCAT Tab Page. This completes the operation to compare and merge with the actual network configuration.

Removed is displayed in the **Comparison results** Column for a slaves that exist only in the settings on the Sysmac Studio. The following example shows a case where a node address is different in the settings on the Sysmac Studio and in the actual network configuration. As a result, **Added** and **Removed** are displayed in the **Comparison results** Column.



Compare and Merge with Actual Network Configuration

Node Address	Network configuration on Sysmac Studio
	Master Master
1	E001 R88D-KN01L-ECT Rev
6	E002 R88D-KNA5L-ECT Rev
7	E003 R88D-KNA5L-ECT Rev
98	E004 GX-OD1611 Rev:1.0
11	E005 GX-ID1611 Rev:1.1

Node address	Actual network configuration
	Master
1	R88D-KN01L-ECT Rev:2.1
6	R88D-KNA5L-ECT Rev:1.0
7	R88D-KNA5L-ECT Rev:1.0
98	GX-OD1611 Rev:1.0
12	GX-ID1611 Rev:1.1

Network configura···	Comparison result	Actual network co···
Master	Matched	Master
1 : R88D-KN01L-ECT Rev:2.1	Matched	1 : R88D-KN01L-ECT Rev:2.1
6 : R88D-KNA5L-ECT Rev:1.0	Matched	6 : R88D-KNA5L-ECT Rev:1.0
7 : R88D-KNA5L-ECT Rev:1.0	Matched	7 : R88D-KNA5L-ECT Rev:1.0
98 : GX-OD1611 Rev:1.0	Matched	98 : GX-OD1611 Rev:1.0
----	Added	12 : GX-ID1611 Rev:1.1
11 : GX-ID1611 Rev:1.1	Removed	----

Some slaves such as Power Supply Units are not included in the actual network configuration.

Here, either correct the node address in the network configuration on the Sysmac Studio or the node address in the physical slave in the actual network configuration, and then repeat the compare and merge operation to confirm that the configurations are the same.

Additional Information

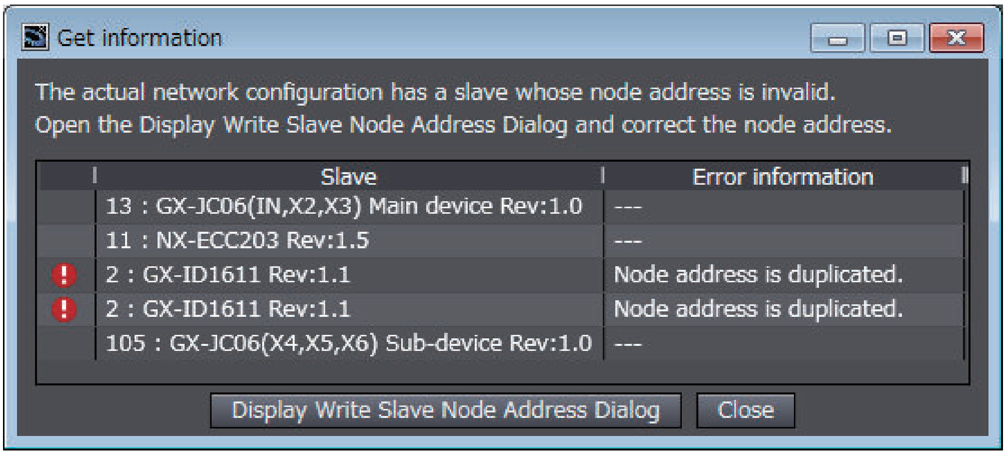
- In the following cases, the **Compare and Merge with Actual Network Configuration** Dialog Box is not displayed. Refer to *5-6-3 When **Compare and Merge with Actual Network Configuration** Dialog Box Is Not Displayed* on page 5-32 to remove the cause of the error, and perform the compare and merge operation.
  a) The slave node address settings are not correct on the actual network.
  b) The communications cable is not wired correctly.
  c) More than the maximum number of slaves are connected.
  In addition, if an *EtherCAT Frame Not Received* (842E0000 hex) event occurs, refer to the *NJ/NX-series Troubleshooting Manual (Cat. No. W503)* to remove the cause of the error, and perform the compare and merge operation.
- After you execute the **Compare and Merge with Actual Network Configuration** and even several tens of seconds has passed, if slaves that are connected on the actual network configuration are not displayed in the **Compare and Merge with Actual Network Configuration** Dialog Box, refer to *5-6-4 When Slaves on Actual Network Configuration Are Not Displayed* on page 5-39 to remove the cause of the error, and perform the compare and merge operation.
- After the operation to compare and merge with the actual network configuration is completed, if there is a ring topology in the network configuration on the Sysmac Studio, refer to *Components of the Ring Topology* on page 4-5 to check that the appropriate slaves are connected.
- When the compare and merge operation is executed for the actual network configuration, the synchronization between the Sysmac Studio and the Controller is lost. Synchronize the Sysmac Studio and Controller before you perform any online operations for the slaves.
  Refer to the *Sysmac Studio Version 1 Operation Manual (Cat. No. W504)* for information on synchronization.
- For project unit version 1.40 or later, if the synchronization between the Sysmac Studio and the Controller is inconsistent, you may not be able to obtain the module configuration information on MDP-compatible slaves. Refer to *A-3-5 Settings for MDP-compatible Slaves from Other Manufacturers* on page A-27 for information on MDP-compatible slaves.
  If you failed to obtain the module configuration information for MDP-compatible slaves, you can obtain the information by the following steps.
  1. Set the **Module config send method** parameter for the MDP-compatible slaves to **Do not send** and perform synchronous transfer.
  2. Perform the compare and merge operation.

5-6-3 When Compare and Merge with Actual Network Configuration Dialog Box Is Not Displayed

If comparison between the network configuration on the Sysmac Studio and the actual network configuration cannot be preformed, an error message is displayed. Take appropriate measures according to the displayed error message and then compare the network configuration on the Sysmac Studio and the actual network configuration again.

When Slave Node Address Settings Are Not Correct

If the slave node address settings are not correct, the following error message is displayed. Write the slave node addresses or reset the node addresses using the hardware switches for the slaves displayed in the message, and then compare the network configuration on the Sysmac Studio and the actual network configuration again. Refer to *2-2 Setting the Node Addresses of the EtherCAT Slaves* on page 2-9 for how to write the slave node addresses or reset the node addresses using the hardware switches.



Get information

The actual network configuration has a slave whose node address is invalid.
Open the Display Write Slave Node Address Dialog and correct the node address.

	Slave	Error information
	13 : GX-JC06(IN,X2,X3) Main device Rev:1.0	---
	11 : NX-ECC203 Rev:1.5	---
!	2 : GX-ID1611 Rev:1.1	Node address is duplicated.
!	2 : GX-ID1611 Rev:1.1	Node address is duplicated.
	105 : GX-JC06(X4,X5,X6) Sub-device Rev:1.0	---

Version Information

Sysmac Studio version 1.22 or higher is required to show the list of the incorrect settings of slave node address.

When Communications Cables Are Not Wired Correctly

If the communications cables are not wired correctly, an error message that is indicated the incorrect wiring is displayed. Correct the wiring for communications cables and then compare the network configuration on the Sysmac Studio and the actual network configuration again.

Version Information

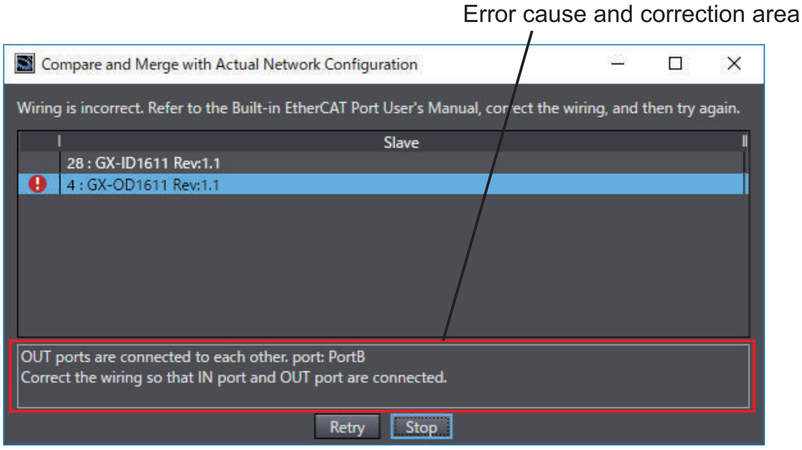
Sysmac Studio version 1.29 or higher is required to use the function to display incorrect wiring in a ring topology.

There are following cases weather you can identify incorrect wiring points of communications cables.

● When You Can Identify Incorrect Wiring Points of Communications Cables

The following dialog box is displayed, and any incorrect wiring point in the list is indicated with an exclamation mark (!).

In addition, in the error cause and correction area, an error message is displayed depending on the error cause.


Error cause and correction area
Compare and Merge with Actual Network Configuration
Slave
Retry
Stop

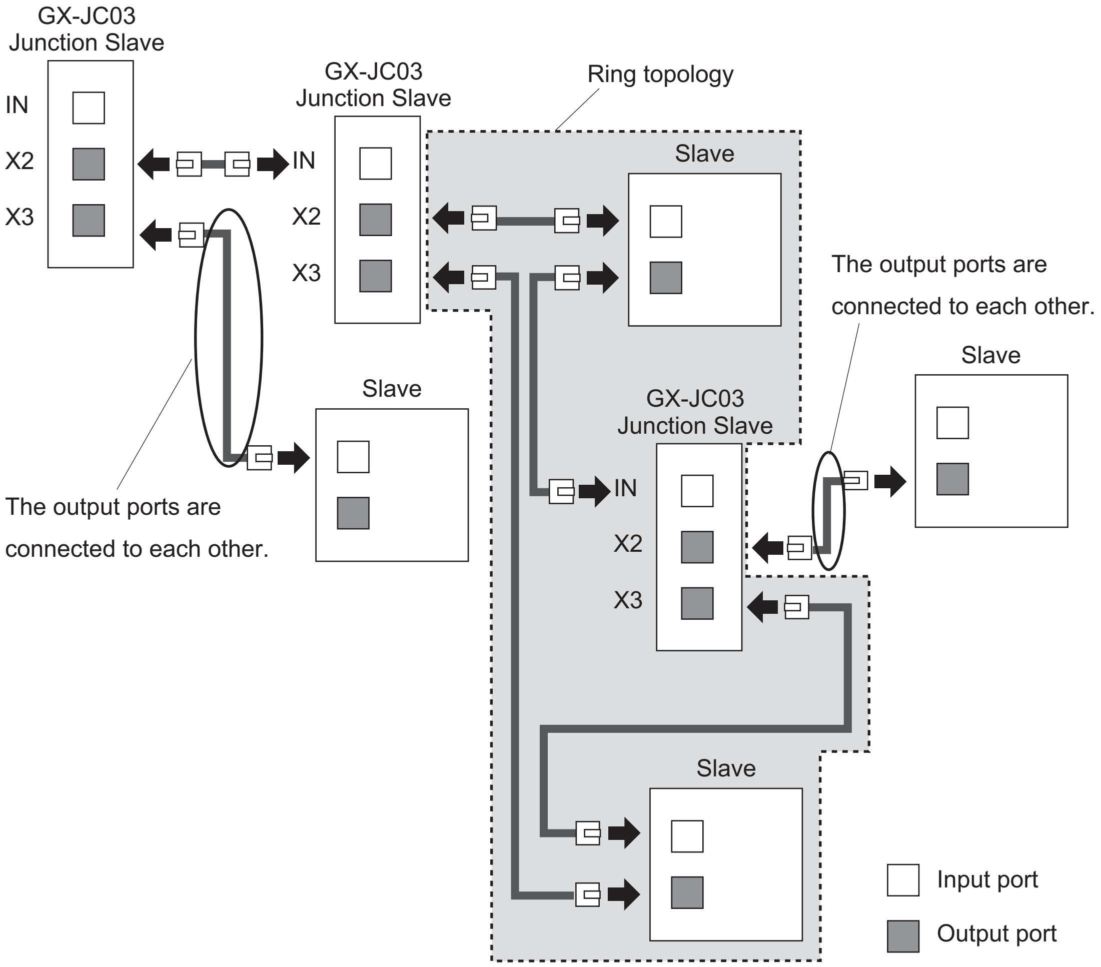
Wiring is incorrect. Refer to the Built-in EtherCAT Port User's Manual, correct the wiring, and then try again.

28 : GX-ID1611 Rev:1.1
4 : GX-OD1611 Rev:1.1

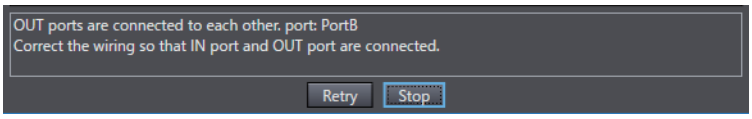
OUT ports are connected to each other. port: PortB
Correct the wiring so that IN port and OUT port are connected.

The error message and correction for each error cause are described below.

- Error cause 1: Output ports are connected to each other between slaves outside the ring topology.



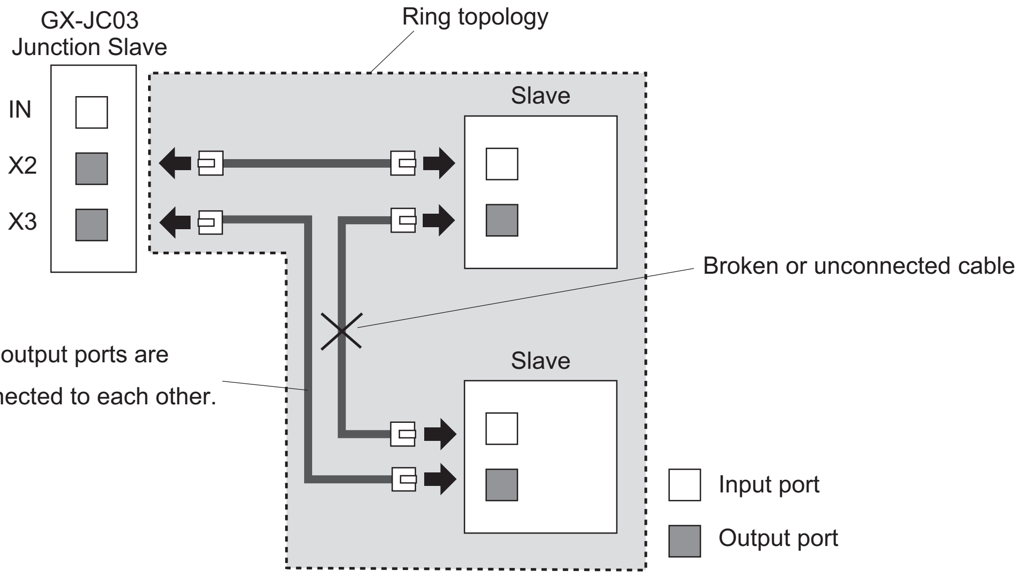
The following error message is displayed.



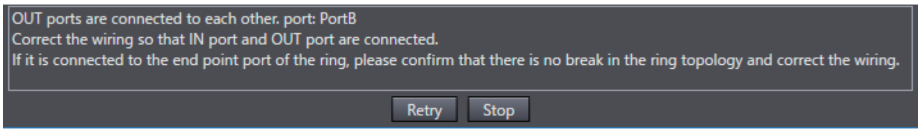
OUT ports are connected to each other. port: PortB
Correct the wiring so that IN port and OUT port are connected.

Correct the error as follows.
Review the wiring to connect the output port of the slave indicated in the list to the input port of the destination slave.

- Error cause 2: There is a broken or unconnected cable in the ring topology.


GX-JC03
Junction Slave
Ring topology
IN
X2
X3
Slave
Broken or unconnected cable
Slave
The output ports are connected to each other.
Input port
Output port

The following error message is displayed.

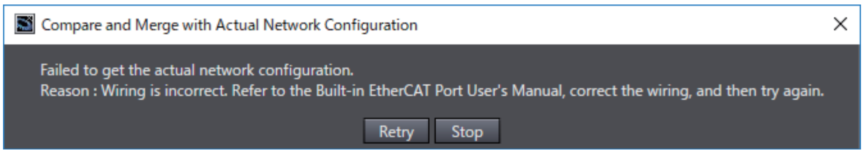


OUT ports are connected to each other. port: PortB
Correct the wiring so that IN port and OUT port are connected.
If it is connected to the end point port of the ring, please confirm that there is no break in the ring topology and correct the wiring.

Correct the error as follows.
If you configure a ring topology, replace the broken cable or make other wiring corrections. If you do not configure a ring topology, review the wiring to the X3 port of the Junction Slave so that output ports are not connected to each other.

● When You Cannot Identify Incorrect Wiring Points of Communications Cables

The following error message is displayed.

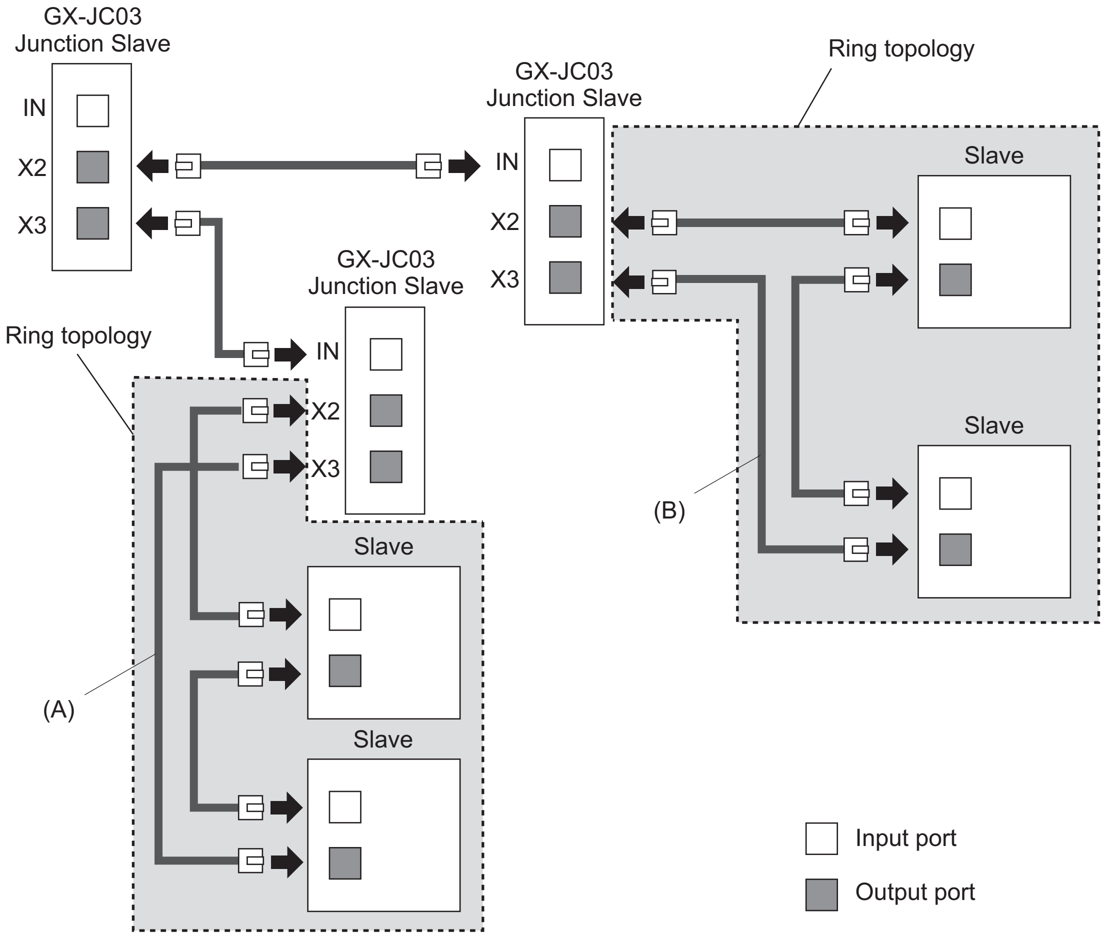


Failed to get the actual network configuration.
Reason : Wiring is incorrect. Refer to the Built-in EtherCAT Port User's Manual, correct the wiring, and then try again.

The correction for each cause of incorrect wiring is described as follows.

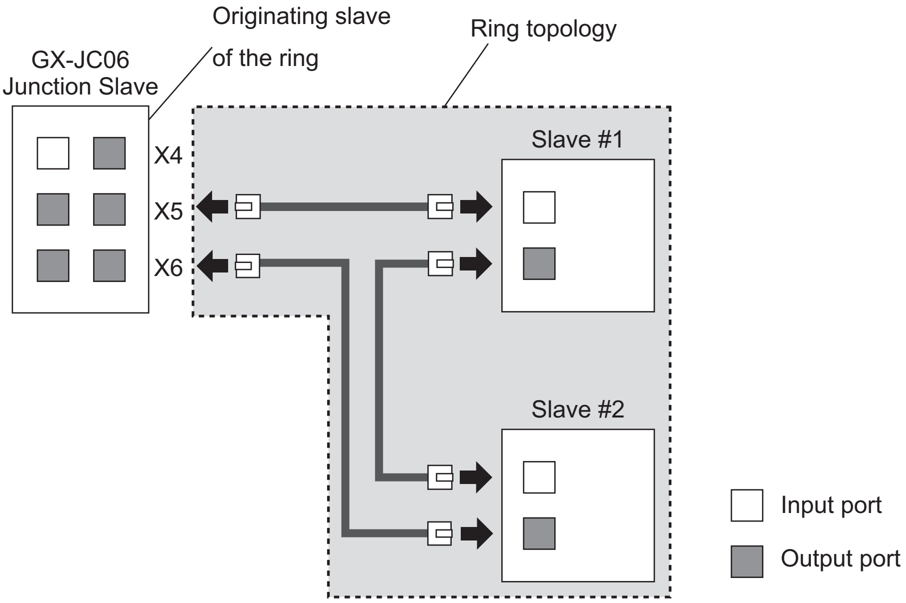
- Cause 1 of incorrect wiring: There is more than one ring topology.
  Check that there is one ring topology.
  You can configure a ring topology in only one location.

In the example below, there are two ring topologies in the actual network configuration. Disconnect cable (A) or (B) that connects output ports to each other to have only one ring topology.



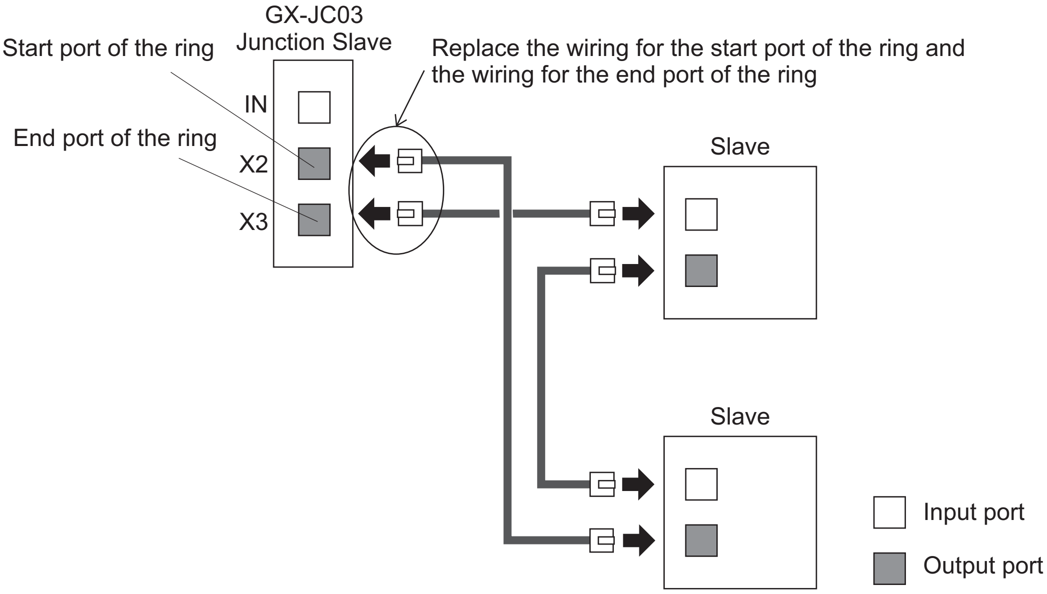
- Cause 2 of incorrect wiring: A ring topology is configured, but the Junction Slave that cannot be used for the originating slave of the ring is used.
  For the collection, if you configure a ring topology, check that the GX-JC03 Junction Slave is used for the originating slave of the ring. If you do not configure a ring topology, review the wiring so that output ports are not connected to each other.

  In the example below, if you configure a ring topology, change the GX-JC06 Junction Slave to the GX-JC03 Junction Slave, and then reconnect the cable that is currently connected to X5 and X6 of the Junction Slave to X2 and X3, respectively. If you do not configure a ring topology, remove the cable that is connected to X6 or connect the cable that is connected to X6 to the input port of the Slave #2.



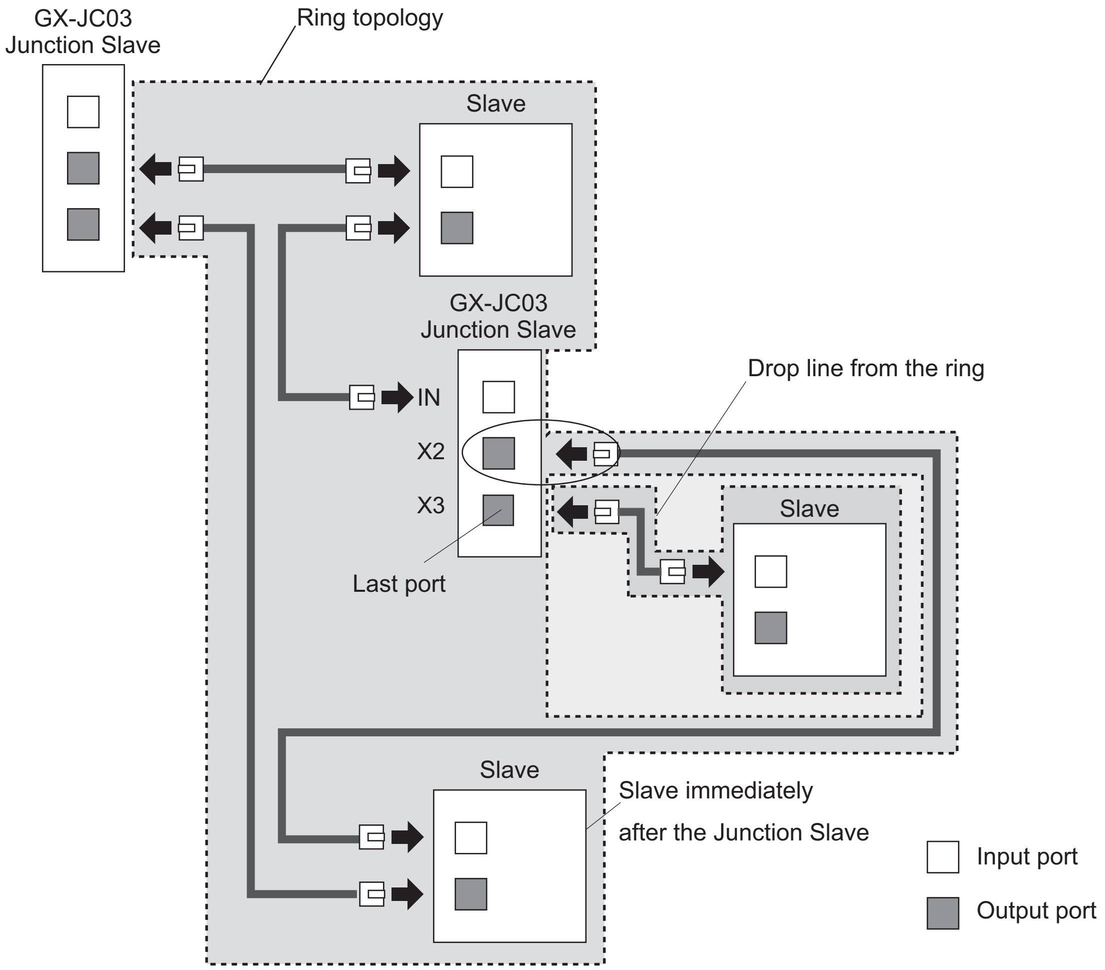
- Cause 3 of incorrect wiring: The wiring for the start port of the ring and the wiring for the end port of the ring are reversed.
  Check that the wiring for the start port of the ring and the wiring for the end port of the ring are not replaced each other.

  In the example below, reconnect the cables that are currently connected to X2 and X3 of the Junction Slave to X3 and X2, respectively.



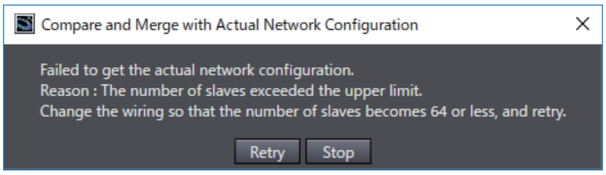
- Cause 4 of incorrect wiring: If a Junction Slave is connected in a ring topology, the output port to which the slave immediately after the Junction Slave is connected is not the last port of the Junction Slave.
  Review the wiring so that the last port is used for the output port of the Junction Slave in the ring topology.

In the example below, the cable that is currently connected to X2 is changed to connect to X3 because the last port of the GX-JC03 Junction Slave in the ring topology is X3. Connect the cable that is currently connected to X3 for the drop line from the ring, to X2.



When More Than the Maximum Number of Slaves Are Connected

If more than the maximum number of slaves are connected in the actual network configuration, the following error message is displayed. Review the network configuration so that the maximum number of slaves is not exceeded. Refer to *1-3 Specifications of Built-in EtherCAT Port* on page 1-10 for the maximum number of slaves.

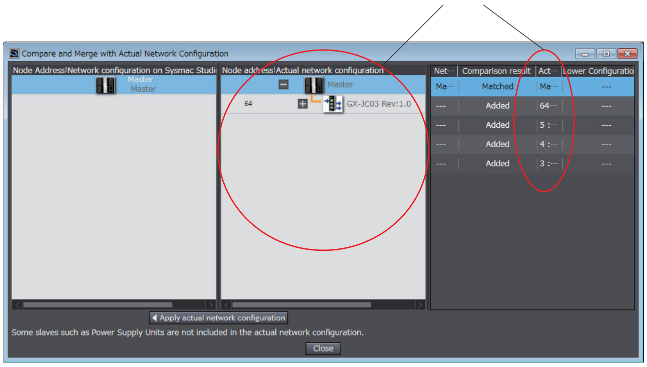

Compare and Merge with Actual Network Configuration

Failed to get the actual network configuration.
Reason : The number of slaves exceeded the upper limit.
Change the wiring so that the number of slaves becomes 64 or less, and retry.

5-6-4 When Slaves on Actual Network Configuration Are Not Displayed

After you perform the **Compare and Merge with Actual Network Configuration**, slaves that are actually connected may not be displayed in the Actual network configuration display area in the **Compare and Merge with Actual Network Configuration** Dialog Box shown below.

Actual network configuration display area



Compare and Merge with Actual Network Configuration

Node Address	Network configuration on Sysmac Studio
	Master Master

Node address	Actual network configuration
	Master
64	GX-JC03 Rev:1.0

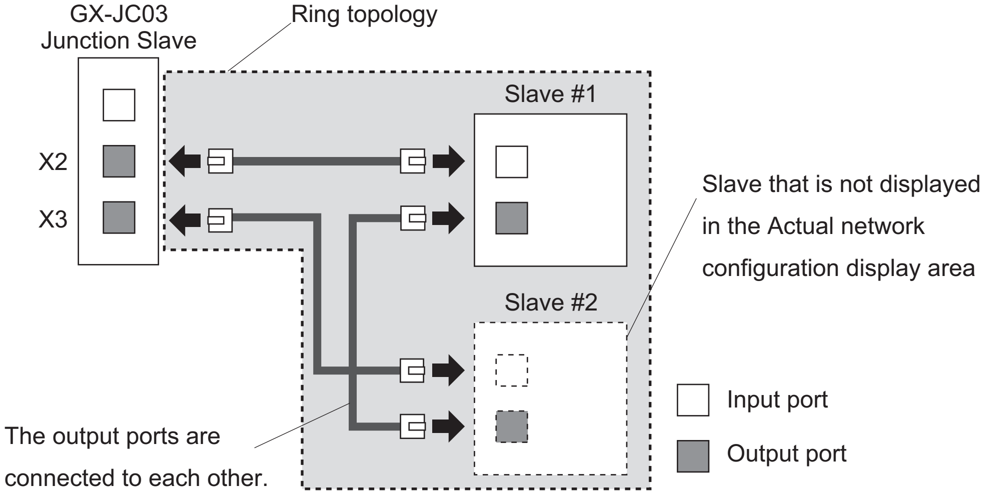
Net···	Comparison result	Act···	Lower Configuratio
Ma···	Matched	Ma···	---
---	Added	64···	---
---	Added	5 :···	---
---	Added	4 :···	---
---	Added	3 :···	---

Some slaves such as Power Supply Units are not included in the actual network configuration.

The cause of which the slaves are not displayed and its collection are described as follows.

- Cause: Output ports are connected to each other between slaves in the ring topology.
  Review the wiring between slaves in the ring topology so that output ports are not connected to each other.

  In the example below, Slave #2 is not displayed in the Actual network configuration display area because output ports are connected to each other between Slave #1 and Slave #2. For the correction, the connections for the input port and output port of the Slave #2 are replaced.

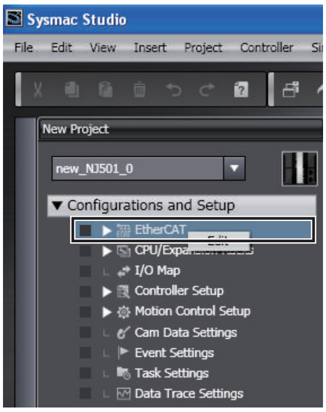

GX-JC03
Junction Slave
Ring topology
X2
X3
Slave #1
Slave #2
Slave that is not displayed in the Actual network configuration display area
The output ports are connected to each other.
Input port
Output port

5-7 Automatically Creating the Network Configuration

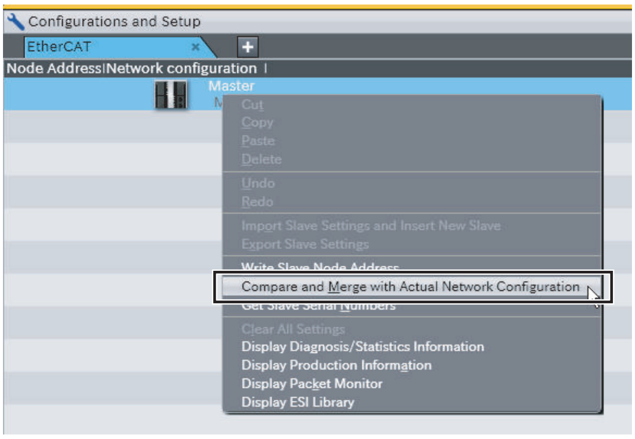
Instead of manually setting the network configuration offline, you can also automatically create the network configuration on the Sysmac Studio based on the actual network configuration.
Use the following procedure to automatically duplicate the network configuration on the Sysmac Studio.

1 Start the Sysmac Studio and go online with the Controller.

2 Double-click **EtherCAT** under **Configurations and Setup** on the Multiview Explorer. Or, right-click **EtherCAT** under **Configurations and Setup** and select **Edit**.

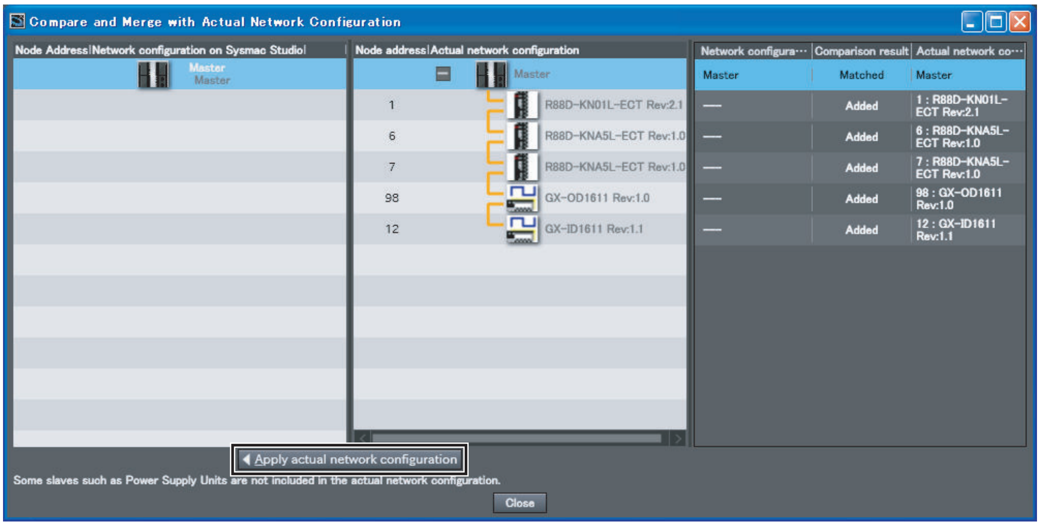


3 Right-click the EtherCAT master that is displayed in the EtherCAT Tab Page and select **Compare and Merge with Actual Network Configuration**.


Configurations and Setup
EtherCAT
Node Address | Network configuration
Master
Cut
Copy
Paste
Delete
Undo
Redo
Import Slave Settings and Insert New Slave
Export Slave Settings
Write Slave Node Address
Compare and Merge with Actual Network Configuration
Get Slave Serial Numbers
Clear All Settings
Display Diagnosis/Statistics Information
Display Production Information
Display Packet Monitor
Display ESI Library

The **Compare and Merge with Actual Network Configuration** Dialog Box is displayed.

4 Click the **Apply actual network configuration** Button.


Compare and Merge with Actual Network Configuration
Apply actual network configuration
Close

Node Address	Network configuration on Sysmac Studio
	Master Master

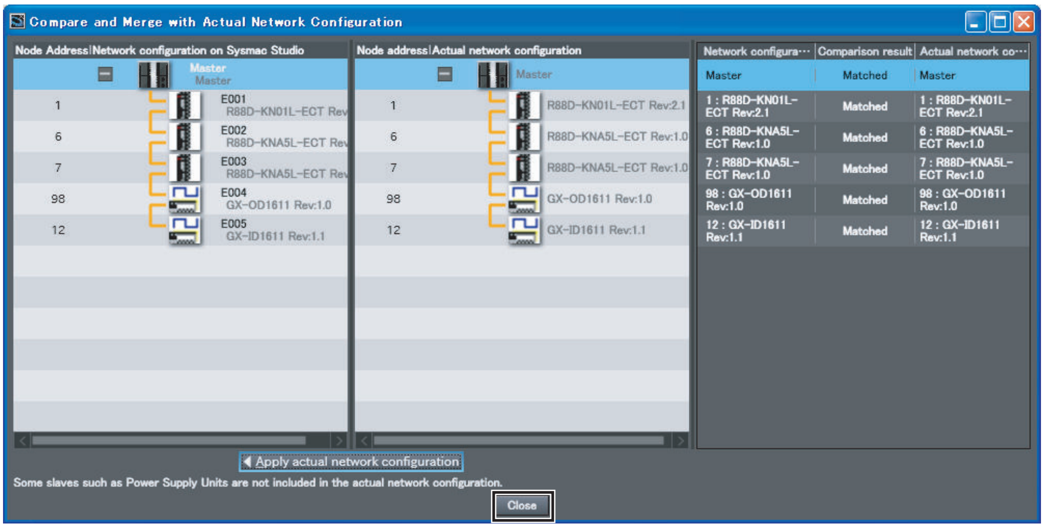
Node address	Actual network configuration
	Master
1	R88D-KN01L-ECT Rev:2.1
6	R88D-KNA5L-ECT Rev:1.0
7	R88D-KNA5L-ECT Rev:1.0
98	GX-OD1611 Rev:1.0
12	GX-ID1611 Rev:1.1

Network configura···	Comparison result	Actual network co···
Master	Matched	Master
—	Added	1 : R88D-KN01L-ECT Rev:2.1
—	Added	6 : R88D-KNA5L-ECT Rev:1.0
—	Added	7 : R88D-KNA5L-ECT Rev:1.0
—	Added	98 : GX-OD1611 Rev:1.0
—	Added	12 : GX-ID1611 Rev:1.1

Some slaves such as Power Supply Units are not included in the actual network configuration.

The configuration in the **Actual network configuration** is duplicated in the **Network configuration on Sysmac Studio**, and the **Comparison results** Column shows that everything in the configurations matches. (The network configuration on the Sysmac Studio is created based on the actual network configuration.)

5 Click the **Close** Button.


Compare and Merge with Actual Network Configuration
Apply actual network configuration
Close

Node Address	Network configuration on Sysmac Studio
	Master Master
1	E001 R88D-KN01L-ECT Rev
6	E002 R88D-KNA5L-ECT Rev
7	E003 R88D-KNA5L-ECT Rev
98	E004 GX-OD1611 Rev:1.0
12	E005 GX-ID1611 Rev:1.1

Node address	Actual network configuration
	Master
1	R88D-KN01L-ECT Rev:2.1
6	R88D-KNA5L-ECT Rev:1.0
7	R88D-KNA5L-ECT Rev:1.0
98	GX-OD1611 Rev:1.0
12	GX-ID1611 Rev:1.1

Network configura···	Comparison result	Actual network co···
Master	Matched	Master
1 : R88D-KN01L-ECT Rev:2.1	Matched	1 : R88D-KN01L-ECT Rev:2.1
6 : R88D-KNA5L-ECT Rev:1.0	Matched	6 : R88D-KNA5L-ECT Rev:1.0
7 : R88D-KNA5L-ECT Rev:1.0	Matched	7 : R88D-KNA5L-ECT Rev:1.0
98 : GX-OD1611 Rev:1.0	Matched	98 : GX-OD1611 Rev:1.0
12 : GX-ID1611 Rev:1.1	Matched	12 : GX-ID1611 Rev:1.1

Some slaves such as Power Supply Units are not included in the actual network configuration.

6 Go offline, and then assign device variables, set the axis parameters, set up the tasks, and set the master and slave parameters.

Additional Information

- In the following cases, the **Compare and Merge with Actual Network Configuration** Dialog Box is not displayed. Refer to *5-6-3 When* ***Compare and Merge with Actual Network Configuration*** *Dialog Box Is Not Displayed* on page 5-32 to remove the cause of the error, and perform the compare and merge operation.
  a) The slave node address settings are not correct on the actual network.
  b) The communications cable is not wired correctly.
  c) More than the maximum number of slaves are connected.

  In addition, if an *EtherCAT Frame Not Received* (842E0000 hex) event occurs, refer to the *NJ/NX-series Troubleshooting Manual (Cat. No. W503)* to remove the cause of the error, and perform the compare and merge operation.
- After you execute the **Compare and Merge with Actual Network Configuration** and even several tens of seconds has passed, if slaves that are connected on the actual network configuration are not displayed in the **Compare and Merge with Actual Network Configuration** Dialog Box, refer to *5-6-4 When Slaves on Actual Network Configuration Are Not Displayed* on page 5-39 to remove the cause of the error, and perform the compare and merge operation.
- Even if you have previously set them on the Sysmac Studio, the following configuration and settings are discarded when you automatically create the network configuration on the Sysmac Studio based on the actual network configuration: network configuration, master settings, and any slave settings (including disable settings, PDO map settings, setting parameter settings, backup parameter settings, device variable assignments in the I/O map, slave assignments to Axes Variables registered in the axis settings, and task settings to control slaves that are registered in the task settings).

  To merge the actual network configuration information without losing the current settings in the Sysmac Studio, use the compare and merge operation to create the network configuration. Refer to *5-6-2 Operation Procedure for Comparing and Merging EtherCAT Network Configurations* on page 5-28 for information on the compare and merge operation.
- After automatically creating the network configuration, if there is a ring topology in the network configuration on the Sysmac Studio, refer to *Components of the Ring Topology* on page 4-5 to check that the appropriate slaves are connected.
- When the compare and merge operation is executed for the actual network configuration, the synchronization between the Sysmac Studio and the Controller is lost. Synchronize the Sysmac Studio and the Controller before you perform any online operations for the slaves. Refer to the *Sysmac Studio Version 1 Operation Manual (Cat. No. W504)* for information on synchronization.
- For project unit version 1.40 or later, if the synchronization between the Sysmac Studio and the Controller is inconsistent, you may not be able to obtain the module configuration information on MDP-compatible slaves. Refer to *A-3-5 Settings for MDP-compatible Slaves from Other Manufacturers* on page A-27 for information on MDP-compatible slaves.

  If you failed to obtain the module configuration information for MDP-compatible slaves, you can obtain the information by the following steps.
  1. Set the **Module config send method** parameter for the MDP-compatible slaves to **Do not send** and perform synchronous transfer.
  2. Perform the compare and merge operation.

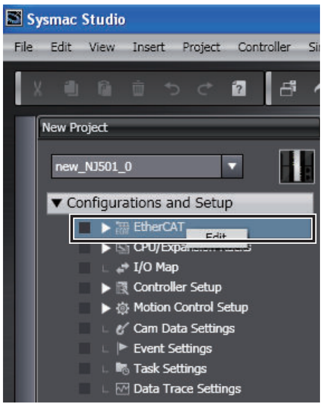
5-8 Using the Sysmac Studio to Obtain Serial Numbers from the Actual Network Configuration

If the **Serial Number Check Method** in the EtherCAT master settings is set to **Setting = Actual device**, you must download the network configuration information in which the slave serial numbers are set to the CPU Unit.

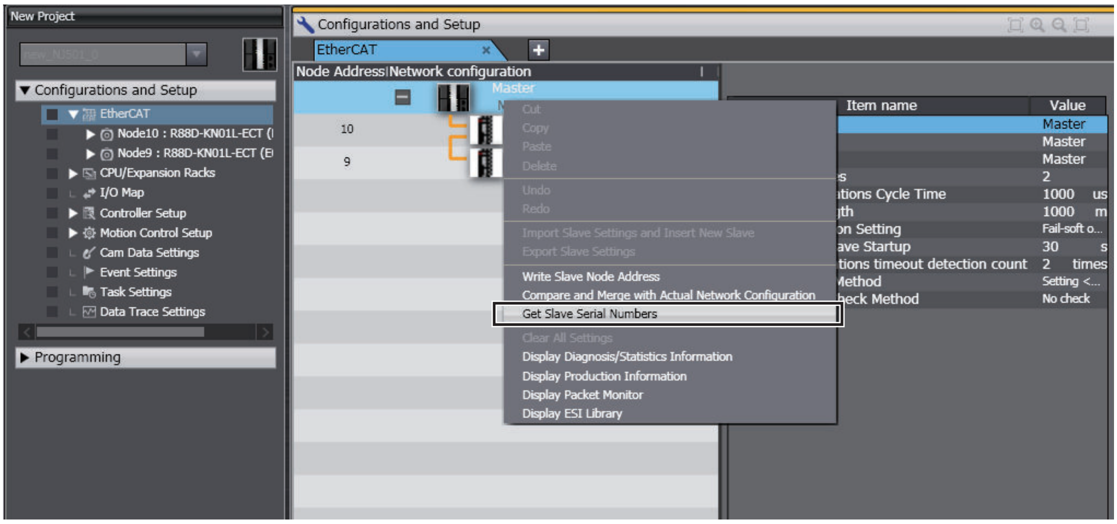
Use the following procedure to get the serial numbers of the physical slaves and apply them as the serial numbers of the slaves in the settings on the Sysmac Studio.

1. Start the Sysmac Studio and go online with the Controller.

2. Double-click **EtherCAT** under **Configurations and Setup** on the Multiview Explorer. Or, right-click **EtherCAT** under **Configurations and Setup** and select **Edit**.



3. Right-click the EtherCAT master that is displayed in the EtherCAT Tab Page and select **Get Slave Serial Numbers**.



The serial numbers of the physical slaves are saved as the serial numbers in the slave settings on the Sysmac Studio.

Additional Information

- Make sure that the communications cables between the master and slaves are connected correctly before you perform this operation. You cannot get the serial numbers of the slaves unless the connections are correct.
- You cannot get the serial numbers of the slaves if there is a slave in the actual network configuration for which the node address is not set or if the same address is set for more than one slave in the actual network configuration. Make sure that node addresses are set correctly for the slaves in the actual network configuration before you perform this operation.
- You cannot get the serial numbers of the slaves unless the **Compare and Merge with Actual Network Configuration** Dialog Box shows that the entire slave configurations agree. Make sure that the **Compare and Merge with Actual Network Configuration** Dialog Box shows that the entire slave configurations agree before you perform this operation.
  However, you can get the serial numbers of the slaves even if the entire slave configurations do not agree if there are slaves in the Sysmac Studio settings that are disabled.
- If you get the serial numbers of the slaves when there are disabled slaves that do not exist in the actual network configuration, the serial numbers of the disabled slaves will be cleared to 0x00000000 in the settings on the Sysmac Studio.
  If the disabled slaves exist in the actual network configuration, the serial numbers of the slaves in the actual network configuration are saved in the settings on the Sysmac Studio.
- Any serial numbers that are set for slaves in the settings on the Sysmac Studio are overwritten when the serial numbers are obtained from the actual network configuration. Make sure that it is OK to overwrite the serial numbers on the Sysmac Studio before you perform this operation.
- When you get the serial numbers from the slaves on the actual network configuration, the synchronization between the Sysmac Studio and the Controller is lost. Synchronize the Sysmac Studio and Controller before you perform any online operations for the slaves. Refer to the *Sysmac Studio Version 1 Operation Manual (Cat. No. W504)* for information on synchronization.

5-9 Downloading the Network Configuration Information from the Sysmac Studio

You must download the network configuration information in the project from the Sysmac Studio to the NJ/NX-series CPU Unit.

Use the synchronize operation to download the network configuration information.

Refer to the *Sysmac Studio Version 1 Operation Manual (Cat. No. W504)* for information on synchronization.

Precautions for Safe Use

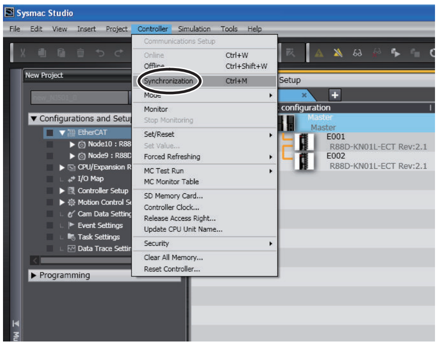
After you transfer the user program, the CPU Unit is restarted and communications with the EtherCAT slaves are cut off. During that period, the slave outputs behave according to the slave specifications. The time that communications are cut off depends on the EtherCAT network configuration.

Before you transfer the user program, confirm that the system will not be adversely affected.

Precautions for Correct Use

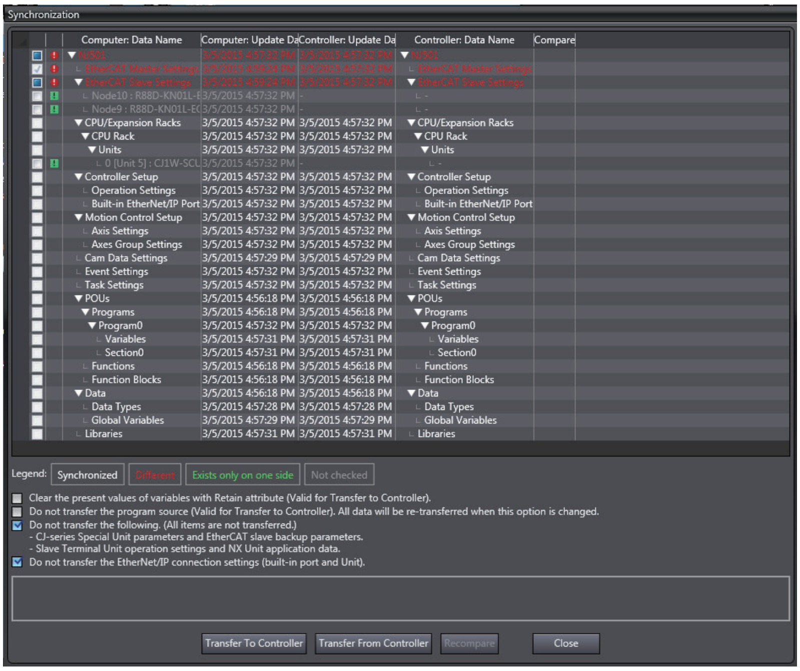
If a *Link OFF Error* (84200000 hex) event occurs, remove the cause of the error and retry downloading the network configuration information after the error is reset.

1 Go online and select **Synchronization** from the **Controller** Menu.



The Synchronization Window is displayed.

2 Click the **Transfer to Controller** Button.



Synchronization

		Computer: Data Name	Computer: Update Da	Controller: Update Da	Controller: Data Name	Compare
■	!	▼ NJ501	3/5/2015 4:57:32 PM	3/5/2015 4:57:32 PM	▼ NJ501	
☑	!	└ EtherCAT Master Settings	3/5/2015 4:59:24 PM	3/5/2015 4:57:32 PM	└ EtherCAT Master Settings	
■	!	▼ EtherCAT Slave Settings	3/5/2015 4:59:24 PM	3/5/2015 4:57:32 PM	▼ EtherCAT Slave Settings	
□	!	└ Node10 : R88D-KN01L-E	3/5/2015 4:57:32 PM	-	└ -	
□	!	└ Node9 : R88D-KN01L-EC	3/5/2015 4:57:32 PM	-	└ -	
□		▼ CPU/Expansion Racks	3/5/2015 4:57:32 PM	3/5/2015 4:57:32 PM	▼ CPU/Expansion Racks	
□		▼ CPU Rack	3/5/2015 4:57:32 PM	3/5/2015 4:57:32 PM	▼ CPU Rack	
□		▼ Units	3/5/2015 4:57:32 PM	3/5/2015 4:57:32 PM	▼ Units	
□	!	└ 0 [Unit 5] : CJ1W-SCU	3/5/2015 4:57:32 PM	-	└ -	
□		▼ Controller Setup	3/5/2015 4:57:32 PM	3/5/2015 4:57:32 PM	▼ Controller Setup	
□		└ Operation Settings	3/5/2015 4:57:32 PM	3/5/2015 4:57:32 PM	└ Operation Settings	
□		└ Built-in EtherNet/IP Port	3/5/2015 4:57:32 PM	3/5/2015 4:57:32 PM	└ Built-in EtherNet/IP Port	
□		▼ Motion Control Setup	3/5/2015 4:57:32 PM	3/5/2015 4:57:32 PM	▼ Motion Control Setup	
□		└ Axis Settings	3/5/2015 4:57:32 PM	3/5/2015 4:57:32 PM	└ Axis Settings	
□		└ Axes Group Settings	3/5/2015 4:57:32 PM	3/5/2015 4:57:32 PM	└ Axes Group Settings	
□		└ Cam Data Settings	3/5/2015 4:57:29 PM	3/5/2015 4:57:29 PM	└ Cam Data Settings	
□		└ Event Settings	3/5/2015 4:57:32 PM	3/5/2015 4:57:32 PM	└ Event Settings	
□		└ Task Settings	3/5/2015 4:57:32 PM	3/5/2015 4:57:32 PM	└ Task Settings	
□		▼ POUs	3/5/2015 4:56:18 PM	3/5/2015 4:56:18 PM	▼ POUs	
□		▼ Programs	3/5/2015 4:56:18 PM	3/5/2015 4:56:18 PM	▼ Programs	
□		▼ Program0	3/5/2015 4:57:32 PM	3/5/2015 4:57:32 PM	▼ Program0	
□		└ Variables	3/5/2015 4:57:31 PM	3/5/2015 4:57:31 PM	└ Variables	
□		└ Section0	3/5/2015 4:57:31 PM	3/5/2015 4:57:31 PM	└ Section0	
□		└ Functions	3/5/2015 4:56:18 PM	3/5/2015 4:56:18 PM	└ Functions	
□		└ Function Blocks	3/5/2015 4:56:18 PM	3/5/2015 4:56:18 PM	└ Function Blocks	
□		▼ Data	3/5/2015 4:56:18 PM	3/5/2015 4:56:18 PM	▼ Data	
□		└ Data Types	3/5/2015 4:57:28 PM	3/5/2015 4:57:28 PM	└ Data Types	
□		└ Global Variables	3/5/2015 4:57:29 PM	3/5/2015 4:57:29 PM	└ Global Variables	
□		└ Libraries	3/5/2015 4:57:31 PM	3/5/2015 4:57:31 PM	└ Libraries	

Legend: Synchronized | Different | Exists only on one side | Not checked

□ Clear the present values of variables with Retain attribute (Valid for Transfer to Controller).

□ Do not transfer the program source (Valid for Transfer to Controller). All data will be re-transferred when this option is changed.

☑ Do not transfer the following. (All items are not transferred.)
- CJ-series Special Unit parameters and EtherCAT slave backup parameters.
- Slave Terminal Unit operation settings and NX Unit application data.

☑ Do not transfer the EtherNet/IP connection settings (built-in port and Unit).

The network configuration information is transferred from the Sysmac Studio to the Controller.

Additional Information

The backup parameters of EtherCAT slaves are out of the scope of synchronization and are not transferred to the slaves for the default settings.
To transfer the backup parameters, clear the selection of the **Do not transfer Special Unit parameters and backup parameters of EtherCAT slaves (out of synchronization scope).** Check Box in the Synchronization Window before you click the **Transfer To Controller** Button.

5-10 Confirming Communications after Completing EtherCAT Configuration and Settings

If normal network configuration information is downloaded to the CPU Unit, EtherCAT communications start automatically regardless of the operating mode of the CPU Unit.
After the start of EtherCAT communications, check the process data communications status to ensure that process data communications are performed normally between the EtherCAT master and all EtherCAT slaves.

Checking the Process Data Communications Status

Use one of the following methods to check the process data communications status.

- Controller status monitor of the Sysmac Studio
- System-defined variable
- Indicator

The contents of checking the process data communications status using these methods are given below.

● Controller Status Monitor of the Sysmac Studio

On the detailed view of the Controller status monitor, check the item for **EtherCAT Process Data Communications**. If the item for **EtherCAT Process Data Communications** is *Communicating*, process data communications are performed normally. Refer to the *Sysmac Studio Version 1 Operation Manual (Cat. No. W504)* for how to display the Controller status monitor of the Sysmac Studio.

● System-defined Variable

Check the *_EC_PDActive* (Process Data Communications Status) system-defined variable in the user program. If *_EC_PDActive* is TRUE, process data communications are performed normally.

● Indicator

EtherCAT communications are in Operational state (process data communications are performed normally) if the EtherCAT NET RUN indicator on the front of the NJ/NX-series CPU Unit is lit green.

Checking and Correction Methods if Process Data Communications are not Performed Normally

If process data communications are not performed normally, check the event log in the Sysmac Studio and take suitable measures. Refer to *Error Tables* in *Errors in the EtherCAT Master Function Module* in the *NJ/NX-series Troubleshooting Manual (Cat. No. W503)* for event logs.

Precautions for Safe Use

- EtherCAT communications are not always established immediately after the power supply is turned ON. Use the system-defined variable in the user program to confirm that communications are established before you attempt to control device variables and Axis Variables.
- After you change any EtherCAT slave or Special Unit settings, carefully check the safety of the controlled system before you restart the Unit.

Precautions for Correct Use

- When you change the EtherCAT slave configuration and reconfigure the axis allocation settings, you must download the network configuration information again.
- If network configuration information is not downloaded to the Controller, the EtherCAT master will not perform process data communications. Because of this, notification of errors will not occur. The EtherCAT communications will be in the Init state (where both process data communications and SDO communications are disabled).

Checking the EtherCAT Network Line Quality

If you need to confirm that the network was correctly installed after configuring an EtherCAT network, diagnose the EtherCAT network line quality. For how to diagnose the EtherCAT network line quality, refer to *9-2-4 Diagnostic and Statistical Information* on page 9-3.

6

Process Data Communications and SDO Communications

This section describes the timing of communications, response times, and special instructions for process data communications and SDO communications. It also provides sample programming.

6-1 Process Data Communications (PDO Communications)

Process data communications cyclically exchanges data between the master and slaves in the process data communications cycle (i.e., the task period of primary periodic task or priority-5 periodic task). From the user program in the NJ/NX-series CPU Unit, slave data is accessed through allocated variables.

Additional Information

You can use the priority-5 periodic task only with NX701 CPU Units.

6-1-1 Allocated Variables for Process Data Communications

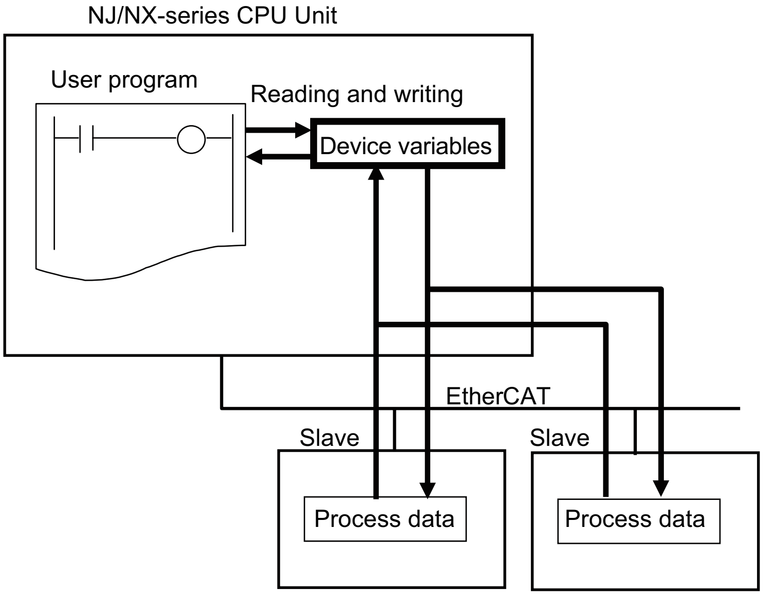
The variables that are allocated depend on the slave type as shown in the following table.

Slave type	Allocated variables	Operation screen in Sysmac Studio
EtherCAT slaves which are not assigned to axes	Device Variables	I/O Map
EtherCAT slaves which are assigned to axes	Axis Variables	Axis Settings

Allocated variables are specified for parameters of instructions in the user program.

Device Variables

The user program in an NJ/NX-series CPU Unit can read and write EtherCAT slave process data that is not assigned to axes through device variables. (The slave process data is I/O data that is exchanged between the master and slaves in each period.)

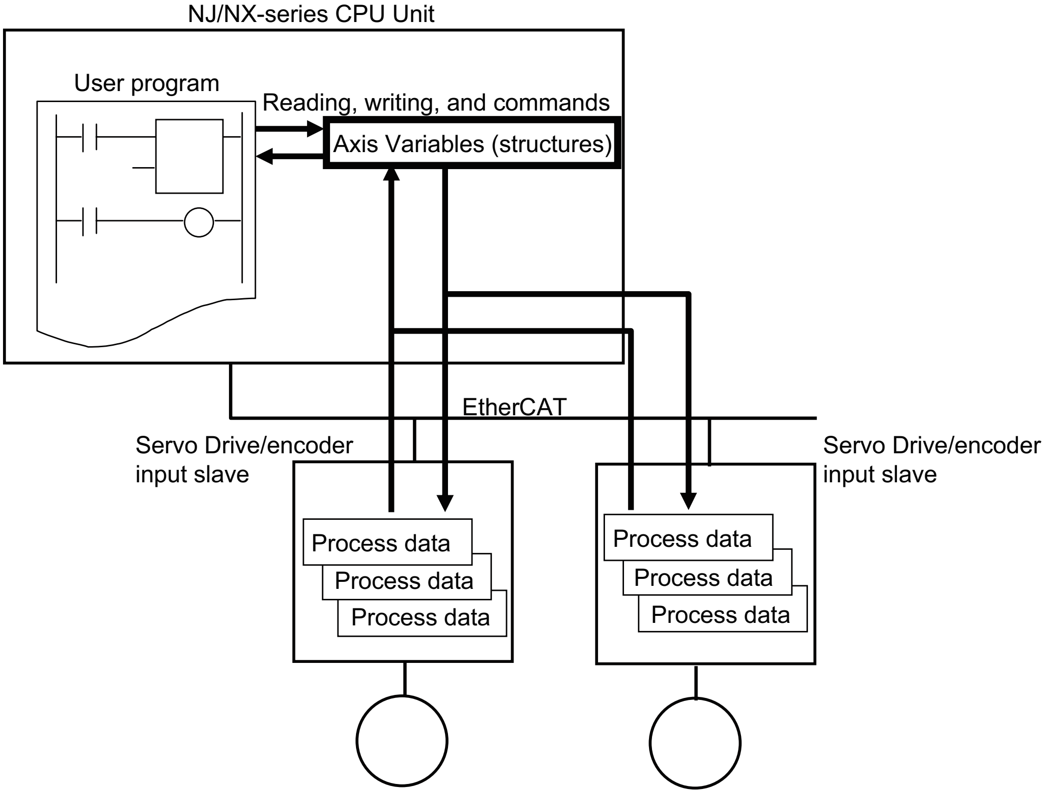


Axis Variables

Servo Drives and encoder input slaves that are assigned to axes on EtherCAT can be controlled by specifying Axis Variables (structures) as parameters for motion control instruction in-out variables.

Axis Variables (structures) consist of the following data.

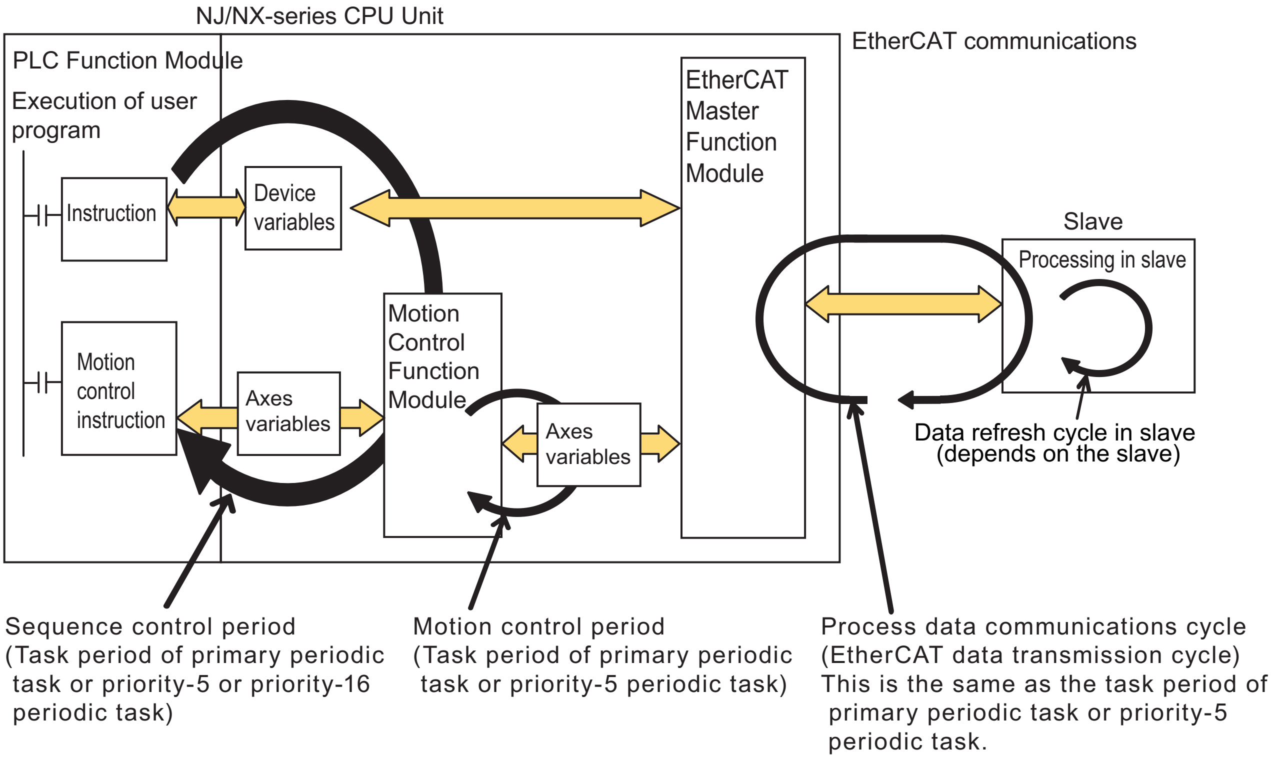
- Parameters: Maximum acceleration rate, software limits, etc.
- I/O: Home proximity input, Servo ON, etc.
- Operating status: Accelerating, waiting for in-position state, etc.
- Error and warning status: Excessive following error, acceleration error, etc.


NJ/NX-series CPU Unit
User program
Reading, writing, and commands
Axis Variables (structures)
EtherCAT
Servo Drive/encoder input slave
Process data
Process data
Process data
Servo Drive/encoder input slave
Process data
Process data
Process data

6-1-2 Process Data Communications Refresh Timing

The process data communications cycle is the same as the task period of the primary periodic task or the priority-5 periodic task for sequence control. It is also the same as the motion control period.

The control periods between each function module and between the function module and the slave are shown as follows.


NJ/NX-series CPU Unit
PLC Function Module
Execution of user program
Instruction
Device variables
Motion control instruction
Axes variables
Motion Control Function Module
Axes variables
EtherCAT Master Function Module
EtherCAT communications
Slave
Processing in slave
Data refresh cycle in slave (depends on the slave)
Sequence control period (Task period of primary periodic task or priority-5 or priority-16 periodic task)
Motion control period (Task period of primary periodic task or priority-5 periodic task)
Process data communications cycle (EtherCAT data transmission cycle) This is the same as the task period of primary periodic task or priority-5 periodic task.

Task period of primary periodic task or priority-5 periodic task = Motion control period = Process data communications cycle

Precautions for Correct Use

- You can use the priority-5 periodic task only with NX701 CPU Units.
- With an NX701 CPU Unit, you can perform process data communications separately in each task period of the primary periodic task and the priority-5 periodic task. If these two process data communications cycles must be distinguished, the cycle for the primary periodic task is called process data communications cycle 1 and the cycle for the priority-5 periodic task is called process data communications cycle 2.
- With an NX701 CPU Unit, you can execute motion control in the primary periodic task and in the priority-5 periodic task. If these two motion controls must be distinguished, the one executed by the primary periodic task is called motion control 1 and the other executed by the priority-5 periodic task is called motion control 2.
- The NX502 CPU Units, NX102 CPU Units, NX1P2 CPU Units, and NJ-series CPU Units perform process data communications only in the primary periodic task.

Relationship Between Process Data Communications Cycle and Task Setting

The process data communications cycles of the primary periodic task and priority-5 periodic task are synchronized with the task period of the task to which I/O refreshing is assigned for each EtherCAT slave.

Assigning I/O refreshing to tasks, setting procedure on Sysmac Studio, and I/O refreshing timing are shown as follows.

● Assigning I/O Refreshing to Tasks

I/O refreshing of the EtherCAT slaves is assigned to the tasks.

Tasks to which assignment is possible and unit of assignment are different depending on the I/O refreshing target. Unit of assignment refers to a target or a group of targets for I/O refreshing that can be assigned to one I/O controlling task. For example, when the unit of assignment is Slave Terminal, you can assign I/O refreshing to only one task even if more than one NX Unit is connected to a Communications Coupler Unit.

The following table shows the relationship among the I/O refreshing target, the assignable task, and the unit of assignment.

I/O refreshing target	Assignable task	Unit of assignment
Communications Coupler Unit with an NX Unit assigned to an axis on the Slave Terminal	Primary periodic task or priority-5 periodic task*1	Slave Terminal
Communications Coupler Unit without an NX Unit assigned to an axis on the Slave Terminal	Primary periodic task or priority-5 and priority-16 periodic tasks*1	
EtherCAT slaves to which axes are assigned	Primary periodic task or priority-5 periodic task*1	Slave
Other EtherCAT slaves	Primary periodic task or priority-5 and priority-16 periodic tasks*1	

*1. You can use the priority-5 periodic task only with NX701 CPU Units.

● Sysmac Studio Setting Procedure

For the slaves and Units that are not assigned to axes, set the tasks in which to perform I/O refreshing in **I/O Control Task Settings** under **Configuration and Setup** - **Task Settings** of the Sysmac Studio.

Refer to the *NJ/NX-series CPU Unit Software User's Manual (Cat. No. W501)* for details.

For the slaves and Units that are assigned to axes, specify the motion controls to use in **Motion Control Setup** under **Confugurations and Setup** of the Sysmac Studio. The tasks to perform I/O refreshing are set.

Refer to the *NJ/NX-series CPU Unit Motion Control User’s Manual (Cat. No. W507)* for details.

● Timing of I/O Refreshing

The table below shows when I/O is refreshed for each type of EtherCAT slaves.

Task that controls I/O	Execution period of I/O refreshing
Primary periodic task	Task period of the primary periodic task*1
Priority-5 periodic task*2	Task period of the priority-5 periodic task*1
Priority-16 periodic task	Task period of the primary periodic task*1*3

*1. Refer to relevant manuals for each slave to check if the I/O refreshing period agrees with the communications cycle supported with the slave. An error occurs if you use the Sysmac Studio to set the I/O refreshing timing to a communications cycle that is not supported with the slave. The error occurs only with the OMRON synced slaves.

*2. You can use the priority-5 periodic task only with NX701 CPU Units.

*3. EtherCAT communications is executed during I/O refreshing in the primary periodic task. If the priority-16 periodic task is used to control EtherCAT slaves, data will be refreshed by I/O refreshing in the task period of the priority-16 periodic task.

Relationship between tasks to execute I/O refreshing, process data communications cycle, and motion control is shown as follows.

- NX701 CPU Units

Task to execute I/O refreshing	Process data communications cycle	Motion control
Primary periodic task	Process data communications cycle 1	Motion control 1
Priority-5 periodic task	Process data communications cycle 2	Motion control 2

- NX502, NX102, and NX1P2 CPU Units, and NJ-series CPU Units

Task to execute I/O refreshing	Process data communications cycle	Motion control
Primary periodic task	Process data communications cycle	Motion control

● Restrictions for Multiple Communications Cycles

With an NX701 CPU Unit, you can perform process data communications separately in each task period of the primary periodic task and the priority-5 periodic task. On the other hand, there is only one EtherCAT network line. Therefore, some restrictions are placed on the process data communications in multiple communications cycles, as shown below.

- When the number of slaves controlled by the master increases, the process data communications cycles of both primary periodic task and priority-5 periodic task will get longer.
  Consider the following countermeasures if the process data communications cycle for each task exceeds the task period.
  - If you want to maintain the number of the slaves, increase the task period of the task whose process data communications cycle exceeds the task period.
  - If you want to maintain the task period, decrease the number of the slaves.
- The process data communications for the primary periodic task is processed in priority over the process data communications for the priority-5 periodic task. Therefore, the size of process data for the priority-5 periodic task that can be processed becomes smaller than that for the primary periodic task even if both tasks have the same task period.
  Consider the following countermeasures if the process data communications cycle for the priority-5 periodic task exceeds the task period.
  - If you want to maintain the process data size of priority-5 periodic task, increase the period of the priority-5 periodic task.
  - If you want to maintain the period of priority-5 periodic task, decrease the process data size of the priority-5 periodic tasks that are assigned for the entire slaves.

6-1-3 Checking Validity of Process Data

When reading and writing the device variables and axes variables that input and output process data with slaves, the validity of the process data must be checked.

When EtherCAT communications are not established, such as immediately after the power supply to the CPU Unit is turned ON, the process data is invalid. Thus values become invalid for the device variables and axes variables that input and output the process data. If EtherCAT communications were established previously, the device variables and axes variables retain the values updated while EtherCAT commutations were established then.

Once the EtherCAT communications are established, the process data becomes valid, therefore values of the device variables and axes variables that input and output the process data become also valid. Be sure to read and write the device variables and axes variables that input and output the process data, after confirming that the process data is valid.

Use the following system-defined variables to check the validity of the process data.

- Checking the validity of the process data for each slave.
  _EC_PDSlavTbl (Process Data Communicating Slave Table)
- Checking the validity of the process data for the entire slaves.
  _EC_PDActive (Process Data Communications Status)

Precautions for Correct Use

If noise occurs or an EtherCAT slave is disconnected from the network, any current communications frames may be lost. If frames are lost, slave I/O data is not communicated, and unintended operation may occur. The slave outputs behave according to the slave specifications. For details, refer to relevant manuals for each slave. If a noise countermeasure or slave replacement is required, perform the following processing.

- Program the *_EC_InDataInvalid* (Input Data Invalid), *_EC_InData1Invalid* (Input Data1 Invalid), or *_EC_InData2Invalid* (Input Data2 Invalid) system-defined variable as an interlock condition in the user program.
- Set the **PDO communications timeout detection count** setting in the EtherCAT master to at least 2. Refer to *5-4 EtherCAT Master Parameter Settings* on page 5-14 for the setting procedure.

A sample programming to check the validity of the process data is shown in the next section.

Sample Programming to Check Validity of Process Data for Each Slave

With this sample program, validity of the process data is checked individually for each slave, and the data inputs and outputs with the salve are controlled.
The slave with which the process data is input and output is given the node address 1 and named "slave 1".
The *_EC_PDSlavTbl* (Process Data Communicating Slave Table) system-defined variable is used to see if the process data inputs and outputs are valid for the slave 1.

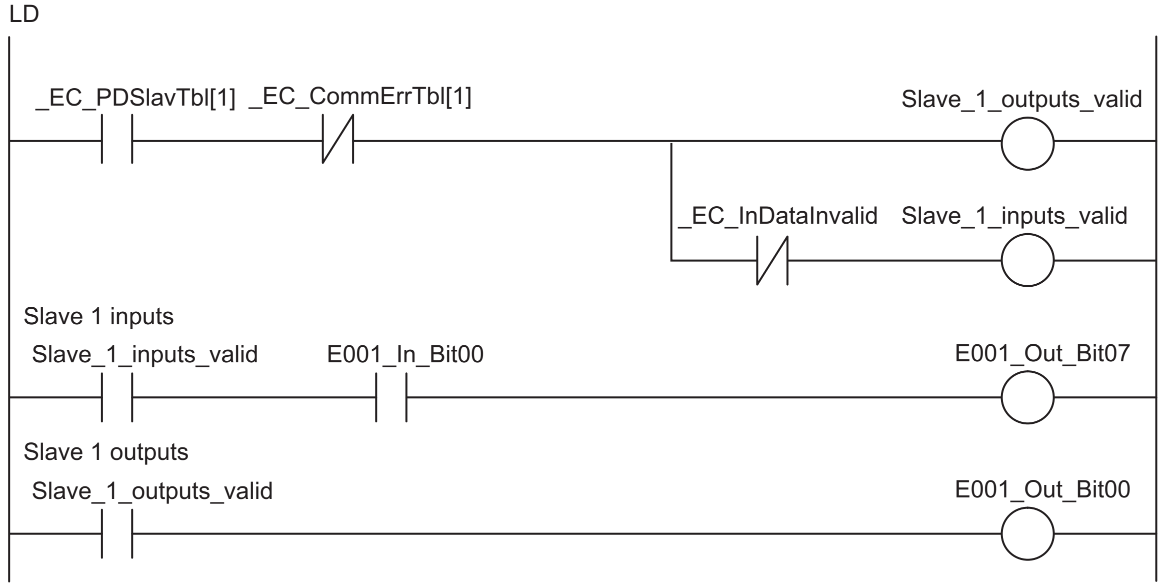
If the process data inputs and outputs with the slave 1 are valid, *E001_Out_Bit07* changes to TRUE when *E001_In_Bit00* changes to TRUE. Also, when the process data outputs with the slave 1 are valid, *E001_Out_Bit00* changes to TRUE.

Precautions for Correct Use

If you execute the instructions below, the *_EC_InDataInvalid* (Input Data Invalid), *_EC_InData1Invalid* (Input Data1 Invalid), and *_EC_InData2Invalid* (Input Data2 Invalid) system-defined variables will temporarily change to TRUE depending on the task to which the specified EtherCAT slave is assigned.

- EC_DisconnectSlave (Disconnect EtherCAT Slave)
- EC_ConnectSlave (Connect EtherCAT Slave)
- EC_ChangeEnableSetting (Enable/Disable EtherCAT Slave)

● LD



● ST

```
IF _EC_PDSlavTbl[1]=TRUE AND _EC_CommErrTbl[1]=FALSE THEN
    A:=TRUE;
ELSE
    A:=FALSE;
END_IF;

(*Output valid condition*)
IF A=TRUE THEN
    Slav_Out:=TRUE;
ELSE
    Slav_Out:=FALSE;
END_IF;

(*Input valid condition*)
IF A=TRUE AND _EC_InDataInvalid=FALSE THEN
    Slav_In:=TRUE;
ELSE
    Slav_In:=FALSE;
END_IF;

(*Output data valid*)
IF Slav_Out=TRUE THEN
    E001_Out_Bit00:=TRUE;
ELSE
    E001_Out_Bit00:=FALSE;
END_IF;
```

```
(*Input data valid*)
IF Slav_In=TRUE AND E001_In_Bit00=TRUE  THEN
    E001_Out_Bit07:=TRUE;
ELSE
    E001_Out_Bit07:=FALSE;
END_IF;
```

Additional Information

You can read the status of the *_EC_PDSlavTbl* (Process Data Communicating Slave Table) system-defined variable from the user program to see if I/O refreshing is normal.

Sample Programming to Check Validity of Process Data for Entire Slaves

With this sample program, validity of process data is checked collectively for all slaves that are connected to the network, and the process data inputs and outputs with the slaves are controlled.
The slave with which the process data is input and output is given the node address 1.
The *_EC_PDActive* (Process Data Communications Status) system-defined variable is used to see if the process data inputs and outputs are valid for all of the slaves.

If the process data inputs and outputs with all slaves are valid, *E001_Out_Bit07* changes to TRUE when *E001_In_Bit00* changes to TRUE. Also, when the process data outputs with all slaves are valid, *E001_Out_Bit00* changes to TRUE.

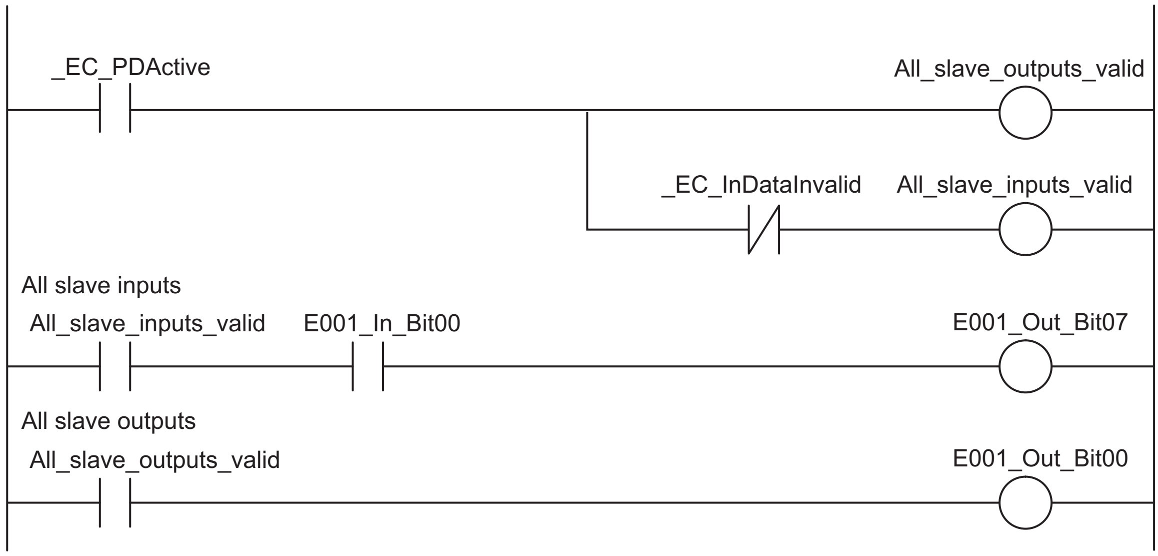
Precautions for Correct Use

If you execute the instructions below, the *_EC_InDataInvalid* (Input Data Invalid), *_EC_InData1Invalid* (Input Data1 Invalid), and *_EC_InData2Invalid* (Input Data2 Invalid) system-defined variables will temporarily change to TRUE depending on the task to which the specified EtherCAT slave is assigned.

- EC_DisconnectSlave (Disconnect EtherCAT Slave)
- EC_ConnectSlave (Connect EtherCAT Slave)
- EC_ChangeEnableSetting (Enable/Disable EtherCAT Slave)

● LD

LD



● ST

```
IF _EC_PDActive=TRUE THEN
    A:=TRUE;
ELSE
    A:=FALSE;
END_IF;

(*Output valid condition*)
IF A=TRUE THEN
    AllSlav_Out:=TRUE;
ELSE
    AllSlav_Out:=FALSE;
END_IF;

(*Input valid condition*)
IF A=TRUE AND _EC_InDataInvalid=FALSE THEN
    AllSlav_In:=TRUE;
ELSE
    AllSlav_In:=FALSE;
END_IF;

(*Output data valid*)
IF ALLSlav_Out=TRUE THEN
    E001_Out_Bit00:=TRUE;
ELSE
    E001_Out_Bit00:=FALSE;
```

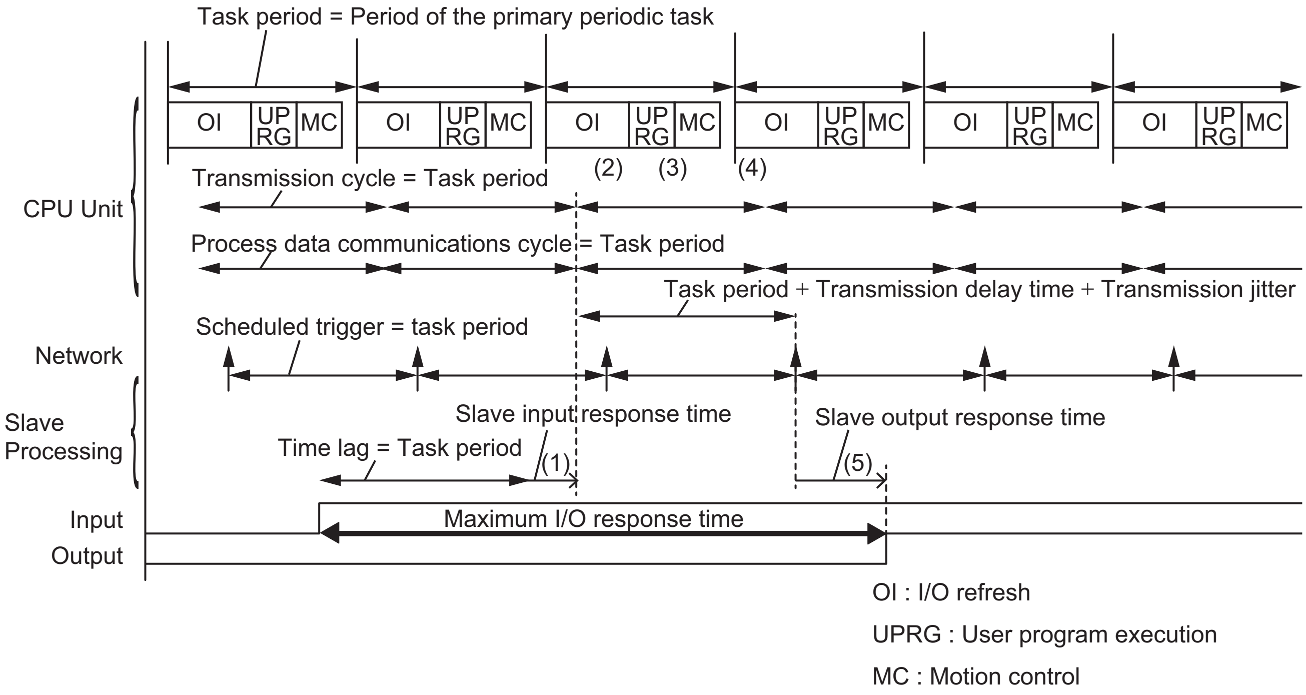
```
END_IF;

(*Input data valid*)
IF ALLSlav_In=TRUE AND E001_In_Bit00=TRUE THEN
    E001_Out_Bit07:=TRUE;
ELSE
    E001_Out_Bit07:=FALSE;
END_IF;
```

6-1-4 System Response Time in Process Data Communications

This section defines the maximum I/O response time of process data communications in the system.

Standard Synchronization Timing

The following timing chart shows when sequence control and motion control are performed within the task period of the primary periodic task in which EtherCAT communications were refreshed.


Task period = Period of the primary periodic task
OI
UPRG
MC
(2)
(3)
(4)
CPU Unit
Transmission cycle = Task period
Process data communications cycle = Task period
Task period + Transmission delay time + Transmission jitter
Scheduled trigger = task period
Network
Slave Processing
Slave input response time
Slave output response time
Time lag = Task period
(1)
(5)
Input
Output
Maximum I/O response time
OI : I/O refresh
UPRG : User program execution
MC : Motion control

1. The slave generates process data send frames with I/O information from sensors and other devices.
2. The EtherCAT master receives the process data frames that was sent from the slave through I/O refreshing for the CPU Unit, and reads it as input data in the CPU Unit.
3. Based on the input data, the CPU Unit processes user program execution and motion control to determine the output data.
4. The EtherCAT master generates a send frame of the process data with the output data produced in step 3, and sends the frame to the slave before the slave performs reception processing.

5. The slave receives the send frame of the process data from the EtherCAT master in the scheduled trigger timing, and outputs data to relays and other output devices.

Calculation Formula for Maximum I/O Response Time

The maximum I/O response time is calculated for each periodic task of the CPU Unit in which I/O is refreshed for EtherCAT slaves.

The formula for calculating the maximum I/O response time is as follows:

Maximum I/O response time =	Task period of periodic task (= process data communications cycle) × 2 + Slave input response time + Slave output response time + Transmission delay time + Transmission jitter

Precautions for Correct Use

The values that are found in the above formula are guidelines instead of guaranteed values. You must use a physical device to check the performance before the actual operation.

The following describes how to find each element value in the formula.

● Task Period of Periodic Task

This indicates the task period of the primary periodic task or the priority-5 periodic task. The priority-5 periodic task is applicable only when you use an NX701 CPU Unit.

● Transmission Delay Time

The following methods are provided to calculate the transmission delay times for the CPU Unit and the periodic tasks in which I/O is refreshed.

Precautions for Correct Use

The calculation of the transmission delay times described below assume that OMRON slaves are used with recommended cables.

CPU Unit	Periodic task of the CPU Unit in which I/O is refreshed	Calculation method
NX701	Primary periodic task	• Calculation with the Sysmac Studio • Calculation by a formula
	Priority-5 periodic task	Calculation with the Sysmac Studio
NX502	Primary periodic task	• Calculation with the Sysmac Studio • Calculation by a formula
NX102 and NX1P2	Primary periodic task	• Calculation with the Sysmac Studio • Calculation by a formula
NJ-series	Primary periodic task	Calculation by a formula

1. Calculation with the Sysmac Studio

You can use the Sysmac Studio to display the transmission delay time for the EtherCAT master. Refer to *Displaying Transmission Delay Time for EtherCAT Master with the Sysmac Studio* on page 6-14 for details on this display method.

2. Calculation by a formula

   You can use the following formula for the calculation by a formula. The value that is found in the formula is a roughly-estimated value.

Transmission delay time (μs) = 1.24[*1] [μs] × Number of EtherCAT slaves + 0.082 [μs] × EtherCAT frame length [byte] + 0.01 [μs] × Total cable length[*2] [m]

[*1] For project unit version 1.40 or later, this value is 1.60.

[*2] When the total cable length is 1,000 m or less, it is calculated as 1,000 m.

The parameters necessary for calculation are described below.

Parameter	Description
Number of EtherCAT Slaves	This is the number of all of the EtherCAT slaves in the same EtherCAT network. An EtherCAT Slave Terminal is counted as one EtherCAT slave.
EtherCAT Frame Length in Bytes	The EtherCAT frame length is the data byte size that is calculated under the following conditions for the EtherCAT slaves in the same EtherCAT network and for which I/O is refreshed in the primary periodic task. • If the total data size is less than 64 bytes, use 64 bytes in the calculation. • For EtherCAT slaves that have both input and output data, use the larger of the input data size and output data size. For example, the EtherCAT frame length in bytes for the following configuration is as given below. Configuration Example • Two OMRON GX-series EtherCAT slaves: Data size of 20 bytes/slave • One EtherCAT Slave Terminal: Input data size of 50 bytes and output data size of 30 bytes EtherCAT frame length = 20 bytes + 20 bytes + 50 bytes = 90 bytes
Total Cable Length	This is the total cable length, including the cable between the EtherCAT master and a slave and cables between slaves.

● Transmission Jitter

The value of transmission jitter depends on the CPU Unit.

CPU Unit	Transmission jitter
NX701	10 μs
NX502	10 μs
NX102	10 μs
NX1P2	10 μs
NJ-series	100 μs

● Slave Input Response Time

This is the time from when a slave reads input information from output devices until the slave transmits a frame as a process data on the EtherCAT communications. This is the time required for internal processing that is specific to each slave. Refer to relevant manuals for each slave.

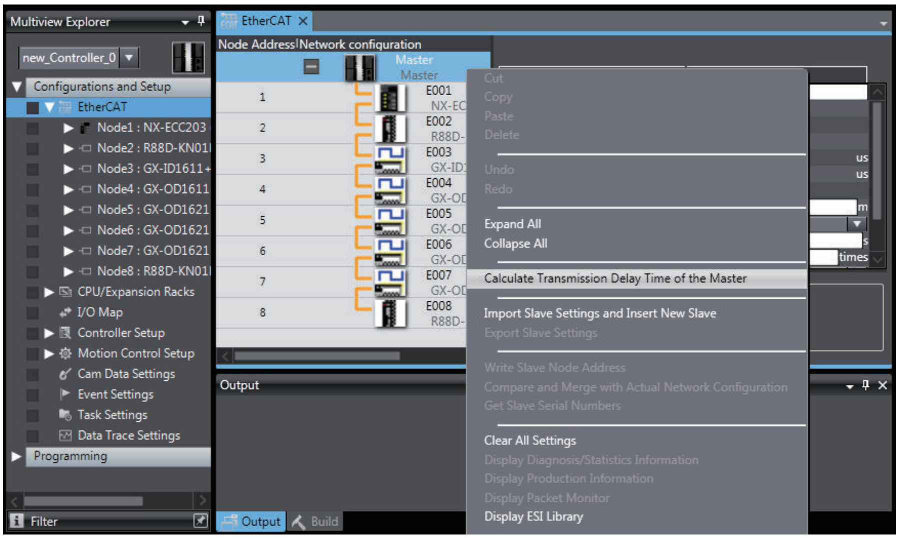
- **Slave Output Response Time**

  This is the time from when a slave receives a frame as a process data on the EtherCAT communications until the slave outputs the frame as output data to external output devices. This is the time required for internal processing that is specific to each slave. Refer to relevant manuals for each slave.

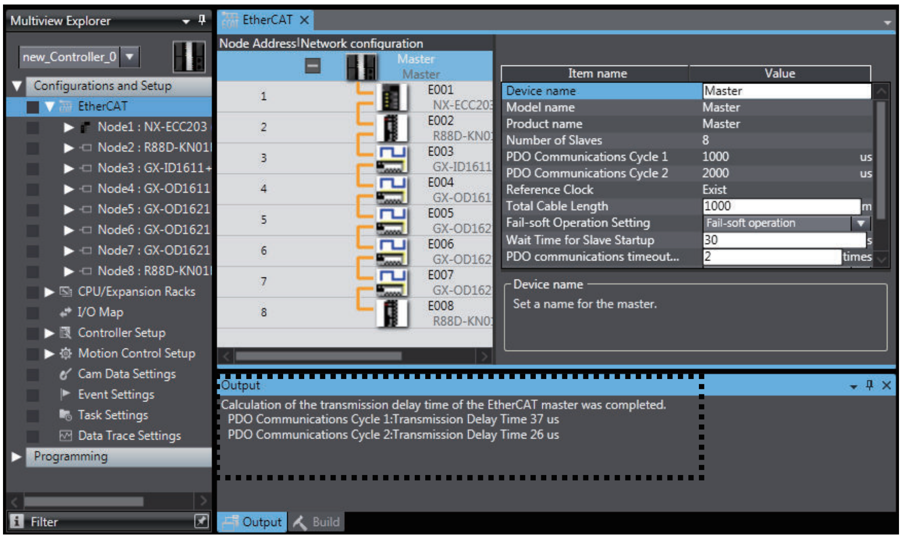
Displaying Transmission Delay Time for EtherCAT Master with the Sysmac Studio

You can use the Sysmac Studio to calculate and display the transmission delay time of the built-in EtherCAT port in the NX-series CPU Unit. The procedure to display the transmission delay time with the Sysmac Studio is given below.

1 Right-click the EtherCAT master or EtherCAT slave in the EtherCAT Tab Page and select **Calculate Transmission Delay Time of the Master**.



2 The transmission delay times of PDO communications cycle 1 and PDO communications cycle 2 are displayed in the Output Tab Page.



Item name	Value
Device name	Master
Model name	Master
Product name	Master
Number of Slaves	8
PDO Communications Cycle 1	1000 us
PDO Communications Cycle 2	2000 us
Reference Clock	Exist
Total Cable Length	1000 m
Fail-soft Operation Setting	Fail-soft operation
Wait Time for Slave Startup	30 s
PDO communications timeout...	2 times

Device name

Set a name for the master.

Output

```
Calculation of the transmission delay time of the EtherCAT master was completed.
  PDO Communications Cycle 1:Transmission Delay Time 37 us
  PDO Communications Cycle 2:Transmission Delay Time 26 us
```

The applied transmission delay times of the EtherCAT master are different depending on the periodic tasks of the CPU Units that refresh I/O with the EtherCAT slaves, as shown below.

Periodic task of the CPU Unit in which I/O is refreshed	Applied transmission delay time of the EtherCAT master
Primary periodic task	Transmission delay time for PDO communications cycle 1
Priority-5 periodic task	Transmission delay time for PDO communications cycle 2

Precautions for Correct Use

- Sysmac Studio does not support the display function for the transmission delay time of the built-in EtherCAT port in the NJ-series CPU Unit.
- You can use the Sysmac Studio to calculate and display the transmission delay times of the EtherCAT master according to the configuration information such as the number of EtherCAT slaves and the EtherCAT frame lengths. When you changed the EtherCAT configurations or setup, you must perform this function again.
- For project unit version 1.40 or later, when you select **Calculation result from the measured value in the actual network configuration** on the **Transmission Delay Time** in the EtherCAT master settings, the value of transmission delay time that is calculated at the actual measurement is displayed in the Output Tab Page.

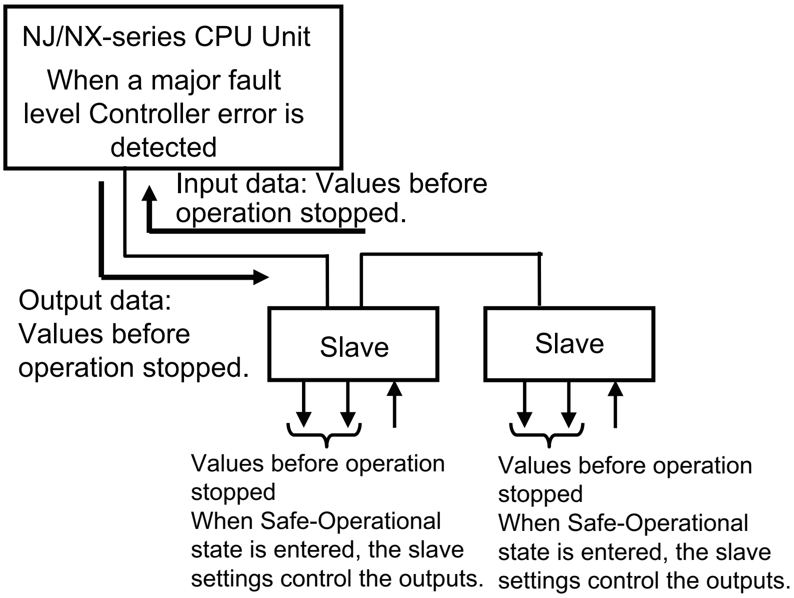
Version Information

Sysmac Studio version 1.13 or higher is required to use the transmission delay time display function for the EtherCAT master.

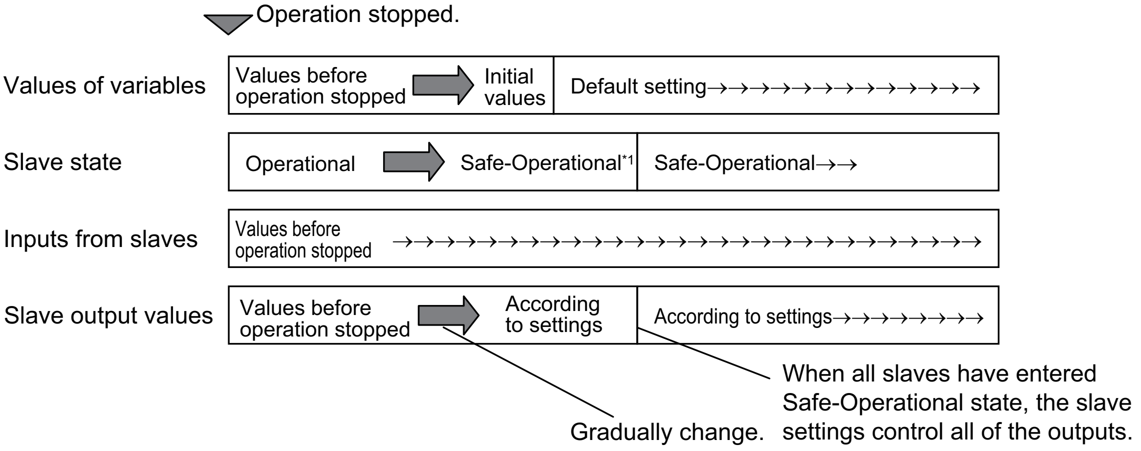
6-1-5 I/O Operations for Major Fault Level Controller Errors and I/O Refreshing with Specified Values

I/O Operation for Major Fault Level Controller Errors

If the NJ/NX-series CPU Unit detects a major fault level Controller error, all slave outputs will retain the process data values from before operation stopped in Operational state. Then the slaves will enter Safe-Operational state[*1] and the slave outputs are processed according to settings in the slave. The inputs from the slaves will retain the process data values from before operation stopped. Message communications are continued.


NJ/NX-series CPU Unit
When a major fault level Controller error is detected
Input data: Values before operation stopped.
Output data: Values before operation stopped.
Slave
Slave
Values before operation stopped
When Safe-Operational state is entered, the slave settings control the outputs.
Values before operation stopped
When Safe-Operational state is entered, the slave settings control the outputs.

When a major fault level Controller error occurs, the values of the variables are not output to the slaves even if they are initialized. The slave outputs are processed according to settings in the slave.


Operation stopped.
Values of variables
Values before operation stopped
Initial values
Default setting→→→→→→→→→→→→→→
Slave state
Operational
Safe-Operational*1
Safe-Operational→→
Inputs from slaves
Values before operation stopped
→→→→→→→→→→→→→→→→→→→→→→→→→→→→→→→
Slave output values
Values before operation stopped
According to settings
According to settings→→→→→→→→
When all slaves have entered Safe-Operational state, the slave settings control all of the outputs.
Gradually change.

*1. The following errors may occur when the slave enters to the Safe-Operational state.

- *Slave Application Error* (84280000 hex) event
- *Slave Initialization Error* (84230000 hex) event
- *Slave State Transition Failed* (84300001 hex) event
- *Slave AL Status Error Detected* (84360000 hex) event

The formula and the maximum time required for all the slaves to enter Safe-Operational state and for the slave output to reach to the value set from the slave settings are shown below. This formula applies to only OMRON slaves.

Maximum time required for the slave output to reach to the value set from the slave settings[*1] =
PDO communications timeout detection count × Task period of the task assigned to slave + 10 ms

*1. The maximum time is 100 to 110 ms if the calculation result is 110 ms or less.

Precautions for Correct Use

You cannot obtain current errors from slaves after a major fault occurs except for those in EtherCAT Slave Terminals.

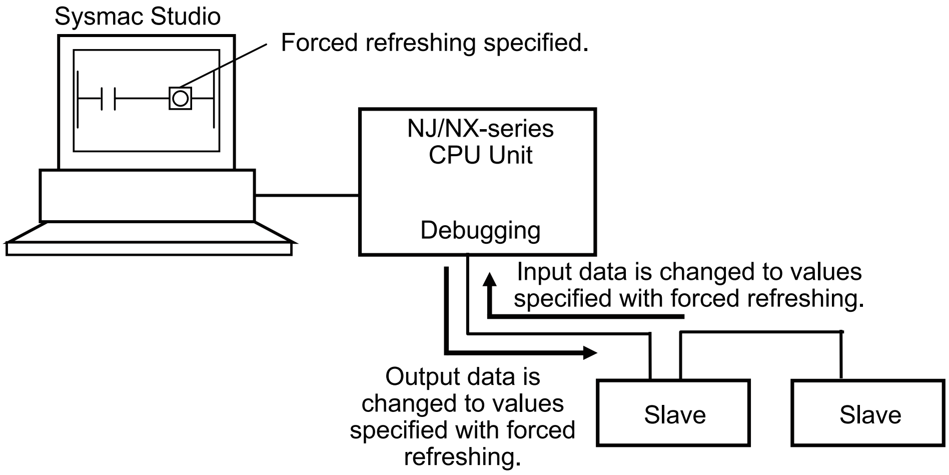
Additional Information

When a major fault level Controller error occurs, the related system-defined variables have the following values.

- For NX701 CPU Units
  _EC_InData1Invalid = TRUE
  _EC_InData2Invalid = TRUE
  _EC_PDSlavTbl[] = FALSE
  _EC_MBXSlavTbl[] = Values before operation stopped
- For NX502, NX102, and NX1P2 CPU Units, and NJ-series CPU Units
  _EC_InDataInvalid = TRUE
  _EC_PDSlavTbl[] = FALSE
  _EC_MBXSlavTbl[] = Values before operation stopped

Forced Refreshing

You can specify forced refreshing from the Sysmac Studio for debugging. This allows you to change process data output values to the slaves and input values from the slaves to the values that you specify in advance.


Sysmac Studio
Forced refreshing specified.
NJ/NX-series CPU Unit
Debugging
Input data is changed to values specified with forced refreshing.
Output data is changed to values specified with forced refreshing.
Slave
Slave

Relationship between Major Fault Level Controller Errors, Normal Operation, and Forced Refreshing

Forced refreshing functions as shown in the following table for major fault level Controller errors and for normal operation.

Condition		Major fault level Controller error	Normal operation
Forced refreshing	**Enabled**	Output data: Values from before operation stopped (Operational state). Slave settings control the outputs (Safe-Operational state). Input data: Values from before operation stopped (Operational state).	Output data: Forced refreshing values Input data: Forced refreshing values
	Disabled	Output data: Values from before operation stopped (Operational state). Slave settings control the outputs (Safe-Operational state). Input data: Values from before operation stopped (Operational state).	Output data: Process data Input data: Process data

Precautions for Correct Use

- You can select whether the master continues or stops communications with all slaves when a communications error occurs. Refer to *5-4 EtherCAT Master Parameter Settings* on page 5-14 for details.
- If noise occurs or an EtherCAT slave is disconnected from the network, any current communications frames may be lost. If frames are lost, slave I/O data is not communicated, and unintended operation may occur. The slave outputs behave according to the slave specifications. For details, refer to relevant manuals for each slave. If a noise countermeasure or slave replacement is required, perform the following processing.
  - Program the *_EC_InDataInvalid* (Input Data Invalid), *_EC_InData1Invalid* (Input Data 1 Invalid), or *_EC_InData2Invalid* (Input Data 2 Invalid) system-defined variable as an interlock condition in the user program. Refer to *6-1-3 Checking Validity of Process Data* on page 6-6.
  - Set the **PDO communications timeout detection count** setting in the EtherCAT master to at least 2. Refer to *5-4 EtherCAT Master Parameter Settings* on page 5-14 for the setting procedure.
- If a communications error prevents the slaves from receiving signals from the master, the slave outputs are processed according to settings in the slave. During the time to change from normal operation to a communications error status, frames will be lost. The outputs for lost frames are different for synced slaves (Servo Drives and encoders) and non-synced slaves. The outputs from synced slaves are processed according to settings in the slaves. The previous values are retained for the slave outputs for non-synced slaves.

Communications status with master	Normal operation	Frames lost	Communications error status
Outputs from synced slaves (Servo Drives and encoders)	Controlled by the values of device variables.	Controlled by the slave settings.	Controlled by the slave settings.
Outputs from non-synced slaves	Controlled by the values of device variables.	The previous values are output.	Controlled by the slave settings.

For details, refer to relevant manuals for each slave.

Additional Information

You can read the status of the *_EC_PDSlavTbl* (Process Data Communicating Slave Table) system-defined variables from the user program to see if I/O refreshing is normal.

6-2 SDO Communications

SDO communications are performed by using EtherCAT communications instructions to access SDO data in slaves when required.

6-2-1 EtherCAT Communications Instructions

You can perform the following SDO communications with EtherCAT communications instructions.

- Reading and writing SDO data

Reading and Writing SDO data

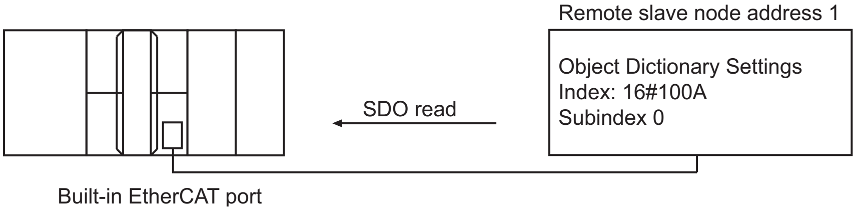
Function	Instruction	Description
CoE messages (Read CoE SDO)	EC_CoESDORead	You set parameters to read data from the slave's object dictionary (SDO data).
CoE messages (Write CoE SDO)	EC_CoESDOWrite	You set parameters to write data to the slave's object dictionary (SDO data).

6-2-2 Sample Programming

This sample uses an EtherCAT communications instruction to read the software version of an OMRON R88D-KN01L-ECT Servo Drive. The node address of the slave is 1.

The object index for the software version is 16#100A. The subindex is 0.

The read value is stored in STRING variable *VersionInfo*.



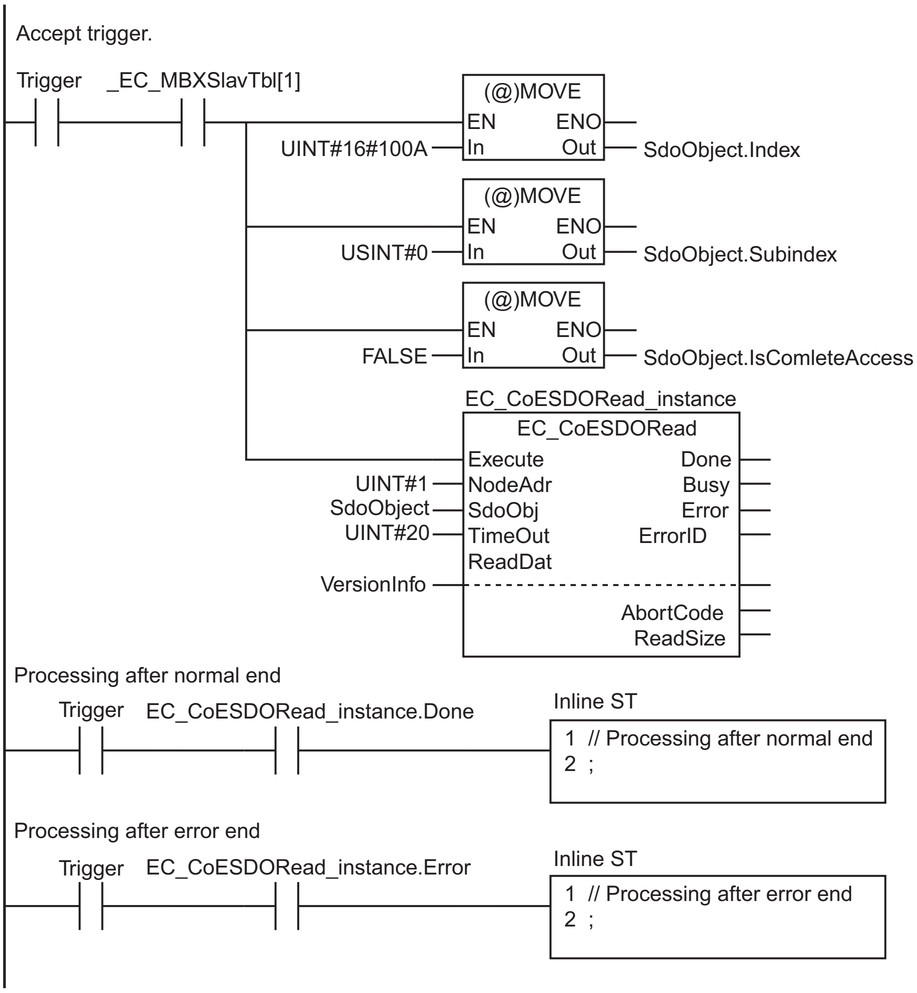
- **LD**

Internal Variables	Variable	Data type	Initial value	Comment
	Trigger	BOOL	False	Execution condition
	SdoObject	_sSDO_ACCESS	(Index:=0, Subindex:=0, IsCompleteAccess:=False)	SDO parameter
	VersionInfo	STRING[256]	‘’	Read data
	EC_CoESDORead_instance	EC_CoESDORead		

External Variables	Variable	Data type	Constant	Comment

_EC_MBXSlavTbl	ARRAY[1..192] OF BOOL[*1]	☑	Message Communications Enabled Slave Table

*1. This applies to an NX102 CPU Unit. The range of array elements varies with the model of the CPU Unit. Refer to the *_EC_MBXSlavTbl* system-defined variable in *7-1-3 EtherCAT Master Function Module, Category Name: _EC* on page 7-9.



● ST

Internal Variables	Variable	Data type	Initial value	Comment
	Trigger	BOOL	False	Execution condition
	SdoObject	_sSDO_ACCESS	(Index:=0, Subindex:=0, IsCompleteAccess:=False)	SDO parameter
	DoSdoRead	BOOL	False	Processing
	VersionInfo	STRING[256]	‘’	Read data
	NormalEnd	UINT	0	Normal end
	ErrorEnd	UINT	0	Error end
	EC_CoESDORead_instance	EC_CoESDORead		

External Variables	Variable	Data type	Constant	Comment
	_EC_MBXSlavTbl	ARRAY[1..192] OF BOOL[*1]	☑	Message Communications Enabled Slave Table

*1. This applies to an NX102 CPU Unit. The range of array elements varies with the model of the CPU Unit. Refer to the *_EC_MBXSlavTbl* system-defined variable in *7-1-3 EtherCAT Master Function Module, Category Name: _EC* on page 7-9.

```
// Detect when Trigger changes to TRUE.
IF ( (Trigger=TRUE) AND (DoSdoRead=FALSE) AND (_EC_MBXSlavTbl[1]=TRUE) ) THEN
    DoSdoRead                     :=TRUE;
    SdoObject.Index               :=UINT#16#100A;
    SdoObject.Subindex            :=USINT#0;
    SdoObject.IsCompleteAccess:=FALSE;
    EC_CoESDORead_instance(
        Execute:=FALSE,             // Initialize instance.
        ReadDat:=VersionInfo);      // Dummy
END_IF;

// Execute EC_CoESDORead instruction.
IF (DoSdoRead=TRUE) THEN
    EC_CoESDORead_instance(
        Execute  :=TRUE,
        NodeAdr :=UINT#1,           // Node address 1
        SdoObj   :=SdoObject,       // SDO parameter
        TimeOut :=UINT#20,          // Timeout time: 2.0 s
        ReadDat:=VersionInfo);      // Read data

    IF (EC_CoESDORead_instance.Done=TRUE) THEN
        // Processing after normal end
        NormalEnd:=NormalEnd+UINT#1;
    ELSIF (EC_CoESDORead_instance.Error=TRUE) THEN
        // Processing after error end
        ErrorEnd    :=ErrorEnd+UINT#1;
    END_IF;

END_IF;
```

6-2-3 Execution Timing of SDO Communications

SDO communications are executed in the system service time of the NJ/NX-series CPU Unit. System services are executed in the period between execution of all of the tasks. Refer to the *NJ/NX-series CPU Unit Software User's Manual (Cat. No. W501)* for details on the execution timing of system services.

6-2-4 Minimum Message Response Time for SDO Communications

The minimum message response time for SDO communications is the time from when the SDO communications instruction is executed in the user program until execution of the instruction is completed. When the instruction is executed in the user program, the master sends a frame to the slave in the system service. When the slave receives the frame, it processes it. When the slave completes processing, the master receives a response from the slave to complete execution of the instruction.

Additional Information

The minimum message response time depends on the unused time in task execution, slave SDO size, and slave SDO processing time. In the timing chart below, the time from when the SDO communications instruction is executed until execution of the instruction is completed should be used for reference only.

NX-series CPU Units

The calculation formula for the minimum message response time differs between when the task period of primary periodic task is smaller than 4 ms and when the task period is 4 ms or longer.

● When the Task Period of the Primary Periodic Task is Smaller than 4 ms

Minimum message response time (ms) =

Task period of the task that executes the SDO communications instruction

+ (Slave SDO processing time[*1]/Task period of primary periodic task)[*2] × Task period of primary periodic task

+ (SDO response data size/Transmission size)[*2] × Task period of primary periodic task

*1. The slave SDO processing time depends on the slaves. Refer to relevant manuals for each slave.

*2. Round up the calculation result in parentheses to make an integer.

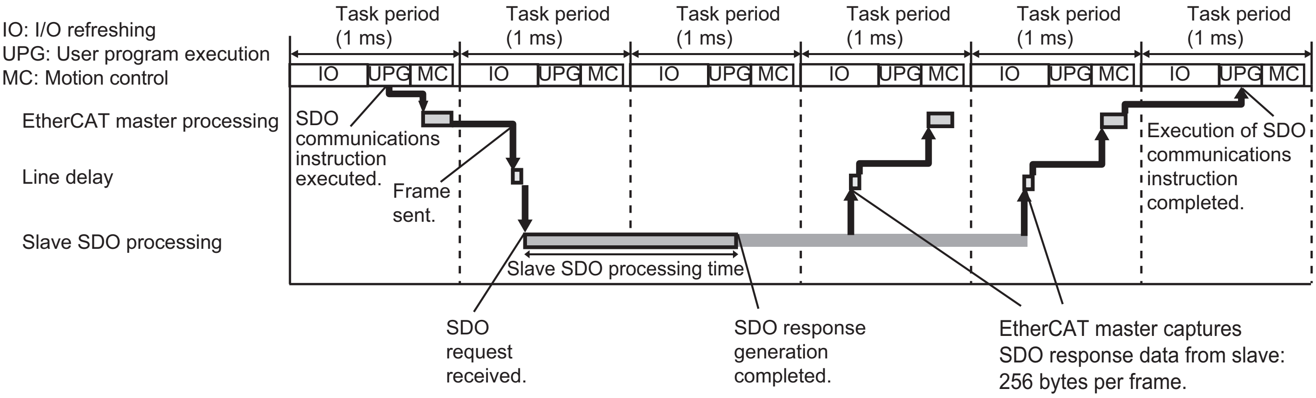
The transmission size is determined by the task period of primary periodic task as shown below.

Task period of the primary periodic task	Transmission size
125 µs	64 bytes
250 µs	128 bytes
500 µs or longer	256 bytes

The following timing chart shows an example of the timing from an execution of the SDO communications instruction to the completion of the instruction execution, based on the performance model below.
Here, it is assumed that the program that executes the SDO communications instructions is assigned to the primary periodic task.

Performance Model Example

Task period of primary periodic task [ms]	1
SDO response data size [byte]	512
Slave SDO processing time [ms]	1.2
Message response time [ms]	5


IO: I/O refreshing
UPG: User program execution
MC: Motion control
Task period (1 ms)
Task period (1 ms)
Task period (1 ms)
Task period (1 ms)
Task period (1 ms)
Task period (1 ms)
IO
UPG
MC
EtherCAT master processing
Line delay
Slave SDO processing
SDO communications instruction executed.
Frame sent.
Slave SDO processing time
Execution of SDO communications instruction completed.
SDO request received.
SDO response generation completed.
EtherCAT master captures SDO response data from slave: 256 bytes per frame.

● When the Task Period of the Primary Periodic Task is 4 ms or Longer

Minimum message response time (ms) =

Task period of the task that executes the SDO communications instruction

+ (Slave SDO processing time[*1]/Divided period[*2])[*3] × Divided period

+ (SDO response data size/256 bytes)[*3] × Divided period

*1. The slave SDO processing time depends on the slaves. Refer to relevant manuals for each slave.

*2. The divided period is obtained by dividing the task period of primary periodic task. If the task period of primary periodic task is equal to or more than 4 ms, the EtherCAT master sends a message communications frame to a slave and receives a message response data from the slave, according to the divided period. Assign 1 ms to the divided period for this calculation.

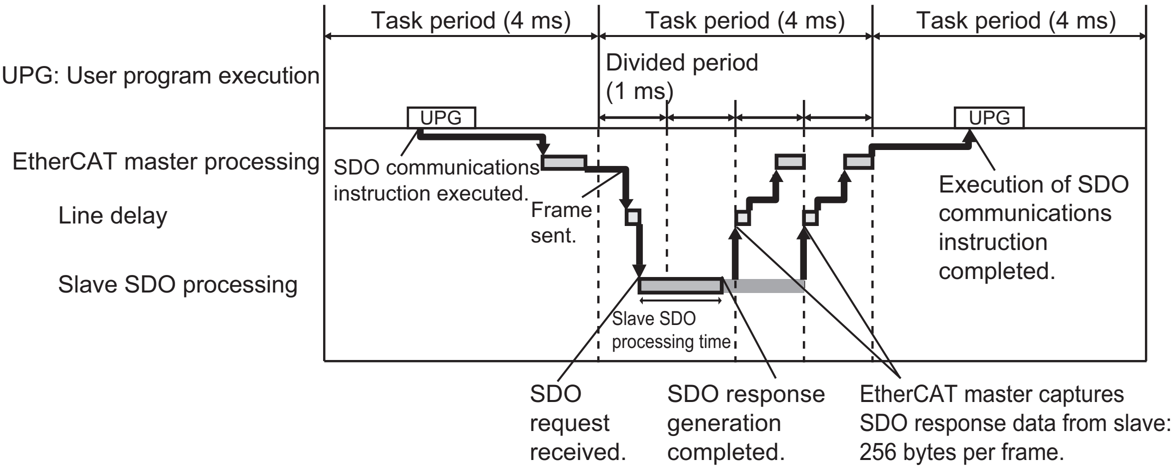
*3. Round up the calculation result in parentheses to make an integer.

The following timing chart shows an example of the timing from an execution of the SDO communications instruction to the completion of the instruction execution, based on the performance model below.

Here, it is assumed that the program that executes the SDO communications instructions is assigned to the primary periodic task.

Performance Model Example

Task period of primary periodic task [ms]	4
Divided period [ms]	1
SDO response data size [byte]	512
Slave SDO processing time [ms]	1.2
Message response time [ms]	8



NJ-series CPU Units

The calculation formula for the minimum message response time is given below.

Minimum message response time (ms) =

Task period of the task that executes the SDO communications instruction

+ (Slave SDO processing time[*1]/Task period of primary periodic task)[*2] x Task period of primary periodic task

+ (SDO response data size/256 bytes)[*2] × Task period of primary periodic task

*1. The slave SDO processing time depends on the slaves. Refer to relevant manuals for each slave.

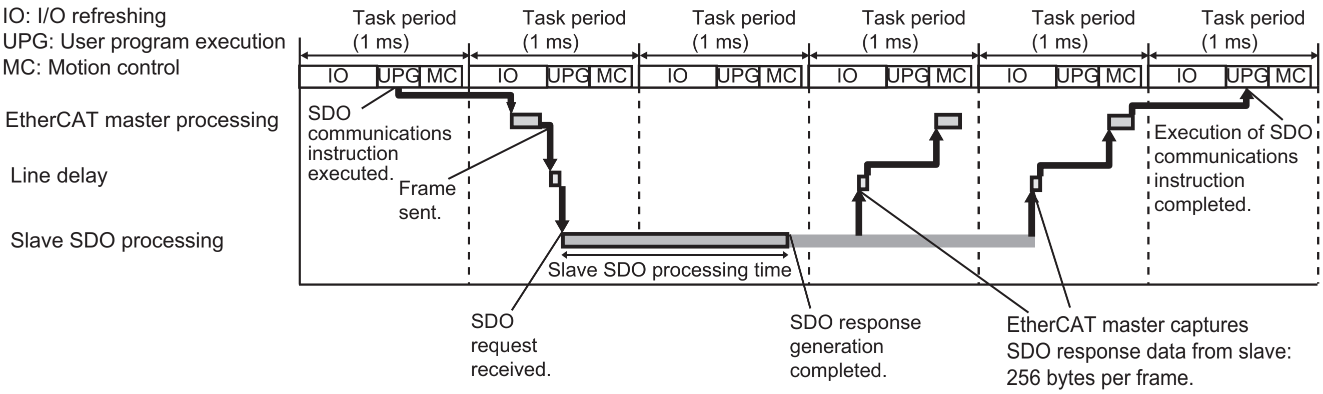
*2. Round up the calculation result in parentheses to make an integer.

The following timing chart shows an example of the timing from an execution of the SDO communications instruction to the completion of the instruction execution, based on the performance model below.

Here, it is assumed that the program that executes the SDO communications instructions is assigned to the primary periodic task.

Performance Model Example

Task period of primary periodic task [ms]	1
SDO response data size [byte]	512
Slave SDO processing time [ms]	1.2
Message response time [ms]	5


IO: I/O refreshing
UPG: User program execution
MC: Motion control
EtherCAT master processing
Line delay
Slave SDO processing
Task period (1 ms)
Task period (1 ms)
Task period (1 ms)
Task period (1 ms)
Task period (1 ms)
Task period (1 ms)
IO
UPG
MC
SDO communications instruction executed.
Frame sent.
Slave SDO processing time
Execution of SDO communications instruction completed.
SDO request received.
SDO response generation completed.
EtherCAT master captures SDO response data from slave: 256 bytes per frame.

6-3 Instructions Used in EtherCAT Communications

6-3-1 EtherCAT Communications Instructions

Function	Instruction	Description
Start EtherCAT Packet Monitor	EC_StartMon[*1]	Starts executing the packet monitoring function that the EtherCAT master of the NJ/NX-series CPU Unit has.
Stop EtherCAT Packet Monitor	EC_StopMon[*1]	Stops executing the packet monitoring function that the EtherCAT master of the NJ/NX-series CPU Unit has.
Save Packet Data File	EC_SaveMon[*1]	Saves the monitored data into system of the NJ/NX-series CPU Unit, by using the packet monitoring function that the EtherCAT master of the NJ/NX-series CPU Unit has. The captured files in the system of the NJ/NX-series CPU Unit are not retained when the power is interrupted.
Copy Packet Data File to SD Memory Card	EC_CopyMon[*1]	Saves (Copies) to an SD Memory Card the packet data that was saved into the system of the NJ/NX-series CPU Unit from the EtherCAT master of the NJ/NX-series CPU Unit. The files saved to an SD Memory Card are retained after the power is interrupted.
Disconnect EtherCAT Slave	EC_DisconnectSlave	Temporarily disconnects a slave from the EtherCAT network for maintenance, such as replacement of the slave.
Connect EtherCAT Slave	EC_ConnectSlave	Reconnects a temporarily disconnected slave to the EtherCAT network after maintenance, such as replacement of the slave.
Enable/Disable EtherCAT Slave	EC_ChangeEnableSetting	Enables or disables an EtherCAT slave.
Get EtherCAT Error Status	GetECError	Gets the status of Controller errors (partial faults or minor faults) that occur in the EtherCAT master and the highest-level event code for the current errors.
Reset EtherCAT Error	ResetECError	Resets Controller errors in the EtherCAT master. (Execute this instruction only after eliminating the cause of the error.)

[*1] For the NJ301-□□□□ CPU Units, the CPU Unit with unit version 1.10 or later and Sysmac Studio version 1.12 or higher are required to use the packet monitoring. However, for project unit version 1.40 or later, the packet monitoring cannot be used regardless of the models of the CPU Units. If the packet monitoring cannot be used, the *_EC_PktMonStop* system-defined variable, which shows the operating status of packet monitoring, will always be TRUE.
In addition, if you execute any EtherCAT instructions for packet monitoring (EC_StartMon, EC_StopMon, EC_SaveMon, or EC_CopyMon) in the user program, an error that shows packet monitoring cannot be used occurs.

Note Refer to the *NJ/NX-series Instructions Reference Manual (Cat. No. W502)* for details on the instructions that are used with EtherCAT communications.

7

System-defined Variables Related to the Built-in EtherCAT Port

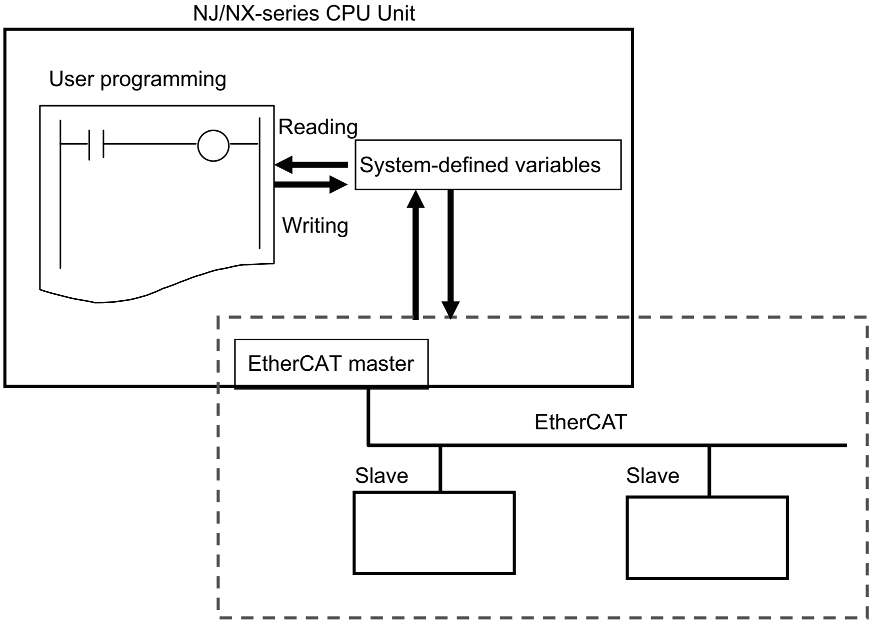
This section describes the system-defined variables that are related to the built-in EtherCAT port.

7-1 System-defined Variables That Are Related to the Built-in EtherCAT Port

7-1-1 What Are System-defined Variables?

System-defined variables are variables that are defined by the system for use with EtherCAT communications. These are provided in advance in the global variable table.

The user program can input status and set parameters for the EtherCAT master and slaves by reading and writing system-defined variables.



7-1-2 System-defined Variables

● Functional Classification: EtherCAT Communications Errors

Variable name	Meaning	Function	Data type	Range of values	Reference
_EC_ErrSta	EtherCAT Error	This system-defined variable provides the collective status of errors in the EtherCAT Master Function Module. Refer to *Meanings of Error Status Bits* for the meanings of the error status bits.	WORD	16#0000 to 16#40F0	page 7-10
_EC_PortErr	Communications Port Error	This system-defined variable provides the collective status of errors in the communications ports for the EtherCAT master. Refer to *Meanings of Error Status Bits* for the meanings of the error status bits.	WORD	16#0000 to 16#00F0	page 7-10
_EC_MstrErr	Master Error	This system-defined variable provides the collective status of EtherCAT master errors and slave errors detected by the EtherCAT master. Refer to *Meanings of Error Status Bits* for the meanings of the error status bits.	WORD	16#0000 to 16#00F0	page 7-10

Variable name	Meaning	Function	Data type	Range of values	Reference
_EC_SlavErr	Slave Error	This system-defined variable provides the collective status of all the error status for EtherCAT slaves. Refer to *Meanings of Error Status Bits* for the meanings of the error status bits.	WORD	16#0000 to 16#00F0	page 7-10.
_EC_SlavErrTbl	Slave Error Table	This system-defined variable gives the error status for each EtherCAT slave. This error status shows status of error in each slave in the actual system configuration. This variable array indicates slaves in which there are errors. Status is provided for each EtherCAT slave node address (1 to 512)*1. Refer to *Meanings of Error Status Bits* for the meanings of the error status bits.	Array [1..512] OF WORD*1	16#0000 to 16#00F0	page 7-11
_EC_MacAdrErr	MAC Address Error	TRUE if the MAC Address Error (14400000 hex) event occurred.	BOOL	TRUE or FALSE	page 7-11
_EC_LanHwErr	Communications Controller Error	TRUE if the Communications Controller Error (047C0000 hex) event occurred.	BOOL	TRUE or FALSE	page 7-11
_EC_LinkOffErr	Link OFF Error	TRUE if the Link OFF Error (84200000 hex) event occurred.	BOOL	TRUE or FALSE	page 7-11
_EC_NetCfgErr	Network Configuration Information Error	TRUE if the Network Configuration Information Error (34400000 hex) event occurred.	BOOL	TRUE or FALSE	page 7-11
_EC_NetCfgCmpErr	Network Configuration Verification Error	TRUE if one of the following events occurred. • Network Configuration Verification Error (84220000 hex) • Network Configuration Verification Error (Slave Unconnected) (84380000 hex) • Network Configuration Verification Error (Unnecessary Slave Connected) (84320003 hex) • Network Configuration Verification Error (Mismatched Slave) (84330004 hex) • Network Configuration Verification Error (Incorrect Ring Wiring) (843A0000 hex)	BOOL	TRUE or FALSE	page 7-12
_EC_NetTopologyErr	Network Configuration Error	TRUE if one of the following events occurred. • Network Configuration Error (84210000 hex) • Incorrect Wiring Detected (843C0000 hex)	BOOL	TRUE or FALSE	page 7-12
_EC_PDCommErr	Process Data Communications Error	TRUE if one of the following events occurred. • Process Data Communications Error (842C0000 hex) • Illegal Slave Disconnection Detected (84310002 hex) • Slave PDI WDT Error Detected (84340000 hex)	BOOL	TRUE or FALSE	page 7-12
_EC_PDTimeoutErr	Process Data Reception Timeout Error	TRUE if the Process Data Reception Timeout (842B0000 hex) event occurred.	BOOL	TRUE or FALSE	page 7-12

Variable name	Meaning	Function	Data type	Range of values	Reference
_EC_PDSendErr	Process Data Transmission Error	TRUE if the Process Data Transmission Error (84290000 hex) event occurred.	BOOL	TRUE or FALSE	page 7-12
_EC_SlavAdrDupErr	Slave Node Address Duplicated Error	TRUE if the Slave Node Address Duplicated (24200000 hex) event occurred.	BOOL	TRUE or FALSE	page 7-13
_EC_SlavInitErr	Slave Initialization Error	TRUE if one of the following events occurred. • Slave Initialization Error (84230000 hex) • Slave State Transition Failed (84300001 hex)	BOOL	TRUE or FALSE	page 7-13
_EC_SlavAppErr	Slave Application Error	TRUE if one of the following events occurred. • Slave Application Error (84280000 hex) • Slave AL Status Error Detected (84360000 hex)	BOOL	TRUE or FALSE	page 7-13
_EC_MsgErr	EtherCAT Message Error	TRUE if one of the following events occurred. • EtherCAT Message Error (842D0000 hex) • Illegal Mailbox Received (84350000 hex)	BOOL	TRUE or FALSE	page 7-13
_EC_SlavEmergErr	Emergency Message Detected	TRUE if the Emergency Message Detected (64200000 hex) event occurred.	BOOL	TRUE or FALSE	page 7-13
_EC_IndataInvalidErr (Ver.1.13)	Input Process Data Invalid Error	TRUE if the Input Process Data Invalid Error (842F0000 hex) event occurred.	BOOL	TRUE or FALSE	page 7-14
_EC_CommErrTbl	Communications Error Slave Table	Slaves are given in the table in the order of slave node addresses. The corresponding slave element is TRUE if the master detected an error for the slave.	Array [1..512] OF BOOL[*1]	TRUE or FALSE	page 7-14
_EC_CycleExceeded	EtherCAT Communications Cycle Exceeded	TRUE if the EtherCAT Communications Cycle Exceeded (34410000 hex) event occurred. **Note** You can use this system-defined variable only for NX-series CPU Units.	BOOL	TRUE or FALSE	page 7-14

*1. For the NX102 CPU Units, NX1P2 CPU Units, and NJ-series CPU Units, the node address is 1 to 192 and the data type is ARRAY [1..192] OF WORD.
For the NX502 CPU Units, the node address is 1 to 256 and the data type is Array [1..256] OF WORD.

Additional Information

Typical Relationships for the EtherCAT Error Flags

Variable name	Meaning	Variable name	Meaning	Variable name	Meaning	Event Levels
_EC_ErrSta	EtherCAT Error	_EC_PortErr	Communications Port Error	_EC_MacAdrErr	MAC Address Error	Partial fault level
				_EC_LanHwErr	Communications Controller Error	
				_EC_LinkOffErr	Link OFF error	
		_EC_MstrErr	Master Error	_EC_NetCfgErr	Network Configuration Information Error	Minor fault level
				_EC_NetCfgCmpErr	Network Configuration Verification Error	
				_EC_NetTopologyErr	Network Configuration Error	
				_EC_PDCommErr	Process Data Communications Error	
				_EC_PDTimeoutErr	Process Data Reception Timeout Error	
				_EC_PDSendErr	Process Data Transmission Error	
				_EC_SlavAdrDupErr	Slave Node Address Duplicated Error	
				_EC_SlavInitErr	Slave Initialization Error	
				_EC_SlavAppErr	Slave Application Error	
				_EC_CommErrTbl	Communications Error Slave Table	
				_EC_CycleExceeded	EtherCAT Communications Cycle Exceeded	
				_EC_MsgErr	EtherCAT Message Error	Observation
				_EC_SlavEmergErr	Emergency Message Detected	
		_EC_SlavErr	Slave Error	_EC_SlavErrTbl	Slave Error Table	Defined by the slave.

Note The values of all system-defined variables that are related to errors in EtherCAT communications do not change until the cause of the error is removed and then the error in the Controller is reset with the troubleshooting functions of the Sysmac Studio or the ResetECError instruction.

● Functional Classification: EtherCAT Communications Status

Variable name	Meaning	Function	Data type	Range of values	Reference
_EC_RegSlavTbl	Registered Slave Table	This table indicates the slaves that are registered in the network configuration information. Slaves are given in the table in the order of slave node addresses. The element for a slave is TRUE if the corresponding slave is registered.	Array [1..512] OF BOOL*1	TRUE or FALSE	page 7-14
_EC_EntrySlavTbl	Network Connected Slave Table	This table indicates which slaves are connected to the network. Slaves are given in the table in the order of slave node addresses. The element for a slave is TRUE if the corresponding slave has entered the network.	Array [1..512] OF BOOL*1	TRUE or FALSE	page 7-15
_EC_MBXSlavTbl	Message Communications Enabled Slave Table	This table indicates the slaves that can perform message communications. Slaves are given in the table in the order of slave node addresses. The element for a slave is TRUE if message communications are enabled for it (pre-operational, safe-operation, or operational state). **Note** Use this variable to confirm that message communications are possible for the relevant slave before you execute message communications with an EtherCAT slave.	Array [1..512] OF BOOL*1	TRUE or FALSE	page 7-15
_EC_PDSlavTbl	Process Data Communicating Slave Table	This table indicates the slaves that are performing process data communications. Slaves are given in the table in the order of slave node addresses. The element for a slave is TRUE if process data of the corresponding slave is enabled (operational) for both slave inputs and outputs. **Note** Use this variable to confirm that the data for the relevant slave is valid before controlling an EtherCAT slave.	Array [1..512] OF BOOL*1	TRUE or FALSE	page 7-15
_EC_DisconnSlavTbl	Disconnected Slave Table	Slaves are given in the table in the order of slave node addresses. The element for a slave is TRUE if the corresponding slave was disconnected.	Array [1..512] OF BOOL*1	TRUE or FALSE	page 7-16
_EC_DisableSlavTbl	Disabled Slave Table	Slaves are given in the table in the order of slave node addresses. The element for a slave is TRUE if the corresponding slave is disabled.	Array [1..512] OF BOOL*1	TRUE or FALSE	page 7-16
_EC_PDActive	Process Data Communications Status	TRUE when process data communications are performed with all slaves*. *Disabled slaves are not included.	BOOL	TRUE or FALSE	page 7-16
_EC_PktMonStop	Packet Monitoring Stopped	TRUE when packet monitoring is stopped.	BOOL	TRUE or FALSE	page 7-16
_EC_LinkStatus	Link Status	TRUE if the Communications Controller link status is Link ON.	BOOL	TRUE or FALSE	page 7-17

Variable name	Meaning	Function	Data type	Range of values	Reference
_EC_PktSaving	Saving Packet Data File	Shows whether a packet data file is being saved. TRUE: Packet data file being saved. FALSE: Packet data file not being saved.	BOOL	TRUE or FALSE	page 7-17
_EC_InDataInvalid	Input Data Invalid	TRUE when process data communications established in the primary periodic task are not normal and the input data is not valid.	BOOL	TRUE or FALSE	page 7-17
_EC_InData1Invalid	Input Data1 Invalid	TRUE when process data communications established in the primary periodic task are not normal and the input data is not valid. **Note** You can use this system-defined variable only for NX-series CPU Units.	BOOL	TRUE or FALSE	page 7-17
_EC_InData2Invalid	Input Data2 Invalid	TRUE when process data communications established in the priority-5 periodic task are not normal and the input data is not valid. **Note** You can use this system-defined variable only for NX-series CPU Units. **Note** This variable is always TRUE for the NX502 CPU Units, NX102 CPU Units, and NX1P2 CPU Units.	BOOL	TRUE or FALSE	page 7-18
_EC_RingBreaking (Ver.1.40)	Ring Disconnection	TRUE when all slaves in the ring topology except for disabled slaves are connected and if there is only one point of disconnection in the communications cables in the ring topology.	BOOL	TRUE or FALSE	page 7-18
_EC_RingBreakNodeAdr (Ver.1.40)	Slave Node Address Before Ring Disconnection	When the *_EC_RingBreaking* (Ring Disconnection) system-defined variable is TRUE, the slave node address before point of disconnection is stored. When the *_EC_RingBreaking* (Ring Disconnection) system-defined variable is FALSE, “0” is stored.	UINT	0 to maximum number of node addresses	page 7-18

*1. For the NX102 CPU Units, NX1P2 CPU Units, and NJ-series CPU Units, the data type is ARRAY [1..192] OF BOOL.
For the NX502 CPU Units, the data type is Array [1..256] OF BOOL.

Note All system-defined variables that are related to the status of EtherCAT communications give the current status.

● Functional Classification: EtherCAT Communications Diagnosis/Statistics Log

Variable name	Meaning	Function	Data type	Range of values	Reference
_EC_StatisticsLogEnable (Ver.1.11)	Diagnosis/Statistics Log Enable	Changes to TRUE when the diagnosis/statistics log is started. Changes to FALSE when the diagnosis/statistics log is ended.	BOOL	TRUE or FALSE	page 7-18

Variable name	Meaning	Function	Data type	Range of values	Reference
_EC_StatisticsLogCycleSec (Ver.1.11)	Diagnosis/Statistics Log Cycle	Specifies the interval to write the diagnostic and statistical information of the diagnosis/statistics log in units of seconds. When 0 is specified, the diagnostic and statistical information is written only once when the diagnosis/statistics log is ended. **Note** The write interval does not change even if you change the value of this system-defined variable while the diagnosis/statistics log operation is in progress.	UINT	0, or 30 to 1800	page 7-19
_EC_StatisticsLogBusy (Ver.1.11)	Diagnosis/Statistics Log Busy	TRUE while the diagnosis/statistics log operation is in progress.	BOOL	TRUE or FALSE	page 7-19
_EC_StatisticsLogErr (Ver.1.11)	Diagnosis/Statistics Log Error	TRUE when the diagnosis/statistics log failed to start or it is impossible to write into the log. The value of this flag is determined when *_EC_StatisticsLogBusy* (Diagnosis/Statistics Log Busy) changes to FALSE after the diagnosis/statistics log operation is started. The error end is caused by the following. • Another records cannot be added in the log file because the capacity of the SD Memory Card is fully used. • The SD Memory Card is write-protected. • There is no SD Memory Card. • The function cannot be started because the value specified for *_EC_StatisticsLogCycleSec* (Diagnosis/Statistics Log Cycle) is invalid.	BOOL	TRUE or FALSE	page 7-19

● Meanings of Error Status Bits

The meanings of the individual bits in the error status are shown in the following table.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
WORD			-	-	-	-	-	-					-	-	-	-

Bit	Description
15	Master-detected error: This bit indicates whether the master detected a Controller error in the Unit/slave for the error status of the Controller error. TRUE: The master detected a Controller error. FALSE: The master has not detected a Controller error.
14	Collective slave error status: This bit indicates if a Controller error is detected for levels (e.g., a Unit, slave, axis, or axes group) that are lower than the event source (i.e., for a function module). TRUE: A Controller error has occurred at a lower level. FALSE: A Controller error has not occurred at a lower level.
13 to 8	Reserved.
7	This bit indicates whether a major fault level Controller error has occurred. TRUE: A major fault level Controller error has occurred. FALSE: A major fault level Controller error has not occurred.

Bit	Description
6	This bit indicates whether a partial fault level Controller error has occurred. TRUE: A partial fault level Controller error has occurred. FALSE: A partial fault level Controller error has not occurred.
5	This bit indicates whether a minor fault level Controller error has occurred. TRUE: A minor fault level Controller error has occurred. FALSE: A minor fault level Controller error has not occurred.
4	This bit indicates whether an observation level Controller error has occurred. TRUE: An observation level Controller error has occurred. TRUE: An observation level Controller error has not occurred.
3 to 0	Reserved.

A list of variables for error status is given below. The following table shows whether bit14 and bit15 of each variable are valid or invalid and whether they can be used in the user program.

Variable name	Valid or invalid for bit 15	Valid or invalid for bit 14	Usage in user program
_ErrSta (Controller Error Status)	Valid	Valid	Not possible[*1]
_PLC_ErrSta (PLC Function Module Error Status)	Invalid	Invalid	Possible
_CJB_ErrSta (I/O Bus Error Status)	Valid	Valid	Not possible[*2]
_CJB_MstrErrSta (I/O Bus Master Error Status)	Invalid	Invalid	
_CJB_UnitErrSta (I/O Bus Unit Error Status)	Valid	Invalid	
_NXB_ErrSta (NX Bus Function Module Error Status)	Invalid	Valid	Not recommended[*3]
_NXB_MstrErrSta (NX Bus Function Module Master Error Status)	Invalid	Valid	
_NXB_UnitErrStaTbl (NX Bus Function Module Unit Error Status)	Invalid	Valid	
_MC_ErrSta (MC Error Status)	Invalid	Valid	Possible
_MC_ComErrSta (MC Common Error Status)	Invalid	Invalid	
_MC_AX_ErrSta (Axis Error Status)	Invalid	Invalid	
_MC_GRP_ErrSta (Axes Group Error Status)	Invalid	Invalid	
_EC_ErrSta (EtherCAT Error)	Invalid	Valid	Possible
_EC_PortErr (Communications Port Error)	Invalid	Invalid	
_EC_MstrErr (Master Error)	Invalid	Invalid	
_EC_SlavErr (Slave Error)	Invalid	Invalid	
_EC_SlavErrTbl (Slave Error Table)	Invalid	Invalid	
_EIP_ErrSta (EtherNet/IP Error)	Invalid	Invalid	Possible
_EIP_PortErr (Communications Port Error), *_EIP1_PortErr* (Communications Port1 Error), *_EIP2_PortErr* (Communications Port2 Error)	Invalid	Invalid	
_EIP_CipErr (CIP Communications Error), *_EIP1_CipErr* (CIP Communications1 Error), *_EIP2_CipErr* (CIP Communications2 Error)	Invalid	Invalid	
_EIP_TcpAppErr (TCP Application Communications Error)	Invalid	Invalid	
_XBU_ErrSta (X Bus Function Module Error Status)	Invalid	Valid	Not recommended[*3]
_XBU_MstrErr (X Bus Function Module Master Error Status)	Invalid	Invalid	
_XBU_UnitErr (X Bus Function Module Unit Error Status)	Invalid	Valid	
_XBU_UnitErrTbl (X Bus Function Module Unit Error Status Table)	Invalid	Invalid	

*1. Do not use this variable in the user program. There may be a delay in updating it and concurrency problems in relation to the error status of the function module. Use this variable only to access status through communications from an external device.

*2. Do not use this variable in the user program. There may be a delay in updating it. Use this variable only to access status through communications from an external device such as an HMI.

*3. We do not recommend the use of this variable in the user program. There may be a delay in updating it. Use this variable only to access status through communications from an external device such as an HMI.

7-1-3 EtherCAT Master Function Module, Category Name: _EC

● Functional Classification: EtherCAT Communications Errors

Variable name	_EC_ErrSta				
Meaning	EtherCAT Error			**Global/local**	Global
Function	This system-defined variable provides the collective status of errors in the EtherCAT Master Function Module. Refer to *Meanings of Error Status Bits* on page 7-8 for the meanings of the error status bits.				
Data type	WORD			**Range of values**	16#0000 to 16#40F0
R/W access	R	**Retained**	Not retained.	**Network Publish**	Published.
Usage in user program	Possible.	**Related instructions**	Get EtherCAT Error Status • GetECError Reset EtherCAT Error • ResetECError		

Variable name	_EC_PortErr				
Meaning	Communications Port Error			**Global/local**	Global
Function	This system-defined variable provides the collective status of errors in the communications ports for the EtherCAT master. Refer to *Meanings of Error Status Bits* on page 7-8 for the meanings of the error status bits.				
Data type	WORD			**Range of values**	16#0000 to 16#00F0
R/W access	R	**Retained**	Not retained.	**Network Publish**	Published.
Usage in user program	Possible.	**Related instructions**	Get EtherCAT Error Status • GetECError Reset EtherCAT Error • ResetECError		

Variable name	_EC_MstrErr				
Meaning	Master Error			**Global/local**	Global
Function	This system-defined variable provides the collective status of EtherCAT master errors and slave errors detected by the EtherCAT master. Refer to *Meanings of Error Status Bits* on page 7-8 for the meanings of the error status bits.				
Data type	WORD			**Range of values**	16#0000 to 16#00F0
R/W access	R	**Retained**	Not retained.	**Network Publish**	Published.
Usage in user program	Possible.	**Related instructions**	Get EtherCAT Error Status • GetECError Reset EtherCAT Error • ResetECError		

Variable name	_EC_SlavErr				
Meaning	Slave Error			**Global/local**	Global
Function	This system-defined variable provides the collective status of all the error status for EtherCAT slaves. Refer to *Meanings of Error Status Bits* on page 7-8 for the meanings of the error status bits.				
Data type	WORD			**Range of values**	16#0000 to 16#00F0
R/W access	R	**Retained**	Not retained.	**Network Publish**	Published.
Usage in user program	Possible.	**Related instructions**	Get EtherCAT Error Status • GetECError Reset EtherCAT Error • ResetECError		

Variable name	_EC_SlavErrTbl				
Meaning	Slave Error Table			**Global/local**	Global
Function	This system-defined variable gives the error status for each EtherCAT slave. This error status shows status of error in each slave in the actual system configuration. This variable array indicates slaves in which there are errors. Status is provided for each EtherCAT slave node address (1 to 512)*1. Refer to *Meanings of Error Status Bits* on page 7-8 for the meanings of the error status bits.				
Data type	Array [1..512] OF WORD*1			**Range of values**	16#0000 to 16#00F0
R/W access	R	**Retained**	Not retained.	**Network Publish**	Published.
Usage in user program	Possible.	**Related instructions**	Get EtherCAT Error Status • GetECError Reset EtherCAT Error • ResetECError		

*1. For the NX102 CPU Units, NX1P2 CPU Units, and NJ-series CPU Units, the node address is 1 to 192 and the data type is ARRAY [1..192] OF WORD.
For the NX502 CPU Units, the node address is 1 to 256 and the data type is Array [1..256] OF WORD.

Variable name	_EC_MacAdrErr				
Meaning	MAC Address Error			**Global/local**	Global
Function	TRUE if the MAC Address Error (14400000 hex) event occurred.				
Data type	BOOL			**Range of values**	TRUE or FALSE
R/W access	R	**Retained**	Not retained.	**Network Publish**	Published.
Usage in user program	Possible.	**Related instructions**	---		

Variable name	_EC_LanHwErr				
Meaning	Communications Controller Error			**Global/local**	Global
Function	TRUE if the Communications Controller Error (047C0000 hex) event occurred.				
Data type	BOOL			**Range of values**	TRUE or FALSE
R/W access	R	**Retained**	Not retained.	**Network Publish**	Published.
Usage in user program	Possible.	**Related instructions**	---		

Variable name	_EC_LinkOffErr				
Meaning	Link OFF Error			**Global/local**	Global
Function	TRUE if the Link OFF Error (84200000 hex) event occurred.				
Data type	BOOL			**Range of values**	TRUE or FALSE
R/W access	R	**Retained**	Not retained.	**Network Publish**	Published.
Usage in user program	Possible.	**Related instructions**	Reset EtherCAT Error • ResetECError		

Variable name	_EC_NetCfgErr				
Meaning	Network Configuration Information Error			**Global/local**	Global
Function	TRUE if the Network Configuration Information Error (34400000 hex) event occurred.				
Data type	BOOL			**Range of values**	TRUE or FALSE
R/W access	R	**Retained**	Not retained.	**Network Publish**	Published.
Usage in user program	Possible.	**Related instructions**	---		

Variable name	_EC_NetCfgCmpErr				
Meaning	Network Configuration Verification Error			**Global/local**	Global
Function	TRUE if one of the following events occurred. • Network Configuration Verification Error (84220000 hex) • Network Configuration Verification Error (Slave Unconnected) (84380000 hex) • Network Configuration Verification Error (Unnecessary Slave Connected) (84320003 hex) • Network Configuration Verification Error (Mismatched Slave) (84330004 hex) • Network Configuration Verification Error (Incorrect Ring Wiring) (843A0000 hex)				
Data type	BOOL			**Range of values**	TRUE or FALSE
R/W access	R	**Retained**	Not retained.	**Network Publish**	Published.
Usage in user program	Possible.	**Related instructions**	Reset EtherCAT Error • ResetECError		

Variable name	_EC_NetTopologyErr				
Meaning	Network Configuration Error			**Global/local**	Global
Function	TRUE if one of the following events occurred. • Network Configuration Error (84210000 hex) • Incorrect Wiring Detected (843C0000 hex)				
Data type	BOOL			**Range of values**	TRUE or FALSE
R/W access	R	**Retained**	Not retained.	**Network Publish**	Published.
Usage in user program	Possible.	**Related instructions**	Reset EtherCAT Error • ResetECError		

Variable name	_EC_PDCommErr				
Meaning	Process Data Communications Error			**Global/local**	Global
Function	TRUE if one of the following events occurred. • Process Data Communications Error (842C0000 hex) • Illegal Slave Disconnection Detected (84310002 hex) • Slave PDI WDT Error Detected (84340000 hex)				
Data type	BOOL			**Range of values**	TRUE or FALSE
R/W access	R	**Retained**	Not retained.	**Network Publish**	Published.
Usage in user program	Possible.	**Related instructions**	Reset EtherCAT Error • ResetECError		

Variable name	_EC_PDTimeoutErr				
Meaning	Process Data Reception Timeout Error			**Global/local**	Global
Function	TRUE if the Process Data Reception Timeout (842B0000 hex) event occurred.				
Data type	BOOL			**Range of values**	TRUE or FALSE
R/W access	R	**Retained**	Not retained.	**Network Publish**	Published.
Usage in user program	Possible.	**Related instructions**	Reset EtherCAT Error • ResetECError		

Variable name	_EC_PDSendErr				
Meaning	Process Data Transmission Error			**Global/local**	Global
Function	TRUE if the Process Data Transmission Error (84290000 hex) event occurred.				
Data type	BOOL			**Range of values**	TRUE or FALSE
R/W access	R	**Retained**	Not retained.	**Network Publish**	Published.
Usage in user program	Possible.	**Related instructions**	Reset EtherCAT Error • ResetECError		

Variable name	_EC_SlavAdrDupErr				
Meaning	Slave Node Address Duplicated Error			**Global/local**	Global
Function	TRUE if the Slave Node Address Duplicated (24200000 hex) event occurred.				
Data type	BOOL			**Range of values**	TRUE or FALSE
R/W access	R	**Retained**	Not retained.	**Network Publish**	Published.
Usage in user program	Possible.	**Related instructions**	Reset EtherCAT Error • ResetECError		

Variable name	_EC_SlavInitErr				
Meaning	Slave Initialization Error			**Global/local**	Global
Function	TRUE if one of the following events occurred. • Slave Initialization Error (84230000 hex) • Slave State Transition Failed (84300001 hex)				
Data type	BOOL			**Range of values**	TRUE or FALSE
R/W access	R	**Retained**	Not retained.	**Network Publish**	Published.
Usage in user program	Possible.	**Related instructions**	Reset EtherCAT Error • ResetECError		

Variable name	_EC_SlavAppErr				
Meaning	Slave Application Error			**Global/local**	Global
Function	TRUE if one of the following events occurred. • Slave Application Error (84280000 hex) • Slave AL Status Error Detected (84360000 hex)				
Data type	BOOL			**Range of values**	TRUE or FALSE
R/W access	R	**Retained**	Not retained.	**Network Publish**	Published.
Usage in user program	Possible.	**Related instructions**	Reset EtherCAT Error • ResetECError		

Variable name	_EC_MsgErr				
Meaning	EtherCAT Message Error			**Global/local**	Global
Function	TRUE if one of the following events occurred. • EtherCAT Message Error (842D0000 hex) • Illegal Mailbox Received (84350000 hex)				
Data type	BOOL			**Range of values**	TRUE or FALSE
R/W access	R	**Retained**	Not retained.	**Network Publish**	Published.
Usage in user program	Possible.	**Related instructions**	CoE messages (Read EtherCAT CoE SDO) • EC_CoESDORead CoE messages (Write EtherCAT CoE SDO) • EC_CoESDOWrite		

Variable name	_EC_SlavEmergErr				
Meaning	Emergency Message Detected			**Global/local**	Global
Function	TRUE if the Emergency Message Detected (64200000 hex) event occurred.				
Data type	BOOL			**Range of values**	TRUE or FALSE
R/W access	R	**Retained**	Not retained.	**Network Publish**	Published.
Usage in user program	Possible.	**Related instructions**	Reset EtherCAT Error • ResetECError		

Variable name	_EC_IndataInvalidErr[*1]				
Meaning	Input Process Data Invalid Error			**Global/local**	Global
Function	TRUE if the Input Process Data Invalid Error (842F0000 hex) event occurred.				
Data type	BOOL			**Range of values**	TRUE or FALSE
R/W access	R	**Retained**	Not retained.	**Network Publish**	Published.
Usage in user program	Possible.	**Related instructions**	Reset EtherCAT Error • ResetECError		

[*1] This system-defined variable was added for unit version 1.13 of the CPU Unit.

Variable name	_EC_CommErrTbl				
Meaning	Communications Error Slave Table			**Global/local**	Global
Function	Slaves are given in the table in the order of slave node addresses. The corresponding slave element is TRUE if the master detected an error for the slave.				
Data type	Array [1..512] OF BOOL[*1]			**Range of values**	TRUE or FALSE
R/W access	R	**Retained**	Not retained.	**Network Publish**	Published.
Usage in user program	Possible.	**Related instructions**	Reset EtherCAT Error • ResetECError		

[*1] For the NX102 CPU Units, NX1P2 CPU Units, and NJ-series CPU Units, the data type is ARRAY [1..192] OF BOOL. For the NX502 CPU Units, the data type is Array [1..256] OF BOOL.

Note The values of all system-defined variables that are related to errors in EtherCAT communications do not change until the cause of the error is removed and then the error in the Controller is reset with the troubleshooting functions of the Sysmac Studio or the ResetECError instruction.

Variable name	_EC_CycleExceeded				
Meaning	EtherCAT Communications Cycle Exceeded			**Global/local**	Global
Function	TRUE if the EtherCAT Communications Cycle Exceeded (34410000 hex) event occurred. **Note** You can use this system-defined variable only for NX-series CPU Units.				
Data type	BOOL			**Range of values**	TRUE or FALSE
R/W access	R	**Retained**	Not retained.	**Network Publish**	Published.
Usage in user program	Possible.	**Related instructions**	---		

● Functional Classification: EtherCAT Communications Status

Variable name	_EC_RegSlavTbl				
Meaning	Registered Slave Table			**Global/local**	Global
Function	This table indicates the slaves that are registered in the network configuration information. Slaves are given in the table in the order of slave node addresses. The element for a slave is TRUE if the corresponding slave is registered.				
Data type	Array [1..512] OF BOOL[*1]			**Range of values**	TRUE or FALSE
R/W access	R	**Retained**	Not retained.	**Network Publish**	Published.
Usage in user program	Possible.	**Related instructions**	---		

[*1] For the NX102 CPU Units, NX1P2 CPU Units, and NJ-series CPU Units, the data type is ARRAY [1..192] OF BOOL. For the NX502 CPU Units, the data type is Array [1..256] OF BOOL.

Variable name	_EC_EntrySlavTbl				
Meaning	Network Connected Slave Table			**Global/local**	Global
Function	This table indicates which slaves are connected to the network. Slaves are given in the table in the order of slave node addresses. The element for a slave is TRUE if the corresponding slave has entered the network.				
Data type	Array [1..512] OF BOOL[*1]			**Range of values**	TRUE or FALSE
R/W access	R	**Retained**	Not retained.	**Network Publish**	Published.
Usage in user program	Possible.	**Related instructions**	---		

*1. For the NX102 CPU Units, NX1P2 CPU Units, and NJ-series CPU Units, the data type is ARRAY [1..192] OF BOOL. For the NX502 CPU Units, the data type is Array [1..256] OF BOOL.

Variable name	_EC_MBXSlavTbl				
Meaning	Message Communications Enabled Slave Table			**Global/local**	Global
Function	This table indicates the slaves that can perform message communications. Slaves are given in the table in the order of slave node addresses. The element for a slave is TRUE if message communications are enabled for it (pre-operational, safe-operation, or operational state). **Note** Use this variable to confirm that message communications are possible for the relevant slave before you execute message communications with an EtherCAT slave.				
Data type	Array [1..512] OF BOOL[*1]			**Range of values**	TRUE or FALSE
R/W access	R	**Retained**	Not retained.	**Network Publish**	Published.
Usage in user program	Possible.	**Related instructions**	Disconnect EtherCAT Slave • EC_DisconnectSlave Connect EtherCAT Slave • EC_ConnectSlave		

*1. For the NX102 CPU Units, NX1P2 CPU Units, and NJ-series CPU Units, the data type is ARRAY [1..192] OF BOOL. For the NX502 CPU Units, the data type is Array [1..256] OF BOOL.

Variable name	_EC_PDSlavTbl				
Meaning	Process Data Communicating Slave Table			**Global/local**	Global
Function	This table indicates the slaves that are performing process data communications. Slaves are given in the table in the order of slave node addresses. The element for a slave is TRUE if process data of the corresponding slave is enabled (operational) for both slave inputs and outputs. **Note** Use this variable to confirm that the data for the relevant slave is valid before controlling an EtherCAT slave.				
Data type	Array [1..512] OF BOOL[*1]			**Range of values**	TRUE or FALSE
R/W access	R	**Retained**	Not retained.	**Network Publish**	Published.
Usage in user program	Possible.	**Related instructions**	Disconnect EtherCAT Slave • EC_DisconnectSlave Connect EtherCAT Slave • EC_ConnectSlave		

*1. For the NX102 CPU Units, NX1P2 CPU Units, and NJ-series CPU Units, the data type is ARRAY [1..192] OF BOOL. For the NX502 CPU Units, the data type is Array [1..256] OF BOOL.

Variable name	_EC_DisconnSlavTbl				
Meaning	Disconnected Slave Table			**Global/local**	Global
Function	Slaves are given in the table in the order of slave node addresses. The element for a slave is TRUE if the corresponding slave was disconnected.				
Data type	Array [1..512] OF BOOL[*1]			**Range of values**	TRUE or FALSE
R/W access	R	**Retained**	Not retained.	**Network Publish**	Published.
Usage in user program	Possible.	**Related instructions**	Disconnect EtherCAT Slave • EC_DisconnectSlave Connect EtherCAT Slave • EC_ConnectSlave		

[*1] For the NX102 CPU Units, NX1P2 CPU Units, and NJ-series CPU Units, the data type is ARRAY [1..192] OF BOOL. For the NX502 CPU Units, the data type is Array [1..256] OF BOOL.

Variable name	_EC_DisableSlavTbl				
Meaning	Disabled Slave Table			**Global/local**	Global
Function	Slaves are given in the table in the order of slave node addresses. The element for a slave is TRUE if the corresponding slave is disabled.				
Data type	Array [1..512] OF BOOL[*1]			**Range of values**	TRUE or FALSE
R/W access	R	**Retained**	Not retained.	**Network Publish**	Published.
Usage in user program	Possible.	**Related instructions**	---		

[*1] For the NX102 CPU Units, NX1P2 CPU Units, and NJ-series CPU Units, the data type is ARRAY [1..192] OF BOOL. For the NX502 CPU Units, the data type is Array [1..256] OF BOOL.

Variable name	_EC_PDActive				
Meaning	Process Data Communications Status			**Global/local**	Global
Function	TRUE when process data communications are performed with all slaves[*1].				
Data type	BOOL			**Range of values**	TRUE or FALSE
R/W access	R	**Retained**	Not retained.	**Network Publish**	Published.
Usage in user program	Possible.	**Related instructions**	Disconnect EtherCAT Slave • EC_DisconnectSlave Connect EtherCAT Slave • EC_ConnectSlave		

[*1] Disabled slaves are not included.

Variable name	_EC_PktMonStop				
Meaning	Packet Monitoring Stopped			**Global/local**	Global
Function	TRUE when packet monitoring is stopped.				
Data type	BOOL			**Range of values**	TRUE or FALSE
R/W access	R	**Retained**	Not retained.	**Network Publish**	Published.
Usage in user program	Possible.	**Related instructions**	Stop Packet Monitor • EC_StopMon Start Packet Monitor • EC_StartMon		

Variable name	_EC_LinkStatus				
Meaning	Link Status			**Global/local**	Global
Function	TRUE if the Communications Controller link status is Link ON.				
Data type	BOOL			**Range of values**	TRUE or FALSE
R/W access	R	**Retained**	Not retained.	**Network Publish**	Published.
Usage in user program	Possible.	**Related instructions**	---		

Variable name	_EC_PktSaving				
Meaning	Saving Packet Data File			**Global/local**	Global
Function	Shows whether a packet data file is being saved. TRUE: Packet data file being saved. FALSE: Packet data file not being saved.				
Data type	BOOL			**Range of values**	TRUE or FALSE
R/W access	R	**Retained**	Not retained.	**Network Publish**	Published.
Usage in user program	Possible.	**Related instructions**	Saving Packet Data File • EC_SaveMon		

Variable name	_EC_InDataInvalid				
Meaning	Input Data Invalid			**Global/local**	Global
Function	TRUE when process data communications established in the primary periodic task are not normal and the input data is not valid.				
Data type	BOOL			**Range of values**	TRUE or FALSE
R/W access	R	**Retained**	Not retained.	**Network Publish**	Published.
Usage in user program	Possible.	**Related instructions**	---		

Note 1. All system-defined variables that are related to the status of EtherCAT communications give the current status.

Note 2. The variable temporarily changes to TRUE if the EC_DisconnectSlave (Disconnect EtherCAT Slave) instruction, EC_ConnectSlave (Connect EtherCAT Slave) instruction, or EC_ChangeEnableSetting (Enable/Disable EtherCAT Slave) instruction is executed.

Variable name	_EC_InData1Invalid				
Meaning	Input Data1 Invalid			**Global/local**	Global
Function	TRUE when process data communications established in the primary periodic task are not normal and the input data is not valid. **Note** You can use this system-defined variable only for NX-series CPU Units.				
Data type	BOOL			**Range of values**	TRUE or FALSE
R/W access	R	**Retained**	Not retained.	**Network Publish**	Published.
Usage in user program	Possible.	**Related instructions**	---		

Note 1. All system-defined variables that are related to the status of EtherCAT communications give the current status.

Note 2. The variable temporarily changes to TRUE if the EC_DisconnectSlave (Disconnect EtherCAT Slave) instruction, EC_ConnectSlave (Connect EtherCAT Slave) instruction, or EC_ChangeEnableSetting (Enable/Disable EtherCAT Slave) instruction is executed.

Variable name	_EC_InData2Invalid				
Meaning	Input Data2 Invalid			Global/local	Global
Function	TRUE when process data communications established in the priority-5 periodic task are not normal and the input data is not valid. **Note** You can use this system-defined variable only for NX-series CPU Units. **Note** This variable is always TRUE for the NX502 CPU Units, NX102 CPU Units, and NX1P2 CPU Units.				
Data type	BOOL			Range of values	TRUE or FALSE
R/W access	R	Retained	Not retained.	Network Publish	Published.
Usage in user program	Possible.	Related instructions	---		

Note 1. All system-defined variables that are related to the status of EtherCAT communications give the current status.

Note 2. The variable temporarily changes to TRUE if the EC_DisconnectSlave (Disconnect EtherCAT Slave) instruction, EC_ConnectSlave (Connect EtherCAT Slave) instruction, or EC_ChangeEnableSetting (Enable/Disable EtherCAT Slave) instruction is executed.

Variable name	_EC_RingBreaking[*1]				
Meaning	Ring Disconnection			Global/local	Global
Function	TRUE when all slaves in the ring topology except for disabled slaves are connected and if there is only one point of disconnection in the communications cables in the ring topology.				
Data type	BOOL			Range of values	TRUE or FALSE
R/W access	R	Retained	Not retained.	Network Publish	Published.
Usage in user program	Possible.	Related instructions	---		

[*1] This system-defined variable was added for unit version 1.40 of the CPU Unit.

Variable name	_EC_RingBreakNodeAdr[*1]				
Meaning	Slave Node Address Before Ring Disconnection			Global/local	Global
Function	When the *_EC_RingBreaking* (Ring Disconnection) system-defined variable is TRUE, the slave node address before point of disconnection is stored. When the *_EC_RingBreaking* (Ring Disconnection) system-defined variable is FALSE, “0” is stored.				
Data type	UINT			Range of values	0 to maximum number of node addresses
R/W access	R	Retained	Not retained.	Network Publish	Published.
Usage in user program	Possible.	Related instructions	---		

[*1] This system-defined variable was added for unit version 1.40 of the CPU Unit.

● Functional Classification: EtherCAT Communications Diagnosis/Statistics Log

Variable name	_EC_StatisticsLogEnable[*1]				
Meaning	Diagnosis/Statistics Log Enable			Global/local	Global
Function	Changes to TRUE when the diagnosis/statistics log is started. Changes to FALSE when the diagnosis/statistics log is ended.				
Data type	BOOL			Range of values	TRUE or FALSE
R/W access	RW	Retained	Not retained.	Network Publish	Published.
Usage in user program	Possible.	Related instructions	---		

[*1] This system-defined variable was added for unit version 1.11 of the CPU Unit.

Variable name	_EC_StatisticsLogCycleSec[*1]				
Meaning	Diagnosis/Statistics Log Cycle			**Global/local**	Global
Function	Specifies the interval to write the diagnostic and statistical information of the diagnosis/statistics log in units of seconds. When 0 is specified, the diagnostic and statistical information is written only once when the diagnosis/statistics log is ended. **Note** The write interval does not change even if you change the value of this system-defined variable while the diagnosis/statistics log operation is in progress.				
Data type	UINT			**Range of values**	0, or 30 to 1800
R/W access	RW	**Retained**	Not retained.	**Network Publish**	Published.
Usage in user program	Possible.	**Related instructions**	---		

[*1] This system-defined variable was added for unit version 1.11 of the CPU Unit.

Variable name	_EC_StatisticsLogBusy[*1]				
Meaning	Diagnosis/Statistics Log Busy			**Global/local**	Global
Function	TRUE while the diagnosis/statistics log operation is in progress.				
Data type	BOOL			**Range of values**	TRUE or FALSE
R/W access	R	**Retained**	Not retained.	**Network Publish**	Published.
Usage in user program	Possible.	**Related instructions**	---		

[*1] This system-defined variable was added for unit version 1.11 of the CPU Unit.

Variable name	_EC_StatisticsLogErr[*1]				
Meaning	Diagnosis/Statistics Log Error			**Global/local**	Global
Function	TRUE when the diagnosis/statistics log failed to start or it is impossible to write into the log. The value of this flag is determined when *_EC_StatisticsLogBusy* (Diagnosis/Statistics Log Busy) changes to FALSE after the diagnosis/statistics log operation is started. The error end is caused by the following. • Another records cannot be added in the log file because the capacity of the SD Memory Card is fully used. • The SD Memory Card is write-protected. • There is no SD Memory Card. • The function cannot be started because the value specified for *_EC_StatisticsLogCycleSec* (Diagnosis/Statistics Log Cycle) is invalid.				
Data type	BOOL			**Range of values**	TRUE or FALSE
R/W access	R	**Retained**	Not retained.	**Network Publish**	Published.
Usage in user program	Possible.	**Related instructions**	---		

[*1] This system-defined variable was added for unit version 1.11 of the CPU Unit.

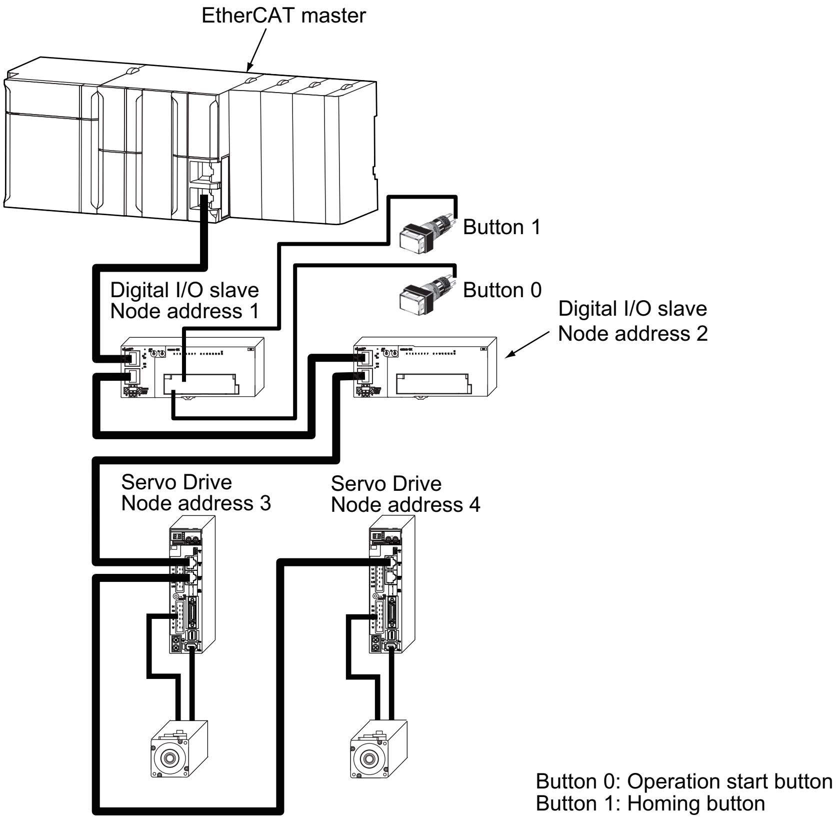
8

Example of Operations for EtherCAT Communications

This section provides a series of example operations for when an NJ/NX-series CPU Unit is connected to slaves.

8-1 Example of Operations for EtherCAT Communications

8-1-1 System Configuration Example



There are no restrictions on the order of node addresses.

8-1-2 Wiring and Settings

Wiring

- Install the Controller and slaves.
- Connect communications cables to the EtherCAT master and slaves.
- Connect the power supply.

Settings

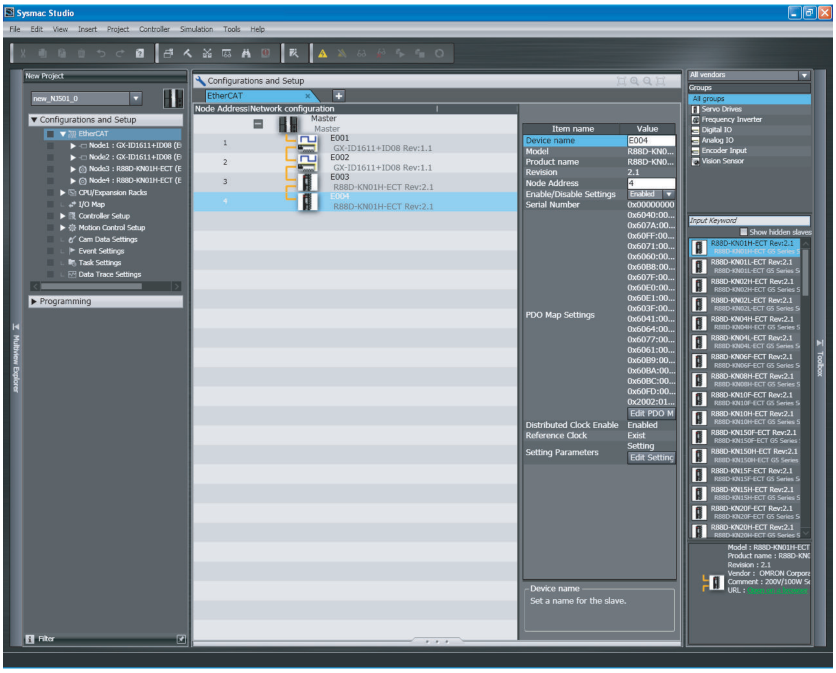
Set the node address for each slave.

8-1-3 Setting the EtherCAT Network Configuration

Start the Sysmac Studio and make the following settings.

● Creating the EtherCAT Network Configuration

Use the EtherCAT Configuration Editor to create the slave configuration.



Node Address	Network configuration
	Master Master
1	E001 GX-ID1611+ID08 Rev:1.1
2	E002 GX-ID1611+ID08 Rev:1.1
3	E003 R88D-KN01H-ECT Rev:2.1
4	E004 R88D-KN01H-ECT Rev:2.1

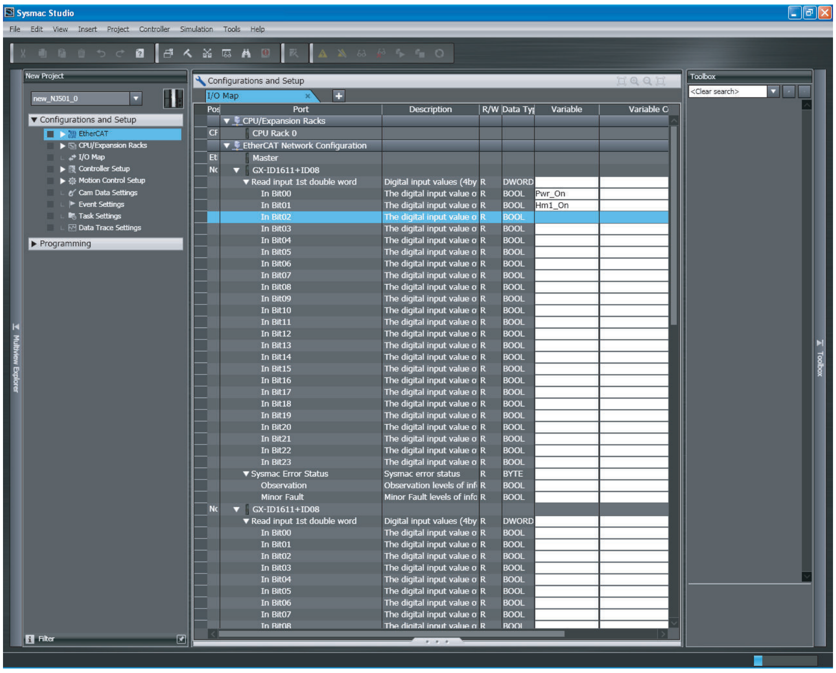
Item name	Value
Device name	E004
Model	R88D-KN0...
Product name	R88D-KN0...
Revision	2.1
Node Address	4
Enable/Disable Settings	Enabled
Serial Number	0x00000000
PDO Map Settings	0x6040:00... 0x607A:00... 0x60FF:00... 0x6071:00... 0x6060:00... 0x60B8:00... 0x607F:00... 0x60E0:00... 0x60E1:00... 0x603F:00... 0x6041:00... 0x6064:00... 0x6077:00... 0x6061:00... 0x60B9:00... 0x60BA:00... 0x60BC:00... 0x60FD:00... 0x2002:01... Edit PDO M
Distributed Clock Enable	Enabled
Reference Clock	Exist
Setting Parameters	Setting Edit Setting

Device name
Set a name for the slave.

In this example, digital I/O slaves are set to node addresses 1 and 2 and Servo Drives are set to node addresses 3 and 4.

● Assigning Device Variables to Digital I/O Slaves (Node Addresses 1 and 2)

Use the I/O Map to assign device variables to the I/O ports of the slaves.

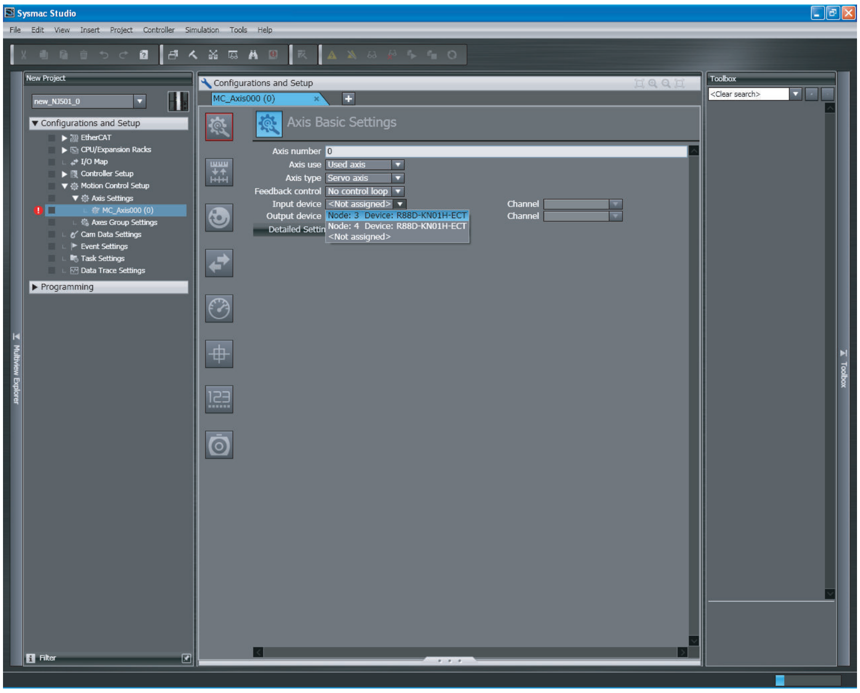


Pos	Port	Description	R/W	Data Typ	Variable	Variable C
	▼ CPU/Expansion Racks					
CF	CPU Rack 0					
	▼ EtherCAT Network Configuration					
Et	Master					
Nc	▼ GX-ID1611+ID08					
	▼ Read input 1st double word	Digital input values (4by	R	DWORD		
	In Bit00	The digital input value o	R	BOOL	Pwr_On	
	In Bit01	The digital input value o	R	BOOL	Hm1_On	
	In Bit02	The digital input value o	R	BOOL		
	In Bit03	The digital input value o	R	BOOL		
	In Bit04	The digital input value o	R	BOOL		
	In Bit05	The digital input value o	R	BOOL		
	In Bit06	The digital input value o	R	BOOL		
	In Bit07	The digital input value o	R	BOOL		
	In Bit08	The digital input value o	R	BOOL		
	In Bit09	The digital input value o	R	BOOL		
	In Bit10	The digital input value o	R	BOOL		
	In Bit11	The digital input value o	R	BOOL		
	In Bit12	The digital input value o	R	BOOL		
	In Bit13	The digital input value o	R	BOOL		
	In Bit14	The digital input value o	R	BOOL		
	In Bit15	The digital input value o	R	BOOL		
	In Bit16	The digital input value o	R	BOOL		
	In Bit17	The digital input value o	R	BOOL		
	In Bit18	The digital input value o	R	BOOL		
	In Bit19	The digital input value o	R	BOOL		
	In Bit20	The digital input value o	R	BOOL		
	In Bit21	The digital input value o	R	BOOL		
	In Bit22	The digital input value o	R	BOOL		
	In Bit23	The digital input value o	R	BOOL		
	▼ Sysmac Error Status	Sysmac error status	R	BYTE		
	Observation	Observation levels of inf	R	BOOL		
	Minor Fault	Minor Fault levels of info	R	BOOL		
Nc	▼ GX-ID1611+ID08					
	▼ Read input 1st double word	Digital input values (4by	R	DWORD		
	In Bit00	The digital input value o	R	BOOL		
	In Bit01	The digital input value o	R	BOOL		
	In Bit02	The digital input value o	R	BOOL		
	In Bit03	The digital input value o	R	BOOL		
	In Bit04	The digital input value o	R	BOOL		
	In Bit05	The digital input value o	R	BOOL		
	In Bit06	The digital input value o	R	BOOL		
	In Bit07	The digital input value o	R	BOOL		
	In Bit08	The digital input value o	R	BOOL		

In this example, *Pwr_On* is assigned to bit 0 of slave 1 and *Hm1_On* is assigned to bit 1 of slave 1.

● Axis Settings for the Servo Drives (Node Addresses 3 and 4)

Add an axis to the Motion Control Setup and then assign the Servo Drive with node address 3 to the axis to set the axis.



Axis number 0
Axis use Used axis
Axis type Servo axis
Feedback control No control loop
Input device <Not assigned>
Output device
Node: 3 Device: R88D-KN01H-ECT
Node: 4 Device: R88D-KN01H-ECT
<Not assigned>
Detailed Settin
Channel
Channel

In the same way, add an axis and assign the Servo Drive with node address 4 to it.

● Setting EtherCAT Master Parameters

Set the parameters for the EtherCAT master from the EtherCAT master settings.

● Setting Slave Parameters

Set the slave parameters from the EtherCAT configuration slave settings.

If a communications error prevents the slaves from receiving signals from the EtherCAT master, the slave outputs are processed according to settings in the slave.

● Assigning Tasks to Slaves

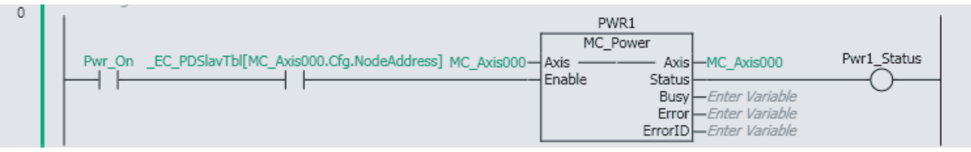
Use **Task Settings** to assign tasks to slaves.

- Set **Period/Execution Conditions** for each task from **Task Settings**.
- Set names of the tasks that are assigned to the slaves from **I/O Control Task Settings**.

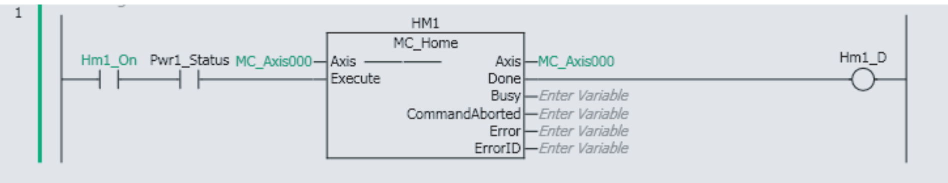
8-1-4 Programming

● Programming

When the operation start button is pressed, the *Pwr_On* variable changes to TRUE and axis control is enabled. (See rung 0.)



When the homing button is pressed, the *Hm1_On* variable changes to TRUE and homing is performed. (See rung 1.)



● Assigning Programs to Tasks

Use **Task Settings** to assign programs to tasks and set the program execution order.
Assign programs to tasks and set the program execution order from **Program Assignment Settings**.

8-1-5 Offline Debugging

You can use the Simulator to check the program and task execution times with offline debugging.

8-1-6 Turning the Power ON

Turn ON the following power supplies.

- Slave unit power supply (The PWR indicator on the slave will light when the power supply turns ON.)
- Slave I/O power supply
- Controller power supply

8-1-7 Online Debugging

Compare and merge the network configuration that was set on the Sysmac Studio and the actual configuration.

8-1-8 Downloading the Network Configuration Information and the User Program

Download the network configuration information and the user program that were created on the Sysmac Studio to the Controller.

Note Use the synchronization operation of the Sysmac Studio to download the data.

8-1-9 Confirming the Start of Communications

Check to make sure that all registered slaves are participating in the network and that communications start.

Make sure that the master indicators are in the following status.

NET RUN indicator	Lit
NET ERR indicator	Not lit
LINK/ACT indicator (physical layer LINK)	Flashing

Make sure that the status indicators on all slaves are in the following status.

PWR indicator	Lit
RUN indicator	Lit

ERR indicator	Not lit
L/A IN (physical layer LINK inputs)	Flashing
L/A OUT (physical layer LINK outputs)	Flashing (Not lit on the last slave.)

9

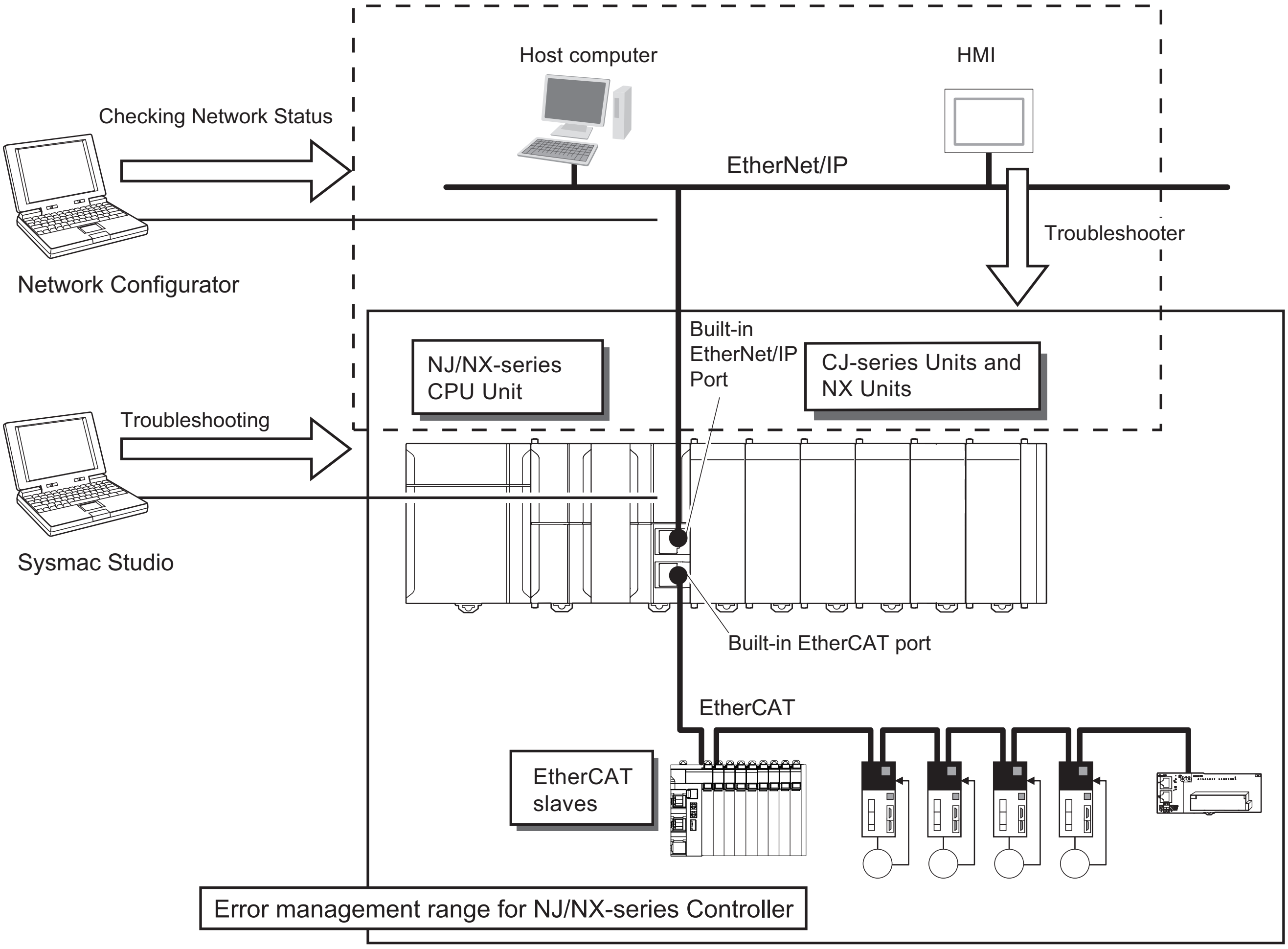
Troubleshooting

This section describes overview of troubleshooting, network diagnosis procedure using diagnostic and statistical information, functions of the Sysmac Studio and the CPU Unit for acquiring diagnostic and statistical information, functions of the Sysmac Studio for identifying error slaves and error causes, and precautions and methods of replacing slaves during communications.

9-1 Overview of Troubleshooting

You manage all of the errors that occur on the NJ/NX-series Controller as events. This allows you to see what errors have occurred and find corrections for them with the same methods for the entire range of errors that is managed (i.e., CPU Unit, NX Units, NX-series Slave Terminals, EtherCAT slaves[*1], and CJ-series Units).

[*1] Only Sysmac devices are supported.



You can use the troubleshooting functions of the Sysmac Studio or the Troubleshooter on an HMI to quickly check for errors that have occurred and find corrections for them.
Refer to the *NJ/NX-series Troubleshooting Manual (Cat. No. W503)* for error types and details, specific corrections when errors occur, and troubleshooting information on the entire NJ/NX-series Controller.

9-1-1 How to Check for Errors

Refer to *Checking for Non-fatal Errors* in the *NJ/NX-series Troubleshooting Manual (Cat. No. W503)* for checking for errors.

9-1-2 Errors Related to the EtherCAT Master Function Module

Refer to *Errors Related to the EtherCAT Master Function Module* in the *NJ/NX-series Troubleshooting Manual (Cat. No. W503)* for errors related to the EtherCAT Master Function Module.

9-2 Troubleshooting

This section describes errors (events) that can occur and the corrections for them, network diagnosis procedure using diagnostic and statistical information, functions of the Sysmac Studio and the CPU Unit for acquiring diagnostic and statistical information, and functions of the Sysmac Studio for identifying error slaves and error causes.

9-2-1 Error Table

Refer to *Error Tables* in *Errors in the EtherCAT Master Function Module* in the *NJ/NX-series Troubleshooting Manual (Cat. No. W503)* for tables of errors (events) that occur in the EtherCAT Master Function Module.

9-2-2 Error Descriptions

Refer to *Error Descriptions* in *Errors in the EtherCAT Master Function Module* in the *NJ/NX-series Troubleshooting Manual (Cat. No. W503)* for description of errors (events) that occur in the EtherCAT Master Function Module.

9-2-3 Resetting Errors

Refer to *Resetting Errors in the EtherCAT Master Function Module* in *Errors Related to the EtherCAT Master Function Module* in the *NJ/NX-series Troubleshooting Manual (Cat. No. W503)* for how to reset errors.

9-2-4 Diagnostic and Statistical Information

The diagnostic and statistical information provides statistics on the number of communications frames sent and received by the EtherCAT master and EtherCAT slaves as well as the number of frames for which errors were detected.
You can use it to diagnosis the EtherCAT network line qualify for the following:

- Confirming that the EtherCAT network was correctly installed during a test run
- Finding the causes of communications errors that occur during normal operation
- Checking the EtherCAT network line quality during normal operation

EtherCAT Network Diagnostic Procedure

You can diagnose the EtherCAT network with the diagnostic and statistical information for the master and slaves. The procedure to diagnose the EtherCAT network is given below.
If the diagnostic results show that the EtherCAT network is not operating normally, you can find the location of the error.

An outline procedure from EtherCAT network diagnosis through correction is given below.

1 Acquire the diagnostic and statistical information for the master and slaves.

2 Check for errors in the trends shown in the master diagnostic and statistical information.

3 Find the locations of the errors with trends in the slave diagnostic and statistical information.

4 Implement corrections for the error locations that you found.

5 Confirm status after implementation of the correction.

● Step 1: Acquiring Diagnostic and Statistical Information for Master and Slaves

There are two methods to acquire the diagnostic and statistical information for the master and slaves. The following table describes each method. Use either of the methods to acquire the diagnostic and statistical information for the master and slaves.

Acquisition method	Description	Reference
Using the diagnostic and statistical information display of Sysmac Studio	Use the Sysmac Studio to acquire the diagnostic and statistical information. You can save the acquired diagnostic and statistical information in the computer.	Refer to *Diagnostic and Statistical Information Display of Sysmac Studio* on page 9-11.
Using the diagnosis/statistics log of CPU Unit	The CPU Unit acquires the diagnostic and statistical information periodically. The acquired diagnostic and statistical information is saved in an SD Memory Card that is mounted on the CPU Unit.	Refer to *Diagnosis/Statistics Log of CPU Unit* on page 9-16.

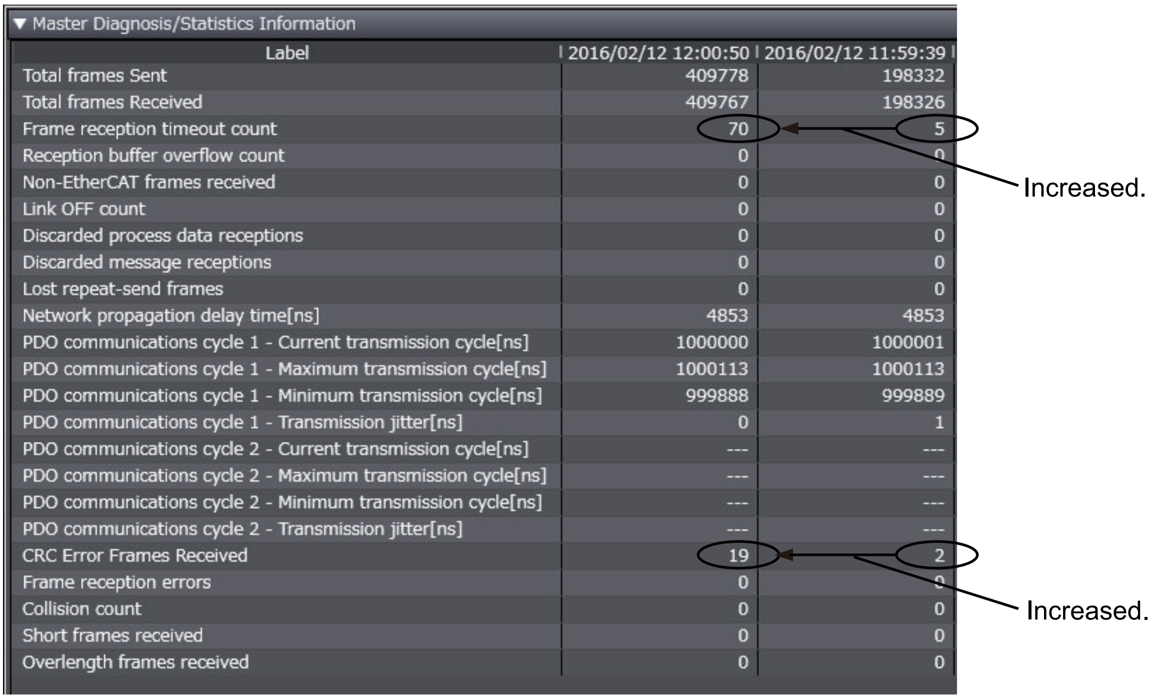
Precautions for Correct Use

When the Sysmac Studio's diagnostic and statistical information display is used, the maximum number of error frames recorded for the slave diagnostic and statistical information is 255. If the number of error frames exceeds 255, increasing trends of the number of error frames cannot be recognized. If the number of error frames for the slave diagnostic and statistical information is assumed to exceed 255, execute the clear operation for the slave diagnostic and statistical information before acquiring the slave diagnostic and statistical information.

● Step 2: Checking for Errors in Trends in Master Diagnostic and Statistical Information

Check for trends in the items in the acquired master diagnostic and statistical information to diagnose errors in the EtherCAT network.

Example of the Master Diagnosis/Statistics Tab Page of Sysmac Studio



▼ Master Diagnosis/Statistics Information

Label	2016/02/12 12:00:50	2016/02/12 11:59:39
Total frames Sent	409778	198332
Total frames Received	409767	198326
Frame reception timeout count	70	5
Reception buffer overflow count	0	0
Non-EtherCAT frames received	0	0
Link OFF count	0	0
Discarded process data receptions	0	0
Discarded message receptions	0	0
Lost repeat-send frames	0	0
Network propagation delay time[ns]	4853	4853
PDO communications cycle 1 - Current transmission cycle[ns]	1000000	1000001
PDO communications cycle 1 - Maximum transmission cycle[ns]	1000113	1000113
PDO communications cycle 1 - Minimum transmission cycle[ns]	999888	999889
PDO communications cycle 1 - Transmission jitter[ns]	0	1
PDO communications cycle 2 - Current transmission cycle[ns]	---	---
PDO communications cycle 2 - Maximum transmission cycle[ns]	---	---
PDO communications cycle 2 - Minimum transmission cycle[ns]	---	---
PDO communications cycle 2 - Transmission jitter[ns]	---	---
CRC Error Frames Received	19	2
Frame reception errors	0	0
Collision count	0	0
Short frames received	0	0
Overlength frames received	0	0

Increased.

Increased.

If the value of the frame reception timeout count or number of CRC error frames received increases, then the EtherCAT network is not operating normally.
If a certain number of the frame reception timeout count or a certain number of CRC error frames received is detected, the EtherCAT network may not be operating normally.

If there is an error or possibility of an error in the EtherCAT network, find the error location by performing *Step 3: Finding Locations of Errors with Trends in Slave Diagnostic and Statistical Information* on page 9-5.

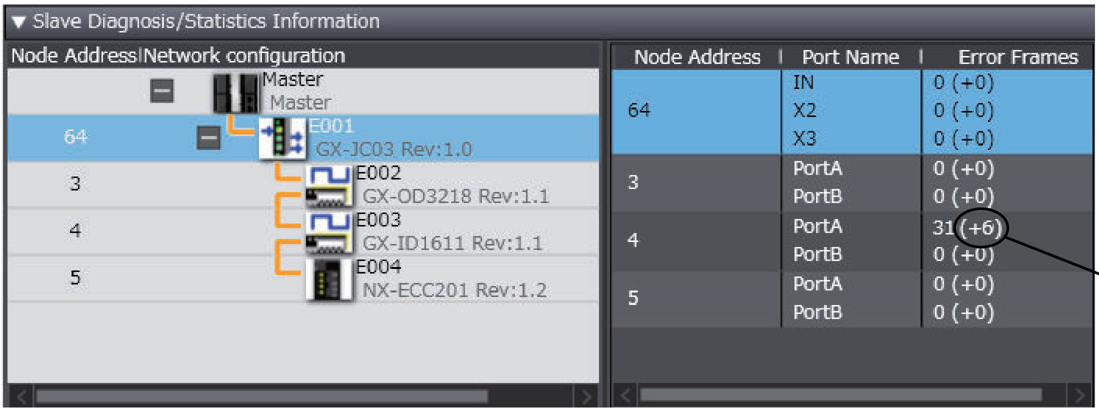
Additional Information

A certain number of the frame reception timeout count or a certain number of CRC error frames received is also detected if a power OFF or disconnection occurs in an EtherCAT slave.

● Step 3: Finding Locations of Errors with Trends in Slave Diagnostic and Statistical Information

You can check for trends of values in the slave diagnostic and statistical information to find the locations of the errors.

Example of the Slave Diagnosis/Statistics Tab Page of Sysmac Studio



▼ Slave Diagnosis/Statistics Information

Node Address	Network configuration
	Master Master
64	E001 GX-JC03 Rev:1.0
3	E002 GX-OD3218 Rev:1.1
4	E003 GX-ID1611 Rev:1.1
5	E004 NX-ECC201 Rev:1.2

Node Address	Port Name	Error Frames
64	IN	0 (+0)
	X2	0 (+0)
	X3	0 (+0)
3	PortA	0 (+0)
	PortB	0 (+0)
4	PortA	31 (+6)
	PortB	0 (+0)
5	PortA	0 (+0)
	PortB	0 (+0)

Increased.

The following are the points to check to find error locations based on the number of error frames.

- A certain number of error frames is detected.
- The number of error frames is increased compared to the value acquired last time.
- "Failed" is displayed for the number of error frames.

If a certain number of error frames is detected for more than one port, start finding error locations from the port with the highest number of error frames.

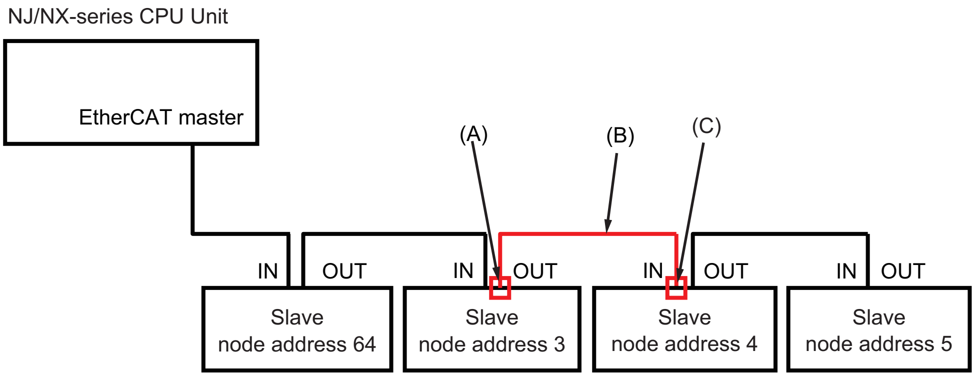
The error locations that you find will change depending on the configuration of EtherCAT slave connection. Refer to the network configuration diagram to find error locations.

This section explains how to find error locations with two examples of EtherCAT network configurations in which the EtherCAT slave connection configurations are different.

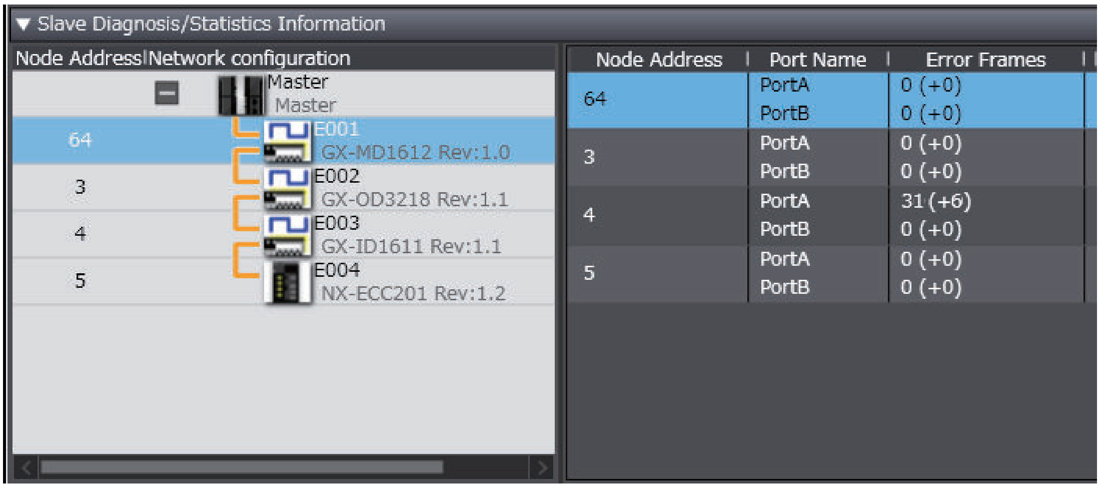
Examples of Finding Error Locations

Example 1: Network Configuration Where an EtherCAT Junction Slave Is Not Used

Network Configuration Diagram



Slave Diagnosis/Statistics Tab Page



▼ Slave Diagnosis/Statistics Information

Node Address	Network configuration
	Master Master
64	E001 GX-MD1612 Rev:1.0
3	E002 GX-OD3218 Rev:1.1
4	E003 GX-ID1611 Rev:1.1
5	E004 NX-ECC201 Rev:1.2

Node Address	Port Name	Error Frames
64	PortA	0 (+0)
	PortB	0 (+0)
3	PortA	0 (+0)
	PortB	0 (+0)
4	PortA	31 (+6)
	PortB	0 (+0)
5	PortA	0 (+0)
	PortB	0 (+0)

The number of error frames for the input port (PortA) for node address 4 is 31, so you can see that error frames were received on the input port for node address 4.

Therefore, you can assume that there is a problem between the output port (PortB) for node address 3 and the input port for node address 4.

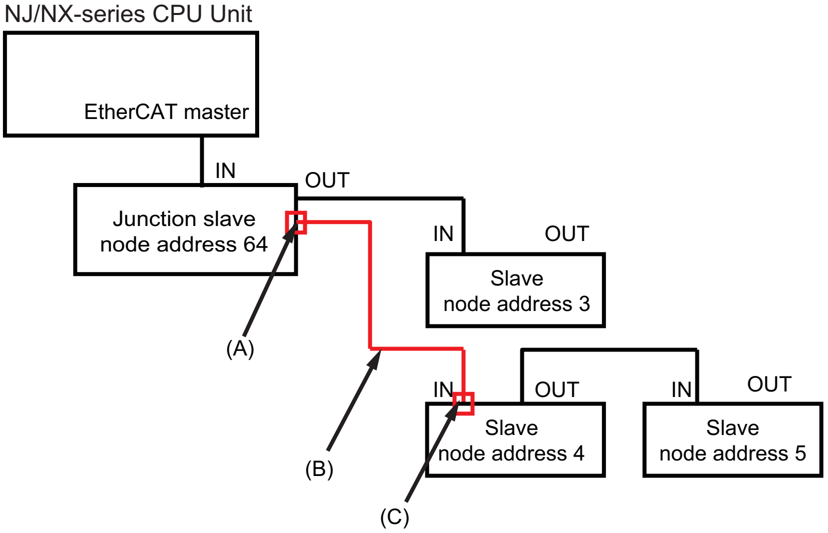
This corresponds to location (A) (B) or (C) in the network configuration diagram.

Concretely, you can assume the following possible error locations.

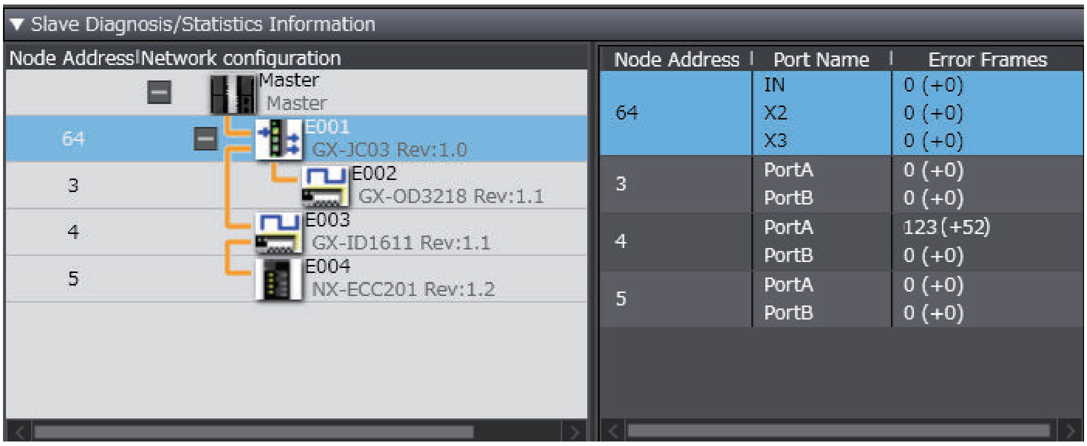
- The device at node address 3
- The cable between the output port at node address 3 and the input port at node address 4 or the connectors at those ports
- The device at node address 4

Example 2: Network Configuration Where an EtherCAT Junction Slave Is Used

Network Configuration Diagram



Slave Diagnosis/Statistics Tab Page



Node Address	Network configuration
	Master Master
64	E001 GX-JC03 Rev:1.0
3	E002 GX-OD3218 Rev:1.1
4	E003 GX-ID1611 Rev:1.1
5	E004 NX-ECC201 Rev:1.2

Node Address	Port Name	Error Frames
64	IN	0 (+0)
	X2	0 (+0)
	X3	0 (+0)
3	PortA	0 (+0)
	PortB	0 (+0)
4	PortA	123 (+52)
	PortB	0 (+0)
5	PortA	0 (+0)
	PortB	0 (+0)

The number of error frames for the input port (PortA) for node address 4 is 123, so you can see that error frames were received on the input port for node address 4.

Therefore, you can assume that there is a problem between the output port (X3) for node address 64 and the input port for node address 4.

This corresponds to location (A) (B) or (C) in the network configuration diagram.

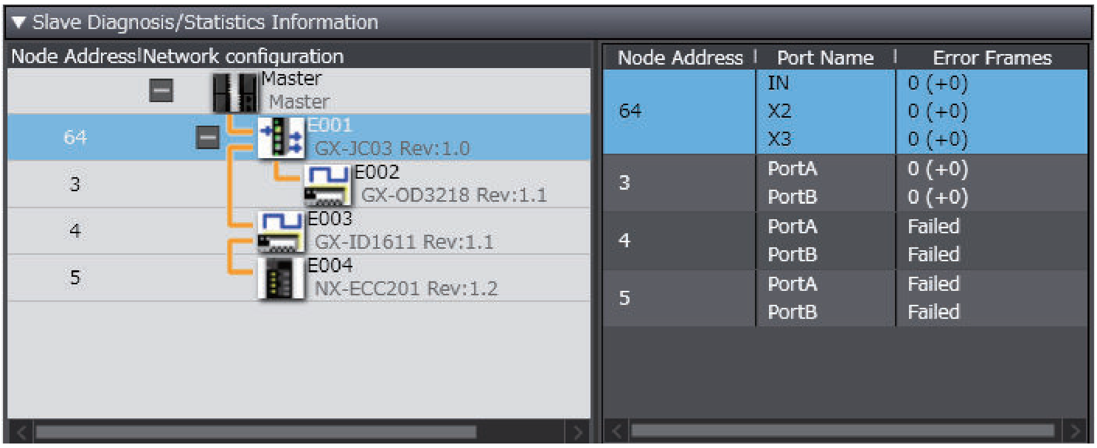
Concretely, you can assume the following possible error locations.

- The device at node address 64
- The cable between the output port at node address 64 and the input port at node address 4 or the connectors at those ports
- The device at node address 4

This is the procedure to find error locations when "Failed" is displayed for the number of error frames.

In the EtherCAT network configuration, "Failed" is sometimes displayed for more than one EtherCAT slave.

In the network configuration for example 2, assume that "Failed" is displayed for the number of error frames for node addresses 4 and 5.



▼ Slave Diagnosis/Statistics Information

Node Address	Network configuration
	Master Master
64	E001 GX-JC03 Rev:1.0
3	E002 GX-OD3218 Rev:1.1
4	E003 GX-ID1611 Rev:1.1
5	E004 NX-ECC201 Rev:1.2

Node Address	Port Name	Error Frames
64	IN	0 (+0)
	X2	0 (+0)
	X3	0 (+0)
3	PortA	0 (+0)
	PortB	0 (+0)
4	PortA	Failed
	PortB	Failed
5	PortA	Failed
	PortB	Failed

If "Failed" is displayed for the number of error frames, the EtherCAT slave currently cannot communicate. If an EtherCAT slave cannot communicate, all of the EtherCAT slaves connected after it may not be able to communicate. In the network configuration diagram of example 2, node address 5 is connected after node address 4.

Therefore, you can assume that there is an error in node address 4 that prevents communications.

Concretely, you can assume the following possible error locations.

- The device at node address 64
- The cable between the output port at node address 64 and the input port at node address 4 or the connectors at those ports
- The device at node address 4

● Step 4: Implementing Corrections for Error Locations

Implement corrections for the error locations that you found.

To check the position of the EtherCAT slave with an error in the network configuration, refer to the **Node Address|Network configuration** display under Slave Diagnostic and Statistical Information.

The following table provides corrections for the causes as assumed from the diagnostic and statistical information.

Master diagnosis and statistical information trend		Slave diagnosis and statistical information trend	Assumed error cause	Possible correction
The frame reception timeout count is increasing.	The number of CRC error frames received is not increasing.	"Failed" is displayed for the number of error frames.	The power is not supplied to the EtherCAT slave.	Supply the power to the EtherCAT slave.
			A connector on the Ethernet cable is disconnected, the contact is faulty, or parts are faulty.	Make sure the connector is mated correctly, or reconnect the connector.
			The Ethernet cable is broken or the specified cable was not used.	Replace the Ethernet cable.
			A general-purpose Ethernet hub is connected.	Replace the general-purpose Ethernet hub with an EtherCAT Junction Slave.
			The EtherCAT slave failed.	Replace the EtherCAT slave.
		The number of error frames is not increasing.	The EtherCAT master communications cycle is too short.	Go online with the Sysmac Studio, lengthen the task period (communications cycle), and set it in the CPU Unit again.
	The number of CRC error frames received is increasing.	"Failed" is displayed for the number of error frames and the number of error frames is increasing.	A connector on the Ethernet cable is disconnected, the contact is faulty, or parts are faulty.	Make sure the connector is mated correctly, or reconnect the connector.
			The EtherCAT slave failed.	Replace the EtherCAT slave.
		"Failed" is displayed for the number of error frames or the number of error frames is increasing.	There is noise.	Implement noise countermeasures.

The following is a description of the case which does not correspond to any of above conditions.
If a certain number of the frame reception timeout count and CRC error frames received for the master diagnostic and statistical information, as well as a certain number of error frames for the slave diagnostic and statistical information are detected but are not increasing, these numbers were detected temporarily due to any of the following error causes.

- A connector on the Ethernet cable is disconnected, the contact is faulty, or parts are faulty.
- There is noise.
- The Ethernet cable is broken or the specified cable was not used.

Implement corrections for the error locations that you found. Refer to the table above for what to do for corrections.

● Step 5: Confirming After Implementing Corrections

To confirm that the error locations were corrected, perform the procedure in *Step 2: Checking for Errors in Trends in Master Diagnostic and Statistical Information* on page 9-4 again to make sure the network is operating normally.

Diagnostic and Statistical Information Display of Sysmac Studio

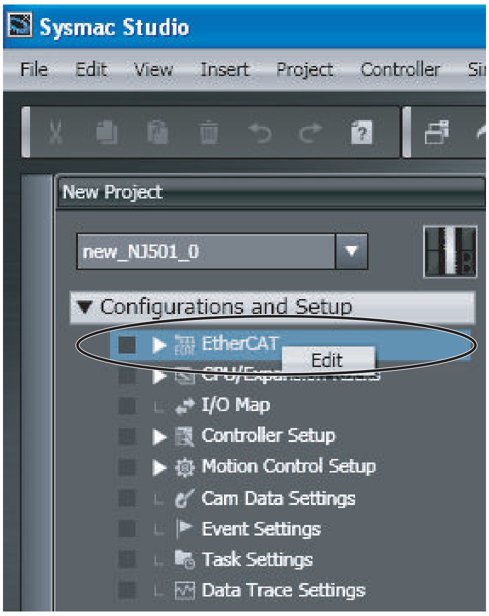
This section describes how to activate the diagnostic and statistical information display of Sysmac Studio. The functions of the operation buttons and displayed items are also explained.

● Activation

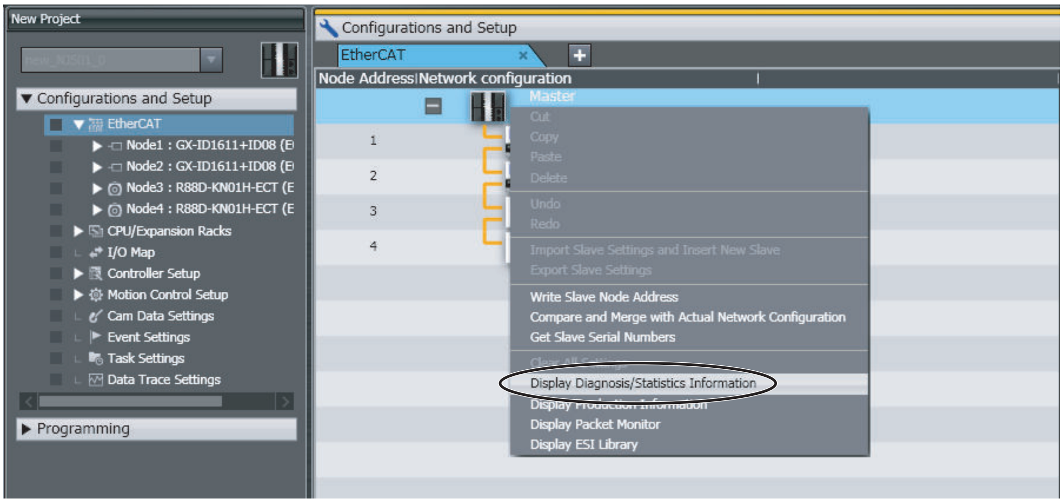
Activate the diagnostic and statistical information display of Sysmac Studio as follows.

1 Start the Sysmac Studio and go online with the Controller.

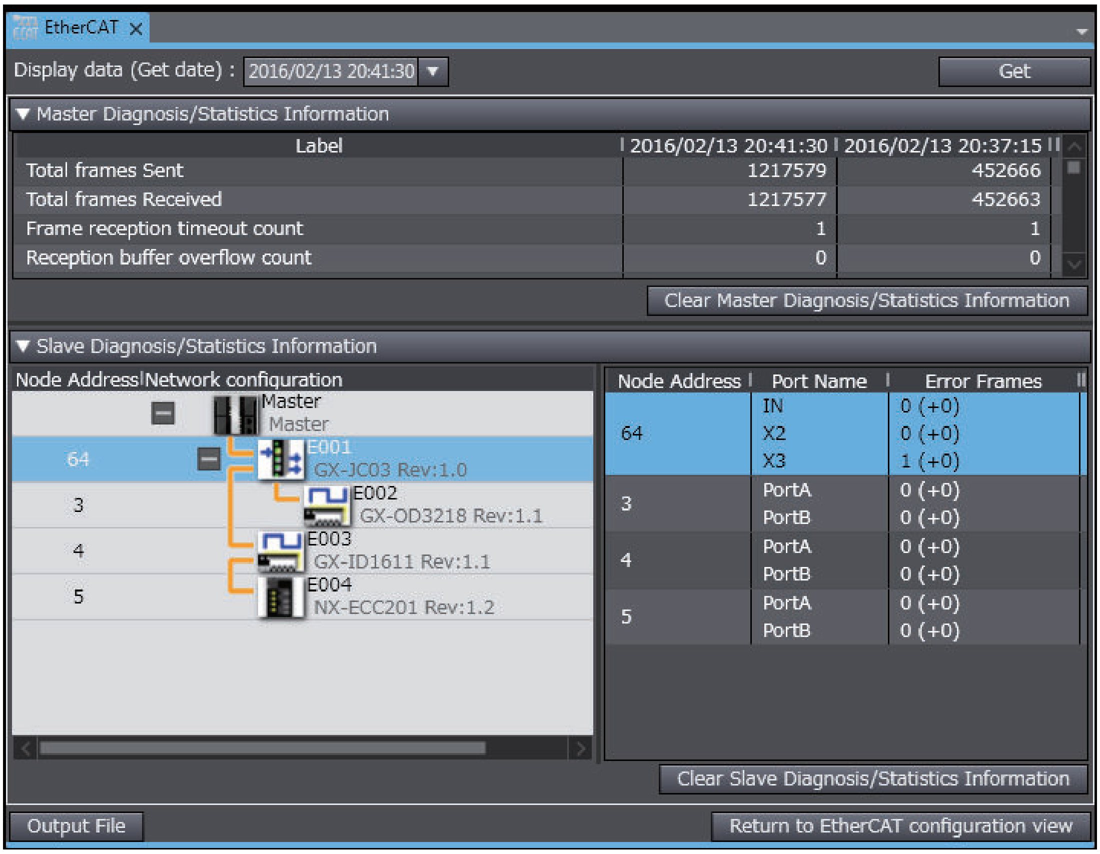
2 Double-click **EtherCAT** under **Configurations and Setup** on the Multiview Explorer. Or, right-click **EtherCAT** under **Configurations and Setup** and select **Edit**.



3 Right-click the EtherCAT master on the EtherCAT Tab Page and select **Display Diagnosis/ Statistics Information** from the menu.



The diagnostic and statistical information for the master and slaves are displayed.


EtherCAT
Get
▼ Master Diagnosis/Statistics Information
Clear Master Diagnosis/Statistics Information
▼ Slave Diagnosis/Statistics Information
Clear Slave Diagnosis/Statistics Information
Output File
Return to EtherCAT configuration view

Display data (Get date) : 2016/02/13 20:41:30

Label	2016/02/13 20:41:30	2016/02/13 20:37:15
Total frames Sent	1217579	452666
Total frames Received	1217577	452663
Frame reception timeout count	1	1
Reception buffer overflow count	0	0

Node Address	Network configuration
	Master Master
64	E001 GX-JC03 Rev:1.0
3	E002 GX-OD3218 Rev:1.1
4	E003 GX-ID1611 Rev:1.1
5	E004 NX-ECC201 Rev:1.2

Node Address	Port Name	Error Frames
64	IN	0 (+0)
	X2	0 (+0)
	X3	1 (+0)
3	PortA	0 (+0)
	PortB	0 (+0)
4	PortA	0 (+0)
	PortB	0 (+0)
5	PortA	0 (+0)
	PortB	0 (+0)

● Operation Button Functions

The following table describes the buttons to operate the diagnostic and statistical information.

Button	Function
Get	Acquires the diagnostic and statistical information from the EtherCAT master and EtherCAT slaves*1 and updates the display. You can retain a maximum of 100 diagnostic and statistical data that you acquired. When the **Display Diagnosis/Statistics Information** Tab Page is closed, the diagnostic and statistical information that you acquired are discarded.
Clear Master Diagnosis/ Statistics Information	Clears all values of the master diagnostic and statistical information to zeros.
Clear Slave Diagnosis/Statistics Information*2	Clears all values of the slave diagnostic and statistical information to zeros.
Output File	Outputs the diagnostic and statistical information to a CSV or a Viewer Format file*3 in a specified folder. The following data are output. • Master diagnostic and statistical information • Slave diagnostic and statistical information*4

*1. Only for the EtherCAT slaves that are registered in the network configuration information of the project.

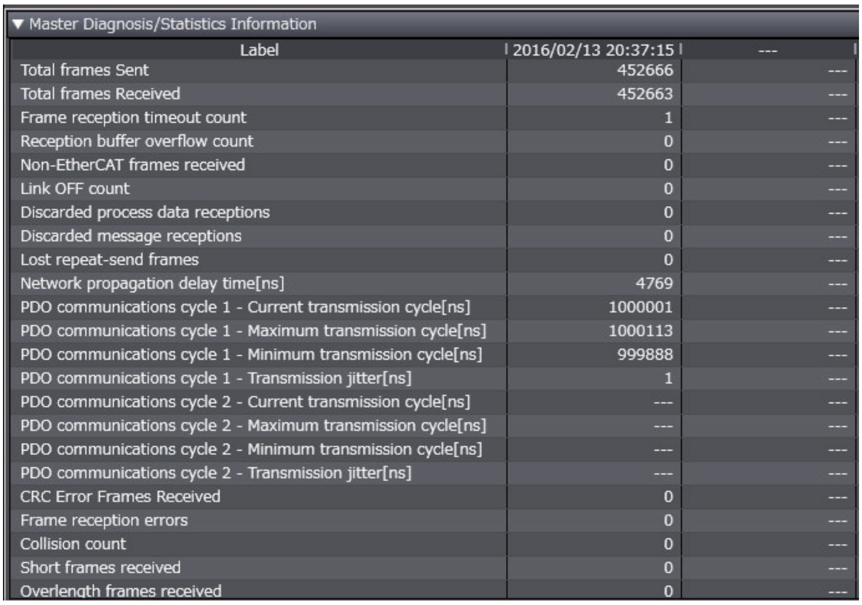
*2. This appears only when the unit version set in the Sysmac Studio project is 1.11 or later.

*3. The Viewer Format file is used to review the obtained diagnostic and statistical information in the EtherCAT Diagnosis/Statistics Information Viewer on the Sysmac Studio. Refer to *Diagnostic and Statistical Information* in the *Sysmac Studio Version 1 Operation Manual (Cat. No. W504)* for details on the EtherCAT Diagnosis/Statistics Information Viewer on the Sysmac Studio. Sysmac Studio version 1.29 or higher is required to use the EtherCAT Diagnosis/Statistics Information Viewer.

*4. For a CSV file, if the number of error frames exceeds 255, 255 is output.

● Master Diagnostic and Statistical Information

The following are the items displayed for the master diagnostic and statistical information.



▼ Master Diagnosis/Statistics Information

Label	2016/02/13 20:37:15	---
Total frames Sent	452666	---
Total frames Received	452663	---
Frame reception timeout count	1	---
Reception buffer overflow count	0	---
Non-EtherCAT frames received	0	---
Link OFF count	0	---
Discarded process data receptions	0	---
Discarded message receptions	0	---
Lost repeat-send frames	0	---
Network propagation delay time[ns]	4769	---
PDO communications cycle 1 - Current transmission cycle[ns]	1000001	---
PDO communications cycle 1 - Maximum transmission cycle[ns]	1000113	---
PDO communications cycle 1 - Minimum transmission cycle[ns]	999888	---
PDO communications cycle 1 - Transmission jitter[ns]	1	---
PDO communications cycle 2 - Current transmission cycle[ns]	---	---
PDO communications cycle 2 - Maximum transmission cycle[ns]	---	---
PDO communications cycle 2 - Minimum transmission cycle[ns]	---	---
PDO communications cycle 2 - Transmission jitter[ns]	---	---
CRC Error Frames Received	0	---
Frame reception errors	0	---
Collision count	0	---
Short frames received	0	---
Overlength frames received	0	---

Items that indicate the values may increase if the EtherCAT network is not operating normally.

Items that indicate the values may increase if the EtherCAT network is not operating normally.

The last two master diagnostic and statistical data acquired from the EtherCAT master are displayed. If the master diagnostic and statistical data is not acquired, "---" is displayed.

Precautions for Correct Use

- All values of the master diagnostic and statistical information are cleared to zeros when the power supply to the Controller is turned ON.
- The network propagation delay time is not cleared to zero when the master diagnostic and statistical information is cleared.
- Values of the network propagation delay time and transmission cycle are valid only if an EtherCAT slave that supports a distributed clock is assigned to the primary periodic task. If an EtherCAT slave that supports a distributed clock is not assigned to the primary periodic task, "---" is displayed.
- The items for PDO communications cycle 2 are valid only if an EtherCAT slave that supports a distributed clock is assigned to the priority-5 periodic task. If the CPU Unit does not have a priority-5 periodic task or if an EtherCAT slave that supports a distributed clock is not assigned to the priority-5 periodic task, "---" is displayed.

The values of the following items may increase if there is an EtherCAT network error.

The following table gives the description of each item, assumed cause of increase, and correction.

Item	Displayed information	Assumed cause of increase	Possible correction
Frame reception timeout count	The number of frame reception timeouts that occurred.	A cause to a *Process Data Reception Timeout* error occurred. The Frame reception timeout count will increase also when the count in the CRC Error Frames Received, Short frames received, or Overlength frames received increases.	Refer to *EtherCAT Network Diagnostic Procedure* on page 9-3.
Reception buffer overflow count	The number of frames discarded due to buffer overflows when receiving data.	There are too many frames on the EtherCAT network that are not EtherCAT frames.	See if there are computers or other non-EtherCAT devices connected to the EtherCAT network and remove them if there are any.
Non-EtherCAT frames received	The number of frames received other than EtherCAT frames.	There are frames on the EtherCAT network that are not EtherCAT frames.	See if there are computers or other non-EtherCAT devices connected to the EtherCAT network and remove them if there are any.
Link OFF count	The number of times link OFF was detected.	A cause to a *Link OFF Error* (84200000 hex) event occurred.	Refer to a *Link OFF Error* (84200000 hex) event in the *NJ/NX-series Troubleshooting Manual (Cat. No. W503)* and remove the cause of the error.
Discarded process data receptions	The number of process data packets discarded when receiving process data.	Operations for which multi-execution is prohibited in *9-3-2 Prohibition to Physically Disconnecting a Slave and Resetting an Error or Connecting a Slave at the Same Time* on page 9-25 were executed. The value sometimes increases when the network is started or communicating EtherCAT slaves are replaced.	Refer to *9-3-2 Prohibition to Physically Disconnecting a Slave and Resetting an Error or Connecting a Slave at the Same Time* on page 9-25.
Discarded message receptions	The number of messages discarded in mailbox reception.	A cause to an *EtherCAT Message Error* (842D0000 hex) or *Illegal Mailbox Received* (84350000 hex) event occurred.	Refer to an *EtherCAT Message Error* (842D0000 hex) or *Illegal Mailbox Received* (84350000 hex) event in the *NJ/NX-series Troubleshooting Manual (Cat. No. W503)* and remove the cause of the error.
CRC error frames received	The number of frames received that resulted in CRC errors.	The frames on the EtherCAT network are corrupted.	Refer to *EtherCAT Network Diagnostic Procedure* on page 9-3.
Frame reception errors	The number of frames resulting in reception errors from the Ethernet controller (EtherMAC).	The frames on the EtherCAT network are corrupted due to noise.	Implement noise countermeasures. Refer to *4-2-1 Installation Precautions* on page 4-12 for details.

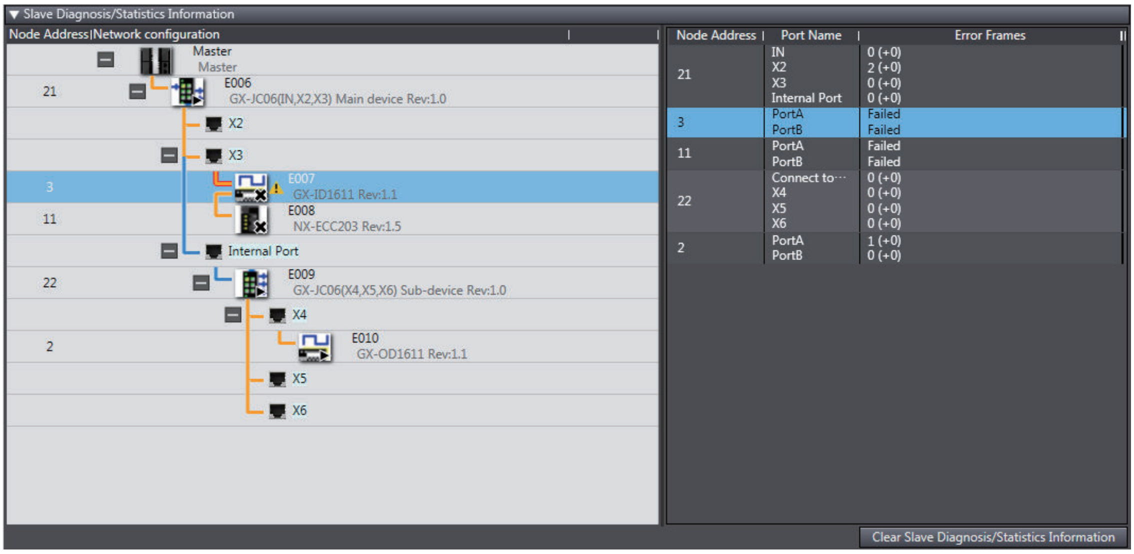
Item	Displayed information	Assumed cause of increase	Possible correction
Collision count	The total number of delay collisions on the line after start of transmissions.	A repeater hub is connected.	You cannot use repeater hubs with EtherCAT communications. See if there are repeater hubs connected to the EtherCAT network and remove them if there are any.
Short frames received	The number of frames received with less than 64 bytes.	The frames on the EtherCAT network are corrupted due to noise. Computers or other non-EtherCAT devices are connected.	Refer to *EtherCAT Network Diagnostic Procedure* on page 9-3.
Overlength frames received	The number of frames received with more than 1,522 bytes.	The frames on the EtherCAT network are corrupted due to noise. Computers or other non-EtherCAT devices are connected.	Refer to *EtherCAT Network Diagnostic Procedure* on page 9-3.

Precautions for Correct Use

Even if communications are normal, values other than 0 may occur for the frame reception timeout count and the number of CRC error frames received. Refer to *EtherCAT Network Diagnostic Procedure* on page 9-3 to determine if the EtherCAT network is operating normally.

● Slave Diagnostic and Statistical Information

The following are the items displayed for the slave diagnostic and statistical information.


▼ Slave Diagnosis/Statistics Information
Node Address|Network configuration
Master
Master
21
E006
GX-JC06(IN,X2,X3) Main device Rev:1.0
X2
X3
3
E007
GX-ID1611 Rev:1.1
11
E008
NX-ECC203 Rev:1.5
Internal Port
22
E009
GX-JC06(X4,X5,X6) Sub-device Rev:1.0
X4
2
E010
GX-OD1611 Rev:1.1
X5
X6
Clear Slave Diagnosis/Statistics Information

Node Address	Port Name	Error Frames
21	IN X2 X3 Internal Port	0 (+0) 2 (+0) 0 (+0) 0 (+0)
3	PortA PortB	Failed Failed
11	PortA PortB	Failed Failed
22	Connect to··· X4 X5 X6	0 (+0) 0 (+0) 0 (+0) 0 (+0)
2	PortA PortB	1 (+0) 0 (+0)

Item	Displayed information
Node Address|Network configuration	EtherCAT network configuration of the project
Node Address	Node addresses of the EtherCAT slaves
Port Name	Port names of the EtherCAT slave input or output ports Port names that are defined in the EtherCAT slave's ESI file • If no name is defined, Port A, Port B, Port C, or Port D is displayed. If there are two ports, an input and an output port, i.e., *Port A* and *Port B* is displayed. The input port which is the only input port is displayed first followed by the output ports (some EtherCAT slaves have more than one output port).

Item	Displayed information
Error Frames	Number of error frames received on the EtherCAT slave's input and output ports Range: 0 to 254 • *255 or more* is displayed if 255 or more error frames were received. • When the diagnostic and statistical information is acquired twice or more, the increase from the previous acquisition is displayed in parentheses on the right of the number of error frames. Example: When the number acquired at the previous acquisition was 6 and increased 4: 10 (+4) • *Failed* is displayed when the number of error frames cannot be acquired from EtherCAT slaves for reasons such as absence of the relevant EtherCAT slave. • You cannot acquire the slave diagnostic and statistical information when the diagnosis/statistics log of the CPU Unit is in progress. If you click the **Get** Button, *Auto logging* is displayed for the number of error frames. Note that if the result of the previous acquisition was *Failed*, *Failed* will be displayed again.

Precautions for Correct Use

When you display the slave diagnostic and statistical information, first open a Sysmac Studio project where the network configuration agrees with the actual network. If the network configuration in the project does not agree with the actual network, the slave diagnostic and statistical information is not displayed correctly. If a project where the network configuration agrees with the actual network is not available, create a new project, make the network configuration in the project agree with the actual network, and then display the slave diagnostic and statistical information. Refer to the *Sysmac Studio Version 1 Operation Manual (Cat. No. W504)* for information on building the network configuration.

Diagnosis/Statistics Log of CPU Unit

The periodic auto-acquisition of the diagnostic and statistical information is allowed for the following purposes during a test run or normal operation of equipment.

- Checking the EtherCAT network line quality for predictive monitoring and preventive maintenance
- Finding locations of errors when they occur

Version Information

- A CPU Unit with unit version 1.11 or later is required to use this function.

● Overview

This function acquires the diagnostic and statistical information that the EtherCAT master and slaves have. The information is acquired at the specified cycle and saved as a log file in an SD Memory Card that is mounted on the CPU Unit.

Use system-defined variables to set the execution command for this function and the cycle at which the diagnostic and statistical information is saved.

Precautions for Correct Use

- To use this function, you need an SD Memory Card. Refer to “Specifications of Supported SD Memory Cards, Folders, and Files” in the *NJ/NX-series CPU Unit Software User's Manual (Cat. No. W501)* for information such as the SD Memory Card types that the CPU Unit supports.
- When the diagnosis/statistics log is in progress, you cannot acquire or clear the slave diagnostic and statistical information from the diagnostic and statistical information display of the Sysmac Studio. Try it again after the diagnosis/statistics log is completed.
- When the diagnosis/statistics log is in progress, the response time in message communications with EtherCAT slaves may be extended.

● Diagnostic and Statistical Information to Acquire

The following table shows the information to acquire and describes the information.

Information to acquire	Description
Master diagnostic and statistical information	Same as the master diagnostic and statistical information displayed by the diagnostic and statistical information display of Sysmac Studio. Refer to *Master Diagnostic and Statistical Information* on page 9-13 for details.
Slave diagnostic and statistical information[*1]	Same as the number of error frames for the slave diagnostic and statistical information displayed by the diagnostic and statistical information display of Sysmac Studio. Refer to *Slave Diagnostic and Statistical Information* on page 9-15 for details.

[*1] Only for the EtherCAT slaves that are registered in the EtherCAT master network configuration information of the project.

Precautions for Correct Use

If this function is executed when the EtherCAT master network configuration information is in the following status, only the master diagnostic and statistical information is saved in a log file. The slave diagnostic and statistical information and system-defined variables are not saved.

- The network configuration information is cleared for the Clear All Memory operation.
- No EtherCAT slave is registered in the network configuration information.

● Setting Cycle to Save Diagnostic and Statistical Information in a Log File

You can specify the cycle to save the acquired diagnostic and statistical information in a log file within the range from 30 seconds to 30 minutes in units of seconds. You can also specify the cycle so that the diagnostic and statistical information is saved only once into a log file when the execution of this function is completed.

● Specifications of Log File

The acquired diagnostic and statistical information is saved as a CSV log file in the SD Memory Card. You can save a maximum of 1,000 diagnostic and statistical data in a log file.

The file name, file type and save location of the log file saved in the SD Memory Card are given below.

Item	Specifications
File name	ECAT_STATISTICS.csv
File format	CSV
Save location	Root folder

Additional Information

The following available space is required in an SD Memory Card to save the diagnostic and statistical information.
Size of a data = 273 bytes + 51 bytes × Number of EtherCAT slaves
If the number of EtherCAT slaves is 512 and you want to save 1,000 diagnostic and statistical data, an available space of approximately 26 MB is required.

The specification of data saved in a log file is given below.

- For the first row, data are output in the following order: the CPU Unit model, software version, package version, hardware version, and serial ID. For the following rows, the date and time of acquisition and the diagnostic and statistical data are output.
  Refer to *A-5 Items to Be Output to Log File of Diagnostic and Statistical Information, and Order of Outputs* on page A-33 for details on the items of diagnostic and statistical information to be output and the order of outputs.
- The number of error frames for the slave diagnostic and statistical information is left blank if the number of error frames cannot be acquired from the relevant EtherCAT slave.

● Related System-defined Variables

The following system-defined variables are used to control execution of the diagnosis/statistics log.
Refer to *7-1-2 System-defined Variables* on page 7-2 for details on system-defined variables.

Variable name	Meaning
_EC_StatisticsLogEnable	Diagnosis/Statistics Log Enable
_EC_StatisticsLogCycleSec	Diagnosis/Statistics Log Cycle
_EC_StatisticsLogBusy	Diagnosis/Statistics Log Busy
_EC_StatisticsLogErr	Diagnosis/Statistics Log Error

● Procedures

The procedure to use the diagnosis/statistics log is as follows.

1 Set a value for *_EC_StatisticsLogCycleSec* (Diagnosis/Statistics Log Cycle).
The cycle to save the diagnostic and statistical information in a log file is set.
Set the value in units of seconds. (Example: set 30 for 30 seconds, set 1,800 for 30 minutes).
If 0 is set, the diagnostic and statistical information is saved only once when execution of this function is completed.

2 Change *_EC_StatisticsLogEnable* (Diagnosis/Statistics Log Enable) from FALSE to TRUE.
The execution of the diagnosis/statistics log is started.

- *_EC_StatisticsLogErr* (Diagnosis/Statistics Log Error) changes to FALSE.
- *_EC_StatisticsLogBusy* (Diagnosis/Statistics Log Busy) changes to TRUE.

During execution of the diagnosis/statistics log, the diagnostic and statistical information is acquired and saved in a log file at the cycle specified in step 1.

If one of the following (a) to (f) is detected during the diagnosis/statistics log, the diagnosis/statistics log is automatically terminated.

a) The maximum number of diagnostic and statistical data are saved in a log file.
b) The SD Memory Card does not have sufficient available space.

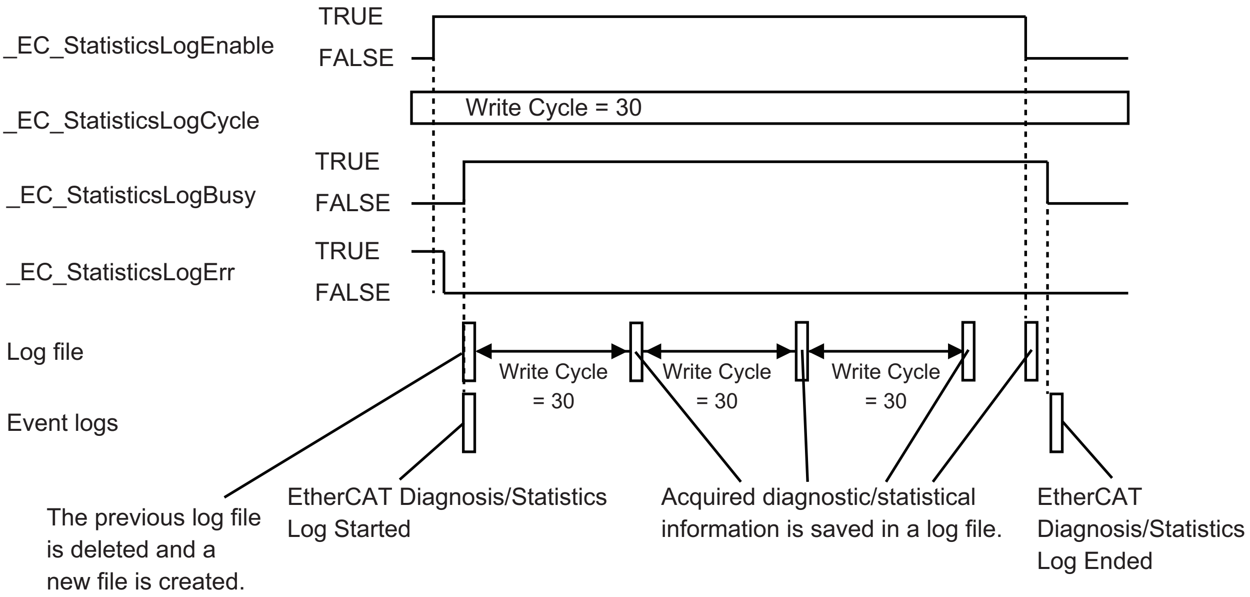
c) The SD Memory Card is write-protected.
d) There is no SD Memory Card.
e) A synchronization (or downloading) is executed on the Sysmac Studio.
f) The value set for *_EC_StatisticsLogCycleSec* is out of range.

- An EtherCAT Diagnosis/Statistics Log Started error is recorded in the event log.
- *_EC_StatisticsLogBusy* changes to FALSE.
- If (b), (c), (d) or (f) occurs, *_EC_StatisticsLogErr* changes to TRUE.
- An EtherCAT Diagnosis/Statistics Log Ended error is recorded in the event log.

3 Change *_EC_StatisticsLogEnable* (Diagnosis/Statistics Log Enable) from TRUE to FALSE.
Execution of the diagnosis/statistics log is terminated. The diagnostic and statistical information is saved in a log file when this variable changes to FALSE, regardless of the write cycle setting.

- *_EC_StatisticsLogBusy* changes to FALSE.
- An EtherCAT Diagnosis/Statistics Log Ended error is recorded in the event log.

The following figure shows the timing chart.


_EC_StatisticsLogEnable
TRUE
FALSE
_EC_StatisticsLogCycle
Write Cycle = 30
_EC_StatisticsLogBusy
TRUE
FALSE
_EC_StatisticsLogErr
TRUE
FALSE
Log file
Write Cycle = 30
Write Cycle = 30
Write Cycle = 30
Event logs
The previous log file is deleted and a new file is created.
EtherCAT Diagnosis/Statistics Log Started
Acquired diagnostic/statistical information is saved in a log file.
EtherCAT Diagnosis/Statistics Log Ended

Precautions for Correct Use

- To retain the log file saved in the SD Memory Card, rename the log file or save it in another memory before you execute the diagnosis/statistics log. Once the diagnosis/statistics log is started, the log file in the SD Memory Card is deleted and a new log file is created.
- During the diagnosis/statistics log, you cannot re-execute the diagnosis/statistics log. Check that *_EC_StatisticsLogBusy* is FALSE before you execute the diagnosis/statistics log.
- Even if you change the value of *_EC_StatisticsLogCycleSec* during the diagnosis/statistics log, the change is not reflected to the saving cycle of the diagnostic and statistical information. The changed saving cycle is used during the next execution of the diagnosis/statistics log. To use the changed saving cycle, you need to terminate the diagnosis/statistics log that is currently executed.
- Access *_EC_StatisticsLogErr* after *_EC_StatisticsLogBusy* changes from TRUE to FALSE. The value of *_EC_StatisticsLogErr* varies while *_EC_StatisticsLogBusy* is TRUE.

Additional Information

The following are the examples of methods to retain the log file saved in the SD Memory Card.

- Use the SD Memory Card instruction such as FileRename or FileCopy to change the file name.
- Use the FTP server function for the built-in EtherNet/IP to take the log file out.
- Remove the SD Memory Card and save the file in another medium.

9-2-5 Identifying an Error Slave and Cause of Error

Sysmac Studio provides the following functions to identify the slave in which an error occurred and the cause of the error.

- EtherCAT configuration view of the current error
- EtherCAT configuration information view of the event log

EtherCAT Configuration View of the Current Error

The following information is displayed on the EtherCAT configuration view in the slave diagnosis/statistics information display.

- Slave state
- Whether a slave with error exist or not
- Location where communications stopped

Version Information

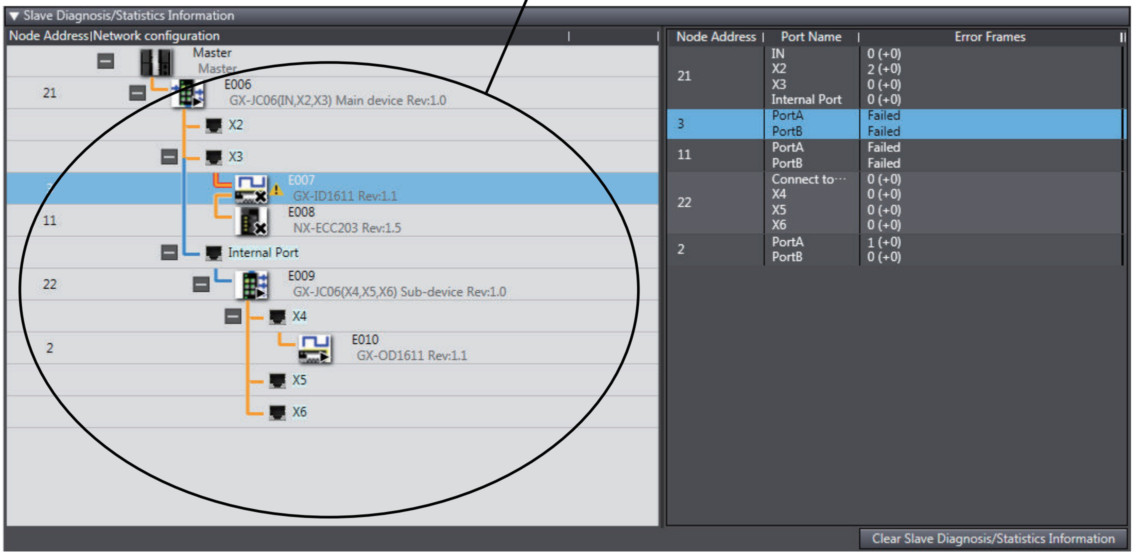
Sysmac Studio version 1.22 or higher is required to use the EtherCAT configuration view of the current error.

● Display Method

Right-click the master on the EtherCAT Tab Page and select **Display Diagnosis/Statistics Information** from the menu.

The network configuration information is displayed in topology diagram on the EtherCAT configuration view in the slave diagnosis/statistics information display.

EtherCAT configuration view



▼ Slave Diagnosis/Statistics Information
Node Address | Network configuration
Master
Master
21
E006
GX-JC06(IN,X2,X3) Main device Rev:1.0
X2
X3
3
E007
GX-ID1611 Rev:1.1
11
E008
NX-ECC203 Rev:1.5
Internal Port
22
E009
GX-JC06(X4,X5,X6) Sub-device Rev:1.0
X4
2
E010
GX-OD1611 Rev:1.1
X5
X6
Clear Slave Diagnosis/Statistics Information

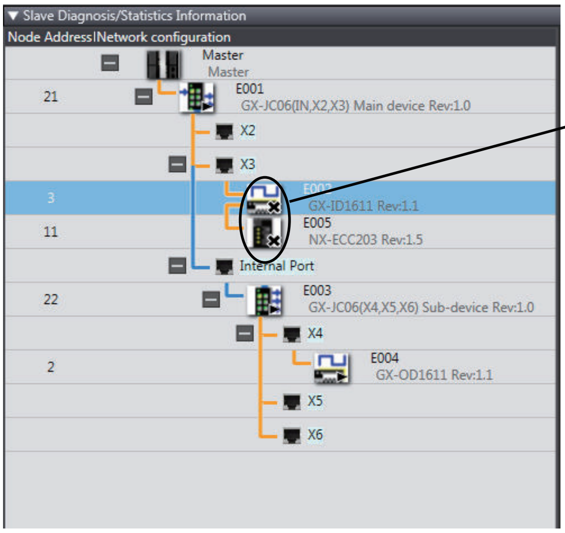
Node Address	Port Name	Error Frames
21	IN X2 X3 Internal Port	0 (+0) 2 (+0) 0 (+0) 0 (+0)
3	PortA PortB	Failed Failed
11	PortA PortB	Failed Failed
22	Connect to··· X4 X5 X6	0 (+0) 0 (+0) 0 (+0) 0 (+0)
2	PortA PortB	1 (+0) 0 (+0)

● Displayed Information

The following information is displayed on the EtherCAT configuration view in the slave diagnosis/statistics information display.

a. Slave state

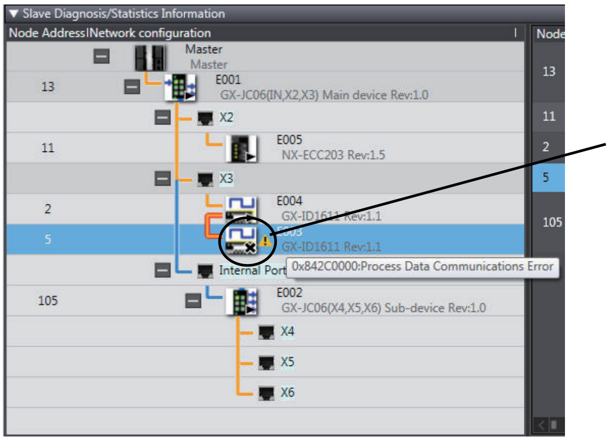
Slaves in Disabled, Disconnected and Not Matched states are shown by icons with marks. See below for an example of the pane showing slaves in Not Matched state.


Slave Diagnosis/Statistics Information
Node Address|Network configuration
Master
Master
21
E001
GX-JC06(IN,X2,X3) Main device Rev:1.0
X2
X3
3
E002
GX-ID1611 Rev:1.1
11
E005
NX-ECC203 Rev:1.5
Internal Port
22
E003
GX-JC06(X4,X5,X6) Sub-device Rev:1.0
X4
2
E004
GX-OD1611 Rev:1.1
X5
X6

Not Matched

b. Whether a slave with error exist or not

Slaves in which an error occurred are displayed with an error icon for emphasis.


Slave Diagnosis/Statistics Information
Node Address|Network configuration
Node
Master
Master
13
E001
GX-JC06(IN,X2,X3) Main device Rev:1.0
X2
11
E005
NX-ECC203 Rev:1.5
X3
2
E004
GX-ID1611 Rev:1.1
5
E003
GX-ID1611 Rev:1.1
Internal Port
105
E002
GX-JC06(X4,X5,X6) Sub-device Rev:1.0
X4
X5
X6
13
11
2
5
105

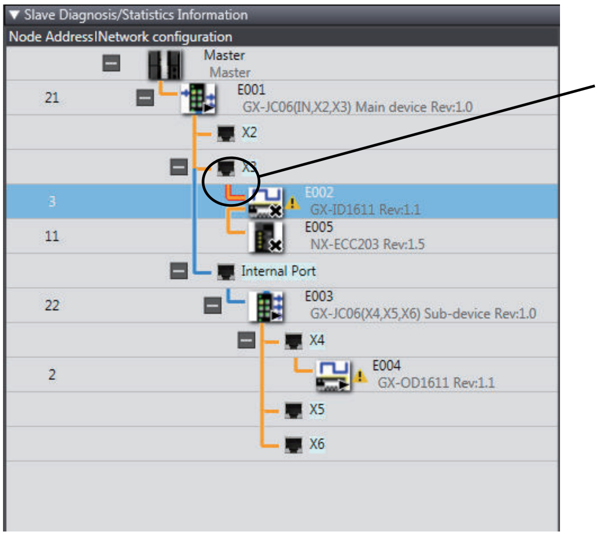
0x842C0000:Process Data Communications Error

Slave with an error

The event name of the current error on the relevant slave is displayed in the tooltip of the error icon.

c. Location where communications stopped

The connecting line between the ports at which communications are stopped are highlighted.


Slave Diagnosis/Statistics Information
Node Address|Network configuration
Master
Master
21
E001
GX-JC06(IN,X2,X3) Main device Rev:1.0
X2
X3
3
E002
GX-ID1611 Rev:1.1
11
E005
NX-ECC203 Rev:1.5
Internal Port
22
E003
GX-JC06(X4,X5,X6) Sub-device Rev:1.0
X4
2
E004
GX-OD1611 Rev:1.1
X5
X6

Location where communications stopped

EtherCAT Configuration Information View of the Event Log

The following information is displayed on the EtherCAT configuration information (Event log) view.

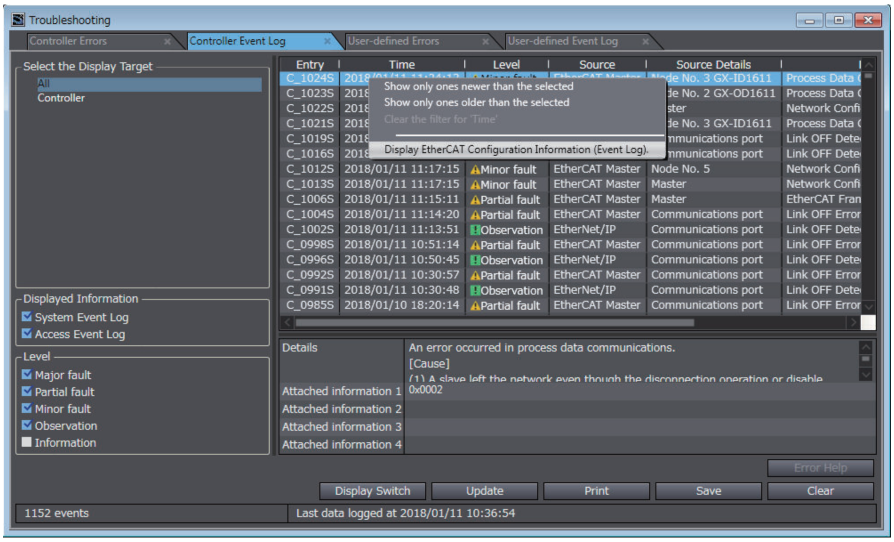
- Event in the event log whose event source is the EtherCAT Master Function Module

Version Information

Sysmac Studio version 1.22 or higher is required to use the EtherCAT configuration information view of the event log.

● Display Procedure

Right-click in the Controller Event Log Tab Page and select **Display EtherCAT Configuration Information (Event Log)** from the menu.



Troubleshooting
Controller Errors
Controller Event Log
User-defined Errors
User-defined Event Log
Show only ones newer than the selected
Show only ones older than the selected
Clear the filter for 'Time'
Display EtherCAT Configuration Information (Event Log).
Error Help
Display Switch
Update
Print
Save
Clear

Select the Display Target
All
Controller

Displayed Information
☑ System Event Log
☑ Access Event Log

Level
☑ Major fault
☑ Partial fault
☑ Minor fault
☑ Observation
□ Information

Entry	Time	Level	Source	Source Details	
C_1024S	2018/01/11 11:24:12	Minor fault	EtherCAT Master	Node No. 3 GX-ID1611	Process Data (
C_1023S	2018			de No. 2 GX-OD1611	Process Data (
C_1022S	2018			ster	Network Confi
C_1021S	2018			de No. 3 GX-ID1611	Process Data (
C_1019S	2018			mmunications port	Link OFF Dete
C_1016S	2018			mmunications port	Link OFF Dete
C_1012S	2018/01/11 11:17:15	⚠Minor fault	EtherCAT Master	Node No. 5	Network Confi
C_1013S	2018/01/11 11:17:15	⚠Minor fault	EtherCAT Master	Master	Network Confi
C_1006S	2018/01/11 11:15:11	⚠Partial fault	EtherCAT Master	Master	EtherCAT Fran
C_1004S	2018/01/11 11:14:20	⚠Partial fault	EtherCAT Master	Communications port	Link OFF Error
C_1002S	2018/01/11 11:13:51	!Observation	EtherNet/IP	Communications port	Link OFF Dete
C_0998S	2018/01/11 10:51:14	⚠Partial fault	EtherCAT Master	Communications port	Link OFF Error
C_0996S	2018/01/11 10:50:45	!Observation	EtherNet/IP	Communications port	Link OFF Dete
C_0992S	2018/01/11 10:30:57	⚠Partial fault	EtherCAT Master	Communications port	Link OFF Error
C_0991S	2018/01/11 10:30:48	!Observation	EtherNet/IP	Communications port	Link OFF Dete
C_0985S	2018/01/10 18:20:14	⚠Partial fault	EtherCAT Master	Communications port	Link OFF Error

Details	An error occurred in process data communications. [Cause] (1) A slave left the network even though the disconnection operation or disable
Attached information 1	0x0002
Attached information 2	
Attached information 3	
Attached information 4	

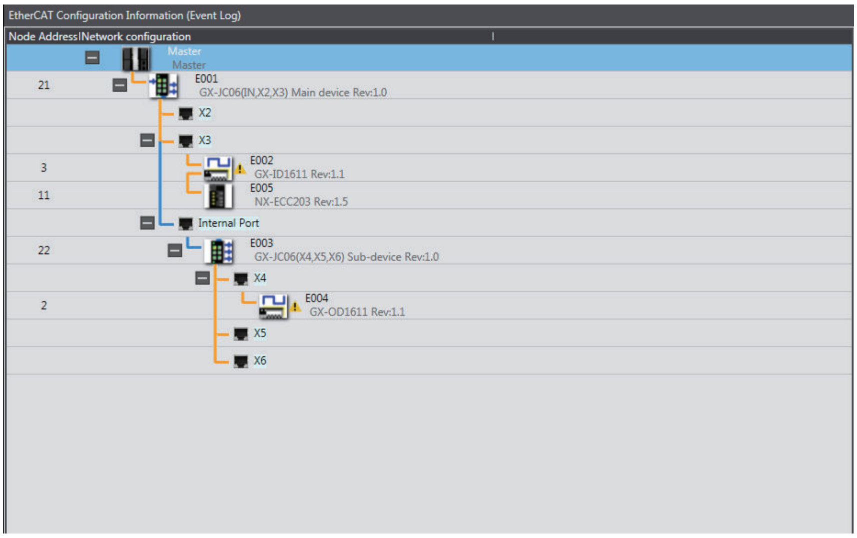
1152 events

Last data logged at 2018/01/11 10:36:54

Additional Information

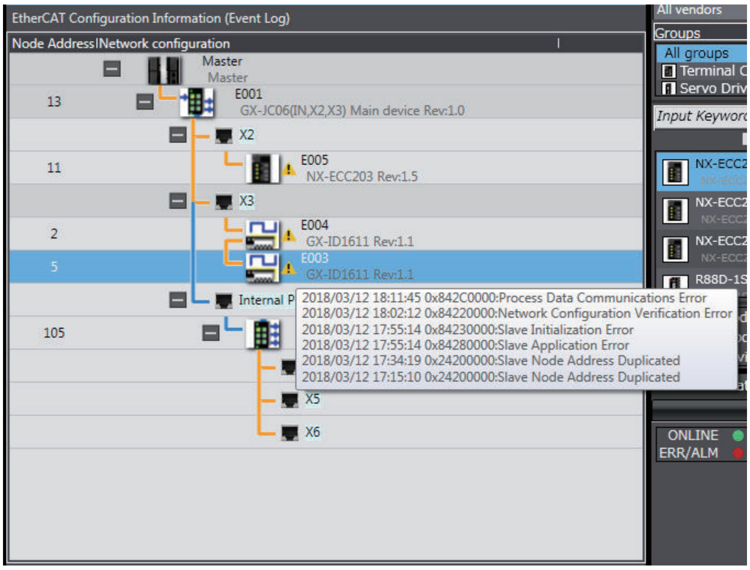
You can narrow down the range to display the event log using time information in which events occurred.
Right-click on the event log to select the condition to display from the menu.

The network configuration information is displayed on the EtherCAT configuration information (Event log) view in topology diagram.


EtherCAT Configuration Information (Event Log)
Node Address | Network configuration
Master
Master
21
E001
GX-JC06(IN,X2,X3) Main device Rev:1.0
X2
X3
3
E002
GX-ID1611 Rev:1.1
11
E005
NX-ECC203 Rev:1.5
Internal Port
22
E003
GX-JC06(X4,X5,X6) Sub-device Rev:1.0
X4
2
E004
GX-OD1611 Rev:1.1
X5
X6

● Display Information

Slaves are shown with error icons in the display when they have an error whose source is EtherCAT Master Function Module and whose level is higher than the minor fault level.



```
2018/03/12 18:11:45 0x842C0000:Process Data Communications Error
2018/03/12 18:02:12 0x84220000:Network Configuration Verification Error
2018/03/12 17:55:14 0x84230000:Slave Initialization Error
2018/03/12 17:55:14 0x84280000:Slave Application Error
2018/03/12 17:34:19 0x24200000:Slave Node Address Duplicated
2018/03/12 17:15:10 0x24200000:Slave Node Address Duplicated
```

9-3 Precautions When Connecting or Disconnecting Slaves during Communications

9-3-1 Procedure for Connecting and Disconnecting Slaves during Communications

Always use the following procedure to turn OFF the power supply to the slave or connect/disconnect cables during EtherCAT master communications.[*1]

[*1] This includes the Safe-Operational and Operational states.

Step 1. Use the Sysmac Studio or an instruction to send a command to disconnect the slave.
Step 2. Confirm that the slave was disconnected normally.
Step 3. Turn OFF the power supply to the slave or disconnect the cable.

If you turn OFF the power supply or disconnect the cable without performing steps 1 and 2, the slaves that are operating may be adversely affected.

9-3-2 Prohibition to Physically Disconnecting a Slave and Resetting an Error or Connecting a Slave at the Same Time

If you perform the following operation a or b at the same time as operation c, a *Slave Initialization Error* (84230000 hex) event will occur.

a. Turn OFF the power supply to the slave or disconnect the cable.
b. Turn ON the power supply to the slave or connect the cable.
c. Reset an error in the EtherCAT Master Function Module or connect the slave.[*1]

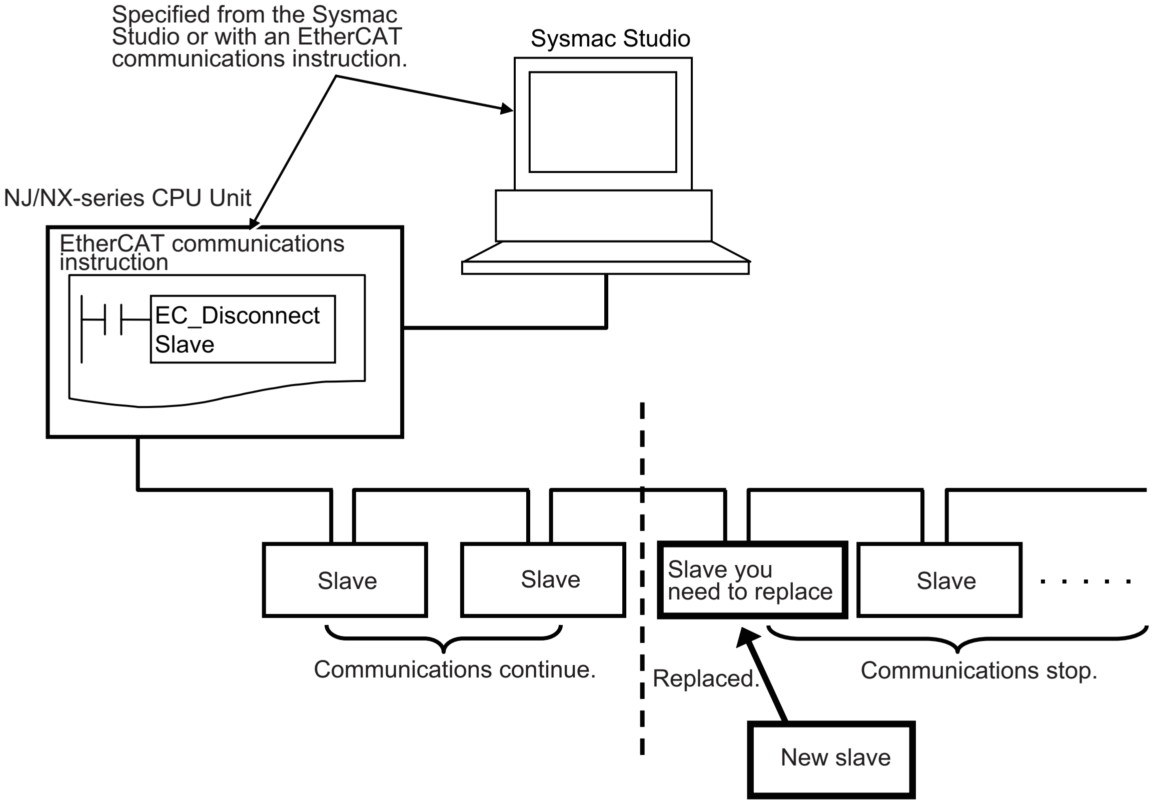
[*1] This can happen when the ResetECError (Reset EtherCAT Error) instruction or EC_ConnectSlave (Connect EtherCAT Slave) instruction is cyclically executed in the user program.

If you perform these operations at the same time, the EtherCAT master may access a slave with a different node address than the specified node address, or other unintended operations may occur. Therefore, never turn OFF the power supply to the slave or disconnect the cable at the same time as you reset an error or connect a slave.

9-4 Replacing Slaves during Communications

9-4-1 Introduction

You can temporarily stop and start communications with a specified slave and all subsequent slaves without stopping the entire communications system. This makes it possible to replace slaves during communications for maintenance, or for replacement when a slave malfunctions.


Specified from the Sysmac Studio or with an EtherCAT communications instruction.
Sysmac Studio
NJ/NX-series CPU Unit
EtherCAT communications instruction
EC_Disconnect Slave
Slave
Slave
Slave you need to replace
Slave
Communications continue.
Replaced.
Communications stop.
New slave

Precautions for Safe Use

Make sure that the communications distance, number of nodes connected, and method of connection for EtherCAT are within specifications.
Do not connect EtherCAT communications to EtherNet/IP, a standard in-house LAN, or other networks. An overload may cause the network to fail or malfunction.

Precautions for Correct Use

- An error occurs if you physically disconnect a slave from the network without executing the EC_DisconnectSlave (Disconnect EtherCAT Slave) instruction.
- An error may occur if the correct node address is not set for a replaced slave and a connection is made to a different port than the one that was used for the EC_DisconnectSlave (Disconnect EtherCAT Slave) instruction.

9-4-2 Slave Replacement Methods

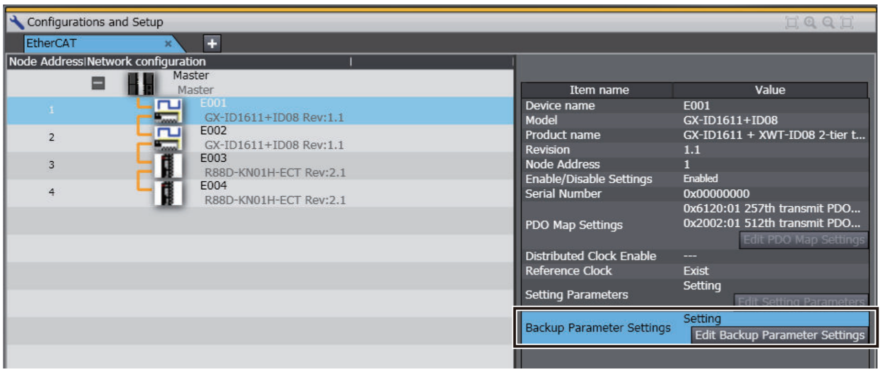
The slave to be replaced is first disconnected from the network and then reconnected after you replace it.

Use one of the following methods to disconnect and reconnect the slave.

- Method 1: Connecting and Reconnecting Specified Slaves from the Sysmac Studio
- Method 2: Executing the Disconnect EtherCAT Slave and Connect EtherCAT Slave Instructions. Execute the following instructions in the user program. You can use a pushbutton or an HMI to input the execution condition.

Function	Instruction	Description
Disconnect EtherCAT Slave	EC_DisconnectSlave	Temporarily disconnects a slave from the EtherCAT network for maintenance, such as replacement of the slave.
Connect EtherCAT Slave	EC_ConnectSlave	Reconnects a temporarily disconnected slave to the EtherCAT network after maintenance, such as replacement of the slave.

If the EtherCAT slave to replace has backup parameters, we recommend that you use the Sysmac Studio to replace the slave. (You can specify backup parameter settings in the EtherCAT slave from the Sysmac Studio.)



Configurations and Setup

EtherCAT

Node Address	Network configuration
	Master Master
1	E001 GX-ID1611+ID08 Rev:1.1
2	E002 GX-ID1611+ID08 Rev:1.1
3	E003 R88D-KN01H-ECT Rev:2.1
4	E004 R88D-KN01H-ECT Rev:2.1

Item name	Value
Device name	E001
Model	GX-ID1611+ID08
Product name	GX-ID1611 + XWT-ID08 2-tier t...
Revision	1.1
Node Address	1
Enable/Disable Settings	Enabled
Serial Number	0x00000000
PDO Map Settings	0x6120:01 257th transmit PDO... 0x2002:01 512th transmit PDO... Edit PDO Map Settings
Distributed Clock Enable	---
Reference Clock	Exist
Setting Parameters	Setting Edit Setting Parameters
Backup Parameter Settings	Setting Edit Backup Parameter Settings

Precautions for Correct Use

For the slaves in the ring topology, you cannot perform the disconnect operation from the Sysmac Studio and execute the Disconnect EtherCAT Slave instruction. If you perform the disconnect operation from the Sysmac Studio or execute the Disconnect EtherCAT Slave instruction to the slaves in the ring topology, the slaves end in errors.

If you replace the salve in the ring topology, you need to disconnect the originating slave of the ring and then disconnect the entire ring topology. When the originating slave of the ring is disconnected, the EtherCAT master stops process data communications with slaves in the ring topology and slaves that are connected to the drop line from the ring.

Additional Information

For slaves that are disabled with the EC_ChangeEnableSetting (Enable/Disable EtherCAT Slave) instruction or EtherCAT Slave Enable/Disable Settings, you cannot disconnect and reconnect them from the Sysmac Studio. If you execute the Disconnect EtherCAT Slave and Connect EtherCAT Slave instructions, they end in errors.

9-4-3 Backing Up Settings

Before you replace an EtherCAT slave that is currently performing communications, the settings of the EtherCAT slave to replace are backed up.

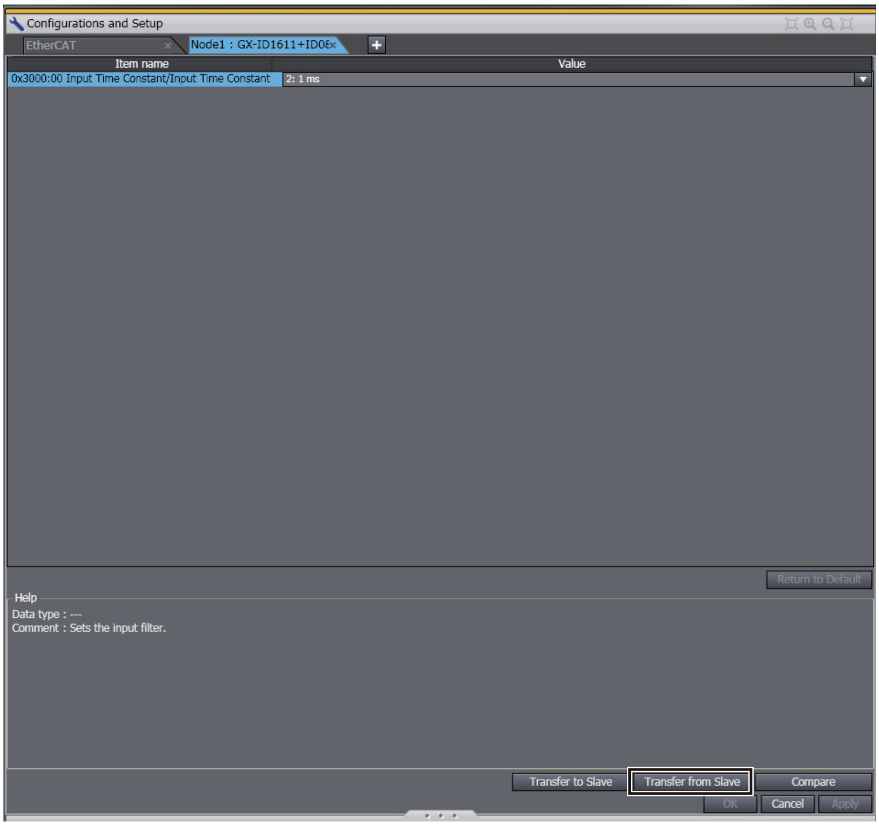
The settings of an EtherCAT slave are divided according to the storage locations and setting methods into initial parameters and backup parameters, as shown in the following table.

Slave settings	Storage location	Slave setting method
Initial parameters	EtherCAT master	These parameters are set automatically from the EtherCAT master when EtherCAT communications start or when a slave is connected.
Backup parameters	EtherCAT slaves	You set these parameters by transferring them to the slaves from the Backup Parameter Settings Tab Page of the Sysmac Studio. The data can also be transferred from the EtherCAT Drive Tab Page.

To replace an EtherCAT slave, you first back up the backup parameters that are stored in the EtherCAT slave.

1 Click the **Edit Backup Parameter Settings** Button in the EtherCAT Slave Setting Tab Page.
The Backup Parameter Settings Tab Page is displayed.

2 Click the **Transfer from Slave** Button.
All of the backup parameters that are stored in the EtherCAT slave and displayed in the list are transferred from the EtherCAT slave.



Item name	Value
0x3000:00 Input Time Constant/Input Time Constant	2: 1 ms

Data type : ---
Comment : Sets the input filter.

3 Click the **Compare** Button.

Check to be sure that you have correctly obtained the backup parameters in step 2.

Additional Information

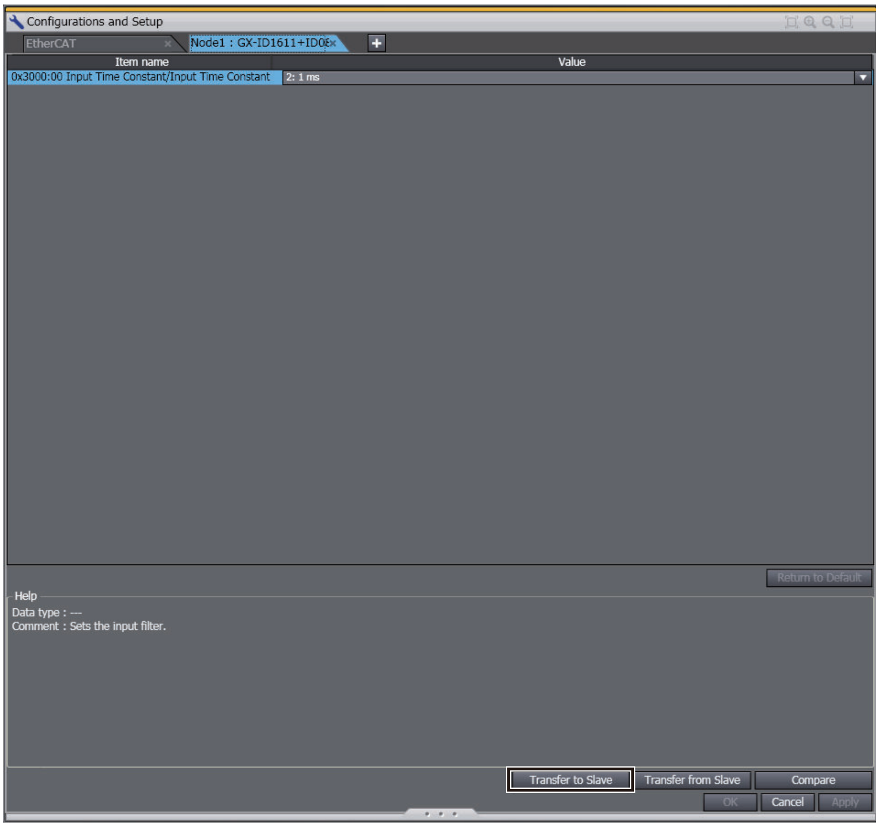
- You can back up the EtherCAT slave settings for any EtherCAT slave that is connected to the network (i.e., whenever the _EC_EntrySlavTbl[] (Network Connected Slave Table) system-defined variable is TRUE) either before or after the disconnection command is sent to the EtherCAT slave.
- You can also set the EtherCAT drive slaves from the EtherCAT Drive Tab Page. Back up the settings information from the EtherCAT Drive Tab Page. Refer to the *Sysmac Studio Version 1 Operation Manual (Cat. No. W504)* for the applicable EtherCAT drive slaves.
- You do not need to back up the settings to replace an EtherCAT slave that does not have backup parameters.

9-4-4 Restoring Settings

After you replace the EtherCAT slave, you restore the settings that you backed up before you replaced the slave. The backup parameters that were backed up are restored to the EtherCAT slave.

1. Click the **Edit Backup Parameter Settings** Button in the EtherCAT Slave Setting Tab Page.
   The Backup Parameter Settings Tab Page is displayed.

2. Click the **Transfer to Slave** Button.
   All of the backup parameters that are stored in the EtherCAT slave and displayed in the list are downloaded to the EtherCAT slave.



Item name	Value
0x3000:00 Input Time Constant/Input Time Constant	2: 1 ms

Help
Data type : ---
Comment : Sets the input filter.

3. Click the **Compare** Button.
   Check to be sure that you have correctly transferred the backup parameters in step 2.

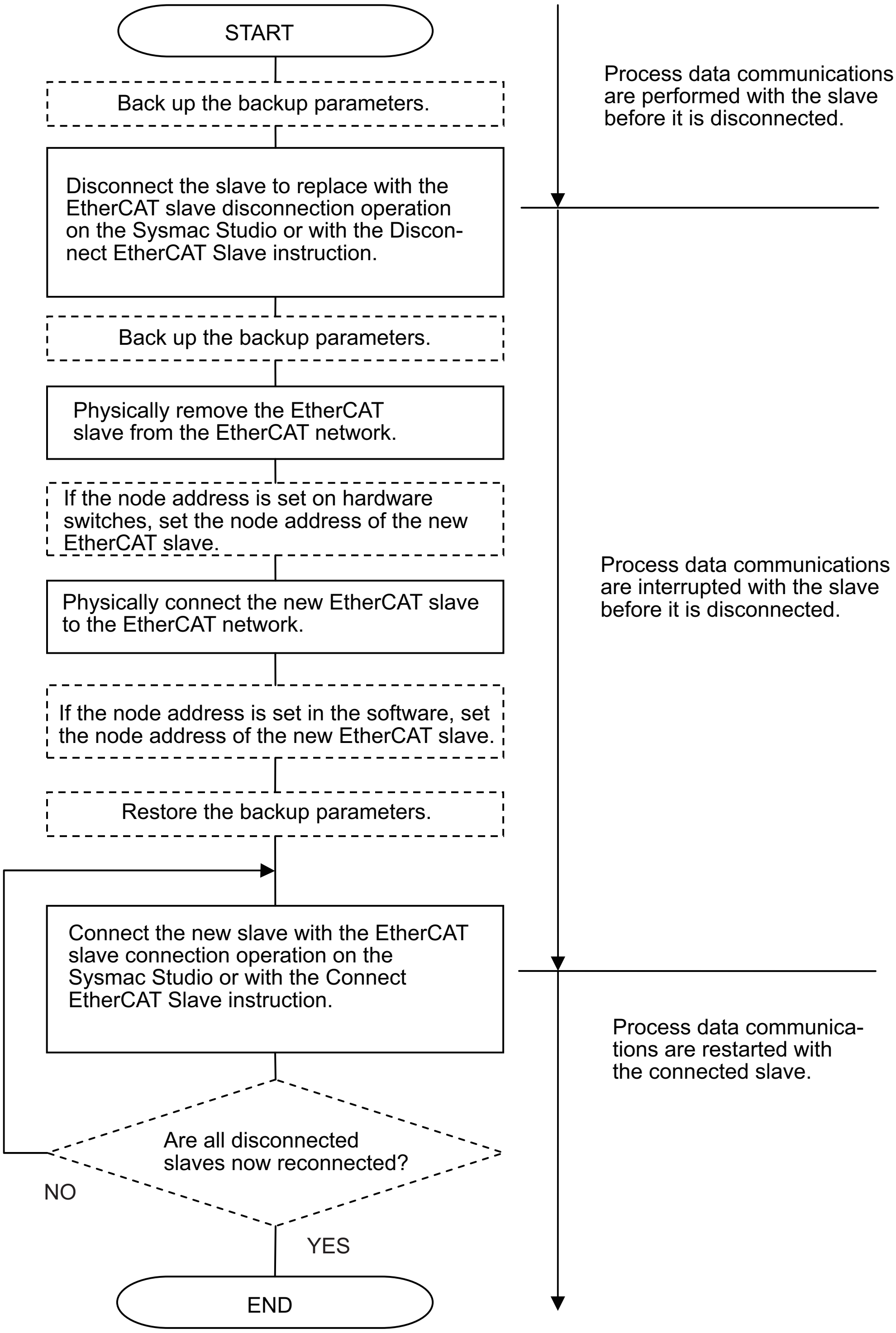
Precautions for Correct Use

Restore the settings to the EtherCAT slave before you connect the EtherCAT slave to restart process data communications. Restore the EtherCAT slave settings while the slave is connected to the network (i.e., whenever the _EC_EntrySlavTbl[] (Network Connected Slave Table) system-defined variable is TRUE). You must set the node address of the EtherCAT slave before it can participate in the network.

Additional Information

- You can set the EtherCAT drive slaves from the EtherCAT Drive Tab Page. Restore the settings information from the EtherCAT Drive Tab Page. Refer to the *Sysmac Studio Version 1 Operation Manual (Cat. No. W504)* for the applicable EtherCAT drive slaves.
- You do not need to restore the settings to replace an EtherCAT slave that does not have backup parameters.

9-4-5 Replacement Procedure


START
Back up the backup parameters.
Disconnect the slave to replace with the EtherCAT slave disconnection operation on the Sysmac Studio or with the Disconnect EtherCAT Slave instruction.
Back up the backup parameters.
Physically remove the EtherCAT slave from the EtherCAT network.
If the node address is set on hardware switches, set the node address of the new EtherCAT slave.
Physically connect the new EtherCAT slave to the EtherCAT network.
If the node address is set in the software, set the node address of the new EtherCAT slave.
Restore the backup parameters.
Connect the new slave with the EtherCAT slave connection operation on the Sysmac Studio or with the Connect EtherCAT Slave instruction.
Are all disconnected slaves now reconnected?
NO
YES
END
Process data communications are performed with the slave before it is disconnected.
Process data communications are interrupted with the slave before it is disconnected.
Process data communications are restarted with the connected slave.

Note Steps shown in dotted boxes depend on the system.

Precautions for Safe Use

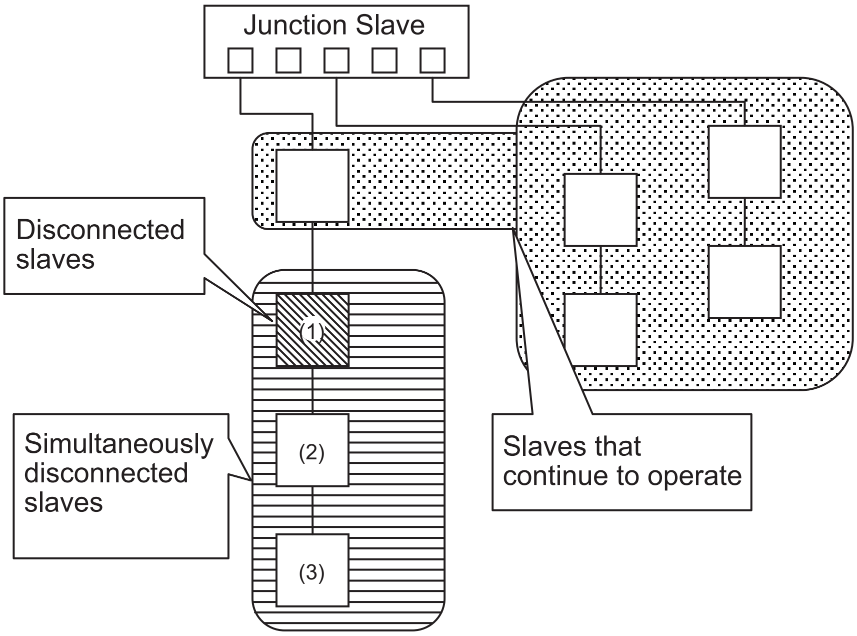
- When an EtherCAT slave is disconnected from the Sysmac Studio or with an instruction, communications will stop and control of the outputs will be lost not only for the disconnected slave, but for all slaves connected after it.
  Always confirm system safety before you disconnect an EtherCAT slave.
- If noise occurs or an EtherCAT slave is disconnected from the network, any current communications frames may be lost. If frames are lost, slave I/O data is not communicated, and unintended operation may occur. The slave outputs behave according to the slave specifications. For details, refer to relevant manuals for each slave. If a noise countermeasure or slave replacement is required, perform the following processing.
  - Program the *_EC_InDataInvalid* (Input Data Invalid) system-defined variable as an interlock condition in the user program. Refer to *6-1-3 Checking Validity of Process Data* on page 6-6.
  - Set the **PDO communications timeout detection count** setting in the EtherCAT master to at least 2. Refer to *5-4 EtherCAT Master Parameter Settings* on page 5-14 for the setting procedure.

Precautions for Correct Use

If you replace a slave in a ring topology, when the wiring is incorrect and input ports or output ports are connected each other, an EtherCAT Frame Not Received will occur and message communications and process data communications may stop. Therefore, make sure that the wiring of communications cables for the slave that you replace is correct.

Disconnecting Slaves

If a slave is disconnected, slaves connected after the designated slave (on the output side) in a daisy chain are disconnected at the same time. Slaves connected before the disconnected slave (on the input side) and slaves connected beyond Junction Slaves continue to operate.
(If (1) is disconnected in the following figure, (2) and (3) are also disconnected.)


Junction Slave
Disconnected slaves
(1)
Simultaneously disconnected slaves
(2)
(3)
Slaves that continue to operate

When the connected slave is disconnected with an instruction or from the Sysmac Studio, the status of the system-defined variables for the slaves are listed in the following table.

System-defined variable	Description	Value
_EC_RegSlavTbl[] (Registered Slave Table)	This variable shows the slaves that are registered in the network configuration information.	Remains TRUE.
_EC_EntrySlavTbl[] (Network Connected Slave Table)	This variable shows the slaves that are registered in the network configuration information and connected to the network.	Remains TRUE.[*1]
_EC_MBXSlavTbl[] (Message Communications Enabled Slave Table)	This variable shows the slaves that are registered in the network configuration information and able to perform message communications.	Changes to FALSE.[*2]
_EC_PDSlavTbl[] (Process Data Communicating Slave Table)	This variable shows the slaves that are registered in the network configuration information and performing process data communications.	Changes to FALSE.
_EC_DisconnSlavTbl[] (Disconnected Slave Table)	This variable shows the slaves that are registered in the network configuration information and disconnected from the network.	Changes to TRUE.

*1. After the slave is disconnected, it changes to FALSE when the slave is removed from the actual network configuration.

*2. For project unit version 1.40 or later, it changes to FALSE when the slave is removed from the actual network configuration.

Reconnecting Slaves

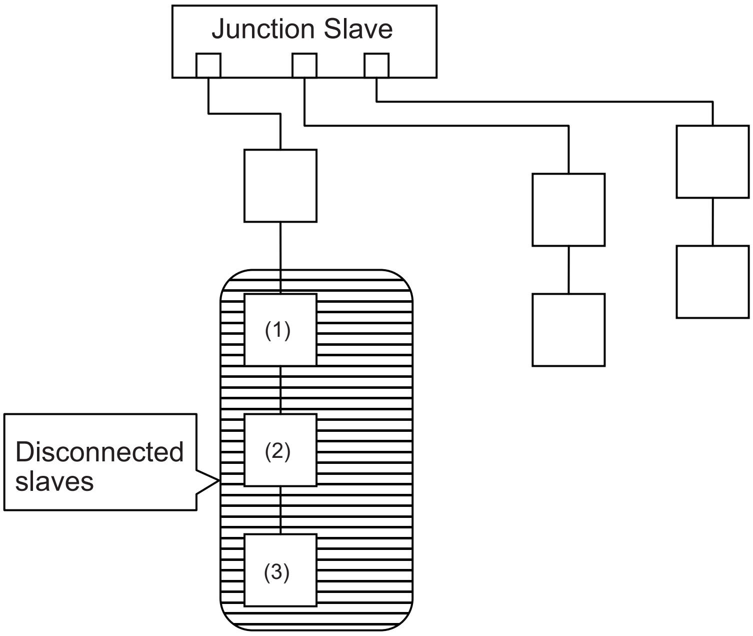
When reconnecting disconnected slaves after replacement or inspection, make sure that the following conditions are met and then reconnect them. If the following conditions are met, *_EC_EntrySlavTbl* (Network Connected Slave Table) system-defined variable will be TRUE.

- Make sure that the slave's node address is set correctly.
- Make sure that there are no errors in the order that the slaves are connected.

Additional Information

If slaves are reconnected without meeting above conditions, process data communications with the slaves will not start again, and a *Network Configuration Verification Error* (84220000 hex) event will occur. However, for project unit version 1.40 or later, a *Network Configuration Verification Error* (84220000 hex) event will not occur.

There are no restrictions on the order when reconnecting slaves that are in a daisy chain. For example, even if slaves (1) to (3) in a configuration such as the one shown in the following figure are disconnected, there are no restriction on the reconnection order of slaves (1) to (3). (For example, the slaves can be reconnected in the order (3), (2), (1) or (1), (3), (2).)



When the disconnected slave is reconnected with an instruction or from the Sysmac Studio, the status of the system-defined variables for the slaves are listed in the following table.

System-defined variable	Description	Value
_EC_RegSlavTbl[] (Registered Slave Table)	This variable shows the slaves that are registered in the network configuration information.	Remains TRUE.
_EC_EntrySlavTbl[] (Network Connected Slave Table)	This variable shows the slaves that are registered in the network configuration information and connected to the network.	*1

System-defined variable	Description	Value
_EC_MBXSlavTbl[] (Message Communications Enabled Slave Table)	This variable shows the slaves that are registered in the network configuration information and able to perform message communications.	*1
_EC_PDSlavTbl[] (Process Data Communicating Slave Table)	This variable shows the slaves that are registered in the network configuration information and performing process data communications.	Changes to TRUE.
_EC_DisconnSlavTbl[] (Disconnected Slave Table)	This variable shows the slaves that are registered in the network configuration information and disconnected from the network.	Changes to FALSE.

*1. The value changes to TRUE when the slave is connected to the actual network configuration. It remains TRUE even if a salve is reconnected after the change.

Additional Information

For project unit version earlier than 1.40, synchronization correction processing is performed to reconnect synced slaves. Therefore, several seconds may be required per slave until reconnection is completed.

A

Appendices

The appendices describe the relation of EtherCAT communications to overall CPU Unit status, packet monitoring functions, and multi-vendor application.

A-1 EtherCAT Status in Relation to CPU Unit Status

The status of EtherCAT master memory, the ability to download master and slave settings, and the slave status are given below for different CPU Unit operating modes, Controller errors, and other status.

A-1-1 When the Power Supply Is Turned ON

Memory related to EtherCAT master				Downloading master and slaves settings	Slaves	
Device variables	Category _EC system-defined variables	Network configuration information	Diagnostic and statistical information/ Protocol monitor		Slave outputs	Slave communications status
Devices variables that are not retained: Initial values Axes variables: Initial values	Initial values	---	Cleared to all zeros.	---	The status of slave outputs before the start of EtherCAT communications depend on the slave. Slaves output values of device variables after EtherCAT communications start.	Enters the Operational state.

A-1-2 CPU Unit Operating Modes

CPU Unit Status	Memory related to EtherCAT master					Downloading master and slaves settings	Slaves	
	Device variables	Category _EC system-defined variables	Network configuration information	Diagnostic and statistical information/ Protocol monitor	Current errors		Slave outputs	Slave communications status
PROGRAM mode	Shows the I/O data exchanged on the EtherCAT communications.	Continually shows EtherCAT communications status.*2	Does not change.*2	Continually shows EtherCAT communications status.*2	Retained.*2	• Master settings: OK • Slave settings: OK*3	Continually shows the output data sent from the EtherCAT master.*2	Does not change.*2
Switching between PROGRAM and RUN modes and *Device Output Hold Configuration1 disabled**	Devices variables that are not retained: Initial values Axis variables: Retained					---		
Switching between PROGRAM and RUN modes and *Device Output Hold Configuration1 enabled**	Device variables that are not retained: Retained Axis variables: Retained							
RUN mode	Shows the I/O data exchanged on the EtherCAT communications.					• Master settings: Not possible. • Slave settings: OK*3		

*1. Refer to the *NJ/NX-series CPU Unit Software User's Manual (Cat. No. W501)* for the Device Output Hold Configuration.

*2. It does not depend on the operating mode of the CPU Unit.

*3. However, the setting may not be possible depending on the status of the slave.

A-1-3 Controller Errors Other Than Errors in the EtherCAT Master

CPU Unit Status	Memory related to EtherCAT master					Downloading master and slaves settings	Slaves	
	Device variables	Category _EC system-defined variables	Network configuration information	Diagnostic and statistical information/ Protocol monitor	Current errors		Slave outputs	Slave communications status
A major fault level Controller error occurs.	Devices variables that are not retained: Initial values Axis variables: Retained	Continually shows EtherCAT communications status.*1	Does not change.*1	Continually shows EtherCAT communications status.*1	Retained.*1	---	In Operational state, the values from before operation stopped are output. When the slaves have entered Safe-Operational state, the outputs depend on the slave settings. Inputs are enabled.	EtherCAT communications stop. Enters the Safe-Operational state.
A partial fault level Controller error occurs.	Shows the I/O data exchanged on the EtherCAT communications.						Continually shows the output data sent from the EtherCAT master.*1	Does not change.*1
A minor fault level Controller error occurs.								

*1. It does not depend on the operating mode of the CPU Unit.

Refer to *9-1 Overview of Troubleshooting* on page 9-2 if a Controller error occurs in the EtherCAT master.

A-1-4 Others

CPU Unit Status	Memory related to EtherCAT master					Downloading master and slaves settings	Slaves	
	Device variables	Category _EC system-defined variables	Network configuration information	Diagnostic and statistical information/ Protocol monitor	Current errors		Slave outputs	Slave communications status
Memory all clear	Cleared to all zeros.	Entirely initialized (0).	Deleted.	Cleared to all zeros.	Cleared to all zeros.	---	Depends on the slave settings.	Enters the Initialized state.
Downloading data and *Device Output Hold Configuration1 disabled**	Devices variables that are not retained: Initial values Axis variables: Retained	Continually shows EtherCAT communications status.*2	Does not change.*2	Cleared to all zeros.	Cleared to all zeros.*3			• Enters the Initialized state when parameters are transferred to the master. • Enters the Pre-Operational state when backup parameters are downloaded to slaves.
Downloading data and *Device Output Hold Configuration1 enabled**	Device variables that are not retained: Retained Axis variables: Retained			Continually shows EtherCAT communications status.*2	Retained.*2		Continually shows the output data sent from the EtherCAT master.*2	Does not change.*2

*1. Refer to the *NJ/NX-series CPU Unit Software User's Manual (Cat. No. W501)* for the Device Output Hold Configuration.

*2. It does not depend on the operating mode of the CPU Unit.

*3. However, after downloading data, the EtherCAT master restarts, and records any event as a current error if the cause of the error has not been removed.

A-1-5 When the Power Supply Is Turned OFF

Memory related to EtherCAT master					Downloading master and slaves settings	Slaves	
Device variables	Category _EC system-defined variables	Network configuration information	Diagnostic and statistical information/Protocol monitor	Current errors		Slave outputs	Slave communications status
Does not change.	Does not change.	Does not change.	Updating the protocol monitor stops.	Cleared to all zeros.	---	Depends on the slave settings.	Depends on the behavior of slaves when EtherCAT communications stop.

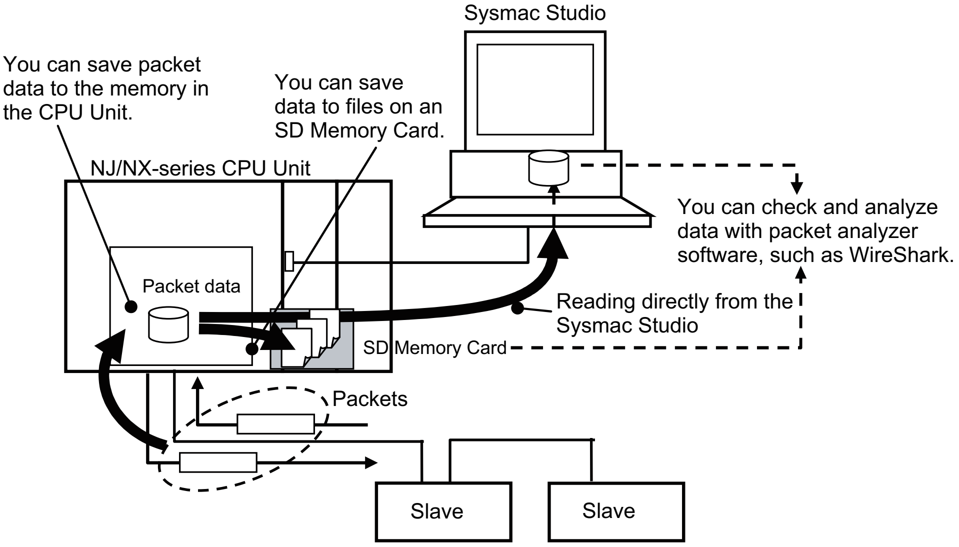
A-2 Monitoring Packets

Packet monitoring stores a certain number of the most recent packets sent and received by the EtherCAT master along with time information.
You can use an EtherCAT communications instruction or the Sysmac Studio to save the captured packet data in packet data files in the system in the CPU Unit.

You can use any of the following methods to obtain the packet data saved in the system memory in the CPU Unit.

- Reading directly from the Sysmac Studio
- Saving to an SD Memory Card inserted in the CPU Unit

You can view the captured packet data with packet analyzer software, such as WireShark. You can also use the data for analysis applications, such as error analysis and data mining.


Sysmac Studio
You can save packet data to the memory in the CPU Unit.
You can save data to files on an SD Memory Card.
NJ/NX-series CPU Unit
Packet data
You can check and analyze data with packet analyzer software, such as WireShark.
Reading directly from the Sysmac Studio
SD Memory Card
Packets
Slave
Slave

Version Information

For the NJ301-□□□□ CPU Units, the CPU Unit with unit version 1.10 or later and Sysmac Studio version 1.12 or higher are required to use the packet monitoring. However, for project unit version 1.40 or later, the packet monitoring cannot be used regardless of the models of the CPU Units.
If the packet monitoring cannot be used, the *_EC_PktMonStop* system-defined variable, which shows the operating status of packet monitoring, will always be TRUE.
In addition, if you execute any EtherCAT instructions for packet monitoring (EC_StartMon, EC_StopMon, EC_SaveMon, or EC_CopyMon) in the user program, an error that shows packet monitoring cannot be used occurs.

Starting and Stopping Packet Monitor

You can start and stop packet monitoring either with instructions in the user program or with operations on the Sysmac Studio.

● Using Instructions in the User Program

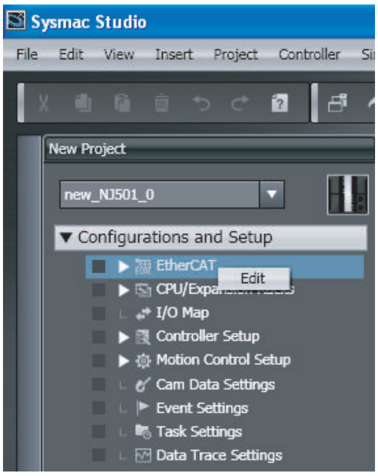
- EC_StartMon (Start EtherCAT Packet Monitor) instruction: Starts the execution of packet monitoring and continues to update a fixed number of packets.

- EC_StopMon (Stop EtherCAT Packet Monitor) instruction: Stops the execution of packet monitoring.

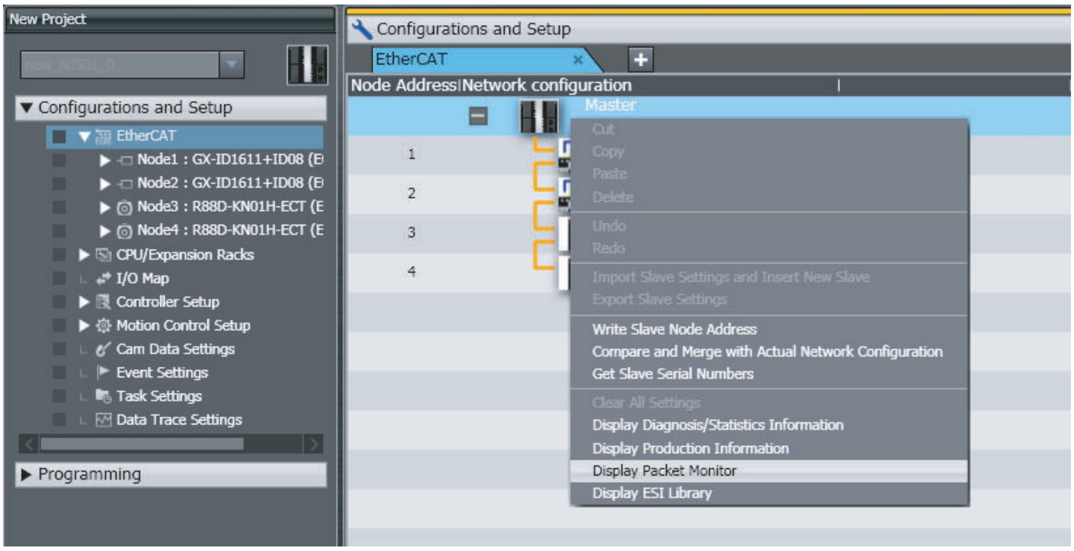
● Operation from the Sysmac Studio

1 Start the Sysmac Studio and go online with the Controller.

2 Double-click **EtherCAT** under **Configurations and Setup** on the Multiview Explorer. Or, right-click **EtherCAT** under **Configurations and Setup** and select **Edit**.

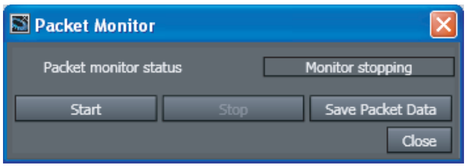


3 Right-click the master on the EtherCAT Tab Page and select **Display Packet Monitor** from the menu.

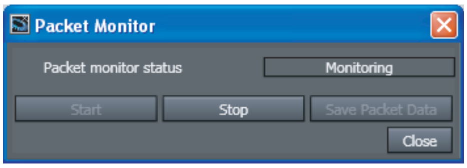


The **Packet Monitor** Dialog Box is displayed.

4 Click the **Start** Button to begin monitoring packets.


Packet monitor status: Monitor stopping

5 Click the **Stop** Button to stop monitoring packets.



Packet Monitor

Packet monitor status: Monitoring

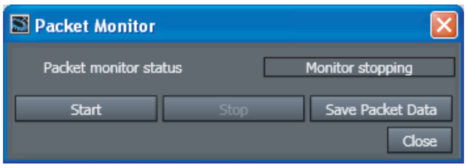
Saving Packet Data

● Reading Data from the Sysmac Studio

After you stop packet monitoring, you can use the Sysmac Studio to read the packet data and save it in a file.

1 Stop monitoring packets.

2 Click the **Save Packet Data** Button to save the packet data.



Packet Monitor

Packet monitor status: Monitor stopping

The **Save** Dialog Box is displayed.

3 Input a file name, and then click the **Save** Button.
A TCPDump packet data file with a .cap extension is saved.

● Saving to an SD Memory Card from Memory in the CPU Unit

Stop monitoring packets, and then execute the EC_SaveMon (Save EtherCAT Packets) instruction to save a given quantity of collected packet data in memory in the CPU Unit.
Then, execute the EC_CopyMon (Transfer EtherCAT Packets) instruction to save the packet data that was saved in the CPU Unit to a file on an SD Memory Card inserted into the CPU Unit.
You can specify the name of the file that is saved in the SD Memory Card. The number of files is limited only by the space that is available on the SD Memory Card.

Additional Information

- Packet data in the CPU Unit's memory is not retained when the power is interrupted.
- Packet data cannot be saved while packets are being monitored.
- Packet monitoring cannot be started while saving packet data.
- If an SD Memory Card is not inserted and you execute the instruction to copy the data to the SD Memory Card, then an error is returned. If there is no SD Memory Card, only the one file in the CPU Unit's memory can be read to the Sysmac Studio.
- Packet monitoring starts when the power is turned ON.

Packet Monitoring Specifications

Item	Specification
Maximum data size of one packet data file	12 MB
Maximum number of packets that can be obtained	3,904 packets
Format of packet data file	TCPDump format (cap)
Time information	Recorded (unit: μs) The elapsed time starting when packet monitoring begins is recorded. (Packet monitoring can be started when power turns ON, for an EtherCAT instruction, or for a Sysmac Studio operation.)
Save location for packet data file	CPU Unit's system: 1 file SD Memory Card inserted in CPU Unit: Multiple files (up to capacity of SD Memory Card)

Sample Programming

This sample transfers EtherCAT communications packets to an SD Memory Card when an EtherCAT slave error occurs. The file name is 'PacketFile.'
The processing procedure is as follows:

1. The *_EC_ErrSta* (EtherCAT Error) system-defined variable is monitored and processing is started if an error occurs.

2. The EC_StopMon instruction is used to stop execution of packet monitoring for EtherCAT communications.

3. The EC_SaveMon instruction is used to save EtherCAT communications packet data to a file in the system of the CPU Unit.

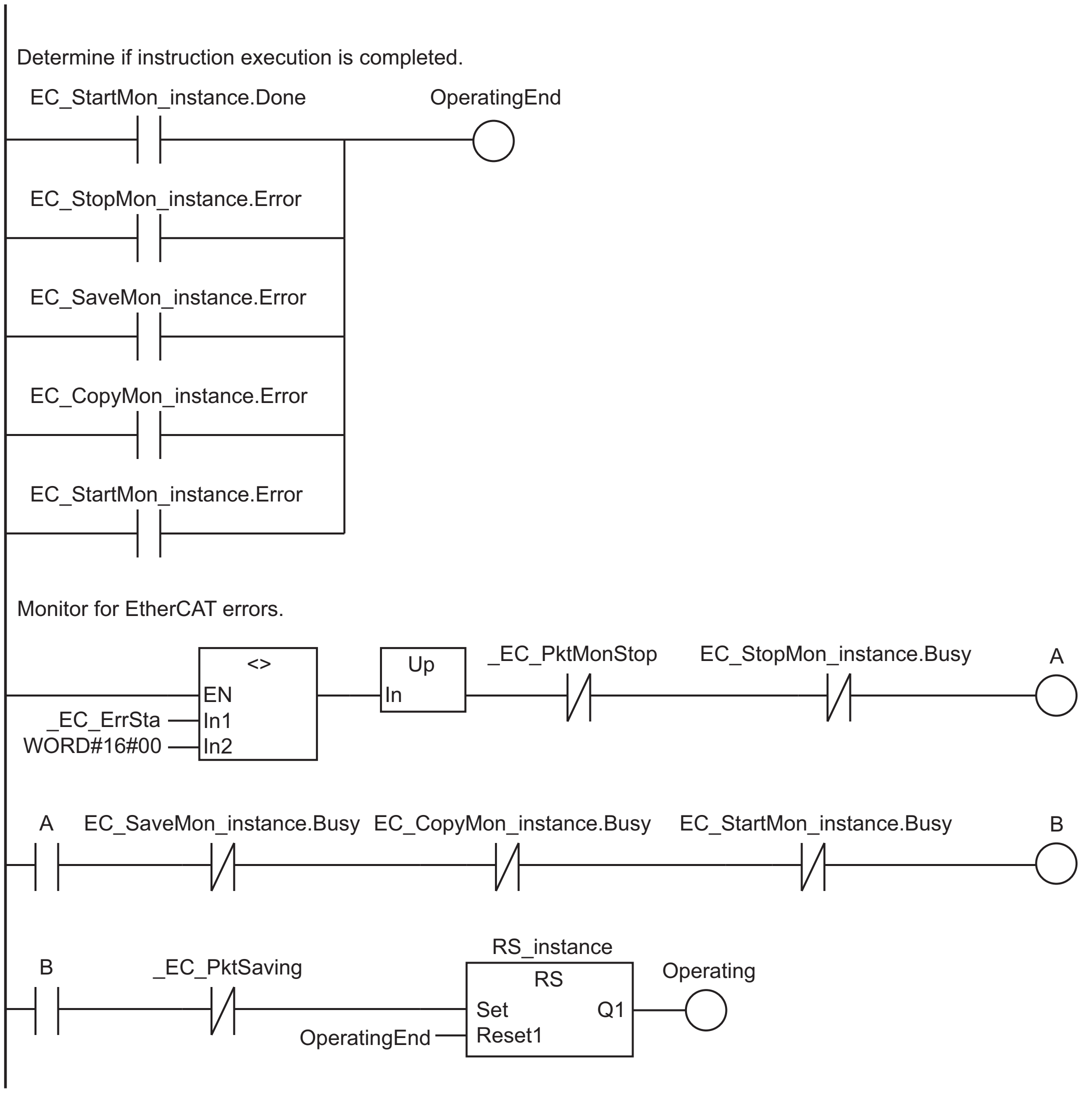
4. The EC_CopyMon instruction is used to copy that file to the SD Memory Card.

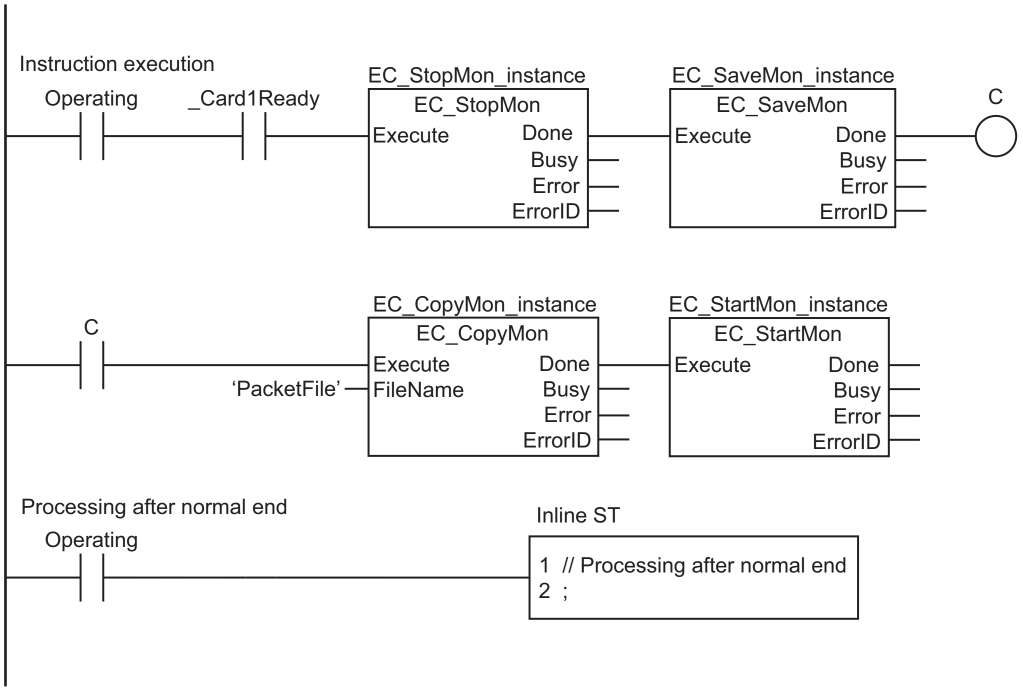
5. The EC_StartMon instruction is used to restart execution of packet monitoring for EtherCAT communications.

● LD

Internal Variables	Variable	Data type	Initial value	Comment
	OperatingEnd	BOOL	False	Processing completed
	Operating	BOOL	False	Execution condition
	A	BOOL	False	
	B	BOOL	False	
	C	BOOL	False	
	RS_instance	RS		
	EC_StopMon_instance	EC_StopMon		
	EC_SaveMon_instance	EC_SaveMon		
	EC_CopyMon_instance	EC_CopyMon		
	EC_StartMon_instance	EC_StartMon		

External Variables	Variable	Data type	Constant	Comment
	_EC_ErrSta	WORD	☑	EtherCAT Error
	_EC_PktMonStop	BOOL	☑	Packet Monitoring Stopped
	_EC_PktSaving	BOOL	☑	Saving Packet Data File
	_Card1Ready	BOOL	☑	SD Memory Card Ready Flag


Determine if instruction execution is completed.
EC_StartMon_instance.Done
OperatingEnd
EC_StopMon_instance.Error
EC_SaveMon_instance.Error
EC_CopyMon_instance.Error
EC_StartMon_instance.Error
Monitor for EtherCAT errors.
<>
EN
In1
In2
_EC_ErrSta
WORD#16#00
Up
In
_EC_PktMonStop
EC_StopMon_instance.Busy
A
A
EC_SaveMon_instance.Busy
EC_CopyMon_instance.Busy
EC_StartMon_instance.Busy
B
B
_EC_PktSaving
RS_instance
RS
Set
Reset1
Q1
OperatingEnd
Operating


Instruction execution
Operating
_Card1Ready
EC_StopMon_instance
EC_StopMon
Execute
Done
Busy
Error
ErrorID
EC_SaveMon_instance
EC_SaveMon
Execute
Done
Busy
Error
ErrorID
C
C
EC_CopyMon_instance
EC_CopyMon
Execute
FileName
'PacketFile'
Done
Busy
Error
ErrorID
EC_StartMon_instance
EC_StartMon
Execute
Done
Busy
Error
ErrorID
Processing after normal end
Operating
Inline ST
1 // Processing after normal end
2 ;

- **ST**

Internal Variables	Variable	Data type	Initial value	Comment
	EC_Err	BOOL	False	Controller error in the EtherCAT Master Function Module.
	EC_Err_Trigger	BOOL	False	Detect when EC_Err changes to TRUE.
	DoEC_PktSave	BOOL	False	Processing
	Stage	INT	0	Stage change
	R_TRIG_instance	R_TRIG		
	EC_StopMon_instance	EC_StopMon		
	EC_SaveMon_instance	EC_SaveMon		
	EC_CopyMon_instance	EC_CopyMon		
	EC_StartMon_instance	EC_StartMon		

External Variables	Variable	Data type	Constant	Comment
	_EC_ErrSta	WORD	☑	EtherCAT Error
	_EC_PktMonStop	BOOL	☑	Packet Monitoring Stopped
	_EC_PktSaving	BOOL	☑	Saving Packet Data File
	_Card1Ready	BOOL	☑	SD Memory Card Ready Flag

```
// Start sequence when _EC_ErrSta changes to TRUE.
EC_Err:=(_EC_ErrSta <> WORD#16#00);
R_TRIG_instance(Clk:=EC_Err, Q=>EC_Err_Trigger);

IF ( (EC_Err_Trigger=TRUE) AND (DoEC_PktSave=FALSE) AND (_EC_PktMonStop=FALSE)
    AND (_EC_PktSaving=FALSE) AND (_Card1Ready=TRUE) ) THEN
   DoEC_PktSave:=TRUE;
   Stage            :=INT#1;
   EC_StopMon_instance(Execute:=FALSE);    // Initialize instance.
   EC_SaveMon_instance(Execute:=FALSE);
   EC_CopyMon_instance(Execute:=FALSE);
   EC_StartMon_instance(Execute:=FALSE);
END_IF;

 // Instruction execution
IF (DoEC_PktSave=TRUE) THEN
   CASE Stage OF
   1 :                      // Stop EtherCAT packet monitor.
      EC_StopMon_instance(
         Execute    :=TRUE);

      IF (EC_StopMon_instance.Done=TRUE) THEN
         Stage:=INT#2;                  // Normal end
      ELSIF (EC_StopMon_instance.Error=TRUE) THEN
         Stage:=INT#10;                 // Error end
      END_IF;

   2 :                      // Save EtherCAT packet data to file in system.
      EC_SaveMon_instance(
         Execute :=TRUE);

      IF (EC_SaveMon_instance.Done=TRUE) THEN
         Stage:=INT#3;                  // Normal end
      ELSIF (EC_SaveMon_instance.Error=TRUE) THEN
         Stage:=INT#20;                 // Error end
      END_IF;

   3 :                      // Copy EtherCAT packet data file to the SD Memory Card.
      EC_CopyMon_instance(
         Execute  :=TRUE,
         FileName:='PacketFile');

      IF (EC_CopyMon_instance.Done=TRUE) THEN
         Stage:=INT#4;                  // Normal end
      ELSIF (EC_CopyMon_instance.Error=TRUE) THEN
         Stage:=INT#30;                 // Error end
      END_IF;

   4 :         // Restart EtherCAT packet monitoring.
      EC_StartMon_instance(
         Execute  :=TRUE);

      IF (EC_StartMon_instance.Done=TRUE) THEN
         Stage:=INT#0;   // Normal end
      ELSIF (EC_StartMon_instance.Error=TRUE) THEN
         Stage:=INT#40;  // Error end
      END_IF;

   0 :                     // Processing after normal end
      DoEC_PktSave:=FALSE;

   ELSE                    // Processing after error end
      DoEC_PktSave:=FALSE;
   END_CASE;
END_IF:
```

A-3 Multi-vendor Environments

This section provides precautions and describes documentation for multi-vendor environments.

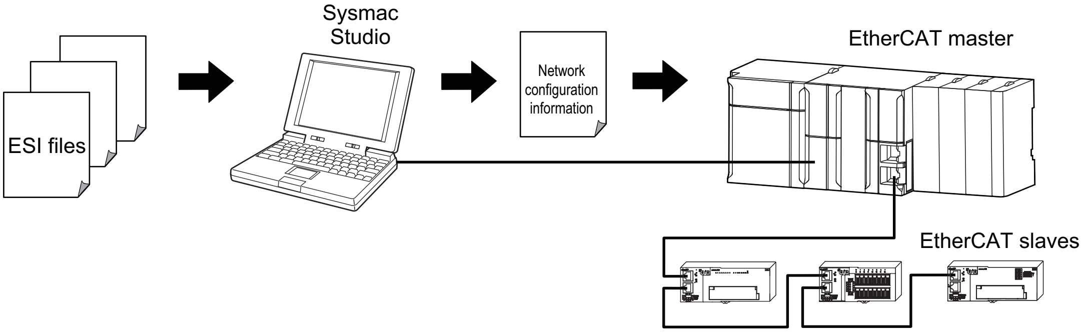
A-3-1 EtherCAT Slave Information File (ESI Files)

Setting information for EtherCAT is defined in EtherCAT slave information (ESI) files. These files are provided by the individual slave manufacturers. Various EtherCAT communications settings are defined based on the ESI definitions of connected slaves.

You can install the ESI files in the Sysmac Studio to use them to edit the EtherCAT configuration in the Sysmac Studio and create the network configuration information.

You can download the network configuration information to the EtherCAT master to configure the EtherCAT network.

Contact the manufacturer or the point of purchase to obtain the most recent ESI files for slaves that are manufactured by other companies.



Communications are started according to the communications settings and the network configuration in the ESI files that are installed.

Additional Information

ESI file versions that are lower than ESI version-1.0.1 specifications are not supported. If you install an ESI file version that is not supported, a message saying that you cannot use the slave is displayed by the Sysmac Studio.

The NJ/NX-series CPU Units do not support variables with some of the data types that are defined in ETG.1020. If variables with any unsupported data types are included in the slave information (ESI), you cannot use the slave.

The following table gives the data types that are supported by different Sysmac Studio versions.

Data type	Sysmac Studio Ver.		
	1.09 or higher	1.08	1.07 or lower
BIT1, BIT2, BIT3, BIT4, BIT5, BIT6, BIT7, and BIT8	Supported.	Supported.	Not supported.
BITARR8, BITARR16, and BITARR32	Supported.	Supported.	Not supported.
ARRAY[0..n] OF BYTE	Supported.	Not supported.[*1]	Not supported.[*1]
ARRAY[0..n] OF UINT	Supported.	Not supported.[*1]	Not supported.[*1]
INT24, INT40, INT48, and INT56	Not supported.	Not supported.	Not supported.
UINT24, UINT40, UINT48, and UINT56	Not supported.	Not supported.	Not supported.
STRING(n)	Not supported.	Not supported.	Not supported.

[*1] With Sysmac Studio version 1.06, 1.07, or 1.08, you can read ESI files and assign device variables. However, you cannot monitor them in the I/O Map.

If variables with any of these data types are included in the slave information (ESI), a message saying that you cannot use the slave is displayed in the **ESI Library** Dialog Box of the Sysmac Studio.

A-3-2 Connecting Slaves from Other Manufacturers to an OMRON Master

You can install the ESI file for a slave from another manufacturer in the Sysmac Studio to handle the slave in the same way as an OMRON slave. (Only the ESI files for OMRON slaves for which connectability has been confirmed are installed in the Sysmac Studio in advance.)

When connecting a slave from another manufacturer to an OMRON master, refer to the manuals for the other manufacturer's slaves, and then ask your OMRON representative if you have any questions.

Additional Information

EtherCAT setup software that is provided by other manufacturers cannot be connected to NJ/NX-series CPU Units.

A-3-3 Installing ESI files

● Procedure to Install ESI Files

An ESI (EtherCAT slave information) file is an XML file that describes the connection information and profile of the EtherCAT slave.

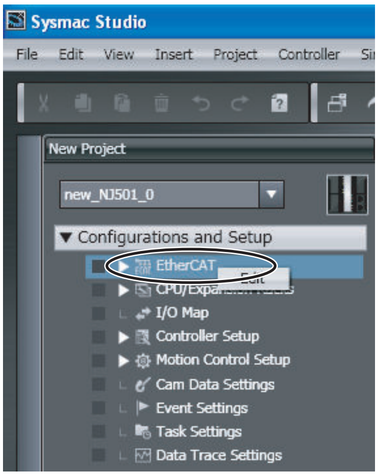
To connect an EtherCAT slave that is manufactured by other company to an NJ/NX-series master, you must install the ESI file for that slave in the Sysmac Studio to enable setting the slave.

Precautions for Correct Use

Obtain the ESI file to install from the slave manufacturer. The ESI file must conform to the most recent ETG ESI specifications.

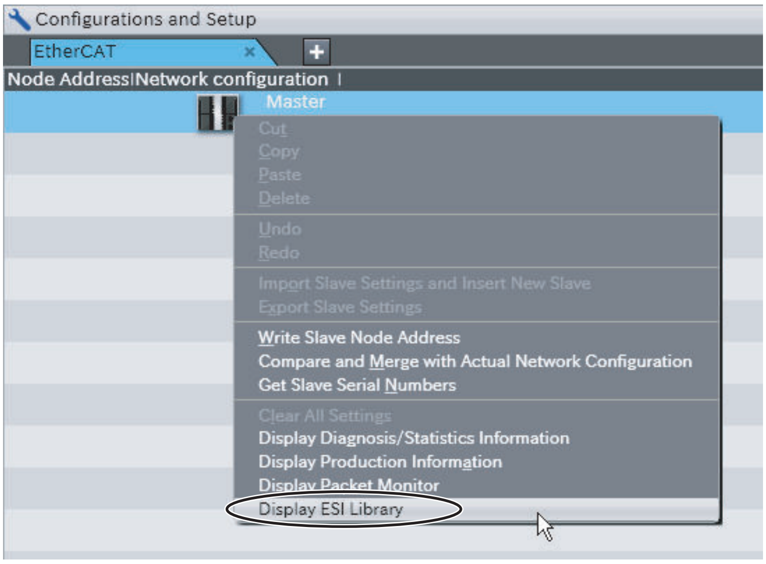
Use the following procedure to install an ESI file.

1 Double-click **EtherCAT** under **Configurations and Setup** on the Multiview Explorer. Or, right-click **EtherCAT** under **Configurations and Setup** and select **Edit**.



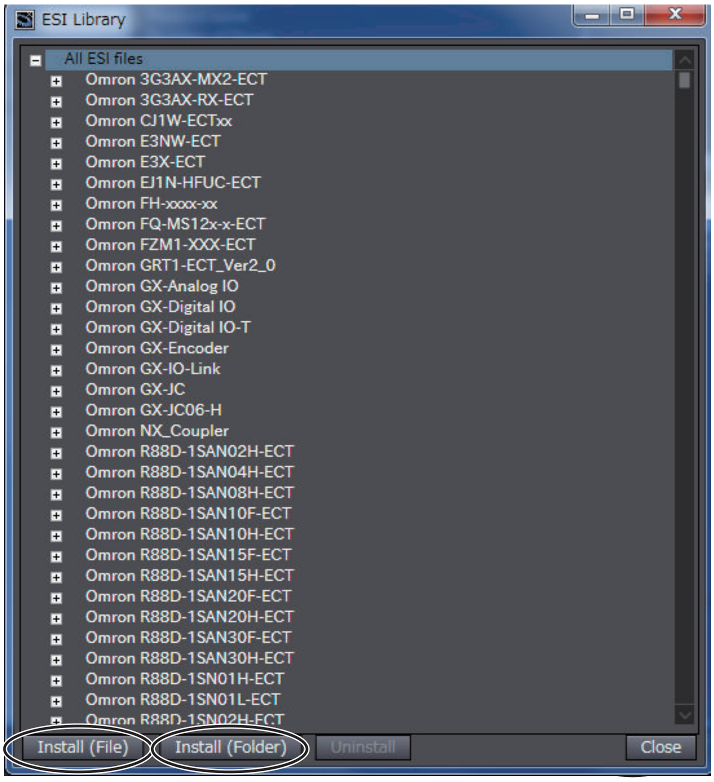
The EtherCAT Tab Page is displayed.

2 Right-click the EtherCAT master that is displayed in the EtherCAT Tab Page and select **Display ESI Library**.


Configurations and Setup
EtherCAT
Node Address|Network configuration
Master
Cut
Copy
Paste
Delete
Undo
Redo
Import Slave Settings and Insert New Slave
Export Slave Settings
Write Slave Node Address
Compare and Merge with Actual Network Configuration
Get Slave Serial Numbers
Clear All Settings
Display Diagnosis/Statistics Information
Display Production Information
Display Packet Monitor
Display ESI Library

The **ESI Library** Dialog Box is displayed.

3 Click the **Install (File)** or **Install (Folder)** Button.
If you install the ESI files individually, click the **Install (File)** Button. If you install all the ESI files in the folder simultaneously, click the **Install (Folder)** Button.



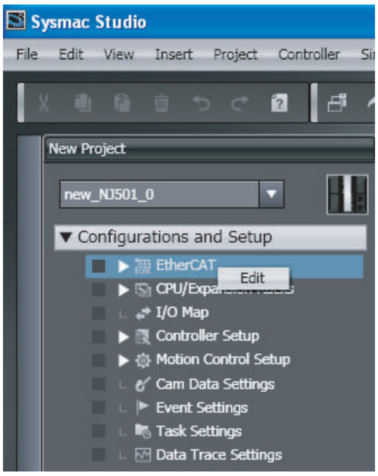
The dialog box to select folders or files is displayed.

4 Select the ESI file and then click the **Open** Button, or select the folder in which ESI files are stored and then click the **OK** Button.
The ESI files are installed and the corresponding slaves are displayed in the **ESI Library** Dialog Box.

● Procedure to Confirm ESI File Installation

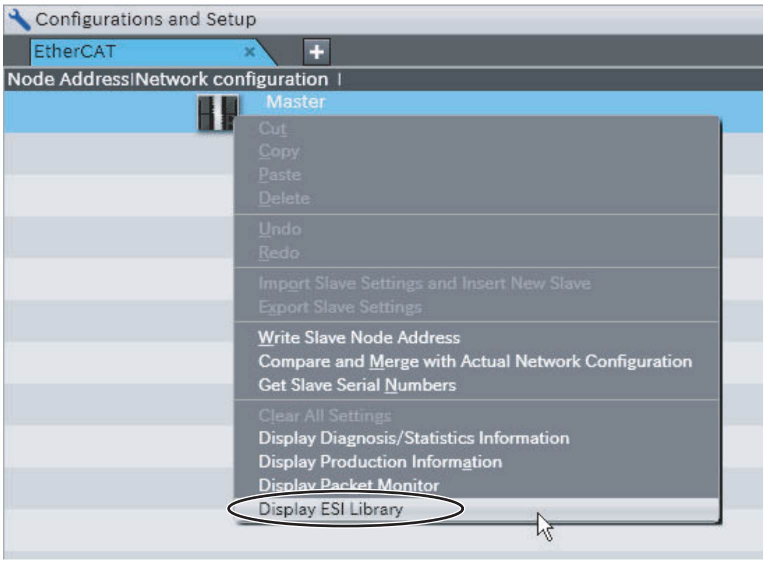
Use the following procedure to confirm that an ESI file was installed correctly

1 Double-click **EtherCAT** under **Configurations and Setup** on the Multiview Explorer. Or, right-click **EtherCAT** under **Configurations and Setup** and select **Edit**.



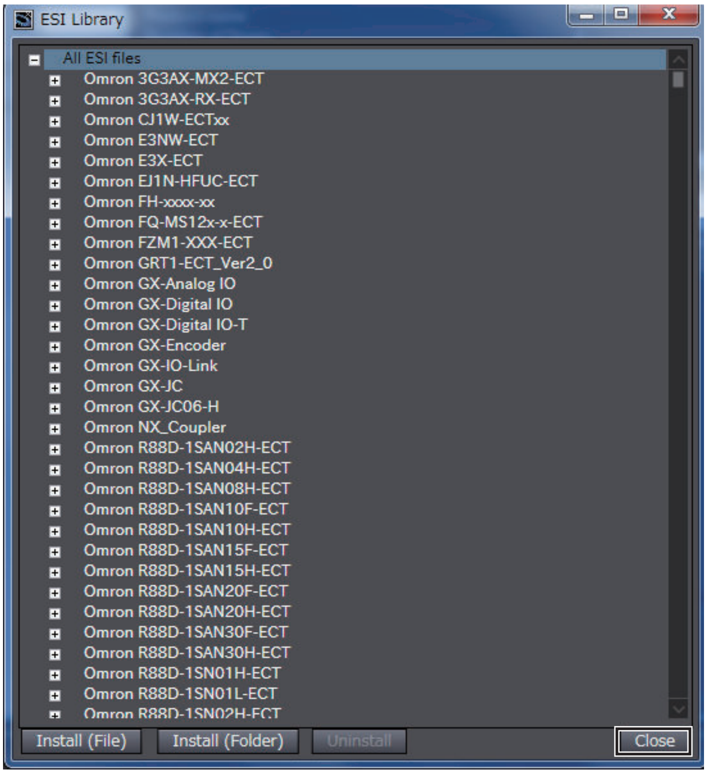
The EtherCAT Tab Page is displayed.

2 Right-click the EtherCAT master that is displayed in the EtherCAT Tab Page and select **Display ESI Library**.


Configurations and Setup
EtherCAT
Node Address|Network configuration
Master
Cut
Copy
Paste
Delete
Undo
Redo
Import Slave Settings and Insert New Slave
Export Slave Settings
Write Slave Node Address
Compare and Merge with Actual Network Configuration
Get Slave Serial Numbers
Clear All Settings
Display Diagnosis/Statistics Information
Display Production Information
Display Packet Monitor
Display ESI Library

The **ESI Library** Dialog Box is displayed.

3 Click the ⊞ Icon to the left of the name of the ESI file that was added.

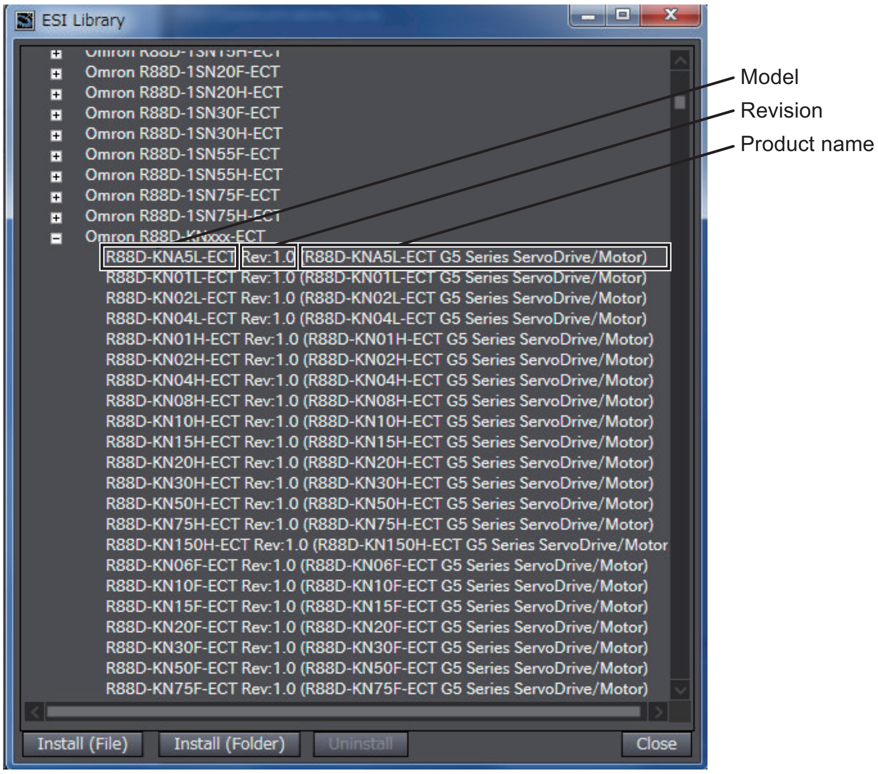


ESI Library

- All ESI files
  - Omron 3G3AX-MX2-ECT
  - Omron 3G3AX-RX-ECT
  - Omron CJ1W-ECTxx
  - Omron E3NW-ECT
  - Omron E3X-ECT
  - Omron EJ1N-HFUC-ECT
  - Omron FH-xxxx-xx
  - Omron FQ-MS12x-x-ECT
  - Omron FZM1-XXX-ECT
  - Omron GRT1-ECT_Ver2_0
  - Omron GX-Analog IO
  - Omron GX-Digital IO
  - Omron GX-Digital IO-T
  - Omron GX-Encoder
  - Omron GX-IO-Link
  - Omron GX-JC
  - Omron GX-JC06-H
  - Omron NX_Coupler
  - Omron R88D-1SAN02H-ECT
  - Omron R88D-1SAN04H-ECT
  - Omron R88D-1SAN08H-ECT
  - Omron R88D-1SAN10F-ECT
  - Omron R88D-1SAN10H-ECT
  - Omron R88D-1SAN15F-ECT
  - Omron R88D-1SAN15H-ECT
  - Omron R88D-1SAN20F-ECT
  - Omron R88D-1SAN20H-ECT
  - Omron R88D-1SAN30F-ECT
  - Omron R88D-1SAN30H-ECT
  - Omron R88D-1SN01H-ECT
  - Omron R88D-1SN01L-ECT
  - Omron R88D-1SN02H-ECT

The slave definitions in the ESI file are expanded so that you can check the following items.

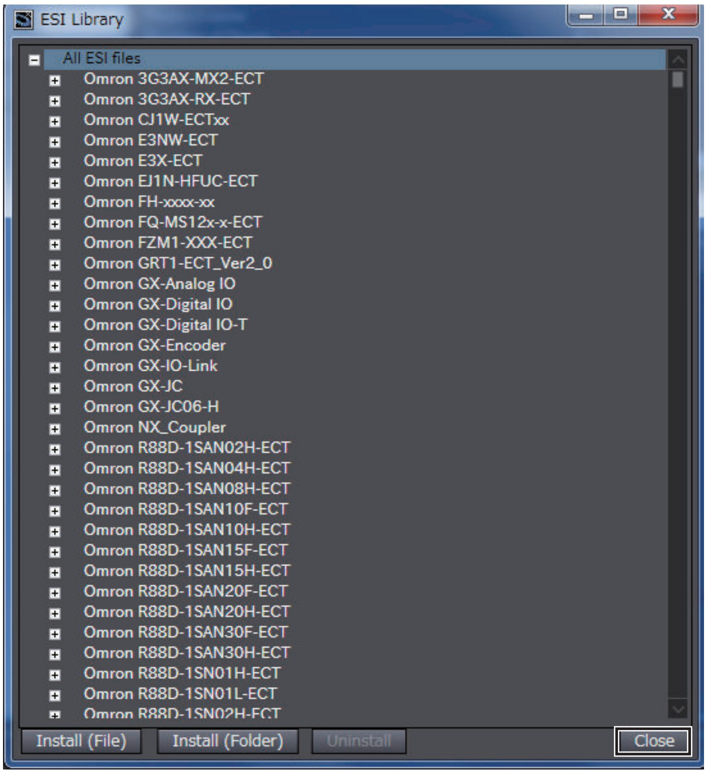
- Model
- Revision
- Product name



ESI Library

- Omron R88D-1SN15H-ECT
- Omron R88D-1SN20F-ECT
- Omron R88D-1SN20H-ECT
- Omron R88D-1SN30F-ECT
- Omron R88D-1SN30H-ECT
- Omron R88D-1SN55F-ECT
- Omron R88D-1SN55H-ECT
- Omron R88D-1SN75F-ECT
- Omron R88D-1SN75H-ECT
- Omron R88D-KNxxx-ECT
  - R88D-KNA5L-ECT Rev:1.0 (R88D-KNA5L-ECT G5 Series ServoDrive/Motor)
  - R88D-KN01L-ECT Rev:1.0 (R88D-KN01L-ECT G5 Series ServoDrive/Motor)
  - R88D-KN02L-ECT Rev:1.0 (R88D-KN02L-ECT G5 Series ServoDrive/Motor)
  - R88D-KN04L-ECT Rev:1.0 (R88D-KN04L-ECT G5 Series ServoDrive/Motor)
  - R88D-KN01H-ECT Rev:1.0 (R88D-KN01H-ECT G5 Series ServoDrive/Motor)
  - R88D-KN02H-ECT Rev:1.0 (R88D-KN02H-ECT G5 Series ServoDrive/Motor)
  - R88D-KN04H-ECT Rev:1.0 (R88D-KN04H-ECT G5 Series ServoDrive/Motor)
  - R88D-KN08H-ECT Rev:1.0 (R88D-KN08H-ECT G5 Series ServoDrive/Motor)
  - R88D-KN10H-ECT Rev:1.0 (R88D-KN10H-ECT G5 Series ServoDrive/Motor)
  - R88D-KN15H-ECT Rev:1.0 (R88D-KN15H-ECT G5 Series ServoDrive/Motor)
  - R88D-KN20H-ECT Rev:1.0 (R88D-KN20H-ECT G5 Series ServoDrive/Motor)
  - R88D-KN30H-ECT Rev:1.0 (R88D-KN30H-ECT G5 Series ServoDrive/Motor)
  - R88D-KN50H-ECT Rev:1.0 (R88D-KN50H-ECT G5 Series ServoDrive/Motor)
  - R88D-KN75H-ECT Rev:1.0 (R88D-KN75H-ECT G5 Series ServoDrive/Motor)
  - R88D-KN150H-ECT Rev:1.0 (R88D-KN150H-ECT G5 Series ServoDrive/Motor
  - R88D-KN06F-ECT Rev:1.0 (R88D-KN06F-ECT G5 Series ServoDrive/Motor)
  - R88D-KN10F-ECT Rev:1.0 (R88D-KN10F-ECT G5 Series ServoDrive/Motor)
  - R88D-KN15F-ECT Rev:1.0 (R88D-KN15F-ECT G5 Series ServoDrive/Motor)
  - R88D-KN20F-ECT Rev:1.0 (R88D-KN20F-ECT G5 Series ServoDrive/Motor)
  - R88D-KN30F-ECT Rev:1.0 (R88D-KN30F-ECT G5 Series ServoDrive/Motor)
  - R88D-KN50F-ECT Rev:1.0 (R88D-KN50F-ECT G5 Series ServoDrive/Motor)
  - R88D-KN75F-ECT Rev:1.0 (R88D-KN75F-ECT G5 Series ServoDrive/Motor)

4 If an exclamation mark ! is not displayed, click the **Close** Button.



- All ESI files
  - Omron 3G3AX-MX2-ECT
  - Omron 3G3AX-RX-ECT
  - Omron CJ1W-ECTxx
  - Omron E3NW-ECT
  - Omron E3X-ECT
  - Omron EJ1N-HFUC-ECT
  - Omron FH-xxxx-xx
  - Omron FQ-MS12x-x-ECT
  - Omron FZM1-XXX-ECT
  - Omron GRT1-ECT_Ver2_0
  - Omron GX-Analog IO
  - Omron GX-Digital IO
  - Omron GX-Digital IO-T
  - Omron GX-Encoder
  - Omron GX-IO-Link
  - Omron GX-JC
  - Omron GX-JC06-H
  - Omron NX_Coupler
  - Omron R88D-1SAN02H-ECT
  - Omron R88D-1SAN04H-ECT
  - Omron R88D-1SAN08H-ECT
  - Omron R88D-1SAN10F-ECT
  - Omron R88D-1SAN10H-ECT
  - Omron R88D-1SAN15F-ECT
  - Omron R88D-1SAN15H-ECT
  - Omron R88D-1SAN20F-ECT
  - Omron R88D-1SAN20H-ECT
  - Omron R88D-1SAN30F-ECT
  - Omron R88D-1SAN30H-ECT
  - Omron R88D-1SN01H-ECT
  - Omron R88D-1SN01L-ECT
  - Omron R88D-1SN02H-ECT

Additional Information

If an ESI file for a slave cannot be used in the Sysmac Studio, an exclamation mark ❗ is displayed to the left of the file name. If an exclamation mark ❗ is displayed, obtain an ESI file with the cause corrected from the slave manufacturer, and then install the ESI file again.

● Procedure to Uninstall ESI Files

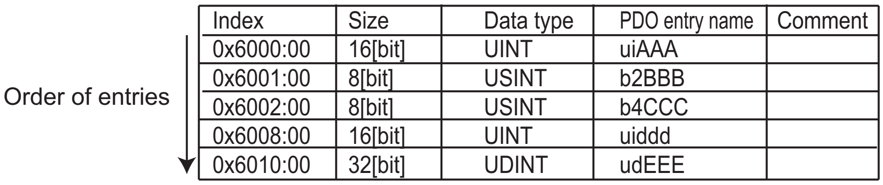
Use the following procedure to uninstall the installed ESI file.

1 In the **ESI Library** Dialog Box, select the slave for which you want to uninstall the ESI file and click the **Uninstall** Button.
The dialog box to confirm execution is displayed.

2 Click the **Yes** Button.
The ESI file is uninstalled and the corresponding slaves are removed from the **ESI Library** Dialog Box.

A-3-4 Editing PDO Entry Tables

A PDO entry table maps objects in a slave to the process data that is used for communications between the master and the slave.
You cannot necessarily edit all of the PDO mappings in a PDO entry table.
You can add and delete objects in a PDO entry table or you can change the order of the entries.
The Sysmac Studio provides the functionality to edit slave PDO entry tables.



Order of entries

Index	Size	Data type	PDO entry name	Comment
0x6000:00	16[bit]	UINT	uiAAA	
0x6001:00	8[bit]	USINT	b2BBB	
0x6002:00	8[bit]	USINT	b4CCC	
0x6008:00	16[bit]	UINT	uiddd	
0x6010:00	32[bit]	UDINT	udEEE	

Editing a PDO Entry Table

Initially, a slave PDO entry table contains the default objects. Some slaves have objects other than these default objects.

You can add objects other than the default objects to the PDO entry table, or you can delete objects from the PDO entry table.

However, the following restrictions apply:

● Byte Boundaries

An OMRON master must arrange entry objects by byte boundaries. (This applies only to objects that are one byte or larger in size.)

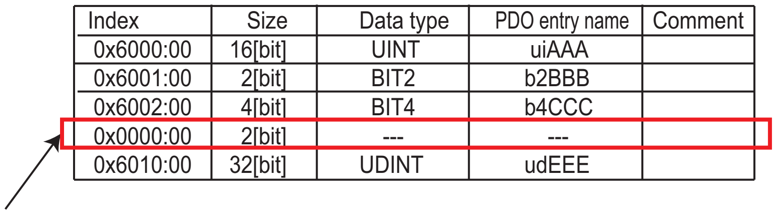
If an entry object is bit data, padding must be entered to adjust the boundaries.

Padding is meaningless data that is added to the end of an entry so that an object ends at a boundary of the specified size.

You can use the following methods to arrange objects within byte boundaries.

- Click the **Align** Button.
  Padding is automatically entered. Refer to the descriptions of the buttons below for details.
- Add padding to the PDO entries manually.

In the following example, 2 bits of padding are entered to adjust the byte boundary when entering objects with BIT2 and BIT4 data types for PDO entries.



Index	Size	Data type	PDO entry name	Comment
0x6000:00	16[bit]	UINT	uiAAA	
0x6001:00	2[bit]	BIT2	b2BBB	
0x6002:00	4[bit]	BIT4	b4CCC	
0x0000:00	2[bit]	---	---	
0x6010:00	32[bit]	UDINT	udEEE	

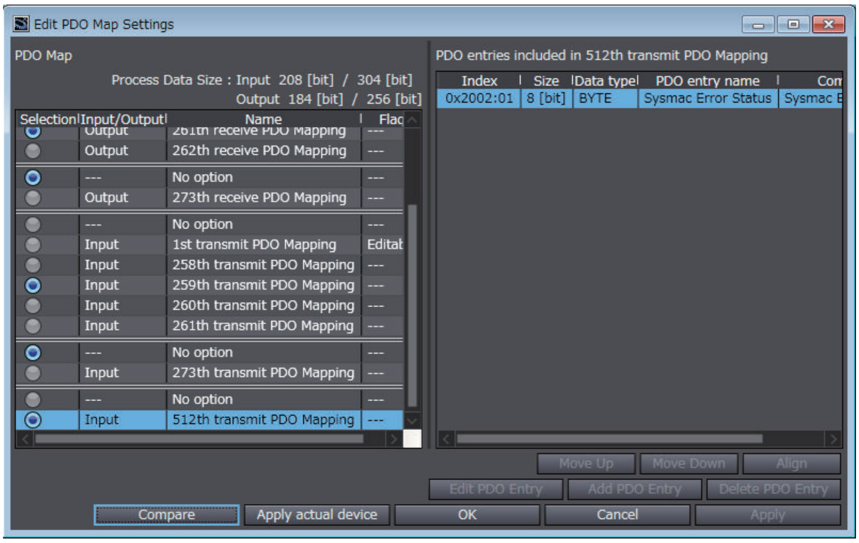
Because an object with 2 bits and an object with 4 bits are entered,
2 bits of padding must be added to adjust the byte boundary (8 bits).

● Connecting to Slaves from Other Manufacturers

When connecting a slave from another manufacturer, refer to the manuals for the other manufacturer's slaves, and then ask your OMRON representative if you have any questions.

● Specifications of Edit PDO Entry Tables Window

The following buttons are used to edit a PDO entry table. The meanings of the buttons are given below.



Edit PDO Map Settings

PDO Map

Process Data Size : Input 208 [bit] / 304 [bit]
Output 184 [bit] / 256 [bit]

Selection	Input/Output	Name	Flag
●	Output	261th receive PDO Mapping	---
○	Output	262th receive PDO Mapping	---
●	---	No option	---
○	Output	273th receive PDO Mapping	---
○	---	No option	---
○	Input	1st transmit PDO Mapping	Editat
○	Input	258th transmit PDO Mapping	---
●	Input	259th transmit PDO Mapping	---
○	Input	260th transmit PDO Mapping	---
○	Input	261th transmit PDO Mapping	---
●	---	No option	---
○	Input	273th transmit PDO Mapping	---
○	---	No option	---
●	Input	512th transmit PDO Mapping	---

PDO entries included in 512th transmit PDO Mapping

Index	Size	Data type	PDO entry name	Com
0x2002:01	8 [bit]	BYTE	Sysmac Error Status	Sysmac E

Button	Description
Add PDO Entry Button	Use these buttons to add objects to and delete objects from the PDO entry table. “0x0000:00” is for padding. The default size is 8 bits.
Delete PDO Entry Button	
Edit PDO Entry Button	Use this button to change the size of padding (i.e., the number of bits) in the PDO mapping.
Move Up Button	Use these buttons to change the order of the PDO entries by moving objects up and down.
Move Down Button	
Align Button	Use this button to automatically order the PDO entries and adjust byte boundaries. The PDO entires are ordered according to the type, descending PDO entry size, ascending indices, and then ascending subindices. For the type, data arranged by byte boundaries (INT, BYTE, STRING, etc.) comes first, followed by data arranged by bit boundaries (BIT, BIT8, etc.).
Compare Button*1*2	Use this button to compare the PDO map settings of the project and that of the actual slave.
Apply actual device Button *1 *2	Use this button to upload the PDO map settings from the actual slave and update only the PDO mapping data in the PDO Map Settings Window.

*1. This button is displayed only when Sysmac Studio is connected online with the CPU Unit.

*2. This button is not displayed in MDP-compatible slave setting.

Version Information

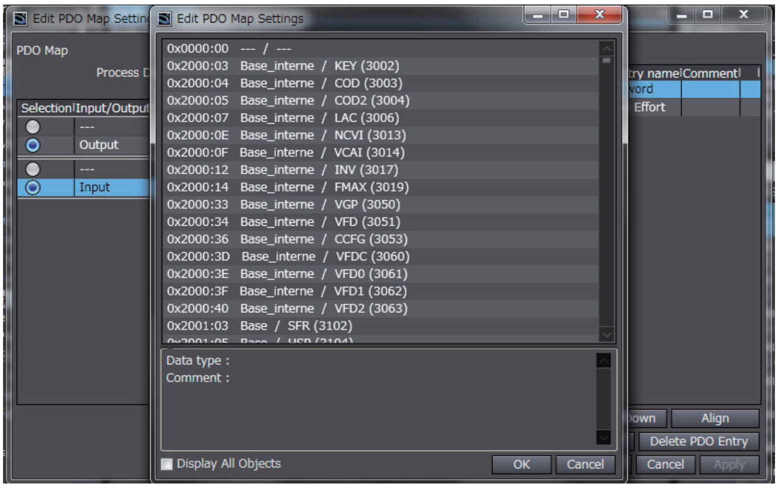
- Sysmac Studio version 1.08 or higher is required to use the following buttons to edit PDO entry tables: **Edit PDO Entry**, **Move Up**, **Move Down**, and **Align** Buttons.
- Sysmac Studio version 1.08 or higher is required to add “0x0000:00” to a PDO entry table.
- Sysmac Studio version 1.22 or higher is required to use the following buttons to edit PDO entry tables: **Compare** and **Apply Actual Device** Buttons.

Editing Methods for PDO Entries

Adding PDO Entires

Click the **Add PDO Entry** Button. The entry table from the slave, which is a list of objects defined in the ESI files of the slave, is displayed. Select the object to add with the cursor and click the **OK** Button to add it.

The selected object is added at the location of the cursor in the PDO entry table.



0x0000:00 --- / ---
0x2000:03 Base_interne / KEY (3002)
0x2000:04 Base_interne / COD (3003)
0x2000:05 Base_interne / COD2 (3004)
0x2000:07 Base_interne / LAC (3006)
0x2000:0E Base_interne / NCVI (3013)
0x2000:0F Base_interne / VCAI (3014)
0x2000:12 Base_interne / INV (3017)
0x2000:14 Base_interne / FMAX (3019)
0x2000:33 Base_interne / VGP (3050)
0x2000:34 Base_interne / VFD (3051)
0x2000:36 Base_interne / CCFG (3053)
0x2000:3D Base_interne / VFDC (3060)
0x2000:3E Base_interne / VFD0 (3061)
0x2000:3F Base_interne / VFD1 (3062)
0x2000:40 Base_interne / VFD2 (3063)
0x2001:03 Base / SFR (3102)

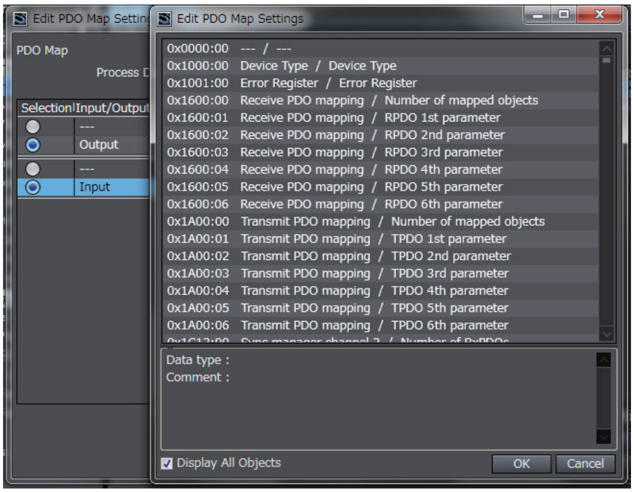
Data type :
Comment :

☐ Display All Objects

Select the **Display All Objects** Check Box to display all objects in the slave.

Version Information

Sysmac Studio version 1.23 or higher is required to display all objects in the slave.



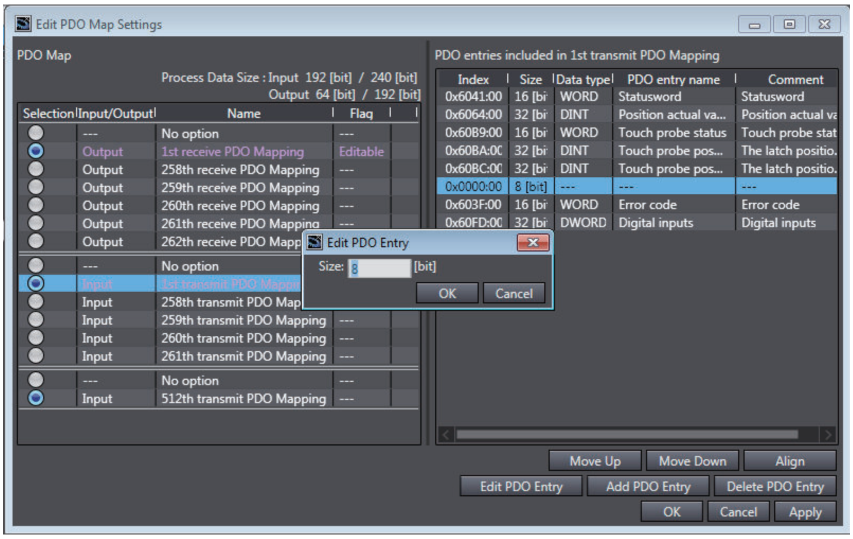
0x0000:00 --- / ---
0x1000:00 Device Type / Device Type
0x1001:00 Error Register / Error Register
0x1600:00 Receive PDO mapping / Number of mapped objects
0x1600:01 Receive PDO mapping / RPDO 1st parameter
0x1600:02 Receive PDO mapping / RPDO 2nd parameter
0x1600:03 Receive PDO mapping / RPDO 3rd parameter
0x1600:04 Receive PDO mapping / RPDO 4th parameter
0x1600:05 Receive PDO mapping / RPDO 5th parameter
0x1600:06 Receive PDO mapping / RPDO 6th parameter
0x1A00:00 Transmit PDO mapping / Number of mapped objects
0x1A00:01 Transmit PDO mapping / TPDO 1st parameter
0x1A00:02 Transmit PDO mapping / TPDO 2nd parameter
0x1A00:03 Transmit PDO mapping / TPDO 3rd parameter
0x1A00:04 Transmit PDO mapping / TPDO 4th parameter
0x1A00:05 Transmit PDO mapping / TPDO 5th parameter
0x1A00:06 Transmit PDO mapping / TPDO 6th parameter

Data type :
Comment :

☑ Display All Objects

Editing PDO Entries

Select padding (0x0000:00) in the PDO entry table and click the **Edit PDO Entry** Button. Enter the size and click the **OK** Button. The size of the padding will change.



Edit PDO Map Settings

PDO Map

Process Data Size : Input 192 [bit] / 240 [bit]
Output 64 [bit] / 192 [bit]

Selection	Input/Output	Name	Flag
○	---	No option	---
●	Output	1st receive PDO Mapping	Editable
○	Output	258th receive PDO Mapping	---
○	Output	259th receive PDO Mapping	---
○	Output	260th receive PDO Mapping	---
○	Output	261th receive PDO Mapping	---
○	Output	262th receive PDO Mapp	
○	---	No option	
●	Input	1st transmit PDO Mappi	
○	Input	258th transmit PDO Map	
○	Input	259th transmit PDO Mapping	---
○	Input	260th transmit PDO Mapping	---
○	Input	261th transmit PDO Mapping	---
○	---	No option	---
●	Input	512th transmit PDO Mapping	---

PDO entries included in 1st transmit PDO Mapping

Index	Size	Data type	PDO entry name	Comment
0x6041:00	16 [bi	WORD	Statusword	Statusword
0x6064:00	32 [bi	DINT	Position actual va...	Position actual va
0x60B9:00	16 [bi	WORD	Touch probe status	Touch probe stat
0x60BA:0C	32 [bi	DINT	Touch probe pos...	The latch positio.
0x60BC:0C	32 [bi	DINT	Touch probe pos...	The latch positio.
0x0000:00	8 [bit]	---	---	---
0x603F:00	16 [bi	WORD	Error code	Error code
0x60FD:0C	32 [bi	DWORD	Digital inputs	Digital inputs

Edit PDO Entry

Size: 8 [bit]

Comparing and Uploading the PDO Mapping

You can upload the PDO map settings from the actual slave or compare with the PDO map settings of the actual slave. The procedure to compare and upload the PDO map settings is described below.

Version Information

Sysmac Studio version 1.22 or higher is required to compare and upload the PDO mapping.

Precautions for Correct Use

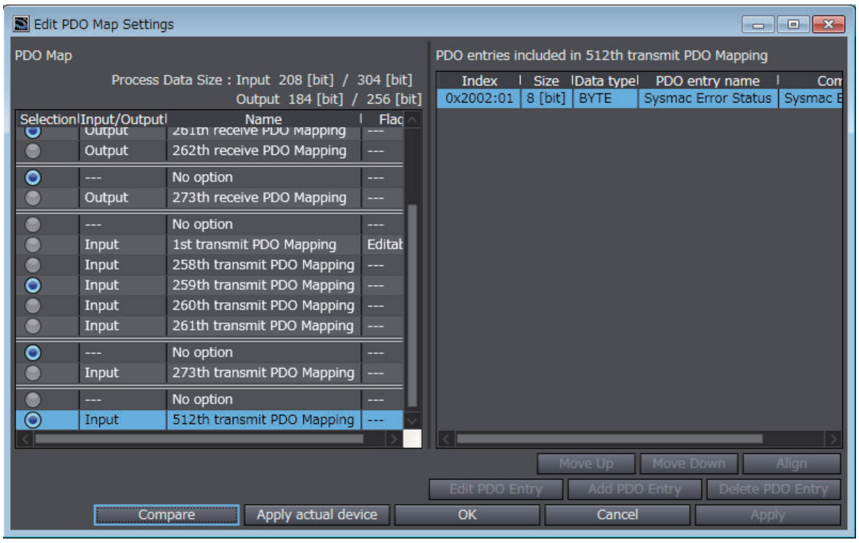
When the Sysmac Studio is online, you cannot change PDO mapping selection or edit PDO entries.

Additional Information

- The PDO Map Settings Window can be displayed while the Sysmac Studio is online.
- You can execute to compare and upload the PDO mapping even when the slave is in the Init state.

Use the following procedure to compare and upload the PDO map settings.

1 In the **PDO Map Settings** of the EtherCAT slave settings, click the **Edit PDO Map Settings** Button.
The Edit PDO Map Settings Window is displayed.



Process Data Size : Input 208 [bit] / 304 [bit]
Output 184 [bit] / 256 [bit]

Selection	Input/Output	Name	Flag
●	Output	261th receive PDO Mapping	---
○	Output	262th receive PDO Mapping	---
●	---	No option	---
○	Output	273th receive PDO Mapping	---
○	---	No option	---
○	Input	1st transmit PDO Mapping	Editat
○	Input	258th transmit PDO Mapping	---
●	Input	259th transmit PDO Mapping	---
○	Input	260th transmit PDO Mapping	---
○	Input	261th transmit PDO Mapping	---
●	---	No option	---
○	Input	273th transmit PDO Mapping	---
○	---	No option	---
●	Input	512th transmit PDO Mapping	---

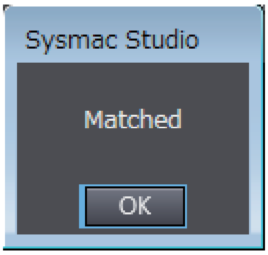
PDO entries included in 512th transmit PDO Mapping

Index	Size	Data type	PDO entry name	Con
0x2002:01	8 [bit]	BYTE	Sysmac Error Status	Sysmac E

● Comparing

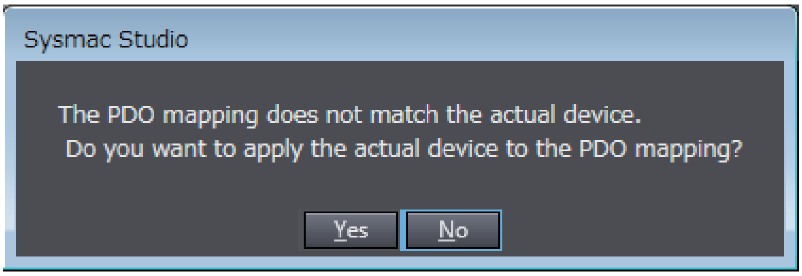
Click the **Compare** Button to read the PDO mapping list and PDO entry list from the actual slave and compare them with the PDO map settings in the project.

a. When the PDO mapping matches the actual device
The following dialog box is displayed.



Matched

b. When the PDO mapping does not match the actual device
The following dialog box is displayed.



The PDO mapping does not match the actual device.
Do you want to apply the actual device to the PDO mapping?

When the PDO mapping does not match the actual device, click the **Yes** Button to update the information in the PDO Map Settings Window to match the PDO mapping list and the PDO entry list of the actual slave. Click the **OK** Button or **Apply** Button in the Edit PDO Map Settings Window to update the project with the PDO mapping list and PDO entry list of the actual slave.

● Uploading

Click the **Apply actual device** Button to read the PDO mapping list and PDO entry list from the actual slave and update the information in the PDO Map Settings Window to match the PDO mapping list and the PDO entry list of the actual slave.

Click the **OK** Button or **Apply** Button in the Edit PDO Map Settings Window to update the project with the PDO mapping list and PDO entry list of the actual slave.

A-3-5 Settings for MDP-compatible Slaves from Other Manufacturers

The MDP (Modular Device Profile) defines the data structure for the settings in EtherCAT slaves. It is an EtherCAT specification. You can use slaves that support MDPs from other manufacturers on the EtherCAT Tab Page.

There are two sets of settings, slave and module, for MDP-compatible slaves.
The setting items and setting procedure for MDP-compatible slaves with the Sysmac Studio are described below.

Setting Items

This section describes each setting item for slave settings and module settings.

● Slave Setting Items

Some items in the slave settings only display the current setting. The other items in the slave settings are changeable.

a. Items for which the current setting is displayed only.
The names of items and their settings are as follows:

Name	Meaning of setting
Device name	The device name that is set in the EtherCAT configuration
Model	The model number of the slave
Product name	The product name of the slave
Revision	The revision of the slave
Number of modules	The number of modules that are connected

b. Items for which the current setting is changeable.

The items that you can edit are the **PDO Map Settings**, **Module config send method**, **Setting Parameters**, and **Backup Parameter Settings**. The meanings of these settings are given below.

PDO Map Settings

- The PDO Map Settings displays the settings for the process data assigned for the slave.
- When valid PDOs are present, a list of them is displayed.
- When valid PDOs are not present, "---" is displayed.
- Process data is assigned for each slave in default by the Sysmac Studio. To change the process data assignments, click the **Edit PDO Map Settings** Button. Refer to *A-3-4 Editing PDO Entry Tables* on page A-21 for details.

Module config send method

- This is the area to select whether to send the module configuration information to the device or not.

- When you select **Send**, module configuration information commands are created for the devices. However, the slaves must support this function.
- When you select **Do not send**, module configuration information commands are not created for the devices.
- The default is **Do not send**.

Setting Parameters

- The set values of the Setting Parameters are displayed.
- The Setting Parameters are assigned for each module in default by the Sysmac Studio. To edit the setting parameters, click the **Edit Setting Parameters** Button.

Backup Parameter Settings

- The set values of Backup Parameter Settings are displayed.
- The Backup Parameter Settings are assigned for each module in default by the Sysmac Studio. To edit the backup parameters, click the **Edit Backup Parameter Settings** Button.

Version Information

Sysmac Studio version 1.29 or higher is required to use the Setting Parameters and Backup Parameter Settings.

● Module Setting Items

Some items in the module settings only display the current setting. The other items in the module settings are changeable.

a. Items for which the current setting is displayed only.
The names of items and their settings are as follows:

Name	Meaning of setting
Model	The model name of the device
Product name	The product name of the device
Connected position	The connected position of the device

b. Items for which the current setting is changeable.
The items that you can edit are the **Device name**, **PDO Map Settings**, **Setting Parameters**, and **Backup Parameter Settings**. The meanings of these settings are given below.

Device name

- The name of the module is displayed.
- The default is M#. "#" is a serial number that starts from 1.

PDO Map Settings

- The PDO Map Settings displays the settings for the process data assigned for the module.
- When valid PDOs are present, a list of them is displayed.
- When valid PDOs are not present, "---" is displayed.
- Process data is assigned for each module in default by the Sysmac Studio. To change the process data assignments, click the **Edit PDO Map Settings** Button. Refer to *A-3-4 Editing PDO Entry Tables* on page A-21 for details.

Setting Parameters

- The set values of the Setting Parameters are displayed.

- The Setting Parameters are assigned for each module in default by the Sysmac Studio. To edit the setting parameters, click the **Edit Setting Parameters** Button.

Backup Parameter Settings

- The set values of Backup Parameter Settings are displayed.
- The Backup Parameter Settings are assigned for each module in default by the Sysmac Studio. To edit the backup parameters, click the **Edit Backup Parameter Settings** Button.

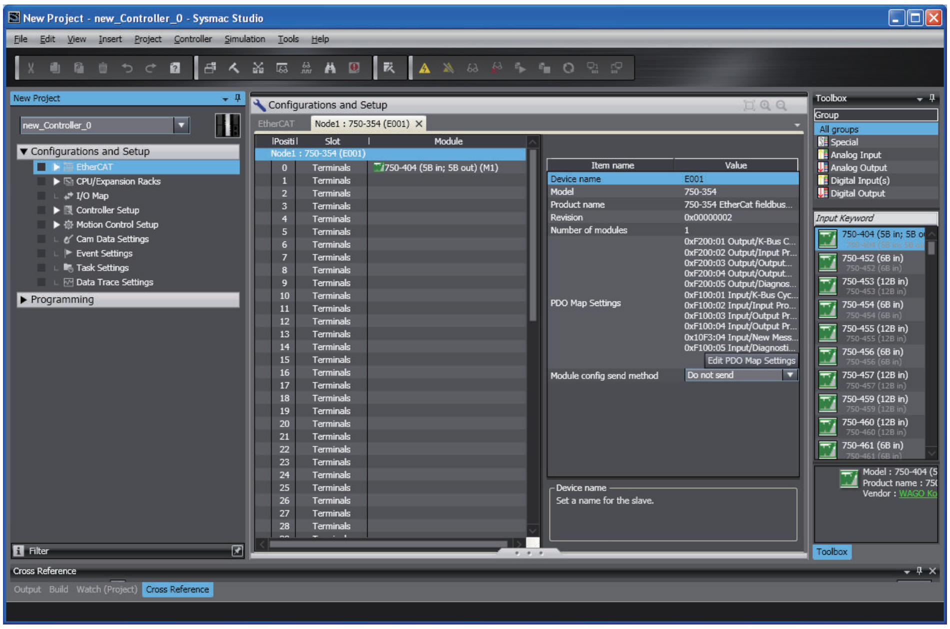
Version Information

Sysmac Studio version 1.29 or higher is required to use the Setting Parameters and Backup Parameter Settings.

Setting Procedure

1 Right-click the slave in the EtherCAT Tab Page and select **Edit Module Configuration**.

An edit pane for the module configuration is displayed.



Positi	Slot	Module
Node1 : 750-354 (E001)		
0	Terminals	750-404 (5B in; 5B out) (M1)
1	Terminals	
2	Terminals	
3	Terminals	
4	Terminals	
5	Terminals	
6	Terminals	
7	Terminals	
8	Terminals	
9	Terminals	
10	Terminals	
11	Terminals	
12	Terminals	
13	Terminals	
14	Terminals	
15	Terminals	
16	Terminals	
17	Terminals	
18	Terminals	
19	Terminals	
20	Terminals	
21	Terminals	
22	Terminals	
23	Terminals	
24	Terminals	
25	Terminals	
26	Terminals	
27	Terminals	
28	Terminals	

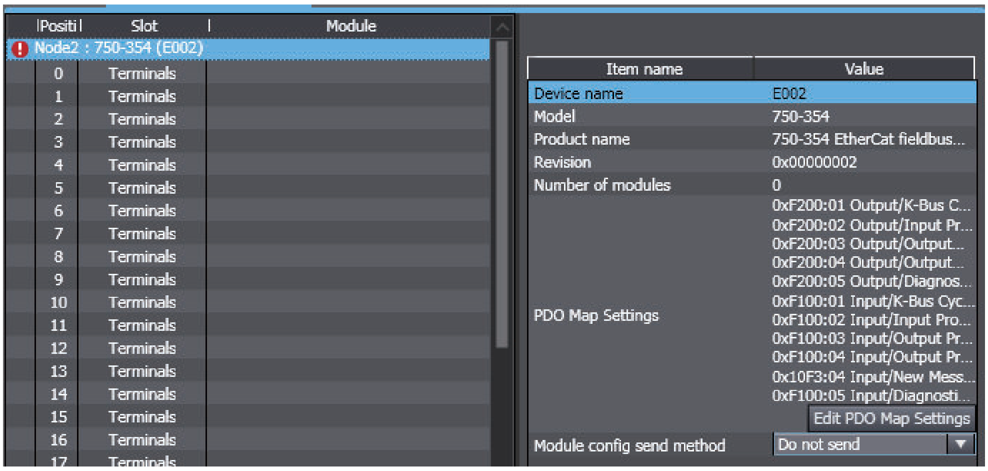
Item name	Value
Device name	E001
Model	750-354
Product name	750-354 EtherCat fieldbus...
Revision	0x00000002
Number of modules	1
PDO Map Settings	0xF200:01 Output/K-Bus C... 0xF200:02 Output/Input Pr... 0xF200:03 Output/Output... 0xF200:04 Output/Output... 0xF200:05 Output/Diagnos... 0xF100:01 Input/K-Bus Cyc... 0xF100:02 Input/Input Pro... 0xF100:03 Input/Output Pr... 0xF100:04 Input/Output Pr... 0x10F3:04 Input/New Mess... 0xF100:05 Input/Diagnosti... Edit PDO Map Settings
Module config send method	Do not send

Device name

Set a name for the slave.

2 Select a slave on the edit pane for the module configuration.

A list of the slave settings is displayed.

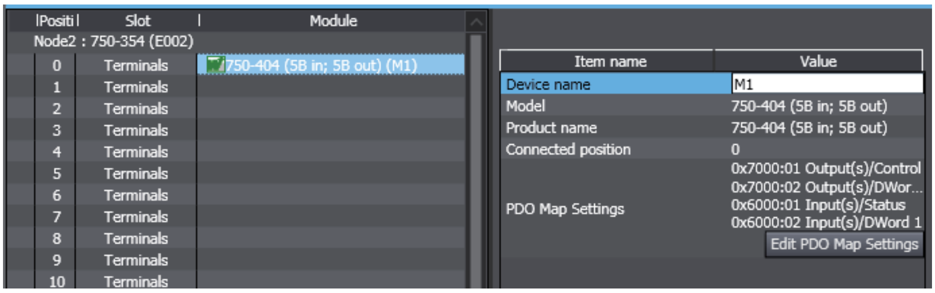


Positi	Slot	Module
Node2 : 750-354 (E002)		
0	Terminals	
1	Terminals	
2	Terminals	
3	Terminals	
4	Terminals	
5	Terminals	
6	Terminals	
7	Terminals	
8	Terminals	
9	Terminals	
10	Terminals	
11	Terminals	
12	Terminals	
13	Terminals	
14	Terminals	
15	Terminals	
16	Terminals	
17	Terminals	

Item name	Value
Device name	E002
Model	750-354
Product name	750-354 EtherCat fieldbus...
Revision	0x00000002
Number of modules	0
PDO Map Settings	0xF200:01 Output/K-Bus C... 0xF200:02 Output/Input Pr... 0xF200:03 Output/Output... 0xF200:04 Output/Output... 0xF200:05 Output/Diagnos... 0xF100:01 Input/K-Bus Cyc... 0xF100:02 Input/Input Pro... 0xF100:03 Input/Output Pr... 0xF100:04 Input/Output Pr... 0x10F3:04 Input/New Mess... 0xF100:05 Input/Diagnosti... Edit PDO Map Settings
Module config send method	Do not send

3 Select the location at which to register a module and double-click the module to register in the **Toolbox**. Or, drag the module from the **Toolbox** to the location to register it on the edit pane for the module configuration.

The module is registered and a list of module settings is displayed.



Positi	Slot	Module
Node2 : 750-354 (E002)		
0	Terminals	750-404 (5B in; 5B out) (M1)
1	Terminals	
2	Terminals	
3	Terminals	
4	Terminals	
5	Terminals	
6	Terminals	
7	Terminals	
8	Terminals	
9	Terminals	
10	Terminals	

Item name	Value
Device name	M1
Model	750-404 (5B in; 5B out)
Product name	750-404 (5B in; 5B out)
Connected position	0
PDO Map Settings	0x7000:01 Output(s)/Control 0x7000:02 Output(s)/DWor... 0x6000:01 Input(s)/Status 0x6000:02 Input(s)/DWord 1 Edit PDO Map Settings

A-4 Setting Transmission Delay Time by Actual Measurement

This section describes the procedure to set the transmission delay time that is measured and calculated in the actual network configuration to the CPU Unit.

Version Information

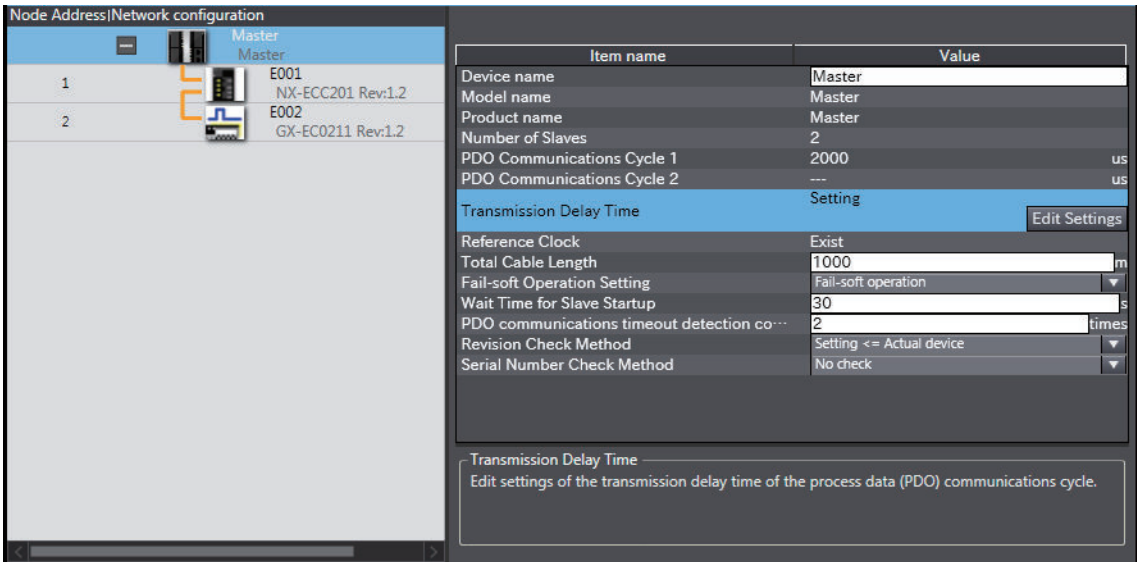
Project unit version 1.40 or later is required to set the transmission delay time by the actual measurement.

Precautions for Correct Use

Make sure to confirm the followings before you perform the actual measurement. Otherwise, the correct transmission delay time cannot be calculated.

- Connect all slaves, including disconnected and temporarily disabled slaves, to the network.
- If a ring topology is included, remove the cable connected to the end port of the ring.

1 Go online and click the **Edit Settings** Button for **Transmission Delay Time** in the master settings.

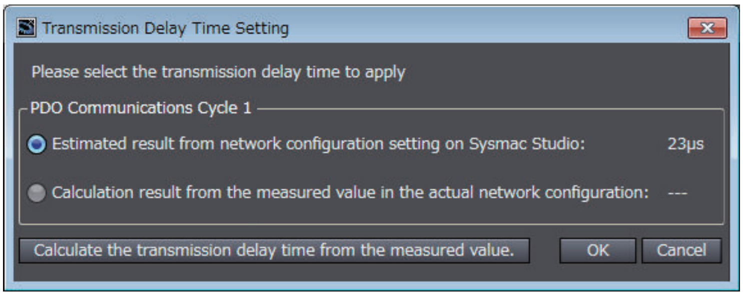


Item name	Value
Device name	Master
Model name	Master
Product name	Master
Number of Slaves	2
PDO Communications Cycle 1	2000 us
PDO Communications Cycle 2	--- us
Transmission Delay Time	Setting Edit Settings
Reference Clock	Exist
Total Cable Length	1000 m
Fail-soft Operation Setting	Fail-soft operation
Wait Time for Slave Startup	30 s
PDO communications timeout detection co···	2 times
Revision Check Method	Setting <= Actual device
Serial Number Check Method	No check

Transmission Delay Time

Edit settings of the transmission delay time of the process data (PDO) communications cycle.

The **Transmission Delay Time Setting** Dialog Box is displayed.



Transmission Delay Time Setting

Please select the transmission delay time to apply

PDO Communications Cycle 1

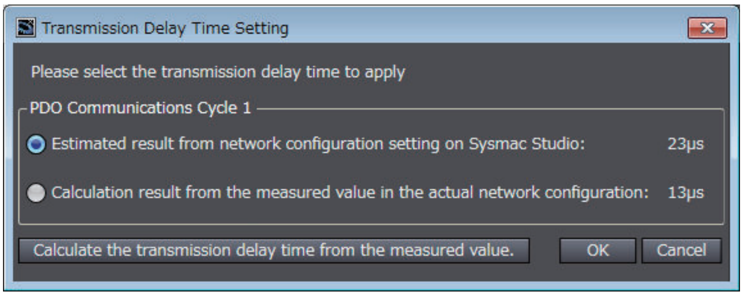
● Estimated result from network configuration setting on Sysmac Studio: 23μs

○ Calculation result from the measured value in the actual network configuration: ---

Calculate the transmission delay time from the measured value.
OK
Cancel

2 Click the **Calculate the transmission delay time from the measured value** Button.

The transmission delay time is calculated based on the measured value in the actual network configuration. After the actual measurement is completed, the calculated transmission delay time will be displayed on **Calculation result from the measured value in the actual network configuration**.



Transmission Delay Time Setting

Please select the transmission delay time to apply

PDO Communications Cycle 1

● Estimated result from network configuration setting on Sysmac Studio: 23µs

○ Calculation result from the measured value in the actual network configuration: 13µs

Calculate the transmission delay time from the measured value.
OK
Cancel

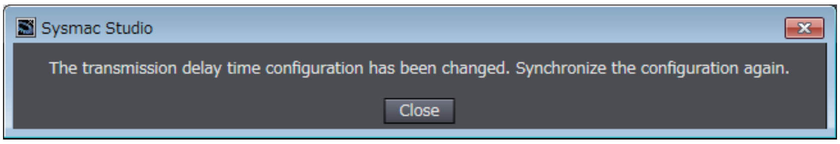
3 Select **Calculation result from the measured value in the actual network configuration** and click the **OK** Button.

The calculated transmission delay time is set to the transmission delay time in the network configuration information for the project.

Additional Information

The set transmission delay time is displayed on **Calculation result from the measured value in the actual network configuration** in the **Transmission Delay Time Setting** Dialog Box, even offline.

The following dialog box is displayed.



Sysmac Studio

The transmission delay time configuration has been changed. Synchronize the configuration again.

4 Click the **OK** Button and then perform synchronous transfer to send the project to the CPU Unit.

Precautions for Correct Use

If you changed the actual network configuration after the transmission delay time that was calculated by actual measurement was set to the project, perform the actual measurement again. When you transfer a project without actual measurement, the transmission delay time that was calculated by the previous actual measurement is set to the CPU Unit.

A-5 Items to Be Output to Log File of Diagnostic and Statistical Information, and Order of Outputs

The items to be output to the CSV-format log file of diagnostic and statistical information are as shown in the table below.
As for the order, items on the table are output from top to bottom.

(○: To be output, ×: Not to be output)

Item	Project unit version	
	Version earlier than 1.40	Version 1.40 or later
Date and time of record writing	○	○
Total number of frames sent	○	○
Total number of frames received	○	○
Number of frame reception timeouts	○	○
Number of frames discarded due to buffer overflows	○	×
Number of non-EtherCAT frames received	○	×
Link OFF count	○	○
Number of process data discarded when receiving	○	○
Number of messages discarded in mailbox reception	○	○
Number of lost repeat-send frames[*1]	○	×
Network propagation delay time	○	○
PDO communications cycle 1 current transmission cycle	○	○
PDO communications cycle 1 maximum transmission cycle	○	○
PDO communications cycle 1 minimum transmission cycle	○	○
PDO communications cycle 1 transmission jitter	○	○
PDO communications cycle 2 current transmission cycle	○	×
PDO communications cycle 2 maximum transmission cycle	○	×
PDO communications cycle 2 minimum transmission cycle	○	×
PDO communications cycle 2 transmission jitter	○	×
Number of lost repeat-send frames[*1]	×	○
Number of CRC error frames received	○	○
Number of frame reception errors	○	○
Collision count	○	○
Number of short frames received	○	○
Number of frames received with more than specified bytes	○	○
Number of slave error frames received	○	○
Network Connected Slave Table	○	○
Message Communications Enabled Slave Table	○	○
Process Data Communicating Slave Table	○	○

[*1] The order of outputs depends on the project unit versions.

A-6 Terminology

Use the following list of EtherCAT terms for reference.

Term	Abbreviation	Description
AL status	---	Status for indicating information on errors that occur in an application on a slave.
CAN in Automation	CiA	CiA is the international users' and manufacturers' group that develops and supports higher-layer protocols.
CAN application protocol over EtherCAT	CoE	A CAN application protocol service implemented on EtherCAT.
EtherCAT slave controller	ESC	A controller for EtherCAT slave communication.
EtherCAT state machine	ESM	An EtherCAT communication state machine.
EtherCAT slave information	ESI	An XML file that contains setting information for an EtherCAT slave.
EtherCAT Master Function Module	---	One of the function modules. This module controls the EtherCAT slaves as the EtherCAT master.
EtherCAT Technology Group	ETG	The ETG is a global organization in which OEM, End Users and Technology Providers join forces to support and promote the further technology development.
PDO communications	---	An acronym for process data communications.
SDO communications	---	One type of EtherCAT communications that uses service data objects (SDOs) for communicating information when required.
Slave Imformation Interface	SII	Slave information that is stored in non-volatile memory in the slave.
WireShark	---	Freeware for monitoring and capturing packets.
index	---	Address of an object within an application process.
object	---	An abstract representation of a particular component within a device, which consists of data, parameters, and methods.
object dictionary	OD	A data structure addressed by Index and Subindex that contains description of data type objects, communications objects and application objects.
Operational	---	A state in EtherCAT communications where SDO communications and I/O are possible.
service data object	SDO	CoE asynchronous mailbox communications where all objects in the object dictionary can be read and written.
subindex	---	Sub-address of an object within the object dictionary.
system-defined variable	---	A variable in the NJ/NX-series CPU Unit for providing information such as the EtherCAT communications status and error status. Status information about EtherCAT communications is obtained by reading system-defined variables from user applications in the NJ/NX-series CPU Unit.
receive PDO	RxPDO	A process data object received by an EtherCAT slave.
sync manager	SM	Collection of control elements to coordinate access to concurrently used objects.
slave disconnection	---	A state where an EtherCAT slave does not existed from the network due to the following causes. • Cable disconnection • Slave power interruption
Safe-Operational	---	A state in EtherCAT communications where only SDO communications and reading input data from slaves are possible. Outputs from slaves are not performed.

Term	Abbreviation	Description
project unit version	---	A unit version to be set for the project. It is set for the project in the Select Device Area of the Project Properties Dialog Box on the Sysmac Studio.
transmission jitter	---	The jitter in the process data transmission period (ns).
transmit PDO	TxPDO	A process data object sent from an EtherCAT slave.
distributed clocks	DC	Clock distribution mechanism used to synchronize EtherCAT slaves and the EtherCAT master.
device profile	---	Collection of device dependent information and functionality providing consistency between similar devices of the same device type.
device variables	---	Variables in the NJ/NX-series CPU Unit to which process data on EtherCAT slaves are allocated. Slave process data is accessed by directly reading and writing these device variables from user applications on the NJ/NX-series CPU Unit.
sync jitter	---	The offset in the operation timing between slaves.
network configuration information	---	The EtherCAT network configuration information held by the EtherCAT master.
packet monitoring	---	A function that enables the EtherCAT master to capture packets flowing over an EtherCAT line and store them in the master. You can start and stop capturing packets from the Sysmac Studio or with EtherCAT communications instructions.
Pre-Operational	---	A state in EtherCAT communications where only SDO communications are possible without being able to perform I/O.
process data	---	Collection of application objects designated to be transferred cyclically or acyclically for the purpose of measurement and control.
process data object	PDO	Structure described by mapping parameters containing one or several process data entities.
process data communications	---	One type of EtherCAT communications that uses process data objects (PDOs) to exchange information in realtime with a fixed cycle. This is also called *PDO communications*.
Originating slave of the ring	---	A slave that becomes the starting point of the ring topology. To be specific, any of the following slaves for which the cable redundancy is enabled. • GX-JC03 Junction Slave
Start port of the ring	---	A port that becomes the start point of the ring topology.
End port of the ring	---	A port that becomes the end point of the ring topology.

A-7 Version Information

A-7-1 Functions That Were Added or Changed for Each Unit Version

This section gives the functions that were added or changed for each unit version of the CPU Unit.

● Changes in and Additions to Functional Specifications

The following table gives the unit version of the CPU Units and the Sysmac Studio version for each addition or change to the functional specifications.

Function	Addition/ change	Reference	Unit version	Sysmac Studio version
Backup and restore operations	Addition	page 9-28 and page 9-30	1.03	1.04
Enabling/disabling	Addition	page 5-20 and page 5-25	1.04	1.05
Packet monitoring	Addition[*1]	page 6-25 and page A-7	1.10	1.12
	Change		1.40[*2]	1.29
Diagnosis/statistics log	Addition	page 9-16	1.11	---
Cable redundancy function	Addition	page 5-5	1.40[*3]	1.29

[*1] This function was added to the NJ301-□□□□ CPU Units. The NJ501-□□□□ and NJ101-□□□□ CPU Units support packet monitoring with all versions.

[*2] Packet monitoring can be used with project unit version earlier than 1.40. It cannot be used with project unit version 1.40 or later.

[*3] The cable redundancy function can be used with project unit version 1.40 or later. It cannot be used with project unit version earlier than 1.40.

● Adding the EtherCAT Slaves That You Can Use

If you use a newer unit version of the CPU Unit, you can increase the OMRON EtherCAT Slaves that you can use. Refer to *1-2-1 System Configuration* on page 1-4 for details.
For information on the most recent lineup of the OMRON EtherCAT slaves, NX-series EtherCAT Communications Coupler Units and NX Units, refer to catalogs or OMRON websites, or ask your OMRON representative.

A-7-2 Differences in Functions and Operations Depending on the Project Unit Version

The following table shows the differences in functions and operations for the EtherCAT master depending on the project unit version.

Item	Description	Project unit version		
		Version earlier than 1.40	**Version 1.40 and 1.41**	**Version 1.42 or later**
Function	Packet monitoring	Supported.	Not supported.	
	Cable redundancy function	Not supported.	Supported. However, synced slaves cannot be connected in a ring topology and in a drop line from the ring.	Supported. Synced slaves can be connected in a ring topology and in a drop line from the ring.
	Actual measurement of network transmission delay time in actual configuration	Not supported.	Supported.	
	DC Synchronous Correction	Selectable between Enable slave monitoring option and Disable slave monitoring option	Fixed to Enable slave monitoring option	
	Node address setting for slaves from other manufacturers	Can be set with the Sysmac Studio only.	Can be set with the hardware switches or Sysmac Studio.	
	PDO communications timeout detection count that can be set	1 to 8 errors	1 to 50 errors	
	Set value for total cable length[*1]	1 to (100 × Maximum number of slaves) [m]	1 to 51,200 [m]	
Operation	State in which the EtherCAT Junction Slave ports actually connected to slaves do not agree with the Sysmac Studio network configuration	This status is not detected as an error.	One of the following errors occurs. • *Network Configuration Verification Error (Unnecessary Slave Connected)* (84320003 hex) event • *Network Configuration Verification Error (Mismatched Slave)* (84330004 hex) event • *Network Configuration Verification Error (Slave Unconnected)* (84380000 hex) event	
	Operation of the slaves that are disconnected from the Sysmac Studio or with a special instruction	• Control state: Init state • PDO communications for input data: Stop • PDO communications for output data: Stop • Message communications: Not possible	• Control state: Pre-Operational state • PDO communications for input data: Stop • PDO communications for output data: Stop • Message communications: Possible	
	Detection timing of the *EtherCAT Frame Not Received* (842E0000 hex) event	• At power ON • At Controller reset • When a cable is connected to EtherCAT master	• At power ON • At Controller reset • When a cable is connected to EtherCAT master • During communications	

Item	Description	Project unit version		
		Version earlier than 1.40	Version 1.40 and 1.41	Version 1.42 or later
	Communication state of master and slave if a minor fault occurs when **Fail-soft Operation** is set to **Stop**	• State for the master and all of the slaves: Pre-Operational state • PDO communications for input data: Stop • PDO communications for output data: Stop • Message communications: Possible	• State for the master and all of the slaves: Safe-Operational state • PDO communications for input data: Continue • PDO communications for output data: Stop • Message communications: Possible	
	Operation when more than the maximum number of slaves are connected during communications	• A *Network Configuration Error* (84210000 hex) event occurs. • Communications continue with salves that are during communications.	• A *Network Configuration Verification Error (Unnecessary Slave Connected)* (84320003 hex) event occurs. • Communications continue with salves that are during communications. However, an error may occur in synced slaves.	
	Event that will occur	Refer to *Correspondence of Events between Project Unit Version Earlier than 1.40 and Project Unit Version 1.40 or Later* in the appendices of the *NJ/NX-series Troubleshooting Manual (Cat. No. W503)* for specific events.		

*1. Only when the total of cable lengths exceeds the default (1,000 m), set an appropriate value to the total cable length.

Index

Index

OMRON Corporation **Industrial Automation Company**

Kyoto, JAPAN **Contact : www.ia.omron.com**

Regional Headquarters

OMRON EUROPE B.V.
Wegalaan 67-69, 2132 JD Hoofddorp
The Netherlands
Tel: (31) 2356-81-300 Fax: (31) 2356-81-388

OMRON ASIA PACIFIC PTE. LTD.
438B Alexandra Road, #08-01/02 Alexandra
Technopark, Singapore 119968
Tel: (65) 6835-3011 Fax: (65) 6835-2711

OMRON ELECTRONICS LLC
2895 Greenspoint Parkway, Suite 200
Hoffman Estates, IL 60169 U.S.A.
Tel: (1) 847-843-7900 Fax: (1) 847-843-7787

OMRON (CHINA) CO., LTD.
Room 2211, Bank of China Tower,
200 Yin Cheng Zhong Road,
PuDong New Area, Shanghai, 200120, China
Tel: (86) 21-5037-2222 Fax: (86) 21-5037-2200

Authorized Distributor: