OMRON

Industrial PC Platform

NY-series

IPC Machine Controller
Industrial Panel PC / Industrial Box PC

Built-in EtherCAT® Port
User's Manual

NY532-1500
NY532-1400
NY532-1300
NY512-1500
NY512-1400
NY512-1300

Industrial Panel PC
Industrial Box PC

sysmac
always in control

W562-E1-01

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, and Visual Basic 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.
- Intel and Intel Core are trademarks of Intel Corporation in the U.S. and / or other countries.

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

Copyrights

Microsoft product screen shots reprinted with permission from Microsoft Corporation.

Introduction

Thank you for purchasing an NY-series IPC Machine Controller Industrial Panel PC / Industrial Box PC.

This manual provides a collective term of Industrial Panel PC and Industrial Box PC which are applicable products as the NY-series Industrial PC. This manual also provides the range of devices that are directly controlled by the Controller functions embedded the Real-Time OS in the NY-series Industrial PC as the Controller.

This manual contains information that is necessary to use the NY-series Controller. Please read this manual and make sure you understand the functionality and performance of the NY-series Controller 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.

- NY-series IPC Machine Controller Industrial Panel PC
    - NY532-15□□
    - NY532-14□□
    - NY532-13□□

- NY-series IPC Machine Controller Industrial Box PC
    - NY512-15□□
    - NY512-14□□
    - NY512-13□□

Part of the specifications and restrictions for the Industrial PC are given in other manuals. Refer to *Relevant Manuals* on page 2 and *Related Manuals* on page 18.

Relevant Manuals

The following table provides the relevant manuals for the NY-series Controller.
Read all of the manuals that are relevant to your system configuration and application before you use the NY-series Controller.
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					NY-series IPC Machine Controller Industrial Panel PC / Industrial Box PC Motion Control User's Manual	NY-series Motion Control Instructions Reference Manual	NY-series IPC Machine Controller Industrial Panel PC / Industrial Box PC Built-in EtherCAT Port User's Manual	NY-series IPC Machine Controller Industrial Panel PC / Industrial Box PC Built-in EtherNet/IP Port User's Manual	NY-series Troubleshooting Manual
	NY-series IPC Machine Controller Industrial Panel PC Hardware User's Manual	NY-series IPC Machine Controller Industrial Box PC Hardware User's Manual	NY-series IPC Machine Controller Industrial Panel PC / Industrial Box PC Setup User's Manual	NY-series IPC Machine Controller Industrial Panel PC / Industrial Box PC Software User's Manual	NY-series Instructions Reference Manual					
Introduction to NY-series Panel PCs	O									
Introduction to NY-series Box PCs		O								
Setting devices and hardware										
Using motion control	O	O				O				
Using EtherCAT	O	O						O		
Using EtherNet/IP	O	O							O	
Making setup*1										
Making initial settings			O							
Preparing to use Controllers			O							
Software settings										
Using motion control				O		O				
Using EtherCAT				O				O		
Using EtherNet/IP				O					O	
Writing the user program										
Using motion control				O	O	O	O			
Using EtherCAT				O	O			O		
Using EtherNet/IP				O	O				O	
Programming error processing				O	O					O
Testing operation and debugging										
Using motion control				O		O				
Using EtherCAT				O				O		
Using EtherNet/IP				O					O	
Learning about error management and corrections*2										O
Maintenance										
Using motion control	O	O				O				
Using EtherCAT	O	O						O		
Using EtherNet/IP	O	O							O	

*1 Refer to the *NY-series Industrial Panel PC / Industrial Box PC Setup User's Manual* (Cat. No. W568) for how to set up and how to use the utilities on Windows.

*2 Refer to the *NY-series Troubleshooting Manual* (Cat. No. W564) for the error management concepts and the error items.

Manual Structure

Some of the descriptions of functions in this manual are common to NJ/NX-series. Therefore, note the following conditions.

- The same function names are used for the common functions of the NJ/NX/NY-series. If the term "CPU Unit" is included in the function names, such as the CPU Unit names, CPU Unit write protection and other functions, it indicates the "Controller" in the NY-series.
- The "CPU Unit" that is described in a list of function specifications in this manual also indicates the "Controller" in the NY-series.

Page Structure

The following page structure is used in this manual.
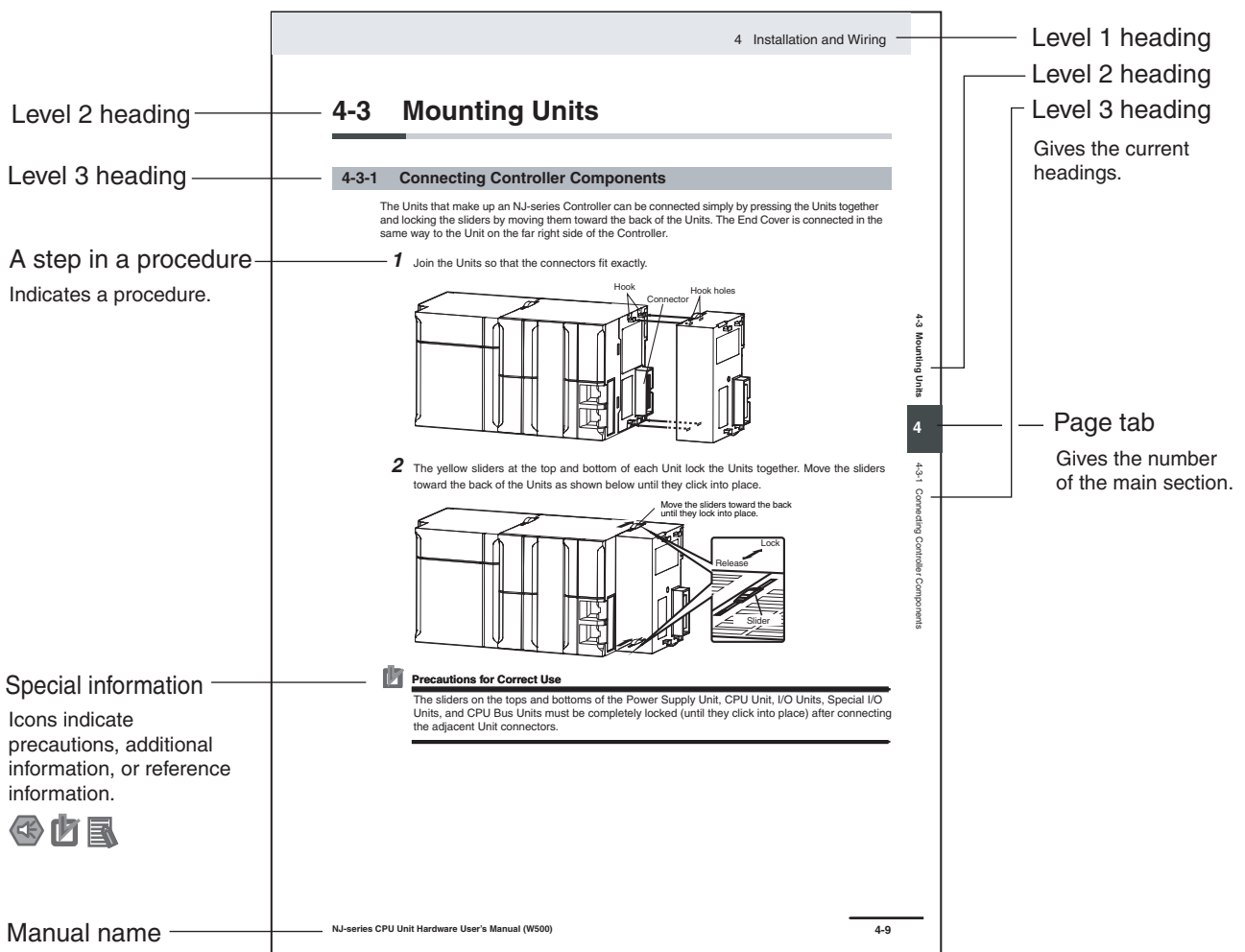
This illustration is provided only as a sample. It may not literally appear in this manual.

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.

Note References are provided to more detailed or related information.

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

1	Introduction	**A**	Appendices	
2	Part Names and Slave Settings	**I**	Index	
3	EtherCAT Communications			
4	EtherCAT Network Wiring			
5	Setting Up EtherCAT Communications with the Sysmac Studio			
6	Process Data Communications and SDO Communications			
7	System-defined Variables That Are Related to the Built-in EtherCAT Port			
8	Example of Operations for EtherCAT Communications			
9	Replacing Slaves during Communications			
10	Diagnostic and Statistical Information			

1 A
2 I
3
4
5
6
7
8
9
10

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 That Are Related to the Built-in EtherCAT Port

Section 8 Example of Operations for EtherCAT Communications

Section 9 Replacing Slaves during Communications

Section 10   Diagnostic and Statistical Information

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 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.

Safety Precautions

Refer to the following manuals for safety precautions.

- NY-series Industrial Panel PC Hardware User's Manual (Cat. No. W557)
- NY-series Industrial Box PC Hardware User's Manual (Cat. No. W556)
- NY-series Industrial Panel PC / Industrial Box PC Software User's Manual (Cat. No. W558)

Precautions for Safe Use

Refer to the following manuals for precautions for safe use.

- NY-series Industrial Panel PC Hardware User's Manual (Cat. No. W557)
- NY-series Industrial Box PC Hardware User's Manual (Cat. No. W556)
- NY-series Industrial Panel PC / Industrial Box PC Software User's Manual (Cat. No. W558)

Precautions for Correct Use

Refer to the following manuals for precautions for correct use.

- NY-series Industrial Panel PC Hardware User's Manual (Cat. No. W557)
- NY-series Industrial Box PC Hardware User's Manual (Cat. No. W556)
- NY-series Industrial Panel PC / Industrial Box PC Software User's Manual (Cat. No. W558)

Regulations and Standards

Conformance to EU Directives

Applicable Directives

- EMC Directives

Concepts

● **EMC Directive**

OMRON devices that comply with EU Directives also conform to the related EMC standards so that they can be more easily built into other devices or the overall machine. The actual products have been checked for conformity to EMC standards.*
Whether the products conform to the standards in the system used by the customer, however, must be checked by the customer. EMC-related performance of the OMRON devices that comply with EU Directives will vary depending on the configuration, wiring, and other conditions of the equipment or control panel on which the OMRON devices are installed. The customer must, therefore, perform the final check to confirm that devices and the overall machine conform to EMC standards.

* Applicable EMC (Electromagnetic Compatibility) standards are as follows:
EMS (Electromagnetic Susceptibility): EN 61131-2
EMI (Electromagnetic Interference): EN 61131-2 (Radiated emission: 10-m regulations)

● **Conformance to EU Directives**

The NY-series Controllers comply with EU Directives. To ensure that the machine or device in which the NY-series Controller is used complies with EU Directives, the Controller must be installed as follows:

- The NY-series Controller must be installed within a control panel.
- You must use the power supply in SELV specifications for the DC power supplies connected to DC Power Supply Units and I/O Units.
- NY-series Controllers that comply with EU Directives also conform to the Common Emission Standard (EN 61000-6-4). Radiated emission characteristics (10-m regulations) may vary depending on the configuration of the control panel used, other devices connected to the control panel, wiring, and other conditions.
  You must therefore confirm that the overall machine or equipment complies with EU Directives.

Software Licenses and Copyrights

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/.

Versions

Hardware revisions and unit versions are used to manage the hardware and software in NY-series Controllers 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.

Checking Versions

You can check versions on the ID information indications or with the Sysmac Studio.

Checking Unit Versions on ID Information Indications

The unit version is given on the ID information indication on the back side of the product.

The ID information on an NY-series NY5□2-1□□□ Controller is shown below.

ID information indication

Unit version

Ver.1.□□
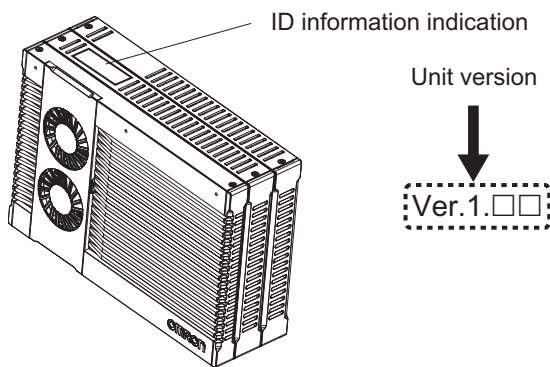
Checking Unit Versions with the Sysmac Studio

You can use the Sysmac Studio to check unit versions. The procedure is different for Units and for EtherCAT slaves.

● **Checking the Unit Version of an NY-series Controller**

You can use the Production Information while the Sysmac Studio is online to check the unit version of a Unit. You can only do this for the Controller.

1 Right-click **CPU Rack** under **Configurations and Setup** - **CPU/Expansion Racks** in the Multiview Explorer and select **Production Information**.

The Production Information Dialog Box is displayed.

● **Changing Information Displayed in Production Information Dialog Box**

1 Click the **Show Detail** or **Show Outline** Button at the lower right of the Production Information Dialog Box.

The view will change between the production information details and outline.
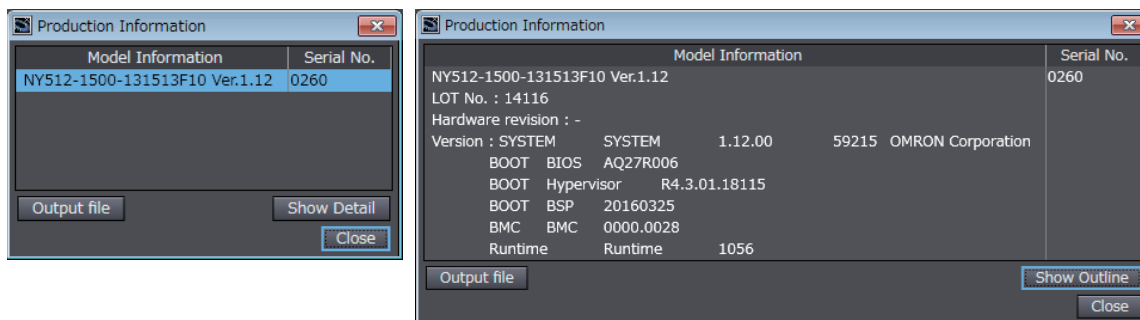
Outline View   Detail View

The information that is displayed is different for the Outline View and Detail View. The Detail View displays the unit version, hardware revision, and other versions. The Outline View displays only the unit version.

● **Checking the Unit Version of an EtherCAT Slave**

You can use the Production Information while the Sysmac Studio is online to check the unit version of an EtherCAT slave. Use the following procedure to check the unit version.

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

The EtherCAT Tab Page is displayed.

2 Right-click the master on the EtherCAT Tab Page and select **Display Production Information**.

The Production Information Dialog Box is displayed.
The unit version is displayed after "Rev."

● **Changing Information Displayed in Production Information Dialog Box**

1 Click the **Show Detail** or **Show Outline** Button at the lower right of the Production Information Dialog Box.

The view will change between the production information details and outline.
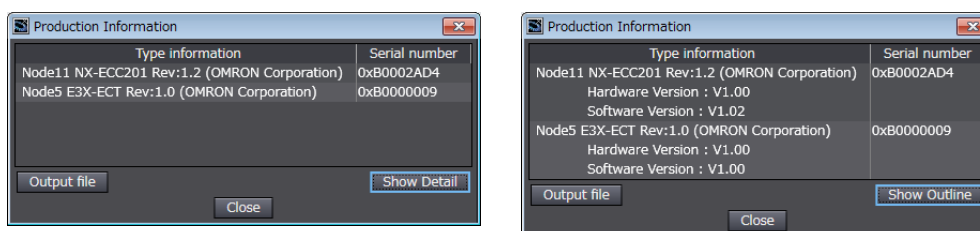
Outline View   Detail View

Related Manuals

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

Manual name	Cat. No.	Model numbers	Application	Description
NY-series IPC Machine Controller Industrial Panel PC Hardware User's Manual	W557	NY532-1□□□	Learning the basic specifications of the NY-series Industrial Panel PCs, including introductory information, designing, installation, and maintenance. Mainly hardware information is provided.	An introduction to the entire NY-series system is provided along with the following information on the Industrial Panel PC. • Features and system configuration • Introduction • Part names and functions • General specifications • Installation and wiring • Maintenance and inspection
NY-series IPC Machine Controller Industrial Panel PC Hardware User's Manual	W556	NY512-1□□□	Learning the basic specifications of the NY-series Industrial Box PCs, including introductory information, designing, installation, and maintenance. Mainly hardware information is provided.	An introduction to the entire NY-series system is provided along with the following information on the Industrial Box PC. • Features and system configuration • Introduction • Part names and functions • General specifications • Installation and wiring • Maintenance and inspection
NY-series IPC Machine Controller Industrial Panel PC / Industrial Box PC Setup User's Manual	W568	NY532-1□□□ NY512-1□□□	Learning the initial settings of the NY-series Industrial PCs and preparations to use Controllers.	The following information is provided on an introduction to the entire NY-series system. • Two OS systems • Initial settings • Industrial PC Support Utility • NYCompolet • Industrial PC API • Backup and recovery
NY-series IPC Machine Controller Industrial Panel PC / Industrial Box PC Software User's Manual	W558	NY532-1□□□ NY512-1□□□	Learning how to program and set up the Controller functions of an NY-series Industrial PC.	The following information is provided on the NY-series Controller functions. • Controller operation • Controller features • Controller settings • Programming based on IEC 61131-3 language specifications
NY-series Instructions Reference Manual	W560	NY532-1□□□ NY512-1□□□	Learning detailed specifications on the basic instructions of an NY-series Industrial PC.	The instructions in the instruction set (IEC 61131-3 specifications) are described.
NY-series IPC Machine Controller Industrial Panel PC / Industrial Box PC Motion Control User's Manual	W559	NY532-1□□□ NY512-1□□□	Learning about motion control settings and programming concepts of an NY-series Industrial PC.	The settings and operation of the Controller and programming concepts for motion control are described.
NY-series Motion Control Instructions Reference Manual	W561	NY532-1□□□ NY512-1□□□	Learning about the specifications of the motion control instructions of an NY-series Industrial PC.	The motion control instructions are described.
NY-series IPC Machine Controller Industrial Panel PC / Industrial Box PC Built-in EtherCAT® Port User's Manual	W562	NY532-1□□□ NY512-1□□□	Using the built-in EtherCAT port in an NY-series Industrial PC.	Information on the built-in EtherCAT port is provided. This manual provides an introduction and provides information on the configuration, features, and setup.

Manual name	Cat. No.	Model numbers	Application	Description
NY-series IPC Machine Controller Industrial Panel PC / Industrial Box PC Built-in EtherNet/IP™ Port User's Manual	W563	NY532-1□□□ NY512-1□□□	Using the built-in EtherNet/IP port in an NY-series Industrial PC.	Information on the built-in EtherNet/IP port is provided. Information is provided on the basic setup, tag data links, and other features.
NY-series Troubleshooting Manual	W564	NY532-1□□□ NY512-1□□□	Learning about the errors that may be detected in an NY-series Industrial PC.	Concepts on managing errors that may be detected in an NY-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.
NX-series EtherCAT® Coupler Unit User's Manual	W519	NX-ECC□□□	Learning how to use an 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.

Revision History

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

Revision code	Date	Revised content
01	September 2016	Original production

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.

- Output data addressed to the local node is read.
- Input data is written.

Ethernet frames
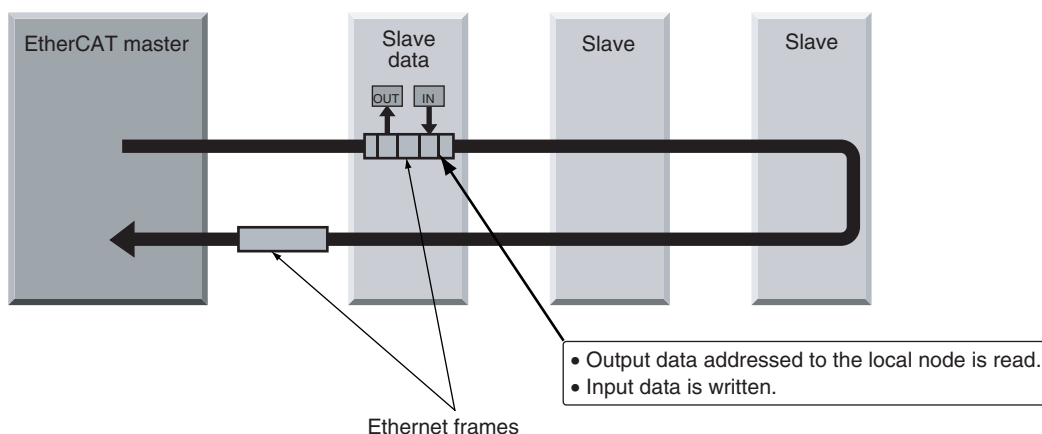
1-1-3   EtherCAT Features for NY-series Controllers

EtherCAT in the NY-series Controllers has the following features.

Synchronization of the Controller Processing Period and the EtherCAT Communications Cycle

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

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

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 Ether-CAT 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 Industrial PC Platform NY-series IPC Machine Controller Industrial Panel PC / Industrial Box PC together with Sysmac devices(*) and the Sysmac Studio Automation Software to achieve optimum functionality and ease of operation.

* "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.
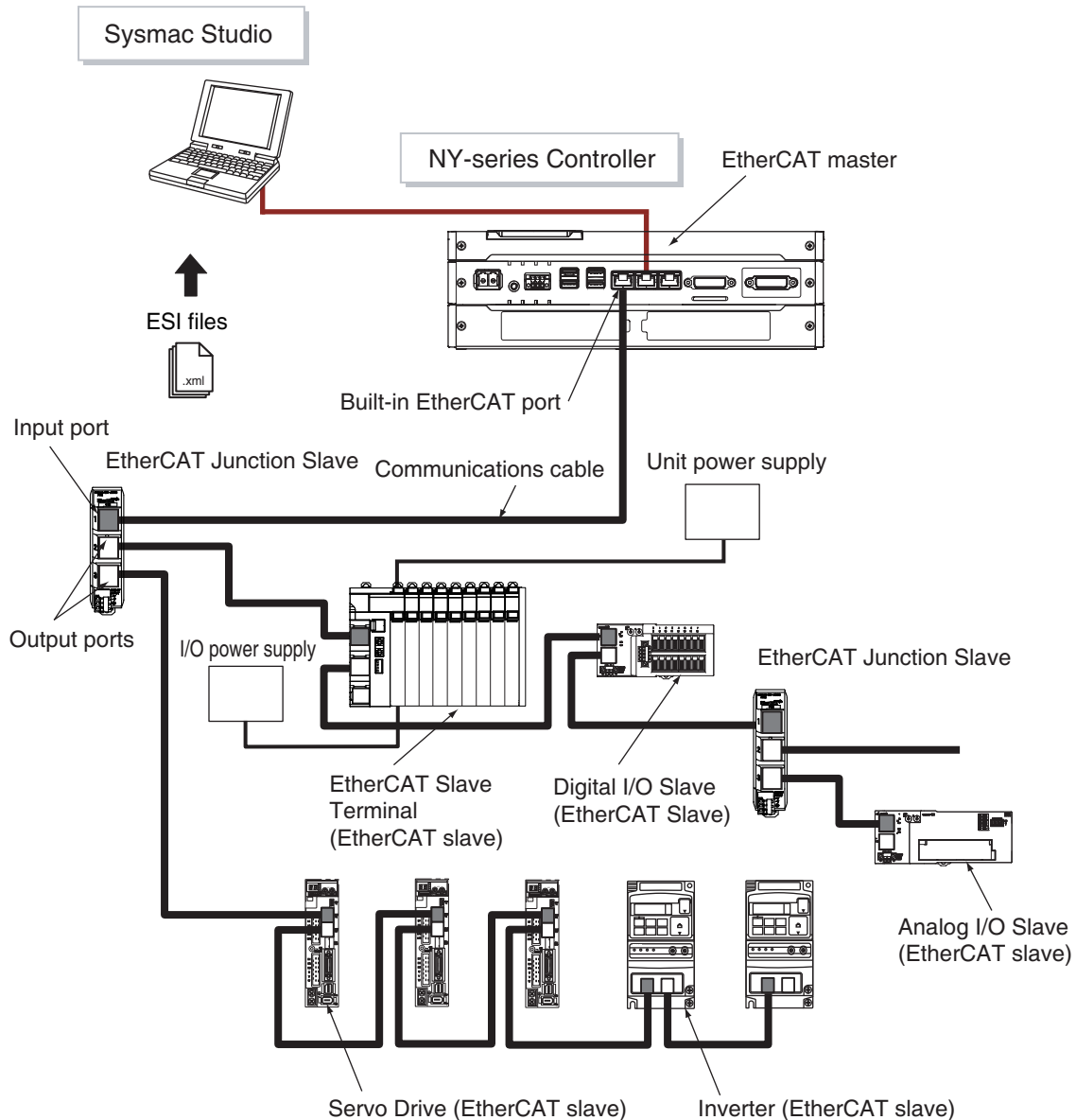
Outlines of the configuration devices are given below.

● **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.
- The EtherCAT slaves can be classified into two: whether they support synchronization based on a distributed clock (DC) or not.

  a) Non-synced Slaves
     These slaves perform sequence control that does not require synchronization between them. Digital I/O Slaves and Analog I/O Slaves are among this type. They do not support a distributed clock (DC).

  b) Synced Slaves
     These slaves are synchronized based on the distributed clock (DC). Servo Drives and encoder input slaves are among this type.

Note 1 Refer to *1-3-1 Performance Specifications* for details on the maximum number of slaves and the settable node address range.

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

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

Name	Type	Model	Synced/ Non-synced Slaves	Assigning an axis
Digital I/O Slaves	Slaves with screw terminals and 2-tier terminal block	GX-□D16□1/OC1601	Non-synced Slaves	Not possible
	Slaves with screw terminals and 3-tier terminal block	GX-ID16□2/OD16□2/MD16□2	Non-synced Slaves	Not possible
	Slaves with e-CON connectors	GX-□D16□8/□D32□8	Non-synced Slaves	Not possible
Analog I/O Slaves	Slaves with screw terminals and 2-tier terminal block	GX-AD0471/DA0271	Non-synced Slaves	Not possible
NX-series EtherCAT Coupler Unit	NX Series	NX-ECC201 NX-ECC202 NX-ECC203	Synced Slaves	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	Not possible
Advanced General-purpose Inverters	RX Series	3G3RX-V1 with EtherCAT Communications Unit 3G3AX-RX-ECT	Non-synced Slaves	Not possible

Name	Type	Model	Synced/ Non-synced Slaves	Assigning an axis
AC Servo Drive	G5-series Servo Drive with EtherCAT communications	R88M-K/R88D-KN□-ECT	Synced Slaves	Possible
Linear Servo Drives	G5-series Linear Servo Drives with EtherCAT communications	R88L-EC/R88D-KN□-ECT-L	Synced Slaves	Possible
Encoder Input Slaves	Slave with 3-tier terminal block	GX-EC0211/EC0241	Synced Slaves	Possible
Vision Sensors	FH Series	FH-1□□□ FH-3□□□	Synced Slaves	Not possible
Specialized Vision Sensors for Positioning	FQ-M-series Sensors with EtherCAT Communications	FQ-MS12□(-□)-ECT	Synced Slaves	Not possible
	FZM1-series Vision Sensors	FZM1-35□-ECT	Synced Slaves	Not possible
Digital Sensor Communications Unit	E3NW Series	E3NW-ECT	Synced Slaves	Not possible
Fiber Sensor Communications Unit	E3X Series	E3X-ECT	Synced Slaves	Not possible
Confocal Fiber Type Displacement Sensors	ZW Series	ZW-CE1□	Synced Slaves	Not possible

EtherCAT Junction Slave

This is a special Unit for branching EtherCAT network wiring. 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 use one of these, a major fault level error or other error may occur.

Precautions for Correct Use

- 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 will occur.
- A GX-JC03 EtherCAT Junction Slave needs only one node address. A GX-JC06 EtherCAT Junction Slave needs two node addresses.

> **Additional Information**
>
> A 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 In port of Sub-device appears as "Internal Port" in the Sysmac Studio and it is internally connected to the Main device.
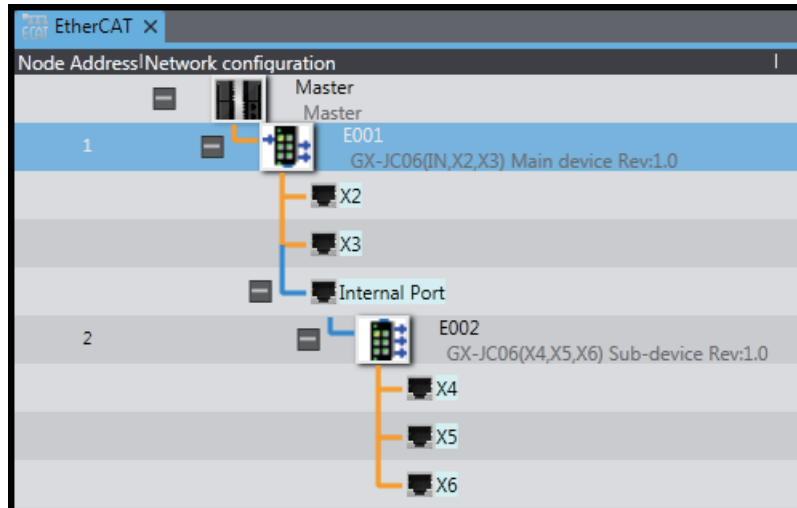
● Sysmac Studio

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 Supply

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**
>
> 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.
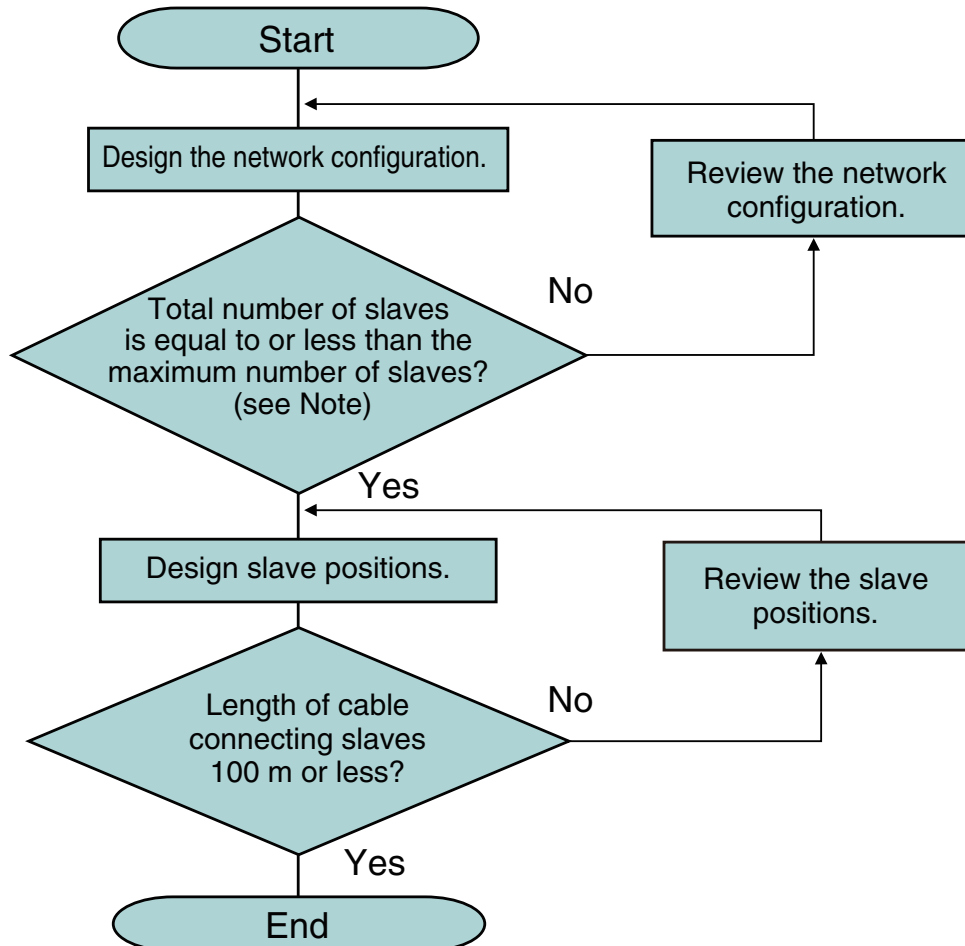
Note   Refer to *1-3-1 Performance Specifications* for details on the maximum number of slaves.

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

1-3-1 Performance Specifications

Item	Specification
	NY5□2-1□□□
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[3]	Line, daisy chain, and branching
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	192
Settable node address range	1 to 512
Maximum process data size	Input: 5,736 bytes Output: 5,736 bytes However, the data must not exceed 4 frames (the maximum number of process data frames).
Maximum size per slave	Input: 1,434 bytes Output: 1,434 bytes
Maximum message size	2,048 bytes
Communications cycle	500 to 8,000 μs (in 250-μs increments)
Sync jitter	1 μs max.

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

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

*3   Wiring in a ring configuration is not possible.

List of Supported Functions

Function name	Description
Process data communica-tions	PDO mapping using CoE
	Fail-soft operation for slave communications errors
	Stop operation for slave communications errors
DC (distributed clock)	Clock synchronization
SDO communications	CoE • Emergency message server (receptions from slaves) • SDO requests and responses • Communications between slaves
Configuration	Automatic setting of device variable names, etc.
	Setting node address using hardware switches on the slaves
	Setting node address using the Sysmac Studio
	Network scan (collection of information on devices connected to the network)
	Display of network configuration information (display of supported topology)
RAS functions	Slave configuration check when starting network
	Reading of error information (emergency error history, subscription information, etc.)
	Packet monitoring
	Troubleshooting information
	Diagnosis/statistics log
Operations during errors	When error occurs (stop communications or change to fail-soft operation)
	Restoring communications when errors are cleared (Moves to operational state.)
Slave information	Enabling/disabling slaves
	Disconnecting/reconnecting slaves

1-4 EtherCAT Communications Procedure

1-4-1 Overview

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

1-4-2 Details

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.) See the 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*
↓				
2. Laying EtherCAT Communications Cables		Connect the EtherCAT slaves to the EtherCAT port of the NY-series Controller. If there is more than one EtherCAT slave, connect them using a daisy chain or branch wiring.		*Section 4 EtherCAT Network Wiring*
↓				

3. Creating the EtherCAT Network Configuration		• Use the Sysmac Studio to create a new project.
		  (a) Create EtherCAT network configuration offline.
		  or
		  (b) Go online and create the EtherCAT network configuration from the actual network devices. Do so after making the online connection that is described in step 7.
			Create EtherCAT network configuration with **EtherCAT** under **Configurations and Setup**.	*5-2 Creating the EtherCAT Network Configuration*

↓				
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*
	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*
↓				

Step		Description	Sysmac Studio operation	Section
5	Setting EtherCAT Parameters			
	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.	Create an EtherCAT master with **EtherCAT** under **Configurations and Setup**.	*5-4-1 Setting EtherCAT Master*
	Setting EtherCAT Slave Parameters	Set the EtherCAT slave parameters. Example: Enable/disable slaves.	Create EtherCAT slaves with **EtherCAT** under **Configurations and Setup**.	*5-4-2 Setting EtherCAT Slaves*

↓

Step		Description	Sysmac Studio operation	Section
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)*
		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*

↓

Step	Description	Sysmac Studio operation	Section
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 NY-series Controller. • Use the Sysmac Studio to set communications with the NY-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* *3-3 State Transitions for EtherCAT Communications*

↓

Step	Description	Sysmac Studio operation	Section
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 Actual network configuration.	*5-5 Comparing and Merging EtherCAT Network Configurations*

↓

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.	*5-6 Downloading the Network Configuration Information*

↓

Step	Description	Sysmac Studio operation	Section
10. Checking the EtherCAT Process Data Communications Status	• Check the ACT indicator on the NY-series Controller. A flashing yellow ACT indicator shows that data is being transmitted and received after the link is established. • In the Sysmac Studio, check that process data communications with all EtherCAT slaves are performed normally.	Display the Controller Status Pane and switch to the detailed view.	*5-7 Confirming Communications after Completing EtherCAT Configuration and Settings*

2

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 NY-series Controller.

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.

Ethernet port
Built-in EtherNet/IP port
Built-in EtherCAT port
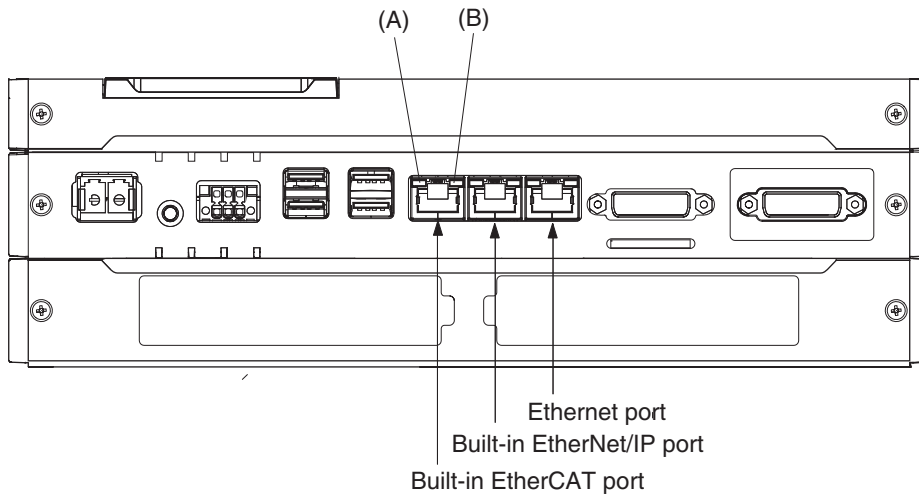
The operating status corresponding to colors and status of the built-in EtherCAT port indicators are shown below.

Item	Indicator	Color	Status	Operating status
(A)	LINK/ ACT	■Yellow	Lit	The link is established.
			Flashing	Data communications are in progress after establishing link. The indicator flashes whenever data is sent or received.
			Not lit	The link is not established.
(B)	10/100/ 1000	■Orange	Lit	The link is established at 1,000 M/sec.
		■Green	Lit	The link is established at 100 M/sec.
			Not lit	The link is established at 10 M/sec, or the link is not established.

Refer to *3-3-2 Control States for EtherCAT Communications* 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.
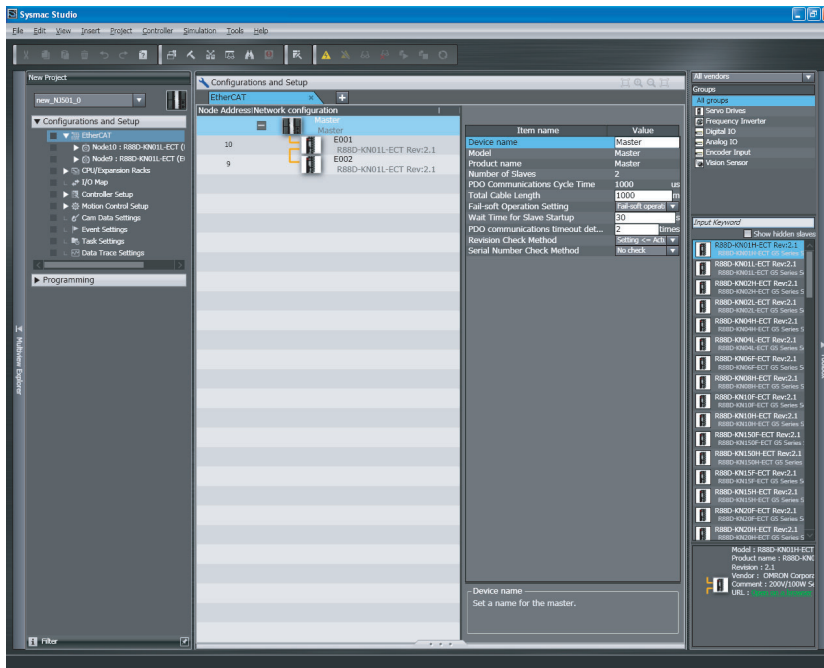
● **Setting EtherCAT Slave Variables and Axes**

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.
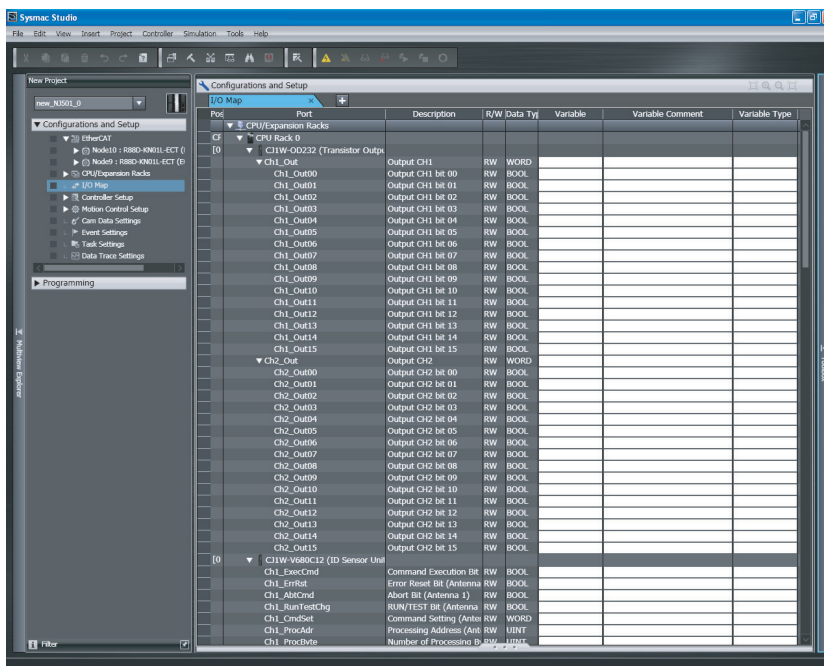
Axis Basic Settings: Used to create Axis Variables and set parameters for Servo Drive and encoder input slaves.

Use the Axis Basic Settings to assign Axis Variables to the Servo Drive/encoder input slaves.
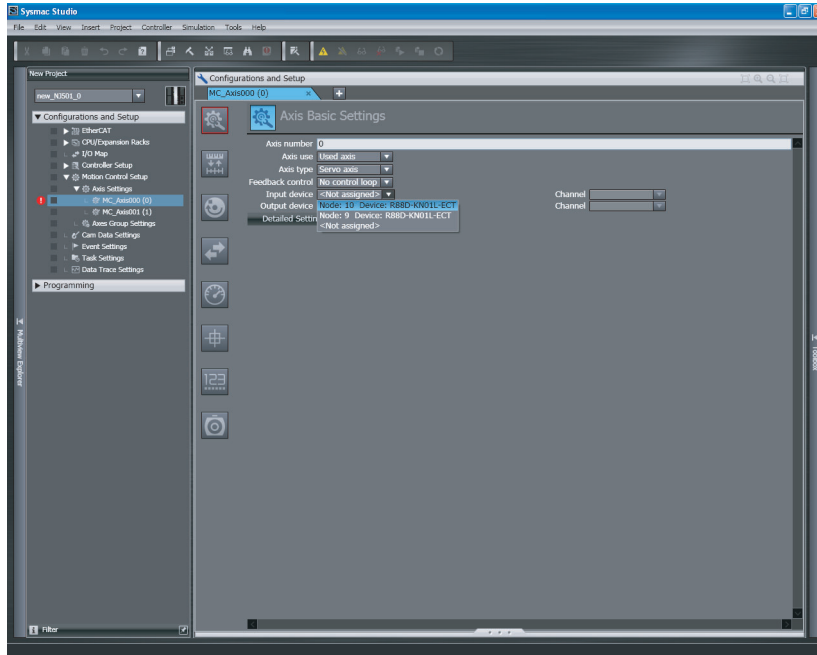
Setting Axis Parameters:

Use the Axis Setting Table to set axis parameters.
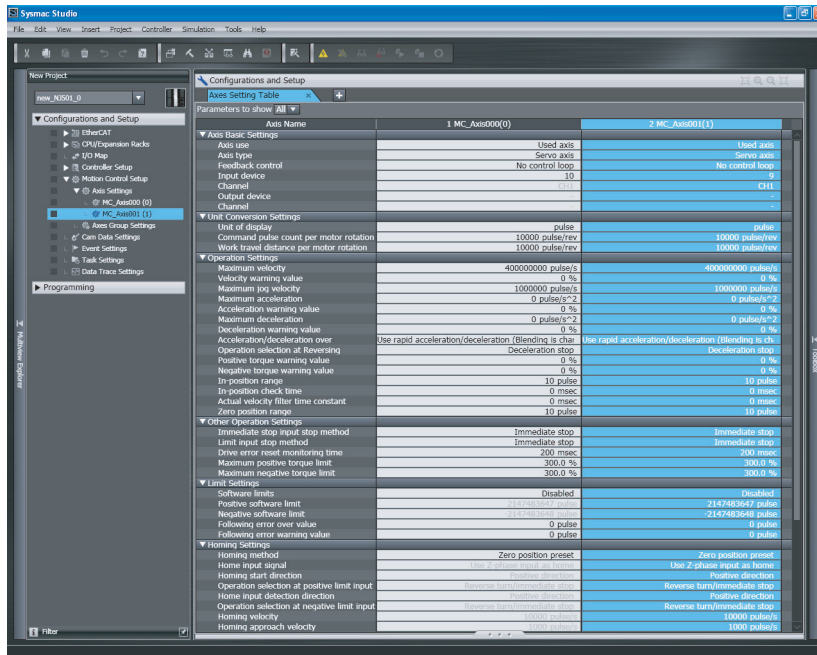
● **EtherCAT Master and Slave Parameters Settings**

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

• Parameter Setting for the EtherCAT Master
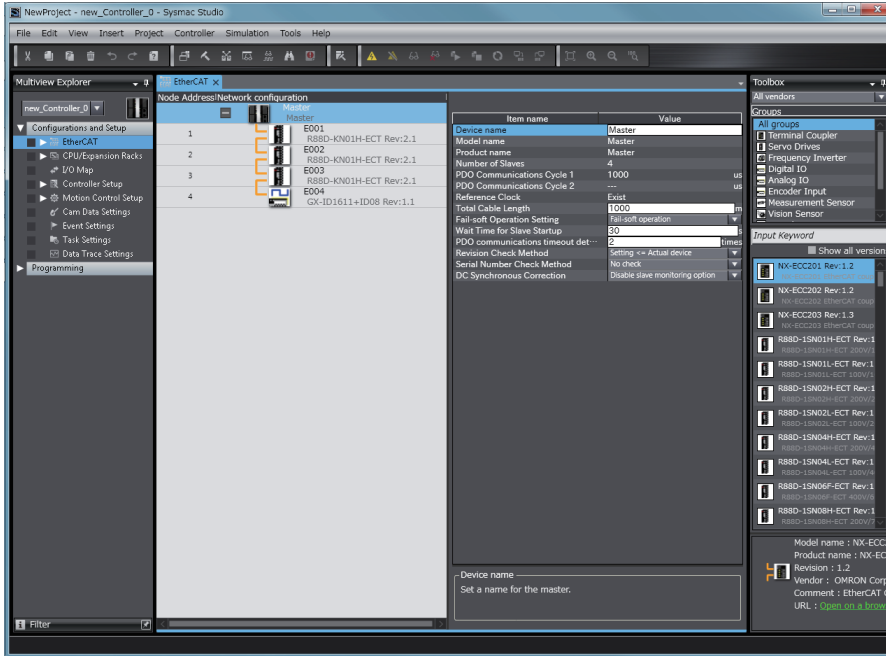
• Parameter Setting for the EtherCAT Slaves
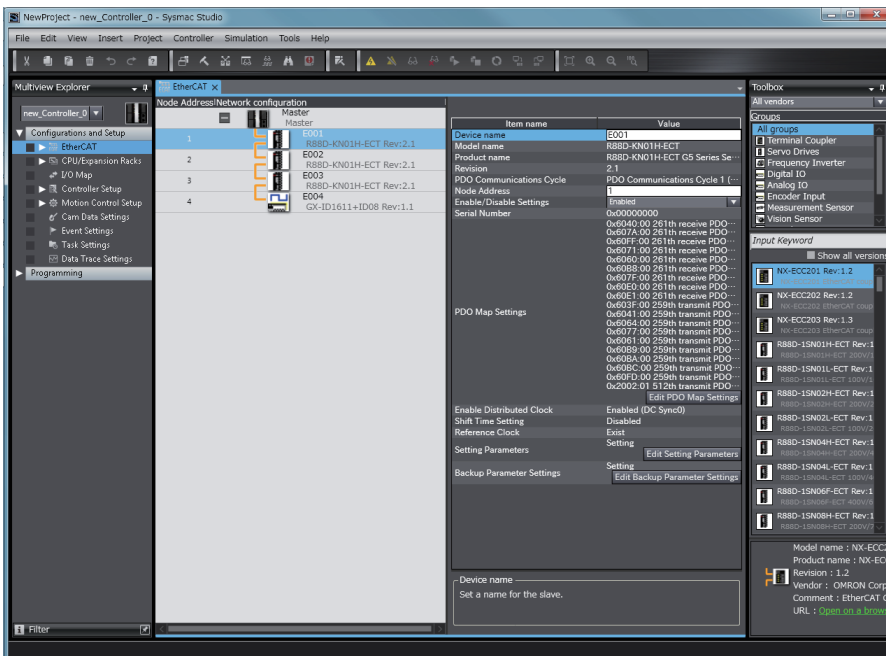
Refer to *Section 5 Setting Up EtherCAT Communications with the Sysmac Studio* for the Sysmac Studio procedures.

2-1-3   Connecting the Sysmac Studio

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

1 Select **Communications Setup** from the Controller Menu.

The Communications Setup Dialog Box is displayed.
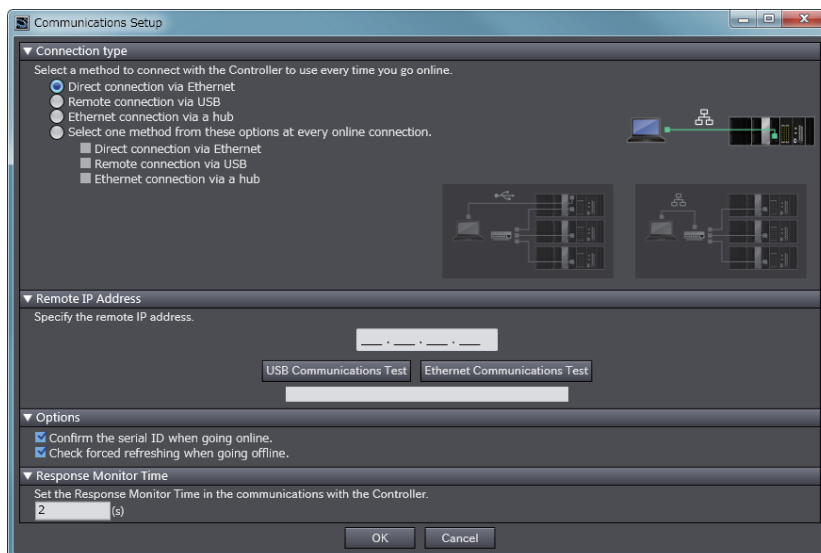
2 Select the connection method for the connection configuration from the Connection type Area.

If you specify a Remote connection via USB or an Ethernet connection via a hub, enter the IP address of the Controller in the Remote IP Address Area. Also set the Options and Response Monitor Time parameter to the required time. Refer to *Communications Setup Dialog Box Settings* on the next page for information on the settings.

3 Click the **OK** Button.

This completes the setup.

● **Communications Setup Dialog Box Settings**

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 will connect to.	
	Ethernet Communications Test Button	These buttons perform a communication test with the Controller at the specified 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.	If you select this option, the names and serial IDs are compared between the project and the Controller when you go online to make sure that a connection is made to the intended Controller.
	Check forced refreshing when going offline.	If you select this option, a check is made to see if any forced refreshing values are still in effect before going offline.
Response Monitor Time	You can set the response monitor 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

● **Setting Node Address Using Hardware Switches**

Setting the node address is described here. An OMRON GX-series Remote I/O Terminal is used as the slave. With a GX-series Remote I/O Terminal, the node address is set on hardware switches. With a GX-series Remote I/O Terminal, these switches are used to set the node address as a slave on the EtherCAT network.
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.

Node address setting, × 1
Node address setting, × 10
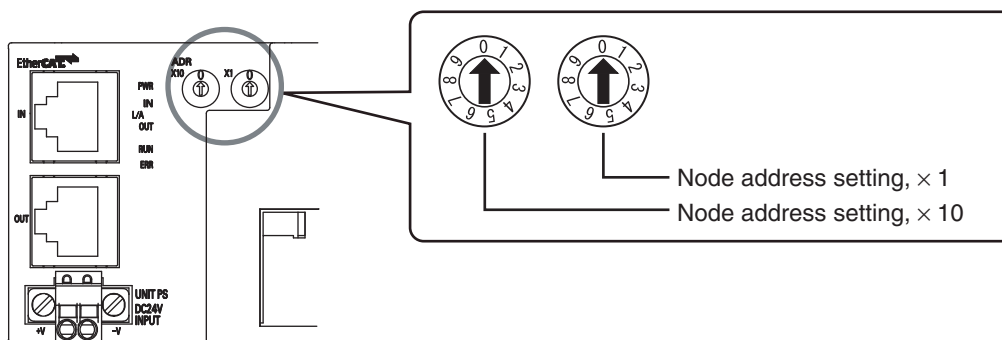
The node address setting values are described in the following table.

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

* Refer to *1-3-1 Performance Specifications* for details on the settable node address range.

● **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.

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

2 Double-click **EtherCAT** under **Configurations and Setups** on the Multiview Explorer. Or, right-click **EtherCAT** under **Configurations and Setups** and select *Edit*.
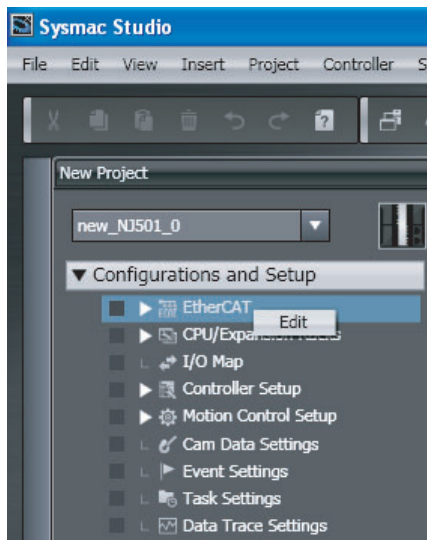
3 Right-click the EtherCAT master that is displayed in the Edit Pane and select *Write Slave Node Address*.
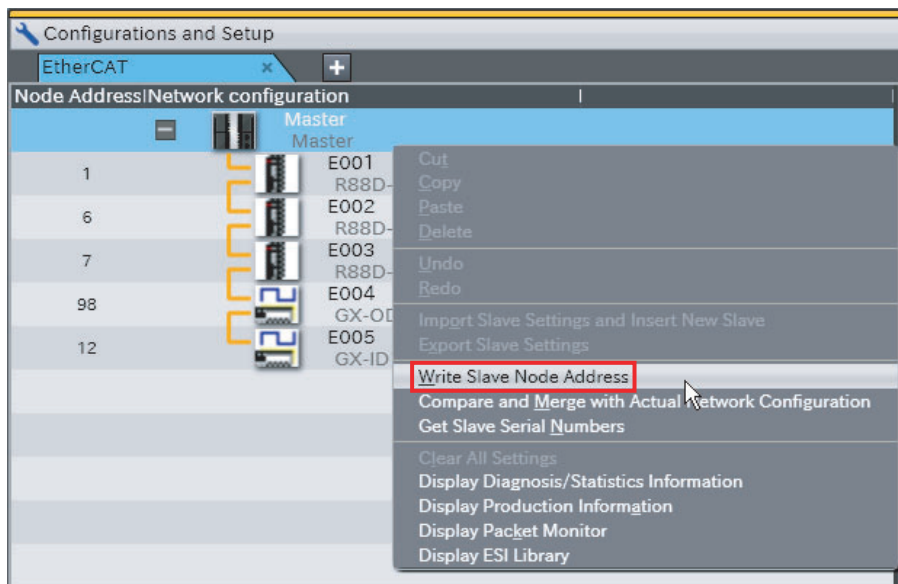
The Slave Node Address Writing Dialog Box is displayed.

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.)
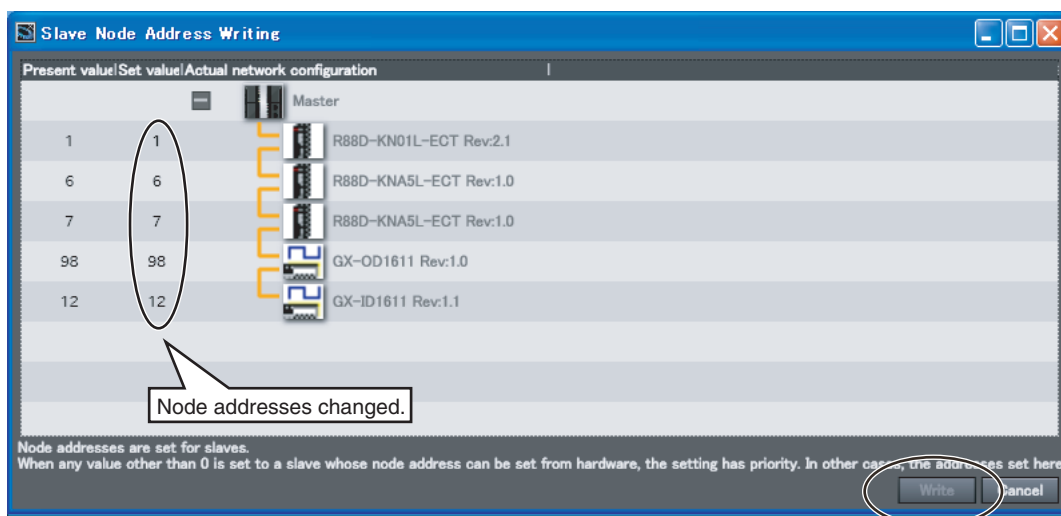
The node addresses are written to the actual slaves.

📑 **Additional Information**

- The 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 occurs and the operation of the slave stops. If a duplicated node address is set, the system-defined variable *_EC_SlavAdrDupErr* (Slave Node Address Duplicated Error) changes to TRUE.

- 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.

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 Industrial PC Platform NY-series IPC Machine Controller Industrial Panel PC / Industrial Box PC 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 Industrial PC Platform NY-series IPC Machine Controller Industrial Panel PC / Industrial Box PC to use the following features.

Sysmac device feature	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 the *NY-series Troubleshooting Manual* (Cat. No. W564) 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 NY-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 slaves are in Init state. When a slave in Init state where a Sysmac error was detected is connected, you cannot confirm the Sysmac error that was detected by the slave until communications are restarted with that slave. Therefore, if an error that results in a slave going to Init state and a slave Sysmac error are detected at the same time, first reset the slave Init state, restart communications with the slave, and then check the Sysmac error. Then, eliminate the cause of the error and reset the error. Refer to the *NY-series Troubleshooting Manual* (Cat. No. W564) for the errors that result in the Slave entering Init state.
> - 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.
> - 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 *NY-series Troubleshooting Manual* (Cat. No. W564) 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 *9-2-3 Backing Up Settings* for the procedure to back up slave data from the Sysmac Studio.
- Refer to *9-2-4 Restoring Settings* for the procedure to restore slave data from the Sysmac Studio.
- Refer to *A-3 Multi-vendor Environments* for information 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
  0: The node address is set from the Sysmac Studio.
  Not 0: 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 every slave so that the slaves can be identified on the EtherCAT network. Do not assign the same node address more than once.
- If you connect slaves from other manufacturers to the NY-series EtherCAT master, set the node addresses for them from the Sysmac Studio. (The NY-series EtherCAT master recognizes the value that is stored at address 0x0012 in the EtherCAT slave controller of the slave as the node address.)
- For OMRON slaves that are not Sysmac devices, set the node addresses on the hardware switches. If the node address switches are set to 0, a Network Configuration Verification Error occurs.

Additional Information

- Refer to *2-2 Setting the Node Addresses of the EtherCAT Slaves* or the manuals for each slave for the procedure to set the node addresses.
- Refer to *5-4-2 Setting EtherCAT Slaves* for information on setting node addresses with 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 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 *5-4-1 Setting EtherCAT Master* for information on checking serial numbers.

2-3-3   List of Sysmac Devices

The following table lists the OMRON Sysmac slaves. Refer to the manual for the 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 Unit*	NX-ECC201 NX-ECC202 NX-ECC203	Revision 1.0 or later Revision 1.2 or later 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 Drive	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

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/NY-series Controller NX5□2-1□□□ NX701-□□□□ NJ501-□□□□ NJ301-□□□□ NJ101-□□□□	CJ1W-NC□8□ CJ-series Position Control Unit	
Troubleshooting	Supported.	Not supported.[1]	Not supported.[1]
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]   Error notification is provided with emergency messages.

[2]   EtherCAT slaves for which data can be backed up have specific precautions. Refer to the *NY-series Industrial Panel PC / Industrial Box PC Software User's Manual* (Cat. No. W558) for details on the precautions.

[3]   You cannot specify data to back up and restore.

[4]   The ability to back up and restore data depends on the ability of the EtherCAT master from another manufacturer.

[5]   The ability to store node addresses depends on the ability of the EtherCAT master from another manufacturer.

[6]   The ability to verify serial numbers depends on the ability of the EtherCAT master from another manufacturer.

OMRON Slaves That Do Not Support Sysmac Devices

Sysmac device feature	OMRON EtherCAT master		EtherCAT master from another manufacturer
	NJ/NX/NY-series Controller NX5□2-1□□□ NX701-□□□□ NJ501-□□□□ NJ301-□□□□ NJ101-□□□□	CJ1W-NC□8□ CJ-series Position Control Unit	
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]   Error notification is provided with emergency messages.

[2]   EtherCAT slaves for which data can be backed up have specific precautions. Refer to the *NY-series Industrial Panel PC / Industrial Box PC Software User's Manual* (Cat. No. W558) for details on the precautions.

[3]   You cannot specify the data that is backed up and restored.

[4]   The ability to back up and restore data depends on the ability of the EtherCAT master from another manufacturer.

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

[6]   The ability to store node addresses depends on the ability of the EtherCAT master from another manufacturer.

*7   The ability to verify serial numbers depends on the ability of the EtherCAT master from another manufacturer.

Slaves from Other Manufacturers

If you connect slaves from other manufacturers to an OMRON EtherCAT master, some slave functions are not supported or restricted as follows.

- When you use an ESI file that is not supported by the Sysmac Studio, some slave functions may be restricted or not settable, or 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)

The EtherCAT port built into the NJ/NX-series CPU Unit uses CoE (CAN application protocol over EtherCAT) to exchange information with slaves over 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.
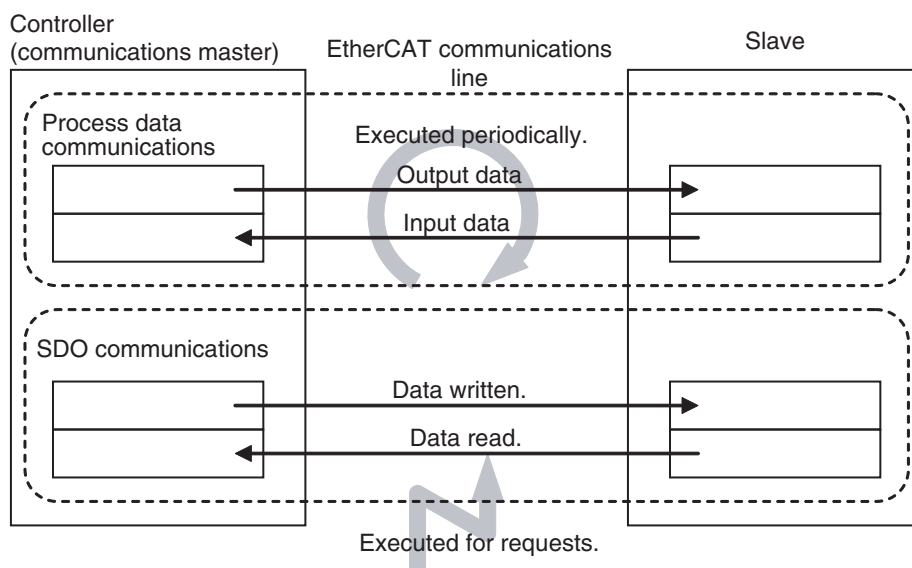
The EtherCAT port built into the NJ/NX-series CPU Unit uses process data communications for commands to refresh I/O data in a fixed control period, including I/O data for EtherCAT slaves, and position control data for Servomotors. It uses SDO communications for commands to read and write data, such as for parameter transfers, at specified times.

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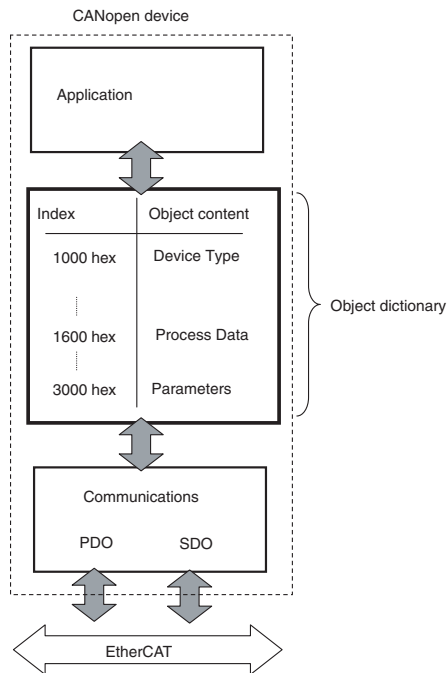
Process data communications is used to read and write the process data in the CoE communications area in these objects.
SDO communications is used to read and write the parameters within the manufacturer-specific area in these objects.

📑 **Additional Information**

Distributed Clock (DC)

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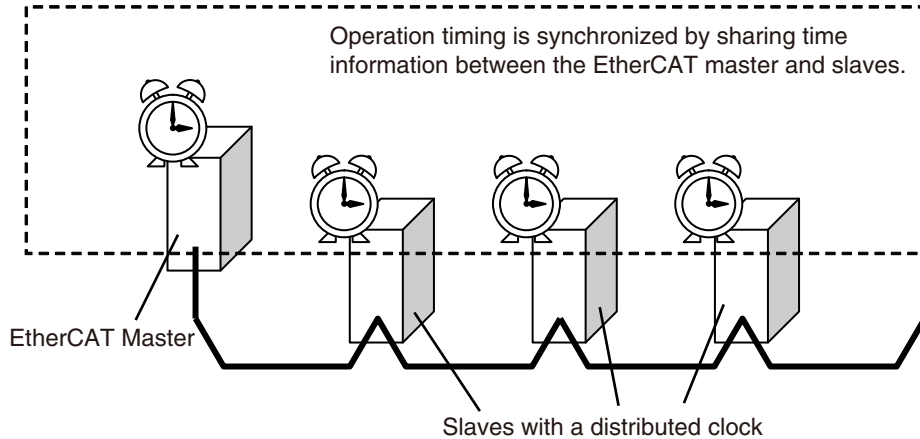
The clock that gives the standard network time is called a "reference clock". The reference clock is provided either by the master or a slave that supports the function to provide a reference clock, to the other slaves. As described in the table below, whether the master or slave provides a reference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

When NX-series CPU Units Is Used	When NJ-series CPU Units 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.*

* 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 that support a distributed clock (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
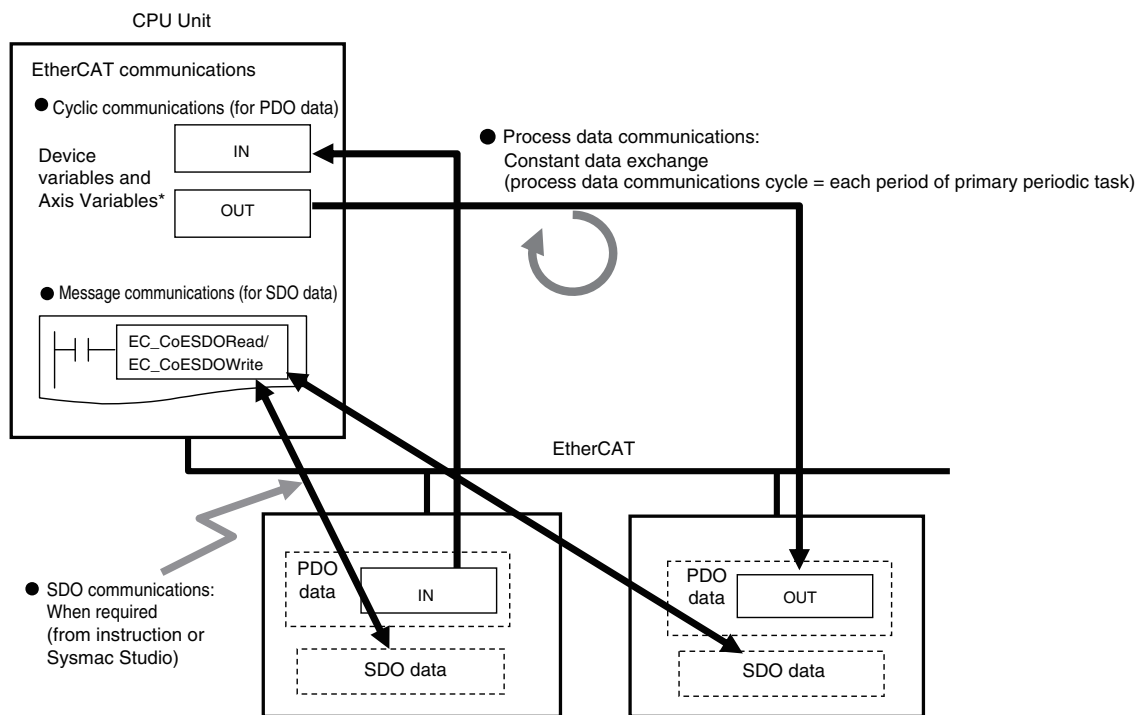
* Device variables for EtherCAT slaves
  Axis Variables only for Servo Drive and encoder input slaves to which axes are allocated

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 NX-series CPU Units.

SDO Communications
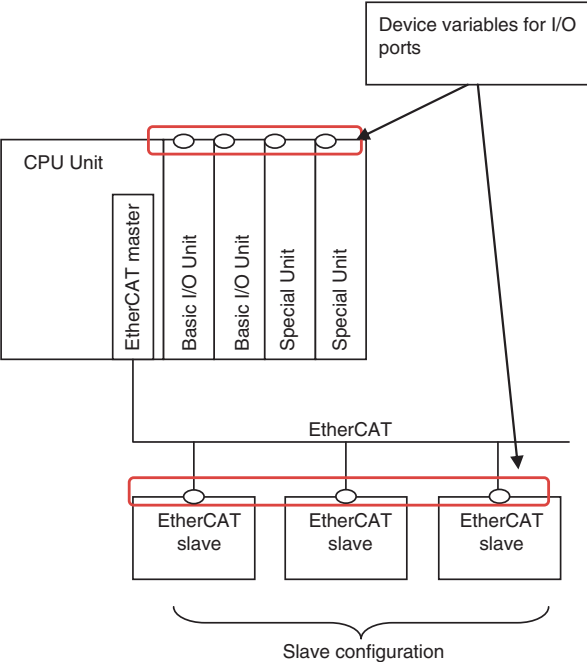
SDO communications are used to read and write specified slave data from the master when required. This is called SDO communications.
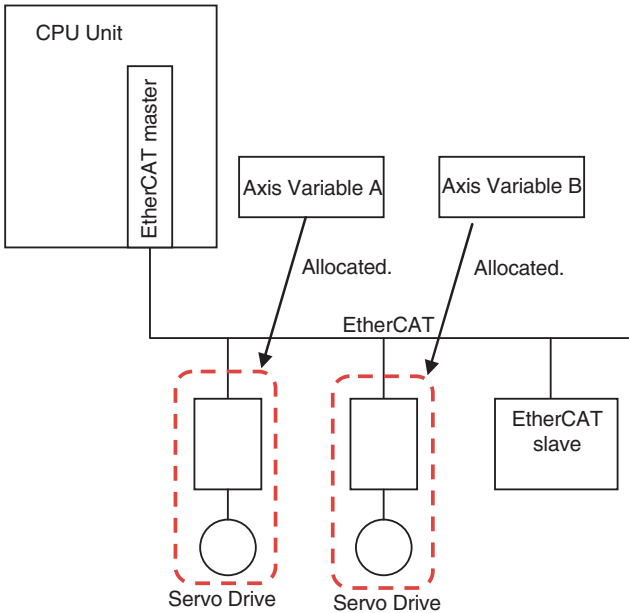
You can read/write the following specified slave data with the EC_CoESDORead (Read CoE SDO) instruction or the EC_CoESDOWrite (Write CoE SDO) instruction.

• 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.

Variable type		Description
Allocated variables	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.
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* 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 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 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 system-defined variables as EtherCAT master errors if errors are detected.

Diagnosis	Detected error type	System-defined variables
Diagnosis of network configuration information	Network configuration information error	_EC_NetCfgErr
Diagnosis of communications port	MAC address error	_EC_MacAdrErr
	Communications controller error	_EC_LanHwErr

Refer to the *NY-series Troubleshooting Manual* (Cat. No. W564) for details on error types.

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.
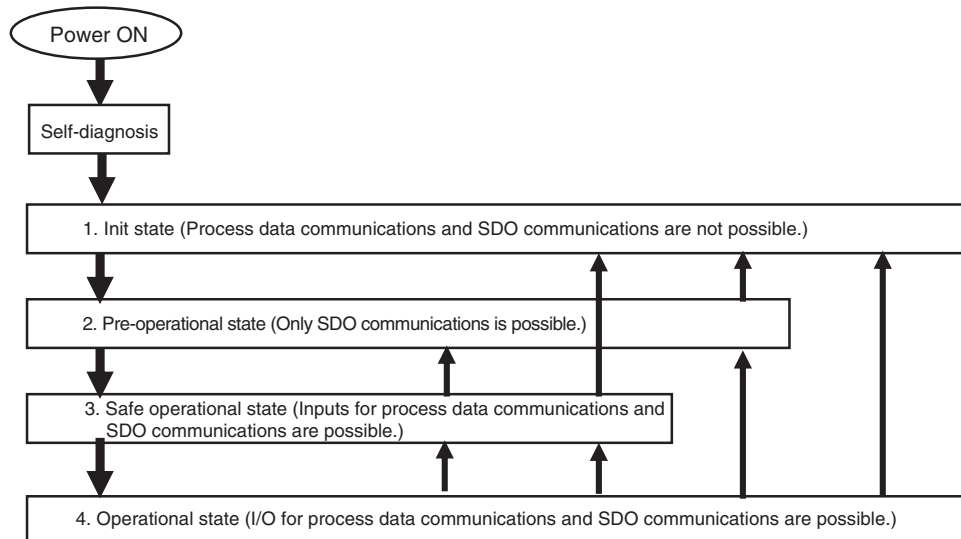
After the power is turned ON, the communications master and slaves go from the Init state to the Pre-operational state, Safe-operational state, and then Operational state before starting EtherCAT communications. Afterwards, EtherCAT communications are performed while the state changes automatically between 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.	Not allowed	Not allowed	Not lit
	This state continues if the network is not configured.			
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_ActSlavTbl[1..n], _EC_MBXSlavTbl[1..n], _EC_PDSlavTbl[1..n], _EC_DisconnSlavTbl[1..n], and _EC_DisableSlavTbl[1..n]
Here, *n* is the maximum value of the settable node address. Refer to *1-3-1 Performance Specifications* 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 system-defined variable _EC_MBXSlavTbl for  │           │
Applicable slave │   the relevant slave changes to TRUE, message         │
parameters can be set. │ communications for the relevant slave are possible.│
 │  └────────────────────────────────────────────────────────┘           │
 │               │                                                        │
 │               ▼                                                        │
 │  ┌────────────────────────────────────────────────────────┐           │
 │  │ 5. When the system-defined variable _EC_PDSlavTbl for   │           │
I/O refreshing started │ the relevant slave changes to TRUE, I/O         │
for the relevant slave. │ refreshing for the relevant slave is enabled.  │
 │  └────────────────────────────────────────────────────────┘           │
 │                                                                        ▼
                                                          User program executed.
```

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

Refer to *A-1 EtherCAT Status in Relation to Controller Status* 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   Laying the EtherCAT Network

This section describes how to install EtherCAT networks.

4-1-1   Supported Network Topologies

The NY-series Controller can be connected using daisy chain connections with no branching, or with branching connections using Junction Slaves. This supports redundancy, and all topologies other than rings. Examples of topology without branching and with branching (Junction Slaves) are shown below.

● **No Branching**
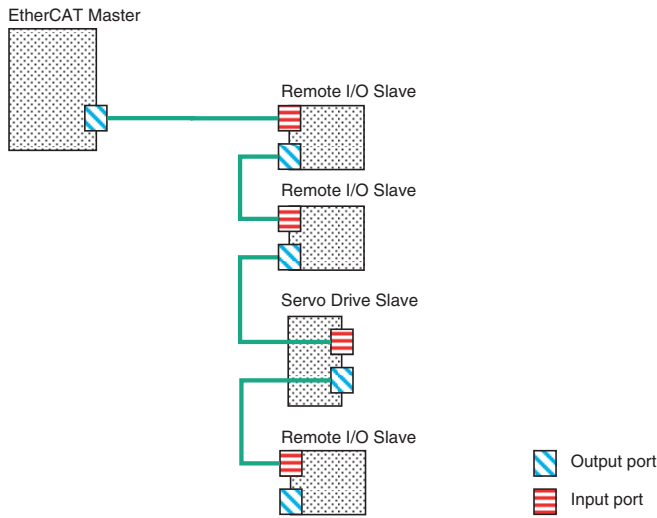
● **Branching**
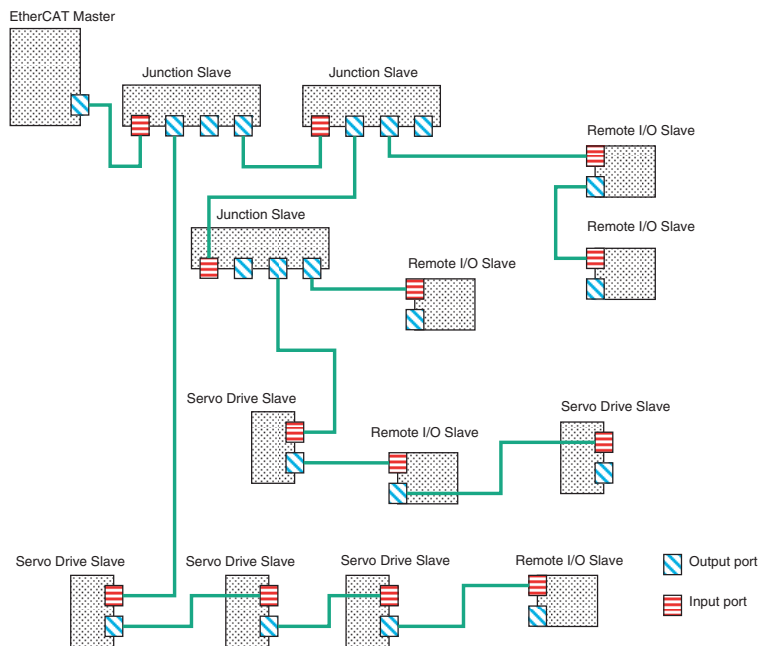
> **▣ Precautions for Safe Use**
>
> You cannot use standard Ethernet hubs or repeater hubs with EtherCAT communications. If you use one of these, a major fault level error or other error may occur.

> **▤ Additional Information**
>
> If you use EtherCAT Junction Slaves, the EtherCAT Junction Slave ports that are actually connected to slaves will communicate normally even if they do not agree with the Sysmac Studio network configuration.
>
> If you need to make the Sysmac Studio network configuration consistent with the ports that are actually connected to slaves, use the compare and merge function of the Sysmac Studio to check the actual configuration.

4-1-2   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 earthing methods may decrease noise immunity and cause device damage. To prevent ground loops caused by differences in potential between device earthing points, the reference potential between the devices must be stabilized. Design earthing appropriately so that noise current does not flow to ground lines between the devices. For earthing methods, refer to the *NY-series Industrial Box PC Hardware User's Manual* (Cat. No. W556) or *NY-series Industrial Panel PC Hardware User's Manual* (Cat. No. W557).

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.

4-1-3   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

● Sizes and Conductor Pairs: AWG 27 × 4 Pairs

Product name	Manufac-turer	Cable length [m]	Model	Contact infor-mation
Standard type * **Cable with Connectors on Both Ends (RJ45/RJ45)**	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	

* 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.

● Sizes and Conductor Pairs: AWG 22 × 2 Pairs

Product name	Manufac-turer	Cable length [m]	Model	Contact infor-mation
Rugged type * **Cable with Connectors on Both Ends (RJ45/RJ45)**	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	
Rugged type * **Cable with Connectors on Both Ends (M12 Straight/M12 Straight) Shield Strengthening cable**	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	
Rugged type * **Cable with Connectors on Both Ends (M12 Straight/RJ45) Shield Strengthening cable**	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	

* 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.

Cables and Connectors

● Sizes and Conductor Pairs: AWG 24 × 4 Pairs

Part name	Manufacturer	Model	Contact information
Cables	Tonichi Kyosan Cable, Ltd.	NETSTAR-C5E SAB 0.5 × 4P*	Planning Department, Kanetsu Co., Ltd.
	Kuramo Electric Co.	KETH-SB*	Kuramo Electric Co.
	SWCC Showa Cable Systems Co.	FAE-5004*	SWCC Showa Cable Systems Co.
RJ45 Connectors	Panduit Corporation	MPS588*	Panduit Corporation US Headquarters

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

● Sizes and Conductor Pairs: AWG 22 × 2 Pairs

Part name	Manufacturer	Model	Contact information
Cables	Kuramo Electric Co.	KETH-PSB-OMR*	Kuramo Electric Co.
	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.

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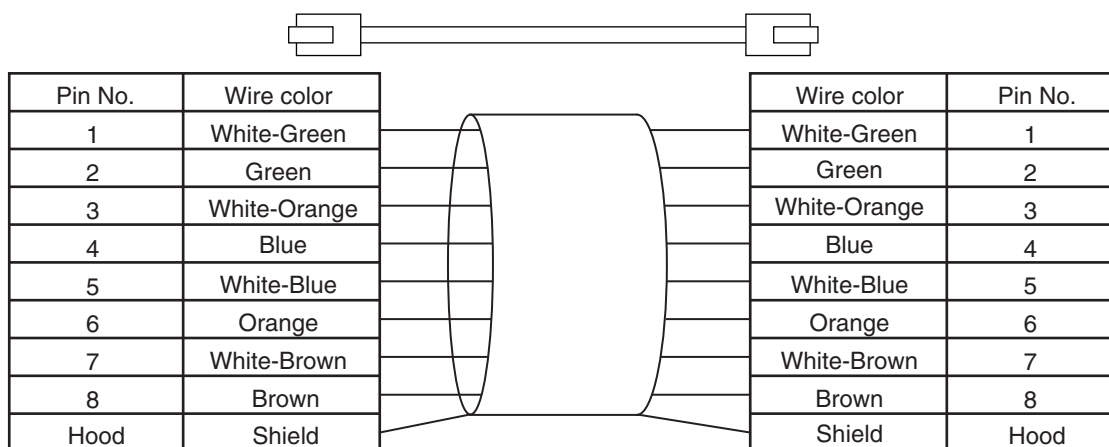
*1   Connect the cable shield to the connector hood at both ends of the cable.

*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	---

4-1-4   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.
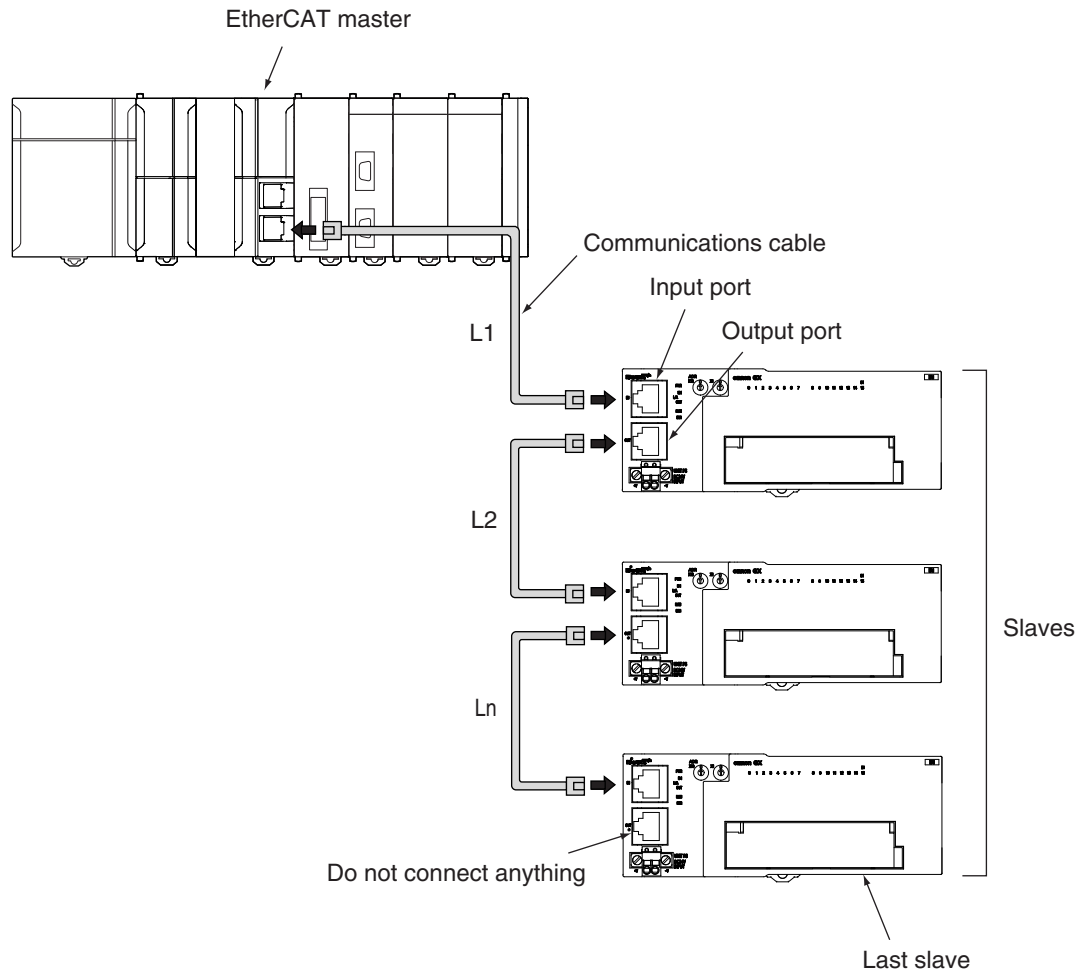
The cable between any two nodes (L1, L2 ... Ln) must be 100 m or less. Use the Sysmac Studio to confirm if communications are possible for the total cable length. Firmly connect the communications cable connector until it clicks into place.

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 *9-2 Replacing Slaves during Communications* 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.

4-1-5   Cable Connection Procedure

![icon] **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 Controller that are used. Refer to the *NY-series Industrial Box PC Hardware User's Manual* (Cat. No. W556) or *NY-series Industrial Panel PC Hardware User's Manual* (Cat. No. W557).

1   Lay the Ethernet communications cable.

2   Connect the Ethernet communications cable to the built-in EtherCAT port on the NY-series Controller. Firmly insert the connector until it locks into place.
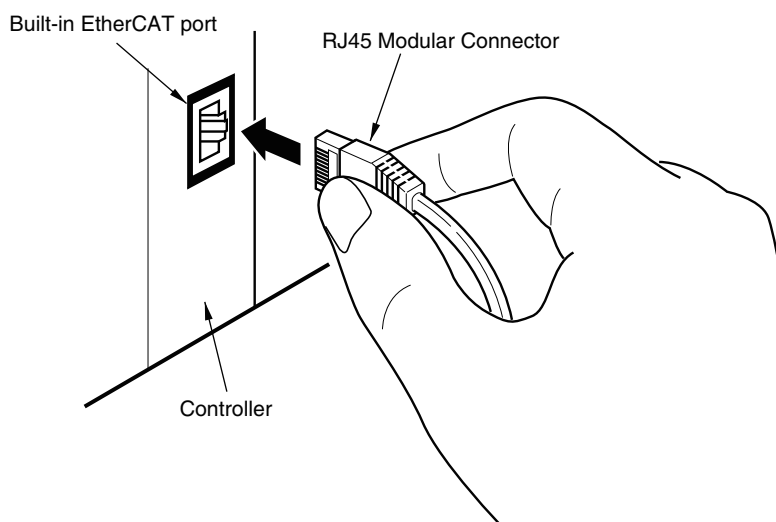
Built-in EtherCAT port

RJ45 Modular Connector

Controller

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 Controller. When this data is set, the network configuration information is automatically created.

When you download the network configuration information to the Controller, 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	1. EtherCAT network configuration
	2. Process data information This is allocation information for PDO data in the slaves. It is also called PDO mapping data.
	3. EtherCAT master and slave parameter settings (Examples: Fail-soft Operation Setting and Wait Time for Slave Startup)
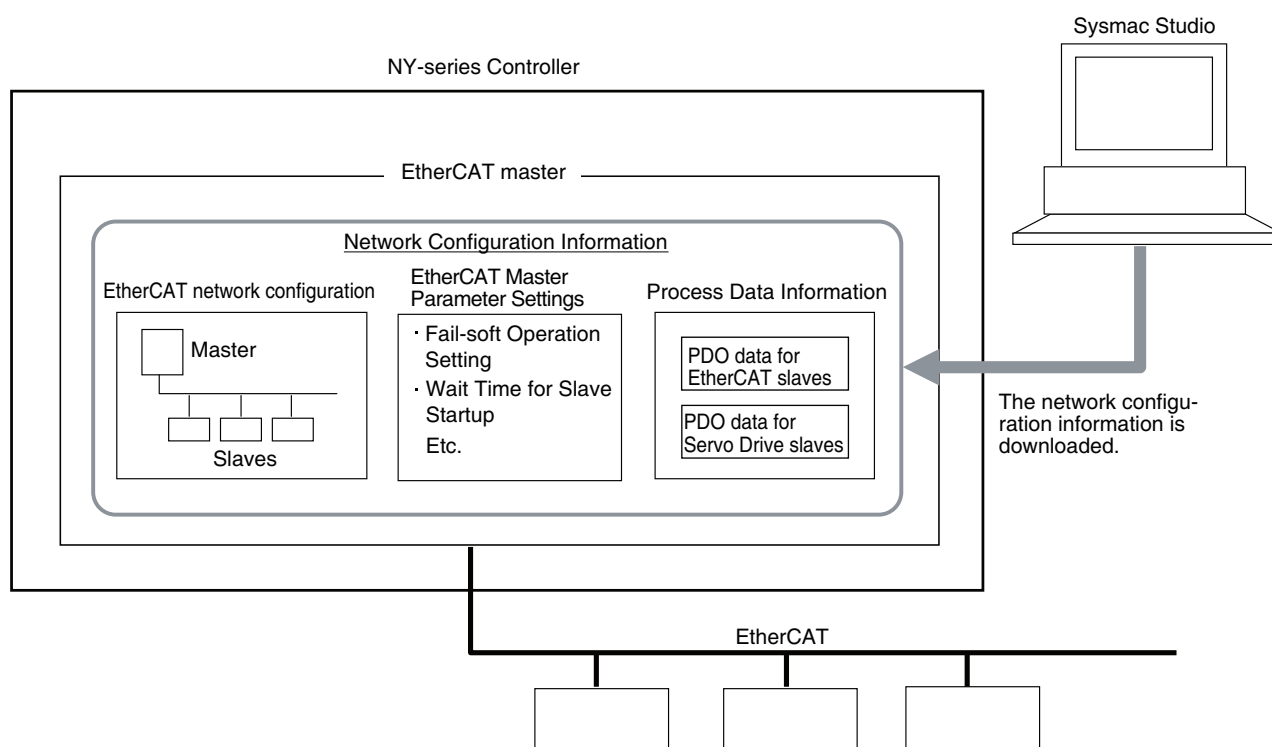
5-2 Creating the EtherCAT Network Configuration

This section describes how 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*.
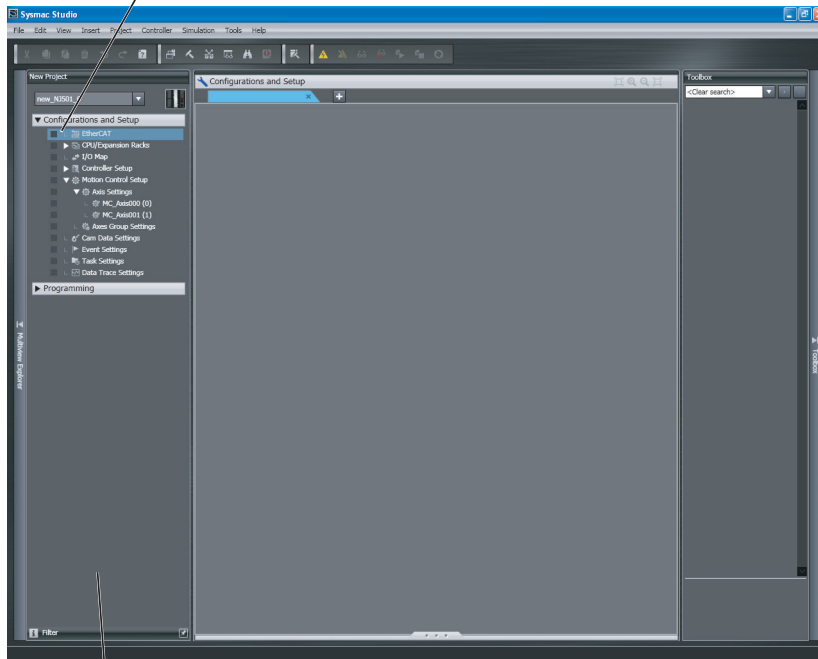
Multi-view Explorer

EtherCAT master will be displayed in the Edit Pane.
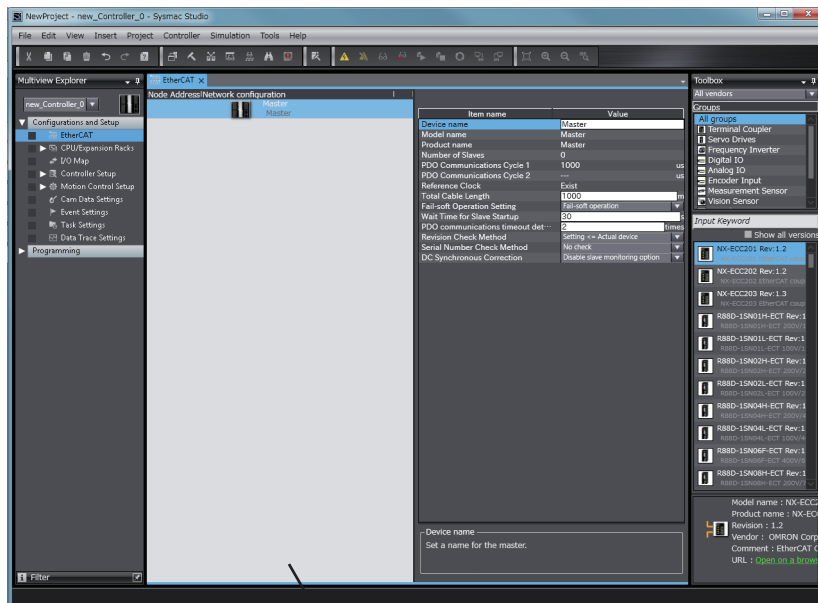
Edit Pane

3 Select a slave in the Toolbox, drag it to the Edit Pane, and drop it under the master. The slave will be added under the master.
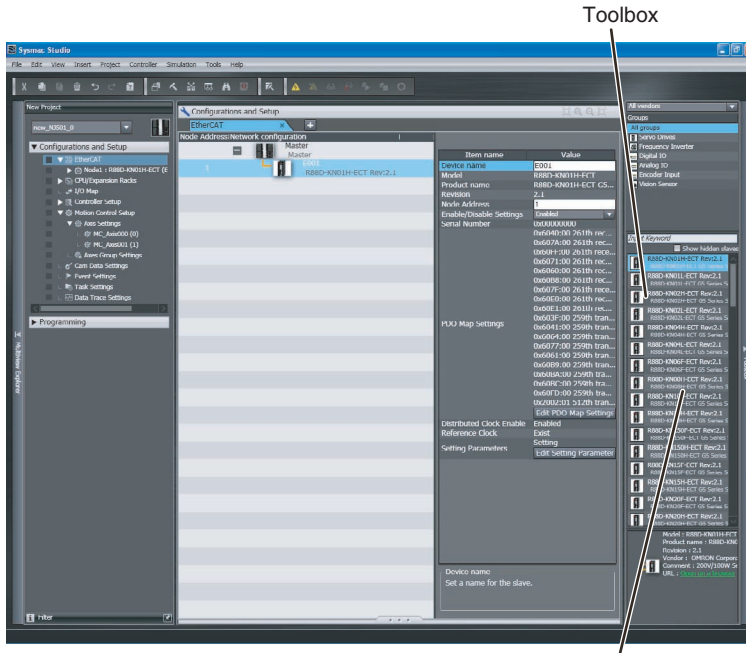
Toolbox

Drag the slaves.

4 Select another slave in the Toolbox, drag it to the Edit Pane, and drop it under the slave to which you want to connect it to. The slave will be added under the slave.
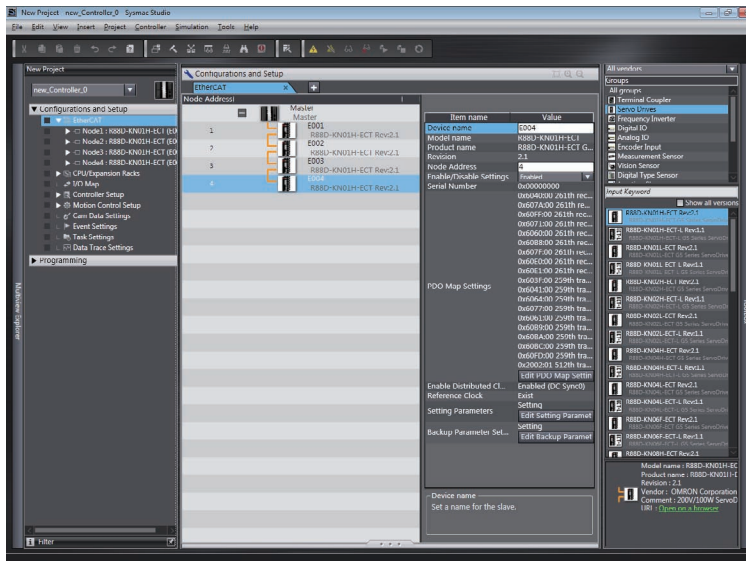
For detailed procedures, such as those for deleting slaves, or copying and pasting slaves, refer to the *Sysmac Studio Version 1 Operation Manual* (Cat. No. W504).

Refer to the *NX-series EtherCAT Coupler Unit 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 *A-8 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 *5-5-2 Automatically Creating the Network Configuration*.

5-2 Creating the EtherCAT Network Configuration

5

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 Tab Page 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 programming 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*.
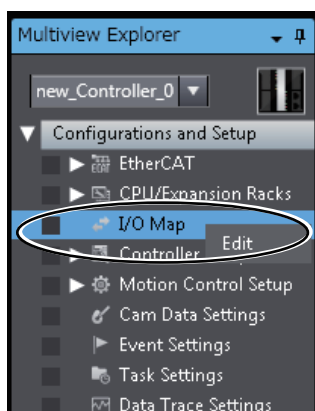
5 Select the variables from the pull-down list in the I/O Map Tab Page to assign them to I/O ports.
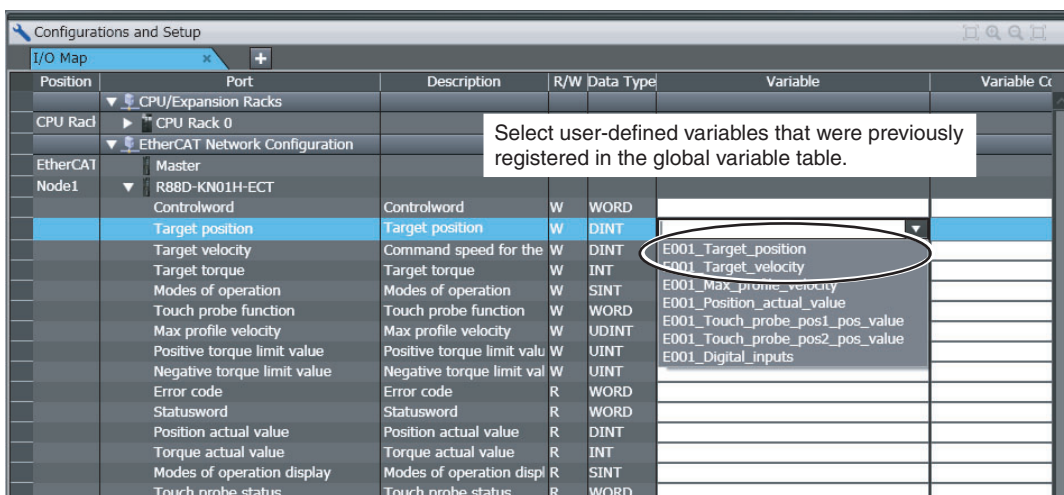
Inputting New Device Variable Names

You can input the required device variable names. For example, this method can be used to register device variables in the following case.

• 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 Tab Page and enter the variable name in the *Variable* Column.
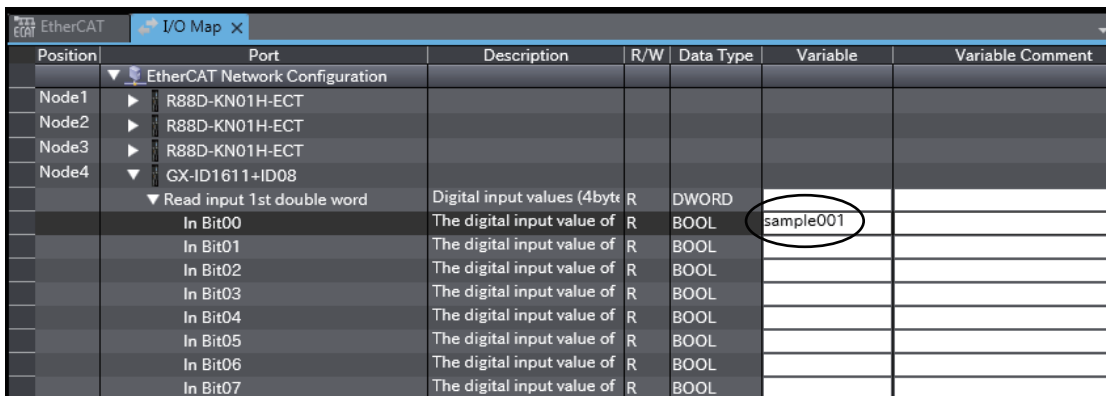
Device variables are assigned to the I/O ports of the slaves.

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

Automatically Creating Device Variable Names

The device variable names will 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 case.

- 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*.
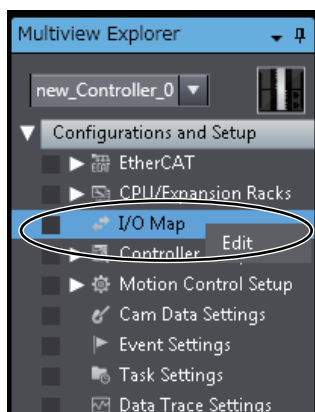
The I/O Map Tab Page will be displayed.

2 Select a slave or I/O ports on the I/O Map Tab Page, right-click, and select **Create Device Variable**.
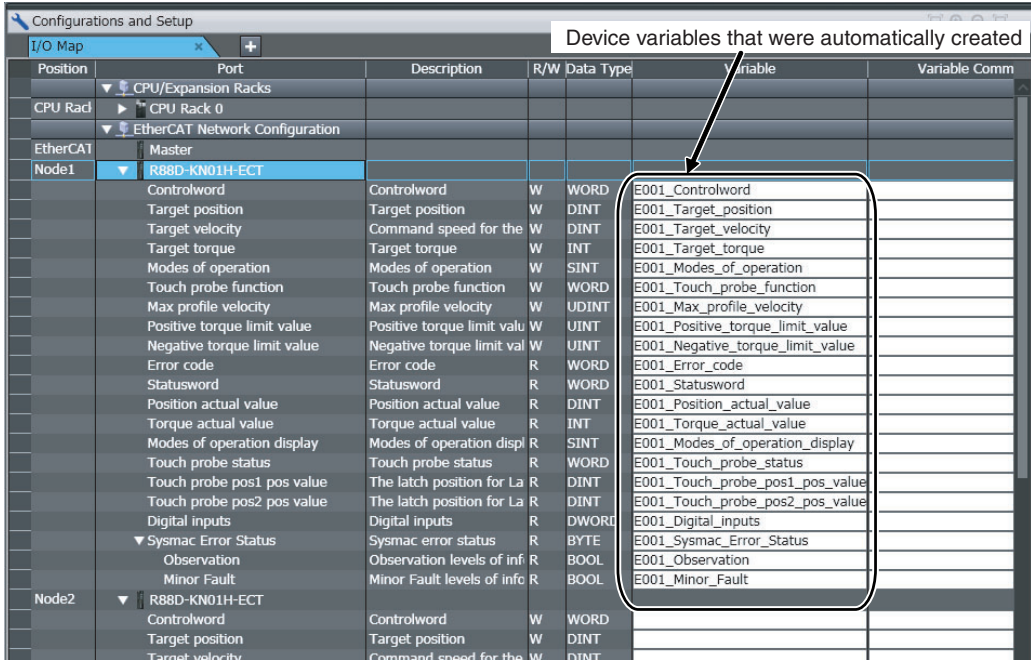
3 If you specify a variable table for the scope, specify the *Variable Type* afterward.
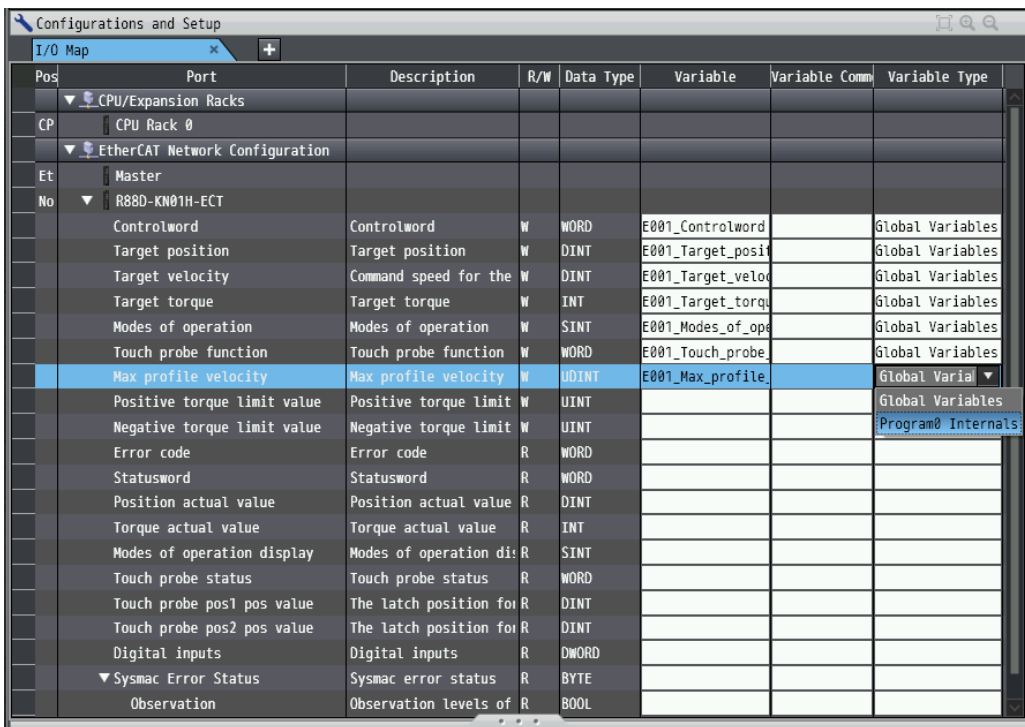
Device variables are automatically assigned to the I/O ports of the slaves and registered in the variable table that is specified by the variable type.

📑 **Additional Information**

- We recommend that you set 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 *NY-series Industrial Panel PC / Industrial Box PC Software User's Manual* (Cat. No. W558) for details on device 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. For details on the items and procedures of the axes setting, refer to the *NY-series Industrial Panel PC / Industrial Box PC Motion Control User's Manual* (Cat. No. W559).

5-4   EtherCAT Master and Slave Parameter Settings

5-4-1   Setting EtherCAT Master

1   Double-click **EtherCAT** under **Configurations and Setup** on the Multiview Explorer. Or, right-click **EtherCAT** under **Configurations and Setup** and select *Edit*.
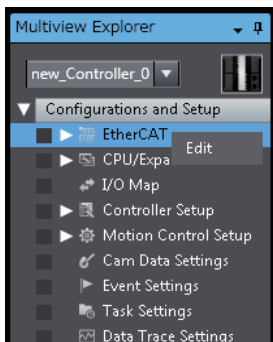
2   Select the EtherCAT master that is displayed in the Edit Pane.
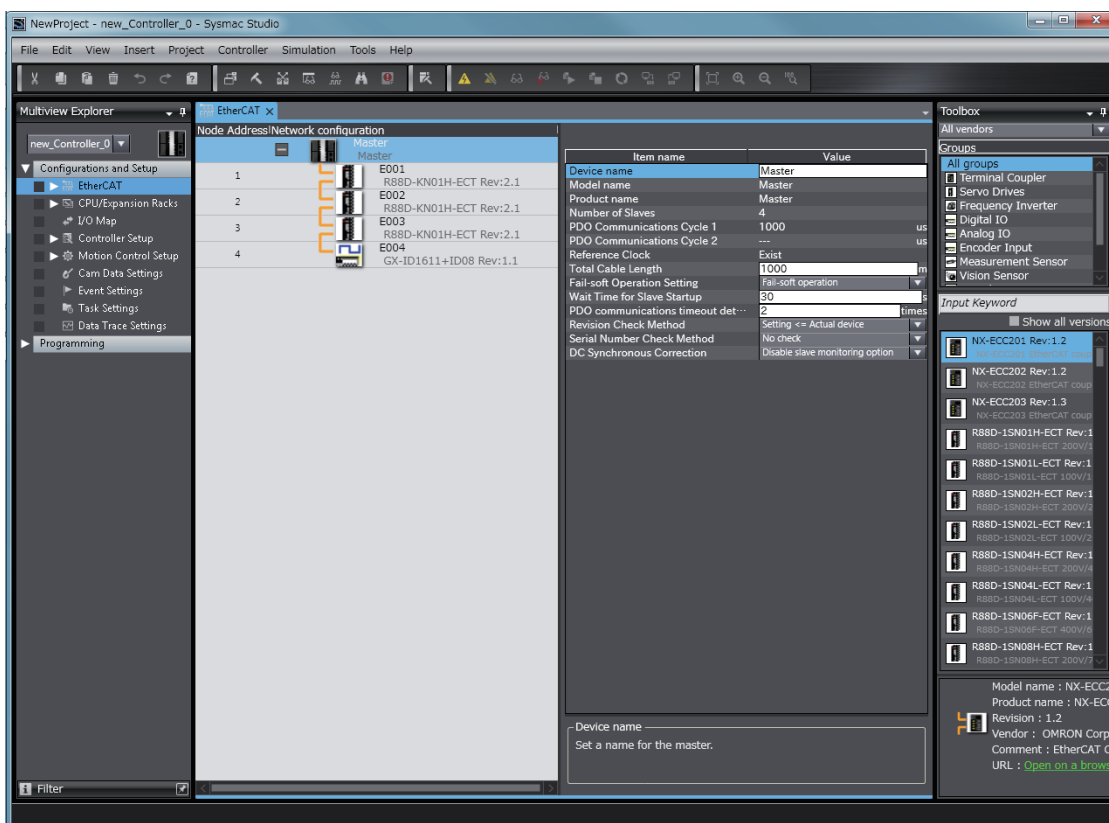
The EtherCAT master settings are listed below.

Name	Editing	Remarks
Device name	OK	Displays the name of the EtherCAT master.
		Default setting: Master
Model	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	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 8,000 μs.
		Default setting: 1,000 μs
Reference Clock	Not allowed.	Displays whether the master provides a reference clock.[2]
		Exist: The master provides a reference clock.
		Not Exist: The master does not provides a reference clock.
Total Cable Length[3]	OK	This is the total cable length between the slaves.
		Setting range: 1 to the maximum number of slaves $\times$ 100 m[1]
		Default setting: 1,000 m
Fail-soft Operation Setting	OK	Select *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
		Refer to the *NY-series Troubleshooting Manual* (Cat. No. W564) for details on errors.
Wait Time for Slave Startup	OK	Set the time to wait from the detection of the link on the EtherCAT port until all slaves join 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 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 error occurs if process data (PDO) communications timeouts occur continuously more often than the specified number of times.
		Increase the setting if Process Data Reception Timeout errors occur frequently, such as in high-noise environments.
		Set the value to 2 or higher for a slave that needs to be replaced.
		Setting range: 1 to 8 errors
		Default setting: 2 errors

Name	Editing	Remarks
Revision Check Method	OK	Specify 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 will occur.
		Setting values: Setting <= Actual device, Setting = Actual device, or No check[*4]
		Default setting: Setting <= Actual device
Serial Number Check Method[*5]	OK	Specify 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 will occur.
		Setting values: Setting = Actual device or No check
		Default setting: No check
DC Synchronous Correction	OK	Specify 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 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 *1-3-1 Performance Specifications* for the maximum number of slaves.

*2 *Exist* is displayed with the NY-series Controller.

*3 The total cable length is set because it is used for simulation of the task execution times on the Sysmac Studio. Setting the total cable length will enable taking into consideration the delay that is caused by the cable lengths when the Sysmac Studio estimates the task execution times.
The operation of the Controller is not affected even if the setting of the total cable length is not correct (e.g., if you do not change the default setting). If you use the default setting, the simulation results will reflect the delay for a 1,000-meter total cable length. If this setting is different from the actual total cable length, the delay on the actual system will be different than the delay in the simulation results.

*4 Refer to the following table, *Communications with Slaves Based on the Result of Revision Number Check*, for the operation of communications with slaves that are determined by the result of revision number check.

*5 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.

5-4 EtherCAT Master and Slave Parameter Settings

5

5-4-1 Setting EtherCAT Master

● **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.

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 verifying revisions is disabled, 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 disable the revision check, make sure that only compatible slaves are connected before transferring the parameters.

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, the system-defined variable for a Network Configuration Verification Error provides notification.

This setting applies to the entire system and cannot be changed for individual slaves.

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][2]	Serial numbers of all slaves

*1 Options are available.

*2 Serial number verification is performed for all slaves in the EtherCAT topology. You cannot specify different verification methods for individual slaves. The default setting is *No check*.

5-4-2   Setting EtherCAT Slaves

1   Double-click **EtherCAT** under **Configurations and Setup** on the Multiview Explorer. Or, right-click **EtherCAT** under **Configurations and Setup** and select **Edit**.
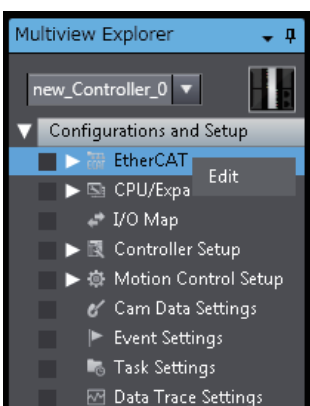
2   Select an EtherCAT slave that is displayed in the Edit Pane.
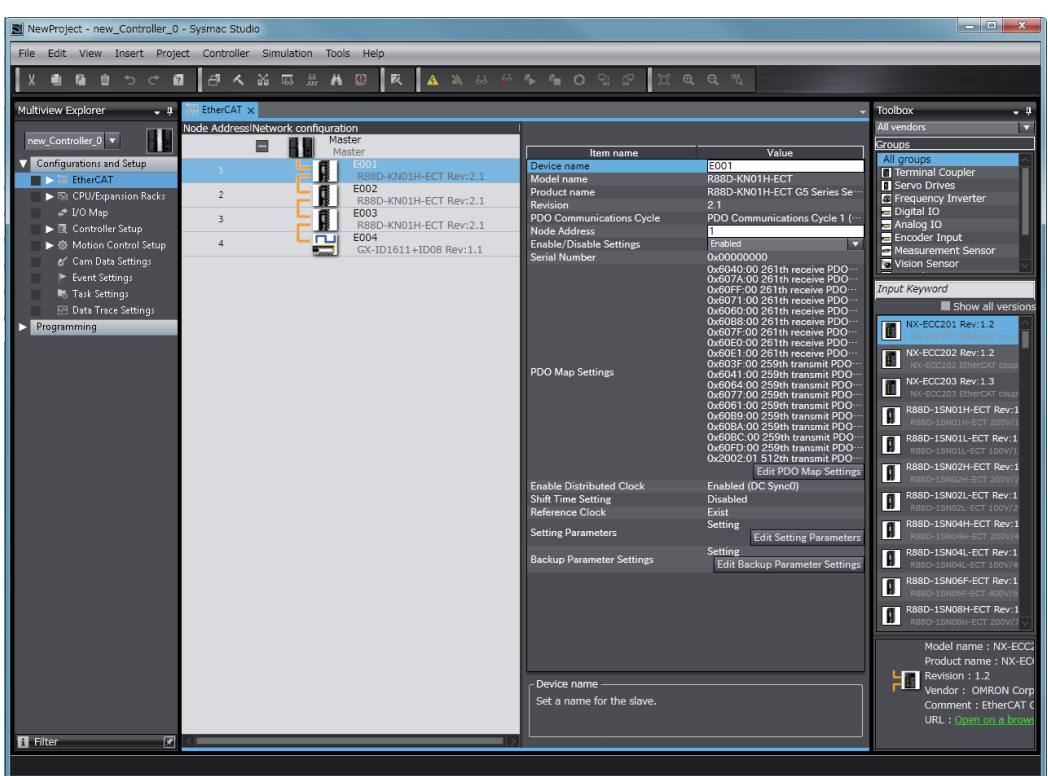
The EtherCAT slave settings are listed as below.

Name	Editing	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	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.

Name	Editing	Remarks
PDO Communications Cycle	Not allowed.	Displays the process data communications cycle assigned for the slave. • PDO Communications Cycle 1 (XXXX µs) Communications cycle is the task period of the primary periodic task. XXXX indicates the set value for the current communications cycle. Example: PDO Communications Cycle 1 (8,000 µs)
Node Address	OK	Sets the node address. The default value is set automatically when a slave is added. Setting range: Settable node address range[*1]
Enable/Disable Settings	OK	Enables or disables the slave as a communications target. • Enabled: The slave will operate. • Disabled: The slave will not operate.[*2] 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 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 *5-5-3 Using the Sysmac Studio to Obtain Serial Numbers from the Actual Network Configuration* 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 *Changing the PDO Map Settings* for details.

Name	Editing	Remarks
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. *Exist* is displayed if the slave provides a reference clock. *Not exist* is displayed if the slave does not provide a reference clock.
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 Refer to *1-3-1 Performance Specifications* for the settable node address range.

*2 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 to 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.

5-4 EtherCAT Master and Slave Parameter Settings

5

5-4-2 Setting EtherCAT Slaves

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.
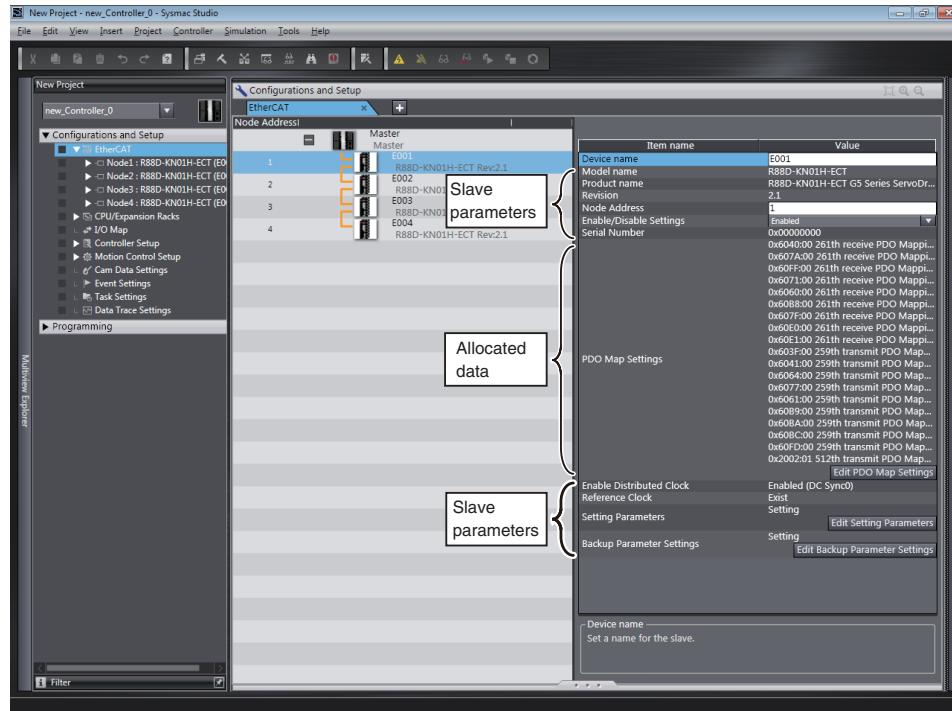
2 Click the **Edit PDO Map Settings** Button at the lower right of the allocated data in the pane.
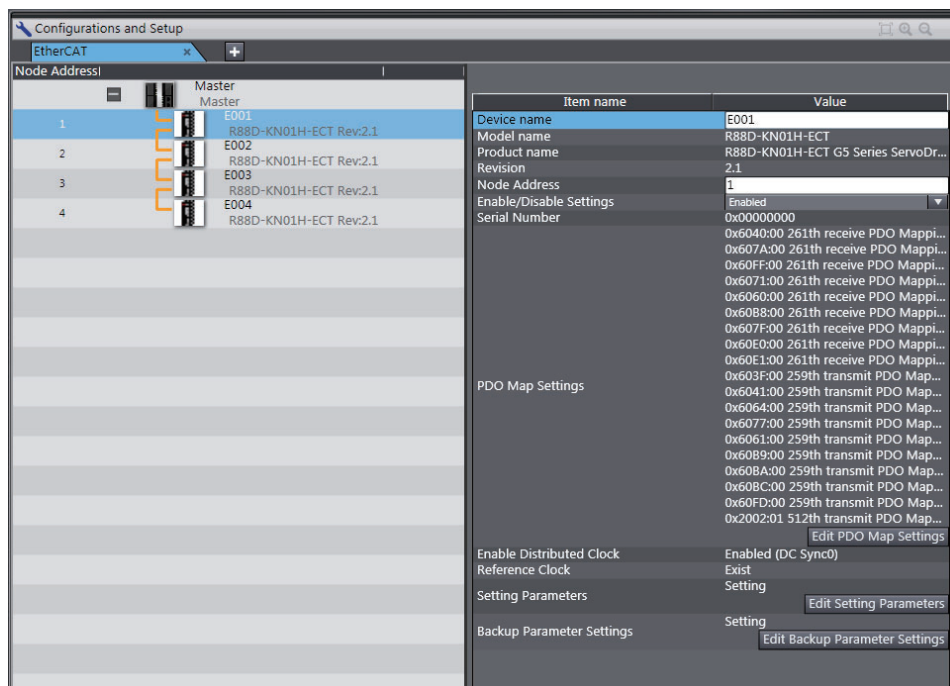
The Edit PDO Map Settings Window will appear. Edit the allocated data as required.
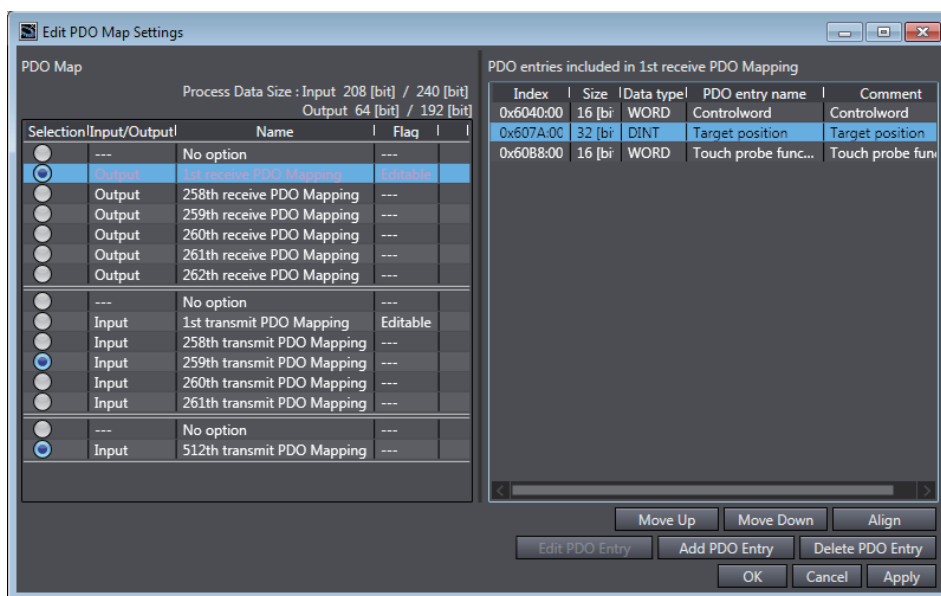
3 Select the output data (RxPDO) and input data (TxPDO) in the PDO Map.
You can add or delete the PDOs with Editable Flags as entries for the objects to use for the slaves. Refer to *A-3-4 Editing PDO Entry Tables* for details.

4 Click the OK Button.

EtherCAT Slave Enable/Disable Settings

● Relation between Enable/Disable Settings and Instruction Settings

The following table gives the relation between the enable/disable setting of an EtherCAT slave and the results of instruction execution.

EtherCAT slave setting	Instruction setting	Result
Enable	Enable	Enabled (normal end)
	Disable	Disabled
Disabled with physical slaves[1], [2]	Enable	Enabled
	Disable	Disabled (normal end)
Disabled with no physical slave, or disconnected	Enable or disable	Error (error end)

[1] Even if a slave physically exits, it cannot be enabled or disabled if there are any of the following errors.
   • If there is a Network Configuration Information Error or Network Configuration Error in the master.
   • If the slave is disconnected from the network or if it has a Slave Initialization Error, Process Data Communications Error, Slave Application Error, or EtherCAT Frame Not Received Error.

[2] If a slave is disabled (either with or without a physical slave), you cannot disconnect the slaves.

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 Controller 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 Controller.

● **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

For a slave that changed from enabled to disabled, the values of the device variables from before the change are retained.

Use the following processing to confirm if a slave is enabled or disabled.

(1) **Confirm that the slave is enabled before you access the device variables of the input device.**
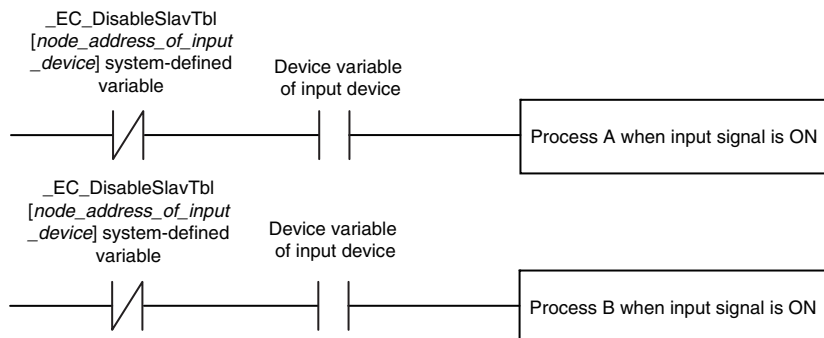
(2)   If the input device is disabled, always write the value of the device variable.
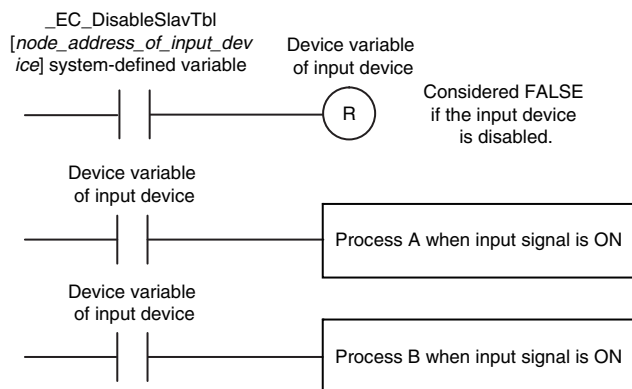
● **Relation between Operation and the Slave Information at a Disabled Slave**

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 com- munications**	**Process data communica- tions**	**Error**	**Back up/ Restore/verify**
Same	Same	Performed	Not performed	Normal opera- tion (no error)	Performed
Same	Different	Not performed	Not performed	Network Config- uration Verifica- tion Error	Not performed
Different	Same	Not performed	Not performed	Network Config- uration Verifica- tion Error	Not performed
Different	Different	Not performed	Not performed	Network Config- uration Verifica- tion Error	Not performed

5-5   Comparing and Merging EtherCAT Network Configurations

5-5-1   Comparing and Merging with the Actual Network Configuration from the Sysmac Studio

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 lines and branch locations. The following items are compared.

- Node addresses
- Vendor IDs
- Product codes
- Revisions
- Connected ports

If the settings in the Sysmac Studio and the actual network configuration match completely, it is possible to determine the locations of errors when they occur. If differences are shown in the comparison results, merge the configurations from the Sysmac Studio.
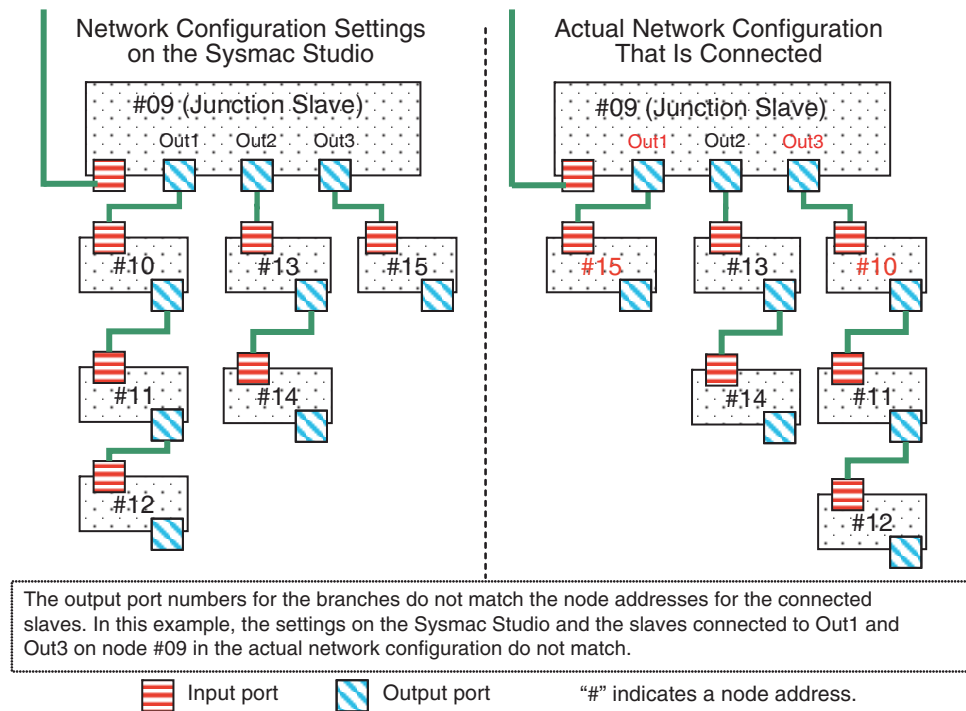
Use the following procedure to compare and merge the network configuration settings on the Sysmac Studio and the actual network configuration.

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

2   Double-click **EtherCAT** under **Configurations and Setup** in the Multiview Explorer. Or, right-click **EtherCAT** under **Configurations and Setup** and select **Edit**.
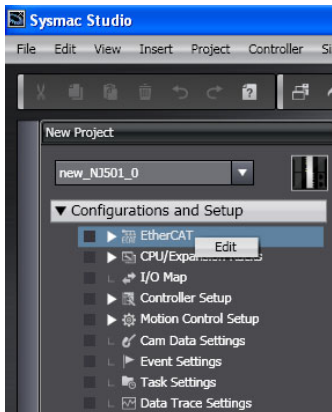
The Edit Pane is displayed.

3   Right-click the EtherCAT master that is displayed in the Edit Pane and select **Compare and Merge with Actual Network Configuration**.
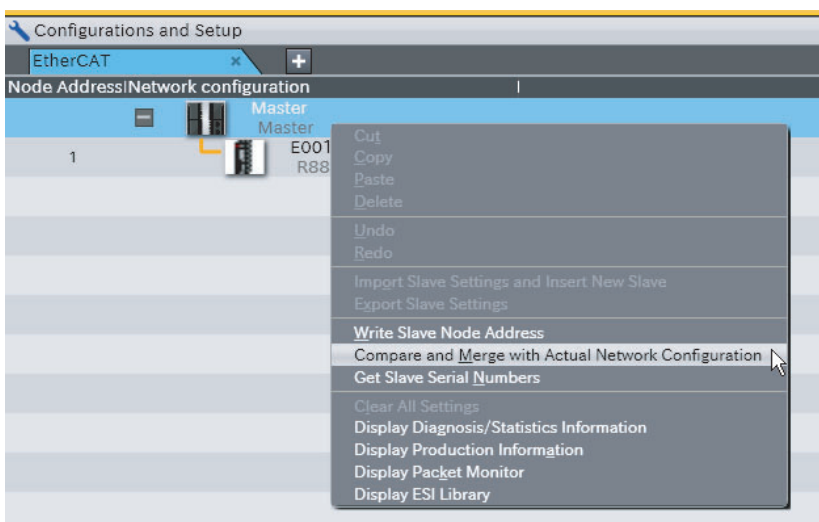
The Compare and Merge with Actual Network Configuration Dialog Box is displayed. The results of comparing the settings on the Sysmac Studio with the actual network configuration are displayed in the *Comparison results* Column.
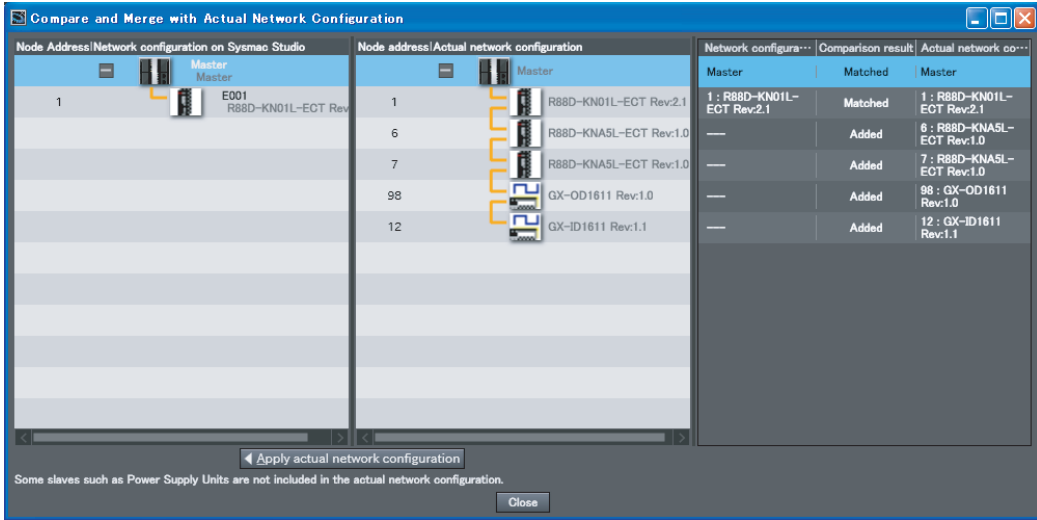
4 If there is a slave that exists only on the actual network, "Added" is displayed in the *Comparison results* Column. Drag the slave in the actual network configuration to the Sysmac Studio network configuration.
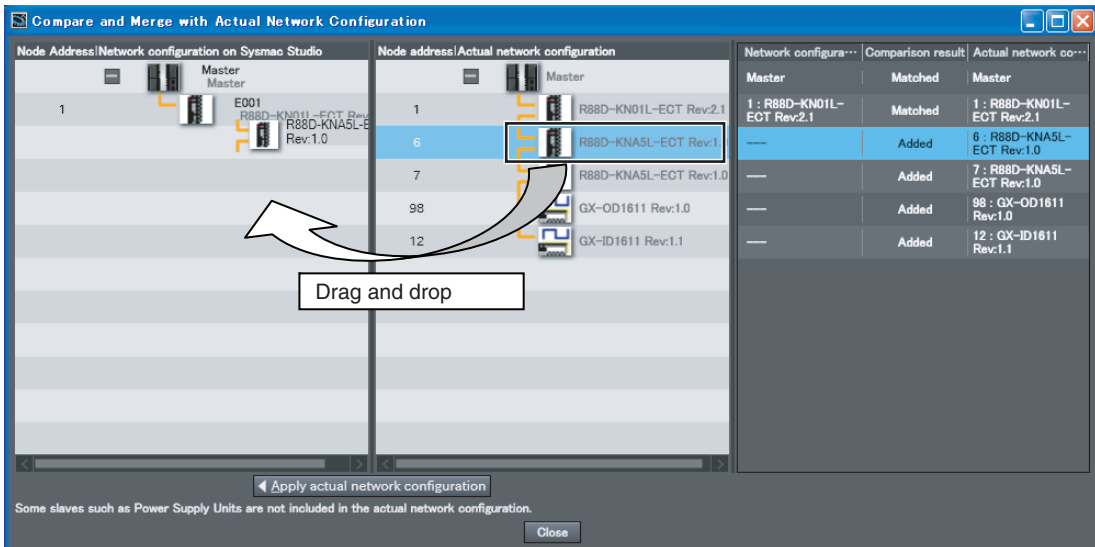
After you drag the missing slaves and add them to the network configuration on the Sysmac Studio, the *Comparison results* Column will show that everything in the configurations matches.

5 Click the **Close** Button.
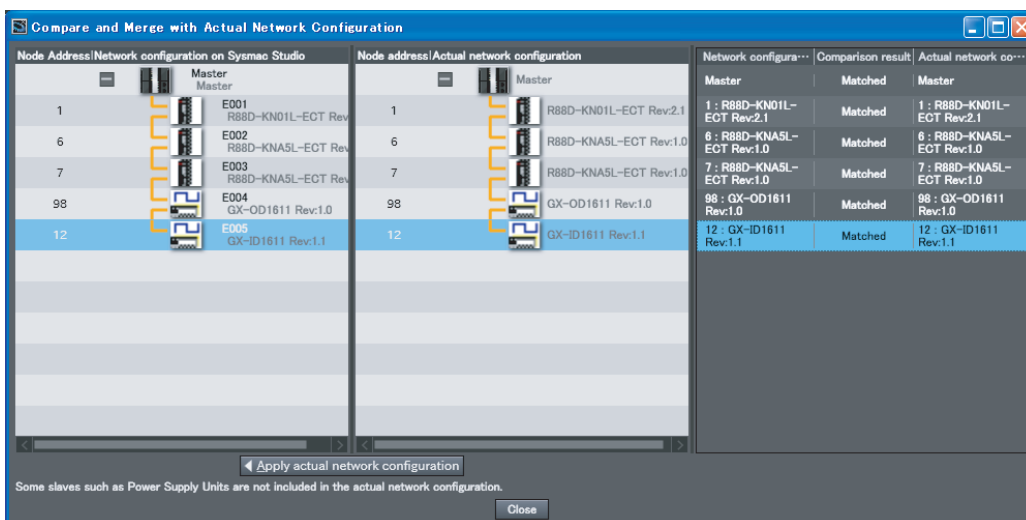
The Sysmac Studio returns to the Edit Pane. This completes the operation to compare and merge with the actual network configuration.

"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.

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**

- Make sure that the communications cables between the master and slaves are connected correctly before you perform the compare and merge operation. The Compare and Merge with Actual Network Configuration Dialog Box is not displayed if the connections are not correct.
- The Compare and Merge with Actual Network Configuration Dialog Box is not displayed 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 the compare and merge operation.
- The Compare and Merge with Actual Network Configuration Dialog Box is not displayed if an EtherCAT Frame Not Received error occurs. Remove the cause of the error and perform the compare and merge operation. Refer to the *NY-series Troubleshooting Manual* (Cat. No. W564) for the cause of the error.
- When the compare and merge operation is performed, 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.

5-5-2   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 actual 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** in the Multiview Explorer. Or, right-click **EtherCAT** under **Configurations and Setup** and select *Edit*.
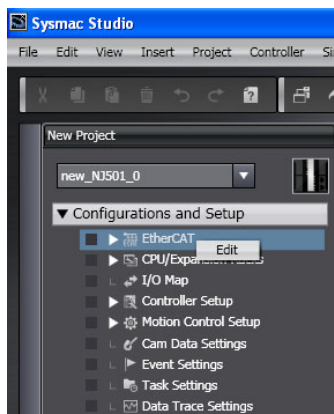
3 Right-click the EtherCAT master that is displayed in the Edit Pane and select **Compare and Merge with Actual Network Configuration**.

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

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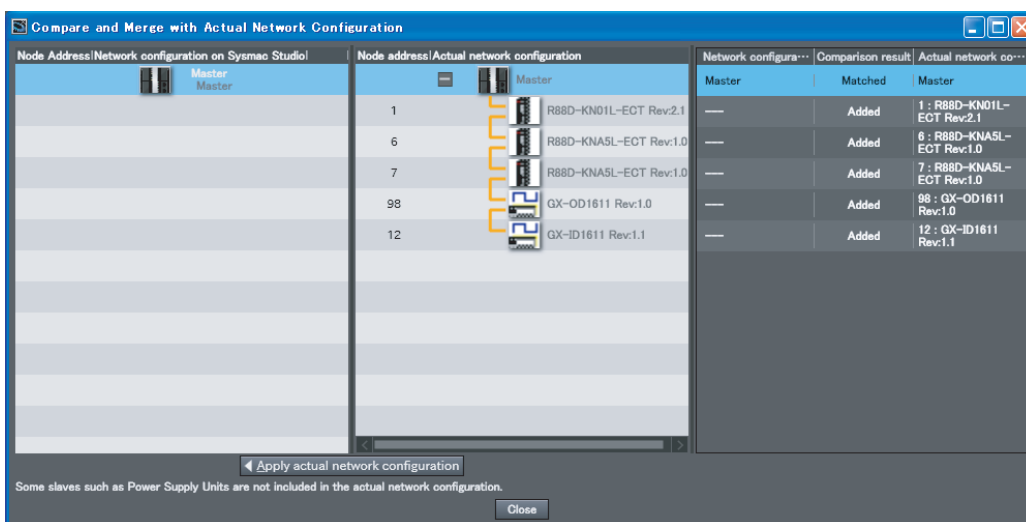
The actual network configuration is duplicated in the network configuration on Sysmac Studio, and the *Comparison results* Column shows that everything in the configurations matches. (The network configuration on the Sysmac Studio is created based on the actual network configuration.)

5 Click the **Close** Button.
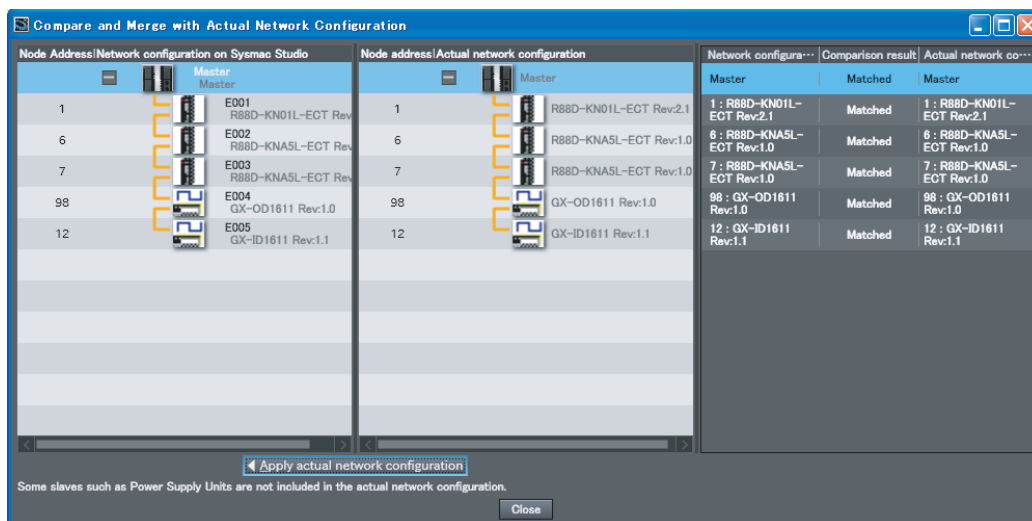
6 Go offline, and then assign device variables, set the axis parameters, set up the tasks, and set the master and slave parameters.

> **Additional Information**
>
> - Make sure that the communications cables between the master and slaves are connected correctly before you perform this operation. The Compare and Merge with Actual Network Configuration Dialog Box is not displayed if the connections are not correct.
>
> - The Compare and Merge with Actual Network Configuration Dialog Box is not displayed 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.
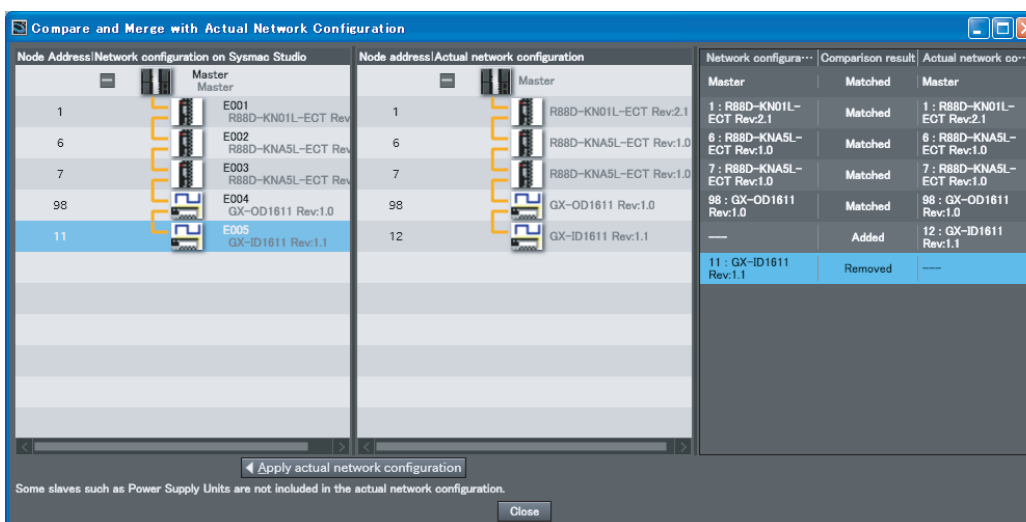>
> - The Compare and Merge with Actual Network Configuration Dialog Box is not displayed if an EtherCAT Frame Not Received error occurs. Remove the cause of the error and perform the compare and merge operation. Refer to the *NY-series Troubleshooting Manual* (Cat. No. W564) for the cause of the error.
>
> - 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 enable 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 master settings to control slaves that are registered in the task setup). 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 *5-5-1 Comparing and Merging with the Actual Network Configuration from the Sysmac Studio* for information on the operation to compare and merge with the actual network configuration on the Sysmac Studio.
>
> - When the compare and merge operation is performed, 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.

5-5-3   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 Controller. 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** in the Multiview Explorer. Or, right-click **EtherCAT** under **Configurations and Setup** and select *Edit*.
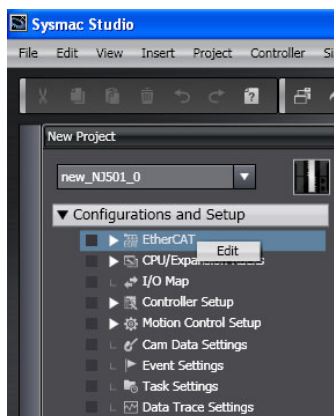
3 Right-click the EtherCAT master that is displayed in the Edit Pane and select *Get Slave Serial Numbers*.
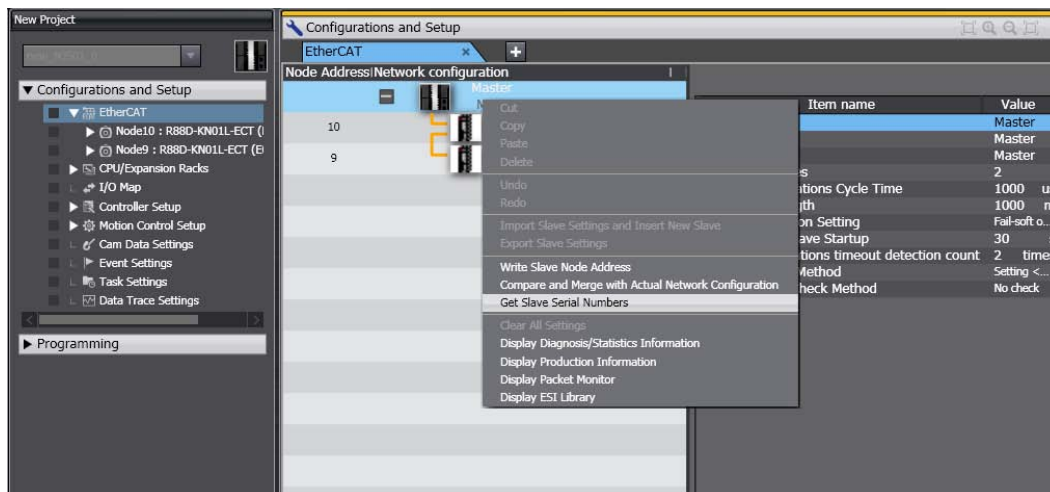
The serial numbers of the physical slaves are saved as the serial numbers in the slave settings on the Sysmac Studio.

📑 **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 if an EtherCAT Frame Not Received error occurs. Remove the cause of the error and perform the compare and merge operation. Refer to the *NY-series Troubleshooting Manual* (Cat. No. W564) for the cause of the error.

- 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.
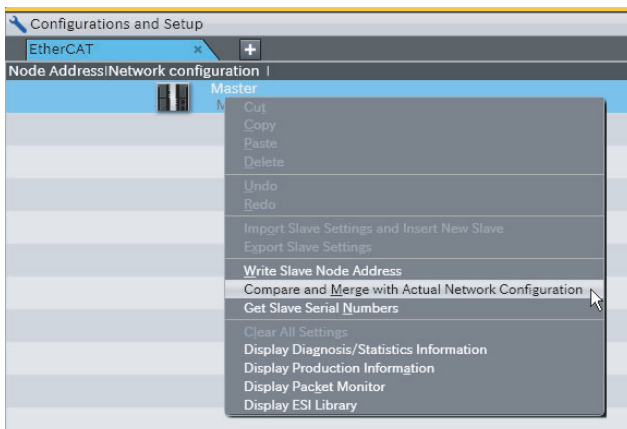
5   Setting Up EtherCAT Communications with the Sysmac Studio

5-6   Downloading the Network Configuration Information

5-6-1   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 NY-series Controller. 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 Controller 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 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.
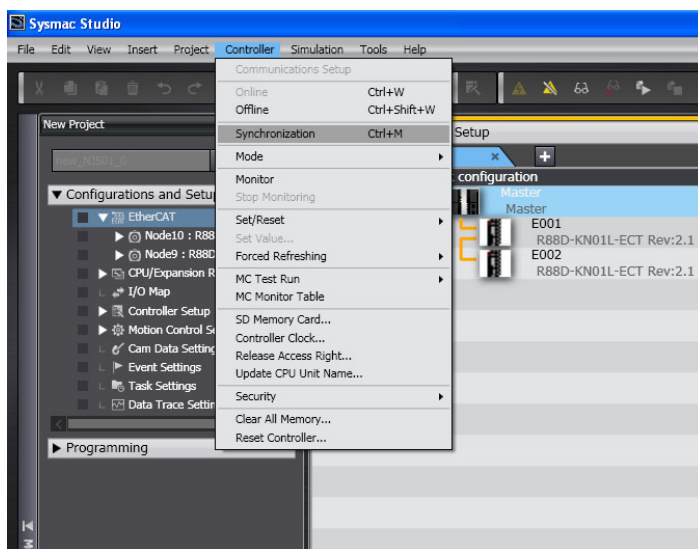
The Synchronization Dialog Box is displayed.

5-6 Downloading the Network Configuration Information

5

5-6-1 Downloading the Network Configuration Information from the Sysmac Studio

NY-series Industrial Panel PC / Industrial Box PC Built-in EtherCAT Port User's Manual (W562)

5-31

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

The network configuration information is downloaded 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 downloaded 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 Pane.

5-7 Confirming Communications after Completing EtherCAT Configuration and Settings

If normal network configuration information is downloaded to the NY-series Controller, EtherCAT communications start automatically regardless of the operating mode of the Controller.
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

The content of checking the process data communications status using these methods is given below.

● Controller Status Monitor of the Sysmac Studio

On the detailed view of the Controller status monitor, check the item *EtherCAT Process Data Communications*. If the item *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 the procedure 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.

If process data communications are not performed normally, check the event log in the Sysmac Studio and take suitable measures. Refer to the *NY-series Troubleshooting Manual* (Cat. No. W564) 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 *Section 10 Diagnostic and Statistical Information*.

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

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).

From the user program in the NY-series Controller, slave data is accessed through allocated variables.

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 NY-series Controller 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.)
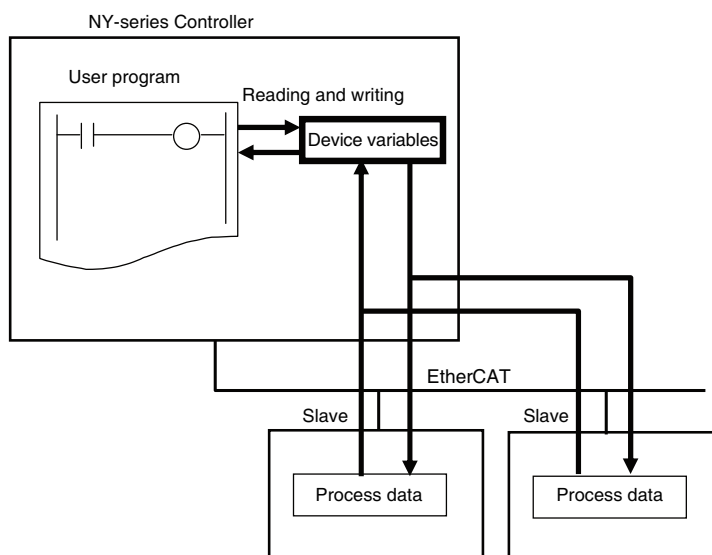
Axis Variables

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.
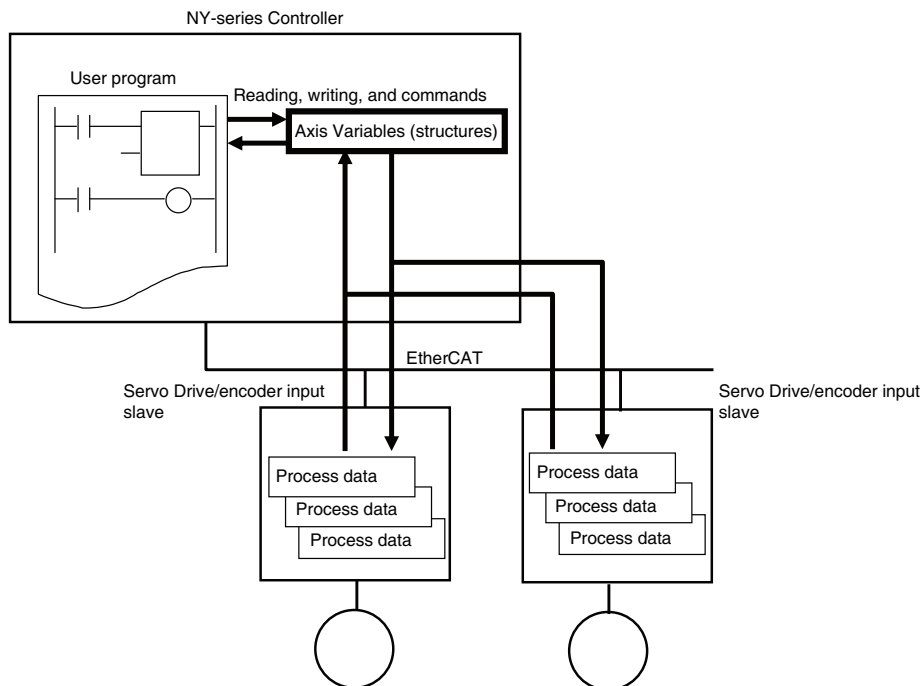
6-1-2   Process Data Communications Refresh Timing

The process data communications cycle is the same as the task period of the primary periodic task 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.
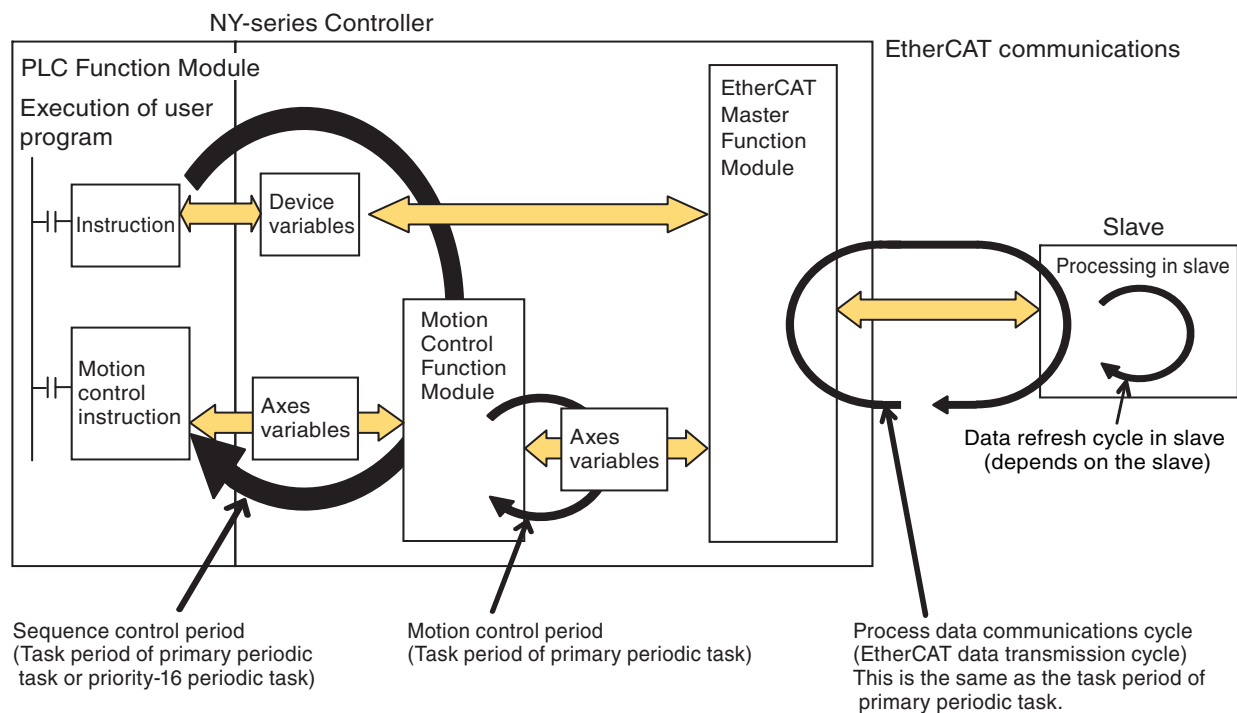
Task period of primary periodic task = Motion control period = Process data communications cycle

Relationship Between Process Data Communications Cycle and Task Setting

The process data communications cycle of the primary periodic task is 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	Slave Terminal
Communications Coupler Unit without an NX Unit assigned to an axis on the Slave Terminal	Primary periodic task or priority-16 periodic task	
EtherCAT slaves to which axes are assigned	Primary periodic task	Slave
Other EtherCAT slaves	Primary periodic task or priority-16 periodic task	

● **Sysmac Studio Setting Procedure**

For each slave and Unit without axis assignments, set the tasks to perform I/O refreshing in the **I/O Control Task Settings** on the **Task Settings** Tab Page of the Sysmac Studio.

For details, refer to the *NY-series Industrial Panel PC / Industrial Box PC Software User's Manual* (Cat. No. W558).

For each slave and Unit with axis assignments, select the motion control instruction to use in the **Motion Control Setup** of the Sysmac Studio. This sets the tasks to perform I/O refreshing.

For details, refer to the *NY-series Industrial Panel PC / Industrial Box PC Motion Control User's Manual* (Cat. No. W559).

● **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-16 periodic task	Task period of the primary periodic task[1][2]

*1 Refer to the manual for the specific 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 slaves that is set for synchronization.

*2 EtherCAT communications take place in the I/O refreshing timing for 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.

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

6-1-3   Checking the Validity of the 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 Controller 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 Ether-CAT commutations were established then.

Once the EtherCAT communications are established, the process data becomes valid, therefore values of the device variables and axes variables which 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 depend on the slave specifications. For details, refer to relevant manuals for each slave. If a noise countermeasures 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.

• Set the PDO communications timeout detection count setting in the EtherCAT master to 2 or higher. Refer to *5-4-1 Setting EtherCAT Master* 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 the Validity of the 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.

● LD
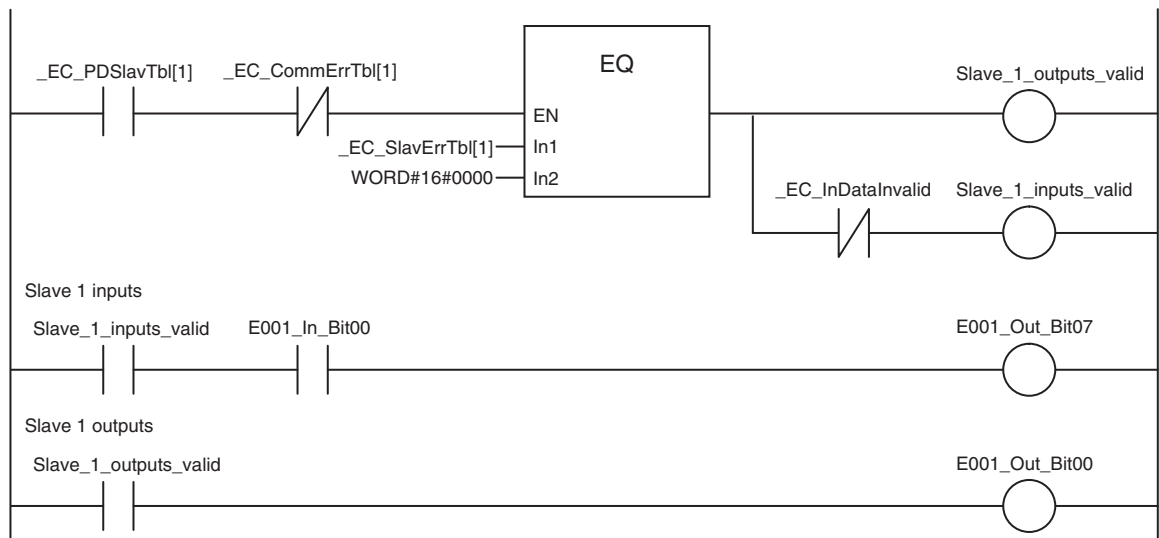
● ST

```
IF _EC_PDSlavTbl[1]=TRUE AND _EC_CommErrTbl[1]=FALSE AND
_EC_SlavErrTbl[1]=WORD#16#0 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 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 variables from the user program to see if I/O refreshing is normal.

Sample Programming to Check the Validity of the 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.

● **LD**
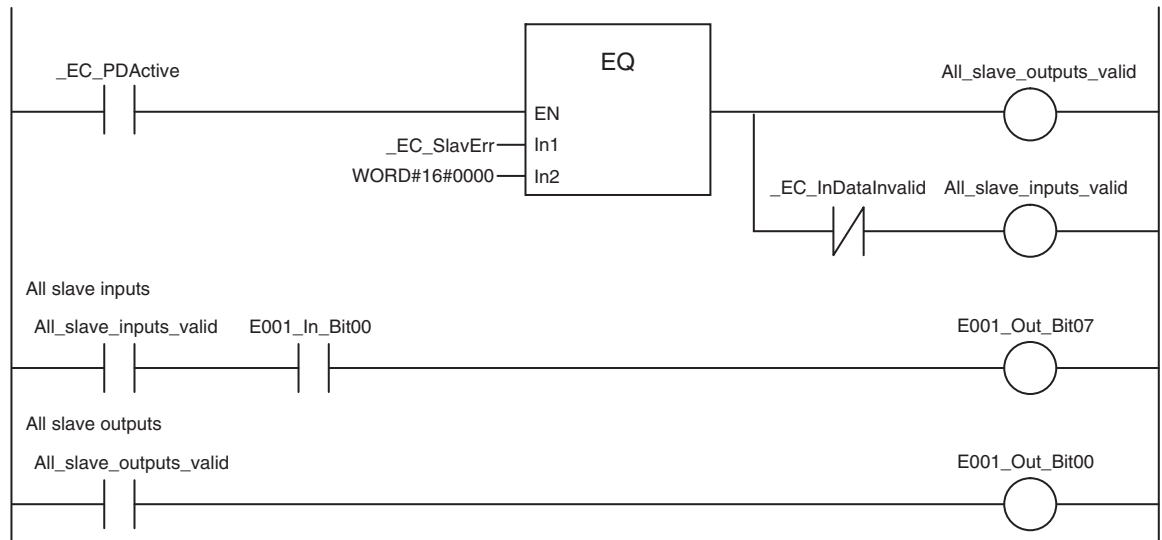
● **ST**

```
IF _EC_PDActive=TRUE AND _EC_SlavErr=WORD#16#0 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 InDataInvalid=FALSE THEN
   AllSlav_In:=TRUE;
  ELSE
   AllSlav_In:=FALSE;
  END_IF;

(* Output data valid *)
IF ALLSlav_Out=TRUE THEN
   E002_Out_Bit00:=TRUE;
  ELSE
   E002_Out_Bit00:=FALSE;
  END_IF;
```

6-1 Process Data Communications (PDO Communications)

6

6-1-3 Checking the Validity of the Process Data

```
(* Input data valid *)
IF ALLSlav_In=TRUE AND E002_In_Bit00=TRUE THEN
   E002_Out_Bit07:=TRUE;
  ELSE
   E002_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.

OI : I/O refresh

UPRG : User program execution

MC : Motion control
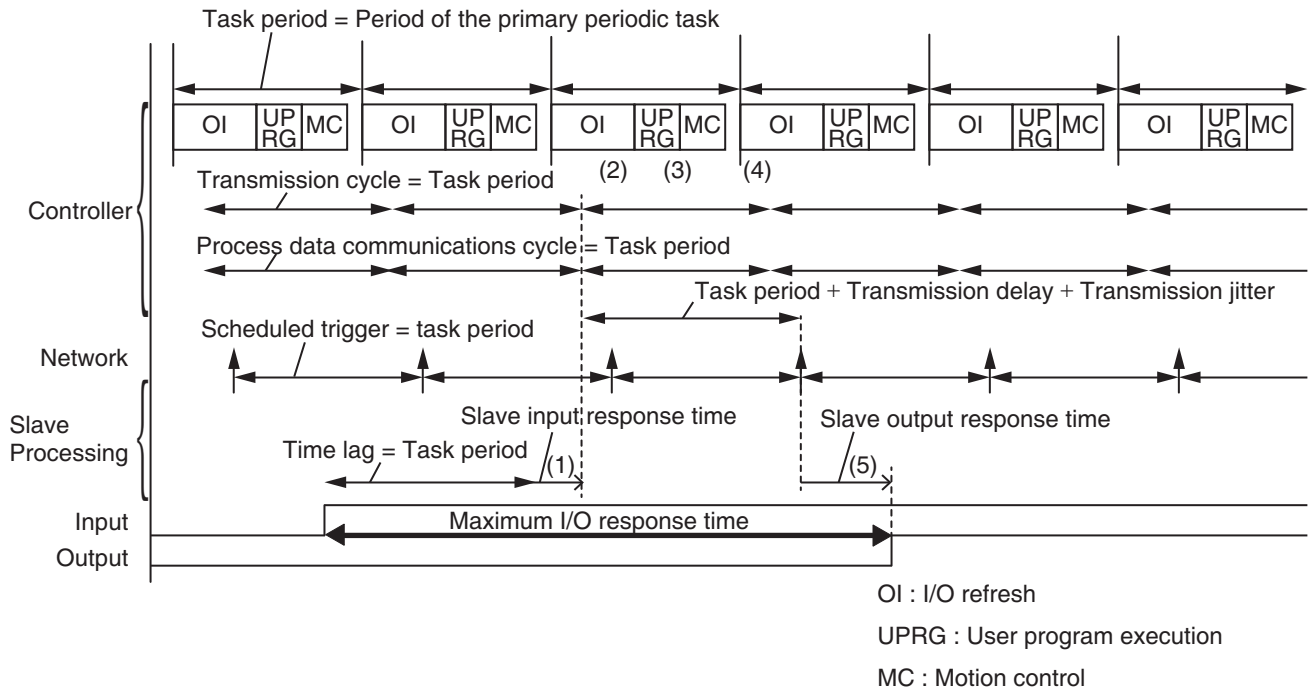
(1) The slave generates process data send frames with I/O information from sensors and other devices.

(2) The EtherCAT receives the process data frames that was sent from the slave through I/O refreshing for the Controller, and reads it as input data in the Controller.

(3) Based on the input data, the Controller 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 (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 periodic 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 Controller 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 =	Process data communication cycle (= Task period of the periodic task[*1]) × 2
	+ Slave input response time + Slave output response time
	+ Transmission delay + Transmission jitter

*1 The task period of the primary periodic task.

● **Transmission Delay**

This section defines the transmission delay for the EtherCAT master. The following methods are provided to calculate the transmission delay times for the Controller series and the periodic tasks in which I/O is refreshed.

Controller series	Periodic task of the Controller in which I/O is refreshed	Calculation method
NY-series	Primary periodic task	• Calculation with Sysmac Studio • 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. For details on this display method, refer to *Displaying the Transmission Delay Time for the EtherCAT Master with the Sysmac Studio* on page 6-13.

(2) Calculation by a formula

Transmission delay (μs) = 1.24 (μs) × number of EtherCAT slaves + 0.082 (μs) × EtherCAT frame length (bytes)

The parameters necessary for the 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

● **Transmission Jitter**

This section defines the transmission jitter for the EtherCAT master.

The value of transmission jitter varies depending on the Controller series.

Controller series	Transmission jitter
NY-series	10 μs

● **Slave Input Response Time**

This is the time from 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 the manuals for the relevant slaves.

● **Slave Output Response Time**

This is the time from 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 the manuals for the relevant slaves.

Displaying the Transmission Delay Time for the EtherCAT Master with the Sysmac Studio

You can use the Sysmac Studio to display the transmission delay time for the built-in EtherCAT port of the NY-series Controller. Use the following procedure to display the transmission delay time with the Sysmac Studio.

1 Right-click the EtherCAT master or EtherCAT slave in the EtherCAT Tab Page and select **Calculate Transmission Delay Time of the Master**.
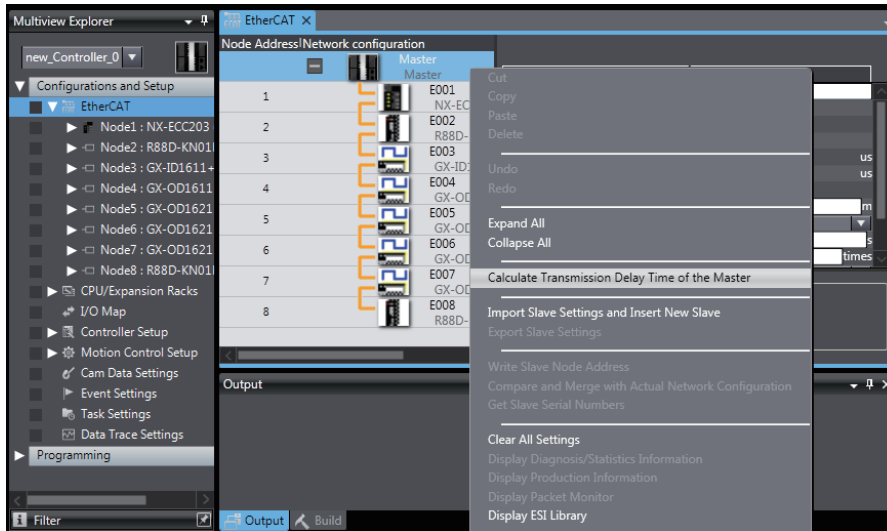
2 The transmission delay times for PDO communications cycle 1 and PDO communications cycle 2 are displayed in the Output Tab Page.
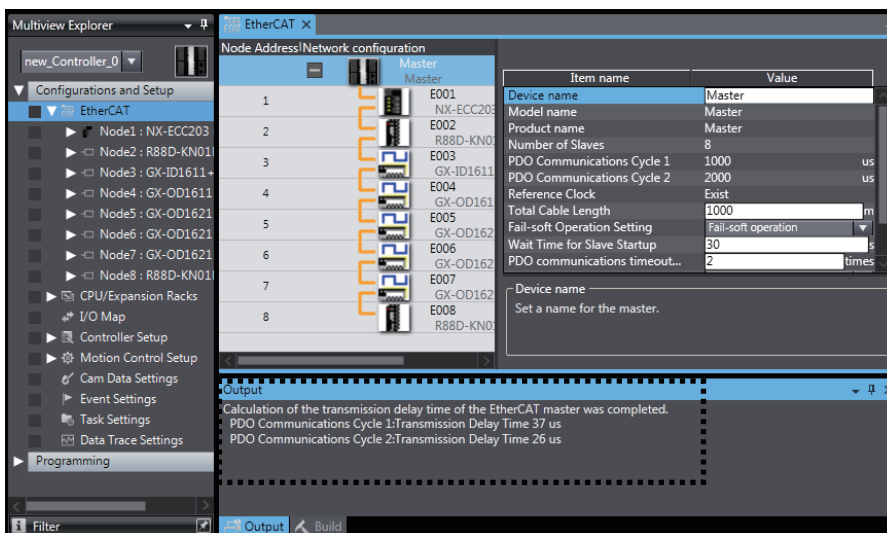
The transmission delay times of the EtherCAT master depend on the periodic tasks of the Controller that refresh I/O with the EtherCAT slaves, as shown below.

Periodic task of the Controller in which I/O is refreshed	Applicable transmission delay time of the EtherCAT master
Primary periodic task	Transmission delay time for PDO communications cycle 1

> **Precautions for Correct Use**
>
> 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.

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

I/O Operations for Major Fault Level Controller Errors

If the NY-series Controller 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 move to Safe-Operational state and the slave settings will control the slave outputs. The inputs from the slaves will retain the process data values from before operation stopped. Message communications are continued.[*]

NY-series Controller

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 settings will control the slave outputs.
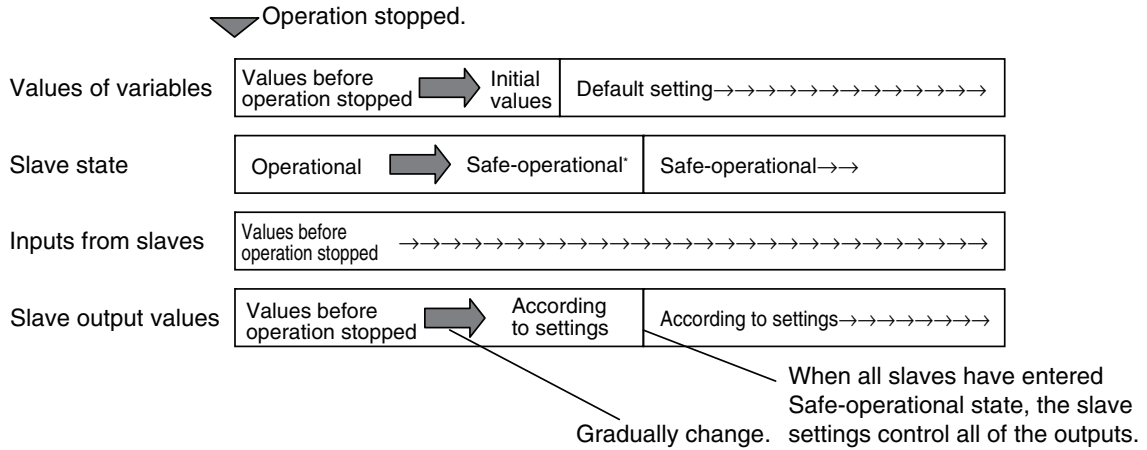
* The following errors may occur when the slave enters to the Safe-Operational state.
  • Slave Application Error
  • Slave Initialization Error

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

Controller	Formula	Maximum time
NY-series	PDO communications timeout detection count × Task period of the task assigned to slave[*1] + 10 ms	The result given from the left formula. The maximum time is 100 to 110 ms if the calculation result is 110 ms or less.

*1 If multiple tasks are assigned, the longest task period is used for calculation.

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.

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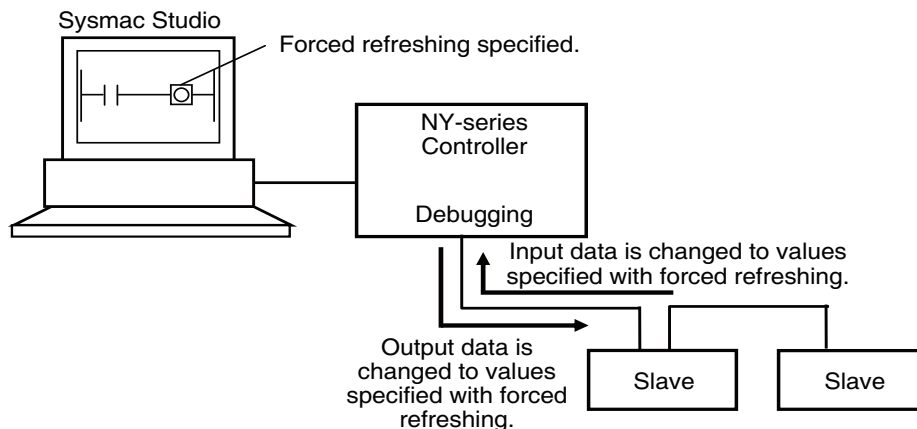
Relationship between Major Fault Level Controller Errors, Normal Operation, and Forced Refreshing

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

* If a major fault level Controller error occurs, the output values from before operation stopped are retained while the slaves are in Operational state and the slave settings control the outputs after the slaves enter Safe-operational state.

Precautions for Safe Use

- You can select whether the master continues or stops communications with all slaves when a communications error occurs. Refer to *5-4-1 Setting EtherCAT Master* 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 countermeasures 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 the Validity of the Process Data*.
  - Set the PDO communications timeout detection count setting in the EtherCAT master to at least 2. Refer to *5-4-1 Setting EtherCAT Master* for the setting procedure.
- If a communications error prevents the slaves from receiving signals from the master, the slave settings will control the slave outputs. During the time that is required 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 slave settings will control the slave outputs for synced slaves. The previous values are retained for the slave outputs for non-synced slaves.

	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 check from the user program to see if I/O refreshing is normal by checking one of the system-defined variables *_EC_PDSlavTbl* (Process Data Communicating Slave Table).

6-2   SDO Communications

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

6-2-1   EtherCAT Instructions

You can perform the following SDO communications with EtherCAT instructions.

• Reading and writing of 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*.
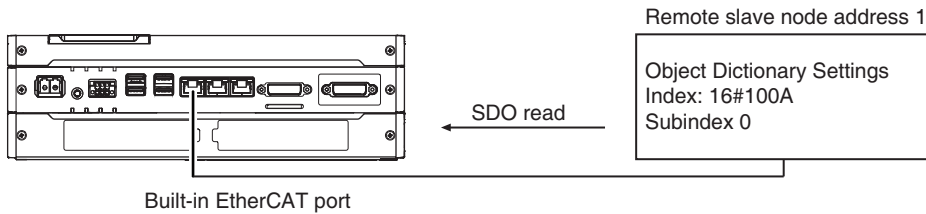
Remote slave node address 1

Object Dictionary Settings
Index: 16#100A
Subindex 0

SDO read

Built-in EtherCAT port

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	☑	Message Communications Enabled Slave Table

Accept trigger.

```
Trigger    _EC_MBXSlavTbl[1]
  │──┤├──────────┤├──────────┐      (@)MOVE
  │                          │     EN      ENO
  │                    UINT#16#100A ─ In    Out ─ SdoObject.Index
  │
  │                                 (@)MOVE
  │                                EN      ENO
  │                        USINT#0 ─ In    Out ─ SdoObject.Subindex
  │
  │                                 (@)MOVE
  │                                EN      ENO
  │                          FALSE ─ In    Out ─ SdoObject.IsComleteAccess
  │
  │                          EC_CoESDORead_instance
  │                              EC_CoESDORead
  └─────────────────────── Execute         Done ──
                     UINT#1 ─ NodeAdr       Busy ──
                  SdoObject ─ SdoObj       Error ──
                    UINT#20 ─ TimeOut    ErrorID ──
                              ReadDat
                VersionInfo ─ ─ ─ ─ ─ ─ ─ ─ ─ ─
                                       AbortCode ──
                                        ReadSize ──
```

Processing after normal end

```
Trigger    EC_CoESDORead_instance.Done       Inline ST
  │──┤├──────────┤├──────────              1 // Processing after normal end
  │                                        2 ;
```

Processing after error end

```
Trigger    EC_CoESDORead_instance.Error      Inline ST
  │──┤├──────────┤├──────────              1 // Processing after error end
  │                                        2 ;
```

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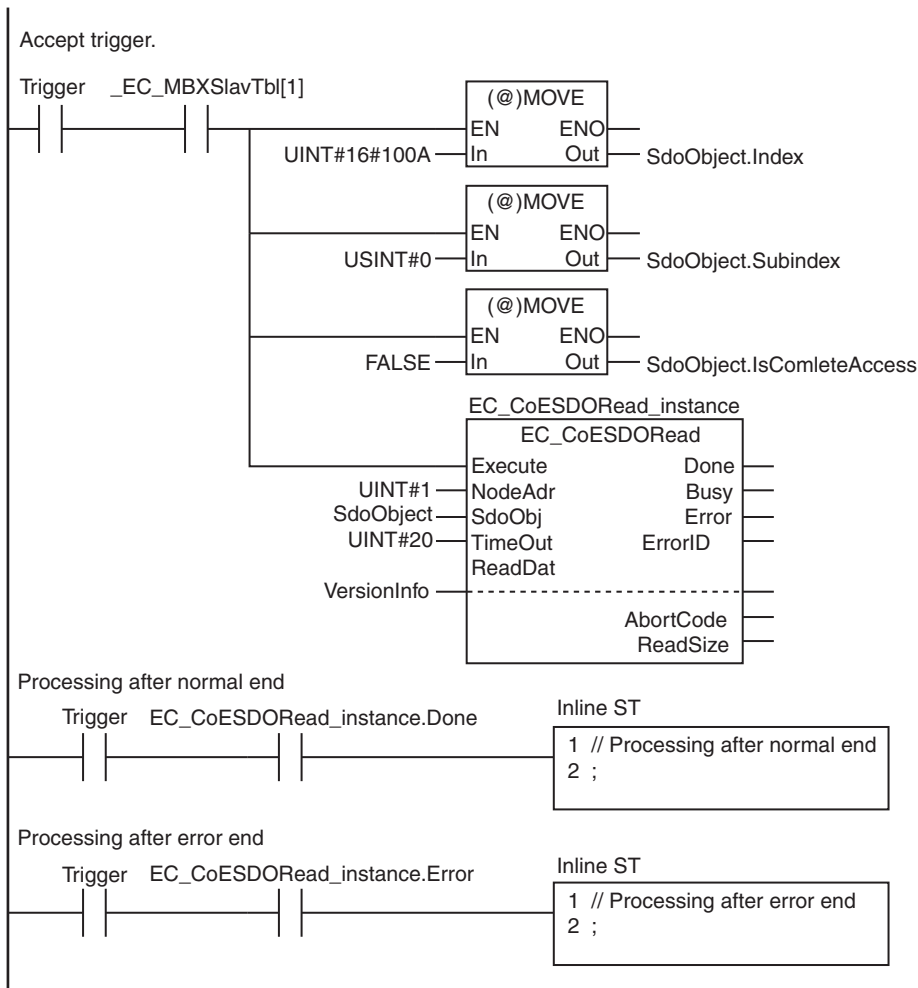
Internal Variables	Variable	Data type	Initial value	Comment
	Trigger	BOOL	False	Execution condition
	SdoObject	_sSDO_ACCESS	(Index:=0, Subindex:=0, IsCompleteAccess:=False)	SDO parameter
	DoSdoRead	BOOL	False	Processing
	VersionInfo	STRING[256]	''	Read data
	NormalEnd	UINT	0	Normal end
	ErrorEnd	UINT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

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

```
// 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 NY-series Controller. System services are executed in the period between execution of all of the tasks.

Refer to the *NY-series Industrial Panel PC / Industrial Box PC Software User's Manual* (Cat. No. W558) 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.

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 the manual for the 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 primary periodic task	Transmission size
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
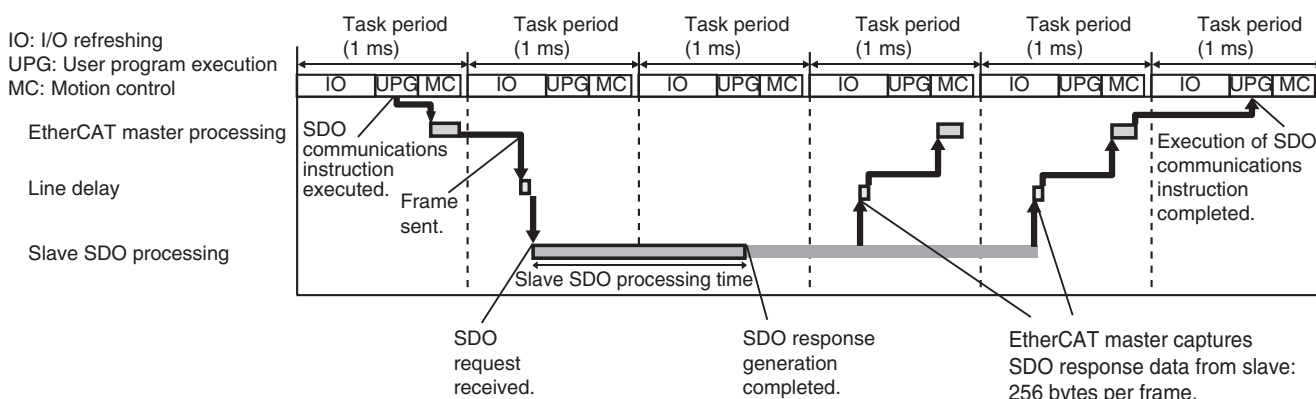
● When the Task Period of the Primary Periodic Task is 4 ms or Longer

Minimum message response time (ms) =   Task period of the task that executes SDO communications instruction

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

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

*1   The slave SDO processing time depends on the slaves. Refer to the manual for the 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
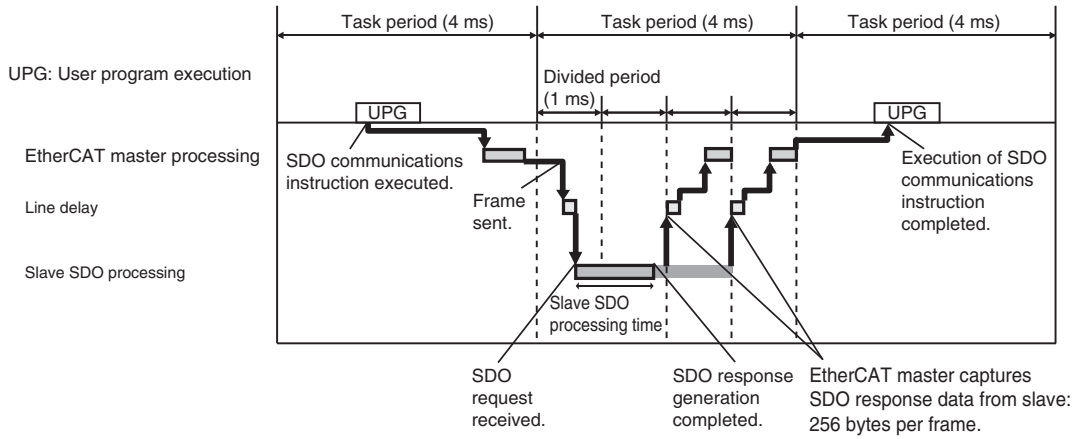
6-3 Instructions Used in EtherCAT Communications

6-3-1 EtherCAT Instructions

Function	Instruction	Description
Start EtherCAT Packet Monitor	EC_StartMon	Starts packet monitoring of the EtherCAT master built into the NY-series Controller.
Stop EtherCAT Packet Monitor	EC_StopMon	Stops packet monitoring of the EtherCAT master built into the NY-series Controller.
Save Packet Data File	EC_SaveMon	Saves captured data in a file in the system of the NY-series Controller. The packet monitor is a function of the EtherCAT master built into the NY-series Controller. The captured files in the system of the Controller are not retained when the power is interrupted.
Copy Packet Data File to SD Memory Card	EC_CopyMon	Copies the captured data that was saved in the system of the NY-series Controller to a file on a Virtual SD Memory Card. The captured data is previously saved from EtherCAT master built into the NY-series Controller into the system of the NY-series Controller. The files saved to a Virtual 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.)

Note Refer to the *NY-series Instructions Reference Manual* (Cat. No. W560) for details on the instructions that are used with EtherCAT communications.

📝 **Additional Information**

"Enable/disable," "Connected/disconnected," and "physical slave/no physical slave" are defined as follows:

- Enable/Disable Setting
  This setting tells whether process data communications are performed for an EtherCAT slave. You can make this setting from the Sysmac Studio and you can change the setting with the EC_ChangeEnableSetting (Enable/Disable EtherCAT Slave) instruction.

- Connected/Disconnected Setting
  This setting is used to temporarily disconnect a slave from the network when the slave is replaced and then to connect the slave to the network again.
  However, if an EtherCAT slave in a daisy-chain connection is disconnected, all of the slaves that are connected after it are disconnected. This point is different from the enable/disable setting.

- Physical Slave/No Physical Slave
  This indicates whether the EtherCAT slave is physically connected.

The following table shows how these states combine.

Enable/disable setting	Connected/discon-nected setting	Physical slave/no physical slave	Combined state
Enabled	Connected	Present	Present
		Not installed	None (You cannot connect a non-exiting slave to the network.)
	Disconnected	Present	Present
		Not installed	
Disabled	--- (The concept of connected/disconnected does not apply if a slave is disabled.)	Present	Present
		Not installed	

7

System-defined Variables That Are Related to the Built-in EtherCAT Port

This section describes the system-defined variables that are related to the built-in EtherCAT port.

7

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.
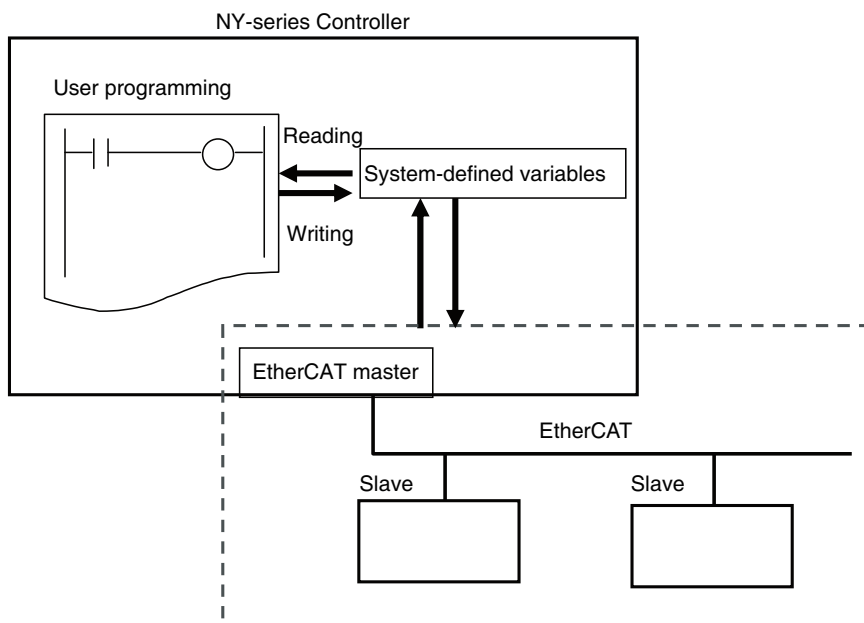
7-1-2   System-defined Variables

● Functional Classification: EtherCAT Communications Errors

Variable name	Meaning	Function	Data type	Range of values	Reference
_EC_ErrSta	Built-in 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* given later for the meaning of individual bits.	WORD	16#0000 to 16#40F0	page 7-8
_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* given later for the meaning of individual bits.	WORD	16#0000 to 16#00F0	page 7-8
_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* given later for the meaning of individual bits.	WORD	16#0000 to 16#00F0	page 7-8

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* given later for the meaning of individual bits.	WORD	16#0000 to 16#00F0	page 7-8
_EC_SlavErrTbl	Slave Error Table	This system-defined variable gives the error status for each EtherCAT slave. The error status is given for 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). Refer to *Meanings of Error Status Bits* given later for the meaning of individual bits.	ARRAY [1..512] OF WORD	16#0000 to 16#00F0	page 7-9
_EC_MacAdrErr	MAC Address Error	TRUE if there is an illegal MAC address.	BOOL	TRUE or FALSE	page 7-9
_EC_LanHwErr	Communications Controller Error	TRUE if there is a communications controller hardware error.	BOOL	TRUE or FALSE	page 7-9
_EC_LinkOffErr	Link OFF Error	TRUE if the communications controller link is not established.	BOOL	TRUE or FALSE	page 7-9
_EC_NetCfgErr	Network Configuration Information Error	TRUE if there is illegal network configuration information.	BOOL	TRUE or FALSE	page 7-9
_EC_NetCfgCmpErr	Network Configuration Verification Error	TRUE if the network configuration information does not match the actual network configuration.	BOOL	TRUE or FALSE	page 7-10
_EC_NetTopologyErr	Network Configuration Error	TRUE if there is a network configuration error (too many devices connected or ring connection).	BOOL	TRUE or FALSE	page 7-10
_EC_PDCommErr	Process Data Communications Error	TRUE if there is an unexpected slave disconnection or connection or if a slave WDT error is detected during process data communications.	BOOL	TRUE or FALSE	page 7-10
_EC_PDTimeoutErr	Process Data Reception Time-out Error	TRUE if a timeout occurs while receiving process data.	BOOL	TRUE or FALSE	page 7-10
_EC_PDSendErr	Process Data Transmission Error	TRUE if there is a process data transmission error (cannot send within the process data communications cycle or transmission jitter is over the limit).	BOOL	TRUE or FALSE	page 7-10
_EC_SlavAdrDupErr	Slave Node Address Duplicated Error	TRUE if the same node address is set for more than one slave.	BOOL	TRUE or FALSE	page 7-10
_EC_SlavInitErr	Slave Initialization Error	TRUE if there is an error in an initialization command addressed to a slave.	BOOL	TRUE or FALSE	page 7-11
_EC_SlavAppErr	Slave Application Error	TRUE if there is an error in the slave's application status register.	BOOL	TRUE or FALSE	page 7-11
_EC_MsgErr	EtherCAT Message Error	TRUE when a message is sent to a slave that does not support messages or when there is an error in the format of the response to a message that was sent to a slave.	BOOL	TRUE or FALSE	page 7-11
_EC_SlavEmergErr	Emergency Message Detected	TRUE if the master detects an emergency message that was sent by a slave.	BOOL	TRUE or FALSE	page 7-11
_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	TRUE or FALSE	page 7-11
_EC_CycleExceeded	EtherCAT Communications Cycle Exceeded	TRUE if the Controller cannot establish communications within the set communications period at startup.	BOOL	TRUE or FALSE	page 7-12

📝 **Additional Information**

Typical Relationships for the Built-in EtherCAT Error Flags

Variable Name	Meaning	Variable Name	Meaning	Variable Name	Meaning	Event level
_EC_ErrSta	Built-in 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	Minor fault level
		_EC_MstrErr	Master Error	_EC_NetCfgErr	Network Configuration Information Error	
				_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	TRUE or FALSE	page 7-12
_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	TRUE or FALSE	page 7-12
_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	TRUE or FALSE	page 7-12
_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	TRUE or FALSE	page 7-13
_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	TRUE or FALSE	page 7-13
_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	TRUE or FALSE	page 7-13
_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 7-13
_EC_PktMonStop	Packet Monitoring Stopped	TRUE when packet monitoring is stopped.	BOOL	TRUE or FALSE	page 7-14
_EC_LinkStatus	Link Status	TRUE if the communications controller link status is Link ON.	BOOL	TRUE or FALSE	page 7-14

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 7-14
_EC_InDataInvalid	Input Data Invalid	TRUE when process data communications in the primary periodic task are not normal and the input data is not valid.	BOOL	TRUE or FALSE	page 7-14
_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.	BOOL	TRUE or FALSE	page 7-14

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_StatisticsLogEn-able	Diagnosis/Statis-tics Log Enable	Changes to TRUE when the diagnosis/statis-tics log is started. Changes to FALSE when the diagnosis/statis-tics log is ended.	BOOL	TRUE or FALSE	page 7-15
_EC_StatisticsLogCy-cleSec	Diagnosis/Statis-tics Log Cycle	Specifies the interval to write the diagnostic and statistical information of the diagnosis/sta-tistics log in units of seconds. When 0 is specified, the diagnostic and statis-tical 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 diag-nosis/statistics log operation is in prog-ress.	UINT	0, or 30 to 1800	page 7-15
_EC_StatisticsLog-Busy	Diagnosis/Statis-tics Log Busy	TRUE while the diagnosis/statistics log opera-tion is in progress.	BOOL	TRUE or FALSE	page 7-15
_EC_StatisticsLogErr	Diagnosis/Statis-tics 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 diag-nosis/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 Virtual SD Memory Card is fully used. • There is no Virtual SD Memory Card. • The function cannot be started because the value specified for *_EC_StatisticsLogCy-cleSec* (Diagnosis/Statistics Log Cycle) is invalid.	BOOL	TRUE or FALSE	page 7-15

● **Meanings of Error Status Bits**

The meanings of the individual bits in the above error status variables are given below.

Bit	Name	Description	Value	Meaning
15	Master Detection	This bit indicates whether the master detected an error in the slaves that it manages.	TRUE	Error
			FALSE	No error
14	Slave Summary[1]	Indicates whether there is an error at a level below the function module.	TRUE	Error
			FALSE	No error
8 to 13	Not used.			
7	Major Fault	Indicates if there is a major fault level error.	TRUE	Error
			FALSE	No error
6	Partial Fault	Indicates if there is a partial fault level error.	TRUE	Error
			FALSE	No error
5	Minor Fault	Indicates if there is a minor fault level error.	TRUE	Error
			FALSE	No error
4	Observation	Indicates if there is an observation level error.	TRUE	Error
			FALSE	No error
0 to 3	Not used.			

*1   For the EtherCAT Master Function Module, only _EC_ErrSta (Built-in EtherCAT Error) is used.

7-1-3   EtherCAT Master Function Module, Category Name: _EC

● **Functional Classification: EtherCAT Communications Errors**

Variable name	_EC_ErrSta				
Meaning	Built-in 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* provided above 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* provided above 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* provided above 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* provided above 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. The error status is given for 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). Refer to *Meanings of Error Status Bits* provided above for the meanings of the error status bits.				
Data type	Array [1..512] OF 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_MacAdrErr				
Meaning	MAC Address Error			Global/local	Global
Function	TRUE if there is an illegal MAC address.				
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_LanHwErr				
Meaning	Communications Controller Error			Global/local	Global
Function	TRUE if there is a communications controller hardware error.				
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_LinkOffErr				
Meaning	Link OFF Error			Global/local	Global
Function	TRUE if the communications controller link is not established.				
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 there is illegal network configuration information.				
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_NetCfgCmpErr				
Meaning	Network Configuration Verification Error		Global/local	Global	
Function	TRUE if the network configuration information does not match the actual network configuration.				
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 there is a network configuration error (too many devices connected or ring connection).				
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 there is an unexpected slave disconnection or connection or if a slave WDT error is detected during process data communications.				
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 a timeout occurs while receiving process data.				
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 there is a process data transmission error (cannot send within the process data communications period or transmission jitter is over the limit).				
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 same node address is set for more than one slave.				
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 there is an error in an initialization command addressed to a slave.				
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 there is an error in the slave's application status register.				
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 when a message is sent to a slave that does not support messages or when there is an error in the format of the response to a message that was sent to a slave.				
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 CoE SDO) • EC_CoESDORead CoE messages (Write CoE SDO) • EC_CoESDOWrite		

Variable name	_EC_SlavEmergErr				
Meaning	Emergency Message Detected			Global/local	Global
Function	TRUE if the master detects an emergency message that was sent by a slave.				
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_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			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		

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 Controller cannot establish communications within the set communications period at startup.				
Data types	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			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_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			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_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			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		

Variable name	_EC_PDSlavTbl				
Meaning	Process Data Communicating Slave Table			Global/local	Global
Function	This is a table that 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			Range of values	TRUE or FALSE
R/W access	R	Retained	Not retained.	Network Publish	Published.
Usage in user program	Possible.	Related instruc-tions	Disconnect EtherCAT Slave • EC_DisconnectSlave Connect EtherCAT Slave • EC_ConnectSlave		

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			Range of values	TRUE or FALSE
R/W access	R	Retained	Not retained.	Network Publish	Published.
Usage in user program	Possible.	Related instruc-tions	Disconnect EtherCAT Slave • EC_DisconnectSlave Connect EtherCAT Slave • EC_ConnectSlave		

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			Range of values	TRUE or FALSE
R/W access	R	Retained	Not retained.	Network Publish	Published.
Usage in user program	Possible.	Related instruc-tions	---		

Variable name	_EC_PDActive				
Meaning	Process Data Communications Status			Global/local	Global
Function	TRUE when process data communications are performed with all slaves*.				
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 instruc-tions	Disconnect EtherCAT Slave • EC_DisconnectSlave Connect EtherCAT Slave • EC_ConnectSlave		

* 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 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 All system-defined variables that are related to the status of EtherCAT communications give the current status.

Variable name	_EC_InData1Invalid				
Meaning	Input Data1 Invalid			Global/local	Global
Function	TRUE when process data communications established in the primary periodic period 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 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	_EC_StatisticsLogEnable		
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	R/W	Retained	Not retained.	Network Publish	Published.
Usage in user program	Possible.	Related instructions	---

Variable name	_EC_StatisticsLogCycleSec		
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	R/W	Retained	Not retained.	Network Publish	Published.
Usage in user program	Possible.	Related instructions	---

Variable name	_EC_StatisticsLogBusy		
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	---

Variable name	_EC_StatisticsLogErr		
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 Virtual SD Memory Card is fully used. • There is no Virtual 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	---

7-1 System-defined Variables That Are Related to the Built-in EtherCAT Port

7

7-1-3 EtherCAT Master Function Module, Category Name: _EC

8

Example of Operations for EtherCAT Communications

This section provides a series of example operations for when an NY-series Controller is connected to slaves.

8

8-1   Example of Operations for EtherCAT Communications

8-1-1   System Configuration Example
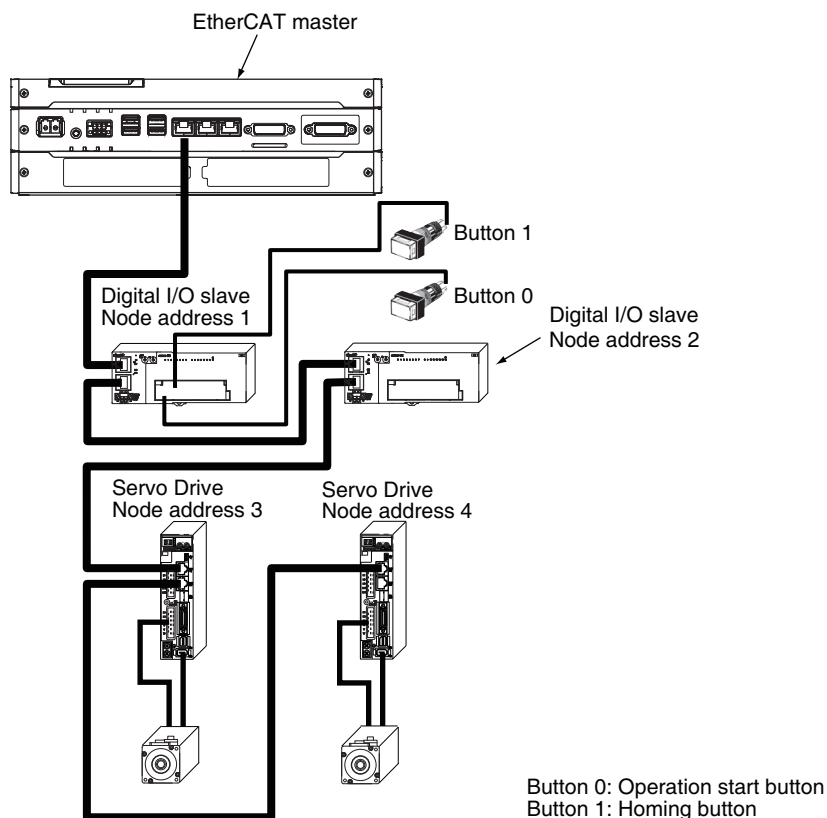
There are no restrictions on the order of node addresses.

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.
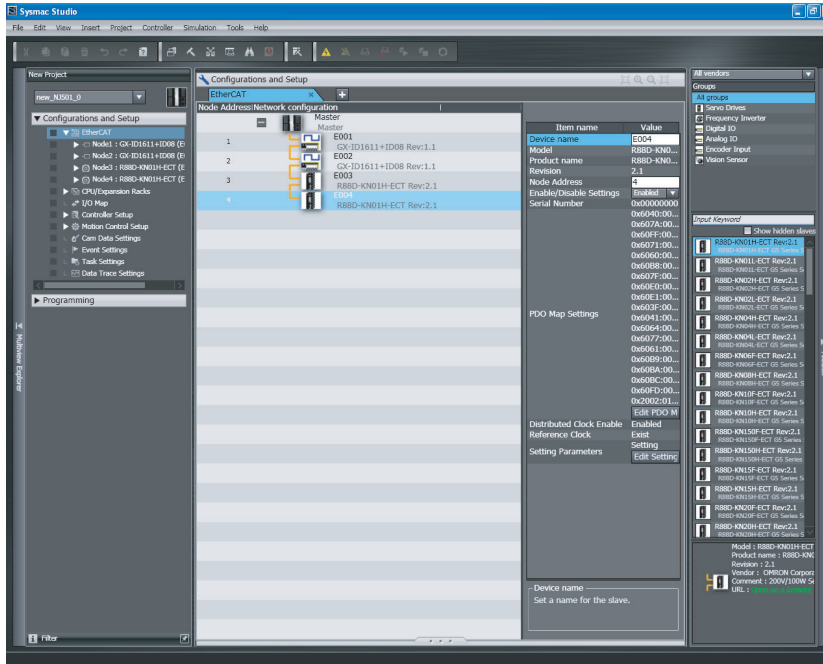
In this example, digital I/O slaves are set to node addresses 1 and 2 and Servo Drives are set to node addresses 3 and 4.

● 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.
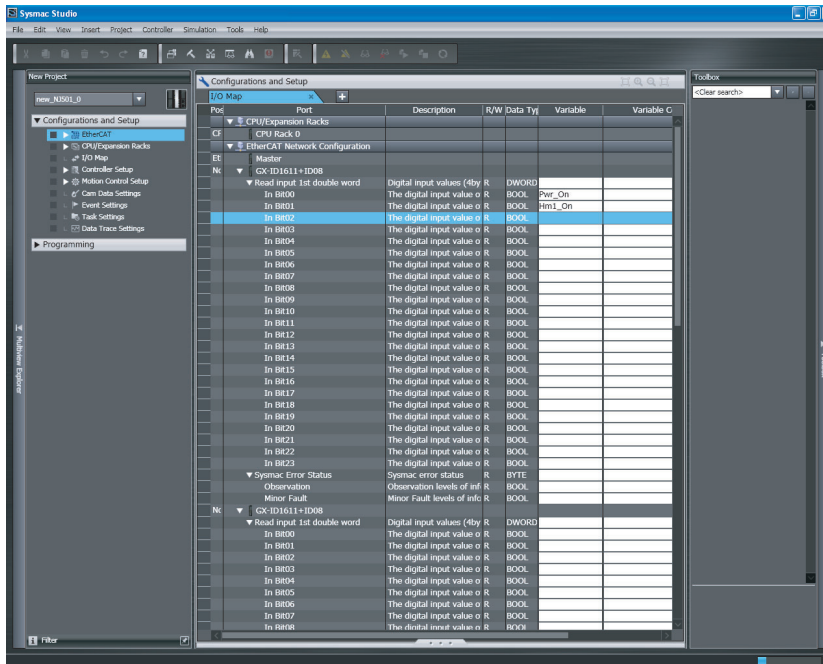
In this example, *Pwr_On* is assigned to bit 0 of slave 1 and *Hm1_On* is assigned to bit 1 of slave 1.

● **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.
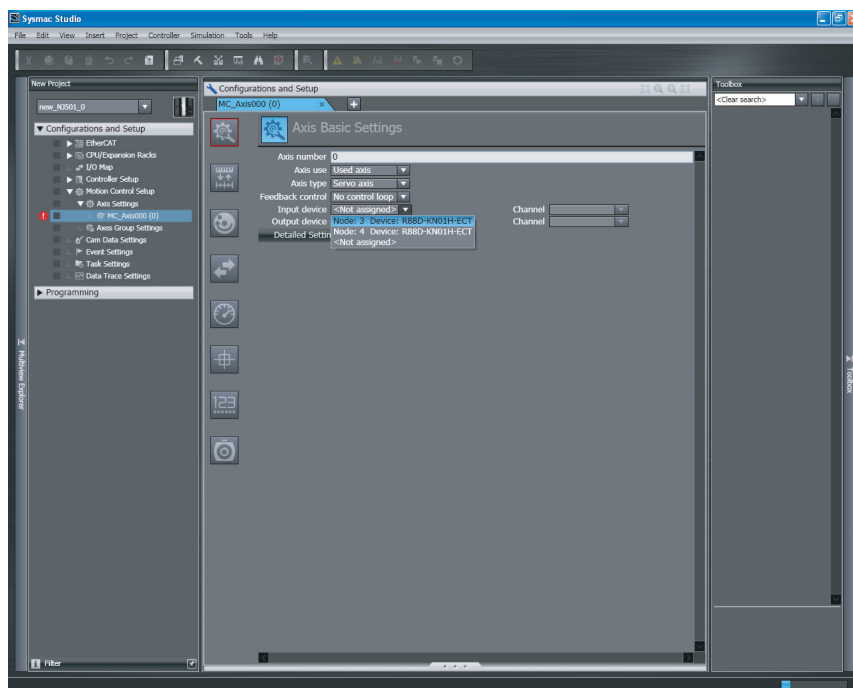
In the same way, add an axis and assign the Servo Drive with node address 4 to it.

● **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 settings will control the slave outputs.

● **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.)
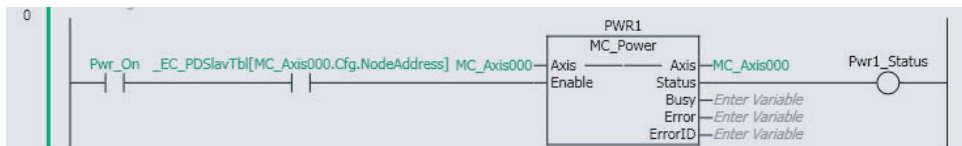
When the homing button is pressed, the *Hm1_On* variable changes to TRUE and homing is performed. (See rung 1.)
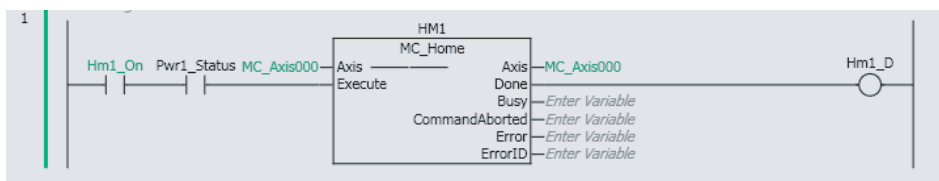
● Assigning Programs to Tasks

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.

Refer to *5-7 Confirming Communications after Completing EtherCAT Configuration and Settings* for the check procedure.

9

Replacing Slaves during Communications

This section describes the procedure and precautions for replacing slaves during communications.

9-1   Precautions When Connecting or Disconnecting Slaves during Communications

9-1-1   Procedure for Connecting and Disconnecting Slaves during Communications

Always use the following procedure to turn OFF the slave power supply or connect/disconnect cables during EtherCAT master communications.*

* This includes the pre-operational, 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.

9-1-2   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 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.*

* This can happen when the Reset EtherCAT Error (ResetECError) instruction or Connect EtherCAT Slave (EC_-ConnectSlave) 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 unexpected 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.

9-2   Replacing Slaves during Communications

9-2-1   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.
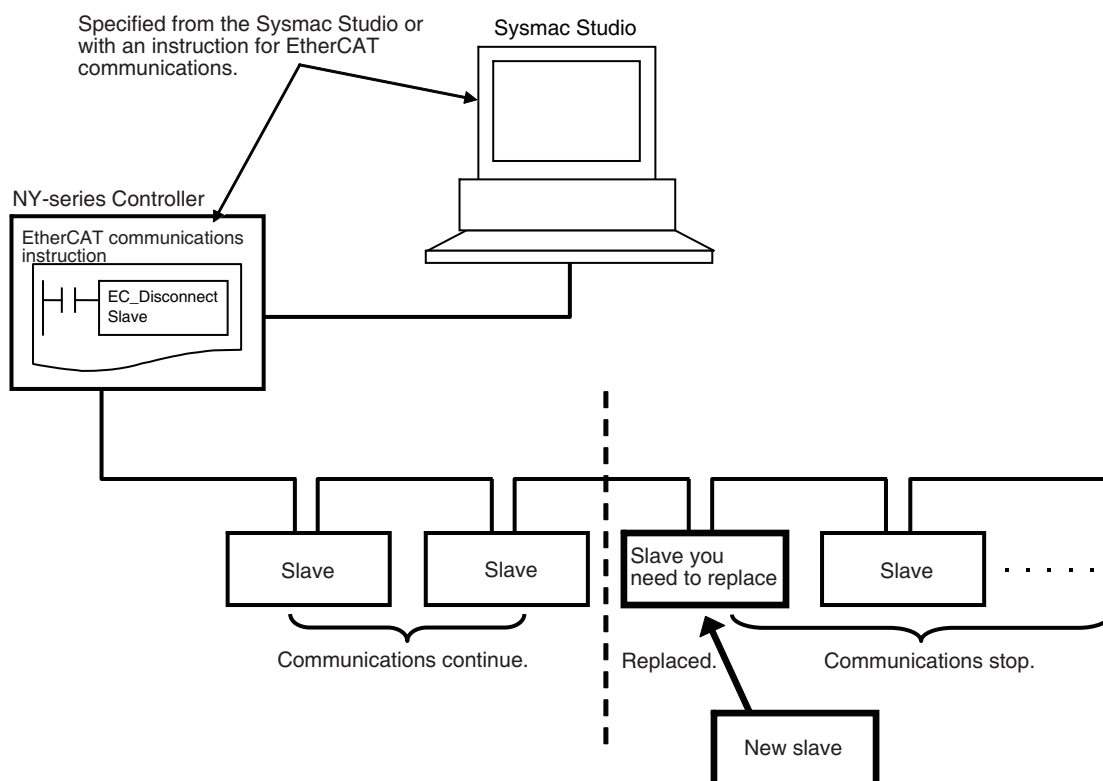
Precautions for Correct Use

- A Process Data Communications Error occurs if you physically disconnect a slave from the network without executing the disconnection instruction.
- A Network Configuration Verification Error occurs if the correct node address is not set for a slave that was replaced with the disconnection instruction and a connection is made to a different port than the one that was used for the disconnection instruction.

9-2-2   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: Execute 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 backing up the parameters in the EtherCAT slave from the Sysmac Studio.)
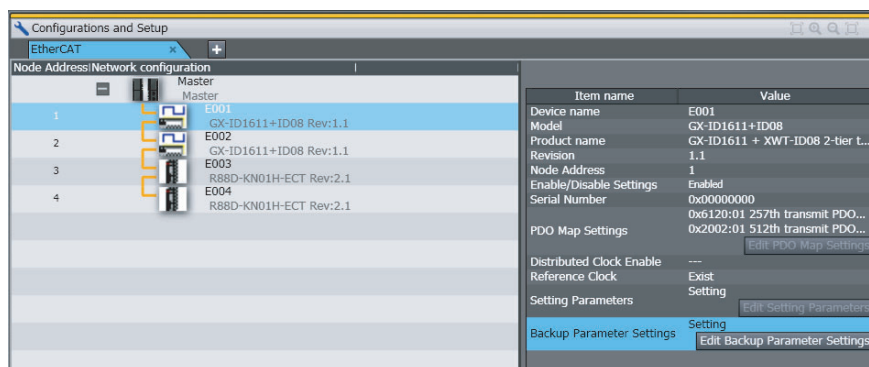
📖 **Additional Information**

The disconnection and reconnection operations are not required for slaves that are disabled with the EtherCAT Slave Enable/Disable Settings. You therefore cannot disconnect and reconnect them from the Sysmac Studio. If you execute the disconnection and connection instructions, they will result in errors.

9-2-3   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.
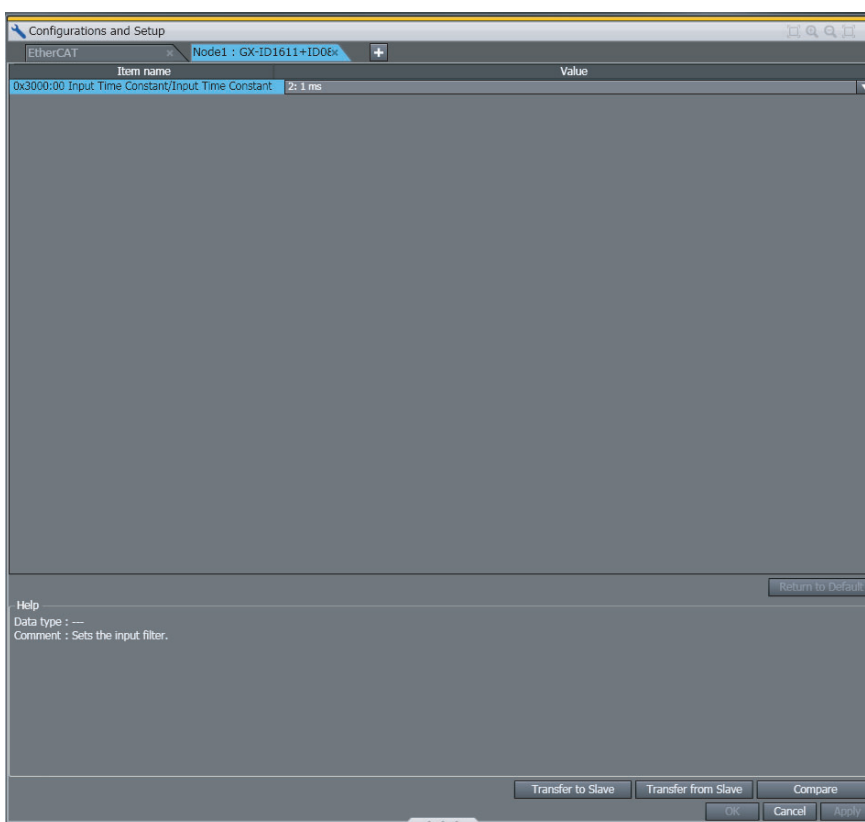
3   Click the **Compare** Button.

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 of 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.

9-2-4 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.
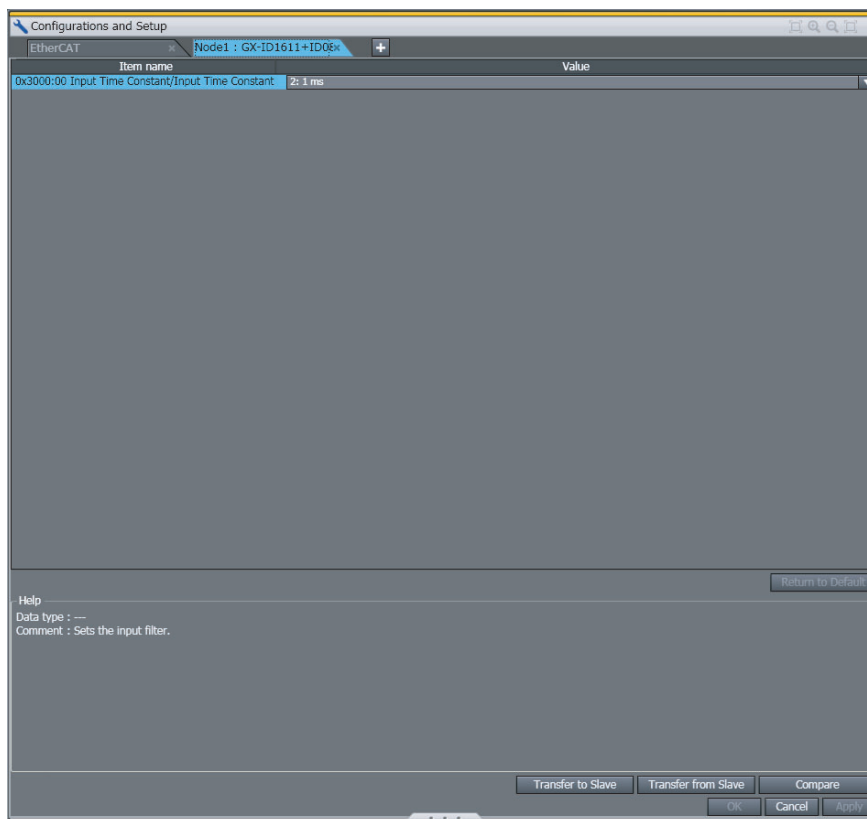
3 Click the **Compare** Button.

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.

9-2-5   Replacement Procedure

```
                          ┌─────────────────┐
                          │      START       │
                          └─────────────────┘
                                   │
            ┌─ ─ ─ ─ ─ ─ ─ ─ ─ ─ ─ ─ ─ ─ ─ ─ ─┐
            │   Back up the backup parameters.  │        Process data communications
            └─ ─ ─ ─ ─ ─ ─ ─ ─ ─ ─ ─ ─ ─ ─ ─ ─┘        are performed with the slave
                                   │                      before it is disconnected.
            ┌─────────────────────────────────┐
            │ 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.│
            └─ ─ ─ ─ ─ ─ ─ ─ ─ ─ ─ ─ ─ ─ ─ ─ ─┘        Process data communications
                                   │                      are interrupted with the slave.
            ┌─────────────────────────────────┐
            │ Physically connect the new        │
            │ EtherCAT slave to the EtherCAT    │
            │ network.                          │
            └─────────────────────────────────┘
                                   │
            ┌─ ─ ─ ─ ─ ─ ─ ─ ─ ─ ─ ─ ─ ─ ─ ─ ─┐
            │ If the node address is set is 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   │        Process data communica-
            │ or with the Connect EtherCAT     │        tions are restarted with
            │ Slave instruction.               │        the connected slave.
            └─────────────────────────────────┘
                                   │
                          ╱ ─ ─ ─ ─ ─ ─ ╲
                        ╱ Are all disconnected ╲
              NO       ╱  slaves now reconnected? ╲
            ◄─ ─ ─ ─ ─╲                           ╱
                        ╲                         ╱
                          ╲ ─ ─ ─ ─ ─ ─ ─ ╱
                                   │  YES
                          ┌─────────────────┐
                          │       END        │
                          └─────────────────┘
```

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 countermeasures 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 the Validity of the Process Data*.
  - Set the PDO communications timeout detection count setting in the EtherCAT master to at least 2. Refer to *5-4-1 Setting EtherCAT Master* for the setting procedure.

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.)
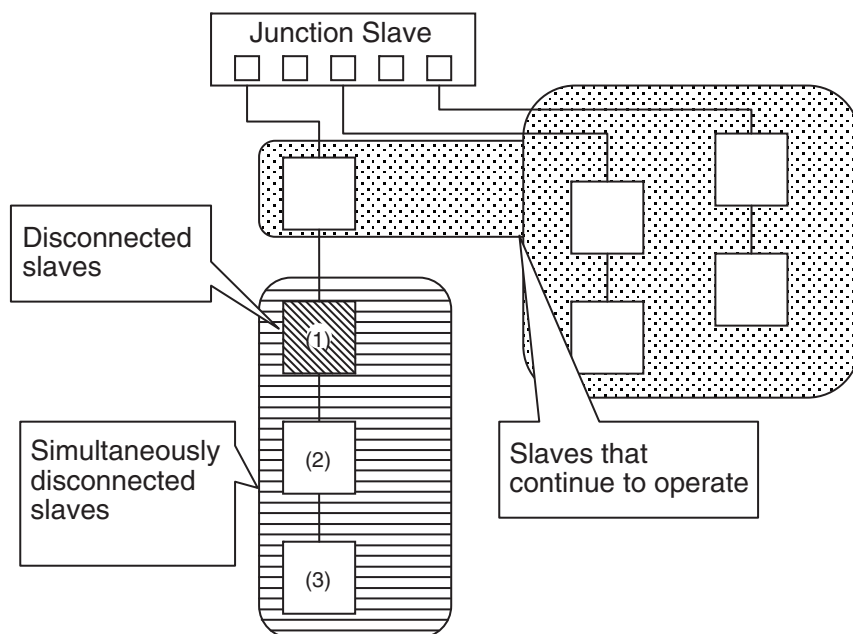
The status of the system-defined variables for disconnected slaves are listed in the following table. The status in the following table are reached immediately after disconnection. This status does not change even if a slave is physically removed from the actual network configuration or physically connected to the actual network configuration again. The corresponding element in the Network Connected Slave Table changes to FALSE when the slave is physically removed from the actual network configuration.

System-defined variable name	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.	The corresponding element changes to FALSE when the slave is removed from the actual network configuration.
_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.	The corresponding element changes to FALSE after the EtherCAT slave is disconnected from the Sysmac Studio or with an instruction.
_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.	The corresponding element changes to FALSE after the EtherCAT slave is disconnected from the Sysmac Studio or with an instruction.
_EC_DisconnSlavTbl[] (Disconnected Slave Table)	This variable shows the slaves that are registered in the network configuration information and disconnected from the network.	The corresponding element changes to TRUE after the EtherCAT Slave is disconnected from the Sysmac Studio or with an instruction.

Reconnecting Slaves

When reconnecting disconnected slaves after replacement or inspection, make sure that the following conditions are met and then reconnect them. If they are reconnected without meeting following conditions, process data communications with the slaves will not start again, and a network configuration verification error will occur. If the conditions are met, _EC_EntrySlavTbl 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.

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).)
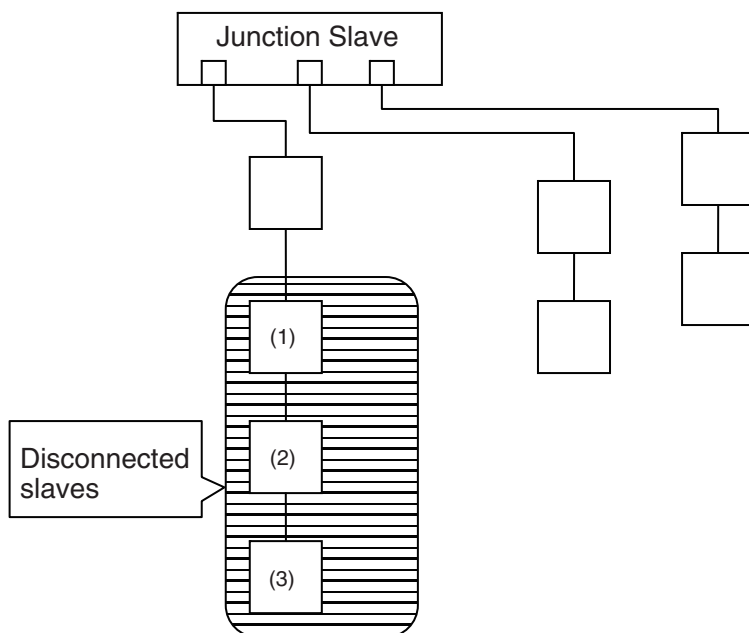
The status of the system-defined variables for disconnected slaves that have been successfully reconnected are listed in the following table.

System-defined variable name	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.	Shows actual network configuration (changes to FALSE for disconnection and to TRUE for connection).
_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.	The corresponding element changes to TRUE after the EtherCAT slave is connected from the Sysmac Studio or with an instruction
_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.	The corresponding element changes to TRUE after the EtherCAT slave is connected from the Sysmac Studio or with an instruction
_EC_DisconnSlavTbl[] (Disconnected Slave Table)	This variable shows the slaves that are registered in the network configuration information and disconnected from the network.	The corresponding element changes to FALSE after the EtherCAT Slave is disconnected from the Sysmac Studio or with an instruction.

Additional Information

Synchronization correction processing is performed to reconnect synced slaves. Therefore, several seconds may be required per slave until reconnection is completed.

10

Diagnostic and Statistical Information

This section describes the procedure to diagnose the EtherCAT network line quality using diagnostic and statistical information, as well as the methods to acquire the diagnostic and statistical information.

10

10-1 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 diagnose the EtherCAT network line quality 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

10-2 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.

10-2-1 Overall EtherCAT Network Diagnostic Procedure

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.

10-2-2 Detailed EtherCAT Network Diagnostic Procedure

A detailed procedure from EtherCAT network diagnosis through correction is given below.

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 as a CSV file on the computer.	page 10-11
Using the diagnosis/statistics log of Controller	The Controller acquires the diagnostic and statistical information periodically. The acquired diagnostic and statistical information is saved in a Virtual SD Memory Card on the Controller.	page 10-18

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.

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

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 *Finding Locations of Errors with Trends in Slave Diagnostic and Statistical Information* on page 10-5.

📓 **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.

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

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.

10-2 EtherCAT Network Diagnostic Procedure

10

10-2-2 Detailed EtherCAT Network Diagnostic Procedure

Examples of Finding Error Locations

Example 1: Network Configuration Where an EtherCAT Junction Slave Is Not Used

Network configuration

Slave Diagnosis/Statistics Tab Page

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

Slave Diagnosis/Statistics Tab Page

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

10-2 EtherCAT Network Diagnostic Procedure

10

10-2-2 Detailed EtherCAT Network Diagnostic Procedure

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 Ether-CAT 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.

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

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 Diagnosis/Statistics.

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 for some slaves.	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 Controller 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.

5   Confirming After Implementing Corrections

To confirm that the error locations were corrected, perform *Checking for Errors in Trends in Master Diagnostic and Statistical Information* on page 10-4 again to make sure the network is operating normally.

10-3 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.

10-3-1 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** in the Multiview Explorer. Or, right-click **EtherCAT** under **Configurations and Setup** and select *Edit*.
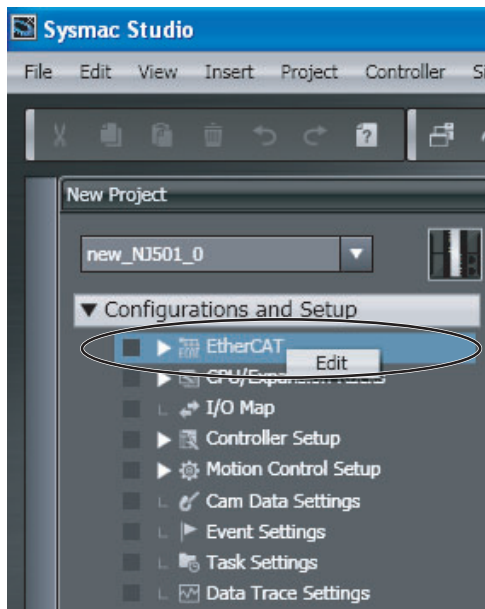
3 Right-click the EtherCAT master on the EtherCAT Tab Page and select ***Display Diagnosis/Statistics Information*** from the menu.
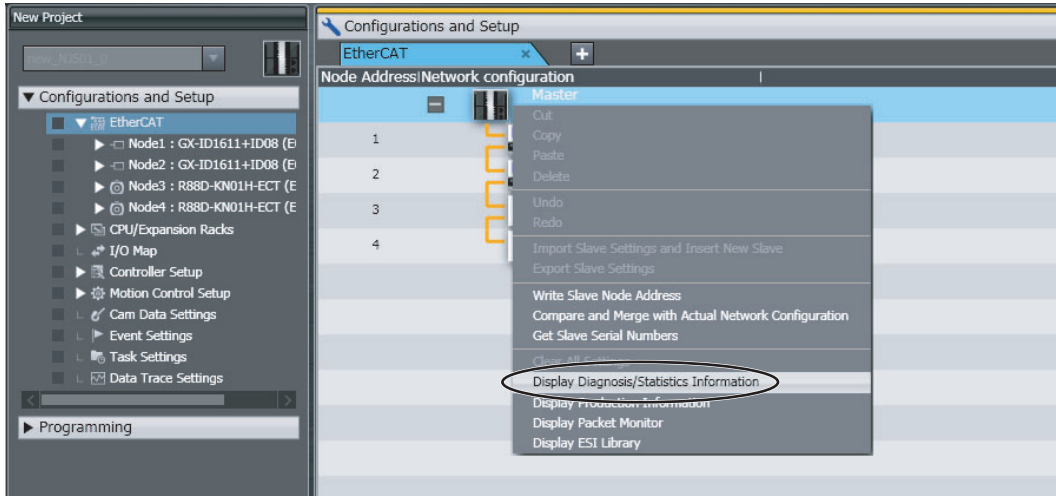
The diagnostic and statistical information for the master and slaves are displayed.

10-3-2 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	Clears all values of the slave diagnostic and statistical information to zeros.
Output File	Outputs the diagnostic and statistical information to a CSV file in a specified folder. The following data are output. • Master diagnostic and statistical information • Slave diagnostic and statistical information [2]

*1 Only for the EtherCAT slaves that are registered in the network configuration information of the project.

*2 If the number of error frames exceeds 255, 255 is output.

10-3-3 Displaying Master Diagnostic and Statistical Information

The following are the items displayed for the master diagnostic and statistical information.

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 Controller 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 time-out count	The number of frame reception timeouts that occurred.	A cause to a Process Data reception Timeout occurred, which increases the Frame reception time-out count. The Frame reception time-out 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 10-3.
Reception buffer over-flow 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 occurred.	Refer to the *NY-series Troubleshooting Manual* (Cat. No. W564) and remove the cause of the error.

Item	Displayed information	Assumed cause of increase	Possible correction
Discarded process data receptions	The number of process data packets discarded when receiving process data.	Operations for which multi-execution is prohibited in *Prohibition to Physically Disconnecting a Slave and Resetting an Error or Connecting a Slave at the Same Time* on page 9-2 were executed. The value sometimes increases when the network is started or communicating EtherCAT slaves are replaced.	Refer to *Prohibition to Physically Disconnecting a Slave and Resetting an Error or Connecting a Slave at the Same Time* on page 9-2.
Discarded message receptions	The number of messages discarded in mailbox reception.	A cause to an EtherCAT Message Error occurred.	Refer to the *NY-series Troubleshooting Manual* (Cat. No. W564) 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 10-3.
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.
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 10-3.
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 10-3.

Precautions for Correct Use

Even if communications are normal, values other than 0 may occur for the frame reception time-out count and the number of CRC error frames received. Refer to *EtherCAT Network Diagnostic Procedure* on page 10-3 to determine if the EtherCAT network is operating normally.

10-3-4 Displaying Slave Diagnostic and Statistical Information

The following are the items displayed for the slave diagnostic and statistical information.

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).
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: 10 (+5)
	• *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 Controller 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.

10

10-4 Diagnosis/Statistics Log of Controller

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

10-4-1 Overview of the Diagnosis/Statistics Log

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 a Virtual SD Memory Card on the Controller.

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 the settings for the SD Memory Card operations in the Controller. Refer to the *NY-series Industrial Panel PC / Industrial Box PC Software User's Manual* (Cat. No. W558) for information on setting the SD Memory Card operations.
- 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 diagnostic and statistical information to acquire and describes the information.

Diagnostic and statistical 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 *Displaying Master Diagnostic and Statistical Information* on page 10-13 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 *Displaying Slave Diagnostic and Statistical Information* on page 10-16 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.

Specifications of Log File

The acquired diagnostic and statistical information is saved as a CSV log file in the Virtual 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 Virtual 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 a Virtual 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 Controller 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.
• 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.

Additional Information

The items of information such as the master diagnostic and statistical information output to a CSV file by this function are arranged in the same order as the diagnostic and statistical information display of Sysmac Studio. You can make the same file format by exchanging the places of the row and column of each item.

10-4-2   Setting the Diagnosis/Statistics Log

● 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.

10-4-3   System-defined Variables Related to the Diagnosis/Statistics Log

The following system-defined variables are used to control execution of the diagnosis/statistics log. Refer to *7-1-2 System-defined Variables* 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

10-4-4   Procedure to Use the Diagnosis/Statistics Log

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 (e) 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 Virtual SD Memory Card does not have sufficient available space.**
- **(c) The Virtual SD Memory Card settings are not configured.**
- **(d) A synchronization (or downloading) is executed on the Sysmac Studio.**
- **(e) 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) or (e) 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.

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
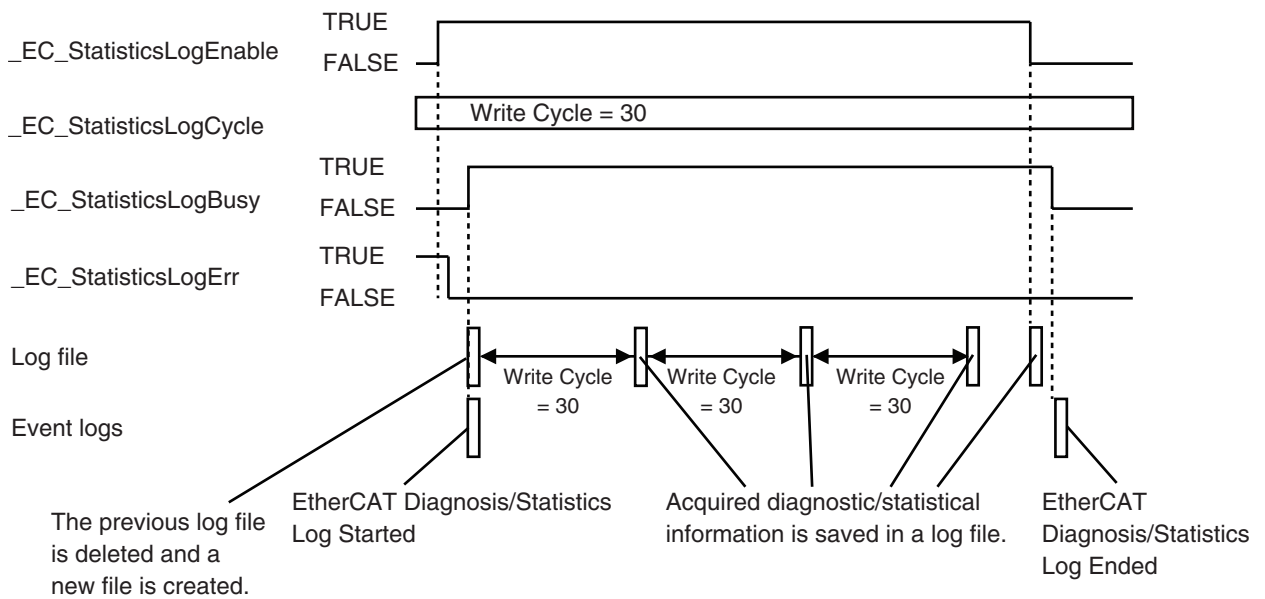
Precautions for Correct Use

- To retain the log file saved in the Virtual 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 Virtual 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 Virtual 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.
- Take the log file out from the Virtual SD Memory Card and save it in another medium.

10-4 Diagnosis/Statistics Log of Controller

10

10-4-4 Procedure to Use the Diagnosis/Statistics Log

A

Appendices

The appendices describe the relation of EtherCAT communications to overall Controller status, packet monitoring functions, and multi-vendor application.

A-1 EtherCAT Status in Relation to Controller Status

The status of EtherCAT master memory, the ability to download master and slave settings, and the slave status are given below for different Controller 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   Controller Operating Modes

Controller 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		Slave outputs	Slave communications status
PROGRAM mode	Shows the I/O data exchanged on the EtherCAT communications.	Continually shows EtherCAT communications status. (Does not depend on the operating mode of the Controller.)	Does not change. (Does not depend on the operating mode of the Controller.)	Continually shows EtherCAT communications status. (Does not depend on the operating mode of the Controller.)	• Master settings: OK • Slave settings: OK (However, setting may not be possible depending on the status of the slave.)	Continually shows the output data sent from the EtherCAT master. (Does not depend on the operating mode of the Controller.)	Does not change. (Does not depend on the operating mode of the Controller.)
Switching between PROGRAM and RUN mode	Device variables that are not retained: Initial values Axes variables: Retained				---		
RUN mode	Shows the I/O data exchanged on the EtherCAT communications.				• Master settings: Not possible. • Slave settings: OK (However, setting may not be possible depending on the status of the slave.)		

A-1-3 Controller Errors Other Than Errors in the Built-in EtherCAT Master

Controller 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		Slave outputs	Slave communications status
A major fault level Controller error occurs.	Device variables that are not retained: Initial values Axes variables: Retained	Continually shows EtherCAT communications status. (Does not depend on the operating mode of the Controller.)	Does not change. (Does not depend on the operating mode of the Controller.)	Continually shows EtherCAT communications status. (Does not depend on the operating mode of the Controller.)	---	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. (Does not depend on the operating mode of the Controller.)	Does not change. (Does not depend on the operating mode of the Controller.)
A minor fault level Controller error occurs.							

Refer to the *NY-series Troubleshooting Manual* (Cat. No. W564) if a Controller error occurs in the built-in EtherCAT master.

A-1-4 Others

Controller 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		Slave outputs	Slave communications status
Memory all clear	Cleared to all zeros.	Entirely initialized (0).	Deleted.	Cleared to all zeros.	---	Depends on the slave settings.	EtherCAT communications stop.

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		Slave outputs	Slave communications status
Does not change.	Does not change.	Does not change.	Updating the protocol monitor stops.	---	Depends on the slave settings.	EtherCAT communications stop.

A-2 Monitoring Packets

Packet monitoring stores a certain number of the most recent packets sent and received by the Ether-CAT 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 Controller.

You can use any of the following methods to obtain the packet data saved in the system memory in the Controller.

- Reading directly from the Sysmac Studio
- Saving to a Virtual SD Memory Card in the Controller

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.
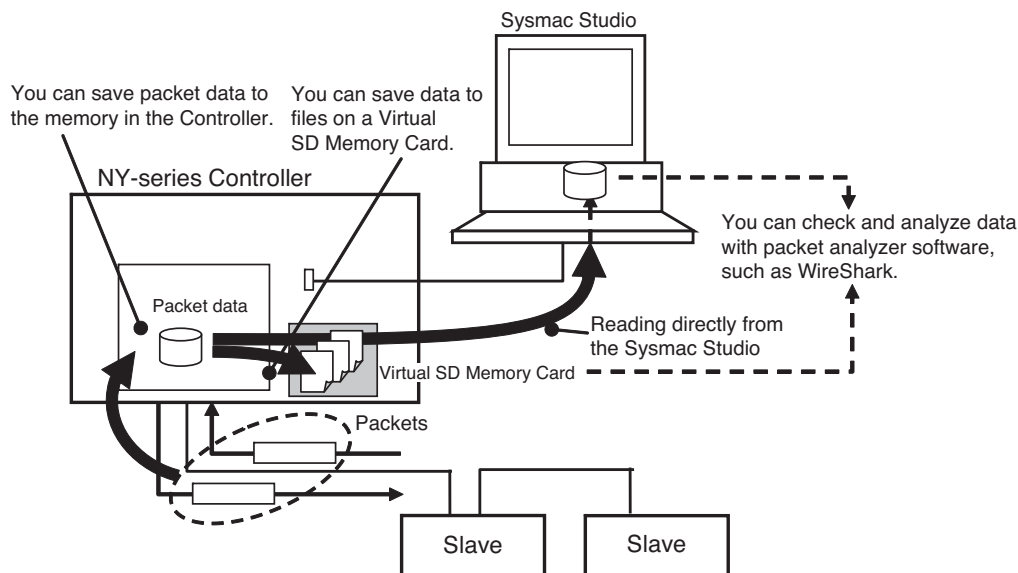
Precautions for Correct Use

To save packet data on a Virtual SD Memory Card, you need the settings for using the SD Memory Card operations in the Controller. For information on using the SD Memory Card operations, refer to the *NY-series Industrial Panel PC / Industrial Box PC Software User's Manual* (Cat. No. W558).

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 Setups** in the Multiview Explorer. Or, right-click **EtherCAT** under **Configurations and Setups** and select *Edit*.
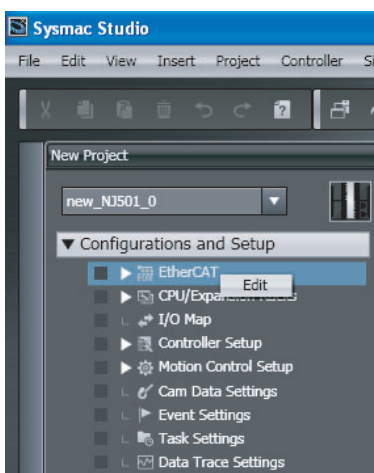
3 Right-click in the EtherCAT Network Configuration Tab Page and select *Display Packet Monitor* from the menu.
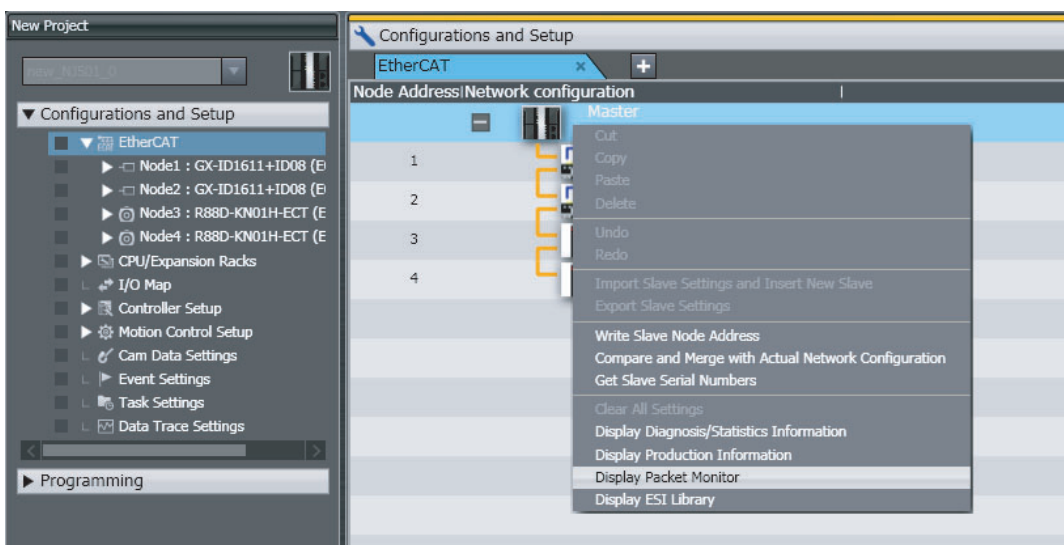
The Packet Monitor Dialog Box is displayed.

4 Click the **Start** Button to begin monitoring packets.
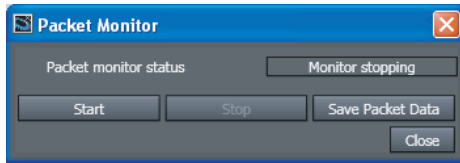
5 Click the **Stop** Button to stop monitoring packets.
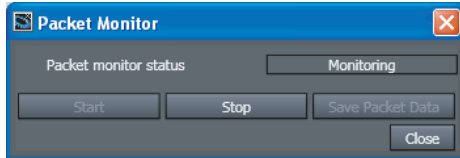
Saving Packet Data

● **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.
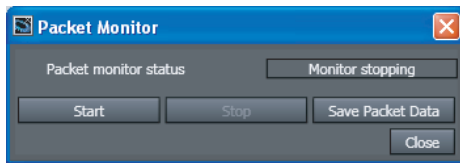
The Save Dialog Box is displayed.

3 Input a file name, and then click the **Save** Button.

A TCPDump packet data file with a .cap extension is saved.

● **Saving to a Virtual SD Memory Card from Memory in the Controller**

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 Controller. Then, execute the EC_CopyMon (Transfer EtherCAT Packets) instruction to save the packet data that was saved in the Controller to a file on a Virtual SD Memory Card in the Controller. You can specify the name of the file that is saved in the Virtual SD Memory Card. The number of files is limited only by the space that is available on the Virtual SD Memory Card.

> **Additional Information**
>
> - Packet data in the Controller'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 you execute the instruction to copy the data to a Virtual SD Memory Card without the settings for using the SD Memory Card operations, then an error is returned. In that case, only the one file in the Controller'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	Controller's system: 1 file Virtual SD Memory Card in Controller: Multiple files (up to capacity of Virtual SD Memory Card)

Sample Programming

This sample transfers EtherCAT communications packets to a Virtual SD Memory Card when an EtherCAT slave error occurs. The file name is 'PacketFile.' The processing procedure is as follows:

1 The system-defined variable *_EC_ErrSta* (EtherCAT Error) 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 Controller.

4 The EC_CopyMon instruction is used to copy that file to the Virtual 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	☑	Built-in EtherCAT Error
	_EC_PktMonStop	BOOL	☑	Packet Monitoring Stopped
	_EC_PktSaving	BOOL	☑	Saving Packet Data File
	_Card1Ready	BOOL	☑	SD Memory Card Enabled 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.

```
        <>              Up    _EC_PktMonStop   EC_StopMon_instance.Busy    A
        EN              In
_EC_ErrSta —In1
WORD#16#00 —In2
```

A   EC_SaveMon_instance.Busy   EC_CopyMon_instance.Busy   EC_StartMon_instance.Busy   B

```
                    RS_instance
B     _EC_PktSaving          RS        Operating
                       Set      Q1
      OperatingEnd —  Reset1
```

NY-series Industrial Panel PC / Industrial Box PC Built-in EtherCAT Port User's Manual (W562)

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              Done
'PacketFile' —FileName    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	☑	Built-in EtherCAT Error
	_EC_PktMonStop	BOOL	☑	Packet Monitoring Stopped
	_EC_PktSaving	BOOL	☑	Saving Packet Data File
	_Card1Ready	BOOL	☑	SD Memory Card Enabled 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 Ether-CAT 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.
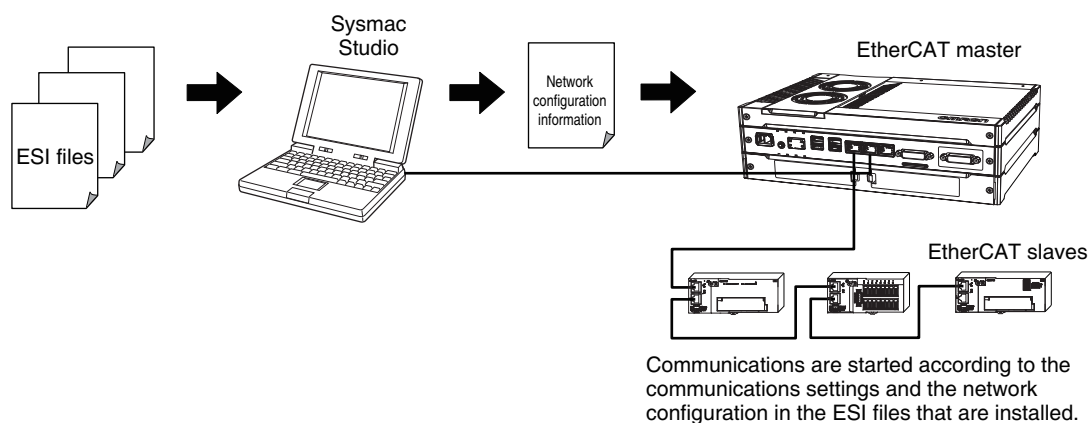
📝 **Additional Information**

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 NY-series Controllers 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 tables gives the data types that are supported by different Sysmac Studio versions.

Data Type	Sysmac Studio Ver.
	1.17
BIT1, BIT2, BIT3, BIT4, BIT5, BIT6, BIT7, and BIT8	Supported.
BITARR8, BITARR16, and BITARR32	Supported.
ARRAY[0..n] OF BYTE	Supported.
ARRAY[0..n] OF UINT	Supported.
INT24, INT40, INT48, and INT56	Not supported.
UINT24, UINT40, UINT48, and UINT56	Not supported.
STRING(n)	Not supported.

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 NY-series Controller.

A-3-3 Installing ESI Files

● **Procedure**

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 NY-series EtherCAT master, you must install the ESI file for that slave in the Sysmac Studio to enable setting the slave.

Use the following procedure to install an ESI file.

1 Double-click **EtherCAT** under **Configurations and Setup** in the Multiview Explorer. Or, right-click **EtherCAT** under **Configurations and Setup** and select *Edit*.
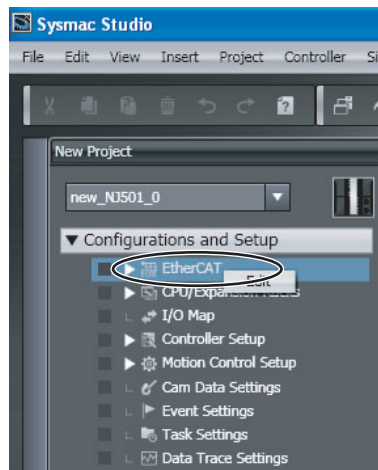
The edit pane is displayed.

2 Right-click the EtherCAT master that is displayed in the edit pane and select *Display ESI Library*.
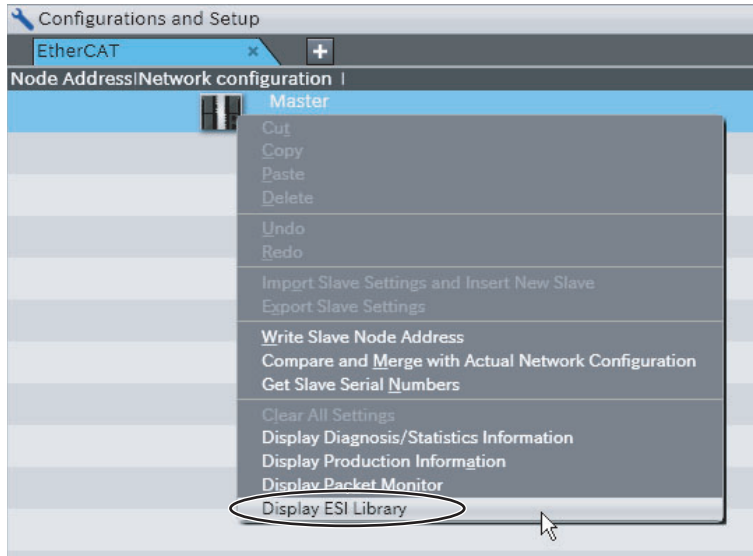
The ESI Library Dialog Box is displayed.

3 Click the **this folder** link in the ESI Library Dialog Box.
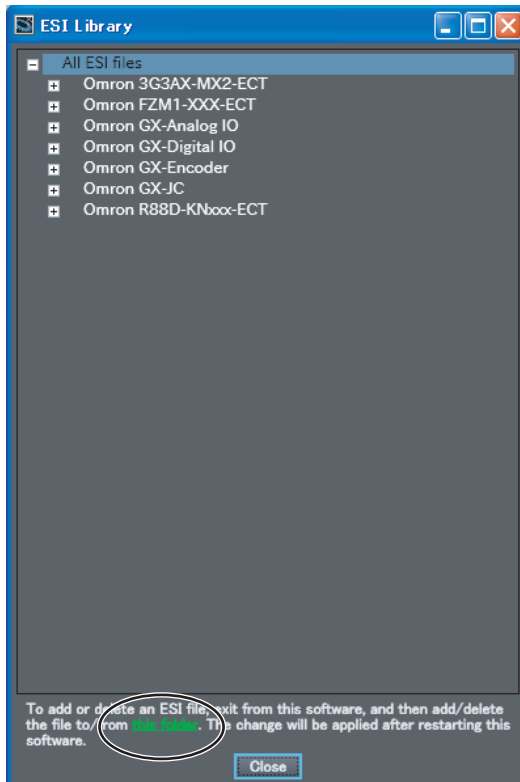
The contents of the EsiFiles folder are displayed.

4 Copy the ESI file for the EtherCAT slave that is manufactured by another company and paste it in the EsiFiles folder. Obtain the ESI file from the slave manufacturer. The ESI file must conform to the most recent ETG ESI specifications.
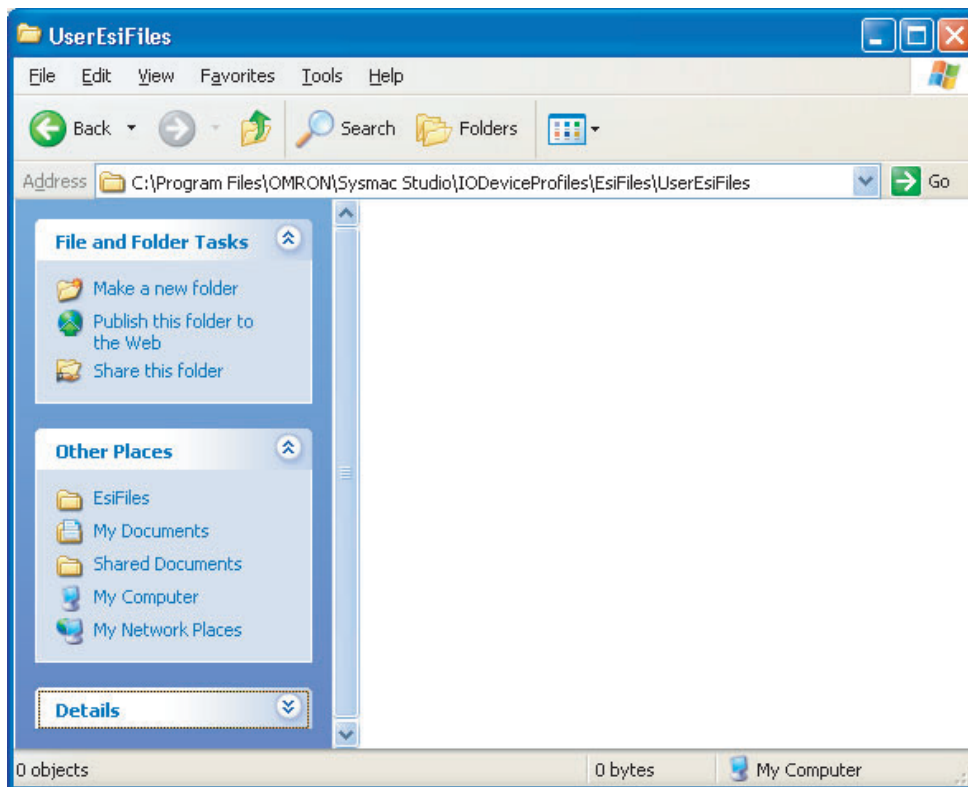
5 Restart the Sysmac Studio.

The ESI file that you installed is displayed in the ESI Library Dialog Box after you restart the Sysmac Studio.

● **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** in the Multiview Explorer. Or, right-click **EtherCAT** under **Configurations and Setup** and select *Edit*.
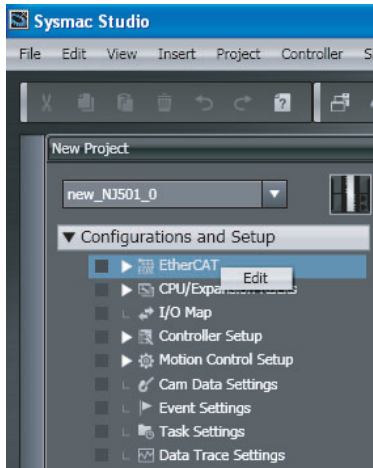
The edit pane is displayed.

2 Right-click the EtherCAT master that is displayed in the edit pane and select *Display ESI Library*.
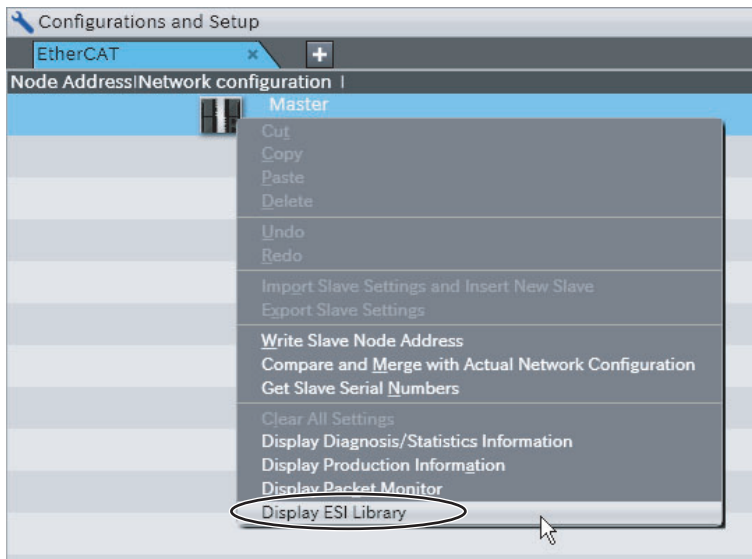
The ESI Library Dialog Box is displayed.

3 Click the + Icon to the left of the name of the ESI file that was added.
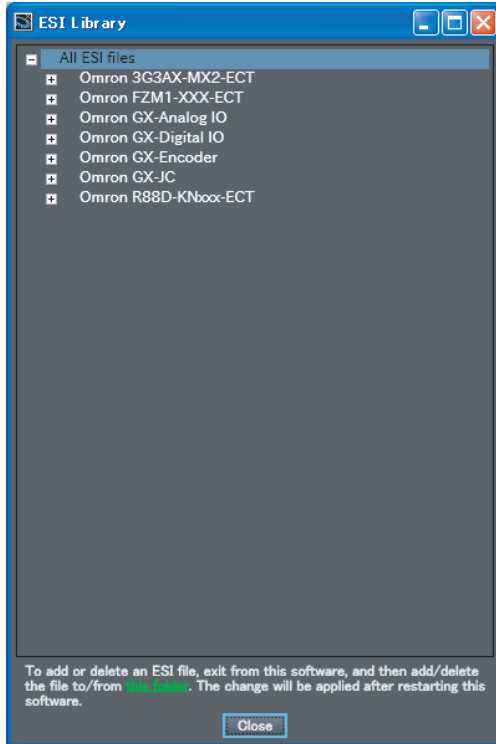
The slave definitions in the ESI file are expanded so that you can check the following items.

- Model
- Revision
- Product name
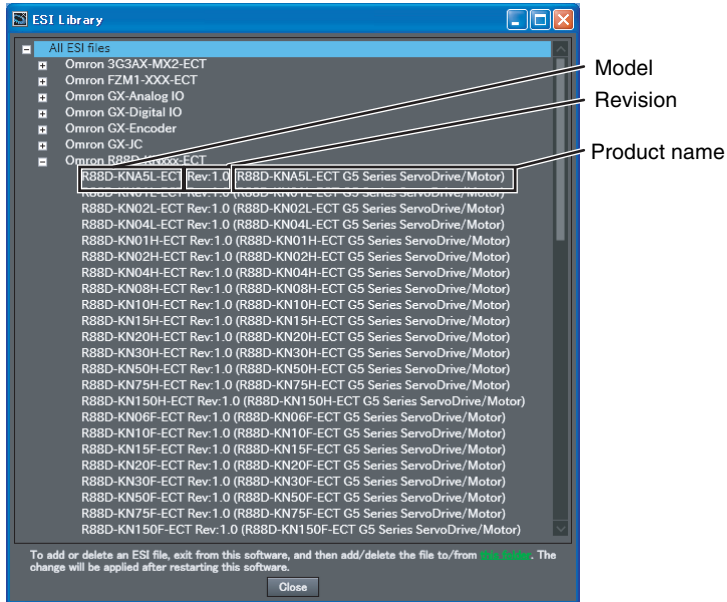
4 If an exclamation mark is not displayed, click the **Close** Button.
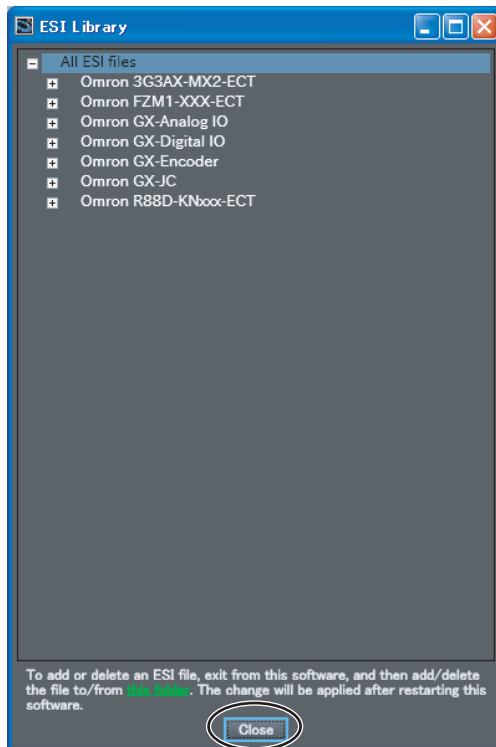
📓 **Additional Information**

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.

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.

The following buttons are used to edit a PDO entry table. The meanings of the buttons are given below.
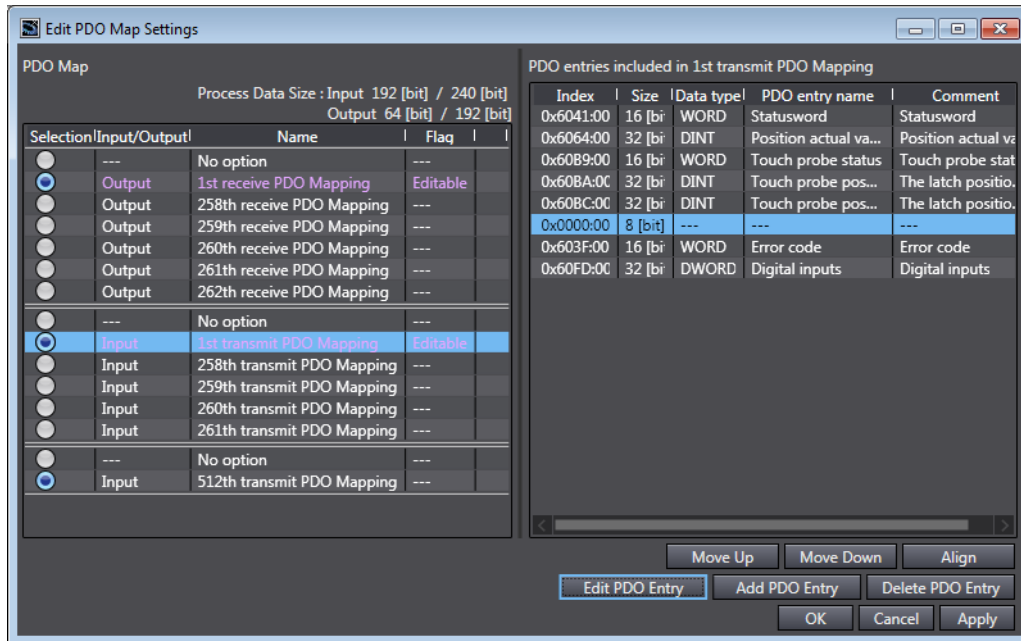
● **Add PDO Entry Button and Delete PDO Entry Button**

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.

● **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 and Move Down Button**

Use these buttons to change the order of the PDO entries by moving objects up and down.

● **Align Button**

Use this button to automatically order the PDO entries and adjust byte boundaries.

The PDO entries 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.).

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.
- Sysmac Studio version 1.08 or higher is required to add "0x0000:00" to a PDO entry table.

Editing Methods for PDO Entries

● Adding PDO Entries

Click the **Add PDO Entry** Button. The entry table from 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.
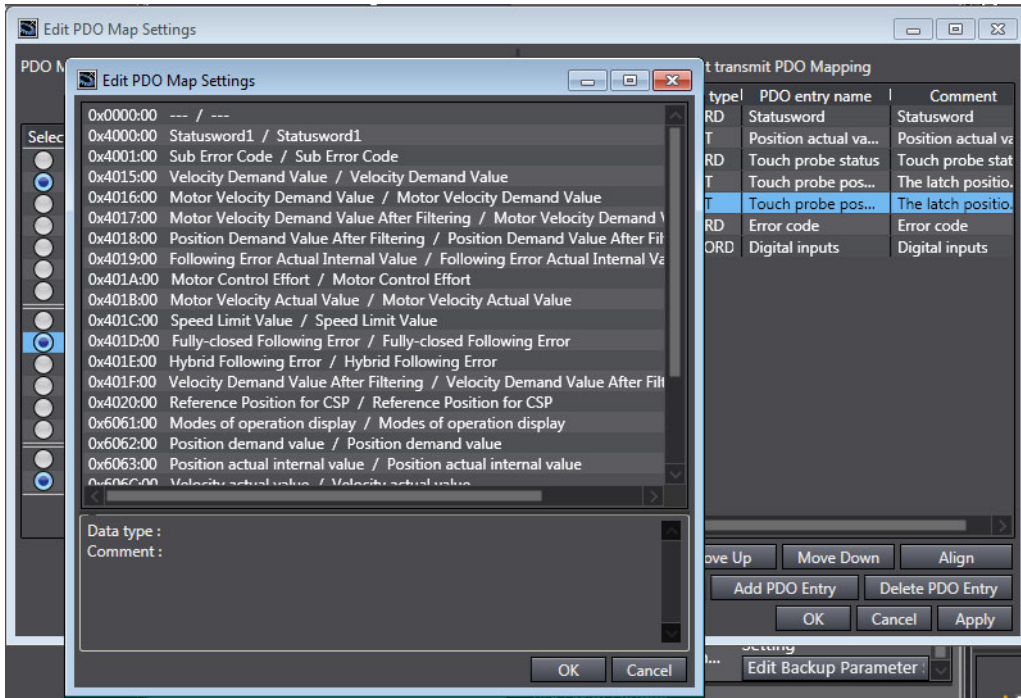
● Editing PDO Entries

Use the cursor to 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.
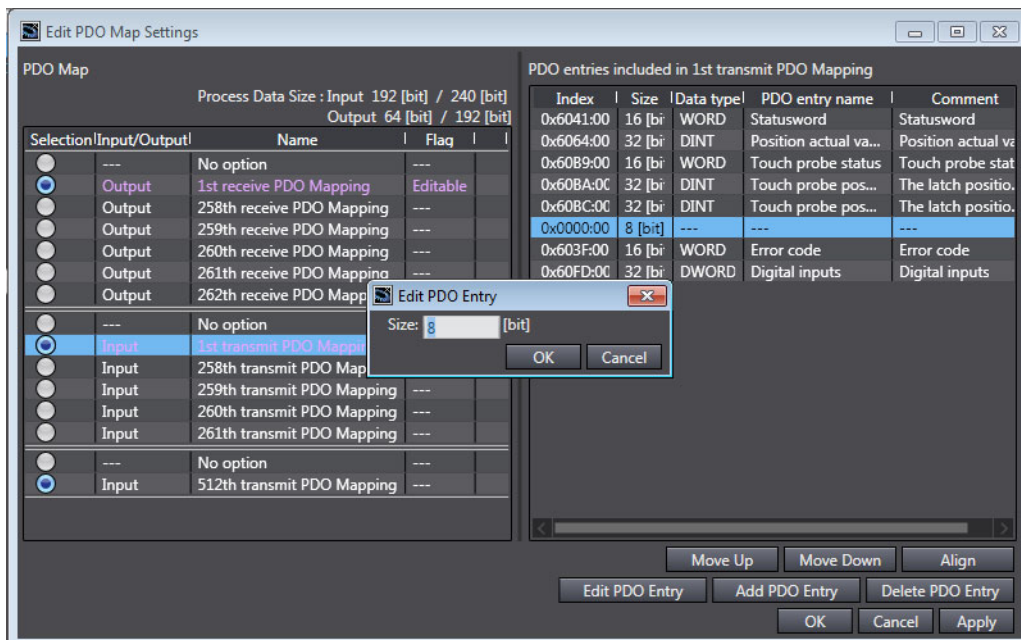
A-3-5　Settings for MDP-compatible Slaves from Other Manufacturers

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.

1 Right-click the slave in the EtherCAT Tab Page and select *Edit Module Configuration*.

An Edit Pane for the module configuration is displayed.
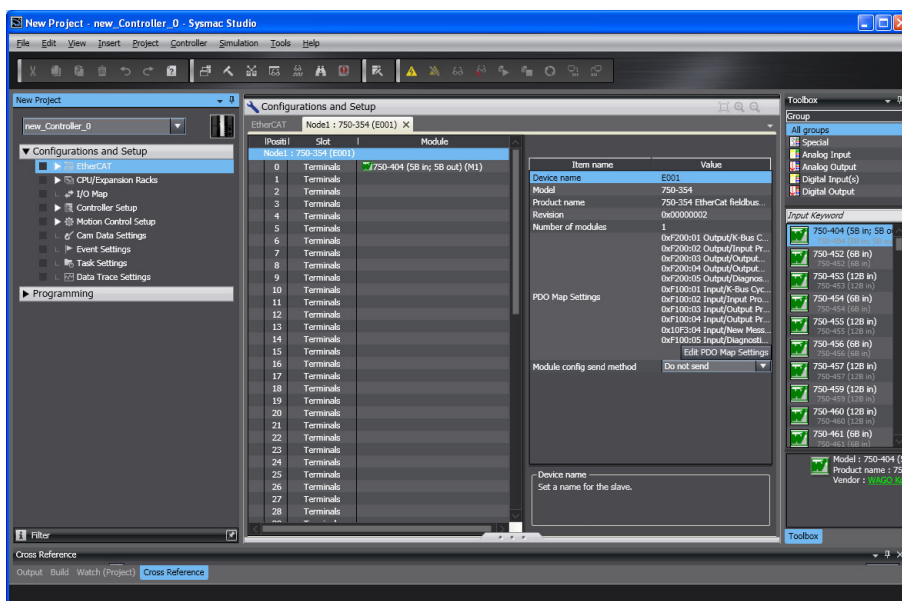
Index	Size	Data type	PDO entry name	Comment
0x6000:00	16[bit]	UINT	uiAAA	
0x6001:00	2[bit]	BIT2	b2BBB	
0x6002:00	4[bit]	BIT4	b4CCC	
0x0000:00	2[bit]	---	---	
0x6010:00	32[bit]	UDINT	udEEE	

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).

2 Select a slave on the Edit Pane for the module configuration.

A list of the slave settings is displayed.
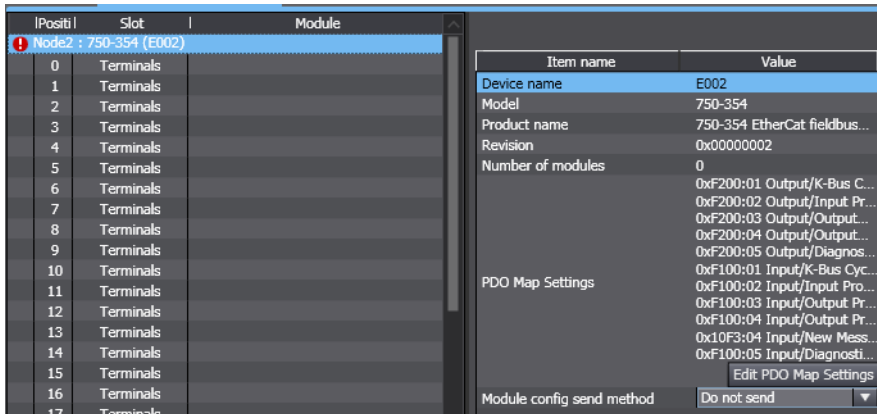
Some items in the slave settings only display the current setting. The other items in the slave settings are changeable. The set values and their meanings are given below for the settings that are only displayed.

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

You can edit the *PDO Map Settings* and *Module config send method*. 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. For details, refer to *5-3 Setting EtherCAT Slave Variables and Axes*.

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*.

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.
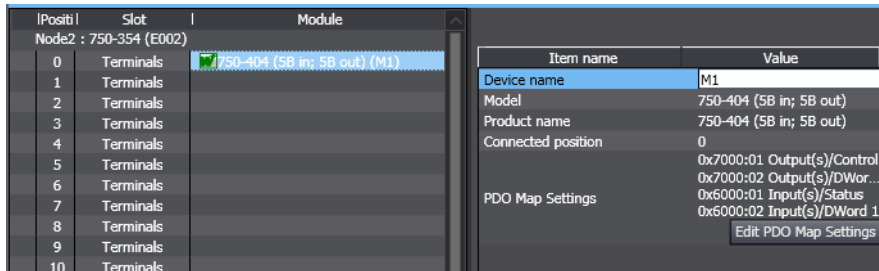
Some items in the module settings only display the current setting. The other items in the module settings are changeable. The set values and their meanings are given below for the settings that are only displayed.

Name	Set value that is displayed
Model	The model name of the device is displayed.
Product name	The product name of the device is displayed.
Connected position	The connected position of the device is displayed.

You can edit the *Device name* and the *PDO Map 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. For details, refer to *5-3 Setting EtherCAT Slave Variables and Axes*.

A-4 Glossary

Use the following list of EtherCAT terms for reference.

Term	Abbrevia-tion	Description
AL status (application layer status)	–	Status for indicating information on errors that occur in an application on a slave.
CAN application protocol over Ether-CAT	CoE	A CAN application protocol service implemented on EtherCAT.
CAN in Automation	CiA	CiA is the international users' and manufacturers' group that develops and supports higher-layer protocols.
device profile	–	Collection of device dependent information and functionality providing consistency between similar devices of the same device type.
device variables	–	Variables in the NY-series Controller to which process data on Ether-CAT slaves are allocated. Slave process data is accessed by directly reading and writing these device variables from user applications in the NY-series Controller.
distributed clocks	DC	Clock distribution mechanism used to synchronize EtherCAT Slave Units and the EtherCAT Master Units.
EtherCAT Master Function Module	–	One of the function modules. This module controls the EtherCAT slaves as the EtherCAT master.
EtherCAT slave controller	ESC	A controller for EtherCAT slave communication.
EtherCAT slave information	ESI	An XML file that contains setting information for an EtherCAT Slave Unit.
EtherCAT state machine	ESM	An EtherCAT communication state machine.
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.
index	–	Address of an object within an application process.
network configuration information	–	The EtherCAT network configuration information held by the EtherCAT master.
object	–	Abstract representation of a particular component within a device, which consists of data, parameters, and methods.
object dictionary	OD	Data structure addressed by Index and Sub-index that contains description of data type objects, communication objects and application objects.
Operational	–	A state in EtherCAT communications where SDO communications and I/O are possible.
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.
PDO communications	–	An acronym for process data communications.
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 downloaded cyclically or acyclically for the purpose of measurement and control.
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.
process data object	PDO	Structure described by mapping parameters containing one or several process data entities.
receive PDO	RxPDO	A process data object received by an EtherCAT Slave Unit.

Term	Abbrevia-tion	Description
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.
SDO communications	–	One type of EtherCAT communications that uses service data objects (SDOs) for communicating information when required.
service data object	SDO	CoE asynchronous mailbox communications where all objects in the object dictionary can be read and written.
Slave Information Interface	SII	Slave information that is stored in non-volatile memory in the slave.
subindex	–	Sub-address of an object within the object dictionary.
sync jitter	–	The offset in the operation timing between slaves.
sync manager	SM	Collection of control elements to coordinate access to concurrently used objects.
system-defined variable	–	A variable in the NY-series Controller 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 NY-series Controller.
transmission jitter	–	The jitter in the process data transmission period (ns).
transmit PDO	TxPDO	A process data object sent from an EtherCAT Slave Unit.
WireShark	–	Freeware for monitoring and capturing packets.

A-5　Version Information

This appendix describes the changes in and additions to functional specifications that were made for changes in the unit version of the Controller.

● **Adding the EtherCAT Slaves That You Can Use**

If you use a newer unit version of the Controller, you can increase the OMRON EtherCAT slaves that you can use. Refer to *1-2-1 System Configuration* for an example of the OMRON EtherCAT slaves that are available.

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.

Index

I

Index

I