OMRON

Machine Automation Controller

NX-series

# Safety Control Unit

## User's Manual

NX-SL□□□□
NX-SI□□□□
NX-SO□□□□




SYSMAC
always in control

Z930-E1-32

**NOTE**

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

**Trademarks**

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

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

**Copyrights**

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

# Introduction

Thank you for purchasing an NX-series Safety Control Unit.
This manual contains information that is necessary to use the NX-series Safety Control Unit.
Please read this manual and make sure you understand the functionality and performance of the Unit before you attempt to use it in a control system.
Keep this manual in a safe place where it will be available for reference during operation.

## Intended Audience

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

- Personnel in charge of introducing FA systems.
- Personnel in charge of designing FA systems.
- Personnel in charge of installing and maintaining FA systems.
- Personnel in charge of managing FA systems and facilities.
- Personnel with the qualifications, authority, and responsibility for providing safety at each phase of the lifecycle of the machine: design, installation, operation, maintenance, and disposal.
- Personnel with a knowledge of functional safety.

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

## Applicable Products

This manual covers the following products.

- NX-series Safety Control Units
  NX-SL□□□□
  NX-SI□□□□
  NX-SO□□□□

Note that this manual provides information for using an NX-series Safety Control Unit described above together with an NJ/NX-series CPU Unit, an EtherCAT Coupler Unit, or with an EtherNet/IP Coupler Unit. When you use it with an NX-CSG320 Communication Control Unit, refer to the *NX-series Safety Control Unit / Communication Control Unit User's Manual (Cat. No. Z395)*.

# Sections in this Manual


1 Overview

2 Specifications

3 Installation and Wiring

4 Safety Control Unit Operation

5 System Configuration and Setup

6 Programming

7 Checking Operation and Actual Operation

8 Calculating Safety Reaction Times

9 Communications Load

10 Safety Unit Restore

11 Safety Data Logging

12 Troubleshooting

13 Inspection and Maintenance

A Appendices

I Index

# CONTENTS

**Introduction** ..... **1**
Intended Audience ..... 1
Applicable Products ..... 1

**Sections in this Manual** ..... **3**

**Relevant Manuals** ..... **12**

**Manual Structure** ..... **14**
Page Structure ..... 14
Special Information ..... 15
Precaution on Terminology ..... 15

**Terms and Conditions Agreement** ..... **16**
Warranty, Limitations of Liability ..... 16
Application Considerations ..... 17
Disclaimers ..... 17
Statement of security responsibilities for assumed use cases and against threats ..... 18

**Safety Precautions** ..... **19**
Definition of Precautionary Information ..... 19
Symbols ..... 19
Warnings ..... 20

**Precautions for Safe Use** ..... **26**

**Precautions for Correct Use** ..... **31**

**Regulations and Standards** ..... **34**
Conformance to EU Directives ..... 34
Conformance to EN ISO 13849-1 ..... 35
Conformance to UL and CSA Standards ..... 36
Conformance to Shipbuilding Standards ..... 36
Conformance to KC Certification ..... 36

**Unit Versions** ..... **38**
Unit Versions ..... 38
Unit Versions of Units and Sysmac Studio Versions ..... 39

**Related Manuals** ..... **40**

**Terminology** ..... **43**

**Revision History** ..... **47**

## Section 1 Overview

**1-1 Introduction and Features** ..... **1-2**
1-1-1 Overview of Safety Control Unit ..... 1-2
1-1-2 Features of Safety Control Unit ..... 1-5

**1-2 System Configuration and Configuration Devices** ..... **1-8**
1-2-1 Safety Control System Configuration on NX bus of CPU Unit or EtherCAT Network ..... 1-8
1-2-2 Safety Control System Configuration on EtherNet/IP Network ..... 1-10

1-2-3 Standalone Safety Control System Configuration ..... 1-10
1-2-4 Types of Safety Control Unit ..... 1-11

**1-3 Support Software ..... 1-12**
1-3-1 Applicable Support Software ..... 1-12
1-3-2 Connection Method and Procedures for EtherCAT Coupler Units ..... 1-12
1-3-3 Connection Method and Procedures for EtherNet/IP Coupler Units ..... 1-14

**1-4 Commissioning Procedures ..... 1-15**
1-4-1 Overall Procedure ..... 1-15
1-4-2 Detailed Procedures ..... 1-16

## Section 2 Specifications

**2-1 Safety CPU Unit ..... 2-2**
2-1-1 Models and Specifications ..... 2-2
2-1-2 Part Names and Functions ..... 2-8
2-1-3 Indicators ..... 2-10

**2-2 Safety Input Unit ..... 2-17**
2-2-1 Models and Specifications ..... 2-17
2-2-2 Part Names and Functions ..... 2-23
2-2-3 Indicators ..... 2-25

**2-3 Safety Output Unit ..... 2-30**
2-3-1 Models and Specifications ..... 2-30
2-3-2 Part Names and Functions ..... 2-37
2-3-3 Indicators ..... 2-40

**2-4 PFH ..... 2-44**

## Section 3 Installation and Wiring

**3-1 Installing Units ..... 3-2**
3-1-1 Installing NX Units ..... 3-2
3-1-2 Attaching Markers ..... 3-4
3-1-3 Removing Units ..... 3-5
3-1-4 Installation Orientation ..... 3-6

**3-2 Power Supply Types and Wiring ..... 3-8**
3-2-1 Applications of I/O Power Supply and Supply Methods ..... 3-8
3-2-2 Calculating the Total Current Consumption from I/O Power Supply ..... 3-9

**3-3 Wiring the Terminals ..... 3-10**
3-3-1 Wiring to the Screwless Clamping Terminal Blocks ..... 3-10
3-3-2 Checking Wiring ..... 3-23

## Section 4 Safety Control Unit Operation

**4-1 Relationship between Units and Types of Communications ..... 4-2**
4-1-1 Safety Control System on CPU Rack or EtherCAT Network ..... 4-2
4-1-2 Safety Control System on EtherNet/IP Network ..... 4-4
4-1-3 Standalone Safety Control System ..... 4-6

**4-2 I/O System ..... 4-8**
4-2-1 Relationship between the Types of Signals and the Types of Communications ..... 4-8
4-2-2 Safety Data Types and Standard Data Types ..... 4-8
4-2-3 Specifying Safety Data Types and Standard Data Types ..... 4-9

**4-3 Safety I/O Function ..... 4-10**
4-3-1 Safety Input Function ..... 4-10
4-3-2 Safety Output Function ..... 4-32

## Section 5 System Configuration and Setup


**5-1 Configuration and Setup Procedures ..... 5-2**

**5-2 Part Names and Functions of the Sysmac Studio Window ..... 5-3**

**5-3 Controller Configuration and Setup of the Safety Control Units ..... 5-5**
5-3-1 Procedures for Creating the Controller Configuration for Safety Control ..... 5-6
5-3-2 Setting and Viewing the Safety Control Unit Settings ..... 5-9
5-3-3 Procedure to Change the Model of the Safety Control Unit ..... 5-10

**5-4 Setting Up the Safety Process Data Communications ..... 5-12**
5-4-1 Setting Up the FSoE Communications ..... 5-12
5-4-2 Setting Up the CIP Safety Communications ..... 5-13
5-4-3 Calculating the Number of Connections ..... 5-37

**5-5 Safety I/O Functions ..... 5-40**

**5-6 Registering Device Variables ..... 5-43**

**5-7 Exposing Variables to Standard Controllers ..... 5-49**
5-7-1 Exposing Global Variables ..... 5-49
5-7-2 Setting Exposed Variables ..... 5-50
5-7-3 Safety CPU Unit Status ..... 5-54
5-7-4 I/O Ports for Safety I/O Units ..... 5-55
5-7-5 I/O Refreshing Method ..... 5-55

**5-8 Setting Standard Process Data Communications ..... 5-57**
5-8-1 Using an CPU Unit or EtherCAT Coupler Unit ..... 5-57
5-8-2 Using an EtherNet/IP Coupler Unit ..... 5-57

**5-9 Exporting/Importing Settings Data ..... 5-59**
5-9-1 Exporting/Importing the Settings for the Entire Slave Terminal ..... 5-59
5-9-2 Exporting/Importing the Settings for All the NX Units on a CPU Rack or Slave Terminal ..... 5-61
5-9-3 Exporting/Importing Data for Individual Safety CPU Unit ..... 5-62
5-9-4 Importing the Safety Unit Restore File ..... 5-63

**5-10 Offline Comparison ..... 5-65**
5-10-1 Procedure for Offline Comparison ..... 5-65
5-10-2 Checking the Comparison Results ..... 5-66
5-10-3 Detailed Comparison ..... 5-67
5-10-4 Target Data of Offline Comparison ..... 5-69


## Section 6 Programming


**6-1 POUs (Program Organization Units) ..... 6-3**
6-1-1 POU ..... 6-3
6-1-2 Overview of the Three Types of POUs ..... 6-3
6-1-3 Differences between Programs, Functions, and Function Blocks ..... 6-4
6-1-4 Details on Programs ..... 6-5
6-1-5 Details on Function Blocks ..... 6-6
6-1-6 Details on Functions ..... 6-10
6-1-7 Instructions ..... 6-11

**6-2 Variables ..... 6-12**
6-2-1 Variables ..... 6-12
6-2-2 Types of Variables ..... 6-12
6-2-3 Type of User-defined Variable ..... 6-12
6-2-4 Attributes of Variables ..... 6-13
6-2-5 Data Type ..... 6-14
6-2-6 Variable Attributes Other Than Data Type ..... 6-16
6-2-7 Function Block Instances ..... 6-17
6-2-8 Restrictions on Variable Names and Other Safety Program-related Names ..... 6-17

**6-3 Constants (Literals) ..... 6-19**
6-3-1 Constants ..... 6-19
6-3-2 Types of Constants ..... 6-19

**6-4 Programming Languages....6-21**
6-4-1 Programming Languages....6-21
6-4-2 FBD Language....6-21

**6-5 Programming Operations....6-26**
6-5-1 Programming Layer on the Sysmac Studio....6-26
6-5-2 Registering POUs....6-27
6-5-3 Registering Variables....6-37
6-5-4 FBD Programming....6-45
6-5-5 Program Pattern Copy....6-66
6-5-6 Function Block Conversion for Programs....6-69
6-5-7 Building....6-70
6-5-8 Searching and Replacing....6-72
6-5-9 Safety Task Settings....6-75
6-5-10 Variable Comment Switching Function....6-77

**6-6 Automatic Programming....6-86**
6-6-1 Generation Algorithms for Automatic Programming....6-86
6-6-2 Automatic Programming Settings....6-89
6-6-3 Automatic Programming Execution Procedure....6-92

**6-7 Monitoring Memory Usage for Safety Control Unit....6-96**

**6-8 Offline Debugging....6-98**
6-8-1 Offline Safety Program Debugging....6-98
6-8-2 Monitoring....6-101
6-8-3 Controlling BOOL Variables, Changing Present Values, and Using Forced Refreshing....6-101
6-8-4 Cross References....6-101
6-8-5 Setting the Initial Values of Variables....6-101
6-8-6 Feedback Settings....6-102
6-8-7 Simple Automatic Test....6-103

## Section 7 Checking Operation and Actual Operation

**7-1 Procedures before Operation and Transferring the Required Data....7-3**
7-1-1 Commissioning Procedure....7-3
7-1-2 Data That You Must Transfer before Operation and Data Transfer Procedures....7-4

**7-2 Transferring the Configuration Information....7-7**
7-2-1 Overview....7-7
7-2-2 Transfer Procedure for a Connection to NJ/NX-series CPU Unit....7-8
7-2-3 Transfer Procedure for a Connection to Communications Coupler Unit....7-8

**7-3 Operating Modes of the Safety CPU Unit....7-11**
7-3-1 Startup Operating Mode and Changing the Operating Mode....7-11
7-3-2 Operation When Changing Operating Mode....7-13
7-3-3 Executable Functions in Each Mode of the Safety CPU Unit....7-14

**7-4 Changing to DEBUG Mode....7-17**

**7-5 Checking External Device Wiring....7-20**
7-5-1 Overview of Functions for Checking Wiring....7-20
7-5-2 Monitoring Safety I/O Units....7-20
7-5-3 Troubleshooting Safety I/O Terminals....7-24
7-5-4 Clear All Memory Operation for Safety I/O Units....7-25

**7-6 Functions for Checking Operation....7-28**
7-6-1 Overview of Functions for Checking Operation....7-28
7-6-2 Starting and Stopping the Safety Programs in DEBUG Mode....7-28
7-6-3 Monitoring Variables in the FBD Editor....7-29
7-6-4 Monitoring Variables in a Watch Tab Page....7-30
7-6-5 Controlling BOOL Variables, Changing Present Values, and Using Forced Refreshing....7-32
7-6-6 Cross References....7-40

**7-7 Online Functional Test....7-44**
7-7-1 Online Functional Test Settings....7-44
7-7-2 Online Functional Test Execution Procedure....7-48

**7-8 Search FB Where Safety Output Is OFF ..... 7-54**
7-8-1 Procedure for Operating Search FB Where Safety Output Is OFF ..... 7-54
7-8-2 Editing Function Blocks to be Searched for ..... 7-55
7-8-3 Registering a Data Type in the Function Block Search Settings Window ..... 7-56

**7-9 Node Name ..... 7-58**

**7-10 Security Settings ..... 7-59**
7-10-1 Setting the Safety Password ..... 7-59
7-10-2 Data Protection ..... 7-60

**7-11 Performing Safety Validation and Operation ..... 7-65**
7-11-1 Performing Safety Validation ..... 7-65
7-11-2 Changing to RUN Mode ..... 7-67
7-11-3 Changing to PROGRAM Mode ..... 7-68

**7-12 Starting and Stopping the Safety Application Monitoring ..... 7-70**
7-12-1 Procedure to Start and Stop the Safety Application Monitoring ..... 7-70
7-12-2 Changing the Monitoring Options for the Safety Application ..... 7-71

**7-13 Uploading Configuration Information and Safety Application Data ..... 7-72**
7-13-1 Outline ..... 7-72
7-13-2 Uploading Data for a Connection to an NJ/NX-series CPU Unit ..... 7-72
7-13-3 Uploading Data for a Connection to a Communications Coupler Unit ..... 7-73

**7-14 Transferring Safety Application Data ..... 7-75**

**7-15 Monitoring Controller Status ..... 7-76**

**7-16 Restarting and Clearing All Memory ..... 7-78**
7-16-1 Restarting ..... 7-78
7-16-2 Clear All Memory Operation ..... 7-79

## Section 8 Calculating Safety Reaction Times

**8-1 Safety Reaction Time ..... 8-2**
8-1-1 Calculating the Safety Reaction Time ..... 8-2
8-1-2 Verifying Safety Reaction Times ..... 8-4

**8-2 Safety Task ..... 8-5**
8-2-1 Safety Task ..... 8-5
8-2-2 Operation of Safety Task ..... 8-5
8-2-3 Minimum Safety Task Period ..... 8-5
8-2-4 Setting the Safety Task Period ..... 8-7

**8-3 FSoE Watchdog Timer ..... 8-9**
8-3-1 FSoE Watchdog Timers ..... 8-9
8-3-2 Checking FSoE Watchdog Timers ..... 8-9
8-3-3 Settings for FSoE Watchdog Timers ..... 8-10
8-3-4 Changing FSoE Watchdog Timers ..... 8-10

**8-4 EPI (Data Packet Interval) ..... 8-11**
8-4-1 Changing the EPI ..... 8-11
8-4-2 EPI Restrictions ..... 8-11

## Section 9 Communications Load

**9-1 Adjusting the Communications Load ..... 9-2**
9-1-1 Checking Bandwidth Usage for Tag Data Links ..... 9-4
9-1-2 Checking the Device Bandwidth Usage of the CIP Safety Routing ..... 9-6
9-1-3 Relationship between the Number of Packets Used per Second and Packet Intervals ..... 9-6
9-1-4 Adjusting the Device Bandwidth Usage ..... 9-7

## Section 10 Safety Unit Restore

**10-1 Safety Unit Restore ..... 10-2**
10-1-1 Generate Safety Unit Restore File Function ..... 10-2
10-1-2 Safety Unit Restore Function ..... 10-4
10-1-3 Specifications of a Safety Unit Restore File ..... 10-6
**10-2 Versions for Which Safety Unit Restore Can Be Used ..... 10-7**

## Section 11 Safety Data Logging

**11-1 Outline of the Safety Data Logging Function ..... 11-2**
**11-2 Creating a Safety Data Logging Settings File with the Sysmac Studio ..... 11-4**
**11-3 Safety Data Logging Operation Procedure ..... 11-6**
**11-4 Checking the Logging Status ..... 11-7**
11-4-1 Checking the Seven-segment Indicator ..... 11-7
11-4-2 Checking with System-defined Variables ..... 11-7
**11-5 Log File Specifications ..... 11-8**
**11-6 Versions for Which Safety Data Logging Can Be Used ..... 11-10**

## Section 12 Troubleshooting

**12-1 How to Check for Errors ..... 12-2**
**12-2 Checking for Errors with the Indicators ..... 12-3**
12-2-1 Troubleshooting the Main Errors in the Safety CPU Unit ..... 12-3
12-2-2 Troubleshooting the Main Errors in the Safety I/O Units ..... 12-11
**12-3 Checking for Errors with the Sysmac Studio ..... 12-14**
12-3-1 Checking for Errors from the Sysmac Studio ..... 12-14
12-3-2 Types of Errors ..... 12-15
12-3-3 Event Codes for Errors and Troubleshooting Procedures ..... 12-17
12-3-4 Error Descriptions ..... 12-33
**12-4 Checking Errors of CIP Safety Target Devices Using Sysmac Studio ..... 12-93**
12-4-1 Troubleshooting the CIP Safety Target Device Errors ..... 12-93
12-4-2 CIP Safety Connection Status Codes and Troubleshooting ..... 12-98
**12-5 Resetting Errors ..... 12-102**
**12-6 Troubleshooting Flow When Errors Occur ..... 12-103**

## Section 13 Inspection and Maintenance

**13-1 Cleaning and Inspection ..... 13-2**
13-1-1 Cleaning ..... 13-2
13-1-2 Periodic Inspections ..... 13-2
**13-2 Maintenance Procedures ..... 13-5**
13-2-1 Replacing the Safety CPU Unit ..... 13-5
13-2-2 Replacing Safety I/O Units ..... 13-7

## Appendices

**A-1 Dimensions ..... A-3**
A-1-1 Safety CPU Unit ..... A-3
A-1-2 Safety I/O Units ..... A-4

**A-2 NX Objects ..... A-5**
A-2-1 Format of NX Object Descriptions ..... A-5
A-2-2 Safety CPU Unit (NX-SL3300/NX-SL3500) ..... A-5
A-2-3 Safety CPU Unit (NX-SL5500/NX-SL5700) ..... A-7
A-2-4 NX-SID800 Safety Input Unit ..... A-11
A-2-5 NX-SIH400 Safety Input Unit ..... A-16
A-2-6 NX-SOD400 Safety Output Unit ..... A-21
A-2-7 NX-SOH200 Safety Output Unit ..... A-25

**A-3 Application Examples ..... A-30**
A-3-1 Emergency Stop Pushbutton Switches ..... A-30
A-3-2 Safety Doors ..... A-32
A-3-3 Safety Laser Scanners ..... A-36
A-3-4 Safety Door Switches with Magnetic Locks and Key Selector Switches ..... A-39
A-3-5 Enable Switches ..... A-43
A-3-6 Two-hand Switches ..... A-47
A-3-7 D40A Non-contact Door Switches ..... A-50
A-3-8 D40A-2/D40Z Non-contact Door Switches ..... A-53
A-3-9 Safety Mats and Safety Light Curtains ..... A-56
A-3-10 Safety Edges ..... A-61
A-3-11 Single Beam Safety Sensor ..... A-63
A-3-12 Multi-axis Servo Drives (STO) ..... A-66

**A-4 Change Tracking ..... A-70**

**A-5 Safety CPU Unit Status ..... A-72**
A-5-1 NX-SL3300/NX-SL3500 ..... A-72
A-5-2 NX-SL5500/NX-SL5700 ..... A-72

**A-6 I/O Ports of Safety I/O Units ..... A-74**
A-6-1 NX-SIH400 Safety Input Unit ..... A-74
A-6-2 NX-SID800 Safety Input Unit ..... A-76
A-6-3 NX-SOH200 Safety Output Unit ..... A-77
A-6-4 NX-SOD400 Safety Output Unit ..... A-78

**A-7 CIP Response Codes ..... A-81**
A-7-1 General Status Codes ..... A-81
A-7-2 Extended Status Codes ..... A-83

**A-8 Icon list for Safety Slave Unit Parameters ..... A-86**
A-8-1 External Device Icons for Input Devices ..... A-86
A-8-2 Contact Icons for Input Devices ..... A-90
A-8-3 External Device Icons for Output Devices ..... A-92
A-8-4 Contact Icons for Output Devices ..... A-93

**A-9 Calculating I/O Sizes for CPU Rack or Slave Terminals ..... A-94**

**A-10 Printing ..... A-95**
A-10-1 Selecting the Items to Print ..... A-95
A-10-2 Items that are Printed ..... A-95

**A-11 List of Screwless Clamping Terminal Block Models ..... A-97**
A-11-1 Model Notation ..... A-97
A-11-2 List of Terminal Block Models ..... A-97

**A-12 I/O Response Times for Communications between NX Units on EtherNet/IP Slave Terminals ..... A-99**

**A-13 Units That Support Communications between NX Units ..... A-101**

**A-14 Checking the Signature Code on the Seven-segment Indicator ..... A-102**

**A-15 Execution Scenarios for the Simple Automatic Test ..... A-103**

**A-16 Differences in Checking Operation between the Simulator and Safety CPU Unit ..... A-106**

**A-17 I/O Data Enable Flag for CIP Safety Connections ..... A-107**

**A-18 Safety: Update Configurations and Setup Transfer Data ..... A-109**

**A-19 CIP Safety Target Device Operation ..... A-110**

**A-20 Version Information ..... A-113**

A-20-1 Relationship between Unit Versions and Sysmac Studio Versions ..... A-113
A-20-2 Functions That Were Added or Changed for Each Version ..... A-118

# Index

# Relevant Manuals

The following table provides the relevant manuals for this product. Read all of the manuals that are relevant to your system configuration and application before you use the product.
Most operations on this product are performed from the Sysmac Studio Automation Software. For details on the Sysmac Studio, refer to the *Sysmac Studio Version 1 Operation Manual (Cat. No. W504)*.

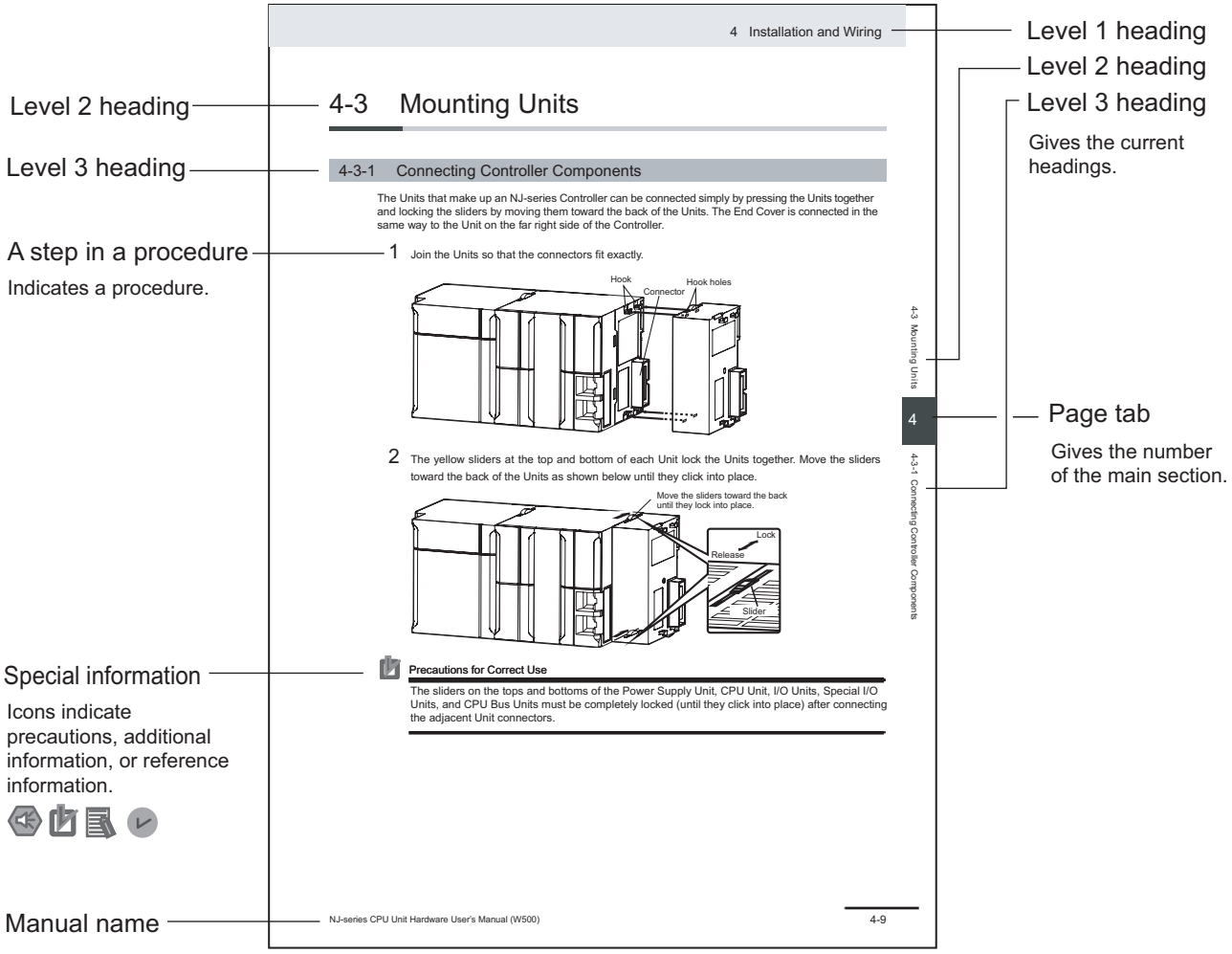
| Purpose of use | NX-series Safety Control Unit User's Manual | NX-series Safety Control Unit / Communication Control Unit User's Manual | NX-series Communication Control Unit Built-in Function User's Manual | NX-series Safety Control Unit Instructions Reference Manual |
|---|---|---|---|---|
| Building a safety control system integrated with NJ/NX-series CPU Units | ● | | | |
| Building a standalone safety control system with EtherNet/IP Coupler Units | ● | | | |
| Building a safety network control system with Communication Control Units | | ● | | |
| Introduction to Safety Control Unit | ● | ● | | |
| Setting devices and hardware | | | | |
| NX-SL5□□□ Safety CPU Unit | ● | ● | | |
| NX-SL3□□□ Safety CPU Unit | ● | | | |
| NX-SI□□□□ and NX-SO□□□□ Safety I/O Units | ● | ● | | |
| NX-CSG□□□ Communication Control Unit | | ● | | |
| Software settings | | | | |
| NX-SL5□□□ Safety CPU Unit | ● | ● | | |
| NX-SL3□□□ Safety CPU Unit | ● | | | |
| NX-SI□□□□ and NX-SO□□□□ Safety I/O Units | ● | ● | | |
| NX-CSG□□□ Communication Control Unit | | ● | ● | |
| Creating safety programs | ● | ● | | ● |
| Testing operation and debugging | | | | |
| Safety programs | ● | ● | | ● |
| Safety process data communications | ● | ● | | |
| Safety I/O functions | ● | ● | | |
| Tag data links | | ● | | |
| Built-in functions for Communication Control Unit | | ● | ● | |
| Learning about error corrections | | | | |
| NX-SL5□□□ Safety CPU Unit | ● | ● | | ● |
| NX-SL3□□□ Safety CPU Unit | ● | | | ● |
| NX-SI□□□□ and NX-SO□□□□ Safety I/O Units | ● | ● | | |
| NX-CSG□□□ Communication Control Unit | | ● | | |
| Maintenance | | | | |

| Purpose of use | | NX-series Safety Control Unit User's Manual | NX-series Safety Control Unit / Communication Control Unit User's Manual | NX-series Communication Control Unit Built-in Function User's Manual | NX-series Safety Control Unit Instructions Reference Manual |
|---|---|---|---|---|---|
| | NX-SL5□□□ Safety CPU Unit | ● | ● | | |
| | NX-SL3□□□ Safety CPU Unit | ● | | | |
| | NX-SI□□□□ and NX-SO□□□□ Safety I/O Units | ● | ● | | |
| | NX-CSG□□□ Communication Control Unit | | ● | | |

# Manual Structure

## 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**

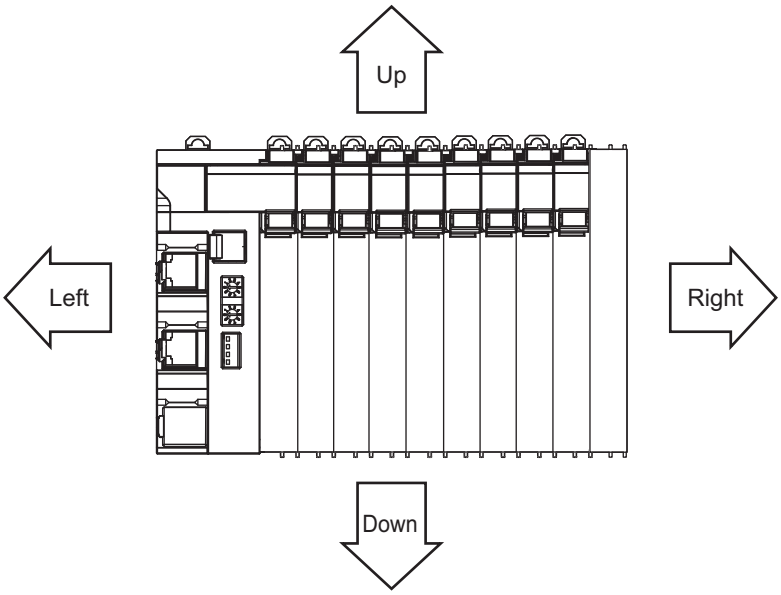
Additional information to read as required.
This information is provided to increase understanding or make operation easier.

**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.

## Precaution on Terminology

In this manual, the directions in relation to the Units are given in the following figure, which shows upright installation.



# 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 https://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.

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

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

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

## Application Considerations

### Suitability of Use

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

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

### Programmable Products

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

## Disclaimers

## Performance Data

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

## Change in Specifications

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

## Errors and Omissions

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

## Statement of security responsibilities for assumed use cases and against threats

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

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

# Safety Precautions

## Definition of Precautionary Information

The following notation is used in this manual to provide precautions required to ensure safe usage of the NX-series Safety Control Unit. The safety precautions that are provided are extremely important to safety. Always read and heed the information provided in all safety precautions.
The following notation is used.

| | |
|---|---|
| ⚠ WARNING | Indicates a potentially hazardous situation which, if not avoided, could result in death or serious injury. Additionally, there may be severe property damage. |
| ⚠ Caution | Indicates a potentially hazardous situation which, if not avoided, may result in minor or moderate injury, or property damage. |

## Symbols

| | |
|---|---|
|  | The circle and slash symbol indicates operations that you must not do.<br>The specific operation is shown in the circle and explained in text.<br>This example indicates prohibiting disassembly. |
|  | The triangle symbol indicates precautions (including warnings).<br>The specific operation is shown in the triangle and explained in text.<br>This example indicates a precaution for electric shock. |
|  | The triangle symbol indicates precautions (including warnings).<br>The specific operation is shown in the triangle and explained in text.<br>This example indicates a general precaution. |
|  | The filled circle symbol indicates operations that you must do.<br>The specific operation is shown in the circle and explained in text.<br>This example shows a general precaution for something that you must do. |

## Warnings

Serious injury may possibly occur due to loss of required safety functions.
When building the system, observe the following warnings to ensure the integrity of the safety-related components.

**⚠ WARNING**

### Setting Up a Risk Assessment System

The process of selecting these products should include the development and execution of a risk assessment system early in the design development stage to help identify potential dangers in your equipment and optimize safety product selection.
Related International Standards:
- ISO 12100 General Principles for Design - Risk Assessment and Risk Reduction

### Protective Measure

When developing a safety system for the equipment and devices that use safety products, make every effort to understand and conform to the entire series of international and industry standards available, such as the examples given below.
Related International Standards:
- ISO 12100 General Principles for Design - Risk Assessment and Risk Reduction
- IEC 60204-1 Electrical Equipment of Machines - Part 1: General Requirements
- ISO 13849-1, -2 Safety-related Parts of Control Systems
- ISO 14119 Interlocking Devices Associated with Guards - Principles for Design and Selection
- IEC 62046 Application of Protective Equipment to Detect the Presence of Persons
- IEC 62061 Functional Safety of Safety-related Electrical, Electronic and Programmable Electronic Control Systems
- IEC 61508 Functional Safety of Electrical/Electronic/Programmable Electronic Safety-related Systems

### Role of Safety Products

Safety products incorporate standardized safety functions and mechanisms, but the benefits of these functions and mechanisms are designed to attain their full potential only within properly designed safety-related systems. Make sure you fully understand all functions and mechanisms, and use that understanding to develop systems that will ensure optimal usage.
Related International Standards:
- ISO 14119 Interlocking Devices Associated with Guards - Principles for Design and Selection
- ISO 13857 Safety Distances to Prevent Hazard Zones being Reached by Upper and Lower Limbs

## Installing Safety Products

Qualified engineers must develop your safety-related system and install safety products in devices and equipment. Prior to machine commissioning verify through testing that the safety products works as expected.
Related International Standards:

- ISO 12100 General Principles for Design - Risk Assessment and Risk Reduction
- IEC 60204-1 Electrical Equipment of Machines - Part 1: General Requirements
- ISO 13849-1, -2 Safety-related Parts of Control Systems
- ISO 14119 Interlocking Devices Associated with Guards - Principles for Design and Selection
- IEC 62061 Functional Safety of Safety-related Electrical, Electronic and Programmable Electronic Control Systems
- IEC 61508 Functional Safety of Electrical/Electronic/Programmable Electronic Safety-related Systems



## Observing Laws and Regulations

Safety products must conform to pertinent laws, regulations, and standards. Make sure that they are installed and used in accordance with the laws, regulations, and standards of the country where the devices and equipment incorporating these products are distributed.



## Observing Usage Precautions

Carefully read the specifications and precautions as well as all items in the Instruction Manual for your safety product to learn appropriate usage procedures. Any deviation from instructions will lead to unexpected device or equipment failure not anticipated by the safety-related system.



## Transferring Devices and Equipment

When transferring devices and equipment, be sure to retain one copy of the Instruction Manual and supply another copy with the device or equipment so the person receiving it will have no problems with operation and maintenance.
Related International Standards:

- ISO 12100 General Principles for Design - Risk Assessment and Risk Reduction
- IEC 60204-1 Electrical Equipment of Machines - Part 1: General Requirements
- ISO 13849-1, -2 Safety-related Parts of Control Systems
- IEC 62061 Functional Safety of Safety-related Electrical, Electronic and Programmable Electronic Control Systems
- IEC 61508 Functional Safety of Electrical/Electronic/Programmable Electronic Safety-related Systems



## Design

Confirm that the calculated reaction times meet the required specifications for all safety chains.
Serious injury may possibly occur due to loss of required safety functions.



All safety devices and components that are connected to an NX-series Safety Control Unit must be selected and used to meet the required level of safety and the relevant safety category.
Serious injury may possibly occur due to loss of required safety functions.



Do not use indicators on the NX-series Safety Control Units for safety operations.
Serious injury may possibly occur due to loss of required safety functions.



Check during the import of the program that the CRC of the program is correct.
Serious injury may possibly occur due to loss of required safety functions.



Check during the import of the user defined function that the CRC of the imported function block is correct.
Serious injury may possibly occur due to loss of required safety functions.



Do not use non-safety signals, including tag data links, explicit messages, and exposed variables, as safety signals.
Serious injury may possibly occur due to loss of required safety functions.



If you select “Open Only” for the Open Type setting, make sure to verify that the originator/target have correct configurations.
Serious injury may possibly occur due to loss of required safety functions.



Before connecting an NX Series Safety Control Unit to the network, clear the previous settings.
Serious injury may possibly occur due to loss of required safety functions.



## Debugging

Before you perform safety validation of the safety programs, complete debugging of the safety programs.
Otherwise, the Safety CPU Unit will start with safety programs that are not fully debugged and may cause serious personal injury.



Make sure that the area around the system is safe before you change the operating mode, change present values, or execute forced refreshing. The outputs may operate and may cause serious injury.



Make sure that the area around the system is safe before you start the system operation while the online functional test is in progress. The outputs may operate and may cause serious injury.



## Testing Operation

Before you start the system, perform user testing to make sure that all safety devices operate correctly. The safety signature is validated upon completion of the user testing. Serious injury may possibly occur due to loss of required safety functions.



After you perform safety validation, check items for safety validation printed out to confirm Safety Control Units are correctly configured.



Although the Simulator and Simple Automatic Test simulate the operation of the Safety CPU Unit, there are differences from the Safety CPU Unit in operation and timing. Always confirm operation on the actual equipment before you operate the equipment. Accidents may occur if the controlled system performs unexpected operation.



## Wiring

Wire the safety input and output lines so that they do not touch other lines.
Serious injury may possibly occur due to loss of required safety functions.



Wire the Safety Control Unit properly so that 24-VDC lines do not touch output lines accidentally or unintentionally.
Serious injury may possibly occur due to loss of required safety functions.



Wire the safety output lines and 24-VDC lines so that ground faults will not cause the loads to turn ON.
Serious injury may possibly occur due to loss of required safety functions.



The wiring information that is displayed on the Sysmac Studio is for reference only and may differ from the actual wiring diagrams. Always confirm the actual wiring and performing suitable wiring.
Serious injury may possibly occur due to loss of required safety functions.



## During Power Supply

Do not touch any of the terminals or terminal blocks while the power is being supplied.
Doing so may result in electric shock.



Do not attempt to take any Unit apart.
In particular, high-voltage parts are present in Units that supply power while power is supplied or immediately after power is turned OFF. Touching any of these parts may result in electric shock.
There are sharp parts inside the Unit that may cause injury.



## Replacing Units

When replacing a Safety Control Unit, confirm that the model of the Unit is correct, confirm that the Unit and terminal block mounting positions are correct, configure the replacement Unit suitably, and confirm that the Unit operates correctly.



## Voltage and Current Inputs

Make sure that the voltages and currents that are input to the Units and slaves are within the specified ranges.
Inputting voltages or currents that are outside of the specified ranges may cause accidents or fire.



## Transferring

Always confirm safety at the destination before you transfer the unit configuration information, parameters, set values, or other data from tools such as the Sysmac Studio.
The devices or machines may perform unexpected operation regardless of the operating mode of the Controller.



## Restoring Data

When you restore the Safety CPU Unit, confirm that the safety signature for the restored programs is correct. Serious injury may possibly occur due to loss of required safety functions.



## Security Measures

**Anti-virus protection**
Install the latest commercial-quality antivirus software on the computer connected to the control system and maintain to keep the software up-to-date.



**Security measures to prevent unauthorized access**
Take the following measures to prevent unauthorized access to our products.

- Install physical controls so that only authorized personnel can access control systems and equipment.
- Reduce connections to control systems and equipment via networks to prevent access from untrusted devices.
- Install firewalls to shut down unused communications ports and limit communications hosts and isolate control systems and equipment from the IT network.
- Use a virtual private network (VPN) for remote access to control systems and equipment.
- Adopt multifactor authentication to devices with remote access to control systems and equipment.
- Set strong passwords and change them frequently.
- Scan virus to ensure safety of USB drives or other external storages before connecting them to control systems and equipment.



**Data input and output protection**
Validate backups and ranges to cope with unintentional modification of input/output data to control systems and equipment.

- Checking the scope of data
- Checking validity of backups and preparing data for restore in case of falsification and abnormalities
- Safety design, such as emergency shutdown and fail-soft operation in case of data tampering and abnormalities



| | |
|---|---|
| **Data recovery**<br>Backup data and keep the data up-to-date periodically to prepare for data loss. |  |
| When using an intranet environment through a global address, connecting to an unauthorized terminal such as a SCADA, HMI or to an unauthorized server may result in network security issues such as spoofing and tampering. You must take sufficient measures such as restricting access to the terminal, using a terminal equipped with a secure function, and locking the installation area by yourself. |  |
| When constructing an intranet, communication failure may occur due to cable disconnection or the influence of unauthorized network equipment. Take adequate measures, such as restricting physical access to network devices, by means such as locking the installation area. |  |
| When using a device equipped with the SD Memory Card function, there is a security risk that a third party may acquire, alter, or replace the files and data in the removable media by removing the removable media or unmounting the removable media. Please take sufficient measures, such as restricting physical access to the Controller or taking appropriate management measures for removable media, by means of locking the installation area, entrance management, etc., by yourself. |  |

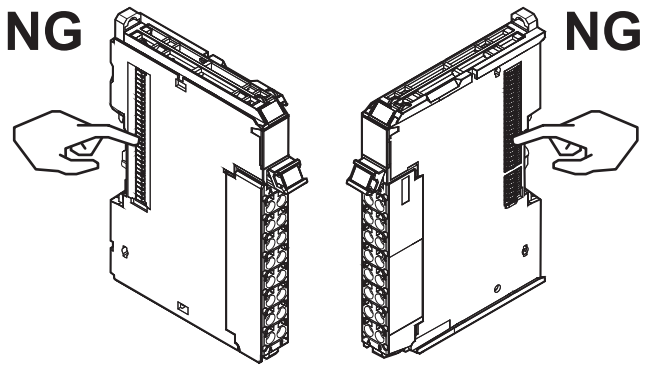
# Precautions for Safe Use

## Transporting

- Do not attempt to disassemble, repair, or modify any Units. Doing so may result in malfunction or fire.
- Do not drop any Unit or subject it to abnormal vibration or shock. Doing so may result in Unit malfunction or burning.
- When transporting any Unit, use the special packing box for it. Also, do not subject the Unit to excessive vibration or shock during transportation.

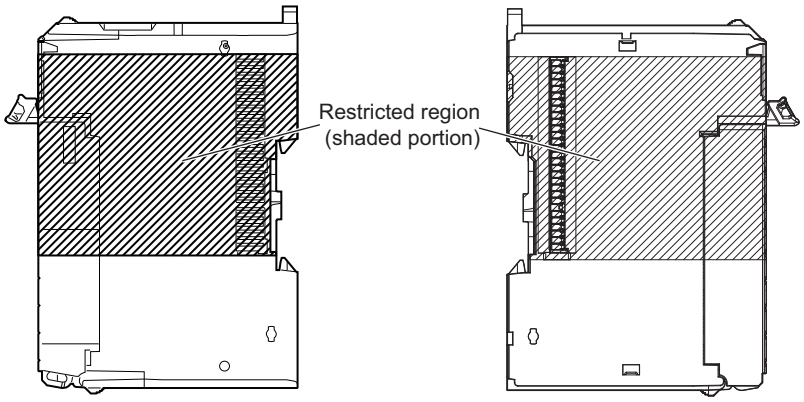
## Mounting

- Always turn OFF the power supply before mounting a Unit. If the power supply is not OFF, the Unit may malfunction or may be damaged.
- Mount terminal blocks and connectors only after checking the mounting location carefully. Be sure that the terminal blocks, expansion cables, and other items with locking devices are properly locked into place.
- Do not apply labels or tape to the Unit. When the Units are installed or removed, adhesive or scraps may adhere to the pins in the NX bus connector, which may result in malfunctions.
- Do not touch the pins in the NX bus connector on the Unit. Dirt may adhere to the pins in the NX bus connector, which may result in malfunctions.

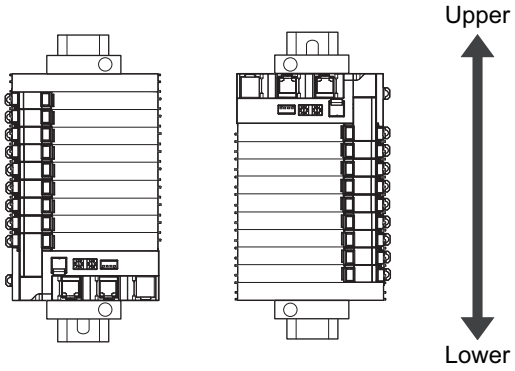



Example: NX Unit (12 mm width)

- Do not write on the an NX Unit with ink within the restricted region that is shown in the following figure. Also do not get this area dirty. When the Unit is installed or removed, ink or dirt may adhere to the pins in the NX bus connector, which may result in malfunctions in the Controller.
  Refer to the user's manual of the CPU Unit or Communications Coupler Unit to be connected for details on the restricted region of the CPU Unit or Communications Coupler Unit.



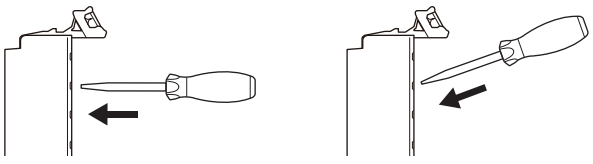

- For the installation orientations in the following figure, support the cables, e.g., with a duct, so that the End Plate on the bottom is not subjected to the weight of the cables. The weight of the cables may cause the bottom End Plate to slide downward so that the Slave Terminal is no longer secured to the DIN Track, which may result in malfunctions.




## Wiring

- Follow the instructions in this manual to correctly perform wiring.
- Double-check all switch settings to make sure that they are correct before turning ON the power supply.
- Use the correct wiring parts and tools when you wire the system. Otherwise, cables may be disconnected to cause short-circuit or wire breakage.
- Do not pull on the cables or bend the cables beyond their natural limit. Do not place any heavy objects on the cables or other wiring lines. Doing so may severe the cables.
- When wiring or installing the Units, do not allow metal fragments to enter the Units.
- Do not press the flat-blade screwdriver straight into the release hole on the screwless clamping terminal block. Doing so may damage the terminal block.

**NG** **OK**



- When you insert a flat-blade screwdriver into a release hole on the screwless clamping terminal block, press the screwdriver down with a force of 30 N or less. Applying excessive force may damage the terminal block.
- Do not tilt or twist the flat-blade screwdriver while it is pressed into the release hole on the screwless clamping terminal block. Doing so may damage the terminal block.

**NG** **NG**



## Power Supply Design

- Use the I/O power supply capacity within the range that is given in the Unit specifications.
- Provide suitable power supply capacity according to the reference manuals.
- Use the power supply voltage that is specified in the related manuals.
- Do not apply voltages that exceed the rated value to any Input Unit.

## Debugging

- With forced refreshing, the values of variables are overwritten with specified values and then the safety programs are executed. If forced refreshing is used for variables that give the results of program processing, the variables will first take the specified values, but they will then be overwritten by the safety program.
- Depending on the difference in the forced status, the control system may operate unexpectedly.
- After you clear the memory, the Controller operates in the same way as immediately after you create the system configuration with the Controller in the factory default condition.
- Verify that the safety communications with a remote node will be established in the debug mode of the Safety CPU Unit.

## Turning ON the Power Supply or Restarting after Safety Validation

- Remember that if safety validation is successful, the next time the Safety CPU Unit is started, it will automatically start in RUN mode.
- When you download the parameters for the NJ/NX-series CPU Units, the Communications Coupler Unit and NX Units, the Safety CPU Unit automatically restarts.

## Turning ON the Power Supply

- Double-check all wiring connections and switch settings to make sure that they are correct before turning ON the power supply. Use the correct wiring parts and tools when you wire the system.

- Make sure that the voltages and currents that are input to the Units and slaves are within the specified ranges. Inputting voltages or currents that are outside of the specified ranges may damage the Units or slaves or cause fire.

## Actual Operation

- The relevant Units will maintain the safe states for I/O data with safety connections after an error is detected in safety process data communications. However, when the cause of the error is removed, safety process data communications will recover automatically.
  If you need to prevent equipment from restarting when safety process data communications recover automatically, implement suitable restart conditions in the user program.

## Turning OFF the Power Supply

- Do not disconnect the cable or turn OFF the power supply to the Controller or a Slave Terminal when downloading data or the user program from the Support Software.
- Always turn OFF the external power supply to the Units before attempting any of the following.
  a) Mounting or removing an NX Unit.
  b) Assembling Units
  c) Setting DIP switches or rotary switches
  d) Connecting or wiring cables
  e) Attaching or removing terminal blocks or connectors
  The Power Supply Unit may continue to supply power to the Controller for a few seconds after the power supply turns OFF. The UNIT PWR and I/O PWR indicators are lit during this time. Confirm that the UNIT PWR and the I/O PWR indicators are not lit before you perform any of the above actions.

## General Communications

- Do not exceed the ranges that are given in the specifications for the communications distance and number of connected Units.
- Refer to the user's manual for the CPU Unit or Communications Coupler Unit for precautions for the safe use of communications with the connected CPU Unit or Communications Coupler Unit.

## Standards

- The customer is responsible for attaining conformance of the entire system to standards.

## Maintenance

- Test the functionality every six months to detect welded contactor contacts.
  To detect electrical and mechanical failures, use a combination of redundant semiconductor output contacts and redundant mechanical output devices.

## Unit Replacement

- After you replace the Safety Control Unit, set the program and all configuration settings that are necessary to resume operation. Make sure that the safety functions operate normally before you start actual operation.
- When you replace a Unit, start operation only after you transfer the settings and variables that are required for operation to the new Unit.

## Disposal

- Dispose of the product according to local ordinances as they apply.

# Precautions for Correct Use

## Storage and Installation

- Follow the instructions in this manual to correctly perform installation and wiring.
- Do not operate or store the Units in the following locations. Doing so may result in burning, in operation stopping, or in malfunction.
  a) Locations subject to direct sunlight
  b) Locations subject to temperatures or humidity outside the range specified in the specifications
  c) Locations subject to condensation as the result of severe changes in temperature
  d) Locations subject to corrosive or flammable gases
  e) Locations subject to dust (especially iron dust) or salts
  f) Locations subject to exposure to water, oil, or chemicals
  g) Locations subject to shock or vibration
  h) Locations subject to static electricity or other forms of noise
- Take appropriate and sufficient countermeasures when installing the Controller in the following locations.
  a) Locations subject to strong, high-frequency noise
  b) Locations subject to static electricity or other forms of noise
  c) Locations subject to strong electromagnetic fields
  d) Locations subject to possible exposure to radioactivity
  e) Locations close to power lines
- Before touching a Unit, be sure to first touch a grounded metallic object in order to discharge any static build-up.
- Use the rated power supply voltage for the Units that supply power. Take appropriate measures to ensure that the specified power with the rated voltage and frequency is supplied in places where the power supply is unstable.

## Wiring

- Do not allow foreign matter to enter the openings in the Unit. Doing so may result in Unit burning, electric shock, or failure.

## Actual Operation

- Make sure that you are connected to the correct Safety CPU Unit before you perform any online operations with the Safety CPU Unit.
- Before you transfer safety application data to the Safety CPU Unit, check the safety signature and make sure the data is the intended data.
- Always confirm the destination before you transfer configuration information and safety application data from the Sysmac Studio.
- You cannot monitor or perform certain online operations with the same Safety CPU Unit from more than one copy of the Sysmac Studio at the same time.

## Turning OFF the Power Supply

- Do not turn OFF the power supply while data is being transferred.

## General Communications

- Refer to the user's manual for the CPU Unit, EtherNet/IP Unit, or Communications Coupler Unit for precautions for the correct use of communications with the connected CPU Unit, EtherNet/IP Unit, or Communications Coupler Unit.

## Debugging

- The task period affects the safety response performance. If the task period changes due to changes in the configuration or programs, recalculate the safety reaction times.
- If you change the I/O for a variable to publish to a NJ/NX-series CPU Unit, the device variable assignments to the NJ/NX-series CPU Unit will be canceled. In this case, you need to assign the device variables, and then transfer the settings and programs to the NJ/NX-series CPU Unit. After you transfer the settings and programs, check that the operation of the NJ/NX-series CPU Unit is correct.
- If you change the exposed variable in the EtherNet/IP Slave Terminal configuration, the I/O allocations of the EtherNet/IP Couplers Unit will be changed. Check that the operation of the EtherNet/IP Coupler Unit is correct.
- For security purposes, we recommend that you set a password for the Safety CPU Unit and the project file. To avoid the leakage, keep the passwords under strict control. Especially, when you transmit password data through the Internet, implement a measure to secure the transmission such as by using the public key encryption.
- If you lose the password that is set to the Safety CPU Unit, you will no longer be able to make changes to the Safety CPU Unit. Take caution not to lose the password. If you want to reset the configured password, contact your OMRON representative.
- For safety data logging, make sure to use the settings file generated from the same project file as the logging target.
- After editing the CIP Safety connection settings, you need to transfer the NJ/NX-series CPU Unit settings and programs. Once the settings and programs are transferred, check to see if the NJ/NX-series CPU Unit runs normally.
- After editing the SRA parameters for the safety servo drive, you need to transfer the NJ/NX-series CPU Unit settings and programs. Once the settings and programs are transferred, check to see if the NJ/NX-series CPU Unit runs normally.
- After changing the unit conversion settings for the axis to which the safety servo drive is assigned, make sure that the position and speed settings for the safety function of the safety servo drive are correct.

## Periodic Inspections and Maintenance

- Do not disassemble, repair, or modify the Safety Control Unit. Doing so may lead to loss of safety functions.

## Disposal

- Be careful not to injure yourself when dismantling the Safety Control Unit.

# Regulations and Standards

The NX-series Safety Control Units are certified for the following standards.

- Safety CPU Unit NX-SL5500 / NX-SL5700

| Certification body | Standards |
|---|---|
| TÜV Rheinland*1 | • EN ISO 13849-1: 2015<br>• EN ISO 13849-2: 2012<br>• IEC 61508 parts 1-7: 2010<br>• IEC/EN 61131-2: 2007<br>• IEC 61326-3-1: 2017<br>• IEC 61131-6: 2012 |
| UL | NRAG (UL61010-1, UL61010-2-201, and UL121201)<br>NRAG7 (CSA C22.2 No.61010-1, CSA C22.2 No.61010-2-201, and CSA C22.2 No.213)<br>FSPC (IEC 61508 and ISO 13849) |

*1. The FSoE protocol was certified for applications in which OMRON FSoE devices are connected to each other.
For compatibility with FSoE devices other than OMRON FSoE devices, the customer must validate FSoE communications.

- Safety CPU Unit NX-SL3300 / NX-SL3500 and Safety I/O Unit NX-SID800 / NX-SIH400 / NX-SOD400 / NX-SOH200

| Certification body | Standards |
|---|---|
| TÜV Rheinland*1 | • EN ISO 13849-1: 2015<br>• EN ISO 13849-2: 2012<br>• IEC 61508 parts 1-7: 2010<br>• IEC/EN 61131-2: 2007<br>• IEC 61326-3-1: 2017 |
| UL | NRAG (UL508 and ANSI/ISA 12.12.01)<br>NRAG7 (CSA C22.2 No.142 and CSA C22.2 No.213) |

*1. The FSoE protocol was certified for applications in which OMRON FSoE devices are connected to each other.
For compatibility with FSoE devices other than OMRON FSoE devices, the customer must validate FSoE communications.

The NX-series Safety Control Units allow you to build a safety control system that meets the following standards.

- Requirements for SIL 3 (Safety Integrity Level 3) in IEC 61508, IEC/EN 62061, (Functional Safety of Electrical/Electronic/Programmable Electronic Safety-related Systems)
- Requirements for PLe (Performance Level e) and for safety category 4 in EN ISO13849-1

The NX-series Safety Control Units are also registered for RCM, EAC, and KC compliance.

## Conformance to EU Directives

### Applicable Directives

- EMC Directives
- Low Voltage Directive

## Concepts

### ● EMC Directives

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.*1

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.

*1. 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).

### ● Machinery Directive

The Machinery Directive requires ensuring the required safety for safety components used for machinery safety.

Applicable standards: EN ISO 13849-1.

### ● Conformance to EU Directives

The NX-series Units comply with EU Directives. To ensure that the machine or device in which the NX-series Units are used complies with EU Directives, the following precautions must be observed.

- The NX-series Units must be installed within a control panel.
- You must use SELV power supply for the DC power supplies that are connected as the Unit power supplies and I/O power supplies for the NX-series Units.
  EMC standard compliance was confirmed for the recommended Power Supplies. Refer to the user's manual for the connected CPU Unit for information on the recommended Power Supplies for the CPU Rack of the NX-series CPU Unit. If your High-speed Analog Input Unit is connected to a Slave Terminal, we recommend that you use the OMRON S8VK-S-series Power Supplies.
- NX-series Units that comply with EU Directives also conform to the Common Emission Standard. 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 in which the NX-series Units are used complies with EU Directives.
- This is a Class A product (for industrial environments). In a residential environment, it may cause radio interference. If radio interference occurs, the user may be required to take appropriate measures.

## Conformance to EN ISO 13849-1

EN ISO 13849-1 requires functional safety management to avoid systematic failure during the software development. This is required in all phases of the life cycle of software programming and software design (e.g., basic software design, safety circuit system design, and software upgrades) in safety control systems to be developed using safety controllers.

Therefore, functional safety management is required for design and development of software for facilities and equipment that use the function blocks provided in the Safety Controller.
The customer must implement measures to ensure compliance with these standards.

You can download the reliability data for safety of machinery that is required to verify the safety performance of your equipment from the following URL: http://www.ia.omron.com/support/sistemalibrary/index.html.

## Conformance to UL and CSA Standards

The NX-series Safety Control Units comply with the following UL and CSA standards. The application conditions for standard compliance are defined. Refer to the *Instruction Sheet* that is provided with each Unit before application.

## Conformance to Shipbuilding Standards

Some NX-series products comply with shipbuilding standards. If you use an NX-series product that complies with shipbuilding standards and the machinery or system in which you use the NX-series product must also comply with the standards, consult with your OMRON representative. Application conditions are defined according to the installation location. Application may not be possible for some installation locations.

### Usage Conditions for NK and LR Shipbuilding Standards

- A Safety Control Unit must be installed within a control panel.
- Gaps in the door to the control panel must be completely filled or covered with gaskets or other material.
- The following noise filter must be connected to the power supply line.

#### Noise Filter

| Name | Manufacturer | Model |
|---|---|---|
| Noise filter | Cosel Co., Ltd. | TAH-06-683 |

## Conformance to KC Certification

When you use this product in South Korea, observe the following precautions.

| 사 용 자 안 내 문 |
|---|
| 이 기기는 업무용 환경에서 사용할 목적으로 적합성평가를 받은<br>기기로서 가정용 환경에서 사용하는 경우 전파간섭의 우려가 있습니다. |

This product meets the electromagnetic compatibility requirements for business use. There is a risk of radio interference when this product is used in home.

## Usage Conditions for KC Certification

Take the same measures as those described in *Conformance to EU Directives* on page 34. In addition, attach a clamp core to the port side of the EtherNet/IP cable.
The recommended clamp core is given below.

**Recommended Clamp Core**

| Manufacturer | Product | Model | Turns of cable |
|---|---|---|---|
| TOKIN | Clamp core | ESD-SR-250 | 1 turn |

# Unit Versions

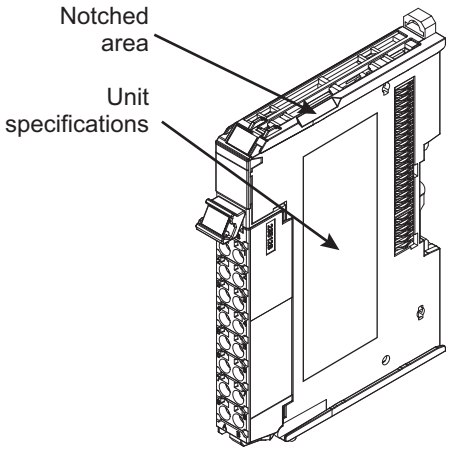
This section describes the notation that is used for unit versions, the confirmation method for unit versions, and the relationship between unit versions and Sysmac Studio versions.

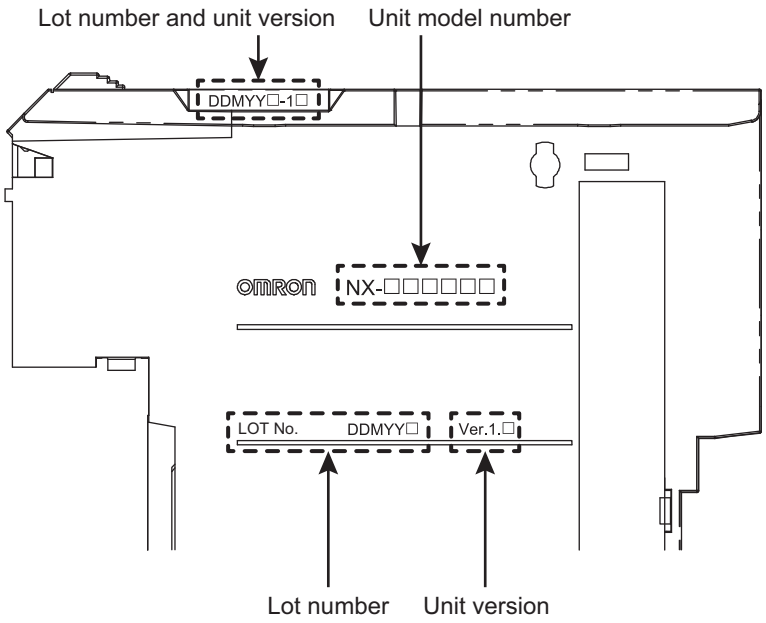
## Unit Versions

A “unit version” refers to a revision number of the respective NX-series Units. A “unit version” has been introduced to manage the Units in the NX Series according to differences in functionality accompanying Unit upgrades.

### Notation of Unit Versions on Products

The unit version is given with the Unit specifications on the side of the Unit or in the notched area.



The following information is provided in the Unit specifications on the Unit.

| Name | Function |
|---|---|
| Unit model number | Gives the model of the Unit. |
| Unit version | Gives the unit version of the Unit. |
| Lot number | Gives the lot number of the Unit.<br>DDMYY□: Lot number, □: Used by OMRON.<br>“M” gives the month (1 to 9: January to September, X: October, Y: November, Z: December) |

The following information is provided in the notched area on the Unit.

| Name | Function |
|---|---|
| Lot number and unit version | Gives the lot number and unit version of the Unit.<br>• DDMYY□: Lot number, □: Used by OMRON.<br>“M” gives the month (1 to 9: January to September, X: October, Y: November, Z: December)<br>• 1□: Unit version<br>The decimal portion of the unit version is omitted. (It is provided in the Unit specifications.) |

## Confirming Unit Versions with the Support Software

If your NX Unit is connected to a CPU Unit, refer to the user’s manual of the connected CPU Unit for the confirmation method for the unit version of the NX Unit.
If your NX Unit is connected to a Communications Coupler Unit, refer to the user’s manual of the connected Communications Coupler Unit for the confirmation method for the unit version of the Communications Coupler Unit and NX Unit.

# Unit Versions of Units and Sysmac Studio Versions

The functions that are supported depend on the unit version of the Unit. The version of Sysmac Studio that supports the functions that were added for an upgrade is also required to use those functions.
Refer to *A-20 Version Information* on page A-113 for the relationship between the unit versions of the Units and the Sysmac Studio versions, and for the functions that are supported by each unit version.

# Related Manuals

The followings are the manuals related. Use these manuals for reference.

| Manual name | Cat. No. | Model numbers | Application | Description |
|---|---|---|---|---|
| NX-series<br>Safety Control Unit<br>User's Manual | Z930 | NX-SL□□□□<br>NX-SI□□□□<br>NX-SO□□□□ | Learning how to use NX-series Safety Control Units. | Describes the hardware, setup methods, and functions of the NX-series Safety Control Units. |
| NX-series<br>Safety Control Unit<br>Instructions Reference Manual | Z931 | NX-SL□□□□ | Learning about the specifications of instructions for the Safety CPU Unit. | Describes the instructions for the Safety CPU Unit. |
| GI-S Series<br>Safety I/O Terminal<br>User's Manual | Z400 | GI-S□□□□□□ | Learning how to use the GI-S Series Safety I/O Terminals. | Describes the hardware, setup methods, and functions of the GI-S Series Safety I/O Terminals. |
| Environment-resistive Remote Terminal NXR-S-series Simple Logic Safety I/O Block for CIP Safety<br>User's Manual | W645 | NXR-SXD1204□-CST | Learning how to use the NXR-S Series Simple Logic Safety I/O Block. | Describes the hardware, setup methods, and functions of the NXR-S Series Simple Logic Safety I/O Block. |
| Environment-resistive Remote Terminal NXR-S-series Simple Logic Safety I/O Block for EtherNet/IP<br>User's Manual | W646 | NXR-SXD1204-EIT | | |
| NX-series<br>Data Reference Manual | W525 | NX-□□□□□□ | Referencing lists of the data that is required to configure systems with NX-series Units. | Lists of the power consumptions, weights, and other NX Unit data that is required to configure systems with NX-series Units are provided. |
| Sysmac Studio Version 1<br>Operation Manual | W504 | SYSMAC<br>-SE2□□□ | Learning about the operating procedures and functions of the Sysmac Studio. | Describes the operating procedures of the Sysmac Studio. |
| NJ/NX-series<br>Troubleshooting Manual | W503 | NX701-□□□□<br>NX502-□□□□<br>NX102-□□□□<br>NX1P2-□□□□<br>NJ501-□□□□<br>NJ301-□□□□<br>NJ101-□□□□ | Learning about the errors that may be detected in an NJ/NX-series Controller. | Concepts on managing errors that may be detected in an NJ/NX-series Controller and information on individual errors are described. |
| NX-series<br>EtherCAT® Coupler Unit<br>User's Manual | W519 | NX-ECC□□□ | Learning how to use the NX-series EtherCAT Coupler Unit and EtherCAT Slave Terminals. | The following items are described: the overall system and configuration methods of an EtherCAT Slave Terminal (which consists of an NX-series EtherCAT Coupler Unit and NX Units), and information on hardware, setup, and functions to set up, control, and monitor NX Units through EtherCAT. |
| NX-series<br>EtherNet/IP™<br>Coupler Unit<br>User's Manual | W536 | NX-EIC□□□ | Learning how to use an NX-series EtherNet/IP Coupler Unit and EtherNet/IP Slave Terminals. | The following items are described: the overall system and configuration methods of an EtherNet/IP Slave Terminal (which consists of an NX-series EtherNet/IP Coupler Unit and NX Units), and information on hardware, setup, and functions to set up, control, and monitor NX Units. |

| Manual name | Cat. No. | Model numbers | Application | Description |
|---|---|---|---|---|
| NX-series<br>System Units<br>User's Manual | W523 | NX-PD1□□□<br>NX-PF0□□□<br>NX-PC0□□□<br>NX-TB□□□X | Learning how to use NX-series System Units | The hardware and functions of the NX-series System Units are described. |
| NX-series<br>EtherNet/IP™ Unit User's Manual | W627 | NX-EIP201 | Learning how to use the NX-series EtherNet/IP Unit. | Information on the NX-series EtherNet/IP Unit is provided.<br>Information is provided on the basic setup, tag data links, and other features. |
| NX-series<br>NX502 CPU Unit<br>Hardware<br>User's Manual | W629 | NX502-□□□□ | Learning the basic specifications of the NX502 CPU Units, including introductory information, designing, installation, and maintenance.<br>Mainly hardware information is provided. | An introduction to the entire NX502 system is provided along with the following information on the CPU Unit.<br>• Features and system configuration<br>• Introduction<br>• Part names and functions<br>• General specifications<br>• Installation and wiring<br>• Maintenance and inspection |
| NX-series<br>NX102 CPU Unit<br>Hardware<br>User's Manual | W593 | NX102-□□□□ | Learning the basic specifications of the NX102 CPU Units, including introductory information, designing, installation, and maintenance.<br>Mainly hardware information is provided. | An introduction to the entire NX102 system is provided along with the following information on the CPU Unit.<br>• Features and system configuration<br>• Introduction<br>• Part names and functions<br>• General specifications<br>• Installation and wiring<br>• Maintenance and inspection |
| NJ/NX-series<br>CPU Unit<br>Built-in EtherNet/IP™ Port<br>User's Manual | W506 | NX701-□□□□<br>NX502-□□□□<br>NX102-□□□□<br>NX1P2-□□□□<br>NJ501-□□□□<br>NJ301-□□□□<br>NJ101-□□□□ | Using the built-in EtherNet/IP port on an NJ/NX-series CPU Unit. | Information on the built-in EtherNet/IP port is provided.<br>Information is provided on the basic setup, tag data links, and other features. |
| NJ/NX-series CPU Unit<br>Software User's Manual | W501 | NX701-□□□□<br>NX502-□□□□<br>NX102-□□□□<br>NX1P2-□□□□<br>NJ501-□□□□<br>NJ301-□□□□<br>NJ101-□□□□ | Learning how to program and set up an NJ/NX-series CPU Unit.<br>Mainly software information is provided. | The following information is provided on a Controller built with an NJ/NX-series CPU Unit.<br>• CPU Unit operation<br>• CPU Unit features<br>• Initial settings<br>• Programming based on IEC 61131-3 language specifications |
| NJ/NX-series Instructions<br>Reference Manual | W502 | NX701-□□□□<br>NX502-□□□□<br>NX102-□□□□<br>NX1P2-□□□□<br>NJ501-□□□□<br>NJ301-□□□□<br>NJ101-□□□□ | Learning detailed specifications on the basic instructions of an NJ/NX-series CPU Unit. | The instructions in the instruction set (IEC 61131-3 specifications) are described. |
| AC Servomotors/Servo Drives<br>G5 Series with<br>Built-in EtherCAT® Communications User's Manual | I576 | R88M-K□<br>R88D-KN□-ECT | Learning how to use the AC Servomotors/Servo Drives with built-in EtherCAT Communications. | Describes the hardware, setup methods and functions of the AC Servomotors/Servo Drives with built-in EtherCAT Communications.<br>The Linear Motor Type models and dedicated models for position control are available in G5-series. |
| | I577 | R88L-EC-□<br>R88D-KN□-ECT-L | | |

| Manual name | Cat. No. | Model numbers | Application | Description |
|---|---|---|---|---|
| Multi-function Compact Inverter<br>MX2-series V1<br>User's Manual | I585 | 3G3MX2-A□□□□-V1 | Learning how to use the MX2-series Multi-function Compact Inverter. | Describes the hardware, setup methods, and functions of the MX2-series Multi-function Compact Inverter. |
| AC Servomotors/Servo Drives<br>1S-series with Built-in EtherCAT® Communications<br>User's Manual | I586 | R88M-1L□/-1M□<br>R88D-1SN□-ECT | Learning how to use the AC Servomotors/ Servo Drives with built-in EtherCAT Communications. | Describes the hardware, setup methods and functions of the AC Servomotors/ Servo Drives with built-in EtherCAT Communications. |
| | I696 | R88M-1L□/-1M□<br>R88D-1SN□-ECT-51 | | |
| | I621 | R88M-1AL□/-1AM□<br>R88D-1SAN□-ECT | | |

# Terminology

| Term | Description |
|---|---|
| standard | The generic term for devices, functions, and data that are used for general control purposes as opposed to those that are used for safety measures. |
| safety function | A function that is executed by the safety control system to achieve a safe state for a machine hazard. |
| safe state | The status of a device or piece of equipment when the risk of danger to humans has been reduced to an acceptable level. |
| safety signal | A signal that is used for safety controls.<br>In this safety control system, the data type of a variable determines whether a signal is related to the safety controls. Broadly speaking, there are two data types: safety data types and standard data types. |
| standard signal | A signal or data that is used for general control purposes. |
| Safety data type | The data type for a safety signal. |
| Standard data type | The data type for a standard signal. |
| safety reaction time | The time required for the system to enter a safe state in a worst-case scenario after the occurrence of a safety-related input (press of an emergency stop pushbutton switch, interruption of a light curtain, opening of a safety door, etc.) or device failure.<br>The reaction time of the system includes the reaction times of sensors and actuators, just like the reaction time for a Controller or network. |
| safety control | A type of control that uses devices, functions, and data that are designed with special safety measures. |
| standard control | A type of control that use devices, functions, and data that are designed for general control purposes. This term is used to differentiate from a safety control |
| safety process data communications | A type of I/O data communications that is used for safety control purposes. |
| standard process data communications | A type of I/O data communications that is used for standard control purposes. |
| Safety I/O connection | A type of connections that is used for safety process data communications. |
| CIP Safety connection | Safety I/O connection that is used to transmit safety process data by the communication protocol called CIP Safety. CIP Safety originator connection and CIP Safety target connection are available depending on the roles of communications. |
| CIP Safety originator connection | A CIP Safety connection when an own node is a CIP Safety originator. |
| CIP Safety target connection | A CIP Safety connection when an own node is a CIP Safety target. |
| CIP Safety originator | A role in CIP Safety communications. A CIP Safety originator manages a CIP Safety connection for a CIP Safety target. CIP Safety originator exists in a connection unit, and not in a device unit. |
| CIP Safety target | A role in CIP Safety communications. A CIP Safety target receives a request to open a CIP Safety connection from a CIP Safety originator. CIP Safety target exists in a connection unit, and not in a device unit. |
| single-cast connection | A safety process data communications method in CIP Safety. A CIP Safety originator and a CIP Safety target communicate one-to-one in this connection.<br>You can apply this setting for both of input data and output data. |
| multi-cast connection | A safety process data communications method in CIP Safety. A CIP Safety target sends the input data of the CIP Safety target in multi-cast to multiple CIP Safety originators in this connection.<br>This is a connection type you can set only for input data of a CIP Safety target. You cannot apply this setting for output data. |
| FSoE master connection | Master safety I/O connection that is used to transmit safety process data by the communication protocol called FSoE. |
| safety master connection | The generic term for the CIP Safety originator connection and the FSoE master connection. |
| exposing global variables to the NJ/NX-series CPU Unit | Exposing specified global variables to the NJ/NX-series CPU Unit to allow the exchange of standard signals between the NJ/NX-series CPU Unit and the Safety CPU Unit. |
| Safety Control Unit | The generic term for a Unit that is used in safety controls. |
| Safety CPU Unit | A CPU Unit that is used for safety controls. This is a type of NX Unit. |
| Safety I/O Unit | An I/O Unit that is used for safety controls. This is a type of NX Unit. |

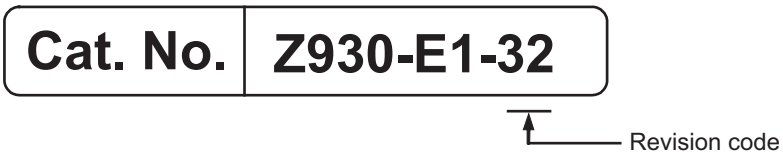
| Term | Description |
|---|---|
| safety input device | An input device that is designed with special safety measures for use in safety controls.<br>The generic term for safety input devices, such as emergency stop pushbutton switches and safety switches. |
| safety output device | An output device that is designed with special safety measures for use in safety controls.<br>The generic term for safety output devices, such as safety relays. |
| EtherNet/IP Slave Terminal | An EtherNet/IP Slave Unit Terminal is a building-block slave that is created by mounting a group of NX Units. |
| Safety program | User programming for safety controls in the Safety CPU Unit.<br>This term is used to differentiate safety programs from user programs in the NJ/NX-series CPU Unit.<br>Safety programs are programmed in the FBD language. |
| FBD language | The abbreviation for the function block diagram programming language. This is a graphical language used to program algorithms with connecting lines that represent the flow of inputs and data, and rectangular boxes that represent functions or function blocks.<br>Unlike the ladder diagram language, the FBD language does not have bus bars, and the connecting lines represent the flow of inputs and data rather than the power flow.<br>Algorithms are executed in order from top to bottom in units that are called networks. A network consists of configuration elements that use connecting lines to connect inputs to outputs. The FBD language does not have an END instruction. Execution for the task period ends when the last network is executed.<br>You use the FBD language to write safety programs for the Safety CPU Unit. |
| user program | All of the programs that are created by the user. It refers to the programs for standard controls of the NJ/NX-series CPU Units and the safety program of the Safety CPU Units. |
| operating mode | The status of the Safety CPU Unit, when it is in normal operation, that the user changes to run or check the operation of the Safety CPU Unit.<br>There are the three modes: PROGRAM mode, DEBUG mode, and RUN mode.<br>You can use DEBUG mode only when the Sysmac Studio is online with the Safety CPU Unit. |
| safety validation | The process of appending confirmation information to the safety application data if safety validation testing demonstrates that the safety controls meet the required specifications of a safety system.<br>You execute the safety validation from the Sysmac Studio when the Safety CPU Unit is in DEBUG mode. The validated safety programs are automatically transferred to the non-volatile memory of the Safety CPU Unit. |
| DEBUG mode | The mode that is used to debug unvalidated safety programs. DEBUG mode is only available when the Sysmac Studio is online with the Safety CPU Unit.<br>Use this mode to check that the safety programs and external devices operate correctly. After you confirm that the system meets the required specifications, perform the safety validation. This will enable you to change to RUN mode.<br>When you change from PROGRAM mode to DEBUG mode, the unvalidated safety programs are automatically transferred to the main memory of the Safety CPU Unit. |
| DEBUG mode (RUN) | A status that indicates that an unvalidated safety program is in execution in DEBUG mode. You can control BOOL variables, use forced refreshing, and change present values. |
| DEBUG mode (STOPPED) | A status that indicates that an unvalidated safety program is stopped in DEBUG mode. You can control BOOL variables, use forced refreshing, and change present values. |
| PROGRAM mode | A mode indicates that execution of the safety program is stopped. You cannot control BOOL variables, use forced refreshing, or change present values. |
| RUN mode | A mode that indicates that execution of the validated safety programs is in progress. Unlike DEBUG mode (RUN), the validated safety programs in the non-volatile memory of the Safety CPU Unit are executed. You cannot control BOOL variables, use forced refreshing, or change present values. |
| before safety validation | A status that indicates that safety validation has not been performed on the safety application data from the Sysmac Studio because it has not yet been determined whether the safety controls meet the required specifications of the safety system. |
| after safety validation | A status indicates that safety validation has been performed on the safety application data from the Sysmac Studio because it has been determined that the safety controls meet the required specifications of the safety system. |

| Term | Description |
|---|---|
| CPU Rack | A Rack to which a CPU Unit or Communication Control Unit is mounted. For NX-series CPU Units to which NX Units can be connected, a CPU Rack has a CPU Unit with NX Units and an End Cover mounted to it. For NX-series Communication Control Units, a CPU Rack has a Communication Control Unit with NX Units and an End Cover mounted to it. |
| CPU Rack settings | Consists of the following items.<br>• Configuration information<br>• Unit operation settings<br>• Unit application data |
| Slave Terminal settings | Consists of the following items.<br>• Configuration information<br>• Unit operation settings<br>• Unit application data |
| configuration information | It consists of the following data:<br>• Unit configuration information<br>• I/O allocation information |
| I/O allocation information | The set of information that specifies the I/O data to be processed by I/O refreshing.<br>On the Sysmac Studio, this is shown as configuration information and includes the Unit configuration information. |
| Unit configuration information | The set of information that specifies the configuration of the NX Units that are connected to the NX bus master.<br>On the Sysmac Studio, this is shown as configuration information and includes the I/O allocation information. |
| safety application data | The data that contains the settings that are used to operate the NX-series Safety Control Units.<br>It consists of the safety programs, safety task, and variables. You use the Sysmac Studio to create this data, and then transfer and execute it on the Safety CPU Unit.<br>On the Sysmac Studio, this data is shown as the slave parameters.<br>The location where the safety application data is stored on the Safety CPU Unit depends on whether the safety programs have been validated. (Unvalidated safety programs are stored in the main memory, while validated safety programs are stored in the non-volatile memory.) |
| safety input function | A function that evaluates whether the signals that are input on a safety input terminal are normal or abnormal.<br>Specific safety evaluation functions include test pulse evaluation and dual channel evaluation.<br>When the evaluation result shows an abnormality, the safety input data is made inactive (OFF). |
| safety output function | A function that evaluates whether the values of safety output data and the output signals on safety output terminals are normal or abnormal.<br>Specific safety evaluation functions include test pulse evaluation and dual channel evaluation.<br>When the evaluation result shows an abnormality, the output signal on the safety output terminal is turned OFF. |
| dual channel evaluation | This function uses a pair of safety input or safety output terminals as redundant terminals that are checked for consistency to evaluate the status of the safety input or safety output. |
| single channel | The input or output is used as a single point. |
| dual channels | Two inputs or outputs are used as a pair of points for redundancy. |
| test pulse evaluation | This function outputs a test pulse that is used to evaluate a safety input or safety output for failures or wiring errors with the connected external device. |
| change tracking | A pin is used to manage whether the safety application data has been changed after the finalized data is created. |
| UNID | An ID assigned to a device so that it can be uniquely identified by all the networks on the safety system for CIP Safety communications. An UNITD is a 10-byte value, consisting of a 6-byte Safety Network Number and a 4-byte Node ID. |
| Safety Network Number (SNN) | A number assigned to a safety network so that it can be uniquely identified for CIP Safety communications. The Safety Network Number is set for the NX bus, the built-in EtherNet/IP ports 1 and 2. |

| Term | Description |
|---|---|
| Node ID | An ID assigned to each of devices on a network so that the devices with the same Safety Network Number (SNN) can be uniquely identified for CIP Safety communications.<br>The Safety CPU Unit is the only CIP Safety device on the NX bus, and the Node ID is always *1*. For a CIP Safety device on an EtherNet/IP network, its IP address is used as the Node ID. |

# 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 | June 2013 | Original production |
| 02 | September 2013 | • Added NX-SL3500.<br>• Added information on offline debugging.<br>• Added information on 10-A Terminal Blocks.<br>• Corrected mistakes. |
| 03 | December 2013 | • Added procedures to upload data.<br>• Corrected mistakes. |
| 04 | August 2014 | • Made changes accompanying the upgrade to NX-SL□□□□unit version 1.1.<br>• Made changes accompanying the upgrade to NX-SIH400 unit version 1.1.<br>• Made revisions accompanying connection to EtherNet/IP Coupler Units.<br>• Corrected mistakes. |
| 05 | January 2015 | • Made revisions accompanying the upgrade to Sysmac Studio version 1.12. |
| 06 | April 2015 | • Made revisions accompanying the upgrade to Sysmac Studio version 1.13.<br>• Corrected mistakes. |
| 07 | July 2015 | • 4-1-2 Calculating the Safety Reaction Time<br>Corrected numeric values for OMRON Special Safety Input Devices that are used when calculating safety sensor/switch response time.<br>• 4-1-2 Calculating the Safety Reaction Time<br>Corrected the calculation method for safety input refresh time and safety output refresh time for safety I/O refresh time.<br>• 4-1-2 Calculating the Safety Reaction Time<br>Added precaution for conformance to ISO 13856-1:2013. |
| 08 | April 2016 | • Made revisions accompanying the upgrade to Sysmac Studio version 1.15.<br>• 3-3-1 Safety Input Function<br>Changed connection wiring diagrams.<br>• A-3 Application Examples<br>Changed wiring examples.<br>• Added precautions for correct use.<br>• Corrected mistakes. |
| 09 | July 2016 | • Made revisions accompanying the upgrade to Sysmac Studio version 1.16.<br>• Corrected mistakes. |

| Revision code | Date | Revised content |
|---|---|---|
| 10 | October 2016 | • Made revisions accompanying the upgrade to Sysmac Studio version 1.17.<br>• 6-9-2 Exporting/Importing Data for the Entire NX Unit<br>• 7-5-2 Registering POUs<br>Added program import/export functions.<br>• 8-5 Checking External Device Wiring<br>• 8-7-2 Data Protection<br>Added the program data protection function.<br>• Corrected mistakes. |
| 11 | April 2018 | • Made revisions accompanying connection to NX-series CPU Units.<br>• Made revisions associated with the addition of UM Safety Mats.<br>• Corrected mistakes. |
| 12 | July 2018 | • Added NX-SL5700 and NX-SL5500<br>• Corrected mistakes. |
| 13 | August 2018 | Added PFH values. |
| 14 | October 2018 | • Made revisions accompanying the appearance change of the indicators.<br>• Corrected mistakes. |
| 15 | April 2019 | • Made revisions accompanying the addition of R88D-1SAN□-ECT.<br>• Added precautions for correct use.<br>• Made revisions accompanying the upgrade to Sysmac Studio version 1.27.<br>• Corrected mistakes. |
| 16 | July 2019 | • Made revisions accompanying the upgrade to Sysmac Studio version 1.29.<br>• Corrected mistakes. |
| 17 | January 2020 | • Made revisions accompanying the upgrade to Sysmac Studio version 1.31.<br>• Corrected mistakes. |
| 18 | April 2020 | • Made changes accompanying the upgrade toNX-SL5500/NX-SL5700unit version 1.4.<br>• Made revisions accompanying the upgrade to Sysmac Studio version 1.40.<br>• Corrected mistakes. |
| 19 | July 2020 | • Made revisions accompanying the upgrade to Sysmac Studio version 1.41.<br>• Corrected mistakes. |
| 20 | April 2021 | • Made revisions accompanying the upgrade to Sysmac Studio version 1.45.<br>• Corrected mistakes. |
| 21 | April 2022 | • Made revisions accompanying the upgrade to Sysmac Studio version 1.49.<br>• Corrected mistakes. |
| 22 | June 2022 | • Corrected mistakes. |
| 23 | October 2022 | • Revisions for adding safety precautions regarding security. |
| 24 | April 2023 | • Made revisions accompanying the release of NX502 CPU Units and NX-series EtherNet/IP Units.<br>• Made revisions accompanying the upgrade to Sysmac Studio version 1.54. |
| 25 | October 2023 | • Made revisions accompanying the upgrade to Sysmac Studio version 1.56. |
| 26 | April 2024 | • Made revisions accompanying the upgrade to Sysmac Studio version 1.58. |
| 27 | June 2024 | • Made revisions accompanying the addition of D40A-2 Non-contact Door Switch.<br>• Added troubleshooting description for safety input.<br>• Corrected mistakes. |
| 28 | April 2025 | • Made revisions accompanying the upgrade to Sysmac Studio version 1.62. |
| 29 | July 2025 | • Made revisions accompanying the addition of precautions for correct use when setting TUNID to the target device. |
| 30 | April 2026 | • Corrected mistakes. |

| Revision code | Date | Revised content |
|---|---|---|
| 31 | July 2026 | • Made revisions accompanying the upgrade to Sysmac Studio version 1.67. |
| 32 | October 2026 | • Corrected mistakes. |

# 1



# Overview

This section introduces and describes the features, system configuration, and application procedure of the NX-series Safety Control Units.

**1-1 Introduction and Features ..... 1-2**
1-1-1 Overview of Safety Control Unit ..... 1-2
1-1-2 Features of Safety Control Unit ..... 1-5

**1-2 System Configuration and Configuration Devices ..... 1-8**
1-2-1 Safety Control System Configuration on NX bus of CPU Unit or EtherCAT Network ..... 1-8
1-2-2 Safety Control System Configuration on EtherNet/IP Network ..... 1-10
1-2-3 Standalone Safety Control System Configuration ..... 1-10
1-2-4 Types of Safety Control Unit ..... 1-11

**1-3 Support Software ..... 1-12**
1-3-1 Applicable Support Software ..... 1-12
1-3-2 Connection Method and Procedures for EtherCAT Coupler Units ..... 1-12
1-3-3 Connection Method and Procedures for EtherNet/IP Coupler Units ..... 1-14

**1-4 Commissioning Procedures ..... 1-15**
1-4-1 Overall Procedure ..... 1-15
1-4-2 Detailed Procedures ..... 1-16

# 1-1 Introduction and Features

## 1-1-1 Overview of Safety Control Unit

The NX-series Safety Control Units are part of the lineup of Sysmac devices. They are used to execute safety controls. Safety Control Units are classified as NX Units and they are used connected to an NX-seriess, EtherCAT Coupler Unit or EtherNet/IP Coupler Unit.
You use the integrated development environment that is provided by the Sysmac Studio Automation Software to build the safety control system, and perform all settings, programming, and debugging of the Safety Control Unit.

### Safety Control System on CPU Rack or EtherCAT Network

The NX-series Safety Control Units can be connected to the following Units.*1

- NX-series CPU Unit*2
- NX-series EtherCAT Coupler Unit

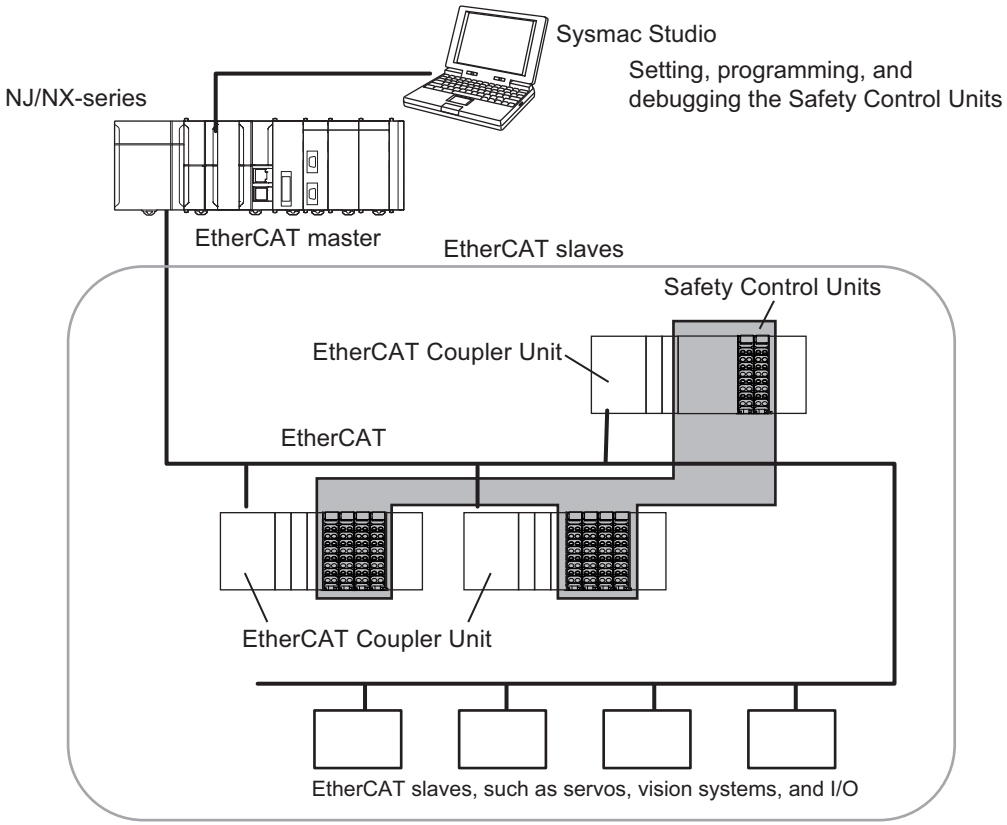
When using both the CPU Unit and EtherCAT Coupler Unit, you can integrate the installation, wiring, and setting methods of the NX Unit, which will enable design costs to be reduced.
These Units achieve safety controls on an CPU Rack or EtherCAT network when they are used in a sequence and motion control system that is based on the NJ/NX-series Machine Automation Controller.

*1. Only one Safety CPU Unit can be used in this system.
*2. Refer to the user's manual of the CPU Unit to be used for details on whether the Safety Control Unit can be connected to the CPU Unit to be used.

The Safety Control Unit connected to a CPU Unit performs I/O communications with the CPU Unit.
The Safety Control Unit connected to an EtherCAT Coupler Unit performs remote I/O communications with the EtherCAT master through the EtherCAT Coupler Unit. Both methods support the FSoE (Safety over EtherCAT) protocol through each communication and perform safety I/O communications between the FSoE master and slaves through the CPU Unit that has a built-in EtherCAT master. This allows you to use safety process data communications and standard process data communications on the same EtherCAT network through the NX bus of the CPU Unit that has a built-in EtherCAT master.

**Additional Information**

- NX502 CPU Units or NX102 CPU Units support connection of NX-SL□□□□ Units, NX-SI□□□□ Units, and NX-SO□□□□ Units.
- NX-series EtherCAT Coupler Units support connection of NX-SL3□□□ Units, NX-SI□□□□ Units, and NX-SO□□□□ Units. It does not support connection to NX-SL5□□□ Units.




When you set up the Safety System configuration on the Sysmac Studio, you automatically achieve safety process data communications (Safety over EtherCAT) on any EtherCAT network that was installed for standard process data communications.

## Safety Control System on EtherNet/IP Network

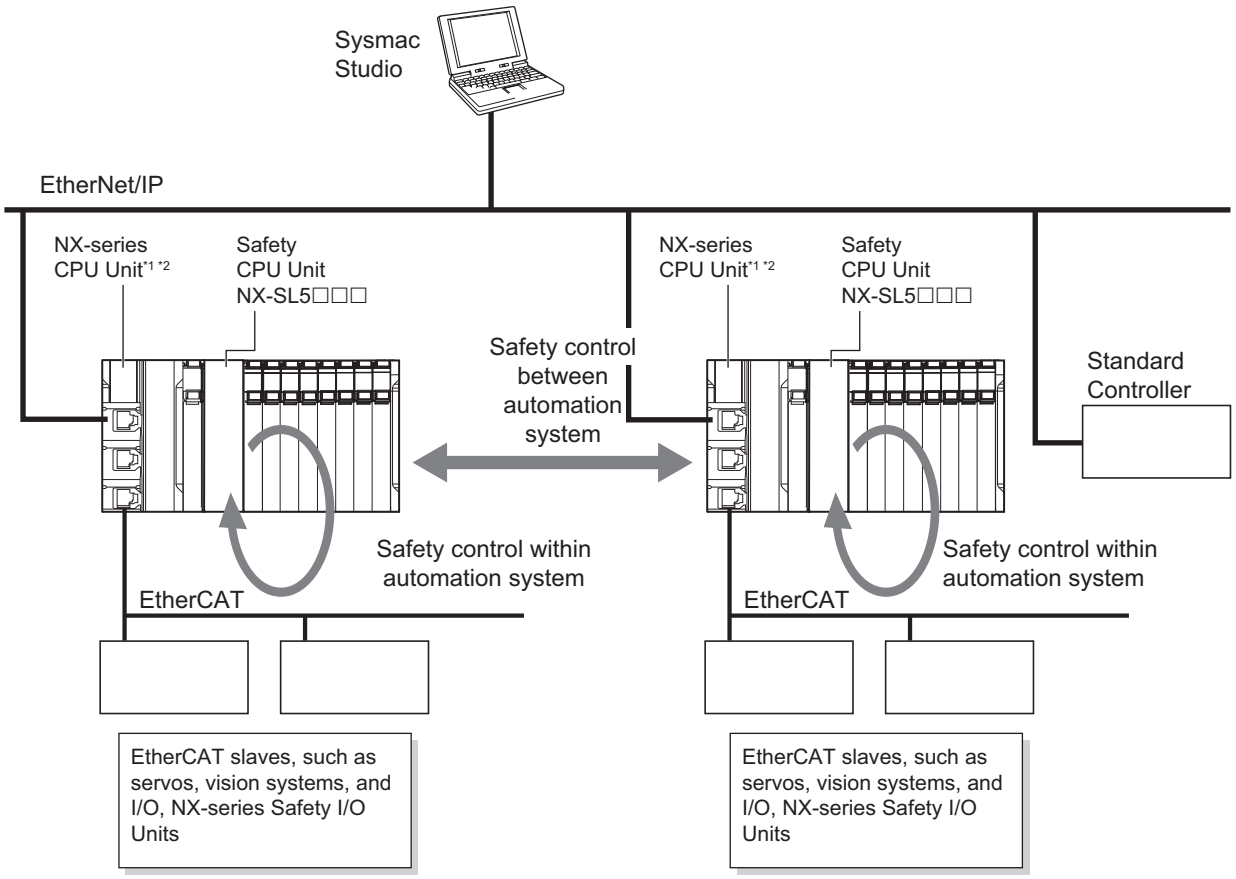
You can build a safety control system which performs CIP Safety communications with other controllers on the same EtherNet/IP network when you connect an NX-series NX-SL5□□□ Unit with an NX-series CPU Unit*1 *2.

*1. Refer to *A-20 Version Information* on page A-113 for the model numbers and unit versions of the applicable NX-series CPU Units.

*2. For the NX502 CPU Units, CIP Safety communications via the built-in EtherNet/IP port can be used with the NX502 CPU Unit with unit version 1.64 or later. If you set CIP Safety communications with the NX502 CPU Unit with unit version earlier than 1.64, use an NX-series EtherNet/IP Unit.

*3. When the NX502 CPU Unit with unit version 1.66 or later is used with the NX-series EtherNet/IP Unit with unit version 1.01 or later, you can use tag data link and CIP Safety communications together in one NX-series EtherNet/IP Unit.

The NX-series NX-SL5□□□ can also be connected to an NX-series Communication Control Unit NX-CSG320 to create a safety control system with CIP Safety communications in a system without an NX-series CPU Unit. Refer to the *NX-series Safety Control Unit / Communication Control Unit User's Manual (Cat. No. Z395)* for information on systems that use the NX-series Communication Control Unit.




*1. Refer to *A-20 Version Information* on page A-113 for the model numbers and unit versions of the applicable NX-series CPU Units.
*2. For the NX502 CPU Units, CIP Safety communications via the built-in EtherNet/IP port can be used with the NX502 CPU Unit with unit version 1.64 or later. If you set CIP Safety communications with the NX502 CPU Unit with unit version earlier than 1.64, use an NX-series EtherNet/IP Unit.
*3. When the NX502 CPU Unit with unit version 1.66 or later is used with the NX-series EtherNet/IP Unit with unit version 1.01 or later, you can use tag data link and CIP Safety communications together in one NX-series EtherNet/IP Unit.

This system enables the safety controls between respective automation systems upon building up each automation system that has the safety control built around EtherCAT which feature high speed and high precision.

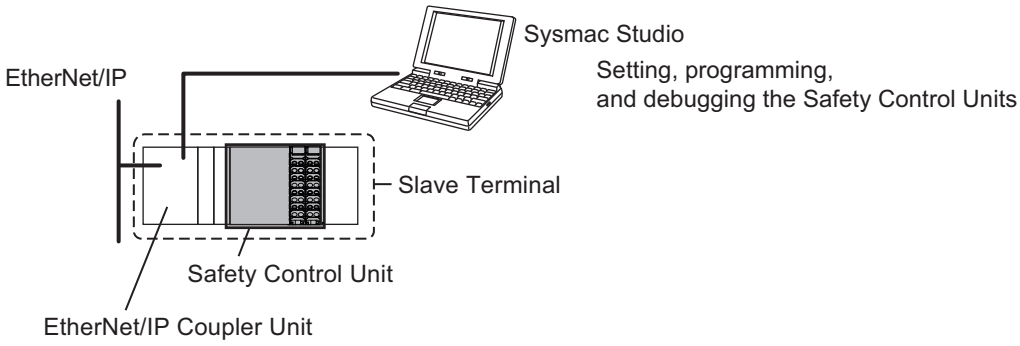
**Additional Information**

CIP Safety communications are only available for the combination where an NX-SL5□□□ is connected to an NX-series CPU Unit or to a Communication Control Unit. It cannot be used with NX-SL3□□□ Units.

## Standalone Safety Control System

The NX-SL3300 Unit is used connected to an EtherNet/IP Coupler Unit to achieve safety controls in a Slave Terminal.

The Safety Control Unit performs remote I/O communications with a standard controller through an EtherNet/IP Coupler Unit.




**Additional Information**

- Standalone safety control systems do not support CIP Safety communications.
- The EtherNet/IP Coupler Units support connection of NX-SL3300 Units, NX-SI□□□□ Units and NX-SO□□□□ Units. It does not support connection to NX-SL5□□□ Units and NX-SL3500 Units.

## 1-1-2 Features of Safety Control Unit

### Achieving Safety Control System on CPU Rack or EtherCAT Network

#### ● Integrating Safety Controls in a Sequence and Motion Control System

You can use NX-series Safety Control Units to integrate a safety control system into a sequence and motion control system.

The safety I/O communications support the FSoE protocol and can be built in as a subsystem on the NX Bus of CPU Unit or EtherCAT network. Therefore, no special safety control communications cables or interface devices are required for safety communications.

#### ● Easy Creation of an Interface with the Standard Controls

You can exchange data between the safety controls that are based on the Safety CPU Unit and the standard controls that are based on the NJ/NX-series CPU Unit.

This allows you to maintain the independent nature of the previously separate safety controls and standard controls while easily interfacing monitoring and commands between them.

### Achieving Safety Control System on EtherNet/IP Network

#### ● CIP Safety on EtherNet/IP Is Supported

You can combine an NX-series NX-SL5□□□ Unit with an NX-series CPU Unit*1 to enable CIP Safety-based communications between devices that support CIP Safety on EtherNet/IP or other Safety CPU Units.

*1. For the NX502 CPU Units, CIP Safety communications via the built-in EtherNet/IP port can be used with the NX502 CPU Unit with unit version 1.64 or later. If you set CIP Safety communications with the NX502 CPU Unit with unit version earlier than 1.64, use an NX-series EtherNet/IP Unit.

*2. When the NX502 CPU Unit with unit version 1.66 or later is used with the NX-series EtherNet/IP Unit with unit version 1.01 or later, you can use tag data link and CIP Safety communications together in one NX-series EtherNet/IP Unit.

## Achieving Standalone Safety Control System on EtherNet/IP Slave Terminal

### ● Constructing a Small Safety Control System

You can configure a small standalone safety control system on the EtherNet/IP Slave Terminal by using an NX-series Safety Control Unit.

### ● Easy Creation of an Interface with the Standard I/O Data

You can easily exchange data between Safety CPU Unit and standard I/O Units.

## Excellent Connectability with OMRON Safety I/O Devices

You can directly connect OMRON's wide lineup of Safety I/O Devices to Safety I/O Units without using any special units.

## Integrating Setting and Debugging Operations for Safety Controls into the Sysmac Studio

Setting and debugging operations for safety controls are integrated into the Sysmac Studio Automation Software.
The shared concepts, such as IEC 61131-3, consistent operating procedures, one-project management, integrated debugging, and integrated troubleshooting, reduce the software workload.

## Support for the IEC 61131-3 Programming Environment

### ● Program Languages Based on the IEC 61131-3 International Standard

Programming is possible with the FBD language, which is part of the programming language specifications of IEC 61131-3. And the safety function blocks that are defined in PLCopen® TC5 Safety are also supported.

### ● Programming with Variables

Programming with variables eliminates the need to specify memory addresses so that you can create user programs that are not dependent on any hardware considerations, such as the model of the Controller or the system configuration. This allows you to reuse user programming, even for different Controller models or system configurations.

## Complete Advanced Validation

## ● Checking Safety Programs and Safety Parameters

You can verify beforehand whether your safety programs (user program for safety controls that runs on the Safety CPU Unit) and safety parameters (parameters that are used for safety controls) meet the validity and safety aspects that are outlined below.

- Validity and safety issues related to function block diagram programs, such as missing or incorrect connection for function blocks
- Safety issues, such as the incorrect connection of a standard input to a safety input parameter of a function block
- Validity of the safety task period

These checks help to prevent design regression and help to ensure the reliability of the safety designs.

## ● Debugging function

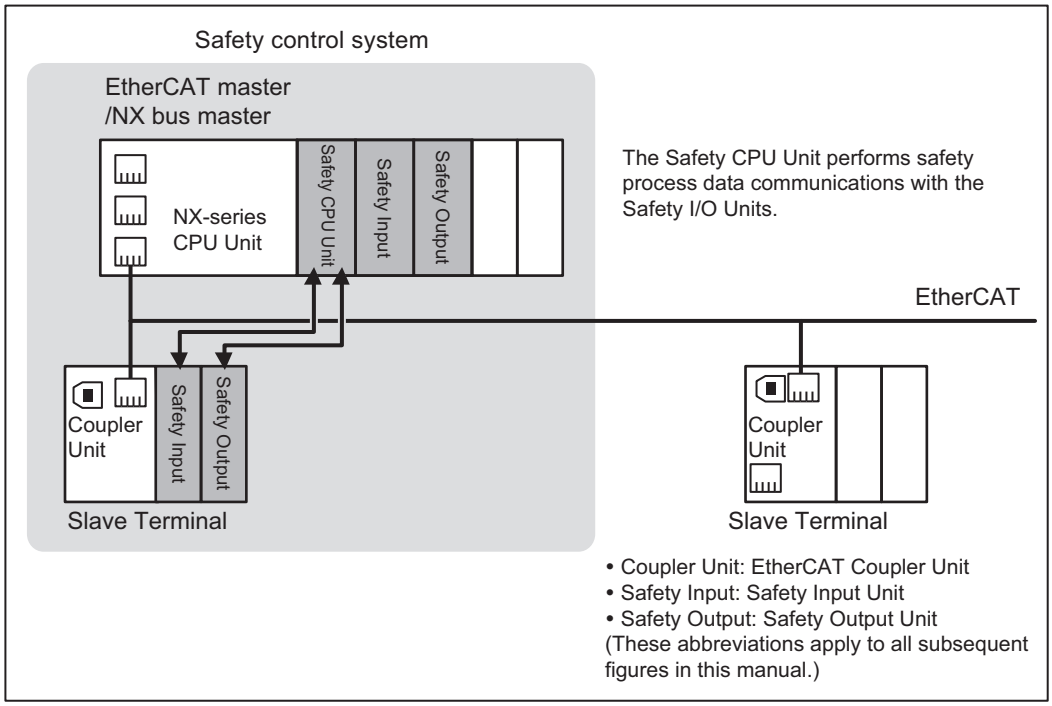
You can connect the Sysmac Studio to perform various types of debugging, including monitoring, changing present values, and forced refreshing.

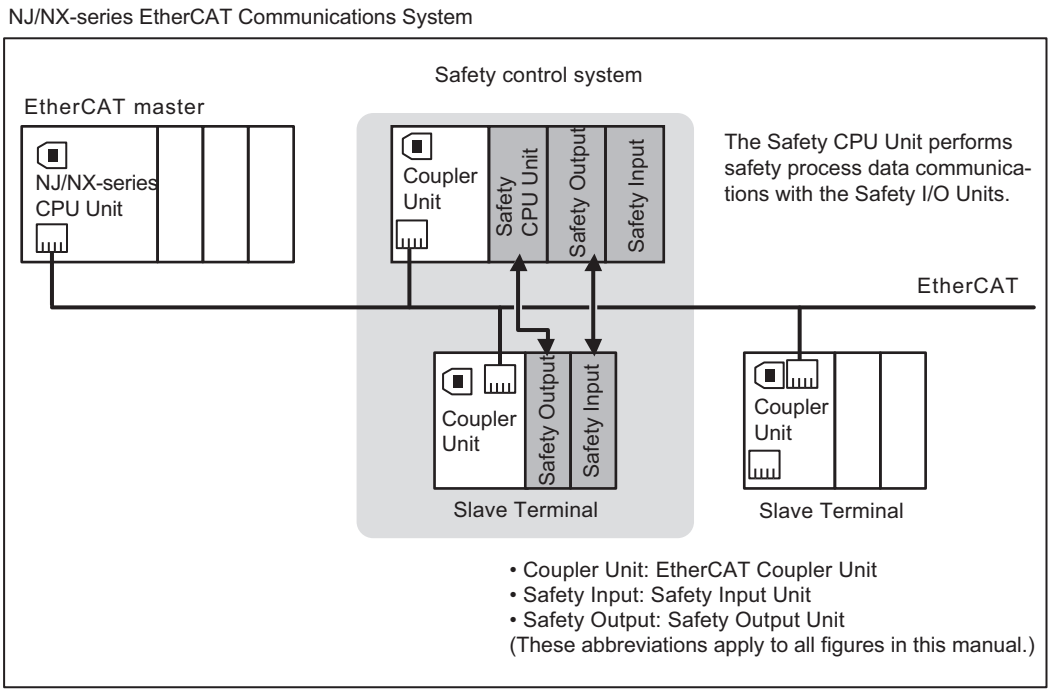
# 1-2 System Configuration and Configuration Devices

## 1-2-1 Safety Control System Configuration on NX bus of CPU Unit or EtherCAT Network

In this configuration, the safety control system operates on the NX bus of the NX-series CPU Unit and on the EtherCAT network.
The safety control system consists of the NX-series CPU Unit with the NX bus master function, CPU Unit with the EtherCAT master function, and the NX-series EtherCAT Coupler Unit with the EtherCAT slave function.

NX-series NX Bus/EtherCAT Integrated System



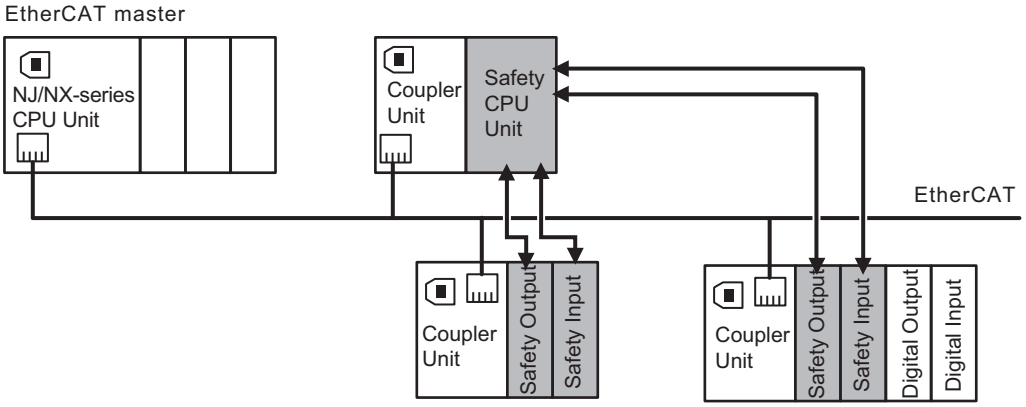



The Safety CPU Unit serves the FSoE master function to control the Safety I/O Units with the FSoE slave function served and the FSoE slave.*1 As shown in the above figure, the Safety CPU Unit can be used to control not only the Safety I/O Units that are mounted to the same CPU Rack or EtherCAT Slave Terminal, but also it can control Safety I/O Units that are mounted to other EtherCAT Slave Terminals through the EtherCAT network. You can use only one Safety CPU Unit on each EtherCAT network.

*1. The connectivity of FSoE communications between the OMRON NX-series Safety Control Unit is confirmed using the OMRON 1S-series Servo Drive.

You can also mount just a Safety CPU Unit to an EtherCAT Coupler Unit without mounting Safety I/O Units, as shown in the following figure. Standard I/O Units*1 can also be mounted in the same Slave Terminal, but they cannot be controlled by the Safety CPU Unit.

*1. This refers to NX Units used for standard control, such as Digital Input Units and Digital Output Units.




Refer to the *NX-series EtherCAT Coupler Unit User's Manual (Cat. No. W519)* for information on EtherCAT network configurations.

### ● Applicable NJ/NX-series CPU Units and NX-series EtherCAT Coupler Unit

Refer to *A-20 Version Information* on page A-113 for the model numbers and unit versions of the NJ/NX-series CPU Units and the NX-series EtherCAT Coupler Units that can be used together.

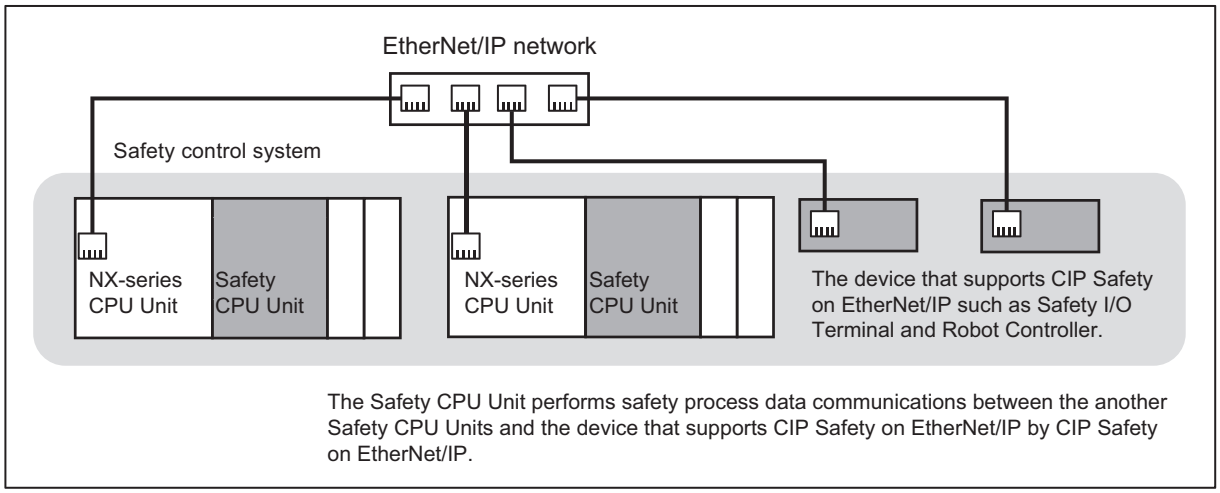
**Precautions for Correct Use**

If you use an NX-series CPU Unit, you cannot set both the priority-5 periodic task and another task for an EtherCAT Coupler Unit that contains a Safety Control Unit.
Refer to the *NX-series EtherCAT Coupler Unit User's Manual (Cat. No. W519)* for details on task settings.

## 1-2-2 Safety Control System Configuration on EtherNet/IP Network

The safety control system operates in the EtherNet/IP network.

Safety Control System in EtherNet/IP Network



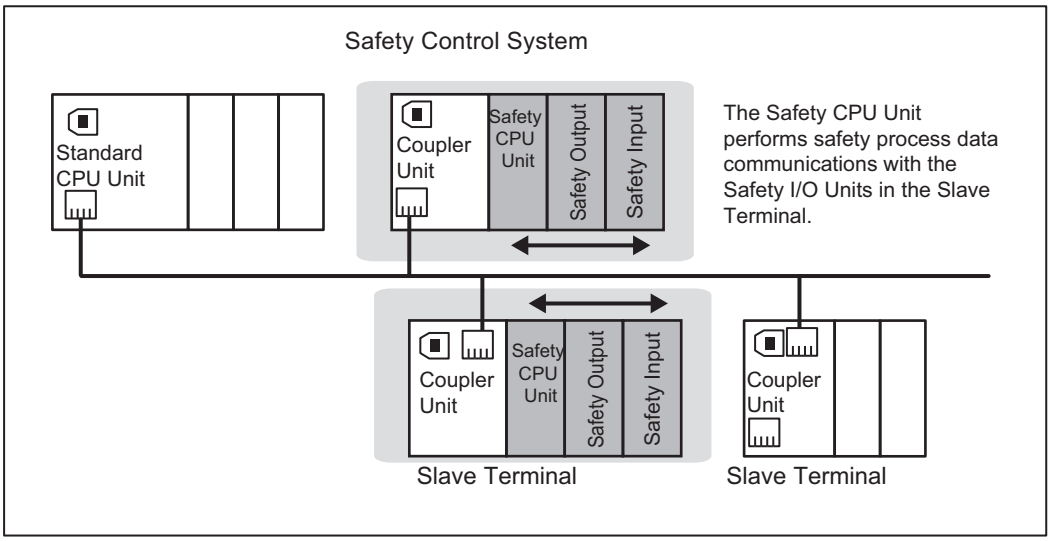

### ● Applicable NX-series CPU Units

Refer to *A-20 Version Information* on page A-113 for the model numbers and unit versions of the applicable NX-series CPU Units.
For the NX502 CPU Units, CIP Safety communications via the built-in EtherNet/IP port can be used with the NX502 CPU Unit with unit version 1.64 or later. If you set CIP Safety communications with the NX502 CPU Unit with unit version earlier than 1.64, use an NX-series EtherNet/IP Unit.
When the NX502 CPU Unit with unit version 1.66 or later is used with the NX-series EtherNet/IP Unit with unit version 1.01 or later, you can use tag data link and CIP Safety communications together in one NX-series EtherNet/IP Unit.

## 1-2-3 Standalone Safety Control System Configuration

The safety control system operates in the Slave Terminals of the EtherNet/IP Coupler Units.

Safety Control System in EtherNet/IP Slave Terminals




You can control Safety I/O Units that are mounted to the same Slave Terminal as the Safety CPU Unit You cannot use the EtherNet/IP network to control the operation of Safety I/O Units that are mounted to other Slave Terminals. You can use only one Safety CPU Unit on each Slave Terminal.

### ● Applicable NX-series EtherNet/IP Coupler Units

Refer to *A-20 Version Information* on page A-113 for the model numbers and unit versions of the applicable NX-series EtherNet/IP Coupler Units.

## 1-2-4 Types of Safety Control Unit

This section provides the Classification of NX-series Safety Control Unit.

| Unit classification | | Introduction |
|---|---|---|
| Safety CPU Unit | | This Unit has safety control functions.<br>It operates as an NX Unit.<br>It also operates as an FSoE master.<br>It operates as a CIP-Safety-on-EtherNet/IP device. |
| Safety I/O Units | | These Units have safety input functions or safety output functions.<br>They operate as NX Units.<br>These Units operate as FSoE slaves. |
| | Safety Input Unit | These Units have safety input functions. |
| | Safety Output Unit | These Units have Safety Output functions. |

Refer to *Section 2 Specifications* on page 2-1 for details on individual Units.

# 1-3 Support Software

You use the Support Software to set up the safety control system for the Safety Control Unit and to perform programming and debugging.

## 1-3-1 Applicable Support Software

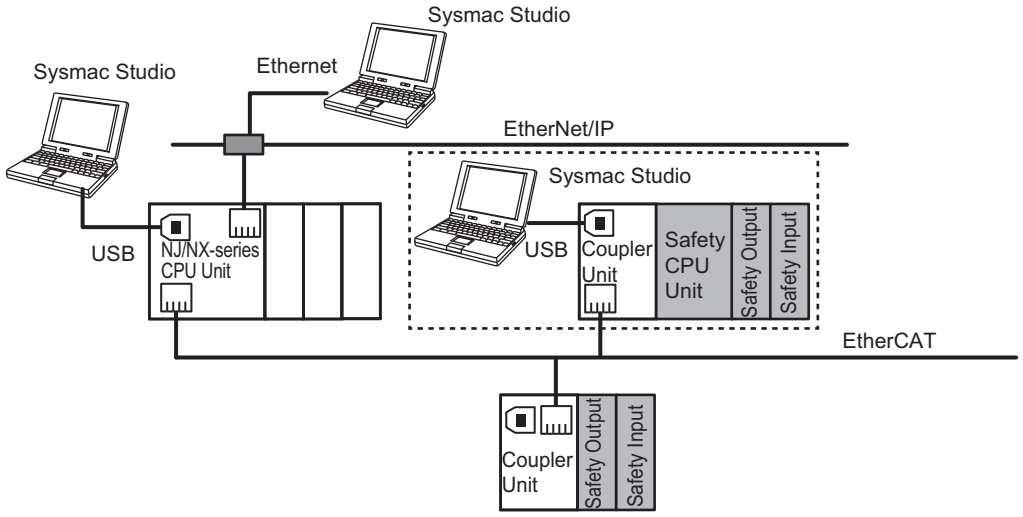
You use the Support Software to set up the safety control system for the Safety Control Unit, and to perform programming and debugging. Refer to *A-20 Version Information* on page A-113 for information on combinations of the applicable Support Software and versions.

Refer to the *Sysmac Studio Version 1 Operation Manual (Cat. No. W504)* for the system requirements of the Sysmac Studio.

## 1-3-2 Connection Method and Procedures for EtherCAT Coupler Units

In general, there are the following two ways to connect the Sysmac Studio.

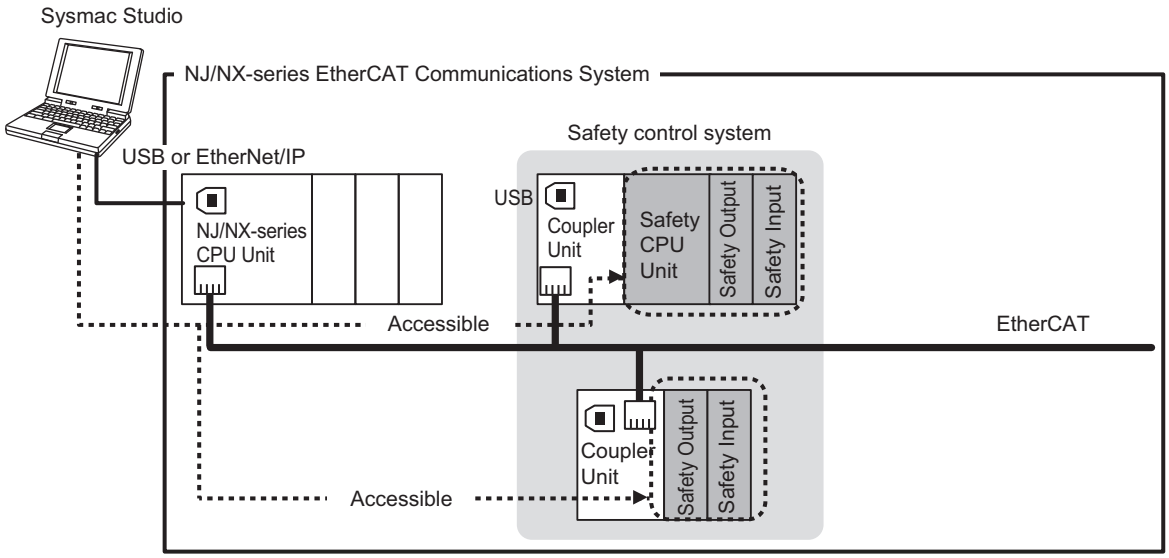
- USB Connection or Ethernet Connection to the NJ/NX-series CPU Unit
  The USB connection and Ethernet connection are functionally identical. This is the most common connection method. For the NX502 CPU Units, you can also use the Ethernet connection with the NX-series EtherNet/IP Unit.
- USB Connection to the NX-series EtherCAT Coupler Unit
  This method is functionally limited when compared with a connection to the NJ/NX-series CPU Unit. This connection method is used mainly for debugging.




Refer to the *Sysmac Studio Version 1 Operation Manual (Cat. No. W504)* for the connection procedure.

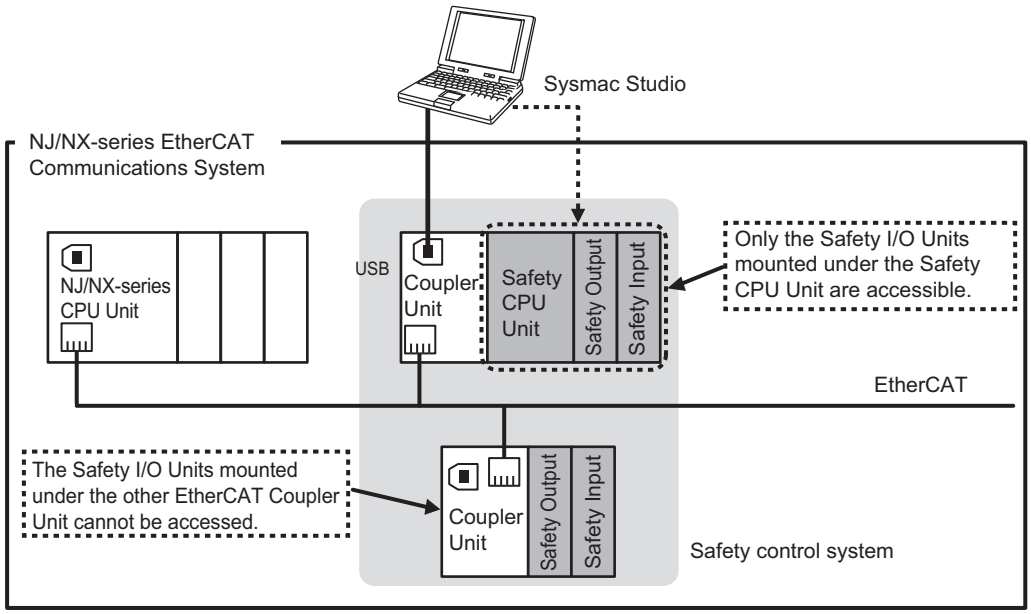
## USB Connection or Ethernet Connection to the NJ/NX-series CPU Unit

You use the Sysmac Studio to connect to the USB port or the built-in EtherNet/IP port of the NJ/NX-series CPU Unit. For the NX502 CPU Units, you can also connect to the EtherNet/IP port of the NX-series EtherNet/IP Unit.
This connection allows you to download safety programs to the Safety CPU Unit and to monitor the Safety I/O Units on the EtherCAT network.




## USB connection to the EtherCAT Coupler Unit

You can connect the Sysmac Studio to the USB port on the EtherCAT Coupler Unit.
This connection allows you to download, upload, and monitor the safety programs for only the Safety CPU Unit and Safety I/O Units that are under the EtherCAT Coupler Unit that the Sysmac Studio is online with. The other devices cannot be accessed.



**Precautions for Correct Use**

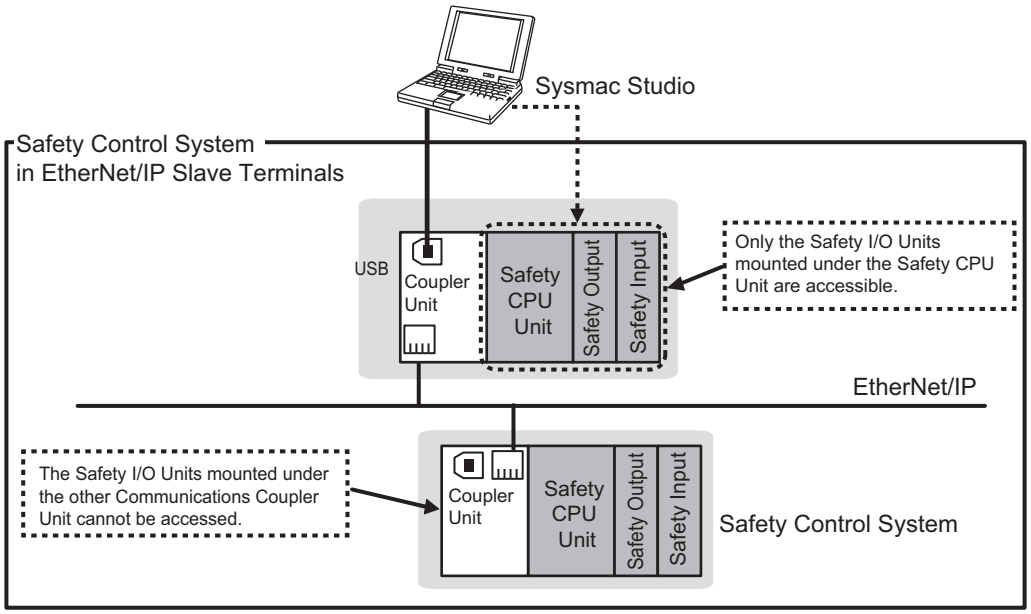
There are functional restrictions when you connect to the EtherCAT Coupler Unit via the USB port in comparison with connecting to the NJ/NX-series CPU Unit. We therefore recommend that you connect to the NJ/NX-series CPU Unit.

## Functional Differences on the Sysmac Studio Based on the Connection Point

The functions that you can use on the Sysmac Studio depend on what the Sysmac Studio is connected to. Refer to the *NX-series EtherCAT Coupler Unit User's Manual (Cat. No. W519)* for details.

## 1-3-3 Connection Method and Procedures for EtherNet/IP Coupler Units

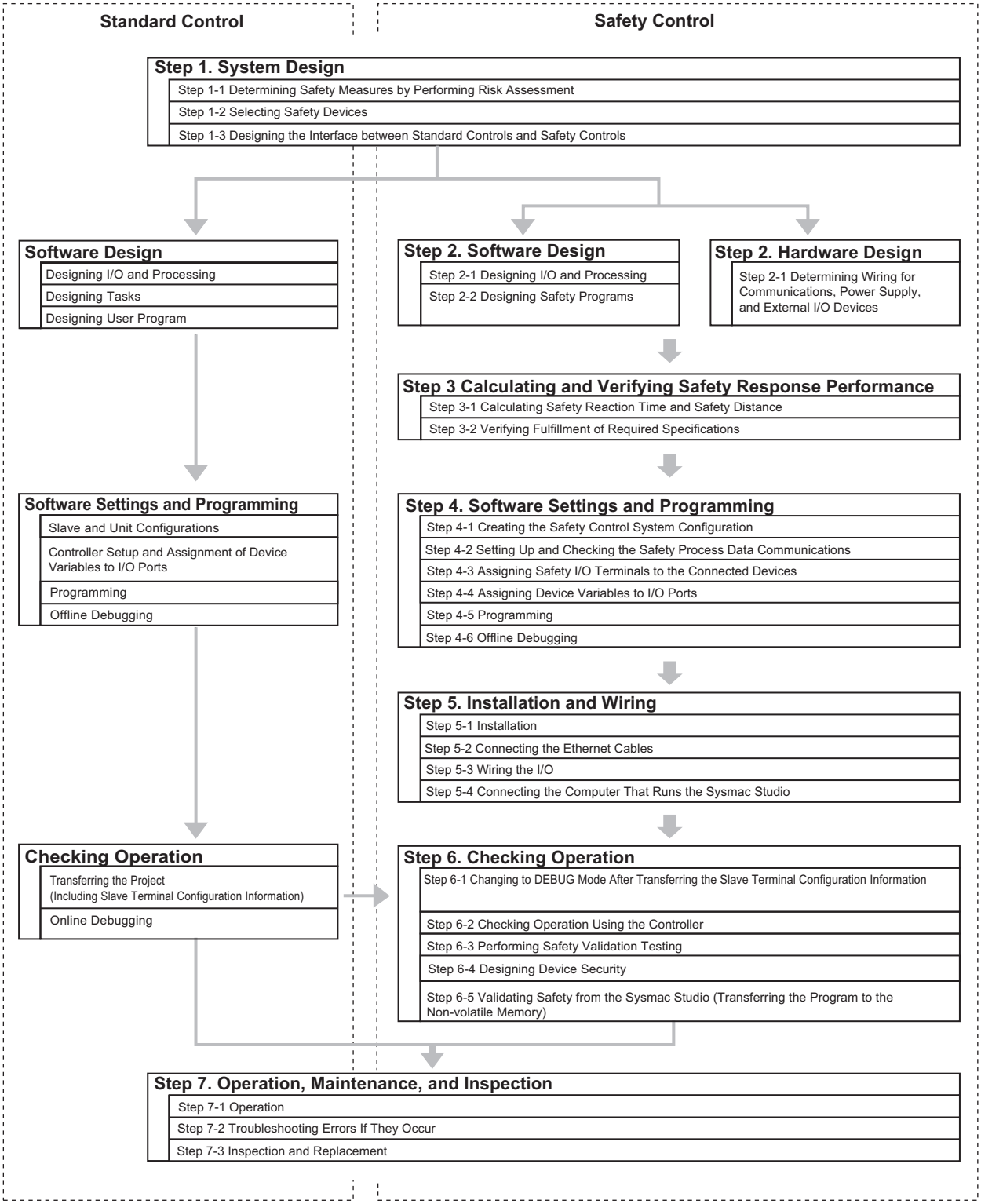
Connect the Sysmac Studio to the USB port on the EtherNet/IP Coupler Unit.
This connection allows you to download, upload, and monitor the safety programs for only the Safety CPU Unit and Safety I/O Units that are under the EtherNet/IP Coupler Unit that the Sysmac Studio is online with. The other devices cannot be accessed.



# 1-4 Commissioning Procedures

## 1-4-1 Overall Procedure

Use the following procedure to build a safety system.
The procedure is divided into steps for standard control and safety control.



## 1-4-2 Detailed Procedures

As described in the previous section, the standard controls and safety controls are linked with one another throughout the setup procedures.
This section describes the detailed procedures for the safety controls.
Refer to *NJ/NX-series CPU Unit Software User's Manual (Cat. No. W501)*for the detailed procedures for using NJ/NX-series CPU Units for standard control.

Step 1. System Design

| Step | Description | Reference |
|---|---|---|
| Step 1-1 Determining Safety Measures by Performing Risk Assessment | • Identify potential danger factors and perform risk assessment.<br>• Study and decide on measures to reduce risks. | --- |
| ↓ | | |
| Step 1-2 Selecting Safety Devices | Select the safety devices for inputs, logic, and outputs of the safety controls. | *1-2 System Configuration and Configuration Devices* on page 1-8<br>*Section 2 Specifications* on page 2-1<br>*2-1-2 Part Names and Functions* on page 2-8 |
| ↓ | | |
| Step 1-3 Designing the Interface between Standard Controls and Safety Controls | Design the interface between the standard controls and safety controls.<br>This is done by exposing the global variables to the NJ/NX-series CPU Unit. | *5-7 Exposing Variables to Standard Controllers* on page 5-49 |

Step 2. Software Design

| Step | Description | Reference |
|---|---|---|
| Step 2-1 Designing I/O and Processing | Design the configuration of the safety I/O devices and Safety I/O Units.<br>• Safety I/O devices<br>• Program contents | *2-1-2 Part Names and Functions* on page 2-8 |
| ↓ | | |

| Step | Description | Reference |
|---|---|---|
| Step 2-2 Designing Safety Programs | Design the POUs (Program Organization Units).<br>• Programs<br>• Function blocks<br><br>Design of Variables:<br>• Design the data types of the variables (particularly the design of safety data types and standard data types).<br>• Define the variables that you will use in more than one POU and variables that you will use in only specific POUs.<br>• Define the variable names for the device variables that you use to access Safety I/O Units.<br>• Define the attributes of variables, such as the Name attribute.<br>• Design the variables to expose to the user program for the standard controls.<br>• Design the interface with the safety controls of the user program for the standard controls. | *Section 6 Programming* on page 6-1 |

Step 2. Hardware Design

| Step | Description | Reference |
|---|---|---|
| Step 2-1 Determining Wiring for Communications, Power Supply, and External I/O Devices | Determine the wiring for the communications network, power supply, and safety I/O devices. | *2-1-2 Part Names and Functions* on page 2-8<br>Manuals for specific Communications Coupler Units<br>Manuals for the specific NX Units |

Step 3. Calculating and Verifying Safety Response Performance

| Step | Description | Reference |
|---|---|---|
| Step 3-1 Calculating Safety Reaction Time and Safety Distance | Calculate the safety reaction times and use them to find the safety distances. | *Section 8 Calculating Safety Reaction Times* on page 8-1 |
|  | | |
| Step 3-2 Verifying Fulfillment of Required Specifications | Check to see if requirements are met. If requirements are not met, reconsider the designs again starting with the system design. | *Section 8 Calculating Safety Reaction Times* on page 8-1 |

Step 4. Software Settings and Programming

| Step | Description | Reference |
|---|---|---|
| Step 4-1 Creating the Safety Control System Configuration | On the Sysmac Studio, configure the NX bus master, Safety CPU Unit and Safety I/O Units. | *5-3 Controller Configuration and Setup of the Safety Control Units* on page 5-5 |
|  | | |
| Step 4-2 Setting Up and Checking the Safety Process Data Communications | Check the settings for the safety process data communications and make any necessary changes. | *5-4 Setting Up the Safety Process Data Communications* on page 5-12 |

| Step | Description | Reference |
|---|---|---|
| Step 4-3 Assigning Safety I/O Terminals to the Connected Devices | On the Parameter Settings Pane for the Safety I/O Units, select the safety I/O devices that are connected to the safety I/O terminals. | *5-5 Safety I/O Functions* on page 5-40 |
| Step 4-4 Assigning Device Variables to I/O Ports | Register the device variable in the variable table.<br>(You can use either user-defined or automatically assigned variable names.) | *5-6 Registering Device Variables* on page 5-43 |
| Step 4-5 Programming | Variable Registration:<br>• Register the variables that are used by more than one POU in the global variable table with the Sysmac Studio.<br>• Register the variables that are used in only a specific program in the local variable table for that program.<br>• Register the variables that are used in only a specific function block in the local variable table for that function block.<br><br>Writing Algorithms for POUs:<br>Write the algorithms for the POUs (programs and function blocks) using the FBD language. | *6-5 Programming Operations* on page 6-26 |
| Step 4-6 Offline Debugging | The Simulator is used to debug the program. | *6-8 Offline Debugging* on page 6-98 |

Step 5 Installation and Wiring

| Step | Description | Reference |
|---|---|---|
| Step 5-1 Installation | • Connect the Units to each other.<br>• Mount the connected Units on a DIN Track. | *Section 3 Installation and Wiring* on page 3-1 |
| Step 5-2 Connecting the Ethernet Cables | Connect the NJ/NX-series CPU Units or other standard controllers to the Communications Coupler Units. | Manuals for specific Communications Coupler Units |
| Step 5-3 Wiring the I/O | • Wire the Safety I/O Units. | *Section 3 Installation and Wiring* on page 3-1 |

| Step | Description | Reference |
|---|---|---|
| Step 5-4 Connecting the Computer That Runs the Sysmac Studio | Use one of the following connections.<br>• Connect a USB cable to the NJ/NX-series CPU Unit*1<br>• Connect an Ethernet cable to the built-in EtherNet/IP port on the NJ/NX-series CPU Unit.*1<br>• Connect a USB cable to the Communications Coupler Unit | *1-3 Support Software* on page 1-12<br>Refer to the *Sysmac Studio Version 1 Operation Manual (Cat. No. W504)*<br>Manuals for specific Communications Coupler Units |

*1. You can use this connection method only if you use an NJ/NX-series CPU Unit and EtherCAT Coupler Unit. For the NX502 CPU Units, you can also connect to the EtherNet/IP port of the NX-series EtherNet/IP Unit.

Step 6. Checking Operation

| Step | Description | Reference |
|---|---|---|
| Step 6-1 Changing to DEBUG Mode After Transferring the Slave Terminal Configuration Information | Place the Sysmac Studio online with the NJ/NX-series CPU Unit*1 or the Communications Coupler Unit and then transfer the Slave Terminal configuration information.<br>Then, change the Safety CPU Unit from the Safety CPU Unit Setup and Programming View. This transfers the safety application data to the Safety CPU Unit and enables debugging. | *7-2 Transferring the Configuration Information* on page 7-7<br>*7-3 Operating Modes of the Safety CPU Unit* on page 7-11<br>*7-4 Changing to DEBUG Mode* on page 7-17 |
| ↓ | | |
| Step 6-2 Checking Operation Using the Controller | Check all wiring and the operation of the program to check that the Safety Control Unit operates as intended. | *7-6 Functions for Checking Operation* on page 7-28 |
| ↓ | | |
| Step 6-3 Performing Safety Validation Testing | Test all safety functions to see if they operate according to designs. | *7-6 Functions for Checking Operation* on page 7-28 |
| ↓ | | |
| Step 6-4 Designing Device Security | Set the safety passwords. | *7-10 Security Settings* on page 7-59 |
| ↓ | | |
| Step 6-5 Validating Safety from the Sysmac Studio | After the safety validation testing has been passed, execute the Safety Validation operation from the Sysmac Studio.<br>This transfers the safety application data to the non-volatile memory in the Safety CPU Unit and enables operation. | *7-11 Performing Safety Validation and Operation* on page 7-65 |

*1. You can use this connection method only if you use an NJ/NX-series CPU Unit and EtherCAT Coupler Unit.

Step 7. Operation, Maintenance, and Inspection

| Step | Description | Reference |
|---|---|---|
| Step 7-1 Operation | Restart the Safety CPU Unit.<br>If the Safety CPU Unit has a validated user program, the Safety CPU Unit will automatically start in RUN mode. | *7-11 Performing Safety Validation and Operation* on page 7-65 |
|  | | |
| Step 7-2 Troubleshooting Errors If They Occur | If an error occurs, use the troubleshooting function of the Sysmac Studio to check the error and determine the cause. Then, remove the error. | *Section 12 Troubleshooting* on page 12-1 |
|  | | |
| Step 7-3 Inspection and Replacement | Perform periodic maintenance.<br>If you find any defects or problems during the inspection, replace the affected devices. | *Section 13 Inspection and Maintenance* on page 13-1 |

# 2



# Specifications

This section provides the specifications of the Safety CPU Unit and the Safety I/O Units.

**2-1 Safety CPU Unit ..... 2-2**
2-1-1 Models and Specifications ..... 2-2
2-1-2 Part Names and Functions ..... 2-8
2-1-3 Indicators ..... 2-10
**2-2 Safety Input Unit ..... 2-17**
2-2-1 Models and Specifications ..... 2-17
2-2-2 Part Names and Functions ..... 2-23
2-2-3 Indicators ..... 2-25
**2-3 Safety Output Unit ..... 2-30**
2-3-1 Models and Specifications ..... 2-30
2-3-2 Part Names and Functions ..... 2-37
2-3-3 Indicators ..... 2-40
**2-4 PFH ..... 2-44**

# 2-1 Safety CPU Unit

This section describes the models and specifications of the Safety CPU Units as well as the names and functions of the parts.

## 2-1-1 Models and Specifications

The Safety CPU Unit specifications are described below.

### Models

The following table specifies the list of Safety CPU Unit models.

| Model | Maximum number of safety I/O points | Program capacity | Number of safety I/O connections | I/O refreshing method |
|---|---|---|---|---|
| NX-SL3300 | 256 points | 512 KB | 32*1 | Free-Run refreshing |
| NX-SL3500 | 1,024 points | 2,048 KB | 128*1 | Free-Run refreshing |
| NX-SL5500 | 1,024 points | 2,048 KB | 128*2 | Free-Run refreshing |
| NX-SL5700 | 2,032 points | 4,096 KB | 254*2 | Free-Run refreshing |

*1. Only FSoE master connections are supported.
*2. CIP Safety originator connections, CIP Safety target connections, and FSoE master connections are supported.

### General Specifications

This section provides the general specifications of the Safety CPU Unit.

| Item | Specification |
|---|---|
| **Enclosure** | Mounted in a panel (open) |
| **Grounding method** | Ground to 100 Ω or less |

| Item | | Specification |
|---|---|---|
| **Operating environment** | **Ambient operating temperature** | 0 to 55°C |
| | **Ambient operating humidity** | 10% to 95% (with no condensation or icing) |
| | **Atmosphere** | Must be free from corrosive gases. |
| | **Ambient storage temperature** | −25 to 70°C (with no condensation or icing) |
| | **Altitude** | 2,000 m max. |
| | **Pollution degree** | 2 or less: Meets IEC 61010-2-201 |
| | **Noise immunity** | Conforms to IEC 61131-2, 2 kV (power supply line) |
| | **Insulation class** | CLASS III (SELV) |
| | **Overvoltage category** | Category II: Meets IEC 61010-2-201 |
| | **EMC immunity level** | Zone B |
| | **Vibration resistance** | Conforms to IEC 60068-2-6<br>5 to 8.4 Hz with amplitude of 3.5 mm<br>8.4 to 150 Hz, acceleration of 9.8 $m/s^2$<br>100 min. in each X, Y, and Z directions (10 sweeps of 10 min. each = 100 min. total) |
| | **Shock resistance** | Conforms to IEC 60068-2-27<br>147 $m/s^2$, 3 times in each X, Y, and Z directions |
| **Installation method** | | DIN Track (IEC 60715 TH35-7.5/TH35-15) |

## Individual Specifications

This section provides the individual specifications of the Safety CPU Unit.

### ● Datasheet Items for Safety CPU Unit

The following table gives the meaning of the datasheet items for the Safety CPU Unit.

| Item | Description |
|---|---|
| **Maximum number of safety I/O points** | This is the number of safety I/O points that Safety CPU Unit can control. |
| **Program capacity** | This is the capacity of the safety programs in the Unit. |
| **Number of safety I/O connections*1** | This is the maximum number of Safety I/O connections that can be set to this Unit. The value is the total number of CIP Safety originator connections, CIP Safety target connections, and FSoE master connections. |
| **Number of CIP Safety originator connections*2** | This is the maximum number of CIP Safety originator connections that can be set to this Unit. |
| **Number of CIP Safety target connections*2** | This is the maximum number of CIP Safety target connections that can be set to this Unit. |
| **Number of originators that can be connected with a multi-cast connection** | When this unit is a CIP Safety target, it is the number of CIP Safety originators that can be connected with a multi-cast connection by a single CIP Safety target connection. |
| **CIP Safety maximum data size per connection** | This is the maximum data size per connection for CIP Safety. |
| **Number of FSoE master connections** | It is the maximum number of FSoE master connections that can be configured to this unit. |
| **I/O refreshing method** | The I/O refreshing methods that are used by the Unit. |

| Item | Description |
|---|---|
| **External connection terminals** | The type of terminal block and connector that is used for connecting the Unit. This specification includes the number of terminals for a screwless clamping terminal block. |
| **Indicators** | The type of indicators on the Unit and the layout of those indicators. |
| **Hardware switch settings** | It is the type and layout of hardware switches for the Unit. |
| **Dimensions (mm)** | These are the external dimensions of the Unit. The dimensions are given in the form W × H × D. The dimensions are given in "millimeters". |
| **I/O power supply method** | The method for supplying I/O power for the Unit. The supply method is determined for each Unit. The power can be supplied either from the NX bus or from an external source. |
| **Current capacity of I/O power supply terminal** | The current capacity of the I/O power supply terminals (IOV/IOG) of the Unit. Do not exceed this value when supplying the I/O power to the connected external devices. |
| **NX Unit power consumption** | The power consumption of the NX Unit power supply of the Unit. |
| **Current consumption from I/O power supply** | The current consumption from I/O power supply of the Unit. This value does not include the load current of any external connection loads or the current consumption of any connected external devices. |
| **Weight** | The weight of the Unit. |
| **Installation orientation and restrictions** | This is the installation orientation of the Unit. Any restrictions to specifications that result from the installation orientation are also given. |

*1. Refer to *5-4-3 Calculating the Number of Connections* on page 5-37 for how to calculate the number of connections.

*2. The number of CIP Safety connections that can be actually set depends on the maximum number of routable CIP Safety connections of the NX-series CPU Unit.
For NX102 CPU Units, the maximum number of routable CIP Safety connections is 16.

## ● NX-SL3300

| Item | Specification |
|---|---|
| **Maximum number of safety I/O points** | 256 points |
| **Program capacity** | 512 KB |
| **Number of safety I/O connections** | 32 |
| **Number of CIP Safety originator connections** | 0 |
| **Number of CIP Safety target connections** | 0 |
| **Number of originators that can be connected with a multi-cast connection** | 0 |
| **Number of FSoE master connections** | 32 |
| **I/O refreshing method** | Free-Run refreshing |
| **External connection terminals** | None |
| **Indicators** | [FS] LED, [VALID] LED, [DEBUG] LED, [TS] LED, [RUN] LED   |
| **Hardware switch settings** | None |
| **Dimensions (mm)** | 30 × 100 × 71 mm (W × H × D) |
| **I/O power supply method** | Not supplied. |
| **Current capacity of I/O power supply terminals** | No I/O power supply terminals |

| Item | Specification |
|---|---|
| **NX Unit power consumption*1** | • Connected to a CPU Unit<br>1.25 W max.<br>• Connected to a Communications Coupler Unit<br>0.90 W max. |
| **Current consumption from I/O power supply** | No consumption |
| **Weight** | 75 g max. |
| **Installation orientation and restrictions** | Installation orientation:<br>• Connected to a CPU Unit*2<br>Possible in the upright installation orientation.<br>• Connected to a Communications Coupler Unit<br>Six possible orientations.<br>Restriction: None. |

*1. The cable length for the Units that supply power to the corresponding Unit must be up to 20 m.
*2. Only NX502 CPU Units and NX102 CPU Units can be connected. NX1P2 CPU Units cannot be connected.

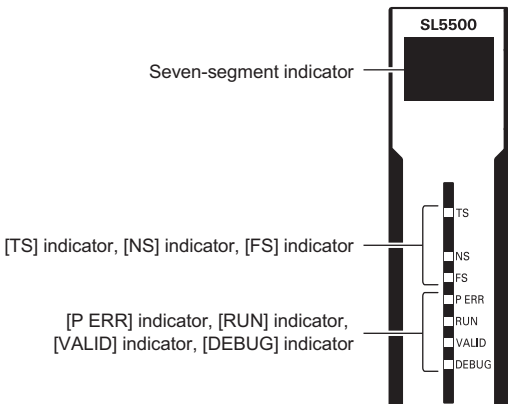
## ● NX-SL3500

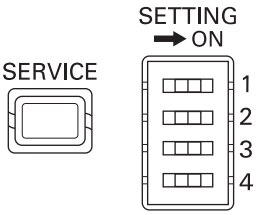
| Item | Specification |
|---|---|
| **Maximum number of safety I/O points** | 1,024 points |
| **Program capacity** | 2,048 KB |
| **Number of safety I/O connections** | 128 |
| **Number of CIP Safety originator connections** | 0 |
| **Number of CIP Safety target connections** | 0 |
| **Number of originators that can be connected with a multi-cast connection** | 0 |
| **Number of FSoE master connections** | 128 |
| **I/O refreshing method** | Free-Run refreshing |
| **External connection terminals** | None |
| **Indicators** | [FS] LED, [VALID] LED, [DEBUG] LED, [TS] LED, [RUN] LED<br>  |
| **Hardware switch settings** | None |
| **Dimensions (mm)** | 30 × 100 × 71 mm (W × H × D) |
| **I/O power supply method** | Not supplied. |
| **Current capacity of I/O power supply terminals** | No I/O power supply terminals |
| **NX Unit power consumption*1** | • Connected to a CPU Unit<br>1.25 W max.<br>• Connected to a Communications Coupler Unit<br>0.90 W max. |
| **Current consumption from I/O power supply** | No consumption |
| **Weight** | 75 g max. |

| Item | Specification |
|---|---|
| **Installation orientation and restrictions** | Installation orientation:<br>• Connected to a CPU Unit*2<br>Possible in the upright installation orientation.<br>• Connected to a Communications Coupler Unit<br>Six possible orientations.<br>Restriction: None. |

*1. The cable length for the Units that supply power to the corresponding Unit must be up to 20 m.
*2. Only NX502 CPU Units and NX102 CPU Units can be connected. NX1P2 CPU Units cannot be connected.

## ● NX-SL5500

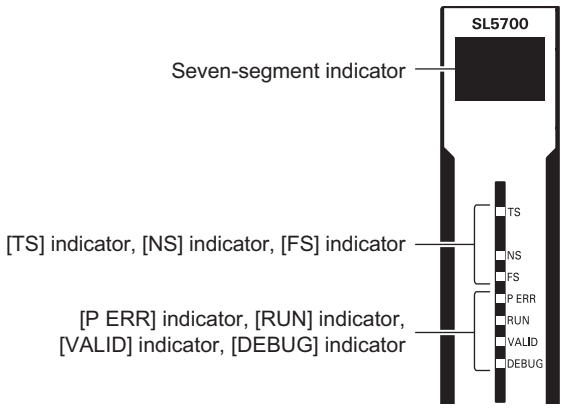
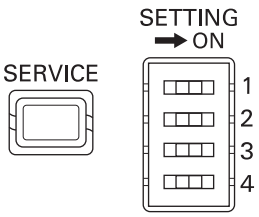
| Item | Specification |
|---|---|
| **Maximum number of safety I/O points** | 1,024 points |
| **Program capacity** | 2,048 KB |
| **Number of safety I/O connections** | 128 |
| **Number of CIP Safety originator connections** | 128 |
| **Number of CIP Safety target connections** | 4 |
| **Number of originators that can be connected with a multi-cast connection** | 8 |
| **CIP Safety maximum data size per connection** | 16 bytes |
| **Number of FSoE master connections** | 128 |
| **I/O refreshing method** | Free-Run refreshing |
| **External connection terminals** | None |
| **Indicators** | [TS] indicator, [NS] indicator, [FS] indicator, [P ERR] indicator, [RUN] indicator, [VALID] indicator, [DEBUG] indicator, seven-segment indicator<br>  |

| Item | Specification |
|---|---|
| **Hardware switch settings** | [SERVICE] switch, [SETTING] switch<br>  |
| **Dimensions (mm)** | 30 × 100 × 71 mm (W × H × D) |
| **I/O power supply method** | Not supplied. |
| **Current capacity of I/O power supply terminal** | No I/O power supply terminals |
| **NX Unit power consumption*1** | 3.35 W max. |
| **Current consumption from I/O power supply** | No consumption |
| **Weight** | 130 g max. |
| **Installation orientation and restrictions*2** | Installation orientation: Upright installation<br>Restriction: None. |

*1. The cable length for the Units that supply power to the corresponding Unit must be up to 20 m.
*2. Only NX502 CPU Units, NX102 CPU Units, and Communication Control Units can be connected. NX1P2 CPU Units or Communications Coupler Units cannot be connected.

## ● NX-SL5700

| Item | Specification |
|---|---|
| **Maximum number of safety I/O points** | 2,032 points |
| **Program capacity** | 4,096 KB |
| **Number of safety I/O connections** | 254 |
| **Number of CIP Safety originator connections** | 254 |
| **Number of CIP Safety target connections** | 4 |
| **Number of originators that can be connected with a multi-cast connection** | 8 |
| **CIP Safety maximum data size per connection** | 16 bytes |
| **Number of FSoE master connections** | 254 |
| **I/O refreshing method** | Free-Run refreshing |
| **External connection terminals** | None |

| Item | Specification |
|---|---|
| **Indicators** | [TS] indicator, [NS] indicator, [FS] indicator, [P ERR] indicator, [RUN] indicator, [VALID] indicator, [DEBUG] indicator, seven-segment indicator<br>  |
| **Hardware switch settings** | [SERVICE] switch, [SETTING] switch<br>  |
| **Dimensions (mm)** | 30 × 100 × 71 (W × H × D) |
| **I/O power supply method** | Not supplied. |
| **Current capacity of I/O power supply terminals** | No I/O power supply terminals |
| **NX Unit power consumption*1** | 3.35 W |
| **Current consumption from I/O power supply** | No consumption |
| **Weight** | 130 g max. |
| **Installation orientation and restrictions*2** | Installation orientation: Upright installation<br>Restriction: None. |

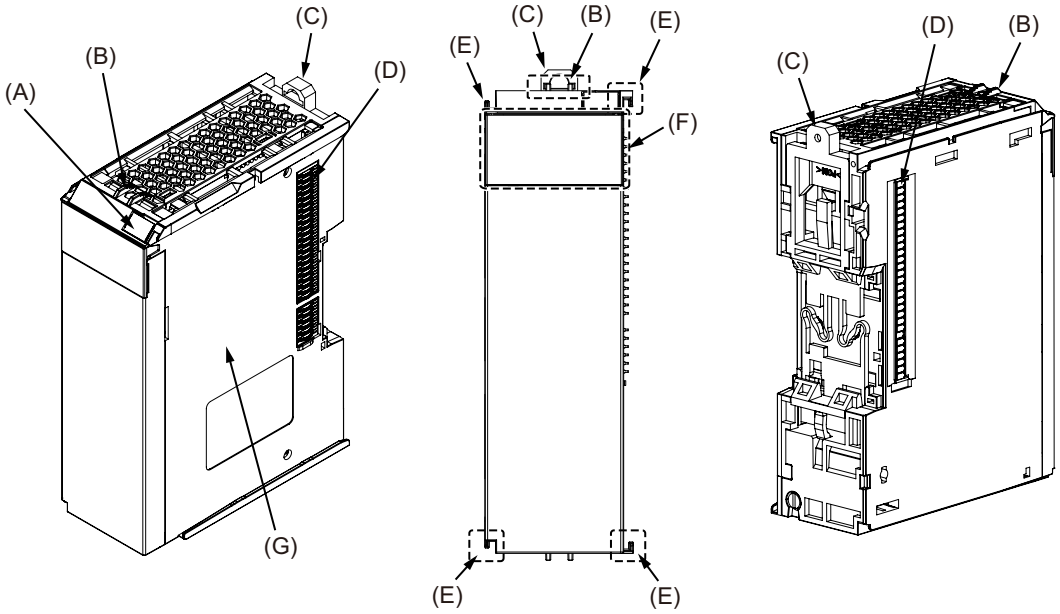
*1. The cable length for the Units that supply power to the corresponding Unit must be up to 20 m.

*2. Only NX502 CPU Units, NX102 CPU Units, and Communication Control Units can be connected. NX1P2 CPU Units or Communications Coupler Units cannot be connected.

## 2-1-2 Part Names and Functions

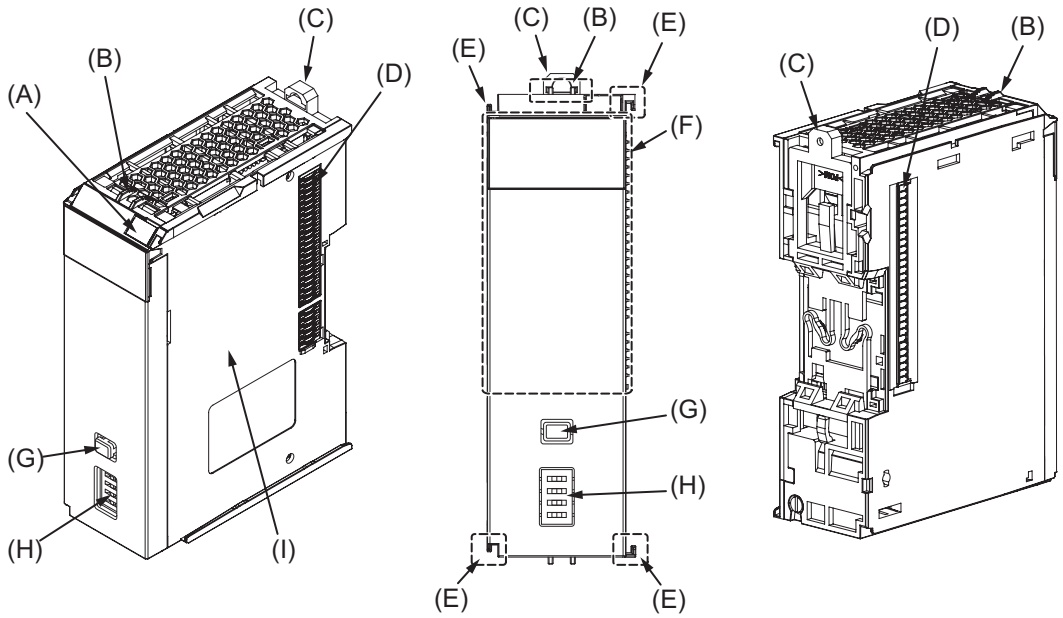
This section describes the names and functions of the Safety CPU Unit components.

## ● NX-SL3300 or NX-SL3500




| Letter | Name | Function |
|---|---|---|
| (A) | Marker attachment locations | The locations where markers are attached. The markers made by OMRON are installed for the factory setting. Commercially available markers can also be installed.<br>Refer to *3-1-2 Attaching Markers* on page 3-4. |
| (B) | Protrusions for removing the Unit | The protrusions to hold when removing the Unit. |
| (C) | DIN Track mounting hook | These hooks are used to mount the NX Unit to a DIN Track. |
| (D) | NX bus connector | This is the NX-series bus connector. |
| (E) | Unit hookup guides | These guides are used to connect two Units. |
| (F) | Indicators | The indicators show the current operating status and power supply status of the Safety CPU Unit.<br>Refer to *2-1-3 Indicators* on page 2-10. |
| (G) | Unit specifications | The specifications of the Safety CPU Unit are given. |

● **NX-SL5500/SL5700**




| Letter | Name | Function |
|---|---|---|
| (A) | Marker attachment location | The locations where markers are attached. The markers made by OMRON are installed for the factory setting. Commercially available markers can also be installed.<br>Refer to *3-1-2 Attaching Markers* on page 3-4. |
| (B) | Protrusions for removing the Unit | The protrusions to hold when removing the Unit. |
| (C) | DIN Track mounting hook | This hook is used to mount the NX Unit to a DIN Track. |
| (D) | NX bus connector | This is the NX-series bus connector. |
| (E) | Unit hookup guides | These guides are used to connect two Units. |
| (F) | Indicators | The indicators show the current operating status and power supply status of the Safety CPU Unit.<br>Refer to *2-1-3 Indicators* on page 2-10. |
| (G) | Service switch | This switch is used for the start trigger of various functions. |
| (H) | DIP switch | This switch is used for the Safety Unit Restore and the safety data logging function.<br>Refer to *10-1-2 Safety Unit Restore Function* on page 10-4 for details on the Safety Unit Restore.<br>Refer to *11-3 Safety Data Logging Operation Procedure* on page 11-6 for details on the Safety Data Logging function. |
| (I) | Unit specifications | The specifications of the Safety CPU Unit are given. |

## 2-1-3 Indicators

The Safety CPU Unit has indicators that show the current operating status and communications status.

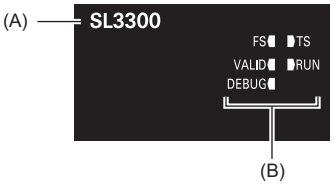
**WARNING**

Do not use the status of the indicators on the NX-series Safety Control Units for safety operations.
This will compromise the safety functions of the Unit and may cause serious injury in the event of an accident.

The appearance of the indicators on NX-SL3300 and NX-SL3500 is different depending on whether the lot number of the Unit represents the date of or before September 19, 2018, or the date of or after September 20, 2018. In this manual, indicators for lot numbers of or after September 20, 2018 are shown.

For details on the differences in appearance of the indicators, refer to *Differences in Appearance of the Indicators (NX-SL3300/SL3500)* on page 2-16.
Refer to *Notation of Unit Versions on Products* on page 38 for how to identify the lot number of the Unit.

## NX-SL3300/SL3500




| Letter | Name | Function |
|---|---|---|
| (A) | Model number display | Displays part of the model number of the Safety CPU Unit. |
| (B) | Indicators | Show the current operating status and communications status of the Safety CPU Unit. |

The rest of this section gives the indicator specifications.

### ● TS Indicator

The TS indicator shows the current status of the Safety CPU Unit and the communications status with the NX bus master.

The following table lists the possible states for this indicator and what they mean.

| Color | Status | | Meaning |
|---|---|---|---|
| Green |  | Lit. | The Unit is operating normally. |
| |  | Flashing (at 2-s intervals) | Initialization is in progress (from when the power supply is turned ON until RUN or PROGRAM mode is entered), or I/O allocation information data is being downloaded from the Sysmac Studio. |
| Red |  | Lit. | A hardware error, WDT error, or other critical error has occurred. |
| |  | Flashing (at 1-s intervals) | An NX bus communications error, I/O allocation information data error, or other recoverable minor error that is attributed to the NX bus has occurred. |

| Color | Status | | Meaning |
|---|---|---|---|
| --- |  | Not lit. | • There is no Unit power supply.<br>• Restarting the Unit<br>• Waiting for initialization to start |

## ● FS Indicator

The FS indicator shows the safety communications status and safety function status of the Safety CPU Unit.

The following table lists the possible states for this indicator and what they mean.

| Color | Status | | Meaning |
|---|---|---|---|
| Green |  | Lit. | All FSoE connections are established and there are no errors in any Safety CPU Unit functions. |
| |  | Flashing (at 1-s intervals) | One or more FSoE connections are not established or are currently being established and there are no errors in any Safety CPU Unit functions. |
| Red |  | Flashing (at 1-s intervals) | An FSoE communications error, program execution error, or other minor error that is attributed to the safety application has occurred.*1 |
| Green/Red |  | Alternates at 1-s intervals. | The safety application data has not been stored. |
| --- |  | Not lit. | Power is not being supplied or a fatal fault has occurred. |

*1. For approximately 30 seconds after the power supply to the Safety CPU Unit is turned ON, a *Safety Process Data Communications Initialization Error* event is not registered as an error if the Safety I/O Units do not exist. During that time, the FS indicator will flash green.

## ● RUN Indicator

The RUN indicator shows the execution status of the safety programs.

The following table lists the possible states for this indicator and what they mean.

| Color | Status | | Meaning |
|---|---|---|---|
| Green |  | Lit. | Execution of a safety program is in progress (operation is in progress in RUN mode, or DEBUG mode (RUN)). |
| |  | Flashing (at 1-s intervals) | Initialization is in progress (from when the power supply is turned ON until RUN or PROGRAM mode is entered). |
| --- |  | Not lit. | Operation is in progress in PROGRAM mode or DEBUG mode (STOPPED), or a fatal fault has occurred. |

## ● DEBUG Indicator

The DEBUG indicator shows the status whether the debug function is executable on Safety CPU Unit.

Refer to *6-8-3 Controlling BOOL Variables, Changing Present Values, and Using Forced Refreshing* on page 6-101 for the debug function of Safety CPU Unit.

The following table lists the possible states for this indicator and what they mean.

| Color | Status | | Meaning |
|---|---|---|---|
| Yellow |  | Lit. | Operation is in progress in DEBUG mode. |

| Color | Status | | Meaning |
|---|---|---|---|
| --- | □ | Not lit. | Operation is in progress in a mode other than DEBUG mode or a fatal fault has occurred. |

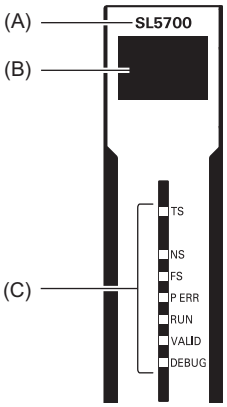
### ● VALID Indicator

The VALID indicator shows whether safety validation has been performed.
The following table lists the possible states for this indicator and what they mean.

| Color | Status | | Meaning |
|---|---|---|---|
| Yellow | ☀ | Lit. | Safety application data from the execution of the safety validation is stored in the non-volatile memory. |
| --- | □ | Not lit. | Safety application data from the execution of the safety validation is not stored in the non-volatile memory, or a fatal fault has occurred. |

Refer to *7-3 Operating Modes of the Safety CPU Unit* on page 7-11 for details on the relationship between the operating modes of the Safety CPU Unit and the indicators.

## NX-SL5500/SL5700




| Letter | Name | Function |
|---|---|---|
| (A) | Model number display | Displays part of the model number of the Safety CPU Unit. |
| (B) | Seven-segment Indicator | Displays detailed information on the Safety CPU Unit. |
| (C) | Indicators | Show the current operating status and communications status of the Safety CPU Unit. |

The rest of this section gives the indicator specifications.

### ● TS Indicator

The TS indicator shows the current status of the Safety CPU Unit and the communications status with the NX Bus Master.

The following table lists the possible states for this indicator and what they mean.

| Color | Status | | Meaning |
|---|---|---|---|
| Green |  | Lit | The Unit is operating normally. |
| |  | Flashing (at 2-s intervals) | Initialization is in progress (from when the power supply is turned ON until RUN or PROGRAM mode is entered), or I/O allocation information data is being downloaded from the Sysmac Studio. |
| Red |  | Lit | A hardware error, WDT error, or other critical error has occurred. |
| |  | Flashing (at 1-s intervals) | An NX bus communications error, I/O allocation information data error, or other recoverable minor error that is attributed to the NX bus has occurred. |
| --- |  | Not lit | • There is no Unit power supply<br>• The Unit is restarting<br>• Waiting for initialization to start |

## ● NS Indicator

The NS indicator shows the CIP Safety communications status of the Safety CPU Unit.

The following table lists the possible states for this indicator and what they mean.

| Color | Status | | Meaning |
|---|---|---|---|
| Green |  | Lit | The CIP Safety connections are established. |
| |  | Flashing (at 1-s intervals) | The CIP Safety connections are not established. |
| Red |  | Flashing (at 1-s intervals) | A CIP Safety communications error occurred.*1 |
| --- |  | Not lit | The CIP Safety communications are not executed. |

*1. For approximately 30 seconds after the power supply to the Safety CPU Unit is turned ON, the *CIP Safety Target Does Not Exist* is not registered as an error. During that time, the NS indicator will flash red.

## ● FS Indicator

The FS indicator shows the FSoE communications status of the Safety CPU Unit.

The following table lists the possible states for this indicator and what they mean.

| Color | Status | | Meaning |
|---|---|---|---|
| Green |  | Lit | All FSoE connections are established. |
| |  | Flashing (at 1-s intervals) | The FSoE connections are being established. |
| Red |  | Flashing (at 1-s intervals) | An FSoE communications error occurred.*1 |
| --- |  | Not lit | FSoE communications are not executed. |

*1. For approximately 30 seconds after the power supply to the Safety CPU Unit is turned ON, a *FSoE Master Connection Not Established Error* event is not registered as an error if the Safety I/O Units do not exist. During that time, the FS indicator will flash green.

## P ERR Indicator

The P ERR indicator shows the error status of the running program or settings of the Safety CPU Unit.

The following table lists the possible states for this indicator and what they mean.

| Color | Status | | Meaning |
|---|---|---|---|
| Red |  | Lit | The safety program, CIP Safety communications, and FSoE communications stopped due to the running program or settings, which resulted in error. |
| |  | Flashing (at 1-s intervals) | Local error occurred in the running program. |
| --- |  | Not lit | No errors in the running program or settings |

## RUN Indicator

The RUN indicator shows the execution status of the programs for the Safety CPU Unit.

The following table lists the possible states for this indicator and what they mean.

| Color | Status | | Meaning |
|---|---|---|---|
| Green |  | Lit | Execution of a safety program is in progress (operation is in progress in RUN mode, or DEBUG mode (RUN)). |
| |  | Flashing (at 1-s intervals) | Initialization is in progress (from when the power supply is turned ON until RUN or PROGRAM mode is entered). |
| --- |  | Not lit | Operation is in progress in PROGRAM mode or DEBUG mode (STOPPED), or a fatal fault has occurred. |

## DEBUG Indicator

The DEBUG indicator shows the status whether the debug function can be executed on Safety CPU Unit.
Refer to *6-8-3 Controlling BOOL Variables, Changing Present Values, and Using Forced Refreshing* on page 6-101 for the debug function of Safety CPU Unit.

The following table lists the possible states for this indicator and what they mean.

| Color | Status | | Meaning |
|---|---|---|---|
| Yellow |  | Lit | Operation is in progress in DEBUG mode. (the debug function can be executed) |
| --- |  | Not lit | Operation is in progress in a mode other than DEBUG mode or a fatal fault has occurred. (the debug function cannot be executed) |

## VALID Indicator

The VALID indicator shows whether safety validation has been performed on the safety application data in the Safety CPU Unit.

The following table lists the possible states for this indicator and what they mean.

| Color | Status | | Meaning |
|---|---|---|---|
| Yellow |  | Lit | Safety application data from the execution of the safety validation is stored in the non-volatile memory. |
| --- |  | Not lit | Safety application data from the execution of the safety validation is not stored in the non-volatile memory, or a fatal fault has occurred. |

### ● Seven-segment Indicator

The two-digit seven-segment indicator shows the detailed information on the Safety CPU Unit.

| Item | Meaning |
|---|---|
| At normal operation | It shows the lowest one byte of the safety signature for the safety program that is operating. If the safety signature is not confirmed, "—" is displayed.<br>While a CIP Safety originator connection is being established, the indicator flashes. When all the CIP Safety originator connections are established, the indicator is lit. |
| When an error occurs | Refer to *12-2-1 Troubleshooting the Main Errors in the Safety CPU Unit* on page 12-3. |
| When a signature code is checked | Refer to *A-14 Checking the Signature Code on the Seven-segment Indicator* on page A-102. |
| When the Safety Unit Restore is executed | Refer to *10-1-2 Safety Unit Restore Function* on page 10-4. |
| When the Safety Data Logging is executed | Refer to *11-4 Checking the Logging Status* on page 11-7. |

## Differences in Appearance of the Indicators (NX-SL3300/SL3500)

The appearance of the indicators on NX-SL3300 and NX-SL3500 is different depending on whether the lot number of the Unit represents the date of or before September 19, 2018, or the date of or after September 20, 2018.
The following table shows how the appearance of the indicators is different depending on the lot number.

| Indicator | Description | Lot number | |
|---|---|---|---|
| | | Date of or before Sep.19, 2018 | Date of or after Sep. 20, 2018 |
| TS indicator<br>FS indicator<br>RUN indicator<br>DEBUG indicator<br>VALID indicator | The shape of the light emitting parts is changed from a square to a pentagon. |  |  |



**Additional Information**

The appearance of the indicators on NX-SL5500 and NX-SL5700 is not different depending on the lot number of the Unit.

# 2-2 Safety Input Unit

This section describes the models and specifications of the Safety Input Units as well as the names and functions of the parts.



## 2-2-1 Models and Specifications

The Safety Input Unit specifications are described below.

### Models

The following table specifies the list of Safety Input Unit models.

| Model | Number of safety input points | Number of test output points | Internal I/O common | Rated input voltage | OMRON Special Safety Input Devices | Number of safety slave connections | I/O refreshing method |
|---|---|---|---|---|---|---|---|
| NX-SIH400 | 4 points | 2 points | Sinking inputs (PNP) | 24 VDC | Can be connected. | 1 | Free-Run refreshing |
| NX-SID800 | 8 points | 2 points | Sinking inputs (PNP) | 24 VDC | Cannot be connected. | 1 | Free-Run refreshing |

### General Specifications

This section provides the general specifications of the Safety Input Unit.

| Item | Specification |
|---|---|
| **Enclosure** | Mounted in a panel (open) |
| **Grounding method** | Ground to 100 Ω or less |

| Item | | Specification |
|---|---|---|
| **Operating environment** | **Ambient operating temperature** | 0 to 55°C |
| | **Ambient operating humidity** | 10% to 95% (with no condensation or icing) |
| | **Atmosphere** | Must be free from corrosive gases. |
| | **Ambient storage temperature** | −25 to 70°C (with no condensation or icing) |
| | **Altitude** | 2,000 m max. |
| | **Pollution degree** | 2 or less: Meets IEC 61010-2-201 |
| | **Noise immunity** | Conforms to IEC 61131-2, 2 kV (power supply line) |
| | **Insulation class** | CLASS III (SELV) |
| | **Overvoltage category** | Category II: Meets IEC 61010-2-201 |
| | **EMC immunity level** | Zone B |
| | **Vibration resistance** | Conforms to IEC 60068-2-6<br>5 to 8.4 Hz with amplitude of 3.5 mm<br>8.4 to 150 Hz, acceleration of 9.8 $m/s^2$<br>100 min. in each X, Y, and Z directions (10 sweeps of 10 min. each = 100 min. total) |
| | **Shock resistance** | Conforms to IEC 60068-2-27<br>147 $m/s^2$, 3 times in each X, Y, and Z directions |
| | **Isolation method** | Photocoupler isolation |
| | **Insulation resistance** | 20 MΩ between isolated circuits (at 100 VDC) |
| | **Dielectric strength** | 510 VAC between isolated circuits for 1 minute at a leakage current of 5 mA max. |
| **Installation method** | | DIN Track (IEC 60715 TH35-7.5/TH35-15) |

## Individual Specifications

This section provides the individual specifications of the Safety Input Unit.

### ● Datasheet Items for Safety Input Unit

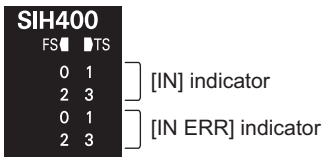
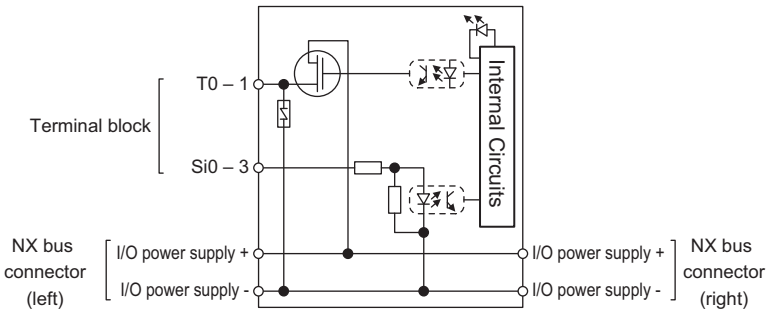
The following table gives the meaning of the datasheet items for the Safety Input Units.

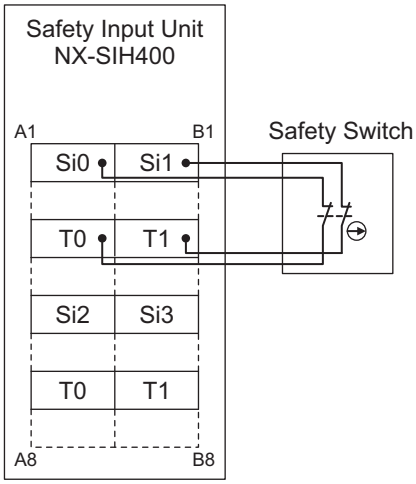
| Item | Specification |
|---|---|
| **Number of safety input points** | This is the number of safety input points on the Unit. |
| **Number of test output points** | This is the number of test output points on the Unit. The test output points are used with the safety input terminals. |
| **Internal I/O common** | This is the polarity that the Unit uses to connect to input devices. |
| **Rated input voltage** | This is the rated input voltage of the Unit. |
| **OMRON Special Safety Input Devices** | This tells whether the Unit supports the connection of OMRON Special Safety Input Devices (D40A-2 Non-contact Door Switches, E3FS Single Beam Safety Sensors, etc.). |
| **Number of safety slave connections** | This is the number of slave connections that the Unit can have through FSoE. You can perform communications with one FSoE master device per connection. |
| **I/O refreshing method** | This is the I/O refreshing method for the Unit. |
| **External connection terminals** | This is the type of terminal block and connector that is used to wire the Unit. This specification includes the number of terminals for a screwless clamping terminal block. |

| Item | Specification |
|---|---|
| **Indicators** | This gives the names and the layout of the indicators on the Unit. |
| **Safety input current** | This is the input current at the rated voltage of the safety inputs on the Unit. |
| **Safety input ON voltage** | This is the input voltage at which the safety inputs on the Unit turn ON. |
| **Safety input OFF voltage/OFF current** | These are the input voltage and input current at which the safety inputs on the Unit turn OFF. |
| **Test output type** | This is the polarity that the Unit uses to connect to devices. |
| **Test output rated current** | This is the maximum load current for each test output on the Unit. |
| **Test output ON residual voltage** | This is the residual voltage when the test output on the Unit is ON. |
| **Test output leakage current** | This is the leakage current when the test output on the Unit is OFF. |
| **Dimensions (mm)** | These are the external dimensions of the Unit. The dimensions are given in the form W × H × D. The dimensions are given in "millimeters". |
| **Isolation method** | This is the method that is used to isolate the input circuits from the internal circuits of the Unit. |
| **Insulation resistance** | This is the insulation resistance between the input circuits and the internal circuits of the Unit. |
| **Dielectric strength** | This is the dielectric strength between the input circuits and the internal circuits of the Unit. |
| **I/O power supply method** | This is the method that is used to supply I/O power to the Unit. |
| **Current capacity of I/O power supply terminal** | This is the current capacity of the I/O power supply terminals (IOV/IOG) on the Unit. When you supply I/O power to external devices that are connected to the Unit, make sure that the total power does not exceed this value. |
| **NX Unit power consumption** | This is the power consumption of the Unit from the NX bus power supply. |
| **Current consumption from I/O power supply** | This is the current consumption of the Unit from the I/O power supply. This value does not include the load current of any external connection loads or the current consumption of any connected external devices. |
| **Weight** | This is the weight of the Unit. |
| **Circuit layout** | This is the internal circuits of the Unit. |
| **Terminal connection diagram** | This is the connection diagram between the Unit and connected external devices. |
| **Installation orientation and restrictions** | This is the installation orientation of the Unit. If the installation orientation imposes any restrictions on the specifications, those restrictions are also described. |
| **Protective functions** | These are the protective functions that are supported by the Unit. |

## ● NX-SIH400

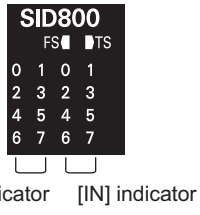
| Item | Specification |
|---|---|
| **Number of safety input points** | 4 points |
| **Number of test output points** | 2 points |
| **Internal I/O common** | PNP (sinking inputs) |
| **Rated input voltage** | 24 VDC (20.4 to 28.8 VDC) |
| **OMRON Special Safety Input Devices** | Can be connected. |
| **Number of safety slave connections** | 1 |
| **I/O refreshing method** | Free-Run refreshing |
| **External connection terminals** | Screwless clamping terminal block (8 terminals) |

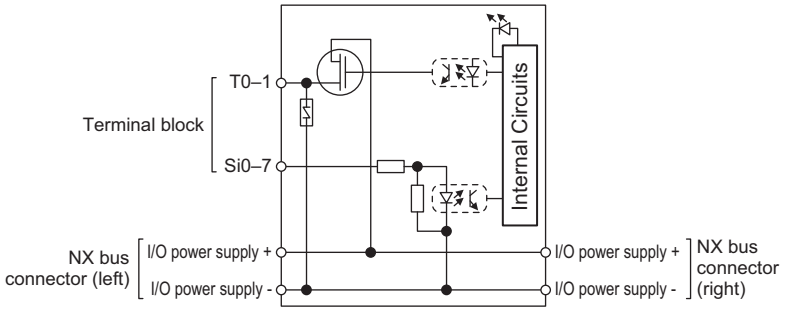
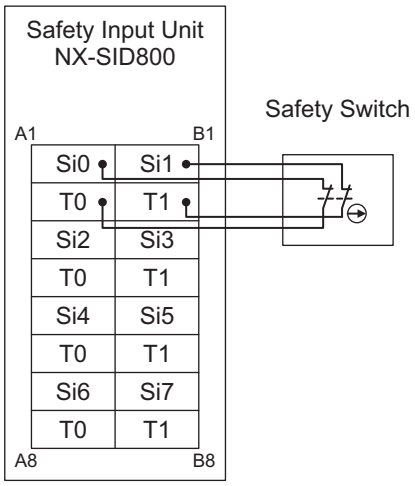
| Item | Specification |
|---|---|
| **Indicators** | [TS] indicator, [FS] indicator, [IN] indicator, [IN ERR] indicator<br>  |
| **Safety input current** | 4.5 mA typical |
| **Safety input ON voltage** | 11 VDC min. |
| **Safety input OFF voltage/OFF current** | 5 VDC max./1 mA max. |
| **Test output type** | Sourcing outputs (PNP) |
| **Test output rated current** | 25 mA max. |
| **Test output ON residual voltage** | 1.2 V max. (IOV and all output terminals) |
| **Test output leakage current** | 0.1 mA max. |
| **Dimensions (mm)** | 12 × 100 × 71 (W × H × D) |
| **Isolation method** | Photocoupler isolation |
| **Insulation resistance** | 20 MΩ min. between isolated circuits (at 100 VDC) |
| **Dielectric strength** | 510 VAC between isolated circuits for 1 minute at a leakage current of 5 mA max. |
| **I/O power supply method** | Power supplied from the NX bus |
| **Current capacity of I/O power supply terminals** | No applicable terminals. |
| **NX Unit power consumption** | • Connected to a CPU Unit or a Communication Control Unit<br>1.10 W max.<br>• Connected to a Communications Coupler Unit<br>0.70 W max. |
| **Current consumption from I/O power supply** | 20 mA max. |
| **Weight** | 70 g max. |
| **Circuit layout** |  |

| Item | Specification |
|---|---|
| **Terminal connection diagram** | Si0 to Si3: Safety input terminals<br>T0, T1: Test output terminals<br>  <br>Refer to *4-3-1 Safety Input Function* on page 4-10 for details. |
| **Installation orientation and restrictions** | Installation orientation:<br>• Connected to a CPU Unit or a Communication Control Unit *1<br>Possible in the upright installation orientation.<br>• Connected to a Communications Coupler Unit<br>Six possible orientations.<br>Restriction: Maximum ambient temperature is 50°C for any orientation other than upright installation. |
| **Protective functions** | Overvoltage protection circuit and short detection (test outputs) |

*1. Only NX502 CPU Units, NX102 CPU Units, and Communication Control Units can be connected. NX1P2 CPU Units cannot be connected.

## ● NX-SID800

| Item | Specification |
|---|---|
| **Number of safety input points** | 8 points |
| **Number of test output points** | 2 points |
| **Internal I/O common** | PNP (sinking inputs) |
| **Rated input voltage** | 24 VDC (20.4 to 28.8 VDC) |
| **OMRON Special Safety Input Devices** | Cannot be connected. |
| **Number of safety slave connections** | 1 |
| **I/O refreshing method** | Free-Run refreshing |
| **External connection terminals** | Screwless clamping terminal block (16 terminals) |
| **Indicators** | [TS] indicator, [FS] indicator, [IN] indicator, [IN ERR] indicator<br>  |
| **Safety input current** | 3.0 mA typical |
| **Safety input ON voltage** | 15 VDC min. |

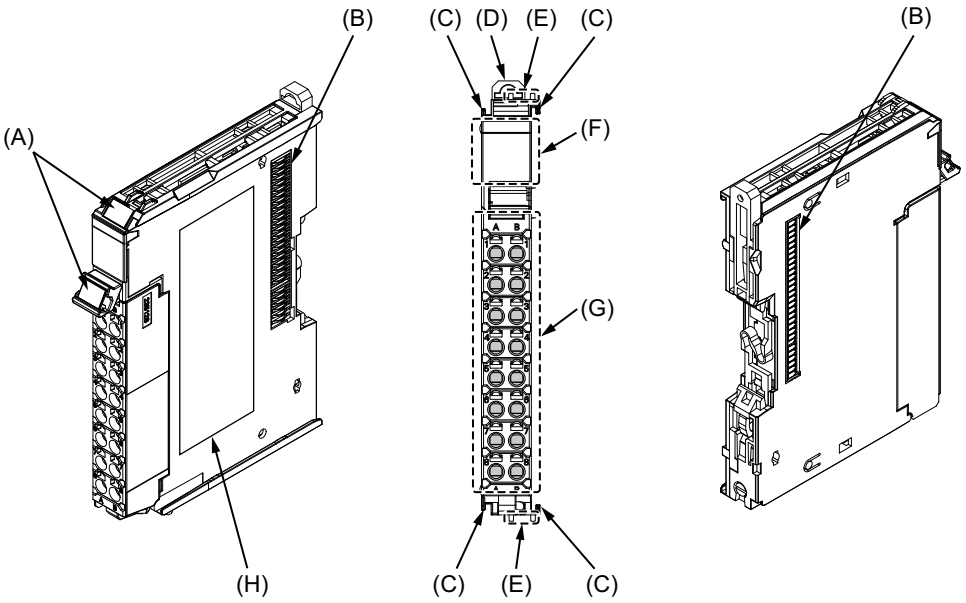
| Item | Specification |
|---|---|
| **Safety input OFF voltage/OFF current** | 5 VDC max./1 mA max. |
| **Test output type** | Sourcing outputs (PNP) |
| **Test output rated current** | 50 mA max. |
| **Test output ON residual voltage** | 1.2 V max. (IOV and all output terminals) |
| **Test output leakage current** | 0.1 mA max. |
| **Dimensions (mm)** | 12 × 100 × 71 (W × H × D) |
| **Isolation method** | Photocoupler isolation |
| **Insulation resistance** | 20 MΩ min. between isolated circuits (at 100 VDC) |
| **Dielectric strength** | 510 VAC between isolated circuits for 1 minute at a leakage current of 5 mA max. |
| **I/O power supply method** | Power supplied from the NX bus |
| **Current capacity of I/O power supply terminals** | No applicable terminals. |
| **NX Unit power consumption** | • Connected to a CPU UNIT or a Communication Control Unit<br>1.10 W max.<br>• Connected to a Communications Coupler Unit<br>0.75 W max. |
| **Current consumption from I/O power supply** | 20 mA max. |
| **Weight** | 70 g max. |
| **Circuit layout** |  |
| **Terminal connection diagram** | Si0 to Si7: Safety input terminals<br>T0, T1: Test output terminals<br>  <br>Refer to *4-3-1 Safety Input Function* on page 4-10 for details. |

| Item | Specification |
|---|---|
| **Installation orientation and restrictions** | Installation orientation:<br>• Connected to a CPU Unit or a Communication Control Unit *1<br>Possible in the upright installation orientation.<br>• Connected to a Communications Coupler Unit<br>Six possible orientations.<br>Restriction: Maximum ambient temperature is 50°C for any orientation other than upright installation. |
| **Protective functions** | Overvoltage protection circuit and short detection (test outputs) |

*1. Only NX502 CPU Units, NX102 CPU Units, and Communication Control Units can be connected. NX1P2 CPU Units cannot be connected.

## 2-2-2 Part Names and Functions

This section provides the names and functions of the parts of the Safety Input Unit.

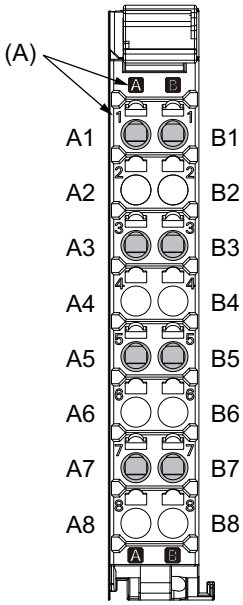



| Letter | Name | Function |
|---|---|---|
| (A) | Marker attachment location | The locations where markers are attached. The markers made by OMRON are installed for the factory setting. Commercially available markers can also be installed.<br>Refer to *3-1-2 Attaching Markers* on page 3-4. |
| (B) | NX bus connector | This is the NX-series bus connector. |
| (C) | Unit hookup guides | These guides are used to connect two Units. |
| (D) | DIN Track mounting hooks | These hooks are used to mount the NX Unit to a DIN Track. |
| (E) | Protrusions for removing the Unit | The protrusions to hold when removing the Unit. |
| (F) | Indicators | The indicators show the current operating status of the Safety Input Unit or signal input status.<br>Refer to *2-2-3 Indicators* on page 2-25. |
| (G) | Terminal block | The terminal block is used to connect external devices. |
| (H) | Unit specifications | The specifications of the Safety Input Unit are given here. |

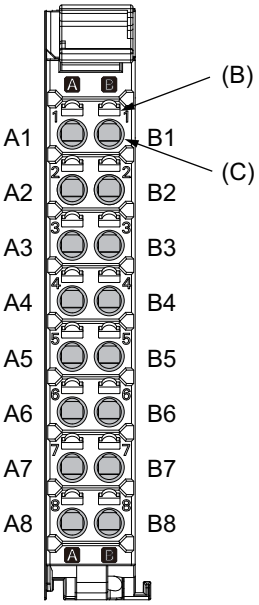
## Terminal Blocks

There are two models of screwless clamping terminal blocks: NX-TB□□□2 and NX-TB□□□1. The following models of Terminal Blocks can be mounted to Safety Input Units.

NX-TB□□□2

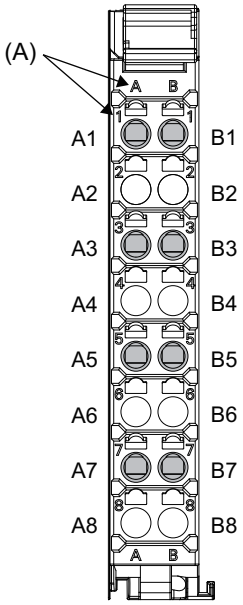



8-terminal type

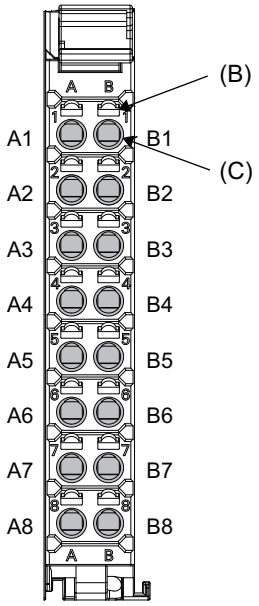



16-terminal type

NX-TB□□□1




8-terminal type




16-terminal type

| Letter | Name | Function |
|---|---|---|
| (A) | Terminal number indications | The terminal numbers are given by column letters A and B, and row numbers 1 to 8. The combination of the "column" and "row" gives the terminal numbers from A1 to A8 and B1 to B8.<br>The terminal number indicators are the same regardless of the number of terminals on the terminal block, as shown above. |
| (B) | Release holes | Insert a flat-blade screwdriver into these holes to connect or remove the wires. |
| (C) | Terminal holes | The wires are inserted into these holes. |

Terminal Blocks for Safety Input Units come in two types depending on the number of terminals that can be used. There are 8-terminal and 16-terminal Terminal Blocks.
The terminal block must have the same number of terminals that the Unit is designed for.

**Additional Information**

The 8-terminal type does not have terminal holes and release holes for the following terminals.
- A2, A4, A6, A8, B2, B4, B6, and B8.

There are two types of Terminal Blocks in terms of current capacity, 10 A for the NX-TB□□□2 Terminal Blocks and 4 A for the NX-TB□□□1 Terminal Blocks.
To differentiate between the two types of Terminal Blocks, use the terminal number column indications.
The Terminal Block with white letters on a dark background is the NX-TB□□□2 Terminal Block.
You can mount either type of Terminal Block to a Unit with a terminal current capacity of 4 A max.
You can only mount the NX-TB□□□2 Terminal Block to the Units that the current capacity specification of the terminals is greater than 4 A.

- Applicable Terminal Blocks for Each Unit Model

The following table gives the Terminal Blocks that are applicable to each Unit.

| Unit model number | Terminal block | | | |
|---|---|---|---|---|
| | Model number | Number of terminals | Ground terminal mark | Current capacity |
| NX-SIH400 | NX-TBA081 | 8 | Not provided | 4 A |
| | NX-TBA082 | | | 10 A |
| NX-SID800 | NX-TBA161 | 16 | Not provided | 4 A |
| | NX-TBA162 | | | 10 A |

**Precautions for Correct Use**

You can mount either NX-TB□□□1 or NX-TB□□□2 Terminal Block to a Unit with a terminal current capacity of 4 A.
Even if you mount an NX-TB□□□2 Terminal Block, the current specification does not change because the terminal current capacity of the Unit is 4 A.
Refer to the I/O power supply terminal current capacities given in *Individual Specifications* on page 2-18 for the terminal current capacity specifications of the Units.
Refer to *A-11 List of Screwless Clamping Terminal Block Models* on page A-97 or the model numbers of the Terminal Blocks.

## 2-2-3 Indicators

This section describes the indicators of the Safety Input Unit.

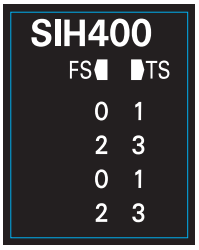
**⚠ WARNING**

Do not use the status of the indicators on the NX-series Safety Control Units for safety operations.
This will compromise the safety functions of the Unit and may cause serious injury in the event of an accident.

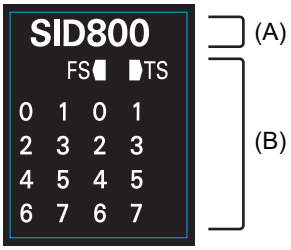
The appearance of the indicators is different depending on whether the lot number of the Unit represents the date of or before September 19, 2018, or the date of or after September 20, 2018. In this manual, indicators for lot numbers of or after September 20, 2018 are shown.

For details on the differences in appearance of the indicators, refer to *Differences in Appearance of the Indicators* on page 2-29.
Refer to *Notation of Unit Versions on Products* on page 38 for how to identify the lot number of the Unit.

The indicator pattern depends on the number of input points, as shown below.




Unit with 4 I/O Points



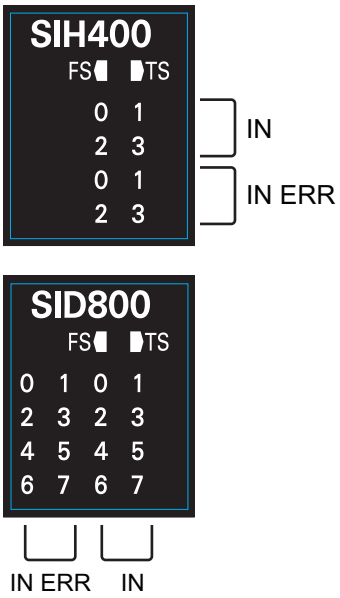

Unit with 8 I/O Points

| Letter | Name | Function |
|---|---|---|
| (A) | Model number display | Displays part of the model number of the Safety I/O Units.<br>The model number indication is red on all Safety Control Units. |
| (B) | Indicators | Show the current operating status and communications status of the Safety I/O Units. |

## Safety Input Unit Operation Status Indicators

Indicators to show the operation status of the Safety Input Unit are located in the center of the front side of the Safety Input Unit.

The following section describes the specifications of each indicator.




## ● TS Indicator

The TS indicator shows the current status of the Safety Input Unit and its communications status with the NX Bus Master.

The following table lists the possible states for this indicator and what they mean.

| Color | Status | | Meaning |
|---|---|---|---|
| Green | | Lit | The Unit is operating normally. |
| | | Flashing (at 2-s intervals) | Initializing, or I/O allocation information data is being downloaded from the Sysmac Studio. |
| Red | | Lit | A hardware error, WDT error, or other critical error has occurred. |
| | | Flashing (at 1-s intervals) | An NX bus communications error, I/O allocation information data error, or other recoverable minor error that is attributed to the NX bus has occurred. |
| --- | | Not lit | • There is no Unit power supply<br>• Restarting the Unit<br>• Waiting for initialization to start |

## ● FS Indicator

The FS indicator shows the FSoE communications status and safety function status of the Safety Input Unit.

The following table lists the possible states for this indicator and what they mean.

| Color | Status | | Meaning |
|---|---|---|---|
| Green | | Lit | The FSoE connections are established and there are no errors in any Safety I/O Units functions. |
| | | Flashing (at 1-s intervals) | The FSoE connections are being established. |
| Red | | Flashing (at 1-s intervals) | An FSoE communications error, safety input terminal error, or other minor error has occurred. |

| Color | Status | | Meaning |
|---|---|---|---|
| --- |  | Not lit | Power is not being supplied or a fatal fault has occurred. |

## IN Indicator

The IN indicator shows the signal input status of the safety input terminal.

The following table lists the possible states for this indicator and what they mean.
The table shows an example of safety input terminal Si01. The safety input terminal number is lit or not lit.

| Color | Status | | Meaning |
|---|---|---|---|
| Yellow |  | Lit | The safety input terminal is ON and there are no errors. |
| --- |  | Not lit | The safety input terminal is OFF or an error has occurred. |



**Additional Information**

If the lot number of the Unit represents the date of or before September 19, 2018, the indicators have square-shaped light emitters. For details on the differences in appearance of the indicators, refer to *Differences in Appearance of the Indicators* on page 2-29.

| Color | Status | | Meaning |
|---|---|---|---|
| Yellow |  | Lit | The safety input terminal is ON and there are no errors. |
| --- |  | Not lit | The safety input terminal is OFF or an error has occurred. |

## IN ERR Indicator

The IN ERR indicator shows the error status of the safety input terminal.

The following table lists the possible states for this indicator and what they mean.
The table shows an example of safety input terminal Si01. The safety input terminal number is lit, flashing or not lit.

| Color | Status | | Meaning |
|---|---|---|---|
| Red |  | Lit | An error has occurred in the safety input terminal. |
| |  | Flashing (at 1-s intervals) | An error has occurred in the safety input terminal for the other channel of the dual channel I/O. |
| --- |  | Not lit | There are no errors in the safety input terminal. |

**Additional Information**

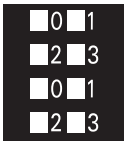
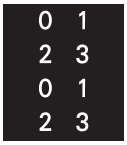
If the lot number of the Unit represents the date of or before September 19, 2018, the indicators have square-shaped light emitters. For details on the differences in appearance of the indicators, refer to *Differences in Appearance of the Indicators* on page 2-29.

| Color | Status | | Meaning |
|---|---|---|---|
| Red |  | Lit | An error has occurred in the safety input terminal. |
| |  | Flashing (at 1-s intervals) | An error has occurred in the safety input terminal for the other channel of the dual channel I/O. |
| --- |  | Not lit | An error has occurred in the safety input terminal. |

Refer to *12-2-1 Troubleshooting the Main Errors in the Safety CPU Unit* on page 12-3 for details on the relationship between the errors of the Safety Input Unit and the indicators.

## Differences in Appearance of the Indicators

The appearance of the indicators is different depending on whether the lot number of the Unit represents the date of or before September 19, 2018, or the date of or after September 20, 2018.
The following table shows how the appearance of the indicators is different depending on the lot number.

| Indicator | Description | Lot number | |
|---|---|---|---|
| | | Date of or before Sep.19, 2018 | Date of or after Sep. 20, 2018 |
| TS indicator<br>FS indicator | The shape of the light emitting parts is changed from a square to a pentagon. |  |  |
| IN indicator<br>IN ERR indicator | Square-shaped light emitters of the indicators are changed to light-emitting terminal numbers. |  |  |

# 2-3 Safety Output Unit

This section describes the models and specifications of the Safety Output Units as well as the names and functions of the parts.

## 2-3-1 Models and Specifications

The Safety Output Unit specifications are described below.

### Models

The following table specifies the list of the Safety Output Unit models.

| Model | Number of safety output points | Internal I/O common | Maximum load current | Rated voltage | Number of safety slave connections | I/O refreshing method |
|---|---|---|---|---|---|---|
| NX-SOH200 | 2 points | Sourcing outputs (PNP) | 2.0 A/point,<br>4.0 A/Unit at 40°C<br>2.5 A/Unit at 55°C | 24 VDC | 1 | Free-Run refreshing |
| NX-SOD400 | 4 points | Sourcing outputs (PNP) | 0.5 A/point and<br>2.0 A/Unit | 24 VDC | 1 | Free-Run refreshing |

### General Specifications

This section provides the general specifications of the Safety Output Unit.

| Item | Specification |
|---|---|
| **Enclosure** | Mounted in a panel (open) |
| **Grounding method** | Ground to 100 Ω or less |

| Item | | Specification |
|---|---|---|
| **Operating environment** | **Ambient operating temperature** | 0 to 55°C |
| | **Ambient operating humidity** | 10% to 95% (with no condensation or icing) |
| | **Atmosphere** | Must be free from corrosive gases. |
| | **Ambient storage temperature** | −25 to 70°C (with no condensation or icing) |
| | **Altitude** | 2,000 m max. |
| | **Pollution degree** | 2 or less: Meets IEC 61010-2-201 |
| | **Noise immunity** | Conforms to IEC 61131-2, 2 kV (power supply line) |
| | **Insulation class** | CLASS III (SELV) |
| | **Overvoltage category** | Category II: Meets IEC 61010-2-201 |
| | **EMC immunity level** | Zone B |
| | **Vibration resistance** | Conforms to IEC 60068-2-6<br>5 to 8.4 Hz with amplitude of 3.5 mm<br>8.4 to 150 Hz, acceleration of 9.8 $m/s^2$<br>100 min. in each X, Y, and Z directions (10 sweeps of 10 min. each = 100 min. total) |
| | **Shock resistance** | Conforms to IEC 60068-2-27<br>147 $m/s^2$, 3 times in each X, Y, and Z directions |
| | **Isolation method** | Photocoupler isolation |
| | **Insulation resistance** | 20 MΩ between isolated circuits (at 100 VDC) |
| | **Dielectric strength** | 510 VAC between isolated circuits for 1 minute at a leakage current of 5 mA max. |
| **Installation method** | | DIN Track (IEC 60715 TH35-7.5/TH35-15) |

## Individual Specifications

This section provides the individual specifications of the Safety Output Unit.

### ● Datasheet Items for Safety Output Unit

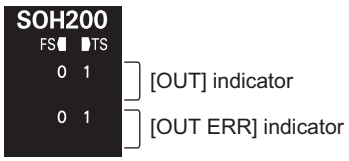
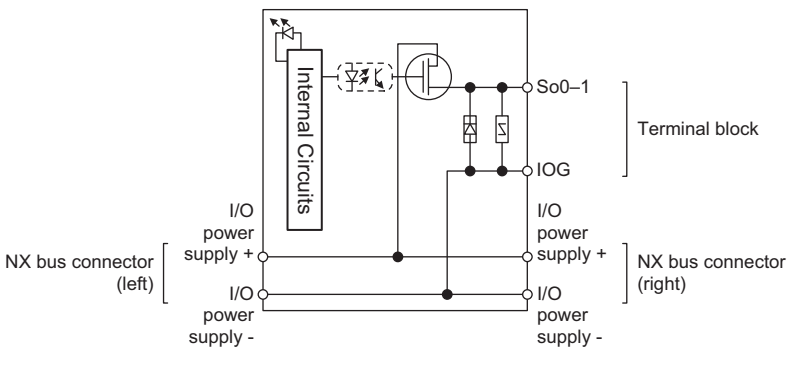
The following table gives the meaning of the datasheet items for the Safety Output Units.

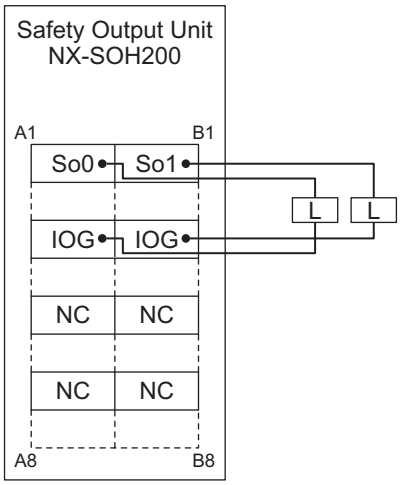
| Item | Specification |
|---|---|
| **Number of safety output points** | This is the number of safety output points on the Unit. |
| **Internal I/O common** | This is the polarity that the Unit uses to connect to output devices. |
| **Maximum load current** | This is the maximum load current for outputs on the Unit. A specification is given for each output and each Unit. |
| **Rated voltage** | This is the rated voltage of the outputs on the Unit. |
| **Number of safety slave connections** | This is the number of slave connections that the Unit can have through FSoE. You can perform communications with one FSoE master device per connection. |
| **I/O refreshing method** | The I/O refreshing methods that are used by the Unit. |
| **External connection terminals** | The type of terminal block and connector that is used for connecting the Unit. This specification includes the number of terminals for a screwless clamping terminal block. |
| **Indicators** | This gives the names and the layout of the indicators on the Unit. |

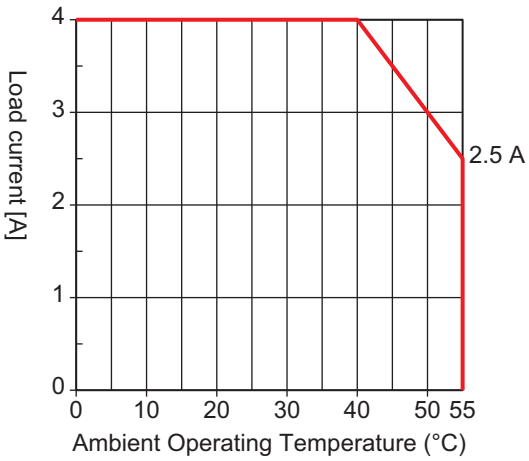
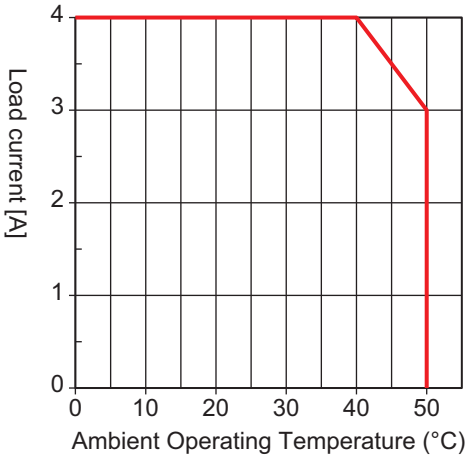
| Item | Specification |
|---|---|
| **Safety output rated current** | This is the maximum load current for safety outputs on the Unit. The inrush current of the external connection load must be lower than this value. |
| **Safety output ON residual voltage** | This is the residual voltage when a safety output on the Unit is ON. |
| **Safety output OFF residual voltage** | This is the residual voltage when a safety output on the Unit is OFF. |
| **Safety output leakage current** | This is the leakage current when a safety output on the Unit is OFF. |
| **Dimensions (mm)** | These are the external dimensions of the Unit. The dimensions are given in the form W × H × D. The dimensions are given in millimeters. |
| **Isolation method** | This is the method that is used to isolate the output circuits from the internal circuits of the Unit. |
| **Insulation resistance** | This is the insulation resistance between the output circuits and the internal circuits of the Unit. |
| **Dielectric strength** | This is the dielectric strength between the output circuits and the internal circuits of the Unit. |
| **I/O power supply method** | The method for supplying I/O power for the Unit. |
| **Current capacity of I/O power supply terminal** | This is the current capacity of the I/O power supply terminals (IOV/IOG) on the Unit. When you supply I/O power to external devices that are connected to the Unit, make sure that the total power does not exceed this value. |
| **NX Unit power consumption** | This is the power consumption of the Unit from the NX bus power supply. |
| **Current consumption from I/O power supply** | This is the current consumption of the Unit from the I/O power supply. This value does not include the load current of any external connection loads or the current consumption of any connected external devices. |
| **Weight** | This is the weight of the Unit. |
| **Circuit layout** | This is the internal circuits of the Unit. |
| **Terminal connection diagram** | This is the connection diagram between the Unit and external devices. |
| **Installation orientation and restrictions** | This is the installation orientation of the Unit. If the installation orientation imposes any restrictions on the specifications, those restrictions are also described. |
| **Protective functions** | These are the protective functions that are supported by the Unit. |

## ● NX-SOH200

| Item | Specification |
|---|---|
| **Number of safety output points** | 2 points |
| **Internal I/O common** | PNP (sourcing outputs) |
| **Maximum load current** | 2.0 A/point<br>4.0 A/Unit at 40°C<br>2.5 A/Unit at 55°C<br>The maximum load current depends on the installation orientation and ambient temperature. |
| **Rated input voltage** | 24 VDC (20.4 to 28.8 VDC) |
| **Number of safety slave connections** | 1 |
| **I/O refreshing method** | Free-Run refreshing |

| Item | Specification |
|---|---|
| **External connection terminals** | Screwless clamping terminal block (8 terminals) |
| **Indicators** | [TS] indicator, [FS] indicator, [OUT] indicator, [OUT ERR] indicator<br>  |
| **Safety output ON residual voltage** | 1.2 V max. (between IOV and all output terminals) |
| **Safety output OFF residual voltage** | 2 V max. (between IOG and all output terminals) |
| **Safety output leakage current** | 0.1 mA max. |
| **Dimensions (mm)** | 12 × 100 × 71 (W × H × D) |
| **Isolation method** | Photocoupler isolation |
| **Insulation resistance** | 20 MΩ min. between isolated circuits (at 100 VDC) |
| **Dielectric strength** | 510 VAC between isolated circuits for 1 minute at a leakage current of 5 mA max. |
| **I/O power supply method** | Power supplied from the NX bus |
| **Current capacity of I/O power supply terminals** | IOG: 2 A/terminal max. |
| **NX Unit power consumption** | • Connected to a CPU UNIT or a Communication Control Unit<br>1.05 W max.<br>• Connected to a Communications Coupler Unit<br>0.70 W max. |
| **Current consumption from I/O power supply** | 40 mA max. |
| **Weight** | 65 g max. |
| **Circuit layout** |  |

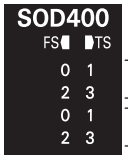
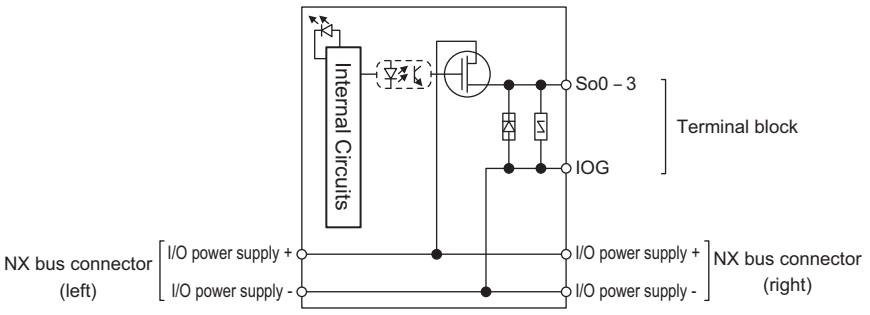
| Item | Specification |
|---|---|
| **Terminal connection diagram** | So0 to So1: Safety output terminals<br>IOG: I/O power supply 0 V<br>  <br>Refer to *4-3-2 Safety Output Function* on page 4-32 for details. |

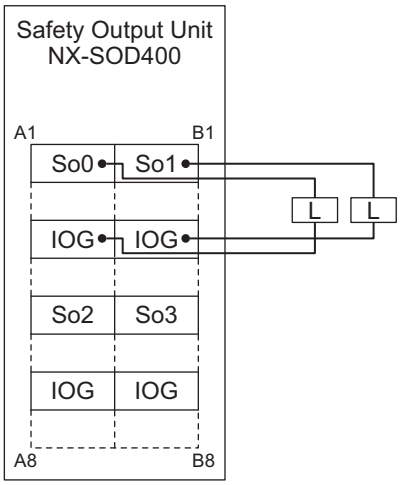
| Item | Specification |
|---|---|
| **Installation orientation and restrictions** | Installation orientation:<br>• Connected to a CPU Unit or a Communication Control Unit*1<br>Possible in the upright installation orientation.<br>• Connected to a Communications Coupler Unit<br>Six possible orientations.<br>Restriction: For upright installation, the ambient temperature is restricted as shown below according to the total Unit load current.<br>  <br><br>For all installation orientations other than upright installation, the ambient temperature is restricted as shown below according to the total Unit load current.<br>  <br> |
| **Protective functions** | Overvoltage protection circuit and short detection |

*1. Only NX502 CPU Units, NX102 CPU Units, and Communication Control Units can be connected. NX1P2 CPU Units cannot be connected.

## ● NX-SOD400

| Item | Specification |
|---|---|
| **Number of safety output points** | 4 points |
| **Internal I/O common** | PNP (sourcing outputs) |
| **Maximum load current** | 0.5 A/point and 2.0 A/Unit |
| **Rated input voltage** | 24 VDC (20.4 to 28.8 VDC) |
| **Number of safety slave connections** | 1 |

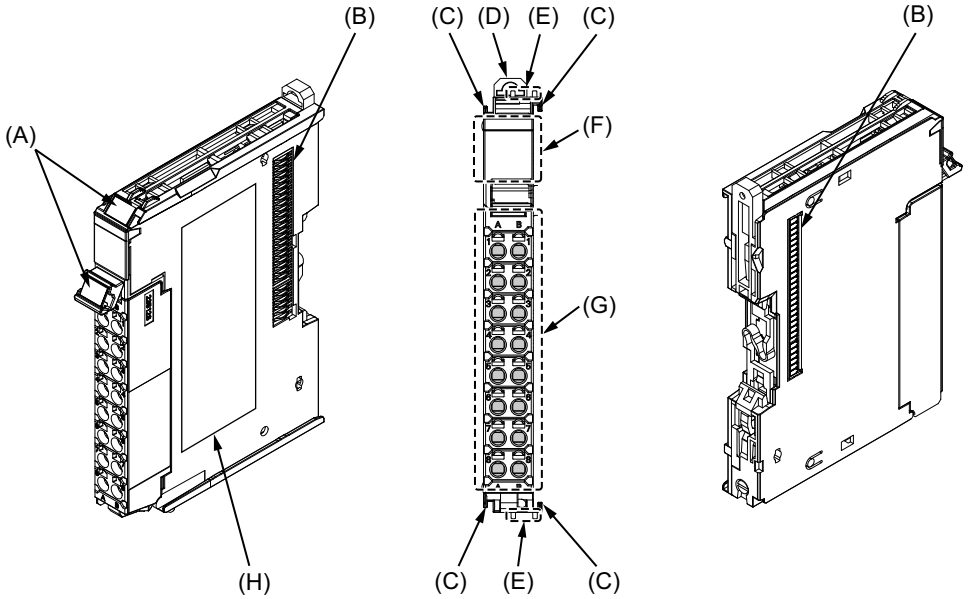
| Item | Specification |
|---|---|
| **I/O refreshing method** | Free-Run refreshing |
| **External connection terminals** | Screwless clamping terminal block (8 terminals) |
| **Indicators** | [TS] indicator, [FS] indicator, [OUT] indicator, [OUT ERR] indicator<br>  |
| **Safety output ON residual voltage** | 1.2 V max. (between IOV and all output terminals) |
| **Safety output OFF residual voltage** | 2 V max. (between IOG and all output terminals) |
| **Safety output leakage current** | 0.1 mA max. |
| **Dimensions (mm)** | 12 × 100 × 71 (W × H × D) |
| **Isolation method** | Photocoupler isolation |
| **Insulation resistance** | 20 MΩ min. between isolated circuits (at 100 VDC) |
| **Dielectric strength** | 510 VAC between isolated circuits for 1 minute at a leakage current of 5 mA max. |
| **I/O power supply method** | Power supplied from the NX bus |
| **Current capacity of I/O power supply terminals** | IOG (A3 and B3): 2 A/terminal max.<br>IOG (A7 and B7): 0.5 A/terminal max. |
| **NX Unit power consumption** | • Connected to a CPU UNIT or a Communication Control Unit<br>1.10 W max.<br>• Connected to a Communications Coupler Unit<br>0.75 W max. |
| **Current consumption from I/O power supply** | 60 mA max. |
| **Weight** | 65 g max. |
| **Circuit layout** |  |

| Item | Specification |
|---|---|
| **Terminal connection diagram** | So0 to So3: Safety output terminals<br>IOG: I/O power supply 0 V<br>  <br>Refer to *4-3-2 Safety Output Function* on page 4-32 for details. |
| **Installation orientation and restrictions** | Installation orientation:<br>• Connected to a CPU Unit or a Communication Control Unit*1<br>Possible in the upright installation orientation.<br>• Connected to a Communications Coupler Unit<br>Six possible orientations.<br>Restriction: None. |
| **Protective functions** | Overvoltage protection circuit and short detection |

*1. Only NX502 CPU Units, NX102 CPU Units, and Communication Control Units can be connected. NX1P2 CPU Units cannot be connected.

## 2-3-2 Part Names and Functions

This section provides the names and functions of the parts of the Safety Output Unit.

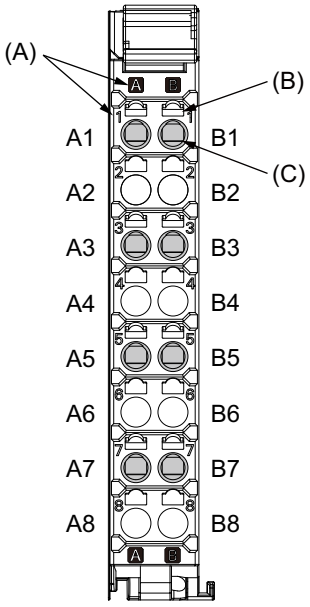


| Letter | Name | Function |
|---|---|---|
| (A) | Marker attachment location | The locations where markers are attached. The markers made by OMRON are installed for the factory setting. Commercially available markers can also be installed.<br>Refer to *3-1-2 Attaching Markers* on page 3-4. |
| (B) | NX bus connector | This is the NX-series bus connector. |
| (C) | Unit hookup guides | These guides are used to connect two Units. |
| (D) | DIN Track mounting hook | This hook is used to mount the NX Unit to a DIN Track. |
| (E) | Protrusions for removing the Unit | The protrusions to hold when removing the Unit. |
| (F) | Indicators | The indicators show the current operating status of the Safety Output Unit or signal I/O status.<br>Refer to *2-3-3 Indicators* on page 2-40. |
| (G) | Terminal block | The terminal block is used to connect external devices. |
| (H) | Unit specifications | The specifications of the Safety Output Unit are given. |

## Terminal Blocks

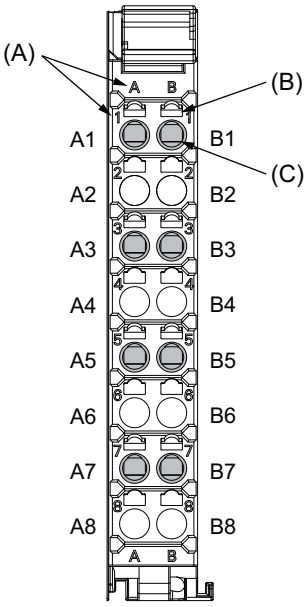
There are two models of screwless clamping terminal blocks: NX-TB□□□2 and NX-TB□□□1. The following models of Terminal Blocks can be mounted to Safety Output Units.

NX-TB□□□2




8-terminal type

NX-TB□□□1



8-terminal type

| Letter | Name | Function |
|---|---|---|
| (A) | Terminal number indications | The terminal numbers are given by column letters A and B, and row numbers 1 to 8. The combination of the "column" and "row" gives the terminal numbers from A1 to A8 and B1 to B8.<br>The terminal number indicators are the same regardless of the number of terminals on the terminal block, as shown above. |
| (B) | Release hole | Insert a flat-blade screwdriver into these holes to connect or remove the wires. |
| (C) | Terminal hole | The wires are inserted into these holes. |

Only 8-terminal type blocks can be inserted to the terminal blocks of Safety Output Units.
To prevent incorrect insertion, terminal blocks in any other types besides the 8-terminal type cannot be mounted.

**Additional Information**

The 8-terminal type does not have terminal holes and release holes for following terminal numbers.
- A2, A4, A6, A8, B2, B4, B6, and B8.

There are two types of Terminal Blocks in terms of current capacity, 10 A for the NX-TB□□□2 Terminal Blocks and 4 A for the NX-TB□□□1 Terminal Blocks.
To differentiate between the two types of Terminal Blocks, use the terminal number column indications.
The Terminal Block with white letters on a dark background is the NX-TB□□□2 Terminal Block.
You can mount either type of Terminal Block to a Unit with a terminal current capacity of 4 A.
You can only mount the NX-TB□□□2 Terminal Block to the Units that the current capacity specification of the terminals is greater than 4 A.

- Applicable Terminal Blocks for Each Unit Model
  The following table gives the Terminal Blocks that are applicable to each Unit.

| Unit model number | Terminal block | | | |
|---|---|---|---|---|
| | Model | Number of terminals | Ground terminal mark | Current capacity |
| NX-SOH200 | NX-TBA081 | 8 | Not provided | 4 A |
| | NX-TBA082 | | | 10 A |
| NX-SOD400 | NX-TBA081 | 8 | Not provided | 4 A |
| | NX-TBA082 | | | 10 A |

**Precautions for Correct Use**

You can mount either NX-TB□□□1 or NX-TB□□□2 Terminal Blocks to the Units that the current capacity specification of the terminals is 4 A or less.
Even if you mount an NX-TB□□□2 Terminal Block, the current specification does not change because the terminal current capacity of the Unit is 4 A.
Refer to the I/O power supply terminal current capacities given in *Individual Specifications* on page 2-31 for the terminal current capacity specifications of the Units.
Refer to *A-11 List of Screwless Clamping Terminal Block Models* on page A-97 for information on the models of terminal blocks.

## 2-3-3 Indicators

This section describes the indicators of the Safety Output Unit.

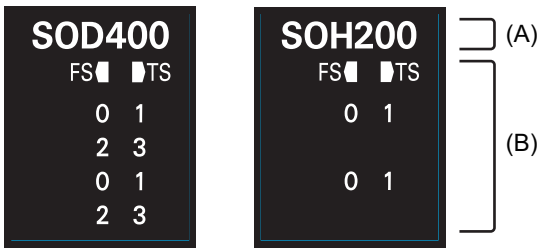
**⚠ WARNING**

Do not use the status of the indicators on the NX-series Safety Control Units for safety operations.
This will compromise the safety functions of the Unit and may cause serious injury in the event of an accident.

The appearance of the indicators is different depending on whether the lot number of the Unit represents the date of or before September 19, 2018, or the date of or after September 20, 2018. In this manual, indicators for lot numbers of or after September 20, 2018 are shown.

For details on the differences in appearance of the indicators, refer to *Differences in Appearance of the Indicators* on page 2-43.
Refer to *Notation of Unit Versions on Products* on page 38 for how to identify the lot number of the Unit.

The indicator pattern depends on the number of output points, as shown below.




Unit with 4 I/O Points　　Unit with 2 I/O Points

| Letter | Name | Function |
|---|---|---|
| (A) | Model number display | Displays part of the model number of the Safety I/O Units.<br>The model number indication is red on Safety Control Units. |

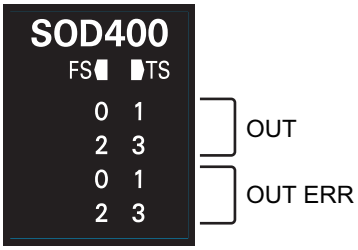
| Letter | Name | Function |
|---|---|---|
| (B) | Indicators | Show the current operating status and communications status of the Safety I/O Units. |

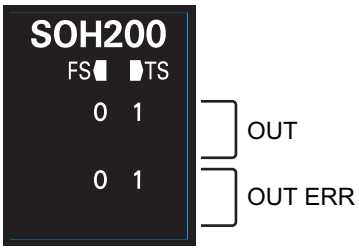


## Safety Output Unit Operation Status Indicators

There are the indicators to show the operation status of Safety Output Unit in the center of the front side of the Safety Output Unit.

The following section describes the specifications of each indicator.




### ● TS Indicator

The TS indicator shows the current status of the Safety Output Unit and its communications status with the NX Bus Master.

The following table lists the possible states for this indicator and what they mean.

| Color | Status | | Meaning |
|---|---|---|---|
| Green |  | Lit | The Unit is operating normally. |
| |  | Flashing (at 2-s intervals) | Initialization is in progress, or I/O allocation information data is being downloaded from the Sysmac Studio. |
| Red |  | Lit | A hardware error, WDT error, or other critical error has occurred. |
| |  | Flashing (at 1-s intervals) | An NX bus communications error, I/O allocation information data error, or other recoverable minor error that is attributed to the NX bus has occurred. |
| --- |  | Not lit | • There is no Unit power supply<br>• The Unit is restarting<br>• Waiting for initialization to start |

### ● FS Indicator

The FS indicator shows the FSoE communications status and safety function status of the Safety Output Unit.

The following table lists the possible states for this indicator and what they mean.

| Color | Status | | Meaning |
|---|---|---|---|
| Green |  | Lit | The FSoE connections are established and there are no errors in any Safety I/O Units functions. |
| |  | Flashing (at 1-s intervals) | The FSoE connections are being established. |
| Red |  | Flashing (at 1-s intervals) | An FSoE communications error, safety output terminal error, or other minor error has occurred. |
| --- |  | Not lit | Power is not being supplied or a fatal fault has occurred. |

## ● OUT Indicator

The OUT indicator shows the signal input status of the safety output terminal.

The following table lists the possible states for this indicator and what they mean.
The table shows an example of safety output terminal So01. The safety output terminal number is lit or not lit.

| Color | Status | | Meaning |
|---|---|---|---|
| Yellow |  | Lit | Safety output terminal is ON and there are no errors. |
| --- |  | Not lit | Safety output terminal is OFF or an error has occurred. |



### Additional Information

If the lot number of the Unit represents the date of or before September 19, 2018, the indicators have square-shaped light emitters. For details on the differences in appearance of the indicators, refer to *Differences in Appearance of the Indicators* on page 2-29.

| Color | Status | | Meaning |
|---|---|---|---|
| Yellow |  | Lit | Safety output terminal is ON and there are no errors. |
| --- |  | Not lit | Safety output terminal is OFF or an error has occurred. |

## ● OUT ERR Indicator

The OUT ERR indicator shows the error status of the safety output terminals.

The following table lists the possible states for this indicator and what they mean.
The table shows an example of safety output terminal So01. The safety output terminal number is lit, flashing or not lit.

| Color | Status | | Meaning |
|---|---|---|---|
| Red |  | Lit | An error has occurred in the safety output terminal. |
| |  | Flashing (at 1-s intervals) | An error has occurred in the safety output terminal for the other channel of the dual channel I/O. |
| --- |  | Not lit | There are no errors in the safety output terminal. |

**Additional Information**

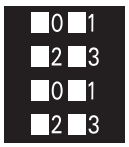
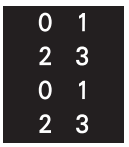
If the lot number of the Unit represents the date of or before September 19, 2018, the indicators have square-shaped light emitters. For details on the differences in appearance of the indicators, refer to *Differences in Appearance of the Indicators* on page 2-29.

| Color | Status | | Meaning |
|---|---|---|---|
| Red |  | Lit | An error has occurred in the safety output terminal. |
| |  | Flashing (at 1-s intervals) | An error has occurred in the safety output terminal for the other channel of the dual channel I/O. |
| --- |  | Not lit | There are no errors in the safety output terminal. |

Refer to *12-2-1 Troubleshooting the Main Errors in the Safety CPU Unit* on page 12-3 for details on the relationship between errors of the Safety Output Unit and the indicators.

## Differences in Appearance of the Indicators

The appearance of the indicators is different depending on whether the lot number of the Unit represents the date of or before September 19, 2018, or the date of or after September 20, 2018.
The following table shows how the appearance of the indicators is different depending on the lot number.

| Indicator | Description | Lot number | |
|---|---|---|---|
| | | Date of or before Sep.19, 2018 | Date of or after Sep. 20, 2018 |
| TS indicator<br>FS indicator | The shape of the light emitting parts is changed from a square to a pentagon. |  |  |
| OUT indicator<br>OUT ERR indicator | Square-shaped light emitters of the indicators are changed to light-emitting terminal numbers. |  |  |

# 2-4 PFH

This section gives the PFH ($PFH_D$) values of the NX-series Safety CPU Unit and the Safety I/O Units.

**Precautions for Correct Use**

Go to the following URL for the most recent PFH values: http://www.ia.omron.com/support/sistemalibrary/index.html

**Additional Information**

The NX-series Safety Control Unit is a Type B subsystem that is defined by IEC 61508 with HFT = 1 and SFF > 90%.

## ● Safety CPU Unit

| Model | PFH |
|---|---|
| NX-SL3300 | 3.1E-10 |
| NX-SL3500 | 3.0E-10 |
| NX-SL5500 | 5.0E-11 |
| NX-SL5700 | 5.0E-11 |

## ● Safety Input Unit

| Model | PFH |
|---|---|
| NX-SID800 | 1.5E-9*1<br>4.3E-10*2 |
| NX-SIH400 | 1.4E-9*1<br>3.1E-10*2 |

*1. This value includes PFH value of FSoE connection.

*2. This value does not include PFH value of FSoE connection.

## ● Safety Output Unit

| Model | PFH |
|---|---|
| NX-SOD400 | 1.6E-9*1<br>5.5E-10*2 |
| NX-SOH200 | 1.4E-9*1<br>3.6E-10*2 |

*1. This value includes PFH value of FSoE connection.

*2. This value does not include PFH value of FSoE connection.

# 3

# Installation and Wiring



This section describes how to install and wire the Safety Control Units.

**3-1 Installing Units ..... 3-2**
3-1-1 Installing NX Units ..... 3-2
3-1-2 Attaching Markers ..... 3-4
3-1-3 Removing Units ..... 3-5
3-1-4 Installation Orientation ..... 3-6
**3-2 Power Supply Types and Wiring ..... 3-8**
3-2-1 Applications of I/O Power Supply and Supply Methods ..... 3-8
3-2-2 Calculating the Total Current Consumption from I/O Power Supply ..... 3-9
**3-3 Wiring the Terminals ..... 3-10**
3-3-1 Wiring to the Screwless Clamping Terminal Blocks ..... 3-10
3-3-2 Checking Wiring ..... 3-23

# 3-1 Installing Units

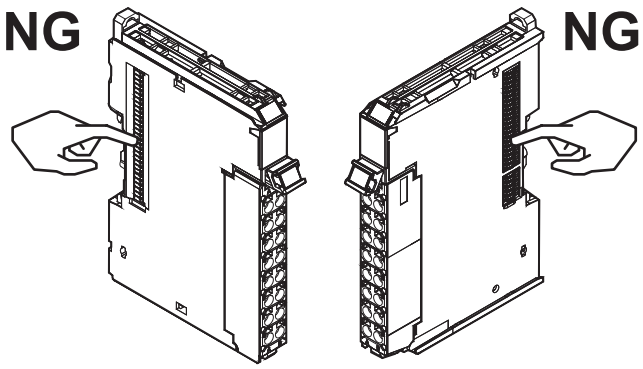
The NX-series Safety Control Units are installed in the same way as the NX Units. This section describes how to install and remove NX Units and how to attach markers.
Refer to the user's manual of the NX bus master, that the Safety Control Unit is to be connected to, for details on preparations for installation, installation in a control panel, etc.

## 3-1-1 Installing NX Units

This section describes how to mount two NX Units to each other.
Always turn OFF the power supply before you mount NX Units.
Always mount NX Units one at a time. If you attempt to mount multiple NX Units that are already connected together, the connections between the NX Units may separate from each other and fall.

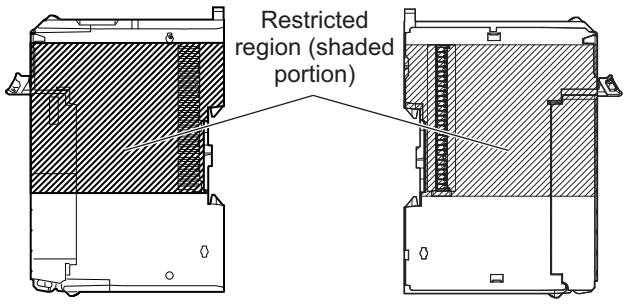
**Precautions for Safe Use**

- Do not apply labels or tape to the Unit. When the Unit is installed or removed, adhesive or scraps may adhere to the pins in the NX bus connector, which may result in malfunctions.
- Do not touch the pins in the NX bus connector on the Unit. Dirt may adhere to the pins in the NX bus connector, which may result in malfunctions.




Example: NX Unit (12 mm width)

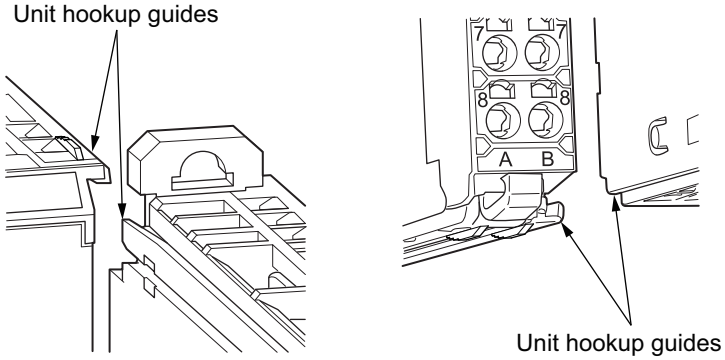
- Do not write on an NX Unit with ink within the restricted region that is shown in the following figure. Also do not get this area dirty. When the Unit is installed or removed, ink or dirt may adhere to the pins in the NX bus connector, which may result in malfunctions in the CPU Rack or the Slave Terminal.
  Refer to the user's manual for the connected CPU Unit, Communications Coupler Unit, or Communication Control Unit for details on the restricted region on the CPU Unit, Communications Coupler Unit, or Communication Control Unit.



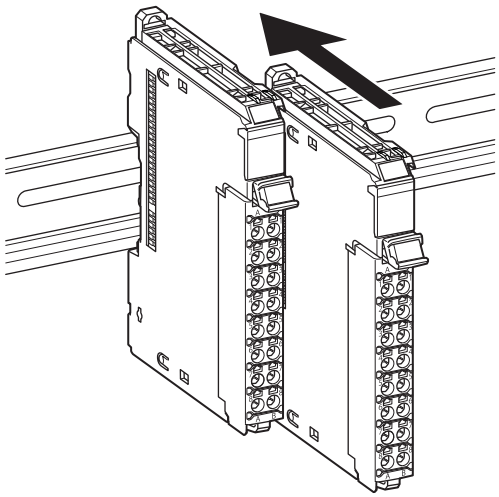
**Precautions for Correct Use**

- When you install an NX Unit, do not touch or bump the pins in the NX bus connector.
- When you handle an NX Unit, be careful not to apply any stress to the pins in the NX bus connector. If the NX Unit is installed and the power supply is turned ON when the pins in the NX bus connector are deformed, contact failure may cause malfunctions.

***1*** From the front of the previously mounted NX Unit, engage the Unit hookup guides on a new Unit with the Unit hookup guides on the previously mounted NX Unit.




***2*** Slide the NX Unit in on the hookup guides.



***3*** Press the NX Unit with a certain amount of force against the DIN Track until you hear the DIN Track mounting hook lock into place.

When you mount the NX Unit, it is not necessary to unlock the DIN Track mounting hook on the NX Unit.

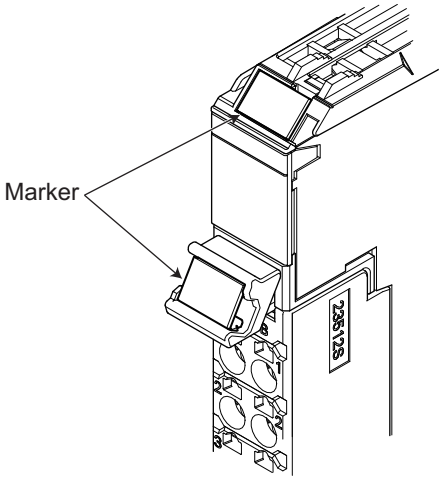
After you mount the NX Unit, make sure that it is locked to the DIN Track.

**Additional Information**

- It is not normally necessary to unlock the DIN Track mounting hook when you mount the NX Unit. If you mount an NX Unit on a DIN Track that is not one of the recommended DIN Tracks, the DIN Track mounting hook may not lock into place. If that happens, first unlock the DIN Track mounting hook, mount the NX Unit to the DIN Track, then lock the DIN Track mounting hook.
- Refer to the hardware user's manual for the CPU Unit to which NX Units can be connected for information on how to mount the CPU Unit and how to mount NX Units to the CPU Unit.
- Refer to the user's manual for the Communications Coupler Unit for information on how to mount the Communications Coupler Unit and how to mount the NX Unit to the Communications Coupler Unit.

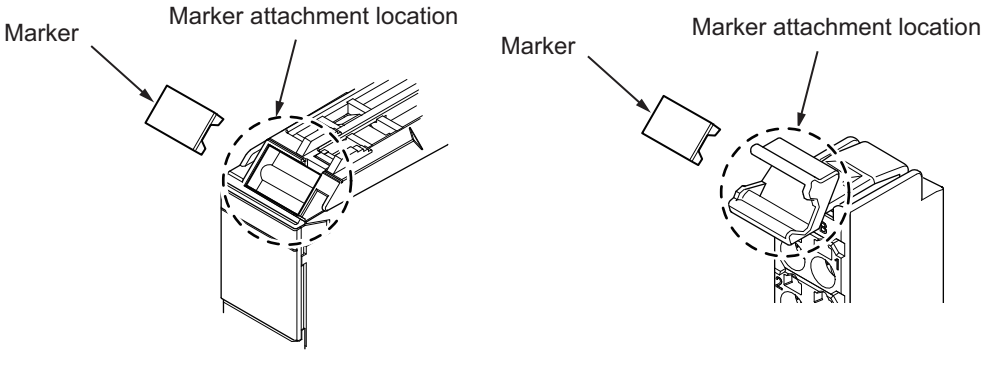
## 3-1-2 Attaching Markers

You can attach markers to the NX Units and to the terminal blocks to identify them.
The plastic markers made by OMRON are installed for the factory setting. The ID information can be written on them.
Commercially available markers can also be installed.
Replace the markers made by OMRON if you use commercially available markers now.




### ● Installation Method

Insert the protrusions on the markers into the marker attachment locations on the NX Units and terminal blocks on NX Units.




<Unit> <Terminal block>

### ● Commercially Available Markers

Commercially available markers are made of plastic and can be printed on with a special printer. To use commercially available markers, purchase the following products.

| Product name | Model number | |
|---|---|---|
| | Manufactured by Phoenix Contact | Manufactured by Weidmuller |
| Markers | UC1-TMF8 | DEK 5/8 |
| Special marker printer | UM EN BLUEMARK X1 | PrintJet PRO |

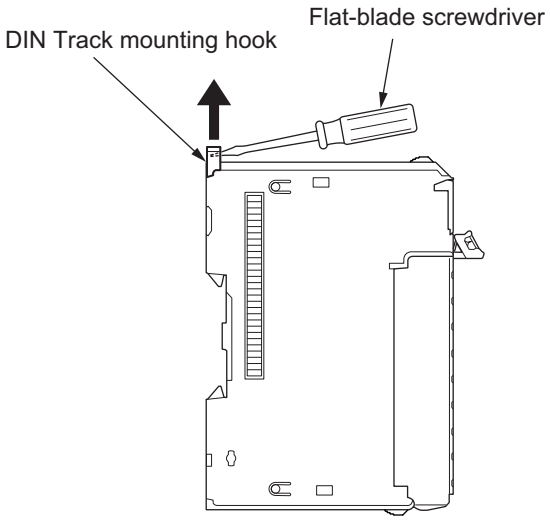
The markers made by OMRON cannot be printed on with commercially available special printers.

## 3-1-3 Removing Units

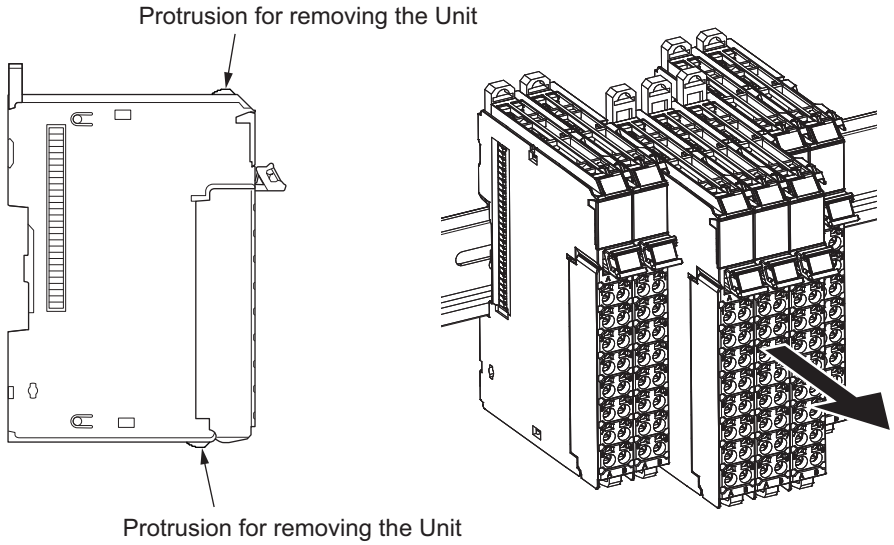
**Precautions for Safe Use**

Always turn OFF the Unit power supply and I/O power supply before you remove the NX Unit.

***1*** Use a flat-blade screwdriver or similar tool to pull up the DIN Track mounting hook on the NX Unit to remove.




***2*** As shown in the following figure, place your fingers on the protrusions on more than one NX Unit, including the NX Unit to remove, and pull the NX Units straight forward.



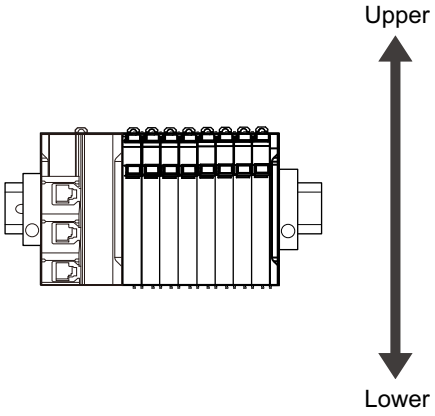

**Precautions for Correct Use**

- To remove an NX Unit, remove multiple NX Units together including the one you need to remove. If you attempt to remove only one NX Unit, it may be tight and difficult to pull out.
- Do not unlock the DIN Track mounting hooks on all of the NX Units at the same time. If you release the DIN Track mounting hooks on all of the NX Units at the same time, all of the NX Units will come off.

## 3-1-4 Installation Orientation

### Installation Orientation in the Case of a CPU Unit

Orientation is possible only in the upright installation orientation.

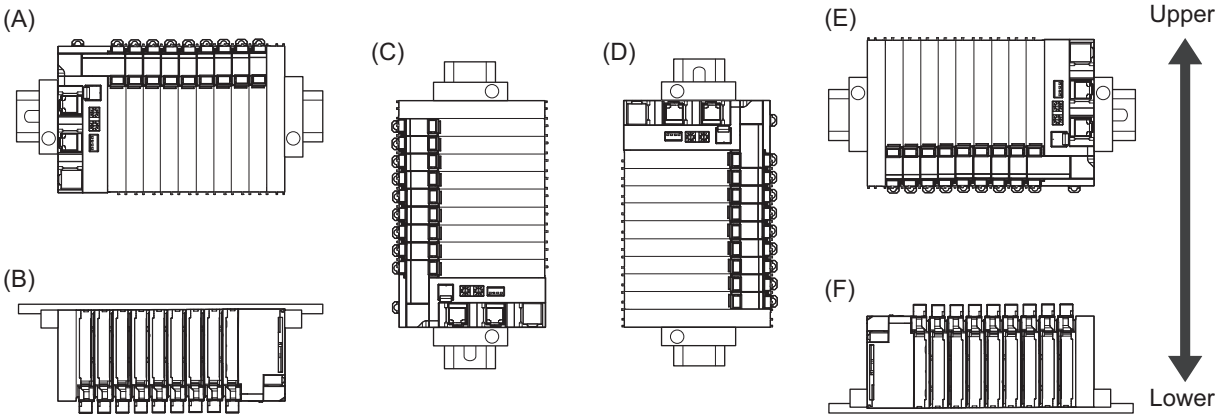



However, there are restrictions on the specifications depending on the NX Units to be used.
Refer to the user's manuals for the NX Units and System Units that you will use for details on restrictions.

### Installation Orientation on Slave Terminals

The Slave Terminal can be installed in any of the following six orientations.

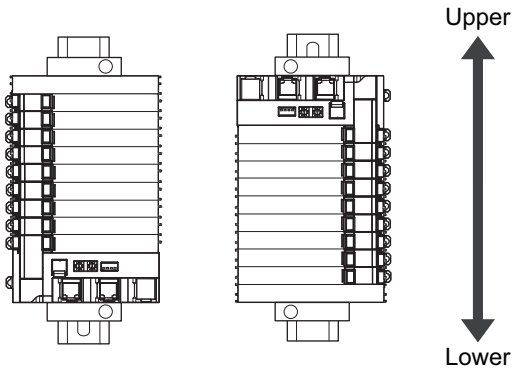
(A) is the upright installation orientation and (B) to (F) are installation orientations other than upright.




However, there are restrictions on the installation orientation and restrictions to the specifications that can result from the Communications Coupler Units and NX Units that are used.
For detailed restrictions, refer to descriptions of installation orientations and restrictions for each model in the user's manuals for the Communications Coupler Units, NX Units, and NX-series System Units that you will use.

**Precautions for Safe Use**

- For the installation orientations in the following figure, support the cables, e.g., with a duct, so that the End Plate on the bottom is not subjected to the weight of the cables. The weight of the cables may cause the bottom End Plate to slide downward so that the Slave Terminal is no longer secured to the DIN Track, which may result in malfunctions.



# 3-2 Power Supply Types and Wiring

There are the following two types of power supplies that supply power to the NX Units.

| Power supply type | Description |
|---|---|
| Unit power supply | This power supply is used for operating the NX Units. |
| I/O power supply | This power supply is used for driving the I/O circuits of the NX Units and for the connected external devices. |

The method for supplying power to the NX Units and the wiring method depend on the specifications for the CPU Unit to which NX Units are connected or the specifications for the Slave Terminal. Refer to *Designing the Power Supply System* or *Wiring*, which are described both in the hardware user's manual for the CPU Unit to which NX Units are connected and user's manual for the Communications Coupler Unit, for details on the method for supplying power to the NX Units and the wiring method.

The subsequent sections describe the applications of I/O power supply for the Safety Control Unit and supply methods, and how to calculate the total current consumption from the I/O power supply.

## 3-2-1 Applications of I/O Power Supply and Supply Methods

The applications of I/O power supply and supply methods for the Safety Control Unit are given as follows. Note that the Safety Control Unit does not use I/O power supply.

### Applications of I/O Power Supply

The I/O power supply is used for the following applications.

- Driving the I/O circuits of the Safety Input Unit or Safety Output Unit
- Input current to the Safety Input Unit
- Load current of external loads of Safety Output Unit
- Power for connected external devices

### I/O Power Supply Methods

I/O power is supplied to a Safety Control Unit from the *NX bus*.

This power is supplied through the NX bus connectors by connecting an I/O power supply to the I/O power supply terminals on the Communications Coupler Unit or Additional I/O Power Supply Unit.

Refer to *Designing the Power Supply System* and *Wiring* in the hardware user's manual for the connected CPU Unit for details on Units to which I/O power is supplied in a CPU Rack.
Refer to *Designing the Power Supply System* and *Wiring* in the user's manual for the connected Communications Coupler Unit for details on Units to which I/O power is supplied in a Slave Terminal.



**Additional Information**

**Power Supply-related Units for the NX-series**
The following three NX-series are related to power supply.
- Additional NX Unit Power Supply Unit
- Additional I/O Power Supply Unit
- I/O Power Supply Connection Units

Refer to the *NX-series System Unit User's Manual (Cat. No. W523)* for the specifications of these Units.
For a complete list of the latest power supply Units in the NX-series, refer to the *product catalog* or OMRON websites, or contact your OMRON representatives.

## 3-2-2 Calculating the Total Current Consumption from I/O Power Supply

The total current consumption of I/O power supplied from the NX bus must be within the range of the maximum I/O power supply current of the Communications Coupler Unit or the Additional I/O Power Supply Unit. Furthermore, for the CPU Rack, this must be less than the maximum I/O power supply current specified for the CPU Unit. Refer to the hardware user's manual for the CPU Unit to which the NX Units are connected for details on restrictions for CPU Racks. To confirm this and to calculate the I/O power supply capacity, calculate the total current consumption of the I/O power supply from the NX bus.
To confirm this and to calculate the I/O power supply capacity, calculate the total current consumption of the I/O power supply from the NX bus.

The total I/O current consumption from the NX bus is the sum of the following: the current consumption from the I/O power supply for the NX Units that receive power from the I/O power supply from the NX bus, the current consumption of those I/O circuits, and the current consumption of connected external devices.
The current consumption of external connection loads and connected external devices is not included in the *current consumption from I/O power supply* that is given for each model in the datasheets.

The total current consumption from I/O power supply of the Safety Control Unit is calculated as follows:
- **Total Current Consumption from I/O Power Supply of Safety Input Unit**
  = (Current consumption from I/O power supply of Safety Input Unit) + (Input current of Safety Input Unit × Number of inputs used) + (Total current consumption of connected external devices)
- **Total Current Consumption from I/O Power Supply of Safety Output Unit**
  = (Current consumption from I/O power supply of Safety Output Unit) + (Total connected load current) + (Total current consumption of connected external devices)

Refer to *Section 2 Specifications* on page 2-1 for the current consumption from I/O power supply of each model of Safety I/O Unit and the input current for each model of Safety Input Unit.

# 3-3 Wiring the Terminals

This section provides information on wiring the terminals on Safety I/O Units.

**⚠ WARNING**

Make sure that the voltages and currents that are input to the Units and slaves are within the specified ranges.
Inputting voltages or currents that are outside of the specified ranges may cause accidents or fire.

## 3-3-1 Wiring to the Screwless Clamping Terminal Blocks

This section describes how to connect wires to the screwless clamping terminal blocks, the installation and removing methods, and functions for preventing incorrect attachment.

You can connect ferrules that are attached to the twisted wires to the screwless clamping terminal block. You can also connect the twisted wires or the solid wires to the screwless clamping terminal block. If you connect the ferrules, all you need to do to connect the wires is to insert the ferrules into the terminal holes.

### Wiring Terminals

The terminals to be wired are as follows.

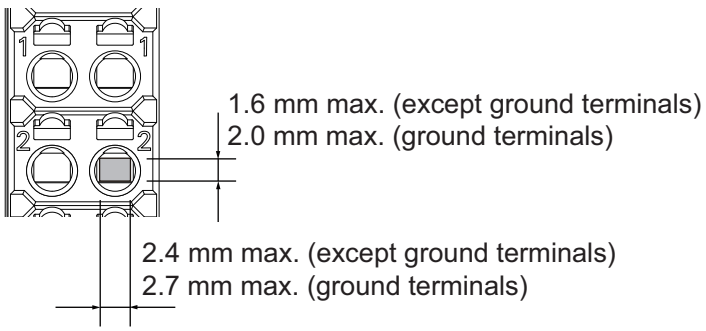
- I/O power supply terminals
- I/O terminals

### Applicable Wires

You can connect twisted wires, solid wires, or ferrules attached to twisted wires to the screwless clamping terminal block. The applicable wire dimensions and preparation methods are given below.

#### ● Dimensions of Wires Connected to the Terminal Block

The dimensions of wires that you can connect into the terminal holes of the screwless clamping terminal block are as in the figure below.
Process the applicable wires that are specified in the following description to apply the dimensions.



## Using Ferrules

If you use ferrules, attach the twisted wires to them.
Observe the application instructions for your ferrules for the wire stripping length when attaching ferrules.

**Precautions for Correct Use**

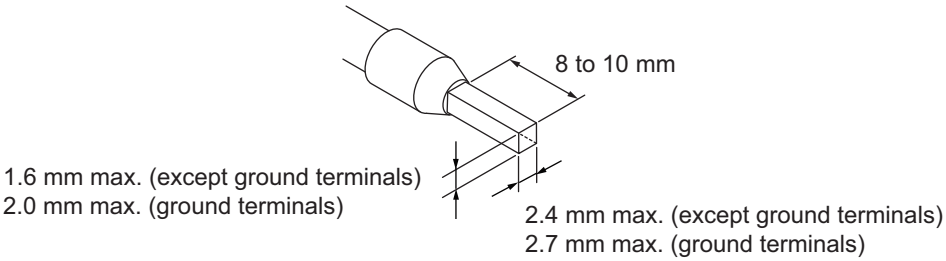
Always use plated one-pin ferrules. Do not use unplated ferrules or two-pin ferrules.

The applicable ferrules, wires, and crimping tools are given in the following table.

| Terminal type | Manufacturer | Ferrule model | Applicable wire ($mm^2$ (AWG)) | Crimping tool |
|---|---|---|---|---|
| All terminals except ground terminals | Phoenix Contact | AI0,34-8 | 0.34 (#22) | Phoenix Contact (Applicable wire sizes are given in parentheses.)<br>CRIMPFOX 6 (0.25 to 6 $mm^2$, AWG24 to 10) |
| | | AI0,5-8 | 0.5 (#20) | |
| | | AI0,5-10 | | |
| | | AI0,75-8 | 0.75 (#18) | |
| | | AI0,75-10 | | |
| | | AI1,0-8 | 1.0 (#18) | |
| | | AI1,0-10 | | |
| | | AI1,5-8 | 1.5 (#16) | |
| | | AI1,5-10 | | |
| Ground terminals | | AI2,5-10 | 2.0*1 | |
| All terminals except ground terminals | Weidmuller | H0.14/12 | 0.14 (#26) | Weidmuller (Applicable wire sizes are given in parentheses.)<br>PZ6 Roto (0.14 to 6 $mm^2$, AWG26 to 10) |
| | | H0.25/12 | 0.25 (#24) | |
| | | H0.34/12 | 0.34 (#22) | |
| | | H0.5/14 | 0.5 (#20) | |
| | | H0.5/16 | | |
| | | H0.75/14 | 0.75 (#18) | |
| | | H0.75/16 | | |
| | | H1.0/14 | 1.0 (#18) | |
| | | H1.0/16 | | |
| | | H1.5/14 | 1.5 (#16) | |
| | | H1.5/16 | | |

*1. For the ferrule type AI2,5-10, use wires with a diameter of 2.0 to 2.1 $mm^2$. Wires with a diameter exceeding 2.1 $mm^2$ cannot be used with the screwless clamping terminal block.

When you use any ferrules other than those in the above table, crimp them to the twisted wires so that the following processed dimensions are achieved.



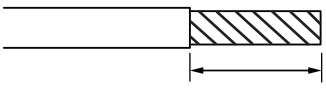

## Using Twisted or Solid Wires

If you use twisted wires or solid wires, use the following table to determine the correct wire specifications.

| Terminals | | Wire type | | | | Wire size | Conductor length (stripping length) |
|---|---|---|---|---|---|---|---|
| | | Twisted wires | | Solid wires | | | |
| Classification | Current capacity | Plated | Unplated | Plated | Unplated | | |
| All terminals except ground terminals | 2 A max. | Possible | Possible | Possible | Possible | 0.08 to 1.5 $mm^2$ (AWG28 to 16) | 8 to 10 mm |
| | Greater than 2 A and 4 A or less | | Not possible | Possible*1 | Not possible | | |
| | Greater than 4 A | Possible*1 | | Not possible | | | |
| Ground terminals | --- | Possible | Possible | Possible*2 | Possible*2 | 2.0 $mm^2$ | 9 to 10 mm |

*1. Secure wires to the screwless clamping terminal block. Refer to *Securing Wires* on page 3-13 for how to secure wires.

*2. With the NX-TB□□□1 Terminal Block, use twisted wires to connect the ground terminal. Do not use a solid wire.

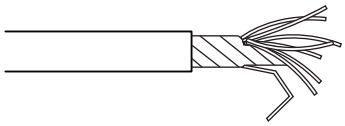


Conductor length (stripping length)

### Precautions for Correct Use

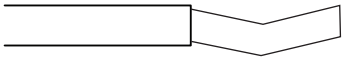
- Use cables with suitable wire sizes for the carrying current. There are also restrictions on the current due to the ambient temperature. Refer to the manuals for the cables and use the cables correctly for the operating environment.
- For twisted wires, strip the sheath and twist the conductor portion. Do not unravel or bend the conductor portion of twisted wires or solid wires.

NG



Unravel wires

NG



Bend wires

**Additional Information**

If more than 2 A will flow on the wires, use plated wires or use ferrules.

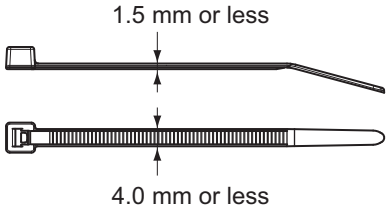
## Securing Wires

It is necessary to secure wires to the screwless clamping terminal block depending on the wire types that are used or the current flows on the wires.
The following table gives the necessity for securing wires.

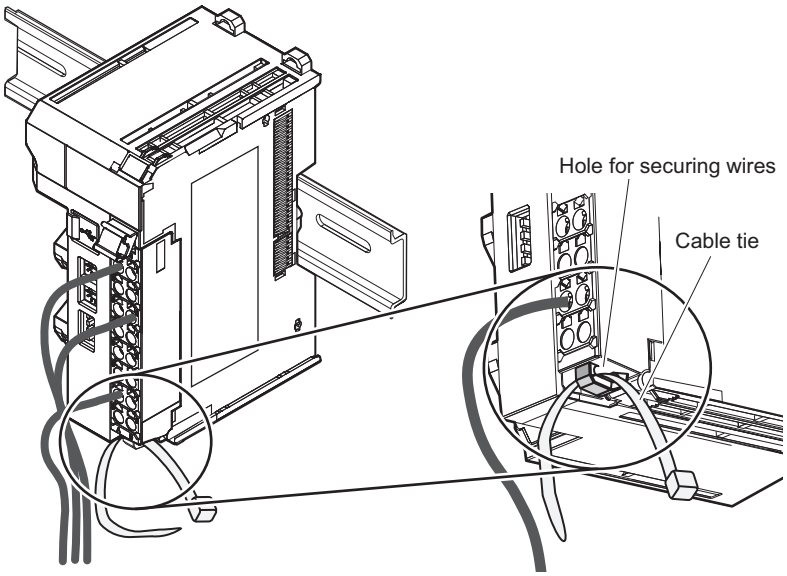
| Terminals | | Wire type | | | | |
|---|---|---|---|---|---|---|
| Classification | Current capacity | Ferrule | Twisted wires | | Solid wires | |
| | | | Plated | Unplated | Plated | Unplated |
| All terminals except ground terminals | 2 A max. | No | No | No | No | No |
| | Greater than 2 A and 4 A or less | | | Not possible | Yes | Not possible |
| | Greater than 4 A | | Yes | | Not possible | |
| Ground terminals | --- | | No | No | No | No |

Use the following procedure to secure the wires.

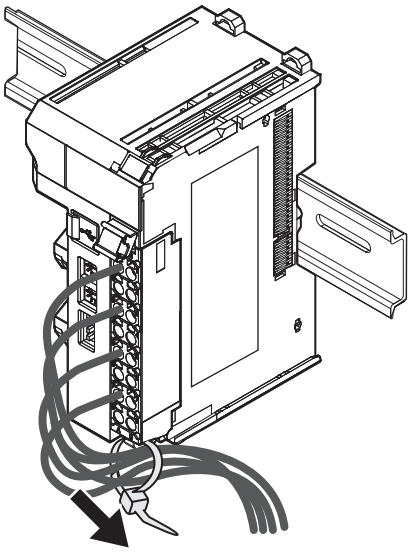
***1*** Prepare a cable tie.
A cable tie can be used with a width of 4 mm or less and a thickness of 1.5 mm or less. Select a cable tie correctly for the operating environment.



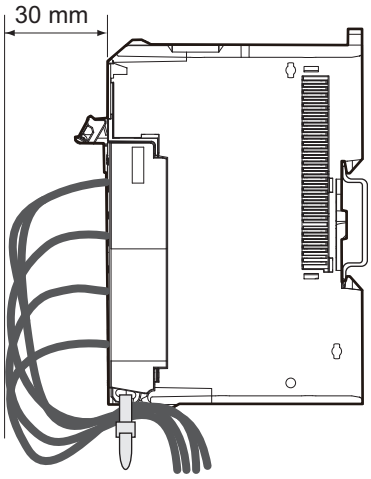

***2*** Pass a cable tie through the hole for securing wires on the bottom of the screwless clamping terminal block.




***3*** Bundle the wires with a cable tie and secure them to the screwless clamping terminal block.



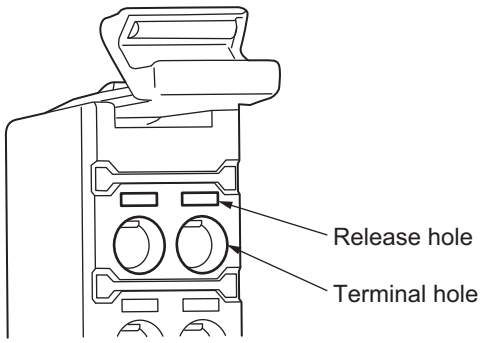
Secure wires within the range of 30 mm from the screwless clamping terminal block.



## Connecting and Removing Wires

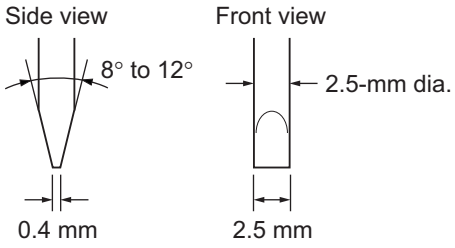
This section describes how to connect and remove wires.

### ● Terminal Block Parts and Names




### ● Required Tools

Use a flat-blade screwdriver to connect and remove wires.
Use the following flat-blade screwdriver.

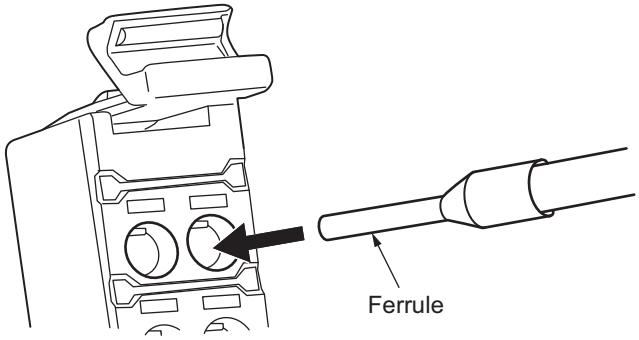



Recommended screwdriver

| Model | Manufacturer |
|---|---|
| SZF 0-0,4X2,5 | Phoenix Contact |

### ● Connecting Ferrules

Insert the ferrule straight into the terminal hole.
You do not need to insert a flat-blade screwdriver into the release hole.

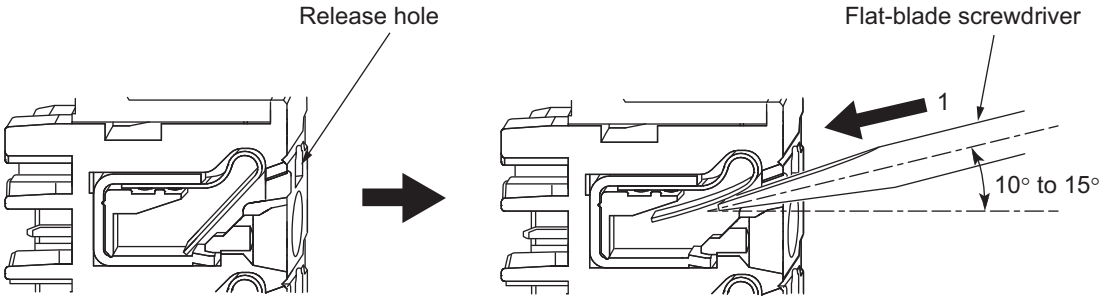


After you make a connection, make sure that the ferrule is securely connected to the terminal block.

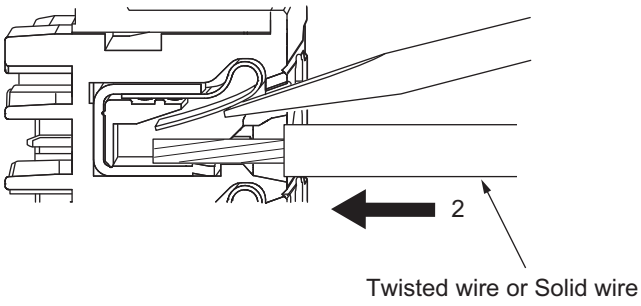
## ● Connecting Twisted Wires/Solid Wires

Use the following procedure to connect the twisted wires or solid wires to the terminal block.

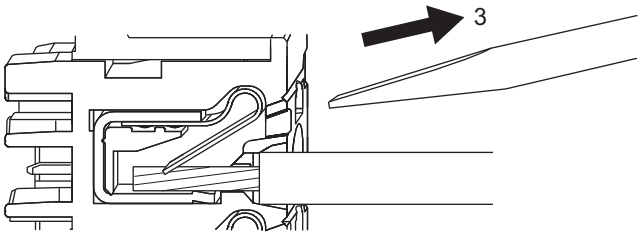
***1*** Press the flat-blade screwdriver diagonally into the release hole.
The optimal angle for insertion is between 10° to 15°. If you press in the screwdriver correctly, you will feel the spring in the release hole.




***2*** Leave the flat-blade screwdriver pressed into the release hole and insert the twisted wire or the solid wire into the terminal hole.
Insert the twisted wire or the solid wire until the stripped portion is no longer visible to prevent shorting.




***3*** Remove the flat-blade screwdriver from the release hole.

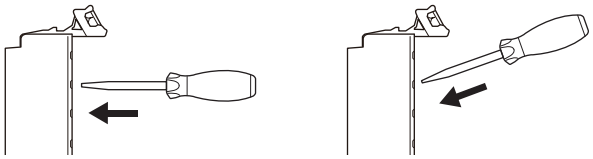



After you make a connection, lightly pull the twisted wire or the solid wire to make sure that the wire is securely connected to the terminal block.

**Precautions for Safe Use**

- Do not press the flat-blade screwdriver straight into the release hole. Doing so may break the terminal block.

**NG** **OK**



- When you insert a flat-blade screwdriver, press it down with a force of 30 N or less. Applying excessive force may damage the terminal block.
- Do not tilt or twist the flat-blade screwdriver while it is pressed into the hole. Doing so may break the terminal block.

**NG** **NG**

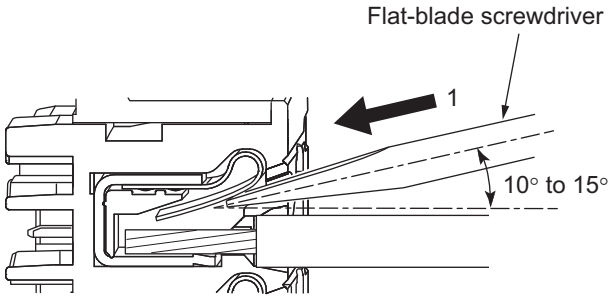


- Make sure that all wiring is correct.
- Do not bend the cable forcibly. Doing so may severe the cable.

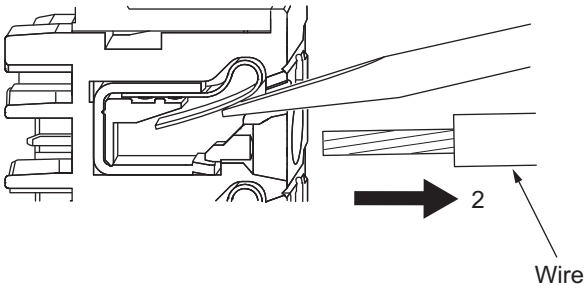
## ● Removing Wires

Use the following procedure to remove the wires from the terminal block.
The removal process is the same for both ferrules and twisted/solid wires.

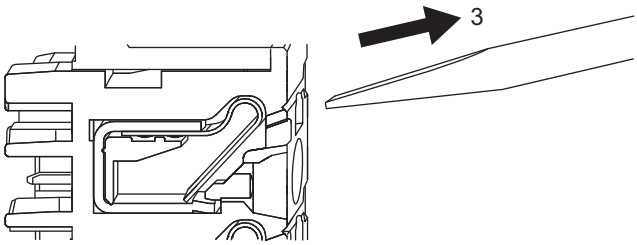
***1*** Press the flat-blade screwdriver diagonally into the release hole.
The optimal angle for insertion is between 10° to 15°. If you press in the screwdriver correctly, you will feel the spring in the release hole.




***2*** Leave the flat-blade screwdriver pressed into the release hole and pull out the wire.



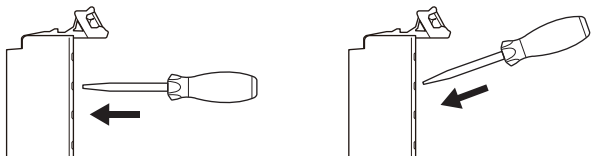

***3*** Remove the flat-blade screwdriver from the release hole.




**Precautions for Safe Use**

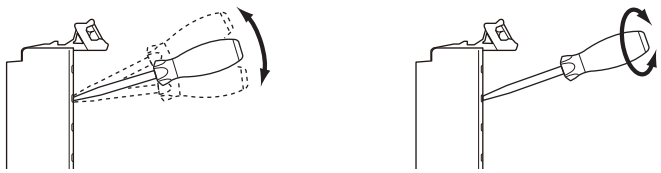
- Do not press the flat-blade screwdriver straight into the release hole. Doing so may break the terminal block.

**NG** **OK**



- When you insert a flat-blade screwdriver into a release hole, press it down with a force of 30 N or less. Applying excessive force may damage the terminal block.
- Do not tilt or twist the flat-blade screwdriver while it is pressed into the release hole. Doing so may break the terminal block.

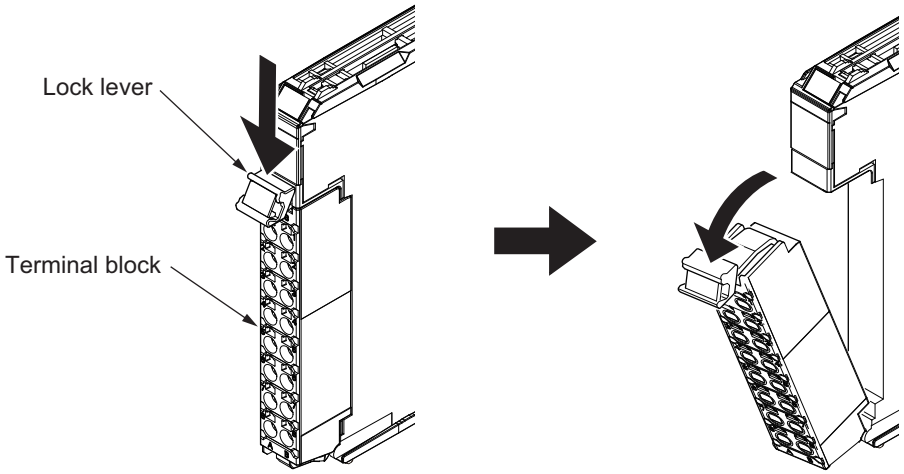
**NG** **NG**



- Make sure that all wiring is correct.
- Do not bend the cable forcibly. Doing so may severe the cable.

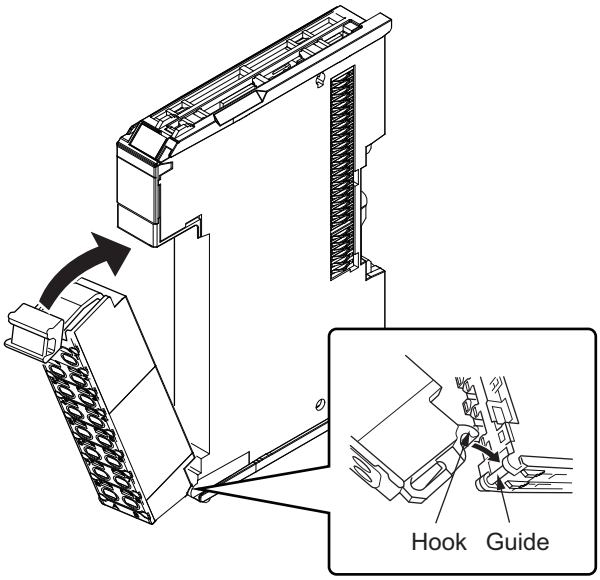
## Removing a Terminal Block

***1*** Press the lock lever on the terminal block and pull out the top of the terminal block to remove it.




## Attaching a Terminal Block

***1*** Mount the terminal block hook that is applicable to each Unit model on the guide at the bottom of the NX Unit, lift up the terminal block, and press in on the top of the terminal block until you hear it engage.
The terminal block will click into place on the Unit. After you mount the terminal block, make sure that it is locked to the Unit.

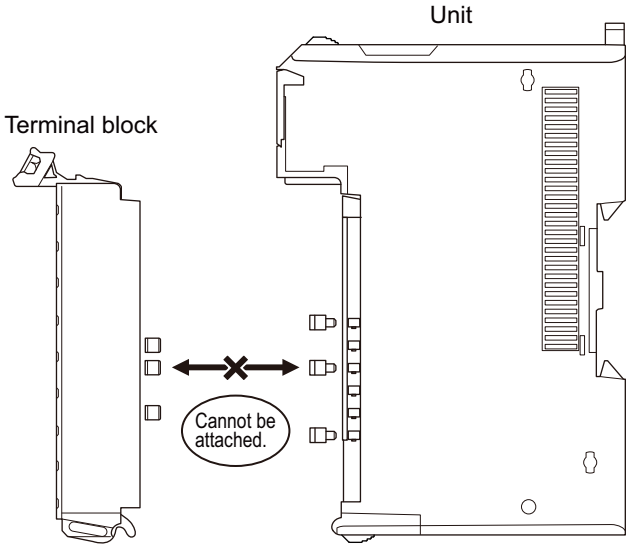



Mount a Terminal Block that is applicable to the model of each Unit.
Refer to Applicable Terminal Blocks for Each Unit Model (*Terminal Blocks* on page 2-24, *Terminal Blocks* on page 2-38) for the applicable Terminal Blocks.

## Preventing Incorrect Attachment of Terminal Blocks

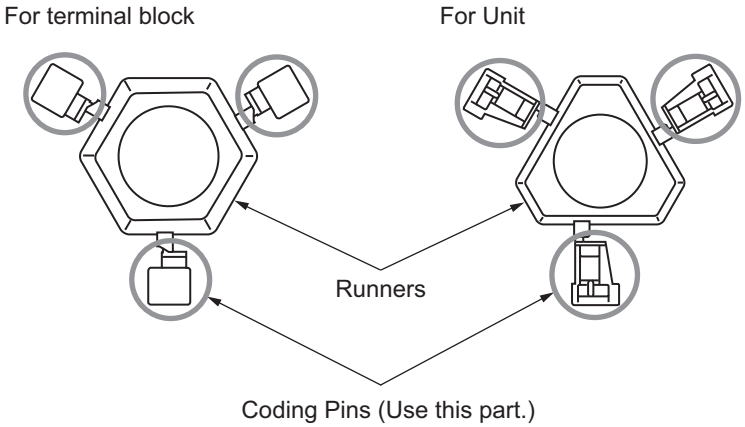
In order to prevent unintentionally installing the wrong terminal block, you can limit the combination of a Unit and a terminal block.

Insert three Coding Pins (NX-AUX02) into three of the six incorrect attachment prevention holes on the Unit and on the terminal block. Insert these pins into positions so that they do not interfere with each other when the Unit and terminal block are connected to each other.
You can use these pins to create a combination in which the wrong terminal block cannot be attached because the pin patterns do not match.




### ● Types of Coding Pins

There are two types of Coding Pins, both with their own unique shape: one for terminal blocks and one for Units.
Three pins come with each runner.



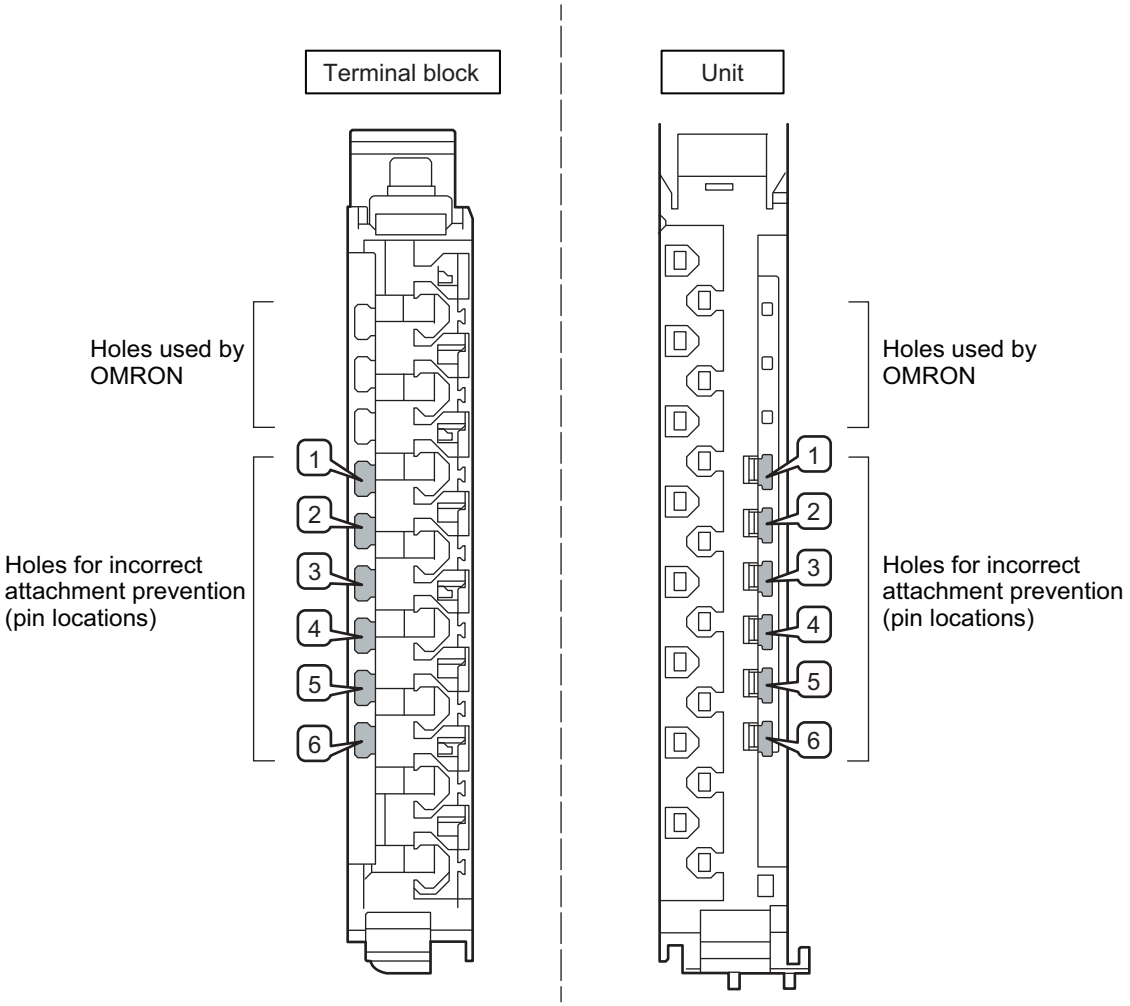

Use the following Coding Pins.

| Name | Model | Specification |
|---|---|---|
| Coding Pin | NX-AUX02 | For 10 Units (Terminal block: 30 pins, Unit: 30 pins) |

## ● Insertion Locations and Patterns of Coding Pins

Insert three Coding Pins each on the terminal block and on the Unit at the positions designated by the numbers 1 through 6 in the figure below.

As shown in the following table, there are 20 unique pin patterns that you can use.




○: Pin inserted

| Pattern | Pin locations for terminal block | | | | | | Pin locations for Unit | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | 1 | 2 | 3 | 4 | 5 | 6 | 1 | 2 | 3 | 4 | 5 | 6 |
| No.1 | ○ | ○ | ○ | | | | | | | ○ | ○ | ○ |
| No.2 | ○ | ○ | | ○ | | | | | ○ | | ○ | ○ |
| No.3 | ○ | ○ | | | ○ | | | | ○ | ○ | | ○ |
| No.4 | ○ | ○ | | | | ○ | | | ○ | ○ | ○ | |
| No.5 | ○ | | ○ | ○ | | | | ○ | | | ○ | ○ |
| No.6 | ○ | | ○ | | ○ | | | ○ | | ○ | | ○ |
| No.7 | ○ | | ○ | | | ○ | | ○ | | ○ | ○ | |
| No.8 | ○ | | | ○ | ○ | | | ○ | ○ | | | ○ |
| No.9 | ○ | | | ○ | | ○ | | ○ | ○ | | ○ | |
| No.10 | ○ | | | | ○ | ○ | | ○ | ○ | ○ | | |
| No.11 | | ○ | ○ | ○ | | | ○ | | | | ○ | ○ |
| No.12 | | ○ | ○ | | ○ | | ○ | | | ○ | | ○ |
| No.13 | | ○ | ○ | | | ○ | ○ | | | ○ | ○ | |
| No.14 | | ○ | | ○ | ○ | | ○ | | ○ | | | ○ |
| No.15 | | ○ | | ○ | | ○ | ○ | | ○ | | ○ | |
| No.16 | | ○ | | | ○ | ○ | ○ | | ○ | ○ | | |
| No.17 | | | ○ | ○ | ○ | | ○ | ○ | | | | ○ |
| No.18 | | | ○ | ○ | | ○ | ○ | ○ | | | ○ | |
| No.19 | | | ○ | | ○ | ○ | ○ | ○ | | ○ | | |
| No.20 | | | | ○ | ○ | ○ | ○ | ○ | ○ | | | |

Two sets of NX-AUX02 Pins are required to make the maximum of 20 pin patterns. (One set for 10 Units.)

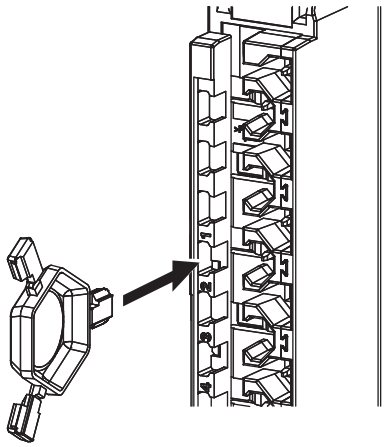
**Precautions for Correct Use**

- The holes not designated by the numbers 1 through 6 in the above figure are used by OMRON. If you insert any Coding Pins into the holes reserved for use by OMRON, you will not be able to mount the terminal block to the Unit.
- Do not use Coding Pins that have been attached and removed.

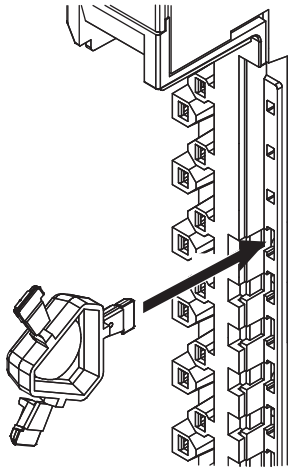
## ● Inserting the Coding Pins

***1*** Hold the pins by the runner and insert a pin into one of the incorrect attachment prevention holes on the terminal block or on the Unit.

Terminal block

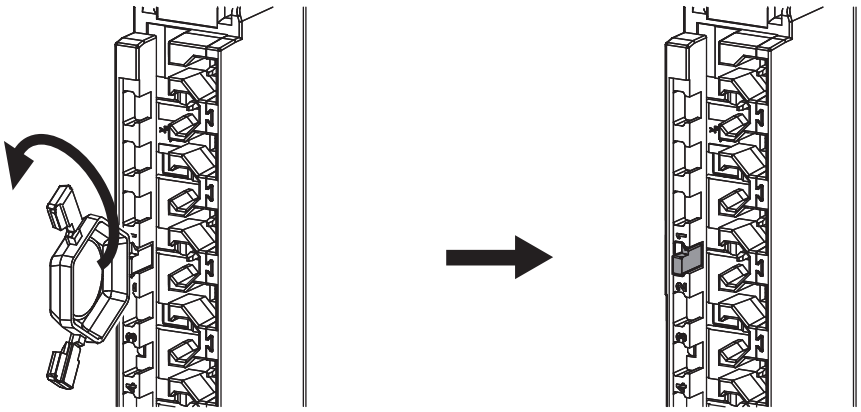


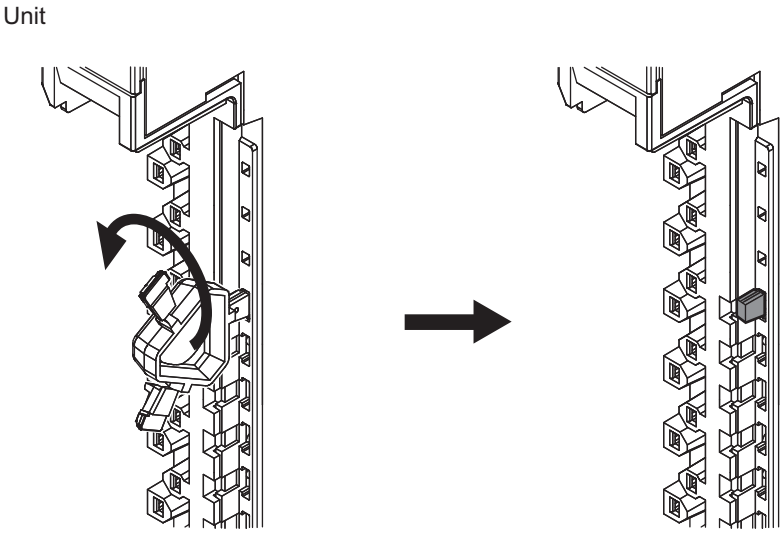
Unit



***2*** Rotate the runner to break off the Coding Pin.

Terminal block








## 3-3-2 Checking Wiring

For Input Units, you can turn ON and OFF an input from the external device that is connected to the Unit you need to check and monitor the results.
For Output Units, you can use forced refreshing to control the I/O outputs to the Unit you need to check to confirm the operation of the connected external device.
If you use the I/O Map, you can also monitor and perform forced refreshing even if the variables are not defined or the algorithms are not created. Therefore, you can easily check the wiring.
Refer to *Section 7 Checking Operation and Actual Operation* on page 7-1 for the monitoring and forced refreshing operations.

# 4

# Safety Control Unit Operation

This section provides information that is necessary to use the Safety Control Unit, including how the Safety Control Unit works.



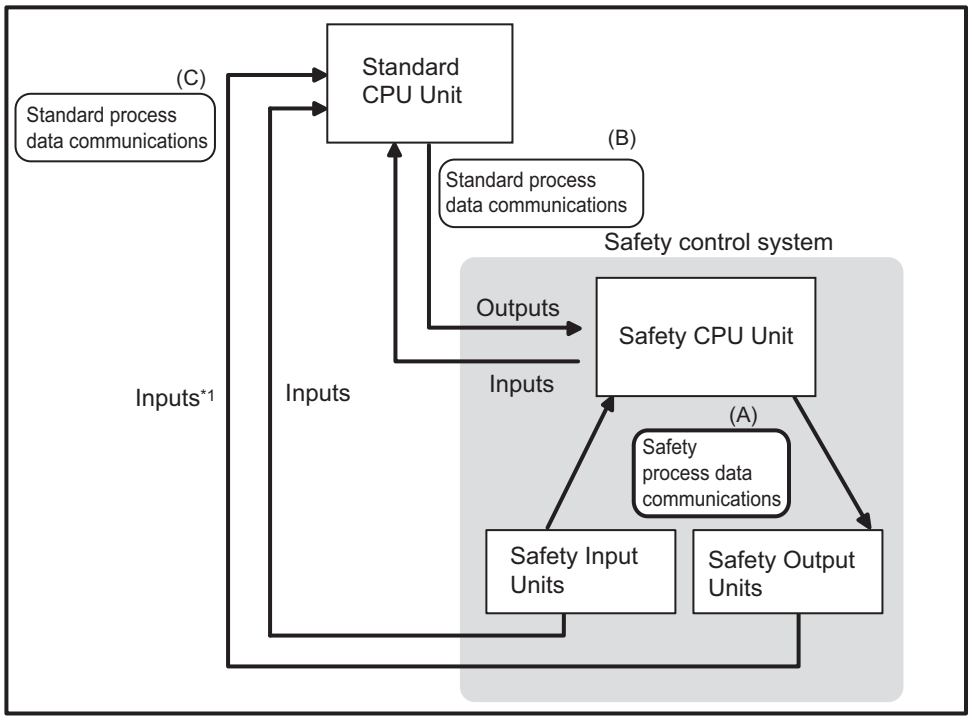
**4-1 Relationship between Units and Types of Communications ..... 4-2**
4-1-1 Safety Control System on CPU Rack or EtherCAT Network ..... 4-2
4-1-2 Safety Control System on EtherNet/IP Network ..... 4-4
4-1-3 Standalone Safety Control System ..... 4-6
**4-2 I/O System ..... 4-8**
4-2-1 Relationship between the Types of Signals and the Types of Communications ..... 4-8
4-2-2 Safety Data Types and Standard Data Types ..... 4-8
4-2-3 Specifying Safety Data Types and Standard Data Types ..... 4-9
**4-3 Safety I/O Function ..... 4-10**
4-3-1 Safety Input Function ..... 4-10
4-3-2 Safety Output Function ..... 4-32

# 4-1 Relationship between Units and Types of Communications

This section describes the relationships between Units and communications between Units in safety control systems that use CPU Units, EtherCAT Coupler Units and EtherNet/IP Coupler Units.

## 4-1-1 Safety Control System on CPU Rack or EtherCAT Network

The section describes the relationships of the Safety CPU Unit, Safety I/O Units, and a standard CPU Unit within the system, and provides details on communications between the Units based on the following figure.




*1. You cannot send outputs to the Safety Output Units. You can input the output values.

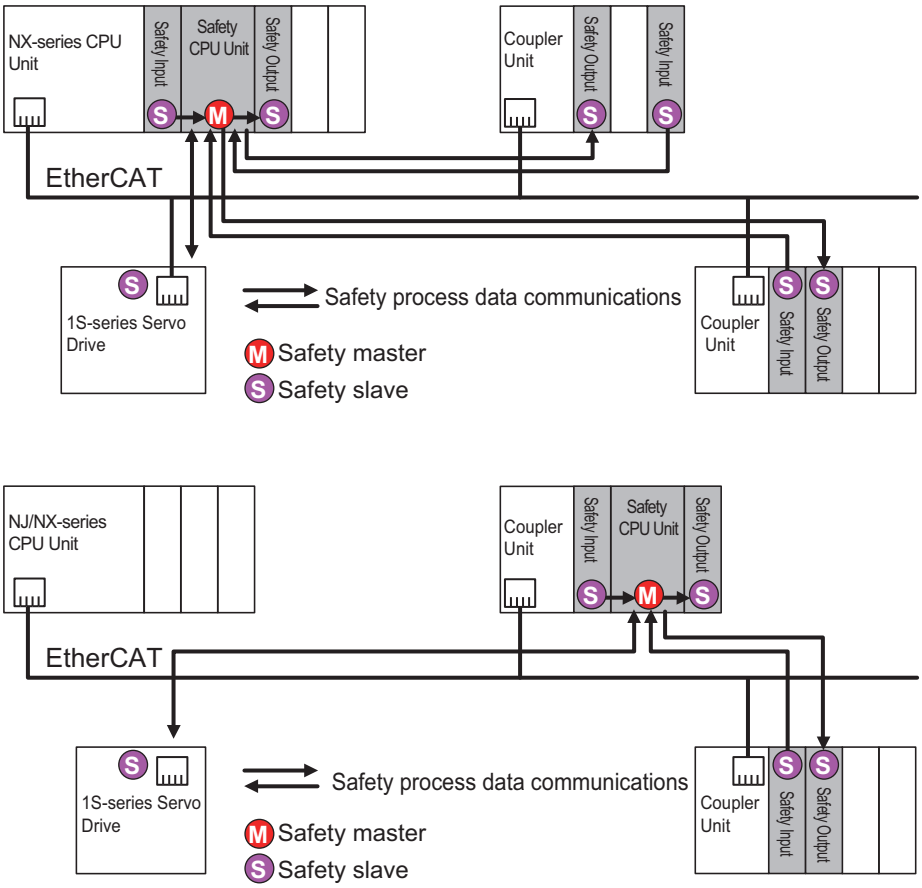
The safety control system consists of the Safety CPU Unit and Safety I/O Units.

The Safety CPU Unit performs safety controls with the Safety I/O Units through "safety process data communications". (Section (A) in the above figure.)
The Safety CPU Unit can perform standard I/O control with a standard CPU Unit through "standard process data communications". (Section (B) in the above figure.)

The standard CPU Unit can also perform "standard process data communications" with Safety I/O Units, but only for inputs. (Section (C) in the above figure.)

## Relationship between Master and Slaves during Safety Process Data Communications

The following figure shows the safety master/slave relationship during safety process data communications.




The Safety CPU Unit performs safety process data communications with the following devices using the FSoE protocol.

- Safety I/O Unit on the same CPU rack or the same Slave Terminal
- Safety I/O Unit on the Slave Terminal on the EtherCAT network
- EtherCAT slave with the FSoE slave function on the EtherCAT network (OMRON 1S-series Servo Drive)

In the example above, the Safety CPU Unit operates as a master of safety process data communications. The 1S-series Servo Drive or the Safety I/O Unit operates as a slave of safety process data communications.

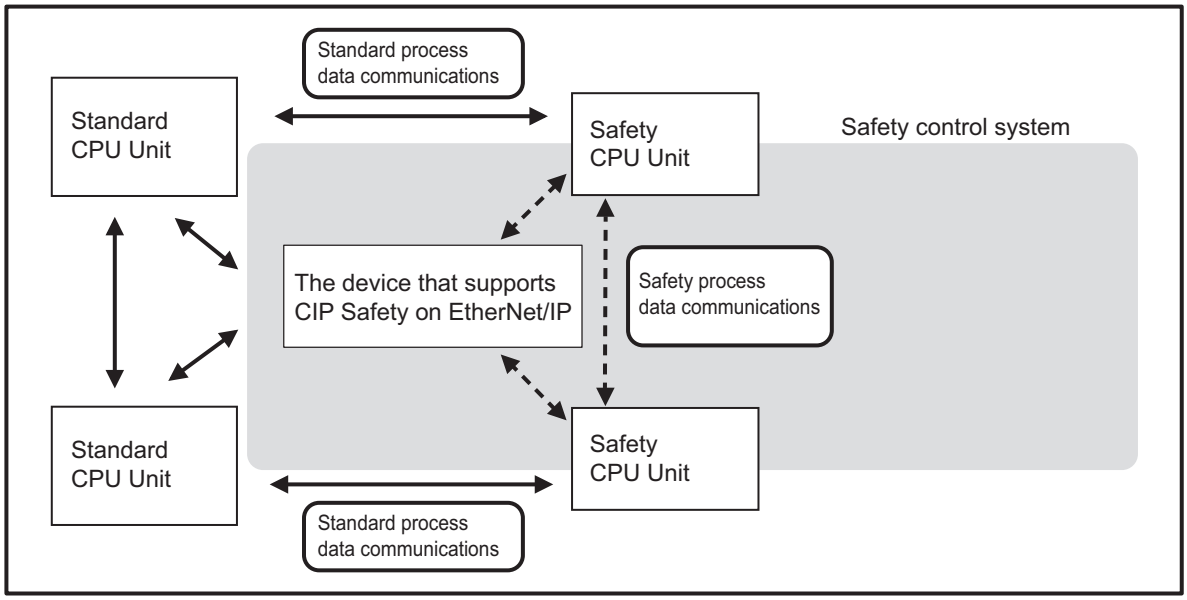
The NJ/NX-series CPU Unit and the Communications Coupler Unit relay safety process data communications between a safety master unit and a safety slave unit.

## Relationship between Master and Slaves during Standard Process Data Communications

The Safety Control Unit operates as a standard slave of a standard CPU Unit through the connected NX bus master. Safety process data communications with the standard CPU Unit are not available.

## 4-1-2 Safety Control System on EtherNet/IP Network

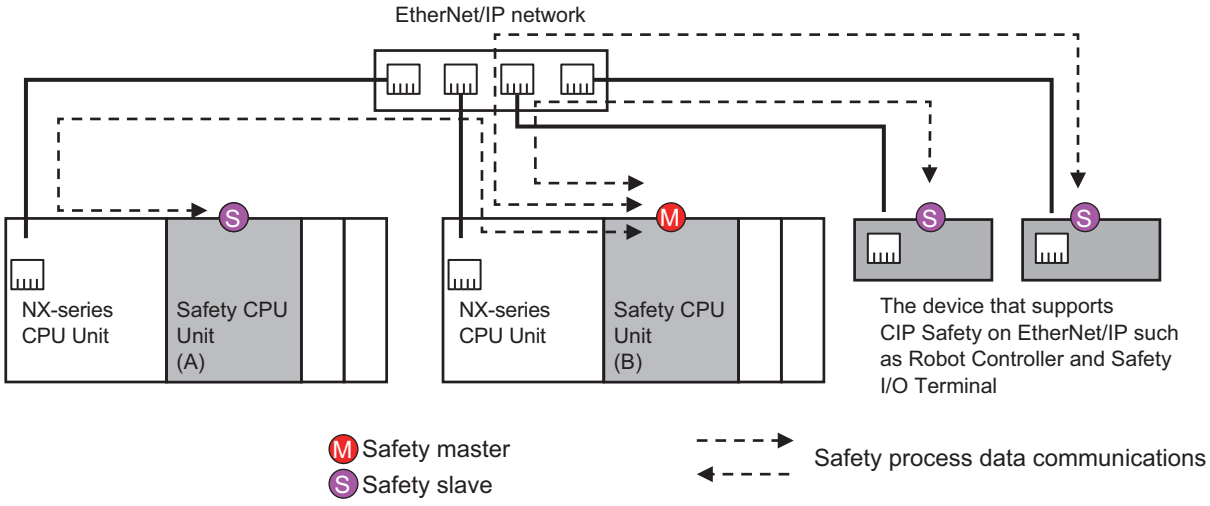
The section describes the relationships of the Safety CPU Unit, CIP Safety on EtherNet/IP device, standard CPU Units, and NX-series EtherNet/IP Units within the system, and provides details on communications between the Units based on the following figure.




The safety control system consists of a Safety CPU Unit and a CIP Safety on EtherNet/IP device. The Safety CPU Unit performs safety controls with another Safety CPU Unit and a CIP Safety on EtherNet/IP device through "safety process data communications". (Dashed arrows in the above figure.)
The Safety CPU Unit can perform standard I/O control with a standard CPU Unit and a NX-series EtherNet/IP Unit through "standard process data communications". (Solid arrows in the above figure.)

### Relationship between Master and Slaves during Safety Process Data Communications

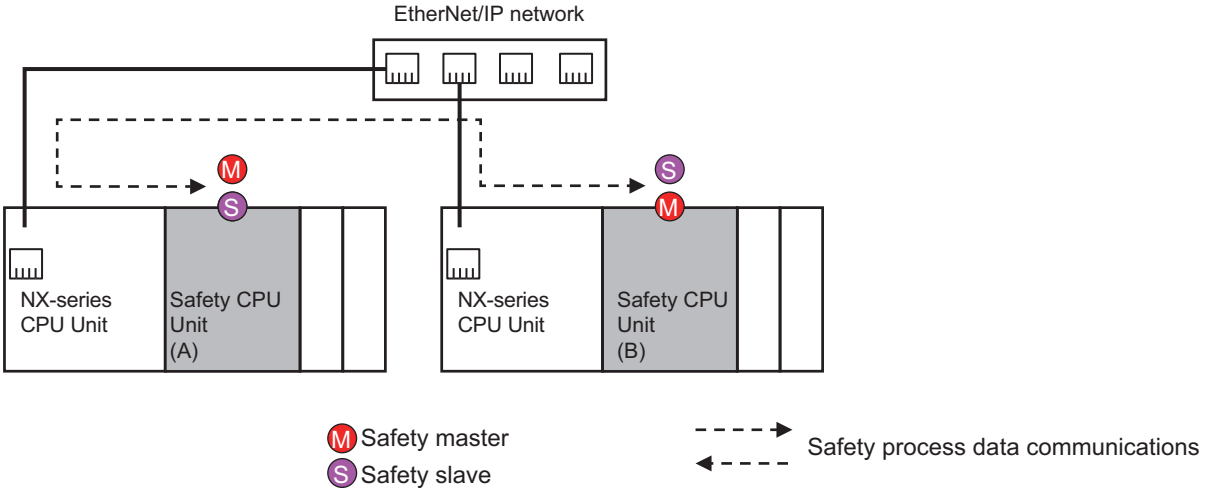
The relationships for safety process data communications are shown in the following figure.




The Safety CPU Unit performs safety process data communications with the following devices using the CIP Safety protocol.

- The Safety CPU Unit connected to another NX-series CPU Unit
- The NX-series EtherNet/IP Unit connected to another CPU Unit
- CIP Safety on EtherNet/IP-compatible devices

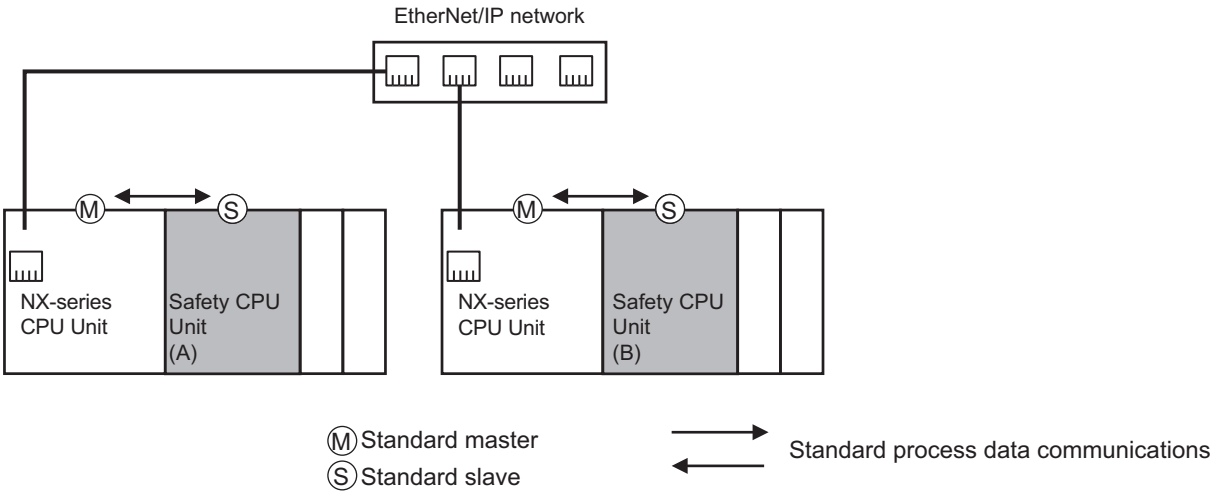
To explain the operation better, the originator is referred to as “master” and the target is referred to as “slave”in this section, although the originator/target model is adopted for CIP Safety.
In the example above, the Safety CPU Unit (B) operates as a master of safety process data communications. On the other hand, the Safety CPU Unit (A) and the CIP Safety on EtherNet/IP devices operate as slaves in the safety process data communications.
The NX-series CPU Units or NX-series EtherNet/IP Units relay safety process data communications between a safety master unit and a safety slave unit.




The Safety CPU Unit can also behave as a slave in the safety process data communications, while serving as a master of the safety process data communications. Using this feature, you can configure a system where the two Safety CPU Units can mutually monitor and control their operations as shown in the above figure.

## Relationship between Master and Slaves during Standard Process Data Communications

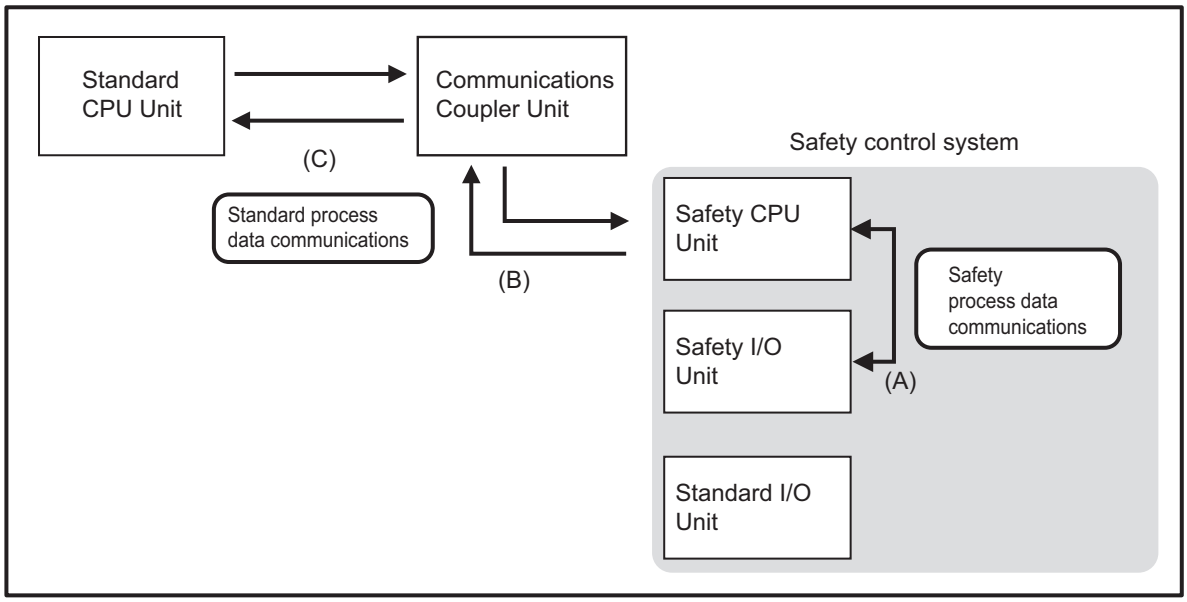
The relationships for standard process data communications are shown in the following figure.



The Safety CPU Unit serves as a standard slave of the connected CPU Unit or NX-series EtherNet/IP Unit. The Safety CPU Unit does not serve as a master of the standard process data communications.

## 4-1-3 Standalone Safety Control System

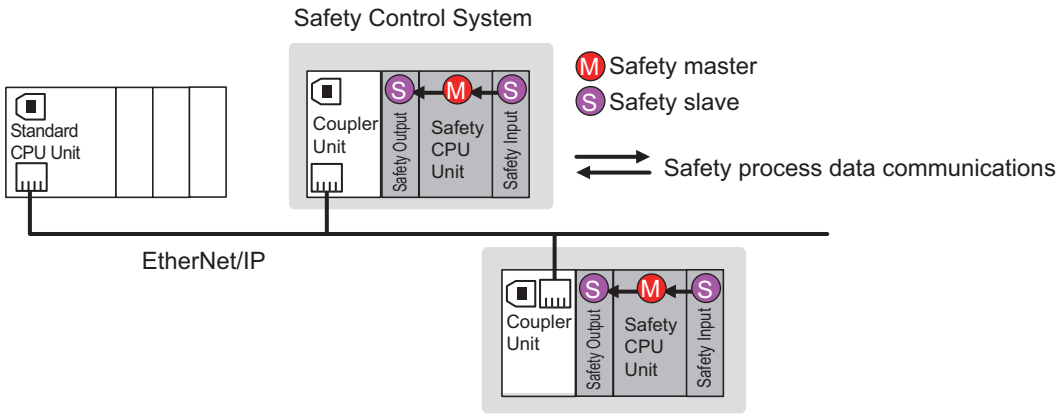
The section describes the relationships of the Safety CPU Unit, EtherNet/IP Coupler Unit, and a standard CPU Unit within the system, and provides details on communications between the Units based on the following figure.




The safety control system consists of the Safety CPU Unit, Safety I/O Units, and standard I/O Units.
The Safety CPU Unit performs safety controls with Safety I/O Units through "safety process data communications". (Section (A) in the above figure.)
The Safety CPU Unit, Safety I/O Units and standard I/O Units can perform standard I/O with the Communications Coupler Unit through "standard process data communications". (Section (B) in the above figure.)
The standard CPU Unit can indirectly monitor the operating status of the safety control system through "standard process data communications" when the Communications Coupler Unit is connected. (Section (C) in the above figure.)

### Relationship between Master and Slaves during Safety Process Data Communications

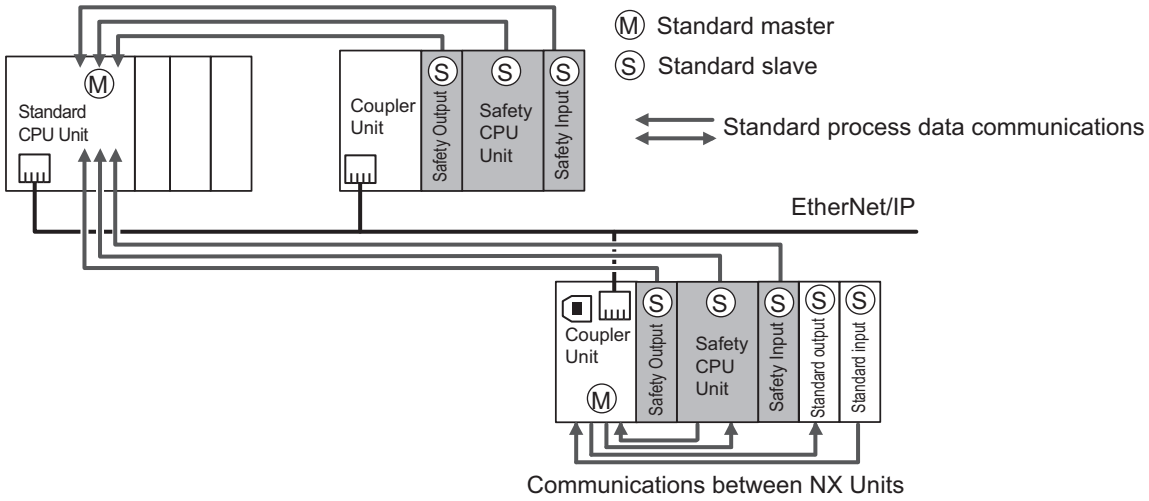
The relationships for safety process data communications are shown in the following figure.




The Safety CPU Unit performs safety process data communications with the Safety I/O Units on the same Slave Terminal.
Safety process data communications are not performed with Safety I/O Units on other Slave Terminals on the EtherNet/IP network.
Here, the Safety CPU Unit operates as the master in the safety process data communications. Meanwhile, the Safety I/O Units operate as slaves in the safety process data communications.
The Communications Coupler Unit relays safety process data communications between a safety master unit and a safety slave unit.

## Relationship between Master and Slaves during Standard Process Data Communications

The relationships for standard process data communications are shown in the following figure.




The standard controller and the NX-series EtherNet/IP Unit perform standard process data communications with the Safety CPU Unit and the Safety I/O Units on the EtherNet/IP network.
A standard controller and an NX-series EtherNet/IP Unit cannot output signals to Safety I/O Units.

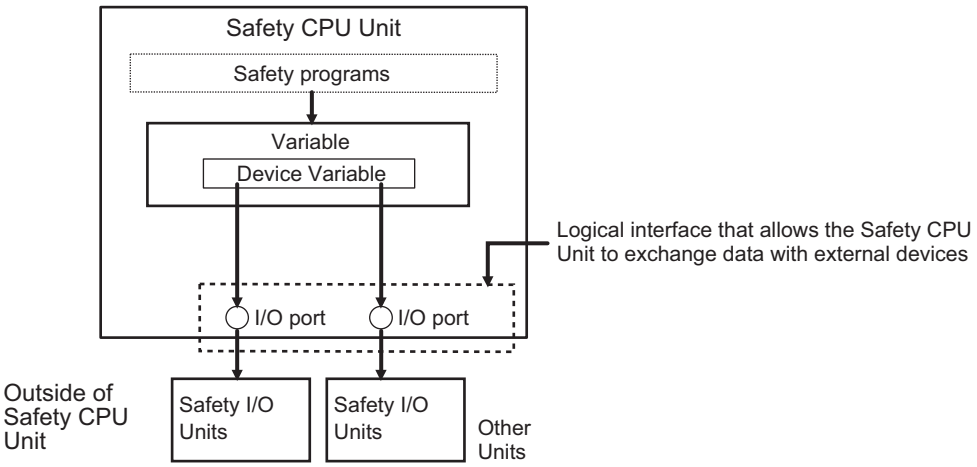
Also, the Safety CPU Unit performs standard process data communications with the Standard I/O Units on the same Slave Terminal.

# 4-2 I/O System

This section describes how the Safety CPU Unit processes I/O with external devices such as Safety I/O Units.

When the Safety CPU Unit exchanges signals with Safety I/O Units and other external devices, it does so through logical interfaces that are called "I/O ports".

I/O ports are created automatically when you create the control configuration for safety controls on the Sysmac Studio and set up the safety process data communications.
You assign device variables to I/O ports to gain access to the external devices from the safety programs.




You can check the **I/O ports** in the **I/O Map** of the Sysmac Studio.

## 4-2-1 Relationship between the Types of Signals and the Types of Communications

The safety control system uses the communications that are described below to process all I/O with safety inputs, safety outputs, standard inputs, standard outputs, standard controllers, and NX-series EtherNet/IP Units.

- The exchange of signals with safety inputs and safety outputs is done with safety process data communications.
- The exchange of standard signals with safety inputs, safety outputs, and standard controllers is done with standard process data communications.

## 4-2-2 Safety Data Types and Standard Data Types

In this safety control system, the data type of a variable determines whether a signal is related to the safety controls. Broadly speaking, there are the following two data types: safety data types and standard data types.

## Safety Data Type Variables

A safety data type variable is a variable that specifies data that is related to safety controls.
The names of safety data type variables have the word *SAFE* appended to a *standard data type name* such as *SAFEBOOL* and *SAFEBYTE*.
Refer to *6-2-5 Data Type* on page 6-14 for details on the safety data types.

## Standard Data Type Variables

These variables represent data that is not related to safety controls.
Refer to *5-7 Exposing Variables to Standard Controllers* on page 5-49 for details on how to access standard data type variables from the standard controllers.

Each type of signal is defined with a standard data type or safety data type as shown below.

| Type of signal | Data type of variable to use |
|---|---|
| Outputs from safety control to standard control | Standard data type |
| Inputs from standard control to safety control | |
| Safety inputs from external devices | Safety data type |
| Safety outputs to external devices | |
| Standard inputs from external devices | Standard data type |
| Standard outputs to external devices | |
| Internal safety-related signals with no I/O with any external devices | Safety data type |
| Internal standard-related signals with no I/O with any external devices | Standard data type |

# 4-2-3 Specifying Safety Data Types and Standard Data Types

The safety data type variables and standard data type variables are used depending on the type of communications.

- Signals that are input and output through safety process data communications must be defined as safety data type variables.
- Signals that are input and output through standard process data communications must be defined as standard data type variables.

# 4-3 Safety I/O Function

The following sections describe the safety I/O functions of the Safety I/O Units.

## 4-3-1 Safety Input Function

### Connectable Input Devices

The Safety Input Unit diagnoses the connected external devices and other things through the safety input terminals.
The safety input devices and standard input devices that can be connected to the safety input terminals of the Safety Input Unit are listed in the following table.

| Input device name | Type of input device | Type of contacts |
|---|---|---|
| Emergency stop switch | Device with mechanical contacts | Single channel<br>Dual-channel equivalent input |
| Safety door switch | Device with mechanical contacts | Single channel<br>Dual-channel equivalent input<br>Dual-channel complementary input |
| Safety limit switch | Device with mechanical contacts | Single channel<br>Dual-channel equivalent input<br>Dual-channel complementary input |
| Two-hand switches | Device with mechanical contacts | Dual-channel complementary input |
| Safety key selector switch | Device with mechanical contacts | Single channel<br>Dual-channel equivalent input<br>Dual-channel complementary input |
| Enabling switch | Device with mechanical contacts | Dual-channel equivalent input |
| EDM feedback | Device with mechanical contacts | Single channel |
| Reset switch | Device with mechanical contacts, standard input | Single Channel with Test Pulse<br>Single Channel without Test Pulse |
| Safety light curtain | Device with semiconductor output | Dual-channel equivalent input |
| Safety laser scanner | Device with semiconductor output | Dual-channel equivalent input |
| High-coded door switch | Device with semiconductor output | Dual-channel equivalent input |

You can set the above parameters for the following general-purpose input devices.

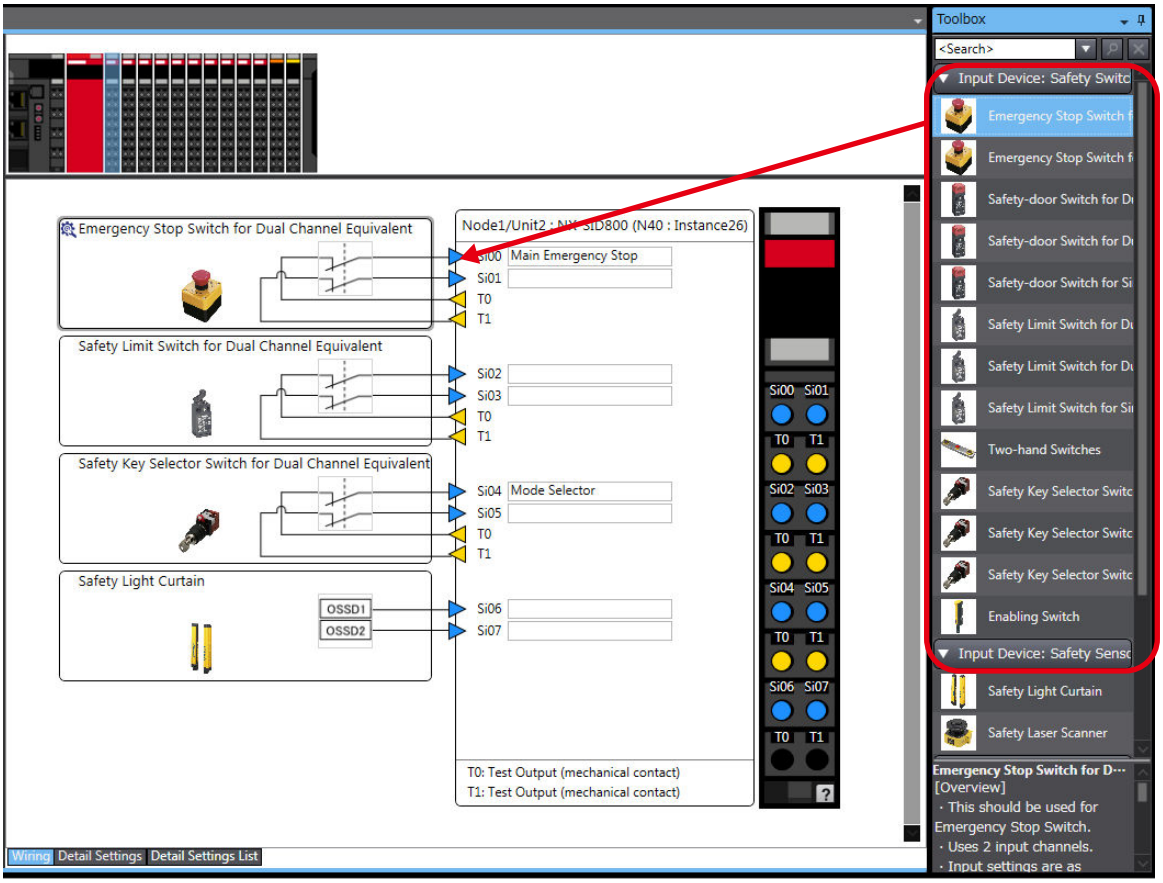
| Type | Possible settings |
|---|---|
| Safety input devices with mechanical contacts<br>• Mechanical Contact for Single Channel<br>• Mechanical Contact for Dual Channel Equivalent<br>• Mechanical Contact for Dual Channel Complementary | Emergency stop switch, safety door switch, safety limit switch, two-hand switches, safety key selector switch, Enabling switch, and EDM feedback |
| Safety input device with semiconductor output<br>• Semiconductor Output for Single Channel<br>• Semiconductor Output for Dual Channel Equivalent<br>• Semiconductor Output for Dual Channel Complementary | Safety light curtain, safety laser scanner, and high-coded door switch |

The following OMRON Special Safety Input Devices can be connected directly without a special controller. (This applies only to the NX-SIH400.)

| Type | Examples | |
|---|---|---|
| OMRON Single-beam Safety Sensors | E3ZS and E3FS | * Conforms to Type 2 and PLc. |
| OMRON Non-contact Door Switches | D40A-2 | * Conforms to PLe and Safety Category 4. |
| | D40Z | * Conforms to PLe and Safety Category 4. |
| | D40A | * Conforms to PLd and Safety Category 3. |
| OMRON Safety Mats | UM and UMA | * Conforms to PLd and Safety Category 3. |
| OMRON Safety Edges | SGE (4-wire connection) | * Conforms to PLd and Safety Category 3. |

## Setting Up Safety Functions

You can easily set the safety functions of the safety input terminals from the Sysmac Studio by selecting the types of external devices that are connected. Refer to the *4-3 Safety I/O Function* on page 4-10 for details.



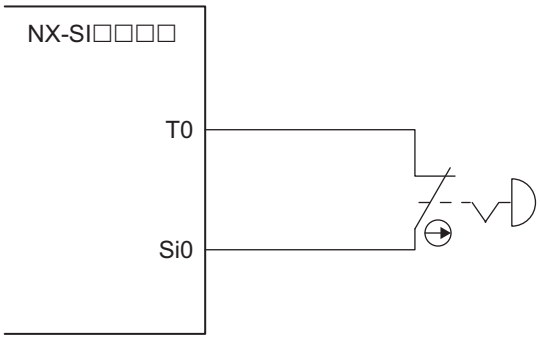

## Connecting Input Devices

This section describes the connection methods for input devices.

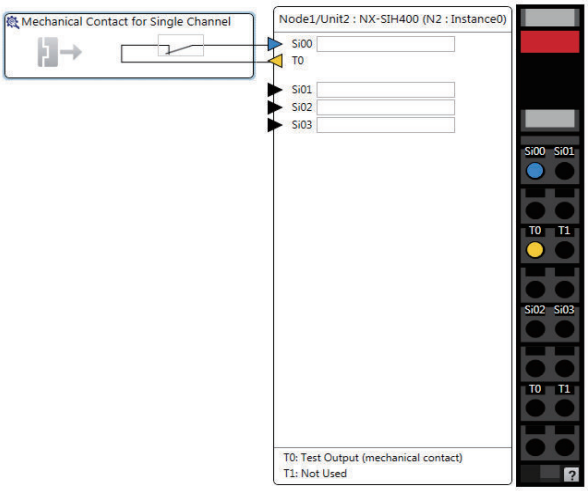
### ● Devices with Mechanical Contacts

A device with mechanical contacts, such as an emergency stop pushbutton or safety limit switch, is used with the safety input terminal (Si) and test output terminal (To).

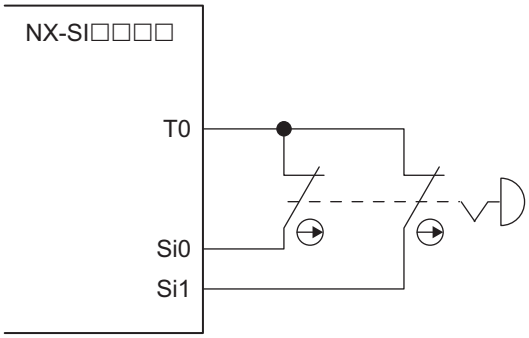
- Single-channel Input



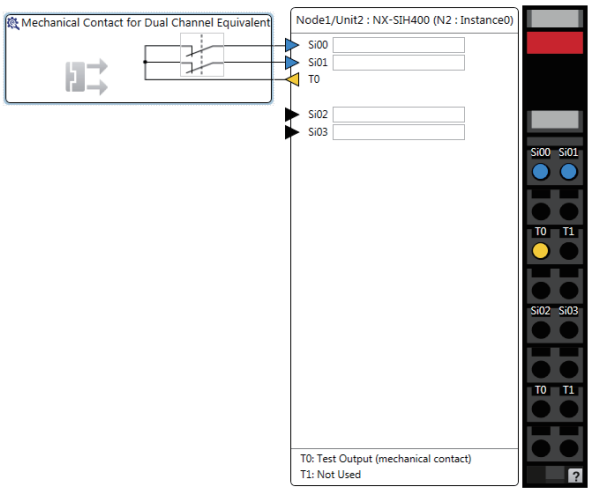

Example of Sysmac Studio Settings:



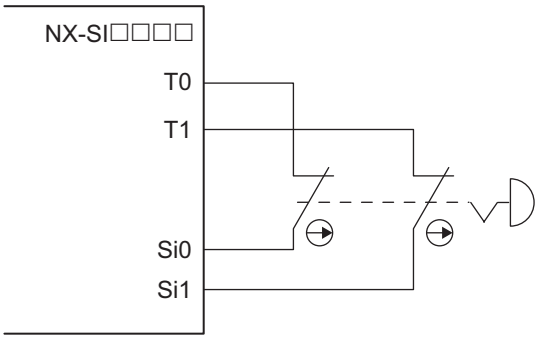

- Dual-channel Input When I/O Short Detection between Lines Is Not Required



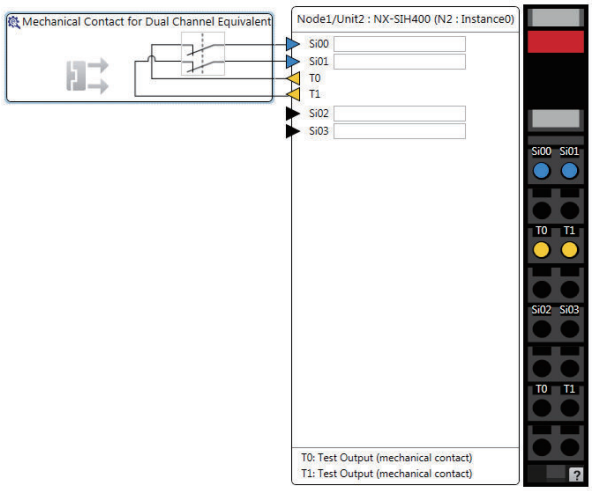

Example of Sysmac Studio Settings:




- Dual-channel Input When I/O Short Detection between Lines Is Required

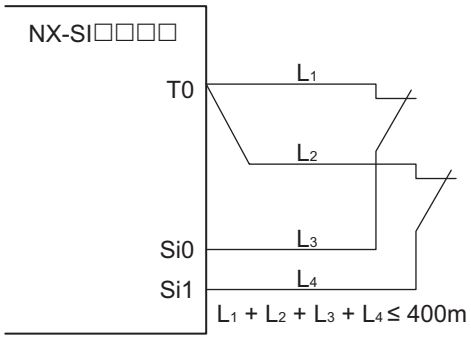



Example of Sysmac Studio Settings:



**Precautions for Correct Use**

- Configure dual-channel inputs with safety input terminals on the same Unit. It is not always possible to detect short circuits between safety input terminals on different Units.
- The total length of cable connected to one test output must be as follows:
  NX-SIH400 and NX-SID800: 400 m max.

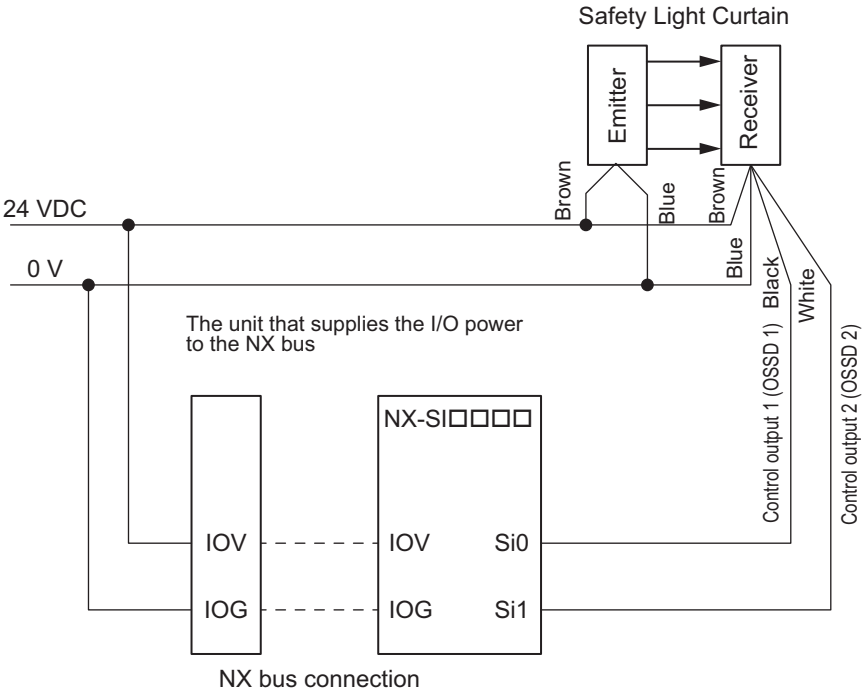



**Additional Information**

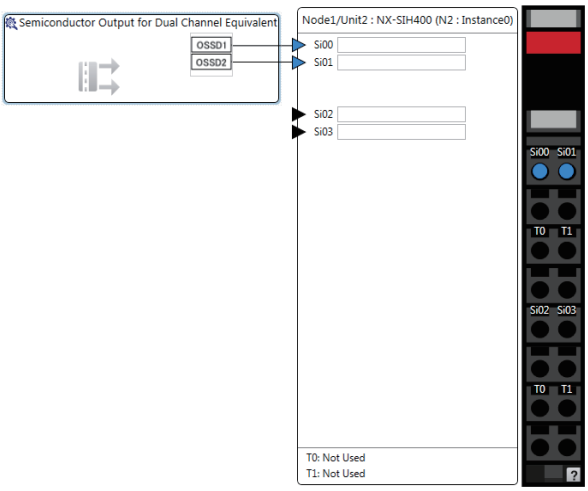
You can detect short-circuits between two input channels with the following methods:
- Dual-channel equivalent input: "With Test Pulse"
- Dual-channel complementary input: "Without Test Pulse" and "With Test Pulse"

## ● Devices with Semiconductor Outputs

The signal from a device with a semiconductor output, such as a light curtain, is input to a safety input terminal (Si).



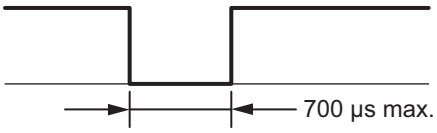

Example of Sysmac Studio Settings:



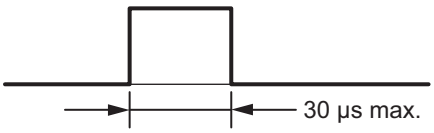

### Precautions for Correct Use

Safety devices with semiconductor outputs, such as safety light curtains, sometimes provide a pulse output that is used to detect wiring errors. Observe the following when connecting a Safety Device with a semiconductor output to a safety input terminal.

- OFF pulse width when semiconductor output is ON: 700 μs max.



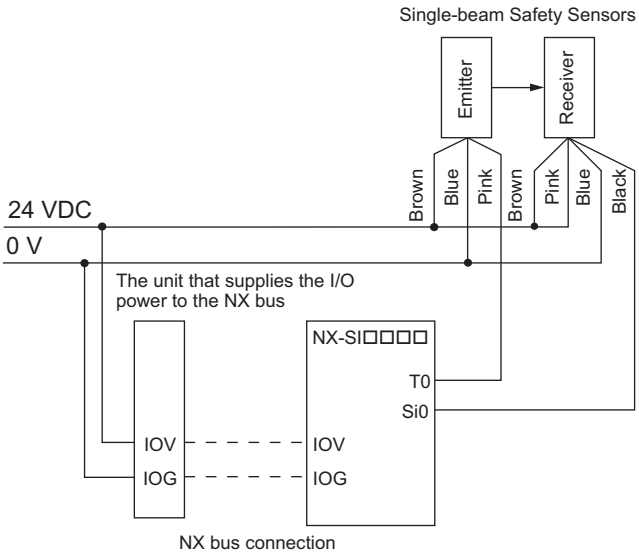

- ON pulse width when semiconductor output is OFF: 30 μs max.



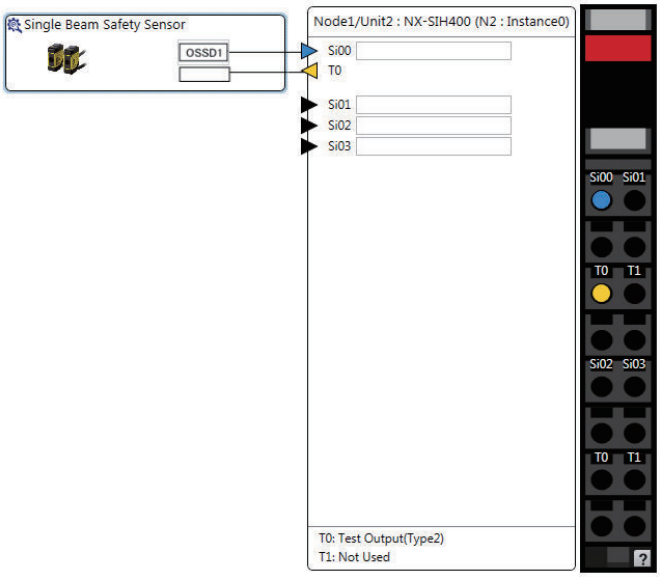

Check the specifications of the connected device for the maximum cable length.

## ● E3ZS/E3FS Single-beam Safety Sensors

An OMRON E3ZS/E3FS Single-beam Safety Sensor is connected as shown in the following figure.

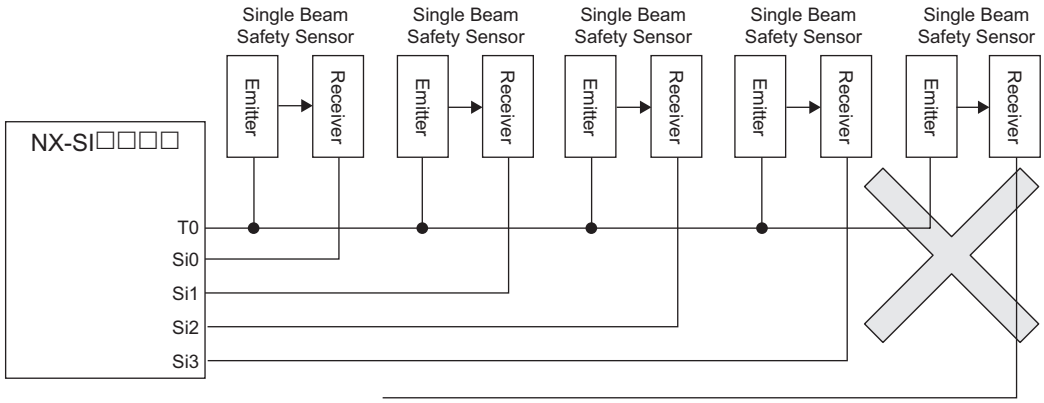


Example of Sysmac Studio Settings:

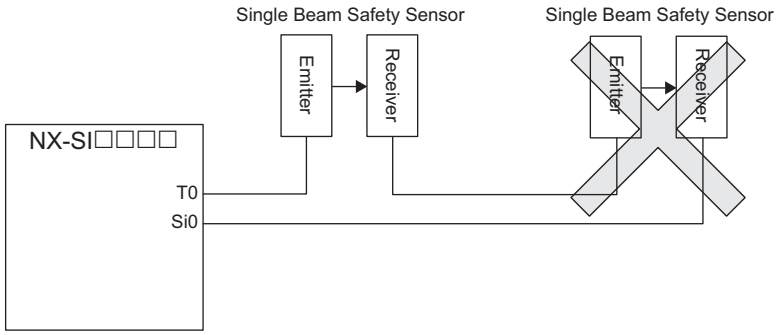


**Precautions for Correct Use**

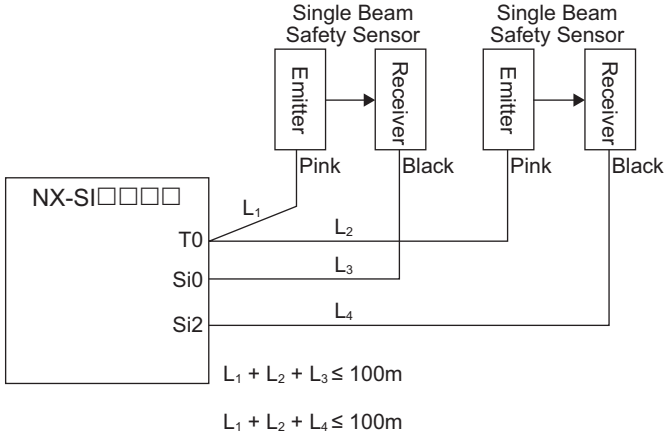
- The maximum number of connections per Unit is as follows:
  NX-SIH400: 4
- You can branch the connections to up to four Single-beam Safety Sensors for each test output.




- Series connections are not possible.



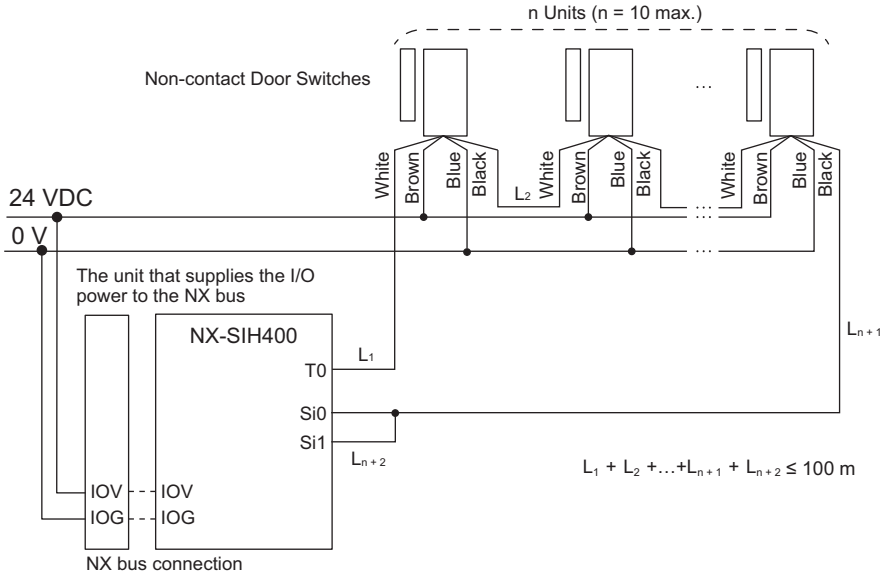

- The total wiring length for the E3ZS/E3FS Single-beam Safety Sensors is 100 m max.



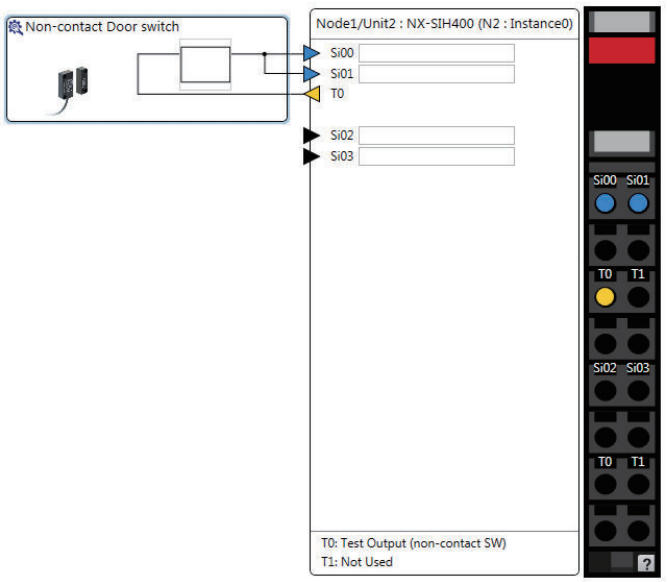

- The E3ZS/E3FS Single-beam Safety Sensor can be used in a Type 2 or lower or PLc or lower application. It cannot be used in a Type 3 or higher, or PLd or higher application.
- If you use more than one Single-beam Safety Sensor, it may not always be possible to detect shorts between wires. Therefore, to satisfy safety category 2, the cables must be protected from external damage for connections to single beam safety sensors. Use ducts or separate cables for each system to protect the cables from external damage when you connect the Single-beam Safety Sensors. You can also use special XS2F Cables for protection.
- The test period for a Single-beam Safety Sensor is 1,200 ms.
  Use this value for reference to determine the standard compliance of your system.

## ● D40A-2/D40A/D40Z Non-contact Door Switches

The non-contact door switch output (black line) from the OMRON D40A-2, D40A or D40Z Non-contact Door Switch is input to a safety input terminal. This is a one-line signal. When connecting it, branch it as shown at Si0 and Si1 in the following figure. Only one test output terminal is used. Connect the D40A-2, D40A or D40Z Non-Contact Door Switch input (white line).

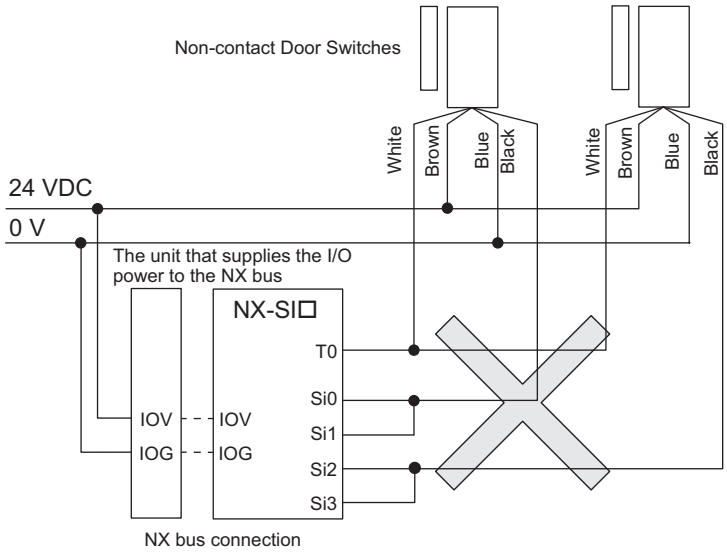



Example of Sysmac Studio Settings:



**Precautions for Correct Use**

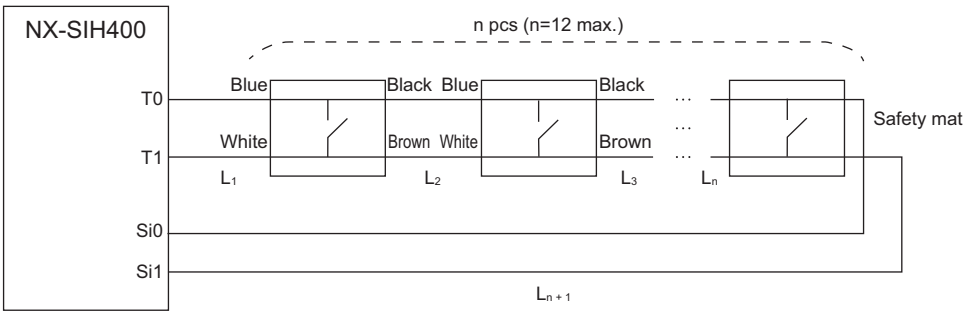
- The maximum number of connections per Unit is as follows:
  NX-SIH400: 20 (10 connected in series × 2 series)
- You can connect up to 10 Non-contact Door Switches to each test output terminal.
- You cannot branch the connections to more than one Non-contact Door Switch from the same test output terminal.



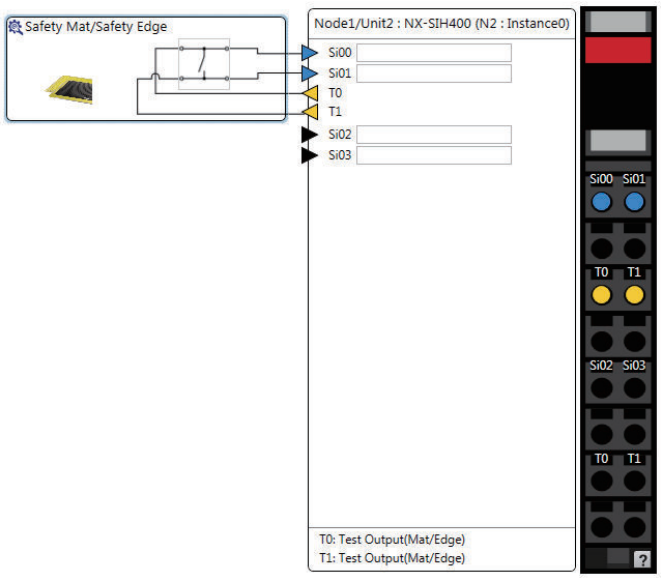

- The total wiring length (L1 + L2 +…+ Ln+2 in the figure above) for the D40A-2, D40A or D40Z Non-contact Switches is 100 m max.
- The D40A-2 Non-contact Door Switch can be used in a Safety Category 4 or lower or a PLe or lower application.
- The D40A Non-contact Door Switch can be used in a Safety Category 3 or lower or a PLd or lower application. It cannot be used in a Safety Category 4 or PLe application.
- The D40Z Non-contact Door Switch can be used in a Safety Category 4 or lower or a PLe or lower application.

## ● UM/UMA Safety Mats

OMRON UM/UMA Safety Mats are connected as shown in the following figure.



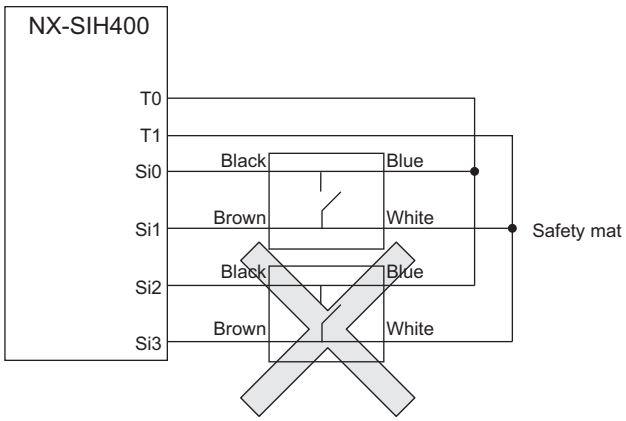

Example of Sysmac Studio Settings:




Refer to *Precaution for Conformance to ISO 13856-1:2013* on page 8-4 for a precaution on conformance to ISO 13856-1:2013.

**Precautions for Correct Use**

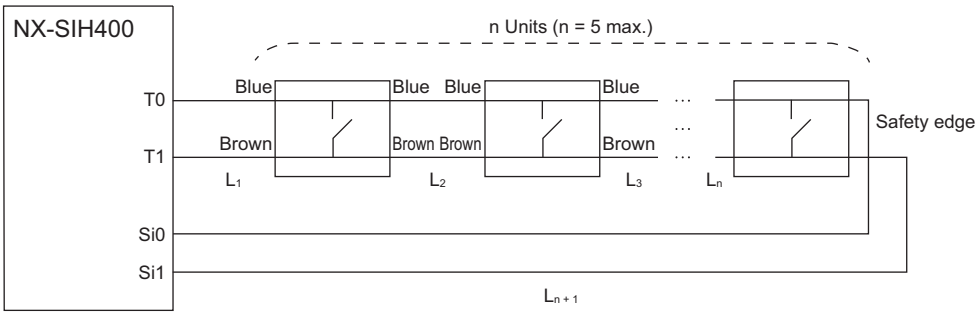
- The maximum number of connections per Unit is as follows:
  NX-SIH400: 12 (12 connected in series × 1 series)
- A Safety Mat requires two test output terminals when it is used. If a Safety Mat is connected to the NX-SIH400 Unit, you cannot connect any additional safety input devices that require test output terminals.
- You can connect up to 12 Safety Mats to the two test output terminals.
- You cannot branch the connections to more than one Safety Mat or Safety Edge from the same test output terminal.



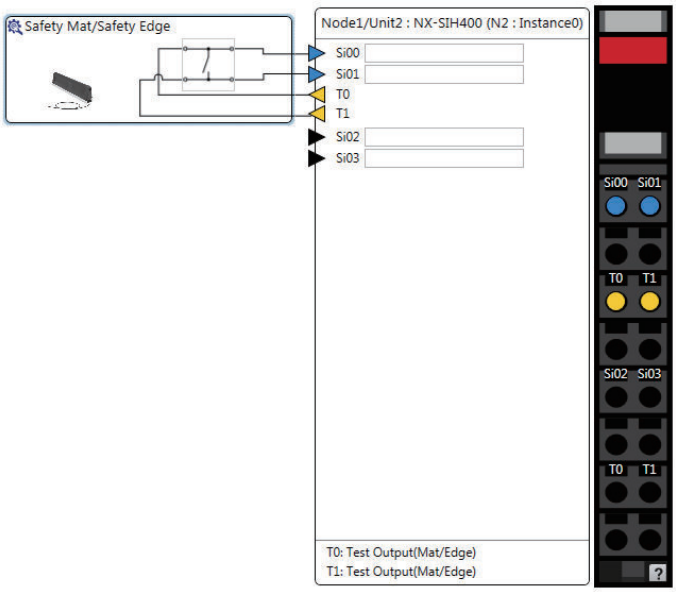

- The total wiring length (L1 + L2 +…+ Ln+1) for the UM/UMA Safety Mats is 100 m max.
- The UM/UMA Safety Mats can be used in a Safety Category 3 or lower or a PLd or lower application. They cannot be used in a Safety Category 4 or PLe application.

## ● SGE Safety Edges

OMRON SGE Safety Edges are connected as shown in the following figure.

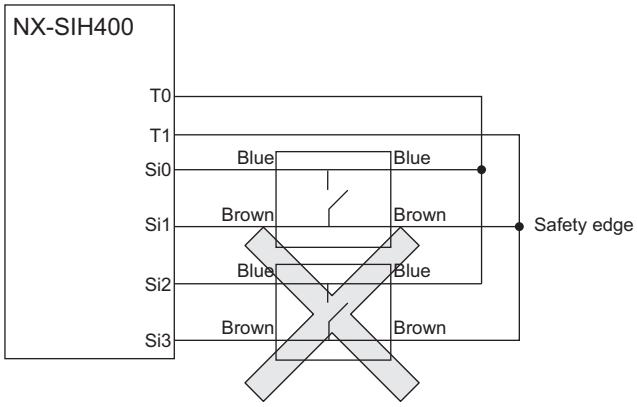



Example of Sysmac Studio Settings:



**Precautions for Correct Use**

- The maximum number of connections per Unit is as follows:
  NX-SIH400: 5 (5 connected in series × 1 series)
- A Safety Edge requires two test output terminals when it is used. If a Safety Edge is connected to the NX-SIH400 Unit, you cannot connect any additional safety input devices that require test output terminals.
- You can connect up to five Safety Eddges to the two test output terminals.
- You cannot branch the connections to more than one Safety Edge or Safety Mat from the same test output terminal.

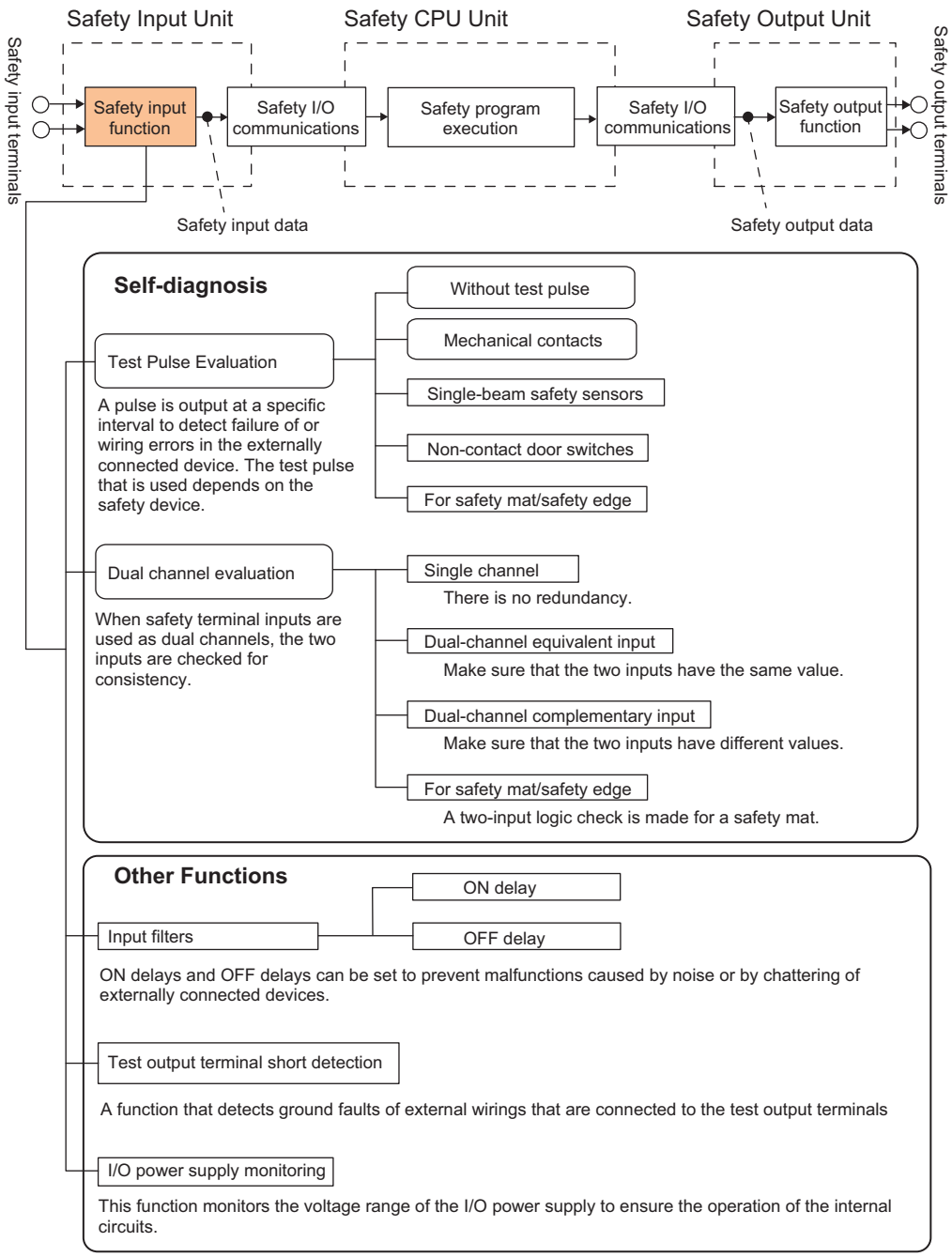



- Safety Edges can be connected only with two wires on each side (no terminating resistance). You cannot connect terminating resistance.
- The total wiring length (L1 + L2 +…+ Ln+1 in the figure above) for the SGE Safety Edges is 100 m max.
- The SGE Safety Edge can be used in a Safety Category 3 or lower or a PLd or lower application. It cannot be used in a Safety Category 4 or PLe application.

## Types of Safety Input Functions

The types of safety input functions that are performed by the Safety Input Unit is shown below.

For the safety input functions, the safety signals that are input to the safety input terminals are evaluated and safety input data that can be used in the safety program is created.
The configuration of the safety input functions is shown in the following figure.
The values that are read from the safety input terminals are passed to the safety program only after they are evaluated by the safety input functions.




The following pages describe the details of the individual safety input functions.

## Test Pulse Evaluation

A test pulse with a specific period is output on the 24-VDC power line from a test output terminal to detect wiring errors and failure of the externally connected device. The following parameters are also used.

- Test Pulse Diagnosis
- Test Source
- Test Pulse Mode

### ● Test Pulse Diagnosis

The Test Pulse Diagnosis setting determines whether to output a test pulse with a specific period from the test output terminal. The parameter determines whether test pulse evaluation is used. This parameter is set according to the type of external device that is connected to the safety input terminal.

| Setting | Description |
|---|---|
| Without Test Pulse | A test pulse from the test output terminal is not output for diagnosis.<br>This setting is used for safety devices with semiconductor outputs that diagnose the OSSD output themselves, such as safety light curtains, and for standard devices. |
| With Test Pulse | A test pulse from the test output terminal is output for diagnosis.<br>The mode of the test output to use as the test source is selected according to the safety device that is connected. |

### ● Test Source

The Test Source setting determines the test output terminal to use when the Test Pulse Diagnosis parameter is set to *with Test Pulse*. The test output terminal is automatically assigned by the Sysmac Studio, but it can be changed to any test output terminal.

### ● Test Pulse Mode

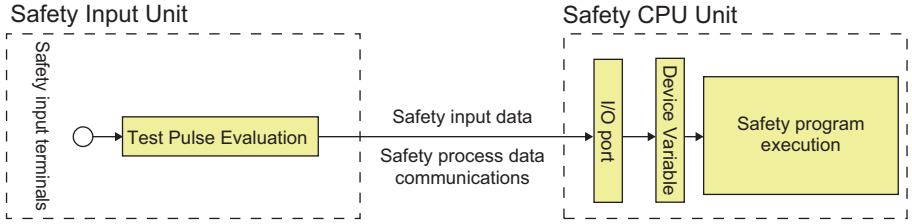
The Test Pulse Mode setting is used to output a test pulse that is suitable for the external device when the Test Pulse Diagnosis parameter is set to *with Test Pulse*.

| Setting | Description |
|---|---|
| Mechanical Contact | The test pulse is connected to a device with mechanical contacts. The test output signal (pulse output) is input to the safety input terminal through the mechanical contact device. The following can be detected: Contact of the input signal line with the positive side of the power supply line, ground faults, and short-circuits to the other input signal lines. |
| Single Beam Safety Sensor | An OMRON E3ZS/E3FS Single-beam Safety Sensor is connected. A test signal for Single-beam Safety Sensor diagnosis is output. |
| Non-contact Door Switch | An OMRON D40A-2, D40A or D40Z Non-contact Door Switch is connected. Test signals for the D40A-2, D40A or D40Z will be output. |
| Safety Mat/Safety Edge | An OMRON UM/UMA Safety Mat or SGE Safety Edge (4-wire) is connected. A test signal for Safety Mat/Safety Edge diagnosis is output. |

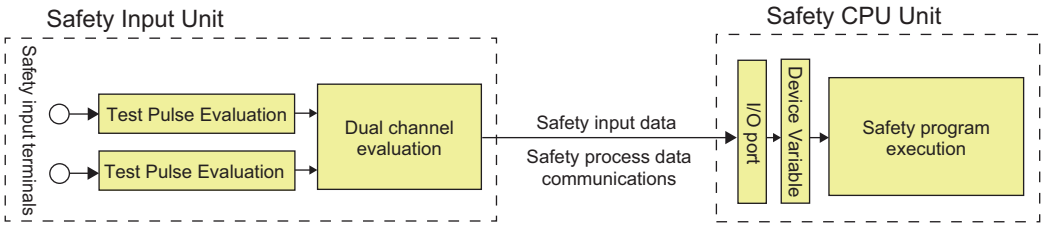
## Dual Channel Evaluation

Safety input terminals can be used as dual channels (one pair). The dual channel evaluation evaluates the data for two inputs to check for discrepancy.

- Single Channel




- Dual Channels




The following parameters are also used.

- Single/Dual
- Discrepancy Time

## ● Single/Dual

Set the evaluation method to use with the safety input terminals.

| Setting | Description |
|---|---|
| Single Channel | The safety input terminals are used as independent safety input terminals. |
| Dual Channel Equivalent | The safety input terminals are used as dual-channel-equivalent inputs. |
| Dual Channel Complementary | The safety input terminals are used as dual-channel-complementary inputs. |
| Safety Mat/Safety Edge | The safety input terminals are used as safety mat/safety edge inputs. |

## ● Discrepancy Time

For two inputs set in a Dual Channel Mode, the time is monitored from a change in the value of one input to a change in the value of the other input. An error occurs if the value of the other input does not change within the set discrepancy time.

The discrepancy time can be set to any of the following eight values between 500 ms and 64,000 ms.

1: 500 [ms], 2: 1000 [ms], 3: 2000 [ms], 4: 4000 [ms], 5: 8000 [ms], 6: 16000 [ms], 7: 32000 [ms], 8: 64000 [ms]

The discrepancy time cannot be set in Single Channel Mode.

## ● Relationship between the Single/Dual Setting and Safety Input Data

The signals that are input to safety input terminals are evaluated as shown in the following table. This safety input data can be used in the safety program in the Safety CPU Unit.

- Relationship between Input Signals to Safety Input Terminals and Safety Input Data for Single-channel Inputs

| Single/Dual | Input signals on the safety input terminals | Safety input data | Meaning of status |
|---|---|---|---|
| | Si (x) | Si (x) | |
| Single Channel | 0 | 0 | Inactive (OFF) |
| | 1 | 1 | Active (ON) |

- Relationship between Input Signals to Safety Input Terminals and Safety Input Data for Dual-channel Inputs

n = Even number

| Single/Dual | Input signals on the safety input terminals | | Safety input data | | Meaning of status |
|---|---|---|---|---|---|
| | Si (n) | Si (n+1) | Si (n) | Si (n+1) | |
| Dual Channel Equivalent | 0 | 0 | 0 | 0*1 | Inactive (OFF) |
| | 0 | 1 | 0 | 0*1 | Discrepant status |
| | 1 | 0 | 0 | 0*1 | Discrepant status |
| | 1 | 1 | 1 | 0*1 | Active (ON) |
| Dual Channel Complementary | 0 | 0 | 0 | 0*1 | Discrepant status |
| | 0 | 1 | 0 | 0*1 | Inactive (OFF) |
| | 1 | 0 | 1 | 0*1 | Active (ON) |
| | 1 | 1 | 0 | 0*1 | Discrepant status |

*1. If the terminals are set to Dual Channel Mode, the safety program in the Safety CPU Unit must access the safety input data for the even-numbered terminal.

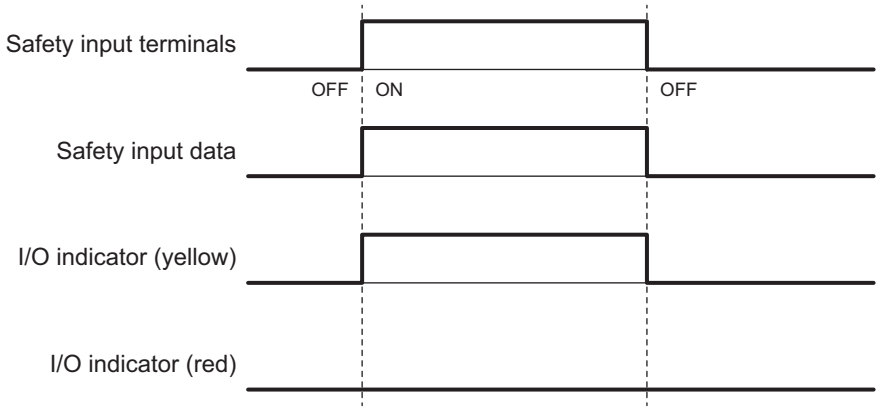
- Relationship between Safety Mat Status and Safety Input Data for Safety Mat/Safety Edge Inputs

n = Even number

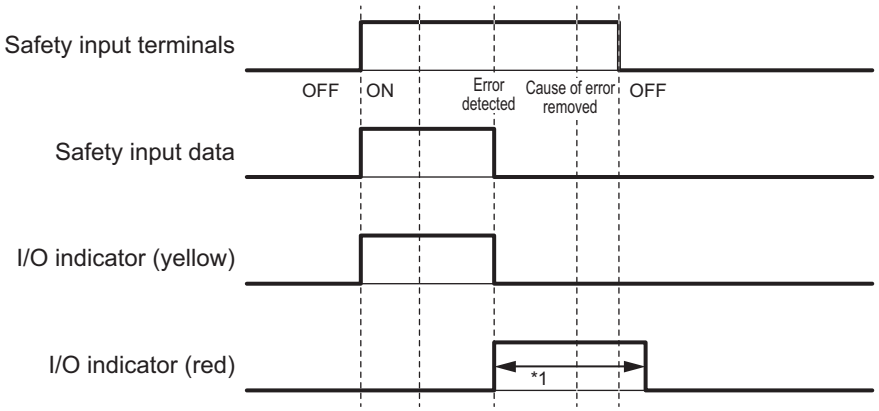
| Single/Dual | Safety mat/safety edge status | Safety input data | | Meaning of status |
|---|---|---|---|---|
| | | Si (n) | Si (n+1) | |
| Safety Mat/Safety Edge | Without load | 1 | 0*1 | Active (ON) |
| | With load | 0 | 0*1 | Inactive (OFF) |

*1. If the terminals are set to Dual Channel Mode, the safety program in the Safety CPU Unit must access the safety input data for the even-numbered terminal.

- Operation for Single Channel: Normal Operation

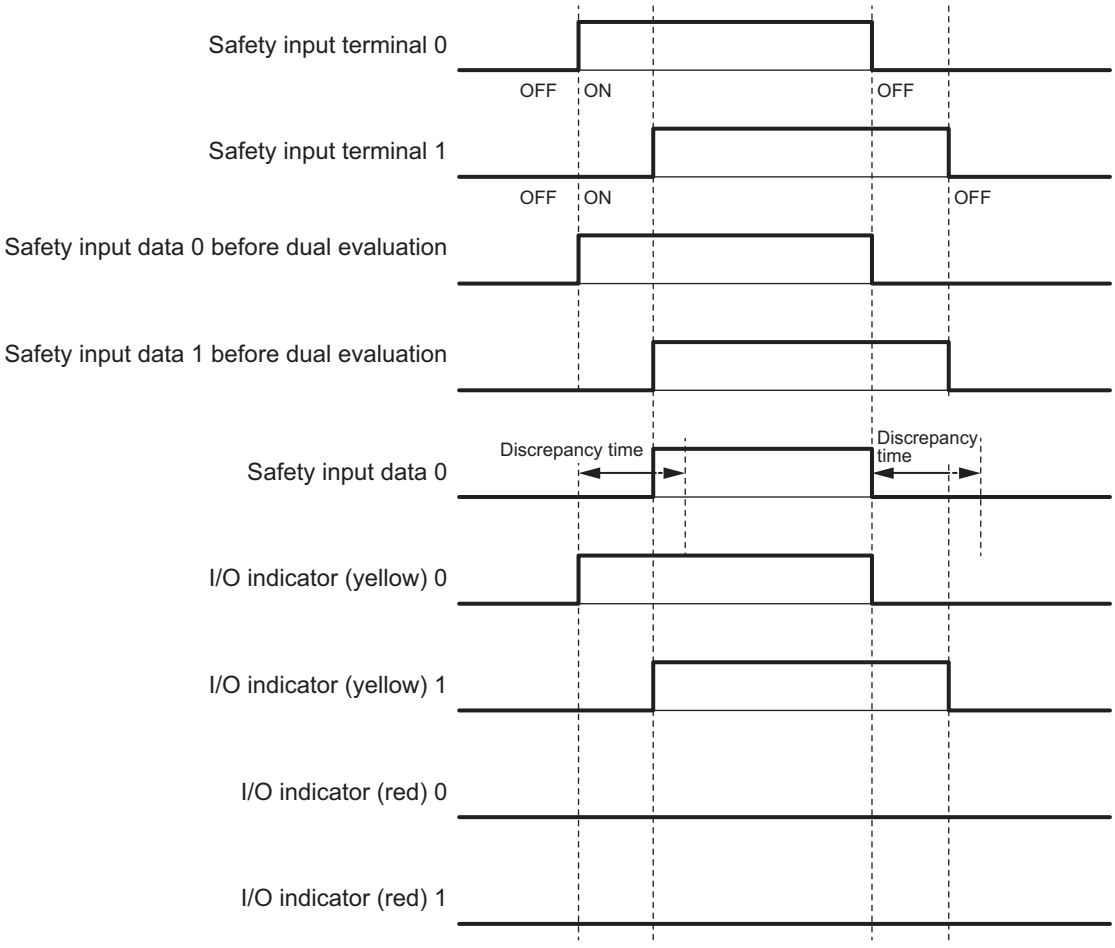



- Operation for Single Channel: Test Pulse Evaluation Error by Stuck-at-high Error

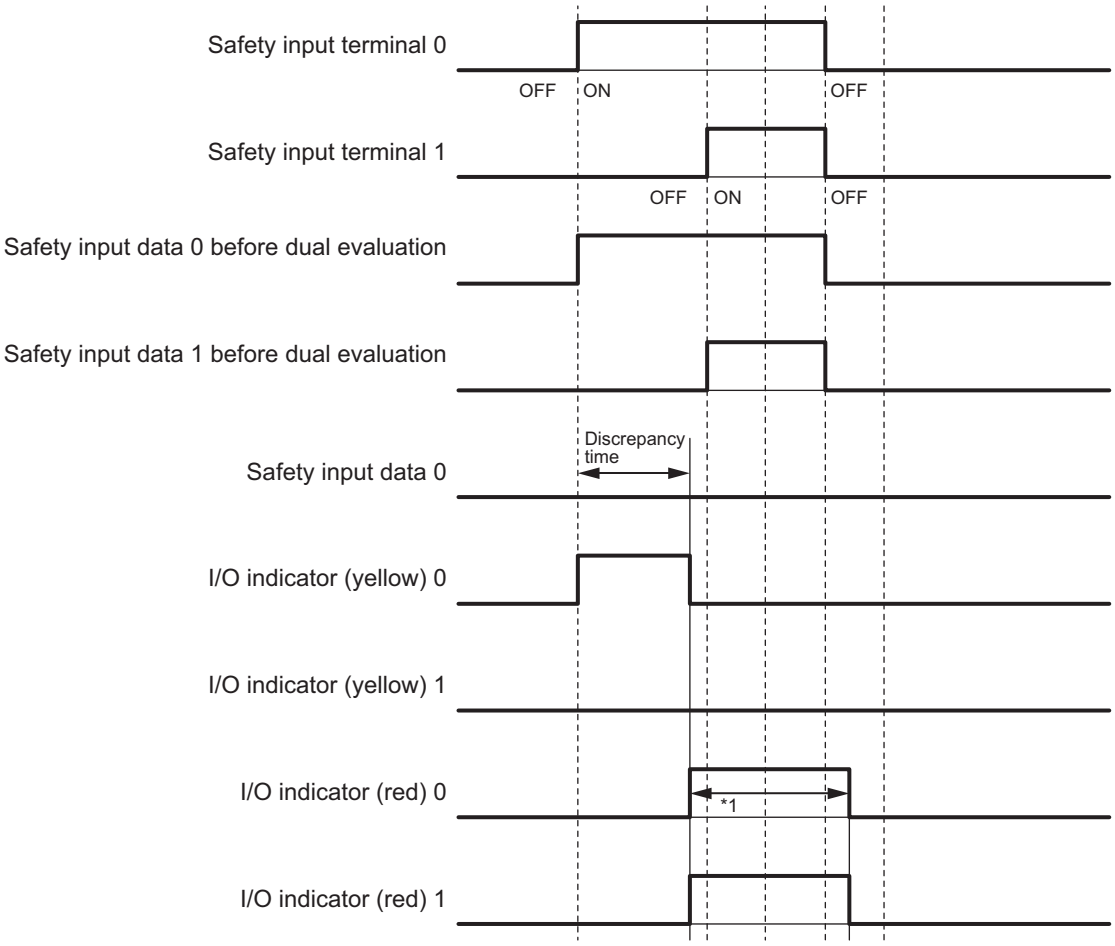



*1. This is the time that the error status (control data, status data, and indicator status) is held (1 s min.).

- Operation for Dual-channel Equivalent Inputs: Normal Operation




- Operation for Dual-channel Equivalent Inputs: Discrepancy Error




*1. This is the time that the error status (control data, status data, and indicator status) is held (1 s min.).

## Errors Detected during Self-diagnosis

The errors that can be detected for safety input terminals are determined by the parameter settings. The following table gives the errors that are detected for each parameter setting.

### ● Devices with Mechanical Contacts and Devices with Semiconductor Outputs

| Setting | | Error detection | | | |
|---|---|---|---|---|---|
| Single/Dual | Test pulse | Contact with positive side of power line | Ground fault*1 | Disconnection | Short circuits in input wiring |
| Single Channel | Without Test Pulse | Not detectable. | Not detectable. | Not detectable. | --- |
| | With Test Pulse | Detectable. | Detectable when input turns ON. | Not detectable. | --- |

| Setting | | Error detection | | | |
|---|---|---|---|---|---|
| Single/Dual | Test pulse | Contact with positive side of power line | Ground fault*1 | Disconnection | Short circuits in input wiring |
| Dual Channel Equivalent | Without Test Pulse | Not detectable.*2 | Not detectable. | Detectable when input turns ON. | Not detectable.*2 |
| | Same test source for pair of safety input terminals | Detectable. | Detectable when input turns ON. | Detectable when input turns ON. | Not detectable. |
| | Different test sources for pair of safety input terminals | Detectable. | Detectable when input turns ON. | Detectable when input turns ON. | Detectable. |
| Dual Channel Complementary | Without Test Pulse | Detectable when input turns ON or OFF. | Detectable when input turns ON or OFF. | Detectable when input turns ON or OFF. | Detectable. |
| | Same test source for pair of safety input terminals | Detectable. | Detectable when input turns ON or OFF. | Detectable when input turns ON or OFF. | Detectable. |
| | Different test sources for pair of safety input terminals | Detectable. | Detectable when input turns ON or OFF. | Detectable when input turns ON or OFF. | Detectable. |

*1. To detect ground faults, the 0-V line of the external power supply must be grounded.
*2. Detection is possible with the OSSD diagnostic function of the light curtain or laser scanner.

## ● Single-beam Safety Sensors, Non-contact Door Switches, Safety Mats, and Safety Edges

| Input device | Error detection | | | | | |
|---|---|---|---|---|---|---|
| | Contact with positive side of power line | Ground fault*1 | Disconnection | Short circuits in input wiring | Failure of input device | Sensor bypass |
| Single-beam Safety Sensor | Detectable. | Not detectable. | Not detectable. | --- | Not detectable. | Detectable. |
| D40A-2/D40A Non-contact Door Switch | Detectable. | Not detectable. | Not detectable. | --- | Detectable. | Detectable. |
| D40Z Non-contact Switch | Detectable. | Not detectable. | Not detectable. | --- | Detectable. | Detectable. |
| Safety Mat/Safety Edge | Detectable. | Detectable. | Detectable. | Not detectable. | Not detectable. | --- |

*1. To detect ground faults, the 0-V line of the external power supply must be grounded.



**Additional Information**

To detect burnouts in a muting lamp, use a PIT si1.2 Muting Lamp manufactured by Pilz, which supports defective lamp detection.

## Input Filters

The input filter helps prevent malfunctions that are sometimes caused by chattering or noise from the external device that is connected to a safety input terminal.
You can filter out chattering and noise from the external device for the widths that are set with the ON delay time and OFF delay time.
ON delays and OFF delays can be set to one of the 10 options given below, from 0 to 1,536 ms, for each safety input terminal.

1: 0 [ms], 2: 6 [ms], 3: 12 [ms], 4: 24 [ms], 5: 48 [ms], 6: 96 [ms], 7: 192 [ms],
8: 384 [ms], 9: 768 [ms], 10: 1536 [ms],

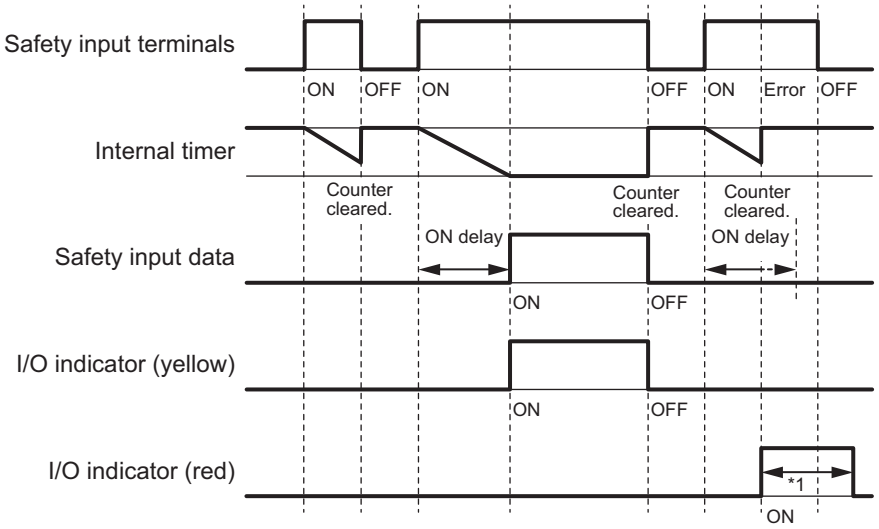
The effect of chattering from external devices can be reduced more by increasing the delay time, but this will slow the response to input signals.
The input filter can be used with dual channel evaluation.

**Precautions for Correct Use**

If an OFF delay is used, the OFF delay time affects the safety reaction time. Add the OFF delay time to the safety reaction time. (Refer to *Section 8 Calculating Safety Reaction Times* on page 8-1.)

### ● Operation with an ON Delay

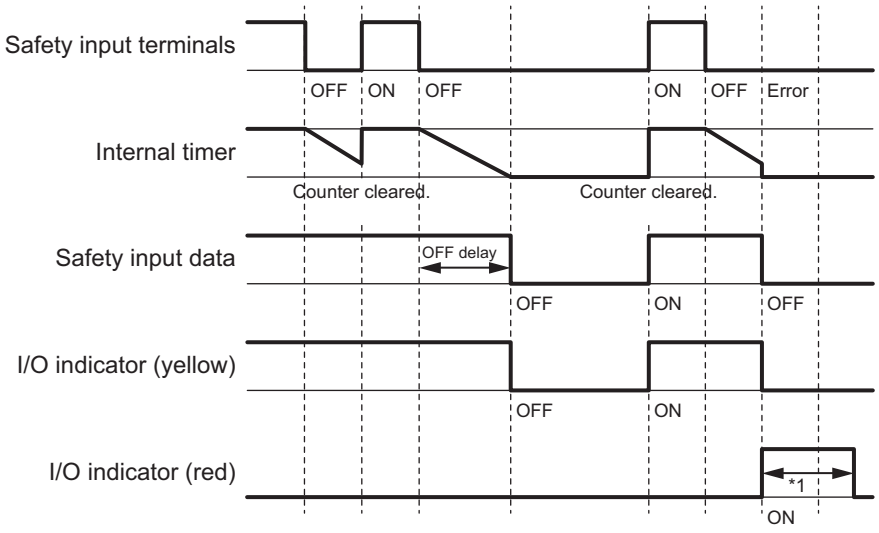
You can filter out ON pulses for the width that is set with the ON delay time.




*1. This is the time that the error status (control data, status data, and indicator status) is held (1 s min.).

### ● Operation with an OFF Delay

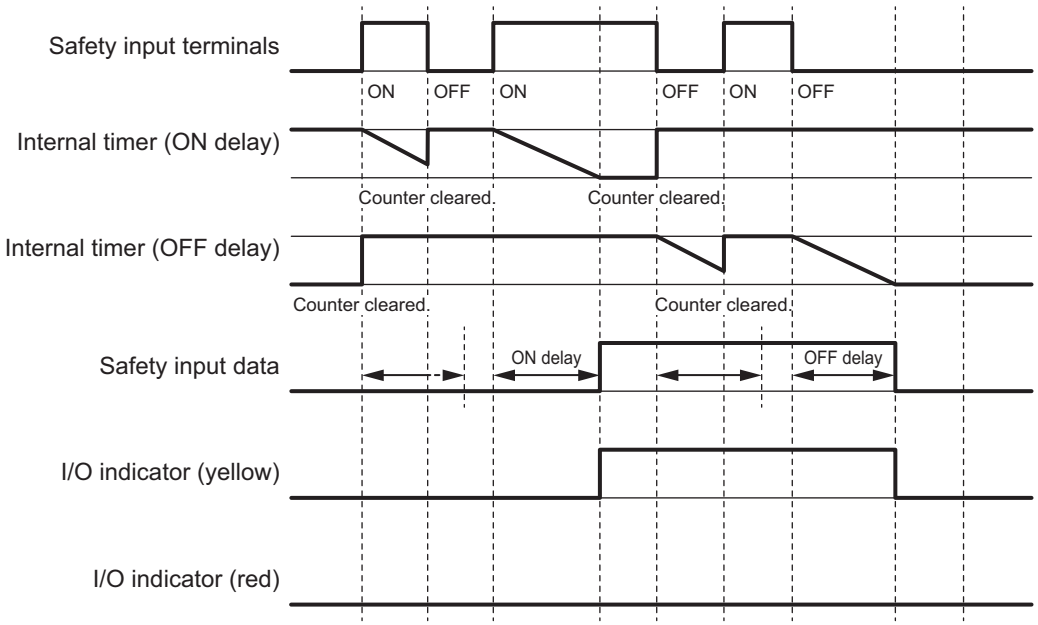
You can filter out OFF pulses for the width that is set with the OFF delay time.




*1. This is the time that the error status (control data, status data, and indicator status) is held (1 s min.).

### ● Operation with Both an ON Delay and OFF Delay

You can filter out ON pulses for the width that is set with the ON delay time and filter out OFF pulses for the width that is set with the OFF delay time.




## Test Output Terminal Short Detection

The test output terminal short detection prevents the internal circuits of the test output terminals from being destroyed if an overcurrent flows due to a ground fault or other cause.

If an overcurrent is detected, the safety input data for the safety input terminal that is being used as the test source for the terminal is turned OFF.

At this time, an *Overload Detected at Test Output* event will occur. To troubleshoot errors, refer to *Section 12 Troubleshooting* on page 12-1.

## I/O Power Supply Monitoring

I/O power supply monitoring monitors the voltage range of the I/O power supply.
If a voltage that is less than the specified range is detected, all safety inputs for the Unit are turned OFF.
At this time, an *I/O Power Supply Voltage Error* occurs. To troubleshoot errors, refer to *Section 12 Troubleshooting* on page 12-1.
This function does not work if all of the terminals are set as unused terminals.

# 4-3-2 Safety Output Function

## Connectable Output Devices

The Safety Output Unit diagnoses the connected external devices through the safety output terminals. The general-purpose safety output devices that can be connected to the safety output terminals of a Safety Output Unit are listed in the following table.

| Type | Examples |
|---|---|
| Safety devices that can be connected to PNP outputs | Safety relays, contactors, or safety inputs from servo drives |

**Additional Information**

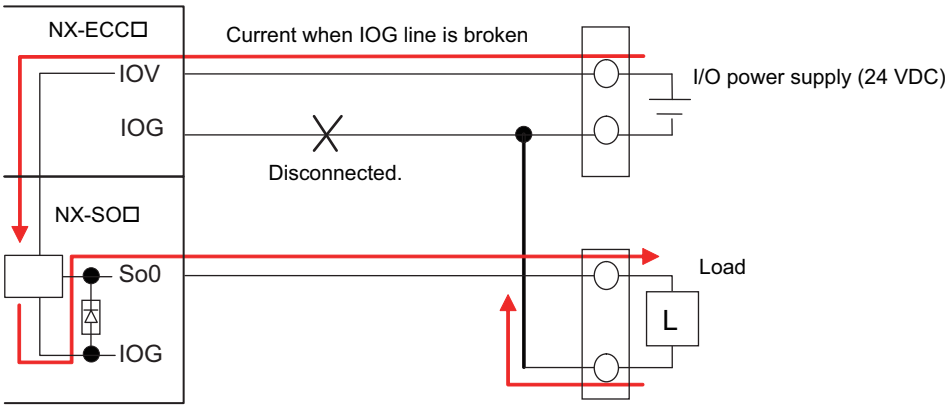
The connection of incandescent lamps is not supported. Connect them to an NX-series Digital Output Unit.

## Setting the Safety Functions for Safety Output Terminals

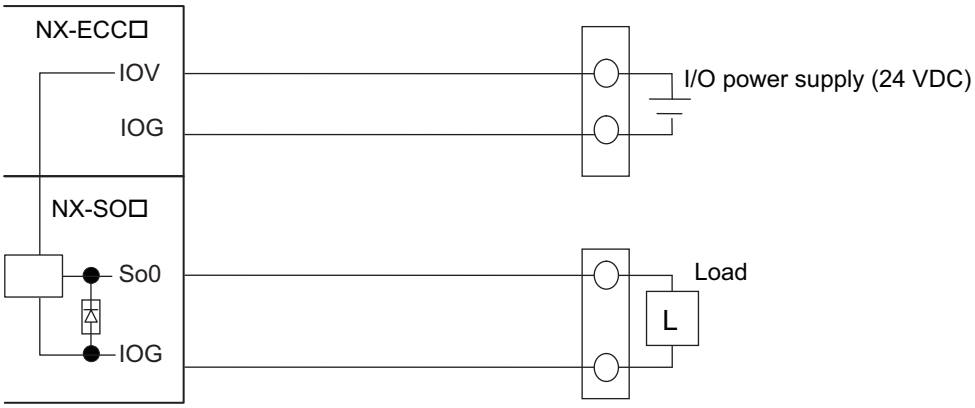
You can easily set the safety functions of the safety output terminals from the Sysmac Studio by selecting the external devices that are connected.
Refer to *4-3 Safety I/O Function* on page 4-10 for details.

## Connecting the I/O Power Supply

This section describes the connection methods for the I/O power supply.
If the Safety Output Unit is wired as shown in the following figure and the IOG wire breaks, a floating condition will result. If that occurs, a few volts may be applied to the output terminals of the Safety Output Unit, turning ON the load.




Use the wiring that is shown in the following figure to prevent a floating condition for the IOG of the Safety Output Unit even if the IOG line is broken.

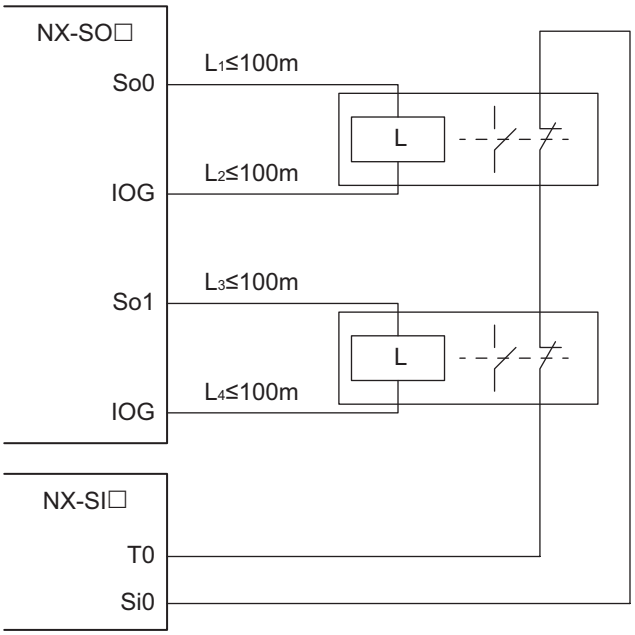



## Connecting Output Devices

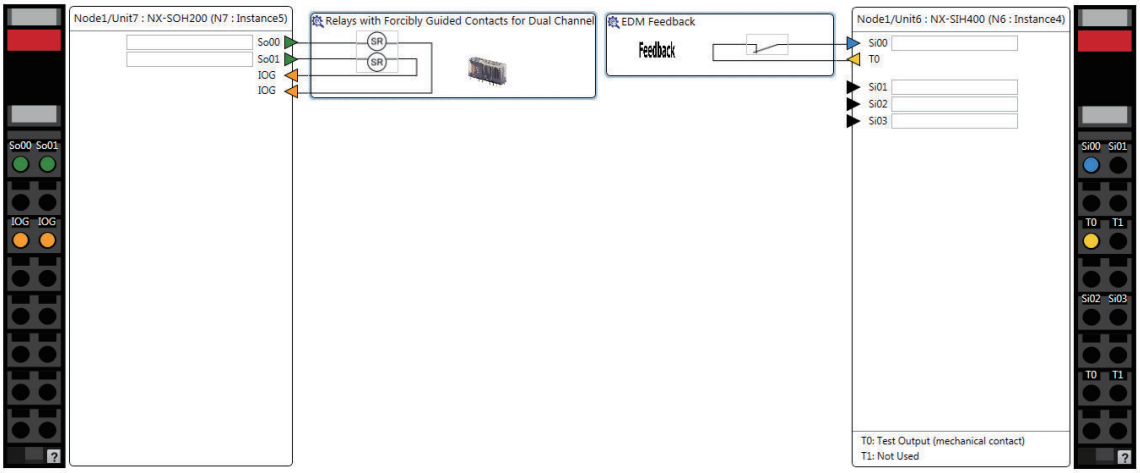
This section describes the connection methods for output devices.

### ● Safety Relays and Contactors

Connect a safety relay or contactor as shown in the following figure.




Example of Sysmac Studio Settings:

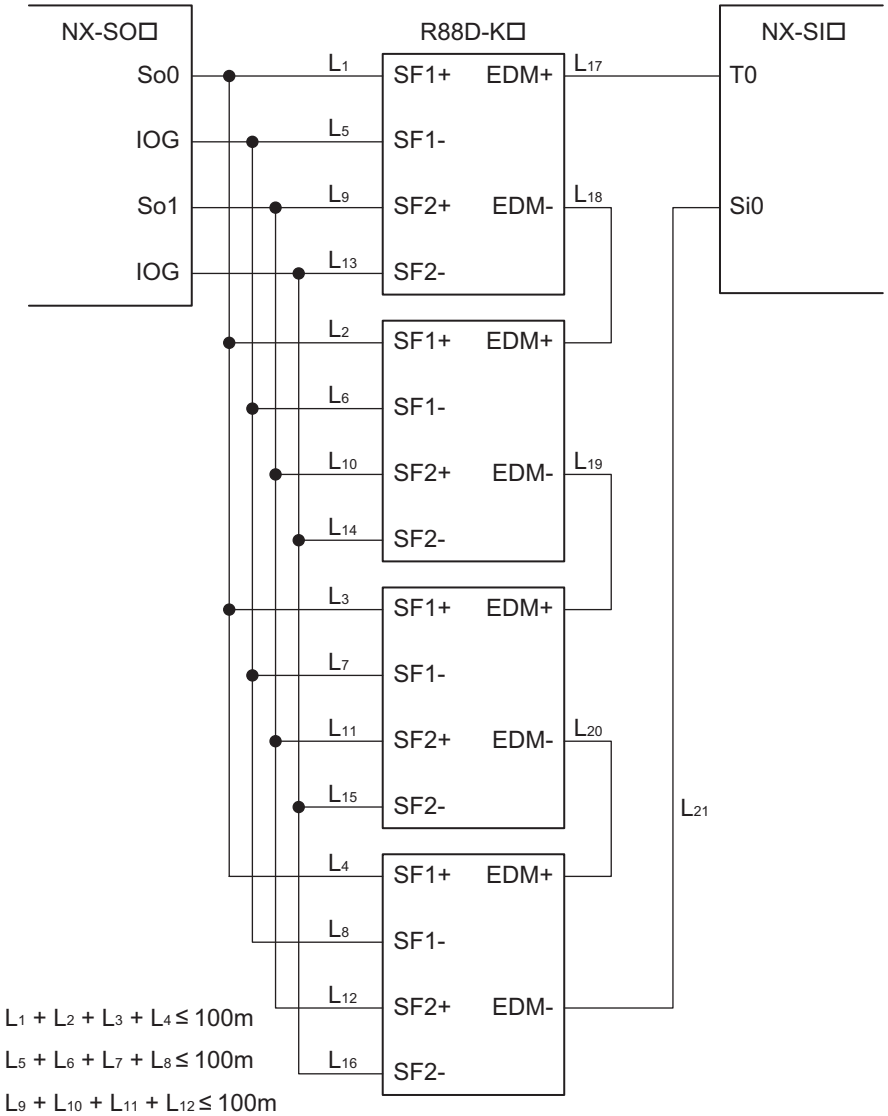



### Precautions for Correct Use

- The line length from the safety output terminals to the output devices (L1, L2, L3, and L4) is 100 m max. for each line.
- The total length of cable that is connected to one test output must be as described in *4-3-1 Safety Input Function* on page 4-10.

## ● Servo Drive

OMRON R88D-K☐ Servo Drives are connected as shown in the following figure.




$L_1 + L_2 + L_3 + L_4 \leq 100m$

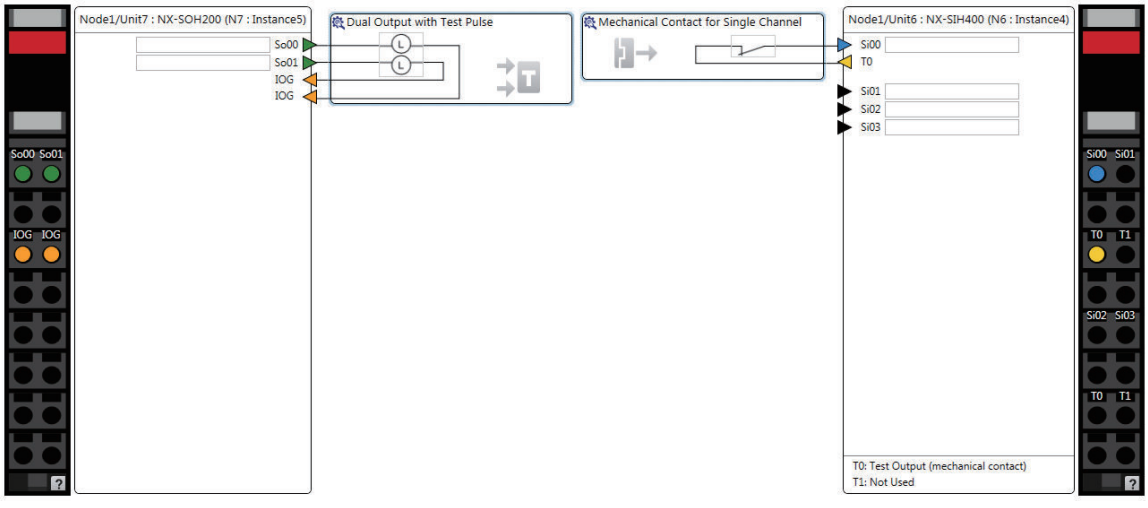
$L_5 + L_6 + L_7 + L_8 \leq 100m$

$L_9 + L_{10} + L_{11} + L_{12} \leq 100m$

$L_{13} + L_{14} + L_{15} + L_{16} \leq 100m$

$L_{17} + L_{18} + L_{19} + L_{20} + L_{21} \leq 100m$

Example of Sysmac Studio Settings:



### Precautions for Correct Use

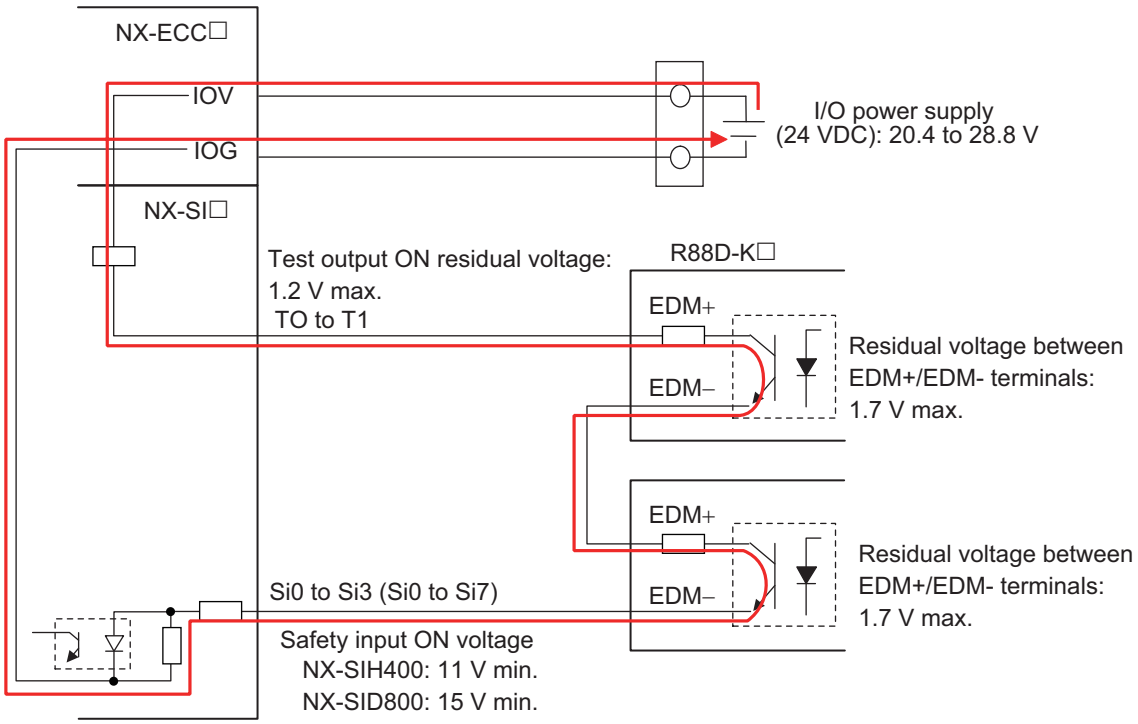
- If you directly connect more than one EDM terminal to one safety input terminal, the voltage at the safety input terminal will be reduced proportionately to the number of connections. When determining the number of direct connections, consider the voltage drop between the EDM terminals in the design.

  **Formula**

  {[I/O power supply − Test output ON residual voltage − Safety input ON voltage}
  /Residual voltage between EDM terminals ≥ Number of direct connections

  **Theoretical Values**

  NX-SID800: Two Units max. per safety input terminal
  NX-SIH400: Four Units max. per safety input terminal



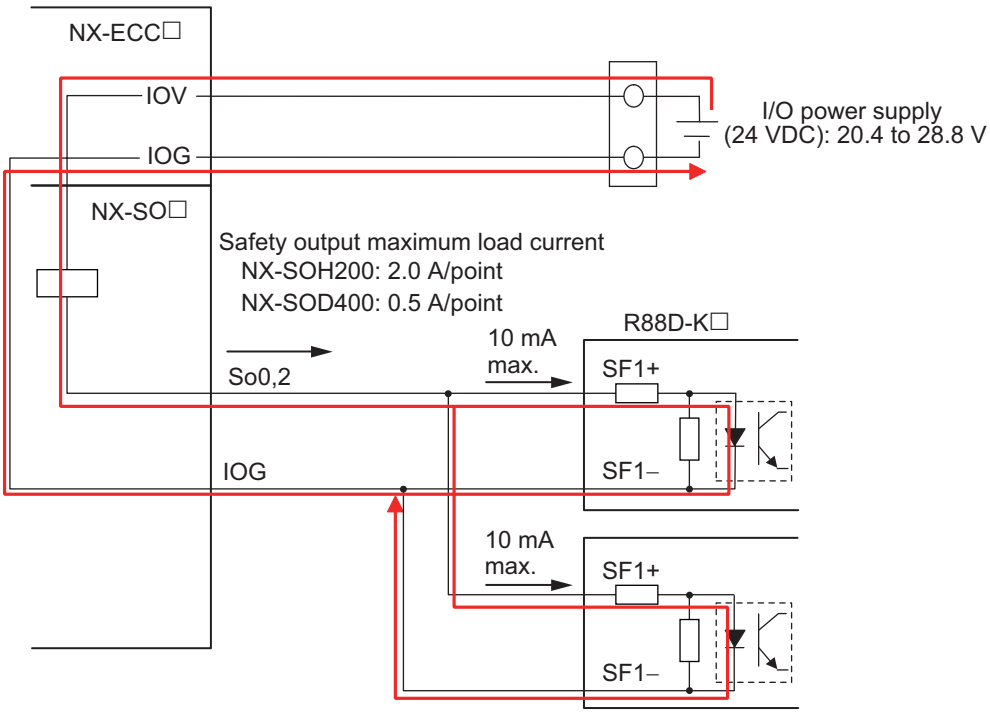

- If you branch connections to more than one SF terminal from one safety output terminal, the load current at the safety output terminal will be increased proportionately to the number of connections. When determining the number of branch connections, consider the input current to the SF terminals in the design.

  **Formula**

  {Maximum load current per safety output terminal}
  / {Input current per SF terminal} ≥ Number of branch connections

  **Theoretical Values**

  NX-SOD400: 50 Units max. per safety output terminal
  NX-SOH200: 200 Units max. per safety output terminal
  It is also necessary to consider the mounting directions, ambient temperature, and Unit total load current.



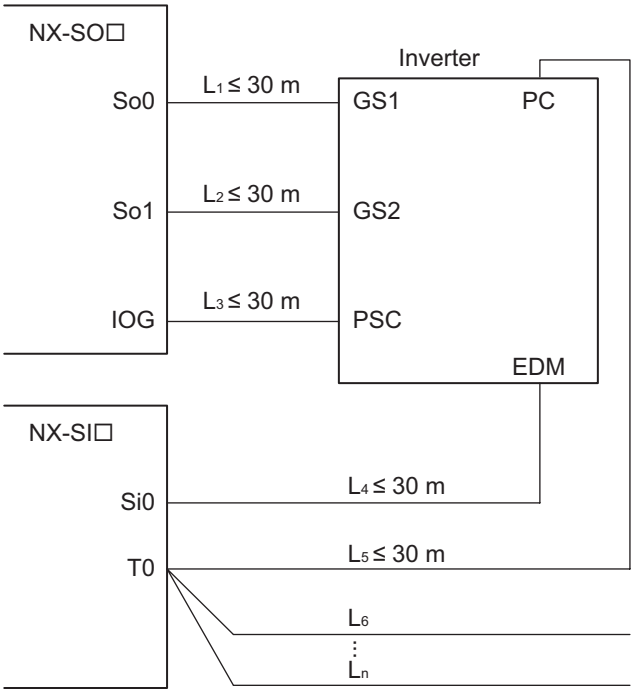

- The total wiring length from the safety output terminal to the output device (L1 + L2 + L3 + L4, L5 + L6 + L7 + L8, L9 + L10 + L11 + L12, and L13 + L14 + L15 + L16) is 100 m max.
- The total wiring length of cables (L17 + L18 + L19 + L20 + L21) that can be connected to one test output is 100 m max.
- Set the input device to*Mechanical Contact Type* to set the NX-SI□ EDM connection terminals.
- An RD88-K□ Servo Drive can be used in a Safety Category 3 or lower or a PLd or lower application. It cannot be used in a Safety Category 4 or PLe application.
- Refer to the following manuals for details on the safety function settings and the precautions for the correct use of the R88D-K□ Servo Drive.
  a) *AC Servomotors/Servo Drives G5-series with Built-in EtherCAT Communications User's Manual (Cat. No. I576)*
  b) *AC Servomotors/Servo Drives G5-series with Built-in EtherCAT Communications Linear Motor Type User's Manual (Cat. No. I577)*

### Additional Information

- A special connector (R88A-CNK81S) is required to connect the RD88D-K□.
- The wiring diagram shown above is an example that turns OFF four axes simultaneously. Alternatively, each axis can be wired to a separate safety I/O terminal.

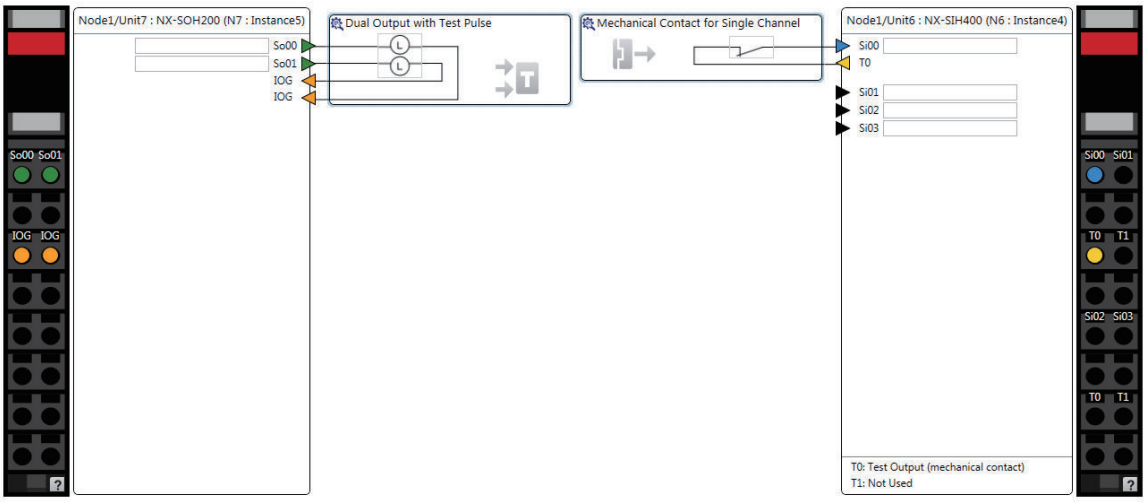
## ● Inverters

OMRON 3G3MX2 Inverters are connected as shown in the following figure.




$L_4 + L_5 + L_6 + ... + L_n \leq 100$ m

Example of Sysmac Studio Settings:



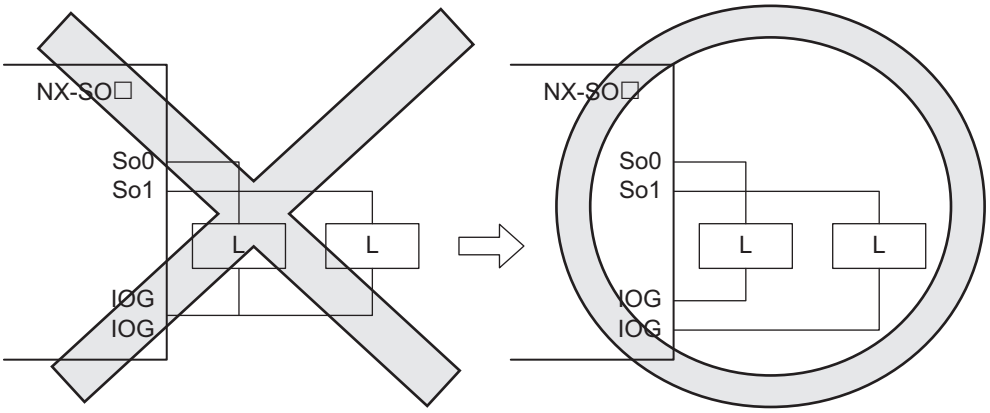
**Precautions for Correct Use**

- The maximum number of connections per Unit is as follows:
  When NX-SOD400 and NX-SI□ are used: 2 (1 connected in series × 2 series)
  When NX-SOH200 and NX-SI□are used: 1 (1 connected in series × 1 series)
- The total wiring length of cables (L4 + L5 + L6 + ... + Ln) that can be connected to one test output is 100 m max.
- Set the input device to *Mechanical Contact Type* to set the NX-SI□ EDM connection terminals.
- The total wiring length of cable that can be connected from the safety output terminal to the output device (L1 + L2 + L3) is 30 m max.
- The 3G3MX2 Inverter can be used in a Safety Category 3 or lower or a PLd or lower application.It cannot be used in a Safety Category 4 or PLe application.
- Refer to the *Multi-function Compact Inverter 3G3MX2-V1 User's Manual (Cat. No. I585)* for the safety function settings and application precautions for a 3G3MX2 Inverter.

### ● Connecting More Than One Output Device

The IOG terminals on the Safety Output Unit are connected internally in the Unit. Make sure that the current that flows through each IOG terminal is less than the current capacity of the I/O power supply terminals.

If the wiring is shared for the IOG lines to the output devices, the sum of the output currents will flow in the IOG line. Therefore, wire the IOG lines separately.

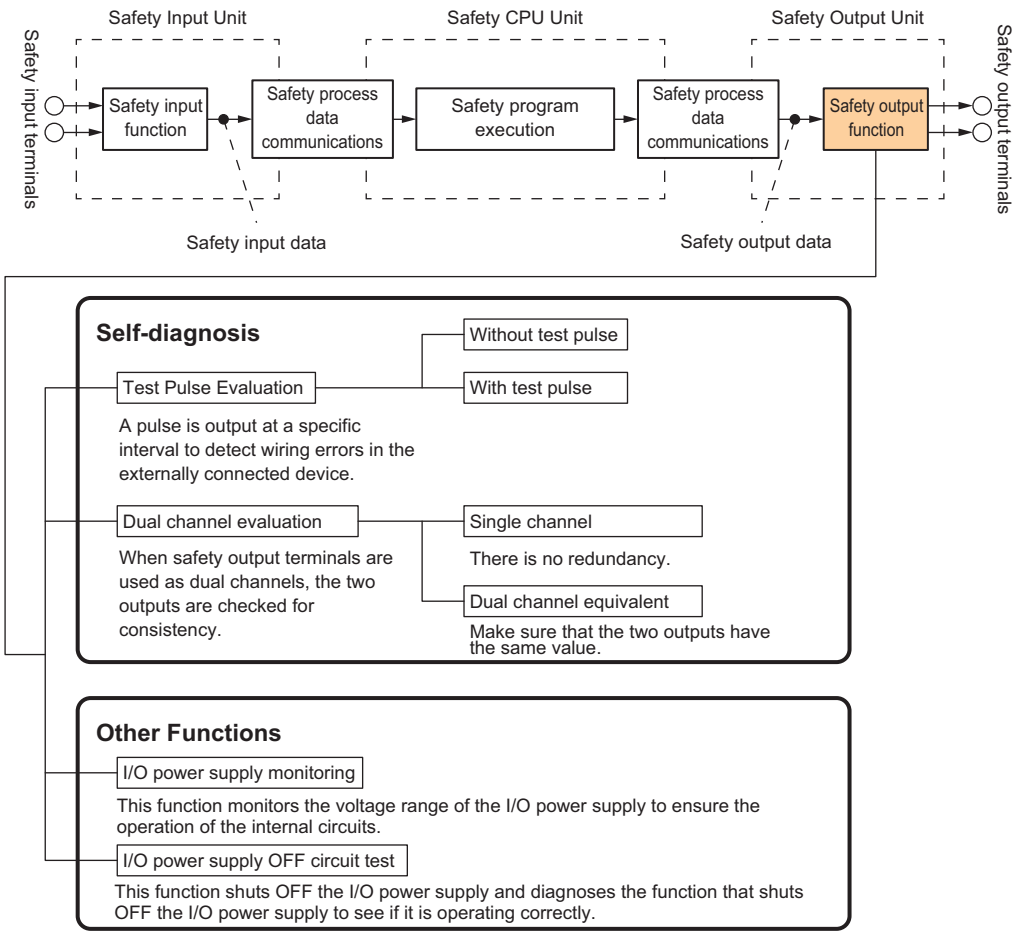



## Types of Safety Output Functions

The types of safety output functions that are performed by the Safety Output Unit is shown below.

The safety output functions diagnose the outputs to the safety output terminals and the external device wiring based on the safety output data from the safety program.

The execution results of the safety program are evaluated by the safety output functions and the evaluation results are output from the safety output terminals.




## Test Pulse Evaluation

The test pulse evaluation outputs a test pulse with a specific period on the 24-VDC power line from a safety output terminal to detect errors in wiring to the externally connected device. This evaluation is achieved through the *Test Pulse Diagnosis* parameter.

### ● Test Pulse Diagnosis

The Diagnosis setting determines whether to output a test pulse with a specific period from the safety output terminal. The parameter determines whether test pulse evaluation is used.
The errors that can be detected are determined by the parameter settings.
Refer to *Errors Detected during Self-diagnosis* on page 4-28 for the errors that can be detected for each parameter setting.

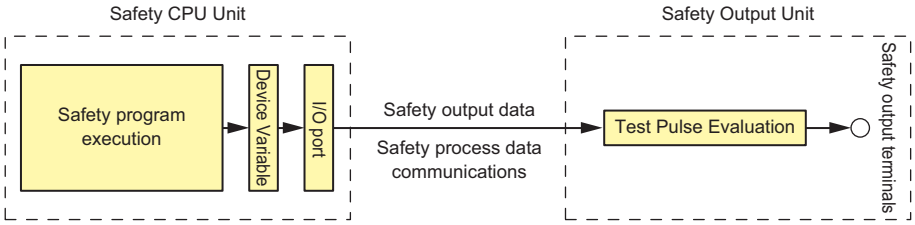
**Precautions for Correct Use**

When the Test Pulse Diagnosis parameter is set to *with Test Pulse*, OFF pulse signals with a pulse width of 640 µs are output while the safety output is ON to diagnose the output circuit. Check the input response time of the connected control device to make sure it will not malfunction due to these OFF pulses.

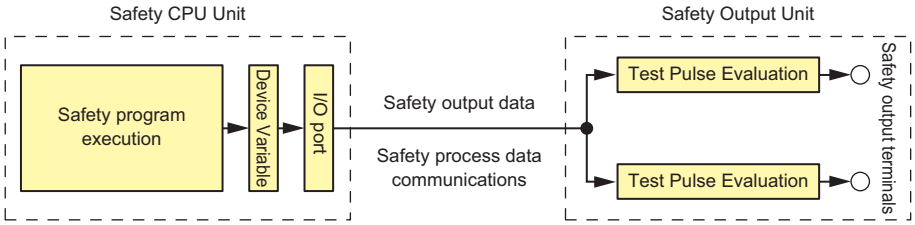
## Dual Channel Evaluation

The use of dual-channel-equivalent outputs lets you control two safety output terminals with one safety output data from the safety program. If an error is detected in either of the two output terminals, the outputs to the external devices are both turned OFF.

- Single Channel




- Dual Channels




This evaluation is achieved through the *Single/Dual* parameter.

### ● Single/Dual

Set the evaluation method to use with the safety output terminals.

| Setting | Description |
|---|---|
| Single Channel | The safety output terminals are used as independent safety output terminals. |
| Dual Channel Equivalent | The pair of safety output terminals are used as dual channel outputs. The output is ON if the paired safety output terminals are both normal. |

### ● Relationship between the Single/Dual Setting and Safety Output Data

The safety output data that is used in the safety program is output to the safety output terminals according to the Single/Dual parameter as shown below.

- Relationship between Safety Output Data and Signals Output from Safety Output Terminals for Single-channel Outputs

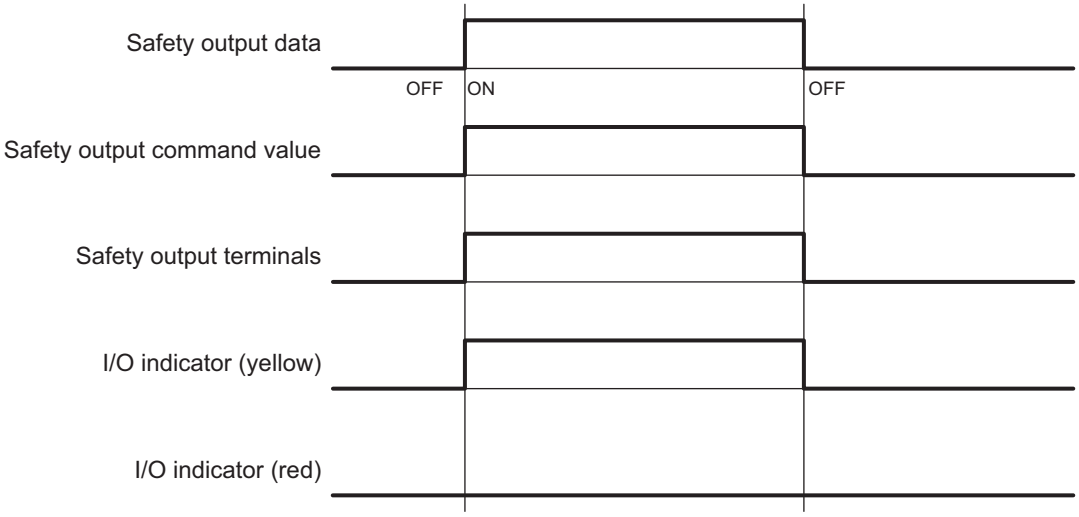
| Single/Dual | Safety output data | Output signal on the safety output terminal | Meaning of status |
|---|---|---|---|
| | So (x) | So (x) | |
| Single Channel | 0 | 0 | Inactive (OFF) |
| | 1 | 1 | Active (ON) |

- Relationship between Safety Output Data and Signals Output from Safety Output Terminals for Dual-channel Equivalent Outputs

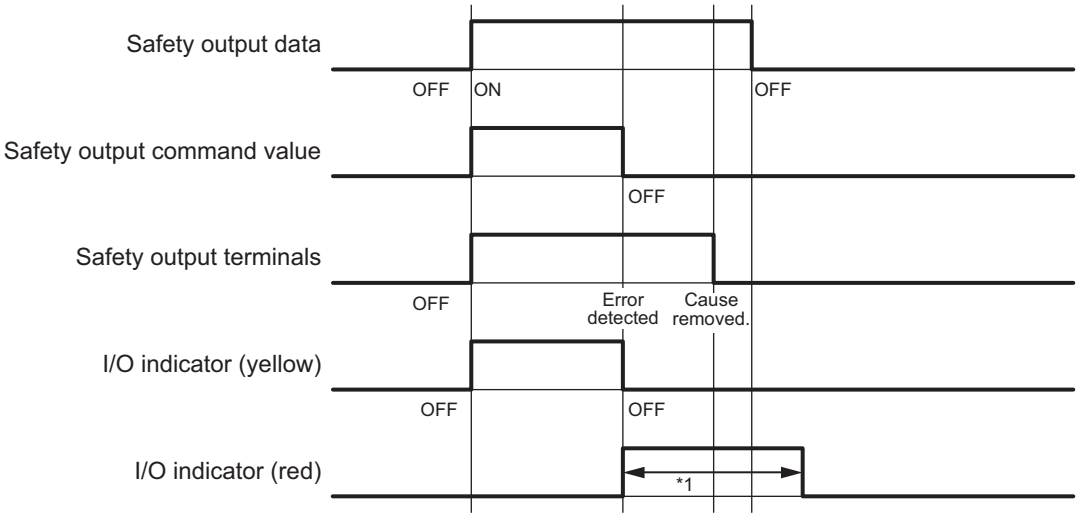
n = Even number

| Single/Dual | Safety output data | Output signal on the safety output terminal | | Meaning of status |
|---|---|---|---|---|
| | So (x) | So (n) | So (n+1) | |
| Dual Channel Equivalent | 0 | 0 (OFF) | 0 (OFF) | Inactive (OFF) |
| | 1 | 1 (ON) | 1 (ON) | Active (ON) |

- Operation for Single Channel: Normal Operation

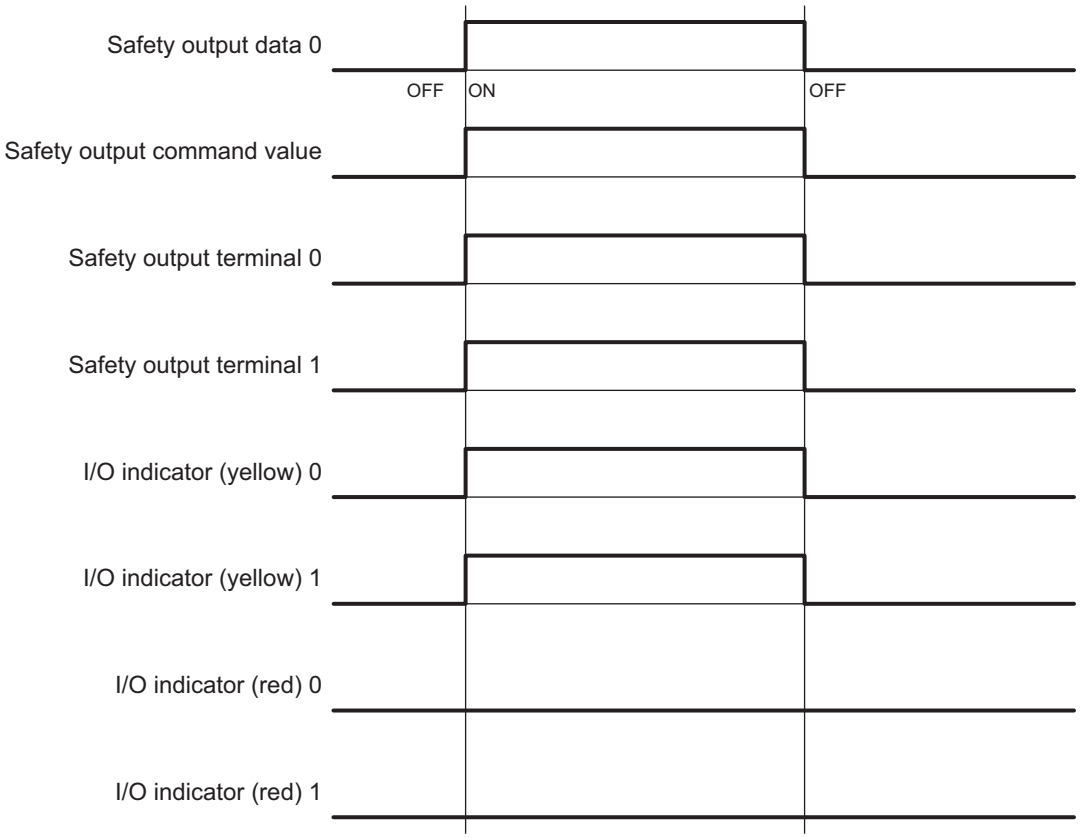



- Operation for Single Channel: Test Pulse Evaluation Error by Stuck-at-high Error

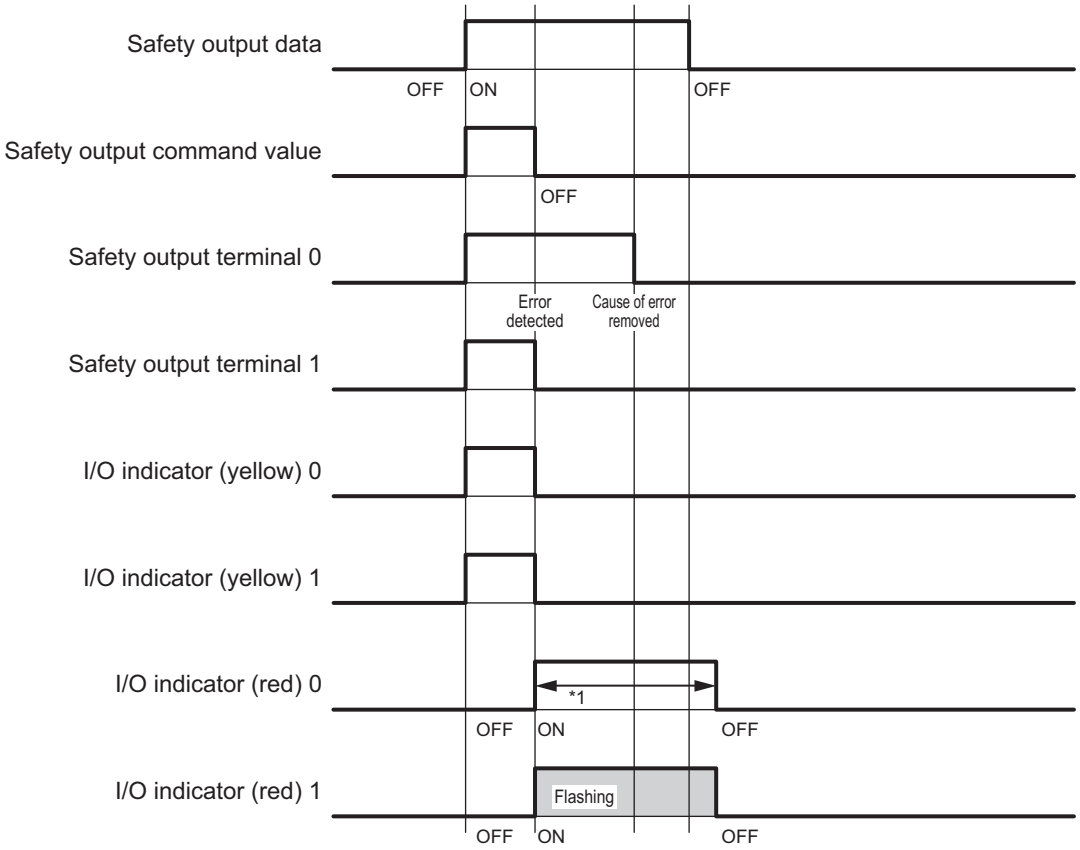



*1. This is the time that the error status (control data, status data, and indicator status) is held (1 s min.).

- Operation for Dual-channel Equivalent Outputs: Normal Operation




- Operation for Dual-channel Equivalent Outputs: Test Pulse Evaluation Error




*1. This is the time that the error status (control data, status data, and indicator status) is held (1 s min.).

## Errors Detected during Self-diagnosis

The errors that can be detected for safety output terminals are determined by the parameter settings. The following table gives the errors that are detected for each parameter setting.

| Test pulse diagnosis | Description of operation | Error detection | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | | Contact with positive side of power line | | Ground fault*1 | | Short circuits in output wiring | | |
| | | Output ON | Output OFF | Output ON | Output OFF | Short circuit when both outputs are ON | Short circuit when both outputs are OFF | Short circuit when one output is ON and the other is OFF |
| Without Test Pulse | Test pulses are not output when the output is ON. | Not detectable. | Detectable. | Detectable. | Not detectable. | Not detectable. | Not detectable. | Detectable. |
| With Test Pulse | Test pulses are output when the output is ON. | Detectable. | Detectable. | Detectable. | Not detectable. | Detectable. | Not detectable. | Detectable. |

*1. To detect ground faults, the 0-V line of the external power supply must be grounded.

## Safety Output Terminal Short Detection

The safety output terminal short detection prevents the internal circuits of the safety output terminals from being destroyed if an overcurrent flows due to a ground fault or other cause.
If an overcurrent is detected, the safety output terminal is turned OFF.
At this time, a *Short Circuit Detected at Safety Output* event will occur. To troubleshoot errors, refer to *Section 12 Troubleshooting* on page 12-1.

## I/O Power Supply Monitoring

I/O power supply monitoring monitors the voltage range of the I/O power supply to ensure the operation of the internal circuits.
If a voltage that is outside of the specified range is detected, all safety output terminals for the Unit are turned OFF.
At this time, an *I/O Power Supply Voltage Error* occurs. To troubleshoot errors, refer to *Section 12 Troubleshooting* on page 12-1.
This function does not work if all of the terminals are set as unused terminals.

## I/O Power Supply OFF Circuit Test (Internal Circuit Diagnosis)

The I/O power supply OFF circuit test shuts OFF the I/O power supply and diagnoses the function that shuts OFF the I/O power supply to see if it is operating correctly.

If an internal circuit fault is detected by this function, all safety output terminals for the Unit are turned OFF.
This test is executed at the following two times. The I/O power supply OFF circuit test is executed only once when the Unit power supply is turned ON. The test is not performed again until the Unit power supply is turned ON again.

### ● When the Status Changes to Refreshing Status

If the I/O power supply is turned ON before the status changes to refreshing status, the I/O power supply OFF circuit test is performed when the status changes to refreshing status.

### ● When I/O Power Supply is Turned ON

If the I/O power supply is turned ON after the status changes to refreshing status, the I/O power supply OFF circuit test is performed when the I/O power supply is turned ON.

# 5

# System Configuration and Setup

This section describes how to use the Sysmac Studio to configure and set up the safety control system.



**5-1 Configuration and Setup Procedures ..... 5-2**

**5-2 Part Names and Functions of the Sysmac Studio Window ..... 5-3**

**5-3 Controller Configuration and Setup of the Safety Control Units ..... 5-5**
5-3-1 Procedures for Creating the Controller Configuration for Safety Control ..... 5-6
5-3-2 Setting and Viewing the Safety Control Unit Settings ..... 5-9
5-3-3 Procedure to Change the Model of the Safety Control Unit ..... 5-10

**5-4 Setting Up the Safety Process Data Communications ..... 5-12**
5-4-1 Setting Up the FSoE Communications ..... 5-12
5-4-2 Setting Up the CIP Safety Communications ..... 5-13
5-4-3 Calculating the Number of Connections ..... 5-37

**5-5 Safety I/O Functions ..... 5-40**

**5-6 Registering Device Variables ..... 5-43**

**5-7 Exposing Variables to Standard Controllers ..... 5-49**
5-7-1 Exposing Global Variables ..... 5-49
5-7-2 Setting Exposed Variables ..... 5-50
5-7-3 Safety CPU Unit Status ..... 5-54
5-7-4 I/O Ports for Safety I/O Units ..... 5-55
5-7-5 I/O Refreshing Method ..... 5-55

**5-8 Setting Standard Process Data Communications ..... 5-57**
5-8-1 Using an CPU Unit or EtherCAT Coupler Unit ..... 5-57
5-8-2 Using an EtherNet/IP Coupler Unit ..... 5-57

**5-9 Exporting/Importing Settings Data ..... 5-59**
5-9-1 Exporting/Importing the Settings for the Entire Slave Terminal ..... 5-59
5-9-2 Exporting/Importing the Settings for All the NX Units on a CPU Rack or Slave Terminal ..... 5-61
5-9-3 Exporting/Importing Data for Individual Safety CPU Unit ..... 5-62
5-9-4 Importing the Safety Unit Restore File ..... 5-63

**5-10 Offline Comparison ..... 5-65**
5-10-1 Procedure for Offline Comparison ..... 5-65
5-10-2 Checking the Comparison Results ..... 5-66
5-10-3 Detailed Comparison ..... 5-67
5-10-4 Target Data of Offline Comparison ..... 5-69

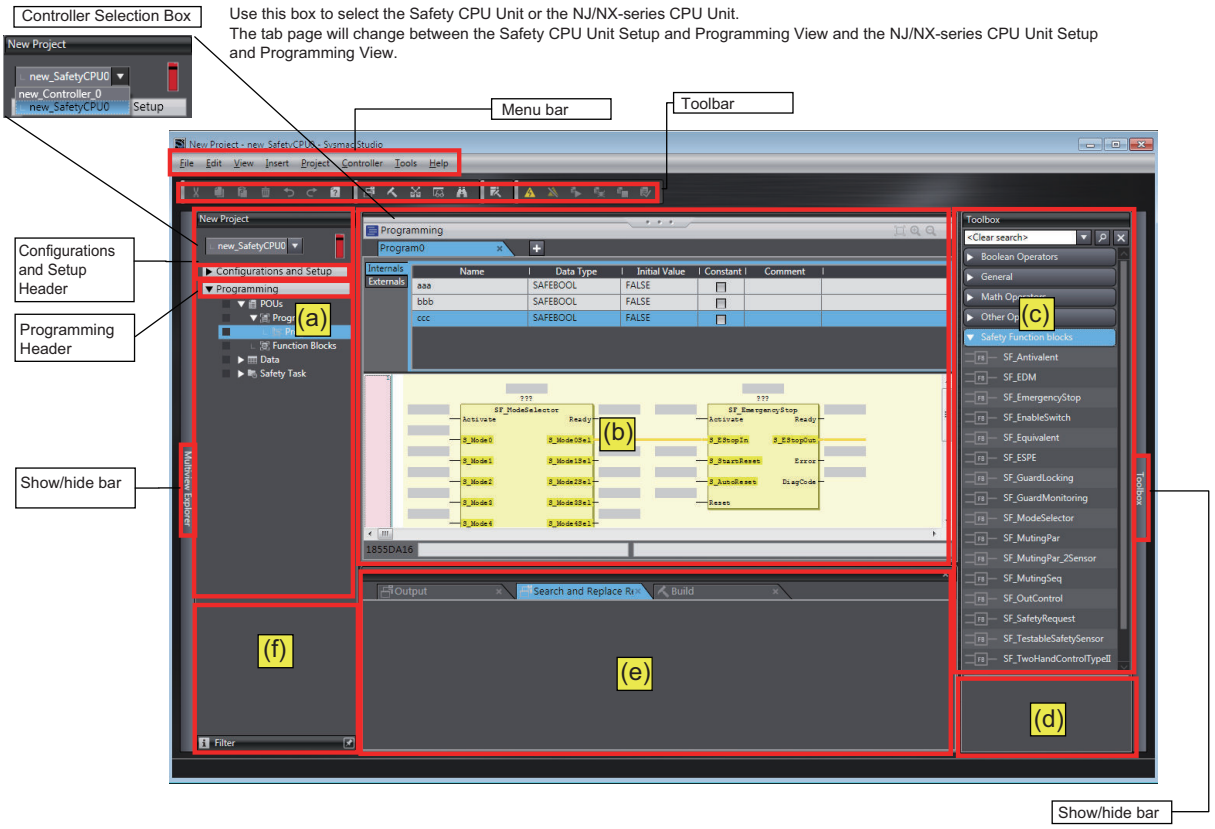
# 5-1 Configuration and Setup Procedures

This section describes the procedures for using the Sysmac Studio to configure and set up the safety control system. Make the settings in the following order.

1. Configure and set up the safety control system.
2. Set up safety process data communications.
3. Select the connected input devices and output devices to set up the input terminals and output terminals.
4. Register the device variables
5. Expose the variables to the standard controller.

# 5-2 Part Names and Functions of the Sysmac Studio Window

This section gives the names of the parts of the Sysmac Studio Window.




| Item | Name | Function |
|---|---|---|
| (a) | Multiview Explorer | This pane is your access point for all Sysmac Studio data that is related to the Safety CPU Unit.<br>It has a **Controller Selection** Box, and is separated into a **Configurations and Setup** Layer and a **Programming** Layer.<br>Use the **Controller Selection** Box to select the Safety CPU Unit or NJ/NX-series CPU Unit. |
| (b) | Edit Pane | The Edit Pane is used to display and edit the data for any of the items. |
| (c) | Toolbox | The Toolbox shows the objects that you can use to edit the data that is displayed in the Edit Pane. |
| | Search and Replace Pane | In this pane, you can search for and replace strings in the data in the **Programming** Layer of the Multiview Explorer. |
| (d) | Controller Status Pane | This pane shows the operating status of the Safety CPU Unit. The Controller Status Pane is displayed only while the Sysmac Studio is online with the Safety CPU Unit or when the Simulator is running. |
| | Simulation Pane | This pane is used to start and stop the Safety CPU Unit Simulator. |

| Item | Name | Function |
|---|---|---|
| (e) | Output Tab Page | The Output Tab Page shows the results of building. |
| | Watch Tab Page | The Watch Tab Page shows the monitor results of the Simulator or the online Safety CPU Unit. |
| | Build Tab Page | The Build Tab Page shows the results of program checks and building. |
| | Search and Replace Results Tab Page | The Search and Replace Results Tab Page shows the results when Search All or Replace All is executed. |
| (f) | Filter Pane | The Filter Pane allows you to search for color codes and for items with an error icon. The results are displayed in a list. |

This manual describes only the functions and operations of the Sysmac Studio that are related to the safety control system.
Refer to the *Sysmac Studio Version 1 Operation Manual (Cat. No. W504)* for details on Sysmac Studio operation.

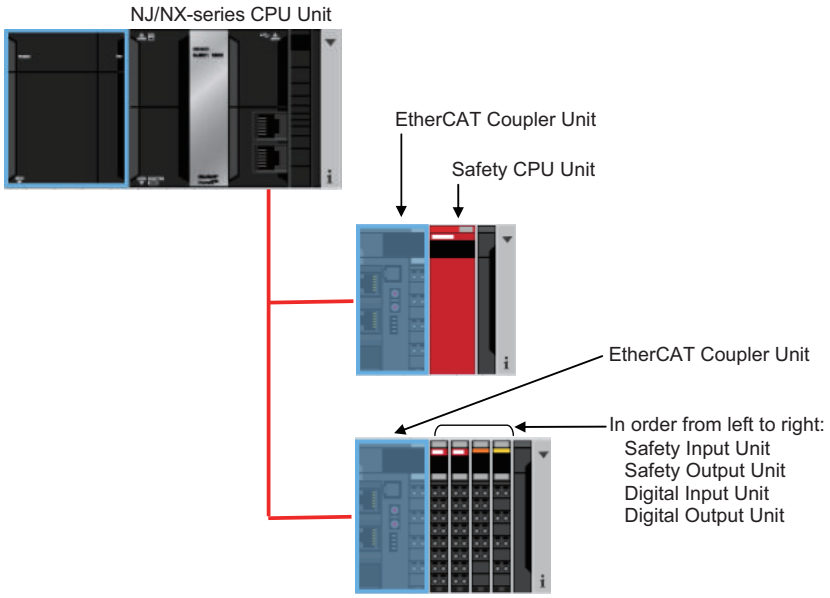
# 5-3 Controller Configuration and Setup of the Safety Control Units

This section describes the Safety Control Unit configuration and setting procedures for the CPU Rack of an NX-series CPU Unit, EtherCAT Slave Terminals and EtherNet/IP Slave Terminals.

## ● CPU Rack or EtherCAT Slave Terminal Configuration and Setting Procedures

You configure and set up the CPU rack, EtherCAT network and EtherCAT Slave Terminals where the Safety Control Units are mounted as part of the Controller Configuration and Setup of the NJ/NX-series CPU Unit.

This section describes the operations to perform based on the following configuration.
Configuration Example:

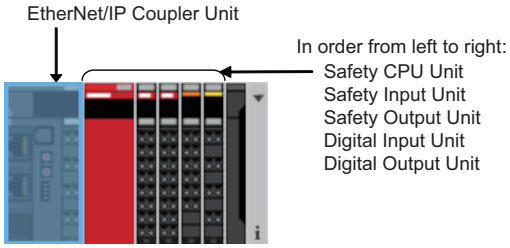



Refer to the *Sysmac Studio Version 1 Operation Manual (Cat. No. W504)* for details on configuring and setting up the NJ/NX-series CPU Unit.

## ● EtherNet/IP Slave Terminal Configuration and Setting Procedures

Set up the Slave Terminal (create the configuration and set the parameters) on the Edit EtherNet/IP Slave Terminal Configuration Tab Page.

This section describes the operations to perform based on the following configuration.
Configuration Example:



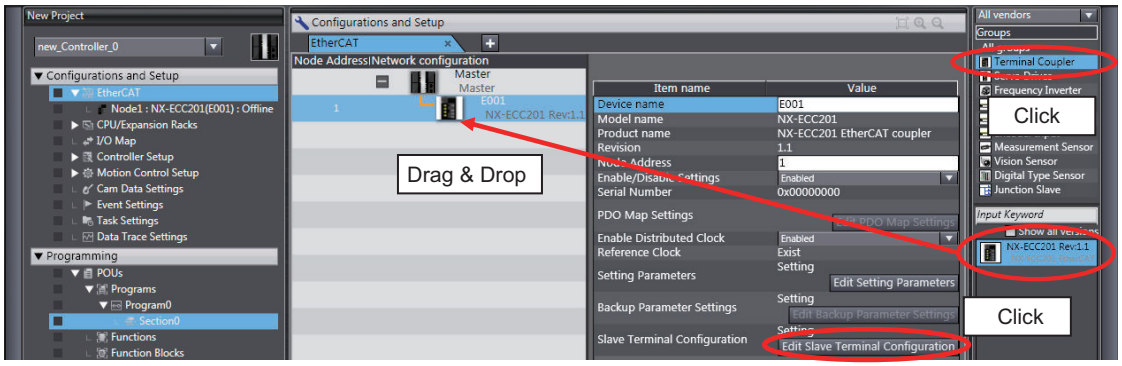

## 5-3-1 Procedures for Creating the Controller Configuration for Safety Control

Use the following procedure to create the Controller configuration for the Safety Control Unit.

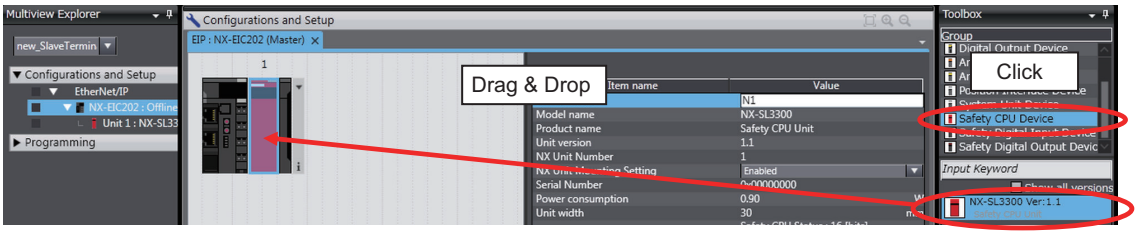
### Procedures for Creating the Controller Configuration for Safety Control

Use the following procedure to create the Controller configuration for the Safety Control Units on EtherCAT Slave Terminals.

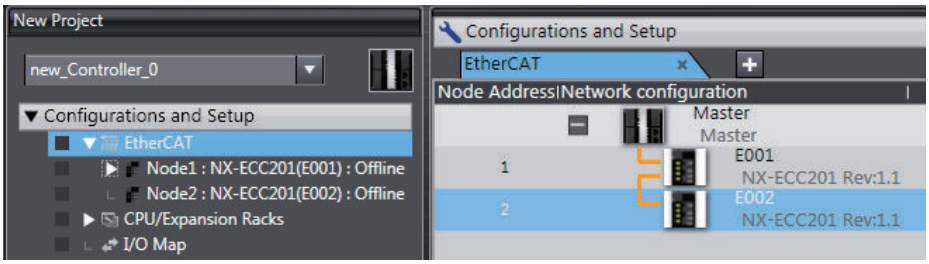
1. Start the Sysmac Studio.
2. Select the model and version of the NJ/NX-series CPU Unit, and create a project file.
3. Double-click **EtherCAT** under **Configurations and Setup**in the Multiview Explorer.
4. Select **Terminal Coupler** from the Groups List in the Toolbox. The EtherCAT Coupler Unit is displayed below it. Drag the Unit and add it to the configuration.




5. Click the **Edit Slave Terminal Configuration** Button at the bottom of the list of EtherCAT Coupler Unit settings. Or, right-click the EtherCAT Coupler Unit and select **Edit Slave Terminal Configuration** from the menu.
   The Slave Terminal Tab Page is displayed.
6. Select **Safety CPU Device** from the Groups List in the Toolbox. The Safety CPU Unit is displayed below it. Drag the Safety CPU Unit to the Slave Terminal and add it to the configuration.

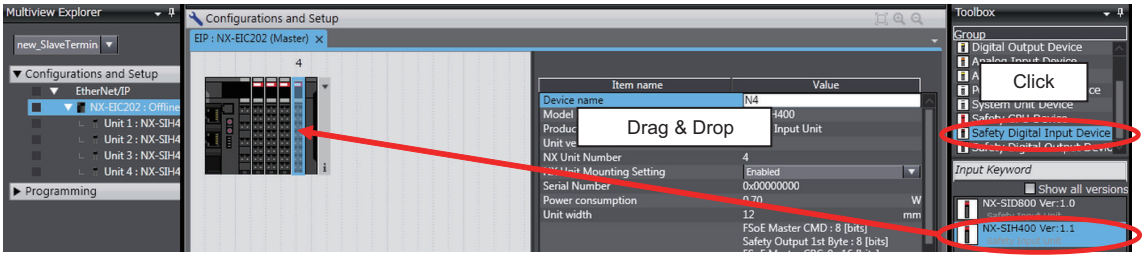



**7** Perform steps 3 and 4 to add another EtherCAT Coupler Unit.



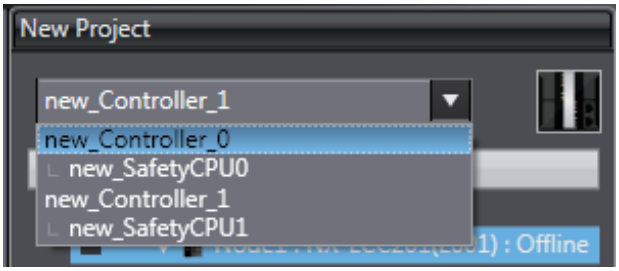

**8** Or, perform step 5 for the EtherCAT Coupler Unit that was added to display the Slave Terminal Tab Page.

**9** Select **Safety Digital Input Device** or **Safety Digital Output Device** from the Groups List in the Toolbox. The Safety I/O Unit is displayed below it. Drag the model of the Safety I/O Unit to the Slave Terminal and add it to the configuration.




This completes the creation of the Controller configuration for an NJ/NX-series CPU Unit that includes Safety Control Units.

After the Safety CPU Unit is added to the NJ/NX-series Controller configuration, the Safety CPU Unit will be displayed in the Controller Selection Box in the Multiview Explorer. The Safety CPU Unit that was added is displayed below the NJ/NX-series Controller (i.e., the EtherCAT master).



**Additional Information**

- You can place only one Safety CPU Unit on a CPU Rack or in the EtherCAT network. If you add more than one Safety CPU Unit, the exclamation icon is displayed under all of the Safety CPU Units, and it will cause an error during the synchronization and download processes.
- Refer to the *NX-series EtherCAT Coupler Unit User's Manual (Cat. No. W519-E1-02 or later)* for the number of NX Units that can be mounted to a Slave Terminal.
- Use the Multiview Explorer to move NX Units between Slave Terminals.

## EtherNet/IP Slave Terminal Configuration and Setting Procedures

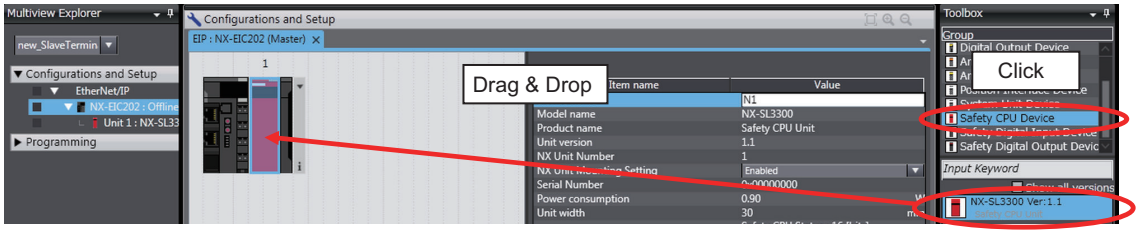
Use the following procedure to create the Controller configuration for the Safety Control Unit on EtherNet/IP Slave Terminals.

**1** Start the Sysmac Studio.

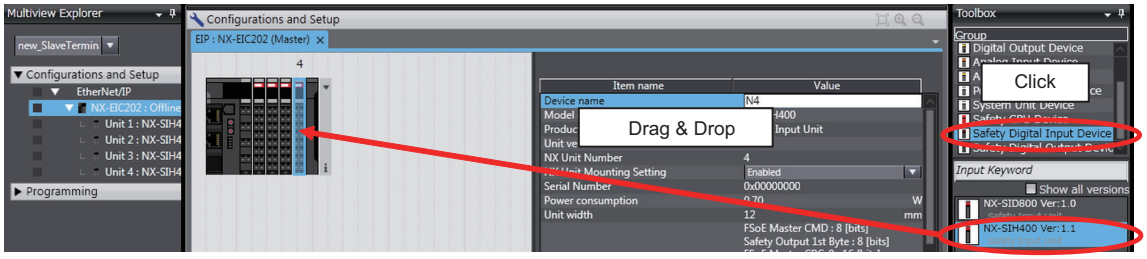
**2** Select the EtherNet/IP Coupler Unit on the Slave Terminal and create a project file.

**3** Double-click **NX-EIC202** under **EtherNet/IP** – **Configurations and Setup** in the Multiview Explorer. The Slave Terminal Tab Page is displayed.

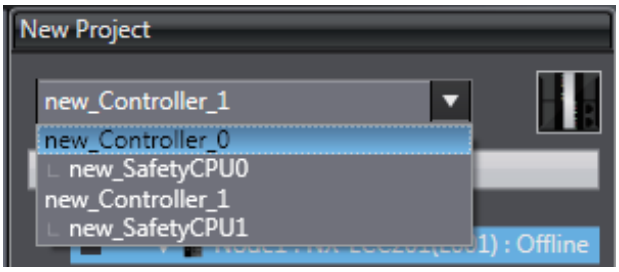
**4** Select **Safety CPU Device** from the Groups List in the Toolbox. The Safety CPU Unit is displayed below it. Drag the Safety CPU Unit to the Slave Terminal and add it to the configuration.




**5** Select **Safety Digital Input Device** or **Safety Digital Output Device** from the Groups List in the Toolbox. The Safety I/O Unit is displayed below it. Drag the Safety I/O Unit to the Slave Terminal and place it in the configuration.




This completes the creation of the EtherNet/IP Slave Terminal configuration for Safety Control Units. After the Safety CPU Unit is added to the configuration, this Safety CPU Unit will be displayed in the Controller Selection Box in the Multiview Explorer. The Safety CPU Unit that was added is displayed below the Coupler Unit.




**Additional Information**

- You can place only one Safety CPU Unit on an EtherNet/IP Slave Terminal. If you add more than one Safety CPU Unit, the exclamation icon is displayed under all of the Safety CPU Units, and it will cause an error during the synchronization and download processes.
- Refer to the *NX-series EtherNet/IP Coupler Unit User's Manual (Cat. No. W536)* for the number of NX Units that can be mounted to a Slave Terminal.

## 5-3-2 Setting and Viewing the Safety Control Unit Settings

Set or view the settings for the Safety CPU Unit and Safety I/O Units (hereinafter, Safety Control Unit) as necessary.

You can change the device names of registered Safety Control Unit, and enable or disable individual Units as NX Units.

| Item | Editing | Description | Data range | Default |
|---|---|---|---|---|
| Device name | Possible. | This is the name of the Safety Control Unit.<br>The device name is automatically assigned when you register the Safety CPU Unit or Safety I/O Units. You can change the device name if necessary.<br>Device names must be unique within the same safety control configuration. If you specify the same name for more than one Unit, an error will occur. | Text string | N* (where * is a serial number from 1) |
| Model name | Not possible. | Model of Safety Control Unit. | --- | --- |
| Product name | Not possible. | This is the product name. | --- | --- |
| Unit version | Not possible. | Unit version of the Safety Control Unit. | --- | --- |
| NX Unit number | Not possible. | This number represents the logical position of the Safety Control Unit as an NX Unit. Numbers are automatically assigned from the leftmost mounting position. | --- | --- |
| NX Unit mounting setting | Possible. | This setting enables or disables I/O refreshing for the Safety Control Unit.<br>Refer to the *NX-series EtherCAT Coupler Unit User's Manual (Cat. No. W519)* for details on this function. | Enabled or disabled | Enabled |
| Serial number | Not possible. | Serial number of the Safety Control Unit. You can get the serial number to set the serial number of the actual EtherCAT Coupler Unit. | --- | 00000000 hex |

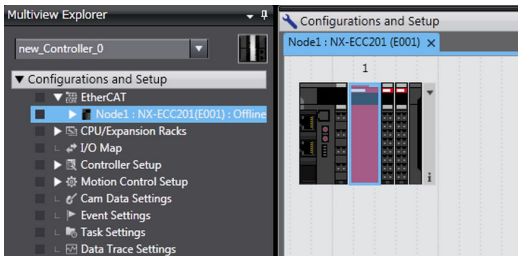
| Item | Editing | Description | Data range | Default |
|---|---|---|---|---|
| Power consumption [W] | Not possible. | The power consumption of the Safety Control Unit from the NX bus.<br>This setting applies to Units other than an Additional NX Unit Power Supply Unit. | --- | Depends on the model of the Unit. |
| Unit width [mm] | Not possible. | Width of the Safety Control Unit. | --- | Depends on the model of the Unit. |
| I/O allocation settings | Not possible. | I/O allocation settings for the Safety Control Unit. | --- | |
| Unit operation settings | Not possible. | There are no settings that you can edit for the Safety Control Unit | --- | |

## 5-3-3 Procedure to Change the Model of the Safety Control Unit

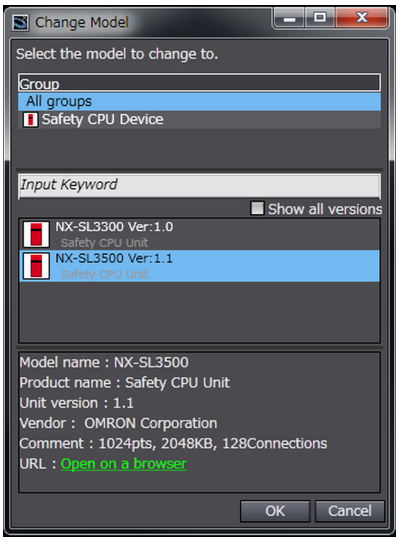
After you create the Controller configuration, you can change the model or unit version of the Safety Control Unit without changing the saved settings information.

Use the following procedure to change the Safety CPU Unit. You can also change the model of a Safety I/O Unit with essentially the same procedure.

***1*** Display the Slave Terminal Tab Page for the CPU Rack or Slave Terminal where the Safety CPU Unit to change is located.




***2*** Right-click the Safety CPU Unit to change and select **Change Model**.
The Change Model Dialog Box is displayed.




**3** Select the model or unit version to change to and then click the **OK** Button.

### Precautions for Correct Use

- Safety validation is not executed when you change the model of a Safety Control Unit. Always execute safety validation again after you change the model.
- If you change the model of a Safety Control Unit, recalculate the safety reaction times. The safety reaction times may change if you change the model.

# 5-4 Setting Up the Safety Process Data Communications

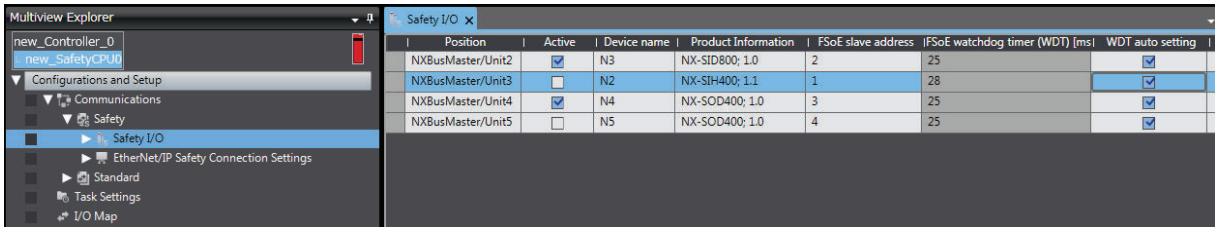
## 5-4-1 Setting Up the FSoE Communications

When the Safety Control Unit is placed on the NX bus in the Sysmac Studio CPU rack configuration, EtherCAT network configuration, and EtherCAT slave terminal configuration, the FSoE communications are set up automatically.

Use the following procedure to view or change the settings for the FSoE communications.

1 In the Multiview Explorer, select the target Safety CPU Unit in the Controller Selection Box.

2 Select **Configurations and Setup** – **Communications** – **Safety**, and then double-click **Safety I/O**.




The meanings of the items in the Safety I/O Tab Page are given below.

| Item | Editing | Description |
|---|---|---|
| Position | Not possible | • Node address<br>Fixed to NXBusMaster when connected to the CPU rack.<br>For a connection to an EtherCAT Coupler Unit, the node number of the Unit as an EtherCAT slave is displayed.<br>"EIP" is displayed for a connection to an EtherNet/IP Coupler Unit.<br>• Unit number<br>It displays the NX Unit number of the Safety I/O Unit. |
| Active (Enable/Disable) | Possible | The communications setting for FSoE communications is displayed.<br>Selected: *Enabled*<br>This setting assigns the Safety I/O Unit to the Safety CPU Unit as a communications node, and displays the I/O ports for that Unit in the I/O Map.<br>Not selected: *Disabled*<br>This setting does not assign the Safety I/O Unit to the Safety CPU Unit as a communications node, and does not display the I/O ports for that Unit in the I/O Map. |
| Device | Not possible | This is the name of the Safety Control Unit. |
| Product Information | Not possible | This is the model and version of the Unit. |
| FSoE Slave Address | Possible | When the Active setting described above is set to **Enabled**, the **FSoE Slave Address** is automatically set as an internal address for use with FSoE communications.<br>You can change the FSoE slave address. The setting range is from 1 to 65,535.<br>Set a unique FSoE slave address for each FSoE slave in the FSoE communications.<br>If the same FSoE slave address is set more than once, an error is displayed on the Sysmac Studio. |

| Item | Editing | Description |
|---|---|---|
| FSoE Watchdog Timer*1 | Possible | This is the setting of the timeout value for FSoE communications between the Safety CPU Unit and the Safety I/O Unit.<br>The setting range is from the lowest value of the FSoE watchdog timers to 65,535 ms. |
| WDT auto setting*1 | Possible | This setting specifies the setting method for the FSoE watchdog timer (WDT).<br>Cleared: You can set the FSoE watchdog timer (WDT) to the desired value. |

*1. Refer to *Section 8 Calculating Safety Reaction Times* on page 8-1 for information on setting the FSoE watchdog timers and the WDT auto setting.

**Precautions for Safe Use**

The relevant Units will maintain the safe states for I/O data with FSoE connections after an error is detected in FSoE communications. However, when the cause of the error is removed, FSoE communications will recover automatically.
If you need to prevent equipment from restarting when FSoE communications recover automatically, implement suitable restart conditions in the user program.

**Version Information**

- You can change the FSoE watchdog timers and the WDT auto setting with Sysmac Studio version 1.10 or higher.
- You can change the FSoE slave addresses with Sysmac Studio version 1.11 or higher.

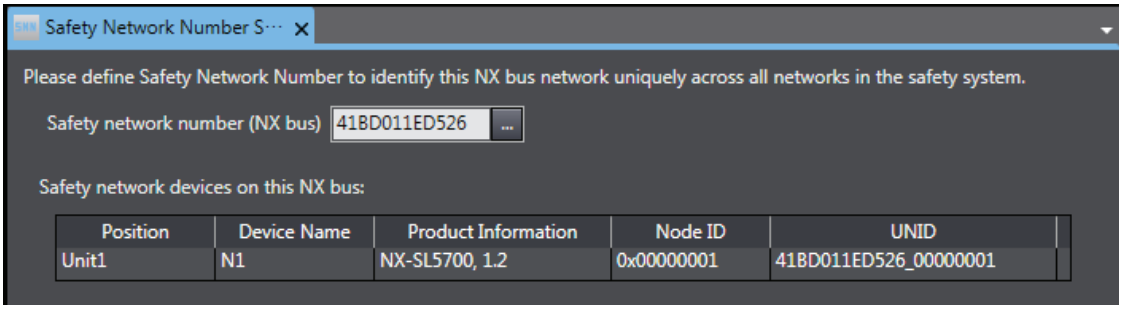
## 5-4-2 Setting Up the CIP Safety Communications

### Safety Network Number Settings for the NX Bus

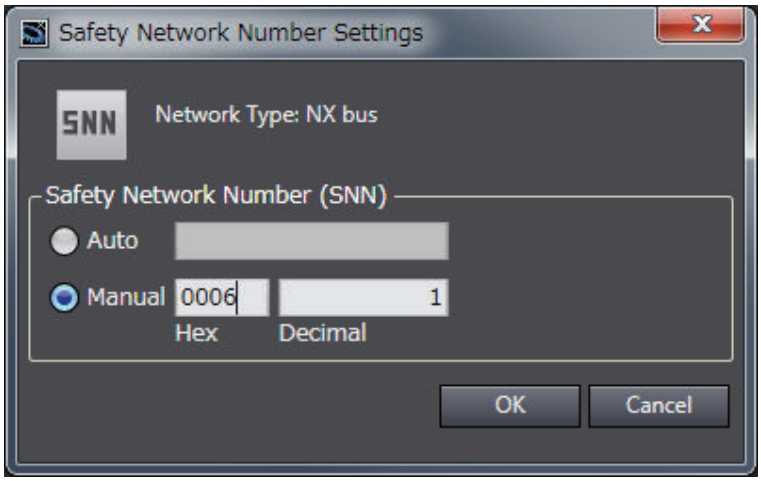
Use the following procedure to set the Safety Network Number (SNN) for the NX bus.

**1** In the Multiview Explorer, select the target Safety CPU Unit in the Controller Selection Box.

**2** Select **Configurations and Setup** – **Communications** – **Safety** – **EtherNet/IP Safety Connection Settings** and double-click **Safety Network Number Settings**.
The Safety Network Number Settings Tab Page is displayed.




**3** Click the button located on the right side of Safety network number (NX bus).
The Safety Network Number Settings Dialog Box is displayed.




When you click the **Auto** Option, the Safety Network Number (SNN) is automatically generated from the current time information of the computer on which the Sysmac Studio is running.
When you click the **Manual** Option, enter a unique number that is not used in any other system.

***4*** Click the OK Button.
The Safety Network Number (SNN) is determined, and UNID is updated.

**Additional Information**

You can also enter the Safety Network Number (SNN) directly in the Safety Network Number Settings Tab Page.

## Originator Connection Settings

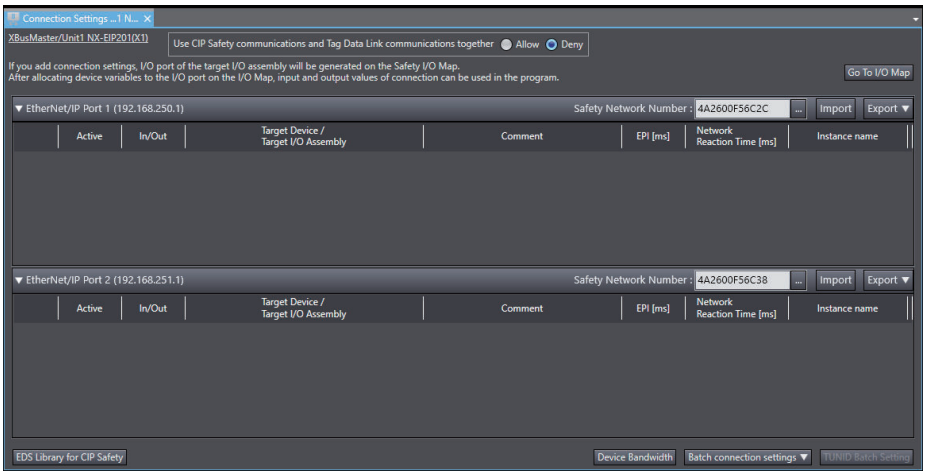
To use the Safety CPU Unit as an originator device for CIP Safety, the CIP Safety communications need to be set. Use the following procedure to set the CIP Safety communications.

### ● Adding Target Devices

Use the following procedure to add a target device to open a CIP Safety connection.

***1*** In the Multiview Explorer, select the target Safety CPU Unit in the Controller Selection Box.

***2*** Select **Configurations and Setup** – **Communications** – **Safety** – **EtherNet/IP Safety Connection Settings** and double-click **Connection Settings (Originator)**.
Connection Settings (Originator) Tab Page is displayed.
A list of target devices that can be set for the connection is shown in the Toolbox.




The Connection Settings (Originator) Tab Page consists of the items shown in the following table.

| Item | Description |
|---|---|
| Use CIP Safety communications and Tag Data Link communications together*1 | Select whether the CIP Settings on the NX-series EtherNet/IP Unit allow CIP Safety communications and tag data link communications to be used together.<br>Allow: Allows CIP Safety communications and tag data link communications to be used together<br>Deny: Do not allow CIP Safety communications and tag data link communications to be used together |
| EtherNet/IP Port 1 | Settings of CIP Safety connections via the EtherNet/IP port 1 are listed.<br>Safety Network Number: Specifies the SNN for the EtherNet/IP network for which the port 1 is connected<br>Connection list: Connections grouped together by each target device are listed. |
| EtherNet/IP Port 2 | Settings of CIP Safety connections via the EtherNet/IP port 2 are listed.<br>Safety Network Number: Specifies the SNN for the EtherNet/IP network for which the port 2 is connected<br>Connection list: Connections grouped together by each target device are listed. |
| Target Device | IP address, Unit number, and device name are displayed for the target device with which a CIP Safety connection is to be opened. |
| In/Out | Shows the data direction of each connection.<br>  : In - Data to be received from the target device<br>  : Out - Data to be sent to the target device |
| Target I/O Assembly | I/O Assembly name of the target device with which a connection is to be opened is displayed. |
| Active | Enable/disable each connection.<br>Selected: Connection is active.<br>Not selected: Connection is inactive. |
| EPI [ms] | Specifies the Expected Packet Interval (data update cycle) in ms. |
| Network Reaction Time [ms] | Displays the value of the Network Reaction Time in ms. This is used in calculating the safety reaction time. |
| Instance name | Name of the instance being managed in the program. Connection status can be checked if the connection is registered in the Watch Tab Page. |
| **Go To I/O Map** Button | Click this button to open the I/O Map Tab Page. |
| **Device Bandwidth** Button | Click this button to show bandwidth usage of originator connection. |

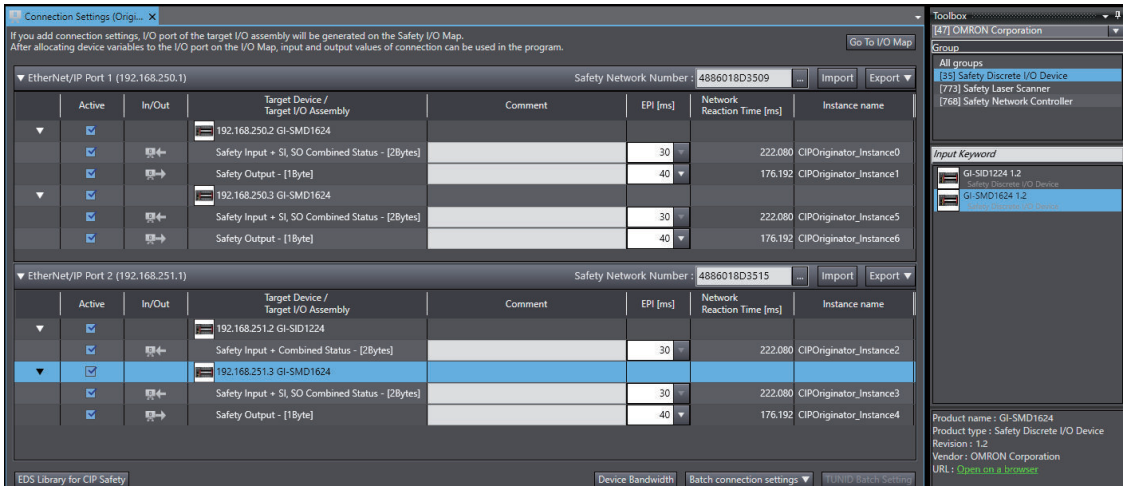
| Item | Description |
|---|---|
| **Import** Button | Click this button to open the Import Dialog Box. |
| **Export** Button | **Output export file** or **Migrate to other EtherNet/IP port** can be selected from the pull-down menu. |
| **EDS Library for CIP Safety** Button | Click this button to display the **EDS Library for CIP Safety** Dialog Box. You can install, uninstall, create, and export EDS files. |

*1. This item is displayed when the NX502 CPU Unit with unit version 1.66 or later and the NX-series EtherNet/IP Unit with unit version 1.01 or later are used, and setting the connection settings of the NX-series EtherNet/IP Unit.

### Additional Information

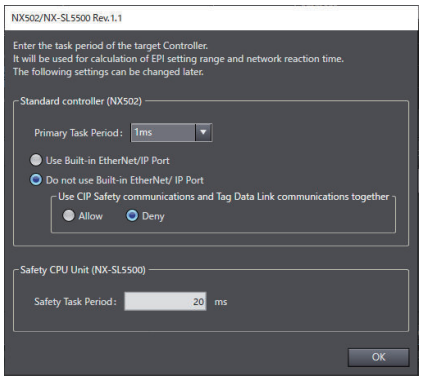
- If more than one CPU Unit or Communication Control Unit is connected to the same EtherNet/IP network via a EtherNet/IP port, the same Safety Network Number should be set for the EtherNet/IP port of each Unit.
- For the NX502 CPU Units, CIP Safety communications via the built-in EtherNet/IP port can be used with the NX502 CPU Unit with unit version 1.64 or later. If you set CIP Safety communications with the NX502 CPU Unit with unit version earlier than 1.64, use an NX-series EtherNet/IP Unit.
- When the NX502 CPU Unit with unit version 1.66 or later is used with the NX-series EtherNet/IP Unit with unit version 1.01 or later, you can use tag data link and CIP Safety communications together in one NX-series EtherNet/IP Unit.
- Since the I/O refresh time of the NX502 CPU Unit changes depending on the increase or decrease of the safety connections for CIP Safety communications, check the effect on the task execution time.

**3** Select a target device to open a connection from the Toolbox. Drag and drop it on the connection list for either Port 1 or Port 2 to add the target device and default connection.



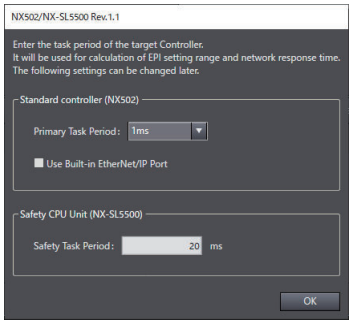

**4** The dialog box is displayed to enter settings to calculate the EPI setting range and network reaction time. Enter the task period of the target controller and other settings. The dialog box and setting items displayed change with the target device and Sysmac Studio version as follows.

- Dialog box displayed when the target devices are the NX502 CPU Unit and Safety CPU Unit and Sysmac Studio version 1.58 or higher is used




| Item | | Description |
|---|---|---|
| Standard controller | Primary task period | Set the task period of the primary periodic task of the CPU Unit. |
| | Use of the built-in EtherNet/IP port | Select whether to use the built-in EtherNet/IP port of the CPU Unit as the communications port.<br><br>Use Built-in EtherNet/IP Port: Select this option to use the built-in EtherNet/IP port on the CPU Unit.<br>Do not use Built-in EtherNet/IP Port: Select this option to use the EtherNet/IP port on the NX-series EtherNet/IP Unit. |
| | Use CIP Safety communications and Tag Data Link communications together | Select whether the CIP Settings on the NX-series EtherNet/IP Unit allow CIP Safety communications and tag data link communications to be used together.<br><br>Allow: Allows CIP Safety communications and tag data link communications to be used together<br>Deny: Do not allow CIP Safety communications and tag data link communications to be used together |
| Safety CPU Unit | Safety task period | Set the safety task period of the Safety CPU Unit. |

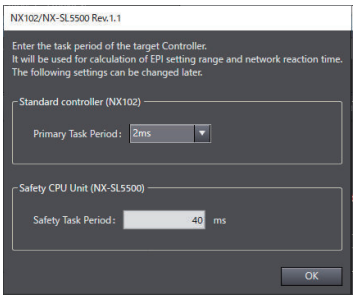
- Dialog box displayed when the target devices are the NX502 CPU Unit and Safety CPU Unit and Sysmac Studio version lower than 1.58 is used




| Item | | Description |
|---|---|---|
| Standard controller | Primary task period | Set the task period of the primary periodic task of the CPU Unit. |
| | Use Built-in EtherNet/IP Port | Select this check box to use the built-in EtherNet/IP port of the CPU Unit as the communications port.<br><br>Uncheck this check box to use the EtherNet/IP port of the NX-series EtherNet/IP Unit as the communications port. |

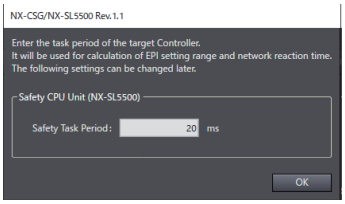
| Item | | Description |
|---|---|---|
| Safety CPU Unit | Safety task period | Set the safety task period of the Safety CPU Unit. |

- Dialog box displayed when the target devices are the NX102 CPU Unit and Safety CPU Unit




| Item | | Description |
|---|---|---|
| Standard controller | Primary task period | Set the task period of the primary periodic task of the CPU Unit. |
| Safety CPU Unit | Safety task period | Set the safety task period of the Safety CPU Unit. |

- Dialog box displayed when the target devices are the Communication Control Unit and Safety CPU Unit



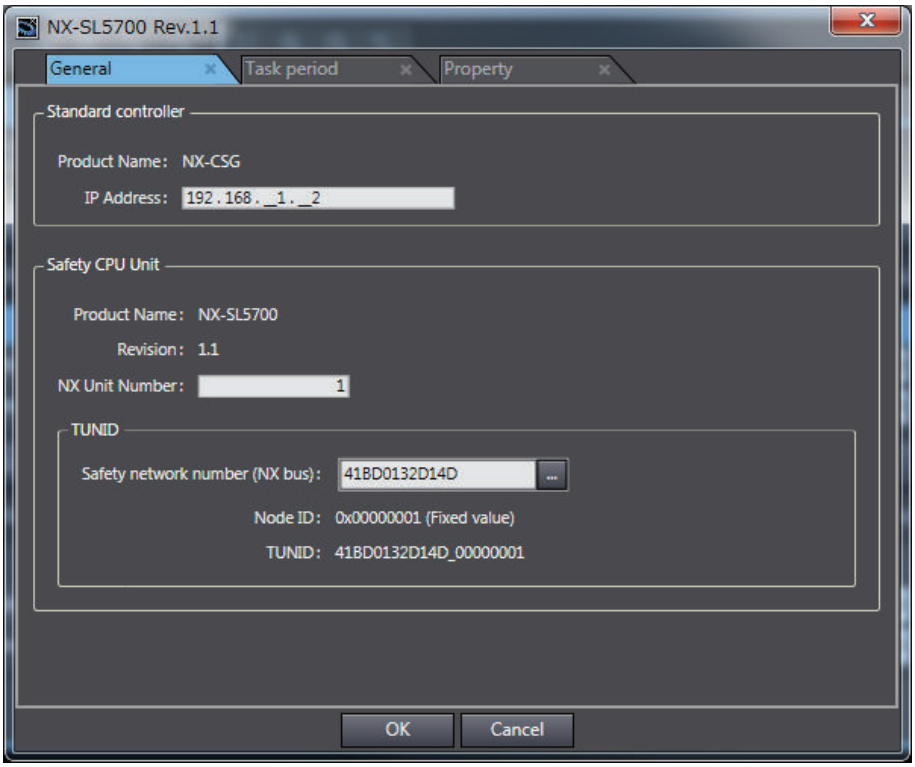

| Item | | Description |
|---|---|---|
| Safety CPU Unit | Safety task period | Set the safety task period of the Safety CPU Unit. |

- When the target devices are different from the above combinations
  The configuration dialog box is not displayed.

## ● Setting the Target Device IP Address

Use the following procedure to set the address of the target device to open CIP Safety connection.

***1*** Select a target device in the connection list and right-click on it. Click the **Edit** Menu.
The target device settings pane is displayed.



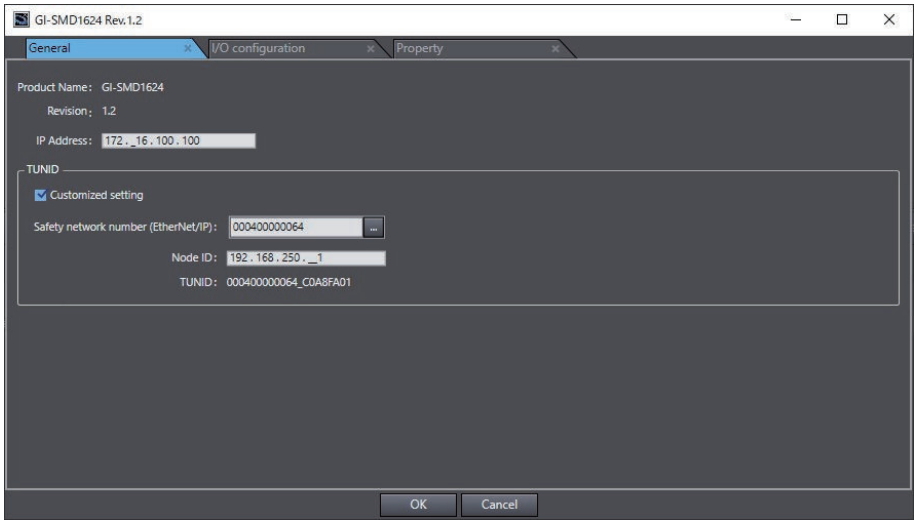

Change the displayed settings according to the actual target device settings.

The settings depend on a type of the target device as shown in the following table.

| Target device type | Setting Items |
|---|---|
| Safety CPU Unit | IP Address: Specify the IP address for the EtherNet/IP port of the CPU Unit, Communication Control Unit, or EtherNet/IP Unit.<br>NX Unit Number: Specify the NX Unit number of the Safety CPU Unit.<br>Safety network number (NX bus): Specify the SNN in the TUNID that is set in the target Safety CPU Unit. |
| Other EtherNet/IP Safety devices | IP Address: Specify the IP address of the target device.<br>TUNID - Customized setting: Select the check box to set a different safety network number or connection destined for Node ID. |

## ● Customized Setting for TUNID

When this setting is selected, you can set a desired TUNID for each target device as follows.

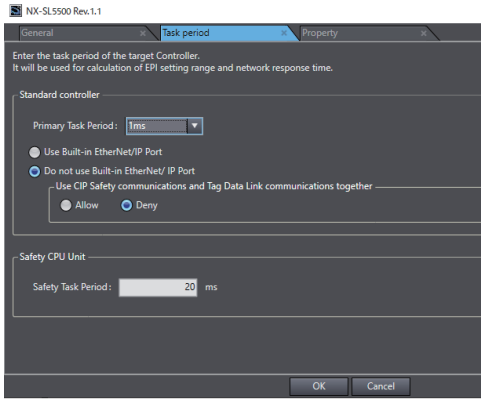


When this setting is used, you can set the connection for CIP Safety even in the following cases.

- When a desired safety network number is specified on the target device side
- When the target device is accessed via a router that uses the NAT function or NAPT function

**Additional Information**

On the **Task period** Tab Page in the target device settings pane, you can set the task period of the target controller. The setting items shown on the tab page are the same as the items on the dialog box to configure the task period, etc in *Adding Target Devices* on page 5-14.
For example, when the target devices are the NX502 CPU Unit and Safety CPU Unit and Sysmac Studio version 1.58 or higher is used, **Task period** Tab Page is displayed as follows.



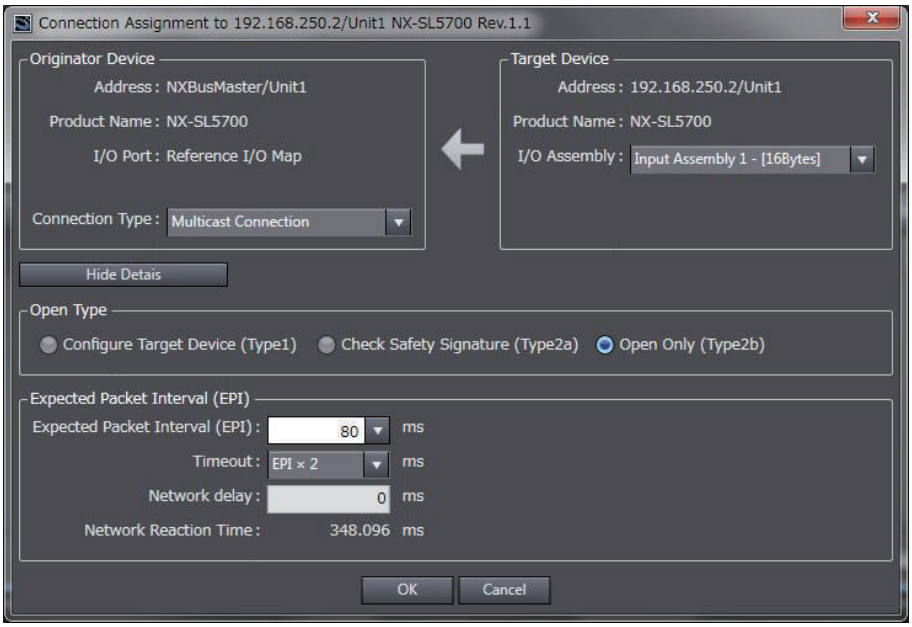

## ● Editing Connection Parameters

Use the following procedure to edit connection parameters for CIP Safety connections.

**⚠ WARNING**

If you select “Open Only” for the Open Type setting, make sure to verify that the originator/target have correct configurations.
Serious injury may possibly occur due to loss of required safety functions.

1 Select a connection for which the parameters need to be edited in the connection list and right-click on it. Click **Edit** Menu.
The Connection Assignment Dialog Box is displayed.




| Item | Description |
|---|---|
| I/O Assembly | I/O assemblies of the target device for which a safety connection can be opened are listed. |
| Connection Type | For input connection (receiving from the target device), select either **Point to Point Connection** or **Multicast Connection**. |
| Open Type | Select a type for connection opening.<br>Configure Target Device: Perform configuration of the target device when a connection is established. For the Safety CPU Unit, this item cannot be selected.<br>Check Safety Signature: Uses the safety signature to check if the target devices are properly configured when establishing the connection. The safety signature consists of the Safety Configuration CRC and the Safety Configuration Time Stamp. When the target device is the Safety CPU Unit, this item cannot be selected.<br>Open Only: Configuration check is not executed when establishing the connection. |
| Expected Packet Interval (EPI) | Set an interval for communications of safety process data between the originator and the target. |
| Timeout | Specify a timeout time using a multiple of the EPI value, allowed for determining a communications error. The default setting is EPI x 2 [ms] (timeout is allowed just once). |
| Network Delay | Set the transmission delay time on the network. The default setting is 0 [ms]. |
| Network Reaction Time | Value of the connection response performance is shown in ms. This is used in calculating safety reaction time. |

***2*** Specify the connection parameter and click the **OK** Button.
A new connection is added to the connection list.

## ● Adding Connections

Use the following procedure to add a CIP Safety connection with the target device.

***1*** Select a target device in the connection list and right-click on it. Click the **Create New Connection** Menu.

The Connection Assignment Dialog Box is displayed.



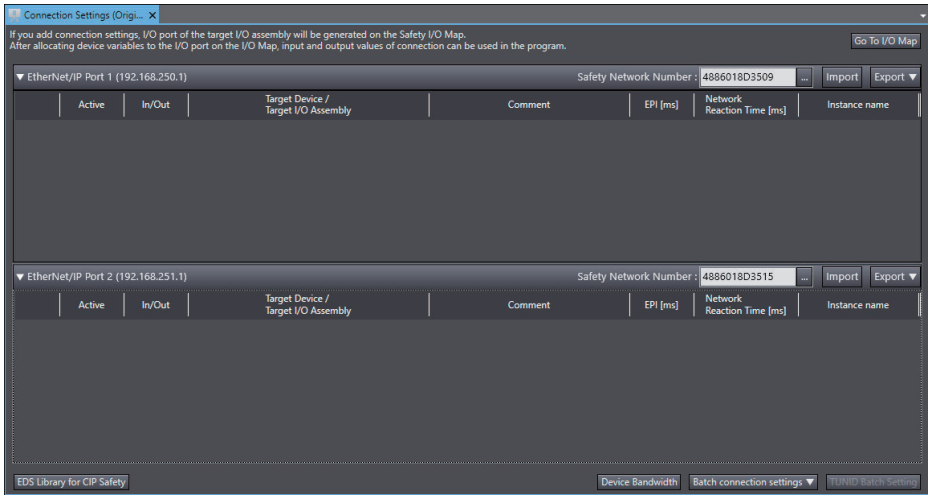

**2** Specify the connection parameters and click the **OK** Button.
A new connection is added to the connection list.

## ● EPI Batch Setting

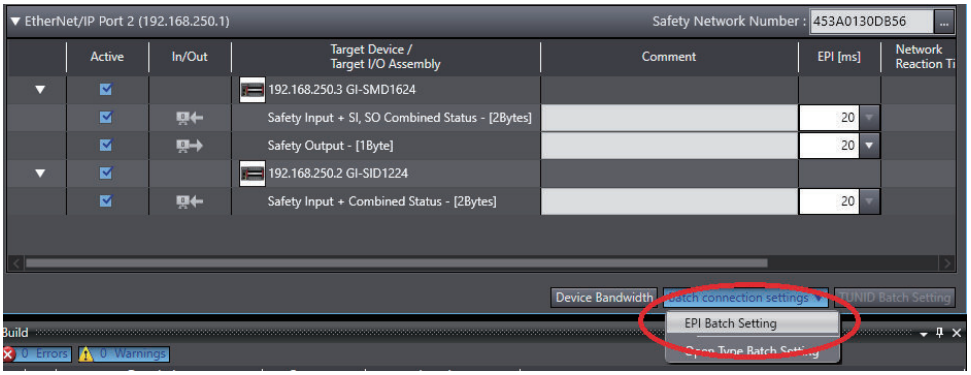
The function of EPI Batch Setting can change the EPI of the CIP Safety connections at once.
Refer to *8-4 EPI (Data Packet Interval)* on page 8-11 for the EPI (Data Packet Interval).
The procedure to use EPI Batch Setting is described below.

**1** In the Multiview Explorer, select the target Safety CPU Unit in the Controller Selection Box.

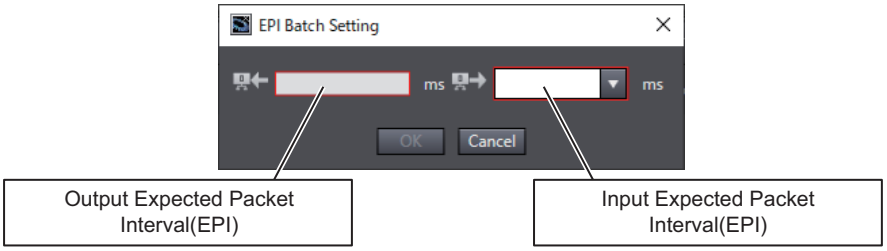
**2** Go to **Configurations and Setup** – **Communications** – **Safety** – **EtherNet/IP Safety Connection Settings** and double-click **Connection Settings (Originator)**.
The following Connection Settings (Originator) Tab Page is displayed.



**3** Click the **Batch connection settings** button to select **EPI Batch Setting**.




The EPI Batch Setting window is displayed.




| Item | Description |
|---|---|
| Input Expected Packet Interval (EPI) | Enter an EPI value for the CIP Safety input connections. |
| Output Expected Packet Interval (EPI) | Enter an EPI value for the CIP Safety output connections. You can select from the EPI values that can be set. |

**4** Click the **OK** button.
The EPI value that you entered or selected will be reflected in each CIP Safety connection setting in the project file.

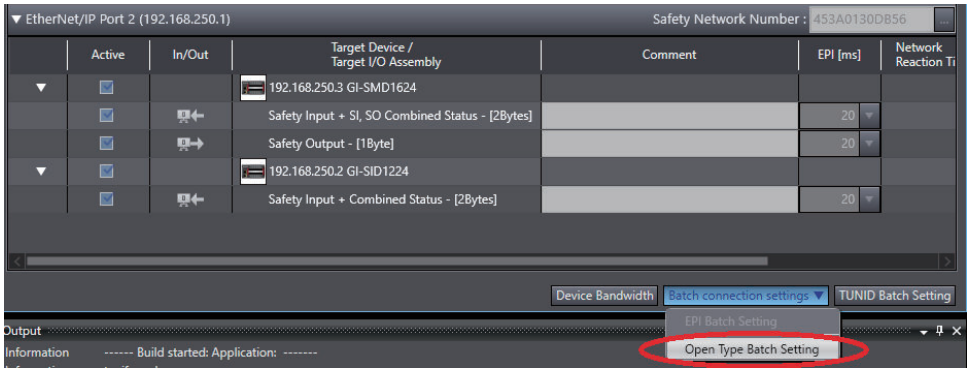
## ● Connection Open Type Batch Setting

The function of Connection Open Type Batch Setting can switch open types of connection parameters of CIP Safety connections at once. In addition, you can obtain safety signatures online from each target device on the network and reflect it in the target device settings.
Refer to *Editing Connection Parameters* on page 5-20 for connection open types.
The procedure to use Open Type Batch Setting is described below.

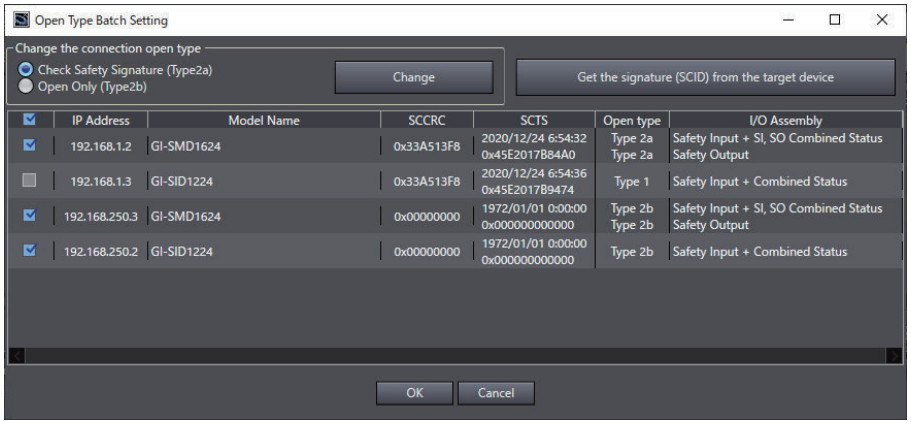
**1** In the Multiview Explorer, select the target Safety CPU Unit in the Controller Selection Box.

**2** Go to **Configurations and Setup** – **Communications** – **Safety** – **EtherNet/IP Safety Connection Settings** and double-click **Connection Settings (Originator)**.
The following Connection Settings (Originator) Tab Page is displayed.

**3** Click the **Batch connection settings** button to select **Open Type Batch Setting**.



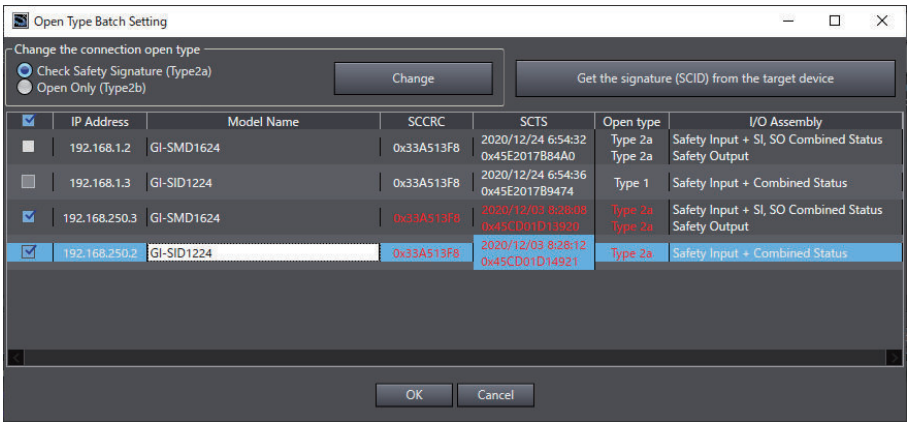

The Open Type Batch Setting window is displayed.




The Open Type Batch Setting window consists of the items shown in the following table.

| Item | Description |
|---|---|
| Target Device Selection Check Box | Select target devices that change the open types or those that obtain and change the signature codes. Unchecked target devices are not be subject to setting changes. |
| Target List | Displays IP addresses, model names, signature codes (SCCRC and SCTS), open types of each connection, and I/O assemblies, of target devices. |
| The Type 2a/Type 2b Selection Button | Select the open type.<br>Type2a: Check Safety Signature<br>Type2b: Open Only |
| The **Change** Button | Reflects the open types selected with the Type 2a/Type 2b buttons in the target list.<br>At that point in time when you click the **Change** button, they are not reflected in the project file. Only after you click the OK button, they will be reflected in the project file. |
| The **Get the signature (SCID) from the target device** Button | When you connect to the Communication Control Unit online, you can obtain the signature codes from the target devices for which Type 2a are set in the open types. |

The text colors will change to red when you change any settings.




**4** Confirm the changed contents and click the **OK** button.
The set open types and safety signatures (SCCRC and SCTS) of the target devices will be reflected in the project file.

### Additional Information

Change the open type to Type2a and then press the **Get the signature (SCID) from the target device** button to obtain safety signatures from the target devices.

## Target Device Operation

This function sets and operates a target device on the network online. The availability of each function depends on the operation specifications of the target device.

| Function | Description | Calling method |
|---|---|---|
| Clear Memory | Clears the settings of target device. | Right-click the device to be set and select**Target Device**–**Clear Memory**. |
| TCP/IP Settings | Sets such as the IP address of the target device. | Right-click the device to be set and select**Target Device**–**TCP/IP Settings**. |
| TUNID Setting (Individually) | Sets TUNID of the target device. | Right-click the device to be set and select**Target Device**–**TUNID Setting**. |
| TUNID Batch Setting | Sets TUNID of multiple target devices at once. | Click the**TUNID Batch Setting**button. |
| Restart | Restarts the target device. | Right-click the device to be set and select**Target Device**–**Restart**. |
| Lock | Locks the configuration for the target device. | Right-click the device to be set and select**Target Device**–**Lock**. |
| Unlock | Unlocks the configuration for the target device. | Right-click the device to be set and select**Target Device**–**Unlock**. |

### Precautions for Correct Use

- This function depends on the operation specifications of the target device. Some operations are not supported depending on the device functions.
- If you set TUNID to the target device, make sure that the correct TUNID is set to the intended device under your responsibility. Also, ensure that all TUNIDs are unique across the system.

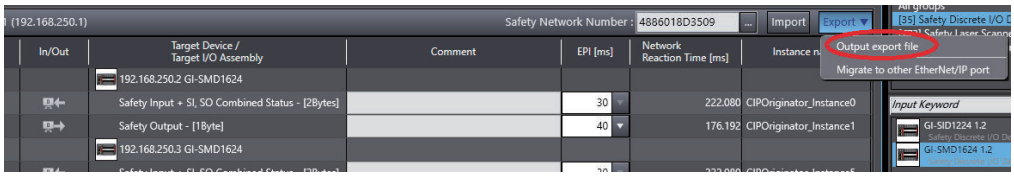
## Batch Export

You can use the batch export function to output CIP Safety connection settings for each EtherNet/IP port to an export file or move the settings to other EtherNet/IP port. The procedure for outputting to an export file is shown below.

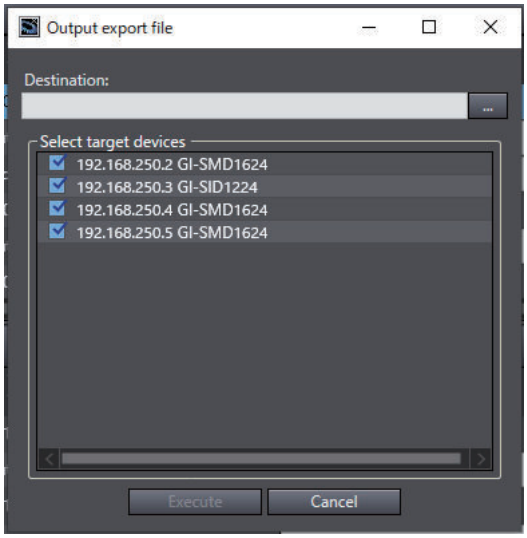
1 In the Multiview Explorer, select the target Safety CPU Unit in the Controller Selection Box.

2 Go to **Configurations and Setup** – **Communications** – **Safety** – **EtherNet/IP Safety Connection Settings** and double-click **Connection Settings (Originator)**.

3 Click the **Export** Button and select **Output export file**.




The Output export file Dialog Box is displayed.




The contents of the setting dialog box are as follows.

| Item | Description |
|---|---|
| Destination | Specify the export file (.tdsg extension) for the destination. |
| Select target devices | Select the target devices to output to the export file. |

4 Select the destination and target devices and click the **Execute** Button.
A batch export file for CIP Safety connection settings with a .tdsg extension is saved.

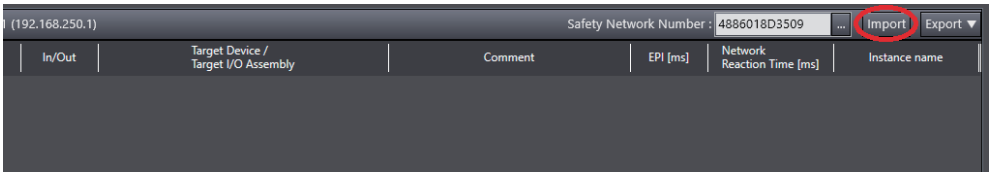
## Batch Import

You can use the batch import function to import CIP Safety connection settings for each EtherNet/IP port. Use the following procedure.

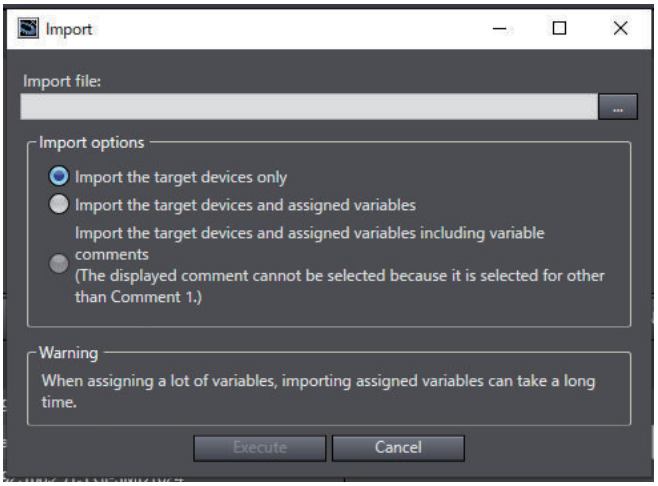
1 In the Multiview Explorer, select the target Safety CPU Unit in the Controller Selection Box.

**2** Go to **Configurations and Setup** – **Communications** – **Safety** – **EtherNet/IP Safety Connection Settings** and double-click **Connection Settings (Originator)**.

**3** Click the **Import** Button.




The Import Dialog Box is displayed.




The contents of the setting dialog box are as follows.

| Item | Description |
|---|---|
| Import file | Specify the file to be imported (extension .tdsg). |
| Import options | Select the import option. You can import data including assigned variables and variable comments.<br>If the display comment is selected other than comment 1 in the option setting, the variable comments cannot be imported. |

**4** Select the import file and import option, and click the **Execute** Button.
The connection settings to be imported are added to the Connection Settings (Originator) Tab Page.

**Precautions for Correct Use**

When you import the target device configuration file, the data integrity is not checked by CRC. Always validate the correct configuration under your responsibility after the import and assure proper execution before you use it for actual operation.

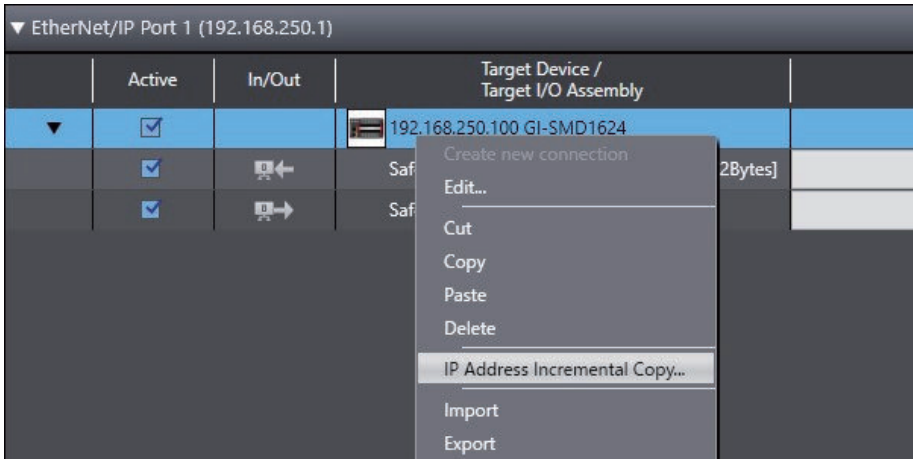
## ● IP Address Incremental Copy

You can use the IP address incremental copy function to replicate the CIP Safety connection setting that has a difference only in the IP address part. Use the following procedure.

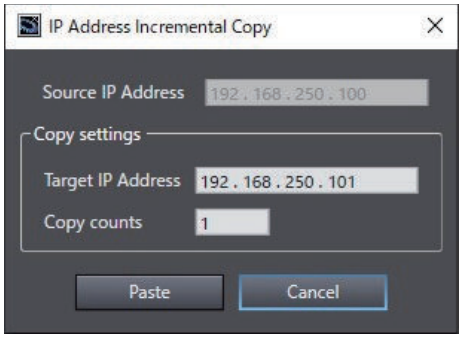
**1** In the Multiview Explorer, select the Safety CPU Unit in the Controller Selection Box.

**2** Go to **Configurations and Setup** – **Communications** – **Safety** – **EtherNet/IP Safety Connection Settings** and double-click **Connection Settings (Originator)**.

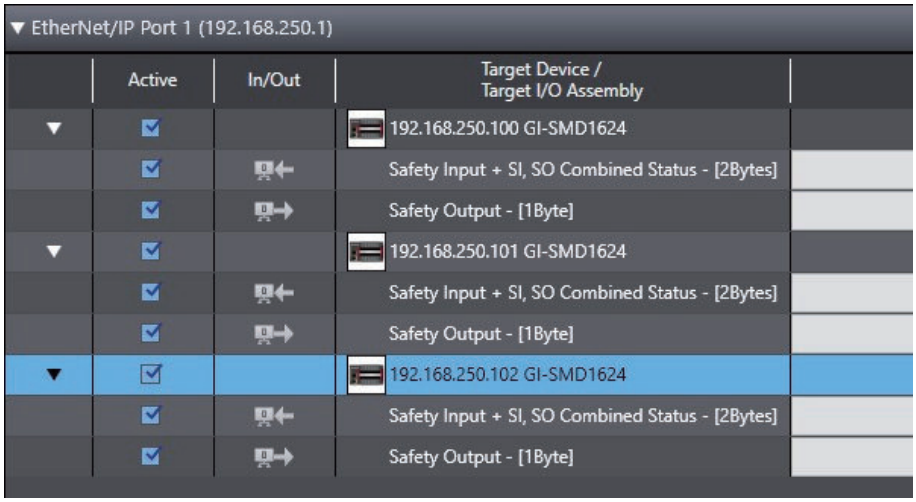
**3** Select desired CIP Safety connection, and right-click the connection and select **IP Address Incremental Copy**.




The IP Address Incremental Copy window is displayed.




**4** Enter the target IP address and copy counts, and then click the Paste button.
The CIP Safety connection setting will be replicated by adding one to the IP address part for the amount of the copy counts.

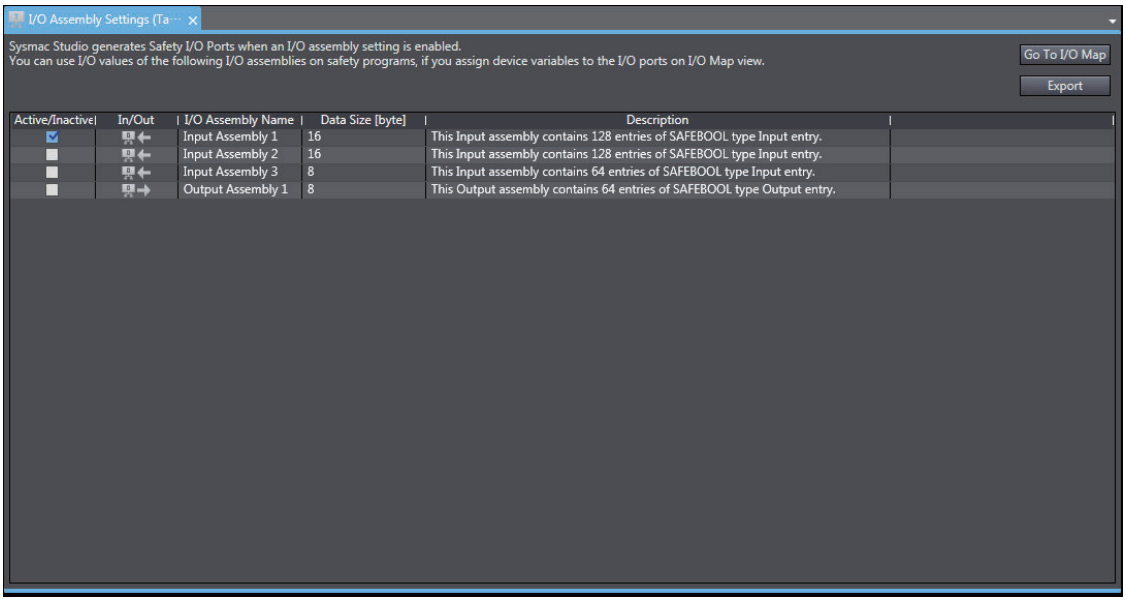


## Target I/O Assembly Settings

To execute the CIP Safety I/O communications between the Safety CPU Units, you need to configure the I/O Assembly settings on the target Safety CPU Unit.
This section describes the procedure to enable or disable the I/O Assembly on the I/O Assembly Settings Tab Page in the Sysmac Studio.

**1** In the Multiview Explorer, select the target Safety CPU Unit in the Controller Selection Box.

**2** Select **Configurations and Setup** – **Communications** – **Safety** – **EtherNet/IP Safety Connection Settings** and double-click **I/O Assembly Settings (Target)**.
The I/O Assembly Settings (Target) Tab Page is displayed.



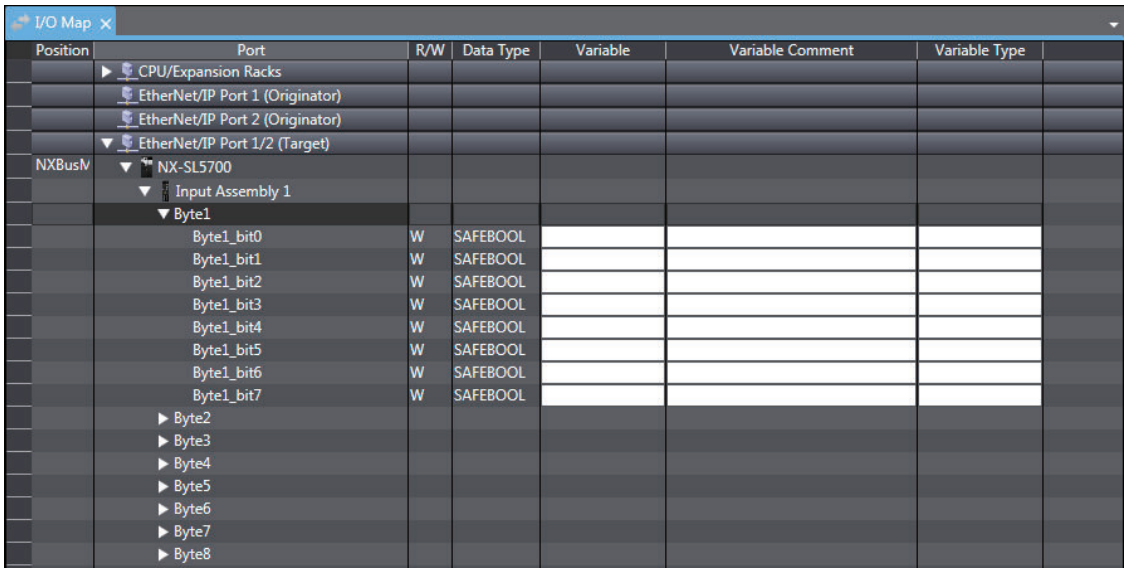

The I/O Assembly Settings (Target) Tab Page consists of the items shown in the following table.

| Item | Description |
|---|---|
| Active/Inactive | Use this box to enable or disable each I/O Assembly. When you enable the assembly, a port is added to the I/O Map.<br>Selected: I/O Assembly is enabled.<br>Not selected: I/O Assembly is disabled. |
| In/Out | Shows the data direction of each I/O Assembly.<br>: In – Data sent to the originator device.<br>: Out – Data received from the originator device. |
| I/O Assembly Name | The name of each I/O Assembly is displayed. It is same as the port name displayed on the I/O Map. |
| Data Size [byte] | The byte size of each I/O Assembly is displayed. |
| Description | The description of each I/O Assembly is displayed. |
| **Go To I/O Map** Button | Click this button to open the I/O Map Tab Page. |
| **Export** Button | Exports the target device settings as a file (extension .tds). |

The following lists the types of selectable I/O Assembly.

| Name | Assembly number | Data direction | Size |
|---|---|---|---|
| InputAssembly 1 | 768(0x300) | Input – Data sent to the originator device | 16 bytes (SAFEBOOL × 128) |
| InputAssembly 2 | 769(0x0301) | Input – Data sent to the originator device | 16 bytes (SAFEBOOL × 128) |
| InputAssembly 3 | 770(0x0302) | Input – Data sent to the originator device | 8 bytes (SAFEBOOL × 64) |
| OutputAssembly 1 | 928(0x03A0) | Output – Data received from the originator device | 8 bytes (SAFEBOOL × 64) |

The activated I/O Assembly is displayed as a port on the I/O Map of the originator device to communicate with as shown below. You can use it on a safety program by assigning a variable to the I/O port.




**Precautions for Correct Use**

The I/O Assembly for the Safety CPU Unit cannot be used with the tag data link connection.

## Connecting Target Devices of Other Manufacturers

To configure the CIP Safety communication settings for target devices manufactured by other companies, you need to create or install the EDS file for non-OMRON target devices.

### ● Creating an EDS File

This function can create an EDS file to set CIP Safety connections with a CIP Safety target device.

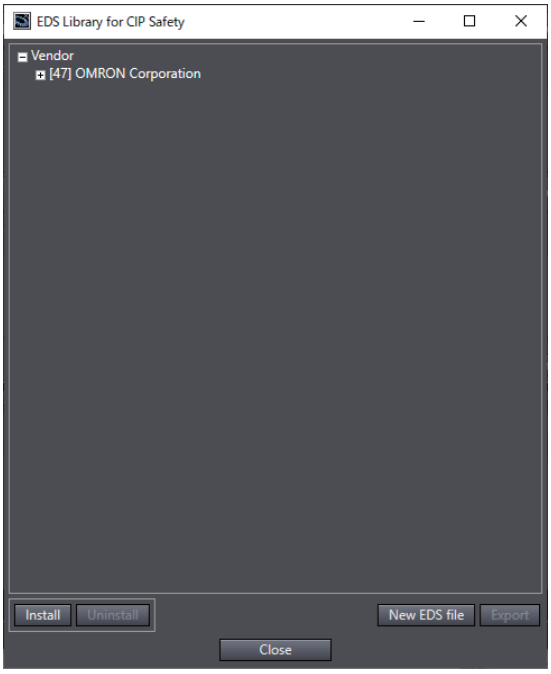
**Precautions for Correct Use**

- For contents to be specified, contact the manufacturer of the target devices.
- Always validate correct operations of the connection established by using the created EDS files under your responsibility before you use it for actual operation. If there is an error in the created EDS file, the Safety CPU Unit and the target device may not be able to connect.
- You can determine the created EDS (target device) by the EDSFileCRC since the EDSFileCRC is automatically generated according to the contents of the EDS file. Make sure that the displayed EDSFileCRC is what you intended, when you use the created EDS (target device).

The following procedure describes how to create an EDS file.

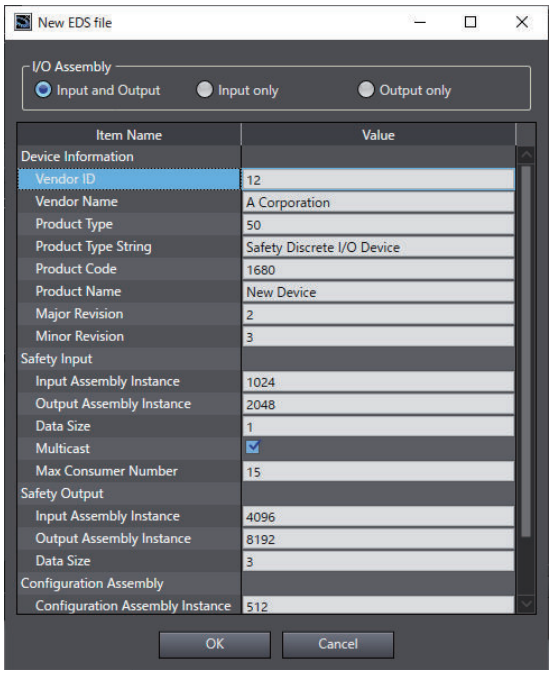
***1*** Right-click on the Connection Settings (Originator) screen and select the menu **EDS Library for CIP Safety**.

The EDS Library for CIP Safety screen is displayed.




***2*** Click the **New EDS file** button.

The New EDS file window is displayed.

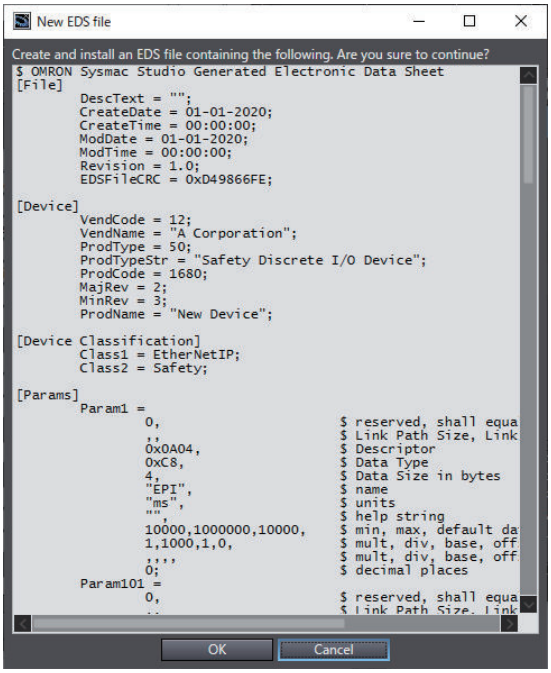



Specify the information of the target device.

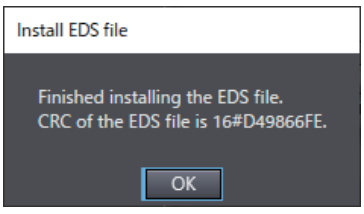
| Item | Description |
|---|---|
| I/O Assembly | Select the I/O assembly supported by the target device.<br>Input and Output: Select Input and Output when the target device supports the input and output connections.<br>Input only: Select Input only when the target device supports the input connection only.<br>Output only: Select Output only when the target device supports the output connection only. |
| Device Information | Specify the ID information of the device. |
| Vendor ID | Specify the vendor ID of the target device.<br>Used for the connection settings with the target device. |
| Vendor Name | Specify the vendor name of the target device.<br>Used for display on the Sysmac Studio. |
| Product Type | Specify the product type of the target device.<br>Used for the connection settings with the target device. |
| Product Type String | Specify the vendor name of the target device.<br>Used for display on the Sysmac Studio. |
| Product Code | Specify the product code of the target device.<br>Used for the connection settings with the target device. |
| Product Name | Specify the product name of the target device.<br>Used for display on the Sysmac Studio. |
| Major Revision | Specify the major revision of the target device.<br>Used for the connection settings with the target device. |
| Minor Revision | Specify the minor revision of the target device.<br>Used for the connection settings with the target device. |
| Safety Input | Enter the safety input connection information. |
| Input Assembly Instance | Specify the instance number of the input assembly for the safety input connection. |
| Output Assembly Instance | Specify the instance number of the output assembly for the safety input connection. |
| Data Size | Specify the data size of the safety input connection. |

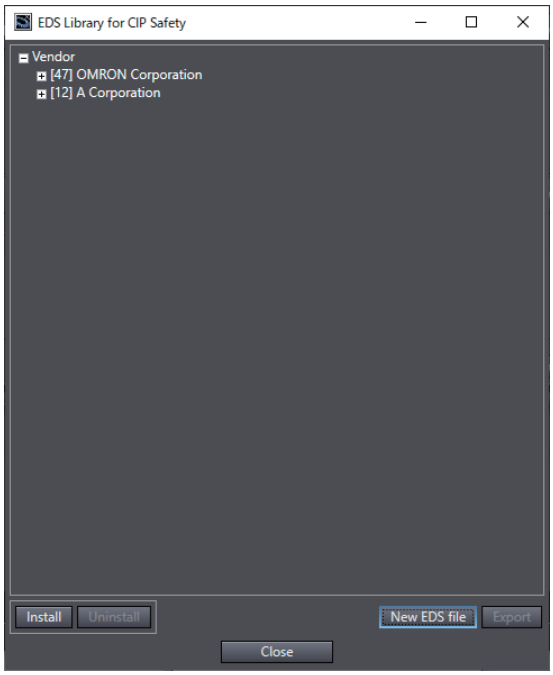
| Item | | Description |
|---|---|---|
| | Multicast | Check the check box to use the multicast connection. |
| | Max Consumer Number | Specify the maximum number of consumers when you use the multicast connection. |
| Safety output | | Enter the safety output connection information. |
| | Input Assembly Instance | Specify the instance number of the input assembly for the safety output connection. |
| | Output Assembly Instance | Specify the instance number of the output assembly for the safety output connection. |
| | Data Size | Specify the data size of the safety output connection. |
| Configuration Assembly | | Enter the configuration assembly information. |
| | Configuration Assembly Instance | Specify the instance number of the configuration assembly. |
| Format Type | | Specify the format type of the CIP Safety communications. |
| | Safety Format Support | Select the supported format from the following.<br>Base Only: Only Base format is supported.<br>Extended Only: Only Extended format is supported (default).<br>Base and Extended: Both Base and Extended formats are supported. |

***3*** Click the **OK** button to display the EDS contents to be created.
Check the contents and click the **OK** button.




***4*** The created EDS file is registered in the EDS Library for CIP Safety screen. After the registration is completed, the EDSFileCRC will be displayed in the dialog.

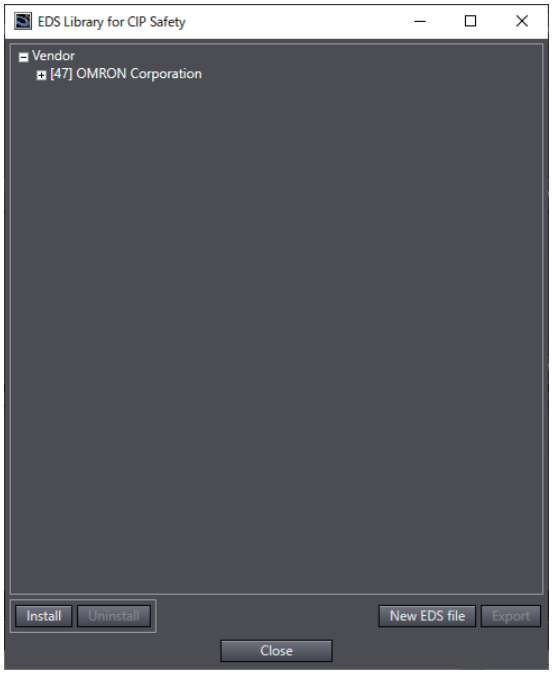




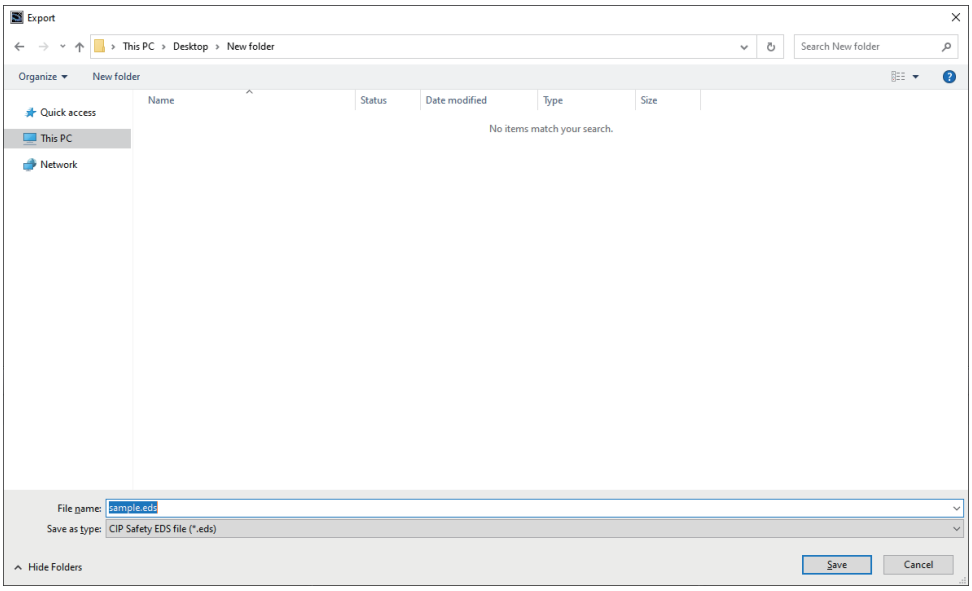

## ● Exporting the EDS File

You can reuse the created EDS file on other computers by exporting it.
The procedure to export the EDS file is described below.

***1*** Right-click on the Connection Settings (Originator) screen and select the menu **EDS Library for CIP Safety**.
The EDS Library for CIP Safety screen is displayed.




***2*** Click the **Export** button.
The export window is displayed.

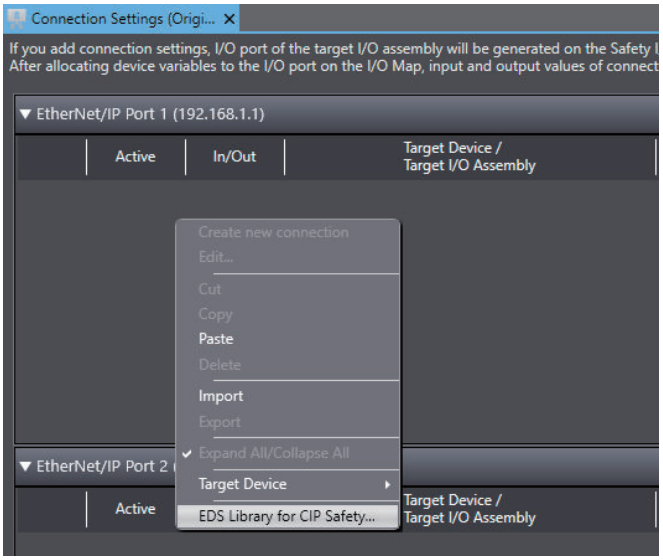



Specify the folder and the file name and then click the **Save** button to save the EDS file.

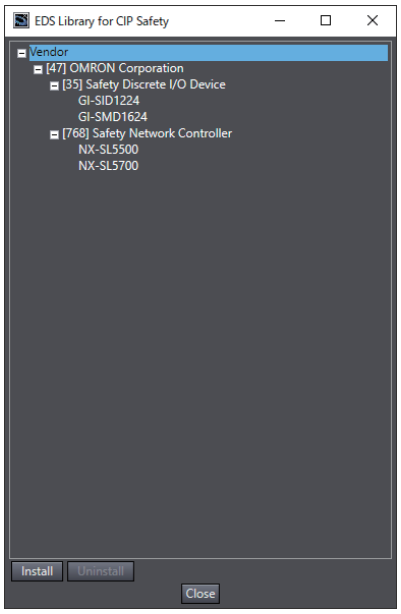
## ● Installing the EDS File

You can install and reuse EDS files created on other computers.
The procedure to install the EDS file is described below.

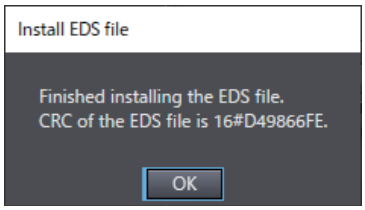
***1*** Right-click on the Connection Settings (Originator) screen and select the menu **EDS Library for CIP Safety**.




The EDS Library for CIP Safety screen is displayed.




**2** Click the **Install** button and select the EDS file for the target device you will add. Then, click the **Open** button.
After the installation is completed, the EDSFileCRC will be displayed in the dialog. Make sure the intended EDS is installed.




**3** The non-OMRON target device is added to the EDS Library for CIP Safety.

**4** Click the **Close** button to close the EDS Library for CIP Safety screen. The device registered to the EDS Library for CIP Safety is displayed in the list of connectable target devices in the Toolbox on the Connection Settings (Originator) screen.

### Precautions for Correct Use

- If you install third-party EDS files, always validate correct operations of the connection established by using the installed EDS files under your responsibility before you use it for actual operation.
- Make sure that the displayed EDSFileCRC is what you intended, when you use the installed EDS (target device).
- EDS files (target devices) with different EDSFileCRCs are registered as different target devices.
- In order to open a project file, you need to install the EDS files for the target devices included in the project file.

### Additional Information

For details on the target devices that can be configured in the CIP Safety communication settings, contact the manufacturer of the target devices.

### ● Uninstalling the EDS File

To uninstall the EDS file, select a target device to delete from the list on the EDS Library for CIP Safety screen, and then click the **Uninstall** button.

## 5-4-3 Calculating the Number of Connections

### ● Calculating the Number of Safety I/O Connections for the Safety CPU Unit

You must specify the total number of CIP Safety originator connections, CIP Safety target connections, and FSoE master connections, which must be set within the number of safety I/O connections that you can set for the Safety CPU Unit. You cannot set the number that exceeds the number of safety I/O connections that you can set.

**Precautions for Correct Use**

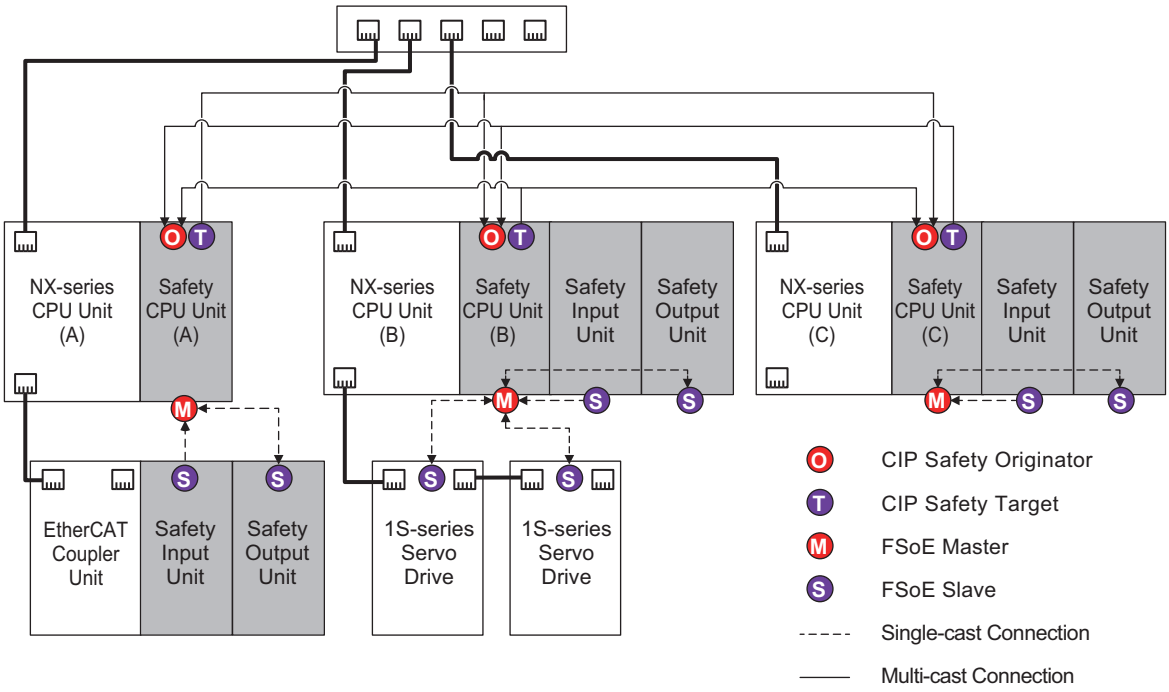
There is a restriction on the data size that can be communicated between the Safety CPU Unit and the NX Bus Master, as well as the restrictions on the number of Safety I/O connections. This data size includes exposed variables. You can check the usage on the Memory Usage Tab Page of the Sysmac Studio.

- Counting the Number of CIP Safety Originator Connections
  For the CIP Safety originator connections, you can establish a unidirectional input or output communication with a CIP Safety target per connection. Two connections are used to establish bidirectional input and output communications.

- Counting the Number of CIP Safety Target Connections
  CIP Safety target connections have the single-cast connection and the multi-cast connection.
  For the single-cast connection, you can establish a unidirectional input or output communication with a CIP Safety originator per connection. Two connections are used to establish bidirectional input and output communications.
  For the multi-cast connection, you can establish a unidirectional input communication with multiple CIP Safety originators per connection.

- Counting the Number of FSoE Master Connections
  For the FSoE master connections, you can establish bidirectional input and output communications with a FSoE slave per connection.

- Example of Calculating the Number of Safety I/O Connections
  A calculation example of using the combination of the NX-series CPU Units and Safety CPU Units is given below.




| | Number of CIP Safety originator connections (X) | Number of CIP Safety target connections (Y) | Number of FSoE master connections (Z) | Total (X+Y +Z) |
|---|---|---|---|---|
| Safety CPU Unit (A) | 2 | 1 | 2 | 5 |
| Safety CPU Unit (B) | 2 | 1 | 4 | 7 |
| Safety CPU Unit (C) | 2 | 1 | 2 | 5 |

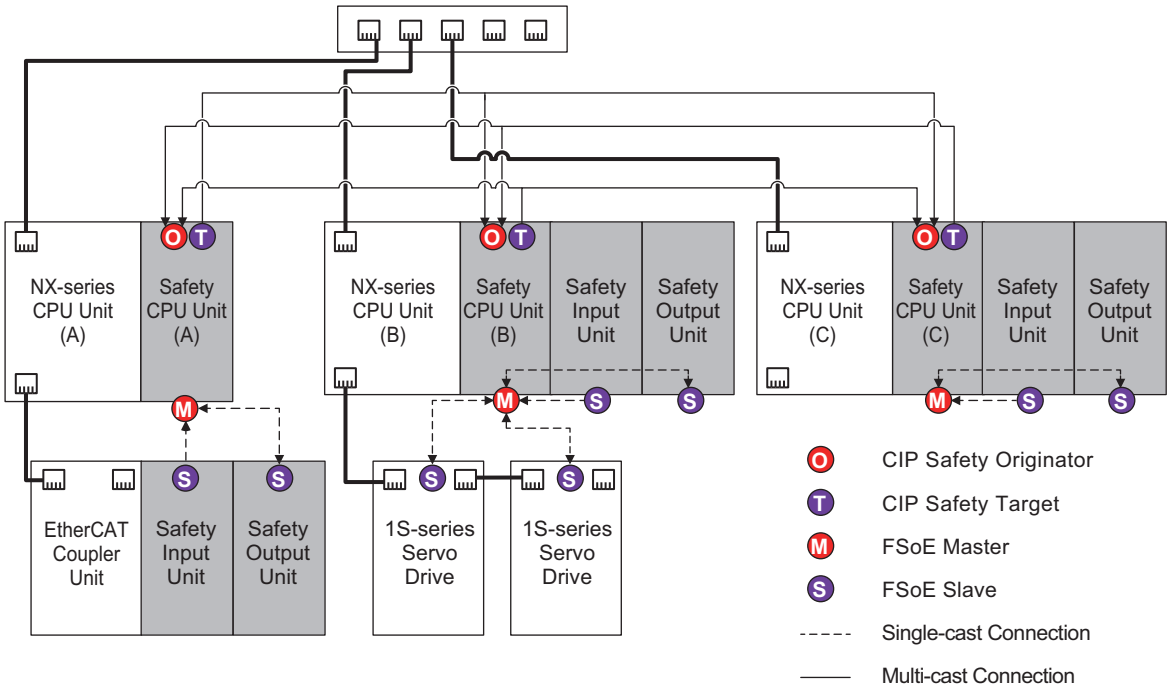
## ● Calculating the Number of Routable CIP Safety Connections for the NX-series CPU Unit or NX-series EtherNet/IP Unit

You must specify the total number (m+n) of targets that establish connections as an originator (m) and originators that establish connections as a target (n), which must be set within the maximum number of routable CIP Safety connections for the NX-series CPU Unit. If you set a number that exceeds the maximum number of routable CIP Safety connections, the *CIP Safety Originator Connection Not Established Error* (80310000 hex) will occur. In addition, if the total number of CIP Safety originator connections and CIP Safety target connections exceeds the number of routable CIP Safety connections, you cannot perform the connection settings.

If a multi-cast connection is opened for multiple CIP Safety originators and connections, the number of originators that establish connections as a target becomes the number of CIP Safety originators.

If more than one single-cast connection is established for a CIP Safety target device, the number of targets that have established connections as an originator is equal to the number of single-cast connections.

- Example of Calculating the Maximum Number of Routable CIP Safety Connections
  A calculation example of using the combination of the NX-series CPU Units and Safety CPU Units is given below.




| | Number of targets that establish connections as an originator (m) | Number of originators that establish connections as a target (n) | Total (m+n) |
|---|---|---|---|
| NX-series CPU Unit (A) | 2 | 2 | 4 |
| NX-series CPU Unit (B) | 2 | 2 | 4 |
| NX-series CPU Unit (C) | 2 | 2 | 4 |

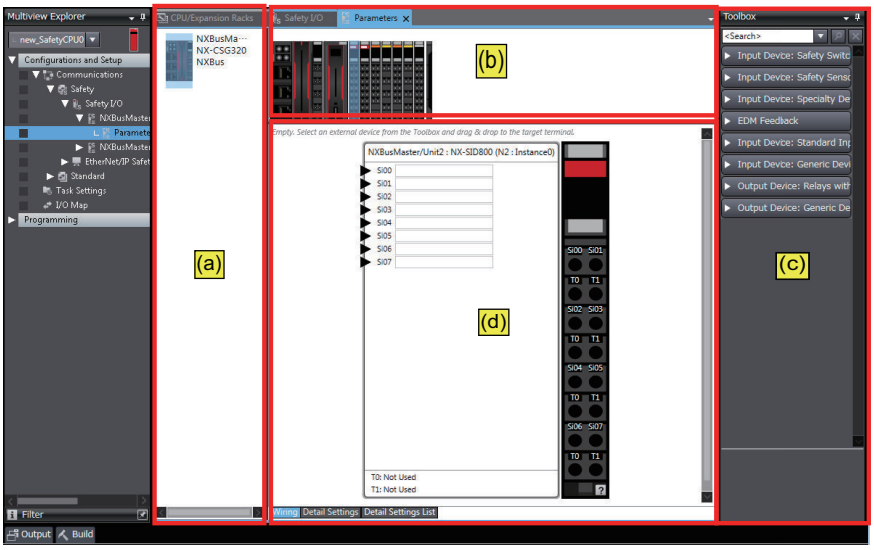
# 5-5 Safety I/O Functions

You set the safety input functions and safety output functions of the Safety I/O Units when you assign input devices and output devices to the Safety I/O Units with the Sysmac Studio.

This section describes how to assign devices that are connected. Refer to *4-3-1 Safety Input Function* on page 4-10 and *4-3-2 Safety Output Function* on page 4-32 for details on the safety input functions and safety output functions.

**1** In the Multiview Explorer, select the target Safety CPU Unit in the Controller Selection Box.

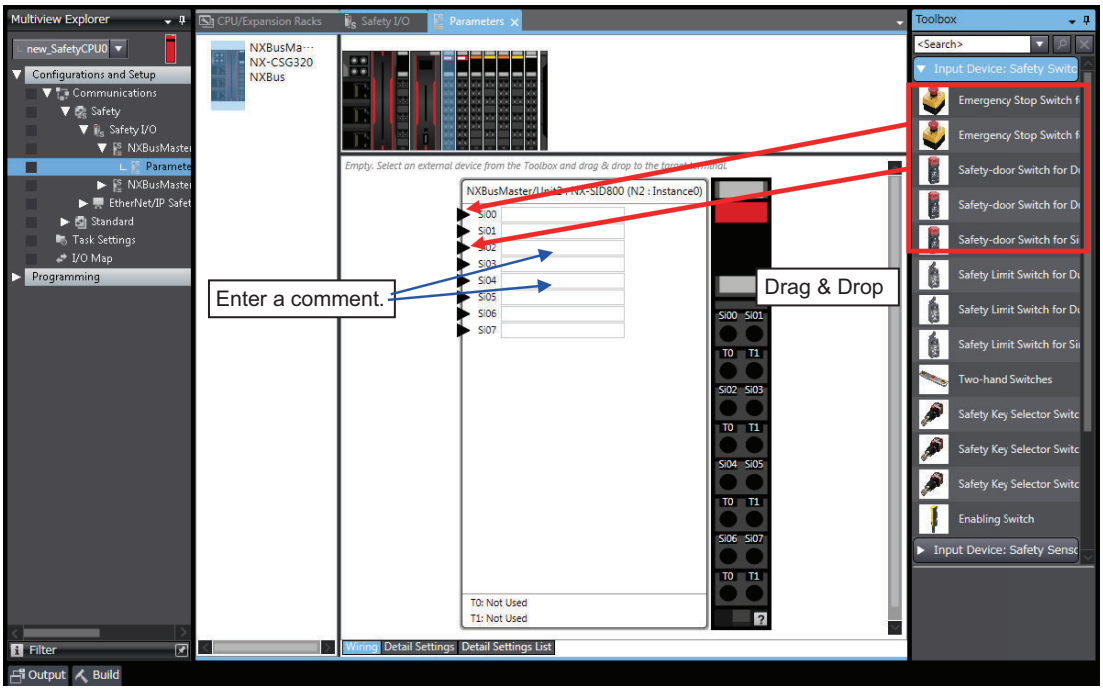
**2** Go to **Configurations and Setup** – **Communications** – **Safety** – **Safety I/O**. Under the Safety I/O Unit name, double-click **Parameters**.
The Parameters Tab Page shown below is displayed.



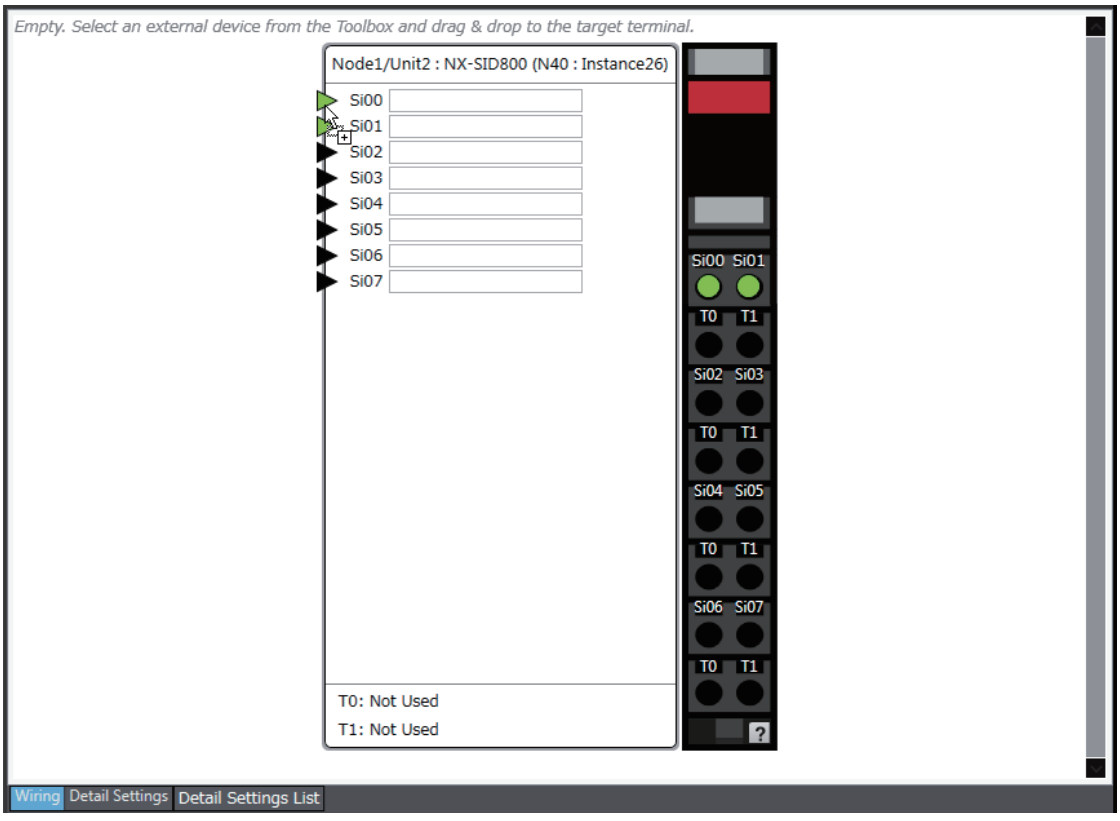

The items in the Parameters Tab Page of safety slave unit setting are described below.

| Letter | Name | Function |
|---|---|---|
| (a) | NX Bus Master List Pane | This pane lists NX bus masters to which the Safety I/O Unit that can perform safety communications with the Safety CPU Unit is mounted. |
| (b) | NX Unit Configuration Pane | This pane shows the configuration of the NX Units that are connected to the NX bus master. |
| (c) | Toolbox | The toolbox displays the input and output devices that can be set for the Safety I/O Units. |
| (d) | Parameter Settings Pane | This pane is used to make settings for the input and output devices that are set for the Safety I/O Units. You can arrange the devices, set diagnosis functions, change icons, etc.<br><br>Refer to *A-8 Icon list for Safety Slave Unit Parameters* on page A-86 for details on changing icons. |

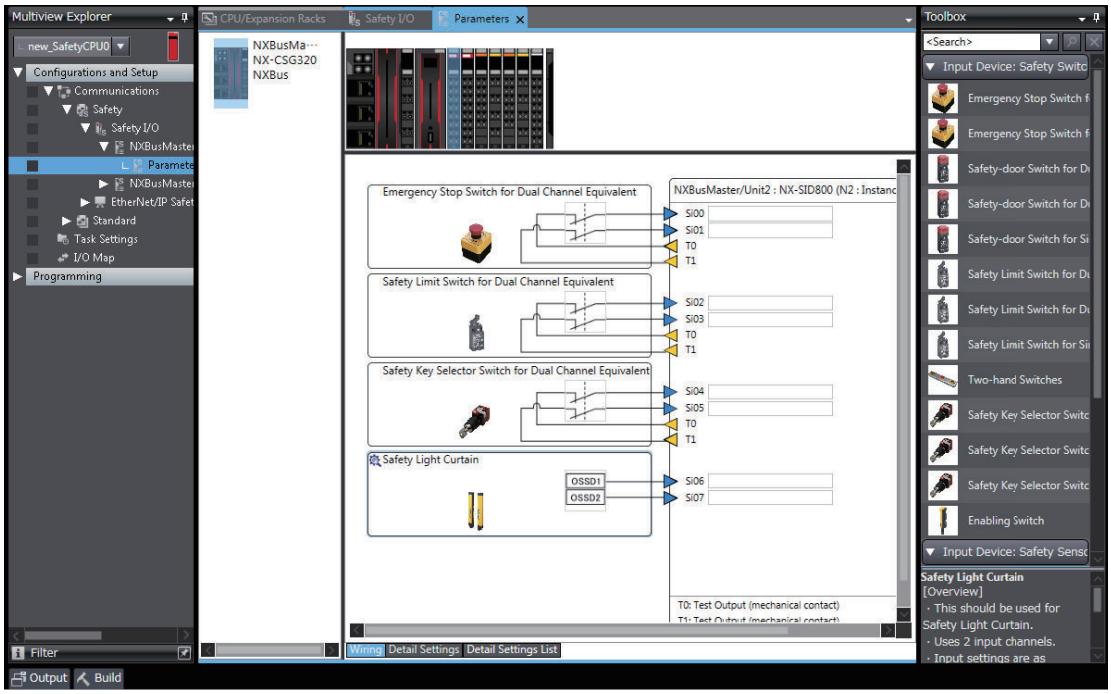
**3** Select a device from the Toolbox to connect to the safety input terminal or safety output terminal of the Safety I/O Units, and drag it to the desired I/O terminal.




When you drag the device to connect to a terminal where it can be dropped, a + mark appears below the mouse cursor as shown below.




**4** When you complete the settings, the following is displayed. Change the settings and enter comments.




Refer to *4-3-1 Safety Input Function* on page 4-10 and *4-3-2 Safety Output Function* on page 4-32 for the I/O devices that you can connect and the settings for each I/O device.

### Precautions for Correct Use

If you select an input device that cannot be set for a Safety Input Unit, an error will occur and the frame around the input device will be displayed in red.




### Additional Information

The I/O terminal comments on the Parameters Tab Page are linked to the device variable comments and the global variable comments.

# 5-6 Registering Device Variables

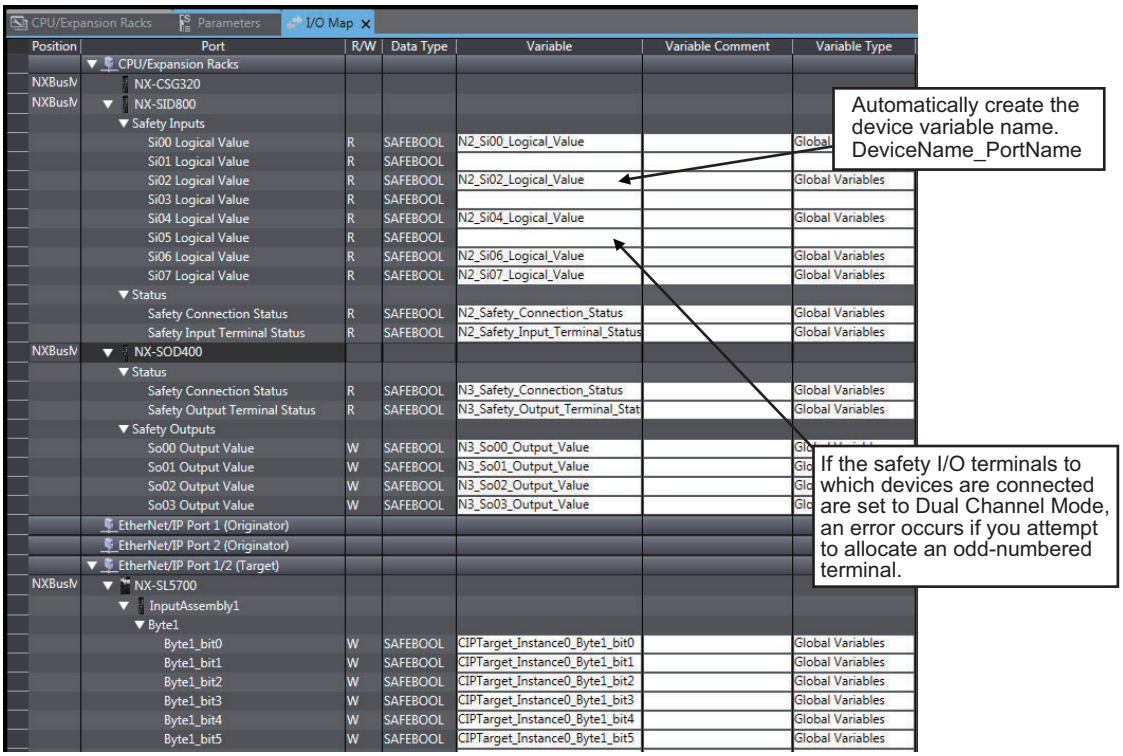
Device variables are used to access data in devices (slaves and Units).
This data is accessed through a port that acts as an interface to an external device. This logical port is called an "I/O Port".

To make the values of the I/O on the Safety I/O Units, Safety I/O Terminal and the other Safety CPU Unit available in the safety program in the Safety CPU Unit, you must register device variables for the I/O ports on the Safety I/O Units, Safety I/O Terminal and the other Safety CPU Unit.
This section describes how to assign device variables to I/O ports through the I/O Map of the Safety CPU Unit.

## ● Registering New Variables or Creating Them Automatically

If the Controller configuration and the external devices to connect are finalized before you register the variables that are used in the program, you can create the device variable for the I/O ports by manually entering the device variable name, or by creating them automatically.

**1** On the Safety CPU Unit Setup and Programming View, select **Configurations and Setup** and double-click **I/O Map**.
The I/O Map will be displayed.




**2** Select an I/O port in the I/O Map for the Safety CPU Unit, and enter a variable name directly in the **Variable** Column. Or, select a Unit or I/O port, and then right-click and select **Create Device Variable** from the menu.
If you choose the **Create Device Variable** command, the device variables are automatically named with the device name and port name. The device variables that you enter or automatically create are registered in the global variable table.

## Selecting from the Registered Variables

If the variables that are used in the program are registered before you finalize on the Controller configuration and the external devices to connect, you can select and assign variables to the I/O ports for the safety I/O terminals as long as the variables are registered in the variable table.

1 On the Safety CPU Unit Setup and Programming View, select **Configurations and Setup** and double-click **I/O Map**.
The I/O Map will be displayed.

2 Select an I/O port and select a user-defined variable from the list of variables that are registered in the variable table to assign the variable to that I/O port.

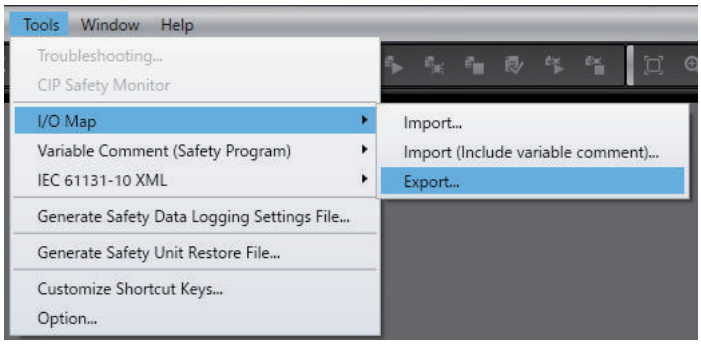
**Additional Information**

If the I/O terminals on the Safety I/O Units are set to Dual Channel Mode, the device variable can only be assigned to an even-numbered terminal.

## Importing from a CSV File

You can register device variables all at once by exporting the current I/O Map display content as a CSV format file and importing the content after editing with a spreadsheet application.

1 On the **Tools** menu, select **I/O Map** - **Export**.



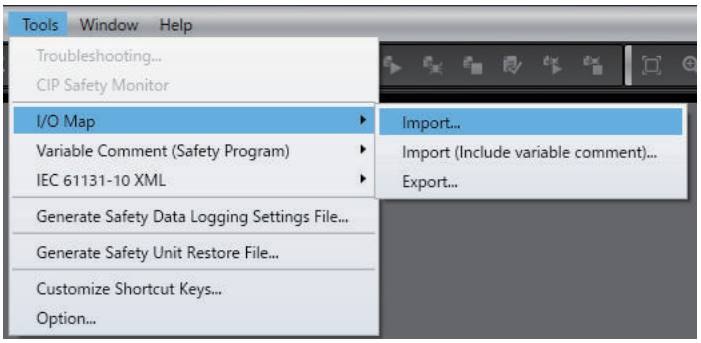

2 Save the CSV file to any location.

3 Open the CSV file in a spreadsheet application.

| Sysmac StVer.1.0 | | | | | | |
|---|---|---|---|---|---|---|
| Level | Position | Port | R/W | Data Type | Variable | Variable Comment |
| 0 | [Network] | CPU/Expansion Racks | | | | |
| 1 | NXBusMaster | NX102-1200 | | | | |
| 1 | NXBusMaster/Unit2 | NX-SID800 | | | | |
| 2 | | Safety Inputs | | | | |
| 3 | | Si00 Logical Value | R | SAFEBOOL | ESTOP_Si01 | Emergency Stop S-IN01 |
| 3 | | Si01 Logical Value | R | SAFEBOOL | | |
| 3 | | Si02 Logical Value | R | SAFEBOOL | Door_Si02_R | Door S-IN02 |
| 3 | | Si03 Logical Value | R | SAFEBOOL | | |
| 3 | | Si04 Logical Value | R | SAFEBOOL | Door_Si03_L | Door S-IN03 |
| 3 | | Si05 Logical Value | R | SAFEBOOL | | |
| 3 | | Si06 Logical Value | R | SAFEBOOL | ResetSwitch_R01 | Reset Switch R01 |
| 3 | | Si07 Logical Value | R | SAFEBOOL | | |
| 2 | | Status | | | | |
| 3 | | Safety Connection Status | R | SAFEBOOL | N2_Safety_Connection_Status | |
| 3 | | Safety Input Terminal Status | R | SAFEBOOL | N2_Safety_Input_Terminal_Status | |
| 1 | NXBusMaster/Unit3 | NX-SIH400 | | | | |
| 2 | | Safety Inputs and Status | | | | |
| 3 | | Si00 Logical Value | R | SAFEBOOL | SR1_EDM_Si05 | EDM_S-IN05 |
| 3 | | Si01 Logical Value | R | SAFEBOOL | SR1_EDM_Si06 | EDM_S-IN06 |
| 3 | | Si02 Logical Value | R | SAFEBOOL | SR2_EDM_Si07 | EDM_S-IN07 |
| 3 | | Si03 Logical Value | R | SAFEBOOL | SR2_EDM_Si08 | EDM_S-IN08 |
| 3 | | Safety Connection Status | R | SAFEBOOL | N3_Safety_Connection_Status | |
| 3 | | Safety Input Terminal Status | R | SAFEBOOL | N3_Safety_Input_Terminal_Status | |

***4*** In the **Variable** and **Variable Comment** columns, set any variable name and variable comment.
After you edit data with a spreadsheet application, save the data in the CSV format (UTF-8).

***5*** From the main menu, select **Tools - I/O Map - Import** or **Import (Include variable comment)**.




**Import** imports only the **Variable** column of the CSV file. Even if you edited the **Variable Comment** column in the CSV file, the changes are not applied to the I/O Map.
**Import (Include variable comment)** imports the content of the **Variable** column and the **Variable Comment** column of the CSV file. If the registered variable is assigned by the **Import (Include variable comment)** function, the content of the **Variable Comment** column is applied.

### Precautions for Correct Use

- When you register device variables by importing a CSV file, the data integrity is not checked by CRC. Always validate the correct variable assignments under your responsibility after the import and assure proper execution before you use it for actual operation.
- When the display comment is set to other than **Comment 1**, the I/O Map **Import (Include variable comment)** and **Export** cannot be executed.
- When you import a CSV file with its **Variable** column left blank, the variable assignments are canceled.

## I/O Ports for Safety I/O Units That Are Displayed in the I/O Map of the Safety CPU Unit

The I/O ports for Safety I/O Units that are displayed in the I/O Map of the Safety CPU Unit are described in this section.

### ● NX-SIH400 Safety Input Unit

| Port | Data type | R/W | Name | Description | Default |
|---|---|---|---|---|---|
| Si00 Logical Value | SAFEBOOL | R | Si00 Logical Value | Gives the status of safety input terminal Si00.<br>0: OFF, 1: ON | 0 |
| Si01 Logical Value | SAFEBOOL | R | Si01 Logical Value | Gives the status of safety input terminal Si01.<br>0: OFF, 1: ON | 0 |
| Si02 Logical Value | SAFEBOOL | R | Si02 Logical Value | Gives the status of safety input terminal Si02.<br>0: OFF, 1: ON | 0 |
| Si03 Logical Value | SAFEBOOL | R | Si03 Logical Value | Gives the status of safety input terminal Si03.<br>0: OFF, 1: ON | 0 |
| Safety Connection Status | SAFEBOOL | R | Safety Connection Status | This flag indicates when a safety connection is active. Use it for an input to the Activate terminal on a safety FB or for safety connection/disconnection applications. | 0 |
| Safety Input Terminal Status | SAFEBOOL | R | Safety Input Terminal Status | This flag indicates the status of the safety input terminals.<br>0: An error has occurred on one of the safety input terminals.<br>1: All of the safety input terminals are normal (no errors). | 0 |

### ● NX-SID800 Safety Input Unit

| Port | Data type | R/W | Name | Description | Default |
|---|---|---|---|---|---|
| Si00 Logical Value | SAFEBOOL | R | Si00 Logical Value | Gives the status of safety input terminal Si00.<br>0: OFF, 1: ON | 0 |
| Si01 Logical Value | SAFEBOOL | R | Si01 Logical Value | Gives the status of safety input terminal Si01.<br>0: OFF, 1: ON | 0 |
| Si02 Logical Value | SAFEBOOL | R | Si02 Logical Value | Gives the status of safety input terminal Si02.<br>0: OFF, 1: ON | 0 |
| Si03 Logical Value | SAFEBOOL | R | Si03 Logical Value | Gives the status of safety input terminal Si03.<br>0: OFF, 1: ON | 0 |
| Si04 Logical Value | SAFEBOOL | R | Si04 Logical Value | Gives the status of safety input terminal Si04.<br>0: OFF, 1: ON | 0 |
| Si05 Logical Value | SAFEBOOL | R | Si05 Logical Value | Gives the status of safety input terminal Si05.<br>0: OFF, 1: ON | 0 |
| Si06 Logical Value | SAFEBOOL | R | Si06 Logical Value | Gives the status of safety input terminal Si06.<br>0: OFF, 1: ON | 0 |
| Si07 Logical Value | SAFEBOOL | R | Si07 Logical Value | Gives the status of safety input terminal Si07.<br>0: OFF, 1: ON | 0 |

| Port | Data type | R/W | Name | Description | Default |
|---|---|---|---|---|---|
| Safety Connection Status | SAFEBOOL | R | Safety Connection Status | This flag indicates when a safety connection is active. Use it for an input to the Activate terminal on a safety FB or for safety connection/disconnection applications. | 0 |
| Safety Input Terminal Status | SAFEBOOL | R | Safety Input Terminal Status | This flag indicates the status of the safety input terminals.<br>0: An error has occurred on one of the safety input terminals.<br>1: All of the safety input terminals are normal (no errors). | 0 |

## ● NX-SOH200 Safety Output Unit

| Port | Data type | R/W | Name | Description | Default |
|---|---|---|---|---|---|
| Safety Connection Status | SAFEBOOL | R | Safety Connection Status | This flag indicates when a safety connection is active.<br>Use it for an input to the Activate terminal on a safety FB or for safety connection/disconnection applications. | 0 |
| Safety Output Terminal Status | SAFEBOOL | R | Safety Output Terminal Status | This flag indicates the status of the safety output terminals.<br>0: An error has occurred on one of the safety output terminals.<br>1: All of the safety output terminals are normal (no errors). | 0 |
| So00 Output Value | SAFEBOOL | W | So00 Output Value | Gives the status of safety output terminal So00.<br>0: OFF, 1: ON | 0 |
| So01 Output Value | SAFEBOOL | W | So01 Output Value | Gives the status of safety output terminal So01.<br>0: OFF, 1: ON | 0 |

## ● NX-SOD400 Safety Output Unit

| Port | Data type | R/W | Name | Description | Default |
|---|---|---|---|---|---|
| Safety Connection Status | SAFEBOOL | R | Safety Connection Status | This flag indicates when a safety connection is active.<br>Use it for an input to the Activate terminal on a safety FB or for safety connection/disconnection applications. | 0 |
| Safety Output Terminal Status | SAFEBOOL | R | Safety Output Terminal Status | This flag indicates the status of the safety output terminals.<br>0: An error has occurred on one of the safety output terminals.<br>1: All of the safety output terminals are normal (no errors). | 0 |
| So00 Output Value | SAFEBOOL | W | So00 Output Value | Gives the status of safety output terminal So00.<br>0: OFF, 1: ON | 0 |
| So01 Output Value | SAFEBOOL | W | So01 Output Value | Gives the status of safety output terminal So01.<br>0: OFF, 1: ON | 0 |
| So02 Output Value | SAFEBOOL | W | So02 Output Value | Gives the status of safety output terminal So02.<br>0: OFF, 1: ON | 0 |

| Port | Data type | R/W | Name | Description | Default |
|---|---|---|---|---|---|
| So03 Output Value | SAFE-BOOL | W | So03 Output Value | Gives the status of safety output terminal So03.<br>0: OFF, 1: ON | 0 |

# 5-7 Exposing Variables to Standard Controllers

This section describes how to control and monitor a Safety CPU Unit from a standard controller through standard process data communications.

## 5-7-1 Exposing Global Variables

To ensure that global variables in the Safety CPU Unit are not accidentally used by a program of the standard controller, they are not assigned to the standard controller at the time of their setting.
When you set global variables in the Safety CPU Unit for standard process data communications, the variables are exposed as I/O ports in the I/O Map of the CPU Rack and Communications Coupler Unit.

When the target unit is connected to the CPU Unit or the EtherCAT Coupler Unit is used, the global variable of the Safety CPU Unit can be accessed using the program of the NJ/NX-series CPU Unit by allocating the variable to the exposed I/O port.
The I/O Map of the NJ/NX-series CPU Unit displays the data types that correspond to the data types of the exposed Safety CPU Unit variables.
The following table gives the variable data types that can be exposed for Safety CPU Unit and the corresponding data types that are displayed for the NJ/NX-series CPU Unit.

| Variable data type that can be exposed for Safety CPU Unit | Data type displayed for NJ/NX-series CPU Unit | Data size in bytes |
|---|---|---|
| BOOL | BOOL | 1 |
| BYTE | USINT | 1 |
| WORD | UINT | 2 |
| INT | INT | 2 |
| DINT | DINT | 4 |

The variables that can be exposed for the Safety CPU Unit are restricted as given in the following table.

| Item | | Restriction |
|---|---|---|
| Number of exposed variables | Inputs | 254 variables max. |
| | Outputs | 253 variables max. |

**Precautions for Correct Use**

- There is a restriction on the data size that can be communicated between the Safety CPU Unit and the NX bus master, as well as the restrictions on the number of exposed variables. This data size limit includes FSoE communications and CIP Safety communications. You can check the usage on the Memory Usage Tab Page of the Safety CPU Unit.
- If the settings cause a limit to be exceeded, a red "!" icon is displayed by **Exposed Variables** in the Multiview Explorer.

## 5-7-2 Setting Exposed Variables

This section describes how to expose Safety CPU Unit variables to a standard controller. Exposed Safety CPU Unit variables (exposed variables) appear in the I/O Map of the NX Bus Master.
Use one of the following methods to set exposed variables.

- Register new variables on the Exposed Variables Tab Page.
- Select global variables on the Exposed Variables Tab Page.
- Set the Expose Column for global variables.
- Copy global variables and paste them on the Exposed Variables Tab Page.

The procedures for these methods are given below.

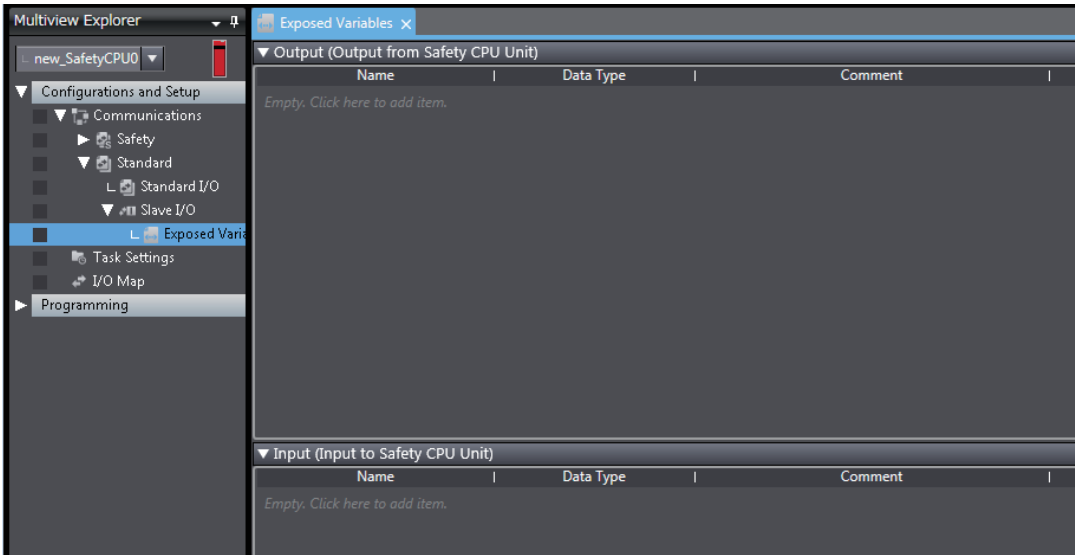
**Precautions for Correct Use**

- If you change the input/output value of the exposed variable, the device variable allocated to the standard CPU Unit will be cancelled. In this case, you need to assign the device variables, and then transfer the settings and programs to the standard CPU Unit.
- If you change the exposed variable, you must transfer the settings and programs to the standard CPU Unit. After you transfer the settings and programs, check that the operation of the standard CPU Unit is correct.
- If you change the exposed variable in the EtherNet/IP Slave Terminal configuration, the I/O allocations of the EtherNet/IP Couplers Unit will be changed. Check that the operation of the EtherNet/IP Coupler Unit is correct.

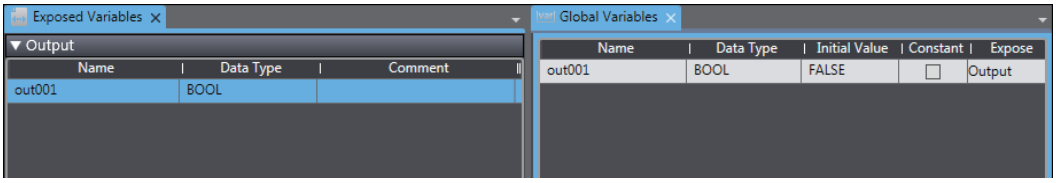
### Registering New Variables on the Exposed Variables Tab Page

Use the following procedure to register variables on the Exposed Variables Tab Page when those variables are not registered as global variables. The registered exposed variables are automatically registered as global variables.

*1* Double-click **Exposed Variables** under **Configurations and Setup** – **Communications** – **Standard** – **Slave I/O**.
The Exposed Variables Tab Page is displayed.

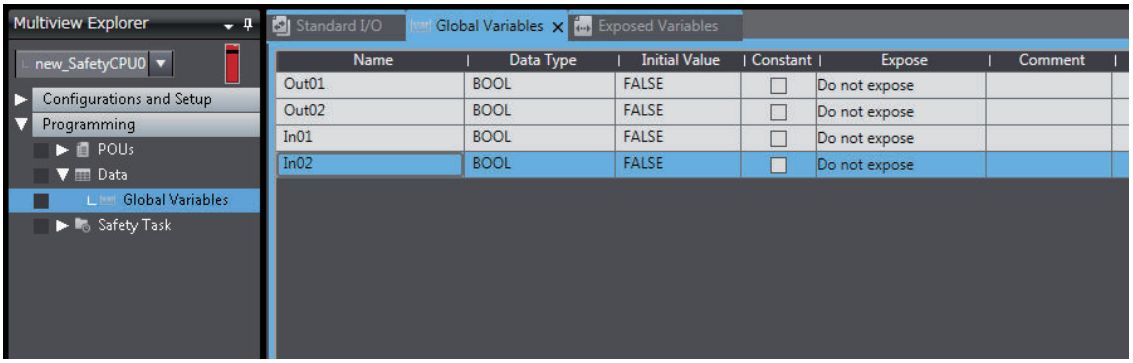


**2** Enter the variable name in the Name Column of the output table or input table.
The variable that you entered is registered in the exposed variable table and in the global variable table.

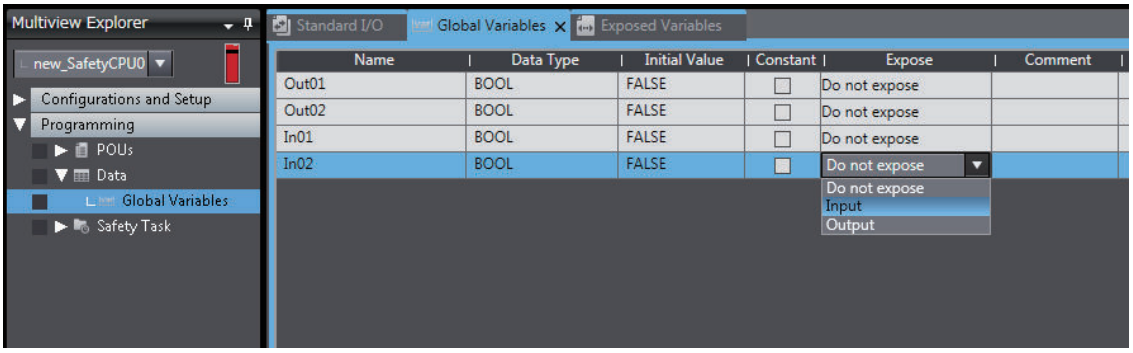



## Setting the Expose Column for Global Variables

**1** Register global variables with standard data types.
Refer to *6-5-3 Registering Variables* on page 6-37 for details on registering variables.



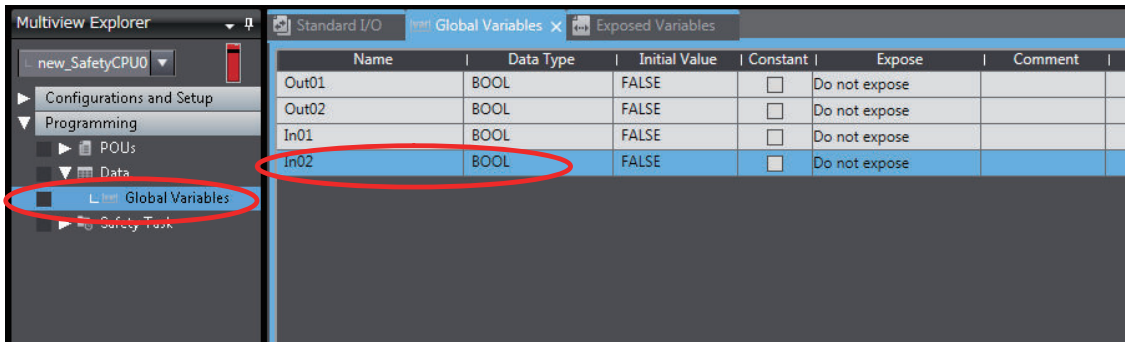

**2** Select the Expose Column for global variables, and then select Input or Output.
The selected variable is registered in the exposed variable input table or output table.



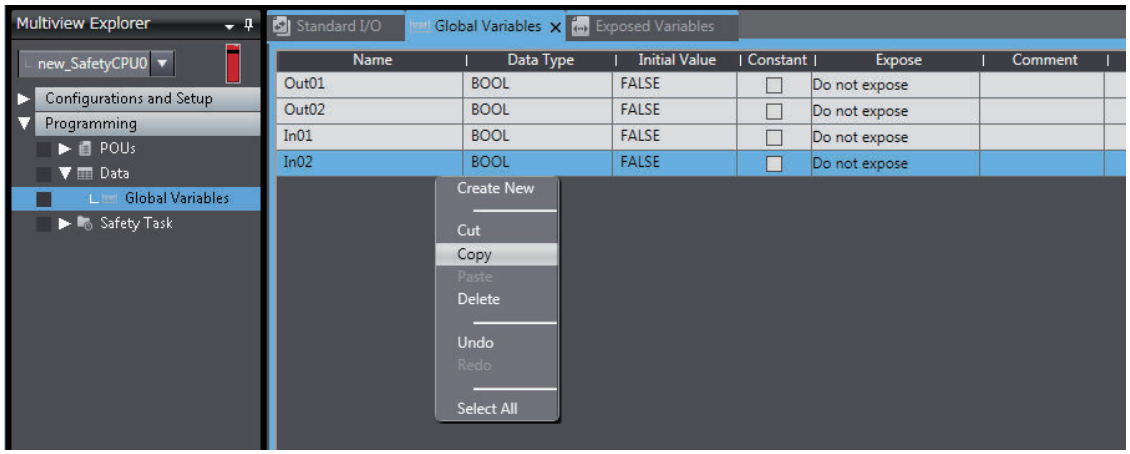

## Copying Global Variables and Pasting Them on the Exposed Variables Tab Page

Use the following procedure to select variables on the global variable table and set them as exposed variables.
You can select more than one global variable.

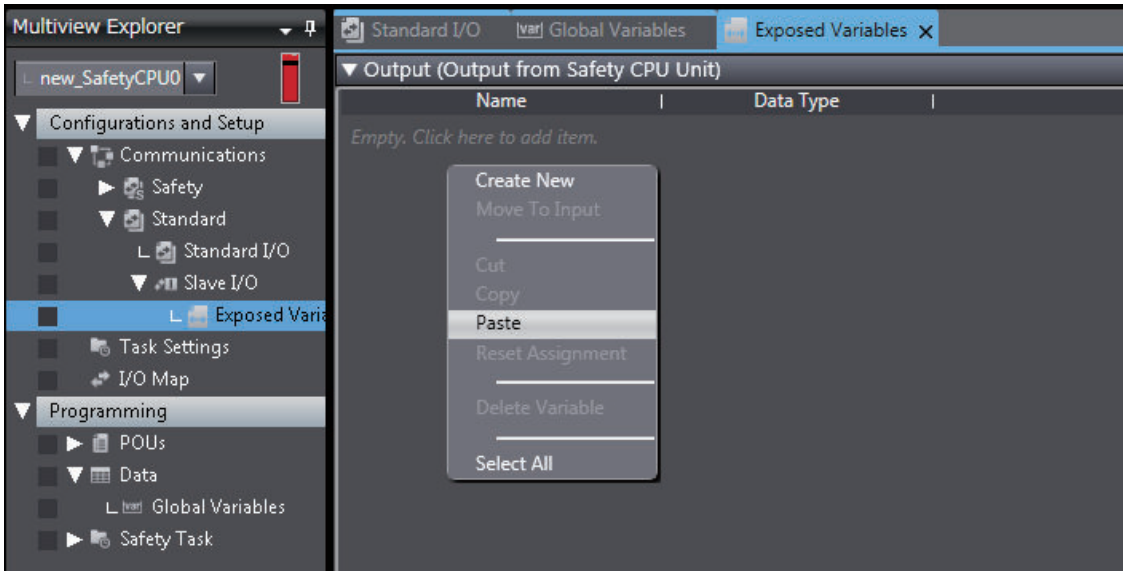
**1** Register global variables with standard data types.
Refer to *6-5-3 Registering Variables* on page 6-37 for details on registering variables.



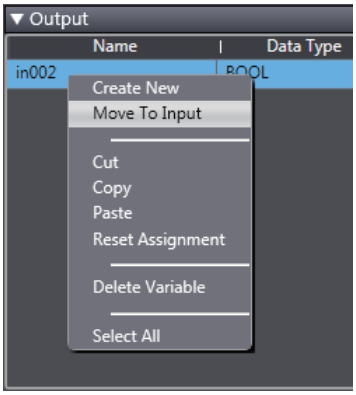

**2** Right-click one or more global variables and select **Copy** from the menu.
The selected global variables are copied.
Press the **Shift** Key or **Ctrl** Key to select more than one global variable.

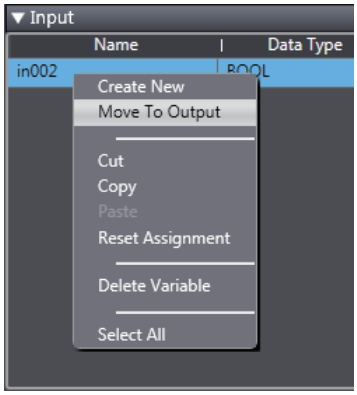



**3** Right-click in the Exposed Variables Tab Page and select **Paste** from the menu.
The global variables are set as exposed variables.




**4** To change the **I/O** setting for an exposed variable, right-click the exposed variable and select **Move To Output** or **Move To Input** from the menu.
Press the **Shift** Key or **Ctrl** Key to select more than one exposed variable.

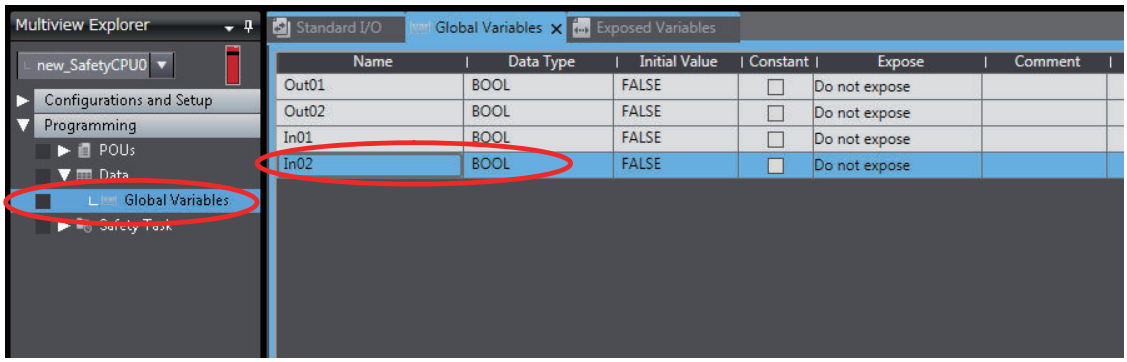


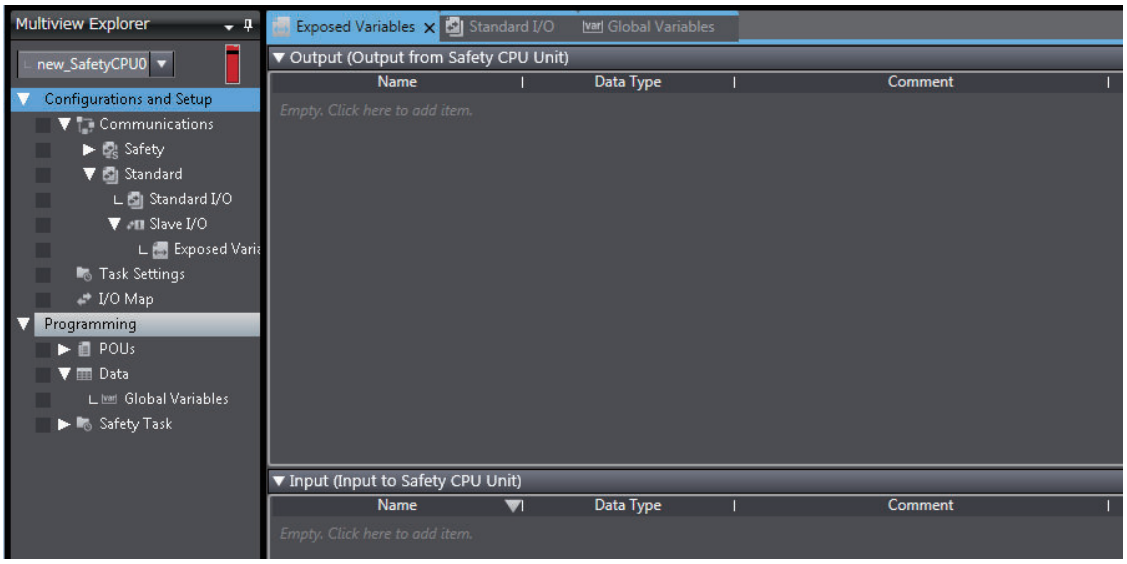

## Selecting Global Variables on the Exposed Variables Tab Page

Use the following procedure to select registered global variables on the Exposed Variables Tab Page and set them as exposed variables.

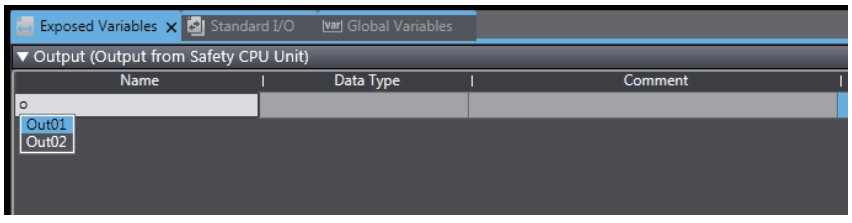
**1** Register global variables with standard data types.
Refer to *6-5-3 Registering Variables* on page 6-37 for details on registering variables.




**2** Double-click **Exposed Variables** under **Configurations and Setup** – **Communications** – **Standard** – **Slave I/O**.
The Exposed Variables Tab Page is displayed.




**3** Enter the name of the variable to expose (the global variable that was registered in step 1) to the NJ/NX-series CPU Unit.
You can also enter the first letter of the global variable in the Name Box to display a list of candidates, and then double-click the desired variable.



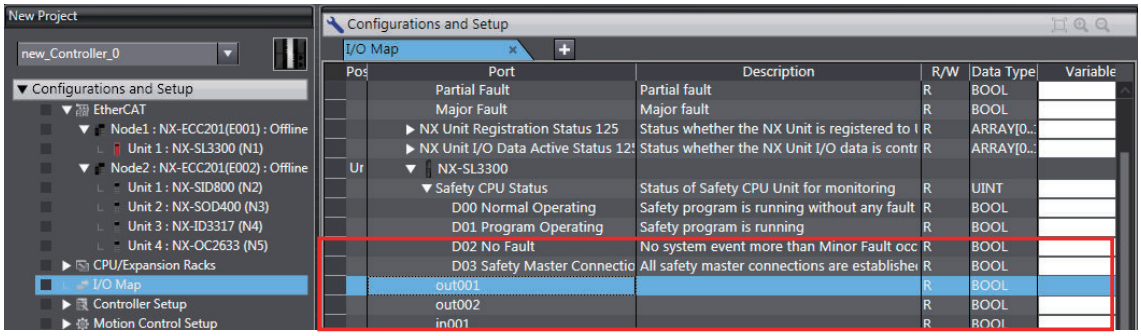

## 5-7-3 Safety CPU Unit Status

If you place a Safety CPU Unit on the NX bus of a NX Bus Master, standard process data communications are performed automatically.

### ● Using an CPU Unit or EtherCAT Coupler Unit

The status of a Safety CPU Unit is displayed as I/O ports in the I/O Map of the NJ/NX-series CPU Unit.

You can use the I/O ports to monitor the status of the Safety CPU Unit from the NJ/NX-series CPU Unit.




Refer to *A-5 Safety CPU Unit Status* on page A-72 for details on the items in the Safety CPU Unit status.

### ● Using an EtherNet/IP Coupler Unit

The Safety CPU Unit status is assigned in the I/O Map of the Communications Coupler Unit. You can check the I/O allocation information of Communications Coupler Unit on the **Edit I/O Allocation Settings Pane**.

You can use this status to monitor the status of the Safety CPU Unit.

## 5-7-4 I/O Ports for Safety I/O Units

You can access the values of the ports for Safety I/O Units from a standard controller.

### ● Using an CPU Unit or EtherCAT Coupler Unit

To access the value of an I/O port that is displayed in the I/O Map of the Safety CPU Unit, assign a variable to the corresponding I/O port in the I/O Map of the NJ/NX-series CPU Unit.

Refer to *A-6 I/O Ports of Safety I/O Units* on page A-74 for descriptions of the I/O ports for Safety I/O Units that are displayed in the I/O Map of the NJ/NX-series CPU Unit.

### ● Using an EtherNet/IP Coupler Unit

I/O port values are allocated in the I/O Map of the Communications Coupler Unit. You can check the I/O allocation information of Communications Coupler Unit on the **Edit I/O Allocation Settings** Pane.

Refer to *A-6 I/O Ports of Safety I/O Units* on page A-74 for descriptions of the I/O ports for Safety I/O Units.

**Additional Information**

You can only read the values of the ports of Safety I/O Units from an NJ/NX-series CPU Unit or other standard controller. You cannot write the values.

## 5-7-5 I/O Refreshing Method

This section describes the I/O refreshing method of the Safety Control Unit.

## ● I/O Refreshing Method

Only Free-Run refreshing can be used for Safety Control Unit.
With Free-Run refreshing, the refresh cycle of the NX bus and the I/O refresh cycle of the NX Units operate asynchronously.
The Safety CPU Unit reads inputs and refreshes outputs according to the safety task period.

# 5-8 Setting Standard Process Data Communications

This section describes procedures for using standard process data communications between a Safety CPU Unit and standard I/O Units.

## 5-8-1 Using an CPU Unit or EtherCAT Coupler Unit

For standard process data communications between the Safety CPU Unit and Safety CPU Unit the standard controller exchanges the data between the Safety CPU Unit and standard I/O Units.
Refer to *5-7 Exposing Variables to Standard Controllers* on page 5-49 for the procedure to expose Safety CPU Unit variables to standard controllers.

## 5-8-2 Using an EtherNet/IP Coupler Unit

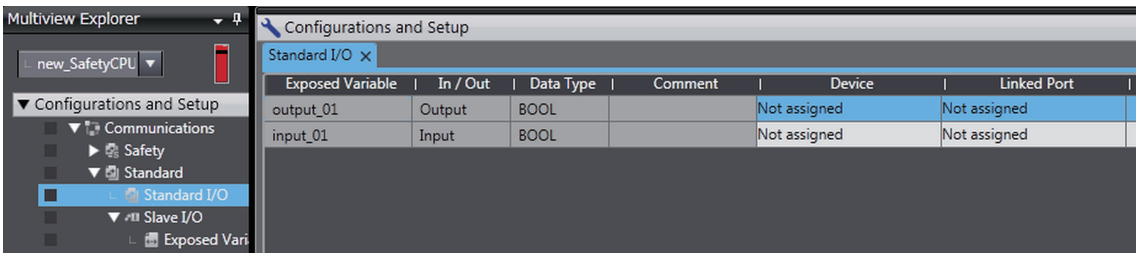
Communications between NX Units are used for standard process data communications between a Safety CPU Unit and standard I/O Units. Communications between NX Units are performed by allocating the data of the standard I/O Units for the exposed variables of the Safety CPU Unit.
Refer to *A-13 Units That Support Communications between NX Units* on page A-101 for the models of the NX Units that you can connect.
Refer to *A-12 I/O Response Times for Communications between NX Units on EtherNet/IP Slave Terminals* on page A-99 for the I/O response times between standard I/O Units and a Safety CPU Unit when using an EtherNet/IP Coupler Unit.

Use the following procedure to set up NX Unit communications between a Safety CPU Unit and standard I/O Units.

1. In the Multiview Explorer, select the target Safety CPU Unit in the Controller Selection Box.

2. Register the exposed variables in the Safety CPU Unit.
   Refer to *5-7-2 Setting Exposed Variables* on page 5-50 for information on registering exposed variables.
   Set the data types of the exposed variables to the same data types as the allocated Standard I/O Unit data.

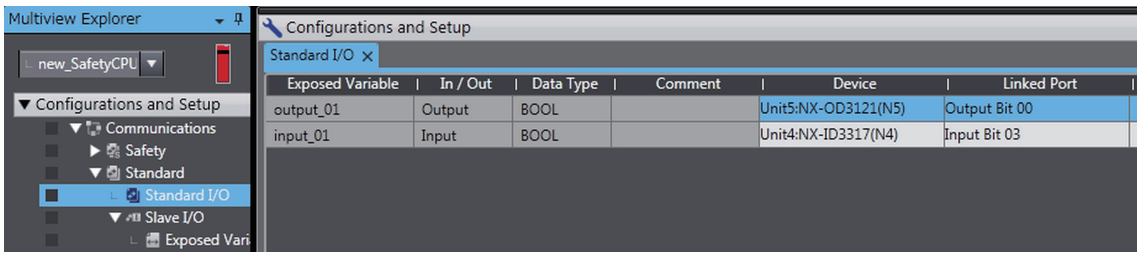
3. Double-click **Standard I/O** under **Configurations and Setup** − **Communications** − **Standard**.
   The Standard I/O Unit Setting Tab Page is displayed.




The meanings of the items in the Standard I/O Unit Setting Tab Page are given below.

| Item | Editing | Description |
|---|---|---|
| Exposed Variables | Not possible. | The registered exposed variables are displayed. |
| Input/Output | Not possible. | Whether the exposed variable is an input or output variable is displayed. |
| Data Type | Not possible. | The data type of the variable is displayed. |
| Comment | Not possible. | The comment set for the exposed variable is set. |
| Device | Possible. | Set the NX Unit to which the exposed variable is allocated. |
| Linked Port | Possible. | Set the NX Unit I/O port to which the exposed variable is allocated. |

**4** Set the devices and ports of the standard I/O Units that correspond to the exposed variables of the Safety CPU Unit.
The I/O data of the standard I/O units will be allocated to the exposed variables of Safety CPU Unit.




### Precautions for Correct Use

- Use different NX Units to set up standard process data communications for the I/O ports used by the standard controller and the I/O ports used by the Safety CPU Unit.
  The I/O data in an NX Unit that is set for communications between NX Units is not allocated I/O data in the EtherNet/IP Coupler Unit.
- The exposed variables of the Safety CPU Unit that is set for communications between NX Units are not allocated I/O data in the EtherNet/IP Coupler Unit.

# 5-9 Exporting/Importing Settings Data

This section describes how to reuse the settings data for the entire Slave Terminal in the Sysmac Studio or the safety application data in the Safety CPU Unit.

You can export and import the data for the entire Slave Terminal or the safety application data in the Safety CPU Unit as a single file.
You use these functions in the following instances.

- When the standard control system and the safety control system are being developed by more than one person and you need to merge the settings for the entire Slave Terminal or the safety application data.
- When you need to reuse the safety application data from another project.
- When you reuse a safety control system created for an EtherNet/IP Slave Terminal on an CPU Rack or EtherCAT Slave Terminal.

You can export or import the four groups of data that are given below.

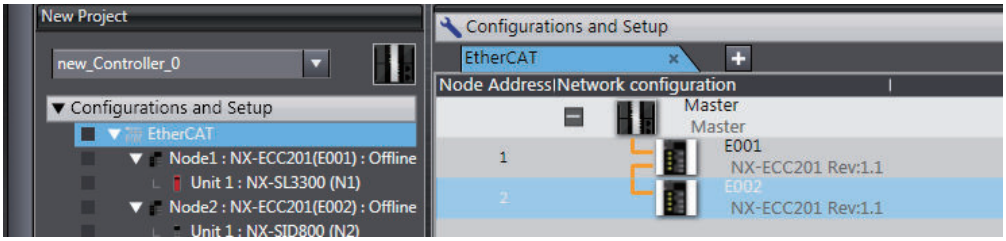
- Settings for Entire Slave Terminal
  The data for the entire Slave Terminal consists of the Slave Terminal configuration information for the Communications Coupler Unit and all NX Units that are connected to that Coupler Unit. It also contains the safety application data.
- All NX Units of the CPU Rack or Slave Terminal
  The settings data for all the NX Units of the CPU Rack or Slave Terminal includes the operating settings and the application data (including safety application data) for all the NX Units, except for the NX bus master.
- Safety application data for Only the Safety CPU Unit
- Safety application data in the Safety Unit Restore File
  The safety application data consists of the safety program and the safety tasks and settings.

## 5-9-1 Exporting/Importing the Settings for the Entire Slave Terminal

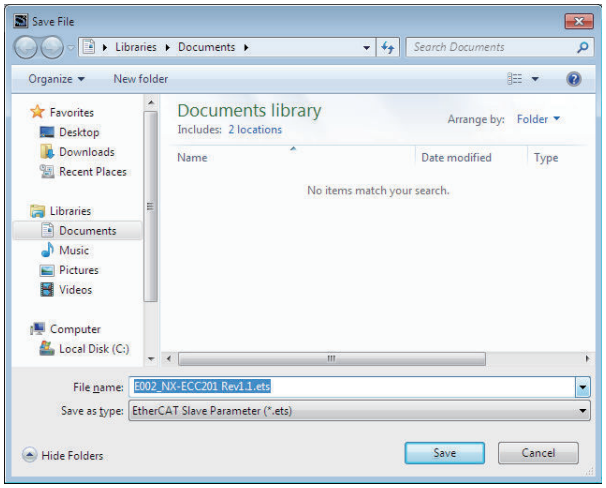
You can export the settings for the entire Slave Terminal into a single file (extension .ets).
The exported settings file for the entire Slave Terminal can be imported to reuse the settings for an entire Slave Terminal with the same settings in a different project on the Sysmac Studio, or a project for which a Safety CPU Unit has not been registered.

The procedure when you use an EtherCAT Coupler Unit is given below as an example.

***1*** Select the NJ/NX-series CPU Unit as the Controller and double-click **EtherCAT** under **Configurations and Setup** in the Multiview Explorer to display the EtherCAT Tab Page.

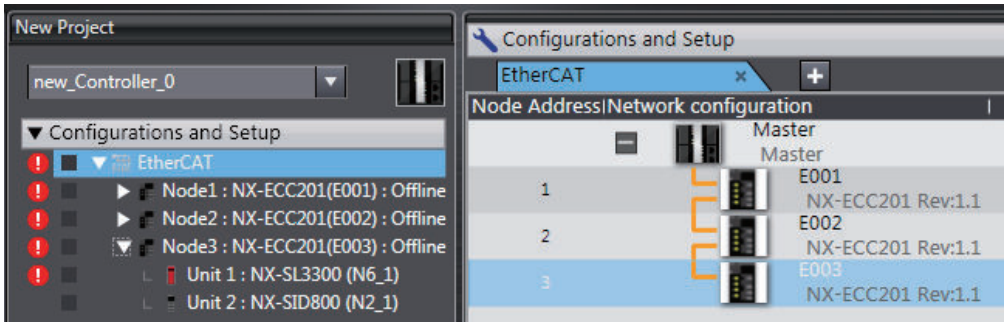


**2** Right-click the target EtherCAT Coupler Unit and select **Export Slave Settings** from the menu.
The **Save File** Dialog Box is displayed.




**3** Enter a file name, and then click the **Save** Button.
An EtherCAT slave parameter file with an .ets extension is saved.

**4** To import a file, select the Unit above the point where you wish to add the slave on the EtherCAT Tab Page, and then right-click and select **Import Slave Settings and Insert New Slave** from the menu.
The EtherCAT Coupler Unit to import is added to the EtherCAT Tab Page.



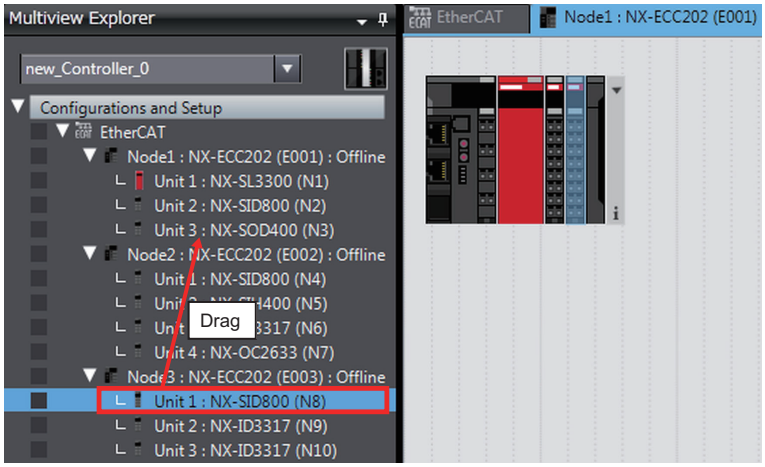

### Precautions for Correct Use

- Use the following import procedure if you are using an EtherCAT Coupler Unit.
  First, import the Slave Terminals that do not contain a Safety CPU Unit and set the device names and node addresses for exporting.
  Then, import the Slave Terminals that contain a Safety CPU Unit and set the device names and node addresses for exporting.
- If importing data results in two or more Safety CPU Units, an error will occur. Delete the Safety CPU Units that are not used.
- To reuse safety programs and variable comments, export them from the Safety CPU Unit, and then import them. This data is included in the safety application data for the Safety CPU Unit.
- You cannot import settings data that contains Safety CPU Unit for which program data protection is set to Sysmac Studio version 1.16 or lower.
- You cannot import settings data that contains a Safety CPU Unit for which user-defined function block data protection is set to Sysmac Studio version 1.11 or lower.

**Additional Information**

If you distribute multiple Safety Control Unit and set the device names and node addresses for exporting.

- Saving the Safety Application Data
  Place all of the Safety Control Unit on one Slave Terminal and export the settings.
  Use the Multiview Explorer to move NX Units between Slave Terminals.

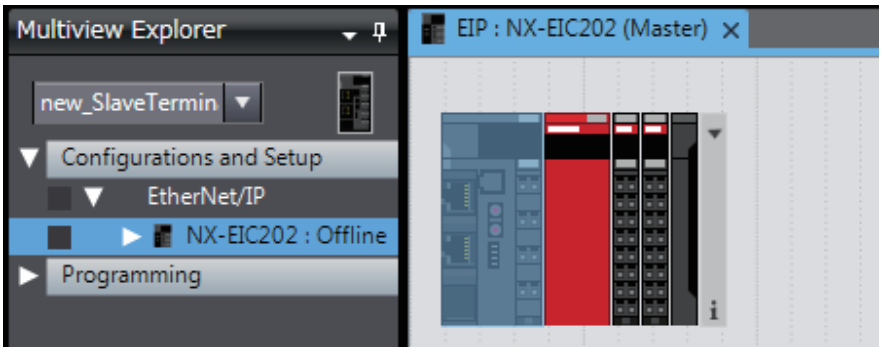



- Reusing the Safety Application Data
  Import the safety control file.
  Move the Safety Control Unit from the Slave Terminal where you imported the file to the other Slave Terminals.
  from the Slave Terminal where you imported the file to the other Slave Terminals.

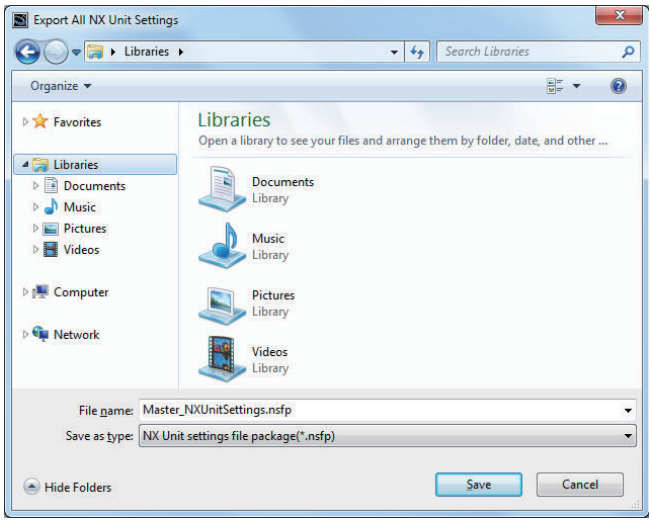
## 5-9-2 Exporting/Importing the Settings for All the NX Units on a CPU Rack or Slave Terminal

You can export the operating settings and the application data (including safety application data) for all of the NX Units, except for the NX bus master, in the CPU Rack settings or the Slave Terminal settings into a single file (extension .nsfp).

**1** Display the CPU Rack or Slave Terminal Tab Page where the Safety CPU Unit to export is configured.




**2** Right-click the CPU Unit or the Communications Coupler Unit and select **Export All NX Unit Settings** from the menu.
The **Export All NX Unit Settings** Dialog Box is displayed.




**3** Enter a file name, and then click the **Save** Button.
A configuration file for all the NX Units with an .nsfp extension is saved.

**4** To import a file, select the CPU Unit on the Edit CPU Rack Configuration Tab Page or select the Communications Coupler Unit on the Slave Terminal Tab Page. Then, select **Import All NX Unit Settings** from the menu that appears when you click the right mouse button.
All NX Units including the Safety CPU Unit to import are added to the CPU Rack or the Slave Terminal Tab Page.

### Precautions for Correct Use

- If even one NX Unit has been added to the Slave Terminal that is being imported, then the Slave Terminal cannot be imported. Delete all of the NX Units, and then execute the import.
- If you import a settings file for all NX Units including a Safety CPU Unit that has standard I/O settings on an EtherNet/IP Slave Terminal, the standard I/O settings data is not restored.

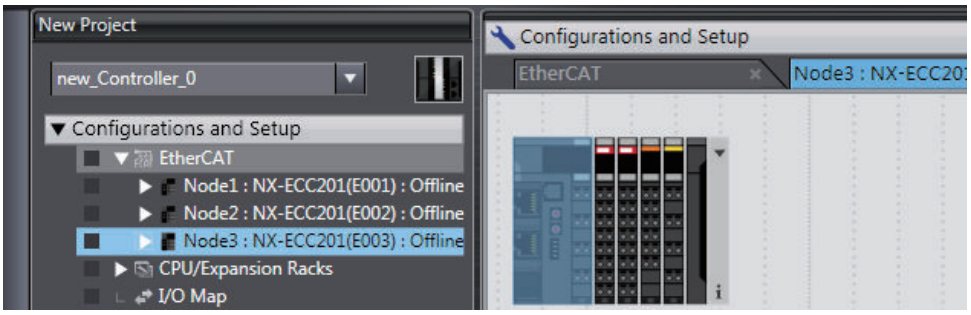
### Version Information

Exporting and importing all NX Unit settings is supported by Sysmac Studio version 1.17 or higher.

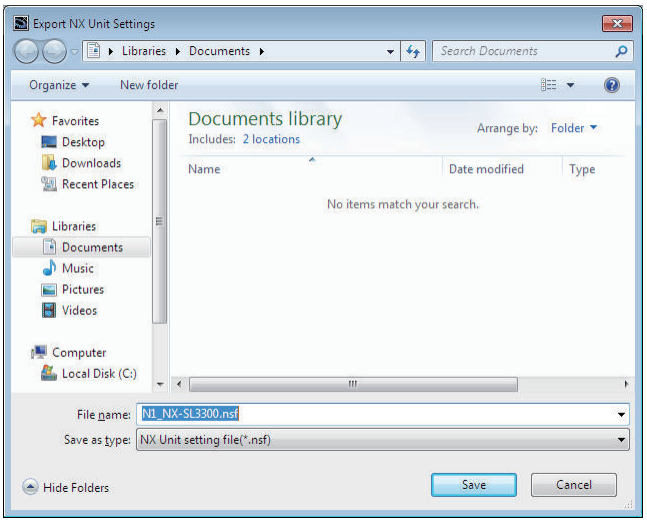
## 5-9-3 Exporting/Importing Data for Individual Safety CPU Unit

You can export and import the safety application data of each Safety CPU Unit as a single file (extension .nsf).
The exported Safety CPU Unit settings file can be imported to reuse the safety application data for a Safety CPU Unit with the same settings. To do this, go into the Edit Slave Terminal Configuration Tab Page in a different project on the Sysmac Studio, or a project for which a Safety CPU Unit has not been registered.

**1** Display the Slave Terminal Tab Page where the Safety CPU Unit to export is configured.




**2** Right-click the Safety CPU Unit to export and select **Export NX Unit Settings** from the menu.
The **Export NX Unit Settings** Dialog Box is displayed.




**3** Enter a file name, and then click the **Save** Button.
An NX Unit configuration file with an .nsf extension is saved.

**4** To import a file, select the Unit to the left of the point where you wish to add the slave on the Slave Terminal Tab Page, and then right-click and select **Import Slave Settings and Insert New Slave** from the menu.
The Safety CPU Unit to import is added to the Slave Terminal Tab Page.

**Precautions for Correct Use**

- When you import the data to the Safety CPU Unit, first configure all of the Safety I/O Units.
- If importing data results in two or more Safety CPU Units, an error will occur. Delete the Safety CPU Units that are not used.
- You cannot import settings data that contains Safety CPU Unit for which program data protection is set to Sysmac Studio version 1.16 or lower.
- You cannot import settings data that contains Safety CPU Unit for which user-defined function block data protection is set to Sysmac Studio version 1.11 or lower.

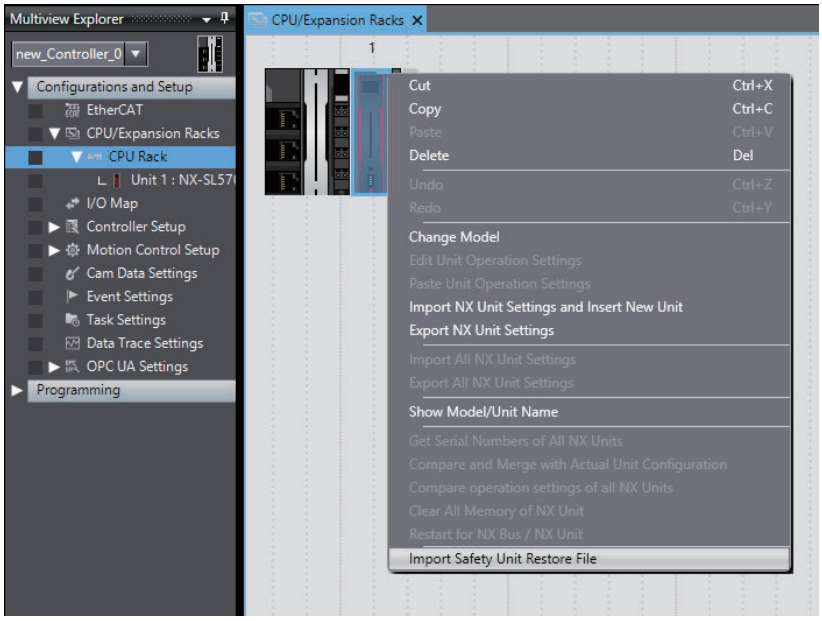
## 5-9-4 Importing the Safety Unit Restore File

You can import safety application data stored in the Safety Unit Restore File into the Safety CPU Unit in the project.

For details on the Safety Unit Restore File, refer to *Section 10 Safety Unit Restore* on page 10-1.

**1** Open the CPU and Expansion Racks Tab Page where the Safety CPU Unit is placed.

**2** Right-click the Safety CPU Unit and select **Import Safety Unit Restore File** from the menu.




**3** Select a Safety Unit Restore File to import and click the **Open** button.
The safety application data of the Safety CPU Unit is replaced with the safety application data stored in the Safety Unit Restore File.

### Precautions for Correct Use

- You cannot import the file if the format and the unit version of the Safety CPU Unit in the Safety Unit Restore File are not consistent with those for the project.
- If the safety application data having the identical name as the safety application data in the Safety Unit Restore File exists in another Safety CPU Unit in the project, you cannot import the file.

# 5-10 Offline Comparison

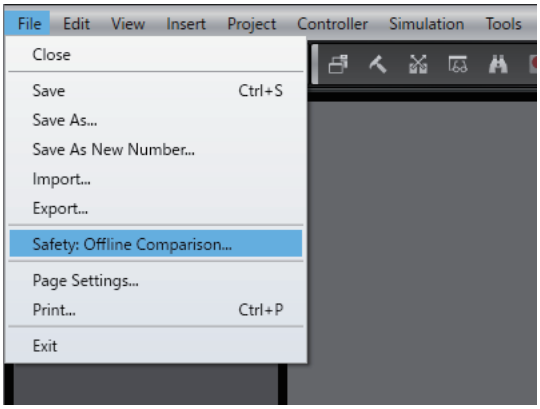
This section describes the function for comparing the safety application data between the currently opened project and another project file.
This function can be used for managing changes in safety application data. In addition, target data can be reflected in the current project if the data supports detailed comparison.

## 5-10-1 Procedure for Offline Comparison

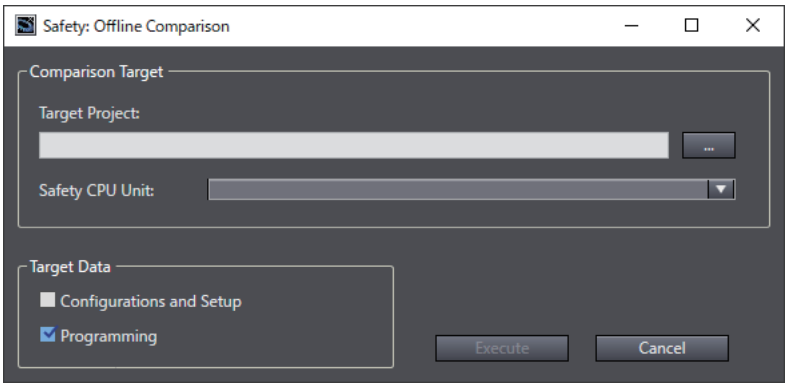
1 Start Sysmac Studio and open the source project file.

2 Select Safety CPU Unit from the Controller Selection Box in the Multiview Explorer and switch to the **Safety CPU Unit Setup and Programming** View.

3 From the Controller Menu, select **File** – **Safety: Offline Comparison**.




The comparison target setting window for offline safety comparison is displayed.

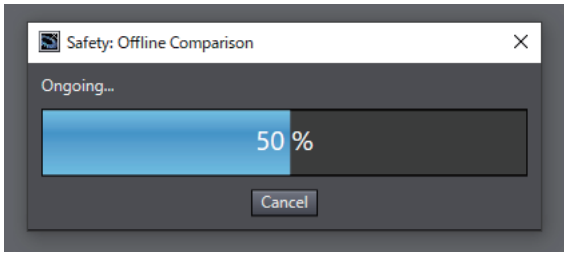



The contents of the setting window are as follows.

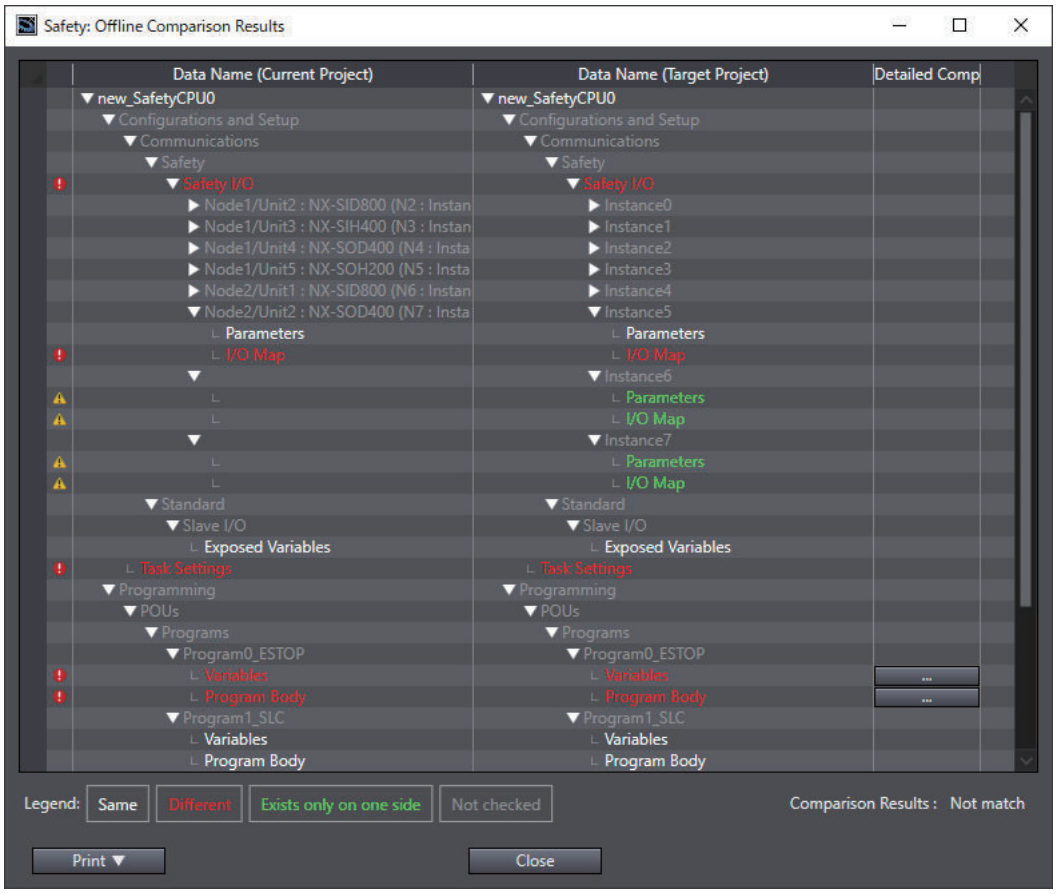
| Item | Description |
|---|---|
| Target Project | A project file to compare with the currently opened project. Select a previously exported project file (smc2, csm2, smc, or csm). |
| Safety CPU Unit | Select Safety CPU Unit to compare. |

| Item | Description |
|---|---|
| Target Data | As data to compare, you can select *Configurations and Setup* or *Programming*. By default, *Programming* is selected.<br>Refer to *5-10-4 Target Data of Offline Comparison* on page 5-69 for details of target data. |

**4** Select the target project, Safety CPU Unit to be compared, and target data, and then click the **Execute** Button.
The target data is loaded. The following progress dialog appears during offline comparison.
Depending on the project size, it may take time until the comparison results window appears.

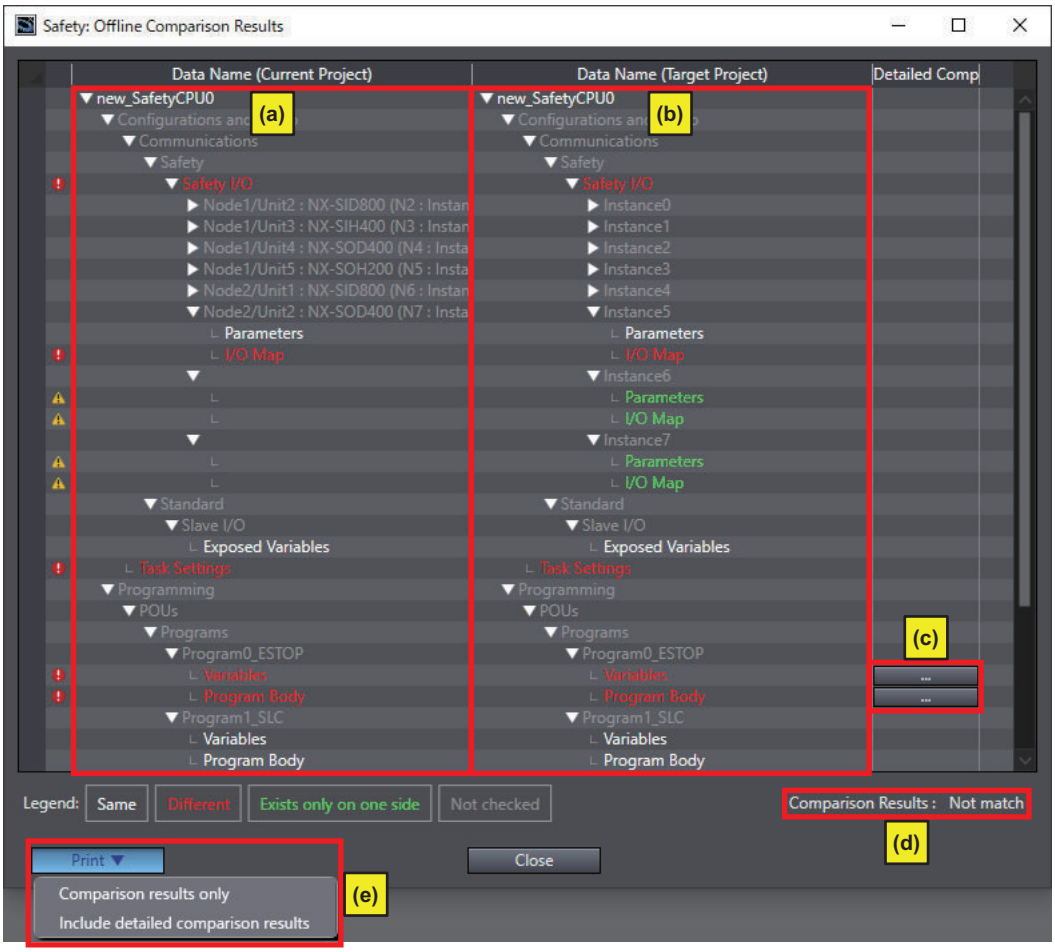



The offline comparison results are displayed.




## 5-10-2 Checking the Comparison Results

This section describes the displayed contents of the offline comparison results.




The contents of this window are as follows.

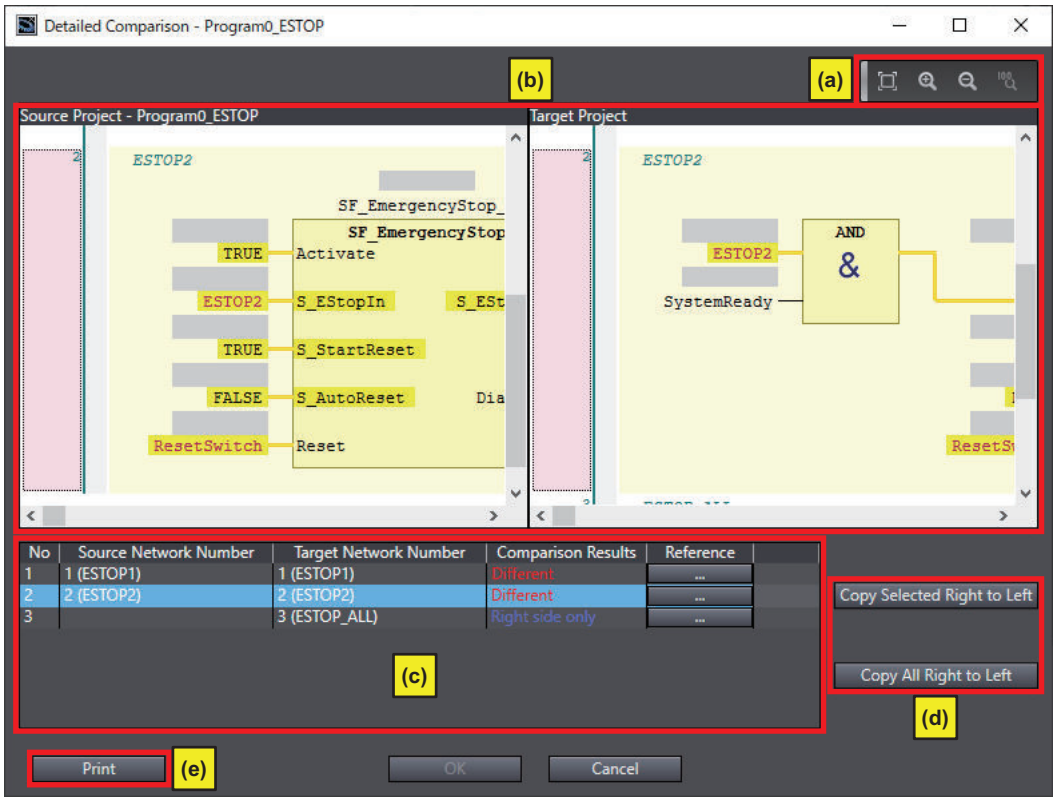
| Letter | Name | Description |
|---|---|---|
| (a) | Current Project | Displays the source project in a tree format. |
| (b) | Target Project | Displays the target project in a tree format. |
| (c) | Detailed Comparison Button | Displays the detailed comparison window for data that has any differences in the comparison results AND that supports detailed comparison. Refer to *5-10-3 Detailed Comparison* on page 5-67 for details. |
| (d) | Comparison Results | Displays the comparison results for safety application data. |
| (e) | Print Button | Prints the comparison results. The operations of each button are as follows.<br><br>Comparison results only:<br>Only prints the results of comparison between source and target project trees.<br>Include detailed comparison results<br>In addition to the comparison results, prints the detailed comparison results for data that includes detailed comparison. |

Refer to *5-10-4 Target Data of Offline Comparison* on page 5-69 for details of target data.

## 5-10-3 Detailed Comparison

Clicking the **Detailed Comparison** button on Offline Comparison Results displays the following detailed comparison window. From the detailed comparison window, you can reflect differences in the source project.

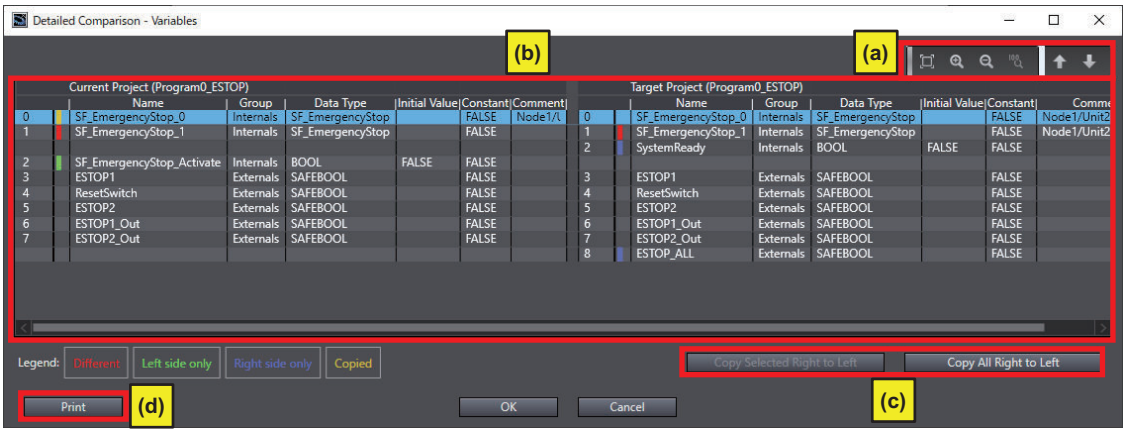
## Safety Programs




The contents of this window are as follows.

| Letter | Name | Description |
|---|---|---|
| (a) | Zoom in/out Button | Zooms in/out on the program comparison window. |
| (b) | Program Comparison Window | Displays the source and target programs. |
| (c) | Program Comparison Results List | Displays differences against each program network. Click the **Reference** button to display differences on the program comparison window. |
| (d) | Difference Copy Button | Copies the differences selected on the program comparison results window to the source project.<br>After **OK** is clicked on the Detailed Comparison window, it will be reflected in the project. |
| (e) | Print Button | Prints the window of the detailed comparison results. |

## Variable Table




The contents of this window are as follows.

| Letter | Name | Description |
|---|---|---|
| (a) | Zoom in/out and Jumping to Differences Button | Used to zoom in/out on the variable table comparison window and to jump to differences. |
| (b) | Variable Table Comparison Window | Displays the results of variable table comparison between the source and target projects. |
| (c) | Difference Copy Button | Copies the differences selected on the variable table comparison display to the source project.<br>After **OK** is clicked on the Detailed Comparison window, it will be reflected in the project. |
| (d) | Print Button | Prints the window of the detailed comparison results. |

### Precautions for Correct Use

- After a program or variable is reflected in the current project using Offline Comparison's detailed comparison function, the safety program in the project will be in an unvalidated state. Before you perform safety validation of the safety programs, complete debugging of the safety programs.
- If a safety program that exists only in the target project is reflected in the source project, the program assignment setting (execution order) will be registered at the end. Make sure that the program execution order is correct in the task settings.
- Detailed Comparison does not compare safety programs that exist only in the source program.

## 5-10-4 Target Data of Offline Comparison

The following data is compared by Offline Comparison. Data not included in the safety application is not compared.

| Displayed name | Target data | Detailed Comparison |
|---|---|---|
| **SafetyCPU Device Name** | • Safety Signature<br>• Project options, etc. | Not applicable |

| Displayed name | Target data | Detailed Comparison |
|---|---|---|
| **Configurations and Setup**<br>**Communications**<br>**Safety**<br>Safety I/O | • Whether FSoE communications settings are enabled or disabled<br>• Whether FSoE Watchdog Timer is set to auto setting or not<br>• Slot comment settings for safety slave unit parameters | Not applicable |
| **Configurations and Setup**<br>**Communications**<br>**Safety**<br>**Safety I/O**<br>**Node/Unit**<br>Safety Slave Unit Parameters | • Safety slave unit parameter settings<br>• Settings for FSoE Slave Address<br>• Settings for FSoE Watchdog Timers | Not applicable |
| **Configurations and Setup**<br>**Communications**<br>**Safety**<br>**Safety I/O**<br>**Node/Unit**<br>SRA Parameters | • SRA parameter settings for the R88D-1SA series<br>• Settings for FSoE Slave Address<br>• Settings for FSoE Watchdog Timers | Not applicable |
| **Configurations and Setup**<br>**Communications**<br>**Safety**<br>**Safety I/O**<br>**Node/Unit**<br>I/O Map | • I/O map settings of an FSoE slave | Not applicable |
| **Configurations and Setup**<br>**Communications**<br>**Safety**<br>**EtherNet/IP Safety Connection Settings**<br>Safety Network Number | • Safety network number settings for CIP Safety | Not applicable |
| **Configurations and Setup**<br>**Communications**<br>**Safety**<br>**EtherNet/IP Safety Connection Settings**<br>**Connection Settings (Originator)**<br>**CIP Safety Device**<br>CIP Safety Parameters | • Connection settings for CIP Safety | Not applicable |
| **Configurations and Setup**<br>**Communications**<br>**Safety**<br>**EtherNet/IP Safety Connection Settings**<br>**Connection Settings (Originator)**<br>**CIP Safety Device**<br>I/O Map | • I/O map settings for a CIP Safety device | Not applicable |
| **Configurations and Setup**<br>**Communications**<br>**Safety**<br>**EtherNet/IP Safety Connection Settings**<br>**Connection Settings (Originator)**<br>I/O Assembly Settings (Target) | • I/O assembly settings for CIP Safety | Not applicable |

| Displayed name | Target data | Detailed Comparison |
|---|---|---|
| **Configurations and Setup**<br>**Communications**<br>**Safety**<br>**EtherNet/IP Safety Connection Settings**<br>**Connection Settings (Originator)**<br>**I/O Assembly Settings (Target)**<br>I/O Map | • I/O assembly settings for a CIP Safety I/O assembly | Not applicable |
| **Configurations and Setup**<br>**Communications**<br>**standard**<br>Standard I/O | • Assignment settings for standard I/O and exposed variables | Not applicable |
| **Configurations and Setup**<br>**Communications**<br>**standard**<br>**Slave I/O**<br>Exposed Variables | • Settings for exposed variables | Applicable |
| **Configurations and Setup**<br>**Task Setup** | • Safety task period<br>• Program assignment settings | Not applicable |
| **Programming**<br>**POU**<br>**Programs**<br>**Program names**<br>Variable tables | • Program local variables (internal variables, external variables) | Applicable |
| **Programming**<br>**POU**<br>**Programs**<br>**Program names**<br>Program Body | • Program body (algorithm) | Applicable |
| **Programming**<br>**POU**<br>**Function blocks**<br>**Function block names**<br>Variable tables | • Function block local variables (internal variables, I/O variables) | Applicable |
| **Programming**<br>**POU**<br>**Function blocks**<br>**Function block names**<br>Program Body | • Function block program body (algorithm) | Applicable |
| **Programming**<br>**Data**<br>Global variable | • Global variables (including exposed variables) | Applicable |

# 6

# Programming

This section describes variables, instructions, and other elements that are used to create safety programs. It also describes the programming operations that are used on the Sysmac Studio.


**6-1 POUs (Program Organization Units) ..... 6-3**
6-1-1 POU ..... 6-3
6-1-2 Overview of the Three Types of POUs ..... 6-3
6-1-3 Differences between Programs, Functions, and Function Blocks ..... 6-4
6-1-4 Details on Programs ..... 6-5
6-1-5 Details on Function Blocks ..... 6-6
6-1-6 Details on Functions ..... 6-10
6-1-7 Instructions ..... 6-11

**6-2 Variables ..... 6-12**
6-2-1 Variables ..... 6-12
6-2-2 Types of Variables ..... 6-12
6-2-3 Type of User-defined Variable ..... 6-12
6-2-4 Attributes of Variables ..... 6-13
6-2-5 Data Type ..... 6-14
6-2-6 Variable Attributes Other Than Data Type ..... 6-16
6-2-7 Function Block Instances ..... 6-17
6-2-8 Restrictions on Variable Names and Other Safety Program-related Names. 6-17

**6-3 Constants (Literals) ..... 6-19**
6-3-1 Constants ..... 6-19
6-3-2 Types of Constants ..... 6-19

**6-4 Programming Languages ..... 6-21**
6-4-1 Programming Languages ..... 6-21
6-4-2 FBD Language ..... 6-21

**6-5 Programming Operations ..... 6-26**
6-5-1 Programming Layer on the Sysmac Studio ..... 6-26
6-5-2 Registering POUs ..... 6-27
6-5-3 Registering Variables ..... 6-37
6-5-4 FBD Programming ..... 6-45
6-5-5 Program Pattern Copy ..... 6-66
6-5-6 Function Block Conversion for Programs ..... 6-69
6-5-7 Building ..... 6-70
6-5-8 Searching and Replacing ..... 6-72
6-5-9 Safety Task Settings ..... 6-75
6-5-10 Variable Comment Switching Function ..... 6-77

**6-6 Automatic Programming ..... 6-86**

6-6-1 Generation Algorithms for Automatic Programming ..... 6-86
6-6-2 Automatic Programming Settings ..... 6-89
6-6-3 Automatic Programming Execution Procedure ..... 6-92

**6-7 Monitoring Memory Usage for Safety Control Unit ..... 6-96**

**6-8 Offline Debugging ..... 6-98**
6-8-1 Offline Safety Program Debugging ..... 6-98
6-8-2 Monitoring ..... 6-101
6-8-3 Controlling BOOL Variables, Changing Present Values, and Using Forced Refreshing ..... 6-101
6-8-4 Cross References ..... 6-101
6-8-5 Setting the Initial Values of Variables ..... 6-101
6-8-6 Feedback Settings ..... 6-102
6-8-7 Simple Automatic Test ..... 6-103

# 6-1 POUs (Program Organization Units)

The safety program that runs on a Safety CPU Unit is made from a combination of POUs (program organization units).
This section describes the configuration and specifications of POUs.
Refer to *6-5 Programming Operations* on page 6-26 for the procedures to create POUs on the Sysmac Studio.
Refer to *Sysmac Studio Version 1 Operation Manual (Cat. No. W504)* for the procedures to create POUs that are used with an NJ/NX-series CPU Unit.



## 6-1-1 POU

A POU (program organization unit) is a unit that is defined in the IEC 61131-3 user program execution model. A POU includes a local variable table and an algorithm (i.e., a series of code or logic). It is the basic unit used to build the safety program.
You combine POUs to build a complete safety program.
There are three types of POUs, as described below.

- Programs
  A program corresponds to a main routine. It is the main type of POU that is used for algorithms. You can place any instruction, function, or function block in the algorithm of a program.
- Function Blocks ("FBs")
  A function block can output different values even with the same inputs. Function blocks are executed when they are called from a program or another function block.
- Functions ("FUNs")
  A function always outputs the same values for the same inputs. Functions are executed when they are called from a program, another function, or a function block.

The POU consists of a combination of these three types of POUs. You can create many POUs.
You assign the created programs to a safety task to execute them. Only one safety task can be used by the Safety CPU Unit.

## 6-1-2 Overview of the Three Types of POUs

### Program

#### ● Executing Programs and Execution Conditions

- You execute a safety task to execute the programs that are assigned to that safety task.
- Programs are always executed.

#### ● Notation

- The POUs must include at least one program. More than one program can be assigned to the safety task.

## Function Blocks (FBs)

### ● Executing Function Blocks and Execution Conditions

- You can call function blocks from programs or other function blocks to execute them.
- Function blocks are always executed.
- To execute a function block for only specific conditions, pass a TRUE value to the *Activate* input variable of that function block. The function block is not executed if the value of the *Activate* input variable is FALSE.

### ● Notation

- There are both user-defined function blocks and system-defined function blocks.
  User-defined function blocks are sometimes called "user-defined FBs". System-defined function blocks are sometimes called "FB instructions".

For details on function blocks, refer to *6-1-5 Details on Function Blocks* on page 6-6.

## Functions (FUNs)

### ● Executing Function and Execution Conditions

- You can call functions from programs or other function blocks to execute them.
- Functions are always executed.

### ● Notation

- You cannot create user-defined functions.
- System-defined functions are sometimes called "FUN instructions".
- The values of internal variables are not retained. The output value remains constant as long as input value is constant.

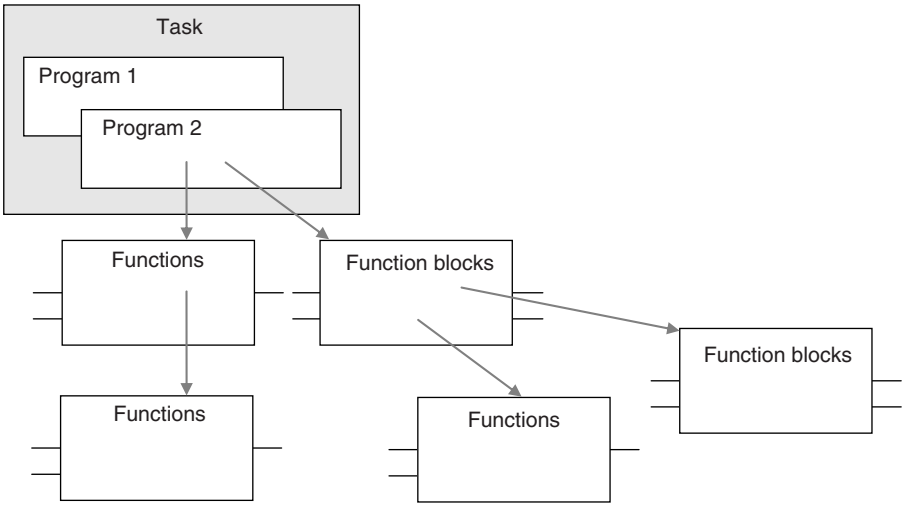
For details on functions, refer to *6-1-6 Details on Functions* on page 6-10.

## 6-1-3 Differences between Programs, Functions, and Function Blocks

| POU type / Item | | Programs | Function blocks | Functions |
|---|---|---|---|---|
| **Types** | | User-defined only | Instructions or user-defined | Instructions only (User-defined functions not supported.) |
| **Execution method** | | Executed upon execution of the safety task. | Called from a program or another function block. | Called from a program or function block. |
| **Algorithm** | **Any instructions** | Supported. | Supported. | Not supported. |
| | **User-defined function blocks** | Supported. | Supported. | Not supported. |

| POU type / Item | Programs | Function blocks | Functions |
|---|---|---|---|
| **Execution condition** | Always executed. | Always executed. Specify the execution condition with an input variable. | Always executed. |

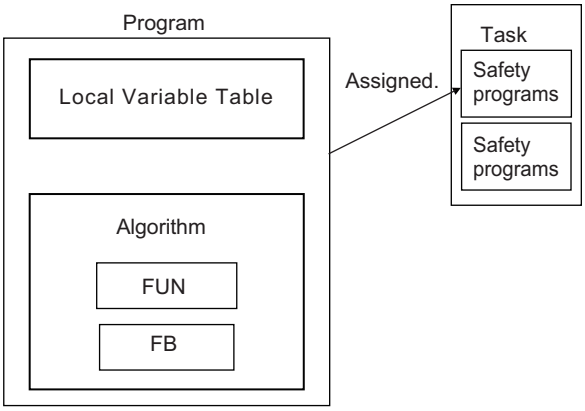
The hierarchical relationships between programs, functions, and function blocks are shown in the following figure.




## 6-1-4 Details on Programs

### Program Structure

Programs consist of a local variable table and an algorithm.
The algorithm is programmed in the FBD language.
You can use any instructions or user-defined function blocks in the algorithm.




### Program Execution Conditions

Programs are executed when the safety task they are assigned to is executed.

### ● Order of Execution

You can set the order of execution of all programs in a safety task.
You set this order in the **Program Assignment Settings** Display of the **Task Settings** Tab Page on the Safety CPU Unit Setup and Programming View on the Sysmac Studio. Refer to *6-5 Programming Operations* on page 6-26 for programming operations.

## 6-1-5 Details on Function Blocks

You can use system-defined function blocks (instructions) and user-defined function blocks in the Safety Control Unit.

### Procedure to Create Function Blocks

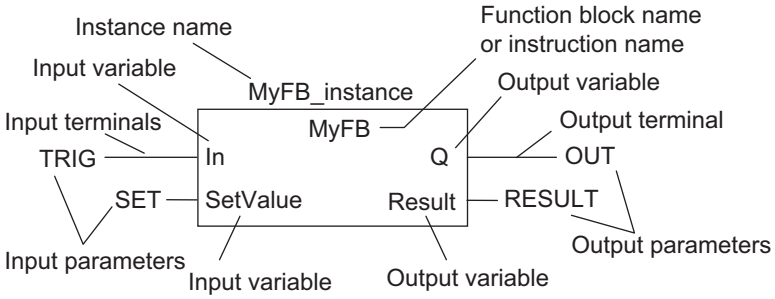
A function block consists of a "function block definition" that is made in advance and "instances" that are used in the actual programs.
Create function blocks in the following order.

***1*** Create the function block definition.
Create the algorithm.

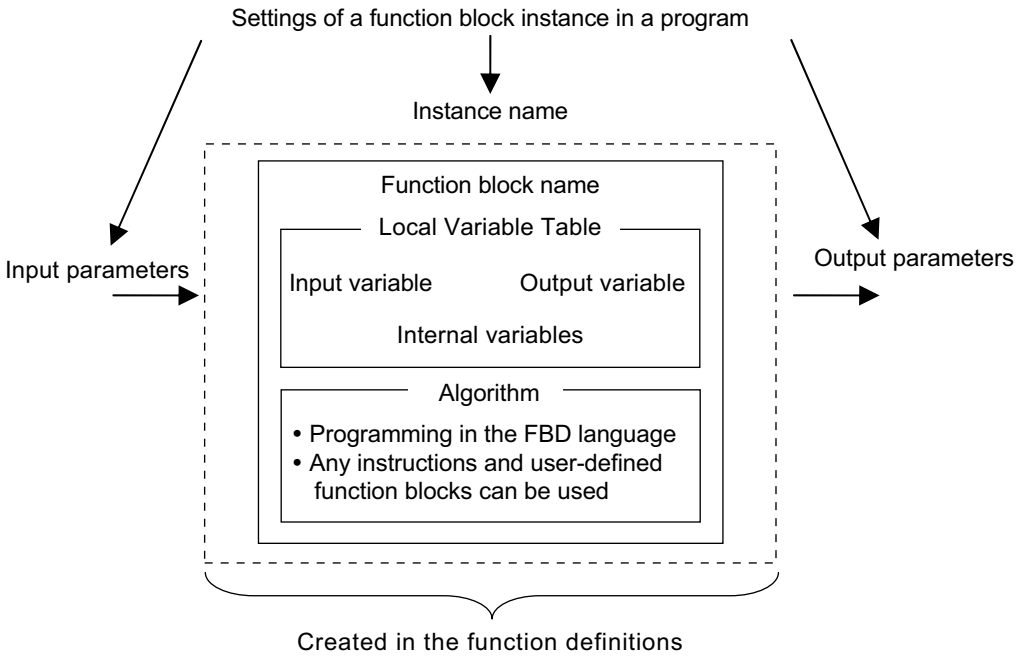
***2*** Place an instance of the function block definition in the program.
Call the function block definition from a program or another function block. You can call the same function block definition from more than one program or function block. After you place an instance of a function block definition in a program or in another function block, you can manipulate and execute it as an independent entity.

### Structure of Function Blocks

With the FBD language, function blocks are represented as rectangular boxes as shown below.
Function blocks consist of the following parts.




- Function Block Settings
  When you create an instance of a function block definition, make the following settings.




## ● Function Block Name or Instruction Name

This is the name of the user-defined function block or the instruction.

## ● Instance Name

You give an instance name to a function block instance in a program to enable managing it.
You specify an instance name when you call a function block definition from a program or another function block.

## ● Algorithm

Algorithms are programmed in the FBD language.
You cannot use the ladder diagram language (LD) or the structured text language (ST).
You can use any instructions or user-defined function blocks in the algorithm.

## ● Local Variable Table

The local variable table contains the definitions for input variables, output variables, and internal variables.

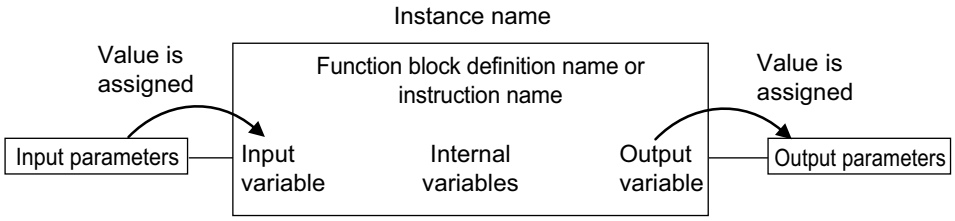
## ● Parameters

- Input Parameters to Input Variables
  An input parameter passes a value to an input variable in a function block when function block execution begins. An input parameter can be either a variable or a constant.
- Output Parameters from Output Variables
  An output parameter receives a value from an output variable in a function block when function block execution is completed. A variable is given as the parameter.

**Additional Information**

You can omit input and output parameters. For details on the operation, refer to the *NX-series Safety Control Unit Instructions Reference Manual (Cat. No. Z931)*.

## Variable Designations for Function Blocks




The specifications for variables in function blocks are given below.

| Variables | Number*1 | Specification |
|---|---|---|
| **Input variables** | 1 to 64 | Input variables are used as input arguments within the function block. They cannot be changed inside the function block. |
| | | • When the function block is executed, the input variables are set to the values of the input parameters.<br>• You can specify either constants or variables for input parameters.<br>• Omitting Input Parameters:<br>Refer to the *NX-series Safety Control Unit Instructions Reference Manual (Cat. No. Z931)*. |
| **Output variables** | 1 to 64 | Output variables are used as output arguments from the function block. |
| | | • The output parameters are set to the values of the output variables at the end of execution.<br>• You cannot specify a constant for an output parameter. Only variables may be specified.<br>• You can omit output parameter connections. If you omit an output parameter, the value of the output variable is not assigned to any parameter.<br>• Omitting Output Parameters:<br>Refer to the *NX-series Safety Control Unit Instructions Reference Manual (Cat. No. Z931)*.<br>• You can access the values from outside of the function block. Access these values with the following format: *InstanceName.OutputVariableName*. However, you cannot write values directly to an output variable. |
| **Internal variables** | No limit | Internal variables are used for temporary storage within a function block. |
| | | • The values of internal variables are retained regardless of whether the function block is executed.<br>• The values cannot be referenced from outside of the function block. |

*1. The individual restrictions are listed in the above table. The actual upper limits depend on the overall program capacity and internal memory capacity.

Refer to *6-2-4 Attributes of Variables* on page 6-13 for details on the variable attributes that can be set for each type of variable.

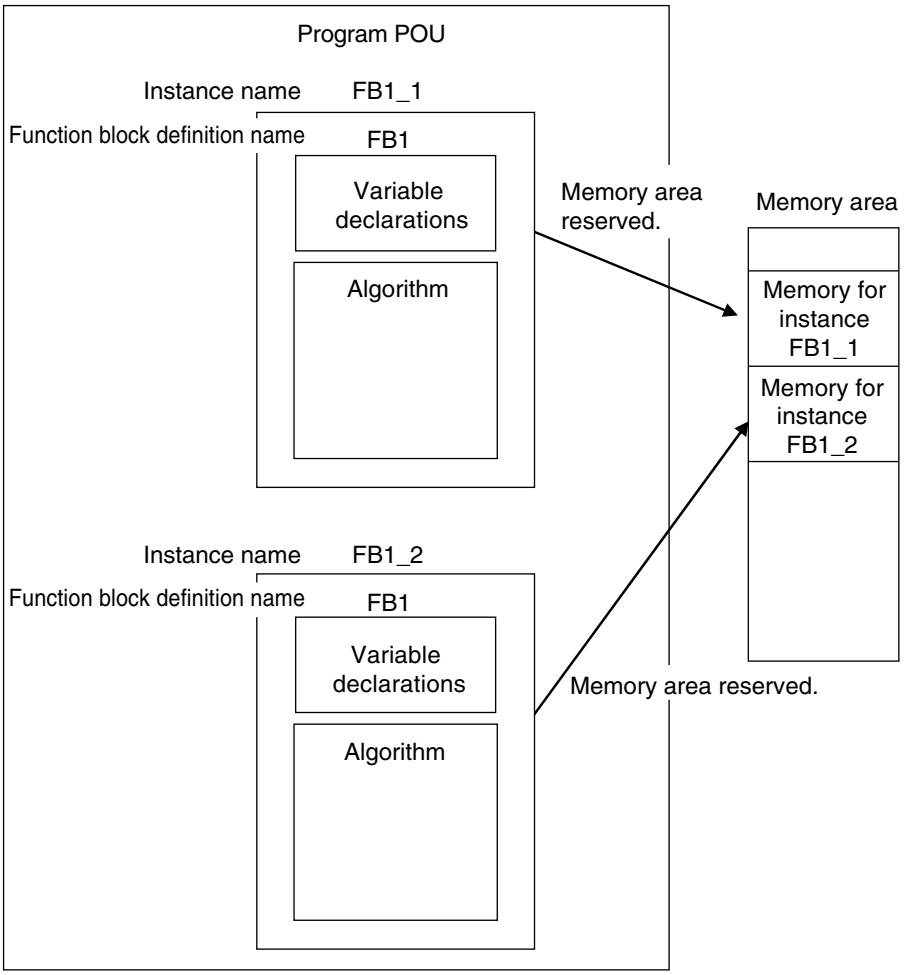
## Function Block Definitions and Instances

A function block consists of a "function block definition" that is made in advance and "instances" that are used in the actual programs.
All instances of a function block are based on the "function block definition".
A "function block definition" consists of an algorithm and a local variable table.

## Function Block Instance

When you place an instance of a function block definition in a program or another function block, the function block definition is treated as a part of that program or function block.
Function block definitions that are called from a program or another function block are called "instances".
Every instance of a function block has an identifier known as an instance name associated with it, and every instance uses memory.

You can use a single function block definition to create more than one "instance". This allows you to process different I/O data with the same function.




If you place instance names *FB1_1* and *FB1_2* for function block FB1 in the program, each instance requires its own space in memory.
"Instances" cannot be read from other programs or function blocks. If an instance with the same name as another instance is placed in a different program or another function block, that instance will operate as a completely separate instance.

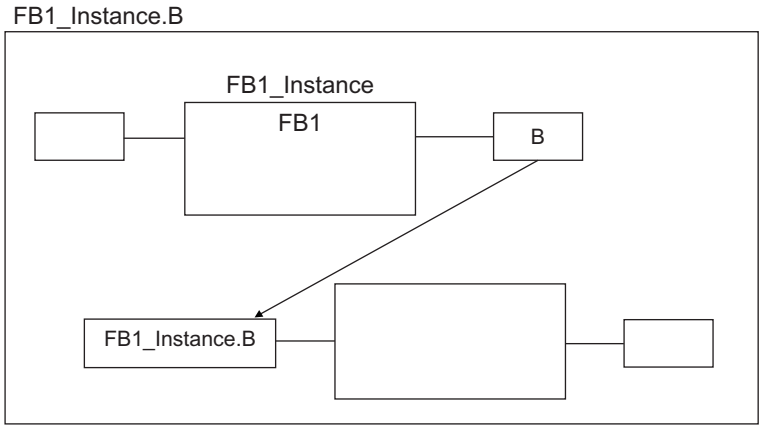
**Precautions for Correct Use**

In the following conditions, a user-defined function block will cause an error during the program check when the program is built.
- The same function block instance was called more than once in the POU.
- The instance of the function block was registered as a global variable.

## Accessing Variables in a Function Block from Outside the Function Block

You can access the input and output variables of a function block from outside the function block. Variables are written as follows:
InstanceName.VariableName

Example: To Access Output Variable *B* of Function Block Instance *FB1_Instance*



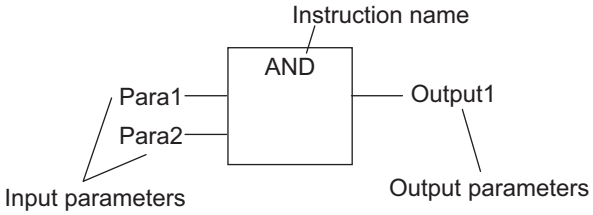

You can access the input and output variables for a function block only within the program that contains the function block.
However, you cannot access these variables from within other function block instances even if they are in the same program. You cannot access them from other programs.

# 6-1-6 Details on Functions

You cannot create user-defined functions for Safety Control Units. Only system-defined functions (instructions) are allowed.

## Structure of Functions

With the FBD language, functions are represented as rectangular boxes as shown below.
A function consists of the following parts.
This function is expressed in the FBD language:




### ● Instruction Name

This is the instruction name.

### ● Instance Name

Functions do not have instance names.

## 6-1-7 Instructions

Instructions are the smallest unit of the processing elements that are provided by OMRON for use in POU algorithms.
There are FB instructions and FUN instructions.
Programs and user-defined function blocks consist of a combination of these instructions.

You can press a shortcut key to display help for an instruction.
Select an instruction that was previously entered in the FBD editor or select an instruction in the Instruction Navigator in the Toolbox, and then press the **F1** Key.



**Additional Information**

An instruction refers to a system-defined function or function block.
The following table shows the relationship between user-defined and system-designed functions and the FUN or FB.

| | User-defined functions | System-defined = Instructions |
|---|---|---|
| FB | Supported | Supported |
| FUN | Not supported | Supported |

For details on instructions, refer to the *NX-series Safety Control Unit Instructions Reference Manual (Cat. No. Z931)*.



**Version Information**

You can display help with a shortcut key with Sysmac Studio version 1.12 or higher.

# 6-2 Variables

In the Safety CPU Unit, variables are used to exchange I/O information with external devices, to perform data calculations, and to perform other processes.
This section describes variable designations in detail.

## 6-2-1 Variables

Variables store I/O data for exchange with external devices or temporary data that is used for internal POU processing. In other words, a variable is a data container with attributes, such as the name and data type.

You do not need to assign a memory address to a variable. The Sysmac Studio automatically allocates memory addresses in the memory area for variables.

## 6-2-2 Types of Variables

Variables are broadly classified into the following two types.

- **User-defined Variables**

  The user defines all of the attributes of a user-defined variable.
  The rest of this section describes "user-defined variables".

- **Semi-user-defined Variables**

  For semi-user-defined variables, some attributes are designed by the system, while others are defined by the user.
  This includes variables that are used to access specific devices and data.
  This is the equivalent of a "device variable" in the Safety Control Unit.

## 6-2-3 Type of User-defined Variable

There are five types of user-defined variables as defined according to their function in a POU.

| Type of user-defined variable | | POU type | |
|---|---|---|---|
| | | Programs | FB |
| Local variables | Internal variables | Provided | Provided |
| | Input variables | Not provided | Provided |
| | Output variables | Not provided | Provided |
| | External variables | Provided | Not provided |
| Global variables | | Provided*1 | Not provided |

*1. You can define global variables as external variables to access the global variables through the external variables.

## Local Variables

A local variable can be read and written only inside the POU (program or function block) in which it is defined.
“Local variables” is the generic term for internal variables, input variables, output variables, and external variables.

### ● Internal Variables

An internal variable can be used only within one POU.
An internal variable is declared in the local variable table of the POU.
You cannot access the values of internal variables from outside of the POU.
You can declare an internal variable with the same name in different POUs. In this case, memory is allocated separately for each variable.

### ● Input Variables

When a POU is called, the values of the input parameters are assigned to the input variables from the calling POU. An input variable is declared in the local variable table of the POU.

### ● Output Variables

Before processing a POU is completed, the output parameters returned to the calling POU are assigned to the output variables. An output variable is declared in the local variable table of the POU.

### ● External Variables

External variables are used to access global variables from a POU.

## Global Variables

A global variable is declared in the global variable table.

Device variables that are automatically created from the Slave Terminal configuration are automatically registered as global variables.

# 6-2-4 Attributes of Variables

You can set the following attributes for variables.

## Variable Attributes According to Variable Type

### ● Attributes of Variables

| Attribute | Description | Specification | Default |
|---|---|---|---|
| **Variable Name** | The variable name is used to identify the variable. | UTF-8 format<br>127 bytes max. | Name |
| **Data Type** | The data type defines the format of the data that is stored in the variable. | --- | BOOL |

| Attribute | Description | Specification | Default |
|---|---|---|---|
| **Initial Value** | Specify a value for the variable for one of the following situations:<br>• When the power supply is turned ON<br>• When the mode is changed to RUN mode or DEBUG mode. | This setting is required. | FALSE for BOOL and SAFEBOOL variables, and 0 for numeric variables. |
| **Constant** | If you set the Constant attribute, you can set the initial value of the variable when it is downloaded, but you cannot overwrite the value afterward. | Specify making the value a constant or not a constant. | Do not specify a constant. |
| **Comment** | You can add comments to variables. | UTF-8 format<br>127 bytes max. | None (empty). |

### ● Attributes Supported by Each Type of Variable

| Type of variable | | Variable Name | Data Type | Initial Value | Constant | Comment |
|---|---|---|---|---|---|---|
| **Global variables** | | Supported. | Supported. | Supported. | Supported. | Supported. |
| **Programs** | **Internal variables** | Supported. | Supported. | Supported. | Supported. | Supported. |
| | **External variables** | Not supported. | Not supported. | Not supported. | Supported. | Supported. |
| **Function blocks** | **Internal variables** | Supported. | Supported. | Supported. | Supported. | Supported. |
| | **Input variables** | Supported. | Supported. | Supported. | Not supported. | Supported. |
| | **Output variables** | Supported. | Supported. | Supported. | Not supported. | Supported. |

**Additional Information**

The following comments are linked. If you change a comment, the comment for the corresponding parameter is also changed.
- Device variable comments in the I/O Map
- Global variable comments
- I/O terminal comments in the Parameters Tab Page

## 6-2-5 Data Type

The Data Type attribute defines the type of data and range of data that is expressed by a variable. The amount of memory that is allocated when you declare a variable depends on the data type of that variable. The more memory allocated, the larger the range of values that the variable can express.
The data types for the input and output variables of instructions depend on the instruction. Set the data types of input and output parameters for the instruction arguments according to the data types of the input and output variables for that instruction.
The Safety Control Unit allows the use of only pre-defined basic data types. You cannot use user-defined derivative data types, such as structures, unions, and enumerations, or array specifications.

### Basic Data Types

The basic data types that you can use with the Safety Control Unit are listed below.

| Type | Definition |
|---|---|
| **Boolean** | A data type with a value of either TRUE or FALSE. |
| **Bit string** | A data type that represents a value as a bit string. |
| **Integers** | A data type that represents an integer value. |

| Type | Definition |
|---|---|
| **Duration** | A data type that represents a time duration (days, hours, minutes, seconds, and milliseconds). |

## Safety Data Types and Standard Data Types

The Safety Control Unit classifies data types into the following two types to distinguish between safety data and standard data.

- Safety data types: These data types represent data related to safety control.
- Standard data types: These data types represent data related to standard control.

"SAFE" is prefixed to the names of the standard data types such as SAFEBOOL, to create the names of the safety data types.

You can input a signal for a safety data type variable to a standard data type variable.
You cannot input a signal for a standard data type variable to a safety data type variable. A building error will occur.

## Basic Data Types

The basic data types are given below.

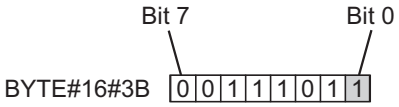
| Classification | Data type | Safety/ standard data type | Range of values | Notation |
|---|---|---|---|---|
| **Boolean** | **BOOL** | Standard data type | FALSE or TRUE | bool#0 or bool#1<br>FALSE or TRUE |
| | **SAFEBOOL** | Safety data type | | |
| **Bit strings** | **BYTE**[*1][*2] | Standard data type | byte#16#00 to byte#16#FF | byte#2#0101010<br>byte#2#0101_1010<br>byte#16#5A<br>You can use the separator character "_". |
| | **SAFEBYTE** | Safety data type | | |
| | **WORD**[*3] | Standard data type | word#16#0000 to word#16#FFFF | |
| | **SAFEWORD**[*4] | Safety data type | | |
| **Integers** | **INT** | Standard data type | int#-32768 to int#-32767 | 100<br>int#100<br>int#2#00000000_1100100<br>int#16#64<br>-100 |
| | **SAFEINT** | Safety data type | | |
| | **DINT** | Standard data type | dint#-2147483648 to dint#-2147483647 | |
| | **SAFEDINT** | Safety data type | | |
| **Duration** | **TIME**[*4] | Standard data type | t#0ms(t#0d0h0m0s0ms) to t#4294967295ms (t#49d17h02m47s295ms) | t#3000ms |
| | **SAFETIME**[*4] | Safety data type | | |

*1. The BYTE data type cannot be used for an internal variable.
*2. If you use the BYTE data type for a global variable, you must define an exposed variable.

*3. If you use the WORD data type for a global variable, you must define an exposed variable or use a constant.
*4. If you use the SAFEWORD, TIME, or SAFETIME data type for a global variable, you must set a constant.

## Bit String Data Format

This section describes the data format for bit string data.
"Bit 0" is the least significant bit of a bit string variable.
Bit values are expressed as 1 or 0.




# 6-2-6 Variable Attributes Other Than Data Type

This section describes the variable attributes other than the Data Type.

## Variable Name Attribute

The variable name is used to identify the variable.
Each variable in a POU must have a unique name. However, you can declare local variables with the same variable name in different POUs. These are treated as two separate variables.
You cannot declare an internal variable with the same variable name as a global variable.

## Initial Value Attribute

The variable is set to the initial value in the following situations.
- When the power supply is turned ON
- When the mode is changed to RUN mode
- When the mode is changed to DEBUG mode (STOPPED)

### ● Types of Variables That Can Have Initial Values

You can set initial values for only some types of variables. A list is provided below.

| Variables | Initial Value Settings |
|---|---|
| **Global variables** | Yes (required) |
| **Internal variables** | |
| **Input variables** | |
| **Output variables** | |
| **External variables** | Not possible. |

You must set initial values for all variables that allow them.

### Constant Attribute

The Constant attribute prohibits instructions from writing values to a variable. Setting the Constant attribute will prevent any program from overwriting the variable.
The values of variables with a Constant attribute cannot be written from instructions after the initial value is set. If there is an instruction in a POU that attempts to write a value to a variable with the Constant attribute, an error will occur when the program is built.

## 6-2-7 Function Block Instances

Function block instances are added to and displayed in the local variable table or the global variable table as data types.



**Additional Information**

A function block instance is treated as a local variable of the program in which the instance is created. As such, the instance is added to and displayed in the local variable table of the program.

## 6-2-8 Restrictions on Variable Names and Other Safety Program-related Names

The following table lists the restrictions on variable names and other safety program-related names.

## Character Restrictions

| Safety program-related name | Applicable characters | Reserved words | Multi-byte character compatibility | Case sensitivity | Maximum size *1 (not including NULL) | Character code |
|---|---|---|---|---|---|---|
| Variable names (including POU instance names) | **Usable characters**<br>• 0 to 9, A to Z, and a to z<br>• _ (underlines)<br>Refer to *Reserved Words* below for a list of the reserved words.<br>**Characters that cannot be used together**<br>• A text string that starts with a number (0 to 9)<br>• A text string that starts in an underline (_) character<br>• A text string that contains two or more consecutive underline (_) character<br>• A text string that ends in an underline (_) character<br>• Identifiers formed from a string of characters that is prefixed or suffixed with one or more expansion characters or spaces | Refer to *Reserved Words* on page 6-18 below. | Not supported | Not case sensitive. | 127 bytes | ASCII |
| POU definition names | | | | | | |
| Full path of variable names (Example: This includes the number of characters for the instance name and period, which is InstanceName.OutputVariableName when accessing the output variable of a function block.) | | | | | 511 bytes | |
| Device names | | | | | 127 bytes | |
| Variable comments | | None | Supported | --- | 127 bytes | UTF-8 |

*1. The individual restrictions are as listed in the table. The actual upper limits depend on the overall program capacity and memory capacity for variables.

## Reserved Words

An error is detected during the program check for the following names.

- Use of the same name as any of the instructions that are described in the *NX-series Safety Control Unit Instructions Reference Manual (Cat. No. Z931)*
- Words that are reserved by the system

## Names that Must Be Unique

The following names must be unique. A building error will occur.

- Global variable names in the same Safety CPU Unit
- Variable names in the same POU
- Local variable names and global variable names

# 6-3 Constants (Literals)

This section describes constants.

## 6-3-1 Constants

The value of a variable changes depending on the data that is assigned to that variable. The value of a constant never changes.
Unlike variables, constants are not stored in memory. You can use constants in the algorithm of a POU without the need to declare them.
Constants have a "data type" in the same way as variables.

## 6-3-2 Types of Constants

The following types of constants can be used with Safety Control Units.

- Bits
- Numbers
- Bit strings
- Times

The following tables show the notation to define different constants for the Safety Control Unit. The constant is normalized after it is entered.

### Bits

| Notation | Example | Remarks |
|---|---|---|
| TRUE or FALSE | FALSE or TRUE | |
| {data_type}#{numeric_value} | bool#0 or bool#1 | Data type: BOOL |

### Numbers

#### ● Integers

| Notation | Example | Remarks |
|---|---|---|
| {data_type}#{base}#{numeric_value} | int#10#1 | • Data type: int or dint<br>• Base: 2, 8, 10, or 16<br>The editor on the Sysmac Studio does not show the base of 10. Values entered as the base of 8 are converted to decimal numbers.<br>• Numeric values cannot be signed (+ or −). |
| {data_type}#{numeric_value} | int#1 | This is interpreted as decimal data. |
| {numeric_value} | -100 | This is interpreted as SAFEINT or SAFEDINT data. |

## Bit Strings

### ● Bit String Data

| Notation | Example | Example |
|---|---|---|
| {data_type}#{base}#{numeric_value} | word#16#0064 | • Data type: BYTE or WORD<br>• Base: 2, 8, 10, or 16<br>The normalizing processing omits the base of 10 and converts values entered as base of 8 to decimal numbers. |
| {data_type}#{numeric_value} | word#100 | This is interpreted as decimal data. |

## Times

### ● Durations

| Notation | Example | Remarks |
|---|---|---|
| {Data type}#{days}d{hours}h{minutes}m{seconds}s{milliseconds}ms | t#61m5s | Data Type: t |

# 6-4 Programming Languages

This section describes the programming languages in detail.
Refer to *6-5 Programming Operations* on page 6-26 to learn how to enter the programming languages on the Sysmac Studio.

## 6-4-1 Programming Languages

The languages used to express the algorithms in a POU (program or function block) are called the programming languages.
FBD is the only programming language that can be used with the Safety Control Unit.

## 6-4-2 FBD Language

The FBD language is a graphical programming language that is used for programmable controllers and is defined by IEC 61131-3.
You use connecting lines to show the data flow, and rectangular boxes to represent functions and function blocks to write algorithms.

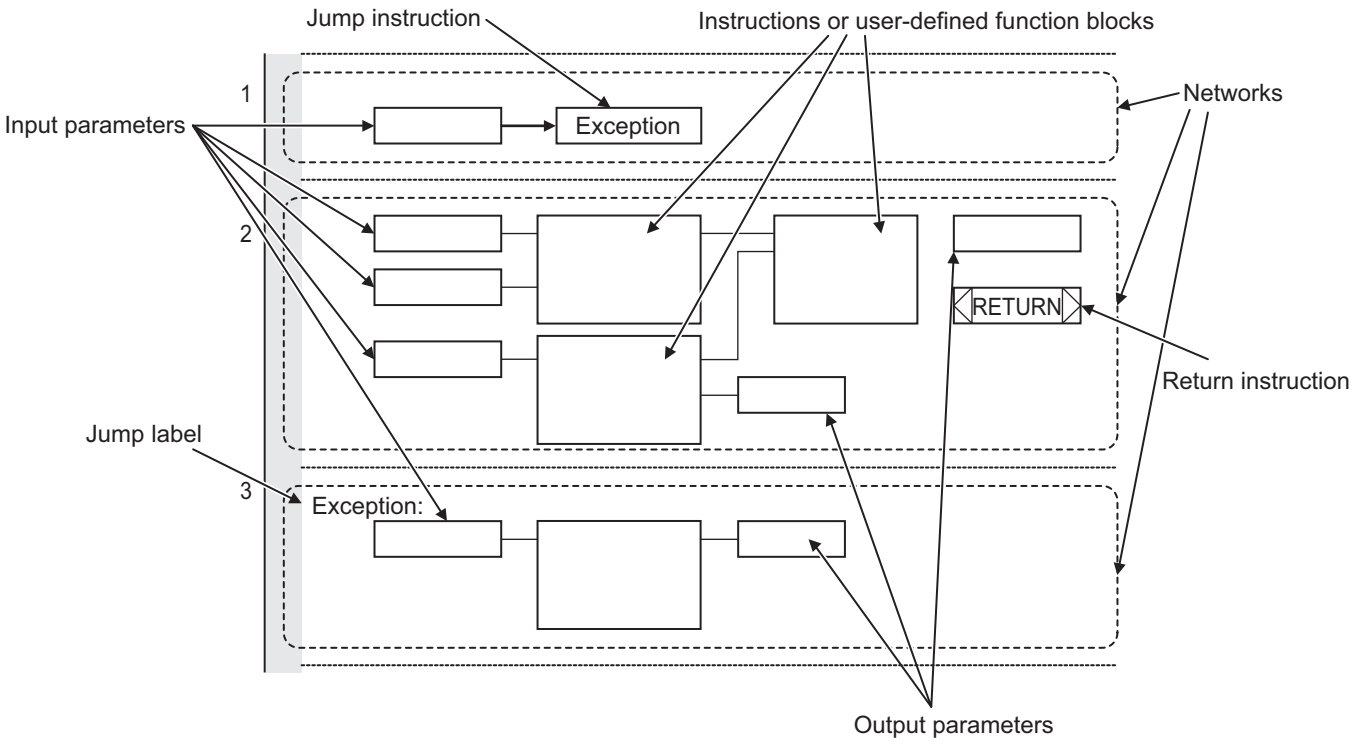
### Elements of the FBD Language

An algorithm in the FBD language is a unit made up of a series of elements, called "networks", that connect the inputs to the outputs. The networks consist of the following elements.

- Input parameters
- Connecting lines
- Instructions (FUN or FB instructions) or user-defined function blocks
- Output parameters

In a network, signals flow from the inputs on the left to the outputs on the right.

- Input parameters and output parameters are arguments that are written with variables or constants. These arguments are written in the areas that are connected to the terminals of input variables or output variables inside the instructions or user-defined function blocks with connecting lines.
- The connecting lines show the flow of the following three types of signals.
  a) Flow between input and output parameters and instructions
  b) Flow between terminals on user-defined function blocks
  c) Horizontal and vertical flow between instructions or between user-defined function blocks
- Instructions and user-defined function blocks are represented by rectangular boxes. You can use connecting lines to connect input variables or output variables. Some terminals do not need to be connected with a connecting line.



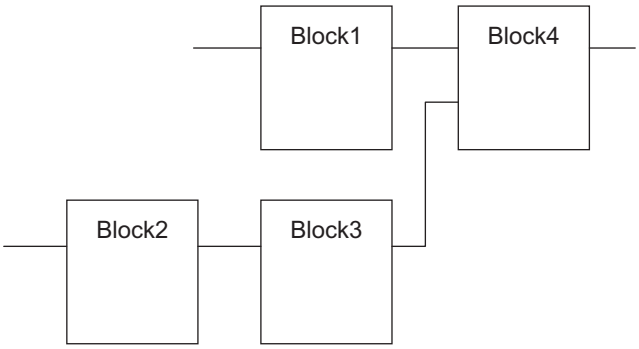

The networks shown above include a Jump instruction that changes the top-to-bottom flow of execution between networks, a label that shows the network to jump to, and a Return instruction.
Refer to *Execution Order of Safety Programs Written in the FBD Language* on page 6-22 and *Execution Control* on page 6-23 later in this manual, and also to the *NX-series Safety Control Unit Instructions Reference Manual (Cat. No. Z931)* for details.

**Additional Information**

- Unlike the ladder diagram language, the FBD language does not have bus bars. The connecting lines do not indicate power flow. They indicate the flow of data. The FBD language does not have an END instruction. Execution for the task period ends when the last network is executed.
- In this manual, "FBD network" is sometimes used to differentiate programming networks from physical networks, such as EtherCAT networks.

## Execution Order of Safety Programs Written in the FBD Language

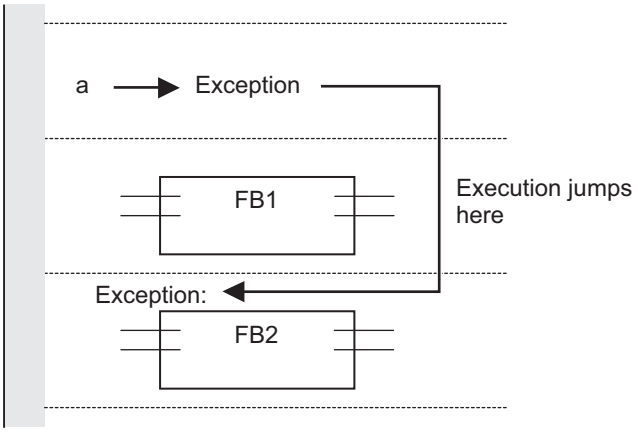
In POUs that are written in the FBD language, networks are executed in order from top to bottom. Processing ends when the network at the very bottom of the program is executed.
Elements in the same network are executed from top to bottom for FUN and FB inputs and left to right for blocks that are connected in series. In the following example, execution is in the following order: Block 1, Block 2, Block 3, and then Block 4.




However, if there is a Return instruction in the middle of a program and the execution condition is met, the POU is ended and a return is made to the source of the call. No processes after Return instruction are executed.

## Execution Control

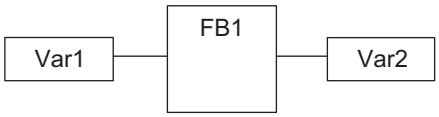
Safety programs that are written in the FBD language are generally executed from top to bottom, but you can use the Jump instruction to change the execution order.
For example, when the value of variable *a* changes to TRUE in the following example, execution will move to the network labeled "Exception".
You cannot jump to a network that is above the current network.



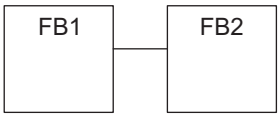
## Connecting Instructions or User-defined Function Blocks

### ● Correct Connection Configurations

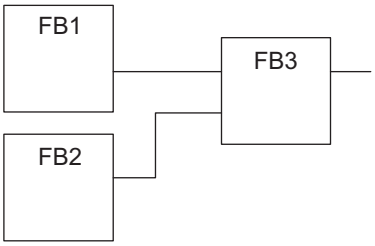
Parameters and commands, or user-defined function blocks can be connected with a connecting line.




The connecting line can connect two instructions or two user-defined function blocks.

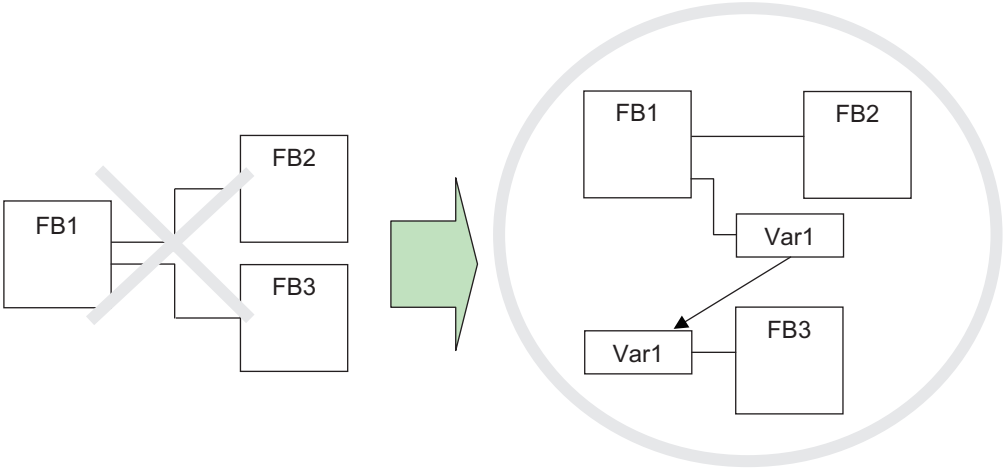



Multiple instructions or user-defined function blocks can be connected to a single instruction or user-defined function block.

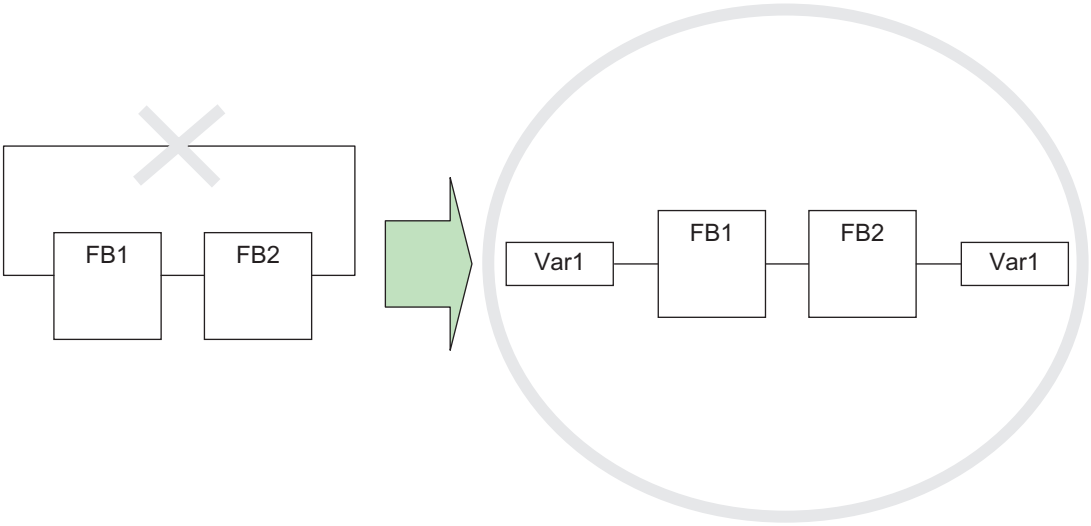



### ● Incorrect Connection Configurations

You cannot connect more than one instruction or user-defined function block to the right of another instruction or user-defined function block. In this case, you must pass the signal to a variable as shown in the following figure.




You cannot route a connecting line from the output to the input. In this case, you must pass the signal to a variable as shown in the following figure.

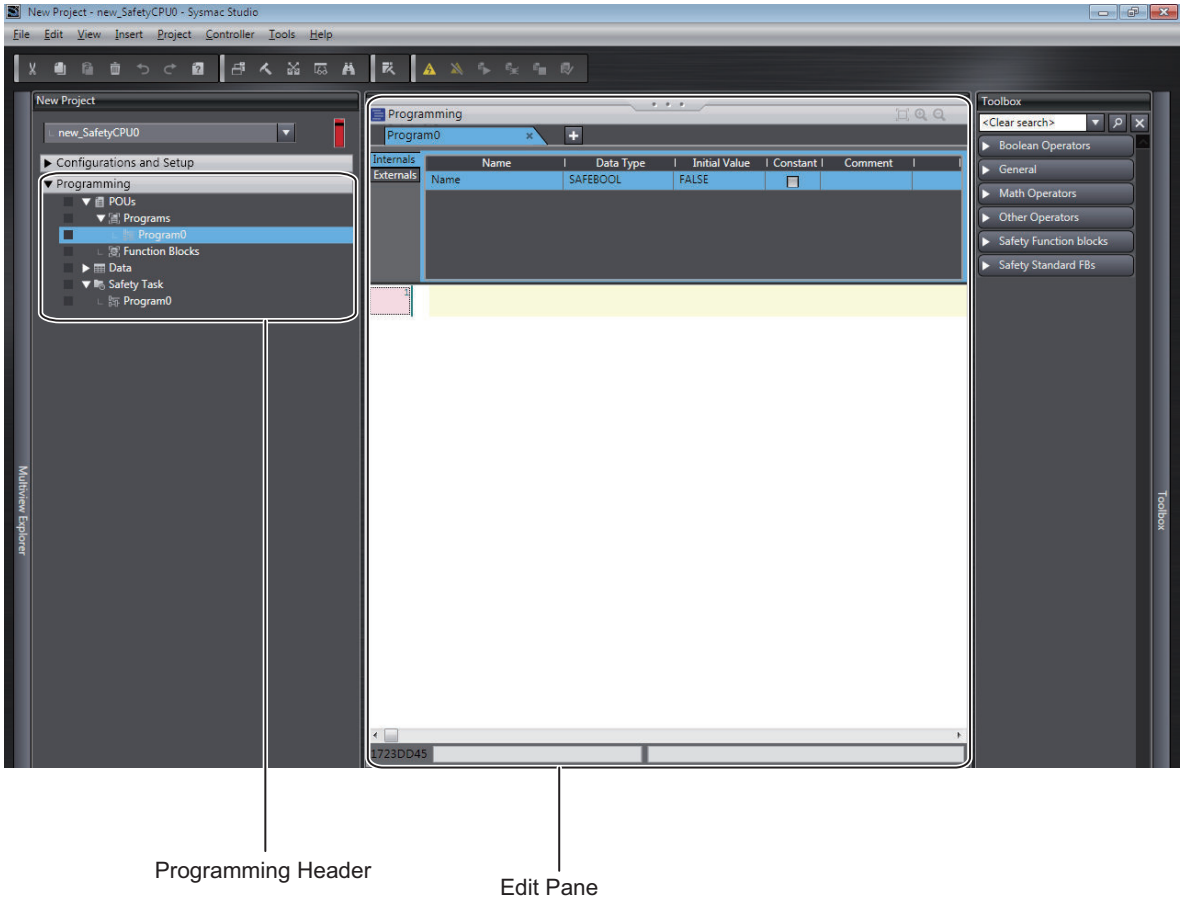

FB1
FB2
Var1
FB1
FB2
Var1

# 6-5 Programming Operations

This section describes the procedures on the Sysmac Studio that you use to create safety programs for the Safety CPU Unit.

## 6-5-1 Programming Layer on the Sysmac Studio

You use the Programming Layer with the Controller set to the Safety CPU Unit on the Sysmac Studio as shown below to create safety programs for the Safety CPU Unit.

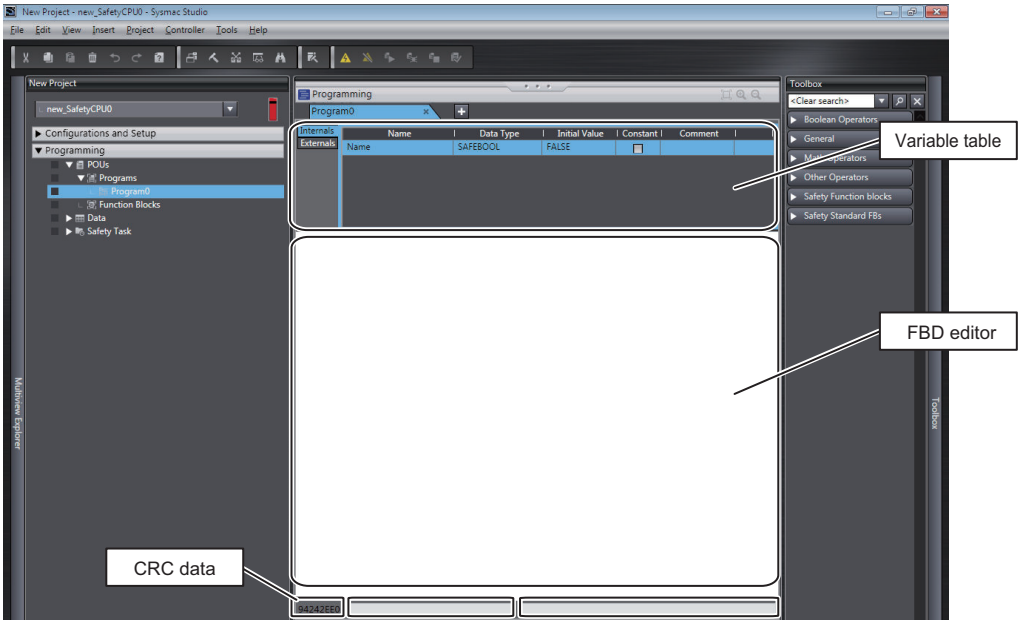



The Programming Headers of the Multiview Explorer are organized as shown below.

| Programming Header | Description |
|---|---|
| POUs | |
| Programs | |
| Program0 | The list of programs is displayed.<br>"Program0" is created when a new project is created.<br>Double-click a program to display it in the FBD editor and begin editing.<br>In the Multiview Explorer, you can change the names of programs or delete, copy, paste, and cut programs. |
| Program1 | |
| Function Blocks | |

| Programming Header | | | Description |
|---|---|---|---|
| | | FunctionBlock0 | A list of user-defined function blocks is displayed.<br>There are no function blocks when you create a new project.<br>Double-click a function block to display it in the FBD editor and begin editing.<br>You can change the name, delete, copy, paste, and cut function blocks. |
| | | FunctionBlock1 | |
| Data | | | |
| | Global variables | | Double-click Global Variables to display the Global Variable Table and begin editing. |

The structure of the Edit Pane is shown below.



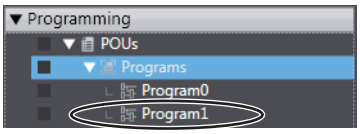

| Item | Editing | Description |
|---|---|---|
| Variable table | Possible | Displays the local variables. |
| FBD editor | Possible | Displays the FBD network. |
| CRC data | Not possible | Displays the CRC data of the POU. |

## 6-5-2 Registering POUs

This section shows how to register programs and function blocks.

### Registering New Programs

**1** Right-click **Programs** under **Programming** - **POUs** and select **Add** - **Program** from the menu. A new program is added under **Programs**.




**2** Double-click the program that was added.

The variable table and FBD editor are displayed in the Edit Pane. From here you can edit programs.
Refer to *6-5-3 Registering Variables* on page 6-37 for information on how to register variables, and *6-5-4 FBD Programming* on page 6-45 for information on programming in the FBD editor.

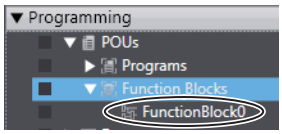
## Registering Function Blocks

Function blocks are written in the FBD language. You can call them from safety programs as required. You can use functions inside function blocks.
Refer to *6-1-2 Overview of the Three Types of POUs* on page 6-3 for a detailed description of function blocks.

### ● Registering New Function Block

This section describes the procedures for registering a new user-defined function block.
Function block instructions are registered in the Sysmac Studio in advance. You do not need to register function block instructions to use them.

***1*** Right-click **Function Blocks** under **Programming** - **POUs** in the Multiview Explorer and select **Add** - **FunctionBlock** from the menu.
A new function block is added under **Function Blocks**.



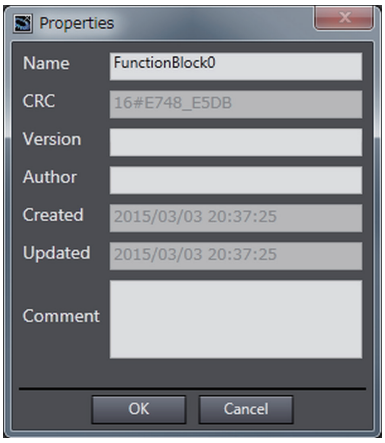

***2*** Double-click the new function block.
The variable table for the function block and the FBD editor are displayed in the Edit Pane.
From here you can create local variables and FBD networks.

Refer to *6-5-3 Registering Variables* on page 6-37 for information on how to register variables, and *6-5-4 FBD Programming* on page 6-45 for information on programming in the FBD editor.

## Displaying Properties

This section describes how to display program and function block properties.

Right-click the registered program or function block and select **Properties** from the menu.
The Properties Dialog Box is displayed.




The following table describes the items in the Property Dialog Box.

| Item | Editing | Description |
|---|---|---|
| Name | Possible | This is the name that is currently set for the POU. |
| CRC | Not possible | This is the CRC data of the POU.<br>You can also check the CRC data on the **Change Tracking** Tab Page. |
| Version | Possible | This is the version information for the POU.<br>You can also check the version information on the **Change Tracking** Tab Page. |
| Author | Possible | This is the person that created the POU. |
| Created | Not possible | This is the date that the POU was created. |
| Updated | Not possible | This is the date that the POU was updated. |
| Comment | Possible | This is a comment for the POU. |

**Precautions for Correct Use**

The author, creation date, and update date cannot be saved on Sysmac Studio version 1.12 or lower.
If you open a project on Sysmac Studio version 1.12 or lower, this data will be lost.

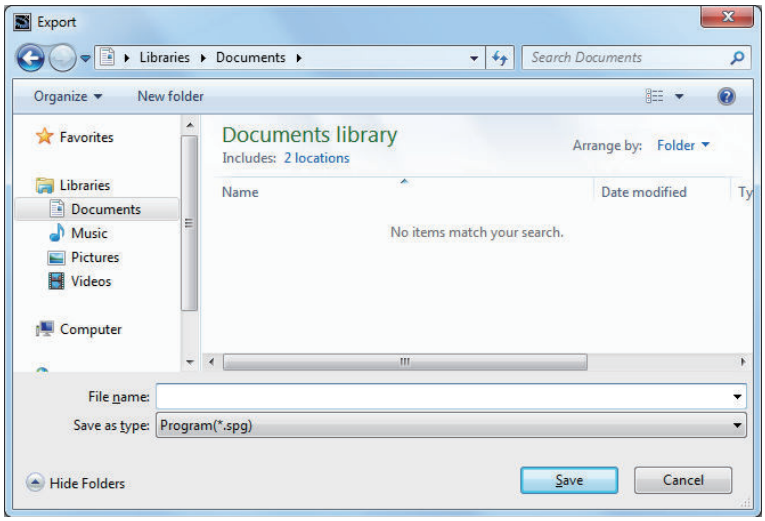
**Version Information**

Sysmac Studio version 1.13 or higher is required to display these properties.

## Exporting Programs

You can export a program to a file (extension .spg).
You can import an exported program to reuse it in another project.
Use the following procedure.

1 Right-click a program displayed under **Programming** - **POUs** - **Programs** in the Multiview Explorer, and then select **Export** from the menu.
The **Export** Dialog Box is displayed.




**2** Enter a file name, and then click the **Save** Button.
A program file with an .spg extension is saved.

### Precautions for Correct Use

- Do not delete the program CRC data that is displayed after the program is exported. It is used to check the data when importing the program.
  You can check the CRC data on the **Change Tracking** Tab Page or Properties Dialog Box.
- The spg program file includes information on global variables that are used as external variables.

### Version Information

You can export programs with Sysmac Studio version 1.17 or higher.

## Importing Programs

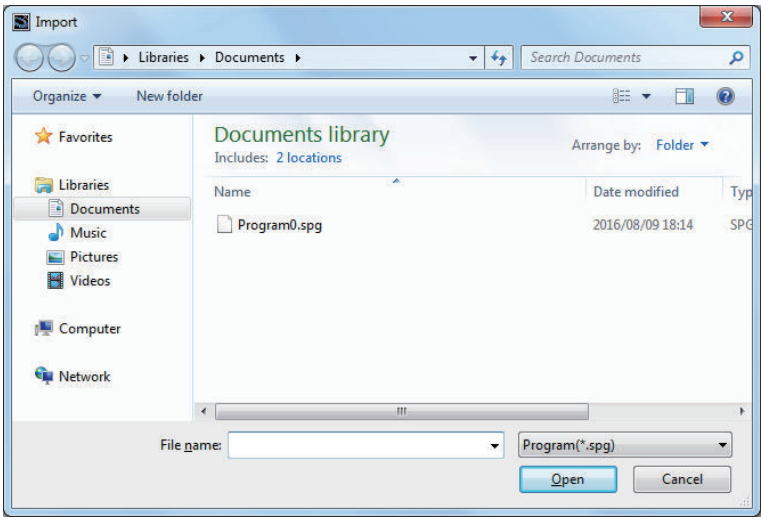
You can import an exported program to reuse it in another project.

**⚠ WARNING**

Check during the import of the program that the CRC of the program is correct. Serious injury may possibly occur due to loss of required safety functions.

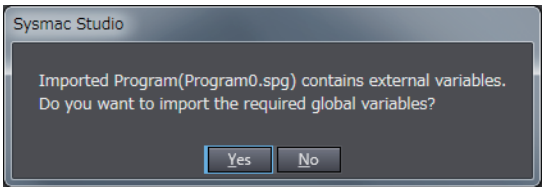
Use the following procedure to import a program that was previously exported.

**1** Right-click **Programs** under **Programming-POUs** in the Multiview Explorer and select **Import** from the menu.
The **Import** Dialog Box is displayed.

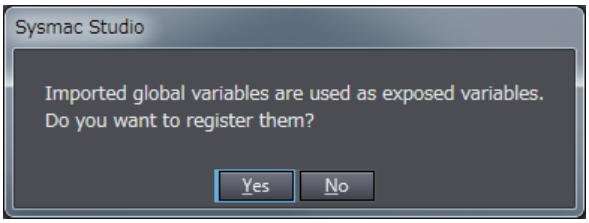



When you select the files to import, you can press the**Shift**Key or**Ctrl**Key to select more than one file.

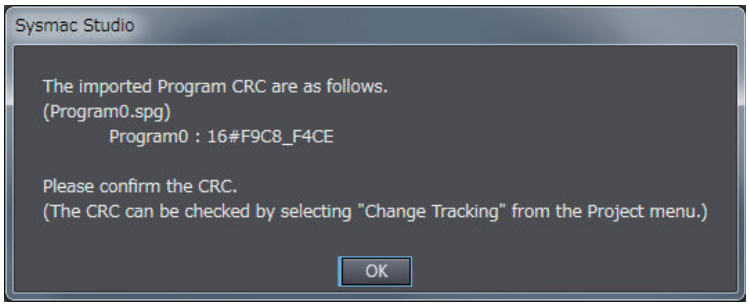
**2** Select the .spg file to import, and then click the**Open**Button.
When external variables are used in the program to import, the following dialog box is displayed.
Select**Yes**to register the variables as global variables.




When the registered global variables are used as exposed variables, the following dialog box is displayed.
Select**Yes**to register the variables as exposed variables.




A dialog box to check the CRC data is displayed.




**3** Check the CRC data, and click the **OK** Button.
The program is imported and added to the project.

### Precautions for Correct Use

After a function block is imported, the safety program in the project will be in an unvalidated state. Always execute safety validation again.

### Additional Information

You can check the CRC data on the **Change Tracking** Tab Page or Properties Dialog Box.

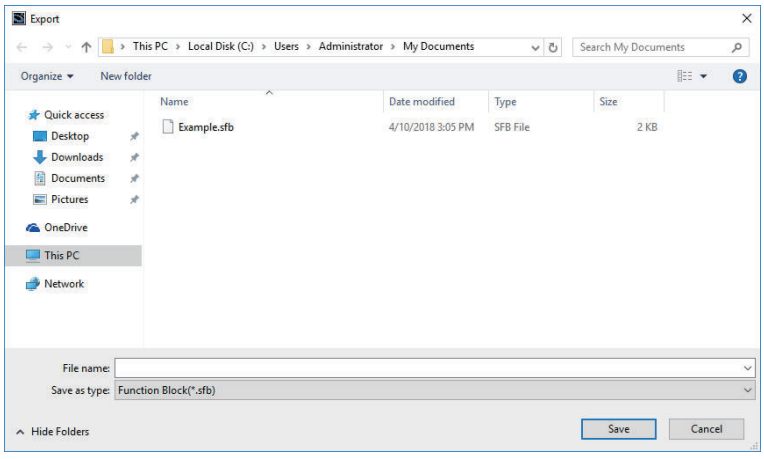
### Version Information

You can import programs with Sysmac Studio version 1.17 or higher.

## Exporting User-defined Function Blocks

You can export a user-defined function block to a file (extension .sfb).
You can import the exported user-defined function block to reuse it in another project.
Use the following procedure.

**1** Right-click the user-defined function block under **Programming** - **POUs** - **Function Blocks** in the Multiview Explorer and select **Export** from the menu.
The **Export** Dialog Box is displayed.



**2** Enter a file name, and then click the **Save** Button.
A function block file with an .sfb extension is saved.

### Precautions for Correct Use

- Do not delete the function block CRC data that is displayed after the function block is exported. It is used for data confirmation when the function block is imported.
  You can check the CRC data on the **Change Tracking** Tab Page or Properties Dialog Box.
- A help file is not included with the user-defined function block file (sfb).

### Version Information

You can export user-defined function blocks with Sysmac Studio version 1.12 or higher.

## Importing User-defined Function Blocks

You can import the exported user-defined function block to reuse it in another project.

**⚠ WARNING**

Check during the import of the user defined function that the CRC of the imported function block is correct. Serious injury may possibly occur due to loss of required safety functions.

Use the following procedure to import a user-defined function block that was previously exported.

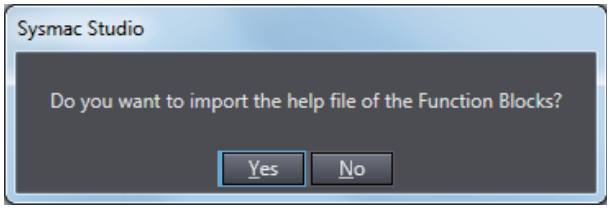
**1** Right-click **Function Blocks** under **Programming** - **POUs** in the Multiview Explorer and select **Import** from the menu.
The **Import** Dialog Box is displayed.




When you select the files to import, you can press the **Shift** Key or **Ctrl** Key to select more than one file.

**2** Select the .sfb file to import, and then click the **Open** Button.

The following dialog box is displayed. To import the function block help file, select **Yes**.

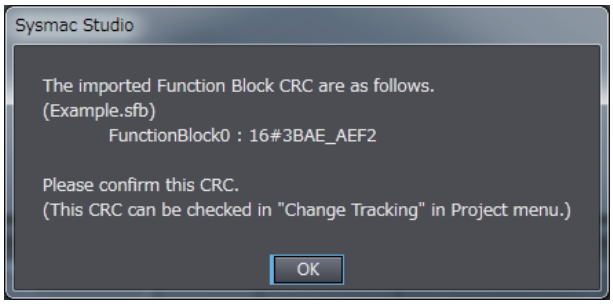



When a function block help file is imported, the following **Import** Dialog Box is displayed.




If you select a help file (.pdf extension) with the same name as the imported function block, the specified file is copied to the save location for the user-defined help file.

A dialog box to check the CRC data is displayed.




***3*** Check the CRC data, and click the **OK** Button.
The function block is imported and added to the project.

**Precautions for Correct Use**

After a function block is imported, the safety program in the project will be in an unvalidated state. Always execute safety validation again.

**Additional Information**

You can check the CRC data on the **Change Tracking** Tab Page or Properties Dialog Box.

**Version Information**

You can import user-defined function blocks with Sysmac Studio version 1.12 or higher.

## Help for User-defined Function Blocks

In the FBD editor, select the user-defined function block and then press the **F1** Key to open the help file.

### ● Setting a Help File

Use the following procedure to set a help file for a user-defined function block.

***1*** Prepare a help file with the same name as the function block (extension .pdf).

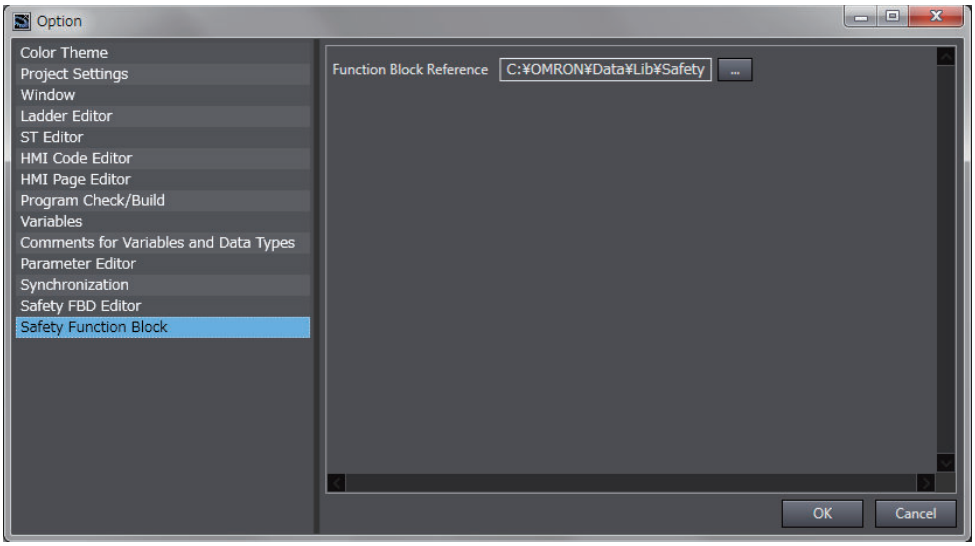
***2*** Save the help file in the specified folder.
The default folder is C:\OMRON\Data\Lib\Safety.

### ● Changing the Save Location for Help Files

You can change the folder in which help files are saved.
Use the following procedure.

***1*** Select **Option** from the **Tools** Menu.
The **Option** Dialog Box is displayed.

***2*** Click the **Safety Function Block**.
The help save location is displayed.




***3*** Select the folder in which to save the help files and click the **OK** Button.

The save location for help files for user-defined function blocks is changed.

**Version Information**

You can use help for user-defined function blocks with Sysmac Studio version 1.12 or higher.

## Importing IEC 61131-10 XML Files

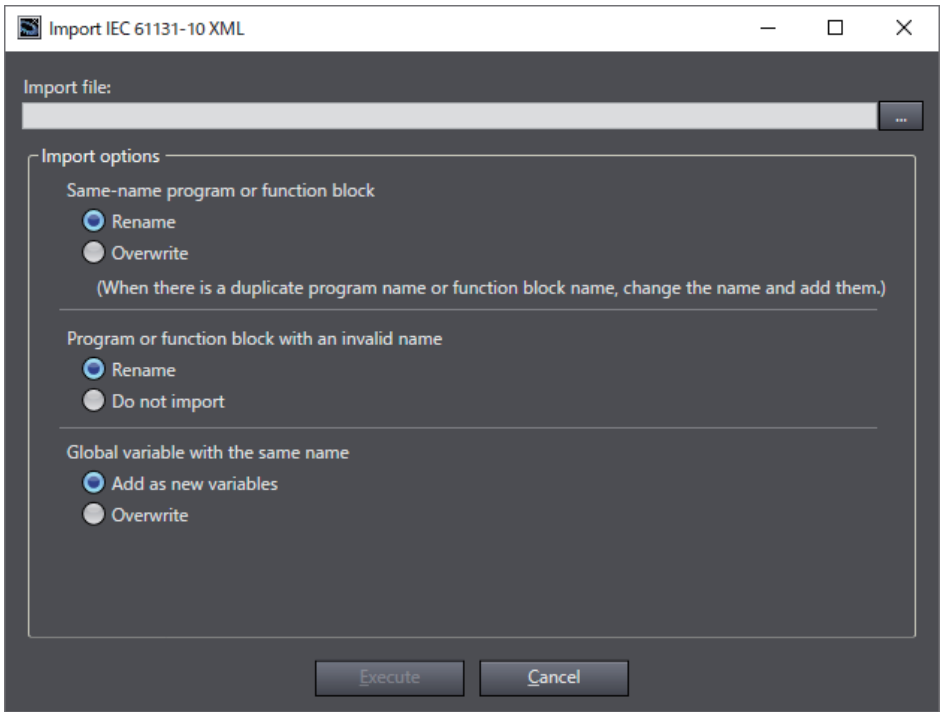
This function imports XML files that conform to IEC 61131-10. You can import global variables and POUs described in FBD language from the XML file.

An example of writing XML is installed by default in the following folder. To check the data supported by this function, refer to the XML schema in the same folder.

- **Sysmac Studio (32 bit)**
  For 32-bit OS:
  C:\Program Files\OMRON\Sysmac Studio\Sample\IEC 61131-10 XML\Safety
  For 64-bit OS:
  C:\Program Files (x86)\OMRON\Sysmac Studio\Sample\IEC 61131-10 XML\Safety
- **Sysmac Studio (64 bit)**
  C:\Program Files\OMRON\Sysmac Studio\Sample\IEC 61131-10 XML\Safety

The procedure to import the IEC 61131-10 XML files is as follows.

***1*** Select **IEC 61131-10 XML** - **Import** from the **Tools** Menu.
The Import IEC 61131-10 XML Dialog Box is displayed.



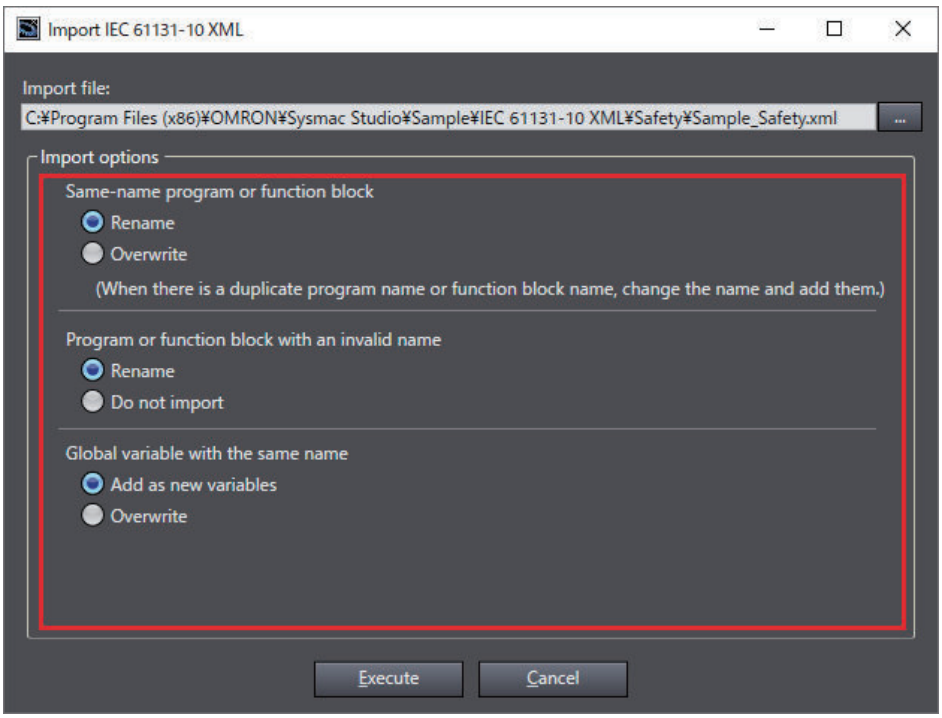

***2*** Click the **View File Selection Dialog Box** Button.




The Select File Dialog Box is displayed.

***3*** In the Select File Dialog Box, select the XML file to import, and then click the **Open** Button.
The screen returns to the Import IEC 61131-10 XML Dialog Box, in which the path to the file that you selected in the Select File Dialog Box is displayed as the import file.

***4*** Select the options before importing the file.




***5*** Click the **Execute** Button.
The data is imported and the global variables and POUs in the XML file are added to the project.
If the project already has any global variable or POU with the same name, data is imported as you configured in the import options. You can check the imported data on the Output Tab Page.

**Precautions for Correct Use**

The data integrity of the XML file is not checked by CRC when it is imported.
Always validate the correct logic under your responsibility after the import and assure proper execution before you use it for actual operation.

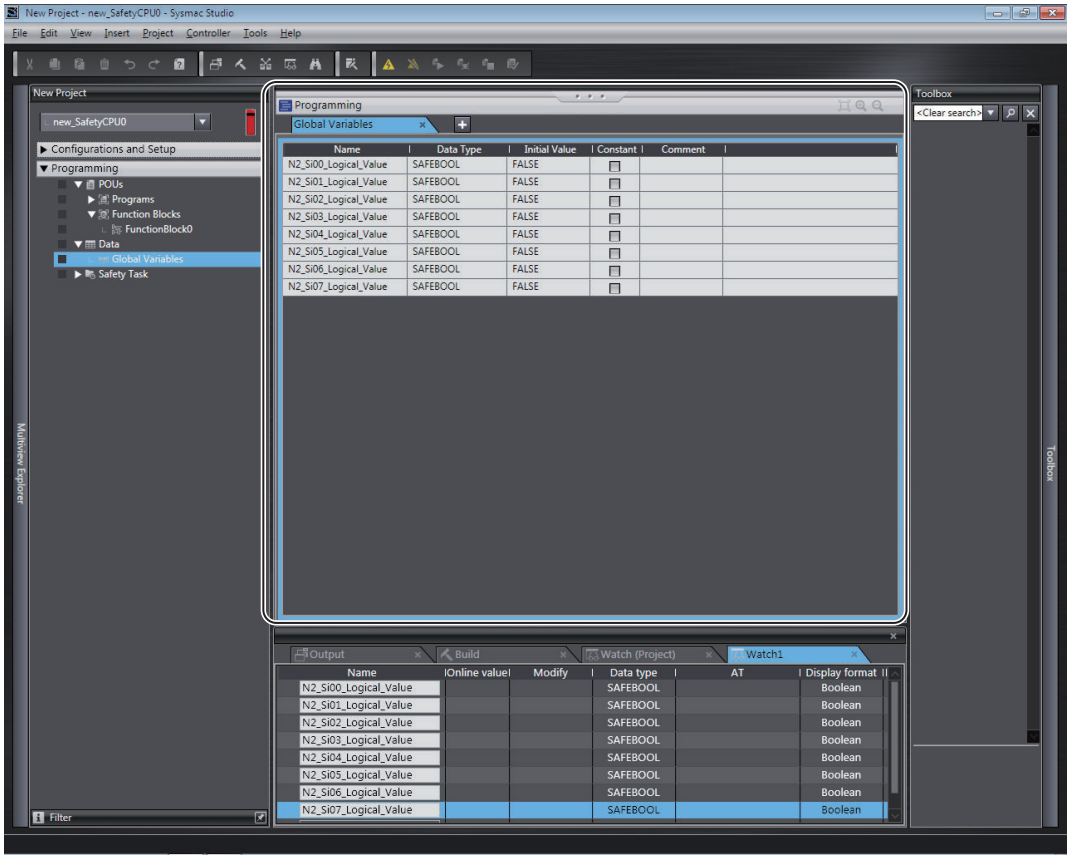
## 6-5-3 Registering Variables

This section describes how to register global variables and local variables.

# Registering Global Variables

## ● Opening the Global Variable Table

Double-click **Global Variables** under **Programming** - **Data** in the Multiview Explorer. Or, right-click **Global Variables** under **Programming** - **Data** and select **Edit** from the menu.
The global variable table is displayed.

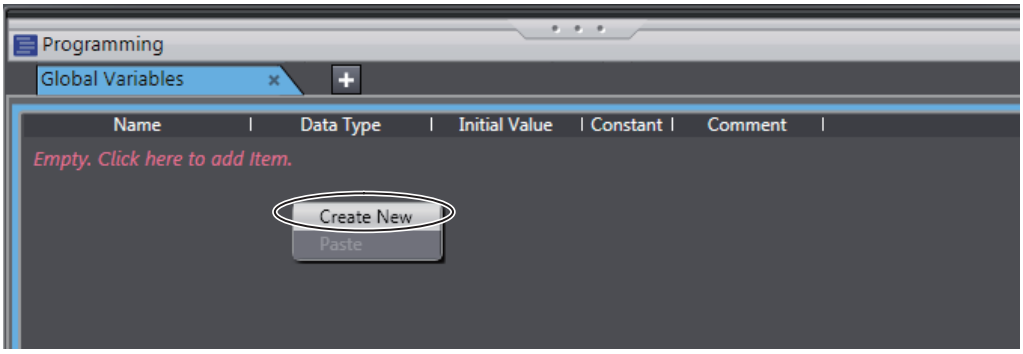



| Field | Description | Restrictions |
|---|---|---|
| Name | Enter a name to use to identify the variable. | Only single-byte alphanumeric characters are allowed.<br>Multi-byte characters, such as those used for Japanese, are not allowed.<br>The maximum size is 127 bytes. |
| Data Type | Set the type of data that is stored in the variable. Refer to *6-2-5 Data Type* on page 6-14 for the data types that you can use. | --- |
| Initial Value | Set the value to use when the power is turned ON, when the mode changes to RUN mode, or DEBUG mode (STOPPED). This parameter must be specified. | --- |
| Constant | Select the check box in the Constant column to set the initial value of the variable when it is downloaded, but prevent it from being changed afterward.<br>Select the check box or clear the selection (default). | --- |

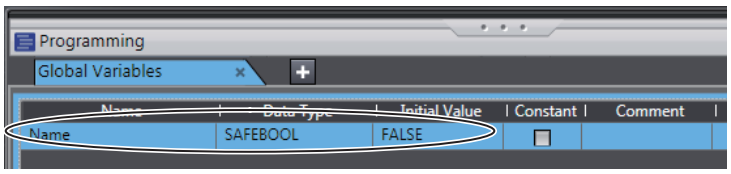
| Field | Description | Restrictions |
|---|---|---|
| Comment | Set any comments for the variable. | The maximum size is 127 bytes. |

## ● Registering New Global Variables

**1** Press the **Insert** Key in the global variable table, or right-click in the global variable table and select **Create New** from the menu.




**2** Enter values for each item, and then press the **Enter** Key.




The variable is registered. Always set the variable name and the data type.

### Additional Information

If there are no registered variables at all, the message *Empty. Click here to add Item* is displayed. Click to add a new variable.

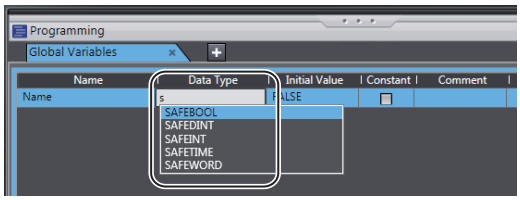
## ● Automatically Registering Global Variables

When you register device variables or exposed variables, any of the variables that are not already registered as global variables are automatically registered as global variables.

Refer to *5-6 Registering Device Variables* on page 5-43 for details on registering device variables.
Refer to *5-7 Exposing Variables to Standard Controllers* on page 5-49 for details on registering exposed variables.

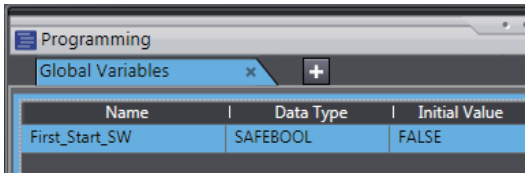
## ● Editing Global Variables

**1** Click the cell to edit for the registered variable.




You can use the autocompletion to enter a data type in the Data Type cells. When you enter the first letter (example: S), a list of data types that begin with the letter is displayed. Select a data type from the list.

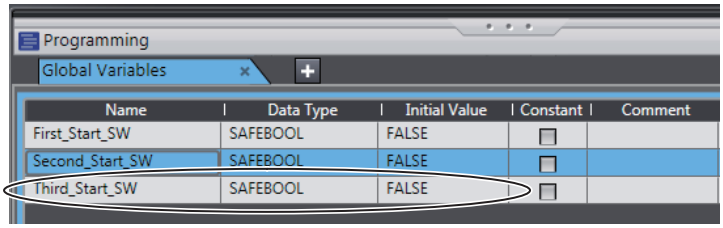
***2*** Change the value or the settings, and then press the **Enter** key.
The change is applied to the variable.




- Entry candidates you can select are displayed as follows:
  Entry candidates are displayed in the Name, Data Type, and Initial Value cells.
  Entry candidates that match the characters in the entered text string are displayed as you edit the text string.
- Entry candidates you can select by drag and drop are displayed as follows:
  You can move the position at which the selected variable is defined. You cannot select multiple variables.
  If you select multiple variables, the variable at the very bottom row will be the target of the drag and drop operation.

## ● Deleting Global Variables

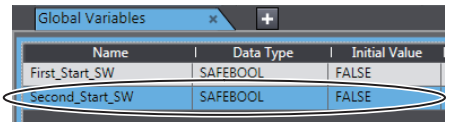
***1*** Click any cell on the line of the variable to delete to select the entire line.




***2*** Press the **Delete** Key. Or, right-click a row and select **Delete** from the menu.
The variable is deleted.

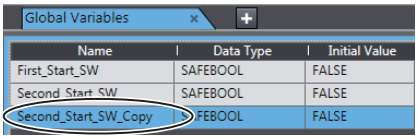
## ● Copying and Pasting Global Variables

***1*** Click any cell on the line of the variable to copy.




**2** Press the **Ctrl** + **C** Keys. Or, right-click the row and select **Copy** from the menu.
The specified variable is copied.

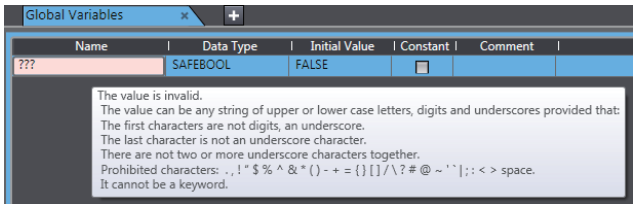
**3** Press the **Ctrl** + **V** Keys. Or, right-click and select **Paste** from the menu.
A copy of the variable is registered with *_Copy* added to the name of the variable that was copied on the next row.



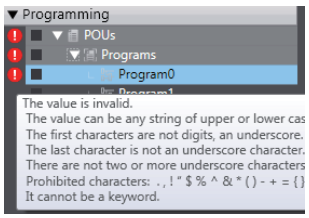

### Precautions for Correct Use

If you enter any invalid characters or out of range values, the cell is highlighted in pink. An error will occur when the program is built. A red icon "!" is displayed in the Multiview Explorer.
The error message is displayed when the mouse cursor is moved over the cell where the error exists or over the exclamation icon.
Refer to *6-2-8 Restrictions on Variable Names and Other Safety Program-related Names* on page 6-17 for details on the restrictions on variable names.

- Variable Table




- POUs (when the mouse cursor is moved over a POU, including the local variable where the error exists)




### Additional Information

- The global variable comments are linked to the device variable comments in the I/O Map and the I/O terminal comments on the Parameters Tab Page for the Safety Slave Unit.
- If the same variable names exist when variables are copied and pasted from other variable tables or spreadsheets, a dialog box that lists the source and destination variables is displayed. Select the variable to overwrite and execute the paste.

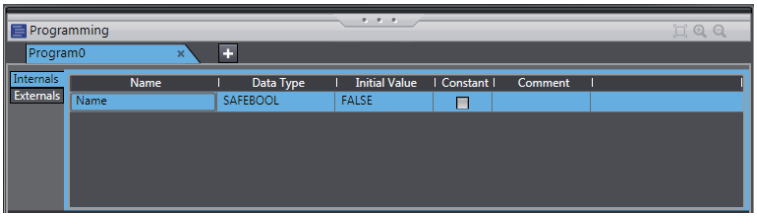
## Registering Local Variables

Registration of local variables refers to the registration of variables that can be used only inside POUs (programs and function blocks). Local variables include internal variables, input variables, output variables, and external variables.

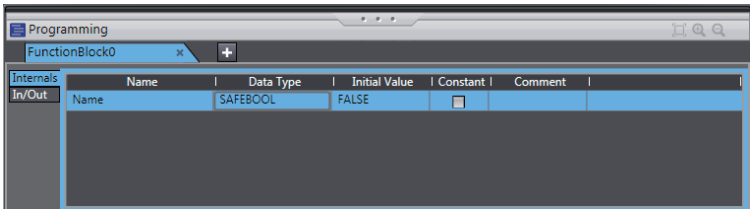
### ● Registering and Editing Local Variables

**1** Double-click a program under **Programming** - **POUs** - **Programs** in the Multiview Explorer. Or, right-click the program and select **Edit** from the menu.
The local variable table for the program or the local variable table for the function block is displayed in the Edit Pane.

- The local variable table for programs is shown below.




- The local variable table for function blocks is shown below.




| Field | Description | Restrictions |
|---|---|---|
| Name | Enter a name to use to identify the variable. | Only single-byte alphanumeric characters are allowed.<br>Multi-byte characters, such as those used for Japanese, are not allowed.<br>The maximum size is 127 bytes. |
| Data Type | Set the type of data that is stored in the variable. Refer to *6-2-5 Data Type* on page 6-14 for the data types that you can use. | --- |
| Initial Value | Set the value to use when the power is turned ON, when the mode changes to RUN mode, or DEBUG mode (STOPPED). This parameter must be specified. | --- |
| Constant | Select the check box in the Constant column to set the initial value of the variable when it is downloaded, but prevent it from being changed afterward.<br>Select the check box or clear the selection (default). | --- |
| Comment | Set any comments for the variable. | The maximum size is 127 bytes. |

**Note** The items that can be set and viewed depend on the type of the local variable. Refer to *6-2-4 Attributes of Variables* on page 6-13 for details.

**2** Select the tab for internal variables, in-out variables (function blocks only), or external variables, and then register and edit the local variables.
You can also register them directly in the FBD editor.

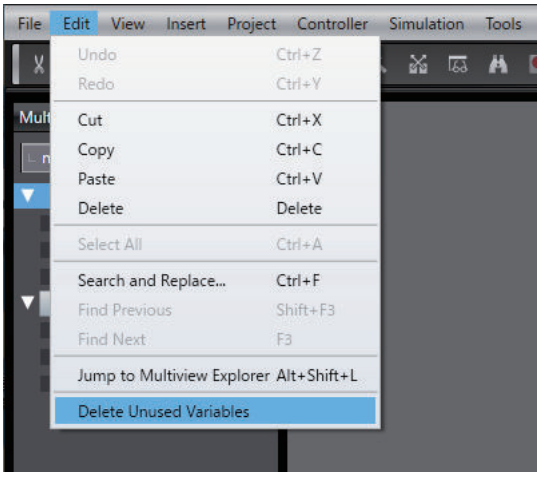
**Additional Information**

The operating procedures for local variables are the same as the procedures used for global variables.
Refer to *6-5-3 Registering Variables* on page 6-37.

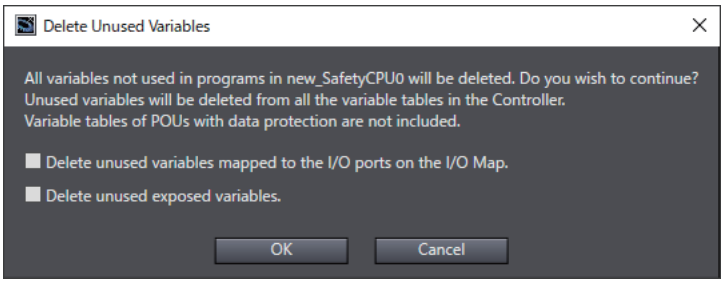
## Deleting Unused Variables

You can delete unused variables in the Safety CPU Unit programs all at once.

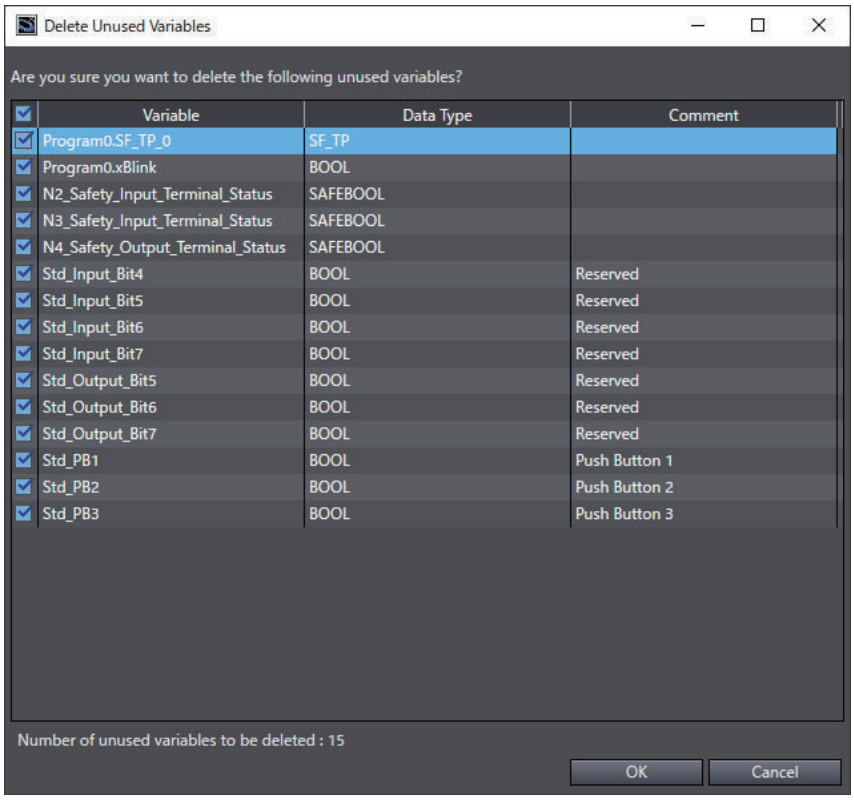
**1** Select **Delete Unused Variables** from the **Edit** menu.



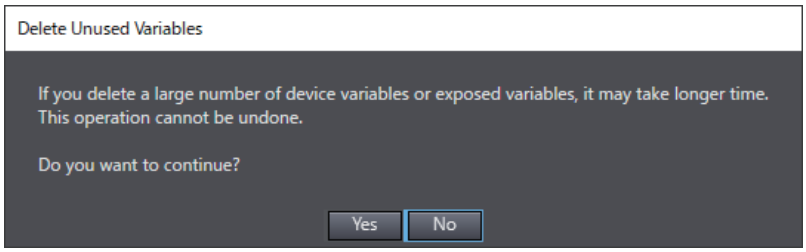

The confirmation message is displayed.



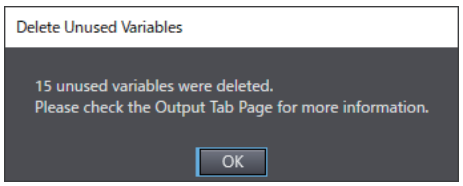

**2** To include unused variables mapped to the I/O ports on the I/O Map and/or unused exposed variables to the deletion target, select the corresponding checkbox(es) and click the **OK** Button.
A list of unused variables is displayed.




***3*** Select the checkbox(es) of the variable(s) to delete and click the **OK** Button.
The following confirmation dialog is displayed.




***4*** Check the contents and click the **Yes** Button.
The unused variables are deleted. The window displays the number of deleted entries.




### Precautions for Correct Use

- This deletion does not work for variables with data protection enabled and the I/O variables of function blocks.
- The deletion also does not work for the exposed variables with the device variables assigned by the standard controllers.

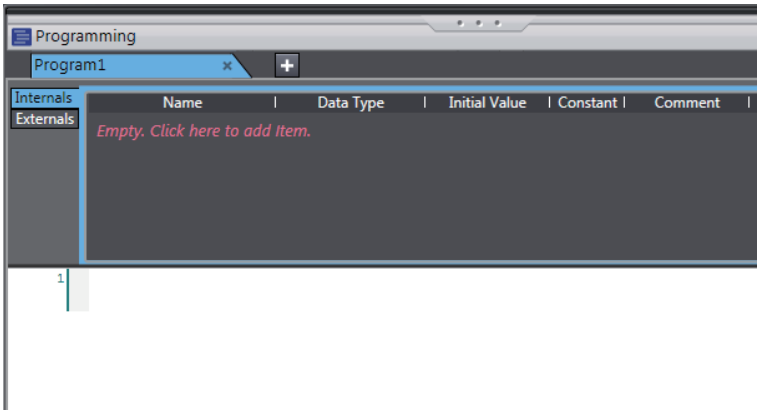
## 6-5-4 FBD Programming

With the Safety CPU Unit, you use the FBD language to express algorithms that are inside the POUs (programs and function blocks). You add and connect functions and function blocks in the FBD editor to build algorithms inside POUs (programs and function blocks).
This section describes how to use the FBD editor.

### Opening and Using the FBD Editor

#### ● Programs

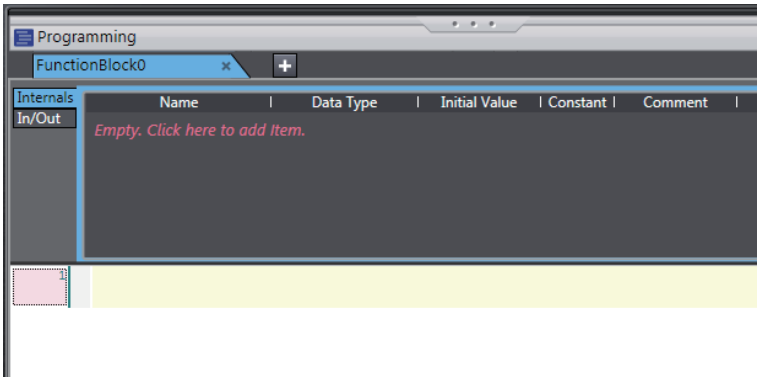
**1** Double-click a program under **Programming** - **POUs** - **Programs** in the Multiview Explorer. Or, right-click the program and select **Edit** from the menu.
The FBD editor for the program is displayed.




Refer to *6-5-2 Registering POUs* on page 6-27 for the program registration procedure.

#### ● Function Blocks

**1** Double-click a registered function block under **Programming** − **POUs** − **Function Blocks** in the Multiview Explorer. Or, right-click the function block and select **Edit** from the menu.
The FBD editor for the function block is displayed.




Refer to *6-5-2 Registering POUs* on page 6-27 for the function block registration procedure.

## Zooming In and Zooming Out of the FBD Editor

Use the icons that are displayed in the toolbar to zoom in or zoom out of the FBD editor.



## Inserting FBD Networks

There are three ways to insert an FBD network.

### ● Method 1

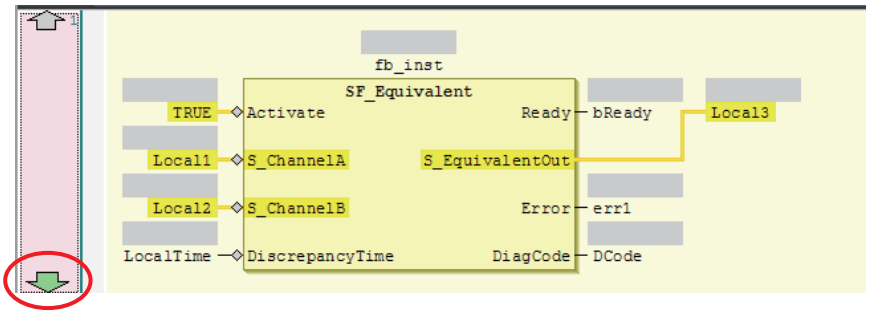
Right-click the FBD network and select **Insert Network Above** from the menu.
An empty FBD network is inserted before the selected FBD network.

### ● Method 2

Right-click the FBD network and select **Insert Network Below** from the menu.
An empty FBD network is inserted after the selected FBD network.

### ● Method 3

Drag a **Network** from **General** in the Toolbox to the FBD editor.
An empty network is inserted at one of the positions given in the following table, and the focus moves to the inserted network.




| Drop point | Position where network is added |
|---|---|
| Upward arrow on the network number | An empty network is inserted before the FBD network where the network was dropped. |
| Downward arrow on the network number | An empty network is inserted after the FBD network where the network was dropped. |
| Input terminal | An empty network is inserted after the FBD network where the network was dropped. |

## Deleting FBD Networks

There are two methods to delete an FBD network.

### ● Method 1

Right-click the FBD network and select **Delete** from the menu.

The selected FBD network is deleted and the focus moves to the next network.

### ● Method 2

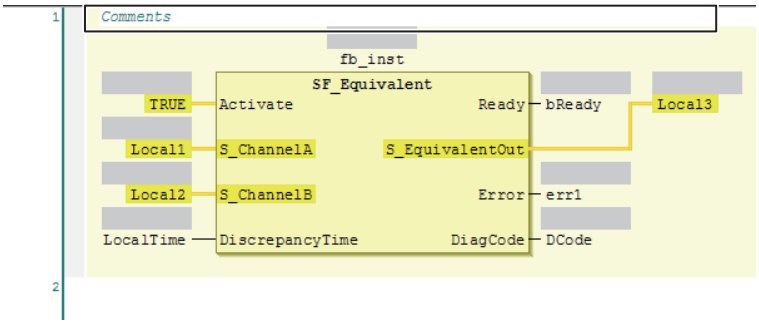
Select the FBD network and press the **Delete** Key.

The selected FBD network is deleted and the focus moves to the next network.

## Editing Comments for FBD Networks

Use the following procedure to edit the comment for an FBD network.

- Select the comment portion of the FBD network and edit it.




## Commenting Out FBD Networks and Restoring Them

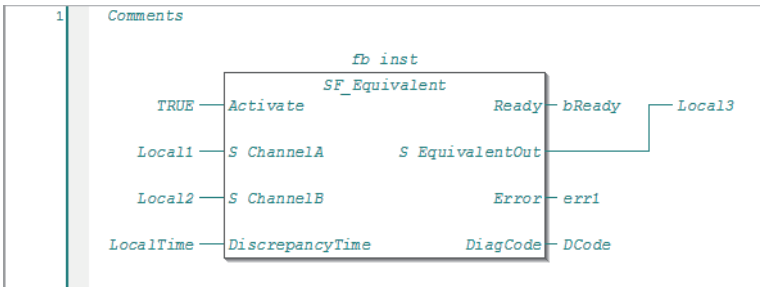
The following operation allows you to comment out an FBD network and then restore it. When a network is commented out, it is no longer executed.

Right-click the FBD network and select **Toggle Network Comment State** from the menu.

**Precautions for Correct Use**

You cannot select more than one network. If you select more than one network, the comment status of last network that you select will change.

Networks that are commented out are displayed in blue italic letters as shown below.




If you select a commented network, the network is changed to an uncommented network.

## Inserting a FUN or FB

There are following two ways to insert a FUN or FB.

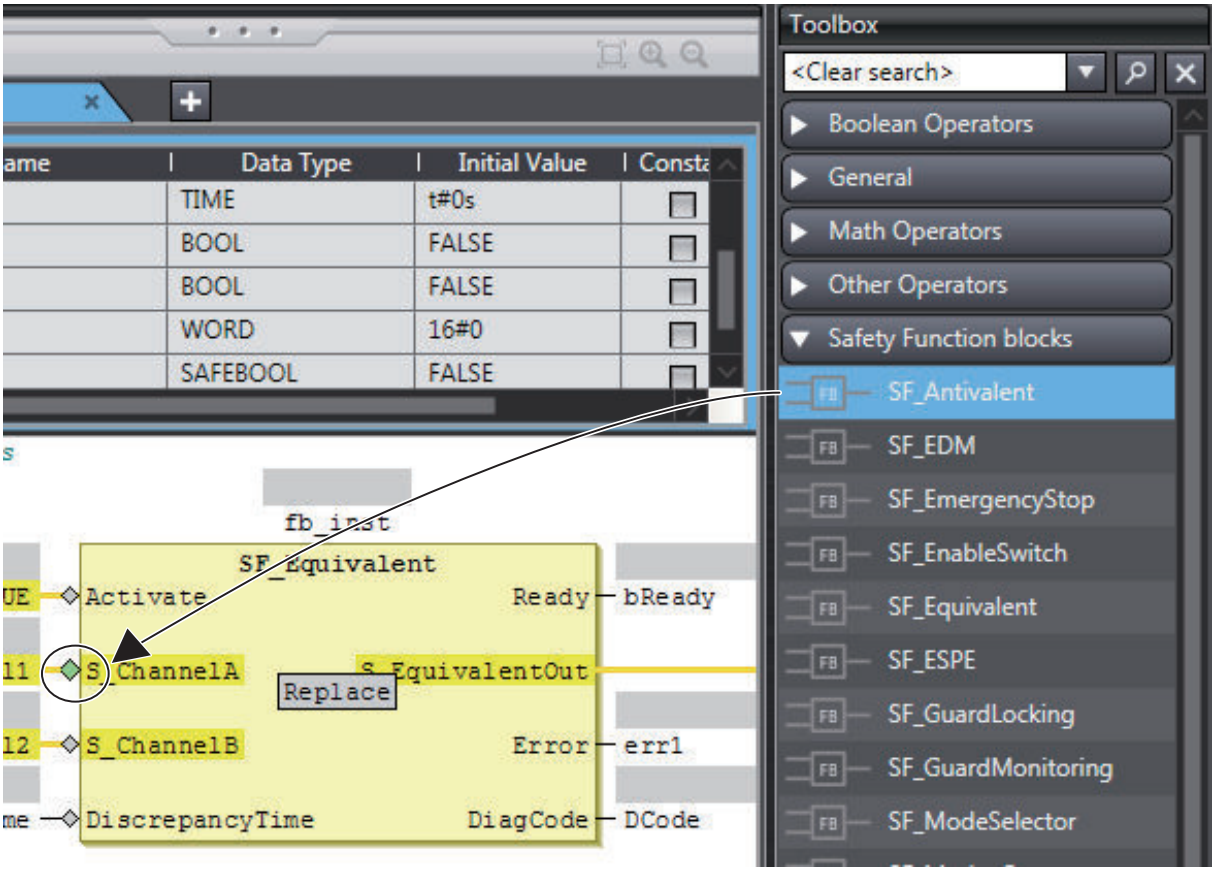
| Method | Operation |
|---|---|
| Method 1 | Drag a FUN or FB from the Toolbox to an I/O terminal on the FUN or FB in the FBD editor, or to a new network where the words **Start Here** are displayed. |
| Method 2 | Right-click the FBD network, select **Insert Function Block** from the menu, and specify FUN or FB. |

**Precautions for Correct Use**

Not all of the FUNs and FBs that you can use are displayed in the Toolbox. To use a FUN or FB that is not shown in the Toolbox, use Method 2.

## ● Procedure for Method 1

***1*** Drag a FB or FUN from the Toolbox to an I/O terminal on the FUN or FB in the FBD editor, or to a new network where the words **Start Here** are displayed. Positions where you can drop the function block are indicated with gray rectangular or diamond-shape boxes. These boxes change to green when you move the cursor over them.

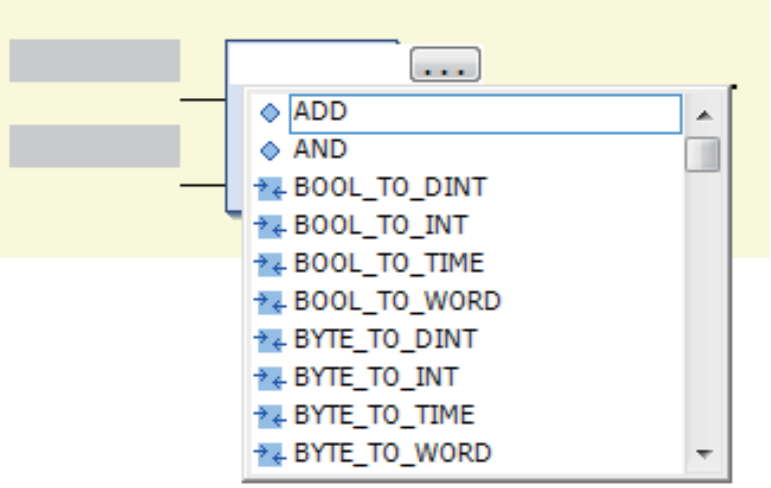



***2*** Drop the FUN or FB on a green diamond-shape box to insert it.

## ● Procedure for Method 2

***1*** Right-click the FBD network and select **Insert Function Block** from the menu.
An empty FB is inserted.

**2** Click in the FB name text box and press the **Ctrl** + **Space** Keys to display a list of the FUNs and FBs that you can enter.




**3** Select a FUN or FB from the list.

### Additional Information

You can click the Input Assistance Button (...) to the right of the FB name text box to display the **Input Assistant** Dialog Box. You can select an FB from the **Items** in this dialog box to insert the selected FB.

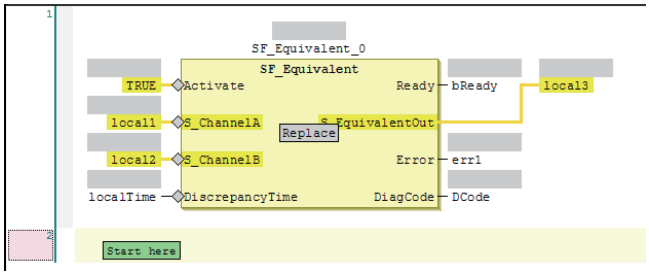
| Area | Description |
|---|---|
| Categories | Displays the FB categories. The FBs that belong to the selected category are displayed in the **Items** Area. |
| Items | Displays a list of FBs. |
| Documentation | Any additional information that is available for the FB that you select in the **Items** Area is displayed. |

The **Input Assistant** Dialog Box is also displayed when you right-click an FBD network and select **Insert Function Block** from the menu.

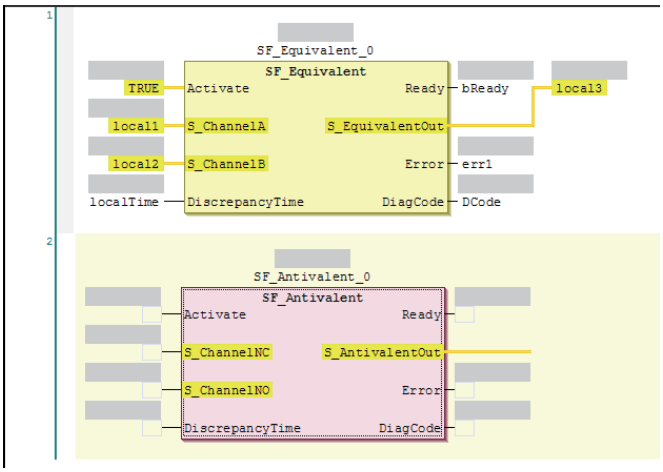
## ● Inserting Position of a FUN or FB

The position where the FUN or FB is inserted depends on the insertion method, as described below.

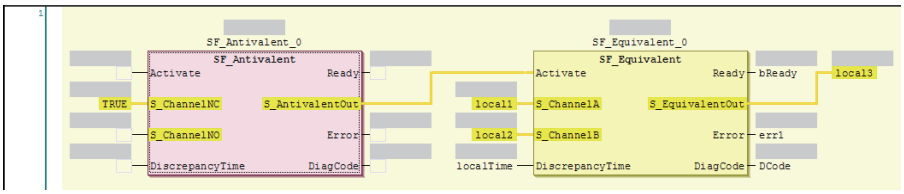
- If you drag a FB or FUN (SF_Antivalent in the example) from the Toolbox, the FB or FUN is inserted at the position shown below.



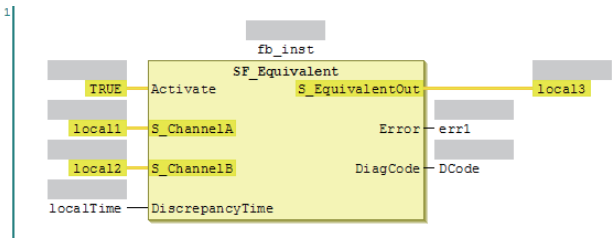
If you drop the SF_Antivalent FB on the network where the words **Start here** are displayed, the FB is inserted in the new network.



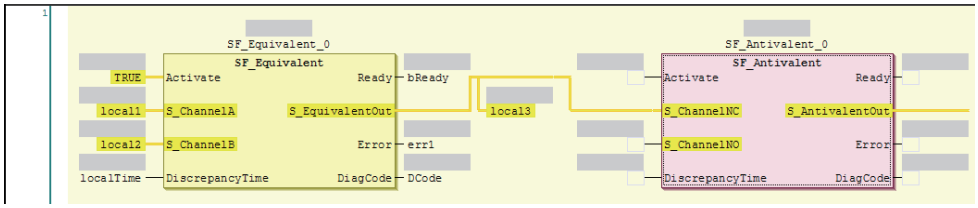

If you drop the SF_Antivalent FB on an input terminal, the FB is inserted before the input terminal.




- If you select a FB or FUN (in this example, SF_Antivalent) from the right-click menu, the FUN or FB is inserted at the location shown below.
  a) Before the FB is inserted, the network is as shown below.




  b) When the SF_Antivalent is selected from the right-click menu, the network is as shown below.




## Deleting a FUN or FB

Use one of the following procedures to delete a FUN or FB.

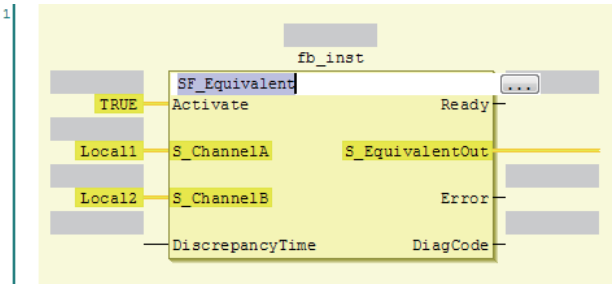
| Method | Operation |
|---|---|
| Method 1 | Right-click the FUN or FB on the FBD network and select **Delete** from the menu. |
| Method 2 | Select the FUN or FB on the FBD network and press the **Delete** Key. |

## Replacing a FB or FUN

You can replace a FUN or FB with a different instruction without changing the input and output parameters. Use one of the following procedures.

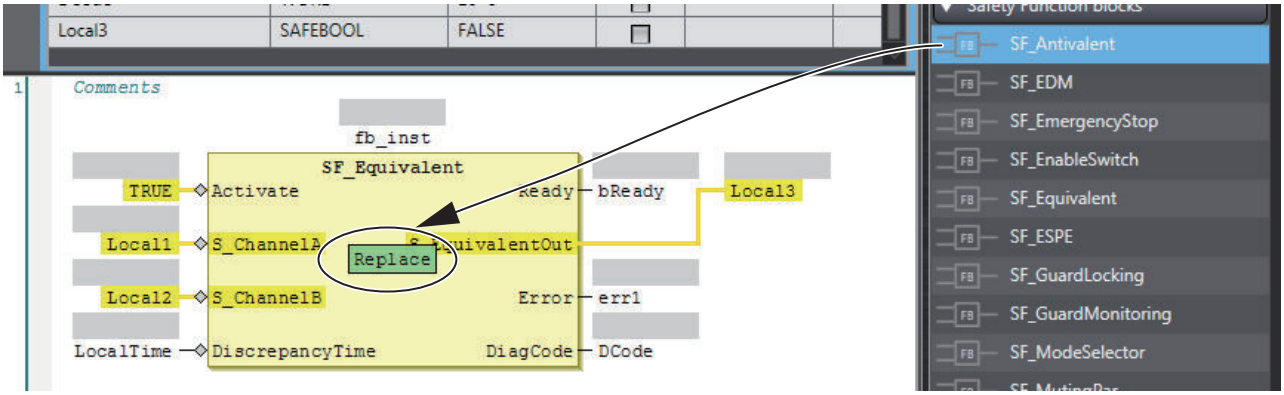
| Method | Operation |
|---|---|
| Method 1 | Drag the FUN or FB to change to in the Toolbox to the FUN or FB in the FBD editor. |
| Method 2 | Select the FUN or FB name on the FBD network and directly enter the FUN or FB name. |

- Before the FB is edited, the network is as shown below.




### ● Procedure for Method 1

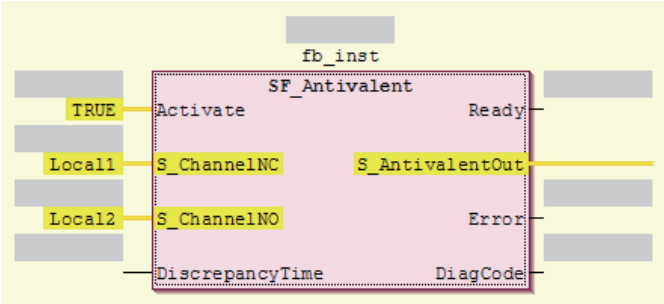
**1** Drag a FUN or FB from the Toolbox to the FB to replace. A **Replace** area is displayed.




**2** Drop the FUN or FB in the **Replace** area to replace the FUN or FB.

### ● Procedure for Method 2

If you directly enter the FB or FUN name, the FB or FUN is replaced when you press the **Enter** Key.

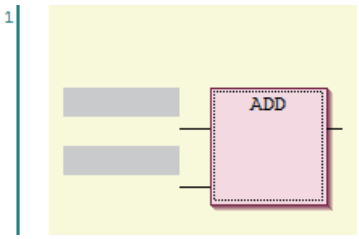



## Adding Input Parameters to AND, OR, ADD, MUL, and MUX

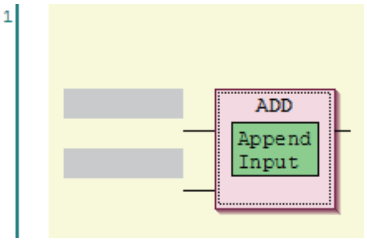
You can add input parameters to the AND, OR, ADD, MUL, and MUX instructions. Use one of the following procedures.

| Method | Operation |
|---|---|
| Method 1 | Drag **Input** from **General** in the Toolbox to the **Append Input** area in the instruction in the FBD editor. |
| Method 2 | Right-click the FB on the FBD network and select **Add Input** from the menu. |

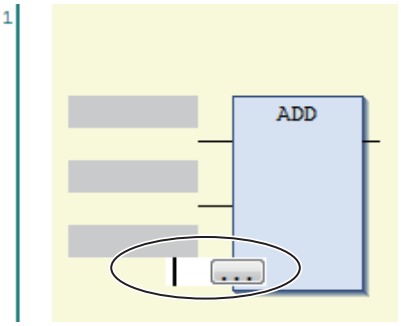
- Before the input parameter is added, the network is as shown below.




- When you drag an **Input Variable** from the toll box, the network is as shown below.




- After the input parameter is added, the network is as shown below.



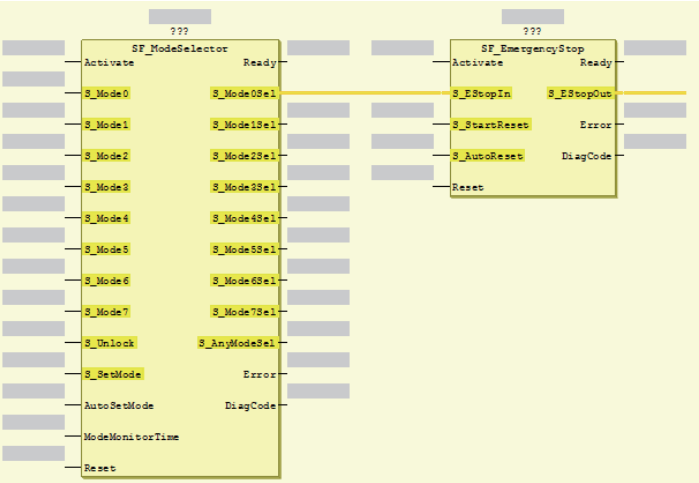
## Changing the Output Terminals of a FUN or FB

Use the following procedure to change the output terminals of a FUN or FB.

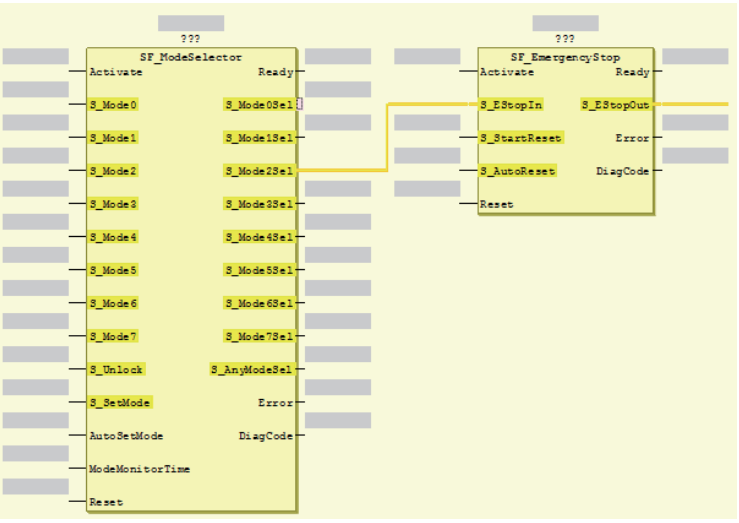
Right-click an output terminal on a FUN or FB on the FBD network and select **Set Output Connection** from the menu.

The selected output terminal is connected to the input terminal of the next FUN or FB.

- Before the output terminal is changed, the network is as shown below.




- When **S_Mode2Sel** is selected and **Set Output Connection** is executed, the network is as shown below.



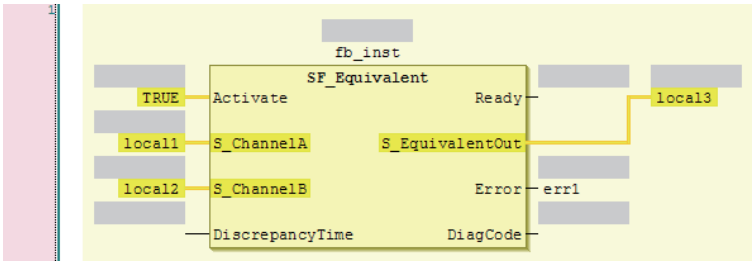

## Deleting Unused Parameters from a FUN or FB

Use the following procedure to delete any unused parameters from a FUN or FB.

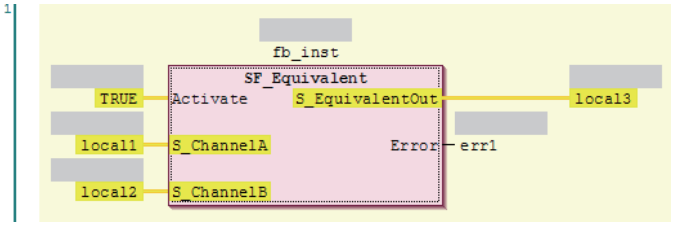
Right-click the FB or FUN on the FBD network and select **Remove unused FB call parameters** from the menu.

All of the unused parameters are deleted.

- Before the unused parameters of the FUN or FB are deleted, the network is as shown below.




- After the unused parameters of the FUN or FB are deleted, the network is as shown below.



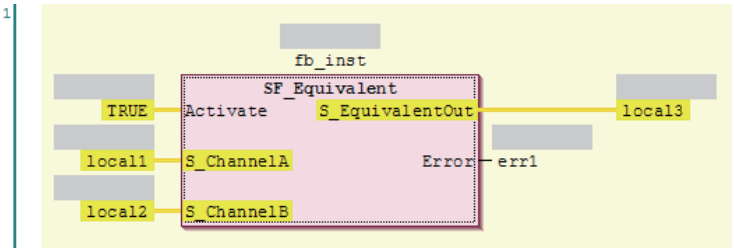

## Updating the Input Parameters and Output Parameters of a FUN or FB

Use the following procedure to display the input parameters and output parameters of a FUN or FB.

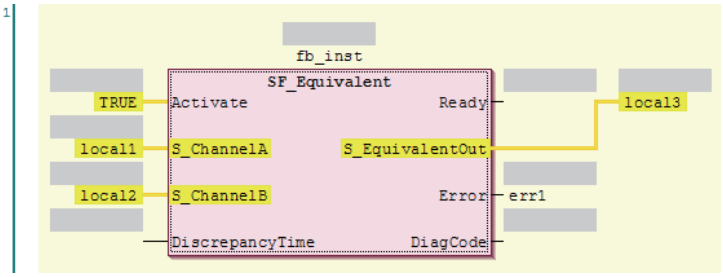
Right-click a FUN or FB on the FBD network and select **Update** from the menu.

The input parameters and output parameters for the FUN or FB are displayed along with any unused parameters.

- Before the input parameters and output parameters of the FUN or FB are updated, the network is as shown below.




- After the input parameters and output parameters of the FUN or FB are updated, the network is as shown below.

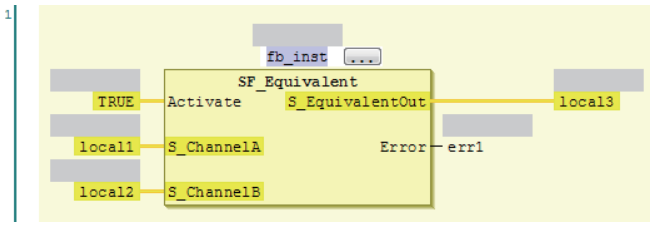


## Editing Function Block Instance Variables

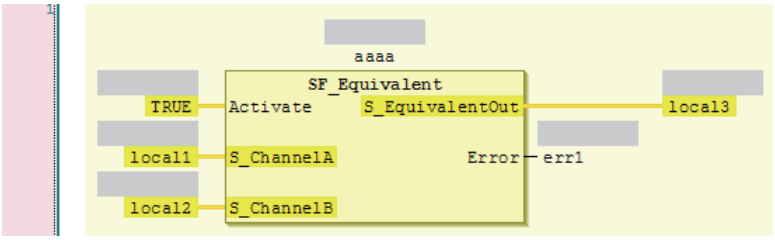
Use one of the following methods to edit existing function block instance variables.
If you specify a variable name that does not exist in the local variable table, that variable will be registered as a local variable.

Select an FB instance variable on the FBD network and directly enter the variable name.

- Before the function block instance variable is edited, the network is as shown below.




- After the function block instance variable is edited, the network is as shown below.




**Additional Information**

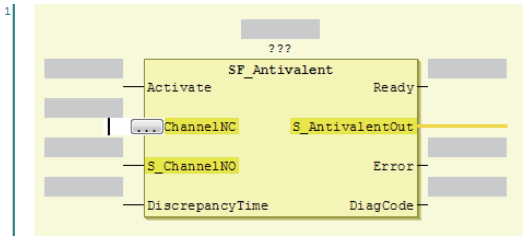
To create a new function block instance variable, enter the variable name and press the **Enter** key. The function block instance variable is registered as an instance of the FB, and it is also registered as a local variable in the local variable table.

## Entering Parameters

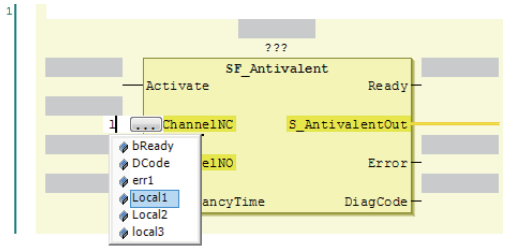
Use one of the following procedures to enter parameters.
If you specify a variable name that does not exist in the local variable table, that variable will be registered in the local variable table.

Select a parameter on the FBD network and directly enter the variable name.

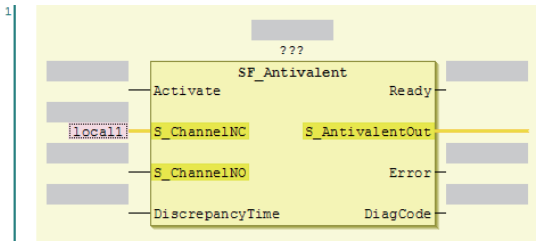
- Before the parameter is edited, the network is as shown below.



If you enter the variable name directly, a list of variable names that you can select from is displayed when you enter the first letter of the variable name. Use the **Up** and **Down** Keys to select the variable name from the list, and then press the **Enter** Key. The selected variable name is registered as an input parameter. If you press the **Ctrl** + **Space** Keys when nothing is displayed, list of variable name candidates is displayed.




- After the input parameter is edited, the network is as shown below.




To delete an input parameter assigned, select the parameter and press the **Delete** key.

**Additional Information**

- To create a new input variable, enter the variable name and press the **Enter** key. The input parameter is registered, and it is also registered as a local variable in the local variable table.
- You can click the Input Assistance Button (...) to the right of the text box to display the **Input Assistant** Dialog Box. You can select a variable from the **Items** in this dialog box to insert the selected variable.

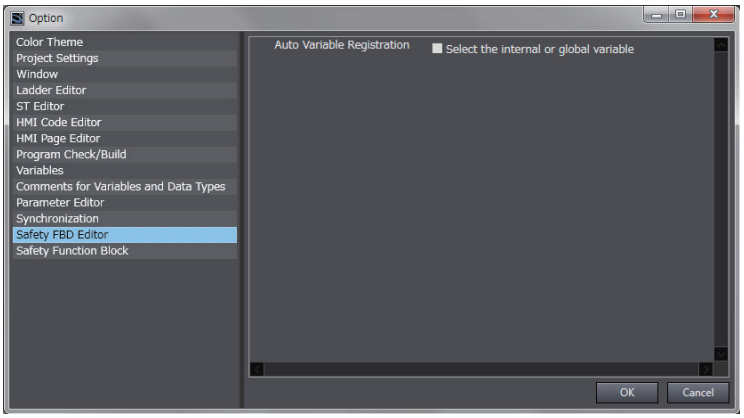
| Area | Description |
|---|---|
| Categories | Displays the variable categories. The variables that belong to the selected category are displayed in the **Items** Area. |
| Items | Displays a list of variables. |
| Documentation | Any additional information that is available for the variable that is selected in the **Items** Area is displayed. |

## Changing Variable Registration Locations

You can use the following option to enable the selection of the variable registration locations when you enter parameters.

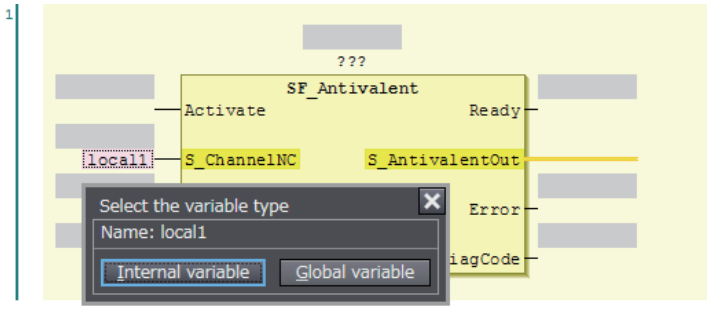
***1*** Select **Option** from the **Tools** Menu.
The **Option** Dialog Box is displayed.

***2*** Click **Safety FBD Editor**.
The Auto Variable Registration Option is displayed.




**3** Select the **Select the internal or global variable** Check Box and click the **OK** Button.

If you select the above option, the following dialog box is displayed when a variable is registered automatically from the Safety FBD Editor.




If you click the **Internal variable** Button, the variable is registered as an internal variable in the local variable table.
If you click the **Global variable** Button, the variable is registered in the global variable table and as an external variable in the local variable table.

## Assigning Output Parameters

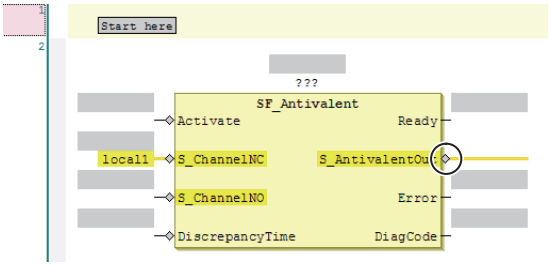
You can insert output variables at specified locations. Use one of the following procedures to assign an output parameter to an output variable of a FUN or FB.

| Method | Operation |
|---|---|
| Method 1 | Drag **Assignment** from **General** in the Toolbox to a terminal in the FBD editor, or to a new FBD network where the words **Start here** are displayed. |
| Method 2 | Right-click the FBD network and select **Insert Assignment** from the menu. |

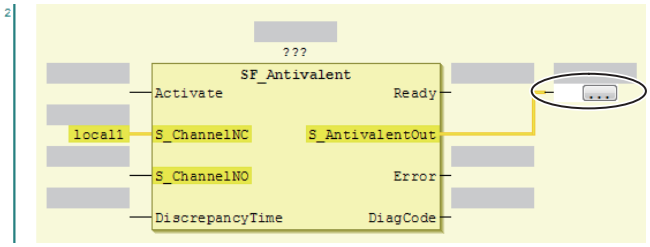
- If you select a network, the output variable is added to the last output area (before the output terminal or the output variable) on the network.
- If you drag **Assignment** from the Toolbox, the point of insertion depends on where you drop the output variable.

| Drop point | Position where function block is added |
|---|---|
| “Start here” on the network | Input parameters and output parameters are added to the new network. |
| Input terminal | The output parameter is inserted on a branch that is created before the input terminal. |
| Output terminal | The output parameter is inserted after the output terminal. |
| Before the output parameter | The output parameter is inserted on a branch that is created before the output variable. |

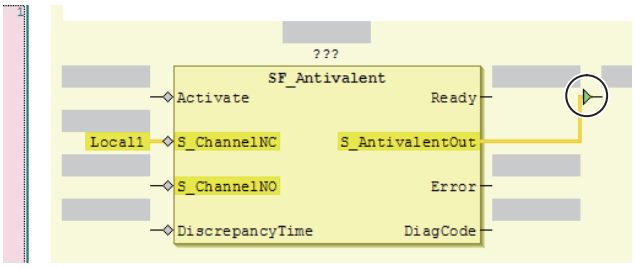
## ● Example 1 of a Dragged Output Parameter Object



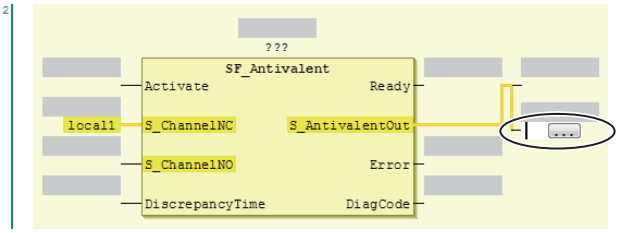

When an output parameter is added to the output terminal, the network is as shown below.




## ● Example 2 of a Dragged Output Parameter Object




When an output parameter is added before another output parameter, the network is as shown below.




The output parameter is inserted on a branch that is created before the output variable.

## Deleting Output Parameters

Use the following procedure to delete output parameters.

Right-click the Output parameters on the FBD network and select **Delete** from the menu.

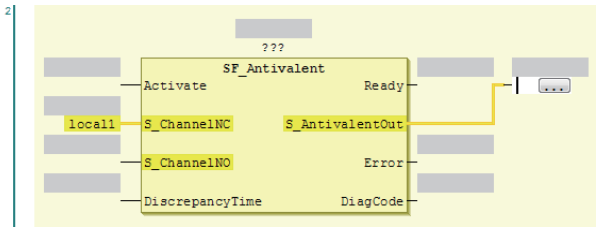
The selected output parameter is deleted. You cannot select multiple variables.

## Editing Output Parameters

Use one of the following procedures to edit output parameters.
If you specify a variable name that does not exist in the local variable table, that variable will be registered as a local variable.

Select the output variable on the FBD network and directly enter the variable name.

Before the output variable is edited, the network is as shown below.



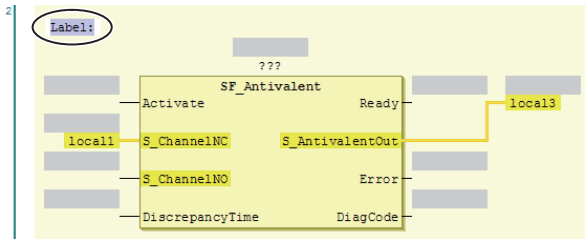

- If you prefer to enter the variable name directly, a list of variable names that you can select from is displayed when you begin entering the variable name.
- To create a new output variable, enter the variable name and press the **Enter** Key. The output variable is registered, and it is also registered as a local variable in the local variable table.

## Inserting Jump Labels

Use the following procedure to insert jump labels into an FBD network.

Right-click the FBD network and select **Insert Jump Label** from the menu.

After the jump label is added, the network is as shown below.




You cannot add another jump label to a network if it already has one.

## Deleting Jump Labels

Use one of the following procedure to delete jump labels.

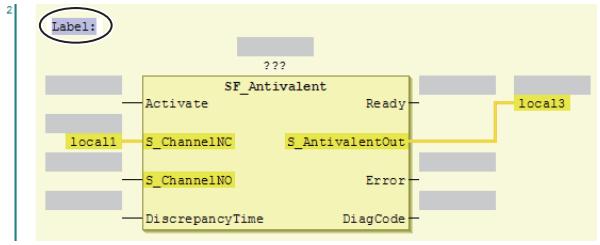
| Method | Operation |
|---|---|
| Method 1 | Right-click the Jump label and select **Delete** from the menu. |
| Method 2 | Select the Jump label and press the **Delete** Key. |

## Editing Jump Labels

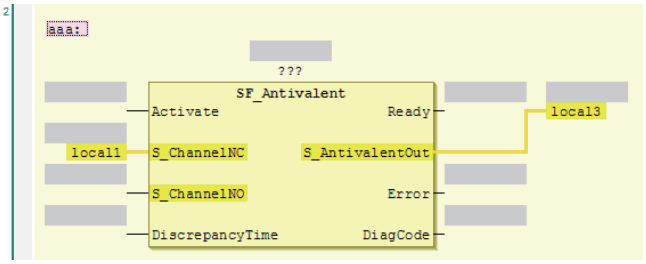
Use the following procedure to edit jump labels.

Select a jump label on the network and edit it.

- The jump label becomes editable when you click it. After you finish editing, press the **Enter** key.
  a) After the jump label is selected, the network is as shown below.




  b) After the jump label is edited, the network is as shown below.



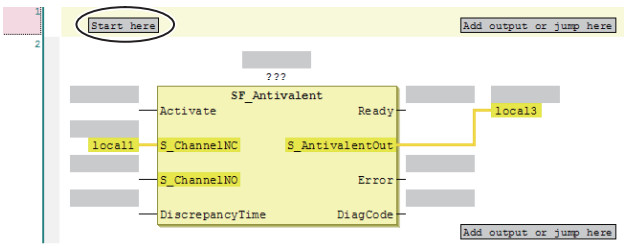

## Inserting Jump Instructions

Use one of the following procedures to insert a Jump instruction in a network.

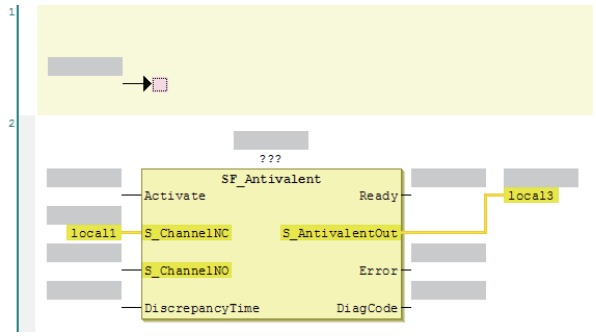
| Method | Operation |
|---|---|
| Method 1 | Drag **Jump** from **General** in the Toolbox to the words **Start here** on a new network. |
| Method 2 | Right-click the FBD network and select **Insert Jump** from the menu. |

The Jump instruction is inserted in the network.

- Before the Jump instruction is dropped, the network is as shown below.




- After the Jump instruction is inserted, the network is as shown below.




## Deleting Jump Instructions

Use one of the following procedures to delete a Jump instruction.

| Method | Operation |
|---|---|
| Method 1 | Right-click the Jump Instruction select **Delete** from the menu. |
| Method 2 | Select the Jump Instruction and press the **Delete** Key. |

The selected Jump instruction is deleted.

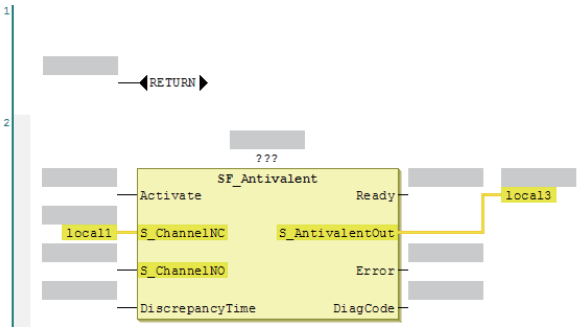
## Inserting Return Instructions

Use one of the following procedures to insert a Return instruction in a network.

| Method | Operation |
|---|---|
| Method 1 | Drag **Return** from **General** in the Toolbox to a terminal in the FBD editor, or to a new FBD network where the words **Start here** are displayed. |
| Method 2 | Right-click the FBD network and select **Insert Return** from the menu. |

The Return instruction is inserted in the network.

- After you insert a return instruction, the network is laid out as shown below.




## Deleting Return Instructions

Use one of the following procedures to delete a Return instruction.

| Method | Operation |
|---|---|
| Method 1 | Right-click the Return instruction and select **Delete** from the menu. |

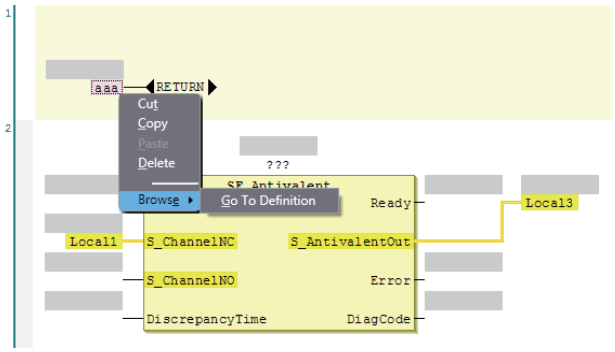
| Method | Operation |
|---|---|
| Method 2 | Select the Return instruction and press the **Delete** Key. |

The selected Return instruction is deleted.

## Viewing the Locations of Variable Definitions

You can view where variables are defined.
Use the following procedure.

Right-click the variable and click **Browse** − **Go To Definition** from the menu.

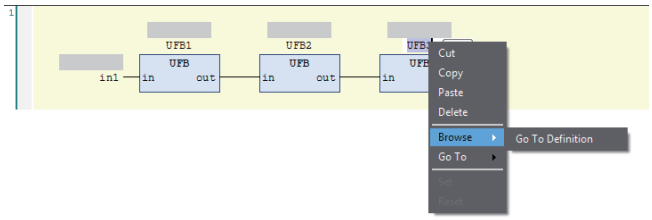



The focus moves to the location where the variable is defined (local variable).

## Viewing the Locations of User-defined Function Block Definitions

You can view where user-defined function blocks are defined.
Use the following procedure.

Right-click the user-defined function block and click **Browse** − **Go To Definition** from the menu.

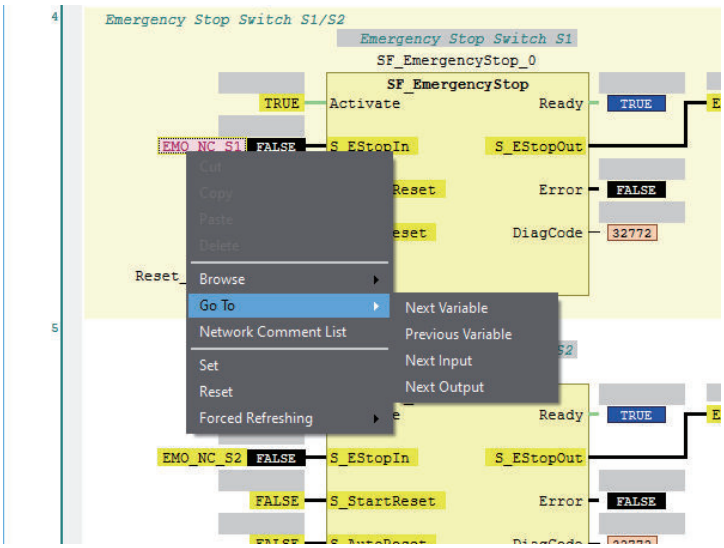



The focus moves to the location where the user-defined function block is defined.

## Jumping to Variables in POUs

You can search sequentially for a variable used in the POUs (programs and function blocks) and then sequentially jump to the locations where the variable is used.
Use the following procedure.

Right-click a variable in the POU and select the destination to jump to from the menu.

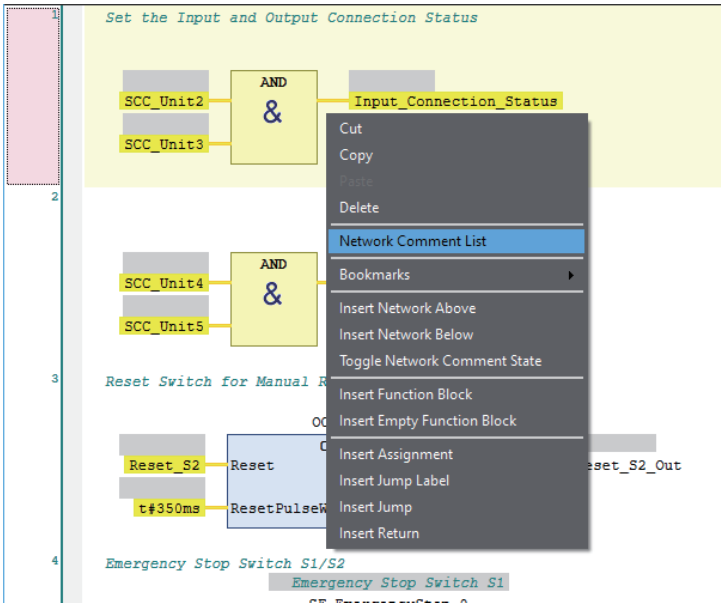



| Menu command | Description |
|---|---|
| Next Variable | The display will jump to the next location where the selected variable is used. |
| Previous Variable | The display will jump to the previous location where the selected variable is used. |
| Next Input | The display will jump to the next location where the selected variable is used as a program input. |
| Next Output | The display will jump to the next location where the selected variable is used as a program output. |
| (Retrace Search*1) | If the selected variable is used as an output in the program, the search will start from the beginning of the program to look for a location where the selected variable is used as an input, and the display will jump to the location.<br>If the selected variable is used as an input in the program, the display will jump to the location where the selected variable is used as a program output. |

*1. This item is not displayed on the menu. Use the shortcut key **Space**.

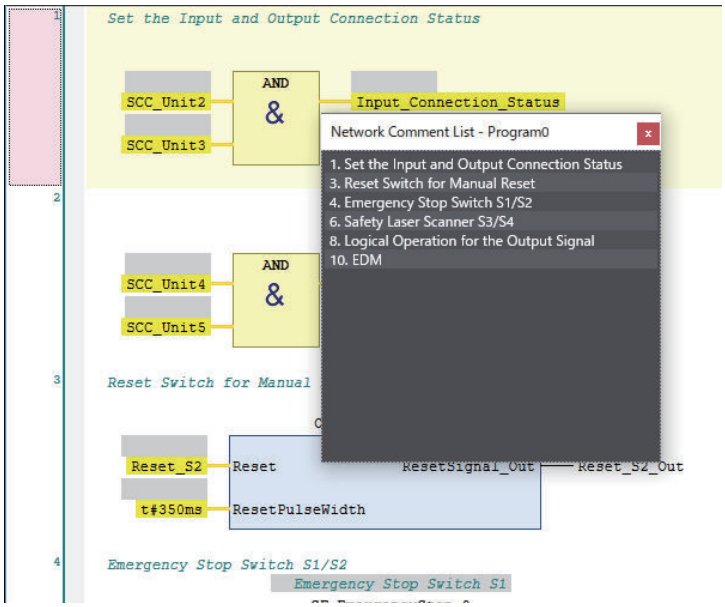
## Displaying Network Comment List

You can display the list of network comments that are set in the POU (programs and function blocks), and then jump to a specific network comment.
You can use the following two methods to display the network comment list.

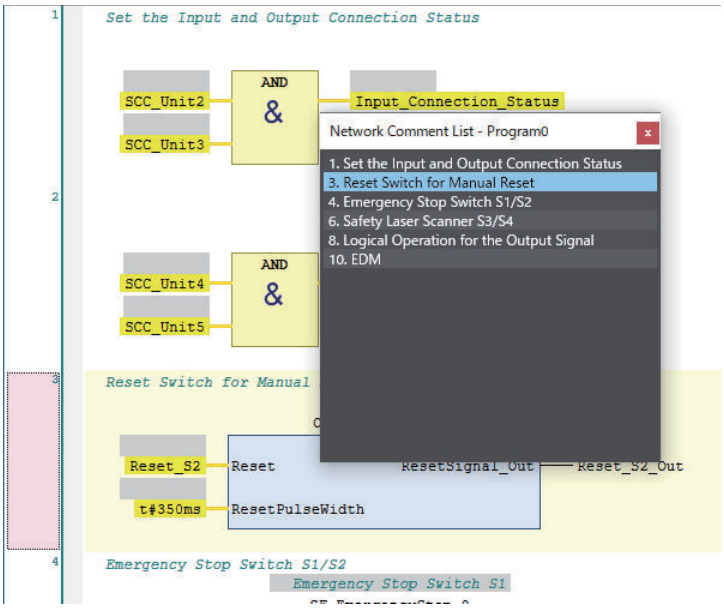
| Method | Description |
|---|---|
| Method 1 | Right-click on the FBD editor, and then select **Network Comment List** from the menu.  |
| Method 2 | On the FBD editor, press the **L** Key. |



The list of network comment that are set in the current POU is displayed.




Select a network comment to jump to the location where the selected network comment is set.

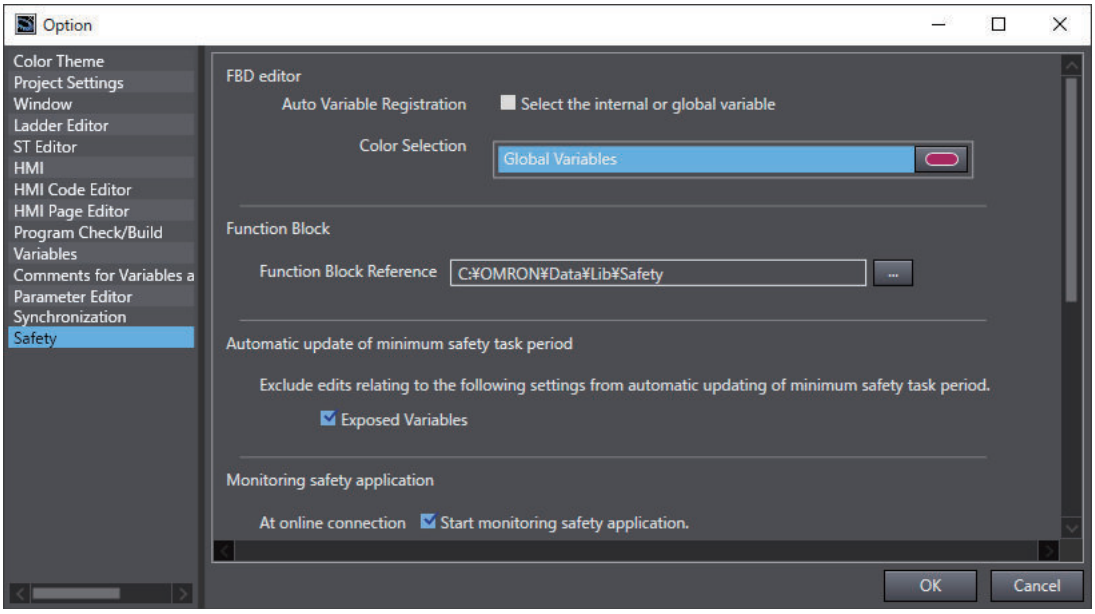



## Color Selection for the FBD Editor

You can change the display color of global variables that are displayed in the FBD editor.

**1** Select **Option** from the **Tools** menu.
The Option Dialog Box is displayed.

**2** Select the **Safety** tab and click the button displayed at Color Selection.

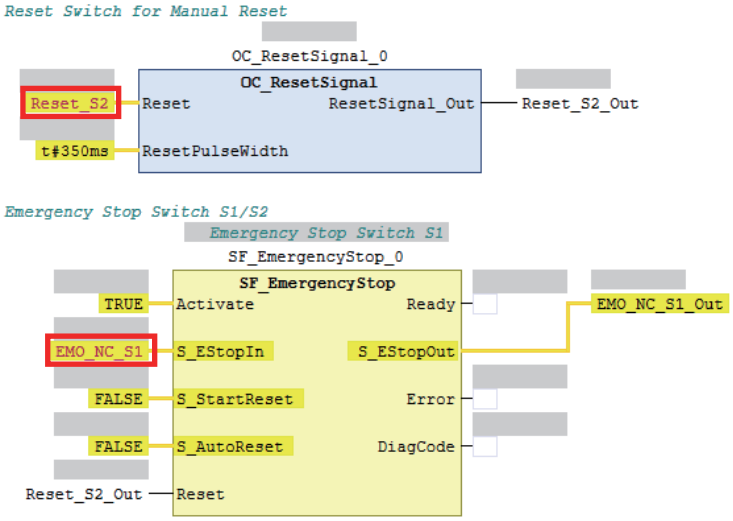



The Color Settings Dialog Box is displayed.




**3** Select any color and click the **OK** button.

**4** In the Option Dialog Box, click the **OK** button.
The color selection settings are applied.



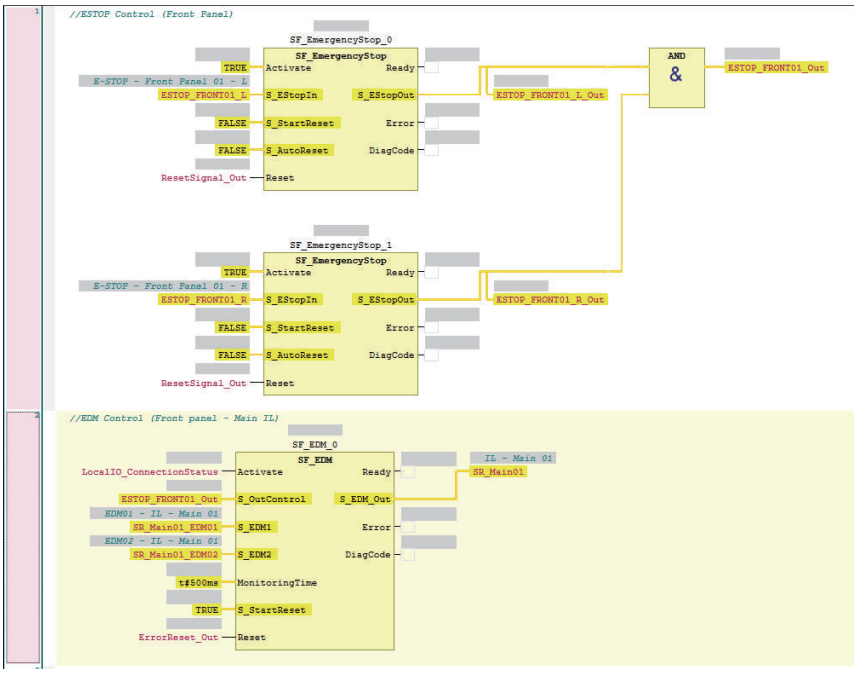

## 6-5-5 Program Pattern Copy

Program Pattern Copy is a function that allows you to easily replicate the FBD program having the same program pattern (logic part). The variable name of the replicated program can be automatically configured according to the variable name generation rules.

### Operating Procedure

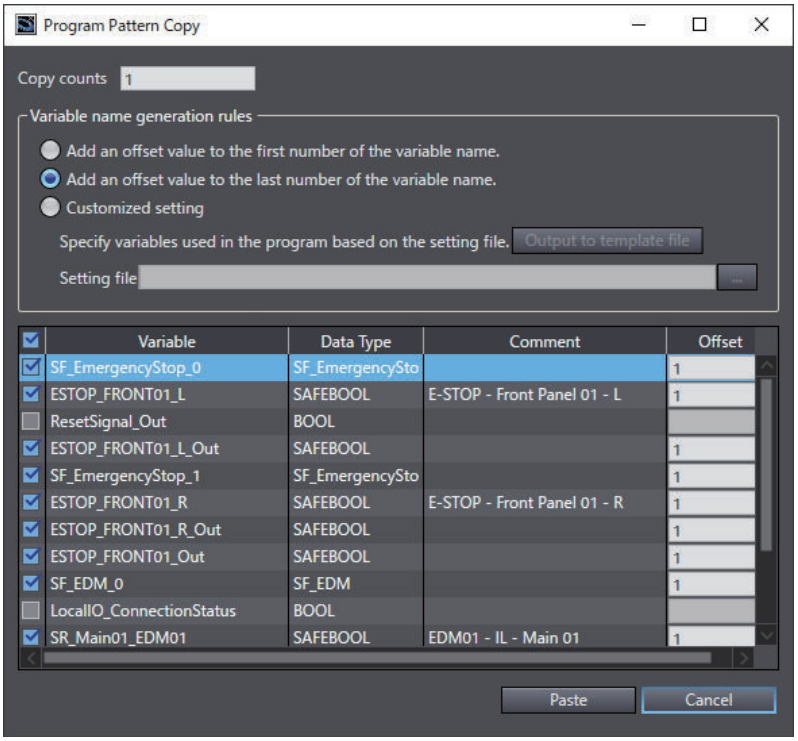
The procedure to copy a program pattern is explained below.

**1** Select any FBD network.
You can select more than one FBD network by holding down the **Shift** or **Ctrl** Key.




**2** Right-click a row and select **Program Pattern Copy** from the menu. Or, press the **Ctrl**+**Shift**+**V** Keys.

The following setting window is displayed.




**3** Set the copy counts, variable name generation rules, variable name to be changed, and offset value.

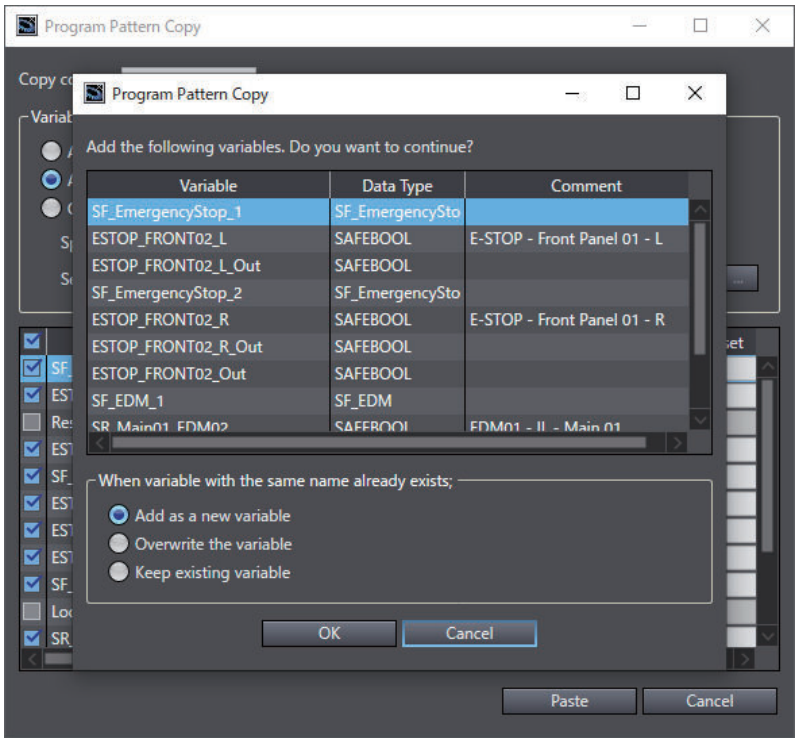
The setting items are given in the following table.

| Item | Description |
|---|---|
| Copy counts | Set the number of times that the program can be replicated. |

| Item | Description |
|---|---|
| Variable name generation rules | Select a generation rule for the variable name. The rules you can select are as follows:<br>• **Add an offset value to the first number of the variable name.**<br>This rule generates another variable name by adding the offset value to the first number contained in the variable name.<br>• **Add an offset value to the last number of the variable name.**<br>This rule generates another variable name by adding the offset value to the last number contained in the variable name.<br>• **Customized setting**<br>This rule generates any variable name and comment by loading a setting file. Refer to *File Format for Customized Setting* on page 6-69 for details. |
| Variable name to be changed (Checkbox) | Select a variable to be renamed. Unchecked variables use the same variable names as those in the source program.<br>Variables that do not include any number in the name are automatically excluded. Note that this setting is invalid when the variable name generation rules are set to **Customized setting**. |
| Offset | Set the offset value to be added to the variable name according to the variable name generation rules.<br>Variables that do not include any number in the names are automatically excluded. Note that this setting is invalid when the variable name generation rules are set to **Customized setting**. |

***4*** Click the **Paste** Button.

The following dialog is displayed when a variable is added by this function.

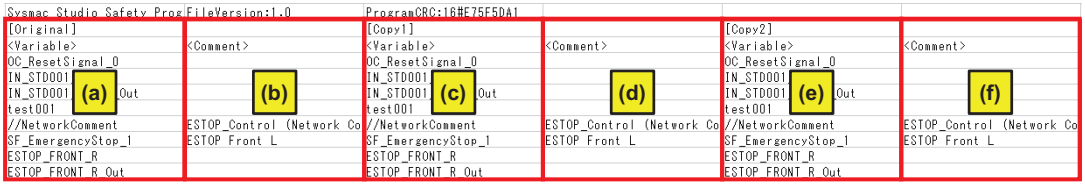



***5*** Select an option in **When variable with the same name already exists** and click the **OK** Button.

A replicated program is inserted under the selected FBD network. Check to see if the program and variables have been generated in the way you intended.

## File Format for Customized Setting

The file format for the customized setting is shown below. Since the setting file contains the CRC of the selected FBD network, use the CSV file saved with **Output to template file**.




| No. | Name | Description | Remarks |
|---|---|---|---|
| (a) | Source Variable Name | This is the variable name used in the source program to copy from. | Do not change this item. |
| (b) | Source Variable Comment | This is the variable comment used in the source program to copy from. | Do not change this item. |
| (c) | Copy 1 Variable Name | This is the variable name used for the first copy destination.<br>You can set any variable name. | |
| (d) | Copy 1 Variable Comment | This is the comment for the variable used for the first copy destination.<br>You can set any comment for the variable. | |
| (e) | Copy 2 Variable Name | This is the variable name used for the second copy destination.<br>You can set any variable name. | |
| (f) | Copy 2 Variable Comment | This is the comment for the variable used for the second copy destination.<br>You can set any comment for the variable. | |

The **Variable Name** and **Variable Comment** columns will be repeated continuously.

### Precautions for Correct Use

- After you edit the CSV file with a spreadsheet application, save the data in the CSV format (UTF-8).
- Network comments can be specified when the variable name generation rules are set to **Customized setting**. Since the identifier *//NetworkComment* is set in the variable name column of the setting file that is output as a template, set any text string for the variable comment.

# 6-5-6 Function Block Conversion for Programs

You can convert a generated program to a function block.

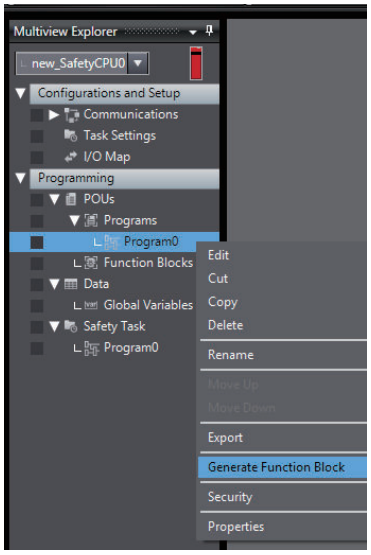
## Conversion Method

Use the following conversion procedure.

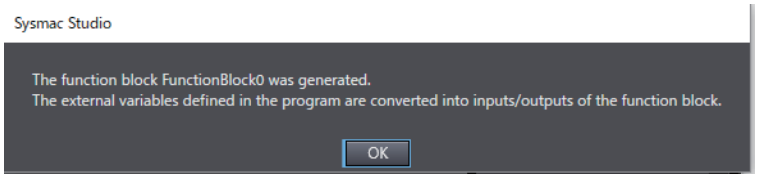
***1*** Select the Safety CPU Unit from the Controller Selection Box in the Multiview Explorer and open the Safety CPU Unit Setup and Programming View.

**2** In the Multiview Explorer, select **Programming** – **POUs** – **Programs**. Then, select a program you want to convert into a function block.

**3** Right-click the program and select **Generate Function Block** from the menu.




**4** The following message appears and the converted function block is added under **Function Blocks**, which is located below **Programming** – **POUs** in the Multiview Explorer.




External variables defined in the program are automatically converted as shown below. Edit each item as necessary.

| Item | Conversion method |
|---|---|
| Type of Variables | External variables are converted into input or output variables of the function block.<br>If a value is written to an external variable in the POU, it is converted into an output variable. Otherwise, it is converted into an input variable. |
| Variable Name | It is converted into *FB_original variable name*. |
| Data Type | The data type of the external variable is applied to the input variable or output variable. |
| Initial Value | The initial value of the global variable referenced by the external variable is applied to the input variable or output variable. |
| Comment | The comments for the external variable are applied to the input variable or output variable. If the comments for the external variable are blank, the ones for the global variable are applied. |

## 6-5-7 Building

Building is the process of converting the safety programs in your project into a format that is executable on the Safety CPU Unit.

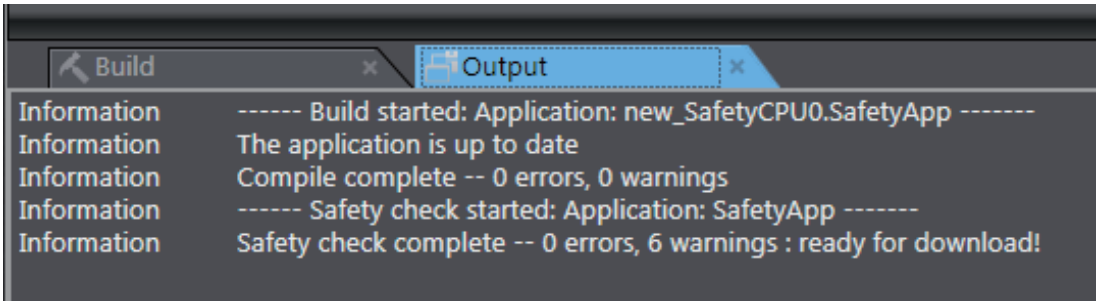
A check is performed on the programs and variables during this process. If there are any errors, the build is not performed and the errors are displayed in the Build Tab Page.

## Executing the Build Process

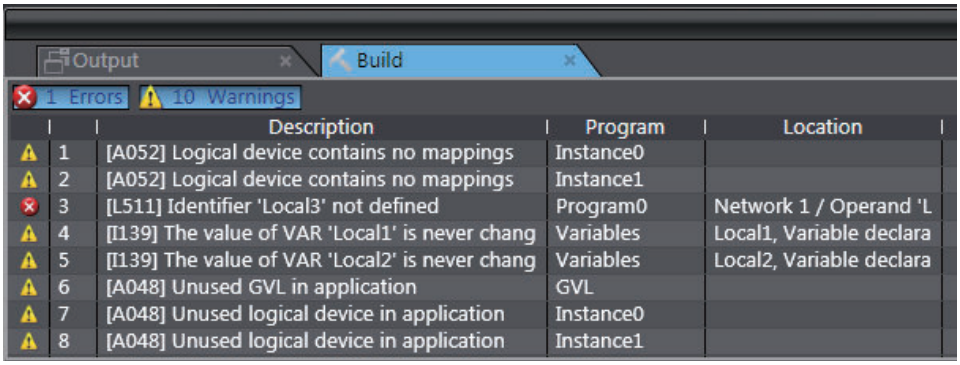
**1** Use one of the following procedures to execute the build process.

| Method | Operation |
|---|---|
| Method 1 | Select **Build Controller** from the **Project** Menu. |
| Method 2 | Press the **F8** Key. |
| Method 3 | Click the **Build Controller** Button on the toolbar. |

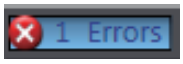
The build is started and the status during the build is displayed in the Output Tab Page.

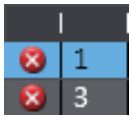
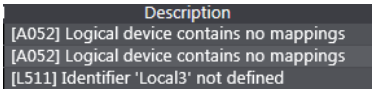



**2** Click the **Build** Tab.
The Build Tab Page is displayed.
If there are any errors, a list of them is displayed.




The following items are displayed in the Build Tab Page.
If there is an error, double-click a line in the list to display the location of the error, and then correct the error.

| Item | Example | Description |
|---|---|---|
| Number of errors |   | Displays the total number of errors. |

| Item | Example | Description |
|---|---|---|
| Number of warnings |  | Displays the total number of warnings. |
| Error or warning number |  | Displays the errors or warnings in the order in which they were found. |
| Description |  | Displays a description of the error or warning. |
| Location |  | Displays the location where the error or warning occurred. You can jump directly to the location of the error. |

**Additional Information**

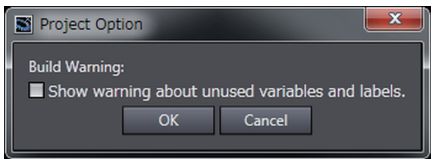
If the data size of the program exceeds the program capacity of the Safety CPU Unit, a Capacity Exceeded Error will occur when you change to DEBUG mode.
Reduce the number of FBs or variables that are used.

## Changing Build Options

You can change the warning setting to change the build warning display settings for unused variables and jump labels.

Use the following procedure to change warning levels.

***1*** Select **Project Options** from the **Project** Menu.
The Project Option Dialog Box is displayed.




***2*** Select the check box and click the **OK** Button.
The specified warnings are shown or hidden accordingly.

## 6-5-8 Searching and Replacing

You can search for and replace strings in the data of a project.

## Scope of Searching and Replacing

You can search for and replace text strings in the following items.

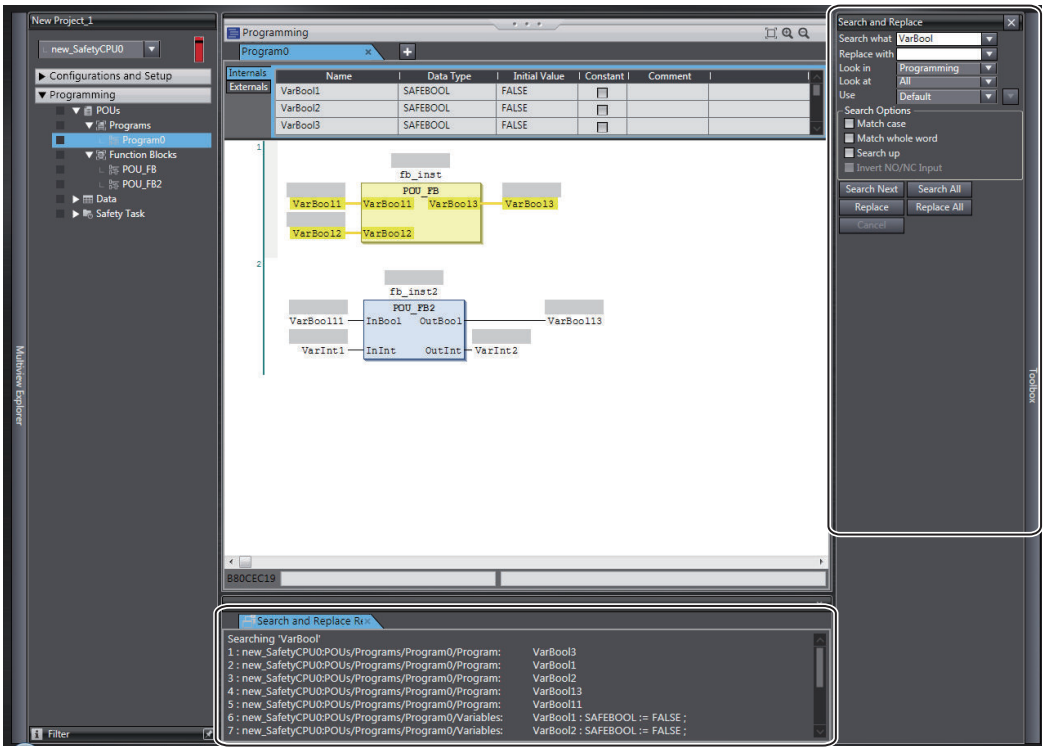
| Selected item | Scope of Searching and Replacing |
|---|---|
| All items (text strings) | Variable names, variable comments, FBD network comments, jump labels, and Jump instructions. |
| Variable | Variable names |
| Instruction | Function block instance names |

## Search and Replace Pane

**1** Use one of the following procedures to display the Search and Replace Pane in place of the Toolbox.

| Method | Operation |
|---|---|
| Method 1 | Select **Search and Replace** from the **Edit** Menu. |
| Method 2 | Press the **Ctrl** + **F** Keys. |
| Method 3 | Click the **Search and Replace** Button on the toolbar. |

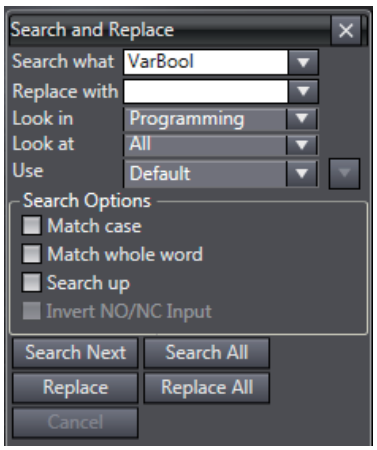
The Search and Replace Pane is displayed.




**2** In the Search and Replace Pane, enter the text string and set the applicable search conditions, and then click one of the buttons for search or replace.
The results of the search and replace process are displayed on the **Search and Replace Results** Tab Page.
Double-click the line in the displayed results to move the focus to the corresponding location.

## Setting Items

The setting items in the Search and Replace Pane are explained below.




| Setting Items | Description |
|---|---|
| Search what | • Enter a search string.<br>• You can select from previous search strings in the list. |
| Replace with | • Enter the string to replace the search string with.<br>• You can select from previous replacement strings in the list.<br>You cannot use wildcard characters. |
| Look in | Specify the range to search. You can select from the following.<br>Programming : The search is performed on the program to which the selected element belongs when the search is executed. If the search is made on the Safety CPU Unit Setup and Programming View, the search is performed only for the program in the Safety CPU Unit.<br>Current view : The current view is searched. |
| Look at | Specify the items to search. You can perform a search on the following items.<br>All : Variable tables (variable names in the table) and programs (jump labels, Jump instruction names, and variable names in the programs)<br>Variable Name : Searches all variable names.<br>Instruction : Function block instance names |
| Use | Specify if you want to use wildcard *1 characters.<br>Default : Do not use wildcard characters.<br>Wildcard : Use wildcard characters.<br>If you select to use wildcard characters, you can click the  Button to the right to view a list of characters used for wildcard characters. Select any of these characters to enter them in the **Search** string.<br>  |

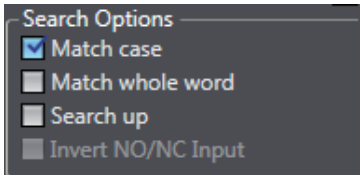


*1. The characters that you can use as wildcard characters are given on the next page.

### ● Wildcards

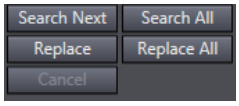
| Character | Meaning | Description | Example |
|---|---|---|---|
| * | Zero or more characters | Searches for a text string that contains a variable text string. | “new*” matches “newfile.txt”. |
| ? | Any single character | Searches for a text string with a variable character. | “A?C” matches “ABC”, “AdC”, and “AzC”. |
| # | Any single number | Searches for any single number. | “7#” matches “71”.<br>“ABC#” matches “ABC5”. |
| [ ] | Character in a set | Searches for a single character in the set. | “ABC[xyz]” matches “ABCx” and “ABCy”.<br>“ABC[x-z]” matches “ABCx” and “ABCy”. |
| [! ] | Character not in a set | Searches for a single character that is not in the set. | “ABC[!xyz]” matches “ABCa” and “ABCd”.<br>“ABC[!x-z]” matches “ABCa” and “ABCd”. |

### ● Search Options




| Item | Function |
|---|---|
| Match case | When this option is selected, searches are case sensitive. |
| Match whole word | When selected, only exact string matches are returned. |
| Search up | When selected, the search is performed backward from the cursor position. |

### ● Button Functions




| Item | Function |
|---|---|
| Search Next | Performs a search according to the selected options. |
| Search All | Searches all items and lists the results in the Output Tab Page. |
| Replace | Performs a replace according to the selected options. |
| Replace All | Replaces all items and lists the results in the Output Tab Page. |
| Cancel | Cancels the current search and replace operation. |

## 6-5-9 Safety Task Settings

This section describes the procedures that are used to select the programs to execute in the safety task and the execution order of the selected programs. It also describes how to set the task period of the safety task.

Refer to *Section 8 Calculating Safety Reaction Times* on page 8-1 for details on safety tasks.

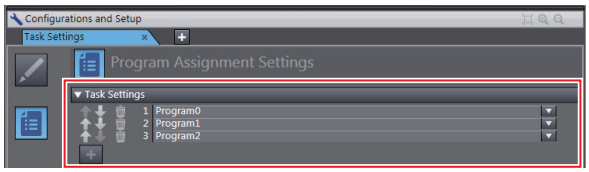
## Program Assignments

The program assignment settings are used to assign the programs to the safety task and set the program execution order.
The programs that are assigned to the task are executed in the specified order.

1 Double-click **Task Settings** under **Configurations and Setup** in the Multiview Explorer.

2 Click the **Program Assignment Settings** Button (  ) in Safety Task Settings Tab Page in the Edit Pane.

3 The buttons shown within the red frame below allow you to change the program assignments and their execution order.



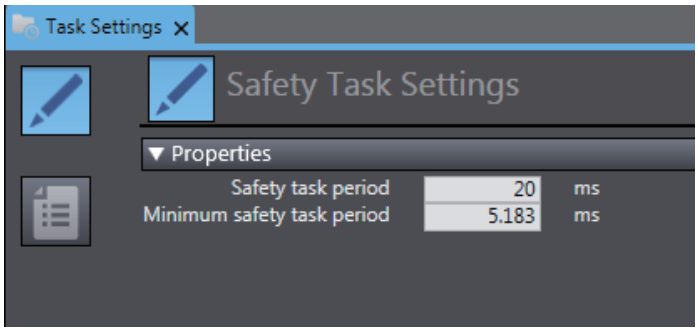

**Precautions for Correct Use**

Any program you assign must already be registered under **Programming** - **POUs**.

## Setting the Task Period

1 Double-click **Task Settings** under **Configurations and Setup** in the Multiview Explorer.
The Safety Task Settings Tab Page is displayed.

2 If the **Safety Task Settings** Display does not appear on the Safety Task Settings Tab Page in the Edit Pane, click the **Safety Task Settings** Button (  ).




The minimum safety task period is automatically calculated and displayed based on the program and settings information.

**3** Set the task period for the safety task.
Set the safety task period to a value that is within 100 ms of the minimum safety task period.

## 6-5-10 Variable Comment Switching Function

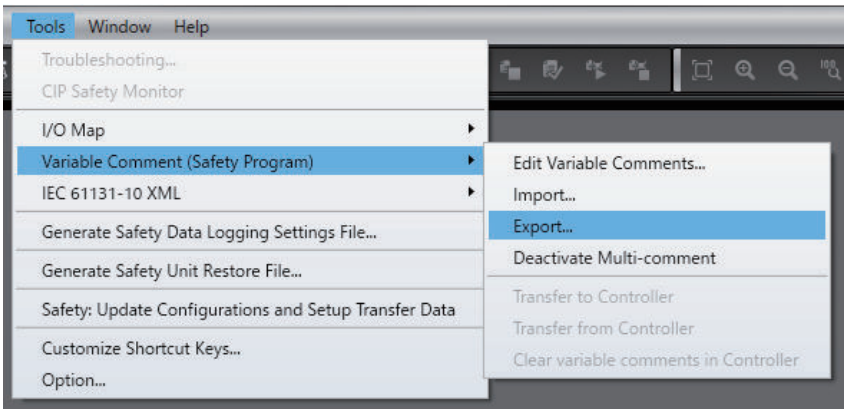
The variable comment switching function is designed to switch the variable comment displayed in the safety program or on the variable table to another comment.
To use this function, add any variable comment by using one of the following methods.

- **Importing a Comment File**
  Export the original variable comment file, add the variable comments used in the comment switching destination and import the comment file.
- **Editing with the Edit Variable Comments Function**
  Use the Edit Variable Comments function to edit displayed comments used for the variable comment switching function.

The operation procedure of this function is explained below.

### Exporting Variable Comments

**1** Select a Safety CPU Unit from which the variable comment is exported. On the **Tools** Menu, select **Variable Comment (Safety Program)** - **Export**.




The Save File Dialog Box is displayed.

**2** Enter a file name, and then click the **Save** Button.
The comments for variables are saved in the CSV format.

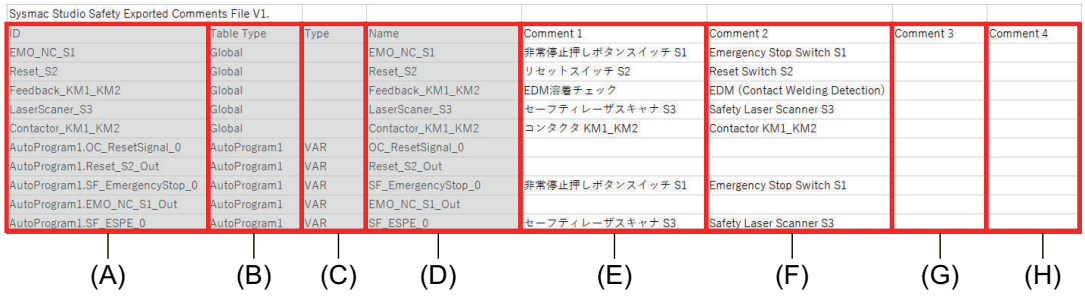
**Additional Information**

The variable comments of which displaying is disabled by the data protection feature will not be exported. Before you export such comments, temporarily disable the data protection that is being applied.

## Editing a Variable Comment File

Edit comments for variables in the exported CSV file.

**1** Open the CSV file on a spreadsheet application.



| Sysmac Studio Safety Exported Comments File V1. | | | | | | | |
|---|---|---|---|---|---|---|---|
| ID | Table Type | Type | Name | Comment 1 | Comment 2 | Comment 3 | Comment 4 |
| EMO_NC_S1 | Global | | EMO_NC_S1 | 非常停止押しボタンスイッチ S1 | Emergency Stop Switch S1 | | |
| Reset_S2 | Global | | Reset_S2 | リセットスイッチ S2 | Reset Switch S2 | | |
| Feedback_KM1_KM2 | Global | | Feedback_KM1_KM2 | EDM溶着チェック | EDM (Contact Welding Detection) | | |
| LaserScaner_S3 | Global | | LaserScaner_S3 | セーフティレーザスキャナ S3 | Safety Laser Scanner S3 | | |
| Contactor_KM1_KM2 | Global | | Contactor_KM1_KM2 | コンタクタ KM1_KM2 | Contactor KM1_KM2 | | |
| AutoProgram1.OC_ResetSignal_0 | AutoProgram1 | VAR | OC_ResetSignal_0 | | | | |
| AutoProgram1.Reset_S2_Out | AutoProgram1 | VAR | Reset_S2_Out | | | | |
| AutoProgram1.SF_EmergencyStop_0 | AutoProgram1 | VAR | SF_EmergencyStop_0 | 非常停止押しボタンスイッチ S1 | Emergency Stop Switch S1 | | |
| AutoProgram1.EMO_NC_S1_Out | AutoProgram1 | VAR | EMO_NC_S1_Out | | | | |
| AutoProgram1.SF_ESPE_0 | AutoProgram1 | VAR | SF_ESPE_0 | セーフティレーザスキャナ S3 | Safety Laser Scanner S3 | | |

(A) (B) (C) (D) (E) (F) (G) (H)

| No. | Name | Description | Remarks |
|---|---|---|---|
| (A) | ID | Unique ID automatically assigned to a variable. | Do not change this item. |
| (B) | Table Type | The type of the variable table is displayed.<br>• Global: Global variable<br>• POU name: Local variable | Do not change this item. |
| (C) | Type | The type of the local variable table is displayed.<br>• VAR: Internal variable<br>• VAR_EXTERNAL: External variable<br>• VAR_INPUT: Input variable<br>• VAR_OUTPUT: Output variable | Do not change this item. |
| (D) | Name | The variable name is displayed. | Do not change this item. |
| (E) | Comment 1 | This is a comment for the registered variable. | |
| (F) | Comment 2 | This is a comment for the variable registered at Comment 2. | |
| (G) | Comment 3 | This is a comment for the variable registered at Comment 3. | |
| (H) | Comment 4 | This is a comment for the variable registered at Comment 4. | |

**2** For **Comment 2**, **Comment 3** or **Comment 4**, enter a variable comment to be switched.

**3** Save the CSV file.

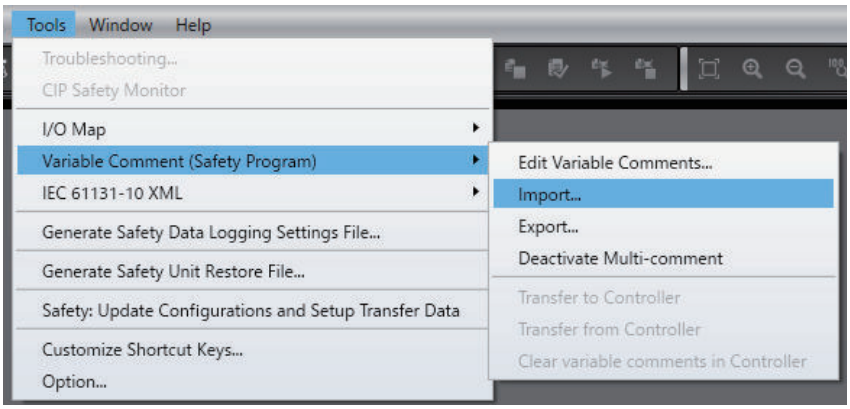
**Precautions for Correct Use**

- Do not change the contents of the variable comment file except for **Comment 1** through **Comment 4**.
- If you leave the comment field of **Comment 2** through **Comment 4** blank, the target variable comment becomes empty.

## Importing a Variable Comment File

Import the edited CSV file to the original project.

**1** Select a Safety CPU Unit to which the variable comment is imported. On the **Tools** Menu, select **Variable Comment (Safety Program)** - **Import**.




The File Selection Dialog Box is displayed.

**2** Select an CSV file to import, and then click the **Open** Button.
The Confirmation Dialog Box for the import is displayed.
When you click the **OK** Button, the import function executes and replaces the comments in the project with the comments in the CSV file.
If an error occurs during the import processing, the error details are displayed in the Output Tab Page.

**Precautions for Correct Use**

- When you import a variable comment file, the comments in the target Safety CPU Unit inside the project are replaced with the comments in the variable comment file. Please remember that the original comments cannot be restored.
- After importing a variable comment file for project data, the project data cannot be imported using Sysmac Studio Ver.1.28 and earlier.
- If the content of **Comment 1** was modified by the import function of the variable comment file, the safety validation status becomes unvalidated. Changing the contents of **Comment 2** through **Comment 4** will not change the validated status.
- If **Comment 1** in the variable comment file is empty, the contents of **Comment 2** through **Comment 4** will not be imported.
- If there is no corresponding variable in the import destination of the variable comment file, the comment for the variable is excluded from the import target.
- If there are duplicated entries in **Comment 1**, the first found entry in **Comment 2** through **Comment 4** in CSV will be imported.

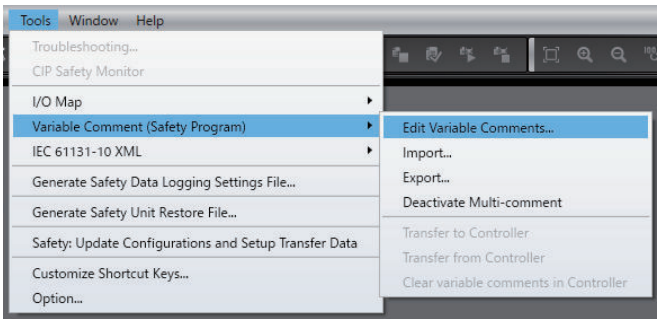
**Additional Information**

The variable comments of POUs of which editing is disabled by the data protection feature will not be imported. Before you import such comments, temporarily disable the data protection that is being applied.

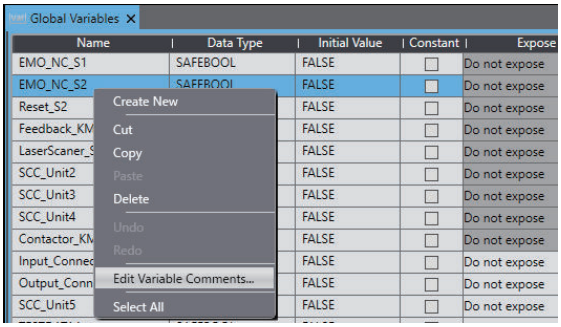
## Edit Variable Comments Function

This function edits variable comments used for the variable comment switching function.
Use the following procedure to display the Edit Variable Comments Window, and add or edit any variable comment.

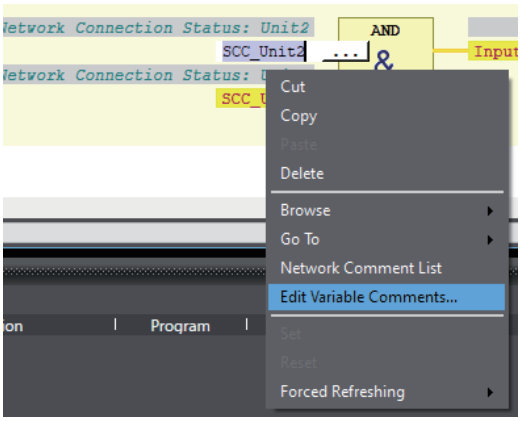
- **Method 1: How to start the Edit Variable Comments Window from the Tools menu**
  Select **Variable Comment (Safety Program)** - **Edit Variable Comments** from the **Tools** menu.




- **Method 2: How to start the Edit Variable Comments Window from the variable table**
  Select any variable in the variable table, and right-click the variable and select **Edit Variable Comments**.

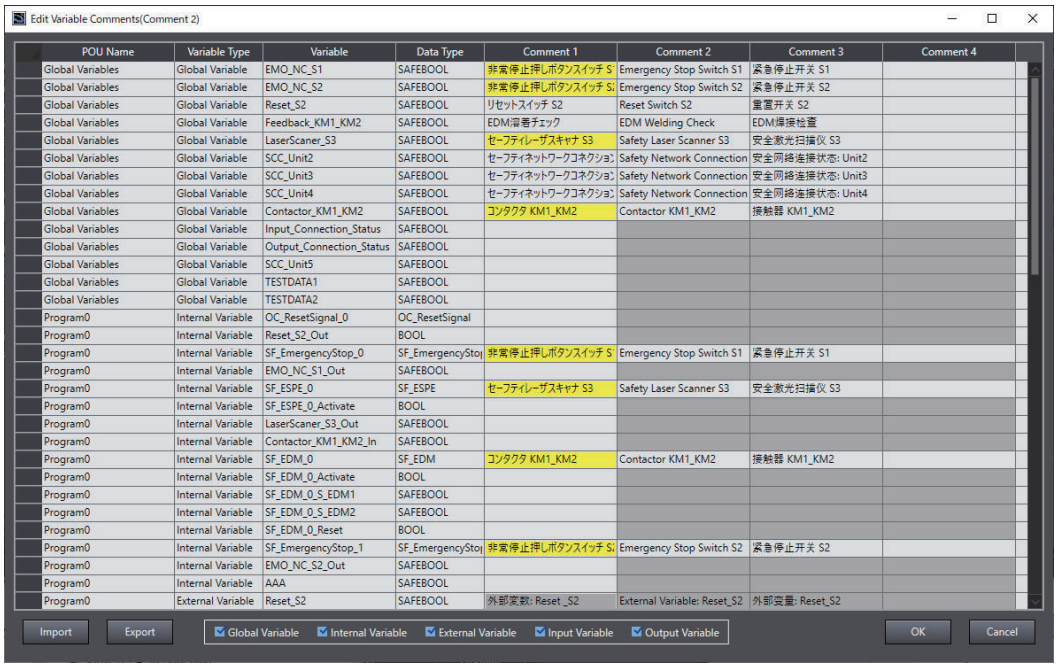



- **Method 3: How to start the Edit Variable Comments Window from the FBD editor**
  Select any variable in the FBD editor, and right-click the variable and select **Edit Variable Comments**.




When the Edit Variable Comments Window is started by method 2 or 3, it is displayed with the variable selected.

- **The Edit Variable Comments Window**
  When the Edit Variable Comments function is executed, the following screen is displayed.




The display contents of the Edit Variable Comments Window are as follows.

| Item | Description | Remarks |
|---|---|---|
| POU Name | This is the name of POUs (programs and function blocks) in which variables are defined. The variables defined in the global variable table are displayed as *Global Variables*. | (Cannot be edited.) |
| Variable Type | Displays the type of variable table in which the variable is defined. | (Cannot be edited.) |
| Variable | Displays the variable name. | (Cannot be edited.) |
| Data type | Displays the data type of the variable. | (Cannot be edited.) |
| Comment 1 | Displays the comment registered at the variable. | |
| Comment 2 | Displays the comment for the variable registered at Comment 2. | |
| Comment 3 | Displays the comment for the variable registered at Comment 3. | |
| Comment 4 | Displays the comment for the variable registered at Comment 4. | |
| The **Import** button | Import the variable comments to the Edit Variable Comments Window. Refer to *Editing a Variable Comment File* on page 6-78 for the format of a variable comment file. | |
| The **Export** button | Export the variable comments from the Edit Variable Comments Window. Refer to *Editing a Variable Comment File* on page 6-78 for the format of a variable comment file. | |
| Variable Type Filtering check box | Displays only variable types for which the check boxes are selected. | |

**Note** If the content of **Comment 1** is duplicated with other variable comments, the background of it is yellow. The variable comment switching function reflects the contents of **Comment 2** to **Comment 4** of the variable first found in the Edit Variable Comments Window, if the contents of **Comment 1** are duplicated.

Edit the contents of **Comment 1** to **Comment 4** of any variable. When you click the **OK** button, the contents of **Comment 1** to **Comment 4** are reflected in the project.

**Precautions for Correct Use**

- Project data, for which **Comment 2** to **Comment 4** are set by the Edit Variable Comments function, cannot be imported by using the Sysmac Studio Ver.1.28 and earlier.
- If the content of **Comment 1** was modified by the Edit Variable Comments function, the safety validation status becomes unvalidated. Changing the contents of **Comment 2** through **Comment 4** will not change the validated status.
- If **Comment 1**is empty, the comments from **Comment 2** to **Comment 4** will not be set.

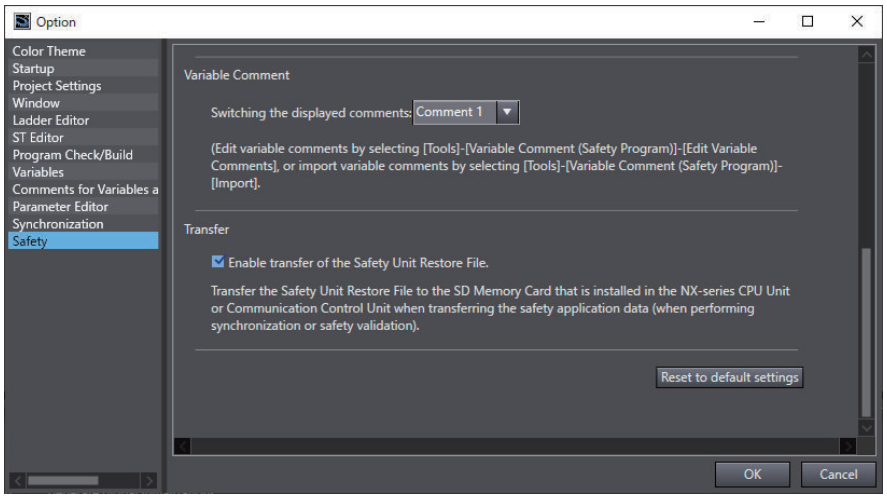
**Additional Information**

Variable comments of POUs of which editing is disabled by the data protection feature cannot be edited. Before you edit such comments, temporarily disable the data protection that is being applied.

## Switching the Displayed Comments

The procedure below switches display to the comment in the imported variable comment file.

1. On the **Tools** Menu, select **Option** and open the Option Settings Dialog Box.
2. Select the **Safety** Tab. Go to **Variable Comment - Switching the displayed comments**. From the dropdown list, select a comment to show and click the **OK** Button.




The switched comment appears when you display the variable table or open the FBD editor.

**Precautions for Correct Use**

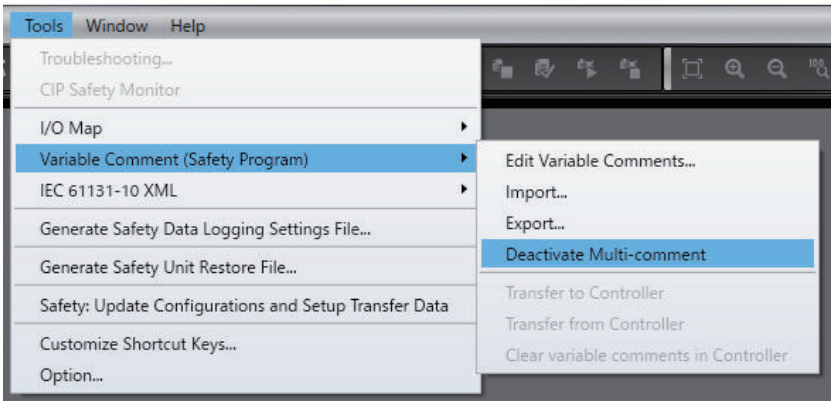
If the display comment is set to other than **Comment 1**, the variable comment cannot be edited except by using the Edit Variable Comments Function.

## Deactivating Multi-comment

This section describes the procedure to delete variable comments entered in **Comment 2** through **Comment 4** that are used for the switching variable comment function.

The project containing the imported variable comment file cannot be opened on Sysmac Studio Ver.1.28 and earlier because the switching variable comment function is not supported. When you delete variable comments in **Comment 2** through **Comment 4** in the following procedure, the project can be opened on Sysmac Studio Ver.1.28 and earlier.

***1*** On the **Tools** Menu, select **Variable Comment (Safety Program)** - **Deactivate Multi-comment**.




If **Deactivate Multi-comment** is executed, the content of **Comment 1** is always displayed as the variable comment regardless of the option settings configured in **Safety** - **Variable Comment** - **Switching the displayed comments**.

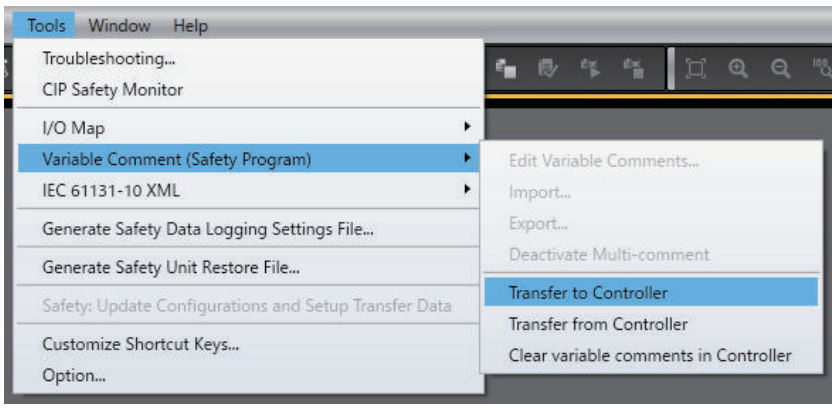
## Transferring Variable Comments

The settings for **Comment 2** and the subsequent comments can be downloaded to the controller by using the function introduced below.
If you downloaded the settings for **Comment 2** and the subsequent comments to the controller, the settings for **Comment 2** and the subsequent comments can be restored by the upload.

### ● Downloading Settings for Comment 2 and the Subsequent Comments to the Controller

Use the following procedure to download the settings for **Comment 2** and the subsequent comments.

- Establish an online connection with the controller. Go to the **Tools** Menu and select **Variable Comment (Safety Program)** - **Transfer to Controller**.



**Precautions for Correct Use**

Since the settings for **Comment 2** and the subsequent comments are not included in the safety application data, they are not transferred to the Safety CPU Unit.

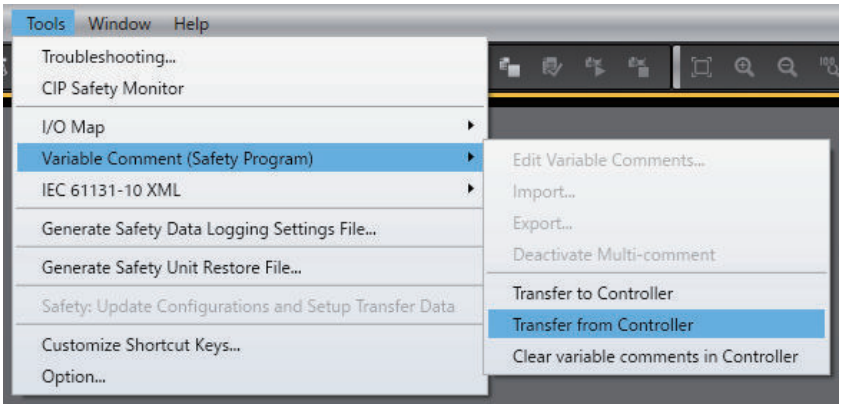
**Additional Information**

The settings for **Comment 2** and the subsequent comments are downloaded when the safety application data is transferred, such as during the execution of validation check or transfer by synchronization.

## ● Uploading Settings for Comment 2 and the Subsequent Comments from the Controller

Use the following procedure to upload the settings for **Comment 2** and the subsequent comments.

- Establish an online connection with the controller. Go to the **Tools** Menu and select **Variable Comment (Safety Program)** - **Transfer from Controller**.



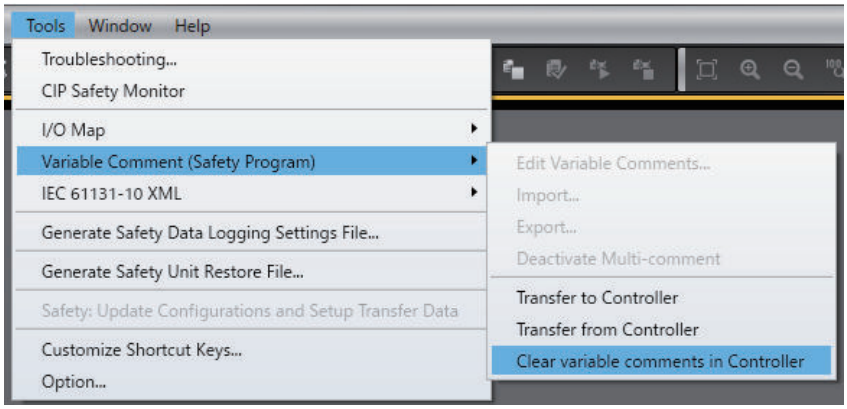

**Additional Information**

The settings for **Comment 2** and the subsequent comments are uploaded when the transfer of safety application data is executed by synchronization.

## ● Clearing Settings for Comment 2 and the Subsequent Comments from the Controller

Use the following procedure to clear the settings for **Comment 2** and the subsequent comments.

- Execute the Clear All Memory operation for the controller.
- Establish an online connection with the controller. Go to the **Tools** Menu and select **Variable Comment (Safety Program)** - **Clear variable comments in Controller**.




### Precautions for Correct Use

If you use Sysmac Studio Ver.1.28 and earlier to upload safety application data from the controller and the Safety CPU Unit containing a downloaded variable comment file for the Safety CPU Unit, the data for **Comment 2** and the subsequent comments are not restored because Ver.1.28 and earlier does not support the variable comment switching function.

### Additional Information

- If you use the SD Memory Card backup function of the controller, the settings for **Comment 2** and the subsequent comments are included in the backup or restored target.
- If you use the controller backup functions of Sysmac Studio, the settings for **Comment 2** and the subsequent comments are included in the backup or restored target.
- If you use the importing/exporting backup files function of Sysmac Studio, the settings for **Comment 2** and the subsequent comments are not included in the import or export target.

# 6-6 Automatic Programming

This section describes "automatic programming", which generates safety programs based on required specifications for the safety system using the Sysmac Studio.

**⚠ WARNING**

Programs generated by the automatic programming do not guarantee functional safety. Before you perform validation test of the safety programs, complete debugging of the safety programs.
Otherwise, the Safety CPU Unit will start with safety programs that are not fully debugged, and may cause serious personal injury.

## 6-6-1 Generation Algorithms for Automatic Programming

When a program is generated by executing the automatic programming function, the Sysmac Studio generates a safety program in the following steps:

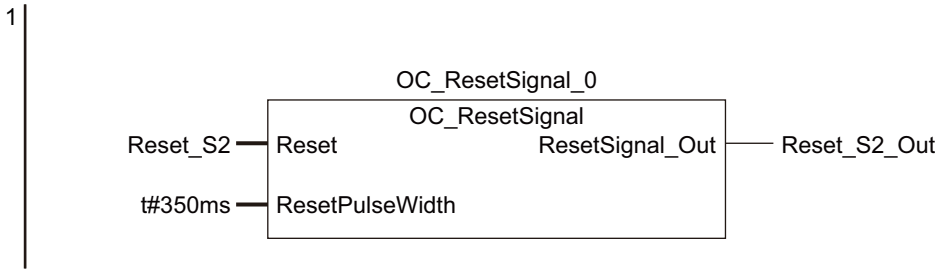
- Reset Signal Generation
- Input Signal Generation
- Generation of Logical Operation Circuit
- Output Signal Generation

### ● Reset Signal Generation

The automatic programming function creates a function block that generates a reset signal on the first network.

When a variable set in the **Basic Settings** on the Automatic Programming Settings Tab Page changes to FALSE after staying TRUE for a certain period of time, the reset signal detects the change of the variable and is set to TRUE for only one cycle. The default pulse width is set to 350 ms.

The generated reset signal here is used as a reset signal of the safety function block when an input signal is generated.



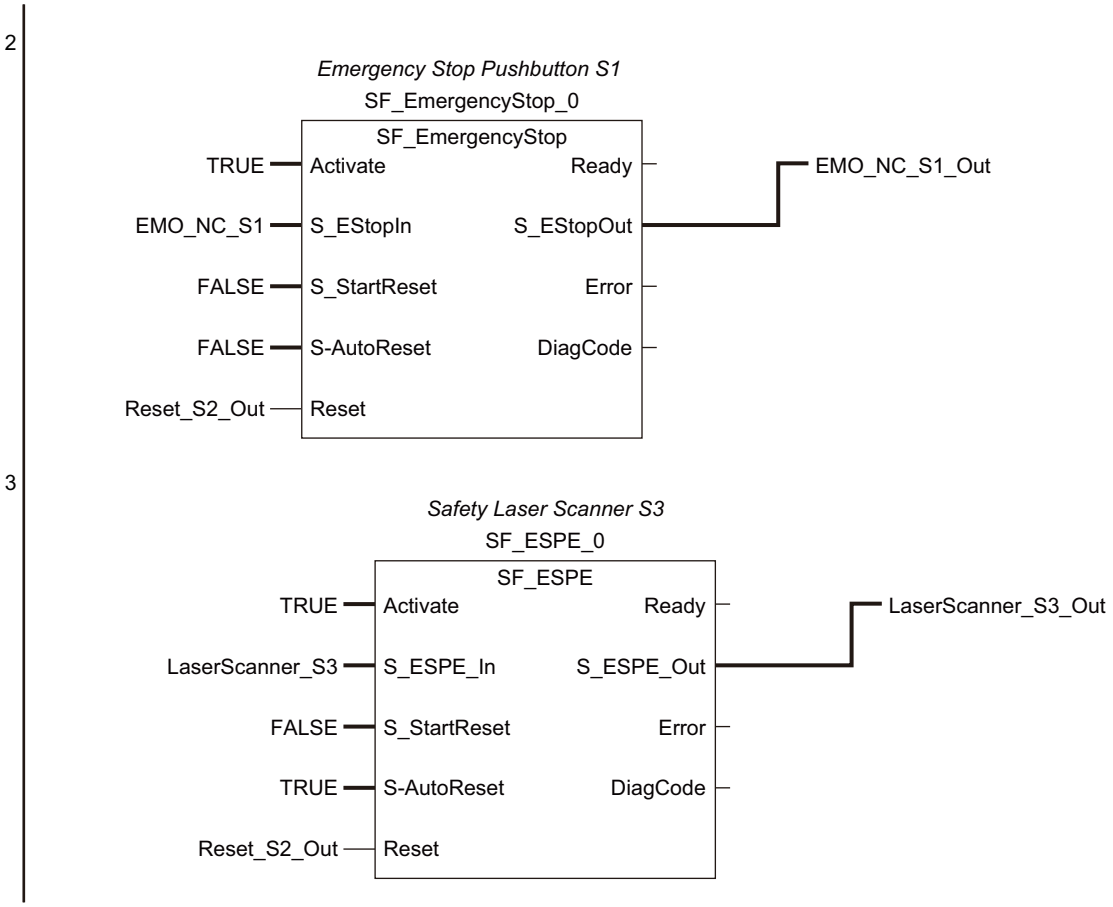

### ● Input Signal Generation

This step creates the function block that is defined in **Input Settings** on the Automatic Programming Settings Tab Page.

The variable name specified in **Input Settings** in the Automatic Programming Settings Tab Page is used for input variables for the function block. In addition, the output variable of the function block is automatically named as an internal variable, and is used as an input signal when a logical operation circuit is generated.

The following table shows values set to input variables of the function block.

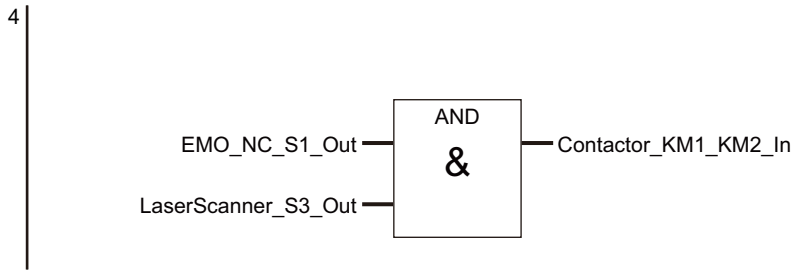
| Input variable name | Setting value |
|---|---|
| Activate | TRUE |
| S_StartReset | FALSE |
| S_AutoReset | Value specified for the reset type<br>Manual: FALSE<br>Auto: TRUE |
| Reset | Variable generated in the reset signal generation step |
| MonitoringTime | t#300ms |
| Any other input variables | Variable automatically generated based on the following naming rule:<br>**FB Instance Name_Input Variable Name** |




## ● Generation of Logical Operation Circuit

At this step, an AND logic function is generated for each output variable based on the **Expected Value Settings** in the Automatic Programming Settings Tab Page.

The internal variables, which are automatically generated at the input signal generation, are used as input variables for the AND function. In addition, the output variable of the AND function is automatically generated as an internal variable and is used as an input signal at the following output signal generation.

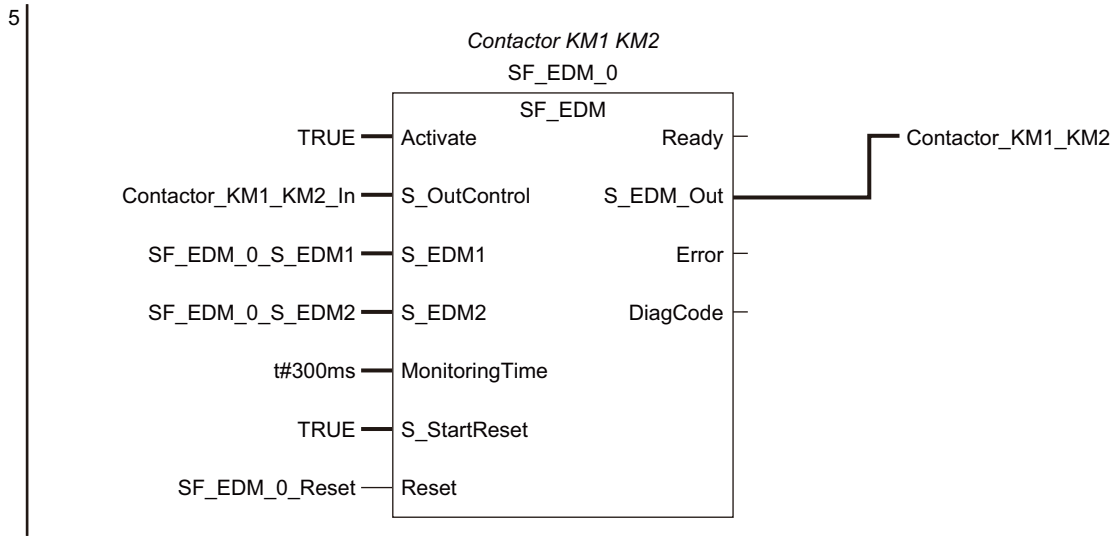



## Output Signal Generation

At this step, an SF_EDM function block is generated based on the Use EDM settings defined for **Output Settings** on the Automatic Programming Settings Tab Page.
The internal variable, which is automatically generated at the generation of logical operation circuit, is used as an input signal for the SF_EDM function block.

In addition, the variable name specified in **Output Settings** in the Automatic Programming Settings Tab Page is used for an output variable of the SF_EDM function block. The output variable, which is generated at the generation of logical operation circuit, is applied if the SF_EDM function block is not used. The following table shows values set to the input variables for the SF_EDM function block.

| Input variable name | Setting value |
|---|---|
| S_StartReset | TRUE |
| MonitoringTime | t#300ms |
| Any other input variables | Variable automatically generated based on the following naming rule:<br>**FB Instance Name_Input Variable Name** |




### Precautions for Correct Use

- The function block **OC_ResetSignal**, which is added by the automatic programming function, is protected with the display prohibition setting. You can display the Help menu for **OC_ResetSignal** by selecting **OC_ResetSignal** in the program or toolbox and pressing the **F1** Key.
- Variables and constants defined for the instances of each function block are necessary information for running the program. You can edit and fine-tune them in accordance with the equipment configuration and applications.

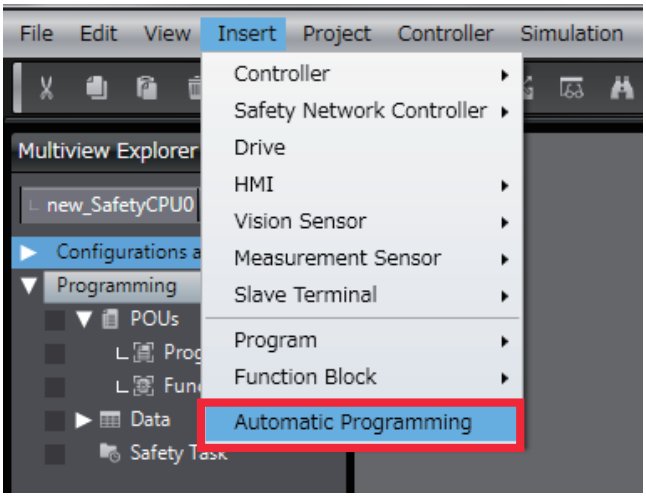
## 6-6-2 Automatic Programming Settings

To execute the automatic programming, you need to specify the following items on the Automatic Programming Settings Tab Page.

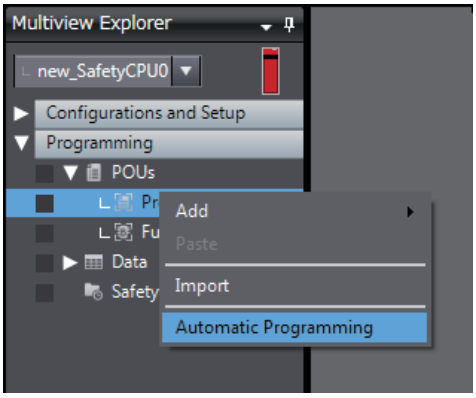
- Reset signal
- Input signal
- Output signal
- Expected value of the output signal corresponding to the input signal

You can open the Automatic Programming Settings Tab Page in either of the following procedures:

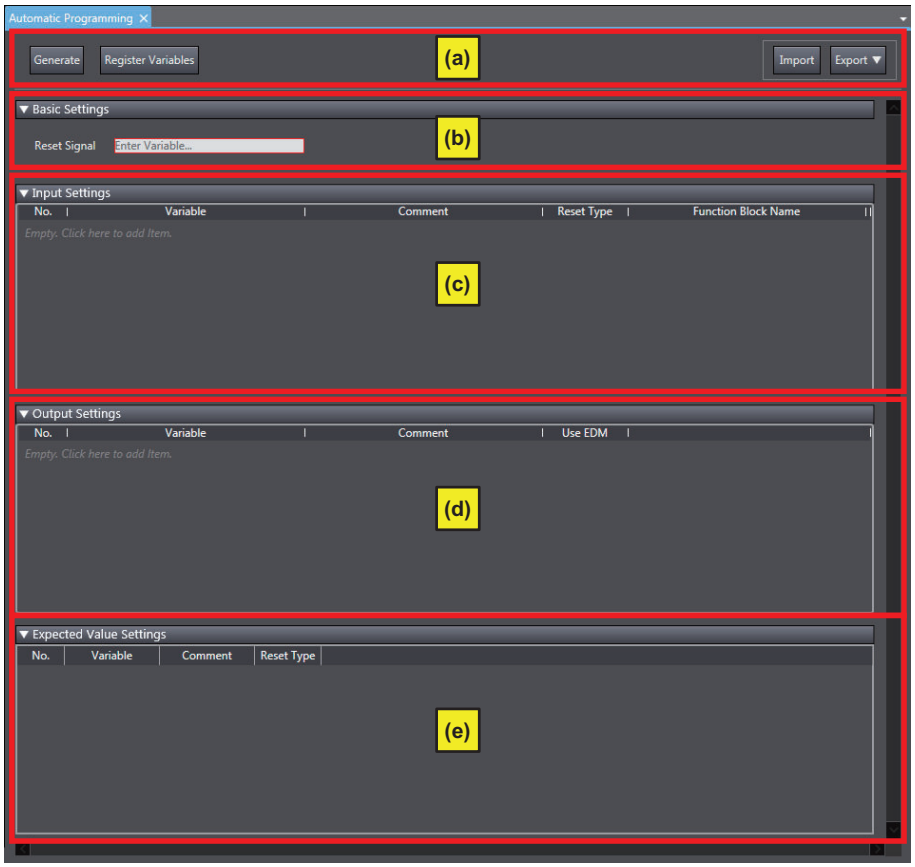
- Select **Automatic Programming** from the **Insert** Menu.




- In the Multiview Explorer, select **Programming** – **POUs** – **Programs**. On the right-click menu, select **Automatic Programming**.




The Automatic Programming Settings Tab Page consists of the items shown in the following table.




| Letter | Name | Function | |
|---|---|---|---|
| (a) | Operation Panel | The operation panel is used to make settings for and execute the automatic programming. | |
| | | Generate Button | Generates a program using the automatic programming. |
| | | Register Variables Button | Registers the variables that are used for the automatic programming. |
| | | Import Button | Imports the settings for the automatic programming from a CSV file.<br>If you import the settings, the current settings are overwritten. |
| | | Export – CSV File Output | Exports the current automatic programming settings to a CSV file. |
| | | Export – Export to Simple Automatic Test | Exports the current automatic programming settings to the simple automatic test settings.<br>The settings of the export destination are overwritten. |
| | | Export – Export to Online Functional Test | Exports the current automatic programming settings to the online functional test settings.<br>The settings of the export destination are overwritten. |
| (b) | Basic Settings | Set the variable to use as a reset signal for the safety program. You can set any SAFEBOOL or BOOL variable that is defined in the global variable table.<br>When a program is generated, the variable name for the configured reset signal is assigned to the input variable *Reset* of the safety function block that is linked to each input signal. | |

| Letter | Name | Function | |
|---|---|---|---|
| (c) | Input Settings | Specify a variable for the input signal used for automatic programming.<br>In addition to the variable name, you need to specify a reset type and a linked safety function block for the input signal. | |
| | | Variable | Specify a variable name used as input signal for the program. You can specify a SAFEBOOL or BOOL-type variable for the variable name. |
| | | Reset Type | Select a reset type in the box. The default value of the reset time is a manual reset.<br>The selected reset type is assigned to the input variable *S_AutoReset* of the safety function block that is linked to each input signal.<br>When a program is generated, *S_AutoReset* is set t to FALSE for a manual reset, and set to TRUE for an automatic reset. |
| | | Function Block Name | Specify the name of a safety function block that is linked to each input signal.<br>You can also specify a user-defined function block. If you specify a user-defined function block, you need to define at least one SAFEBOOL or BOOL-type variable as the input and output variables. |
| (d) | Output Settings | Specify a variable for output signal used for automatic programming.<br>Specify a variable name, and enable or disable EDM for the output signal. | |
| | | Variable | Specify a variable name used as the output signal for the program. You can specify a SAFEBOOL or BOOL-type variable for the variable name. |
| | | Use EDM | If it is TRUE, the SF_EDM function block is used when a program is generated. |
| (e) | Expected Value Settings | A matrix of the variables that are specified in the input settings and the output settings is displayed. Set the expected value of the output variable for each input variable.<br>The expected values set for the output variables have the following meanings:<br>• 0<br>If the input variable changes to FALSE, the output variable changes to FALSE.<br>• 1<br>If the input variable changes to FALSE, the output variable changes to TRUE.<br>You can edit the expected value settings by importing or exporting the values, and copying and pasting the values with a spreadsheet program or any other application software. | |

## Setting Example

This section provides an example of automatic programming settings for the application example given in *A-3-3 Safety Laser Scanners* on page A-36.

Application Overview of Safety Laser Scanner

| Safety category/PL | Safety device | Stop category | Reset |
|---|---|---|---|
| Equivalent to 3/PLd | Safety Laser Scanner | 0 | Auto |
| | Emergency Stop Pushbutton Switch | 0 | Manual |

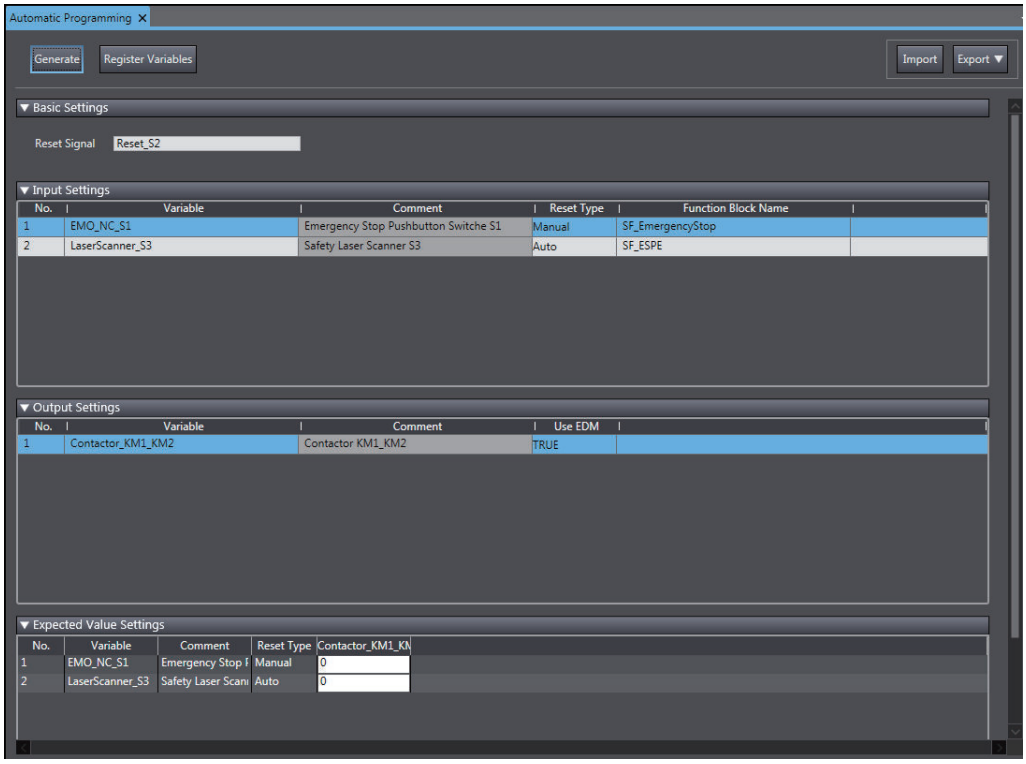
AGV stops when the emergency stop pushbutton S1 is pressed.

AGV stops also when the safety laser scanner detects someone or an object approaching to the safety zone.

At that time, enter the following variables of the safety laser scanner on the Automatic Programming Setting Tab Page.

- Variable that is assigned to the reset switch
- Variables that are assigned to the safety laser scanner and the Emergency stop pushbutton
- Variables that are assigned to contactors

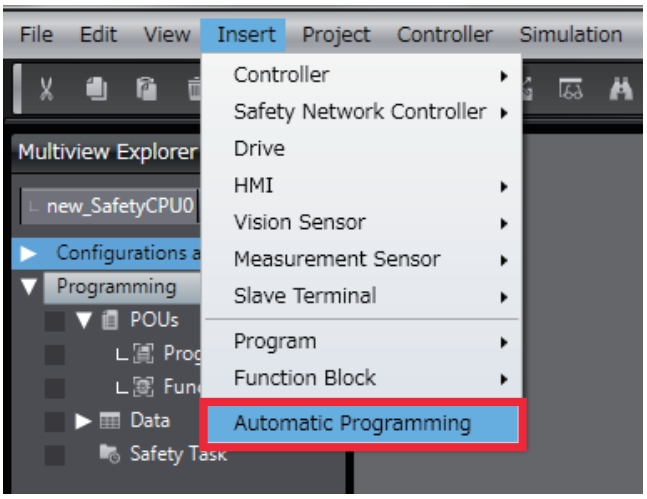
The settings for the above application example are configured in the Automatic Programming Settings Tab Page as shown below.




## 6-6-3 Automatic Programming Execution Procedure

The execution procedure of automatic programming is explained below.

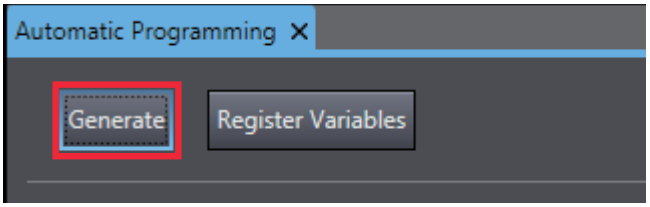
***1*** Select **Insert** - **Automatic Programming**.



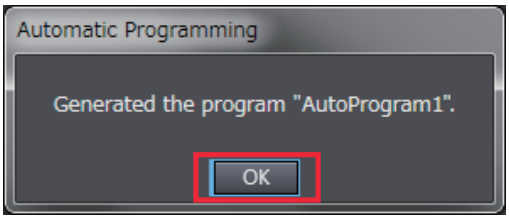

The Automatic Programming Settings Tab Page is displayed.

**2** Set **Basic Settings**, **Input Settings**, **Output Settings**, and **Expected Value Settings**.

**3** On the operation panel of the Automatic Programming Settings Tab Page, click the **Generate** Button.

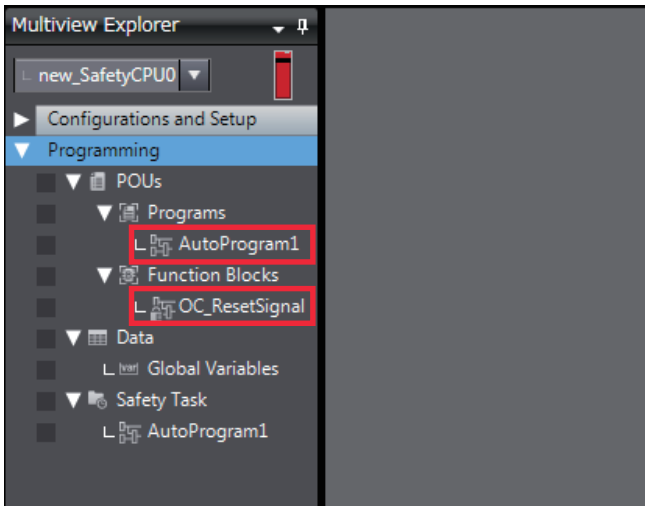



Once a program is generated, the following dialog is displayed.



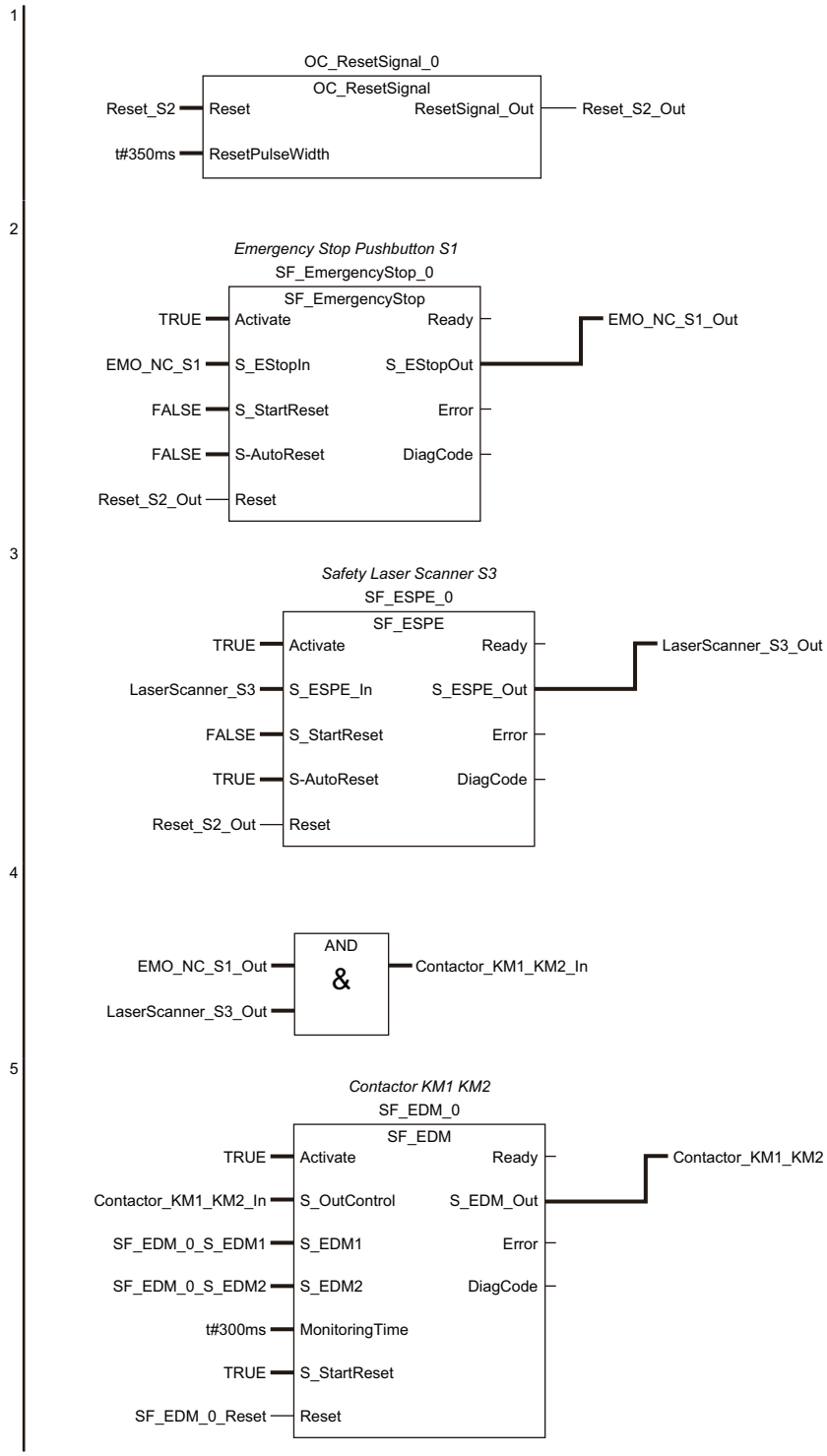

**4** Click the **OK** Button.

Once a program is generated, a safety program generated based on the settings and a function block (OC_ResetSignal) used to identify the reset signal are added to the project.




## Program Generated through Automatic Programming

Based on the example settings provided in *6-6-2 Automatic Programming Settings* on page 6-89, the following program will be generated.




### Additional Information

You can also manually add the function block (OC_ResetSignal) used to identify the reset signal in the following procedure.

- Right-click **Function Blocks** under **Programming** – **POUs** in the Multiview Explorer and select **Import OC ResetSignal** from the menu.

# 6-7 Monitoring Memory Usage for Safety Control Unit

You can display the memory usage of the safety control system and the safety network usage including the I/O data size.

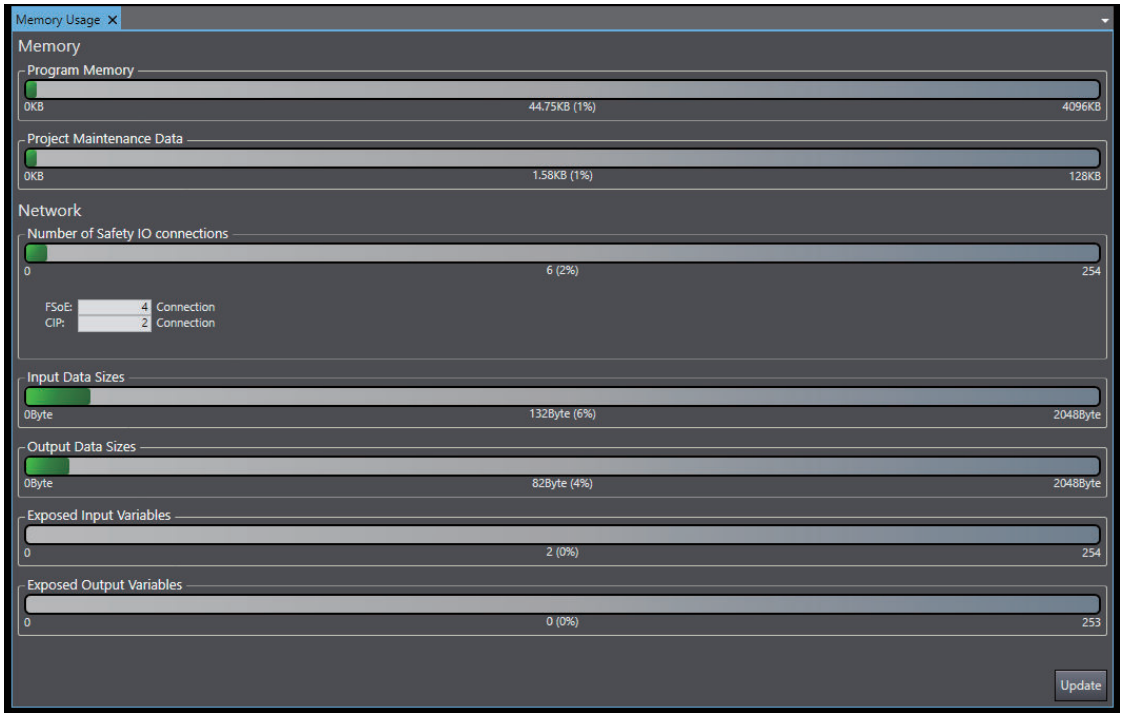
**Precautions for Correct Use**

Check the memory usage when there are no building errors. If there is a building error, the memory usage is not displayed correctly.

## Displaying Memory Usage

Use the following procedure to display memory usage.

Select the Safety CPU Unit from the Controller Selection Box in the Multiview Explorer and open the **Safety CPU Unit Setup and Programming** View.
Select **Memory Usage** from the **Project** Menu.
The Memory Usage Tab Page is displayed.



| Item | Display content | Related parameters |
|---|---|---|
| Program Memory | Displays the memory usage of the safety program. | • Number of the connected Safety I/O Units<br>• Number of CIP Safety connections, number of target I/O Assemblies<br>• Number of functions blocks and functions in the safety program<br>• Number of exposed variables, global variables and device variables<br>• Comment sizes for exposed variables, global variables, and device variables<br>• Function block names<br>• Number of device settings in the Safety Slave Unit parameter settings<br>• Terminal comments in the parameters for Safety Slave Units |
| Project Maintenance Data | Displays the memory usage of the project. | • Program names<br>• Number of exposed variables<br>• Comment sizes for exposed variables, global variables, and device variables<br>• Number of device settings in the Safety Slave Unit parameter settings<br>• Terminal comments in the parameters for Safety Slave Units |
| Number of Safety I/O connections | Displays the number of safety slaves connected to the Safety CPU Unit.<br>**FSoE** shows the number of connections with Safety I/O Units. **CIP** shows the number of CIP Safety connections. | • Number of the connected Safety I/O Units<br>• Number of CIP Safety connections, number of target I/O Assemblies |
| Input Data Sizes | Displays the input data usage by the Safety CPU Unit. | • Number of the connected Safety I/O Units<br>• Number of CIP Safety connections<br>• Data size of the exposed variables used for input settings |
| Output Data Sizes | Displays the output data usage by the Safety CPU Unit. | • Number of the connected Safety I/O Units<br>• Number of CIP Safety connections<br>• Data size of the exposed variables used for output settings |
| Exposed Input Variables | Displays the number of exposed variables in the input settings. | Number of exposed variables used for input settings |
| Exposed Output Variables | Displays the number of exposed variables in the output settings. | Number of exposed variables used for output settings |

# 6-8 Offline Debugging

Offline debugging allows you to debug a program when you are not connected online to a Safety CPU Unit.
You can debug on a Simulator to check control program logic before transferring the project to the Safety CPU Unit.

**⚠ WARNING**

Although the Simulator simulates the operation of the Safety CPU Unit, there are difference from the Safety CPU Unit in operation and timing. After you debug the safety program on the Simulator, always check operation on the physical Safety CPU Unit before you use the user program to operate the controlled system.
Accidents may occur if the controlled system performs unexpected operation.



**Precautions for Correct Use**

Simulation of the operation of the Safety CPU Unit cannot be linked with the operation of the NJ/NX-series CPU Unit.



**Additional Information**

Refer to *A-16 Differences in Checking Operation between the Simulator and Safety CPU Unit* on page A-106 for the differences between the Safety CPU Unit and the Simulator.

## 6-8-1 Offline Safety Program Debugging

To debug a safety program, it is best to simulate the safety program on the computer first to check the operation logic and parameter settings.
This is called offline debugging.

### Simulation Procedures

Use the following procedure to start the Sysmac Studio and connect to the Simulator.

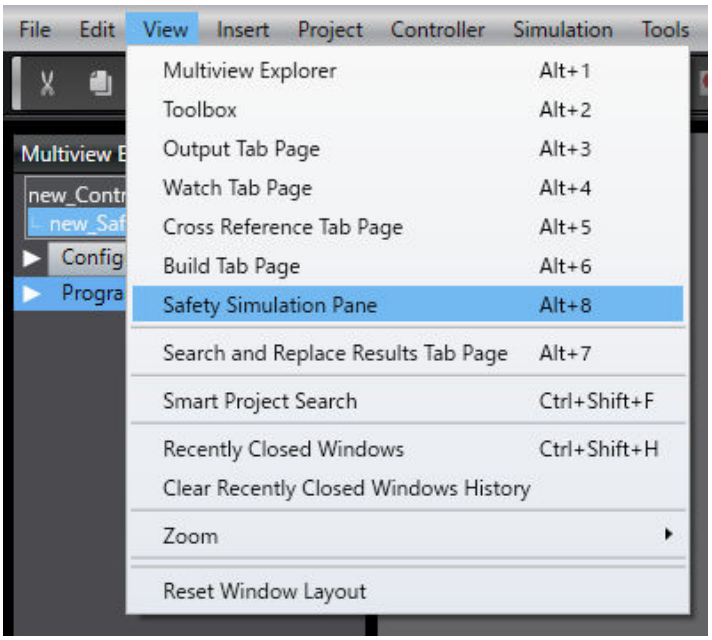
1. Start the Sysmac Studio and create a project.
2. In the Multiview Explorer, select the target Safety CPU Unit in the Controller Selection Box.
3. Use the Sysmac Studio to set the Controller Configurations and Setup and create a safety program.
4. Select **Build Controller** from the **Project** Menu to build the program.



**Additional Information**

You cannot connect to the Simulator if the program is not built.

5. Select **Safety Simulation Pane** from the **View** Menu.



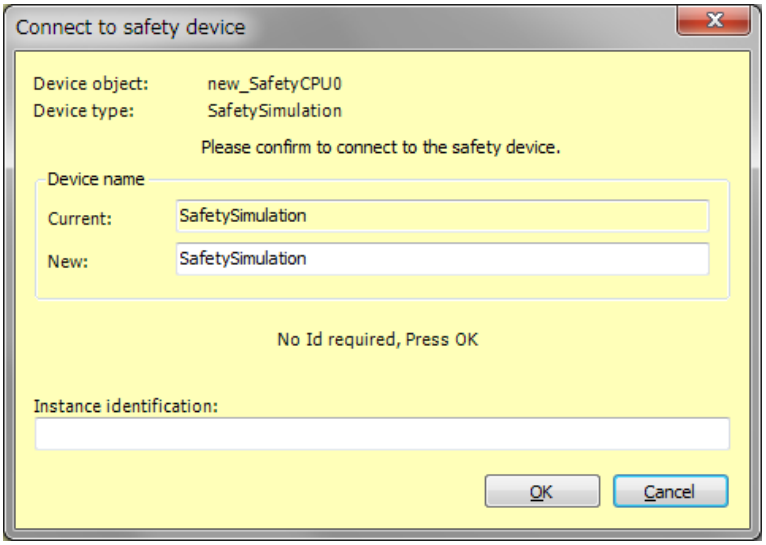

The Safety Simulation Pane is displayed below the Toolbox Pane on the right of the window.




**6** Click the **Run** Button in the Safety Simulation Pane. Or, select **Run** from the **Simulation** Menu.

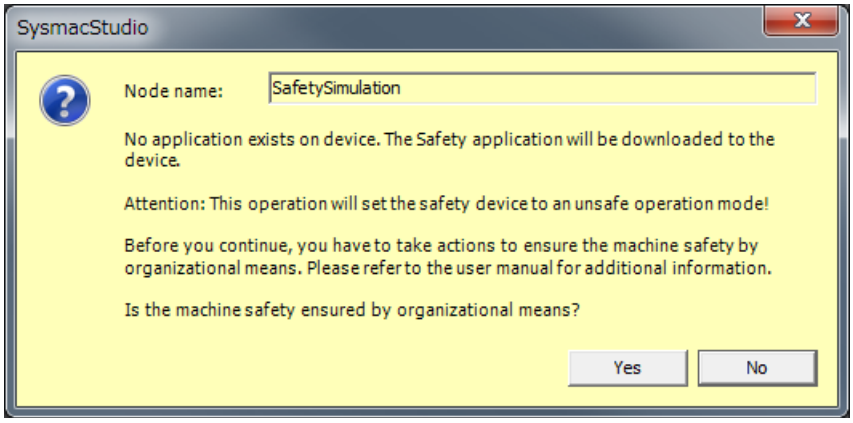



The following Connect to safety device Dialog Box is displayed.




**Note 1.** It is not necessary to change the name from the current node name.
**Note 2.** You do not need to enter anything in the **Instance identification** Box.

**7** Click the **OK** Button.
The following transfer confirmation dialog box is displayed.

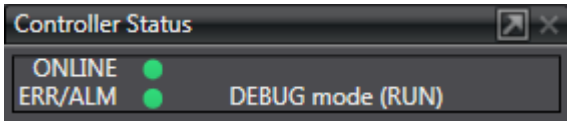



**8** Click the **Yes** Button.
The following device confirmation dialog box is displayed.




**9** Click the **OK** Button.
**Note** You do not need to enter anything in the **Password** Box.

**10** After the Simulator is started and connected, the Online Indicator in status monitor is lit green.
In this status, the project is completely transferred to the Simulator and RUN mode continues.




**11** To stop the simulation, click the **Stop** Button in the Safety Simulation Pane.




**12** To end the simulation, click the Close Button **x** in the Safety Simulation Pane.
The values of variables return to their initial values.

### Pausing

Use the following procedure to pause a simulation.




The values of variables are retained.
Program execution stops at the start of the program.

## 6-8-2 Monitoring

You can monitor the present values of variables in the FBD editor or Watch Tab Page to debug the safety programs.

Refer to *7-6-3 Monitoring Variables in the FBD Editor* on page 7-29 and *7-6-4 Monitoring Variables in a Watch Tab Page* on page 7-30 for details.

## 6-8-3 Controlling BOOL Variables, Changing Present Values, and Using Forced Refreshing

You can debug the safety program by controlling BOOL variables (Set/Reset), changing present values, and executing forced refreshing from the Sysmac Studio.

Refer to *7-6-5 Controlling BOOL Variables, Changing Present Values, and Using Forced Refreshing* on page 7-32 for detailed procedures.

## 6-8-4 Cross References

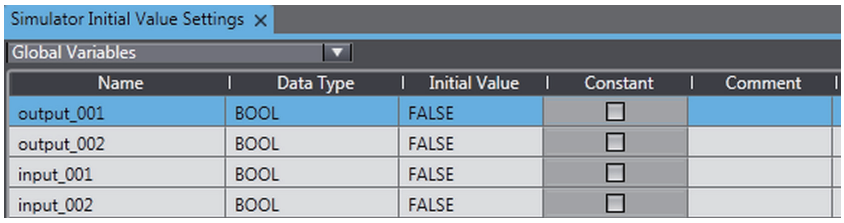
Cross references allow you to see the programs and locations where variables of the safety program are used.
Refer to *7-6-6 Cross References* on page 7-40 for a detailed procedure.

## 6-8-5 Setting the Initial Values of Variables

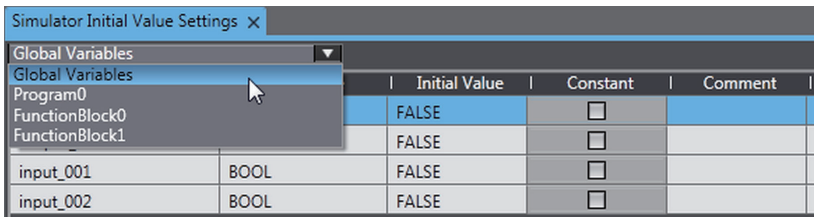
You can set the initial values of variables when you start execution of simulation.
This is useful for reproducing the actual conditions of the system or to evaluate test cases of similar input conditions.
Simulation must be stopped to set the initial values of variables.

Use the following setting procedure.

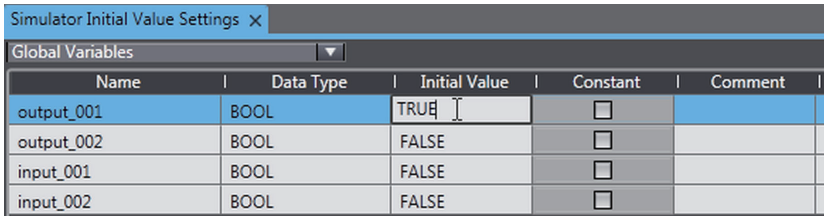
***1*** Select **Simulation** – **Initial Value Settings**.
The Initial Value Settings Dialog Box is displayed.




**2** Select the type of variables.
The selected type of variables is displayed. The following types of variables can be selected: global variables, programs, and function blocks.




**3** Change the initial values of the variables.

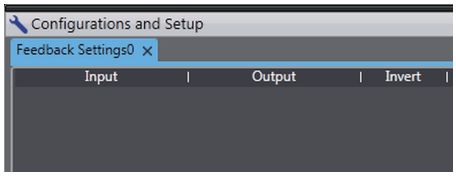



## 6-8-6 Feedback Settings

You can set input status that is linked to changes in output status, such as feedback inputs for safety relays.
Simulation must be stopped to make the feedback settings.

Use the following setting procedure.

**1** Select **Simulation** – **Feedback Settings**.
The Feedback Setting Dialog Box is displayed.



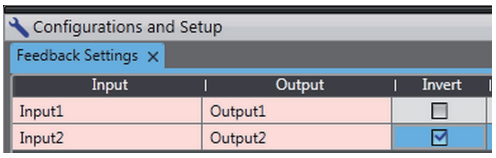

| Item | Description | Supported variable types |
|---|---|---|
| Input | Set a global variable*1 that is the destination of the feedback.<br>You cannot set a variable that is already set as a feedback destination. | BOOL, SAFEBOOL*2 |
| Output | Set the global variable*1 that is the source of the feedback. | BOOL, SAFEBOOL*2 |

| Item | Description | Supported variable types |
|---|---|---|
| Invert | This check box is used to invert the input value.<br>If you select this check box, the input value is inverted. | --- |

*1. With Sysmac Studio version 1.11, you can use device variables and exposed variables only.
*2. With Sysmac Studio version 1.11, set variables with the same variable type for both the source and destination.
With Sysmac Studio version 1.12 and later, you can set different variable types for the source and destination.

**2** Press the **Insert** Key in the feedback setting table, or right-click in the feedback setting table and select **Create New** from the menu.
Cells are added for new settings.
You can set up to 128 sets of feedback settings.

**3** Set the input source variable and the output destination variable, and if necessary, select the Invert Check Box.
The feedback settings are applied within a maximum of 300 ms.




**Precautions for Correct Use**

When you use the feedback settings on the Simulator, set **MonitoringTime** in the SF_EDM instruction 300 ms or more.
Make sure to return the setting to the original value when you transfer the program to the physical Safety CPU Unit.

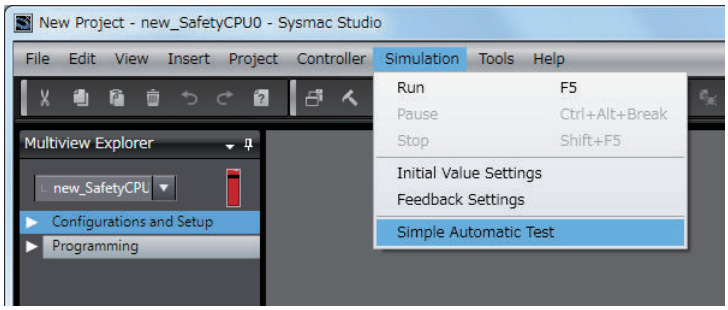
## 6-8-7 Simple Automatic Test

You can use the Simulator of the Safety CPU Unit to easily confirm if the input signals to the program produce the expected output signal values.

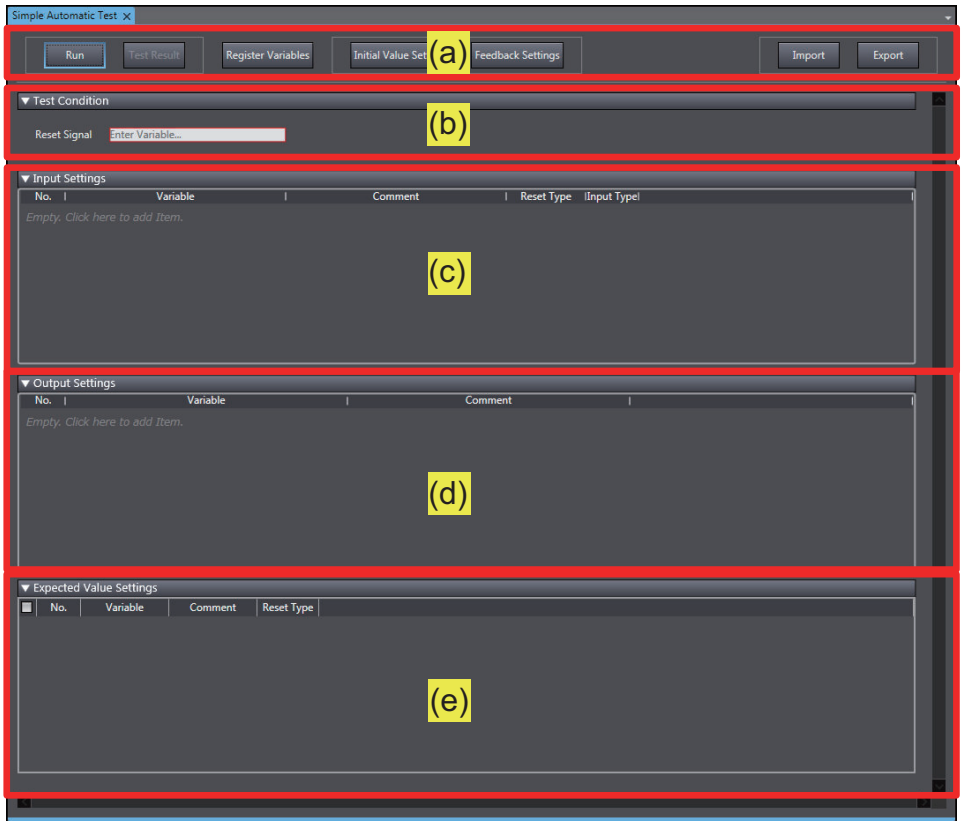
### Settings for a Simple Automatic Test

The Simple Automatic Test automatically executes offline debugging operations for safety programs. To use the Simple Automatic Test, set the reset signal, input signals, and output signals, as well as the expected values of the output signals based on the input signals.
Execute the Simple Automatic Test after you have created the program. Also, set initial values and feedback settings if they are necessary.

Select **Simulation** - **Simple Automatic Test** from the menu to display the setting areas for a Simple Automatic Test.




The following Simple Automatic Test Tab Page is displayed.




The setting areas of the Simple Automatic Test are configured as described in the following table.

| Item | Name | Function | |
|---|---|---|---|
| (a) | Simple Automatic Test Operation Panel | The operation panel is used to make settings for and execute the Simple Automatic Test. You can perform the following operations. | |
| | | **Run** Button | Executes the Simple Automatic Test. |
| | | **Test Result** Button | Shows the previous test results for the Simple Automatic Test. |
| | | **Register Variables** Button | Used to register the variables that are used for the Simple Automatic Test. |
| | | **Initial Value Settings** Button | Displays the Initial Value Settings Dialog Box. |
| | | **Feedback Settings** Button | Displays the Feedback Settings Dialog Box. |

| Item | Name | Function | |
|---|---|---|---|
| | | **Import** Button | Imports the settings for a Simple Automatic Test from a CSV file. If you import the settings, the current settings are overwritten. |
| | | **Export** Button | Exports the current settings for the Simple Automatic Test to a CSV file. |
| (b) | Test Condition Area | Select the variable to use as a reset signal for the safety program. You can set any SAFEBOOL or BOOL variable that is defined in the global variable table. | |
| (c) | Input Setting Area | Set the variables of the input signals to use in the Simple Automatic Test.<br>In addition to the variable name, set the reset types and input types of the input signals. | |
| | | Variable | Set the names of the variables used as the input signals in the program. You can set any SAFEBOOL or BOOL variable that is defined in the global variable table. |
| | | Reset Type | Select the reset type in the box. The default value of the reset time is a manual reset.<br>• Manual<br>If a manual reset is used for an input signal, a test is executed for the manual reset scenario that uses the reset signal set in the *Test Condition Area (b)*.<br>• Auto<br>If an automatic reset is used for an input signal, a test is executed according to the auto reset scenario. |
| | | Input Type | To set a test for two input signals, such as for a safety door or two-hand switches, set the input type to **2 Inputs**. If you set the input type to **2 Inputs**, a row is added to specify another variable. |
| (d) | Output Setting Area | Set the variables of the output signals to use in the Simple Automatic Test. You can set any SAFEBOOL or BOOL variable that is defined in the global variable table. | |
| (e) | Expected Value Setting Area | A matrix of the variables that are specified in the input settings and output setting areas is displayed. Set the test target and the expected value of the output variable for each input variable.<br>The expected values that you set for the output variables have the following meanings:<br>• 0: If the input variable changes to FALSE, the output variable changes to FALSE.<br>• 1: If the input variable changes to FALSE, the output variable changes to TRUE.<br><br>You can import and export expected value settings to use spreadsheets or other applications to easily edit the settings by copying and pasting them. | |

## Setting Example

A setting example for the Simple Automatic Test is provided in this section for the application example given in *A-3-2 Safety Doors* on page A-32.

Application Overview from *A-3-2 Safety Doors* on page A-32 is as follows.

| Safety category/PL | Safety device | Stop category | Reset |
|---|---|---|---|
| Equivalent to 4/PLe (Safety Door) | Safety limit switches 1 and 2 | 0 | Auto |
| | Emergency stop pushbutton | 0 | Manual |

M1 stops when safety door 1 (S3, S4) is opened.
M2 stops when safety door 2 (S5, S6) is opened.
Both M1 and M2 stop when the emergency stop pushbutton S1 is pressed.

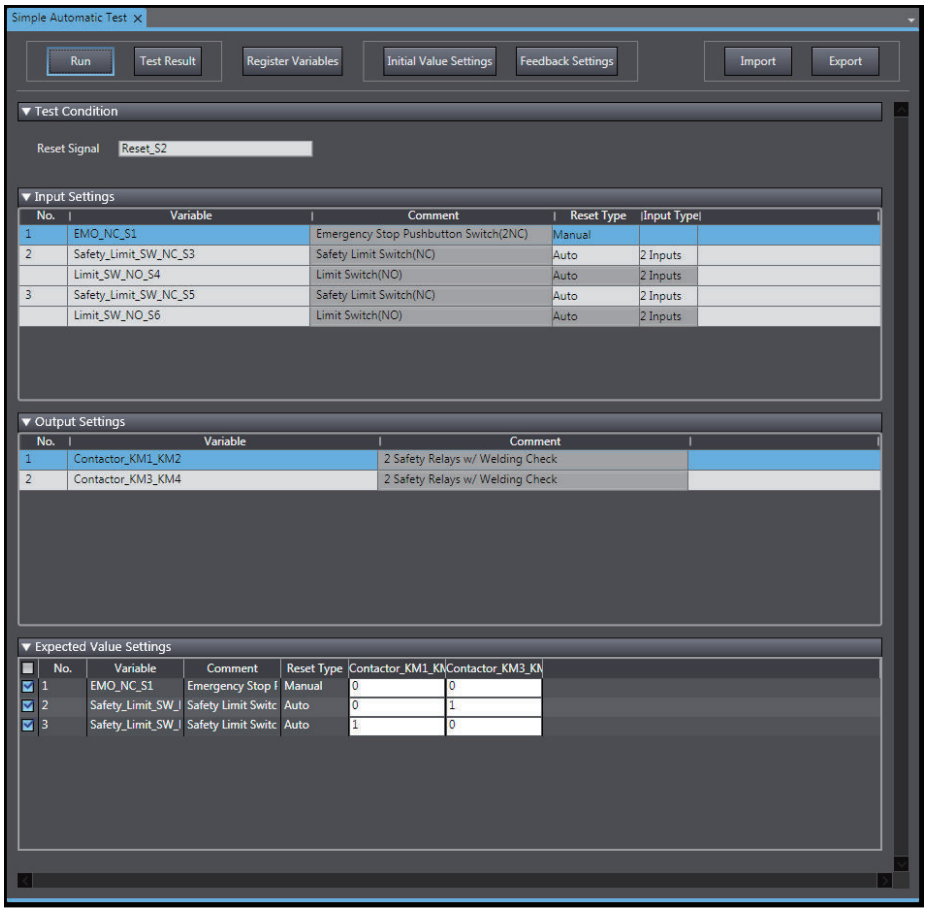
The settings for the Simple Automatic Test are derived from the following points.

- Variable that is assigned to the reset switch
- Variables that are assigned to safety input devices (except for EDM feedback)
- Variables that are assigned to safety output devices

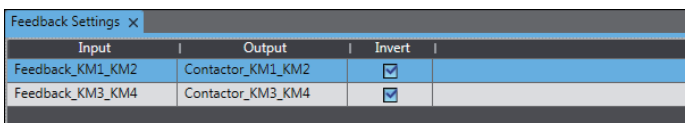
**Note 1.** To make the program work correctly, set the initial values for simulation so that Activate terminal on the safety function block will be to TRUE.

**Note 2.** To make the SF_EDM function block work correctly, set feedback for simulation.

The settings for the above application are shown in the setting areas for the Simple Automatic Test.




Feedback Settings Tab Page is shown below.



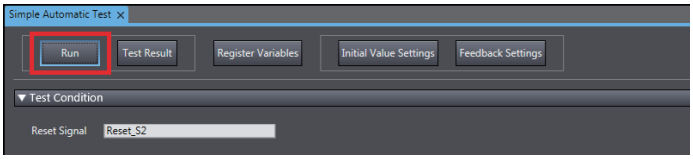

## Execution Procedure for a Simple Automatic Test

Use the following procedure to execute the Simple Automatic Test.

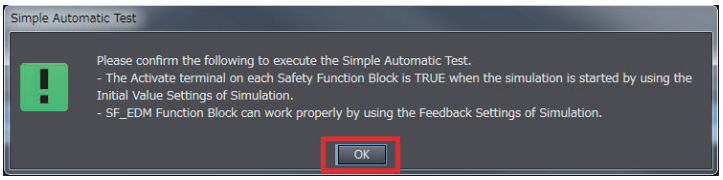
**1** Select **Simulation** - **Simple Automatic Test** from the menu.
The Simple Automatic Test Tab Page is displayed.

**2** Set the **Test Condition**, **Input Settings**, **Output Settings**, and **Expected Value Settings**.

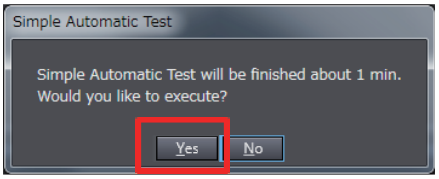
**3** Click the **Run** Button on the control panel on the Simple Automatic Test Tab Page.



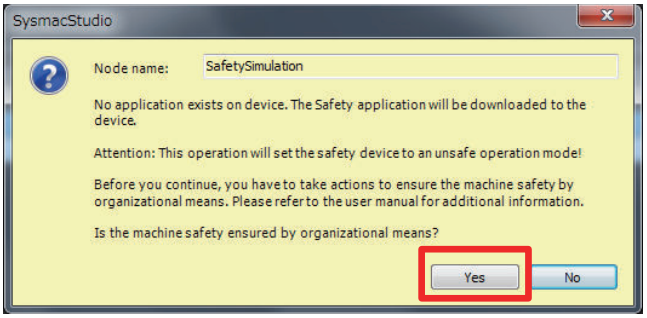

**4** A precaution is displayed to ensure proper execution of the Simple Automatic Test. Read the precaution and then click the **OK** Button.



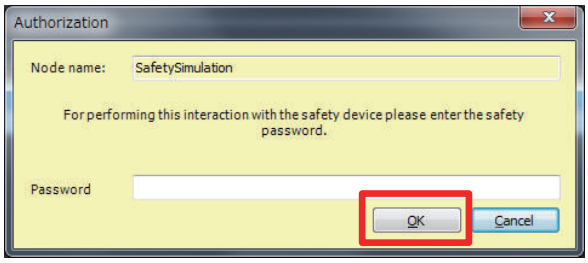

**5** The estimated execution time for the Simple Automatic Test is displayed. Click the **Yes** Button to execute the test.



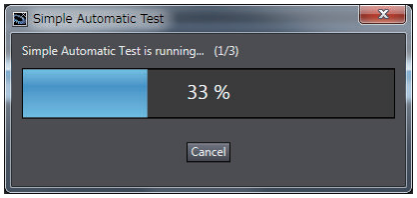

**6** The Simulator for the Safety CPU Unit starts. Following the on-screen instructions to complete starting the Simulator.
Click the **Yes** Button.



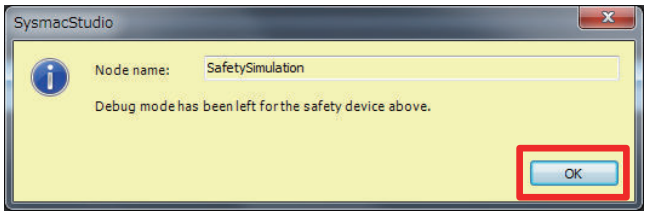

Click the **OK** Button.



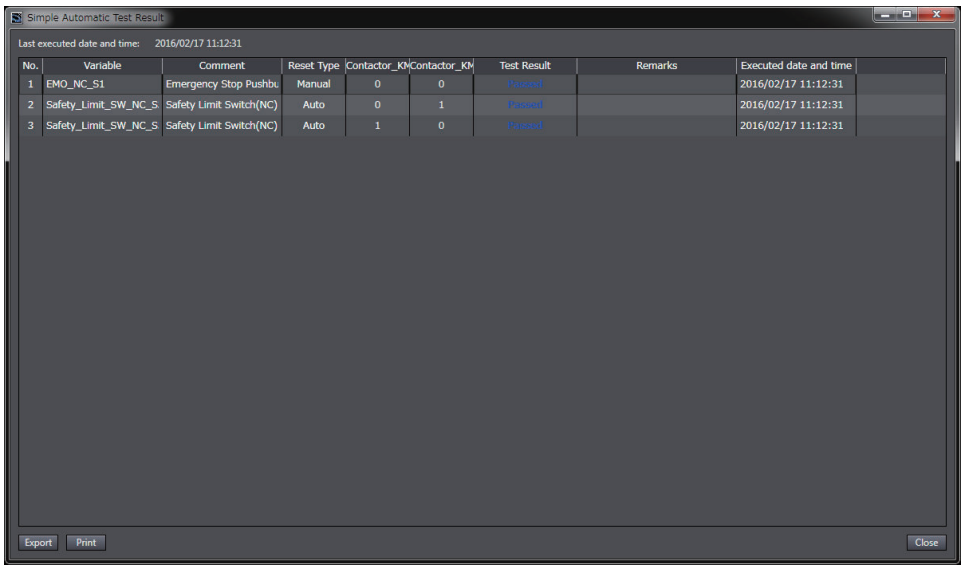

You can check the progress of the Simple Automatic Test in the following dialog box.




The Simulator stops when the test is completed. Click the **OK** Button.




**7** The Simple Automatic Test Result Window is displayed when the test is completed. If the results show that the test has failed, correct the safety program or check the test settings.



**Precautions for Correct Use**

The Simple Automatic Test is provided as a simple means to check the output signal results for the input signals. It may not be able to determine correct test results for complicated conditions or special cases. If necessary, check the program logic manually.

**Additional Information**

The actual test operations that are performed for the Simple Automatic Test are given in *A-15 Execution Scenarios for the Simple Automatic Test* on page A-103.

# 7

# Checking Operation and Actual Operation

This section describes the procedures to perform before you can operate the Safety CPU Unit. It describes the operating modes of the Safety CPU Unit, checking operation in DEBUG mode, and the procedures to perform safety validation.


**7-1 Procedures before Operation and Transferring the Required Data ..... 7-3**
7-1-1 Commissioning Procedure ..... 7-3
7-1-2 Data That You Must Transfer before Operation and Data Transfer Procedures ..... 7-4

**7-2 Transferring the Configuration Information ..... 7-7**
7-2-1 Overview ..... 7-7
7-2-2 Transfer Procedure for a Connection to NJ/NX-series CPU Unit ..... 7-8
7-2-3 Transfer Procedure for a Connection to Communications Coupler Unit ..... 7-8

**7-3 Operating Modes of the Safety CPU Unit ..... 7-11**
7-3-1 Startup Operating Mode and Changing the Operating Mode ..... 7-11
7-3-2 Operation When Changing Operating Mode ..... 7-13
7-3-3 Executable Functions in Each Mode of the Safety CPU Unit ..... 7-14

**7-4 Changing to DEBUG Mode ..... 7-17**

**7-5 Checking External Device Wiring ..... 7-20**
7-5-1 Overview of Functions for Checking Wiring ..... 7-20
7-5-2 Monitoring Safety I/O Units ..... 7-20
7-5-3 Troubleshooting Safety I/O Terminals ..... 7-24
7-5-4 Clear All Memory Operation for Safety I/O Units ..... 7-25

**7-6 Functions for Checking Operation ..... 7-28**
7-6-1 Overview of Functions for Checking Operation ..... 7-28
7-6-2 Starting and Stopping the Safety Programs in DEBUG Mode ..... 7-28
7-6-3 Monitoring Variables in the FBD Editor ..... 7-29
7-6-4 Monitoring Variables in a Watch Tab Page ..... 7-30
7-6-5 Controlling BOOL Variables, Changing Present Values, and Using Forced Refreshing ..... 7-32
7-6-6 Cross References ..... 7-40

**7-7 Online Functional Test ..... 7-44**
7-7-1 Online Functional Test Settings ..... 7-44
7-7-2 Online Functional Test Execution Procedure ..... 7-48

**7-8 Search FB Where Safety Output Is OFF ..... 7-54**
7-8-1 Procedure for Operating Search FB Where Safety Output Is OFF ..... 7-54
7-8-2 Editing Function Blocks to be Searched for ..... 7-55
7-8-3 Registering a Data Type in the Function Block Search Settings Window ..... 7-56

**7-9 Node Name ..... 7-58**

**7-10 Security Settings ..... 7-59**
7-10-1 Setting the Safety Password ..... 7-59
7-10-2 Data Protection ..... 7-60

**7-11 Performing Safety Validation and Operation ..... 7-65**
7-11-1 Performing Safety Validation ..... 7-65
7-11-2 Changing to RUN Mode ..... 7-67
7-11-3 Changing to PROGRAM Mode ..... 7-68

**7-12 Starting and Stopping the Safety Application Monitoring ..... 7-70**
7-12-1 Procedure to Start and Stop the Safety Application Monitoring ..... 7-70
7-12-2 Changing the Monitoring Options for the Safety Application ..... 7-71

**7-13 Uploading Configuration Information and Safety Application Data ..... 7-72**
7-13-1 Outline ..... 7-72
7-13-2 Uploading Data for a Connection to an NJ/NX-series CPU Unit ..... 7-72
7-13-3 Uploading Data for a Connection to a Communications Coupler Unit ..... 7-73

**7-14 Transferring Safety Application Data ..... 7-75**

**7-15 Monitoring Controller Status ..... 7-76**

**7-16 Restarting and Clearing All Memory ..... 7-78**
7-16-1 Restarting ..... 7-78
7-16-2 Clear All Memory Operation ..... 7-79

# 7-1 Procedures before Operation and Transferring the Required Data

This section describes the procedures to perform after you create the safety programs and are ready to operate the Safety CPU Unit. It also describes the data that you must transfer.

## 7-1-1 Commissioning Procedure

After you write the safety programs, use the following procedure to start operating the Safety CPU Unit.

***1*** Place the Sysmac Studio online with Safety CPU Unit.

***2*** Transfer the configuration information to the NJ/NX-series CPU Unit*1 *2, Communications Coupler Unit, Safety CPU Unit, and NX-series EtherNet/IP Unit*2.
When transferring the validated safety application data, the following procedure is not necessary.
After transferring the validated safety application data, check if the data transferred from the safety signature shown in the Sysmac Studio is correct.

*1. Transferred when you use an NJ/NX-series CPU Unit and EtherCAT Coupler Unit.
*2. Transferred when you use CIP Safety communications using an NX502 CPU Unit and NX-series EtherNet/IP Unit.

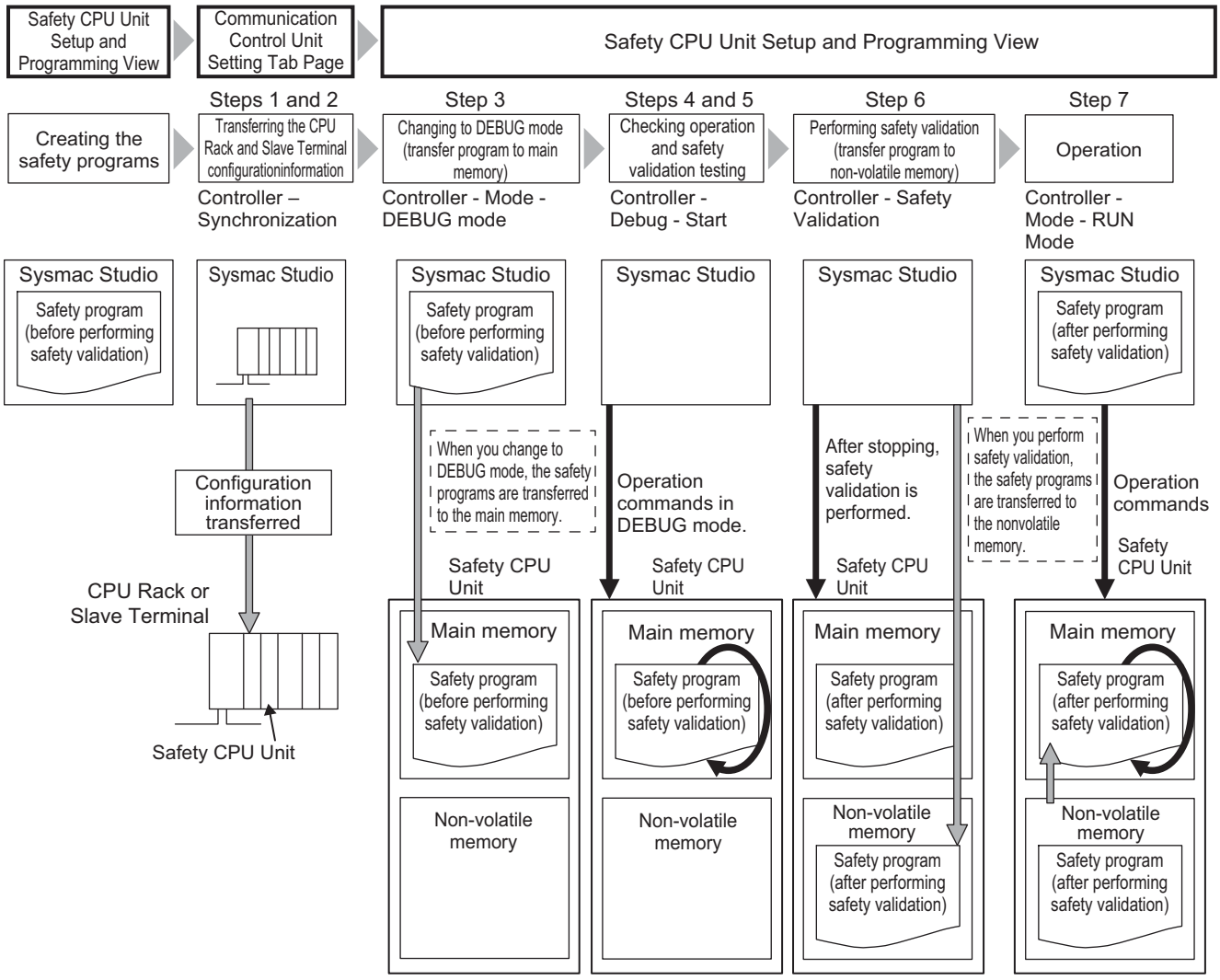
***3*** Place the Safety CPU Unit in DEBUG mode.
The safety programs are transferred to the main memory of the Safety CPU Unit.

***4*** Check the wiring and the operation of the safety programs.

***5*** Perform safety validation testing.

***6*** Validate safety from the Sysmac Studio.
The safety programs are transferred to the non-volatile memory of the Safety CPU Unit to enter the safety-validated status.

***7*** Place the Safety CPU Unit in RUN mode.
The safety programs in the non-volatile memory of the Safety CPU Unit are executed.




## 7-1-2 Data That You Must Transfer before Operation and Data Transfer Procedures

Before you change to RUN mode, you must transfer the CPU Rack or the Slave Terminal settings, including the Safety Control Unit settings, to the Safety CPU Unit. The settings and transfer procedures are given below.

### ● Configuration in which the Safety Control Unit is included in the CPU rack

| CPU rack settings | | Transfer method | Transfer destination |
|---|---|---|---|
| CPU rack configuration information | Unit configuration information | Perform the following operation with the Controller set to the NX-series CPU Unit.<br>Transfer the NX bus master settings in the Synchronization Window.*1 | NX-series CPU Unit |
| | I/O allocation information | | NX-series CPU Unit and Safety CPU Unit |

| CPU rack settings | | Transfer method | Transfer destination |
|---|---|---|---|
| Safety application data | Unvalidated safety application data | The safety application data is transferred automatically when you change to DEBUG mode from the Safety CPU Unit Setup and Programming View. | Safety CPU Unit |
| | Validated safety application data | In DEBUG mode, execute Safety Validation from the Safety CPU Unit Setup and Programming View. This will cause the safety application data to be saved to the non-volatile memory. | |
| | | Perform the following operation with the Controller set to the NX-series CPU Unit.<br>Transfer the NX bus master settings in the Synchronization Window.*1 | |

*1. The entire CPU rack of the NX-series CPU Unit restarts after the NX bus master settings are transferred.

## ● Configuration in which the Safety Control Unit is included in the EtherCAT Slave Terminal

| Slave Terminal settings | | Transfer method | | Transfer destination |
|---|---|---|---|---|
| | | Connection to an NJ/NX-series CPU Unit*1 | Connection to USB port on EtherCAT Coupler Unit*2 | |
| EtherCAT network configuration information*1 | | Perform the following operation with the Controller set to the NJ/NX-series CPU Unit.<br>Transfer the EtherCAT network configuration information from the Synchronization Window. | Transfer is not allowed. | NJ/NX-series CPU Unit |
| Slave Terminal configuration information | Unit configuration information | Perform the following operation with the Controller set to the NJ/NX-series CPU Unit.<br>Transfer the EtherCAT network configuration information and the Slave Terminal configuration information from the Synchronization Window. | Transferred on the Slave Terminal Tab Page. | Communications Coupler Unit |
| | I/O allocation information | | | Communications Coupler Unit and Safety CPU Unit |
| Safety application data | Unvalidated safety application data | The safety application data is transferred automatically when you change to DEBUG mode from the Safety CPU Unit Setup and Programming View. | | Safety CPU Unit |
| | Validated safety application data | In DEBUG mode, execute **Safety Validation** from the Safety CPU Unit Setup and Programming View. This will cause the safety application data to be saved to the non-volatile memory. | | |
| | | Perform the following operation with the Controller set to the NJ/NX-series CPU Unit.<br>Transfer the EtherCAT network configuration information and the Slave Terminal configuration information from the Synchronization Window.*3 | Transferred on the Slave Terminal Tab Page.*3 | |

*1. This is required only when you use an EtherCAT network.

*2. In the following cases, only the Slave Terminal is restarted after the data is transferred.

- When you transfer the data through the NJ/NX-series CPU Unit without using the Sysmac Studio's synchronization operation.
- When you transfer the data with the Sysmac Studio connected to the USB port on the Communications Coupler Unit

The restart may cause the EtherCAT master to detect an error. If an error is detected, you need to reset the error in the EtherCAT master.

*3. Supported only by Sysmac Studio version 1.10 or higher.

**Version Information**

You can download safety-validated safety application data with Sysmac Studio version 1.10 or higher and NX-SL□□□□ version 1.1 or later.

# 7-2 Transferring the Configuration Information

This section describes how to start communications and transfer configuration information from the Sysmac Studio to the NJ/NX-series CPU Unit*1, the NX-series Communications Coupler Unit, and the NX-series Safety CPU Unit.

*1. This can be selected and transferred when you use an NJ/NX-series CPU Unit and EtherCAT Coupler Unit.

## 7-2-1 Overview

You must transfer the CPU Rack settings and Slave Terminal settings to the NJ/NX-series CPU Unit*1, the Communications Coupler Unit, and the Safety CPU Unit before you can place the Sysmac Studio online with the Safety CPU Unit and begin debugging.

*1. This can be selected and transferred when you use an NJ/NX-series CPU Unit and EtherCAT Coupler Unit.

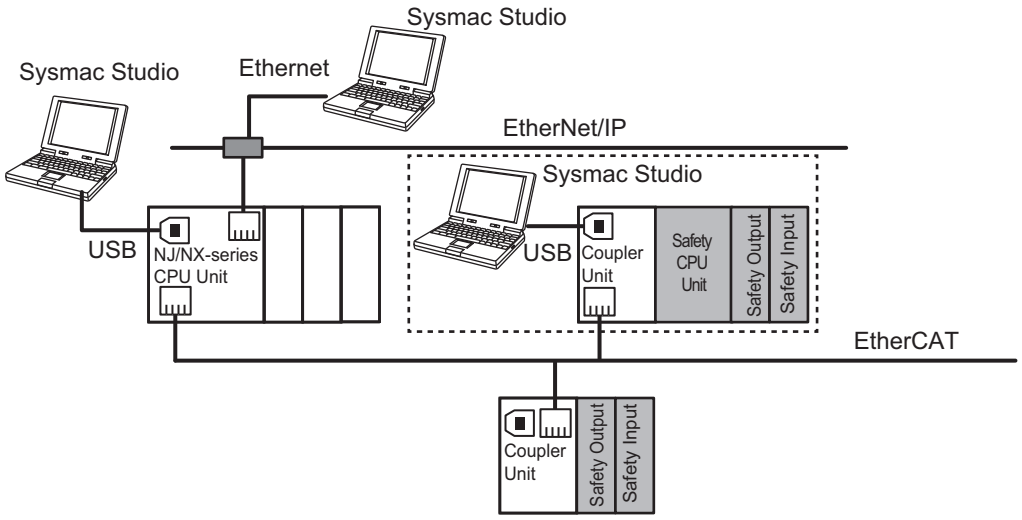
### Paths for Going Online

There are the following two ways to connect the Sysmac Studio to the Safety CPU Unit.

- USB Connection or Ethernet Connection to the NJ/NX-series CPU Unit*1, or Ethernet connection to the NX-series EtherNet/IP Unit
- USB connection to the EtherCAT Coupler Unit

*1. This can be selected and transferred when you use an NJ/NX-series CPU Unit and EtherCAT Coupler Unit.

A configuration example for an EtherCAT Slave Terminal is given below.



**⚠ WARNING**

Always confirm safety at the destination node before you transfer Unit configuration information, parameters, settings, or other data from tools such as the Sysmac Studio. The devices or machines may perform unexpected operation regardless of the operating mode of the CPU Unit.

## 7-2-2 Transfer Procedure for a Connection to NJ/NX-series CPU Unit

When you use an EtherCAT Coupler Unit, you can connect the Sysmac Studio to a USB or Ethernet port on the NJ/NX-series CPU Unit or EtherNet/IP port on the NX-series EtherNet/IP Unit to transfer the configuration information.
Use the following procedure to go online with the Safety CPU Unit.

1. Select the NJ/NX-series CPU Unit from the Controller Selection Box in the Multiview Explorer on the Sysmac Studio to change to the NJ/NX-series CPU Unit Setup and Programming View.
2. Set the communications path to the NJ/NX-series CPU Unit.
3. Select **Online** from the **Controller** Menu. Or, click the Go Online Button (  ) in the toolbar.
4. Select **Synchronization** from the **Controller** Menu.
5. Click the **Transfer to Controller** Button.*1 *2
   The Sysmac Studio is enabled for communications with the EtherCAT Coupler Unit and the Safety CPU Unit. This also transfers the Slave Terminal configuration information to the EtherCAT Coupler Unit and the Safety CPU Unit.

*1. Always click this button when you go online with the Safety CPU Unit for the first time, or if you change a Safety I/O Unit or variable data.
*2. If the CIP Safety communications are performed with the EtherNet/IP port on the mounted NX-series EtherNet/IP Unit, when the settings for CIP Safety communications are changed, always transfer the settings to the EtherNet/IP Unit.

**Additional Information**

- Refer to the *Sysmac Studio Version 1 Operation Manual (Cat. No. W504)* for information on connecting and synchronizing with the NJ/NX-series CPU Unit or NX-series EtherNet/IP Unit.
- Refer to *7-4 Changing to DEBUG Mode* on page 7-17 for details on transferring the unvalidated safety program.

## 7-2-3 Transfer Procedure for a Connection to Communications Coupler Unit

You can connect the Sysmac Studio to a USB port on a Communications Coupler Unit to transfer the configuration information.
Use one of the following two procedures to go online with the Safety CPU Unit.

## ● Online Procedure from the NJ/NX-series CPU Unit Setup and Programming View

You can use this method only if you use an EtherCAT Coupler Unit.

***1*** Connect the computer to the EtherCAT Coupler Unit with a USB cable.

***2*** Select the NJ/NX-series CPU Unit from the Controller Selection Box in the Multiview Explorer on the Sysmac Studio to change to the NJ/NX-series CPU Unit Setup and Programming View.

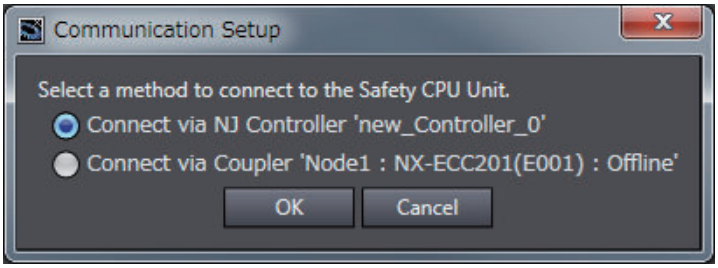
***3*** Right-click the EtherCAT Coupler Unit on the EtherCAT Slave Terminal Tab Page, and select **Coupler Connection (USB)** − **Online** from the menu.
The Sysmac Studio goes online with the EtherCAT Slave Terminal.

## ● Online Procedure from the Safety CPU Unit Setup and Programming View

***1*** Connect the computer to the Communications Coupler Unit with a USB cable.

***2*** Select Safety CPU Unit from the Controller Selection Box in the Multiview Explorer of Sysmac Studio and open the Safety CPU Unit Setup and Programming View.

***3*** Select **Communication Setup** from the **Controller** Menu.
The following communications settings are displayed.



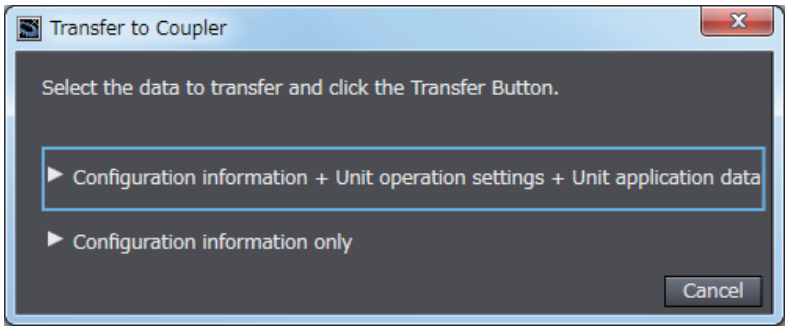

***4*** Select *Connect via Coupler* and then click the **OK** Button.

***5*** Select **Online** from the **Controller** Menu. Or, click the Go Online Button ( ) in the toolbar.
The Sysmac Studio goes online with the Slave Terminal.

## ● Procedure to Transfer the Configuration Information

Use the following procedure to transfer the configuration information to an online Slave Terminal.

***1*** Right-click the Communications Coupler Unit on the Slave Terminal Tab Page, and select **Transfer to Coupler** from the menu.
The following dialog box is displayed.




**2** Click **Configuration information only** or **Configuration + Unit operation settings + Unit application data.**.

An execution confirmation dialog box for **Transfer to Coupler** is displayed.

For both options, only the Slave Terminal configuration information is transferred to the Communications Coupler Unit and the Safety CPU Unit because safety validation has not been performed for the safety program.

**3** Click the **Yes** Button.

The settings are transferred to all Units that are connected to the Communications Coupler Unit.

# 7-3 Operating Modes of the Safety CPU Unit

This section describes the operating modes of the Safety CPU Unit, state changes, and the functions that can be executed in each mode.

## 7-3-1 Startup Operating Mode and Changing the Operating Mode

The operating mode of the Safety CPU Unit changes to PROGRAM mode or RUN mode after the power is turned ON, depending on whether the safety programs are validated, as shown in the following figure.

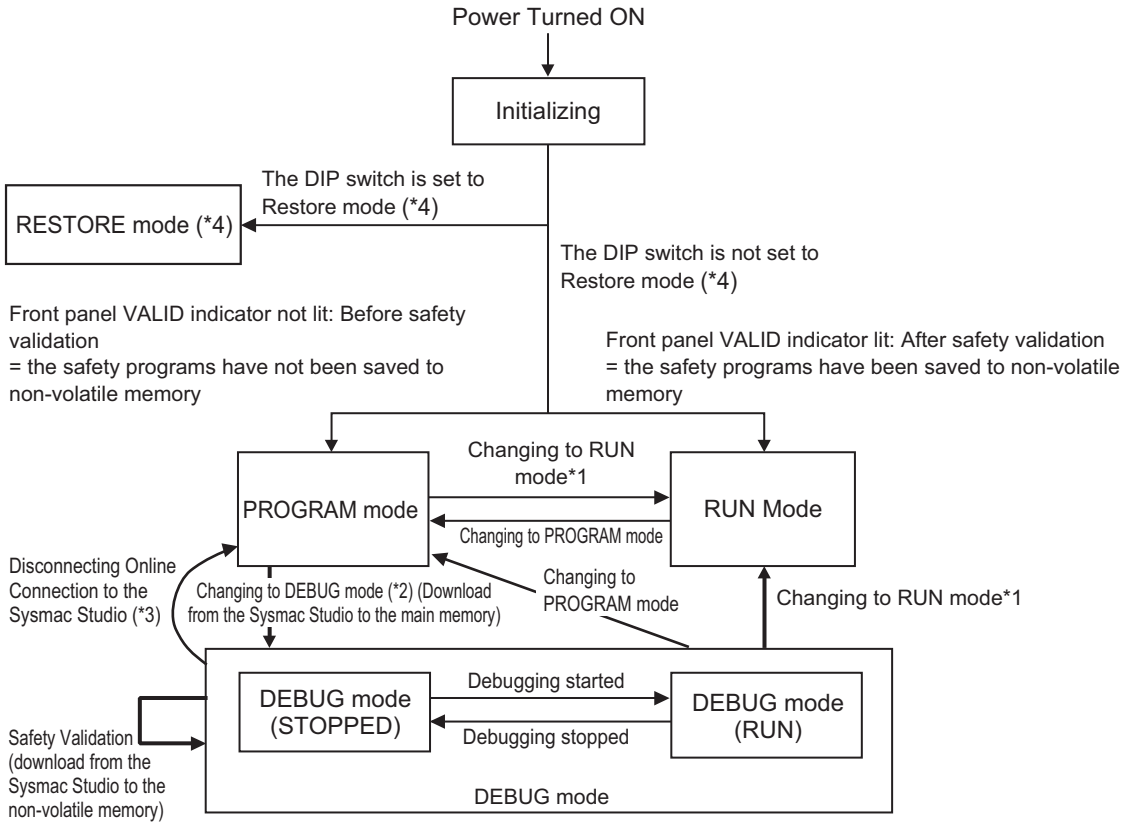
### ● Before Safety Validation

The Safety CPU Unit starts in PROGRAM mode. This prevents the Safety CPU Unit from running a safety application that has not been validated for safety.

### ● After Safety Validation

The Safety CPU Unit starts in the RUN mode.

To change the operating mode of the Safety CPU Unit, select the Safety CPU Unit as the Controller and place the Sysmac Studio online with the Safety CPU Unit, and then select the desired operating mode.




*1. The Safety CPU Unit can be operated only after safety validation is performed.

*2. When the operating mode changes from PROGRAM mode to DEBUG mode, the safety application data in the non-volatile memory of the Safety CPU Unit is deleted.
*3. The Safety CPU Unit automatically enters PROGRAM mode if 30 seconds elapses after the connection to the Sysmac Studio is disconnected due to a damaged cable or any other reason.
*4. The DIP switch setting and the restore mode are not available with NX-SL3300 and NX-SL3500.



**Additional Information**

If you need to use debugging to change present values or other settings while operating in RUN mode (for instance, to troubleshoot a validated safety program), you must stop the machine, and then change the Safety CPU Unit to PROGRAM mode before you can change to DEBUG mode. However, when the operating mode changes from PROGRAM mode to DEBUG mode, the safety programs in the non-volatile memory of the Safety CPU Unit is deleted. Download the safety-validated safety programs to the non-volatile memory in the Safety CPU Unit again.



**Version Information**

The processing that is performed when the operating mode changes from RUN mode to PROGRAM mode depends on the version of the Sysmac Studio that you use.

- Sysmac Studio version 1.09 or lower
  The safety programs in the non-volatile memory of the Safety CPU Unit are deleted. If you return to RUN mode, download the safety-validated safety programs to the non-volatile memory in the Safety CPU Unit again.
- Sysmac Studio version 1.10 or higher
  The safety programs in the non-volatile memory of the Safety CPU Unit are not deleted.

## Operating Modes of Safety CPU Unit and Indicators

The following table specifies details of the Safety CPU Unit operating modes and the indicators displayed for each operating mode.

| Operating mode | | Description | Indicators | |
|---|---|---|---|---|
| | | | RUN | DEBUG |
| Initializing | | This mode indicates that the Safety CPU Unit is starting up and performing hardware self-diagnosis. | Flashing green. | Not lit |
| PROGRAM mode | | This mode indicates that safety communications and the safety program execution are stopped.<br>You can clear or download the safety application data from the Sysmac Studio (Safety CPU Unit Setup and Programming View). | Not lit | Not lit |
| RUN mode | | This mode indicates that safety communications are executed and the validated safety programs are running. | Lit green | Not lit |
| DEBUG mode | STOPPED | This mode indicates that safety communications are executed and the unvalidated safety programs are on standby.<br>You can control BOOL variables, change present values of data, and use forced refreshing. | Not lit | Lit yellow |
| | RUN | This mode indicates that safety communications are executed and unvalidated safety programs are running.<br>You can control BOOL variables, change present values of data, and use forced refreshing. | Lit green | Lit yellow |
| RESTORE mode | | This mode allows you to edit user programs and settings of the Safety CPU Unit by using an SD Memory Card.<br>Safety communications and safety programs are not executed. | Flashing green. | Flashing yellow |

### ● Relationship between the Operating Modes of the Safety CPU Unit and the NJ/NX-series CPU Unit

The operating mode of the Safety CPU Unit is independent from the operating mode of the NJ/NX-series CPU Unit.

This means that changes in the operating mode of the NJ/NX-series CPU Unit will not affect the operating mode of the Safety CPU Unit.

Conversely, changes in the operating mode of the Safety CPU Unit will not affect the operating mode of the NJ/NX-series CPU Unit.

### ● Restrictions in DEBUG Mode

Safety certification bodies prohibit safety controllers from entering full operational state while forced refreshing is enabled. Therefore, the Safety CPU Unit imposes the following restrictions.

- You can perform forced refreshing only in DEBUG mode.
- You can change to DEBUG mode only when the Sysmac Studio is online with the Safety CPU Unit. If an online connection is broken in DEBUG mode, the Safety CPU Unit will automatically change to PROGRAM mode after 30 seconds elapses.
- Only one copy of the Sysmac Studio can go online with the Safety CPU Unit at the same time.

## I/O Data in Each Operating Mode

The following table explains how I/O data is processed, depending on the operating mode of the Safety CPU Unit.

Not received = Not receive data, Cleared = Clear and reset to initial value, Refreshed = Refresh data

| Operating mode | | Safety Input (FSoE/CIP Safety) | Standard Input (Exposed Variable) | Safety Output (FSoE/CIP Safety) | Standard Output (Exposed Variable) | Status (Safety CPU Status) |
|---|---|---|---|---|---|---|
| Initializing | | Not received | Not received | Cleared | Cleared | Cleared |
| PROGRAM mode | | Not received | Not received | Cleared | Cleared | Refreshed |
| RUN mode | | Refreshed | Refreshed | Refreshed | Refreshed | Refreshed |
| DEBUG mode | STOPPED | Refreshed | Refreshed | Refreshed | Refreshed | Refreshed |
| | RUN | Refreshed | Refreshed | Refreshed | Refreshed | Refreshed |
| RESTORE mode | | Not received | Not received | Cleared | Cleared | Cleared |

## 7-3-2 Operation When Changing Operating Mode

If you change the operating mode of the Safety CPU Unit, the Safety CPU Unit will operate as shown in the following table.

| Before change | → | After change | Operation |
|---|---|---|---|
| PROGRAM mode | → | DEBUG mode (STOPPED) | • Safety process data communications start. |
| PROGRAM mode | → | RUN mode | • Safety process data communications start.<br>• The variables are initialized and the safety programs are executed from the beginning. |
| DEBUG mode (STOPPED) | → | DEBUG mode (RUN) | • Safety process data communications continue.<br>• The variables are initialized and the safety programs are executed from the beginning. |

| Before change | → | After change | Operation |
|---|---|---|---|
| DEBUG mode (RUN) | → | DEBUG mode (STOPPED) | • Safety process data communications continue.<br>• The variables are initialized and the safety programs are stopped.<br>• The forced status of variables is cleared. |
| RUN mode | → | PROGRAM mode | • Safety process data communications stop.<br>The safety input data from the Safety Input Units is initialized to 0.<br>The safety output data to the Safety Output Units is initialized to 0.<br>• The variables are initialized and the safety programs are stopped.<br>• The safety programs are deleted from the non-volatile memory of the Safety CPU Unit. |
| DEBUG mode (RUN) | → | PROGRAM mode | • Safety process data communications stop.<br>The safety input data from the Safety Input Units is initialized to 0.<br>The safety output data to the Safety Output Units is initialized to 0.<br>• The variables are initialized and the safety programs are stopped.<br>• The forced status of variables is cleared. |
| DEBUG mode (STOPPED) | → | PROGRAM mode | • Safety process data communications stop.<br>The safety input data from the Safety Input Units is initialized to 0.<br>The safety output data to the Safety Output Units is initialized to 0. |

### Relationship between Establishing Safety Communications and Execution of the Safety Programs

The Safety CPU Unit starts execution of the safety programs at the same time the safety process data communications are established.
The input data that is refreshed from the Safety Input Units is used for processing.

## 7-3-3 Executable Functions in Each Mode of the Safety CPU Unit

The following table shows the executable functions in each mode of the Safety CPU Unit.

| Function *1 | Initializing | PROGRAM mode | DEBUG mode | | RUN mode | RESTORE mode*5 |
|---|---|---|---|---|---|---|
| | | | STOPPED | RUN | | |
| **Safety program execution** | Not possible | Not possible | Not possible | Possible | Possible | Not possible |
| **Controlling BOOL variables, forced refreshing, and changing present values** | Not possible | Not possible | Possible | Possible | Not possible | Not possible |
| **Message communications** | Possible | Possible | Possible | Possible | Possible | Possible |
| **NX bus communications** | Possible*2 | Possible*2 | Possible*2 | Possible*2 | Possible*2 | Possible*2 |

| Function *1 | | Initializing | PROGRAM mode | DEBUG mode | | RUN mode | RESTORE mode*5 |
|---|---|---|---|---|---|---|---|
| | | | | STOPPED | RUN | | |
| Safety process data communications | | Not possible | Not possible | Possible | Possible | Possible | Not possible |
| Downloading (transferring data from the Computer to the Controller) | Configuration information (I/O allocation information) | Not possible | Possible | Possible | Possible | Possible | Not possible |
| | Validated safety application data | | | Not possible | Not possible | Possible*4 | |
| Uploading (transferring data from the Controller to the Computer) | | Not possible | Not possible | Not possible | Not possible | Possible | Not possible |
| Clear All Memory operation for Unit | | Not possible | Possible | Not possible | Not possible | Not possible | Not possible |
| Clear All Memory Operation for Slave Terminal, Clear All Memory Operation for NJ/NX-series CPU Unit*3 | | Not possible | Not possible | Not possible | Not possible | Not possible | Not possible |
| Restarting NX Bus/NX Unit | | Possible | Possible | Possible | Possible | Possible | Possible |
| Restarting Slave Terminal and Resetting Controller for NJ/NX-series CPU Unit | | Possible | Possible | Possible | Possible | Possible | Possible |
| Monitoring Controller status | | Not possible | Possible | Possible | Possible | Possible | Possible |
| Monitoring programs | | Not possible | Not possible | Possible | Possible | Possible | Not possible |
| Monitoring in a Watch Tab Page | | Not possible | Not possible | Possible | Possible | Possible | Not possible |
| Monitoring for troubleshooting | | Not possible | Possible | Possible | Possible | Possible | Possible |
| Changing the safety password | | Not possible | Possible | Not possible | Not possible | Possible | Not possible |
| Node Name Change | | Not possible | Possible | Not possible | Not possible | Possible | Not possible |
| Safety Data Logging*5 | | Not possible | Not possible | Not possible | Not possible | Possible | Not possible |
| Safety Unit Restore*5 | | Not possible | Not possible | Not possible | Not possible | Not possible | Possible |
| Online Functional Test | | Not possible | Not possible | Not possible | Possible | Possible | Not possible |

*1. Hardware Self-diagnosis
In the initializing state, self-diagnosis is performed for all hardware.
In other operating modes, hardware self-diagnosis is performed at fixed intervals.

*2. NX bus communications
The data is refreshed depending on the operating mode, as shown in the following table.

| Operating mode | | Refreshing |
|---|---|---|
| **Initializing** | | The input data is discarded. The output data is fixed to 0. |
| **PROGRAM mode** | | The input data is discarded. The output data that carries status information is transferred. All data outputs from the safety programs change to 0. |
| **RUN mode** | | I/O data refreshing is performed with the safety programs. |
| **DEBUG mode** | **STOPPED** | Input data refreshing is performed with the safety programs.<br>The output data that carries status information is transferred. All data outputs from the safety programs change to 0. |
| | **RUN** | I/O data refreshing is performed with the safety programs. |

*3. You can execute the Clear All Memory operation for the Safety CPU Unit regardless of the operating status of the Safety CPU Unit, but it will always fail for the Safety CPU Unit.

*4. Sysmac Studio version 1.10 or higher and NX-SL□□□□ unit version 1.1 or later are required.

*5. It is available only for NX-SL5500 and NX-SL5700.

# 7-4 Changing to DEBUG Mode

This section describes how to place the Safety CPU Unit into DEBUG mode. When you change to DEBUG mode, the safety application data is transferred to the Safety CPU Unit.

**⚠ WARNING**

Before you use the Sysmac Studio to change the operating mode of the Safety CPU Unit to DEBUG Mode, make sure that it is safe to do so at the destination for the safety application data.
The outputs may operate and may cause serious injury.

## DEBUG Mode Application

DEBUG mode is used to check that the safety communications, the safety programs, and the external devices operate properly before you operate the Safety CPU Unit.
When you place the Safety CPU Unit in DEBUG mode, the unvalidated safety programs are automatically transferred from the Sysmac Studio to the main memory of the Safety CPU Unit.
As a safety precaution, the Sysmac Studio allows you to switch the Safety CPU Unit to DEBUG mode only when the Safety CPU Unit is in PROGRAM mode.

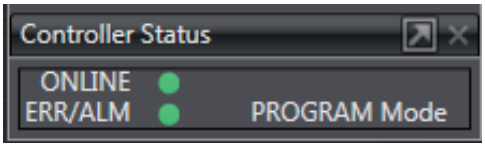
**Precautions for Safe Use**

Verify that the safety communications with a remote node will be established in the debug mode of the Safety CPU Unit.

## Procedure for Changing to DEBUG Mode

Use the following procedure to switch the operating mode of Safety CPU Unit from the PROGRAM to DEBUG mode.
When you change the operating mode to the DEBUG mode, the safety programs must be ready for building.

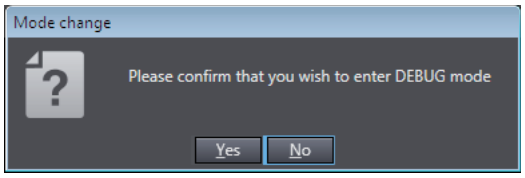
1. Make sure that the Safety CPU Unit is in the PROGRAM mode.
2. Connect to the Safety CPU Unit online.
3. Select the Safety CPU Unit from the Controller Selection Box in the Multiview Explorer of Sysmac Studio and open the Safety CPU Unit Setup and Programming View.
   When you move to the view for Safety CPU Unit, the Sysmac Studio goes online with the Safety CPU Unit, and the Controller status is displayed in the lower right of the Sysmac Studio Window, as shown below.



**4** On the Safety CPU Unit Setup and Programming View, perform one of the following operations.

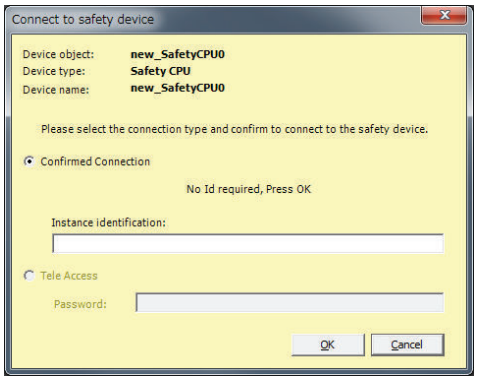
- Select **Mode** – **DEBUG Mode** from the **Controller** Menu.
- Press the **Ctrl** + **2** Keys.
- Click the **DEBUG Mode** Button on the toolbar.

The following mode confirmation dialog box is displayed.




**5** Click the **Yes** Button.

The following Connect to safety device Dialog Box is displayed.

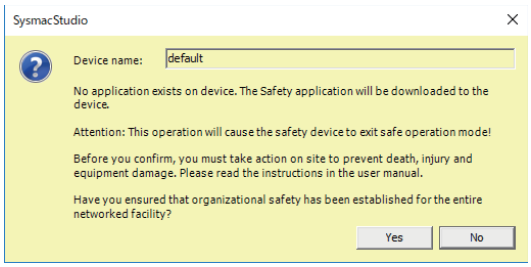



**Note 1.** When you change the operating mode for the first time, the above dialog box is displayed and allows you to set the node name. You can change the factory-default node name of the Safety CPU Unit.
If you do not want to change the node name, leave the field blank and click the **OK** Button. Refer to *7-9 Node Name* on page 7-58 for details on the node name settings.

**Note 2.** You do not need to enter anything in the *Instance identification* Box.

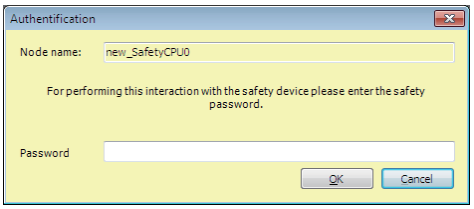
**6** Click the **OK** Button.

The following transfer confirmation dialog box is displayed.




**7** Check the safety of the system and then click the **Yes** Button.

The following password confirmation dialog box is displayed.




**8** When you use the DEBUG mode for the first time, or when the safety password is not specified, leave the **Password** field blank and click the **OK** Button.
When a safety password is specified, enter the security password and click the **OK** Button.
Refer to *7-10 Security Settings* on page 7-59 for the procedure to set a safety password.

Unvalidated safety programs are transferred from the Sysmac Studio to the main memory of the Safety CPU Unit, and the Safety CPU Unit enters DEBUG (STOPPED) mode.

**Precautions for Correct Use**

Before the safety validation is executed, the safety programs are stored in the main memory of the Safety CPU Unit. When the Sysmac Studio goes offline or when you power off Safety CPU Unit, the safety programs are deleted. Because of this, after you cycle the power supply, you must use the Sysmac Studio to change to DEBUG mode again and transfer the safety programs to the main memory again.

## Changing to PROGRAM Mode

If you need to change the safety program, you must change to PROGRAM mode.
Use the following procedure to change the Safety CPU Unit from DEBUG mode to PROGRAM mode.

With the Safety CPU Unit in DEBUG mode, perform one of the following operations.

- Select **Controller** - **Operating Mode** - **PROGRAM Mode**.
- Press the **Ctrl** + **1** Keys.
- Click the **PROGRAM Mode** Button on the toolbar.

The Safety CPU Unit enters PROGRAM mode.

# 7-5 Checking External Device Wiring

This section describes the functions that you use on the Sysmac Studio to check the wiring of external devices connected to the Safety I/O Units.
You can obtain information on the Safety I/O Units to which external devices are connected and confirm that the external devices are correctly wired by placing the Sysmac Studio online with the Safety CPU Unit and changing to DEBUG mode.

## 7-5-1 Overview of Functions for Checking Wiring

This section describes the functions that you use on the Sysmac Studio to check external device wiring.

| Functions for checking wiring | Reference |
|---|---|
| Monitoring Safety I/O terminals | *7-5-2 Monitoring Safety I/O Units* on page 7-20 |
| Monitoring Safety I/O data | *7-5-2 Monitoring Safety I/O Units* on page 7-20 |
| Troubleshooting Safety I/O terminals | *7-5-3 Troubleshooting Safety I/O Terminals* on page 7-24 |
| Clear All Memory Operation for Safety I/O Unit | *7-5-4 Clear All Memory Operation for Safety I/O Units* on page 7-25 |

The functions to check wiring should be executed while you are online to the Safety CPU Unit in DEBUG mode or RUN mode.
If the safety application monitoring is stopped, you need to start monitoring. For the procedure to start monitoring, refer to *7-12 Starting and Stopping the Safety Application Monitoring* on page 7-70.

## 7-5-2 Monitoring Safety I/O Units

This section describes how to monitor I/O terminal information of the Safety I/O Units to check external device wiring.

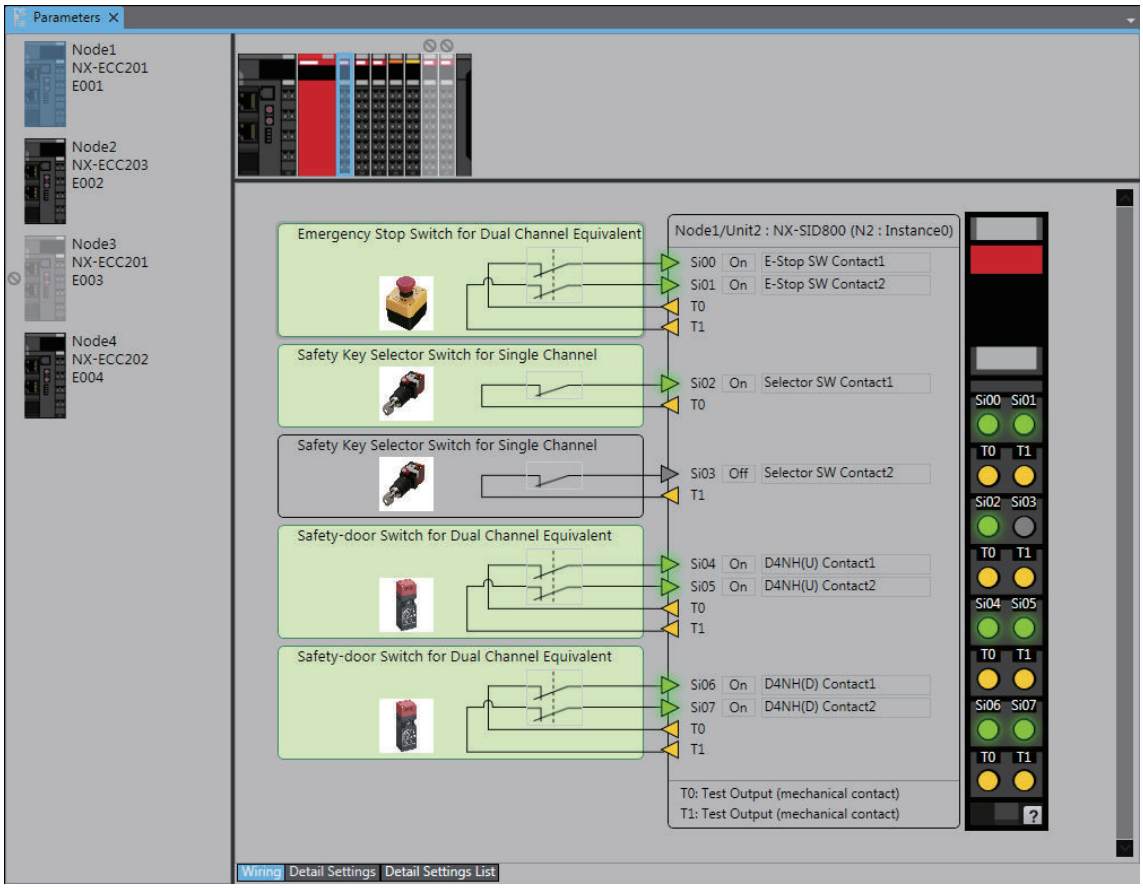
### Executing the Monitor for Safety I/O Terminals

#### ● When Sysmac Studio Is Connected to NJ/NX-series CPU Unit

**1** Connect the Sysmac Studio online to the NJ/NX-series CPU Unit.
In the Multiview Explorer, select the target Safety CPU Unit in the Controller Selection Box.

**2** Place the Safety CPU Unit in DEBUG mode.
Refer to *7-4 Changing to DEBUG Mode* on page 7-17 for a detailed procedure.

**3** Double-click **Parameters** under the name of the Safety I/O Unit under **Configurations and Setup** − **Communications** − **Safety** − **Safety I/O**.
The Parameters Tab Page shown below is displayed. Select the Safety I/O Unit to monitor.

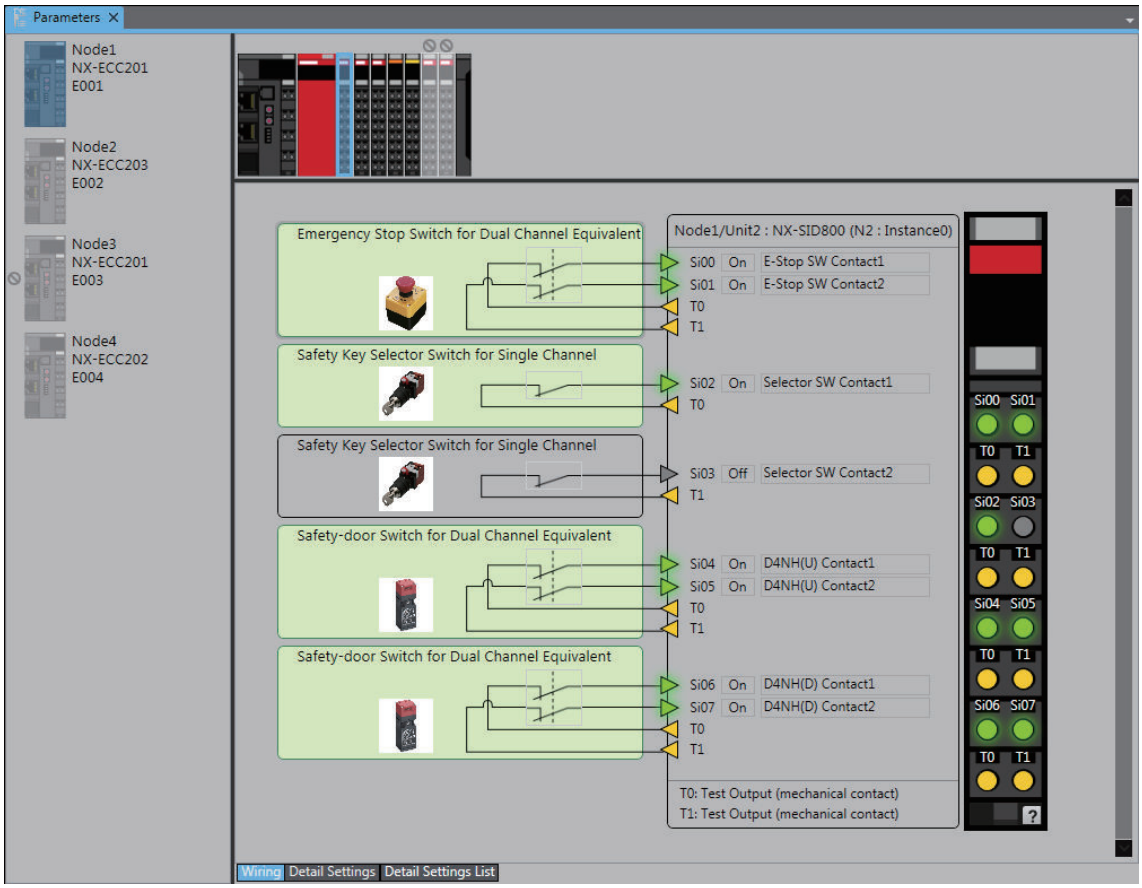



## ● When Sysmac Studio Is Connected to Communications Coupler Unit

***1*** Place the Sysmac Studio online with the Communications Coupler Unit.
Place the Sysmac Studio online with the Communications Coupler Unit that the Safety CPU Unit is connected to. The following procedure cannot be executed on a Communications Coupler Unit that the Safety CPU Unit is not connected to.

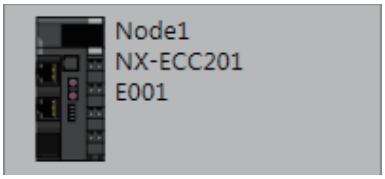
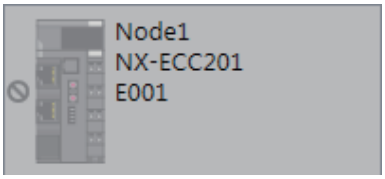
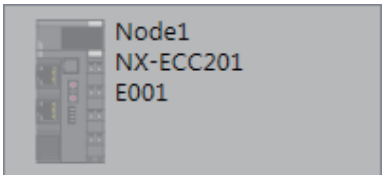
***2*** Place the Safety CPU Unit in DEBUG mode.
Refer to *7-4 Changing to DEBUG Mode* on page 7-17 for a detailed procedure.

***3*** Double-click **Parameters** under the name of the Safety I/O Unit under **Configurations and Setup** − **Communications** − **Safety** − **Safety I/O**.
The Parameters Tab Page shown below is displayed. Select the Safety I/O Unit to monitor.




## Description of Monitor Tab Page

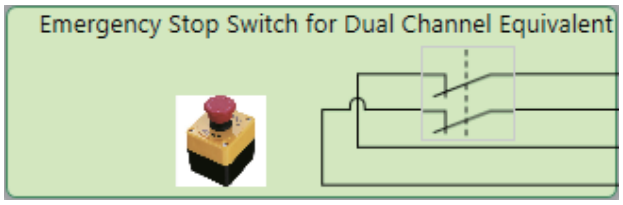
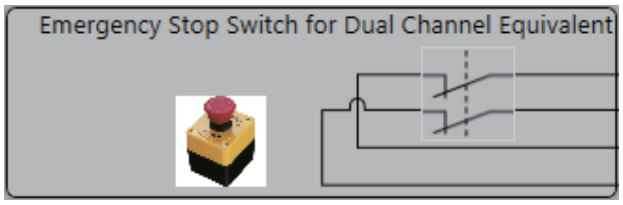
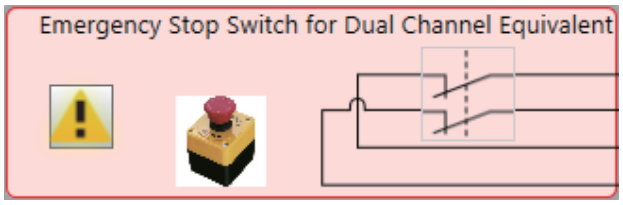
- NX Bus Master Displays
  This section describes for the Communications Coupler Unit as an example.

| Display | Monitor function | Description |
|---|---|---|
|  | Enabled | The NX bus master is present in the actual configuration and communications are established with it. |
|  | Disabled | The Communications Coupler Unit is disabled in the EtherCAT network configuration.<br>Safety I/O Units connected to this Communications Coupler Unit are not monitored. |
|  | Disabled | Either the NX bus master is not present in the actual configuration or communications cannot be established with it.<br>Safety I/O Units connected to this NX bus master are not monitored. |

- NX Unit Displays

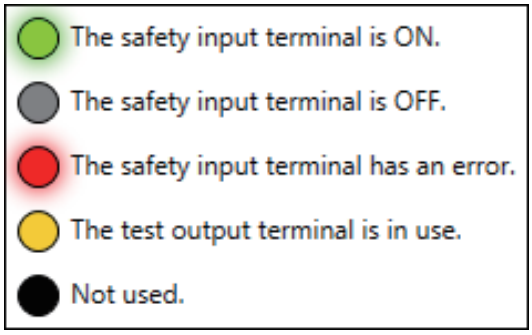
| Display | Monitor function | Description |
|---|---|---|
|  | Enabled | The Safety I/O Unit is present in the actual configuration, and communications are established with it.<br>This shows that there is no error at the safety I/O terminals of the Safety I/O Unit. |
|  | Enabled | The Safety I/O Unit is present in the actual configuration, and communications are established with it.<br>This shows that there is an error at the safety I/O terminals of the Safety I/O Unit. |
|  | Disabled | The Safety I/O Unit is present in the actual configuration, and safety process data communications are disabled for it.<br>This Safety I/O Unit is not subject to monitoring. |
|  | Disabled | The Safety I/O Unit is present in the actual configuration, and safety process data communications are not established with it.<br>This Safety I/O Unit is not subject to monitoring. |
|  | Disabled | The NX Unit mounting settings are disabled for the Safety I/O Unit.<br>This Safety I/O Unit is not subject to monitoring. |
|  | Disabled | The Safety I/O Unit is not present in the actual configuration or communications cannot be established with it.<br>This Safety I/O Unit is not subject to monitoring. |

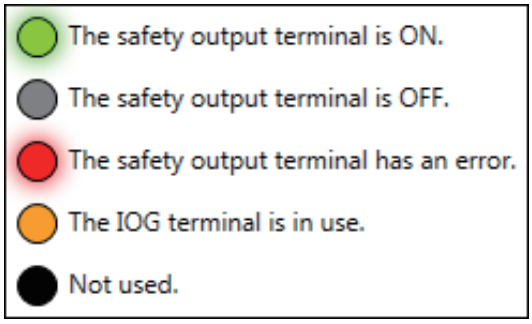
- External Device Displays

| Display | Description |
|---|---|
|  | In this state, safety I/O data from external devices is treated as ON.<br>This shows that there is no error at the safety I/O terminals connected to this external device. |
|  | In this state, safety I/O data from external devices is treated as OFF.<br>This shows that there is no error at the safety I/O terminals connected to this external device. |
|  | In this state, safety I/O data from external devices is treated as OFF.<br>This shows that there is an error at a safety I/O terminal connected to this external device. |

- Safety I/O Terminal Displays

Click the ? Icon below the safety input I/O terminals to display the legend for the safety I/O terminal displays.



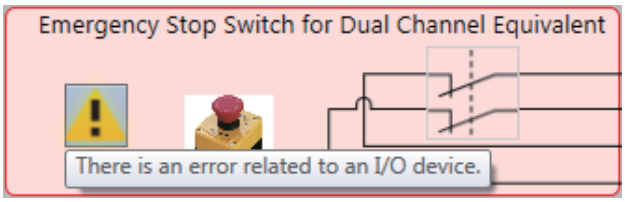

## 7-5-3 Troubleshooting Safety I/O Terminals

This section describes troubleshooting errors that may occur at a safety I/O terminal because of incorrect external device wiring or incorrect safety I/O settings.

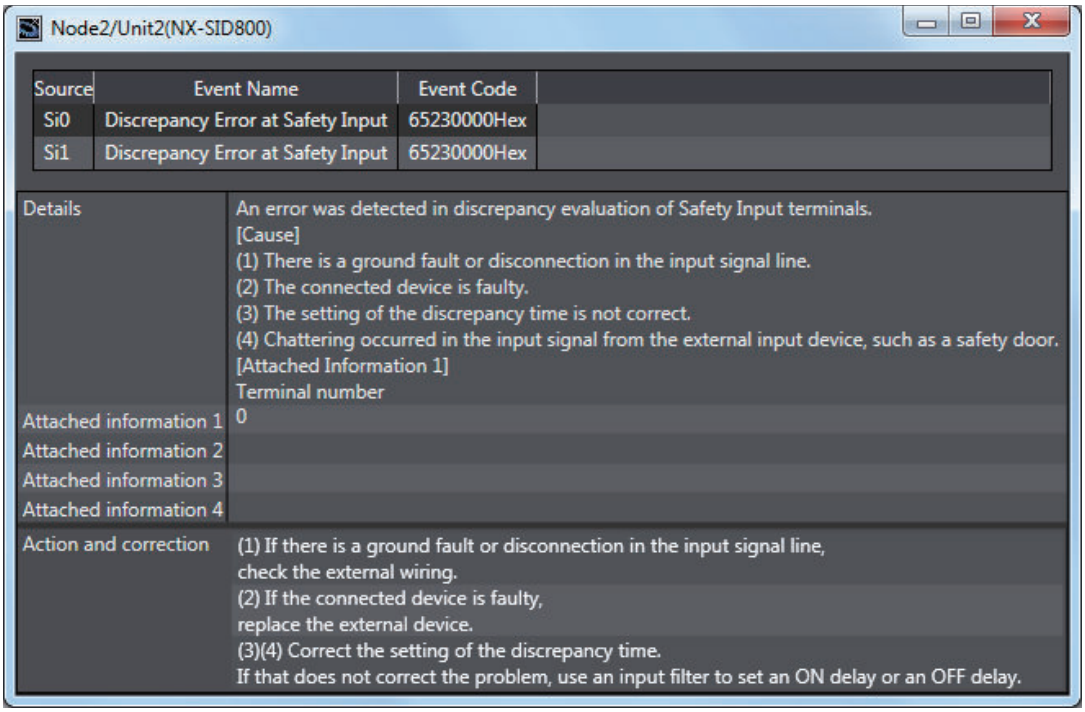
## Displaying Safety I/O Terminal Troubleshooting

**1** Open the Monitor Tab Page for the Safety I/O Unit for which there is an error in a safety I/O terminal.

**2** Click the " ! " Icon for the external device for which the error is displayed or right-click the external device and select **Troubleshooting...** from the menu.




**3** The Troubleshooting Dialog Box for the safety I/O terminals is displayed. Check details and corrections, and then eliminate the error.



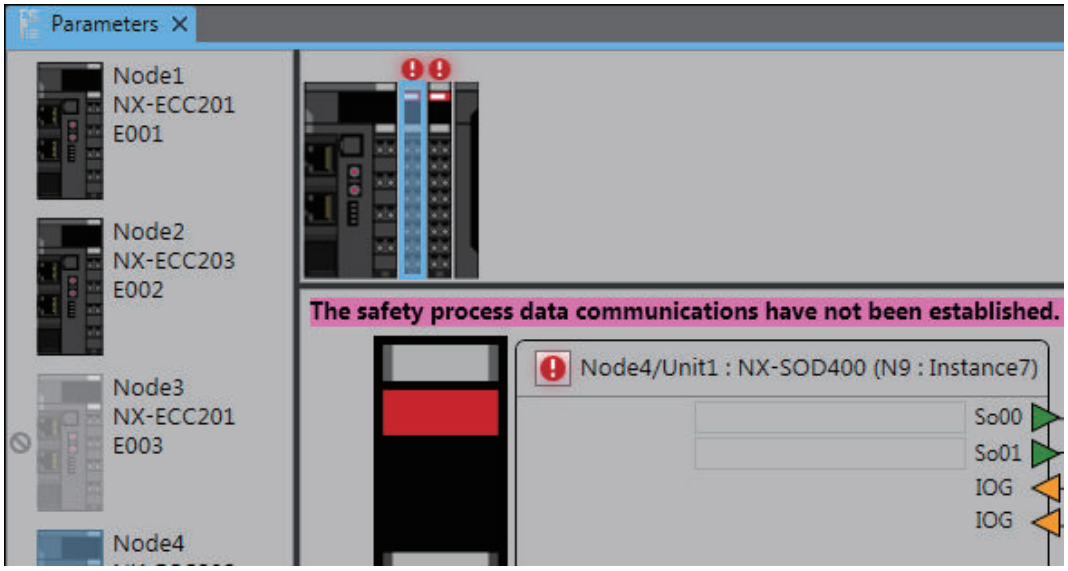

For details on the safety I/O terminal errors, refer to *12-3-4 Error Descriptions* on page 12-33.

## 7-5-4 Clear All Memory Operation for Safety I/O Units

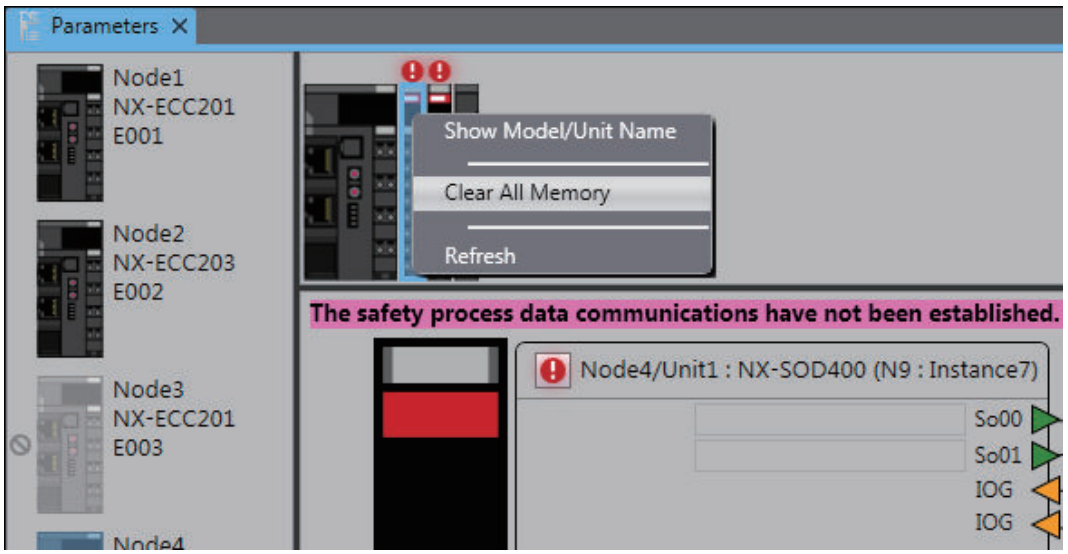
If an attempt to establish safety process data communications fails due to an illegal FSoE Slave Address, you must execute a Clear All Memory operation for the Safety I/O Units.

## Executing the Clear All Memory Operation for Safety I/O Units

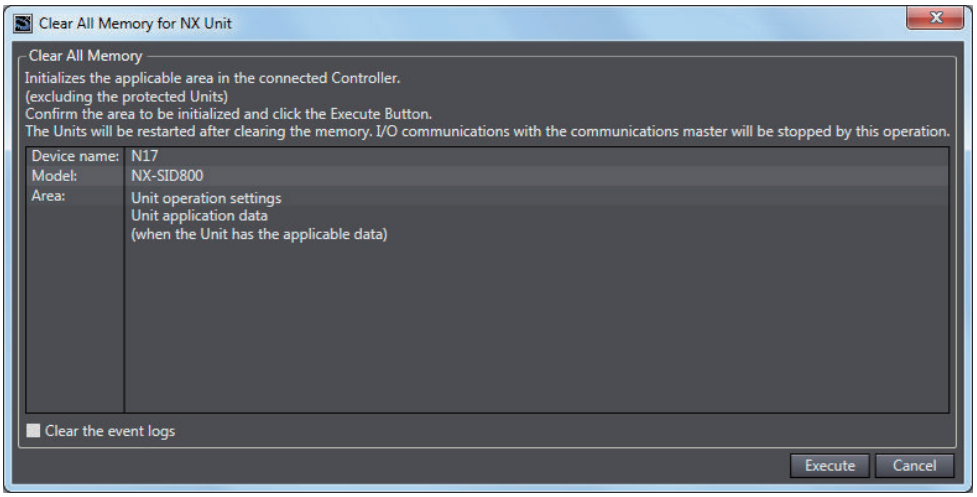
**1** Select the Safety I/O Unit in which the error has occurred from the NX Unit list.




**2** Right-click the Safety I/O Unit and select **Clear All Memory** from the menu.




**3** Click the **Execute** Button on the Clear All Memory Tab Page for the NX Unit.


Clear All Memory for NX Unit
Clear All Memory
Initializes the applicable area in the connected Controller.
(excluding the protected Units)
Confirm the area to be initialized and click the Execute Button.
The Units will be restarted after clearing the memory. I/O communications with the communications master will be stopped by this operation.
Device name:
N17
Model:
NX-SID800
Area:
Unit operation settings
Unit application data
(when the Unit has the applicable data)
Clear the event logs
Execute
Cancel

# 7-6 Functions for Checking Operation

This section describes the functions that you use on the Sysmac Studio to check the operation on the Safety CPU Unit.
You can check and adjust the operation of safety programs through an online connection between the Sysmac Studio and the Safety CPU Unit. This allows you to control BOOL variables, change present values, and perform other debugging tasks.

## 7-6-1 Overview of Functions for Checking Operation

This section describes the functions that you use on the Sysmac Studio to check the operation on the Safety CPU Unit.

| Functions for Checking Operation | Reference |
|---|---|
| Monitoring | *7-6-3 Monitoring Variables in the FBD Editor* on page 7-29<br>*7-6-4 Monitoring Variables in a Watch Tab Page* on page 7-30 |
| Monitoring in a Watch Tab Page | *7-6-5 Controlling BOOL Variables, Changing Present Values, and Using Forced Refreshing* on page 7-32 |
| Controlling BOOL variables | |
| Forced refreshing | |
| Changing present values of data | |
| Clear All Memory | *7-16-2 Clear All Memory Operation* on page 7-79 |
| Monitoring Controller status | *7-15 Monitoring Controller Status* on page 7-76 |
| Changing the operating mode | *7-3 Operating Modes of the Safety CPU Unit* on page 7-11 |
| Troubleshooting | *Section 12 Troubleshooting* on page 12-1 |
| Monitoring error information | |
| Displaying error logs | |

Procedures to check operation are performed when online to the Safety CPU Unit.

## 7-6-2 Starting and Stopping the Safety Programs in DEBUG Mode

**⚠ WARNING**

Before you start the system, perform user testing to make sure that all safety devices operate correctly.
Serious injury may possibly occur due to loss of required safety functions.

**⚠ WARNING**

Always confirm safety at the destination node before you transfer Unit configuration information, parameters, settings, or other data from tools such as the Sysmac Studio. The devices or machines may operate unexpectedly, regardless of the operating mode of the Controller.

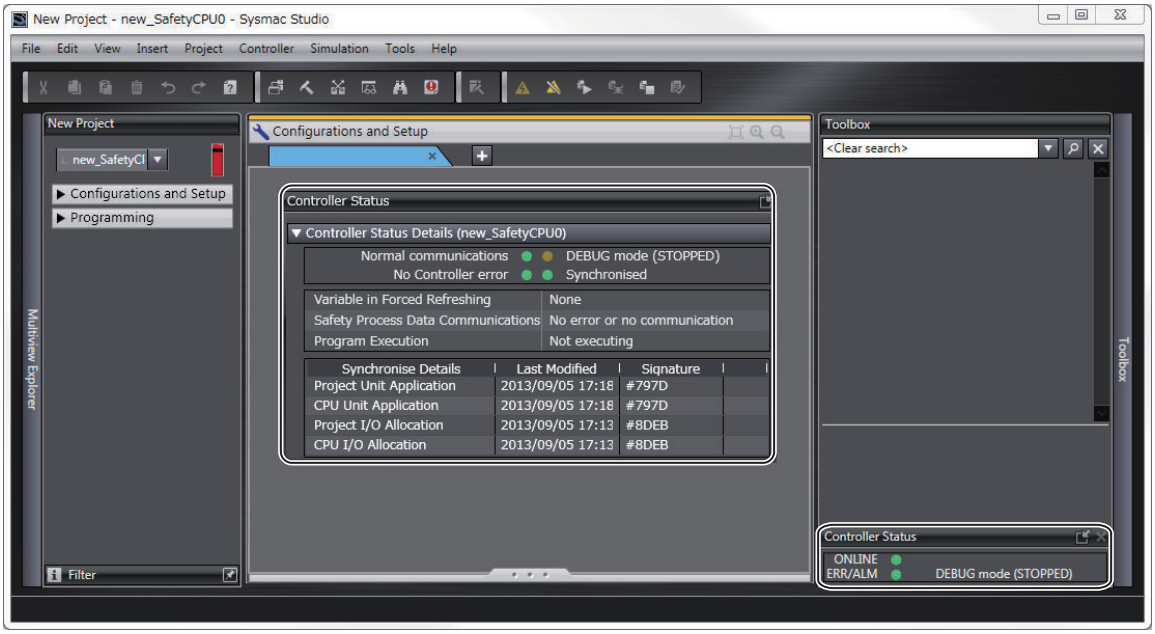
## Starting and Stopping the Safety Programs

Use the following procedure to change the Safety CPU Unit to DEBUG mode (RUN) or DEBUG mode (STOPPED).

With the Safety CPU Unit in DEBUG mode, perform one of the following operations.

- Select **Debug** – **Run** or **Stop** from the **Controller** Menu.
- Click the **Start Debugging** or **Stop Debugging** Button in the toolbar.

The Safety CPU Unit moves to DEBUG mode (RUN) or DEBUG mode (STOPPED).




## 7-6-3 Monitoring Variables in the FBD Editor

This section describes the procedures to monitor the present values of variables in the FBD editor to debug the safety programs.

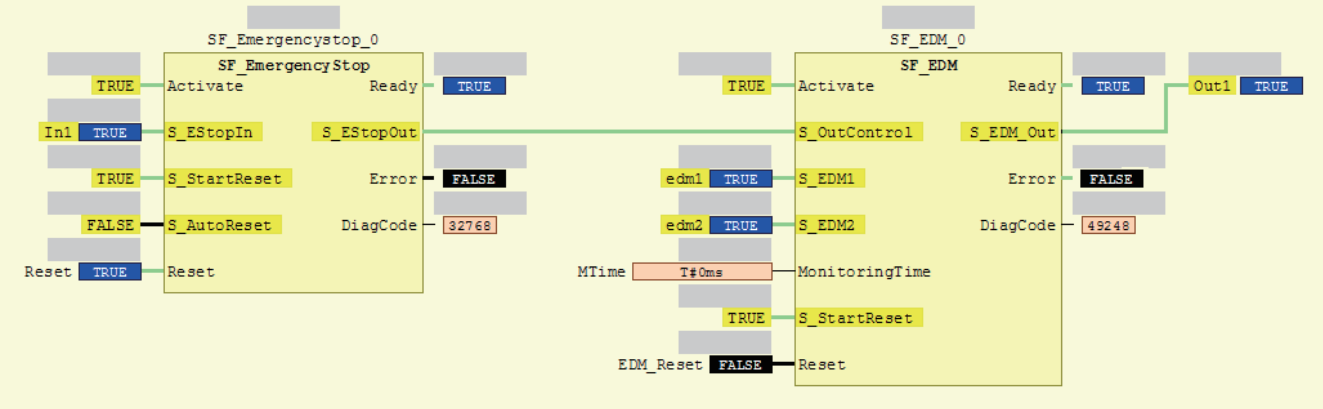
To be able to execute the monitoring function for variables on the FBD editor, the Sysmac Studio must be connected to the Safety CPU Unit that is either in the DEBUG mode or in the RUN mode. If the safety application monitoring is stopped, you need to start monitoring. For the procedure to start monitoring, refer to *7-12 Starting and Stopping the Safety Application Monitoring* on page 7-70.

## Executing the Operation Monitor for the Safety Programs

You can monitor the present values of variables in the FBD editor. Use the following procedure.

Double-click the program to monitor, in the Multiview Explorer.

The operating status of the selected POU is displayed in the FBD editor.




- The value of the variable is displayed in the frame on the right side of the variable name. “FALSE” is displayed with a black background, and “TRUE” is displayed with a blue background. Numeric values are displayed as decimal numbers. Use the Watch Tab Page to check numerical values as binary or hexadecimal numbers.
- The connecting lines between variables and FBs appear in green when the signal is ON. They appear in black when the signal is OFF.

## 7-6-4 Monitoring Variables in a Watch Tab Page

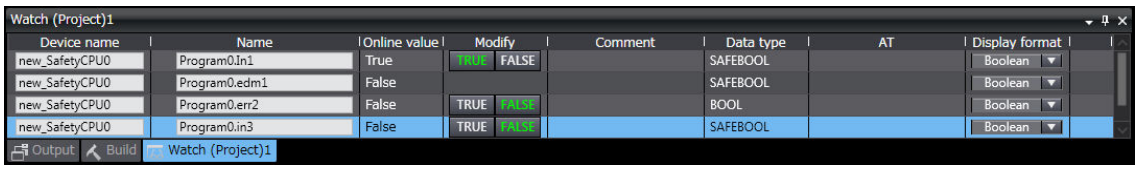
This section describes the procedures to monitor the present values of variables in a Watch Tab Page to debug the safety programs.
To be able to execute the monitoring function for variables on the Watch Tab Page, the Sysmac Studio must be connected to the Safety CPU Unit that is either in the DEBUG mode or RUN mode. If the safety application monitoring is stopped, you need to start monitoring. For the procedure to start monitoring, refer to *7-12 Starting and Stopping the Safety Application Monitoring* on page 7-70.

### Monitoring in a Watch Tab Page

You can check the present value of one or more variables in the Watch Tab Page.

### Displaying a Watch Tab Page

Select **Watch Tab Page** from the **View** Menu.
The Watch Tab Page is displayed.




To close a Watch Tab Page, click the Close Button for the tab page.
To display a Watch Tab Page that you closed, select **Watch Tab Page** from the **View** Menu again.

### Contents of the Watch Tab Page

The following table gives variable-related information displayed in a Watch Tab Page.

You can right-click an column and use the displayed menu to display or hide the following items: Comment, Data type, AT, and Display format.

YES: Editable, NO: Not editable

| Item | Description | Editing |
|---|---|---|
| Device name | The device name is displayed. | YES |
| Name | The variable name is displayed. | YES |
| Online value | The present value of the variable is displayed. | |
| Modify | The new value is displayed. | YES |
| Comment | The comment for the variable is displayed. | NO |
| Data type | The data type is displayed. | NO |
| AT | No information is displayed. | NO |
| Display format | The display format (decimal, hexadecimal, etc.) of the present value and modify value is displayed. | YES |

## Registering Variables in the Watch Tab Page

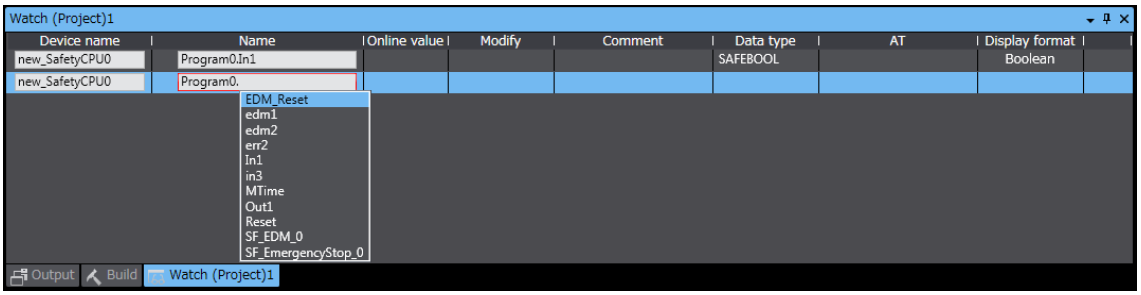
There are two ways to register variables.
Method 1: Enter the variable name in the name cell in the Watch Tab Page.
Method 2: Drag the variable to the Watch Tab Page from a variable table.

### ● Procedure for Method 1

**1** Click the cell that says *Input Name* at the bottom of the Watch Tab Page.

**2** Enter the variable name to display the present value.

**3** As you enter characters, a list of candidate variable names is displayed. Select the variable name from the list.
The variable name is registered.




### ● Procedure for Method 2

Drag a variable from a variable table to the Watch Tab Page.
The variable is registered.

## Deleting Variable Names from the Watch Tab Page

Right-click the variable name to delete in the Watch Tab Page and select **Delete** from the menu. Or, press the **Delete** Key to delete the variable name directly.
The variable name and the row it was displayed on are deleted.

# 7-6-5 Controlling BOOL Variables, Changing Present Values, and Using Forced Refreshing

You can debug the safety program by controlling BOOL variables (Set/Reset), changing present values, and executing forced refreshing from the Sysmac Studio. These functions can be executed only when the Safety CPU Unit is in DEBUG mode (RUN) or DEBUG mode (STOPPED).

**⚠ WARNING**

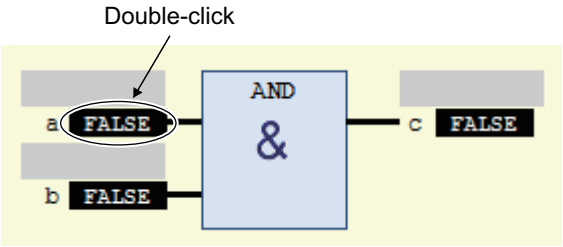
Make sure that the area around the system is safe before you control BOOL variables (Set/Reset), change present values, and execute forced refreshing.
The outputs may operate and may cause serious injury.

## Controlling BOOL Variables (Set/Reset)

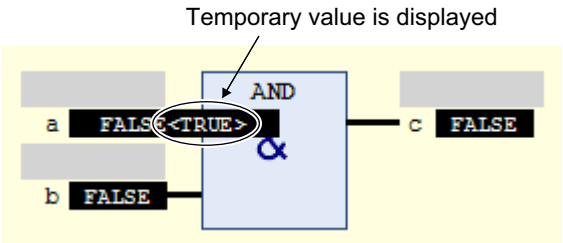
This function allows you to change the values of BOOL variables in the FBD editor or Watch Tab Page to debug safety programs.

### ● Controlling BOOL Variables in the FBD Editor (Set/Reset)

***1*** Double-click the present value of the BOOL variable to change.
Example: To set the variable *a*, double-click the present value of *FALSE*.

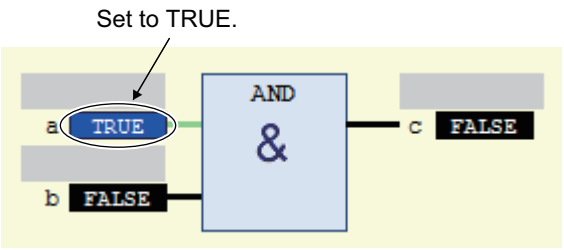



The value changes to a temporary status.
A temporary value appears in <> on the right side of the present value. This indicates that the temporary value, either TRUE or FALSE, is available to replace the present value.



Each double-click toggles the temporary value.

**2** Select **Write Values** from the **Controller** Menu.
The temporary value is removed and reflected as the present value.
Example: The present value of variable *a* is set to *TRUE*.

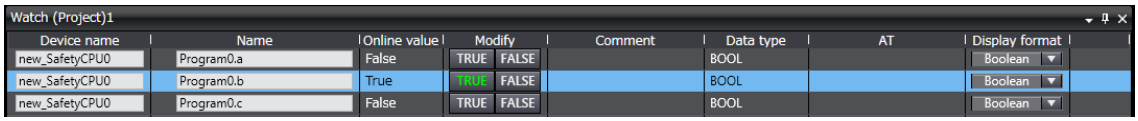



**Additional Information**

You can replace more than one present value in a single operation. To do this, set multiple BOOL variables with temporary values, and then select **Write Values** from the **Controller** Menu.

### ● Controlling BOOL Variables in the Watch Tab Page (Set/Reset)

Select **TRUE** in the **Modify** Column to change the variable to TRUE. Select **FALSE** in the **Modify** Column to change the variable to FALSE.
The present value is displayed in the Watch Tab Page as **TRUE** when set, and **FALSE** when reset.

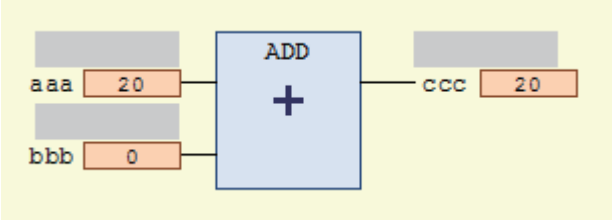



## Changing the Present Values of Variables

This function allows you to change the present value of non-BOOL variables to desired values in the FBD editor or Watch Tab Page to debug safety programs.

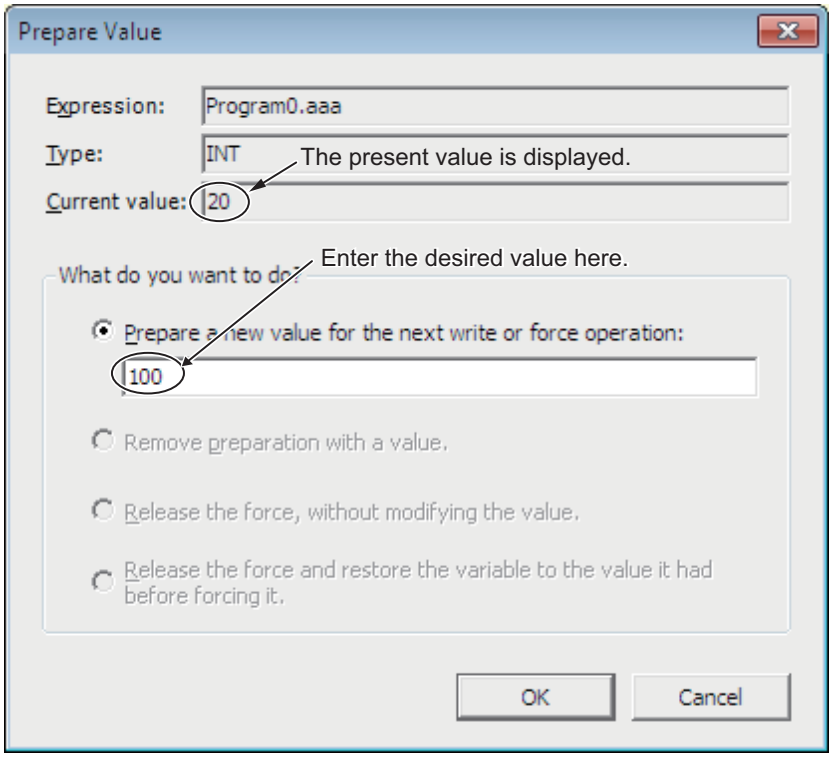
### ● Changing Present Values on the FBD Editor

**1** Double-click the present value of the non-BOOL variable to change.
Example: To change the present value of variable *aaa*, double-click the present value of *20*.

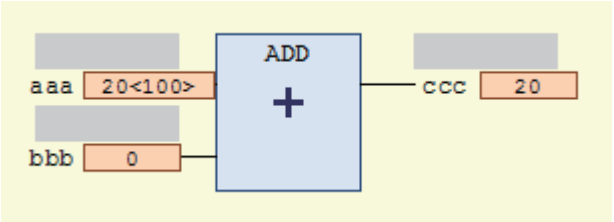



The **Prepare Value** Dialog Box is displayed.

**2** Select the **Prepare a new value for the next write or force operation** Option and enter the new value.
Example: This example changes the value to *100*.

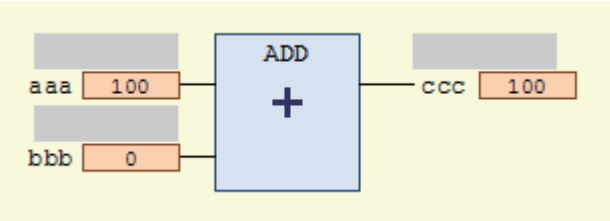



**3** Click the **OK** Button.
The **Prepare Value** Dialog Box closes and the new value is prepared as the temporary value. The temporary value appears in <> on the right side of the present value. This indicates that the temporary value is available to replace the present value.




To cancel the temporary value, double-click the present value of the desired variable again. Select the **Remove preparation with a value** Option in the **Prepare Value** Dialog Box, and then click the **OK** Button.

**4** Select **Write Values** from the **Controller** Menu.
The temporary value is removed and reflected as the present value.
Example: The present value of variable *aaa* is changed to *100*.




**Additional Information**

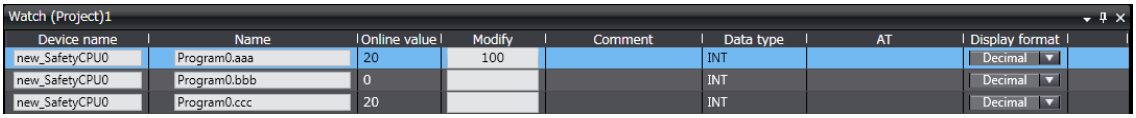
You can replace more than one present value in a single operation. To do this, set multiple present values with temporary values, and then select **Write Values** from the **Controller** Menu.

## ● Changing Present Values on a Watch Tab Page

Use the following procedure to change present values from the Watch Tab Page.

***1*** Select **Watch Tab Page** from the **View** Menu to display a Watch Tab Page.

***2*** Move the cursor to the cell in the **Modify** Column on the Watch Tab Page, enter a value that is compatible with the format that is given in the **Display format** Column, and then press the **Enter** Key. Press the **Esc** Key to cancel the entry.
The present value is changed.




The format for entering a value in the **Modify** Column depends on the **Display format** type that is given in the Data format Column.
Refer to *7-6-4 Monitoring Variables in a Watch Tab Page* on page 7-30 for details.
Press the **Esc** Key to cancel the entry.

- Examples of Entries in the **Modify** Column:

| Display format type | Example |
|---|---|
| Boolean | TRUE or FALSE |
| Decimal | 10, -100 |
| Real number | 123.4, 1.234e2, 1.234E2, -1.23e-3 |
| Hexadecimal | 1001, FFFF8000 |
| Binary | 11110000 |
| String | abc, ABC |

**Additional Information**

If you enter an illegal value in the **Modify** Column, it is detected as an error and the cell is highlighted in red.

## Forced Refreshing

Forced refreshing allows you to refresh external inputs and outputs with user-specified values from the Sysmac Studio to debug the system. You execute this in the FBD editor or Watch Tab Page.
Forced refreshing is executed for the specified variables.
The state that is specified with forced refreshing is retained until forced refreshing is cleared from the Sysmac Studio.
All forced refreshing is cleared when a fatal error occurs in the Safety CPU Unit, when a Clear All Memory operation is performed, when the operating mode is changed, when power is interrupted, or when the project is downloaded.
You can use forced refreshing for the following data types.

| | |
|---|---|
| Boolean | BOOL and SAFEBOOL |
| Bit string | BYTE and WORD |
| Integers | INT, SAFEINT, DINT, and SAFEDINT |
| Times of Day | TIME and SAFETIME |

**Precautions for Safe Use**

- With forced refreshing, the values of variables are overwritten with specified values and then the safety programs are executed.
  If forced refreshing is used for variables that give the results of program processing, the variables will first take the specified values, but they will then be overwritten by the safety program.
- Depending on the difference in the forced status, the control system may operate unexpectedly.

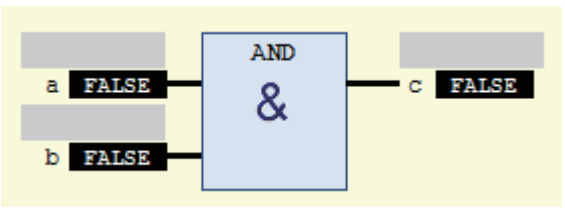
**Precautions for Correct Use**

- Forced status for forced refreshing is not removed when you change from DEBUG mode (STOPPED) to DEBUG mode (RUN).
- You can use forced refreshing for the following variables: device variables assigned to Safety I/O Units and user-defined variables.
- Even if you use forced refreshing for the device variables assigned to the input terminal to a Safety Input Unit, the forced value will not be applied to the variable that is assigned to the I/O port of an NJ/NX-series CPU Unit.

### ● Forced Refreshing of BOOL Variables in the FBD Editor

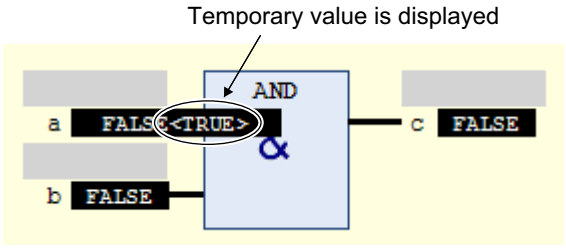
Use the following procedure to execute forced refreshing on BOOL variables.

***1*** Click the present value of the BOOL variable to change.
Example: To force-refresh variable *a*, click the present value of *FALSE*.



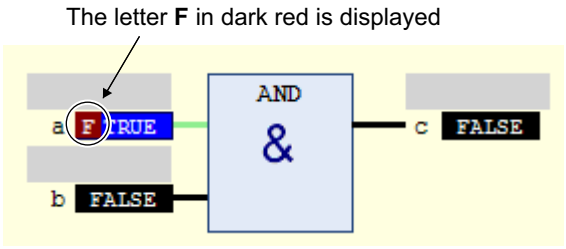

The value changes to a temporary status.

A temporary value appears in <> on the right side of the present value. This indicates that the temporary value is available for forced refreshing.




Each click toggles the temporary value.

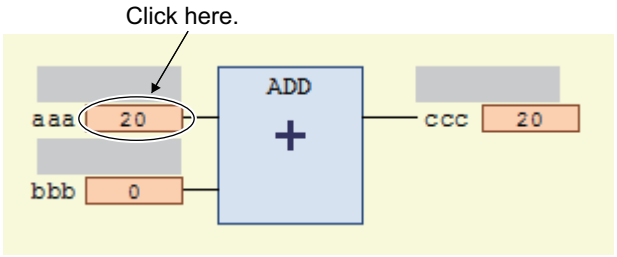
**2** Select **Force Values** from the **Controller** Menu.
Forced refreshing is performed with the temporary values.
This removes the temporary value and places the letter [F] in dark red on the left side of the variable.
Example: The value of variable *a* is forced-refreshed to *TRUE*.




## ● Forced Refreshing of Non-BOOL Variables in the FBD Editor

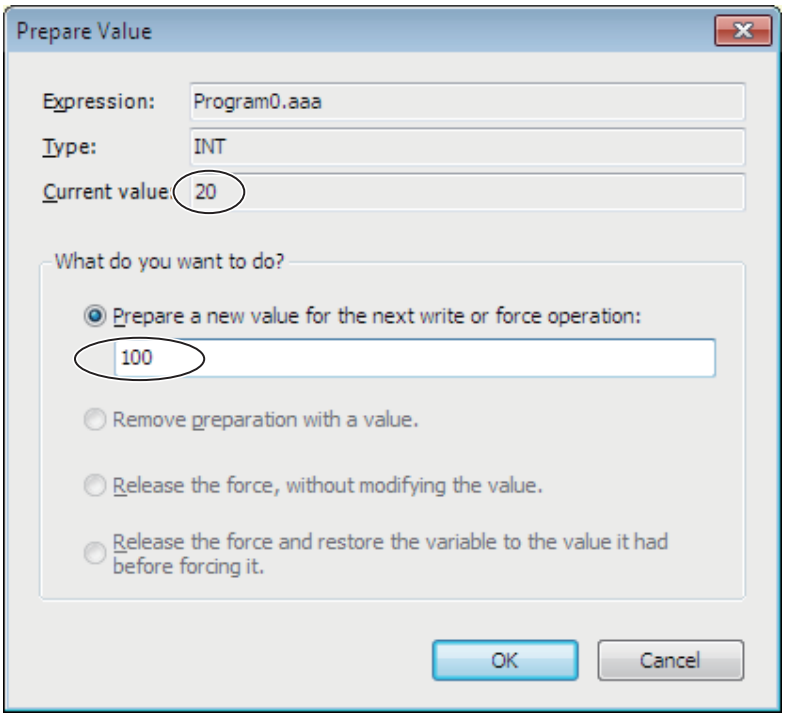
Use the following procedure to execute forced refreshing for non-BOOL variables.

**1** Click the present value for the non-BOOL variable to change.
Example: To force-refresh the present value of the variable *aaa*, click the present value *20*.




The **Prepare Value** Dialog Box is displayed.

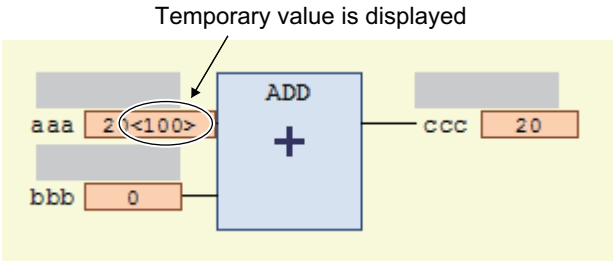
**2** Select the **Prepare a new value for the next write or force operation** Option and enter a new value.
Example: This example changes the value to *100*.




**3** Click the **OK** Button.

The **Prepare Value** Dialog Box closes and the value changes to a temporary value.

A temporary value appears in <> on the right side of the present value.



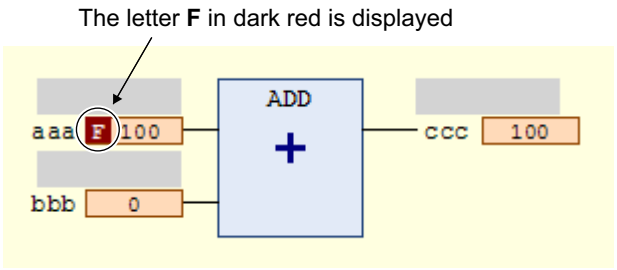

To cancel the temporary value, click the present value of the variable again. Select the **Remove preparation with a value** Option in the **Prepare Value** Dialog Box, and then click the **OK** Button.

**4** Select **Force Values** from the **Controller** Menu.

Forced refreshing is performed with the temporary values.

This removes the temporary value and places the letter [F] in dark red on the left side of the variable.

Example: The value of variable *aaa* is forced-refreshed to *100*.




**Additional Information**

The number of variables you can force-refresh simultaneously is listed below.
NX-SL3300/NX-SL3500: Up to 19 variables
NX-SL5500/NX-SL5700: Up to 256 variables

## ● Procedure to Cancel All Forced Refreshing from the FBD Editor

Use the following procedure to batch-clear forced refreshing.

Select **Unforce Values** from the **Controller** Menu.

All forced refreshing is cleared at once.

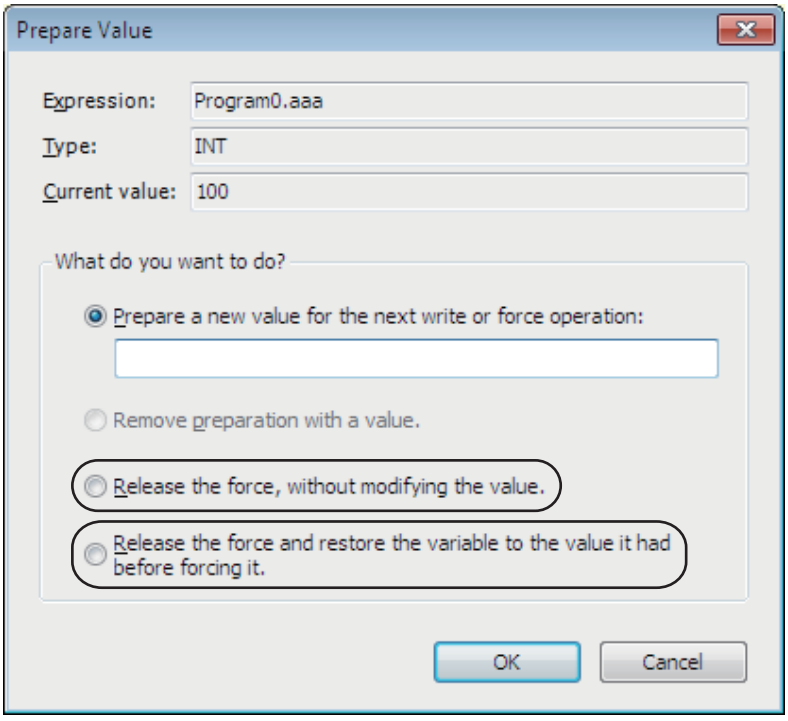
The letter [F] in dark red of all forced refreshing values are removed. The value will not change.

## ● Procedure to Cancel Individual Forced Refreshing from the FBD Editor

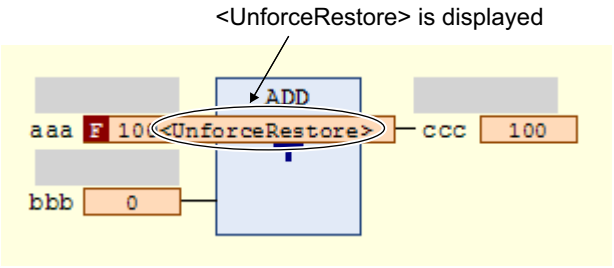
Use the following procedure to individually clear forced refreshing.

***1*** Click the present value of the variable to change.

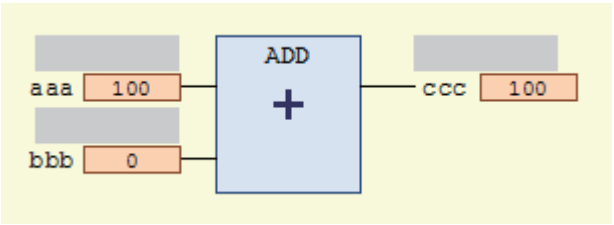
The following **Prepare Value** Dialog Box is displayed.



**2** To clear the forced refreshing value and restore the original value, select the **Release the force and restore the variable to the value it had before forcing it** Option, and then click the **OK** Button.
To clear forced refreshing without changing the present values, select the **Release the force, without modifying the value** Option, and then click the **OK** Button.




**3** Select **Force Values** from the **Controller** Menu.
The forced refreshing is cleared individually. The letter [F] in dark red is removed.




**Additional Information**

The number of variables for which you can clear forced refreshing simultaneously is listed below.
NX-SL3300/NX-SL3500: Up to 19 variables
NX-SL5500/NX-SL5700: Up to 256 variables

## 7-6-6 Cross References

### Cross References

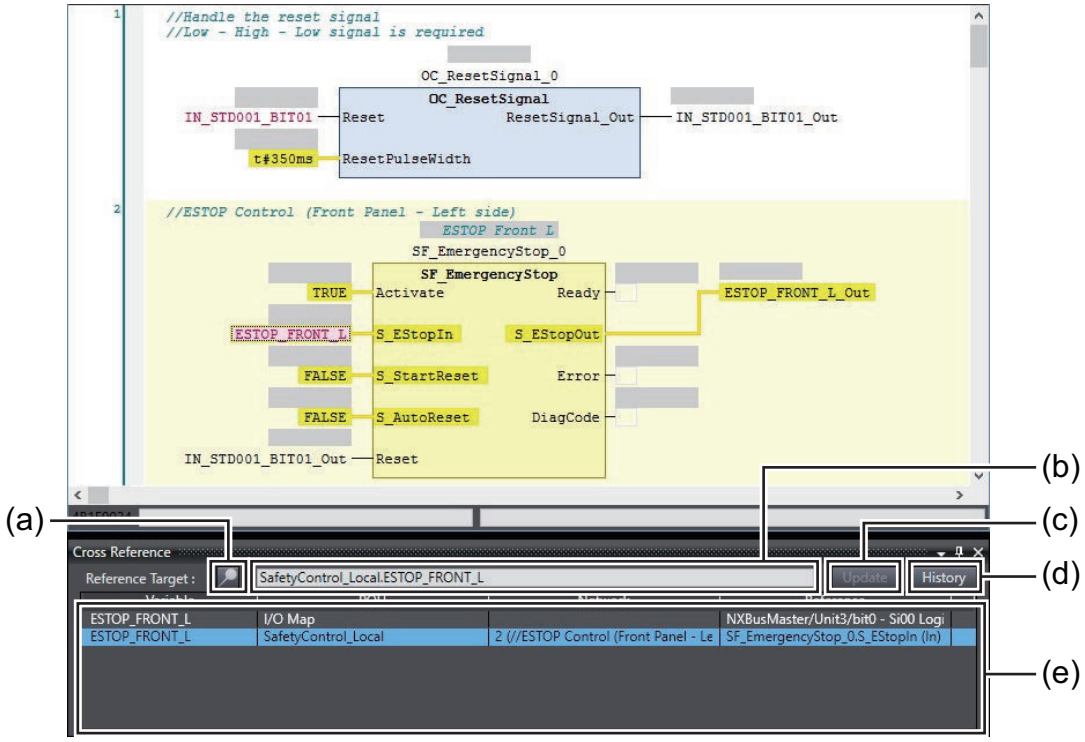
Cross references allow you to see the programs and locations where variables of the safety program are used. You can view all locations where an element is used from this list.

### Displaying and Manipulating Cross References

**1** Select **Cross Reference Tab Page** from the **View** Menu.
The Cross Reference Tab Page is displayed.

**2** Select a referenced element.
The name of the selected referenced element is displayed in the Reference Target field, and the locations where the element is used are listed under Cross Reference.

You can directly enter the variable name or member name in the Reference Target field.
Refer to *Referenced Element You Can Select* on page 7-41 for the elements you can select for reference.
For the areas where cross reference can be performed, refer to *Areas for Cross Reference* on page 7-42.




| | Item | Description |
|---|---|---|
| (a) | Lock Button | Click this button to lock the display of the referenced element and cross reference list. Click the button again to release the lock. |
| (b) | Reference Target field | The referenced element that is currently selected is displayed. Or, you can directly enter a variable name or variable member name as the reference target in this field.*1 |
| (c) | **Update** Button | This button is enabled only after the lock button is clicked and the display is locked. When the **Update** Button is clicked, the display of the cross reference list is updated. |
| (d) | **History** Button | When the History Button is clicked, the history window for the referenced element is displayed. On the reference history window, you can select the referenced element that was searched for in the past. |
| (e) | Cross reference list | The locations in which the referenced element is used are displayed. Refer to *Cross Reference List Details* on page 7-42, for details on items in the list. |

*1. Note that an error occurs if the following variables are entered. Also, an error occurs if the directly entered element is not a variable.
- A variable in a POU that is protected with the display prohibition setting

## Referenced Element You Can Select

The following table shows the referenced elements you can select and the locations from which you can select the referenced elements. You can select only the elements displayed with a focus in the following locations.

| Element | | Location |
|---|---|---|
| Variable | Global variable | • Global variable table<br>• I/O parameter of functions, I/O parameter of function blocks, or instance name of function blocks on the FBD editor<br>• I/O map |
| | Internal variable, external variable, input variable and output variable | • Internal variable tab, external variable tab and in-out variable tab in local variable tables<br>• I/O parameter of functions, I/O parameter of function blocks or instance name of function blocks on the FBD editor |

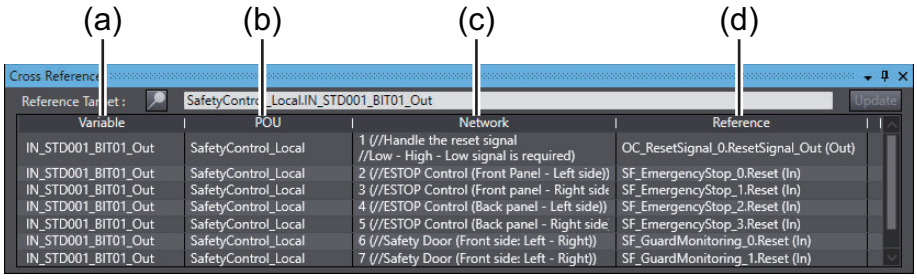
## Areas for Cross Reference

The following table shows the areas where the usage locations of referenced element are detected. The following areas are displayed in the cross reference list if the referenced element is used in the areas.

| Referenced element | Areas where usage locations are detected |
|---|---|
| Variable | • I/O parameter of functions, I/O parameter of function blocks, or instance name of function blocks on the FBD editor<br>• I/O Map |

## Cross Reference List Details

The following table describes the contents of items displayed in the cross reference list.



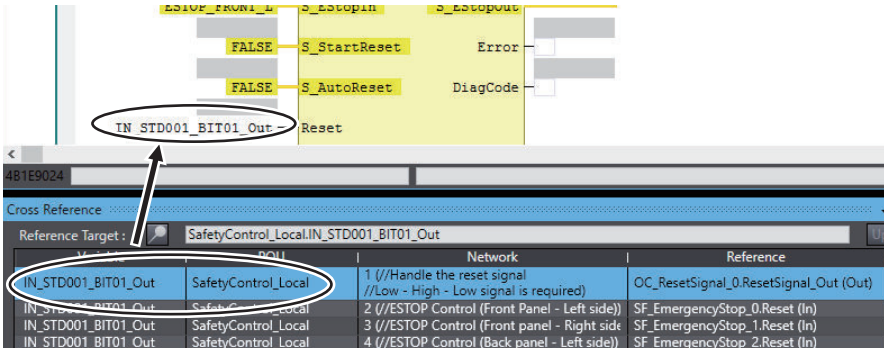

| | Item | Description |
|---|---|---|
| (a) | Variable | The name of the referenced variable is displayed. |
| (b) | POU | The name of POU where the variable is used is displayed. |
| (c) | Network | The network numbers and network comments of the usage locations are displayed. |
| (d) | Reference | The name of function or function block where the referenced variable is used is displayed. |

**Additional Information**

If a cross reference is for an input variable or output variable of a function block instance, the function block instance is also displayed in the cross reference list.

## Viewing Usage Locations from the Cross Reference List

You can view where elements are used from the cross reference list. Select the item you want to find references for from the list. The locations where that item is used appear in the Edit Pane.



# 7-7 Online Functional Test

This section explains how to perform Online Functional Test to check operation of safety functions of the safety system using the Sysmac Studio. Connect the Sysmac Studio and the Safety CPU Unit online, and operate the Safety Input Units and the reset switch to confirm that the output device is operating normally, and then output the test result.

The Sysmac Studio checks expected values based on variable values assigned to input devices and output devices. The operator should verify whether the actual devices are operating properly or not.

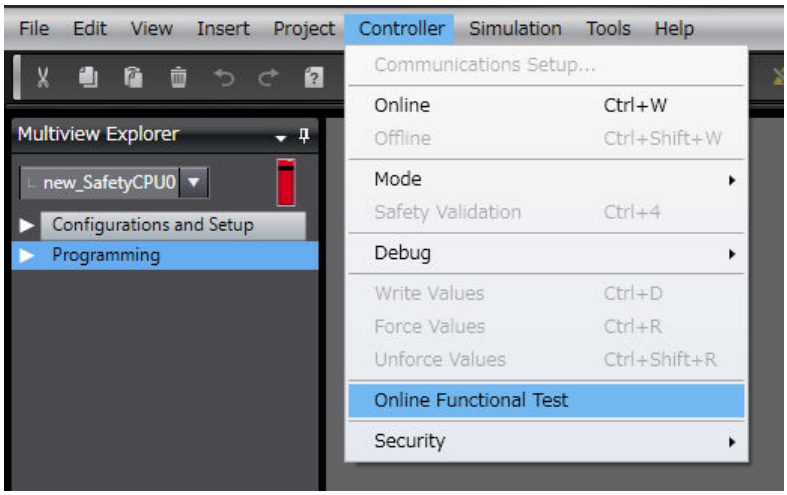
## 7-7-1 Online Functional Test Settings

Set the following signals and value in the Online Functional Test setting tab page to execute the Online Functional Test.

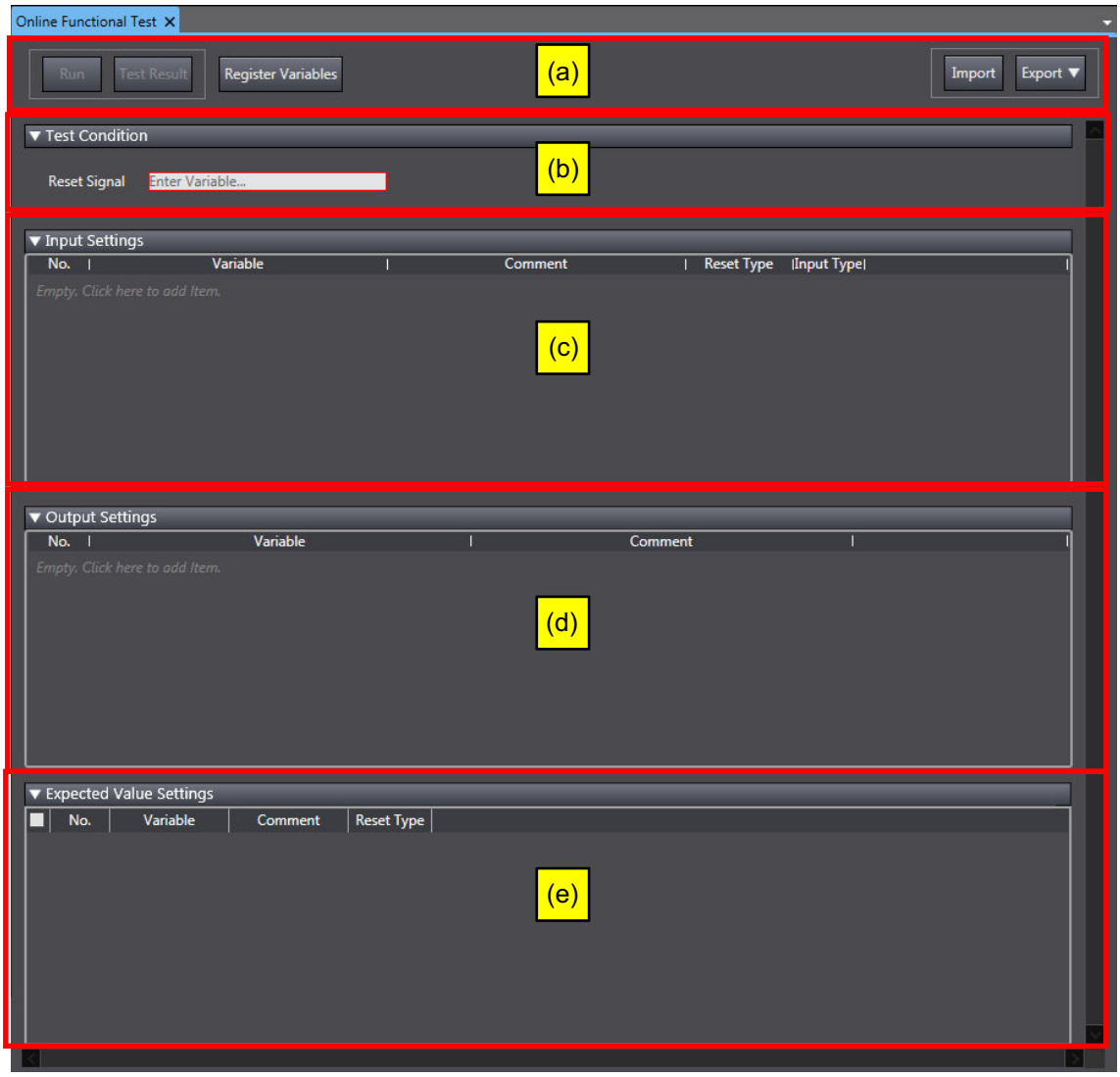
- Reset signal
- Input signal
- Output signal
- Expected value of the output signal corresponding to the input signal

You can display the settings window for the Online Functional Test in the following procedure.

- From the main menu, select **Controller** – **Online Functional Test**.




The following Online Functional Test setting tab page appears.




The configuration of the Online Functional Test setting tab page is as follows:

| Letter | Name | Function | |
|---|---|---|---|
| (a) | Operation panel | The operation panel is used to make settings for and execute the online functional test. | |
| | | Run Button | Starts an online functional test.<br>You can execute the test only when the controller is connected to the Safety CPU Unit online.<br>To run the online functional test, the Safety CPU Unit must be in RUN mode or DEBUG mode. |
| | | Test Result Button | Displays results of the previous online functional test. |
| | | Register Variables Button | Registers the variables that are used for the online functional test. |
| | | Import Button | Imports the settings for the online functional test from a CSV file.<br>If you import the settings, the current settings are overwritten. |

| Letter | Name | Function | |
|---|---|---|---|
| | | Export – CSV File Output | Exports the current online functional test settings to a CSV file. |
| | | Export – Export to Automatic Programming | Exports the current online functional test settings to the automatic programming settings. The settings of the export destination are overwritten. |
| | | Export – Export to Simple Automatic Test | Exports the current online functional test settings to the simple automatic test settings. The settings of the export destination are overwritten. |
| | | Export – Export to Online Functional Test | Exports the current automatic programming settings to the online functional test settings. The settings of the export destination are overwritten. |
| (b) | Test Condition | Specify a variable assigned to the reset switch of the safety system. You can set any SAFEBOOL or BOOL variable that is defined in the global variable table.<br>You will use the reset switch when you check the manual reset operation. | |
| (c) | Input Settings | Specify a variable assigned to the safety input device used in the online functional test.<br>In addition to the variable name of the input device, specify a reset type and an input type as well. | |
| | | Variable | Specify a variable name assigned to the safety input device of the safety system.<br>You can specify the SAFEBOOL or BOOL-type variable for the variable name. |
| | | Reset Type | Selects a reset type of the safety input device from the box.<br>The default value of the reset type is Manual.<br>• Manual<br>If manual reset is used for the input device, a test is executed for the manual reset scenario that uses the reset signal set in the Test Condition.<br>• Auto<br>If automatic reset is specified for the input device, a test is executed according to the auto reset scenario. |
| | | Input Type | To set a test for two input signals, such as for a safety door or two-hand switches, set the input type to **2 Inputs**.<br>If you set the input type to **2 inputs**, a row is added to specify another variable. |
| (d) | Output Settings | Specify a variable assigned to the safety output device used for the online functional test.<br>You can set any SAFEBOOL or BOOL variable that is defined in the global variable table. | |

| Letter | Name | Function |
|---|---|---|
| (e) | Expected Value Settings | Displays the matrix of the variables specified in the input setting and output setting. Sets the values subject to test and the expected value of the output variables for each input variable.<br>The expected values that you can set for output variables have the following meanings:<br>• 0: If the input variable changes to FALSE, the output variable changes to FALSE.<br>• 1: If the input variable changes to FALSE, the output variable changes to TRUE.<br><br>You can edit the expected value settings by importing or exporting the values, and copying and pasting the values with a spreadsheet program or any other application software. |

## Setting Example

This section provides an example of the Online Functional Test settings based on the application example given in *A-3-2 Safety Doors* on page A-32.

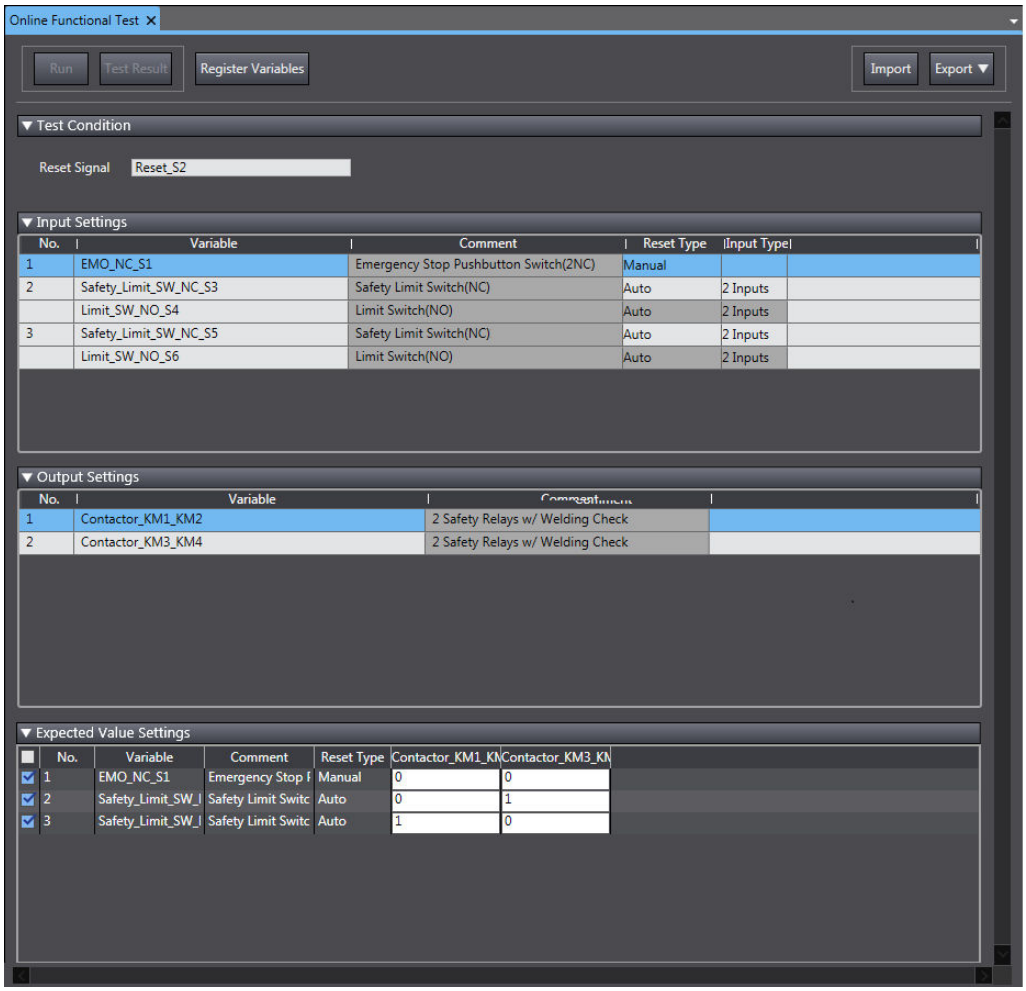
Application Overview from Safety Doors

| Safety category/PL | Safety device | Stop category | Reset |
|---|---|---|---|
| Equivalent to 4/PLe (Safety Doors) | Safety Limit Switches 1, 2 | 0 | Auto |
| | Emergency Stop Switch | 0 | Manual |

M1 stops when safety door 1 (S3, S4) is opened.
M2 stops when safety door 2 (S5, S6) is opened.
Both M1 and M2 stop when the emergency stop pushbutton S1 is pressed.

At this time, enter the following safety door variables in the setting items on the Online Functional Test setting tab page.
- Variable that is assigned to the reset switch
- Variables assigned to the safety limit switch and the limit switch
- Variables assigned to the safety relays

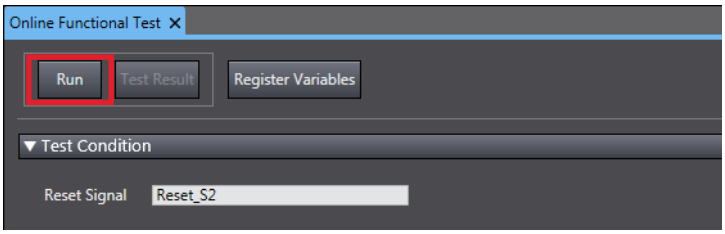
The settings for the above application are shown in the setting areas for the Online Functional Test.



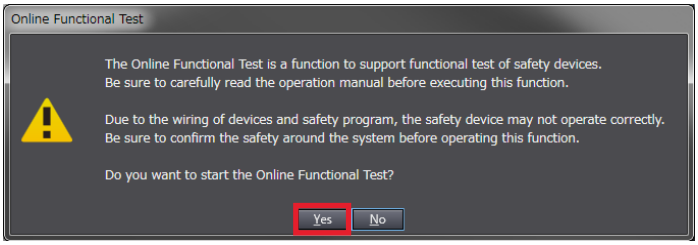

## 7-7-2 Online Functional Test Execution Procedure

Use the following procedure to execute the Online Functional Test.

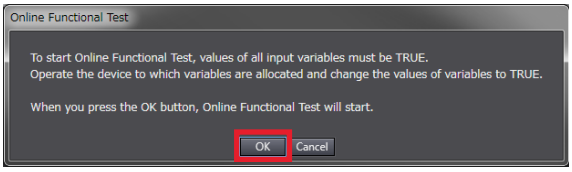
1. Place the Sysmac Studio online with the Safety CPU Unit.
2. Place the Safety CPU Unit in DEBUG mode.
   Refer to *7-4 Changing to DEBUG Mode* on page 7-17 for a detailed procedure.
3. Select **Controller** - **Online Functional Test**.
   The Online Functional Test Tab Page appears.
4. Set the **Test Condition**, **Input Settings**, **Output Settings**, and **Expected Value Settings**.
5. Click the **Run** Button on the control panel for the Online Functional Test Tab Page.



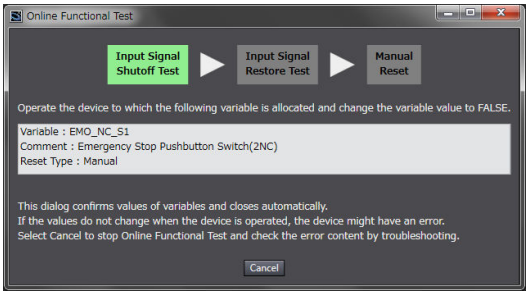

**6** Check the precautions for executing the Online Functional Test and select **Yes**.



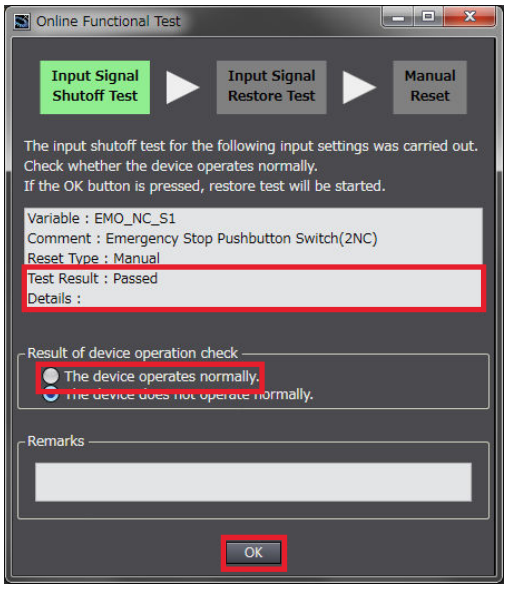

**7** Operate the device as instructed on the screen. When you complete the preparations for starting a test, select **OK**.




**8** Operate the device as instructed on the screen. Execute a shutoff test.
The operating instructions for the device to be tested will appear. Operate the input device as instructed on the screen and specify FALSE for the variable assigned to the input device.

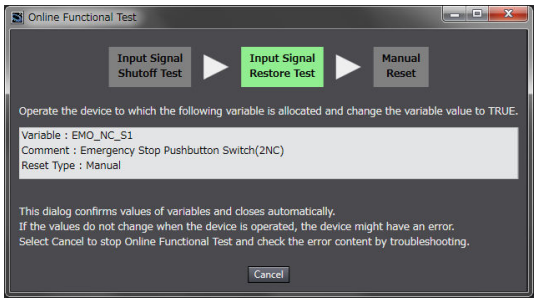



**9** Make sure that the device operated normally.
The Controller detects the values of the assigned variables whose statement became FALSE and defines whether the statement of the variables assigned to all the output devices is consistent with the settings of the expected values. The definition results are displayed in the below dialog box. If the device is running in accordance with the settings of the expected values, check the box **The device operates normally** and select **OK**.




**10** Operate the device as instructed on the screen. Execute a restore test.

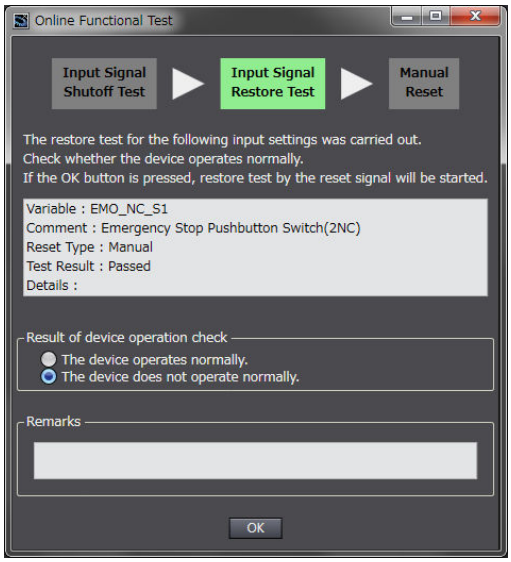
The operating instructions for the device to be tested will appear. Operate the device as instructed on the screen and specify TRUE for the variable assigned to the input device.



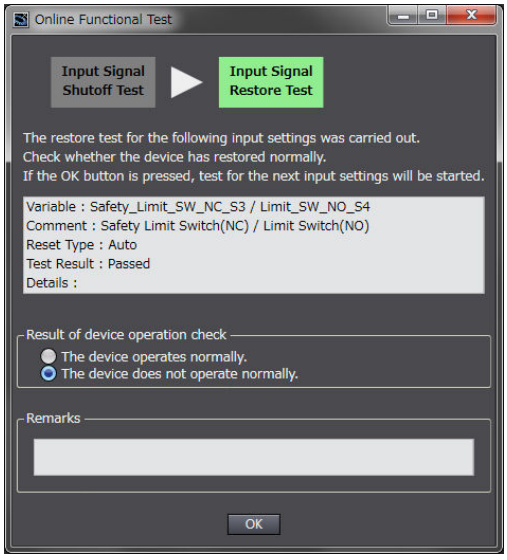

**11** Make sure that the device was restored normally.

- Reset Type: Manual

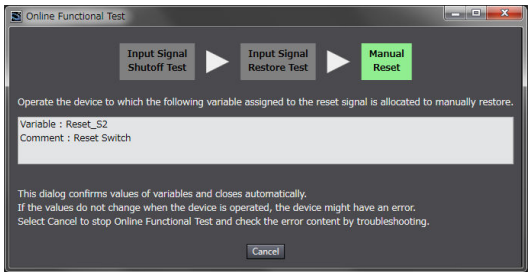
  The Controller checks if no change is made to the variables assigned to all the output devices when detecting that the specified variable is set to TRUE. The test result appears in the dialog shown below. If the devices operate normally as specified in the expected value settings, select the checkbox stating **The device operates normally** and then select **OK**.




- Reset Type: Auto
  The Controller checks if the variables assigned to all the output devices are set to TRUE when detecting that the specified variables are set to TRUE. The test result appears in the dialog shown below. If the devices operate normally as specified in the expected value settings, select the checkbox stating **The device operates normally** and then select **OK**.



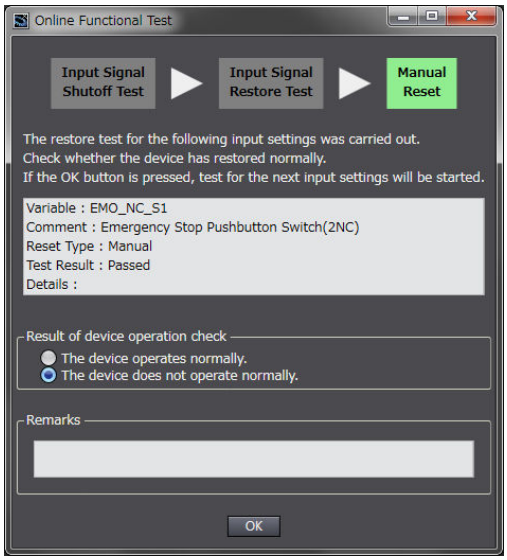

**12** Operate the device as instructed on the screen. Execute a function reset.
It is displayed for manual reset only.
The window shows instructions to reset the function, which will allow you to confirm that the tested input device is normally restored. Operate the reset switch as instructed on the screen and change the variable assigned to the reset switch from FALSE to TRUE and then FALSE again.




***13*** Check the operation of function reset.

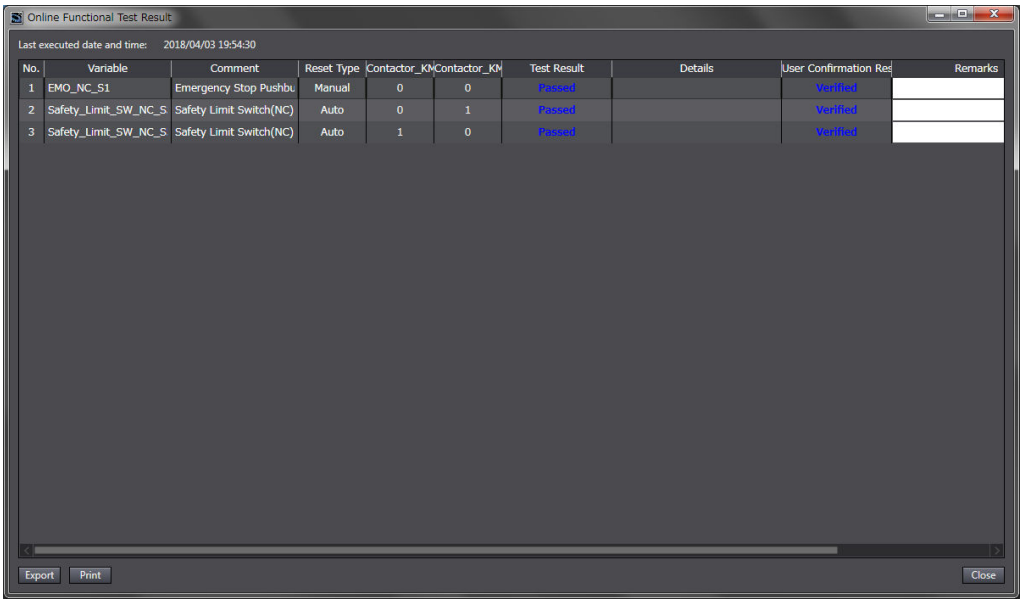
It is displayed only for Manual Reset.

The Controller checks if the variables assigned to all the output devices are set to TRUE when detecting that the variable assigned to the reset switch changed from FALSE to TRUE, and then FALSE again. The test result appears in the dialog shown below. If the devices operate normally as specified in the expected value settings, select the checkbox stating **The device operates normally** and then select **OK**.




***14*** Repeat the steps 8 through 13 for all the input devices.

Once the operations for all the input devices are completed, the Online Functional Test Result Page shown below appears.




***15*** Export or print the test results if a CSV file if needed.

### Precautions for Correct Use

The Online Functional Test allows you to perform basic operation check by executing manual reset or auto reset. The test result may not be accurate for complex conditions or special cases. Perform advanced operation check separately if needed.

### Additional Information

If you print the results of the Online Functional Test, when the safety validation of the safety programs is in process, a safety signature is printed in the lower-right of each page.

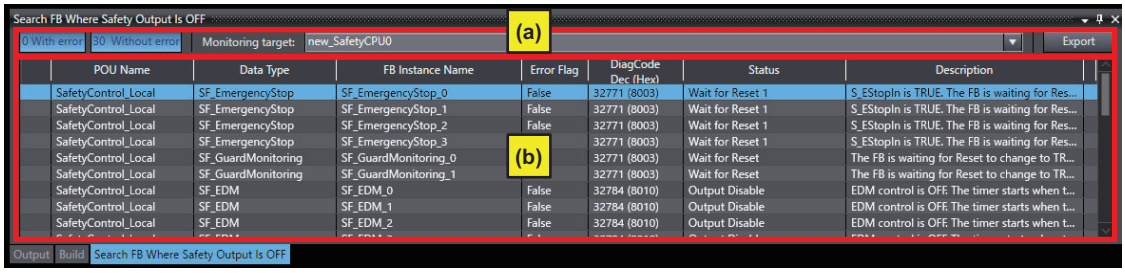
# 7-8 Search FB Where Safety Output Is OFF

This section describes the function for searching for a function block instance that causes safety output to be turned off in the safety program currently being executed by using Sysmac Studio.
Using this function, you can identify a function block instance where output has been turned off after an input signal had been disrupted or an error had occurred, without analyzing a safety program.

## 7-8-1 Procedure for Operating Search FB Where Safety Output Is OFF

**1** Start Sysmac Studio. Monitor safety application.
For the procedure to start monitoring, refer to *7-12 Starting and Stopping the Safety Application Monitoring* on page 7-70.

**2** From the Controller Menu, select **Controller** – **Search FB Where Safety Output Is OFF** – **Start**.
The Search FB Where Safety Output Is OFF window is displayed.




The Search FB Where Safety Output Is OFF window is configured as described in the following table:

| Letter | Name | Function | |
|---|---|---|---|
| (a) | Operation panel | This is an operation panel for performing setup or export for Search FB Where Safety Output Is OFF. | |
| | | Total number with error | Displays the total number of function block instances where safety output is turned off due to an error during execution of the relevant function block.<br>You can show or hide this by clicking it. |
| | | Total number without error | Displays the total number of function block instances where safety output is turned off due to a cause other than a function block error, such as input signal disruption.<br>You can show or hide this by clicking it. |
| | | Monitor Target | Select the Safety CPU Unit to monitor.<br>To search for a function block where safety output is off, it is necessary that the Safety CPU Unit be in the RUN or DEBUG mode. |
| | | Export | Exports the current search results to a CSV file. |

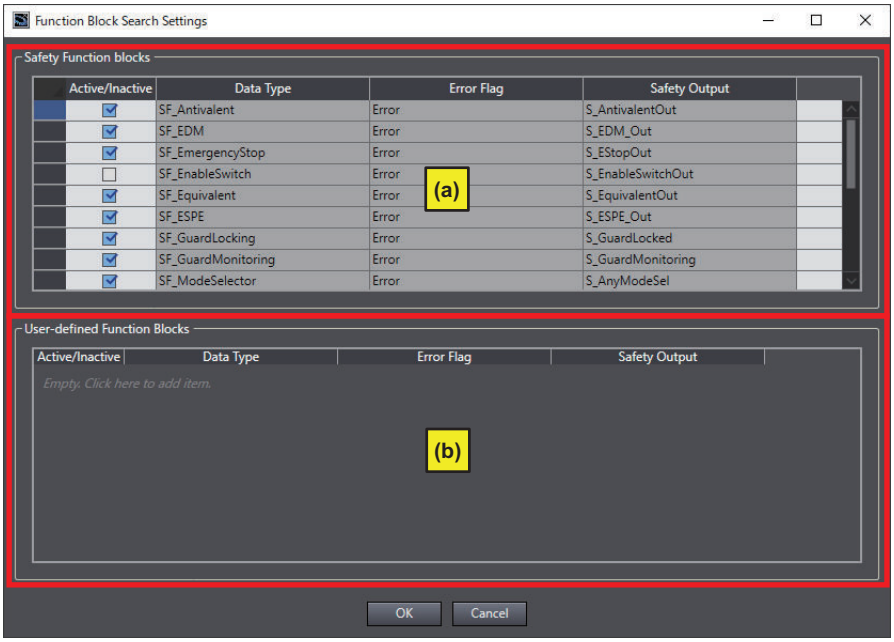
| Letter | Name | Function | |
|---|---|---|---|
| (b) | Search results | This section lists function block instances where safety output is turned off in the safety program currently being executed. | |
| | | POU Name | Displays the name of the POU where relevant function block instance is being executed. |
| | | Data Type | Displays the name of the relevant function block. |
| | | FB Instance Name | Displays the name of the relevant function block instance. |
| | | Error Flag | Displays the present value of the output variable set as an error flag on the Function Block Search Settings window. |
| | | DiagCode Dec (Hex) | Displays the present value of the *DiagCode* (diagnostic code), which is an output variable of a safety function block, as decimal and hexadecimal numbers. |
| | | Status | Displays the name of the status described in the Instructions Reference based on the *DiagCode* (diagnostic code). |
| | | Description | Displays the description of the status described in the Instructions Reference based on the *DiagCode* (diagnostic code). |

You can jump to the point of execution of a function block instance by double-clicking a row in the search results list.
To display the Help menu for function blocks, select a row in the search results list and press the **F1** key, or right-click a row and select **Display Help**.

## 7-8-2 Editing Function Blocks to be Searched for

By default, this function searches for safety function blocks. Use the following procedure to exclude function blocks to be searched for or add user-defined function blocks.

From the Controller Menu, select **Controller** – **Search FB Where Safety Output Is OFF** – **Set**.
The Function Block Search Settings window is displayed.




The Function Block Search Settings window is configured as described in the following table:

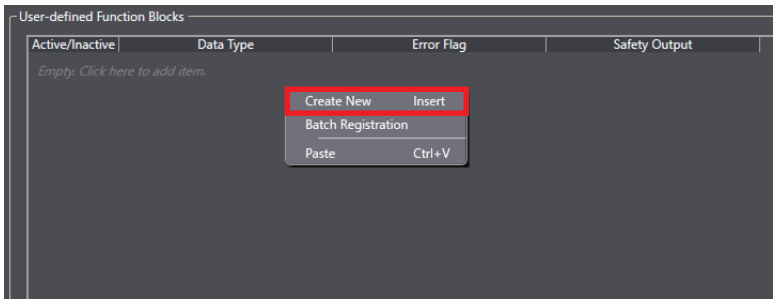
| Letter | Name | Description | |
|---|---|---|---|
| (a) | Safety Function block search settings table | Displays search settings for safety function blocks.<br>Search settings are predefined for available safety function blocks by default. | |
| | | Active/Inactive | Select the check box to include the function block as a search target, or clear the selection of the check box to exclude it. |
| | | Data Type | Displays the name of the data type of the target function block. |
| | | Error Flag | Displays the name of the output variable used as an error judgment condition. |
| | | Safety Output | Displays the name of the output variable used as a condition for judging if safety output is off. |
| (b) | User-defined Function Block search settings table | Displays search settings for user-defined safety function blocks.<br>Set user-defined function blocks to be searched for on Search FB Where Safety Output Is OFF. | |
| | | Active/Inactive | Select the check box to include the function block as a search target, or clear the selection of the check box to exclude it. |
| | | Data Type | Set the name of the data type of the target function block. |
| | | Error Flag | Set the name of a BOOL or SAFEBOOL output variable used as an error judgment condition. |
| | | Safety Output | Set the name of the SAFEBOOL output variable used as a condition for judging if safety output is off. |

## 7-8-3 Registering a Data Type in the Function Block Search Settings Window

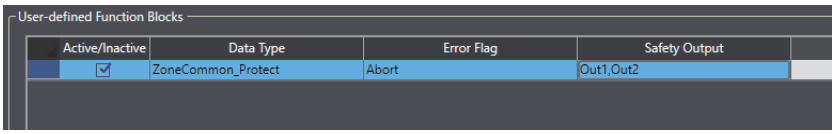
This section describes the procedure for registering a function block to be searched for on the Function Block Search Settings window.

***1*** From the Controller Menu, select **Controller** – **Search FB Where Safety Output Is OFF** – **Set**.

***2*** Press the **Insert** Key in the search setting table in User-defined Function Blocks, or right-click in the table and select **Create New** from the menu.




***3*** Enter necessary items in the added row, and then press the **Enter** key.


User-defined Function Blocks
Active/Inactive
Data Type
Error Flag
Safety Output
ZoneCommon_Protect
Abort
Out1,Out2

# 7-9 Node Name

This section describes the node name setting for the Safety CPU Unit.

## ● Node Name Application

The node name is a unique name that you assign to each Safety CPU Unit within the project. This helps you recognize the correct Safety CPU Unit when you begin online operations.
Check the node name that is displayed before you begin operation to prevent you from controlling the wrong Safety CPU Unit.

The node name that you set is stored in the Safety CPU Unit.
The node name that you set is displayed in the confirmation dialog box when you begin online operations.

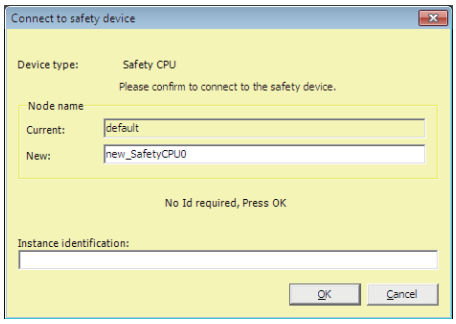
## ● Characters Allowed for Node Names

The following characters can be used for node names.
The name must have 79 or less printable ASCII characters.
The default node name for all Safety CPU Units is *default*.

## ● Setting the Node Name

You set the node name in the Connect to safety device Dialog Box, which is displayed when you go online with the Safety CPU Unit. The Connect to safety device Dialog Box is displayed when you perform one of the following operations.

- Connecting the Safety CPU Unit online for the first time with the factory default settings.
- Connecting the Safety CPU Unit online for a new project file after performing online operations for other projects.




**Note** The factory-default node name is displayed.

Check to see if the destination node is correct. Type a node name and click the **OK** Button.
The node name that you set is stored in the Safety CPU Unit. After this point of time, the new node name is displayed in the confirmation dialog box, which appears before you start online operations on the Safety CPU Unit.

### Precautions for Correct Use

Set a unique node name for the Safety CPU Unit.

# 7-10 Security Settings

You can use the Sysmac Studio to restrict operations and to protect POUs (programs and function blocks) in order to prevent unauthorized access to safety functions and to protect assets.

An overview of the applications and functions of security settings is given below.

| Function | Application | Outline of function |
|---|---|---|
| Safety Password | To prevent unauthorized access to safety functions | You can set a password for the Safety CPU Unit in order to prevent unauthorized operations, including changing the operating mode and performing the Clear All Memory operation. |
| Data Protection | To protect assets | You can set passwords for POUs (programs and function blocks), so that they cannot be individually displayed or changed. |

## 7-10-1 Setting the Safety Password

This section describes the safety password setting for the Safety CPU Unit.

### ● Safety Password Application

The safety password prevents unauthorized access to the safety functions of the Safety CPU Unit. When a safety password is set, the user is required to enter the password before performing an operation that affects the safety functions.

After you enter the safety password, it is retained in the Sysmac Studio. You do not need to enter it again until you take the Safety CPU Unit offline or close the project.

The safety password protects the following online operations on the Safety CPU Unit.

- Changing the operating mode (This does not apply when changing between DEBUG mode (STOPPED) and DEBUG mode (RUN).)
- Downloading (transferring data from the Computer to the Controller)
- Uploading (transferring data from the Controller to the Computer)
- Changing the safety password
- Clear All Memory operation*1
- Performing safety validation

*1. The password must be entered each time for this operation.

The safety password is empty by default.

You can set a safety password before or after you perform safety validation.

**Precautions for Correct Use**

- For security purposes, we recommend that you set a safety password for the Safety CPU Unit.
- If you lose the password set to the Safety CPU Unit, you will no longer be able to make changes to the Safety CPU Unit. Take caution not to lose the password. If the password is lost and needs to be reset, contact your OMRON representative.

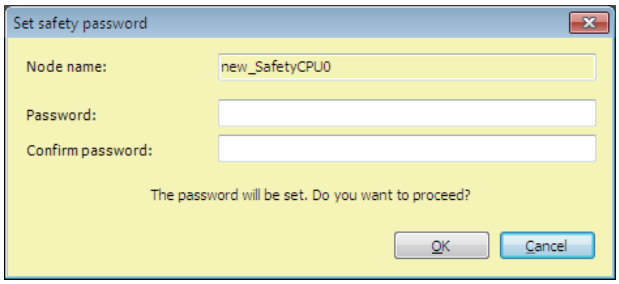
### ● Characters Allowed for Passwords

The following characters can be used for the password.

| Item | Description |
|---|---|
| Number of Characters | 32 characters max. |
| Applicable Characters | Single-byte alphanumeric characters (case sensitive) |

### ● Setting a New Safety Password

**1** Go online with the Safety CPU Unit and then select **Security** − **Set Safety Password** from the **Controller** Menu on the Safety CPU Unit Setup and Programming View.
The **Set safety password** Dialog Box is displayed.




**2** Enter the safety password in the **Password** Box. Enter the same password in the Confirm password Box, and click the **OK** Button.
The password is set.

**Additional Information**

We recommend that you set text strings that contain both letters and numbers.
The login name and password are case sensitive.
Do not use words that would be easily guessed by another person, words that are in dictionaries, or text strings like abcdefg.

## 7-10-2 Data Protection

This section describes the data protection of the Safety CPU Unit.

### ● Data Protection

The data protection function allows you to set passwords for individual data units to restrict displaying and changing them (access restrictions). You can enter the password to temporarily release the protection from a data unit. Data protection is set and released offline.

### ● Types of Access Restrictions

The following table describes the types of access restrictions.

| Access restriction | Operation | Remarks |
|---|---|---|
| **Display prohibition** | The restricted data cannot be displayed. | This restriction applies to jumping from other windows, viewing search results, and printing previews. |
| **Change prohibition** | The restricted data cannot be changed. The data can be displayed but not changed on the display. | This restriction applies to changing or replacing text in POU names. |

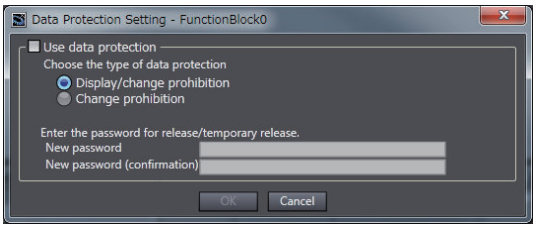
## ● Data That Can Be Protected

The following table lists the data that you can protect from being displayed and changed.

| Target data | | Display prohibition | | Change prohibition | |
|---|---|---|---|---|---|
| | | Display | Change | Display | Change |
| **Program** | **Program names** | Possible | Not possible | Possible | Not possible |
| | **Variable tables** | Not possible | Not possible | Possible | Not possible |
| | **FBD editor** | Not possible | Not possible | Possible | Not possible |
| **Function Block** | **Function block names** | Possible | Not possible | Possible | Not possible |
| | **Variable tables** | Not possible | Not possible | Possible | Not possible |
| | **FBD editor** | Not possible | Not possible | Possible | Not possible |

## ● Procedure to Set Protection

***1*** Select the data to protect, and then select**Security**−**Set/Release Data Protection**from the**Controller**Menu. Or right-click the data to protect, and select**Security**−**Set/Release Data Protection**from the menu.
The Data Protection Setting Dialog Box is displayed.




***2*** Select the**Use data protection**Check Box and select the access restrictions. Enter the password, and click the**OK**Button.
Data protection is set and the icon on the protected data changes in the Multiview Explorer.

| Icon | Description |
|---|---|
|  | Data protection is disabled. |
|  | Data protection is enabled. |

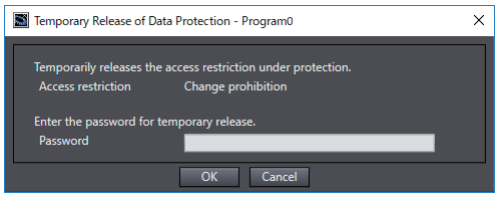
**Precautions for Correct Use**

- You will not be able to temporarily release protection or remove the protection setting if you forget the password. Protection settings are also transferred for synchronization operations. If you forget the passwords for protected data that was transferred to the Controller, you will no longer be able to display or change the protected data. Record the password in case you forget it.
- If you change the data protection setting, the safety program in the project will change to an unvalidated state. Execute safety validation again.

## ● Temporarily Releasing Protection

To perform operations on protected data, you can enter the password to temporarily release the protection.

***1*** Perform the restricted operation for the protected data. When you temporarily release data protected with **Change prohibition**, right-click the data and select **Security** − **Temporary Release of Change Prohibition** from the menu.
The Temporary Release of Data Protection Dialog Box is displayed.




***2*** Enter the password that was set when data protection was set.
The access restrictions are temporarily released.
The following table gives the lengths of time for which the access restrictions are temporarily released.

| Access restriction | Length of time that protection is cleared |
|---|---|
| Display prohibition | While the project is open |
| Change prohibition | While the project is open<br>While the project is open or until **Finish Temporary Release of Change Prohibition** is selected from the menu |

The Sysmac Studio is locked for 10 minutes if you consecutively enter wrong passwords 5 times for the same Controller. The Temporarily Release of Data Protection Dialog Box is displayed again in 10 minutes.

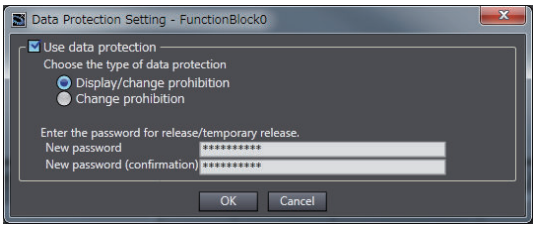
**Precautions for Correct Use**

Even if the access restrictions for **Display prohibition** are set, the Temporary Release of Data Protection Dialog Box is not displayed for printing or printing previews.
Display the data from the Multiview Explorer and temporarily release protection before you use printing or print previews.

## ● Procedure to Release Protection

**1** Select the data for which to release protection, and then select **Security** − **Set/Release Data Protection** from the **Controller** Menu. Or right-click the data for which to release protection, and select **Security** − **Set/Release Data Protection** from the menu.
The Temporary Release of Data Protection Dialog Box is displayed.

**2** Enter the password, and click the **OK** Button.
The Data Protection Setting Dialog Box is displayed.




**3** Clear the **Use data protection** Check Box and click the **OK** Button.
Data protection is released and the protection icon returns to the normal icon.

**Version Information**

You can use program data protection with Sysmac Studio version 1.17 or higher.
You can import user-defined function blocks with Sysmac Studio version 1.12 or higher.

## ● Data Protection Version

You can increase the level of protection for POU data by setting a higher Data Protection Version. The Data Protection Version can be set for each Safety CPU Unit. We recommend setting a higher Data Protection Version.

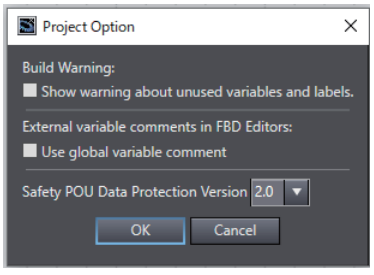
**Version Information**

- You can set Data Protection Version with Sysmac Studio version 1.54 or higher.
- If a new project is created, the Data Protection Version is 1.0. Set a higher version if necessary.

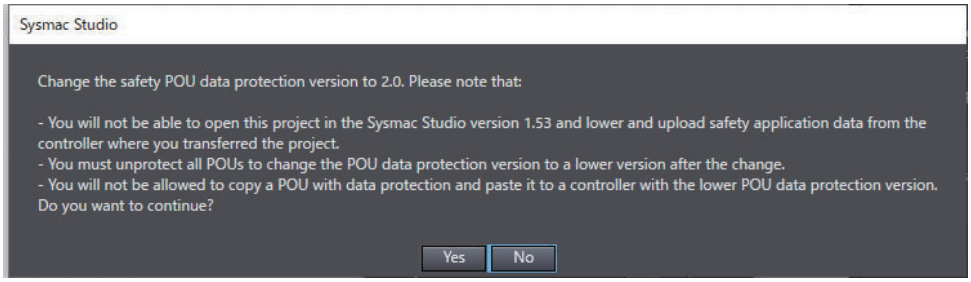
## ● Procedure to Change the Data Protection Version

**1** Select **Project Option** from the **Project** Menu on the Safety CPU Unit Setup and Programming View.

**2** Select the Safety POU Data Protection Version in the Project Option Dialog Box, and then click the **OK** Button.

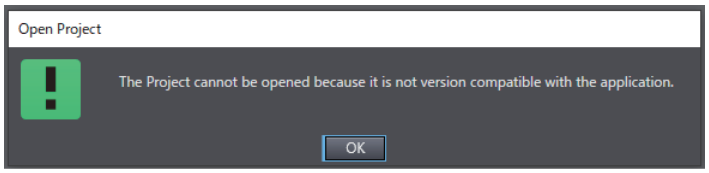


**3** Check the contents of the confirmation message to change the Data Protection Version, and then click the **Yes** Button.




### Precautions for Correct Use

- The Data Protection Version cannot be lowered unless the protection setting is released. Refer to *Procedure to Release Protection* on page 7-62 for the procedure for releasing data protection settings.
- A project whose Data Protection Version is set to 2.0 cannot be opened with Sysmac Studio version 1.53 or lower. The following message is displayed.




- If you set a lower Data Protection Version, use it only for the purpose of maintaining compatibility with lower versions of Sysmac Studio.

# 7-11 Performing Safety Validation and Operation

This section describes the procedure for safety validation testing. Safety validation testing is used to confirm that all safety functions and all Safety Control Units meet the required specifications of the safety system. If safety validation testing demonstrates that the safety controls meet the required specifications of a safety system, the safety application data is appended with confirmation information through a process called "safety validation".
When you perform safety validation on a Safety CPU Unit that is operating in DEBUG mode, the safety application data is saved in the non-volatile memory of the Safety CPU Unit.
Note that a safety program for which safety validation has been performed is saved in the project file as a validated program with a safety signature assigned.

This section describes how to perform safety validation and start operation after you have debugged the safety programs.

## 7-11-1 Performing Safety Validation

You must perform safety validation before you change Safety CPU Unit to the RUN mode and start any safety control system that uses safety application data*1 that is created with Sysmac Studio.
You perform safety validation after you perform safety validation testing with the Safety CPU Unit in DEBUG mode (RUN) to make sure that all safety functions operate as intended.
To perform safety validation, it is necessary that the Safety CPU Unit be in DEBUG mode.

*1. The safety application data includes the safety programs and the safety task settings and variables. Refer to *7-1 Procedures before Operation and Transferring the Required Data* on page 7-3 for details.

**⚠ WARNING**

Before you perform safety validation of the safety programs, complete debugging of the safety programs.

Otherwise, the Safety CPU Unit will start with safety programs that are not fully debugged and may cause serious personal injury.

**⚠ WARNING**

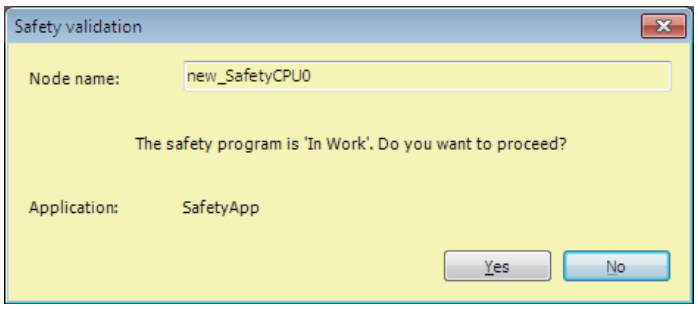
Verify the calculated reaction times for all safety chains to confirm that they satisfy the required specifications.
Serious injury may possibly occur due to loss of required safety functions.

**Additional Information**

You can manage changes in the safety application data before you perform safety validation after debugging is completed, or after you perform safety validation. Refer to *A-4 Change Tracking* on page A-70 for details.

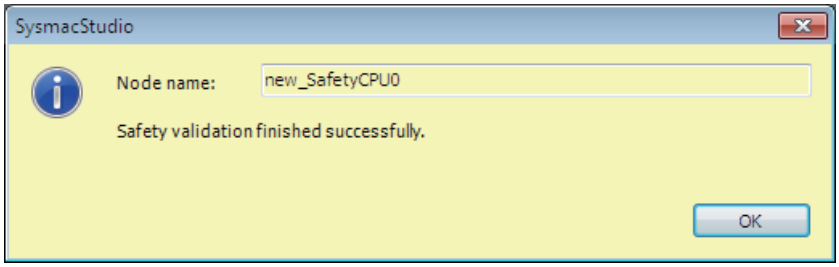
### ● Performing Safety Validation

**1** Connect the Sysmac Studio online with the Safety CPU Unit, place the Safety CPU Unit in DEBUG mode, and select **Safety Validation** from the **Controller** Menu.
The following confirmation dialog is displayed.




**2** Click the **Yes** button.
After the validated safety programs are saved to non-volatile memory in the Safety CPU Unit, the following dialog is displayed to indicate the process was completed, and then the Safety CPU Unit enters the state shown below.

- Safety CPU Unit is set to the validated state, and assigned with a safety signature.
- The safety program in the project file is set to the validated state, and assigned with a safety signature.
- The VALID indicator of the Safety CPU Unit changes from not lit to lit yellow.
- When you cycle the power, the Safety CPU Unit starts in RUN mode.




**3** Click the **OK** button.

**Precautions for Safe Use**

Note that the Safety CPU Unit automatically starts in RUN Mode at the next start-up if the safety validation is successful.
When you download the parameters for the EtherCAT Coupler Unit and NX Units, the Safety CPU Unit automatically restarts.

## ● Checking the Validated State

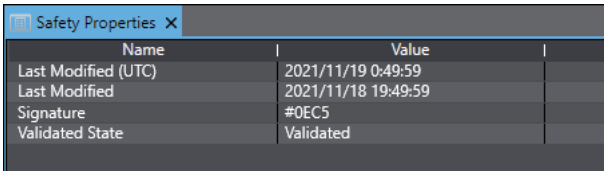
Use the following procedure to check if the safety program in the project has been validated.

**1** Start Sysmac Studio. Open a project containing safety application data.

**2** In the Multiview Explorer, select Safety CPU Unit from Controller Selection.

**3** From the Controller Menu, select **Project** – **Safety Properties**.

The following window is displayed.




The contents of each item are described in the following table.

| Item | Description |
|---|---|
| Last Modified (UTC) | The time on the computer on which safety validation has been performed. |
| Last Modified | |
| Signature | Displays the signature code (four-digit hexadecimal number) of the validated safety program. **---** is displayed if safety validation has not been performed. |
| Validated State | Displays whether safety validation has been performed. |

If *Validated* is displayed for Validated State, the safety program in the project file has been validated.

## 7-11-2 Changing to RUN Mode

After you perform safety validation, you can change the Safety CPU Unit to RUN mode.
Use one of the following procedures to change the Safety CPU Unit to RUN mode.

- Cycle the power supply to the Safety CPU Unit
- Change to RUN mode via the Sysmac Studio.

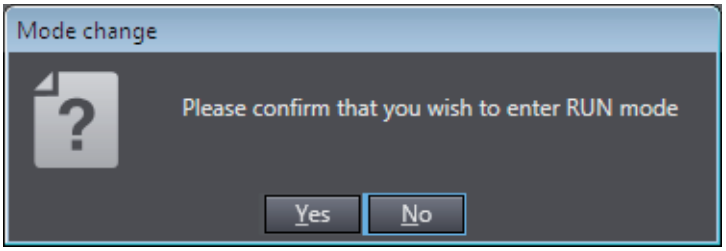
### Change to RUN Mode via the Sysmac Studio

The safety programs must be validated.

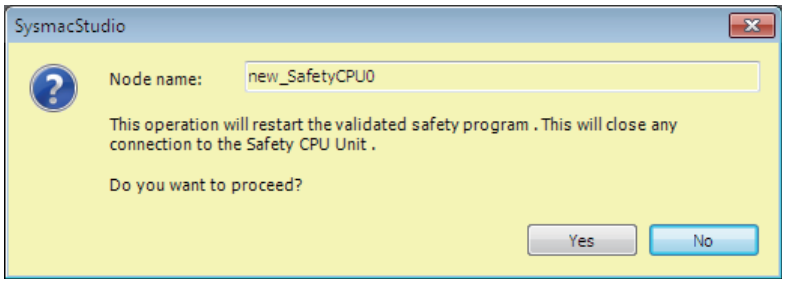
***1*** With the Safety CPU Unit connected online, perform one of the following operations on the Safety CPU Unit Setup and Programming View.

- Select **Mode** – **RUN Mode** from the **Controller** Menu.
- Press the **Ctrl** + **3** Keys.
- Click the **RUN Mode** Button on the toolbar.

A confirmation dialog box is displayed.




Click the **Yes** Button. A dialog box is displayed to confirm the node.




**2** Check the node name, and click the **Yes** Button.
A Mode Change Confirmation Dialog Box is displayed. Click the **OK** Button to change the Safety CPU Unit to RUN mode.

## 7-11-3 Changing to PROGRAM Mode

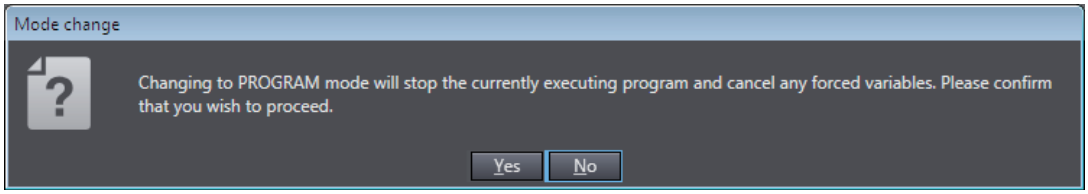
If you need to change the safety programs, or if you need to change the operating mode of the Safety CPU Unit from RUN mode to DEBUG mode, you must first change to PROGRAM mode.

### Changing to PROGRAM Mode

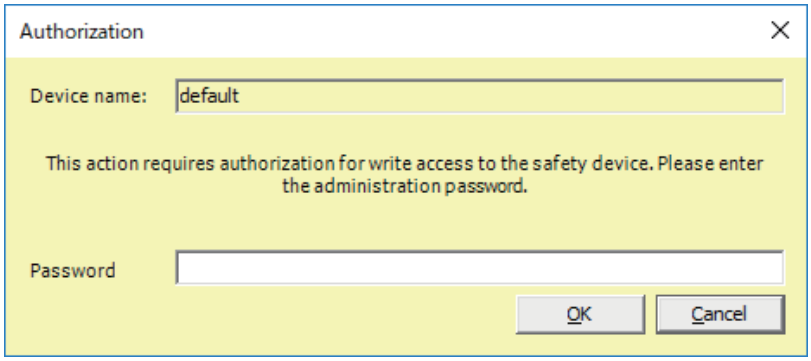
Use the following procedure to change the Safety CPU Unit from RUN mode to PROGRAM mode.

**1** With the Safety CPU Unit connected online, perform one of the following operations.
- Select **Mode** – **PROGRAM Mode** from the **Controller** Menu.
- Press the **Ctrl** + **1** Keys.
- Click the **PROGRAM Mode** Button on the toolbar.

A confirmation dialog box is displayed.

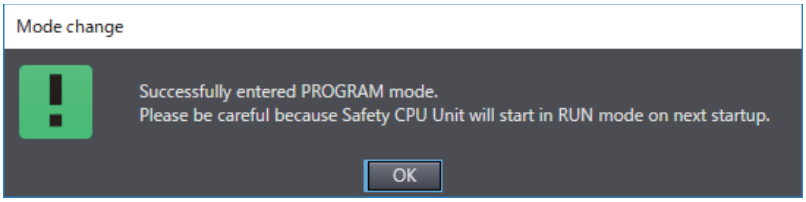



When you click the **Yes** Button, a dialog box to confirm the Safety Password appears.




**2** Enter the safety password, and click the **OK** Button.
The Safety CPU Unit enters PROGRAM mode.

The following dialog box is displayed.




Click the **OK** Button.

# 7-12 Starting and Stopping the Safety Application Monitoring

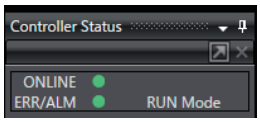
This section describes how to start and stop the monitoring function for variables and the Safety I/O Unit by using Sysmac Studio. This function can be executed only when the Safety CPU Unit is in the RUN mode.
For details on monitoring the variables, refer to *7-6-3 Monitoring Variables in the FBD Editor* on page 7-29 and *7-6-4 Monitoring Variables in a Watch Tab Page* on page 7-30. For details on monitoring the Safety I/O Unit, refer to *7-5-2 Monitoring Safety I/O Units* on page 7-20.

## 7-12-1 Procedure to Start and Stop the Safety Application Monitoring

***1*** Make sure that the Safety CPU Unit is in the RUN mode.

***2*** Connect to Safety CPU Unit online.

***3*** Select Safety CPU Unit from the Controller Selection Box in the Multiview Explorer of Sysmac Studio and open the Safety CPU Unit Setup and Programming View.
When you open the view for the Safety CPU Unit, the Controller status is displayed in the lower-right corner of the screen as shown below.

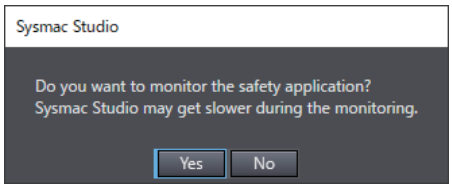



***4*** On the Safety CPU Unit Setup and Programming View, perform one of the following operations.

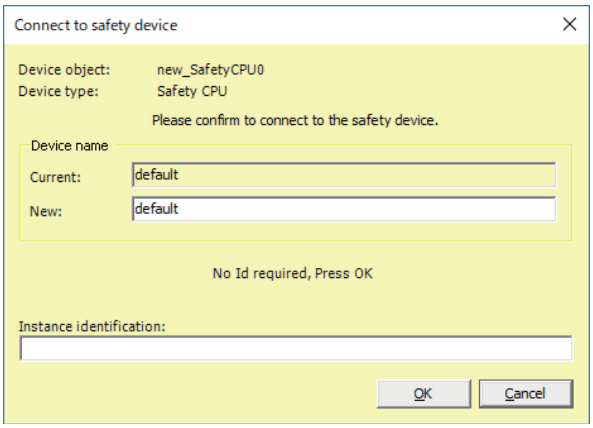
**To Start Monitoring:**

- On the menu bar, select **Controller** - **Monitor**.
- Click the **Monitor** Button on the toolbar.

Before monitoring starts, the following confirmation dialog box appears. Click the **Yes** button.




A confirmation dialog box for connection to a safety device is displayed. Click the **OK** button.




**To Stop Monitoring:**

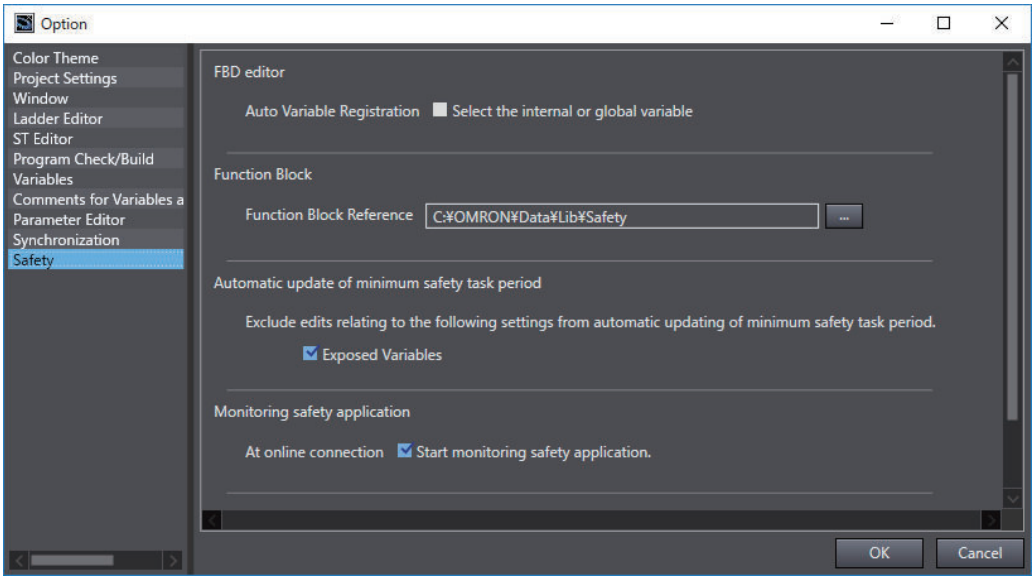
- On the menu bar, select **Controller** - **Stop Monitoring**.
- Click the **Stop Monitoring** Button on the toolbar.

## 7-12-2 Changing the Monitoring Options for the Safety Application

You can select whether you will start the safety application monitoring when Sysmac Studio is connected online. Use the following procedure.

***1*** On the menu bar, select **Tools** - **Option**.
The **Option** Dialog Box is displayed.

***2*** Click the **Safety** Tab.
The monitoring options for the safety application will appear.




***3*** Specify the option **At online connection Start monitoring safety application** and click the **OK** Button.
If this option is disabled, the safety application monitoring will not start. For the procedure to start monitoring, refer to *7-12-1 Procedure to Start and Stop the Safety Application Monitoring* on page 7-70.

# 7-13 Uploading Configuration Information and Safety Application Data

This section describes how to use the Sysmac Studio to transfer the configuration information and safety programs from the Safety Control Unit to the computer.

## 7-13-1 Outline

You can transfer the settings of the Safety Control Unit and the safety programs from the Safety Control Unit to the computer.

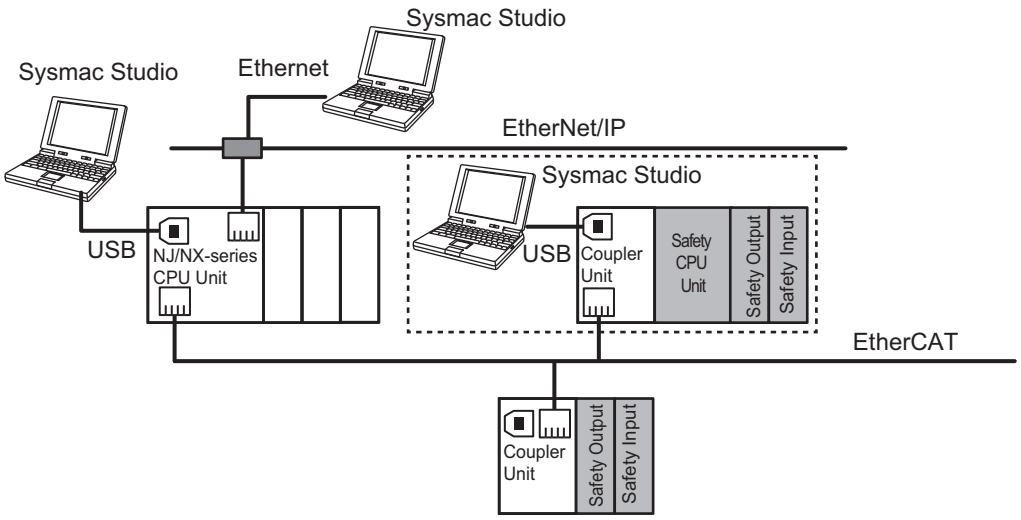
### Paths for Going Online

There are the following two ways to connect the Sysmac Studio to the Safety CPU Unit.

- USB connection or Ethernet connection to the NJ/NX-series CPU Unit*1
- USB Connection to the Communications Coupler Unit

*1. Connection is possible when you use an NJ/NX-series CPU Unit and EtherCAT Coupler Unit.

A configuration example for an EtherCAT Slave Terminal is given below.




## 7-13-2 Uploading Data for a Connection to an NJ/NX-series CPU Unit

When you use an EtherCAT Coupler Unit, you can connect the Sysmac Studio to a USB or Ethernet port on the NJ/NX-series CPU Unit to upload the configuration information and safety application data. Use the following procedure to upload the data.

***1*** Select the NJ/NX-series CPU Unit from the Controller Selection Box in the Multiview Explorer on the Sysmac Studio to change to the NJ/NX-series CPU Unit Setup and Programming View.

***2*** Set the communications path to the NJ/NX-series CPU Unit.

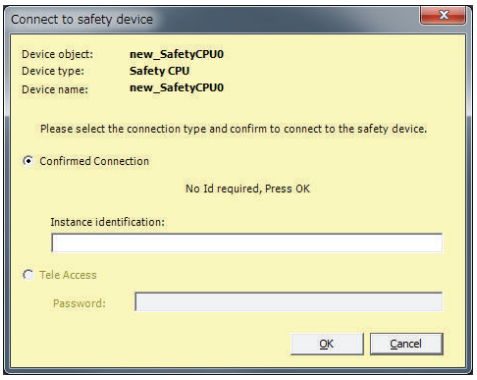
**3** Select **Online** from the **Controller** Menu. Or, click the Go Online Button ( ) in the toolbar.

**4** Select **Synchronization** from the **Controller** Menu.

**5** In the Synchronization Window, clear the selections of the following check boxes.
Do not transfer the following. (All items are not synchronized.)
- CJ-series Special Unit parameters and EtherCAT slave backup parameters
- Slave Terminal Unit operation settings and NX Unit application data

**6** Click the **Transfer from Controller** Button.
An execution confirmation dialog box is displayed.

**7** Click the **Yes** Button.
The following Connect to safety device Dialog Box is displayed.




**Note** You do not need to enter anything in the *Instance identification* Box.

**8** Click the **OK** Button.
A password confirmation dialog box is displayed.

**9** Enter the password, and click the **OK** Button.
The settings of all the Units that are connected to the EtherCAT Coupler Unit are transferred to the Sysmac Studio.

## 7-13-3 Uploading Data for a Connection to a Communications Coupler Unit

If the Sysmac Studio is connected to the USB port on the Communications Coupler Unit, use the following procedure to upload data from the Safety CPU Unit.

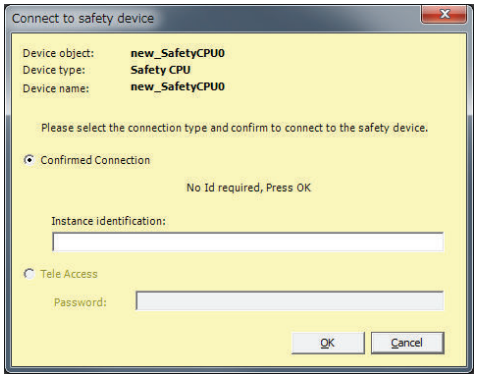
**1** Right-click the Communications Coupler Unit on the Slave Terminal Tab Page, and select **CouplerConnection (USB)** – **Online** from the menu.
The Sysmac Studio goes online with the Slave Terminal.
**Note** If you are using a new project and there is no EtherCAT Coupler Unit in the EtherCAT Slave Terminal Tab Page, add an EtherCAT Coupler Unit and set the node address.

**2** Right-click the Communications Coupler Unit on the Slave Terminal Tab Page, and select **Transfer from Coupler** from the menu.

An execution confirmation dialog box is displayed.

**3** Click the **Yes** Button.

- When There Is No Safety CPU Unit in the Slave Terminal Configuration
  The settings of all the Units that are connected to the EtherCAT Coupler Unit are transferred to the Sysmac Studio. This completes the upload procedure.
- When There Is a Safety CPU Unit in the Slave Terminal Configuration
  A confirmation dialog box for connection to a safety device is displayed.
  Go to the next step and continue the procedure.




**Note** You do not need to enter anything in the *Instance identification* Box.

**4** Click the **OK** Button.
A password confirmation dialog box is displayed.

**5** Enter the password, and click the **OK** Button.
The settings of all the Units that are connected to the Communications Coupler Unit are transferred to the Sysmac Studio.

### Precautions for Correct Use

If you use an EtherCAT Coupler Unit, upload the data from the Safety CPU Unit only after you have added all of the Safety I/O Units to the EtherCAT Slave Terminal Tab Page.
If you use the following procedure to upload data and then reset the warning, you cannot monitor the project.
If you upload the data from the Safety CPU Unit when not all of the Safety I/O Units are registered, a warning is displayed.
You can reset the warning by uploading data from all of the Safety I/O Units.

# 7-14 Transferring Safety Application Data

To use the validated safety application data described in the previous section *7-11 Performing Safety Validation and Operation* on page 7-65 in another Safety CPU Unit, change the other Safety CPU Unit to DEBUG mode and perform safety validation.

# 7-15 Monitoring Controller Status

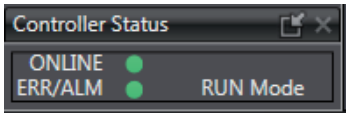
This section describes how to display the status of Safety CPU Unit that is connected to Safety CPU Unit online or the status when the Simulator is connected.

## Controller Status Monitor

Control status monitoring is used to display the status of the connected Safety CPU Unit or Simulator in the Controller Status Pane. The Controller Status Pane is displayed when the Sysmac Studio is online or the Simulator is connected.

## Displaying the Controller Status Pane

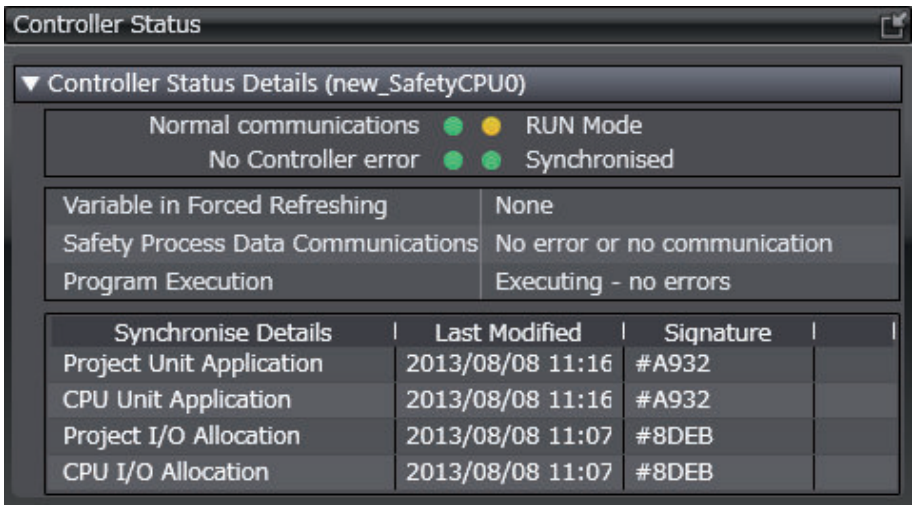
The basic view of the Controller Status Pane is displayed in the Toolbox in the lower right corner of the Safety CPU Unit Setup and Programming View when the Safety CPU Unit is online or the Simulator is connected.




## Expansion Operations in the Controller Status Pane

Use the buttons (  ) in the title bar of the Controller Status Pane to switch between the basic and detailed views.

- Detailed View



| Information | Displayed information and indicator colors |
|---|---|
| Communications Status | Displays the communications status between the Safety CPU Unit and Sysmac Studio or displays the Simulator startup status.<br>• Normal communications: Lit green<br>• Communications error: Flashing red |
| Error Status | Displays the error status of the Safety CPU Unit or Simulator.<br>• No Controller error: Lit green<br>• Partial or minor fault level Controller error occurs: Lit yellow. |
| Operating Mode | Displays the operating mode of the Safety CPU Unit.<br>• RUN Mode: Lit yellow.<br>• DEBUG mode (RUN): Flashing yellow.<br>• DEBUG mode (STOPPED): Flashing yellow.<br>• PROGRAM Mode: Not lit.<br>• UNKNOWN Mode: Flashing red. |
| Synchronization Status | Displays the comparison results between the project file on the computer and the data in the Safety CPU Unit.<br>• Synchronized: Lit green.<br>• Not synchronized/not executed: Lit yellow. |
| Variable in Forced Refreshing | Displays the forced-refreshing status of variables in the safety programs.<br>• None<br>• Present |
| Safety Data Communications | It shows the status of communications among the Safety CPU Unit, the FSoE slave device, and the CIP Safety target device.<br>• No error or no communications<br>• Communications error |
| Program Execution | Displays the execution status of the safety programs.<br>• Executing - no errors<br>• Executing - instruction execution error<br>• Not executing |
| Synchronise Details | Displays the synchronization information item, last modified date, and signature (CRC data). |



**Additional Information**

You can use the color of the top of the Edit Pane to check if you are online with the Safety CPU Unit or connected to the Safety Simulator.

- Connected to the Safety CPU Unit online
  The top of the Edit Pane is yellow.
- Connected to the Safety Simulator:
  The top of the Edit Pane is green.

# 7-16 Restarting and Clearing All Memory

## 7-16-1 Restarting

Restarting allows you to restart the CPU Rack or Slave Terminal that includes the Safety CPU Unit and Safety I/O Units without cycling the NX bus master power.
Refer to the user's manual for the connected CPU Unit or Communications Coupler Unit for details.

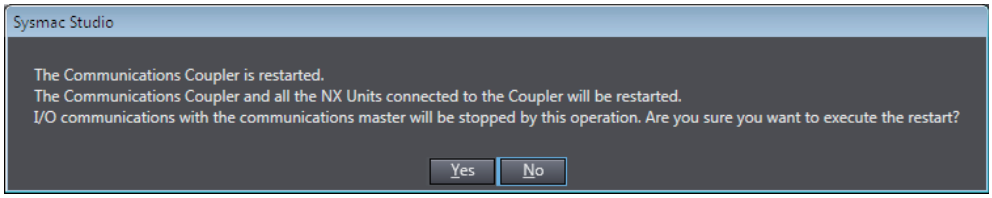
**Precautions for Safe Use**

If the safety application data in the Safety CPU Unit is validated, be careful when you execute the Restart operation because the Safety CPU Unit will automatically start in RUN mode.

### Restarting Slave Terminals

Use the following procedure to restart all of the Units in the Slave Terminal.

*1* Go online, right-click the Communications Coupler Unit in the Slave Terminal Tab Page, and select **Restart**.
The Restart Confirmation Dialog Box is displayed.

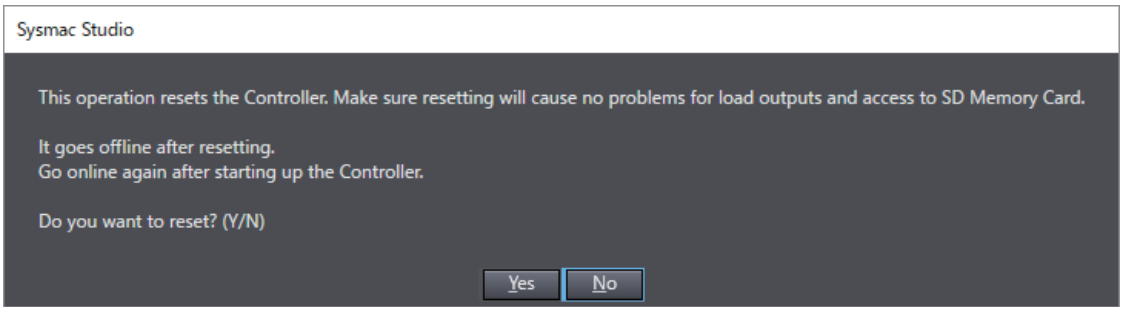



*2* Click the **Yes** Button.
After the Units are restarted, a Restart Completion Dialog Box is displayed.

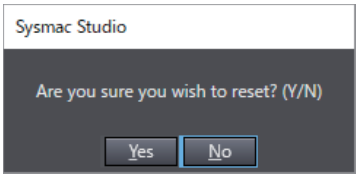
### Controller Reset

All the units on the CPU rack can be restarted using the controller reset function. The controller reset function is available only when the CPU Unit operating mode is set to PROGRAM mode.

*1* Go online, select **Reset Controller** from the **Controller** Menu.
The following Confirmation Dialog Box is displayed.




**2** Click the **Yes** Button.
The following Confirmation Dialog Box is displayed.




**3** Click the **Yes** Button.
The online state is cancelled, and the system restarts. To continue online operations, enable online mode again.

## 7-16-2 Clear All Memory Operation

For the Clear All Memory operation, you use the Sysmac Studio to initialize the contents of the Safety CPU Unit and Safety I/O Units to the default settings.
For the Safety Control Unit, the Clear All Memory operation can be performed in the following two ways.

| Type | Function |
|---|---|
| Clear All Memory Operation for NX Units | This method clears all memory contents from the Safety CPU Unit and Safety I/O Units. |
| Clear All Memory operation for a Slave Terminal | This method clears all memory from the Communications Coupler Unit and all NX Units, including the Safety I/O Units that are connected to the Communications Coupler Unit. The Safety CPU Unit memory cannot be cleared. |
| Clear All Memory Operation for Controllers | This method clears all memory from the NX-series CPU Unit and all NX Unit, including the Safety I/O Units that are connected to the NX-series CPU Unit. The Safety CPU Unit memory cannot be cleared. |

Refer to the user's manual for the connected Communications Coupler Unit for information on the Clear All Memory operation for Slave Terminals.
For information on the Clear All Memory operation of the controller, refer to the user's manual for the connected NX-series CPU Unit.

**Precautions for Correct Use**

- The memory in the Safety CPU Unit is not cleared even if you right-click the Communications Coupler Unit in the Slave Terminal Tab Page and select **Clear All Memory** for all Units. To clear the memory of a Safety CPU Unit, use the Clear All Memory operation for NX Units.
- The memory in the Safety CPU Unit is not cleared even when you perform the Clear All Memory operation for controllers. To clear the memory of the Safety CPU Unit, use the Clear All Memory operation for NX Units.
- You can execute the Clear All Memory operation for a Safety CPU Unit only when the Safety CPU Unit is in PROGRAM mode.

## Scope of Data to Clear and State of Memory After It Is Cleared

### ● Safety CPU Unit

| Data item | Status after clear all memory operation |
|---|---|
| I/O allocation information | This data is set to the default settings (I/O size = 0 bytes). |
| Safety programs | This data is set to the default settings (no programs). |
| Safety password | This data is set to the default settings (no password). |
| Event logs | Event logs are cleared if you select the Clear the event logs Option when you execute the Clear All Memory operation. |

### ● Safety I/O Units

| Data item | Status after clear all memory operation |
|---|---|
| FSoE slave address | This data is set to the default setting (no setting). |
| Event logs | Event logs are cleared if you select the Clear the event logs Option when you execute the Clear All Memory operation. |

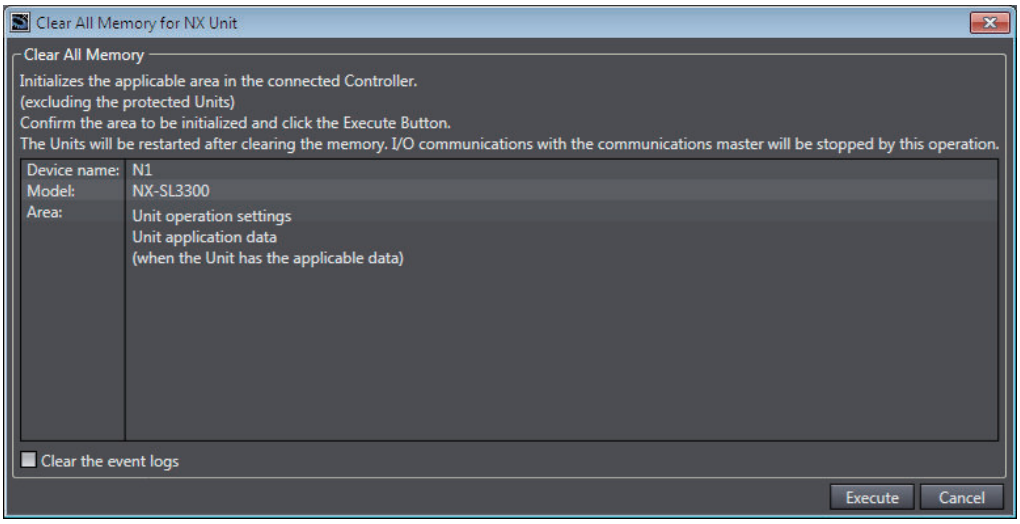
**Precautions for Safe Use**

After you clear the memory, the Controller operates in the same way as immediately after you create the system configuration with the Controller in the factory default condition.

## Procedure for Clear All Memory Operation

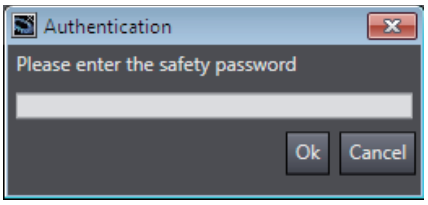
### ● Clear All Memory Operation for NX Units

*1* Go online, right-click a Safety CPU Unit or Safety I/O Unit on the Slave Terminal Tab Page, and select **Clear All Memory** from the menu. You can select this menu command only when the Safety CPU Unit is in PROGRAM mode.
The Clear All Memory for NX Unit Dialog Box is displayed.

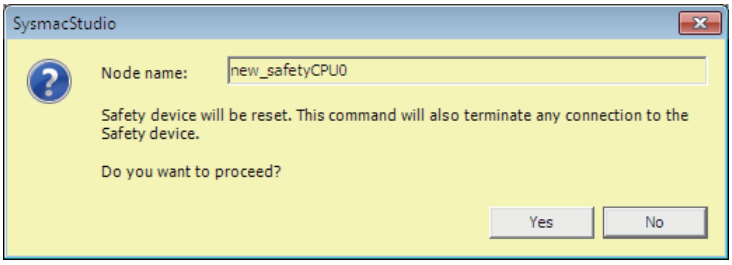



**2** Click the **Execute** Button. The Clear All Memory Confirmation Dialog Box is displayed.

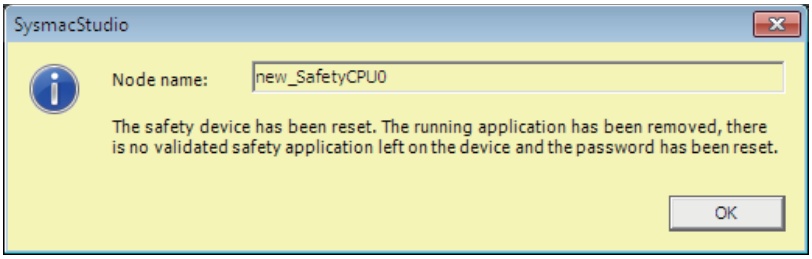
**3** Click the **Yes** Button. The Authentication Dialog Box is displayed.




**4** Enter the password, and click the **OK** Button. If a password is not set, leave the box empty and click the **OK** Button.
A dialog box is displayed to confirm the node.



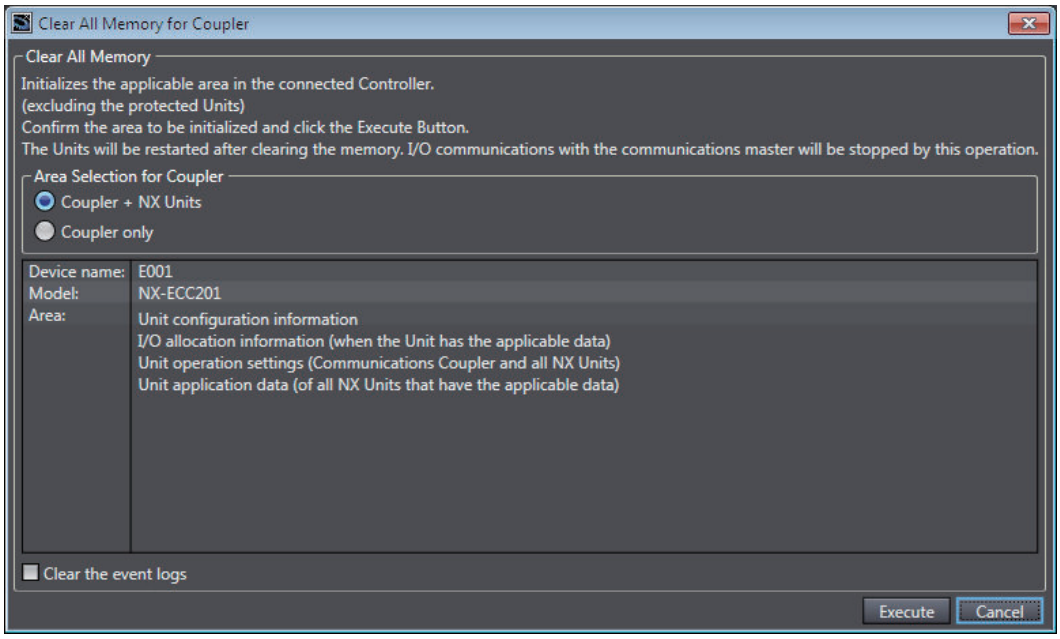

**5** Click the **Yes** Button. The following dialog box is displayed.



**6** Click the **OK** Button.
After memory is cleared, the Memory All Cleared Dialog Box is displayed.

## ● Clear All Memory operation for a Slave Terminal

**1** Go online, right-click the Communications Coupler Unit on the Slave Terminal Tab Page, and select **Clear All Memory** from the menu.
The **Clear All Memory** dialog box is displayed.



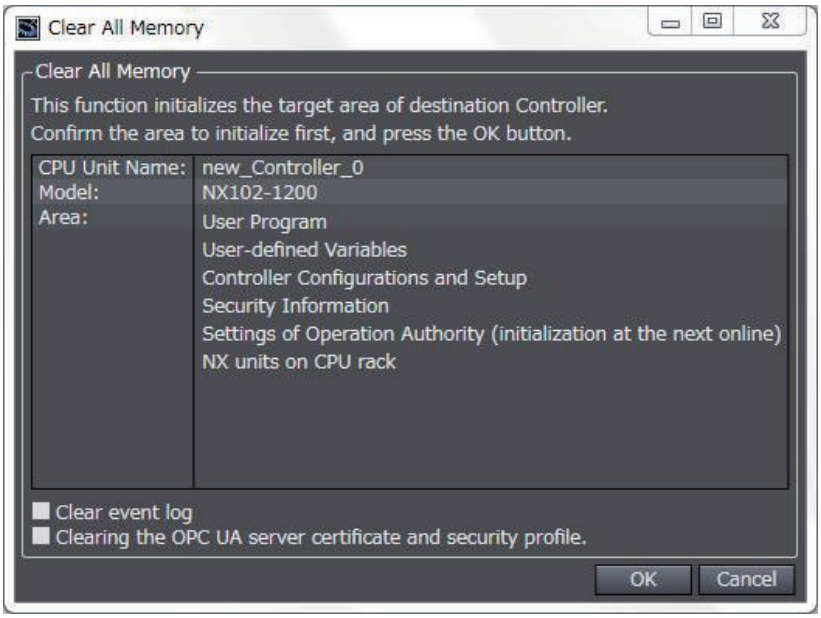

**2** Check the areas to clear and then click the **Execute** Button.
- To clear the event logs, select the **Clear the event logs** check box.
- To clear the memory in all Units, select the **Coupler + NX Units** Option in the **Area Selection for Coupler** Area.

An execution confirmation dialog box is displayed.

**3** Click the **Yes** Button.
After memory is cleared, the Memory All Cleared Dialog Box is displayed.

## ● Clear All Memory Operation for Controllers

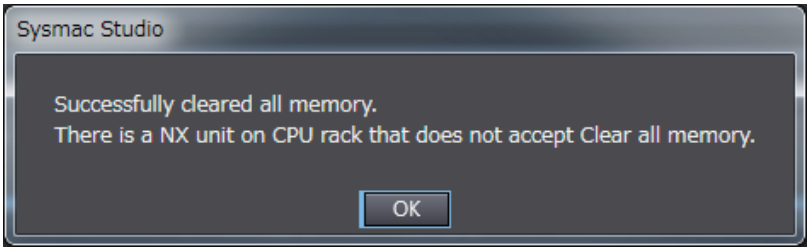
**1** Go online, and select **Controller** – **Clear All Memory** from the NJ/NX-series Unit Setup and Programming View.
The **Clear All Memory** Dialog Box is displayed.




**2** Check the areas to clear and then click the **OK** Button.

- To clear the event logs, select the **Clear the event logs** check box.

After memory is cleared, the Memory All Cleared Dialog Box is displayed.



# 8

# Calculating Safety Reaction Times

This section describes how to calculate safety reaction times for Safety Control Units.


**8-1 Safety Reaction Time ..... 8-2**
8-1-1 Calculating the Safety Reaction Time ..... 8-2
8-1-2 Verifying Safety Reaction Times ..... 8-4

**8-2 Safety Task ..... 8-5**
8-2-1 Safety Task ..... 8-5
8-2-2 Operation of Safety Task ..... 8-5
8-2-3 Minimum Safety Task Period ..... 8-5
8-2-4 Setting the Safety Task Period ..... 8-7

**8-3 FSoE Watchdog Timer ..... 8-9**
8-3-1 FSoE Watchdog Timers ..... 8-9
8-3-2 Checking FSoE Watchdog Timers ..... 8-9
8-3-3 Settings for FSoE Watchdog Timers ..... 8-10
8-3-4 Changing FSoE Watchdog Timers ..... 8-10

**8-4 EPI (Data Packet Interval) ..... 8-11**
8-4-1 Changing the EPI ..... 8-11
8-4-2 EPI Restrictions ..... 8-11

# 8-1 Safety Reaction Time

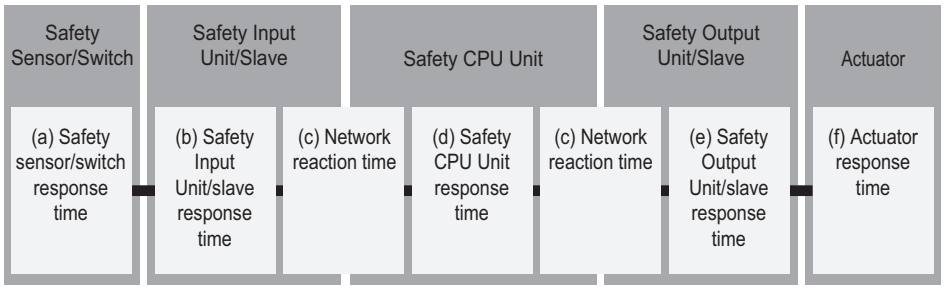
This section describes the safety reaction time (i.e., the safety response performance) of Safety Control Units.
The safety reaction time is the maximum time required to turn OFF an output in consideration of possible failures and breakdowns over safety chains*1. In the safety system design, the safety distance is calculated based on the safety reaction time. For all safety chains, the longest time required to stop moving equipment from when a safety input was activated must satisfy the required specifications.

*1. The safety chain is the logical connections that are required to achieve a safety function, including the safety input device, Safety Control Units, and the safety output device.

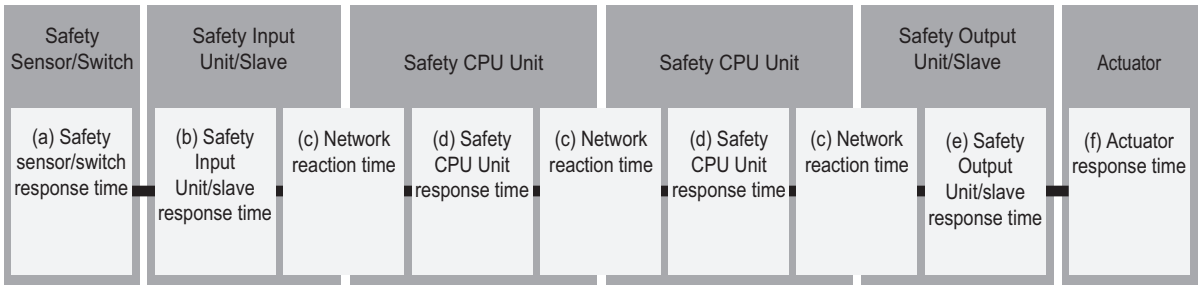
## 8-1-1 Calculating the Safety Reaction Time

As shown in the figure below, the safety reaction time is the sum of (a) safety sensor/switch response time, (b) Safety Input Unit/slave response time, (c) Network reaction time, (d) Safety CPU Unit response time, (e) Safety Output Unit/slave response time, and (f) actuator response time. The number of elements and the duration of time vary depending on the safety chain route.

Basic Configuration




Network Configuration between Controllers




Details of each time element are described in the following table.

| Letter | Time element | Description |
|---|---|---|
| (a) | Safety sensor/ switch response time | This is the response time that is required for a safety sensor or switch, such as a light curtain, to turn OFF. The value is defined for each sensor or switch.<br>Use the following values when an OMRON Special Safety Input Device is connected to a Safety Input Unit.<br>E3ZS/E3FS Single-beam Safety Sensors : 14 ms<br>D40A-2/D40A Non-contact Door Switches : 6 ms + 0.4 ms x No. of linked Switches*1<br>D40Z Non-contact Door Switches : 29 ms<br>UM/UMA Safety Mats : 24 ms*2<br>SGE Safety Edges : 24 ms |
| (b) | Safety Input Unit/ slave response time | This is the input response time required for a FSoE salve or CIP Safety slave, such as Safety Input Units. A specific value is defined for each device.<br>The response time of Safety Input Units is as follows.<br>NX-SIH400: 9 ms + On-Off delay time<br>NX-SID800: 5 ms + On-Off delay time<br>GI-SID1224: 11 ms + On-Off delay time<br>GI-SMD1624: 11 ms + On-Off delay time<br>NXR-SXD1204□-CST: 10ms + Safety input terminal On-Off delay time<br>For any other device, refer to the manual for the device. |
| (c) | Network reaction time | This is the response time required for the CIP Safety connections or the FSoE connections. These values can be verified on the Sysmac Studio. For details on how to check the network reaction time, refer to *5-4-2 Setting Up the CIP Safety Communications* on page 5-13.<br>For FSoE connections: FSoE Watchdog Timer value<br>For CIP Safety connections: Network reaction time value calculated by the Sysmac Studio |
| (d) | Safety CPU Unit response time | Response time of the Safety CPU Unit. This value is an integer multiple of the safety task period. The value varies, depending on the safety chain route as follows.<br>When the FSoE connections are used for both input and output : Safety task period x 0<br>When the CIP Safety connections are used for both input and output : Safety task period x 2<br>When the FSoE connections are used for input and CIP Safety connections are used for output : Safety task period x 1<br>When the CIP Safety connections are used for input and the FSoE connections are used for output : Safety task period x 1 |
| (e) | Safety Output Unit/slave response time | This is the output response time required for a FSoE salve or CIP Safety slave, such as Safety Output Units. A specific value is defined for each device.<br>The response time of the Safety Output Units is as follows.<br>NX-SOH200: 1 ms<br>NX-SOD400: 1 ms<br>R88D-1SN□: 4 ms<br>R88D-1SN□-51: 4 ms (STO)*3<br>R88D-1SAN□: 7 ms (STO)*4<br>GI-SMD1624: 4.5 ms + Output test pulse width<br>NXR-SXD1204□-CST: 5ms + Safety output terminal On-Off delay time<br>For any other device, refer to the manual for the device. |

| Letter | Time element | Description |
|---|---|---|
| (f) | Actuator response time | This is the response time that is required for an actuator, such as a safety relay, to turn OFF. The value is defined for each actuator. |

*1. The fault detection time for a 24 V short-circuit fault in a D40A-2/D40A Non-contact Door Switch is 18 ms. If usage is for an application other than a Door Switch, use a safety sensor/switch response time of 18 ms.

*2. Refer to *Precaution for Conformance to ISO 13856-1:2013* on page 8-4 for a precaution on conformance to ISO 13856-1:2013.

*3. For details on using safety functions other than the STO function, refer to the *AC Servomotors/Servo Drives 1S-series with Built-in EtherCAT® Communications User's Manual (Cat. No. I696)*.

*4. For details on using safety functions other than the STO function, refer to the *AC Servomotors/Servo Drives 1S-series with Built-in EtherCAT® Communications and Safety Functionality User's Manual (Cat. No. I621)*.

**Precautions for Correct Use**

- If the safety task period changes due to changes in the safety program or other reasons, re-calculate the safety reaction times.
- To calculate the safety reaction times, add the "delaying influences from the input filter delay settings", the "safety program function block delay settings", and the "safety program loopback connections".

## Precaution for Conformance to ISO 13856-1:2013

If you use UM/UMA Safety Mats to build a pressure-sensitive protective device that conforms to ISO 13856-1:2013 (Safety of machinery -- Pressure-sensitive protective devices -- Part 1: General principles for design and testing of pressure-sensitive mats and pressure-sensitive floors), the NX-series Safety Control Unit must meet the following condition in order to satisfy the requirement for the specified reaction time.

- The value of the FSoE watchdog timer in the NX-SIH400 that is connected to the UM/UMA Safety Mats must be 42 ms or less.

## 8-1-2 Verifying Safety Reaction Times

Verify the calculated safety reaction times for all safety chains to confirm that they satisfy the required specifications.

If a calculated safety reaction time exceeds the required specifications, consider the following measures and correct the software or hardware design.

- Shorten the safety task period.
  Example: Reduce the size of the safety program.
  Reduce the number of Safety I/O Units.
- Reduce EPI values of CIP Safety connections.

# 8-2 Safety Task

This section describes the safety task of the Safety CPU Unit.
The safety task period of the Safety CPU Unit affects the safety reaction times.

## 8-2-1 Safety Task

The safety task is used to assign an execution condition to a series of processes, such as for data exchange with the Safety I/O Units and safety program execution.
The Safety CPU Unit executes one safety task.
The safety task is executed on a fixed period.

More than one program*1 can be assigned to a safety task. The programs that are assigned are executed in the order that they are assigned. Execution of all of the programs assigned to the task is called "program execution".

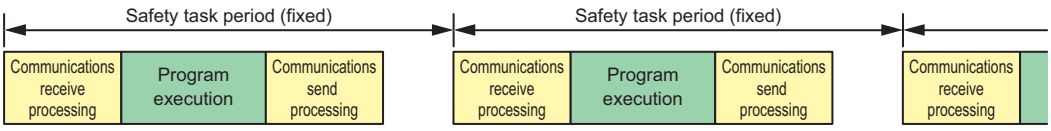
*1. There is no limit to the number of programs.

Data exchange between a Safety CPU Unit and safety slaves, including Safety I/O Units and safety I/O terminals, or between Safety CPU Units is called "communications receive processing" and "communications send processing".

| Type of task | Number of tasks | Task execution priority | Execution conditions | Main processing contents |
|---|---|---|---|---|
| Safety task | 1 | None | The safety task is executed once every safety task period during operation in RUN or DEBUG mode. | Communications receive processing, program execution, and communications send processing |

## 8-2-2 Operation of Safety Task

The following operation is performed for the safety task.
Input data processing for I/O refreshing, user program execution, and output data processing for I/O refreshing are performed repetitively, i.e., each safety task period.




The safety task period is the time interval at which the safety task is executed. The upper limit of the safety task period setting is 100 ms. A building error will occur for any safety program that requires a safety task period that is longer than 100 ms. If that occurs, change the safety program.

## 8-2-3 Minimum Safety Task Period

This section describes how to find the minimum safety task period.

## Safety Control System on CPU Ruck or EtherCAT Network

The minimum safety task period depends on the communications setup of the Slave Terminal.

### ● When DC Synchronization Is Set in All of the Communications Setups of the EtherCAT Slave Terminals to Which the Safety Control Units Are Connected

The minimum safety task period is displayed by the Sysmac Studio as the **Minimum safety task period**.

### ● When Free-Run Refreshing Is Set in Even One Communications Setup of the EtherCAT Slave Terminals to Which the Safety Control Units Are Connected

The minimum safety task period is calculated by adding a correction to the value displayed by the Sysmac Studio as the **Minimum safety task period**.

Use the following formula to find the value of the correction.

Correction = 11.4 ms − 2 x "EtherCAT communications cycle"

If the value of the correction is less than 0, use 0 as the correction.

**Precautions for Correct Use**

- If you change any of the following values, the minimum safety task period will change. Check the minimum safety task period again.
  a) Safety program execution time (This time depends on the sizes of the programs and the function blocks that are used.)
  b) Number of connections with Safety I/O Units
  c) Task period of the NJ/NX-series CPU Unit
  d) NX bus refresh cycle for EtherCAT Slave Terminals
- If the minimum safety task period calculated according to the process described in this section exceeds 100 ms, adjust the above values so that the minimum safety task period will be shorter than 100 ms.

**Version Information**

The minimum safety task period has been improved for combinations of NX-SIH400 version 1.1 or later and NX-SL□□□□ version 1.1 or later.
Therefore, if NX-SIH400 version 1.1 is replaced with NX-SIH400 version 1.0, the safety reaction time increases, resulting in an error.

## Safety Control System on EtherNet/IP Network

The minimum safety task period is displayed by the Sysmac Studio as the **Minimum safety task period**.

**Precautions for Correct Use**

If you change any of the following values, the minimum safety task period will change. Check the minimum safety task period again.

- Safety program execution time (This time depends on the sizes of the programs and the function blocks that are used.)
- Number of connections with FSoE slave devices
- Number of connections with CIP Safety devices

## 8-2-4 Setting the Safety Task Period

This section describes how to find an approximation of the value to set for the safety task period.

Use the following formula to find the safety task period.

Safety task period = "Minimum safety task period" + "Expansion communications monitoring time"

If the calculated value exceeds 100 ms, adjust the value to be less than 100 ms.

Refer to *6-5-9 Safety Task Settings* on page 6-75 for information on setting the safety task period in the Sysmac Studio.

### ● Expansion Communications Monitoring Time

The expansion communications monitoring time represents the additional time to wait for receptions in safety process data communications. You can set an expansion communications monitoring time to help prevent timeouts in safety process data communications when noise on the communications path or other factors result in communications retries.
Set this time as required.

### ● Finding the Expansion Communications Monitoring Time

The method to find the expansion communications monitoring time depends on the NX bus master that is connected.

- NX Units connected to a CPU Unit or EtherCAT Coupler Unit
  Set the expansion communications monitoring time to at least the length of the EtherCAT communications timeout value and to an integer multiple of the slave control period of the FSoE slave.
- NX Units connected to an EtherNet/IP Coupler Unit
  Set the expansion communications monitoring time to an integer multiple of the slave control period of the Safety I/O Units.

a. Slave Control Period of an FSoE Slave
   Use the following value for the slave control periods of the Safety I/O Units.
   Refer to the manuals for individual FSoE slaves for the slave control period of the FSoE slaves other than the Safety I/O Units.

| Model | Slave control period |
|---|---|
| NX-SIH400 | 6 ms |
| NX-SID800 | 3 ms |
| NX-SOH200 | 3 ms |
| NX-SOD400 | 3 ms |

b. Communications Timeout Value of an EtherCAT

For the EtherCAT communications timeout value, multiply the EtherCAT communications cycle time by (the "PDO communications timeout detection count" minus one) and use the resulting value.



**Precautions for Correct Use**

- To increase the communications timeout time, you much change both the safety task period and the value of the FSoE watchdog timer.
  Refer to *8-3-3 Settings for FSoE Watchdog Timers* on page 8-10 for information on changing the FSoE watchdog timer.
- If the EtherCAT communications cycle is 250 us or lower, use the following minimum safety task periods.
  NX-SL□□□□ version 1.1 or later: 9 ms or higher
  NX-SL□□□□ version 1.0 or later: 13 ms or higher



**Additional Information**

We recommend that you set the safety task period with plenty of leeway to allow for the possibility of expanding the safety control system or safety programs in the future.

# 8-3 FSoE Watchdog Timer

This section describes the FSoE watchdog timers.

## 8-3-1 FSoE Watchdog Timers

An FSoE watchdog timer is used for timeouts in safety process data between the Safety CPU Unit and Safety I/O Units.
The FSoE watchdog timers affect the safety reaction times.

You can increase the settings of the FSoE watchdog timers to help prevent timeouts in safety process data communications. Set this time as required.
To increase the communications timeout time, you must change both the value of the FSoE watchdog timer and the safety task period. Refer to *8-2-4 Setting the Safety Task Period* on page 8-7 for information on setting the safety task period.

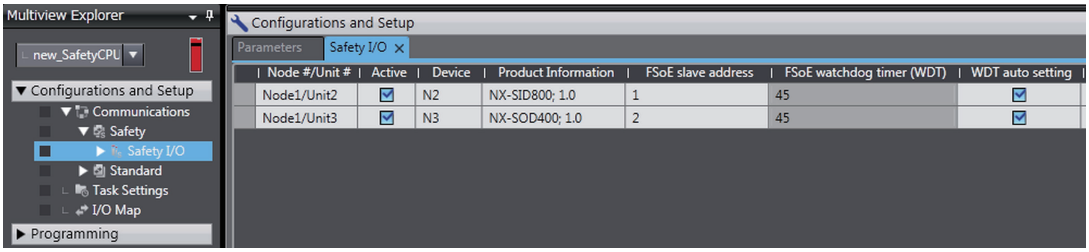
## 8-3-2 Checking FSoE Watchdog Timers

Use the following procedure to check the FSoE watchdog timers.

**1** In the Multiview Explorer, select the target Safety CPU Unit in the Controller Selection Box.

**2** Double-click **Safety I/O** under **Configurations and Setup** − **Communications** − **Safety**.

The following Safety I/O Unit Setting Tab Page is displayed.




The values of the FSoE watchdog timers are displayed in the **FSoE Watchdog Timer** Column for the Safety I/O Units.

### Precautions for Correct Use

If you use NX-SL□□□□ version 1.1 and Free-Run refreshing is set in even one communications setup of the EtherCAT Slave Terminals to which the Safety Control Units are connected, you must change the FSoE watchdog timer.
Set the FSoE watchdog timer to at least the sum of the **FSoE Watchdog Timer** displayed by the Sysmac Studio and the value of the following correction.

- When EtherCAT Communications Cycle Is Greater Than 1.9 ms

  Correction = *EtherCAT communications cycle*
- When EtherCAT Communications Cycle Is 1.9 ms or Lower

  Correction = 3.8ms - *EtherCAT communications cycle*

## 8-3-3 Settings for FSoE Watchdog Timers

This section describes how to find the set values for the FSoE watchdog timers.
The set value for an FSoE watchdog timer is different from the set value of the safety task period.

- **Safety Task Period < 100 ms**

  Set the FSoE watchdog timer to the same value as the expansion communications monitoring time that you added to the safety task period.

- **Safety Task Period = 100 ms**

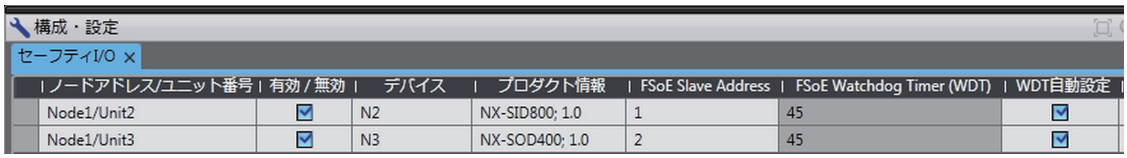
  Set the FSoE watchdog timer to a value that is an integer multiple of the safety task period.

## 8-3-4 Changing FSoE Watchdog Timers

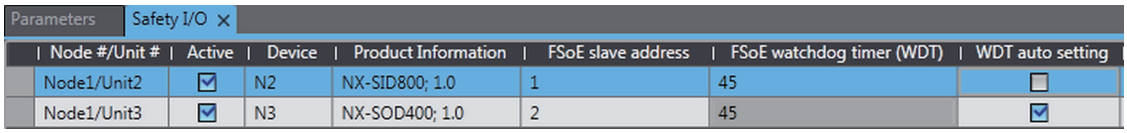
Use the following procedure to change a FSoE watchdog timer.

***1*** In the Multiview Explorer, select the target Safety CPU Unit in the Controller Selection Box.

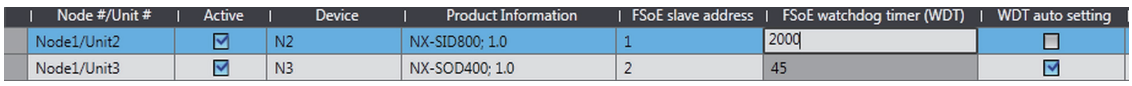
***2*** Double-click **Safety I/O** under **Configurations and Setup** – **Communications** – **Safety**.
The following Safety I/O Unit Setting Tab Page is displayed.




***3*** Clear the selection of the **WDT auto setting** Check Box for the Unit to change.
This enables changing the value of the **FSoE watchdog timer**.




***4*** Double-click the cell in the **FSoE watchdog timer** Column for the Unit to change and set the desired FSoE watchdog timer value.




**Version Information**

You can change the FSoE watchdog timers with Sysmac Studio version 1.10 or higher.

# 8-4 EPI (Data Packet Interval)

EPI stands for Expected Packet Interval and refers to the transmission interval of safety data packets in the CIP Safety communications.
The EPI affects the safety reaction time.

If you specify a smaller EPI, it shortens the network reaction time and the safety reaction time, but it increases the communications load on the EtherNet/IP communications port. For details on the communications load on the EtherNet/IP communications port, refer to *Section 9 Communications Load* on page 9-1.

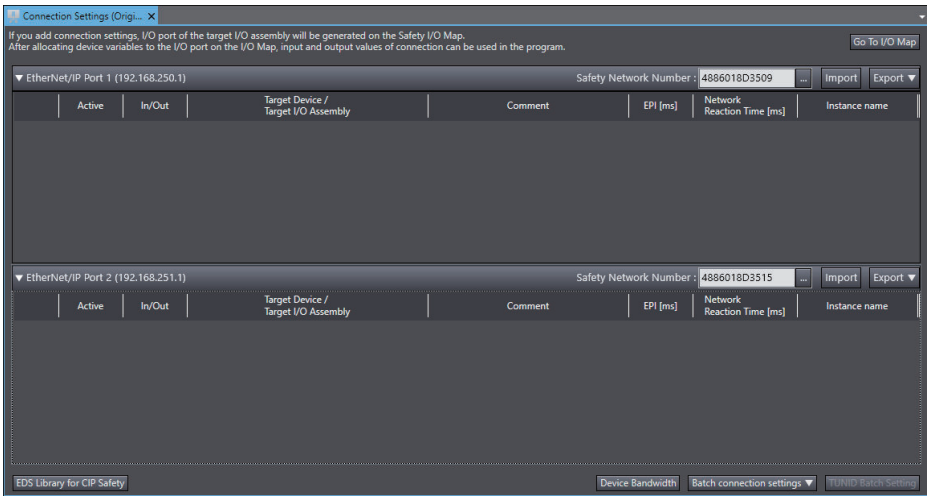
## 8-4-1 Changing the EPI

The EPI is set separately for each connection. Use the following procedure to change the EPI.

***1*** In the Multiview Explorer, select the target Safety CPU Unit in the Controller Selection Box.

***2*** Go to **Configurations and Setup** – **Communications** – **Safety** – **EtherNet/IP Safety Connection Settings** and double-click **Connection Settings (Originator)**.

The following Connection Settings (Originator) Tab Page is displayed.




***3*** Select a connection setting to change its EPI value and edit the **EPI** Column.

## 8-4-2 EPI Restrictions

The allowable range for EPI is automatically calculated and displayed by the Sysmac Studio.

# 9

# Communications Load

This section describes how to adjust communications load in order to realize high-speed and stable communications.


**9-1 Adjusting the Communications Load ..... 9-2**

9-1-1 Checking Bandwidth Usage for Tag Data Links ..... 9-4

9-1-2 Checking the Device Bandwidth Usage of the CIP Safety Routing ..... 9-6

9-1-3 Relationship between the Number of Packets Used per Second and Packet Intervals ..... 9-6

9-1-4 Adjusting the Device Bandwidth Usage ..... 9-7

# 9-1 Adjusting the Communications Load

In an Ethernet network using an Ethernet switch, the network bandwidth is not shared by all of the nodes, and independent transmission paths are established between individual nodes through the Ethernet switch.
A dedicated communications buffer is established in the Ethernet switch for communications between the nodes and full-duplex communications (simultaneous transmission and reception) are performed asynchronously with other transmission paths. The communications load in other transmission paths does not affect communications, therefore packet collisions do not occur, and stable high-speed communications can be performed.
The Ethernet switch functions shown in the following table determine the performance of the CIP Safety connections and tag data links.

| Item | Description |
|---|---|
| Buffer capacity | This is the amount of data that can be buffered when the data packets accumulate at the Ethernet switch. |
| Multi-cast filtering | The function transfers multi-cast packets to specific nodes only. |
| QoS function | The function performs priority control of packet transfers. |

The following table shows the specifications within which the CIP Safety Connection settings and the tag data link settings can be made for the built-in EtherNet/IP port. In the NX102 CPU Unit, both CIP safety communications and tag data link communications can be used on the same port.

| Item | Meaning | NX-series CPU Unit | |
|---|---|---|---|
| | | NX502 | NX102 |
| Network bandwidth | Physical Ethernet baud rate | 1 Gbps, 100 Mbps, or 10 Mbps | 100 Mbps or 10 Mbps |
| Maximum number of packets per second | The maximum number of packets that can be processed in one second (pps: packet per second) | 20,000 pps max. (total of 20,000 pps with two ports) | 12,000 pps max. (total of 12,000 pps with two ports) |
| Number of CIP Safety routing connections | Maximum safety I/O connections supported for routing | 128 max. (total of 128 with two ports)*1*2 | 16 max. (total of 16 with two ports) |
| Packet interval of CIP Safety connections (EPI: Expected Packet Interval) | Data refresh period of CIP Safety connections | The allowable range depends on the target device. *1*2 | |
| Tag data link connection resources | The number of tag data link connections that can be opened | 64 max. (total of 128 with two ports) | 32 max. (total of 64 with two ports) |

| Item | Meaning | NX-series CPU Unit | |
|---|---|---|---|
| | | NX502 | NX102 |
| Packet interval of tag data links (RPI: Requested Packet Interval) | Refresh period for tag data | 1 to 10,000 ms in 1-ms increments | 1 to 10,000 ms in 1-ms increments |

*1. For the NX502 CPU Units, CIP Safety communications via the built-in EtherNet/IP port can be used with the NX502 CPU Unit with unit version 1.64 or later. If you set CIP Safety communications with the NX502 CPU Unit with unit version earlier than 1.64, use an NX-series EtherNet/IP Unit.

*2. When the NX502 CPU Unit with unit version 1.66 or later is used with the NX-series EtherNet/IP Unit with unit version 1.01 or later, you can use tag data link and CIP Safety communications together in one NX-series EtherNet/IP Unit.

The following table shows the specifications within which the CIP Safety connection settings and the tag data link settings can be made for the EtherNet/IP port on the NX-series EtherNet/IP Unit.

| Item | Meaning | NX-series EtherNet/IP Unit |
|---|---|---|
| | | NX-EIP201 |
| Network bandwidth | Physical Ethernet baud rate | 1 Gbps, 100 Mbps, or 10 Mbps |
| Maximum number of packets per second | The maximum number of packets that can be processed in one second (pps: packet per second) | When you use the NX-series EtherNet/IP Unit with unit version earlier than 1.01, or when you do not combine CIP Safety communications with tag data link in the NX-series EtherNet/IP Unit with unit version 1.01 or later: 40000 pps max. (total of 40000 pps with two ports)<br>When you use CIP Safety communications and tag data link together in the NX-series EtherNet/IP Unit with unit version 1.01 or later: 6,000 pps max. (total of 6,000 pps with two ports)*1 |
| Number of CIP Safety routing connections | Maximum safety I/O connections supported for routing | When you use the NX-series EtherNet/IP Unit with unit version earlier than 1.01, or when you do not combine CIP Safety communications with tag data link in the NX-series EtherNet/IP Unit with unit version 1.01 or later: 84 max. (total of 84 with two ports)<br>When you combine CIP Safety communications with tag data link in the NX-series EtherNet/IP Unit with unit version 1.01 or later: 64 max. (total of 64 with two ports) |
| Packet interval of CIP Safety connections (EPI: Expected Packet Interval) | Data refresh period of CIP Safety connections | 1 to 500 ms in 1-ms increments. The allowable range depends on the target device. |
| Tag data link connection resources | The number of tag data link connections that can be opened | 256 max. (total of 512 with two ports) |

| Item | Meaning | NX-series EtherNet/IP Unit |
|---|---|---|
| | | NX-EIP201 |
| Packet interval of tag data links (RPI: Requested Packet Interval) | Refresh period for tag data | When you use the NX-series EtherNet/IP Unit with unit version earlier than 1.01, or when you do not combine CIP Safety communications with tag data link in the NX-series EtherNet/IP Unit with unit version 1.01 or later: 1 to 10000 ms<br><br>When you combine CIP Safety communications with tag data link in the NX-series EtherNet/IP Unit with unit version 1.01 or later: 1 to 100 ms |

*1. When performing tag data link communications and CIP Safety communications where the allowed communications bandwidth per Unit is 6,000 pps or more, the following functions may not be used properly. To use those functions, use the built-in EtherNet/IP port on the CPU Unit or an EtherNet/IP port of a different NX-series EtherNet/IP Unit.
- Connecting to the Sysmac Studio online from the Ethernet/IP port of the NX-series EtherNet/IP Unit
- Connecting to the Network Configurator online from the EtherNet/IP port of the NX-series EtherNet/IP Unit
- Connecting to the NA-series HMI from the Ethernet/IP port of the NX-series EtherNet/IP Unit
- Port forward function via the NX-series EtherNet/IP Unit
- CIP message communications
- SNMP function

It is possible to use these functions in the relevant NX-series EtherNet/IP Unit from the built-in EtherNet/IP port on the CPU Unit or from the Ethernet/IP port of another NX-series Ethernet/IP Unit via the X Bus.

When the CIP Safety connection settings or the tag data link settings exceed the capabilities of the Ethernet switch, adjust (increase) the packet interval value (RPI/EPI).
Particularly when using an Ethernet switch that does not support multi-cast filtering, the settings must be made considering that multi-cast packets will be sent even to nodes without connection settings.

**Additional Information**

If the connection type is set to **Multi-cast connection** in the connection settings of the CIP Safety or the tag data links, multi-cast packets are used. If the connection type is set to **Point to Point connection**, multi-cast packets are not used.

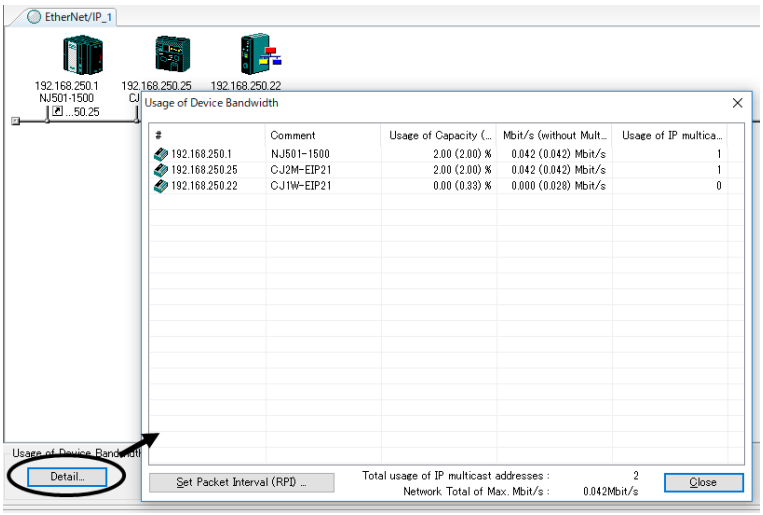
In addition, if the required CIP Safety performance or tag data link performance cannot be achieved within the specifications, reevaluate the overall network configuration and correct it by taking steps such as selecting a different Ethernet switch or splitting the network.
The following sections show how to check the device bandwidth to be used for the CIP Safety routings and the tag data links in the designed network, and how to change the values.

**Additional Information**

In order to provide stable communications, the connection settings need to be adjusted so that the total device bandwidth usage of tag data links and the CIP Safety routing will not exceed 80%.

## 9-1-1 Checking Bandwidth Usage for Tag Data Links

The Network Configurator can display the bandwidth actually used for tag data links at each built-in EtherNet/IP port, based on the connections set in the network configuration.
The device bandwidth used by tag data links can be checked by clicking the **Detail** Button in the **Usage of Device Bandwidth** Area at the bottom of the Network Configuration Pane.




| Item | Description |
|---|---|
| # | The IP address of the device |
| Comment | A description of the device. The comment is displayed below the device icon. The model number of the device is displayed by default. |
| Usage of Capacity (without Multicast filter) | The ratio of the device's packet usage to the maximum number of packets per second.<br>*Number of packets used per second* / *Number of maximum packets per second*<br>The values outside parentheses are for when multi-cast filtering is used.<br>The values inside parentheses are for when multi-cast filtering is not used. |
| Mbit/s (without Multicast filter) | The network bandwidth usage for tag data link communications with the device.<br>The values outside parentheses are for when multi-cast filtering is used.<br>The values inside parentheses are for when multi-cast filtering is not used. |
| Usage of IP multicast addresses | The number of IP multi-cast addresses actually used for communications with the device. |
| Total usage of IP multicast addresses | The number of IP multi-cast addresses used in the entire network. This value is used to estimate the number of multi-cast filters for switching. |
| Network Total of Max. Mbit/s | The total network bandwidth used for tag data link communications in the entire network.<br>Tag data links will not operate normally if the network bandwidth that can be set is exceeded. |

## ● Checking the Packet Usage Rate to the Maximum Number of Packets per Second and the Network Bandwidth Usage

The window displays the ratio of the packet usage to the maximum number of packets per second for each EtherNet/IP port, and the network bandwidth usage in Mbit/s.

The usage of capacity and used network bandwidth that are displayed in parentheses are for an Ethernet switch that does not use multicast filtering. In this case, multicast packets will be sent to even the nodes without connection settings, so the displayed values will include these packets as well.

These values can be adjusted by changing the RPI.

### ● Checking the Total Number of Multi-cast IP Addresses in the Network

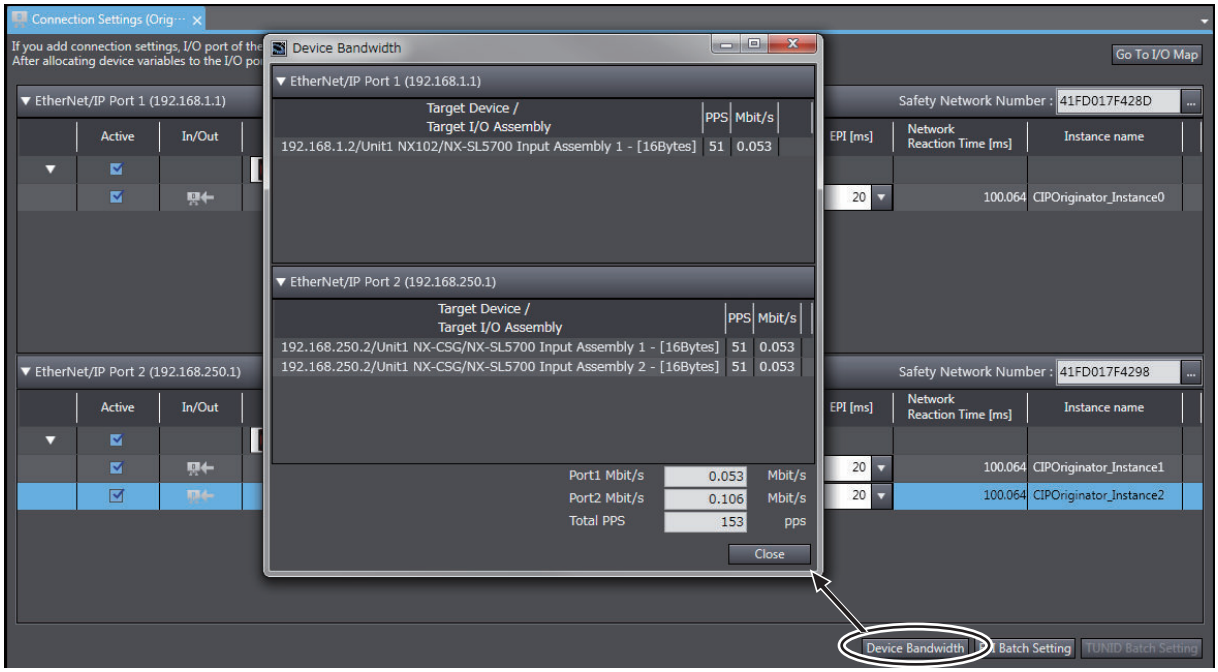
When using an Ethernet switch that provides multicast filtering, there must be enough multicast filters for the network to be used. The Network Configurator shows the number of multi-cast IP addresses used for the entire network based on the connection settings.

Make sure that the number of multicast IP addresses used in the entire network does not exceed the number of multicast filters supported by the Ethernet switch. If necessary, replace the Ethernet switch with another one that has the sufficient number of multi-cast filters, or adjust the usage rate to the maximum number of packets per second or the network bandwidth usage by using values given for Ethernet switches without multicast filtering (i.e., the values in parentheses). These values can be adjusted by changing the RPI.

## 9-1-2 Checking the Device Bandwidth Usage of the CIP Safety Routing

The Sysmac Studio displays the bandwidth used for the CIP Safety routing function for each CIP Safety connection configured on in the Safety CPU Unit.

You can verify the device bandwidth usage status for CIP Safety routing by clicking the **Device Bandwidth** Button located at the bottom of the Connection Settings (Originator) Tab Page.




| Item | Meaning |
|---|---|
| PPS | Displays the number of packets used for each CIP Safety connection per second and the total sum of used packets |
| Mbit/s | Displays the network bandwidth used for each CIP Safety connection. |

## 9-1-3 Relationship between the Number of Packets Used per Second and Packet Intervals

The usage rate to the maximum number of packets per second can be adjusted by modifying the settings for the packet interval (PRI) for the tag data link or the packet interval (EPI) for the CIP safety connection.

The shorter the packet interval is, the larger the usage rate to the maximum number of packets per second will become.
Conversely, the longer the packet interval is, the less the usage rate to the maximum number of packets per second will become.

## 9-1-4 Adjusting the Device Bandwidth Usage

This section describes how to adjust the device bandwidth usage.

**Precautions for Correct Use**

The Ethernet switch should support the maximum network bandwidth value that can be set for the NX502 CPU Unit or NX102 CPU Unit. The maximum network bandwidth value of the NX502 CPU Unit is 1 Gbit/s, and the maximum network bandwidth value of the NX102 CPU Unit is 100 Mbit/s.

### Ethernet Switches without Multicast Filtering

- Does the total number of packets transmitted to the built-in EtherNet/IP port per second exceed the maximum number of packets allowed per second?
  If it exceeds the maximum number, check and modify the connection settings, including RPI and EPI values.
- Does the total network bandwidth usage of any transmission path exceed the network bandwidth that can be set for the path?
  If it exceed the bandwidth that can be set for the transmission path, the tag data link and CIP safety connection may not work properly. If the bandwidth is exceeded, rewire the network or increase the bandwidth between Ethernet switches (e.g., to 1 Gbps). If these countermeasures are not possible, change the connection settings, including the RPI settings.

### Ethernet Switches with Multicast Filtering

- Does the total network bandwidth usage of any transmission path exceed the network bandwidth that can be set for the path?
  If it exceed the bandwidth that can be set for the transmission path, the tag data link and CIP safety connection may not work properly. If the bandwidth is exceeded, rewire the network or increase the bandwidth between Ethernet switches (e.g., to 1 Gbps). If these countermeasures are not possible, change the connection settings, e.g., the RPI settings.
- Does the total number of packets transmitted to the EtherNet/IP port per second exceed the maximum number of packets allowed per second?
  If it exceeds the maximum number, check whether the multicast filtering on Ethernet switches is functioning correctly. Calculate the number of multicast filters required for each Ethernet switch on the network, and make sure that the resulting number does not exceed the number of multicast filters of any Ethernet switch. If the number of multicast filters of an Ethernet switch is not sufficient, replace it with another switch with enough multicast filters, or modify the connection settings, including the RPI and EPI settings.

# 10



# Safety Unit Restore

This section describes the procedures to execute the Safety Unit Restore.

**10-1 Safety Unit Restore ..... 10-2**
10-1-1 Generate Safety Unit Restore File Function ..... 10-2
10-1-2 Safety Unit Restore Function ..... 10-4
10-1-3 Specifications of a Safety Unit Restore File ..... 10-6
**10-2 Versions for Which Safety Unit Restore Can Be Used ..... 10-7**

# 10-1 Safety Unit Restore

The safety unit restore is a function designed to transfer safety programs and settings to Safety CPU Unit via an SD Memory Card. Refer to *10-2 Versions for Which Safety Unit Restore Can Be Used* on page 10-7 for information on the Unit models and unit versions that can use this function, and corresponding Sysmac Studio versions.
You can use the Safety Unit Restore in the following instances.

| Item | Application |
|---|---|
| Program and setting changes | When you change the safety program and settings for equipment that is currently in operation |
| Hardware replacement | When you replace the hardware for the Safety CPU Unit |
| Manufacture of equipment | When you want to manufacture the same equipment and need to transfer the data from the existing equipment to new equipment in its initial state. |

Safety Unit Restore is executed combining the following functions.

| Function | Description |
|---|---|
| Generate Safety Unit Restore File function | This function allows to generate the Safety Unit Restore File using the Sysmac Studio. |
| Safety Unit Restore function | This function allows to transfer the Safety Unit Restore File data stored in an SD Memory Card to a Safety CPU Unit. |

Note that the Safety Unit Restore function restores only the target data in NX-SL5□□□ Units. To restore the entire system, you need to execute this function together with the SD Memory Card backup function.

**Precautions for Correct Use**

- To execute the Safety Unit Restore, you need to execute the restore function for the NX502 CPU Unit or NX102 CPU Unit as well. In addition, you need to generate a Safety Unit Restore File and a backup file of the NX502 CPU Unit or NX102 CPU Unit from a same project or from a same physical unit from which a project is transferred. If the settings for the Safety Unit Restore File and that for the backup file of the NX502 CPU Unit or NX102 CPU Unit are not consistent, the Safety Control Unit does not operate normally.
- Before executing the Safety Unit Restore, make sure to confirm the safety of the transfer destination.
- Before executing the Safety Unit Restore, verify that the operation target is correct.
- Before executing the Safety Unit Restore, verify that the signature displayed on the seven-segment indicator of the Safety CPU Unit is correct.
- To prevent accessing a wrong Safety Unit Restore File, make sure to control the file access and configuration properly.
- After executing the Safety Unit Restore, verify that the Unit is configured correctly and the Unit behaves as intended.
- To prevent executing the Safety Unit Restore by unauthorized person, make sure to keep under access control to SD Memory Cards and Safety Unit Restore Files.

## 10-1-1 Generate Safety Unit Restore File Function

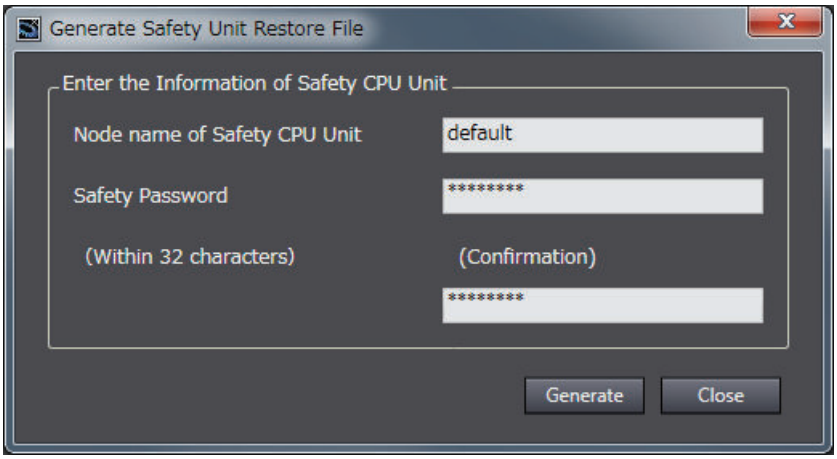
This function uses the Sysmac Studio to generate a Safety Unit Restore File from a project that includes safety application data.

Safety validation must be completed for the safety application data. Refer to *7-11 Performing Safety Validation and Operation* on page 7-65 for details on the safety validation of safety application data.

## Procedure

**1** Startup the Sysmac Studio. Open a project which contains the validated safety application data.

**2** From the Controller selection in the Multiview Explorer, select Safety CPU Unit, then select **Tools** – **Generate Safety Unit Restore File** from the menu.
The Generate Safety Unit Restore File Dialog Box is displayed.




**3** Enter the node name and the safety password that are set for the Safety CPU Unit to restore, and click the **Generate** Button.
The Browse Folder Dialog is displayed.

| Item | Initial value | Description |
|---|---|---|
| Node Name | default | Enter a node name for Safety CPU Unit to restore data. If the node name does not match with the actual unit, the restore operation fails. If the node name of the actual unit is unchanged from the factory default settings, the node name you entered here will be reflected to the actual unit. |
| Safety Password | --- | Enter a safety password for Safety CPU Unit to restore data. If the safety password does not match, the restore operation fails. If the safety password is not configured for the actual unit, the safety password you entered here will be reflected to the actual unit. |

**4** Specify the folder to save the files and click the **OK** Button.
The Safety Unit Restore File (file name: SLSystem.dat) will be generated to the specified folder.

**5** Store the Safety Unit Restore File in the root directory on the SD Memory Card.

## 10-1-2 Safety Unit Restore Function

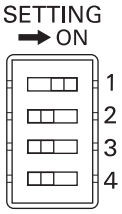
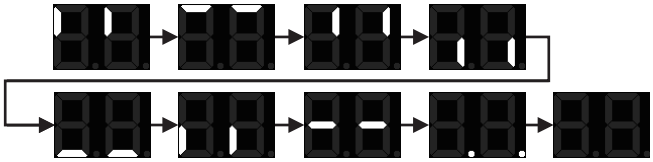
Insert an SD Memory Card to the NX502 CPU Unit or NX102 CPU Unit connected to the Safety CPU Unit and then transfer data of the Safety Unit Restore File stored in the memory card to the Safety CPU Unit.

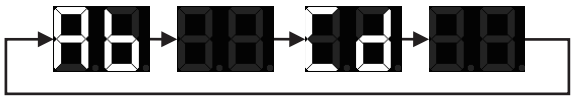
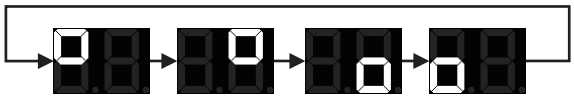
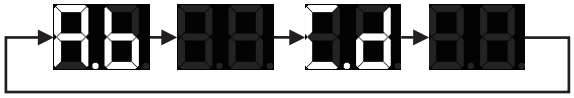
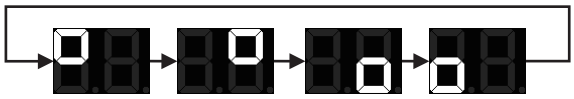
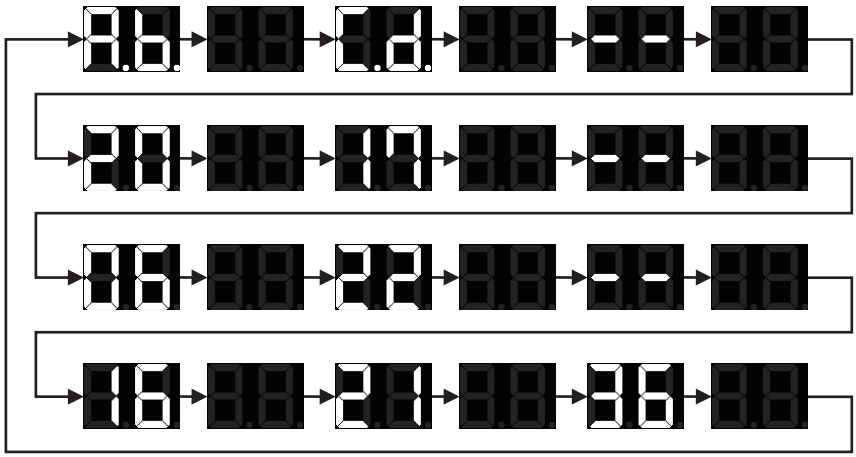
Change the DIP switch settings and start the Safety CPU Unit in the Restore mode then execute the Safety Unit Restore using the SD Memory Card.
The node name and safety password stored in the Safety CPU Unit to restore must be consistent with those stored in the Safety Unit Restore File or unchanged from the factory default settings.

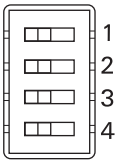
**Precautions for Correct Use**

To execute the Safety Unit Restore, you need to execute the restore function for the NX502 CPU Unit or NX102 CPU Unit as well. In addition, you need to generate a Safety Unit Restore File and a backup file of the NX502 CPU Unit or NX102 CPU Unit from a same project or from a same physical unit from which a project is transferred. If the settings for the Safety Unit Restore File and that for the backup file of the NX502 CPU Unit or NX102 CPU Unit are not consistent, the Safety Control Unit does not operate normally.

### Procedure

| Processing stage | Procedure and Display |
|---|---|
| Insert an SD Memory Card | Insert the SD Memory Card where the Safety Unit Restore File is stored under the root directory into the NX502 CPU Unit or NX102 CPU Unit. |
| Start RESTORE Mode | Set the pins 1 to 4 of the DIP switch on the Safety CPU Unit as follows: 1 to ON, 2 to OFF, 3 to OFF, and 4 to OFF, and turn ON the power supply to the Controller. The Safety CPU Unit starts in RESTORE mode.<br>  |
| Initializing | Seven-segment indicators in the Safety CPU Unit repeat turning ON and OFF in sequence for each, to test if the devices are lit properly.<br>  <br>• If initialization ended in an error, an error code is shown in the seven-segment indicators in the Safety CPU Unit. |

| Processing stage | Procedure and Display |
|---|---|
| Wait for Start command | The safety signature of the Safety Unit Restore File stored in the SD Memory Card is repeatedly shown as a four-digit hexadecimal number in the seven-segment indicators in the Safety CPU Unit.<br>  <br>(Example: Supposing the safety signature is 0xABCD)<br>Check the safety signature. If it is correct, press and hold the service switch for one second or more and release.<br>Processing starts. |
| Processing | Seven-segment indicators in the Safety CPU Unit repeat turning ON and OFF in four at a time.<br>  <br>• If processing ended in an error, an error code is shown in the seven-segment indicators in the Safety CPU Unit. |
| Wait for Completion Command | The safety signature for the settings information transferred to the Safety CPU Unit is repeatedly shown as a four-digit hexadecimal number in the seven-segment indicators of the Safety CPU Unit.<br>  <br>(Example: Supposing the safety signature is 0xABCD)<br>Check the safety signature. If it is correct, press and hold the service switch for one second or more and release.<br>Completion processes starts. |
| Processing Completion | Seven-segment indicators in the Safety CPU Unit repeat turning ON and OFF in four at a time.<br>  <br>• If processing ended in an error, an error code is shown in the seven-segment indicators in the Safety CPU Unit. |
| Done | The safety signature including the date and time (UTC) is repeatedly shown in the seven-segment indicators of the Safety CPU Unit.<br>  <br>(Example: Supposing the safety signature is 0xABCD, and the date is 16:21:36 of June 22, 2017 (UTC)) |

| Processing stage | Procedure and Display |
|---|---|
| Restart | After turning OFF the power supply to the Controller, set the pins 1 to 4 of the DIP switch on the Safety CPU Unit as follows: 1 to OFF, 2 to OFF, 3 to OFF, and 4 to OFF, and turn ON the power supply to the Controller. The Safety CPU Unit starts in RUN mode.<br>SETTING<br>➡ ON<br>  <br> |

Refer to *Section 12 Troubleshooting* on page 12-1 for errors that can occur while restoring the Safety Control Units.

## 10-1-3 Specifications of a Safety Unit Restore File

A Safety Unit Restore File is named as follows:

### ● File Name

| File | File name |
|---|---|
| Safety Unit Restore File | SLSystem.dat |

# 10-2 Versions for Which Safety Unit Restore Can Be Used

This section describes combinations of unit models and unit versions that can use Safety Unit Restore, and corresponding Sysmac Studio versions.

You can use the following version or later/higher.

| Safety Control Unit | | CPU Unit | | Sysmac Studio version |
|---|---|---|---|---|
| Model | Version | Model | Version | |
| NX-SL5□□□ | Ver.1.3 | NX502 | Ver.1.60 | Ver.1.54 |
| | | NX102 | Ver.1.31 | Ver.1.24 |

# 11



# Safety Data Logging

This section describes the Safety Data Logging function.

**11-1 Outline of the Safety Data Logging Function ..... 11-2**

**11-2 Creating a Safety Data Logging Settings File with the Sysmac Studio ..... 11-4**

**11-3 Safety Data Logging Operation Procedure ..... 11-6**

**11-4 Checking the Logging Status ..... 11-7**

11-4-1 Checking the Seven-segment Indicator ..... 11-7

11-4-2 Checking with System-defined Variables ..... 11-7

**11-5 Log File Specifications ..... 11-8**

**11-6 Versions for Which Safety Data Logging Can Be Used ..... 11-10**

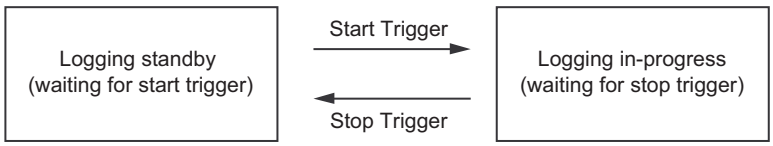
# 11-1 Outline of the Safety Data Logging Function

The safety data logging is a function that records exposed variables and device variables used in the safety program of the Safety CPU Unit in a chronological order. The function logs the specified variables in the volatile memory and saves the log data into the SD Memory Card before and after the trigger condition is met. Refer to *11-6 Versions for Which Safety Data Logging Can Be Used* on page 11-10 for information on the Unit models and unit versions that can use this function, and corresponding Sysmac Studio versions.

This section provides the specifications of the safety data logging.

| Item | Specifications |
|---|---|
| Number of logging settings | 2 max.<br>Specify the respective settings including sampling interval, trigger conditions, and logging target. |
| Number of records | 3000 records per setting |
| Sampling interval | Based on the task period of the NX502 CPU Unit or NX102 CPU Unit, select one of the four options specified by the Sysmac Studio for each setting. |
| Trigger condition | One variable can be specified as a trigger condition for each setting.<br>BOOL variables: True or False<br>Non-BOOL variables: Select any of the following and specify a constant:<br>=: Equal to the constant<br>≠: Not equal to the constant<br>>: Greater than the constant<br><: Less than the constant<br>≥: Equal to or greater than the constant<br>≤: Equal to or less than the constant |
| Logging target | Up to 100 exposed variables and devices variables can be specified for each setting. |
| Storage location of the safety data logging settings file and log files | The data is saved in the following folder of the SD Memory Card that is mounted in the NX502 CPU Unit or NX102 CPU Unit.<br>/SFLog/ |

The safety data logging function works as follows:




| Status | Operation |
|---|---|
| Logging standby (waiting for start trigger) | The safety data logging is not in progress. When the start trigger is detected, it enters the "logging in-progress" state. |

| Status | Operation |
|---|---|
| Logging in-progress (waiting for stop trigger) | The safety data logging is in progress. The log data is constantly recorded in the ring buffer of the volatile memory. When the buffer becomes full, the oldest log is discarded in order.<br>When the unit enters the logging standby state with the stop trigger, logs in the buffer are saved into the SD Memory Card. |

## ● Start Trigger

When the following condition is met in the logging standby state, the start trigger is activated and the unit enters into the logging in-progress state.

- The service switch is pressed for operation after all the safety master connections were established.

However, if the logging settings file is configured to start logging at a startup, you do not need to press the service switch to start the logging in-progress state for the first logging.

## ● Stop Trigger

The stop trigger is activated by the following factors:

| Stop factors | Logging to be stopped | Log files to be saved |
|---|---|---|
| When the trigger condition specified in a safety data logging settings file is met | Stop only the logging for which the trigger condition is met | Save only log files recorded for the logging setting for which the trigger condition is met. |
| When the stop operation is performed with the service switch | Stop all the loggings in progress | Save log files for all the logging settings in progress. |
| When a communication error occurs in the safety master connections (FSoE master connection or CIP Safety originator connection) | Stop all the loggings in progress | Save log files for all the logging settings in progress. |
| When the operation to stop the FSoE or CIP Safety communications is performed<br>• When the operating mode of the Safety CPU Unit changes<br>• The setting is transferred to the NX502 CPU Unit or NX102 CPU Unit<br>• The NX bus restart is executed,<br>etc. | Stop all the loggings in progress | Save log files for all the logging settings in progress. |

When the unit enters the logging standby state with the stop trigger, logs into the buffer are saved into the SD Memory Card.



**Precautions for Correct Use**

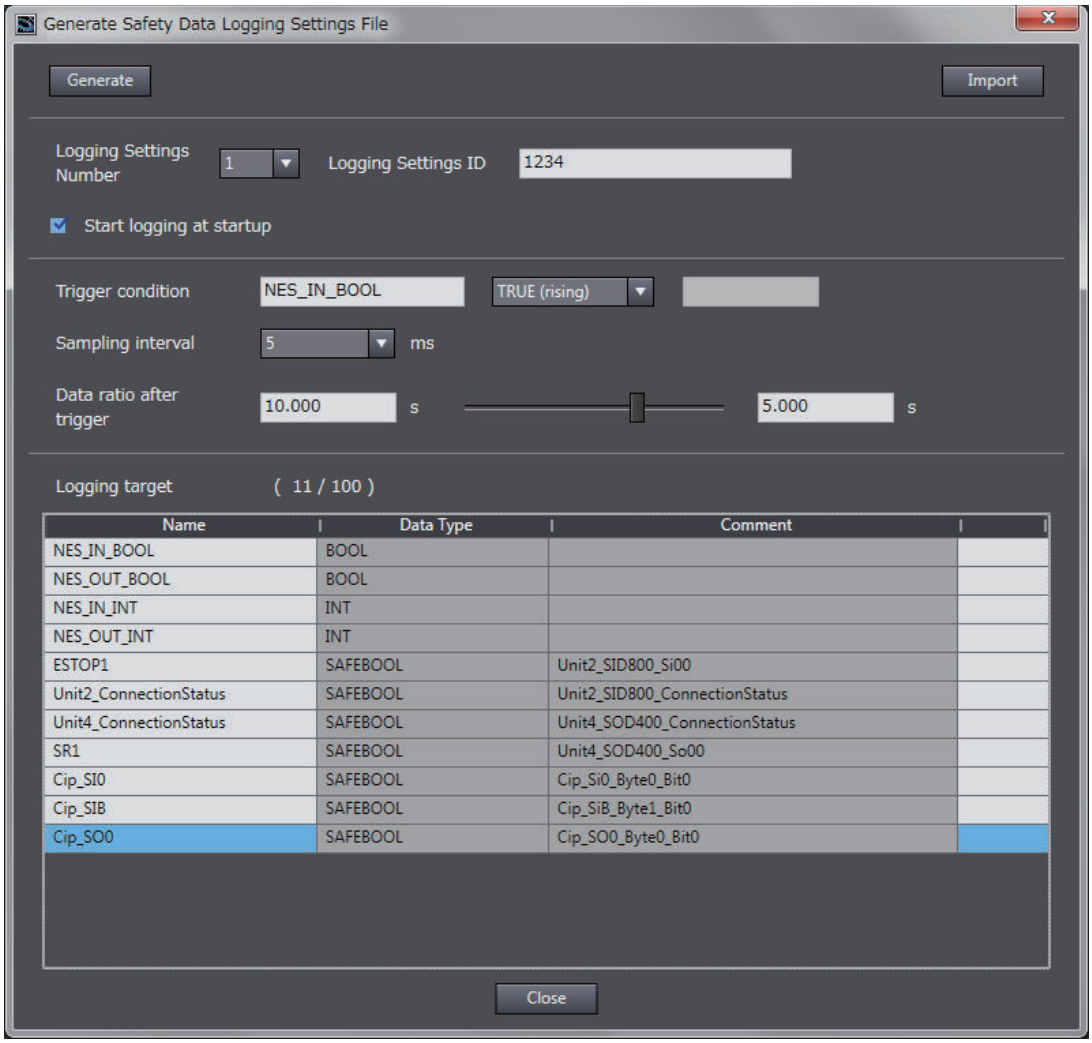
Do not turn OFF the power supply to the NX502 CPU Unit or NX102 CPU Unit while data is being transferred. Otherwise, the files may be damaged.

# 11-2 Creating a Safety Data Logging Settings File with the Sysmac Studio

The Sysmac Studio allows you to create a safety data logging settings file.
Safety validation must be completed for the safety application data. Refer to *7-11 Performing Safety Validation and Operation* on page 7-65 for details on the safety validation of safety application data.

Use the following procedure to create a safety data logging settings file.

**1** From the Controller selection in the Multiview Explorer, select the Safety CPU Unit and then go to **Tools** – **Generate Safety Data Logging Settings File**.
The **Generate Safety Data Logging Settings File** Window shown below appears.




The structure of the **Safety Data Logging Settings File** Window is as follows:

| Item | Description |
|---|---|
| Logging Settings Number | Select 1 or 2.<br>Up to two safety data logging settings files can be saved in the SD Memory Card. The logging settings ID for each file must be unique. |

| Item | Description |
|---|---|
| Logging Settings ID | An ID number is specified to associate a logging settings file with relevant log files as its logging result. This ID number is used as part of their file names so that the safety data logging settings file and the corresponding log files can be identified. |
| Start logging at startup | Selected: The safety data logging is executed at a startup without pressing the service switch.<br>Not selected: The service switch operation is required for executing the safety data logging. |
| Trigger condition | Specify a variable and a condition expression as the trigger to stop the safety data logging.<br>Left box: Specify a variable name.<br>Right box: Specify a constant value to compare.<br>List in the middle: Select from the following.<br>For BOOL: Select either True or False.<br>For other than BOOL: Select one of the following options:<br>=: Equal to the constant<br>≠: Not equal to the constant<br>>: Greater than the constant<br><: Less than the constant<br>≥: Equal to or greater than the constant<br>≤: Equal to or less than the constant |
| Sampling interval | Select a sampling interval. |
| Data ratio after trigger | Specify a ratio of the log data after the trigger to that before the trigger. |
| Logging target | Up to 100 variables can be registered for logging.<br>Name: Specify the name of the variable is specified.<br>Data Type: The data type of the variable is displayed.<br>Comment: The comment for the variable is displayed. |
| **Generate** Button | This button allows you to save a safety data logging settings file into a folder you specify. |
| **Import** Button | This button allows you to import a safety data logging settings file. |

***2*** Specify the settings and click the **Generate** Button.

When you specify a save destination in the displayed dialog box and click the **OK** Button, the following file will be generated.

| File name*1 | Application |
|---|---|
| SFLogConfig_<N>_<ID>_<SIGNATURE>.dat | Safety data logging settings file. This file is used for the safety data logging function. |
| SFLogConfig_<N>_<ID>_<SIGNATURE>.txt | Confirmation file for safety data logging settings. A text file in which the settings are visualized. This file is not used for the safety data logging function. |

*1. The meaning of the extensions used for file names is as follows:
N: Logging Settings Number
ID: Logging Settings ID
SIGNATURE: Safety Signature

***3*** Store the safety data logging settings file in /SFLog/ on the SD Memory Card.

# 11-3 Safety Data Logging Operation Procedure

Use the following procedure to execute the safety data logging function:

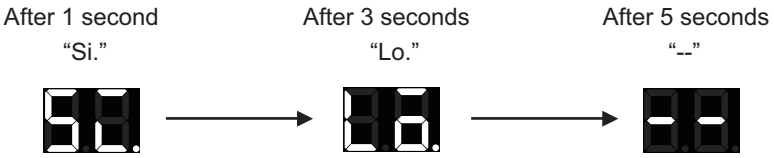
***1*** Find the SD Memory Card that contains the safety data logging settings file in the specified folder. Insert the SD Memory Card into the SD Card Slot of the NX502 CPU Unit or NX102 CPU Unit to which the Safety CPU Unit is connected.

***2*** Set the DIP switch on the front panel of the Safety CPU Unit to turn ON SW2, and OFF SW1, SW3 and SW4.

***3*** Start or restart the NX502 CPU Unit, NX102 CPU Unit, and Safety CPU Unit.

***4*** If the logging settings file is configured to start logging at a startup, the logging execution starts when all the safety master connections are established.
If the logging settings file is not configured to start logging at a startup, press the service switch on the front panel of the Safety CPU Unit after confirming that all the safety master connections are established.

If you hold down the service switch, the display of the seven-segment indicator will change as shown below.




Release the service switch while the indicator shows "Lo".
If you release the service switch while "--" is displayed, the function is not executed and the original state is restored.

***5*** Check the left dot of the seven-segment indicator on the Safety CPU Unit to see if the logging execution started normally.
If it starts normally, the left dot is lit. If it fails, the left dot is flashing.

***6*** Check the left dot of the seven-segment indicator on the Safety CPU Unit to see if the trigger condition was met and the log files were output.
When output of all the log files is completed, the left dot goes out and the right dot lights up.
If the log files are not saved successfully, the right dot starts flashing.

By following Step 4, you can manually stop the logging that is in progress.

***7*** Remove the SD Memory Card.

# 11-4 Checking the Logging Status

You can verify the logging status in the following methods:

- Checking the seven-segment indicator of the Safety CPU Unit.
- Checking with system-defined variables

## 11-4-1 Checking the Seven-segment Indicator

The logging status is displayed with the right and left dots of the seven-segment indicator of the Safety CPU Unit as shown below.

| Display area | Display | Description |
|---|---|---|
| Left dot of the seven-segment indicator of the Safety CPU Unit | Not lit | No logging is in progress. |
| | Lit | Logging is in progress. |
| | Flashing (0.5-s interval) | Logging did not start. |
| Right dot of the seven-segment indicator of the Safety CPU Unit | Not lit | No log file was output. |
| | Lit | A log file was saved successfully. |
| | Flashing (0.5-s interval) | A log file was not saved successfully. |

## 11-4-2 Checking with System-defined Variables

You can verify the logging execution status by checking the system-defined variables of the NX502 CPU Unit or NX102 CPU Unit.

The NX502 CPU Unit and NX102 CPU Unit have system-defined variables for each logging setting. For details on the system-defined variables, refer to *System-defined Variables* in the *NJ/NX-series CPU Unit Software User's Manual (Cat. No. W501)*.

# 11-5 Log File Specifications

Log files are stored in the SD Memory Card.

A log file is saved as a comma-delimited text file with a csv extension.
Log file specifications are described below.

| Item | Specifications |
|---|---|
| Log file name*1 *2 | SFLog_<N>_<ID>_<SIGNATURE>_<Date of output>_<Time of output>.csv |
| File location | /SFLog/ |

*1. N, ID, and SIGNATURE in a log file name are each set with the same text string set for the corresponding identifier in the file name of the relevant safety data logging settings file. The meaning of each identifier is as follows.
N: Logging Settings Number
ID: Logging Settings ID
SIGNATURE: Safety Signature

*2. As identifiers, the date of output is set as YYYYMMDD, and the time of output is as hhmmss.

The header row of a log record is output in the first row. The contents of the log record are output in the second and following rows.

| Column | Item | Description |
|---|---|---|
| 1st column | Record index (index) | Index number of the record. 0 to 2999 |
| 2nd column | Sampling time (time) | Sampling execution time. It is based on the time information configured in the NX502 CPU Unit or NX102 CPU Unit.<br>YYYY-MM-DD hh:mm:ss.ms |
| 3rd column | Condition is met (condition) | When the trigger condition is met for the record, 1 is output.<br>* If the logging is stopped by pressing the service switch, there may be no record with 1 for this value. |
| 4th column and later | Sampling data (variable name) | It lists all the sampling data corresponding to the number of logging target variables. All data is expressed in decimal notation.<br>BOOL/SAFEBOOL: 0 (FALSE), 1 (TRUE)<br>BYTE: 0 to 255<br>WORD/SAFEWORD: 0 to 65535<br>INT/SAFEINT: -32768 to 32767<br>DINT/SAFEDINT: -2147483648 to 2147483647 |

After the final row of the record, the additional information (footer) shown below is output following a blank row.

| Row No. | Item | Description |
|---|---|---|
| 1st footer row | Trigger variable (condition) | The trigger condition used for the logging is output. |
| 2nd footer row | Reason for logging stopped (StopType) | The reason for the stopped logging is provided.<br>condition: The logging stopped because the trigger condition specified in the logging setting was met.<br>other: The logging stopped due to any reason other than the trigger condition. |

This is a log file example:
(condition)

Logging completion condition: VarX changes to FALSE
Logging target: Var1(SAFEBOOL), Var2(SAFEBOOL), Var3(WORD)
Reason for logging stopped: The trigger condition is met. (VarX changes to FALSE.) The condition was met when No2205 was read in the sampling.

```
"index","time","condition","Var1:SAFEBOOL","Var2:SAFEBOOL","Var3:WORD"
:
"2200","2017-06-16 11:16:40.443","0","1","1","153"
"2201","2017-06-16 11:16:40.448","0","1","0","153"
"2202","2017-06-16 11:16:40.453","0","1","0","153"
"2203","2017-06-16 11:16:40.458","0","1","0","153"
"2204","2017-06-16 11:16:40.463","0","1","0","153"
"2205","2017-06-16 11:16:40.468","1","1","0","153"
"2206","2017-06-16 11:16:40.473","0","1","0","153"
"2207","2017-06-16 11:16:40.478","0","1","0","153"
"2208","2017-06-16 11:16:40.483","0","1","0","150"
"2209","2017-06-16 11:16:40.488","0","1","0","150"
:
"2999","2017-06-16 11:16:40.488","0","1","0","150”

"condition:VarX=false"
"StopType:condition"
```

# 11-6 Versions for Which Safety Data Logging Can Be Used

This section describes combinations of unit models and unit versions that can use the safety data logging, and corresponding Sysmac Studio versions.

You can use the following version or later/higher.

| Safety Control Unit | | CPU Unit | | Sysmac Studio version |
|---|---|---|---|---|
| Model | Version | Model | Version | |
| NX-SL5□□□ | Ver.1.3 | NX502 | Ver.1.60 | Ver.1.54 |
| | | NX102 | Ver.1.31 | Ver.1.24 |

# 12

# Troubleshooting



This section describes troubleshooting for the Safety CPU Unit and Safety I/O Units.

**12-1 How to Check for Errors ..... 12-2**

**12-2 Checking for Errors with the Indicators ..... 12-3**
12-2-1 Troubleshooting the Main Errors in the Safety CPU Unit ..... 12-3
12-2-2 Troubleshooting the Main Errors in the Safety I/O Units ..... 12-11

**12-3 Checking for Errors with the Sysmac Studio ..... 12-14**
12-3-1 Checking for Errors from the Sysmac Studio ..... 12-14
12-3-2 Types of Errors ..... 12-15
12-3-3 Event Codes for Errors and Troubleshooting Procedures ..... 12-17
12-3-4 Error Descriptions ..... 12-33

**12-4 Checking Errors of CIP Safety Target Devices Using Sysmac Studio .. 12-93**
12-4-1 Troubleshooting the CIP Safety Target Device Errors ..... 12-93
12-4-2 CIP Safety Connection Status Codes and Troubleshooting ..... 12-98

**12-5 Resetting Errors ..... 12-102**

**12-6 Troubleshooting Flow When Errors Occur ..... 12-103**

# 12-1 How to Check for Errors

You can check to see if an error has occurred in the Safety Control Units with the following methods.

| Checking method | What you can check |
|---|---|
| Checking the Indicators | Operating status of the NJ/NX-series CPU Unit, Safety CPU Unit, and Safety I/O Units |
| Checking with the Troubleshooting Function of the Sysmac Studio | Current errors in the Safety CPU Unit and Safety I/O Units, error logs in the Safety CPU Unit and Safety I/O Units, and the sources, causes, and corrections for errors |

In addition, you can check to see if an error has occurred with the CIP Safety target device by using the following methods.

| Checking method | What you can check |
|---|---|
| Checking the Indicators of the CIP Safety Target Device | Device status and error status, etc. |
| Checking with the CIP Safety Monitor Function of Sysmac Studio | • Device status<br>• Connection status<br>• Parameter monitor value<br>• Event logs (for OMRON CIP Safety target devices only) |

# 12-2 Checking for Errors with the Indicators

You can use the indicators on the NX Units to check the NX Unit status and errors.
This section describes the meanings of errors that the indicators show and the troubleshooting procedures for them.

## 12-2-1 Troubleshooting the Main Errors in the Safety CPU Unit

### NX-SL3300/NX-SL3500

| TS indicator | FS indicator | Cause | Corrective action |
|---|---|---|---|
|  |  | --- | Status is normal. |
|  |  | System Error | Refer to System Error ( page 12-34). |
|  |  1-second interval | NX Bus I/O Communications Stopped | Refer to NX Bus I/O Communications Stopped ( page 12-46). |
|  1-second interval |  1-second interval | NX Unit I/O Communications Error | Refer to NX Unit I/O Communications Error ( page 12-37). |
|  2-second interval | --- | • Initializing<br>• Downloading | Status is normal. Wait until processing is completed. |

| TS indicator | FS indicator | Cause | Corrective action |
|---|---|---|---|
| ☐ | --- | No power is supplied by the Unit power supply. | Check the following items and make sure that power is correctly supplied from the Unit power supply.<br>Checks Related to the Power Supply<br>• Make sure that the power supply cable is wired properly.<br>• Make sure that there are no breaks in the power supply cable.<br>• Make sure that the power supply voltage is within the specified range.<br>• Make sure that the power supply has enough capacity.<br>• Make sure that the power supply has not failed.<br>If you cannot resolve the problem after you check the above items and cycle the Slave Terminal power supply, the Unit may have a hardware failure. In that case, replace the Unit. |
| ☐ | --- | • Waiting for initialization to start<br>• Restarting the Unit | Status is normal. Wait until processing is completed. |
| --- |  <br>1-second interval | Safety Process Data Communications Not Established Error | Refer to Safety Process Data Communications Not Established Error ( page 12-35). |
| --- |  <br>1-second interval | Safety Process Data Communications Timeout | Refer to Safety Process Data Communications Timeout ( page 12-39). |
| --- |  <br>1-second interval | The safety connections are not established (including when they are currently being established). | Wait until processing is completed. |
| --- |  <br>1-second interval | NX Bus Communications Settings Read Error | Refer to NX Bus Communications Settings Read Error ( page 12-59). |
| --- |  <br>1-second interval | Safety Application Data Read Error | Refer to Safety Application Data Read Error ( page 12-59). |

| TS indicator | FS indicator | Cause | Corrective action |
|---|---|---|---|
| --- |  1-second interval | NX Bus Communications Settings and Safety Application Data Mismatch | Refer to NX Bus Communications Settings and Safety Application Data Mismatch ( page 12-60). |
| --- |  1-second interval | Non-volatile Memory Access Error | Refer to Non-volatile Memory Access Error ( page 12-61). |
| --- |  1-second interval | Division by Zero | Refer to Division by Zero ( page 12-56). |
| --- |  1-second interval | Cast Error | Refer to Cast Error ( page 12-57). |
| --- |  1-second interval | MUX Error | Refer to MUX Error ( page 12-58). |
| --- |  1-second interval | SF_Antivalent Error | Refer to SF_Antivalent Error ( page 12-47). |
| --- |  1-second interval | SF_EDM Error | Refer to SF_EDM Error ( page 12-47). |
| --- |  1-second interval | SF_EmergencyStop Error | Refer to SF_EmergencyStop Error ( page 12-48). |
| --- |  1-second interval | SF_EnableSwitch Error | Refer to SF_EnableSwitch Error ( page 12-48). |
| --- |  1-second interval | SF_Equivalent Error | Refer to SF_Equivalent Error ( page 12-49). |
| --- |  1-second interval | SF_ESPE Error | Refer to SF_ESPE Error ( page 12-49). |
| --- |  1-second interval | SF_GuardLocking Error | Refer to SF_GuardLocking Error ( page 12-50). |
| --- |  1-second interval | SF_GuardMonitoring Error | Refer to SF_GuardMonitoring Error ( page 12-50). |
| --- |  1-second interval | SF_ModeSelector Error | Refer to SF_ModeSelector Error ( page 12-51). |
| --- |  1-second interval | SF_MutingPar Error | Refer to SF_MutingPar Error ( page 12-51). |
| --- |  1-second interval | SF_MutingPar_2Sensor Error | Refer to SF_MutingPar_2Sensor Error ( page 12-52). |

| TS indicator | FS indicator | Cause | Corrective action |
|---|---|---|---|
| --- |  1-second interval | SF_MutingSeq Error | Refer to SF_MutingSeq Error ( page 12-52). |
| --- |  1-second interval | SF_OutControl Error | Refer to SF_OutControl Error ( page 12-53). |
| --- |  1-second interval | SF_SafetyRequest Error | Refer to SF_SafetyRequest Error ( page 12-53). |
| --- |  1-second interval | SF_TestableSafetySensor Error | Refer to SF_TestableSafetySensor Error ( page 12-54). |
| --- |  1-second interval | SF_TwoHandControlTypeII Error | Refer to SF_TwoHandControlTypeII Error ( page 12-54). |
| --- |  1-second interval | SF_TwoHandControlTypeIII Error | Refer to SF_TwoHandControlTypeIII Error ( page 12-55). |
| --- | --- | NX Message Communications Error | Refer to NX Message Communications Error ( page 12-69). |

## NX-SL5500/NX-SL5700

| TS indicator | NS Indicator | FS indicator | P ERR Indicator | Seven-segment Indicator | Cause | Corrective action |
|---|---|---|---|---|---|---|
|  |  |  |  | --- | --- | Status is normal. |
|  |  |  |  | [H3] | System Error | Refer to System Error ( page 12-34). |
|  |  |  |  | [L3] | An error has occurred in the software. | Cycle the power supply. If the error reoccurs, contact your OMRON distributor. |
|  | --- | --- | --- | --- | NX Bus I/O Communications Stopped | Refer to NX Bus I/O Communications Stopped ( page 12-46). |
|  1-second interval | --- | --- | --- | --- | NX Unit I/O Communications Error | Refer to NX Unit I/O Communications Error ( page 12-37). |
|  2-second interval | --- | --- | --- | --- | • Initializing<br>• Downloading | Status is normal. Wait until processing is completed. |

| TS indicator | NS Indicator | FS indicator | P ERR Indicator | Seven-segment Indicator | Cause | Corrective action |
|---|---|---|---|---|---|---|
| ☐ | --- | --- | --- | --- | No power is supplied by the Unit power supply. | Check the following items and make sure that power is correctly supplied from the Unit power supply.<br>Checks Related to the Power Supply<br>• Make sure that the power supply cable is wired properly.<br>• Make sure that there are no breaks in the power supply cable.<br>• Make sure that the power supply voltage is within the specified range.<br>• Make sure that the power supply has enough capacity.<br>• Make sure that the power supply has not failed.<br>If you cannot resolve the problem after you check the above items and cycle the Slave Terminal power supply, the Unit may have a hardware failure. In that case, replace the Unit. |
| ☐ | --- | --- | --- | --- | • Waiting for initialization to start<br>• Restarting the Unit | Status is normal. Wait until processing is completed. |
| --- | 1-second interval | --- | --- | [d6] --> [iP] --> Remote IP address | CIP Safety Originator Connection Not Established Error | Refer to CIP Safety Originator Connection Not Established Error ( page 12-41). |
| --- | 1-second interval | --- | --- | [dA] --> [iP] --> Remote IP address | CIP Safety Originator Connection Timeout | Refer to CIP Safety Originator Connection Timeout ( page 12-43). |

| TS indicator | NS Indicator | FS indicator | P ERR Indicator | Seven-segment Indicator | Cause | Corrective action |
|---|---|---|---|---|---|---|
| --- |  1-second interval | --- | --- | [d5] --> [iP] --> [Remote IP address] | CIP Safety Target Does Not Exist | Refer to CIP Safety Target Does Not Exist ( page 12-44). |
| --- |  1-second interval | --- | --- | [A0] --> Assembly No. Upper Byte --> Assembly No. Lower Byte | CIP Safety Target Connection Timeout | Refer to CIP Safety Target Connection Timeout ( page 12-45). |
| --- | --- |  1-second interval | --- | --- | FSoE Master Connection Not Established Error | Refer to FSoE Master Connection Not Established Error ( page 12-36). |
| --- | --- |  1-second interval | --- | --- | FSoE Master Connection Timeout | Refer to FSoE Master Connection Timeout ( page 12-40). |
| --- | --- |  1-second interval | --- | --- | The safety connections are not established (including when they are currently being established). | Wait until processing is completed. |
| --- | --- | --- |  | --- | NX Bus Communications Settings Read Error | Refer to NX Bus Communications Settings Read Error ( page 12-59). |
| --- | --- | --- |  | --- | Safety Application Data Read Error | Refer to Safety Application Data Read Error ( page 12-59). |
| --- | --- | --- |  | --- | NX Bus Communications Settings and Safety Application Data Mismatch | Refer to NX Bus Communications Settings and Safety Application Data Mismatch ( page 12-60). |

| TS indicator | NS Indicator | FS indicator | P ERR Indicator | Seven-segment Indicator | Cause | Corrective action |
|---|---|---|---|---|---|---|
| --- | --- | --- |  | --- | Non-volatile Memory Access Error | Refer to Non-volatile Memory Access Error ( page 12-61). |
| --- | --- | --- |  | --- | Division by Zero | Refer to Division by Zero ( page 12-56). |
| --- | --- | --- |  | --- | Cast Error | Refer to Cast Error ( page 12-57). |
| --- | --- | --- |  | --- | MUX Error | Refer to MUX Error ( page 12-58). |
| --- | --- | --- |  1-second interval | --- | SF_Antivalent Error | Refer to SF_Antivalent Error ( page 12-47). |
| --- | --- | --- |  1-second interval | --- | SF_EDM Error | Refer to SF_EDM Error ( page 12-47). |
| --- | --- | --- |  1-second interval | --- | SF_EmergencyStop Error | Refer to SF_EmergencyStop Error ( page 12-48). |
| --- | --- | --- |  1-second interval | --- | SF_EnableSwitch Error | Refer to SF_EnableSwitch Error ( page 12-48). |
| --- | --- | --- |  1-second interval | --- | SF_Equivalent Error | Refer to SF_Equivalent Error ( page 12-49). |
| --- | --- | --- |  1-second interval | --- | SF_ESPE Error | Refer to SF_ESPE Error ( page 12-49). |
| --- | --- | --- |  1-second interval | --- | SF_GuardLocking Error | Refer to SF_GuardLocking Error ( page 12-50). |
| --- | --- | --- |  1-second interval | --- | SF_GuardMonitoring Error | Refer to SF_GuardMonitoring Error ( page 12-50). |
| --- | --- | --- |  1-second interval | --- | SF_ModeSelector Error | Refer to SF_ModeSelector Error ( page 12-51). |

| TS indicator | NS Indicator | FS indicator | P ERR Indicator | Seven-segment Indicator | Cause | Corrective action |
|---|---|---|---|---|---|---|
| --- | --- | --- |  1-second interval | --- | SF_MutingPar Error | Refer to SF_MutingPar Error ( page 12-51). |
| --- | --- | --- |  1-second interval | --- | SF_MutingPar_2Sensor Error | Refer to SF_MutingPar_2Sensor Error ( page 12-52). |
| --- | --- | --- |  1-second interval | --- | SF_MutingSeq Error | Refer to SF_MutingSeq Error ( page 12-52). |
| --- | --- | --- |  1-second interval | --- | SF_OutControl Error | Refer to SF_OutControl Error ( page 12-53). |
| --- | --- | --- |  1-second interval | --- | SF_SafetyRequest Error | Refer to SF_SafetyRequest Error ( page 12-53). |
| --- | --- | --- |  1-second interval | --- | SF_TestableSafetySensor Error | Refer to SF_TestableSafetySensor Error ( page 12-54). |
| --- | --- | --- |  1-second interval | --- | SF_TwoHandControlTypeII Error | Refer to SF_TwoHandControlTypeII Error ( page 12-54). |
| --- | --- | --- |  1-second interval | --- | SF_TwoHandControlTypeIII Error | Refer to SF_TwoHandControlTypeIII Error ( page 12-55). |
| --- | --- | --- | --- | --- | NX Message Communications Error | Refer to NX Message Communications Error ( page 12-69). |
| --- | --- | --- | --- | [E1] --> [01] | Safety Unit Restore Operation Failed to Start (SD Memory Card Access Failed) | Refer to Safety Unit Restore Operation Failed to Start (SD Memory Card Access Failed) ( page 12-62). |
| --- | --- | --- | --- | [E1] --> [02] | Safety Unit Restore Operation Failed to Start (Safety Unit Restore File Read Failure) | Refer to Safety Unit Restore Operation Failed to Start (Safety Unit Restore File Read Failure) ( page 12-63). |
| --- | --- | --- | --- | [E1] --> [03] | Safety Unit Restore Operation Failed to Start (Model Mismatch) | Refer to Safety Unit Restore Operation Failed to Start (Model Mismatch) ( page 12-64). |

| TS indicator | NS Indicator | FS indicator | P ERR Indicator | Seven-segment Indicator | Cause | Corrective action |
|---|---|---|---|---|---|---|
| --- | --- | --- | --- | [E1] --> [04] | Safety Unit Restore Operation Failed to Start (Version Mismatch) | Refer to Safety Unit Restore Operation Failed to Start (Version Mismatch) ( page 12-65). |
| --- | --- | --- | --- | [E1] --> [05] | Safety Unit Restore Operation Failed to Start (Node Name Mismatch) | Refer to Safety Unit Restore Operation Failed to Start (Node Name Mismatch) ( page 12-66). |
| --- | --- | --- | --- | [E1] --> [06] | Safety Unit Restore Operation Failed to Start (Safety Password Mismatch) | Refer to Safety Unit Restore Operation Failed to Start (Safety Password Mismatch) ( page 12-67). |
| --- | --- | --- | --- | [E1] --> [10] | Safety Unit Restore Operation Failed | Refer to Safety Unit Restore Operation Failed ( page 12-68). |
| --- | --- | --- | --- | [E8] | Incorrect DIP Switch Setting | Refer to Incorrect DIP Switch Setting ( page 12-61). |

## 12-2-2 Troubleshooting the Main Errors in the Safety I/O Units

### NX-SIH400/NX-SID800/NX-SOD400/NX-SOH200

| TS indicator | FS indicator | Cause | Corrective action |
|---|---|---|---|
|  |  | --- | Status is normal. |
|  |  1-second interval | Safety Process Data Communications Not Established - Incorrect Unit Parameter Error | Refer to Safety Process Data Communications Not Established - Incorrect Unit Parameter Error ( page 12-73). |
|  |  1-second interval | Safety Process Data Communications Not Established, Incorrect FSoE Slave Address Error | Refer to Safety Process Data Communications Not Established, Incorrect FSoE Slave Address Error ( page 12-74). |
|  |  1-second interval | Safety Process Data Communications Not Established, Incorrect Frame Error | Refer to Safety Process Data Communications Not Established, Incorrect Frame Error ( page 12-75). |
|  |  1-second interval | I/O Power Supply Voltage Error | Refer to I/O Power Supply Voltage Error ( page 12-83). |
|  |  1-second interval | Output Power Interrupt Circuit Error | Refer to Output Power Interrupt Circuit Error ( page 12-84). |

| TS indicator | FS indicator | Cause | Corrective action |
|---|---|---|---|
|  |  1-second interval | External Test Signal Failure at Safety Input | Refer to External Test Signal Failure at Safety Input ( page 12-85). |
|  |  1-second interval | Internal Circuit Error at Safety Input | Refer to Internal Circuit Error at Safety Input ( page 12-80). |
|  |  1-second interval | Discrepancy Error at Safety Input | Refer to Discrepancy Error at Safety Input ( page 12-86). |
|  |  1-second interval | Overload Detected at Test Output | Refer to Overload Detected at Test Output ( page 12-87). |
|  |  1-second interval | Stuck-at-high Detected at Test Output | Refer to Stuck-at-high Detected at Test Output ( page 12-88). |
|  |  1-second interval | Internal Circuit Error at Test Output | Refer to Internal Circuit Error at Test Output ( page 12-81). |
|  |  1-second interval | Short Circuit Detected at Safety Output | Refer to Short Circuit Detected at Safety Output ( page 12-89). |
|  |  1-second interval | Stuck-at-high Detected at Safety Output | Refer to Stuck-at-high Detected at Safety Output ( page 12-90). |
|  |  1-second interval | Internal Circuit Error at Safety Output | Refer to Internal Circuit Error at Safety Output ( page 12-82). |
|  2-second interval | --- | Initializing | Status is normal. Wait until processing is completed. |
|  |  | System Error | Refer to System Error ( page 12-72). |
|  |  1-second interval | NX Bus I/O Communications Stopped | Refer to NX Bus I/O Communications Stopped ( page 12-79). |
|  1-second interval |  1-second interval | NX Unit I/O Communications Error | Refer to NX Unit I/O Communications Error ( page 12-76). |

| TS indicator | FS indicator | Cause | Corrective action |
|---|---|---|---|
| ☐ | --- | No power is supplied by the Unit power supply. | Check the following items and make sure that power is correctly supplied from the Unit power supply.<br>Checks Related to the Power Supply<br>• Make sure that the power supply cable is wired properly.<br>• Make sure that there are no breaks in the power supply cable.<br>• Make sure that the power supply voltage is within the specified range.<br>• Make sure that the power supply has enough capacity.<br>• Make sure that the power supply has not failed.<br>If you cannot resolve the problem after you check the above items and cycle the Slave Terminal power supply, the Unit may have a hardware failure. In that case, replace the Safety I/O Unit. |
| ☐ | --- | • Waiting for initialization to start<br>• Restarting the Unit | Status is normal. Wait until processing is completed. |
| --- |  <br>1-second interval | The safety connections are not established (including when they are currently being established). | Wait until processing is completed. |
| --- |  <br>1-second interval | Safety Process Data Communications Timeout | Refer to Safety Process Data Communications Timeout ( page 12-78). |
| --- | --- | NX Message Communications Error | Refer to NX Message Communications Error ( page 12-91). |

# 12-3 Checking for Errors with the Sysmac Studio

Error management on the NX Series is based on the methods used for the NJ/NX-series Controllers. This allows you to use the Sysmac Studio to check the meanings of errors and troubleshooting procedures.

## 12-3-1 Checking for Errors from the Sysmac Studio

When an error occurs, you can place the Sysmac Studio online to the Controller or the Communications Coupler Unit to check current Controller errors and the log of past Controller errors.

Refer to the user's manual for the connected NX bus master for information on checking errors.

### Current Errors

Open the Sysmac Studio's **Controller Error** Tab Page to check the current *error's level*, *source*, *source details*, *event name*, *event codes*, *details*, *attached information 1 to 4*, and *correction*. Errors in the *observation level* are not displayed.



**Additional Information**

**Number of Current Errors**
The following table gives the number of errors that are reported simultaneously as current errors in each Unit.

| Unit | Number of simultaneous errors |
|---|---|
| Safety CPU Unit | NX-SL3300 / NX-SL3500: 32<br>NX-SL5500 / NX-SL5700: 64 |
| Safety I/O Units | 16 |

If the number of errors exceeds the maximum number of reportable current errors, errors are reported with a priority given to the oldest and highest-level errors. The errors that occur beyond this limit are not reported.
Errors that are not reported are still shown in the error status.

### Log of Past Errors

You can check the following information on past errors on the **Controller Event Log** Tab Page in the Sysmac Studio: *times*, *levels*, *sources*, *source details*, *event names*, *event codes*, *details*, *attached information 1 through 4*, and *corrections*.

**Additional Information**

**Number of Events in Log of Past Errors**

The following table gives the number of events that each event log can record. When an event log is full and a new event occurs, the oldest data in the log is replaced with the new event information

| Event log category | Unit | |
|---|---|---|
| | Safety CPU Unit | Safety I/O Units |
| System event log | NX-SL3300 / NX-SL3500: 32<br>NX-SL5500 / NX-SL5700: 128 | 10 |
| Access event log | 32 events total | 10 |

Refer to the *NJ/NX-series Troubleshooting Manual (Cat. No. W503)* and the *Sysmac Studio Version 1 Operation Manual (Cat. No. W504)* for information on the items you can check and for how to check for errors.

Refer to *12-3-3 Event Codes for Errors and Troubleshooting Procedures* on page 12-17 for information on event codes.

## 12-3-2 Types of Errors

This section describes the errors that can originate in the Safety CPU Unit and Safety I/O Units and how to correct them.

### Safety CPU Unit

The errors that can occur in the Safety CPU Unit and the operation that is performed for each are described in the following table.

| Type | Overview | Operation |
|---|---|---|
| System Error | Errors that occur in hardware self-diagnosis in the Safety CPU Unit | The Safety CPU Unit will stop.<br>The Safety I/O Units will detect this and make the safety I/O data inactive (OFF). |
| Communications Errors | Errors that occur in safety process data communications | The Safety CPU Unit will continue operation.<br>The relevant safety process data communications will stop.<br>The Unit that detects the safety process data communications error will make the safety I/O data inactive (OFF). |
| Operation continuation error related to program execution | Errors that occur in the safety function blocks in the Safety CPU Unit | The Safety CPU Unit will continue operation.<br>Refer to *NX-series Safety Control Unit Instructions Reference Manual (Cat. No. Z931)* for the operation of function blocks in which errors occur. |
| Operation stop error related to program execution | Errors that occur in the safety function blocks in the Safety CPU Unit | The Safety CPU Unit operation is continued; however, the execution of user programs and all the safety process data communications are stopped.<br>Refer to *NX-series Safety Control Unit Instructions Reference Manual (Cat. No. Z931)* for the operation of function blocks in which errors occur. |
| Setting error | Safety CPU Unit setting error | The Safety CPU Unit continues operation. Refer to the list of errors for details. |

| Type | Overview | Operation |
|---|---|---|
| Other Errors | Errors other than those given above | The Safety CPU Unit will continue operation.<br>Refer to the list of errors for details. |

Events are recorded in the log when the Safety CPU Unit is accessed by the Sysmac Studio.

| Type | Overview | Operation |
|---|---|---|
| User access log | The Safety CPU Unit was accessed by the Sysmac Studio. | The Safety CPU Unit will continue operation. |

## Safety I/O Units

The errors that can occur in the Safety I/O Units and the operation that is performed for each are described in the following table.

| Type | Overview | Operation |
|---|---|---|
| System error | Errors that occur in hardware self-diagnosis in the Safety I/O Units | The Safety I/O Unit will stop. |
| Communications errors | Errors that occur in safety process data communications with the Safety CPU Units | The Safety I/O Unit will continue operation.<br>The Safety I/O Unit will make the safety I/O data inactive (OFF). |
| Safety I/O errors | Errors that occur in safety I/O in the Safety I/O Units | The Safety I/O Unit will continue operation.<br>Safety process data communications will continue.<br>The safety I/O data will become inactive (OFF). |
| Other errors | Errors other than those given above | The Safety I/O Unit will continue operation.<br>Refer to the list of errors for details. |

Events are recorded in the log when the Safety I/O Unit is accessed by the Sysmac Studio.

| Type | Overview | Operation |
|---|---|---|
| User access log | The Safety I/O Unit was accessed by the Sysmac Studio. | The Safety I/O Unit will continue operation. |

## 12-3-3 Event Codes for Errors and Troubleshooting Procedures

This section lists the errors (events) that can originate in the Safety CPU Unit and Safety I/O Units.

Event levels are given in the tables as follows:

Maj : Major fault level
Prt : Partial fault level
Min : Minor fault level
Obs : Observation
Info : Information

Refer to the *NJ/NX-series Troubleshooting Manual (Cat. No. W503)* for all of the event codes that may occur for an NJ/NX-series Controller.

### Safety CPU Unit

The errors (events) that can occur in the Safety CPU Unit are listed in the following tables.

#### ● System Error

| Event code | Event name | Meaning | Assumed cause | Level | | | | | Reference |
|---|---|---|---|---|---|---|---|---|---|
| | | | | Maj | Prt | Min | Obs | Info | |
| 05200000 hex | System Error | A hardware error was detected during self-diagnosis of the hardware. | • Hardware has failed.<br>• A memory error occurred due to a transient cause, such as a software error or excessive noise. | | | ○ | | | page 12-34 |

#### ● Communications Errors

| Event code | Event name | Meaning | Assumed cause | Level | | | | | Reference |
|---|---|---|---|---|---|---|---|---|---|
| | | | | Maj | Prt | Min | Obs | Info | |
| 35200000 hex<br>[NX-SL3□□□] | Safety Process Data Communications Not Established Error | Safety process data communications was not established with one or more safety slaves. | • The communications settings for safety process data are not correct, the safety slave is not in the correct status, etc.<br>• The safety slave for safety process data communications is not connected.<br>• The NX Unit Mounting Setting for the safety slave for safety process data communications is set to Disabled. | | | ○ | | | page 12-35 |

| Event code | Event name | Meaning | Assumed cause | Level | | | | | Reference |
|---|---|---|---|---|---|---|---|---|---|
| | | | | Maj | Prt | Min | Obs | Info | |
| 35200001 hex<br>[NX-SL5□□□] | FSoE Master Connection Not Established Error | FSoE communications were not established with the FSoE slave. | • The FSoE communications settings are not correct, the FSoE slave is not in the correct status, etc.<br>• The FSoE slave for FSoE communications is not connected.<br>• The FSoE slave set in the NX Unit Mounting Setting for FSoE communications is disabled.<br>• The version of the FSoE slave to be communicated is older than the configured version. | | | ○ | | | page 12-36 |

| Event code | Event name | Meaning | Assumed cause | Level | | | | | Reference |
|---|---|---|---|---|---|---|---|---|---|
| | | | | Maj | Prt | Min | Obs | Info | |
| 80200000 hex | NX Unit I/O Communications Error | An I/O communications error occurred between the NX bus master and the NX Unit. | NX Bus of the CPU Unit<br>• An error occurred in the CPU Unit, which prevents the NX bus communications from being carried out normally.<br>• The NX Unit is not mounted properly.<br>• The power cable for the Unit power supply is disconnected. Or, the wiring from the Unit power supply to the NX Units is incorrect.<br>• The power cable for the Unit power supply is broken.<br>• The Unit power voltage is out of the supported range, or the Unit power capacity is not sufficient.<br>• There is a hardware error in the NX Unit.<br>Communications Coupler Unit<br>• An error occurred in the Communications Coupler Unit, which prevents the NX bus communications from being carried out normally.<br>• The NX Unit is not mounted properly.<br>• The power cable for the Unit power supply is disconnected. Or, the wiring from the Unit power supply to the NX Units is incorrect.<br>• The power cable for the Unit power supply is broken.<br>• The Unit power voltage is out of the supported range, or the Unit power capacity is not sufficient.<br>• There is a hardware error in the NX Unit. | | | ○ | | | page 12-37 |

| Event code | Event name | Meaning | Assumed cause | Level | | | | | Reference |
|---|---|---|---|---|---|---|---|---|---|
| | | | | Maj | Prt | Min | Obs | Info | |
| 80300000 hex [NX-SL3□□□] | Safety Process Data Communications Timeout | A communications timeout occurred in safety process data communications with the Safety Control Unit. | • A setting is not correct. The setting of the safety task period is too short.<br>• There is excessive noise.<br>• The Safety CPU Unit or safety slave entered a status where it could not continue safety process data communications.<br>• The process data communications were not performed correctly because an error or status change occurred in the NX bus master to which the Unit is connected. | | | ○ | | | page 12-39 |
| 80300001 hex [NX-SL5□□□] | FSoE Master Connection Timeout | A communication timeout occurred in FSoE communications with the FSoE slave. | • A setting is not correct. The setting of the safety task period is too short.<br>• There is excessive noise.<br>• The Safety CPU Unit or FSoE slave entered a state where it could not continue FSoE communications.<br>• The process data communications were not performed correctly because an error or status change occurred in the NX bus master to which the Unit is connected. | | | ○ | | | page 12-40 |

| Event code | Event name | Meaning | Assumed cause | Level | | | | | Reference |
|---|---|---|---|---|---|---|---|---|---|
| | | | | Maj | Prt | Min | Obs | Info | |
| 80310000 hex [NX-SL5□□□] | CIP Safety Originator Connection Not Established Error | CIP safety originator connection was not established. | • The target node is different.<br>• The target node is not configured.<br>• The target node status is invalid.<br>• The **Do not use** Option is selected for CIP message server setting in the EtherNet/IP Port Settings.<br>• The settings of the NX-series CPU Unit and NX-series EtherNet/IP Unit related to CIP Safety communications do not match the settings of the Safety CPU Unit.<br>• Setting to use tag data link communications was made to the NX-series EtherNet/IP Unit that is included in the CIP Safety connection settings.<br>• The NX-series EtherNet/IP Unit with tag data link communications was added to the CIP Safety connection settings.<br>• The **Do not use** Option is selected for CIP Safety communications in the Built-in EtherNet/IP Port Settings. | | | ○ | | | page 12-41 |
| 80320000 hex [NX-SL5□□□] | CIP Safety Originator Connection Timeout | A timeout occurred in CIP safety originator connection. | • The communications cable is disconnected or broken.<br>• The target node entered a state where it could not accept the connection.<br>• The timeout value in the communications setup is too small.<br>• CIP message communications at the target node are stopped.<br>• When the Packet Filter function is enabled in the EtherNet/IP Port Settings, CIP Safety packets from the target node are not allowed.<br>• CIP Safety packets are not allowed by the Packet Filter (Simple) or Packet Filter functions on the target node or the devices on the communication path.<br>• There is excessive noise. | | | ○ | | | page 12-43 |

| Event code | Event name | Meaning | Assumed cause | Level | | | | | Reference |
|---|---|---|---|---|---|---|---|---|---|
| | | | | Maj | Prt | Min | Obs | Info | |
| 80330000 hex [NX-SL5□□□] | CIP Safety Target Does Not Exist | The target node does not exist. | • The communications cable is disconnected or broken.<br>• The target node entered a state where it could not accept the connection.<br>• The timeout value in the communications setup is too small.<br>• CIP message communications at the target node are stopped.<br>• CIP Safety packets from the originator node are not allowed by the Packet Filter (Simple) or Packet Filter functions on the target node or the devices on the communication path.<br>• There is excessive noise. | | | ○ | | | page 12-44 |
| 80340000 hex [NX-SL5□□□] | CIP Safety Target Connection Timeout | A timeout occurred in the CIP Safety Target connection. | • The communications cable is disconnected or broken.<br>• The originator device entered a state where it could not accept the connection.<br>• The timeout value for the communications settings is too small.<br>• CIP Safety packets from the originator node are not allowed by the Packet Filter (Simple) or Packet Filter functions in the EtherNet/IP Port Settings or on the devices on the communication path.<br>• The CIP message server setting for the originator node is set to Do not use.<br>• When the Packet Filter function of the originator node is enabled, CIP Safety packets from the target node are not allowed.<br>• There is excessive noise.<br>• The **Do not use** Option is selected for CIP Safety communications in the Built-in EtherNet/IP Port Settings. | | | ○ | | | page 12-45 |
| 84F00000 hex | NX Bus I/O Communications Stopped | An error occurred in I/O communications between the NX bus master and an NX Unit. | • There is a hardware error in the NX bus master or an NX Unit. | | | ○ | | | page 12-46 |

## ● Operation Continuation Error Related to Program Execution Function

| Event code | Event name | Meaning | Assumed cause | Level | | | | | Reference |
|---|---|---|---|---|---|---|---|---|---|
| | | | | Maj | Prt | Min | Obs | Info | |
| 74A00000 hex | SF_Antivalent Error | An error was detected in execution of a safety function block. | Refer to information on the diagnostic code that is given for attached information 1 in the *NX-series Safety Control Unit Instructions Reference Manual (Cat. No. Z931)*. | | | ○ | | | page 12-47 |
| 74A10000 hex | SF_EDM Error | An error was detected in execution of a safety function block. | Refer to information on the diagnostic code that is given for attached information 1 in the *NX-series Safety Control Unit Instructions Reference Manual (Cat. No. Z931)*. | | | ○ | | | page 12-47 |
| 74A20000 hex | SF_EmergencyStop Error | An error was detected in execution of a safety function block. | Refer to information on the diagnostic code that is given for attached information 1 in the *NX-series Safety Control Unit Instructions Reference Manual (Cat. No. Z931)*. | | | ○ | | | page 12-48 |
| 74A30000 hex | SF_EnableSwitch Error | An error was detected in execution of a safety function block. | Refer to information on the diagnostic code that is given for attached information 1 in the *NX-series Safety Control Unit Instructions Reference Manual (Cat. No. Z931)*. | | | ○ | | | page 12-48 |
| 74A40000 hex | SF_Equivalent Error | An error was detected in execution of a safety function block. | Refer to information on the diagnostic code that is given for attached information 1 in the *NX-series Safety Control Unit Instructions Reference Manual (Cat. No. Z931)*. | | | ○ | | | page 12-49 |
| 74A50000 hex | SF_ESPE Error | An error was detected in execution of a safety function block. | Refer to information on the diagnostic code that is given for attached information 1 in the *NX-series Safety Control Unit Instructions Reference Manual (Cat. No. Z931)*. | | | ○ | | | page 12-49 |
| 74A60000 hex | SF_GuardLocking Error | An error was detected in execution of a safety function block. | Refer to information on the diagnostic code that is given for attached information 1 in the *NX-series Safety Control Unit Instructions Reference Manual (Cat. No. Z931)*. | | | ○ | | | page 12-50 |
| 74A70000 hex | SF_GuardMonitoring Error | An error was detected in execution of a safety function block. | Refer to information on the diagnostic code that is given for attached information 1 in the *NX-series Safety Control Unit Instructions Reference Manual (Cat. No. Z931)*. | | | ○ | | | page 12-50 |

| Event code | Event name | Meaning | Assumed cause | Level | | | | | Reference |
|---|---|---|---|---|---|---|---|---|---|
| | | | | Maj | Prt | Min | Obs | Info | |
| 74A80000 hex | SF_ModeSelector Error | An error was detected in execution of a safety function block. | Refer to information on the diagnostic code that is given for attached information 1 in the *NX-series Safety Control Unit Instructions Reference Manual (Cat. No. Z931)*. | | | ○ | | | page 12-51 |
| 74A90000 hex | SF_MutingPar Error | An error was detected in execution of a safety function block. | Refer to information on the diagnostic code that is given for attached information 1 in the *NX-series Safety Control Unit Instructions Reference Manual (Cat. No. Z931)*. | | | ○ | | | page 12-51 |
| 74AA0000 hex | SF_MutingPar_2Sensor Error | An error was detected in execution of a safety function block. | Refer to information on the diagnostic code that is given for attached information 1 in the *NX-series Safety Control Unit Instructions Reference Manual (Cat. No. Z931)*. | | | ○ | | | page 12-52 |
| 74AB0000 hex | SF_MutingSeq Error | An error was detected in execution of a safety function block. | Refer to information on the diagnostic code that is given for attached information 1 in the *NX-series Safety Control Unit Instructions Reference Manual (Cat. No. Z931)*. | | | ○ | | | page 12-52 |
| 74AC0000 hex | SF_OutControl Error | An error was detected in execution of a safety function block. | Refer to information on the diagnostic code that is given for attached information 1 in the *NX-series Safety Control Unit Instructions Reference Manual (Cat. No. Z931)*. | | | ○ | | | page 12-53 |
| 74AD0000 hex | SF_SafetyRequest Error | An error was detected in execution of a safety function block. | Refer to information on the diagnostic code that is given for attached information 1 in the *NX-series Safety Control Unit Instructions Reference Manual (Cat. No. Z931)*. | | | ○ | | | page 12-53 |
| 74AE0000 hex | SF_TestableSafetySensor Error | An error was detected in execution of a safety function block. | Refer to information on the diagnostic code that is given for attached information 1 in the *NX-series Safety Control Unit Instructions Reference Manual (Cat. No. Z931)*. | | | ○ | | | page 12-54 |
| 74AF0000 hex | SF_TwoHandControlTypeII Error | An error was detected in execution of a safety function block. | Refer to information on the diagnostic code that is given for attached information 1 in the *NX-series Safety Control Unit Instructions Reference Manual (Cat. No. Z931)*. | | | ○ | | | page 12-54 |

| Event code | Event name | Meaning | Assumed cause | Level | | | | | Reference |
|---|---|---|---|---|---|---|---|---|---|
| | | | | Maj | Prt | Min | Obs | Info | |
| 74B00000 hex | SF_TwoHandControlTypeIII Error | An error was detected in execution of a safety function block. | Refer to information on the diagnostic code that is given for attached information 1 in the *NX-series Safety Control Unit Instructions Reference Manual (Cat. No. Z931)*. | | | ○ | | | page 12-55 |

## ● Operation Stop Error Related to Program Execution

| Event code | Event name | Meaning | Assumed cause | Level | | | | | Reference |
|---|---|---|---|---|---|---|---|---|---|
| | | | | Maj | Prt | Min | Obs | Info | |
| 55000000 hex | Division by Zero | Division by zero was detected. | The divisor is zero. | | | ○ | | | page 12-56 |
| 55010000 hex | Cast Error | A casting error was detected. | A value was input that exceeded the range of the receiving variable. | | | ○ | | | page 12-57 |
| 55020000 hex | MUX Error | An MUX instruction error was detected. | The value of the selection input (K) to the MUX instruction is not correct. | | | ○ | | | page 12-58 |

## ● Setting Errors

| Event code | Event name | Meaning | Assumed cause | Level | | | | | Reference |
|---|---|---|---|---|---|---|---|---|---|
| | | | | Maj | Prt | Min | Obs | Info | |
| 10500000 hex | NX Bus Communications Settings Read Error | There is an error in the NX Bus communications settings that are saved in non-volatile memory. | • A hardware failure occurred in the non-volatile memory.<br>• Power was turned OFF while saving data to the non-volatile memory. | | | ○ | | | page 12-59 |
| 10510000 hex | Safety Application Data Read Error | There is an error in the safety application data that is saved in non-volatile memory. | • A hardware failure occurred in the non-volatile memory.<br>• Power was turned OFF while saving data to the non-volatile memory. | | | ○ | | | page 12-59 |
| 10520000 hex | NX Bus Communications Settings and Safety Application Data Mismatch | There is an error in the safety application data that is saved in non-volatile memory. | • The NX bus communications settings that were transferred to the Safety CPU Unit do not match the safety application data. | | | ○ | | | page 12-60 |
| 10530000 hex | Non-volatile Memory Access Error | Reading/writing non-volatile memory failed. | Non-volatile memory failure. | | | ○ | | | page 12-61 |

| Event code | Event name | Meaning | Assumed cause | Level | | | | | Reference |
|---|---|---|---|---|---|---|---|---|---|
| | | | | Maj | Prt | Min | Obs | Info | |
| 24AA0000 hex [NX-SL5□□□] | Incorrect DIP Switch Setting | The DIP switch setting is not correct. | • The DIP switch setting was changed to an incorrect value. | | | | ○ | | page 12-61 |

## ● Restore Function Errors

| Event code | Event name | Meaning | Assumed cause | Level | | | | | Reference |
|---|---|---|---|---|---|---|---|---|---|
| | | | | Maj | Prt | Min | Obs | Info | |
| 35250000 hex [NX-SL5□□□] | Safety Unit Restore Operation Failed to Start (SD Memory Card Access Failed) | Access to the SD Memory Card failed when you start the restore operation for a Safety Unit. | • An SD Memory Card is not inserted.<br>• The SD Memory Card type is not correct.<br>• The format of the SD Memory Card is not correct.<br>• The SD Memory Card is damaged. | | | | ○ | | page 12-62 |
| 35260000 hex [NX-SL5□□□] | Safety Unit Restore Operation Failed to Start (Safety Unit Restore File Read Failure) | Reading the Safety Unit Restore File failed. | • The Safety Unit Restore File is not stored in the specified folder.<br>• The Safety Unit Restore File is corrupted. | | | | ○ | | page 12-63 |
| 35270000 hex [NX-SL5□□□] | Safety Unit Restore Operation Failed to Start (Model Mismatch) | A model mismatch was detected during pre-execution checks for a restore operation for a Safety Unit. | • The restore operation for a Safety Unit was performed on an incorrect Unit.<br>• An incorrect Safety Unit Restore File was used. | | | | ○ | | page 12-64 |
| 35280000 hex [NX-SL5□□□] | Safety Unit Restore Operation Failed to Start (Version Mismatch) | A version mismatch was detected during pre-execution checks for a restore operation for a Safety Unit. | • The restore operation for a Safety Unit was performed on an incompatible unit.<br>• An incorrect Safety Unit Restore File was used. | | | | ○ | | page 12-65 |
| 35290000 hex [NX-SL5□□□] | Safety Unit Restore Operation Failed to Start (Node Name Mismatch) | A node name mismatch was detected during pre-execution checks for a restore operation for a Safety Unit. | • The restore operation for a Safety Unit was performed on an incorrect Unit.<br>• An incorrect Safety Unit Restore File was used.<br>• The node name specified when the Safety Unit Restore File was generated is incorrect. | | | | ○ | | page 12-66 |

| Event code | Event name | Meaning | Assumed cause | Level | | | | | Reference |
|---|---|---|---|---|---|---|---|---|---|
| | | | | Maj | Prt | Min | Obs | Info | |
| 352A0000 hex [NX-SL5□□□] | Safety Unit Restore Operation Failed to Start (Safety Password Mismatch) | A safety password mismatch was detected during pre-execution checks for a restore operation for a Safety Unit. | • The restore operation for a Safety Unit was performed on an incorrect Unit.<br>• An incorrect Safety Unit Restore File was used.<br>• The safety password specified when the Safety Unit Restore File was generated is incorrect. | | | | ○ | | page 12-67 |
| 352B0000 hex [NX-SL5□□□] | Safety Unit Restore Operation Failed | The restore operation for a Safety Unit ended in an error. | • The SD Memory Card was removed during a restore operation for a Safety Unit.<br>• Data was read from or written to the SD Memory Card via the Support Software or an FTP client during a restore operation for a Safety Unit. | | | | ○ | | page 12-68 |

## ● Other Errors

| Event code | Event name | Meaning | Assumed cause | Level | | | | | Reference |
|---|---|---|---|---|---|---|---|---|---|
| | | | | Maj | Prt | Min | Obs | Info | |
| 80220000 hex | NX Message Communications Error | An error was detected in message communications for an NX Unit and the message frame was discarded. | • The message communications load is high.<br>• The communications cable is disconnected or broken. This cause does not apply if attached information 2 is 0 (NX bus).<br>• Message communications were cut off as the result of executing a synchronization or restoration operation on the Sysmac Studio or as the result of disconnecting an EtherCAT slave. | | | | ○ | | page 12-69 |
| 951E0000 hex | Sysmac Studio Communications Connection Timeout | A communications timeout occurred between the Sysmac Studio and the Safety CPU Unit. | • The communications cable was disconnected. | | | | | ○ | page 12-70 |
| 951F0000 hex | Clear All Memory Rejected | Clearing all of memory failed. | • The Clear All Memory operation for a Controller or a Slave Terminal was performed. | | | | | ○ | page 12-70 |

### ● User Access Log

| Event code | Event name | Meaning | Assumed cause | Level | | | | | Reference |
|---|---|---|---|---|---|---|---|---|---|
| | | | | Maj | Prt | Min | Obs | Info | |
| 90400000 hex | Event Log Cleared | The event log was cleared. | • The event log was cleared by the user. | | | | | ○ | page 12-71 |
| 90430000 hex | Memory All Cleared | The Unit settings were cleared. | • The Clear All Memory operation was performed. | | | | | ○ | page 12-71 |

## Safety I/O Units

The errors (events) that can occur in the Safety I/O Units are listed in the following tables.

### ● System Error

| Event code | Event name | Meaning | Assumed cause | Level | | | | | Reference |
|---|---|---|---|---|---|---|---|---|---|
| | | | | Maj | Prt | Min | Obs | Info | |
| 05200000 hex | System Error | A hardware error was detected during self-diagnosis of the hardware. | • Hardware has failed.<br>• A memory error occurred due to a transient cause, such as a software error or excessive noise. | | | ○ | | | page 12-72 |

### ● Communications Errors

| Event code | Event name | Meaning | Assumed cause | Level | | | | | Reference |
|---|---|---|---|---|---|---|---|---|---|
| | | | | Maj | Prt | Min | Obs | Info | |
| 35210000 hex | Safety Process Data Communications Not Established - Incorrect Unit Parameter Error | Safety process data communications was not established with the Safety CPU Unit. | • The model or safety I/O terminal settings are not correct. | | | ○ | | | page 12-73 |
| 35230000 hex | Safety Process Data Communications Not Established, Incorrect FSoE Slave Address Error | Safety process data communications was not established with the Safety CPU Unit because of an incorrect FSoE slave address. | • The setting of the FSoE slave address in the safety process data communications settings is different from the setting in the Unit. | | | ○ | | | page 12-74 |

| Event code | Event name | Meaning | Assumed cause | Level | | | | | Reference |
|---|---|---|---|---|---|---|---|---|---|
| | | | | Maj | Prt | Min | Obs | Info | |
| 35240000 hex | Safety Process Data Communications Not Established, Incorrect Frame Error | Safety process data communications were not established with the Safety CPU Unit because an incorrect frame was received. | • An incorrect frame was received in safety process data communications.<br>• There is excessive noise. | | | ○ | | | page 12-75 |
| 80200000 hex | NX Unit I/O Communications Error | An I/O communications error occurred between the NX bus master and the NX Unit. | • The NX Unit is not mounted properly.<br>• The power cable for the Unit power supply is disconnected. Or, the wiring from the Unit power supply to the NX Units is incorrect.<br>• The power cable for the Unit power supply is broken.<br>• The unit power voltage is out of the supported range, or the unit power capacity is not sufficient.<br>• There is a hardware error in the NX Unit. | | | ○ | | | page 12-76 |
| 80300000 hex | Safety Process Data Communications Timeout | A communications timeout occurred in safety process data communications with the Safety Control Unit. | • A setting is not correct. The setting of the safety task period is too short.<br>• There is excessive noise.<br>• The Safety CPU Unit or safety slave entered a status where it could not continue safety process data communications.<br>• The process data communications were not performed correctly because an error or status change occurred in the NX bus master to which the Unit is connected. | | | ○ | | | page 12-78 |
| 84F10000 hex | NX Bus I/O Communications Stopped | An error occurred in I/O communications between the NX bus master and an NX Unit. | There is a hardware error in the NX bus master or an NX Unit. | | | ○ | | | page 12-79 |

## ● Safety I/O Errors

| Event code | Event name | Meaning | Assumed cause | Level | | | | | Reference |
|---|---|---|---|---|---|---|---|---|---|
| | | | | Maj | Prt | Min | Obs | Info | |
| 05210000 hex | Internal Circuit Error at Safety Input | A fault was detected in the internal circuit for the safety input terminal. | • The internal circuit for the safety input terminal is faulty.<br>• A memory error or signal error occurred due to a transient cause, such as an excessive noise. | | | ○ | | | page 12-80 |
| 05220000 hex | Internal Circuit Error at Test Output | A fault was detected in the internal circuit for the test output terminal. | • The internal circuit for the test output terminal is faulty.<br>• A memory error or signal error occurred due to a transient cause, such as an excessive noise. | | | ○ | | | page 12-81 |
| 05230000 hex | Internal Circuit Error at Safety Output | A fault was detected in the internal circuit for the safety output terminal. | • The internal circuit for the safety output terminal is faulty.<br>• A memory error or signal error occurred due to a transient cause, such as an excessive noise. | | | ○ | | | page 12-82 |
| 65200000 hex | I/O Power Supply Voltage Error | An incorrect I/O power supply voltage was detected. | • The input power or output power is not supplied correctly. | | | ○ | | | page 12-83 |
| 65210000 hex | Output Power Interrupt Circuit Error | An error was detected by the output power interruption test. | • The wiring is not correct or there is a fault in the hardware. | | | ○ | | | page 12-84 |
| 65220000 hex | External Test Signal Failure at Safety Input | An error was detected in test pulse evaluation of the safety input terminals. | • The positive power supply wire is in contact with the input signal line.<br>• The input signal lines are shorted.<br>• The external device is faulty.<br>• An error occurred in the test output terminal used with the safety input terminal. | | | ○ | | | page 12-85 |
| 65230000 hex | Discrepancy Error at Safety Input | An error was detected in discrepancy evaluation of safety input terminals. | • There is a ground fault or disconnection in the input signal line.<br>• The connected device is faulty.<br>• The setting of the discrepancy time is not correct.<br>• Chattering occurred in the input signal from the external input device, such as a safety door.<br>• An error occurred in the test output terminal used with the safety input terminal. | | | ○ | | | page 12-86 |
| 65240000 hex | Overload Detected at Test Output | An overcurrent was detected at the test output terminal. | • There is a ground fault on the output signal line.<br>• The external device is faulty. | | | ○ | | | page 12-87 |

| Event code | Event name | Meaning | Assumed cause | Level | | | | | Reference |
|---|---|---|---|---|---|---|---|---|---|
| | | | | Maj | Prt | Min | Obs | Info | |
| 65250000 hex | Stuck-at-high Detected at Test Output | It was detected that the test output terminal is stuck ON. | • The positive power supply line is in contact with the output signal line.<br>• The internal circuit is faulty.<br>• A memory error or signal error occurred due to a transient cause, such as a software error or excessive noise. | | | ○ | | | page 12-88 |
| 65270000 hex | Short Circuit Detected at Safety Output | A ground fault was detected on the safety output terminal. | • There is a ground fault on the output signal line. | | | ○ | | | page 12-89 |
| 65280000 hex | Stuck-at-high Detected at Safety Output | It was detected that the safety output terminal is stuck ON. | • The positive power supply line is in contact with the output signal line.<br>• The output power supply is outside the specifications.<br>• The internal circuit is faulty.<br>• A memory error or signal error occurred due to a transient cause, such as a software error or excessive noise. | | | ○ | | | page 12-90 |

## ● Other Errors

| Event code | Event name | Meaning | Assumed cause | Level | | | | | Reference |
|---|---|---|---|---|---|---|---|---|---|
| | | | | Maj | Prt | Min | Obs | Info | |
| 80220000 hex | NX Message Communications Error | An error was detected in message communications for an NX Unit and the message frame was discarded. | • The message communications load is high.<br>• The communications cable is disconnected or broken.<br>• Message communications were cut off as the result of executing a synchronization or restoration operation on the Sysmac Studio or as the result of disconnecting an EtherCAT slave. | | | | ○ | | page 12-91 |

## ● User Access Log

| Event code | Event name | Meaning | Assumed cause | Level | | | | | Reference |
|---|---|---|---|---|---|---|---|---|---|
| | | | | Maj | Prt | Min | Obs | Info | |
| 90400000 hex | Event Log Cleared | The event log was cleared. | The event log was cleared by the user. | | | | | ○ | page 12-92 |

| Event code | Event name | Meaning | Assumed cause | Level | | | | | Reference |
|---|---|---|---|---|---|---|---|---|---|
| | | | | Maj | Prt | Min | Obs | Info | |
| 90430000 hex | Memory All Cleared | The Unit settings were cleared. | The Clear All Memory operation was performed. | | | | | ○ | page 12-92 |

## 12-3-4 Error Descriptions

This section describes the information that is given for individual errors.

### Controller Error Descriptions

The items that are used to describe individual errors (events) are described in the following copy of an error table.

| | | | | | | |
|---|---|---|---|---|---|---|
| **Event name** | Gives the name of the error. | | | **Event code** | Gives the code of the error. | |
| **Meaning** | Gives a short description of the error. | | | | | |
| **Source** | Gives the source of the error. | | **Source details** | Gives details on the source of the error. | **Detection timing** | Tells when the error is detected. |
| **Error attributes** | **Level** | Tells the level of influence on standard control.*1 | **Recovery** | Gives the recovery method.*2 | **Log category** | Tells which log the error is saved in.*3 |
| **Effects** | **User program** | Tells what will happen to execution of the user program in the NJ/NX-series CPU Unit.*4 | **Operation** | Provides specific information on the operation that results from the error. | | |
| **Indicators** | Gives the status of the indicators on the Safety Control Unit, which show status other than the event level. | | | | | |
| **System-defined variables** | **Variable** | | **Data type** | | **Name** | |
| | Lists the variable names, data types, and names for system-defined variables that provide direct error notification, that are directly affected by the error, or that contain settings that cause the error. | | | | | |
| **Cause and correction** | **Assumed cause** | | **Correction** | | **Prevention** | |
| | Lists the possible causes, corrections, and preventive measures for the error. | | | | | |
| **Attached information** | This is the attached information that is displayed by the Sysmac Studio. | | | | | |
| **Precautions/ Remarks** | Provides precautions, restrictions, and supplemental information. | | | | | |

*1. This is the level of influence on standard control, and not the level of influence on safety control.
One of the following:
Major fault: Major fault level
Partial fault: Partial fault level
Minor fault: Minor fault level
Observation
Information

*2. One of the following:
Automatic recovery: Normal status is restored automatically when the cause of the error is removed.
Error reset: Normal status is restored when the error is reset after the cause of the error is removed.
Cycle the power supply: Normal status is restored when the power supply to the Controller is turned OFF and then back ON after the cause of the error is removed.
Controller reset: Normal status is restored when the Controller is reset after the cause of the error is removed.
Controller reset: Normal status is restored when the Controller is reset after the cause of the error is removed.

*3. One of the following:
System: System event log

Access: Access event log

*4. This status is for the execution of the user program in the NJ-series CPU Unit, and not for the execution of the safety program in the Safety CPU Unit. One of the following:
Continues: Execution of the user program in the NJ/NX-series CPU Unit will continue.
Stops: Execution of the user program in the NJ/NX-series CPU Unit stops.
Starts: Execution of the user program in the NJ/NX-series CPU Unit starts.

## Safety CPU Unit

Details on the errors (events) that can occur in the Safety CPU Unit are given in the following tables.

### ● System Error

| | | | | | | |
|---|---|---|---|---|---|---|
| **Event name** | System Error | | | **Event code** | 05200000 hex | |
| **Meaning** | A hardware error was detected during self-diagnosis of the hardware. | | | | | |
| **Source** | Depends on where the Sysmac Studio is connected and the system configuration. | | **Source details** | NX Unit | **Detection timing** | Continuously |
| **Error attributes** | **Level** | Minor fault | **Recovery** | Cycle the power supply to the Unit. | **Log category** | System |
| **Effects** | **User program** | Continues. | **Operation** | The Unit stops operating and the I/O data changes to the safe states. | | |
| **System-defined variables** | **Variable** | | **Data type** | | **Name** | |
| | None | | None | | None | |
| **Cause and correction** | **Assumed cause** | | **Correction** | | **Prevention** | |
| | Hardware has failed. | | Cycle the power supply. If the error occurs again, replace the Unit. | | If cycling the power supply restores normal operation, there may be excessive noise near the Unit. Implement noise countermeasures. | |
| | A memory error occurred due to a transient cause, such as a software error or excessive noise. | | | | | |
| **Attached information** | Attached information 1: System information, status code<br>Attached information 2: System information, status code<br>Attached information 3: System information, status code<br>Attached information 4: System information, status code | | | | | |
| **Precautions/ Remarks** | None | | | | | |

## ● Communications Errors

| | | | | | | |
|---|---|---|---|---|---|---|
| **Event name** | Safety Process Data Communications Not Established Error | | | **Event code** | 35200000 hex | |
| **Meaning** | Safety process data communications was not established with one or more safety slaves. | | | | | |
| **Source** | Depends on where the Sysmac Studio is connected and the system configuration. | | **Source details** | NX Unit | **Detection timing** | In DEBUG mode (STOPPED), DEBUG mode (RUN), or RUN mode |
| **Error attributes** | **Level** | Minor fault | **Recovery** | Automatic recovery when cause of error is removed | **Log category** | System |
| **Effects** | **User program** | Continues. | **Operation** | The Unit continues to operate, but the safe states are used for the I/O data of safety connection where the error was detected. | | |
| **System-defined variables** | **Variable** | | **Data type** | | **Name** | |
| | None | | None | | None | |
| **Cause and correction** | **Assumed cause** | | **Correction** | | **Prevention** | |
| | The communications settings for safety process data are not correct, the safety slave is not in the correct status, etc. | | Refer to the error for the safety slave and correct the problem. | | Refer to the errors for the safety slaves and implement countermeasures. | |
| | The safety slave for safety process data communications is not connected. | | Make sure the safety slave is connected correctly. | | Make sure that all of the safety slaves to communicate with are connected before you change the Safety CPU Unit to DEBUG mode (STOPPED), DEBUG mode (RUN), or RUN mode. | |
| | The safety slave set in the NX Unit Mounting Setting for safety process data communications is disabled. | | Set the disabled safety slaves in the NX Unit Mounting Setting so that they do not participate in safety process data communications and then transfer the data to the Safety CPU Unit. | | Set the disabled safety slaves in the NX Unit Mounting Setting so that they do not participate in safety process data communications. | |
| **Attached information** | None | | | | | |
| **Precautions/ Remarks** | The relevant Units will maintain the safe states for I/O data with safety connections after an error is detected. However, when the cause of the error is removed, safety process data communications will recover automatically. | | | | | |

| Event name | FSoE Master Connection Not Established Error | | Event code | 35200001 hex | |
|---|---|---|---|---|---|
| Meaning | FSoE communications were not established with the FSoE slave. | | | | |
| Source | Depends on where the Sysmac Studio is connected and the system configuration. | Source details | NX Unit | Detection timing | In DEBUG mode (STOPPED), DEBUG mode (RUN), or RUN mode |
| Error attributes | Level: Minor fault | Recovery | Automatic recovery when cause of error is removed | Log category | System |
| Effects | User program: Continues. | Operation | The Unit continues to operate, but the safe states are used for the I/O data of the safety connection where the error was detected. | | |

| System-defined variables | Variable | Data type | Name |
|---|---|---|---|
| | None | None | None |

| Cause and correction | Assumed cause | Correction | Prevention |
|---|---|---|---|
| | The FSoE communications settings are not correct, the FSoE slave is not in the correct status, etc. | Refer to the error for the FSoE slave and correct the problem. | Refer to the errors for the FSoE slaves and implement countermeasures. |
| | The FSoE slave for FSoE communications is not connected. | Make sure the FSoE slave is connected correctly. | Make sure that all of the FSoE slaves to communicate with are connected before you change the Safety CPU Unit to DEBUG mode (STOPPED), DEBUG mode (RUN), or RUN mode. |
| | The FSoE slave set in the NX Unit Mounting Setting for FSoE communications is disabled. | Set the disabled FSoE slaves in the NX Unit Mounting Setting so that they do not participate in FSoE communications and then transfer the data to the Safety CPU Unit. | Set the disabled FSoE slaves in the NX Unit Mounting Setting so that they do not participate in FSoE communications. |
| | The version of the FSoE slave to be communicated is older than the configured version. | Change the version of the FSoE slave in the project to the version of the FSoE to be communicated, and then transfer the setting to the Safety CPU Unit again. Or, replace to a Unit that has a newer version than the Unit that is set with the FSoE salve to be communicated. | Keep the version of the FSoE slave in the project consistent with the version of the FSoE slave to be communicated. |

| Attached information | None |
|---|---|
| Precautions/ Remarks | The relevant Units will maintain the safe states for I/O data with FSoE connections after an error is detected. However, when the cause of the error is removed, FSoE communications will recover automatically. |

| Event name | NX Unit I/O Communications Error | | Event code | 80200000 hex | |
|---|---|---|---|---|---|
| Meaning | An I/O communications error occurred between the NX bus master and the NX Unit. | | | | |
| Source | Depends on where the Support Software is connected and the system configuration. | Source details | NX Unit | Detection timing | Continuously |
| Error attributes | Level | Minor fault | Log category | System | |
| | Recovery | [NX Bus of the CPU Unit]<br>[Fail-soft Operation Setting is *Stop*]<br>Errors are reset in the NX Bus Function Module.<br>[Fail-soft Operation Setting is *Fail-soft Operation*]<br>Errors are reset in the NX Unit.<br>[Communications Coupler Unit]<br>[Fail-soft Operation Setting is *Stop*]<br>If errors are detected in the controller, errors are reset in the controller.<br>If errors are not detected in the controller, errors are reset in the Communications Coupler Unit and the NX Unit.<br>[Fail-soft Operation Setting is *Fail-soft Operation*]<br>Errors are reset in the Communications Coupler Unit and the NX Unit. | | | |
| Effects | User program | Continues. | Operation | The NX Unit will continue to operate.<br>Input data: Updating input values stops.<br>Output data: The output values depend on the Load Rejection Output Setting. | |

| System-defined variables | Variable | Data type | Name |
|---|---|---|---|
| | None | None | None |

| Cause and correction | Assumed cause | Correction | Prevention |
|---|---|---|---|
| | [NX Bus of the CPU Unit] | | |
| | An error occurred in the CPU Unit, which prevents the NX bus communications from being carried out normally. | Check the error in the CPU Unit and try the solutions specified in Correction. | Try the solutions specified in Prevention for the error in the CPU Unit. |
| | The NX Unit is not mounted properly. | Mount the NX Units and End Cover securely and secure them with End Plates. | Mount the NX Units and End Cover securely and secure them with End Plates. |
| | The power cable for the Unit power supply is disconnected. Or, the wiring from the Unit power supply to the NX Units is incorrect. | Correctly wire the Unit power supply to the NX Units. | Correctly wire the Unit power supply to the NX Units. |
| | The power cable for the Unit power supply is broken. | If the power cable connecting the Unit power supply and the NX Units is broken, replace the power cable. | None |
| | The Unit power voltage is out of the supported range, or the Unit power capacity is not sufficient. | Configure the power supply system correctly in accordance with the power supply specifications. | Configure the power supply system correctly in accordance with the power supply specifications. |
| | There is a hardware error in the NX Unit. | If the error occurs again even after you make the above correction, replace the NX Unit. | None |

| | | | |
|---|---|---|---|
| **Cause and correction** | [Communications Coupler Unit] | | |
| | An error occurred in the Communications Coupler Unit, which prevents the NX bus communications from being carried out normally. | Check the error in the Communications Coupler Unit and try the solutions specified in Correction. | Try the solutions specified in Prevention for the error in the Communications Coupler Unit. |
| | The NX Unit is not mounted properly. | Mount the NX Units and End Cover securely and secure them with End Plates. | Mount the NX Units and End Cover securely and secure them with End Plates. |
| | The power cable for the Unit power supply is disconnected. Or, the wiring from the Unit power supply to the NX Units is incorrect. | Correctly wire the Unit power supply to the NX Units. | Correctly wire the Unit power supply to the NX Units. |
| | The power cable for the Unit power supply is broken. | If the power cable connecting the Unit power supply and the NX Units is broken, replace the power cable. | None |
| | The Unit power voltage is out of the supported range, or the Unit power capacity is not sufficient. | Configure the power supply system correctly in accordance with the power supply specifications. | Configure the power supply system correctly in accordance with the power supply specifications. |
| | There is a hardware error in the NX Unit. | If the error occurs again even after you make the above correction, replace the NX Unit. | None |
| **Attached information** | None | | |
| **Precautions/ Remarks** | None | | |

| Event name | Safety Process Data Communications Timeout | | | Event code | 80300000 hex | |
|---|---|---|---|---|---|---|
| Meaning | A communications timeout occurred in safety process data communications with the Safety Control Unit. | | | | | |
| Source | Depends on where the Sysmac Studio is connected and the system configuration. | | Source details | NX Unit | Detection timing | In DEBUG mode (STOPPED), DEBUG mode (RUN), or RUN mode |
| Error attributes | Level | Minor fault | Recovery | Automatic recovery when cause of error is removed | Log category | System |
| Effects | User program | Continues. | Operation | The Unit continues to operate, but the safe states are used for the I/O data of safety connection where the error was detected. | | |
| System-defined variables | Variable | | Data type | | Name | |
| | None | | None | | None | |
| Cause and correction | Assumed cause | | Correction | | Prevention | |
| | A setting is not correct. The setting of the safety task period is too short. | | Increase the safety task period and then transfer the settings to the Safety CPU Unit. | | Set the system configuration and setup according to the corrections that are given on the left. | |
| | There is excessive noise. | | Implement noise countermeasures. | | Implement noise countermeasures if excessive noise caused the error. | |
| | The Safety CPU Unit or safety slave entered a status where it could not continue safety process data communications. | | Check the status of the Safety CPU Unit or safety slave. | | Refer to troubleshooting information for the Safety CPU Unit or safety slave. | |
| | An error or status change occurred in the NX Bus Master to which the Unit is connected, preventing correct process data communications. | | Check the status of the NX Bus Master to which the Unit is connected. | | Set the system configuration and setup according to the corrections that are given on the left. | |
| Attached information | None | | | | | |
| Precautions/ Remarks | The relevant Units will maintain the safe states for I/O data with safety connections after an error is detected. However, when the cause of the error is removed, safety process data communications will recover automatically. | | | | | |

| | | | | | | |
|---|---|---|---|---|---|---|
| **Event name** | FSoE Master Connection Timeout | | | **Event code** | 80300001 hex | |
| **Meaning** | A communication timeout occurred in FSoE communications with the FSoE slave. | | | | | |
| **Source** | Depends on where the Sysmac Studio is connected and the system configuration. | | **Source details** | NX Unit | **Detection timing** | In DEBUG mode (STOPPED), DEBUG mode (RUN), or RUN mode |
| **Error attributes** | **Level** | Minor fault | **Recovery** | Automatic recovery when cause of error is removed | **Log category** | System |
| **Effects** | **User program** | Continues. | **Operation** | The Unit continues to operate, but the safe states are used for the I/O data of the safety connection where the error was detected. | | |
| **System-defined variables** | **Variable** | | **Data type** | | **Name** | |
| | None | | None | | None | |
| **Cause and correction** | **Assumed cause** | | **Correction** | | **Prevention** | |
| | A setting is not correct. The setting of the safety task period is too short. | | Increase the safety task period and then transfer the settings to the Safety CPU Unit. | | Set the system configuration and setup according to the corrections that are given on the left. | |
| | There is excessive noise. | | Implement noise countermeasures. | | Implement noise countermeasures if excessive noise caused the error. | |
| | The Safety CPU Unit or FSoE slave entered a state where it could not continue FSoE communications. | | Check the status of the Safety CPU Unit or FSoE slave. | | Refer to troubleshooting information for the Safety CPU Unit or FSoE slave. | |
| | The process data communications were not performed correctly because an error or status change occurred in the NX bus master to which the Unit is connected. | | Check the status of the NX bus master to which the Unit is connected. | | Set the system configuration and setup according to the corrections that are given on the left. | |
| **Attached information** | None | | | | | |
| **Precautions/ Remarks** | The relevant Units will maintain the safe states for I/O data with FSoE connections after an error is detected. However, when the cause of the error is removed, FSoE communications will recover automatically. | | | | | |

| Event name | CIP Safety Originator Connection Not Established Error | | | Event code | 80310000 hex | |
|---|---|---|---|---|---|---|
| Meaning | CIP safety originator connection was not established. | | | | | |
| Source | Depends on where the Sysmac Studio is connected and the system configuration. | | Source details | NX Unit | Detection timing | In DEBUG mode (STOPPED), DEBUG mode (RUN), or RUN mode |
| Error attributes | Level | Minor fault | Recovery | Automatic recovery when cause of error is removed | Log category | System |
| Effects | User program | Continues. | Operation | The Unit continues to operate, but the safe states are used for the I/O data of the safety connection where the error was detected. | | |

| System-defined variables | Variable | Data type | Name |
|---|---|---|---|
| | None | None | None |

| Cause and correction | Assumed cause | Correction | Prevention |
|---|---|---|---|
| | The target node is different. | Check if the target node is correct. | Set the system configuration and setup according to the corrections that are given on the left. |
| | The target node is not configured. | Check if the target node is configured. | |
| | The target node is not in the correct status. | Check if the target node is in a state where it can accept the connection. | |
| | The **Do not use** Option is selected for CIP message server setting in the EtherNet/IP Port Settings. | Check if the **Use** Option is selected for the CIP message server setting in the EtherNet/IP Port Settings. | |
| | The settings of the NX-series CPU Unit and NX-series EtherNet/IP Unit related to CIP Safety communications do not match the settings of the Safety CPU Unit. | Enable transfer of the settings for X Bus Units (settings of CIP Safety connections) in the Synchronization Window on the Sysmac Studio, and check whether they are downloaded. | |
| | Setting to use tag data link communications was made to the NX-series EtherNet/IP Unit that is included in the CIP Safety connection settings. | Do not make the setting to use tag data link communications to the NX-series EtherNet/IP Unit that is included in the CIP Safety connection settings. | |
| | The NX-series EtherNet/IP Unit with tag data link communications was added to the CIP Safety connection settings. | Do not add the NX-series EtherNet/IP Unit with tag data link communications to the CIP Safety connection settings. | |
| | The **Do not use** Option is selected for CIP Safety communications in the Built-in EtherNet/IP Port Settings. | Check if the **Use** Option is selected for the CIP safety communication setting in the Built-in EtherNet/IP Port Settings. | |

| | | |
|---|---|---|
| **Attached information** | Attached information 1: | IP address of the target node<br>(example: C0A8FA01 hex = address 192.168.250.1) |
| | Attached information 2: | General Status defined in CIP specifications |
| | Attached information 3: | Extended Status defined in CIP specifications |
| **Precautions/ Remarks** | None | |

| | | | | | | |
|---|---|---|---|---|---|---|
| **Event name** | CIP Safety Originator Connection Timeout | | **Event code** | 80320000 hex | | |
| **Meaning** | A timeout occurred in CIP safety originator connection. | | | | | |
| **Source** | Depends on where the Sysmac Studio is connected and the system configuration. | | **Source details** | NX Unit | **Detection timing** | In DEBUG mode (STOPPED), DEBUG mode (RUN), or RUN mode |
| **Error attributes** | **Level** | Minor fault | **Recovery** | Automatic recovery when cause of error is removed | **Log category** | System |
| **Effects** | **User program** | Continues. | **Operation** | The Unit continues to operate, but the safe states are used for the I/O data of the safety connection where the error was detected. | | |
| **System-defined variables** | **Variable** | | **Data type** | | **Name** | |
| | None | | None | | None | |
| **Cause and correction** | **Assumed cause** | | **Correction** | | **Prevention** | |
| | The communications cable is disconnected or broken. | | Connect the communications cable securely. | | Set the system configuration and setup according to the corrections that are given on the left. | |
| | The target node entered a status where it could not accept the connection. | | Check if the target node is in a state where it can accept the connection. | | | |
| | The timeout value for the communications settings is too small. | | Increase the timeout value in the communications setup and transfer the settings. | | | |
| | CIP message communications at the target node are stopped. | | Check if CIP message communications are started at the target node. | | | |
| | When the Packet Filter function is enabled in the EtherNet/IP Port Settings, CIP Safety packets from the target node are not allowed. | | Allow CIP Safety packets from the target node in the Packet Filter settings of the EtherNet/IP Port Settings. | | | |
| | CIP Safety packets are not allowed by the Packet Filter (Simple) or Packet Filter functions on the target node or the devices on the communication path. | | Allow CIP Safety packets in the Packet Filter (Simple) or Packet Filter functions on the target node or the devices on the communication path. | | | |
| | There is excessive noise. | | Implement noise countermeasures. | | | |
| **Attached information** | Attached information 1: | IP address of the target node (example: C0A8FA01 hex = address 192.168.250.1) | | | | |
| **Precautions/ Remarks** | None | | | | | |

| | | | | | | |
|---|---|---|---|---|---|---|
| **Event name** | CIP Safety Target Does Not Exist | | | **Event code** | 80330000 hex | |
| **Meaning** | The target node does not exist. | | | | | |
| **Source** | Depends on where the Sysmac Studio is connected and the system configuration. | | **Source details** | NX Unit | **Detection timing** | In DEBUG mode (STOPPED), DEBUG mode (RUN), or RUN mode |
| **Error attributes** | **Level** | Minor fault | **Recovery** | Automatic recovery when cause of error is removed | **Log category** | System |
| **Effects** | **User program** | Continues. | **Operation** | The Unit continues to operate, but the safe states are used for the I/O data of the safety connection where the error was detected. | | |

| | | | |
|---|---|---|---|
| **System-defined variables** | **Variable** | **Data type** | **Name** |
| | None | None | None |
| **Cause and correction** | **Assumed cause** | **Correction** | **Prevention** |
| | The communications cable is disconnected or broken. | Connect the communications cable securely. | Set the system configuration and setup according to the corrections that are given on the left. |
| | The target node entered a state where it could not accept the connection. | Check if the target node is in a state where it can accept the connection. | |
| | The timeout value for the communications settings is too small. | Increase the timeout value in the communications setup and transfer the settings. | |
| | CIP message communications at the target node are stopped. | Check if CIP message communications are started at the target node. | |
| | CIP Safety packets from the originator node are not allowed by the Packet Filter (Simple) or Packet Filter functions on the target node or the devices on the communication path. | Allow CIP Safety packets from the originator node in the Packet Filter (Simple) or Packet Filter functions on the target node or the devices on the communication path. | |
| | There is excessive noise. | Implement noise countermeasures. | |

| | | |
|---|---|---|
| **Attached information** | Attached information 1: | IP address of the target node<br>(example: C0A8FA01 hex = address 192.168.250.1) |
| | Attached information 2: | General Status defined in CIP specifications |
| | Attached information 3: | Extended Status defined in CIP specifications |
| **Precautions/ Remarks** | None | |

| | | | | | | |
|---|---|---|---|---|---|---|
| **Event name** | CIP Safety Target Connection Timeout | | | **Event code** | 80340000 hex | |
| **Meaning** | A timeout occurred in the CIP Safety Target connection. | | | | | |
| **Source** | Depends on where the Sysmac Studio is connected and the system configuration. | | **Source details** | NX Unit | **Detection timing** | In DEBUG mode (STOPPED), DEBUG mode (RUN), or RUN mode |
| **Error attributes** | **Level** | Minor fault | **Recovery** | Automatic recovery when cause of error is removed | **Log category** | System |
| **Effects** | **User program** | Continues. | **Operation** | The Unit continues to operate, but the safe states are used for the I/O data of the safety connection where the error was detected. | | |

| | | | |
|---|---|---|---|
| **System-defined variables** | **Variable** | **Data type** | **Name** |
| | None | None | None |

| | | | |
|---|---|---|---|
| **Cause and correction** | **Assumed cause** | **Correction** | **Prevention** |
| | The communications cable is disconnected or broken. | Connect the communications cable securely. | Set the system configuration and setup according to the corrections that are given on the left. |
| | The originator device entered a status where it could not accept the connection. | Check if the originator device is in a state where it can accept the connection. | |
| | The timeout value for the communications settings is too small. | Increase the timeout value in the communications setup and transfer the settings. | |
| | CIP Safety packets from the originator node are not allowed by the Packet Filter (Simple) or Packet Filter functions in the EtherNet/IP Port Settings or on the devices on the communication path. | Allow CIP Safety packets from the originator node in the Packet Filter (Simple) or Packet Filter functions in the EtherNet/IP Port Settings or on the devices on the communication path. | |
| | The CIP message server setting for the originator node is set to **Do not use**. | Set the CIP message server setting of the originator node to **Use**. | |
| | When the Packet Filter function of the originator node is enabled, CIP Safety packets from the target node are not allowed. | Allow CIP Safety packets from the target node in the Packet Filter function of the originator node. | |
| | There is excessive noise. | Implement noise countermeasures. | |
| | The **Do not use** Option is selected for CIP Safety communications in the Built-in EtherNet/IP Port Settings. | Check if the **Use** Option is selected for the CIP safety communication setting in the Built-in EtherNet/IP Port Settings. | |

| | | |
|---|---|---|
| **Attached information** | Attached information 1: | Assembly Instance No.<br>0300 hex: Input Assembly 1<br>0301 hex: Input Assembly 2<br>0302 hex: Input Assembly 3<br>03A0 hex: Output Assembly 1 |
| **Precautions/ Remarks** | None | |

<table>
<tr><td>Event name</td><td colspan="3">NX Bus I/O Communications Stopped</td><td>Event code</td><td colspan="2">84F00000 hex</td></tr>
<tr><td>Meaning</td><td colspan="6">An error occurred in I/O communications between the NX bus master and an NX Unit.</td></tr>
<tr><td>Source</td><td colspan="2">Depends on where the Sysmac Studio is connected and the system configuration.</td><td>Source details</td><td>NX Unit</td><td>Detection timing</td><td>Continuously</td></tr>
<tr><td>Error attributes</td><td>Level</td><td>Minor fault</td><td>Recovery</td><td>Cycle the power supply to the NX bus master and NX Units.</td><td>Log category</td><td>System</td></tr>
<tr><td>Effects</td><td>User program</td><td>Continues.</td><td>Operation</td><td colspan="3">The NX Units will continue to operate.<br>• Input data:<br>An error occurs in safety process data communications because refreshing is stopped. The values of the status and exposed variables in the standard process data are not refreshed.<br>• Output data:<br>An error occurs in safety process data communications because 0’s are output.</td></tr>
<tr><td rowspan="2">System-defined variables</td><td colspan="2">Variable</td><td colspan="2">Data type</td><td colspan="2">Name</td></tr>
<tr><td colspan="2">None</td><td colspan="2">None</td><td colspan="2">None</td></tr>
<tr><td rowspan="2">Cause and correction</td><td colspan="2">Assumed cause</td><td colspan="2">Correction</td><td colspan="2">Prevention</td></tr>
<tr><td colspan="2">There is a hardware error in the NX bus master or an NX Unit.</td><td colspan="2">If the error occurs again even after you cycle the power supply to the NX Units, replace the NX bus master or the NX Unit.</td><td colspan="2">None</td></tr>
<tr><td>Attached information</td><td colspan="6">None</td></tr>
<tr><td>Precautions/ Remarks</td><td colspan="6">None</td></tr>
</table>

## ● Operation Continuation Error Related to Program Execution Function

| | | | | | | |
|---|---|---|---|---|---|---|
| **Event name** | SF_Antivalent Error | | | **Event code** | 74A00000 hex | |
| **Meaning** | An error was detected in execution of a safety function block. | | | | | |
| **Source** | Depends on where the Sysmac Studio is connected and the system configuration. | | **Source details** | NX Unit | **Detection timing** | In DEBUG mode (RUN) or RUN mode |
| **Error attributes** | **Level** | Minor fault | **Recovery** | Implement the correction. | **Log category** | System |
| **Effects** | **User program** | Continues. | **Operation** | The Unit continues to operate. | | |
| **System-defined variables** | **Variable** | | **Data type** | | **Name** | |
| | None | | None | | None | |
| **Cause and correction** | **Assumed cause** | | **Correction** | | **Prevention** | |
| | Refer to information on the diagnostic code that is given for attached information 1 in the *NX-series Safety Control Unit Instructions Reference Manual (Cat. No. Z931)*. | | Implement the correction for the relevant cause of the diagnostic code that is given for attached information 1. | | Program operation considering the corrections that are given on the left. | |
| **Attached information** | Attached information 1: Diagnostic code | | | | | |
| **Precautions/ Remarks** | None | | | | | |

| | | | | | | |
|---|---|---|---|---|---|---|
| **Event name** | SF_EDM Error | | | **Event code** | 74A10000 hex | |
| **Meaning** | An error was detected in execution of a safety function block. | | | | | |
| **Source** | Depends on where the Sysmac Studio is connected and the system configuration. | | **Source details** | NX Unit | **Detection timing** | In DEBUG mode (RUN) or RUN mode |
| **Error attributes** | **Level** | Minor fault | **Recovery** | Implement the correction. | **Log category** | System |
| **Effects** | **User program** | Continues. | **Operation** | The Unit continues to operate. | | |
| **System-defined variables** | **Variable** | | **Data type** | | **Name** | |
| | None | | None | | None | |
| **Cause and correction** | **Assumed cause** | | **Correction** | | **Prevention** | |
| | Refer to information on the diagnostic code that is given for attached information 1 in the *NX-series Safety Control Unit Instructions Reference Manual (Cat. No. Z931)*. | | Implement the correction for the relevant cause of the diagnostic code that is given for attached information 1. | | Program operation considering the corrections that are given on the left. | |
| **Attached information** | Attached information 1: Diagnostic code | | | | | |
| **Precautions/ Remarks** | None | | | | | |

| Event name | SF_EmergencyStop Error | | Event code | 74A20000 hex | | |
|---|---|---|---|---|---|---|
| Meaning | An error was detected in execution of a safety function block. | | | | | |
| Source | Depends on where the Sysmac Studio is connected and the system configuration. | | Source details | NX Unit | Detection timing | In DEBUG mode (RUN) or RUN mode |
| Error attributes | Level | Minor fault | Recovery | Implement the correction. | Log category | System |
| Effects | User program | Continues. | Operation | The Unit continues to operate. | | |
| System-defined variables | Variable | | Data type | | Name | |
| | None | | None | | None | |
| Cause and correction | Assumed cause | | Correction | | Prevention | |
| | Refer to information on the diagnostic code that is given for attached information 1 in the *NX-series Safety Control Unit Instructions Reference Manual (Cat. No. Z931)*. | | Implement the correction for the relevant cause of the diagnostic code that is given for attached information 1. | | Program operation considering the corrections that are given on the left. | |
| Attached information | Attached information 1: Diagnostic code | | | | | |
| Precautions/ Remarks | None | | | | | |

| Event name | SF_EnableSwitch Error | | Event code | 74A30000 hex | | |
|---|---|---|---|---|---|---|
| Meaning | An error was detected in execution of a safety function block. | | | | | |
| Source | Depends on where the Sysmac Studio is connected and the system configuration. | | Source details | NX Unit | Detection timing | In DEBUG mode (RUN) or RUN mode |
| Error attributes | Level | Minor fault | Recovery | Implement the correction. | Log category | System |
| Effects | User program | Continues. | Operation | The Unit continues to operate. | | |
| System-defined variables | Variable | | Data type | | Name | |
| | None | | None | | None | |
| Cause and correction | Assumed cause | | Correction | | Prevention | |
| | Refer to information on the diagnostic code that is given for attached information 1 in the *NX-series Safety Control Unit Instructions Reference Manual (Cat. No. Z931)*. | | Implement the correction for the relevant cause of the diagnostic code that is given for attached information 1. | | Program operation considering the corrections that are given on the left. | |
| Attached information | Attached information 1: Diagnostic code | | | | | |
| Precautions/ Remarks | None | | | | | |

| Event name | SF_Equivalent Error | | | Event code | | 74A40000 hex |
|---|---|---|---|---|---|---|
| Meaning | An error was detected in execution of a safety function block. | | | | | |
| Source | Depends on where the Sysmac Studio is connected and the system configuration. | | Source details | NX Unit | Detection timing | In DEBUG mode (RUN) or RUN mode |
| Error attributes | Level | Minor fault | Recovery | Implement the correction. | Log category | System |
| Effects | User program | Continues. | Operation | The Unit continues to operate. | | |
| System-defined variables | Variable | | Data type | | Name | |
| | None | | None | | None | |
| Cause and correction | Assumed cause | | Correction | | Prevention | |
| | Refer to information on the diagnostic code that is given for attached information 1 in the *NX-series Safety Control Unit Instructions Reference Manual (Cat. No. Z931)*. | | Implement the correction for the relevant cause of the diagnostic code that is given for attached information 1. | | Program operation considering the corrections that are given on the left. | |
| Attached information | Attached information 1: Diagnostic code | | | | | |
| Precautions/ Remarks | None | | | | | |

| Event name | SF_ESPE Error | | | Event code | | 74A50000 hex |
|---|---|---|---|---|---|---|
| Meaning | An error was detected in execution of a safety function block. | | | | | |
| Source | Depends on where the Sysmac Studio is connected and the system configuration. | | Source details | NX Unit | Detection timing | In DEBUG mode (RUN) or RUN mode |
| Error attributes | Level | Minor fault | Recovery | Implement the correction. | Log category | System |
| Effects | User program | Continues. | Operation | The Unit continues to operate. | | |
| System-defined variables | Variable | | Data type | | Name | |
| | None | | None | | None | |
| Cause and correction | Assumed cause | | Correction | | Prevention | |
| | Refer to information on the diagnostic code that is given for attached information 1 in the *NX-series Safety Control Unit Instructions Reference Manual (Cat. No. Z931)*. | | Implement the correction for the relevant cause of the diagnostic code that is given for attached information 1. | | Program operation considering the corrections that are given on the left. | |
| Attached information | Attached information 1: Diagnostic code | | | | | |
| Precautions/ Remarks | None | | | | | |

| Event name | SF_GuardLocking Error | | | Event code | 74A60000 hex | |
|---|---|---|---|---|---|---|
| Meaning | An error was detected in execution of a safety function block. | | | | | |
| Source | Depends on where the Sysmac Studio is connected and the system configuration. | | Source details | NX Unit | Detection timing | In DEBUG mode (RUN) or RUN mode |
| Error attributes | Level | Minor fault | Recovery | Implement the correction. | Log category | System |
| Effects | User program | Continues. | Operation | The Unit continues to operate. | | |
| System-defined variables | Variable | | Data type | | Name | |
| | None | | None | | None | |
| Cause and correction | Assumed cause | | Correction | | Prevention | |
| | Refer to information on the diagnostic code that is given for attached information 1 in the *NX-series Safety Control Unit Instructions Reference Manual (Cat. No. Z931)*. | | Implement the correction for the relevant cause of the diagnostic code that is given for attached information 1. | | Program operation considering the corrections that are given on the left. | |
| Attached information | Attached information 1: Diagnostic code | | | | | |
| Precautions/ Remarks | None | | | | | |

| Event name | SF_GuardMonitoring Error | | | Event code | 74A70000 hex | |
|---|---|---|---|---|---|---|
| Meaning | An error was detected in execution of a safety function block. | | | | | |
| Source | Depends on where the Sysmac Studio is connected and the system configuration. | | Source details | NX Unit | Detection timing | In DEBUG mode (RUN) or RUN mode |
| Error attributes | Level | Minor fault | Recovery | Implement the correction. | Log category | System |
| Effects | User program | Continues. | Operation | The Unit continues to operate. | | |
| System-defined variables | Variable | | Data type | | Name | |
| | None | | None | | None | |
| Cause and correction | Assumed cause | | Correction | | Prevention | |
| | Refer to information on the diagnostic code that is given for attached information 1 in the *NX-series Safety Control Unit Instructions Reference Manual (Cat. No. Z931)*. | | Implement the correction for the relevant cause of the diagnostic code that is given for attached information 1. | | Program operation considering the corrections that are given on the left. | |
| Attached information | Attached information 1: Diagnostic code | | | | | |
| Precautions/ Remarks | None | | | | | |

| Event name | SF_ModeSelector Error | | | Event code | | 74A80000 hex |
|---|---|---|---|---|---|---|
| Meaning | An error was detected in execution of a safety function block. | | | | | |
| Source | Depends on where the Sysmac Studio is connected and the system configuration. | | Source details | NX Unit | Detection timing | In DEBUG mode (RUN) or RUN mode |
| Error attributes | Level | Minor fault | Recovery | Implement the correction. | Log category | System |
| Effects | User program | Continues. | Operation | The Unit continues to operate. | | |
| System-defined variables | Variable | | Data type | | Name | |
| | None | | None | | None | |
| Cause and correction | Assumed cause | | Correction | | Prevention | |
| | Refer to information on the diagnostic code that is given for attached information 1 in the *NX-series Safety Control Unit Instructions Reference Manual (Cat. No. Z931)*. | | Implement the correction for the relevant cause of the diagnostic code that is given for attached information 1. | | Program operation considering the corrections that are given on the left. | |
| Attached information | Attached information 1: Diagnostic code | | | | | |
| Precautions/ Remarks | None | | | | | |

| Event name | SF_MutingPar Error | | | Event code | | 74A90000 hex |
|---|---|---|---|---|---|---|
| Meaning | An error was detected in execution of a safety function block. | | | | | |
| Source | Depends on where the Sysmac Studio is connected and the system configuration. | | Source details | NX Unit | Detection timing | In DEBUG mode (RUN) or RUN mode |
| Error attributes | Level | Minor fault | Recovery | Implement the correction. | Log category | System |
| Effects | User program | Continues. | Operation | The Unit continues to operate. | | |
| System-defined variables | Variable | | Data type | | Name | |
| | None | | None | | None | |
| Cause and correction | Assumed cause | | Correction | | Prevention | |
| | Refer to information on the diagnostic code that is given for attached information 1 in the *NX-series Safety Control Unit Instructions Reference Manual (Cat. No. Z931)*. | | Implement the correction for the relevant cause of the diagnostic code that is given for attached information 1. | | Program operation considering the corrections that are given on the left. | |
| Attached information | Attached information 1: Diagnostic code | | | | | |
| Precautions/ Remarks | None | | | | | |

| Event name | SF_MutingPar_2Sensor Error | | | Event code | | 74AA0000 hex |
|---|---|---|---|---|---|---|
| Meaning | An error was detected in execution of a safety function block. | | | | | |
| Source | Depends on where the Sysmac Studio is connected and the system configuration. | | Source details | NX Unit | Detection timing | In DEBUG mode (RUN) or RUN mode |
| Error attributes | Level | Minor fault | Recovery | Implement the correction. | Log category | System |
| Effects | User program | Continues. | Operation | The Unit continues to operate. | | |
| System-defined variables | Variable | | Data type | | Name | |
| | None | | None | | None | |
| Cause and correction | Assumed cause | | Correction | | Prevention | |
| | Refer to information on the diagnostic code that is given for attached information 1 in the *NX-series Safety Control Unit Instructions Reference Manual (Cat. No. Z931)*. | | Implement the correction for the relevant cause of the diagnostic code that is given for attached information 1. | | Program operation considering the corrections that are given on the left. | |
| Attached information | Attached information 1: Diagnostic code | | | | | |
| Precautions/ Remarks | None | | | | | |

| Event name | SF_MutingSeq Error | | | Event code | | 74AB0000 hex |
|---|---|---|---|---|---|---|
| Meaning | An error was detected in execution of a safety function block. | | | | | |
| Source | Depends on where the Sysmac Studio is connected and the system configuration. | | Source details | NX Unit | Detection timing | In DEBUG mode (RUN) or RUN mode |
| Error attributes | Level | Minor fault | Recovery | Implement the correction. | Log category | System |
| Effects | User program | Continues. | Operation | The Unit continues to operate. | | |
| System-defined variables | Variable | | Data type | | Name | |
| | None | | None | | None | |
| Cause and correction | Assumed cause | | Correction | | Prevention | |
| | Refer to information on the diagnostic code that is given for attached information 1 in the *NX-series Safety Control Unit Instructions Reference Manual (Cat. No. Z931)*. | | Implement the correction for the relevant cause of the diagnostic code that is given for attached information 1. | | Program operation considering the corrections that are given on the left. | |
| Attached information | Attached information 1: Diagnostic code | | | | | |
| Precautions/ Remarks | None | | | | | |

| Event name | SF_OutControl Error | | | Event code | | 74AC0000 hex |
|---|---|---|---|---|---|---|
| Meaning | An error was detected in execution of a safety function block. | | | | | |
| Source | Depends on where the Sysmac Studio is connected and the system configuration. | | Source details | NX Unit | Detection timing | In DEBUG mode (RUN) or RUN mode |
| Error attributes | Level | Minor fault | Recovery | Implement the correction. | Log category | System |
| Effects | User program | Continues. | Operation | The Unit continues to operate. | | |
| System-defined variables | Variable | | Data type | | Name | |
| | None | | None | | None | |
| Cause and correction | Assumed cause | | Correction | | Prevention | |
| | Refer to information on the diagnostic code that is given for attached information 1 in the *NX-series Safety Control Unit Instructions Reference Manual (Cat. No. Z931)*. | | Implement the correction for the relevant cause of the diagnostic code that is given for attached information 1. | | Program operation considering the corrections that are given on the left. | |
| Attached information | Attached information 1: Diagnostic code | | | | | |
| Precautions/ Remarks | None | | | | | |

| Event name | SF_SafetyRequest Error | | | Event code | | 74AD0000 hex |
|---|---|---|---|---|---|---|
| Meaning | An error was detected in execution of a safety function block. | | | | | |
| Source | Depends on where the Sysmac Studio is connected and the system configuration. | | Source details | NX Unit | Detection timing | In DEBUG mode (RUN) or RUN mode |
| Error attributes | Level | Minor fault | Recovery | Implement the correction. | Log category | System |
| Effects | User program | Continues. | Operation | The Unit continues to operate. | | |
| System-defined variables | Variable | | Data type | | Name | |
| | None | | None | | None | |
| Cause and correction | Assumed cause | | Correction | | Prevention | |
| | Refer to information on the diagnostic code that is given for attached information 1 in the *NX-series Safety Control Unit Instructions Reference Manual (Cat. No. Z931)*. | | Implement the correction for the relevant cause of the diagnostic code that is given for attached information 1. | | Program operation considering the corrections that are given on the left. | |
| Attached information | Attached information 1: Diagnostic code | | | | | |
| Precautions/ Remarks | None | | | | | |

| Event name | SF_TestableSafetySensor Error | | Event code | 74AE0000 hex | | |
|---|---|---|---|---|---|---|
| Meaning | An error was detected in execution of a safety function block. | | | | | |
| Source | Depends on where the Sysmac Studio is connected and the system configuration. | | Source details | NX Unit | Detection timing | In DEBUG mode (RUN) or RUN mode |
| Error attributes | Level | Minor fault | Recovery | Implement the correction. | Log category | System |
| Effects | User program | Continues. | Operation | The Unit continues to operate. | | |
| System-defined variables | Variable | | Data type | | Name | |
| | None | | None | | None | |
| Cause and correction | Assumed cause | | Correction | | Prevention | |
| | Refer to information on the diagnostic code that is given for attached information 1 in the *NX-series Safety Control Unit Instructions Reference Manual (Cat. No. Z931)*. | | Implement the correction for the relevant cause of the diagnostic code that is given for attached information 1. | | Program operation considering the corrections that are given on the left. | |
| Attached information | Attached information 1: Diagnostic code | | | | | |
| Precautions/ Remarks | None | | | | | |

| Event name | SF_TwoHandControlTypeII Error | | Event code | 74AF0000 hex | | |
|---|---|---|---|---|---|---|
| Meaning | An error was detected in execution of a safety function block. | | | | | |
| Source | Depends on where the Sysmac Studio is connected and the system configuration. | | Source details | NX Unit | Detection timing | In DEBUG mode (RUN) or RUN mode |
| Error attributes | Level | Minor fault | Recovery | Implement the correction. | Log category | System |
| Effects | User program | Continues. | Operation | The Unit continues to operate. | | |
| System-defined variables | Variable | | Data type | | Name | |
| | None | | None | | None | |
| Cause and correction | Assumed cause | | Correction | | Prevention | |
| | Refer to information on the diagnostic code that is given for attached information 1 in the *NX-series Safety Control Unit Instructions Reference Manual (Cat. No. Z931)*. | | Implement the correction for the relevant cause of the diagnostic code that is given for attached information 1. | | Program operation considering the corrections that are given on the left. | |
| Attached information | Attached information 1: Diagnostic code | | | | | |
| Precautions/ Remarks | None | | | | | |

| Event name | SF_TwoHandControlTypeIII Error | | | Event code | 74B00000 hex | |
|---|---|---|---|---|---|---|
| Meaning | An error was detected in execution of a safety function block. | | | | | |
| Source | Depends on where the Sysmac Studio is connected and the system configuration. | | Source details | NX Unit | Detection timing | In DEBUG mode (RUN) or RUN mode |
| Error attributes | Level | Minor fault | Recovery | Implement the correction. | Log category | System |
| Effects | User program | Continues. | Operation | The Unit continues to operate. | | |
| System-defined variables | Variable | | Data type | | Name | |
| | None | | None | | None | |
| Cause and correction | Assumed cause | | Correction | | Prevention | |
| | Refer to information on the diagnostic code that is given for attached information 1 in the *NX-series Safety Control Unit Instructions Reference Manual (Cat. No. Z931)*. | | Implement the correction for the relevant cause of the diagnostic code that is given for attached information 1. | | Program operation considering the corrections that are given on the left. | |
| Attached information | Attached information 1: Diagnostic code | | | | | |
| Precautions/ Remarks | None | | | | | |

## ● Operation Stop Error Related to Program Execution

<table>
<tr><td>Event name</td><td colspan="3">Division by Zero</td><td>Event code</td><td colspan="2">55000000 hex</td></tr>
<tr><td>Meaning</td><td colspan="6">Division by zero was detected.</td></tr>
<tr><td>Source</td><td colspan="2">Depends on where the Sysmac Studio is connected and the system configuration.</td><td>Source details</td><td>NX Unit</td><td>Detection timing</td><td>In DEBUG mode (RUN) or RUN mode</td></tr>
<tr><td rowspan="2">Error attributes</td><td>Level</td><td>Minor fault</td><td>Recovery</td><td>Automatic recovery</td><td>Log category</td><td>System</td></tr>
<tr><td>Effects</td><td>User program</td><td>Continues.</td><td>Operation</td><td colspan="3">The CPU Unit executes NX bus communications but execution of the user program stops. (All I/O data will remain at 0.)</td></tr>
<tr><td rowspan="2">System-defined variables</td><td colspan="2">Variable</td><td colspan="2">Data type</td><td colspan="2">Name</td></tr>
<tr><td colspan="2">None</td><td colspan="2">None</td><td colspan="2">None</td></tr>
<tr><td rowspan="2">Cause and correction</td><td colspan="2">Assumed cause</td><td colspan="2">Correction</td><td colspan="2">Prevention</td></tr>
<tr><td colspan="2">The divisor is zero.</td><td colspan="2">Correct the program so that the divisor is not 0.<br>Perform the following corrections according to the operating mode of the Safety CPU Unit.<br>• RUN mode: Change to PROGRAM mode and transfer the corrected user program.<br>• DEBUG mode (RUN): Change to PROGRAM mode and transfer the corrected user program.</td><td colspan="2">Program operation considering the corrections that are given on the left.</td></tr>
<tr><td>Attached information</td><td colspan="6">None</td></tr>
<tr><td>Precautions/ Remarks</td><td colspan="6">The CPU Unit executes NX bus communications but execution of the user program stops. (All I/O data will remain at 0.)</td></tr>
</table>

<table>
<tr><td>Event name</td><td colspan="3">Cast Error</td><td>Event code</td><td colspan="2">55010000 hex</td></tr>
<tr><td>Meaning</td><td colspan="6">A casting error was detected.</td></tr>
<tr><td>Source</td><td colspan="2">Depends on where the Sysmac Studio is connected and the system configuration.</td><td>Source details</td><td>NX Unit</td><td>Detection timing</td><td>In DEBUG mode (RUN) or RUN mode</td></tr>
<tr><td>Error attributes</td><td>Level</td><td>Minor fault</td><td>Recovery</td><td>Automatic recovery</td><td>Log category</td><td>System</td></tr>
<tr><td>Effects</td><td>User program</td><td>Continues.</td><td>Operation</td><td colspan="3">The CPU Unit executes NX bus communications but execution of the user program stops. (All I/O data will remain at 0.)</td></tr>
<tr><td rowspan="2">System-defined variables</td><td colspan="2">Variable</td><td colspan="2">Data type</td><td colspan="2">Name</td></tr>
<tr><td colspan="2">None</td><td colspan="2">None</td><td colspan="2">None</td></tr>
<tr><td rowspan="2">Cause and correction</td><td colspan="2">Assumed cause</td><td colspan="2">Correction</td><td colspan="2">Prevention</td></tr>
<tr><td colspan="2">A value was input that exceeded the range of the receiving variable.</td><td colspan="2">Do not allow the value to exceed the range of the receiving variable.<br>Perform the following corrections according to the operating mode of the Safety CPU Unit.<br>• RUN mode: Change to PROGRAM mode and transfer the corrected user program.<br>• DEBUG mode (RUN): Change to PROGRAM mode and transfer the corrected user program.</td><td colspan="2">Program operation considering the corrections that are given on the left.</td></tr>
<tr><td>Attached information</td><td colspan="6">Attached information 1: Error details<br><br>0x01000ADF: The conversion between the signed and unsigned data types was failed.<br>0x01000AE0: The positive upper limit of the data type after conversion was exceeded.<br>0x01000AE1: The negative upper limit of the data type after conversion was exceeded.</td></tr>
<tr><td>Precautions/ Remarks</td><td colspan="6">The CPU Unit executes NX bus communications but execution of the user program stops. (All I/O data will remain at 0.)</td></tr>
</table>

<table>
<tr><td>Event name</td><td colspan="3">MUX Error</td><td>Event code</td><td colspan="2">55020000 hex</td></tr>
<tr><td>Meaning</td><td colspan="6">An MUX instruction error was detected.</td></tr>
<tr><td>Source</td><td colspan="2">Depends on where the Sysmac Studio is connected and the system configuration.</td><td>Source details</td><td>NX Unit</td><td>Detection timing</td><td>In DEBUG mode (RUN) or RUN mode</td></tr>
<tr><td>Error attributes</td><td>Level</td><td>Minor fault</td><td>Recovery</td><td>Automatic recovery</td><td>Log category</td><td>System</td></tr>
<tr><td>Effects</td><td>User program</td><td>Continues.</td><td>Operation</td><td colspan="3">The CPU Unit executes NX bus communications but execution of the user program stops. (All I/O data will remain at 0.)</td></tr>
<tr><td rowspan="2">System-defined variables</td><td colspan="2">Variable</td><td colspan="2">Data type</td><td colspan="2">Name</td></tr>
<tr><td colspan="2">None</td><td colspan="2">None</td><td colspan="2">None</td></tr>
<tr><td rowspan="2">Cause and correction</td><td colspan="2">Assumed cause</td><td colspan="2">Correction</td><td colspan="2">Prevention</td></tr>
<tr><td colspan="2">The value of the selection input (K) to the MUX instruction is not correct.</td><td colspan="2">Correct the program so that the value of the selection input (K) to the MUX instruction is in range.<br>Perform the following corrections according to the operating mode of the Safety CPU Unit.<br>• RUN mode: Change to PROGRAM mode and transfer the corrected user program.<br>• DEBUG mode (RUN): Change to PROGRAM mode and transfer the corrected user program.</td><td colspan="2">Program operation considering the corrections that are given on the left.</td></tr>
<tr><td>Attached information</td><td>Attached information 1:</td><td colspan="5">0x01000ADD: The value of the selection input (K) is negative.<br>0x01000ADE: The value of the selection input (K) exceeded the upper limit of the selection range.</td></tr>
<tr><td>Precautions/ Remarks</td><td colspan="6">The CPU Unit executes NX bus communications but execution of the user program stops. (All I/O data will remain at 0.)</td></tr>
</table>

## ● Setting Errors

| Event name | NX Bus Communications Settings Read Error | | | Event code | 10500000 hex | |
|---|---|---|---|---|---|---|
| Meaning | There is an error in the NX Bus communications settings that are saved in non-volatile memory. | | | | | |
| Source | Depends on where the Sysmac Studio is connected and the system configuration. | | Source details | NX Unit | Detection timing | At power ON or restart |
| Error attributes | Level | Minor fault | Recovery | When settings are transferred | Log category | System |
| Effects | User program | Continues. | Operation | I/O refreshing stops for the CPU Unit. | | |
| System-defined variables | Variable | | Data type | | Name | |
| | None | | None | | None | |
| Cause and correction | Assumed cause | | Correction | | Prevention | |
| | A hardware failure occurred in the non-volatile memory. | | Transfer the configuration information to the Safety CPU Unit again. Replace the CPU Unit if the error occurs again. | | None | |
| | Power was turned OFF while saving data to the non-volatile memory. | | | | Do not turn OFF the power supply while transferring parameters from the Sysmac Studio. | |
| Attached information | None | | | | | |
| Precautions/ Remarks | None | | | | | |

| Event name | Safety Application Data Read Error | | | Event code | 10510000 hex | |
|---|---|---|---|---|---|---|
| Meaning | There is an error in the safety application data that is saved in non-volatile memory. | | | | | |
| Source | Depends on where the Sysmac Studio is connected and the system configuration. | | Source details | NX Unit | Detection timing | At power ON or restart |
| Error attributes | Level | Minor fault | Recovery | When settings are transferred | Log category | System |
| Effects | User program | Continues. | Operation | The safety program is not executed in the Safety CPU Unit and it operates in PROGRAM mode. | | |
| System-defined variables | Variable | | Data type | | Name | |
| | None | | None | | None | |
| Cause and correction | Assumed cause | | Correction | | Prevention | |
| | A hardware failure occurred in the non-volatile memory. | | Transfer the safety application data to the Safety CPU Unit again. | | None | |
| | Power was turned OFF while saving data to the non-volatile memory. | | | | Do not turn OFF the power supply while transferring parameters from the Sysmac Studio. | |
| Attached information | None | | | | | |
| Precautions/ Remarks | None | | | | | |

| | | | | | | |
|---|---|---|---|---|---|---|
| **Event name** | NX Bus Communications Settings and Safety Application Data Mismatch | | **Event code** | 10520000 hex | | |
| **Meaning** | There is an error in the safety application data that is saved in non-volatile memory. | | | | | |
| **Source** | Depends on where the Sysmac Studio is connected and the system configuration. | | **Source details** | NX Unit | **Detection timing** | When applicable |
| **Error attributes** | **Level** | Minor fault | **Recovery** | When settings are transferred | **Log category** | System |
| **Effects** | **User program** | Continues. | **Operation** | The CPU Unit executes NX bus communications but refreshing for the user program stops. (All I/O data will remain at 0.) | | |
| **System-defined variables** | **Variable** | | **Data type** | | **Name** | |
| | None | | None | | None | |
| **Cause and correction** | **Assumed cause** | | **Correction** | | **Prevention** | |
| | The NX bus communications settings that were transferred to the Safety CPU Unit do not match the safety application data. | | Transfer the Slave Terminal configuration information and Safety Application Data to the Safety CPU Unit again. | | None | |
| **Attached information** | None | | | | | |
| **Precautions/ Remarks** | None | | | | | |

When *NX Bus Communications Settings and Safety Application Data Mismatch* with regular troubleshooting, refer to *A-18 Safety: Update Configurations and Setup Transfer Data* on page A-109.

| Event name | Non-volatile Memory Access Error | | | Event code | 10530000 hex | |
|---|---|---|---|---|---|---|
| Meaning | Reading/writing non-volatile memory failed. | | | | | |
| Source | Depends on where the Sysmac Studio is connected and the system configuration. | | Source details | NX Unit | Detection timing | When power is turned ON to the NX Unit, when the safety application is transferred, when Clear All Memory is executed for NX Unit |
| Error attributes | Level | Minor fault | Recovery | Clear All Memory operation for the Unit | Log category | System |
| Effects | User program | Continues. | Operation | The Unit continues to operate. | | |
| System-defined variables | Variable | | Data type | | Name | |
| | None | | --- | | --- | |
| Cause and correction | Assumed cause | | Correction | | Prevention | |
| | Non-volatile memory failure | | Perform the Clear All Memory operation or download the settings again. Replace the CPU Unit if the error occurs again. | | None | |
| Attached information | None | | | | | |
| Precautions/ Remarks | None | | | | | |

| Event name | Incorrect DIP Switch Setting | | | Event code | 24AA0000 hex | |
|---|---|---|---|---|---|---|
| Meaning | The DIP switch setting is not correct. | | | | | |
| Source | Depends on where the Sysmac Studio is connected and the system configuration. | | Source details | NX Unit | Detection timing | At startup |
| Error attributes | Level | Observation | Recovery | --- | Log category | System |
| Effects | User program | Continues. | Operation | --- | | |
| System-defined variables | Variable | | Data type | | Name | |
| | None | | None | | None | |
| Cause and correction | Assumed cause | | Correction | | Prevention | |
| | The DIP switch setting was changed to an incorrect value. | | Check the DIP switch setting and change the value to a correct value. | | If you change the DIP switch setting, make sure that the DIP switch setting that you changed is correct. | |
| Attached information | Attached information 1: | DIP switch set value | | | | |
| Precautions/ Remarks | None | | | | | |

## ● Errors Related to Restore Function

| | | | | | | |
|---|---|---|---|---|---|---|
| **Event name** | Safety Unit Restore Operation Failed to Start (SD Memory Card Access Failed) | | | **Event code** | 35250000 hex | |
| **Meaning** | Access to the SD Memory Card failed when you start the restore operation for a Safety Unit. | | | | | |
| **Source** | Depends on where the Sysmac Studio is connected and the system configuration. | | **Source details** | NX Unit | **Detection timing** | When Safety Unit restoring data is specified by the user |
| **Error attributes** | **Level** | Observation | **Recovery** | --- | **Log category** | System |
| **Effects** | **User program** | Continues. | **Operation** | --- | | |
| **System-defined variables** | **Variable** | | **Data type** | | **Name** | |
| | None | | None | | None | |
| **Cause and correction** | **Assumed cause** | | **Correction** | | **Prevention** | |
| | An SD Memory Card is not inserted. | | Insert an SD Memory Card. | | Insert an SD Memory Card. | |
| | The SD Memory Card type is not correct. | | Replace the SD Memory Card with an SD or SDHC card. | | Use an SD or SDHC card. | |
| | The format of the SD Memory Card is not correct. | | Format the SD Memory Card with the Sysmac Studio. | | Use a formatted SD Memory Card. | |
| | The SD Memory Card is damaged. | | If none of these causes apply, replace the SD Memory Card. | | Do not remove the SD Memory Card or turn OFF the power supply while the SD BUSY indicator is lit. Replace the SD Memory Card periodically according to the write life of the SD Memory Card. | |
| **Attached information** | None | | | | | |
| **Precautions/ Remarks** | None | | | | | |

| | | | | | | |
|---|---|---|---|---|---|---|
| **Event name** | Safety Unit Restore Operation Failed to Start (Safety Unit Restore File Read Failure) | | | **Event code** | 35260000 hex | |
| **Meaning** | Reading the Safety Unit Restore File failed. | | | | | |
| **Source** | Depends on where the Sysmac Studio is connected and the system configuration. | | **Source details** | NX Unit | **Detection timing** | When Safety Unit restoring data is specified by the user |
| **Error attributes** | **Level** | Observation | **Recovery** | --- | **Log category** | System |
| **Effects** | **User program** | Continues. | **Operation** | --- | | |
| **System-defined variables** | **Variable** | | **Data type** | | **Name** | |
| | None | | None | | None | |
| **Cause and correction** | **Assumed cause** | | **Correction** | | **Prevention** | |
| | The Safety Unit Restore File is not stored in the specified folder. | | Store the Safety Unit Restore File in the specified folder again. | | Use a formatted SD Memory Card and store the Safety Unit Restore File in the specified folder. Do not edit the Safety Unit Restore File.<br>Do not remove the SD Memory Card or turn OFF the power supply while the SD BUSY indicator is lit. | |
| | The Safety Unit Restore File is corrupted. | | Create the Safety Unit Restore File again. | | | |
| **Attached information** | None | | | | | |
| **Precautions/ Remarks** | None | | | | | |

<table>
<tr><td>Event name</td><td colspan="3">Safety Unit Restore Operation Failed to Start (Model Mismatch)</td><td>Event code</td><td colspan="2">35270000 hex</td></tr>
<tr><td>Meaning</td><td colspan="6">A model mismatch was detected during pre-execution checks for a restore operation for a Safety Unit.</td></tr>
<tr><td>Source</td><td colspan="2">Depends on where the Sysmac Studio is connected and the system configuration.</td><td>Source details</td><td>NX Unit</td><td>Detection timing</td><td>When Safety Unit restoring data is specified by the user</td></tr>
<tr><td>Error attributes</td><td>Level</td><td>Observation</td><td>Recovery</td><td>---</td><td>Log category</td><td>System</td></tr>
<tr><td>Effects</td><td>User program</td><td>Continues.</td><td>Operation</td><td colspan="3">---</td></tr>
<tr><td rowspan="2">System-defined variables</td><td colspan="2">Variable</td><td colspan="2">Data type</td><td colspan="2">Name</td></tr>
<tr><td colspan="2">None</td><td colspan="2">None</td><td colspan="2">None</td></tr>
<tr><td rowspan="3">Cause and correction</td><td colspan="2">Assumed cause</td><td colspan="2">Correction</td><td colspan="2">Prevention</td></tr>
<tr><td colspan="2">The restore operation for a Safety Unit was performed on an incorrect Unit.</td><td colspan="2">After making sure that the destination is set to an intended unit, cycle the power supply and then perform the restore operation for a Safety Unit again.</td><td colspan="2">After making sure that the destination is set to an intended unit, start the restore operation for a Safety Unit.</td></tr>
<tr><td colspan="2">An incorrect Safety Unit Restore File was used.</td><td colspan="2">On the front indicators of the Safety CPU Unit, check the safety signature for the Safety Unit Restore File that is stored in the SD Memory Card.<br>After making sure that the safety signature is the intended signature, cycle the power supply and then perform the restore operation for a Safety Unit again.</td><td colspan="2">On the front indicators of the Safety CPU Unit, check the safety signature for the Safety Unit Restore File that is stored in the SD Memory Card. After making sure that the safety signature is the intended signature, start the restore operation for a Safety Unit.</td></tr>
<tr><td>Attached information</td><td colspan="6">None</td></tr>
<tr><td>Precautions/ Remarks</td><td colspan="6">None</td></tr>
</table>

| | | | | | | |
|---|---|---|---|---|---|---|
| **Event name** | Safety Unit Restore Operation Failed to Start (Version Mismatch) | | | **Event code** | 35280000 hex | |
| **Meaning** | A version mismatch was detected during pre-execution checks for a restore operation for a Safety Unit. | | | | | |
| **Source** | Depends on where the Sysmac Studio is connected and the system configuration. | | **Source details** | NX Unit | **Detection timing** | When Safety Unit restoring data is specified by the user |
| **Error attributes** | **Level** | Observation | **Recovery** | --- | **Log category** | System |
| **Effects** | **User program** | Continues. | **Operation** | --- | | |
| **System-defined variables** | **Variable** | | **Data type** | | **Name** | |
| | None | | None | | None | |
| **Cause and correction** | **Assumed cause** | | **Correction** | | **Prevention** | |
| | The restore operation for a Safety Unit was performed on an incompatible unit. | | After making sure that the destination is set to a compatible unit, cycle the power supply and then perform the restore operation for a Safety Unit again. | | After making sure that the destination is set to a compatible unit, start the restore operation for a Safety Unit. | |
| | An incorrect Safety Unit Restore File was used. | | On the front indicators of the Safety CPU Unit, check the safety signature for the Safety Unit Restore File that is stored in the SD Memory Card.<br>After making sure that the safety signature is the intended signature, cycle the power supply and then perform the restore operation for a Safety Unit again. | | On the front indicators of the Safety CPU Unit, check the safety signature for the Safety Unit Restore File that is stored in the SD Memory Card. After making sure that the safety signature is the intended signature, start the restore operation for a Safety Unit. | |
| **Attached information** | None | | | | | |
| **Precautions/ Remarks** | None | | | | | |

<table>
<tr><td>Event name</td><td colspan="3">Safety Unit Restore Operation Failed to Start (Node Name Mismatch)</td><td>Event code</td><td colspan="2">35290000 hex</td></tr>
<tr><td>Meaning</td><td colspan="6">A node name mismatch was detected during pre-execution checks for a restore operation for a Safety Unit.</td></tr>
<tr><td>Source</td><td colspan="2">Depends on where the Sysmac Studio is connected and the system configuration.</td><td>Source details</td><td>NX Unit</td><td>Detection timing</td><td>When Safety Unit restoring data is specified by the user</td></tr>
<tr><td>Error attributes</td><td>Level</td><td>Observation</td><td>Recovery</td><td>---</td><td>Log category</td><td>System</td></tr>
<tr><td>Effects</td><td>User program</td><td>Continues.</td><td>Operation</td><td colspan="3">---</td></tr>
<tr><td rowspan="2">System-defined variables</td><td colspan="2">Variable</td><td colspan="2">Data type</td><td colspan="2">Name</td></tr>
<tr><td colspan="2">None</td><td colspan="2">None</td><td colspan="2">None</td></tr>
<tr><td rowspan="4">Cause and correction</td><td colspan="2">Assumed cause</td><td colspan="2">Correction</td><td colspan="2">Prevention</td></tr>
<tr><td colspan="2">The restore operation for a Safety Unit was performed on an incorrect Unit.</td><td colspan="2">After making sure that the destination is set to an intended unit, cycle the power supply and then perform the restore operation for a Safety Unit again.</td><td colspan="2">After making sure that the destination is set to an intended unit, start the restore operation for a Safety Unit.</td></tr>
<tr><td colspan="2">An incorrect Safety Unit Restore File was used.</td><td colspan="2">On the front indicators of the Safety CPU Unit, check the safety signature for the Safety Unit Restore File that is stored in the SD Memory Card.<br>After making sure that the safety signature is the intended signature, cycle the power supply and then perform the restore operation for a Safety Unit again.</td><td colspan="2">On the front indicators of the Safety CPU Unit, check the safety signature for the Safety Unit Restore File that is stored in the SD Memory Card. After making sure that the safety signature is the intended signature, start the restore operation for a Safety Unit.</td></tr>
<tr><td colspan="2">The node name specified when the Safety Unit Restore File was generated is incorrect.</td><td colspan="2">Specify a correct node name for the Safety Unit Restore File.</td><td colspan="2">When you enter a node name, visually check that the specified node name is correct.</td></tr>
<tr><td>Attached information</td><td colspan="6">None</td></tr>
<tr><td>Precautions/ Remarks</td><td colspan="6">None</td></tr>
</table>

| Event name | Safety Unit Restore Operation Failed to Start (Safety Password Mismatch) | | | Event code | 352A0000 hex | |
|---|---|---|---|---|---|---|
| Meaning | A safety password mismatch was detected during pre-execution checks for a restore operation for a Safety Unit. | | | | | |
| Source | Depends on where the Sysmac Studio is connected and the system configuration. | | Source details | NX Unit | Detection timing | When Safety Unit restoring data is specified by the user |
| Error attributes | Level | Observation | Recovery | --- | Log category | System |
| Effects | User program | Continues. | Operation | --- | | |
| System-defined variables | Variable | | Data type | | Name | |
| | None | | None | | None | |
| Cause and correction | Assumed cause | | Correction | | Prevention | |
| | The restore operation for a Safety Unit was performed on an incorrect Unit. | | After making sure that the destination is set to an intended unit, cycle the power supply and then perform the restore operation for a Safety Unit again. | | After making sure that the destination is set to an intended unit, start the restore operation for a Safety Unit. | |
| | An incorrect Safety Unit Restore File was used. | | On the front indicators of the Safety CPU Unit, check the safety signature for the Safety Unit Restore File that is stored in the SD Memory Card.<br>After making sure that the safety signature is the intended signature, cycle the power supply and then perform the restore operation for a Safety Unit again. | | On the front indicators of the Safety CPU Unit, check the safety signature for the Safety Unit Restore File that is stored in the SD Memory Card. After making sure that the safety signature is the intended signature, start the restore operation for a Safety Unit. | |
| | The safety password specified when the Safety Unit Restore File was generated is incorrect. | | Specify a correct safety password for the Safety Unit Restore File. | | Specify a correct password when you enter the safety password. | |
| Attached information | None | | | | | |
| Precautions/ Remarks | None | | | | | |

| Event name | Safety Unit Restore Operation Failed | | Event code | 352B0000 hex | |
|---|---|---|---|---|---|
| Meaning | The restore operation for a Safety Unit ended in an error. | | | | |
| Source | Depends on where the Sysmac Studio is connected and the system configuration. | Source details | NX Unit | Detection timing | When Safety Unit restoring data is specified by the user |
| Error attributes | Level | Observation | Recovery | --- | Log category | System |
| Effects | User program | Continues. | Operation | --- | | |
| System-defined variables | Variable | | Data type | | Name | |
| | None | | None | | None | |
| Cause and correction | Assumed cause | | Correction | | Prevention | |
| | The SD Memory Card was removed during a restore operation for a Safety Unit. | | Insert the SD Memory Card that contains the Safety Unit Restore File, cycle the power supply and then perform the restore operation for a Safety Unit again. | | Do not remove the SD Memory Card during the restore operation for a Safety Unit. | |
| | Data was read from or written to the SD Memory Card via the Support Software or an FTP client during a restore operation for a Safety Unit. | | Store the correct Safety Unit Restore File to the SD Memory Card, cycle the power supply and then perform the restore operation for a Safety Unit again. | | Do not read data from or write data to the SD Memory Card via the Support Software or an FTP client during the restore operation for a Safety Unit. | |
| Attached information | None | | | | | |
| Precautions/ Remarks | None | | | | | |

## ● Other Errors

| | | | | | | |
|---|---|---|---|---|---|---|
| **Event name** | NX Message Communications Error | | | **Event code** | 80220000 hex | |
| **Meaning** | An error was detected in message communications for an NX Unit and the message frame was discarded. | | | | | |
| **Source** | Depends on where the Sysmac Studio is connected and the system configuration. | | **Source details** | NX Unit | **Detection timing** | During NX message communications |
| **Error attributes** | **Level** | Observation | **Recovery** | --- | **Log category** | System |
| **Effects** | **User program** | Continues. | **Operation** | Not affected. | | |
| **System-defined variables** | **Variable** | | **Data type** | | **Name** | |
| | None | | None | | None | |
| **Cause and correction** | **Assumed cause** | | **Correction** | | **Prevention** | |
| | The message communications load is high. | | Reduce the number of times that instructions are used to send NX messages.<br>Refer to the appendix of the *NJ/NX-series Instructions Reference Manual (Cat. No. W502)* for information on the instructions that send messages. | | Reduce the number of times that instructions are used to send NX messages. | |
| | The communications cable is disconnected or broken. This cause does not apply if attached information 2 is 0 (NX bus). | | Connect the communications cable securely | | Connect the communications cable securely. | |
| | Message communications were cut off as the result of executing a synchronization or restoration operation on the Sysmac Studio or as the result of disconnecting an EtherCAT slave. | | --- | | --- | |
| **Attached information** | Attached information 1: | System information | | | | |
| | Attached information 2: | Type of communications where error occurred<br>0: NX bus<br>1: EtherCAT<br>2: Serial communications (USB)<br>65535: Internal Unit communications (routing) | | | | |
| **Precautions/ Remarks** | None | | | | | |

| Event name | Sysmac Studio Communications Connection Timeout | | Event code | 951E0000 hex | | |
|---|---|---|---|---|---|---|
| Meaning | A communications timeout occurred between the Sysmac Studio and the Safety CPU Unit. | | | | | |
| Source | Depends on where the Sysmac Studio is connected and the system configuration. | | Source details | NX Unit | Detection timing | When applicable |
| Error attributes | Level | Information | Recovery | --- | Log category | System |
| Effects | User program | Continues. | Operation | If the CPU Unit was in DEBUG mode, it automatically enters PROGRAM mode. | | |
| System-defined variables | Variable | | Data type | | Name | |
| | None | | None | | None | |
| Cause and correction | Assumed cause | | Correction | | Prevention | |
| | The communications cable was disconnected. | | Do not do anything to disconnect communications with the Sysmac Studio while the Safety CPU Unit is operating in DEBUG mode. | | Perform debugging considering the corrections that are given on the left. | |
| Attached information | None | | | | | |
| Precautions/ Remarks | None | | | | | |

| Event name | Clear All Memory Rejected | | Event code | 951F0000 hex | | |
|---|---|---|---|---|---|---|
| Meaning | Clearing all of memory failed. | | | | | |
| Source | Depends on where the Sysmac Studio is connected and the system configuration. | | Source details | NX Unit | Detection timing | When commanded from user |
| Error attributes | Level | Information | Recovery | --- | Log category | System |
| Effects | User program | Continues. | Operation | --- | | |
| System-defined variables | Variable | | Data type | | Name | |
| | None | | None | | None | |
| Cause and correction | Assumed cause | | Correction | | Prevention | |
| | The Clear All Memory operation for a Controller or a Slave Terminal was performed. | | Specify the Units individually and perform the Clear All Memory operation. | | Specify the Units individually and perform the Clear All Memory operation. | |
| Attached information | Attached information 1: The Clear All Memory operation for the Controller or the Slave Terminal was performed. | | | | | |
| Precautions/ Remarks | The Clear All Memory operation for the Safety CPU Unit cannot be performed for the Controller or the Slave Terminal. | | | | | |

### ● User Access Log

| | | | | | | |
|---|---|---|---|---|---|---|
| **Event name** | Event Log Cleared | | | **Event code** | 90400000 hex | |
| **Meaning** | The event log was cleared. | | | | | |
| **Source** | Depends on where the Sysmac Studio is connected and the system configuration. | | **Source details** | NX Unit | **Detection timing** | When commanded from user |
| **Error attributes** | **Level** | Information | **Recovery** | --- | **Log category** | Access |
| **Effects** | **User program** | Continues. | **Operation** | Not affected. | | |
| **System-defined variables** | **Variable** | | **Data type** | | **Name** | |
| | None | | None | | None | |
| **Cause and correction** | **Assumed cause** | | **Correction** | | **Prevention** | |
| | The event log was cleared by the user. | | --- | | --- | |
| **Attached information** | Attached information 1: | Events that were cleared.<br>1: The system event log was cleared.<br>2: The access event log was cleared. | | | | |
| **Precautions/ Remarks** | None | | | | | |

| | | | | | | |
|---|---|---|---|---|---|---|
| **Event name** | Memory All Cleared | | | **Event code** | 90430000 hex | |
| **Meaning** | The Unit settings were cleared. | | | | | |
| **Source** | Depends on where the Sysmac Studio is connected and the system configuration. | | **Source details** | NX Unit | **Detection timing** | When commanded from user |
| **Error attributes** | **Level** | Information | **Recovery** | --- | **Log category** | Access |
| **Effects** | **User program** | Continues. | **Operation** | The Unit settings are cleared. | | |
| **System-defined variables** | **Variable** | | **Data type** | | **Name** | |
| | None | | None | | None | |
| **Cause and correction** | **Assumed cause** | | **Correction** | | **Prevention** | |
| | The Clear All Memory operation was performed. | | --- | | --- | |
| **Attached information** | Attached information 1: | Unit number of the NX Unit where the Clear All Memory operation was performed | | | | |
| | Attached information 2: | Execution results<br>0: Successful<br>1: Hardware error<br>2: Initialization failed<br>3: Initialization not possible | | | | |
| **Precautions/ Remarks** | Refer to the attached information for the results of the Clear All Memory operation. | | | | | |

## Safety I/O Units

Details on the errors (events) that can occur in the Safety I/O Units are given in the following tables.

## ● System Error

| | | | | | | |
|---|---|---|---|---|---|---|
| **Event name** | System Error | | | **Event code** | 05200000 hex | |
| **Meaning** | A hardware error was detected during self-diagnosis of the hardware. | | | | | |
| **Source** | Depends on where the Sysmac Studio is connected and the system configuration. | | **Source details** | NX Unit | **Detection timing** | Continuously |
| **Error attributes** | **Level** | Minor fault | **Recovery** | Cycle the power supply to the Unit. | **Log category** | System |
| **Effects** | **User program** | Continues. | **Operation** | The Unit stops operating and the I/O data changes to the safe states. | | |
| **System-defined variables** | **Variable** | | **Data type** | | **Name** | |
| | None | | None | | None | |
| **Cause and correction** | **Assumed cause** | | **Correction** | | **Prevention** | |
| | Hardware has failed. | | Cycle the power supply. If the error occurs again, replace the Unit. | | If cycling the power supply restores normal operation, there may be excessive noise near the Unit. Implement noise countermeasures. | |
| | A memory error occurred due to a transient cause, such as a software error or excessive noise. | | | | | |
| **Attached information** | Attached information 1: System information, status code<br>Attached information 2: System information, status code<br>Attached information 3: System information, status code<br>Attached information 4: System information, status code | | | | | |
| **Precautions/ Remarks** | None | | | | | |

## ● Communications Errors

| | | | | | | |
|---|---|---|---|---|---|---|
| **Event name** | Safety Process Data Communications Not Established - Incorrect Unit Parameter Error | | | **Event code** | 35210000 hex | |
| **Meaning** | Safety process data communications was not established with the Safety CPU Unit. | | | | | |
| **Source** | Depends on where the Sysmac Studio is connected and the system configuration. | | **Source details** | NX Unit | **Detection timing** | When safety process data communications are established |
| **Error attributes** | **Level** | Minor fault | **Recovery** | For request to establish communications from Safety CPU Unit after removing cause of error | **Log category** | System |
| **Effects** | **User program** | Continues. | **Operation** | The Unit stops operating and the I/O data changes to the safe states. | | |
| **System-defined variables** | **Variable** | | **Data type** | | **Name** | |
| | None | | None | | None | |
| **Cause and correction** | **Assumed cause** | | **Correction** | | **Prevention** | |
| | The model or safety I/O terminal settings are not correct. | | Check the safety I/O terminal settings, correct any errors, and then transfer the settings to the Safety CPU Unit. | | Set the parameters considering the corrections that are given on the left. | |
| | | | Check the model of the Safety I/O Unit to see if it is correct. | | | |
| **Attached information** | None | | | | | |
| **Precautions/ Remarks** | None | | | | | |

<table>
<tr><td>Event name</td><td colspan="3">Safety Process Data Communications Not Established, Incorrect FSoE Slave Address Error</td><td>Event code</td><td colspan="2">35230000 hex</td></tr>
<tr><td>Meaning</td><td colspan="6">Safety process data communications was not established with the Safety CPU Unit because of an incorrect FSoE slave address.</td></tr>
<tr><td>Source</td><td colspan="2">Depends on where the Sysmac Studio is connected and the system configuration.</td><td>Source details</td><td>NX Unit</td><td>Detection timing</td><td>When safety process data communications are established</td></tr>
<tr><td>Error attributes</td><td>Level</td><td>Minor fault</td><td>Recovery</td><td>For request to establish communications from Safety CPU Unit after removing cause of error</td><td>Log category</td><td>System</td></tr>
<tr><td>Effects</td><td>User program</td><td>Continues.</td><td>Operation</td><td colspan="3">The Unit stops operating and the I/O data changes to the safe states.</td></tr>
<tr><td rowspan="2">System-defined variables</td><td colspan="2">Variable</td><td colspan="2">Data type</td><td colspan="2">Name</td></tr>
<tr><td colspan="2">None</td><td colspan="2">None</td><td colspan="2">None</td></tr>
<tr><td rowspan="2">Cause and correction</td><td colspan="2">Assumed cause</td><td colspan="2">Correction</td><td colspan="2">Prevention</td></tr>
<tr><td colspan="2">The setting of the FSoE slave address in the safety process data communications settings is different from the setting in the Unit.</td><td colspan="2">Perform the Clear All Memory operation for the Unit.</td><td colspan="2">If you use a Safety I/O Unit for which safety process data communications were previously established in another system, perform the Clear All Memory operation before you use the Unit.</td></tr>
<tr><td>Attached information</td><td colspan="6">None</td></tr>
<tr><td>Precautions/ Remarks</td><td colspan="6">None</td></tr>
</table>

| | | | | | | |
|---|---|---|---|---|---|---|
| **Event name** | Safety Process Data Communications Not Established, Incorrect Frame Error | | | **Event code** | 35240000 hex | |
| **Meaning** | Safety process data communications was not established with the Safety CPU Unit because an incorrect frame was received. | | | | | |
| **Source** | Depends on where the Sysmac Studio is connected and the system configuration. | | **Source details** | NX Unit | **Detection timing** | When safety process data communications are established |
| **Error attributes** | **Level** | Minor fault | **Recovery** | For request to establish communications from Safety CPU Unit after removing cause of error | **Log category** | System |
| **Effects** | **User program** | Continues. | **Operation** | The Unit stops operating and the I/O data changes to the safe states. | | |
| **System-defined variables** | **Variable** | | **Data type** | | **Name** | |
| | None | | None | | None | |
| **Cause and correction** | **Assumed cause** | | **Correction** | | **Prevention** | |
| | An incorrect frame was received in safety process data communications. | | Make sure that the system configurations and model numbers agree for the Safety CPU Unit and Safety I/O Units. | | Set the system configuration and setup according to the corrections that are given on the left. | |
| | There is excessive noise. | | Implement noise countermeasures. | | Implement noise countermeasures if excessive noise caused the error. | |
| **Attached information** | None | | | | | |
| **Precautions/ Remarks** | None | | | | | |

<table>
<tr><td>Event name</td><td colspan="3">NX Unit I/O Communications Error</td><td>Event code</td><td colspan="2">80200000 hex</td></tr>
<tr><td>Meaning</td><td colspan="6">An I/O communications error occurred between the NX bus master and the NX Unit.</td></tr>
<tr><td>Source</td><td colspan="2">Depends on where the Support Software is connected and the system configuration.</td><td>Source details</td><td>NX Unit</td><td>Detection timing</td><td>Continuously</td></tr>
<tr><td rowspan="2">Error attributes</td><td>Level</td><td colspan="2">Minor fault</td><td>Log category</td><td colspan="2">System</td></tr>
<tr><td>Recovery</td><td colspan="5">[NX Bus of the CPU Unit]<br>When Fail-soft Operation Is Set to Stop<br>Errors are reset in the NX Bus Function Module.<br>When Fail-soft Operation Is Set to Fail-soft<br>Reset the error in the NX Unit.<br>[Communications Coupler Unit]<br>When Fail-soft Operation Is Set to Stop<br>If errors are detected in the controller, errors are reset in the controller.<br>If errors are not detected in the controller, errors are reset in the Communications Coupler Unit and the NX Unit.<br>When Fail-soft Operation Is Set to Fail-soft<br>Errors are reset in the Communications Coupler Unit and the NX Unit.</td></tr>
<tr><td>Effects</td><td>User program</td><td colspan="2">Continues.</td><td>Operation</td><td colspan="2">The NX Unit will continue to operate.<br>Input data: Updating input values stops.<br>Output data: The output values depend on the Load Rejection Output Setting.</td></tr>
<tr><td>System-defined variables</td><td colspan="2">Variable</td><td colspan="2">Data type</td><td colspan="2">Name</td></tr>
<tr><td></td><td colspan="2">None</td><td colspan="2">None</td><td colspan="2">None</td></tr>
<tr><td rowspan="8">Cause and correction</td><td colspan="2">Assumed cause</td><td colspan="2">Correction</td><td colspan="2">Prevention</td></tr>
<tr><td colspan="6">[NX Bus of the CPU Unit]</td></tr>
<tr><td colspan="2">An error occurred in the CPU Unit, which prevents the NX bus communications from being carried out normally.</td><td colspan="2">Check the error in the CPU Unit and try the solutions specified in Correction.</td><td colspan="2">Try the solutions specified in Prevention for the error in the CPU Unit.</td></tr>
<tr><td colspan="2">The NX Unit is not mounted properly.</td><td colspan="2">Mount the NX Units and End Cover securely and secure them with End Plates.</td><td colspan="2">Mount the NX Units and End Cover securely and secure them with End Plates.</td></tr>
<tr><td colspan="2">The power cable for the Unit power supply is disconnected. Or, the wiring from the Unit power supply to the NX Units is incorrect.</td><td colspan="2">Correctly wire the Unit power supply to the NX Units.</td><td colspan="2">Correctly wire the Unit power supply to the NX Units.</td></tr>
<tr><td colspan="2">The power cable for the Unit power supply is broken.</td><td colspan="2">If the power cable connecting the Unit power supply and the NX Units is broken, replace the power cable.</td><td colspan="2">None</td></tr>
<tr><td colspan="2">The Unit power voltage is out of the supported range, or the Unit power capacity is not sufficient.</td><td colspan="2">Configure the power supply system correctly in accordance with the power supply specifications.</td><td colspan="2">Configure the power supply system correctly in accordance with the power supply specifications.</td></tr>
<tr><td colspan="2">There is a hardware error in the NX Unit.</td><td colspan="2">If the error occurs again even after you make the above correction, replace the NX Unit.</td><td colspan="2">None</td></tr>
</table>

| | | | |
|---|---|---|---|
| **Cause and correction** | [Communications Coupler Unit] | | |
| | An error occurred in the Communications Coupler Unit, which prevents the NX bus communications from being carried out normally. | Check the error in the Communications Coupler Unit and try the solutions specified in Correction. | Try the solutions specified in Prevention for the error in the Communications Coupler Unit. |
| | The NX Unit is not mounted properly. | Mount the NX Units and End Cover securely and secure them with End Plates. | Mount the NX Units and End Cover securely and secure them with End Plates. |
| | The power cable for the Unit power supply is disconnected. Or, the wiring from the Unit power supply to the NX Units is incorrect. | Correctly wire the Unit power supply to the NX Units. | Correctly wire the Unit power supply to the NX Units. |
| | The power cable for the Unit power supply is broken. | If the power cable connecting the Unit power supply and the NX Units is broken, replace the power cable. | None |
| | The Unit power voltage is out of the supported range, or the Unit power capacity is not sufficient. | Configure the power supply system correctly in accordance with the power supply specifications. | Configure the power supply system correctly in accordance with the power supply specifications. |
| | There is a hardware error in the NX Unit. | If the error occurs again even after you make the above correction, replace the NX Unit. | None |
| **Attached information** | None | | |
| **Precautions/ Remarks** | None | | |

<table>
<tr><td>Event name</td><td colspan="3">Safety Process Data Communications Timeout</td><td>Event code</td><td colspan="2">80300000 hex</td></tr>
<tr><td>Meaning</td><td colspan="6">A communications timeout occurred in safety process data communications with the Safety Control Unit.</td></tr>
<tr><td>Source</td><td colspan="2">Depends on where the Sysmac Studio is connected and the system configuration.</td><td>Source details</td><td>NX Unit</td><td>Detection timing</td><td>When establishing or during safety process data communications</td></tr>
<tr><td>Error attributes</td><td>Level</td><td>Minor fault</td><td>Recovery</td><td>For request to establish communications from Safety CPU Unit after removing cause of error</td><td>Log category</td><td>System</td></tr>
<tr><td>Effects</td><td>User program</td><td>Continues.</td><td>Operation</td><td colspan="3">The Unit continues to operate. The safe states are used for the I/O data of the safety connection where the error was detected.</td></tr>
<tr><td rowspan="2">System-defined variables</td><td colspan="2">Variable</td><td colspan="2">Data type</td><td colspan="2">Name</td></tr>
<tr><td colspan="2">None</td><td colspan="2">None</td><td colspan="2">None</td></tr>
<tr><td rowspan="5">Cause and correction</td><td colspan="2">Assumed cause</td><td colspan="2">Correction</td><td colspan="2">Prevention</td></tr>
<tr><td colspan="2">A setting is not correct. The setting of the safety task period is too short.</td><td colspan="2">Increase the safety task period and then transfer the settings to the Safety CPU Unit.</td><td colspan="2">Set the system configuration and setup according to the corrections that are given on the left.</td></tr>
<tr><td colspan="2">There is excessive noise.</td><td colspan="2">Implement noise countermeasures.</td><td colspan="2">Implement noise countermeasures if excessive noise caused the error.</td></tr>
<tr><td colspan="2">The Safety CPU Unit or safety slave entered a status where it could not continue safety process data communications.</td><td colspan="2">Check the status of the Safety CPU Unit or safety slave.</td><td colspan="2">Refer to troubleshooting information for the Safety CPU Unit or safety slave.</td></tr>
<tr><td colspan="2">An error or status change occurred in the NX Bus Master to which the Unit is connected, preventing correct process data communications.</td><td colspan="2">Check the status of the NX Bus Master to which the Unit is connected.</td><td colspan="2">Set the system configuration and setup according to the corrections that are given on the left.</td></tr>
<tr><td>Attached information</td><td colspan="6">None</td></tr>
<tr><td>Precautions/ Remarks</td><td colspan="6">None</td></tr>
</table>

| Event name | NX Bus I/O Communications Stopped | | | Event code | 84F10000 hex | |
|---|---|---|---|---|---|---|
| Meaning | An error occurred in I/O communications between the NX Bus Master and an NX Unit. | | | | | |
| Source | Depends on where the Sysmac Studio is connected and the system configuration. | | Source details | NX Unit | Detection timing | Continuously |
| Error attributes | Level | Minor fault | Recovery | Cycle the power supply to the NX bus master and NX Units. | Log category | System |
| Effects | User program | Continues. | Operation | The NX Units will continue to operate.<br>• Input data:<br>An error occurs in safety process data communications because refreshing is stopped. The values of the status in standard process data are not refreshed.<br>• Output data:<br>An error occurs in safety process data communications because 0's are output. | | |
| System-defined variables | Variable | | Data type | | Name | |
| | None | | None | | None | |
| Cause and correction | Assumed cause | | Correction | | Prevention | |
| | There is a hardware error in the NX bus master or an NX Unit. | | If the error occurs again even after you cycle the power supply to the NX Units, replace the NX Bus Master or the NX Unit. | | None | |
| Attached information | None | | | | | |
| Precautions/ Remarks | None | | | | | |

## ● Safety I/O Errors

| | | | | | | |
|---|---|---|---|---|---|---|
| **Event name** | Internal Circuit Error at Safety Input | | **Event code** | 05210000 hex | | |
| **Meaning** | A fault was detected in the internal circuit for the safety input terminal. | | | | | |
| **Source** | Depends on where the Sysmac Studio is connected and the system configuration. | | **Source details** | NX Unit | **Detection timing** | During refreshing |
| **Error attributes** | **Level** | Minor fault | **Recovery** | Cycle the power supply to the Unit. | **Log category** | System |
| **Effects** | **User program** | Continues. | **Operation** | The safety input terminal retains the safe state until the power supply is cycled. | | |
| **System-defined variables** | **Variable** | | **Data type** | | **Name** | |
| | None | | None | | None | |
| **Cause and correction** | **Assumed cause** | | **Correction** | | **Prevention** | |
| | The internal circuit for the safety input terminal is faulty. | | Cycle the power supply. If the error occurs again, replace the Unit. | | If cycling the power supply restores normal operation, there may be excessive noise near the Unit. Implement noise countermeasures. | |
| | A memory error or signal error occurred due to a transient cause, such as an excessive noise. | | | | | |
| **Attached information** | Attached information 1: Terminal number | | | | | |
| **Precautions/ Remarks** | None | | | | | |

| Event name | Internal Circuit Error at Test Output | | | Event code | 05220000 hex | |
|---|---|---|---|---|---|---|
| Meaning | A fault was detected in the internal circuit for the test output terminal. | | | | | |
| Source | Depends on where the Sysmac Studio is connected and the system configuration. | | Source details | NX Unit | Detection timing | During refreshing |
| Error attributes | Level | Minor fault | Recovery | Cycle the power supply to the Unit. | Log category | System |
| Effects | User program | Continues. | Operation | The test output terminal retains the safe state until the power supply is cycled. Also, an External Test Signal Failure at Safety Input event (65220000 hex) will occur for the safety input terminal that is the test source of the test output terminal. | | |
| System-defined Variables | Variable | | Data type | | Name | |
| | None | | None | | None | |
| Cause and correction | Assumed cause | | Correction | | Prevention | |
| | The internal circuit for the test output terminal is faulty. | | Cycle the power supply. If the error occurs again, replace the Unit. | | If cycling the power supply restores normal operation, there may be excessive noise near the Unit. Implement noise countermeasures. | |
| | A memory error or signal error occurred due to a transient cause, such as an excessive noise. | | | | | |
| Attached information | Attached information 1: Terminal number | | | | | |
| Precautions/ Remarks | None | | | | | |

<table>
<tr><td>Event name</td><td colspan="3">Internal Circuit Error at Safety Output</td><td>Event code</td><td colspan="2">05230000 hex</td></tr>
<tr><td>Meaning</td><td colspan="6">A fault was detected in the internal circuit for the safety output terminal.</td></tr>
<tr><td>Source</td><td colspan="2">Depends on where the Sysmac Studio is connected and the system configuration.</td><td>Source details</td><td>NX Unit</td><td>Detection timing</td><td>During refreshing</td></tr>
<tr><td>Error attributes</td><td>Level</td><td>Minor fault</td><td>Recovery</td><td>Cycle the power supply to the Unit.</td><td>Log category</td><td>System</td></tr>
<tr><td>Effects</td><td>User program</td><td>Continues.</td><td>Operation</td><td colspan="3">The safety output terminal retains the safe state until the power supply is cycled.</td></tr>
<tr><td rowspan="2">System-defined variables</td><td colspan="2">Variable</td><td colspan="2">Data type</td><td colspan="2">Name</td></tr>
<tr><td colspan="2">None</td><td colspan="2">None</td><td colspan="2">None</td></tr>
<tr><td rowspan="3">Cause and correction</td><td colspan="2">Assumed cause</td><td colspan="2">Correction</td><td colspan="2">Prevention</td></tr>
<tr><td colspan="2">The internal circuit for the safety output terminal is faulty.</td><td colspan="2" rowspan="2">Cycle the power supply. If the error occurs again, replace the Unit.</td><td colspan="2" rowspan="2">If cycling the power supply restores normal operation, there may be excessive noise near the Unit. Implement noise countermeasures.</td></tr>
<tr><td colspan="2">A memory error or signal error occurred due to a transient cause, such as an excessive noise.</td></tr>
<tr><td>Attached information</td><td colspan="6">Attached information 1: Terminal number</td></tr>
<tr><td>Precautions/ Remarks</td><td colspan="6">None</td></tr>
</table>

<table>
<tr><td>Event name</td><td colspan="2">I/O Power Supply Voltage Error</td><td>Event code</td><td colspan="2">65200000 hex</td></tr>
<tr><td>Meaning</td><td colspan="5">An incorrect I/O power supply voltage was detected.</td></tr>
<tr><td>Source</td><td colspan="2">Depends on where the Sysmac Studio is connected and the system configuration.</td><td>Source details</td><td>NX Unit</td><td>Detection timing</td><td>During refreshing</td></tr>
<tr><td>Error attributes</td><td>Level</td><td>Minor fault</td><td>Recovery</td><td>When cause of error is removed</td><td>Log category</td><td>System</td></tr>
<tr><td>Effects</td><td>User program</td><td>Continues.</td><td>Operation</td><td colspan="3">The Unit continues to operate, but the I/O data retains the safe states.</td></tr>
<tr><td rowspan="2">System-defined variables</td><td colspan="2">Variable</td><td colspan="2">Data type</td><td colspan="2">Name</td></tr>
<tr><td colspan="2">None</td><td colspan="2">None</td><td colspan="2">None</td></tr>
<tr><td rowspan="2">Cause and correction</td><td colspan="2">Assumed cause</td><td colspan="2">Correction</td><td colspan="2">Prevention</td></tr>
<tr><td colspan="2">The input power or output power is not supplied correctly.</td><td colspan="2">Check the following and supply the rated power.<br>• Is the power supply voltage within the specifications?<br>• Is the wiring correct and not disconnected?<br>• Is 24 V applied to the safety output terminal and is the safety output terminal not touching the positive power supply wire?<br>If the voltage that is measured is correct, the Unit may be faulty. In that case, replace the CPU Unit.</td><td colspan="2">Design the system considering the corrections that are given on the left.</td></tr>
<tr><td>Attached information</td><td colspan="5">None</td></tr>
<tr><td>Precautions/ Remarks</td><td colspan="5">None</td></tr>
</table>

<table>
<tr><td>Event name</td><td colspan="3">Output Power Interrupt Circuit Error</td><td>Event code</td><td colspan="2">65210000 hex</td></tr>
<tr><td>Meaning</td><td colspan="6">An error was detected by the output power interruption test.</td></tr>
<tr><td>Source</td><td colspan="2">Depends on where the Sysmac Studio is connected and the system configuration.</td><td>Source details</td><td>NX Unit</td><td>Detection timing</td><td>During refreshing</td></tr>
<tr><td rowspan="2">Error attributes</td><td>Level</td><td>Minor fault</td><td>Recovery</td><td>When cause of error is removed and then one of the following is performed<br>• The I/O power supply is turned OFF.<br>• Safety process data communications are stopped.</td><td>Log category</td><td>System</td></tr>
<tr><td>Effects</td><td>User program</td><td>Continues.</td><td>Operation</td><td colspan="3">The Unit continues to operate, but the I/O data retains the safe states.</td></tr>
<tr><td rowspan="2">System-defined variables</td><td colspan="2">Variable</td><td colspan="2">Data type</td><td colspan="2">Name</td></tr>
<tr><td colspan="2">None</td><td colspan="2">None</td><td colspan="2">None</td></tr>
<tr><td rowspan="2">Cause and correction</td><td colspan="2">Assumed cause</td><td colspan="2">Correction</td><td colspan="2">Prevention</td></tr>
<tr><td colspan="2">The wiring is not correct or there is a fault in the hardware.</td><td colspan="2">Check the following and supply the rated power.<br>• Is the power supply voltage within the specifications?<br>• Is the wiring correct and not disconnected?<br>• Is 24 V applied to the safety output terminal and is the safety output terminal not touching the positive power supply wire?<br>If the voltage that is measured is correct, the Unit may be faulty. In that case, replace the CPU Unit.</td><td colspan="2">Design the system considering the corrections that are given on the left.</td></tr>
<tr><td>Attached information</td><td colspan="6">None</td></tr>
<tr><td>Precautions/ Remarks</td><td colspan="6">None</td></tr>
</table>

| | | | | | | |
|---|---|---|---|---|---|---|
| **Event name** | External Test Signal Failure at Safety Input | | | **Event code** | 65220000 hex | |
| **Meaning** | An error was detected in test pulse evaluation of the safety input terminals. | | | | | |
| **Source** | Depends on where the Sysmac Studio is connected and the system configuration. | | **Source details** | NX Unit | **Detection timing** | During refreshing |
| **Error attributes** | **Level** | Minor fault | **Recovery** | When safety input terminal goes inactive after cause of error is removed | **Log category** | System |
| **Effects** | **User program** | Continues. | **Operation** | The safety input terminal retains the safe state until the error is cleared. | | |
| **System-defined variables** | **Variable** | | **Data type** | | **Name** | |
| | None | | None | | None | |
| **Cause and correction** | **Assumed cause** | | **Correction** | | **Prevention** | |
| | The positive power supply wire is in contact with the input signal line. | | Check the external wiring. | | Set the parameters and wire the system considering the corrections that are given on the left. | |
| | The input signal lines are shorted. | | | | | |
| | The external device is faulty. | | Replace the external device. | | | |
| | An error occurred in the test output terminal used with the safety input terminal. | | Check the error in the test output terminal and try the solutions specified in Correction. | | Try the solutions specified in Prevention for the error in the test output terminal. | |
| **Attached information** | Attached information 1: Terminal number | | | | | |
| **Precautions/ Remarks** | None | | | | | |

<table>
<tr><td>Event name</td><td colspan="3">Discrepancy Error at Safety Input</td><td>Event code</td><td colspan="2">65230000 hex</td></tr>
<tr><td>Meaning</td><td colspan="6">An error was detected in discrepancy evaluation of safety input terminals.</td></tr>
<tr><td>Source</td><td colspan="2">Depends on where the Sysmac Studio is connected and the system configuration.</td><td>Source details</td><td>NX Unit</td><td>Detection timing</td><td>During refreshing</td></tr>
<tr><td>Error attributes</td><td>Level</td><td>Minor fault</td><td>Recovery</td><td>When safety input terminal goes inactive after cause of error is removed</td><td>Log category</td><td>System</td></tr>
<tr><td>Effects</td><td>User program</td><td>Continues.</td><td>Operation</td><td colspan="3">The safety input terminal retains the safe state until the error is cleared.</td></tr>
<tr><td rowspan="2">System-defined variables</td><td colspan="2">Variable</td><td colspan="2">Data type</td><td colspan="2">Name</td></tr>
<tr><td colspan="2">None</td><td colspan="2">None</td><td colspan="2">None</td></tr>
<tr><td rowspan="6">Cause and correction</td><td colspan="2">Assumed cause</td><td colspan="2">Correction</td><td colspan="2">Prevention</td></tr>
<tr><td colspan="2">There is a ground fault or disconnection in the input signal line.</td><td colspan="2">Check the external wiring.</td><td colspan="2" rowspan="4">Set the parameters and wire the system considering the corrections that are given on the left.</td></tr>
<tr><td colspan="2">The connected device is faulty.</td><td colspan="2">Replace the external device.</td></tr>
<tr><td colspan="2">The setting of the discrepancy time is not correct.</td><td colspan="2" rowspan="2">Correct the setting of the discrepancy time. If that does not correct the problem, use an input filter to set an ON delay or an OFF delay.</td></tr>
<tr><td colspan="2">Chattering occurred in the input signal from the external input device, such as a safety door.</td></tr>
<tr><td colspan="2">An error occurred in the test output terminal used with the safety input terminal.</td><td colspan="2">Check the error in the test output terminal and try the solutions specified in Correction.</td><td colspan="2">Try the solutions specified in Prevention for the error in the test output terminal.</td></tr>
<tr><td>Attached information</td><td colspan="6">Attached information 1: Terminal number</td></tr>
<tr><td>Precautions/ Remarks</td><td colspan="6">None</td></tr>
</table>

| | | | | | | |
|---|---|---|---|---|---|---|
| **Event name** | Overload Detected at Test Output | | | **Event code** | 65240000 hex | |
| **Meaning** | An overcurrent was detected at the test output terminal. | | | | | |
| **Source** | Depends on where the Sysmac Studio is connected and the system configuration. | | **Source details** | NX Unit | **Detection timing** | During refreshing |
| **Error attributes** | **Level** | Minor fault | **Recovery** | When safety input terminal goes inactive after cause of error is removed | **Log category** | System |
| **Effects** | **User program** | Continues. | **Operation** | The safety input terminal that is the test source of the test output terminal retains the safe state until the error is removed. | | |
| **System-defined variables** | **Variable** | | **Data type** | | **Name** | |
| | None | | None | | None | |
| **Cause and correction** | **Assumed cause** | | **Correction** | | **Prevention** | |
| | There is a ground fault on the output signal line. | | Check the external wiring. | | Set the parameters and wire the system considering the corrections that are given on the left. | |
| | The external device is faulty. | | Replace the external device. | | | |
| **Attached information** | Attached information 1: Terminal number | | | | | |
| **Precautions/ Remarks** | None | | | | | |

<table>
<tr><td>Event name</td><td colspan="3">Stuck-at-high Detected at Test Output</td><td>Event Code</td><td colspan="2">65250000 hex</td></tr>
<tr><td>Meaning</td><td colspan="6">It was detected that the test output terminal is stuck ON.</td></tr>
<tr><td>Source</td><td colspan="2">Depends on where the Sysmac Studio is connected and the system configuration.</td><td>Source details</td><td>NX Unit</td><td>Detection timing</td><td>During refreshing</td></tr>
<tr><td rowspan="2">Error attributes</td><td>Level</td><td>Minor fault</td><td>Recovery</td><td>When safety input terminal goes inactive after cause of error is removed</td><td>Log category</td><td>System</td></tr>
<tr><td>Effects</td><td>User program</td><td>Continues.</td><td>Operation</td><td colspan="3">The safety input terminal that is the test source of the test output terminal retains the safe state until the error is removed.</td></tr>
<tr><td rowspan="2">System-defined variables</td><td colspan="2">Variable</td><td colspan="2">Data type</td><td colspan="2">Name</td></tr>
<tr><td colspan="2">None</td><td colspan="2">None</td><td colspan="2">None</td></tr>
<tr><td rowspan="4">Cause and correction</td><td colspan="2">Assumed cause</td><td colspan="2">Correction</td><td colspan="2">Prevention</td></tr>
<tr><td colspan="2">The positive power supply line is in contact with the output signal line.</td><td colspan="2">Check the external wiring.</td><td colspan="2">Set the parameters and wire the system considering the corrections that are given on the left.</td></tr>
<tr><td colspan="2">The internal circuit is faulty.</td><td colspan="2" rowspan="2">Cycle the power supply. If the error occurs again, replace the Unit.</td><td colspan="2" rowspan="2">If cycling the power supply restores normal operation, there may be excessive noise near the Unit. Implement noise countermeasures.</td></tr>
<tr><td colspan="2">A memory error or signal error occurred due to a transient cause, such as a software error or excessive noise.</td></tr>
<tr><td>Attached information</td><td colspan="6">Attached information 1: Terminal number</td></tr>
<tr><td>Precautions/ Remarks</td><td colspan="6">None</td></tr>
</table>

| Event name | Short Circuit Detected at Safety Output | | Event code | 65270000 hex | |
|---|---|---|---|---|---|
| Meaning | A ground fault was detected on the safety output terminal. | | | | |
| Source | Depends on where the Sysmac Studio is connected and the system configuration. | Source details | NX Unit | Detection timing | During refreshing |
| Error attributes | Level | Minor fault | Recovery | When safety output terminal goes inactive after cause of error is removed | Log category | System |
| Effects | User program | Continues. | Operation | The safety output terminal retains the safe state until the error is cleared. | | |
| System-defined variables | Variable | | Data type | | Name | |
| | None | | None | | None | |
| Cause and correction | Assumed cause | | Correction | | Prevention | |
| | There is a ground fault on the output signal line. | | Check the external wiring. | | Set the parameters and wire the system considering the corrections that are given on the left. | |
| Attached information | Attached information 1: Terminal number | | | | | |
| Precautions/ Remarks | None | | | | | |

<table>
<tr><td>Event name</td><td colspan="3">Stuck-at-high Detected at Safety Output</td><td>Event Code</td><td colspan="2">65280000 hex</td></tr>
<tr><td>Meaning</td><td colspan="6">It was detected that the safety output terminal is stuck ON.</td></tr>
<tr><td>Source</td><td colspan="2">Depends on where the Sysmac Studio is connected and the system configuration.</td><td>Source details</td><td>NX Unit</td><td>Detection timing</td><td>During refreshing</td></tr>
<tr><td rowspan="2">Error attributes</td><td>Level</td><td>Minor fault</td><td>Recovery</td><td>When safety output terminal goes inactive after cause of error is removed</td><td>Log category</td><td>System</td></tr>
<tr><td>Effects</td><td>User program</td><td>Continues.</td><td>Operation</td><td colspan="3">The safety output terminal retains the safe state until the error is cleared.</td></tr>
<tr><td rowspan="2">System-defined variables</td><td colspan="2">Variable</td><td colspan="2">Data type</td><td colspan="2">Name</td></tr>
<tr><td colspan="2">None</td><td colspan="2">None</td><td colspan="2">None</td></tr>
<tr><td rowspan="5">Cause and correction</td><td colspan="2">Assumed cause</td><td colspan="2">Correction</td><td colspan="2">Prevention</td></tr>
<tr><td colspan="2">The positive power supply line is in contact with the output signal line.</td><td colspan="2">Check the external wiring.</td><td colspan="2" rowspan="2">Set the parameters and wire the system considering the corrections that are given on the left.</td></tr>
<tr><td colspan="2">The output power supply is outside the specifications.</td><td colspan="2">Check the output power supply.</td></tr>
<tr><td colspan="2">The internal circuit is faulty.</td><td colspan="2" rowspan="2">Cycle the power supply. If the error occurs again, replace the Unit.</td><td colspan="2" rowspan="2">If cycling the power supply restores normal operation, there may be excessive noise near the Unit. Implement noise countermeasures.</td></tr>
<tr><td colspan="2">A memory error or signal error occurred due to a transient cause, such as a software error or excessive noise.</td></tr>
<tr><td>Attached information</td><td colspan="6">Attached information 1: Terminal number</td></tr>
<tr><td>Precautions/ Remarks</td><td colspan="6">None</td></tr>
</table>

## ● Other Errors

| | | | | | | |
|---|---|---|---|---|---|---|
| **Event name** | NX Message Communications Error | | | **Event code** | 80220000 hex | |
| **Meaning** | An error was detected in message communications for an NX Unit and the message frame was discarded. | | | | | |
| **Source** | Depends on where the Sysmac Studio is connected and the system configuration. | | **Source details** | NX Unit | **Detection timing** | During NX message communications |
| **Error attributes** | **Level** | Observation | **Recovery** | --- | **Log category** | System |
| **Effects** | **User program** | Continues. | **Operation** | Not affected. | | |
| **System-defined variables** | **Variable** | | **Data type** | | **Name** | |
| | None | | None | | None | |
| **Cause and correction** | **Assumed cause** | | **Correction** | | **Prevention** | |
| | The message communications load is high. | | Reduce the number of times that instructions are used to send NX messages.<br><br>Refer to the appendix of the *NJ/NX-series Instructions Reference Manual (Cat. No. W502)* for information on the instructions that send messages. | | Reduce the number of times that instructions are used to send NX messages. | |
| | The communications cable is disconnected or broken. This cause does not apply if attached information 2 is 0 (NX bus). | | Connect the communications cable securely. | | Connect the communications cable securely. | |
| | Message communications were cut off as the result of executing a synchronization or restoration operation on the Sysmac Studio or as the result of disconnecting an EtherCAT slave. | | --- | | --- | |
| **Attached information** | Attached information 1: | System information | | | | |
| | Attached information 2: | Type of communications where error occurred<br>0: NX bus<br>1: EtherCAT<br>2: Serial communications (USB)<br>65535: Internal Unit communications (routing) | | | | |
| **Precautions/ Remarks** | None | | | | | |

## ● User Access Log

| | | | | | | |
|---|---|---|---|---|---|---|
| **Event name** | Event Log Cleared | | | **Event code** | 90400000 hex | |
| **Meaning** | The event log was cleared. | | | | | |
| **Source** | Depends on where the Sysmac Studio is connected and the system configuration. | | **Source details** | NX Unit | **Detection timing** | When commanded from user |
| **Error attributes** | **Level** | Information | **Recovery** | --- | **Log category** | Access |
| **Effects** | **User program** | Continues. | **Operation** | Not affected. | | |
| **System-defined variables** | **Variable** | | **Data type** | | **Name** | |
| | None | | None | | None | |
| **Cause and correction** | **Assumed cause** | | **Correction** | | **Prevention** | |
| | The event log was cleared by the user. | | --- | | --- | |
| **Attached information** | Attached information 1: | Cleared events<br>1: The system event log was cleared.<br>2: The access event log was cleared. | | | | |
| **Precautions/ Remarks** | None | | | | | |

| | | | | | | |
|---|---|---|---|---|---|---|
| **Event name** | Memory All Cleared | | | **Event code** | 90430000 hex | |
| **Meaning** | The Unit settings were cleared. | | | | | |
| **Source** | Depends on where the Sysmac Studio is connected and the system configuration. | | **Source details** | NX Unit | **Detection timing** | When commanded from user |
| **Error attributes** | **Level** | Information | **Recovery** | --- | **Log category** | Access |
| **Effects** | **User program** | Continues. | **Operation** | The Unit settings are cleared. | | |
| **System-defined variables** | **Variable** | | **Data type** | | **Name** | |
| | None | | None | | None | |
| **Cause and correction** | **Assumed cause** | | **Correction** | | **Prevention** | |
| | The Clear All Memory operation was performed. | | --- | | --- | |
| **Attached information** | Attached information 1: | Unit number of the NX Unit where the Clear All Memory operation was performed | | | | |
| | Attached information 2: | Execution results<br>0: Successful<br>1: Hardware error<br>2: Initialization failed<br>3: Initialization not possible | | | | |
| **Precautions/ Remarks** | Refer to the attached information for the results of the Clear All Memory operation. | | | | | |

# 12-4 Checking Errors of CIP Safety Target Devices Using Sysmac Studio

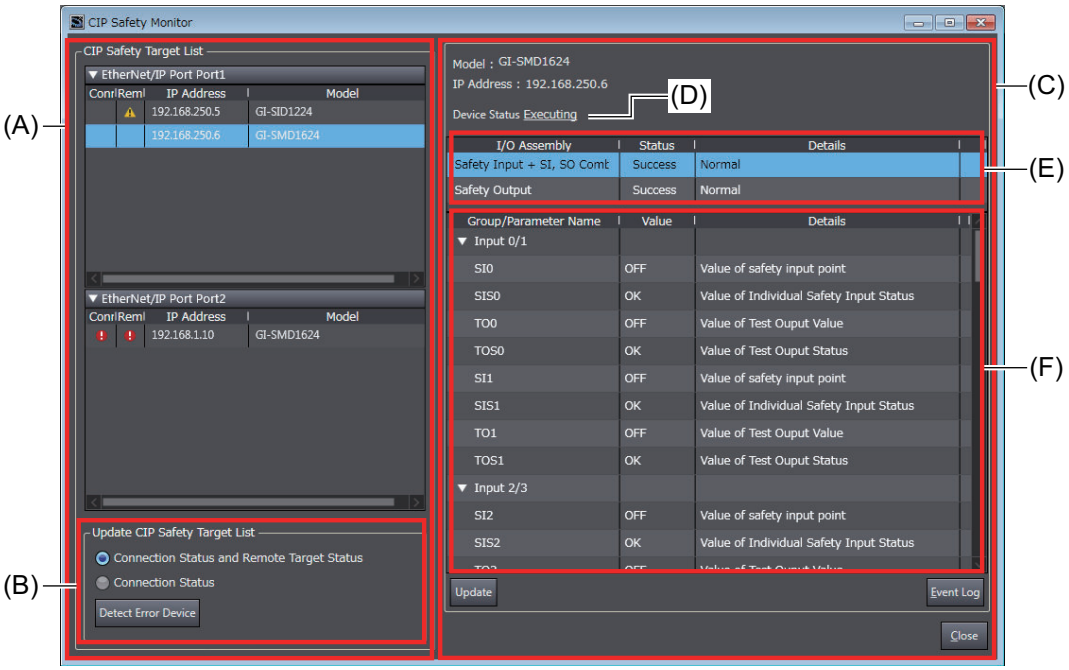
You can check errors of the CIP Safety target devices by using Sysmac Studio.

## 12-4-1 Troubleshooting the CIP Safety Target Device Errors

Sysmac Studio has the CIP Safety Monitor function.
The CIP Safety Monitor function allows you to monitor the device status of the CIP Safety target devices, the connection status with the Safety CPU Unit, as well as the parameter monitor values. If you are using an OMRON CIP Safety target device, the event logs can also be checked.

### Part Names and Functions of the CIP Safety Monitor

This section describes the names and functions of the components on the CIP Safety Monitor.




| Letter | Name | Function |
|---|---|---|
| (A) | CIP Safety Target List | A list of CIP Safety target devices registered in the Safety CPU Unit. |
| (B) | Update CIP Safety Target List | Updates the status of the CIP Safety Target List. |
| (C) | Monitor Information | Monitor information of the CIP Safety target device selected in the CIP Safety Target List. |
| (D) | Device Status | Device status information of the CIP Safety target device. |
| (E) | Connection Status | Status information of the connections configured for the CIP Safety target device. |
| (F) | Parameter Monitor Value | Information of monitored parameters of the CIP Safety target device. |

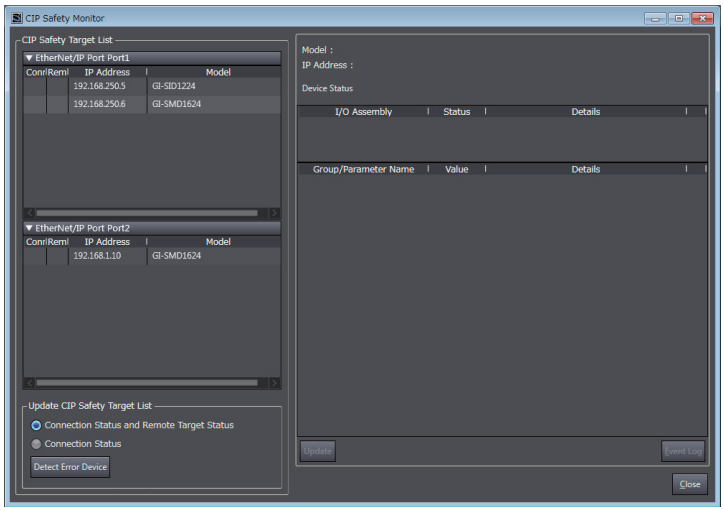
## Starting the CIP Safety Monitor

Use the following procedure to start the CIP Safety Monitor.

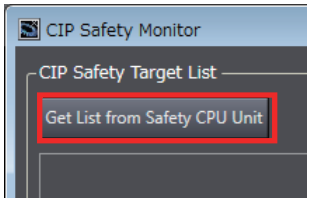
**1** Select **Online** from the **Controller** Menu. Or, click the **Go Online** Button ( ) in the toolbar.

**2** In the Multiview Explorer, select the Safety CPU Unit in the Controller Selection Box.

**3** Select **Tools** - **CIP Safety Monitor**. Or, click the **CIP Safety Monitor** Button ( ) in the toolbar.
If the information of the CIP Safety target device matches between the Safety CPU Unit and Sysmac Studio, **CIP Safety Target List** displays CIP Safety target devices that are registered in the Safety CPU Unit as shown below.




If the list of CIP Safety target devices does not appear, click the **Get List from Safety CPU Unit** Button. This will retrieve the connection settings from the Safety CPU Unit and display the device data in the list.



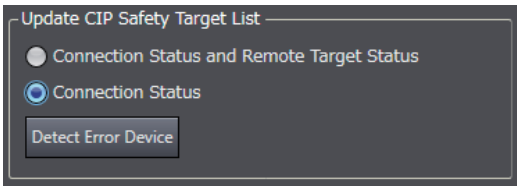

**Precautions for Correct Use**

The CIP Safety Monitor function is available only when the Safety CPU Unit is in the RUN or DEBUG mode.

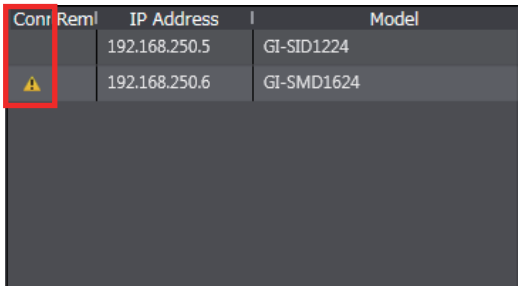
## Identifying a Target Device with the Connection Error and its Cause

In the CIP Safety Target List, you can identify a CIP Safety target device with a connection error and its cause.

**1** Go to **Update CIP Safety target list** and select the checkbox for **Connection Status**, and then click the **Detect Error Device** Button.

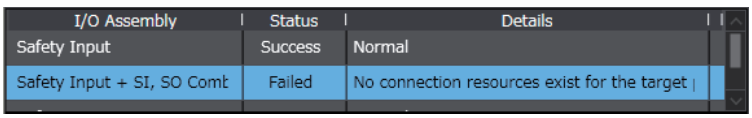



In the **CIP Safety Target List**, the display of the connection status error icon is refreshed.
If an error is present in the connection, an error icon is displayed.




| Icon | Description |
|---|---|
|  | A connection between the target device cannot be established. |
|  | No target device is found. |

**2** In the **CIP Safety Target List**, click a CIP Safety target device where a connection error is present.
The monitor information of the selected CIP Safety target is updated.
The connections configured to the CIP Safety target device and their statuses are displayed in the list. You will be able to identify the cause of the error by checking the details.

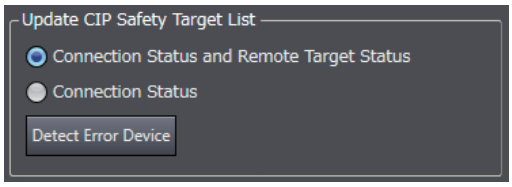



## Checking the Parameter Monitor Values

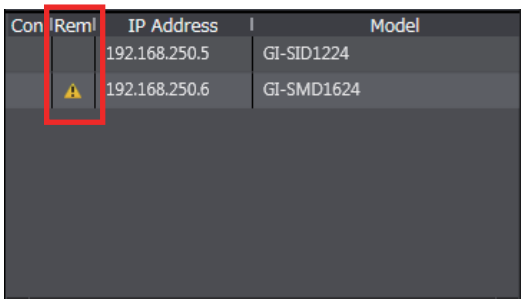
The supported parameters defined in the EDS file can be monitored.

The cause of the error can be identified by monitoring the CIP Safety device-specific I/O data and status information.

**1** Go to **Update CIP Safety target list** and select the checkbox for **Connection Status and Remote Target Status**, and then click the **Detect Error Device** Button.

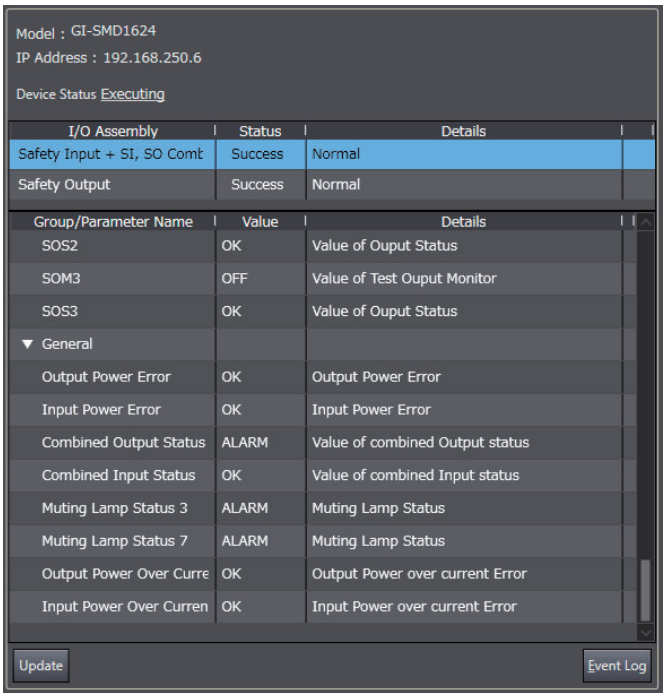



In the **CIP Safety Target List**, the display of the remote target status error icon is refreshed. If an error is present in the device status of the CIP Safety target devices, an error icon is displayed.




| Icon | Description |
|---|---|
|  | An error is present in the device status. |
|  | No target device is found. |

**2** In the **CIP Safety Target List**, click a CIP Safety target device where an error is present. In the monitor information of the selected CIP Safety target, the device status, connection status, and parameter monitor values are displayed.




For details on the parameter monitor values, refer to the manuals for the corresponding CIP Safety target devices.

**Precautions for Correct Use**

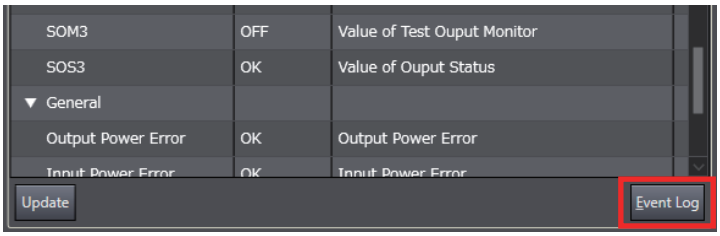
The monitor values of CIP Safety Monitor are not automatically refreshed. Click the Update Button to refresh the values.

## Checking the Event Logs

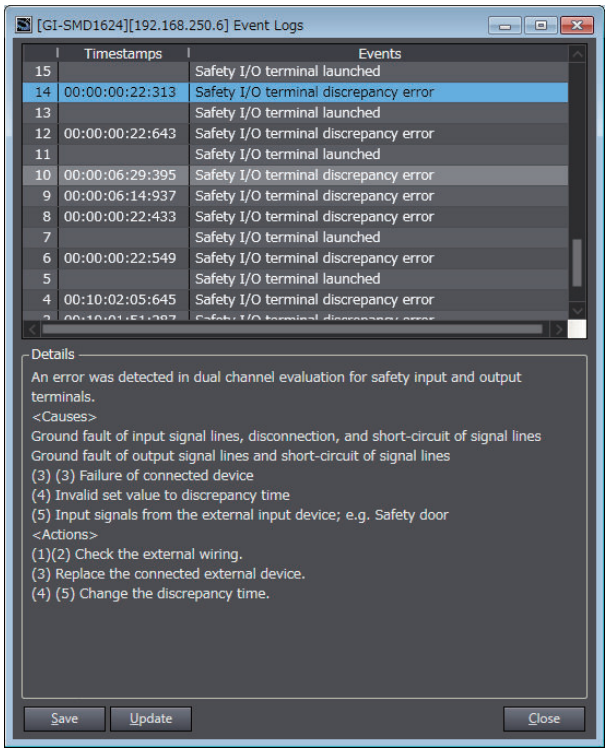
You can check event logs of the OMRON CIP Safety target devices.

1. Go to **CIP Safety Target List** and select an OMRON CIP Safety target device that you want to check the event logs.

2. Click the **Event Log** Button.




The event logs are displayed.




For details on the event logs, refer to the manuals for the corresponding CIP Safety target devices.

## 12-4-2 CIP Safety Connection Status Codes and Troubleshooting

This section explains how to identify and correct errors based on the connection status of CIP Safety connection. If the CIP Safety Originator Connection Not Established error or the CIP Safety Target Missing error occurs with the Safety CPU Unit, the connection status of the CIP Safety connection is registered to the Attached information.

**Additional Information**

The connection status has the same meaning as the Connection Manager's General and Additional error response codes, as defined in the CIP specifications.

The following table shows the likely causes of the errors and the troubleshooting methods for each connection status.

| Connection status | | Description | Cause | Correction |
|---|---|---|---|---|
| General status (HEX) | Extended status (HEX) | | | |
| 01 | 0100 | Connection in use or duplicate forward open. | The connection with the CIP Safety target device is already used for connecting another originator. | Check the CIP Safety connection settings. |
| 01 | 0105 | Ownership Conflict or OUNID Mismatch. The configuration is already owned by another originator. | The connection with the CIP Safety target device is not available because it was used for connecting another originator. | Delete the memory of the CIP Safety target device and reconfigure the device. (The method for clearing the memory varies by the CIP Safety target devices.) |

| Connection status | | Description | Cause | Correction |
|---|---|---|---|---|
| General status (HEX) | Extended status (HEX) | | | |
| 01 | 0106 | Ownership Conflict or OUNID Mismatch. The output connection was already owned by another originator. | The connection with the CIP Safety target device is not available because it was used for connecting another originator. | Delete the memory of the CIP Safety target device and reconfigure the device. (The method for clearing the memory varies by the CIP Safety target devices.) |
| 01 | 0110 | Device not configured. | Attempted to open a connection for an unconfigured CIP Safety target device by selecting "Check Safety Signature (Type2a)" or "Open Only (Type2b)" for the Open Type setting. | • Use Configuration (Type1) open type for the CIP Safety target device.<br>• Configure the CIP Safety target device accordingly. (The configuration method varies by the CIP Safety target devices.) |
| 01 | 0111 | RPI not supported. May also indicate problem with connection time-out multiplier, or production inhibit time. | Attempted to open a connection using an EPI that is not supported on the device. | Check the CIP Safety connection settings and the specifications of the CIP Safety target device. |
| 01 | 0113 | Connection Manager cannot support any more connections. | It has exceeded the maximum number of connections that can be used simultaneously. | Check the CIP Safety connection settings and the specifications of the CIP Safety target device. |
| | | | Connections of tag data link communications and CIP Safety communications are used together. | Check whether a Unit Configuration Error, Combined Use of CIP Safety and Tag Data Link (342C0000 hex) observation level event occurred in the NX-series EtherNet/IP Unit which is on the path of CIP Safety communications. |
| | | | The NX502 CPU Unit is set so that CIP Safety communications cannot be used. | Check that the NX502 CPU Unit on the path of CIP Safety communications can use for CIP Safety communications. |
| 01 | 0114 | Either the vendor ID or the product code in the key segment does not match the device. | Incorrect CIP Safety target device settings or IP address settings | Check the CIP Safety connection settings and the IP address settings of the CIP Safety target device. |
| 01 | 0116 | Major or minor revision information in the key segment does not match the device. | The version of the device does not match. | Check the version of the CIP Safety target device and the CIP Safety connection settings. |
| 01 | 0117 | Invalid connection point. | The CIP Safety I/O assembly on the CIP Safety target device is not available. | Check the connection settings of the CIP Safety target device. (The configuration method varies by the CIP Safety target devices.)<br>CIP Safety Target Device is a Safety CPU Unit:<br>Check the CIP Safety I/O assembly settings on the CIP Safety target device. |

| Connection status | | Description | Cause | Correction |
|---|---|---|---|---|
| General status (HEX) | Extended status (HEX) | | | |
| 01 | 011A | Target application cannot support any more connections. | It has exceeded the maximum number of objects that can be used simultaneously for the CIP Safety target device. | Check the CIP Safety connection settings and the specifications of the CIP Safety target device. |
| 01 | 0134 | Mismatched T→O Network Connection Fixvar | Attempted to open a multicast connection for the CIP Safety target device that has already opened a single-cast connection with another originator. | Check the CIP Safety connection settings. |
| 01 | 0204 | Unconnected_Send service timed out while waiting for a response. | The IP address settings on the CIP Safety target device are incorrect, or the Ethernet cable is disconnected.<br>CIP Safety Target Device is a Safety CPU Unit:<br>The node number of the Safety CPU Unit is incorrect. | Check the CIP Safety connection settings. |
| 01 | 031F | No connection resources exist for the target path. | Attempted to open a single-cast connection for the CIP Safety target device that has already opened a connection with another originator. | Check the CIP Safety connection settings. |
| 01 | 0801 | Incompatible Multi-cast RPI. An existing connection has been established at a different RPI. | Attempted to open a multicast connection with different EPI for the CIP Safety target device that has already opened a multi-cast connection with another originator. | Check the CIP Safety connection settings. |
| 01 | 0802 | Invalid Safety Connection Size | The connection of the selected size is not available for the CIP Safety target device. | Check the CIP Safety connection settings and the CIP Safety target device settings. |
| 01 | 0805 | Invalid Ping Interval EPI Multiplier | Attempted to open a multicast connection with different timeout settings for the CIP Safety target device that has already opened a multi-cast connection with another originator. | Check the CIP Safety connection settings. |
| 01 | 0809 | Invalid Max Consumer Number | It has exceeded the maximum number of originators that can be simultaneously connected in the multicast connection with the CIP Safety target device. | Check the CIP Safety connection settings and the specifications of the CIP Safety target device. |
| 01 | 080C | SCID Mismatch. The SCID was non-zero and did not match the value in the target | Attempted to open a connection for an already configured CIP Safety target device by selecting "Check Safety Signature (Type2a)" for the Open Type setting, but the safety signature did not match. | Check the CIP Safety connection settings and the CIP Safety target device settings. |
| 01 | 080D | TUNID not set. Device is out-of-box and TUNID has not been set, so connections are not allowed. | TUNID for the CIP Safety target device is not configured. | Configure TUNID of the CIP Safety target device. (The configuration method varies by the CIP Safety target devices.) |

| Connection status | | Description | Cause | Correction |
|---|---|---|---|---|
| General status (HEX) | Extended status (HEX) | | | |
| 01 | 080E | TUNID Mismatch. The TUNID provided does not match. The message was likely routed to this node in error | TUNID for the CIP Safety target device is configured to another value. | Reconfigure TUNID for the CIP Safety target device. (The configuration method varies by the CIP Safety target devices.) |
| 02 | --- | Resources needed for the object to perform the requested service were unavailable. | There was no open resource in the CIP Safety target device. | Check the CIP Safety target device settings. |

# 12-5 Resetting Errors

Refer to the user’s manual for the connected NX bus master for the standard flow for troubleshooting.

# 12-6 Troubleshooting Flow When Errors Occur

Refer to the user’s manual for the connected NX bus master for the standard flow for troubleshooting.

# 13

# Inspection and Maintenance

This section describes the procedures for cleaning, inspecting, and replacing Safety Control Units.



**13-1 Cleaning and Inspection ..... 13-2**
13-1-1 Cleaning ..... 13-2
13-1-2 Periodic Inspections ..... 13-2
**13-2 Maintenance Procedures ..... 13-5**
13-2-1 Replacing the Safety CPU Unit ..... 13-5
13-2-2 Replacing Safety I/O Units ..... 13-7

# 13-1 Cleaning and Inspection

This section describes daily maintenance and the cleaning and inspection methods.

## 13-1-1 Cleaning

Clean the Safety Control Unit regularly as described below in order to keep it in optimal operating condition.

- Wipe the equipment over with a soft, dry cloth when performing daily cleaning.
- If dirt remains even after wiping with a soft, dry cloth, wipe with a cloth that has been wet with a sufficiently diluted detergent (2%) and wrung dry.
- A smudge may remain on the Units from gum, vinyl, or tape that was left on for a long time. Remove the smudge when cleaning.

**Precautions for Correct Use**

- Never use volatile solvents, such as paint thinner, benzene, or chemical wipes.
- Do not touch the NX bus connectors.

## 13-1-2 Periodic Inspections

Although the major components in Safety Control Unit have an extremely long life time, they can deteriorate under improper environmental conditions. Periodic inspections are thus required.
Inspection is recommended at least once every six months to a year, but more frequent inspections will be necessary in adverse environments.
Take immediate steps to correct the situation if any of the conditions in the following table are not met.

- Make sure that the Units are used within the ranges of specifications.
- Make sure that the Units are mounted and wired correctly.
- To maintain the operating reliability of the safety functions at a consistent level, diagnose the safety functions.
- Use the error log to check whether non-fatal errors have occurred.

### Periodic Inspection Items

| No. | Inspection item | Inspection details | Criteria | Correction |
|---|---|---|---|---|
| 1 | External power supply | Is the power supply voltage measured at the terminal block within standards? | Within the power supply voltage range | Use a voltage tester to check the power supply at the terminals. Take necessary steps to bring the power supply within the power supply voltage range. |
| 2 | I/O power supply | Is the power supply voltage measured at the I/O terminal block within standards? | Voltages must be within I/O specifications of each NX Unit. | Use a voltage tester to check the power voltage at the terminals. Take necessary steps to bring the I/O power supply within NX Unit standards. |

| No. | Inspection item | Inspection details | Criteria | Correction |
|---|---|---|---|---|
| 3 | Ambient environment | Is the ambient operating temperature within standards? | 0 to 55°C | Use a thermometer to check the temperature and ensure that the ambient operating temperature remains within the allowed range of 0 to 55°C. |
| | | Is the ambient operating humidity within standards? | Relative humidity must be 10% to 95% with no condensation. | Use a hygrometer to check the humidity and ensure that the ambient operating humidity remains between 10% and 95%. Make sure that condensation does not occur due to rapid changes in temperature. |
| | | Is it subject to direct sunlight? | Not in direct sunlight | Protect the Controller if necessary. |
| | | Is there an accumulation of dirt, dust, salt, metal powder, etc.? | No accumulation | Clean and protect the Controller if necessary. |
| | | Is there water, oil, or chemical sprays hitting the Controller? | No spray | Clean and protect the Controller if necessary. |
| | | Are there corrosive or flammable gases in the area of the Controller? | No gases | Check by smell or use a sensor. |
| | | Is the Unit subject to shock or vibration? | Vibration resistance and shock resistance must be within specifications. | Install cushioning or other vibration and shock absorbing equipment if necessary. |
| | | Are there noise sources near the Controller? | No significant noise sources | Either separate the Controller and noise source or protect the Controller. |
| 4 | Installation and wiring | Are the DIN Track mounting hooks for each NX Unit securely locked? | No looseness | Securely lock the DIN Track mounting hooks. |
| | | Are the cable connectors fully inserted and locked? | No looseness | Correct any improperly installed connectors. |
| | | Are there any loose screws on the End Plates (PFP-M)? | No looseness | Tighten loose screws with a Phillips-head screwdriver. |
| | | Are the NX Units connected to each other along the hookup guides and until they touch the DIN track? | You must connect and fix the NX Units to the DIN track. | Connect the NX Units to each other along the hookup guides and insert them until they touch the DIN track. |
| | | Are there any damaged external wiring cables? | No visible damage | Check visually and replace cables if necessary. |
| 5 | Safety validation testing (user testing) | Check to be sure that all safety functions operate correctly. | All functions must operate as intended. | Remove the cause of errors and check the operation of all safety functions again. |

## Tools Required for Inspections

### Required Tools

- Phillips screwdriver
- Flat-blade screwdriver
- Voltage tester or digital voltmeter
- Industrial alcohol and pure cotton cloth

### Tools Required Occasionally

- Oscilloscope
- Thermometer and hygrometer

# 13-2 Maintenance Procedures

If the inspection reveals any problems that require you to replace a Safety Control Unit, observe the following precautions.

- Never disassemble, repair, or modify a Safety Control Unit. This will compromise the integrity of the safety function and is dangerous.
- Make sure that you can replace the Unit under safe conditions.
- Perform all replacements with the power supply turned OFF to prevent electric shock, or unexpected movement of the machinery.
- Check the new Unit to make sure that there are no errors.

**Precautions for Safe Use**

After you replace the Unit, reset the program and all configuration settings that are necessary to resume operation. Make sure that the safety functions operate normally before you start actual operation.

Remove the faulty Unit, and then replace and wire the new Unit.
Refer to *2-1-2 Part Names and Functions* on page 2-8and to *Section 3 Installation and Wiring* on page 3-1 for information on installing, removing, and wiring Units.

The following section introduces how to replace Safety CPU Unit and Safety I/O Units.

**Precautions for Correct Use**

The backup/restore functions of the NJ/NX-series CPU Unit do not apply to Safety Control Unit. Use the Sysmac Studio when you replace a Unit.

## 13-2-1 Replacing the Safety CPU Unit

### Precautions before Replacing the Unit

- Before replacing the Unit, make sure there is a Sysmac Studio project file that corresponds to the current safety control system. Alternatively, you can upload the project file from the actual safety control system.
- Make sure that the Sysmac Studio project file is the intended file before replacing the Unit.
- Replacement work must be performed only by personnel with knowledge of safety controls.
- To ensure the safety of all workers, turn OFF the power supply to all hazard sources (i.e., actuators, etc.). Alternatively, place the NJ/NX-series CPU Unit in PROGRAM mode.

### Replacement Procedure

Remove the Safety CPU Unit to replace and attach the new Safety CPU Unit.

## ● When Sysmac Studio Is Connected to NJ/NX-series CPU Unit

You can use this connection method only if you use an NJ/NX-series CPU Unit and EtherCAT Coupler Unit.

***1*** Connect the Sysmac Studio online to the NJ/NX-series CPU Unit.
When you add a new Safety CPU Unit to the system, a Slave Initialization Error will occur in the NJ/NX-series CPU Unit.

***2*** Select **Synchronization** from the **Controller** Menu. Click the **Transfer to Controller** Button in the Synchronization Window to transfer the Slave Terminal configuration information from the computer to the Safety CPU Unit.
Refer to *7-2-2 Transfer Procedure for a Connection to NJ/NX-series CPU Unit* on page 7-8 for a detailed procedure.

***3*** Reset the error from the Troubleshooting Dialog Box.

***4*** Select the Safety CPU Unit as the Controller.
The Sysmac Studio goes online with the Safety CPU Unit.

***5*** Place the Safety CPU Unit in DEBUG mode.
Refer to *7-4 Changing to DEBUG Mode* on page 7-17 for a detailed procedure.

***6*** On the Safety CPU Unit Setup and Programming View, select **Safety Validation** from the **Controller** Menu to transfer the safety programs to the non-volatile memory in the Safety CPU Unit.
Refer to *7-11-1 Performing Safety Validation* on page 7-65 for a detailed procedure.

***7*** Change the operating mode of the Safety CPU Unit to RUN mode. Or, cycle the power supply to Safety CPU Unit.
Refer to *7-11-2 Changing to RUN Mode* on page 7-67 for a detailed procedure.

## ● When Sysmac Studio Is Connected to Communications Coupler Unit

***1*** Place the Sysmac Studio online with the Communications Coupler Unit.
When you add a new Safety CPU Unit to the system, a Slave Initialization Error will occur in the Communications Coupler Unit.

***2*** Right-click the Communications Coupler Unit and select **Transfer to Controller** from the menu to transfer the Slave Terminal configuration information to the Safety CPU Unit.
Refer to *7-2-3 Transfer Procedure for a Connection to Communications Coupler Unit* on page 7-8 for a detailed procedure.

***3*** Reset the error from the Troubleshooting Dialog Box.

***4*** Select the Safety CPU Unit as the Controller.
The Sysmac Studio goes online with the Safety CPU Unit.

***5*** Place the Safety CPU Unit in DEBUG mode.

Refer to *7-4 Changing to DEBUG Mode* on page 7-17 for a detailed procedure.

***6*** On the Safety CPU Unit Setup and Programming View, select **Safety Validation** from the **Controller** Menu to transfer the safety programs to the non-volatile memory.
Refer to *7-11-1 Performing Safety Validation* on page 7-65 for a detailed procedure.

***7*** Change the operating mode of the Safety CPU Unit to RUN mode. Or, cycle the power supply to Safety CPU Unit.
Refer to *7-11-2 Changing to RUN Mode* on page 7-67 for a detailed procedure.

### ● When Using an SD Memory Card

Refer to *Section 10 Safety Unit Restore* on page 10-1 for details when you use an SD Memory Card.

## Checking after Replacing a Unit

- After a Unit is replaced, make sure that the intended data was transferred to the Safety CPU Unit by using the following methods.
  a) Make sure that the safety signature that is shown in the Safety Properties Dialog Box for the safety project is the same as the safety signature that is shown in the Controller Status Pane.
  b) When you use an SD Memory Card, make sure that the safety signature that is shown in the seven-segment indicator of the Safety CPU Unit is the intended signature.
- After the replacement is completed, always perform user testing to make sure that the safety functions operate correctly.
- If necessary, clear the event log of any events that remain in Safety CPU Unit due to the replacement work.

## 13-2-2 Replacing Safety I/O Units

This section describes the ACR (auto configuration restart) function.

### ● Precautions before Replacing the Units

- The replaced Safety I/O Units must be in the default status before the replacement.
  If you are unsure of whether the Safety I/O Units are in the default state, perform the Clear All Memory operation for all Safety I/O Units that were replaced. Refer to *7-16 Restarting and Clearing All Memory* on page 7-78 for detailed procedures.
- Replacement work must be performed only by personnel with knowledge of safety controls.
- To ensure the safety of all workers, turn OFF the power supply to all hazard sources (i.e., actuators, etc.). Alternatively, place the NJ/NX-series CPU Unit in PROGRAM mode.

### ● Replacement Procedure

***1*** Record the relationship between the wiring and the terminal numbers before you remove the terminal block from the Safety I/O Units.

***2*** Remove the Safety I/O Units to replace.

3 Mount the new Safety I/O Units.

4 Return the terminal block to the new Safety I/O Units.

5 Cycle the power supply to the Safety CPU Unit.

## Checking after Replacing Safety I/O Units

- After the replacement is completed, always perform user testing to make sure that the safety functions operate correctly. Make sure that the terminal block is inserted into the correct location on the Safety I/O Units, and check by performing user testing.



**Precautions for Correct Use**

**Checking the Serial Numbers of NX Units**

- If the **Serial Number Check Method**setting on the NX bus master is set to **Setting = Actual device**, **Serial Number Check Method** temporarily change this setting to **No check**, and then replace the NX Unit. Get the serial number of the new NX Unit, and then set the **Serial Number Check Method** setting on the NX bus master to **Setting = Actual device** again. If you replace the NX Unit with the **Serial Number Check Method** setting set to **Setting = Actual** device, a *Unit Configuration Verification Error* will occur.
- Refer to the user's manual for the NX bus master for details on the serial number checking function for NX bus master.



**Additional Information**

If you replace a Safety I/O Unit while the Safety CPU Unit is in operation, a communications error event will be logged in the following Units. After the replacement, clear the event logs as necessary.

- Safety CPU Unit
- Safety I/O Units on the same Slave Terminal that were not replaced
- The NX bus master to which the Safety CPU Unit or Safety I/O Unit where a communications error was detected is connected



# Appendices

The appendices provide the general specifications, dimensions of the Units, application examples, and other information.



**A-1 Dimensions ..... A-3**
A-1-1 Safety CPU Unit ..... A-3
A-1-2 Safety I/O Units ..... A-4

**A-2 NX Objects ..... A-5**
A-2-1 Format of NX Object Descriptions ..... A-5
A-2-2 Safety CPU Unit (NX-SL3300/NX-SL3500) ..... A-5
A-2-3 Safety CPU Unit (NX-SL5500/NX-SL5700) ..... A-7
A-2-4 NX-SID800 Safety Input Unit ..... A-11
A-2-5 NX-SIH400 Safety Input Unit ..... A-16
A-2-6 NX-SOD400 Safety Output Unit ..... A-21
A-2-7 NX-SOH200 Safety Output Unit ..... A-25

**A-3 Application Examples ..... A-30**
A-3-1 Emergency Stop Pushbutton Switches ..... A-30
A-3-2 Safety Doors ..... A-32
A-3-3 Safety Laser Scanners ..... A-36
A-3-4 Safety Door Switches with Magnetic Locks and Key Selector Switches ..... A-39
A-3-5 Enable Switches ..... A-43
A-3-6 Two-hand Switches ..... A-47
A-3-7 D40A Non-contact Door Switches ..... A-50
A-3-8 D40A-2/D40Z Non-contact Door Switches ..... A-53
A-3-9 Safety Mats and Safety Light Curtains ..... A-56
A-3-10 Safety Edges ..... A-61
A-3-11 Single Beam Safety Sensor ..... A-63
A-3-12 Multi-axis Servo Drives (STO) ..... A-66

**A-4 Change Tracking ..... A-70**

**A-5 Safety CPU Unit Status ..... A-72**
A-5-1 NX-SL3300/NX-SL3500 ..... A-72
A-5-2 NX-SL5500/NX-SL5700 ..... A-72

**A-6 I/O Ports of Safety I/O Units ..... A-74**
A-6-1 NX-SIH400 Safety Input Unit ..... A-74
A-6-2 NX-SID800 Safety Input Unit ..... A-76
A-6-3 NX-SOH200 Safety Output Unit ..... A-77
A-6-4 NX-SOD400 Safety Output Unit ..... A-78

**A-7 CIP Response Codes ..... A-81**
A-7-1 General Status Codes ..... A-81

A-7-2 Extended Status Codes ..... A-83

**A-8 Icon list for Safety Slave Unit Parameters ..... A-86**
A-8-1 External Device Icons for Input Devices ..... A-86
A-8-2 Contact Icons for Input Devices ..... A-90
A-8-3 External Device Icons for Output Devices ..... A-92
A-8-4 Contact Icons for Output Devices ..... A-93

**A-9 Calculating I/O Sizes for CPU Rack or Slave Terminals ..... A-94**

**A-10 Printing ..... A-95**
A-10-1 Selecting the Items to Print ..... A-95
A-10-2 Items that are Printed ..... A-95

**A-11 List of Screwless Clamping Terminal Block Models ..... A-97**
A-11-1 Model Notation ..... A-97
A-11-2 List of Terminal Block Models ..... A-97

**A-12 I/O Response Times for Communications between NX Units on EtherNet/IP Slave Terminals ..... A-99**

**A-13 Units That Support Communications between NX Units ..... A-101**

**A-14 Checking the Signature Code on the Seven-segment Indicator ..... A-102**

**A-15 Execution Scenarios for the Simple Automatic Test ..... A-103**

**A-16 Differences in Checking Operation between the Simulator and Safety CPU Unit ..... A-106**

**A-17 I/O Data Enable Flag for CIP Safety Connections ..... A-107**

**A-18 Safety: Update Configurations and Setup Transfer Data ..... A-109**

**A-19 CIP Safety Target Device Operation ..... A-110**

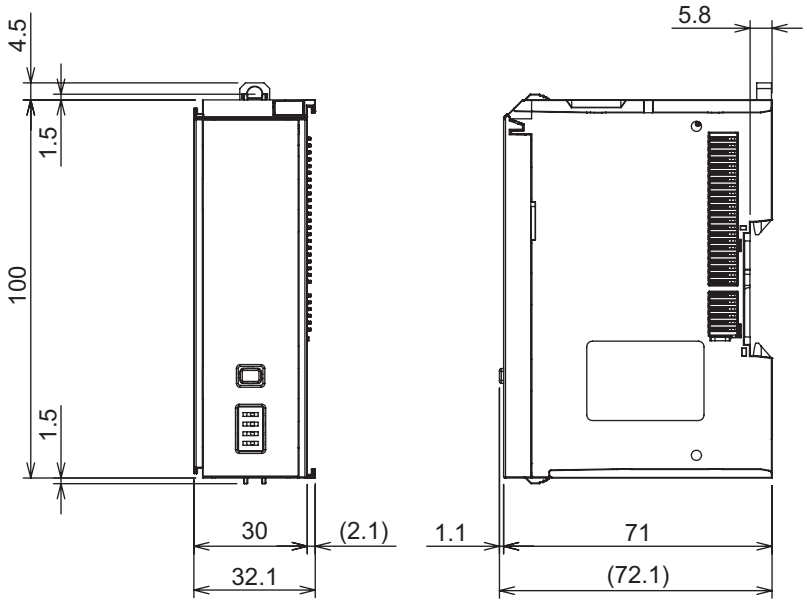
**A-20 Version Information ..... A-113**
A-20-1 Relationship between Unit Versions and Sysmac Studio Versions ..... A-113
A-20-2 Functions That Were Added or Changed for Each Version ..... A-118

# A-1 Dimensions

Dimensions of the Units are shown as follows. The unit of dimension is millimeter.

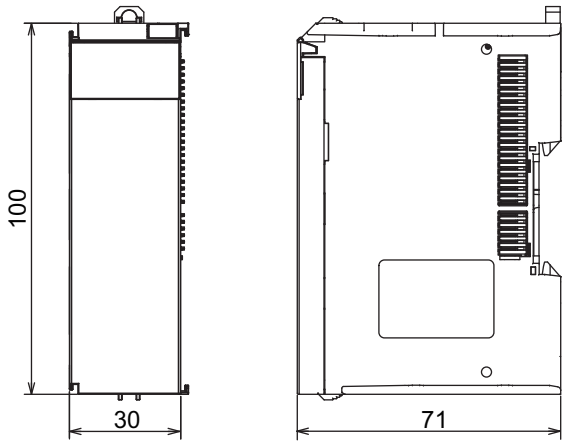
## A-1-1 Safety CPU Unit

### NX-SL5500/NX-SL5700



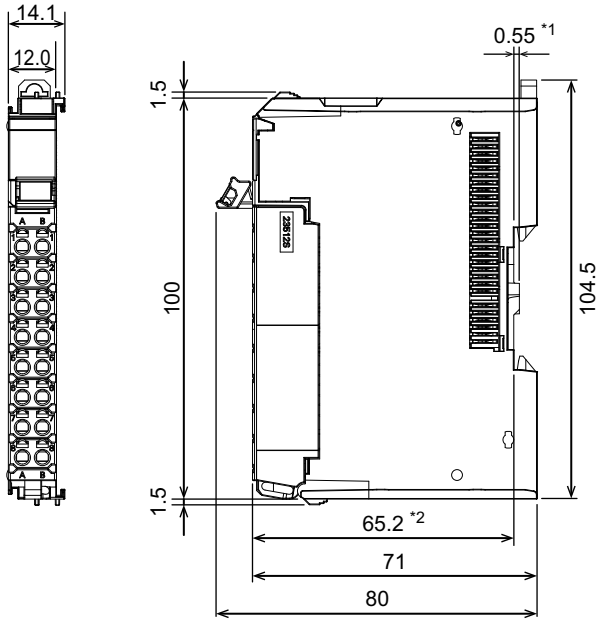

Unit: mm

### NX-SL3300/NX-SL3500



## A-1-2 Safety I/O Units

### NX-SID800/NX-SIH400/NX-SOD400/NX-SOH200




Unit: mm

*1. The dimension is 1.35 mm for Units with lot numbers through December 2014.
*2. The dimension from the attachment surface of the DIN Track to the front surface of the Safety I/O Unit.

# A-2 NX Objects

## A-2-1 Format of NX Object Descriptions

In this manual, NX objects are described with the following format.

| Index (hex) | Subindex (hex) | Object name | Default value | Data range | Unit | Data type | Access | I/O allocation | Data attribute |
|---|---|---|---|---|---|---|---|---|---|
| | | | | | | | | | |

Index (hex) : This is the index of the NX object that is expressed as a four-digit hexadecimal number.
Subindex (hex) : This is the subindex of the NX object that is expressed as a two-digit hexadecimal number.
Object name : This is the name of the object. For a subindex, this is the name of the subindex.
Default value : This is the value that is set by default.
Data range : For a read-only (RO) NX object, this is the range of the data you can read. For a read-write (RW) NX object, this is the setting range of the data.
Unit : The unit is the physical units.
Data Type : This is the data type of the object.
Access : This data tells if the object is read-only or read/write.
RO: Read-only
RW: Read/write
I/O allocation : This tells whether I/O allocation is allowed.
Data attribute : This is the timing when changes to writable NX objects are enabled.
Y: Enabled by restarting
N: Enabled at all times
---: Write-prohibited

## A-2-2 Safety CPU Unit (NX-SL3300/NX-SL3500)

### Unit Information Object

This object gives the product information.

| Index (hex) | Subindex (hex) | Object name | Default value | Data range | Unit | Data Type | Access | I/O allocation | Data attribute |
|---|---|---|---|---|---|---|---|---|---|
| 1000 | --- | NX Bus Identity | --- | --- | --- | --- | --- | --- | --- |
| | 00 | Number of Entries | 7 | 7 | --- | USINT | RO | Not possible. | --- |
| | 02 | Model | *1 | --- | --- | ARRAY [0..11] OF BYTE | RO | Not possible. | --- |
| | 03 | Device Type | 00000A00 hex | --- | --- | UDINT | RO | Not possible. | --- |
| | 04 | Product Code | *2 | --- | --- | UDINT | RO | Not possible. | --- |
| | 05 | Vendor Code | 1 | --- | --- | UDINT | RO | Not possible. | --- |
| | 06 | Unit Version | *3 | --- | --- | UDINT | RO | Not possible. | --- |
| | 07 | Serial Number | *4 | 00000000 to FFFFFFFF hex | --- | UDINT | RO | Not possible. | --- |
| 1001 | --- | Production Info | --- | --- | --- | --- | --- | --- | --- |
| | 00 | Number of Entries | 4 | 4 | --- | USINT | RO | Not possible. | --- |
| | 01 | Lot Number | *5 | 00000000 to FFFFFFFF hex | --- | UDINT | RO | Not possible. | --- |
| | 02 | Hardware Version | *6 | --- | --- | ARRAY [0..19] OF BYTE | RO | Not possible. | --- |
| | 03 | Software Version | *7 | --- | --- | ARRAY [0..19] OF BYTE | RO | Not possible. | --- |

*1. The product models are assigned in ascending order from the lowest number of array elements. Unused elements are padded with spaces.

*2. The product codes are assigned for each product model.
NX-SL3300: 00A03300 hex
NX-SL3500: 00A03500 hex

*3. Bits 24 to 31: Integer part of the Unit version.
Bits 16 to 23: Fractional part of the Unit version.
Bits 0 to 15: Reserved
(Example) For Ver.1.0, 0100□□□□ hex

*4. The unique serial number of the product is given.
Bits 0 to 31: Serial number

*5. The year, month, and day of production are assigned to the "lot number".
Bits 24 to 31: Day of manufacture
Bits 16 to 23: Month of manufacture
Bits 8 to 15: Year of manufacture
Bits 0 to 7: Reserved

*6. The hardware version is given in order in the lowest elements of the array. Unused elements are padded with spaces.

*7. The software version is given in order in the lowest elements of the array. Unused elements are padded with spaces.

## Objects That Accept I/O Allocations

These objects accept I/O allocations.
You cannot access the objects with the Read NX Unit Object instruction or the Write NX Unit Object instruction.
Refer to the *NJ/NX-series Instructions Reference Manual (Cat. No. W502)* for details on the Read NX Unit Object instruction or the Write NX Unit Object instruction.

| Index (hex) | Subindex (hex) | Object name | Default value | Data range | Unit | Data Type | Access | I/O allocation | Data attribute |
|---|---|---|---|---|---|---|---|---|---|
| 6004 | --- | Status | --- | --- | --- | --- | --- | --- | --- |
| | 00 | Number of Entries | 1 | 1 | --- | USINT | RO | Not possible. | --- |
| | 01 | Safety CPU Status | 0000 hex | 0000 to 000F hex*1 | --- | WORD | RO | Possible. | --- |

*1. The details of the "Safety CPU Status" are as follows:
Bit 0: Normal Operating
Bit 1: Program Operating
Bit 2: Program No Fault
Bit 3: Safety Master Connection Status
Bits 4 to 15: Reserved

# A-2-3 Safety CPU Unit (NX-SL5500/NX-SL5700)

## Unit Information Object

This object gives the product information.

| Index (hex) | Subindex (hex) | Object name | Default value | Data range | Unit | Data type | Access | I/O allocation | Data attribute |
|---|---|---|---|---|---|---|---|---|---|
| 1000 | --- | NX Bus Identity | --- | --- | --- | --- | --- | --- | --- |
| | 00 | Number of Entries | 7 | 7 | --- | USINT | RO | Not supported | --- |
| | 02 | Model | *1 | --- | --- | ARRAY [0..11] OF BYTE | RO | Not supported | --- |
| | 03 | Device Type | 00000A00 hex | --- | --- | UDINT | RO | Not supported | --- |
| | 04 | Product Code | *2 | --- | --- | UDINT | RO | Not supported | --- |
| | 05 | Vendor Code | 1 | --- | --- | UDINT | RO | Not supported | --- |
| | 06 | Unit Version | *3 | --- | --- | UDINT | RO | Not supported | --- |
| | 07 | Serial Number | *4 | 00000000 to FFFFFFFF hex | --- | UDINT | RO | Not supported | --- |
| 1001 | --- | Production Info | --- | --- | --- | --- | --- | --- | --- |
| | 00 | Number of Entries | 4 | 4 | --- | USINT | RO | Not supported | --- |
| | 01 | Lot Number | *5 | 00000000 to FFFFFFFF hex | --- | UDINT | RO | Not supported | --- |
| | 02 | Hardware Version | *6 | --- | --- | ARRAY [0..19] OF BYTE | RO | Not supported | --- |
| | 03 | Software Version | *7 | --- | --- | ARRAY [0..19] OF BYTE | RO | Not supported | --- |

*1. The product models are assigned in ascending order from the lowest number of array elements. Unused elements are padded with spaces.

*2. The product codes are assigned for each product model.

NX-SL5700: 00A05700 hex

*3. Bits 24 to 31: Integer part of the Unit version
Bits 16 to 23: Fractional part of the Unit version
Bits 0 to 15: Reserved
(Example) For Ver.1.0, 0100□□□□ hex

*4. The unique serial number of the product is given.
Bits 0 to 31: Serial number

*5. The date of manufacture is given for the "lot number".
Bits 24 to 31: Day of manufacture
Bits 16 to 23: Month of manufacture
Bits 8 to 15: Year of manufacture
Bits 0 to 7: Reserved

*6. The hardware version is given in order in the lowest elements of the array. Unused elements are padded with spaces.

*7. The software version is given in order in the lowest elements of the array. Unused elements are padded with spaces.

## Objects That Accept I/O Allocations

These objects accept I/O allocations.

| Index (hex) | Subindex (hex) | Object name | Default value | Data range | Unit | Data type | Access | I/O allocation | Data attribute |
|---|---|---|---|---|---|---|---|---|---|
| 6004 | --- | Status | --- | --- | --- | --- | --- | --- | --- |
| | 00 | Number of Entries | 1 | 1 | --- | USINT | RO | Not supported | --- |
| | 01 | Safety CPU Status | 0000 hex | 0000 to 007F hex*1 | --- | WORD | RO | Supported | --- |

*1. The details of the Safety CPU Status are as follows:
Bit 0: Normal Operating, All safety master connections established
Bit 1: Program Operating
Bit 2: Program No Fault
Bit 3: Safety Master Connection Status
Bit 4: FSoE Master Connection Status
Bit 5: CIP Safety Originator Connection Status
Bit 6: CIP Safety Target Connection Status
Bits 7 to 15: Reserved

## Other Objects

This section lists other objects.

## ● Safety Signature Objects

| Index (hex) | Subindex (hex) | Object name | Default value | Data range | Unit | Data type | Access | I/O allocation | Data attribute |
|---|---|---|---|---|---|---|---|---|---|
| 5002 | --- | Safety Signature | --- | --- | --- | --- | --- | --- | --- |
| | 00 | Number of Entries | 2 | 2 | --- | USINT | RO | Not supported | --- |
| | 01 | Last Modified*1 | 0x0000000000000000 | 0x0000000000000000 to 0xFFFFFFFFFFFFFFFF | --- | ULINT | RO | Not supported | --- |
| | 02 | Signature*2 | 0x0000 | 0xFFFF | --- | UINT | RO | Not supported | --- |

*1. The default value is given if it is not validated. If safety validation is executed, the elapsed number of seconds from 00:00:00 on January 1, 1970 (UTC) is entered as the update time of the safety signature.

*2. The default value is given if it is not validated.

## ● Node Name Objects

| Index (hex) | Subindex (hex) | Object name | Default value | Data range | Unit | Data type | Access | I/O allocation | Data attribute |
|---|---|---|---|---|---|---|---|---|---|
| 5003 | --- | Node name | --- | --- | --- | --- | --- | --- | --- |
| | 00 | Number of Entries | 1 | 1 | --- | USINT | RO | Not supported | --- |
| | 01 | Node name*1 | 0 | --- | --- | ARRAY [0..79] OF BYTE | RO | Not supported | --- |

*1. If the node name is not set, the default value is given.
If the node name is specified, the node name in the ASCII code will be entered.

### ● Status Objects

| Index (hex) | Subindex (hex) | Object name | Default value | Data range | Unit | Data type | Access | I/O allocation | Data attribute |
|---|---|---|---|---|---|---|---|---|---|
| 5001 | --- | Status | --- | --- | --- | --- | --- | --- | --- |
| | 00 | Number of Entries | 8 | 8 | --- | USINT | RO | Not supported | --- |
| | 01 | Safety CPU Status | 0000 hex | 0000 to 007F hex | --- | WORD | RO | Not supported | --- |
| | 02 | Normal Operating | 0 | 0-1 | --- | BOOL | RO | Not supported | --- |
| | 03 | Program Operating | 0 | 0-1 | --- | BOOL | RO | Not supported | --- |
| | 04 | Program No Fault | 0 | 0-1 | --- | BOOL | RO | Not supported | --- |
| | 05 | Safety Master Connection Status | 0 | 0-1 | --- | BOOL | RO | Not supported | --- |
| | 06 | FSoE Master Connection Status | 0 | 0-1 | --- | BOOL | RO | Not supported | --- |
| | 07 | CIP Safety Originator Connection Status | 0 | 0-1 | --- | BOOL | RO | Not supported | --- |
| | 08 | CIP Safety Target Connection Status | 0 | 0-1 | --- | BOOL | RO | Not supported | --- |

## A-2-4 NX-SID800 Safety Input Unit

### Unit Information Objects

These objects give the product information.

| Index (hex) | Subindex (hex) | Object name | Default value | Data range | Unit | Data type | Access | I/O allocation | Data attribute |
|---|---|---|---|---|---|---|---|---|---|
| 1000 | --- | NX Bus Identity | --- | --- | --- | --- | --- | --- | --- |
| | 00 | Number of Entries | 7 | 7 | --- | USINT | RO | Not supported | --- |
| | 02 | Model | NX-SID800 | --- | --- | ARRAY [0..11] OF BYTE | RO | Not supported | --- |
| | 03 | Device Type | 00000A01 hex | --- | --- | UDINT | RO | Not supported | --- |
| | 04 | Product Code | 00A10800 hex | --- | --- | UDINT | RO | Not supported | --- |
| | 05 | Vendor Code | 1 | --- | --- | UDINT | RO | Not supported | --- |
| | 06 | Unit Version | *1 | --- | --- | UDINT | RO | Not supported | --- |
| | 07 | Serial Number | *2 | 00000000 to FFFFFFFF hex | --- | UDINT | RO | Not supported | --- |
| 1001 | --- | Production Info | --- | --- | --- | --- | --- | --- | --- |
| | 00 | Number of Entries | 4 | 4 | --- | USINT | RO | Not supported | --- |
| | 01 | Lot Number | *3 | 00000000 to FFFFFFFF hex | --- | UDINT | RO | Not supported | --- |
| | 02 | Hardware Version | *4 | --- | --- | ARRAY [0..19] OF BYTE | RO | Not supported | --- |
| | 03 | Software Version | *5 | --- | --- | ARRAY [0..19] OF BYTE | RO | Not supported | --- |

*1. Bits 24 to 31: Integer part of the Unit version
Bits 16 to 23: Fractional part of the Unit version
Bits 0 to 15: Reserved

(Example) For Ver.1.0, 0100□□□□ hex

*2. The unique serial number of the product is given.
Bits 0 to 31: Serial number

*3. The date of manufacture is given for the "lot number".
Bits 24 to 31: Day of manufacture
Bits 16 to 23: Month of manufacture
Bits 8 to 15: Year of manufacture
Bits 0 to 7: Reserved

*4. The hardware version is given in order in the lowest elements of the array. Unused elements are padded with spaces.

*5. The software version is given in order in the lowest elements of the array. Unused elements are padded with spaces.

## Objects That Accept I/O Allocations

These objects accept I/O allocations.
They cannot be accessed through message communications.

| Index (hex) | Subindex (hex) | Object name | Default value | Data range | Unit | Data type | Access | I/O allocation | Data attribute |
|---|---|---|---|---|---|---|---|---|---|
| 6000 | --- | FSoE Slave Frame Elements | --- | --- | --- | --- | --- | --- | --- |
| | 00 | Number of Entries | 3 | 3 | --- | USINT | RO | Not supported | --- |
| | 01 | FSoE Slave CMD | 00 hex | 00 to FF hex | --- | BYTE | RO | Supported | --- |
| | 02 | FSoE Slave Conn_ID | 0000 hex | 0000 to FFFF hex | --- | WORD | RO | Supported | --- |
| | 03 | FSoE Slave CRC_0 | 0000 hex | 0000 to FFFF hex | --- | WORD | RO | Supported | --- |

| Index (hex) | Subindex (hex) | Object name | Default value | Data range | Unit | Data type | Access | I/O allocation | Data attribute |
|---|---|---|---|---|---|---|---|---|---|
| 6001 | --- | Safety Input Data | --- | --- | --- | --- | --- | --- | --- |
| | 00 | Number of Entries | 1 | 1 | --- | USINT | RO | Not supported | --- |
| | 01 | Safety Input 1st Byte | 0000 hex | 0000 to FFFF hex | --- | WORD | RO | Supported | --- |

| Index (hex) | Subindex (hex) | Object name | Default value | Data range | Unit | Data type | Access | I/O allocation | Data attribute |
|---|---|---|---|---|---|---|---|---|---|
| 6002 | --- | Standard Input Data | --- | --- | --- | --- | --- | --- | --- |
| | 00 | Number of Entries | 2 | 2 | --- | USINT | RO | Not supported | --- |
| | 01 | Standard Input 1st Word | 0000 hex | 0000 to FFFF hex*1 | --- | WORD | RO | Supported | --- |
| | 02 | Standard Input 2nd Byte | 00 hex | 00 to FF hex*2 | --- | BYTE | RO | Supported | --- |

*1. The details of the "Standard Input 1st Word" are as follows:
Bit 0: Si00 Logical Value
Bit 1: Si01 Logical Value
Bit 2: Si02 Logical Value
Bit 3: Si03 Logical Value
Bit 4: Si04 Logical Value
Bit 5: Si05 Logical Value
Bit 6: Si06 Logical Value
Bit 7: Si07 Logical Value
Bit 8: Safety Connection Status
Bit 9: Safety Input Terminal Status
Bit 10: Unit Normal Status
Bit 11: IO Power Supply Error Flag
Bits 12 to 15: Reserved

*2. The details of the "Standard Input 2nd Byte" are as follows:
Bit 0: Si00 Status
Bit 1: Si01 Status
Bit 2: Si02 Status
Bit 3: Si03 Status
Bit 4: Si04 Status
Bit 5: Si05 Status
Bit 6: Si06 Status
Bit 7: Si07 Status

| Index (hex) | Subindex (hex) | Object name | Default value | Data range | Unit | Data type | Access | I/O allocation | Data attribute |
|---|---|---|---|---|---|---|---|---|---|
| 7000 | --- | FSoE Master Frame Elements | --- | --- | --- | --- | --- | --- | --- |
| | 00 | Number of Entries | 3 | 3 | --- | USINT | RO | Not supported | --- |
| | 01 | FSoE Master CMD | 00 hex | 00 to FF hex | --- | BYTE | RW | Supported | --- |
| | 02 | FSoE Master Conn_ID | 0000 hex | 0000 to FFFF hex | --- | WORD | RW | Supported | --- |
| | 03 | FSoE Master CRC_0 | 0000 hex | 0000 to FFFF hex | --- | WORD | RW | Supported | --- |

| Index (hex) | Subindex (hex) | Object name | Default value | Data range | Unit | Data type | Access | I/O allocation | Data attribute |
|---|---|---|---|---|---|---|---|---|---|
| 7001 | --- | Safety Output Data | --- | --- | --- | --- | --- | --- | --- |
| | 00 | Number of Entries | 1 | 1 | --- | USINT | RO | Not supported | --- |
| | 01 | Safety Output 1st Word | 0000 hex | 0000 to FFFF hex | --- | WORD | RW | Supported | --- |

| Index (hex) | Subindex (hex) | Object name | Default value | Data range | Unit | Data type | Access | I/O allocation | Data attribute |
|---|---|---|---|---|---|---|---|---|---|
| 7002 | --- | Standard Output Data | --- | --- | --- | --- | --- | --- | --- |
| | 00 | Number of Entries | 2 | 2 | --- | USINT | RO | Not supported | --- |
| | 01 | Standard Output 1st Word | 0000 hex | 0000 hex*1 | --- | WORD | RW | Supported | --- |
| | 02 | Standard Output 2nd Byte | 00 hex | 00 hex*2 | --- | BYTE | RW | Supported | --- |

*1. "Standard Output 1st Word" is reserved by the system.

*2. "Standard Output 2nd Byte" is reserved by the system.

## Other Objects

This section lists other objects.

| Index (hex) | Subindex (hex) | Object name | Default value | Data range | Unit | Data type | Access | I/O allocation | Data attribute |
|---|---|---|---|---|---|---|---|---|---|
| 5000 | --- | Device Safety Address | --- | --- | --- | --- | --- | --- | --- |
| | 00 | Number of Entries | 1 | 1 | --- | USINT | RO | Not supported | --- |
| | 01 | Safety Address | 0000 hex | 0000 to FFFF hex | --- | UINT | RO | Not supported | --- |

# A-2-5 NX-SIH400 Safety Input Unit

## Unit Information Objects

This object gives the product information.

| Index (hex) | Subindex (hex) | Object name | Default value | Data range | Unit | Data type | Access | I/O allocation | Data attribute |
|---|---|---|---|---|---|---|---|---|---|
| 1000 | --- | NX Bus Identity | --- | --- | --- | --- | --- | --- | --- |
| | 00 | Number of Entries | 7 | 7 | --- | USINT | RO | Not supported | --- |
| | 02 | Model | NX-SIH400 | --- | --- | ARRAY [0..11] OF BYTE | RO | Not supported | --- |
| | 03 | Device Type | 00000A02 hex | --- | --- | UDINT | RO | Not supported | --- |
| | 04 | Product Code | 00A20400 hex | --- | --- | UDINT | RO | Not supported | --- |
| | 05 | Vendor Code | 1 | --- | --- | UDINT | RO | Not supported | --- |
| | 06 | Unit Version | *1 | --- | --- | UDINT | RO | Not supported | --- |
| | 07 | Serial Number | *2 | 00000000 to FFFFFFFF hex | --- | UDINT | RO | Not supported | --- |
| 1001 | --- | Production Info | --- | --- | --- | --- | --- | --- | --- |
| | 00 | Number of Entries | 4 | 4 | --- | USINT | RO | Not supported | --- |
| | 01 | Lot Number | *3 | 00000000 to FFFFFFFF hex | --- | UDINT | RO | Not supported | --- |
| | 02 | Hardware Version | *4 | --- | --- | ARRAY [0..19] OF BYTE | RO | Not supported | --- |
| | 03 | Software Version | *5 | --- | --- | ARRAY [0..19] OF BYTE | RO | Not supported | --- |

*1. Bits 24 to 31: Integer part of the Unit version
Bits 16 to 23: Fractional part of the Unit version
Bits 0 to 15: Reserved

(Example) For Ver.1.0, 0100□□□□ hex

*2. The unique serial number of the product is given.
Bits 0 to 31: Serial number

*3. The date of manufacture is given for the "lot number".
Bits 24 to 31: Day of manufacture
Bits 16 to 23: Month of manufacture
Bits 8 to 15: Year of manufacture
Bits 0 to 7: Reserved

*4. The hardware version is given in order in the lowest elements of the array. Unused elements are padded with spaces.

*5. The software version is given in order in the lowest elements of the array. Unused elements are padded with spaces.

## Objects That Accept I/O Allocations

These objects accept I/O allocations.
They cannot be accessed through message communications.

| Index (hex) | Subindex (hex) | Object name | Default value | Data range | Unit | Data type | Access | I/O allocation | Data attribute |
|---|---|---|---|---|---|---|---|---|---|
| 6000 | --- | FSoE Slave Frame Elements | --- | --- | --- | --- | --- | --- | --- |
| | 00 | Number of Entries | 3 | 3 | --- | USINT | RO | Not supported | --- |
| | 01 | FSoE Slave CMD | 00 hex | 00 to FF hex | --- | BYTE | RO | Supported | --- |
| | 02 | FSoE Slave Conn_ID | 0000 hex | 0000 to FFFF hex | --- | WORD | RO | Supported | --- |
| | 03 | FSoE Slave CRC_0 | 0000 hex | 0000 to FFFF hex | --- | WORD | RO | Supported | --- |

| Index (hex) | Subindex (hex) | Object name | Default value | Data range | Unit | Data type | Access | I/O allocation | Data attribute |
|---|---|---|---|---|---|---|---|---|---|
| 6001 | --- | Safety Input Data | --- | --- | --- | --- | --- | --- | --- |
| | 00 | Number of Entries | 1 | 1 | --- | USINT | RO | Not supported | --- |
| | 01 | Safety Input 1st Byte | 00 hex | 00 to FF hex | --- | BYTE | RO | Supported | --- |

| Index (hex) | Subindex (hex) | Object name | Default value | Data range | Unit | Data type | Access | I/O allocation | Data attribute |
|---|---|---|---|---|---|---|---|---|---|
| 6002 | --- | Standard Input Data | --- | --- | --- | --- | --- | --- | --- |
| | 00 | Number of Entries | 2 | 2 | --- | USINT | RO | Not supported | --- |
| | 01 | Standard Input 1st Byte | 00 hex | 00 to FF hex*1 | --- | BYTE | RO | Supported | --- |
| | 02 | Standard Input 2nd Byte | 00 hex | 00 to FF hex*2 | --- | BYTE | RO | Supported | --- |

*1. The details of the "Standard Input 1st Byte" are as follows:
Bit 0: Si00 Logical Value
Bit 1: Si01 Logical Value
Bit 2: Si02 Logical Value
Bit 3: Si03 Logical Value
Bit 4: Safety Connection Status
Bit 5: Safety Input Terminal Status
Bit 6: Unit Normal Status
Bit 7: IO Power Supply Error Flag

*2. The details of the "Standard Input 2nd Byte" are as follows:
Bit 0: Si00 Status
Bit 1: Si01 Status
Bit 2: Si02 Status
Bit 3: Si03 Status
Bits 4 to 7: Reserved

| Index (hex) | Subindex (hex) | Object name | Default value | Data range | Unit | Data type | Access | I/O allocation | Data attribute |
|---|---|---|---|---|---|---|---|---|---|
| 7000 | --- | FSoE Master Frame Elements | --- | --- | --- | --- | --- | --- | --- |
| | 00 | Number of Entries | 3 | 3 | --- | USINT | RO | Not supported | --- |
| | 01 | FSoE Master CMD | 00 hex | 00 to FF hex | --- | BYTE | RW | Supported | --- |
| | 02 | FSoE Master Conn_ID | 0000 hex | 0000 to FFFF hex | --- | WORD | RW | Supported | --- |
| | 03 | FSoE Master CRC_0 | 0000 hex | 0000 to FFFF hex | --- | WORD | RW | Supported | --- |

| Index (hex) | Subindex (hex) | Object name | Default value | Data range | Unit | Data type | Access | I/O allocation | Data attribute |
|---|---|---|---|---|---|---|---|---|---|
| 7001 | --- | Safety Output Data | --- | --- | --- | --- | --- | --- | --- |
| | 00 | Number of Entries | 1 | 1 | --- | USINT | RO | Not supported | --- |
| | 01 | Safety Output 1st Byte | 00 hex | 00 to FF hex | --- | BYTE | RW | Supported | --- |

| Index (hex) | Subindex (hex) | Object name | Default value | Data range | Unit | Data type | Access | I/O allocation | Data attribute |
|---|---|---|---|---|---|---|---|---|---|
| 7002 | --- | Standard Output Data | --- | --- | --- | --- | --- | --- | --- |
| | 00 | Number of Entries | 2 | 2 | --- | USINT | RO | Not supported | --- |
| | 01 | Standard Output 1st Byte | 00 hex | 00 hex *1 | --- | BYTE | RW | Supported | --- |
| | 02 | Standard Output 2nd Byte | 00 hex | 00 hex *2 | --- | BYTE | RW | Supported | --- |

*1. "Standard Output 1st Byte" is reserved by the system.
*2. "Standard Output 2nd Byte" is reserved by the system.

## Other Objects

This section lists other objects.

| Index (hex) | Subindex (hex) | Object name | Default value | Data range | Unit | Data type | Access | I/O allocation | Data attribute |
|---|---|---|---|---|---|---|---|---|---|
| 5000 | --- | Device Safety Address | --- | --- | --- | --- | --- | --- | --- |
| | 00 | Number of Entries | 1 | 1 | --- | USINT | RO | Not supported | --- |
| | 01 | Safety Address | 0000 hex | 0000 to FFFF hex | --- | UINT | RO | Not supported | --- |

# A-2-6 NX-SOD400 Safety Output Unit

## Unit Information Objects

These objects give the product information.

| Index (hex) | Subindex (hex) | Object name | Default value | Data range | Unit | Data type | Access | I/O allocation | Data attribute |
|---|---|---|---|---|---|---|---|---|---|
| 1000 | --- | NX Bus Identity | --- | --- | --- | --- | --- | --- | --- |
| | 00 | Number of Entries | 7 | 7 | --- | USINT | RO | Not supported | --- |
| | 02 | Model | NX-SOD400 | --- | --- | ARRAY [0..11] OF BYTE | RO | Not supported | --- |
| | 03 | Device Type | 00000A03 hex | --- | --- | UDINT | RO | Not supported | --- |
| | 04 | Product Code | 00A30400 hex | --- | --- | UDINT | RO | Not supported | --- |
| | 05 | Vendor Code | 1 | --- | --- | UDINT | RO | Not supported | --- |
| | 06 | Unit Version | *1 | --- | --- | UDINT | RO | Not supported | --- |
| | 07 | Serial Number | *2 | 00000000 to FFFFFFFF hex | --- | UDINT | RO | Not supported | --- |

| Index (hex) | Subindex (hex) | Object name | Default value | Data range | Unit | Data type | Access | I/O allocation | Data attribute |
|---|---|---|---|---|---|---|---|---|---|
| 1001 | --- | Production Info | --- | --- | --- | --- | --- | --- | --- |
| | 00 | Number of Entries | 4 | 4 | --- | USINT | RO | Not supported | --- |
| | 01 | Lot Number | *3 | 00000000 to FFFFFFFF hex | --- | UDINT | RO | Not supported | --- |
| | 02 | Hardware Version | *4 | --- | --- | ARRAY [0..19] OF BYTE | RO | Not supported | --- |
| | 03 | Software Version | *5 | --- | --- | ARRAY [0..19] OF BYTE | RO | Not supported | --- |

*1. Bits 24 to 31: Integer part of the Unit version
Bits 16 to 23: Fractional part of the Unit version
Bits 0 to 15: Reserved
(Example) For Ver.1.0, 0100□□□□ hex

*2. The unique serial number of the product is given.
Bits 0 to 31: Serial number

*3. The date of manufacture is given for the "lot number".
Bits 24 to 31: Day of manufacture
Bits 16 to 23: Month of manufacture
Bits 8 to 15: Year of manufacture
Bits 0 to 7: Reserved

*4. The hardware version is given in order in the lowest elements of the array. Unused elements are padded with spaces.

*5. The software version is given in order in the lowest elements of the array. Unused elements are padded with spaces.

## Objects That Accept I/O Allocations

These objects accept I/O allocations.
They cannot be accessed through message communications.

| Index (hex) | Subindex (hex) | Object name | Default value | Data range | Unit | Data type | Access | I/O allocation | Data attribute |
|---|---|---|---|---|---|---|---|---|---|
| 6000 | --- | FSoE Slave Frame Elements | --- | --- | --- | --- | --- | --- | --- |
| | 00 | Number of Entries | 3 | 3 | --- | USINT | RO | Not supported | --- |
| | 01 | FSoE Slave CMD | 00 hex | 00 to FF hex | --- | BYTE | RO | Supported | --- |
| | 02 | FSoE Slave Conn_ID | 0000 hex | 0000 to FFFF hex | --- | WORD | RO | Supported | --- |
| | 03 | FSoE Slave CRC_0 | 0000 hex | 0000 to FFFF hex | --- | WORD | RO | Supported | --- |

| Index (hex) | Subindex (hex) | Object name | Default value | Data range | Unit | Data type | Access | I/O allocation | Data attribute |
|---|---|---|---|---|---|---|---|---|---|
| 6001 | --- | Safety Input Data | --- | --- | --- | --- | --- | --- | --- |
| | 00 | Number of Entries | 1 | 1 | --- | USINT | RO | Not supported | --- |
| | 01 | Safety Input 1st Byte | 00 hex | 00 to FF hex | --- | BYTE | RO | Supported | --- |

| Index (hex) | Subindex (hex) | Object name | Default value | Data range | Unit | Data type | Access | I/O allocation | Data attribute |
|---|---|---|---|---|---|---|---|---|---|
| 6002 | --- | Standard Input Data | --- | --- | --- | --- | --- | --- | --- |
| | 00 | Number of Entries | 2 | 2 | --- | USINT | RO | Not supported | --- |
| | 01 | Standard Input 1st Byte | 00 hex | 00 to FF hex *1 | --- | BYTE | RO | Supported | --- |
| | 02 | Standard Input 2nd Byte | 00 hex | 00 to FF hex *2 | --- | BYTE | RO | Supported | --- |

*1. The details of the "Standard Input 1st Byte" are as follows:
Bit 0: So00 Monitor Value
Bit 1: So01 Monitor Value
Bit 2: So02 Monitor Value

Bit 3: So03 Monitor Value
Bit 4: Safety Connection Status
Bit 5: Safety Output Terminal Status
Bit 6: Unit Normal Status
Bit 7: IO Power Supply Error Flag

*2. The details of the "Standard Input 2nd Byte" are as follows:
Bit 0: So00 Status
Bit 1: So01 Status
Bit 2: So02 Status
Bit 3: So03 Status
Bits 4 to 7: Reserved

| Index (hex) | Subindex (hex) | Object name | Default value | Data range | Unit | Data type | Access | I/O allocation | Data attribute |
|---|---|---|---|---|---|---|---|---|---|
| 7000 | --- | FSoE Master Frame Elements | --- | --- | --- | --- | --- | --- | --- |
| | 00 | Number of Entries | 3 | 3 | --- | USINT | RO | Not supported | --- |
| | 01 | FSoE Master CMD | 00 hex | 00 to FF hex | --- | BYTE | RW | Supported | --- |
| | 02 | FSoE Master Conn_ID | 0000 hex | 0000 to FFFF hex | --- | WORD | RW | Supported | --- |
| | 03 | FSoE Master CRC_0 | 0000 hex | 0000 to FFFF hex | --- | WORD | RW | Supported | --- |

| Index (hex) | Subindex (hex) | Object name | Default value | Data range | Unit | Data type | Access | I/O allocation | Data attribute |
|---|---|---|---|---|---|---|---|---|---|
| 7001 | --- | Safety Output Data | --- | --- | --- | --- | --- | --- | --- |
| | 00 | Number of Entries | 1 | 1 | --- | USINT | RO | Not supported | --- |
| | 01 | Safety Output 1st Byte | 00 hex | 00 to FF hex | --- | BYTE | RW | Supported | --- |

| Index (hex) | Subindex (hex) | Object name | Default value | Data range | Unit | Data type | Access | I/O allocation | Data attribute |
|---|---|---|---|---|---|---|---|---|---|
| 7002 | --- | Standard Output Data | --- | --- | --- | --- | --- | --- | --- |
| | 00 | Number of Entries | 2 | 2 | --- | USINT | RO | Not supported | --- |
| | 01 | Standard Output 1st Byte | 00 hex | 00 to FF hex | --- | BYTE | RW | Supported | --- |
| | 02 | Standard Output 2nd Byte | 00 hex | 00 to FF hex | --- | BYTE | RW | Supported | --- |

## Other Objects

This section lists other objects.

| Index (hex) | Subindex (hex) | Object name | Default value | Data range | Unit | Data type | Access | I/O allocation | Data attribute |
|---|---|---|---|---|---|---|---|---|---|
| 5000 | --- | Device Safety Address | --- | --- | --- | --- | --- | --- | --- |
| | 00 | Number of Entries | 1 | 1 | --- | USINT | RO | Not supported | --- |
| | 01 | Safety Address | 0000 hex | 0000 to FFFF hex | --- | UINT | RO | Not supported | --- |

# A-2-7 NX-SOH200 Safety Output Unit

## Unit Information Objects

These objects give the product information.

| Index (hex) | Subindex (hex) | Object name | Default value | Data range | Unit | Data type | Access | I/O allocation | Data attribute |
|---|---|---|---|---|---|---|---|---|---|
| 1000 | --- | NX Bus Identity | --- | --- | --- | --- | --- | --- | --- |
| | 00 | Number of Entries | 7 | 7 | --- | USINT | RO | Not supported | --- |
| | 02 | Model | NX-SOH200 | --- | --- | ARRAY [0..11] OF BYTE | RO | Not supported | --- |
| | 03 | Device Type | 00000A04 hex | --- | --- | UDINT | RO | Not supported | --- |
| | 04 | Product Code | 00A40200 hex | --- | --- | UDINT | RO | Not supported | --- |
| | 05 | Vendor Code | 1 | --- | --- | UDINT | RO | Not supported | --- |
| | 06 | Unit Version | *1 | --- | --- | UDINT | RO | Not supported | --- |
| | 07 | Serial Number | *2 | 00000000 to FFFFFFFF hex | --- | UDINT | RO | Not supported | --- |
| 1001 | --- | Production Info | --- | --- | --- | --- | --- | --- | --- |
| | 00 | Number of Entries | 4 | 4 | --- | USINT | RO | Not supported | --- |
| | 01 | Lot Number | *3 | 00000000 to FFFFFFFF hex | --- | UDINT | RO | Not supported | --- |
| | 02 | Hardware Version | *4 | --- | --- | ARRAY [0..19] OF BYTE | RO | Not supported | --- |
| | 03 | Software Version | *5 | --- | --- | ARRAY [0..19] OF BYTE | RO | Not supported | --- |

*1. Bits 24 to 31: Integer part of the Unit version
Bits 16 to 23: Fractional part of the Unit version
Bits 0 to 15: Reserved

(Example) For Ver.1.0, 0100□□□□ hex

*2. The unique serial number of the product is given.
Bits 0 to 31: Serial number

*3. The date of manufacture is given for the "lot number".
Bits 24 to 31: Day of manufacture
Bits 16 to 23: Month of manufacture
Bits 8 to 15: Year of manufacture
Bits 0 to 7: Reserved

*4. The hardware version is given in order in the lowest elements of the array. Unused elements are padded with spaces.

*5. The software version is given in order in the lowest elements of the array. Unused elements are padded with spaces.

## Objects That Accept I/O Allocations

These objects accept I/O allocations.
They cannot be accessed through message communications.

| Index (hex) | Subindex (hex) | Object name | Default value | Data range | Unit | Data type | Access | I/O allocation | Data attribute |
|---|---|---|---|---|---|---|---|---|---|
| 6000 | --- | FSoE Slave Frame Elements | --- | --- | --- | --- | --- | --- | --- |
| | 00 | Number of Entries | 3 | 3 | --- | USINT | RO | Not supported | --- |
| | 01 | FSoE Slave CMD | 00 hex | 00 to FF hex | --- | BYTE | RO | Supported | --- |
| | 02 | FSoE Slave Conn_ID | 0000 hex | 0000 to FFFF hex | --- | WORD | RO | Supported | --- |
| | 03 | FSoE Slave CRC_0 | 0000 hex | 0000 to FFFF hex | --- | WORD | RO | Supported | --- |

| Index (hex) | Subindex (hex) | Object name | Default value | Data range | Unit | Data type | Access | I/O allocation | Data attribute |
|---|---|---|---|---|---|---|---|---|---|
| 6001 | --- | Safety Input Data | --- | --- | --- | --- | --- | --- | --- |
| | 00 | Number of Entries | 1 | 1 | --- | USINT | RO | Not supported | --- |
| | 01 | Safety Input 1st Byte | 00 hex | 00 to FF hex | --- | BYTE | RO | Supported | --- |

| Index (hex) | Subindex (hex) | Object name | Default value | Data range | Unit | Data type | Access | I/O allocation | Data attribute |
|---|---|---|---|---|---|---|---|---|---|
| 6002 | --- | Standard Input Data | --- | --- | --- | --- | --- | --- | --- |
| | 00 | Number of Entries | 2 | 2 | --- | USINT | RO | Not supported | --- |
| | 01 | Standard Input 1st Byte | 00 hex | 00 to FF hex *1 | --- | BYTE | RO | Supported | --- |
| | 02 | Standard Input 2nd Byte | 00 hex | 00 to FF hex *2 | --- | BYTE | RO | Supported | --- |

*1. The details of the "Standard Input 1st Byte" are as follows:
Bit 0: So00 Monitor Value
Bit 1: So01 Monitor Value
Bit 2: Safety Connection Status
Bit 3: Safety Output Terminal Status
Bit 4: Unit Normal Status
Bit 5: IO Power Supply Error Flag
Bits 6 to 7: Reserved

*2. The details of the "Standard Input 2nd Byte" are as follows:
Bit 0: So00 Status
Bit 1: So01 Status
Bits 2 to 7: Reserved

| Index (hex) | Subindex (hex) | Object name | Default value | Data range | Unit | Data type | Access | I/O allocation | Data attribute |
|---|---|---|---|---|---|---|---|---|---|
| 7000 | --- | FSoE Master Frame Elements | --- | --- | --- | --- | --- | --- | --- |
| | 00 | Number of Entries | 3 | 3 | --- | USINT | RO | Not supported | --- |
| | 01 | FSoE Master CMD | 00 hex | 00 to FF hex | --- | BYTE | RW | Supported | --- |
| | 02 | FSoE Master Conn_ID | 0000 hex | 0000 to FFFF hex | --- | WORD | RW | Supported | --- |
| | 03 | FSoE Master CRC_0 | 0000 hex | 0000 to FFFF hex | --- | WORD | RW | Supported | --- |

| Index (hex) | Subindex (hex) | Object name | Default value | Data range | Unit | Data type | Access | I/O allocation | Data attribute |
|---|---|---|---|---|---|---|---|---|---|
| 7001 | --- | Safety Output Data | --- | --- | --- | --- | --- | --- | --- |
| | 00 | Number of Entries | 1 | 1 | --- | USINT | RO | Not supported | --- |
| | 01 | Safety Output 1st Byte | 00 hex | 00 to FF hex | --- | BYTE | RW | Supported | --- |

| Index (hex) | Subindex (hex) | Object name | Default value | Data range | Unit | Data type | Access | I/O allocation | Data attribute |
|---|---|---|---|---|---|---|---|---|---|
| 7002 | --- | Standard Output Data | --- | --- | --- | --- | --- | --- | --- |
| | 00 | Number of Entries | 2 | 2 | --- | USINT | RO | Not supported | --- |
| | 01 | Standard Output 1st Byte | 00 hex | 00 to FF hex | --- | BYTE | RW | Supported | --- |
| | 02 | Standard Output 2nd Byte | 00 hex | 00 to FF hex | --- | BYTE | RW | Supported | --- |

## Other Objects

This section lists other objects.

| Index (hex) | Subindex (hex) | Object name | Default value | Data range | Unit | Data type | Access | I/O allocation | Data attribute |
|---|---|---|---|---|---|---|---|---|---|
| 5000 | --- | Device Safety Address | --- | --- | --- | --- | --- | --- | --- |
| | 00 | Number of Entries | 1 | 1 | --- | USINT | RO | Not supported | --- |
| | 01 | Safety Address | 0000 hex | 0000 to FFFF hex | --- | UINT | RO | Not supported | --- |

# A-3 Application Examples

These examples show safety systems that use Safety Control Unit.
Refer to the *NX-series Safety Control Unit Instructions Reference Manual (Cat. No. Z931) for details on the instructions that are used in each example.*

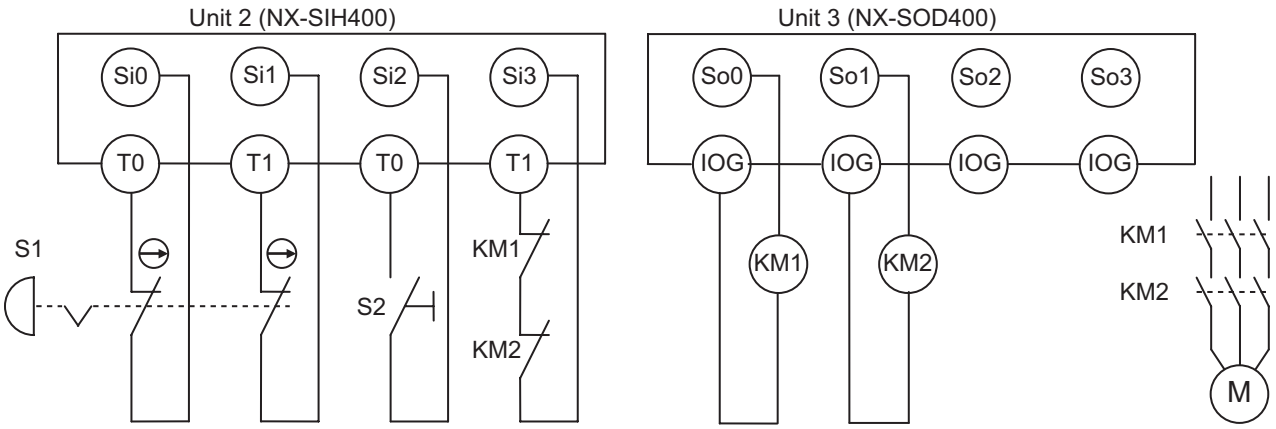
## A-3-1 Emergency Stop Pushbutton Switches

### Application Overview

| Safety category/PL | Safety device | Stop category | Reset |
|---|---|---|---|
| Equivalent to 4/PLe | Emergency stop pushbutton | 0 | Manual |

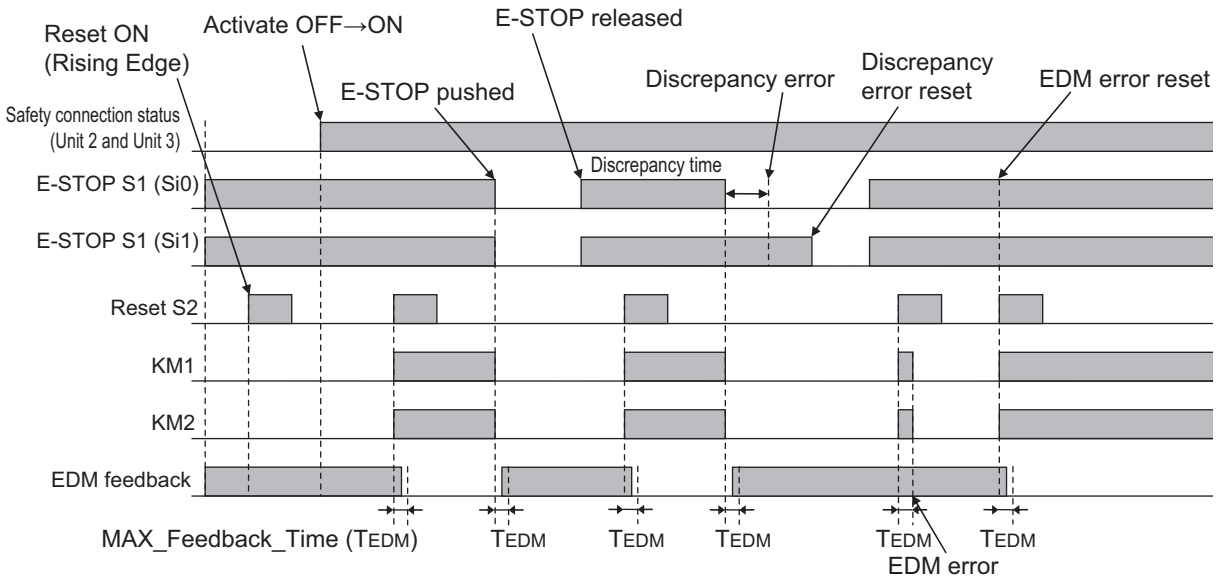
Motor M stops when emergency stop pushbutton S1 is pressed.

### Wiring




S1: Emergency stop pushbutton
S2: Reset switch
KM1, KM2: Contactors
M: Motor

## Timing Chart




## Safety I/O Terminal & I/O Map Setting

### ● Safety I/O Terminal Settings

**Node1/Unit2 : NX-SIH400 (N2 : Instance0)**

| External Device | Channel | Discrepancy | On-Off | Off-On | Test Source | Comment |
|---|---|---|---|---|---|---|
| Mechanical Contact for Dual Channel Equivalent | Si 0 | 500ms | 0ms | 0ms | T0 | Emergency Stop Pushbutton Switch(2NC) |
| | Si 1 | 500ms | 0ms | 0ms | T1 | |
| Mechanical Contact For Single Channel | Si 2 | 0ms | 0ms | 0ms | T0 | Reset Switch |
| Mechanical Contact For Single Channel | Si 3 | 0ms | 0ms | 0ms | T1 | EDM(Contact Welding Detection) |

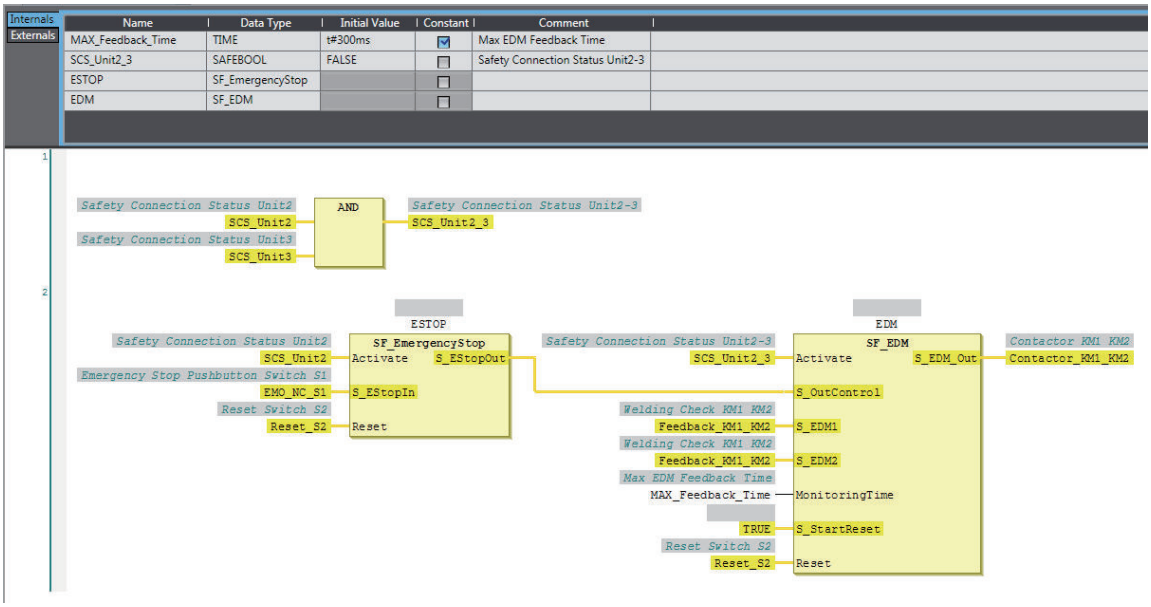
**Node1/Unit3 : NX-SOD400 (N3 : Instance1)**

| External Device | Channel | Comment |
|---|---|---|
| Dual Output with Test Pulse | So 0 | 2 Safety Relays w/ Welding Check |
| | So 1 | |
| | So 2 | |
| | So 3 | |

### ● I/O Map Settings

| Position | Port | R/W | Data Type | Variable | Variable Comment | Variable Type |
|---|---|---|---|---|---|---|
| | EtherCAT Network | | | | | |
| EtherCAT Master | Master | | | | | |
| Node1/Unit2 | NX-SIH400 | | | | | |
| | Safety Inputs and Status | | | | | |
| | Si00 Logical Value | R | SAFEBOOL | EMO_NC_S1 | Emergency Stop Pushbutton Switch S1 | Global Variables |
| | Si01 Logical Value | R | SAFEBOOL | | | |
| | Si02 Logical Value | R | SAFEBOOL | Reset_S2 | Reset Switch S2 | Global Variables |
| | Si03 Logical Value | R | SAFEBOOL | Feedback_KM1_KM2 | Welding Check KM1_KM2 | Global Variables |
| | Safety Connection Status | R | SAFEBOOL | SCS_Unit2 | Safety Connection Status Unit2 | Global Variables |
| | Safety Input Terminal Status | R | SAFEBOOL | | | |
| Node1/Unit3 | NX-SOD400 | | | | | |
| | Status | | | | | |
| | Safety Connection Status | R | SAFEBOOL | SCS_Unit3 | Safety Connection Status Unit3 | Global Variables |
| | Safety Output Terminal Status | R | SAFEBOOL | | | |
| | Safety Outputs | | | | | |
| | So00 Output Value | W | SAFEBOOL | Contactor_KM1_KM2 | Contactor KM1_KM2 | Global Variables |
| | So01 Output Value | W | SAFEBOOL | | | |
| | So02 Output Value | W | SAFEBOOL | | | |
| | So03 Output Value | W | SAFEBOOL | | | |

## Program




**Precautions for Safe Use**

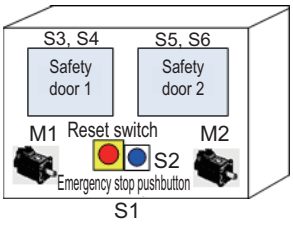
- Test the functionality every six months to detect welded contactor contacts.
- The customer is responsible for attaining conformance of the entire system to standards.
- To detect electrical and mechanical failures, use a combination of redundant semiconductor output contacts and redundant mechanical output devices.

# A-3-2 Safety Doors

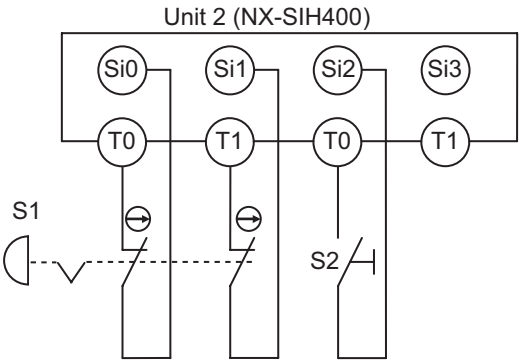
## Application Overview

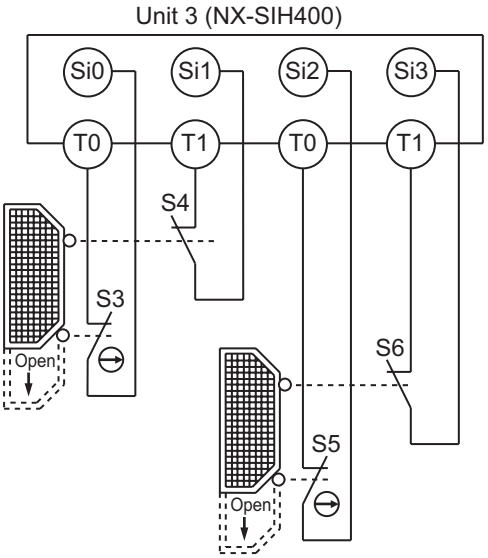
| Safety category/PL | Safety device | Stop category | Reset |
|---|---|---|---|
| Equivalent to 4/PLe (Safety Door) | Safety limit switches 1 and 2 | 0 | Auto |
| | Emergency stop pushbutton | 0 | Manual |

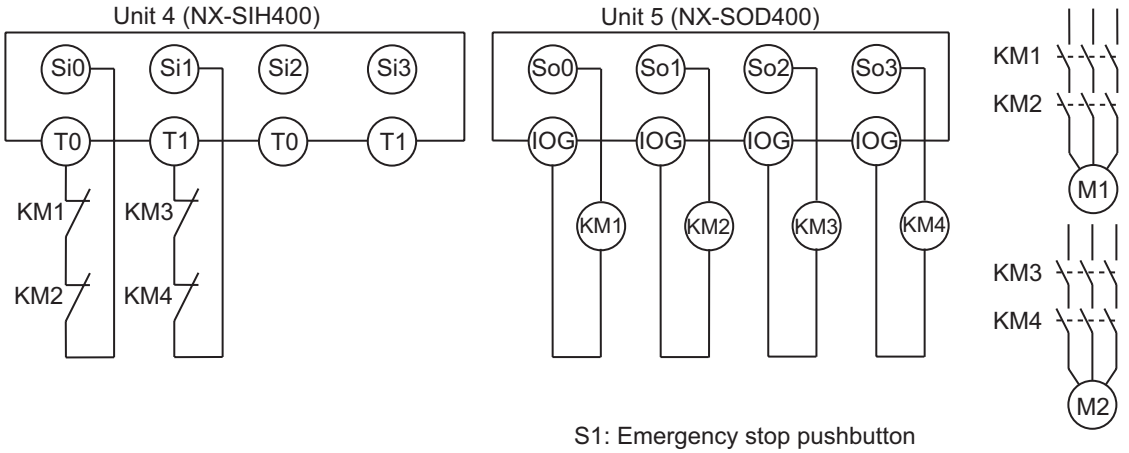
M1 stops when safety door 1 (S3, S4) is opened.
M2 stops when safety door 2 (S5, S6) is opened.
Both M1 and M2 stop when the emergency stop pushbutton S1 is pressed.




## Wiring

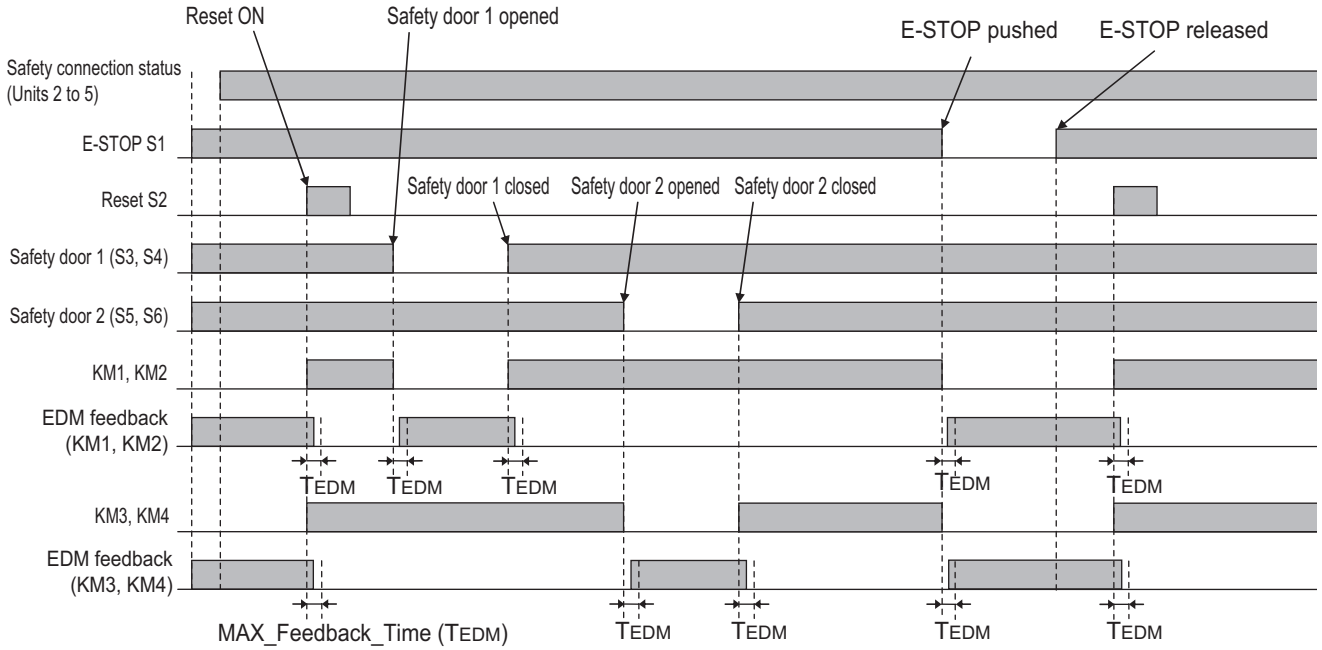





S1: Emergency stop pushbutton
S2: Reset switch
S3, S5: Safety limit switches
S4, S6: Limit switches (NO contacts)
KM1, KM2, KM3, KM4: Contactors
M1, M2: Motors

## Timing Chart




## Safety I/O Terminal & I/O Map Setting

## ● Safety I/O Terminal Settings

**Node1/Unit2 : NX-SIH400 (N2 : Instance0)**

| External Device | Channel | Discrepancy | On-Off | Off-On | Test Source | Comment |
|---|---|---|---|---|---|---|
| Mechanical Contact for Dual Channel Equivalent | Si 0 | 500ms | 0ms | 0ms | T0 | Emergency Stop Pushbutton Switch(2NC) |
| | Si 1 | 500ms | 0ms | 0ms | T1 | |
| Mechanical Contact For Single Channel | Si 2 | 0ms | 0ms | 0ms | T0 | Reset Switch |
| | Si 3 | | | | | |

**Node1/Unit3 : NX-SIH400 (N3 : Instance1)**

| External Device | Channel | Discrepancy | On-Off | Off-On | Test Source | Comment |
|---|---|---|---|---|---|---|
| Mechanical Contact For Single Channel | Si 0 | 0ms | 0ms | 0ms | T0 | Safety Limit Switch(NC) |
| Mechanical Contact For Single Channel | Si 1 | 0ms | 0ms | 0ms | T1 | Limit Switch(NO) |
| Mechanical Contact For Single Channel | Si 2 | 0ms | 0ms | 0ms | T0 | Safety Limit Switch(NC) |
| Mechanical Contact For Single Channel | Si 3 | 0ms | 0ms | 0ms | T1 | Limit Switch(NO) |

**Node1/Unit4 : NX-SIH400 (N4 : Instance2)**

| External Device | Channel | Discrepancy | On-Off | Off-On | Test Source | Comment |
|---|---|---|---|---|---|---|
| Mechanical Contact For Single Channel | Si 0 | 0ms | 0ms | 0ms | T0 | EDM(Contact Welding Detection) |
| Mechanical Contact For Single Channel | Si 1 | 0ms | 0ms | 0ms | T1 | EDM(Contact Welding Detection) |
| | Si 2 | | | | | |
| | Si 3 | | | | | |

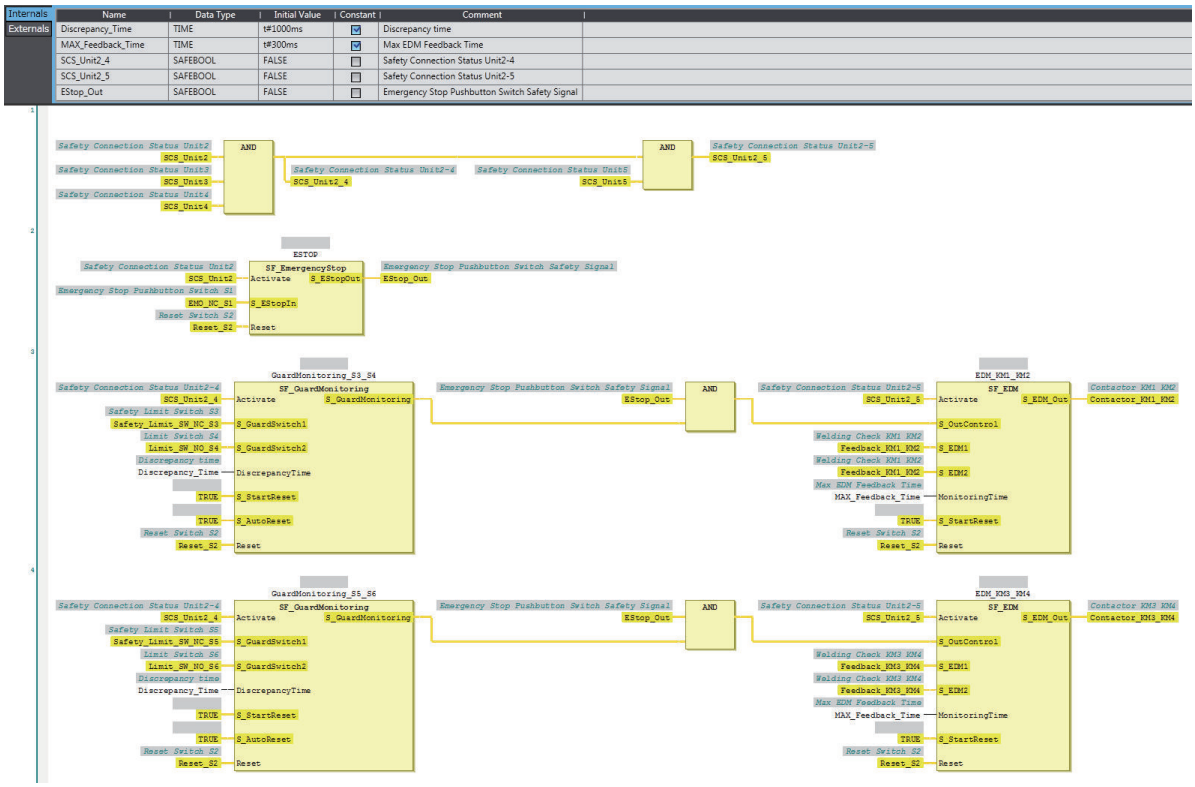
**Node1/Unit5 : NX-SOD400 (N5 : Instance3)**

| External Device | Channel | Comment |
|---|---|---|
| Dual Output with Test Pulse | So 0 | 2 Safety Relays w/ Welding Check |
| | So 1 | |
| Dual Output with Test Pulse | So 2 | 2 Safety Relays w/ Welding Check |
| | So 3 | |

## ● I/O Map Settings

| Position | Port | R/W | Data Type | Variable | Variable Comment | Variable Type |
|---|---|---|---|---|---|---|
| | ▼ EtherCAT Network | | | | | |
| EtherCAT Master | Master | | | | | |
| Node1/Unit2 | ▼ NX-SIH400 | | | | | |
| | ▼ Safety Inputs and Status | | | | | |
| | Si00 Logical Value | R | SAFEBOOL | EMO_NC_S1 | Emergency Stop Pushbutton Switch(2NC) | Global Variables |
| | Si01 Logical Value | R | SAFEBOOL | | | |
| | Si02 Logical Value | R | SAFEBOOL | Reset_S2 | Reset Switch | Global Variables |
| | Si03 Logical Value | R | SAFEBOOL | | | |
| | Safety Connection Status | R | SAFEBOOL | SCS_Unit2 | Safety Connection Status Unit2 | Global Variables |
| | Safety Input Terminal Status | R | SAFEBOOL | | | |
| Node1/Unit3 | ▼ NX-SIH400 | | | | | |
| | ▼ Safety Inputs and Status | | | | | |
| | Si00 Logical Value | R | SAFEBOOL | Safety_Limit_SW_NC_S3 | Safety Limit Switch(NC) | Global Variables |
| | Si01 Logical Value | R | SAFEBOOL | Limit_SW_NO_S4 | Limit Switch(NO) | Global Variables |
| | Si02 Logical Value | R | SAFEBOOL | Safety_Limit_SW_NC_S5 | Safety Limit Switch(NC) | Global Variables |
| | Si03 Logical Value | R | SAFEBOOL | Limit_SW_NO_S6 | Limit Switch(NO) | Global Variables |
| | Safety Connection Status | R | SAFEBOOL | SCS_Unit3 | Safety Connection Status Unit3 | Global Variables |
| | Safety Input Terminal Status | R | SAFEBOOL | | | |
| Node1/Unit4 | ▼ NX-SIH400 | | | | | |
| | ▼ Safety Inputs and Status | | | | | |
| | Si00 Logical Value | R | SAFEBOOL | Feedback_KM1_KM2 | EDM(Contact Welding Detection) | Global Variables |
| | Si01 Logical Value | R | SAFEBOOL | Feedback_KM3_KM4 | EDM(Contact Welding Detection) | Global Variables |
| | Si02 Logical Value | R | SAFEBOOL | | | |
| | Si03 Logical Value | R | SAFEBOOL | | | |
| | Safety Connection Status | R | SAFEBOOL | SCS_Unit4 | Safety Connection Status Unit4 | Global Variables |
| | Safety Input Terminal Status | R | SAFEBOOL | | | |
| Node1/Unit5 | ▼ NX-SOD400 | | | | | |
| | ▼ Status | | | | | |
| | Safety Connection Status | R | SAFEBOOL | SCS_Unit5 | Safety Connection Status Unit5 | Global Variables |
| | Safety Output Terminal Status | R | SAFEBOOL | | | |
| | ▼ Safety Outputs | | | | | |
| | So00 Output Value | W | SAFEBOOL | Contactor_KM1_KM2 | 2 Safety Relays w/ Welding Check | Global Variables |
| | So01 Output Value | W | SAFEBOOL | | | |
| | So02 Output Value | W | SAFEBOOL | Contactor_KM3_KM4 | 2 Safety Relays w/ Welding Check | Global Variables |
| | So03 Output Value | W | SAFEBOOL | | | |

## Program




**Precautions for Safe Use**

- Test the functionality every six months to detect welded contactor contacts.
- The customer is responsible for attaining conformance of the entire system to standards.
- To detect electrical and mechanical failures, use a combination of redundant semiconductor output contacts and redundant mechanical output devices.
- Refer to ISO 14119:2013 for additional measures to minimize the possibility of interlocking devices associated with guards from being disabled.

# A-3-3 Safety Laser Scanners

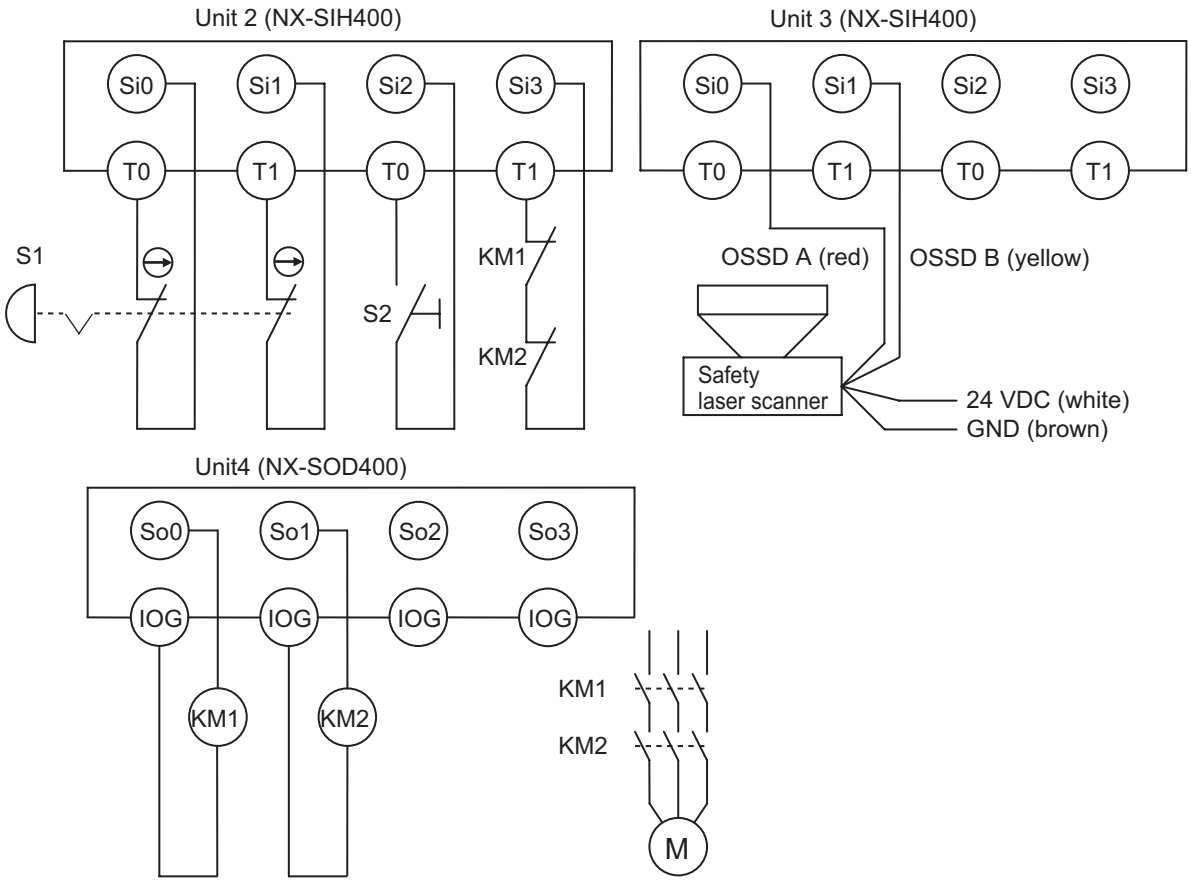
## Application Overview

| Safety category/PL | Safety device | Stop category | Reset |
|---|---|---|---|
| Equivalent to 3/PLd (Laser Scanner) | Laser scanner | 0 | Auto |
| | Emergency stop pushbutton | 0 | Manual |

AGV stops when emergency stop pushbutton S1 is pressed.

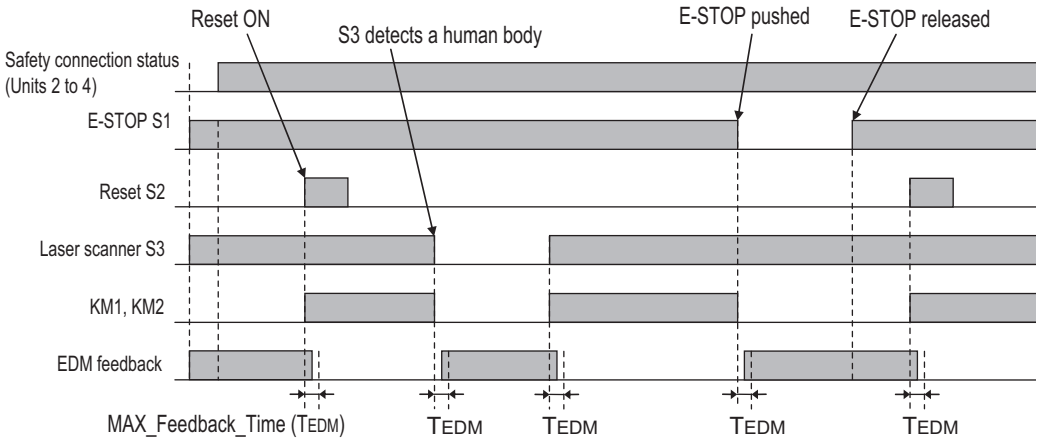
AGV stops when laser scanner S3 detects that persons or objects approach into the safety zone.

## Wiring




S1: Emergency stop pushbutton
S2: Reset switch
S3: Safety laser scanner
KM1, KM2: Contactors
M: Motor

## Timing Chart




## Safety I/O Terminal & I/O Map Setting

## ● Safety I/O Terminal Settings

**Node1/Unit2 : NX-SIH400 (N2 : Instance0)**

| External Device | Channel | Discrepancy | On-Off | Off-On | Test Source | Comment |
|---|---|---|---|---|---|---|
| Mechanical Contact for Dual Channel Equivalent | Si 0 | 500ms | 0ms | 0ms | T0 | Emergency Stop Pushbutton Switch(2NC) |
| | Si 1 | 500ms | 0ms | 0ms | T1 | |
| Mechanical Contact For Single Channel | Si 2 | 0ms | 0ms | 0ms | T0 | Reset Switch |
| Mechanical Contact For Single Channel | Si 3 | 0ms | 0ms | 0ms | T1 | EDM(Contact Welding Detection) |

**Node1/Unit3 : NX-SIH400 (N3 : Instance1)**

| External Device | Channel | Discrepancy | On-Off | Off-On | Test Source | Comment |
|---|---|---|---|---|---|---|
| Semiconductor Output for Dual Channel Equivalent | Si 0 | 500ms | 0ms | 0ms | Not Used | Dual Safety Semiconductor Output(Equivalent) |
| | Si 1 | 500ms | 0ms | 0ms | Not Used | |
| | Si 2 | | | | | |
| | Si 3 | | | | | |

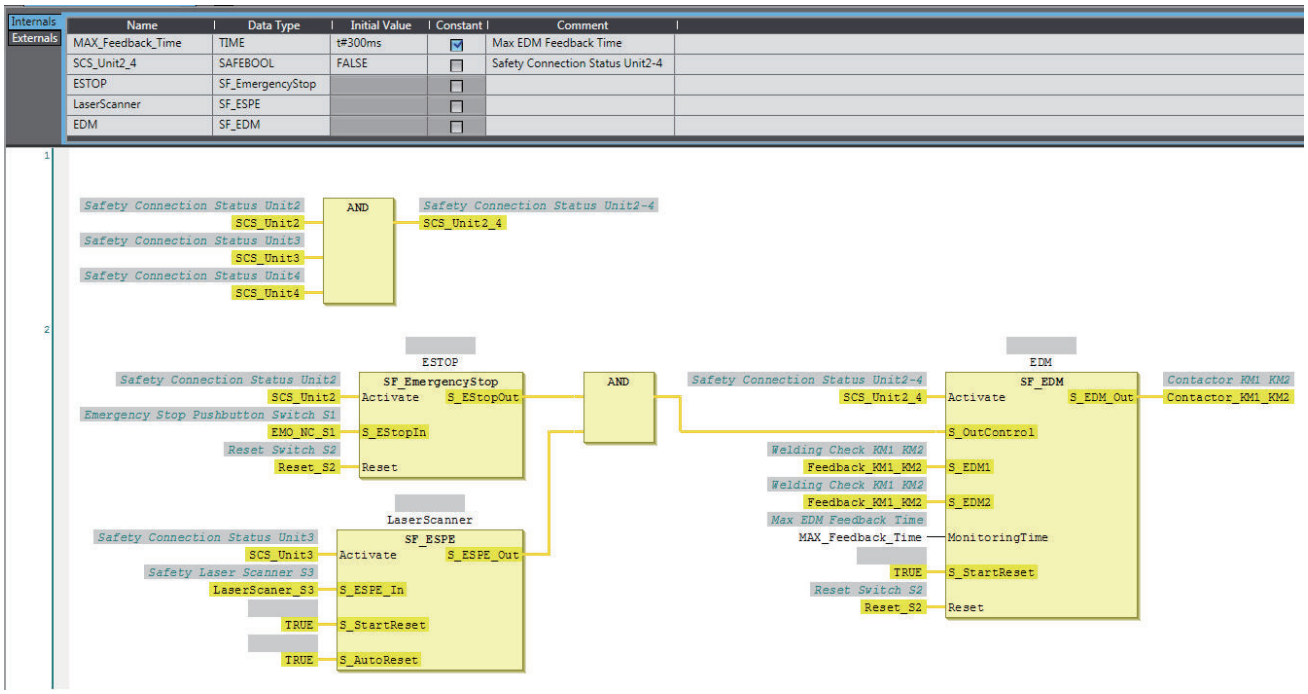
**Node1/Unit4 : NX-SOD400 (N4 : Instance2)**

| External Device | Channel | Comment |
|---|---|---|
| Dual Output with Test Pulse | So 0 | 2 Safety Relays w/ Welding Check |
| | So 1 | |
| | So 2 | |
| | So 3 | |

## ● I/O Map Settings

| Position | Port | R/W | Data Type | Variable | Variable Comment | Variable Type |
|---|---|---|---|---|---|---|
| | ▼ EtherCAT Network | | | | | |
| EtherCAT Master | Master | | | | | |
| Node1/Unit2 | ▼ NX-SIH400 | | | | | |
| | ▼ Safety Inputs and Status | | | | | |
| | Si00 Logical Value | R | SAFEBOOL | EMO_NC_S1 | Emergency Stop Pushbutton Switch S1 | Global Variables |
| | Si01 Logical Value | R | SAFEBOOL | | | |
| | Si02 Logical Value | R | SAFEBOOL | Reset_S2 | Reset Switch S2 | Global Variables |
| | Si03 Logical Value | R | SAFEBOOL | Feedback_KM1_KM2 | Welding Check KM1_KM2 | Global Variables |
| | Safety Connection Status | R | SAFEBOOL | SCS_Unit2 | Safety Connection Status Unit2 | Global Variables |
| | Safety Input Terminal Status | R | SAFEBOOL | | | |
| Node1/Unit3 | ▼ NX-SIH400 | | | | | |
| | ▼ Safety Inputs and Status | | | | | |
| | Si00 Logical Value | R | SAFEBOOL | LaserScaner_S3 | Safety Laser Scanner S3 | Global Variables |
| | Si01 Logical Value | R | SAFEBOOL | | | |
| | Si02 Logical Value | R | SAFEBOOL | | | |
| | Si03 Logical Value | R | SAFEBOOL | | | |
| | Safety Connection Status | R | SAFEBOOL | SCS_Unit3 | Safety Connection Status Unit3 | Global Variables |
| | Safety Input Terminal Status | R | SAFEBOOL | | | |
| Node1/Unit4 | ▼ NX-SOD400 | | | | | |
| | ▼ Status | | | | | |
| | Safety Connection Status | R | SAFEBOOL | SCS_Unit4 | Safety Connection Status Unit4 | Global Variables |
| | Safety Output Terminal Status | R | SAFEBOOL | | | |
| | ▼ Safety Outputs | | | | | |
| | So00 Output Value | W | SAFEBOOL | Contactor_KM1_KM2 | Contactor KM1_KM2 | Global Variables |
| | So01 Output Value | W | SAFEBOOL | | | |
| | So02 Output Value | W | SAFEBOOL | | | |
| | So03 Output Value | W | SAFEBOOL | | | |

## Program




### Precautions for Safe Use

- Test the functionality every six months to detect welded contactor contacts.
- The customer is responsible for attaining conformance of the entire system to standards.
- To detect electrical and mechanical failures, use a combination of redundant semiconductor output contacts and redundant mechanical output devices.

# A-3-4 Safety Door Switches with Magnetic Locks and Key Selector Switches

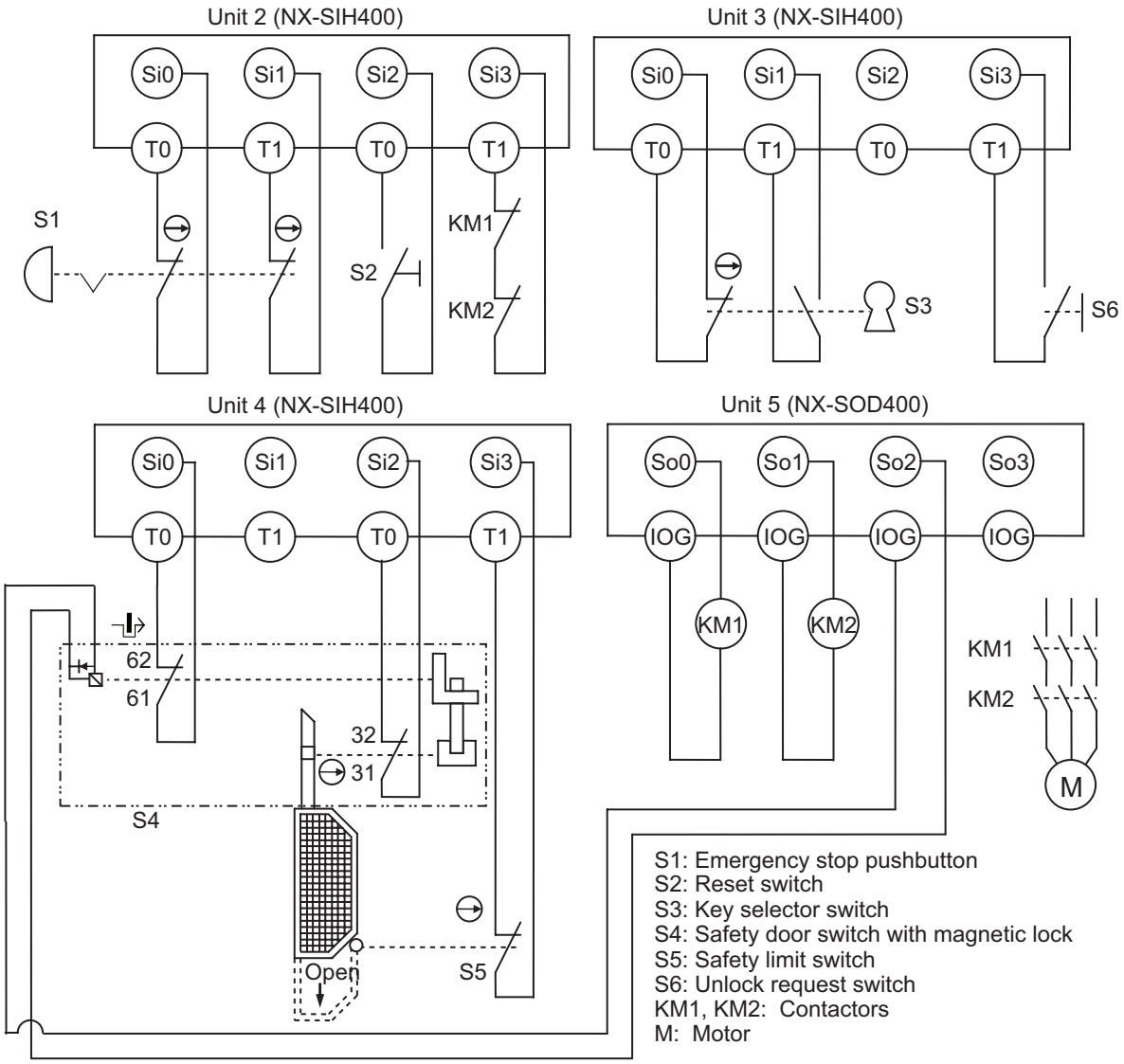
## Application Overview

| Safety category/PL | Safety device | Stop category | Reset |
|---|---|---|---|
| Equivalent to 4/PLe (Safety Door Switches with Magnetic Locks) | • Emergency stop pushbutton<br>• Safety door switch with magnetic lock (mechanical lock type)<br>• Key selector switch | 0 | Manual |

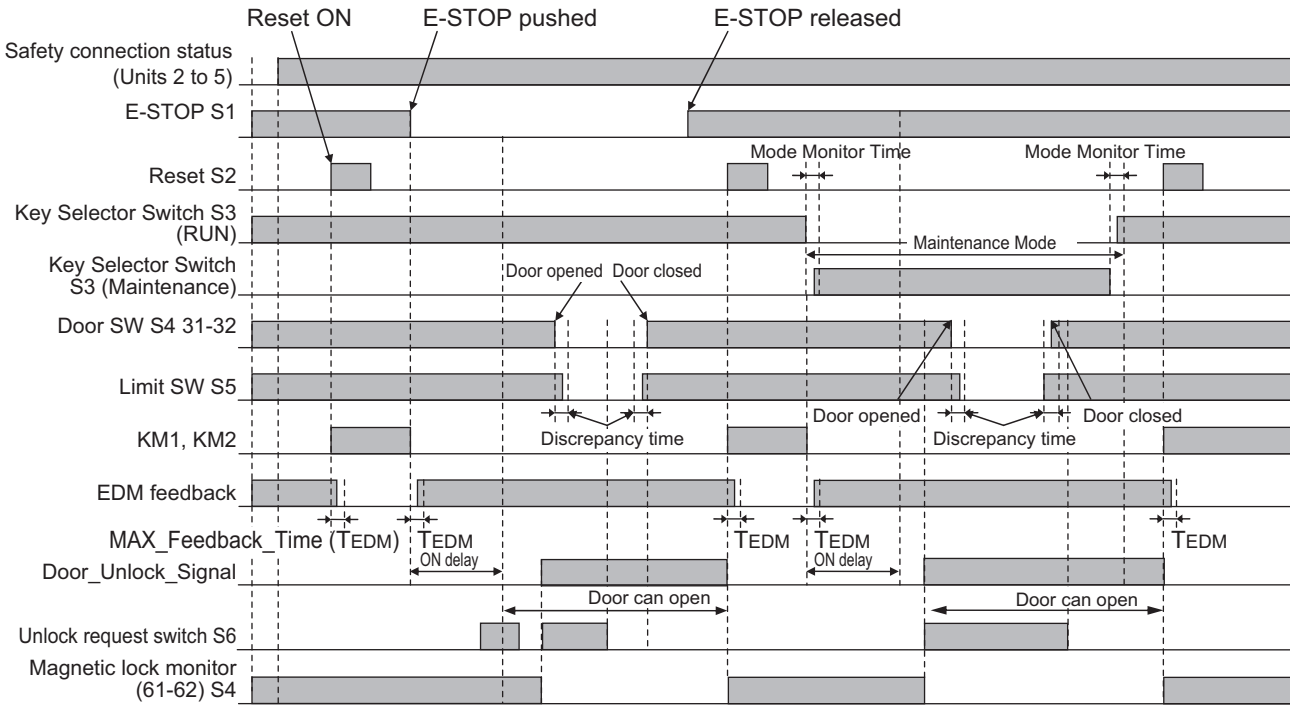
Safety doors S4 and S5 cannot be opened while the user mode is set for normal operation. The outputs are turned OFF by changing to maintenance mode and the safety doors can be opened 5 seconds later.

The outputs also turn OFF when emergency pushbutton S1 is pressed.

## Wiring


Unit 2 (NX-SIH400)
Si0
Si1
Si2
Si3
T0
T1
T0
T1
S1
KM1
S2
KM2
Unit 3 (NX-SIH400)
Si0
Si1
Si2
Si3
T0
T1
T0
T1
S3
S6
Unit 4 (NX-SIH400)
Si0
Si1
Si2
Si3
T0
T1
T0
T1
62
61
32
31
S4
Open
S5
Unit 5 (NX-SOD400)
So0
So1
So2
So3
IOG
IOG
IOG
IOG
KM1
KM2
KM1
KM2
M
S1: Emergency stop pushbutton
S2: Reset switch
S3: Key selector switch
S4: Safety door switch with magnetic lock
S5: Safety limit switch
S6: Unlock request switch
KM1, KM2: Contactors
M: Motor

## Timing Chart




## Safety I/O Terminal & I/O Map Setting

### ● Safety I/O Terminal Settings

**Node1/Unit2 : NX-SIH400 (N2 : Instance0)**

| External Device | Channel | Discrepancy | On-Off | Off-On | Test Source | Comment |
|---|---|---|---|---|---|---|
| Mechanical Contact for Dual Channel Equivalent | Si 0 | 500ms | 0ms | 0ms | T0 | Emergency Stop Pushbutton Switch(2NC) |
| | Si 1 | 500ms | 0ms | 0ms | T1 | |
| Mechanical Contact For Single Channel | Si 2 | 0ms | 0ms | 0ms | T0 | Reset Switch |
| Mechanical Contact For Single Channel | Si 3 | 0ms | 0ms | 0ms | T1 | EDM(Contact Welding Detection) |

**Node1/Unit3 : NX-SIH400 (N3 : Instance1)**

| External Device | Channel | Discrepancy | On-Off | Off-On | Test Source | Comment |
|---|---|---|---|---|---|---|
| Mechanical Contact For Single Channel | Si 0 | 0ms | 0ms | 0ms | T0 | Single Contact |
| Mechanical Contact For Single Channel | Si 1 | 0ms | 0ms | 0ms | T1 | Single Contact |
| | Si 2 | | | | | |
| Mechanical Contact For Single Channel | Si 3 | 0ms | 0ms | 0ms | T1 | Single Contact |

**Node1/Unit4 : NX-SIH400 (N4 : Instance2)**

| External Device | Channel | Discrepancy | On-Off | Off-On | Test Source | Comment |
|---|---|---|---|---|---|---|
| Mechanical Contact For Single Channel | Si 0 | 0ms | 0ms | 0ms | T0 | Safety Switch(NC) |
| | Si 1 | | | | | |
| Mechanical Contact For Single Channel | Si 2 | 0ms | 0ms | 0ms | T0 | Safety Switch(NC) |
| Mechanical Contact For Single Channel | Si 3 | 0ms | 0ms | 0ms | T1 | Safety Limit Switch(NC) |

**Node1/Unit5 : NX-SOD400 (N5 : Instance3)**

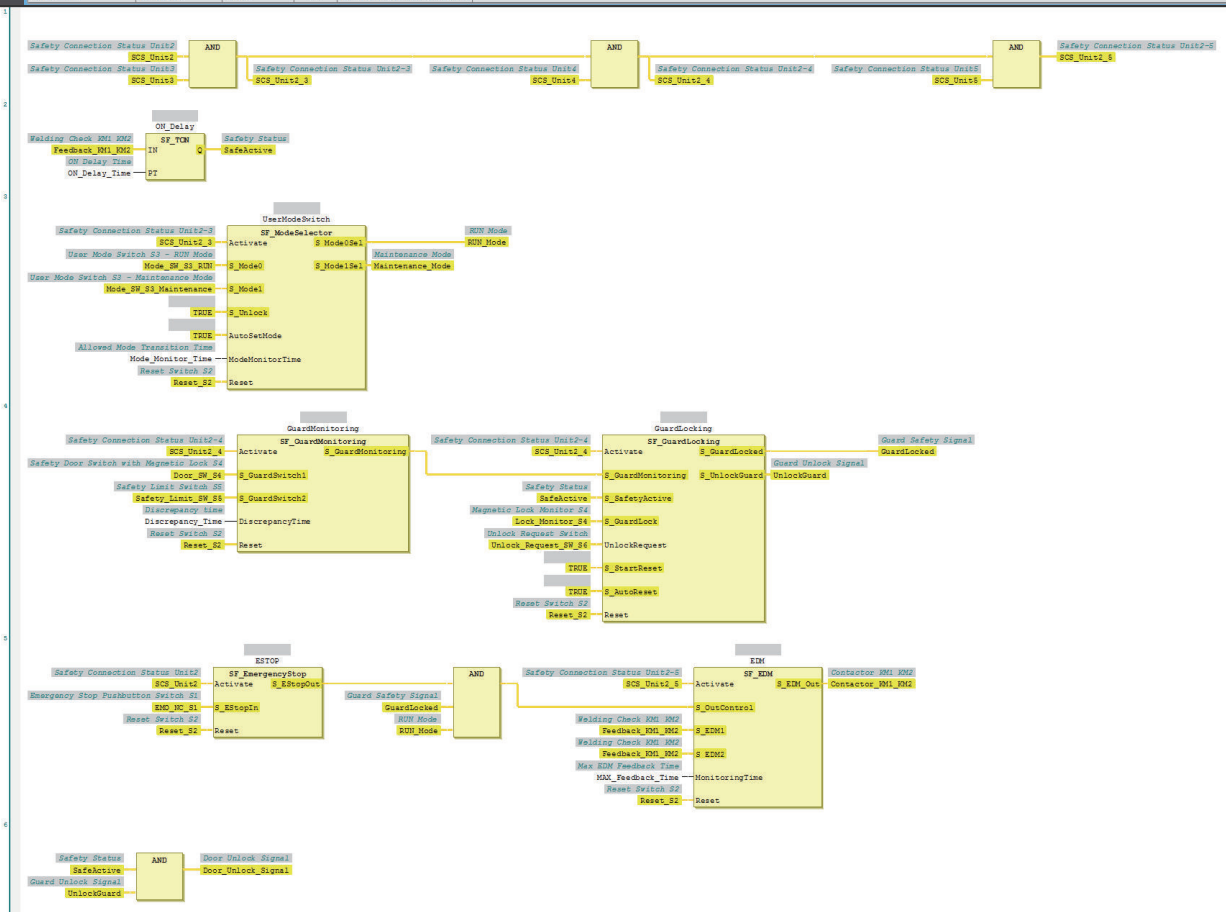
| External Device | Channel | Comment |
|---|---|---|
| Dual Output with Test Pulse | So 0 | 2 Safety Relays w/ Welding Check |
| | So 1 | |
| Single Channel with Test Pulse | So 2 | Door Unlock Signal |
| | So 3 | |

## I/O Map Settings

| Position | Port | R/W | Data Type | Variable | Variable Comment | Variable Type |
|---|---|---|---|---|---|---|
| | EtherCAT Network | | | | | |
| EtherCAT Master | Master | | | | | |
| Node1/Unit2 | NX-SIH400 | | | | | |
| | Safety Inputs and Status | | | | | |
| | Si00 Logical Value | R | SAFEBOOL | EMO_NC_S1 | Emergency Stop Pushbutton Switch(2NC) | Global Variables |
| | Si01 Logical Value | R | SAFEBOOL | | | |
| | Si02 Logical Value | R | SAFEBOOL | Reset_S2 | Reset Switch | Global Variables |
| | Si03 Logical Value | R | SAFEBOOL | Feedback_KM1_KM2 | EDM(Contact Welding Detection) | Global Variables |
| | Safety Connection Status | R | SAFEBOOL | SCS_Unit2 | Safety Connection Status Unit2 | Global Variables |
| | Safety Input Terminal Status | R | SAFEBOOL | | | |
| Node1/Unit3 | NX-SIH400 | | | | | |
| | Safety Inputs and Status | | | | | |
| | Si00 Logical Value | R | SAFEBOOL | Mode_SW_S3_RUN | Single Contact | Global Variables |
| | Si01 Logical Value | R | SAFEBOOL | Mode_SW_S3_Maintenance | Single Contact | Global Variables |
| | Si02 Logical Value | R | SAFEBOOL | | | |
| | Si03 Logical Value | R | SAFEBOOL | Unlock_Request_SW_S6 | Single Contact | Global Variables |
| | Safety Connection Status | R | SAFEBOOL | SCS_Unit3 | Safety Connection Status Unit3 | Global Variables |
| | Safety Input Terminal Status | R | SAFEBOOL | | | |
| Node1/Unit4 | NX-SIH400 | | | | | |
| | Safety Inputs and Status | | | | | |
| | Si00 Logical Value | R | SAFEBOOL | Lock_Monitor_S4 | Safety Switch(NC) | Global Variables |
| | Si01 Logical Value | R | SAFEBOOL | | | |
| | Si02 Logical Value | R | SAFEBOOL | Door_SW_S4 | Safety Switch(NC) | Global Variables |
| | Si03 Logical Value | R | SAFEBOOL | Safety_Limit_SW_S5 | Safety Limit Switch(NC) | Global Variables |
| | Safety Connection Status | R | SAFEBOOL | SCS_Unit4 | Safety Connection Status Unit4 | Global Variables |
| | Safety Input Terminal Status | R | SAFEBOOL | | | |
| Node1/Unit5 | NX-SOD400 | | | | | |
| | Status | | | | | |
| | Safety Connection Status | R | SAFEBOOL | SCS_Unit5 | Safety Connection Status Unit5 | Global Variables |
| | Safety Output Terminal Status | R | SAFEBOOL | | | |
| | Safety Outputs | | | | | |
| | So00 Output Value | W | SAFEBOOL | Contactor_KM1_KM2 | 2 Safety Relays w/ Welding Check | Global Variables |
| | So01 Output Value | W | SAFEBOOL | | | |
| | So02 Output Value | W | SAFEBOOL | Door_Unlock_Signal | Door Unlock Signal | Global Variables |
| | So03 Output Value | W | SAFEBOOL | | | |

## Program

| Name | Data Type | Initial Value | Constant | Comment |
|---|---|---|---|---|
| Mode_Monitor_Time | TIME | t#2s | ☑ | Allowed Mode Transition Time |
| MAX_Feedback_Time | TIME | t#300ms | ☑ | Max EDM Feedback Time |
| ON_Delay_Time | TIME | t#5s | ☑ | ON Delay Time |
| Discrepancy_Time | TIME | t#1000ms | ☑ | Discrepancy time |
| RUN_Mode | SAFEBOOL | FALSE | ☐ | RUN Mode |
| Maintenance_Mode | SAFEBOOL | FALSE | ☐ | Maintenance Mode |
| SafeActive | SAFEBOOL | FALSE | ☐ | Safety Status |
| GuardLocked | SAFEBOOL | FALSE | ☐ | Guard Safety Signal |
| UnlockGuard | SAFEBOOL | FALSE | ☐ | Guard Unlock Signal |
| SCS_Unit2_3 | SAFEBOOL | FALSE | ☐ | Safety Connection Status Unit2-3 |
| SCS_Unit2_4 | SAFEBOOL | FALSE | ☐ | Safety Connection Status Unit2-4 |
| SCS_Unit2_5 | SAFEBOOL | FALSE | ☐ | Safety Connection Status Unit2-5 |




### Precautions for Safe Use

- Test the functionality every six months to detect welded contactor contacts.
- The customer is responsible for attaining conformance of the entire system to standards.
- To detect electrical and mechanical failures, use a combination of redundant semiconductor output contacts and redundant mechanical output devices.
- Refer to ISO 14119:2013 for additional measures to minimize the possibility of interlocking devices associated with guards from being disabled.

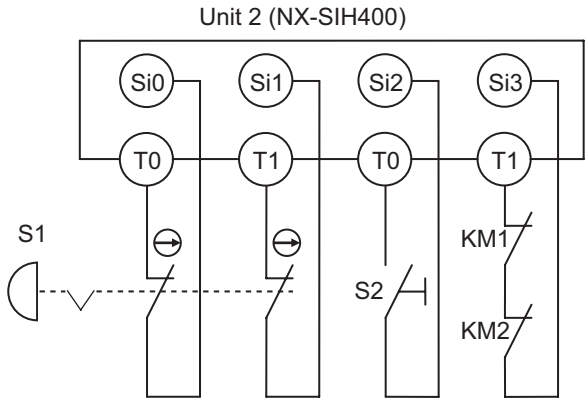
## A-3-5 Enable Switches

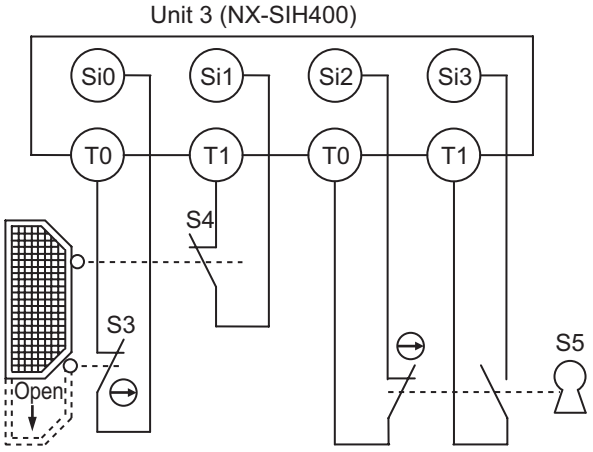
## Application Overview

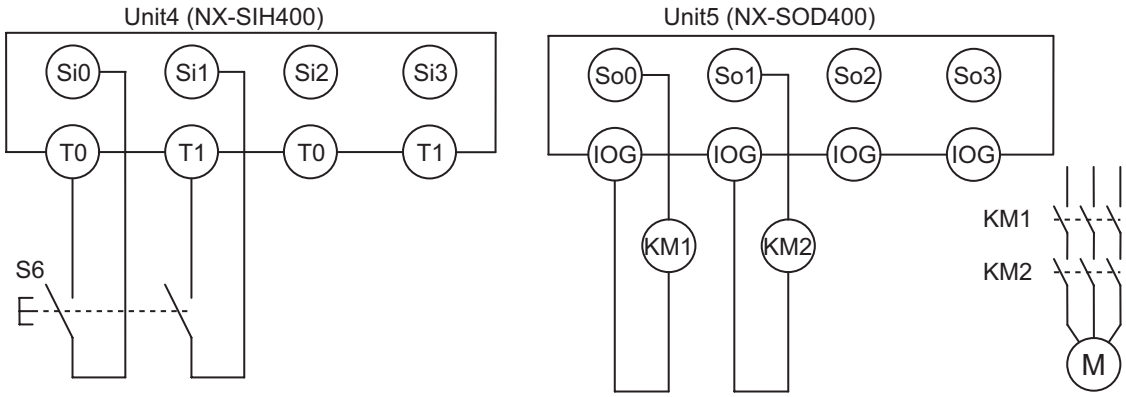
| Safety category/PL | Safety device | Stop category | Reset |
|---|---|---|---|
| Equivalent to 4/PLe<br>(Enable Switch) | • Emergency stop pushbutton<br>• Safety limit switch<br>• Key selector switch<br>• Enable switch | 0 | Manual |

Motor M stops when safety doors S3 and S4 are opened or key Selector switch S5 is maintenance mode.
However, even if key selector switch S5 is set to maintenance mode, motor M will operate if enable switch S6 is ON.

## Wiring

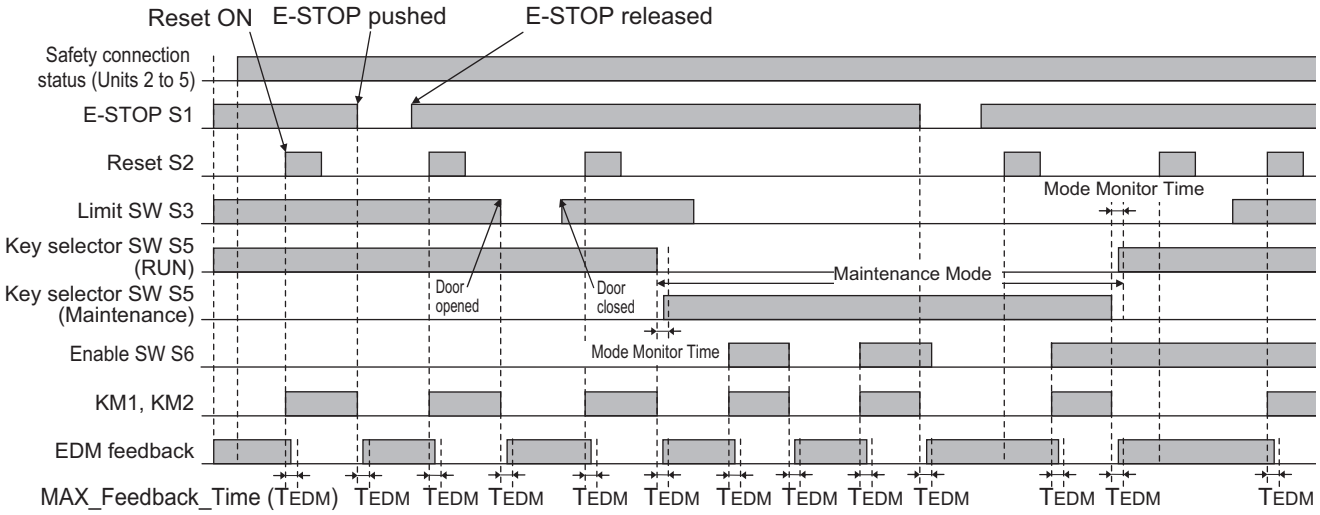





S1: Emergency stop pushbutton
S2: Reset switch
S3: Safety limit switch
S4: Limit switch (NO contacts)
S5: Key selector switch
S6: Enabling switch (When OMRON enabling switch A4E or A4EG is used)
KM1, KM2: Contactors
M: Motor

## Timing Chart




## Safety I/O Terminal & I/O Map Setting

## ● Safety I/O Terminal Settings

**Node1/Unit2 : NX-SIH400 (N2 : Instance0)**

| External Device | Channel | Discrepancy | On-Off | Off-On | Test Source | Comment |
|---|---|---|---|---|---|---|
| Mechanical Contact for Dual Channel Equivalent | Si 0 | 500ms | 0ms | 0ms | T0 | Emergency Stop Pushbutton Switch(2NC) |
| | Si 1 | 500ms | 0ms | 0ms | T1 | |
| Mechanical Contact For Single Channel | Si 2 | 0ms | 0ms | 0ms | T0 | Reset Switch |
| Mechanical Contact For Single Channel | Si 3 | 0ms | 0ms | 0ms | T1 | EDM(Contact Welding Detection) |

**Node1/Unit3 : NX-SIH400 (N3 : Instance1)**

| External Device | Channel | Discrepancy | On-Off | Off-On | Test Source | Comment |
|---|---|---|---|---|---|---|
| Mechanical Contact For Single Channel | Si 0 | 0ms | 0ms | 0ms | T0 | Safety Limit Switch(NC) |
| Mechanical Contact For Single Channel | Si 1 | 0ms | 0ms | 0ms | T1 | Limit Switch(NO) |
| Mechanical Contact For Single Channel | Si 2 | 0ms | 0ms | 0ms | T0 | Single Contact |
| Mechanical Contact For Single Channel | Si 3 | 0ms | 0ms | 0ms | T1 | Single Contact |

**Node1/Unit4 : NX-SIH400 (N4 : Instance2)**

| External Device | Channel | Discrepancy | On-Off | Off-On | Test Source | Comment |
|---|---|---|---|---|---|---|
| Mechanical Contact for Dual Channel Equivalent | Si 0 | 500ms | 0ms | 0ms | T0 | Enable Switch(2NO) |
| | Si 1 | 500ms | 0ms | 0ms | T1 | |
| | Si 2 | | | | | |
| | Si 3 | | | | | |

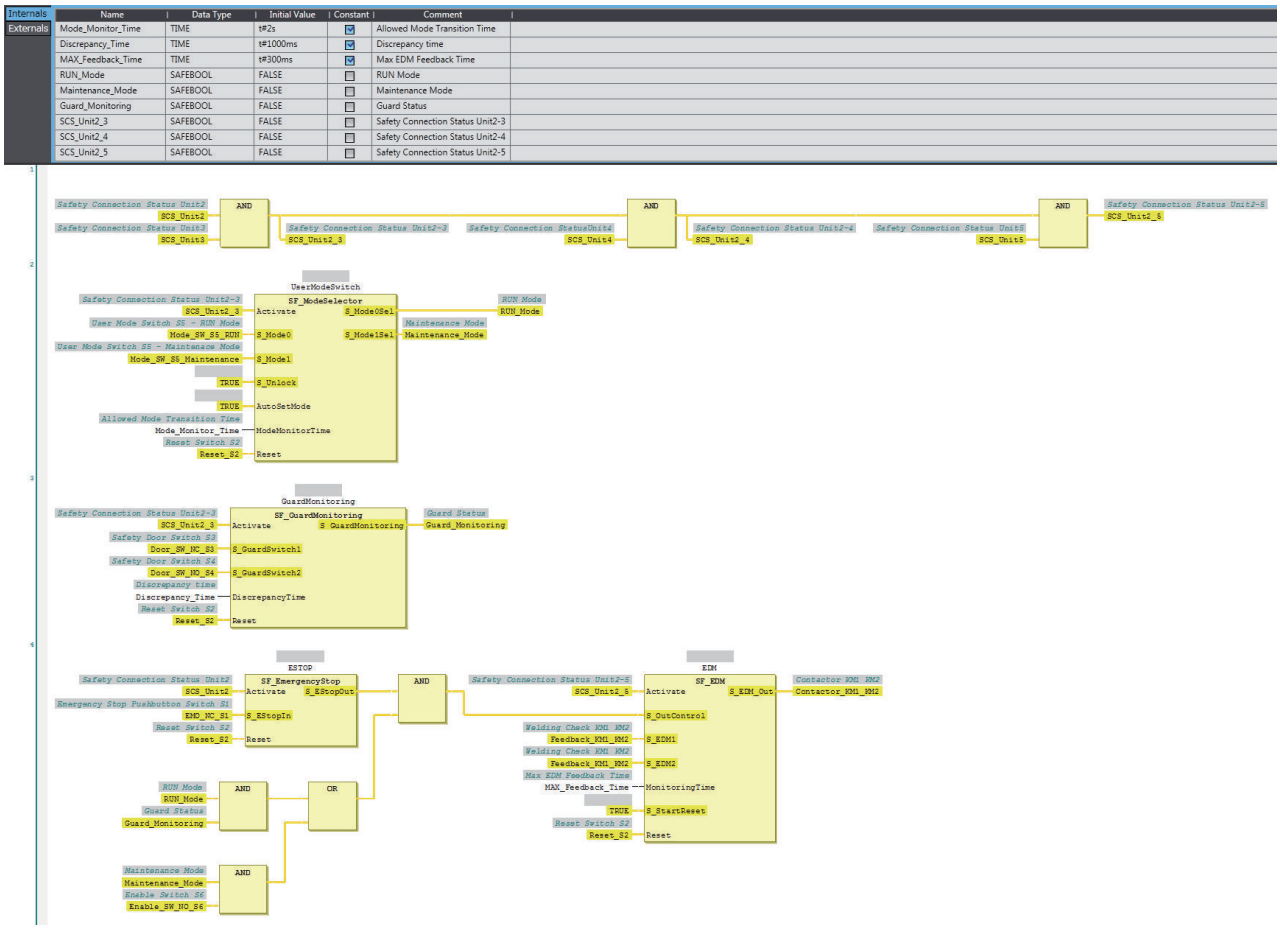
**Node1/Unit5 : NX-SOD400 (N5 : Instance3)**

| External Device | Channel | Comment |
|---|---|---|
| Dual Output with Test Pulse | So 0 | 2 Safety Relays w/ Welding Check |
| | So 1 | |
| | So 2 | |
| | So 3 | |

## ● I/O Map Settings

| Position | Port | R/W | Data Type | Variable | Variable Comment | Variable Type |
|---|---|---|---|---|---|---|
| | ▼ EtherCAT Network | | | | | |
| EtherCAT Master | Master | | | | | |
| Node1/Unit2 | ▼ NX-SIH400 | | | | | |
| | ▼ Safety Inputs and Status | | | | | |
| | Si00 Logical Value | R | SAFEBOOL | EMO_NC_S1 | Emergency Stop Pushbutton Switch(2NC) | Global Variables |
| | Si01 Logical Value | R | SAFEBOOL | | | |
| | Si02 Logical Value | R | SAFEBOOL | Reset_S2 | Reset Switch | Global Variables |
| | Si03 Logical Value | R | SAFEBOOL | Feedback_KM1_KM2 | EDM(Contact Welding Detection) | Global Variables |
| | Safety Connection Status | R | SAFEBOOL | SCS_Unit2 | Safety Connection Status Unit2 | Global Variables |
| | Safety Input Terminal Status | R | SAFEBOOL | | | |
| Node1/Unit3 | ▼ NX-SIH400 | | | | | |
| | ▼ Safety Inputs and Status | | | | | |
| | Si00 Logical Value | R | SAFEBOOL | Door_SW_NC_S3 | Safety Limit Switch(NC) | Global Variables |
| | Si01 Logical Value | R | SAFEBOOL | Door_SW_NO_S4 | Limit Switch(NO) | Global Variables |
| | Si02 Logical Value | R | SAFEBOOL | Mode_SW_S5_RUN | Single Contact | Global Variables |
| | Si03 Logical Value | R | SAFEBOOL | Mode_SW_S5_Maintenance | Single Contact | Global Variables |
| | Safety Connection Status | R | SAFEBOOL | SCS_Unit3 | Safety Connection Status Unit3 | Global Variables |
| | Safety Input Terminal Status | R | SAFEBOOL | | | |
| Node1/Unit4 | ▼ NX-SIH400 | | | | | |
| | ▼ Safety Inputs and Status | | | | | |
| | Si00 Logical Value | R | SAFEBOOL | Enable_SW_NO_S6 | Enable Switch(2NO) | Global Variables |
| | Si01 Logical Value | R | SAFEBOOL | | | |
| | Si02 Logical Value | R | SAFEBOOL | | | |
| | Si03 Logical Value | R | SAFEBOOL | | | |
| | Safety Connection Status | R | SAFEBOOL | SCS_Unit4 | Safety Connection Status Unit4 | Global Variables |
| | Safety Input Terminal Status | R | SAFEBOOL | | | |
| Node1/Unit5 | ▼ NX-SOD400 | | | | | |
| | ▼ Status | | | | | |
| | Safety Connection Status | R | SAFEBOOL | SCS_Unit5 | Safety Connection Status Unit5 | Global Variables |
| | Safety Output Terminal Status | R | SAFEBOOL | | | |
| | ▼ Safety Outputs | | | | | |
| | So00 Output Value | W | SAFEBOOL | Contactor_KM1_KM2 | 2 Safety Relays w/ Welding Check | Global Variables |
| | So01 Output Value | W | SAFEBOOL | | | |
| | So02 Output Value | W | SAFEBOOL | | | |
| | So03 Output Value | W | SAFEBOOL | | | |

## Program




### Precautions for Safe Use

- Test the functionality every six months to detect welded contactor contacts.
- The customer is responsible for attaining conformance of the entire system to standards.
- To detect electrical and mechanical failures, use a combination of redundant semiconductor output contacts and redundant mechanical output devices.
- Refer to ISO 14119:2013 for additional measures to minimize the possibility of interlocking devices associated with guards from being disabled.

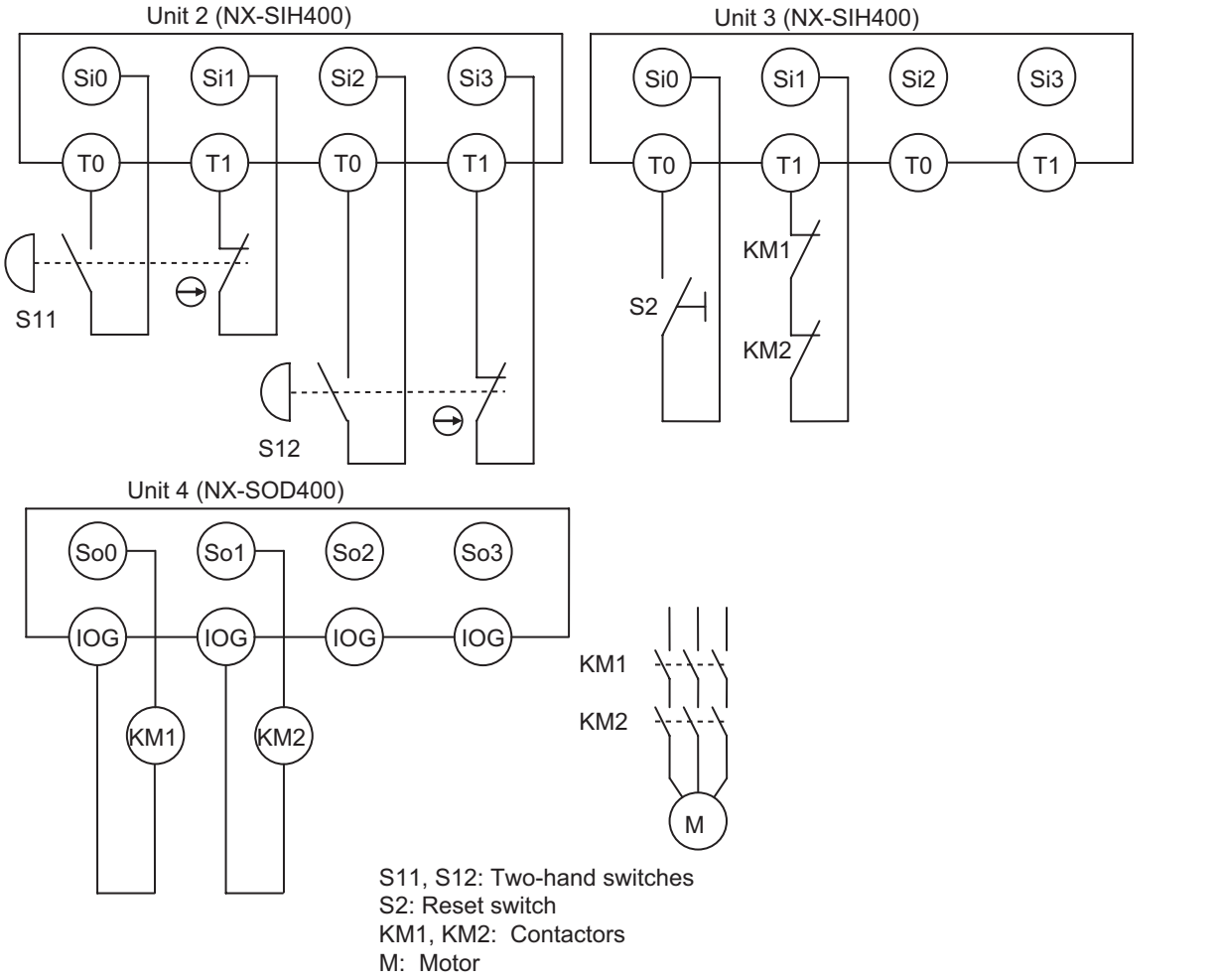
# A-3-6 Two-hand Switches

## Application Overview

| Safety category/PL | Safety device | Stop category | Reset |
|---|---|---|---|
| Equivalent to 4/PLe | Two-hand control switch | 0 | Auto |

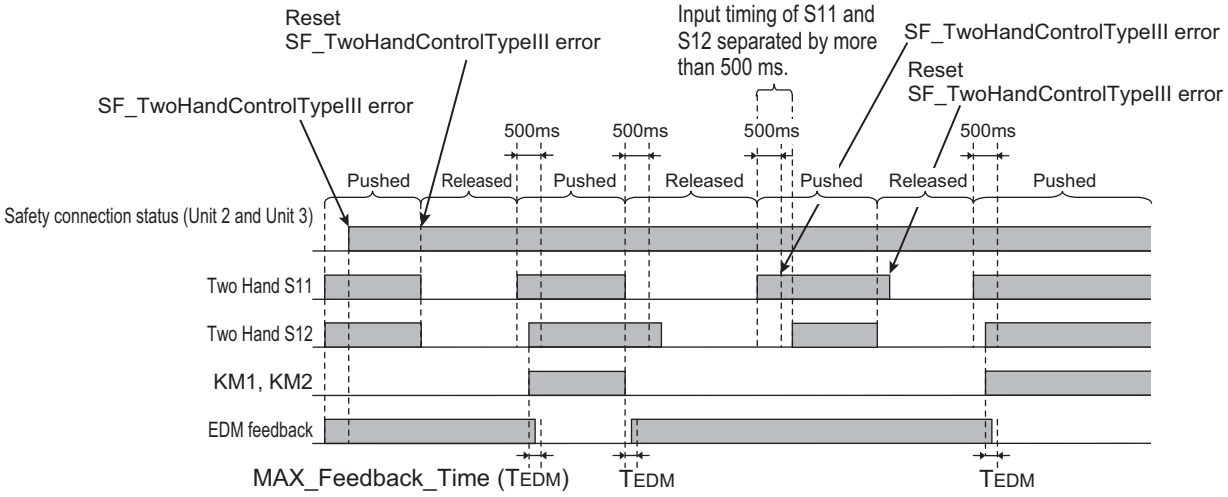
Motor M operates when two-hand control switches S11 and S12 are pressed at the same time.

## Wiring




S11, S12: Two-hand switches
S2: Reset switch
KM1, KM2: Contactors
M: Motor

## Timing Chart



# Safety I/O Terminal & I/O Map Setting

## ● Safety I/O Terminal Settings

**Node1/Unit2 : NX-SIH400 (N2 : Instance0)**

| External Device | Channel | Discrepancy | On-Off | Off-On | Test Source | Comment |
|---|---|---|---|---|---|---|
| Mechanical Contact for Dual Channel Complementary | Si 0 | 500ms | 0ms | 0ms | T0 | Two-hand Control Switch |
| | Si 1 | 500ms | 0ms | 0ms | T1 | |
| Mechanical Contact for Dual Channel Complementary | Si 2 | 500ms | 0ms | 0ms | T0 | Two-hand Control Switch |
| | Si 3 | 500ms | 0ms | 0ms | T1 | |

**Node1/Unit3 : NX-SIH400 (N3 : Instance1)**

| External Device | Channel | Discrepancy | On-Off | Off-On | Test Source | Comment |
|---|---|---|---|---|---|---|
| Mechanical Contact For Single Channel | Si 0 | 0ms | 0ms | 0ms | T0 | Reset Switch |
| Mechanical Contact For Single Channel | Si 1 | 0ms | 0ms | 0ms | T1 | EDM(Contact Welding Detection) |
| | Si 2 | | | | | |
| | Si 3 | | | | | |

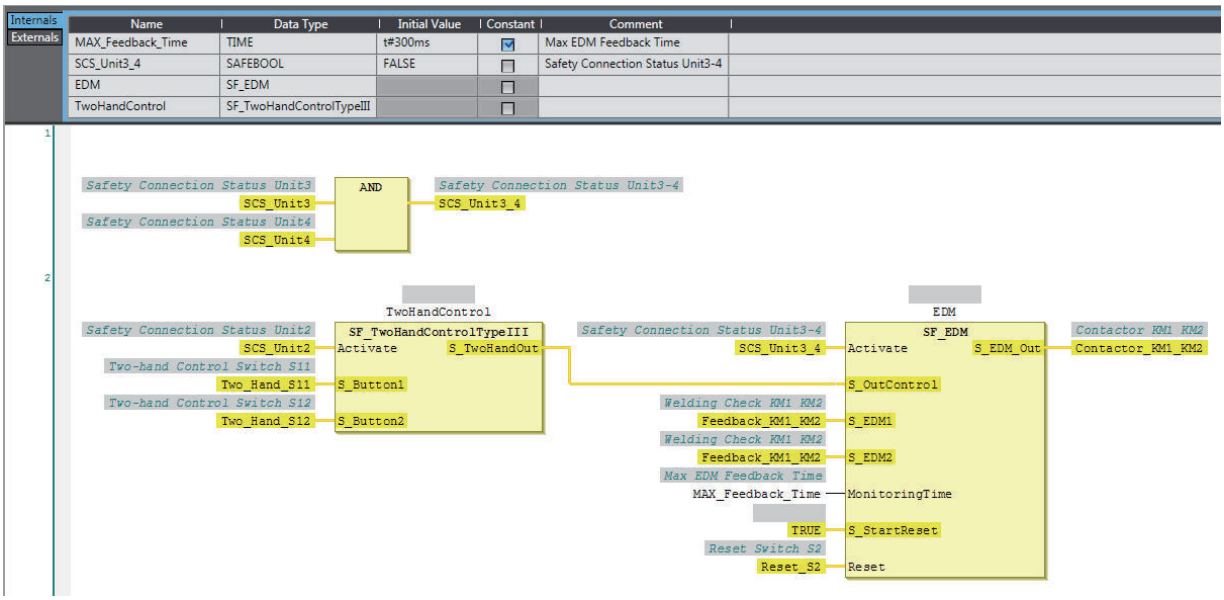
**Node1/Unit4 : NX-SOD400 (N4 : Instance2)**

| External Device | Channel | Comment |
|---|---|---|
| Dual Output with Test Pulse | So 0 | 2 Safety Relays w/ Welding Check |
| | So 1 | |
| | So 2 | |
| | So 3 | |

## ● I/O Map Settings

| Position | Port | R/W | Data Type | Variable | Variable Comment | Variable Type |
|---|---|---|---|---|---|---|
| | ▼ EtherCAT Network | | | | | |
| EtherCAT Master | Master | | | | | |
| Node1/Unit2 | ▼ NX-SIH400 | | | | | |
| | ▼ Safety Inputs and Status | | | | | |
| | Si00 Logical Value | R | SAFEBOOL | Two_Hand_S11 | Two-hand Control Switch S11 | Global Variables |
| | Si01 Logical Value | R | SAFEBOOL | | | |
| | Si02 Logical Value | R | SAFEBOOL | Two_Hand_S12 | Two-hand Control Switch S12 | Global Variables |
| | Si03 Logical Value | R | SAFEBOOL | | | |
| | Safety Connection Status | R | SAFEBOOL | SCS_Unit2 | Safety Connection Status Unit2 | Global Variables |
| | Safety Input Terminal Status | R | SAFEBOOL | | | |
| Node1/Unit3 | ▼ NX-SIH400 | | | | | |
| | ▼ Safety Inputs and Status | | | | | |
| | Si00 Logical Value | R | SAFEBOOL | Reset_S2 | Reset Switch S2 | Global Variables |
| | Si01 Logical Value | R | SAFEBOOL | Feedback_KM1_KM2 | Welding Check KM1_KM2 | Global Variables |
| | Si02 Logical Value | R | SAFEBOOL | | | |
| | Si03 Logical Value | R | SAFEBOOL | | | |
| | Safety Connection Status | R | SAFEBOOL | SCS_Unit3 | Safety Connection Status Unit3 | Global Variables |
| | Safety Input Terminal Status | R | SAFEBOOL | | | |
| Node1/Unit4 | ▼ NX-SOD400 | | | | | |
| | ▼ Status | | | | | |
| | Safety Connection Status | R | SAFEBOOL | SCS_Unit4 | Safety Connection Status Unit4 | Global Variables |
| | Safety Output Terminal Status | R | SAFEBOOL | | | |
| | ▼ Safety Outputs | | | | | |
| | So00 Output Value | W | SAFEBOOL | Contactor_KM1_KM2 | Contactor KM1_KM2 | Global Variables |
| | So01 Output Value | W | SAFEBOOL | | | |
| | So02 Output Value | W | SAFEBOOL | | | |
| | So03 Output Value | W | SAFEBOOL | | | |

# Program




### Precautions for Safe Use

- Test the functionality every six months to detect welded contactor contacts.
- The customer is responsible for attaining conformance of the entire system to standards.
- To detect electrical and mechanical failures, use a combination of redundant semiconductor output contacts and redundant mechanical output devices.

### Additional Information

In this example, a reset switch is used to reset EDM errors.

## A-3-7 D40A Non-contact Door Switches

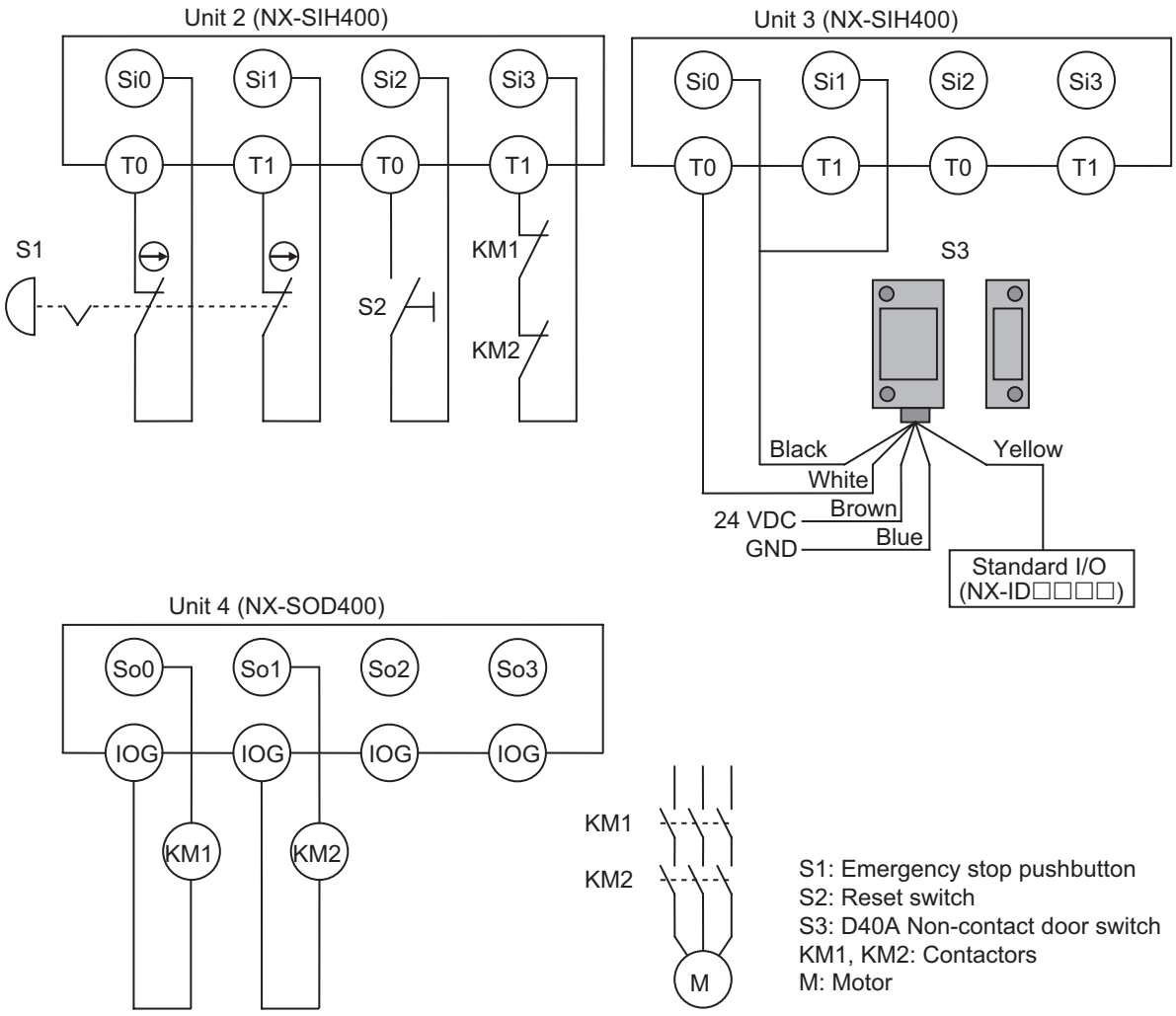
### Application Overview

| Safety category/PL | Safety device | Stop category | Reset |
|---|---|---|---|
| Equivalent to 3/PLd<br>(Non-contact Door Switch) | • Emergency stop pushbutton<br>• D40A Non-contact Door Switch | 0 | Manual |

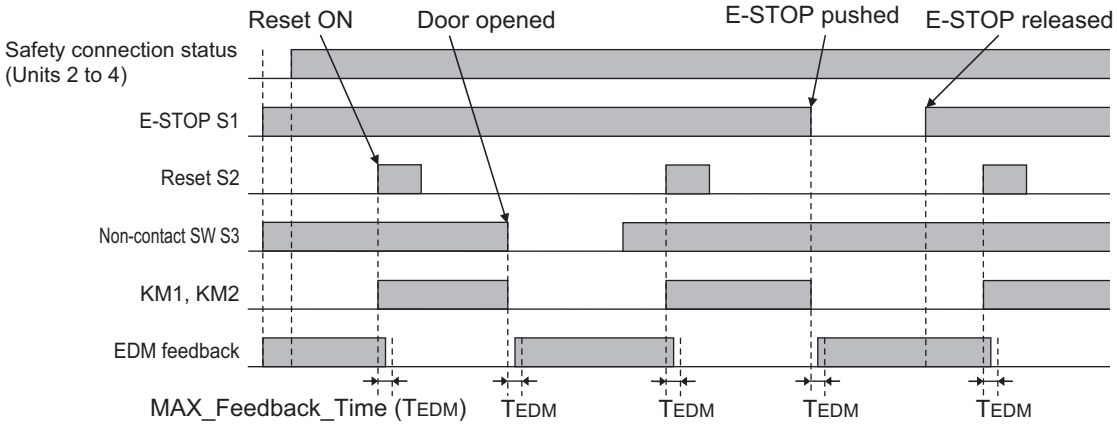
Motor M stops when emergency stop pushbutton S1 is pressed.

If either of the S3 safety doors (D40A Non-contact Door Switches) is opened, motor M will stop.

## Wiring




## Timing Chart




## Safety I/O Terminal & I/O Map Setting

## ● Safety I/O Terminal Settings

**Node1/Unit2 : NX-SIH400 (N2 : Instance0)**

| External Device | Channel | Discrepancy | On-Off | Off-On | Test Source | Comment |
|---|---|---|---|---|---|---|
| Mechanical Contact for Dual Channel Equivalent | Si 0 | 500ms | 0ms | 0ms | T0 | Emergency Stop Pushbutton Switch(2NC) |
| | Si 1 | 500ms | 0ms | 0ms | T1 | |
| Mechanical Contact For Single Channel | Si 2 | 0ms | 0ms | 0ms | T0 | Reset Switch |
| Mechanical Contact For Single Channel | Si 3 | 0ms | 0ms | 0ms | T1 | EDM(Contact Welding Detection) |

**Node1/Unit3 : NX-SIH400 (N3 : Instance1)**

| External Device | Channel | Discrepancy | On-Off | Off-On | Test Source | Comment |
|---|---|---|---|---|---|---|
| Non-contact switch | Si 0 | 0ms | 0ms | 0ms | T0 | Non-contact Door Switch |
| | Si 1 | 0ms | 0ms | 0ms | T0 | |
| | Si 2 | | | | | |
| | Si 3 | | | | | |

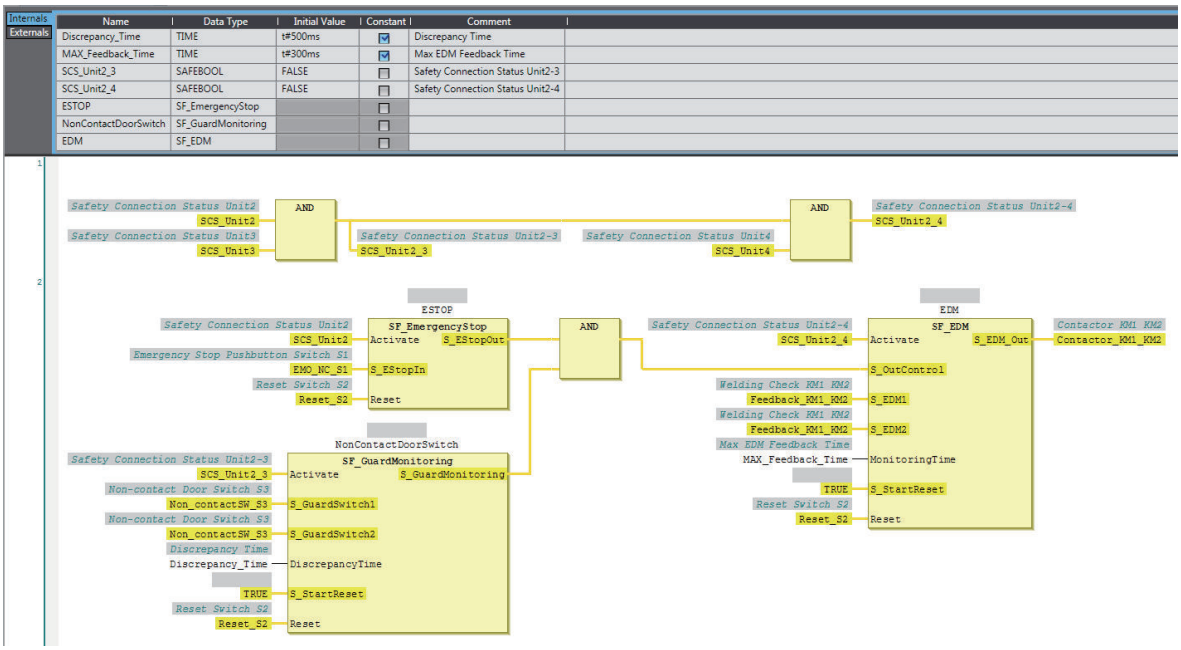
**Node1/Unit4 : NX-SOD400 (N4 : Instance2)**

| External Device | Channel | Comment |
|---|---|---|
| Dual Output with Test Pulse | So 0 | 2 Safety Relays w/ Welding Check |
| | So 1 | |
| | So 2 | |
| | So 3 | |

## ● I/O Map Settings

| Position | Port | R/W | Data Type | Variable | Variable Comment | Variable Type |
|---|---|---|---|---|---|---|
| | ▼ EtherCAT Network | | | | | |
| EtherCAT Master | Master | | | | | |
| Node1/Unit2 | ▼ NX-SIH400 | | | | | |
| | ▼ Safety Inputs and Status | | | | | |
| | Si00 Logical Value | R | SAFEBOOL | EMO_NC_S1 | Emergency Stop Pushbutton Switch S1 | Global Variables |
| | Si01 Logical Value | R | SAFEBOOL | | | |
| | Si02 Logical Value | R | SAFEBOOL | Reset_S2 | Reset Switch S2 | Global Variables |
| | Si03 Logical Value | R | SAFEBOOL | Feedback_KM1_KM2 | Welding Check KM1_KM2 | Global Variables |
| | Safety Connection Status | R | SAFEBOOL | SCS_Unit2 | Safety Connection Status Unit2 | Global Variables |
| | Safety Input Terminal Status | R | SAFEBOOL | | | |
| Node1/Unit3 | ▼ NX-SIH400 | | | | | |
| | ▼ Safety Inputs and Status | | | | | |
| | Si00 Logical Value | R | SAFEBOOL | Non_contactSW_S3 | Non-contact Door Switch S3 | Global Variables |
| | Si01 Logical Value | R | SAFEBOOL | | | |
| | Si02 Logical Value | R | SAFEBOOL | | | |
| | Si03 Logical Value | R | SAFEBOOL | | | |
| | Safety Connection Status | R | SAFEBOOL | SCS_Unit3 | Safety Connection Status Unit3 | Global Variables |
| | Safety Input Terminal Status | R | SAFEBOOL | | | |
| Node1/Unit4 | ▼ NX-SOD400 | | | | | |
| | ▼ Status | | | | | |
| | Safety Connection Status | R | SAFEBOOL | SCS_Unit4 | Safety Connection Status Unit4 | Global Variables |
| | Safety Output Terminal Status | R | SAFEBOOL | | | |
| | ▼ Safety Outputs | | | | | |
| | So00 Output Value | W | SAFEBOOL | Contactor_KM1_KM2 | Contactor KM1_KM2 | Global Variables |
| | So01 Output Value | W | SAFEBOOL | | | |
| | So02 Output Value | W | SAFEBOOL | | | |
| | So03 Output Value | W | SAFEBOOL | | | |

## Program




### Precautions for Safe Use

- Test the functionality every six months to detect welded contactor contacts.
- The customer is responsible for attaining conformance of the entire system to standards.
- To detect electrical and mechanical failures, use a combination of redundant semiconductor output contacts and redundant mechanical output devices.
- Refer to ISO 14119:2013 for additional measures to minimize the possibility of interlocking devices associated with guards from being disabled.

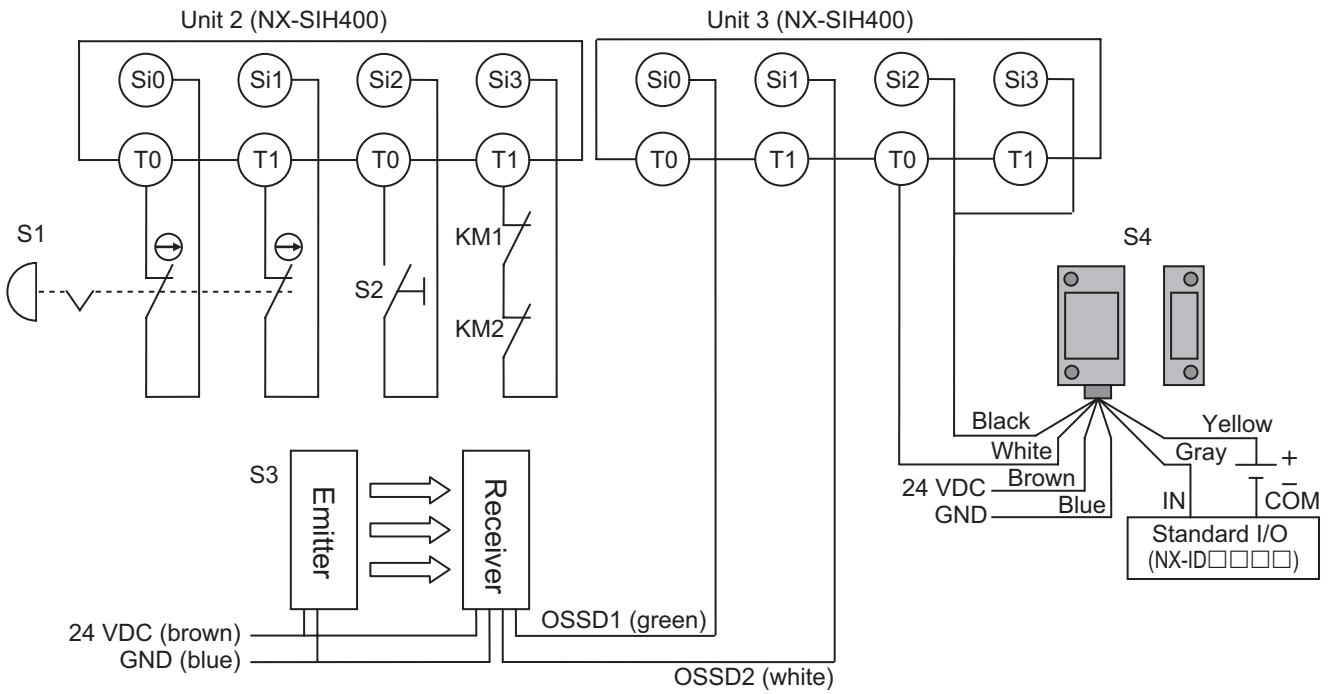
# A-3-8 D40A-2/D40Z Non-contact Door Switches

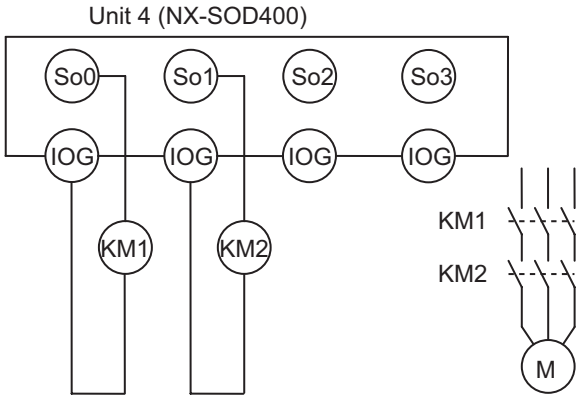
## Application Overview

| Safety category/PL | Safety device | Stop category | Reset |
|---|---|---|---|
| Equivalent to 4/PLe<br>(Safety Light Curtain) | • Safety light curtain<br>• Emergency stop pushbutton | 0 | Manual |

If the light in the safety light curtain S3 is interrupted and the non-contact door switch S4 turns OFF at the same time, the outputs are turned OFF.
The outputs also turn OFF when emergency pushbutton S1 is pressed.

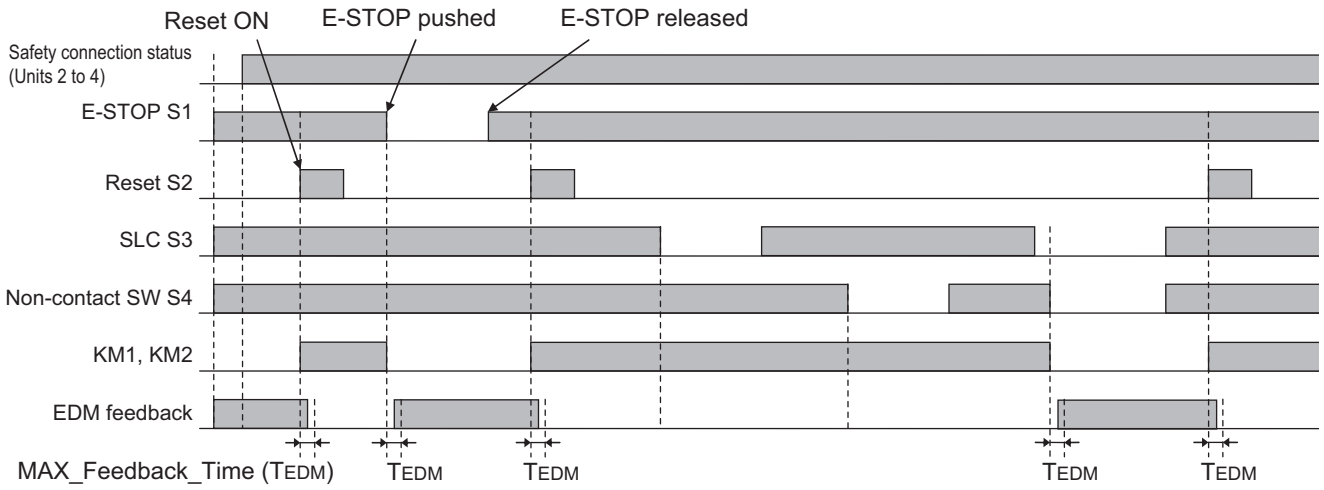
## Wiring




S1: Emergency stop pushbutton
S2: Reset switch
S3: Safety light curtain
S4: D40A-2D□ or D40Z Non-contact door switch
KM1, KM2: Contactors
M: Motor

## Timing Chart




## Safety I/O Terminal & I/O Map Setting

### ● Safety I/O Terminal Settings

**Node1/Unit2 : NX-SIH400 (N2 : Instance0)**

| External Device | Channel | Discrepancy | On-Off | Off-On | Test Source | Comment |
|---|---|---|---|---|---|---|
| Mechanical Contact for Dual Channel Equivalent | Si 0 | 500ms | 0ms | 0ms | T0 | Emergency Stop Pushbutton Switch(2NC) |
| | Si 1 | 500ms | 0ms | 0ms | T1 | |
| Mechanical Contact For Single Channel | Si 2 | 0ms | 0ms | 0ms | T0 | Reset Switch |
| Mechanical Contact For Single Channel | Si 3 | 0ms | 0ms | 0ms | T1 | EDM(Contact Welding Detection) |

**Node1/Unit3 : NX-SIH400 (N3 : Instance1)**

| External Device | Channel | Discrepancy | On-Off | Off-On | Test Source | Comment |
|---|---|---|---|---|---|---|
| Semiconductor Output for Dual Channel Equivalent | Si 0 | 500ms | 0ms | 0ms | Not Used | Safety Light Curtain |
| | Si 1 | 500ms | 0ms | 0ms | Not Used | |
| Non-contact switch | Si 2 | 0ms | 0ms | 0ms | T0 | Non-contact Door Switch |
| | Si 3 | 0ms | 0ms | 0ms | T0 | |

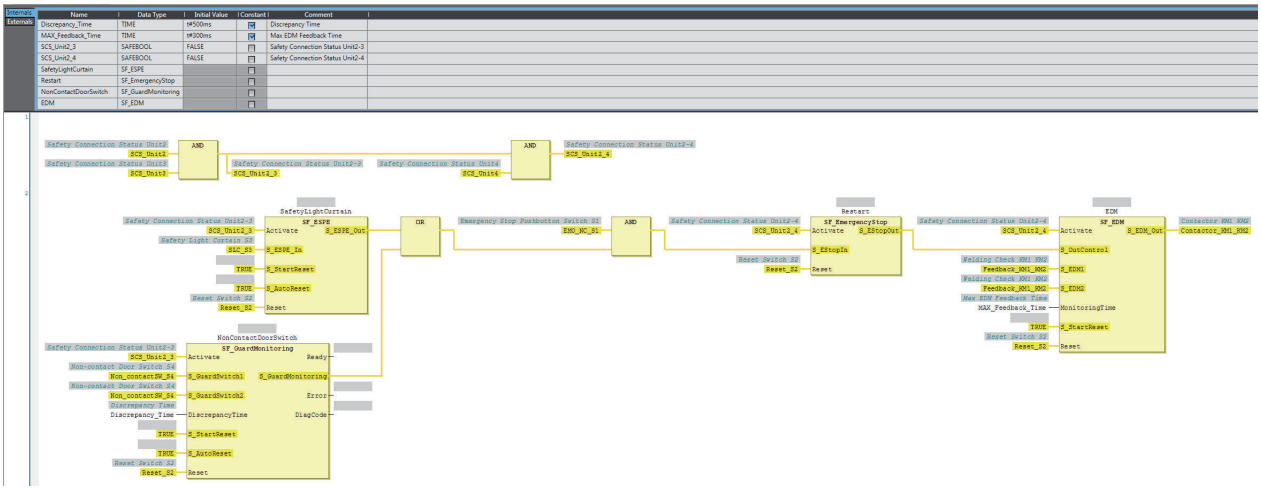
**Node1/Unit4 : NX-SOD400 (N4 : Instance2)**

| External Device | Channel | Comment |
|---|---|---|
| Dual Output with Test Pulse | So 0 | 2 Safety Relays w/ Welding Check |
| | So 1 | |
| | So 2 | |
| | So 3 | |

## ● I/O Map Settings

| Position | Port | R/W | Data Type | Variable | Variable Comment | Variable Type |
|---|---|---|---|---|---|---|
| | EtherCAT Network | | | | | |
| EtherCAT Master | Master | | | | | |
| Node1/Unit2 | NX-SIH400 | | | | | |
| | Safety Inputs and Status | | | | | |
| | Si00 Logical Value | R | SAFEBOOL | EMO_NC_S1 | Emergency Stop Pushbutton Switch S1 | Global Variables |
| | Si01 Logical Value | R | SAFEBOOL | | | |
| | Si02 Logical Value | R | SAFEBOOL | Reset_S2 | Reset Switch S2 | Global Variables |
| | Si03 Logical Value | R | SAFEBOOL | Feedback_KM1_KM2 | Welding Check KM1_KM2 | Global Variables |
| | Safety Connection Status | R | SAFEBOOL | SCS_Unit2 | Safety Connection Status Unit2 | Global Variables |
| | Safety Input Terminal Status | R | SAFEBOOL | | | |
| Node1/Unit3 | NX-SIH400 | | | | | |
| | Safety Inputs and Status | | | | | |
| | Si00 Logical Value | R | SAFEBOOL | SLC_S3 | Safety Light Curtain S3 | Global Variables |
| | Si01 Logical Value | R | SAFEBOOL | | | |
| | Si02 Logical Value | R | SAFEBOOL | Non_contactSW_S4 | Non-contact Door Switch S4 | Global Variables |
| | Si03 Logical Value | R | SAFEBOOL | | | |
| | Safety Connection Status | R | SAFEBOOL | SCS_Unit3 | Safety Connection Status Unit3 | Global Variables |
| | Safety Input Terminal Status | R | SAFEBOOL | | | |
| Node1/Unit4 | NX-SOD400 | | | | | |
| | Status | | | | | |
| | Safety Connection Status | R | SAFEBOOL | SCS_Unit4 | Safety Connection Status Unit4 | Global Variables |
| | Safety Output Terminal Status | R | SAFEBOOL | | | |
| | Safety Outputs | | | | | |
| | So00 Output Value | W | SAFEBOOL | Contactor_KM1_KM2 | Contactor KM1_KM2 | Global Variables |
| | So01 Output Value | W | SAFEBOOL | | | |
| | So02 Output Value | W | SAFEBOOL | | | |
| | So03 Output Value | W | SAFEBOOL | | | |

# Program



**Precautions for Safe Use**

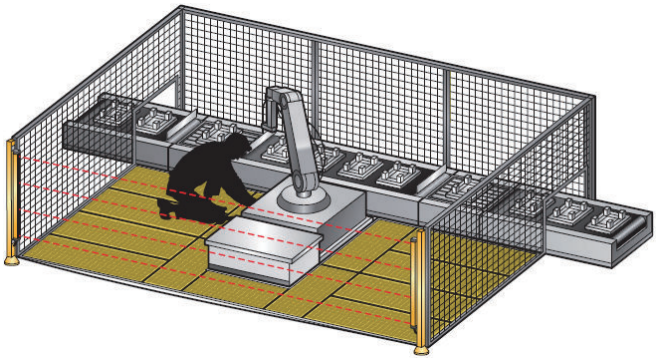
- Test the functionality every six months to detect welded contactor contacts.
- The customer is responsible for attaining conformance of the entire system to standards.
- To detect electrical and mechanical failures, use a combination of redundant semiconductor output contacts and redundant mechanical output devices.

## A-3-9 Safety Mats and Safety Light Curtains

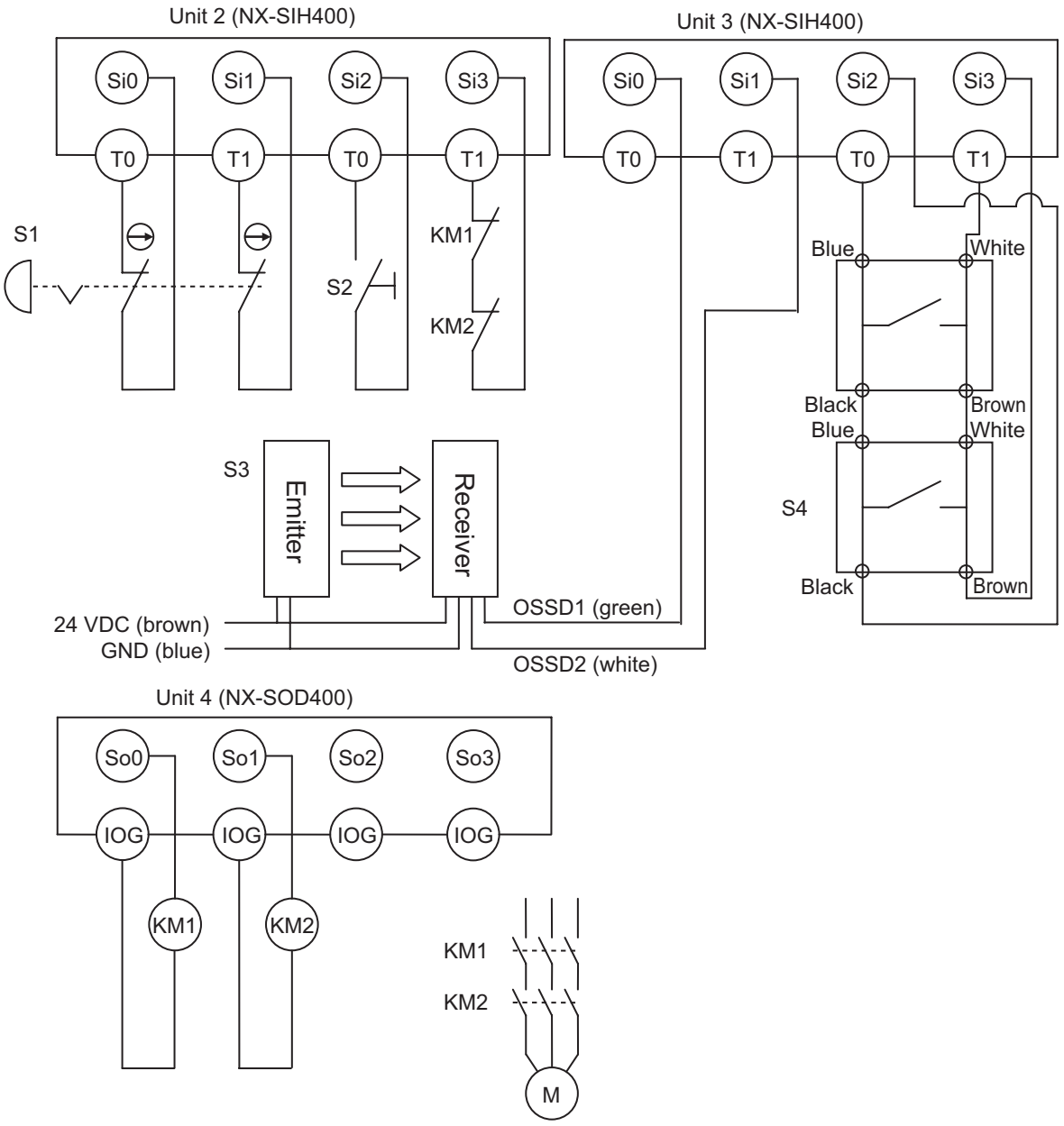
## Application Overview

| Safety category/PL | Safety device | Stop category | Reset |
|---|---|---|---|
| Equivalent to 3/PLd<br>(Safety Mat) | • Emergency stop pushbutton<br>• Safety light curtain<br>• Safety mat | 0 | Manual |

Safety light curtain monitors apertural area of safeguarded space and safety mat monitors inside of safeguarded space.
If the light in safety light curtain S3 is interrupted or safety mat S4 detects a person or object, motor M will stop.

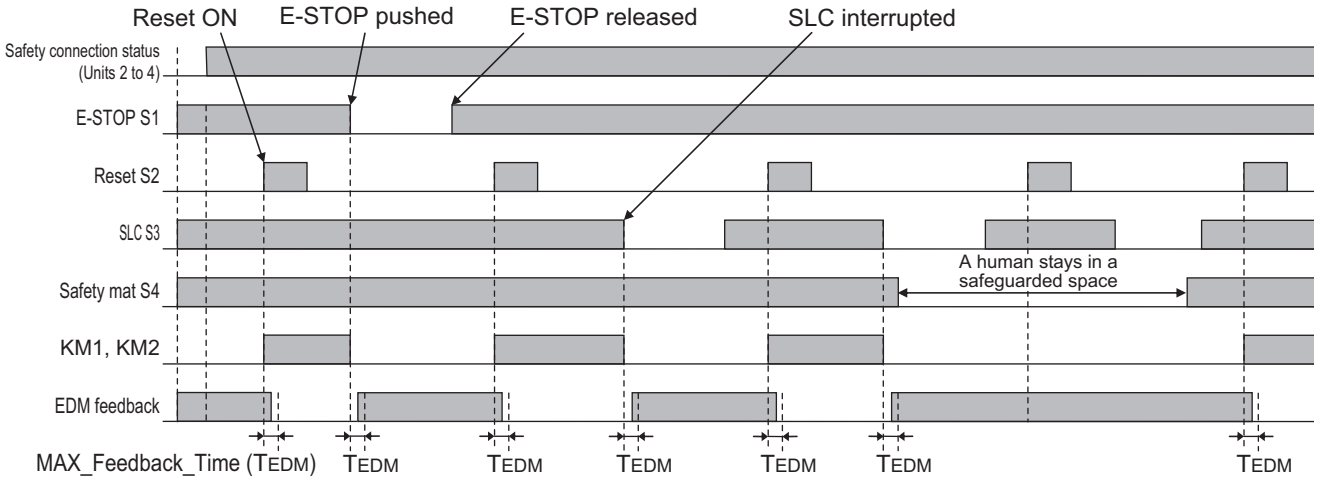


## Wiring




S1: Emergency stop pushbutton
S2: Reset switch
S3: Safety light curtain
S4: Safety mat
KM1, KM2: Contactors
M: Motor

## Timing Chart




## Safety I/O Terminal & I/O Map Setting

### ● Safety I/O Terminal Settings

**Node1/Unit2 : NX-SIH400 (N2 : Instance0)**

| External Device | Channel | Discrepancy | On-Off | Off-On | Test Source | Comment |
|---|---|---|---|---|---|---|
| Mechanical Contact for Dual Channel Equivalent | Si 0 | 500ms | 0ms | 0ms | T0 | Emergency Stop Pushbutton Switch(2NC) |
| | Si 1 | 500ms | 0ms | 0ms | T1 | |
| Mechanical Contact For Single Channel | Si 2 | 0ms | 0ms | 0ms | T0 | Reset Switch |
| Mechanical Contact For Single Channel | Si 3 | 0ms | 0ms | 0ms | T1 | EDM(Contact Welding Detection) |

**Node1/Unit3 : NX-SIH400 (N3 : Instance1)**

| External Device | Channel | Discrepancy | On-Off | Off-On | Test Source | Comment |
|---|---|---|---|---|---|---|
| Semiconductor Output for Dual Channel Equivalent | Si 0 | 500ms | 0ms | 0ms | Not Used | Dual Safety Semiconductor Output(Equivalent) |
| | Si 1 | 500ms | 0ms | 0ms | Not Used | |
| Safety Mat/Safety Edge | Si 2 | 0ms | 0ms | 0ms | T0 | Safety Mat |
| | Si 3 | 0ms | 0ms | 0ms | T1 | |

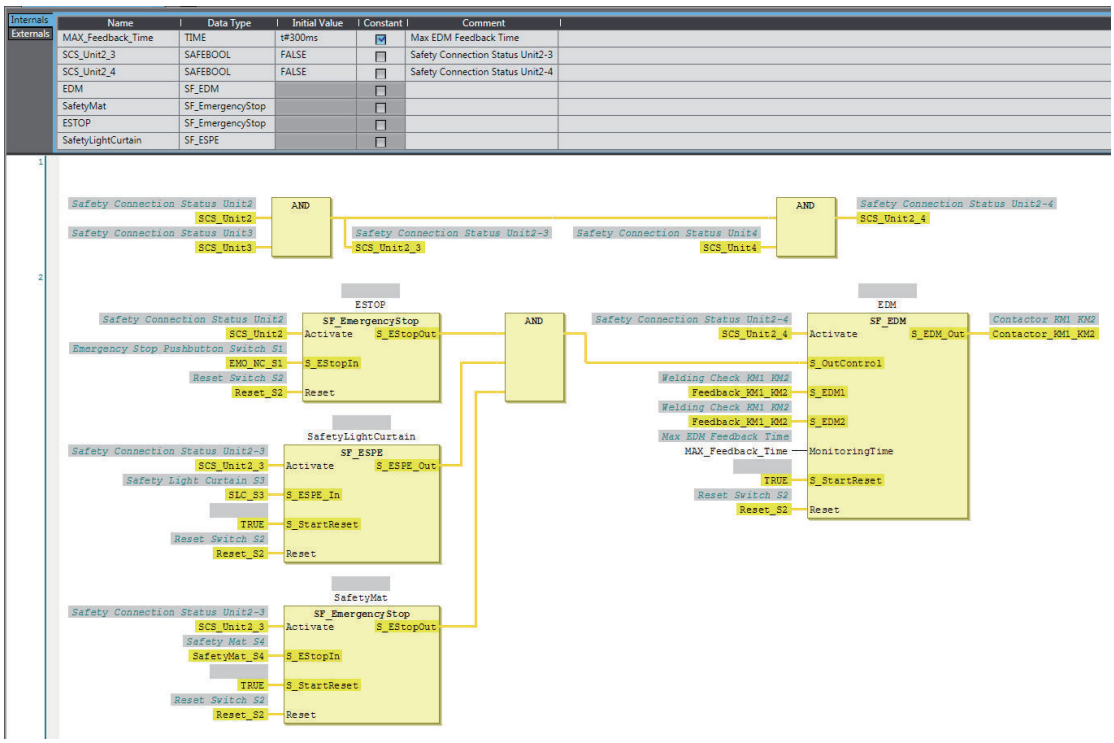
**Node1/Unit4 : NX-SOD400 (N4 : Instance2)**

| External Device | Channel | Comment |
|---|---|---|
| Dual Output with Test Pulse | So 0 | 2 Safety Relays w/ Welding Check |
| | So 1 | |
| | So 2 | |
| | So 3 | |

## ● I/O Map Settings

| Position | Port | R/W | Data Type | Variable | Variable Comment | Variable Type |
|---|---|---|---|---|---|---|
| | ▼ EtherCAT Network | | | | | |
| EtherCAT Master | Master | | | | | |
| Node1/Unit2 | ▼ NX-SIH400 | | | | | |
| | ▼ Safety Inputs and Status | | | | | |
| | Si00 Logical Value | R | SAFEBOOL | EMO_NC_S1 | Emergency Stop Pushbutton Switch S1 | Global Variables |
| | Si01 Logical Value | R | SAFEBOOL | | | |
| | Si02 Logical Value | R | SAFEBOOL | Reset_S2 | Reset Switch S2 | Global Variables |
| | Si03 Logical Value | R | SAFEBOOL | Feedback_KM1_KM2 | Welding Check KM1_KM2 | Global Variables |
| | Safety Connection Status | R | SAFEBOOL | SCS_Unit2 | Safety Connection Status Unit2 | Global Variables |
| | Safety Input Terminal Status | R | SAFEBOOL | | | |
| Node1/Unit3 | ▼ NX-SIH400 | | | | | |
| | ▼ Safety Inputs and Status | | | | | |
| | Si00 Logical Value | R | SAFEBOOL | SLC_S3 | Safety Light Curtain S3 | Global Variables |
| | Si01 Logical Value | R | SAFEBOOL | | | |
| | Si02 Logical Value | R | SAFEBOOL | SafetyMat_S4 | Safety Mat S4 | Global Variables |
| | Si03 Logical Value | R | SAFEBOOL | | | |
| | Safety Connection Status | R | SAFEBOOL | SCS_Unit3 | Safety Connection Status Unit3 | Global Variables |
| | Safety Input Terminal Status | R | SAFEBOOL | | | |
| Node1/Unit4 | ▼ NX-SOD400 | | | | | |
| | ▼ Status | | | | | |
| | Safety Connection Status | R | SAFEBOOL | SCS_Unit4 | Safety Connection Status Unit4 | Global Variables |
| | Safety Output Terminal Status | R | SAFEBOOL | | | |
| | ▼ Safety Outputs | | | | | |
| | So00 Output Value | W | SAFEBOOL | Contactor_KM1_KM2 | Contactor KM1_KM2 | Global Variables |
| | So01 Output Value | W | SAFEBOOL | | | |
| | So02 Output Value | W | SAFEBOOL | | | |
| | So03 Output Value | W | SAFEBOOL | | | |

# Program




### Precautions for Safe Use

- Test the functionality every six months to detect welded contactor contacts.
- The customer is responsible for attaining conformance of the entire system to standards.
- To detect electrical and mechanical failures, use a combination of redundant semiconductor output contacts and redundant mechanical output devices.

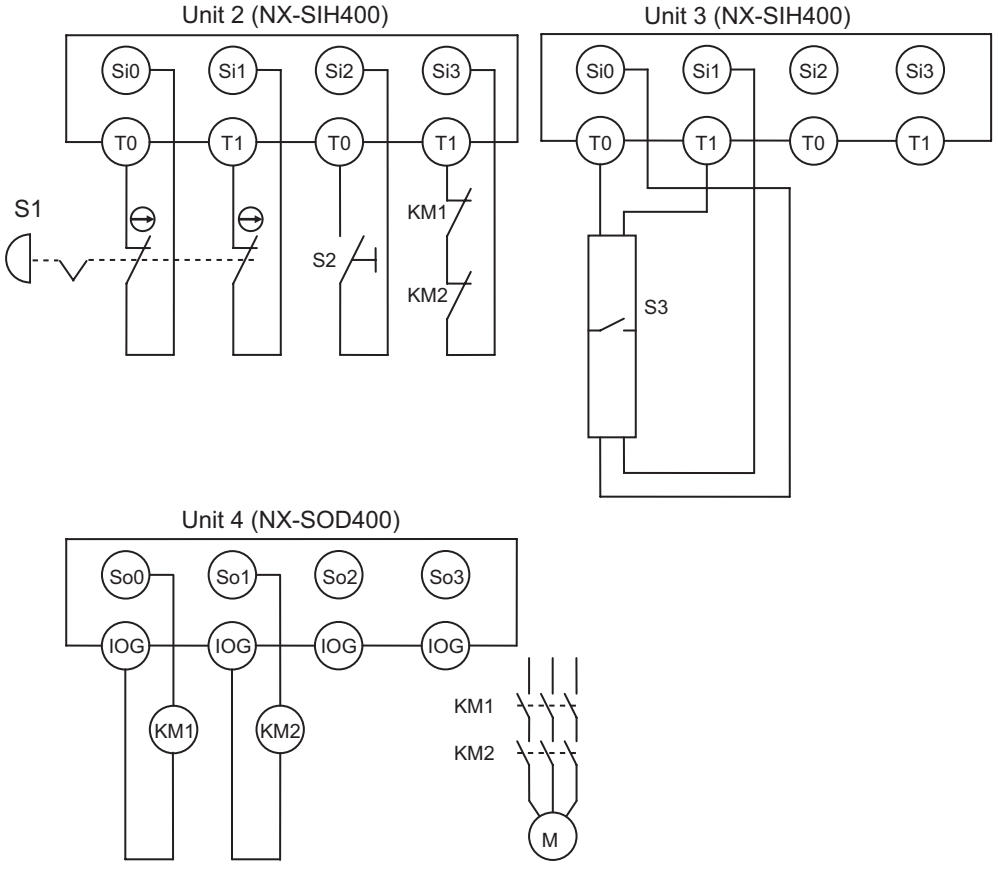
# A-3-10 Safety Edges

## Application Overview

| Safety category/PL | Safety device | Stop category | Reset |
|---|---|---|---|
| Equivalent to 3/PLd (Safety Edge) | • Emergency stop pushbutton<br>• Safety edge (2-wire cable on both sides) | 0 | Manual |

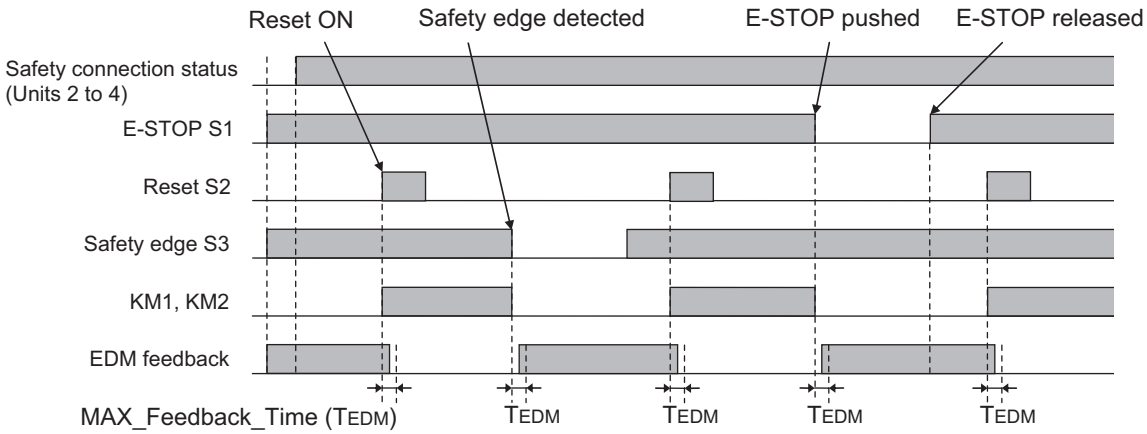
Motor M stops when emergency stop pushbutton S1 is pressed or when edge sensor detects a contact with persons or objects.

## Wiring




S1: Emergency stop pushbutton
S2: Reset switch
S3: Safety edge
KM1, KM2: Contactors
M: Motor

## Timing Chart




## Safety I/O Terminal & I/O Map Setting

### ● Safety I/O Terminal Settings

**Node1/Unit2 : NX-SIH400 (N2 : Instance0)**

| External Device | Channel | Discrepancy | On-Off | Off-On | Test Source | Comment |
|---|---|---|---|---|---|---|
| Mechanical Contact for Dual Channel Equivalent | Si 0 | 500ms | 0ms | 0ms | T0 | Emergency Stop Pushbutton Switch(2NC) |
| | Si 1 | 500ms | 0ms | 0ms | T1 | |
| Mechanical Contact For Single Channel | Si 2 | 0ms | 0ms | 0ms | T0 | Reset Switch |
| Mechanical Contact For Single Channel | Si 3 | 0ms | 0ms | 0ms | T1 | EDM(Contact Welding Detection) |

**Node1/Unit3 : NX-SIH400 (N3 : Instance1)**

| External Device | Channel | Discrepancy | On-Off | Off-On | Test Source | Comment |
|---|---|---|---|---|---|---|
| Safety Mat/Safety Edge | Si 0 | 0ms | 0ms | 0ms | T0 | Safety Edge |
| | Si 1 | 0ms | 0ms | 0ms | T1 | |
| | Si 2 | | | | | |
| | Si 3 | | | | | |

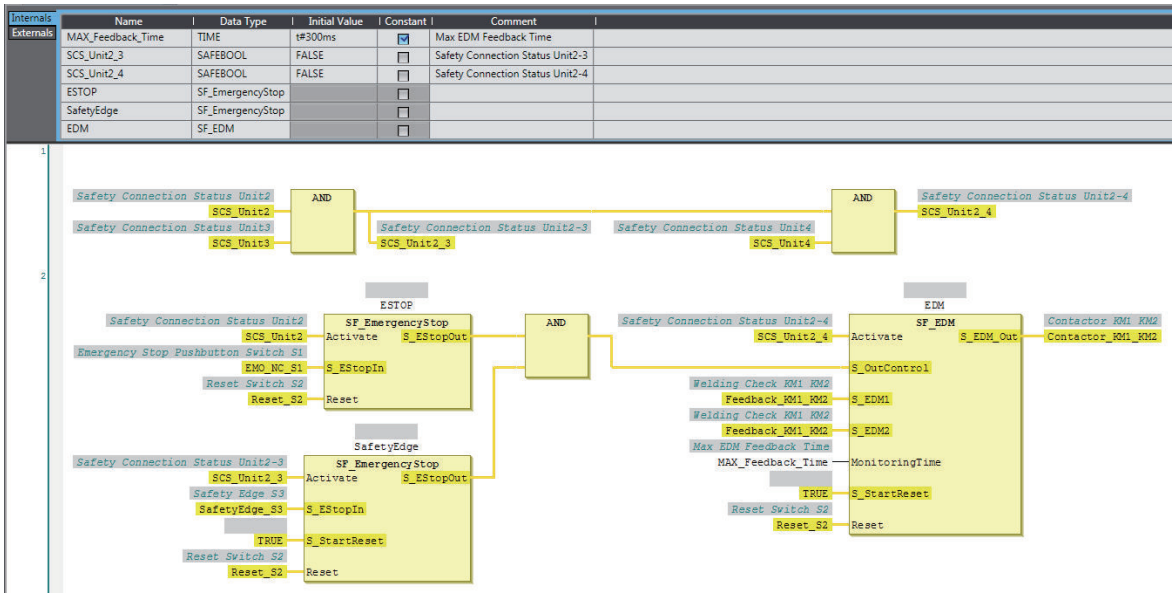
**Node1/Unit4 : NX-SOD400 (N4 : Instance2)**

| External Device | Channel | Comment |
|---|---|---|
| Dual Output with Test Pulse | So 0 | 2 Safety Relays w/ Welding Check |
| | So 1 | |
| | So 2 | |
| | So 3 | |

## ● I/O Map Settings

| Position | Port | R/W | Data Type | Variable | Variable Comment | Variable Type |
|---|---|---|---|---|---|---|
| | EtherCAT Network | | | | | |
| EtherCAT Master | Master | | | | | |
| Node1/Unit2 | NX-SIH400 | | | | | |
| | Safety Inputs and Status | | | | | |
| | Si00 Logical Value | R | SAFEBOOL | EMO_NC_S1 | Emergency Stop Pushbutton Switch S1 | Global Variables |
| | Si01 Logical Value | R | SAFEBOOL | | | |
| | Si02 Logical Value | R | SAFEBOOL | Reset_S2 | Reset Switch S2 | Global Variables |
| | Si03 Logical Value | R | SAFEBOOL | Feedback_KM1_KM2 | Welding Check KM1_KM2 | Global Variables |
| | Safety Connection Status | R | SAFEBOOL | SCS_Unit2 | Safety Connection Status Unit2 | Global Variables |
| | Safety Input Terminal Status | R | SAFEBOOL | | | |
| Node1/Unit3 | NX-SOD400 | | | | | |
| | Status | | | | | |
| | Safety Connection Status | R | SAFEBOOL | SCS_Unit3 | Safety Connection Status Unit3 | Global Variables |
| | Safety Output Terminal Status | R | SAFEBOOL | | | |
| | Safety Outputs | | | | | |
| | So00 Output Value | W | SAFEBOOL | Contactor_KM1_KM2 | Contactor KM1_KM2 | Global Variables |
| | So01 Output Value | W | SAFEBOOL | | | |
| | So02 Output Value | W | SAFEBOOL | | | |
| | So03 Output Value | W | SAFEBOOL | | | |

## Program




### Precautions for Safe Use

- Test the functionality every six months to detect welded contactor contacts.
- The customer is responsible for attaining conformance of the entire system to standards.
- To detect electrical and mechanical failures, use a combination of redundant semiconductor output contacts and redundant mechanical output devices.

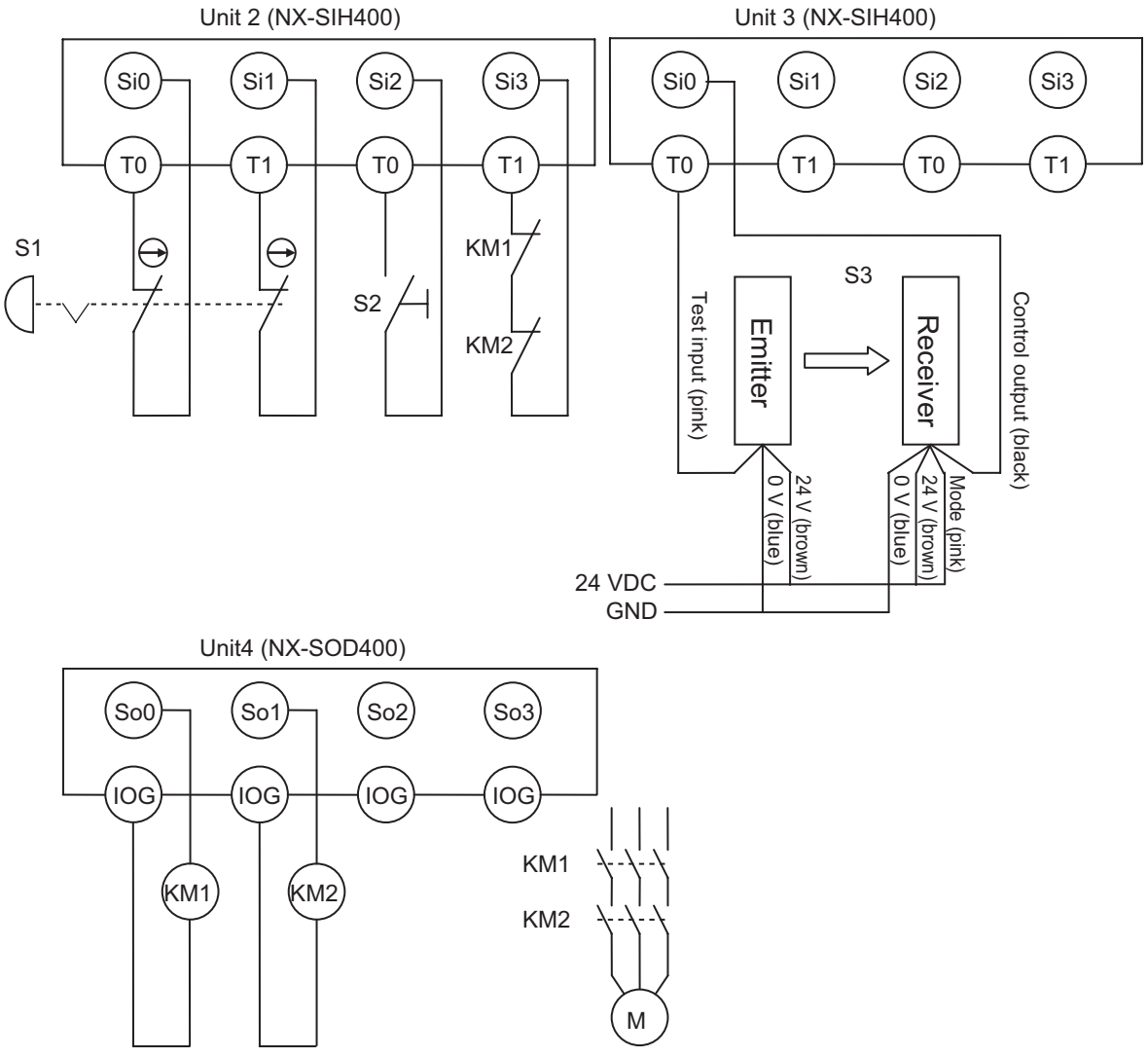
# A-3-11 Single Beam Safety Sensor

## Application Overview

| Safety category/PL | Safety device | Stop category | Reset |
|---|---|---|---|
| Equivalent to 2/PLc (Single Beam Safety Sensor) | • Emergency stop pushbutton<br>• Single beam safety sensor | 0 | Manual |

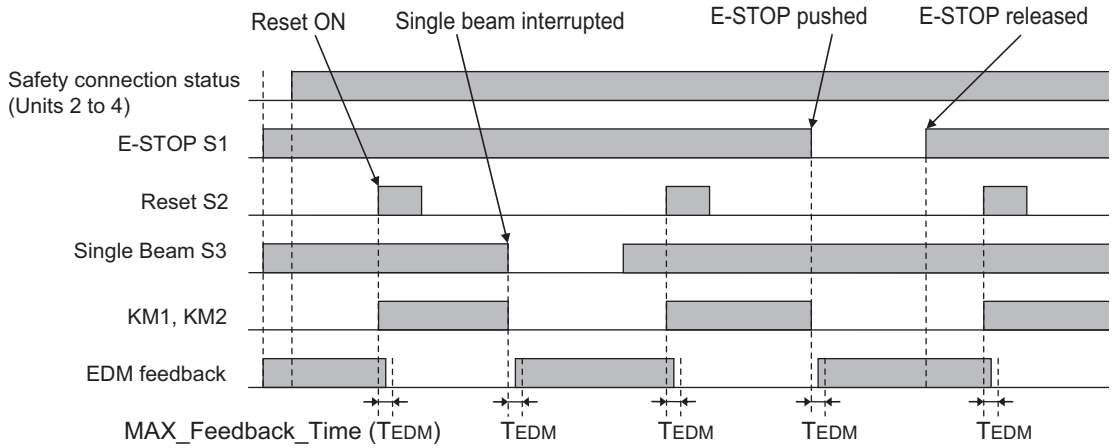
Motor M stops when emergency stop pushbutton S1 is pressed.
Motor M stops when the light in the single beam safety sensor is interrupted.

## Wiring




S1: Emergency stop pushbutton
S2: Reset switch
S3: Single beam safety sensor
KM1, KM2: Contactors
M: Motor

## Timing Chart




## Safety I/O Terminal & I/O Map Setting

### ● Safety I/O Terminal Settings

**Node1/Unit2 : NX-SIH400 (N2 : Instance0)**

| External Device | Channel | Discrepancy | On-Off | Off-On | Test Source | Comment |
|---|---|---|---|---|---|---|
| Mechanical Contact for Dual Channel Equivalent | Si 0 | 500ms | 0ms | 0ms | T0 | Emergency Stop Pushbutton Switch(2NC) |
| | Si 1 | 500ms | 0ms | 0ms | T1 | |
| Mechanical Contact For Single Channel | Si 2 | 0ms | 0ms | 0ms | T0 | Reset Switch |
| Mechanical Contact For Single Channel | Si 3 | 0ms | 0ms | 0ms | T1 | EDM(Contact Welding Detection) |

**Node1/Unit3 : NX-SIH400 (N3 : Instance1)**

| External Device | Channel | Discrepancy | On-Off | Off-On | Test Source | Comment |
|---|---|---|---|---|---|---|
| Single Beam Safety Sensor | Si 0 | 0ms | 0ms | 0ms | T0 | Single Beam Safety Sensor |
| | Si 1 | | | | | |
| | Si 2 | | | | | |
| | Si 3 | | | | | |

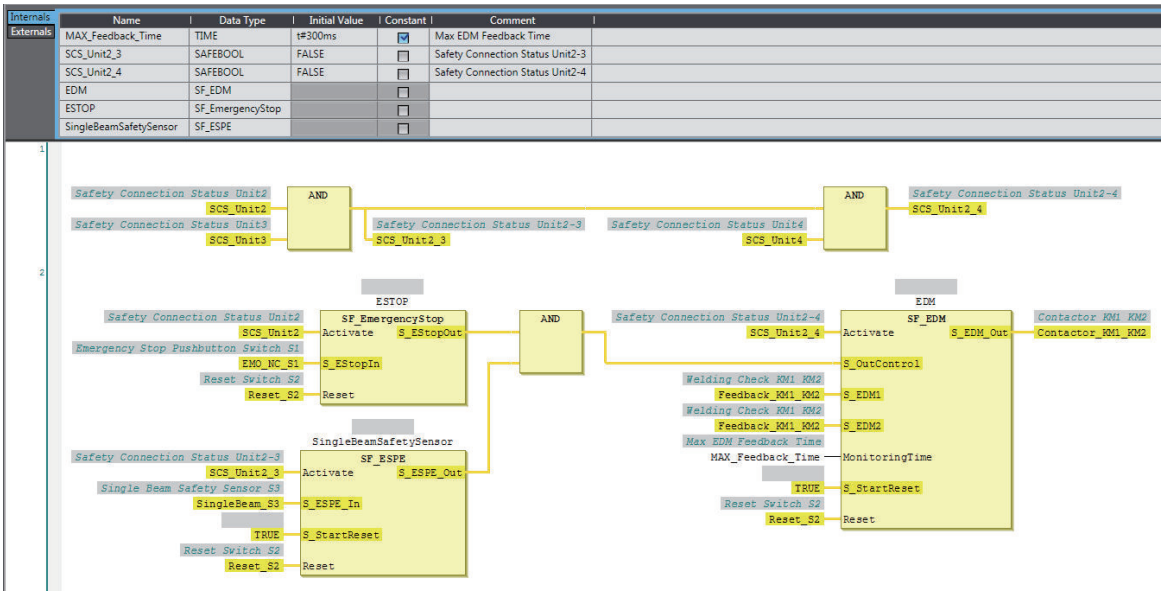
**Node1/Unit4 : NX-SOD400 (N4 : Instance2)**

| External Device | Channel | Comment |
|---|---|---|
| Dual Output with Test Pulse | So 0 | 2 Safety Relays w/ Welding Check |
| | So 1 | |
| | So 2 | |
| | So 3 | |

## ● I/O Map Settings

| Position | Port | R/W | Data Type | Variable | Variable Comment | Variable Type |
|---|---|---|---|---|---|---|
| | ▼ EtherCAT Network | | | | | |
| EtherCAT Master | Master | | | | | |
| Node1/Unit2 | ▼ NX-SIH400 | | | | | |
| | ▼ Safety Inputs and Status | | | | | |
| | Si00 Logical Value | R | SAFEBOOL | EMO_NC_S1 | Emergency Stop Pushbutton Switch S1 | Global Variables |
| | Si01 Logical Value | R | SAFEBOOL | | | |
| | Si02 Logical Value | R | SAFEBOOL | Reset_S2 | Reset Switch S2 | Global Variables |
| | Si03 Logical Value | R | SAFEBOOL | Feedback_KM1_KM2 | Welding Check KM1_KM2 | Global Variables |
| | Safety Connection Status | R | SAFEBOOL | SCS_Unit2 | Safety Connection Status Unit2 | Global Variables |
| | Safety Input Terminal Status | R | SAFEBOOL | | | |
| Node1/Unit3 | ▼ NX-SIH400 | | | | | |
| | ▼ Safety Inputs and Status | | | | | |
| | Si00 Logical Value | R | SAFEBOOL | SingleBeam_S3 | Single Beam Safety Sensor S3 | Global Variables |
| | Si01 Logical Value | R | SAFEBOOL | | | |
| | Si02 Logical Value | R | SAFEBOOL | | | |
| | Si03 Logical Value | R | SAFEBOOL | | | |
| | Safety Connection Status | R | SAFEBOOL | SCS_Unit3 | Safety Connection Status Unit3 | Global Variables |
| | Safety Input Terminal Status | R | SAFEBOOL | | | |
| Node1/Unit4 | ▼ NX-SOD400 | | | | | |
| | ▼ Status | | | | | |
| | Safety Connection Status | R | SAFEBOOL | SCS_Unit4 | Safety Connection Status Unit4 | Global Variables |
| | Safety Output Terminal Status | R | SAFEBOOL | | | |
| | ▼ Safety Outputs | | | | | |
| | So00 Output Value | W | SAFEBOOL | Contactor_KM1_KM2 | Contactor KM1_KM2 | Global Variables |
| | So01 Output Value | W | SAFEBOOL | | | |
| | So02 Output Value | W | SAFEBOOL | | | |
| | So03 Output Value | W | SAFEBOOL | | | |

# Program




### Precautions for Safe Use

- Test the functionality every six months to detect welded contactor contacts.
- The customer is responsible for attaining conformance of the entire system to standards.
- To detect electrical and mechanical failures, use a combination of redundant semiconductor output contacts and redundant mechanical output devices.

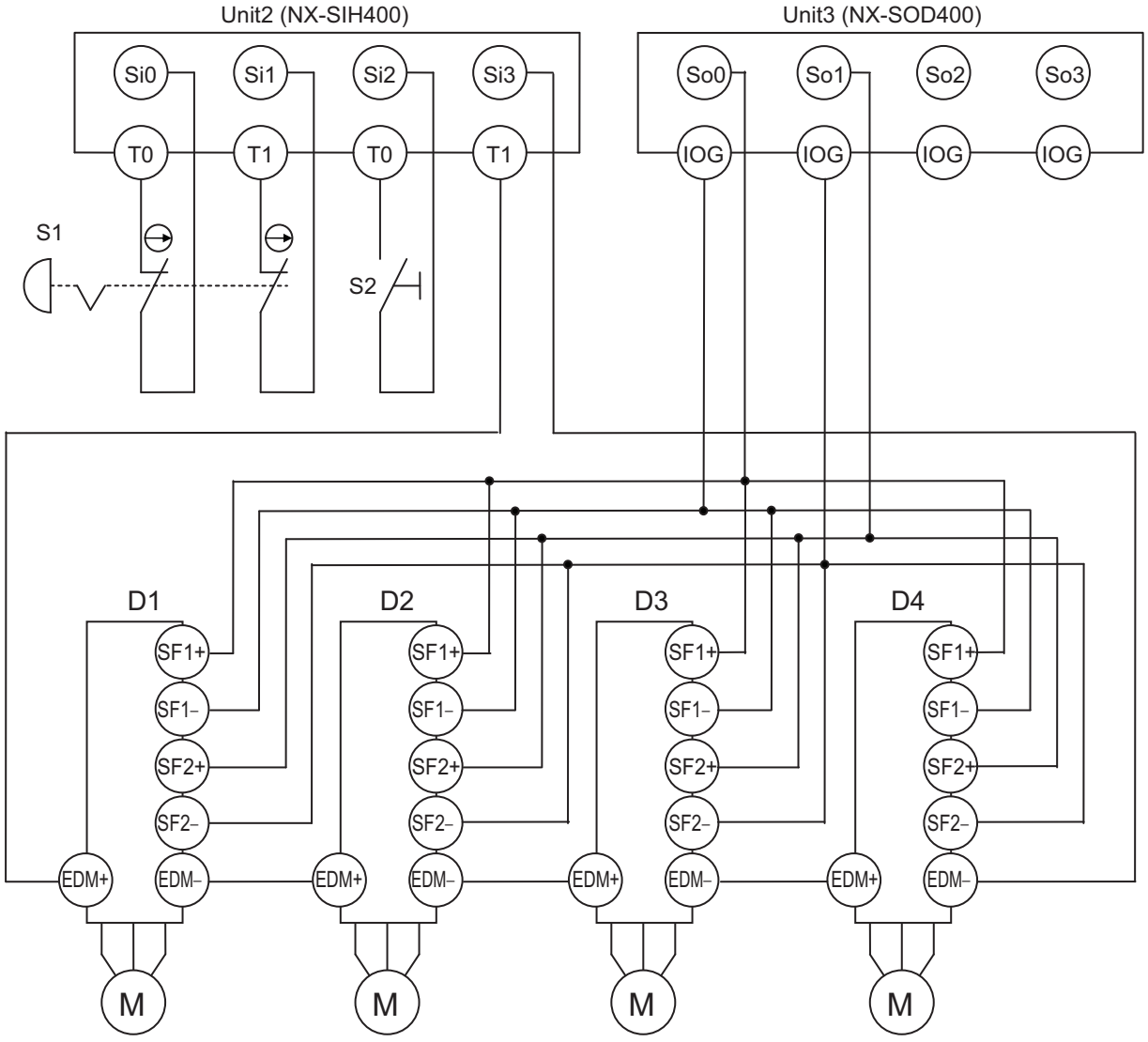
## A-3-12 Multi-axis Servo Drives (STO)

## Application Overview

| Safety category/PL | Safety device | Stop category | Reset |
|---|---|---|---|
| Equivalent to 3/PLd | Emergency stop pushbutton | 0 (STO) | Manual |

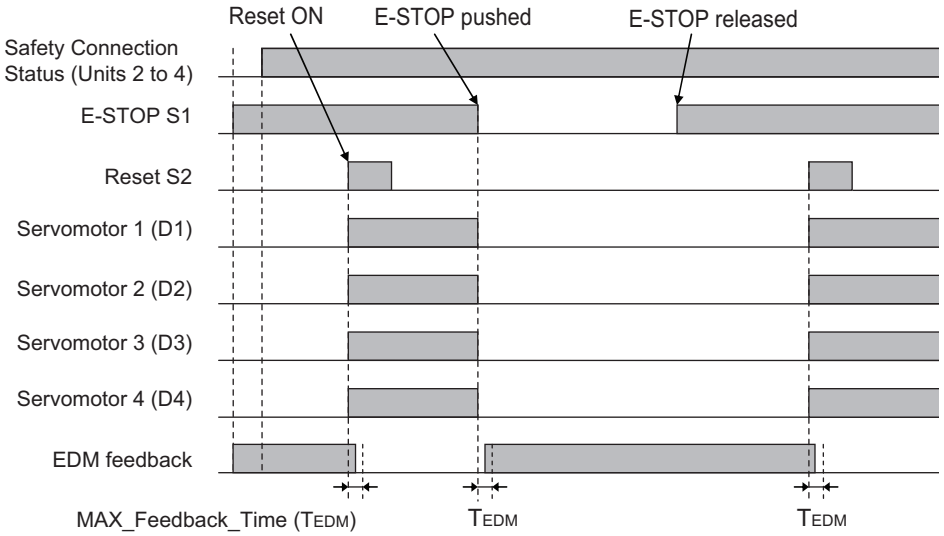
All servomotors M stop when emergency stop pushbutton S1 is pressed.

## Wiring




S1: Emergency stop pushbutton
S2: Reset switch
D1, D2, D3, and D4: Servo drives
M: Motor

## Timing Chart




## Safety I/O Terminal & I/O Map Setting

### ● Safety I/O Terminal Settings

**Node1/Unit2 : NX-SIH400 (N2 : Instance0)**

| External Device | Channel | Discrepancy | On-Off | Off-On | Test Source | Comment |
|---|---|---|---|---|---|---|
| Mechanical Contact for Dual Channel Equivalent | Si 0 | 500ms | 0ms | 0ms | T0 | Emergency Stop Pushbutton(2NC) |
| | Si 1 | 500ms | 0ms | 0ms | T1 | |
| Mechanical Contact For Single Channel | Si 2 | 0ms | 0ms | 0ms | T0 | Reset Switch |
| Mechanical Contact For Single Channel | Si 3 | 0ms | 0ms | 0ms | T1 | EDM(Contact Welding Detection) |

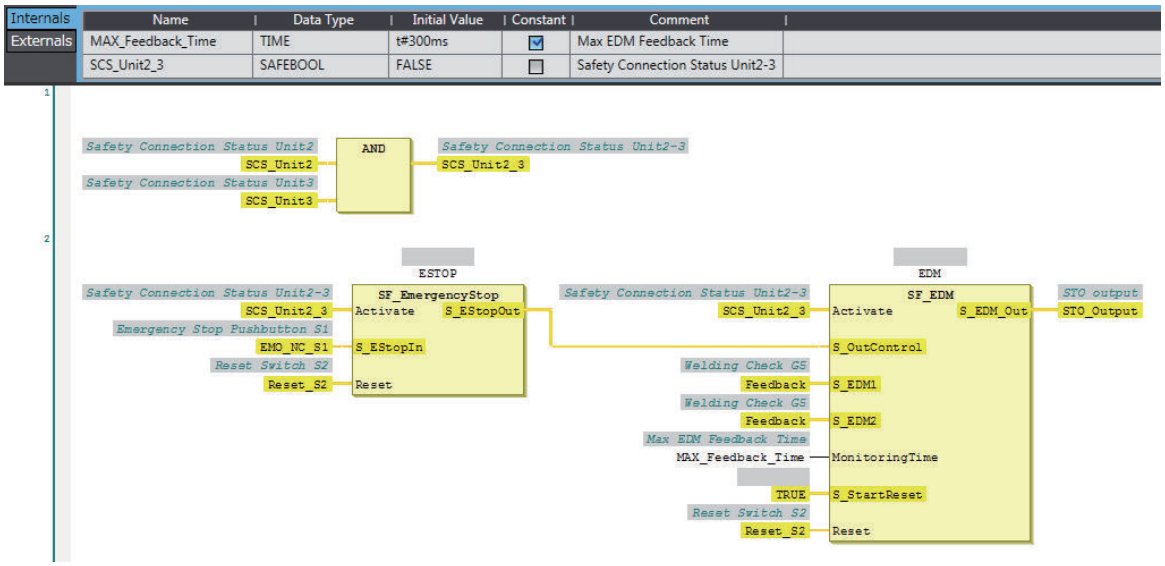
**Node1/Unit3 : NX-SOD400 (N3 : Instance1)**

| External Device | Channel | Comment |
|---|---|---|
| Dual Output with Test Pulse | So 0 | D1, D2, D3, D4 |
| | So 1 | |
| | So 2 | |
| | So 3 | |

### ● I/O Map Settings

| Position | Port | R/W | Data Type | Variable | Variable Comment | Variable Type |
|---|---|---|---|---|---|---|
| | ▼ EtherCAT Network | | | | | |
| EtherCAT Master | Master | | | | | |
| Node1/Unit2 | ▼ NX-SIH400 | | | | | |
| | ▼Safety Inputs and Status | | | | | |
| | Si00 Logical Value | R | SAFEBOOL | EMO_NC_S1 | Emergency Stop Pushbutton(2NC) | Global Variables |
| | Si01 Logical Value | R | SAFEBOOL | | | |
| | Si02 Logical Value | R | SAFEBOOL | Reset_S2 | Reset Switch | Global Variables |
| | Si03 Logical Value | R | SAFEBOOL | Feedback | EDM(Contact Welding Detection) | Global Variables |
| | Safety Connection Status | R | SAFEBOOL | SCS_Unit2 | Safety Connection Status Unit2 | Global Variables |
| | Safety Input Terminal Status | R | SAFEBOOL | | | |
| Node1/Unit3 | ▼ NX-SOD400 | | | | | |
| | ▼Status | | | | | |
| | Safety Connection Status | R | SAFEBOOL | SCS_Unit3 | Safety Connection Status Unit3 | Global Variables |
| | Safety Output Terminal Status | R | SAFEBOOL | | | |
| | ▼Safety Outputs | | | | | |
| | So00 Output Value | W | SAFEBOOL | STO_Output | D1, D2, D3, D4 | Global Variables |
| | So01 Output Value | W | SAFEBOOL | | | |
| | So02 Output Value | W | SAFEBOOL | | | |
| | So03 Output Value | W | SAFEBOOL | | | |

## Program




### Precautions for Safe Use

- Test the functionality every six months to detect faults in the safety input circuits and EDM output circuits of the servo drives.
- The customer is responsible for attaining conformance of the entire system to standards.
- To detect electrical and mechanical failures, use a combination of redundant semiconductor output contacts and redundant safety inputs for the servo drives.

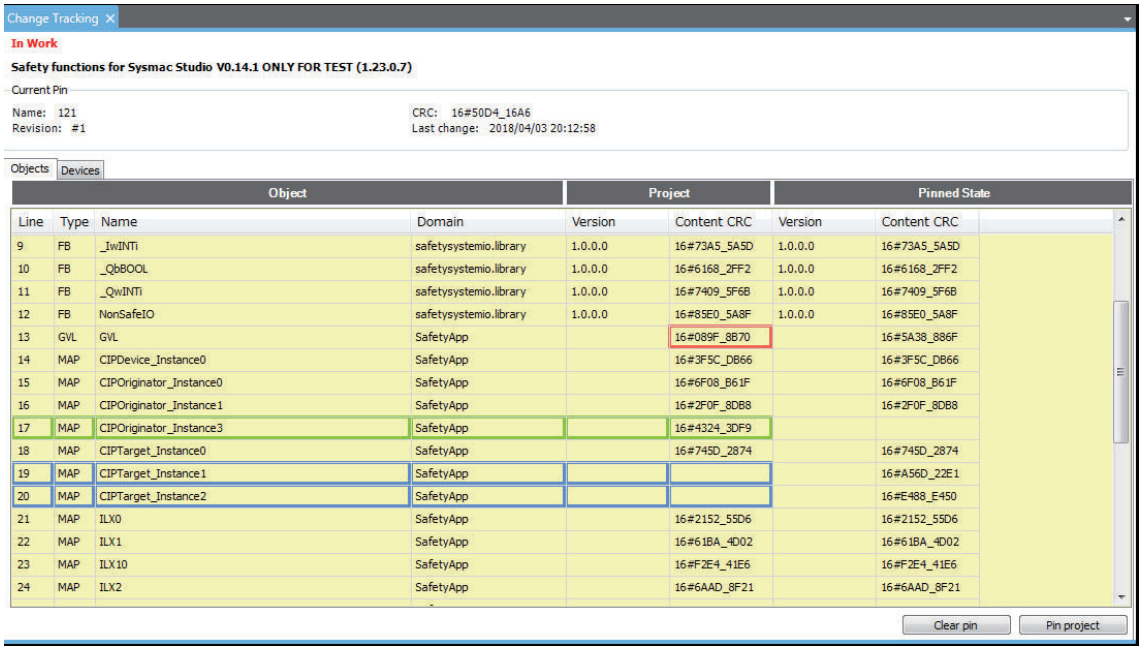
# A-4 Change Tracking

## What is Change Tracking?

Change Tracking is used to display and manage changes in the safety application data after the pin is created.
Tracking information is property information in safety application data. The storage of the safety application data settings at a given point in time is referred to as "creating a pin".
It is primarily used for version management after the safety application data is debugged.

## Change Tracking Procedure and Contents

**1** From the Safety CPU Unit Setup and Programming View, select **Change Tracking** from the **Project** Menu.
The **Change Tracking** Tab Page is displayed.




The outer frames of the cells for each item are displayed in the following colors if a pin has not been created or a change was made after it was created.

| Color | Description |
|---|---|
| Green | • The pin has not been created yet.<br>• Items were added after the pin was created. |
| Red | • Changes were made after the pin was created. |
| Blue | • The pin was deleted after it was created. |

### ● Description of Type

The type of configuration data is displayed in the **Type** column.

**Objects** tab

| Type Name | Description |
|---|---|
| APP | Safety Application |
| TASK | Safety Task |
| PRG | Programs |
| FB | Function Blocks |
| GVL | Global Variables |
| MAP | I/O Map and exposed variables |
| CNF | Safety process data communications |

**Devices** tab

| Type Name | Description |
|---|---|
| SAFEPLC | Safety CPU Unit |
| SAFEDEV | Safety I/O Unit |
| XVARDEF | Exposed variables |



## Pin Operations

This section describes the procedures to create and delete pins, and the jump function.

### ● Creating Pins

When you click the **Pin project** Button, a dialog box to enter the current status name is displayed. The name that you set is shown as the pin information in the upper part of the **Change Tracking** Tab Page.

If the data changes from the data that is in effect at this point, the contents of the change are displayed in the tab page.

### ● Deleting a Pin

When you click the **Clear Pin** Button, the status that you created with the **Pin project** Button is deleted.

### ● Jump Function

When you double-click information on the **Change Tracking** Tab Page, the global variable table, FBD editor, or other corresponding tab page is displayed.

# A-5 Safety CPU Unit Status

The Safety CPU Unit status gives the operating status of the Safety CPU Unit. When a Safety CPU Unit is placed on the NX bus of the Communications Coupler Unit, the status is displayed as an I/O port in the I/O map of the Communications Coupler Unit. Also, when a Safety CPU Unit is placed on the NX bus of the NX-series CPU Unit, the status is displayed as an I/O port in the I/O map of the CPU rack.
If you set a device variable for the I/O port, you can monitor the status of the Safety CPU Unit from the NJ/NX-series CPU Unit.

## A-5-1 NX-SL3300/NX-SL3500

| I/O port | | Description | Condition | R/W | Data type |
|---|---|---|---|---|---|
| Safety CPU Status | | Status monitoring data for the Safety CPU Unit | This is a WORD variable that contains the following status. | R | WORD |
| D00 | Normal Operating | Safety programs operating with no errors.<br>All safety master connections established. | This variable is TRUE when all of the following conditions are met. If even one condition is not met, it is FALSE.<br>Conditions<br>• The safety programs are in RUN status (RUN mode or DEBUG mode (RUN)).<br>• No event with a level of minor fault or higher currently exists for the safety programs.<br>• All FSoE connections are established. | R | BOOL |
| D01 | Program Operating | Safety programs operating | This variable is TRUE when the following condition is met. If the condition is not met, it is FALSE.<br>Condition<br>• The safety programs are in RUN status (RUN mode or DEBUG mode (RUN)). | R | BOOL |
| D02 | Program No Fault | No event with a level of minor fault or higher currently exists for the safety programs. | This variable is TRUE when the following condition is met. If the condition is not met, it is FALSE.<br>Condition<br>• No event with a level of minor fault or higher currently exists for the safety programs. | R | BOOL |
| D03 | Safety Master Connection Status | All safety master connections established. | This variable is TRUE when the following condition is met. If the condition is not met, it is FALSE.<br>Conditions<br>• All FSoE connections are established. | R | BOOL |

## A-5-2 NX-SL5500/NX-SL5700

| I/O port | Description | Condition | R/W | Data type |
|---|---|---|---|---|
| Safety CPU Status | Status monitoring data for the Safety CPU Unit | This is a UINT variable that contains the following status. | R | UINT |

| I/O port | | Description | Condition | R/W | Data type |
|---|---|---|---|---|---|
| D00 | Normal Operating | Safety programs operating with no errors.<br>All safety master connections established. | This variable is TRUE when all of the following conditions are met. If even one condition is not met, it is FALSE.<br>Condition<br>• The safety programs are in RUN status (RUN mode or DEBUG mode (RUN)).<br>• No event with a level of minor fault or higher currently exists for the safety programs.<br>• All FSoE connections are established.<br>• All the CIP Safety Originator connections are established.<br>• No error is present in the CIP Safety Target connections. | R | BOOL |
| D01 | Program Operating | Safety programs operating | This variable is TRUE when the following condition is met. If the condition is not met, it is FALSE.<br>Condition<br>• The safety programs are in RUN status (RUN mode or DEBUG mode (RUN)). | R | BOOL |
| D02 | Program No Fault | No event with a level of minor fault or higher currently exists for the safety programs. | This variable is TRUE when the following condition is met. If the condition is not met, it is FALSE.<br>Condition<br>• No event with a level of minor fault or higher currently exists for the safety programs. | R | BOOL |
| D03 | Safety Master Connection Status | All safety master connections established. | This variable is TRUE when all of the following conditions are met. If even one condition is not met, it is FALSE.<br>Condition<br>• All FSoE connections are established.<br>• All the CIP Safety Originator connections are established. | R | BOOL |
| D04 | FSoE Master Connection Status | All the FSoE safety master connections established. | This variable is TRUE when the following condition is met. If the condition is not met, it is FALSE.<br>Condition<br>• All FSoE connections are established. | R | BOOL |
| D05 | CIP Safety Originator Connection Status | All the CIP Safety Originator connections established. | This variable is TRUE when the following condition is met. If the condition is not met, it is FALSE.<br>Condition<br>• All the CIP Safety Originator connections are established. | R | BOOL |
| D06 | CIP Safety Target Connection Status | No error in the entire CIP Safety Target connections | This variable is TRUE when the following condition is met. If the condition is not met, it is FALSE.<br>Condition<br>• No error is present in the entire CIP Safety Target connections. | R | BOOL |

# A-6 I/O Ports of Safety I/O Units

The I/O ports of the Safety I/O Units are displayed on the I/O map of the Communications Coupler Unit if you place the Safety I/O Units on the Communications Coupler Unit. When you place the Safety I/O Units on the NX bus of the NX-series CPU Unit, their I/O ports are displayed on the I/O map of the CPU Rack.
The names of the I/O ports that correspond to the data in the I/O Map of the Safety CPU Unit are given in the "Corresponding port name" Column. "Same" means that the same name is used.

## A-6-1 NX-SIH400 Safety Input Unit

| Port | Data type | R/W | Name | Description | Default | Corresponding port name |
|---|---|---|---|---|---|---|
| Standard Input 1st Byte | BYTE | R | Standard Input 1st Byte | --- | 00 hex | --- |
| Si00 Logical Value | BOOL | R | Si00 Logical Value | Gives the status of safety input terminal Si00.<br>0: OFF, 1: ON | 0 | Same |
| Si01 Logical Value | BOOL | R | Si01 Logical Value | Gives the status of safety input terminal Si01.<br>0: OFF, 1: ON | 0 | Same |
| Si02 Logical Value | BOOL | R | Si02 Logical Value | Gives the status of safety input terminal Si02.<br>0: OFF, 1: ON | 0 | Same |
| Si03 Logical Value | BOOL | R | Si03 Logical Value | Gives the status of safety input terminal Si03.<br>0: OFF, 1: ON | 0 | Same |
| Safety Connection Status | BOOL | R | Safety Connection Status | This flag indicates when a safety connection is active. Use it for an input to the Activate terminal on a safety FB or for safety connection/disconnection applications. | 0 | Same |
| Safety Input Terminal Status | BOOL | R | Safety Input Terminal Status | This flag indicates the status of the safety input terminals.<br>0: An error has occurred on one of the safety input terminals.<br>1: All of the safety input terminals are normal (no errors). | 0 | Same |
| Unit Normal Status | BOOL | R | Unit Normal Status | This flag indicates the status of the Unit.<br>0: An error has occurred.<br>1: Normal (no errors) | 0 | --- |
| IO Power Supply Error Flag | BOOL | R | IO Power Supply Error Flag | This flag indicates the status of the I/O power supply voltage.<br>0: The I/O power supply voltage is normal.<br>1: The I/O power supply voltage is incorrect or the I/O power supply is OFF. | 0 | --- |

| Port | Data type | R/W | Name | Description | Default | Corresponding port name |
|---|---|---|---|---|---|---|
| Standard Input 2nd Byte | BYTE | R | Standard Input 2nd Byte | --- | 00 hex | --- |
| Si00 Status | BOOL | R | Si00 Status | Gives the status of safety input terminal 00.<br>0: Error<br>1: No error | 0 | --- |
| Si01 Status | BOOL | R | Si01 Status | Gives the status of safety input terminal 01.<br>0: Error<br>1: No error | 0 | --- |
| Si02 Status | BOOL | R | Si02 Status | Gives the status of safety input terminal 02.<br>0: Error<br>1: No error | 0 | --- |
| Si03 Status | BOOL | R | Si03 Status | Gives the status of safety input terminal 03.<br>0: Error<br>1: No error | 0 | --- |

## A-6-2 NX-SID800 Safety Input Unit

| Port | Data type | R/W | Name | Description | Default | Corresponding port name |
|---|---|---|---|---|---|---|
| Standard Input 1st Word | WORD | R | Standard Input 1st Word | --- | 0000 hex | --- |
| Si00 Logical Value | BOOL | R | Si00 Logical Value | Gives the status of safety input terminal Si00.<br>0: OFF, 1: ON | 0 | Same |
| Si01 Logical Value | BOOL | R | Si01 Logical Value | Gives the status of safety input terminal Si01.<br>0: OFF, 1: ON | 0 | Same |
| Si02 Logical Value | BOOL | R | Si02 Logical Value | Gives the status of safety input terminal Si02.<br>0: OFF, 1: ON | 0 | Same |
| Si03 Logical Value | BOOL | R | Si03 Logical Value | Gives the status of safety input terminal Si03.<br>0: OFF, 1: ON | 0 | Same |
| Si04 Logical Value | BOOL | R | Si04 Logical Value | Gives the status of safety input terminal Si04.<br>0: OFF, 1: ON | 0 | Same |
| Si05 Logical Value | BOOL | R | Si05 Logical Value | Gives the status of safety input terminal Si05.<br>0: OFF, 1: ON | 0 | Same |
| Si06 Logical Value | BOOL | R | Si06 Logical Value | Gives the status of safety input terminal Si06.<br>0: OFF, 1: ON | 0 | Same |
| Si07 Logical Value | BOOL | R | Si07 Logical Value | Gives the status of safety input terminal Si07.<br>0: OFF, 1: ON | 0 | Same |
| Safety Connection Status | BOOL | R | Safety Connection Status | This flag indicates when a safety connection is active. Use it for an input to the Activate terminal on a safety FB or for safety connection/disconnection applications. | 0 | Same |
| Safety Input Terminal Status | BOOL | R | Safety Input Terminal Status | This flag indicates the status of the safety input terminals.<br>0: An error has occurred on one of the safety input terminals.<br>1: All of the safety input terminals are normal (no errors). | 0 | Same |
| Unit Normal Status | BOOL | R | Unit Normal Status | This flag indicates the status of the Unit.<br>0: An error has occurred.<br>1: Normal (no errors) | 0 | --- |

| Port | Data type | R/W | Name | Description | Default | Corresponding port name |
|---|---|---|---|---|---|---|
| IO Power Supply Error Flag | BOOL | R | IO Power Supply Error Flag | This flag indicates the status of the I/O power supply voltage.<br>0: The I/O power supply voltage is normal.<br>1: The I/O power supply voltage is incorrect or the I/O power supply is OFF. | 0 | --- |
| Standard Input 2nd Byte | BYTE | R | Standard Input 2nd Byte | --- | 00 hex | --- |
| Si00 Status | BOOL | R | Si00 Status | Gives the status of safety input terminal 00.<br>0: Error<br>1: No error | 0 | --- |
| Si01 Status | BOOL | R | Si01 Status | Gives the status of safety input terminal 01.<br>0: Error<br>1: No error | 0 | --- |
| Si02 Status | BOOL | R | Si02 Status | Gives the status of safety input terminal 02.<br>0: Error<br>1: No error | 0 | --- |
| Si03 Status | BOOL | R | Si03 Status | Gives the status of safety input terminal 03.<br>0: Error<br>1: No error | 0 | --- |
| Si04 Status | BOOL | R | Si04 Status | Gives the status of safety input terminal 04.<br>0: Error<br>1: No error | 0 | --- |
| Si05 Status | BOOL | R | Si05 Status | Gives the status of safety input terminal 05.<br>0: Error<br>1: No error | 0 | --- |
| Si06 Status | BOOL | R | Si06 Status | Gives the status of safety input terminal 06.<br>0: Error<br>1: No error | 0 | --- |
| Si07 Status | BOOL | R | Si07 Status | Gives the status of safety input terminal 07.<br>0: Error<br>1: No error | 0 | --- |

## A-6-3 NX-SOH200 Safety Output Unit

| Port | Data type | R/W | Name | Description | Default | Corresponding port name |
|---|---|---|---|---|---|---|
| Standard Input 1st Byte | BYTE | R | Standard Input 1st Byte | --- | 00 hex | --- |

| Port | Data type | R/W | Name | Description | Default | Corresponding port name |
|---|---|---|---|---|---|---|
| So00 Monitor Value | BOOL | R | So00 Monitor Value | Gives the status of safety output terminal So00.<br>0: OFF, 1: ON | 0 | So00 Output Value |
| So01 Monitor Value | BOOL | R | So01 Monitor Value | Gives the status of safety output terminal So01.<br>0: OFF, 1: ON | 0 | So01 Output Value |
| Safety Connection Status | BOOL | R | Safety Connection Status | This flag indicates when a safety connection is active. Use it for an input to the Activate terminal on a safety FB or for safety connection/disconnection applications. | 0 | Same |
| Safety Output Terminal Status | BOOL | R | Safety Output Terminal Status | This flag indicates the status of the safety output terminals.<br>0: An error has occurred on one of the safety output terminals.<br>1: All of the safety output terminals are normal (no errors). | 0 | Same |
| Unit Normal Status | BOOL | R | Unit Normal Status | This flag indicates the status of the Unit.<br>0: An error has occurred.<br>1: Normal (no errors) | 0 | --- |
| IO Power Supply Error Flag | BOOL | R | IO Power Supply Error Flag | This flag indicates the status of the I/O power supply voltage.<br>0: The I/O power supply voltage is normal.<br>1: The I/O power supply voltage is incorrect or the I/O power supply is OFF. | 0 | --- |
| Standard Input 2nd Byte | BYTE | R | Standard Input 2nd Byte | --- | 00 hex | --- |
| So00 Status | BOOL | R | So00 Status | Gives the status of safety output terminal 00.<br>0: Error<br>1: No error | 0 | --- |
| So01 Status | BOOL | R | So01 Status | Gives the status of safety output terminal 01.<br>0: Error<br>1: No error | 0 | --- |

## A-6-4 NX-SOD400 Safety Output Unit

| Port | Data type | R/W | Name | Description | Default | Corresponding port name |
|---|---|---|---|---|---|---|
| Standard Input 1st Byte | BYTE | R | Standard Input 1st Byte | --- | 00 hex | --- |

| Port | Data type | R/W | Name | Description | Default | Corresponding port name |
|---|---|---|---|---|---|---|
| So00 Monitor Value | BOOL | R | So00 Monitor Value | Gives the status of safety output terminal So00.<br>0: OFF, 1: ON | 0 | So00 Output Value |
| So01 Monitor Value | BOOL | R | So01 Monitor Value | Gives the status of safety output terminal So01.<br>0: OFF, 1: ON | 0 | So01 Output Value |
| So02 Monitor Value | BOOL | R | So02 Monitor Value | Gives the status of safety output terminal So02.<br>0: OFF, 1: ON | 0 | So02 Output Value |
| So03 Monitor Value | BOOL | R | So03 Monitor Value | Gives the status of safety output terminal So03.<br>0: OFF, 1: ON | 0 | So03 Output Value |
| Safety Connection Status | BOOL | R | Safety Connection Status | This flag indicates when a safety connection is active. Use it for an input to the Activate terminal on a safety FB or for safety connection/disconnection applications. | 0 | Same |
| Safety Output Terminal Status | BOOL | R | Safety Output Terminal Status | This flag indicates the status of the safety output terminals.<br>0: An error has occurred on one of the safety output terminals.<br>1: All of the safety output terminals are normal (no errors). | 0 | Same |
| Unit Normal Status | BOOL | R | Unit Normal Status | This flag indicates the status of the Unit.<br>0: An error has occurred.<br>1: Normal (no errors) | 0 | --- |
| IO Power Supply Error Flag | BOOL | R | IO Power Supply Error Flag | This flag indicates the status of the I/O power supply voltage.<br>0: The I/O power supply voltage is normal.<br>1: The I/O power supply voltage is incorrect or the I/O power supply is OFF. | 0 | --- |
| Standard Input 2nd Byte | BYTE | R | Standard Input 2nd Byte | --- | 00 hex | --- |
| So00 Status | BOOL | R | So00 Status | Gives the status of safety output terminal 00.<br>0: Error<br>1: No error | 0 | --- |
| So01 Status | BOOL | R | So01 Status | Gives the status of safety output terminal 01.<br>0: Error<br>1: No error | 0 | --- |
| So02 Status | BOOL | R | So02 Status | Gives the status of safety output terminal 02.<br>0: Error<br>1: No error | 0 | --- |

| Port | Data type | R/W | Name | Description | Default | Corresponding port name |
|---|---|---|---|---|---|---|
| So03 Status | BOOL | R | So03 Status | Gives the status of safety output terminal 03.<br>0: Error<br>1: No error | 0 | --- |

# A-7 CIP Response Codes

This section provides information on the response codes that are defined in the CIP specifications. They are stored in the received data of CIP messages that are sent to an NX502 CPU Unit, NX102 CPU Unit, NX-series EtherNet/IP Unit, or Safety CPU Unit. In case of the CIP Safety not established error, an extended code supplementing the event information may be registered.

## A-7-1 General Status Codes

| General Status Code (hex) | Status Name | Description of Status |
|---|---|---|
| 00 | Success | Service was successfully performed by the object specified. |
| 01 | Connection failure | A connection related to service failed along the connection path. |
| 02 | Resource unavailable | Resources needed for the object to perform the requested service were unavailable. |
| 03 | Invalid parameter value | See Status Code 20 hex. |
| 04 | Path segment error | The path segment identifier or the segment syntax was not understood by the processing node. Path processing stops when a path segment error occurs. |
| 05 | Path destination unknown | The path is referencing an object class, instance, or structure element that is not known or is not contained in the processing node. Path processing stops when a Path Destination Unknown Error occurs. |
| 06 | Partial transfer | Only part of the expected data was transferred. |
| 07 | Connection lost | The message connection was lost. |
| 08 | Service not supported | The requested service was not supported or was not defined for this object class/instance. |
| 09 | Invalid attribute value | Invalid attribute data was detected. |
| 0 A | Attribute list error | An attribute in the Get_Attribute_List or Set_Attribute_List response has a non-zero status. |
| 0B | Already in requested mode/state | The object is already in the mode/state being requested by the service. |
| 0C | Object state conflict | The object cannot perform the requested service in its current mode/state. |
| 0D | Object already exists | The requested instance of object to be created already exists. |
| 0E | Attribute not settable | A request to modify a non-modifiable attribute was received. |
| 0F | Privilege violation | A permission/privilege check failed. |
| 10 | Device state conflict | The device's current mode/state prohibits the execution of the requested service. |
| 11 | Reply data too large | The data to be transmitted in the response buffer is larger than the allocated response buffer. |
| 12 | Fragmentation of a primitive value | The service specified an operation that is going to fragment a primitive data value, i.e. half a REAL data type. |
| 13 | Not enough data | The requested service did not supply enough data to perform the specified operation. |
| 14 | Attribute not supported | The attribute specified in the request is not supported. |
| 15 | Too much data | The service supplied more data than was expected. |

| General Status Code (hex) | Status Name | Description of Status |
|---|---|---|
| 16 | Object does not exist | An object that does not exist was specified for the requested service. |
| 17 | Service fragmentation sequence not in progress | The fragmentation sequence for this service is not currently active for this data. |
| 18 | No stored attribute data | The attribute data of this object was not saved prior to the requested service. |
| 19 | Store operation failure | The attribute data of this object was not saved due to a failure during the attempt. |
| 1A | Routing failure (request packet too large) | The service request packet was too large for transmission on a network in the path to the destination. The routing device was forced to abort the service. |
| 1B | Routing failure (response packet too large) | The service response packet was too large for transmission on a network in the path from the destination. The routing device was forced to abort the service. |
| 1C | Missing attribute list entry data | The service did not supply an attribute in a list of attributes that was needed by the service to perform the requested behavior. |
| 1D | Invalid attribute value list | The service is returning the list of attributes supplied with status information for those attributes that were invalid. |
| 1E | Embedded service error | An embedded service resulted in an error. |
| 1F | Vendor specific error | A vendor-specific error occurred. The Additional Code Field of the error response defines the error. This is a general error code that is used only for errors that do not correspond to any of the error codes in this table and are not in an object class definition. |
| 20 | Invalid parameter | A parameter for the requested service is invalid. This code is used when a parameter does not meet the requirements of the specification and/or the requirements defined in an application object specification. |
| 21 | Write-once value or medium already written | An attempt was made to write to a write-once medium (e.g. WORM drive or PROM) that was previously written or cannot be changed. |
| 22 | Invalid Reply Received | An invalid reply was received. (For example, the reply service code does not match the request service code, or the reply message is shorter than the minimum expected reply size.) This status code is used for other causes of invalid replies. |
| 23 | Buffer Overflow | The message received is larger than the receiving buffer can handle. The entire message was discarded. |
| 24 | Message Format Error | The format of the received message is not supported by the server. |
| 25 | Key Failure in path | The key segment that was included as the first segment in the path does not match the destination module. The object specific status must indicate which part of the key check failed. |
| 26 | Path Size Invalid | The size of the path that was sent with the service request is either too large or too small for the request to be routed to an object. |
| 27 | Unexpected attribute in list | An attempt was made to set an attribute that is not able to be set at this time. |
| 28 | Invalid Member ID | The member ID specified in the request does not exist in the specified class, instance, and attribute. |
| 29 | Member not settable | A request to modify a non-modifiable member was received. |
| 2 A | Group 2 only server general failure | This error code is reported only by group 2 only servers with 4K or less of code space and only in place of Service not supported, Attribute not supported, or Attribute not settable. |

| General Status Code (hex) | Status Name | Description of Status |
|---|---|---|
| 2B | Unknown Modbus Error | A CIP to Modbus translator received an unknown Modbus Exception Code. |
| 2C | Attribute not gettable | A request to read a non-readable attribute was received. |
| 2D | Instance Not Deletable | The requested object instance cannot be deleted. |
| 2E | Service Not Supported for Specified Path | The object supports the service, but not for the designated application path (e.g. attribute).<br>NOTE: Not to be used for any set service (use General Status Code 0x0E or 0x29 instead) |
| 2F-CF | | Reserved by CIP for future extensions. |
| D0-FF | Reserved for Object Class and service errors | This range of error codes is to be used to indicate object class-specific errors. This code range is used only when none of the error codes in this table accurately reflect the error that occurred. The additional code field is used to describe the general error code in more detail. |

## A-7-2 Extended Status Codes

| General Status (hex) | Additional Status (hex) | Explanation |
|---|---|---|
| 01 | 0100 | Connection in use or duplicate forward open. |
| 01 | 0103 | Transport class and trigger combination not supported. |
| 01 | 0105 | Ownership Conflict or OUNID Mismatch. The configuration is already owned by another originator. |
| 01 | 0106 | Ownership Conflict or OUNID Mismatch. The output connection was already owned by another originator. |
| 01 | 0107 | Connection not found at target application. |
| 01 | 0108 | Invalid connection type. There is a problem with either the connection type or priority of the connection. |
| 01 | 0109 | Invalid connection size. |
| 01 | 0110 | Device not configured. |
| 01 | 0111 | RPI not supported. May also indicate problem with connection time-out multiplier, or production inhibit time. |
| 01 | 0112 | RPI VALUE(S) not acceptable |
| 01 | 0113 | Connection Manager cannot support any more connections. |
| 01 | 0114 | Either the vendor ID or the product code in the key segment does not match the device. |
| 01 | 0115 | Device type in the key segment does not match the device. |
| 01 | 0116 | Major or minor revision information in the key segment does not match the device. |
| 01 | 0117 | Invalid connection point. |
| 01 | 0118 | Invalid configuration format. |
| 01 | 0119 | Connection request failed because there is no controlling connection currently open. |
| 01 | 011A | Target application cannot support any more connections. |
| 01 | 011B | RPI is smaller than the production inhibit time. |
| 01 | 011C | Transport Class not supported |

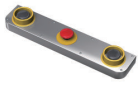
| General Status (hex) | Additional Status (hex) | Explanation |
|---|---|---|
| 01 | 011D | T->O Production Trigger not supported |
| 01 | 011E | Direction not supported |
| 01 | 011F | Invalid originator to target Network Connection Fixvar |
| 01 | 0120 | Invalid target to originator Network Connection Fixvar |
| 01 | 0121 | Invalid originator to target Network Connection Priority |
| 01 | 0122 | Invalid target to originator Network Connection Priority |
| 01 | 0123 | Invalid originator to target Network Connection Type |
| 01 | 0124 | Invalid target to originator Network Connection Type |
| 01 | 0125 | Invalid originator to target Network Connection Redundant_Owner |
| 01 | 0126 | Invalid configuration size |
| 01 | 0127 | Invalid originator to target Network Connection Size |
| 01 | 0128 | Invalid target to originator Network Connection Size |
| 01 | 0129 | Invalid Configuration Application Path |
| 01 | 012A | Invalid Consuming Application Path |
| 01 | 012B | Invalid Producing Application Path |
| 01 | 012C | Configuration Symbol does not exist |
| 01 | 012D | Consuming Symbol does not exist |
| 01 | 012E | Producing Symbol does not exist |
| 01 | 012F | Inconsistent Application Path combination |
| 01 | 0130 | Inconsistent Consume Data Format |
| 01 | 0131 | Inconsistent Produce Data Format |
| 01 | 0132 | Null Forward Open function not supported |
| 01 | 0133 | Connection Timeout Multiplier not acceptable |
| 01 | 0134 | Mismatched T->O Network Connection Size |
| 01 | 0135 | Mismatched T->O Network Connection Fixvar |
| 01 | 0136 | Mismatched T->O Network Connection Priority |
| 01 | 0137 | Mismatched Transport Class |
| 01 | 0138 | Mismatched T->O Production Trigger |
| 01 | 0139 | Mismatched T->O Production Inhibit Time Segment |
| 01 | 0203 | Connection cannot be closed because the connection has timed out. |
| 01 | 0204 | Unconnected_Send service timed out while waiting for a response. |
| 01 | 0205 | Parameter Error in Unconnected Send Service or Parameter Error in SafetyOpen or SafetyClose |
| 01 | 0206 | Message too large for unconnected message service. |
| 01 | 0207 | Unconnected acknowledgement without reply. |
| 01 | 0301 | No buffer memory available. |
| 01 | 0302 | Network bandwidth not available for data. |
| 01 | 0303 | No tag filters available. |
| 01 | 0304 | Not configured to send real-time data. |
| 01 | 0305 | Schedule Signature mismatch |
| 01 | 0306 | Schedule Signature validation not possible |
| 01 | 0311 | Port that was specified in port segment is not available. |
| 01 | 0312 | Link address that was specified in port segment is not available. |
| 01 | 0315 | Invalid segment type or segment value in path. |
| 01 | 0316 | Path and connection were not equal when closing the connection. |

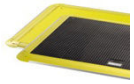
| General Status (hex) | Additional Status (hex) | Explanation |
|---|---|---|
| 01 | 0317 | Either the segment is not present or the encoded value in the network segment is invalid. |
| 01 | 0318 | Link address to self is invalid. |
| 01 | 0319 | Resources on secondary are unavailable. |
| 01 | 031A | Connection is already established. |
| 01 | 031B | Direct connection is already established. |
| 01 | 031C | Others |
| 01 | 031D | Redundant connection mismatch. |
| 01 | 031E | There are no more reception resources available on the sending module. |
| 01 | 031F | No connection resources exist for the target path. |
| 01 | 0320 - 07FF | Vendor specific. |
| 01 | 0801 | Incompatible Multicast RPI. An existing connection has been established at a different RPI. |
| 01 | 0802 | Invalid Safety Connection Size |
| 01 | 0803 | Invalid Safety Connection Format |
| 01 | 0804 | Invalid Time Correction Connection Parameters |
| 01 | 0805 | Invalid Ping Interval EPI Multiplier |
| 01 | 0806 | Time Coordination Msg Min Multiplier |
| 01 | 0807 | Network Time Expectation Multiplier |
| 01 | 0808 | Timeout Multiplier |
| 01 | 0809 | Invalid Max Consumer Number |
| 01 | 080A | Invalid CPCRC |
| 01 | 080B | Time Correction Connection Id Invalid |
| 01 | 080C | SCID Mismatch. The SCID was non-zero and did not match the value in the target |
| 01 | 080D | TUNID not set. Device is out-of-box and TUNID has not been set, so connections are not allowed. |
| 01 | 080E | TUNID Mismatch. The TUNID provided does not match. The message was likely routed to this node in error |
| 01 | 080F | Configuration operation not allowed |
| 01 | 0810 | No target application data available |
| 01 | 0811 | No originator application data available |
| 01 | 0812 | Node address has changed since the network was scheduled |
| 01 | 0813 | Not configured for off-subnet multicast |
| 01 | 0814 | Invalid Produce/Consume Data Format |

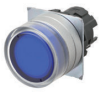
# A-8 Icon list for Safety Slave Unit Parameters

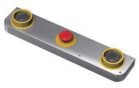
The icons that you can set or change for Safety Slave Unit parameters are listed in the following table.

## A-8-1 External Device Icons for Input Devices

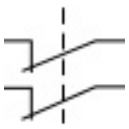
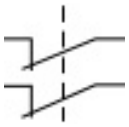
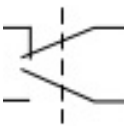
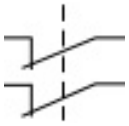
| Category | Device | Settable external device icons | | | |
|---|---|---|---|---|---|
| Safety Switch | Emergency stop switch<br>for dual channel equivalent input |  |  | | |
| | Emergency stop switch<br>for single channel |  |  | | |
| | Safety door switch<br>for dual-channel equivalent input |  |  | | |
| | Safety door switch<br>for dual-channel complementary input |  |  | | |
| | Safety door switch<br>for single channel |  |  | | |
| | Safety limit switch<br>for dual-channel equivalent input |  |  | | |
| | Safety limit switch<br>for dual-channel complementary input |  |  | | |
| | Safety limit switch<br>for single channel |  |  | | |
| | Two-hand switches |  | | | |

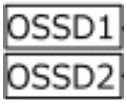
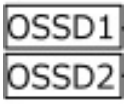
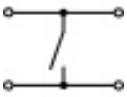
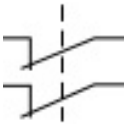
| Category | Device | Settable external device icons | | | |
|---|---|---|---|---|---|
| Safety Switch | Safety key selector switch<br>for dual-channel equivalent input |  | | | |
| | Safety key selector switch<br>for dual-channel complementary input |  | | | |
| | Safety key selector switch<br>for single channel |  | | | |
| | Enabling switch |  | | | |
| Safety Sensor | Safety light curtain |  |  | | |
| | Safety laser scanner |  |  | | |
| | High-Coded Door Switch |  | | | |
| | High-Coded Door Switch (Guard Lock Model) |  | | | |
| | High-Coded Door Switch (For Gate) |  | | | |
| Specialty devices | Non-contact Door Switch |  |  |  | |
| | Single Beam Safety Sensor |  |  | | |
| | Safety Mat/Safety Edge |  |  |  | |
| EDM Feedback | EDM Feedback | Feedback | | | |

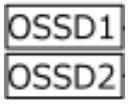
| Category | Device | Settable external device icons | | | |
|---|---|---|---|---|---|
| Standard Input Device | Reset switch with test pulse |  | | | |
| | Reset switch without test pulse |  | | | |

| Category | Device | Settable external device icons | | | |
|---|---|---|---|---|---|
| Generic Device | Mechanical Contact for Single Channel |  |  |  |  |
| | |  |  |  |  |
| | | Feedback | | | |
| | Mechanical Contact for Dual Channel Equivalent |  |  |  |  |
| | |  |  |  |  |
| | |  | | | |
| | Mechanical Contact for Dual Channel Complementary |  |  |  |  |
| | |  |  |  | |
| | Semiconductor Output for Single Channel |  | | | |
| | Semiconductor Output for Dual Channel Equivalent |  |  |  |  |
| | |  |  |  |  |
| | | | | | |
| | Semiconductor Output for Dual Channel Complementary |  | | | |

## A-8-2 Contact Icons for Input Devices

| Category | Device | Settable contact icons | | |
|---|---|---|---|---|
| Safety Switch | Emergency stop switch<br>for dual-channel equivalent input |  <br>2NC | | |
| | Emergency stop switch<br>for single channel |  <br>1NC | | |
| | Safety door switch<br>for dual-channel equivalent input |  <br>2NC |  <br>2NO | |
| | Safety door switch<br>for dual-channel complementary input |  <br>1NC/1NO |  <br>1NO/1NC | |
| | Safety door switch<br>for single channel |  <br>1NC |  <br>1NO | |
| | Safety limit switch<br>for dual-channel equivalent input |  <br>2NC |  <br>2NO | |
| | Safety limit switch<br>for dual-channel complementary input |  <br>1NC/1NO |  <br>1NO/1NC | |
| | Safety limit switch<br>for single channel |  <br>1NC |  <br>1NO | |
| | Two-hand switches |  <br>1NO/1NC |  <br>1NC/1NO | |
| | Safety key selector switch<br>for dual-channel equivalent input |  <br>2NC |  <br>2NO | |
| | Safety key selector switch<br>for dual-channel complementary input |  <br>1NC/1NO |  <br>1NO/1NC | |

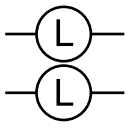
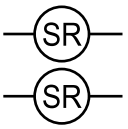
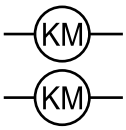
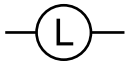
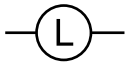
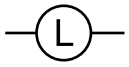
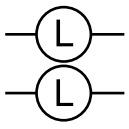
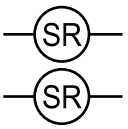
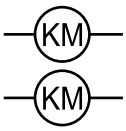
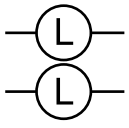
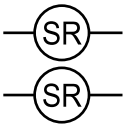
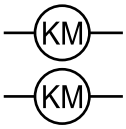
| Category | Device | Settable contact icons | | |
|---|---|---|---|---|
| Safety Switch | Safety key selector switch for single channel |  1NC |  1NO | |
| | Enabling switch |  2NO | | |
| Safety Sensor | Safety light curtain |  OSSD1/OSSD2 | | |
| | Safety laser scanner |  OSSD1/OSSD2 | | |
| Specialty devices | Non-contact Door Switch | | | |
| | Single Beam Safety Sensor |  OSSD1 | | |
| | Safety Mat/Safety Edge |  SafetyMat/ SafetyEdge | | |
| EDM Feedback | EDM Feedback |  1NC | | |
| Standard Input Device | Reset switch with test pulse |  1NO |  1NC | |
| | Reset switch without test pulse |  1NO |  1NC | |
| Generic Device | Mechanical Contact for Single Channel |  1NC |  1NO | |
| | Mechanical Contact for Dual Channel Equivalent |  2NC |  2NO | |

| Category | Device | Settable contact icons | | |
|---|---|---|---|---|
| Generic Device | Mechanical Contact for Dual Channel Complementary |  1NC/1NO |  1NO/1NC | |
| | Semiconductor Output for Single Channel |  OSSD1 | | |
| | Semiconductor Output for Dual Channel Equivalent |  OSSD1/OSSD2 | | |
| | Semiconductor Output for Dual Channel Complementary | | | |

## A-8-3 External Device Icons for Output Devices

| Category | Device | Settable external device icons | | |
|---|---|---|---|---|
| Relays with Forcibly Guided Contacts | Relays with Forcibly Guided Contacts for Dual Channel |  | | |
| | Relays with Forcibly Guided Contacts for Single Channel |  | | |
| Generic Device | Single Channel with Test Pulse |  |  | |
| | Single Channel without Test Pulse |  | | |
| | Dual Output with Test Pulse |  |  | |
| | Dual Output without Test Pulse |  | | |

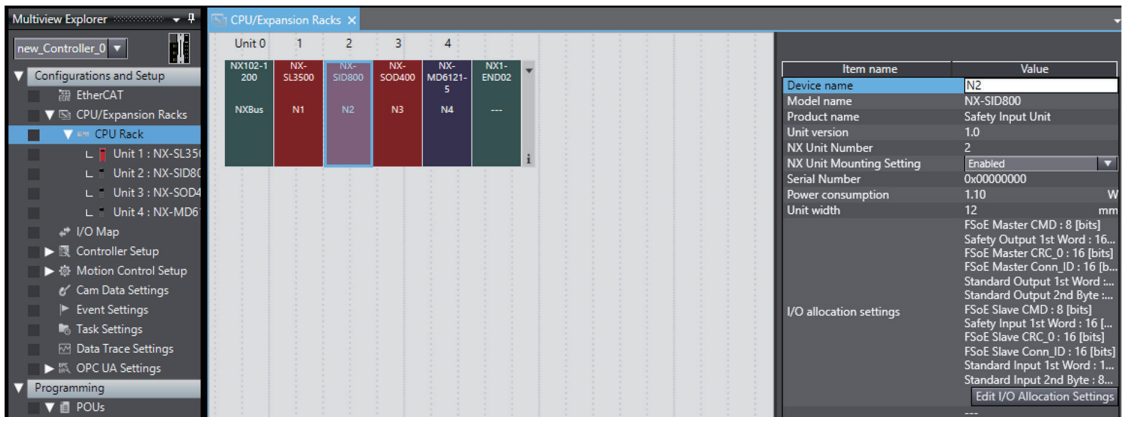
## A-8-4 Contact Icons for Output Devices

| Category | Device | Settable contact icons | | |
|---|---|---|---|---|
| Relays with Forcibly Guided Contacts | Relays with Forcibly Guided Contacts for Dual Channel |  Load |  Relays with Forcibly Guided Contacts |  Contactor |
| | | | | |
| | Relays with Forcibly Guided Contacts for Single Channel |  Load |  Relays with Forcibly Guided Contacts |  Contactor |
| | | | | |
| Generic Device | Single Channel with Test Pulse |  Load |  Relays with Forcibly Guided Contacts |  Contactor |
| | | | | |
| | Single Channel without Test Pulse |  Load |  Relays with Forcibly Guided Contacts |  Contactor |
| | | | | |
| | Dual Output with Test Pulse |  Load |  Relays with Forcibly Guided Contacts |  Contactor |
| | | | | |
| | Dual Output without Test Pulse |  Load |  Relays with Forcibly Guided Contacts |  Contactor |
| | | | | |

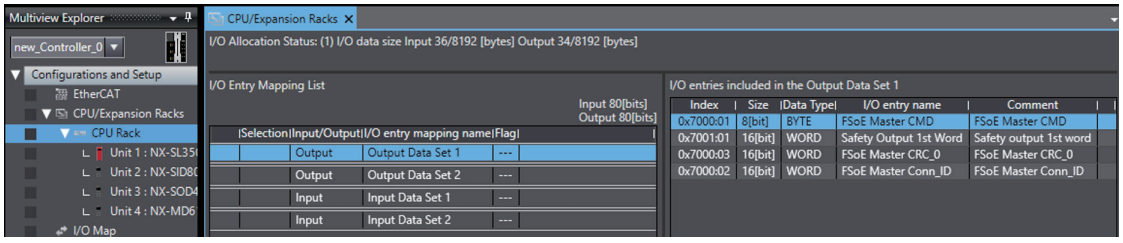
# A-9 Calculating I/O Sizes for CPU Rack or Slave Terminals

To confirm the I/O size of the CPU rack and Slave Terminal, including the Safety Control Unit, use the Edit NX Bus Master Configuration Tab Page for each NX bus master. Refer to the manual of each NX bus master for details on how to display the Edit NX Bus Master Configuration Tab Page.

***1*** On the Edit NX Bus Master Configuration Tab Page, select the NX Unit for which the connected I/O allocation can be changed.
This enables the **Edit I/O Allocation Settings** Button in the Parameter Settings Area on the right side of the tab page.




***2*** Click the **Edit I/O Allocation Settings** Button.
The **Edit I/O Allocation Setting** Tab Page is displayed.




The input and output data sizes of the CPU rack and the entire Slave Terminal are displayed in the I/O Allocation Status Column.

# A-10 Printing

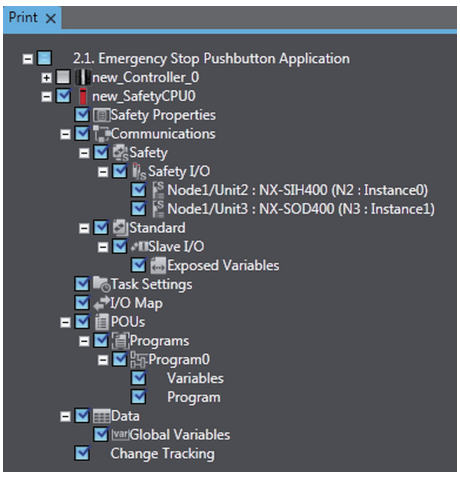
This section describes how to print information related to Safety Control Unit from the Sysmac Studio. Refer to the *Sysmac Studio Version 1 Operation Manual (Cat. No. W504)* for basic printing procedures for the Sysmac Studio.

## A-10-1 Selecting the Items to Print

Use the following procedure to specify Safety Control Unit as the print target.

***1*** Select **File** - **Print**.
The items that you can print will be displayed.

***2*** Select the items for the Safety Control Unit.
The Safety Control Unit will be set up for printing.

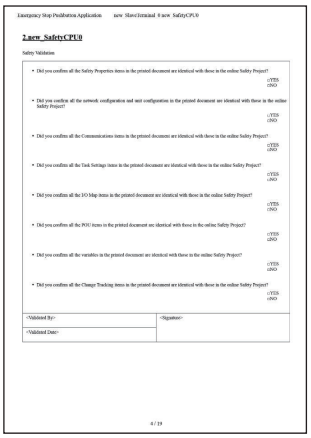



## A-10-2 Items that are Printed

The items that you selected for printing are printed in the order in which the printing items were displayed.
Samples of the information that is printed for a Safety Control Unit are provided below.

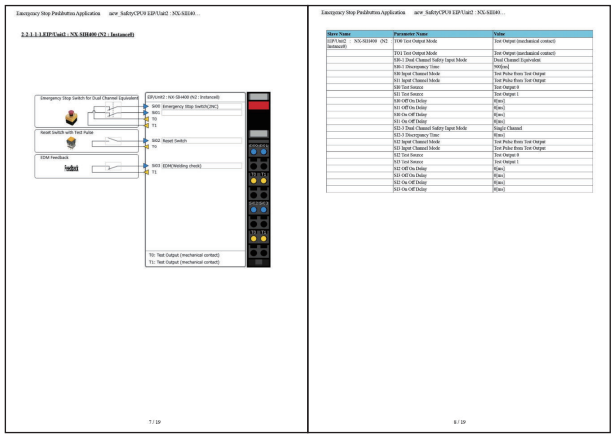
### ● Signature and Validation Boxes

This information is printed on the first page of the Safety Control Unit.
Validation and signature boxes for safety validation are printed.



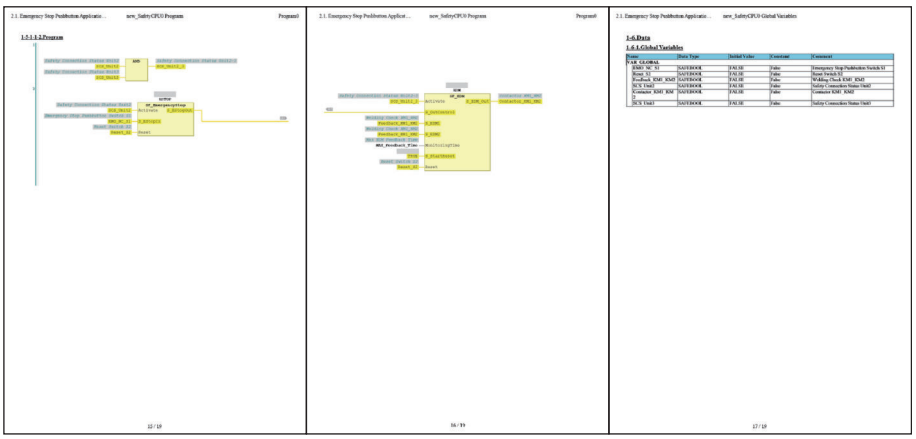
## Wiring Diagrams

The wiring diagrams that are set on the Parameters Tab Page are printed.



## Program

The programs are printed.

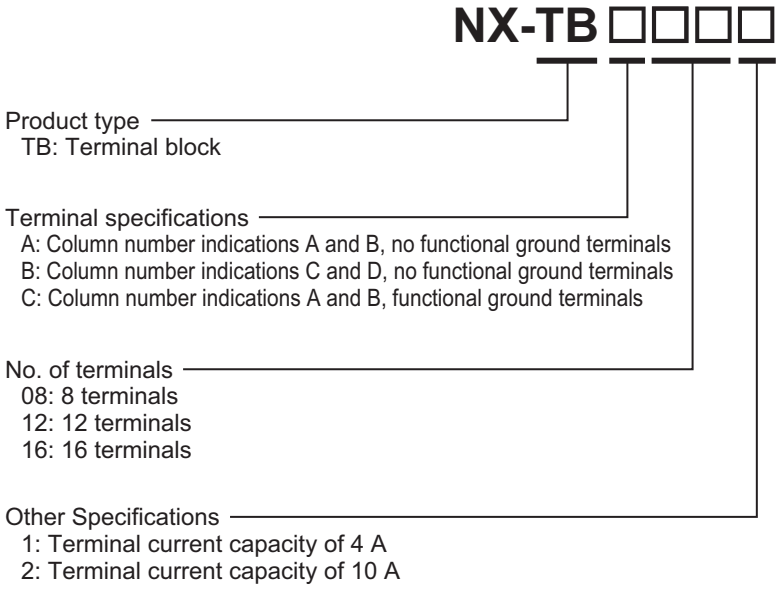


# A-11 List of Screwless Clamping Terminal Block Models

This section explains how to read the screwless clamping terminal block model numbers and shows the model number table.

## A-11-1 Model Notation

The screwless clamping terminal block models are assigned based on the following rules.




## A-11-2 List of Terminal Block Models

The following table shows a list of screwless clamping terminal blocks.

| Terminal block model | Number of terminals | Ground terminal mark | Terminal current capacity |
|---|---|---|---|
| NX-TBA081 | 8 | Not provided | 4 A |
| NX-TBA121 | 12 | | |
| NX-TBA161 | 16 | | |
| NX-TBB121 | 12 | | |
| NX-TBB161 | 16 | | |
| NX-TBA082 | 8 | | 10 A |
| NX-TBA122 | 12 | | |
| NX-TBA162 | 16 | | |
| NX-TBB082 | 8 | | |
| NX-TBB122 | 12 | | |
| NX-TBB162 | 16 | | |
| NX-TBC082 | 8 | Provided | |
| NX-TBC162 | 16 | | |

**Note** When you purchase a terminal block, purchase an NX-TB□□□2.

# A-12 I/O Response Times for Communications between NX Units on EtherNet/IP Slave Terminals

This section describes the input response times and output response times for communications between NX Units on EtherNet/IP Slave Terminals.

## Input Response Times

The input response time gives the time required from when an external signal is input to the NX Unit until it is processed by the Safety CPU Unit.
The maximum input response time is calculated as follows:

Formula: Maximum input response time = Safety task period + Tnx-Indelay + 4.5 ms

The elements in the formulas are as follows:

- Safety task period: Setting of the safety task period in the Safety CPU Unit
- Tnx-Indelay: Input delay time of the NX Unit

## Output Response Times

The output response time is the time from data processing on the Safety CPU Unit until the results are externally output from an NX Unit.
The maximum output response time is calculated as follows:

Formula: Maximum output response time = Safety task period + Tnx-OutProc + Tnx-Outdelay + 4.5 ms

The elements in the formulas are as follows:

- Safety task period: Setting of the safety task period in the Safety CPU Unit
- Tnx-OutProc: Output data processing time of the NX Unit
- Tnx-Outdelay: Output delay time of the NX Unit

## Definition of Formula Elements and Calculation Methods

This section defines the elements and describes the calculation methods.

### ● Tnx-Indelay

This is the input delay time of the NX Unit. It is the time required to read the status of the input terminals into NX Unit memory. There is a unique value for each type of NX Unit.
The NX Unit input delay time is different for each type of NX Unit. The following table gives the definition of the input delay time for each type of NX Unit.

| NXType of NX Unit | Tnx-Indelay | Remarks |
|---|---|---|
| Digital Input Units | ON/OFF response time + Input filter time | The ON/OFF response time depends on the model of the Unit.<br>You can set the input filter time for each Unit. |
| Analog Input Units | Conversion time × Number of points | The conversion time and number of points depend on the model of the Unit. |

Refer to the user's manuals for individual NX Units or to the *NX-series Data Reference Manual (Cat.No. W525)* for the definition of Tnx-Indelay in the above table and use it in the calculations.

## ● Tnx-OutProc

This is the output data processing time of the NX Unit. It is the time from when the NX Unit reads the output data that was sent by the EtherNet/IP Coupler Unit until preparations to update the output data are completed. There is a unique value for each type of NX Unit.
The NX Unit output data processing time has a different definition or value for each type of NX Unit. The following table gives the value or definition of Tnx-OutProc for each type of NX Unit.

| Type of NX Unit | Tnx-OutProc | Remarks |
|---|---|---|
| Digital Output Units | 0 µs | --- |
| Analog Output Units | Conversion time × Number of points | The conversion time and number of points depend on the model of the Unit. |

If a formula rather than a value is given for Tnx-OutProc in the above table, refer to the user's manuals for individual NX Units or to the *NX-series Data Reference Manual (Cat. No. W525)* for the values of the items to make the calculation.

## ● Tnx-Outdelay

This is the output delay time of the NX Unit. This is the time required to change the output terminals according to the updated output data in the NX Unit. There is a unique value for each type of NX Unit.
The NX Unit output delay time has a different definition or value for each type of NX Unit. The following table gives the value or definition of the output delay time for each type of NX Unit.

| Type of NX Unit | Tnx-Outdelay | Remarks |
|---|---|---|
| Digital Output Units | ON/OFF response time | The ON/OFF response time depends on the model of the Unit. |
| Analog Output Units | 0 µs | --- |

If a formula rather than a value is given for Tnx-Outdelay in the above table, refer to the user's manuals for individual NX Units or to the *NX-series Data Reference Manual (Cat. No. W525)* for the values of the items to make the calculation.



**Additional Information**

To calculate the I/O response times between a Safety I/O Unit and a standard I/O Unit, add the input response time or output response time to the FSoE watchdog timer value of the Safety I/O Unit.

# A-13 Units That Support Communications between NX Units

Refer to *A-20 Version Information* on page A-113.

# A-14 Checking the Signature Code on the Seven-segment Indicator

Use the following procedure to display a signature code on the seven-segment indicator on the front panel of the Safety CPU Unit. This function is available for NX-SL5□□□ Units.

***1*** Press the service switch on the front panel of the Safety CPU Unit.
If you hold down the service switch, the display of the seven-segment indicator will change as shown below.

After 1 second "Si." → After 3 seconds "Lo." → After 5 seconds "--"

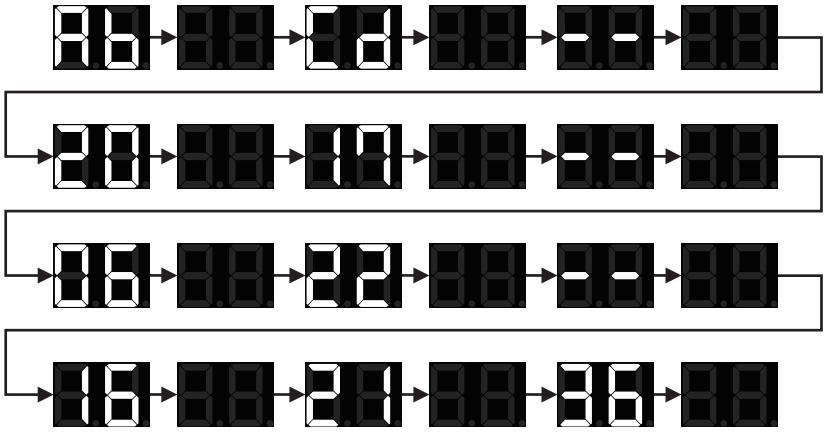


Release the service switch while the indicator shows "Si".
If you release the service switch while "--" is displayed, the function does not execute and the original state is restored.

***2*** The signature code appears.
The indicator displays all nine bytes starting from the creation date (UTC) and then the signature code one byte at a time.

(Signature display start symbol --> Signature code (High) --> Signature code (Low) --> Separator (--) --> Year (High) --> Year (Low) --> Separator (--) --> Month --> Day --> Separator (--) --> Hour --> Minute --> Second)

Example: The signature code is ABCD, and the date is 16:21:36 of June 22, 2017 (UTC)



When the signature code is undefined, "--" appears.

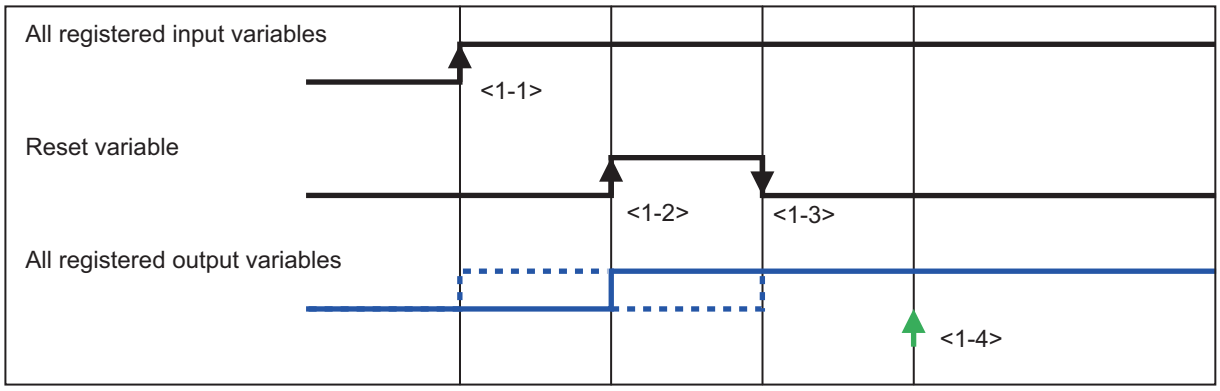
# A-15 Execution Scenarios for the Simple Automatic Test



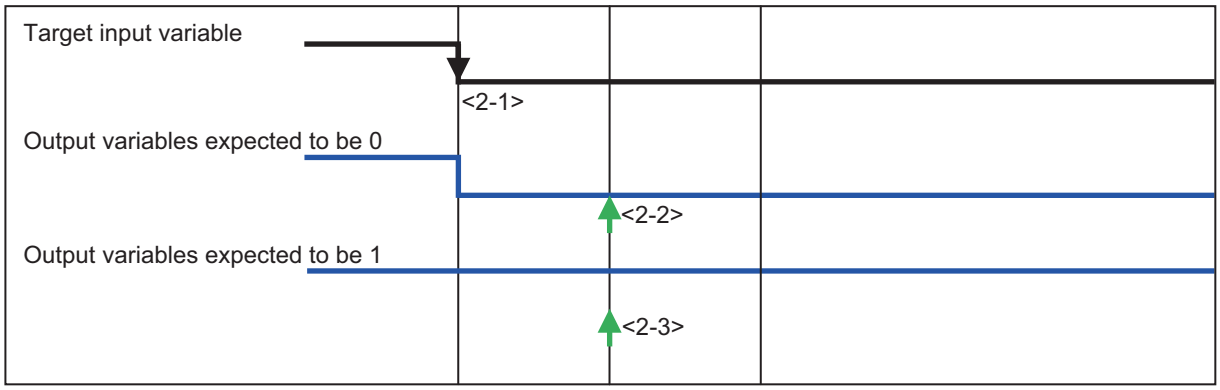
The operations given in the following table are executed in order for the Simple Automatic Test.

| No. | Test phase | Executed processes | Confirmations | Displayed errors |
|---|---|---|---|---|
| 1 | Initialize | Set all the registered input variables to TRUE. <1-1><br>Set the reset variable to TRUE. <1-2><br>Set the reset variable to FALSE. <1-3> | Examine all the registered output variables are TRUE. <1-4> | <Error><br>"Cannot Execute"<br><Remarks><br>"The following output variables are FALSE after the reset." |
| 2 | Shutoff Test | The input variables are set to FALSE. <2-1> | Examine the output variables expected to be 0 in the expected value setting table are FALSE. <2-2> | <Error><br>"Failed"<br><Remarks><br>"The following output variables do not match with the expected values." |
| | | | Examine the output variables expected to be 1 in the expected value setting table are TRUE. <2-3> | <Error><br>"Failed"<br><Remarks><br>"The following output variables do not match with the expected values." |
| 3 | Restore Test <Reset Type: Auto> | Set the input variable to TRUE. <3-1> | Examine all the registered output variables are TRUE. <3-2> | <Error><br>"Failed"<br><Remarks><br>"The following output variables do not return." |
| 4 | Restore Test <Reset Type: Manual> | Set the input variable to TRUE. <4-1><br>Set the reset variable to TRUE. <4-2><br>Set the reset variable to FALSE. <4-3> | Examine the output variables expected to be 0 in the expected value setting table are FALSE. <4-4> | <Error><br>"Failed"<br><Remarks><br>"The following output variables do return." |
| | | | Examine all the registered output variables are TRUE. <4-5> | <Error><br>"Failed"<br><Remarks><br>"The following output variables are FALSE after the reset." |

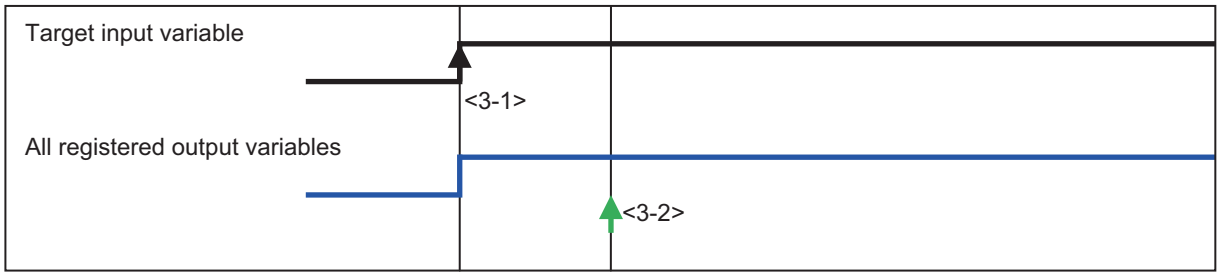
1. Initialization



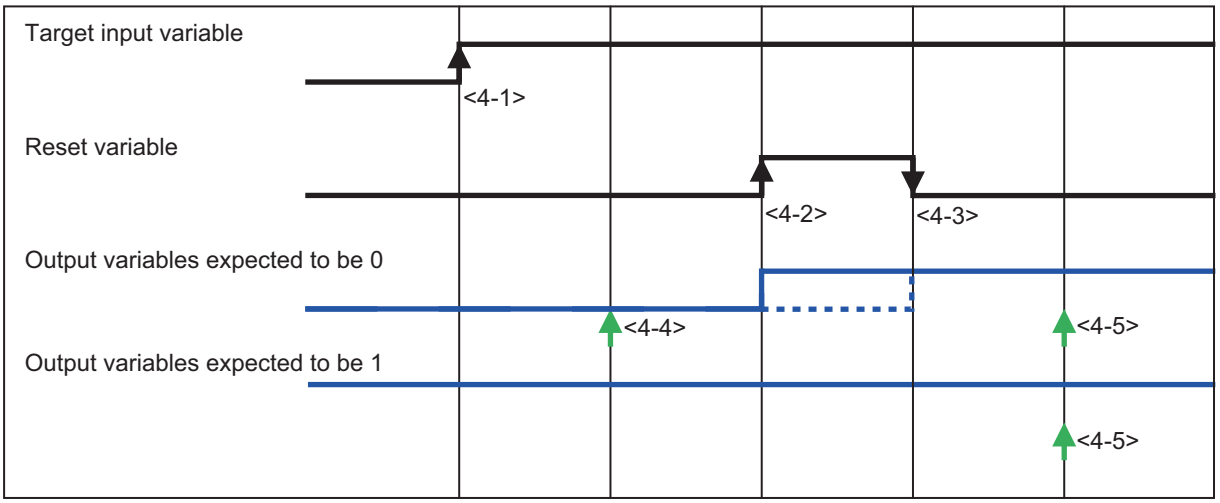

2. Shutoff Test




3. Restore Test (Reset Type: Auto)




4. Restore Test (Reset Type: Manual)


Target input variable
<4-1>
Reset variable
<4-2>
<4-3>
Output variables expected to be 0
<4-4>
<4-5>
Output variables expected to be 1
<4-5>

# A-16 Differences in Checking Operation between the Simulator and Safety CPU Unit

This section describes the differences in the program debugging functions that you can perform on the Sysmac Studio between online debugging on a Safety CPU Unit and offline debugging with the Simulator.

| Debugging function | Safety CPU Unit | Simulator | Reference |
|---|---|---|---|
| Monitoring | Supported. | Supported. | *7-6-3 Monitoring Variables in the FBD Editor* on page 7-29<br>*7-6-4 Monitoring Variables in a Watch Tab Page* on page 7-30 |
| Monitoring in a Watch Tab Page | Supported. | Supported. | *7-6-5 Controlling BOOL Variables, Changing Present Values, and Using Forced Refreshing* on page 7-32 |
| Controlling BOOL variables | Supported. | Supported. | |
| Forced refreshing inputs | Supported. | Supported. | |
| Changing present values of data | Supported. | Supported. | |
| Monitoring Controller status | Supported. | Supported. | *7-15 Monitoring Controller Status* on page 7-76 |
| Clear All Memory | Supported. | Not supported. | *7-16-2 Clear All Memory Operation* on page 7-79 |
| Changing the operating mode | Supported. | Not supported. | *7-3 Operating Modes of the Safety CPU Unit* on page 7-11 |
| Troubleshooting | Supported. | Not supported. | *Section 12 Troubleshooting* on page 12-1 |
| Monitoring error information | Supported. | Not supported. | |
| Displaying error logs | Supported. | Not supported. | |

# A-17 I/O Data Enable Flag for CIP Safety Connections



You can use the I/O data enable flag for CIP Safety connections in the safety program.

## ON and OFF Conditions of the I/O Data Enable Flag for CIP Safety Connections

You can use the I/O data enable flag for CIP Safety connections in the safety program.

| Connection type | Data Direction | ON (data enabling) conditions | OFF (data disabling) conditions |
|---|---|---|---|
| Originator Connection | IN (Target Device →Safety CPU Unit) | Safety connection open and valid data*1 received | Disconnection or invalid data*1 received |
| | OUT (Safety CPU Unit→ Target Device) | Safety connection open and data being transmitted | Disconnection |
| Target Connection (Single-cast) | OUT (Originator Device → Safety CPU Unit) | Safety connection open and valid data*2 received | Disconnection or invalid data*2 received |
| | IN (Safety CPU Unit → Originator Device) | Safety connection open and data being transmitted | Disconnection |
| Target Connection (Multi-cast) | IN (Safety CPU Unit → Originator Device) | At least one connection open and data being transmitted | All connections disconnected |

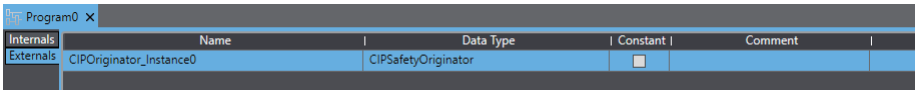
*1. Whether the data is enabled or disabled depends on the target device. When the target device is NX-SL5500 or NX-SL5700, the data is always enabled.

*2. Whether the data is enabled or disabled depends on the originator device. When the originator device is NX-SL5500 or NX-SL5700, the data is always enabled.

## Using the I/O Data Enable Flag for CIP Safety Connections

The following describes the procedure to use the I/O data enable flag for CIP Safety connections.

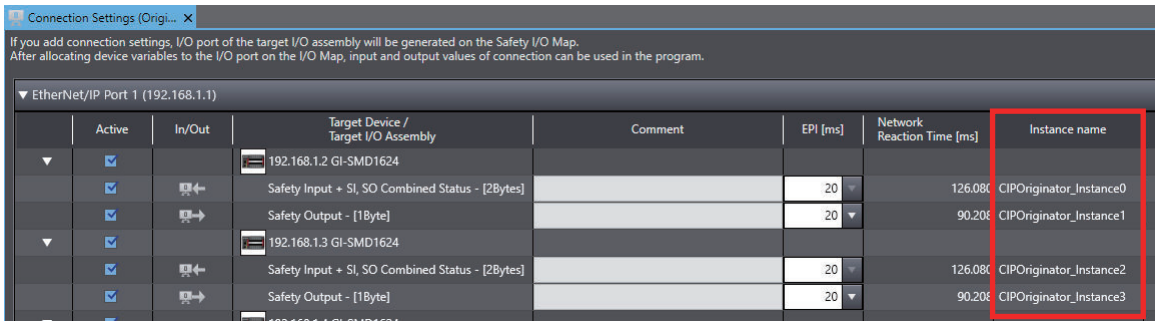
***1*** Register CIP Safety connection instance variables as the external variables of the program POU.



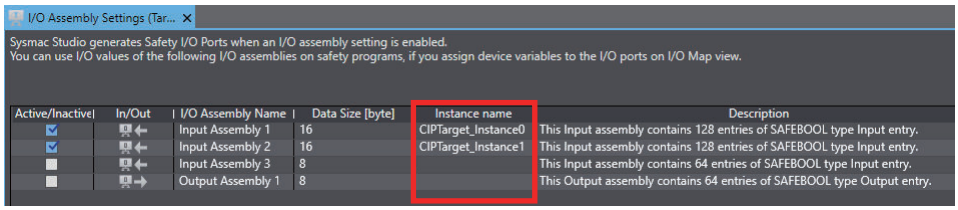

The following list shows a combination of the instance variable name and data type.

| Instance variable name | Data type |
|---|---|
| "CIPOriginator_Instance" + number | CIPSafetyOriginator |
| "CIPTarget_Instance" + number | CIPSafetyTarget |

The instance variable name of the originator can be verified on the **Connection Settings (Originator)** screen.




The target instance variable name can be verified on the **I/O Assembly Settings (Target)** screen.




***2*** Use "InstanceVariableName.S_RunIdleFlag" in the program POU.

### Precautions for Correct Use

The I/O data enable flag for CIP Safety connections (S_RunIdleFlag) is supported by the NX-SL5500/NX-SL5700 unit version 1.4 or higher. Do not use this feature with the unit version 1.3 or lower.

# A-18 Safety: Update Configurations and Setup Transfer Data

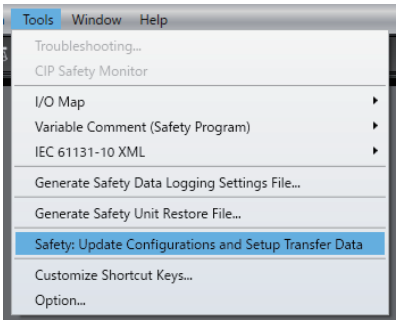


If a standard controller and a Safety CPU Unit are transferred by using different projects, the following error may occur.

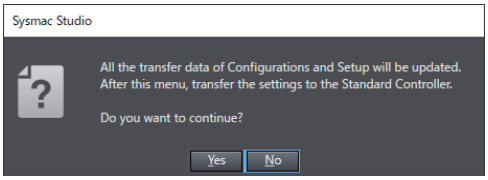
| Event name | Event Code |
|---|---|
| NX Bus Communications Settings and Safety Application Data Mismatch | 10520000 hex |

If you cannot resolve the above error after you implement the normal correction, use the following procedure.

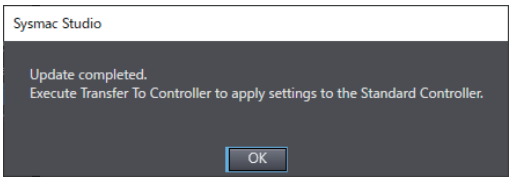
***1*** From the main menu, select **Tools** – **Safety: Update Configurations and Setup Transfer Data**.




The following confirmation dialog is displayed.




***2*** Click the **Yes** button.
When you complete Safety: Update Configurations and Setup Transfer Data, the following message appears.

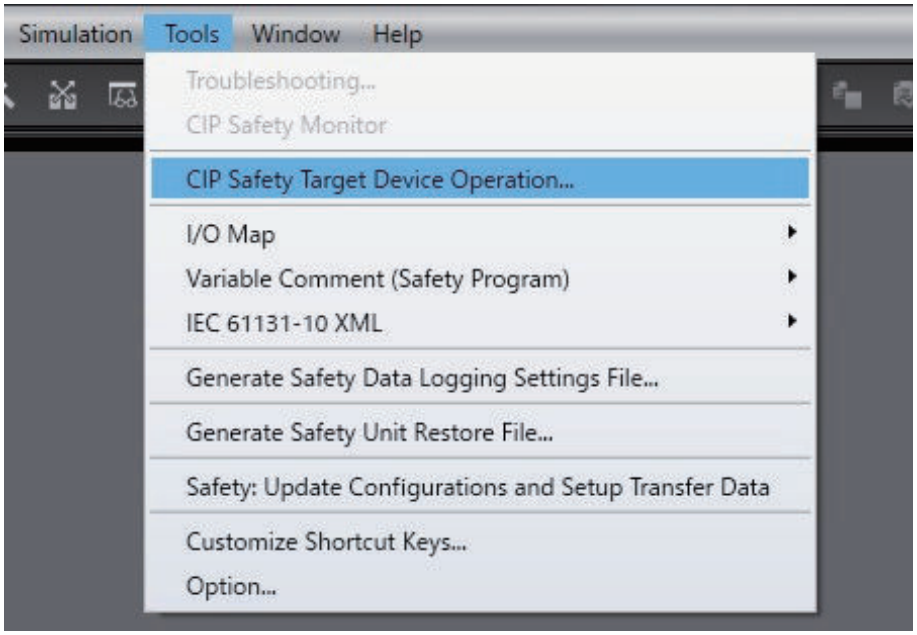



***3*** Click the **OK** button.
Execute **Transfer to Controller** of the standard controller after the update is completed.

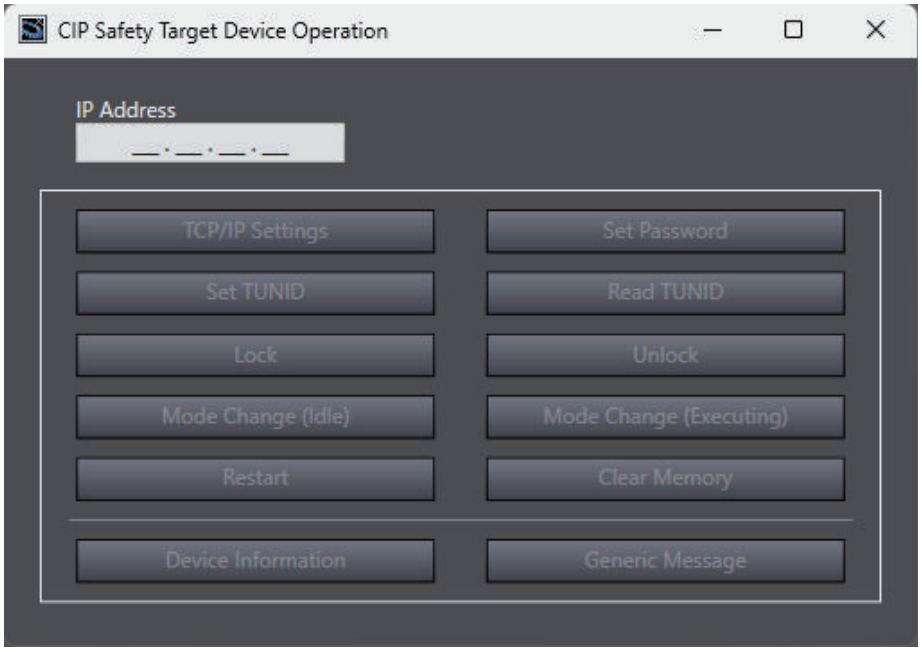
# A-19 CIP Safety Target Device Operation

You can send CIP messages directly to a CIP Safety target device from the computer to perform the memory clear, TUNID setting, and other operations.
To directly operate the target device without using the Safety CPU Unit, perform the following procedure.

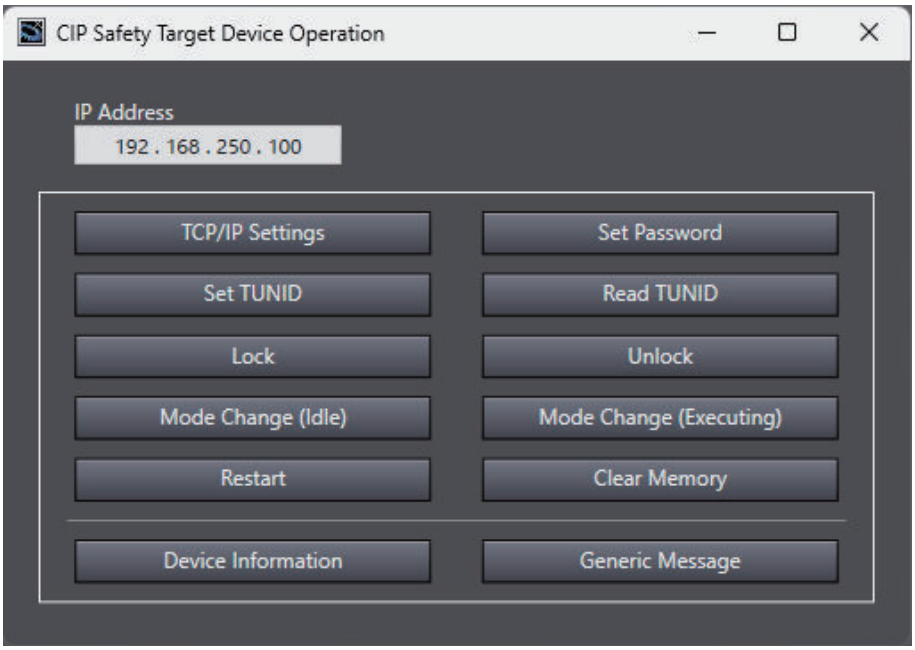
**1** From the Controller Menu, select**Tools**–**CIP Safety Target Device Operation**.



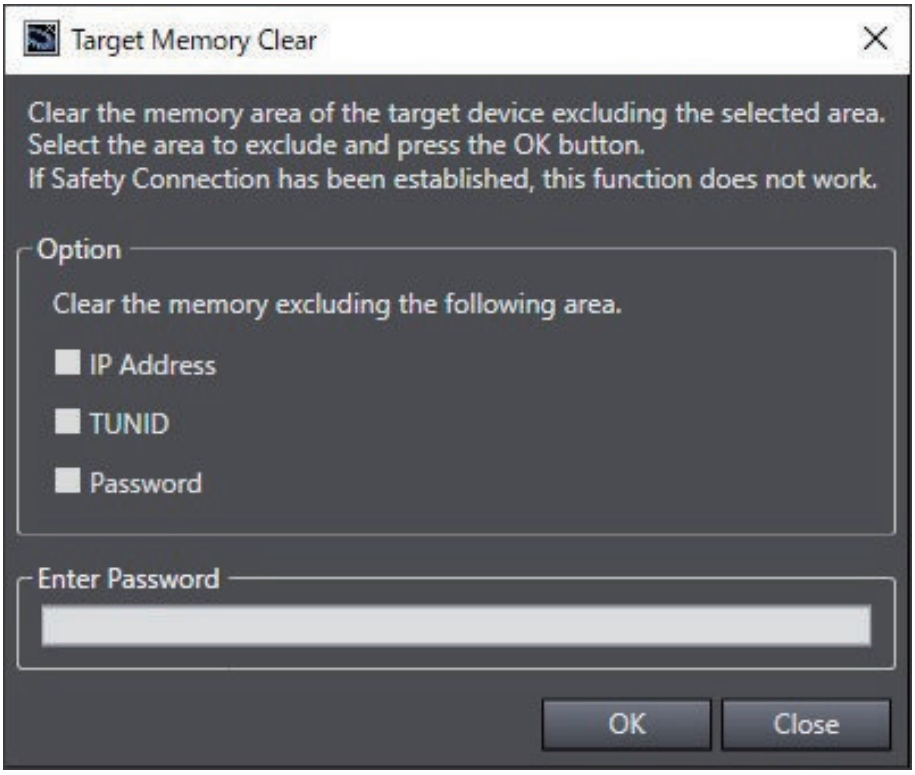

The CIP Safety Target Device Operation window is displayed.




**2** Enter the IP address of the target device, and click a desired operation button.




Example: When the**Clear Memory**button is pressed
The following Target Memory Clear window is displayed.




***3*** Operate the device as instructed on the displayed window.
The operation buttons and the contents to be executed are as follows.

| Button name | Description |
|---|---|
| TCP/IP Settings | Sets such as the IP address of the target device. |
| Set Password | Sets the password of the target device. |
| Set TUNID | Sets TUNID of the target device. |
| Read TUNID | Reads TUNID of the target device. |
| Lock | Locks the configuration for the target device. |

| Button name | Description |
|---|---|
| Unlock | Unlocks the configuration for the target device. |
| Mode Change (Idle) | Change the target device's mode to Idle. |
| Mode Change (Executing) | Change the target device's mode to Executing. |
| Restart | Restarts the target device. |
| Clear Memory | Clears the settings of the target device. |
| Device Information | Reads the device information of the target device. |
| Generic Message | Sends desired CIP messages to the target device. |

**Precautions for Correct Use**

- This function depends on the operation specifications of the target device. Some operations are not supported depending on the device functions.
- If you set TUNID to the target device, make sure that the correct TUNID is set to the intended device under your responsibility. Also, ensure that all TUNIDs are unique across the system.

# A-20 Version Information

This section describes the compatibility between unit versions of the Safety Control Unit and the version of the Sysmac Studio, as well as the functions supported by the Sysmac Studio.

## A-20-1 Relationship between Unit Versions and Sysmac Studio Versions

This section describes the possible combinations of versions of Safety Control Units, NJ/NX-series CPU Units, Communications Coupler Units, Communication Control Units, Sysmac Studio, and other NX Units.

Refer to the manuals of each unit and Sysmac Studio for the relationship of functions and versions of NJ/NX-series CPU Units, Communications Coupler Units, Communication Control Units, and Sysmac Studio.
Refer to the user's manual of the connected Communications Coupler Unit for details on the relationship between Slave Terminal functions and versions.

### EtherCAT Slave Terminal

This configuration is used to connect the Safety Control Unit to the EtherCAT Coupler Unit, and the EtherCAT Slave Terminal to the built-in EtherCAT master of the CPU Unit via EtherCAT.

| Safety Control Unit model and version | | NX bus master: EtherCAT Coupler Unit | | |
|---|---|---|---|---|
| Model | Unit version | Communications Coupler Unit | CPU Unit | Sysmac Studio |
| NX-SL3300 | Ver. 1.0 | Ver. 1.1*1 | Ver. 1.06 | Ver. 1.07 |
| | Ver. 1.1 | | | Ver. 1.10 |
| NX-SL3500 | Ver. 1.0 | Ver. 1.2*1 | Ver. 1.07 | Ver. 1.08 |
| | Ver. .1.1 | | | Ver. 1.10 |
| NX-SL5500 | Ver. 1.3 | --- | --- | --- |
| | Ver. 1.4 | --- | --- | --- |
| NX-SL5700 | Ver. 1.2 | --- | --- | --- |
| | Ver. 1.3 | --- | --- | --- |
| | Ver. 1.4 | --- | --- | --- |
| NX-SIH400 | Ver. 1.0 | Ver. 1.1*1 | Ver. 1.06 | Ver. 1.07 |
| | Ver. 1.1 | | | Ver. 1.10 |
| NX-SID800 | Ver. 1.0 | | | Ver. 1.07 |
| NX-SOH200 | | | | |
| NX-SOD400 | | | | |

*1. Some Units do not have all of the versions given in the above table. If a Unit does not have the specified version, support is provided by the oldest available version after the specified version. Refer to the user's manuals of the specific Units for the relationship between models and versions.

## CPU Rack

This configuration is used to connect the Safety Control Unit to the CPU Unit or Communication Control Unit.

| Safety Control Unit model and version | | NX bus master: CPU Unit | |
|---|---|---|---|
| Model | Unit version | NX502 CPU Unit | Sysmac Studio |
| NX-SL3300 | Ver.1.0 | Ver.1.60 | Ver.1.54 |
| | Ver.1.1 | | |
| NX-SL3500 | Ver.1.0 | | |
| | Ver.1.1 | | |
| NX-SL5500 | Ver.1.3 | | |
| | Ver.1.4 | | |
| NX-SL5700 | Ver.1.2 | --- | --- |
| | Ver.1.3 | Ver.1.60 | Ver.1.54 |
| | Ver.1.4 | | |
| NX-SIH400 | Ver.1.0 | | |
| | Ver.1.1 | | |
| NX-SID800 | Ver.1.0 | | |
| NX-SOH200 | | | |
| NX-SOD400 | | | |

| Safety Control Unit model and version | | NX bus master: CPU Unit | |
|---|---|---|---|
| Model | Unit version | NX102 CPU Unit | Sysmac Studio |
| NX-SL3300 | Ver. 1.0 | Ver. 1.30 | Ver. 1.22 |
| | Ver. 1.1 | | |
| NX-SL3500 | Ver. 1.0 | | |
| | Ver. 1.1 | | |
| NX-SL5500 | Ver. 1.3 | Ver. 1.31 | Ver. 1.24 |
| | Ver. 1.4 | | Ver. 1.40 |
| NX-SL5700 | Ver. 1.2 | --- | --- |
| | Ver. 1.3 | Ver. 1.31 | Ver. 1.24 |
| | Ver. 1.4 | | Ver. 1.40 |
| NX-SIH400 | Ver. 1.0 | Ver. 1.30 | Ver. 1.22 |
| | Ver. 1.1 | | |
| NX-SID800 | Ver. 1.0 | | |
| NX-SOH200 | | | |
| NX-SOD400 | | | |

| Safety Control Unit model and version | | NX bus master: Communication Control Unit | |
|---|---|---|---|
| Model | Unit version | NX-CSG□□□ | Sysmac Studio |
| NX-SL3300 | Ver. 1.0 | --- | --- |
| NX-SL3300 | Ver. 1.1 | | |
| NX-SL3500 | Ver. 1.0 | | |
| NX-SL3500 | Ver. 1.1 | | |
| NX-SL5500 | Ver. 1.3 | Ver. 1.01 | Ver. 1.24 |
| | Ver. 1.4 | | Ver. 1.40 |

| Safety Control Unit model and version | | NX bus master: Communication Control Unit | |
|---|---|---|---|
| Model | Unit version | NX-CSG□□□ | Sysmac Studio |
| NX-SL5700 | Ver. 1.2 | Ver. 1.00 only | Ver. 1.24 |
| | Ver. 1.3 | Ver. 1.01 | |
| | Ver. 1.4 | | Ver. 1.40 |
| NX-SIH400 | Ver. 1.0 | Ver. 1.00 | Ver. 1.24 |
| NX-SIH400 | Ver. 1.1 | | |
| NX-SID800 | Ver. 1.0 | | |
| NX-SOH200 | | | |
| NX-SOD400 | | | |

## EtherNet/IP Slave Terminal

This configuration is used to connect the Safety Control Unit to the EtherNet/IP Coupler Unit.

| Safety Control Unit model and version | | NX bus master: EtherNet/IP Coupler Unit | |
|---|---|---|---|
| Model | Unit version | Communications Coupler Unit | Sysmac Studio |
| NX-SL3300 | Ver. 1.0 | --- | --- |
| | Ver. 1.1 | Ver. 1.0 | Ver. 1.10 |
| NX-SL3500 | Ver. 1.0 | --- | --- |
| | Ver. 1.1 | --- | --- |
| NX-SL5500 | Ver. 1.3 | --- | --- |
| | Ver. 1.4 | --- | --- |
| NX-SL5700 | Ver. 1.2 | --- | --- |
| | Ver. 1.3 | --- | --- |
| | Ver. 1.4 | --- | --- |
| NX-SIH400 | Ver. 1.0 | --- | --- |
| | Ver. 1.1 | Ver. 1.0 | Ver. 1.10 |
| NX-SID800 | Ver. 1.0 | | |
| NX-SOH200 | | | |
| NX-SOD400 | | | |

## Units That Support Communications between NX Units

This section describes how to set up communications between the NX Units of the EtherNet/IP Slave Terminals.

### ● Digital I/O Units

The table below shows the NX Unit models that support communications between the NX Units as well as the combinations of the Unit version and Sysmac Studio version.

| Model | Unit version | Sysmac Studio |
|---|---|---|
| NX-ID3317 | Ver. 1.0 | Ver. 1.10 or higher |
| NX-ID3343 | Ver. 1.0 | Ver. 1.10 or higher |
| NX-ID3344 | Ver. 1.0 | Ver. 1.10 or higher |
| NX-ID3417 | Ver. 1.0 | Ver. 1.10 or higher |
| NX-ID3443 | Ver. 1.0 | Ver. 1.10 or higher |

| Model | Unit version | Sysmac Studio |
|---|---|---|
| NX-ID4342 | Ver. 1.0 | Ver. 1.10 or higher |
| NX-ID4442 | Ver. 1.0 | Ver. 1.10 or higher |
| NX-ID5142-1 | Ver. 1.0 | Ver. 1.22 or higher |
| NX-ID5142-5 | Ver. 1.0 | Ver. 1.10 or higher |
| NX-ID5342 | Ver. 1.0 | Ver. 1.10 or higher |
| NX-ID5442 | Ver. 1.0 | Ver. 1.10 or higher |
| NX-ID6142-5 | Ver. 1.0 | Ver. 1.10 or higher |
| NX-ID6142-6 | Ver. 1.0 | Ver. 1.22 or higher |
| NX-IA3117 | Ver. 1.0 | Ver. 1.10 or higher |
| NX-OD3121 | Ver. 1.0 | Ver. 1.10 or higher |
| NX-OD3153 | Ver. 1.0 | Ver. 1.10 or higher |
| NX-OD3256 | Ver. 1.0 | Ver. 1.10 or higher |
| NX-OD3257 | Ver. 1.0 | Ver. 1.10 or higher |
| NX-OD3268 | Ver. 1.0 | Ver. 1.22 or higher |
| NX-OD4121 | Ver. 1.0 | Ver. 1.10 or higher |
| NX-OD4256 | Ver. 1.0 | Ver. 1.10 or higher |
| NX-OD5121 | Ver. 1.0 | Ver. 1.22 or higher |
| NX-OD5121-1 | Ver. 1.0 | Ver. 1.22 or higher |
| NX-OD5121-5 | Ver. 1.0 | Ver. 1.10 or higher |
| NX-OD5256 | Ver. 1.0 | Ver. 1.10 or higher |
| NX-OD5256-1 | Ver. 1.0 | Ver. 1.22 or higher |
| NX-OD5256-5 | Ver. 1.0 | Ver. 1.10 or higher |
| NX-OD6121-5 | Ver. 1.0 | Ver. 1.10 or higher |
| NX-OD6121-6 | Ver. 1.0 | Ver. 1.22 or higher |
| NX-OD6256-5 | Ver. 1.0 | Ver. 1.10 or higher |
| NX-OC2633 | Ver. 1.0 | Ver. 1.10 or higher |
| NX-OC2733 | Ver. 1.0 | Ver. 1.10 or higher |
| NX-OC4633 | Ver. 1.0 | Ver. 1.22 or higher |
| NX-MD6121-5 | Ver. 1.0 | Ver. 1.10 or higher |
| NX-MD6121-6 | Ver. 1.0 | Ver. 1.22 or higher |
| NX-MD6256-5 | Ver. 1.0 | Ver. 1.10 or higher |

The data types in the following table are supported for communications between NX Units.

| Data Type |
|---|
| BOOL |
| BYTE |
| INT |
| WORD |
| DINT |

## ● Analog I/O Units

The table below shows the NX Unit models that support communications between the NX Units as well as the combinations of the Unit version and Sysmac Studio version.

| Model | Unit version | Sysmac Studio |
|---|---|---|
| NX-AD2203 | Ver. 1.0 | Ver. 1.10 or higher |
| NX-AD2204 | Ver. 1.0 | Ver. 1.10 or higher |

| Model | Unit version | Sysmac Studio |
|---|---|---|
| NX-AD2208 | Ver. 1.0 | Ver. 1.10 or higher |
| NX-AD2603 | Ver. 1.0 | Ver. 1.10 or higher |
| NX-AD2604 | Ver. 1.0 | Ver. 1.10 or higher |
| NX-AD2608 | Ver. 1.0 | Ver. 1.10 or higher |
| NX-AD3203 | Ver. 1.0 | Ver. 1.10 or higher |
| NX-AD3204 | Ver. 1.0 | Ver. 1.10 or higher |
| NX-AD3208 | Ver. 1.0 | Ver. 1.10 or higher |
| NX-AD3603 | Ver. 1.0 | Ver. 1.10 or higher |
| NX-AD3604 | Ver. 1.0 | Ver. 1.10 or higher |
| NX-AD3608 | Ver. 1.0 | Ver. 1.10 or higher |
| NX-AD4203 | Ver. 1.0 | Ver. 1.10 or higher |
| NX-AD4204 | Ver. 1.0 | Ver. 1.10 or higher |
| NX-AD4208 | Ver. 1.0 | Ver. 1.10 or higher |
| NX-AD4603 | Ver. 1.0 | Ver. 1.10 or higher |
| NX-AD4604 | Ver. 1.0 | Ver. 1.10 or higher |
| NX-AD4608 | Ver. 1.0 | Ver. 1.10 or higher |
| NX-DA2203 | Ver. 1.0 | Ver. 1.10 or higher |
| NX-DA2205 | Ver. 1.0 | Ver. 1.10 or higher |
| NX-DA2603 | Ver. 1.0 | Ver. 1.10 or higher |
| NX-DA2605 | Ver. 1.0 | Ver. 1.10 or higher |
| NX-DA3203 | Ver. 1.0 | Ver. 1.10 or higher |
| NX-DA3205 | Ver. 1.0 | Ver. 1.10 or higher |
| NX-DA3603 | Ver. 1.0 | Ver. 1.10 or higher |
| NX-DA3605 | Ver. 1.0 | Ver. 1.10 or higher |

The data types in the following table are supported for communications between NX Units.

| Data Type |
|---|
| BOOL |
| BYTE |
| INT |
| WORD |
| DINT |

## ● Temperature Input Units

The table below shows the NX Unit models that support communications between the NX Units as well as the combinations of the Unit version and Sysmac Studio version.

| Model | Unit version | Sysmac Studio |
|---|---|---|
| NX-TS2101 | Ver. 1.0 | Ver. 1.22 or higher |
| | Ver. 1.1 | Ver. 1.22 or higher |
| NX-TS2102 | Ver. 1.1 | Ver. 1.22 or higher |
| NX-TS2104 | Ver. 1.1 | Ver. 1.22 or higher |
| NX-TS2201 | Ver. 1.0 | Ver. 1.22 or higher |
| | Ver. 1.1 | Ver. 1.22 or higher |
| NX-TS2202 | Ver. 1.1 | Ver. 1.22 or higher |
| NX-TS2204 | Ver. 1.1 | Ver. 1.22 or higher |

| Model | Unit version | Sysmac Studio |
|---|---|---|
| NX-TS3101 | Ver. 1.0 | Ver. 1.22 or higher |
| | Ver. 1.1 | Ver. 1.22 or higher |
| NX-TS3102 | Ver. 1.1 | Ver. 1.22 or higher |
| NX-TS3104 | Ver. 1.1 | Ver. 1.22 or higher |
| NX-TS3201 | Ver. 1.0 | Ver. 1.22 or higher |
| | Ver. 1.1 | Ver. 1.22 or higher |
| NX-TS3202 | Ver. 1.1 | Ver. 1.22 or higher |
| NX-TS3204 | Ver. 1.1 | Ver. 1.22 or higher |

The data types in the following table are supported for communications between NX Units.

| Data Type |
|---|
| BOOL |
| BYTE |
| INT |
| WORD |
| DINT |

## A-20-2 Functions That Were Added or Changed for Each Version

This section gives the Safety Control Unit-related functions that were added or changed for each version.

| Description | Change or addition | Corresponding unit version/ version | | Reference |
|---|---|---|---|---|
| | | Sysmac Studio | Safety CPU Unit NX-SL□□□□ | |
| Supported model added.<br>• NX-SL3500 | Addition | Ver.1.08 or higher | Ver. 1.0 or later | *2-1 Safety CPU Unit* on page 2-2 |
| Offline debugging | Addition | | | *6-8 Offline Debugging* on page 6-98 |
| Function to change models | Addition | Ver. 1.10 or higher | Ver. 1.0 or later | *5-3-3 Procedure to Change the Model of the Safety Control Unit* on page 5-10 |
| Monitoring memory usage | Addition | | | *6-7 Monitoring Memory Usage for Safety Control Unit* on page 6-96 |
| Setting the initial values of variables | Addition | | | *6-8-5 Setting the Initial Values of Variables* on page 6-101 |
| Feedback setting | Addition | | | *6-8-6 Feedback Settings* on page 6-102 |
| Setting watchdog times for safety process data communications | Addition | | Ver. 1.10 or later | *8-3-4 Changing FSoE Watchdog Timers* on page 8-10 |
| Transferring safety-validated safety application data | Addition | | | *7-3-3 Executable Functions in Each Mode of the Safety CPU Unit* on page 7-14 |

| Description | Change or addition | Corresponding unit version/ version | | Reference |
|---|---|---|---|---|
| | | Sysmac Studio | Safety CPU Unit NX-SL□□□□ | |
| Importing/exporting user-defined function blocks | Addition | Ver. 1.12 or higher | Ver. 1.0 or later | *6-5-2 Registering POUs* on page 6-27 |
| Data protection | Addition | | | *7-10-2 Data Protection* on page 7-60 |
| Simple Automatic Test | Addition | Ver. 1.15 or higher | Ver. 1.0 or later | *6-8-7 Simple Automatic Test* on page 6-103 |
| Importing/exporting programs | Addition | Ver. 1.17 or higher | Ver. 1.0 or higher | *6-5-2 Registering POUs* on page 6-27 |
| Program data protection. | Addition | | | *7-10-2 Data Protection* on page 7-60 |
| Checking external device wiring | Addition | | | *7-5 Checking External Device Wiring* on page 7-20 |
| Supported model added.<br>• NX-SL5500<br>• NX-SL5700 | Addition | Ver. 1.24 or higher | Ver. 1.3 or later | *2-1 Safety CPU Unit* on page 2-2 |
| I/O data enable flag for CIP Safety connections | Addition | Ver. 1.40 or higher | Ver. 1.4 or higher | *A-17 I/O Data Enable Flag for CIP Safety Connections* on page A-107 |
| Safety POU Data Protection Version | Addition | Ver. 1.54 or higher | Ver. 1.0 or later | *7-10-2 Data Protection* on page 7-60 |
| Batch import/export of CIP Safety connection settings | Addition | Ver. 1.54 or higher | Ver. 1.2 or later | *5-4-2 Setting Up the CIP Safety Communications* on page 5-13 |

# I

# Index

# Index


## A

Adjusting the communications load..... 9-2
Adjusting the device bandwidth usage..... 9-7
Applicable Wire.....3-10

## C

Checking Bandwidth Usage for Tag Data Links..... 9-4
Checking the Device Bandwidth Usage of the CIP Safety Routing.....9-6
Coding Pin..... 3-20
crimping tool.....3-11

## E

Extended Status Code..... A-83

## F

Ferrule.....3-11

## G

General Status Codes.....A-81
Ground terminal.....3-11, 3-12

## I

incorrect attachment prevention hole..... 3-22

## M

Marker..... 3-4

## O

optimal angle for insertion..... 3-16, 3-17

## P

Periodic Inspection Item..... 13-2
Preventing Incorrect Attachment..... 3-20

## R

Relationship between the Number of Packets Used per Second and Packet Intervals..... 9-6

## S

safety data types ..... 6-15
safety input functions.....4-22
  dual channel evaluation..... 4-24
    dual channel .....4-24, 4-41
    single channel ..... 4-24, 4-41
  input filters.....4-30
  test pulse.....4-23
safety output functions.....4-39
  dual channel evaluation..... 4-41
  test pulse evaluation..... 4-40
safety program.....6-3
Securing Wires..... 3-13
Solid Wire..... 3-12
standard data types ..... 6-15

## T

Twisted Wire.....3-12

## W

Wiring Terminal.....3-10

**OMRON Corporation** **Industrial Automation Company**

**Kyoto, JAPAN** **Contact : www.ia.omron.com**

***Regional Headquarters***

**OMRON EUROPE B.V.**
Wegalaan 67-69, 2132 JD Hoofddorp
The Netherlands
Tel: (31) 2356-81-300 Fax: (31) 2356-81-388

**OMRON ELECTRONICS LLC**
2895 Greenspoint Parkway, Suite 200
Hoffman Estates, IL 60169 U.S.A.
Tel: (1) 847-843-7900 Fax: (1) 847-843-7787

**OMRON ASIA PACIFIC PTE. LTD.**
438B Alexandra Road, #08-01/02 Alexandra
Technopark, Singapore 119968
Tel: (65) 6835-3011 Fax: (65) 6835-3011

**OMRON (CHINA) CO., LTD.**
Room 2211, Bank of China Tower,
200 Yin Cheng Zhong Road,
PuDong New Area, Shanghai, 200120, China
Tel: (86) 21-6023-0333 Fax: (86) 21-5037-2388

**Authorized Distributor:**

©OMRON Corporation 2013-2026 All Rights Reserved.
In the interest of product improvement,
specifications are subject to change without notice.

**Cat. No. Z930-E1-32** 1026