OMRON

Machine Automation Controller

NX-series

Safety Control Unit

User's Manual

NX-SL□□□□
NX-SI□□□□
NX-SO□□□□

Safety Control Unit

SYSMAC
always in control

Z930-E1-01

Trademarks

Introduction

Thank you for purchasing Machine Automation Controller NX-series Safety Control Units.

This manual contains information that is necessary to use the NX-series Safety Control Units. Please read this manual and make sure you understand the functionality and performance of the NX-series Safety Control Units before you attempt to use them 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 B3503.

Applicable Products

This manual covers the following products.

- NX-series Safety Control Units

  NX-SL□□□□

  NX-SID□□□ and NX-SIH□□□

  NX-SOD□□□ and NX-SOH□□□

CONTENTS

Section 6 System Configuration and Setup

Section 7 Programming

Section 8 Checking Operation and Actual Operation

Section 9 Troubleshooting

Section 10 Maintenance and Inspection

Appendix

Index

Relevant Manuals

Two manuals, the *NX-series Safety Control Unit User’s Manual* and the *NX-series Safety Control Unit Instructions Reference Manual*, provide basic information on the NX-series Safety Control Units.
Most operations are performed from the Sysmac Studio Automation Software.
Refer to the *Sysmac Studio Version 1 Operation Manual* (Cat. No. W504) for information on the Sysmac Studio.

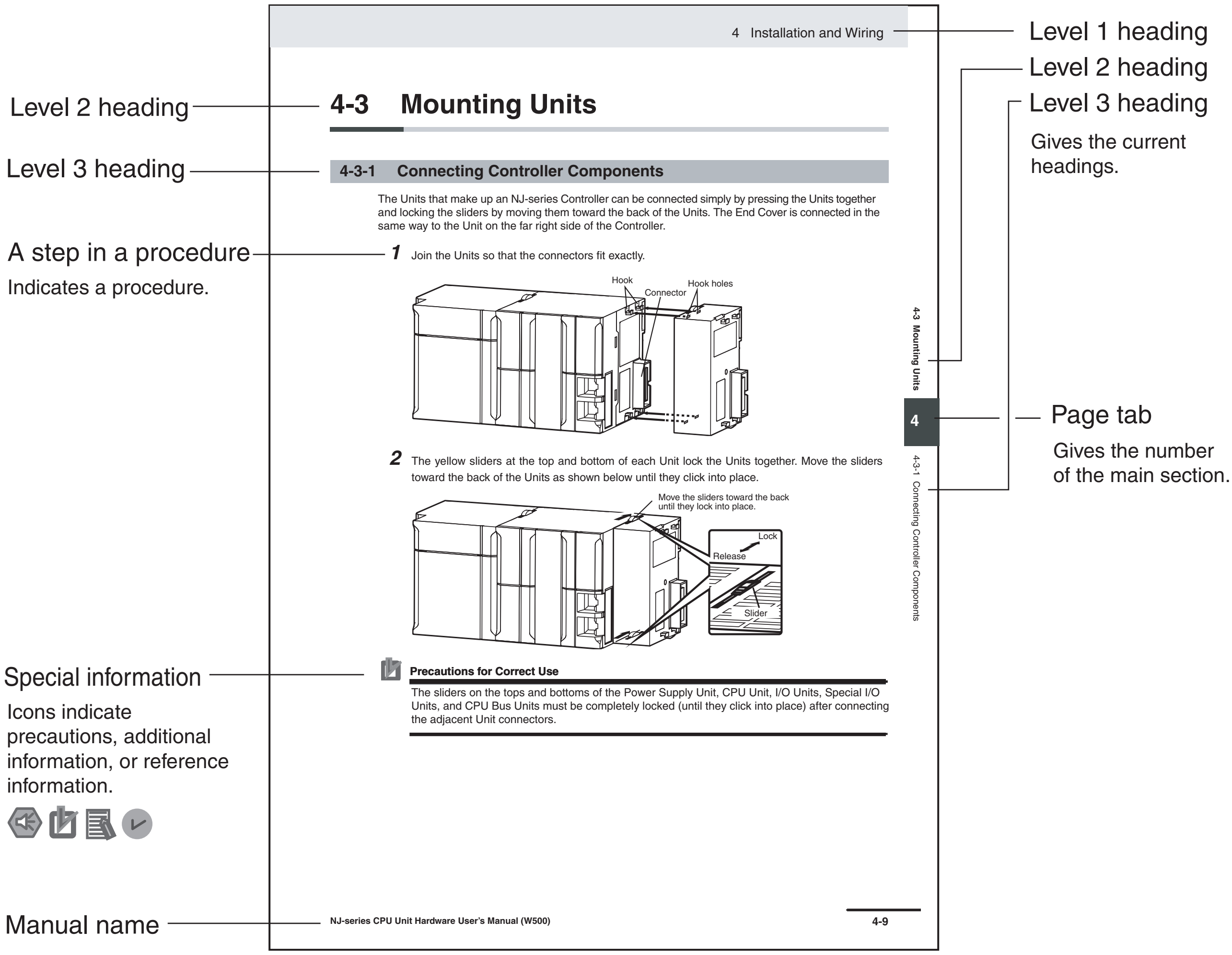
Purpose of use	Manuals								
	NX Series				NJ Series				
	Basic information								
	NX-series Safety Control Unit User’s Manual	NX-series Safety Control Unit Instructions Reference Manual	NX-series EtherCAT Coupler Unit User’s Manual	NX-series Data Reference Manual	NJ-series CPU Unit Hardware User’s Manual	NJ-series CPU Unit Software User’s Manual	NJ-series CPU Unit Built-in EtherCAT Port User’s Manual	NJ-series Instructions Reference Manual	NJ-series Troubleshooting Manual
Learning about Safety Control Units	●								
Mounting, installing, and making hardware settings for Safety Control Units	●		●	●			●		
Making software settings for Safety Control Units	●			●			●		
Creating safety programs	●	●							
Verifying and debugging safety programs	●	●							
Troubleshooting Safety Control Units	●		▲		▲	▲	▲		▲
Maintaining Safety Control Units	●		●						
Learning the application methods of NJ-series Controllers					●	●	●	●	

*1 The *NJ-series Troubleshooting Manual* introduces the error management concepts and error items. Refer to the manuals that are indicated with triangles for details on errors for the corresponding Units.

Manual Structure

Page Structure and Icons

The following page structure and icons are used in this manual.



Note 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.

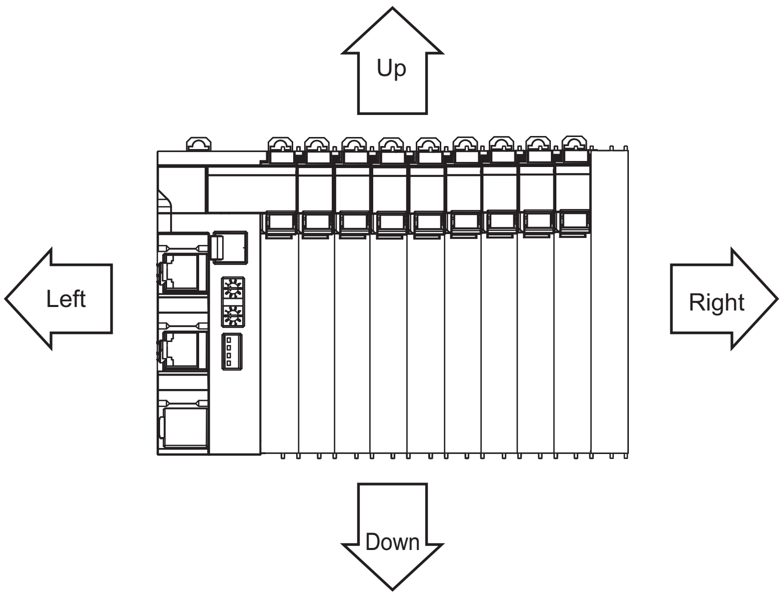
Version Information

Information on differences in specifications and functionality for CPU Units and EtherCAT Coupler Units with different unit versions and for different versions of the Sysmac Studio is given.

Note References are provided to more detailed or related information.

Precaution on Terminology

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



Read and Understand this Manual

Please read and understand this manual before using the products. Please consult your OMRON representative if you have any questions or comments.

Warranty and Limitations of Liability

WARRANTY

OMRON's exclusive warranty is that the products are free from defects in materials and workmanship for a period of one year (or other period if specified) from date of sale by OMRON.

OMRON MAKES NO WARRANTY OR REPRESENTATION, EXPRESS OR IMPLIED, REGARDING NON-INFRINGEMENT, MERCHANTABILITY, OR FITNESS FOR PARTICULAR PURPOSE OF THE PRODUCTS. ANY BUYER OR USER ACKNOWLEDGES THAT THE BUYER OR USER ALONE HAS DETERMINED THAT THE PRODUCTS WILL SUITABLY MEET THE REQUIREMENTS OF THEIR INTENDED USE. OMRON DISCLAIMS ALL OTHER WARRANTIES, EXPRESS OR IMPLIED.

LIMITATIONS OF LIABILITY

OMRON SHALL NOT BE RESPONSIBLE FOR SPECIAL, INDIRECT, OR CONSEQUENTIAL DAMAGES, LOSS OF PROFITS OR COMMERCIAL LOSS IN ANY WAY CONNECTED WITH THE PRODUCTS, WHETHER SUCH CLAIM IS BASED ON CONTRACT, WARRANTY, NEGLIGENCE, OR STRICT LIABILITY.

In no event shall the responsibility of OMRON for any act exceed the individual price of the product on which liability is asserted.

IN NO EVENT SHALL OMRON BE RESPONSIBLE FOR WARRANTY, REPAIR, OR OTHER CLAIMS 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 OR REPAIR.

Application Considerations

SUITABILITY FOR USE

OMRON shall not be responsible for conformity with any standards, codes, or regulations that apply to the combination of products in the customer's application or use of the products.

At the customer's request, OMRON will provide applicable third party certification documents identifying ratings and limitations of use that apply to the products. This information by itself is not sufficient for a complete determination of the suitability of the products in combination with the end product, machine, system, or other application or use.

The following are some examples of applications for which particular attention must be given. This is not intended to be an exhaustive list of all possible uses of the products, nor is it intended to imply that the uses listed may be suitable for the products:

- Outdoor use, uses involving potential chemical contamination or electrical interference, or conditions or uses not described in this manual.
- Nuclear energy control systems, combustion systems, railroad systems, aviation systems, medical equipment, amusement machines, vehicles, safety equipment, and installations subject to separate industry or government regulations.
- Systems, machines, and equipment that could present a risk to life or property.

Please know and observe all prohibitions of use applicable to the products.

NEVER USE THE PRODUCTS FOR AN APPLICATION INVOLVING SERIOUS RISK TO LIFE OR PROPERTY WITHOUT ENSURING THAT THE SYSTEM AS A WHOLE HAS BEEN DESIGNED TO ADDRESS THE RISKS, AND THAT THE OMRON PRODUCTS ARE PROPERLY RATED AND INSTALLED FOR THE INTENDED USE WITHIN THE OVERALL EQUIPMENT OR SYSTEM.

PROGRAMMABLE PRODUCTS

OMRON shall not be responsible for the user's programming of a programmable product, or any consequence thereof.

Disclaimers

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 model numbers when published ratings or features are changed, or when significant construction changes are made. However, some specifications of the products may be changed without any notice. When in doubt, special model numbers may be assigned to fix or establish key specifications for your application on your request. Please consult with your OMRON representative at any time to confirm actual specifications of purchased products.

DIMENSIONS AND WEIGHTS
Dimensions and weights are nominal and are not to be used for manufacturing purposes, even when tolerances are shown.

PERFORMANCE DATA
Performance data given in this manual 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 users must correlate it to actual application requirements. Actual performance is subject to the OMRON Warranty and Limitations of Liability.

ERRORS AND OMISSIONS
The information in this manual has been carefully checked and is believed to be accurate; however, no responsibility is assumed for clerical, typographical, or proofreading errors, or omissions.

Safety Precautions

Definition of Precautionary Information

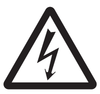
The following notation is used in this manual to provide precautions required to ensure safe usage of the NX-series Safety Control Units.

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. The specific operation is shown in the circle and explained in text. This example indicates prohibiting disassembly.
	The triangle symbol indicates precautions (including warnings). The specific operation is shown in the triangle and explained in text. This example indicates a precaution for electric shock.
	The triangle symbol indicates precautions (including warnings). The specific operation is shown in the triangle and explained in text. This example indicates a general precaution.
	The filled circle symbol indicates operations that you must do. The specific operation is shown in the circle and explained in text. 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/TS 62046 Application of Protective Equipment to Detect the Presence of Persons

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

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.

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.

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

Testing Operation

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.

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.

During Power Supply

Do not attempt to take any Unit apart. In particular, high-voltage parts are present in the Power Supply Unit 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 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.

Precautions for Safe Use

Transporting

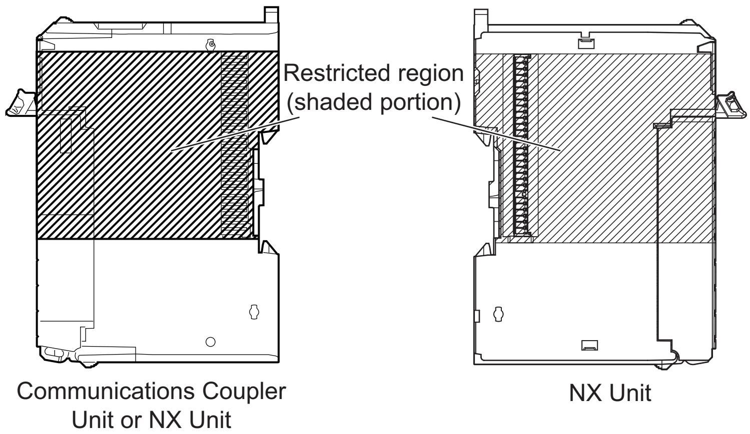
- Do not drop any Unit or subject it to abnormal vibration or shock. Doing so may result in Unit malfunction or burning.
- When transporting any Unit, use the special packing box for it. Also, do not subject the Unit to excessive vibration or shock during transportation.

Mounting

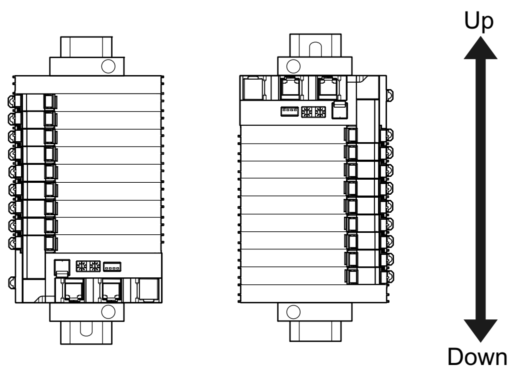
- Mount terminal blocks and connectors only after checking the mounting location carefully. Be sure that the terminal blocks, expansion cables, and other items with locking devices are properly locked into place.

Installation

- Do not apply labels or tape to the Units. 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 write on the Communications Coupler Unit or 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 Slave Terminal.

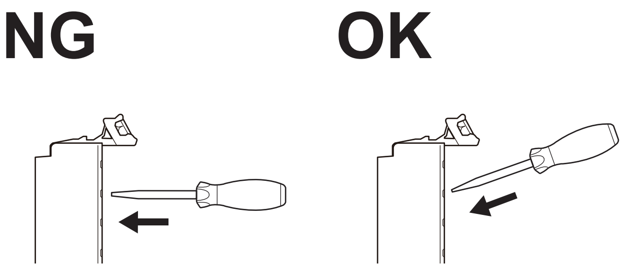


- For the installation orientations in the following figure, support the cables, e.g., with a duct, so that the End Plate on the bottom is not subjected to the weight of the cables. The weight of the cables may cause the bottom End Plate to slide downward so that the Slave Terminal is no longer secured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

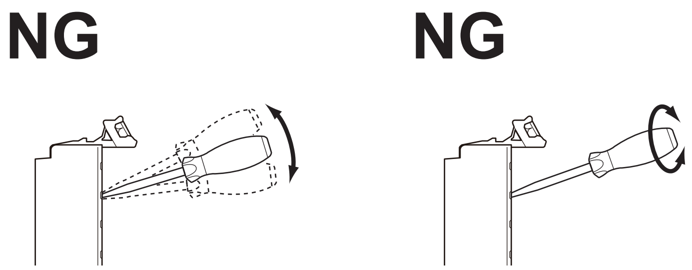


Wiring

- 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.
- Do not bend the cable past its natural bending radius or pull in it with excessive force. Do not place any heavy objects on the cable. Doing so may severe the cable.
- 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 break the terminal block.



- 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 break the terminal block.



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.

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.

Startup

- Double-check all wiring 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

- Before you start operation, always register the NX Units that are connected to the Communications Coupler Unit in the host communications master as the Unit configuration information.
- 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

- Always turn OFF the external power supply to the Units before attempting any of the following.

  Mounting or removing an NX Unit, Communications Coupler Unit, or CPU Unit
  Assembling Units
  Setting DIP switches or rotary switches
  Connecting or wiring cables
  Attaching or removing terminal blocks or connectors

  Units that supply power continue to supply power to the Units for up to several seconds after the power supply is turned OFF. The PWR indicator remains lit as long as power is supplied. Confirm that the PWR indicator is not lit before you perform any of the above.

General Communications

- Do not exceed the ranges that are given in the specifications for the communications distance and number of connected Units.

Unit Replacement

- 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, Mounting, and Wiring

- Follow the instructions in this manual to correctly perform installation.
- Do not operate or store the Units in the following locations. Operation may stop or malfunction may occur.

  Locations subject to direct sunlight
  Locations subject to temperatures or humidity outside the range specified in the specifications
  Locations subject to condensation as the result of severe changes in temperature
  Locations subject to corrosive or flammable gases
  Locations subject to dust (especially iron dust) or salts
  Locations subject to exposure to water, oil, or chemicals
  Locations subject to shock or vibration
  Locations subject to static electricity or other forms of noise
- Take appropriate and sufficient countermeasures during installation in the following locations.

  Locations subject to strong, high-frequency noise
  Locations subject to static electricity or other forms of noise
  Locations subject to strong electromagnetic fields
  Locations subject to possible exposure to radioactivity
  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 locations where the power supply is unstable.

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.

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 variables to publish to a Standard CPU Unit, the device variable assignments to the Safety 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.
- For security purposes, we recommend that you set a password for the Safety CPU Unit and the project file.

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 have obtained certification for the following standards.

Certification body	Standards	
TÜV Rheinland*1	• EN ISO 13849-1:2008+AC:2009	• EN 61000-6-2:2005
	• EN ISO 13849-2:2012	• EN 61000-6-4:2007
	• IEC 61508 parts 1-7:2010	• NFPA79:2012
	• EN 62061:2005	• ANSI RIA 15.06-1999
	• EN 61131-2:2007	• ANSI B11.19:2010
	• EN ISO 13850:2008	• UL1998
	• EN 60204-1:2006+A1:2009+AC:2010	• IEC 61326-3-1:2008
UL	• cULus: Listed (UL508) and ANSI/ISA 12.12.01	

*1. Certification was received for applications in which OMRON FSoE devices are connected to each other.

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, EN 62061, Safety Standard for Safety Instrumented Systems (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 C-Tick and KC compliance.

Conformance to EC Directives

Applicable Directives

- EMC Directive
- Machinery Directive

Concepts

EMC Directives

OMRON devices that comply with EC 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 EC 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 13849-1:2008 and EN 62061 SIL CL3

Conformance to EC Directives

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

- The NX-series Units must be installed within a metallic control cabinet.
- You must meet the following conditions for the DC power supplies that are connected as the Unit power supplies and I/O power supplies for the NX-series Units.
  (a) Use reinforced insulation or double insulation.
  (b) Ensure an output hold time of 20 ms min.
  (c) Use an SELV power supply that meets the requirements of IEC/EN 60950-1 and EN 50178.
  Do not allow the power supply cable length to exceed 3 m.

  We recommend that you use the OMRON S8JX-series Power Supplies. EMC standard compliance was confirmed for the recommended Power Supplies.
- NX-series Units that comply with EC Directives also conform to the Common Emission Standard (EN 61131-2). 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 EC 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 and EN 62061

EN ISO 13849-1 and EN 62061 require process management to avoid system interference and to simplify reading, understanding, testing, and maintaining software. 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, process 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.

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 KC Standards

Observe the following precaution if you use NX-series Units in Korea.

A 급 기기 (업무용 방송통신기자재)
이 기기는 업무용(A 급) 전자파적합기기로서 판매자 또는 사용자는 이 점을 주의하시기 바라며, 가정외의 지역에서 사용하는 것을 목적으로 합니다.

Class A Device (Broadcasting Communications Device for Office Use)

This device obtained EMC registration for office use (Class A), and it is intended to be used in places other than homes.

Sellers and/or users need to take note of this.

Software Licenses and Copyrights

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

Unit Versions

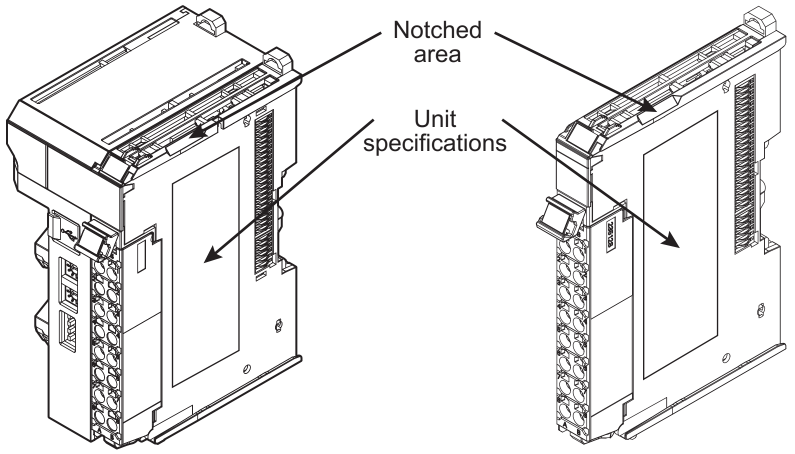
This section describes the notation that is used for unit versions, the confirmation method for unit versions, and the relationship between unit versions and Sysmac Studio versions.

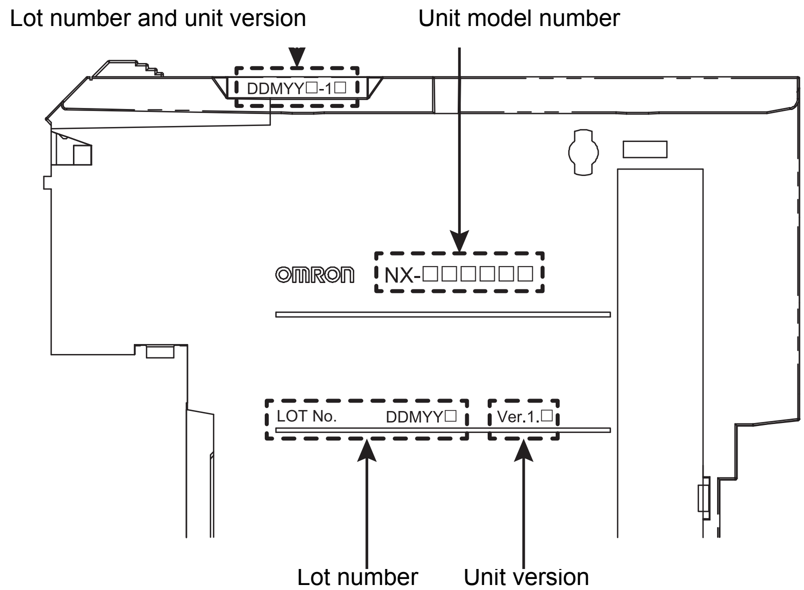
Unit Versions

A “unit version” 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. DDMYY□: Lot number, □: Used by OMRON. “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. • DDMYY□: Lot number, □: Used by OMRON. "M" gives the month (1 to 9: January to September, X: October, Y: November, Z: December) • 1□: Unit version The decimal portion of the unit version is omitted. (It is provided in the Unit specifications.)

Confirming Unit Versions with the Sysmac Studio

You can use the Unit Production Information on the Sysmac Studio to check the unit versions EtherCAT Coupler Unit and NX Units.

1 Double-click **EtherCAT** under **Configurations and Setup** in the Multiview Explorer, and then double-click the EtherCAT Coupler Unit. Or, right-click the EtherCAT Coupler Unit and select ***Edit*** from the menu.
The Edit Slave Terminal Configuration Tab Page is displayed.

You can also display the Edit Slave Terminal Configuration Tab Page with any of the following operations.

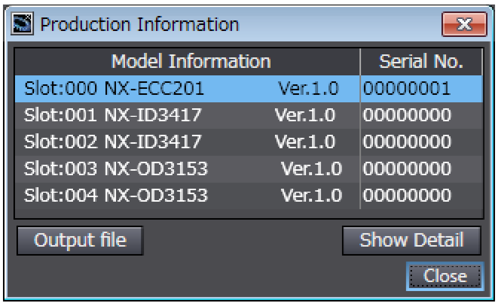
Double-click **EtherCAT** under **Configurations and Setup** in the Multiview Explorer, right-click the EtherCAT Coupler Unit in the EtherCAT Configuration Edit Tab Page, and select ***Edit Slave Terminal Configuration***.

Or, select the EtherCAT Coupler Unit on the EtherCAT Configuration Edit Tab Page click the **Edit Slave Terminal Configuration** Button.

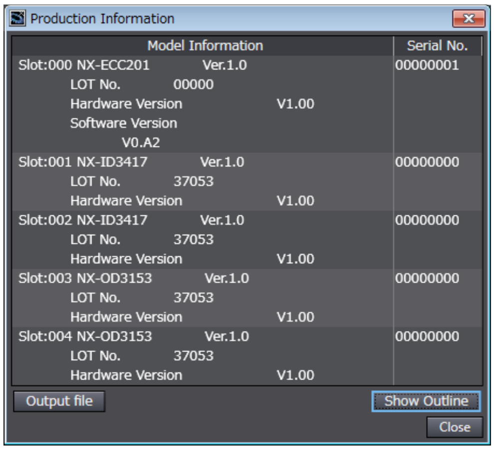
2 Go online.

3 Right-click the EtherCAT Coupler Unit and select ***Display Production Information*** from the menu.

The Production Information Dialog Box is displayed.



Simple Display



Detailed Display

In this example, "Ver.1.0" is displayed next to the Unit model.

The following items are displayed.

- Slot number
- Unit model number
- Unit version
- Serial number
- Lot number

- Hardware version
- Software version

The software version is displayed only for Units that contain software.

Unit Versions 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-8 Version Information* on page A-66 for the functions that are supported by each unit version.

Unit Version Notation

In this User's Manual, unit versions are specified as shown in the following table.

Unit version in Unit specifications on the product	Notation in this manual	Remarks
Unit version 1.0 or later	Ver. 1.□ or later	Unless unit versions are specified, the information in this manual applies to all unit versions.

Related Manuals

The following manuals are related. Use these manuals for reference.

Manual name	Cat. No.	Model numbers	Application	Description
NX-series Safety Control Unit User's Manual	Z930	NX-SL□□□□ NX-SI□□□□ NX-SO□□□□	Learning how to use NX-series Safety Control Units.	The hardware, setup methods, and functions of the NX-series Safety Control Unit are described.
NX-series Safety Control Unit Instructions Reference Manual	Z931	NX-SL□□□□	Learning about the specifications of instructions for the Safety CPU Unit.	The instructions for the Safety CPU Unit are described. When programming, use this manual together with the *NX-series Safety Control Unit User's Manual* (Cat. No. Z930).
NX-series EtherCAT® Coupler Unit User's Manual	W519	NX-ECC□□□	Leaning how to use an NX-series EtherCAT Coupler Unit and EtherCAT Slave Terminals.	The following items are described: the overall system and configuration methods of an EtherCAT Slave Terminal (which consists of an NX-series EtherCAT Coupler Unit and NX Units), and information on hardware, setup, and functions to set up, control, and monitor NX Units through EtherCAT.
NX-series 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.
NJ-series CPU Unit Hardware User's Manual	W500	NJ501-□□□□ NJ301-□□□□	Learning the basic specifications of the NJ-series CPU Units, including introductory information, designing, installation, and maintenance. Mainly hardware information is provided.	An introduction to the entire NJ-series system is provided along with the following information on the CPU Unit. • Features and system configuration • Overview • Part names and functions • General specifications • Installation and wiring • Maintenance and Inspection Use this manual together with the *NJ-series CPU Unit Software User's Manual* (Cat. No. W501).
NJ-series CPU Unit Software User's Manual	W501	NJ501-□□□□ NJ301-□□□□	Learning how to program and set up an NJ-series CPU Unit. Mainly software information is provided.	The following information is provided on an NJ-series CPU Unit. • CPU Unit operation • CPU Unit features • Initial settings • Programming based on IEC 61131-3 language specifications Use this manual together with the *NJ-series CPU Unit Hardware User's Manual* (Cat. No. W500).
NJ-series CPU Unit Built-in EtherCAT® Port User's Manual	W505	NJ501-□□□□ NJ301-□□□□	Using the built-in EtherCAT port on an NJ-series CPU Unit.	Information on the built-in EtherCAT port is provided. This manual provides an introduction and provides information on the configuration, features, and setup. Use this manual together with the *NJ-series CPU Unit Hardware User's Manual* (Cat. No. W500) and *NJ-series CPU Unit Software User's Manual* (Cat. No. W501).
NJ-series Instructions Reference Manual	W502	NJ501-□□□□ NJ301-□□□□	Learning detailed specifications on the basic instructions of an NJ-series CPU Unit.	The instructions in the instruction set (IEC 61131-3 specifications) are described. When programming, use this manual together with the *NJ-series CPU Unit Hardware User's Manual* (Cat. No. W500) and *NJ-series CPU Unit Software User's Manual* (Cat. No. W501).
NJ-series Troubleshooting Manual	W503	NJ501-□□□□ NJ301-□□□□	Learning about the errors that may be detected in an NJ-series Controller.	Concepts on managing errors that may be detected in an NJ-series Controller and information on individual errors are described. Use this manual together with the *NJ-series CPU Unit Hardware User's Manual* (Cat. No. W500) and *NJ-series CPU Unit Software User's Manual* (Cat. No. W501).
Sysmac Studio Version 1 Operation Manual	W504	SYSMAC-SE2□□□	Learning about the operating procedures and functions of the Sysmac Studio.	Describes the operating procedures of the Sysmac Studio.

Terminology

Unless specified otherwise, this manual uses the following terms to describe the NX-series Safety Control Unit.

Term	Description
after safety validation	This 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.
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.
change tracking	A pin is used to manage whether the safety application data has been changed after the finalized data is created.
configuration information	It consists of the following data: • Unit configuration information • I/O allocation information
DEBUG mode	The mode that is used to debug unvalidated safety programs. DEBUG mode is only available when the Sysmac Studio is online with Safety CPU Unit. 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. 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.
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.
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.
dual channels	Two inputs or outputs are used as a pair of points for redundancy.
EtherCAT Slave Terminal	An EtherCAT Slave Terminal is a building-block slave that is created by mounting a group of NX Units to an EtherCAT Coupler Unit.
EtherCAT Slave Terminal setting	It consists of the following data: • Configuration information • Unit operation settings • Unit application data
exposing global variables to the NJ-series CPU Unit	Exposing specified global variables to the Safety CPU Unit to allow the exchange of standard signals between the NJ-series CPU Unit and the Safety CPU Unit.
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. 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. 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. You use the FBD language to write safety programs for the Safety CPU Unit.
I/O allocation information	The set of information that specifies the I/O data to be processed by I/O refreshing. On the Sysmac Studio, this is shown as configuration information and includes the Unit configuration information.

Term	Description
NJ-series CPU Unit	An NJ-series CPU Unit for general control purposes. This differs from the Safety CPU Unit that is used for safety controls.
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. There are the three modes: PROGRAM mode, DEBUG mode, and RUN mode. You can use DEBUG mode only when the Sysmac Studio is online with the Safety CPU Unit.
PROGRAM mode	A mode that indicates that execution of the safety programs 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.
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 application data	The data that contains the settings that are used to operate the NX-series Safety Control Units. 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. On the Sysmac Studio, this data is shown as the slave parameters. 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 control	A type of control that uses devices, functions, and data that are designed with special safety measures.
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 data type	The data type for a safety signal.
safety function	A function that is executed by the safety control system to achieve a safe state for a machine hazard.
Safety I/O Unit	An I/O Unit that is used for safety controls. This is a type of NX Unit.
safety input device	An input device that is designed with special safety measures for use in safety controls. The generic term for safety input devices, such as emergency stop pushbutton switches and safety door switches.
safety input function	A function that evaluates whether the signals that are input on a safety input terminal are normal or abnormal. Specific safety evaluation functions include test pulse evaluation and dual channel evaluation. When the evaluation result shows an abnormality, the safety input data is made inactive (OFF).
safety output device	An output device that is designed with special safety measures for use in safety controls. The generic term for safety output devices, such as safety relays.
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. Specific safety evaluation functions include test pulse evaluation and dual channel evaluation. When the evaluation result shows an abnormality, the output signal on the safety output terminal is turned OFF.
safety process data communications	A type of EtherCAT PDO communications that is used for safety control purposes.
safety program	User programming for safety controls in the Safety CPU Unit. This term is used to differentiate from the user program in the NJ-series CPU Unit. Safety programs are programmed in the FBD language.

Term	Description
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. 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 signal	A signal or data where the risk of danger to humans has been reduced to an acceptable level. 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.
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. 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.
single channel	The input or output is used as a single point.
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.
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.
standard data type	The data type for a standard signal.
standard process data communications	A type of EtherCAT PDO communications that is used for standard controls.
standard signal	A signal or data that is used for general control purposes.
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.
Unit configuration information	The set of information that specifies the configuration of the NX Units that are connected to the Communications Coupler Unit. On the Sysmac Studio, this is shown as configuration information and includes the I/O allocation information.
user program	All of the programs that are created by the user. User program refers to the programs for standard controls in the NJ-series CPU Unit and to the safety programs.

Revision History

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

Revision code	Date	Revised content
01	June 2013	Original production

Sections in this Manual

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

1

Overview

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

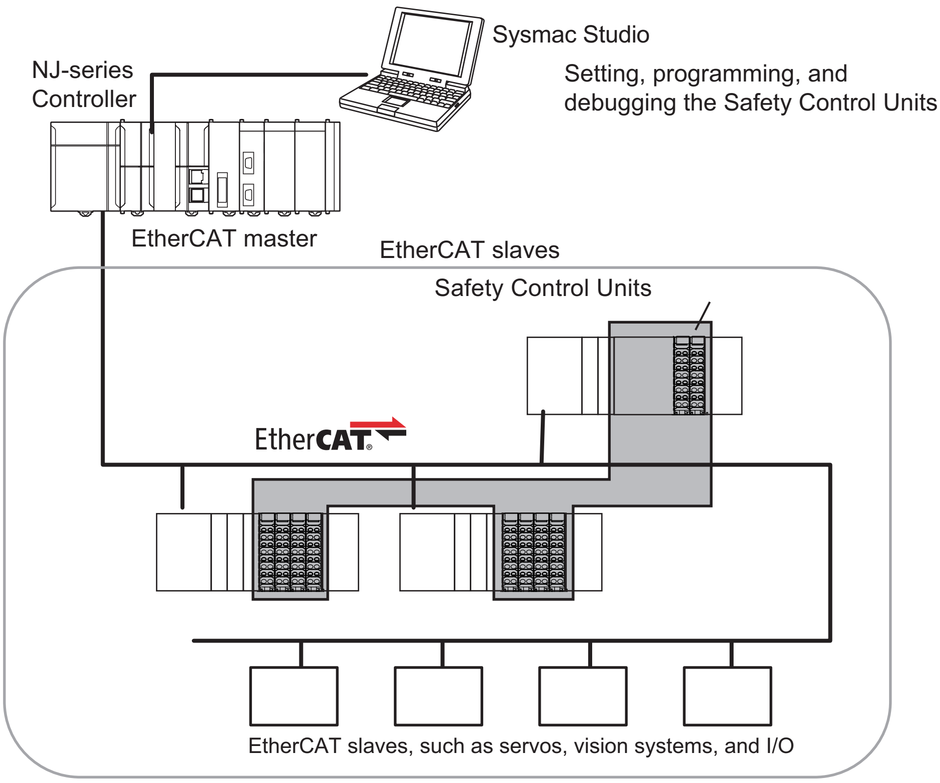
1-1 Introduction and Features

1-1-1 Overview of Safety Control Units

The NX-series Safety Control Units are part of the lineup of Sysmac devices. They are used to execute safety controls. These Units achieve safety controls when they are used in a sequence and motion control system that is based on the NJ-series Machine Automation Controller.

A safety control system that uses NX-series Safety Control Units uses an EtherCAT network.
Safety Control Units are classified as NX Units and perform remote I/O communications with the EtherCAT master through an EtherCAT Coupler Unit. These Units also support the FSoE (Safety over EtherCAT) protocol to perform safety I/O communications between the FSoE master and slaves.This allows you to combine safety process data communications and standard process data communications on the same EtherCAT network.

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 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.

1-1-2 Features of Safety Control Units

Achieving Safety Control Systems on EtherCAT Networks

● 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 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-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.

● 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

● Programming Languages Based on the IEC 61131-3 International Standard

Programming is possible with function block diagrams (FBD), which are 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 of programs written with function block diagrams (for function blocks with missing or incorrect connections, etc.)

- 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

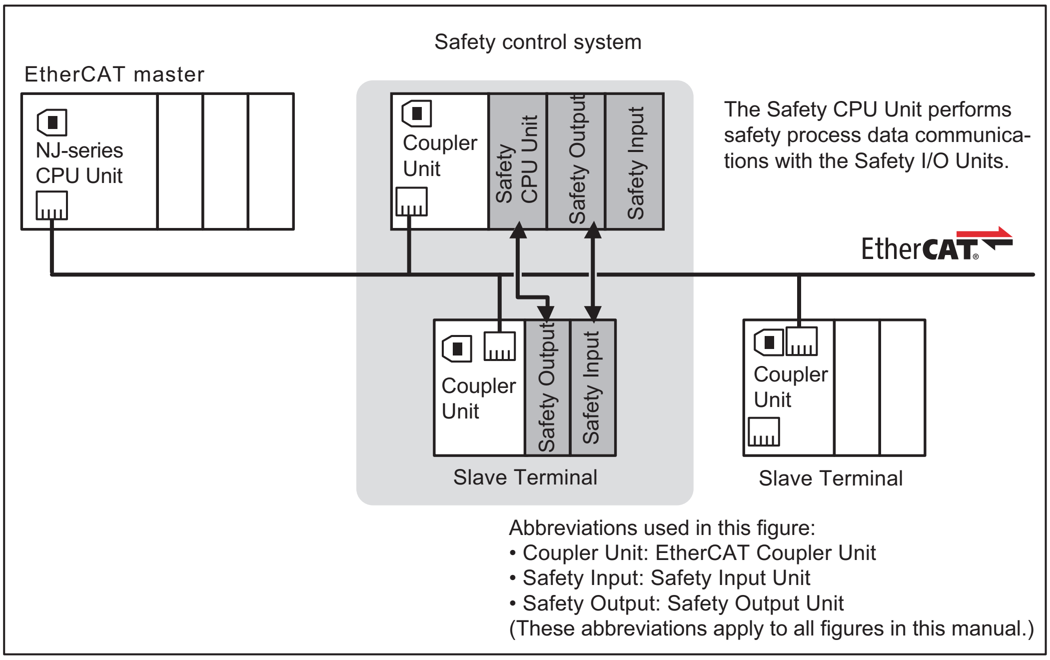
You can connect the Sysmac Studio to perform various types of debugging, including monitoring, changing present values, and forced refreshing.

1-2 System Configuration and Configuration Devices

1-2-1 Safety Control System Configuration

The safety control system operates on the EtherCAT network.
You combine the EtherCAT Master Function Module in the NJ-series CPU Unit with the EtherCAT slave functionality in an NX-series EtherCAT Coupler Unit to build a safety control system.

NJ-series EtherCAT Communications System

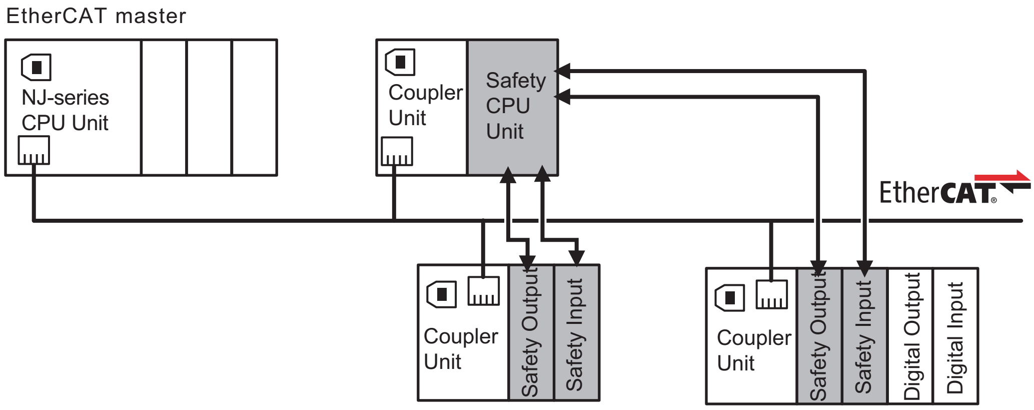


The Safety CPU Unit serves as the FSoE master to control Safety I/O Units that serve as FSoE slaves.[*1]
As shown in the above figure, the Safety CPU Unit can be used to control not only the Safety I/O Units that are mounted to the same EtherCAT Slave Terminal, but also it can control Safety I/O Units 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 has been confirmed between OMRON NX-series Safety Control Units. Preparations for the Safety over EtherCAT Test Center were underway by the EtherCAT Technology Group as of June 2013.

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 *NJ-series CPU Unit Hardware User's Manual* (Cat. No. W500) for details on the configuration of an EtherCAT network.

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

Refer to *A-8 Version Information* on page A-66 in the Appendix for the model numbers and unit versions of the NJ-series CPU Units and the NX-series EtherCAT Coupler Units that can be used together.

1-2-2 Types of Safety Control Units

The following table lists the NX-series Safety Control Units that are available.

Unit type		Outline
Safety CPU Unit		This Unit has safety control functions. It operates as an NX Unit. It also operates as an FSoE master.
Safety I/O Units		These Units have safety input functions or safety output functions. They operate as NX Units. These Units operate as FSoE slaves.
	Safety Input Units	These Units have safety input functions.
	Safety Output Units	These Units have safety output functions.

Refer to *Section 2 Specifications* 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 Units, 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 Units, and to perform programming and debugging. You can use the following Support Software.

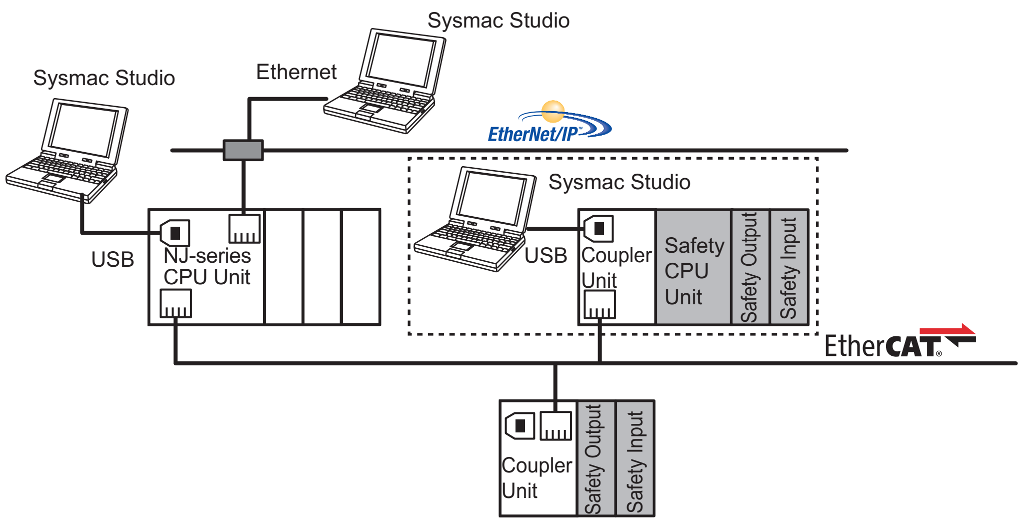
Support Software	Version
Sysmac Studio	Version 1.07 or higher

Refer to the *NJ-Series Sysmac Studio Operation Manual* (Cat. No. W504) for the system requirements of the Sysmac Studio.

1-3-2 Connection Method and Procedures

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

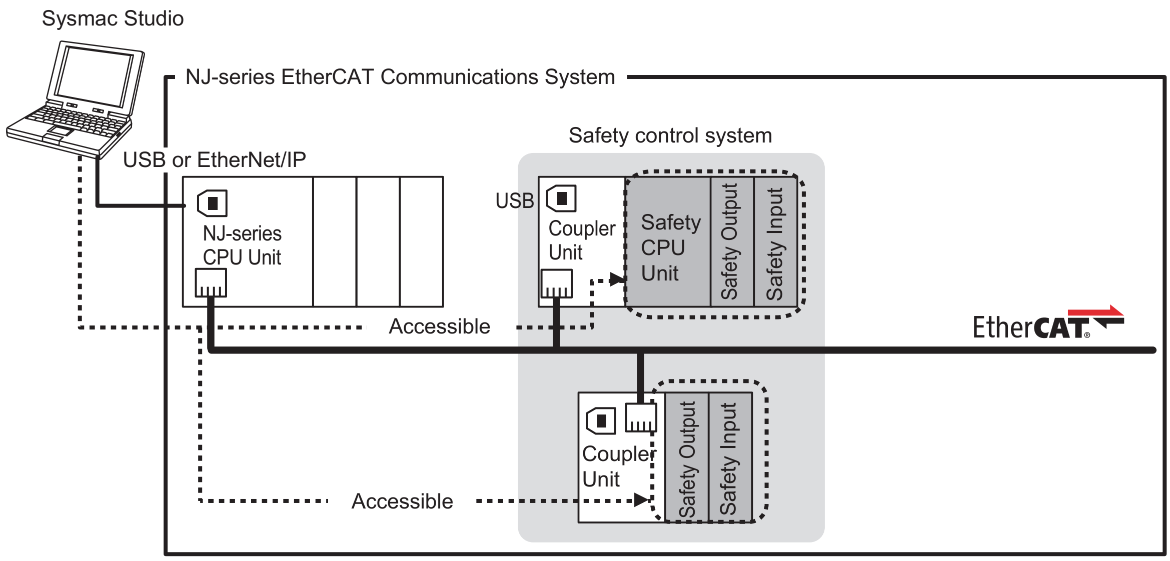
- USB Connection or Ethernet Connection to the NJ-series CPU Unit
  The USB connection and Ethernet connection are functionally identical. This is the most common connection method.
- USB Connection to the NX-series EtherCAT Coupler Unit
  This method is functionally limited when compared with a connection to the NJ-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.

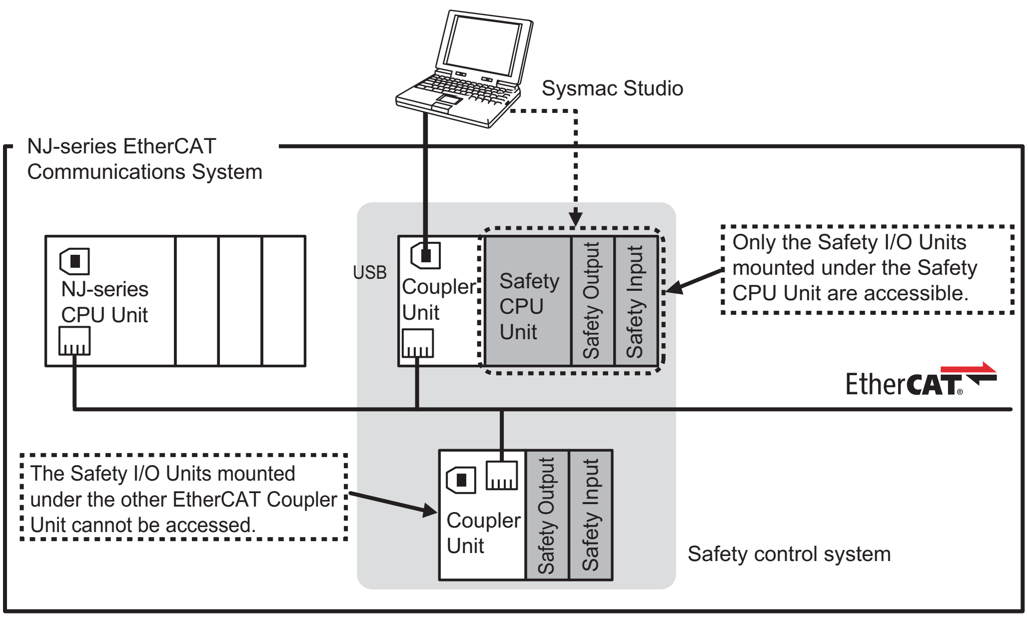
USB Connection or Ethernet Connection to the NJ-series CPU Unit

You use the Sysmac Studio to connect to the USB port or the built-in EtherNet/IP port of the NJ-series CPU Unit.
This connection allows you to download safety programs to the Safety CPU Unit and to monitor the Safety CPU Unit 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 safety programs to 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-series CPU Unit. We therefore recommend that you connect to the NJ-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.

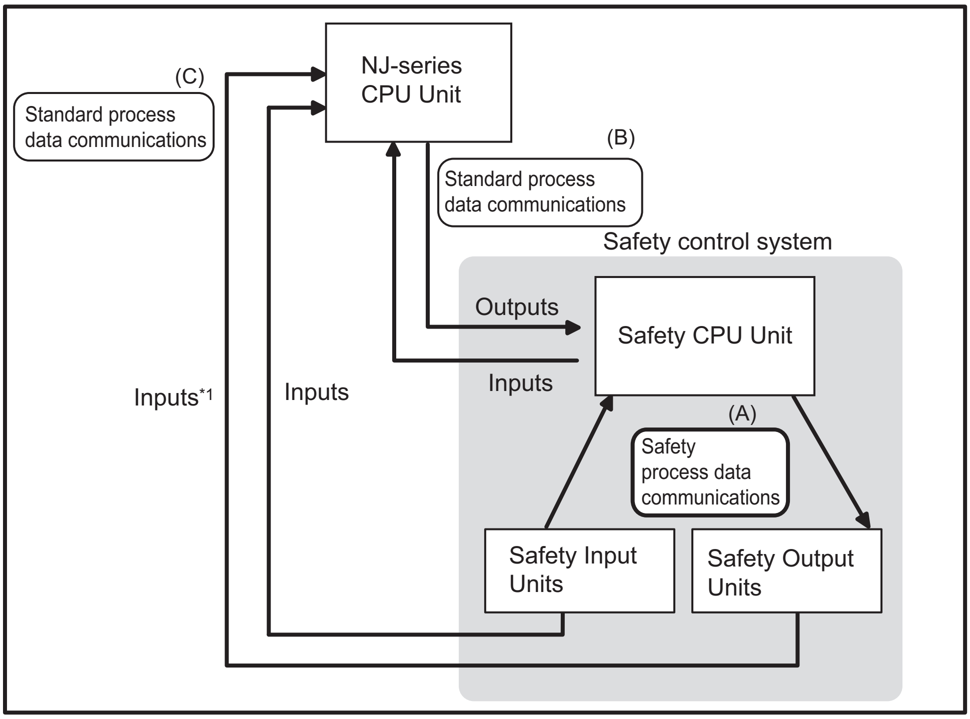
1-4 Exchanging Signals between Units

The safety control system uses EtherCAT communications to exchange signals between Units.
This section describes how signals are exchanged between the Units in the safety control system.

1-4-1 Relationship between Units and Types of Communications

The following figure applies to communications systems where the built-in EtherCAT port on an NJ-series CPU Unit is used as the master. The figure describes the relationship of the Safety CPU Unit, Safety I/O Units, and the NJ-series CPU Unit within the communications system, and provides details on communications between the Units.

Communications System Where the Built-in EtherCAT Port on an NJ-series CPU Unit Is the Master



*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 an NJ-series CPU Unit through standard process data communications. (Section (B) in the above figure.)

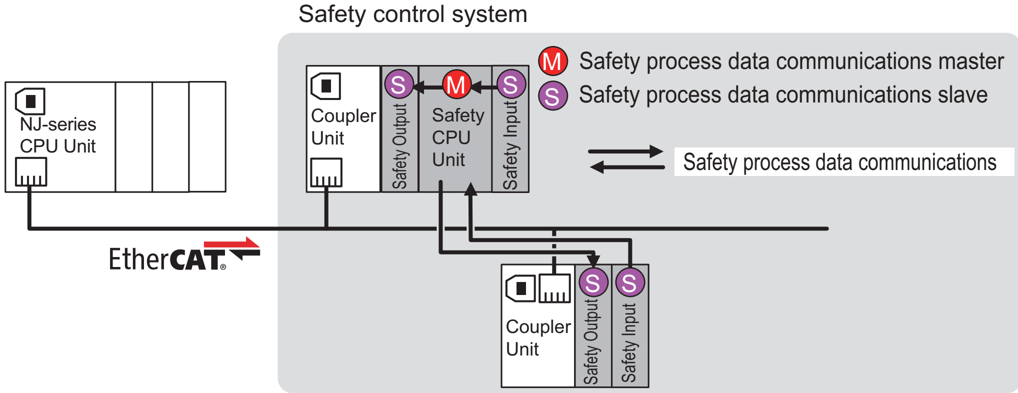
The NJ-series CPU Unit can also perform standard process data control with Safety I/O Units, but only for inputs. (Section (C) in the above figure.)

The following table summarizes the relationship described above.

Units		Communications type	Typical application
Master	Slaves		
Safety CPU Unit	Safety I/O Units	Safety process data communications (Section (A) in the above figure)	Safety control signals from the Safety CPU Unit.
NJ-series CPU Unit	Safety CPU Unit	Standard process data communications (Section (B) in the above figure.)	Monitoring and operation preparation commands for the Safety CPU Unit and Safety I/O Units from the NJ-series CPU Unit
	Safety I/O Units	Standard process data communications (Section (C) in the above figure.)	The NJ-series CPU Unit receives the status information from the Safety I/O Units.

Relationship between Master and Slaves during Safety Process Data Communications

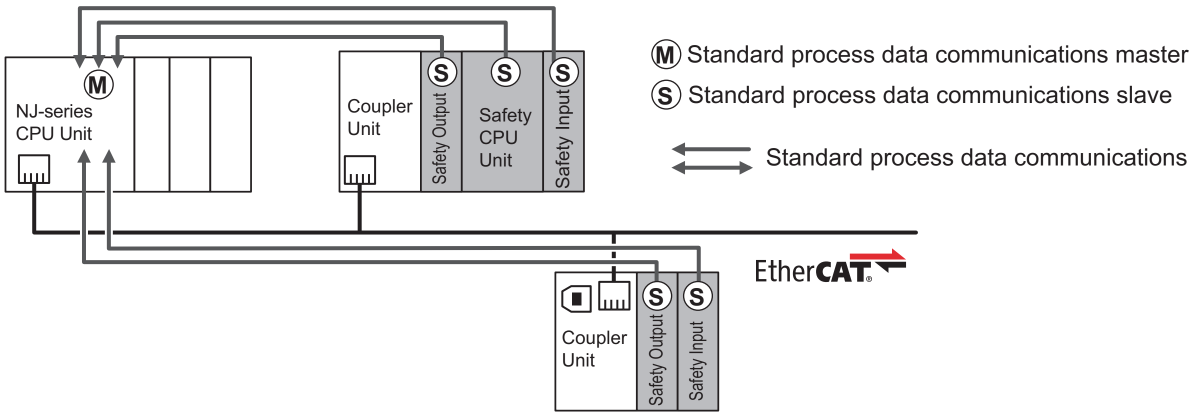
The following figure shows the EtherCAT master/slave relationship during safety process data communications.



The Safety CPU Unit performs safety process data communications with the Safety I/O Units on the same Slave Terminal, or the Safety I/O Units on another Slave Terminal on the EtherCAT network.
In this relationship, 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.

Relationship between Master and Slaves during Standard Process Data Communications

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



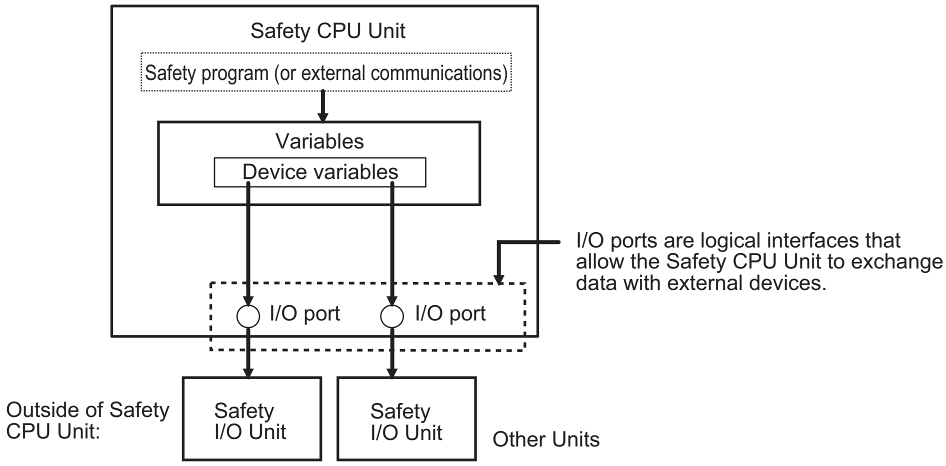
The NJ-series CPU Unit performs standard process data communications with the Safety CPU Unit and the Safety I/O Units on the EtherCAT network.
In this relationship, the NJ-series CPU Unit operates as the master in the standard process data communications. Meanwhile, the Safety CPU Unit and Safety I/O Units operate as slaves in the standard process data communications. The NJ-series CPU Unit can only receive inputs from Safety I/O Units, even if the Unit is a Safety Output Unit.

1-4-2 I/O Processing with Safety I/O Units and other External Devices

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

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.

Types of Signals and Relationship between 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, and standard controls.

- The exchange of signals with safety inputs and safety outputs is done with safety process data communications.
- The exchange of standard signals in the standard control system is done with standard process data communications.

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 *7-2-5 Data Types* on page 7-14 for details on the safety data types.

● Standard Data Type Variables

These variables represent data that is not related to safety controls.
Refer to *6-7 Sharing Variable Data with the NJ-series CPU Unit* on page 6-17 for details on how to access standard data type variables from the NJ-series CPU Unit.

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	
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

Specifying Safety Data Types and Standard Data Types

The following table shows how safety data type variables and standard data type variables are used based on the type of communications.

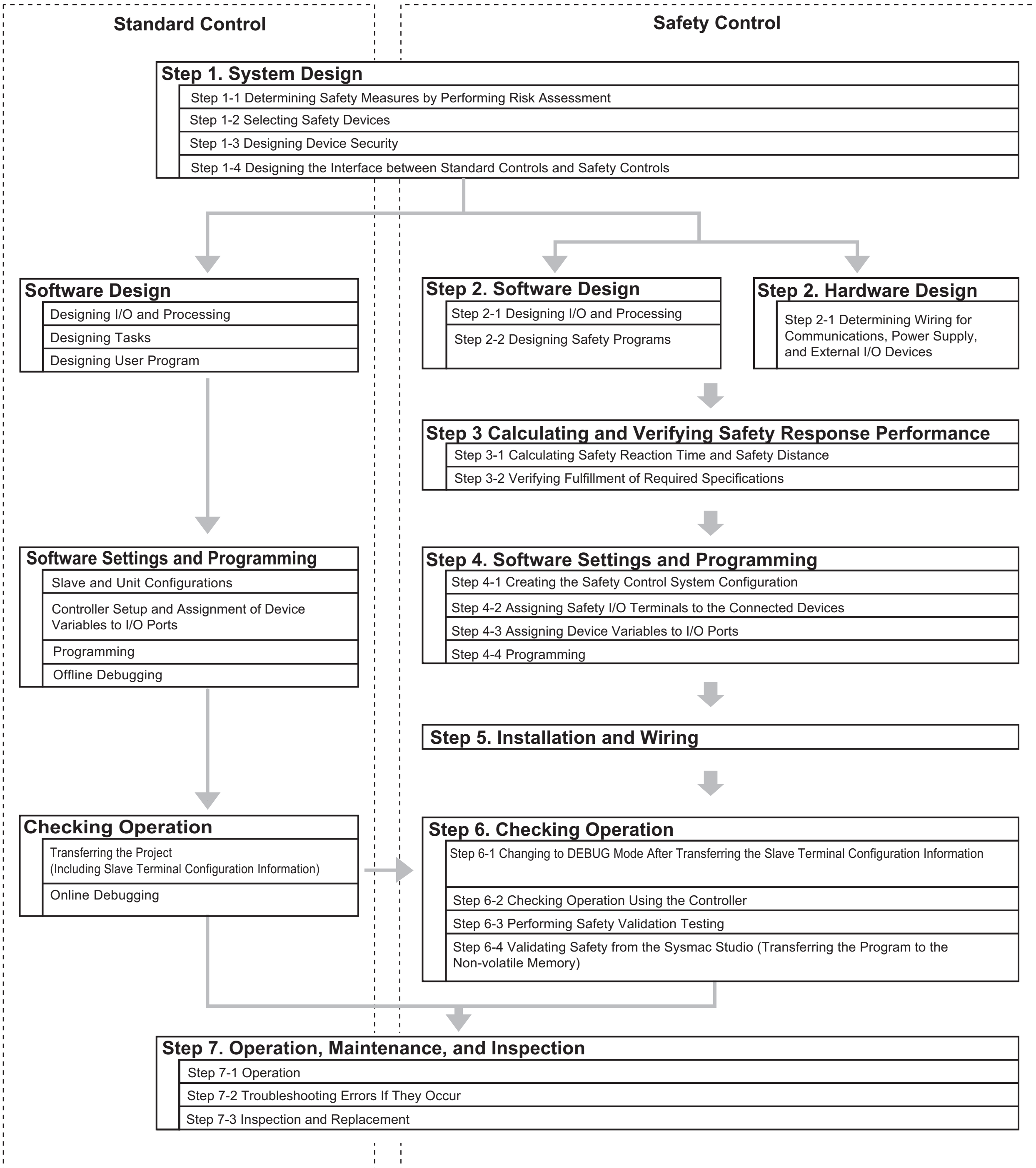
Units		Communications type	Data type of variable to use
Master	Slaves		
Safety CPU Unit	Safety I/O Units	Safety process data communications	Safety data type variables
NJ-series CPU Unit	Safety CPU Unit	Standard process data communications	Standard data type variables
	Safety I/O Units		

- 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.

1-5 Commissioning Procedures

1-5-1 Overall Procedure

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



1-5-2 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-series CPU Unit Software User's Manual* (Cat. No. W501) for the detailed procedures 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. • 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 1-5 *Section 2 Specifications* *Section 3 Part Names and Functions*
Step 1-3 Designing the Interface between Standard Controls and Safety Controls	Design the interface between the standard controls and safety controls. This is done by exposing the global variables to the NJ-series CPU Unit.	*6-7 Sharing Variable Data with the NJ-series CPU Unit* on page 6-17

Step 2. Software Design

Step	Description	Reference
Step 2-1 Designing I/O and Processing	Design the configuration of the external I/O and Safety I/O Units. • Refresh periods for external I/O devices • Program contents	*Section 3 Part Names and Functions* *Section 4 Calculating Safety Reaction Times*
Step 2-2 Designing Safety Programs	Design the POUs (Program Organization Units). • Programs • Function blocks Design of Variables: • Design the data types of the variables (particularly the design of safety data types and standard data types). • Define the variables that you will use in more than one POU and variables that you will use in only specific POUs. • Define the variable names for the device variables that you use to access Safety I/O Units. • Define the attributes of variables, such as the Name attribute. • Design the variables to expose to the user program for the standard controls. • Design the interface with the safety controls of the user program for the standard controls.	*Section 7 Programming*

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 EtherCAT communications, power supply, and external I/O devices.	*Section 3 Part Names and Functions* *Section 7 Wiring* in the *NX-series EtherCAT Coupler Unit User's Manual* (Cat. No. W519) NX Unit User's Manuals

Step 3. Calculating and Verification of Safety Reaction Times

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 4 Calculating Safety Reaction Times*
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.	

Step 4. Software Settings and Programming

Step	Description	Reference
Step 4-1 Creating the Safety Control System Configuration	On the Sysmac Studio, configure the EtherCAT Coupler Units, Safety CPU Units, and Safety I/O Units in the EtherCAT network configuration.	*6-3 Configuration and Setup of the EtherCAT Network and EtherCAT Slave Terminal* on page 6-4
Step 4-2 Assigning Safety I/O Terminals to the Connected Devices	On the parameter setting tab page for the Safety I/O Units, select the external I/O devices that are connected to the safety I/O terminals.	*6-5 Setting the Safety Input and Output Functions* on page 6-10
Step 4-3 Assigning Device Variables to I/O Ports	Register the device variables in the global variable table. (You can use either user-defined or automatically assigned variable names.)	*6-6 Registering Device Variables* on page 6-12
Step 4-4 Programming	Variable Registration: • Register the variables that are used by more than one POU in the global variable table with the Sysmac Studio. • Register the variables that are used in only a specific program in the local variable table for that program. • Register the variables that are used in only a specific function block in the local variable table for that function block. Writing Algorithms for POUs: Write the algorithms for the POUs (programs and function blocks) using function block diagrams (FBD).	*7-5 Programming Operations* on page 7-26

Step 5. Installation and Wiring

Step	Description	Reference
Step 5-1 Installation	• Connect the Units to each other. • Mount the connected Units on a DIN Track.	*Section 5 Installation and Wiring*
Step 5-2 Connecting the Ethernet Cables	• Connect the Ethernet cable to the built-in EtherCAT port on the NJ-series CPU Unit. • Connect the Ethernet cable to the EtherCAT port on the EtherCAT Coupler Unit.	*Section 7 Wiring* in the *NX-series EtherCAT Coupler Unit User's Manual* (Cat. No. W519)
Step 5-3 Wiring the I/O	• Wire the Safety I/O Units.	*Section 5 Installation and Wiring*
Step 5-4 Connecting the Computer That Runs the Sysmac Studio	Use one of the following connections. • Connect a USB cable to the NJ-series CPU Unit Or • Connect an Ethernet cable to the built-in EtherNet/IP port on the NJ-series CPU Unit.	*1-3 Support Software* on page 1-7 *Sysmac Studio Version 1 Operation Manual* (Cat. No. W504) *Section 7 Wiring* in the *NX-series EtherCAT Coupler Unit User's Manual* (Cat. No. W519)

Step 6. Checking Operation

Step	Description	Reference
Step 6-1 Changing to DEBUG Mode After Transferring the Slave Terminal Configuration Information	From the NJ-series CPU Unit Setup and Programming View, go online with the NJ-series CPU Unit (or the EtherCAT Coupler Unit), transfer the EtherCAT network configuration information, and then transfer the Slave Terminal configuration information. Then, change the Safety CPU Unit to DEBUG mode from the Safety CPU Unit Setup and Programming View. This transfers the safety application data to the Safety CPU Unit and enables debugging.	*8-2 Transferring the Configuration Information* on page 8-6 *8-3 Operating Modes of the Safety CPU Unit* on page 8-9 *8-4 Changing to DEBUG Mode* on page 8-15
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.	*8-5 Functions for Checking Operation* on page 8-18
Step 6-3 Performing Safety Validation Testing	Test all safety functions to see if they operate according to designs.	*8-5 Functions for Checking Operation* on page 8-18
Step 6-4 Designing Device Security	Design the safety passwords.	*8-7 Setting the Safety Password* on page 8-33
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. This transfers the safety application data to the non-volatile memory in the Safety CPU Unit and enables operation.	*8-8 Performing Safety Validation and Operation* on page 8-34

Step 7. Operation, Maintenance, and Inspection

Step	Description	Reference
Step 7-1 Operation	Restart the Safety CPU Unit. If the Safety CPU Unit has a validated user program, the Safety CPU Unit will automatically start in RUN mode.	*8-8 Performing Safety Validation and Operation* on page 8-34
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 9 Troubleshooting*
Step 7-3 Inspection and Replacement	Perform periodic maintenance. If you find any defects or problems during the inspection, replace the affected devices.	*Section 10 Maintenance and Inspection*

2

Specifications

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

2-1 General Specifications

The general specifications of the NX-series Safety CPU Unit and Safety I/O Units are given in the following table.

Item		Specification
Enclosure		Mounted in a panel (open)
Grounding method		Ground to 100 Ω or less.
Operating environment	Ambient operating temperature	0 to 55°C (The upper limit of the ambient operating temperature is restricted by the installation orientation.)
	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: Conforms to JIS B3502 and IEC 61131-2.
	Noise immunity	Conforms to IEC 61131-2. 2 kV on power supply line (Conforms to IEC 61000-4-4.)
	Insulation class	Class III (SELV)
	Overvoltage category	Category II: Conforms to JIS B3502 and IEC 61131-2.
	EMC immunity level	Zone B
	Vibration resistance	Conforms to IEC 60068-2-6. 5 to 8.4 Hz with 3.5-mm amplitude, 8.4 to 150 Hz, acceleration of 9.8 m/s^2, 100 minutes each in X, Y, and Z directions (10 sweeps of 10 min each = 100 min total)
	Shock resistance	Conforms to IEC 60068-2-27. 147 m/s^2, 3 times each in X, Y, and Z directions
	Insulation resistance	20 MΩ between isolated circuits (at 100 VDC)
	Dielectric strength	510 VAC for 1 min between isolated circuits, leakage current: 5 mA max.
Installation method		DIN Track (IEC 60715 TH35-7.5/TH35-15)
Applicable standards		IEC 61508: 2010 SIL 3, EN 62061: 2005 SIL CL3 EN ISO 13849-1, 13849-2: 2008 PL e/Safety Category 4 UL 1998 cULus: Listed UL508, ANSI/ISA 12.12.01 EN 61131-2, C-Tick, KC: KC Registration

2-2 Specifications of Individual Units

This section gives the specifications of the NX-series Safety CPU Unit and the Safety I/O Units.

2-2-1 Models

Safety CPU Unit

Model	Maximum number of safety I/O points	Program capacity	Number of safety master connections	I/O refreshing method
NX-SL3300	256 points	512 KB	32	Free-Run refreshing only

Safety Input Units

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 only
NX-SID800	8 points	2 points	Sinking inputs (PNP)	24 VDC	Cannot be connected.	1	Free-Run refreshing only

Safety Output Units

Model	Number of safety input 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, 4.0 A/Unit at 40°C, and 2.5 A/Unit at 55°C	24 VDC	1	Free-Run refreshing only
NX-SOD400	4 points	Sourcing outputs (PNP)	0.5 A/point and 2.0 A/Unit	24 VDC	1	Free-Run refreshing only

2-2-2 Safety CPU Unit

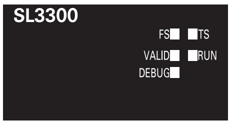
Datasheet Items for the Safety CPU Unit

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

Item	Description
Unit name	This is the name of the Unit.
Model	This is the model number of the Unit.
Maximum number of safety I/O points	This is the number of safety I/O points that the Unit can control.
Program capacity	This is the capacity of the user program in the Unit.
Number of safety master connections	This is the number of safety master connections that the Unit can have through FSoE. You can connect one Safety I/O Unit for each safety master connection.
I/O refreshing method	This is the I/O refreshing method for the Unit. There are two methods: Free-Run refreshing and synchronous I/O refreshing.
External connection terminals	This is the type of terminal block or connector that is used to wire 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.
Dimensions	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	This is the method that is used to supply I/O power to the Unit. Each Unit has its own method for supplying power. The power can be supplied either from the NX bus or from an external source.
Current capacity of I/O power supply terminals	This is the current capacity of the I/O power supply terminals (IOV and 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.
Installation orientation and restrictions	This is the installation orientation of the Slave Terminal that includes this Unit. If the installation orientation imposes any restrictions on the specifications, those restrictions are also described.

Safety CPU Unit

● NX-SL3300

Unit name	Safety CPU Unit
Model	NX-SL3300
Maximum number of safety I/O points	256 points
Program capacity	512 KB
Number of safety master connections	32
I/O refreshing method	Free-Run refreshing
External connection terminals	None
Indicators	FS indicator, VALID indicator, DEBUG indicator, TS indicator, and RUN indicator 
Dimensions	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	0.90 W max.
Current consumption from I/O power supply	No consumption
Weight	75 g max.
Installation orientation and restrictions	Installation orientation: 6 possible orientations Restrictions: None

2-2-3 Safety Input Units

Datasheet Items for Safety Input Unit

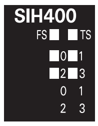
The following table gives the meaning of the datasheet items for the Safety Input Units.

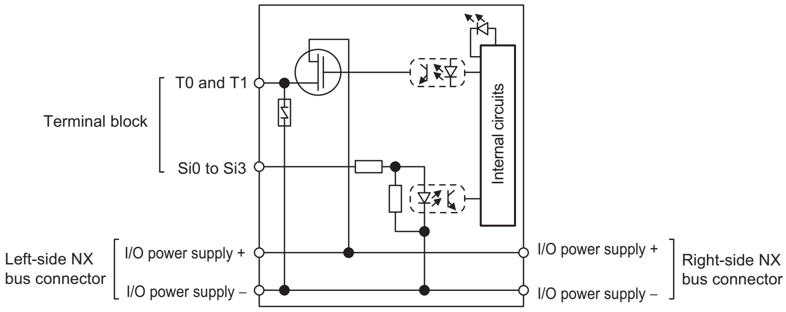
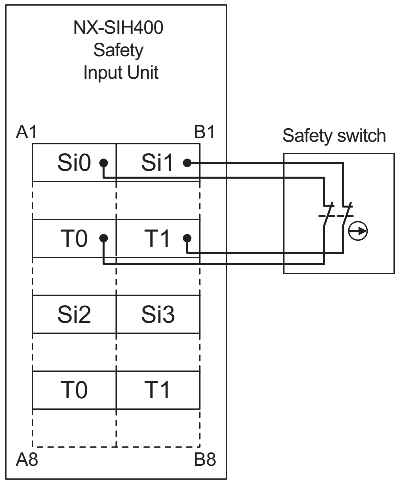
Item	Specification
Unit name	This is the name of the Unit.
Model	This is the model number of the Unit.
Number of safety input points	This is the number of safety input points on the Unit.
Number of test output points	This is the number of test output points on the Unit. The test output points are used with the safety input terminals.
Internal I/O common	This is the polarity that the Unit uses to connect to input devices. There are models with NPN and PNP connections.
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 Non-contact Door Switches, E3FS Single Beam Safety Sensors, etc.).
Number of safety slave connections	This is the number of safety slave connections that the Unit can have through FSoE. You can connect to one Safety CPU Unit for each safety slave connection.
I/O refreshing method	This is the I/O refreshing method for the Unit. Only Free-Run refreshing is supported.
External connection terminals	This is the type of terminal block or connector that is used to wire 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.
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.
Internal I/O common	This is the polarity that the Unit uses to connect to devices. The Unit uses PNP connections.
Test output load current	This is the maximum load current for each test output on the Unit.
Test output 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	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. Each Unit has its own method for supplying power. The power can be supplied either from the NX bus or from an external source.
Current capacity of I/O power supply terminals	This is the current capacity of the I/O power supply terminals (IOV and 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.

Item	Specification
Weight	This is the weight of the Unit.
Circuit layout	This is the internal circuits of the Unit.
Installation orientation and restrictions	This is the installation orientation of the Slave Terminal that includes this 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.

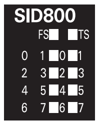
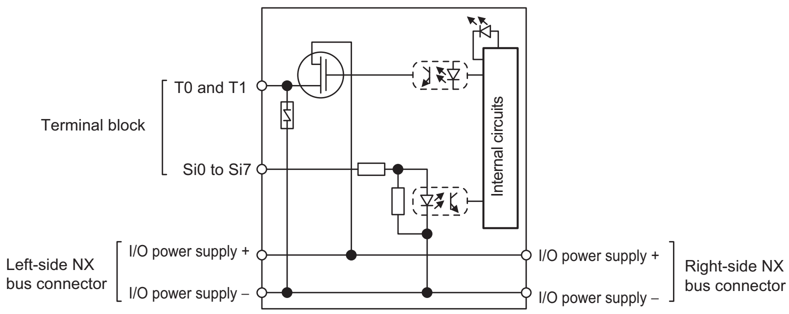
Safety Input Units

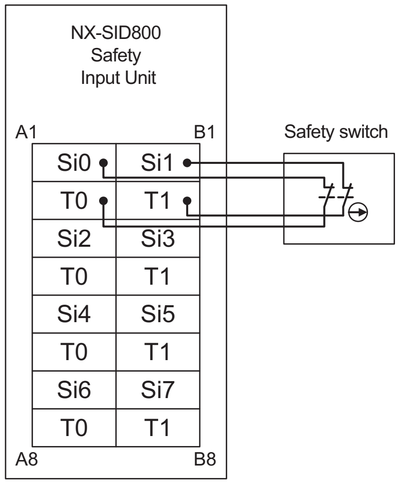
● NX-SIH400

Unit name	Safety Input Unit
Model	NX-SIH400
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 only
External connection terminals	Screwless clamping terminal block (8 terminals)
Indicators	TS indicator, FS indicator, input indicators (yellow), and input error indicators (red) 
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. (Between IOV and all output terminals)
Test output leakage current	0.1 mA max.
Dimensions	12 × 100 × 71 mm (W × H × D)
Isolation method	Photocoupler isolation
Insulation resistance	20 MΩ min. between isolated circuits (at 100 VDC)
Dielectric strength	510 VAC for 1 min between isolated circuits, leakage current: 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	0.70 W max.
Current consumption from I/O power supply	20 mA max.
Weight	70 g max.

Circuit layout	
Terminal connection diagram	Si0 to Si3: Safety input terminals T0 and T1: Test output terminals  Refer to *3-3-1 Safety Input Functions* on page 3-10 for details.
Installation orientation and restrictions	Installation orientation: 6 possible orientations Restrictions: Maximum ambient temperature is 50°C for any orientation other than upright installation.
Protective functions	Overvoltage protection circuit and short detection (test outputs)

● NX-SID800

Unit name	Safety Input Unit
Model	NX-SID800
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 only
External connection terminals	Screwless clamping terminal block (16 terminals)
Indicators	TS indicator, FS indicator, input indicators (yellow), and input error indicators (red) 
Safety input current	3.0 mA typical
Safety input ON voltage	15 VDC min.
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. (Between IOV and all output terminals)
Test output leakage current	0.1 mA max.
Dimensions	12 × 100 × 71 mm (W × H × D)
Isolation method	Photocoupler isolation
Insulation resistance	20 MΩ min. between isolated circuits (at 100 VDC)
Dielectric strength	510 VAC for 1 min between isolated circuits, leakage current: 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	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 T0 and T1: Test output terminals  Refer to *3-3-1 Safety Input Functions* on page 3-10 for details.
Installation orientation and restrictions	Installation orientation: 6 possible orientations Restrictions: Maximum ambient temperature is 50°C for any orientation other than upright installation.
Protective functions	Overvoltage protection circuit and short detection (test outputs)

2-2-4 Safety Output Units

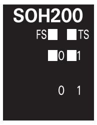
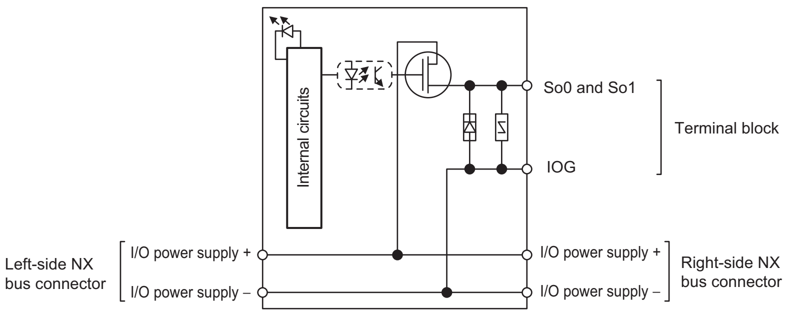
Datasheet Items for Safety Output Unit

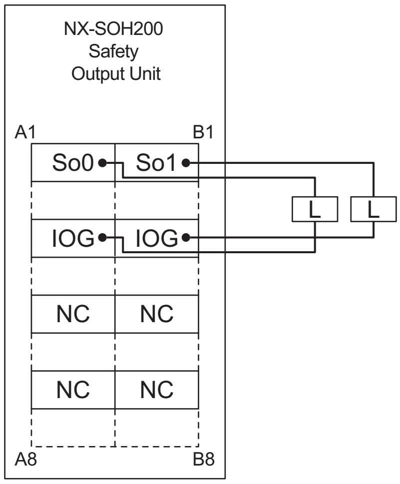
The following table gives the meaning of the datasheet items for the Safety Output Units.

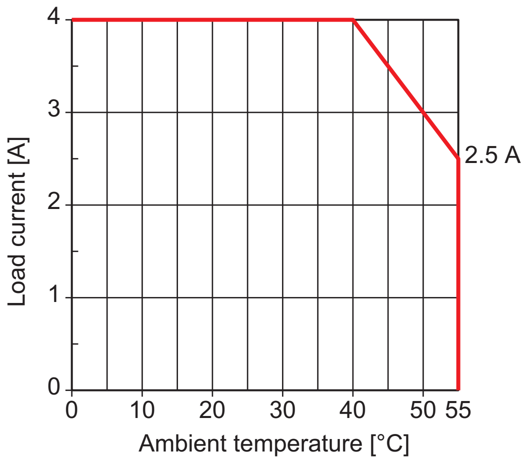
Item	Specification
Unit name	This is the name of the Unit.
Model	This is the model number of the Unit.
Number of safety output points	This is the number of safety output points on the Unit.
Internal I/O common	This is the polarity that the Unit uses to connect to input devices. There are models with NPN and PNP connections.
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 safety slave connections that the Unit can have through FSoE. You can connect to one Safety CPU Unit for each safety slave connection.
I/O refreshing method	This is the I/O refreshing method for the Unit. Only Free-Run refreshing is supported.
External connection terminals	This is the type of terminal block or connector that is used to wire 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.
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	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	This is the method that is used to supply I/O power to the Unit. Each Unit has its own method for supplying power. The power can be supplied either from the NX bus or from an external source.
Current capacity of I/O power supply terminals	This is the current capacity of the I/O power supply terminals (IOV and 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 Slave Terminal that includes this 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.

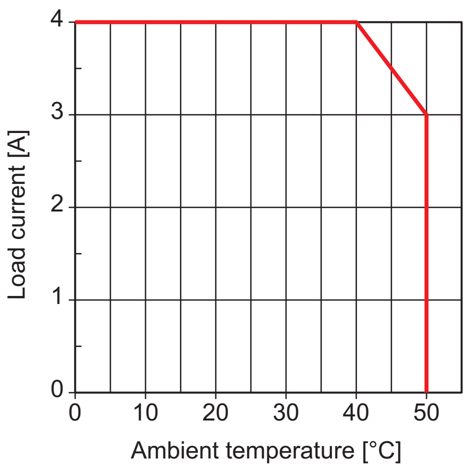
Safety Output Units

● NX-SOH200

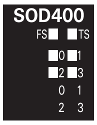
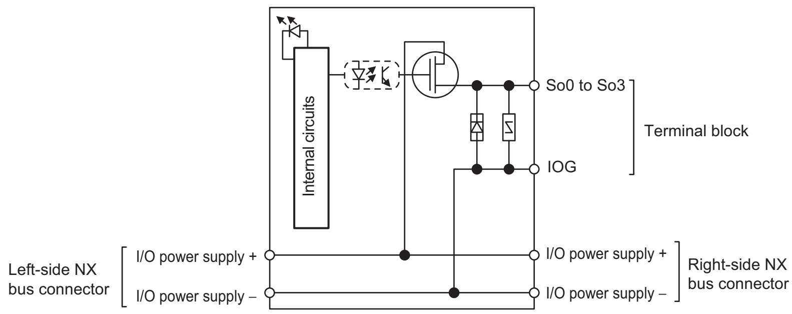
Unit name	Safety Output Unit
Model	NX-SOH200
Number of safety output points	2 points
Internal I/O common	PNP (sourcing outputs)
Maximum load current	2.0 A/point 4.0 A/Unit at 40°C 2.5 A/Unit at 55°C The maximum load current depends on the installation orientation and ambient temperature.
Rated voltage	24 VDC (20.4 to 28.8 VDC)
Number of safety slave connections	1
I/O refreshing method	Free-Run refreshing only
External connection terminals	Screwless clamping terminal block (8 terminals)
Indicators	TS indicator, FS indicator, output indicators (yellow), and output error indicators (red) 
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	12 × 100 × 71 mm (W × H × D)
Isolation method	Photocoupler isolation
Insulation resistance	20 MΩ min. between isolated circuits (at 100 VDC)
Dielectric strength	510 VAC for 1 min between isolated circuits, leakage current: 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 max./terminal
NX Unit power consumption	0.70 W max.
Current consumption from I/O power supply	40 mA max.
Weight	65 g max.
Circuit layout	

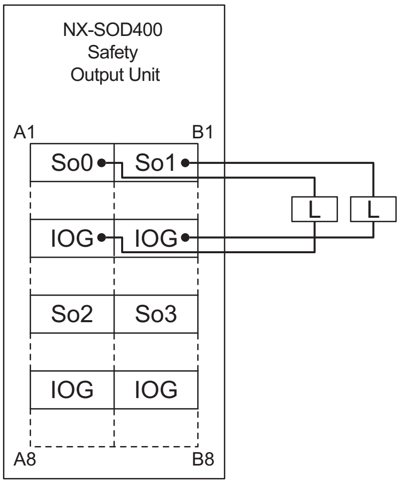
<table>
<tr><td>Terminal connection diagram</td><td>So0 and So1: Safety output terminals
IOG: I/O power supply 0 V


Refer to 3-3-2 Safety Output Functions on page 3-27 for details.</td></tr>
<tr><td>Installation orientation and restrictions</td><td>Installation orientation: 6 possible orientations
Restrictions: For upright installation, the ambient temperature is restricted as shown below according to the total Unit load current.


For all installation orientations other than upright installation, the ambient temperature is restricted as shown below according to the total Unit load current.

</td></tr>
<tr><td>Protective functions</td><td>Overvoltage protection circuit and short detection</td></tr>
</table>

● NX-SOD400

Unit name	Safety Output Unit
Model	NX-SOD400
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 voltage	24 VDC (20.4 to 28.8 VDC)
Number of safety slave connections	1
I/O refreshing method	Free-Run refreshing only
External connection terminals	Screwless clamping terminal block (8 terminals)
Indicators	TS indicator, FS indicator, output indicators (yellow), and output error indicators (red) 
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	12 × 100 × 71 mm (W × H × D)
Isolation method	Photocoupler isolation
Insulation resistance	20 MΩ min. between isolated circuits (at 100 VDC)
Dielectric strength	510 VAC for 1 min between isolated circuits, leakage current: 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 max./terminal IOG (A7 and B7): 0.5 A max./terminal
NX Unit power consumption	0.75 W max.
Current consumption from I/O power supply	60 mA max.
Weight	65 g max.
Circuit layout	

Terminal connection diagram	So0 to So3: Safety output terminals IOG: I/O power supply 0 V  Refer to *3-3-2 Safety Output Functions* on page 3-27 for details.
Installation orientation and restrictions	Installation orientation: 6 possible orientations Restrictions: None
Protective functions	Overvoltage protection circuit and short detection

2-3 PFD and PFH Values

This section gives the PFD and PFH values of the NX-series Safety CPU Unit and the Safety I/O Units.

2-3-1 Safety CPU Unit

Model	PFD proof test interval (years)							
	0.25	0.5	1	2	5	10	15	20
NX-SL3300	7.4E-08	1.5E-07	3.0E-07	5.9E-07	1.5E-06	3.0E-06	4.6E-06	6.1E-06

Model	PFH
NX-SL3300	3.1E-10

2-3-2 Safety Input Units

Model	PFD proof test interval (years)							
	0.25	0.5	1	2	5	10	15	20
NX-SID800	8.4E-08	1.7E-07	3.4E-07	6.7E-07	1.7E-06	3.4E-06	5.0E-06	6.7E-06
NX-SIH400	8.3E-08	1.7E-07	3.3E-07	6.6E-07	1.7E-06	3.3E-06	5.0E-06	6.7E-06

Model	PFH
NX-SID800	4.3E-10
NX-SIH400	3.1E-10

2-3-3 Safety Output Units

Model	PFD proof test interval (years)							
	0.25	0.5	1	2	5	10	15	20
NX-SOD400	8.4E-08	1.7E-07	3.4E-07	6.7E-07	1.7E-06	3.4E-06	5.1E-06	6.7E-06
NX-SOH200	8.3E-08	1.7E-07	3.4E-07	6.6E-07	1.7E-06	3.4E-06	5.0E-06	6.7E-06

Model	PFH
NX-SOD400	5.5E-10
NX-SOH200	3.6E-10

Additional Information

Refer to *Safety validation testing (user testing)* in the *Periodic Inspection Points* given in *10-1-2 Periodic Inspections* on page 10-2 for the proof test perspective.

3

Part Names and Functions

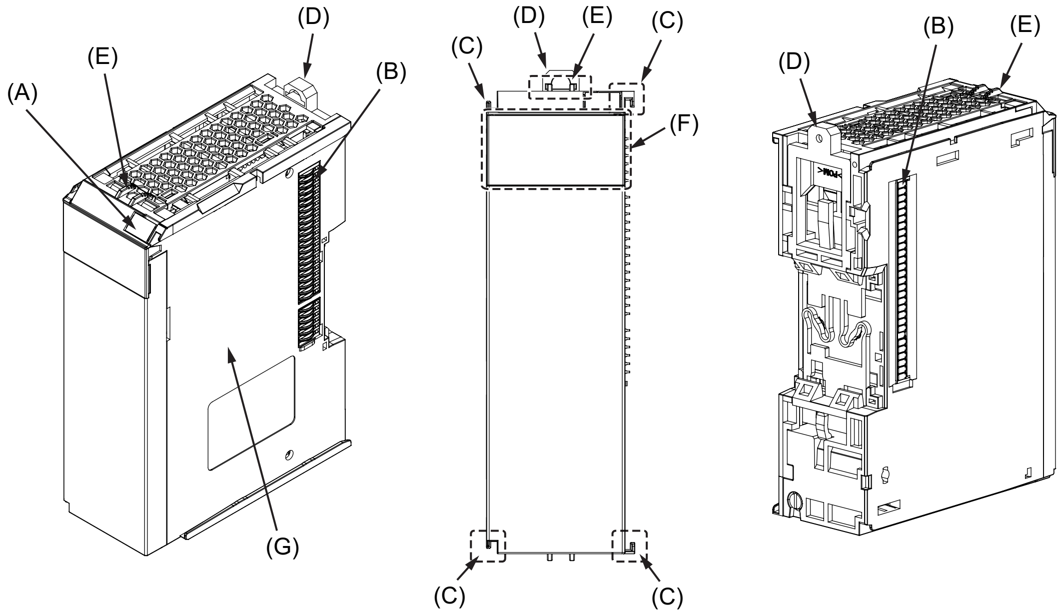
This section gives the names of the parts of the Safety CPU Unit and Safety I/O Units and describes their functions.

3-1 Safety CPU Unit

This section gives the names of the parts of the Safety CPU Unit and describes the meanings of the operation indicators.

3-1-1 Parts and Names

● **NX-SL3300**



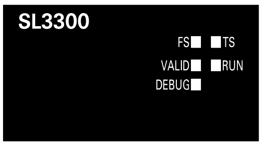
Letter	Name	Function
A	Marker attachment locations	The locations where markers are attached. The markers made by OMRON are installed for the factory setting. Commercially available markers can also be installed. For details, refer to *5-1-2 Attaching Markers* on page 5-4.
B	NX bus connector	This is the NX-series bus connector. It is used to connect an NX-series Safety I/O Unit or other NX Unit.
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 NX Unit or signal I/O status. Refer to *3-1-2 Indicators* on page 3-3. The number of indicators depends on the NX Unit.
G	Unit specifications	The specifications of the NX Unit are given here.

3-1-2 Indicators

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

⚠ WARNING

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



Letter	Name	Function
(A)	Model number display	Displays part of the model number of the Safety CPU Unit. The model number indication is red on Safety Control Units.
(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 Communications Coupler Unit.

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. • The Unit is restarting. • 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 master connections are established and there are no errors in any Safety CPU Unit functions.
		Flashing at 1-s intervals.	One or more FSoE master connections is not established or is 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 to indicate missing Safety I/O Units. 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 safety communications status and safety function status of the Safety CPU Unit.

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

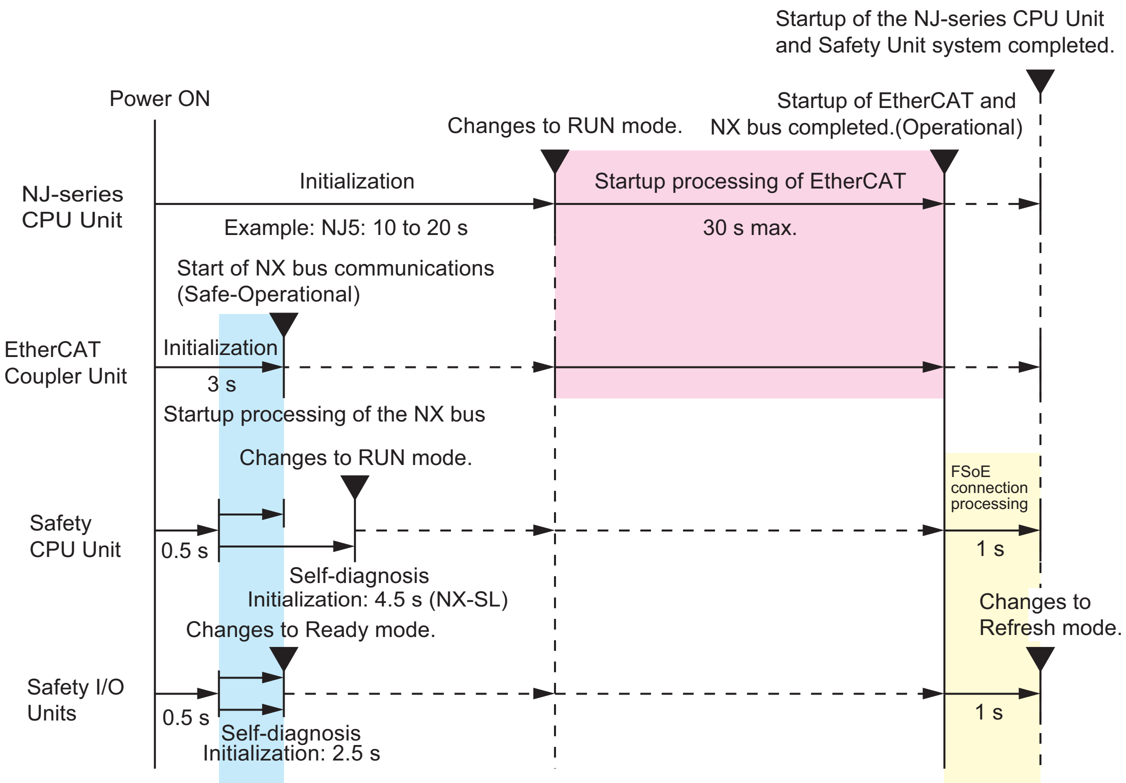
Color	Status		Meaning
Yellow		Lit.	Safety application data from the execution of the safety validation is stored in the non-volatile memory.
---		Not lit.	Safety application data from the execution of the safety validation is not stored in the non-volatile memory, or a fatal fault has occurred.

Refer to *8-3 Operating Modes of the Safety CPU Unit* on page 8-9 for details on the relationship between the operating modes of the Safety CPU Unit and the indicators.

3-1-3 Startup Time

This section describes the startup time of a Safety Control Unit.

The following figure shows the approximate time required for the Unit to enter the state where it can turn ON the safety output terminals (i.e., the Safety CPU Unit is in RUN mode and the FSoE connections are established) after the power is turned ON.

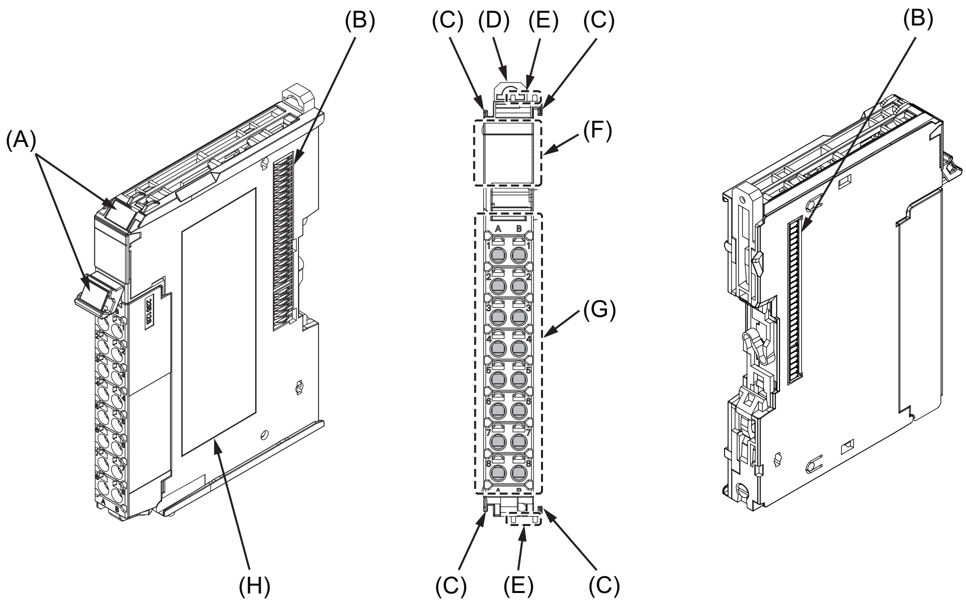


3-2 Safety I/O Units

This section gives the names of the parts of the Safety I/O Units and describes the operation indicators, terminal block layouts, and safety I/O functions.

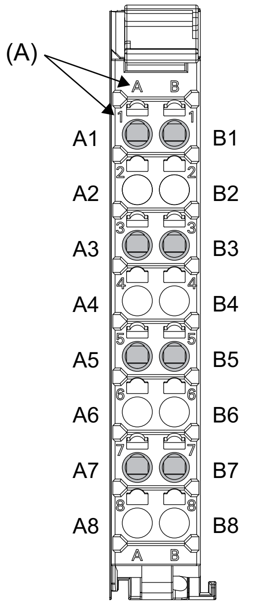
3-2-1 Parts and Names

NX-SIH410, NX-SID800, NX-SOD400, and NX-SOH200

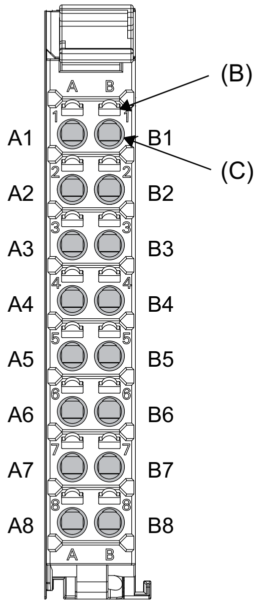


Letter	Name	Function
A	Marker attachment locations	The locations where markers are attached. The markers made by OMRON are installed for the factory setting. Commercially available markers can also be installed. For details, refer to *5-1-2 Attaching Markers* on page 5-4.
B	NX bus connector	This is the NX-series bus connector. Connect this connector to another Unit, such as the NX-series Safety CPU Unit or a Safety I/O Unit.
D	Unit hookup guides	These guides are used to connect two Units.
C	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 NX Unit or signal I/O status. Refer to *3-2-2 Indicators* on page 3-8. The number of indicators depends on the NX Unit.
G	Terminal block	The terminal block is used to connect to external devices. It connects the safety outputs. The number of terminals depends on the NX Unit.
H	Unit specifications	The specifications of the NX Unit are given here.

● Terminal Blocks



Eight-terminal Block



Sixteen-terminal Block

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. 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 come in two types depending on the number of terminals that can be used. There are 8-terminal and 16-terminal blocks.
You can use only one of the two types of terminal blocks given above with a Unit that has a screwless clamping terminal block.

The terminal block must have the same number of terminals that the Unit is designed for.

Additional Information

The 8-terminal block does not have wire holes and release holes for the following terminals.

- A2, A4, A6, A8, B2, B4, B6, and B8

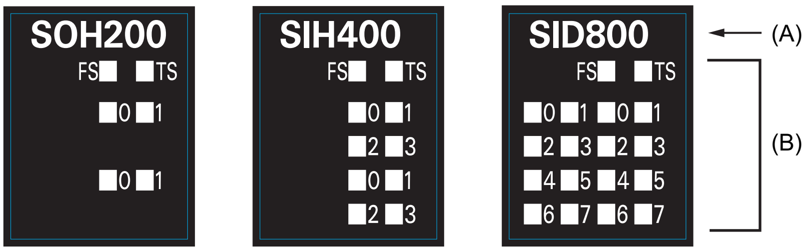
3-2-2 Indicators

A Safety I/O Unit has indicators that give the status of the Unit, communications, and the safety I/O terminals.

⚠ WARNING

Do not use the indicators on the NX-series Safety 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 indicator pattern depends on the number of I/O points, as shown below.



Unit with 2 I/O Points 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 Unit. 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 Unit.

The rest of this section gives the indicator specifications.

TS Indicator

The TS indicator shows the current status of the Safety I/O Unit and the communications status with the Communications Coupler Unit.

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

Color	Status		Meaning
Green		Lit.	The Unit is operating normally. The Unit is ready to refresh I/O.
		Flashing at 2-s intervals.	Initializing
Red		Lit.	A hardware error, WDT error, or other critical error has occurred.
		Flashing at 1-s intervals.	An NX bus communications error or other recoverable minor error that is attributed to the NX bus has occurred.
---		Not lit.	Power is not being supplied.

FS Indicator

The FS indicator shows the safety communications status and safety function status of the Safety I/O Unit.

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

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

IN/OUT Indicator

The IN/OUT indicators show the signal I/O status of the safety input terminals and safety output terminals.

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

IN ERR/OUT ERR Indicator

The IN ERR/OUT ERR indicators show the error status of the safety input terminals and safety output terminals.

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

3-3 Safety I/O Functions

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

3-3-1 Safety Input Functions

Connectable Input Devices

The Safety Input Unit diagnoses the connected external devices through the safety input terminals. The general-purpose 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.

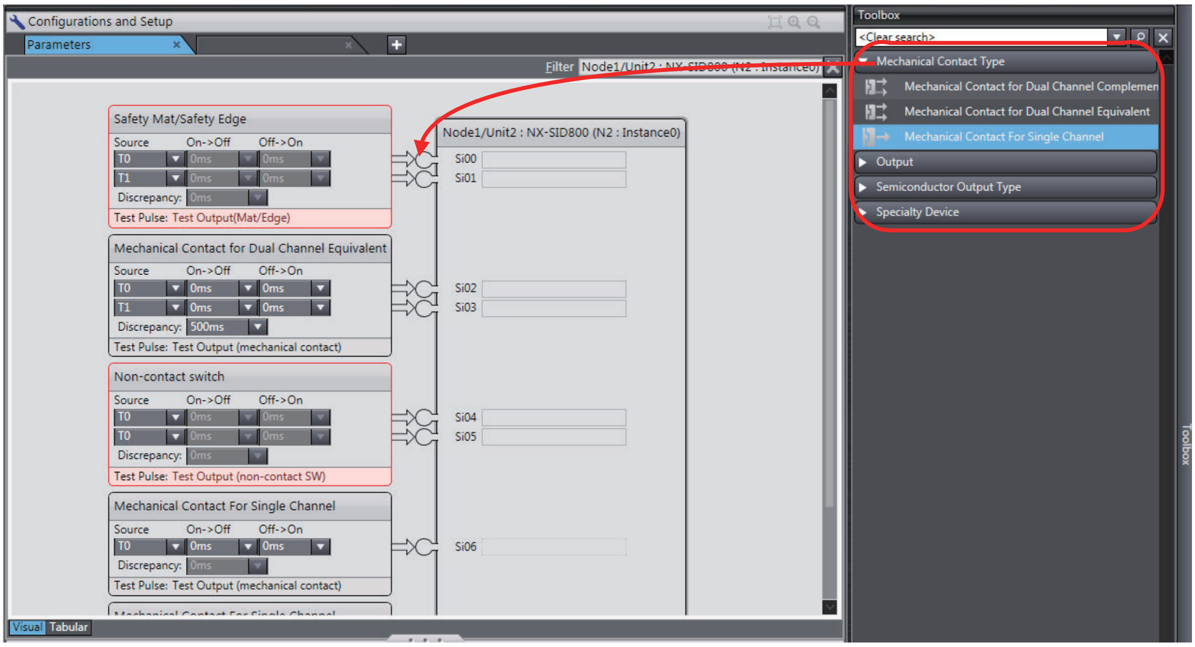
Type	Description
Safety input devices with mechanical contacts	Emergency stop pushbutton switches, safety limit switches, safety door switches, enable switches, two-hand switches, and user mode switches
Safety input devices with semiconductor outputs	Safety light curtains and safety laser scanners
Standard input devices	Reset switches

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

Type	Model and corresponding PL and safety category
OMRON Single-beam Safety Sensors	E3ZS and E3FS * Conforms to Type 2 and PLc.
OMRON Non-contact Door Switches	D40Z * Conforms to PLe and Safety Category 4. D40A * Conforms to PLd and Safety Category 3.
OMRON Safety Mats	UM * 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 *3-3 Safety I/O Functions* on page 3-10 for details.



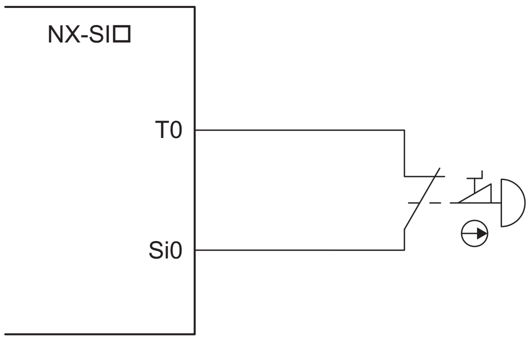
Connecting Input Devices

This section describes the connection methods for input devices.

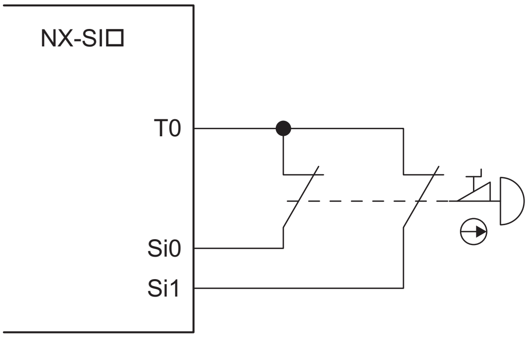
● Devices with Mechanical Contacts

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).

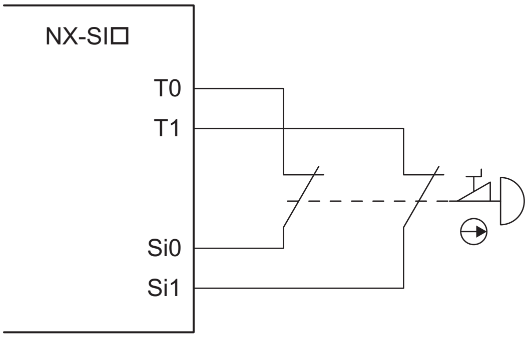
- Single-channel Input



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

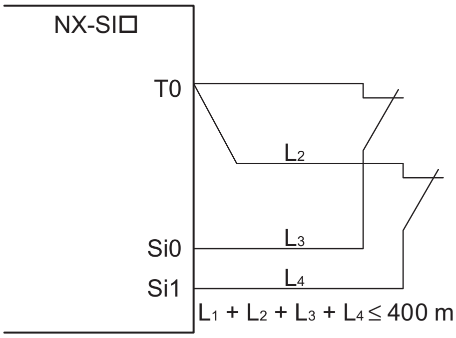


- Dual-channel Input When I/O Short Detection between Lines Is Required



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.



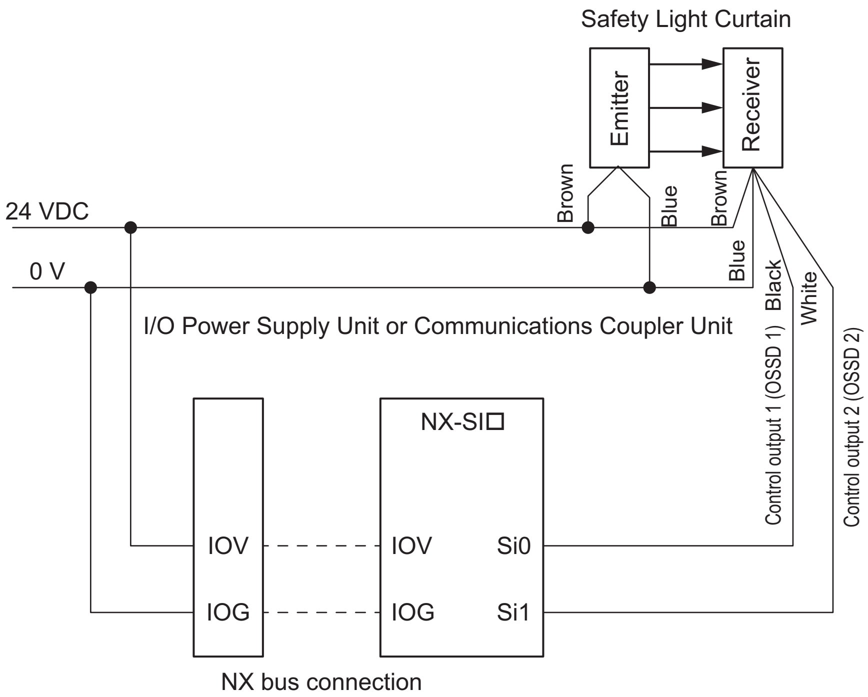
Additional Information

You can detect short-circuits between two input channels with the following methods:

- Dual channel equivalent inputs: With Test Pulse
- Dual channel complementary inputs: Without Test Pulse or With Test Pulse

● Devices with Semiconductor Outputs

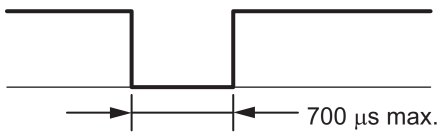
The signal from a device with a semiconductor output, such as a light curtain, is input to a safety input terminal (Si).



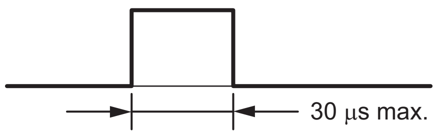
Precautions for Correct Use

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.



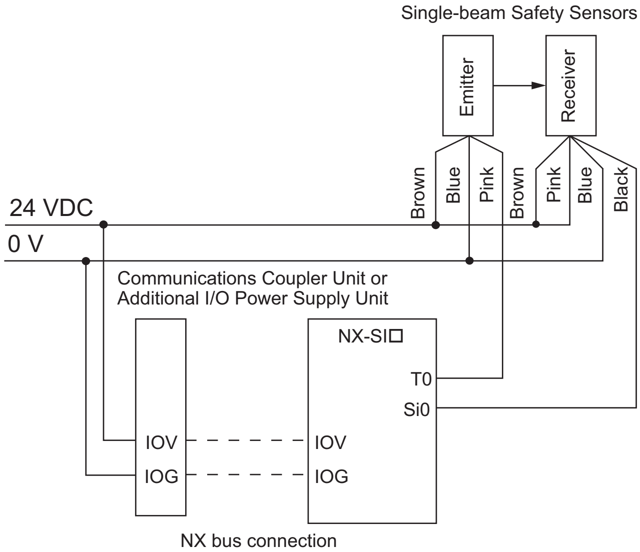
- ON pulse width when semiconductor output is OFF: 30 μs max.



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

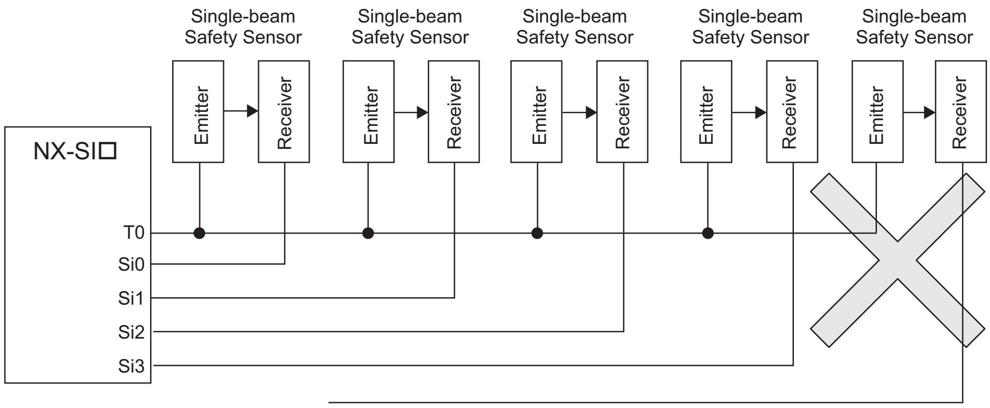
● E3ZS/E3FS Single-beam Safety Sensors

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

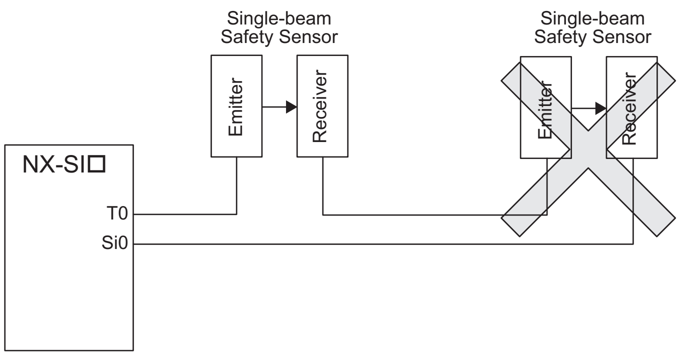


Precautions for Correct Use

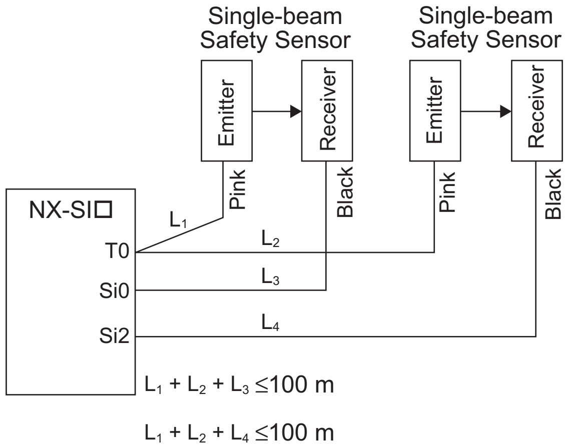
- The maximum number of connections per Unit is as follows:
  NX-SIH400: 4
- You can branch the connections to up to four Single-beam Safety Sensors for each Test Output.



- Series connections are not possible.



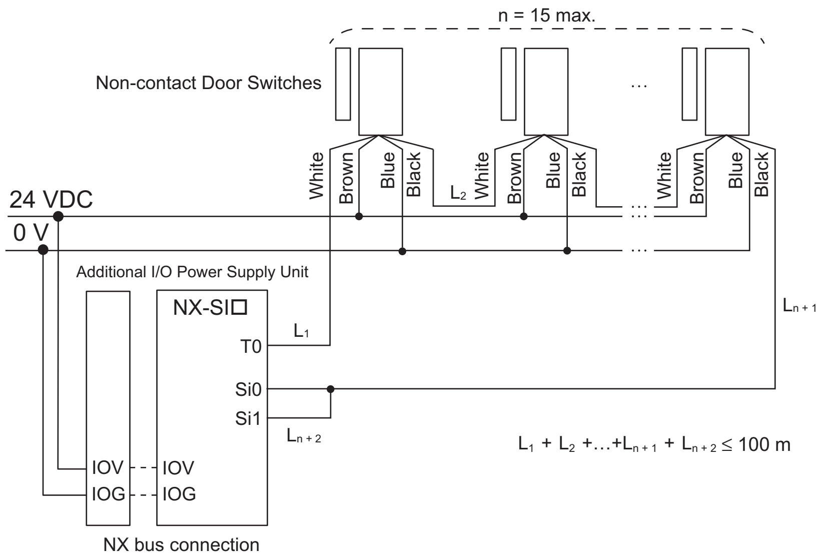
- The total wiring length for the E3ZS/E3FS Single-beam Safety Sensors is 100 m max.



- 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.

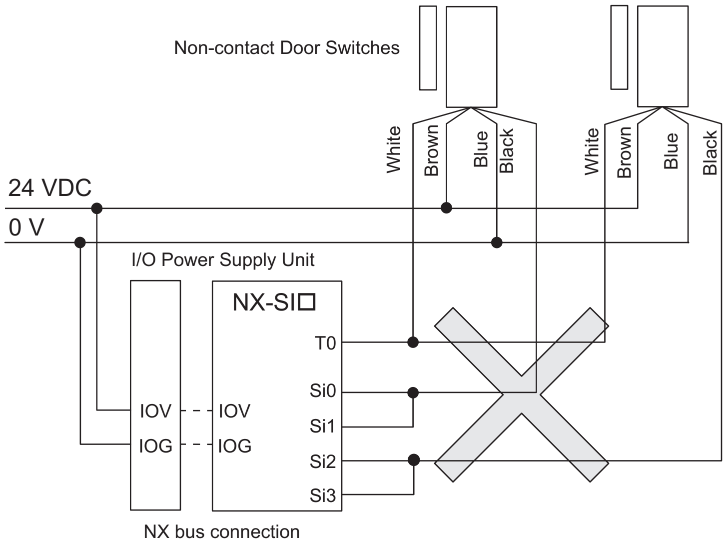
● D40A/D40Z Non-contact Door Switches

The non-contact door switch output (black line) from the OMRON 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/D40Z Non-Contact Door Switch input (white line).



Precautions for Correct Use

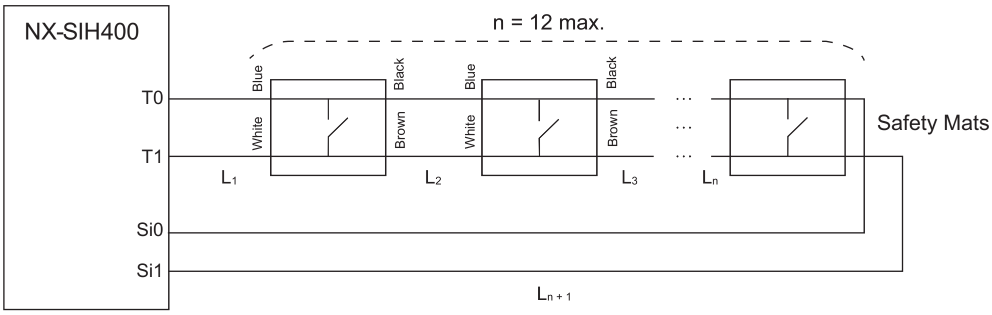
- The maximum number of connections per Unit is as follows:
  NX-SIH400: 20 (10 connected in series × 2 series)
- You can connect up to 10 Non-contact Door Switches to each test output terminal.
- You cannot branch the connections to more than one Non-contact Door Switch from the same test output terminal.



- The total wiring length (L1 + L2 +…+ Ln+2 in the figure above) for the D40A or D40Z Non-contact Switches is 100 m max.
- 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.

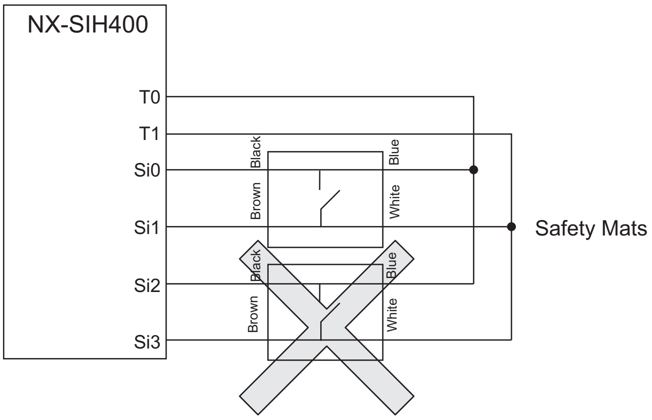
UM Safety Mats

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



Precautions for Correct Use

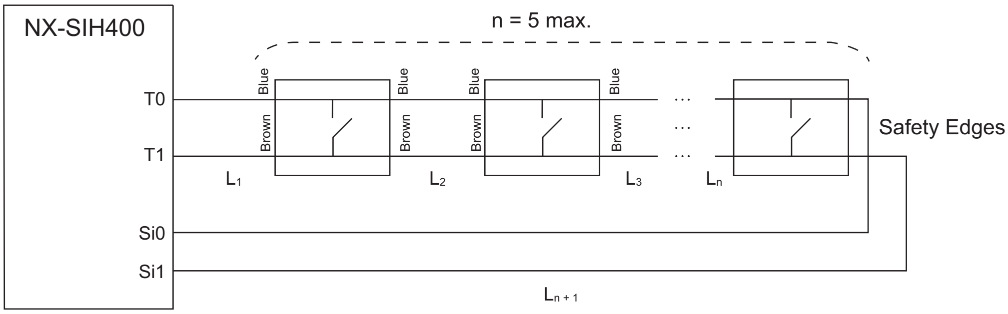
- The maximum number of connections per Unit is as follows:
  NX-SIH400: 12 (12 connected in series × 1 series)
- You can connect up to 12 Safety Mats to the two test output terminals.
- You cannot branch the connections to more than one Safety Mat or Safety Edge from the same test output terminal.



- The total wiring length (L1 + L2 +…+ Ln+1) for the UM Safety Mats is 100 m max.
- The UM Safety Mat 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.

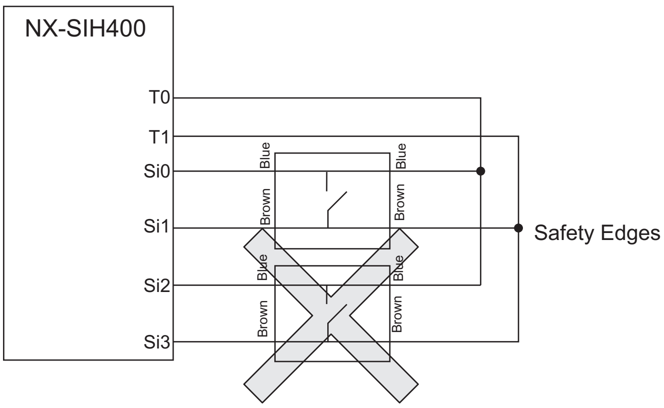
● SGE Safety Edges

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



Precautions for Correct Use

- The maximum number of connections per Unit is as follows:
  NX-SIH400: 5 (5 connected in series × 1 series)
- You can connect up to five Safety Edges 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.

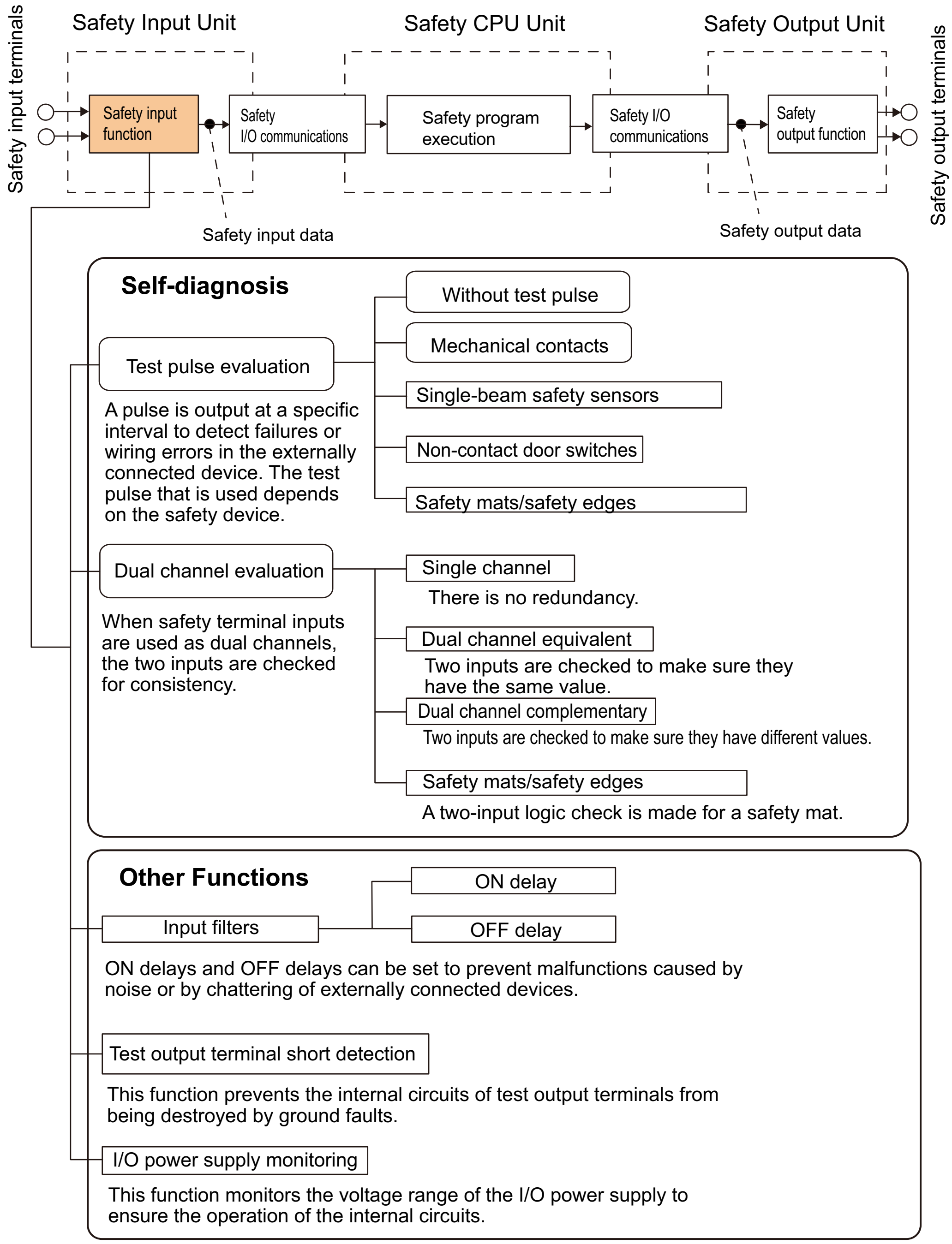
- The SGE Safety Edge can be used in a Safety Category 3 or lower or a PLd or lower application. It cannot be used in a Safety Category 4 or PLe application.

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. 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. 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

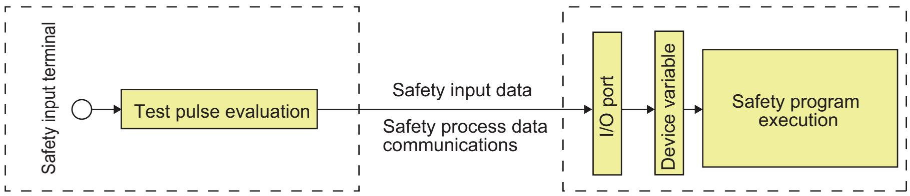
The Test Pulse Mode setting is used to output a test pulse that is suitable for the external device when the Test Pulse Diagnosis parameter is set to *with Test Pulse*.

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 Switch	An OMRON D40A or D40Z Non-contact Door Switch is connected. Test signals for the D40A or D40Z will be output.
Safety Mat/Safety Edge	An OMRON UM Safety Mat or SGE Safety Edge (4-wire) is connected. A test signal for Safety Mat/Safety Edge diagnosis is output.

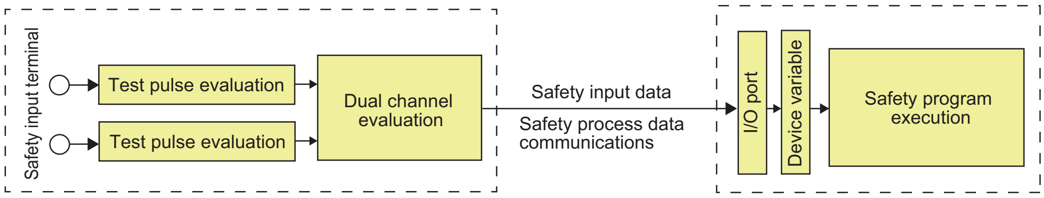
Dual Channel Evaluation

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: 1,000 [ms], 3: 2,000 [ms], 4: 4,000 [ms], 5: 8,000 [ms], 6: 16,000 [ms], 7: 32,000 [ms], 8: 64,000 [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 signal 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

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

n = Even number

*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.

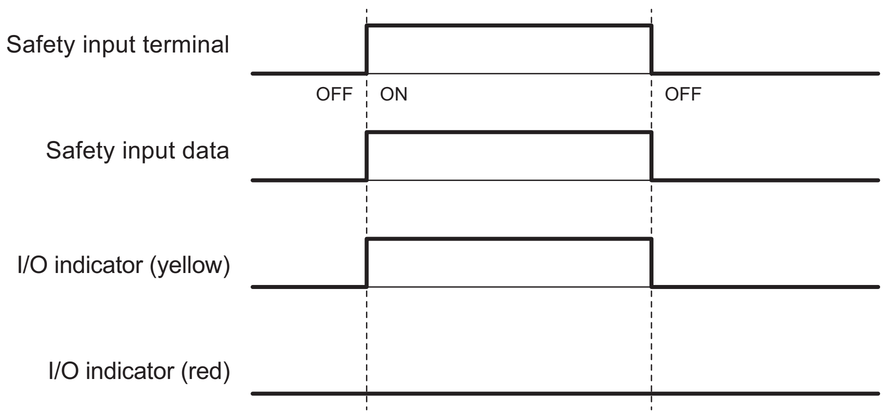
- Relationship between Safety Mat Status and Safety Input Data for Safety Mat/Safety Edge Inputs

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)

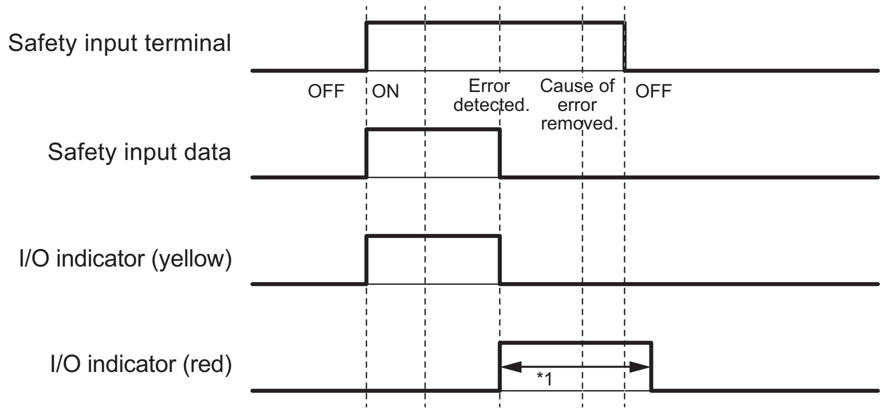
n = Even number

*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

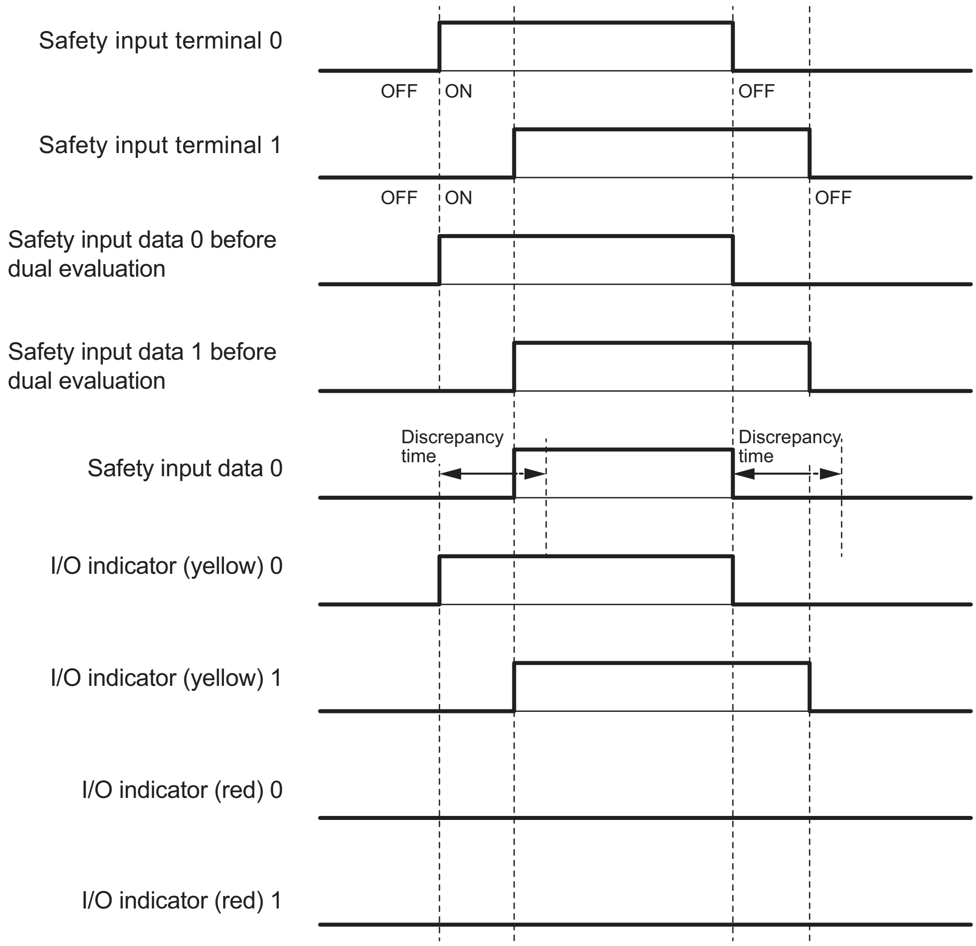


- Operation for Single Channel: Test Pulse Evaluation Error for Stuck-at-high Error

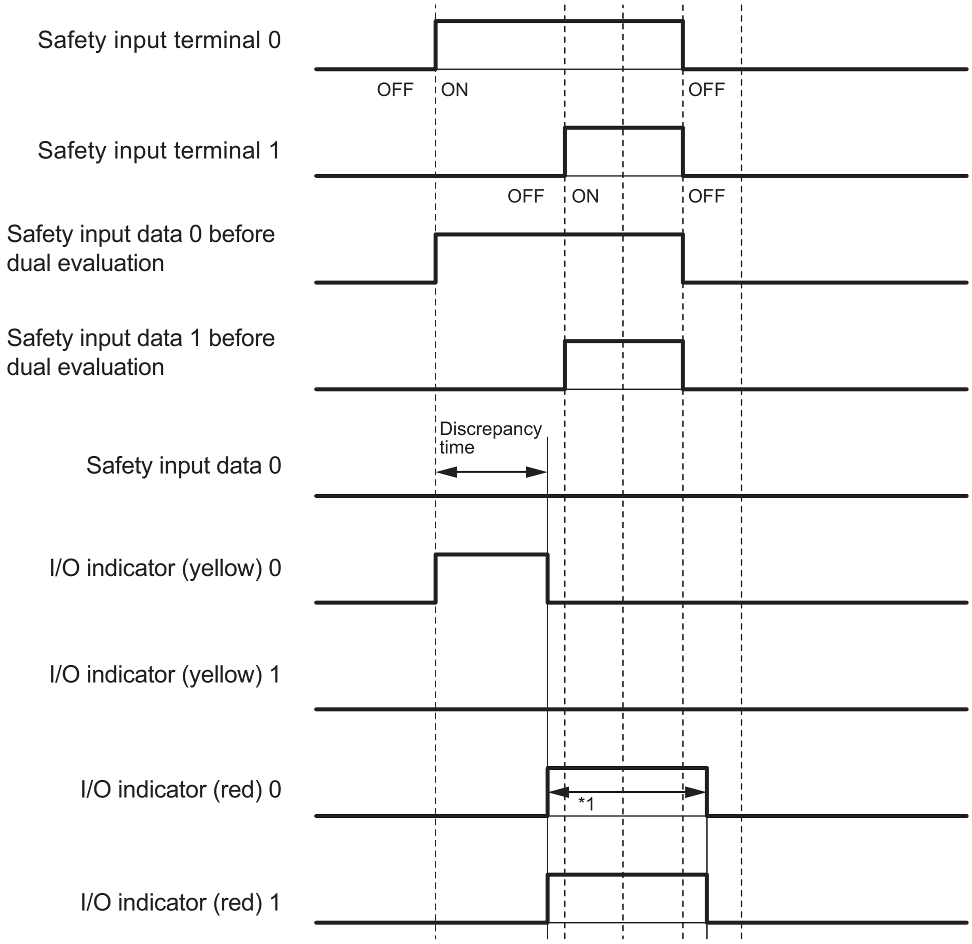


*1. This is the time that the error status (control data, status data, and indicator status) is held (1 s min.).

- 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.	---
Dual Channel Equivalent	Without Test Pulse	Not detectable. [*2]	Not detectable.	Detectable when input turns ON.	Not detectable. [*1]
	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.

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 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	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 Non-contact Switch	Detectable.	Not detectable.	Not detectable.	---	Not detectable.	Not detectable.
D40Z Non-contact Switch	Detectable.	Not detectable.	Not detectable.	---	Detectable.	Detectable.
Safety Mat/Safety Edge	Detectable.	Detectable.	Detectable.	Not detectable.	Not detectable.	---

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: 1,536 [ms]

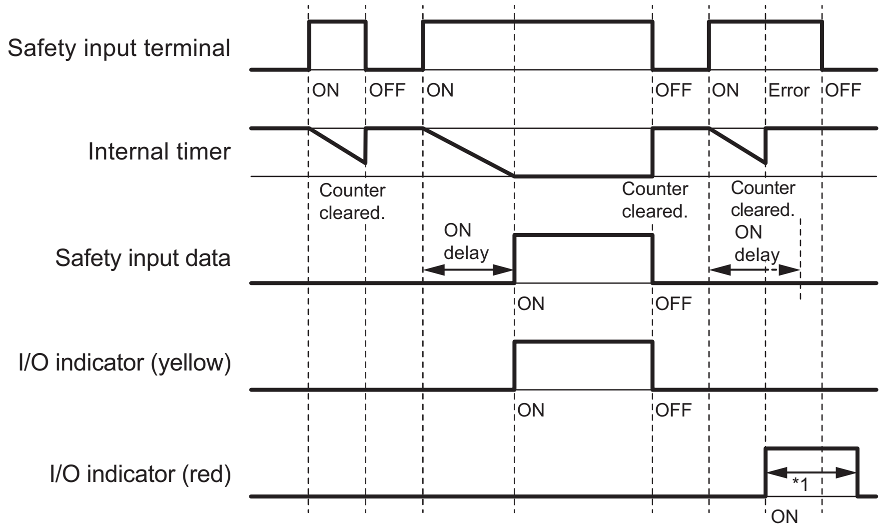
The effect of chattering from external devices can be reduced more by increasing the delay time, but this will slow the response to input signals. The input filter can be used with dual channel evaluation.

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 *4-1 Safety Reaction Times* on page 4-2.)

● Operation with an ON Delay

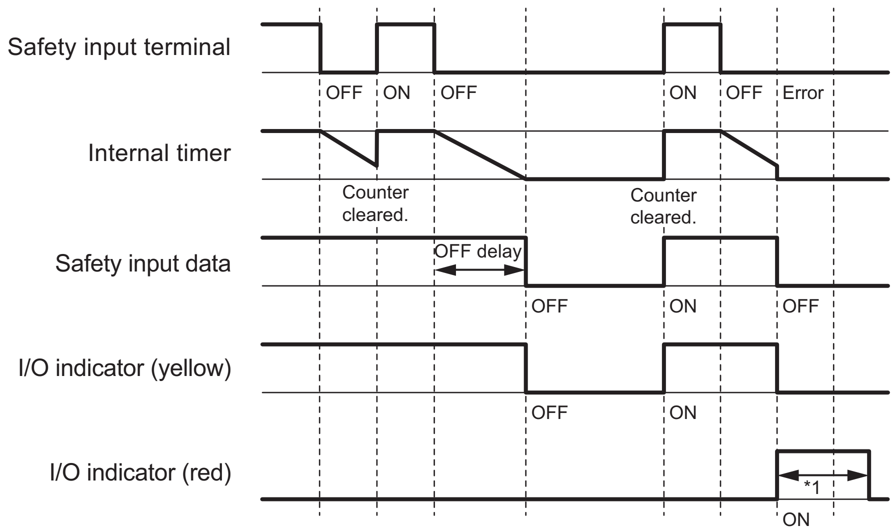
You can filter out ON pulses for the width that is set with the ON delay time.



*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

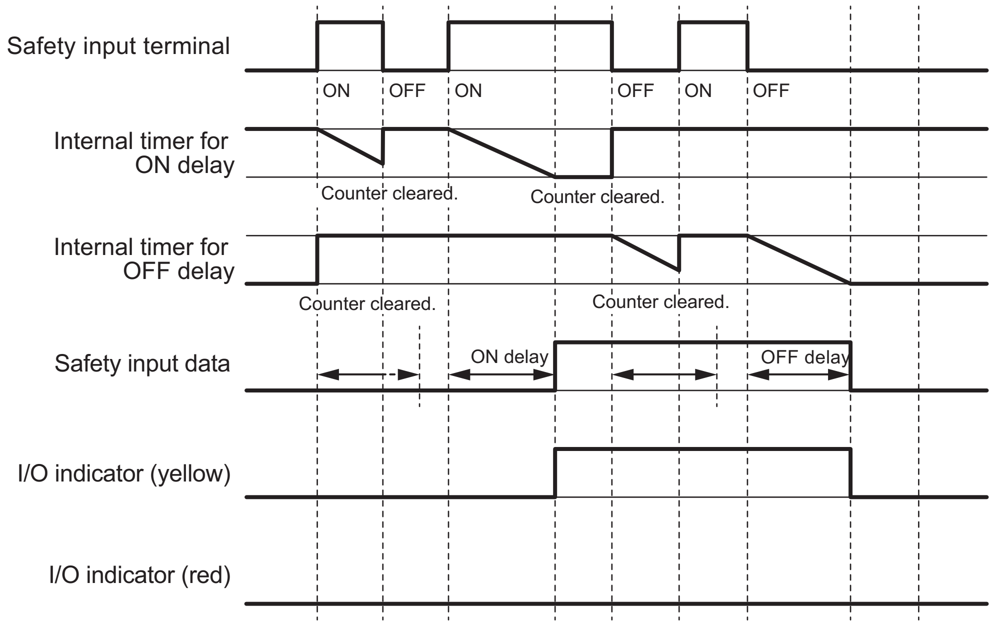
You can filter out OFF pulses for the width that is set with the OFF delay time.



*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 9 Troubleshooting*.

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 9 Troubleshooting*.

This function does not work if all of the terminals are set as unused terminals.

3-3-2 Safety Output Functions

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

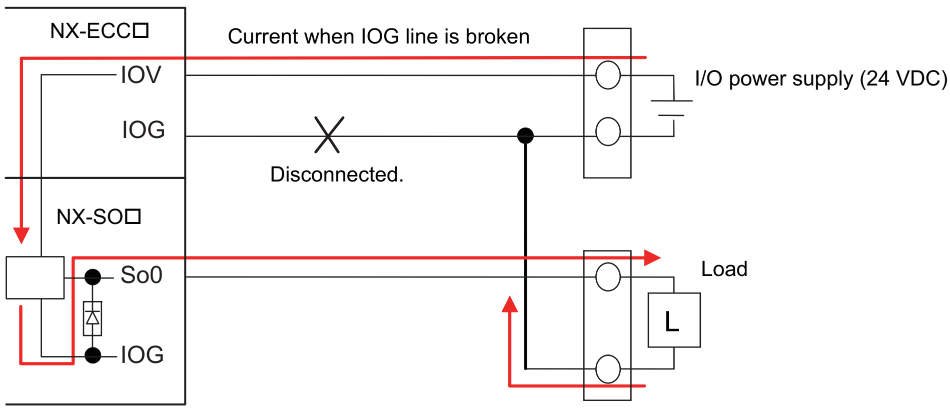
The connection of incandescent lamps is not supported. Connect them to an NX-series Digital Output Unit.

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 *3-3 Safety I/O Functions* on page 3-10 for details.

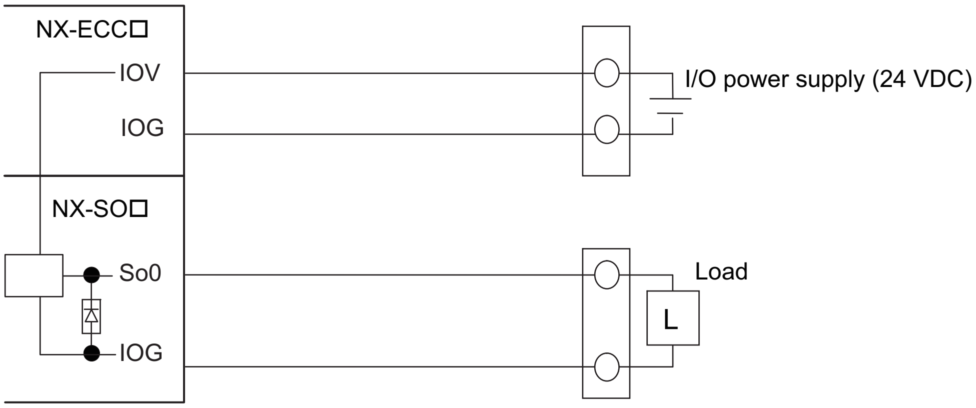
Connecting the I/O Power Supply

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.



3-3 Safety I/O Functions
3
3-3-2 Safety Output Functions

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.

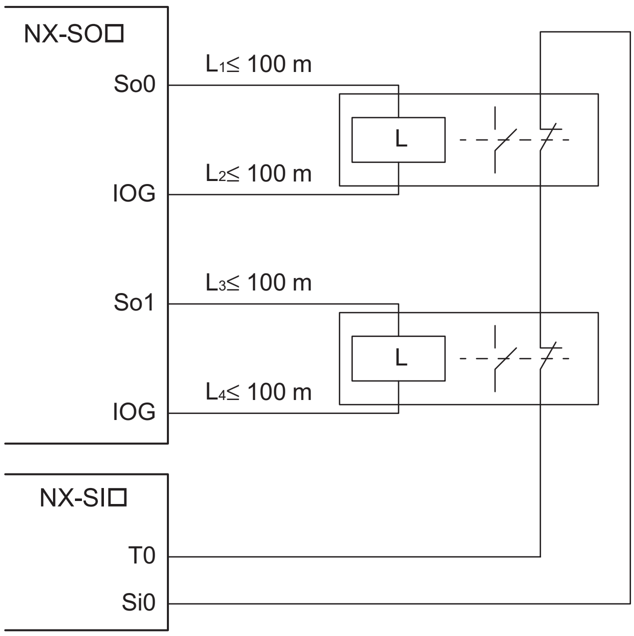


Connecting Output Devices

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.

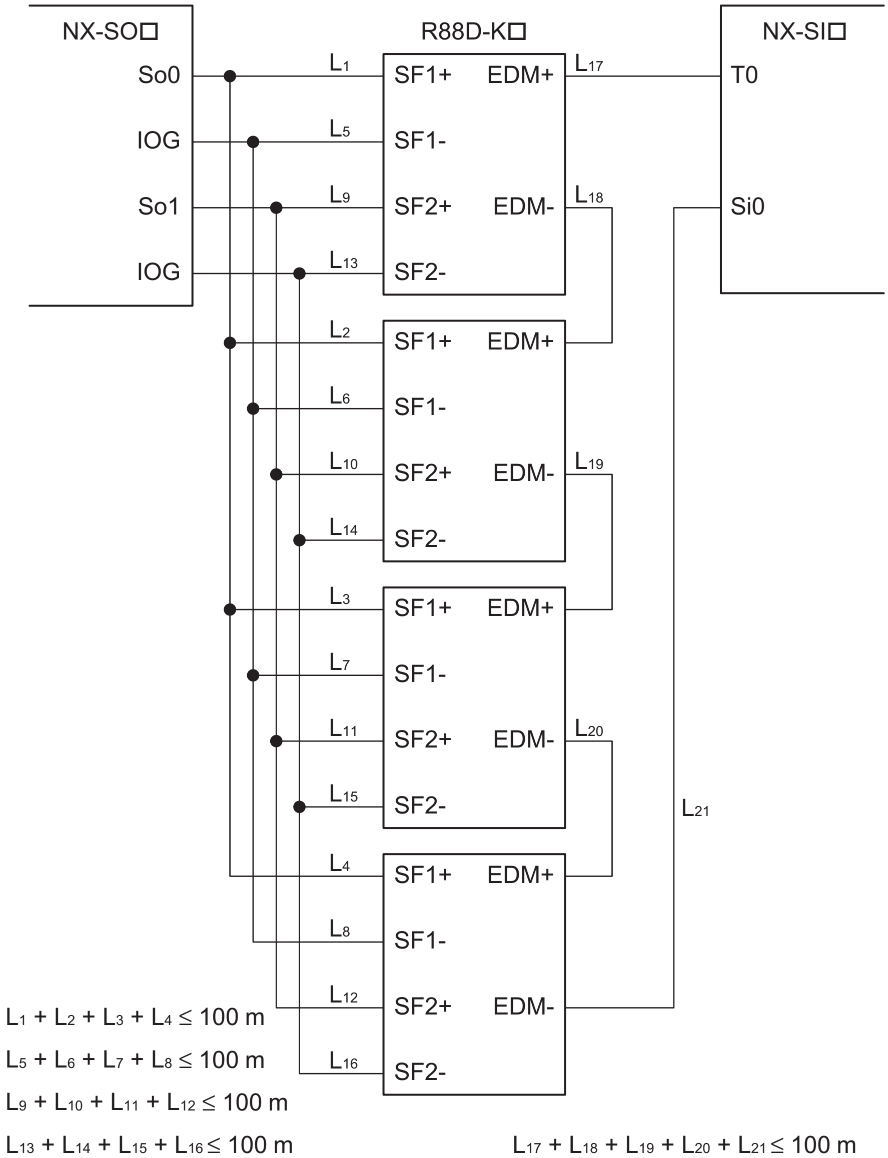


Precautions for Correct Use

- 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 *3-3-1 Safety Input Functions* on page 3-10.

● Servo Drive

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



Precautions for Correct Use

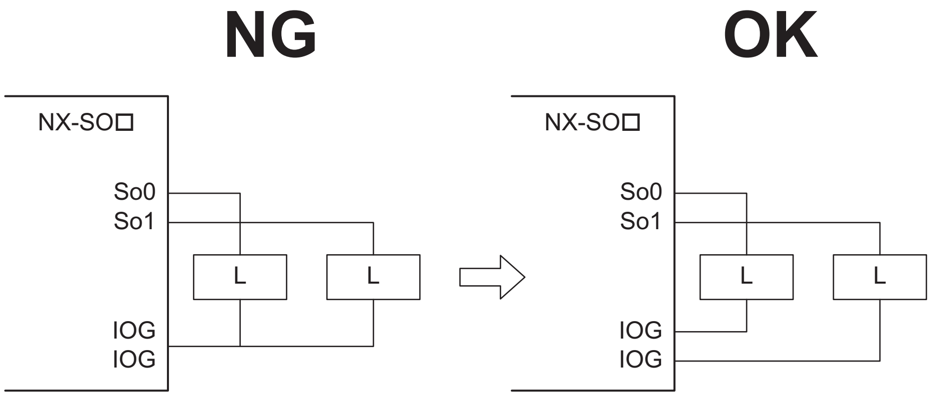
- The maximum number of connections per Unit is as follows:
  When NX-SOD400 and NX-SI☐ are used: 8 (4 connected in series × 2 series)
  When NX-SOH200 and NX-SI☐ are used: 4 (4 connected in series × 1 series)
- You can connect up to four servo drives or inverters for every two safety output terminals and every safety input terminal.
- 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.
- An R88D-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 *OMNUC G5-series User's Manual* (Cat. No. I571) for details on the safety function settings and the precautions for the correct use of the R88D-K☐ Servo Drive.

Additional Information

- A special connector (R88A-CNK81S) is required to connect the R88D-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.

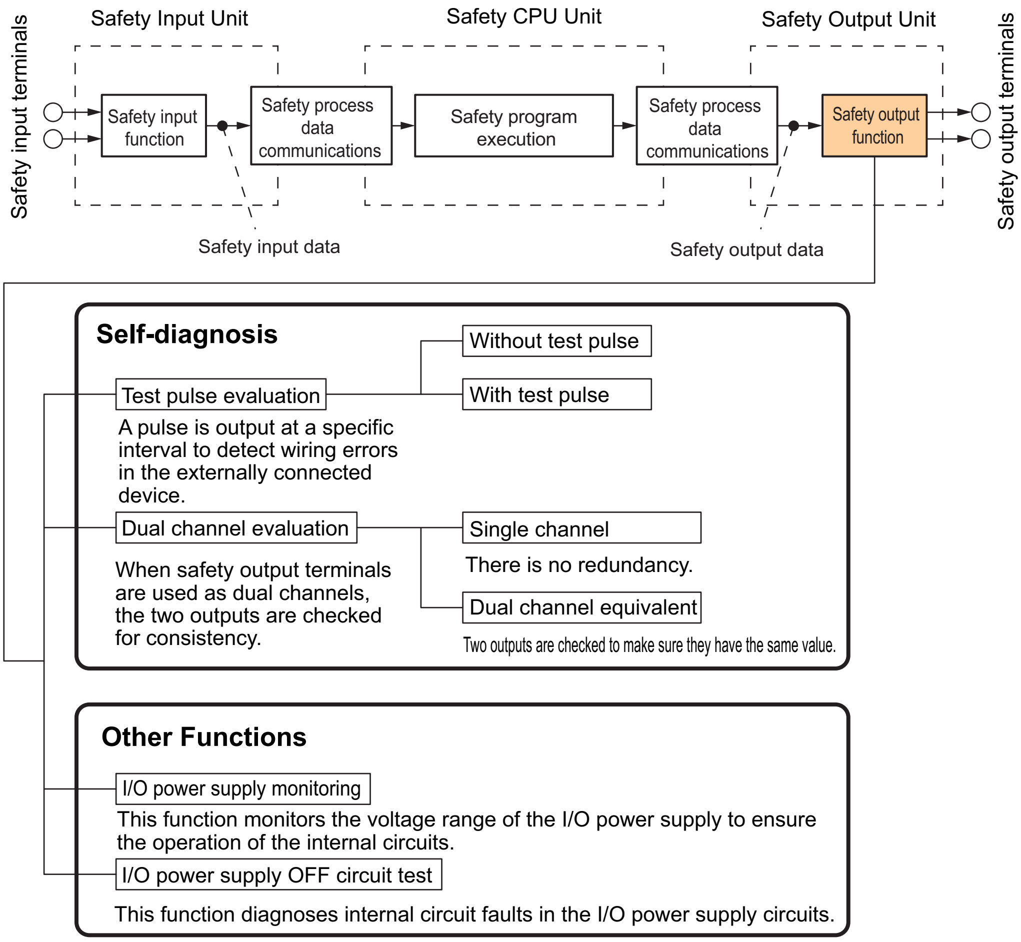
● Connecting More Than One Output Device

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 3-35 for the errors that can be detected for each parameter setting.

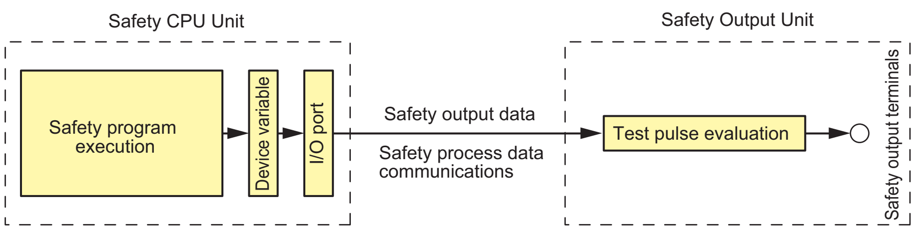
Precautions for Correct Use

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.

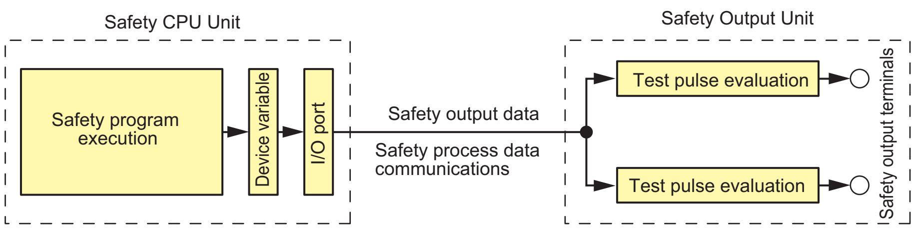
Dual Channel Evaluation

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

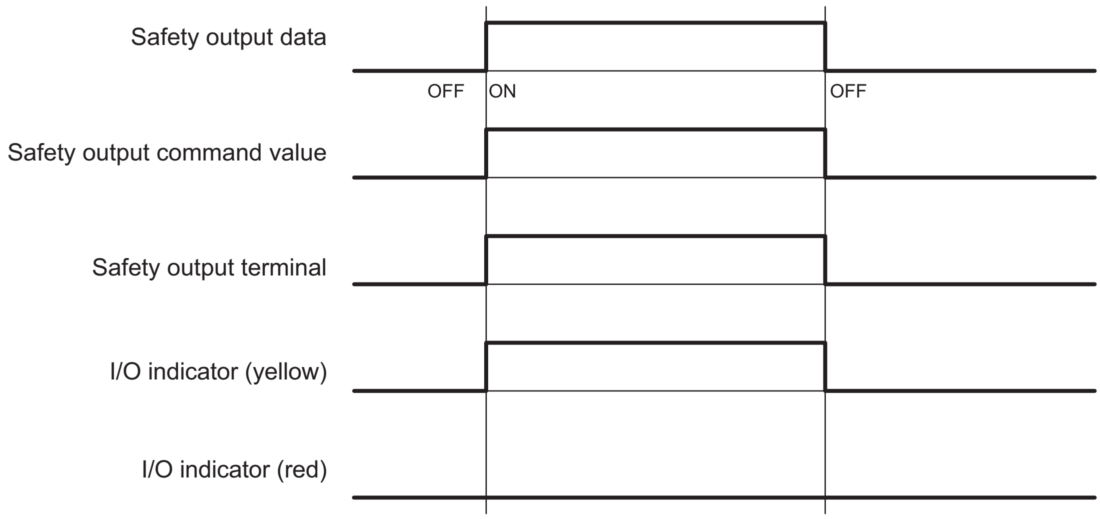
Single/Dual	Safety output data	Output signal on the safety output terminal	Meaning of status
	So (x)	So(x)	
Single Channel	0	0	Inactive (OFF)
	1	1	Active (ON)

- Relationship between Safety Output Data and Signals Output from Safety Output Terminals for Dual-channel-equivalent Outputs

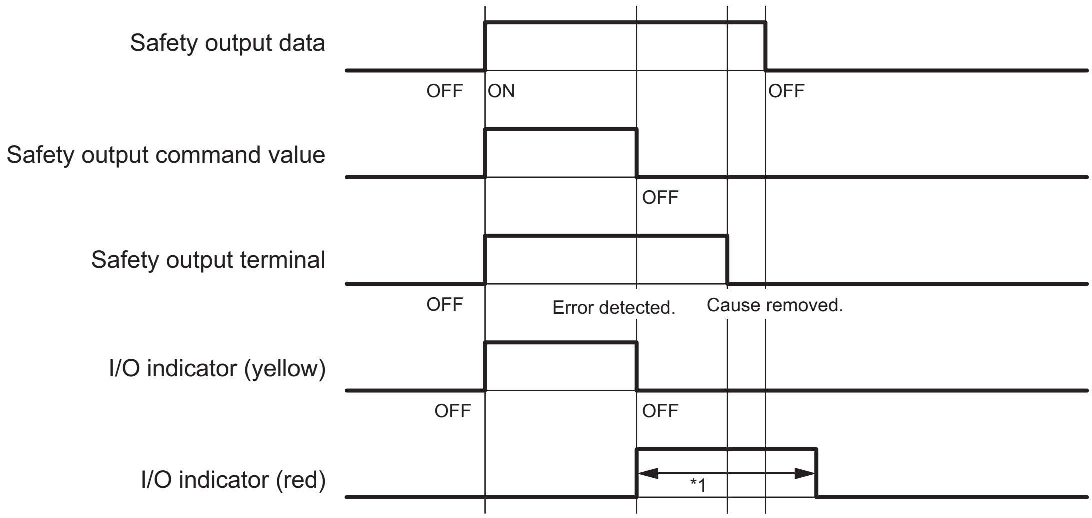
Single/Dual	Safety output data	Output signal on the safety output terminal		Meaning of status
	So (x)	So (n)	So (n+1)	
Dual Channel Equivalent	0	0 (OFF)	0 (OFF)	Inactive (OFF)
	1	1 (ON)	1 (ON)	Active (ON)

n = Even number

- Operation for Single Channel: Normal Operation

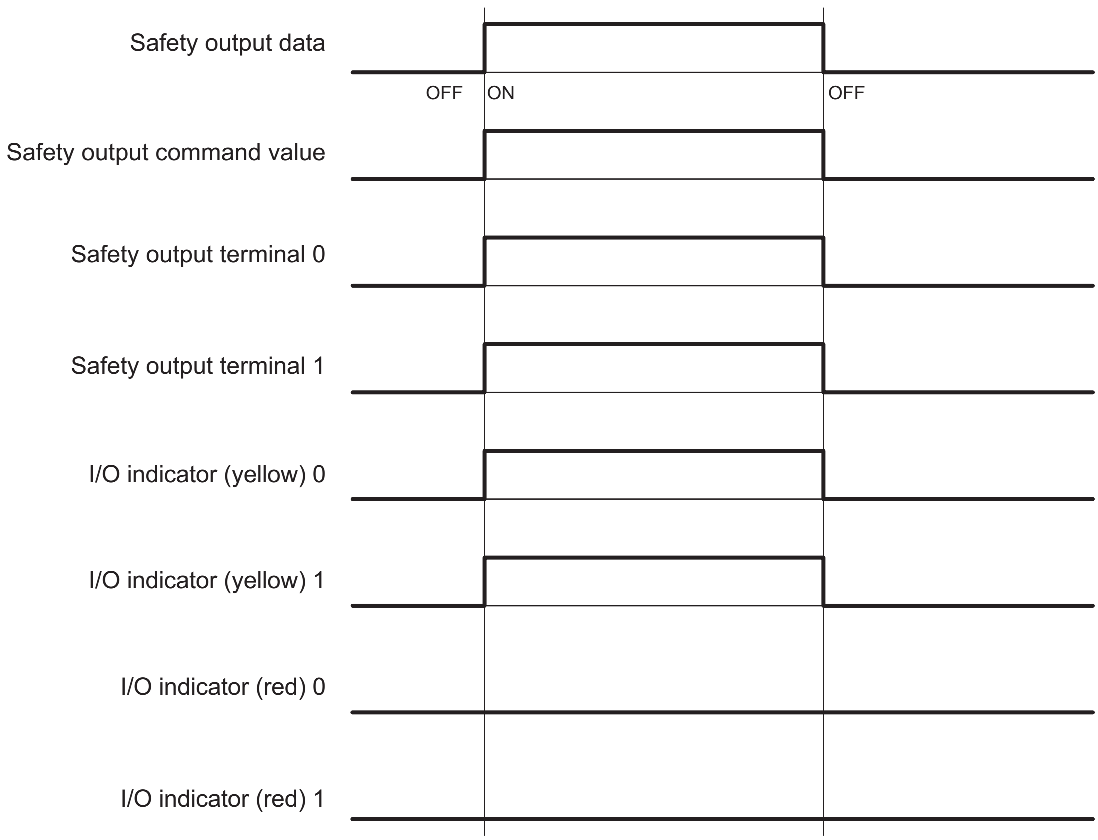


• Operation for Single Channel: Test Pulse Diagnosis Error for Stuck-at-high Error

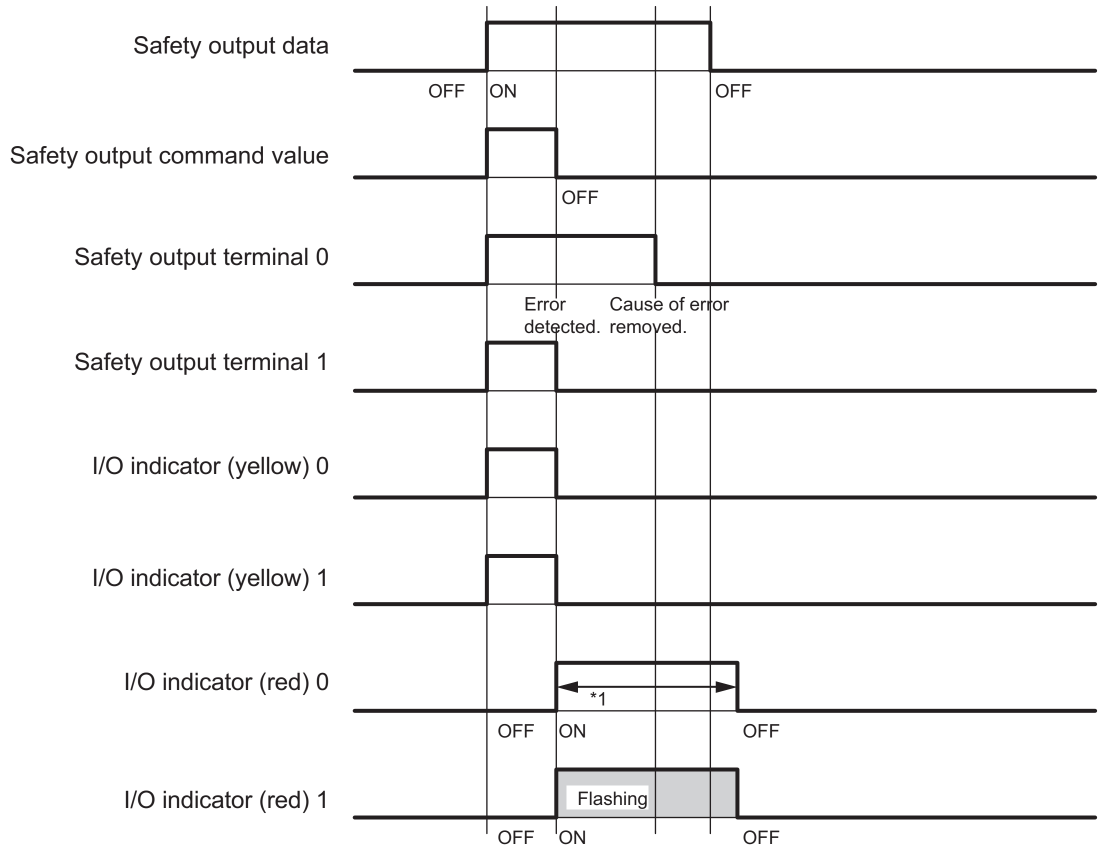


*1. This is the time that the error status (control data, status data, and indicator status) is held (1 s min.).

• Operation for Dual Channel Equivalent Outputs: Normal Operation



- Operation for Dual Channel Equivalent Outputs: Test Pulse Diagnosis 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		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.

Ground Fault Detection for Safety Output Terminals

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 9 Troubleshooting*.

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 9 Troubleshooting*.

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 diagnoses internal circuit faults in the I/O power supply circuits. If an internal circuit fault is detected, 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.

4

Calculating Safety Reaction Times

This section describes how to calculate safety reaction times for Safety Control Units.

4-1 Safety Reaction Times

This section describes the safety reaction times (i.e., the safety response performance) of Safety Control Units.
For all safety chains, the longest time required to stop moving equipment from when a safety input was activated must satisfy the required specifications.

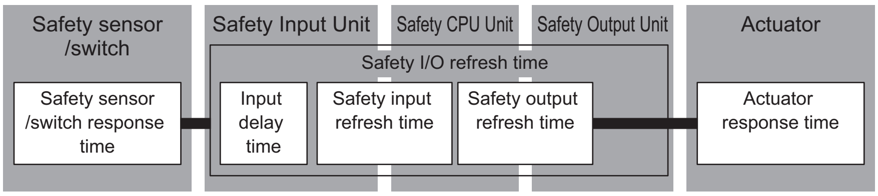
4-1-1 Safety Reaction Times

A safety reaction time is the time required to turn OFF an output when considering failures and breakdowns. Reaction times are used to calculate the safety distances.
The safety chain is the logical connections that are required to achieve a safety function, including the safety input device, NX-series Safety Control Units, and safety output device.

4-1-2 Calculating Safety Reaction Times

This section gives the safety I/O reaction times for the Safety I/O Units.

As shown in the following figure, a safety reaction time is the sum of the safety sensor/switch response time, safety I/O refresh time, and actuator response time.



The contents of each time element is described in the following table.

Time element	Description
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. The following values apply when an OMRON Special Safety Input Device is connected to a Safety Input Unit. E3ZS/E3FS Single-beam Safety Sensors: 10 ms D40A Non-contact Door Switches: 6 ms + 0.4 ms x No. of linked Switches D40Z Non-contact Door Switches: 18 ms UM Safety Mats: 10 ms SGE Safety Edges: 10 ms
Safety I/O refresh time	Calculate the sum of the following configuration elements. This is the time from when the safety input terminal changes until the change goes through the Safety CPU Unit and the safety output terminal turns OFF. Calculation: Find the sum of the following configuration elements. Safety I/O refresh time = Input delay time + Safety input refresh time + Safety output refresh time • The input delay time is the input OFF delay time that is set for the safety input terminal on the Safety Input Unit. • The safety input refresh time is the value of the FSoE watchdog timer between the Safety CPU Unit and Safety Input Unit. • The safety output refresh time is the value of the FSoE watchdog timer between the Safety CPU Unit and Safety Output Unit.
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.

The safety input refresh time and safety output refresh time, i.e., the FSoE connection FSoE watchdog timer times, are automatically calculated by the Sysmac Studio from the following values.

- Safety task period
- Task period of the primary periodic task in the NJ-series CPU Unit
- FSoE slave response time (an inherit value of the FSoE slave)

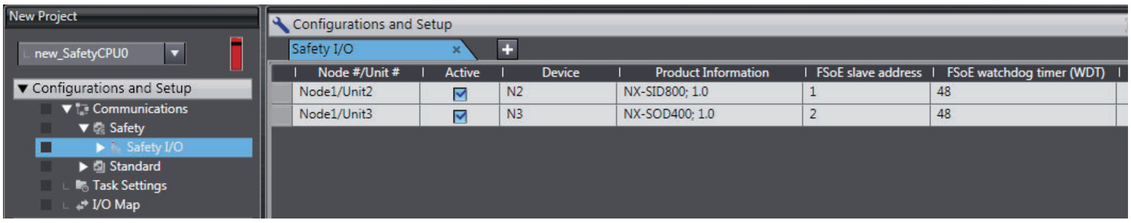
● Checking FSoE Watchdog Timers

Use the following procedure to check the FSoE watchdog timers.

1 In the Multiview Explorer, select the 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 Tab Page is displayed.



The values of the FSoE watchdog timers are displayed in the *FSoE WatchdogTimer* Column for the Safety I/O Units.

Refer to *4-2 Safety Task* on page 4-4 for information on the safety task period.

4-1-3 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 program.
- Reduce the task period of the primary periodic task.
- Reduce the number of Safety I/O Units.
- Reduce the number of variables that are exposed to the NJ-series CPU Unit.

4-2 Safety Task

This section describes the safety task of the Safety CPU Unit.
The safety reaction times are affected by the safety task period of the Safety CPU Unit.

4-2-1 Safety Task

The safety task is used to assign an execution condition to a series of processes, such as I/O refreshing and user 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 the 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.

Data exchange between the Safety CPU Unit and Safety I/O Units is called communications receive processing and communications send processing.

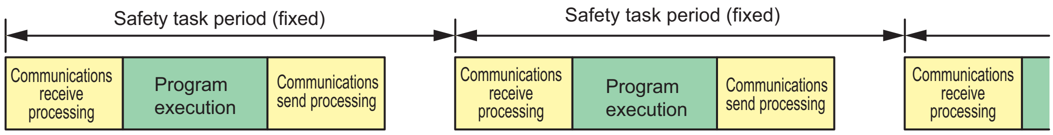
Type of task	Number of tasks	Task execution priority	Execution condition	Main processing contents
Safety task	1	None	The safety task is executed once every safety task period during operation in RUN or DEBUG mode.	Communications receive processing, program execution, and communications send processing

*1. There is no limit to the number of programs.

4-2-2 Operation of Safety Task

The following operation is performed for the safety task.

Input data processing for I/O refreshing (NX bus and EtherCAT communications), user program execution, and output data processing for I/O refreshing (NX bus and EtherCAT communications) are performed repetitively, i.e., each safety task period.



The safety task period is the fixed time period at which the safety task is repetitively 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.

The minimum period is calculated from the following values.

- Safety program execution time (depends on the size of the programs and the function blocks that are used)
- Number of Safety I/O Unit connections
- Primary period of the NJ-series CPU Unit
- NX bus refresh cycle for EtherCAT Slave Terminals

The same I/O refreshing operation as for Safety I/O Units is performed. This means that noise on the NX bus or EtherCAT line or other factors can result in exceeding the safety task period if a certain number of communications retries occurs. If that happens, increase the safety task period to increase noise immunity.

● Calculating the Safety Task Period

The calculation method depends on the communications setups of the Slave Terminals.

- 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 period (minimum safety task period) is displayed by the Sysmac Studio. Set the safety task period to a value that is within 100 ms of 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

  Use the following formula to calculate the minimum safety task period. Then set the safety task period to a value that is within 100 ms of the calculated minimum safety task period.

  Minimum safety task period = Minimum safety task period on Sysmac Studio display + 2 × EtherCAT communications cycle + 9.5 ms

5

Installation and Wiring

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

5-1 Installing Units

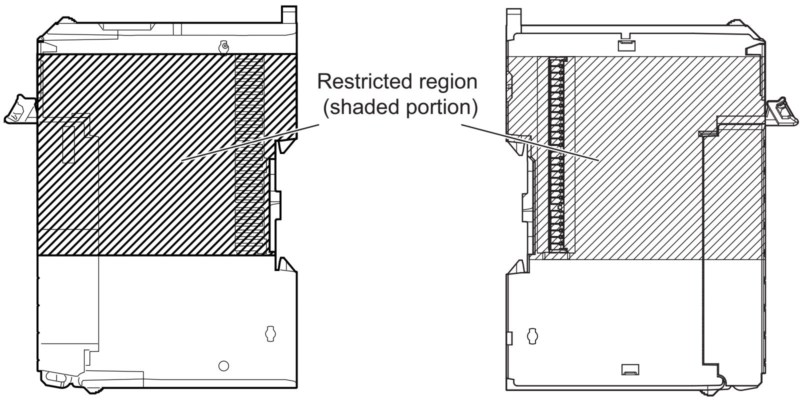
The NX-series Safety Control Units are installed in the same way as the NX Units. This section describes how to install and remove NX Units and how to attach markers.
Refer to the *EtherCAT Coupler Unit User's Manual* (Cat. No. W519) for information on preparations for installation and installation in a control panel.

5-1-1 Installing NX Units

Use the following procedure to mount NX Units to each other.
Always turn OFF the power supply before mounting any Unit. Always mount only one NX Unit at a time. If you try to mount multiple NX Units while they are connected to each other, the NX Units may separate from each other and fall.

Precautions for Safe Use

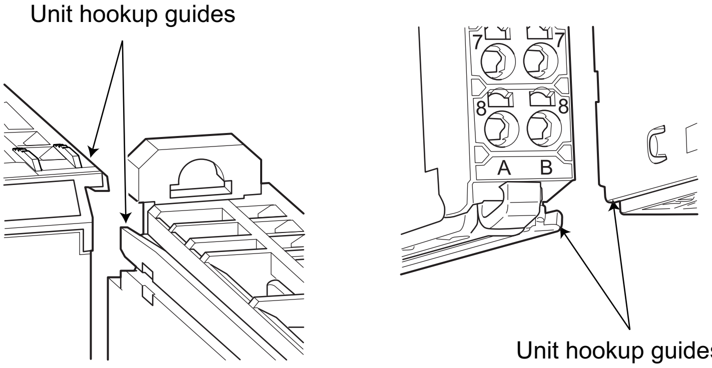
- Do not apply labels or tape to the NX Units. When the NX Unit is installed or removed, adhesive or scraps may adhere to the pins in the NX bus connector, which may result in malfunctions.
- Do not write anything with ink within the restricted region that is shown in the following figure. Also do not 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 Slave Terminal.



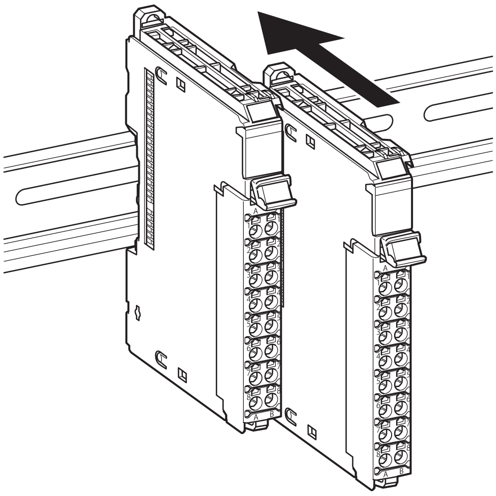
Precautions for Correct Use

- When you handle an NX Unit, be careful not to touch or bump the pins in the NX bus connector.
- When you handle an NX Unit, be careful not to apply 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 NX 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 release the DIN Track mounting hook on the NX Unit.

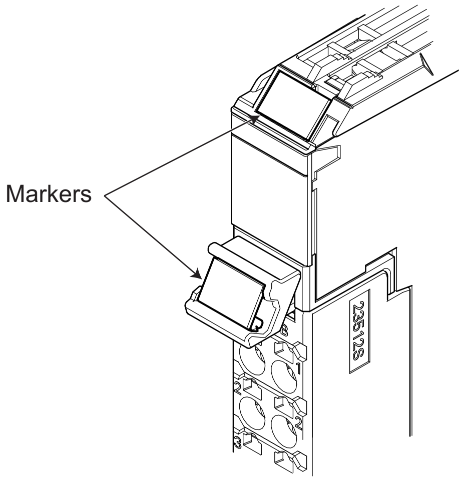
After you mount the NX Unit, make sure that it is locked to the DIN Track.

Additional Information

- When you mount the NX Unit, it is not normally necessary to release the DIN Track mounting hook on 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, unlock the DIN Track mounting hook at the start of the procedure, mount the NX Unit to the DIN Track, and then lock the DIN Track mounting hook.
- Refer to the Communications Coupler Unit user's manual for information on how to mount the Communications Coupler Unit and how to mount the NX Units after the Communications Coupler Unit.

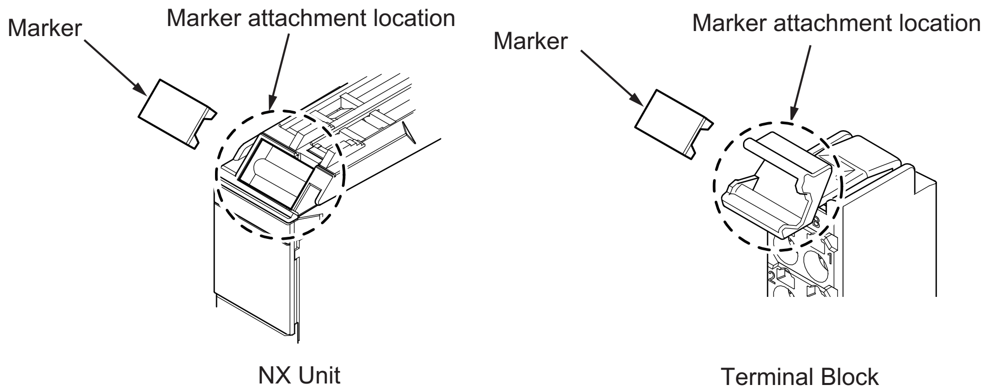
5-1-2 Attaching Markers

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 the terminal blocks on NX Units.



● 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.

Type	Model number	
	Manufactured by Phoenix Contact	Manufactured by Weidmueller
Markers	UC1-TMF8	DEK 5/8
Special marker printer	UM EN BLUEMARK X1	PrintJet PRO

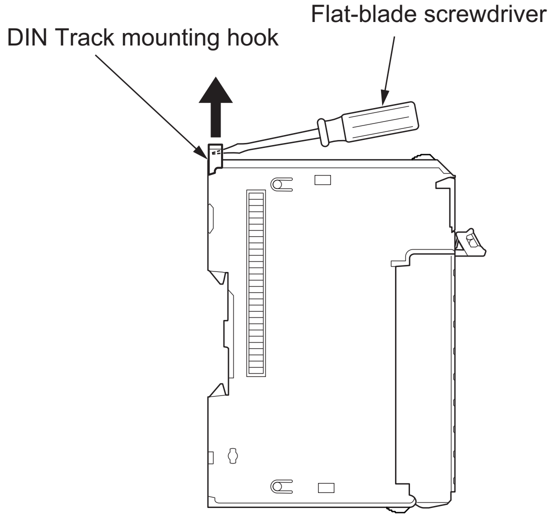
The markers made by OMRON cannot be printed on with commercially available marker printers.

5-1-3 Removing Units

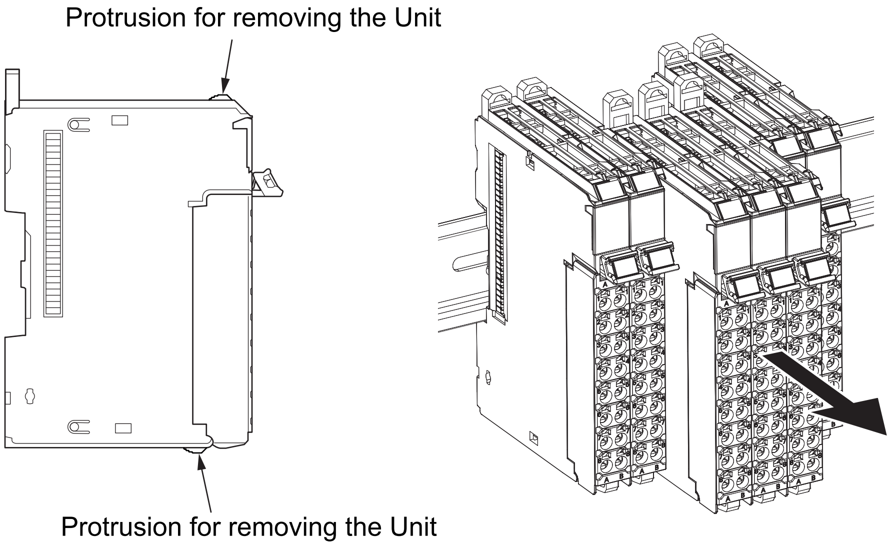
Precautions for Safe Use

Always turn OFF the Unit power supply and I/O power supply before removing any 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.

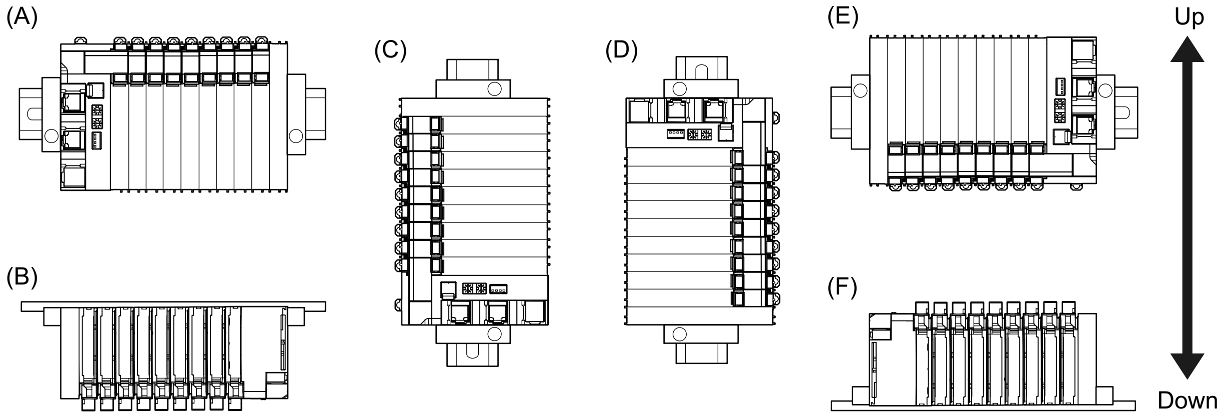


Precautions for Correct Use

- 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.

5-1-4 Installation Orientation

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 the user's manuals for the Communications Coupler Unit, NX Units, and NX-series System Units that you will use.

Precautions for Safe Use

For installation orientations (C) and (D) in the above 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.

5-2 Wiring the Power Supply to the Slave Terminal

This section describes how to supply power to a Slave Terminal and how to wire the power supplies.

5-2-1 Power Supply Types

There are the following two types of power supplies that supply power to the Slave Terminal.

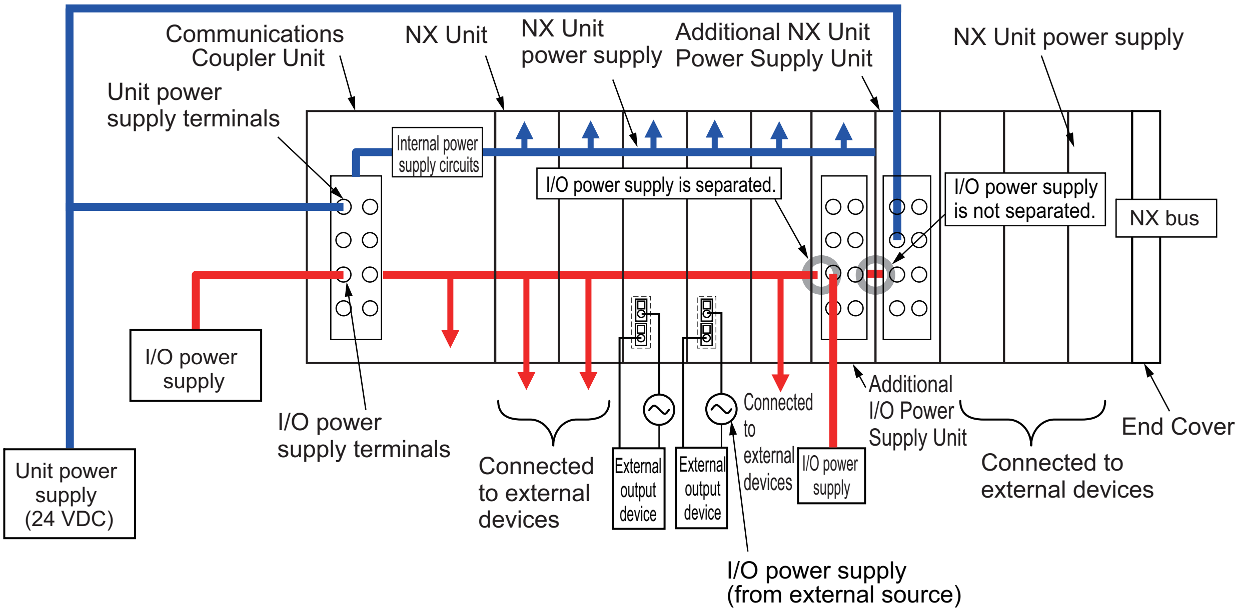
Power supply type	Description
Unit power supply	This power supply is required to generate the NX Unit power supply, which is necessary for the Slave Terminal to operate. This power supply is connected to the Unit power supply terminals on the Communications Coupler Unit or Additional NX Unit Power Supply Units. The internal power supply circuits of the Communications Coupler Unit and Additional NX Unit Power Supply Units generate the NX Unit power supply from the Unit power supply. The internal circuits of the Communications Coupler Unit and of the NX Units operate on the NX Unit power supply. The NX Unit power is supplied to the NX Units in the Slave Terminal through the NX bus connectors.
I/O power supply	This power supply drives the internal I/O circuits of the I/O Units and it is also used for the connected external devices. This power supply is connected to the I/O power supply terminals on the Communications Coupler Unit or Additional NX Unit Power Supply Units. This power supply is used for the following items. • Operation of I/O circuits of the Safety I/O Units • Input current to the Safety Input Units • Load current of external loads of Safety Output Units • Power for connected external devices The I/O power is supplied to the I/O Units through the NX bus connectors.

5-2-2 Power Supply Methods and Wiring

Power is supplied to the Safety Units as described in the following table.

Power supply type	Description
Unit power supply	When the Unit power supply is connected to the Unit power supply terminals on the Communications Coupler Unit or an Additional NX Unit Power Supply Unit, power is supplied to the Position Interface Unit through the NX bus connector.
I/O power supply	Power is supplied with the following methods. Refer to *3-2 Safety I/O Units* on page 3-6 for the power supply method for each model of I/O Unit. • Supplied from the NX bus. 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 an Additional I/O Power Supply Unit. • Supplied from external source Power is supplied to the Units from an external source. I/O power is supplied by connecting an I/O power supply to the I/O power supply terminals on the Units.

Example wiring diagrams are provided in the following figures for the different types of power supplies.



Precautions for Correct Use

Supply power from separate power supplies for the Unit power supply and the I/O power supply. If you supply power from the same power supply, noise may cause malfunctions.

Additional Information

Refer to the *EtherCAT Coupler Unit User's Manual* (Cat. No. W519) for information on the power supply system design for Slave Terminals.

5-2-3 Calculating the Total Current Consumption from the I/O Power Supply

The total current consumption from the I/O power supply from the NX bus must be less than the maximum I/O power supply current of the Communications Coupler Unit or Additional I/O Power Supply Unit. 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 I/O power supply current consumption that is given for each model in the datasheets.

Calculate the total current consumption from the I/O power supply for the Safety I/O Units 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

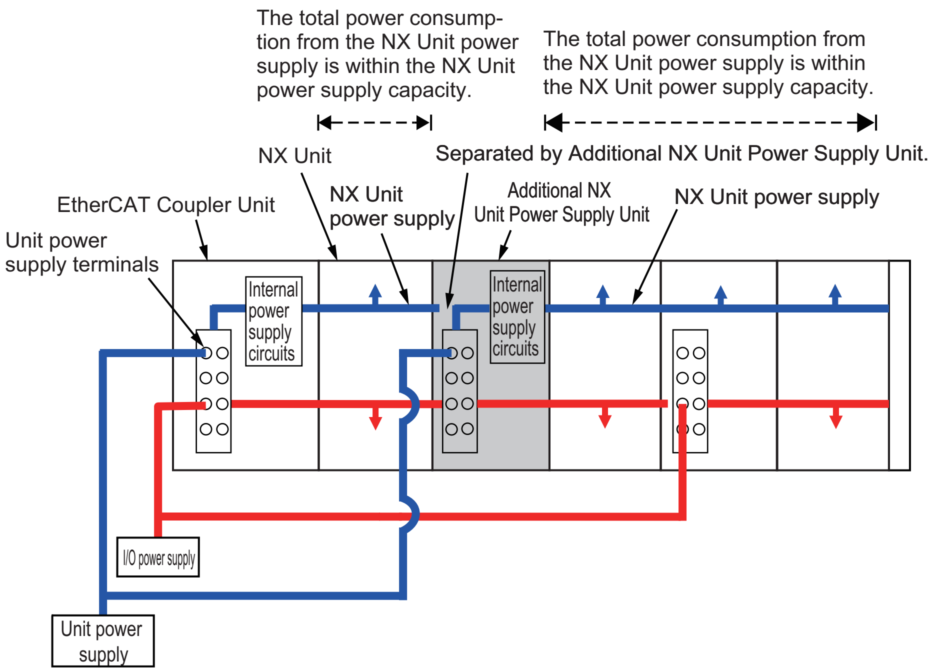
= (Current consumption from I/O power supply of Safety Output Unit) + (Total connected load current) + (Total current consumption of connected external devices)

Refer to *2-2 Specifications of Individual Units* on page 2-3 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.

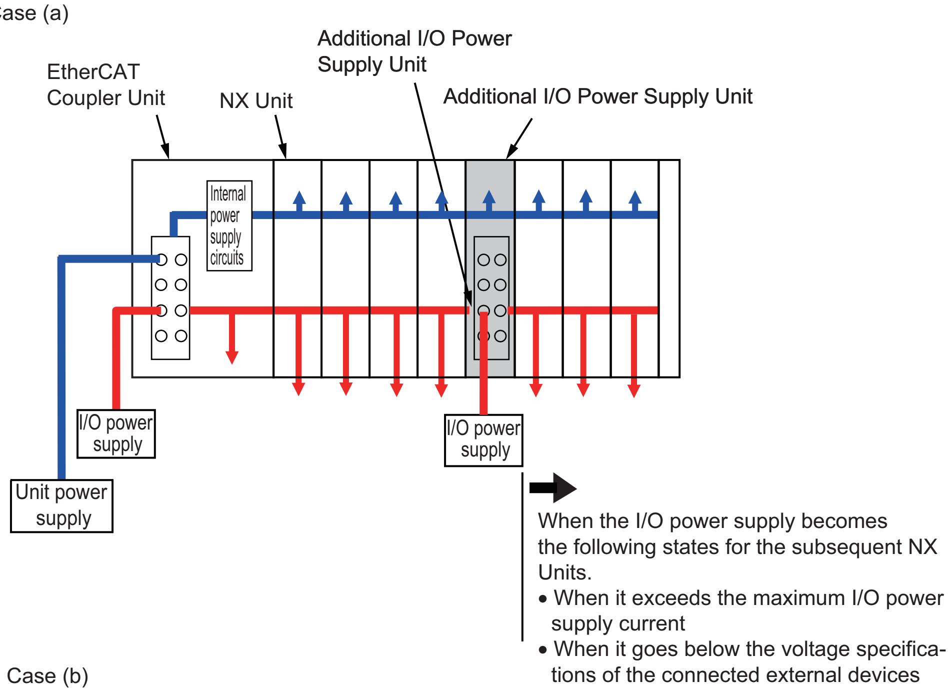
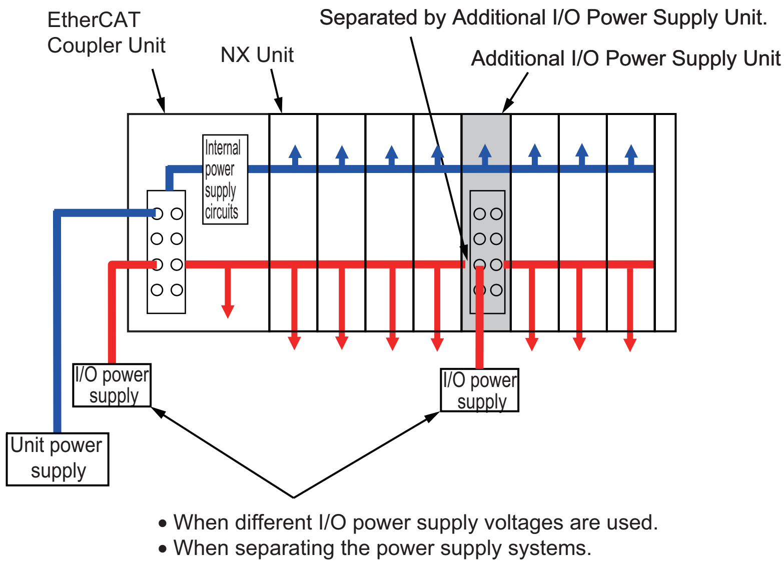
5-2-4 NX-series Power Supply-related Units

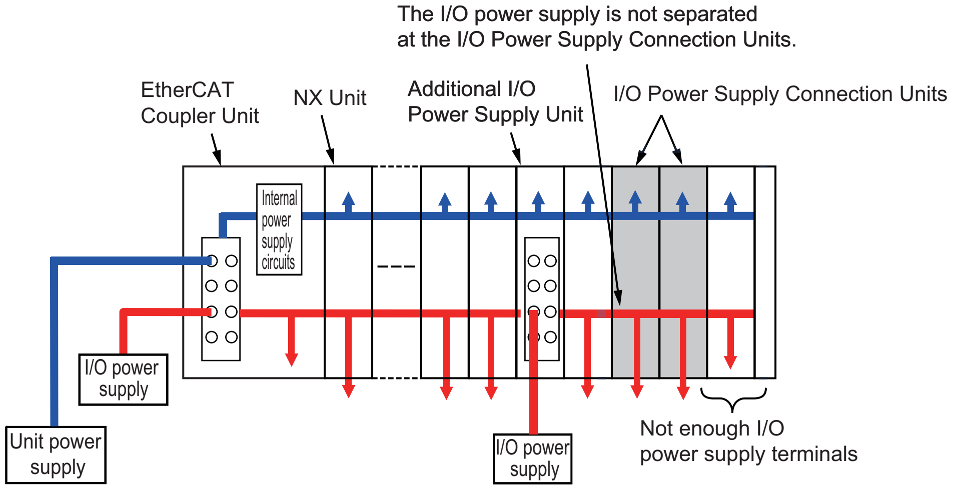
The Communications Coupler Unit supplies the NX Unit power and I/O power to the NX Units in the Slave Terminal. The Units that are related to power supply for the NX Series other than the Communications Coupler Unit are listed in the following table.

For the details on the NX-series power supply-related Units, refer to the *NX-series System Units User's Manual* (Cat. No. W523). For information on the most recent lineup of NX Units, refer to NX-series catalogs or OMRON websites, or ask your OMRON representative.

<table>
<tr><th>Name</th><th>Function</th></tr>
<tr><td>Additional NX Unit Power Supply Unit</td><td>This NX Unit provides the NX Unit power supply.
This NX Unit is used when the total power consumption of the NX Units in the Slave Terminal exceeds the NX Unit power supply capacity of the Communications Coupler Unit.


The I/O power supply for the Additional NX Unit Power Supply Unit is connected to the NX Unit on the left through the NX bus connector.</td></tr>
</table>

Name	Function
Additional I/O Power Supply Unit	This NX Unit provides additional I/O power supply. Use this NX Unit in the following cases. (a) When the I/O power supply capacity is insufficient • When the total current consumption for the I/O power supply exceeds the maximum I/O power supply current of the EtherCAT Coupler Unit • When a voltage drop in the I/O power supply causes the voltage of the I/O power supply to go below the voltage specifications of the I/O circuits or connected external devices (b) Separating the I/O Power Supply • When connected external devices have different I/O power supply voltages • When separating the power supply systems Case (a)  Case (b) 

Name	Function
I/O Power Supply Connection Units	This Unit is used when there are not enough I/O power supply terminals for the external devices that are connected to NX Units such as Safety I/O Units and Analog I/O Units. 

5-3 Wiring the Terminals

This section describes how to wire the terminals on the Safety I/O Units.

⚠ WARNING

Make sure that the voltages and currents that are input to the NX Units and Slave Terminals are within the specified ranges.

Inputting voltages or currents that are outside of the specified ranges may cause accidents or fire.

5-3-1 Wiring to the Screwless Clamping Terminal Block

This section describes wiring the screwless clamping terminal blocks, terminal block mounting and removal methods, and prevention of incorrect attachment.
You can connect ferrules that are attached to twisted wires to the screwless clamping terminal block.
You can also connect twisted wires or 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 following terminals are wired.

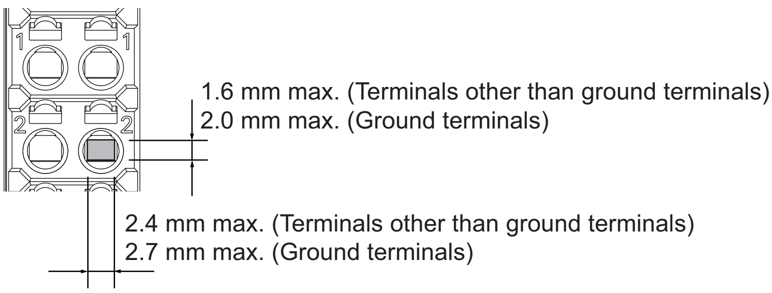
- I/O power supply terminals
- I/O terminals

Applicable Wires

The wires that you can connect to the screwless clamping terminal block are twisted wires, solid wires, and ferrules that are attached to twisted wires. This section gives the dimensions and processing methods for applicable wires.

● 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 as specified in the following description to apply the dimensions.



Using Ferrules

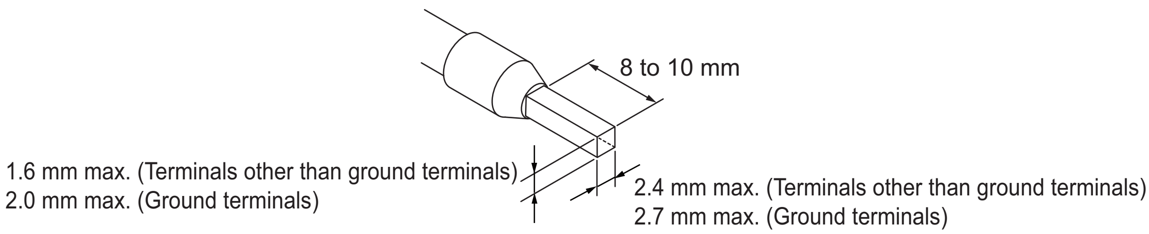
If you use ferrules, attach them to twisted wires.
Observe the application instructions for your ferrules for the wire stripping length when attaching ferrules.
Always use one-pin ferrules. Do not use two-pin ferrules.

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

Terminal type	Manufacturer	Ferrule model	Applicable wire (mm^2 (AWG))	Crimping tool
Terminals other than ground terminals	Phoenix Contact	AI0,34-8	0.34 (#22)	Phoenix Contact (The figure in parentheses is the applicable wire size.) CRIMPFOX 6 (0.25 to 6 mm^2, AWG 24 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	
Terminals other than ground terminals	Weidmueller	H0.14/12	0.14 (#26)	Weidmueller (The figure in parentheses is the applicable wire size.) PZ6 Roto (0.14 to 6 mm^2, AWG 26 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. Some AWG14 wires exceed 2.0 mm^2 and cannot be used in the screwless clamping terminal block.

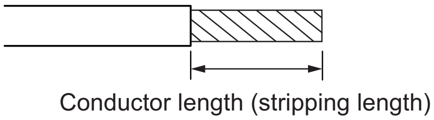
When you use any ferrules other than those in the above table, crimp them to the twisted wires so that the following processed dimensions are achieved.



Using Twisted or Solid Wires

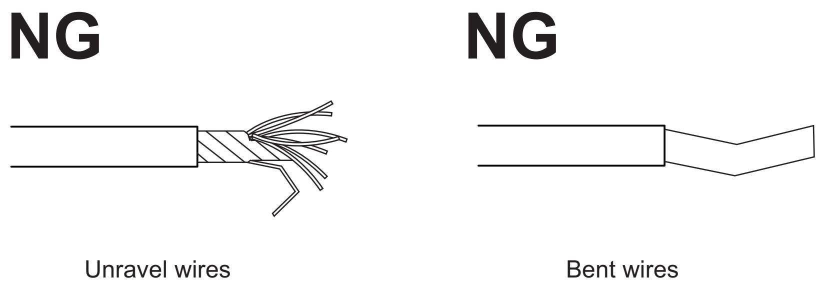
If you use the twisted wires or the solid wires, the applicable wire range and conductor length (stripping length) are as follows.
Use the twisted wires to connect the ground wire to a ground of 100 Ω or less. Do not use the solid wires.

Terminal type	Applicable wires	Conductor length (stripping length)
Ground terminals	2.0 mm^2	9 to 10 mm
Terminals other than ground terminals	0.08 to 1.5 mm^2 AWG28 to 16	8 to 10 mm



Precautions for Correct Use

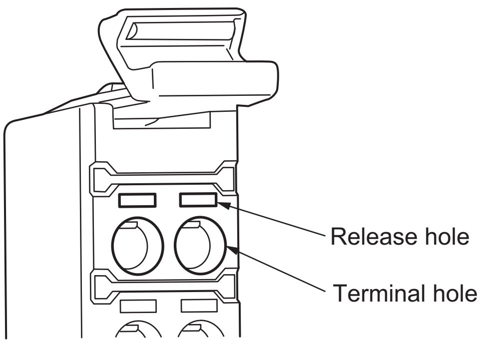
- Use cables with suitable wire sizes for the carrying current. There are also restrictions on the current due to the ambient temperature. Refer to the manuals for the cables and use the cables correctly for the operating environment.
- For twisted wires, strip the sheath and twist the conductor portion.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



Connecting/Removing Wires

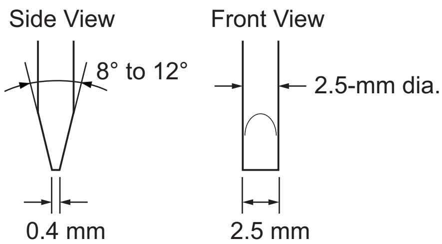
This section describes how to connect and remove wires.

● Terminal Block Parts and Names



● Required Tools

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

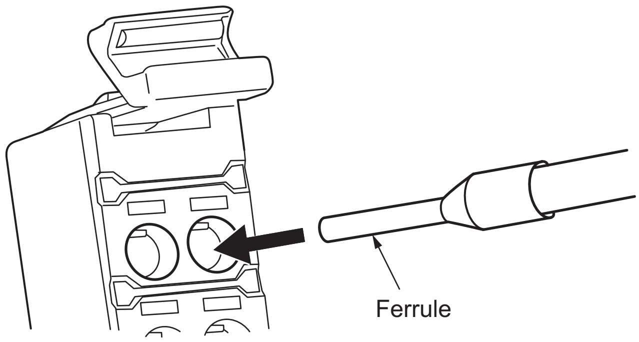


Recommended Screwdriver

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

Connecting Ferrules

Insert the ferrule straight into the terminal hole.
It is not necessary to press a flat-blade screwdriver into the release hole.



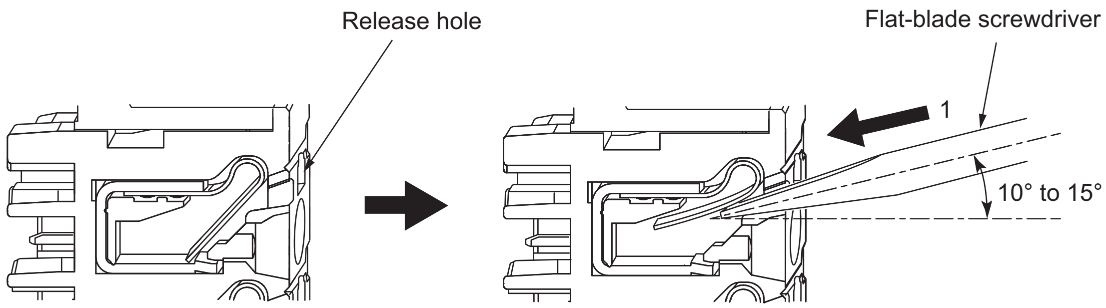
After you make a connection, make sure that the ferrule is securely connected to the terminal block.

Connecting Twisted Wires/Solid Wires

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

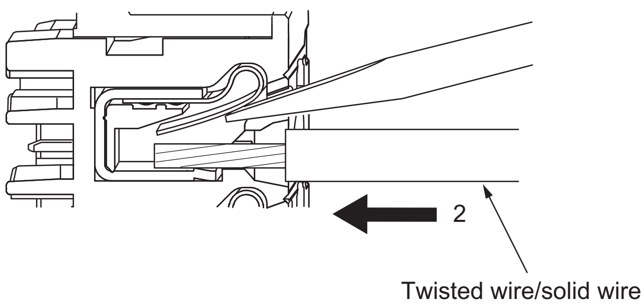
1 Press the flat-blade screwdriver diagonally into the release hole.

Press at an angle of 10° to 15°. If you press in the screwdriver correctly, you will feel the spring in the release hole.

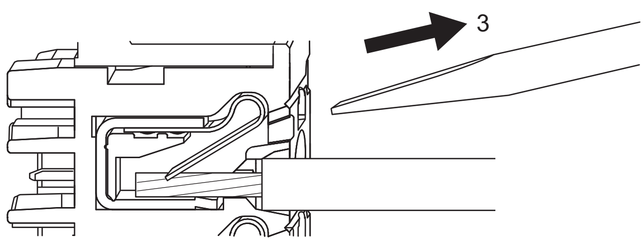


2 Leave the flat-blade screwdriver pressed into the release hole and insert the twisted wire or the solid wire into the terminal hole.

Insert the twisted wire or the solid wire until the stripped portion is no longer visible to prevent shorting.



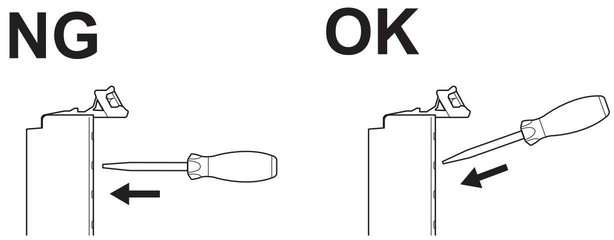
3 Remove the flat-blade screwdriver from the release hole.



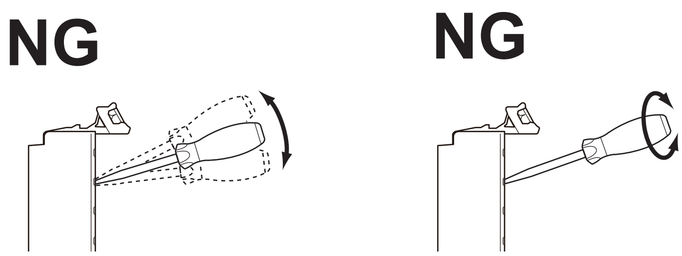
After you make a connection, make sure that the twisted wire or the solid wire is securely connected to the terminal block.

Precautions for Correct Use

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



- 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 damage the terminal block.



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

● Removing Wires

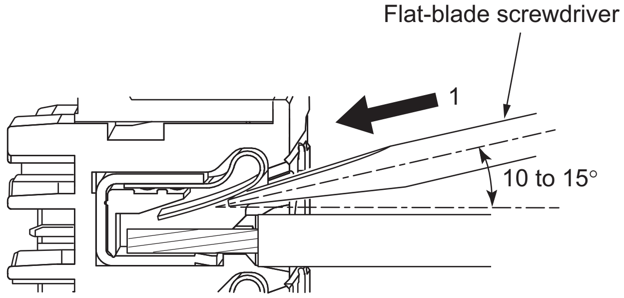
Use the following procedure to remove the wires from the terminal block.

The removal process is the same for both ferrules and twisted/solid wires.

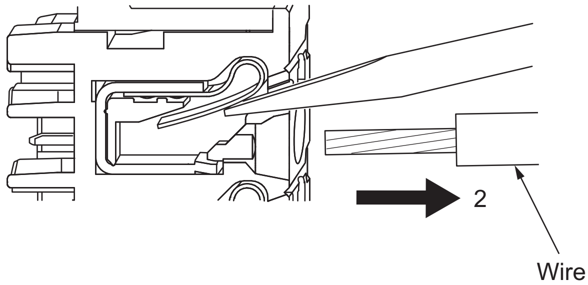
1 Press the flat-blade screwdriver diagonally into the release hole.

Press at an angle of 10° to 15°.

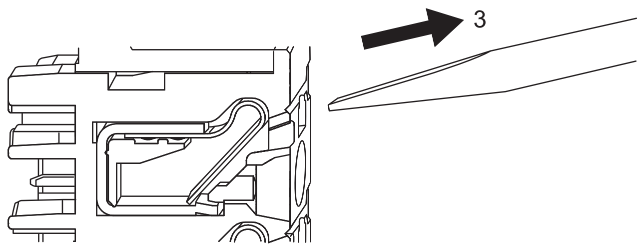
If you press in the screwdriver correctly, you will feel the spring in the release hole.



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

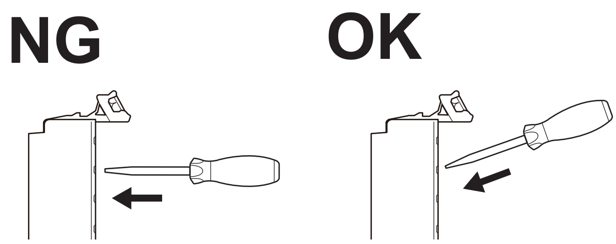


3 Remove the flat-blade screwdriver from the release hole.



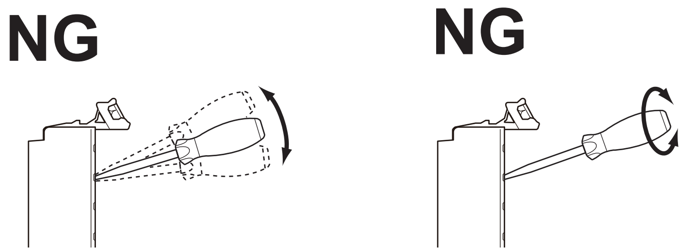
Precautions for Safe Use

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



- 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.

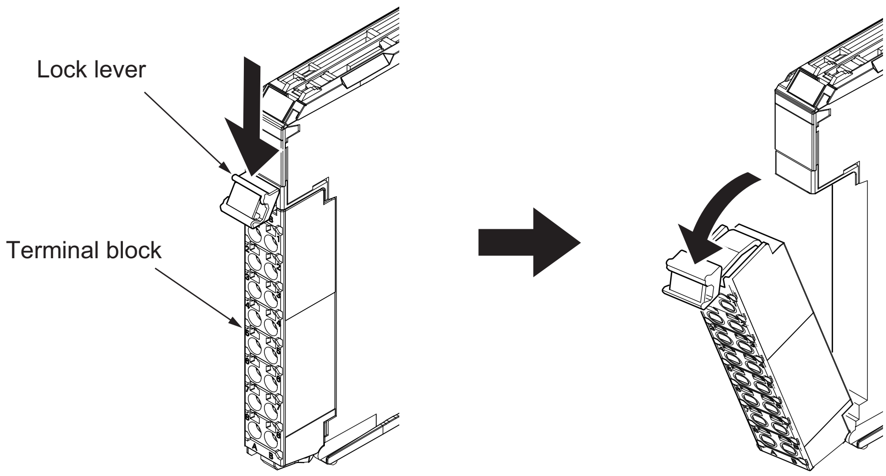
- Do not tilt or twist the flat-blade screwdriver while it is pressed into the release hole. Doing so may break the terminal block.



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

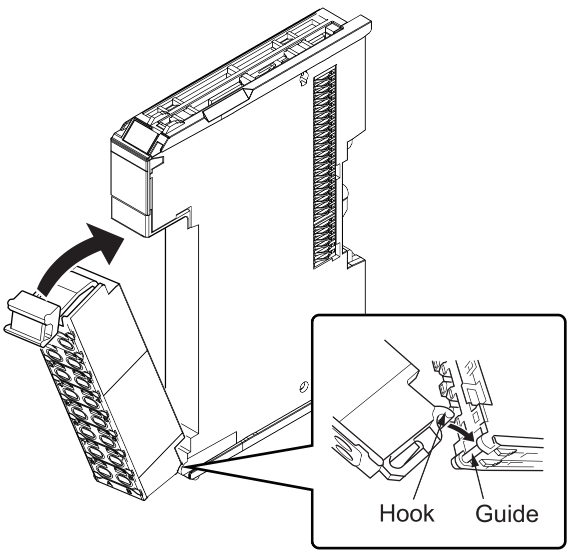
Removing a Terminal Block

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



Attaching a Terminal Block

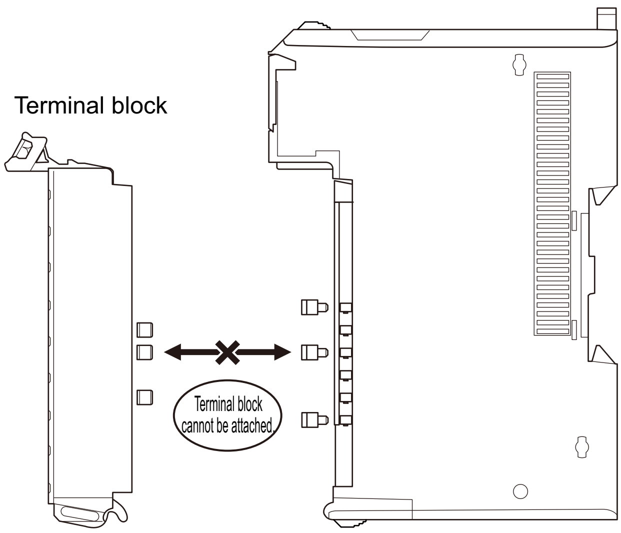
Place the terminal block hook on the guide at the bottom of the NX Unit and press in on the top of the terminal block to attach it.



Terminal blocks for Safety I/O Units come in two types depending on the number of terminals that can be used. There are 8-terminal and 16-terminal blocks. You can use only one of the two types of terminal blocks given above with a Unit that has a screwless clamping terminal block. The terminal block must have the same number of terminals that the Unit is designed for.

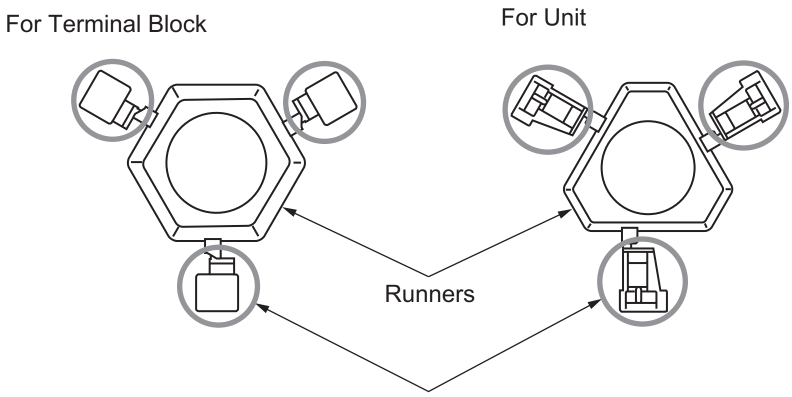
Preventing Incorrect Attachment of Terminal Blocks

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 combinations 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.

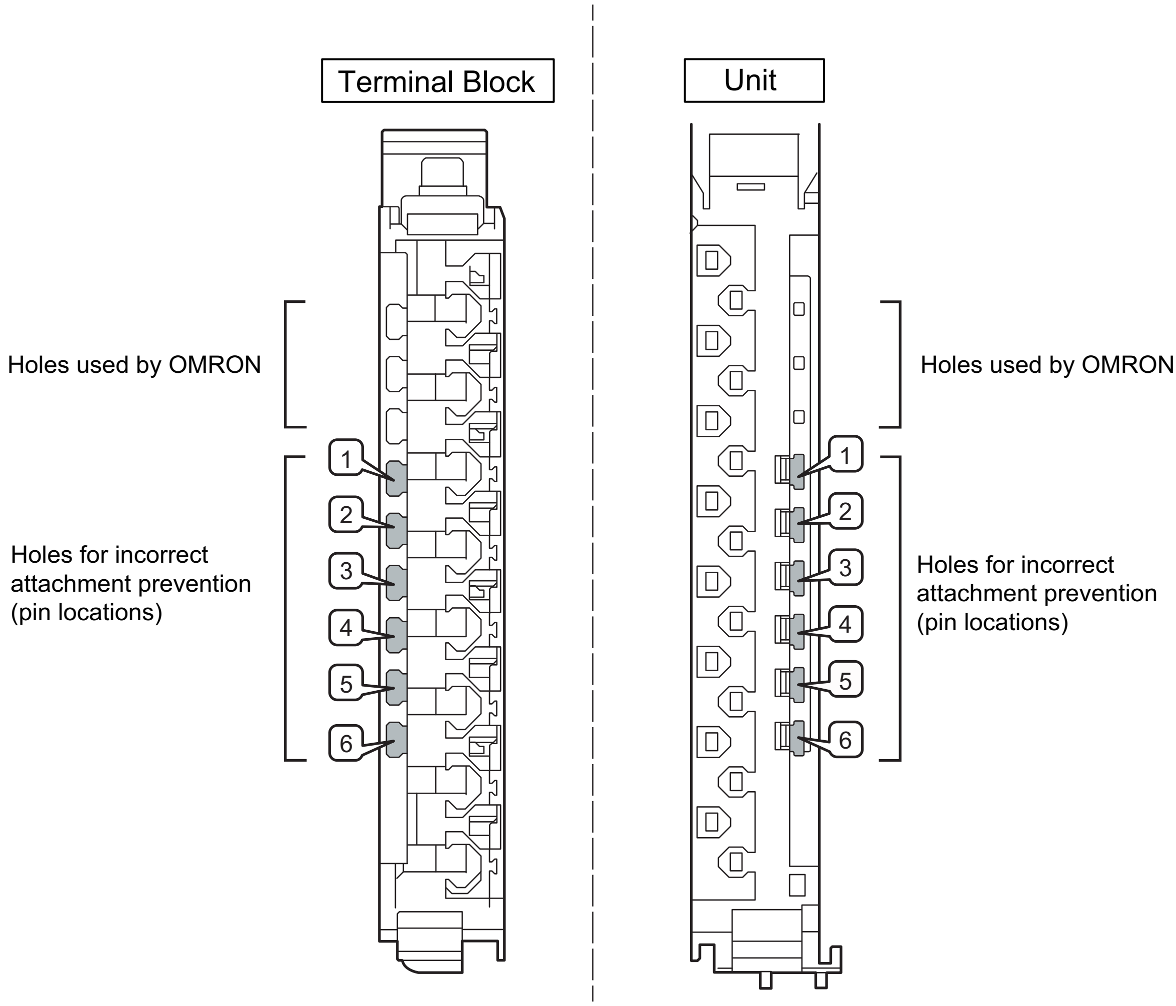


Use the following Coding Pins.

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				○	○	○	○	○	○			

To make the maximum of 20 patterns, purchase two sets of NX-AUX02 Pins. (One set for 10 Units.)

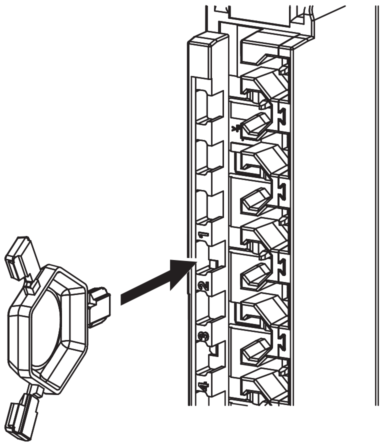
Precautions for Correct Use

- OMRON uses the holes other than No. 1 to 6 in the above figure. If you insert a Coding Pin into one of the holes used by OMRON on the terminal block side, it is impossible to mount the terminal block on a Unit.
- Do not use Coding Pins that have been attached and then removed.

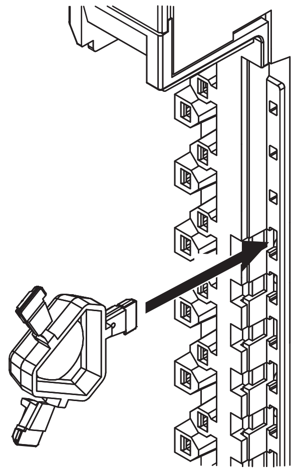
● Inserting the Coding Pins

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

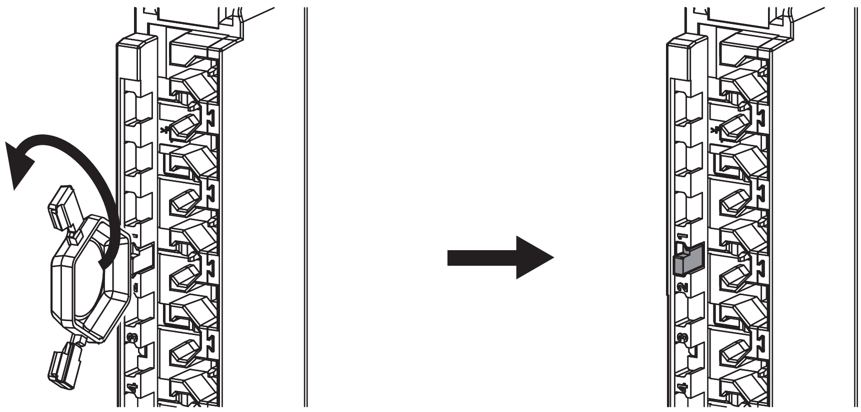


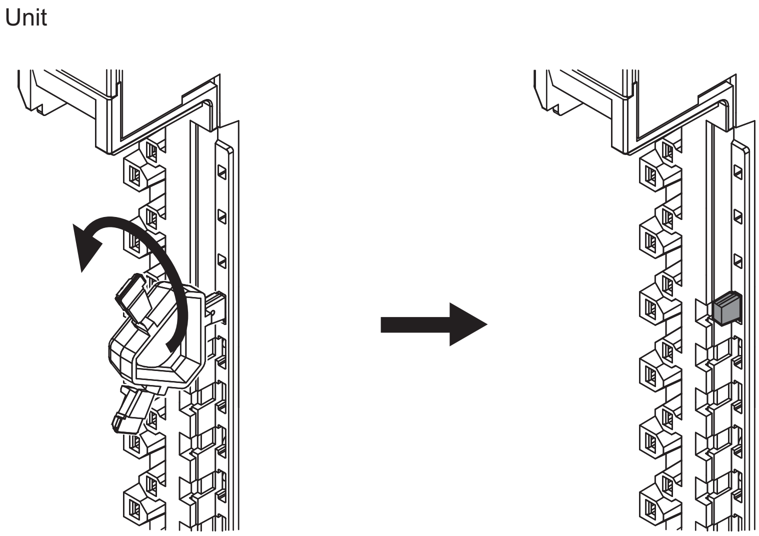
Unit



2 Rotate the runner to break off the Coding Pin.

Terminal Block





5-3-2 Checking Wiring

You can display the I/O Map or Watch Tab Page on the Sysmac Studio to check the 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 output 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 conveniently monitor status or perform forced refreshing without defining variables or creating an algorithm to check the wiring.

Refer to *Section 8 Checking Operation and Actual Operation* for the monitoring and forced refreshing procedures.

6

System Configuration and Setup

This section describes how to use the Sysmac Studio to configure and set up the safety control system.

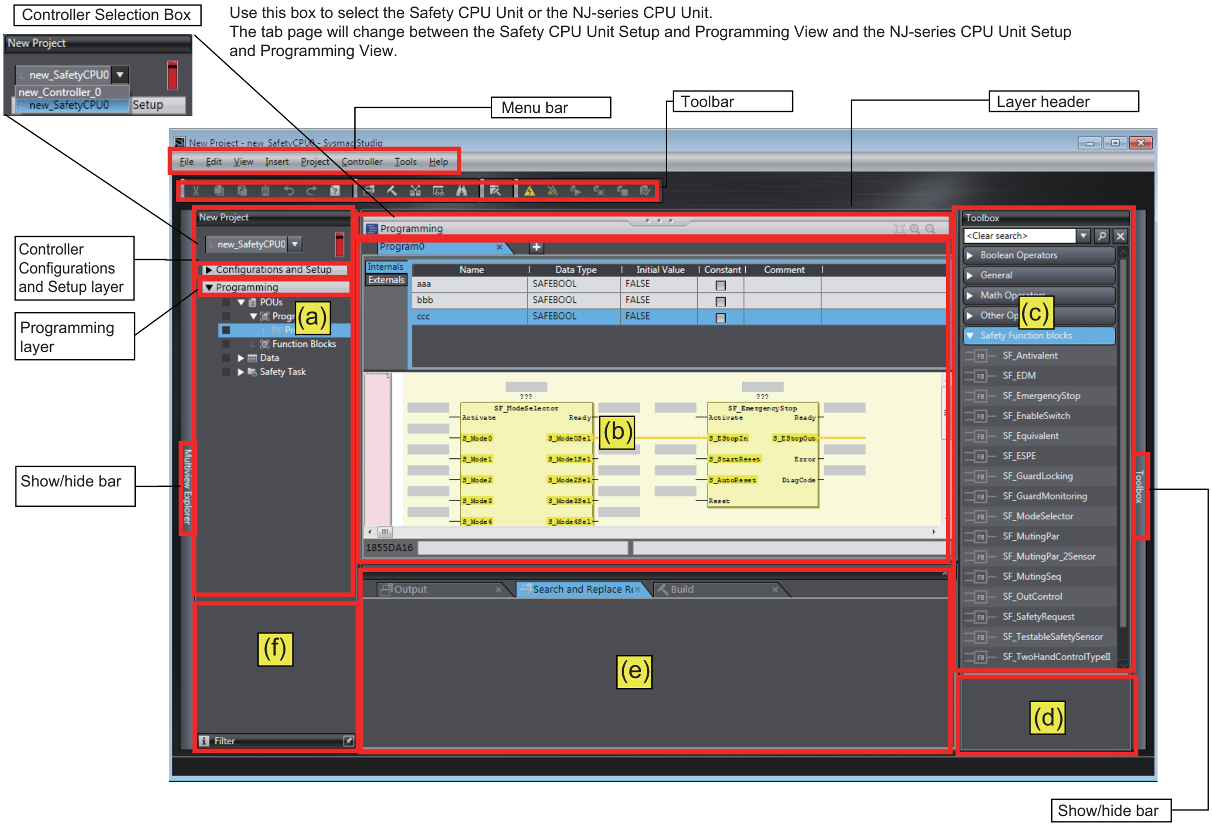
6-1 Configuration and Setup Procedures

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 EtherCAT network and EtherCAT Slave Terminal for use with the NJ-series System.
2. Configure and set up the safety network (FSoE).
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 NJ-series CPU Unit.

6-2 Part Names and Functions of the Sysmac Studio Window

This section gives the names of the parts of the Sysmac Studio Window.



No.	Name	Function
(a)	Multiview Explorer	This pane is your access point for all Sysmac Studio data that is related to the Safety CPU Unit. It has a Controller Selection Box, and is separated into a Configurations and Setup Layer and a Programming Layer. Use the Controller Selection Box to select the Safety CPU Unit or NJ-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.
(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.
(e)	Cross Reference Tab Page	The Cross Reference Tab Page displays a list of where variables, data types, I/O ports, function blocks, and function instructions are used.
	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.

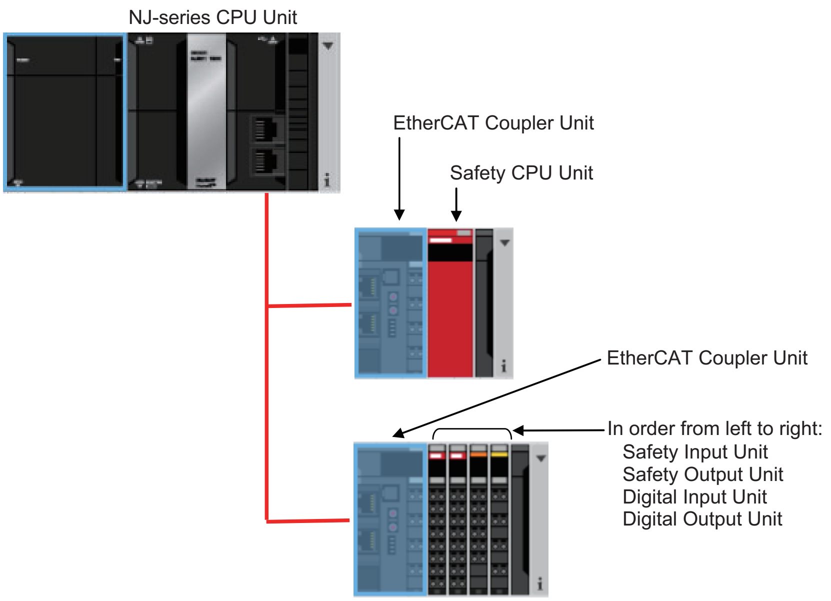
This manual describes only the functions and operations of the Sysmac Studio that are related to the safety control system. Refer to the *Sysmac Studio Version 1 Operation Manual* (Cat. No. W504) for details on Sysmac Studio operation.

6-3 Configuration and Setup of the EtherCAT Network and EtherCAT Slave Terminal

You configure and set up the EtherCAT network and EtherCAT Slave Terminals where the Safety Control Units are mounted as part of the Controller Configuration and Setup of the NJ-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-series CPU Unit.

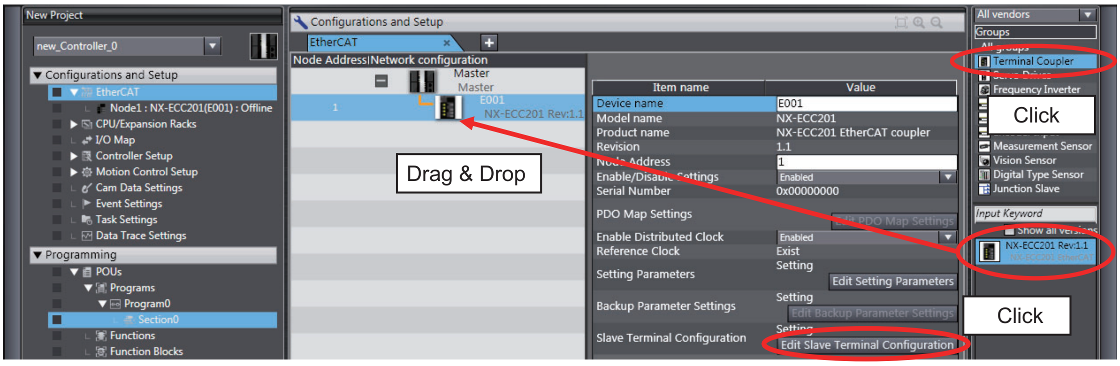
6-3-1 Procedures for Creating the Controller Configuration for Safety Control

Use the following procedure to create the Controller configuration for the Safety Control Units.

1. Start the Sysmac Studio.
2. Select the model and version of the NJ-series CPU Unit, and create a project file.
3. Double-click **EtherCAT** under **Configurations and Setup** in the Multiview Explorer.
   The EtherCAT Tab Page is displayed.

4 Select **Terminal Coupler** from the Groups List in the Toolbox.

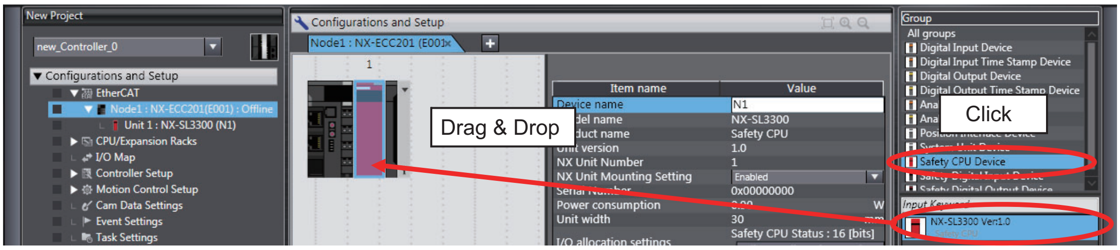
The EtherCAT Coupler Unit is displayed below it. Drag the Unit and add it to the configuration.



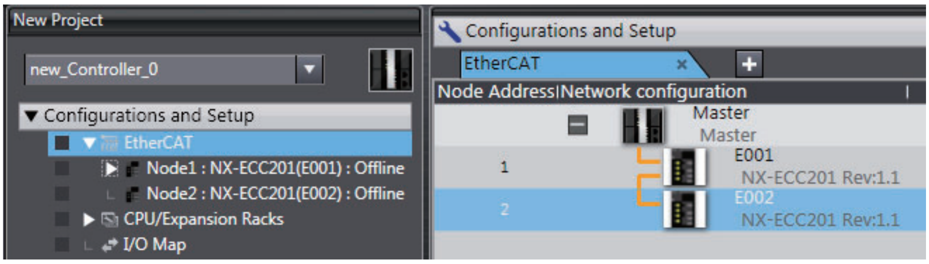
5 Click the **Edit Slave Terminal Configuration** Button at the bottom of the list of EtherCAT Coupler Unit settings. Or, right-click the EtherCAT Coupler Unit and select ***Edit Slave Terminal Configuration*** from the menu.

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 Unit to the Slave Terminal and add it to the configuration.

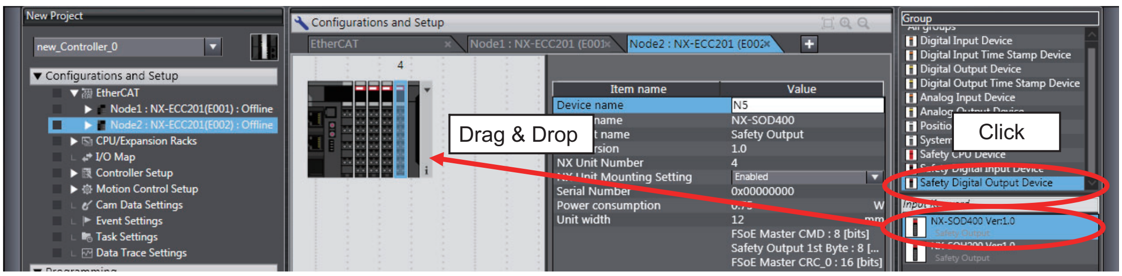


7 Perform steps 3 and 4 to add another EtherCAT Coupler Unit.

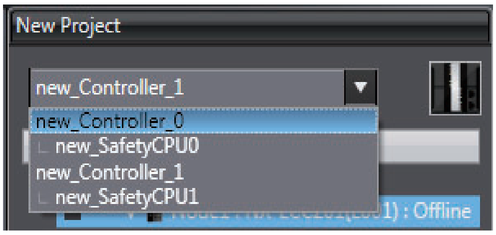


8 Or, perform step 5 for the EtherCAT Coupler Unit that was added to display the Slave Terminal Tab Page.

9 Select **Safety Digital Input Device** or **Safety Digital Output Device** in the Group Field of the Toolbox. The Safety I/O Unit is displayed below it. Drag the Unit to the Slave Terminal and place it in the configuration.



This completes the creation of the Controller configuration for an NJ-series CPU Unit that includes Safety Control Units. After the Safety CPU Unit is added to the NJ-series Controller configuration, it will be displayed in the Controller Selection Box in the Multiview Explorer. The Safety CPU Unit that was added is displayed below the NJ-series Controller (i.e., the EtherCAT master).



Additional Information

- Only one Safety CPU Unit can be placed on the EtherCAT network. If you add more than one Safety CPU Unit, the icon is displayed under all of the Safety CPU Units and it will cause an error during the build process.
- Refer to the *NX-series EtherCAT Coupler Unit User's Manual* (Cat. No. W519-E1-02 or later) for the number of Safety I/O Units that can be mounted to a Slave Terminal.

6-3-2 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 Units") as necessary.

You can change the device names of registered Safety Control Units, 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. The device name is automatically assigned when you register the Safety CPU Unit or Safety I/O Unit. You can change the device name if necessary. 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.	This is the model number of the Safety Control Unit.	---	---
Product name	Not possible.	This is the product name.	---	---
Unit version	Not possible.	This is the 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 left-most mounting position.	---	---
NX Unit mounting setting	Possible.	This setting enables or disables I/O refreshing for the Safety Control Unit. Refer to the *NX-series EtherCAT Coupler Unit User's Manual* (Cat. No. W519) for details on this function.	Enabled or Disabled	Enabled

Item	Editing	Description	Data range	Default
Serial number	Not possible.	This is the 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
Power consumption [W]	Not possible.	This is the power consumption of the Safety Control Unit from the NX bus. 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.	This is the width of the Safety Control Unit.	---	Depends on the model of the Unit.
I/O allocation settings	Not possible.	These are the I/O allocation settings of the Safety Control Unit.	---	---
Unit operation settings	Not possible.	There are no settings that you can edit for the Safety Control Units.	---	---

6-4 Setting Up the Safety Process Data Communications

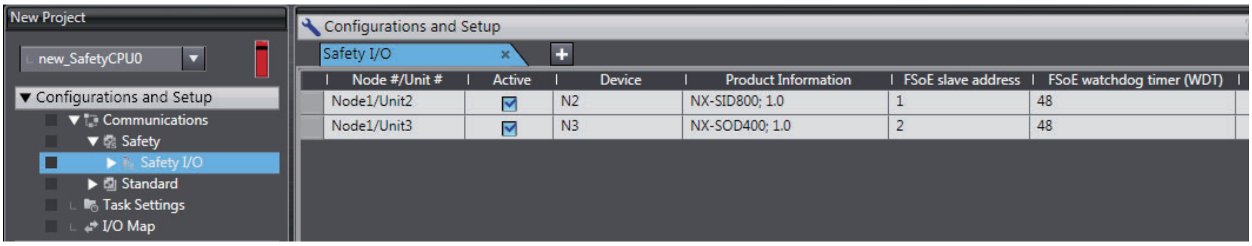
When you add a Safety Control Unit to an EtherCAT Slave Terminal configuration on the Sysmac Studio, the safety process data communications are set up automatically.

Use the following procedure to view or change the settings for the safety process data communications.

1 In the Multiview Explorer, select the 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 Tab Page is displayed.

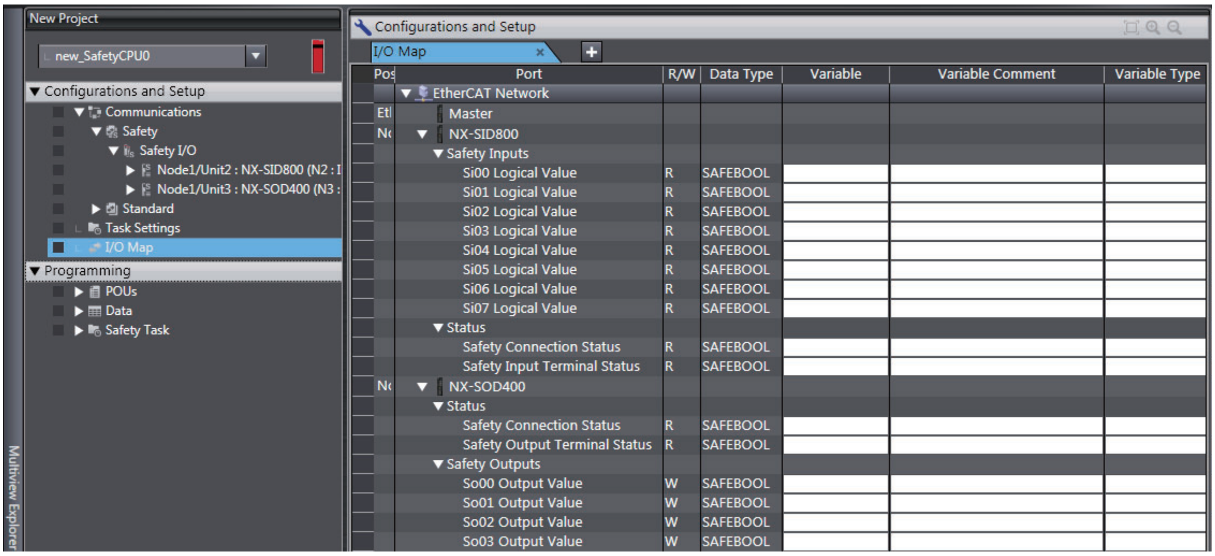


The meanings of the items in the Safety I/O Tab Page are given below.

Item	Editing	Description	Cases when editing is necessary
Node #/Unit #	Not possible.	This is the node number as an EtherCAT slave and the NX Unit number for the Safety I/O Unit.	---
Active (Enable/ Disable)	Possible.	This setting enables or disables safety process data communications. Selected: *Enabled*. 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. Not selected: *Disabled*. 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.	Set this to *Disabled* to restrict the selection of a Safety I/O Unit when the actual Unit is not ready for debugging yet.
Device	Not possible.	This is the name of the Safety Unit.	---
Product Information	Not possible.	This is the model and version of the Unit.	---
FSoE Slave Address	Not 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 safety process data communications.	---
FSoE Watchdog Timer	Not possible.	The safety I/O refresh times are automatically set. The settings cannot be changed.	---

If you are not using safety process data communications with this Safety I/O Unit, clear the selection for the *Active* Check Box. To change the FSoE slave address, set the value in the *FSA* cell.

3 Double-click **I/O Map** to display the I/O ports for the Safety I/O Units that are set to *Enabled* in the *Active* Check Box.



Precautions for Safe Use

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.

6-5 Setting the Safety Input and Output Functions

You set the safety input functions and safety output functions of the Safety I/O Unit when you assign input devices and output devices to the Safety I/O Unit with the Sysmac Studio.

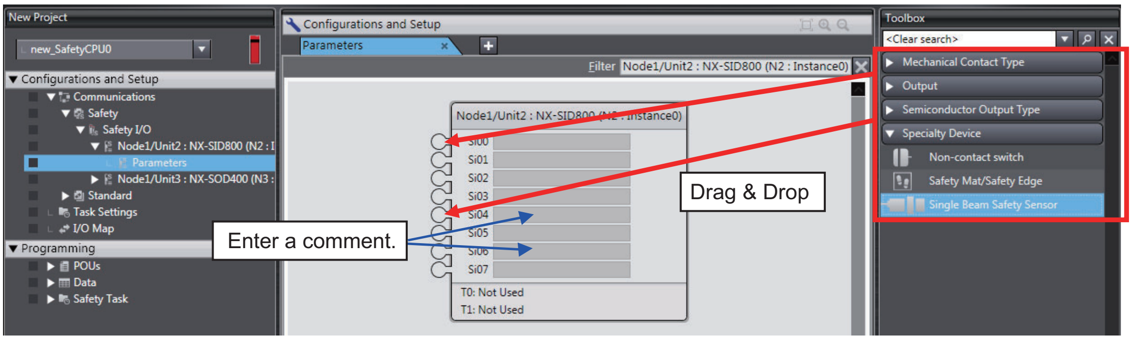
This section describes how to assign devices that are connected. Refer to *3-3-1 Safety Input Functions* on page 3-10 and *3-3-2 Safety Output Functions* on page 3-27 for details on the safety input functions and safety output functions.

1 In the Multiview Explorer, select the Safety CPU Unit in the Controller Selection Box.

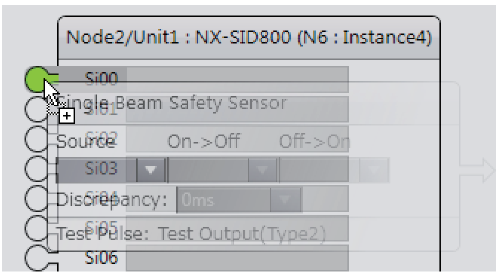
2 Double-click **Parameters** under the name of the Safety I/O Unit under **Configurations and Setup − Communications − Safety − Safety I/O**.

The Parameters Tab Page is displayed.

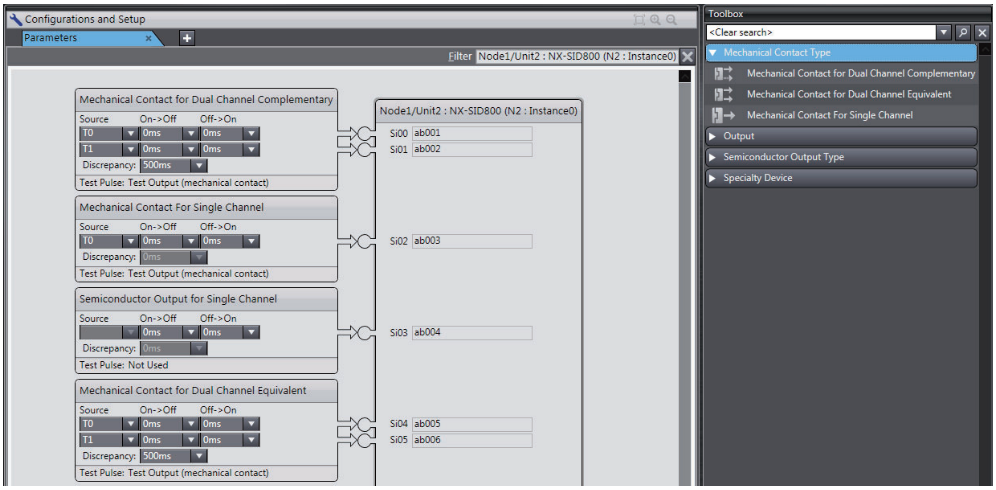
3 Select a device from the Toolbox to connect to the safety input terminal or safety output terminal of the Safety I/O Unit, and drag it to the desired I/O terminal.



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.



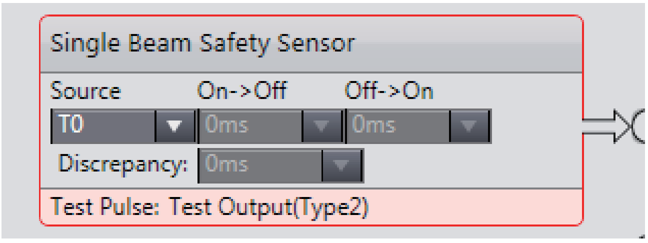
4 When you complete the settings, the following is displayed. Change the settings and enter comments.



Refer to *3-3-1 Safety Input Functions* on page 3-10 and *3-3-2 Safety Output Functions* on page 3-27 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

Comments are used as memos on this Unit setup tab page.

6-6 Registering Device Variables

Device variables are used to access data in 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 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 Unit.

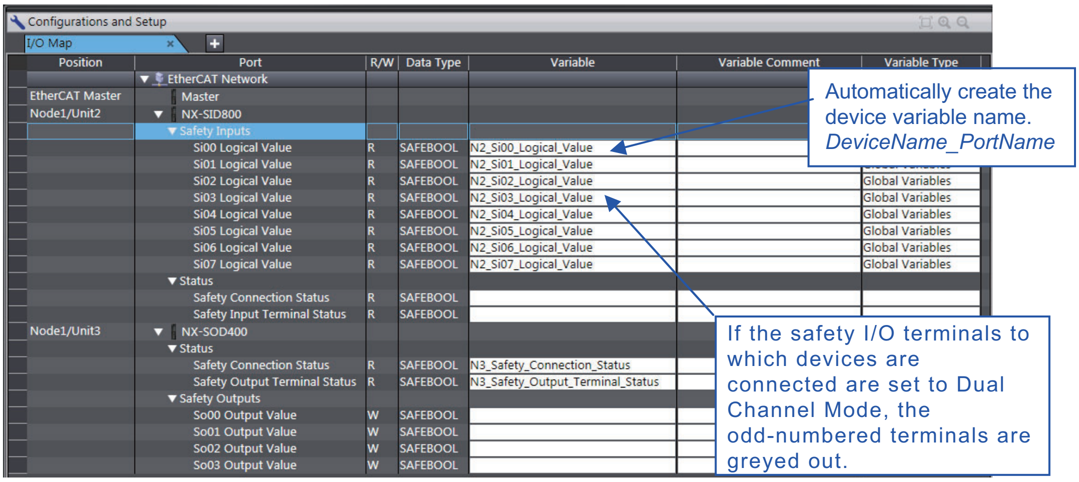
This section describes how to assign device variables to I/O ports through the I/O Map of the Safety CPU Unit.

● 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 Select the Safety CPU Unit as the Controller and double-click **I/O Map** under **Configurations and Setup**.

The I/O Map Tab Page 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 Safety I/O Unit or I/O port, and then right-click and select ***Create Device Variables*** from the menu.

If you choose the ***Create Device Variables*** 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 Select the Safety CPU Unit as the Controller and double-click **I/O Map** under **Configurations and Setup**.

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 Unit are set to Dual Channel Mode, the device variable can only be assigned to an even-numbered terminal.

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	Safety Input Data 00	Gives the status of safety input terminal Si00. 0: OFF, 1: ON	0
Si01 Logical Value	SAFEBOOL	R	Safety Input Data 01	Gives the status of safety input terminal Si01. 0: OFF, 1: ON	0
Si02 Logical Value	SAFEBOOL	R	Safety Input Data 02	Gives the status of safety input terminal Si02. 0: OFF, 1: ON	0
Si03 Logical Value	SAFEBOOL	R	Safety Input Data 03	Gives the status of safety input terminal Si03. 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. 0: An error has occurred on one of the safety input terminals. 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	Safety Input Data 00	Gives the status of safety input terminal Si00. 0: OFF, 1: ON	0
Si01 Logical Value	SAFEBOOL	R	Safety Input Data 01	Gives the status of safety input terminal Si01. 0: OFF, 1: ON	0
Si02 Logical Value	SAFEBOOL	R	Safety Input Data 02	Gives the status of safety input terminal Si02. 0: OFF, 1: ON	0
Si03 Logical Value	SAFEBOOL	R	Safety Input Data 03	Gives the status of safety input terminal Si03. 0: OFF, 1: ON	0
Si04 Logical Value	SAFEBOOL	R	Safety Input Data 04	Gives the status of safety input terminal Si04. 0: OFF, 1: ON	0
Si05 Logical Value	SAFEBOOL	R	Safety Input Data 05	Gives the status of safety input terminal Si05. 0: OFF, 1: ON	0
Si06 Logical Value	SAFEBOOL	R	Safety Input Data 06	Gives the status of safety input terminal Si06. 0: OFF, 1: ON	0
Si07 Logical Value	SAFEBOOL	R	Safety Input Data 07	Gives the status of safety input terminal Si07. 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. 0: An error has occurred on one of the safety input terminals. 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. 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. 0: An error has occurred on one of the safety output terminals. 1: All of the safety output terminals are normal (no errors).	0
So00 Output Value	SAFEBOOL	W	Safety Output Data 00	Gives the status of safety output terminal So00. 0: OFF, 1: ON	0
So01 Output Value	SAFEBOOL	W	Safety Output Data 01	Gives the status of safety output terminal So01. 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. 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. 0: An error has occurred on one of the safety output terminals. 1: All of the safety output terminals are normal (no errors).	0
So00 Output Value	SAFEBOOL	W	Safety Output Data 00	Gives the status of safety output terminal So00. 0: OFF, 1: ON	0
So01 Output Value	SAFEBOOL	W	Safety Output Data 01	Gives the status of safety output terminal So01. 0: OFF, 1: ON	0

Port	Data type	R/W	Name	Description	Default
So02 Output Value	SAFEBOOL	W	Safety Output Data 02	Gives the status of safety output terminal So02. 0: OFF, 1: ON	0
So03 Output Value	SAFEBOOL	W	Safety Output Data 03	Gives the status of safety output terminal So03. 0: OFF, 1: ON	0

6-7 Sharing Variable Data with the NJ-series CPU Unit

To use standard process data communications between the Safety CPU Unit and NJ-series CPU Unit, you must expose the global variables in the Safety CPU Unit to the NJ-series CPU Unit.

This setting allows the NJ-series CPU Unit to monitor and send commands to the Safety CPU Unit through standard process data communications.

6-7-1 Exposing Global Variables to the NJ-series CPU Unit

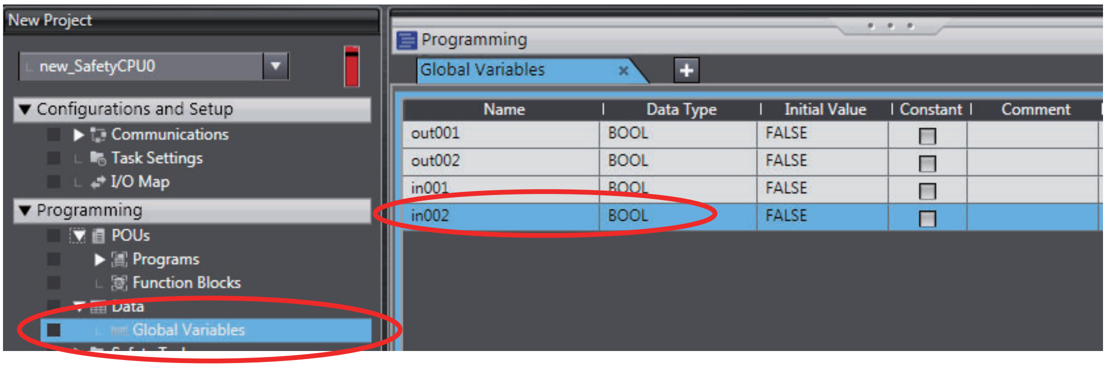
When you set global variables in the Safety CPU Unit for standard process data communications, the variables are exposed as I/O ports in the I/O Map of the NJ-series CPU Unit. When the exposed variables are assigned to the I/O ports, you can access the global variables in the Safety CPU Unit from programs in the NJ-series CPU Unit.

Exposing Global Variables to an NJ-series CPU Unit

1. Select the target Safety CPU Unit from the Controller Selection Box in the Multiview Explorer.
2. Double-click **Global Variables** under **Programming - Data** in the Multiview Explorer.

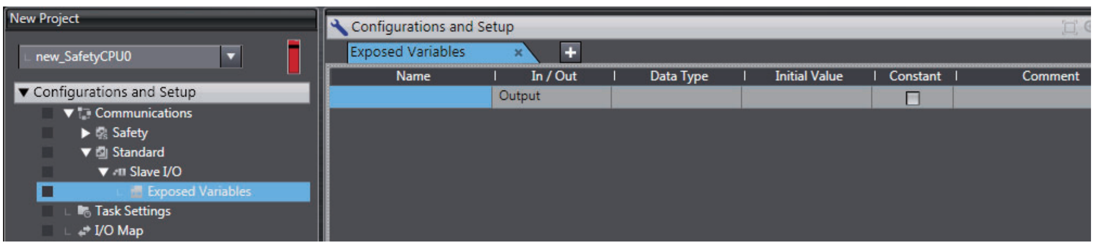
   The global variable table is displayed in the Edit Pane.
3. Register global variables with standard data types.

   Refer to *7-5-3 Registering Variables* on page 7-29 for details on registering variables.



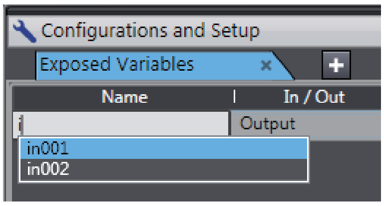
4. Double-click **Exposed Variables** under **Configurations and Setup** – **Communications** – **Standard** – **Slave I/O**.

   The Exposed Variable Tab Page is displayed.



5 Enter the name of the variable to expose (the global variable that was registered in step 3) to the NJ-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.

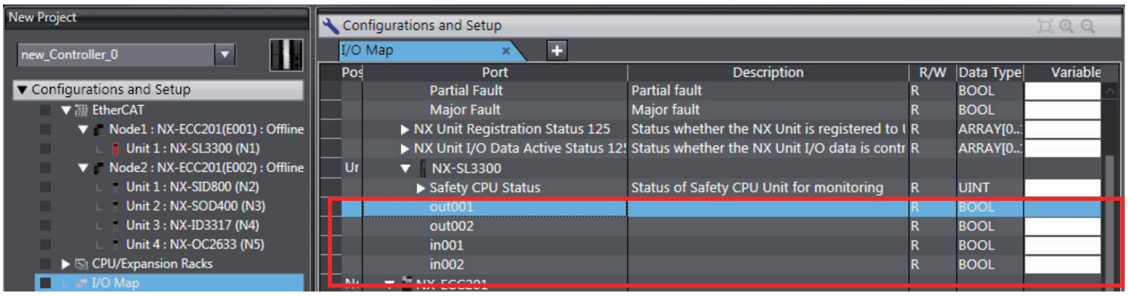


The above settings will cause the exposed variables in the Safety CPU Unit to appear in the I/O Map for the NJ-series CPU Unit.

6 Select the target NJ-series CPU Unit from the Controller Selection Box in the Multiview Explorer.

7 Double-click **I/O Map** under **Configurations and Setup** on the Multiview Explorer.

The exposed variables in the Safety CPU Unit are displayed for the I/O ports.

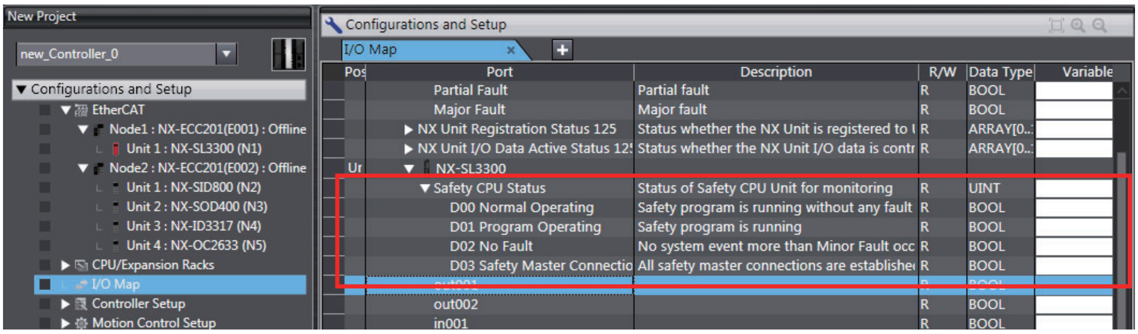


The I/O Map of the NJ-series CPU Unit displays the data types that correspond to the data types of the exposed Safety CPU Unit variables. The following table gives the variable data types that can be exposed for Safety CPU Units and the corresponding data types that are displayed for the NJ-series CPU Unit.

Variable data type that can be exposed for Safety CPU Units	Data type displayed for NJ-series CPU Unit
BOOL	BOOL
BYTE	USINT
DINT	DINT
INT	INT
WORD	UINT

6-7-2 Safety CPU Unit Status

When you place a Safety CPU Unit on the NX bus of an EtherCAT Coupler Unit, standard process data communications are automatically set up and the status of the Safety CPU Unit is displayed as an I/O port in the I/O Map of the NJ-series CPU Unit. You can use the I/O port to monitor the status of the Safety CPU Unit from the NJ-series CPU Unit.



Refer to *A-5 Safety CPU Unit Status* on page A-58 for details on the items in the Safety CPU Unit status.

6-7-3 I/O Ports for Safety I/O Units

You can access the values of the ports for Safety I/O Units from an NJ-series CPU 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-series CPU Unit. Refer to *A-6 I/O Ports for Safety I/O Units That Are Displayed in the I/O Map of the NJ-series CPU Unit* on page A-59 for descriptions of the I/O ports for Safety I/O Units that are displayed in the I/O Map of the NJ-series CPU Unit.

Additional Information

You can only read the values of the ports of Safety I/O Units from an NJ-series CPU Unit. You cannot write the values.

6-7-4 I/O Refreshing Method

This section describes the I/O refreshing method of the Safety Control Units.

● I/O Refreshing Method

Only Free-Run refreshing can be used for Safety Control Units. 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.

6-8 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.

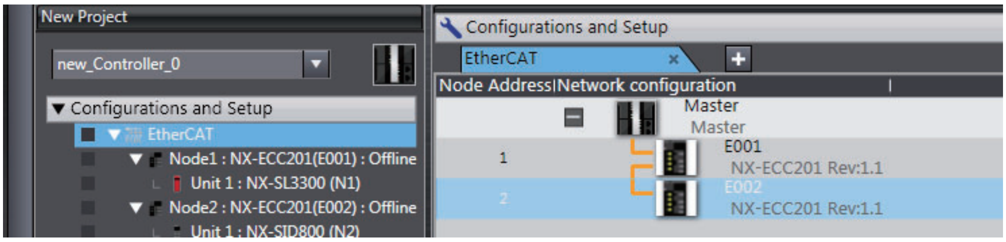
You can export or import the two groups of data that are given below.

- Settings for Entire EtherCAT Slave Terminal
  The data for the entire EtherCAT Slave Terminal consists of the Slave Terminal configuration information for the EtherCAT Coupler Unit and all NX Units that are connected to that Coupler Unit. It also contains the safety application data.
- Safety Application Data for Only the Safety CPU Unit
  The safety application data consists of the safety program and the safety tasks and settings.

6-8-1 Exporting/Importing the Settings for the Entire Slave Terminal

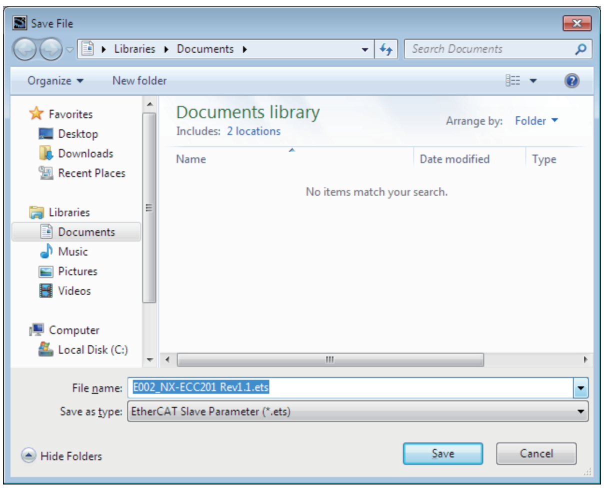
You can export the settings for the entire Slave Terminal into a single file (extension .ets). The exported settings file for the entire Slave Terminal can be imported to reuse the settings for an entire Slave Terminal with the same settings in a different project on the Sysmac Studio, or a project for which a Safety CPU Unit has not been registered.

1 Select the NJ-series CPU Unit as the Controller and double-click **EtherCAT** under **Configurations and Setups** 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.

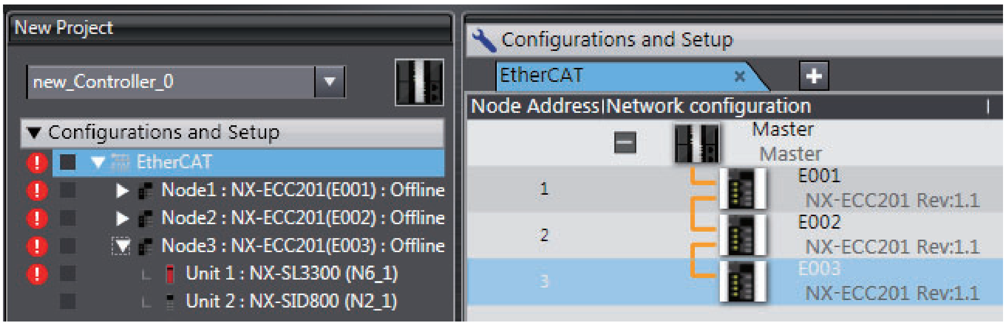


3 Enter a file name, and then click the **Save** Button.

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.



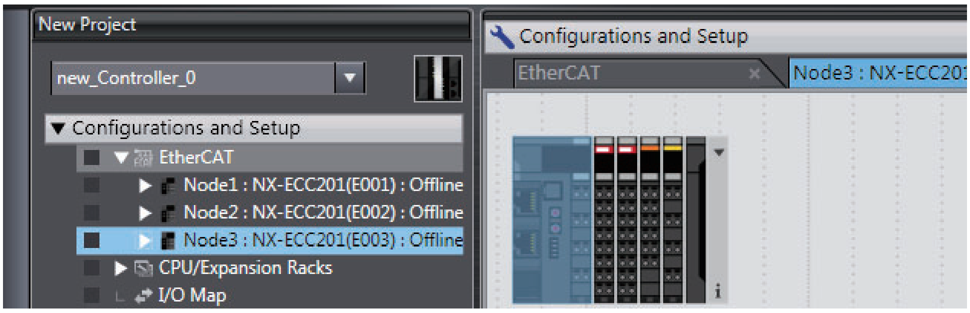
If importing data results in two or more Safety CPU Units, an error will occur. Delete the Safety CPU Units that are not used.

6-8-2 Exporting and Importing Safety Application Data for Only the Safety CPU Unit

You can export and import the safety application data for only the Safety CPU Unit as a single file (extension .nsf).

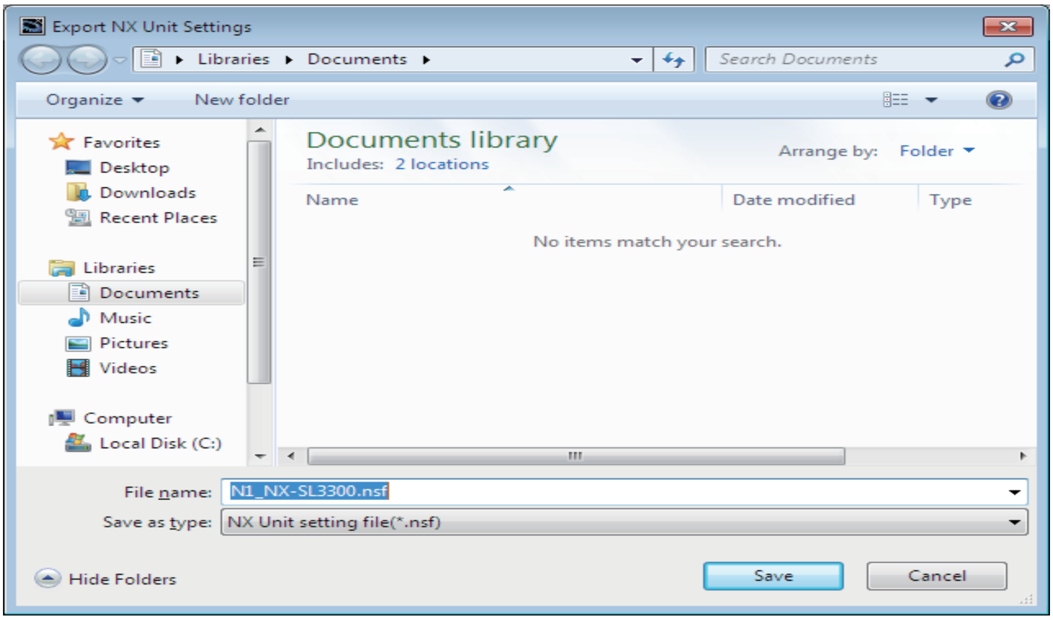
The exported Safety CPU Unit settings file can be imported to reuse the safety application data for a Safety CPU Unit with the same settings. To do this, go into the Slave Terminal Tab Page in a different project on the Sysmac Studio, or a project for which a Safety CPU Unit has not been registered.

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 EtherCAT Tab Page, and then right-click and select ***Import NX Unit Settings and Insert New Unit*** from the menu.

The Safety CPU Unit to import is added to the Slave Terminal Tab Page.

If importing data results in two or more Safety CPU Units, an error will occur. Delete the Safety CPU Units that are not used.

7

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.

7-1 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 *7-5 Programming Operations* on page 7-26 for the procedures to create POUs on the Sysmac Studio.

Refer to the *Sysmac Studio Version 1 Operation Manual* (Cat. No. W504) for the procedures to create POUs that are used with an NJ-series CPU Unit.

7-1-1 What Are POUs?

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 POUs consist of a combination of these three types of POUs. You can create many POUs. You assign the safety programs to a safety task to execute them. Only one safety task can be used by the Safety CPU Unit.

7-1-2 Overview of the Three Types of POUs

Programs

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.
- You cannot use user-defined function blocks inside other user-defined function blocks.

Refer to *7-1-5 Details on Function Blocks* on page 7-5 for details on function blocks.

Functions

Executing Functions and Execution Conditions

- You can call functions from programs or function blocks to execute them.
- Functions are always executed.

Notation

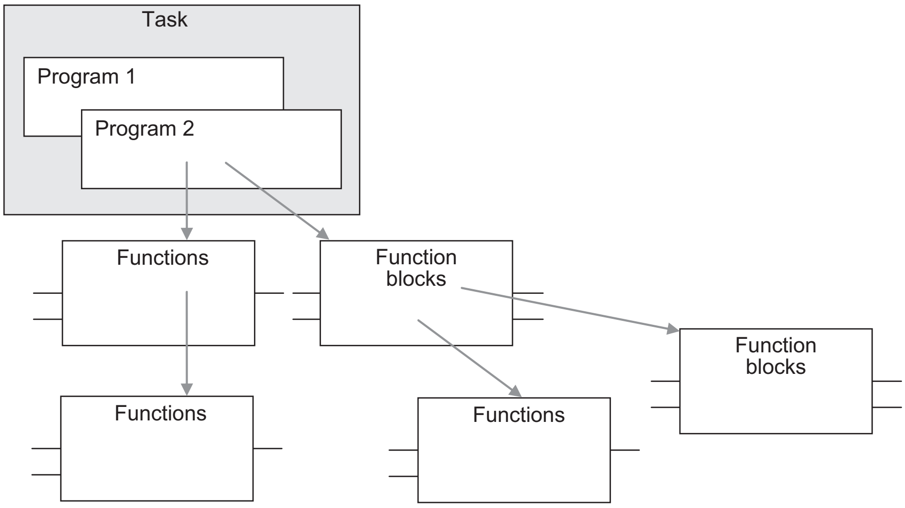
- You cannot create user-defined functions.
- System-defined functions are sometimes called FUN instructions.
- The values of internal variables are not retained. The output value remains constant as long as the input value is constant.

Refer to *7-1-6 Details on Functions* on page 7-9 for details on functions.

7-1-3 Differences between Programs, Functions, and Function Blocks

		Programs	Function blocks (FBs)	Functions (FUNs)
Type		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	**All Instructions**	Supported.	Supported.	---
	User-defined function blocks	Supported.	Supported.	
Execution conditions		Always executed.	Always executed. Specify the execution condition with an input variable.	Always executed.

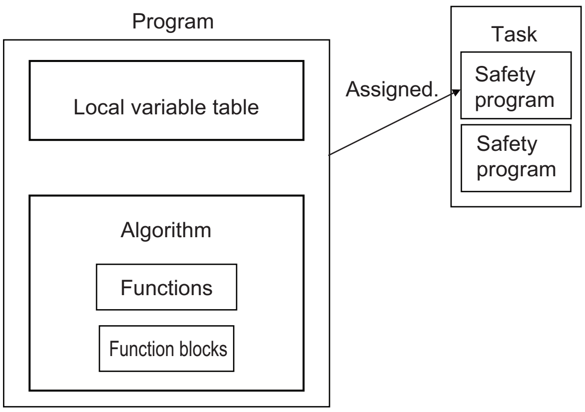
The hierarchical relationships between programs, functions, and function blocks are shown in the following figure.



7-1-4 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 with the Controller set to the Safety CPU Unit on the Sysmac Studio. Refer to *7-5 Programming Operations* on page 7-26 for programming operations.

7-1-5 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.

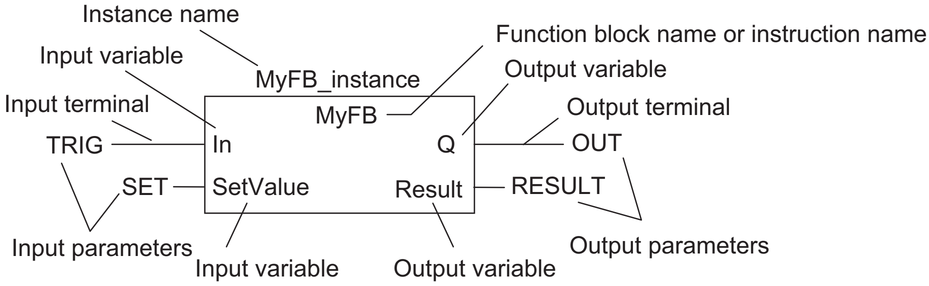
Create the algorithm.

2 Place an instance of the function block definition in the program.

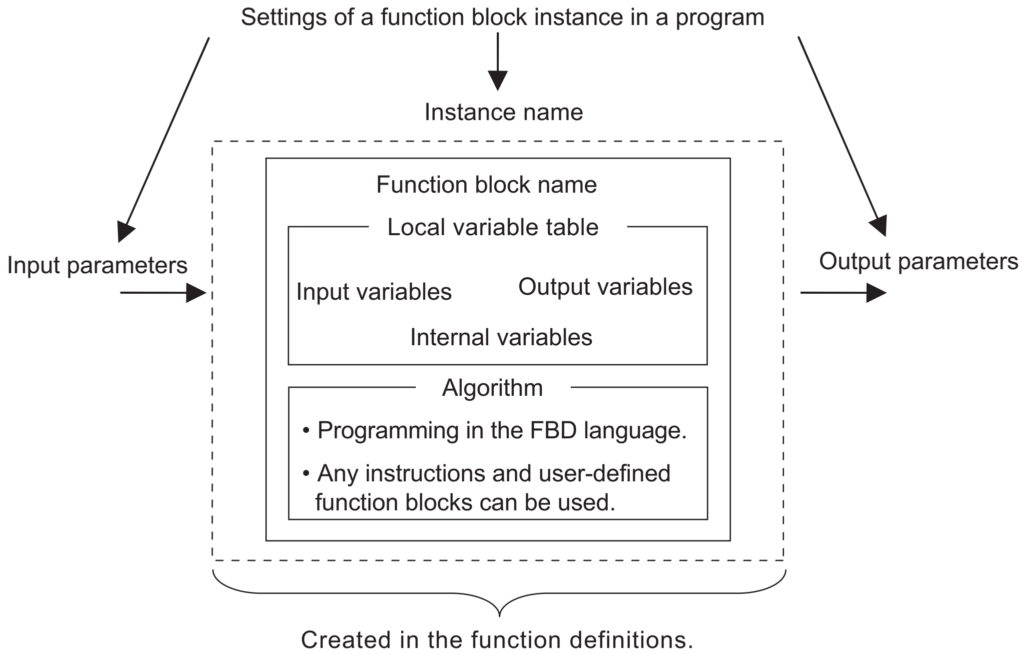
Call the function block definition from a program or another function block. You can call the same function block definition from more than one program or function block. After you place an instance of a function block definition in a program or in another function block, you can manipulate and execute 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.

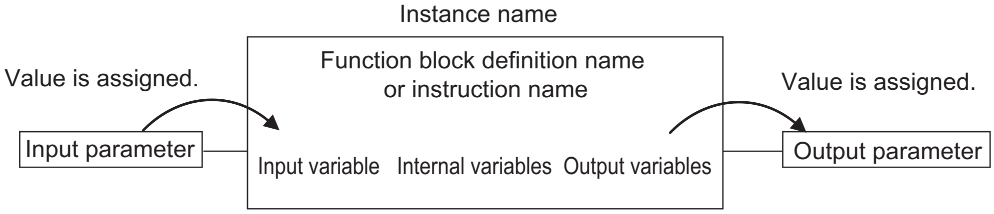
Parameters

- 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. Refer to the *NX-series Safety Control Unit Instruction Command Manual* (Cat. No. Z931) for details on the operation.

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. • You can specify either constants or variables for input parameters. • Omitting Input Parameters: Refer to *Operation When Parameters Are Omitted* in *7-5-3 Common Operations for Functions (FUNs) and Function Blocks (FBs)* in the *NX-series Safety Control Unit Instructions Reference Manual* (Cat. No. Z931). • You can access the values from outside of the function block. Access these values with the following format: *InstanceName.InputVariable-Name*. However, you cannot write values directly to an input variable.
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. • You cannot specify a constant for an output parameter. Only variables may be specified. • You can omit output parameter connections. If you omit an output parameter, the value of the output variable is not assigned to any parameter. • Omitting Output Parameters: Refer to *Operation When Parameters Are Omitted* in *7-5-3 Common Operations for Functions (FUNs) and Function Blocks (FBs)* in the *NX-series Safety Control Unit Instructions Reference Manual* (Cat. No. Z931). • You can access the value from outside of the function block. Access these values with the following format: *InstanceName.OutputVariable-Name*. 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. • 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 *7-2-4 Attributes of Variables* on page 7-13 for details on the variable attributes that can be set for each type of variable.

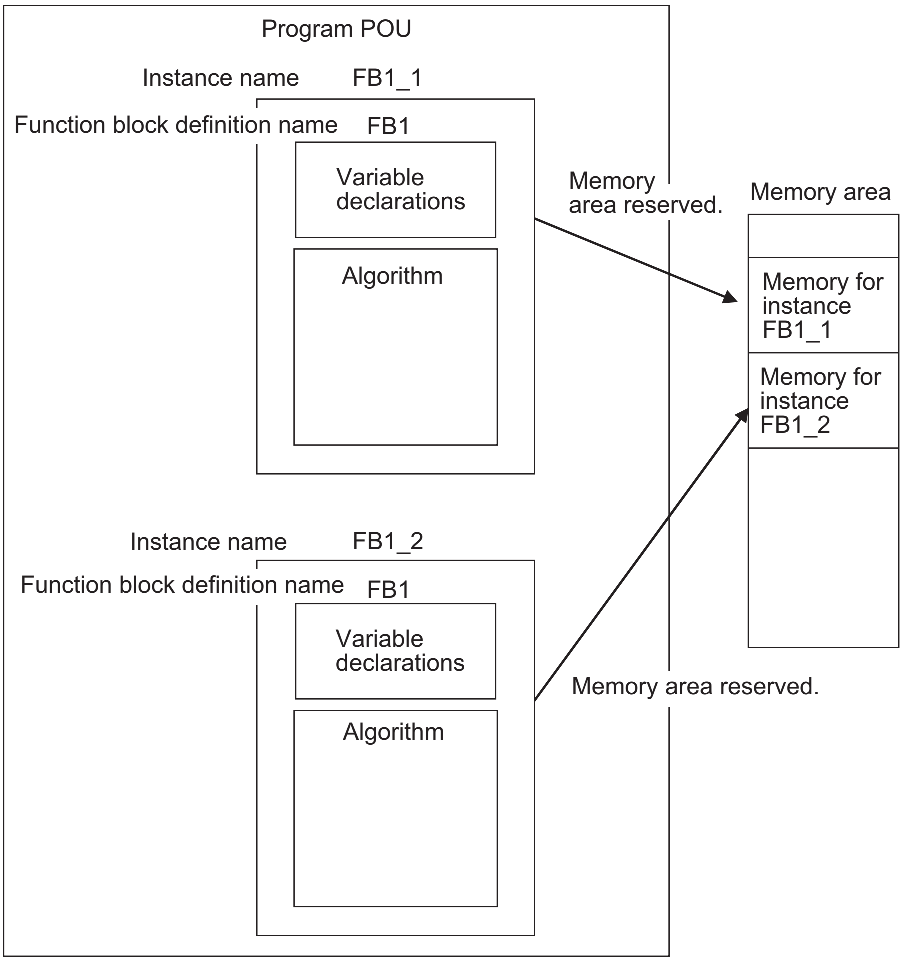
Function Block Definitions and Instances

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 Instances

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.

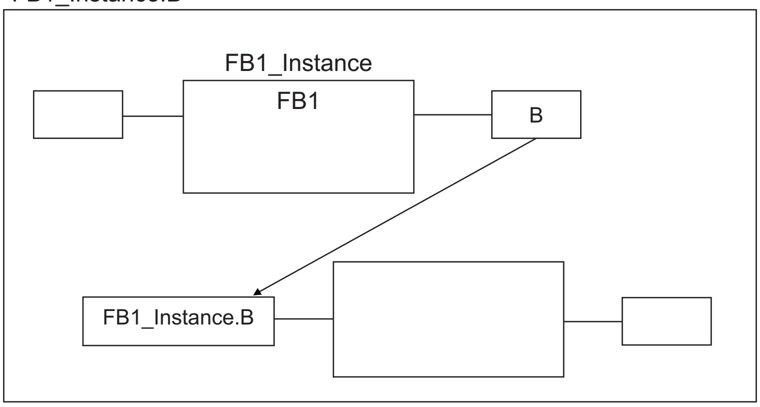
Accessing Variables in a Function Block from Outside the Function Block

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*

FB1_Instance.B



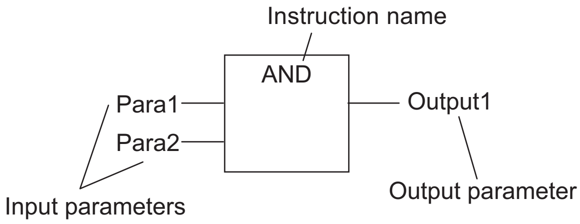
You can access the input and output variables for a function block only within the program that contains the function block instance. However, you cannot access these variables from within other function block instances even if they are in the same program. You cannot access them from other programs.

7-1-6 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.

7-1-7 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.

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 function blocks.

	User-defined	System-defined = Instructions
FB	Supported.	Supported.
FUN	Not supported.	Supported.

Refer to the *NX-series Safety Control Unit Instructions Reference Manual* (Cat. No. Z931) for details on instructions.

7-2 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.

7-2-1 Variables

Variables store I/O data for exchange with external devices or temporary data that is used for internal POU processing. A variable has attributes, such as a 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.

7-2-2 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.

7-2-3 Types of User-defined Variables

There are five types of user-defined variables as defined according to their function in a POU.

OK: Definable.

Type of user-defined variable		POU type	
		Programs	Function blocks
Local variables	Internal variables	OK	OK
	Input variables	None	OK
	Output variables	None	OK
	External variables	OK	None
Global variables		OK*1	None

*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

You declare global variables in the global variable table.

Device variables that are automatically created from the Slave Terminal configuration are automatically registered as global variables.

7-2-4 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.	127 bytes max.	Name
Data Type	The data type defines the format of the data that is stored in the variable.	---	BOOL
Initial Value	Specify a value for the variable for one of the following situations: • When the power supply is turned ON • When the mode is changed to RUN mode or DEBUG mode (STOPPED)	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	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.

7-2-5 Data Types

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.
Integer	A data type that represents an integer value.
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 the following two data types to distinguish between safety signals and standard signals.

- Safety data types: These data types represent signals related to safety control.
- Standard data types: These data types represent signals related to standard control.

The safety data type variables are prefixed with the “SAFE” before the name of the standard data type, as in SAFEBOOL and SAFEBYTE.

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.

The data size and alignments are given for data types that can be assigned to PDOs. These have the following meanings.

- Data size: The actual size of the value.
- Alignment: The unit used to allocate memory.

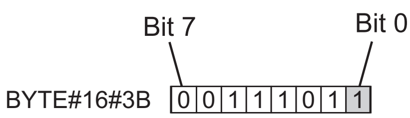
Type	Data type	Safety/standard data type	Data size	Alignment	Range of values	Notation
Boolean	BOOL	Standard data type	8 bits	1 byte	FALSE or TRUE	bool#0 or bool#1 FALSE or TRUE
	SAFEBOOL	Safety data type	---	---		
Bit strings	BYTE	Standard data type	8 bits	1 byte	byte#16#10 to byte#16#FF	byte#2#0101010 byte#2#0101_1010 byte#16#5A You can use the separator character "_".
	SAFEBYTE	Safety data type	---	---		
	WORD	Standard data type	16 bits	2 bytes	word#16#0000 to word#16#FFFF	
	SAFEWORD	Safety data type	---	---		
	DWORD	Standard data type	---	---	dword#16#00000000 to dword#16#FFFFFFFF	
	SAFEDWORD	Safety data type	---	---		
Integers	INT	Standard data type	16 bits	2 bytes	int#–32768 to int#32767	100 int#100 int#2#00000000_1100100 int#16#64 -100
	SAFEINT	Safety data type	---	---		
	DINT	Standard data type	32 bits	4 bytes	dint#-2147483648 to dint#2147483647	
	SAFEDINT	Safety data type	---	---		
Durations	TIME	Standard data type	---	---	t#0ms to t#2147483647ms and t#0d0h0m0s0ms to t#24d20h31m23s647ms	t#3000ms

Bit String Data Format

This section describes the data format for bit string data.

● Bit String Data Format

Bit 0 is the least significant bit of a bit string variable. Bit values are expressed as 1 or 0.



7-2-6 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	Setting initial values
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.

7-2-7 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.

7-2-8 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

<table>
<tr><th>Safety program-related name</th><th>Applicable characters</th><th>Reserved words</th><th>Multibyte character compatibility</th><th>Case sensitivity</th><th>Maximum size (not including NULL)</th><th>Character encoding</th></tr>
<tr><td>Variable name (including POU instance names)</td><td rowspan="4">Usable characters
• 0 to 9, A to Z, and a to z.
• _ (underlines)
Refer to Reserved Words below for a list of the reserved words.
Characters that cannot be used together
• A text string that starts with a number (0 to 9)
• A text string that starts in an underline (_) character
• A text string that contains more than one underline (_) character
• A text string that ends in an underline (_) character
• Identifiers formed from a string of characters that is prefixed or suffixed with one or more expansion characters or spaces.</td><td rowspan="4">Refer to Reserved Words below.</td><td rowspan="4">Not supported.</td><td rowspan="4">Not case sensitive.</td><td rowspan="2">127 bytes</td><td rowspan="4">ASCII</td></tr>
<tr><td>POU definition names</td></tr>
<tr><td>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.)</td><td>511 bytes</td></tr>
<tr><td>Device names</td><td>127 bytes</td></tr>
<tr><td>Variable comments</td><td></td><td>None</td><td>Supported.</td><td>---</td><td>No limit[*1]</td><td>UTF-8</td></tr>
</table>

*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

7-3 Constants (Literals)

This section describes constants.

7-3-1 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.

7-3-2 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 • Base: 2, 8, 10, or 16 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. • 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 data.

Bit Strings

● Bit String Data

Notation	Example	Remarks
{*data_type*}#{*base*}#{*numeric _value*]	word#16#0064	• Data type: byte, word, or dword • Base: 2, 8, 10, or 16 • 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

7-4 Programming Languages

This section describes the programming languages in detail.

Refer to *7-5 Programming Operations* on page 7-26 to learn how to enter the programming languages on the Sysmac Studio.

7-4-1 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.

7-4-2 FBD (Function Block Diagram) 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.

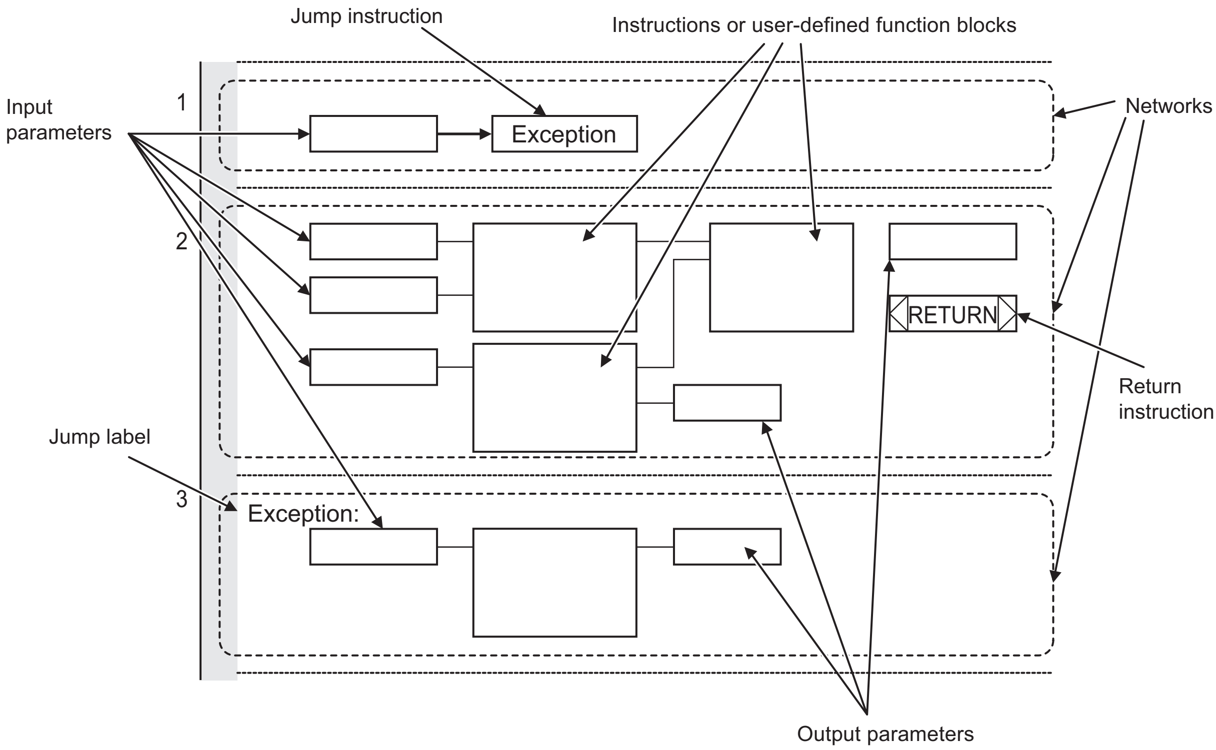
Elements of the FBD Language

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 of user-defined variables between terminals
  c) Horizontal and vertical flow between instructions or between user-defined variables
- Instructions and user-defined variables 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.

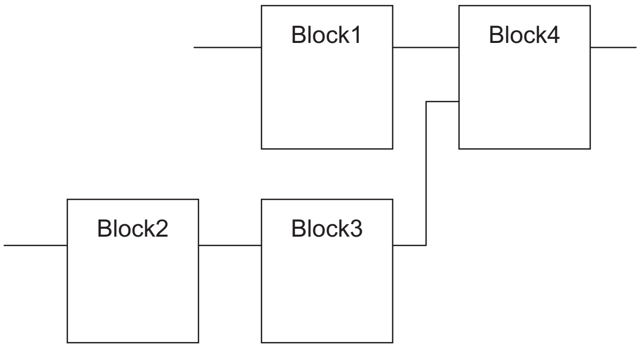
Refer to *Execution Order of Safety Programs Written in the FBD Language* on page 7-23 and *Execution Control* on page 7-23 later in this manual, and also to the *NX-series Safety Control Unit Instructions Reference Manual* (Cat. No. Z931).

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.



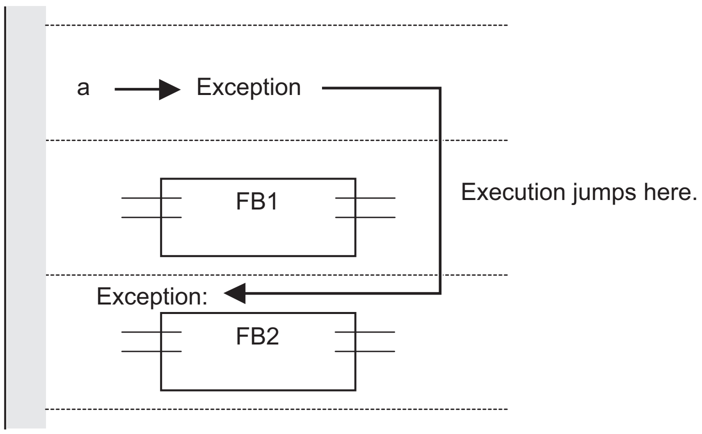
However, if there is a Return instruction in the middle of a program and the execution condition is met, the POU is ended and a return is made to the source of the call. No processes after the Return instruction are executed.

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.”

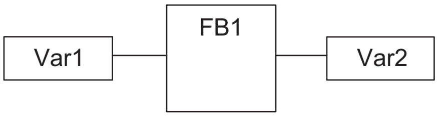
You cannot jump to a network that is above the current network.



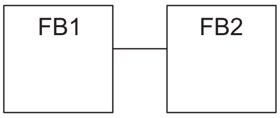
Connecting Instructions or User-defined Function Blocks

● Correct Connection Configurations

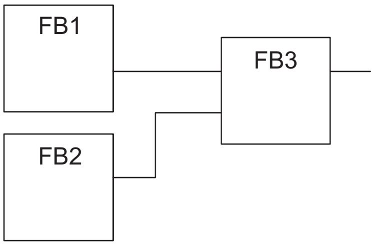
You can connect a parameter to an instruction or a user-defined function block with a connecting line.



You can connect a pair of instructions or a pair of user-defined function blocks with a connecting line.

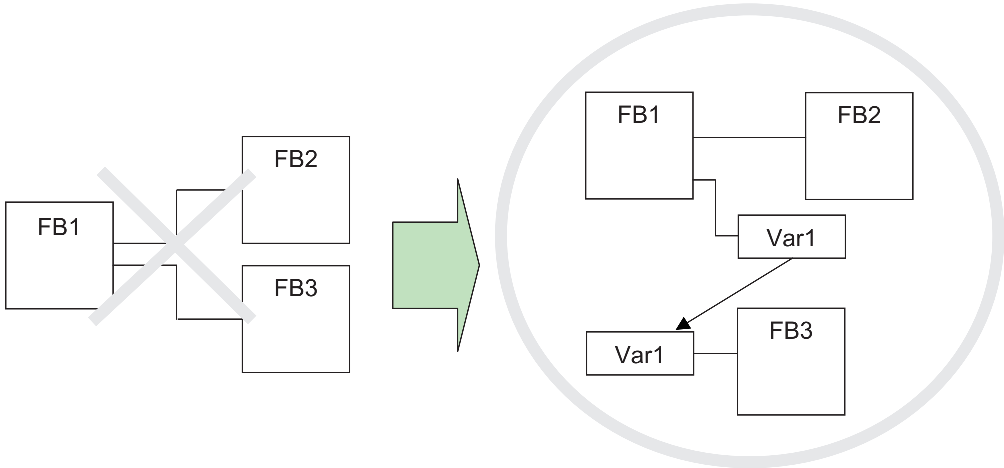


You can also connect more than one instruction or user-defined function block to another instruction or user-defined function block.

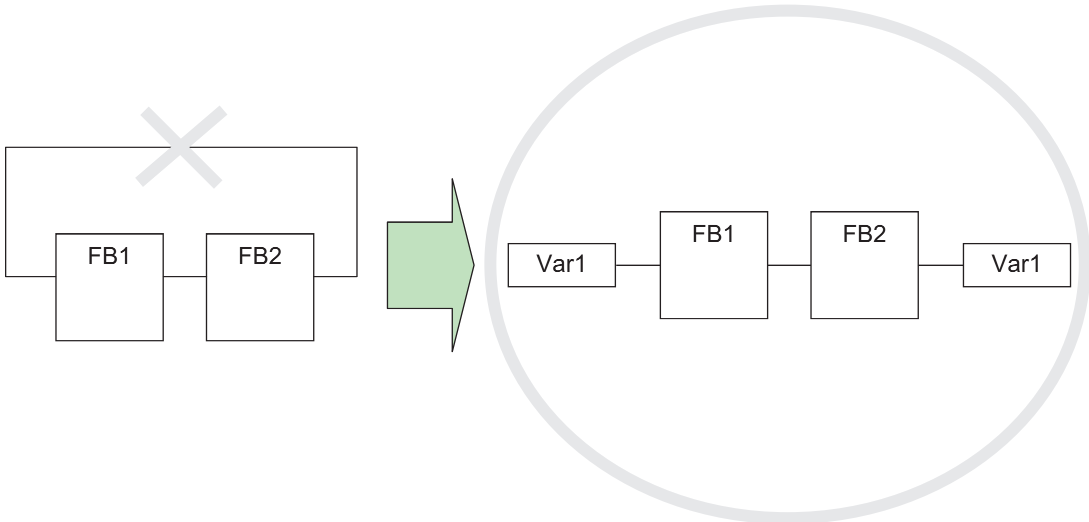


Incorrect Connection Configurations

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.

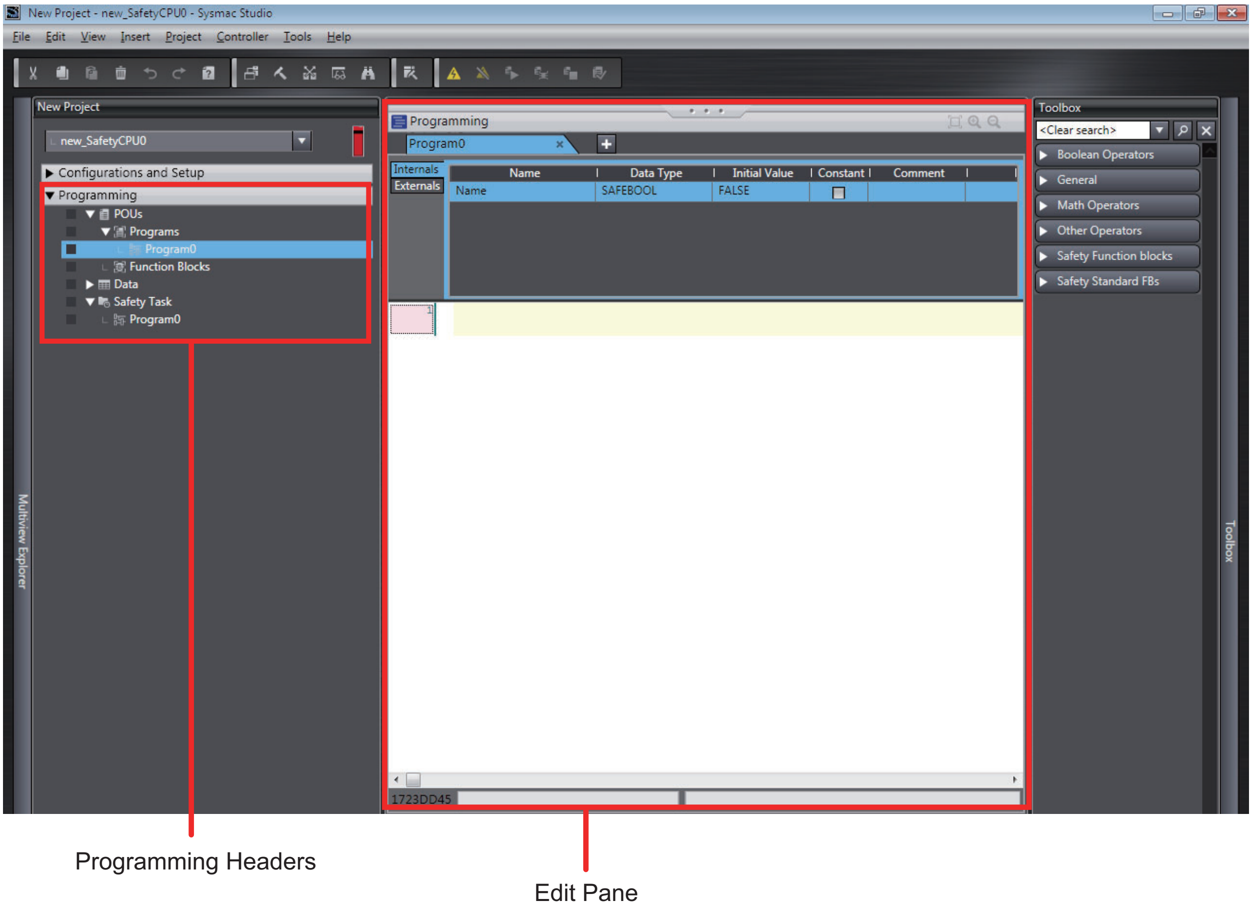


7-5 Programming Operations

This section describes the procedures on the Sysmac Studio that you use to create safety programs for the Safety CPU Unit.

7-5-1 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. Program0 is created when you create a new project. Double-click a program to display it in the FBD editor and begin editing. In the Multiview Explorer, you can change the names of programs or delete, copy, paste, and cut programs.
		Program1	
	Function Blocks		
		FunctionBlock0	A list of user-defined function blocks is displayed. There are no function blocks when you create a new project. Double-click a function block to display it in the FBD editor and begin editing. 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.

7-5-2 Registering POUs

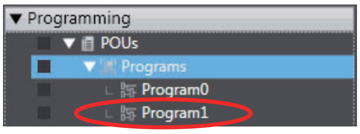
This section shows how to register programs and function blocks.

Registering Programs

● Registering New Programs

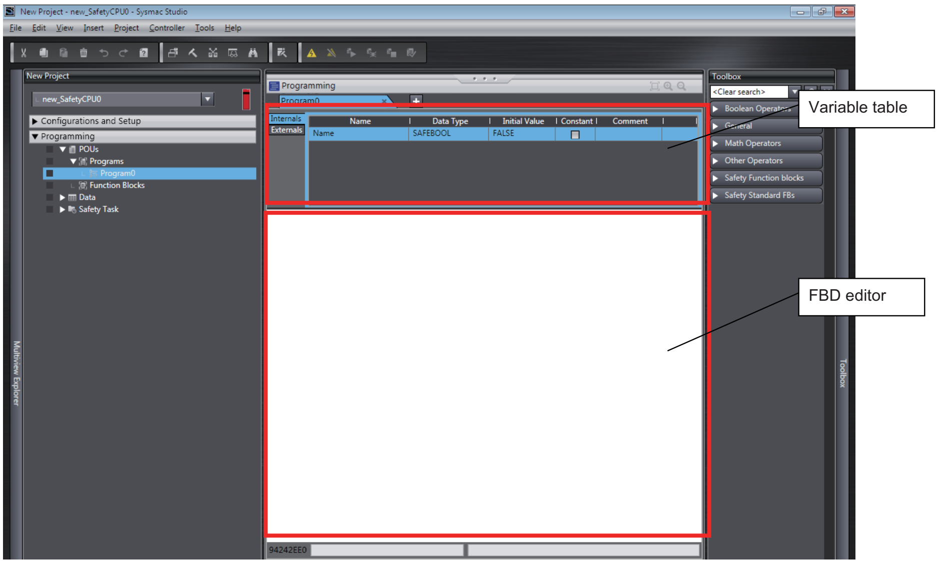
1 Right-click **Programs** under **Programming** – **POUs** and select ***Add*** – ***Program*** from the menu.

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 Programming Layer of the Edit Pane. From here you can edit programs.



Refer to *7-5-3 Registering Variables* on page 7-29 for information on how to register variables, and *7-5-4 FBD Programming* on page 7-34 for information on programming in the FBD editor.

Registering Function Blocks

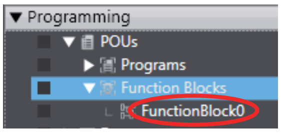
Function blocks are written in the FBD language. You can call them from safety programs as required. You can use functions inside function blocks. Refer to *7-1-2 Overview of the Three Types of POUs* on page 7-3 for a detailed description of function blocks.

● Registering Function Blocks

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 - Function Block*** 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 Programming Layer of the Edit Pane. From here you can create local variables and FBD networks.

Refer to *7-5-3 Registering Variables* on page 7-29 for information on how to register variables, and *7-5-4 FBD Programming* on page 7-34 for information on programming in the FBD editor.

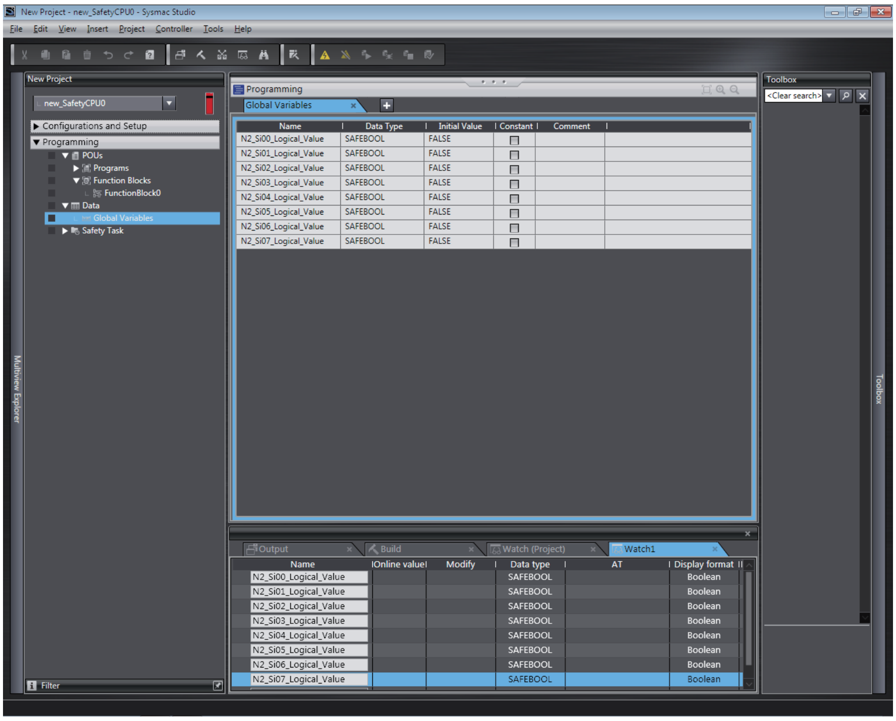
7-5-3 Registering Variables

This section describes how to register global variables and local variables.

Creating Global Variables

● Opening the Global Variable Table

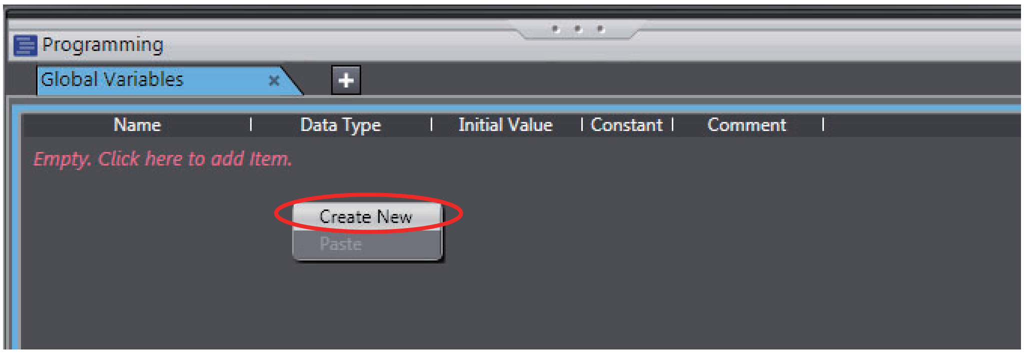
Double-click **Global Variables** under **Programming - Data** in the Multiview Explorer. Or, right-click **Global Variables** under **Programming** – **Data** and select ***Edit*** from the menu. The global variable table is displayed.



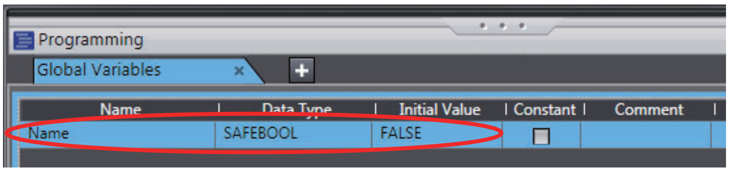
Field	Description	Restrictions
Name	Enter a name to use to identify the variable.	Only single-byte alphanumeric characters are allowed. Multi-byte characters, such as those used for Japanese, are not allowed. The maximum size is 127 bytes.
Data Type	Set the type of data that is stored in the variable. Refer to *7-2-5 Data Types* on page 7-14 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. Either select the check box or clear the selection (default).	---
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

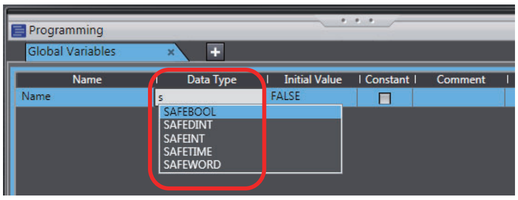
- If there are no registered variables at all, the message "Empty. Click here to add Item." is displayed. Click to add a new variable.

A global variable is registered automatically when you perform any of the following operations.

- When you enter a new variable in the *Variable* Column of the I/O Map with *Global Variables* specified in the *Variable Type* Column.

● Editing Global Variables

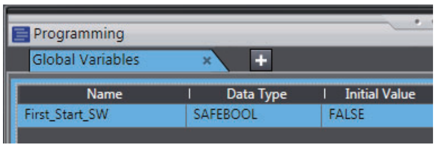
1 Click the cell to edit for the registered variable.



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 that letter is displayed. Select a data type from the list.

2 Change the setting, and then press the **Enter** Key.

The change is applied to the variable.

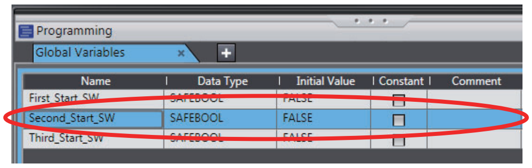


- Displaying and Selecting Entry Candidates
  Entry candidates are displayed in the *Name*, *Data Type*, and *Initial Value* cells.
  Entry candidates that match the characters in the entered text string are displayed as you edit the text string.

- Displaying and Selecting Drag and Drop Entry Candidates
  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.

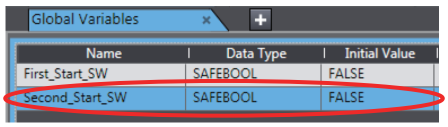


2 Press the **Delete** Key. Or, right-click a row and select ***Delete*** from the menu.

The variable is deleted.

● Copying and Pasting Global Variables

1 Click any cell on the line of the variable to copy.

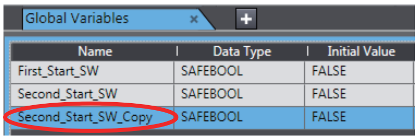


2 Press the **Ctrl + C** Keys. Or, right-click the row and select ***Copy*** from the menu.

The specified variable is copied.

3 Press the **Ctrl + V** Keys. Or, right-click and select ***Paste*** from the menu.

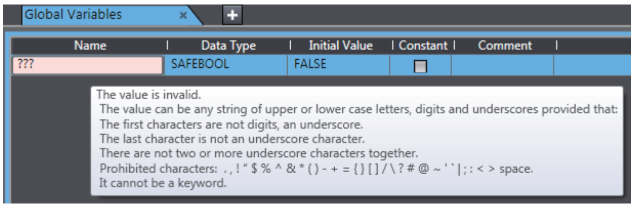
A copy of the variable is registered with "_Copy" added to the name of the variable that was copied on the next row.



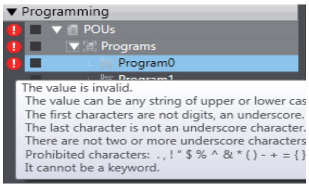
Precautions for Correct Use

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 exclamation 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 *7-2-8 Restrictions on Variable Names and Other Safety Program-related Names* on page 7-17 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)



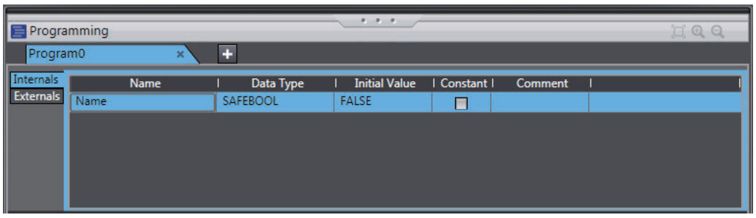
Registering Local Variables

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.

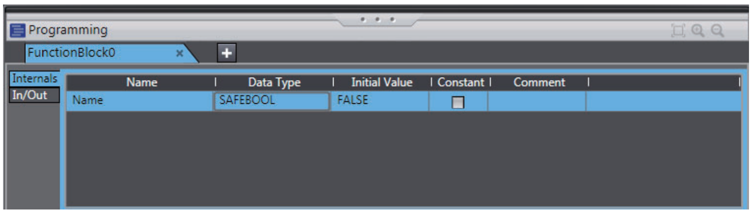
● Registering and Editing Local Variables

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.

• Local Variable Table for Programs



• Local Variable Table for Function Blocks



Field	Description	Restrictions
Name	Enter a name to use to identify the variable.	Only single-byte alphanumeric characters are allowed. Multi-byte characters, such as those used for Japanese, are not allowed. The maximum size is 127 bytes.
Data Type	Set the type of data that is stored in the variable. Refer to *7-2-5 Data Types* on page 7-14 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 (RUN). 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. Press the **Space** Key to select or clear the check box.	---
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 *7-2-4 Attributes of Variables* on page 7-13 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 registering and editing local variables are the same as the procedures used for global variables. Refer to *7-5-3 Registering Variables* on page 7-29.

7-5-4 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).

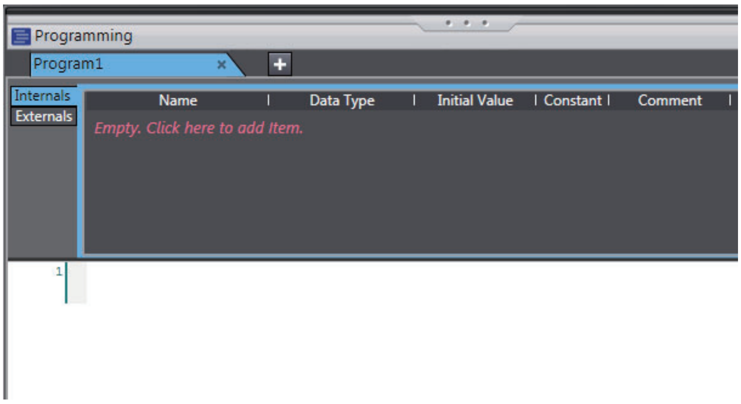
This section describes how to use the FBD editor.

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.

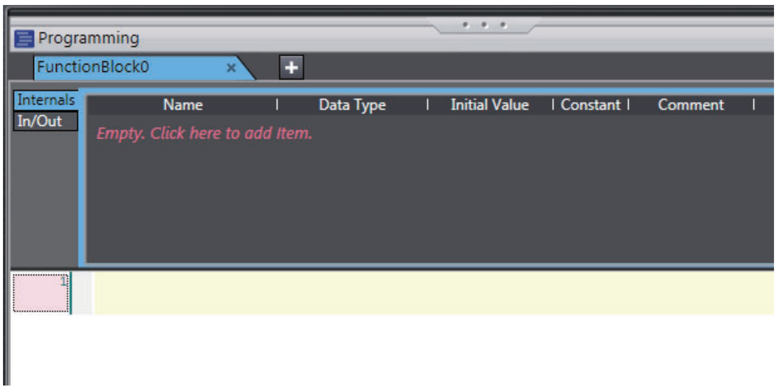


Refer to *7-5-2 Registering POUs* on page 7-27 for the program registration procedure.

● 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 *7-5-2 Registering POUs* on page 7-27 for the function block registration procedure.

Zooming In and Zooming Out of the FBD Editor

Use the icons that are displayed in the upper right of the FBD editor to zoom in or zoom out of the FBD editor.

Inserting FBD Networks

Use one of the following procedures to insert an FBD network.

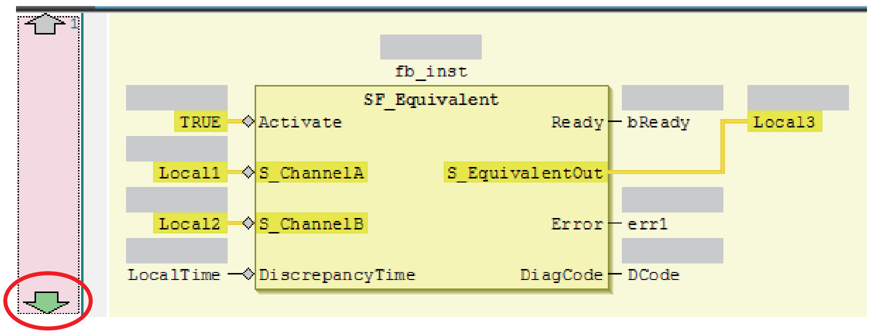
Method 1: Right-click the FBD network and select ***Insert Network Above*** from the menu.

Method 2: Right-click the FBD network and select ***Insert Network Below*** from the menu.

Method 3: Drag a network from the Toolbox to the FBD editor.

An empty FBD network is inserted at the position shown below.

- Right-clicking the FBD Network and Selecting ***Insert Network Above*** from the Menu
  An empty FBD network is inserted before the selected FBD network.
- Right-clicking the FBD Network and Selecting ***Insert Network Below*** from the Menu
  An empty FBD network is inserted after the selected FBD network.
- If you drag a network from the Toolbox:
  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 before the FBD network where the network was dropped.

Deleting FBD Networks

Use one of the following procedures to delete an FBD network.

Method 1: Right-click the FBD network and select ***Delete*** from the menu.

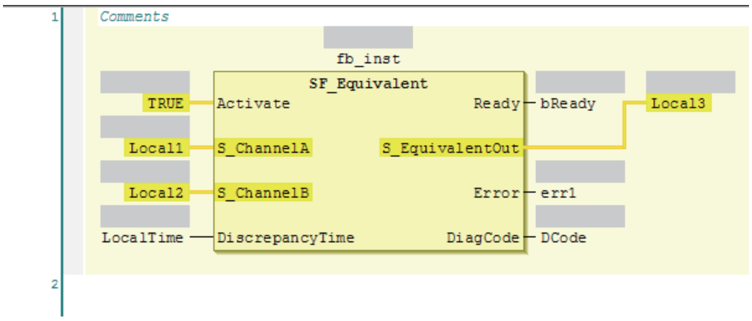
Method 2: Select the FBD network and press the **Delete** Key.

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.

Method: Select the comment portion of the FBD network and edit it.



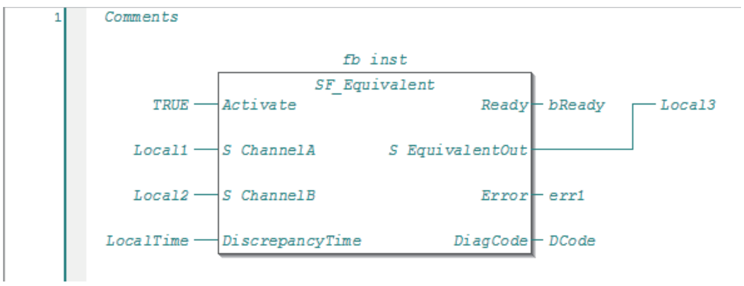
Commenting Out FBD Networks and Restoring Them

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.

Method: Right-click the FBD network and select ***Toggle Network Comment State*** from the menu.

*1. 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 FB or FUN Instruction

Use one of the following procedures to insert a FB or FUN instruction.

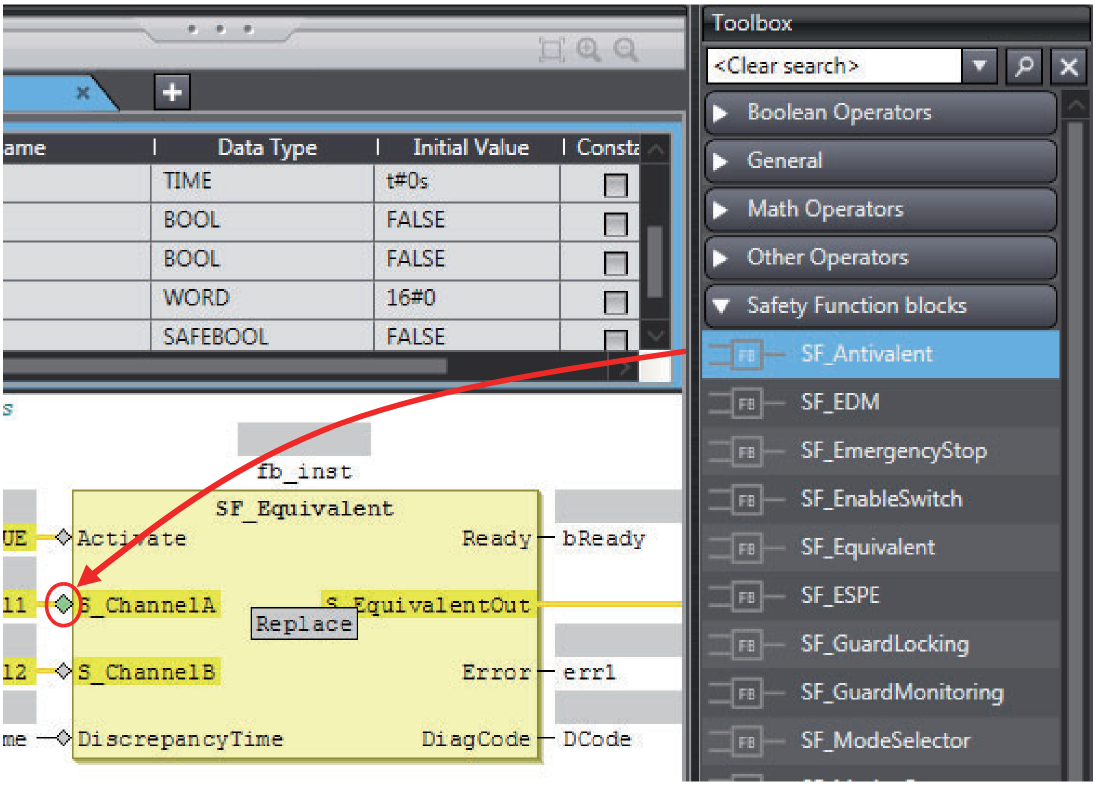
The following procedures are described using the “FB” in place of the phrase “FB or FUN instruction.”

Method 1: Drag a FB from the Toolbox to an I/O terminal on the FB in the FBD editor, or to a new network where the words *Start here* are displayed.

Method 2: Right-click the FBD network and select ***Insert Function Block*** from the menu.

● Procedure for Method 1

1 Drag a 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. 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.



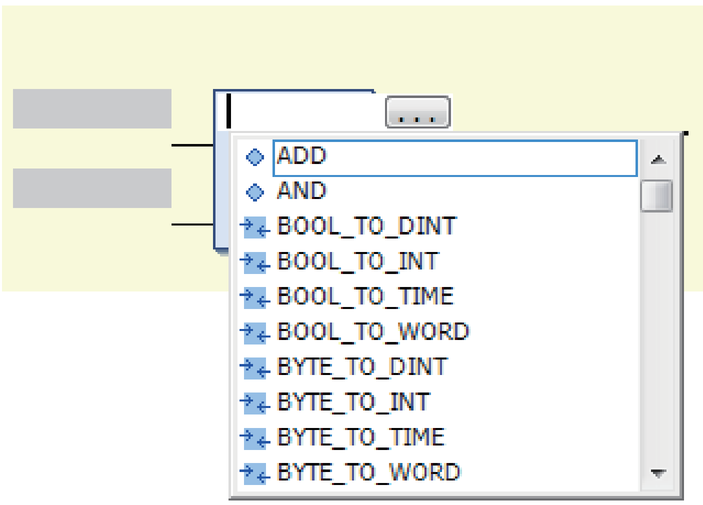
2 Drop the FB on a diamond-shape box to insert it.

● 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 FBs that you can enter.



3 Select a FB from the list.

Precautions for Correct Use

The Toolbox does not show all of the available functions or function blocks. To use a function or function block that is not shown in the Toolbox, use method 2.

Additional Information

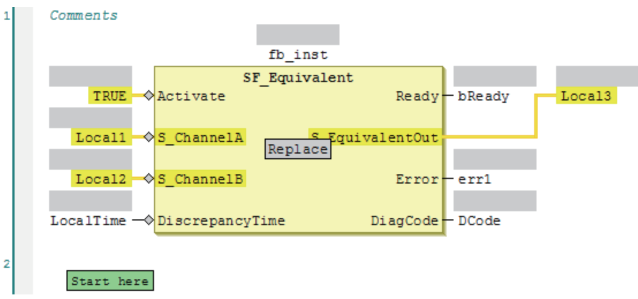
With method 2, you can click the Input Assistant Button (...) to the right of the FB name text box to display the Input Assistant Dialog Box. You can select a FB from the items in this dialog box to insert the selected FB.

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.

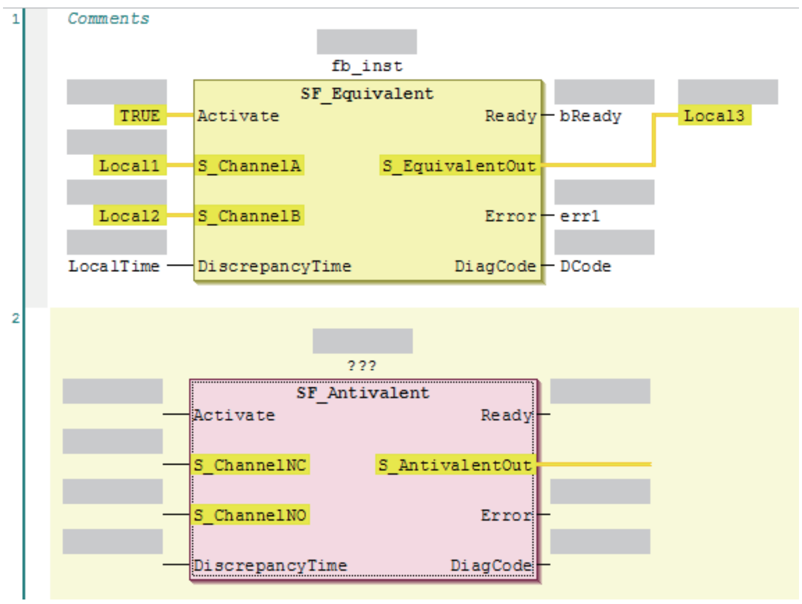
The Input Assistant Dialog Box is also displayed when you right-click an FBD network and select ***Insert Function Block*** from the menu.

The position where the FUN or FB is inserted depends on the insertion method, as described below.

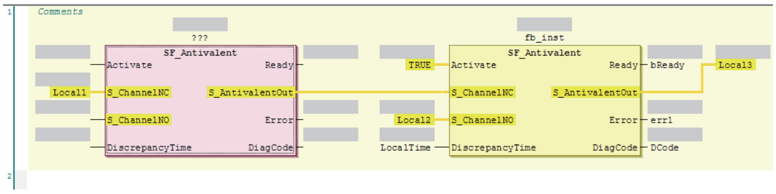
- If you drag a function block (*SF_Antivalent* in the example) from the Toolbox, the function block is inserted at the position shown below.



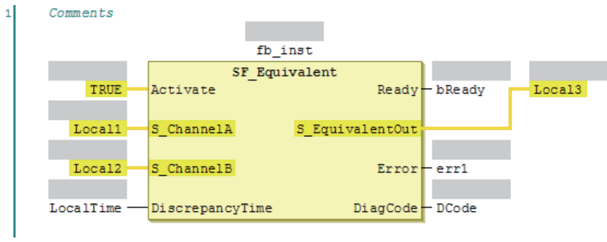
If you drop the *SF_Antivalent* function block on the network where the words *Start here* are displayed, the function block is inserted in the new network.



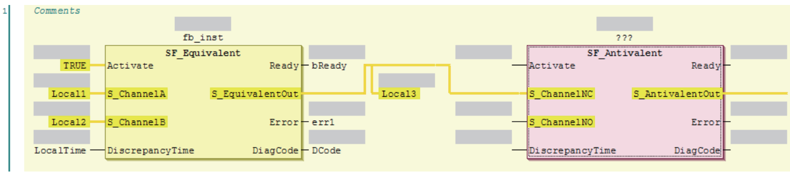
If you drop the *SF_Antivalent* function block on an input terminal, the function block is inserted before the input terminal.



- If you select a function block (in this example, *SF_Antivalent*) from the right-click menu, the FB is inserted at the location shown below.
  - Before the FB Is Inserted



  - When FB *SF_Antivalent* Was Selected from the Right-click Menu



Deleting a FB or FUN Instruction

Use one of the following procedures to delete a FB or FUN instruction.

Method 1: Right-click the FB on the FBD network and select ***Delete*** from the menu.

Method 2: Select the FB on the FBD network and press the **Delete** Key.

Replacing a FB or FUN Instruction

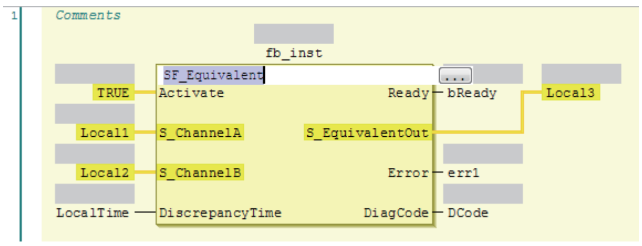
You can replace a FB or FUN instruction with a different instruction without changing the input and output parameters.

Use one of the following procedures.

Method 1: Drag the FB to change to in the Toolbox to the FB on the FBD editor.

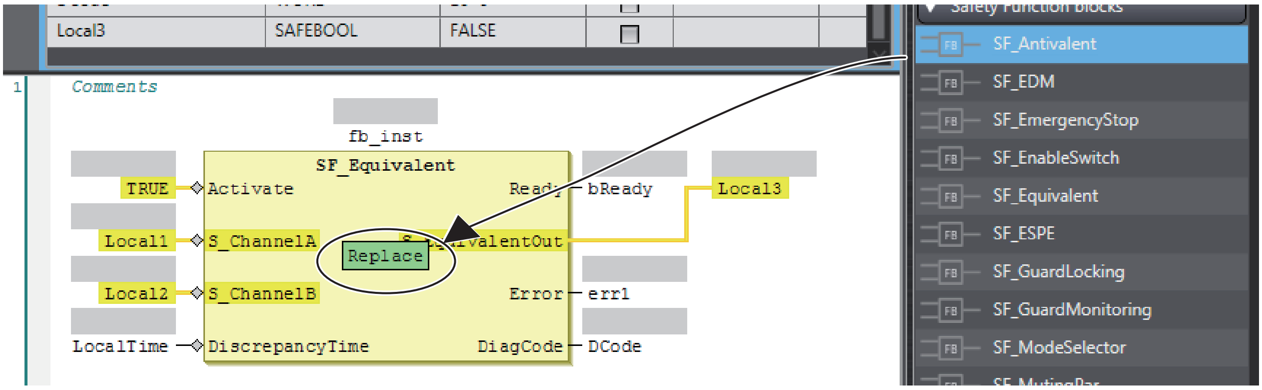
Method 2: Select the function block name on the FBD network and directly enter the function block name.

- Before the Function Block Is Edited



● Procedure for Method 1

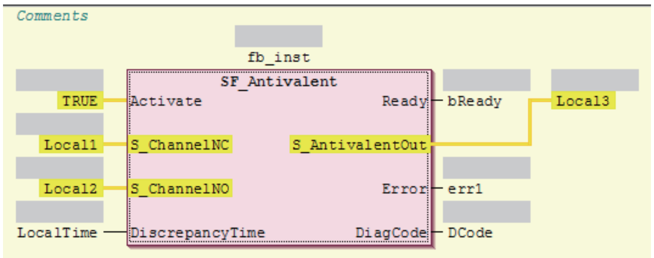
1 Drag a FB from the Toolbox to the FB to replace. A **Replace** area is displayed.



2 Drop the FB in the **Replace** area to replace the FB.

● Procedure for Method 2

- If you directly enter the function block name, the function block is replaced when you press the **Enter** Key.



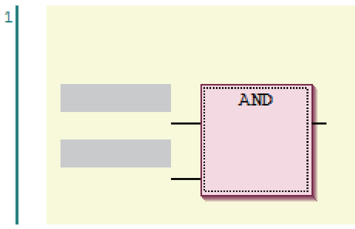
Adding Input Parameters to AND, OR, ADD, MUL, and MUX

You can add input parameters to the AND, OR, ADD, MUL, and MUX instructions. Use one of the following procedures.

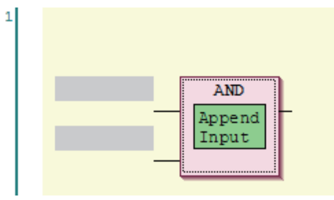
Method 1: Drag *Input* from the General in the Toolbox to the **Append Input** area in the instruction on the FBD editor.

Method 2: Right-click the FB on the FBD network and select ***Add Input*** from the menu.

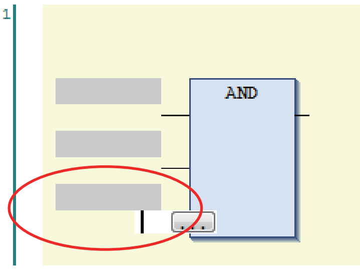
- Before the Input Parameter Is Added



- Dragging an Input Variable from the Toolbox



- After the Input Parameter Is Added



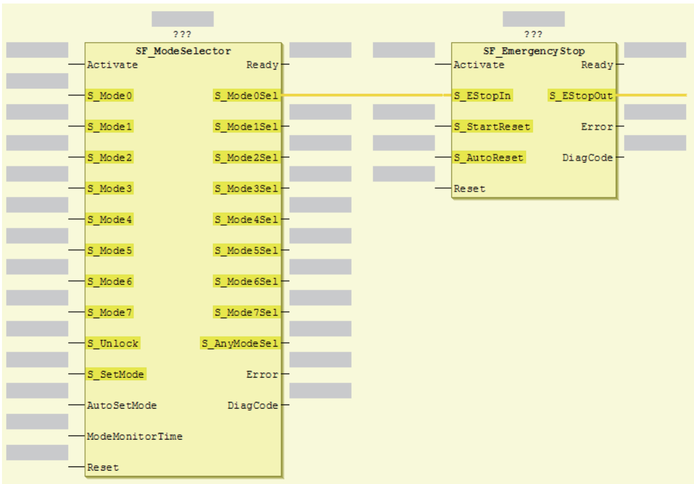
Changing the Output Terminals of a FB or FUN Instruction

Use the following procedure to change the output terminals of a FB or FUN instruction.

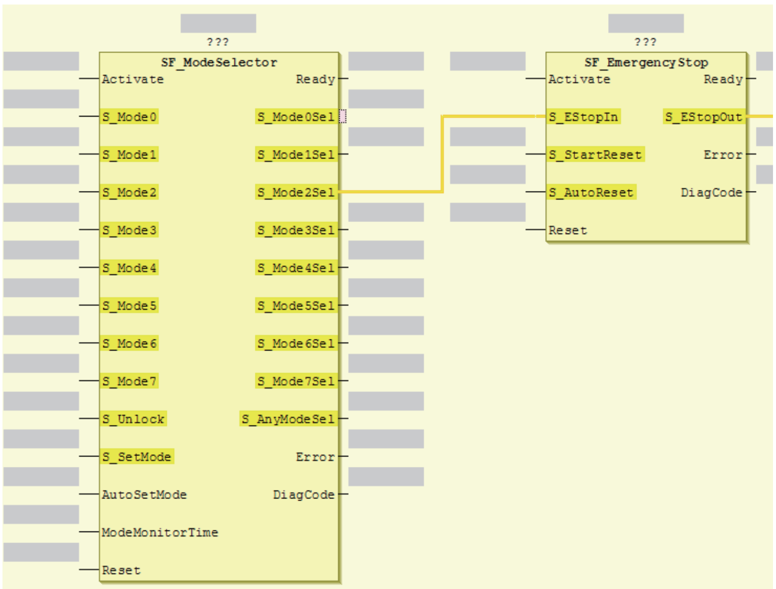
Method: Right-click an output terminal on a FB on the FBD network and select ***Set Output Connection*** from the menu.

The selected output terminal is connected to the input terminal of the next function block.

- Before the Output Terminal Is Changed



- When *S_Mode2Sel* Was Selected and *Set Output Connection* was Executed



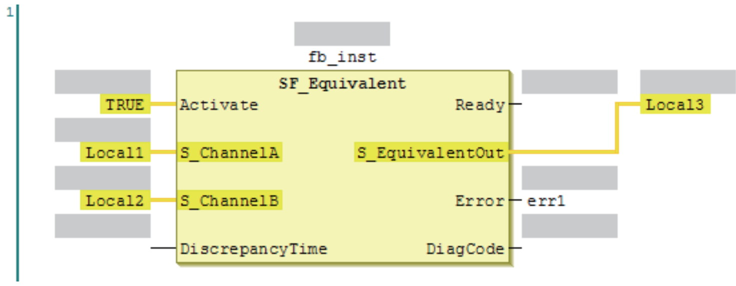
Deleting Unused Parameters from a FB or FUN instruction

Use the following procedure to delete any unused parameters from a FB or FUN instruction.

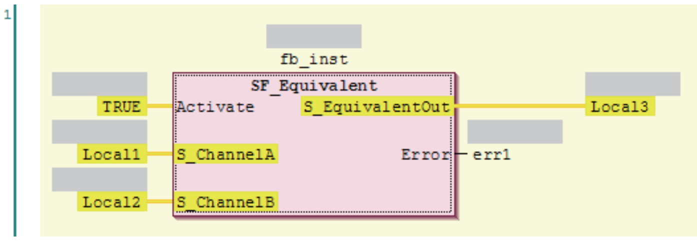
Method: Right-click the FB on the FBD network and select ***Remove unused FB call parameters*** from the menu.

All of the unused parameters are deleted.

- Before the Unused Parameters of the Function Block Are Deleted



- After the Unused Parameters of the Function Block Are Deleted



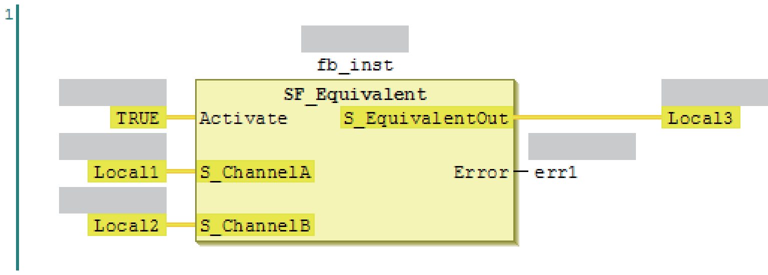
Updating the Input Parameters and Output Parameters of a FB or FUN Instruction

Use the following procedure to display the input parameters and output parameters of a FB or FUN instruction.

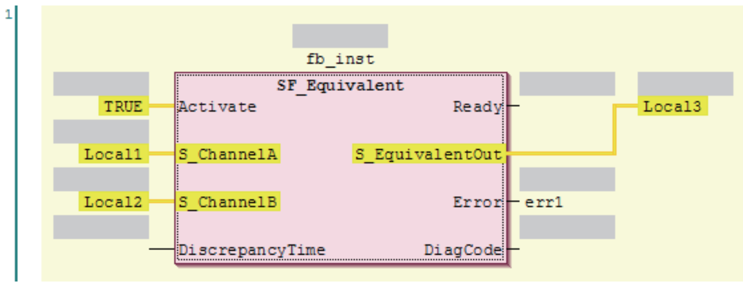
Method: Right-click a FB on the FBD network and select ***Update*** from the menu.

The input parameters and output parameters for the function block are displayed along with any unused parameters.

- Before the Input Parameters and Output Parameters of the Function Block Are Updated



- After the Input Parameters and Output Parameters of the Function Block Are Updated



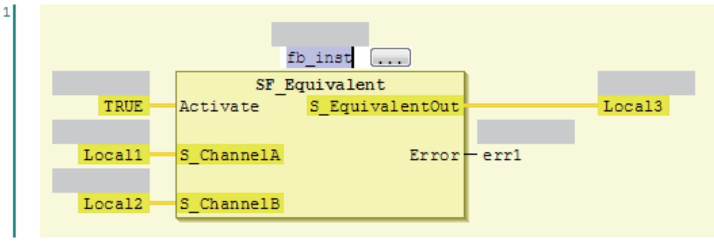
Editing Function Block Instance Variables

Use the one of the following methods to edit existing function block instance variables.

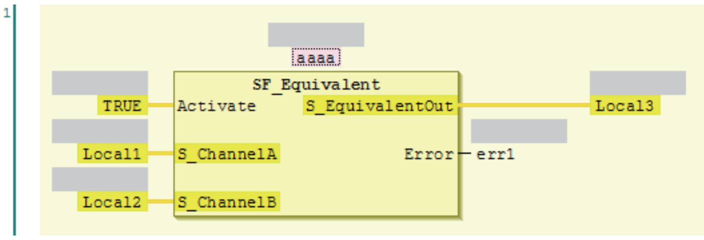
If you specify a variable name that does not exist in the local variable table, that variable will be registered as a local variable.

Method: Select the FB instance variable on the network and directly enter the variable name.

- Before Editing Function Block Instance Variable



- 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.
  - After Editing Function Block Instance Variables



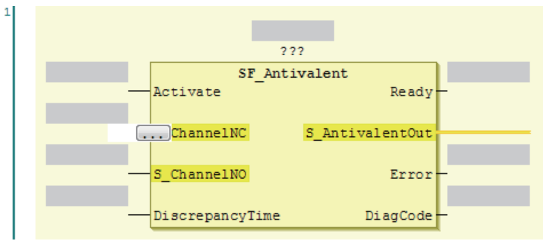
Entering Parameters

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.

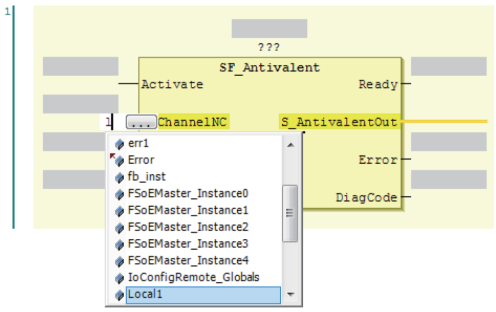
Method: Select the parameter on the FBD network and directly enter the variable name.

- Before the Parameter Is Edited

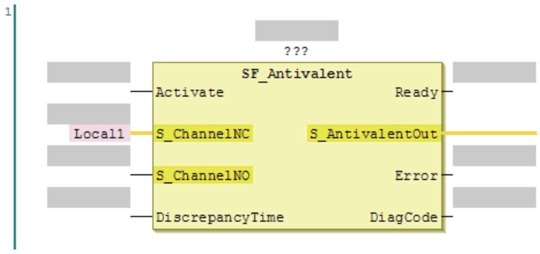


- If you enter the variable name directly, a list of variable names that you can select from is displayed when you enter the first letter of the variable name. Use the **Up** and **Down** Keys to select the variable name from the list, and then press the **Enter** Key. The selected variable name is registered as an input parameter. If you press the **Ctrl + Space** Keys when nothing is displayed, list of variable name candidates is displayed.

- 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.



- Before the Input Parameter Is Edited



To delete an input parameter assignment, select the parameter and press the **Delete** Key.

Additional Information

You can click the Input Assistant 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.

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 instruction.

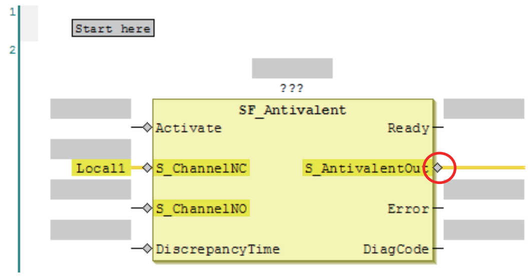
Method 1: Drag *Assignment* from the General in the Toolbox to a terminal in the FBD editor, or to a new FBD network where the words *Start here* are displayed.

Method 2: Right-click the FBD network and select ***Insert Assignment*** from the menu.

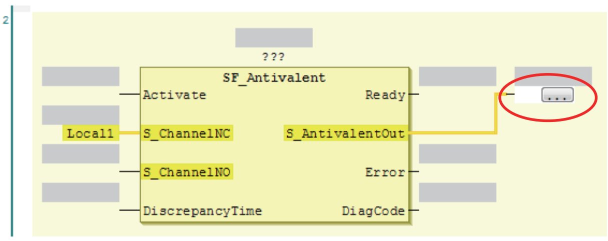
- If you select a network, the output variable is added to the last output area (before the output terminal or the output variable) on the network.
- If you drag *Assignment* from the Toolbox, the point of insertion depends on where you drop the output variable.

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.

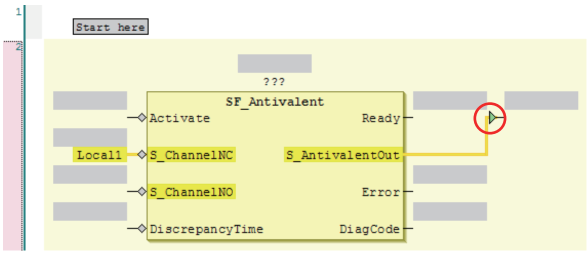
- Example 1 of a Dragged Output Parameter Object



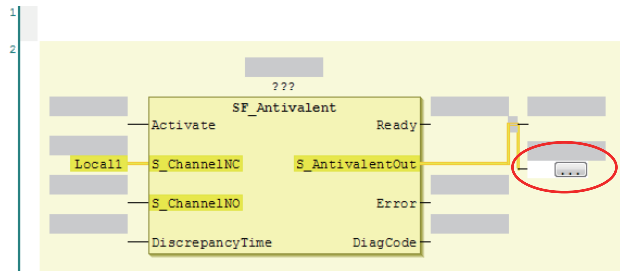
Output Parameter Is Added to the Output Terminal



- Example 2 of a Dragged Output Parameter Object



Output Parameters Is Added before the Output Parameter



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.

Method: Right-click the output parameter on the FBD network and select ***Delete*** from the menu.

- The selected output parameter is deleted. You cannot select multiple parameters.

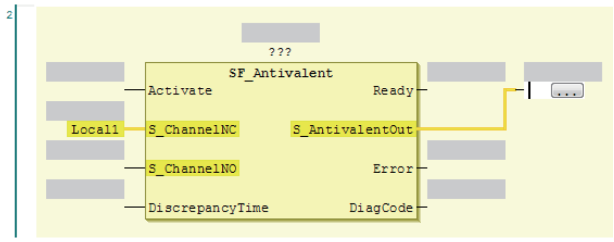
Editing Output Parameters

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.

Method: Select the output variable on the FBD network and directly enter the variable name.

- Before the Output Variable Is Edited



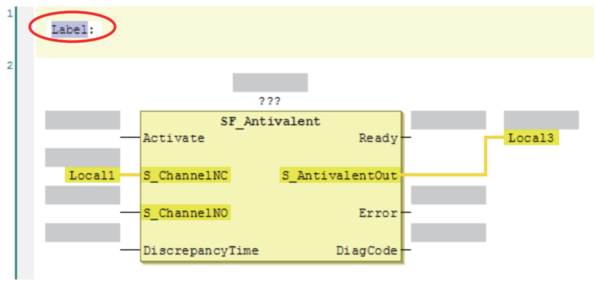
- If you prefer to enter the variable name directly, a list of variable names that you can select from is displayed when you begin entering the variable name.
- To create a new output variable, enter the variable name and press the **Enter** Key. The output variable is registered, and it is also registered as a local variable in the local variable table.

Inserting Jump Labels

Use the following procedure to insert jump labels into an FBD network.

Method: Right-click the FBD network and select ***Insert Jump Label*** from the menu.

- After the Jump Label Is Added



- You cannot add another jump label to a network if it already has one.

Deleting Jump Labels

Use the following procedure to delete jump labels.

Method 1: Right-click the jump label and select ***Delete*** from the menu.

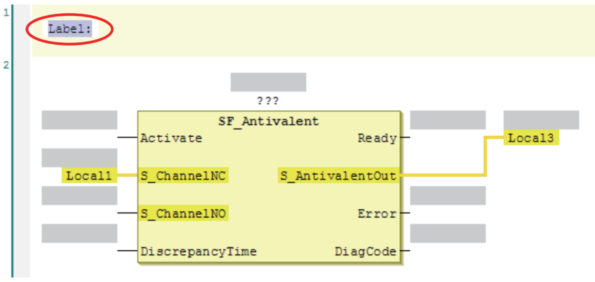
Method 2: Select the jump label and press the **Delete** Key.

Editing Jump Labels

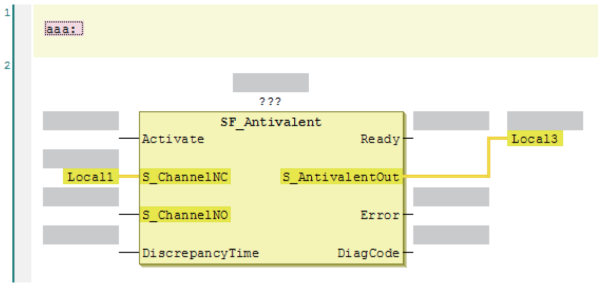
Use the following procedure to edit jump labels.

Method: Select the 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.
  - Selecting Jump Labels



  - After Jump Label Is Edited



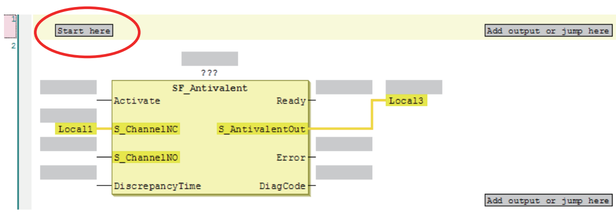
Inserting Jump Instructions

Use one of the following procedures to insert a Jump instruction in a network.

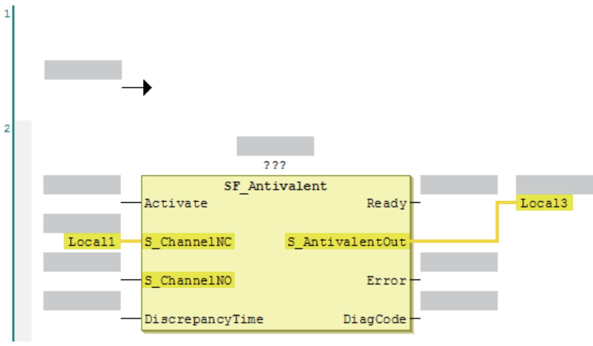
Method 1: Drag *Jump* from the Toolbox to the words *Start here* on a new network.

Method 2: Right-click the FBD network and select ***Insert Jump*** from the menu.

- The Jump instruction is inserted in the network.
  - Before Jump Instruction Is Dropped



  - After Jump Instruction Is Inserted



Deleting Jump Instructions

Use one of the following procedures to delete Jump instructions.

Method 1: Right-click the Jump instruction and select ***Delete*** from the menu.

Method 2: Select the Jump instruction and press the **Delete** Key.

- The selected Jump instruction is deleted.

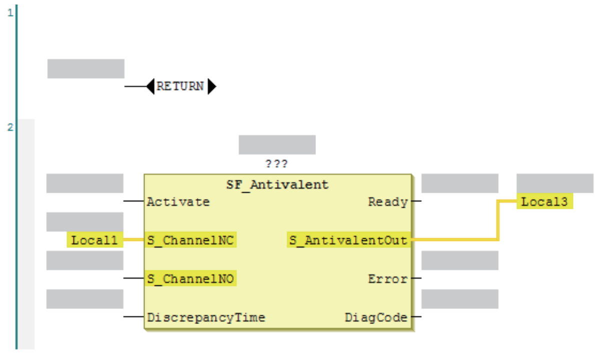
Inserting Return Instructions

Use one of the following procedures to insert a Return instruction in a network.

Method 1: Drag *Return* from the FBD Tools in the Toolbox to a terminal on the FBD editor or to where the words *Start here* are displayed on a new network.

Method 2: Right-click the FBD network and select ***Insert Return*** from the menu.

- The Return instruction is inserted in the network.
  - After Inserting Return Instruction



Deleting Return Instructions

Use one of the following procedures to delete a Return instruction from a network.

Method 1: Right-click the Return instruction on the network and select ***Delete*** from the menu.

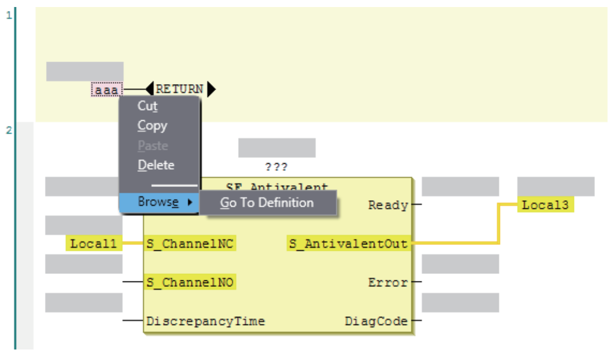
Method 2: Select the Return instruction on the network and press the **Delete** Key.

- The selected Return instruction is deleted.

Viewing the Locations of Variable Definitions

You can view where variables are defined. Use the following procedure.

Method 1: Right-click the variable and click ***Browse − Go To Definition*** from the menu.

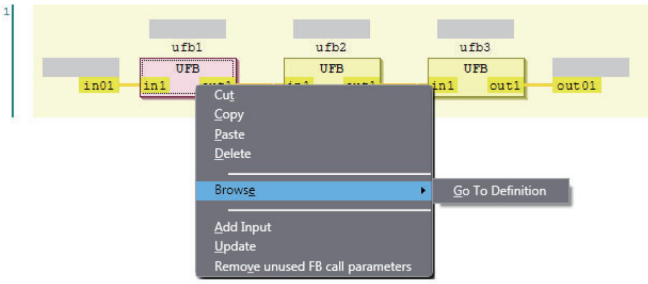


- The focus moves to the location where the variable is defined (local variable).

Viewing the Locations of User-defined Function Block Definitions

You can view where user-defined function blocks are defined. Use the following procedure.

Method 1: Right-click the user-defined function block and click ***Browse*** − ***Go To Definition*** from the menu.



- The focus moves to where the user-defined function block is defined.

7-5-5 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

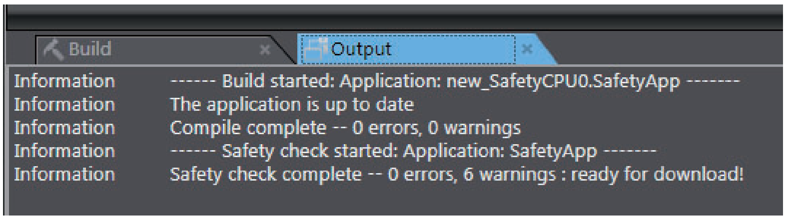
1 Use one of the following procedures to execute the build process.

Method 1: Select ***Build Controller*** from the Project Menu.

Method 2: Press the **F8** Key.

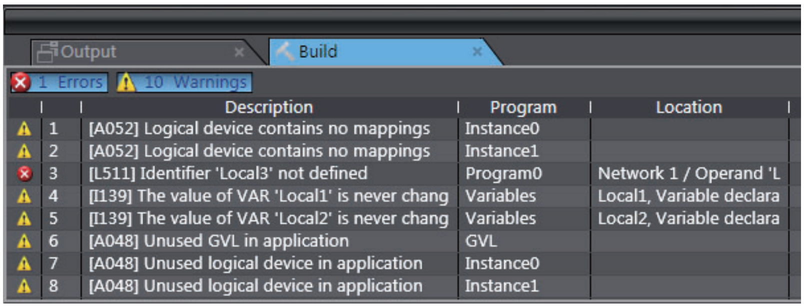
Method 3: Click the **Build Controller** Button on the toolbar.

The build is started and the status during the build is displayed in the Output Tab Page.

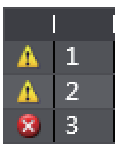
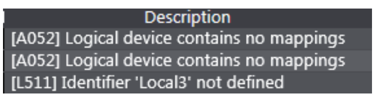
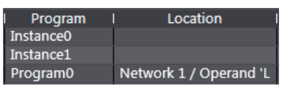


2 Click the **Build** Tab. The Build Tab Page is displayed.

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	Meaning
Number of errors		Displays the total number of errors.
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. Reduce the number of FBs or variables that are used.

7-5-6 Searching and Replacing

You can search 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

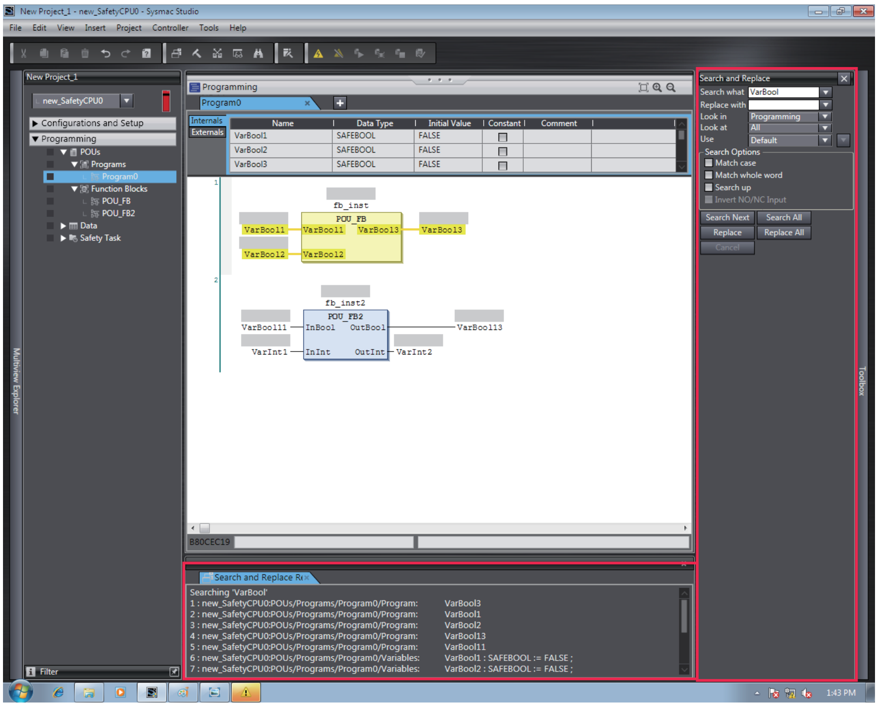
1 Use one of the following procedures to display the Search and Replace Pane in place of the Toolbox.

Method 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.

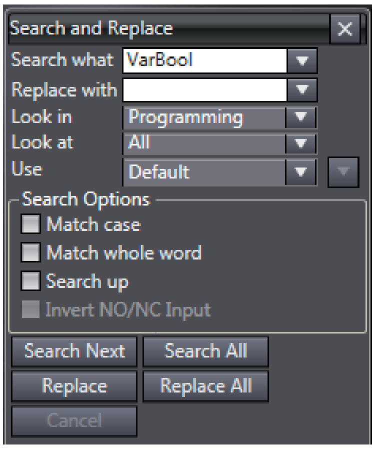


2 In the Search and Replace Pane, enter the text string and set the applicable search conditions, and then click the **Search** or **Replace** Button.

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.



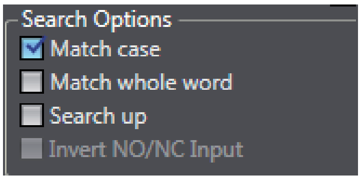
Item	Description
Search what	• Enter a search string. • You can select from previous search strings in the list.
Replace with	• Enter the string to replace the search string with. • You can select from previous replacement strings in the list. You cannot use wildcard characters.
Look in	Specify the range to search. You can select from the following. *Programming:* The search is performed on the program to which the selected element belongs when the search is executed. If the search is made when the Safety CPU Unit is selected as the Controller, the search is performed only for the program in the Safety CPU Unit. *Current view:* The current view is searched.
Look at	Specify the items to search. You can perform a search on the following items. *All:* Variable tables (variable names in the table) and programs (jump labels, Jump instruction names, and variable names in the programs) *Variable name:* Searches all variable names. *Instruction*: Function block instance names
Use	Specify if you want to use wildcard[*1] characters. *Default:* Do not use wildcard characters. *Wildcard:* Use wildcard characters. 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. 

*1. The characters that you can use as wildcard characters are given on the next page.

● Wildcards

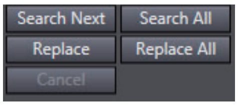
Character	Meaning	Description	Example
*	Zero or more characters	Searches for a text string that contains a variable text string	“new*” matches “newfile.txt”.
?	Any single character	Searches for a text string with a variable character.	“A?C” matches “ABC”, “AdC”, and “AzC”.
#	Any single number	Searches for any single number.	“7#” matches “71”. “ABC#” matches “ABC5”.
[ ]	Character in a set	Searches for a single character in the set.	“ABC[xyz]” matches “ABCx” and “ABCy”. “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”. "ABC[!x-z]" matches "ABCa" and "ABCd".

● Search Options



Item	Description
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



Button	Description
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.

7-5-7 Safety Task Settings

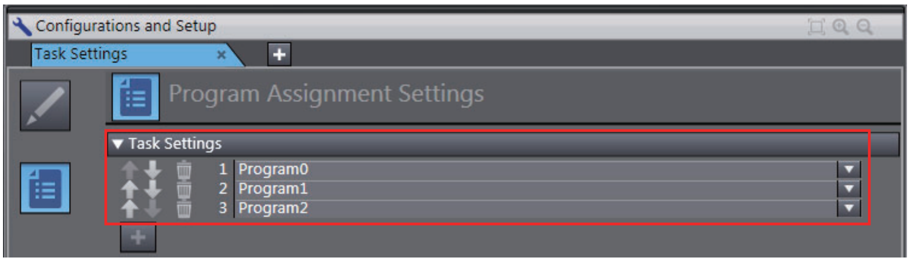
This section describes the procedures that are used to select the programs to execute in the safety task and the execution order of the selected programs. It also describes how to set the task period of the safety task.

Refer to *4-2 Safety Task* on page 4-4 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 Task Settings Tab Page.
3. The buttons shown within the red frame below allow you to change the program assignments and their execution order.



Precautions for Correct Use

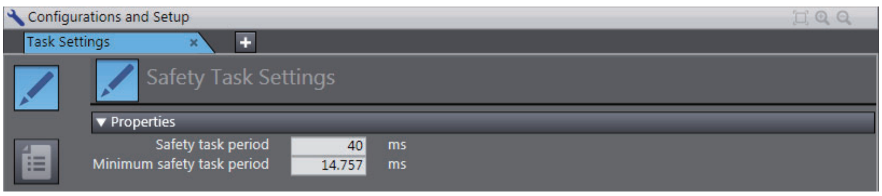
Any program you assign must already be registered under **Programming - POUs**.

Setting the Task Period

1. Double-click **Task Settings** under **Configurations and Setup** in the Multiview Explorer.

   The Task Settings Tab Page is displayed in the Edit Pane.

2. If the Cycle Time Settings Display does not appear on the Task Settings Tab Page in the Edit Pane, click the **Safety Task Settings** Button ( ).



*1. The minimum value of the task period is automatically calculated and displayed based on the program and settings information.

3. Set the task period for the safety task.

8

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.

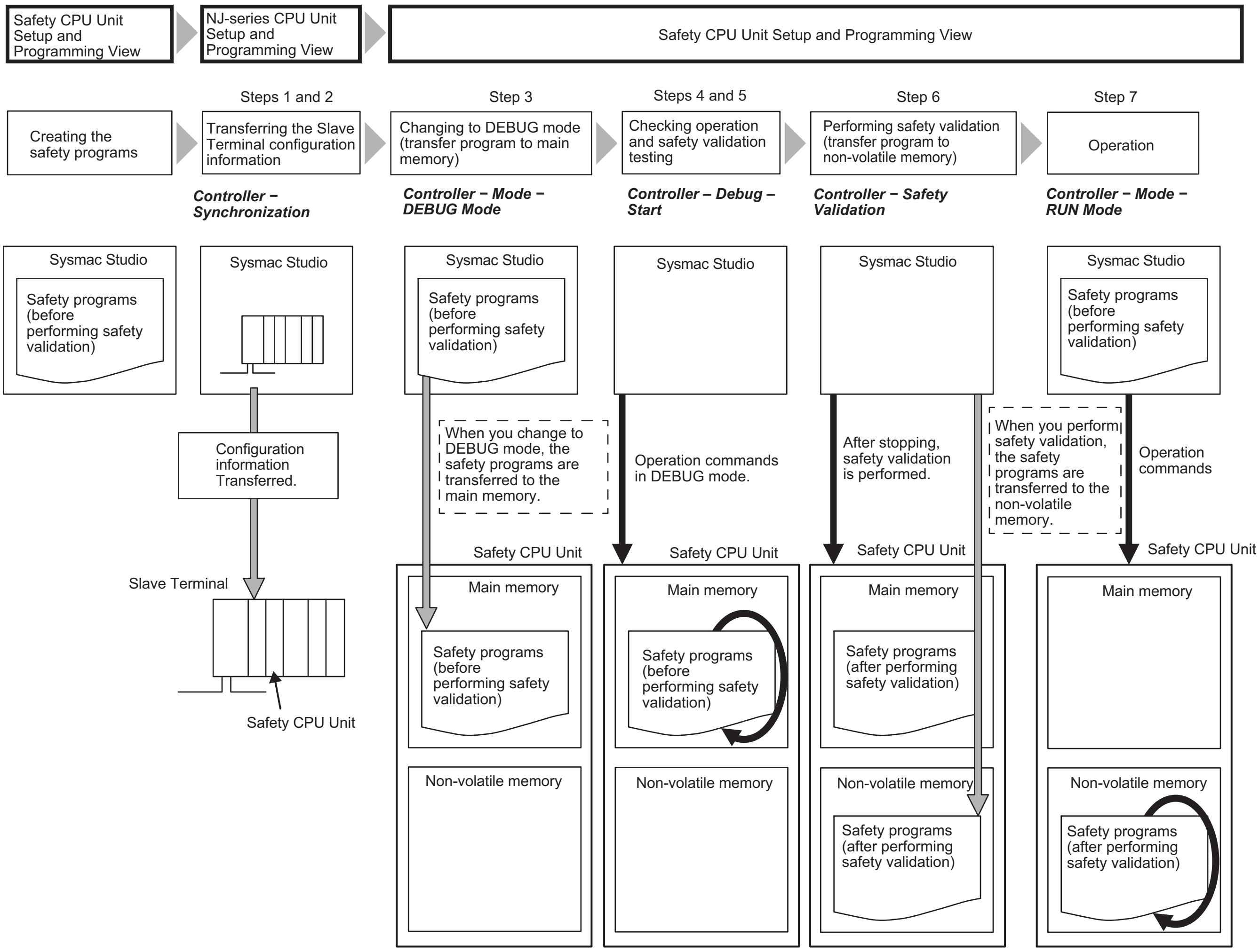
8-1 Procedures before Operation and Transferring the Required Data

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.

8-1-1 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 the NJ-series CPU Unit or the EtherCAT Coupler Unit.
2. Transfer the configuration information to the NJ-series CPU Unit, EtherCAT Coupler Unit, and Safety CPU 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.


Safety CPU Unit Setup and Programming View
NJ-series CPU Unit Setup and Programming View
Safety CPU Unit Setup and Programming View
Creating the safety programs
Steps 1 and 2
Transferring the Slave Terminal configuration information
Step 3
Changing to DEBUG mode (transfer program to main memory)
Steps 4 and 5
Checking operation and safety validation testing
Step 6
Performing safety validation (transfer program to non-volatile memory)
Step 7
Operation
Controller − Synchronization
Controller − Mode − DEBUG Mode
Controller − Debug − Start
Controller − Safety Validation
Controller − Mode − RUN Mode
Sysmac Studio
Safety programs (before performing safety validation)
Configuration information Transferred.
Slave Terminal
Safety CPU Unit
When you change to DEBUG mode, the safety programs are transferred to the main memory.
Operation commands in DEBUG mode.
After stopping, safety validation is performed.
When you perform safety validation, the safety programs are transferred to the non-volatile memory.
Operation commands
Main memory
Non-volatile memory
Safety programs (after performing safety validation)

8-1-2 Data That You Must Transfer before Operation and Data Transfer Procedures

Before you change to RUN mode, you must transfer the Slave Terminal settings. The settings and transfer procedures are given below.

<table>
<tr><th colspan="2" rowspan="2">Slave Terminal settings</th><th colspan="2">Transfer method</th><th rowspan="2">Transfer destination</th></tr>
<tr><th>Connection to an NJ-series CPU Unit</th><th>Connection to USB port on EtherCAT Coupler Unit*1</th></tr>
<tr><td>EtherCAT network configuration information</td><td></td><td>Perform the following operation with the Controller set to the NJ-series CPU Unit. Transfer the EtherCAT network configuration information from the Synchronization Window.</td><td>Transfer is not allowed.</td><td>NJ-series CPU Unit</td></tr>
<tr><td rowspan="2">Slave Terminal configuration information</td><td>Unit configuration information</td><td rowspan="2">Perform the following operation with the Controller set to the NJ-series CPU Unit. Transfer the EtherCAT network configuration information and the Slave Terminal configuration information from the Synchronization Window.</td><td rowspan="2">Transfer the data from the EtherCAT Slave Terminal Tab Page with the Controller set to the NJ-series CPU Unit.</td><td>EtherCAT Coupler Unit</td></tr>
<tr><td>I/O allocation information</td><td>EtherCAT Coupler Unit and Safety CPU Unit</td></tr>
<tr><td rowspan="4">Safety application data</td><td>Unvalidated safety programs</td><td colspan="2" rowspan="2">The safety application data is transferred automatically when you switch to DEBUG mode with the Controller set to the Safety CPU Unit.</td><td rowspan="4">Safety CPU Unit</td></tr>
<tr><td>Safety task settings and variables for the unvalidated safety programs</td></tr>
<tr><td>Validated safety programs</td><td colspan="2" rowspan="2">In DEBUG mode, execute Safety Validation with the Controller set to the Safety CPU Unit. This will cause the safety application data to be saved to the non-volatile memory.</td></tr>
<tr><td>Safety task settings and variables for the validated safety programs</td></tr>
</table>

*1. In the following cases, only the EtherCAT Slave Terminal is restarted after the data is transferred.

- When you transfer the data through the NJ-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 EtherCAT 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.

8-2 Transferring the Configuration Information

This section describes how to start communications and transfer configuration information from the Sysmac Studio to the NJ-series CPU Unit, the NX-series EtherCAT Coupler Unit, and the NX-series Safety CPU Unit.

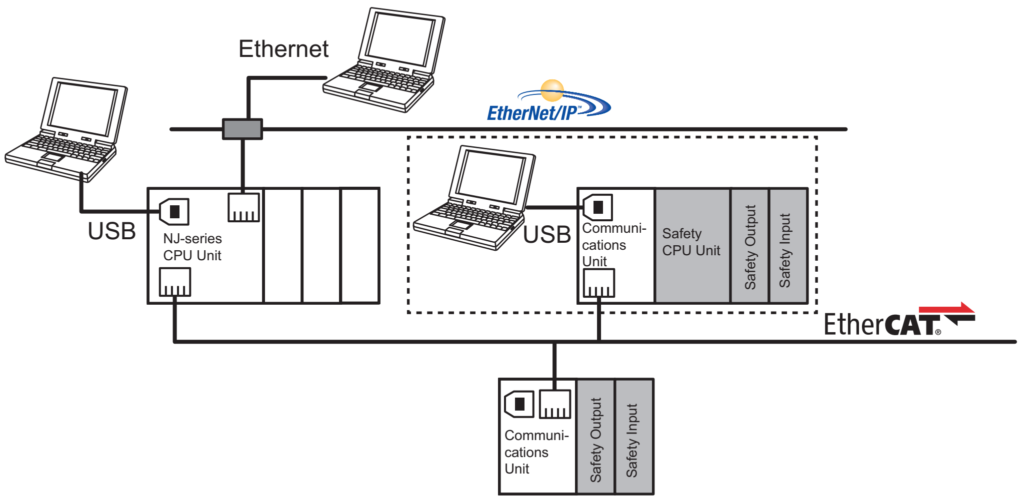
8-2-1 Overview

You must transfer the Slave Terminal settings to the NJ-series CPU Unit, the EtherCAT Coupler Unit, and the Safety CPU Unit before you can place the Sysmac Studio online with the Safety CPU Unit and begin debugging.

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-series CPU Unit
- USB connection to the EtherCAT Coupler Unit



⚠ WARNING

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

8-2-2 Transferring Configuration Information to an NJ-series CPU Unit over a USB Connection or Ethernet Connection

If the Sysmac Studio is connected to the USB port or Ethernet port on the NJ-series CPU Unit, use the following procedure to go online with the Safety CPU Unit.

1. Select the NJ-series CPU Unit from the Controller Selection Box in the Multiview Explorer on the Sysmac Studio to change to the view for the NJ-series CPU Unit.
2. Set the communications path to the NJ-series CPU Unit.
3. Select ***Online*** from the Controller Menu. Or, click the **Online** Button ( ) in the toolbar.
4. Select ***Synchronization*** from the Controller Menu.
5. Click the **Transfer to Controller** Button.*1

   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.

Additional Information

- Refer to the *Sysmac Studio Version 1 Operation Manual* (Cat. No. W504) for information on connecting and synchronizing with the NJ-series CPU Unit.
- Refer to *8-4 Changing to DEBUG Mode* on page 8-15 for details on transferring the unvalidated safety program.

*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.

8-2-3 Transferring Configuration Information to the EtherCAT Coupler Unit When Connected to the USB Port

If the Sysmac Studio is connected to the USB port on the EtherCAT Coupler Unit, use the following procedure to go online with the Safety CPU Unit.

1 Connect the computer to the EtherCAT Coupler Unit with a USB cable.

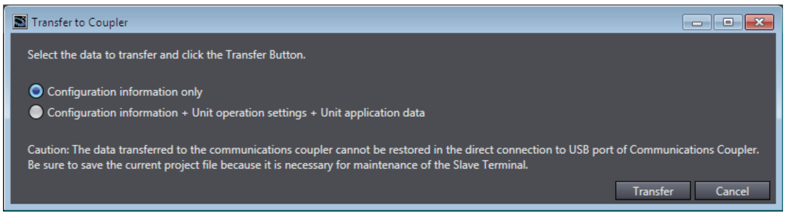
2 Select the NJ-series CPU Unit from the Controller Selection Box in the Multiview Explorer on the Sysmac Studio to change to the view for the NJ-series CPU Unit.

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.

4 Right-click the EtherCAT Coupler Unit and select ***Transfer to Coupler*** from the menu.

5 The following dialog box is displayed.



In this case, the following settings are ignored because the safety programs are not validated. For both options, only the Slave Terminal configuration information is transferred to the EtherCAT Coupler Unit and the Safety CPU Unit.

- Configuration information only: Only the Slave Terminal configuration information is transferred.
- Configuration information + Unit operation settings + Unit application data: All Slave Terminal settings are transferred.

6 Click the **Transfer** Button to transfer the settings to all Units that are connected to the EtherCAT Coupler Unit.

8-3 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.

8-3-1 Startup Operating Mode and Changing the Operating Mode

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.

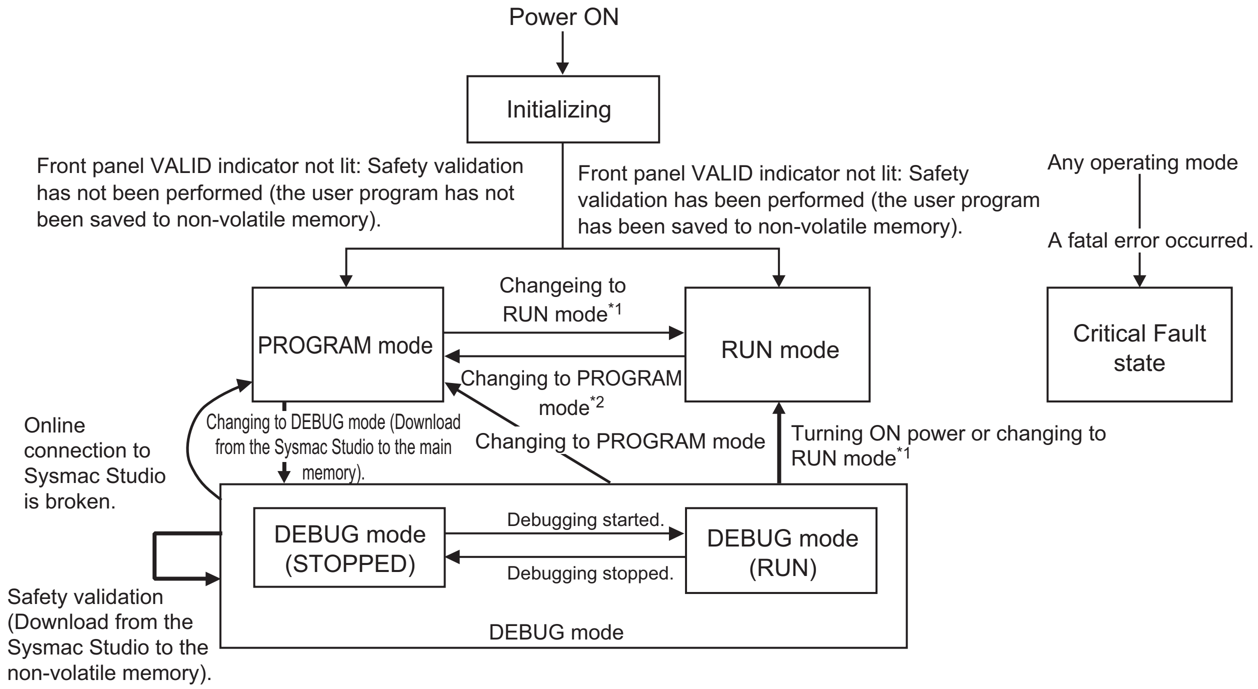
● Before Safety Validation

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 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 RUN mode to PROGRAM mode, the user program in the non-volatile memory of the Safety CPU Unit is deleted. If you return to RUN mode, the safety-validated user program is downloaded to the non-volatile memory in the Safety CPU Unit again.

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. Be aware that if you change from RUN mode to PROGRAM mode, the safety program that was downloaded to the Safety CPU Unit is deleted. Afterwards, change to DEBUG mode. When you change to DEBUG mode, the safety program is transferred from the Sysmac Studio to the Safety CPU Unit.

Operating Modes and Indicators on the Safety CPU Unit

This section describes the operating modes of the Safety CPU Unit and the indicator display pattern for each operating mode.

<table>
<tr><th colspan="2" rowspan="2">Operating mode</th><th rowspan="2">Description</th><th colspan="5">Indicator</th></tr>
<tr><th>TS</th><th>FS</th><th>RUN</th><th>VALID</th><th>DEBUG</th></tr>
<tr><td colspan="2">Initializing</td><td>This mode indicates that the Safety CPU Unit is performing startup processing and hardware self-diagnosis.</td><td>Flashing green.</td><td>Not lit.</td><td>Flashing green.</td><td>Not lit.</td><td>Not lit.</td></tr>
<tr><td colspan="2">PROGRAM mode</td><td>This mode indicates that execution of the safety program is stopped.
You can clear or download the safety application data from the Sysmac Studio (With the Controller set to the Safety CPU Unit).</td><td rowspan="4">Lit green or flashing red.</td><td>Flashing green, flashing red and green, or flashing red.</td><td>Not lit.</td><td>Not lit or lit yellow.</td><td>Not lit.</td></tr>
<tr><td colspan="2">RUN mode*1</td><td>This mode indicates that execution of the validated safety programs is in progress.</td><td rowspan="3">Flashing green, lit green, or flashing red.</td><td>Lit green.</td><td>Lit yellow.</td><td>Not lit.</td></tr>
<tr><td rowspan="2">DEBUG mode</td><td>STOPPED</td><td>This mode indicates that execution of the unvalidated safety programs is on standby.</td><td>Not lit.</td><td rowspan="2">Not lit or lit yellow.</td><td rowspan="2">Lit yellow.</td></tr>
<tr><td>RUN</td><td>This mode indicates that an unvalidated safety program is being executed, and that you can control BOOL variables, change present values of data, and use forced refreshing.
This mode requires that the Sysmac Studio (with the Controller set to the Safety CPU Unit) be online with the Safety CPU Unit because it allows access to debugging functions that can affect the safety of the safety system.
When the online connection is cancelled (intentionally or due to a timeout), the Safety CPU Unit automatically changes to PROGRAM mode.</td><td>Lit green.</td></tr>
<tr><td colspan="2">Critical Fault state</td><td>Operations are fully stopped in this state because a hardware error or other fatal error was found in the Safety CPU Unit.</td><td>Lit red.</td><td>Not lit.</td><td>Not lit.</td><td>Not lit.</td><td>Not lit.</td></tr>
</table>

*1. The TS and FS indicators will flash green when safety process data communications are established after changing to RUN mode.

Relationship between the Operating Modes of the Safety CPU Unit and the NJ-series CPU Unit

The operating mode of the Safety CPU Unit is independent from the operating mode of the NJ-series CPU Unit. This means that changes in the operating mode of the NJ-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-series CPU Unit.

8-3-2 Restrictions in DEBUG Mode

In general, 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.

8-3-3 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 started.
PROGRAM mode	→	RUN mode	• Safety process data communications started. • The variables are initialized and the safety programs are executed from the beginning.
DEBUG mode (STOPPED)	→	DEBUG mode (RUN)	• Safety process data communications continue. • The variables are initialized and the safety programs are executed from the beginning.
DEBUG mode (RUN)	→	DEBUG mode (STOPPED)	• Safety process data communications continue. • The variables are initialized and the safety programs are stopped. • The forced status of variables is cleared.
RUN mode	→	PROGRAM mode	• Safety process data communications stop. The safety input data from the Safety Input Units is initialized to 0. The safety output data to the Safety Output Units is initialized to 0. • The variables are initialized and the safety programs are stopped. • The safety programs are deleted.
DEBUG mode (RUN)	→	PROGRAM mode	• Safety process data communications stop. The safety input data from the Safety Input Units is initialized to 0. The safety output data to the Safety Output Units is initialized to 0. • The variables are initialized and the safety programs are stopped. • The forced status of variables is cleared.
DEBUG mode (STOPPED)	→	PROGRAM mode	• Safety process data communications stop. The safety input data from the Safety Input Units is initialized to 0. The safety output data to the Safety Output Units is initialized to 0.

Before change	→	After change	Operation
Any operating mode	→	Critical Fault state	• Safety process data communications stop. • The safety programs are stopped. • Message communications are stopped. • All safety output data 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.

8-3-4 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: STOPPED	DEBUG mode: RUN	RUN mode	Critical Fault state
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. *2	Possible.	Possible.	Possible.	Possible.	Not possible.
NX bus communications		Possible. *3	Possible. *3	Possible. *3	Possible. *3	Possible. *3	Not possible.*3
Safety communications		Not possible.	Not possible.	Possible. *4	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			Not possible.	Not possible.	Not possible.	
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 Units		Not possible.	Possible.	Not possible.	Not possible.	Not possible.	Not possible.
Clear All Memory operation for a Slave Terminal*5		Not possible.	Not possible.	Not possible.	Not possible.	Not possible.	Not possible.
Restarting Slave Terminals		Possible.	Possible.	Possible.	Possible.	Possible.	Possible.
Monitoring Controller status		Not possible.	Possible.	Possible.	Possible.	Possible.	Not 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.	Not possible.
Changing the safety password		Not possible.	Possible.	Not possible.	Not possible.	Not possible.	Not possible.

*1. Hardware Self-diagnosis

In the initializing state, self-diagnosis is performed for all hardware.

Self-diagnosis for hardware is not performed in the Critical Fault state.

In other operating modes, hardware self-diagnosis is performed at fixed intervals.

*2. Message Communications

Message communications are enabled 0.5 seconds after the power is turned ON.

*3. NX Bus Communications

NX bus communications is enabled 0.5 seconds after the power is turned ON. However, the data that is refreshed depends on the operating mode, as shown in the following table.

<table>
<tr><th colspan="2">Operating mode</th><th>Refreshing</th></tr>
<tr><td colspan="2">Initializing</td><td>The input data is discarded and the output data is fixed to 0.</td></tr>
<tr><td colspan="2">PROGRAM mode</td><td>The input data is discarded and the output data that carries status information is transferred as raw data. All data outputs from the safety programs are fixed to 0.</td></tr>
<tr><td colspan="2">RUN mode</td><td>I/O data refreshing is performed with the safety programs.</td></tr>
<tr><td rowspan="2">DEBUG mode</td><td>STOPPED</td><td>Input data is refreshed in the safety programs as raw data.
The output data that carries status information is transferred as raw data. All data outputs from the safety programs are fixed to 0.</td></tr>
<tr><td>RUN</td><td>I/O data refreshing is performed with the safety programs.</td></tr>
<tr><td colspan="2">Critical Fault state</td><td>NX bus communications cannot be executed.</td></tr>
</table>

*4. The safety I/O connection is established, but the output data for communications is fixed to 0. The input data is raw data.

*5. You can execute the Clear All Memory operation for the Slave Terminal regardless of the operating status of the Safety CPU Unit, but it will always fail for the Safety CPU Unit.

8-4 Changing to DEBUG Mode

This section describes how the safety application data is transferred to the Safety CPU Unit when you place the Safety CPU Unit in DEBUG mode.

⚠ 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

DEBUG mode is used to check that the safety programs and the external devices operate properly before you operate the Safety CPU Unit.

You operate the Safety CPU Unit through an online connection with the Safety CPU Unit set as the Controller on the Sysmac Studio.

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.

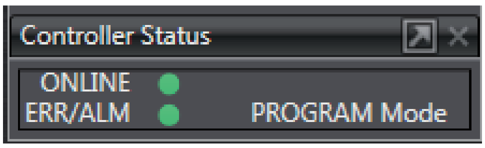
As a safety precaution, the Safety CPU Unit must be in PROGRAM mode for you to be able to place it in DEBUG mode.

Procedure for Changing to DEBUG Mode

Use the following procedure to change the operating mode of the Safety CPU Unit from PROGRAM mode to DEBUG mode. When you change to DEBUG mode, the safety programs must be ready for building.

1. Make sure the Safety CPU Unit is in PROGRAM mode.
2. Go online with the Safety CPU Unit.
3. Select the Safety CPU Unit from the Controller Selection Box in the Multiview Explorer on the Sysmac Studio to change to the Safety CPU Unit Setup and Programming View.

   When you change to the view for the Safety CPU Unit, the Sysmac Studio goes online with the Safety CPU Unit and the Controller status is displayed as shown below in the lower right of the Sysmac Studio Window.

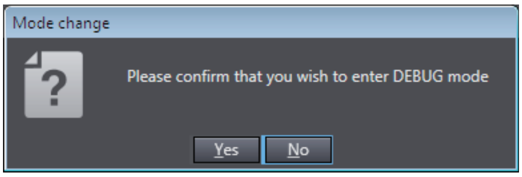
   From this point on, the Sysmac Studio changes to DEBUG mode while it is online with the Safety CPU Unit. When you change to DEBUG mode, the safety programs are automatically transferred to the main memory of the Safety CPU Unit. Perform debugging after this transfer is completed.

   Refer to *8-4 Changing to DEBUG Mode* on page 8-15 for details.

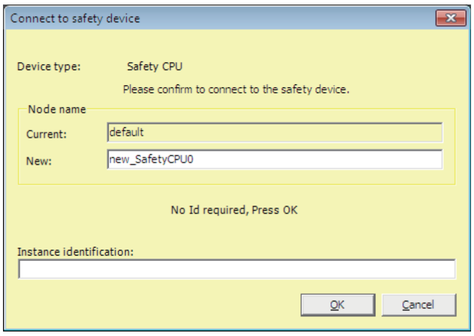
4 On the Safety CPU Unit Setup and Programming View, perform one of the following operations.

- 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.

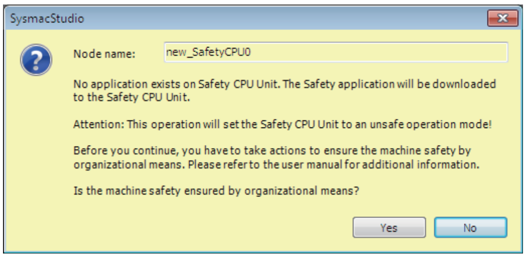


5 Click the **Yes** Button. The following Connect to Safety Device Dialog Box is displayed.



*1. When you change the operating mode for the first time, above dialog box for setting the node name is displayed. You can change the node name that is set by default for 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 *8-6 Node name* on page 8-32 for details on the node name setting. You do not need to enter anything in the **Device 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 device confirmation dialog box is displayed.

8 The first time you use DEBUG mode, or if the security password has not been set, leave the *Password* Box blank and click the **OK** Button.

If the security password is set, enter the security password and click the **OK** Button.

Refer to *8-7 Setting the Safety Password* on page 8-33 for the procedure to set the security password.

The unvalidated safety programs are transferred from the Sysmac Studio to the main memory in the Safety CPU Unit, and the Safety CPU Unit enters the DEBUG mode (STOPPED).

Precautions for Correct Use

Before safety is validated, the safety programs are stored in the main memory of the Safety CPU Unit. When the Sysmac Studio is taken offline or the power supply to the Safety CPU Unit is turned OFF in this state, 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.

1 With the Safety CPU Unit in DEBUG mode, 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.

The Safety CPU Unit enters PROGRAM mode.

8-5 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 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.

8-5-1 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 page
Monitoring	*8-5-3 Monitoring (Displaying the Present Values of Variables)* on page 8-20
Monitoring in a Watch Tab Page	*8-5-4 Controlling BOOL Variables, Changing Present Values, and Using Forced Refreshing* on page 8-22
Controlling BOOL variables	
Forced refreshing (TRUE/FALSE/Cancel)	
Changing present values of data	
Clear All Memory	*8-11-2 Clear All Memory Operation* on page 8-40
Monitoring Controller status	*8-10 Monitoring Controller Status* on page 8-39
Changing the operating mode	*8-3 Operating Modes of the Safety CPU Unit* on page 8-9
Troubleshooting	*Section 9 Troubleshooting*
Monitoring error information	
Displaying error logs	

Procedures to check operation are performed when online to the Safety CPU Unit in DEBUG mode (RUN) or DEBUG mode (STOPPED).

8-5-2 Starting and Stopping the Safety Programs in DEBUG mode

⚠ WARNING

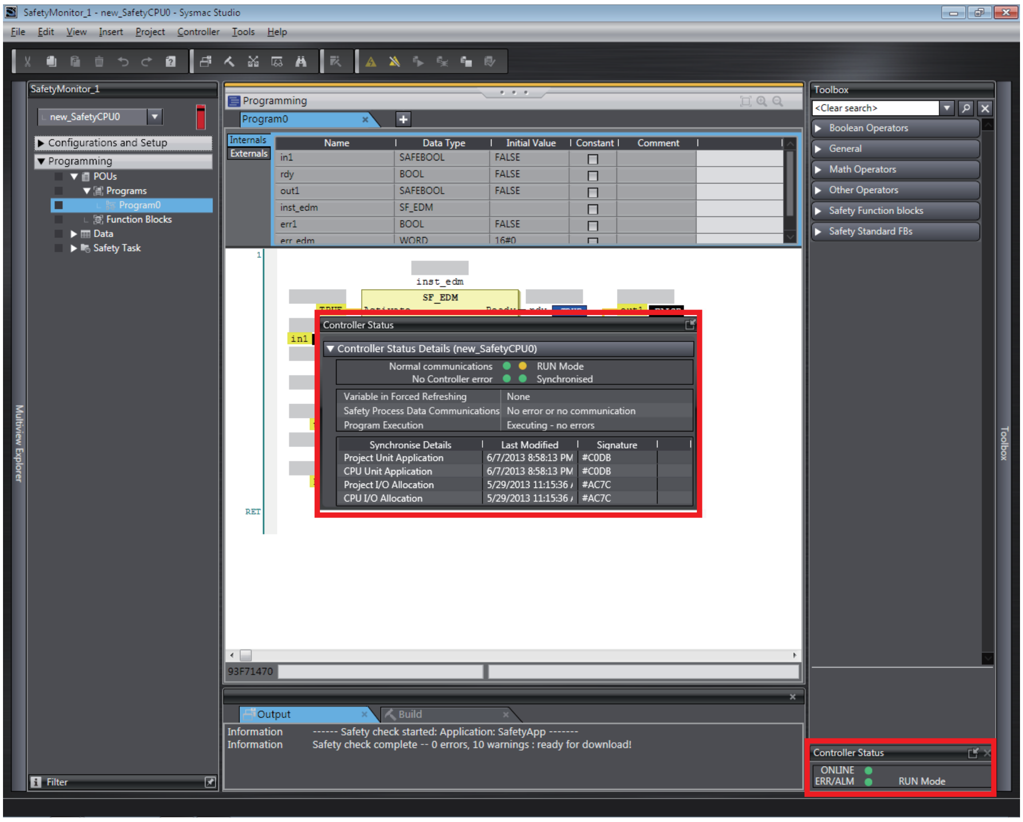
Serious injury may possibly occur due to loss of required safety functions. Before you start the system, perform user testing to make sure that all safety devices operate correctly.	❗
Always confirm safety at the destination before you transfer the unit configuration information, parameters, set values, or other data from the Sysmac Studio or other Support Software. The devices or machines may perform unexpected operation regardless of the operating mode of the CPU Unit.	❗

● Starting and Stopping the Safety Programs

1 With the Safety CPU Unit in DEBUG mode, perform the following operation.

- Select ***Debug* – *Run*** or ***Debug* – *Stop*** from the Controller Menu.

The Safety CPU Unit starts operating (DEBUG mode (RUN)) or remains stopped (DEBUG mode (STOPPED)).



8-5-3 Monitoring (Displaying the Present Values of Variables)

This section describes the procedures to monitor the present values of variables in the FBD editor or Watch Tab Page to debug the safety programs.

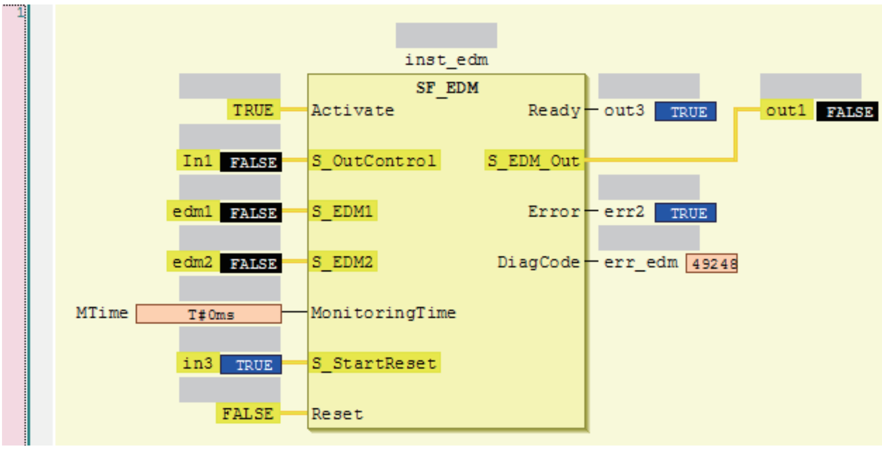
Monitoring the Present Values of Variables in the FBD Editor

You can monitor the present values of variables in the FBD editor. Use the following procedure.

● Executing the Operation Monitor for the Safety Programs

1 Double-click the required program under **Programming** – **Programs** 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.
  You select from the following for the numeric notation: decimal (default), binary, or hexadecimal.
- You cannot monitor connecting lines between variables and FBs.

Monitoring the Present Values of Variables in the Watch Tab Page

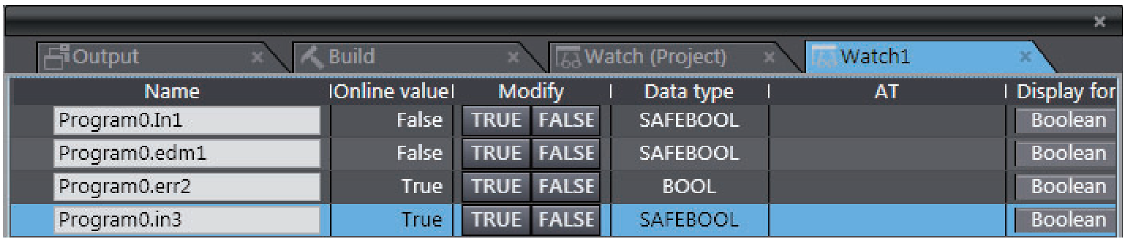
● Monitoring in a Watch Tab Page

You can check the present value of one or more variables in the Watch Tab Page.

● Displaying a Watch Tab Page

1 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

Item	Description	Editing
Name	The variable name is displayed.	Yes
Online value	The present value of the variable is displayed.	No
Modify	The new value is displayed.	Yes
Data type	The data type is displayed.	No
Data format	The display format (decimal, hexadecimal, etc.) of the present value and modify value is displayed.	Yes

Yes: Editable, No: Not editable

● 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.

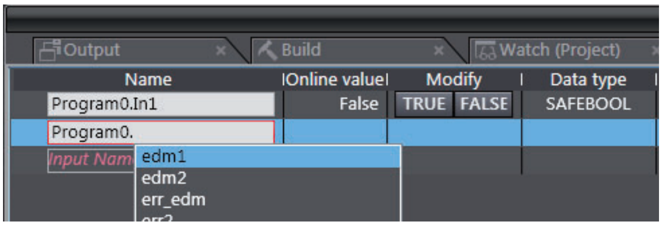
Method 1

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.



Method 2

1 Drag a variable from a variable table to the Watch Tab Page.

The variable is registered.

● Deleting Variable Names from the Watch Tab Page

1 Right-click the variable name to delete in the Watch Tab Page and select ***Delete*** from the menu. Or, press the **Backspace** Key to delete the variable name directly.

The variable name and the row it was displayed on are deleted.

8-5-4 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.

⚠ 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.

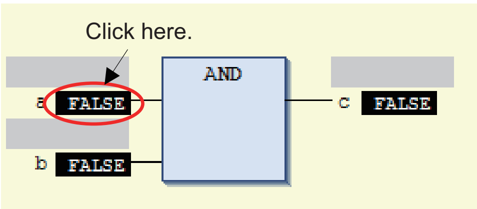
Controlling BOOL Variables (Set/Reset)

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)

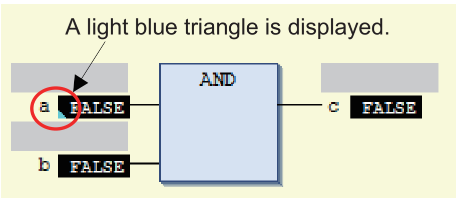
1 Click the present value for the BOOL variable to change.

Example: To set variable *a*, click the present value of *FALSE*.



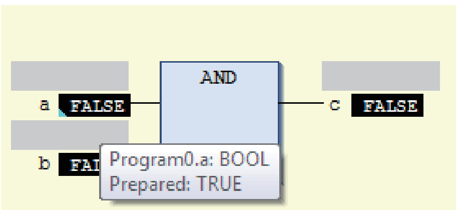
The value changes to a temporary status.

A light blue triangle is displayed at the lower left of the value. This indicates that a temporary TRUE/FALSE value is in effect.



Each click toggles the temporary value through *Present Value − TRUE − FALSE − Present Value*.

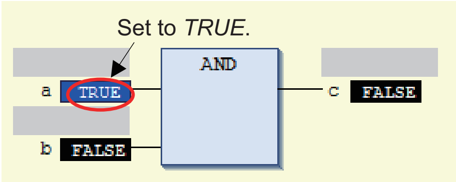
2 To check the temporary value, place the cursor over the value. The word *Prepared* is displayed in the tooltip.



3 Select ***Write Values*** from the Controller Menu.

The light blue triangle is removed and the temporary value is reflected as the present value.

Example: The present value of variable *a* is set to *TRUE*.



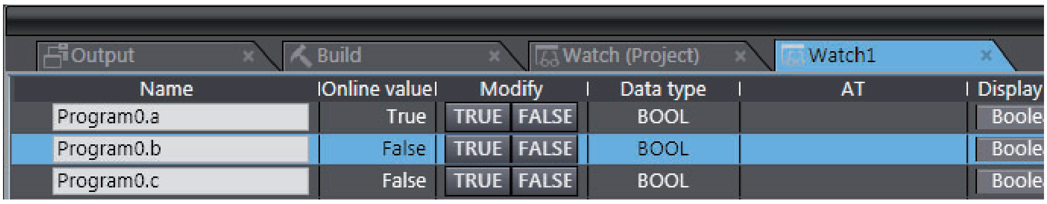
Additional Information

You can reflect more than one temporary value with a single operation. To do this, set multiple BOOL variables with temporary values, and then select ***Write Values*** from the Controller Menu.
If you set the same variable to different values in different places, the value that is set closest to the end of the program is used.

● 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.



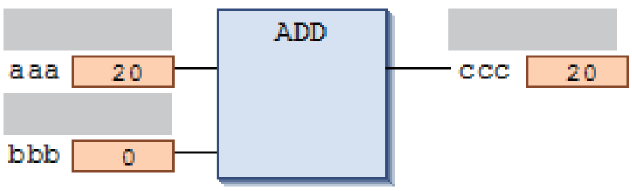
Changing the Present Values of Variables

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 Click the present value for the non-BOOL variable to change.

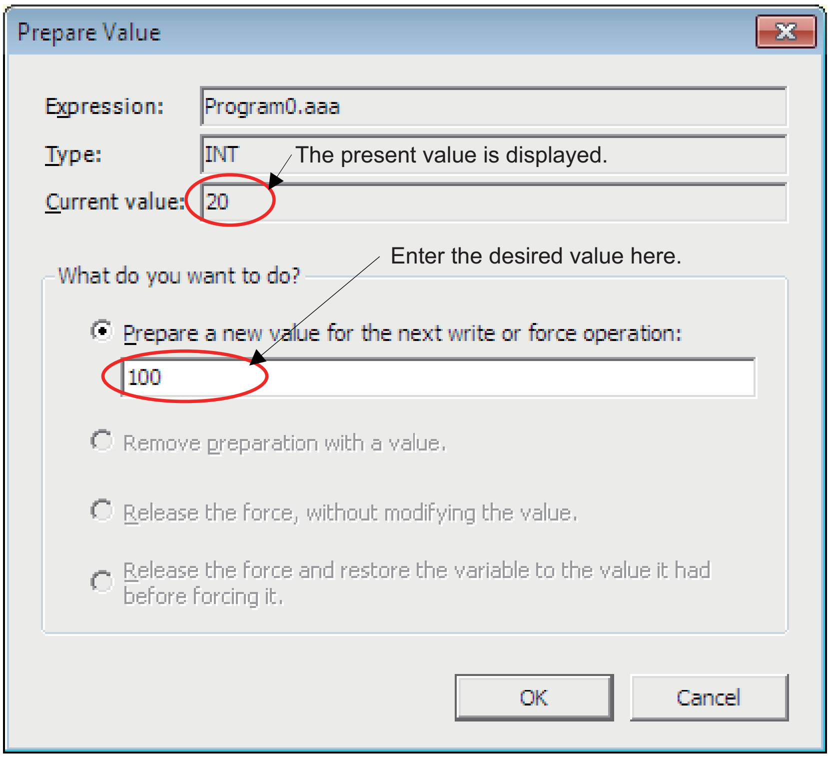
Example: To change the present value of variable *aaa*, click the present value of *20*.



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.

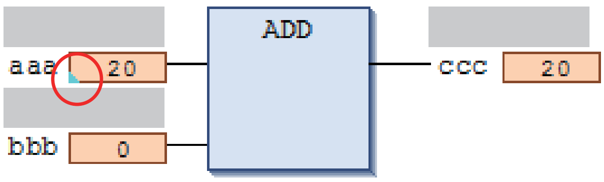
Example: This example changes the value to *100*.



3 Click the **OK** Button.

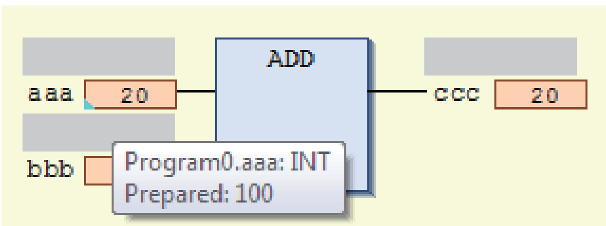
The Prepare Value Dialog Box closes and the value changes to a temporary value.

A light blue triangle is displayed at the lower left of the value. This indicates that a temporary value is in effect.



To cancel the temporary value, 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.

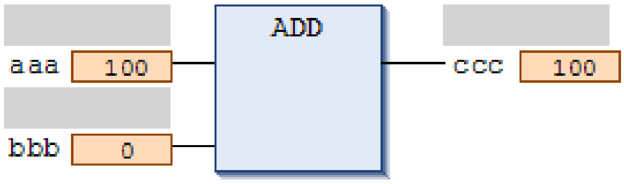
4 To check the temporary value, place the cursor over the value. The word *Prepared* is displayed in the tooltip.



5 Select ***Write Values*** from the Controller Menu.

The light blue triangle is removed and the temporary value is reflected as the present value.

Example: The present value of variable *aaa* is changed to *100*.



Additional Information

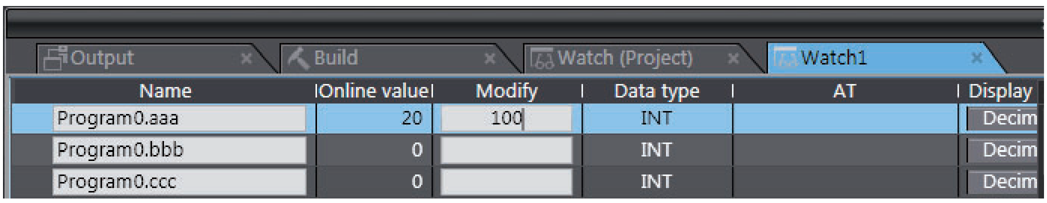
You can reflect more than one temporary value with a single operation. To do this, set multiple present values with temporary values, and then select ***Write Values*** from the Controller Menu. If you set the same variable to different values in different places, the value that is set closest to the end of the program is used.

● Changing Present Values on a 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 *Data type* Column, and then press the **Enter** Key. Press the **Esc** Key to cancel entering a value.

The present value is changed.



The format for entering a value in the *Modify* Column depends on the data type that is given in the *Data format* Column.

Refer to *8-5-3 Monitoring (Displaying the Present Values of Variables)* on page 8-20 for details.

Press the **Esc** Key to cancel entering a value.

- Examples of Entries in the *Modify* Column:

Type of data	Examples
Boolean	FALSE or TRUE
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, an error is detected 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 not for the specified device variables, but for the I/O ports that are assigned to the device 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 strings: BYTE and WORD

Integers: INT, SAFEINT, DINT, and SAFEDINT

Time of day data: TIME and SAFETIME

Precautions for Safe Use

- Forced refreshing ignores the results of safety program execution and refreshes the I/O ports with the specified values.
  If forced refreshing is used for I/O for which I/O refreshing is not supported, the I/O ports will first take the specified values, but they will then be overwritten by the user program.
- Depending on the difference in the forced status, the control system may operate unexpectedly.

Precautions for Correct Use

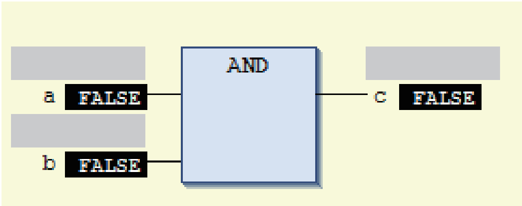
- Forced status for forced refreshing is not removed when you change from DEBUG mode (STOPPED) to DEBUG mode (RUN).
- You can use forced refreshing only for the following variables: device variables assigned to Safety I/O Units and user-defined variables.
- Even if you use forced refreshing for the input terminal to a Safety Input Unit, the forced value will not be applied to the variable that is assigned to the I/O port in the NJ-series CPU Unit.

● Forced Refreshing in the FBD Editor

Use the following procedure to execute forced refreshing on BOOL variables.

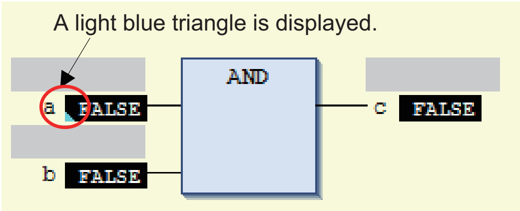
1 Click the present value for the BOOL variable to change.

Example: To force-refresh variable *a*, click the present value of *FALSE*.



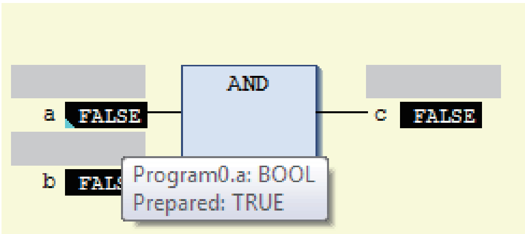
The value changes to a temporary status.

A light blue triangle is displayed at the lower left of the value. This indicates that a forced refreshing value is temporarily in effect.



Each click toggles the temporary value through *Present Value – TRUE – FALSE – Present Value*.

2 To check the temporary value, place the cursor over the value. The word *Prepared* is displayed in the tooltip.

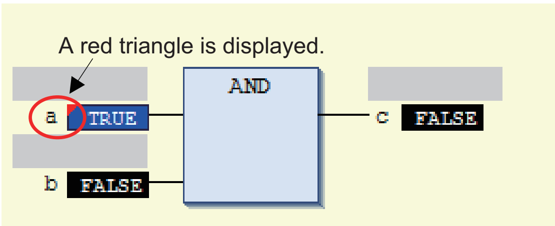


3 Select ***Force Values*** from the Controller Menu.

Forced refreshing is performed with the temporary values.

The light blue triangle at the lower left is removed, and the red triangle is displayed at the upper left.

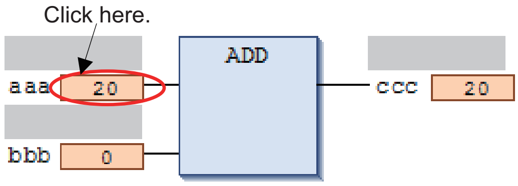
Example: The value of variable *a* is forced-refreshed to *TRUE*.



Use the following procedure to execute forced refreshing on non-BOOL variables.

1 Click the present value for the non-BOOL variable to change.

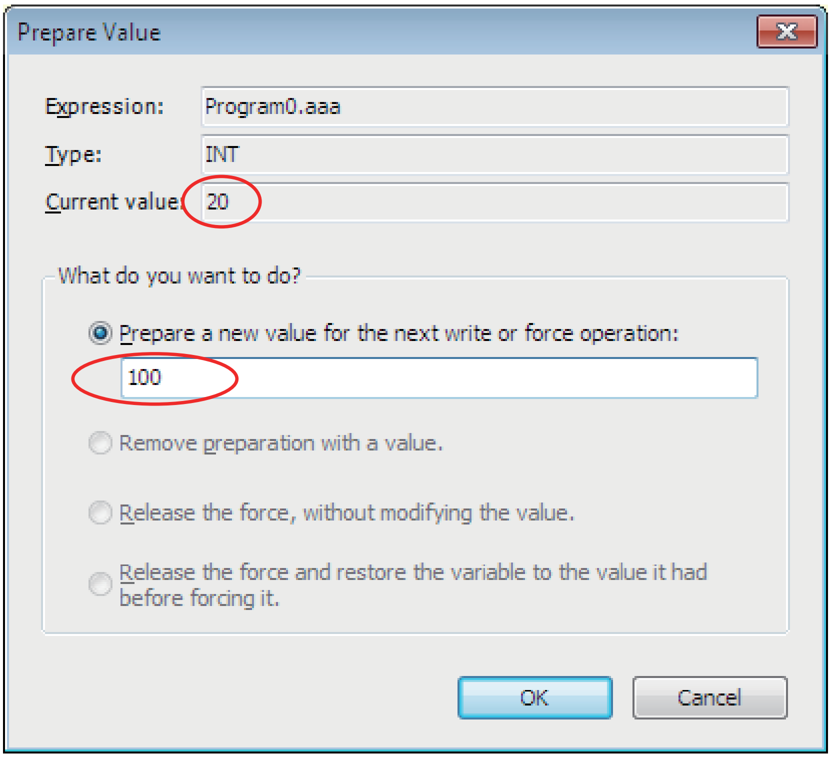
Example: To force-refresh the present value of variable *aaa*, click the present value of 20.



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.

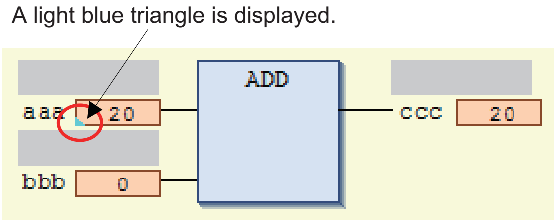
Example: This example changes the value to 100.



3 Click the **OK** Button.

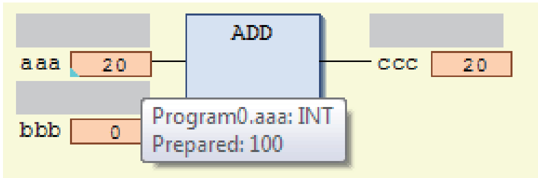
The Prepare Value Dialog Box closes and the value changes to a temporary value.

A light blue triangle is displayed at the lower left of the value. This indicates that a temporary value is in effect.



To cancel the temporary value, 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 To check the temporary value, place the cursor over the value. The word *Prepared* is displayed in the tooltip.



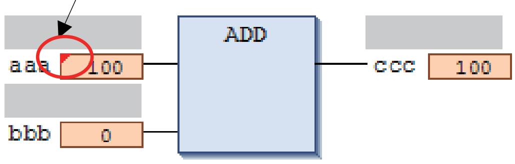
5 Select ***Force Values*** from the Controller Menu.

Forced refreshing is performed with the temporary values.

The light blue triangle at the lower left is removed, and the red triangle is displayed at the upper left.

Example: The value of variable *aaa* is forced-refreshed to 100.

A red triangle is displayed.



Additional Information

You can use forced refreshing for up to 19 variables at the same time

● Procedure to Cancel Forced Refreshing from the FBD editor

Use the following procedure to batch-clear forced refreshing.

1 Select ***Unforce Values*** from the Controller Menu.

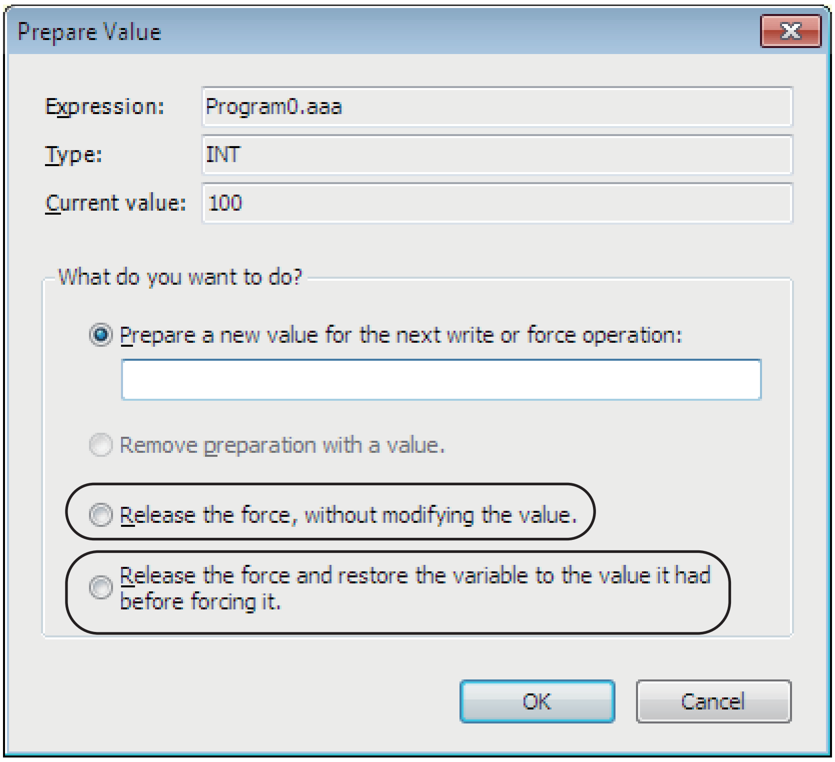
All forced refreshing is cleared at once.

The red triangles at the upper left of all forced refreshing values are removed. The forced values will remain unchanged.

Use the following procedure to individually clear forced refreshing.

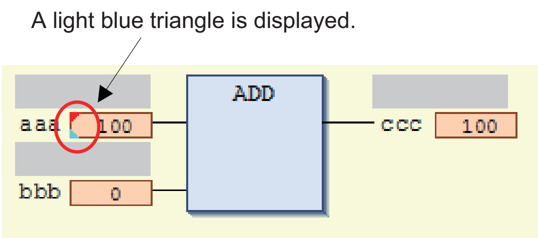
1 Click the present value for the variable to change.

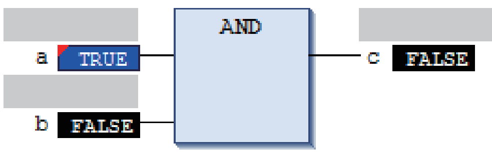
The following Prepare Value Dialog Box is displayed.



2 To clear the forced refreshing value and return the variable to the value that was in effect before forced refreshing, select the *Release the force and restore the variable to the value it had before forcing it* Option, and then click the **OK** Button.

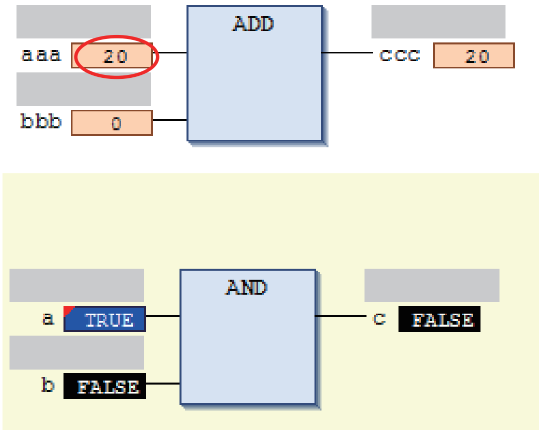
To clear forced refreshing without changing the present values, select the *Release the force, without modifying the values* Option, and then click the **OK** Button.





3 Select ***Force Values*** from the Controller Menu.

The forced refreshing is cleared individually. The red triangle at the upper left is removed.



Additional Information

You can simultaneously select up to 19 variables to clear forced refreshing.

8-6 Node name

This section describes the node name setting for the Safety CPU Unit.

● Node Names

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 in Node Names

The following characters can be used in node names.

The name must have 79 or less printable ASCII characters.

The default node name for all Safety CPU Units is *default*.

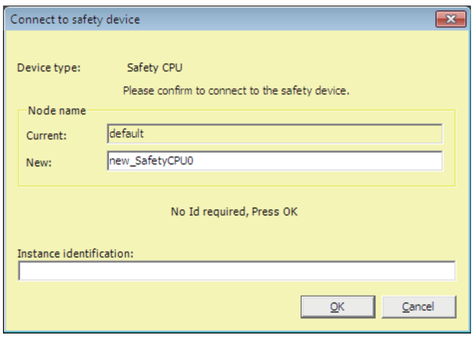
● Setting the Node Name

You set the node name in the Connect to safety device Dialog Box that 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.

- When you perform online operations on a Safety CPU Unit for the first time with the default settings.
- When you perform online operations on a Safety CPU Unit with a project file that is different from the one that was used to perform online operations before.

Use the following procedure.

1 The following Connect to safety device Dialog Box is displayed.



*1. The node name that is set by default is displayed.

2 Enter the node name, and click the **OK** Button.

The node name that you set is stored in the Safety CPU Unit. From this point on, the confirmation dialog box that is displayed when you are about to perform online operations on the Safety CPU Unit shows the node name that was set.

Precautions for Correct Use

Set a unique node name for the Safety CPU Unit.

8-7 Setting the Safety Password

This section describes the security password setting for the Safety CPU Unit.

● Safety Password

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 online 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).)
- 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 the safety password before or after you perform safety validation.

● Characters Allowed in Passwords

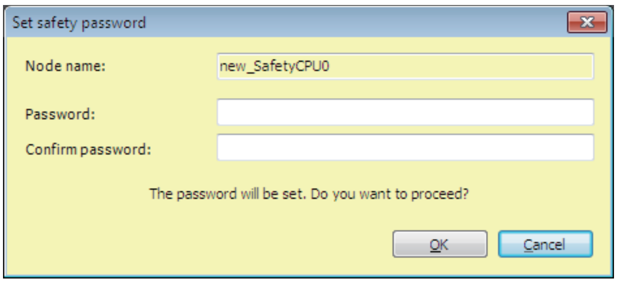
The following characters can be used in the password.

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 Password*** from the Controller Menu with the Controller set to the Safety CPU Unit.

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. The safety password does not need to be entered again until the Safety CPU Unit is taken offline.

Precautions for Correct Use

For security purposes, we recommend that you set a safety password for the Safety CPU Unit.

8-8 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.

This section describes how to perform safety validation and start operation after you have debugged the safety programs.

8-8-1 Performing Safety Validation

You must perform safety validation before you change the Safety CPU Unit to RUN mode and start any safety control system that uses safety application data[*1] that is created in the 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.

⚠ 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.
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.

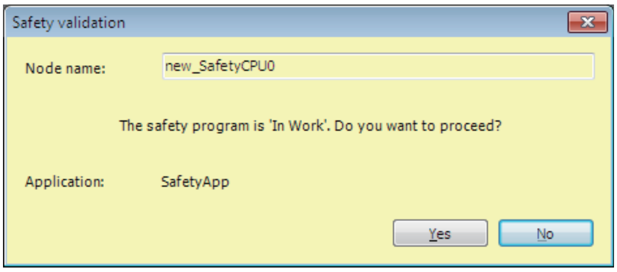
Additional Information

You can perform change management on 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-56 for details.

1 Place 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.

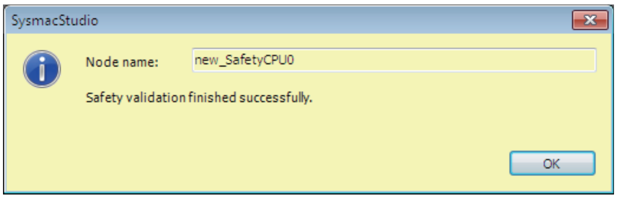
*1. The safety application data includes the safety programs and the safety task settings and variables. Refer to *8-1-2 Data That You Must Transfer before Operation and Data Transfer Procedures* on page 8-5 for details.

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.



- The Safety CPU Unit is set to the validated state, and the VALID indicator 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

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 EtherCAT Coupler Unit and NX Units, the Safety CPU Unit automatically restarts.

8-8-2 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 from the Sysmac Studio.

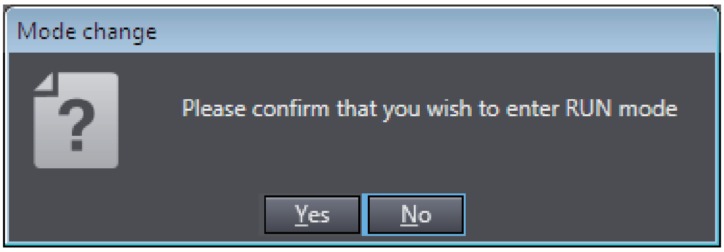
Changing to RUN Mode from the Sysmac Studio

The safety programs must be validated.

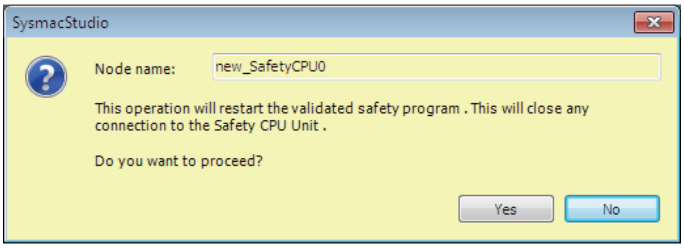
1 With the Safety CPU Unit connected online, perform one of the following operations on the Safety CPU Unit Setup and Programming View.

- 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 **Yes** Button to change the Safety CPU Unit to RUN mode.

8-8-3 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.

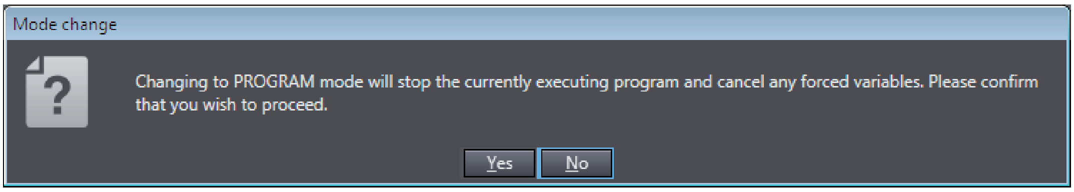
Changing to PROGRAM Mode

Use the following procedure to change the Safety CPU Unit from RUN mode to PROGRAM mode.

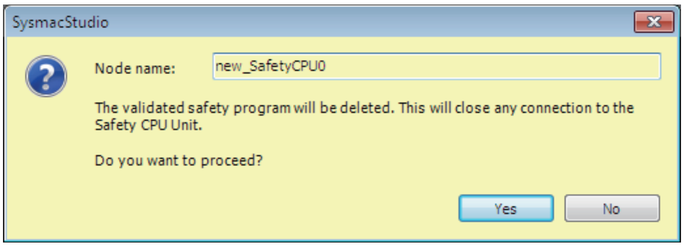
1 With the Safety CPU Unit connected online, perform one of the following operations.

- 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.

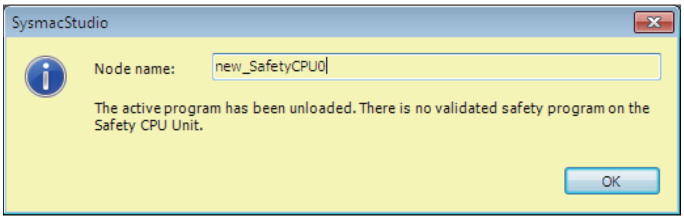


Click the **Yes** Button. A dialog box is displayed to confirm the node.



2 Check the node name, and click the **Yes** Button.

The following dialog box is displayed.



3 Click the **OK** Button.

A Mode Change Confirmation Dialog Box is displayed. Click the **Yes** Button to change the Safety CPU Unit to PROGRAM mode.

Precautions for Correct Use

When you change from RUN mode to PROGRAM mode, the validated safety programs that are saved in the non-volatile memory of the Safety CPU Unit are deleted.

Therefore, to return to RUN mode, you must perform safety validation again, and then transfer the validated safety programs to the non-volatile memory in the Safety CPU Unit.

8-9 Transferring Safety Application Data to a Different Safety CPU Unit

To use the validated safety application data described in the previous section *8-8 Performing Safety Validation and Operation* on page 8-34 in another Safety CPU Unit, change the other Safety CPU Unit to DEBUG mode and perform safety validation.

8-10 Monitoring Controller Status

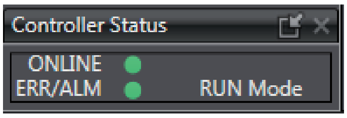
This section describes the procedure for monitoring the status of a Safety CPU Unit that is online.

Controller Status Monitoring

Controller status monitoring is used to display the status of the connected Safety CPU Unit in the Controller Status Pane. You can view the Controller Status Pane only when the Safety CPU Unit is online.

Displaying the Controller Status Pane

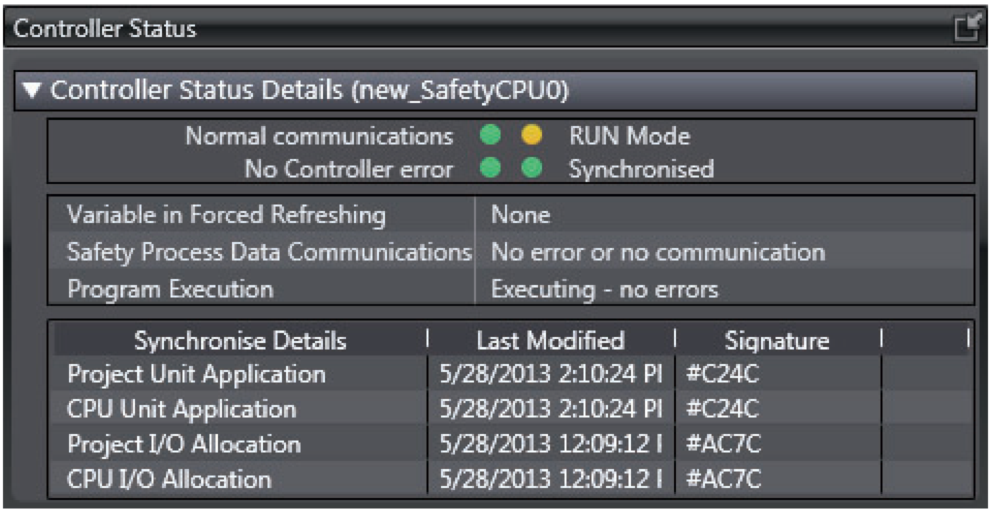
The Controller Status Pane is displayed in place of the Toolbox in the lower right corner of the window when the Safety CPU Unit is online.



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



8-11 Restarting and Clearing All Memory

8-11-1 Restarting

Restarting allows you to restart a Slave Terminal that includes the Safety CPU Unit and Safety I/O Units without cycling the Unit power supply to the EtherCAT Slave Terminal.

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

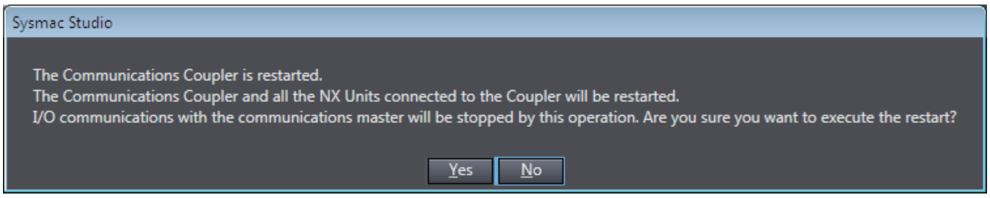
Precautions for Safe Use

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.

Use the following procedure to restart all of the Units in the Slave Terminal.

1 Go online, right-click the EtherCAT Coupler Unit in the Slave Terminal configuration, and select **Restart**.

The following restart confirmation dialog box is displayed.



2 Click the **Yes** Button.

After the Units are restarted, a Restart Completion Dialog Box is displayed.

8-11-2 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.

The Clear All Memory operation can be performed in the following two ways.

Type	Function
Clear All Memory operation for 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 EtherCAT Coupler Unit and all NX Units, including the Safety I/O Units that are connected to the EtherCAT Coupler Unit. The Safety CPU Unit memory cannot be cleared.

Refer to the *NX-series EtherCAT Coupler Unit User's Manual* (Cat. No. W519) for information on the Clear All Memory operation for EtherCAT Slave Terminals.

Precautions for Correct Use

- The memory in the Safety CPU Unit is not cleared even if you right-click the EtherCAT Coupler Unit in the Slave Terminal configuration 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.
- 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 Memory All Clear 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 Memory All Clear operation.

Precautions for Safe Use

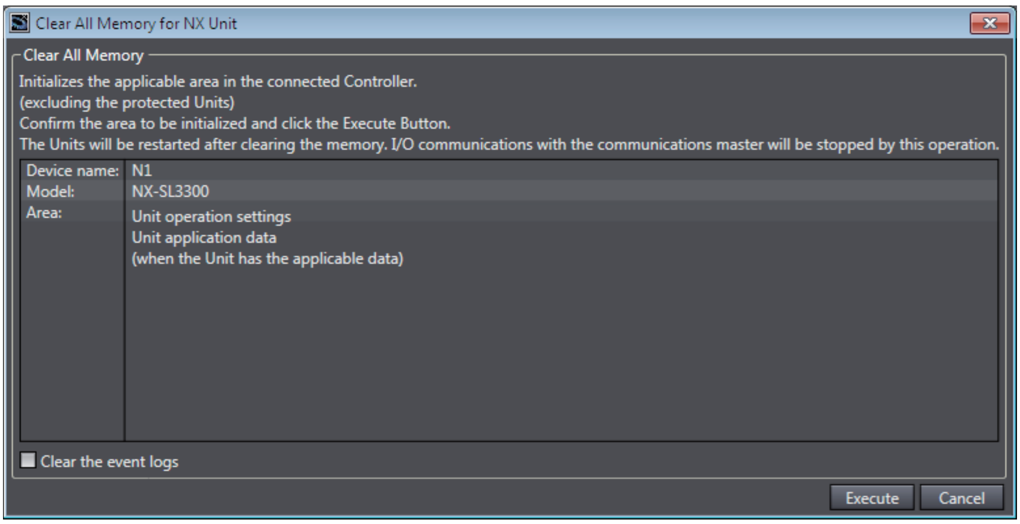
After you clear the memory, the Controller operates in the same way as immediately after you create the system configuration with the Controller in the factory default condition.

Procedure for Clear All Memory Operation

● Clear All Memory Operation for NX Units

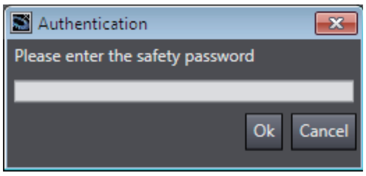
1 Go online, right-click a Safety CPU Unit or Safety I/O Unit on the Slave Terminal Tab Page, and select ***Clear All Memory*** from the menu.

You can select this menu command only when the Safety CPU Unit is in PROGRAM mode. The NX Unit Clear All Memory Dialog Box is displayed.



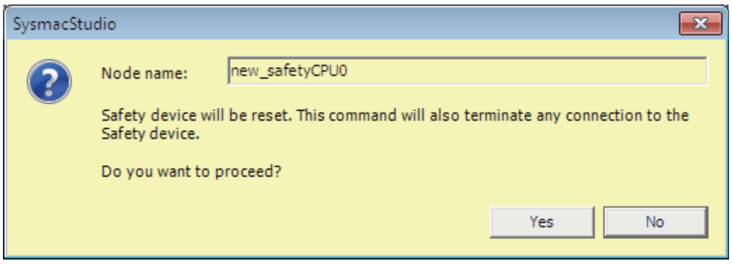
2 Click the **Execute** Button. The Clear All Memory Confirmation Dialog Box is displayed.

3 Click the **Yes** Button. The Authentication Dialog Box is displayed.

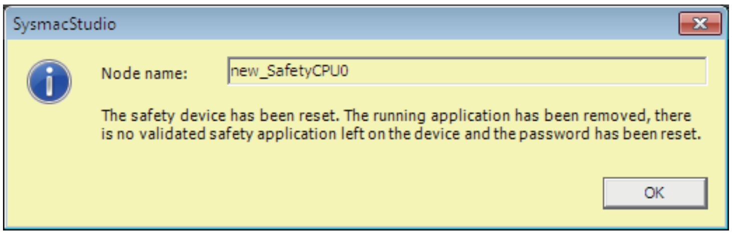


4 Enter the password, and click the **OK** Button. If a password is not set, leave the box empty and click the **OK** Button.

A dialog box is displayed to confirm the node.



5 Click the **Yes** Button. The following dialog box is displayed.



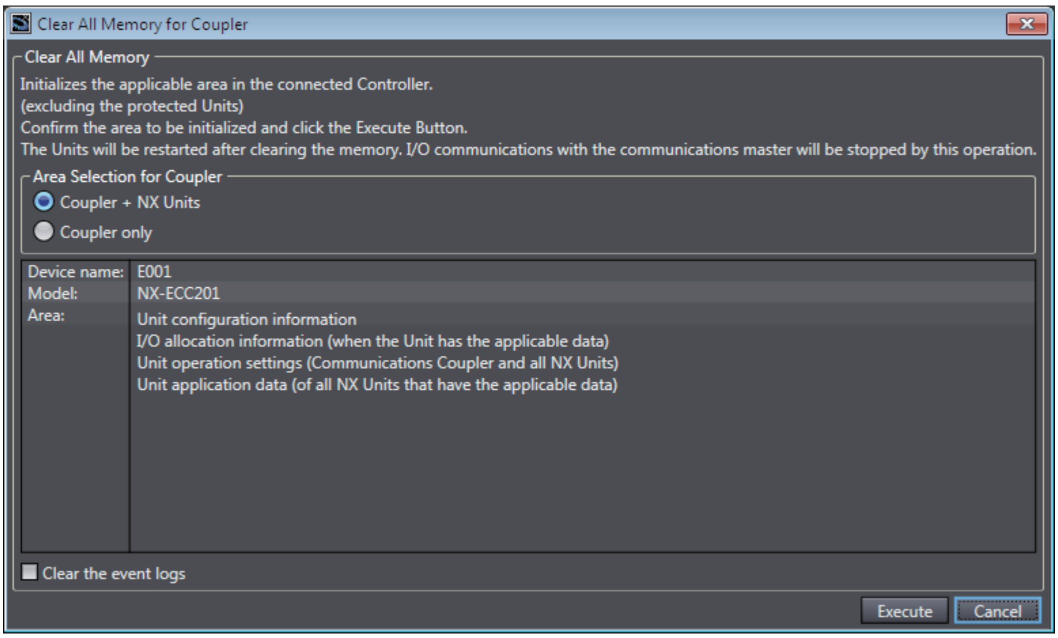
6 Click the **OK** Button.

After memory is cleared, the Memory All Cleared Dialog Box is displayed.

● Clear All Memory Operation for an EtherCAT Slave Terminal

1 Go online, right-click the EtherCAT 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.

9

Troubleshooting

This section describes troubleshooting for the Safety CPU Unit and Safety I/O Units.

9-1 Overview of Errors

This section describes the errors that can originate in the Safety CPU Unit and Safety I/O Units.

9-1-1 Checking for Errors in the Safety CPU Unit and Safety I/O Units

You can check to see if an error has occurred with the following methods.

Checking method	What you can check
Checking the indicators	Operating status of the NJ-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

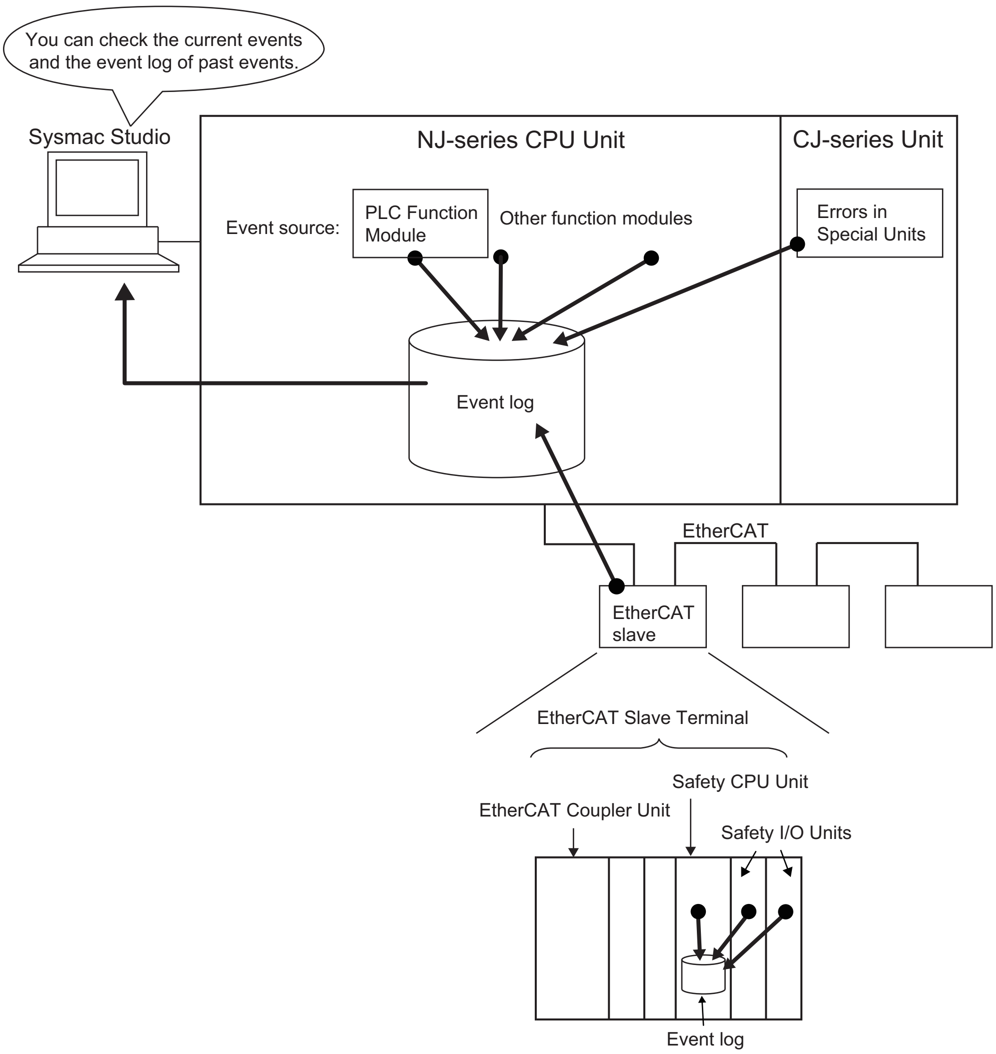
The rest of this section describes the above checking methods.

Checking the Indicators

Refer to *3-1-2 Indicators* on page 3-3 for the meaning of the indicators on the Safety CPU Unit and to *3-2-2 Indicators* on page 3-8 for the meaning of the indicators on the Safety I/O Units.

Checking with the Troubleshooting Function of the Sysmac Studio

If errors or other events* occur in the Safety CPU Unit or Safety I/O Units, they are stored in an event log in the NJ-series CPU Unit as EtherCAT slave events. The events are simultaneously stored in the event log in the Safety CPU Unit.



*1. Here, “events” are unscheduled events that occur in the Controller, such as errors. Event refers to an error or to information that does not indicate an error but for which the user must be notified by the Controller or for a user definition.

When an error occurs, you can place the Sysmac Studio online with the Safety CPU Unit from the NJ-series CPU Unit Setup and Programming View to check current errors in the Safety CPU Unit and Safety I/O Units and the log of past errors. You can also check the cause of any errors and corrections.

- **Items Displayed as Sources**

Sysmac Studio connection	Source	Source details
USB port or the built-in EtherNet/IP port on the NJ-series CPU Unit	EtherCAT Master Function Module	• Model number • Node address of EtherCAT Coupler Unit • Unit number of NX Unit
USB port on the EtherCAT Coupler Unit	EtherCAT Master Function Module	

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

9-1-2 Errors Related to the Safety CPU Unit and Safety I/O Units

Event Levels

The event levels of errors that are detected by the Safety CPU Unit and Safety I/O Units are described in the following table. The event levels that are given here are the event levels for the standard control that is performed by the NJ-series Controller. Therefore, all Controller errors that occur in the Safety CPU Unit or Safety I/O Units are minor faults.

Event level of the error	Operation
Minor fault	Some of the control operations for one of the function modules in the NJ-series Controller stop for errors in this event level.
Information	Events that are classified as information provide information that does not indicate errors but for which the user must be notified.

9-1-3 Confirming the Status of the Safety Control System

You can check the status of the safety control system with the indicators on the Safety CPU Unit and Safety I/O Units. You can obtain more detailed status by checking the indicators on the NJ-series CPU Unit and EtherCAT Coupler Unit. The following table shows the status of the Unit indicators when an error occurs after both the standard control and safety control systems have started normal operation.

Operating status	CPU Unit		EtherCAT Coupler Unit			Safety CPU Unit		Safety I/O Units	
	RUN (green)	ERROR (red)	RUN (green)	ERR (red)	TS (green/ red)	FS (green/ red)	TS (green/ red)	FS (green/ red)	TS (green/ red)
All Unit in normal operation									
During normal operation	Lit.	Not lit.	Lit.	Not lit.	Lit green.	Lit green.	Lit green.	Lit green.	Lit green.
CPU Unit error [*1]									
Major fault	Not lit.	Lit.	Single flash	Double flash	Flashing green.	Flashing red.	Flashing green.	Flashing red.	Flashing green.
Partial fault (EtherCAT stopped.)	Lit.	Flashing.	Single flash	Double flash	Flashing green.	Flashing red.	Flashing green.	Flashing red.	Flashing green.
Partial fault (Functions other than EtherCAT stopped.)	Lit.	Flashing.	Lit.	Not lit.	Lit green.	Lit green.	Lit green.	Lit green.	Lit green.
Minor fault	Lit.	Flashing.	Lit.	Not lit.	Lit green.	Lit green.	Lit green.	Lit green.	Lit green.
Observation	Lit.	Not lit.	Lit.	Not lit.	Lit green.	Lit green.	Lit green.	Lit green.	Lit green.
EtherCAT Coupler Unit error [*2]									
Hardware error (EtherCAT Coupler Unit must be replaced.)	Lit.	Flashing.	Not lit.	Lit or Flickering	Lit red.	Flashing red.	Flashing red.	Flashing red.	Flashing red.

Operating status		CPU Unit		EtherCAT Coupler Unit			Safety CPU Unit		Safety I/O Units	
		RUN (green)	ERROR (red)	RUN (green)	ERR (red)	TS (green/ red)	FS (green/ red)	TS (green/ red)	FS (green/ red)	TS (green/ red)
	Data error Hardware setting error Configuration error (configuration information)	Lit.	Flashing.	Not lit.	Blinking	Lit red.	Flashing red.	Flashing red.	Flashing red.	Flashing red.
	Software error	Lit.	Flashing.	Not lit.	Not lit.	Lit red.	Flashing red.	Flashing red.	Flashing red.	Flashing red.
	Communications errors except for the following Configuration error (communications setting information)	Lit.	Flashing.	Blinking	Blinking	Flashing red.	Flashing red.	Flashing red.	Flashing red.	Flashing red.
	Communications error: Synchronization Error	Lit.	Flashing.	Single flash	Single flash	Flashing red.	Flashing red.	Flashing red.	Flashing red.	Flashing red.
	Communications error: Process Data Communications Timeout	Lit.	Flashing.	Single flash	Double flash	Flashing red.	Flashing red.	Flashing red.	Flashing red.	Flashing red.
Safety CPU Unit error *3										
	System error	Lit.	Flashing.	Single flash	Double flash	Flashing red.	Not lit.	Lit red.	Flashing red.	Flashing red.
	Communications error	Lit.	Flashing.	Single flash	Double flash	Flashing red.	Flashing red.	Flashing red.	Flashing red.	Flashing red.
	Program execution error	Lit.	Flashing.	Lit.	Lit.	Lit.	Flashing red.	Lit green.	Lit green.	Lit green.
Safety I/O Unit error *3										
	System error	Lit.	Flashing.	Single flash	Double flash	Flashing red.	Flashing red.	Flashing red.	Not lit.	Lit red.
	Communications error	Lit.	Flashing.	Single flash	Double flash	Flashing red.	Flashing red.	Flashing red.	Flashing red.	Flashing red.
	Safety I/O error	Lit.	Flashing.	Lit.	Lit.	Lit.	Lit green.	Lit green.	Flashing red.	Lit green.

*1. Refer to the *NJ-series CPU Unit Software User's Manual* (Cat. No. W501) for information on CPU Unit errors.

*2. Refer to the *NX-series EtherCAT Coupler Unit User's Manual* (Cat. No. W519).

*3. For the indicator status for individual errors, refer to *9-1-1 Checking for Errors in the Safety CPU Unit and Safety I/O Units* on page 9-2 and *9-2 Troubleshooting* on page 9-8.

9-1-4 Checking for Errors and Troubleshooting on the Sysmac Studio

Error management on the NX Series is based on the methods used for the NJ-series Controllers.

This allows you to use the Sysmac Studio to check the meanings of errors and troubleshooting procedures.

Checking for Errors from the Sysmac Studio

When an error occurs, you can place the Sysmac Studio online to the Controller or the EtherCAT Coupler Unit to check current Controller errors and the log of past Controller errors.

The methods that are used to check errors depend on the Controller you use.

Controller used	Sysmac Studio connection	Scope of check	Remarks
NJ-series Controller	NJ-series CPU Unit	You can check the errors that are managed by the Controller. This includes errors for the connected EtherCAT Slave Terminals.	You cannot check errors if there is a fatal error in the CPU Unit.
	EtherCAT Coupler Unit	You can check the errors that are managed by the EtherCAT Coupler Unit. You can check errors in the EtherCAT Coupler Unit to which the Sysmac Studio is connected, and errors in the NX Units that are connected after the EtherCAT Coupler Unit.	• You can check errors in the Slave Terminals even if there is a fatal error in the CPU Unit. • You cannot check errors if there is a fatal error in the EtherCAT Coupler Unit. • Some errors in the NX Units cannot be checked if a fatal error occurs in that NX Unit.*1
Other controllers	EtherCAT Coupler Unit	Same as above.	Same as above.

*1. On NX Units that manage their own errors, current errors cannot be checked after a fatal error occurs in that NX Unit. On NX Units that record their own event logs, the error log cannot be checked after a fatal error occurs in that NX Unit.

Refer to the *NJ-series Troubleshooting Manual* (Cat. No. W503) for information on NJ-series error management methods.

Refer to the *Sysmac Studio Version 1 Operation Manual* (Cat. No. W504) for details on troubleshooting with the Sysmac Studio.

If you cannot check for errors from the Sysmac Studio, refer to the information in *9-2 Troubleshooting* on page 9-8 and check for errors.

Additional Information

Checking EtherCAT Slave Terminal Errors from an NS-series PT

You can use an NS-series PT to check for current errors in the EtherCAT Coupler Unit only. You cannot view the event log or current NX Unit 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	32
Safety I/O Unit	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 Unit
System event log	32 events total	10
Access event log	32 events total	10

Refer to the *NJ-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 *9-2-2 Event Codes for Errors and Troubleshooting Procedures* on page 9-9 for information on event codes.

9-2 Troubleshooting

This section describes the errors that can originate in the Safety CPU Unit and Safety I/O Units and how to correct them.

9-2-1 Types of Errors

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. The Safety I/O Units will detect this and make the safety I/O data inactive (OFF).
Communications error	Errors that occur in safety process data communications with the Safety I/O Units	The Safety CPU Unit will continue operation. The relevant safety process data communications will stop. The Unit that detects the safety process data communications error will make the safety I/O data inactive (OFF).
Program execution error	Errors that occur in the safety function blocks in the Safety CPU Unit	The Safety CPU Unit will continue operation. Safety process data communications will continue. Refer to the *NX-series Safety Control Unit Instructions Reference Manual* (Cat. No. Z931) for the operation of function blocks in which errors occur.
Other errors	Errors other than those given above	The Safety CPU Unit will continue operation. 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 CPU Unit will stop. The Safety CPU Unit will detect this and make the safety I/O data inactive (OFF).

Type	Overview	Operation
Communications error	Errors that occur in safety process data communications with the Safety CPU Units	The Safety CPU Unit will continue operation. safety process data communications will stop. The Safety I/O Unit that detects the safety process data communications error will make the safety I/O data inactive (OFF).
Safety I/O error	Errors that occur in safety I/O in the Safety I/O Units	The Safety CPU Unit will continue operation. Safety process data communications will continue. The safety I/O data will become inactive (OFF).

9-2-2 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-series Troubleshooting Manual* (Cat. No. W503) for all of the event codes that may occur for an NJ-series Controller.

Safety CPU Unit

The errors (events) that can occur in the Safety CPU Unit are listed in the following tables.

● System Errors

Event code	Event name	Meaning	Assumed cause	Level					Reference
				Maj	Prt	Min	Obs	Info	
10530000 hex	Non-volatile Memory Access Error	Reading/writing non-volatile memory failed.	Non-volatile memory failed.			√			P. 9-17
05200000 hex	System Error	A hardware error was detected during self-diagnosis of the hardware.	• Hardware has failed. • A memory error occurred due to a transient cause, such as a software error or excessive noise.			√			P. 9-17

● Communications Errors

Event code	Event name	Meaning	Assumed cause	Level					Reference
				Maj	Prt	Min	Obs	Info	
35200000 hex	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. • The safety slave for safety process data communications is not connected. • The safety slave for safety process data communications is disabled.			√			P. 9-18

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 Communications Coupler Unit and the NX Unit.	• The NX Unit is not mounted properly. • There is a hardware error in the NX Unit.			√			P. 9-19
80300000 hex	Safety Process Data Communications Timeout	A communications timeout occurred in safety process data communications with a safety slave.	• A setting is not correct. The setting of the safety task period of the Safety CPU Unit is too short. Or, the setting of the PDO communications safety task period of the EtherCAT master is too short. • There is excessive noise. • The safety slave entered a status where it could not continue safety process data communications. • An error or status change occurred in the EtherCAT Coupler Unit to which the Unit is connected, preventing correct process data communications.			√			P. 9-19
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.				√		P. 9-20

● Program Execution Errors

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.			√			P. 9-20
55010000 hex	Cast Error	A casting error was detected.	A value was input that exceeded the range of the receiving variable.			√			P. 9-21
55020000 hex	MUX Error	An MUX instruction error was detected.	The value of the selection input (K) to the MUX instruction is not correct.			√			P. 9-21
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)			√			P. 9-22
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)			√			P. 9-22
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)			√			P. 9-22

Event code	Event name	Meaning	Assumed cause	Level					Reference
				Maj	Prt	Min	Obs	Info	
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)			√			P. 9-23
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)			√			P. 9-23
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)			√			P. 9-23
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)			√			P. 9-24
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)			√			P. 9-24
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)			√			P. 9-24
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)			√			P. 9-25
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)			√			P. 9-25
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)			√			P. 9-25
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)			√			P. 9-26
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)			√			P. 9-26

Event code	Event name	Meaning	Assumed cause	Level					Reference
				Maj	Prt	Min	Obs	Info	
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)			√			P. 9-26
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)			√			P. 9-27
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)			√			P. 9-27

● Other 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. • Power was turned OFF while saving data to the non-volatile memory.			√			P. 9-28
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. • Power was turned OFF while saving data to the non-volatile memory.			√			P. 9-28
84F00000 hex	NX Bus I/O Communications Stopped	An error occurred in I/O communications between the Communications Coupler Unit and an NX Unit.	There is a hardware error in the Communications Coupler Unit or an NX Unit.			√			P. 9-29
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.			√			P. 9-29
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.					√	P. 9-30
951F0000 hex	Clear All Memory Rejected	Clearing all of memory failed.	• The Clear All Memory operation was performed for the entire Slave Terminal.					√	P. 9-30

- **User Access Log**

Event code	Event name	Meaning	Assumed cause	Level					Reference
				Maj	Prt	Min	Obs	Info	
9040 0000 hex	Event Log Cleared	The event log was cleared.	The event log was cleared by the user.					√	P. 9-31
9043 0000 hex	Memory All Cleared	The Unit settings were cleared.	The Clear All Memory operation was performed.					√	P. 9-31

Safety I/O Units

The errors (events) that can occur in the Safety I/O Units are listed in the following tables.

- **System Errors**

Event code	Event name	Meaning	Assumed cause	Level					Reference
				Maj	Prt	Min	Obs	Info	
0520 0000 hex	System Error	A hardware error was detected during self-diagnosis of the hardware.	• Hardware has failed. • A memory error occurred due to a transient cause, such as a software error or excessive noise.			√			P. 9-32

- **Communications Errors**

Event code	Event name	Meaning	Assumed cause	Level					Reference
				Maj	Prt	Min	Obs	Info	
3521 0000 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.			√			P. 9-32
3523 0000 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.			√			P. 9-33
3524 0000 hex	Safety Process Data Communications Not Established, Incorrect Frame Error	Safety process data communications was not established with the Safety CPU Unit because an incorrect frame was received.	• An incorrect frame was received in safety process data communications. There is excessive noise.			√			P. 9-33
8020 0000 hex	NX Unit I/O Communications Error	An I/O communications error occurred between the Communications Coupler Unit and the NX Unit.	• The NX Unit is not mounted properly. • There is a hardware error in the NX Unit.			√			P. 9-34

Event code	Event name	Meaning	Assumed cause	Level					Reference
				Maj	Prt	Min	Obs	Info	
8030 0000 hex	Safety Process Data Communications Timeout	A communications timeout occurred in safety process data communications with the Safety CPU Unit.	• A setting is not correct. The setting of the safety task period of the Safety CPU Unit is too short. Or, the PDO communications safety task period of the EtherCAT master is too short. • There is excessive noise. • The Safety CPU Unit entered a status where it could not continue safety process data communications. • An error or status change occurred in the EtherCAT Coupler Unit to which the Unit is connected, preventing correct process data communications.			√			P. 9-34
8022 0000 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. • The communications cable is disconnected or broken.				√		P. 9-35

● Safety I/O Errors

Event code	Event name	Meaning	Assumed cause	Level					Reference
				Maj	Prt	Min	Obs	Info	
0521 0000 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. • A memory error or signal error occurred due to a transient cause, such as a software error or excessive noise.			√			P. 9-35
0522 0000 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. • A memory error or signal error occurred due to a transient cause, such as a software error or excessive noise.			√			P. 9-36
0523 0000 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. • A memory error or signal error occurred due to a transient cause, such as a software error or excessive noise.			√			P. 9-36
6520 0000 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.			√			P. 9-37
6521 0000 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.			√			P. 9-38
6522 0000 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. • The input signal lines are shorted. • The external device is faulty.			√			P. 9-39

Event code	Event name	Meaning	Assumed cause	Level					Reference
				Maj	Prt	Min	Obs	Info	
6523 0000 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. • The connected device is faulty. • The setting of the discrepancy time is not correct. • Chattering occurred in the input signal from the external input device, such as a safety door.			√			P. 9-39
6524 0000 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. • The external device is faulty.			√			P. 9-40
6525 0000 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. • The internal circuit is faulty. • A memory error or signal error occurred due to a transient cause, such as a software error or excessive noise.			√			P. 9-40
6527 0000 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.			√			P. 9-41
6528 0000 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. • The output power supply is outside the specifications. • The internal circuit is faulty. • A memory error or signal error occurred due to a transient cause, such as a software error or excessive noise.			√			P. 9-41

● Other Errors

Event code	Event name	Meaning	Assumed cause	Level					Reference
				Maj	Prt	Min	Obs	Info	
84F1 0000 hex	NX Bus I/O Communications Stopped	An error occurred in I/O communications between the Communications Coupler Unit and an NX Unit.	There is a hardware error in the Communications Coupler Unit or an NX Unit.			√			P. 9-42

● User Access Log

Event code	Event name	Meaning	Assumed cause	Level					Reference
				Maj	Prt	Min	Obs	Info	
9040 0000 hex	Event Log Cleared	The event log was cleared.	The event log was cleared by the user.					√	P. 9-43
9043 0000 hex	Memory All Cleared	The Unit settings were cleared.	The Clear All Memory operation was performed.					√	P. 9-43

9-2-3 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.

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. *4	Operation	Provides specific information on the operation that results from the error.		
Indicators	Gives the status of the indicators on the Safety 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.
Depends on cause: The recovery method depends on the cause of the error.

*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-series CPU Unit will continue.
Stops: Execution of the user program in the NJ-series CPU Unit stops.
Starts: Execution of the user program in the NJ-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 Errors

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
Error attributes	Level	Minor fault	Recovery	Perform the Clear All Memory operation for the Unit or transfer the settings again.	Log category	System
Effects	User program	Continues.	Operation	I/O refreshing for the NX Unit stops. Messages cannot be sent to the NX Unit.		
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	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 Attached information 2: System information, status code Attached information 3: System information, status code 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 for safety process data communications is disabled.		Set the disabled safety slaves so that they do not participate in safety process data communications and then transfer the data to the Safety CPU Unit.		Set disabled safety slaves so that they do not participate in communications with the Safety CPU Unit.	
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	NX Unit I/O Communications Error			Event code	8020 0000 hex	
Meaning	An I/O communications error occurred between the Communications Coupler Unit and the 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	Reset error in the NX Unit.	Log category	System
Effects	User program	Continues.	Operation	The NX Unit will continue to operate. Input data: Updating input values stops. Output data: The output values depend on the Load Rejection Output Setting.		
System-defined variables	Variable		Data type		Name	
	None		---		---	
Cause and correction	Assumed cause		Correction		Prevention	
	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.	
	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	8030 0000 hex	
Meaning	A communications timeout occurred in safety process data communications with a safety 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 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 of the Safety CPU Unit is too short. Or, the setting of the PDO communications safety task period of the EtherCAT master is too short.		Increase the safety task period of the Safety CPU Unit and then transfer the settings to the Safety CPU Unit. Or, increase the PDO communications safety task period of the EtherCAT master and transfer the settings to the EtherCAT master.		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 slave entered a status where it could not continue safety process Data communications.		Check the status of the safety slave.		Refer to the causes and corrections for the safety slave.	
	An error or status change occurred in the EtherCAT Coupler Unit to which the Unit is connected, preventing correct process data communications.		Check the status of the EtherCAT Coupler Unit 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	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. Refer to the appendix of the *NJ-series Instructions Reference Manual* (Cat. No. W502-E1-07) for information on the instructions that send messages.		Reduce the number of times that instructions are used to send NX messages.	
Attached information	Attached information 1: System information Attached information 2: Type of communications where error occurred 0: NX bus 65535: Internal Unit communications (routing)					
Precautions/ Remarks	None					

● Program Execution Errors

Event name	Division by Zero			Event code	55000000 hex	
Meaning	Division by zero was detected.					
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	Automatic recovery	Log category	System
Effects	User program	Continues.	Operation	The Safety CPU Unit executes NX bus communications but execution of 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 divisor is zero.		Correct the program so that the divisor is not 0. Perform the following corrections according to the operating mode of the Safety CPU Unit. • RUN mode: Restart the Safety CPU Unit. Or, change to PROGRAM mode and transfer the corrected user program. • DEBUG mode (RUN): Change to PROGRAM mode and transfer the corrected user program.		Program operation considering the corrections that are given on the left.	
Attached information	None					
Precautions/ Remarks	The CPU Unit executes NX bus communications but execution of the user program stops. (All I/O data will remain at 0.)					

Event name	Cast Error			Event code	55010000 hex	
Meaning	A casting error was detected.					
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	Automatic recovery	Log category	System
Effects	User program	Continues.	Operation	The Safety CPU Unit executes NX bus communications but execution of 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	
	A value was input that exceeded the range of the receiving variable.		Do not allow the value to exceed the range of the receiving variable. Perform the following corrections according to the operating mode of the Safety CPU Unit. • RUN mode: Restart the Safety CPU Unit. Or, change to PROGRAM mode and transfer the corrected user program. • DEBUG mode (RUN): Change to PROGRAM mode and transfer the corrected user program.		Program operation considering the corrections that are given on the left.	
Attached information	Attached information 1: Error details 0x01000AE0: The positive upper limit of the data type after conversion was exceeded. 0x01000AE1: The negative upper limit of the data type after conversion was exceeded.					
Precautions/ Remarks	The Safety CPU Unit executes NX bus communications but execution of the user program stops. (All I/O data will remain at 0.)					

Event name	MUX Error			Event code	55020000 hex	
Meaning	An MUX instruction error was detected.					
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	Automatic recovery	Log category	System
Effects	User program	Continues.	Operation	The Safety CPU Unit executes NX bus communications but execution of 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 value of the selection input (K) to the MUX instruction is not correct.		Correct the program so that the value of the selection input (K) to the MUX instruction is in range. Perform the following corrections according to the operating mode of the Safety CPU Unit. • RUN mode: Restart the Safety CPU Unit. Or, change to PROGRAM mode and transfer the corrected user program. • DEBUG mode (RUN): Change to PROGRAM mode and transfer the corrected user program.		Program operation considering the corrections that are given on the left.	
Attached information	Attached information 1: 0x01000ADD: The value of the selection input (K) is negative. 0x01000ADE: The value of the selection input (K) exceeded the upper limit of the selection range.					
Precautions/ Remarks	The Safety CPU Unit executes NX bus communications but execution of the user program stops. (All I/O data will remain at 0.)					

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					

Other Errors

Event name	NX Bus Communications Settings Read Error			Event code	1050 0000 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 Safety 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 NX bus communications settings 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	1051 0000 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 Safety CPU Unit executes NX bus communications with the relevant Units but refreshing for the safety 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 NX bus communications settings and safety application data to the Safety CPU Unit again.		None	
Attached information	None					
Precautions/ Remarks	None					

Event name	NX Bus I/O Communications Stopped			Event code	84F00000 hex	
Meaning	An error occurred in I/O communications between the Communications Coupler Unit 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 EtherCAT Coupler Unit and NX Units.	Log category	System
Effects	User program	Continues.	Operation	The NX Units will continue to operate. Input data: 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. Output data: 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 Communications Coupler Unit or an NX Unit.		If the error occurs again even after you cycle the power supply to the NX Units, replace the Communications Coupler Unit or NX Unit.		None	
Attached information	None					
Precautions/ Remarks	None					

<table>
<tr><td>Event name</td><td colspan="3">Sysmac Studio Communications Connection Timeout</td><td>Event code</td><td colspan="2">951E0000 hex</td></tr>
<tr><td>Meaning</td><td colspan="6">A communications timeout occurred between the Sysmac Studio and the Safety CPU 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 applicable</td></tr>
<tr><td>Error attributes</td><td>Level</td><td>Information</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">If the CPU Unit was in DEBUG mode, it automatically enters PROGRAM mode.</td></tr>
<tr><td rowspan="2">Variable</td><td colspan="2">Data type</td><td colspan="2">Name</td><td colspan="2">System-defined variables</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 communications cable was disconnected.</td><td colspan="2">Do not do anything to disconnect communications with the Sysmac Studio while the Safety CPU Unit is operating in DEBUG mode.</td><td colspan="2">Perform debugging 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>

<table>
<tr><td>Event name</td><td colspan="3">Clear All Memory Rejected</td><td>Event code</td><td colspan="2">951F0000 hex</td></tr>
<tr><td>Meaning</td><td colspan="6">Clearing all of memory failed.</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 commanded from user</td></tr>
<tr><td>Error attributes</td><td>Level</td><td>Information</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="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 Clear All Memory operation was performed for the entire Slave Terminal.</td><td colspan="2">Specify the Units individually and perform the Clear All Memory operation.</td><td colspan="2">Specify the Units individually and perform the Clear All Memory operation.</td></tr>
<tr><td>Attached information</td><td colspan="6">Attached information 1: The Clear All Memory operation was performed for the Slave Terminal.</td></tr>
<tr><td>Precautions/ Remarks</td><td colspan="6">The Clear All Memory operation for the Safety CPU Unit cannot be performed for the entire Slave Terminal at the same time.</td></tr>
</table>

● User Access Log

Event name	Event Log Cleared			Event code	9040 0000 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 1: The system event log was cleared. 2: The access event log was cleared.					
Precautions/ Remarks	None					

Event name	Memory All Cleared			Event code	9043 0000 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 0: Successful 1: Hardware error 2: Initialization failed 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 Errors

<table>
<tr><td>Event name</td><td colspan="3">System Error</td><td>Event code</td><td colspan="2">05200000 hex</td></tr>
<tr><td>Meaning</td><td colspan="6">A hardware error was detected during self-diagnosis of the hardware.</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 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 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="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">Hardware has failed.</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 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: System information, status code
Attached information 2: System information, status code
Attached information 3: System information, status code
Attached information 4: System information, status code</td></tr>
<tr><td>Precautions/ Remarks</td><td colspan="6">None</td></tr>
</table>

● Communications Errors

<table>
<tr><td>Event name</td><td colspan="3">Safety Process Data Communications Not Established - Incorrect Unit Parameter Error</td><td>Event code</td><td colspan="2">35210000 hex</td></tr>
<tr><td>Meaning</td><td colspan="6">Safety process data communications was not established with the Safety CPU 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 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="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" rowspan="2">The model or safety I/O terminal settings are not correct.</td><td colspan="2">Check the safety I/O terminal settings, correct any errors, and then transfer the settings to the Safety CPU Unit.</td><td colspan="2" rowspan="2">Set the parameters considering the corrections that are given on the left.</td></tr>
<tr><td colspan="2">Check the model of the Safety I/O Unit to see if it 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 Process Data Communications Not Established, Incorrect FSoE Slave Address Error		Event code	35230000 hex	
Meaning	Safety process data communications was not established with the Safety CPU Unit because of an incorrect FSoE slave address.				
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 setting of the FSoE slave address in the safety process data communications settings is different from the setting in the Unit.	Perform the Clear All Memory operation for the Unit.		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.	
Attached information	None				
Precautions/ Remarks	None				

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				

Event name	NX Unit I/O Communications Error			Event code	8020 0000 hex	
Meaning	An I/O communications error occurred between the Communications Coupler Unit and the 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	Reset error in the NX Unit.	Log category	System
Effects	User program	Continues.	Operation	The NX Unit will continue to operate. Input data: Updating input values stops. 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	
	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.	
	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	8030 0000 hex	
Meaning	A communications timeout occurred in safety process data communications 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 establishing or during safety process data communications
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	
	A setting is not correct. The setting of the safety task period of the Safety CPU Unit is too short. Or, the PDO communications safety task period of the EtherCAT master is too short.		Increase the safety task period of the Safety CPU Unit and then transfer the settings to the Safety CPU Unit. Or, increase the PDO communications safety task period of the EtherCAT master and transfer the settings to the EtherCAT master.		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 entered a status where it could not continue safety process data communications.		Check the status of the Safety CPU Unit.		Refer to the causes and corrections for the Safety CPU Unit.	
	An error or status change occurred in the EtherCAT Coupler Unit to which the Unit is connected, preventing correct process data communications.		Check the status of the EtherCAT Coupler Unit 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	None					

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. Refer to the appendix of the *NJ-series Instructions Reference Manual* (Cat. No. W502-E1-07) for information on the instructions that send messages.		Reduce the number of times that instructions are used to send NX messages.	
Attached information	Attached information 1: System information Attached information 2: Type of communications where error occurred 0: NX bus 65535: Internal Unit communications (routing)					
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 a software error or 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 a software error or excessive noise.					
Attached information	Attached information 1: Terminal number					
Precautions/ Remarks	None					

Event name	Internal Circuit Error at Safety Output			Event code	05230000 hex	
Meaning	A fault was detected in the internal circuit for 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	Cycle the power supply to the Unit.	Log category	System
Effects	User program	Continues.	Operation	The safety output 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 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 a software error or excessive noise.					
Attached information	Attached information 1: Terminal number					
Precautions/ Remarks	None					

<table>
<tr><td>Event name</td><td colspan="3">I/O Power Supply Voltage Error</td><td>Event code</td><td colspan="2">65200000 hex</td></tr>
<tr><td>Meaning</td><td colspan="6">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.
• Is the power supply voltage within the specifications?
• Is the wiring correct and not disconnected?
• Is 24 V applied to the safety output terminal and is the safety output terminal not touching the positive power supply wire?
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	Output Power Interrupt Circuit Error			Event code	65210000 hex		
Meaning	An error was detected by the output power interruption test.						
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 cause of error is removed and then one of the following is performed • The I/O power supply is turned OFF. • Safety process data communications are stopped.	Log category	System	
Effects	User program	Continues.	Operation	The Unit continues to operate, but the I/O data retains the safe states.			
System-defined variables	Variable		Data type		Name		
	None		None		None		
Cause and correction	Assumed cause		Correction		Prevention		
	The wiring is not correct or there is a fault in the hardware.		Check the following and supply the rated power. • Is the power supply voltage within the specifications? • Is the wiring correct and not disconnected? • Is 24 V applied to the safety output terminal and is the safety output terminal not touching the positive power supply wire? If the voltage that is measured is correct, the Unit may be faulty. In that case, replace the CPU Unit.		Design the system considering the corrections that are given on the left.		
Attached information	None						
Precautions/ Remarks	None						

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.			
Attached information	Attached information 1: Terminal number					
Precautions/ Remarks	None					

Event name	Discrepancy Error at Safety Input			Event code	65230000 hex	
Meaning	An error was detected in discrepancy evaluation of 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	
	There is a ground fault or disconnection in 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 connected device is faulty.		Replace the external device.			
	The setting of the discrepancy time is not correct.		Correct the discrepancy evaluation time. If that does not correct the problem, use an input filter to set an ON delay or an OFF delay.			
	Chattering occurred in the input signal from the external input device, such as a safety door.					
Attached information	Attached information 1: Terminal number					
Precautions/ Remarks	None					

Event name	Overload Detected at Test Output			Event code	6524 0000 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					

Event name	Stuck-at-high Detected at Test Output			Event code	6525 0000 hex	
Meaning	It was detected that the test output terminal is stuck ON.					
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	
	The positive power supply line is in contact with 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 internal circuit 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 a software error or excessive noise.					
Attached information	Attached information 1: Terminal number					
Precautions/ Remarks	None					

Event name	Short Circuit Detected at Safety Output			Event code	6527 0000 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					

Event name	Stuck-at-high Detected at Safety Output			Event code	6528 0000 hex	
Meaning	It was detected that the safety output terminal is stuck ON.					
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	
	The positive power supply line is in contact with 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 output power supply is outside the specifications.		Check the output power supply.			
	The internal circuit 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 a software error or excessive noise.					
Attached information	Attached information 1: Terminal number					
Precautions/ Remarks	None					

Other Errors

Event name	NX Bus I/O Communications Stopped			Event code	84F10000 hex	
Meaning	An error occurred in I/O communications between the Communications Coupler Unit 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 EtherCAT Coupler Unit and NX Units.	Log category	System
Effects	User program	Continues.	Operation	The NX Units will continue to operate. Input data: An error occurs in safety process data communications because refreshing is stopped. The values of the status in standard process data are not refreshed. Output data: 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 Communications Coupler Unit or an NX Unit.		If the error occurs again even after you cycle the power supply to the NX Units, replace the Communications Coupler Unit or NX Unit.		None	
Attached information	None					
Precautions/ Remarks	None					

User Access Log

Event name	Event Log Cleared			Event code	9040 0000 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 1: The system event log was cleared. 2: The access event log was cleared.					
Precautions/ Remarks	None					

Event name	Memory All Cleared			Event code	9043 0000 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 0: Successful 1: Hardware error 2: Initialization failed 3: Initialization not possible					
Precautions/ Remarks	Refer to the attached information for the results of the Clear All Memory operation.					

9-3 Resetting Errors

Current errors in a Slave Terminal are retained, unless you reset them, until you cycle the power supply or restart the Slave Terminal.

To reset errors, you must remove the cause of the current error. If you reset an error without removing the cause, the same error will occur again.

Precautions for Correct Use

Resetting the errors does not remove the cause of the error.

Always remove the cause of the error, and then reset the error.

You can use the following methods to reset errors in a Slave Terminal.

Method	Operation	Scope of error reset	Description
Commands from Sysmac Studio	Resetting Controller errors	All errors in the Controller	Reset the Controller error from the Troubleshooting Dialog Box on the Sysmac Studio.
		All errors in the Slave Terminal	Refer to the user's manual of the Communications Coupler Unit for details on resetting errors in the EtherCAT Slave Terminal.
		Errors for individually specified NX Units	
	Clearing all memory for the Slave Terminal	All errors in the Slave Terminal	If the causes for the Controller errors are removed, all Controller errors in the Slave Terminals are reset.
	Restarting Slave Terminals		
Commands from the user program	Resetting Controller errors in the EtherCAT Master Function Module	All errors in the EtherCAT Master Function Module	Execute the Reset EtherCAT Error (ResetECError) instruction in the user program of the NJ-series Controller.
Cycling the Unit power supply to the Slave Terminal	---	All errors in the Slave Terminal	If the causes for the Controller errors are removed, all Controller errors in the Slave Terminals are reset.

Note On the NS-Series PT, you can only reset all errors for the entire Controller.

Refer to the *Sysmac Studio Version 1 Operation Manual* (Cat. No. W504-E1-07 or later) for Sysmac Studio operating procedures.

For details on the Reset EtherCAT Error (ResetECError) instruction, refer to the *NJ-series Instructions Reference Manual* (Cat. No. W502).

9-4 Troubleshooting Flow When Errors Occur

Refer to the *NX-series EtherCAT Coupler Unit Users Manual* (Cat. No. W519) for the standard flow of operations to perform when an error occurs.

10

Maintenance and Inspection

This section describes the procedures for cleaning, inspecting, and replacing Safety Units.

10-1 Cleaning and Maintenance

10-1-1 Cleaning

Clean the Safety Units regularly as described below in order to keep them in optimal operating condition.

- Wipe the network over with a soft, dry cloth when doing daily cleaning.
- If dirt remains even after wiping with a soft, dry cloth, wipe over 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 connector.

10-1-2 Periodic Inspections

Although the major components in Safety Units 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 Safety Units are used within the ranges of specifications.
- Make sure that the Safety 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.

Additional Information

The periodic inspection interval is influenced by the proof test interval used to determine the PFD. Always consider the PFD when determining periodic inspection intervals.

Periodic Inspection Points

No.	Item	Inspection	Criteria	Corrective action
1	External power supplies	Measure the power supply voltage at the terminal blocks, and make sure that they are within the criteria voltage.	The voltage must be within the power supply voltage range.	Use a voltage tester to check the power supply at the terminals. Take necessary steps to bring voltage of the supplied power to within the power supply voltage range.
2	I/O power supplies	Measure the power supply voltages at the input and output terminal blocks, and make sure that they are within the criteria voltage.	The voltages must be within the I/O specifications for each NX Unit.	Use a voltage tester to check the power supply at the terminals. Take necessary steps to bring voltage of the I/O power supplies to within the I/O specifications of each Unit.

No.	Item	Inspection	Criteria	Corrective action
3	Ambient environment	Check that the ambient operating temperature is within the criteria.	0 to 55°C	Use a thermometer to check the temperature and ensure that the ambient temperature remains within the allowed range of 0 to 55°C.
		Check that the ambient operating humidity is within the criteria.	10% to 95% With no condensation.	Use a hygrometer to check the humidity and ensure that the ambient humidity remains between 10% and 95%. Check that condensation does not occur due to rapid changes in temperature.
		Check that the Units are not in direct sunlight.	Not in direct sunlight	Protect the Units if necessary.
		Check for accumulation of dirt, dust, salt, or metal powder.	No accumulation	Clean and protect the Units if necessary.
		Check for water, oil, or chemical sprays hitting the Units.	No spray	Clean and protect the Units if necessary.
		Check for corrosive or flammable gases in the area.	No corrosive or flammable gases	Check by smell or use a gas sensor.
		Check that the Units are not subject to direct vibration or shock.	Vibration resistance and shock resistance must be within specifications.	Install cushioning or shock absorbing equipment if necessary.
		Check for noise sources nearby the Units.	No significant noise sources	Either separate the Units and noise source or protect the Units.
4	Installation and wiring	Check that the DIN Track mounting hooks on all Units are securely locked.	No looseness	Securely lock all DIN Track mounting hooks.
		Check that cable connectors are fully inserted and locked.	No looseness	Correct any improperly installed connectors.
		Check that the screws on the End Plates (PFP-M) are tight.	No looseness	Tighten loose screws with a Phillips screwdriver.
		Check that each Unit is connected along the hookup guides, and fully inserted until it contacts the DIN Track.	The Units must be connected and securely in place on the DIN Track.	Connect each NX Unit along the hookup guides, and insert each NX Unit until it contacts the DIN Track.
		Check for 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

- Flat-blade screwdriver
- Phillips screwdriver
- Voltage tester or digital voltmeter
- Industrial alcohol and clean cotton cloth

Tools Required Occasionally

- Oscilloscope
- Thermometer and hygrometer

10-2 Maintenance Procedures

If the inspection reveals any problems that require you to replace a Safety Unit, observe the following precautions.

- Never disassemble, repair, or modify a Safety 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 Safety 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.

Remove the faulty Unit, and then replace and wire the new Unit.
Refer to *Section 3 Part Names and Functions* and to *Section 5 Installation and Wiring* for information on installing, removing, and wiring Units.

The following sections give the procedures to replace the Safety CPU Unit and Safety I/O Units.

Precautions for Correct Use

The backup/restore functions of the NJ-series CPU Unit do not apply to Safety Control Units. Use the Sysmac Studio when you replace a Unit.

10-2-1 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-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-series CPU Unit

1 Connect the Sysmac Studio online to the NJ-series CPU Unit.

When you add a new Safety CPU Unit to the system, a Slave Initialization Error will occur in the NJ-series CPU Unit.
Reset the error from the Troubleshooting Dialog Box.

2 Select ***Synchronization*** from the Controller Menu. Click the **Transer to Controller** Button in the Synchronization Window to transfer the Slave Terminal configuration information from the computer to the Safety CPU Unit.

Refer to *8-2-2 Transferring Configuration Information to an NJ-series CPU Unit over a USB Connection or Ethernet Connection* on page 8-7 for a detailed procedure.

3 Select the Safety CPU Unit as the Controller.

The Sysmac Studio goes online with the Safety CPU Unit.

4 Place the Safety CPU Unit in DEBUG mode.

Refer to *8-4 Changing to DEBUG Mode* on page 8-15 for a detailed procedure.

5 With the Controller set to the Safety CPU Unit, select ***Safety Validation*** from the Controller Menu to transfer the safety programs to the non-volatile memory in the Safety CPU Unit.

Refer to *8-8-1 Performing Safety Validation* on page 8-34 for a detailed procedure.

6 Change the operating mode of the Safety CPU Unit to RUN mode. Or, cycle the power supply to Safety CPU Unit.

Refer to *8-8-2 Changing to RUN Mode* on page 8-36 for a detailed procedure.

● When Sysmac Studio Is Connected to EtherCAT Coupler Unit

1 Place the Sysmac Studio online with the EtherCAT Coupler Unit.

When you add a new Safety CPU Unit to the system, a Slave Initialization Error will occur in the EtherCAT Coupler Unit.
Reset the error from the Troubleshooting Dialog Box.

2 Right-click the EtherCAT Coupler Unit and select ***Transfer to Coupler*** from the menu to transfer the Slave Terminal configuration information to the Safety CPU Unit.

Refer to *8-2-3 Transferring Configuration Information to the EtherCAT Coupler Unit When Connected to the USB Port* on page 8-8 for a detailed procedure.

3 Select the Safety CPU Unit as the Controller.

The Sysmac Studio goes online with the Safety CPU Unit.

4 Place the Safety CPU Unit in DEBUG mode.

Refer to *8-4 Changing to DEBUG Mode* on page 8-15 for a detailed procedure.

5 With the Controller set to the Safety CPU Unit, select ***Safety Validation*** from the Controller Menu to transfer the safety programs to the non-volatile memory.

Refer to *8-8-1 Performing Safety Validation* on page 8-34 for a detailed procedure.

6 Change the operating mode of the Safety CPU Unit to RUN mode. Or, cycle the power supply to Safety CPU Unit.

Refer to *8-8-2 Changing to RUN Mode* on page 8-36 for a detailed procedure.

Checking after Replacing a Safety Unit

- After a Unit is replaced, make sure that the intended data was transferred to the Safety CPU Unit by using the following methods.
  - Make sure that the safety signature that is shown in the Properties Dialog Box for the safety project is the same as the safety signature that is shown in the Controller Status Pane.
- 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 the Safety CPU Unit due to the replacement work.

10-2-2 Replacing a Safety I/O Unit

● Precautions before Replacing the Unit

- The replaced Safety I/O Unit must be in the default status before the replacement.
  If you are unsure of whether the Safety I/O Unit is in its default state, perform the Clear All Memory operation for all Safety I/O Units that were replaced. Refer to *8-11 Restarting and Clearing All Memory* on page 8-40 for detailed operating 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-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 Unit.
2. Remove the Safety I/O Unit to replace.
3. Mount the new Safety I/O Unit.
4. Return the terminal block to the new Safety I/O Unit.
5. Change the operating mode of the Safety CPU Unit to RUN mode. Or, cycle the power supply to Safety CPU Unit.

   Refer to *8-8-2 Changing to RUN Mode* on page 8-36 for a detailed procedure.

● Checking after Replacing a Safety I/O Unit

- After a Unit is replaced, make sure that the intended data was transferred to the Safety CPU Unit by using the following methods.
  - Make sure that the safety signature that is shown in the Properties Dialog Box for the project is the same as the safety signature that is shown in the Controller Status Pane.
- 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 Unit, 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 EtherCAT Coupler Unit is set to *Setting* = *Actual device*, 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 EtherCAT Coupler Unit 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 *NX-series EtherCAT Coupler Unit User's Manual* (Cat. No. W519) for details on the serial number checking function for EtherCAT Coupler Units.

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 EtherCAT Slave Terminal that were not replaced
- The EtherCAT Coupler Unit to which the Safety CPU Unit or Safety I/O Unit where a communications error was detected is connected

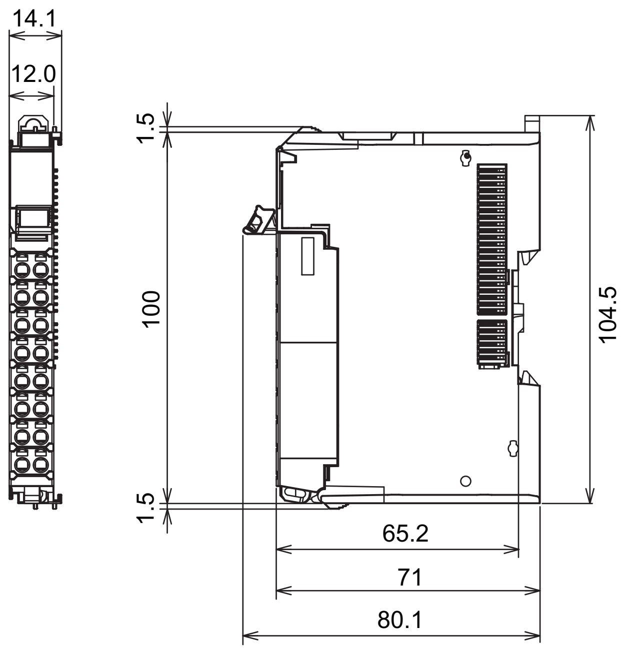
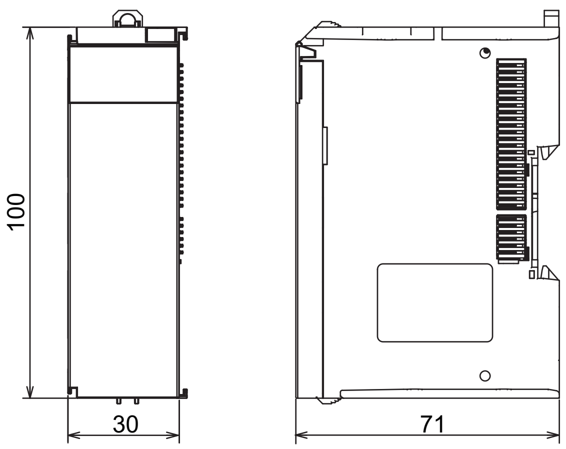
A

Appendix

The appendices provide the dimensions of the Safety Control Units, application examples, and other information.

A-1 Dimensions

The dimensions of the Safety Control Units are given in the following table.

Unit width	Models	Dimensions (mm)
12 mm	NX-SIH400 NX-SID800 NX-SOH200 NX-SOD400	
30 mm	NX-SL3300	

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 Units

Unit Information Object

This object gives the product information.

Index (hex)	Sub-index (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	NX-SL3300	---	---	ARRAY [0..11] OF BYTE	RO	Not possible.	---
	03	Device Type	00000A00 hex	---	---	UDINT	RO	Not possible.	---
	04	Product Code	00A03300 hex	---	---	UDINT	RO	Not possible.	---
	05	Vendor Code	1	---	---	UDINT	RO	Not possible.	---
	06	Unit Version	*1	---	---	UDINT	RO	Not possible.	---
	07	Serial Number	*2	00000000 to FFFFFFFF hex	---	UDINT	RO	Not possible.	---
1001	---	Production Info	---	---	---	---	---	Not possible.	---
	00	Number of Entries	4	4	---	USINT	RO	Not possible.	---
	01	Lot Number	*3	00000000 to FFFFFFFF hex	---	UDINT	RO	Not possible.	---
	02	Hardware Version	*4	---	---	ARRAY [0..19] OF BYTE	RO	Not possible.	---
	03	Software Version	*5	---	---	ARRAY [0..19] OF BYTE	RO	Not possible.	---

*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 version 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 32: 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.

Index (hex)	Sub-index (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	Status Monitoring Data for the Safety CPU Unit	0000 hex	0000 to 000F hex*1	---	WORD	RO	Possible.	---

*1. The details of the Status Monitoring Data for the Safety CPU Unit are as follows:
Bit 0: Safety programs are operating (no errors) and all safety master connections are established.
Bit 1: Safety programs are operating.
Bit 2: No event with a level of minor fault or higher currently exists for the safety programs.
Bit 3: All safety master connections are established.
Bits 4 to 15: Reserved

A-2-3 NX-SID800 Safety Input Unit

Unit Information Object

This object gives the product information.

Index (hex)	Sub-index (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	NX-SID800	---	---	ARRAY [0..11] OF BYTE	RO	Not possible.	---
	03	Device Type	00000A01 hex	---	---	UDINT	RO	Not possible.	---
	04	Product Code	00A10800 hex	---	---	UDINT	RO	Not possible.	---
	05	Vendor Code	1	---	---	UDINT	RO	Not possible.	---
	06	Unit Version	*1	---	---	UDINT	RO	Not possible.	---
	07	Serial Number	*2	00000000 to FFFFFFFF hex	---	UDINT	RO	Not possible.	---
1001	---	Production Info	---	---	---	---	---	---	---
	00	Number of Entries	4	4	---	USINT	RO	Not possible.	---
	01	Lot Number	*3	00000000 to FFFFFFFF hex	---	UDINT	RO	Not possible.	---
	02	Hardware Version	*4	---	---	ARRAY [0..19] OF BYTE	RO	Not possible.	---
	03	Software Version	*5	---	---	ARRAY [0..19] OF BYTE	RO	Not possible.	---

*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 version 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 32: 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.

Index (hex)	Sub-index (hex)	Object name	Default value	Data range	Unit	Data type	Acc ess	I/O allo-cation	Data attri-bute
6000	---	FSoE Slave Frame Elements	---	---	---	---	---	---	---
	00	Number of Entries	3	3	---	USINT	RO	Not possible.	---
	01	FSoE Slave CMD	00 hex	00 to FF hex	---	BYTE	RO	Possible.	---
	02	FSoE Slave Conn_ID	0000 hex	0000 to FFFF hex	---	WORD	RO	Possible.	---
	03	FSoE Slave CRC_0	0000 hex	0000 to FFFF hex	---	WORD	RO	Possible.	---
6001	---	Safety Input Data	---	---	---	---	---	---	---
	00	Number of Entries	1	1	---	USINT	RO	Not possible.	---
	01	Safety Input 1st Word	0000 hex	0000 to FFFF hex	---	WORD	RO	Possible.	---
6002	---	Standard Input Data	---	---	---	---	---	---	---
	00	Number of Entries	2	2	---	USINT	RO	Not possible.	---
	01	Standard Input 1st Word	0000 hex	0000 to FFFF hex*1	---	WORD	RO	Possible.	---
	02	Standard Input 2nd Byte	00 hex	00 to FF hex*2	---	BYTE	RO	Possible.	---

*1. The details of the Standard Input 1st Word are as follows:

Bit 0: Safety input data 00
Bit 1: Safety input data 01
Bit 2: Safety input data 02
Bit 3: Safety input data 03
Bit 4: Safety input data 04
Bit 5: Safety input data 05
Bit 6: Safety input data 06
Bit 7: Safety input data 07
Bit 8: Safety connection status
Bit 9: Safety I/O terminal status
Bit 10: Unit normal status
Bit 11: I/O power supply error flag
Bits 12 to 15: Reserved

*2. The details of the Standard Input 2nd Byte are as follows:

Bit 0: Safety input terminal status 00
Bit 1: Safety input terminal status 01
Bit 2: Safety input terminal status 02
Bit 3: Safety input terminal status 03
Bit 4: Safety input terminal status 04
Bit 5: Safety input terminal status 05
Bit 6: Safety input terminal status 06
Bit 7: Safety input terminal status 07

Index (hex)	Sub-index (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 possible.	---
	01	FSoE Master CMD	00 hex	00 to FF hex	---	BYTE	RW	Possible.	---
	02	FSoE Master Conn_ID	0000 hex	0000 to FFFF hex	---	WORD	RW	Possible.	---
	03	FSoE Master CRC_0	0000 hex	0000 to FFFF hex	---	WORD	RW	Possible.	---
7001	---	Safety Output Data	---	---	---	---	---	---	---
	00	Number of Entries	1	1	---	USINT	RO	Not possible.	---
	01	Safety Output 1st Word	0000 hex	0000 to FFFF hex	---	WORD	RW	Possible.	---
7002	---	Standard Output Data	---	---	---	---	---	---	---
	00	Number of Entries	2	2	---	USINT	RO	Not possible.	---
	01	Standard Output 1st Word	0000 hex	0000 hex*1	---	WORD	RW	Possible.	---
	02	Standard Output 2nd Byte	00 hex	00 hex*2	---	BYTE	RW	Possible.	---

*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)	Sub-index (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 possible.	---
	01	Safety Address	0000 hex	0000 to FFFF hex	---	UINT	RO	Not possible.	---

A-2-4 NX-SIH400 Safety Input Unit

Unit Information Object

This object gives the product information.

Index (hex)	Sub-index (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	NX-SIH400	---	---	ARRAY [0..11] OF BYTE	RO	Not possible.	---
	03	Device Type	00000A02 hex	---	---	UDINT	RO	Not possible.	---
	04	Product Code	00A20400 hex	---	---	UDINT	RO	Not possible.	---
	05	Vendor Code	1	---	---	UDINT	RO	Not possible.	---
	06	Unit Version	*1	---	---	UDINT	RO	Not possible.	---
	07	Serial Number	*2	00000000 to FFFFFFFF hex	---	UDINT	RO	Not possible.	---
1001	---	Production Info	---	---	---	---	---	---	---
	00	Number of Entries	4	4	---	USINT	RO	Not possible.	---
	01	Lot Number	*3	00000000 to FFFFFFFF hex	---	UDINT	RO	Not possible.	---
	02	Hardware Version	*4	---	---	ARRAY [0..19] OF BYTE	RO	Not possible.	---
	03	Software Version	*5	---	---	ARRAY [0..19] OF BYTE	RO	Not possible.	---

*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 version 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.

Index (hex)	Sub-index (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 possible.	---
	01	FSoE Slave CMD	00 hex	00 to FF hex	---	BYTE	RO	Possible.	---
	02	FSoE Slave Conn_ID	0000 hex	0000 to FFFF hex	---	WORD	RO	Possible.	---
	03	FSoE Slave CRC_0	0000 hex	0000 to FFFF hex	---	WORD	RO	Possible.	---
6001	---	Safety Input Data	---	---	---	---	---	---	---
	00	Number of Entries	1	1	---	USINT	RO	Not possible.	---
	01	Safety Input 1st Byte	00 hex	00 to FF hex	---	BYTE	RO	Possible.	---
6002	---	Standard Input Data	---	---	---	---	---	---	---
	00	Number of Entries	2	2	---	USINT	RO	Not possible.	---
	01	Standard Input 1st Byte	00 hex	00 to FF hex*1	---	BYTE	RO	Possible.	---
	02	Standard Input 2nd Byte	00 hex	00 to FF hex*2	---	BYTE	RO	Possible.	---

*1. The details of the Standard Input 1st Byte are as follows:
Bit 0: Safety input data 00
Bit 1: Safety input data 01
Bit 2: Safety input data 02
Bit 3: Safety input data 03
Bit 4: Safety connection status
Bit 5: Safety I/O terminal status
Bit 6: Unit normal status
Bit 7: I/O power supply error flag

*2. The details of the Standard Input 2nd Byte are as follows:
Bit 0: Safety input terminal status 00
Bit 1: Safety input terminal status 01
Bit 2: Safety input terminal status 02
Bit 3: Safety input terminal status 03
Bits 4 to 7: Reserved

Index (hex)	Sub-index (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 possible.	---
	01	FSoE Master CMD	00 hex	00 to FF hex	---	BYTE	RW	Possible.	---
	02	FSoE Master Conn_ID	0000 hex	0000 to FFFF hex	---	WORD	RW	Possible.	---
	03	FSoE Master CRC_0	0000 hex	0000 to FFFF hex	---	WORD	RW	Possible.	---
7001	---	Safety Output Data	---	---	---	---	---	---	---
	00	Number of Entries	1	1	---	USINT	RO	Not possible.	---
	01	Safety Output 1st Byte	00 hex	00 to FF hex	---	BYTE	RW	Possible.	---
7002	---	Standard Output Data	---	---	---	---	---	---	---
	00	Number of Entries	2	2	---	USINT	RO	Not possible.	---
	01	Standard Output 1st Byte	00 hex	00 hex*1	---	BYTE	RW	Possible.	---
	02	Standard Output 2nd Byte	00 hex	00 hex*2	---	BYTE	RW	Possible.	---

*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)	Sub-index (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 possible.	---
	01	Safety Address	0000 hex	0000 to FFFF hex	---	UINT	RO	Not possible.	---

A-2-5 NX-SOD400 Safety Output Unit

Unit Information Object

This object gives the product information.

Index (hex)	Sub-index (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	NX-SOD400	---	---	ARRAY [0..11] OF BYTE	RO	Not possible.	---
	03	Device Type	00000A03 hex	---	---	UDINT	RO	Not possible.	---
	04	Product Code	00A30400 hex	---	---	UDINT	RO	Not possible.	---
	05	Vendor Code	1	---	---	UDINT	RO	Not possible.	---
	06	Unit Version	*1	---	---	UDINT	RO	Not possible.	---
	07	Serial Number	*2	00000000 to FFFFFFFF hex	---	UDINT	RO	Not possible.	---
1001	---	Production Info	---	---	---	---	---	---	---
	00	Number of Entries	4	4	---	USINT	RO	Not possible.	---
	01	Lot Number	*3	00000000 to FFFFFFFF hex	---	UDINT	RO	Not possible.	---
	02	Hardware Version	*4	---	---	ARRAY [0..19] OF BYTE	RO	Not possible.	---
	03	Software Version	*5	---	---	ARRAY [0..19] OF BYTE	RO	Not possible.	---

*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 version 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.

Index (hex)	Sub-index (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 possible.	---
	01	FSoE Slave CMD	00 hex	00 to FF hex	---	BYTE	RO	Possible.	---
	02	FSoE Slave Conn_ID	0000 hex	0000 to FFFF hex	---	WORD	RO	Possible.	---
	03	FSoE Slave CRC_0	0000 hex	0000 to FFFF hex	---	WORD	RO	Possible.	---
6001	---	Safety Input Data	---	---	---	---	---	---	---
	00	Number of Entries	1	1	---	USINT	RO	Not possible.	---
	01	Safety Input 1st Byte	00 hex	00 to FF hex	---	BYTE	RO	Possible.	---
6002	---	Standard Input Data	---	---	---	---	---	---	---
	00	Number of Entries	2	2	---	USINT	RO	Not possible.	---
	01	Standard Input 1st Byte	00 hex	00 to FF hex*1	---	BYTE	RO	Possible.	---
	02	Standard Input 2nd Byte	00 hex	00 to FF hex*2	---	BYTE	RO	Possible.	---

*1. The details of the Standard Input 1st Byte are as follows:

Bit 0: Safety output monitor 00
Bit 1: Safety output monitor 01
Bit 2: Safety output monitor 02
Bit 3: Safety output monitor 03
Bit 4: Safety connection status
Bit 5: Safety output terminal status
Bit 6: Unit normal status
Bit 7: I/O power supply error flag

*2. The details of the Standard Input 2nd Byte are as follows:

Bit 0: Safety output terminal status 00
Bit 1: Safety output terminal status 01
Bit 2: Safety output terminal status 02
Bit 3: Safety output terminal status 03
Bits 4 to 7: Reserved

Index (hex)	Sub-index (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 possible.	---
	01	FSoE Master CMD	00 hex	00 to FF hex	---	BYTE	RW	Possible.	---
	02	FSoE Master Conn_ID	0000 hex	0000 to FFFF hex	---	WORD	RW	Possible.	---
	03	FSoE Master CRC_0	0000 hex	0000 to FFFF hex	---	WORD	RW	Possible.	---
7001	---	Safety Output Data	---	---	---	---	---	---	---
	00	Number of Entries	1	1	---	USINT	RO	Not possible.	---
	01	Safety Output 1st Byte	00 hex	00 to FF hex	---	BYTE	RW	Possible.	---
7002	---	Standard Output Data	---	---	---	---	---	---	---
	00	Number of Entries	2	2	---	USINT	RO	Not possible.	---
	01	Standard Output 1st Byte	00 hex	00 to FF hex	---	BYTE	RW	Possible.	---
	02	Standard Output 2nd Byte	00 hex	00 to FF hex	---	BYTE	RW	Possible.	---

Other Objects

This section lists other objects.

Index (hex)	Sub-index (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 possible.	---
	01	Safety Address	0000 hex	0000 to FFFF hex	---	UINT	RO	Not possible.	---

A-2-6 NX-SOH200 Safety Output Unit

Unit Information Object

This object gives the product information.

Index (hex)	Sub-index (hex)	Object name	Default value	Data range	Unit	Data type	Acc ess	I/O allo-cation	Data attri-bute
1000	---	NX Bus Identity	---	---	---	---	---	---	---
	00	Number of Entries	7	7	---	USINT	RO	Not pos-sible.	---
	02	Model	NX-SOH200	---	---	ARRAY [0..11] OF BYTE	RO	Not pos-sible.	---
	03	Device Type	00000A04 hex	---	---	UDINT	RO	Not pos-sible.	---
	04	Product Code	00A40200 hex	---	---	UDINT	RO	Not pos-sible.	---
	05	Vendor Code	1	---	---	UDINT	RO	Not pos-sible.	---
	06	Unit Version	*1	---	---	UDINT	RO	Not pos-sible.	---
	07	Serial Number	*2	00000000 to FFFFFFFF hex	---	UDINT	RO	Not pos-sible.	---
1001	---	Production Info	---	---	---	---	---	---	---
	00	Number of Entries	4	4	---	USINT	RO	Not pos-sible.	---
	01	Lot Number	*3	00000000 to FFFFFFFF hex	---	UDINT	RO	Not pos-sible.	---
	02	Hardware Version	*4	---	---	ARRAY [0..19] OF BYTE	RO	Not pos-sible.	---
	03	Software Version	*5	---	---	ARRAY [0..19] OF BYTE	RO	Not pos-sible.	---

*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 version 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.

Index (hex)	Sub-index (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 possible.	---
	01	FSoE Slave CMD	00 hex	00 to FF hex	---	BYTE	RO	Possible.	---
	02	FSoE Slave Conn_ID	0000 hex	0000 to FFFF hex	---	WORD	RO	Possible.	---
	03	FSoE Slave CRC_0	0000 hex	0000 to FFFF hex	---	WORD	RO	Possible.	---
6001	---	Safety Input Data	---	---	---	---	---	---	---
	00	Number of Entries	1	1	---	USINT	RO	Not possible.	---
	01	Safety Input 1st Byte	00 hex	00 to FF hex	---	BYTE	RO	Possible.	---
6002	---	Standard Input Data	---	---	---	---	---	---	---
	00	Number of Entries	2	2	---	USINT	RO	Not possible.	---
	01	Standard Input 1st Byte	00 hex	00 to FF hex*1	---	BYTE	RO	Possible.	---
	02	Standard Input 2nd Byte	00 hex	00 to FF hex*2	---	BYTE	RO	Possible.	---

*1. The details of the Standard Input 1st Byte are as follows:

Bit 0: Safety output monitor 00

Bit 1: Safety output monitor 01

Bit 2: Safety connection status

Bit 3: Safety output terminal status

Bit 4: Unit normal status

Bit 5: I/O power supply error flag

Bits 6 and 7: Reserved

*2. The details of the Standard Input 2nd Byte are as follows:

Bit 0: Safety output terminal status 00

Bit 1: Safety output terminal status 01

Bits 2 to 7: Reserved

Index (hex)	Sub-index (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 possible.	---
	01	FSoE Master CMD	00 hex	00 to FF hex	---	BYTE	RW	Possible.	---
	02	FSoE Master Conn_ID	0000 hex	0000 to FFFF hex	---	WORD	RW	Possible.	---
	03	FSoE Master CRC_0	0000 hex	0000 to FFFF hex	---	WORD	RW	Possible.	---
7001	---	Safety Output Data	---	---	---	---	---	---	---
	00	Number of Entries	1	1	---	USINT	RO	Not possible.	---
	01	Safety Output 1st Byte	00 hex	00 to FF hex	---	BYTE	RW	Possible.	---

Index (hex)	Sub-index (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 possible.	---
	01	Standard Output 1st Byte	00 hex	00 to FF hex	---	BYTE	RW	Possible.	---
	02	Standard Output 2nd Byte	00 hex	00 to FF hex	---	BYTE	RW	Possible.	---

Other Objects

This section lists other objects.

Index (hex)	Sub-index (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 possible.	---
	01	Safety Address	0000 hex	0000 to FFFF hex	---	UINT	RO	Not possible.	---

A-3 Application Examples

These examples show safety systems that use Safety Units.

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.

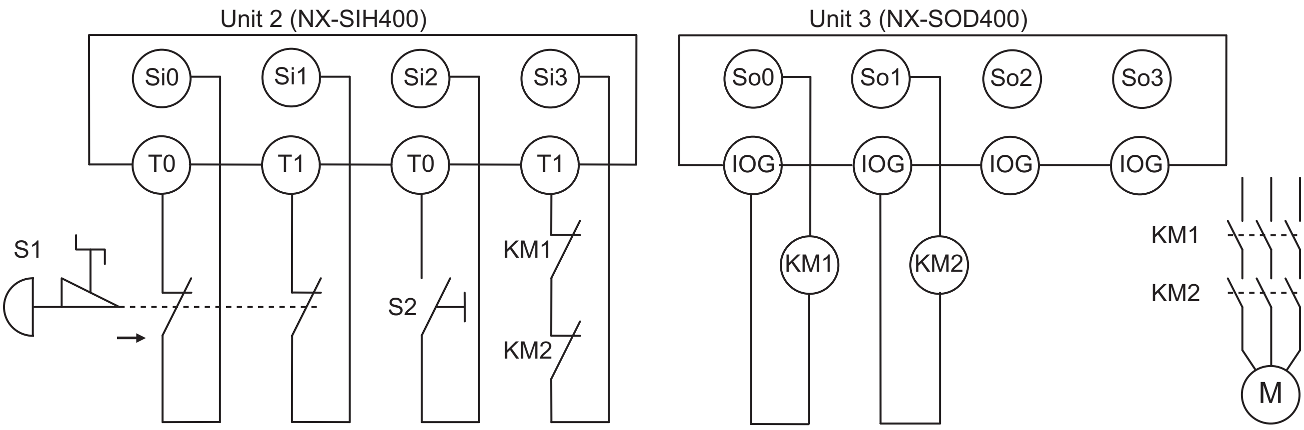
A-3-1 Emergency Stop Pushbutton Switches

Application Overview

Safety category/PL	Safety device	Stop category	Reset
Equivalent to 4/PLe	Emergency stop pushbutton	0	Manual

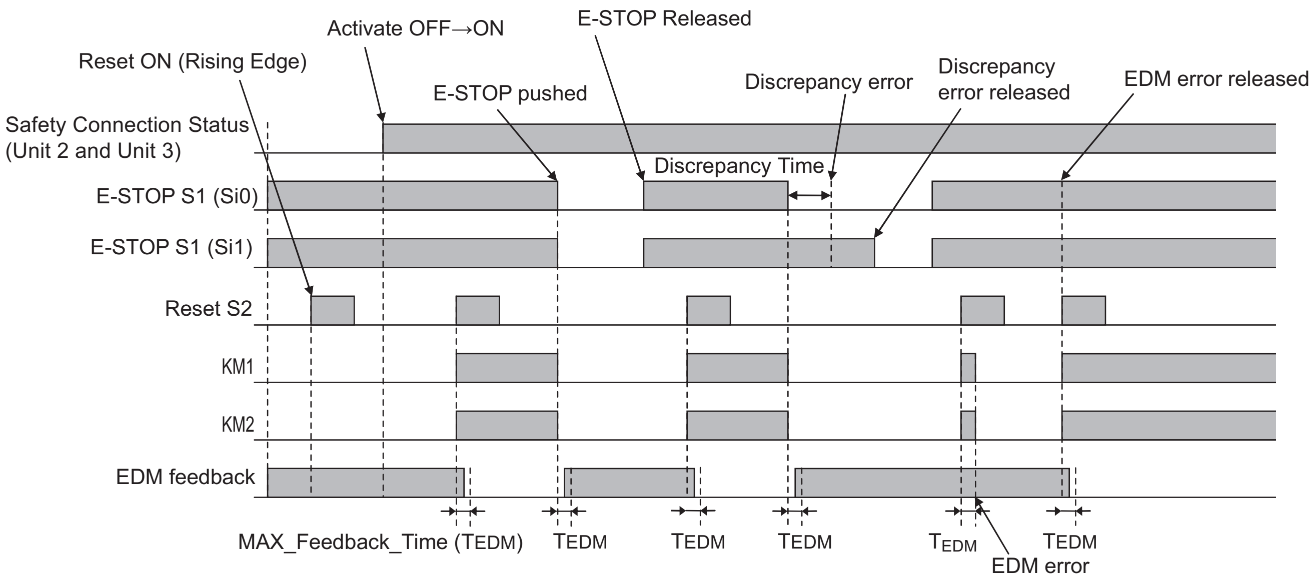
Motor M stops when emergency stop pushbutton S1 is pressed.

Wiring



S1: Emergency stop pushbutton
S2: Reset switch
KM1 and 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)

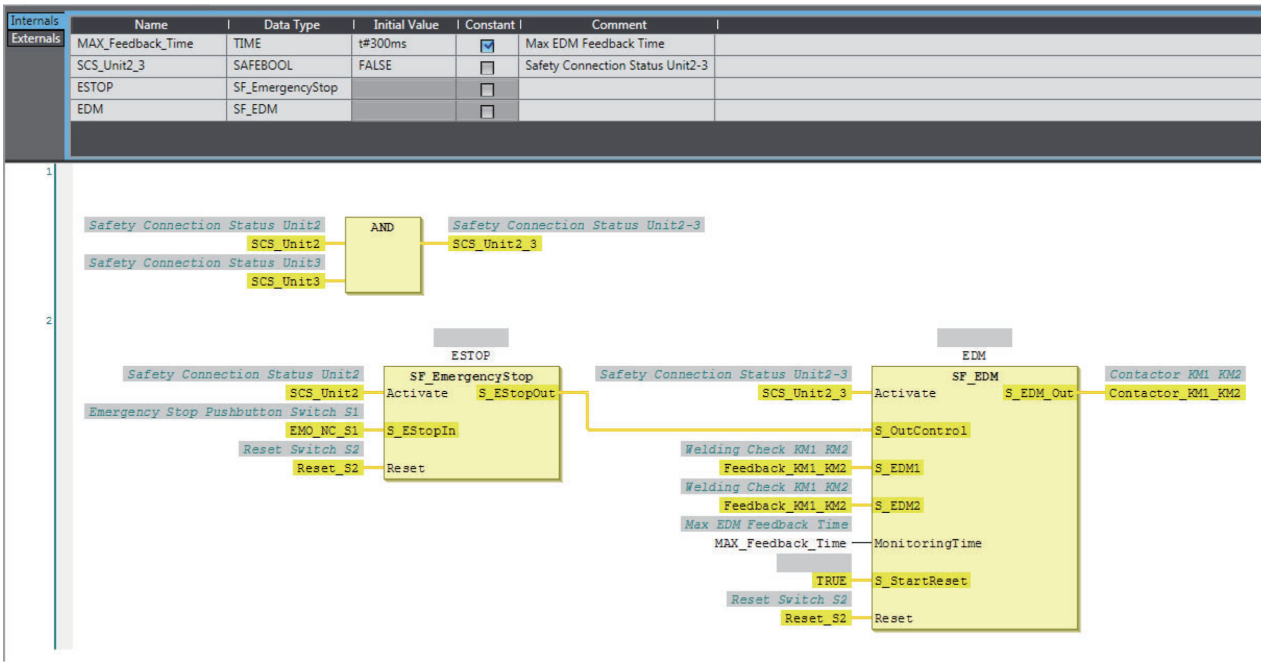
Node1/Unit3 : NX-SOD400 (N3 : Instance1)

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



Name	Data Type	Initial Value	Constant	Comment
MAX_Feedback_Time	TIME	t#300ms	☑	Max EDM Feedback Time
SCS_Unit2_3	SAFEBOOL	FALSE	☐	Safety Connection Status Unit2-3
ESTOP	SF_EmergencyStop		☐	
EDM	SF_EDM		☐	

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

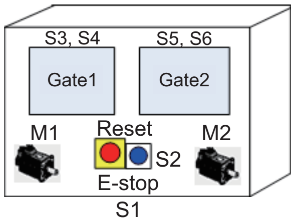
Application Overview

Safety category/PL	Safety device	Stop category	Reset
Equivalent to 4/PLe	Safety Limit Switches 1 and 2	0	Auto
	Emergency stop pushbutton	0	Manual

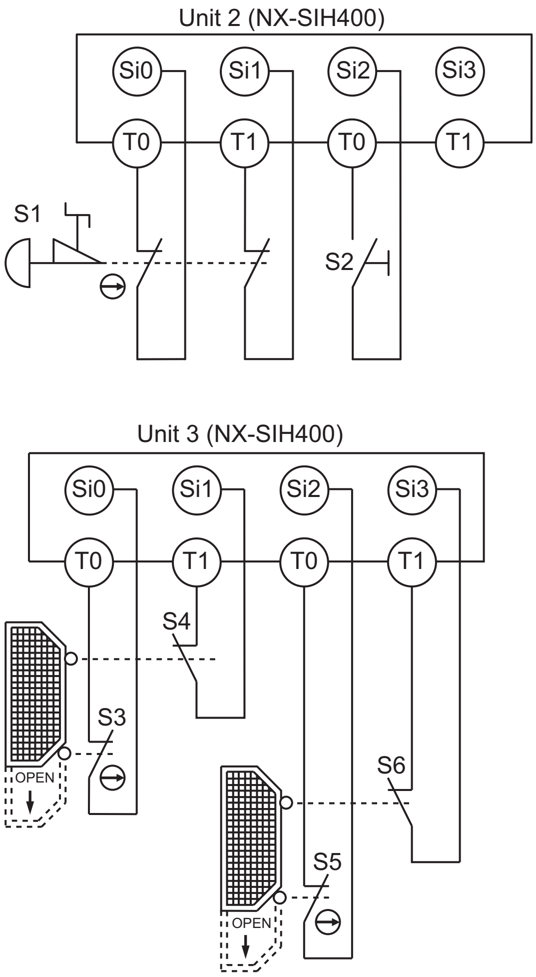
M1 stops when safety gate 1 (S3, S4) is opened.

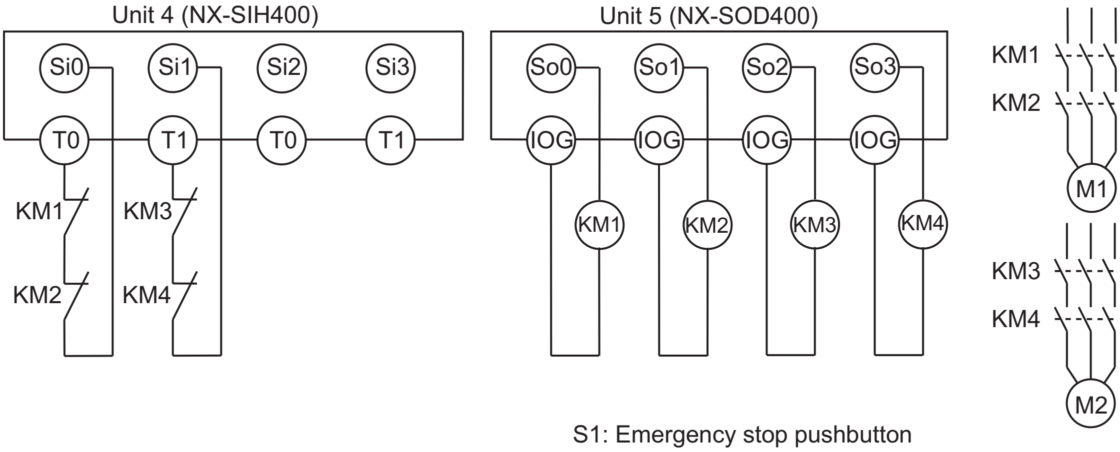
M2 stops when safety gate 2 (S5, S6) is opened.

Both M1 and M2 stop when the emergency stop pushbutton S1 is pressed.



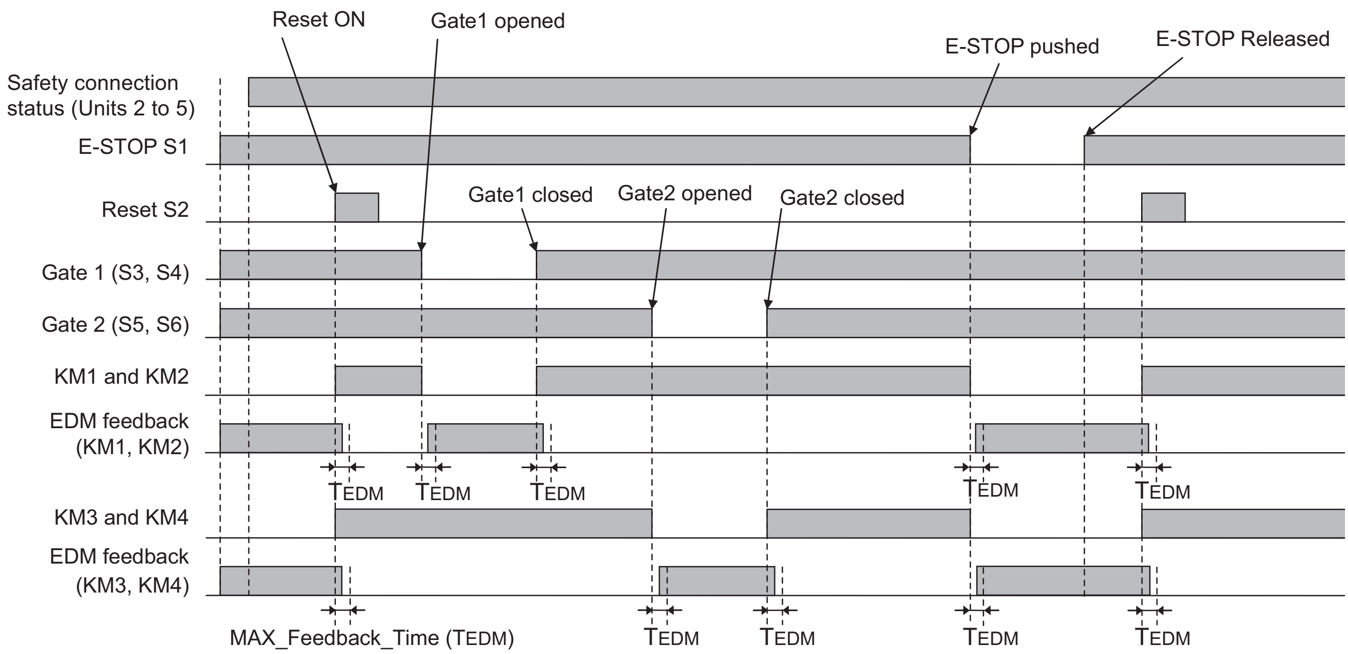
Wiring





S1: Emergency stop pushbutton
S2: Reset switch
S3 and S5: Safety limit switches
S4 and S6: Limit switches (NO contacts)
KM1 and KM2: Contactors
M1 and M2: 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
	Si 3					

Node1/Unit3 : NX-SIH400 (N3 : Instance1)

External Device	Channel	Discrepancy	On-Off	Off-On	Test Source	Comment
Mechanical Contact for Dual Channel Equivalent	Si 0	500ms	0ms	0ms	T0	Safety Switch(2NC)
	Si 1	500ms	0ms	0ms	T1	
Mechanical Contact for Dual Channel Equivalent	Si 2	500ms	0ms	0ms	T0	Safety Switch(2NC)
	Si 3	500ms	0ms	0ms	T1	

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					

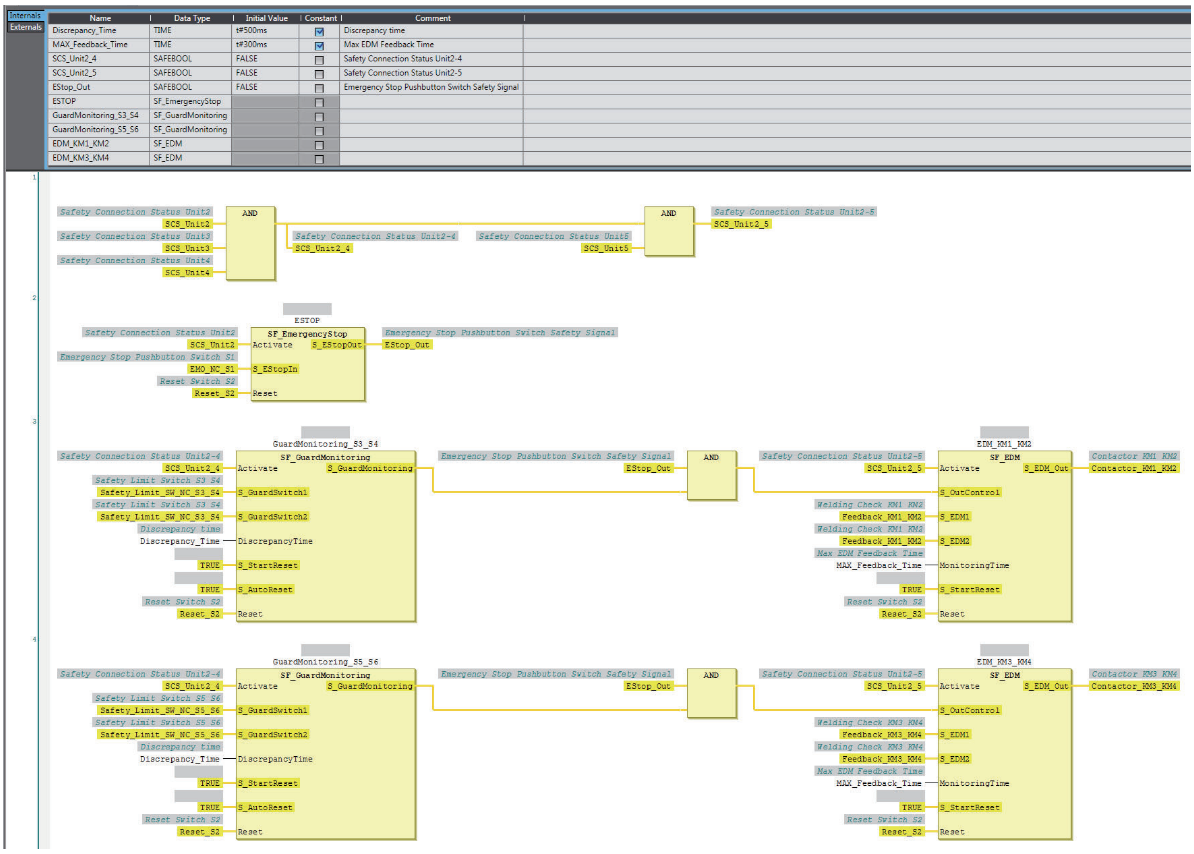
Node1/Unit5 : NX-SOD400 (N5 : Instance3)

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 Pushubutton 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			
	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_S4	Safety Limit Switch S3_S4	Global Variables
	Si01 Logical Value	R	SAFEBOOL			
	Si02 Logical Value	R	SAFEBOOL	Safety_Limit_SW_NC_S5_S6	Safety Limit Switch S5_S6	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-SIH400					
	▼ Safety Inputs and Status					
	Si00 Logical Value	R	SAFEBOOL	Feedback_KM1_KM2	Welding Check KM1_KM2	Global Variables
	Si01 Logical Value	R	SAFEBOOL	Feedback_KM3_KM4	Welding Check KM3_KM4	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	Contactor KM1_KM2	Global Variables
	So01 Output Value	W	SAFEBOOL			
	So02 Output Value	W	SAFEBOOL	Contactor_KM3_KM4	Contactor KM3_KM4	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.

A-3-3 Safety Laser Scanners

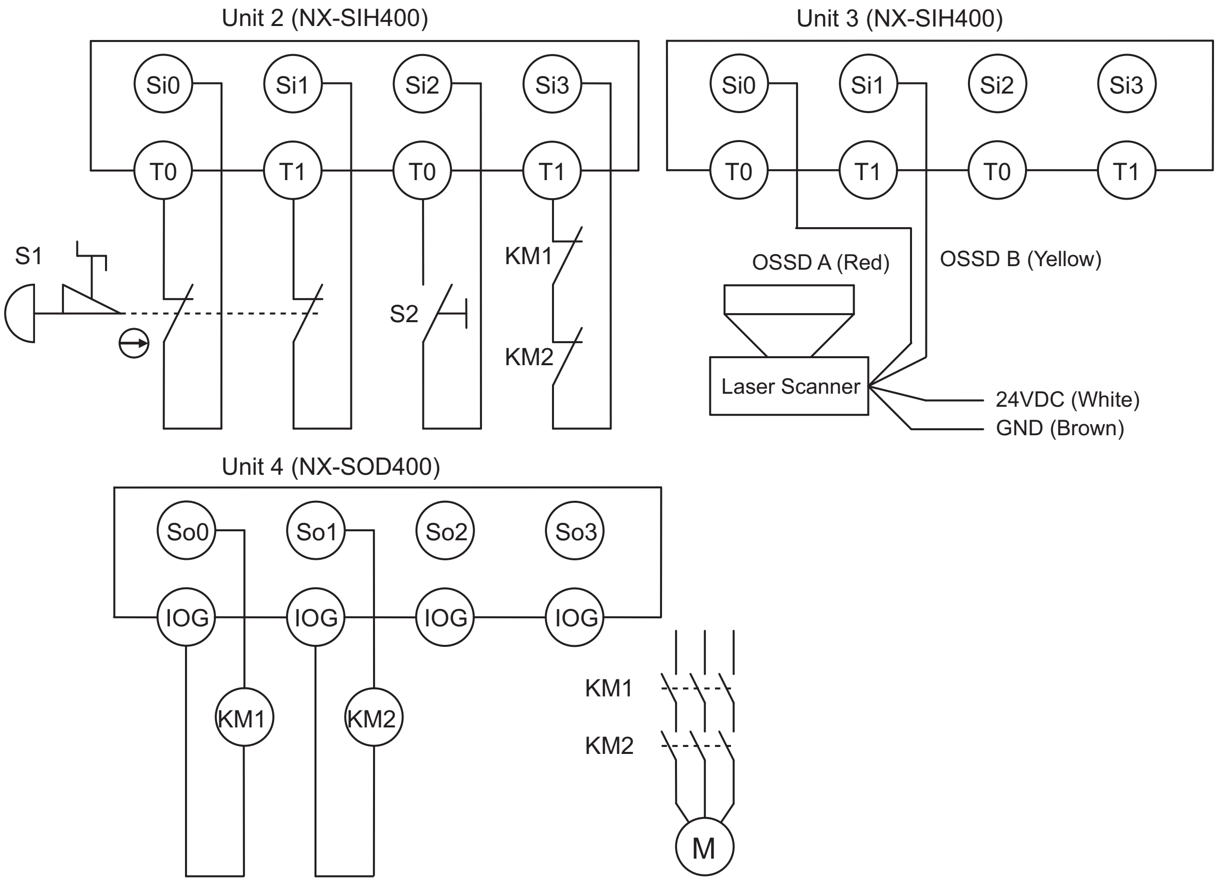
Application Overview

Safety category/PL	Safety device	Stop category	Reset
Equivalent to 3/PLd	Laser scanner	0	Auto
	Emergency stop pushbutton	0	Manual

AGV stops when emergency stop pushbutton S1 is pressed.

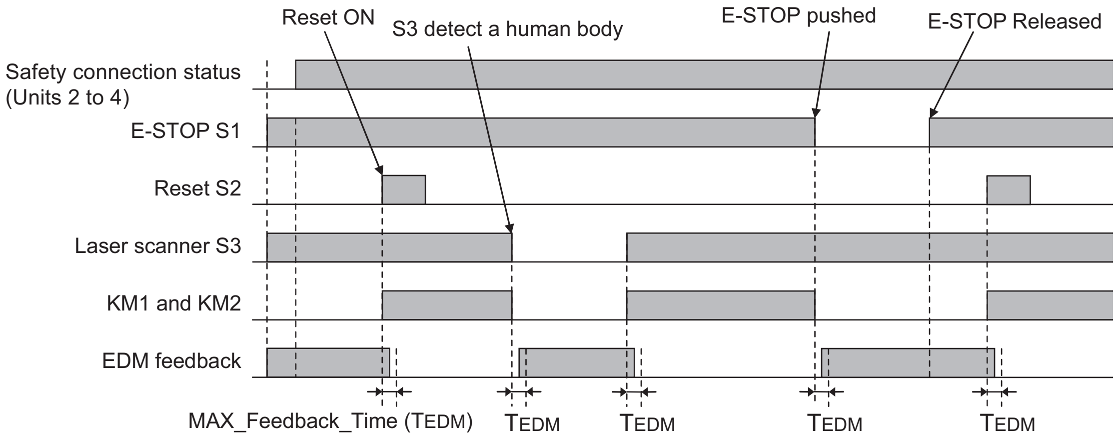
AGV stops when Laser scanner S3 detect that persons or objects approach into the safety zone.

Wiring



S1: Emergency stop pushbutton
S2: Reset switch
S3: Safety laser scanner
KM1 and 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	

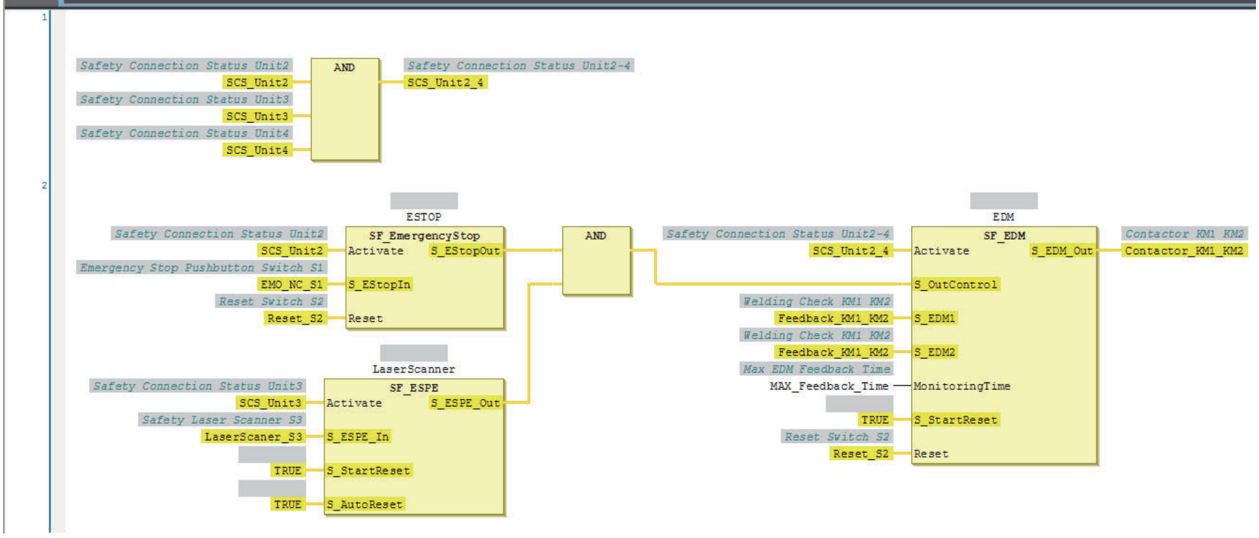
● I/O Map Settings

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

Internals | Externals

Name	Data Type	Initial Value	Constant	Comment
MAX_Feedback_Time	TIME	t#300ms	☑	Max EDM Feedback Time
SCS_Unit2_4	SAFEBOOL	FALSE	☐	Safety Connection Status Unit2-4
ESTOP	SF_EmergencyStop		☐	
LaserScanner	SF_ESPE		☐	
EDM	SF_EDM		☐	



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 Electromagnetic Locks and User Mode Switches

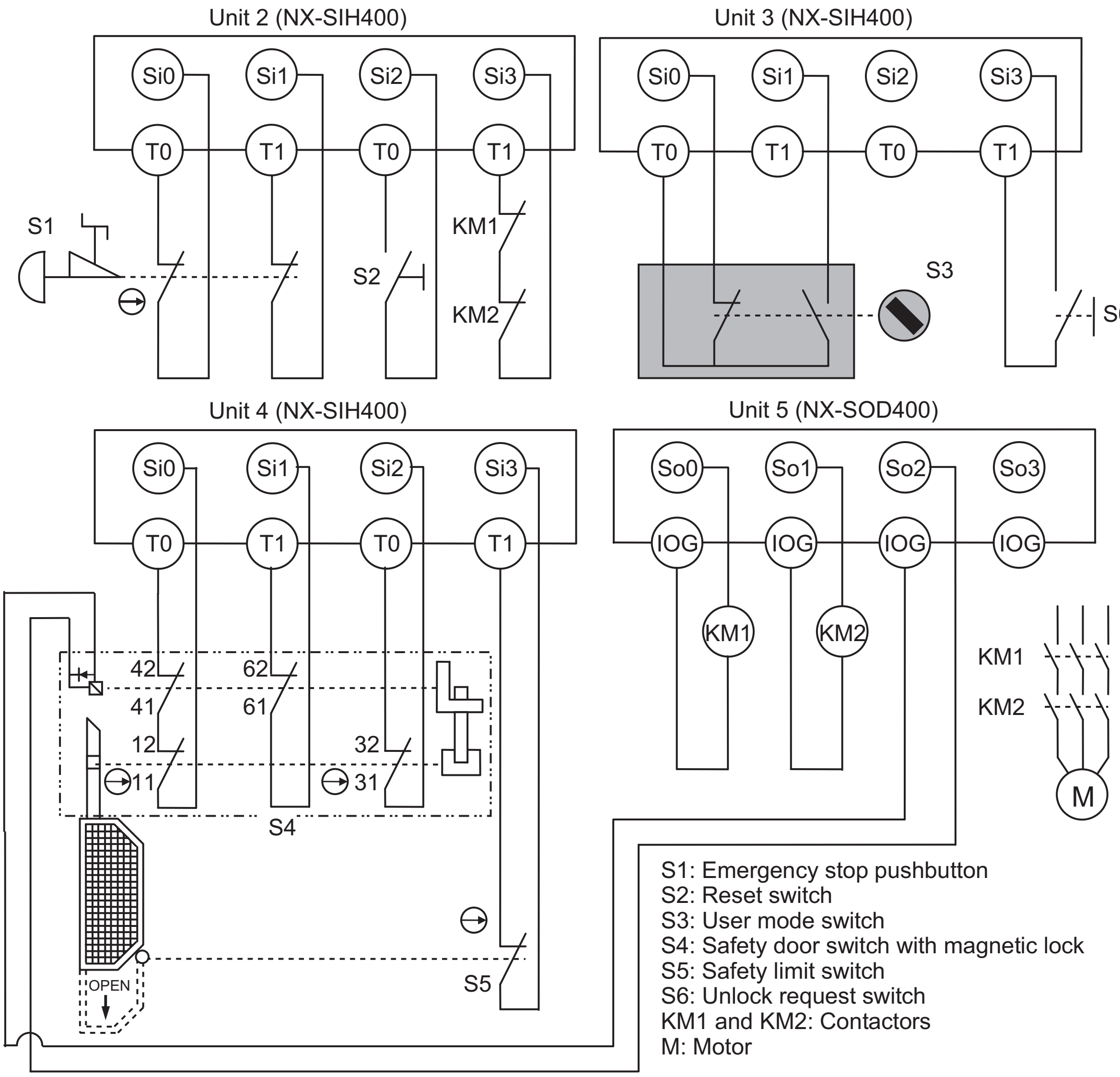
Application Overview

Safety category/PL	Safety device	Stop category	Reset
Equivalent to 4/PLe	• Emergency stop pushbutton • Safety door switch with magnetic lock (mechanical lock type) • User mode switch	0	Manual

The S4 and S5 safety doors 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.

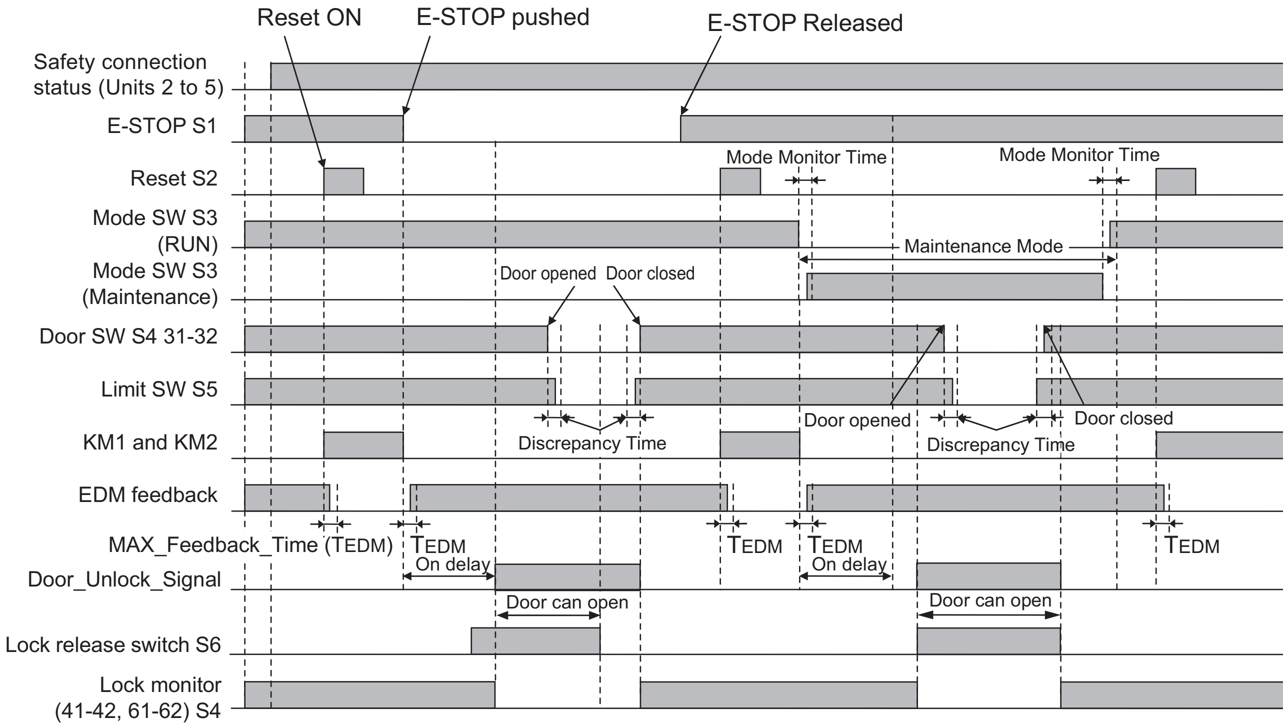
The outputs also go OFF when emergency pushbutton S1 is pressed.

Wiring



S1: Emergency stop pushbutton
S2: Reset switch
S3: User mode switch
S4: Safety door switch with magnetic lock
S5: Safety limit switch
S6: Unlock request switch
KM1 and 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 Dual Channel Equivalent	Si 0	500ms	0ms	0ms	T0	Safety Switch(2NC)
	Si 1	500ms	0ms	0ms	T1	
Mechanical Contact for Dual Channel Equivalent	Si 2	500ms	0ms	0ms	T0	Safety Switch(2NC)
	Si 3	500ms	0ms	0ms	T1	

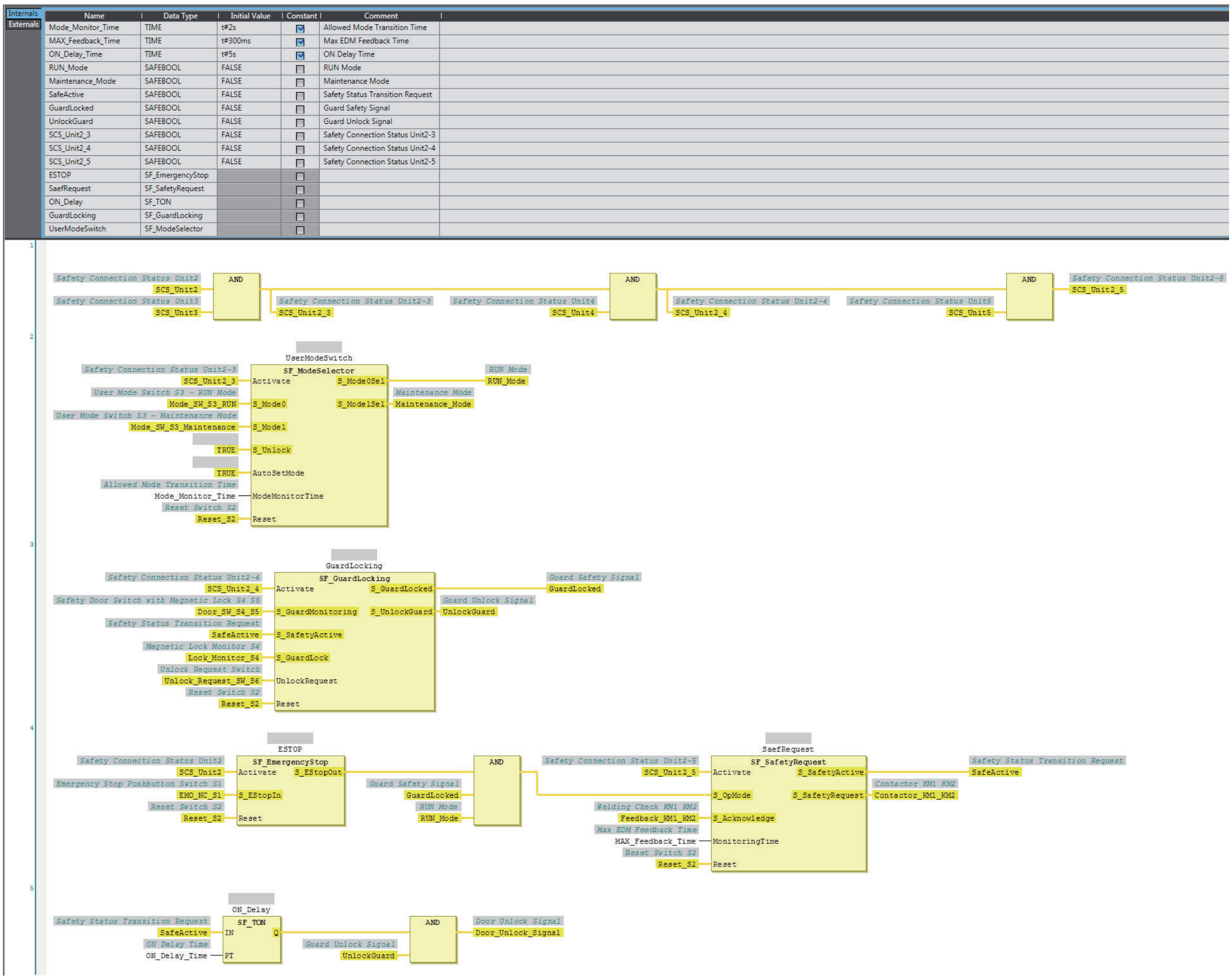
Node1/Unit5 : NX-SOD400 (N5 : Instance3)

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	
	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	Mode_SW_S3_RUN	User Mode Switch S3 - RUN Mode	Global Variables
	Si01 Logical Value	R	SAFEBOOL	Mode_SW_S3_Maintenance	User Mode Switch S3 - Maintenance Mode	Global Variables
	Si02 Logical Value	R	SAFEBOOL			
	Si03 Logical Value	R	SAFEBOOL	Unlock_Request_SW_S6	Unlock Request Switch	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	Magnetic Lock Monitor S4	Global Variables
	Si01 Logical Value	R	SAFEBOOL			
	Si02 Logical Value	R	SAFEBOOL	Door_SW_S4_S5	Safety Door Switch with Magnetic Lock S4_S5	Global Variables
	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	Contactor KM1_KM2	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
RUN_Mode	SAFEBOOL	FALSE	☐	RUN Mode
Maintenance_Mode	SAFEBOOL	FALSE	☐	Maintenance Mode
SafeActive	SAFEBOOL	FALSE	☐	Safety Status Transition Request
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
ESTOP	SF_EmergencyStop		☐	
SaefRequest	SF_SafetyRequest		☐	
ON_Delay	SF_TON		☐	
GuardLocking	SF_GuardLocking		☐	
UserModeSwitch	SF_ModeSelector		☐	

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-3-5 Enable Switches

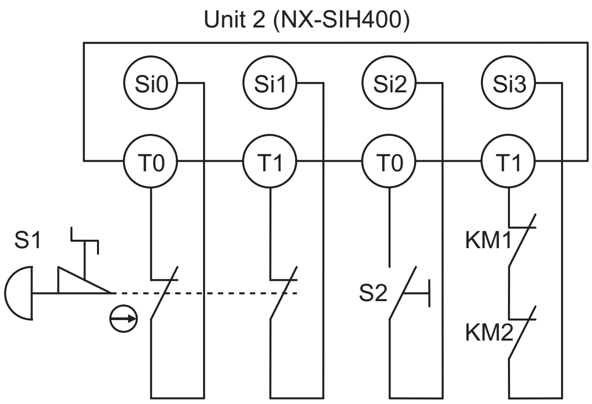
Application Overview

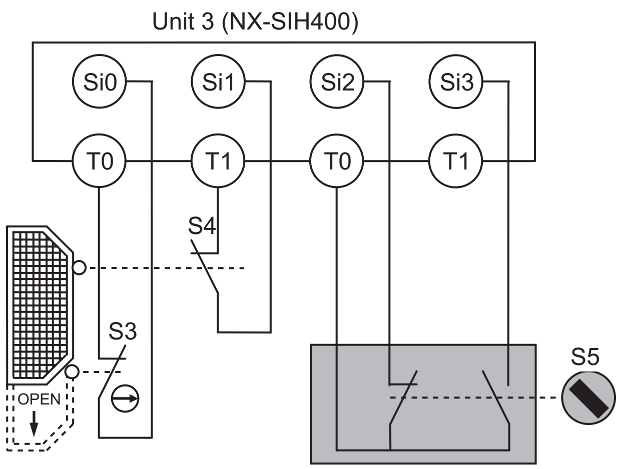
Safety category/PL	Safety device	Stop category	Reset
Equivalent to 4/PLe	• Emergency stop pushbutton • Safety limit switch • User mode switch • Enable switch	0	Manual

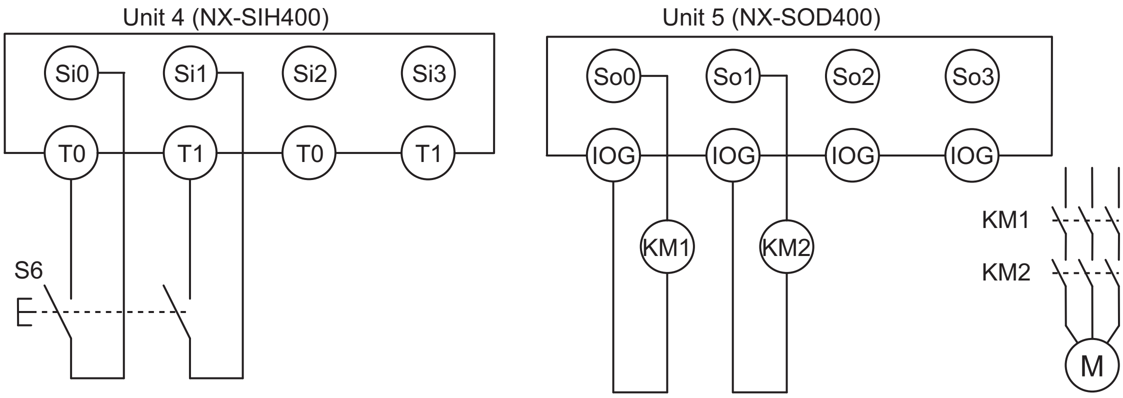
When the safety gate (S3, S4) is opened or user mode switch (S5) is Maintenance mode, M is stopped.

However, even if the S5 user mode switch is set to maintenance mode, motor M will operate if the S6 enable switch is ON.

Wiring

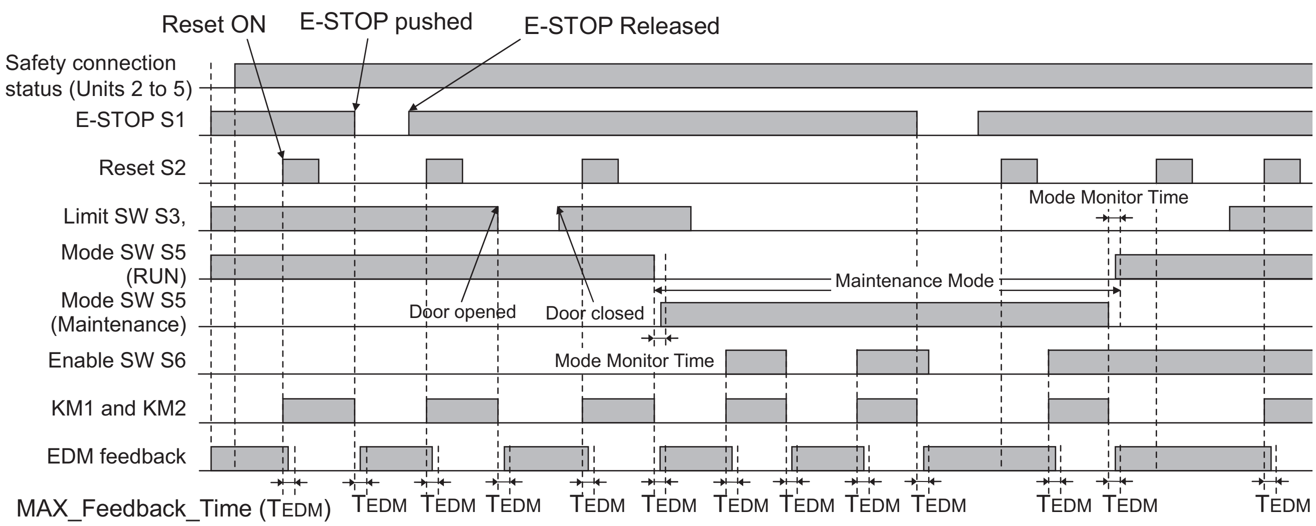






S1: Emergency stop pushbutton
S2: Reset switch
S3 and S4: Safety limit switches
S5: User mode switch
S6: Enable switch
KM1 and KM2: Contactors
M: Motor

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 Dual Channel Equivalent	Si 0	500ms	0ms	0ms	T0	Safety Switch(2NC)
	Si 1	500ms	0ms	0ms	T1	
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					

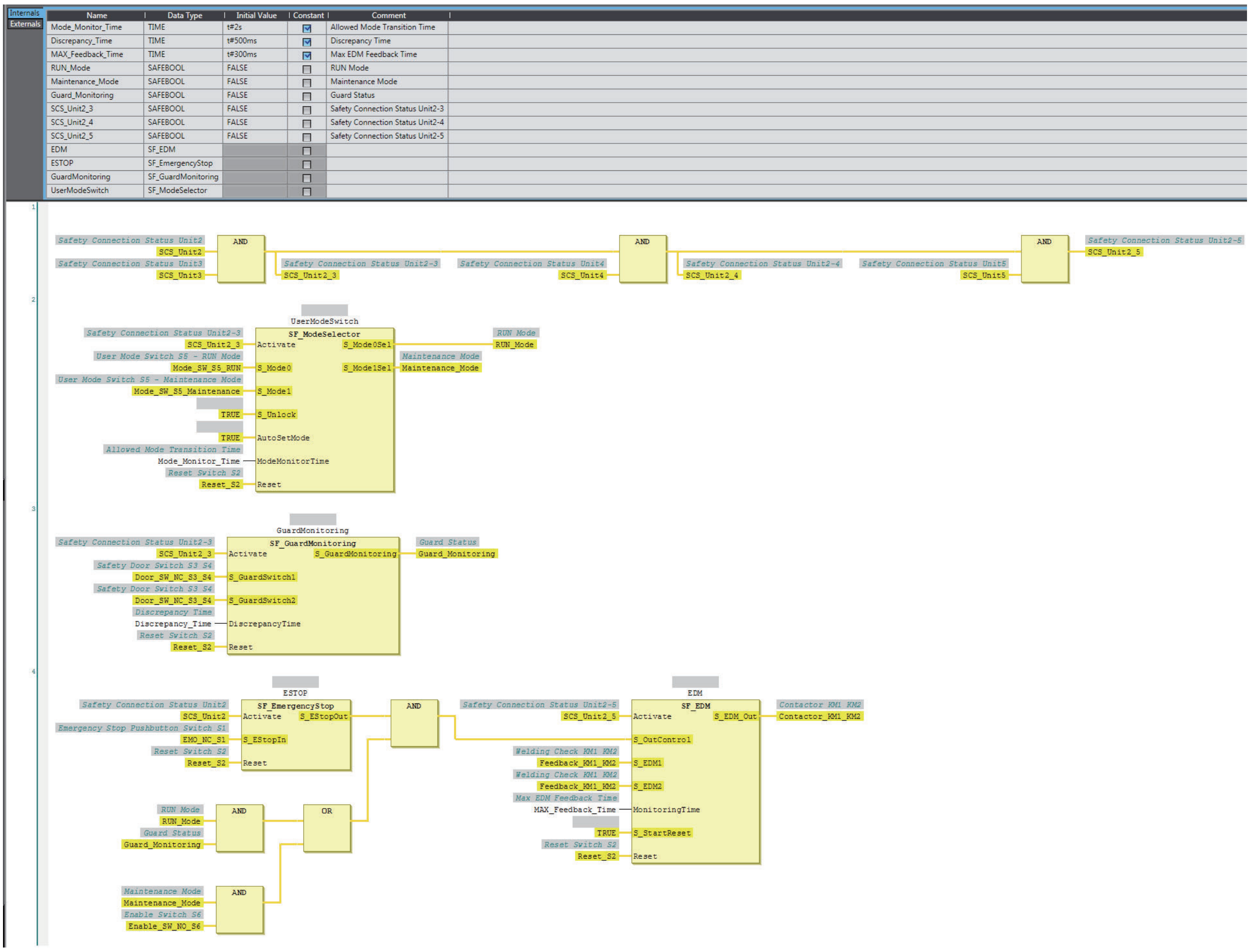
Node1/Unit5 : NX-SOD400 (N5 : Instance3)

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 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	Door_SW_NC_S3_S4	Safety Door Switch S3_S4	Global Variables
	Si01 Logical Value	R	SAFEBOOL			
	Si02 Logical Value	R	SAFEBOOL	Mode_SW_S5_RUN	User Mode Switch S5 - RUN Mode	Global Variables
	Si03 Logical Value	R	SAFEBOOL	Mode_SW_S5_Maintenance	User Mode Switch S5 - Maintenance Mode	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 S6	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	Contactor KM1_KM2	Global Variables
	So01 Output Value	W	SAFEBOOL			
	So02 Output Value	W	SAFEBOOL			
	So03 Output Value	W	SAFEBOOL			

Program



Name	Data Type	Initial Value	Constant	Comment
Mode_Monitor_Time	TIME	t#2s	☑	Allowed Mode Transition Time
Discrepancy_Time	TIME	t#500ms	☑	Discrepancy Time
MAX_Feedback_Time	TIME	t#300ms	☑	Max EDM Feedback Time
RUN_Mode	SAFEBOOL	FALSE	☐	RUN Mode
Maintenance_Mode	SAFEBOOL	FALSE	☐	Maintenance Mode
Guard_Monitoring	SAFEBOOL	FALSE	☐	Guard Status
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
EDM	SF_EDM		☐	
ESTOP	SF_EmergencyStop		☐	
GuardMonitoring	SF_GuardMonitoring		☐	
UserModeSwitch	SF_ModeSelector		☐	

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.

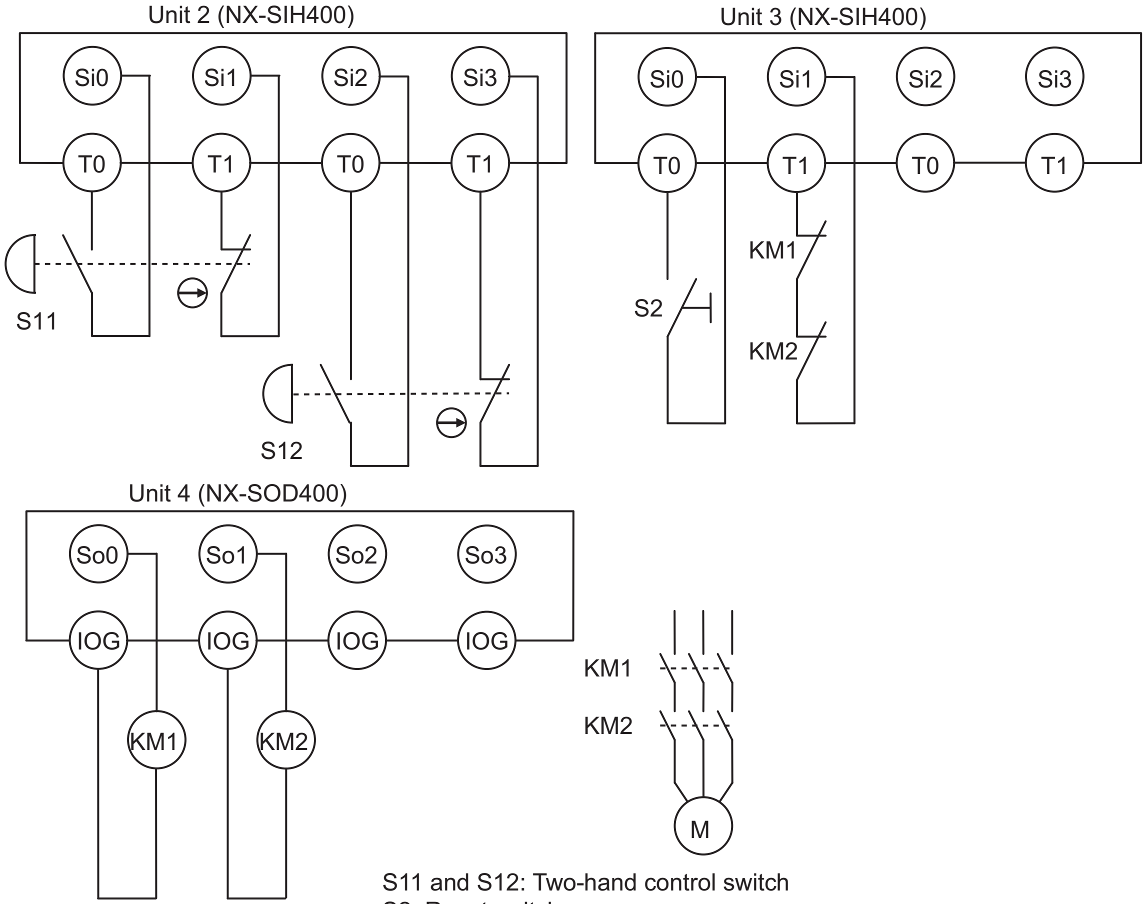
A-3-6 Two-hand Switches

Application Overview

Safety category/PL	Safety device	Stop category	Reset
Equivalent to 4/PLe	• Two-hand control switch	0	Auto

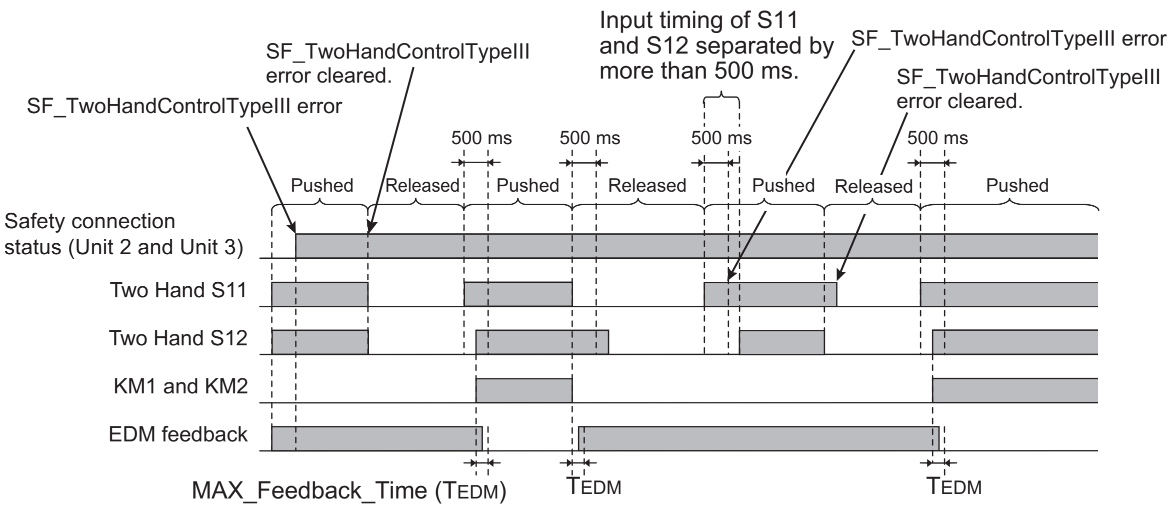
Motor M operates when two-hand control switches S11 and S12 are pressed at the same time.

Wiring



S11 and S12: Two-hand control switch
S2: Reset switch
KM1 and 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	

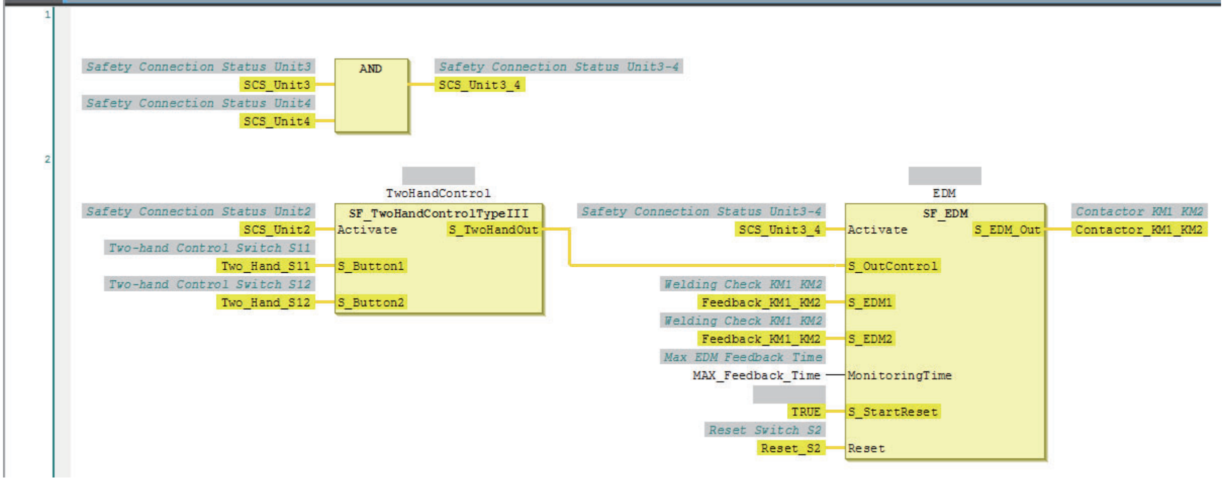
● I/O Map Settings

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

Internals
Externals

Name	Data Type	Initial Value	Constant	Comment
MAX_Feedback_Time	TIME	t#300ms	☑	Max EDM Feedback Time
SCS_Unit3_4	SAFEBOOL	FALSE	☐	Safety Connection Status Unit3-4
EDM	SF_EDM		☐	
TwoHandControl	SF_TwoHandControlTypeIII		☐	



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

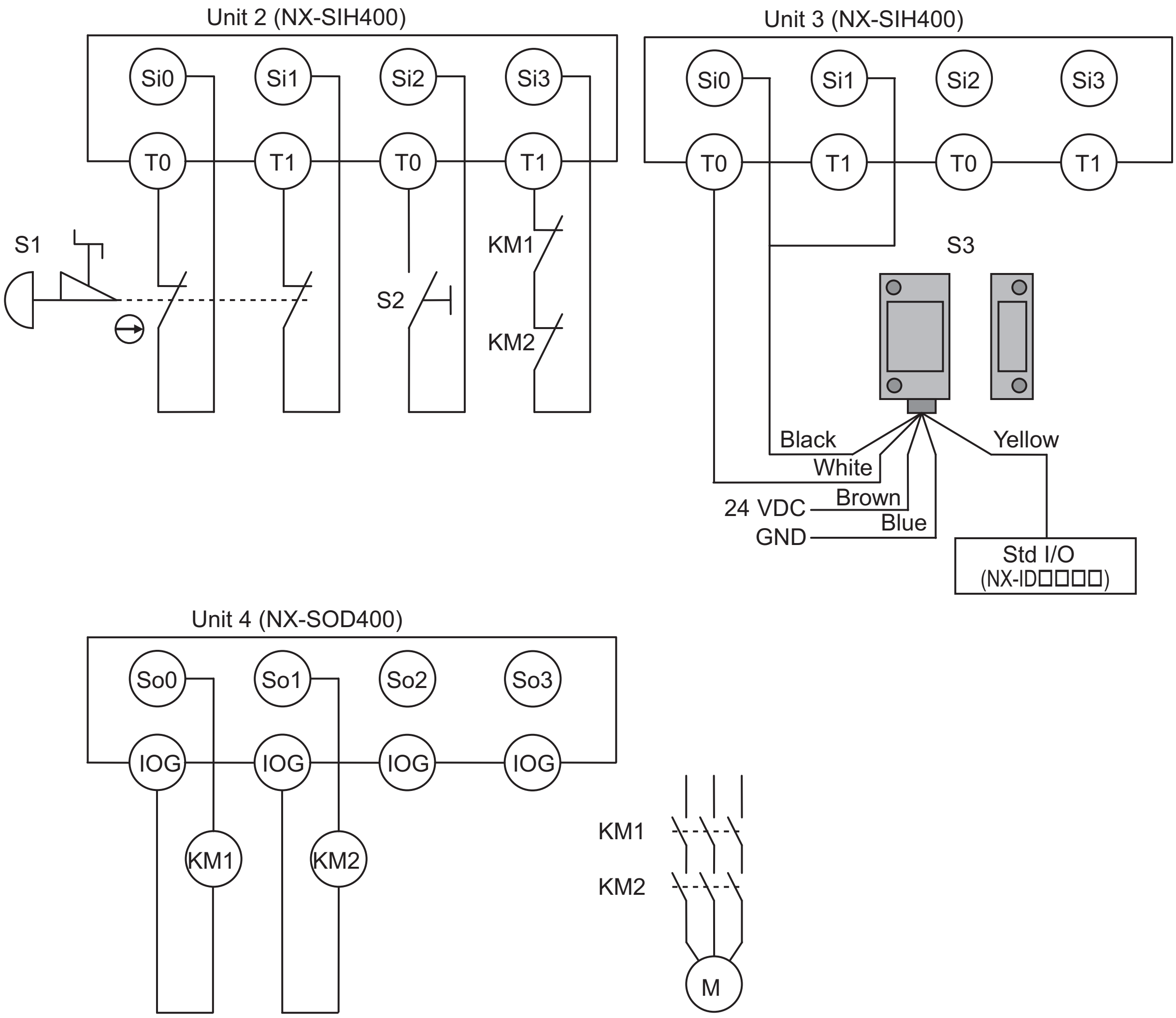
Application Overview

Safety category/PL	Safety device	Stop category	Reset
Equivalent to 3/PLd	• Emergency stop pushbutton • D40A Non-contact Door switch	0	Manual

Motor M stops when emergency stop pushbutton S1 is pressed.

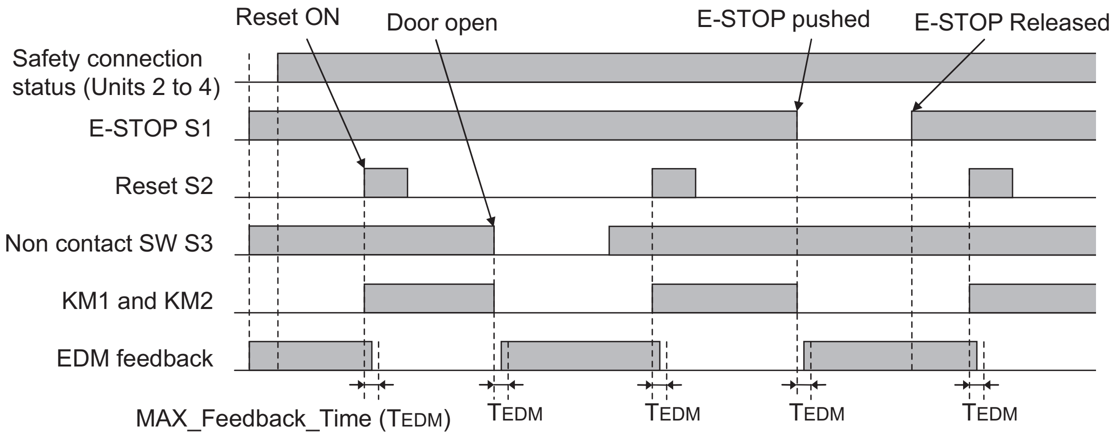
If either of the S3 safety doors (D40A Non-contact Door Switches) is open, motor M will stop.

Wiring



S1: Emergency stop pushbutton
S2: Reset switch
S3: Non contact door switch
KM1 and 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
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	

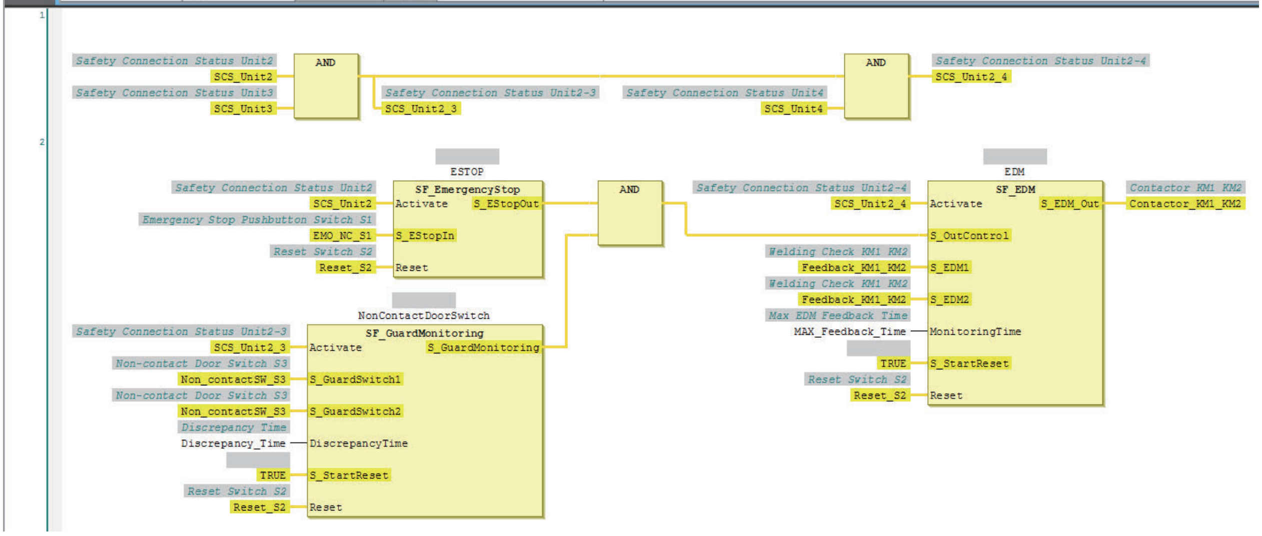
● I/O Map Settings

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

Internals | Externals

Name	Data Type	Initial Value	Constant	Comment
Discrepancy_Time	TIME	t#500ms	☑	Discrepancy Time
MAX_Feedback_Time	TIME	t#300ms	☑	Max EDM Feedback Time
SCS_Unit2_3	SAFEBOOL	FALSE	☐	Safety Connection Status Unit2-3
SCS_Unit2_4	SAFEBOOL	FALSE	☐	Safety Connection Status Unit2-4
ESTOP	SF_EmergencyStop		☐	
NonContactDoorSwitch	SF_GuardMonitoring		☐	
EDM	SF_EDM		☐	



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-3-8 D40Z Non-contact Door Switches

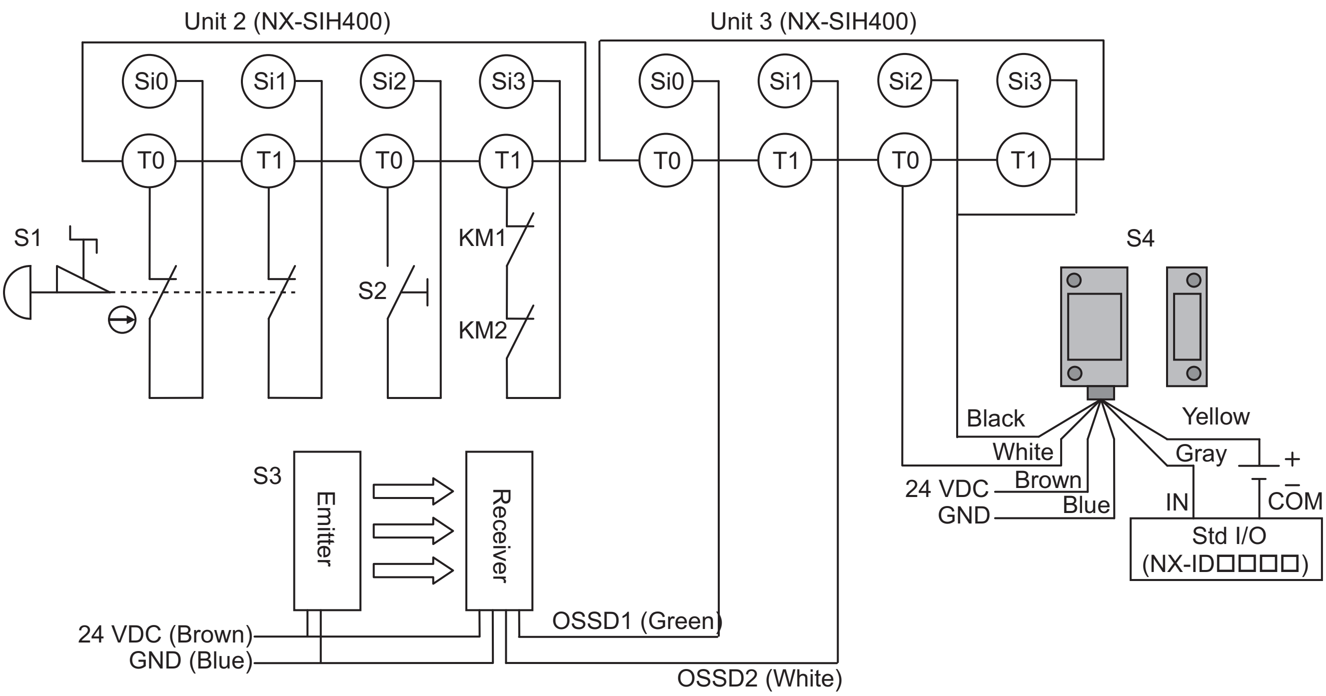
Application Overview

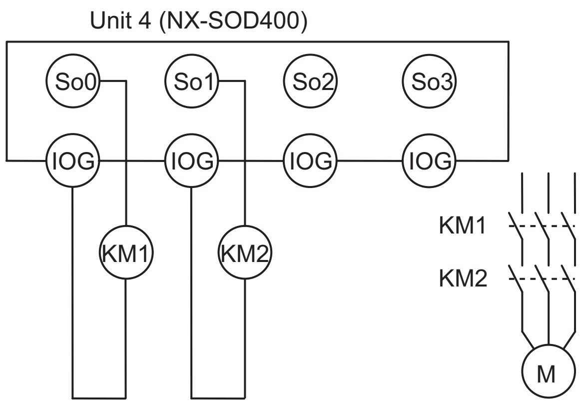
Safety category/PL	Safety device	Stop category	Reset
Equivalent to 4/PLe	• D40Z Non-contact door switch • Safety light curtain • Emergency stop pushbutton	0	Manual

If the light in the S3 safety light curtain is interrupted and the S4 non-contact door switch turns OFF at the same time, the output is turned OFF.

The output also goes OFF when emergency pushbutton S1 is pressed.

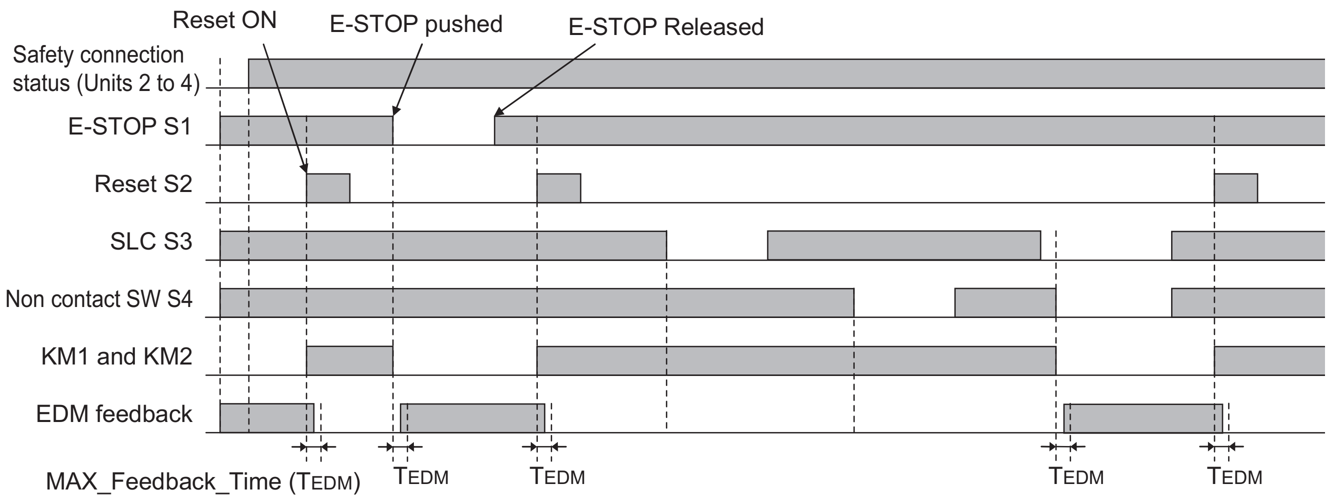
Wiring





S1: Emergency stop pushbutton
S2: Reset switch
S3: Safety light curtain
S4: Non contact door switch
KM1 and 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	

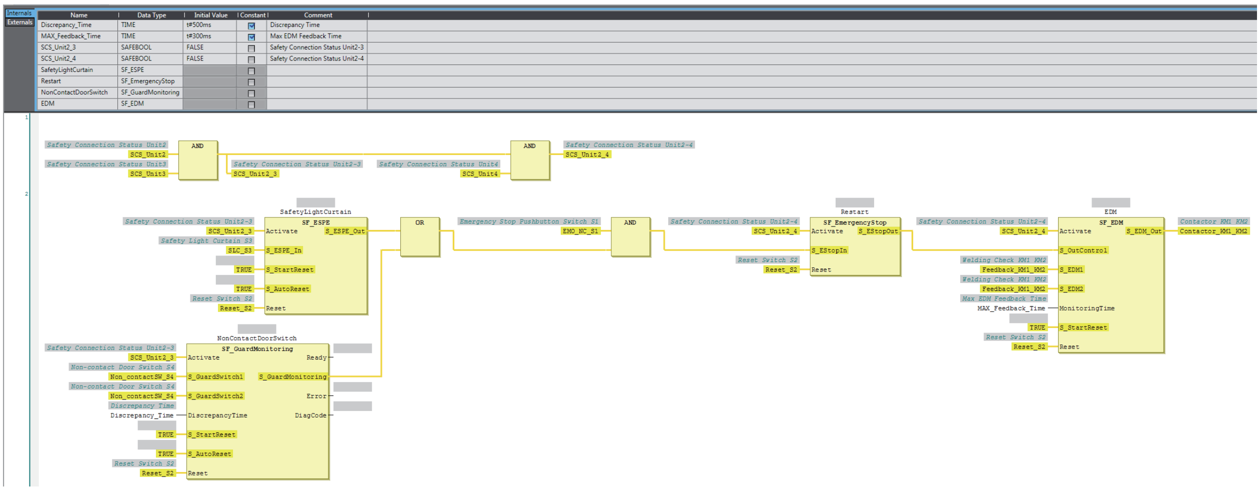
Node1/Unit4 : NX-SOD400 (N4 : Instance2)

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	Non-contact Door Switch 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.

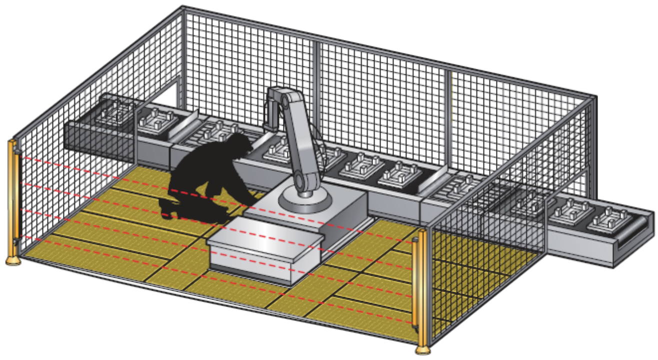
A-3-9 Safety Mats and Safety Light Curtains

Application Overview

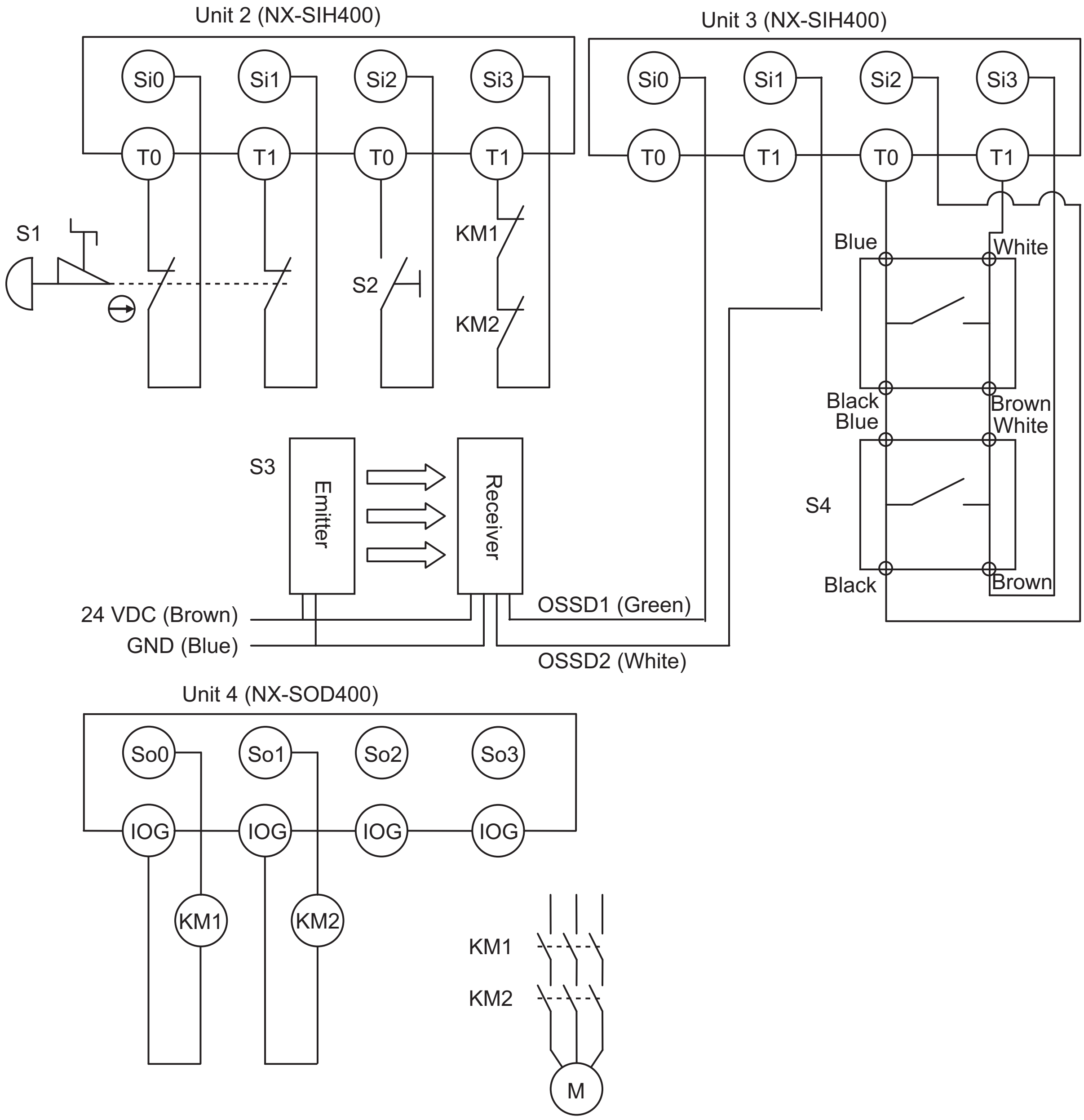
Safety category/PL	Safety device	Stop category	Reset
Equivalent to 3/PLd	• Emergency stop pushbutton • Safety light curtain • Safety mat	0	Manual

Safety light curtain monitors apertural area of safeguarded space and safety mat monitors inside of safeguarded space.

If the light in the S3 safety light curtain is interrupted or the S4 safety mat detects a person or object, motor M will stop.

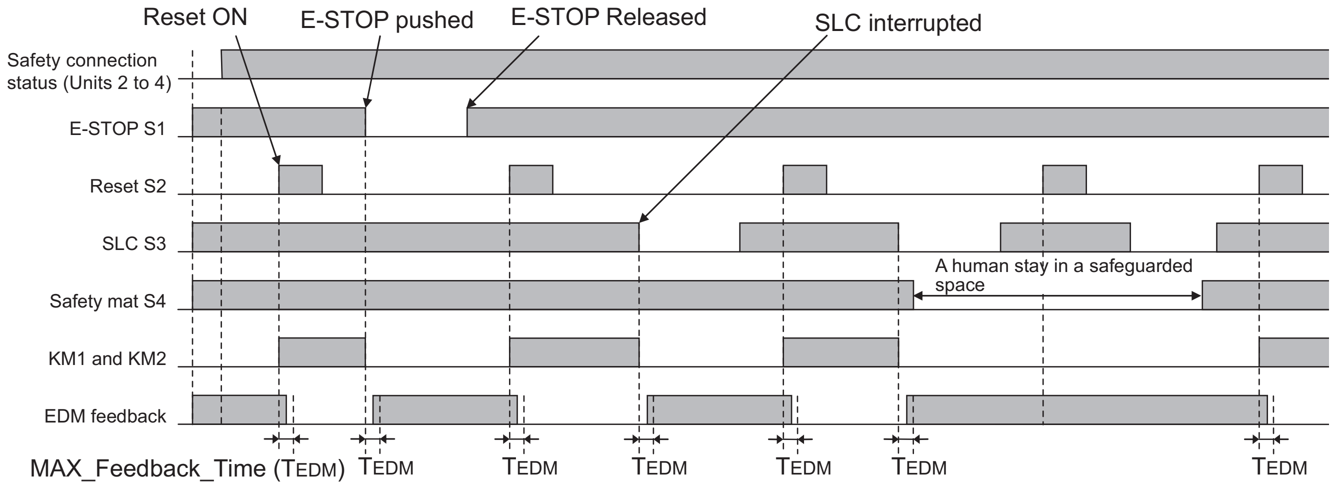


Wiring



S1: Emergency stop pushbutton
S2: Reset switch
S3: Safety light curtain
S4: Safety mat
KM1 and 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	

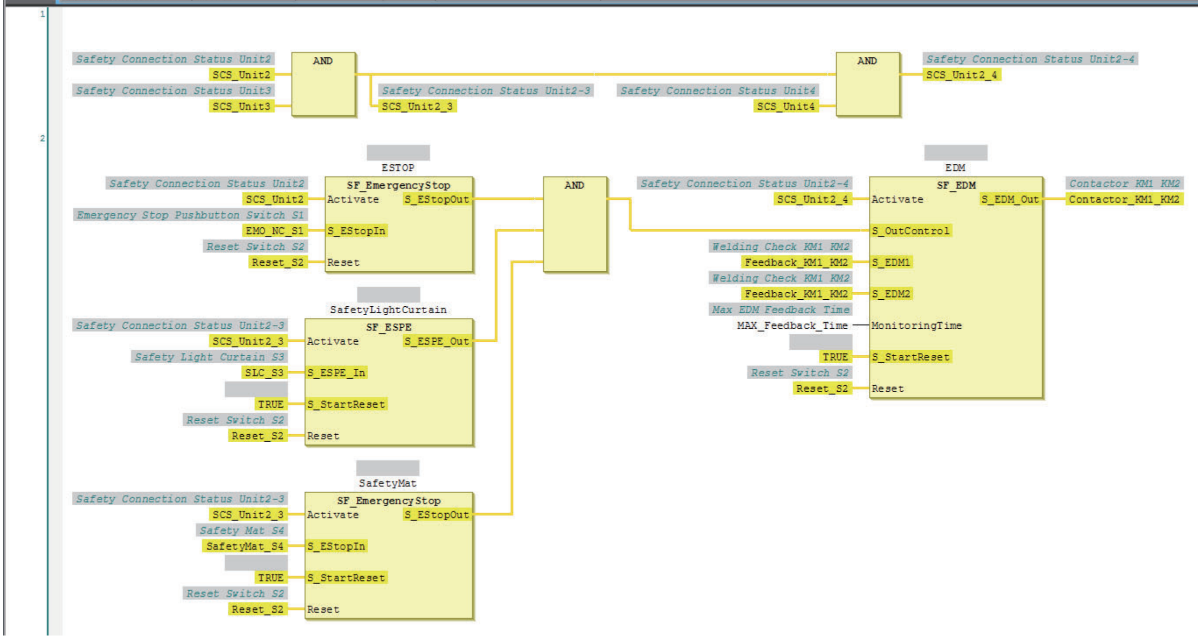
● I/O Map Settings

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

Internals
Externals

Name	Data Type	Initial Value	Constant	Comment
MAX_Feedback_Time	TIME	t#300ms	☑	Max EDM Feedback Time
SCS_Unit2_3	SAFEBOOL	FALSE	☐	Safety Connection Status Unit2-3
SCS_Unit2_4	SAFEBOOL	FALSE	☐	Safety Connection Status Unit2-4
EDM	SF_EDM		☐	
SafetyMat	SF_EmergencyStop		☐	
ESTOP	SF_EmergencyStop		☐	
SafetyLightCurtain	SF_ESPE		☐	



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.

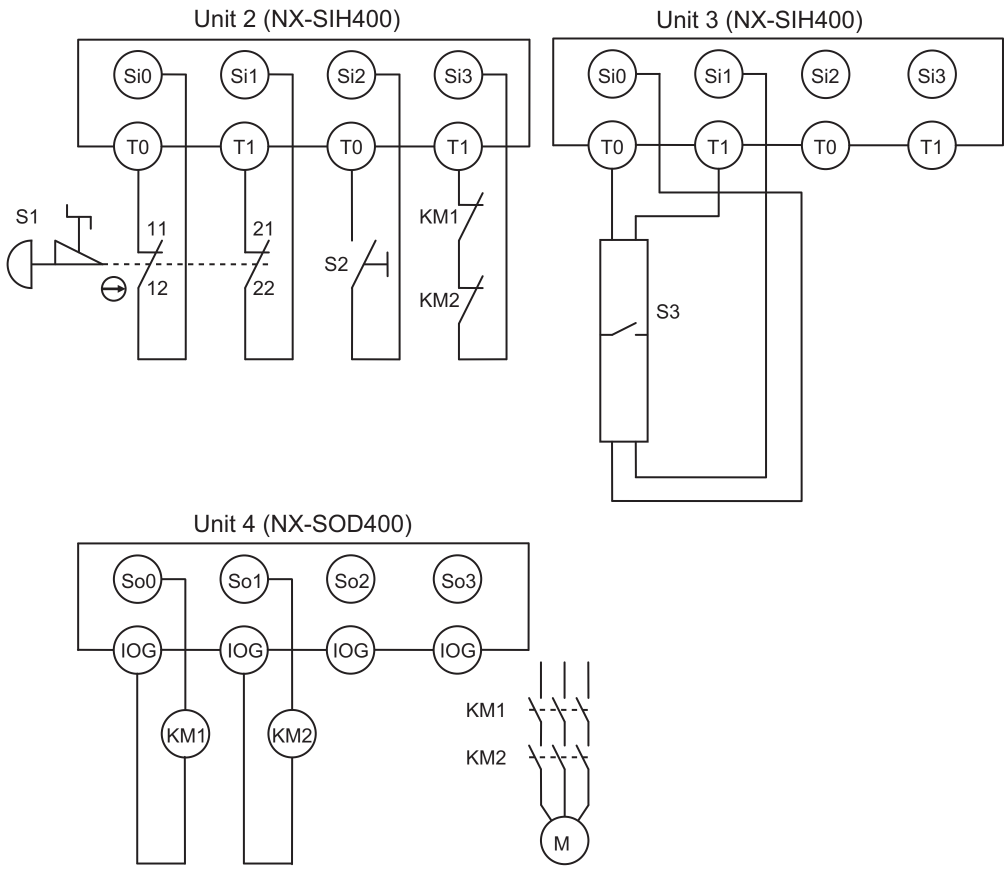
A-3-10 Safety Edges

Application Overview

Safety category/PL	Safety device	Stop category	Reset
Equivalent to 3/PLd	• Emergency stop pushbutton • Safety Edge (2-wire cable on both sides)	0	Manual

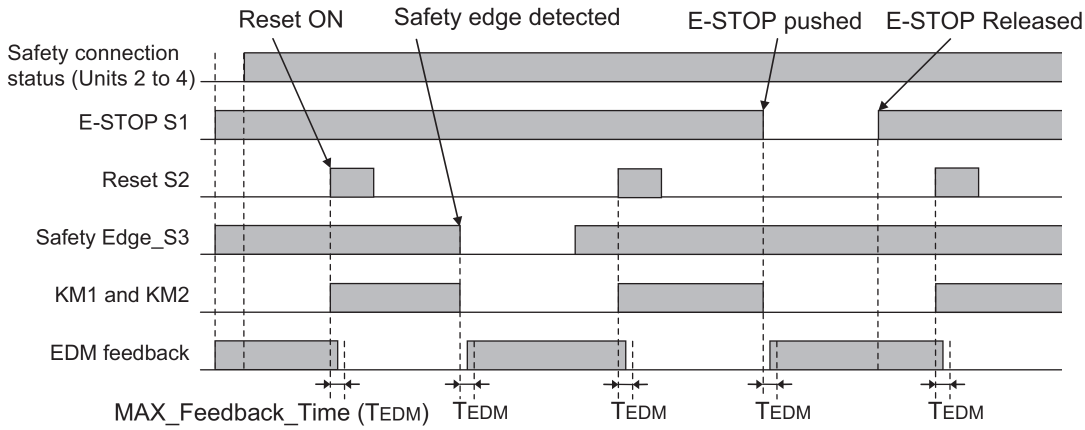
Motor M stops when emergency stop pushbutton S1 is pressed or when edge sensor detect a contact with persons or objects.

Wiring



S1: Emergency stop pushbutton
S2: Reset switch
S3: Safety Edge
KM1 and 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	

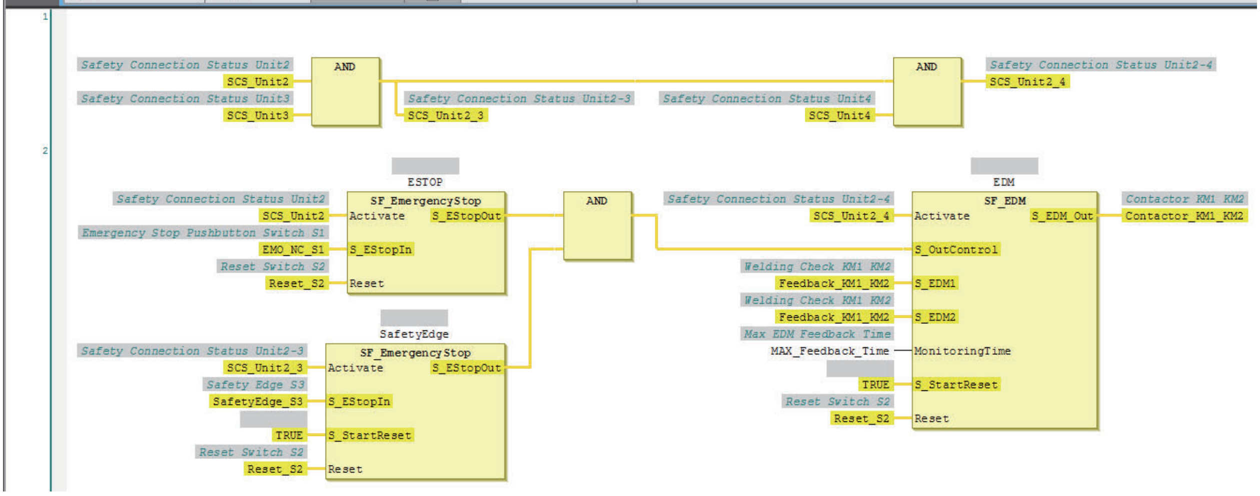
● I/O Map Settings

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-SIH400					
	▼ Safety Inputs and Status					
	Si00 Logical Value	R	SAFEBOOL	SafetyEdge_S3	Safety Edge 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

Internals
Externals

Name	Data Type	Initial Value	Constant	Comment
MAX_Feedback_Time	TIME	t#300ms	☑	Max EDM Feedback Time
SCS_Unit2_3	SAFEBOOL	FALSE	☐	Safety Connection Status Unit2-3
SCS_Unit2_4	SAFEBOOL	FALSE	☐	Safety Connection Status Unit2-4
ESTOP	SF_EmergencyStop		☐	
SafetyEdge	SF_EmergencyStop		☐	
EDM	SF_EDM		☐	



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 Sensors

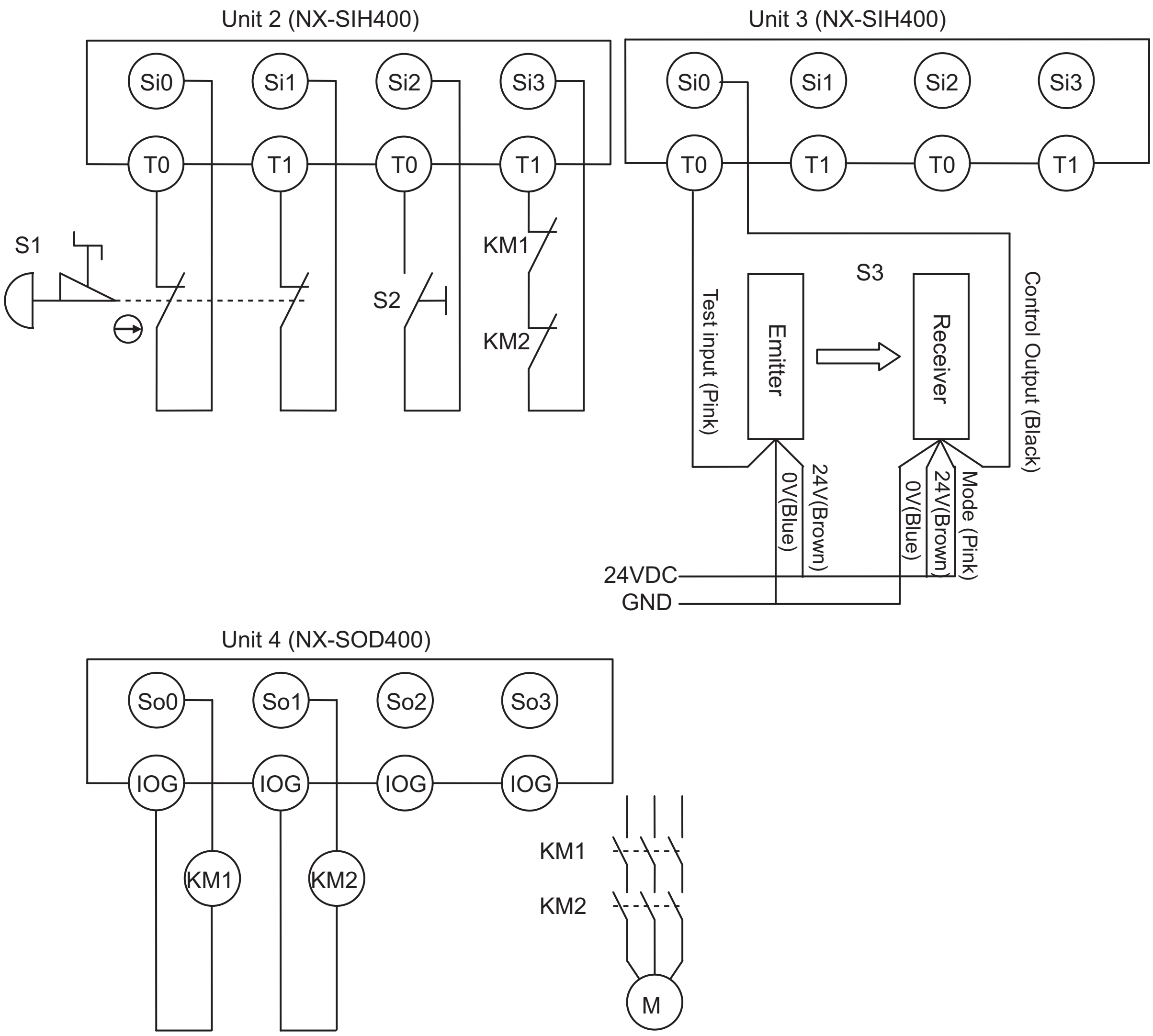
Application Overview

Safety category/PL	Safety device	Stop category	Reset
Equivalent to 2/PLc	• Emergency stop pushbutton • Single beam safety sensor	0	Manual

Motor M stops when emergency stop pushbutton S1 is pressed.

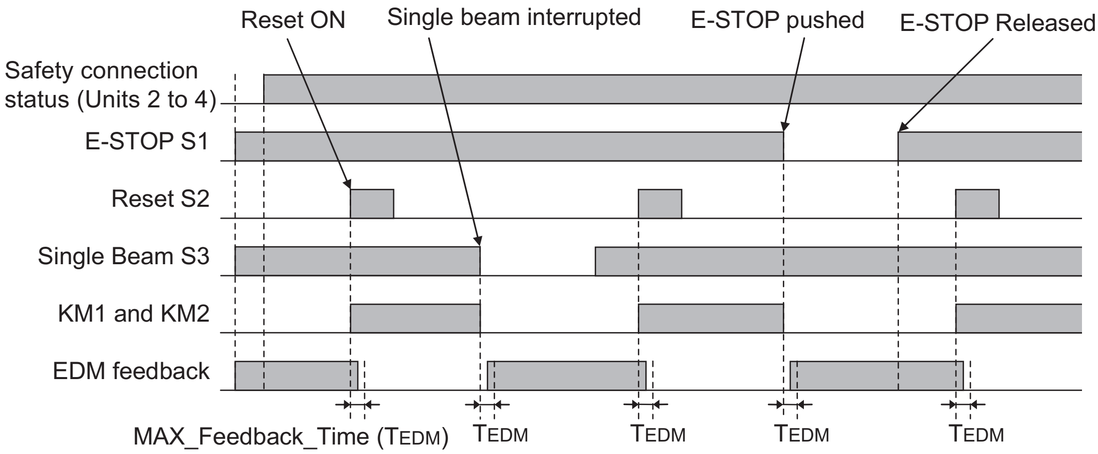
Motor M stops when light is intercepted for the single beam safety sensor.

Wiring



S1: Emergency stop pushbutton
S2: Reset switch
S3: Single beam safety sensor
KM1 and 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	

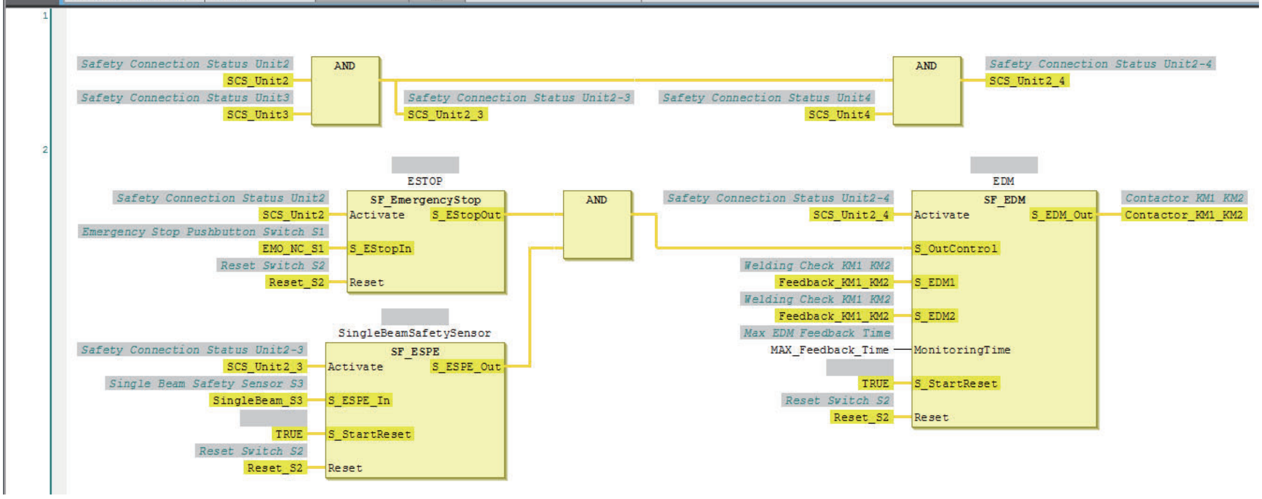
- **I/O Map Settings**

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

Internals / Externals

Name	Data Type	Initial Value	Constant	Comment
MAX_Feedback_Time	TIME	t#300ms	☑	Max EDM Feedback Time
SCS_Unit2_3	SAFEBOOL	FALSE	☐	Safety Connection Status Unit2-3
SCS_Unit2_4	SAFEBOOL	FALSE	☐	Safety Connection Status Unit2-4
EDM	SF_EDM		☐	
ESTOP	SF_EmergencyStop		☐	
SingleBeamSafetySensor	SF_ESPE		☐	



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-4 Change Tracking

What Is Change Tracking?

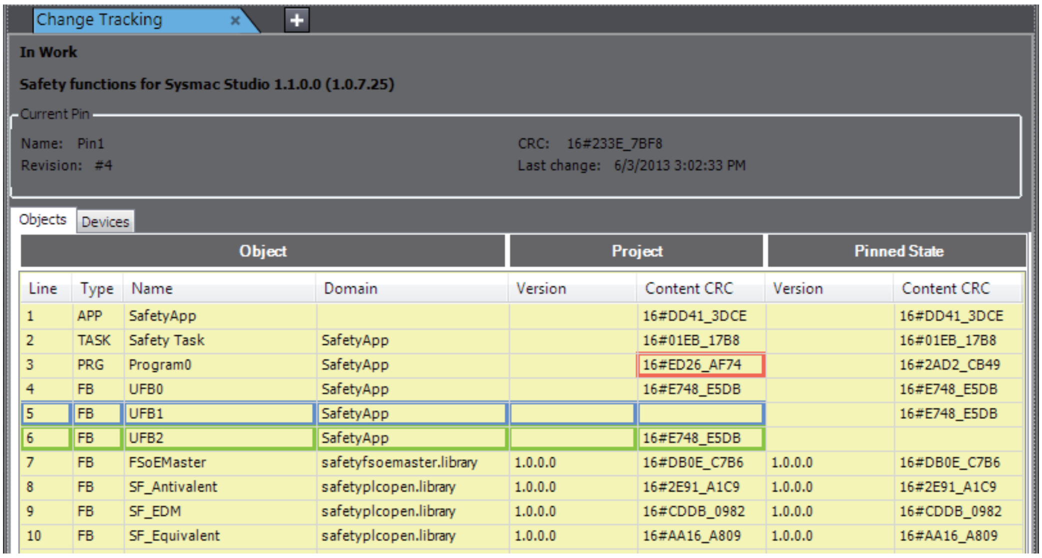
The storage of the safety application data settings at a given point in time is referred to as creating a pin. Change Tracking is used to display and manage changes in the safety application data after the pin is created.

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. • 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.

Pin Operations

This section describes the procedures to create and delete pins, and the jump function.

● Creating Pins

When you click the **Create New Pin** 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 **Create New Pin** 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

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 an EtherCAT Coupler Unit, the status is displayed as an I/O port in the I/O Map of the NJ-series CPU Unit. If you set a device variable for the I/O port, you can monitor the status of the Safety CPU Unit from the NJ-series CPU Unit.

I/O port		Description	Conditions	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. 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. **Conditions** • The safety programs are in RUN status (RUN mode or DEBUG mode (RUN)). • No event with a level of minor fault or higher currently exists for the safety programs. • All FSoE master 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. **Condition** • 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. **Condition** • 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. **Condition** • All FSoE master connections are established.	R	BOOL

A-6 I/O Ports for Safety I/O Units That Are Displayed in the I/O Map of the NJ-series CPU Unit

The I/O ports for Safety I/O Units that are displayed in the I/O Map of the NJ-series CPU Unit are described in this section. 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	Corre-sponding port name
Standard Input 1st Byte	BYTE	R	Standard input 1st byte	---	00 hex	---
Si00 Logical Value	BOOL	R	Safety Input Data 00	Gives the status of safety input terminal Si00. 0: OFF, 1: ON	0	Same
Si01 Logical Value	BOOL	R	Safety Input Data 01	Gives the status of safety input terminal Si01. 0: OFF, 1: ON	0	Same
Si02 Logical Value	BOOL	R	Safety Input Data 02	Gives the status of safety input terminal Si02. 0: OFF, 1: ON	0	Same
Si03 Logical Value	BOOL	R	Safety Input Data 03	Gives the status of safety input terminal Si03. 0: OFF, 1: ON	0	Same
Safety Connection Status	BOOL	R	Safety Con-nection Sta-tus	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 connec-tion/disconnection applications.	0	Same
Safety Input Termi-nal Status	BOOL	R	Safety Input Terminal Status	This flag indicates the status of the safety input terminals. 0: An error has occurred on one of the safety input terminals. 1: All of the safety input terminals are normal (no errors).	0	Same
Unit Normal Status	BOOL	R	Unit Nor-mal Status	This flag indicates the status of the Unit. 0: An error has occurred. 1: Normal (no errors).	0	---

Port	Data type	R/W	Name	Description	Default	Corre-sponding port name
IO Power Supply Error Flag	BOOL	R	I/O Power Supply Error Flag	This flag indicates the status of the I/O power supply voltage. 0: The I/O power supply voltage is normal. 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	Safety input terminal status 00	Gives the status of safety input terminal 00. 0: Error 1: No error	0	---
Si01 Status	BOOL	R	Safety input terminal status 01	Gives the status of safety input terminal 01. 0: Error 1: No error	0	---
Si02 Status	BOOL	R	Safety input terminal status 02	Gives the status of safety input terminal 02. 0: Error 1: No error	0	---
Si03 Status	BOOL	R	Safety input terminal status 03	Gives the status of safety input terminal 03. 0: Error 1: No error	0	---

A-6-2 NX-SID800 Safety Input Unit

Port	Data type	R/W	Name	Description	Default	Corre-sponding port name
Standard Input 1st Byte	BYTE	R	Standard input 1st byte	---	00 hex	---
Si00 Logical Value	BOOL	R	Safety Input Data 00	Gives the status of safety input terminal Si00. 0: OFF, 1: ON	0	Same
Si01 Logical Value	BOOL	R	Safety Input Data 01	Gives the status of safety input terminal Si01. 0: OFF, 1: ON	0	Same
Si02 Logical Value	BOOL	R	Safety Input Data 02	Gives the status of safety input terminal Si02. 0: OFF, 1: ON	0	Same
Si03 Logical Value	BOOL	R	Safety Input Data 03	Gives the status of safety input terminal Si03. 0: OFF, 1: ON	0	Same
Si04 Logical Value	BOOL	R	Safety Input Data 04	Gives the status of safety input terminal Si04. 0: OFF, 1: ON	0	Same

Port	Data type	R/W	Name	Description	Default	Corre-sponding port name
Si05 Logical Value	BOOL	R	Safety Input Data 05	Gives the status of safety input terminal Si05. 0: OFF, 1: ON	0	Same
Si06 Logical Value	BOOL	R	Safety Input Data 06	Gives the status of safety input terminal Si06. 0: OFF, 1: ON	0	Same
Si07 Logical Value	BOOL	R	Safety Input Data 07	Gives the status of safety input terminal Si07. 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. 0: An error has occurred on one of the safety input terminals. 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. 0: An error has occurred. 1: Normal (no errors)	0	---
IO Power Supply Error Flag	BOOL	R	I/O Power Supply Error Flag	This flag indicates the status of the I/O power supply voltage. 0: The I/O power supply voltage is normal. 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	Safety input terminal status 00	Gives the status of safety input terminal 00. 0: Error 1: No error	0	---
Si01 Status	BOOL	R	Safety input terminal status 01	Gives the status of safety input terminal 01. 0: Error 1: No error	0	---
Si02 Status	BOOL	R	Safety input terminal status 02	Gives the status of safety input terminal 02. 0: Error 1: No error	0	---
Si03 Status	BOOL	R	Safety input terminal status 03	Gives the status of safety input terminal 03. 0: Error 1: No error	0	---

Port	Data type	R/W	Name	Description	Default	Corre-sponding port name
Si04 Status	BOOL	R	Safety input terminal status 04	Gives the status of safety input terminal 04. 0: Error 1: No error	0	---
Si05 Status	BOOL	R	Safety input terminal status 05	Gives the status of safety input terminal 05. 0: Error 1: No error	0	---
Si06 Status	BOOL	R	Safety input terminal status 06	Gives the status of safety input terminal 06. 0: Error 1: No error	0	---
Si07 Status	BOOL	R	Safety input terminal status 07	Gives the status of safety input terminal 07. 0: Error 1: No error	0	---

A-6-3 NX-SOH200 Safety Output Unit

Port	Data type	R/W	Name	Description	Default	Corre-sponding port name
Standard Input 1st Byte	BYTE	R	Standard input 1st byte	---	00 hex	---
So00 Monitor Value	BOOL	R	Safety Output Monitor 00	Gives the status of safety output terminal So00. 0: OFF, 1: ON	0	So00 Output Value
So01 Monitor Value	BOOL	R	Safety Output Monitor 01	Gives the status of safety output terminal So01. 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. 0: An error has occurred on one of the safety output terminals. 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. 0: An error has occurred. 1: Normal (no errors)	0	---

Port	Data type	R/W	Name	Description	Default	Corre-sponding port name
IO Power Supply Error Flag	BOOL	R	I/O Power Supply Error Flag	This flag indicates the status of the I/O power supply voltage. 0: The I/O power supply voltage is normal. 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	Safety out-put termi-nal status 00	Gives the status of safety output terminal 00. 0: Error 1: No error	0	---
So01 Status	BOOL	R	Safety out-put termi-nal status 01	Gives the status of safety output terminal 01. 0: Error 1: No error	0	---

A-6-4 NX-SOD400 Safety Output Unit

Port	Data type	R/W	Name	Description	Default	Corre-sponding port name
Standard Input 1st Byte	BYTE	R	Standard input 1st byte	---	00 hex	---
So00 Monitor Value	BOOL	R	Safety Out-put Monitor 00	Gives the status of safety output terminal So00. 0: OFF, 1: ON	0	So00 Out-put Value
So01 Monitor Value	BOOL	R	Safety Out-put Monitor 01	Gives the status of safety output terminal So01. 0: OFF, 1: ON	0	So01 Out-put Value
So02 Monitor Value	BOOL	R	Safety Out-put Monitor 02	Gives the status of safety output terminal So02. 0: OFF, 1: ON	0	So02 Out-put Value
So03 Monitor Value	BOOL	R	Safety Out-put Monitor 03	Gives the status of safety output terminal So03. 0: OFF, 1: ON	0	So03 Out-put Value
Safety Connection Status	BOOL	R	Safety Con-nection Sta-tus	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 connec-tion/disconnection applications.	0	Same
Safety Output Termi-nal Status	BOOL	R	Safety Out-put Termi-nal Status	This flag indicates the status of the safety output terminals. 0: An error has occurred on one of the safety output terminals. 1: All of the safety output terminals are normal (no errors).	0	Same

Port	Data type	R/W	Name	Description	Default	Corre-sponding port name
Unit Normal Status	BOOL	R	Unit Normal Status	This flag indicates the status of the Unit. 0: An error has occurred. 1: Normal (no errors)	0	---
IO Power Supply Error Flag	BOOL	R	I/O Power Supply Error Flag	This flag indicates the status of the I/O power supply voltage. 0: The I/O power supply voltage is normal. 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	Safety output terminal status 00	Gives the status of safety output terminal 00. 0: Error 1: No error	0	---
So01 Status	BOOL	R	Safety output terminal status 01	Gives the status of safety output terminal 01. 0: Error 1: No error	0	---
So02 Status	BOOL	R	Safety output terminal status 02	Gives the status of safety output terminal 02. 0: Error 1: No error	0	---
So03 Status	BOOL	R	Safety output terminal status 03	Gives the status of safety output terminal 03. 0: Error 1: No error	0	---

A-7 Calculating I/O Sizes for Slave Terminals

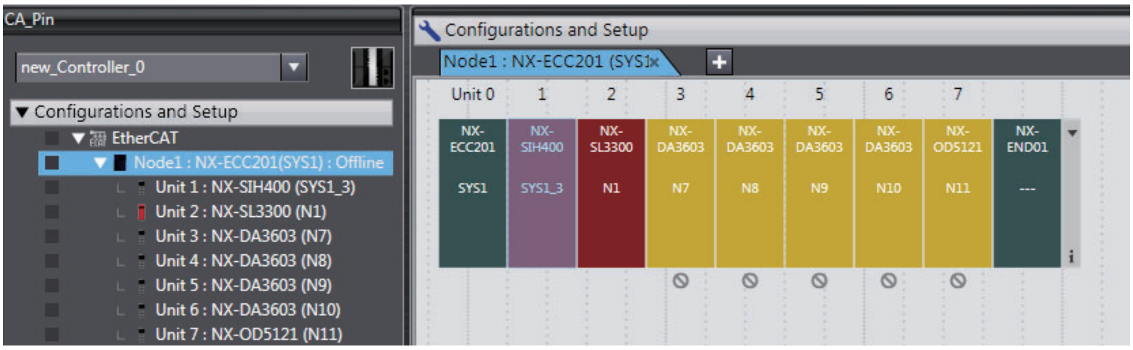
This section describes how to check the I/O sizes of Slave Terminals that have Safety Control Units.

The I/O size of a Safety CPU Unit is determined by the I/O communications with Safety I/O Units and the variables in the safety programs.

Use the following procedure to check the I/O sizes of Slave Terminals that have Safety Control Units.

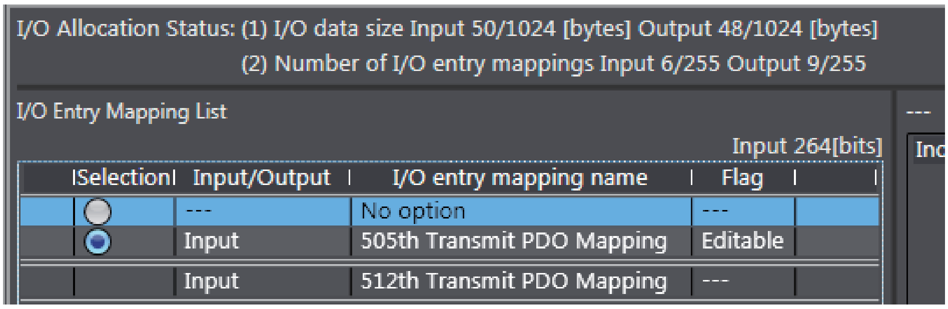
1 Right-click the Slave Terminal under **Configurations and Setup - EtherCAT** in the Multiview Explorer and select ***Edit*** from the menu.

The Edit Slave Terminal Configuration Tab Page appears in the Configurations and Setup Layer.



2 Select the EtherCAT Coupler Unit and then click the **Edit I/O Allocation Settings** Button in the Parameter Settings Area on the right side of the tab page.

The Edit I/O Allocation Settings Tab Page is displayed.



The data sizes for inputs and outputs for the entire Slave Terminal are displayed in the *I/O Allocation Status* Column.

A-8 Version Information

This section describes the combinations that can be used of the unit versions of the Safety Control Units, NJ-series CPU Units, and NX-series EtherCAT Coupler Unit, and the version of the Sysmac Studio.

● Applicable Units and Sysmac Studio

The following table gives the model numbers and unit versions of the NJ-series CPU Units and the NX-series EtherCAT Coupler Unit and the model number and version of the Sysmac Studio that can be used together.

Name	Applicable models and versions
NJ-series CPU Unit	NJ501-□□□□ or NJ301-□□□□ (unit version 1.06 or later)
NX-series EtherCAT Coupler Unit	NX-ECC201 (unit version 1.1 or later)
Sysmac Studio	SYSMAC SE2□□□□ (version 1.07 or higher)

For information on the NJ-series CPU Units, NX-series EtherCAT Coupler Unit, other NX Units, and the Sysmac Studio, refer to applicable Unit or Sysmac Studio manuals.

I

Index

Index

D

E

F

G

I

L

N

P

R

S

T

OMRON Corporation **Industrial Automation Company**
Tokyo, JAPAN

Contact: www.ia.omron.com

Regional Headquarters
OMRON EUROPE B.V.
Wegalaan 67-69-2132 JD Hoofddorp
The Netherlands
Tel: (31)2356-81-300/Fax: (31)2356-81-388

OMRON ASIA PACIFIC PTE. LTD.
No. 438A Alexandra Road # 05-05/08 (Lobby 2),
Alexandra Technopark,
Singapore 119967
Tel: (65) 6835-3011/Fax: (65) 6835-2711

OMRON SCIENTIFIC TECHNOLOGIES INC.
6550 Dumbarton Circle,
Fremont, CA 94555-3605 U.S.A.
Tel: (1) 510-608-3400/Fax: (1) 510-744-1442

OMRON (CHINA) CO., LTD.
Room 2211, Bank of China Tower,
200 Yin Cheng Zhong Road,
PuDong New Area, Shanghai, 200120, China
Tel: (86) 21-5037-2222/Fax: (86) 21-5037-2200

Authorized Distributor:

Cat. No. Z930-E1-01

0613