

Machine Automation Controller

NX-series

NX1P2 CPU Unit

Built-in I/O and Option Board User's Manual

NX1P2-11□□□□

NX1P2-11□□□□1

NX1P2-10□□□□

NX1P2-10□□□□1

NX1P2-90□□□□

NX1P2-90□□□□1

CPU Unit



NOTE

All rights reserved. No part of this publication may be reproduced, stored in a retrieval system, or transmitted, in any form, or by any means, mechanical, electronic, photocopying, recording, or otherwise, without the prior written permission of OMRON.

No patent liability is assumed with respect to the use of the information contained herein. Moreover, because OMRON is constantly striving to improve its high-quality products, the information contained in this manual is subject to change without notice. Every precaution has been taken in the preparation of this manual. Nevertheless, OMRON assumes no responsibility for errors or omissions. Neither is any liability assumed for damages resulting from the use of the information contained in this publication.

Trademarks

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

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

Copyrights

Microsoft product screen shots reprinted with permission from Microsoft Corporation.

Introduction

Thank you for purchasing an NX-series NX1P2 CPU Unit.

This manual contains information that is necessary to use the NX-series NX1P2 CPU Unit. Please read this manual and make sure you understand the functionality and performance of the NX-series NX1P2 CPU Unit before you attempt to use it in a control system.

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

Intended Audience

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

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

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

Applicable Products

This manual covers the following products.

- NX-series NX1P2 CPU Units
 - NX1P2-11□□□□
 - NX1P2-11□□□□1
 - NX1P2-10□□□□
 - NX1P2-10□□□□1
 - NX1P2-90□□□□
 - NX1P2-90□□□□1

Part of the specifications and restrictions for the CPU Units are given in other manuals. Refer to *Relevant Manuals* on page 7 and *Related Manuals* on page 19.

CONTENTS

Introduction	1
Intended Audience	1
Applicable Products	1
CONTENTS	2
Relevant Manuals	7
Manual Structure	8
Page Structure	8
Special Information	9
Precaution on Terminology	9
Terms and Conditions Agreement	10
Warranty, Limitations of Liability	10
Application Considerations	11
Disclaimers	11
Safety Precautions	12
Precautions for Safe Use	13
Precautions for Correct Use	14
Regulations and Standards	15
Versions	16
Checking Versions	16
Unit Versions of CPU Units and Sysmac Studio Versions	18
Related Manuals	19
Terminology	23
Revision History	28
Sections in this Manual	29

Section 1 Introduction to NX1P2 CPU Units

1-1 Function Specifications for NX1P2 CPU Units	1-2
1-2 Overall Operating Procedure	1-7
1-2-1 Overall Operating Procedure	1-7
1-2-2 Procedure Details	1-8

Section 2 Built-in I/O

2-1 Built-in I/O Terminal Allocation	2-2
2-1-1 Terminal Arrangement	2-2
2-2 I/O Data Specifications	2-5
2-2-1 NX1P2-□□24DT/-□□24DT1	2-5
2-2-2 NX1P2-□□40DT/-□□40DT1	2-6

2-3	Built-in I/O Functions	2-7
2-4	Settings	2-8
2-4-1	Built-in I/O Settings	2-8
2-4-2	I/O Map	2-9
2-5	Functions	2-10
2-5-1	Input Filter	2-10
2-5-2	Output Load Rejection Setting	2-12
2-6	I/O Refreshing	2-13
2-6-1	I/O Refresh Timing of Built-in I/O	2-13
2-6-2	I/O Response Time of Built-in I/O	2-15

Section 3 Option Boards

3-1	Option Board Types	3-2
3-1-1	Serial Communications Option Boards	3-3
3-1-2	Analog I/O Option Boards	3-3
3-2	Using Option Boards	3-4
3-2-1	Settings	3-4
3-2-2	System-defined Variables	3-9
3-2-3	Device Variables	3-10
3-2-4	Assigning Device Variables to Option Boards	3-11
3-2-5	Instructions Used for Option Boards	3-13
3-2-6	How Option Boards Operate in Case of an Error	3-14

Section 4 Serial Communications

4-1	Serial Communications Types and Overview	4-2
4-2	Programless Communications with NB-series Programmable Terminals	4-4
4-2-1	Overview	4-4
4-2-2	Procedure	4-4
4-2-3	Settings	4-6
4-2-4	Programming	4-8
4-2-5	Connection Examples	4-9
4-3	Programless Communications with E5□C Digital Temperature Controllers	4-10
4-3-1	Overview	4-10
4-3-2	Procedure	4-12
4-3-3	Settings	4-14
4-3-4	Programming	4-17
4-3-5	Connection Examples	4-17
4-4	Connection with Modbus-RTU Slaves	4-18
4-4-1	Overview	4-18
4-4-2	Procedure	4-19
4-4-3	Settings	4-21
4-4-4	Programming	4-22
4-4-5	Connection Examples	4-24
4-5	Connection with General-purpose Serial Communications Devices	4-25
4-5-1	Overview	4-25
4-5-2	Procedure	4-26
4-5-3	Settings	4-28
4-5-4	Programming	4-29

Section 5 Analog I/O

5-1	Specifications	5-2
5-1-1	Analog I/O Option Boards	5-2

5-1-2	Part Names and Functions.....	5-2
5-1-3	Terminal Arrangement.....	5-3
5-1-4	Input Range and Output Range	5-3
5-2	Procedure.....	5-5
5-3	Settings	5-6
5-3-1	Option Board Settings.....	5-6
5-3-2	Device Variables	5-7
5-4	Programming	5-8
5-4-1	I/O Data.....	5-8
5-4-2	Option Board Status.....	5-9
5-4-3	Special Instructions for Analog I/O Option Boards	5-9
5-4-4	Precautions on Supported Functions	5-10
5-5	Wiring	5-11
5-6	I/O Refreshing	5-12
5-6-1	I/O Refresh Operation	5-12
5-6-2	Response Time	5-13

Section 6 Introduction of Motion Control Functions

6-1	Single-axis Position Control.....	6-3
6-1-1	Outline of Operation	6-3
6-1-2	Absolute Positioning.....	6-4
6-1-3	Relative Positioning.....	6-4
6-1-4	Interrupt Feeding.....	6-5
6-1-5	Cyclic Synchronous Positioning	6-6
6-1-6	Stopping	6-7
6-1-7	Override Factors	6-13
6-2	Single-axis Synchronized Control	6-14
6-2-1	Overview of Synchronized Control.....	6-14
6-2-2	Gear Operation	6-14
6-2-3	Positioning Gear Operation.....	6-15
6-2-4	Cam Operation.....	6-16
6-2-5	Cam Tables	6-17
6-2-6	Synchronous Positioning.....	6-25
6-2-7	Combining Axes	6-26
6-2-8	Master Axis Phase Shift	6-27
6-2-9	Slave Axis Position Compensation.....	6-27
6-2-10	Achieving Synchronized Control in Multi-motion.....	6-28
6-3	Single-axis Velocity Control	6-30
6-3-1	Velocity Control	6-30
6-3-2	Cyclic Synchronous Velocity Control.....	6-31
6-4	Single-axis Torque Control.....	6-32
6-5	Common Functions for Single-axis Control	6-33
6-5-1	Positions.....	6-33
6-5-2	Velocity.....	6-35
6-5-3	Acceleration and Deceleration	6-36
6-5-4	Jerk	6-38
6-5-5	Specifying the Operation Direction.....	6-39
6-5-6	Re-executing Motion Control Instructions	6-43
6-5-7	Multi-execution of Motion Control Instructions (Buffer Mode)	6-48
6-6	Multi-axes Coordinated Control.....	6-54
6-6-1	Outline of Operation	6-54
6-6-2	Linear Interpolation	6-57
6-6-3	Circular Interpolation.....	6-58
6-6-4	Axes Group Cyclic Synchronous Positioning	6-58
6-6-5	Stopping Under Multi-axes Coordinated Control.....	6-59
6-6-6	Overrides for Multi-axes Coordinated Control.....	6-61

6-7	Common Functions for Multi-axes Coordinated Control	6-62
6-7-1	Velocity Under Multi-axes Coordinated Control	6-62
6-7-2	Acceleration and Deceleration Under Multi-axes Coordinated Control	6-63
6-7-3	Jerk for Multi-axes Coordinated Control	6-64
6-7-4	Re-executing Motion Control Instructions for Multi-axes Coordinated Control	6-65
6-7-5	Multi-execution (Buffer Mode) of Motion Control Instructions for Multi-axes Coordinated Control	6-66
6-8	Other Functions	6-74
6-8-1	Changing the Current Position	6-74
6-8-2	Torque Limit	6-75
6-8-3	Latching	6-75
6-8-4	Zone Monitoring	6-76
6-8-5	Software Limits	6-77
6-8-6	Following Error Monitoring	6-78
6-8-7	Following Error Counter Reset	6-79
6-8-8	Axis Following Error Monitoring	6-80
6-8-9	In-position Check	6-80
6-8-10	Changing Axis Use	6-82
6-8-11	Enabling Digital Cam Switch	6-83
6-8-12	Displaying 3D Motion Monitor for User Coordinate System	6-84

Section 7 Introduction of EtherNet/IP Communications Functions

7-1	Communications Services	7-2
7-1-1	CIP (Common Industrial Protocol) Communications Services	7-2
7-1-2	BOOTP Client	7-4
7-1-3	FTP Server	7-4
7-1-4	FTP Client	7-5
7-1-5	Automatic Clock Adjustment	7-5
7-1-6	Socket Service	7-6
7-1-7	Specifying Host Names	7-7
7-1-8	SNMP Agent	7-7

Section 8 Introduction of EtherCAT Communications Functions

8-1	Overview of Communications	8-2
8-1-1	Process Data Communications and SDO Communications	8-2
8-1-2	Other Functions	8-3

Section 9 Troubleshooting

9-1	Overview of Troubleshooting	9-2
9-2	Option Board Errors	9-3
9-2-1	Checking for Errors and Troubleshooting with the ERR Indicator on Option Boards	9-3

Appendices

A-1	Version Information	A-2
------------	----------------------------------	------------

Index

Relevant Manuals

The following table provides the relevant manuals for the NX-series NX1P2 CPU Units. Read all of the manuals that are relevant to your system configuration and application before you use the NX-series NX1P2 CPU Unit.

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

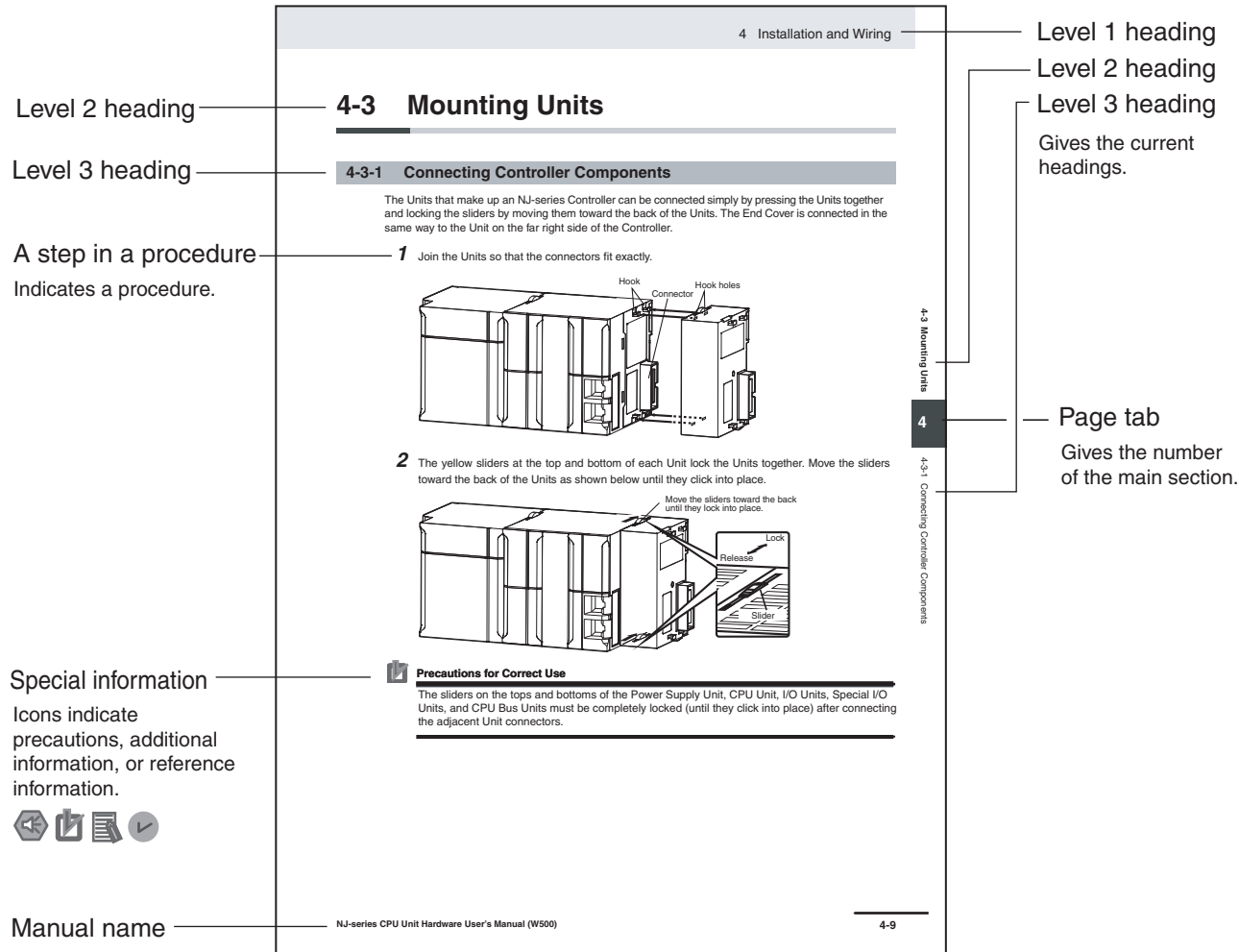
Purpose of use	Manual							
	Basic information							
	NX-series NX1P2 CPU Unit Hardware User's Manual	NJ/NX-series CPU Unit Software User's Manual	NX-series NX1P2 CPU Unit Built-in I/O and Option Board User's Manual	NJ/NX-series Instructions Reference Manual	NJ/NX-series CPU Unit Motion Control User's Manual	NJ/NX-series Motion Control Instructions Reference Manual	NJ/NX-series CPU Unit Built-in EtherCAT Port User's Manual	NJ/NX-series Troubleshooting Manual
Introduction to NX1P2 CPU Units	●							
Setting devices and hardware	●							
Using motion control					●			
Using EtherCAT							●	
Using EtherNet/IP								●
Software settings								
Using motion control					●			
Using EtherCAT		●					●	
Using EtherNet/IP								●
Using the NX1P2 CPU Unit functions			●					
Writing the user program								
Using motion control					●	●		
Using EtherCAT							●	
Using EtherNet/IP		●		●				●
Programming error processing								●
Using the NX1P2 CPU Unit functions			●					
Testing operation and debugging								
Using motion control					●			
Using EtherCAT		●					●	
Using EtherNet/IP								●
Using the NX1P2 CPU Unit functions			●					
Learning about error management and corrections ^{*1}								●
Maintenance								
Using motion control	●				●			
Using EtherCAT							●	
Using EtherNet/IP								●

*1. Refer to the *NJ/NX-series Troubleshooting Manual* (Cat. No. W503) for the error management concepts and the error items.

Manual Structure

Page Structure

The following page structure is 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 with different unit versions and for different versions of the Sysmac Studio is given.

Note References are provided to more detailed or related information.

Precaution on Terminology

In this manual, “download” refers to transferring data from the Sysmac Studio to the physical Controller and “upload” refers to transferring data from the physical Controller to the Sysmac Studio.

For the Sysmac Studio, synchronization is used to both upload and download data. Here, “synchronize” means to automatically compare the data for the Sysmac Studio on the computer with the data in the physical Controller and transfer the data in the direction that is specified by the user.

Terms and Conditions Agreement

Warranty, Limitations of Liability

Warranties

● Exclusive Warranty

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

● Limitations

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

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

● Buyer Remedy

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

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

Limitation on Liability; Etc

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

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

Application Considerations

Suitability of Use

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

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

Programmable Products

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

Disclaimers

Performance Data

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

Change in Specifications

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

Errors and Omissions

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

Safety Precautions

Refer to the following manuals for safety precautions.

- NX-series NX1P2 CPU Unit Hardware User's Manual (Cat. No. W578)

Precautions for Safe Use

Refer to the following manuals for precautions for safe use.

- NX-series NX1P2 CPU Unit Hardware User's Manual (Cat. No. W578)

Precautions for Correct Use

Refer to the following manuals for precautions for correct use.

- NX-series NX1P2 CPU Unit Hardware User's Manual (Cat. No. W578)

Regulations and Standards

Refer to the following manuals for regulations and standards.

- NX-series NX1P2 CPU Unit Hardware User's Manual (Cat. No. W578)

Versions

Hardware revisions and unit versions are used to manage the hardware and software in NX-series Units and EtherCAT slaves.

The hardware revision or unit version is updated each time there is a change in hardware or software specifications. Even when two Units or EtherCAT slaves have the same model number, they will have functional or performance differences if they have different hardware revisions or unit versions.

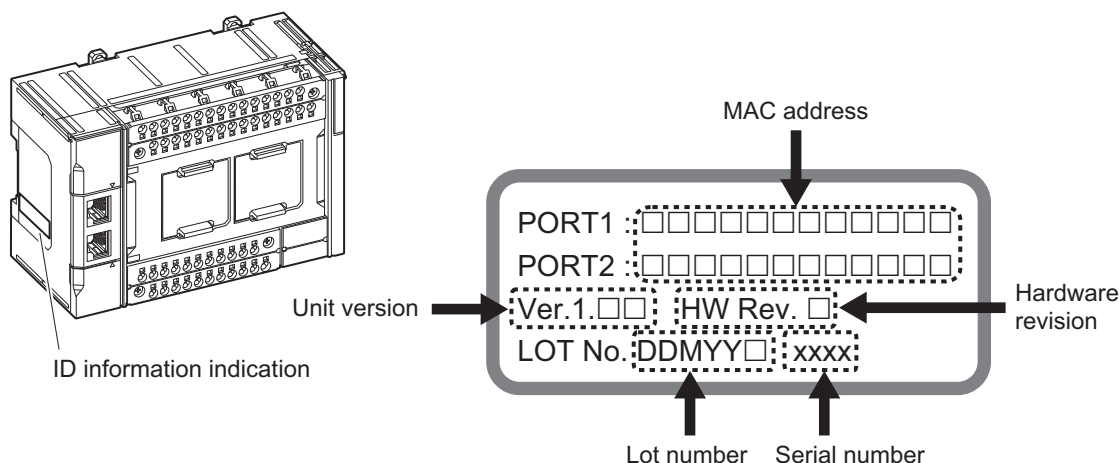
Checking Versions

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

Checking Unit Versions on ID Information Indications

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

The ID information on an NX-series NX1P2-□□□□□□ CPU Unit is shown below.



Note The hardware revision is not displayed for the Unit that the hardware revision is in blank.

Checking Unit Versions with the Sysmac Studio

● Checking the Unit Version of a Unit

You can use the Production Information while the Sysmac Studio is online to check the unit version of a Unit.

You can do this for the CPU Unit, NX Units on the CPU Rack, and Option Boards.

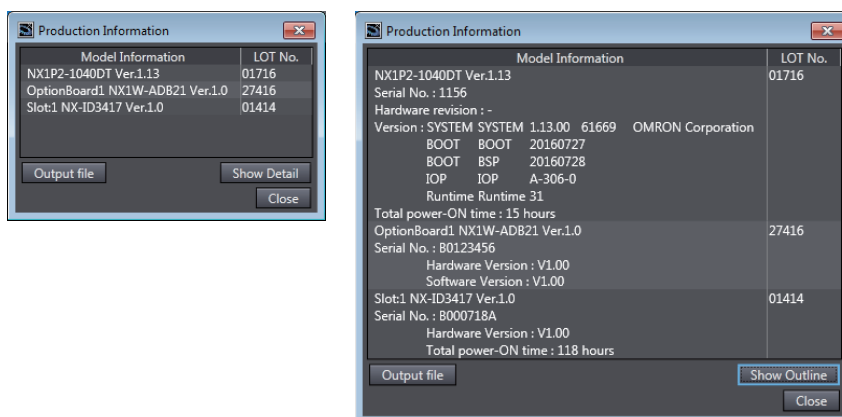
Use the following procedure to check the unit version.

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

The Production Information Dialog Box is displayed.

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

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



Outline View

Detail View

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

Note The hardware revision is separated by "/" and displayed on the right of the hardware version. The hardware revision is not displayed for the Unit that the hardware revision is in blank.

● Checking the Unit Version of an EtherCAT Slave

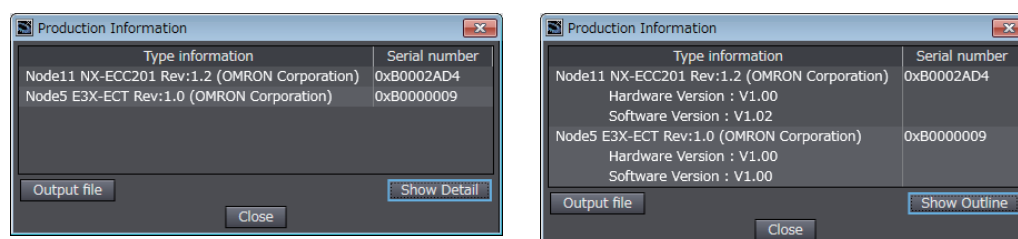
You can use the Production Information while the Sysmac Studio is online to check the unit version of an EtherCAT slave. Use the following procedure to check the unit version.

- 1 Double-click **EtherCAT** under **Configurations and Setup** in the Multiview Explorer. Or, right-click **EtherCAT** under **Configurations and Setup** and select **Edit** from the menu.
The EtherCAT Tab Page is displayed for the Controller Configurations and Setup Layer.
- 2 Right-click the master on the EtherCAT Tab Page and select **Display Production Information**.
The Production Information Dialog Box is displayed.
The unit version is displayed after "Rev."

● Changing Information Displayed in Production Information Dialog Box

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

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



Outline View

Detail View

Unit Versions of CPU Units and Sysmac Studio Versions

The functions that are supported depend on the unit version of the NX-series CPU 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-1 Version Information* on page A-2 for the relationship between the unit versions of the CPU Units and the Sysmac Studio versions, and for the functions that are supported by each unit version.

Related Manuals

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

Manual name	Cat. No.	Model numbers	Application	Description
NX-series NX1P2 CPU Unit Hardware User's Manual	W578	NX1P2-□□□□	Learning the basic specifications of the NX1P2 CPU Units, including introductory information, designing, installation, and maintenance. Mainly hardware information is provided.	An introduction to the entire NX1P2 system is provided along with the following information on the CPU Unit. • Features and system configuration • Introduction • Part names and functions • General specifications • Installation and wiring • Maintenance and inspection
NJ/NX-series CPU Unit Software User's Manual	W501	NX701-□□□□ NX1P2-□□□□ NJ501-□□□□ NJ301-□□□□ NJ101-□□□□	Learning how to program and set up an NJ/NX-series CPU Unit. Mainly software information is provided.	The following information is provided on a Controller built with an NJ/NX-series CPU Unit. • CPU Unit operation • CPU Unit features • Initial settings • Programming based on IEC 61131-3 language specifications
NX-series NX1P2 CPU Unit Built-in I/O and Option Board User's Manual	W579	NX1P2-□□□□	Learning about the details of functions only for an NX-series NX1P2 CPU Unit and an introduction of functions for an NJ/NX-series CPU Unit.	Of the functions for an NX1P2 CPU Unit, the following information is provided. • Built-in I/O • Serial Communications Option Boards • Analog I/O Option Boards An introduction of following functions for an NJ/NX-series CPU Unit is also provided. • Motion control functions • EtherNet/IP communications functions • EtherCAT communications functions
NJ/NX-series Instructions Reference Manual	W502	NX701-□□□□ NX1P2-□□□□ NJ501-□□□□ NJ301-□□□□ NJ101-□□□□	Learning detailed specifications on the basic instructions of an NJ/NX-series CPU Unit.	The instructions in the instruction set (IEC 61131-3 specifications) are described.
NJ/NX-series CPU Unit Motion Control User's Manual	W507	NX701-□□□□ NX1P2-□□□□ NJ501-□□□□ NJ301-□□□□ NJ101-□□□□	Learning about motion control settings and programming concepts.	The settings and operation of the CPU Unit and programming concepts for motion control are described.

Manual name	Cat. No.	Model numbers	Application	Description
NJ/NX-series Motion Control Instruc- tions Reference Manual	W508	NX701-□□□□ NX1P2-□□□□ NJ501-□□□□ NJ301-□□□□ NJ101-□□□□	Learning about the specifications of the motion control instructions.	The motion control instructions are described.
NJ/NX-series CPU Unit Built-in Ether- CAT® Port User's Manual	W505	NX701-□□□□ NX1P2-□□□□ NJ501-□□□□ NJ301-□□□□ NJ101-□□□□	Using the built-in EtherCAT port on an NJ/NX-series CPU Unit.	Information on the built-in EtherCAT port is provided. This manual provides an introduction and provides information on the con- figuration, features, and setup.
NJ/NX-series CPU Unit Built-in EtherNet/IP™ port User's Manual	W506	NX701-□□□□ NX1P2-□□□□ NJ501-□□□□ NJ301-□□□□ NJ101-□□□□	Using the built-in EtherNet/IP port on an NJ/NX-series CPU Unit.	Information on the built-in Ether- Net/IP port is provided. Information is provided on the basic setup, tag data links, and other fea- tures.
NJ/NX-series Troubleshooting Manual	W503	NX701-□□□□ NX1P2-□□□□ NJ501-□□□□ NJ301-□□□□ NJ101-□□□□	Learning about the errors that may be detected in an NJ/NX-series Con- troller.	Describes concepts on managing errors that may be detected in an NJ/NX-series Controller and informa- tion on individual errors.
Sysmac Studio Version 1 Operation Manual	W504	SYSMAC- SE2□□□	Learning about the operating proce- dures and func- tions of the Sysmac Studio.	Describes the operating procedures of the Sysmac Studio.
NX-series EtherCAT® Coupler Unit User's Manual	W519	NX-ECC20□	Learning how to use an NX-series EtherCAT Coupler Unit and Ether- CAT Slave Termi- nals	The following items are described: the overall system and configuration methods of an EtherCAT Slave Termi- nal (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 config- ure 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.

Manual name	Cat. No.	Model numbers	Application	Description
NX-series NX Units User's Manuals	W521	NX-ID□□□□ NX-IA□□□□ NX-OC□□□□ NX-OD□□□□ NX-MD□□□□	Learning how to use NX Units.	Describes the hardware, setup methods, and functions of the NX Units. Manuals are available for the following Units. Digital I/O Units, Analog I/O Units, System Units, Position Interface Units, Communications Interface Units, Load Cell Input Units, and IO-Link Master Units.
	W522	NX-AD□□□□ NX-DA□□□□		
	W566	NX-TS□□□□ NX-HB□□□□		
	W523	NX-PD1□□□ NX-PF0□□□ NX-PC0□□□ NX-TBX01		
	W524	NX-EC0□□□ NX-ECS□□□ NX-PG0□□□		
	W540	NX-CIF□□□		
	W565	NX-RS□□□□		
	W567	NX-ILM□□□		
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.
NA-series Programmable Terminal Software User's Manual	V118	NA5-□W□□□□	Learning about NA-series PT pages and object functions.	Describes the pages and object functions of the NA-series Programmable Terminals.
NS-series Programmable Terminals Programming Manual	V073	NS15-□□□□□ NS12-□□□□□ NS10-□□□□□ NS8-□□□□□ NS5-□□□□□	Learning how to use the NS-series Programmable Terminals.	Describes the setup methods, functions, etc. of the NS-series Programmable Terminals.
NB-series Programmable Terminals NB-Designer Operation Manual	V106	NB□Q-TW01B NB□W-TW01B	Learning about the screens and object functions of NB-series Programmable Terminals.	Describes the screens and object functions of NB-series Programmable Terminals. The procedure for installing the NB-Designer, an overview of managing the screen data of NB-series Programmable Terminals with the NBManager, and information on maintenance after operation and troubleshooting are also provided.
NB-series Programmable Terminals Setup Manual	V107	NB□Q-TW01B NB□W-TW01B	Learning the specifications and settings required to install an NB-series Programmable Terminal and connect peripheral devices.	Information is provided on NB-series Programmable Terminal specifications, part names, installation procedures, procedures to connect an NB-series Programmable Terminal to peripheral devices, and settings required after connection to start communications and operations.
E5□C Digital Temperature Controllers Communications Manual	H175	E5□C	Learning about the communications functions of E5□C Digital Temperature Controllers	Provides an overview of the communications method, communications specifications, and wiring of E5□C Digital Temperature Controllers.

Manual name	Cat. No.	Model numbers	Application	Description
E5□C Digital Temperature Controllers User's Manual	H174	E5□C	Learning about the functions of E5□C Digital Temperature Controllers	Describes how to use E5□C Digital Temperature Controllers.

Terminology

Term	Description
absolute encoder home offsets	This data is used to restore in the CPU Unit the actual position of a Servo Drive with an absolute encoder. The offset is the difference between the command position after homing and the absolute data that is read from the absolute encoder.
array specification	One of the variable specifications. An array variable contains multiple elements of the same data type. The elements in the array are specified by serial numbers called subscripts that start from the beginning of the array.
AT	One of the attributes of a variable. This attribute allows the user to specify what is assigned to a variable. An I/O port or an address in memory used for CJ-series Units can be specified.
axes groups	A functional unit that groups together axes within the Motion Control Function Module.
Axes Group Variable	A system-defined variable that is defined as a structure and provides status information and some of the axes parameters for an individual axes group. An Axes Group Variable is used to specify an axes group for motion control instructions and to monitor the command interpolation velocity, error information, and other information for the axes group.
axis	A functional unit within the Motion Control Function Module. An axis is assigned to the drive mechanism in an external Servo Drive or the sensing mechanism in an external Encoder Input Slave Unit.
Axis Variable	A system-defined variable that is defined as a structure and provides status information and some of the axis parameters for an individual axis. An Axis Variable is used to specify an axis for motion control instructions and to monitor the command position, error information, and other information for the axis.
basic data type	Any of the data types that are defined by IEC 61131-3. They include Boolean, bit string, integer, real, duration, date, time of day, date and time, and text string data types. “Basic data type” is used as opposed to derivative data types, which are defined by the user.
cam data variable	A variable that represents the cam data as a structure array. A cam data variable is an array structure that consists of phases and displacements.
CJ-series Unit	Any of the CJ-series Units that can be used with an NJ-series Controller.
Communications Coupler Unit	The generic name of an interface unit for remote I/O communications on a network between NX Units and a host network master. For example, an EtherCAT Coupler Unit is a Communications Coupler Unit for an EtherCAT network.
Constant	One of the attributes of a variable. If you specify the Constant attribute for a variable, the value of the variable cannot be written by any instructions, ST operators, or CIP message communications.
Controller	The range of devices that are directly controlled by the CPU Unit. In the NX-series System, the Controller includes the CPU Rack and EtherCAT slaves (including general-purpose slaves and Servo Drives). In the NJ-series System, the Controller includes the CPU Rack, Expansion Racks, and EtherCAT slaves (including general-purpose slaves and Servo Drives).
Controller error	Errors that are defined by the NJ/NX-series System. “Controller error” is a collective term for major fault level, partial fault level, minor fault level, and observation Controller events.
Controller event	One of the events in the NJ/NX-series System. Controller events are errors and information that are defined by the system for user notification. A Controller event occurs when the system detects a factor that is defined as a Controller event.
Controller information	Information that is defined by the NJ/NX-series System that is not an error. It represents an information Controller event.

Term	Description
CPU Unit	The Unit that serves as the center of control for a Machine Automation Controller. The CPU Unit executes tasks, refreshes I/O for other Units and slaves, etc. The NJ/NX-series CPU Units include NX701-□□□□, NX1P2-□□□□, NJ501-□□□□, and NJ301-□□□□.
derivative data type	A data type that is defined by the user. Structures, unions, and enumerations are derivative data types.
device	A general term for any Unit or slave that is refreshed by the I/O refreshing that is performed by the CPU Unit. Specifically, it refers to EtherCAT slaves, NX Units on the CPU Unit, built-in I/O, Option Boards, and CJ-series Units.
device output	An output for any Unit or slave that is refreshed by the I/O refreshing that is performed by the CPU Unit.
device variable	A variable that is used to access a specific device through an I/O port.
download	To transfer data from the Sysmac Studio to the Controller with the synchronization operation of the Sysmac Studio.
edge	One of the attributes of a variable. This attribute makes a BOOL variable pass TRUE to a function block when the variable changes from FALSE to TRUE or when it changes from TRUE to FALSE.
enumeration	One of the derivative data types. This data type takes one item from a prepared name list of enumerators as its value.
enumerator	One of the values that an enumeration can take expressed as a character string. The value of an enumeration is one of the enumerators.
EtherCAT Master Function Module	One of the function modules. This function module controls the EtherCAT slaves as the EtherCAT master.
EtherNet/IP Function Module	One of the function modules. This function module controls the built-in EtherNet/IP port.
event log	A function that recognizes and records errors and other events.
Event Setup	Settings that define user-defined errors and user-defined information.
event task	A task that executes a user program only once when the task execution conditions are met.
FB	An acronym for “function block.”
forced refreshing	Forcing the refreshing of an input from an external device or an output to an external device, e.g., when the user debugs a program. Addresses that are subject to forced refreshing can still be overwritten from the user program.
FUN	An abbreviation for “function.”
function	A POU that is used to create an object that determines a unique output for the same input, such as for data processing.
function block	A POU that is used to create an object that can have a different output for the same input, such as for a timer or counter.
function module	One of the functional units of the software configuration of the CPU Unit.
general-purpose slave	Any of the EtherCAT slaves that cannot be assigned to an axis.
global variable	A variable that can be read or written from all POUs (programs, functions, and function blocks).
I/O map settings	Settings that assign variables to I/O ports. Assignment information between I/O ports and variables.
I/O port	A logical interface that is used by the CPU Unit to exchange data with an external device (slave or Unit).
I/O refreshing	Cyclic data exchange with external devices that is performed with predetermined memory addresses.
information	One of the event levels for Controller events or user-defined events. These are not errors, but appear in the event log to notify the user of specific information.

Term	Description
Initial Value	One of the attributes of a variable. The variable is set to the initial value in the following situations. • When power is turned ON • When the CPU Unit changes to RUN mode • When you specify to initialize the values when the user program is transferred • When a major fault level Controller error occurs
inline ST instruction	ST programming that is included within a ladder diagram program.
literal	A constant expression that is used in a user program.
local variable	A variable that can be accessed only from inside the POU in which it is defined. "Local variable" is used as opposed to "global variable." Local variables include internal variables, input variables, output variables, in-out variables, and external variables.
main memory	The memory inside the CPU Unit that is used by the CPU Unit to execute the OS and user program.
major fault level Controller error	An error for which all NJ/NX-series Controller control operations stop. The CPU Unit immediately stops user program execution and turns OFF the loads for all slaves and Units (including remote I/O).
MC Test Run	A function to check motor operation and wiring from the Sysmac Studio.
memory used for CJ-series Units	One type of I/O memory in an NX1P2 CPU Unit and NJ-series CPU Unit. It contains addresses that can be directly specified by the user. It can be accessed only with variables with an AT attribute. This memory is used to access CJ-series Units and CJ-series networks. However, you cannot connect the CJ-series Units to the NX1P2 CPU Units.
minor fault level Controller error	An error for which part of the control operations for one of the function modules in the NJ/NX-series Controller stop. The NJ/NX-series CPU Unit continues operation even after a minor fault level Controller error occurs.
Motion Control Function Module	One of the function modules. The MC Function Module performs motion control based on commands from the motion control instructions that are executed in the user program.
motion control instruction	A function block instruction that executes motion control. The Motion Control Function Module supports instructions that are based on function blocks for PLCopen [®] motion control as well as instructions developed specifically for the Motion Control Function Module.
namespace	A system that is used to group and nest the names of functions, function block definitions, and data types.
Network Publish	One of the attributes of a variable. This attribute allows you to use CIP message communications or tag data links to read/write variables from another Controller or from a host computer.
NX bus	The NX-series internal bus. An NX1P2 CPU Unit has the NX bus.
NX Units	Any of the NX-series Units that perform I/O processing with connected external devices. The Communications Coupler Units are not included with the NX Units.
observation	One of the event levels for Controller events or user-defined events. These are minor errors that do not affect control operations, but appear in the event log to notify the user of specific information.
partial fault level Controller error	An error for which all of the control operations for one of the function modules in the NJ/NX-series Controller stop. An NJ/NX-series CPU Unit continues operation even after a partial fault level Controller error.
PDO communications	An abbreviation for process data communications. Data is exchanged between the master and slaves on a process data communications cycle. (The process data communications cycle is the same as the task period of the primary periodic task.)

Term	Description
periodic task	A task for which user program execution and I/O refreshing are performed each period.
PLC Function Module	One of the function modules. This function module executes the user program, sends commands to the Motion Control Function Module, and provides an interface to the USB and SD Memory Card.
POU	An acronym for “program organization unit.” A POU is a unit in a program execution model that is defined in IEC 61131-3. A POU contains an algorithm and a local variable table and forms the basic unit used to build a user program. There are three types of POUs: programs, functions, and function blocks.
primary periodic task	The task with the highest priority.
process data communications	One type of EtherCAT communications in which process data objects (PDOs) are used to exchange information cyclically and in realtime. Process data communications are also called PDO communications.
program	Along with functions and function blocks, one of the three types of POUs. Programs are assigned to tasks to execute them.
Range Specification	One of the variable specifications. You can specify a range for a variable in advance. The variable can take only values that are in the specified range.
Retain	One of the attributes of a variable. The values of variables with a Retain attribute are held at the following times. (Variables without a Retain attribute are set to their initial values.) • When power is turned ON after power interruption • When the CPU Unit changes to RUN mode • When you specify to not initialize the values when the user program is transferred
SDO communications	One type of EtherCAT communications in which service data objects (SDOs) are used to transmit information whenever required.
Servo Drive/encoder input slave	Any of the EtherCAT slaves that is assigned to an axis. In the NJ/NX-series System, it would be a Servo Drive or Encoder Input Slave Unit.
slave	A device that performs remote I/O for a master.
slave and Unit configurations	A generic term for the EtherCAT configuration and Unit configuration.
Slave Terminal	A building-block remote I/O terminal to which a Communications Coupler Unit and NX Units are mounted. A Slave Terminal is one type of slave.
Special Unit Setup	A generic term for the settings for a Special Unit, including the settings in allocated DM Area words.
structure	One of the derivative data types. It consists of multiple data types placed together into a layered structure.
synchronization	A function that automatically compares the information in the NJ/NX-series Controller with the information in the Sysmac Studio, displays any differences and locations in a hierarchical form, and can be used to synchronize the information.
Sysmac Studio	A computer software application for setting, programming, debugging, and troubleshooting NJ/NX-series Controllers. It also provides operations for motion control and a Simulator.
system common processing	System processing that is performed by the CPU Unit to perform I/O refreshing and the user program execution within a task. Exclusive control of variables between tasks, data trace processing, and other processing is performed.
system service	Processing that is performed by the CPU Unit in unused time between task processing. The system service includes communications processing, SD Memory Card access processing, self-diagnosis processing, and other processing.
system-defined variable	A variable for which all attributes are defined by the system and cannot be changed by the user.
task	An attribute that defines when a program is executed.
task period	The interval at which the primary periodic task or a periodic task is executed.
union	One of the derivative data types. It allows you to handle the same data as different data types.
Unit	A device that mounts to the CPU Rack or an Expansion Rack.

Term	Description
Unit configuration	The configuration information for the Units that are set on the Sysmac Studio. This information tells what Unit models are connected to the CPU Unit and where they are connected.
upload	To transfer data from the Controller to the Sysmac Studio with the synchronization operation of the Sysmac Studio.
user program	All of the programs in one project.
user-defined event	One of the events in the NJ/NX-series System. These events are defined by the user. "User-defined events" is a generic term for user-defined errors and user-defined information.
user-defined variable	A variable for which all of the attributes are defined by the user and can be changed by the user.
variable	A representation of data, such as a numeric value or character string, that is used in a user program. You can change the value of a variable by assigned the required value. "Variable" is used as opposed to "constant," for which the value does not change.
variable memory	A memory area that contains the present values of variables that do not have AT specifications. It can be accessed only with variables without an AT attribute.

Revision History

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

Cat. No.	W579-E1-04
-----------------	-------------------

↑
Revision code

Revision code	Date	Revised content
01	October 2016	Original production
02	January 2017	Corrected mistakes.
03	October 2017	Added shipbuilding standards (LR).
04	January 2019	Made changes accompanying the transfer of explanation for event codes and errors to the <i>NJ/NX-series Troubleshooting Manual</i> .

Sections in this Manual

1	Introduction to NX1P2 CPU Units	A	Appendices	1	A
2	Built-in I/O	I	Index	2	I
3	Option Boards			3	
4	Serial Communications			4	
5	Analog I/O			5	
6	Introduction of Motion Control Functions			6	
7	Introduction of EtherNet/IP Communications Functions			7	
8	Introduction of EtherCAT Communications Functions			8	
9	Troubleshooting			9	

Introduction to NX1P2 CPU Units

This section describes the specifications and operating procedure of the NX1P2 CPU Units.

1-1	Function Specifications for NX1P2 CPU Units	1-2
1-2	Overall Operating Procedure	1-7
1-2-1	Overall Operating Procedure	1-7
1-2-2	Procedure Details	1-8

1-1 Function Specifications for NX1P2 CPU Units

This following table gives the main specifications of the NX1P2 CPU Units.

Item				NX1P2-		
				11□□□□□	10□□□□□	90□□□□□
Processing time	Instruction execution times	LD instruction		3.3 ns		
		Math instructions (for long real data)		70 ns or more		
Program-ming	Program capacity*1	Size		1.5 MB		
		Quantity	Number of POU definitions	450		
			Number of POU instances	1,800		
	Memory capacity for variables*2	Retain attri-butes	Size	32 KB		
			Number of vari-ables	5,000		
		No Retain attributes	Size	2 MB		
			Number of vari-ables	90,000		
	Data types	Number of data types		1,000		
	Memory for CJ-series Units (Can be specified with AT specifica-tions for vari-ables.)	CIO Area		0 to 6,144 words (CIO 0 to CIO 6,143)*3		
		Work Area		0 to 512 words (W0 to W511)*3		
		Holding Area		0 to 1,536 words (H0 to H1,535)*4		
		DM Area		0 to 16,000 words (D0 to D15,999)*4		
		EM Area		---		

Item				NX1P2-			
				11□□□□□	10□□□□□	90□□□□□	
Motion control	Number of controlled axes*5	Maximum number of controlled axes		12 axes	10 axes	4 axes	
			Motion control axes	8 axes	6 axes	---	
			Single-axis position control axes	4 axes			
		Maximum number of used real axes		8 axes	6 axes	4 axes	
			Used motion control servo axes	4 axes	2 axes	---	
			Used single-axis position control servo axes	4 axes			
		Maximum number of axes for linear interpolation axis control	4 axes per axes group			---	
		Number of axes for circular interpolation axis control	2 axes per axes group			---	
		Maximum number of axes groups		8 axes groups			---
	Motion control period		Same as the period for primary periodic task				
	Cams	Number of cam data points	Maximum points per cam table	65,535 points			---
			Maximum points for all cam tables	262,140 points			---
		Maximum number of cam tables		80 tables			---
	Position units			Pulse, mm, μm, nm, degree, and inch			
	Override Factors			0.00% or 0.01% to 500.00%			

Item			NX1P2-			
			11□□□□□	10□□□□□	90□□□□□	
Built-in Ether-Net/IP port	Number of ports		1			
	Physical layer		10BASE-T/100BASE-TX			
	Frame length		1,514 bytes max.			
	Media access method		CSMA/CD			
	Modulation		Baseband			
	Topology		Star			
	Baud rate		100 Mbps (100BASE-TX)			
	Transmission media		STP (shielded twisted-pair) cable of Ethernet category 5, 5e, or higher			
	Maximum transmission distance between Ethernet switch and node		100 m			
	Maximum number of cascade connections		There are no restrictions if an Ethernet switch is used.			
	CIP service: Tag data links (cyclic communications)	Maximum number of connections		32		
		Packet interval*6		Can be set for each connection. 2 to 10,000 ms in 1-ms increments		
		Permissible communications band		3,000 pps*7 (including heartbeat)		
		Maximum number of tag sets		32		
		Tag types		Network variables, CIO, Work, Holding and DM Areas		
		Number of tags per connection (= 1 tag set)		8 (7 tags if Controller status is included in the tag set.)		
		Maximum number of tags		256		
		Maximum link data size per node (total size for all tags)		19,200 bytes		
		Maximum data size per connection		600 bytes		
		Maximum number of registrable tag sets		32 (1 connection = 1 tag set)		
		Maximum tag set size		600 bytes (Two bytes are used if Controller status is included in the tag set.)		
		Multi-cast packet filter*8		Supported		
	CIP message service: Explicit messages	Class 3 (number of connections)		32 (clients plus server)		
UCMM (non-connection type)		Maximum number of clients that can communicate at one time	32			
		Maximum number of servers that can communicate at one time	32			
Number of TCP sockets			30			

Item			NX1P2-		
			11□□□□□	10□□□□□	90□□□□□
Built-in Ether-CAT port	Communications standard		IEC 61158 Type12		
	EtherCAT master specifications		Class B (Feature Pack Motion Control compliant)		
	Physical layer		100BASE-TX		
	Modulation		Baseband		
	Baud rate		100 Mbps (100BASE-TX)		
	Duplex mode		Auto		
	Topology		Line, daisy chain, and branching		
	Transmission media		Twisted-pair cable of category 5 or higher (double-shielded straight cable with aluminum tape and braiding)		
	Maximum transmission distance between nodes		100 m		
	Maximum number of slaves		16		
	Range of node addresses that can be set		1 to 192		
	Maximum process data size		Input: 1,434 bytes Output: 1,434 bytes However, the maximum number of process data frames is 1.		
	Maximum process data size per slave		Input: 1,434 bytes Output: 1,434 bytes		
	Communications cycle		2,000 to 8,000 μs (in 250-μs increments)		
	Sync jitter		1 μs max.		
Serial communications (Serial Communications Option Board)	Communications method		Half duplex		
	Synchronization		Start-stop		
	Baud rate		1.2/2.4/4.8/9.6/19.2/38.4/57.6/115.2 kbps		
	Transmission distance		Depends on Option Board.		
	Supported protocol		Host link (FINS), Modbus-RTU master, and no-protocol		
Unit configuration	Maximum number of connectable Units	Maximum number of NX Units per CPU Rack	8		
		Maximum number of NX Units for entire controller	24 On CPU Rack: 8 On EtherCAT Slave Terminals: 16		
	Power supply	Model	A non-isolated power supply for DC input is built into the CPU Unit.		
		Power OFF detection time	2 to 8 ms		
Option Board	Number of slots		2	2	1
Built-in I/O	Input	Number of points	24	24	14
	Output	Number of points	16	16	10
		Load-short circuit protection	11□□□□/10□□□□/90□□□□: Not provided (NPN) 11□□□□1/10□□□□1/90□□□□1: Provided (PNP)		
		Pulse output	Not provided		
Internal clock	Accuracy		At ambient temperature of 55°C: -3.5 to 0.5 min error per month At ambient temperature of 25°C: -1.5 to 1.5 min error per month At ambient temperature of 0°C: -3 to 1 min error per month		
	Retention time of built-in capacitor		At ambient temperature of 40°C: 10 days		

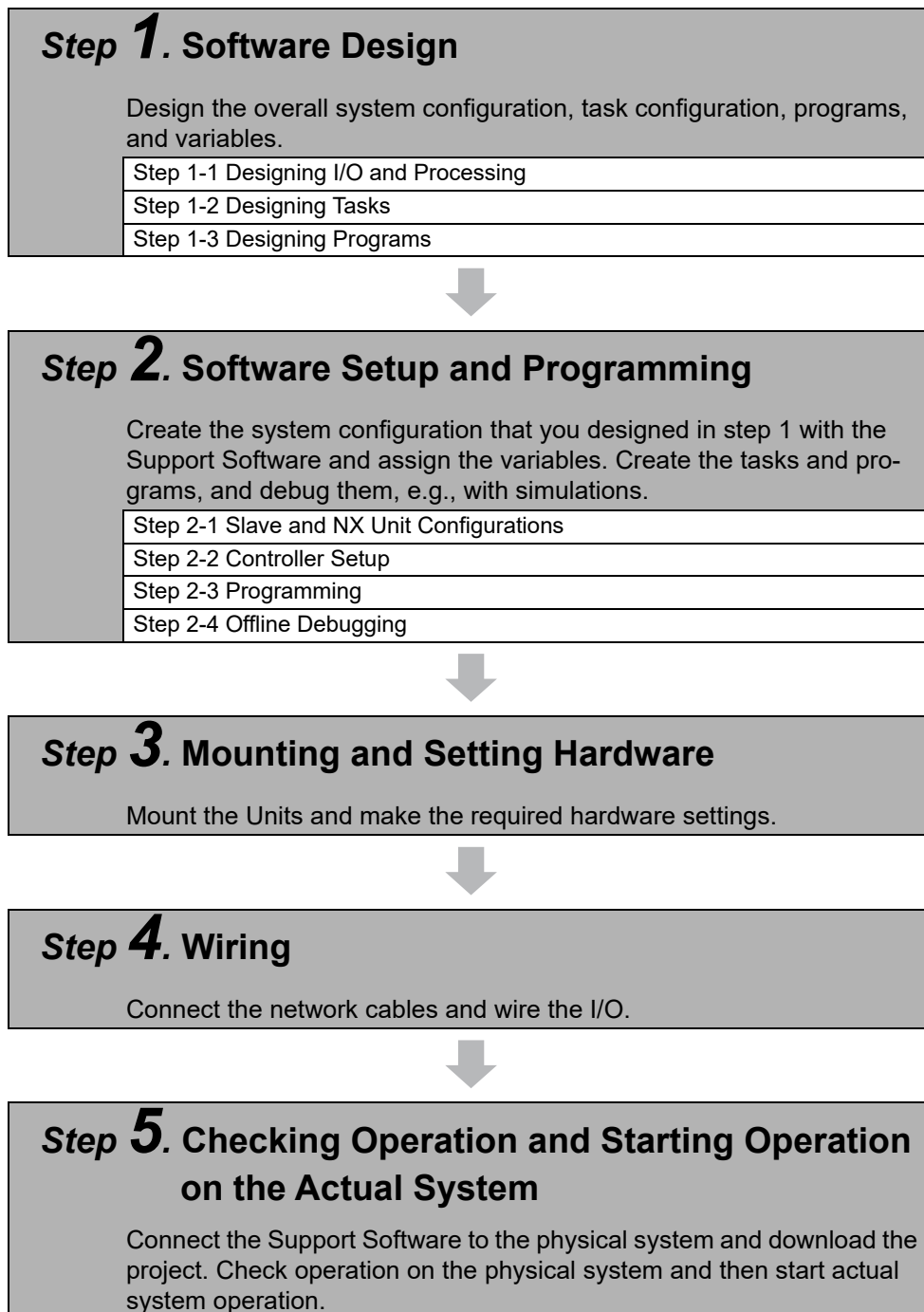
- *1. Execution objects and variable tables (including variable names)
- *2. Memory used for CJ-series Units is included.
- *3. The value can be set in 1-word increments. The value is included in the total size of variables without Retain attributes.
- *4. The value can be set in 1-word increments. The value is included in the total size of variables with Retain attributes.
- *5. For details on each axis, refer to the *NJ/NX-series CPU Unit Motion Control User's Manual* (Cat. No. W507).
- *6. Data will be refreshed at the set interval, regardless of the number of nodes.
- *7. "pps" means packets per second, i.e., the number of communications packets that can be sent or received in one second.
- *8. As the EtherNet/IP port implements the IGMP client, unnecessary multi-cast packets can be filtered by using an Ethernet switch that supports IGMP Snooping.

1-2 Overall Operating Procedure

The overall operating procedure of the NX1P2 CPU Units is shown below, with each step of the procedure described in detail.

1-2-1 Overall Operating Procedure

The overall procedure to use an NX1P2 CPU Unit is given below.



1-2-2 Procedure Details

Step 1. Software Design

Step	Description	Reference
Step 1-1 Designing I/O and Processing	- External I/O devices and unit configuration - Refresh periods for external devices - Program contents	<i>NX-series NX1P2 CPU Unit Hardware User's Manual</i> (Cat. No. W578)



Step 1-2 Designing Tasks	- Task configuration - Relationship between tasks and programs - Task periods - Slave, NX Unit, and built-in I/O refresh times - Exclusive control methods for variables between tasks	<i>NJ/NX-series CPU Unit Software User's Manual</i> (Cat. No. W501)
-------------------------------------	--	---



Step 1-3 Designing Programs		
POU (Program Organization Unit) Design	- Programs - Functions and function blocks - Determining the algorithm languages	<i>NJ/NX-series CPU Unit Software User's Manual</i> (Cat. No. W501)
Variable Design	- Defining variables that you can use in more than one POU and variables that you use in only specific POUs - Defining the variables names for the device variables that you use to access slaves, NX Units, and the built-in I/O - Defining the attributes of variables, such as the Name and Retain attributes - Designing the data types of variables	<i>NJ/NX-series CPU Unit Software User's Manual</i> (Cat. No. W501)



Step 2. Software Setup and Programming

Step	Description	Sysmac Studio Operations	Reference
Project Creation	- Create a project in the Sysmac Studio. - Insert a Controller.	New Project Button Insert - Controller	<i>Sysmac Studio Version 1 Operation Manual</i> (Cat. No. W504)



The following *Controller Configurations and Setup* and the *Programming and Task Settings* can be performed in either order.

Step 2-1 Slave and NX Unit Configurations			
1. Creating the Slave and NX Unit Configurations	- Creating the slave configuration and NX Unit configuration either offline or online. (For online configuration, make the online connection that is described in step 5.) - Setting up any Slave Terminals that are used.	EtherCAT Tab Page CPU and Expansion Racks Tab Page Slave Terminal Tab Page	<i>NJ/NX-series CPU Unit Software User's Manual</i> (Cat. No. W501) <i>NX-series EtherCAT Coupler Units User's Manual</i> (Cat. No. W519)



2. Assigning Device Variables to I/O Ports	Registering device variables in the variable table (Variable names are user-defined or automatically created.)	I/O Map	<i>NJ/NX-series CPU Unit Software User's Manual</i> (Cat. No. W501)
---	---	---------	---



(The following step is for motion control.)

3. Creating the Axes and Assigning Them to the Servo Drive/Encoder Input Slaves	Creating the axes and setting them as real axes or virtual axes. Creating axes groups to perform interpolated axes control.	Configurations and Setup – Motion Control Setup	<i>NJ/NX-series CPU Unit Software User's Manual</i> (Cat. No. W501)
--	--	--	---



Step 2-2 Controller Setup	Setting the following parameters from the Sysmac Studio		<i>NJ/NX-series CPU Unit Software User's Manual</i> (Cat. No. W501)
	Setting the initial values for the PLC Function Module	Configurations and Setup – Controller Setup – Operation Settings	
	(When battery is used)Setting the clock data with the clock function	Controller – Controller Clock	
	Setting the initial values for the NX Bus Function Module	Configurations and Setup – CPU/Expansion Racks – CPU Rack	
	(The following step is for motion control.) Setting the initial settings for the Motion Control Function Module	Configurations and Setup – Motion Control Setup	
	Setting the initial values for the EtherCAT Function Module	Configurations and Setup – EtherCAT	
	Setting the initial values for the EtherNet/IP Function Module	Configurations and Setup – Controller Setup – Built-in EtherNet/IP Port Settings	2-4-1 <i>Built-in I/O Settings</i> on page 2-8
	Setting the initial values for the built-in I/O	Configurations and Setup – Controller Setup – Built-in I/O Settings	
	Setting the initial values for an Option Board	Configurations and Setup – Controller Setup – Option Board Settings	



Step 2-3 Programming			
---------------------------------	--	--	--

1. Registering Variables	- Registering the variables used by more than one POU in the global variable table with Sysmac Studio - Registering the local variable table for each program - Registering the local variable table for each function block and function	Global Variable Table Editor Local Variable Table Editor	<i>Sysmac Studio Version 1 Operation Manual</i> (Cat. No. W504) <i>NJ/NX-series CPU Unit Software User's Manual</i> (Cat. No. W501)
2. Writing Algorithms for POUs	Writing the algorithms for the POUs (programs, function blocks, and functions) in the required languages	Programming Editor	<i>NJ/NX-series CPU Unit Software User's Manual</i> (Cat. No. W501) <i>NJ/NX-series Instructions Reference Manual</i> (Cat. No. W502) <i>NJ/NX-series Motion Control Instructions Reference Manual</i> (Cat. No. W508)
3. Setting the Tasks	Making task settings	Configurations and Setup – Task Settings	<i>NJ/NX-series CPU Unit Software User's Manual</i> (Cat. No. W501)



Step 2-4 Offline Debugging	Checking the algorithms and task execution times on the Simulator (virtual controller)		<i>NJ/NX-series CPU Unit Software User's Manual</i> (Cat. No. W501)
-----------------------------------	--	--	---



Step 3. Mounting and Setting Hardware

Step	Description	Reference
1. Mounting	- Connecting adjacent Units - Mounting to DIN Track	<i>NX-series NX1P2 CPU Unit Hardware User's Manual</i> (Cat. No. W578)
2. Setting Hardware	Setting the node addresses of the EtherCAT slaves	Operation manuals for the EtherCAT slaves



Step 4. Wiring

Step	Description	Reference
1. Connecting the Power Supply to the CPU Unit	Connecting the power supply and ground wires	<i>NX-series NX1P2 CPU Unit Hardware User's Manual</i> (Cat. No. W578)
2. Connecting Ethernet Cable	- Connecting the built-in EtherCAT port - Connecting the built-in EtherNet/IP port	<i>NJ/NX-series CPU Unit Built-in EtherCAT Port User's Manual</i> (Cat. No. W505) <i>NJ/NX-series CPU Unit Built-in EtherNet/IP Port User's Manual</i> (Cat. No. W506)

3. Wiring I/O	- Wiring the built-in I/O - Wiring I/O for NX Units - Wiring Option Boards	<i>NX-series NX1P2 CPU Unit Hardware User's Manual</i> (Cat. No. W578) Manuals for the specific NX Units
	Wiring I/O to EtherCAT slaves	Operation manuals for the EtherCAT slaves <i>NX-series NX1P2 CPU Unit Hardware User's Manual</i> (Cat. No. W578)
	Checking wiring	<i>Sysmac Studio Version 1 Operation Manual</i> (Cat. No. W504)
4. Connecting the Computer That Runs the Sysmac Studio	Connecting the built-in EtherNet/IP port	<i>Sysmac Studio Version 1 Operation Manual</i> (Cat. No. W504)



Step 5. Checking Operation and Starting Operation on the Actual System

Step	Description	Sysmac Studio Operations	Reference
1. Online Connection to Sysmac Studio and Project Download	Turn ON the power supply to the Controller and place the Sysmac Studio online. Then, download the project. *1 Perform this step before you create the slave configuration or Unit configuration from the mounted Units in step 2-1.	Controller – Communications Setup Controller – Synchronization	<i>NJ/NX-series CPU Unit Software User's Manual</i> (Cat. No. W501)



2. Operation Check on Controller	- Check the wiring by using forced refreshing of real I/O from the I/O Map or Watch Tab Page. - For motion control, use the MC Test Run operations in PROGRAM mode to check the wiring. Then check the motor rotation directions for jogging, travel distances for relative positioning (e.g., for electronic gear settings), and homing operation. - Change the Controller to RUN mode and check the operation of the user program.	---	<i>NJ/NX-series CPU Unit Software User's Manual</i> (Cat. No. W501)
---	--	-----	---



3. Actual Controller Operation	Start actual operation.	---	---
---------------------------------------	-------------------------	-----	-----

*1. Use the Synchronize Menu of the Sysmac Studio to download the project.

Built-in I/O

This section describes the built-in I/O of the NX1P2 CPU Units.

2-1 Built-in I/O Terminal Allocation	2-2
2-1-1 Terminal Arrangement	2-2
2-2 I/O Data Specifications	2-5
2-2-1 NX1P2-□□24DT/-□□24DT1	2-5
2-2-2 NX1P2-□□40DT/-□□40DT1	2-6
2-3 Built-in I/O Functions	2-7
2-4 Settings	2-8
2-4-1 Built-in I/O Settings	2-8
2-4-2 I/O Map	2-9
2-5 Functions	2-10
2-5-1 Input Filter	2-10
2-5-2 Output Load Rejection Setting	2-12
2-6 I/O Refreshing	2-13
2-6-1 I/O Refresh Timing of Built-in I/O	2-13
2-6-2 I/O Response Time of Built-in I/O	2-15

2-1 Built-in I/O Terminal Allocation

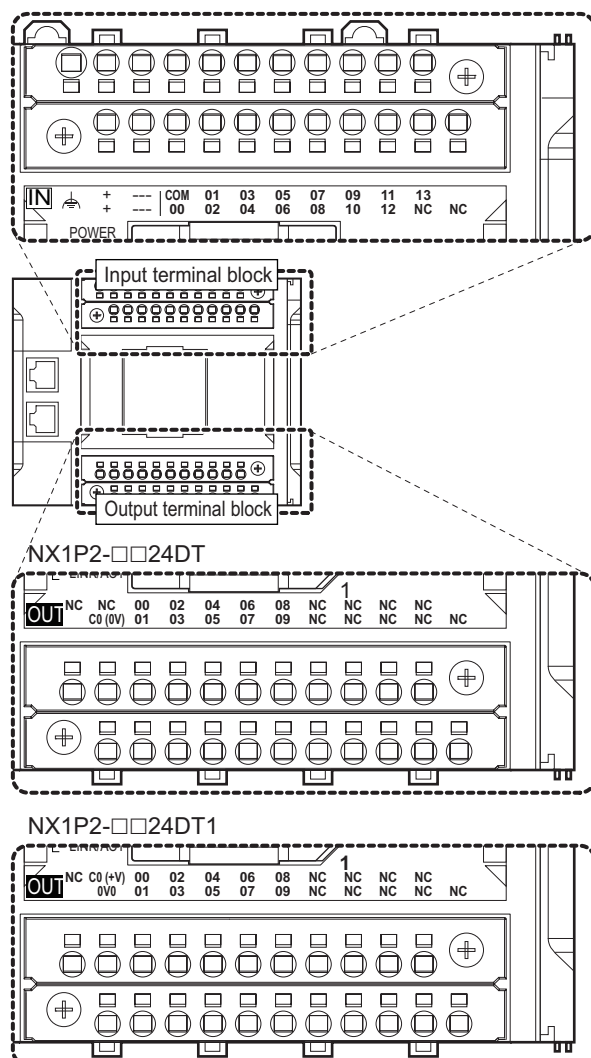
The following describes the allocation of the built-in I/O terminals.

2-1-1 Terminal Arrangement

The built-in I/O terminals are located on the terminal blocks on the front of the CPU Unit.

The arrangement of these terminals is shown below.

NX1P2-□□24DT/-□□24DT1



● Input Terminal Block

	+	-	COM	01	03	05	07	09	11	13	
	+	-	00	02	04	06	08	10	12	NC	NC

● Output Terminal Block

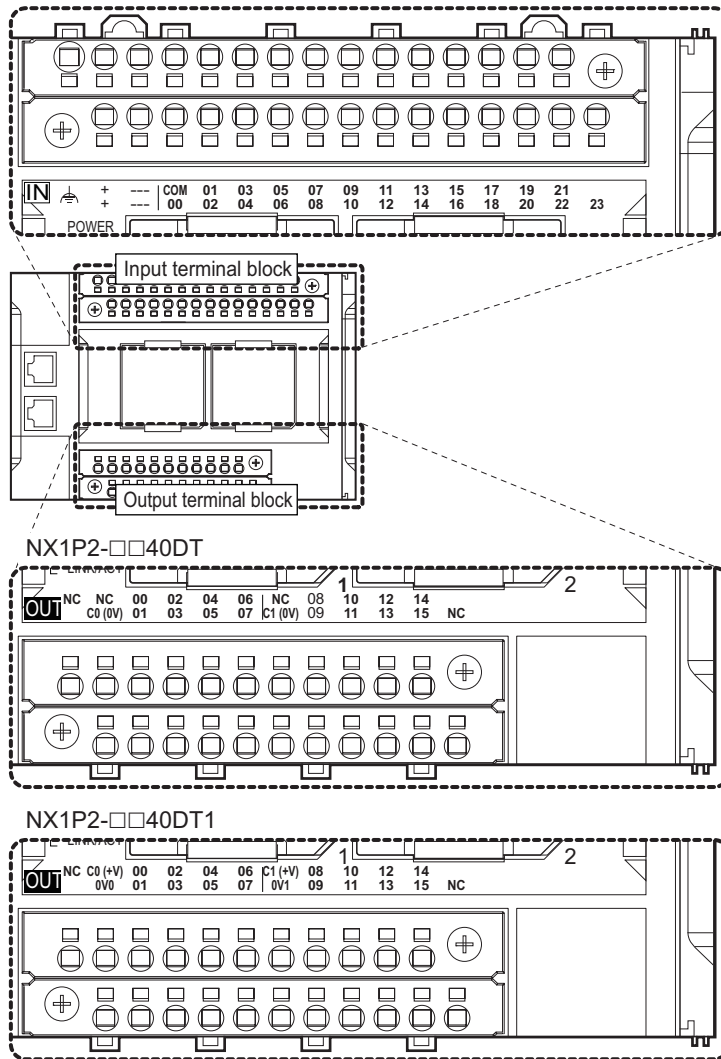
NX1P2-□□24DT

NC	NC	00	02	04	06	08	NC	NC	NC	NC	
	C0 (0V)	01	03	05	07	09	NC	NC	NC	NC	NC

NX1P2-□□24DT1

NC	C0 (+V)	00	02	04	06	08	NC	NC	NC	NC	
	0V0	01	03	05	07	09	NC	NC	NC	NC	NC

NX1P2-□□40DT/-□□40DT1



● Input Terminal Block

	+	-	COM	01	03	05	07	09	11	13	15	17	19	21	
	+	-	00	02	04	06	08	10	12	14	16	18	20	22	23

● Output Terminal Block

NX1P2-□□40DT

NC	NC	00	02	04	06	NC	08	10	12	14	
	C0 (0V)	01	03	05	07	C1 (0V)	09	11	13	15	NC

NX1P2-□□40DT1

NC	C0 (+V)	00	02	04	06	C1 (+V)	08	10	12	14	
	0V0	01	03	05	07	0V1	09	11	13	15	NC

2-2 I/O Data Specifications

The following describes the I/O data specifications for the built-in I/O.

The built-in I/O uses I/O data as I/O ports.

I/O ports are generated automatically by the Sysmac Studio.

To use I/O data in the user program, you use device variables assigned to the relevant I/O ports.

Refer to the *NJ/NX-series CPU Unit Software User's Manual* (Cat. No. W501) for I/O ports and device variables.

Refer to the *Sysmac Studio Version 1 Operation Manual* (Cat. No. W504) for how to register device variables with the Sysmac Studio.

2-2-1 NX1P2-□□24DT/-□□24DT1

● General Input

Data name	Function	Data type	I/O port name
Input Bit 00	The input value for input bit 00.	BOOL	Input Bit 00
Input Bit 01	The input value for input bit 01.	BOOL	Input Bit 01
Input Bit 02	The input value for input bit 02.	BOOL	Input Bit 02
Input Bit 03	The input value for input bit 03.	BOOL	Input Bit 03
Input Bit 04	The input value for input bit 04.	BOOL	Input Bit 04
Input Bit 05	The input value for input bit 05.	BOOL	Input Bit 05
Input Bit 06	The input value for input bit 06.	BOOL	Input Bit 06
Input Bit 07	The input value for input bit 07.	BOOL	Input Bit 07
Input Bit 08	The input value for input bit 08.	BOOL	Input Bit 08
Input Bit 09	The input value for input bit 09.	BOOL	Input Bit 09
Input Bit 10	The input value for input bit 10.	BOOL	Input Bit 10
Input Bit 11	The input value for input bit 11.	BOOL	Input Bit 11
Input Bit 12	The input value for input bit 12.	BOOL	Input Bit 12
Input Bit 13	The input value for input bit 13.	BOOL	Input Bit 13

● General Output

Data name	Function	Data type	I/O port name
Output Bit 00	The output set value for output bit 00.	BOOL	Output Bit 00
Output Bit 01	The output set value for output bit 01.	BOOL	Output Bit 01
Output Bit 02	The output set value for output bit 02.	BOOL	Output Bit 02
Output Bit 03	The output set value for output bit 03.	BOOL	Output Bit 03
Output Bit 04	The output set value for output bit 04.	BOOL	Output Bit 04
Output Bit 05	The output set value for output bit 05.	BOOL	Output Bit 05
Output Bit 06	The output set value for output bit 06.	BOOL	Output Bit 06
Output Bit 07	The output set value for output bit 07.	BOOL	Output Bit 07
Output Bit 08	The output set value for output bit 08.	BOOL	Output Bit 08
Output Bit 09	The output set value for output bit 09.	BOOL	Output Bit 09

2-2-2 NX1P2-□□40DT/-□□40DT1

● General Input

Data name	Function	Data type	I/O port name
Input Bit 00	The input value for input bit 00.	BOOL	Input Bit 00
Input Bit 01	The input value for input bit 01.	BOOL	Input Bit 01
Input Bit 02	The input value for input bit 02.	BOOL	Input Bit 02
Input Bit 03	The input value for input bit 03.	BOOL	Input Bit 03
Input Bit 04	The input value for input bit 04.	BOOL	Input Bit 04
Input Bit 05	The input value for input bit 05.	BOOL	Input Bit 05
Input Bit 06	The input value for input bit 06.	BOOL	Input Bit 06
Input Bit 07	The input value for input bit 07.	BOOL	Input Bit 07
Input Bit 08	The input value for input bit 08.	BOOL	Input Bit 08
Input Bit 09	The input value for input bit 09.	BOOL	Input Bit 09
Input Bit 10	The input value for input bit 10.	BOOL	Input Bit 10
Input Bit 11	The input value for input bit 11.	BOOL	Input Bit 11
Input Bit 12	The input value for input bit 12.	BOOL	Input Bit 12
Input Bit 13	The input value for input bit 13.	BOOL	Input Bit 13
Input Bit 14	The input value for input bit 14.	BOOL	Input Bit 14
Input Bit 15	The input value for input bit 15.	BOOL	Input Bit 15
Input Bit 16	The input value for input bit 16.	BOOL	Input Bit 16
Input Bit 17	The input value for input bit 17.	BOOL	Input Bit 17
Input Bit 18	The input value for input bit 18.	BOOL	Input Bit 18
Input Bit 19	The input value for input bit 19.	BOOL	Input Bit 19
Input Bit 20	The input value for input bit 20.	BOOL	Input Bit 20
Input Bit 21	The input value for input bit 21.	BOOL	Input Bit 21
Input Bit 22	The input value for input bit 22.	BOOL	Input Bit 22
Input Bit 23	The input value for input bit 23.	BOOL	Input Bit 23

● General Output

Data name	Function	Data type	I/O port name
Output Bit 00	The output set value for output bit 00.	BOOL	Output Bit 00
Output Bit 01	The output set value for output bit 01.	BOOL	Output Bit 01
Output Bit 02	The output set value for output bit 02.	BOOL	Output Bit 02
Output Bit 03	The output set value for output bit 03.	BOOL	Output Bit 03
Output Bit 04	The output set value for output bit 04.	BOOL	Output Bit 04
Output Bit 05	The output set value for output bit 05.	BOOL	Output Bit 05
Output Bit 06	The output set value for output bit 06.	BOOL	Output Bit 06
Output Bit 07	The output set value for output bit 07.	BOOL	Output Bit 07
Output Bit 08	The output set value for output bit 08.	BOOL	Output Bit 08
Output Bit 09	The output set value for output bit 09.	BOOL	Output Bit 09
Output Bit 10	The output set value for output bit 10.	BOOL	Output Bit 10
Output Bit 11	The output set value for output bit 11.	BOOL	Output Bit 11
Output Bit 12	The output set value for output bit 12.	BOOL	Output Bit 12
Output Bit 13	The output set value for output bit 13.	BOOL	Output Bit 13
Output Bit 14	The output set value for output bit 14.	BOOL	Output Bit 14
Output Bit 15	The output set value for output bit 15.	BOOL	Output Bit 15

2-3 Built-in I/O Functions

The following functions are available for the built-in I/O.

Function name	Description	Reference
Input filter	This function eliminates the chattering or the noises from input signals. It is used to prevent the error reading due to the noises. You can set the filter time constant.	2-5-1 <i>Input Filter</i> on page 2-10
Output load rejection setting	This function performs a preset output operation if a watchdog timer error or an error in the major fault level occurs in the CPU Unit.	2-5-2 <i>Output Load Rejection Setting</i> on page 2-12

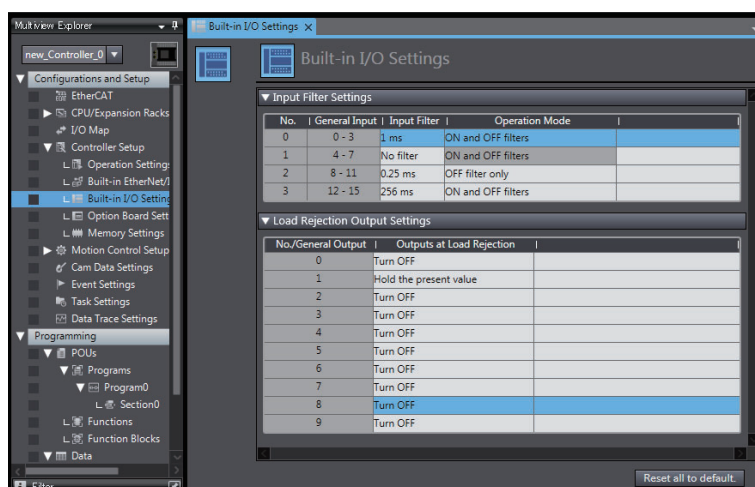
2-4 Settings

The following describes the settings of the built-in I/O.

2-4-1 Built-in I/O Settings

These settings are related to the built-in I/O functions.

Select **Built-in I/O Settings** under **Configurations and Setup - Controller Setup** to display the Built-in I/O Settings Tab Page.



The settings are as follows:

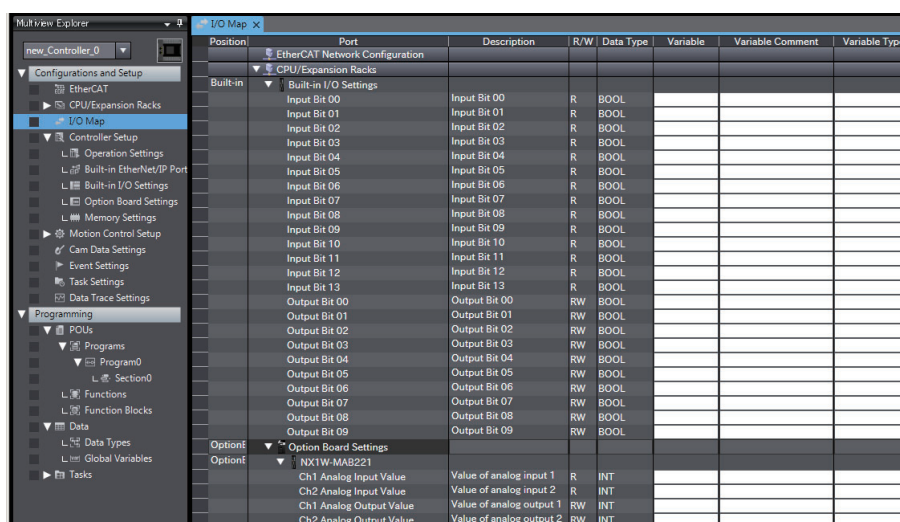
Item	Setting group	Description	Set value	Default	Update timing	Changes in RUN mode
Input Filter Settings	Input Filter	Set the filter time for input signals.	No filter 0.25 ms 0.5 ms 1 ms 2 ms 4 ms 8 ms 16 ms 32 ms 64 ms 128 ms 256 ms	1 ms ^{*1}	When downloaded to CPU Unit	Not allowed.
	Operation Mode ^{*2}	Set the operating mode for the input filter.	- ON and OFF filters - OFF filter only	ON and OFF filters	When downloaded to CPU Unit	Not allowed.
Load Rejection Output Settings		Set the output at load rejection.	Turn OFF Hold the present value	Turn OFF	When downloaded to CPU Unit	Not allowed.

- *1. If a value is set for the input filter time that is smaller than the default value, incorrect input caused by external noises occurs more easily. If an incorrect input occurs, either change the setting to make a long input filter time or take counter-measures, such as separate the Unit or signal lines and noise source, or protect the Unit or signal lines.
- *2. You cannot edit this setting when the **Input Filter** is set to **No filter**.

2-4-2 I/O Map

To use I/O data in the user program, you assign a device variable to each I/O port.

Select **Configurations and Setup - I/O Map** to display the I/O Map.



Position	Port	Description	R/W	Data Type	Variable	Variable Comment	Variable Type
Built-in	Built-in I/O Settings						
	Input Bit 00	Input Bit 00	R	BOOL			
	Input Bit 01	Input Bit 01	R	BOOL			
	Input Bit 02	Input Bit 02	R	BOOL			
	Input Bit 03	Input Bit 03	R	BOOL			
	Input Bit 04	Input Bit 04	R	BOOL			
	Input Bit 05	Input Bit 05	R	BOOL			
	Input Bit 06	Input Bit 06	R	BOOL			
	Input Bit 07	Input Bit 07	R	BOOL			
	Input Bit 08	Input Bit 08	R	BOOL			
	Input Bit 09	Input Bit 09	R	BOOL			
	Input Bit 10	Input Bit 10	R	BOOL			
	Input Bit 11	Input Bit 11	R	BOOL			
	Input Bit 12	Input Bit 12	R	BOOL			
	Input Bit 13	Input Bit 13	R	BOOL			
	Output Bit 00	Output Bit 00	RW	BOOL			
	Output Bit 01	Output Bit 01	RW	BOOL			
	Output Bit 02	Output Bit 02	RW	BOOL			
	Output Bit 03	Output Bit 03	RW	BOOL			
	Output Bit 04	Output Bit 04	RW	BOOL			
	Output Bit 05	Output Bit 05	RW	BOOL			
	Output Bit 06	Output Bit 06	RW	BOOL			
	Output Bit 07	Output Bit 07	RW	BOOL			
	Output Bit 08	Output Bit 08	RW	BOOL			
	Output Bit 09	Output Bit 09	RW	BOOL			
Option	Option Board Settings						
Option	NX1W-MAB221						
	Ch1 Analog Input Value	Value of analog input 1	R	INT			
	Ch2 Analog Input Value	Value of analog input 2	R	INT			
	Ch1 Analog Output Value	Value of analog output 1	RW	INT			
	Ch2 Analog Output Value	Value of analog output 2	RW	INT			

Refer to the *Sysmac Studio Version 1 Operation Manual* (Cat. No. W504) for how to register device variables with the Sysmac Studio.

2-5 Functions

The following shows the details on the functions available for the built-in I/O.

2-5-1 Input Filter

Application

This function prevents data changes and unstable data caused by changes of input data and unstable status of input bits due to chattering and noise.

You can also use this function to make the settings to easily read the pulses that ON time is short.

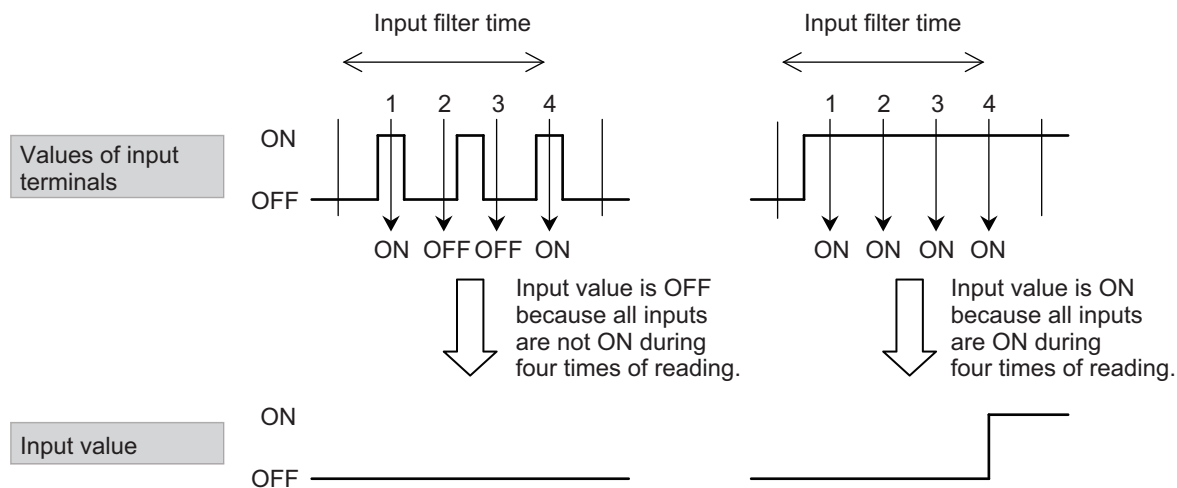
Details on the Function

- If the Operation Mode in the Input Filter Settings is **ON and OFF filters**

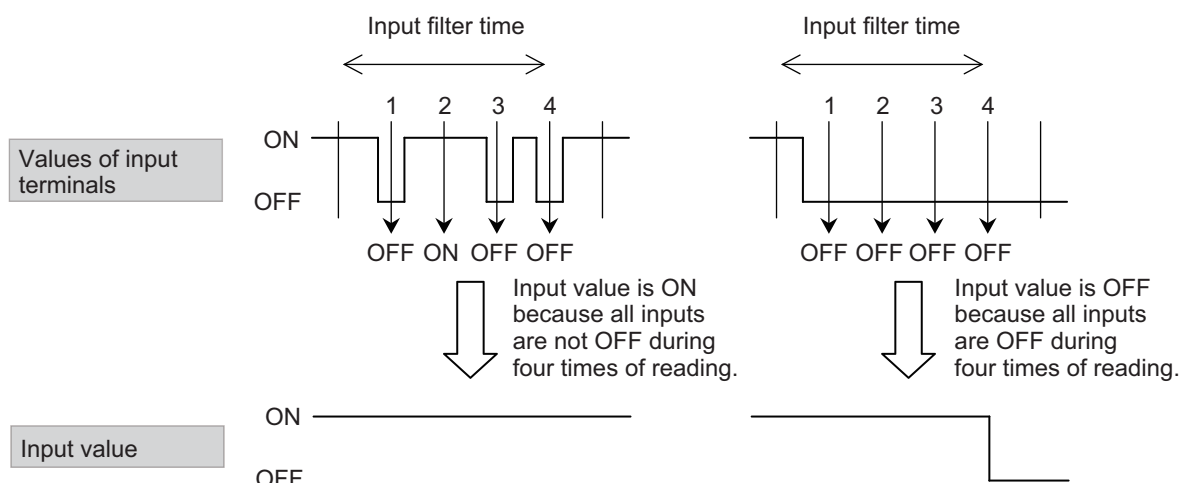
Read the inputs 4 times at a $1/4$ interval of the input filter time. When all inputs are ON or OFF, the input values turn ON or OFF.

This prevents data changes and unstable data.

Operation when the input turns from OFF to ON (ON filter)



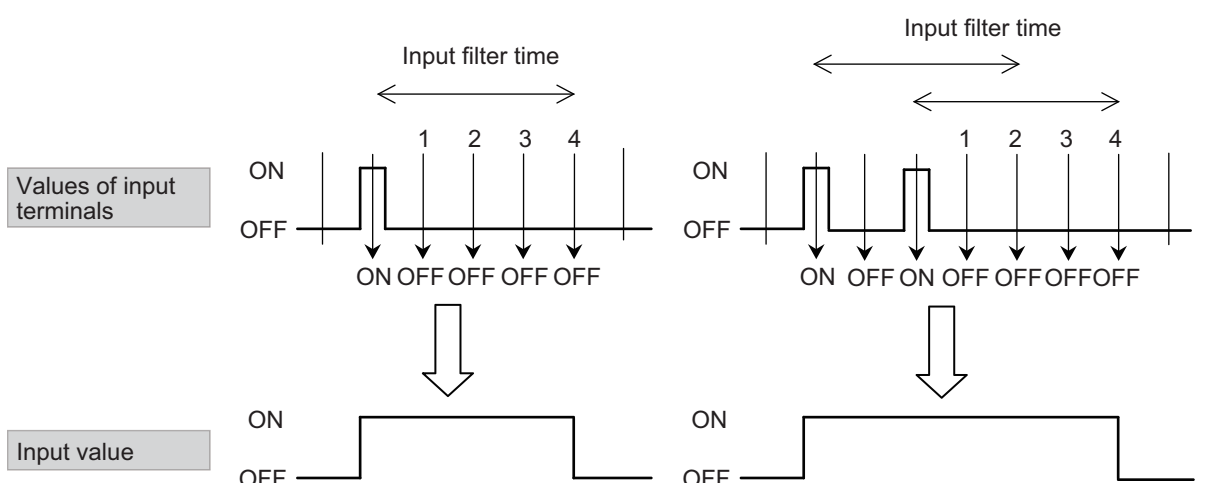
Operation when the input turns from ON to OFF (OFF filter)



● If the Operation Mode in the Input Filter Settings is **OFF filter only**

ON filter is disabled and OFF filter is enabled.

This makes easily to read the pulses that ON time is short.



When input is ON (ON filter disabled)
If the status of input terminals turns ON, the input value will turn ON immediately.

When input is OFF (OFF filter enabled)
If the status of input terminals does not turn ON again during the input filter time, the input value will turn OFF after the input filter time has passed.

When input is ON (ON filter disabled)
If the status of input terminals turns ON, the input value will turn ON immediately.

When input is OFF (OFF filter enabled)
If the status of input terminals turns ON again during the input filter time, the input value stays ON from that time during the input filter time.



Additional Information

If a value is set for the input filter time that is smaller than the default value, incorrect input caused by external noises occurs more easily. If an incorrect input occurs, either change the setting to make a long input filter time or take countermeasures, such as separate the Unit or signal lines and noise source, or protect the Unit or signal lines.

Settings

Configure the settings in the Built-in I/O Settings Tab Page, which is displayed by selecting **Built-in I/O Settings** under **Configurations and Setup - Controller Setup**.

Each setting corresponds to 4 consecutive inputs.

No.	General Input	Input Filter	Operation Mode
0	0 - 3	1 ms	ON and OFF filters
1	4 - 7	No filter	ON and OFF filters
2	8 - 11	0.25 ms	OFF filter only
3	12 - 15	256 ms	ON and OFF filters

Precautions

Note that when you use this function, the time for which the input value actually turns ON or turns OFF is delayed from the initial input to the input terminals until delay time in the following table.

Delay time	Description
ON delay time	ON response time + Input filter time
OFF delay time	OFF response time + Input filter time

2-5-2 Output Load Rejection Setting

Application

This function maintains a safe output status by performing a preset output operation if a watchdog timer error or an error in the major fault level occurs in the CPU Unit.

Details on the Function

This function performs a preset output operation if a watchdog timer error or an error in the major fault level occurs in the CPU Unit.

Set whether to hold the output or turn it OFF if an error occurs.

Settings

Configure the settings in the Built-in I/O Settings Tab Page, which is displayed by selecting **Built-in I/O Settings** under **Configurations and Setup - Controller Setup**.

Each setting corresponds to 1 output.

No./General Output	Outputs at Load Rejection
0	Turn OFF
1	Hold the present value
2	Turn OFF
3	Turn OFF
4	Turn OFF
5	Turn OFF
6	Turn OFF
7	Turn OFF
8	Turn OFF
9	Turn OFF

2-6 I/O Refreshing

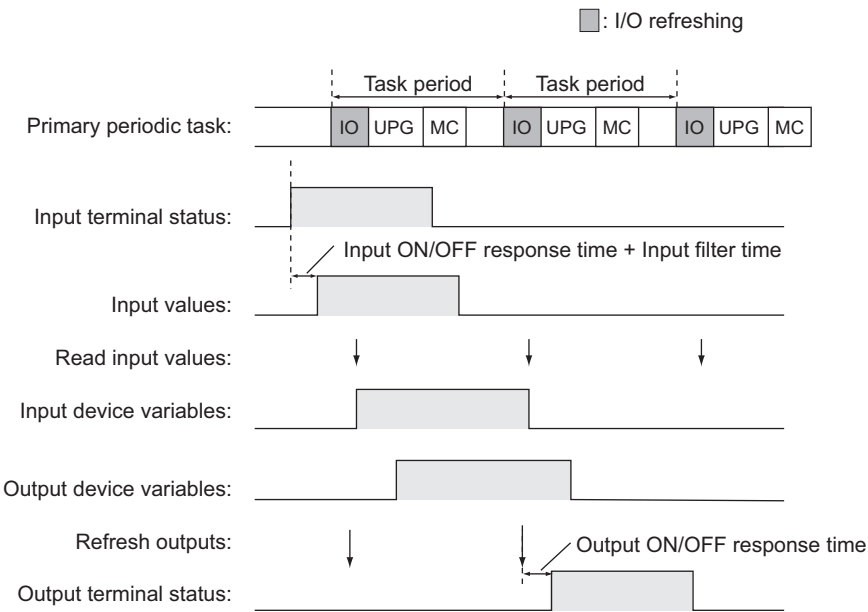
The following describes the I/O refresh timing and I/O response time of the built-in I/O.

2-6-1 I/O Refresh Timing of Built-in I/O

The CPU Unit refreshes the built-in I/O in the task period of the primary periodic task.

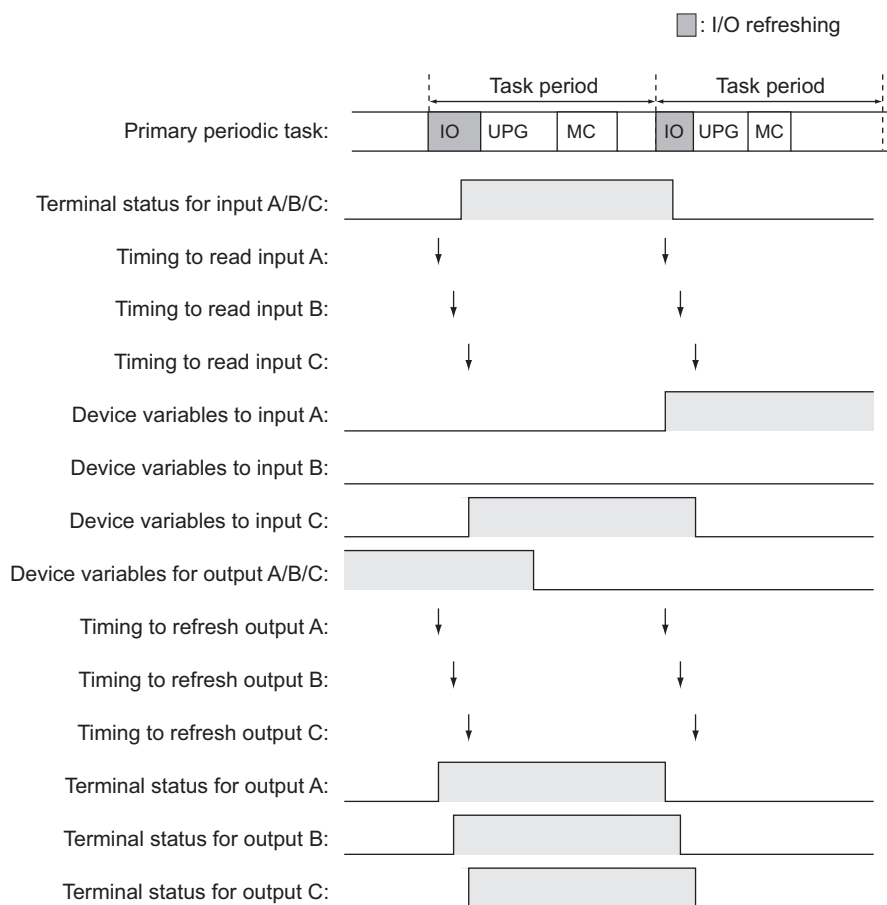
The CPU Unit reads input values into device variables at the time of I/O refreshing.

The CPU Unit updates the outputs with the values of the device variables read at the time of I/O refreshing.



The CPU Unit refreshes outputs and reads inputs for each built-in I/O bit during the period of I/O refreshing in the task period.

Therefore, the timing to read input values varies between input terminals and the timing to refresh outputs varies between output terminals.



Additional Information

If the offset in the timing to read inputs or refresh outputs between terminals is a problem, use a Unit that supports synchronous I/O refreshing. By using a Unit that supports synchronous I/O refreshing, you can synchronize the timing to read inputs and refresh outputs between terminals and Units.

Some NX Units connected to a CPU Unit or EtherCAT Slave Terminal and Units other than NX Units support synchronous I/O refreshing.

Refer to the *NJ/NX-series CPU Unit Software User's Manual* (Cat. No. W501) or the *NX-series EtherCAT Coupler Unit User's Manual* (Cat. No. W519) for synchronous I/O refreshing.

2-6-2 I/O Response Time of Built-in I/O

I/O response time refers to the time from when the input status of inputs is changed until the CPU Unit completes updating the output status of outputs with the execution results of the user program.

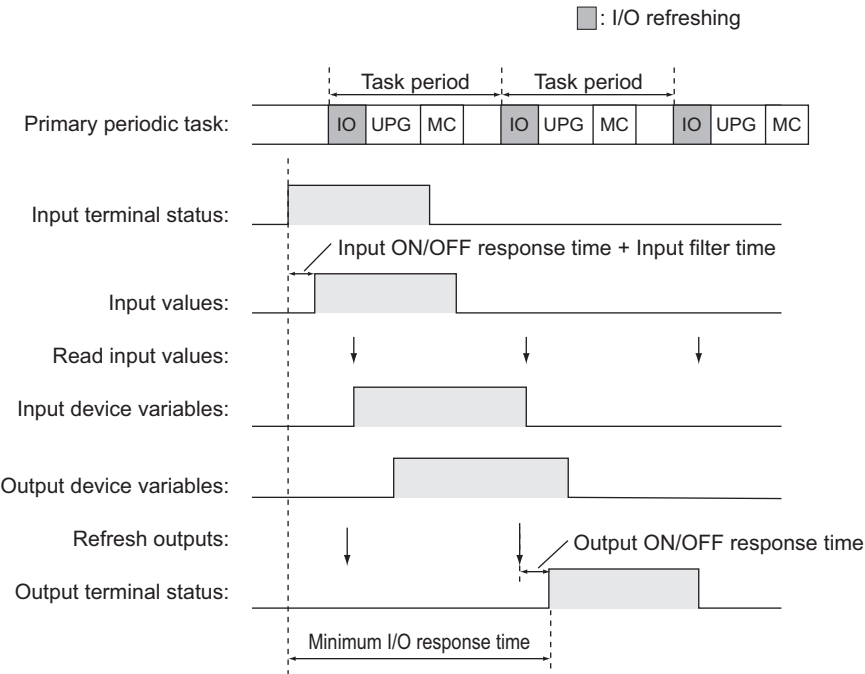
This time varies depending on the timing at which input values change with respect to the task period.

● Minimum I/O Response Time

The I/O response time is minimum when the input values change immediately before the CPU Unit executes I/O refreshing.

The response time at this time can be calculated as follows.

Minimum I/O response time = Input ON/OFF response time + Input filter time + Task period + Output ON/OFF response time

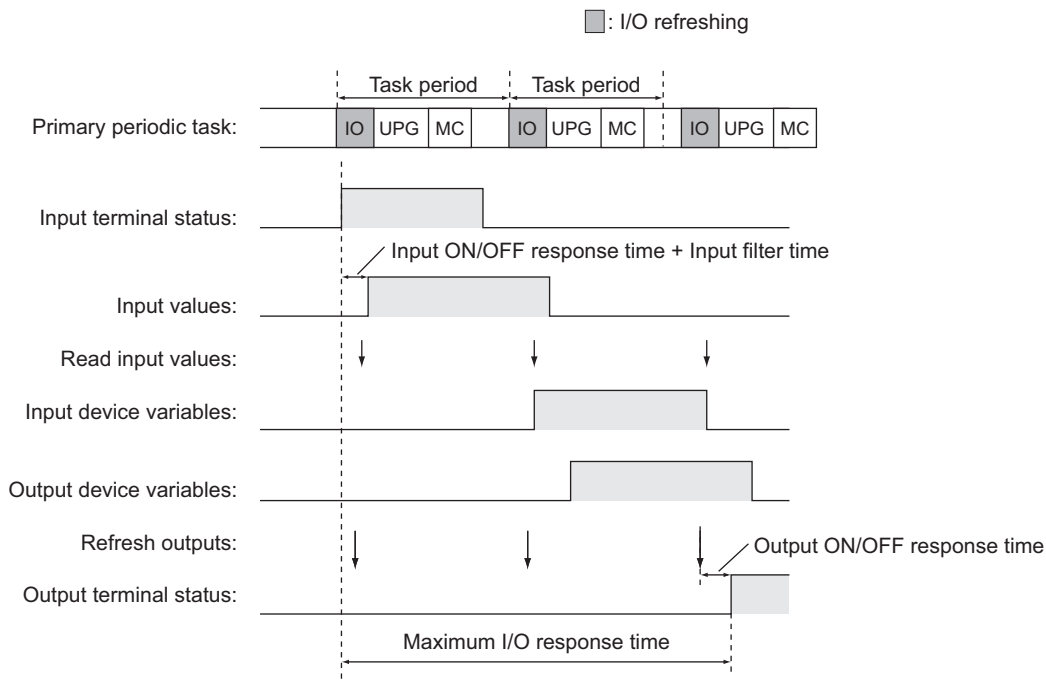


● Maximum I/O Response Time

The I/O response time is maximum when the input values change immediately after the CPU Unit executes I/O refreshing.

The response time at this time can be calculated as follows.

Maximum I/O response time = Input ON/OFF response time + Input filter time + Task period x 3 + Output ON/OFF response time



Additional Information

The input ON/OFF response time and the output ON/OFF response time vary depending on the terminal that is used.

Refer to the *NX-series NX1P2 CPU Unit Hardware User's Manual* (Cat. No. W578) for the specifications of each terminal.

3

Option Boards

This section describes the common functions of Option Boards for the NX1P2 CPU Units.

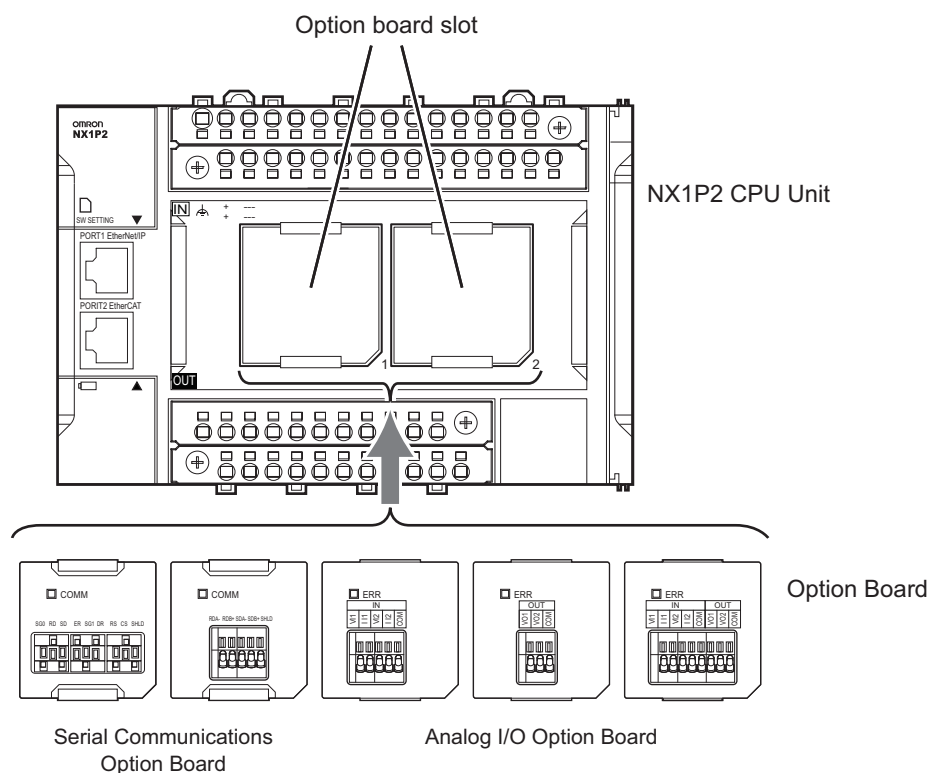
3-1	Option Board Types	3-2
3-1-1	Serial Communications Option Boards	3-3
3-1-2	Analog I/O Option Boards	3-3
3-2	Using Option Boards	3-4
3-2-1	Settings	3-4
3-2-2	System-defined Variables	3-9
3-2-3	Device Variables	3-10
3-2-4	Assigning Device Variables to Option Boards	3-11
3-2-5	Instructions Used for Option Boards	3-13
3-2-6	How Option Boards Operate in Case of an Error	3-14

3-1 Option Board Types

The following describes the types of Option Boards that can be used with the NX1P2 CPU Unit.

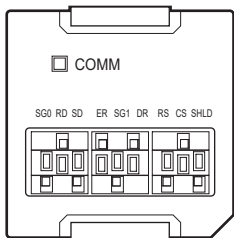
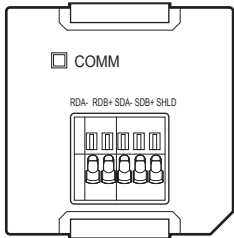
To use an Option Board, mount it to the option board slot on the NX1P2 CPU Unit.

Two types of Option Boards are available: Serial Communications Option Boards and Analog I/O Option Boards.



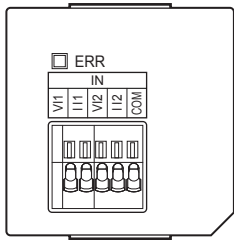
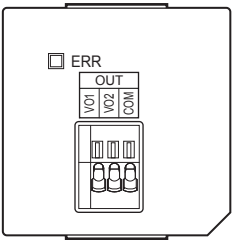
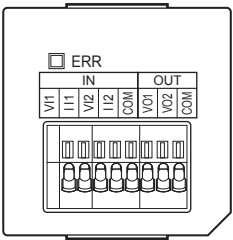
3-1-1 Serial Communications Option Boards

The following table shows the types and summary specifications of Serial Communications Option Boards.

Item	NX1W-CIF01	NX1W-CIF11	NX1W-CIF12
Appearance			
Communications port	RS-232C	RS-422A/485	RS-422A/485
Number of ports	1	1	1
Communications protocol	Host link (FINS), Modbus-RTU master, and no-protocol	Host link (FINS), Modbus-RTU master, and no-protocol	Host link (FINS), Modbus-RTU master, and no-protocol
Isolation	No-isolation	No-isolation	Isolation
External connection terminal	Screwless clamping terminal block	Screwless clamping terminal block	Screwless clamping terminal block

3-1-2 Analog I/O Option Boards

The following table shows the types and summary specifications of Analog I/O Option Boards.

Item	NX1W-ADB21	NX1W-DAB21V	NX1W-MAB221
Appearance			
Analog input	2 inputs	None	2 inputs
Input range	0 to 10 V, 0 to 20 mA	---	0 to 10 V, 0 to 20 mA
Resolution	1/4,000, 1/2,000	---	1/4,000, 1/2,000
Analog output	None	2 outputs	2 outputs
Output range	---	0 to 10 V	0 to 10 V
Resolution	---	1/4,000	1/4,000
Conversion time	4 ms/Option Board	4 ms/Option Board	6 ms/Option Board
Isolation	No-isolation	No-isolation	No-isolation
External connection terminal	Screwless clamping terminal block	Screwless clamping terminal block	Screwless clamping terminal block

3-2 Using Option Boards

The following provides information on using Option Boards, which is commonly applicable to Serial Communications Option Boards and Analog I/O Option Boards.

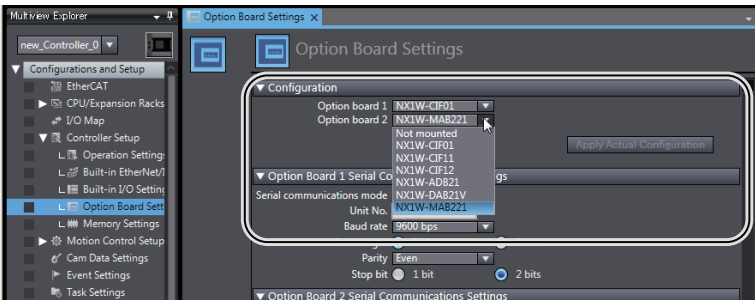
3-2-1 Settings

The description below is related to the settings of Option Boards to use.

Configuration

Specify the models of the Option Boards to use.

Set the Option Board configuration under **Configuration** in the Option Board Settings Tab Page, which is displayed by selecting **Option Board Settings** under **Configurations and Setup - Controller Setup**.



The settings are as follows:

Item	Setting group	Description	Set value	Default	Update timing	Changes in RUN mode
Configuration	Option board 1	Specify the model of the Option Board to use.	Not mounted	Not mounted	When downloaded to CPU Unit	Not allowed.
	Option board 2*1		NX1W-CIF01 NX1W-CIF11 NX1W-CIF12 NX1W-ADB21 NX1W-DAB21V NX1W-MAB221			

*1. You cannot edit this setting if your CPU Unit does not support the second Option Board.



Additional Information

If you change the set Option Board configuration, the I/O Map will be changed automatically. At the same time, the mapping of device variables to the I/O Map will also be deleted.

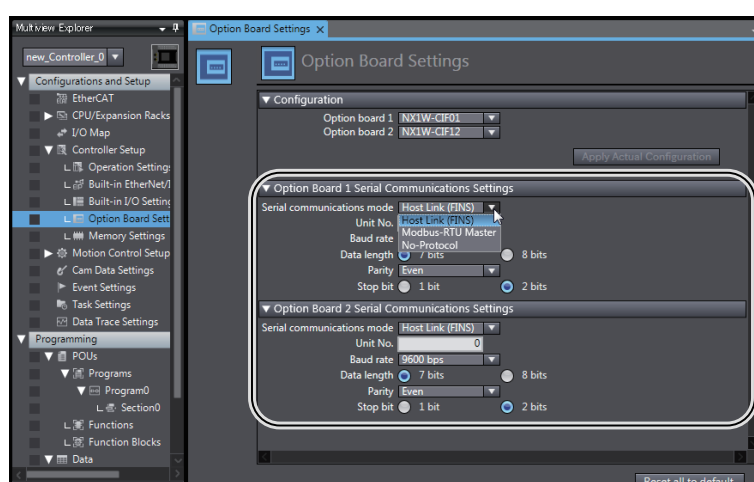
After you change the Option Board configuration, you need to map the device variables again.

Option Board Serial Communications Settings

These are the serial port settings for the Serial Communications Option Boards.

The serial communications settings for each Option Board are enabled only when the corresponding Serial Communications Option Board is specified in the configuration setting.

Set the Option Board Serial Communications Settings under **Option Board 1 Serial Communications Settings/Option Board 2 Serial Communications Settings** in the Option Board Settings Tab Page, which is displayed by selecting **Option Board Settings** under **Configurations and Setup - Controller Setup**.



The settings are as follows:

Item	Setting group	Description	Set value	Default	Update timing	Changes in RUN mode
Option Board 1 Serial Communications Settings	Serial communications mode	Set the serial communications mode.	Host Link (FINS) Modbus-RTU Master No-Protocol	Host Link (FINS)	When downloaded to CPU Unit	Not allowed.
	Unit No.	Set the unit number of the host link when the Serial communications mode is set to Host Link (FINS) .	0 to 31	0	When downloaded to CPU Unit	Not allowed.
	Baud rate	Set the baud rate for the serial port.	1,200 bps 2,400 bps 4,800 bps 9,600 bps 19,200 bps 38,400 bps 57,600 bps 115,200 bps	9,600 bps	When downloaded to CPU Unit	Not allowed.
	Data length	Set the data length.	7 bits 8 bits	7 bits	When downloaded to CPU Unit	Not allowed.
	Parity	Set the parity bit.	Even Odd None	Even	When downloaded to CPU Unit	Not allowed.
	Stop bit	Set the stop bit.	1 bit 2 bits	2 bits	When downloaded to CPU Unit	Not allowed.
Option Board 2 Serial Communications Settings	The settings are the same as those for the Option Board 1 Serial Communications Settings. However, you cannot edit this setting if your CPU Unit does not support the second Option Board.					

Memory Settings for CJ-series Units

These settings are provided to specify the area type and size of the memory used for CJ-series Units when the **Serial communications mode** is set to **Host Link (FINS)**.



Additional Information

The host link (FINS) protocol accesses only the memory used for CJ-series Units out of the entire memory available in the CPU unit.

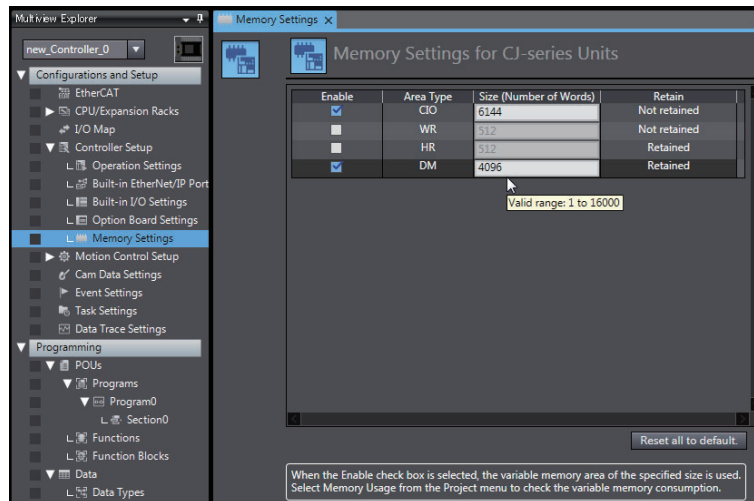
Nevertheless, the NX1P2 CPU Unit does not have memory used for CJ-series Units in the default setting. Therefore, in order to use the host link (FINS) protocol, you need to generate memory used for CJ-series Units in the NX1P2 CPU Unit. The memory settings for CJ-series Units are used for this purpose.

● Setting Procedure

No.	Step	Description	Reference
1	Determining the usable memory	Determine the area type and the number of words of memory used for CJ-series Units to make available for data exchange with connected devices.	Manuals and technical materials for connected devices
2	Memory settings for CJ-series Units	In the Sysmac Studio, set the area type and the number of words of memory used for CJ-series Units to make available for connected devices.	<i>Setting Screen</i> on page 3-7
3	Programming	Create the user program that uses the memory used for CJ-series Units. If the set memory size is not sufficient, return to step 2 and increase the number of words.	<i>4-2-4 Programming</i> on page 4-8 <i>4-3-4 Programming</i> on page 4-17
4	Downloading the project	Download the project from the Sysmac Studio.	<i>NJ/NX-series CPU Unit Software User's Manual</i> (Cat. No. W501)
5	Checking operation and actual operation	Check the operation of the user program and connected devices.	<i>Sysmac Studio Version 1 Operation Manual</i> (Cat. No. W504)

● Setting Screen

Specify the memory used for CJ-series Units in the Memory Settings for CJ-series Units Tab Page, which is displayed by selecting **Memory Settings** under **Configurations and Setup - Controller Setup**.



● Settings

The settings are as follows:

Item	Setting group	Setting	Description	Set value	Default	Update timing	Changes in RUN mode
Memory Settings for CJ-series Units	CIO	Enable	Enable or disable the generation of CIO area type memory used for CJ-series Units.	Enable Disable	Disable	When down-loaded to CPU Unit	Not allowed.
		Size (Number of Words)	Specify the size of memory of area type CIO.	1 to 6,144	6,144	When down-loaded to CPU Unit	Not allowed.
	WR	Enable	Enable or disable the generation of WR area type memory used for CJ-series Units.	Enable Disable	Disable	When down-loaded to CPU Unit	Not allowed.
		Size (Number of Words)	Specify the size of memory of area type WR.	1 to 512	512	When down-loaded to CPU Unit	Not allowed.
	HR	Enable	Enable or disable the generation of HR area type memory used for CJ-series Units.	Enable Disable	Disable	When down-loaded to CPU Unit	Not allowed.
		Size (Number of Words)	Specify the size of memory of area type HR.	1 to 1,536	512	When down-loaded to CPU Unit	Not allowed.
	DM	Enable	Enable or disable the generation of DM area type memory used for CJ-series Units.	Enable Disable	Disable	When down-loaded to CPU Unit	Not allowed.
		Size (Number of Words)	Specify the size of memory of area type DM.	1 to 16,000	4,096	When down-loaded to CPU Unit	Not allowed.

Refer to the *NJ/NX-series CPU Unit Software User's Manual* (Cat. No. W501) for the specifications of memory used for CJ-series Units.

3-2-2 System-defined Variables

The following table shows the system-defined variables available for Option Boards.

Refer to the *NJ/NX-series CPU Unit Software User's Manual* (Cat. No. W501) for details on the specifications of system-defined variables for Option Boards.

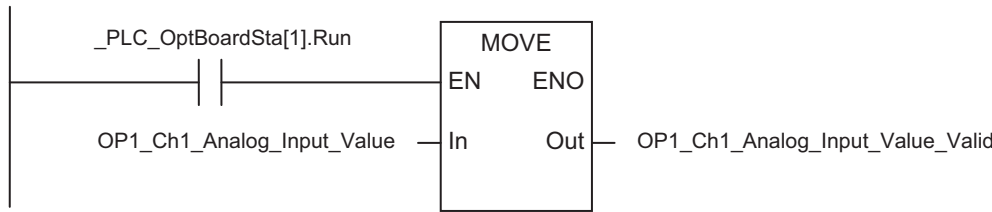
Variable	Meaning	Function	Data type
_PLC_OptBoardSta	Option Board Status	Contains the status of Option Boards. This variable is commonly used regardless of the models of Option Boards. The array element 1 corresponds to the option board slot 1 and array element 2 corresponds to the option board slot 2.	ARRAY[1..2] OF _sOPTBOARD_STA
_PLC_OptSerialErrSta	Serial Option Board Error Status	Contains the error status of an transmission error for the Serial Communications Option Board. When the Serial communications mode of an Serial Communications Option Board is only set to Host Link (FINS) , the value of each member is updated. Other than the above setting, the values of all members are FALSE. The array element 1 corresponds to the option board slot 1 and array element 2 corresponds to the option board slot 2. You cannot use this system-defined variable in the user program. This variable is used only for troubleshooting the serial communications device connection in the Sysmac Studio.	ARRAY[1..2] OF _sOPTSERIALERR_STA

The operation of the *_PLC_OptBoardSta* (Option Board Status) system-defined variable members is shown below.

Status	isDetect	Run	Error
Option Board is not mounted.	FALSE	FALSE	FALSE
Option Board is being initialized.	TRUE	FALSE	FALSE
Option Board is operating normally.	TRUE	TRUE	FALSE
Option Board settings are being changed.	TRUE	FALSE	FALSE
An Option Board event occurred when Option Board is mounted.	TRUE	FALSE	TRUE
An Option Board event occurred when Option Board is not mounted.	FALSE	FALSE	TRUE
Option Board was removed.	FALSE	FALSE	TRUE

To use device variables or communications instructions for an Option Board, program the *.Run* (Option Board Normal Operation) member of the *_PLC_OptBoardSta* (Option Board Status) system-defined variable as an interlock condition in the user program.

Example of reading analog input values from Option Board 1 to the CPU Unit using the Option Board Normal Operation as an interlock condition

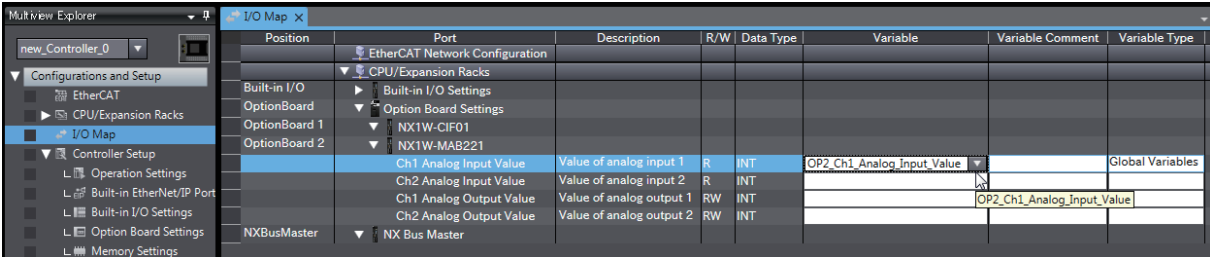


- `_PLC_OptBoardSta[1].Run` Option Board Normal Operation of Option Board 1 Status system-defined variable
- `OP1_Ch1_Analog_Input_Value` Device variable to analog input 1 on Option Board 1
- `OP1_Ch1_Analog_Input_Value_Valid` Input value to analog input 1 read into the CPU Unit

3-2-3 Device Variables

To use I/O data for an Option Board in the user program, you assign a device variable to each I/O port. Specify device variables in the I/O Map Tab Page, which is displayed by selecting **Configurations and Setup - I/O Map**.

The following is an example of Option Board displayed on the I/O Map Tab Page.



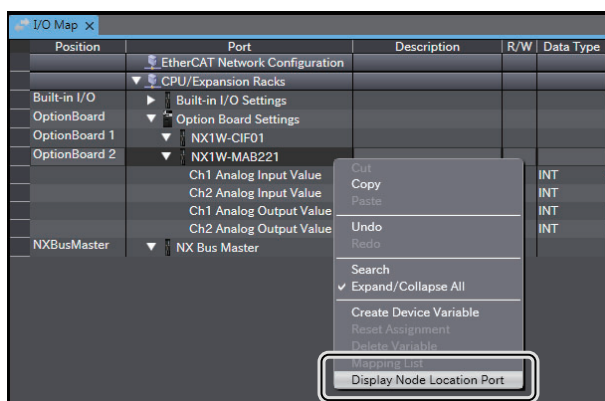
3-2-4 Assigning Device Variables to Option Boards

Some instructions used for Option Boards require that the Option Board be specified in the form of variables. Therefore, you need to assign variables to the Option Boards in advance.

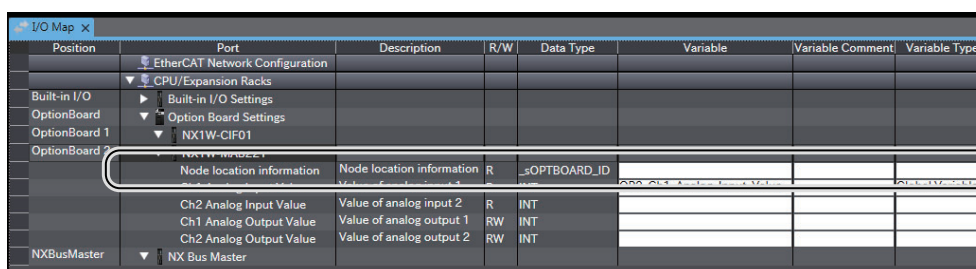
Assignment Procedure

The Sysmac Studio does not automatically create variables that are assigned to Option Boards even if you specify the Option Board configuration. Follow the steps below to configure the settings to assign variables to the Option Boards.

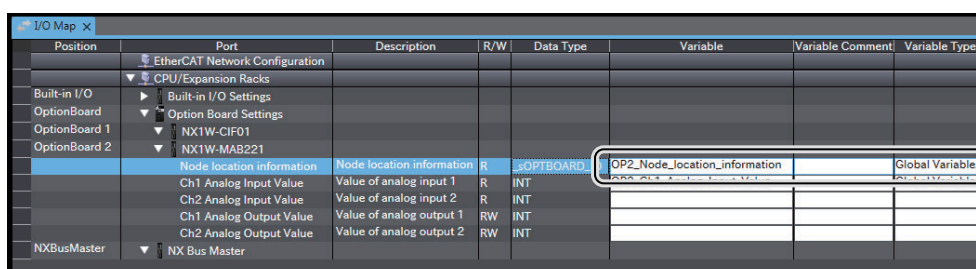
- 1 In the Sysmac Studio, set the Option Board configuration under **Configuration** in the Option Board Settings Tab Page, which is displayed by selecting **Option Board Settings** under **Configurations and Setup - Controller Setup**.
- 2 Select **Configurations and Setup - I/O Map** to display the I/O Map Tab Page.
- 3 Right-click the model of Option Board to which you want to assign variables and select **Display Node Location Port** from the menu.



The *Node location information* port is added on the I/O Map.



- 4 Right-click *Node location information* and select **Create Device Variable** from the menu. The variable name is written to the **Variable** Field of the *Node location information* port.



The data type of variables assigned to Option Boards is `_sOPTBOARD_ID` structure. The details on the `_sOPTBOARD_ID` structure data type are given in the following table.

Variable	Name	Meaning	Data type
User specified			<code>_sOPTBOARD_ID</code>
SlotNo	Slot number	Slot number of the Option Board	UINT
IPAdr	IP address	Not used.	BYTE[5]



Precautions for Correct Use

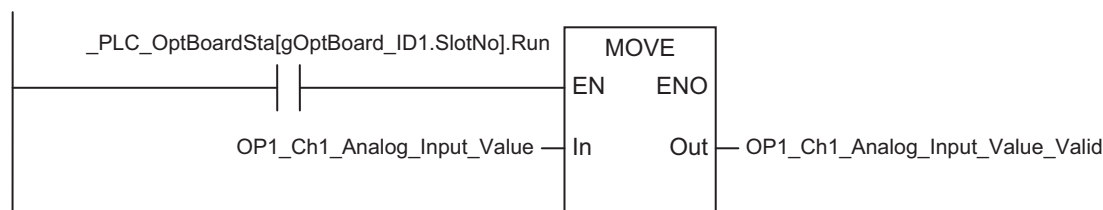
The values of variables that assigned to the Option Boards will be set automatically when you register the variables. Do not change the values of the variables. If you change the value of a variable, the Controller may not perform the intended operation.

Using Variables Assigned to Option Boards

You use variables assigned to an Option Board when you specify the Option Board in the user program.

For this purpose, you need to register the variables in the variable table in advance, using the same names as those of the variables assigned to the Option Board on the I/O Map. The data type of the variables is `_sOPTBOARD_ID` structure.

Example of reading analog input values to the CPU Unit using the Option Board Normal Operation at slot position indicated by the variables assigned to Option Board as an interlock condition



`gOptBoard_ID1.SlotNo`

`_PLC_OptBoardSta[]`.Run

`OP1_Ch1_Analog_Input_Value`

`OP1_Ch1_Analog_Input_Value_Valid`

Slot position indicated by the variables assigned to Option Board

Option Board Normal Operation at the position indicated by `_gOptBoard_ID1`

Device variable to analog input 1 on Option Board 1

Input value to analog input 1 read into the CPU Unit

3-2-5 Instructions Used for Option Boards

The table below shows serial communications instructions that you can use when the **Serial communications mode** of a Serial Option Board is **Modbus-RTU Master** or **No-Protocol**.

Instruction	Name	Outline of function
NX_SerialSend	Send No-Protocol Data	Sends data in No-Protocol Mode from a serial port on a CIF Unit or Option Board.
NX_SerialRcv	Receive No-Protocol Data	Reads data in No-Protocol Mode from a serial port on a CIF Unit or Option Board.
NX_ModbusRtuCmd	Send Modbus RTU General Command	Sends general commands from a serial port on a CIF Unit or Option Board to Modbus-RTU slaves using Modbus-RTU protocol.
NX_ModbusRtuRead	Send Modbus RTURead Command	Sends read commands from a serial port on a CIF Unit or Option Board to Modbus-RTU slaves using Modbus-RTU protocol.
NX_ModbusRtuWrite	Send Modbus RTUWrite Command	Sends write commands from a serial port on a CIF Unit or Option Board to Modbus-RTU slaves using Modbus-RTU protocol.
NX_SerialSigCtl	Serial Control Signal ON/OFF Switching	Turns ON or OFF the ER or RS signal of a serial port on a CIF Unit or Option Board.
NX_SerialSigRead	Read Serial Control Signal	Reads the CS or DR signal of a serial port on an Option Board.
NX_SerialStatusRead	Read Serial Port Status	Reads the status of a serial port on an Option Board.
NX_SerialBufClear	Clear Buffer	Clears the send or receive buffer.

Refer to the *NJ/NX-series Instructions Reference Manual* (Cat. No. W502-E1-17 or later) for details on serial communications instructions.

3-2-6 How Option Boards Operate in Case of an Error

The tables below show how Option Boards operate if an error occurs.

● Errors Not Related to Option Boards

Event level	Serial Communications Option Board	Analog I/O Option Board
Major fault level	No change.	Load turned OFF for analog output. Analog input values at error occurrence retained.
Partial fault level		No change.
Minor fault level		
Observation		
Information		

● Errors Related to Option Boards

Event level	Serial Communications Option Board	Analog I/O Option Board
Option Board Configuration Verification Error	Host link function disabled when Host Link (FINS) is selected.	Load turned OFF for analog output. Analog input value reset to 0.
Unsupported Option Board Mounted	Error generated at execution of a serial communications instruction when Host Link (FINS) is not selected.	
Option Board Error		
Analog Option Board Startup Error	---	
Analog Option Board Communications Error	---	

4

Serial Communications

This section describes the functions of Serial Communications Option Boards for the NX1P2 CPU Units.

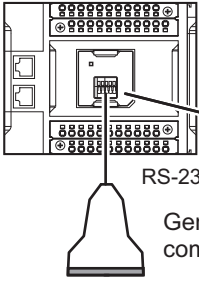
4

4-1	Serial Communications Types and Overview	4-2
4-2	Programless Communications with NB-series Programmable Terminals	4-4
4-2-1	Overview	4-4
4-2-2	Procedure	4-4
4-2-3	Settings	4-6
4-2-4	Programming	4-8
4-2-5	Connection Examples	4-9
4-3	Programless Communications with E5□C Digital Temperature Controllers	4-10
4-3-1	Overview	4-10
4-3-2	Procedure	4-12
4-3-3	Settings	4-14
4-3-4	Programming	4-17
4-3-5	Connection Examples	4-17
4-4	Connection with Modbus-RTU Slaves	4-18
4-4-1	Overview	4-18
4-4-2	Procedure	4-19
4-4-3	Settings	4-21
4-4-4	Programming	4-22
4-4-5	Connection Examples	4-24
4-5	Connection with General-purpose Serial Communications Devices	4-25
4-5-1	Overview	4-25
4-5-2	Procedure	4-26
4-5-3	Settings	4-28
4-5-4	Programming	4-29

4-1 Serial Communications Types and Overview

The following table shows the communications protocols supported by the NX1P2 CPU Unit and examples of connected devices.

Connected device	Serial communications mode*1	Description
NB-series Programmable Terminal RS-232C or RS-422A/485 NX1P2 CPU Unit RS-232C Option Board or RS-422A/485 Option Board	Host Link (FINS)*2	The CPU Unit exchanges data with Programmable Terminals (PTs). For this data exchange, memory used for CJ-series Units is used. No communications program is required on the CPU Unit side.
RS-422A/485 Option Board E5□C Controller No. 0 E5□C Controller No. 1 E5□C Controller No. n 32 max. E5□C Controllers access memory used for CJ-series Units automatically.		The CPU Unit exchanges data with E5□C Digital Temperature Controllers. For this data exchange, memory used for CJ-series Units is used. No communications program is required on the CPU Unit side. You can read and write parameters and run/stop E5□C Temperature Controllers.
NX1P2 CPU Unit RS-232C Option Board or RS-422A/485 Option Board RS-232C or RS-422A/485 Inverter	Modbus-RTU Master*3	The CPU Unit exchanges data with Modbus-RTU slaves. In this data exchange, you use special instructions to send a Modbus-RTU command and receive a response.

Connected device	Serial communications mode ^{*1}	Description
 NX1P2 CPU Unit RS-232C Option Board or RS-422A/485 Option Board RS-232C or RS-422A/485 General-purpose serial communications device	No-Protocol	The CPU Unit exchanges data with general-purpose serial communications devices with RS-232C or RS-422A/485 ports. In this data exchange, instructions to send and receive data from a serial communications port in No-protocol Mode are used. You need to program the communications procedure (protocol) to exchange data with general-purpose devices in the user program.

*1. Select the serial communications mode to use in advance. You cannot change the serial communications mode when the CPU Unit is operating.

*2. Only FINS commands are supported. C-mode commands are not supported.

*3. The function to use NX1P2 CPU Units as Modbus-RTU slaves is not supported.

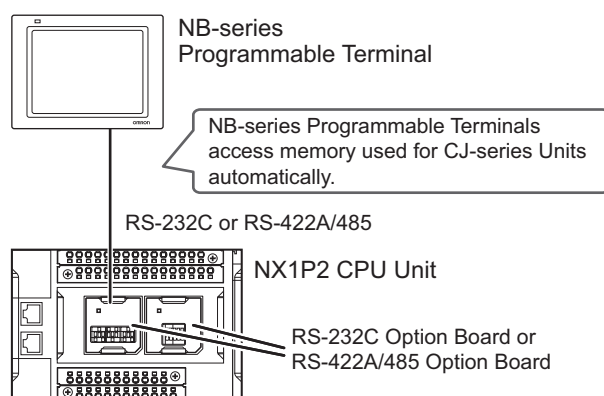
4-2 Programless Communications with NB-series Programmable Terminals

The following describes programless communications with NB-series Programmable Terminals.

4-2-1 Overview

The NX1P2 CPU Unit supports programless communications with NB-series Programmable Terminals (hereafter NB-series Units) using the host link protocol.

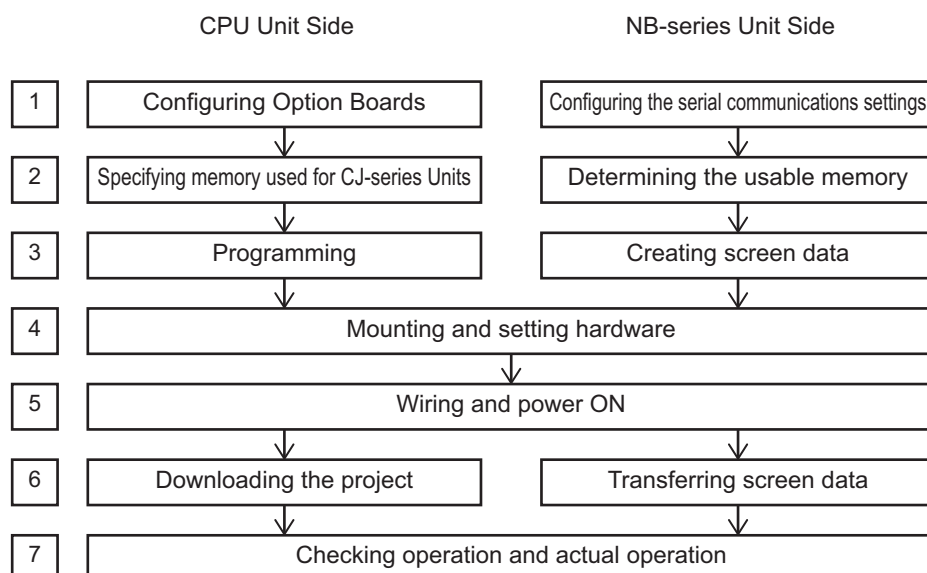
To use this function, you mount a Serial Communications Option Board on the NX1P2 CPU Unit, set its **Serial communications mode** to **Host Link (FINS)** and specify the memory used for CJ-series Units, and connect their serial ports together.



4-2-2 Procedure

The operating procedure is described below.

Overall Procedure



Procedure Details

● CPU Unit Side

No.	Step	Description	Reference
1	Configuring Option Boards	In the Sysmac Studio, specify the Option Board configuration and configure the serial communications mode settings.	4-2-3 <i>Settings</i> on page 4-6 <i>Configuration and Option Board Serial Communications Settings</i> on page 4-6
2	Specifying memory used for CJ-series Units	In the Sysmac Studio, set the area type and the number of words of memory enough to include the memory that will be used for the screen data of the NB-series Unit.	4-2-3 <i>Settings</i> on page 4-6 <i>Memory Settings for CJ-series Units</i> on page 4-6
3	Programming	In the Sysmac Studio, create a program to access the memory used for CJ-series Units by using user-defined variables with AT specifications.	4-2-4 <i>Programming</i> on page 4-8
4	Mounting and setting hardware	If you are using an NX1W-CIF11/CIF12 Option Board, set the operation setting DIP switches on the back. Mount the Option Boards and necessary Units. Install the CPU Unit and NB-series Units.	4-2-5 <i>Connection Examples</i> on page 4-9 <i>NX-series NX1P2 CPU Unit Hardware User's Manual</i> (Cat. No. W578)
5	Wiring and power ON	Connect the serial communications terminals of the Option Boards and NB-series Units. Wire the power supply terminals and turn ON the power supply.	
6	Downloading the project	Download the project from the Sysmac Studio.	<i>NJ/NX-series CPU Unit Software User's Manual</i> (Cat. No. W501)
7	Checking operation and actual operation	Check the operation of the user program and screen data.	<i>Sysmac Studio Version 1 Operation Manual</i> (Cat. No. W504)

● NB-series Unit Side

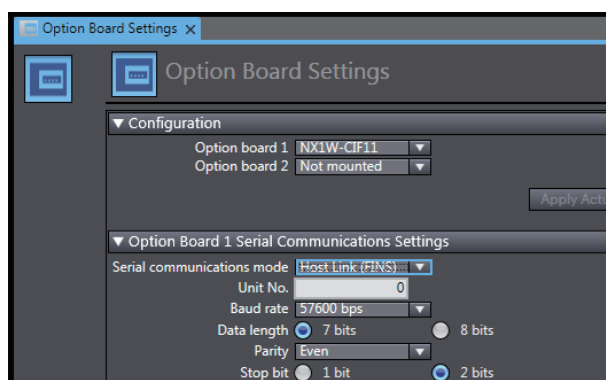
No.	Step	Description	Reference
1	Configuring the serial communications settings	In the NB-Designer, create a project and select COM1 or COM2 in the Configuration and Setup Window .	<i>NB-series Programmable Terminals Setup Manual</i> (Cat. No. V107)
2	Determining the usable memory	Determine the area type and the number of words of memory to make available for screen data.	<i>NB-series Programmable Terminals NB-Designer Operation Manual</i> (Cat. No. V106)
3	Creating screen data	In the NB-Designer, create screen data.	
4	Mounting and setting hardware	Install the CPU Unit and NB-series Units.	
5	Wiring and power ON	Connect the serial communications terminals of the Option Boards and NB-series Units. Wire the power supply terminals and turn ON the power supply.	
6	Transferring screen data	In the NB-Designer, download the project that contains the screen data.	
7	Checking operation and actual operation	Check the operation of the user program and screen data.	

4-2-3 Settings

Settings on the CPU Unit

● Configuration and Option Board Serial Communications Settings

Configure these settings in the Option Board Settings Tab Page, which is displayed by selecting **Option Board Settings** under **Configurations and Setup - Controller Setup**.



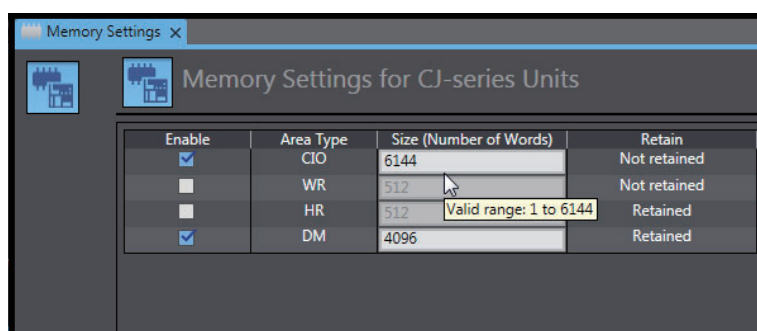
Under **Configuration**, specify the models of the Option Boards to use.

Under **Option Board Settings**, configure the following settings:

Item	Set value
Serial communications mode	Host Link (FINS)
Unit No.	0
Baud rate	Set this to match the setting on the NB-series Unit.
Data length	7 bits
Parity	Even
Stop bit	2 bits

● Memory Settings for CJ-series Units

Specify the memory used for CJ-series Units in the Memory Settings for CJ-series Units Tab Page, which is displayed by selecting **Memory Settings** under **Configurations and Setup - Controller Setup**.



Set the area type and the number of words of memory enough to include the memory that will be used for the screen data to create.



Precautions for Correct Use

Create screen data to avoid using the EM area because the NX1P2 CPU Unit does not support the EM area type.

Settings on the NX1W-CIF11/CIF12 Option Board

The CPU Unit requires an NX1W-CIF11 or NX1W-CIF12 Option Board for connection with external devices via RS-422A/485.

The table below shows the settings of the operation setting DIP switches on the back of the Option Board.

CIF11		CIF12		Setting	Setting description
SW	No.	SW	No.		
SW1	1	SW1	1	ON	With terminating resistance ^{*1}
	2		2	OFF	Four-wire type
	3		3	OFF	Four-wire type
	4		4	OFF	(Not used)
	5	SW2	1	OFF	Without RS control for receive data (Always receive data)
	6		2	OFF	Without RS control for send data (Always send data)

*1. Turn this OFF if the NX1W-CIF11/CIF12 is not the terminating device.

Settings on NB-series Units

In the NB-Designer, create a project and select the serial port to use in the **Configuration and Setup Window**.

The settings are as follows:

Item	Set value
Baud Rate	Set this to match the serial port setting.
Data Bit	7
Parity Check	even
Stop Bit	2

Refer to the NB-series Programmable Terminals NB-Designer Operation Manual (Cat. No. V106) for the detailed settings.

Refer to *4-2-5 Connection Examples* on page 4-9 for information on wiring the serial port.

4-2-4 Programming

Assign the user-defined variables that are used in the user program on the CPU Unit to the memory used for CJ-series Units that will be accessed by the NB-series Unit by using AT specification.

Then, create the user program for communicating with the NB-series Unit.

Refer to the *NJ/NX-series CPU Unit Software User's Manual* (Cat. No. W501) for memory used for CJ-series Units and AT specification.



Additional Information

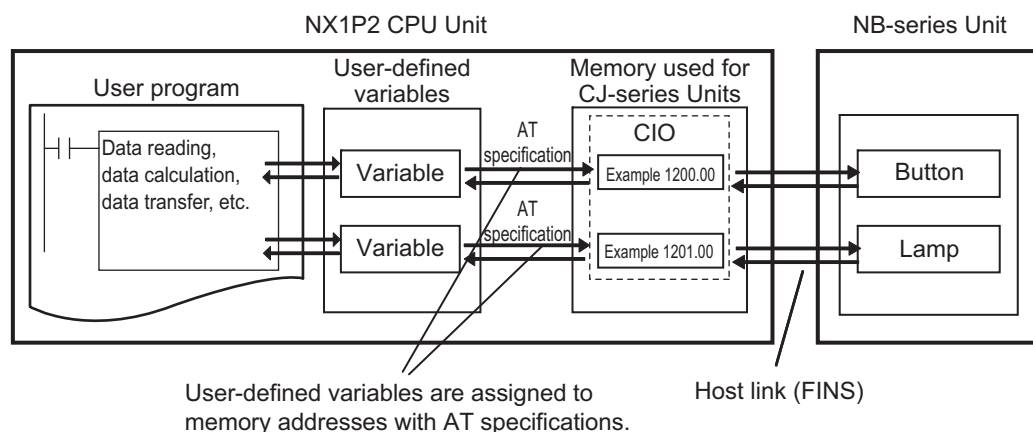
The NX1P2 CPU Unit performs programless communications with NB-series Units using the host link (FINS) protocol.

The host link (FINS) protocol accesses only the memory used for CJ-series Units out of the entire memory available in the CPU unit based on address specification.

In other words, the CPU Unit uses the memory used for CJ-series Units to communicate data with NB-series Units.

On the other hand, the NX1P2 CPU Unit uses variables for all processing tasks, for example, to exchange I/O information with external devices, perform data calculations, and so on.

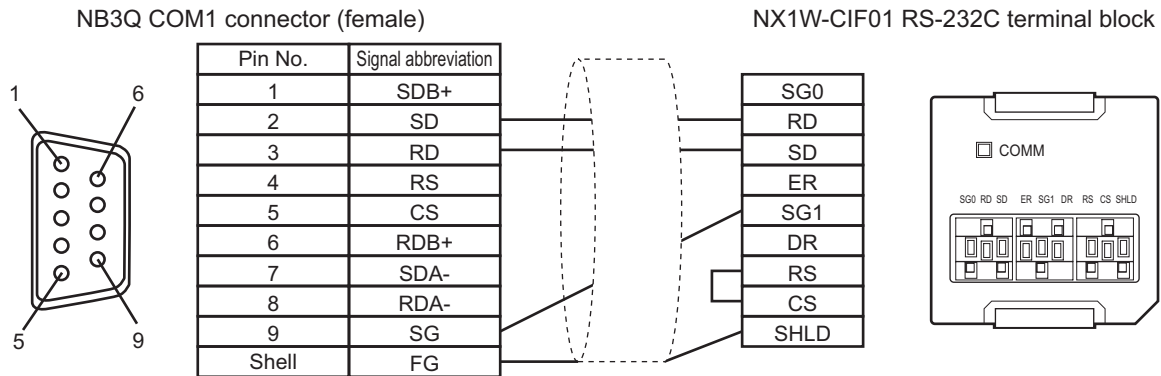
Therefore, to access memory used for CJ-series Units from the user program on the CPU Unit, you need to assign user-defined variables to it. This assignment of user-defined variables to memory used for CJ-series Units is called AT specification.



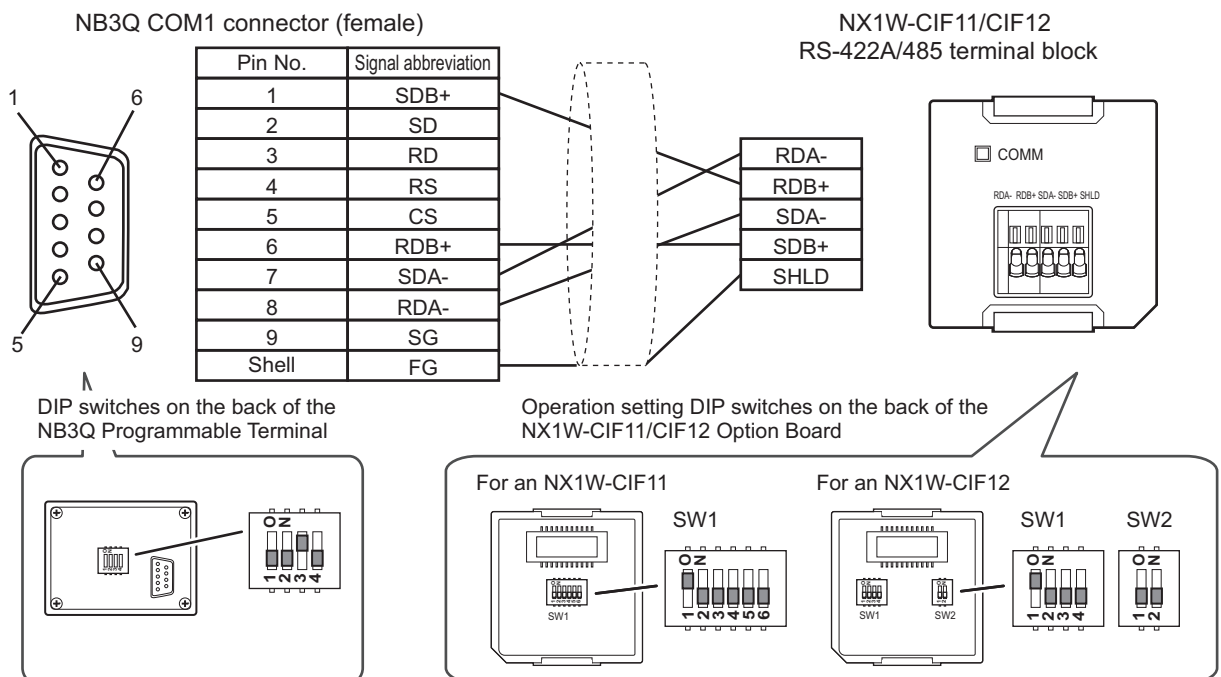
4-2-5 Connection Examples

Examples of connecting the CPU Unit to a serial port on an NB3Q Programmable Terminal are given below.

Connecting the NX1W-CIF01 Option Board to the Serial Port (COM1) on an NB3Q Programmable Terminal (RS-232C)



Connecting the NX1W-CIF11/CIF12 Option Board to the Serial Port (COM1) on an NB3Q Programmable Terminal (RS-422A)



Refer to the *NB-series Programmable Terminals Setup Manual* (Cat. No. V107) for wiring information on connecting the CPU Unit to the serial ports on an NB5Q/NB7W/NB10W Programmable Terminal.

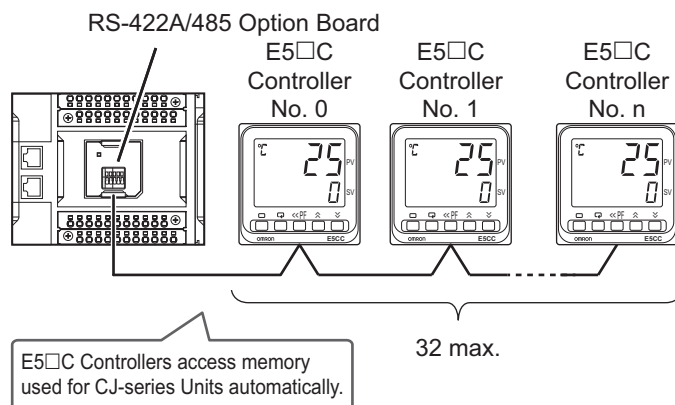
4-3 Programless Communications with E5□C Digital Temperature Controllers

The following describes programless communications with E5□C Digital Temperature Controllers.

4-3-1 Overview

The NX1P2 CPU Unit supports programless communications with E5□C-series Digital Temperature Controllers (hereafter E5□C Controllers) using the host link protocol.

To use this function, you mount a Serial Communications Option Board on the NX1P2 CPU Unit, set its **Serial communications mode** to **Host Link (FINS)** and specify the memory used for CJ-series Units, and connect their serial ports together.



Additional Information

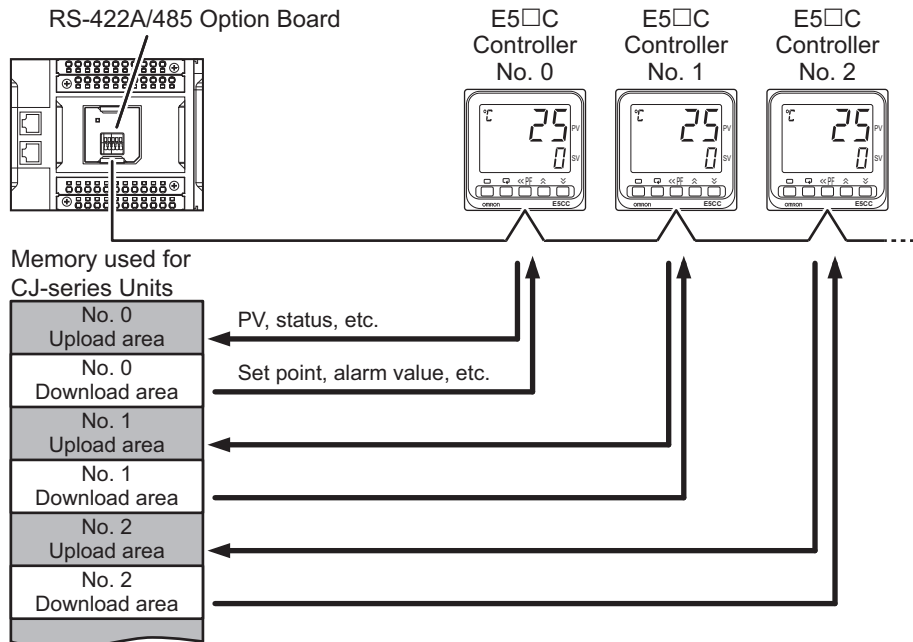
In programless communications with E5□C Controllers, you can read and write E5□C parameters and run/stop the Controller via the memory of a Programmable Controller (hereafter PLC). Because E5□C Controllers communicate with a PLC automatically, there is no need to create a communications program.

Refer to the *E5□C Digital Temperature Controllers Use's Manual* (Cat. No. H174) for the programless communications of E5□C Controllers.

You can connect up to 32 E5□C Controllers to one serial port.

On a E5□C Controller, you set the memory used for CJ-series Units and first address to make available for the E5□C Controller.

Then, the E5□C Controller uses the upload area (for input to the CPU Unit) and download area (for output from the CPU Unit) of PLC memory for each unit number.

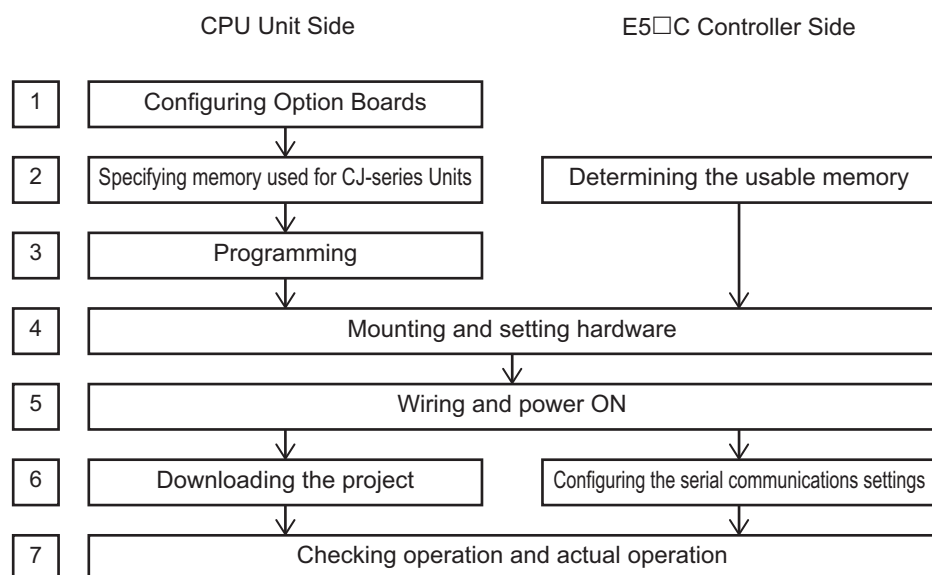


Refer to the *E5□C Digital Temperature Controllers User's Manual* (Cat. No. H174) and the *E5□C Digital Temperature Controllers Communications Manual* (Cat. No. H175) for details on the specifications of E5□C Controllers.

4-3-2 Procedure

The operating procedure is described below.

Overall Procedure



Procedure Details

● CPU Unit Side

No.	Step	Description	Reference
1	Configuring Option Boards	In the Sysmac Studio, specify the Option Board configuration and configure the serial communications mode settings.	4-3-3 <i>Settings</i> on page 4-14 <i>Configuration and Option Board Serial Communications Settings</i> on page 4-14
2	Specifying memory used for CJ-series Units	In the Sysmac Studio, set the number of words of memory for the DM area type memory to make available for the E5□C Controller.	4-3-3 <i>Settings</i> on page 4-14 <i>Memory Settings for CJ-series Units</i> on page 4-14
3	Programming	In the Sysmac Studio, create a program to access the memory used for CJ-series Units by using user-defined variables with AT specifications.	4-3-4 <i>Programming</i> on page 4-17

No.	Step	Description	Reference
4	Mounting and setting hardware	Set the operating setting DIP switches on the back of the NX1W-CIF11/CIF12 Option Board. Mount the Option Boards and necessary Units. Install the CPU Unit and the E5□C Controller.	4-3-5 Connection Examples on page 4-17 NX-series NX1P2 CPU Unit Hardware User's Manual (Cat. No. W578)
5	Wiring and power ON	Connect the serial communications terminals of the Option Board and the E5□C Controller. Wire the power supply terminals and turn ON the power supply.	
6	Downloading the project	Download the project from the Sysmac Studio.	NJ/NX-series CPU Unit Software User's Manual (Cat. No. W501)
7	Checking operation and actual operation	Check the operation of the user program and programless communications with the E5□C Controller.	Sysmac Studio Version 1 Operation Manual (Cat. No. W504)

● E5□C Controller Side

No.	Step	Description	Reference
2	Determining the usable memory	Determine the number of words of memory to make available for the E5□C Controller.	E5□C Digital Temperature Controllers User's Manual (Cat. No. H174) E5□C Digital Temperature Controllers Communications Manual (Cat. No. H175)
4	Mounting and setting hardware	Install the CPU Unit and the E5□C Controller.	
5	Wiring and power ON	Connect the serial communications terminals of the Option Board and the E5□C Controller. Wire the power supply terminals and turn ON the power supply.	
6	Configuring the serial communications settings	On the E5□C Controller, set the serial communications parameters in the <i>communications setting level</i> .	
7	Checking operation and actual operation	Check the operation of the user program and programless communications with the E5□C Controller.	



Additional Information

The CPU Unit fails to receive a command frame if a response frame from the CPU Unit is received and the next command frame from the E5□C Controller is sent too soon.

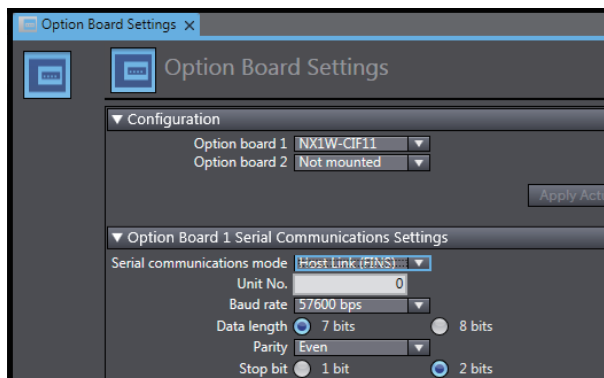
When a receiving of the command frame failed, increase the send data wait time of the E5□C Controller.

4-3-3 Settings

Settings on the CPU Unit

● Configuration and Option Board Serial Communications Settings

Configure these settings in the Option Board Settings Tab Page, which is displayed by selecting **Option Board Settings** under **Configurations and Setup - Controller Setup**.



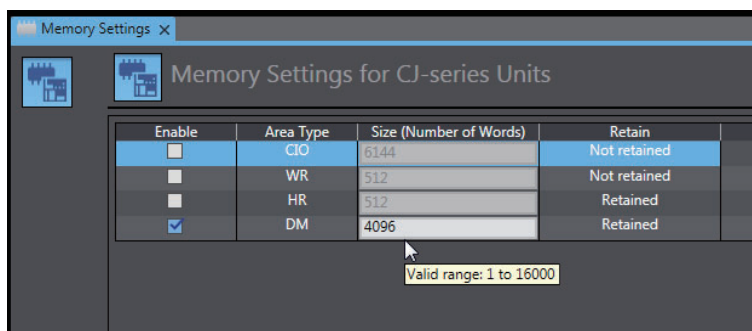
Under **Configuration**, specify the models of the Option Boards to use.

Under **Option Board Settings**, configure the following settings:

Item	Set value
Serial communications mode	Host Link (FINS)
Unit No.	Set the same Communications Node Number as for the E5□C Controller.
Baud rate	57,600 bps
Data length	7 bits
Parity	Even
Stop bit	2 bits

● Memory Settings for CJ-series Units

Specify the memory used for CJ-series Units in the Memory Settings for CJ-series Units Tab Page, which is displayed by selecting **Memory Settings** under **Configurations and Setup - Controller Setup**.



Determine and set the area type and the number of words based on the first address of the E5□C Controller and the number of Controllers to connect.



Precautions for Correct Use

The NX1P2 CPU Unit does not support the EM area type.

Set the **Area** to **DM** on the E5□C Controller.

Settings on the NX1W-CIF11/CIF12 Option Board

You need to use an NX1W-CIF11 or NX1W-CIF12 Option Board to connect via RS-422A/485.

The table below shows the settings of the operation setting DIP switches on the back of the Option Board.

CIF11		CIF12		Setting	Setting description
SW	No.	SW	No.		
SW1	1	SW1	1	ON	With terminating resistance ^{*1}
	2		2	ON	Two-wire type
	3		3	ON	Two-wire type
	4		4	OFF	(Not used)
	5	SW2	1	ON	With RS control for receive data
	6		2	ON	With RS control for send data

*1. Turn this OFF if the NX1W-CIF11/CIF12 is not the terminating device.

Settings on E5□C Controllers

On the E5□C Controller to connect, move from the *operation level*, through the *initial setting level*, to the *communications setting level* and set programless communications parameters.

The settings are as follows:

Parameter name *1, *2	Set value
Protocol Setting	Host Link (FINS)
Communications Unit No.	0*3
Communications Baud Rate	57,600 bps
Communications Data Length*4	7
Communications Parity*4	Even
Communications Stop Bits*4	2
Send Data Wait Time	1 ms*5
Highest Communications Unit No.	0*6
Area*7, First Address Upper Word, First Address Lower Word	Set the area and the first address of memory to make available for the E5□C Controller.*8
Communications Node Number	Set the same value as that is set in the Unit No. in the Option Board Settings Tab Page on the CPU Unit.

*1. Only the required parameters are listed.

*2. When you connect more than one E5□C Controller, set these parameters for all of the E5□C Controllers. Set the same value for each parameter, except for the **Communications Unit No.**

*3. When you connect more than one E5□C Controller, set a series of numbers starting from 0 for the E5□C Controllers.

*4. If you set the **Protocol Setting** parameter to **Host Link (FINS)**, the **Communications Data Length**, **Communications Parity**, and **Communications Stop Bits** parameters are automatically set to 7, Even, and 2, respectively. These parameters cannot be changed.

*5. When the CPU Unit fails to receive a command frame, increase the set value.

*6. When you connect more than one E5□C Controller, set the highest communications unit number.

*7. Set the **Area** parameter to **DM** because the NX1P2 CPU Unit does not support the EM area type.

*8. When you connect more than one E5□C Controller, set the first address to the same value for all of the E5□C Controllers.

Refer to the *E5□C Digital Temperature Controllers Communications Manual* (Cat. No. H175) for details on the parameters used for programless communications of E5□C Controllers.

4-3-4 Programming

Assign the user-defined variables that is used in the user program on the CPU Unit to the memory used for CJ-series Units that will be accessed by the E5□C Controller by using AT specification.

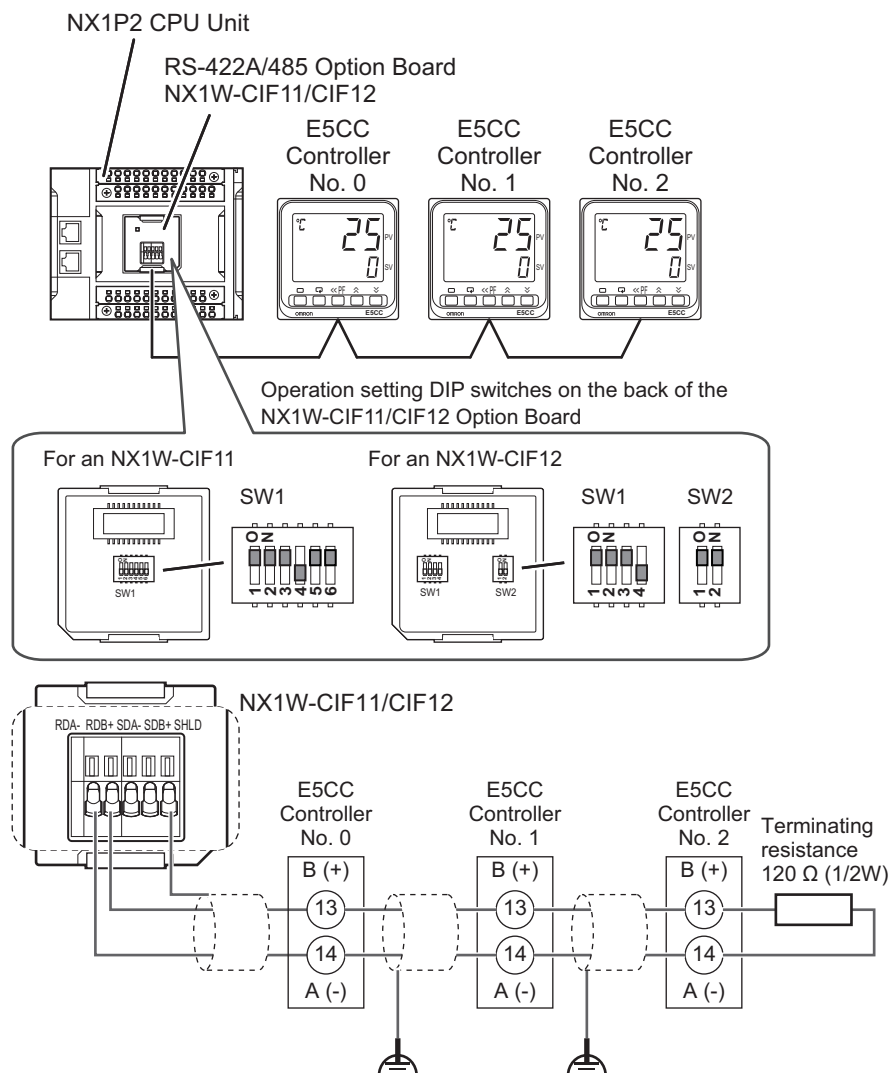
Then, create the user program for communicating with the E5□C Controller.

The user program gets the E5□C Controller status from the upload area of the E5□C Controller and sends commands to the download area of the E5□C Controller.

Refer to the *NJ/NX-series CPU Unit Software User's Manual* (Cat. No. W501) for memory used for CJ-series Units and AT specification.

4-3-5 Connection Examples

An example of connecting three E5CC Controllers is given below.



Refer to the *E5□C Digital Temperature Controllers Communications Manual* (Cat. No. H175) for details on wiring an E5□C Controller.

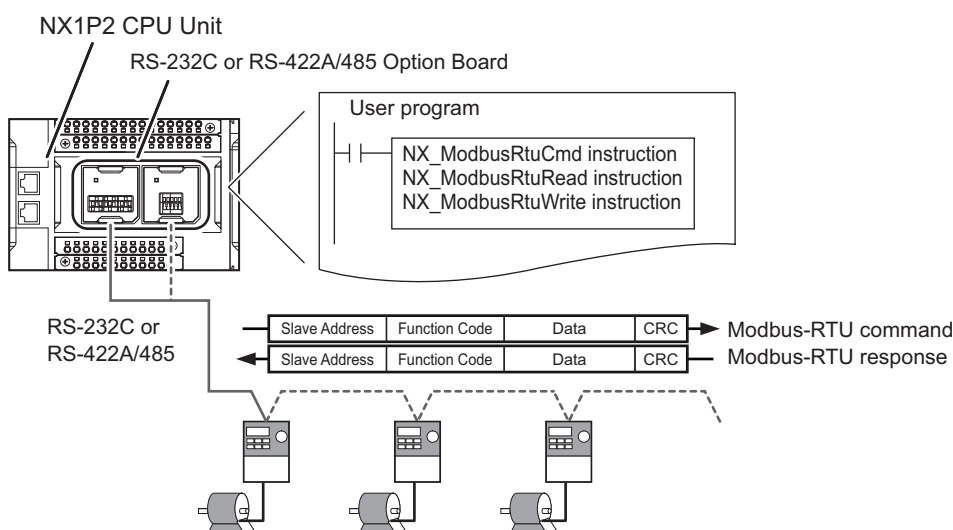
4-4 Connection with Modbus-RTU Slaves

The following describes data exchange between the CPU Unit and Modbus-RTU slaves.

4-4-1 Overview

The NX1P2 CPU Unit can exchange data with Modbus-RTU slaves by using instructions to send a Modbus-RTU command and receive a response.

To use this function, you mount a Serial Communications Option Board on the NX1P2 CPU Unit, set its **Serial communications mode** to **Modbus-RTU Master**, and connect their serial ports together.



The processing of the Modbus-RTU protocol and message frame format is handled by the instructions to send a Modbus-RTU command and receive a response.

Therefore, you can easily create a program to exchange data with Modbus-RTU slaves.

Refer to the *NJ/NX-series Instructions Reference Manual* (Cat. No. W502-E1-17 or later) for the specification of instructions.



Additional Information

The frame format of Modbus-RTU commands is as follows.

Slaves Address	Function Code	Data	CRC
1 byte	1 byte	0 to 252 bytes	2 bytes*

* The byte order of the CRC code is lower byte, then higher byte.

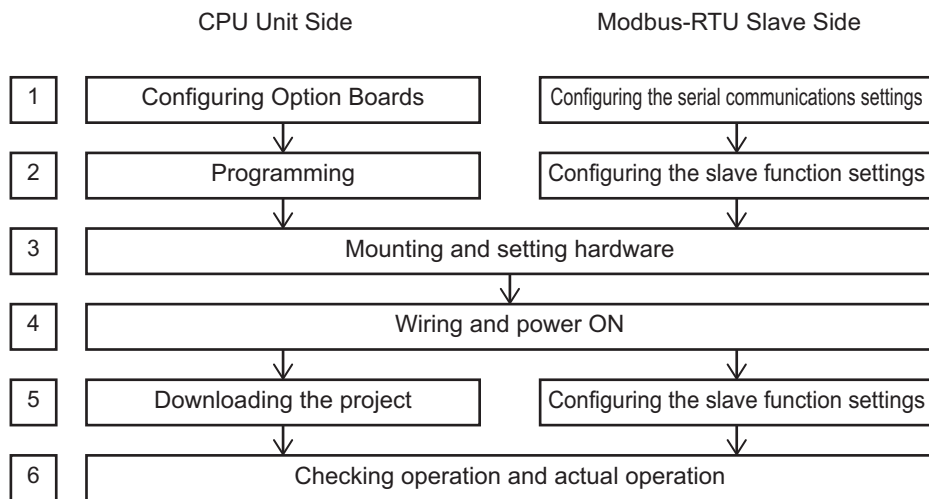
Refer to *MODBUS Application Protocol Specification* for the specifications of the MODBUS communications protocol. You can obtain *MODBUS Application Protocol Specification* from MODBUS Organization, Inc.

<http://www.modbus.org/>

4-4-2 Procedure

The operating procedure is described below.

Overall Procedure



Procedure Details

● CPU Unit Side

No.	Step	Description	Reference
1	Configuring Option Boards	In the Sysmac Studio, specify the Option Board configuration and configure the serial communications mode settings.	4-4-3 <i>Settings</i> on page 4-21 <i>Configuration and Option Board Serial Communications Settings</i> on page 4-21
2	Programming	Create a program to exchange data with Modbus-RTU slaves by using special instructions.	4-4-4 <i>Programming</i> on page 4-22
3	Mounting and setting hardware	If you are using an NX1W-CIF11/CIF12 Option Board, set the operating setting DIP switches on the back. Mount the Option Boards and necessary Units. Install the CPU Unit and Modbus-RTU slaves.	4-4-5 <i>Connection Examples</i> on page 4-24 <i>NX-series NX1P2 CPU Unit Hardware User's Manual</i> (Cat. No. W578)
4	Wiring and power ON	Connect the serial communications terminals of the Option Boards and Modbus-RTU slaves. Wire the power supply terminals and turn ON the power supply.	
5	Downloading the project	Download the project from the Sysmac Studio.	<i>NJ/NX-series CPU Unit Software User's Manual</i> (Cat. No. W501)
6	Checking operation and actual operation	Check the operation of the user program and Modbus-RTU slaves.	<i>Sysmac Studio Version 1 Operation Manual</i> (Cat. No. W504)

● Modbus-RTU Slave Side

No.	Step	Description	Reference
1	Configuring the serial communications settings	Configure serial communications settings on Modbus-RTU slaves.	Manuals and technical materials for Modbus-RTU slaves
2	Configuring the slave function settings	Configure the functions of Modbus-RTU slaves as required.	
3	Mounting and setting hardware	Install the CPU Unit and Modbus-RTU slaves.	
4	Wiring and power ON	Connect the serial communications terminals of the Option Boards and Modbus-RTU slaves. Wire the power supply terminals and turn ON the power supply.	
5	Configuring the slave function settings	Configure the functions of Modbus-RTU slaves as required.	
6	Checking operation and actual operation	Check the operation of the user program and Modbus-RTU slaves.	



Additional Information

When you use a two-wire type of RS-485, be careful as follows:

The CPU Unit fails to receive the response frame if a response frame from the Modbus-RTU slave is sent too soon for the Modbus-RTU command frame.

When a receiving of the response frame failed, the following adjustments are needed.

- Adjustment on the CPU Unit side

Increase the number of retries of serial communications instructions for the Modbus-RTU master in the user program.

The number of retries is set by an *Option.Retry* (Retry count) input variable of the Modbus-RTU master instructions.

- Adjustment on the Modbus-RTU Slave side

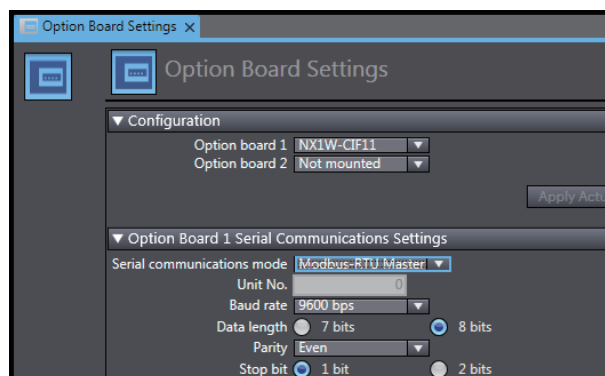
Increase the wait time from when the Modbus-RTU slave receives a command until it sends a response.

4-4-3 Settings

Settings on the CPU Unit

● Configuration and Option Board Serial Communications Settings

Configure these settings in the Option Board Settings Tab Page, which is displayed by selecting **Option Board Settings** under **Configurations and Setup - Controller Setup**.



Under **Configuration**, specify the models of the Option Boards to use.

Under **Option Board Settings**, configure the following settings:

Item	Set value
Serial communications mode	Modbus-RTU Master
Unit No.	Settings not required.
Baud rate	Set this to match the setting on the Modbus-RTU slave to connect.
Data length	8 bits
Parity	Even
Stop bit	1 bit

● Memory Settings for CJ-series Units

There is no need to configure these settings because memory used for CJ-series Units is not used in Modbus-RTU Master mode.

Settings on the NX1W-CIF11/CIF12 Option Board

The CPU Unit requires an NX1W-CIF11 or NX1W-CIF12 Option Board for connection with external devices via RS-422A/485.

Set the operation setting DIP switches on the back according to the specifications of the Modbus-RTU slave to connect.

Settings on Modbus-RTU Slaves

Set the same baud rate as for the NX1P2 CPU Unit.

Set the MODBUS slave address.

Configure the functions of Modbus-RTU slaves as required.

Refer to the manual for the Modbus-RTU slave to connect.

4-4-4 Programming

Create a program to exchange data with Modbus-RTU slaves by using instructions to send a Modbus-RTU command and receive a response.

No program is needed for the processing of the Modbus-RTU protocol and message frame format, because it is handled by the instructions.

Serial Communications Instructions That You Can Use in Modbus-RTU Master Mode

The table below shows serial communications instructions that you can use when the **Serial communications mode** is **Modbus-RTU Master**.

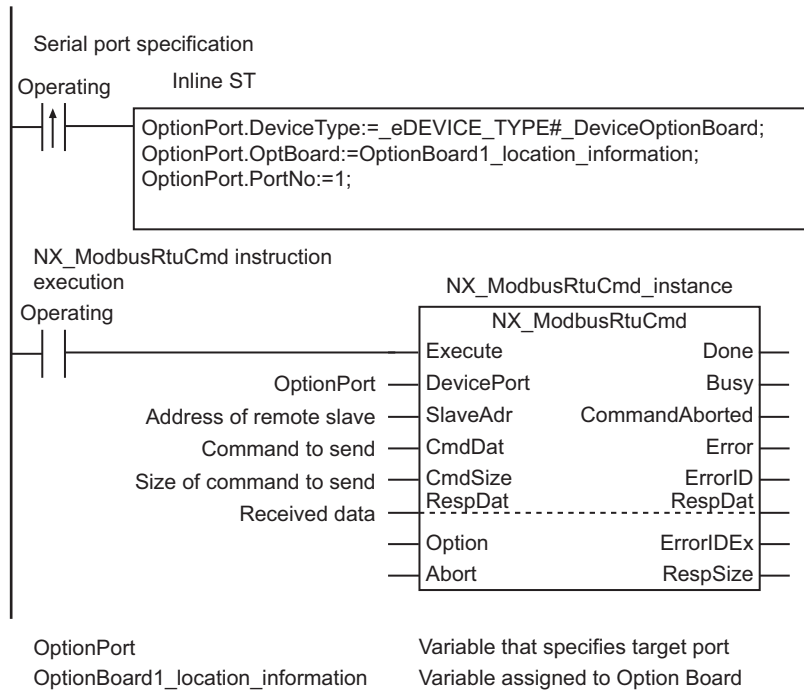
Instruction	Name	Outline of function
NX_ModbusRtuCmd	Send Modbus RTU General Command	Sends general commands from a serial port on a CIF Unit or Option Board to Modbus-RTU slaves using Modbus-RTU protocol.
NX_ModbusRtuRead	Send Modbus RTURead Command	Sends read commands from a serial port on a CIF Unit or Option Board to Modbus-RTU slaves using Modbus-RTU protocol.
NX_ModbusRtuWrite	Send Modbus RTUWrite Command	Sends write commands from a serial port on a CIF Unit or Option Board to Modbus-RTU slaves using Modbus-RTU protocol.

Refer to the *NJ/NX-series Instructions Reference Manual* (Cat. No. W502-E1-17 or later) for details on these instructions.

Option Board Specification

Instructions to send a Modbus-RTU command specify an Option Board by using variables assigned to the Option Board.

Example of specifying an Option Board with *OptionBoard1_location_information* variable assigned to it

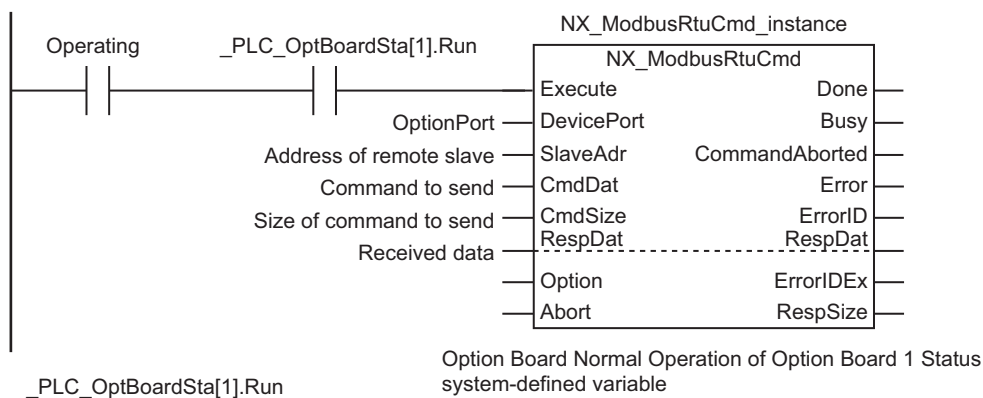


Refer to 3-2-4 *Assigning Device Variables to Option Boards* on page 3-11 for assigning variables to Option Boards.

Option Board Status

To use serial communications instructions, program the *.Run* (Option Board Normal Operation) member of the *_PLC_OptBoardSta* (Option Board Status) system-defined variable as an interlock condition in the user program.

Example of executing a Send Modbus RTU General Command instruction to an Option Board mounted on Option Board 1 using the Option Board Normal Operation as an interlock condition

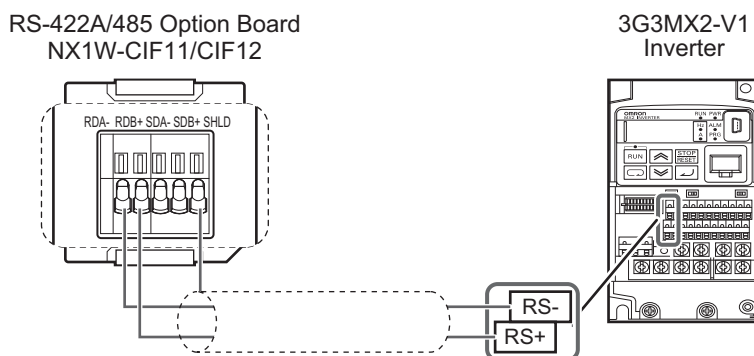


Refer to the *NJ/NX-series CPU Unit Software User's Manual* (Cat. No. W501) for the specifications of the *_PLC_OptBoardSta* (Option Board Status) system-defined variable.

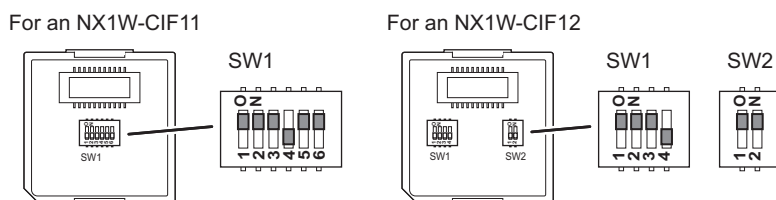
4-4-5 Connection Examples

An example of connecting an OMRON 3G3MX2-V1 Inverter via RS-422A/485 is given below.

● Serial Communications Terminals



● Operation Setting DIP Switches on the Back of the NX1W-CIF11/CIF12 Option Board



CIF11		CIF12		Setting	Setting description
SW	No.	SW	No.		
SW1	1	SW1	1	ON	With terminating resistance
	2		2	ON	Two-wire type
	3		3	ON	Two-wire type
	4		4	OFF	(Not used)
	5	SW2	1	ON	With RS control for receive data
	6		2	ON	With RS control for send data

● Settings on 3G3MX2-V1 Inverters

• Parameter Settings

Item	Set value
Communication Speed Selection (Baud Rate Selection)	Set this to match the serial port setting.
Communication Station No. Selection	Set a desired value.
Communication Parity Selection	Even parity
Communication Stop Bit Selection	1 bit
Communication Wait Time	0 ms ^{*1}
Communication Selection	Modbus communication

*1. When the CPU Unit fails to receive a response frame, increase the set value.

• Switch Setting

Set the terminating resistor selector switch to ON.

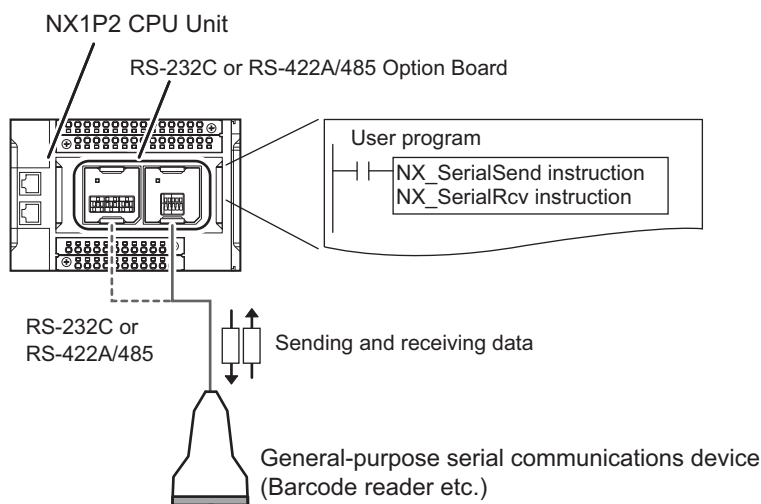
4-5 Connection with General-purpose Serial Communications Devices

The following describes data exchange between the CPU Unit and general-purpose serial communications devices.

4-5-1 Overview

The NX1P2 CPU Unit can exchange data with general-purpose serial communications devices by using the instructions to send and receive data in No-Protocol mode.

To use this function, you mount a Serial Communications Option Board on the NX1P2 CPU Unit, set its **Serial communications mode to No-Protocol**, and connect their serial ports together.



The instruction to send data in No-Protocol mode refers to an instruction to output the specified data from the specified serial port without converting it.

The instruction to receive data in No-Protocol mode refers to an instruction that reads data received at the specified port into the specified variable without converting it.

Refer to the *NJ/NX-series Instructions Reference Manual* (Cat. No. W502-E1-17 or later) for the specification of instructions.



Additional Information

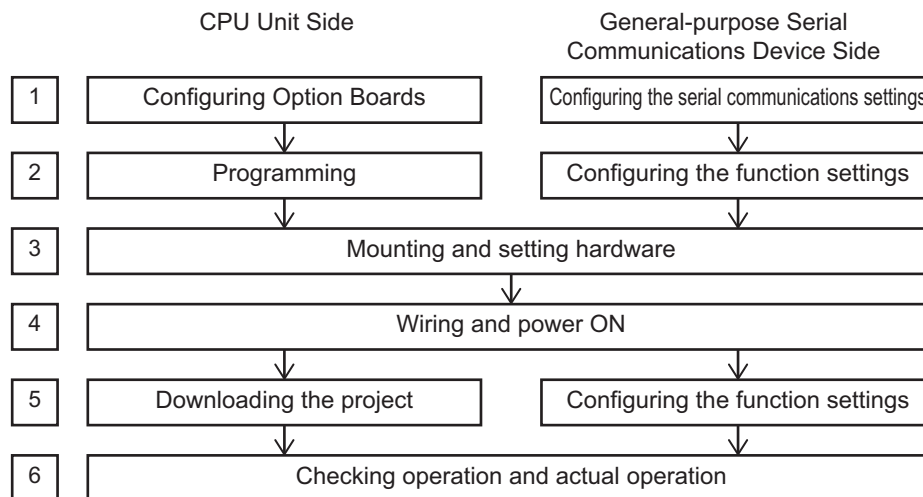
To enable the CPU Unit to exchange data with general-purpose serial communications devices by using the instructions to send and receive data in No-Protocol mode, you must program the communications procedure (protocol) for the remote device.

For example, program the sequence processing and retry processing between the command and the response, data type conversion processing, branching processing, and other processing steps to be performed based on the communications protocol of the remote device.

4-5-2 Procedure

The operating procedure is described below.

Overall Procedure



Procedure Details

● CPU Unit Side

No.	Step	Description	Reference
1	Configuring Option Boards	In the Sysmac Studio, specify the Option Board configuration and configure the serial communications mode settings.	4-4-3 <i>Settings</i> on page 4-21 <i>Configuration and Option Board Serial Communications Settings</i> on page 4-21
2	Programming	Create a program to exchange data with general-purpose serial communications devices by using No-protocol Mode instructions.	4-4-4 <i>Programming</i> on page 4-22
3	Mounting and setting hardware	If you are using an NX1W-CIF11/CIF12 Option Board, set the operation setting DIP switches on the back. Mount the Option Boards and necessary Units. Install the CPU Unit and general-purpose serial communications devices.	4-4-5 <i>Connection Examples</i> on page 4-24 <i>NX-series NX1P2 CPU Unit Hardware User's Manual</i> (Cat. No. W578)
4	Wiring and power ON	Connect the serial communications terminals of the Option Boards and general-purpose serial communications devices. Wire the power supply terminals and turn ON the power supply.	

No.	Step	Description	Reference
5	Downloading the project	Download the project from the Sysmac Studio.	<i>NJ/NX-series CPU Unit Software User's Manual</i> (Cat. No. W501)
6	Checking operation and actual operation	Check the operation of the user program and general-purpose serial communications devices.	<i>Sysmac Studio Version 1 Operation Manual</i> (Cat. No. W504)

● Serial Communications Device Side

No.	Step	Description	Reference
1	Configuring the serial communications settings	Configure the serial communications settings on general-purpose serial communications devices.	Manuals and technical materials for general-purpose serial communications devices
2	Configuring the slave function settings	Configure the functions of general-purpose serial communications devices as required.	
3	Mounting and setting hardware	Install the CPU Unit and general-purpose serial communications devices.	
4	Wiring and power ON	Connect the serial communications terminals of the Option Boards and general-purpose serial communications devices. Wire the power supply terminals and turn ON the power supply.	
5	Configuring the slave function settings	Configure the functions of general-purpose serial communications devices as required.	
6	Checking operation and actual operation	Check the operation of the user program and general-purpose serial communications devices.	



Additional Information

When you use a two-wire type of RS-485, be careful as follows:

The CPU Unit fails to receive data if data from the CPU Unit is received and data from the general-purpose serial communications device is sent too soon.

When a receiving of data failed, the following adjustments are needed.

- Adjustment on the CPU Unit side

Add the retry processing to the sequence processing of data communications in the user program or increase the number of retry processes.

- Adjustment on the general-purpose serial communications device side

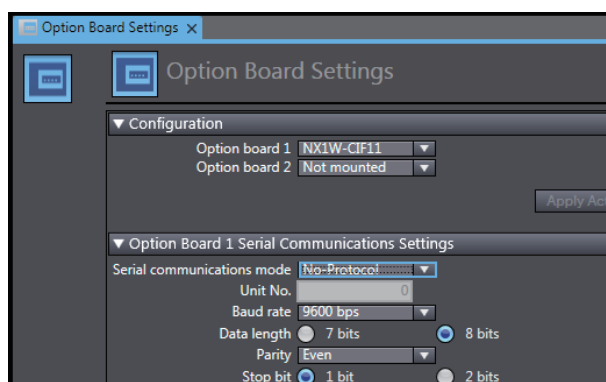
Increase the wait time from when the general-purpose serial communications device receives the data until it sends the data.

4-5-3 Settings

Settings on the CPU Unit

● Configuration and Option Board Serial Communications Settings

Configure these settings in the Option Board Settings Tab Page, which is displayed by selecting **Option Board Settings** under **Configurations and Setup - Controller Setup**.



Under **Configuration**, specify the models of the Option Boards to use.

Under **Option Board Settings**, configure the following settings:

Item	Set value
Serial communications mode	No-Protocol
Unit No.	Settings not required.
Baud rate	Set this to match the setting on the general-purpose serial communications device to connect.
Data length	Set this to match the setting on the general-purpose serial communications device to connect.
Parity	Set this to match the setting on the general-purpose serial communications device to connect.
Stop bit	Set this to match the setting on the general-purpose serial communications device to connect.

● Memory Settings for CJ-series Units

There is no need to configure these settings because memory used for CJ-series Units is not used in No-Protocol mode.

Settings on the NX1W-CIF11/CIF12 Option Board

The CPU Unit requires an NX1W-CIF11 or NX1W-CIF12 Option Board for connection with external devices via RS-422A/485.

Set the operation setting DIP switches on the back according to the specifications of the general-purpose serial communications device to connect.

General-purpose Serial Communications Device Settings

Configure the serial communications settings to match those of the NX1P2 CPU Unit.

Refer to the manual for the general-purpose serial communications device to connect.

4-5-4 Programming

Create a program to exchange data with general-purpose serial communications devices by using the instructions to send and receive data in No-Protocol mode.

For example, program the sequence processing and retry processing between the command and the response, data type conversion processing, branching processing, and other processing steps to be performed based on the communications protocol of the remote device.

Serial Communications Instructions Used in No-Protocol Mode

The table below shows serial communications instructions that you can use when the **Serial communications mode** is **No-Protocol**.

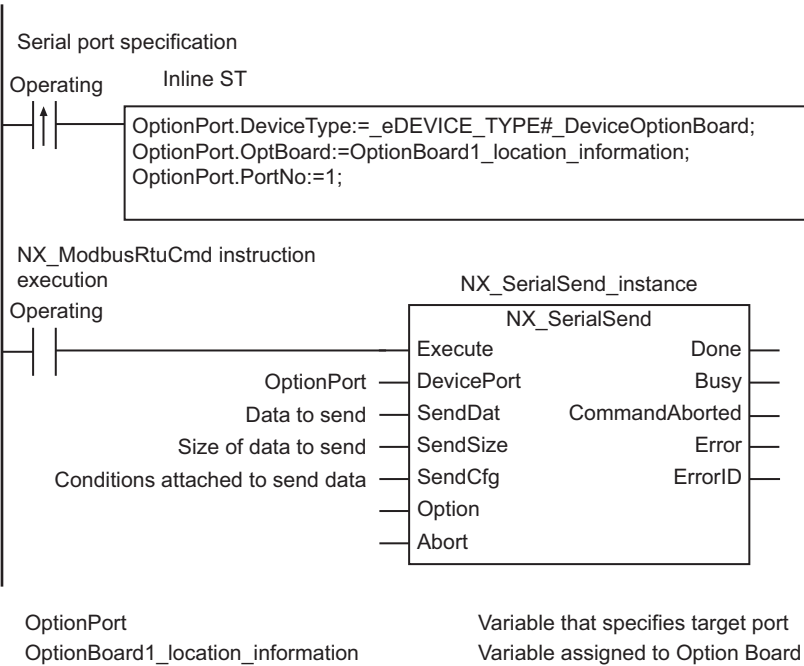
Instruction	Name	Outline of function
NX_SerialSend	Send No-protocol Data	Sends data in No-Protocol mode from a serial port on a CIF Unit or Option Board.
NX_SerialRcv	Receive No-protocol Data	Reads data in No-Protocol Mode from a serial port on a CIF Unit or Option Board.
NX_SerialSigCtl	Serial Control Signal ON/OFF Switching	Turns ON or OFF the ER or RS signal of a serial port on a CIF Unit or Option Board.
NX_SerialSigRead	Read Serial Control Signal	Reads the CS or DR signal of a serial port on an Option Board.
NX_SerialStatus-Read	Read Serial Port Status	Reads the status of a serial port on an Option Board.
NX_SerialBufClear	Clear Buffer	Clears the send or receive buffer.

Refer to the *NJ/NX-series Instructions Reference Manual* (Cat. No. W502-E1-17 or later) for details on these instructions.

Option Board Specification

Instructions that are used in No-Protocol Mode specify an Option Board by using variables assigned to the Option Board.

Example of specifying an Option Board with *OptionBoard1_location_information* variable assigned to it

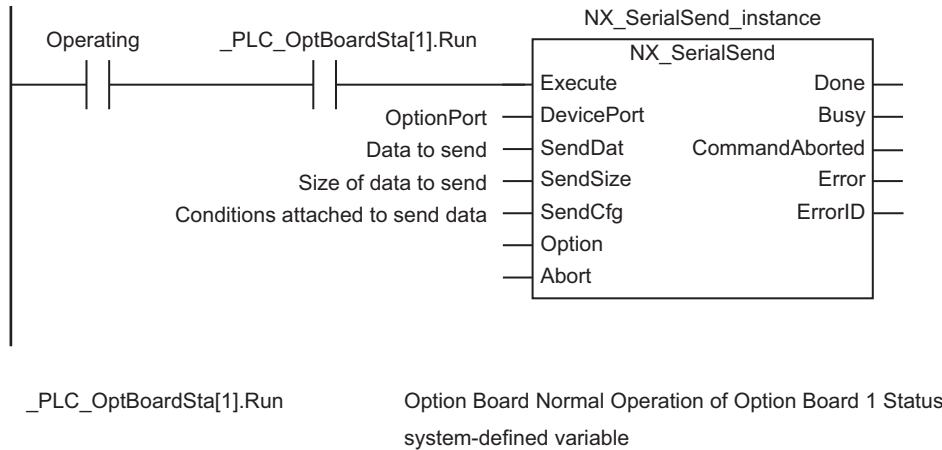


Refer to 3-2-4 *Assigning Device Variables to Option Boards* on page 3-11 for assigning variables to Option Boards.

Option Board Status

To use serial communications instructions, program the *.Run* (Option Board Normal Operation) member of the *_PLC_OptBoardSta* (Option Board Status) system-defined variable as an interlock condition in the user program.

Example of executing a Send No-protocol Data instruction to an Option Board mounted on Option Board 1 using the Option Board Normal Operation as an interlock condition



Refer to the *NJ/NX-series CPU Unit Software User's Manual* (Cat. No. W501) for the specifications of the *_PLC_OptBoardSta* (Option Board Status) system-defined variable.

Analog I/O

This section describes the functions of Analog I/O Option Boards for the NX1P2 CPU Units.

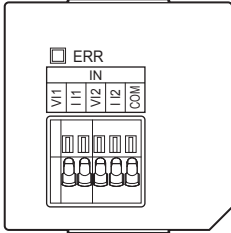
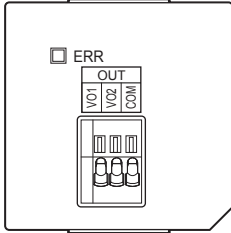
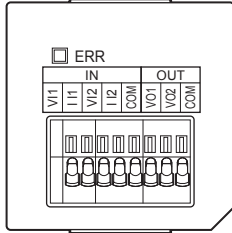
5-1	Specifications	5-2
5-1-1	Analog I/O Option Boards	5-2
5-1-2	Part Names and Functions	5-2
5-1-3	Terminal Arrangement	5-3
5-1-4	Input Range and Output Range	5-3
5-2	Procedure	5-5
5-3	Settings	5-6
5-3-1	Option Board Settings	5-6
5-3-2	Device Variables	5-7
5-4	Programming	5-8
5-4-1	I/O Data	5-8
5-4-2	Option Board Status	5-9
5-4-3	Special Instructions for Analog I/O Option Boards	5-9
5-4-4	Precautions on Supported Functions	5-10
5-5	Wiring	5-11
5-6	I/O Refreshing	5-12
5-6-1	I/O Refresh Operation	5-12
5-6-2	Response Time	5-13

5-1 Specifications

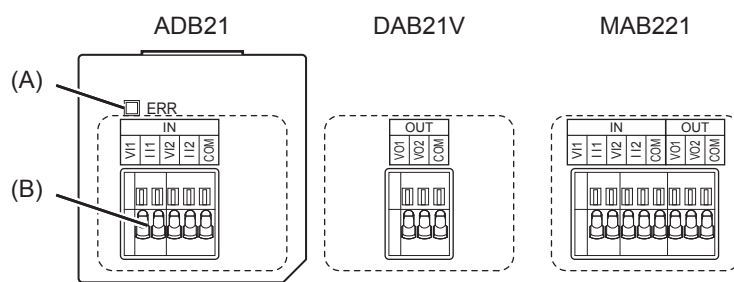
The following describes the types and specifications of Analog I/O Option Boards.

5-1-1 Analog I/O Option Boards

The following table shows the types and summary specifications of Analog I/O Option Boards.

Item	NX1W-ADB21	NX1W-DAB21V	NX1W-MAB221
Appearance			
Analog input	2 inputs	None	2 inputs
Input range	0 to 10 V, 0 to 20 mA	---	0 to 10 V, 0 to 20 mA
Resolution	1/4,000, 1/2,000	---	1/4,000, 1/2,000
Analog output	None	2 outputs	2 outputs
Output range	---	0 to 10 V	0 to 10 V
Resolution	---	1/4,000	1/4,000
Conversion time	4 ms/Option Board	4 ms/Option Board	6 ms/Option Board
Isolation	No-isolation	No-isolation	No-isolation
External connection terminal	Screwless clamping terminal block	Screwless clamping terminal block	Screwless clamping terminal block

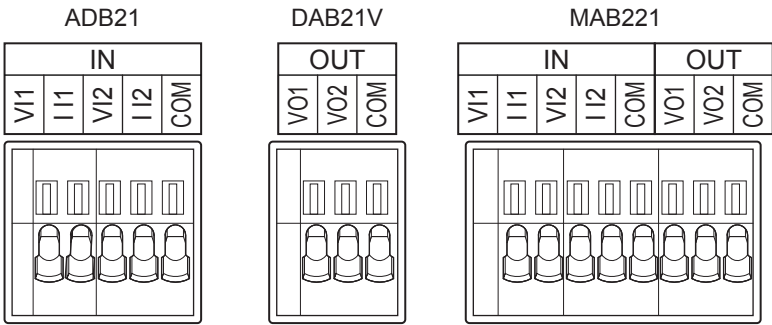
5-1-2 Part Names and Functions



Symbol	Name	Function
A	Status indicator	Displays the operating status of the Analog I/O Option Board.
B	Terminal block	The terminal block for wiring the analog input and analog output terminals.

5-1-3 Terminal Arrangement

The following table shows the terminal arrangement of Analog I/O Option Boards.



Refer to the *NX-series NX1P2 CPU Unit Hardware User's Manual* (Cat. No. W578) for information on wiring Analog I/O Option Boards.

5-1-4 Input Range and Output Range

Input Range and Converted Values

The NX1W-ADB21 and NX1W-MAB221 Analog I/O Option Boards support two analog input ranges in different units: 0 to 10 V and 0 to 20 mA.

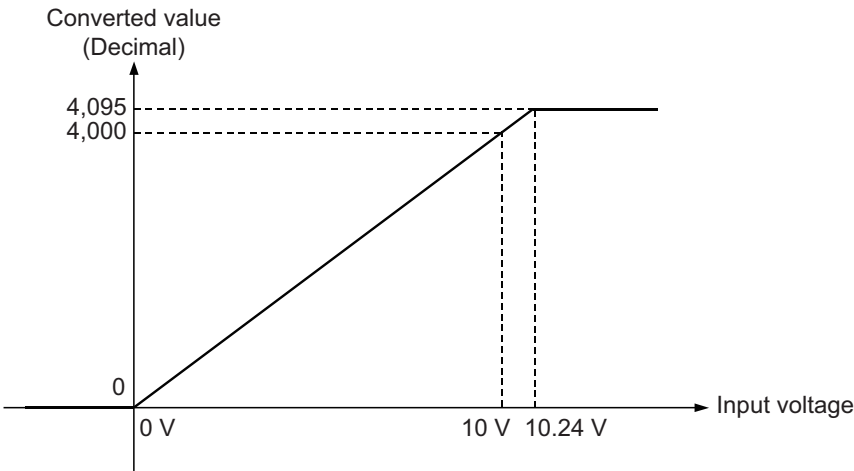
There is no need to select the input range.

If the input signal exceeds the allowable value conversion range, the converted value is fixed at the upper or lower limit.

● 0 to 10 V

An input voltage of 0 to 10 V is converted to a signed integer value (0 to 4,000).

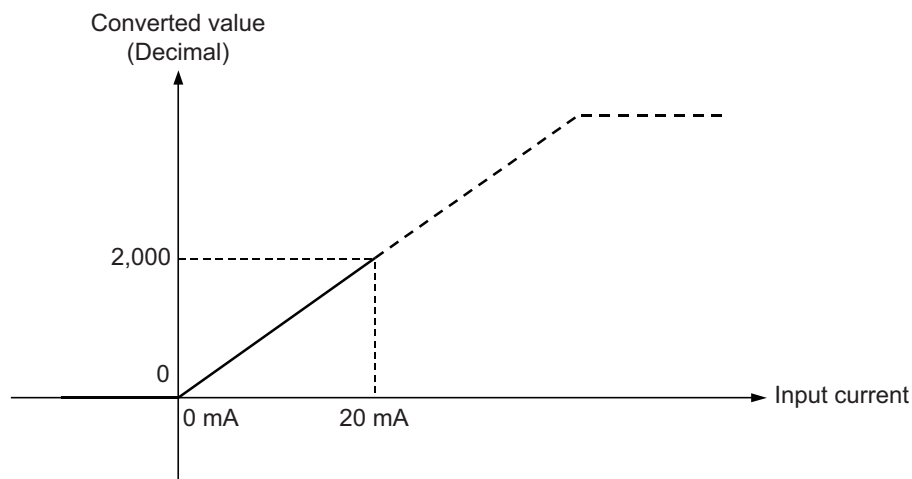
The allowable value conversion range is 0 to 4,095.



● 0 to 20 mA

An input current of 0 to 20 mA is converted to a signed integer value (0 to 2,000).

The allowable value conversion range is 0 to 4,095. However, the input current cannot exceed the absolute maximum rating, which is 30 mA.

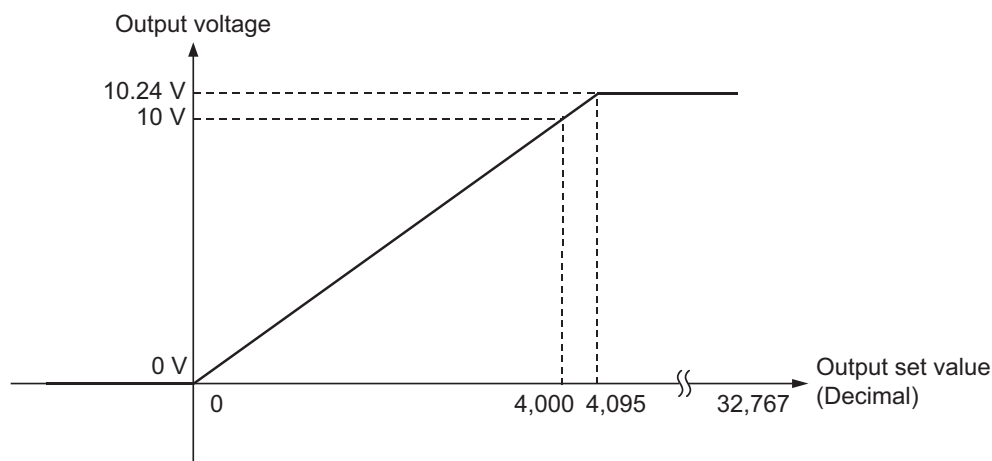


Output Range and Output Set Values

The NX1W-ADB21 and NX1W-MAB221 Analog I/O Option Boards support a single analog output range: 0 to 10 V.

If the output set value exceeds the allowable value conversion range, the analog value is fixed at the upper or lower limit.

The output set value of the signed integer (0 to 4,095) is converted to voltage from 0 to 10 V and output. The allowable output set value conversion range is 0 to 4,095.

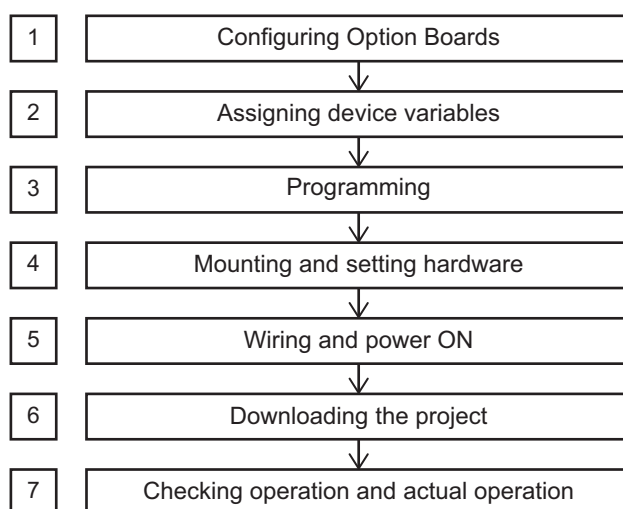


In the output mode at load rejection, the value when the output set value is 0 is output.

5-2 Procedure

The operating procedure is described below.

Overall Procedure



Procedure Details

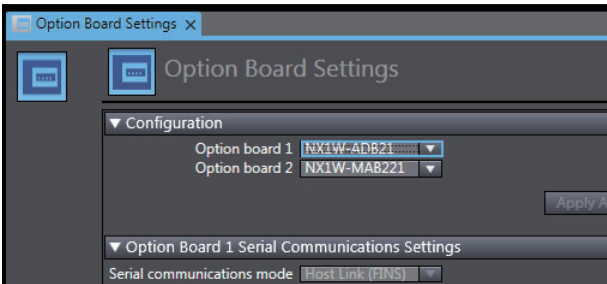
No.	Step	Description	Reference
1	Configuring Option Boards	In the Sysmac Studio, specify the Option Board configuration.	5-3-1 Option Board Settings on page 5-6
2	Assigning device variables	In the Sysmac Studio, assign device variables to I/O ports.	5-3-2 Device Variables on page 5-7
3	Programming	In the Sysmac Studio, create a program that uses the device variables to manipulate I/O data.	5-4 Programming on page 5-8
4	Mounting and setting hardware	Mount the Option Boards and necessary Units. Install the CPU Unit.	NX-series NX1P2 CPU Unit Hardware User's Manual (Cat. No. W578)
5	Wiring and power ON	Wire the Option Boards and connected external device to analog I/O terminals. Wire the power supply terminals and turn ON the power supply.	
6	Downloading the project	Download the project from the Sysmac Studio.	NJ/NX-series CPU Unit Software User's Manual (Cat. No. W501)
7	Checking operation and actual operation	Check the wiring. Check the operation of the user program and screen data.	Sysmac Studio Version 1 Operation Manual (Cat. No. W504)

5-3 Settings

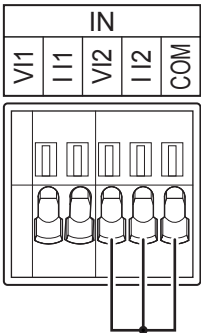
The following describes the settings required to use Analog I/O Option Boards.

5-3-1 Option Board Settings

Specify the models of the Option Boards to use in the Option Board Settings Tab Page, which is displayed by selecting **Option Board Settings** under **Configurations and Setup - Controller Setup**.



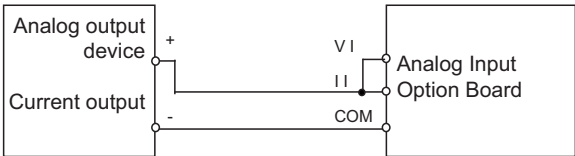
No settings are provided to select each analog input or output to use.
For analog inputs that are not used, short-circuit the V I, I I, and COM input terminals.
For analog outputs that are not used, do not connect the output terminals.



Additional Information

The input terminals are always subjected to AD conversion even when they are not used.
Therefore, if unused input terminals are left unconnected, unintended conversion values may be input.

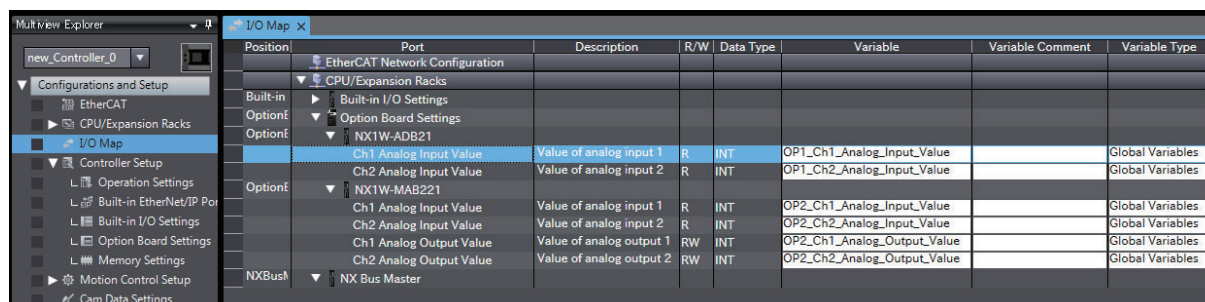
Although the Option Boards support two input ranges, no settings are provided to select the range.
Connect remote devices with the voltage or current input terminals based on their specifications.
When you use the current input range, however, short-circuit the current input terminal with the voltage input terminal.



5-3-2 Device Variables

To use I/O data for an Option Board in the user program, you assign a device variable to each I/O port. Specify device variables in the I/O Map Tab Page, which is displayed by selecting **Configurations and Setup - I/O Map**.

Refer to 5-4-1 I/O Data on page 5-8 for I/O data for Analog I/O Option Boards.



Position	Port	Description	R/W	Data Type	Variable	Variable Comment	Variable Type
	EtherCAT Network Configuration						
	CPU/Expansion Racks						
Built-in	Built-in I/O Settings						
Option Board	Option Board Settings						
	NX1W-ADB21						
	Ch1 Analog Input Value	Value of analog input 1	R	INT	OP1_Ch1_Analog_Input_Value		Global Variables
	Ch2 Analog Input Value	Value of analog input 2	R	INT	OP1_Ch2_Analog_Input_Value		Global Variables
Option Board	NX1W-MAB221						
	Ch1 Analog Input Value	Value of analog input 1	R	INT	OP2_Ch1_Analog_Input_Value		Global Variables
	Ch2 Analog Input Value	Value of analog input 2	R	INT	OP2_Ch2_Analog_Input_Value		Global Variables
	Ch1 Analog Output Value	Value of analog output 1	RW	INT	OP2_Ch1_Analog_Output_Value		Global Variables
	Ch2 Analog Output Value	Value of analog output 2	RW	INT	OP2_Ch2_Analog_Output_Value		Global Variables
NXBusMaster	NX Bus Master						

5-4 Programming

The following provides information on programming for Analog I/O Option Boards.

5-4-1 I/O Data

The tables below show the I/O data available for Analog I/O Option Boards.

Refer to 5-1-4 *Input Range and Output Range* on page 5-3 for the range of I/O data values.

● NX1W-ADB21

Data name	Function	Data type	I/O port name
Ch1 Analog Input Value	Value of analog input 1	INT	Ch1 Analog Input Value
Ch2 Analog Input Value	Value of analog input 2	INT	Ch2 Analog Input Value

● NX1W-DAB21V

Data name	Function	Data type	I/O port name
Ch1 Analog Output Value	Value of analog output 1	INT	Ch1 Analog Output Value
Ch2 Analog Output Value	Value of analog output 2	INT	Ch2 Analog Output Value

● NX1W-MAB221

Data name	Function	Data type	I/O port name
Ch1 Analog Input Value	Value of analog input 1	INT	Ch1 Analog Input Value
Ch2 Analog Input Value	Value of analog input 2	INT	Ch2 Analog Input Value
Ch1 Analog Output Value	Value of analog output 1	INT	Ch1 Analog Output Value
Ch2 Analog Output Value	Value of analog output 2	INT	Ch2 Analog Output Value

For Analog I/O Option Boards, I/O data is used as I/O ports.

I/O ports are generated automatically by the Sysmac Studio when you specify the Option Board configuration.

To use I/O data in the user program, you use device variables assigned to the relevant I/O ports.

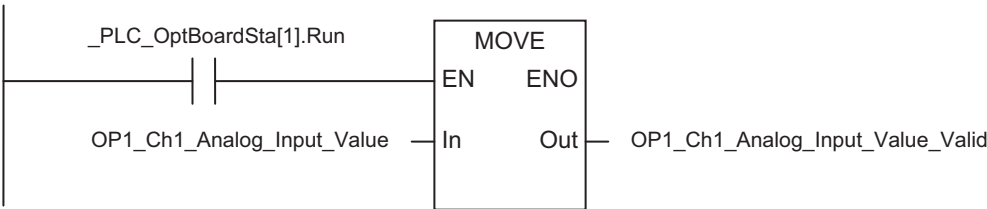
Refer to the *NJ/NX-series CPU Unit Software User's Manual* (Cat. No. W501) for I/O ports and device variables.

Refer to the *Sysmac Studio Version 1 Operation Manual* (Cat. No. W504) for how to register device variables with the Sysmac Studio.

5-4-2 Option Board Status

To use device variables for an Analog I/O Option Board, program the *.Run* (Option Board Normal Operation) member of the *_PLC_OptBoardSta* (Option Board Status) system-defined variable as an interlock condition in the user program.

Example of reading analog input values from Option Board 1 to the CPU Unit using the Option Board Normal Operation as an interlock condition



<code>_PLC_OptBoardSta[1].Run</code>	Option Board Normal Operation of Option Board 1 Status system-defined variable
<code>OP1_Ch1_Analog_Input_Value</code>	Device variable to analog input 1 on Option Board 1
<code>OP1_Ch1_Analog_Input_Value_Valid</code>	Input value to analog input 1 read into the CPU Unit

Refer to the *NJ/NX-series CPU Unit Software User's Manual* (Cat. No. W501) for the specifications of the *_PLC_OptBoardSta* (Option Board Status) system-defined variable.

5-4-3 Special Instructions for Analog I/O Option Boards

No special instruction is available for Analog I/O Option Boards.

5-4-4 Precautions on Supported Functions

There are functions that are provided on NX-series Analog I/O Units, but not available for Analog I/O Option Boards.

The tables below show what you should do to use these functions.

Analog Input Related Functions

Function	Action
Synchronous I/O refreshing method	The same operation is not possible.
Selecting channel to use	Refer to <i>5-3-1 Option Board Settings</i> on page 5-6 for what you should do for unused input terminals.
Moving average	Use the MovingAverage (Moving Average) instruction to perform similar operations.
Input disconnection detection	There is no input range that requires this function.
Over range/under range detection	Use the ZoneCmp (Zone Comparison) instruction to perform similar operations.
User calibration	Use the PWLApprox or PWLApproxNoLineChk (Broken Line Approximation) instruction to perform similar operations.

Analog Output Related Functions

Function	Action
Synchronous I/O refreshing method	The same operation is not possible.
Selecting channel to use	Refer to <i>5-3-1 Option Board Settings</i> on page 5-6 for what you should do for unused input terminals.
Output load rejection setting	The Unit cannot perform operations to output the output set value specified at load rejection.
Over range/under range detection	Use the ZoneCmp (Zone Comparison) instruction to perform similar operations.
User calibration	Use the PWLApprox or PWLApproxNoLineChk (Broken Line Approximation) instruction to perform similar operations.

5-5 Wiring

Refer to the *NX-series NX1P2 CPU Unit Hardware User's Manual* (Cat. No. W578) for information on wiring Analog I/O Option Boards.

5-6 I/O Refreshing

The following describes the I/O refresh operation of Analog I/O Option Boards.

5-6-1 I/O Refresh Operation

I/O refreshing between the CPU Unit and Analog Option Boards is carried out via the option board service. The CPU Unit executes the option board service according to the task execution priority, if it receives a request from an Option Board.

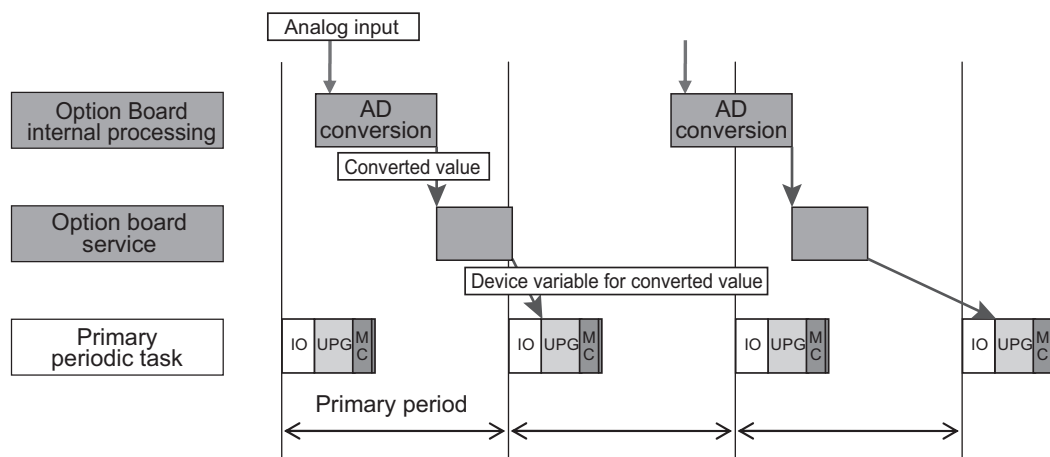
In addition, the internal processing of an Option Board, option board service, and task execution are performed asynchronously.

Therefore, the input and output response time varies depending on the processing time of each processing stage.

Analog Input

Converted values are read during the internal processing of an Option Board and passed to the next option board service for subsequent processing.

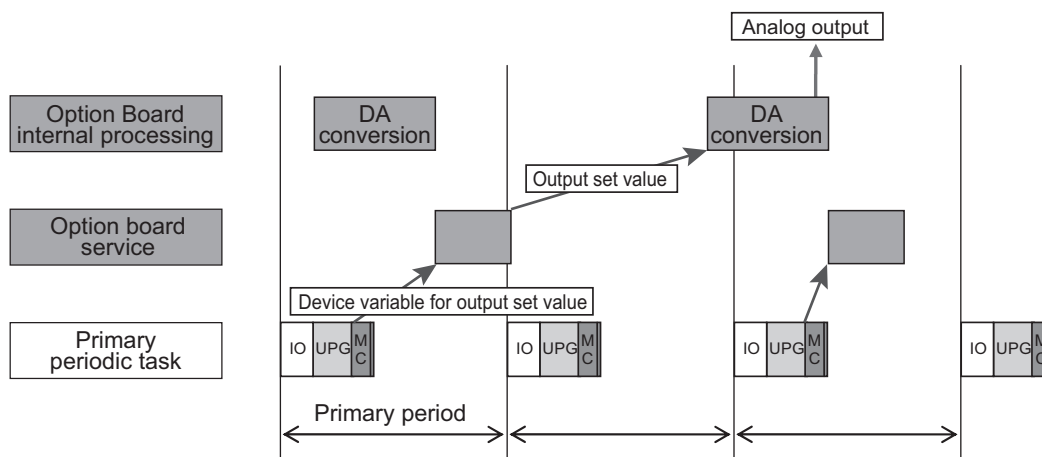
When the processing in the option board service is completed, the new converted values are available for use in user program execution in the next primary periodic task.



Analog Output

New output set values are determined when the user program is executed. The output set values are then processed in the next option board service.

When the processing in the option board service is completed, the processed values are passed to the next internal processing of the Option Board to generate analog output with the new output set values.



5-6-2 Response Time

The input and output response time varies depending on the internal processing of the Option Board, option board service, and processing time of each task execution.

The reference values of input and output response time are given below.

Model	Response time*1	
	Min.	Max.
NX1W-ADB21	2.8 ms	32.0 ms
NX1W-DAB21V	3.0 ms	24.0 ms
NX1W-MAB221 (Input)	2.8 ms	43.0 ms
NX1W-MAB221 (Output)	3.0 ms	38.0 ms

*1. These values are provided for reference only. They are not intended to guarantee the I/O response performance of each model.

Here, the input response time refers to the time from when the voltage or current value changes at an input terminal until the change is reflected on the device variable for the converted value.

The output response time refers to the time from when the device variable for the output set value changes until the change is reflected on the value at the output terminal.

6

Introduction of Motion Control Functions

This section describes the motion control functions that are used when the NX1P2 CPU Unit is connected to an OMRON 1S-series Servo Drive with built-in EtherCAT communications.

6-1	Single-axis Position Control	6-3
6-1-1	Outline of Operation	6-3
6-1-2	Absolute Positioning	6-4
6-1-3	Relative Positioning	6-4
6-1-4	Interrupt Feeding	6-5
6-1-5	Cyclic Synchronous Positioning	6-6
6-1-6	Stopping	6-7
6-1-7	Override Factors	6-13
6-2	Single-axis Synchronized Control	6-14
6-2-1	Overview of Synchronized Control	6-14
6-2-2	Gear Operation	6-14
6-2-3	Positioning Gear Operation	6-15
6-2-4	Cam Operation	6-16
6-2-5	Cam Tables	6-17
6-2-6	Synchronous Positioning	6-25
6-2-7	Combining Axes	6-26
6-2-8	Master Axis Phase Shift	6-27
6-2-9	Slave Axis Position Compensation	6-27
6-2-10	Achieving Synchronized Control in Multi-motion	6-28
6-3	Single-axis Velocity Control	6-30
6-3-1	Velocity Control	6-30
6-3-2	Cyclic Synchronous Velocity Control	6-31
6-4	Single-axis Torque Control	6-32
6-5	Common Functions for Single-axis Control	6-33
6-5-1	Positions	6-33
6-5-2	Velocity	6-35
6-5-3	Acceleration and Deceleration	6-36
6-5-4	Jerk	6-38
6-5-5	Specifying the Operation Direction	6-39
6-5-6	Re-executing Motion Control Instructions	6-43
6-5-7	Multi-execution of Motion Control Instructions (Buffer Mode)	6-48

6-6	Multi-axes Coordinated Control	6-54
6-6-1	Outline of Operation	6-54
6-6-2	Linear Interpolation	6-57
6-6-3	Circular Interpolation	6-58
6-6-4	Axes Group Cyclic Synchronous Positioning	6-58
6-6-5	Stopping Under Multi-axes Coordinated Control	6-59
6-6-6	Overrides for Multi-axes Coordinated Control	6-61
6-7	Common Functions for Multi-axes Coordinated Control	6-62
6-7-1	Velocity Under Multi-axes Coordinated Control	6-62
6-7-2	Acceleration and Deceleration Under Multi-axes Coordinated Control	6-63
6-7-3	Jerk for Multi-axes Coordinated Control	6-64
6-7-4	Re-executing Motion Control Instructions for Multi-axes Coordinated Control	6-65
6-7-5	Multi-execution (Buffer Mode) of Motion Control Instructions for Multi-axes Coordinated Control	6-66
6-8	Other Functions	6-74
6-8-1	Changing the Current Position	6-74
6-8-2	Torque Limit	6-75
6-8-3	Latching	6-75
6-8-4	Zone Monitoring	6-76
6-8-5	Software Limits	6-77
6-8-6	Following Error Monitoring	6-78
6-8-7	Following Error Counter Reset	6-79
6-8-8	Axis Following Error Monitoring	6-80
6-8-9	In-position Check	6-80
6-8-10	Changing Axis Use	6-82
6-8-11	Enabling Digital Cam Switch	6-83
6-8-12	Displaying 3D Motion Monitor for User Coordinate System	6-84

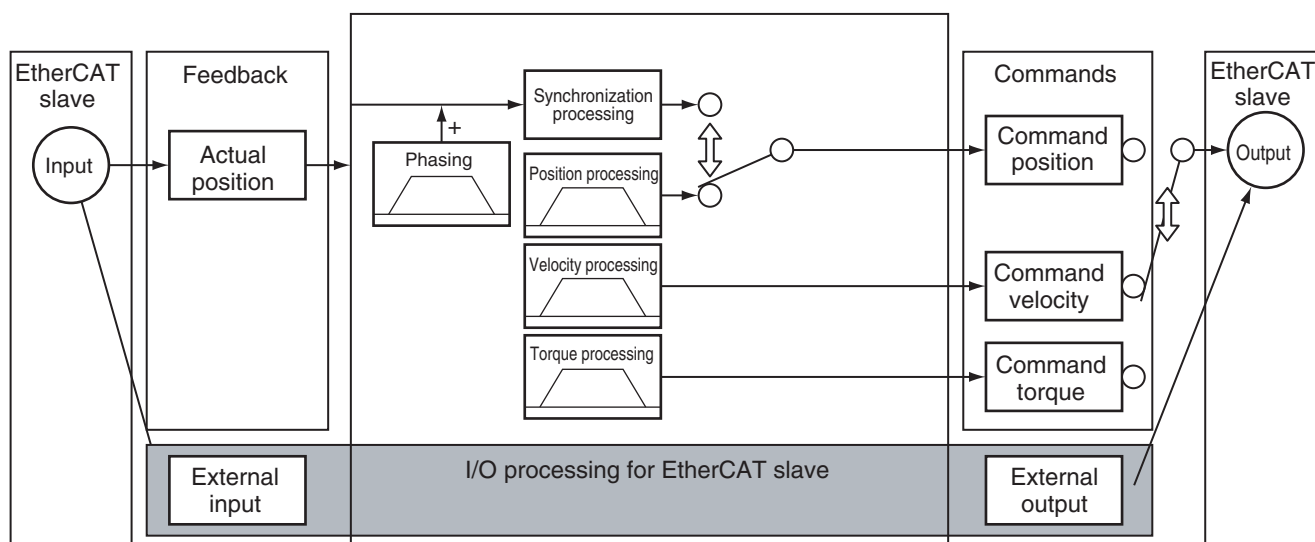
6-1 Single-axis Position Control

The MC Function Module can be connected to OMRON 1S-series Servo Drives with built-in EtherCAT communications or G5-series Servo Drives with built-in EtherCAT communications to implement position control, velocity control, and torque control. This section describes positioning operation for single axes.

Some of the functions of the MC Function Module are different when NX-series Pulse Output Units are used. Refer to the *NX-series Position Interface Units User's Manual* (Cat. No. W524) for details.

6-1-1 Outline of Operation

The single-axis control function of the MC Function Module consists of control for motion profile commands and synchronized control. There are three Control Modes for motion profile commands: position control, velocity control, and torque control. In synchronized control, the slave axis (i.e., the axis being controlled) operates in a synchronized relationship to the master axis, as expressed by a cam profile curve or a gear ratio. Manual operations such as jogging and homing are also supported.



Note You can use the command position or actual position as the input to the synchronization processing.

Resetting Axis Errors

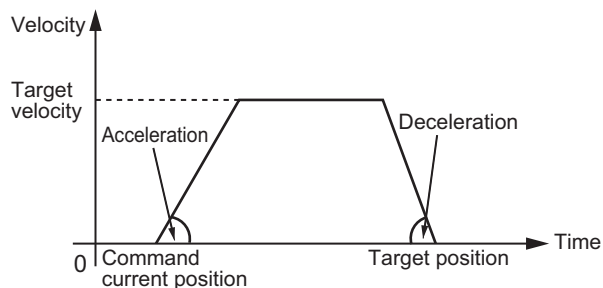
If an error occurs in an axis, you can use the MC_Reset instruction to remove the error once you have eliminated the cause.

For details on resetting axis errors, refer to the MC_Reset (Reset Axis Error) instruction in the *NJ/NX-series Motion Control Instructions Reference Manual* (Cat. No. W508).

Refer to the *NX-series Position Interface Units User's Manual* (Cat. No. W524) for the differences when you use NX-series Pulse Output Units.

6-1-2 Absolute Positioning

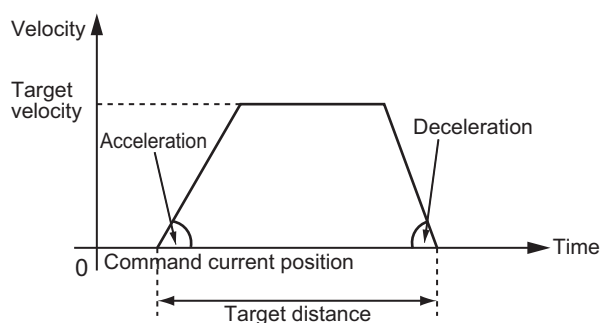
Absolute positioning specifies the absolute coordinates of the target position in relation to home. You can perform positioning, such as shortest way positioning on a rotary table, by setting the Count Mode to Rotary Mode and specifying the operation direction.



For details, refer to the MC_MoveAbsolute (Absolute Positioning) and MC_Move (Positioning) instructions in the *NJ/NX-series Motion Control Instructions Reference Manual* (Cat. No. W508).

6-1-3 Relative Positioning

Relative positioning specifies the distance from the actual position. You can specify a travel distance that exceeds the ring counter range by setting the Count Mode to Rotary Mode.



For details, refer to the MC_MoveRelative (Relative Positioning) and MC_Move (Positioning) instructions in the *NJ/NX-series Motion Control Instructions Reference Manual* (Cat. No. W508).

6-1-4 Interrupt Feeding

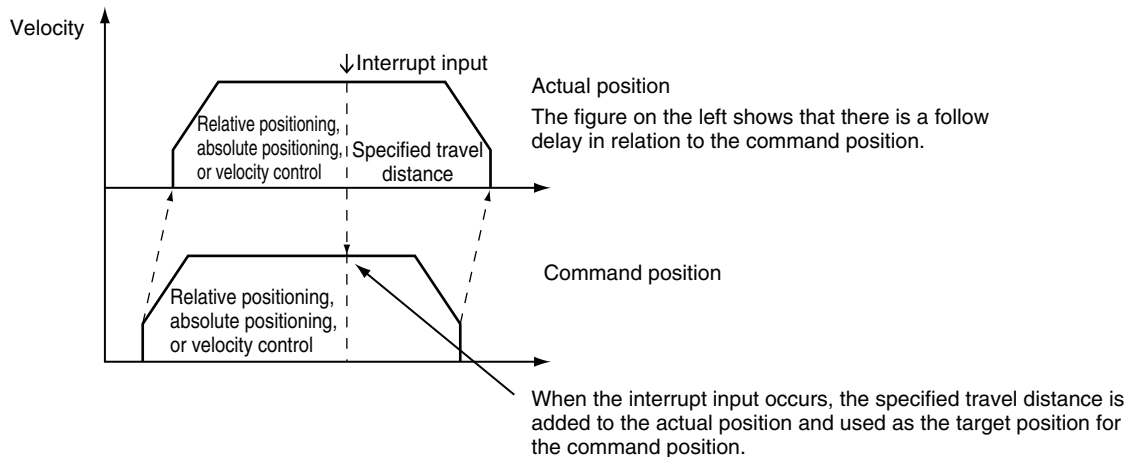
Interrupt feeding feeds the axis at the specified velocity and for the specified distance from the actual position when a trigger signal occurs.

You can also select to output an error if the trigger signal does not occur within the specified travel distance when you specify either absolute or relative positioning.

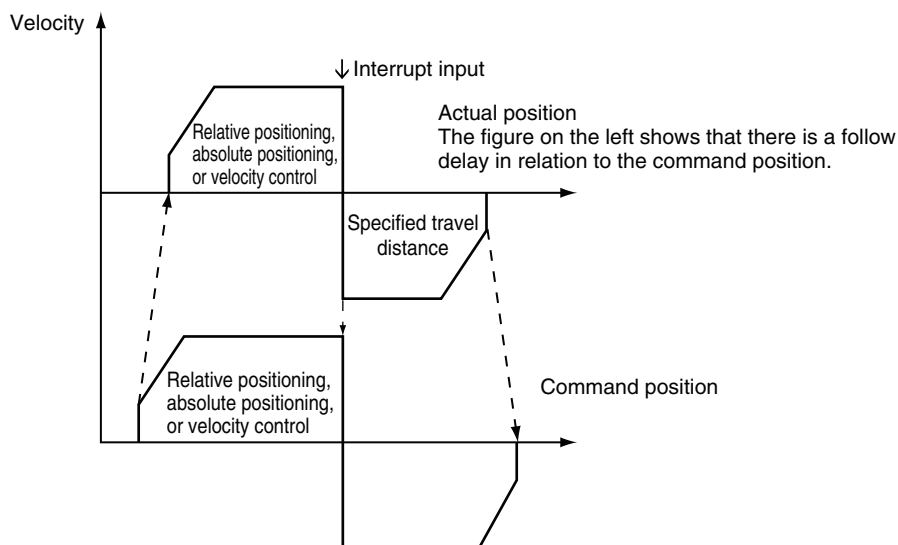
Feeding is not affected by following error. This is achieved by using the latch function of the Servo Drive to determine the actual position when the trigger signal occurs. You can also use the window function to disable trigger signals that occur outside of a specified position range. For applications such as wrapping machines, this enables feeding only on trigger signals for printed marks on films and eliminates other influences.

Motion Relative to the Actual Position

● Feeding for a Specified Distance in the Moving Direction



● Feeding for a Specified Distance in the Direction Opposite to the Moving Direction



If decelerating to a stop after a reverse turn is specified for the Operation Selection at Reversing axis parameter, an acceleration/deceleration curve is used when reversing.

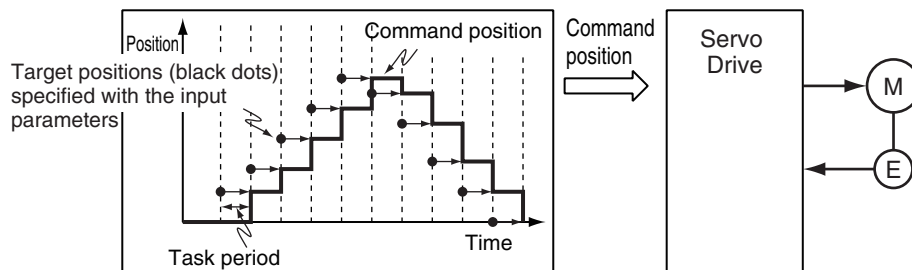
For details, refer to the MC_MoveFeed (Interrupt Feeding) instruction in the *NJ/NX-series Motion Control Instructions Reference Manual* (Cat. No. W508).

Refer to the *NX-series Position Interface Units User's Manual* (Cat. No. W524) for the differences when you use NX-series Pulse Output Units.

6-1-5 Cyclic Synchronous Positioning

Cyclic synchronous positioning is used to output a target position to a specified axis each control period in the primary periodic task or a periodic task. The target position is specified as an absolute position.

You can use it to move in a specific path that you create.



For details, refer to the MC_SyncMoveAbsolute (Cyclic Synchronous Absolute Positioning) instruction in the *NJ/NX-series Motion Control Instructions Reference Manual* (Cat. No. W508).

6-1-6 Stopping

Functions to stop axis operation include immediate stop input signal and limit input signals connected to the Servo Drive, stop functions of motion control instructions in the user program, and stopping due to errors.

Refer to the *NX-series Position Interface Units User's Manual* (Cat. No. W524) for the differences when you use NX-series Pulse Output Units.

Stopping for Servo Drive Input Signals

Axis motion is stopped for the immediate stop input signal or a limit input signal from the Servo Drive. You can select the stop method with the Sysmac Studio.

● Immediate Stop Input

Stop processing in the MC Function Module is executed according to the state of the Servo Drive input signals. You can select one of the following stopping methods for the MC Function Module.

- Immediate stop
- Immediate stop and error reset
- Immediate stop and Servo OFF



Precautions for Correct Use

The immediate stop input for the OMRON 1S-series Servo Drive or G5-series Servo Drive also causes an error and executes stop processes in the Servo Drive itself.

● Limit Inputs (Positive Limit Input or Negative Limit Input)

Stop processing in the MC Function Module is executed according to the state of the Servo Drive input signals. You can select one of the following stopping methods for the MC Function Module.

- Immediate stop
- Deceleration stop
- Immediate stop and error reset
- Immediate stop and Servo OFF



Precautions for Correct Use

- If a limit input signal turns ON, do not execute an instruction for axis command of the axis in the same direction as the limit input signal.
- If a limit input signal is ON for any axis in an axes group, do not execute an instruction for an axes group command for that axes group.
- If the signal to decelerate to a stop is input during execution of a synchronous movement instruction that has a *Deceleration* input variable, the axis decelerates to a stop at the deceleration rate given by *Deceleration*.
- If the signal to decelerate to a stop is input during execution of a synchronous movement instruction that does not have a *Deceleration* input variable, the axis decelerates to a stop at the maximum deceleration rate that is set in the axis parameters.



Additional Information

- You must set up the Servo Drive in order to use the input signals from the Servo Drive. An OMRON 1S-series Servo Drive with built-in EtherCAT communications or G5-series Servo Drive with built-in EtherCAT communications has an immediate stop input and limit input assigned in its default settings.
- Refer to the *NJ/NX-series CPU Unit Motion Control User's Manual* (Cat. No. W507) for setting examples for connection to an OMRON 1S-series Servo Drive.
- Refer to the *NJ/NX-series CPU Unit Motion Control User's Manual* (Cat. No. W507) for setting examples for connection to an OMRON G5-series Servo Drive.

Stopping with Motion Control Instructions

Use the MC_Stop or MC_ImmediateStop instruction to stop single-axis operation.

● MC_Stop Instruction

You can specify the deceleration rate and jerk for single-axis control and synchronized control to decelerate to a stop. Specify a deceleration rate of 0 to send a command that immediately stops the Servo Drive. Other operation commands are not acknowledged while decelerating to a stop for this instruction and while the input variable *Execute* is TRUE.

● MC_ImmediateStop Instruction

You can perform an immediate stop for single-axis control or synchronized control functions. You can also execute this instruction on axes that are enabled in an axes group.

For details, refer to the MC_Stop and MC_ImmediateStop instructions in the *NJ/NX-series Motion Control Instructions Reference Manual* (Cat. No. W508).



Additional Information

When the input variable *Enable* to the MC_Power (Servo ON) instruction changes to FALSE, the MC Function Module immediately stops the command value and turns OFF the Servo. When the Servo is turned OFF, the Servo Drive will operate according to the settings in the Servo Drive.

Stopping Due to Errors or Other Problems

● Stopping for Errors during Single-axis Operation

When an error occurs during single-axis operation, the axis will stop immediately or decelerate to a stop depending on the error. Refer to the *NJ/NX-series CPU Unit Motion Control User's Manual* (Cat. No. W507) for details on the stop method for each error.

● Stopping for a Software Limit

To stop for a software limit, set the Software Limits axis parameter. You can select from the following stop methods for the software limits.

- Enabled for command position. Decelerate to a stop.
- Enabled for command position. Immediate stop.
- Enabled for actual position. Decelerate to a stop.
- Enabled for actual position. Immediate stop.

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

● Stopping Due to Motion Control Period Exceeded Error

If motion control processing does not end within two periods, a Motion Control Period Exceeded error occurs. All axes stop immediately.



Precautions for Correct Use

When you use an NX701 CPU Unit and operate in the multi-motion, all axes in both tasks will stop immediately if a Motion Control Period Exceeded error occurs in either of the tasks. Refer to the *NJ/NX-series CPU Unit Motion Control User's Manual* (Cat. No. W507) for multi-motion.

● Errors That Cause the Servo to Turn OFF

An immediate stop is performed if an error occurs that causes the Servo to turn OFF. When the Servo is turned OFF, the Servo Drive will operate according to the settings in the Servo Drive.

● Stopping Due to Start of MC Test Run

All axes will decelerate to a stop at their maximum deceleration if a MC Test Run is started from the Sysmac Studio.

● Stopping Due to End of MC Test Run

All axes will decelerate to a stop at their maximum deceleration if a MC Test Run is stopped from the Sysmac Studio.

- Click the **Stop MC Test Run** Button on the MC Test Run Tab Page of the Sysmac Studio.
- Close the MC Test Run Tab Page on the Sysmac Studio.
- Exit the Sysmac Studio.

● Stopping Due to Change in CPU Unit Operating Mode

All axes will decelerate to a stop at their maximum deceleration if the CPU Unit operating mode changes.



Precautions for Correct Use

- If an error that results in deceleration to a stop occurs during execution of a synchronous movement instruction that has a *Deceleration* input variable, the axis decelerates to a stop at the deceleration rate given by *Deceleration*.
- If an error that results in deceleration to a stop occurs during execution of a synchronous movement instruction that does not have a *Deceleration* input variable, the axis decelerates to a stop at the maximum deceleration rate that is set in the axis parameters.

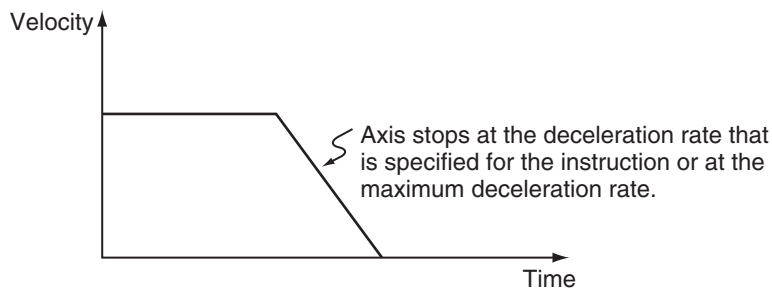


Additional Information

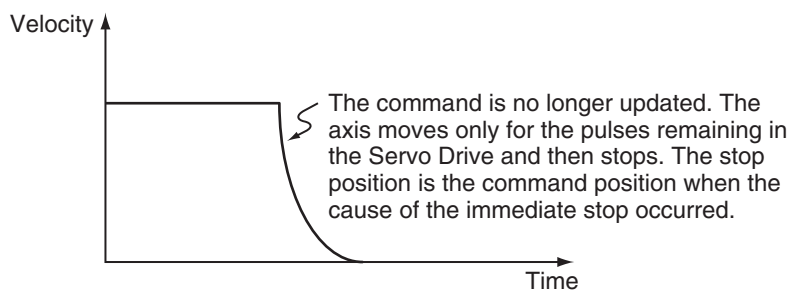
- When RUN mode changes to PROGRAM mode, any motion control instructions for current motions are aborted. The *CommandAborted* output variable from the instructions remains FALSE. The Servo remains ON even after changing to PROGRAM mode.
 - If the operating mode returns to RUN mode while a deceleration stop is in progress after the operating mode changes from RUN to PROGRAM mode, the output variables from motion control instructions are cleared. The *CommandAborted* output variables from the motion control instructions therefore remain FALSE.
 - The save process will continue during a save for the MC_SaveCamTable Instruction.
 - The generation process will continue when generation of the cam table is in progress for the MC_GenerateCamTable (Generate Cam Table) instruction.
-

Stop Method

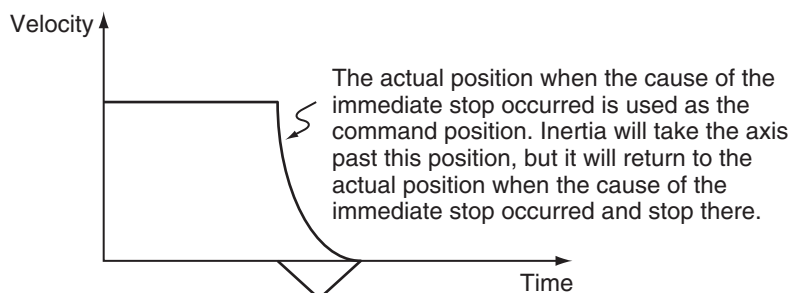
● Deceleration Stop



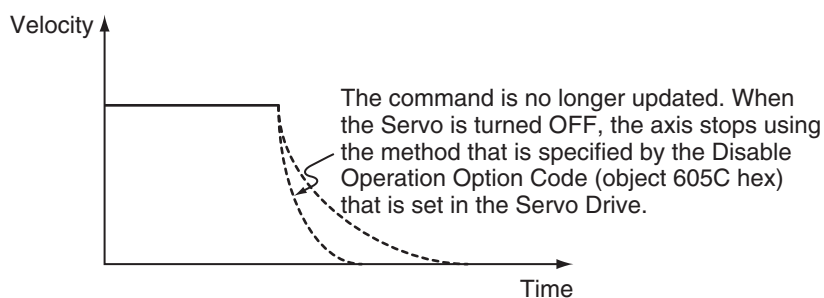
● Immediate Stop



● Immediate Stop and Error Reset



● Immediate Stop and Servo OFF



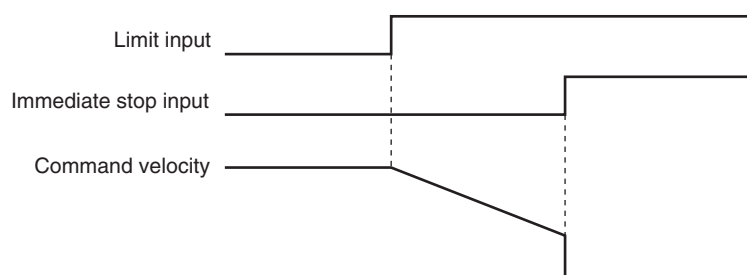
Stop Priorities

The priorities for each stop method are listed in the following table. If a stop with a higher priority stop method occurs while stopping, the stop method will switch to the higher priority method.

Stop method	Priority (higher numbers mean higher priority)
Immediate stop and Servo OFF	4
Immediate stop and error reset	3
Immediate stop	2
Deceleration stop	1

● Example

The following figure is an example of an immediate stop when the limit input signal is ON and the immediate stop input changes to ON during a deceleration to a stop.



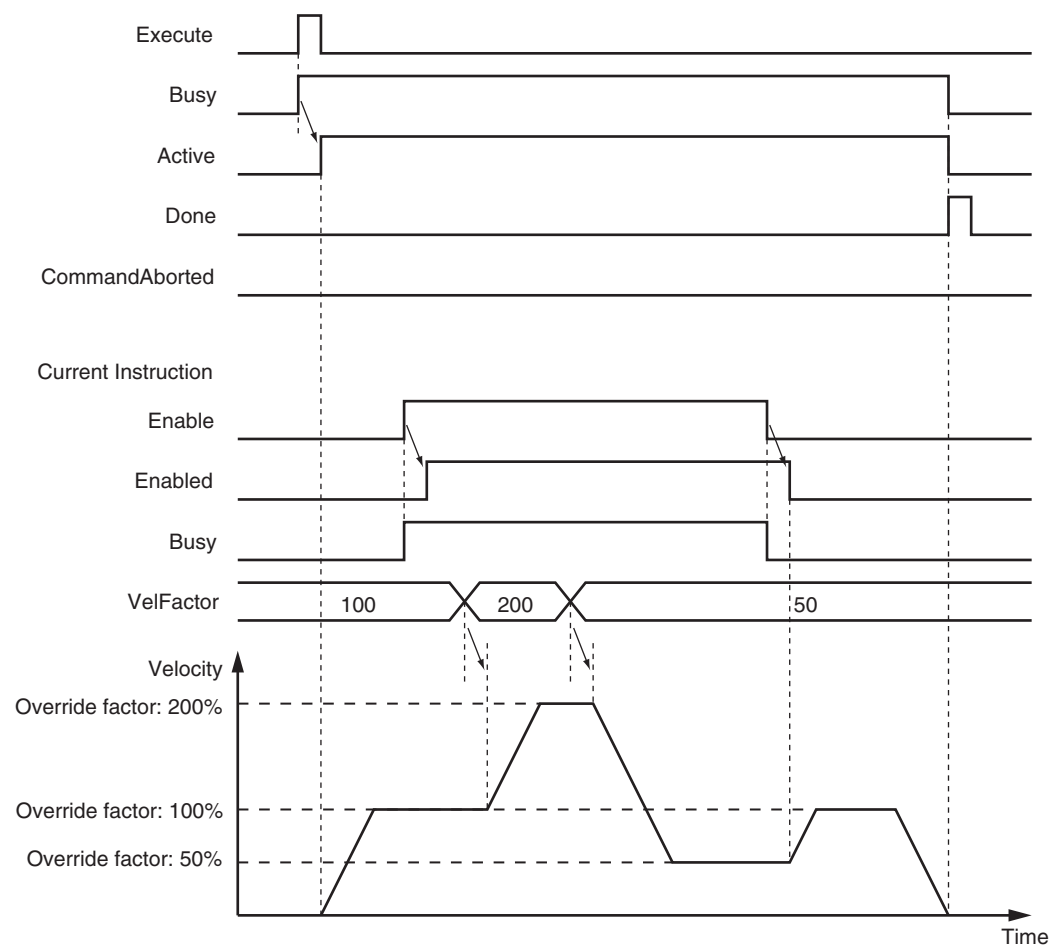
6-1-7 Override Factors

You can use the MC_SetOverride instruction to set override factors for the motion of the axes that are currently in motion. The velocity override factor is set as a percentage of the target velocity. It can be set between 0% and 500%. If an override factor of 0% is set for the target velocity, operating status will continue with the axis stopped as a velocity of 0. The set override factor is read as long as the overrides are enabled. If the overrides are disabled, the override factors return to 100%. If the maximum velocity is exceeded when an override factor is changed, the maximum velocity for the axis is used.

● Overriding the MC_MoveAbsolute Instruction

An example of a time chart for using the Set Override Factors instruction for the MC_MoveAbsolute (Absolute Positioning) instruction is given below.

Previous Instruction: MC_MoveAbsolute



For details, refer to the MC_SetOverride (Set Override Factors) instruction in the *NJ/NX-series Motion Control Instructions Reference Manual* (Cat. No. W508).

6-2 Single-axis Synchronized Control

This section describes the operation of synchronized control for single axes.

6-2-1 Overview of Synchronized Control

Synchronous control synchronizes the position of a slave axis with the position of a master axis. The command position or actual position of any axis can be specified for the master axis. If the command velocity for the slave axis exceeds the maximum velocity that is set in the axis parameters, the command is performed at the maximum velocity of the axis. If this occurs, any insufficient travel distance is distributed and output in the following periods.



Precautions for Correct Use

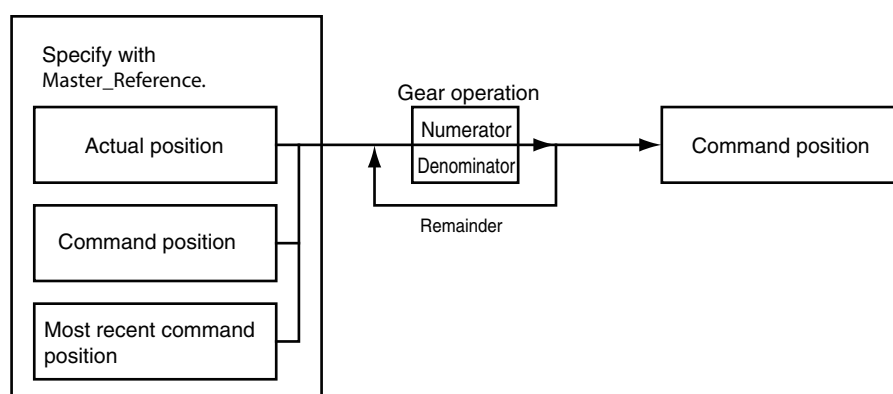
- You cannot specify an encoder axis, virtual encoder axis or single-axis position control axis for the slave axis.
- When you use an NX701 CPU Unit and operate in the multi-motion, assign the master axis and slave axis to the same task.

If you specify the master axis in a different task from the slave axis by executing the synchronized control instructions such as the MC_GearIn (Start Gear Operation) instruction or the MC_Camin (Start Cam Operation) instruction, an Illegal Master Axis Specification (event code: 54620000 hex) occurs.

Refer to the *NJ/NX-series CPU Unit Motion Control User's Manual* (Cat. No. W507) if you desire to specify the master axis in a different task from the slave axis.

6-2-2 Gear Operation

This function specifies the gear ratio between the master axis and the slave axis and starts operation. Start gear operation with the MC_GearIn (Start Gear Operation) instruction. End synchronization with the MC_GearOut (End Gear Operation) instruction or the MC_Stop instruction.



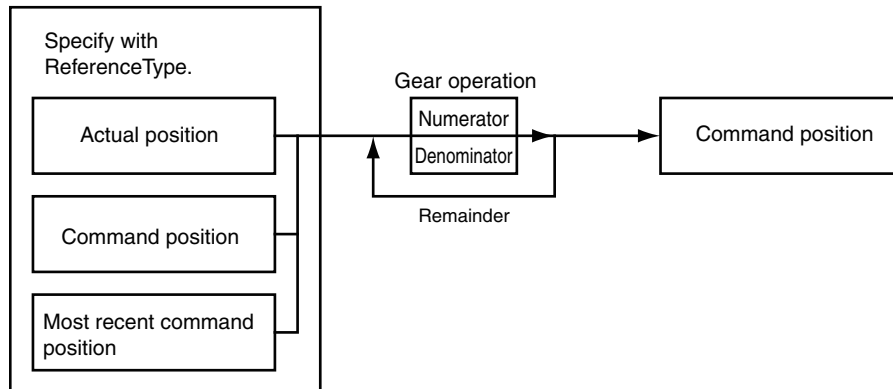
You can set the gear ratio numerator, gear ratio denominator, position type, acceleration rate, and deceleration rate for the slave axis to operate. For the master axis, you can specify the command position, actual position, or most recent command position.

After operation starts, the slave axis uses the velocity of the master axis times the gear ratio for its target velocity, and accelerates/decelerates accordingly. The catching phase exists until the target velocity is reached. The *InGear* phase exists after that. If the gear ratio is positive, the slave axis and master axis move in the same direction. If the gear ratio is negative, the slave axis and master axis move in the opposite directions.

For details on gear operation, refer to the MC_GearIn (Start Gear Operation), MC_GearOut (End Gear Operation), and MC_Stop instructions in the *NJ/NX-series Motion Control Instructions Reference Manual* (Cat. No. W508).

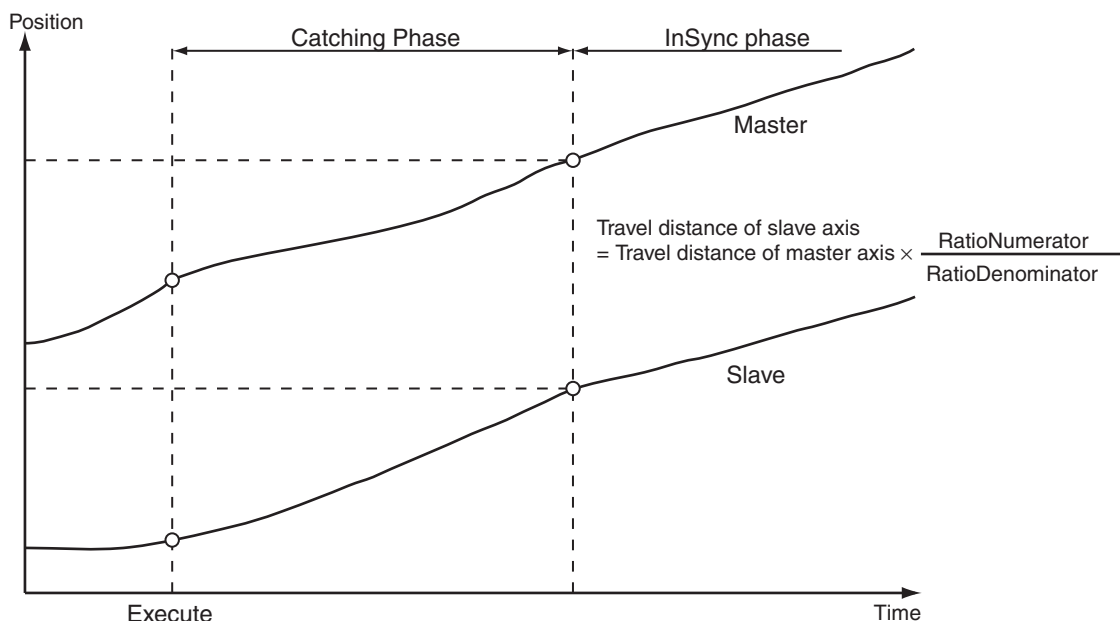
6-2-3 Positioning Gear Operation

This function specifies the gear ratio between the master axis and the slave axis and starts operation. Positioning gear operation allows you to set the positions of the master and slave axes at which to start synchronization. Start positioning gear operation with the MC_GearInPos instruction. End synchronization with the MC_GearOut instruction or the MC_Stop instruction.



You can set the gear ratio numerator, gear ratio denominator, position type, acceleration rate, and deceleration rate for the slave axis to operate. For the master axis, you can specify the command position, actual position, or most recent command position.

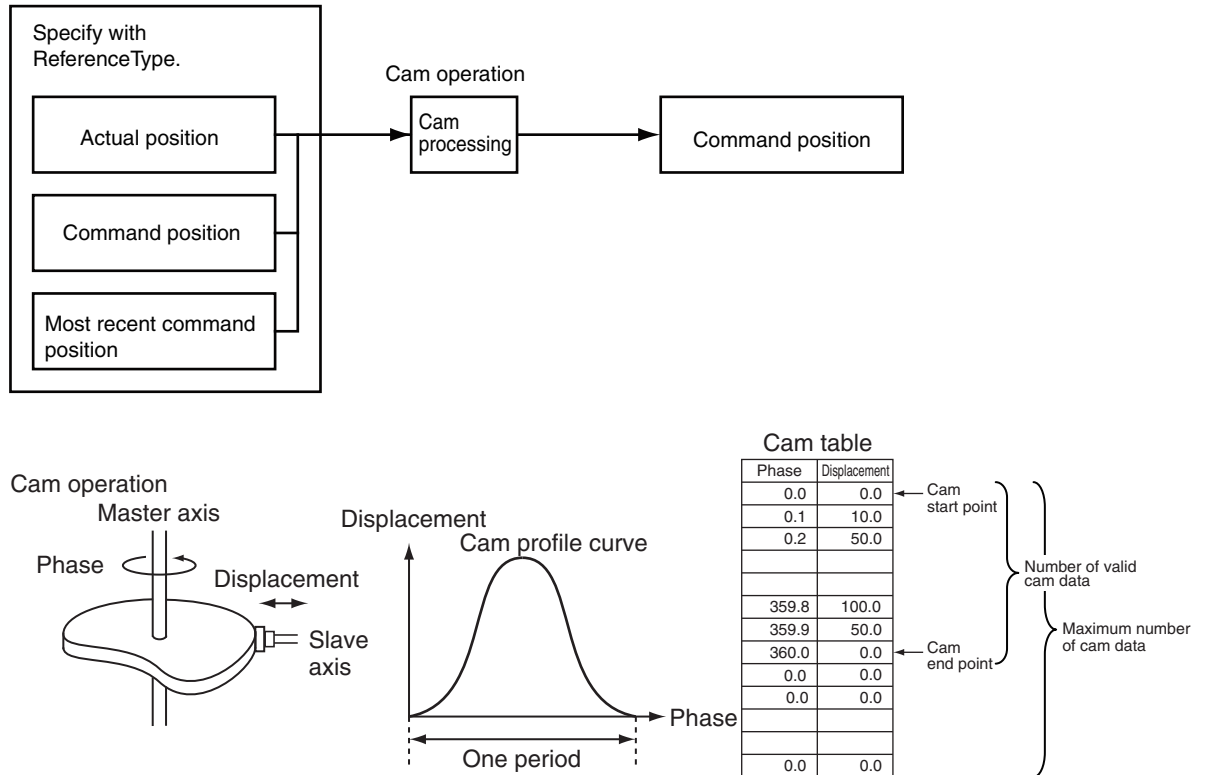
After operation starts, the slave axis uses the velocity of the master axis times the gear ratio for its target velocity, and accelerates/decelerates accordingly. The slave axis is in the catching phase until it reaches the slave sync position. The slave axis enters the *InSync* phase after it reaches the slave sync position. For either, the position of the slave axis is synchronized with the master axis. If the gear ratio is positive, the slave axis and master axis move in the same direction. If the gear ratio is negative, the slave axis and master axis move in the opposite directions. The following figure shows the operation when the gear ratio is positive.



For details on positioning gear operation, refer to the MC_GearInPos (Positioning Gear Operation), the MC_GearOut (End Gear Operation), and the MC_Stop instructions in the *NJ/NX-series Motion Control Instructions Reference Manual* (Cat. No. W508).

6-2-4 Cam Operation

Cam operation synchronizes the position of the slave axis with the master axis according to a cam table. Start cam operation with the MC_CamIn (Start Cam Operation) instruction. End cam operation with the MC_CamOut (End Cam Operation) instruction or the MC_Stop instruction. Create a cam table using the Cam Editor in the Sysmac Studio and download it to the CPU Unit. Use the Synchronization menu command of the Sysmac Studio to download the project to the CPU Unit.



Also, the following operation is possible: if another MC_CamIn (Start Cam Operation) instruction is executed by using multi-execution with the Buffer Mode set for blending while the current MC_CamIn (Start Cam Operation) instruction is executed, the operation can continue using the switched cam table and the slave axis does not stop.

For details on cam operation, refer to the MC_CamIn (Start Cam Operation), MC_CamOut (End Cam Operation), and MC_Stop instructions in the *NJ/NX-series Motion Control Instructions Reference Manual* (Cat. No. W508).

For details on the Cam Editor, refer to the *Sysmac Studio Version 1 Operation Manual* (Cat. No. W504).

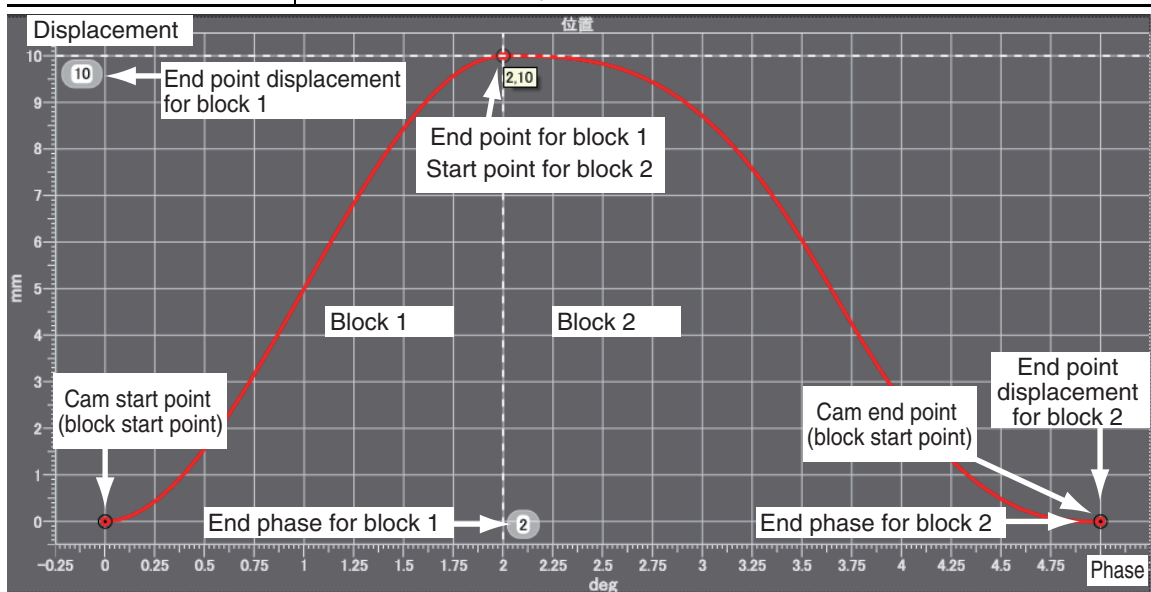
6-2-5 Cam Tables

This section describes the cam tables that are used for cam operation.

Cam Table Terminology

Term	Description
cam operation	An operation that takes one master axis and one slave axis and follows the cam profile curve to derive the displacement of the slave axis from the phase of the master axis.
cam profile curve	A curve that shows the relationship between phases and displacements in a cam operation. The cam profile curve is created on the Sysmac Studio. You can use the cam profile curve with a cam data variable after the cam profile curve is downloaded to the CPU Unit. Use the Synchronization menu command of the Sysmac Studio to download the project to the CPU Unit.
cam block	You can select a cam curve in this block. It represents the area between the end point of the previous cam block and the end point of the current cam block.
cam curve	A curve that represents the cam characteristics. You can select a cam curve for each cam block. The Sysmac Studio calculates the phase widths and displacement widths from the specified points and creates the actual cam profile curve. You can choose from different curves, such as straight line, parabolic, and trapezoid.
cam data	Data made up of phases (master axis) and displacements (slave axis) for cam operation.
cam data variable	A variable that represents the cam data as a structure array.
cam table	A data table that contains cam data. If phase data is not in ascending order the cam table is treated as an illegal cam table.
cam start point	The first point in the cam data.
cam end point	The last point of valid cam data in the cam data. If the cam end point is less than the number of cam data, all phases and displacements after the cam end point will be 0.
cam block start point	The start point for a cam block. It is the same as the cam start point at the start of the cam operation. If the cam profile curve continues, this will be the same as the cam block end point.
cam block end point	The end point for a cam block. It is the same as the cam end point at the end of the cam operation. If the cam profile curve continues, this will be the same as the cam block start point. The cam block end point is defined as (horizontal axis, vertical axis) = (phase end point, displacement end point).
original cam data	Cam data that is created by dividing up the cam profile curve in the Cam Editor.
program-modified cam data	The cam data changed by the user program while the CPU Unit is in operation.
master axis	The axis that serves as the input to the cam operation. You can specify either Linear Mode or Rotary Mode.
slave axis	The axis that serves as the output from the cam operation. You can specify either Linear Mode or Rotary Mode.
phase	The relative distance on the master axis from the start point of the cam table.
displacement	The relative distance on the slave axis from the master following distance.
valid cam data	The cam data other than the cam start point and other than data where the phase is 0.
invalid cam data	The cam data other than the cam start point where the phase is 0.
number of valid cam data	The number of sets of cam data.
maximum number of cam data	The maximum number of sets of cam data that the cam table can contain.
cam data index	The number of the cam data that is executed.
cam table start position	The absolute position of the master axis that corresponds to the cam start point (phase = 0).

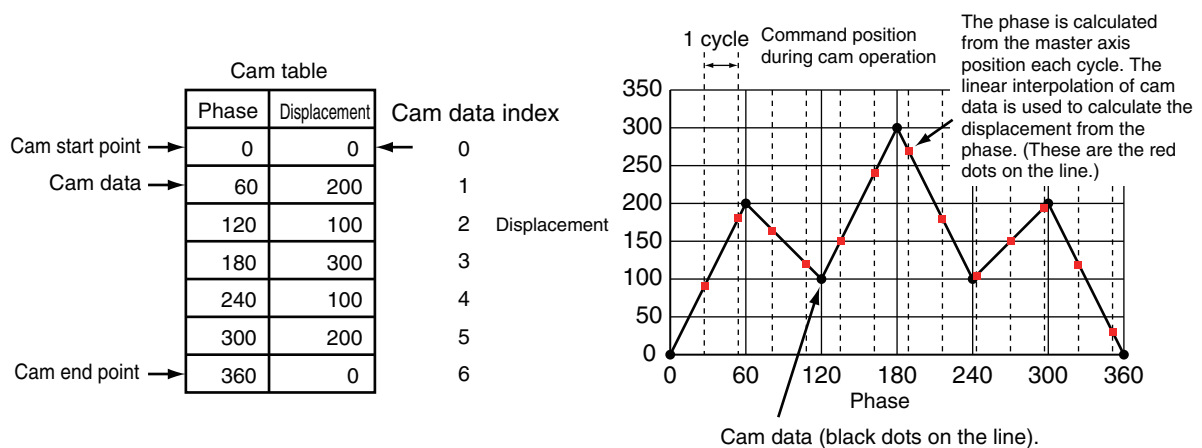
Term	Description
master following distance	The master start distance where the slave axis starts cam operation represented as either an absolute position or relative position. The relative position is based on the cam start point position.
start mode	A specification of whether to represent the master following distance as an absolute position or relative position.
null cam data	Cam data that can be set after the end point where the phase and displacement are 0.
connecting velocity	The connecting velocity that is used to connect cam profile curves. The connecting velocity cannot be specified for some curves.
connecting acceleration	The acceleration rate that is used to connect cam profile curves. The connecting acceleration cannot be specified for some curves.
phase pitch	The width when dividing the cam profile curve by phases (horizontal axis). The points after dividing the curve into the phase pitch correspond to the cam data in the cam table. You must specify the phase pitch for each cam block.



Cam Tables

The MC Function Module defines a single element of data consisting of the phase of the master axis and the displacement of the slave axis as one cam data. A cam table is defined as the combination of multiple sets of cam data. The cam table is created with the Cam Editor in the Sysmac Studio. You can modify cam data in the cam table from the user program.

The phases and displacements in the cam data that makes up the cam table are represented as relative distances from the start point 0.0. During cam operation, the command position sent to the slave axis is the displacement determined by interpolating linearly between the two cam data elements adjacent to the phase of the master axis. The more cam data there is in the cam table, the more accurate the trajectory and the smoother the cam profile curve will be.



Precautions for Correct Use

- Make sure that the cam data is arranged in the cam table so that the phases are in ascending order. An instruction error occurs if a cam operation instruction is executed when the phases are not in ascending order.
- Cam data variables are global variables. You can therefore access or change the values of cam data variables from more than one task. If you change the values of cam data variables from more than one task, program the changes so that there is no competition in writing the value from more than one task.
- If you use exclusive control of global variables between tasks for a cam data variable, do not use the cam data variable for motion control instructions in a task that does not control the variable. An Incorrect Cam Table Specification (event code: 54390000 hex) will occur.

Cam Table Specifications

Item	Description
Maximum number of cam data per cam table	65,535 points
Maximum size of all cam data	1,048,560 points* ¹
Maximum number of cam tables	640 tables* ²
Switching cam operation	You can switch to a different cam operation by executing a motion control instruction
Changing cam data	Cam data can be edited from the user program. Cam data can be overwritten with the Generate Cam Table instruction.
Saving cam data	Cam data can be saved to non-volatile memory by using the Save Cam Table instruction.
Information attached to the cam data	Information can be downloaded or uploaded for display in the Cam Editor* ³

Item	Description
Timing to load cam data to main memory	• When the data is downloaded from the Sysmac Studio • When power is turned ON

- *1. If 65,535 points are used for each cam table, there will be a maximum of 16 cams. A resolution of 0.1° allows for a maximum of 3,600 points per cam table for a maximum of 291 cams.
- *2. The total size is 10 MB max.
- *3. Use the Synchronization menu command of the Sysmac Studio to upload and download the project.

Data Type of Cam Tables

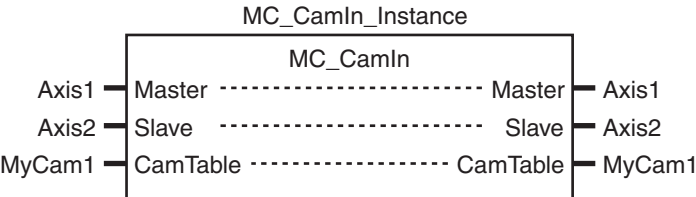
A cam table is declared as an array of cam data structures. The type declaration for the cam data structure is shown below.

```
TYPE
  (*Cam data structure*)
  _sMC_CAM_REF :
  STRUCT
    Phase      : REAL;      (*Phase*)
    Distance   : REAL;      (*Displacement*)
  END_STRUCT;
END_TYPE
```

You must create the cam data with the Cam Editor in the Sysmac Studio and then specify the name of the cam table and the number of cam data (i.e., the size of the array). For example, to make a cam table called *MyCam1* with 1,000 points use the following declaration.

```
VAR
  (*Cam table*)
  MyCam1      :   ARRAY [0..999] OF _sMC_CAM_REF;
END_VAR
```

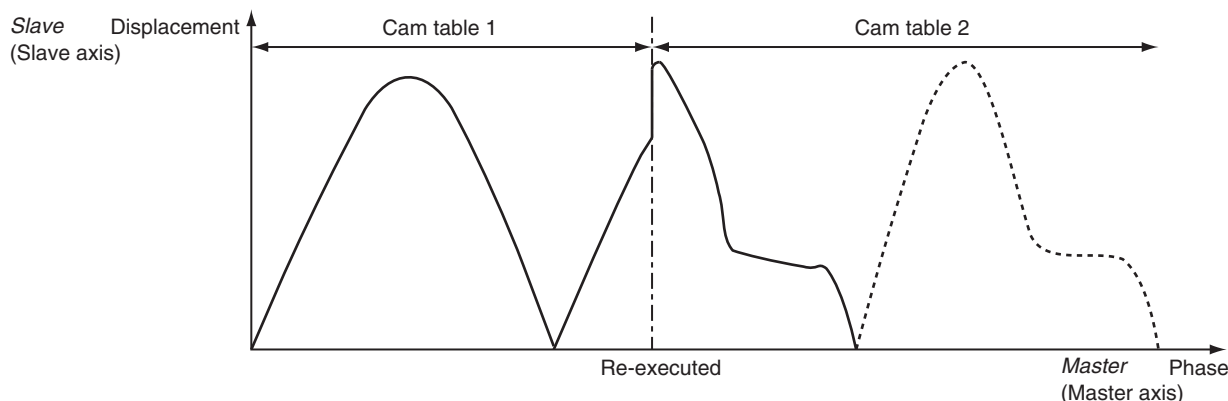
The following notation is used to specify *MyCam1* for a cam operation instruction. In this example, the master axis is *Axis1* and the slave axis is *Axis2*.



An error will occur if the specified cam table does not exist in the Controller. You can also specify the same cam table for more than one axis.

Switching Cam Tables

You can switch cam tables by re-executing the cam operation instruction during cam operation. After switching, cam operation will be performed with the cam table you specified for re-execution of the instruction. The *EndOfProfile* and *Index* output variables from the MC_CamIn instruction are output according to the new cam table.



Precautions for Correct Use

- The cam table you want to switch to must be saved to non-volatile memory before it can be used.
- Switching cam tables during cam operation will cause discontinuous velocities. Adjust the timing for switching the cam table to avoid excessive velocity discontinuity.

Loading/Saving Cam Data and Saving Cam Tables

Cam data can be loaded and saved from the user program just like any other variables. For example, you can use *MyCam1[0].Phase* to specify the phase and *MyCam1[0].Distance* to specify the displacement in the first array elements of a cam table named MyCam1. Cam data overwritten from the user program can be saved to the non-volatile memory in the CPU Unit as a cam table by executing the MC_SaveCamTable instruction.



Precautions for Correct Use

- Overwritten cam data will be lost if the CPU Unit is turned OFF or the cam data is downloaded from the Sysmac Studio before the Save Cam Table instruction is executed or if the instruction fails to save the data for any reason.
- Be careful not to lose the overwritten data when overwriting cam data from the user program in the CPU Unit.
- Cam data saved to non-volatile memory can be loaded by using the upload function of the Sysmac Studio.
- Use the Synchronization menu command of the Sysmac Studio to upload and download the project.

For details on arrays, refer to the *NJ/NX-series CPU Unit Software User's Manual* (Cat. No. W501).

For details on the Save Cam Table instruction, refer to the MC_SaveCamTable instruction in the *NJ/NX-series Motion Control Instructions Reference Manual* (Cat. No. W508).

Updating Cam Table Properties

The MC Function Module must identify the cam end point of the cam table. If an overwrite is performed from the user program during cam operation and the number of valid cam data changes, you must update the number of valid cam data to the latest value. Use the MC_SetCamTableProperty instruction for this.

The cam end point is the data located one cam data before the first cam data with a phase of 0 after the start point in the cam table. All cam data after phase 0 is detected will be invalid.

For example, refer to the following cam table. The *EndPointIndex* (End Point Index) output variable is 999 and the *MaxDataNumber* (Maximum Number of Cam Data) output variable is 5,000 from the MC_SetCamTableProperty instruction.

Cam data structure array

	Phase	Displacement	
MyCam1 [0]	0	0	Cam start point
.	.	.	
.	.	.	
.	.	.	
MyCam1 [997]	359.8	2	Valid data
MyCam1 [998]	359.9	1	
MyCam1 [999]	360.0	0	
MyCam1 [1000]	0	0	Cam end point
.	.	.	
.	.	.	
.	.	.	
MyCam1 [4999]	0	0	Invalid data

Maximum number of data: 5,000



Precautions for Correct Use

- You cannot change the maximum number of cam data from the user program.
- Execute this instruction after overwriting the cam data in any way that changes the number of valid cam data. If the number of valid cam data is not updated, the cam operation and the operation of the *EndOfProfile* (End of Cam Cycle) of the MC_CamIn instruction may not be as expected.

For details on the Set Cam Table Properties instruction, refer to the MC_SetCamTableProperty (Set Cam Table Properties) instruction in the *NJ/NX-series Motion Control Instructions Reference Manual* (Cat. No. W508).

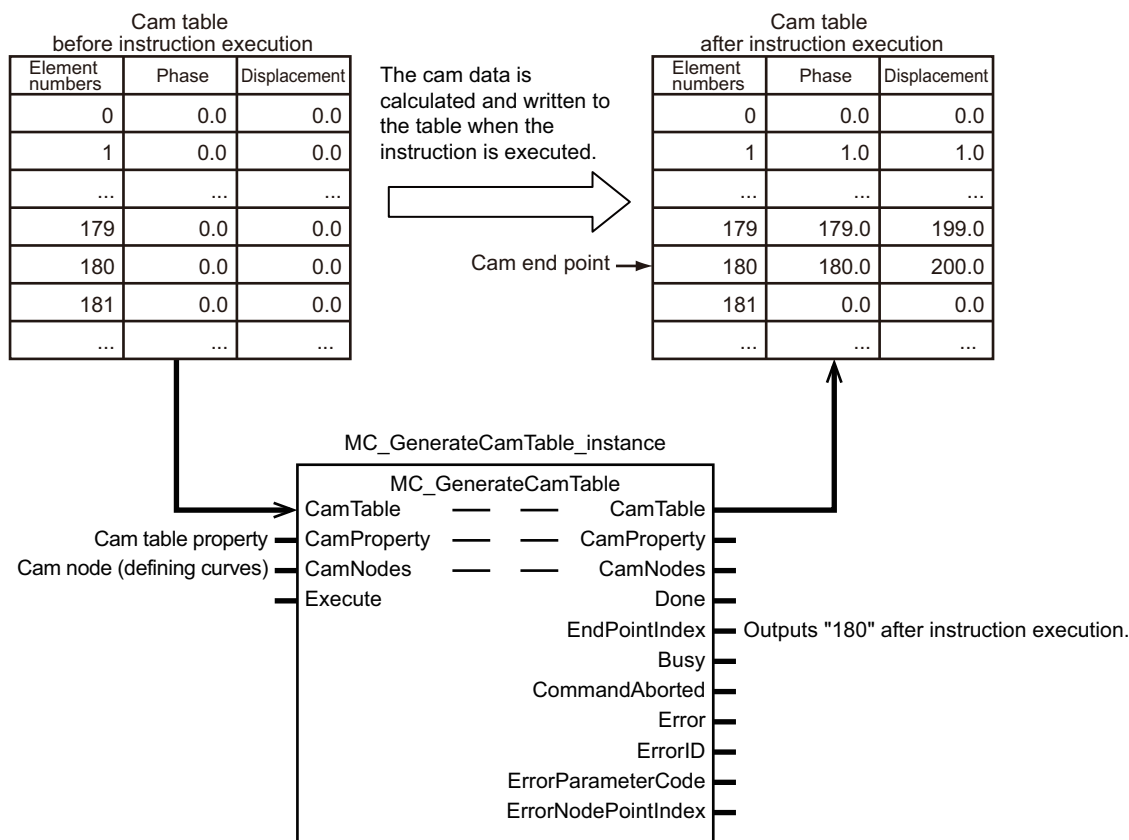
Generate Cam Table

You can generate the cam table by executing the MC_GenerateCamTable (Generate Cam Table) instruction.

The MC_GenerateCamTable instruction calculates the cam data using the values specified for *CamProperty* (Cam Properties) and *CamNodes* (Cam Nodes), and rewrites the cam data variable specified for the *CamTable* (Cam Table) in-out variable.

When rewriting is completed, the MC_GenerateCamTable instruction updates the end point index of the cam table and outputs the element number of the cam end point to *EndPointIndex* (End Point Index).

It is not necessary to execute the MC_SetCamTableProperty (Set Cam Table Properties) instruction after the MC_GenerateCamTable instruction is completed.



The cam data variable is an array variable with the data type of cam data structure `_sMC_CAM_REF`. You create the cam data variable on the Cam Editor of the Sysmac Studio.

For *CamProperty*, specify the cam property variable. The cam property variable is an array variable with the data type of cam property structure `_sMC_CAM_PROPERTY`. You create the cam property variable as a user-defined variable on the global variable table of the Sysmac Studio. Or, you create the variable using the cam data settings on the Sysmac Studio.

For *CamNodes*, specify the cam node variable. The cam node variable is an array variable with the data type of cam node structure `_sMC_CAM_NODE`. You create the cam node variable as a user-defined variable on the global variable table of the Sysmac Studio. Or, you create the variable using the cam data settings on the Sysmac Studio.

The cam property variable and the cam node variable are collectively called "cam definition variable".

If the cam definition variable is created as a user-defined variable, the default of its Retain attribute is Non-retain. You must set the Retain attribute of variable to Retain, if you want to reuse the variable after changing its value and switching the operating mode to PROGRAM mode or cycling the power supply. If you set the variable each time of use from the HMI, etc., the attribute can be left Non-retain.

If the cam definition variable is created with the cam data settings on the Sysmac Studio, the Retain attribute of variable will be fixed to Retain.

By using the HMI, etc. to set the values for the MC_GenerateCamTable instruction, you can create the cam data variable and adjust the cam operation without using the Sysmac Studio. The following is the procedure used to adjust the cam operation.

- 1** Create a user program, in advance, that includes the following processing.
 - Assigning the value of the cam definition variable that is set from the HMI to the Generate Cam Table instruction.
 - Displaying the cam variable that is created by the Generate Cam Table instruction graphically on the HMI.
 - Displaying the value of *EndPointIndex* (End Point Index) on the HMI.
- 2** Set the value of the cam definition variable from the HMI.
- 3** Execute the Generate Cam Table instruction.
- 4** Verify the curve shape of the generated cam table and the value of the end point index displayed on the HMI.
- 5** If there is no problem with the curve shape of the cam table and the number of the cam data, then execute the cam operation.
- 6** Verify the result of the cam operation and consider changing the value of the cam definition variable.
- 7** Repeat steps 2 to 6.

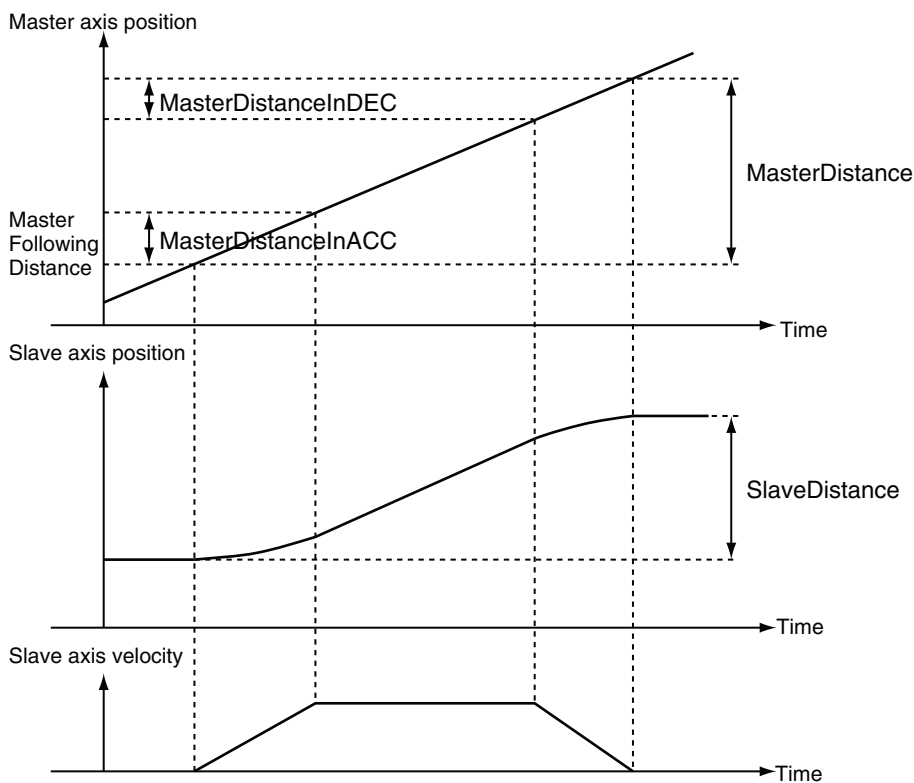
For details on the cam definition variable and the Generate Cam Table instruction, refer to the MC_GenerateCamTable instruction in the *NJ/NX-series Motion Control Instructions Reference Manual* (Cat. No. W508).

Refer to the *Sysmac Studio Version 1 Operation Manual* (Cat. No. W504-E1-10 or later) for information on creating and transferring the cam definition variables using the Sysmac Studio.

6-2-6 Synchronous Positioning

This function performs positioning using a trapezoidal curve while synchronizing the specified slave axis to the specified master axis. This is a type of electronic cam, but it does not use cam tables created in the Cam Editor. Operation starts when the MC_MoveLink (Synchronous Positioning) instruction is executed. Use the MC_Stop instruction to stop the axes in motion. Operation is performed for the *Slave* (Slave Axis) and the following are set: *Master* (Master Axis), *MasterDistance* (Master Axis Travel Distance), *MasterDistanceInACC* (Master Distance In Acceleration), *MasterDistanceInDEC* (Master Distance In Deceleration), *SlaveDistance* (Slave Axis Travel Distance), and *MasterStartDistance* (Master Following Distance). The command position or actual position can be specified for the master axis. You can specify one of the following as the start condition for synchronous operation: start of instruction, when trigger is detected, or when master axis reaches the master following distance.

The velocity and position of the slave axis are determined by the ratio of the travel distances of the master axis and the slave axis as shown in the following figure. The sync start position shown in the following figure represents the position where the sync start condition is met.

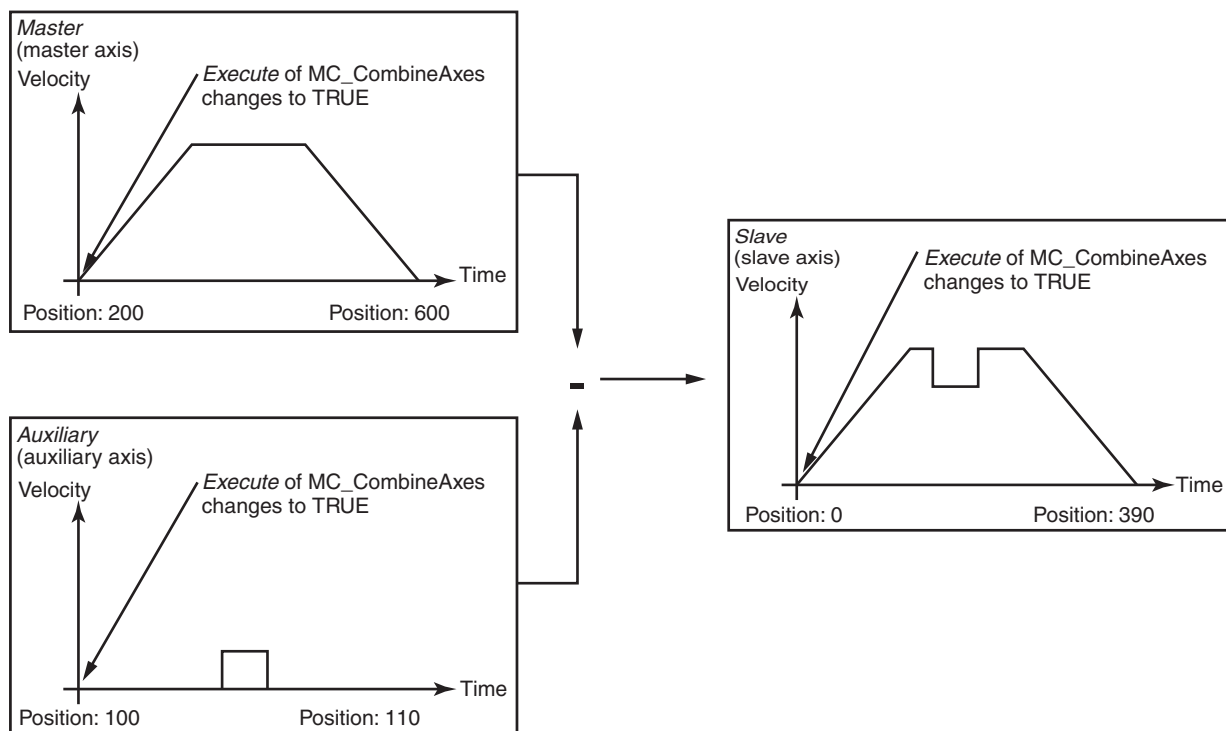


For details on synchronous positioning, refer to the MC_MoveLink (Synchronous Positioning) and MC_Stop instructions in the *NJ/NX-series Motion Control Instructions Reference Manual* (Cat. No. W508).

6-2-7 Combining Axes

The sum or difference of two positions can be used as the command position for the slave axis. Operation starts when the MC_CombineAxes instruction is executed. Use the MC_Stop instruction to stop axes in motion.

The following figure is an example demonstrating operation when subtracting axes. *Slave* (Slave Axis) command current position = *Master* (Master Axis) command current position – *Auxiliary* (Auxiliary Axis) command current position)

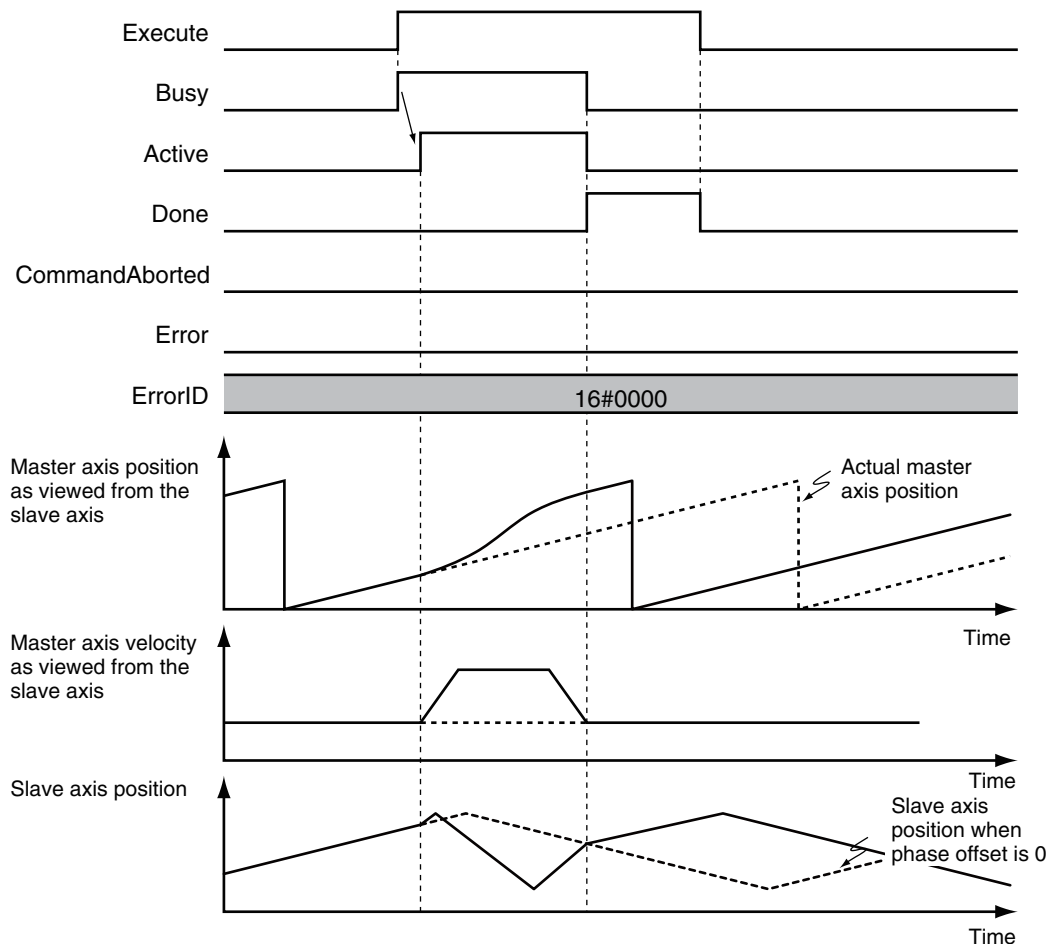


For details on combining axes, refer to the MC_CombineAxes and MC_Stop instructions in the *NJ/NX-series Motion Control Instructions Reference Manual* (Cat. No. W508).

6-2-8 Master Axis Phase Shift

The phase of the master axis as viewed from the slave axis can be shifted for the current instruction. The shift amount as viewed from the slave axis is a relative amount. During synchronization, the slave axis will synchronize to the relative distance of the master axis. You can execute the MC_Phasing (Shift Master Axis Phase) instruction to shift the phase for a synchronized control instruction.

You can specify the phase shift amount, target velocity, acceleration rate, deceleration rate, and jerk for the MC_Phasing (Shift Master Axis Phase) instruction.



For details on the shift master axis phase function and the synchronized control instructions for which a master axis phase shift can be applied, refer to the MC_Phasing (Shift Master Axis Phase) instruction in the *NJ/NX-series Motion Control Instructions Reference Manual* (Cat. No. W508).

6-2-9 Slave Axis Position Compensation

This function compensates the position of the slave axis currently in synchronized control.

An offset calculated from the value of the input variable is added to the command current position. The result is output to the Servo Drive to compensate the position of the slave axis in synchronized control.

Even when the MC Function Module commands the same travel distance to two axes, their actual travel distance may be different due to mechanical strain or other factors. This function can perform compensation in such a case.

To perform position compensation for the slave axis in synchronized control, execute the MC_SyncOffsetPosition (Cyclic Synchronous Position Offset Compensation) instruction.

For details on slave axis position compensation, refer to the MC_SyncOffsetPosition (Cyclic Synchronous Position Offset Compensation) instruction in the *NJ/NX-series Motion Control Instructions Reference Manual* (Cat. No. W508).

6-2-10 Achieving Synchronized Control in Multi-motion

When you use the standard functions of the MC Function Module, if the synchronized control instructions are executed between axes assigned to different tasks in the multi-motion, an Illegal Master Axis Specification (event code: 54620000 hex) occurs.

However, you can perform synchronized control of the master axis that is controlled in the primary periodic task and the slave axis that is controlled in the priority-5 periodic task by using the MC_PeriodicSyncVariables (Periodic Axis Variable Synchronization between Tasks) instruction.

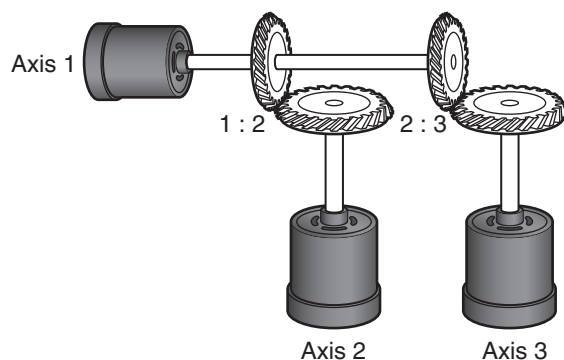
The cam operation and gear operation synchronized with the master axis and slave axis are available for the following combinations.

Master axis task	Slave axis task	
	Primary periodic task	Priority-5 periodic task
Primary periodic task	Synchronized by motion control instructions	Synchronized control is achieved by executing the MC_PeriodicSyncVariables (Periodic Axis Variable Synchronization between Tasks) instruction and using the virtual master axis in the priority-5 periodic task.
Priority-5 periodic task	Not available.	Synchronized by motion control instructions

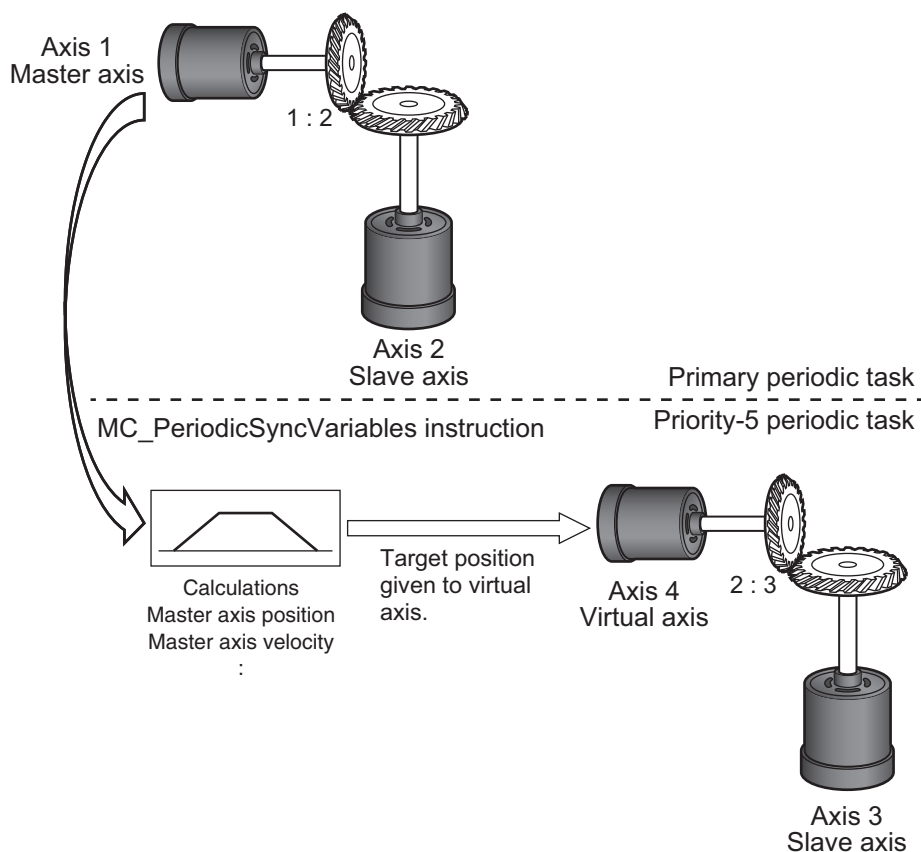
Axis Composition in Operation Examples

In the following figure, axis 1 is the master axis. Axis 2 is a slave axis that requires high-speed and high-precision control. It is assigned to the primary periodic task. Axis 3 is a slave axis that does not require precision. It is assigned to a priority-5 periodic task. The master axis (axis 1) is assigned to the primary periodic task.

● Physical Axis Composition



● Logical Axis Composition



Programming is placed in both the primary periodic task and priority-5 periodic task to achieve the operation for the above application.

Refer to the *NJ/NX-series Motion Control Instructions Reference Manual* (Cat. No. W508) for details on the MC_PeriodicSyncVariables (Periodic Axis Variable Synchronization between Tasks) instruction.

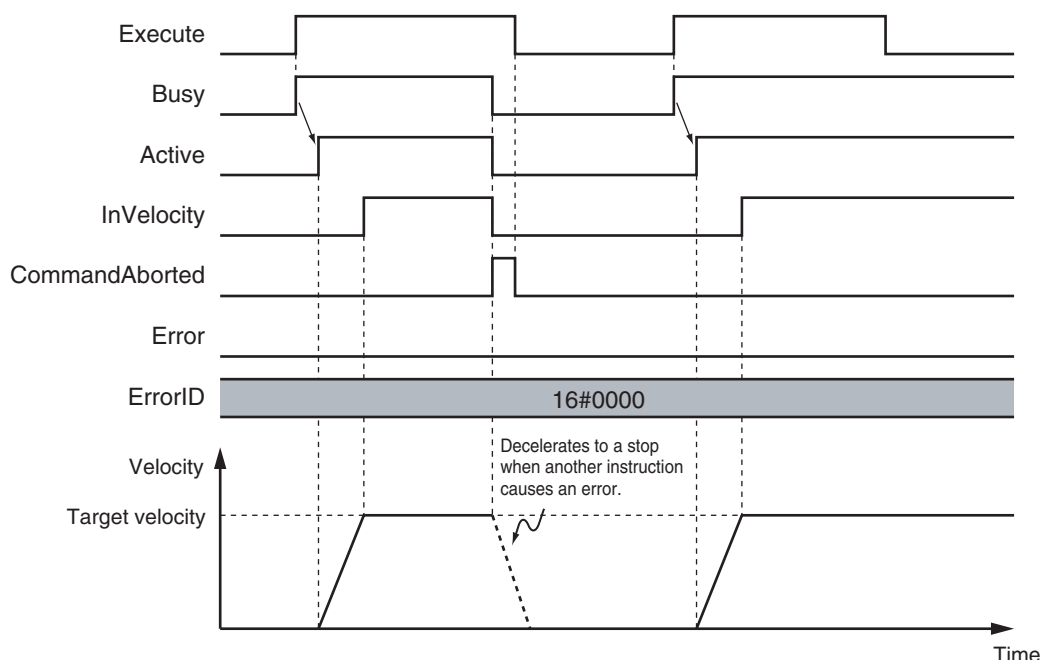
6-3 Single-axis Velocity Control

This section describes the operation of velocity control for single axes.

Refer to the *NX-series Position Interface Units User's Manual* (Cat. No. W524) for the differences when you use NX-series Pulse Output Units.

6-3-1 Velocity Control

Velocity control is used to constantly move an axis at the specified velocity. You can also specify the acceleration rate, deceleration rate, and jerk. To stop an axis, use the MC_Stop instruction or execute another motion instruction. If you specify a target velocity of 0, the axis will not move but the axis status will indicate that it is moving. If any other motion control instruction is executed with multi-execution of instructions during velocity control, the operation will switch only after reaching the target velocity.



The MC Function Module uses Position Control Mode of the Servo Drive or other device and sends target position commands to achieve the specified target velocity.

The position control loop is enabled in the Servo Drive or other device. Therefore, as the command velocity slows down, e.g., due to disturbance, and the following error increases, the velocity will change to eliminate this following error.

For details, refer to the MC_MoveVelocity (Velocity Control) instruction in the *NJ/NX-series Motion Control Instructions Reference Manual* (Cat. No. W508).

6-3-2 Cyclic Synchronous Velocity Control

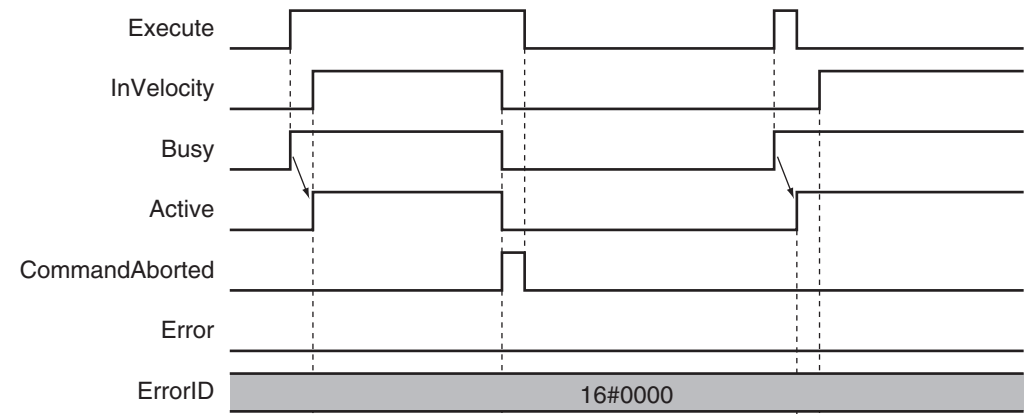
The control mode of the Servo Drive is set to Velocity Control Mode and a command speed is output every control period.

 Precautions for Correct Use

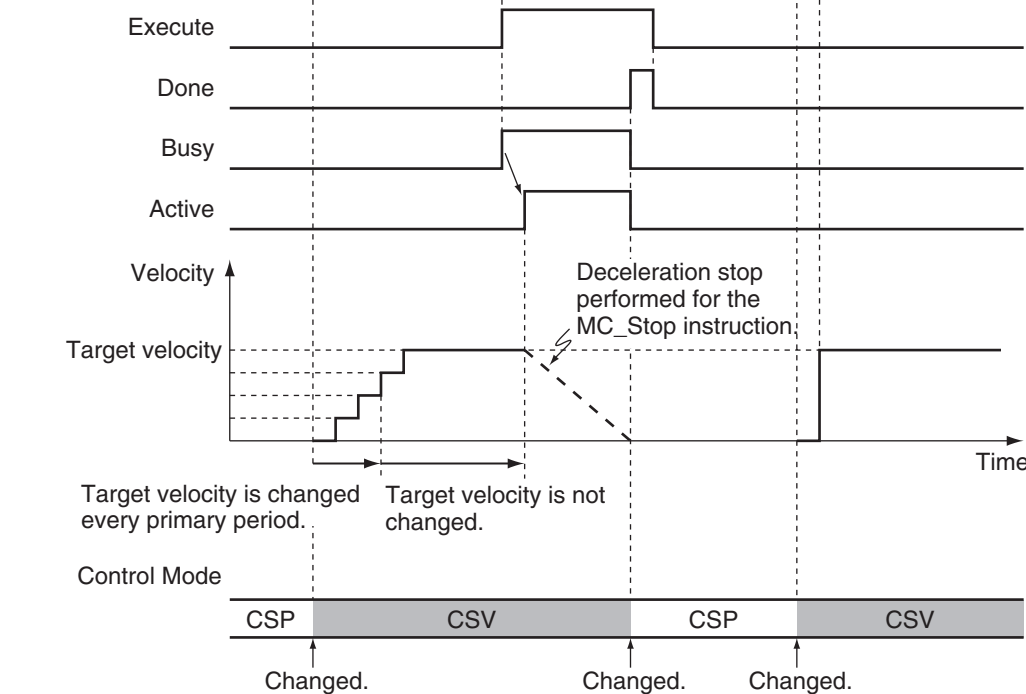
You cannot use cyclic synchronous velocity control for an NX-series Pulse Output Unit.

To stop an axis, use the MC_Stop instruction or execute another motion control instruction. If you specify a target velocity of 0, the axis will not move but the axis status will indicate that it is moving.

MC_SyncMoveVelocity Instruction



MC_Stop Instruction



The Servo Drive will receive commands in the velocity control loop. Therefore, if any disturbance causes the velocity to decrease below the command velocity, no change in velocity will occur to remove the following error.

For details, refer to the MC_SyncMoveVelocity (Cyclic Synchronous Velocity Control) instruction in the *NJ/NX-series Motion Control Instructions Reference Manual* (Cat. No. W508).

6-4 Single-axis Torque Control

Torque control continuously applies the specified amount of torque. You can use *TorqueRamp* to specify the rate of change of the torque until the *Torque* (Target Torque) is reached.

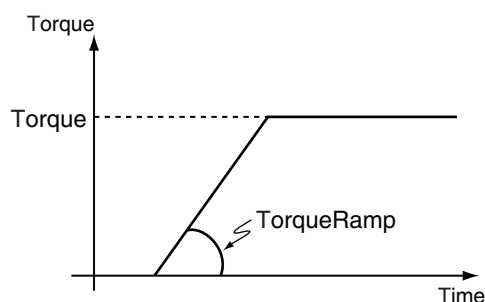


Precautions for Correct Use

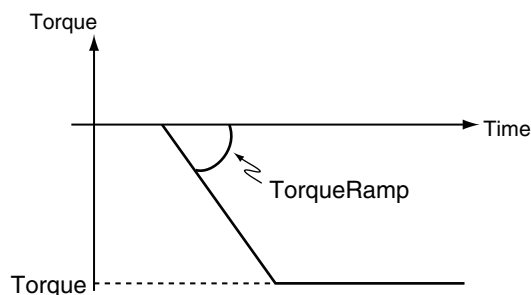
- To be safe, always set a velocity limit value for torque control.
- You cannot use single-axis torque control for an NX-series Pulse Output Unit.

To stop an axis, use the MC_Stop instruction or execute another motion instruction. If you specify a *Torque* (Target Torque) of 0, the axis will not move but the axis status will indicate that it is moving.

● Direction Designation = Positive Direction



● Direction Designation = Negative Direction



The MC Function Module uses the Torque Control Mode of the Servo Drive. The Servo Drive receives the torque command value from the MC Function Module in the torque control loop and to control the torque. You can specify the velocity limit value for the Servo Drive in the *Velocity* (Velocity Limit) input variable to the motion control instruction. You can use this to limit high-speed revolution of the motor when the load on the motor is low in Torque Control Mode.

For details, refer to the MC_TorqueControl instruction in the *NJ/NX-series Motion Control Instructions Reference Manual* (Cat. No. W508).

6-5 Common Functions for Single-axis Control

This section describes the common functions used for single-axis control.

Refer to the *NX-series Position Interface Units User's Manual* (Cat. No. W524) for the differences when you use NX-series Pulse Output Units.

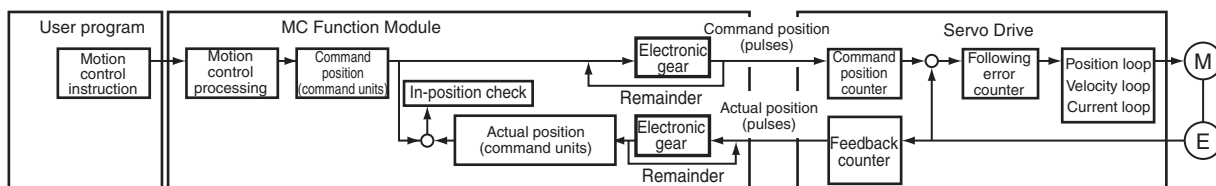
6-5-1 Positions

Types of Positions

The MC Function Module uses the following two types of positions.

Type of position	Definition
Command position	This is the position that the MC Function Module outputs to control an axis.
Actual position	The actual position as input from the Servo Drive or encoder input.

The following figure shows the relationship between the command position and the actual position for an EtherCAT slave Servo Drive.



The command position and actual position share the following items.

Item	Command position	Actual position
Count Mode	You can set Linear Mode or Rotary Mode.	The same Count Mode is used as for the command position.
Position increment	You can set one of the following: mm, μ m, nm, inch, degree, or pulse.	The unit is the same as the unit of the command position.
Software limits	You can set the range of operation of the software.	The range is the same as the range for the command position.
Changing the current position	You can change the actual position to any desired position.	This value will be set to the same position as the command position. ^{*1}
Defining home	Home is either defined or undefined.	The status of home is the same as the command position.

*1. If there is any following error before the change, the following error value is maintained in the actual position.



Additional Information

Refer to the *NX-series Position Interface Units User's Manual* (Cat. No. W524) for information on the NX-series Position Interface Units.

Axis Parameters That Are Related to Positions

Parameter name	Function	Setting range	Default
In-position Range	Set the in-position width. (Unit: command units)	Non-negative long reals	10
In-position Check Time	Set the in-position check time in milliseconds. Set 0 to check for the end of positioning only when you define the home position during homing and not check positioning at other times. (Unit: ms)	0 to 10,000	0
Software Limits	Select the software limit function. 0: Disabled. 1: Deceleration stop for command position 2: Immediate stop for command position 3: Deceleration stop for actual position 4: Immediate stop for actual position	0 to 4	0
Positive Software Limit	Set the software limit in the positive direction. (Unit: command units)	Long reals	2,147,483,647
Negative Software Limit	Set the software limit in the negative direction. (Unit: command units)	Long reals	-2,147,483,648
Following Error Over Value	Set the excessive following error check value. Set 0 to disable the excessive following error check. (Unit: command units)	Non-negative long reals	0
Following Error Warning Value	Set the following error warning check value. Set 0 to disable the following error warning check. (Unit: command units)	Non-negative long reals that are less than or equal to the Following Error Over Value	0

Specifying Target Positions for Axis Operations

The actual position or distance for a positioning motion is specified with the *Position* (Target Position) and *Distance* (Travel Distance) input variables to the motion control instruction.

Monitoring Positions

You can read Axis Variables in the user program to monitor positions.

In the descriptions, a variable name `_MC_AX[*]` is used as an example, but the same information applies to `_MC1_AX[*]` and `_MC2_AX[*]`.

Variable name	Data type	Meaning	Function
<code>_MC_AX[0-255].Cmd.Pos</code>	LREAL	Command Current Position	This is the current value of the command position. When the Servo is OFF and the mode is not the position control mode, the actual current position is output.
<code>_MC_AX[0-255].Act.Pos</code>	LREAL	Actual Current Position	This is the actual current position.

6-5-2 Velocity

Types of Velocities

The following two types of axis velocities are used in the MC Function Module.

Velocity type	Definition
Command velocity	This is the velocity that the MC Function Module outputs to control an axis.
Actual velocity	This is the velocity calculated in the MC Function Module based on the actual position input from the Servo Drive or encoder input.*1

*1. This value is given if the Velocity actual value (606C hex) is mapped in the PDOs and assigned to the Actual Current Velocity.

Velocity Unit

A velocity is given in command units/s. The command unit is the value obtained from unit conversion of the position display unit and the electronic gear.

Axis Parameters That Are Related to Velocities

Parameter name	Function	Setting range	Default
Maximum Velocity	Specify the maximum velocity for the axis. If a target velocity that exceeds the maximum velocity is specified for an axis motion instruction, the axis will move at the maximum velocity.	Positive long reals	400,000,000
Start Velocity	Set the start velocity for each axis. Set a value that does not exceed the maximum velocity. (Unit: command units/s)	Positive long reals	0
Maximum Jog Velocity	Set the maximum jog velocity for each axis.*1 Set a value that does not exceed the maximum velocity. (Unit: command units/s)	Positive long reals	1,000,000
Velocity Warning Value	Set the percentage of the maximum velocity at which to output a velocity warning for the axis. No velocity warning is output if 0 is set. (Unit: %)	0 to 100	0
Actual Velocity Filter Time Constant	Set the time period to calculate the average travel of the actual velocity in milliseconds. The average travel is not calculated if 0 is set. (Unit: ms) Use this to reduce variations in the actual current velocity when axis velocity is slow.	0 to 100	0

*1. The maximum jog velocity is used as the command velocity if you specify a velocity command value that is greater than the maximum jog velocity.

Specifying Target Velocities for Axis Operations

The velocity used in an actual positioning motion is specified by the *Velocity* (Target Velocity) input variable to the motion control instruction.

Monitoring Velocities

You can read Axis Variables in the user program to monitor velocities.

In the descriptions, a variable name `_MC_AX[*]` is used as an example, but the same information applies to `_MC1_AX[*]` and `_MC2_AX[*]`.

Variable name	Data type	Meaning	Function
<code>_MC_AX[0-255].Cmd.Vel</code>	LREAL	Command Current Velocity	This is the current value of the command velocity. A plus sign is added during travel in the positive direction, and a minus sign is added during travel in the negative direction.
<code>_MC_AX[0-255].Act.Vel</code>	LREAL	Actual Current Velocity	This is the actual current velocity. A plus sign is added during travel in the positive direction, and a minus sign is added during travel in the negative direction.

6-5-3 Acceleration and Deceleration

Unit of Acceleration and Deceleration Rates

Acceleration rates and deceleration rates are given in command units/s². The command unit is the value obtained from unit conversion of the position display unit and the electronic gear.

Axis Parameters That Are Related to Acceleration and Deceleration

Parameter name	Function	Setting range	Default
Maximum Acceleration	Set the maximum acceleration rate for an axis operation command. There will be no limit to the acceleration rate if 0 is set. (Unit: command units/s ²)	Non-negative long reals	0
Maximum Deceleration	Set the maximum deceleration rate for an axis operation command. There will be no limit to the deceleration rate if 0 is set. (Unit: command units/s ²)	Non-negative long reals	0
Acceleration/Deceleration Over	Set the operation for when the maximum acceleration/deceleration rate would be exceeded after excessive acceleration/deceleration during acceleration/deceleration control of the axis because stopping at the target position is given priority. *1 0: Use rapid acceleration/deceleration. (Blending is changed to Buffered.) 1: Use rapid acceleration/deceleration. 2: Minor fault stop	0 to 2	0
Acceleration Warning Value	Set the percentage of the maximum acceleration rate at which to output an acceleration warning for the axis. No acceleration warning is output if 0 is set. (Unit: %)	0 to 100	0

Parameter name	Function	Setting range	Default
Deceleration Warning Value	Set the percentage of the maximum deceleration rate at which to output a deceleration warning for the axis. No deceleration warning is output if 0 is set. (Unit: %)	0 to 100	0

*1. Refer to 6-5-7 *Multi-execution of Motion Control Instructions (Buffer Mode)* on page 6-48 for operation with each set value.

Specifying Acceleration and Deceleration Rates for Axis Operation

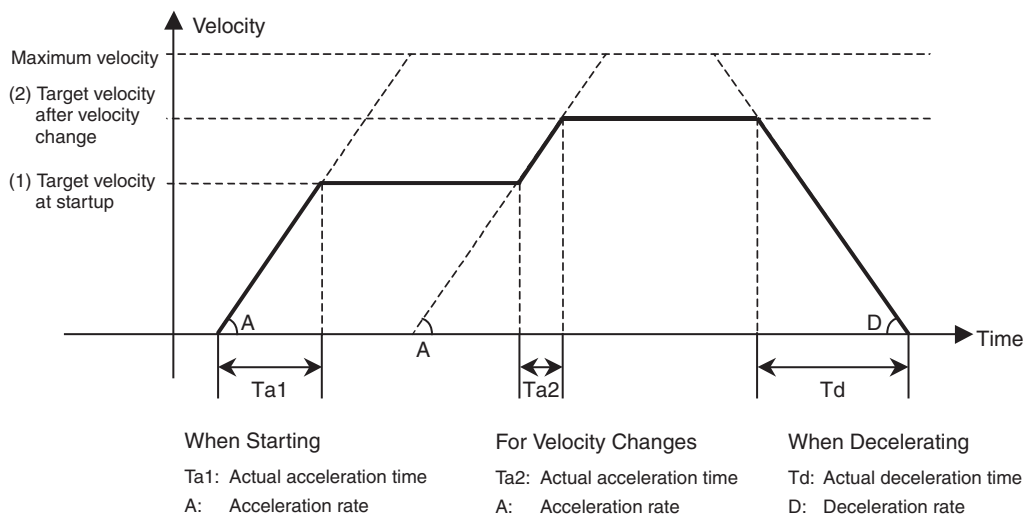
The acceleration and deceleration rates used in an actual positioning motions are specified by the *Acceleration* (Acceleration Rate) and *Deceleration* (Deceleration Rate) input variables to the motion control instruction.

Monitoring Acceleration and Deceleration Rates

You can read Axis Variables in the user program to monitor acceleration and deceleration rates. In the descriptions, a variable name `_MC_AX[*]` is used as an example, but the same information applies to `_MC1_AX[*]` and `_MC2_AX[*]`.

Variable name	Data type	Meaning	Function
<code>_MC_AX[0-255].Cmd.AccDec</code>	LREAL	Command Current Acceleration/Deceleration	This is the current value of the command acceleration/deceleration rate. A plus sign is added for acceleration, and a minus sign is added for deceleration.

Example of Acceleration/Deceleration Operation



If you specify a short travel distance or a low acceleration/deceleration rate, the target velocity may not be reached. If the target position is exceeded after re-execution of the motion control instruction with the newly updated acceleration or deceleration rate, positioning is performed at an acceleration or deceleration rate that will enable stopping at the target position.

6-5-4 Jerk

The jerk specifies the rate of change in the acceleration rate or deceleration rate. If the jerk is specified, the velocity waveform during acceleration will be an S-curve, which will reduce the shock and vibration on the machine.



Additional Information

Jerk is also called jolt, surge and lurch.

Jerk Unit

Jerk is given in command units/s³. The command unit is the value obtained from unit conversion of the position display unit and the electronic gear.

Specifying Jerk for Axis Motion

The jerk used in an actual positioning motion is specified with the *Jerk* input variable to the motion control instruction. The same value is used for acceleration and deceleration.

Use the following formula to calculate the value to set for the jerk.

$\text{Jerk} = \text{Acceleration rate} \div (\text{Time of acceleration} \times \text{Ratio of time to apply jerk during acceleration}/2)$

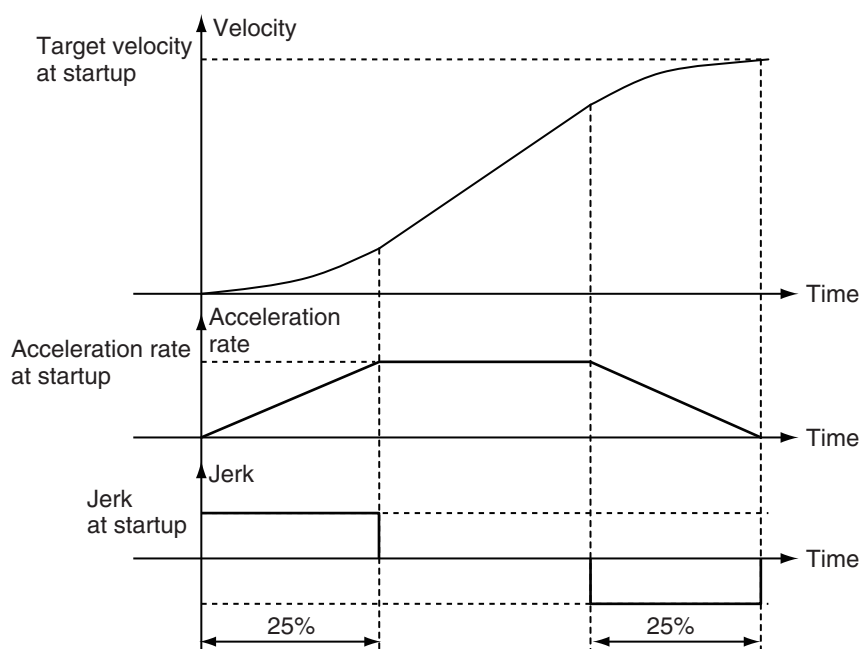
Jerk is applied in two sections: at the start of acceleration and at the end of acceleration. The time that jerk is applied is therefore divided by 2.

● Example of Velocity Control When Jerk Is Specified

The acceleration will change at a constant rate over the range where jerk is specified. The command velocity will form a smooth S curve. A fixed acceleration rate is used in areas where the jerk is set to 0. This command velocity will form a straight line.

Example: Acceleration of 25,000 mm/s², Acceleration Time of 0.1 s, and a Jerk Application Rate of 50%

$\text{Jerk} = 25,000 / (0.1 \times 0.5/2) = 1,000,000 \text{ (mm/s}^3\text{)}$



Monitoring Jerk

You can read Axis Variables in the user program to monitor jerk.

In the descriptions, a variable name `_MC_AX[*]` is used as an example, but the same information applies to `_MC1_AX[*]` and `_MC2_AX[*]`.

Variable name	Data type	Meaning	Function
<code>_MC_AX[0-255].Cmd.Jerk</code>	LREAL	Command Current Jerk	This is the current value of the command jerk.

6-5-5 Specifying the Operation Direction

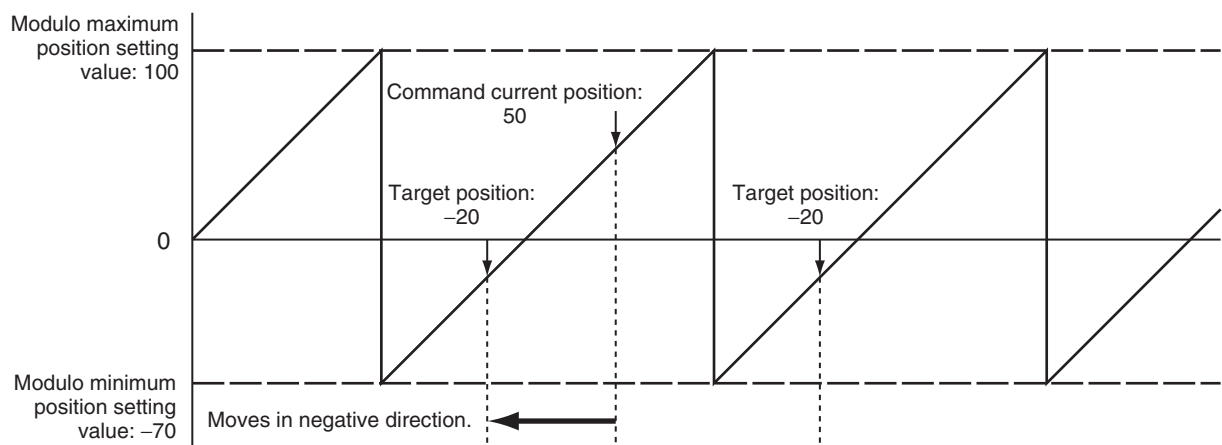
If you want to specify a rotation direction, such as shortest way, using an index table, set the Count Mode to Rotary Mode. Next, set the operation direction with the *Direction* input variable to the motion control instruction for an absolute position. If you set the direction to the shortest way, positive direction, negative direction, or current direction, you can specify a position that is greater than or equal to the modulo minimum position and less than the modulo maximum position within one turn of the ring counter. The *Direction* input variable will be ignored when the Count Mode is set to Linear Mode. Positioning will be performed to the target position.

The following table lists the different directions you can specify in the MC Function Module.

Direction	Operation
Shortest way	Motion starts in the direction where the command current position and the target position are closer to each other.
Positive direction	Motion starts in the positive direction.
Negative direction	Motion starts in the negative direction.
Current direction	Motion starts in the same direction as the previous operation.
No direction specified	Motion starts in the direction that does not pass through the upper and lower limits of the ring counter. With this direction specification, you can specify a target position that exceeds the upper or lower limits of the ring counter. If that occurs, relative positioning is performed using the difference between the target position and the command current position as the target distance. This enables you to perform multi-turn positioning on the ring counter.

Example for Shortest Way

The following example illustrates when positioning is performed towards a target position of -20 when the command current position is 50 .



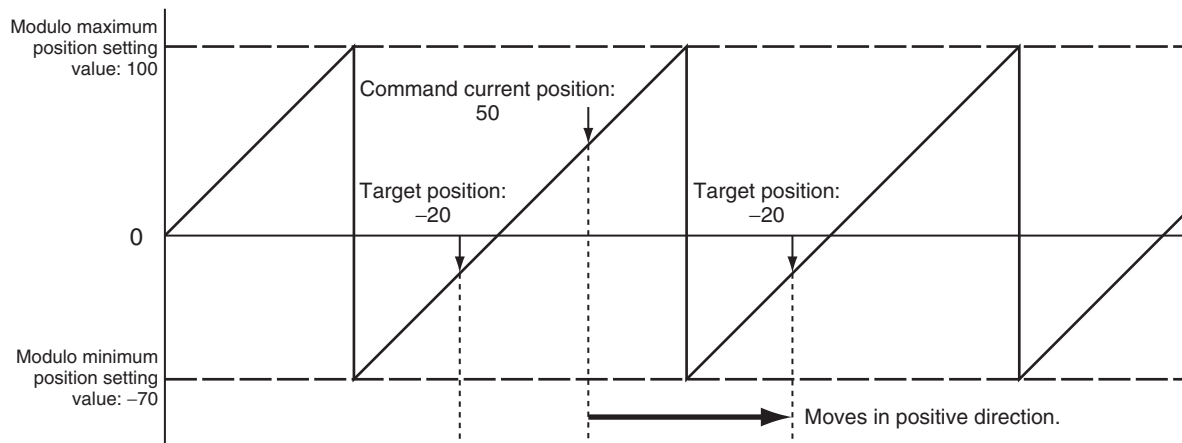


Additional Information

Moves in the same direction as the Current Direction specification if the travel distance is the same in the positive and negative directions.

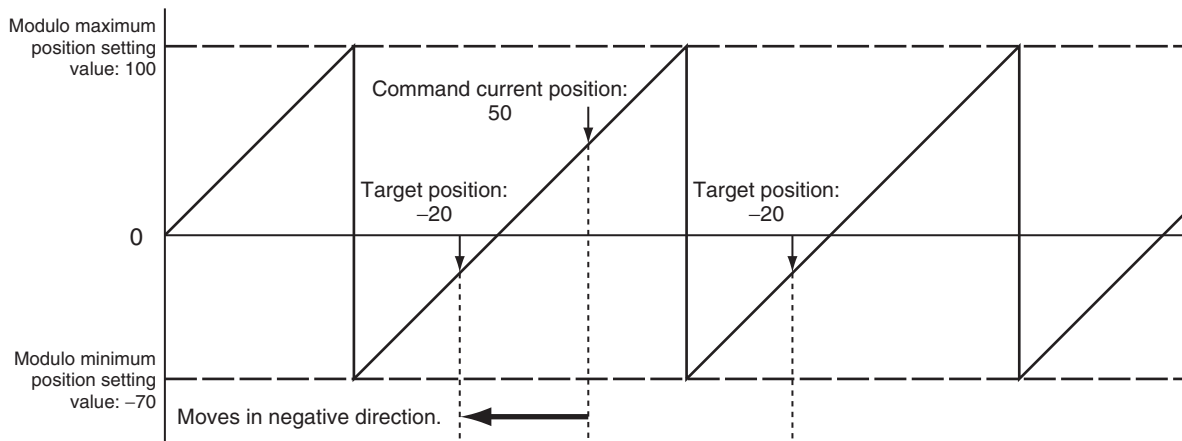
Example for Positive Direction

The following example illustrates when positioning is performed towards a target position of -20 when the command current position is 50.



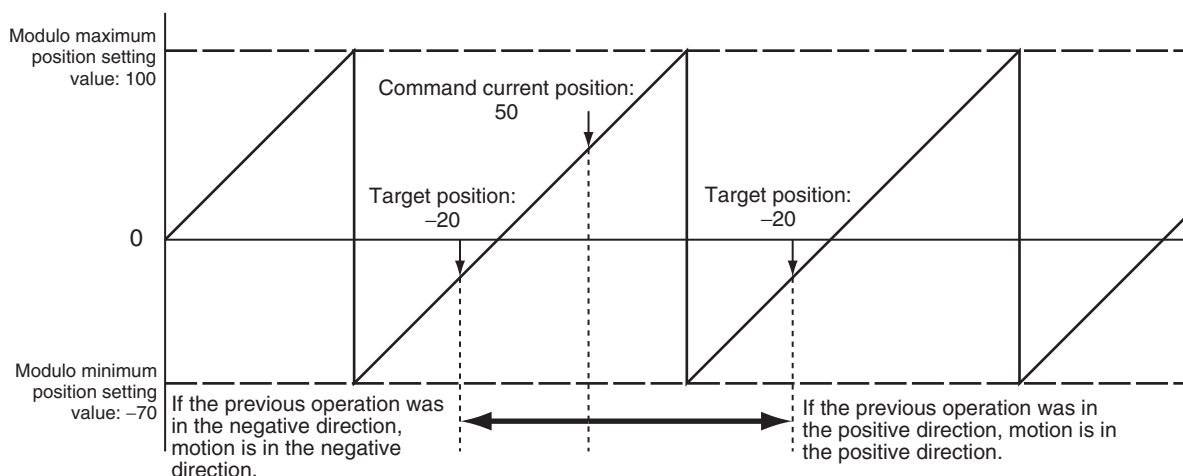
Example for Negative Direction

The following example illustrates when positioning is performed towards a target position of -20 when the command current position is 50.



Example for Current Direction

The following example illustrates when positioning is performed towards a target position of -20 when the command current position is 50 .



The direction of the previous operation is given in the Command Direction in the Axis Variable.



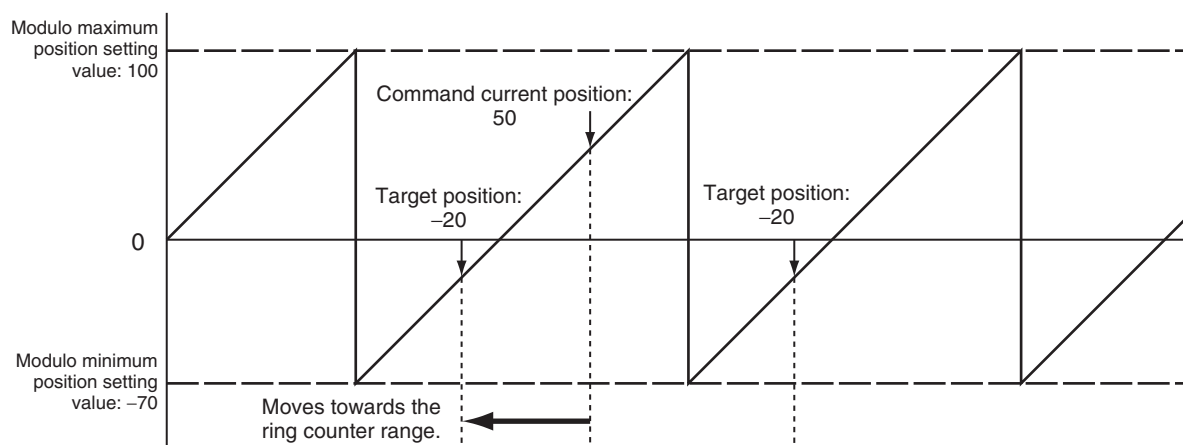
Precautions for Correct Use

Observe the following precautions on the operation direction of the previous operation.

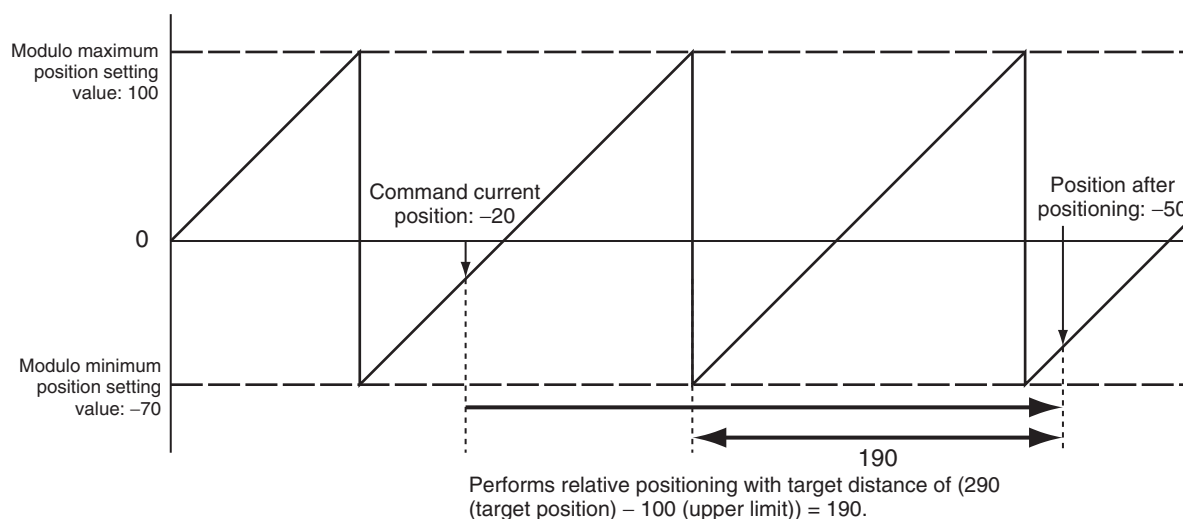
- If the MC_Home or MC_HomeWithParameter instruction exceeds the point where the home input was detected and reverses operation, the opposite direction of the home input detection direction is used.
- If a homing compensation value is set for the MC_Home or MC_HomeWithParameter instruction, the axis will move in the direction of the compensation value.
- If an immediate stop is specified for the MC_TouchProbe (Enable External Latch) instruction, the latch position may be exceeded and the direction may be reversed.
- The direction may be reversed for the MC_MoveFeed (Interrupt Feeding) instruction.
- When the MC_ResetFollowingError instruction is executed, the error is set to zero, so the command direction is used.
- If an immediate stop is specified for an external input signal or resetting the error counter is specified for stopping for a limit input, the operation may reverse direction toward the position where the external input signal was received.

Example for No Direction Specification

The following example illustrates when positioning is performed towards a target position of -20 when the command current position is 50 .



Similarly, the following example illustrates when the ring counter upper limit is 100, the lower limit is -70 , the command current position is -20 , and positioning is performed towards a target position of 290.



6-5-6 Re-executing Motion Control Instructions

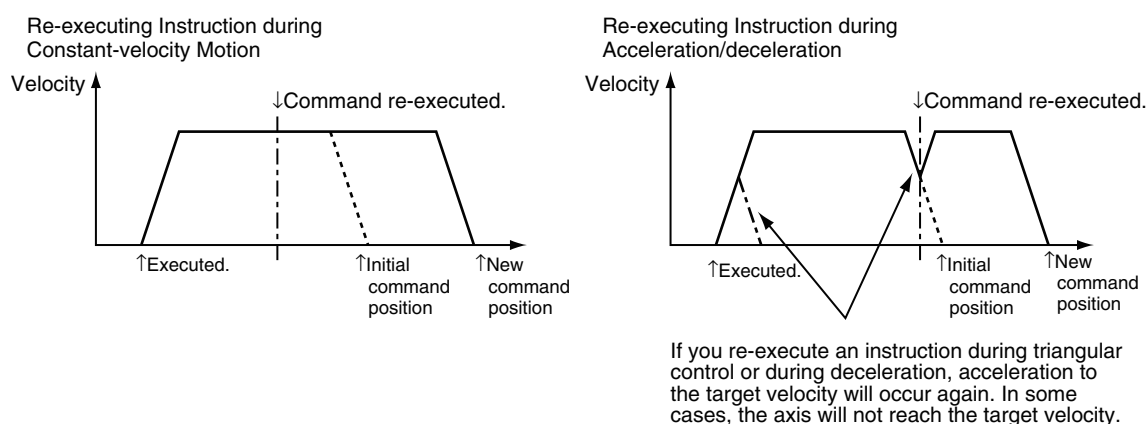
This section describes how to modify input variables of the same instance of a motion control instruction during operation of a single axis and re-execute that instruction. The input variables *Position* (Target Position), *Distance* (Travel Distance), *Velocity* (Target Velocity), *Acceleration* (Acceleration Rate), *Deceleration* (Deceleration Rate), and *Torque* (Target Torque) and sometimes other input variables can be changed by re-execution. An instruction error will occur if you change an input variable that cannot be changed and attempt to re-execute the instruction. If you re-execute an instruction that has been buffered due to multi-execution of instructions, the input variables for the instruction in the buffer will change.

For details on input variables that can be changed, refer to the *NJ/NX-series Motion Control Instructions Reference Manual* (Cat. No. W508).

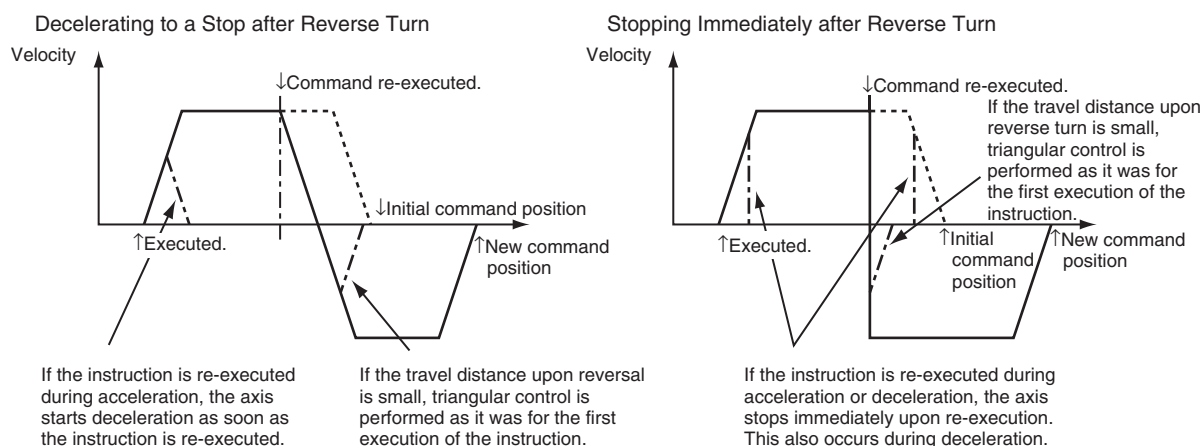
Changing the Target Position

If you change the target position with re-execution, the operation may change depending on the timing of the change and the new target position. If the direction of motion reverses due to a change in the target position, you can choose to decelerate to a stop after a reverse turn or stop immediately after reversing with the Operation selection at Reversing axis parameter.

● When a Reverse Turn Does Not Occur for the New Command Value



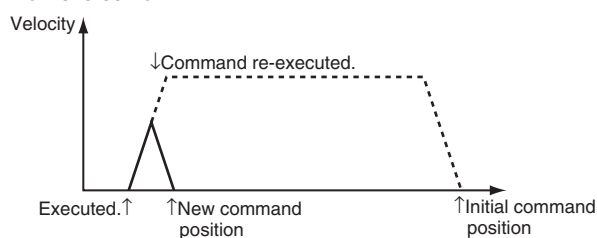
● When a Reverse Turn Occurs for the New Command Value



● Triangular Control Patterns

The triangular control shown in the figure below may result if the travel distance is shortened due to a change in the target position.

No Reverse Turn

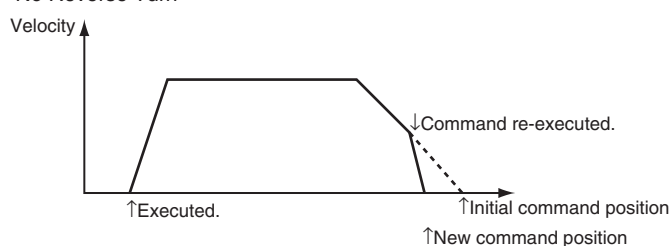


● Excessive Deceleration Patterns

In the following case, priority is given to stopping at the target position. Therefore, the deceleration rate will exceed the specified deceleration rate. If the deceleration rate exceeds the rate that is set in the Maximum Deceleration axis parameter, the operation set in the Acceleration/Deceleration Over axis parameter setting is performed.

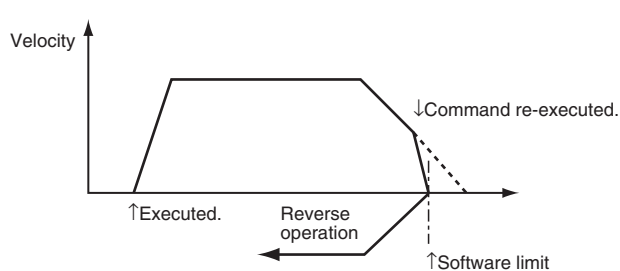
- If There Is No Reverse Turn and the Target Position Would Be Exceeded at the Specified Deceleration Rate

No Reverse Turn



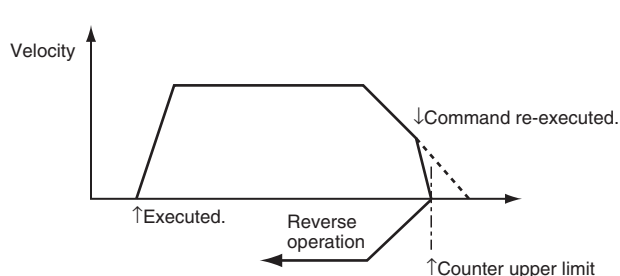
- If There Is A Reverse Turn and Decelerating to a Stop Would Exceed a Software Limit

No Reverse Turn



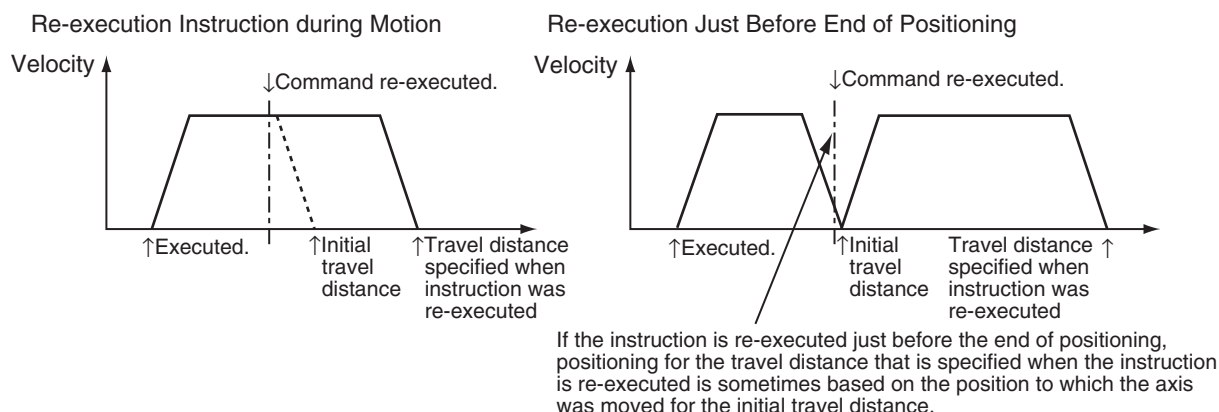
- If There Is A Reverse Turn and Decelerating to a Stop Would Result in Command Current Position Overflow or Underflow

No Reverse Turn



Changing the Travel Distance

Even if you change the travel distance and re-execute the MC_MoveRelative (Relative Positioning) instruction, positioning is performed for the new travel distance in reference to the position where the motion first started. However, if the instruction is executed again just before positioning is completed, it may be executed as a new instruction rather than as a re-execution of the same instruction.



Precautions for Correct Use

Do not change the travel distance and re-execute the instruction just before the end of positioning.

Changing the Target Velocity

The operation is changed only during acceleration (including acceleration for triangular control) and constant-velocity motion. Changes are also accepted when the axis is decelerating, but operation is not affected.

Changing the Acceleration Rate

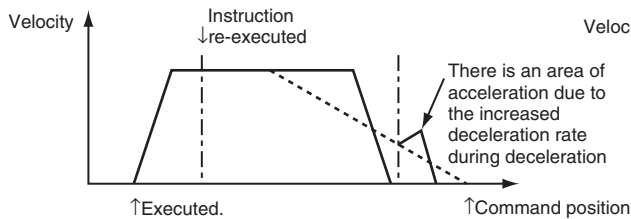
The operation is changed only during acceleration and acceleration during triangular control. If it is changed when moving at a constant speed, the changed rate applies to acceleration for an override. Changes are also accepted when the axis is decelerating, but operation is not affected.

Changing the Deceleration Rate

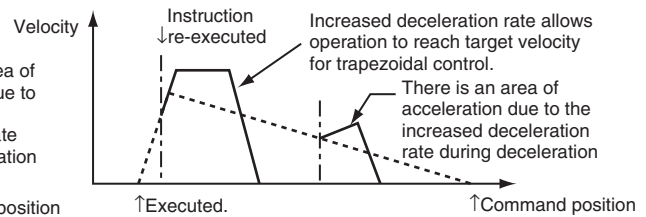
The deceleration rate is changed only during acceleration, constant-velocity motion, deceleration, triangular control, or during deceleration-exceed control. If the new deceleration rate causes the axis to exceed the target position, stopping at the target position is given the highest priority. Therefore, in this case, the actual deceleration rate will exceed the specified deceleration rate.

● Patterns Where Deceleration Rate Increases

Trapezoidal Control

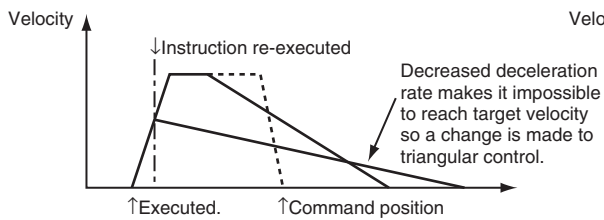


Triangular Control Followed by Trapezoidal Control

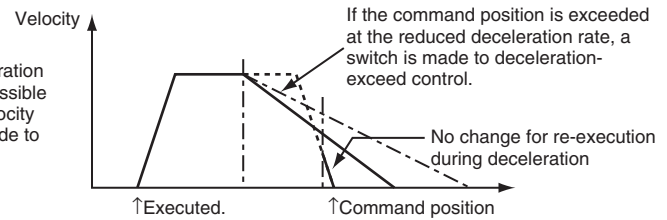


● Patterns Where Deceleration Rate Decreases

Trapezoidal Control or Triangular Control



Deceleration-exceed Control

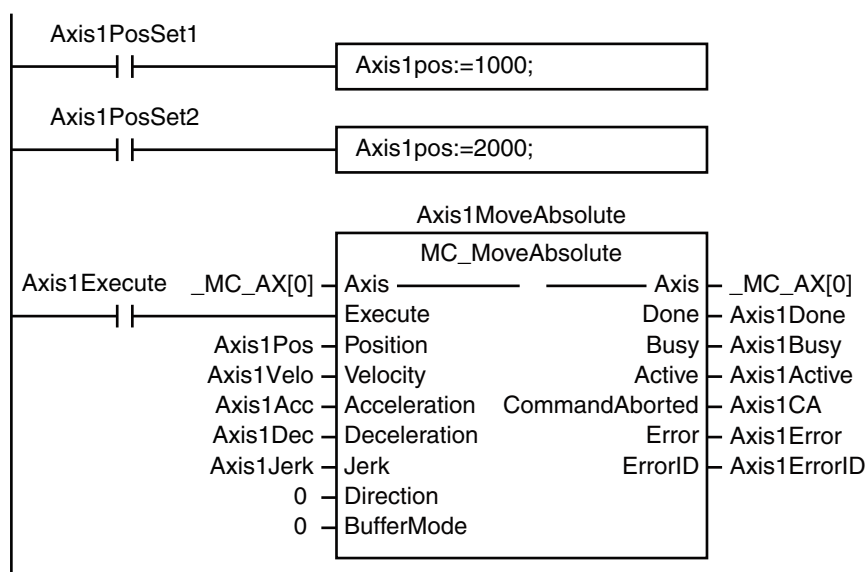


Changing the Torque Command

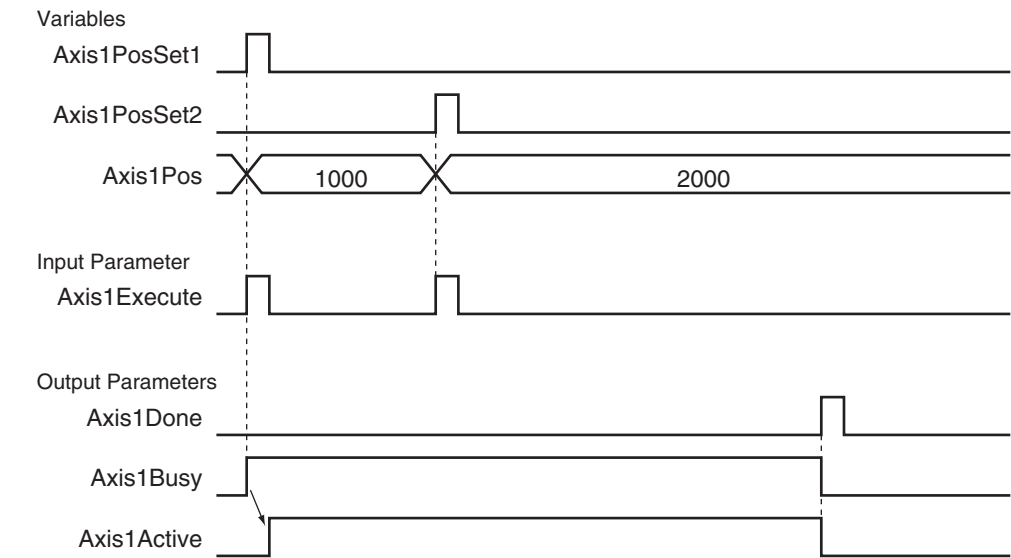
The torque command value will change based on the torque ramp specification when you re-execute a motion control instruction.

Programming Example for Re-execution

This example demonstrates changing the target position from 1000 to 2000 for absolute positioning. In this example, the variable *Axis1Pos* is used as the input parameter to the target position. Specify the target position to 1000 with the MOV instruction and change *Axis1Execute* to TRUE to begin positioning. Specify the target position to 2000 during operation and change *Axis1Execute* to TRUE again to switch to a positioning operation for the new target position of 2000.



● Timing Charts



Precautions for Correct Use

For input variables that are not changed, always use the same values as before re-execution of the instruction.

6-5-7 Multi-execution of Motion Control Instructions (Buffer Mode)

You can execute another motion control instruction while an axis is moving. In the PLCopen® technical specifications, this functionality is defined as Buffer Mode, but in the MC Function Module this is sometimes referred to as multi-execution of instructions. You can use multi-execution of instructions to execute multiple motion control instructions in sequence without stopping the overall motion.

The following terms are used in relation to multi-execution of instructions in the MC Function Module.

Term		Meaning
This manual	PLCopen®	
Current instruction	Previous function block	The motion control instruction that was in operation just before executing the multi-execution instruction.
Buffered instruction	Next function block	A motion control instruction that was executed during an axis motion and is waiting to be executed.
Transit velocity	Blending	When blending is specified, it specifies the command velocity to use by the current instruction to move to the specified target position.

You can set the *BufferMode* (Buffer Mode Selection) input variable to motion control instruction to select one of the following Buffer Modes. The main difference between these modes is the timing at which the buffered instructions are executed and the transit velocity.

Buffer Mode	Description of operation
Aborting	The current instruction is aborted and the multi-executed instruction is executed.
Buffered	The buffered instruction is executed after the operation for the current instruction is normally finished.
Blending	The buffered instruction is executed after the target position of the current instruction is reached. In this mode, no stop is performed between the current instruction and the buffered instruction. You can select from the following transit velocities for when the current instruction reaches the target position.
Blending Low (low velocity)	The transit velocity is set to the target velocity of the current instruction or the buffered instruction, whichever is lowest.
Blending Previous (previous velocity)	The target velocity of the current instruction is used as the transit velocity.
Blending Next (next velocity)	The target velocity of the buffered instruction is used as the transit velocity.
Blending High (high velocity)	The transit velocity is set to the target velocity of the current instruction or the buffered instruction, whichever is highest.

The multi-execution instruction is buffered in the MC Function Module and will be executed at the specified *BufferMode* timing and transit velocity for both buffered and blending modes. There is one buffer for each axis. If aborting is specified, the instruction that was executed last is executed immediately, so it is not buffered.



Precautions for Correct Use

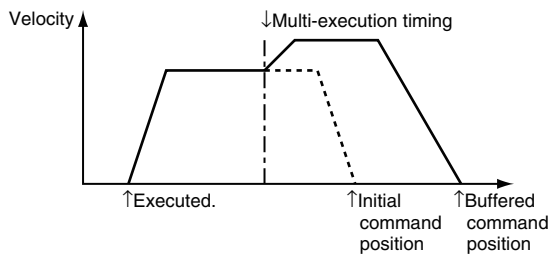
- Only one multi-execution instruction is buffered for each axis. If multi-execution is performed for two or more instructions, an instruction error will occur.
- Multi-execution of multi-axes coordinated control instructions (axes group instructions) is not possible for axes operating as a single axis. Similarly, multi-execution of single-axis control instructions is not possible for axes operating under multi-axes coordinated control (axes group instructions). An instruction error will occur if these rules are broken.

Aborting

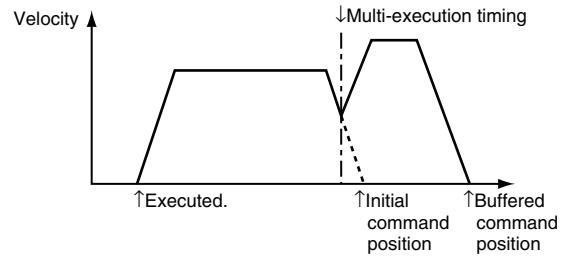
This is the default mode. No buffering is performed in this mode. The current command is aborted and the new instruction is executed. Aborting Mode can be used for multi-execution of instructions for motion control instructions for both single-axis control and synchronized control.

● When a Reverse Turn Does Not Occur for the Command Position of the Multi-execution Instruction

Executing More than One Instruction during Constant-velocity Motion



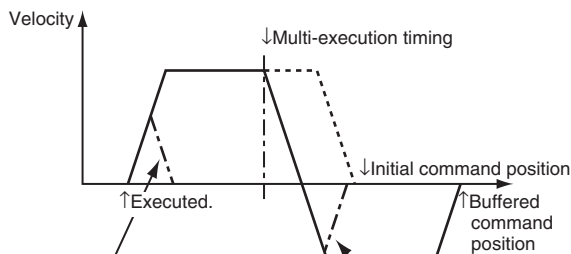
Multi-execution during Acceleration/Deceleration



If you use multi-execution of an instruction during triangular control or during deceleration, the axis will accelerate to the target velocity of the buffered instruction. In some cases, the axis will not reach the target velocity.

● When a Reverse Turn Occurs for the Command Position of the Multi-execution Instruction

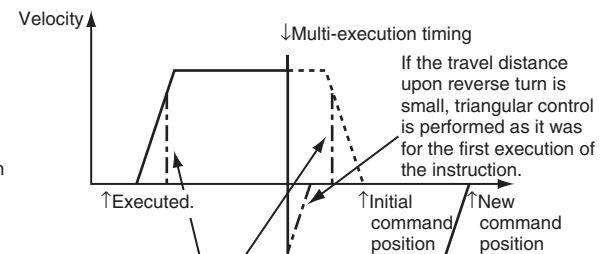
Decelerating to a Stop after Reversing



If an instruction is executed with multi-execution of instructions during acceleration, the axis starts deceleration according to the multi-execution timing.

If the travel distance upon reverse turn is small, triangular control is performed as it was for the first execution of the instruction.

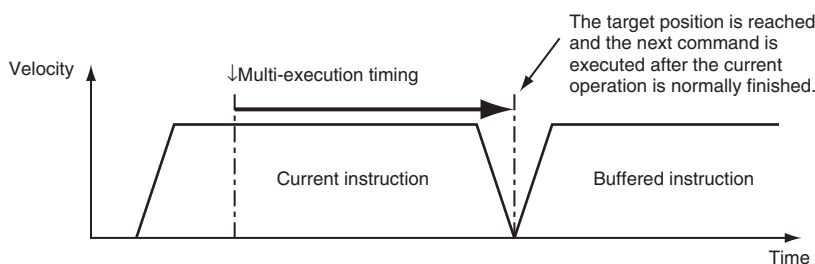
Stopping Immediately after Reversing



If the instruction is executed with multi-execution of instructions during acceleration or deceleration, the axis stops immediately according to the multi-execution timing. This also occurs during deceleration.

Buffered

The buffered instruction remains in the buffer until the operation of the current instruction is finished. The buffered instruction is executed after the operation for the current instruction is normally ended.



Blending

The buffered instruction remains in the buffer until the target position of the current instruction is reached. The buffered instruction is executed after the current instruction's target position is reached. However, motion does not stop at this time. Operation transitions to the next instruction at the velocity specified with the *BufferMode* (Buffer Mode Selection) input variable. For relative travel, the final position will be the total of the values for both instructions. For absolute travel, the final position will be the target position of the second multi-execution instruction. The Acceleration/Deceleration Over axis parameter is used to select one of the following operations for when the target position would be exceeded with the values that are set in the Maximum Acceleration and Maximum Deceleration axis parameters.

- Use rapid acceleration/deceleration. (Blending is changed to Buffered.)
- Use rapid acceleration/deceleration.
- Minor fault stop



Precautions for Correct Use

- In a blending mode, you cannot combine single-axis and synchronized control.
- Blending is not changed to Buffered even if you select *Use rapid acceleration/deceleration*. (*Blending is changed to Buffered.*). In this case, the maximum acceleration/deceleration rate is used and the blending operation is continued.

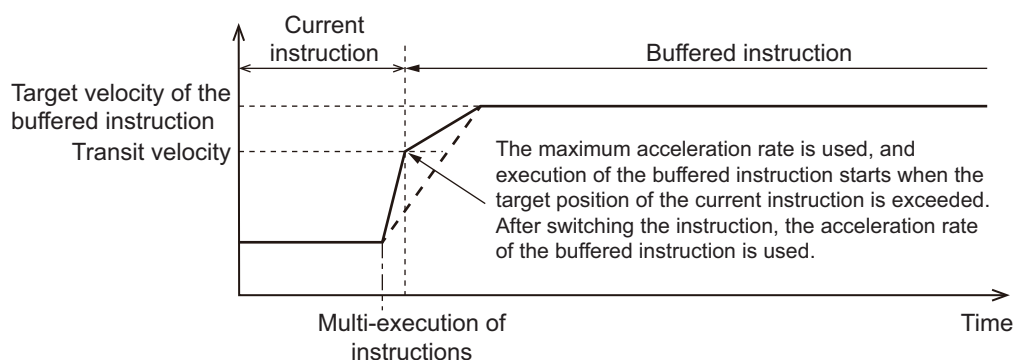
Also, the axis does not stop with an error even if you select *Minor fault stop*. Similar to the previous case, the maximum acceleration/deceleration rate is used and the blending operation is continued.

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

An example for an Acceleration/Deceleration Over operation is given below.

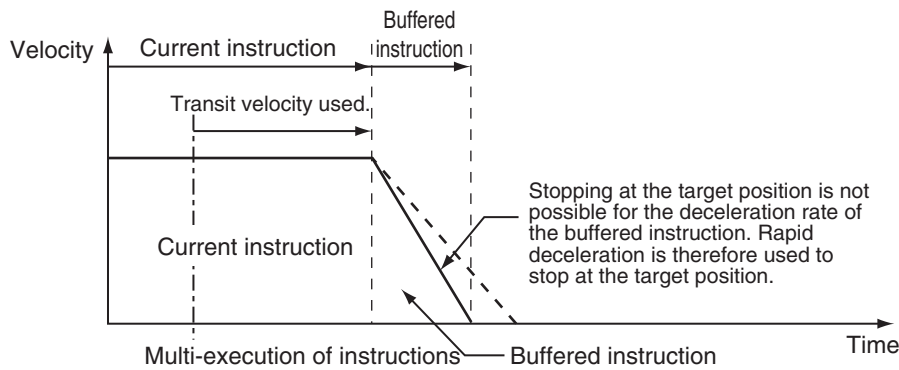
● Use Rapid Acceleration/Deceleration (Blending Is Changed to Buffered)

- The operation with the following setting is shown below.
The operation will be the same even if you select *Minor fault stop*.
Here, BufferMode is set to Blending Next.

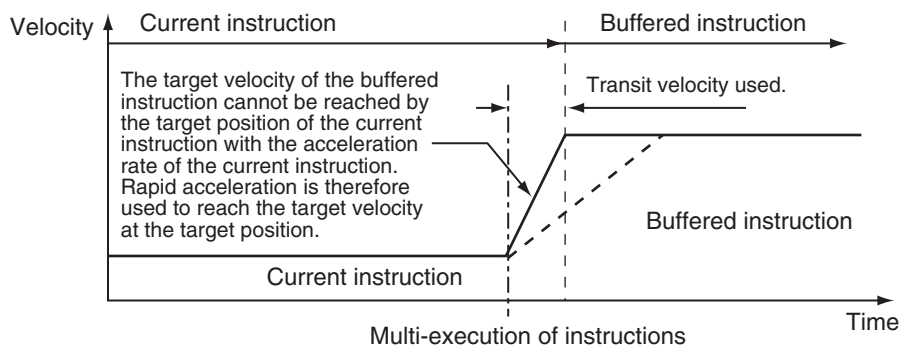


● Use Rapid Acceleration/Deceleration

- BufferMode Is Set to Blending Previous



- BufferMode Is Set to Blending Next



● Minor Fault Stop

- The operation is the same as when *Use rapid acceleration/deceleration*. (*Blending is changed to Buffered.*) is selected.

Blending Low (Low Velocity)

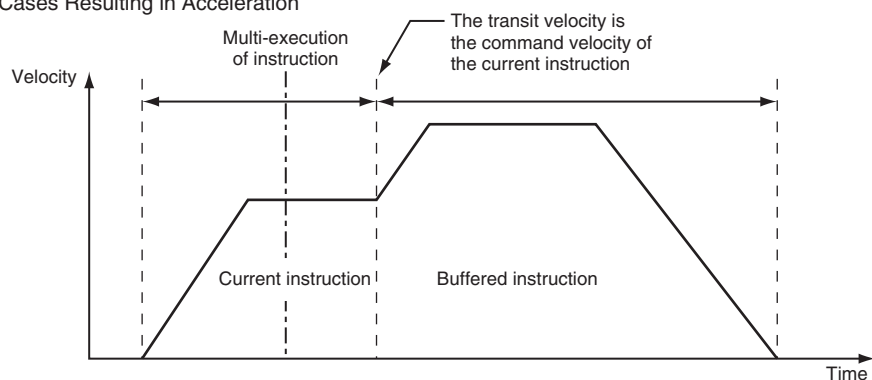
Operation is performed using the target position of the current instruction and the target velocity that is the slower of the target velocities for the current instruction and buffered instruction.

Blending Previous (Previous Velocity)

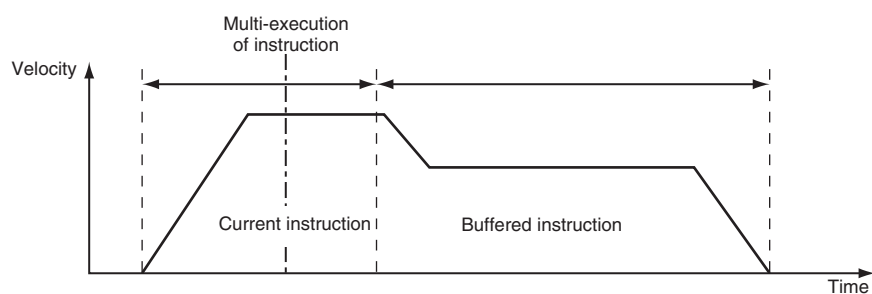
Operation is performed with the target velocity of the current instruction until the target position of the current instruction is reached. Operation is performed after acceleration/deceleration to the target velocity of the buffered instruction once the target position is reached.

● When the Direction of Operation Does Not Change

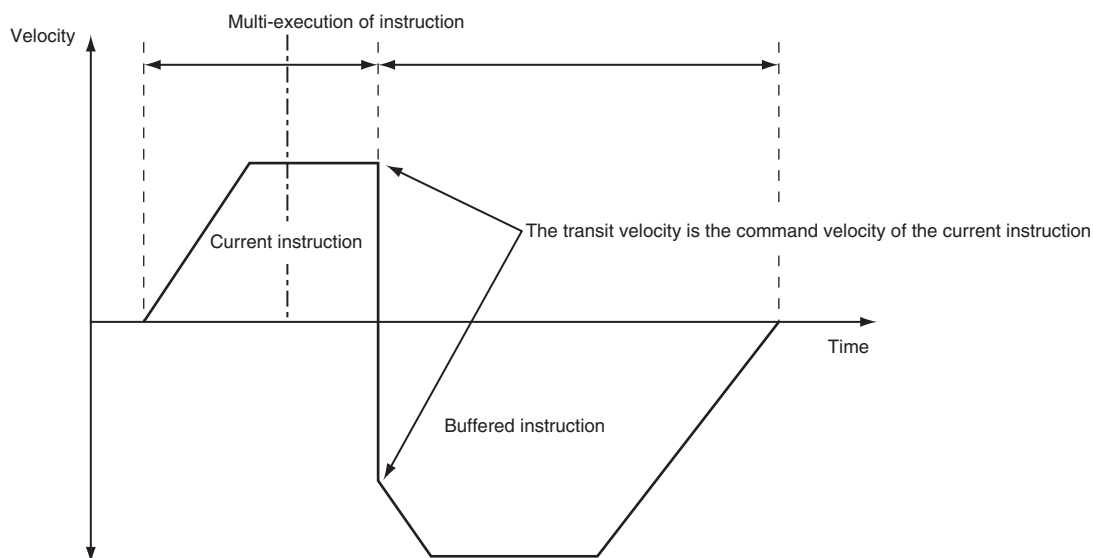
Cases Resulting in Acceleration



Cases Resulting in Deceleration

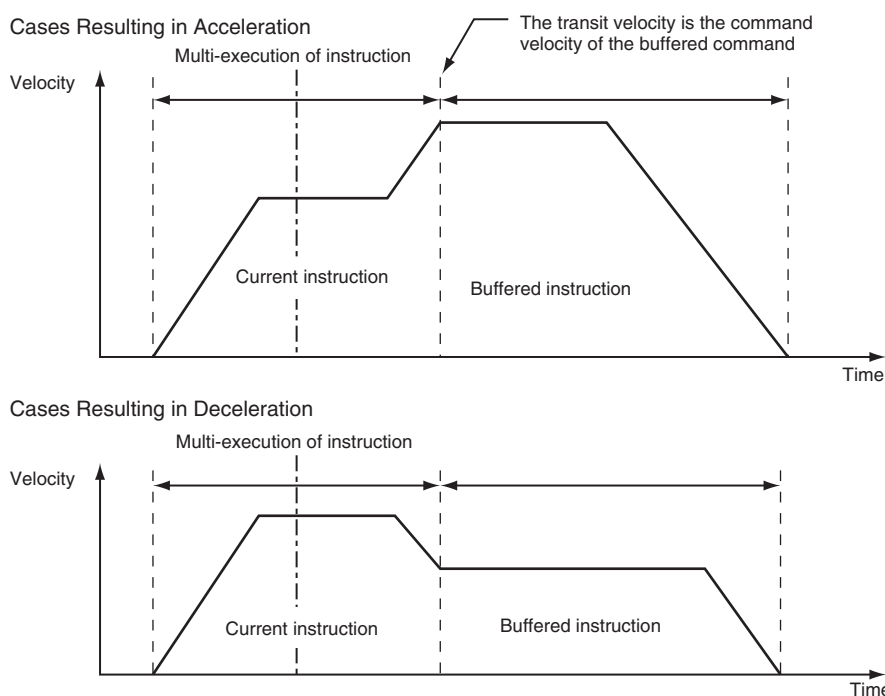


● When the Direction of Operation Changes



Blending Next (Next Velocity)

Operation is performed using the target position of the current instruction and the target velocity of the buffered instruction.



Blending High (High Velocity)

Operation is performed using the target position of the current instruction and the target velocity that is the faster of the target velocities for the current instruction and buffered instruction.

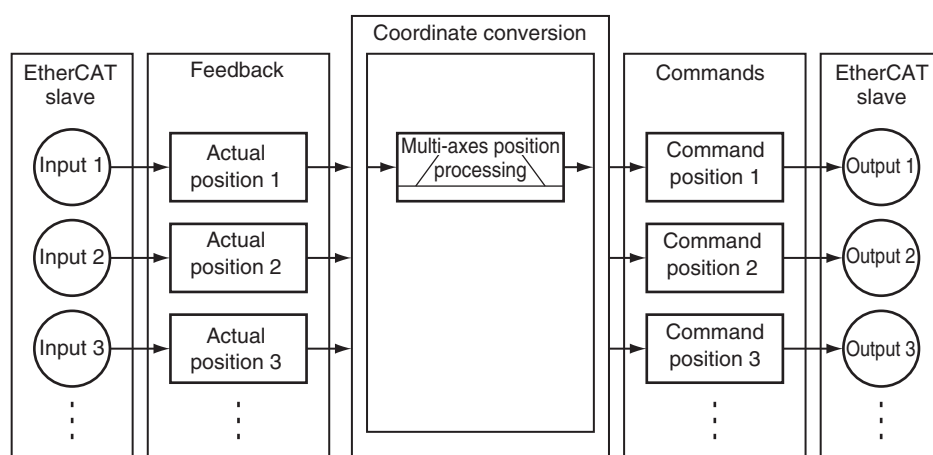
6-6 Multi-axes Coordinated Control

This section describes the operation of multi-axes coordinated control. With the MC Function Module, you can set an axes group in advance from the Sysmac Studio to perform interpolation control for multiple axes.

6-6-1 Outline of Operation

Multi-axes coordinated control performs a motion with multiple related axes together as a single group to control the path of the target control object. The MC Function Module treats all axes that perform coordinated operation as an axes group. Axes groups are set from the Sysmac Studio. In the user program, turn ON the Servo for each axis and then enable the axes group that is going to perform the multi-axes coordinated control. The purpose of multi-axes coordinated control is the coordinated operation of all axes belonging to the target axes group. Therefore, you cannot execute any single-axis operation motion control instructions on the axes in an enabled axes group. Furthermore, if any error occurs for any axis in an axes group, all axes in the axes group will stop according to the setting of the Axis Group Stop Method group axes parameter.

The MC Function Module can perform linear interpolation with two to four axes or circular interpolation with two axes.



Additional Information

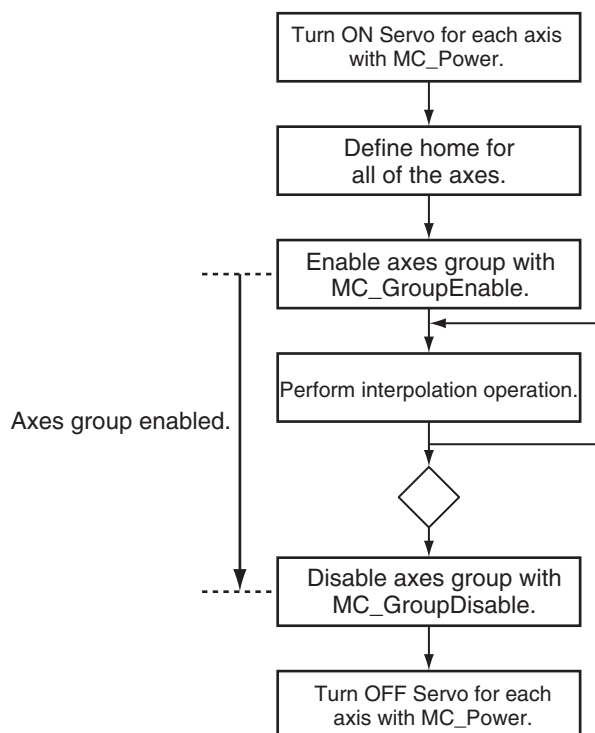
For devices that require you to modify the grouping of axes in motion to perform interpolation control, you must create multiple axes groups that include the axes to modify from the Sysmac Studio beforehand. After completing this step, you can execute by specifying the enabled axes groups from the user program during operation.

You can also use the MC_ChangeAxesInGroup (Change Axes in Group) instruction to change the composition axes for an axes group that is disabled.

For details on axes groups, refer to the *NJ/NX-series CPU Unit Motion Control User's Manual* (Cat. No. W507).

Enabling and Disabling Axes Groups

To enable an axes group, specify the axes group for the MC_GroupEnable (Enable Axes Group) instruction. An instruction error will occur if you try to execute an axes group instruction when the axes group is still disabled. To disable an axes group, specify the axes group for the MC_GroupDisable (Disable Axes Group) instruction. When you disable an axes group that is in operation, all axes in that axes group will decelerate to a stop at the maximum deceleration rate that is specified in their axis parameter settings.



For details on enabling and disabling axes groups, refer to the MC_GroupEnable (Enable Axes Group) and MC_GroupDisable (Disable Axes Group) instructions in the *NJ/NX-series Motion Control Instructions Reference Manual* (Cat. No. W508).

Changing the Axes in an Axes Group

You can use the MC_ChangeAxesInGroup (Change Axes in Group) instruction to temporarily change the composition axes for an axes group that is disabled. If the axes group is enabled, use the MC_GroupDisable (Disable Axes Group) instruction to disable the axes group before you change the composition axes.



Precautions for Correct Use

Changes made using the MC_ChangeAxesInGroup (Change Axes in Group) instruction will not be saved to non-volatile memory in the CPU Unit. If you cycle the power supply or download the settings from the Sysmac Studio, the parameter settings in the non-volatile memory are restored.

For details on changing the composition axes of an axes group, refer to the MC_ChangeAxesInGroup (Change Axes in Group) instruction in the *NJ/NX-series Motion Control Instructions Reference Manual* (Cat. No. W508).

Reading Axes Group Positions

You can use the MC_GroupReadPosition (Read Axes Group Position) instruction to read the command current positions and the actual current positions of an axes group.

For details on reading the axis positions for an axes group, refer to the MC_GroupReadPosition (Read Axes Group Position) instruction in the *NJ/NX-series Motion Control Instructions Reference Manual* (Cat. No. W508).

Resetting Axes Group Errors

If an error occurs in an axes group, you can use the MC_GroupReset instruction to remove the error once you have eliminated the cause.

For details on resetting axes group errors, refer to the MC_GroupReset (Group Reset) instruction in the *NJ/NX-series Motion Control Instructions Reference Manual* (Cat. No. W508).

Refer to the *NX-series Position Interface Units User's Manual* (Cat. No. W524) for the differences when you use NX-series Pulse Output Units.

6-6-2 Linear Interpolation

Linear interpolation is used to move 2 to 4 of the logical axes A0 to A3 in a straight line between a start point and an end point. Either absolute or relative positioning is possible. You can specify the interpolation velocity, interpolation acceleration, interpolation deceleration, and jerk.

The MC Function Modules uses the following three kinds of linear interpolation instructions.

- MC_MoveLinear (Linear Interpolation)

You can specify the *MoveMode* input variable to select between linear interpolation to an absolute value or linear interpolation to a relative value. This instruction is unique to the MC Function Module.

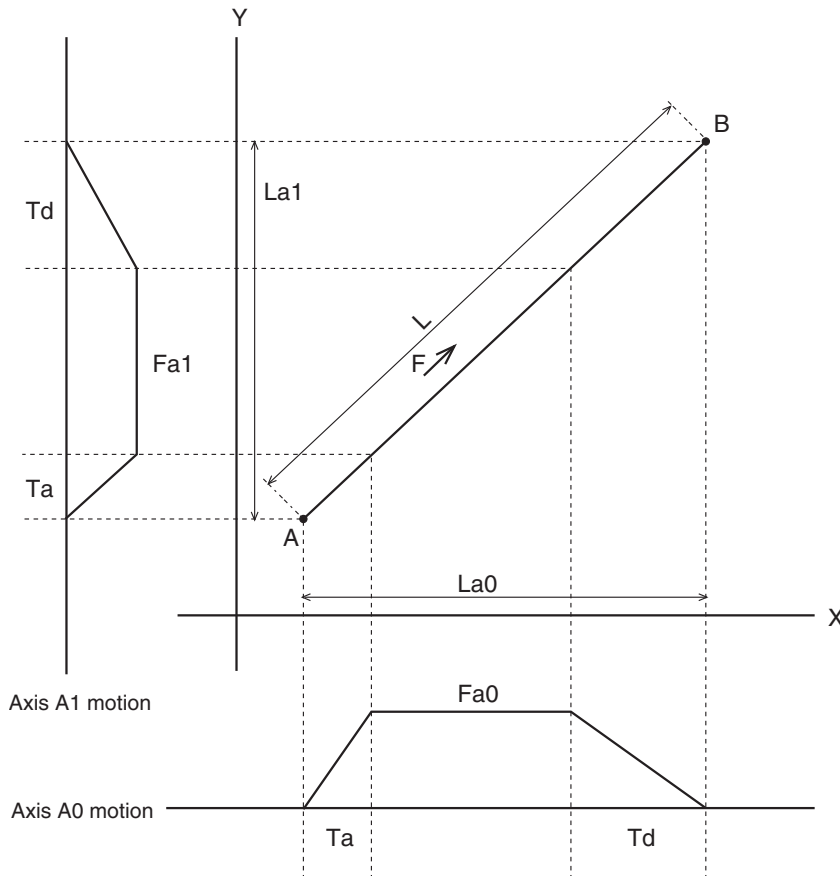
- MC_MoveLinearAbsolute (Absolute Linear Interpolation)

This instruction performs linear interpolation to an absolute value. This instruction is defined in the PLCopen® technical specifications.

- MC_MoveLinearRelative (Relative Linear Interpolation)

This instruction performs linear interpolation to a relative value. This instruction is defined in the PLCopen® technical specifications.

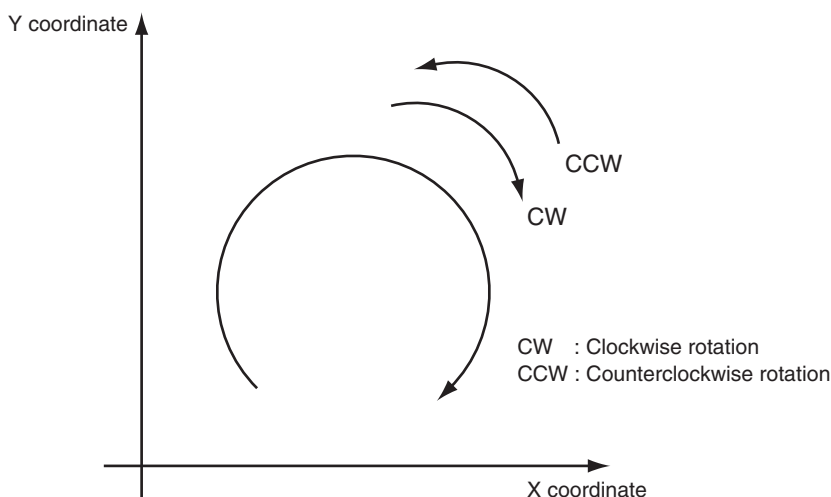
The following figure shows linear interpolation of 2 axes from point A to point B.



For details on linear interpolation, refer to the MC_MoveLinear (Linear Interpolation), MC_MoveLinearAbsolute (Absolute Linear Interpolation), and MC_MoveLinearRelative (Relative Linear Interpolation) instructions in the *NJ/NX-series Motion Control Instructions Reference Manual* (Cat. No. W508).

6-6-3 Circular Interpolation

Circular interpolation is used to move two of the logical axes A0 to A3 in a circular motion on a 2D plane. Either absolute or relative positioning is possible. You can specify the circular interpolation mode, path direction, interpolation velocity, interpolation acceleration, interpolation deceleration, and combined jerk for the two axes.



With the MC Function Module, you can specify the following three kinds of circular interpolation methods with the input variable *CircMode* (Circular Interpolation Mode).

- Border point
- Center
- Radius



Precautions for Correct Use

Set the Count Mode to Linear Mode for the axis that you use for circular interpolation. If the instruction is executed with this axis in Rotary Mode, an instruction error will occur.

6-6-4 Axes Group Cyclic Synchronous Positioning

You can cyclically output specified target positions for the axes in an axes group. You can specify target positions that are calculated in the user program as absolute positions to move the axes in any desired path.

For details on axes group cyclic synchronous positioning for an axes group, refer to the MC_GroupSyncMoveAbsolute (Axes Group Cyclic Synchronous Absolute Positioning) instruction in the *NJ/NX-series Motion Control Instructions Reference Manual* (Cat. No. W508).

6-6-5 Stopping Under Multi-axes Coordinated Control

Multi-axes coordinated control of axes groups will stop when you execute certain motion control instructions in the user program or when an error or some other problem occurs.

Stopping with Motion Control Instructions

Use the MC_GroupStop or MC_GroupImmediateStop instruction to stop axes group operation.

● MC_GroupStop Instruction

For linear interpolation or circular interpolation performed on an axes group, you can decelerate to a stop along the control path. You specify the deceleration rate and jerk. Specify a deceleration rate of 0 to send a command that immediately stops the Servo Drive or other device. Other operation commands are not acknowledged while decelerating to a stop for this instruction and while the input variable *Execute* is TRUE.

● MC_GroupImmediateStop Instruction

You can perform an immediate stop for all axes in the axes group. The immediate stopping method is determined by the setting of the Immediate Stop Input Stop Method axis parameter for each axis. The MC_GroupImmediateStop instruction can also be executed for an axes group that is decelerating to a stop for an MC_GroupStop instruction.

For details, refer to the MC_GroupStop and MC_GroupImmediateStop instructions in the *NJ/NX-series Motion Control Instructions Reference Manual* (Cat. No. W508).

Stopping Due to Errors or Other Problems

● Stopping for Errors during Axes Group Motion

If an error that results in a deceleration stop occurs for any composition axis in the axes group during an axes group motion, all of the axes will decelerate to a stop on the interpolation path at the interpolation deceleration rate. The interpolation deceleration rate is determined by the deceleration rate that is specified for the controlling instruction. If an error that results in an immediate stop occurs for any composition axis in the axes group during an axes group motion, the other axes in the axes group will stop according to the setting of the Axes Group Stop Method parameter in the axes group parameters.

You can select one of the following stop methods for axes groups.

- Immediate stop
- Decelerate axes to a stop at maximum deceleration rate of the axes.
- Immediate stop and Servo OFF

● Stopping Due to Motion Control Period Exceeded Error

If motion control processing does not end within two periods, a Motion Control Period Exceeded error occurs. All axes stop immediately.



Precautions for Correct Use

When you use an NX701 CPU Unit and operate in the multi-motion, all axes in both tasks will stop immediately if a Motion Control Period Exceeded error occurs in either of the tasks.

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

● Stopping Due to Start of MC Test Run

All axes will decelerate to a stop at their maximum deceleration if a MC Test Run is started from the Sysmac Studio.

● Stopping Due to Change in CPU Unit Operating Mode

All axes will decelerate to a stop at their maximum deceleration when the CPU Unit operating mode changes.



Additional Information

-
- If you execute the MC_GroupDisable (Disable Axes Group) instruction during axes group operation, the axes in the group will decelerate to a stop at their maximum deceleration rates.
 - If you execute the MC_Stop instruction while an axes group is in operation, an error will occur for the axes and axes group and the axes group operation will decelerate to a stop with interpolation. The interpolation deceleration rate is determined by the deceleration rate that is specified for the controlling instruction.
 - When the input variable *Enable* to the MC_Power (Servo ON) instruction changes to FALSE during axes group motion, the MC Function Module immediately stops the command value for that axis and turns OFF the Servo. When the Servo is turned OFF, the Servo Drive or other device will operate according to the settings in the Servo Drive or other device. Other axes in that axes group will stop with the stop method that is set in the Axes Group Stop Method axes group parameter. An error will occur for the axes group if this happens.
 - When RUN mode changes to PROGRAM mode, any motion control instructions for current motions are aborted. The *CommandAborted* output variable from the instructions remain TRUE and the Servo remains ON.
 - If the operating mode returns to RUN mode while a deceleration stop is in progress after the operating mode changes from RUN to PROGRAM mode, the output variable *CommandAborted* from the current motion control instructions change to TRUE.
 - The save process will continue during a save for the MC_SaveCamTable Instruction.
 - The generation process will continue when generation of the cam table is in progress for the MC_GenerateCamTable (Generate Cam Table) instruction.
-

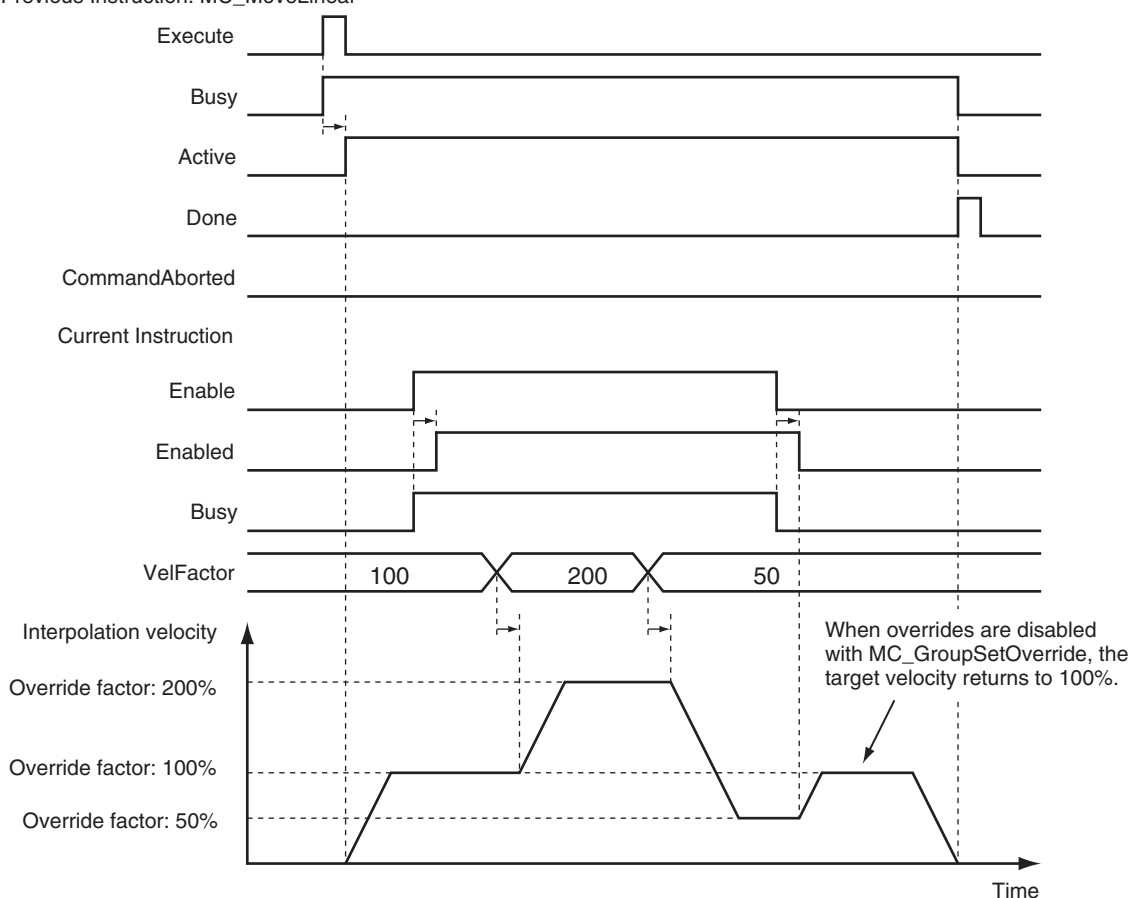
6-6-6 Overrides for Multi-axes Coordinated Control

You can use the MC_GroupSetOverride (Set Group Overrides) instruction to set override factors for multi-axes coordinated control of the axes group in the current interpolation operation. The velocity override factor is set as a percentage of the target velocity for interpolation. It can be set between 0% and 500%. If an override factor of 0% is set for the interpolation target velocity, operating status will continue with the axis stopped at a velocity of 0. The set override factor is read as long as the overrides are enabled. If the overrides are disabled, the override factors return to 100%. If the maximum interpolation velocity is exceeded when an override factor is changed, the maximum interpolation velocity for the axes group is used.

● Overrides for the MC_MoveLinear (Linear Interpolation) Instruction

An example of a time chart for using the Set Override Factors instruction for the MC_MoveLinear (Linear Interpolation) instruction is given below.

Previous Instruction: MC_MoveLinear



For details, refer to the MC_GroupSetOverride (Set Group Overrides) instruction in the *NJ/NX-series Motion Control Instructions Reference Manual* (Cat. No. W508).

6-7 Common Functions for Multi-axes Coordinated Control

This section describes the common functions for multi-axes coordinated control.

6-7-1 Velocity Under Multi-axes Coordinated Control

To specify the velocity for multi-axes coordinated control, specify the interpolation velocity on the path. The unit is the same as for single axes, command units/s.

Types of Velocities

The following is the only type of interpolation velocity for axes groups supported by the MC Function Module.

Velocity type	Definition
Command interpolation velocity	This is the actual value of the command interpolation velocity output by the MC Function Module to control an axes group.

Axis Parameters That Are Related to Velocities

Parameter name	Function	Setting range	Default
Maximum Interpolation Velocity	Set the maximum interpolation velocity for the path. Set 0 for no interpolation velocity limit. If a target velocity that exceeds the maximum interpolation velocity is specified for an axes group operation instruction, the axis will move at the maximum interpolation velocity.	Non-negative long reals	800,000,000
Interpolation Velocity Warning Value	Set the percentage of the maximum interpolation velocity at which to output an interpolation velocity warning. No interpolation velocity warning is output if 0 is set. (Unit: %)	0 to 100	0

Specifying Target Velocities for Axis Operations

The interpolation velocity used in an actual positioning motion is specified by the *Velocity* (Target Velocity) input variable to the motion control instruction.

Monitoring Velocities

You can read Axes Group Variables from the user program to monitor the interpolation velocity. In the descriptions, a variable name `_MC_GRP[*]` is used as an example, but the same information applies to `_MC1_GRP[*]` and `_MC2_GRP[*]`.

Variable name	Data type	Meaning	Function
<code>_MC_GRP[0-63].Cmd.Vel</code>	LREAL	Command Interpolation Velocity	This is the current value of the command interpolation velocity. A plus sign is added during travel in the positive direction, and a minus sign is added during travel in the negative direction.

6-7-2 Acceleration and Deceleration Under Multi-axes Coordinated Control

Multi-axes coordinated control performs control on the path for the interpolation acceleration and interpolation deceleration rates. The unit is the same as for single axes, command units/s².

Axis Parameters That Are Related to Interpolation Acceleration and Interpolation Deceleration

Parameter name	Function	Setting range	Default
Maximum Interpolation Acceleration	Set the maximum interpolation acceleration for the path. Set 0 for no interpolation acceleration limit. (Unit: command units/s ²)	Non-negative long reals	0
Maximum Interpolation Deceleration	Set the maximum interpolation deceleration for the path. Set 0 for no interpolation deceleration limit. (Unit: command units/s ²)	Non-negative long reals	0
Interpolation Acceleration/Deceleration Over	Set the operation for when the maximum interpolation acceleration/deceleration rate would be exceeded after excessive acceleration/deceleration during acceleration/deceleration control of the axes group because stopping at the target position is given priority.*1 0: Use rapid acceleration/deceleration. (Blending is changed to Buffered.) 1: Use rapid acceleration/deceleration. 2: Minor fault stop	0 to 2	0
Interpolation Acceleration Warning Value	Set the percentage of the maximum interpolation acceleration at which to output an interpolation acceleration warning. No interpolation acceleration warning is output if 0 is set. (Unit: %)	0 to 100	0

Parameter name	Function	Setting range	Default
Interpolation Deceleration Warning Value	Set the percentage of the maximum interpolation deceleration rate at which to output an interpolation deceleration warning. No interpolation deceleration warning is output if 0 is set. (Unit: %)	0 to 100	0

*1. Refer to 6-5-7 *Multi-execution of Motion Control Instructions (Buffer Mode)* on page 6-48 for operation with each set value.

Specifying an Interpolation Acceleration and Interpolation Deceleration for an Axes Group

The interpolation acceleration and interpolation deceleration rates used in an actual positioning motion are specified by the *Acceleration* (Acceleration Rate) and *Deceleration* (Deceleration Rate) input variables to the motion control instruction.

Monitoring Interpolation Acceleration and Interpolation Deceleration Rates

You can read Axes Group Variables in the user program to monitor interpolation acceleration and interpolation deceleration rates.

In the descriptions, a variable name `_MC_GRP[*]` is used as an example, but the same information applies to `_MC1_GRP[*]` and `_MC2_GRP[*]`.

Variable name	Data type	Meaning	Function
<code>_MC_GRP[0-63].Cmd.AccDec</code>	LREAL	Command Interpolation Acceleration/Deceleration	This is the current value of the command interpolation acceleration/deceleration rate. A plus sign is added for acceleration, and a minus sign is added for deceleration.

6-7-3 Jerk for Multi-axes Coordinated Control

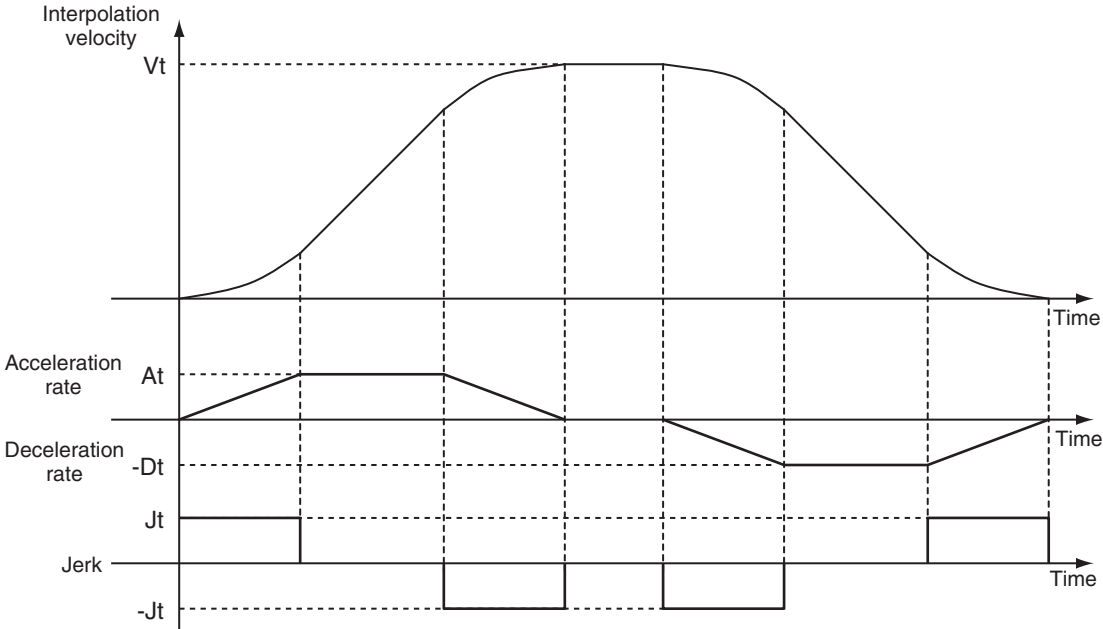
Jerk for multi-axes coordinated control is used to reduce shock and vibration on the machine by smoothing the interpolation acceleration/deceleration rate along the interpolation path into an S-curve. The unit is the same as for single axes, command units/s³.

Specifying Jerk for Axes Group Motion

The jerk used in an actual interpolation is specified by the *Jerk* input variable to the motion control instruction.

Jerk Example (Setting Other than 0)

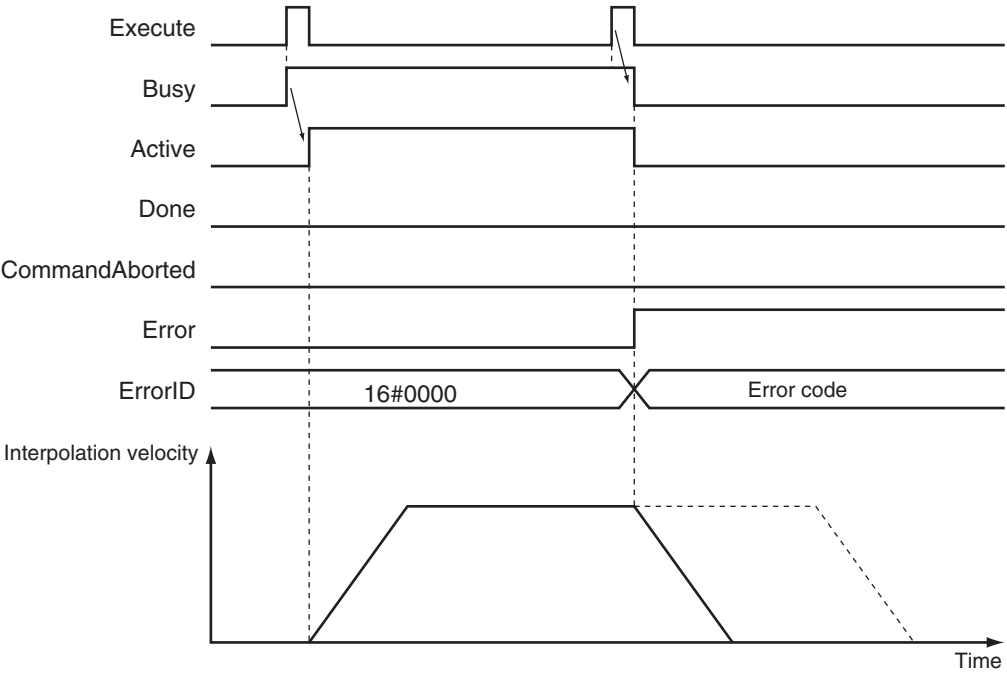
The acceleration/deceleration rate will change at a constant rate over the range where jerk is specified. The command interpolation velocity will form a smooth S-curve. A fixed interpolation acceleration rate is used in areas where the jerk is set to 0. This command interpolation velocity will form a straight line.



Vt: Specified interpolation velocity, At: Specified acceleration rate, Dt: Specified deceleration rate, Jt: Specified jerk

6-7-4 Re-executing Motion Control Instructions for Multi-axes Coordinated Control

If you re-execute a linear interpolation or circular interpolation instruction, an instruction error will occur.



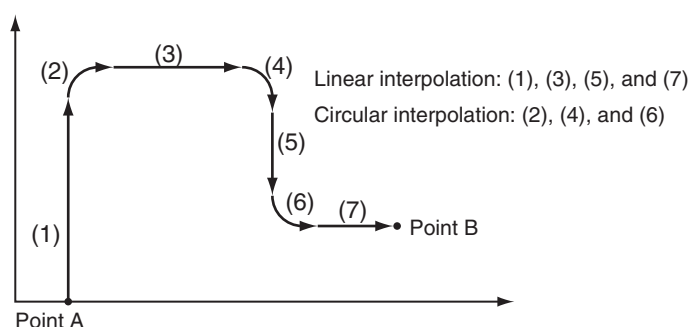
You can change the deceleration rate if you re-execute the MC_GroupStop instruction, but you cannot change the jerk in this way.

If you re-execute the MC_GroupReset instruction, the re-execution command will be ignored and error reset processing will continue.

For details on re-executing motion control instructions, refer to each instruction in the *NJ/NX-series Motion Control Instructions Reference Manual* (Cat. No. W508).

6-7-5 Multi-execution (Buffer Mode) of Motion Control Instructions for Multi-axes Coordinated Control

You can perform multi-execution for multi-axes coordinated control in axes groups the same way as you can for axis operations. You can perform path control for multiple continuous lines and/or arcs if you use Buffer Mode under multi-axes coordinated control.



You can set the *BufferMode* input variable to motion control instruction to select one of the same Buffer Modes as are supported for single-axis operations. There are a total of eight instruction buffers for axes groups. Each axes group has one buffer for the instruction currently in operation and seven buffers for multi-execution instructions. Multi-execution of instruction cannot be used from an axis operation instruction to an axes group operation instruction and vice-versa.



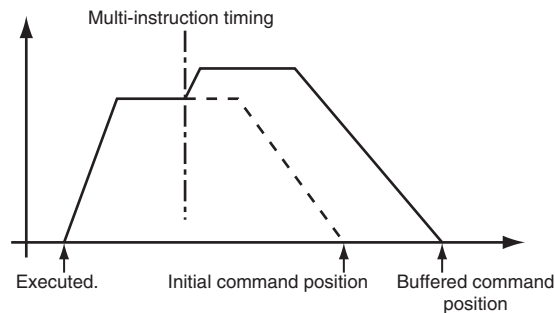
Precautions for Correct Use

- Up to seven instructions can be buffered at the same time for a single axes group. If multi-execution is performed for eight or more instructions, an instruction error will occur.
- Multi-execution of multi-axes coordinated control instructions (axes group instructions) is not possible for axes operating as a single axis. Similarly, multi-execution of single-axis control instructions is not possible for axes operating under multi-axes coordinated control (axes group instructions). An instruction error will occur if these rules are broken.

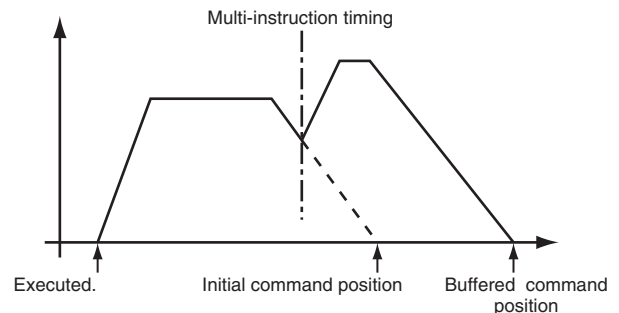
Aborting

This is the default mode. No buffering is performed in this mode. The current command is aborted and the new instruction is executed. Multi-execution of motion control instructions that have no *BufferMode* input variable will operate in Aborting Mode. Operation of the multi-execution instruction starts at the current interpolation velocity when the multi-execution instruction is executed. With Aborting Mode you cannot combine single-axis control, including synchronized single-axis control and axes group control. An instruction error will occur at the time of multi-execution if you execute an axes group operation on an axis currently in a single-axis motion. This will stop both the axes group and the single axis.

Multi-execution during Constant-velocity Motion

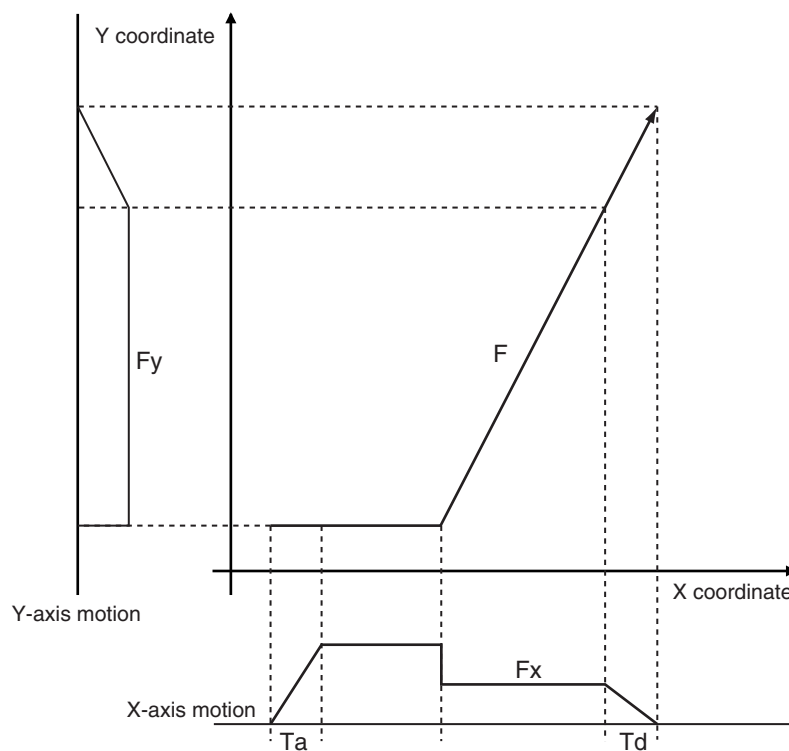


Multi-execution during Acceleration/Deceleration



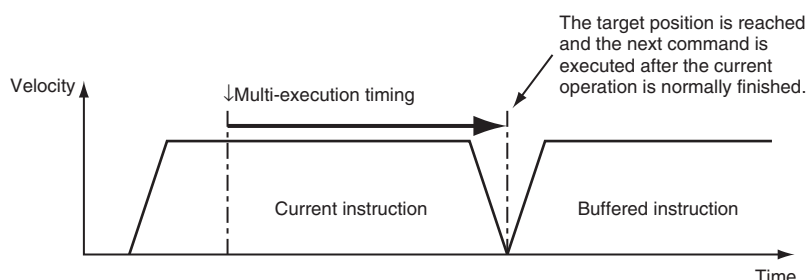
Multi-execution for axes groups is done so that the interpolation velocity remains continuous between instructions. If continuous operation is performed with an instruction with a travel distance of 0, the velocity changes for the axes will not be continuous.

● Example: Interpolation Velocity and Velocities of Axes for Two-axis Cartesian Coordinates



Buffered

The multi-execution instruction remains in the buffer until the current operation is finished. The buffered instruction is executed after the operation for the current instruction is normally ended.



Blending

Blending for axes groups works in the same way as blending for single-axis operations. The buffered instruction remains in the buffer until the target position of the current instruction is reached. The buffered instruction is executed after the target position of the current instruction is reached. The axes do not stop at the target position. The two motions are blended together at the interpolation velocity specified with the *BufferMode* input variable.

The Interpolation Acceleration/Deceleration Over axes group parameter is used to select one of the following operations for when the acceleration/deceleration that is specified in the buffered instruction would exceed the target position.

- Use rapid acceleration/deceleration. (Blending is changed to Buffered.)
- Use rapid acceleration/deceleration.
- Minor fault stop

Refer to 6-5-7 *Multi-execution of Motion Control Instructions (Buffer Mode)* on page 6-48 for operation with each set value.

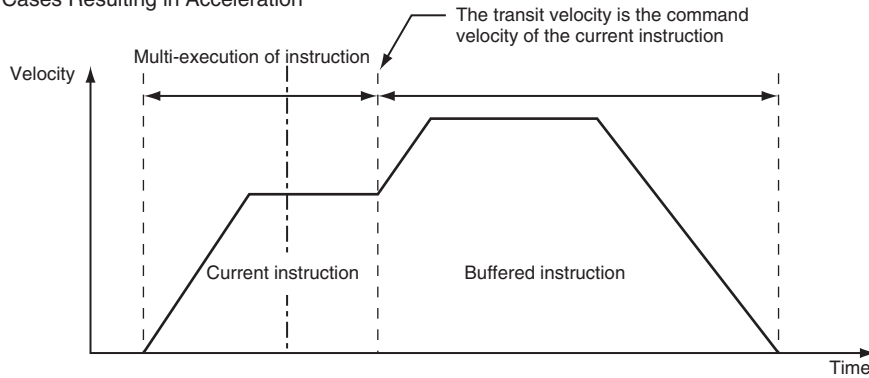
● Blending Low (Low Velocity)

Operation is performed using the target position of the current instruction and the target velocity that is the slower of the target velocities for the current instruction and buffered instruction.

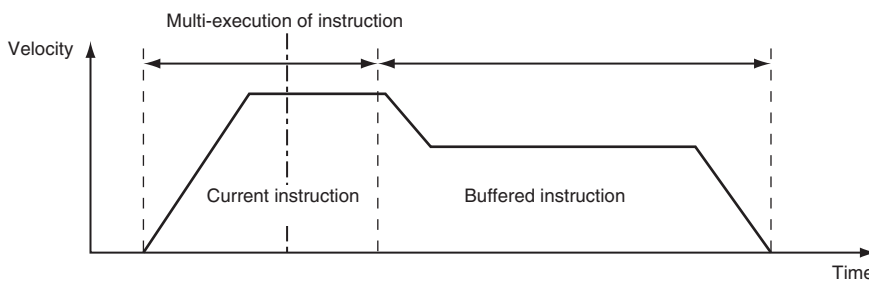
● Blending Previous (Previous Velocity)

Operation is performed with the target velocity of the current instruction until the target position of the current instruction is reached. Operation is performed after acceleration/deceleration to the target velocity of the buffered instruction once the target position is reached.

Cases Resulting in Acceleration



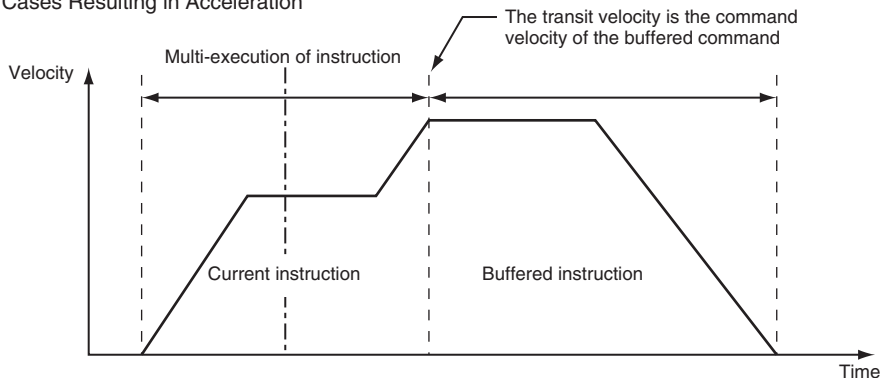
Cases Resulting in Deceleration



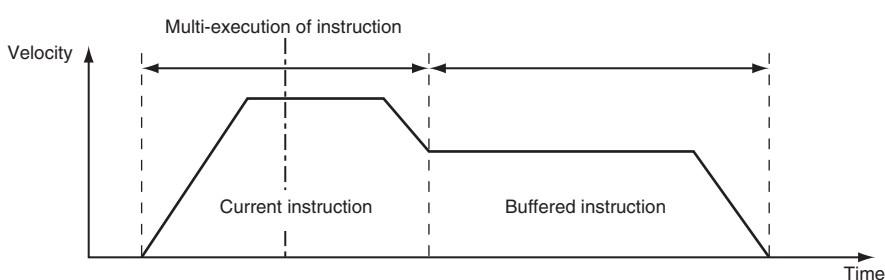
● Blending Next (Next Velocity)

Operation is performed using the target position of the current instruction and the target velocity of the buffered instruction.

Cases Resulting in Acceleration



Cases Resulting in Deceleration



● Blending High (High Velocity)

Operation is performed using the target position of the current instruction and the target velocity that is the faster of the target velocities for the current instruction and buffered instruction.

Transition Modes

Multi-execution of instructions for axes groups may create some shock on the device and/or workpiece due to changes in the direction of the interpolation path. You can specify the *TransitionMode* input variable to the motion control instruction to select a transition method to use between instructions in order to lessen this shock. You can choose from the following transition modes in the MC Function Module.

No.	Transition mode	Description
0	Transition Disabled (<i>_mcTMNone</i>)	Do not perform any processing for transitions (default). No attempt is made to lessen the shock, but this results in a shorter operation time.
10	Superimpose Corners (<i>_mcT-MCornerSuperimposed</i>)	The deceleration of the current instruction is superimposed on the acceleration of the buffered instruction. You can keep the linear velocity of the interpolation path constant.



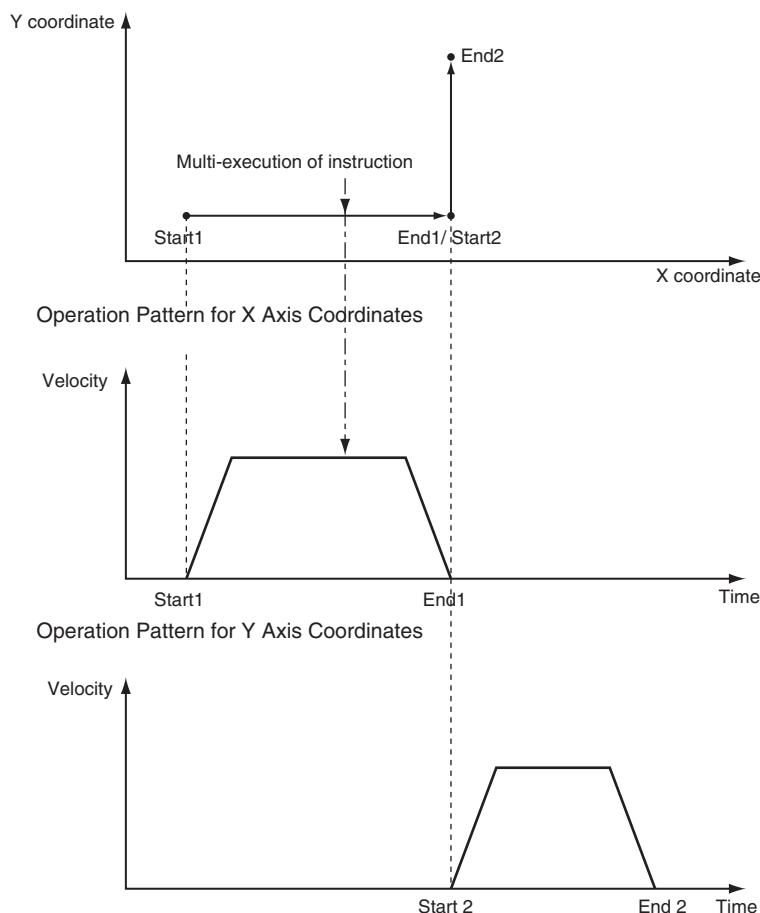
Additional Information

The PLCopen® technology specifications define numbers 0 through 9. Number 10 is unique to the MC Function Module.

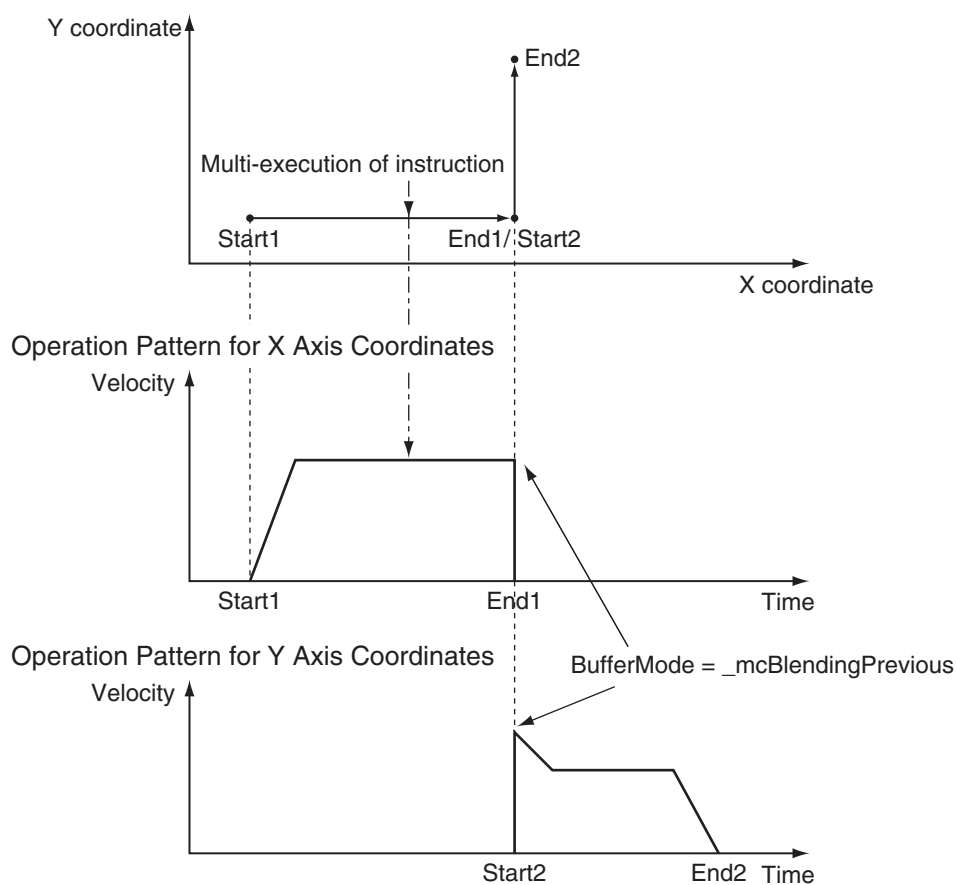
● Transition Disabled (0: *_mcTMNone*)

No processing is performed to connect the two positions.

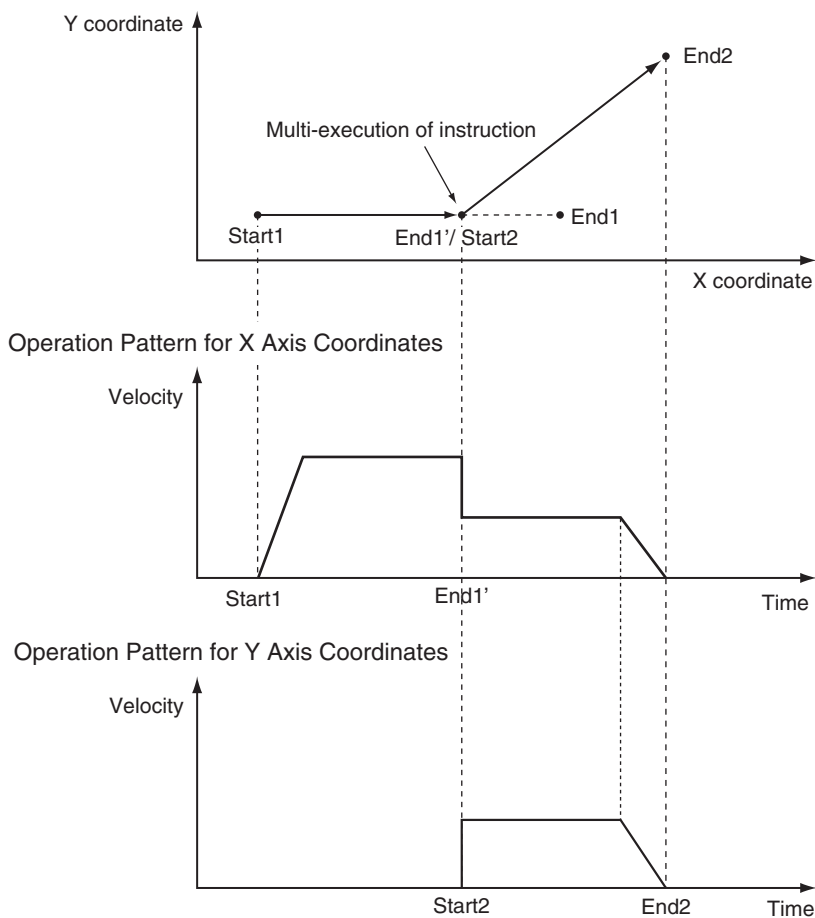
- TransitionMode = *_mcTMNone* and BufferMode = *_mcBuffered*
The axis moves to position End1, stops, and then moves to position End2.



- TransitionMode = *_mcTMNone* and BufferMode = *_mcBlending*
The axis moves to position End1, and then moves to position End2.

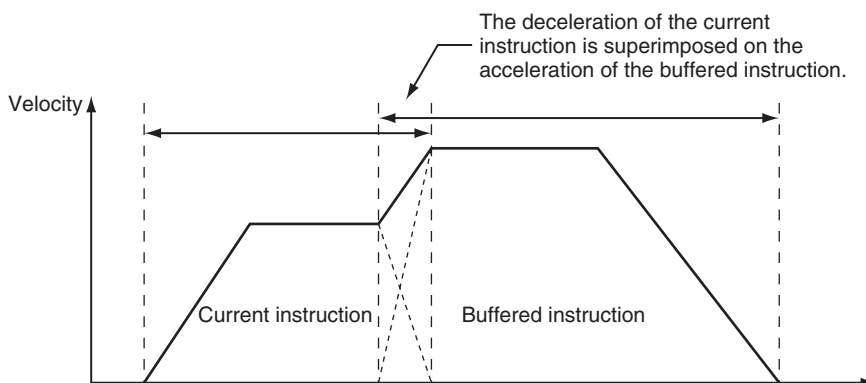


- TransitionMode = `_mcTMNone` and BufferMode = `_mcAborting`
The axis moves from End1' (multi-execution of instruction) to End2.



● Superimpose Corners (10: `_mcTMCornerSuperimposed`)

The deceleration of the current instruction is superimposed on the acceleration of the buffered instruction. Operation is executed in the same amount of time as for the deceleration of the current instruction, no matter what is specified as the acceleration for the buffered instruction. The superimposed area will apply no jerk even if jerk is specified.



The output variable *Done*, which indicates the end of a motion control instruction, will change to TRUE for `_mcTMCornerSuperimposed` when the area of superimposition is completed.



Additional Information

The path linear velocity is constant if the following two conditions are met.

- The target velocities of the current instruction and the buffered instruction are the same.
- The deceleration rate of the current instruction and the acceleration rate of the buffered instruction are the same.

Combining Transition Modes and Multi-execution of Instructions

The following table shows the combinations of Transition Modes and Buffer Modes.

OK: Operation possible. ---: Generates an error and stops.

Transition Mode	Buffer Mode					
	Aborting	Buffered	Blending Low	Blending Previous	Blending Next	Blending High
Transition Disabled (<i>_mcTM-None</i>)	OK	OK	OK	OK	OK	OK
Superimpose Corners ^{*1} (<i>_mcTMCornerSuperimposed</i>)	---	---	OK	OK	OK	OK

*1. For superimpose corners, the deceleration for the current instruction and the acceleration for the buffered instruction will be superimposed.

6-8 Other Functions

This section describes other functions of the MC Function Module.

Refer to the *NX-series Position Interface Units User's Manual* (Cat. No. W524) for the differences when you use NX-series Pulse Output Units.

6-8-1 Changing the Current Position

The command current position of a Servo axis can be changed to a specified value. The actual current position changes to a value that maintains the current following error with the command current position. For an encoder axis, you can change the actual current position. Use the MC_SetPosition instruction to specify the actual position you want to modify.

You can change the actual position even while an axis is in motion. If positioning to an absolute value is being executed, positioning will be performed to the target position using the new absolute coordinates. However, the travel distance will stay the same when you position to a relative value.



Precautions for Correct Use

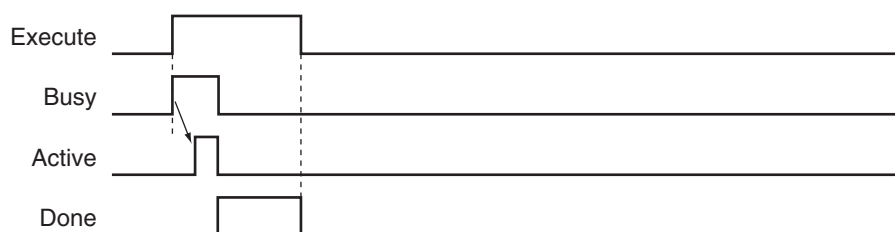
- When the Count Mode is Rotary Mode, an instruction error will occur if you specify a position outside the ring counter range.
- After changing the current position the home will be undefined and you will not be able to use the following functions and instructions.

Software limits

High-speed homing

Interpolation instructions (linear and circular interpolation)

● Timing Chart for Execution While Axis Is Stopped



Additional Information

You can change the actual position while home is defined by specifying a zero position preset for the MC_Home or MC_HomeWithParameter instruction.

For details on the MC_SetPosition instruction, refer to the *NJ/NX-series Motion Control Instructions Reference Manual* (Cat. No. W508).

6-8-2 Torque Limit

The output torque is limited by enabling and disabling the torque limit function of the Servo Drive and by setting the torque limit value.

Different limits can be specified for the positive torque limit and negative torque limit.

For details, refer to the MC_SetTorqueLimit instruction in the *NJ/NX-series Motion Control Instructions Reference Manual* (Cat. No. W508).



Precautions for Correct Use

You cannot use the torque limit function for an NX-series Pulse Output Unit.

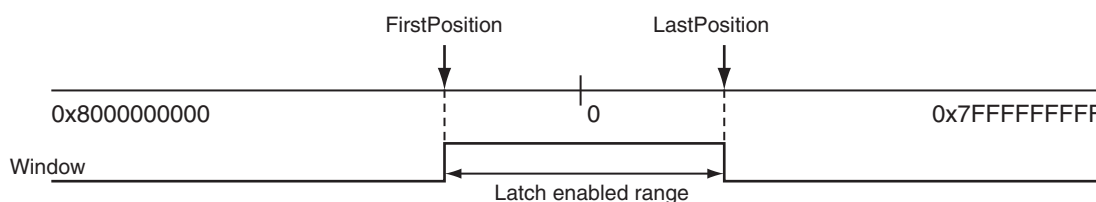
6-8-3 Latching

Latching is used to control positioning based on the position where a trigger signal occurs, such as a signal from a sensor input. The position of the axis is recorded (i.e., latched) when the trigger signal occurs. You can set up to two trigger signals for each axis. Use the MC_TouchProbe (Enable External Latch) instruction to specify the Trigger Input Condition variable, Window Only variable, and Stopping Mode Selection variable for the axis you want to latch. In addition to signals that connect to the Servo Drive, you can also specify variables in the user program to use as a trigger. Use the MC_AbortTrigger (Disable External Latch) instruction to abort latching. You can use latching only with a Servo Drive that support latching (touch probe), such as the OMRON G5-series Servo Drives, or a GX-EC0211/EC0241 Encoder Input Terminal.

Use *WindowOnly* to detect only trigger signals within a specific start point and end point. The following chart shows the ranges for different Count Modes.

● Linear Mode

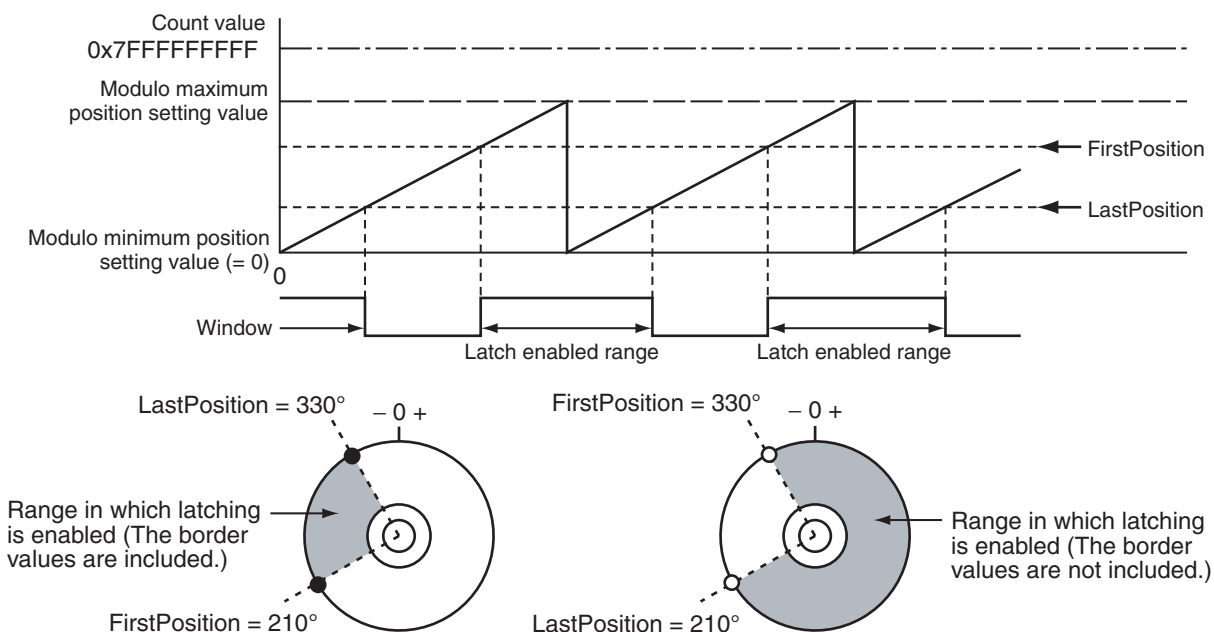
- The *FirstPosition* must be less than or equal to the *LastPosition*.
- An instruction error will occur if the *FirstPosition* is greater than the *LastPosition*.
- An instruction error will occur if a position beyond the position range of Linear Mode is specified.



● Rotary Mode

- The *FirstPosition* can be less than, equal to, or greater than the *LastPosition*. If the *FirstPosition* is greater than the *LastPosition*, the setting will straddle the modulo minimum position setting value.
- An instruction error will occur if a position beyond the upper and lower limits of the ring counter is specified.

	First Position $\leq$ Last Position	First Position $>$ Last Position
Valid range	FirstPosition to LastPosition	LastPosition to FirstPosition



For details on latching, refer to the MC_TouchProbe (Enable External Latch) and MC_AbortTrigger (Disable External Latch) instructions in the *NJ/NX-series Motion Control Instructions Reference Manual* (Cat. No. W508).

Refer to the *NX-series Position Interface Units User's Manual* (Cat. No. W524) for the differences when you use NX-series Pulse Output Units.

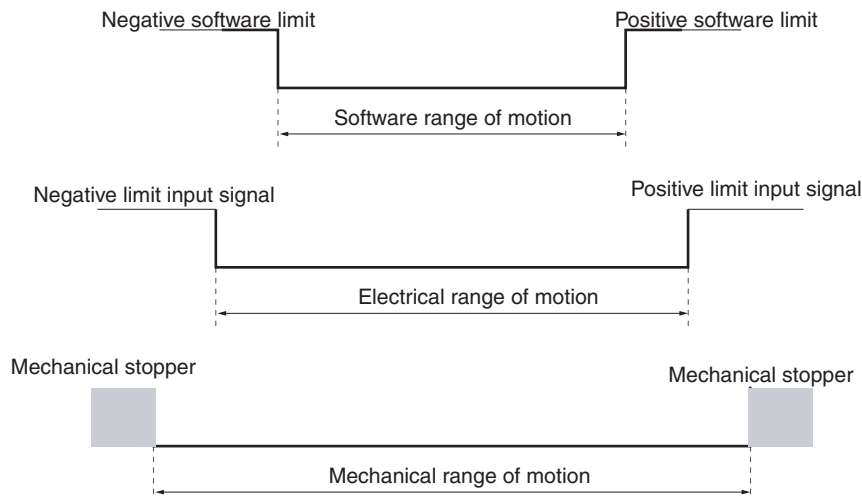
6-8-4 Zone Monitoring

This function detects whether the command position or actual position of an axis is in the specified range (zone). Use the MC_ZoneSwitch (Zone Monitor) instruction to specify the first position and last position of the zone to check. The *InZone* output variable for the Zone Monitor instruction will change to TRUE when the position of the axis enters the specified zone. You can also specify multiple zones for a single axis. Zones can overlap.

For details on zone monitoring, refer to the MC_ZoneSwitch (Zone Monitor) instruction in the *NJ/NX-series Motion Control Instructions Reference Manual* (Cat. No. W508).

6-8-5 Software Limits

Actual positions can be monitored in the MC Function Module software. This function is separate from the hardware-based limit input signals. Set the range to monitor by setting the software limits in the Positive Software Limit and Negative Software Limit axis parameters. During normal positioning, motion is possible within the range of these software limits. Set software limits to prevent potential damage to machinery caused by mistakes in the user program or improper operation.



● Axis Parameters That Are Related to Software Limits

Parameter name	Function	Setting range	Default
Software Limits	Select the software limit function. 0: Disabled 1: Deceleration stop for command position ^{*1} 2: Immediate stop for command position 3: Deceleration stop for actual position ^{*1} 4: Immediate stop for actual position	0 to 4	0: Disabled
Positive Software Limit	Set the software limit in the positive direction. The unit is command units.	Long reals ^{*2}	2,147,483,647
Negative Software Limit	Set the software limit in the negative direction. The unit is command units.		-2,147,483,648

^{*1}. If the actual position goes beyond a software limit during execution of a movement instruction that has a *Deceleration* input variable, the axis decelerates to a stop at the deceleration rate given by *Deceleration*. If the actual position goes beyond a software limit during execution of a movement instruction that does not have a *Deceleration* input variable, the axis decelerates to a stop at the maximum deceleration that is set in the axis parameters.

^{*2}. Positions can be set within a 40-bit signed integer range when converted to pulses.

You can use the axis settings of the Sysmac Studio, the MC_Write (Write MC Setting) instruction, or the MC_WriteAxisParameter (Write Axis Parameters) instruction to set the above axis parameters. If any setting values are changed for an axis or axes group in operation, those settings are enabled when the next operation begins.

Software limits function in the following two cases based on the axis operation state and the motion control instruction that is used.

● Executing Motion Instructions

- When the Actual Position Is within the Software Limits
An instruction error will occur if the target position is outside the software limit range.
- When the Actual Position Is outside the Software Limits
Motion is allowed only toward the software limit range. As long as the motion is toward the range, the target position does not need to be within the software limit range.



Precautions for Correct Use

Do not execute an instruction for an axis command for a target position that is outside of the software limit range.

● During Axis Motion

When the axis is in discrete motion, synchronized motion, continuous motion, or coordinated motion:

- An axis error will occur if the software limits are enabled for the command position and the command position leaves the range.
- An axis error will occur if the software limits are enabled for the actual position and the actual position leaves the range.



Additional Information

Software limits can be enabled when the Count Mode is set to Linear Mode and home is defined. Software limits are disabled in the following situations no matter what axis parameters have been set.

- When Count Mode is set to Rotary Mode.
- When home is not defined.
- During homing.

For details on the instruction to write the MC settings and the instruction to write the axis parameters, refer to the MC_Write instruction and MC_WriteAxisParameter instruction in the *NJ/NX-series Motion Control Instructions Reference Manual* (Cat. No. W508).

6-8-6 Following Error Monitoring

Following error is the difference between the command position and the actual position of an axis. The MC Function Module monitors the following error every motion control period.

If the value of the following error exceeds the Following Error Over Value that is set in the axes parameters, Following Error Limit Exceeded minor fault level error occurs. If it exceeds the Following Error Warning Value, a Following Error Warning observation occurs. Monitoring the following error is disabled during execution of the holding operation for homing.

● Axis Parameters That Are Related to Monitoring the Following Error

You can set the check values for monitoring the following error by setting the appropriate axis parameters. Set the Following Error Warning Value so that it is less than the Following Error Over Value.

Set the axis parameters from the Sysmac Studio.

Parameter name	Function	Setting range	Default
Following Error Over Value	Set the excessive following error check value. Set 0 to disable the excessive following error check. (Unit: command units)	Non-negative long reals	0
Following Error Warning Value	Set the following error warning check value. Set 0 to disable the following error warning check. (Unit: command units)	Non-negative long reals that are less than or equal to the Following Error Over Value	0

Refer to the *NX-series Position Interface Units User's Manual* (Cat. No. W524) for the differences when you use NX-series Pulse Output Units.

6-8-7 Following Error Counter Reset

Resetting the following error counter resets the following error to 0.

Use the MC_ResetFollowingError instruction in the user program to reset the following error counter. You can use the MC_ResetFollowingError instruction for each axis during positioning or during homing. If you execute a following error counter reset while the axis is in motion, the current motion control instruction will be aborted and the command position will be set to the same value as the actual position.

The home will remain defined even after executing a following error counter reset.

For details on resetting the following error counter, refer to the MC_ResetFollowingError instruction in the *NJ/NX-series Motion Control Instructions Reference Manual* (Cat. No. W508).

● Axis Parameters That Are Related to Resetting the Following Error Counter

You can choose to reset the following error counter on an immediate stop, on a limit input stop, or after homing is completed by setting the appropriate axis parameters. Set the axis parameters from the Sysmac Studio.

Parameter name	Function	Setting range	Default
Immediate Stop Input Stop Method	Set the stopping method in the MC Function Module when the immediate stop input is enabled. 0: Immediate stop 2: Immediate stop and error reset 3: Immediate stop and Servo OFF	0, 2, or 3	0
Limit Input Stop Method	Set the stopping method in the MC Function Module when the positive limit input or negative limit input is enabled. 0: Immediate stop 1: Deceleration stop 2: Immediate stop and error reset 3: Immediate stop and Servo OFF	0 to 3	0

Refer to the *NX-series Position Interface Units User's Manual* (Cat. No. W524) for the differences when you use NX-series Pulse Output Units.

6-8-8 Axis Following Error Monitoring

You can monitor the amount of following error for the command position or the actual position between two axes. Use the MC_AxesObserve (Monitor Axis Following Error) instruction to specify the permitted following error and the two axes to monitor. If the permitted following error is exceeded, the *Invalid* output variable for the Monitor Axis Following Error instruction will change to TRUE.

You can use this monitoring function to program the actions to take when the following error between axes grows too large for gantry control and other devices where both axes perform the same operation.



Precautions for Correct Use

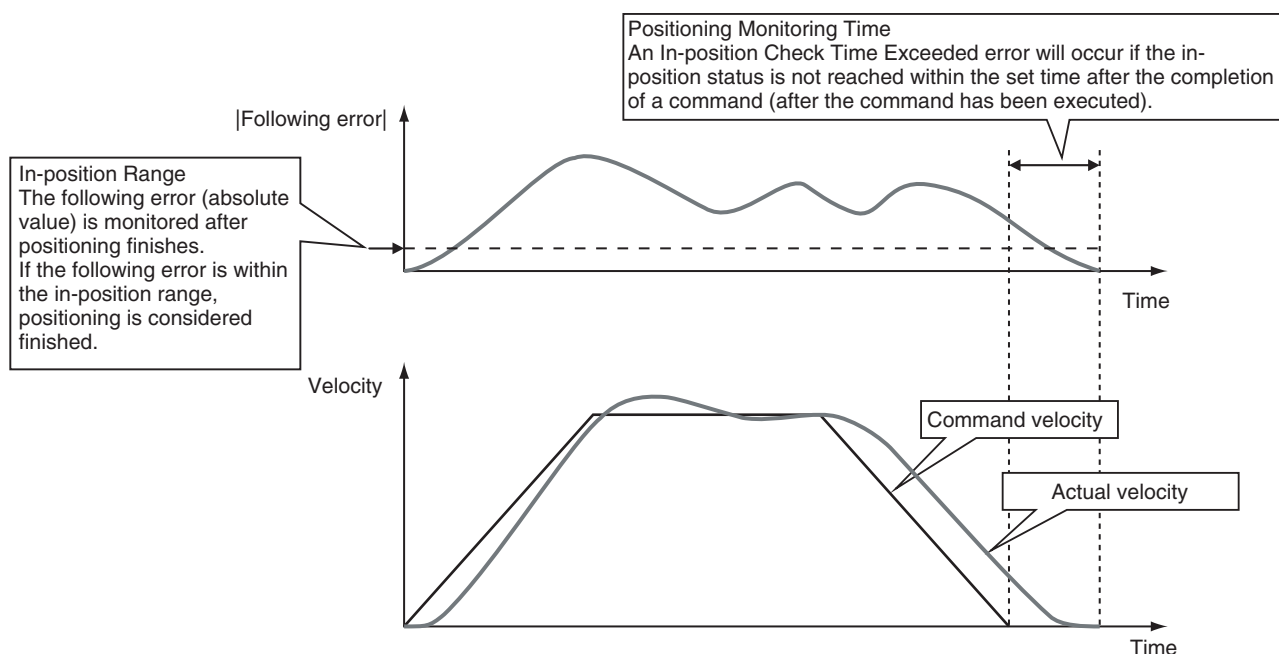
Even if the permitted following error between axes is exceeded, no error will occur in the MC Function Module. Check the *Invalid* output variable to stop axis operation or to take some other action as appropriate in the user program.

For details on axis following error monitoring, refer to the MC_AxesObserve (Monitor Axis Following Error) instruction in the *NJ/NX-series Motion Control Instructions Reference Manual* (Cat. No. W508).

Refer to the *NX-series Position Interface Units User's Manual* (Cat. No. W524) for the differences when you use NX-series Pulse Output Units.

6-8-9 In-position Check

You can check to see if the actual current position has reached the specified range for the target position during positioning or homing. After command output of the target position is completed, positioning is considered to be finished when the difference between the target position and the actual current position is within the in-position range. An instruction error occurs if the position is not within the in-position within the in-position check time.



● Axis Parameters That Are Related to In-position Checks

You can set the check conditions for the in-position check by setting the appropriate axis parameters. Set the in-position check time if you want to start any of the following operations only after confirming that axes are in position.

Parameter name	Function	Setting range	Default
In-position Range	Set the in-position width. (Unit: command units)	Non-negative long reals	10
In-position Check Time	Set the in-position check time in milliseconds. Set 0 to check for the end of positioning only when you define the home position during homing and not check positioning at other times. (Unit: ms)	0 to 10,000	0

You can use the axis settings of the Sysmac Studio, the MC_Write (Write MC Setting) instruction, or the MC_WriteAxisParameter (Write Axis Parameters) instruction to set the above axis parameters.



Additional Information

- The in-position check is processed by the MC Function Module. The function in the Servo Drive is not used.
- Do not set an in-position check time if you want to start the next operation as quickly as possible without waiting for positioning to finish.
- The value set from the Sysmac Studio is restored if power to the CPU Unit is cycled or the user program is downloaded with the Synchronization menu command of the Sysmac Studio. Use the MC_Write (Write MC Setting) and MC_WriteAxisParameter (Write Axis Parameters) instructions only when you need to temporarily change the in-position check time.

● Monitor Information That Is Related to In-position Checks

You can read Axis Variables from the user program to monitor when positioning finishes.

In the descriptions, a variable name `_MC_AX[*]` is used as an example, but the same information applies to `_MC1_AX[*]` and `_MC2_AX[*]`.

Variable name	Data type	Meaning	Function
<code>_MC_AX[0-255].Details.Idle</code>	BOOL	Idle	TRUE when processing is not currently performed for the command value, except when waiting for in-position state.*1 <i>Idle</i> and <i>InPosWaiting</i> are mutually exclusive. They cannot both be TRUE at the same time.
<code>_MC_AX[0-255].Details.InPosWaiting</code>	BOOL	In-position Waiting	TRUE when waiting for in-position state. The in-position check is performed when positioning for the in-position check.

*1. This also includes states where processing is performed while in motion at velocity 0, during following error counter resets, during synchronized control, and during coordinated motion.

You can read Axes Group Variables from the user program to monitor when positioning finishes for the axes group.

In the descriptions, a variable name `_MC_GRP[*]` is used as an example, but the same information applies to `_MC1_GRP[*]` and `_MC2_GRP[*]`.

Variable name	Data type	Meaning	Function
<code>_MC_GRP[0-63].Details.Idle</code>	BOOL	Idle	TRUE when processing is not currently performed for the command value, except when waiting for in-position state.* ¹ <i>Idle</i> and <i>InPosWaiting</i> are mutually exclusive. They cannot both be TRUE at the same time.
<code>_MC_GRP[0-63].Details.InposWaiting</code>	BOOL	In-position Waiting	TRUE when waiting for in-position state for any composition axis.* ² The in-position check is performed when positioning for the in-position check.

*1. This also includes states where processing is performed while in motion at a velocity of 0.

*2. This variable is FALSE when all composition axes in the axes group are within the in-position ranges set in the axis parameters.

For details on the instruction to write the MC settings and the instruction to write the axis parameters, refer to the `MC_Write` (Write MC Setting) and `MC_WriteAxisParameter` (Write Axis Parameters) instruction in the *NJ/NX-series Motion Control Instructions Reference Manual* (Cat. No. W508).

Refer to the *NX-series Position Interface Units User's Manual* (Cat. No. W524) for the differences when you use NX-series Pulse Output Units.

6-8-10 Changing Axis Use

You can use the `MC_ChangeAxisUse` (Change Axis Use) instruction to temporarily change the setting of the Axis Use axis parameter. To change an axis in this way, it must be set as a *Used axis* or as an *Unused axis (changeable to used axis)* in the Axis Use axis parameter. If the Axis Use axis parameter is set to *Unused axis (changeable to used axis)* and the Axis Type parameter is set to a servo axis or virtual servo axis, you can set the axis in an axes group.



Precautions for Correct Use

- Do not attempt to change an axis that is set to *Unused axis (unchangeable to used axis)* to a used axis.
- You cannot set an axis in an axes group if the Axis Use axis parameter is set to *Unused axis (unchangeable to used axis)*.

For details, refer to the `MC_ChangeAxisUse` instruction in the *NJ/NX-series Motion Control Instructions Reference Manual* (Cat. No. W508).

For an application example of the `MC_ChangeAxisUse` instruction, refer to the *NJ/NX-series CPU Unit Software User's Manual* (Cat. No. W501).

6-8-11 Enabling Digital Cam Switch

You can use the MC_DigitalCamSwitch (Enable Digital Cam Switch) instruction to turn the digital outputs ON or OFF according to the axis position.

The setting of the *ValueSource* input variable to the instruction also allows you to adjust for the acceleration or deceleration rate.

Always use this function together with the NX_AryDOutTimeStamp instruction and with a Digital Output Unit that supports time stamp refreshing. The NX_AryDOutTimeStamp instruction turns the specified digital outputs ON or OFF at specified timing of the time stamp.



Precautions for Correct Use

You can use this instruction for an axis that is assigned to an NX-series Position Interface Unit.

The NX Units that can be used are NX-EC0□□□ and NX-ECS□□□, also must be running the time stamping.

Refer to the MC_DigitalCamSwitch (Enable Digital Cam Switch) instruction in the *NJ/NX-series Motion Control Instructions Reference Manual* (Cat. No. W508) for details on enabling digital cam switch.

Refer to the *NJ/NX-series Instructions Reference Manual* (Cat. No. W502) for details on NX_AryDOutTimeStamp instruction.

Refer to the *NX-series Digital I/O Units User's Manual* (Cat. No. W521-E1-02 or later) for Digital Output Unit that supports time stamp refreshing.

Refer to the *NX-series Position Interface Units User's Manual* (Cat. No. W524-E1-02 or later) for time stamping and time stamps.

6-8-12 Displaying 3D Motion Monitor for User Coordinate System

In the case that coordinate systems (such as SCARA robot and vertical articulated robot) other than orthogonal coordinate system are implemented by user programs, this function can be used to display the path of robot hands, etc. in 3D with Sysmac Studio.

You can create an `_sMC_POSITION_REF` type user-defined variable and display in 3D Motion Monitor Display Mode.

● `_sMC_POSITION_REF`

The followings are the members of `_sMC_POSITION_REF` type data.

Member	Data type	Meaning
CommandPosition	ARRAY [0..5] OF LREAL	Command Current Position
ActualPosition	ARRAY [0..5] OF LREAL	Actual Current Position

The following list describes each member.

Member	Description
User-defined variable.CommandPosition[0]	This is an X-axis component for the command current position. This member is assigned a user-defined variable that indicates the X-axis position of the command current position generated by a user program.
User-defined variable.CommandPosition[1]	This is a Y-axis component for the command current position. This member is assigned a user-defined variable that indicates the Y-axis position of the command current position generated by a user program.
User-defined variable.CommandPosition[2]	This is a Z-axis component for the command current position. This member is assigned a user-defined variable that indicates the Z-axis position of the command current position generated by a user program.
User-defined variable.CommandPosition[3] to [5]	Not used.
User-defined variable.ActualPosition[0]	This is an X-axis component for the actual current position. This member is assigned a user-defined variable that indicates the X-axis position of the actual current position handled in a user program.
User-defined variable.ActualPosition[1]	This is a Y-axis component for the actual current position. This member is assigned a user-defined variable that indicates the Y-axis position of the actual current position handled in a user program.
User-defined variable.ActualPosition[2]	This is a Z-axis component for the actual current position. This member is assigned a user-defined variable that indicates the Z-axis position of the actual current position handled in a user program.
User-defined variable.ActualPosition[3] to [5]	Not used.

Each member is assigned a user-defined variable. The followings are the examples.

Name	Data type	Description
3D_position	_sMC_POSITION_REF	User-defined variable for 3D display
MCS_Cmd_TransX	LREAL	User-defined variable that indicates the X-axis position of the command current position generated by a user program
MCS_Cmd_TransY	LREAL	User-defined variable that indicates the Y-axis position of the command current position generated by a user program
MCS_Cmd_TransZ	LREAL	User-defined variable that indicates the Z-axis position of the command current position generated by a user program
MCS_Act_TransX	LREAL	User-defined variable that indicates the X-axis position of the actual current position handled in a user program
MCS_Act_TransY	LREAL	User-defined variable that indicates the Y-axis position of the actual current position handled in a user program
MCS_Act_TransZ	LREAL	User-defined variable that indicates the Z-axis position of the actual current position handled in a user program

```
3D_position.CommandPosition[0] := MCS_Cmd_TransX;
```

```
3D_position.CommandPosition[1] := MCS_Cmd_TransY;
```

```
3D_position.CommandPosition[2] := MCS_Cmd_TransZ;
```

```
3D_position.ActualPosition[0] := MCS_Act_TransX;
```

```
3D_position.ActualPosition[1] := MCS_Act_TransY;
```

```
3D_position.ActualPosition[2] := MCS_Act_TransZ;
```

● Overview of Operating Procedures

- 1 Create an _sMC_POSITION_REF type user-defined variable.
- 2 Create a program in which user-defined variables that indicate the command current position and actual current position for 3D display are assigned to each member of the created user-defined variable.
- 3 Select *Specified coordinate* in the *Type* Box in the 3D Machine Model List.
The _sMC_POSITION_REF data type is displayed in the 3D Machine Model Parameter Settings section.
- 4 Set the created user-defined variable in the *Value* Column in the 3D Machine Model Parameter Settings section.
- 5 Execute the user program.
- 6 Start tracing the data with the data trace to sample the data.
- 7 Check the trace results on the Data Trace Tab Page.

Refer to the *Sysmac Studio Version 1 Operation Manual* (Cat. No. W504) for details on 3D Motion Monitor Display Mode.

7

Introduction of EtherNet/IP Communications Functions

This section describes the communications services of the built-in EtherNet/IP port for an NX1P2 CPU Unit.

7-1	Communications Services	7-2
7-1-1	CIP (Common Industrial Protocol) Communications Services	7-2
7-1-2	BOOTP Client	7-4
7-1-3	FTP Server	7-4
7-1-4	FTP Client	7-5
7-1-5	Automatic Clock Adjustment	7-5
7-1-6	Socket Service	7-6
7-1-7	Specifying Host Names	7-7
7-1-8	SNMP Agent	7-7

7-1 Communications Services

The following describes the communications services of the built-in EtherNet/IP port for an NX1P2 CPU Unit.

For details on this function, refer to the *NJ/NX-series CPU Unit Built-in EtherNet/IP Port User's Manual* (Cat. No. W506)

7-1-1 CIP (Common Industrial Protocol) Communications Services

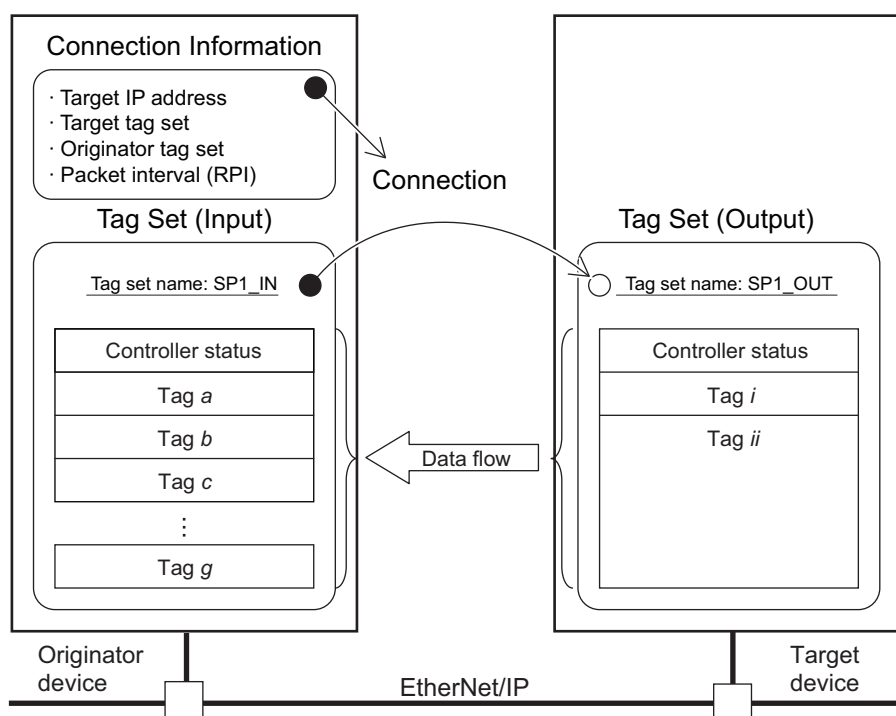
Tag Data Links (Cyclic Communications)

A program is not required to perform cyclic data exchanges with other devices on the EtherNet/IP network.

Normally, a connection is started with the target device for each tag set that was created with the Network Configurator to start communications for tag data links for a built-in EtherNet/IP port. One connection is used per tag set.

The maximum number of connections that can be registered for the NX1P2 CPU Unit is 32.

Refer to the *NJ/NX-series CPU Unit Built-in EtherNet/IP Port User's Manual* (Cat. No. W506) for the built-in EtherNet/IP port tag and tag set specifications.

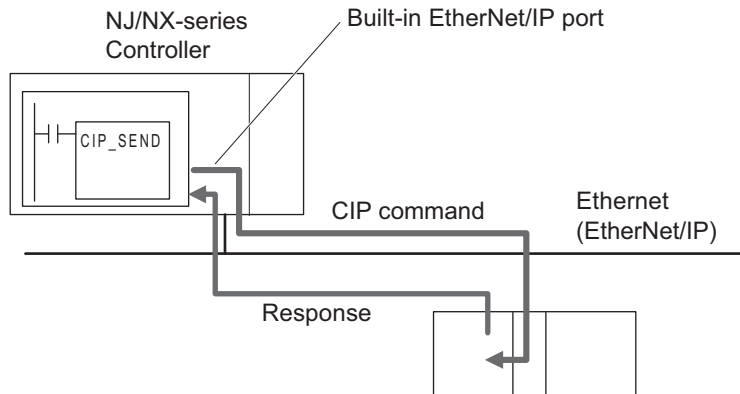


Note In this example, a connection is established with the originator's tag list with tags a to g (inputs), which are in a tag set called SP1_IN, and the target's tag list with tags i and ii (outputs), which are in a tag set called SP1_OUT.

CIP Message Communications

User-specified CIP commands can be sent to devices on the EtherNet/IP network.

CIP commands, such as those for reading and writing data, can be sent and their responses received by executing the CIP communications instructions from the user program in the NJ/NX-series CPU Unit.



By specifying a route path, you can send CIP messages (CIP commands and responses) to a device on another CIP-based network segment via a built-in EtherNet/IP port or the EtherNet/IP Unit (CIP routing function for message communications).

The maximum number of levels of CIP routing via the ports is eight for any combination of CS, CJ, NJ, and NX-series CPU Units. Note that the number of levels of IP routing using an L3 Ethernet switch is not counted in the number of levels of CIP routing via the ports.



Additional Information

In CIP routing, a node (Unit) that routes information subtracts the equivalent of one hop from the timeout, deletes its own address from the route information, and relays the information to the next node (Unit).

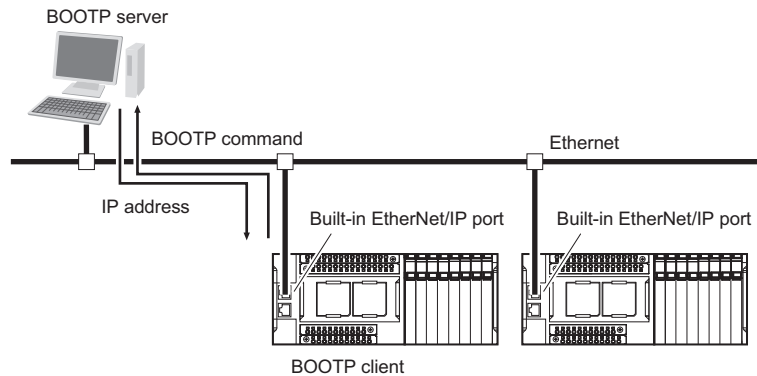
When a timeout is specified, the timeout for the actual request service processing is set in the last hop.

In the case of relay hops, the timeout for the relay route must be added to the timeout for the request.

OMRON products that support CIP subtract 5 seconds per hop.

7-1-2 BOOTP Client

You set the built-in EtherNet/IP port in the BOOTP settings to use the BOOTP client to obtain settings, such as the built-in EtherNet/IP port IP address.

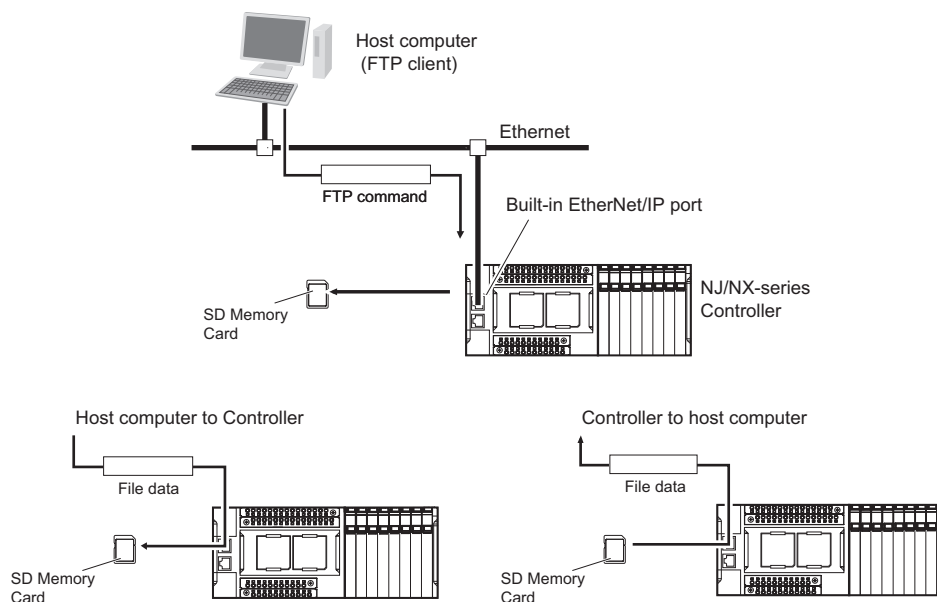


The built-in EtherNet/IP port IP address is obtained from the BOOTP server when the power is turned ON.

7-1-3 FTP Server

An FTP server is built into the built-in EtherNet/IP port so that files can be read from and written to the SD Memory Card in the CPU Unit of the Controller from computers at other Ethernet nodes.

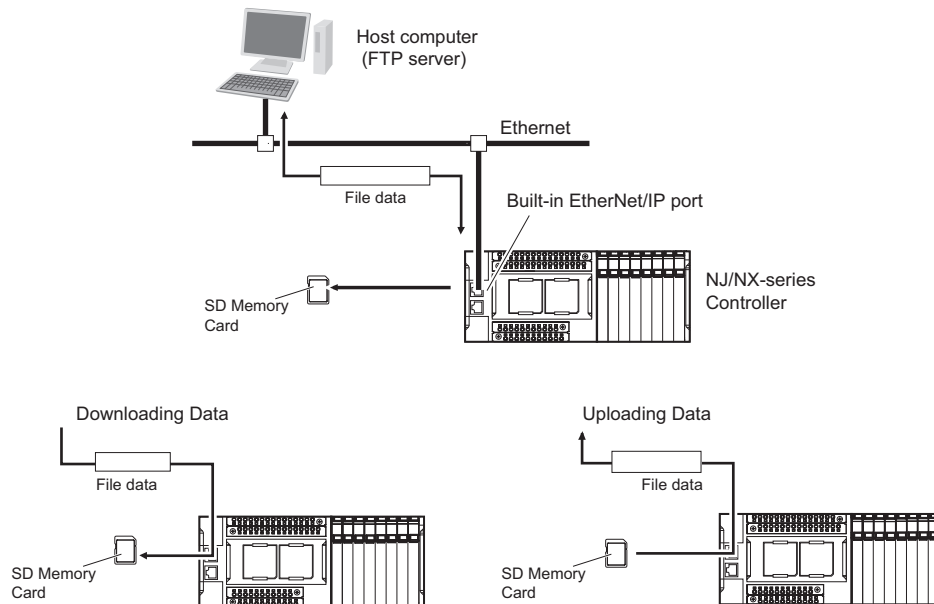
This makes it possible to exchange data files between a host computer and the Controller with the host computer as the FTP client and the Controller as the FTP server.



7-1-4 FTP Client

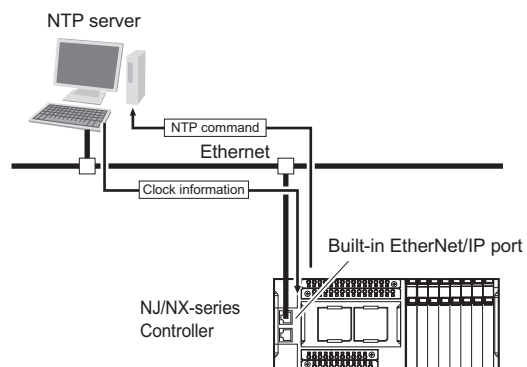
The built-in EtherNet/IP port contains an FTP client. With it, you can use FTP client communications instructions to transfer files between the CPU Unit and host computers on Ethernet.

This makes it possible to exchange data files between a host computer and the Controller with the Controller as the FTP client and the host computer as the FTP server.



7-1-5 Automatic Clock Adjustment

With the built-in EtherNet/IP port, clock information is read from the NTP server at the specified time or at a specified interval after the power supply to the CPU Unit is turned ON. The internal clock time in the CPU Unit is updated with the read time.



Precautions for Correct Use

An NTP server is required to use automatic clock adjustment.

7-1-6 Socket Service

You can send data to and receive data from any node on Ethernet with the UDP or TCP protocol.

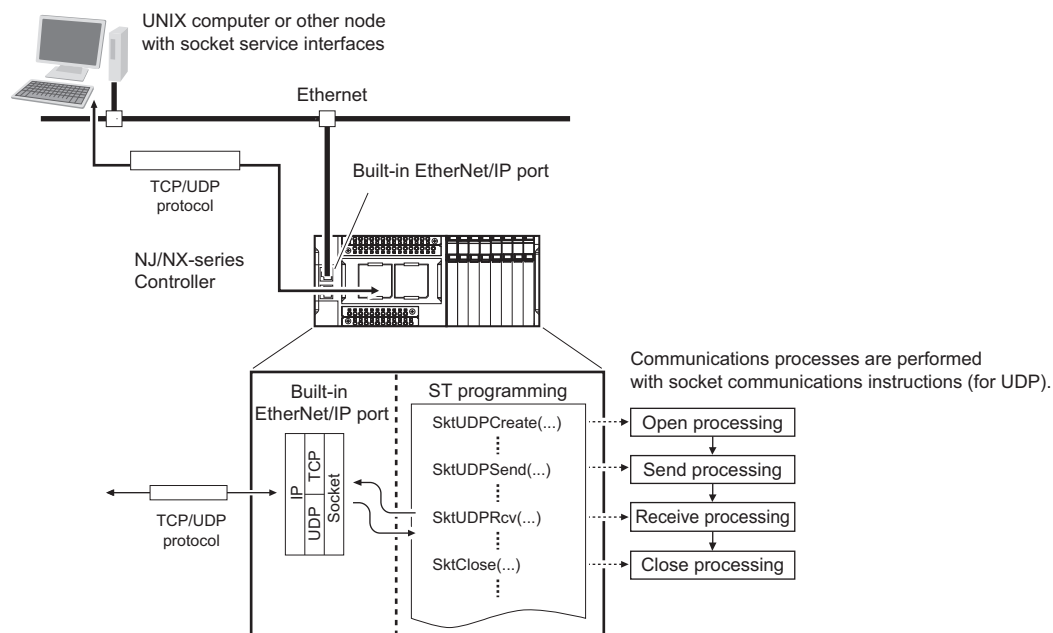
To send/receive data with a socket service, you execute multiple socket communications instructions in sequence in an ST program to execute the required communications processes.

After a connection with the other communications device is opened with an open instruction, the values of the variables that are specified for the send instruction are sent and the data that was received for a receive instruction is stored in the specified variables.

The connection is closed with a close instruction, and communications end.

For TCP, you can also read the socket status and received data.

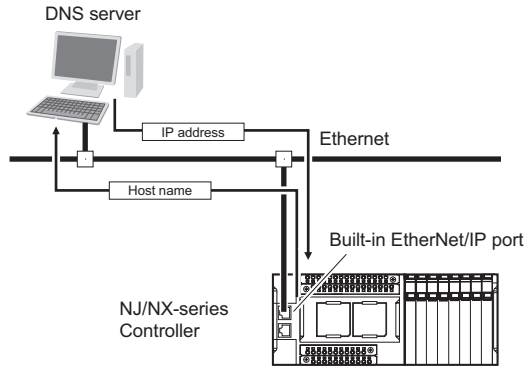
You can use a total of 30 TCP ports and UDP ports.



7-1-7 Specifying Host Names

You can directly specify IP addresses, but you can also use the host names instead of the IP addresses for NTP servers, SNMP managers, or the destinations of socket instructions and CIP communications instructions (DNS client or hosts settings).

Example: Setting the Host Name on a DNS Server

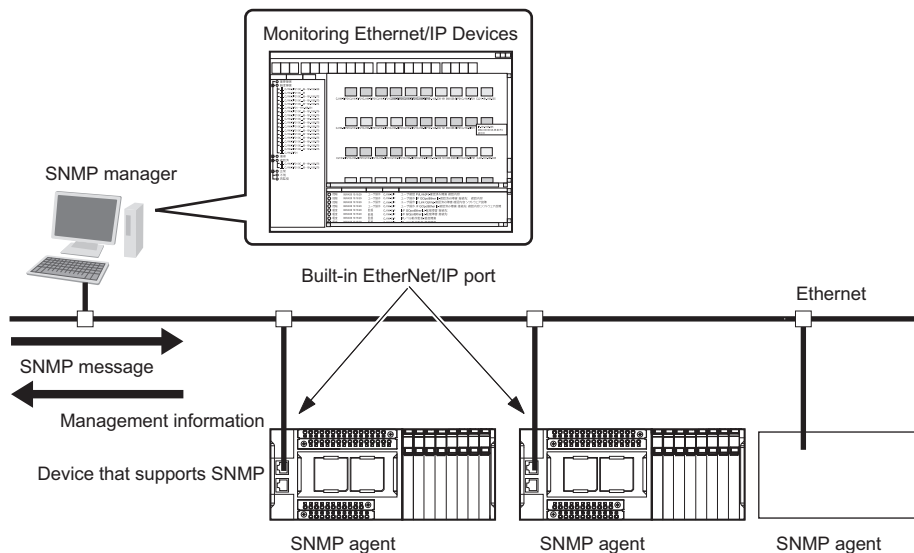


Precautions for Correct Use

A DNS server is required to use the server host names for the DNS client.

7-1-8 SNMP Agent

The SNMP agent passes internal status information from the built-in EtherNet/IP port to network management software that uses an SNMP manager.



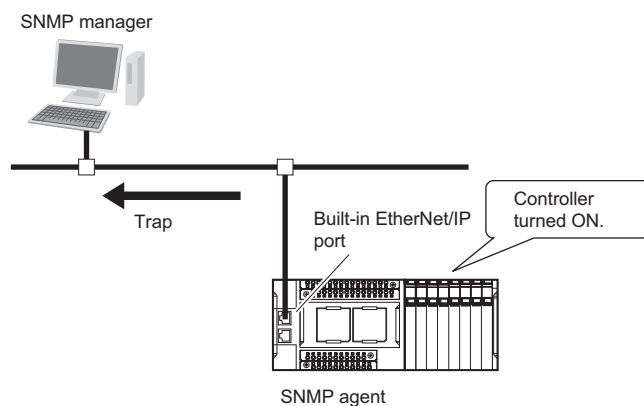
- SNMP Trap

When specific conditions occur, the built-in EtherNet/IP port that is set as the SNMP agent sends status notification reports to the SNMP manager.

The SNMP manager can learn about changes in status even without periodically monitoring of the built-in EtherNet/IP port.

Status notification reports are sent under the following conditions.

- a) When the Controller is turned ON
- b) When links are established
- c) When an SNMP agent fails to be authorized



8

Introduction of EtherCAT Communications Functions

This section describes the communications functions of the built-in EtherCAT port for an NX1P2 CPU Unit.

8-1	Overview of Communications	8-2
8-1-1	Process Data Communications and SDO Communications	8-2
8-1-2	Other Functions	8-3

8-1 Overview of Communications

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

For details on the communications function of the built-in EtherCAT port, refer to the *NJ/NX-series CPU Unit Built-in EtherCAT Port User's Manual* (Cat. No. W505).

8-1-1 Process Data Communications and SDO Communications

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

- Process data communications
- SDO communications

Process Data Communications

“Process data communications” is a cyclic communications method in which control information is exchanged in a fixed cycle between the EtherCAT master and slaves.

The fixed cycle is called a process data communications cycle. The EtherCAT master can exchange information with EtherCAT slaves in realtime in this process data communications cycle.

The same control period is also used for the process data communications cycle for EtherCAT. This enables precise sequence and motion control in a fixed period with very little deviation.

Refer to the *NJ/NX-series CPU Unit Built-in EtherCAT Port User's Manual* (Cat. No. W505) for details on the specifications of process data communications.

SDO Communications

“SDO communications” is a communications method in which control information is exchanged in non-cyclic event communications between the EtherCAT master and slaves.

You can use EtherCAT communications instructions to read and write the SDO data in EtherCAT slaves.

Refer to the *NJ/NX-series CPU Unit Built-in EtherCAT Port User's Manual* (Cat. No. W505) for details on the specifications of SDO communications.

8-1-2 Other Functions

In addition to process data communications and SDO communications, the built-in EtherCAT port also provides functions related to EtherCAT network configurations and setup, as well as communications control and maintenance during operation or errors.

Network Configurations and Setup

● Enable/Disable Setting for Slaves

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

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

You enable/disable each slave in the Sysmac Studio. Refer to the *NJ/NX-series CPU Unit Built-in EtherCAT Port User's Manual* (Cat. No. W505) for the setting procedure.

To enable/disable slaves during system operation, use the Enable/Disable EtherCAT Slave instruction. Refer to the *NJ/NX-series Instructions Reference Manual* (Cat. No. W502) for details on this instruction.

Communications Control during Operation

● Wait Time Setting for Slave Startup

Use this function to set the wait time until all of the EtherCAT slaves are connected to the network.

When you use a slave that takes time to start, use a longer wait time setting to prevent errors.

You set the wait time for slave startup in the Sysmac Studio. Refer to the *NJ/NX-series CPU Unit Built-in EtherCAT Port User's Manual* (Cat. No. W505) for the setting procedure.

Communications Control at Error Occurrence

● Fail-soft Operation

Use this function to continue or stop the communications with EtherCAT slaves that are operating normally, if a communications error occurs.

"Fail-soft operation" refers an operation that only normally operating EtherCAT slaves are allowed to operate continuously.

The EtherCAT master can continue the communications with the EtherCAT slaves until the operation is stopped safely through the user program or user operation.

You enable/disable the fail-soft operation in Sysmac Studio. Refer to the *NJ/NX-series CPU Unit Built-in EtherCAT Port User's Manual* (Cat. No. W505) for the setting procedure.

Maintenance

● Disconnecting/Reconnecting Slaves

Use this function to temporarily stop and start communications with a specified slave.

It is useful because, during system operation, you can replace an erroneous EtherCAT slave without interrupting the communications with EtherCAT slaves that are operating normally.

Refer to the *NJ/NX-series CPU Unit Built-in EtherCAT Port User's Manual* (Cat. No. W505) for details on how to use this function.

● Diagnosis/Statistics Log

The diagnostic and statistical information provides statistics on the number of communications frames sent and received by the EtherCAT master and EtherCAT slaves as well as the number of frames for which errors were detected.

This function acquires the diagnostic and statistical information at the specified cycle and saves the information as a log file in an SD Memory Card that is mounted on the CPU Unit.

You can use it to diagnose the EtherCAT network line quality based on the diagnostic and statistical information.

Use this function for the following applications.

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

Refer to the *NJ/NX-series CPU Unit Built-in EtherCAT Port User's Manual* (Cat. No. W505) for details on how to use this function.

9

Troubleshooting

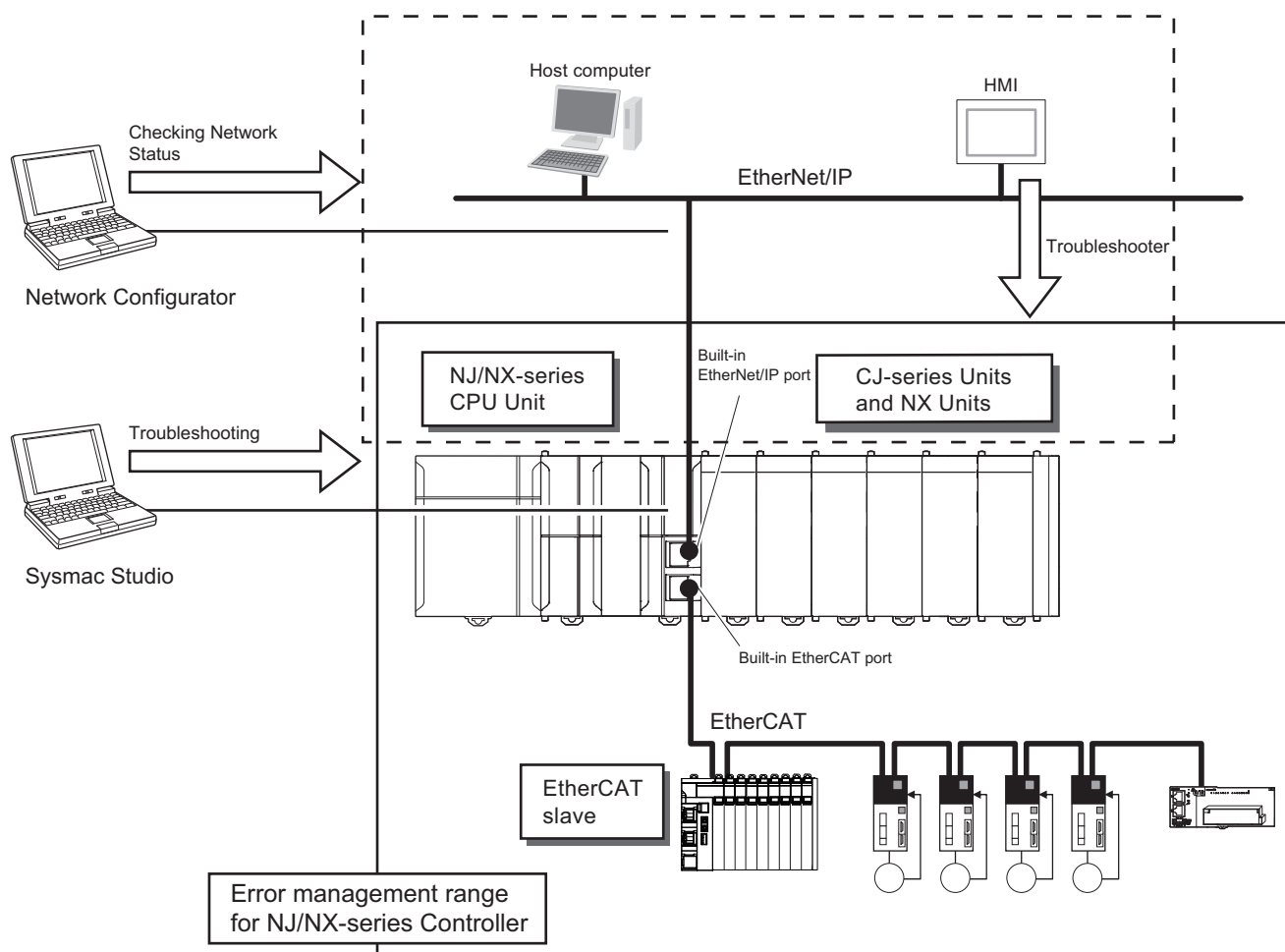
This section describes the overview of methods for checking errors.

9-1	Overview of Troubleshooting	9-2
9-2	Option Board Errors	9-3
9-2-1	Checking for Errors and Troubleshooting with the ERR Indicator on Option Boards	9-3

9-1 Overview of Troubleshooting

You manage all of the errors that occur on the NJ/NX-series Controller as events. This allows you to see what errors have occurred and find corrections for them with the same methods for the entire range of errors that is managed (i.e., CPU Unit, NX Units, NX-series Slave Terminals, EtherCAT slaves,^{*1} and CJ-series Units).

^{*1} Only Sysmac devices are supported.



You can use the troubleshooting functions of the Sysmac Studio or the Troubleshooter on an HMI to quickly check for errors that have occurred and find corrections for them.

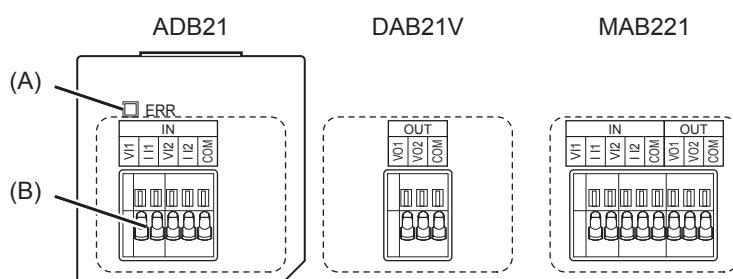
Refer to the *NJ/NX-series Troubleshooting Manual* (Cat. No. W503-E1-19 or later) for types of errors, meanings, specific corrections when errors occur and for troubleshooting information on the entire NJ/NX-series Controller.

9-2 Option Board Errors

The description below is related to Option Board errors.

9-2-1 Checking for Errors and Troubleshooting with the ERR Indicator on Option Boards

You can check the status indicator on an Analog I/O Option Board see if an error occurs in it. This table below gives the meanings of errors that the indicator shows and the troubleshooting information on them.



ERR indicator	Cause	Correction
Lit	Hardware failure	If this error occurs again even after you cycle the Controller power supply, replace the Option Board.
	Option Board Error	Refer to the event <i>Option Board Error</i> in the <i>NJ/NX-series Troubleshooting Manual</i> (Cat. No. W503-E1-19 or later).
	Analog Option Board Startup Error	Refer to the event <i>Analog Option Board Startup Error</i> in the <i>NJ/NX-series Troubleshooting Manual</i> (Cat. No. W503-E1-19 or later).
	Analog Option Board Communications Error	Refer to the event <i>Analog Option Board Communications Error</i> in the <i>NJ/NX-series Troubleshooting Manual</i> (Cat. No. W503-E1-19 or later).
Flashing	Option Board Error	Refer to the event <i>Option Board Error</i> in the <i>NJ/NX-series Troubleshooting Manual</i> (Cat. No. W503-E1-19 or later).
	Analog Option Board Startup Error	Refer to the event <i>Analog Option Board Startup Error</i> in the <i>NJ/NX-series Troubleshooting Manual</i> (Cat. No. W503-E1-19 or later).
	Analog Option Board Communications Error	Refer to the event <i>Analog Option Board Communications Error</i> in the <i>NJ/NX-series Troubleshooting Manual</i> (Cat. No. W503-E1-19 or later).
	Option Board Configuration Verification Error	Refer to the event <i>Option Board Configuration Verification Error</i> in the <i>NJ/NX-series Troubleshooting Manual</i> (Cat. No. W503-E1-19 or later).
Not lit	---	Normal operation

No error indicator is provided on Serial Communications Option Boards.



Appendices

A-1	Version Information	A-2
-----	---------------------------	-----

A-1 Version Information

This section describes the relationship between the unit versions of the NX1P2 CPU Units and the Sysmac Studio versions, and the functions that are supported for each unit version.

The following describes how the unit version of an NX1P2 CPU Unit corresponds to the Sysmac Studio version. Normally use the corresponding versions.

Unit Version and Corresponding Sysmac Studio Version

The following table gives the relationship between the unit version of an NX1P2 CPU Unit and the corresponding Sysmac Studio version.

Unit version of CPU Unit	Corresponding version of Sysmac Studio
Ver. 1.16 ^{*1}	Ver. 1.20
Ver. 1.14	Ver. 1.18
Ver. 1.13 ^{*2}	Ver. 1.17 ^{*3}

*1. There is no NX1P2-□□□□□□ CPU Unit with unit version 1.15.

*2. There is no NX1P2-□□□□□□ CPU Unit with unit version 1.12 or earlier.

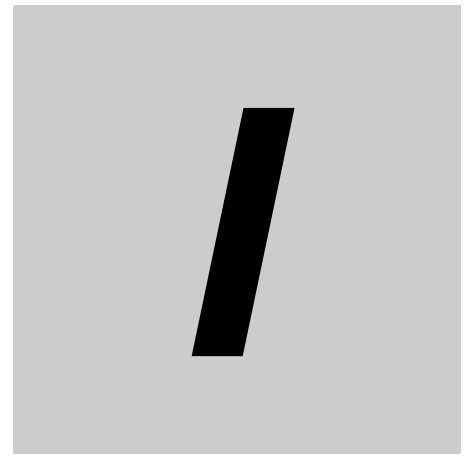
*3. Use an NX1P2-□□□□□□ CPU Unit with Sysmac Studio version 1.17 or higher. You cannot use an NX1P2-□□□□□□ CPU Unit with Sysmac Studio version 1.16 or lower.

Restriction When the Unit Version Does not Correspond to the Sysmac Studio Version

By specification, the following restriction applies when the unit version of the NX1P2 CPU Unit does not correspond to the Sysmac Studio version.

● When the Sysmac Studio Version Is 1.16 or Lower

You cannot use the NX1P2 CPU Unit with Sysmac Studio version 1.16 or lower.



Index

Index

Symbols

.Run (Option Board
Normal Operation)3-10, 4-23, 4-31, 5-9

A

aborting 6-49, 6-67
acceleration and deceleration rates
 unit 6-36
acceleration rate
 changing 6-45
Acceleration Warning Value 6-36
Acceleration/Deceleration Over 6-36
actual position 6-33
actual velocity 6-35
Actual Velocity Filter Time Constant 6-35
allowable value conversion range 5-3, 5-4
AT specification 4-8, 4-17
automatic clock adjustment 7-5
axes group errors
 resetting 6-56
axes groups
 enabling and disabling 6-55

B

blending 6-50, 6-68
BOOTP client 7-4
Buffer Mode 6-48, 6-66
buffered 6-49, 6-68

C

cam block 6-17
cam block end point 6-17
cam block start point 6-17
cam curve 6-17
cam data 6-17
 loading and saving 6-21
cam data index 6-17
cam data variable 6-17
cam end point 6-17
cam operation 6-17
cam profile curve 6-17
cam start point 6-17
Cam table
 Generate Cam Table 6-23
cam table 6-17
cam table start position 6-17

cam tables 6-19
 data type 6-20
 saving 6-21
 specifications 6-19
 switching 6-21
 updating properties 6-22
CIP communications 7-2
CIP message communications 7-3
command position 6-33
command velocity 6-35
communications setting level 4-16
Configuration 3-4
connecting acceleration 6-18
connecting velocity 6-18
converted value 5-3
current direction 6-39

D

deceleration rate
 changing 6-45
deceleration stop of command value 6-11
Deceleration Warning Value 6-37
delay time 2-12
diagnosis/statistics log 8-4
disconnecting/connecting slaves 8-4
displacement 6-17
download area 4-11, 4-17

E

enable/disable setting for slaves 8-3
events 9-2

F

fail-soft operation 8-3
FTP client 7-5
FTP server 7-4

G

General-purpose Serial Communications Device Settings
 4-28

H

Host Link (FINS) 3-6, 4-2, 4-4, 4-10

I

I/O response time 2-15
immediate stop of command value 6-11
immediate stop of command value and error reset 6-11

immediate stop of command value and servo OFF6-11
 Input Filter Settings2-8
 input range5-3
 invalid cam data6-17

J

jerk unit6-38

L

Load Rejection Output Settings2-8

M

master axis6-17
 master following distance6-18
 Maximum Acceleration6-36
 Maximum Deceleration6-36
 Maximum Jog Velocity6-35
 maximum number of cam data6-17
 Maximum Velocity6-35
 Memory Settings for CJ-series Units 3-6, 3-8
 memory used for CJ-series Units 4-2, 4-8, 4-11, 4-17
 Modbus-RTU command 4-18, 4-22
 Modbus-RTU Master 3-13, 4-2, 4-18
 multi-execution of instructions 6-48, 6-66

N

negative direction6-39
 no direction specified6-39
 No-Protocol 3-13, 4-3, 4-25
 null cam data6-18
 number of valid cam data6-17

O

OFF filter only2-11
 ON and OFF filters2-10
 Option Board Normal Operation 3-10, 4-23, 4-31, 5-9
 Option Board Serial Communications Settings 3-5, 3-6
 option board service5-12
 option board slot3-2
 Option Board specification 4-23, 4-30
 Option Board Status 3-9, 4-23, 4-31, 5-9
 original cam data6-17
 output range5-4
 overrides6-61

P

phase6-17
 phase pitch6-18
 _PLC_OptBoardSta (Option Board Status)
 3-9, 3-10, 4-23, 4-31, 5-9
 positions6-33
 types6-33
 positive direction6-39

process data communications8-2
 programless communications 4-4, 4-8, 4-10
 program-modified cam data6-17

R

reading axes group positions6-56
 re-executing instructions6-43
 re-execution of instructions6-65
 resetting axis errors6-3

S

S-curve6-38
 SDO communications8-2
 serial communications instruction 3-13, 4-22, 4-29
 Serial communications mode3-6, 4-2
 settings on Modbus-RTU slaves4-21
 settings on NB-series Units4-7
 settings on the NX1W-CIF11/CIF12 Option Board
 4-7, 4-15, 4-21, 4-28
 shortest way6-39
 slave axis6-17
 SNMP agent7-7
 socket service7-6
 specifying host name7-7
 start mode6-18
 Start Velocity6-35
 status indicator5-2
 stop priorities6-12
 stopping
 due to errors or other problems6-8, 6-59
 immediate stop input6-7
 limit inputs6-7
 MC_GroupImmediateStop instruction6-59
 MC_GroupStop instruction6-59
 MC_ImmediateStop instruction6-8
 MC_Stop instruction6-8
 Servo Drive input signals6-7
 stop method6-11
 superimpose corners6-70, 6-72

T

tag data link (cyclic communications)7-2
 target position
 changing6-43
 excessive deceleration patterns6-44
 triangular control patterns6-44
 when a reverse turn does not occur for the
 new command value6-43
 when a reverse turn occurs for the
 new command value6-43
 target velocity
 changing6-45
 torque command
 changing6-46
 transition disabled6-70
 Transition Modes6-70

travel distance
 changing 6-45

U

upload area 4-11

V

valid cam data 6-17

velocities

 types 6-35

 velocity unit 6-35

Velocity Warning Value 6-35

W

wait time setting for slave startup 8-3

OMRON Corporation Industrial Automation Company
Kyoto, JAPAN

Contact: www.ia.omron.com

Regional Headquarters

OMRON EUROPE B.V.

Wegalaan 67-69, 2132 JD Hoofddorp
The Netherlands
Tel: (31)2356-81-300/Fax: (31)2356-81-388

OMRON ASIA PACIFIC PTE. LTD.

No. 438A Alexandra Road # 05-05/08 (Lobby 2),
Alexandra Technopark,
Singapore 119967
Tel: (65) 6835-3011/Fax: (65) 6835-2711

OMRON ELECTRONICS LLC

2895 Greenspoint Parkway, Suite 200
Hoffman Estates, IL 60169 U.S.A.
Tel: (1) 847-843-7900/Fax: (1) 847-843-7787

OMRON (CHINA) CO., LTD.

Room 2211, Bank of China Tower,
200 Yin Cheng Zhong Road,
PuDong New Area, Shanghai, 200120, China
Tel: (86) 21-5037-2222/Fax: (86) 21-5037-2200

Authorized Distributor:

© OMRON Corporation 2016-2019 All Rights Reserved.
In the interest of product improvement,
specifications are subject to change without notice.

Cat. No. W579-E1-04

0119