



# Machine Automation Controller

## NJ-series

# NJ Robotics CPU Unit

## User's Manual

**NJ501-4500**

**NJ501-4400**

**NJ501-4320**

**NJ501-4310**

**NJ501-4300**

**CPU Unit**




**W539-E1-05**

**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 USA 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 NJ-series NJ Robotics CPU Unit (hereinafter called NJ Robotics CPU Unit).

This manual contains information that is necessary to use the NJ Robotics CPU Unit. Please read this manual and make sure you understand the functionality and performance of this product before you attempt to use it.

When you have finished reading this manual, keep it in a safe location where it will be readily available for reference during operation.

This manual describes only the functions added to NJ501-4□□□ Units.

Refer to the NJ-series manuals listed in *Related Manuals* on page 33 for functions which are common in NJ501-□□□□ Units including NJ501-1□□□ Units.

## 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 connecting 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.

- NJ-series NJ Robotics CPU Unit
  - NJ501-4500
  - NJ501-4400
  - NJ501-4320
  - NJ501-4310
  - NJ501-4300

(Robot Version 1.02 or later)

# Relevant Manuals

The following table provides the relevant manuals for this product. Read all of the manuals that are relevant to your system configuration and application to make the most of this product.

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			NJ/NX-series CPU Unit Motion Control User's Manual	NJ/NX-series Motion Control Instructions Reference Manual	NJ/NX-series CPU Unit Built-in EtherCAT® Port User's Manual	NJ/NX-series CPU Unit Built-in EtherNet/IP™ Port User's Manual	NJ-series Database Connection CPU Units User's Manual	NJ-series NJ Robotics CPU Unit User's Manual	NJ/NX-series Troubleshooting Manual
	NJ-series CPU Unit Hardware User's Manual	NJ/NX-series CPU Unit Software User's Manual	NJ/NX-series Instructions Reference Manual							
Introduction to NJ-series Controllers	●									
Setting devices and hardware										
Using motion control				●						
Using EtherCAT	●					●				
Using EtherNet/IP							●			
Using database connection service								●		
Using robot control									●	
Software settings										
Using motion control				●						
Using EtherCAT		●				●				
Using EtherNet/IP							●			
Using database connection service								●		
Using robot control									●	
Writing the user program										
Using motion control				●	●					
Using EtherCAT						●				
Using EtherNet/IP		●	●				●			
Using database connection service								●		
Using robot control									●	
Programming error processing										●
Testing operation and debugging										
Using motion control				●						
Using EtherCAT		●				●				
Using EtherNet/IP							●			
Using database connection service								●		
Using robot control									●	

Purpose of use	Manual									
	Basic information			N/J/NX-series CPU Unit Motion Control User's Manual	N/J/NX-series Motion Control Instructions Reference Manual	N/J/NX-series CPU Unit Built-in EtherCAT® Port User's Manual	N/J/NX-series CPU Unit Built-in EtherNet/IP™ Port User's Manual	N/J-series Database Connection CPU Units User's Manual	N/J-series NJ Robotics CPU Unit User's Manual	N/J/NX-series Troubleshooting Manual
	N/J/NX-series Instructions Reference Manual	N/J/NX-series CPU Unit Software User's Manual	N/J-series CPU Unit Hardware User's Manual							
Learning about error management and corrections *1	△	△	△			△	△	△	△	●
Maintenance	●									
Using motion control				●						
Using EtherCAT						●				
Using EtherNet/IP							●			
Using database connection service								●		
Using robot control								●		

\*1. Refer to the *NJ/NX-series Troubleshooting Manual* (Cat. No. W503) for the error management concepts and an overview of the error items. For details on each error, refer to the relevant manuals indicated with △ mark.

# Manual Structure

## Page Structure and Symbols

The following page structure and symbols are used in this user's manual.

The diagram illustrates the structure of a manual page with the following annotations:

- Level 1 heading:** Points to the top header "4 Installation and Wiring".
- Level 2 heading:** Points to the section heading "4-3 Mounting Units".
- Level 3 heading:** Points to the sub-section heading "4-3-1 Connecting Controller Components".
- Procedure step:** Points to the numbered step "1 Join the Units so that the connectors fit exactly.".
- Special information:** Points to the "Precautions for Correct Use" section, which includes a warning icon and text: "The sliders on the tops and bottoms of the Power Supply Unit, CPU Unit, I/O Units, Special I/O Units, and CPU Bus Units must be completely locked (until they click into place) after connecting the adjacent Unit connectors." Below this text are four icons: a warning triangle, a document, a list, and a checkmark.
- Manual name:** Points to the footer text "NJ-series CPU Unit Hardware User's Manual (W500)".
- Page tab:** Points to the page number "4" in the right margin.

Additional page elements shown include:

- Page number "4-9" in the bottom right corner.
- Vertical page numbers "4-3 Mounting Units" and "4-3-1 Connecting Controller Components" on the right side.

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

## Special Information

Special information in this user's 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, Position Interface Units, and the Sysmac Studio with different versions.

## Precaution on Terminology

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

# Sections in this Manual

---

<b>1</b>	<b>Overview of NJ Robotics CPU Unit</b>	<b>1</b>
<b>2</b>	<b>Features of NJ Robotics CPU Unit</b>	<b>2</b>
<b>3</b>	<b>Setting Robot Functions</b>	<b>3</b>
<b>4</b>	<b>Robot Instructions</b>	<b>4</b>
<b>5</b>	<b>Vision &amp; Robot Integrated Simulation</b>	<b>5</b>
<b>6</b>	<b>Troubleshooting</b>	<b>6</b>
<b>A</b>	<b>Appendices</b>	<b>A</b>
<b>I</b>	<b>Index</b>	<b>I</b>

# CONTENTS

---

<b>Introduction .....</b>	<b>1</b>
Intended Audience.....	1
Applicable Products.....	1
<b>Relevant Manuals .....</b>	<b>2</b>
<b>Manual Structure .....</b>	<b>4</b>
Page Structure and Symbols .....	4
Special Information .....	5
Precaution on Terminology .....	6
<b>Sections in this Manual .....</b>	<b>7</b>
<b>CONTENTS.....</b>	<b>8</b>
<b>Terms and Conditions Agreement.....</b>	<b>12</b>
Warranty, Limitations of Liability .....	12
Application Considerations .....	13
Disclaimers .....	13
<b>Safety Precautions .....</b>	<b>14</b>
Definition of Precautionary Information.....	14
Symbols.....	14
Warnings.....	15
Cautions.....	17
<b>Precautions for Safe Use.....</b>	<b>19</b>
<b>Precautions for Correct Use.....</b>	<b>26</b>
<b>Regulations and Standards .....</b>	<b>28</b>
Using Product Outside Japan .....	28
Conformance to EC Directives .....	28
Conformance to Shipbuilding Standards .....	29
Using Controllers in Robot Systems.....	29
Software Licenses and Copyrights .....	29
<b>Versions .....</b>	<b>30</b>
Confirming Versions.....	30
<b>Related Manuals .....</b>	<b>33</b>
<b>Terminology .....</b>	<b>35</b>
<b>Revision History .....</b>	<b>37</b>

## Section 1 Overview of NJ Robotics CPU Unit

---

<b>1-1 Features and System Configuration of Unit .....</b>	<b>1-2</b>
1-1-1 Features of NJ Robotics CPU Unit.....	1-2
1-1-2 Introduction to the System Configurations .....	1-4
<b>1-2 Operation Procedure of Unit .....</b>	<b>1-7</b>
<b>1-3 Specifications of Unit.....</b>	<b>1-8</b>
1-3-1 Performance Specifications .....	1-8
1-3-2 Function Specifications .....	1-9

<b>1-4</b>	<b>Robot Control Configuration</b> .....	<b>1-11</b>
------------	--	-------------

## **Section 2 Features of NJ Robotics CPU Unit**

<b>2-1</b>	<b>Controllable Robot Types</b> .....	<b>2-2</b>
<b>2-2</b>	<b>Coordinate System</b> .....	<b>2-5</b>
2-2-1	Overview of Coordinate Systems .....	2-5
2-2-2	Coordinate Systems .....	2-19
<b>2-3</b>	<b>Robot Functions</b> .....	<b>2-20</b>
2-3-1	Kinematics Setting .....	2-20
2-3-2	Workspace Check.....	2-30
2-3-3	User Coordinate System (UCS).....	2-38
2-3-4	Monitoring .....	2-39
2-3-5	Time-specified Absolute Positioning.....	2-39
2-3-6	Conveyor Synchronization.....	2-40
2-3-7	Inverse Kinematics .....	2-41
2-3-8	Robot Tool.....	2-41
2-3-9	Robot Jogging.....	2-44
2-3-10	Checking Maximum Interpolation Velocity and Maximum Interpolation Acceleration/Deceleration.....	2-45
2-3-11	Multi-execution of Instructions with Buffered Mode .....	2-47
2-3-12	Multi-execution of Instructions with Blending Mode.....	2-47
2-3-13	Trajectory Types for Time-specified Motion .....	2-48
2-3-14	Joint Interpolated Point-to-Point Movement.....	2-50
2-3-15	Management of Robots with Multiple Arm Configurations.....	2-51

## **Section 3 Setting Robot Functions**

<b>3-1</b>	<b>Setting Group for Robot</b> .....	<b>3-2</b>
3-1-1	Group Setting Procedures .....	3-2
3-1-2	Axes Group Settings.....	3-3
3-1-3	Axis Settings.....	3-4
<b>3-2</b>	<b>Robot Kinematics Settings</b> .....	<b>3-9</b>
<b>3-3</b>	<b>Homing</b> .....	<b>3-10</b>
<b>3-4</b>	<b>Target Position and Direction Setting Method</b> .....	<b>3-11</b>
3-4-1	Position Settings.....	3-11
3-4-2	Wrist Control .....	3-12
<b>3-5</b>	<b>Checking Wiring from the Sysmac Studio</b> .....	<b>3-15</b>

## **Section 4 Robot Instructions**

<b>4-1</b>	<b>Overview of Robot Instructions</b> .....	<b>4-2</b>
4-1-1	List of Robot Instructions .....	4-2
4-1-2	Axes Group Instructions Enabled/disabled.....	4-2
4-1-3	Robot Execution and Stop Instructions.....	4-4
4-1-4	Unusable Output Variables of Axes Group .....	4-4
	<b>Details on Robot Instructions</b> .....	<b>4-5</b>
	MC_SetKinTransform.....	4-6
	MC_DefineCoordSystem.....	4-17
	MC_DefineToolTransform .....	4-21
	MC_GroupMon.....	4-25
	MC_MoveTimeAbsolute .....	4-31
	MC_MoveDirectAbsolute.....	4-41
	MC_SyncLinearConveyor .....	4-52
	MC_SyncOut.....	4-73
	MC_InverseKin.....	4-82

MC_RobotJog.....	4-87
MC_GroupSyncMoveAbsolute .....	4-96
<b>4-2 Multi-execution of Motion Control Instructions.....</b>	<b>4-98</b>
4-2-1 Multi-execution in Aborting Mode.....	4-98
4-2-2 Multi-execution in Buffered Mode.....	4-99
4-2-3 Multi-execution in Blending Mode .....	4-100
<b>4-3 State Transitions of Robot Instructions .....</b>	<b>4-105</b>
<b>4-4 Sample Programming .....</b>	<b>4-106</b>

## Section 5 Vision & Robot Integrated Simulation

---

<b>5-1 Overview of Simulation.....</b>	<b>5-2</b>
<b>5-2 Models that Support Simulation .....</b>	<b>5-3</b>
<b>5-3 Simulation Procedures .....</b>	<b>5-5</b>

## Section 6 Troubleshooting

---

<b>6-1 Error Table.....</b>	<b>6-2</b>
<b>6-2 Error Description .....</b>	<b>6-23</b>
6-2-1 Interpreting Error Description .....	6-23
6-2-2 Error Descriptions .....	6-24

## Appendices

---

<b>A-1 Sysmac Studio Robot Additional Option .....</b>	<b>A-2</b>
A-1-1 Enabling the Sysmac Studio Robot Additional Option .....	A-2
A-1-2 Disabling the Sysmac Studio Robot Additional Option.....	A-4
A-1-3 3D Equipment Model Creation Wizard.....	A-6
A-1-4 3D Machine Models .....	A-8
A-1-5 Calculation of Calibration Parameters.....	A-19
A-1-6 Display of Images in 3D Motion Monitoring.....	A-20
<b>A-2 Sysmac Studio Robot Options .....</b>	<b>A-21</b>
A-2-1 Enabling Sysmac Studio Robot Options .....	A-21
A-2-2 Disabling Sysmac Studio Robot Options .....	A-23
<b>A-3 Version Information .....</b>	<b>A-25</b>

## Index

---



# Terms and Conditions Agreement

## Warranty, Limitations of Liability

### Warranties

#### ● Exclusive Warranty

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

#### ● Limitations

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

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

#### ● Buyer Remedy

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

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

### Limitation on Liability; Etc

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

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

## Application Considerations

### Suitability of Use

---

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

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

### Programmable Products

---

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

## Disclaimers

### Performance Data

---

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

### Change in Specifications

---

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

### Errors and Omissions

---

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



# Safety Precautions

## Definition of Precautionary Information





The following notation is used in this manual to provide precautions required to ensure safe usage of an NJ Robotics Controller.

The safety precautions that are provided are extremely important to safety. Always read and heed the information provided in all safety precautions.

The following notation is used.

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

## Symbols

	<p>⊘ This symbol indicates operations that you must not do. The specific operation is shown in ⊘ and explained in text. This example indicates prohibiting disassembly.</p>
	<p>⚠ This symbol indicates precautions (including warnings). The specific operation is shown in ⚠ and explained in text. This example indicates a precaution for electric shock.</p>
	<p>⚠ This symbol indicates precautions (including warnings). The specific operation is shown in ⚠ and explained in text. This example indicates a general precaution.</p>
	<p>● This symbol indicates operations that you must do. The specific operation is shown in ● and explained in text. This example shows a general precaution for something that you must do.</p>

## Warnings

### **WARNING**

#### **During Power Supply**

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



Do not disassemble any of the Units.

Particularly the power-supplied Units contain parts with high voltages while power is supplied or immediately after power is turned OFF. Touching any of these parts may result in electric shock. There are sharp parts inside the Units that may cause injury.



#### **Fail-safe Measures**

Provide safety measures in external circuits to ensure safety in the system if an abnormality occurs due to malfunction of the CPU Unit, other Units, or slaves or due to other external factors affecting operation.



Not doing so may result in serious accidents due to incorrect operation.

Emergency stop circuits, interlock circuits, limit circuits, and similar safety measures must be provided in external control circuits.



The Controller outputs may remain ON or OFF due to deposition or burning of the output relays or destruction of the output transistors. As a countermeasure for such problems, external safety measures must be provided to ensure safe operation of the system.



The CPU Unit will turn OFF all outputs from Basic Output Units in the following cases.

- If an error occurs in the power supply
- If the power supply connection becomes faulty
- If a CPU watchdog timer error or CPU reset occurs
- If a major fault level Controller error occurs
- While the CPU Unit is on standby until RUN mode is entered after the power is turned ON.



Provide external safety measures so that the system operates safely if all outputs turn OFF when any of the above conditions occurs.

If external power supplies for slaves or other devices are overloaded or short-circuited, the voltage will drop, outputs will turn OFF, and the system may be unable to read inputs. Provide external safety measures in controls with monitoring of external power supply voltage as required so that the system operates safely in such a case.



Unintended outputs may occur when an error occurs in variable memory or in memory used for CJ-series Units. As a countermeasure for such problems, external safety measures must be provided to ensure safe operation of the system.



Provide measures in the communications system and user program to ensure safety in the overall system even if errors or malfunctions occur in data link communications or remote I/O communications.



## Safety Precautions

---

If there is interference in remote I/O communications or if a major fault level error occurs, output status will depend on the specifications of the product that is used.

Check the product's specifications and see what operation will occur when there is interference in communications or a major fault level error, and implement safety measures.

Set the slave settings correctly for all EtherCAT slaves.




---

The NJ-series Controller continues normal operation for a certain period of time even when a momentary power interruption occurs. This means that the NJ-series Controller may receive incorrect signals from external devices that are also affected by the power interruption.

Accordingly, take suitable actions, such as external fail-safe measures and interlock conditions, to monitor the power supply voltage of the external devices as required.




---

You must take fail-safe measures to ensure safety in the event of incorrect, missing, or abnormal signals caused by broken signal lines, momentary power interruptions, or other causes.



Not doing so may result in serious accidents due to incorrect operation.

---

## Voltage and Current Inputs

---

Make sure that the voltages and currents that are input to the Units and slaves are within the specified ranges.

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



## Downloading

---

Always confirm safety at the destination before you transfer a user program, configuration data, setup data, device variables, or values in memory used for CJ-series Units from the Sysmac Studio.

The devices or machines may perform unexpected operation regardless of the operating mode of the CPU Unit.



## Cautions



### Caution

#### Application

Do not touch any Unit while power is supplied or immediately after the power supply is turned OFF. Doing so may result in burn injury.



#### Wiring

Be sure that all terminal screws and cable connector screws are tightened to the torques specified in this manual or in the reference manuals. The loose screws may result in fire or malfunction.



#### Online Editing

Execute online editing only after confirming that no adverse effects will be caused by deviations in the timing of I/O. If you perform online editing, the task execution time may exceed the task period, I/O may not be refreshed with external devices, input signals may not be read, and output timing may change.



#### Error Message

- **Precaution on Error Message That Says an Instruction May Cause Unintended Operation**

Instructions may result in unexpected operation and affect the system if you clear the *Detect an error when an in-out variable is passed to specific instruction argument* Check Box in the Program Check Area under the Option settings in the Sysmac Studio.

Always confirm that the conditions for use that are given in the *NJ/NX-series Instructions Reference Manual* (Cat. No. W502) are met before you clear this check box.



- **Version Information**

This error message is displayed by and the above option setting is available on Sysmac Studio version 1.02.

## Simulation

---

Although the Sysmac Studio's simulation function simulates the operations of the Controller and vision sensors, there are differences from the Controller and vision sensors in operation and timing. After you use the simulation function to debug the user program, always check operation and perform adjustments on the physical Controller and vision sensors before you use the user program to operate the controlled system. Accidents may occur if the controlled system performs unexpected operation.

---



# Precautions for Safe Use

---

## Disassembly and Dropping

---

- Do not attempt to disassemble, repair, or modify any Units. Doing so may result in malfunction or fire.
- Do not drop any Unit or subject it to abnormal vibration or shock. Doing so may result in Unit malfunction or burning.

## Mounting

---

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

## Installation

---

Always connect to a ground of 100  $\Omega$  or less when installing the Units. To avoid electric shock, be sure to install a ground of 100  $\Omega$  or less especially when shorting the GR and LG terminals on the Power Supply Unit.

## Wiring

---

- Follow the instructions in the *NJ-series CPU Unit Hardware User's Manual* (Cat. No. W500) to correctly perform wiring.  
Double-check all wiring and switch settings before turning ON the power supply.
- Use crimp terminals for wiring.  
Do not connect bare stranded wires directly to terminals.
- Do not pull on the cables or bend the cables beyond their natural limit.  
Do not place heavy objects on top of the cables or other wiring lines. Doing so will damage the cable.
- Mount terminal blocks and connectors only after checking the mounting location carefully.
- Make sure that the terminal blocks, expansion cables, and other items with locking devices are properly locked into place.
- Before you turn ON the power supply, be sure to remove any dustproof labels that are put on the top of the Units when they are shipped. If the labels are not removed, heat will accumulate and malfunctions may occur.
- Before you connect a computer to the CPU Unit, disconnect the power supply plug of the computer from the AC outlet. Also, if the computer has an FG terminal, make the connections so that the FG terminal has the same electrical potential as the GR terminal on the Power Supply Unit.  
A difference in electrical potential between the computer and Controller may cause failure or malfunction.
- If the external power supply to an Output Unit or slave has polarity, connect it with the correct polarity. If the polarity is reversed, current may flow in the reverse direction and damage the connected devices regardless of the operation of the Controller.

## Power Supply Design

- Do not exceed the rated supply capacity of the Power Supply Units used in the Controller system. The rated supply capacities are given in the *NJ-series CPU Unit Hardware User's Manual* (Cat. No. W500).  
If the capacity is exceeded, operation may stop, malfunctions may occur, or data may not be backed up normally for power interruptions.  
Use only NJ-series Power Supply Units on NJ-series CPU Racks and Expansion Racks.  
Operation is not possible if a CJ-series Power Supply Unit is used with an NJ-series CPU Unit or an NJ-series Power Supply Unit is used with a CJ-series Units.
- Do not apply voltages or connect loads to the Output Units or slaves in excess of the maximum ratings.
- Surge current occurs when the power supply is turned ON. When selecting fuses or breakers for external circuits, consider the above precaution and allow sufficient margin in shut-off performance. Refer to the *NJ-series CPU Unit Hardware User's Manual* (Cat. No. W500) for surge current specifications.
- If the full dielectric strength voltage is applied or turned OFF using the switch on the tester, the generated impulse voltage may damage the Power Supply Unit. Use the adjustment on the tester to gradually increase and decrease the voltage.
- Apply the voltage between the Power Supply Unit's L1 or L2 terminal and the GR terminal when testing insulation and dielectric strength.
- Do not supply AC power from an inverter or other device with a square-wave output. Internal temperature rise may result in smoking or burning. Always input a sinusoidal wave with the frequency that is given in the *NJ-series CPU Unit Hardware User's Manual* (Cat. No. W500).
- Install external breakers and take other safety measures against short-circuiting in external wiring.

## When Power Is Turned ON

- It takes up to approximately 10 to 20 seconds to enter RUN mode after the power is turned ON. During that time, outputs will be OFF or will be the values specified in the Unit or slave settings, and external communications cannot be performed. Use the RUN output on the Power Supply Unit, for example, to implement fail-safe circuits so that external devices do not operate incorrectly.
- Configure the external circuits so that the power supply to the control system turns ON only after the power supply to the Controller has turned ON. If the power supply to the Controller is turned ON after the control power supply, temporary errors may result in incorrect control system signals because the output terminals on Output Units may momentarily turn ON when power supply is turned ON to the Controller.

## Actual Operation

Check the user program, data, and parameter settings for proper execution before you use them for actual operation.

## Turning OFF the Power Supply

- Do not turn OFF the power supply to the Controller while the BUSY indicator flashes. While the BUSY indicator is flashing, the user program and settings in the CPU Unit are being backed up in the built-in non-volatile memory. This data will not be backed up correctly if the power supply is turned OFF. The next time that the Controller is started, a Controller error in the major fault level will occur and operation will stop.

- Do not turn OFF the power supply or remove the SD Memory Card while SD Memory Card access is in progress (i.e., while the SD BUSY indicator flashes). Data may become corrupted, and the Controller will not operate correctly if it uses corrupted data. To remove an SD Memory Card from the CPU Unit when power is supplied to the CPU Unit, press the SD Memory Card power supply switch and wait for the SD BUSY indicator to turn OFF before you remove the Card.
- Do not disconnect the cable or turn OFF the power supply to the Controller when downloading data or the user program from the Sysmac Studio.
- Always turn OFF the power supply to the Controller before you attempt any of the following.
  - a) Mounting or removing I/O Units or the CPU Unit
  - b) Assembling the Units
  - c) Setting DIP switches or rotary switches
  - d) Connecting cables or wiring the system
  - e) Connecting or disconnecting the connectors

The Power Supply Unit may continue to supply power to the rest of the Controller for a few seconds after the power supply turns OFF. The PWR indicator is lit during this time. Confirm that the PWR indicator is not lit before you perform any of the above.

## Operation

- Confirm that the controlled system will not be adversely affected before you perform any of the following operations.
  - a) Changing the operating mode of the CPU Unit (including changing the setting of the Operating Mode at Startup)
  - b) Changing the user program or settings
  - c) Changing set values or present values
  - d) Forced Refreshing
- Always sufficiently check the safety at the connected devices before you change the settings of an EtherCAT slave or Special Unit.
- If two different function modules are used together, such as when you use CJ-series Basic Units and EtherCAT slaves, take suitable measures in the user program and external controls to ensure that safety is maintained in the controlled system if one of the function modules stops. The relevant outputs will stop if a partial fault level error occurs in one of the function modules.
- Always confirm safety at the connected equipment before you reset Controller errors with an event level of partial fault or higher for the EtherCAT Master Function Module.

When the error is reset, all slaves that were in any state other than Operational state due to a Controller error with an event level of partial fault or higher (in which outputs are disabled) will go to Operational state and the outputs will be enabled.

Before you reset all errors, confirm that no Controller errors with an event level of partial fault have occurred for the EtherCAT Master Function Module.

- Always confirm safety at the connected equipment before you reset Controller errors for a CJ-series Special Unit. When the Controller error is reset, the Unit where the Controller error with an event level of observation or higher will be restarted.

Before you reset all errors, confirm that no Controller errors with an event level of observation or higher have occurred for the CJ-series Special Unit. Observation level events do not appear on the Controller Error Tab Page, so it is possible that you may restart the CJ-series Special Unit without intending to do so.

You can check the status of the `_CJB_UnitErrSta[0,0]` to `_CJB_UnitErrSta[3,9]` error status variables on a Watch Tab Page to see if an observation level Controller error has occurred.

## Battery Backup

The user program and initial values for the variables are stored in non-volatile memory in the CPU Unit. The present values of variables with the Retain attribute and the values of the Holding, DM, and EM Areas in the memory used for CJ-series Units are backed up by a Battery.

If the Battery is not connected or the Battery is exhausted, the CPU Unit detects a Battery-backup Memory Check Error.

If that error is detected, variables with a Retain attribute are set to their initial values and the Holding, DM, and EM Areas in memory used for CJ-series Units are cleared to all zeros.

Perform thorough verifications and provide sufficient measures to ensure that the devices perform safe operation for the initial values of the variables with Retain attributes and the resulting operation.

## Debugging

- Forced refreshing ignores the results of user program execution and refreshes I/O with the specified values. If forced refreshing is used for inputs for which I/O refreshing is not supported, the inputs will first take the specified values, but they will then be overwritten by the user program. This operation differs from the force-set/reset functionality of the CJ-series PLCs.
- You cannot upload or download information for forced refreshing with the Sysmac Studio.  
After downloading data that contain the forced refreshing targets, change to RUN mode and then execute forced refreshing from the Sysmac Studio.  
Depending on the difference in the forced status, the control system may operate unexpectedly.
- Do not specify the same address for the AT specification for more than one variable.  
Doing so would allow the same entity to be accessed with different variable names, which would make the user program more difficult to understand and possibly cause programming mistakes.

## General Communications

- When you use data link communications, check the error information that is given in `_ErrSta` (Controller Error Status) to make sure that no error has occurred in the source device. Create a user program that uses reception data only when there is no error in the source device.  
If there is an error in the source device, the data for the data link may contain incorrect values.
- Unexpected operation may result if inappropriate data link tables are set. Even if appropriate data link tables have been set, confirm that the controlled system will not be adversely affected before you transfer the data link tables. The data links start automatically after the data link tables are transferred.
- All CPU Bus Units are restarted when routing tables are transferred from Support Software to the CPU Unit. Confirm that the system will not be adversely affected by restarting before you transfer the routing tables.
- Tag data links will stop between related nodes while tag data link parameters are transferred during Controller operation. Confirm that the system will not be adversely affected before you transfer the tag data link parameters.

## EtherNet/IP Communications

- All related EtherNet/IP nodes are reset when you transfer settings for the built-in EtherNet/IP port (including IP addresses and tag data links settings). Confirm that the system will not be adversely affected by resetting nodes before you transfer the settings.
- If EtherNet/IP tag data links (cyclic communications) are used with a repeating hub, the communications load on the network will increase. This will increase collisions and may prevent stable communications. Do not use repeating hubs on networks where tag data links are used. Use an Ethernet switch instead.

## EtherCAT Communications

- Make sure that the communications distance, number of nodes connected, and method of connection for EtherCAT are within specifications.  
Do not connect EtherCAT communications to other types of networks such as EtherNet/IP and a standard in-house LAN. An overload may cause the network to fail or malfunction.
- Malfunctions or unexpected operation may occur for some combinations of EtherCAT revision numbers of the master and slaves. If you disable the revision check in the network settings, you must use the Sysmac Studio to check the slave revision numbers in the master settings against the actual slave revision numbers. Check the functional compatibility by referring to the manuals or other references of the slaves before using. You can check the actual slave revisions from the Sysmac Studio or on slave nameplates.
- After you transfer the user program, the CPU Unit is restarted and communications with the EtherCAT slaves are cut off. During that period, the slave outputs behave according to the slave settings. The time that communications are cut off depends on the EtherCAT network configuration. If the EtherCAT network configuration contains only OMRON EtherCAT slaves, communications are cut off for a maximum of 45 seconds.  
Before you transfer the user program, confirm that the Units will not be adversely affected.
- If the *Fail-soft Operation Setting* parameter is set *Stop*, process data communications will stop for all slaves when an EtherCAT communications error is detected in a slave. For this reason, if Servo Drives are connected, the Servo Drives for all axes will be turned OFF. Make sure that the *Fail-soft Operation Setting* results in safe operation when a device error occurs.
- EtherCAT communications are not always established immediately after the power supply is turned ON. Use the system-defined variables in the user program to confirm that communications are established before attempting control operations.
- If frames sent to EtherCAT slaves are lost due to noise or other causes, slave I/O data is not transmitted, and unexpected operation may occur. If noise countermeasures are required, use the *\_EC\_InDataInvalid* (Input Data Disable) system-defined variable as an interlock condition in the user program.  
Refer to the *NJ/NX-series CPU Unit Built-in EtherCAT Port User's Manual* (Cat. No. W505) for details.  
The slave outputs behave according to the slave settings. Refer to the manuals for the slaves for details.
- When an EtherCAT slave is disconnected, communications will stop and control of the outputs will be lost not only for the disconnected slave, but for all slaves connected after it. Confirm that the system will not be adversely affected before you disconnect a slave.
- If you disconnect the cable from an EtherCAT slave to disconnect it from the network, any current communications frames may be lost. If frames are lost, slave I/O data is not transmitted, and unexpected operation may occur. Perform the following processing for a slave that needs to be replaced.
  - a) Create the program using the *\_EC\_InDataInvalid* (Input Data Disable) system-defined variable as an interlock condition.
  - b) Set the *PDO communications timeout detection count* to at least 2 in the EtherCAT master settings pane.

Refer to the *NJ/NX-series CPU Unit Built-in EtherCAT Port User's Manual* (Cat. No. W505) for details.

## Motion Control

- Confirm the axis number carefully before you perform an MC Test Run.
- The motor is stopped if communications are interrupted between the Sysmac Studio and the CPU Unit during an MC Test Run. Connect the communications cable between the computer and CPU Unit securely and confirm that the system will not be adversely affected before you perform an MC Test Run.
- Always execute the Save Cam Table instruction if you change any of the cam data from the user program in the CPU Unit or from the Sysmac Studio. If the cam data is not saved, the previous condition will be restored when the power is turned ON again, possibly causing unexpected machine operation.
- The positive drive prohibit input (POT), negative drive prohibit input (NOT), and home proximity input (DEC) of the Servo Drive are used by the MC Function Module as the positive limit input, negative limit input, and home proximity input. Make sure that the signal widths for all of these input signals are longer than the control period of the MC Function Module. If the input signal widths are shorter than the control period, the MC Function Module may not be able to detect the input signals, resulting in incorrect operation.
- During setting, ensure a correct relationship between the direction of commands issued by the Controller and the motor rotation direction of the Servo Drive. Otherwise, a robot may operate unexpectedly.
- Select a correct combination of the kinematics type and workspace type for the MC\_SetKinTransform (Set Kinematics Transformation) instruction to ensure proper workspace check. Check that the workspace check function is performed as intended.
- Adjust the home of the robot before you start robot operation.
- In case of SCARA singularity, the following 2 situations require special attention:
  - a) After executing a Home operation of SCARA robot axes, the robot needs to be moved with a Point-to-point movement (MC\_MoveDirectAbsolute) where the arm configuration is set by the instruction.
  - b) After commanding a SCARA robot to a singularity position and re-starting the controller, the robot needs to be moved with a point-to-point movement (MC\_MoveDirectAbsolute) where the arm configuration is set by the instruction.
- Do not operate the robot outside the workspace while the workspace check function is disabled. If you do so, the robot may be damaged.
- The Workspace Check is a robot protection but not a safety function. It checks only the robot end-effector against the workspace, but not the robot arms, neither the entire robot tool, if it is attached to the robot.
- Use the MC\_SyncOut (End Synchronization) instruction to stop the operation caused by the MC\_SyncLinearConveyor (Start Conveyor Synchronization) instruction.
- If you set 0 for the MaxVelocity (Velocity Error Detection Value) or MaxAcceleration (Acceleration Error Detection Value) parameter for input variables, or if you use the default (0) of these variables, the velocity error check or acceleration error check is not performed. Properly set the MaxVelocity (Velocity Error Detection Value) and MaxAcceleration (Acceleration Error Detection Value) to avoid an unexpected velocity and acceleration.
- If a robot tool (ToolID: 1 to 16) other than TCP0 is selected, the system does not perform the pre-check of the MaxVelocity (Velocity Error Detection Value) and MaxAcceleration (Acceleration ErrorDetection Value) parameters in the robotics instruction.
- When using Cartesian 2D kinematics, configure Plane correctly. Otherwise, a robot may operate unexpectedly.
- When Synchronized stop with Deceleration is selected in MC\_SyncOut (End Synchronization) instruction and Deceleration target time T5 is set to long time, it is possible to reach out of the workspace during deceleration. Do not disable workspace check.

- Select *Immediate stop* or *Immediate stop and Servo OFF* when using Delta 3, Delta 3R, Delta 2, Cartesian 2D Gantry or Cartesian 3D Gantry robot.
- Set 0 for Maximum Deceleration of axes, which are linked mechanically, when using Delta 3, Delta 3R, Delta 2, Cartesian 2D Gantry or Cartesian 3D Gantry robot.

## Battery Replacement

- The Battery may leak, rupture, heat, or ignite. Never short-circuit, charge, disassemble, heat, or incinerate the Battery or subject it to strong shock.
- Dispose of any Battery that has been dropped on the floor or otherwise subjected to excessive shock. Batteries that have been subjected to shock may leak if they are used.
- UL standards require that batteries be replaced by experienced technicians. Make sure that an experienced engineer is in charge of Battery replacement.
- Apply power for at least five minutes before you change the Battery. Install a new Battery within five minutes (at temperature of 25°C) after you turn OFF the power supply. If power is not supplied for at least 5 minutes, the saved data may be lost.

## Unit Replacement

- We recommend replacing the Battery with the power turned OFF to prevent the CPU Unit's sensitive internal components from being damaged by static electricity and to prevent malfunctions. The battery can be replaced without turning OFF the power supply. To do so, always touch a grounded piece of metal to discharge static electricity from your body before starting the procedure.  
After you replace the Battery, connect the Sysmac Studio and clear the *Low Battery Voltage* error.
- Make sure that the required data, including the user program, configurations/setup data, variables, and memory used for CJ-series Units, is transferred to the new CPU Unit and externally-connected devices before you restart operation. Be sure to include the routing tables, network parameters, and other CPU Bus Unit data, which are stored in the CPU Unit.

## Disposal

- The disposal of the product and Batteries may be subject to local government regulations. Dispose of the product and Batteries according to local ordinances as they apply.



廢電池請回收

- The following information must be displayed for all products that contain primary lithium batteries with a perchlorate content of 6 ppb or higher when shipped to or transported through the State of California, USA.

Perchlorate Material - special handling may apply.

See [www.dtsc.ca.gov/hazardouswaste/perchlorate](http://www.dtsc.ca.gov/hazardouswaste/perchlorate)

- The CPU Unit contains a primary lithium battery with a perchlorate content of 6 ppb or higher. Place the above information on the individual boxes and shipping boxes when shipping finished products that contain a CPU Unit to the State of California, USA.

# Precautions for Correct Use

---

- Do not install or store the Controller in the following locations. Operation may stop or malfunctions may occur.
  - a) Locations subject to direct sunlight
  - b) Locations subject to temperatures or humidity outside the range specified in the specifications
  - c) Locations subject to condensation as the result of severe changes in temperature
  - d) Locations subject to corrosive or flammable gases
  - e) Locations subject to dust (especially iron dust) or salts
  - f) Locations subject to exposure to water, oil, or chemicals
  - g) Locations subject to shock or vibration
- Take appropriate and sufficient countermeasures when installing the Controller in the following locations.
  - a) Locations subject to strong, high-frequency noise
  - b) Locations subject to static electricity or other forms of noise
  - c) Locations subject to strong electromagnetic fields
  - d) Locations subject to possible exposure to radioactivity
  - e) Locations close to power supplies
- Before touching a Unit, be sure to first touch a grounded metallic object in order to discharge any static build-up.
- Install the Controller away from sources of heat and ensure proper ventilation. Not doing so may result in malfunction, in operation stopping, or in burning.
- An I/O bus check error will occur and the Controller will stop if an I/O Connecting Cable's connector is disconnected from the Rack. Be sure that the connectors are secure.
- Do not allow foreign matter to enter the openings in the Unit. Doing so may result in Unit burning, electric shock, or failure.
- Do not allow wire clippings, shavings, or other foreign material to enter any Unit. Otherwise, Unit burning, failure, or malfunction may occur. Cover the Units or take other suitable countermeasures, especially during wiring work.
- For EtherCAT and EtherNet/IP, use the connection methods and cables that are specified in the *NJ/NX-series CPU Unit Built-in EtherCAT Port User's Manual* (Cat. No. W505) and the *NJ/NX-series CPU Unit Built-in EtherNet/IP Port User's Manual* (Cat. No. W506). Otherwise, communications may be faulty.
- Use the rated power supply voltage for the Power Supply Units. Take appropriate measures to ensure that the specified power with the rated voltage and frequency is supplied in places where the power supply is unstable.
- Make sure that the current capacity of the wire is sufficient. Otherwise, excessive heat may be generated. When cross-wiring terminals, the total current for all the terminals will flow in the wire. When wiring cross-overs, make sure that the current capacity of each of the wires is not exceeded.
- Do not touch the terminals on the Power Supply Unit immediately after turning OFF the power supply. Residual voltage may cause electrical shock.
- If you use reed switches for the input contacts for AC Input Units, use switches with a current capacity of 1 A or greater.
 

If the capacity of the reed switches is too low, surge current may fuse the contacts.

## Error Processing

---

When you create programs for applications that use the results of instructions that read the error status, consider how the detected error affects the system. For example, if a minor error is detected during Battery replacement, it can affect the system operation depending on the processing of the user program.

## Unit Replacement

---

- Refer to the CPU Bus Unit and Special I/O Unit operation manuals for details on the data required by each Unit.
- The absolute encoder home offset is backed up with a Battery in the CPU Unit.

When you change the combination of the CPU Unit and Servomotor, e.g., when you add or replace a Servomotor, define home again.

To restore the information without changing the CPU Unit-Servomotor combination, remove the absolute encoder home offset from the data to restore.

## Task Setup

---

If a Task Period Exceeded error occurs, shorten the programs to fit in the task period or increase the setting of the task period.

## Motion Control

---

- Use the system-defined variable in the user program to confirm that EtherCAT communications are established before you attempt to execute motion control instructions. Motion control instructions are not executed normally if EtherCAT communications are not established.
- Use the system-defined variables to monitor for errors in communications with the slaves that are controlled by the motion control function module. Motion control instructions are not executed normally if an error occur in slave communications.
- Before you start an MC Test Run, make sure that the operation parameters are set correctly.
- Do not download motion control settings during an MC Test Run.

## EtherCAT Communications

---

- Do not disconnect the EtherCAT slave cables during operation. The outputs will become unstable.
- Set the Servo Drives to stop operation if an error occurs in EtherCAT communications between the Controller and a Servo Drive.

## Battery Replacement

---

- Be sure to install a replacement Battery within two years of the production date shown on the Battery label.
- Turn ON the power after replacing the Battery for a CPU Unit that has been unused for a long time. Leaving the CPU Unit unused again without turning ON the power even once after the battery is replaced may result in a shorter battery life.
- When you replace the Battery, use the CJ1W-BAT01 Battery Set.

## SD Memory Card

---

- Insert the SD Memory Card all the way.
- Do not turn OFF the power supply to the Controller during SD Memory Card access. The files may be corrupted.

If there is a corrupted file in the SD Memory Card, the file is automatically deleted by the restoration function when the power supply is turned ON.

# Regulations and Standards

## Using Product Outside Japan

If you export (or provide a non-resident with) this product or a part of this product that falls under the category of goods (or technologies) specified by the Foreign Exchange and Foreign Trade Control Law as those which require permission or approval for export, you must obtain permission or approval (or service transaction permission) pursuant to the law.

## Conformance to EC Directives

### Applicable Directives

- EMC Directives
- Low Voltage Directive

### Concepts

#### ● EMC Directives

OMRON devices that comply with EC Directives also conform to the related EMC standards so that they can be more easily built into other devices or the overall machine. The actual products have been checked for conformity to EMC standards.\*1

Whether the products conform to the standards in the system used by the customer, however, must be checked by the customer. EMC-related performance of the OMRON devices that comply with EC Directives will vary depending on the configuration, wiring, and other conditions of the equipment or control panel on which the OMRON devices are installed. The customer must, therefore, perform the final check to confirm that devices and the overall machine conform to EMC standards.

\*1. Applicable EMC (Electromagnetic Compatibility) standards are as follows: EN 61131-2 and EN 61000-6-2 for EMS (Electromagnetic Susceptibility), and EN 61131-2 and EN 61000-6-4 for EMI (Electromagnetic Interference).

EN 61000-6-4 (Radiated emission) is conformed based on 10-m regulations.

#### ● Low Voltage Directive

Devices operating at voltages of 50 to 1,000 VAC and 75 to 1,500 VDC must meet the required safety standards. The applicable directive is EN 61131-2.

#### ● Conformance to EC Directives

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

- The NJ/NX-series Controller must be installed within a control panel.
- You must use reinforced insulation or double insulation for the DC power supplies connected to DC Power Supply Units and I/O Units.
- NJ/NX-series Controllers that comply with EC Directives also conform to the Common Emission Standard (EN 61000-6-4). Radiated emission characteristics (10-m regulations) may vary depending on the configuration of the control panel used, other devices connected, wiring, and other conditions.

You must therefore confirm that the overall machine or equipment complies with EC Directives.

## Conformance to Shipbuilding Standards

This product complies with the different shipbuilding standards. Applicability to the shipbuilding standards is based on certain usage conditions. It may not be possible to use the product in some locations. Contact your OMRON representative before attempting to use a Controller on a ship.

### Usage Conditions for NK and LR Shipbuilding Standards

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

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

## Using Controllers in Robot Systems

The NJ501-4300/-4310/-4320/-4400/-4500 Controllers do not conform to laws and regulations relating to the safety of industrial robot application.

When you use the NJ501-4300/-4310/-4320/-4400/-4500 Controllers in robot systems that uses industrial robots, be sure to verify the conformance to laws and regulations relating to the safety of industrial robot application. Take measures to ensure safety as required.

## Software Licenses and Copyrights

This product incorporates certain third party software. The license and copyright information associated with this software is given in the web pages relating to the NJ501-1300/-1400/-1500 Controllers at [http://www.fa.omron.co.jp/nj\\_info\\_e/](http://www.fa.omron.co.jp/nj_info_e/).

# Versions

Hardware and software upgrades relating to the NJ-series Units and EtherCAT slaves are managed with the number called "unit version". If any change is made in specifications of hardware or software, the unit version number is replaced. Even when Units or EtherCAT slaves have the same model number, they will have differences in functions and performance if they have different unit versions.

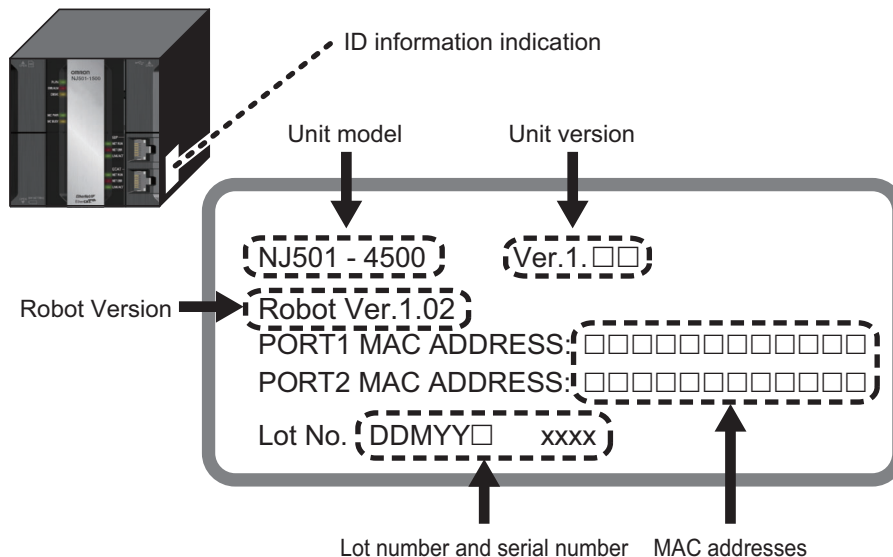
## Confirming Versions

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

### Confirming Versions with ID Information Indications

The version is given on the ID information indication of the products.

The following figure shows the design of the ID information for NJ Robotics NJ501-4500 CPU Units.



## Confirming Versions with Sysmac Studio

You can use the Sysmac Studio to check the version. The procedure to check the versions differs between a Unit and an EtherCAT slave.

### ● Version of Units

The version of Units are given in the Production Information when the Sysmac Studio is online. You can check the versions of the following Units: CPU Unit, CJ-series Special I/O Units, and CJ-series CPU Bus Units. You cannot check the versions of CJ-series Basic I/O Units from the Sysmac Studio.

Use the following procedure.

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

The Unit Editor is displayed.

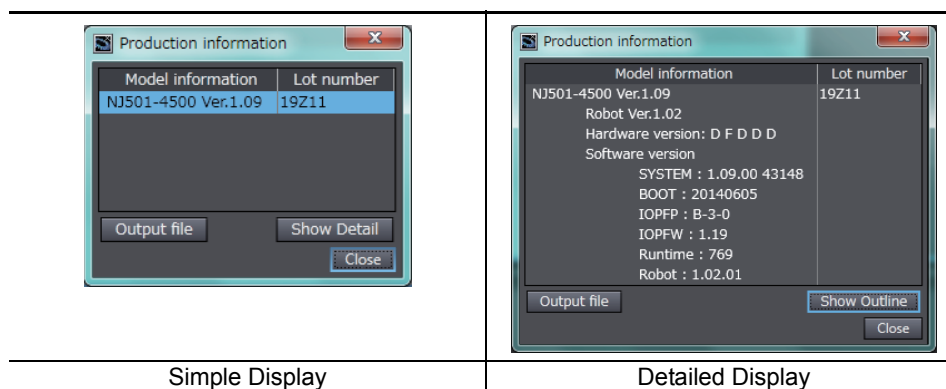
- 2 Right-click any open space in the Unit Editor and select **Display Production Information**.

The Production Information Dialog Box is displayed.

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

- 1 Select either **Show Outline** or **Show Detail** on the lower right corner of the Production Information Dialog Box.

The displayed information in the Production Information Dialog Box is switched between the outline and detail.



The displayed items differ between the simple display and detailed display. The detailed display gives the unit version, hardware version and software version. The simple display gives the unit version only.

## ● Version of EtherCAT Slaves

The version of EtherCAT slaves are given in the Production Information Dialog Box when the Sysmac Studio is online. Use the following procedure.

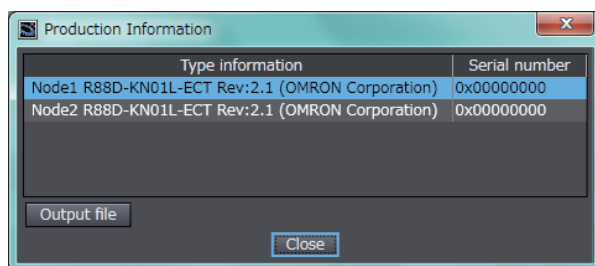
- 1 Double-click **EtherCAT** under **Configurations and Setup** in the Multiview Explorer. Or, right-click **EtherCAT** under **Configurations and Setup** and select **Edit** from the menu.

The EtherCAT Tab Page is displayed.

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

The Production Information Dialog Box is displayed.

The unit version is shown after "Rev".



# Related Manuals

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

Manual name	Cat. No.	Model numbers	Application	Description
NJ-series CPU Unit Hardware User's Manual	W500	NJ501-□□□□ NJ301-□□□□ NJ101-□□□□	Learning the basic specifications of the NJ-series CPU Units, including introductory information, designing, installation, and maintenance.  Mainly hardware information is provided.	Provides an introduction to the entire NJ-series system along with the following information on the CPU Unit. <ul style="list-style-type: none"> <li>• Features and system configuration</li> <li>• Overview</li> <li>• Part names and functions</li> <li>• General specifications</li> <li>• Installation and wiring</li> <li>• Maintenance and inspection</li> </ul> Use this manual together with the <i>NJ/NX-series CPU Unit Software User's Manual</i> (Cat. No. W501).
NJ/NX-series CPU Unit Software User's Manual	W501	NX701-□□□□ NJ501-□□□□ NJ301-□□□□ NJ101-□□□□	Learning how to program and set up an NJ/NX-series CPU Unit.  Mainly software information is provided.	Provides the following information on a Controller built with an NJ/NX-series CPU Unit. <ul style="list-style-type: none"> <li>• CPU Unit operation</li> <li>• CPU Unit features</li> <li>• Initial settings</li> <li>• Language specifications and programming based on IEC 61131-3</li> </ul> Use this manual together with the <i>NJ-series CPU Unit Hardware User's Manual</i> (Cat. No. W500).
NJ/NX-series Instructions Reference Manual	W502	NX701-□□□□ NJ501-□□□□ NJ301-□□□□ NJ101-□□□□	Learning detailed specifications on the basic instructions of an NJ/NX-series CPU Unit.	Describes the instructions in the instruction set (IEC 61131-3 specifications).  When programming, use this manual together with the <i>NJ-series CPU Unit Hardware User's Manual</i> (Cat. No. W500) and <i>NJ/NX-series CPU Unit Software User's Manual</i> (Cat. No. W501).
NJ/NX-series CPU Unit Motion Control User's Manual	W507	NX701-□□□□ NJ501-□□□□ NJ301-□□□□ NJ101-□□□□	Learning about motion control settings and programming concepts.	Describes the settings and operation of the CPU Unit and programming concepts for motion control.  When programming, use this manual together with the <i>NJ-series CPU Unit Hardware User's Manual</i> (Cat. No. W500) and <i>NJ/NX-series CPU Unit Software User's Manual</i> (Cat. No. W501).
NJ/NX-series Motion Control Instructions Reference Manual	W508	NX701-□□□□ NJ501-□□□□ NJ301-□□□□ NJ101-□□□□	Learning about the specifications of the motion control instructions that are provided by OMRON.	Describes the motion control instructions.  When programming, use this manual together with the <i>NJ-series CPU Unit Hardware User's Manual</i> (Cat. No. W500), <i>NJ/NX-series CPU Unit Software User's Manual</i> (Cat. No. W501), and <i>NJ/NX-series CPU Unit Motion Control User's Manual</i> (Cat. No. W507).
NJ/NX-series CPU Unit Built-in EtherCAT® Port User's Manual	W505	NX701-□□□□ NJ501-□□□□ NJ301-□□□□ NJ101-□□□□	Using the built-in EtherCAT port on an NJ/NX-series CPU Unit.	Provides information on the built-in EtherCAT port.  This manual provides an introduction and information on the configuration, features, and setup.  Use this manual together with the <i>NJ-series CPU Unit Hardware User's Manual</i> (Cat. No. W500) and <i>NJ/NX-series CPU Unit Software User's Manual</i> (Cat. No. W501).

## Related Manuals

Manual name	Cat. No.	Model numbers	Application	Description
NJ/NX-series CPU Unit Built-in EtherNet/IP™ Port User's Manual	W506	NX701-□□□□ NJ501-□□□□ NJ301-□□□□ NJ101-□□□□	Using the built-in EtherNet/IP port on an NJ/NX-series CPU Unit.	Provides information on the built-in EtherNet/IP port.  This manual provides information on the basic setup, tag data links, and other features.  Use this manual together with the <i>NJ-series CPU Unit Hardware User's Manual</i> (Cat. No. W500) and <i>NJ/NX-series CPU Unit Software User's Manual</i> (Cat. No. W501).
NJ-series Database Connection CPU Units User's Manual	W527	NJ501-1520 NJ501-1420 NJ501-1320 NJ501-4320	Using the database connection service with NJ-series Controllers.	Describes the database connection service.
NJ-series NJ Robotics CPU Unit User's Manual	W539	NJ501-4□□□	Using the robot control with NJ-series Controllers.	Describes the robot control.  Use this manual together with the <i>NJ/NX-series CPU Unit Motion Control User's Manual</i> (Cat. No. W507) and the <i>NJ/NX-series Motion Control Instructions Reference Manual</i> (Cat. No. W508).
NJ/NX-series Troubleshooting Manual	W503	NX701-□□□□ NJ501-□□□□ NJ301-□□□□ NJ101-□□□□	Learning about the errors that may be detected in an NJ/NX-series Controller.	Describes concepts on managing errors that may be detected in an NJ/NX-series Controller and information on individual errors.  Use this manual together with the <i>NJ-series CPU Unit Hardware User's Manual</i> (Cat. No. W500) and <i>NJ/NX-series CPU Unit Software User's Manual</i> (Cat. No. W501).
Sysmac Studio Version 1 Operation Manual	W504	SYSMAC-SE2□□□	Learning about the operating procedures and functions of the Sysmac Studio.	Describes the operating procedures of the Sysmac Studio.
NX-series Position Interface Units User's Manual	W524	NX-EC0□□□ NX-ECS□□□ NX-PG0□□□	Learning how to use NX-series Position Interface Units.	Describes the hardware, setup, and functions for the NX-series Incremental Encoder Input Units, SSI Input Units, and Pulse Output Unit.

# Terminology

Term	Description
NJ501-1□□□	The model number NJ501-1300, NJ501-1400 or NJ501-1500.
NJ-series NJ Robotics CPU Unit	The model number NJ501-4300, NJ501-4310, NJ501-4320, NJ501-4400 or NJ501-4500. These models may also be written as "NJ501-4□□□".
Sysmac Studio Robot Options	The optional functions of the Sysmac Studio provided to use NJ501-4□□□ robot functions.
Robot	The mechanism (e.g. Delta and SCARA) that consists of multiple links and axes.
Delta	The robots that use parallel link mechanism. This Controller can control 2-parallel-link type robots and 3-parallel-link type robots. Some robots have a rotational axis at their wrist.
Delta3	In this manual, a Delta robot with three parallel links is called Delta3.
Delta3R	In this manual, a Delta robot with three parallel links and a rotational wrist axis is called Delta3R.
Delta2	In this manual, a Delta robot with two parallel links is called Delta2.
Cartesian Robot	A robot arm with prismatic joints, which allows movement along one or more of the three axes in the X, Y, Z coordinate system.
Gantry Robot	A Cartesian robot which moves along with the gantry system as X-axis. The gantry system is composed with two linear axes which positioned in parallel.
H-Bot	A robot composed of two rotary drives which are connected by a single H-shaped circumferential timing belt around two parallel linear rails.
T-Bot	A variation of H-Bot with a T-shape and one linear rail instead of two.
Expansion1 robot	A right-arm planar robot with 3 axes, moving in XY plane.
SCARA	A SCARA robot that could be either RRP or PRR variant.
SCARA RRP	A SCARA robot with 2 rotational and 1 prismatic joints.
SCARA PRR	A SCARA robot with 1 prismatic and 2 rotational joints.
SCARA RRP+R	A SCARA RRP with one additional rotational axis to control the wrist.
SCARA PRR+R	A SCARA PRR with one additional rotational axis to control the wrist.
Kinematics	In this manual, the kinematics refers to a set of expressions required for the following two transforms. <ul style="list-style-type: none"> <li>• Transform from the axis coordinate systems (ACS) of robot links and axes to the robot-specific Cartesian coordinate system (MCS).</li> <li>• Transform from the robot-specific Cartesian coordinate system (MCS) to the axis coordinate systems (ACS) of robot links and axes.</li> </ul>
Inverse Kinematics	Transform from a position in the robot-specific Cartesian coordinate system (MCS) to a position in axis coordinate systems (ACS) of robot links and axes.
Direct Kinematics	Transform from a position in axis coordinate systems (ACS) of robot links and axes to a position in the robot-specific Cartesian coordinate system (MCS).
Axis Coordinate System	A coordinate system for rotation or linear motion specified for each axis of the robot. The abbreviation is ACS.
Machine Coordinate System	A Cartesian coordinate system which is specific to the machine (robot). The abbreviation is MCS.
User Coordinate System	A Cartesian coordinate system defined by the user. The abbreviation is UCS.
Tool Coordinate System	A Cartesian coordinate system which has TCP at its origin. The abbreviation is TCS.
TCS0 (Tool Coordinate System 0)	The default TCS. Its origin is TCP0.

## Terminology

Term	Description
TCS $i$ (Tool Coordinate System $i$ )	A TCS currently selected by the robot. The symbol $i$ represents ToolID of the TCS. The $i$ takes on values from 1 to 16.
TCP (Tool Center Point)	The tip of the machine (robot) that performs tasks. Specify the position of TCP when you perform positioning for the Cartesian coordinate system.
TCP0 (Tool Center Point 0)	This is the default TCP.
TCP $i$ (Tool Center Point $i$ )	The TCP of a certain tool. The symbol $i$ represents ToolID of the TCP. The $i$ takes on values from 1 to 16.
Robot TCP (Robot Tool Center Point)	The default TCP of the robot. This is the same thing as TCP0.
Active TCP (Active Tool Center Point)	A TCP which is currently selected by the robot. If you select ToolID $i$ for a TCP, the selected TCP becomes the active TCP. The $i$ takes on values from 1 to 16.
Fixed Frame	In a Delta robot, it refers to a frame to which the axes motors are attached.
Moving Frame	In a Delta robot, it refers to a frame driven by the axes motors.
Robot Tool	A mechanical part attached to the robot flange. This part performs tasks for workpieces.
Vision & Robot Integrated Simulation	A simulation of the pick-and-place application where this Controller is used in combination with one or more vision sensors and Delta robots.
Robot Additional Option	The optional functions required to perform a Vision & Robot integrated simulation. The option becomes available when you enter a license number in the Sysmac Studio version 1.14 or higher.

# Revision History

---

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

<b>Cat. No.</b>	<b>W539-E1-05</b>
-----------------	-------------------

↑  
Revision code

Revision code	Date	Revised content
01	April 2015	Original production
02	October 2015	Added information on Vision & Robot integrated simulation.
03	April 2016	Made changes accompanying release of robot version 1.03 of the CPU Unit.
04	June 2016	Updated the EtherNet/IP logo.
05	April 2017	Made changes accompanying release of robot version 1.04 of the CPU Unit.

## Revision History



# Overview of NJ Robotics CPU Unit

This section describes the features, system configuration, operating procedure, specifications and control configuration of an NJ-series NJ Robotics CPU Unit.

---

<b>1-1</b>	<b>Features and System Configuration of Unit</b>	<b>1-2</b>
1-1-1	Features of NJ Robotics CPU Unit	1-2
1-1-2	Introduction to the System Configurations	1-4
<b>1-2</b>	<b>Operation Procedure of Unit</b>	<b>1-7</b>
<b>1-3</b>	<b>Specifications of Unit</b>	<b>1-8</b>
1-3-1	Performance Specifications	1-8
1-3-2	Function Specifications	1-9
<b>1-4</b>	<b>Robot Control Configuration</b>	<b>1-11</b>

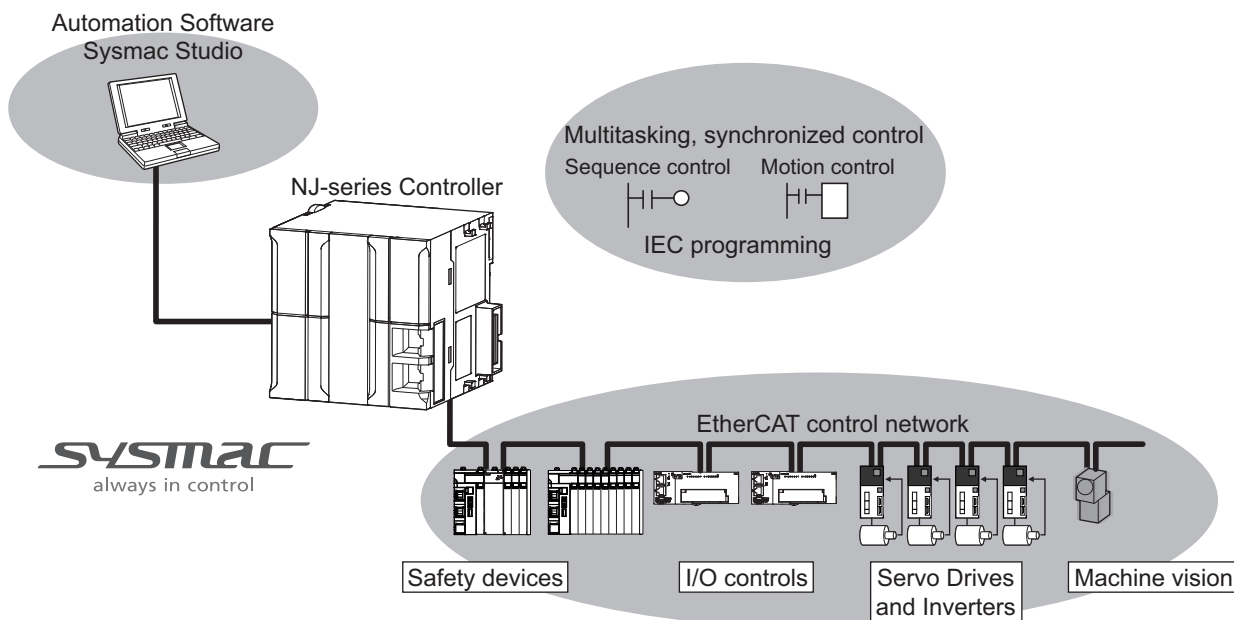
# 1-1 Features and System Configuration of Unit

The NJ-series Machine Automation Controllers are next-generation machine automation controllers that provide the functionality and high-speed performance that are required for machine control. They provide the safety, reliability, and maintainability that are required of industrial controllers.

The NJ-series Controllers provide the functionality of previous OMRON PLCs, and they also provide the functionality that is required for motion control. Synchronized control of I/O devices on high-speed EtherCAT can be applied to safety devices, vision systems, motion equipment, discrete I/O, and more.

OMRON offers the new Sysmac Series of control devices designed with unified communications specifications and user interface specifications. The NJ-series Machine Automation Controllers are part of the Sysmac Series. You can use them together with EtherCAT slaves, other Sysmac products, and the Sysmac Studio Automation Software to achieve optimum functionality and ease of operation.

With a system that is created from Sysmac products, you can connect components and operate the system through unified concepts and usability.



## 1-1-1 Features of NJ Robotics CPU Unit

The NJ Robotics CPU Unit has the following features.

### ● Integrated Sequence Control and Motion Control

An NJ-series CPU Unit can perform both sequence control and motion control. You can simultaneously achieve both sequence control and multi-axes synchronized control. Sequence control, motion control, and I/O refreshing are all executed in the same control period.

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.

### ● Multitasking

You can assign I/O refreshing and programs to tasks and then specify execution conditions and execution order for them to flexibly combine controls that suit the application.

- **Robot Instructions Supported**

Robot operation, single-axis operation and synchronized operation can all be achieved with the Motion Control Function Module (hereinafter called "MC Function Module").

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

The NJ-series Controllers support language specifications that are based on IEC 61131-3. To these, OMRON has added our own improvements. Motion control instructions that are based on PLCopen standards and an instruction set (POUs) that follows IEC rules are provided.

- **A Wealth of Security Features**

The many security features of the NJ-series Controllers include operation authority settings and restriction of program execution with IDs.

- **Complete Controller Monitoring**

The CPU Unit monitors events in all parts of the Controller, including mounted Units and EtherCAT slaves.

Troubleshooting information for errors is displayed on the Sysmac Studio or on an HMI. Events are also recorded in logs.

- **Sysmac Studio Automation Software**

The Sysmac Studio provides an integrated development environment that covers not only the Controller, but also covers peripheral devices and devices on EtherCAT. You can use consistent procedures for all devices regardless of the differences in the devices. The Sysmac Studio supports all phases of Controller application, from designing through debugging, simulations, commissioning, and changes during operation.

- **A Wealth of Simulation Features**

The many simulation features include execution, debugging, and task execution time estimates on a virtual controller.

## 1-1-2 Introduction to the System Configurations

The NJ Robotics CPU Unit supports the following system configurations.

### Basic System Configuration

The NJ-series basic configurations include the EtherCAT network configuration and the Support Software.

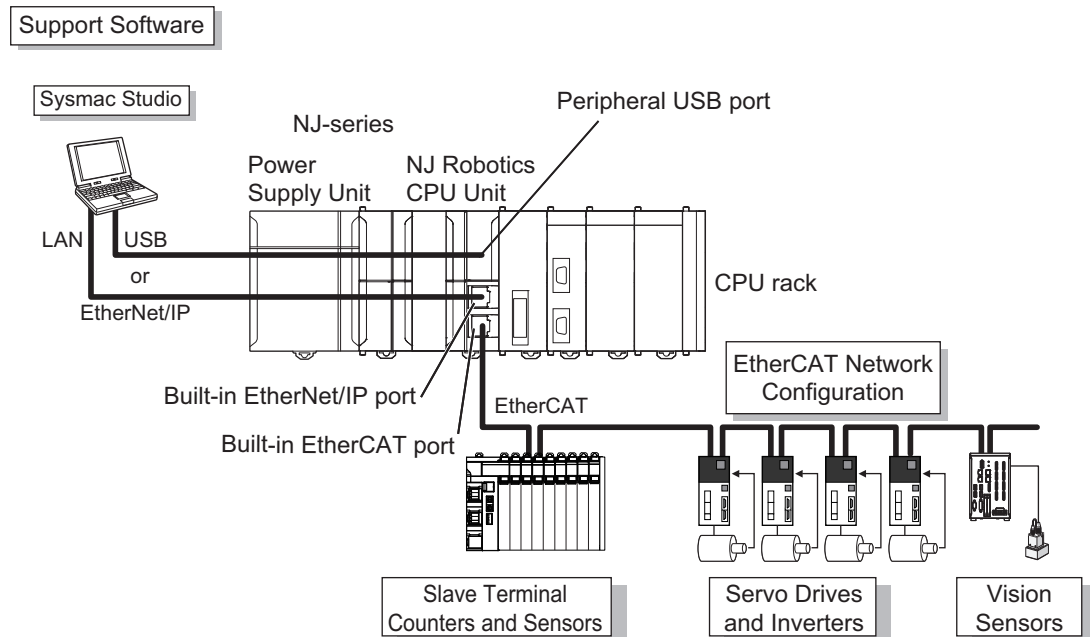
#### ● EtherCAT Network Configuration

You can use the built-in EtherCAT port to connect to EtherCAT Slave Terminals, to general-purpose slaves for analog and digital I/O, and to Servo Drives and encoder input slaves. An EtherCAT network configuration enables precise sequence and motion control in a fixed cycle with very little deviation.

#### ● Support Software

The Support Software is connected to the peripheral USB port on the CPU Unit with a commercially available USB cable. You can also connect it through an Ethernet cable that is connected to the built-in EtherNet/IP port.

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

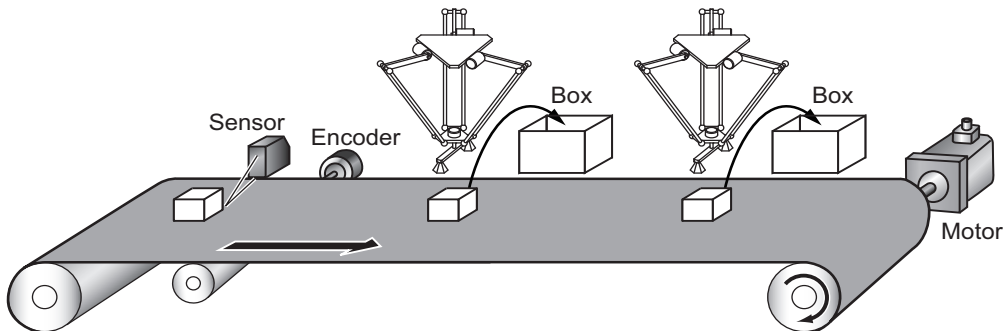


## Robot System Configuration

The following figures show the robot system configurations.

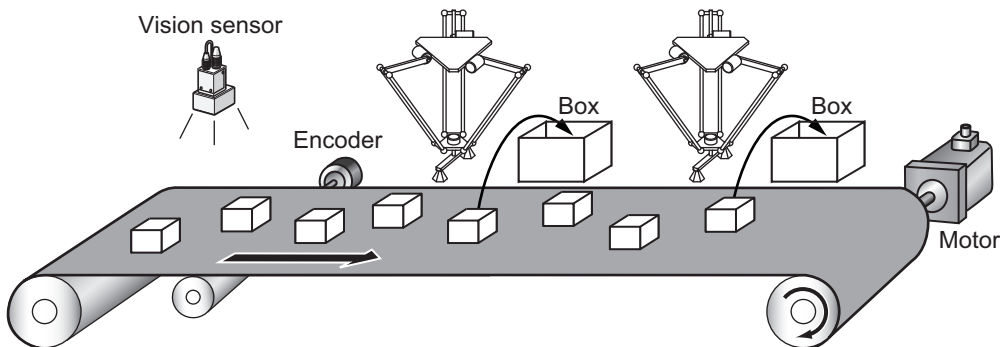
### ● Packing 1

The sensor detects workpieces and the robots put them into boxes.



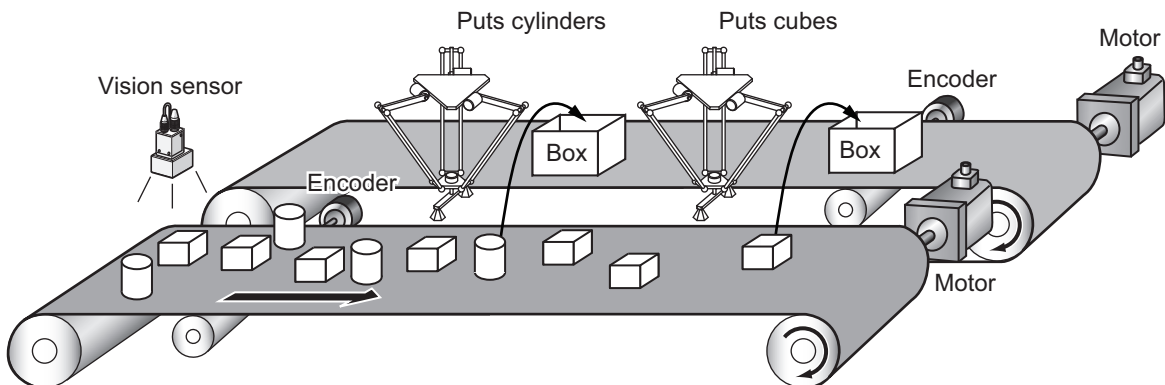
### ● Packing 2

The vision sensor detects workpieces conveyed randomly, and the robots put them into boxes.



### ● Packing 3

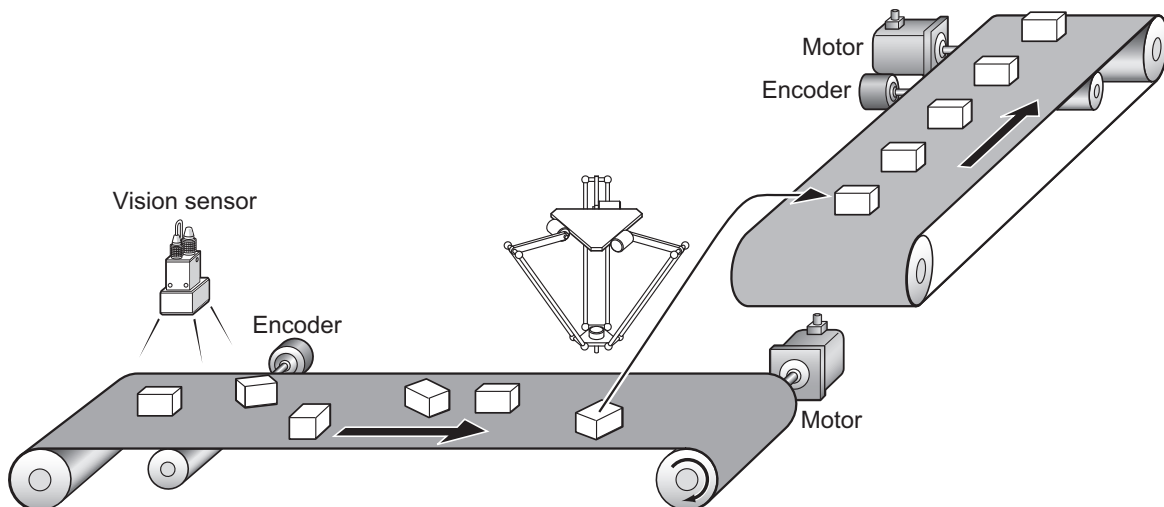
The vision sensor detects different shapes of workpieces and the robots put them accordingly.



# 1 Overview of NJ Robotics CPU Unit

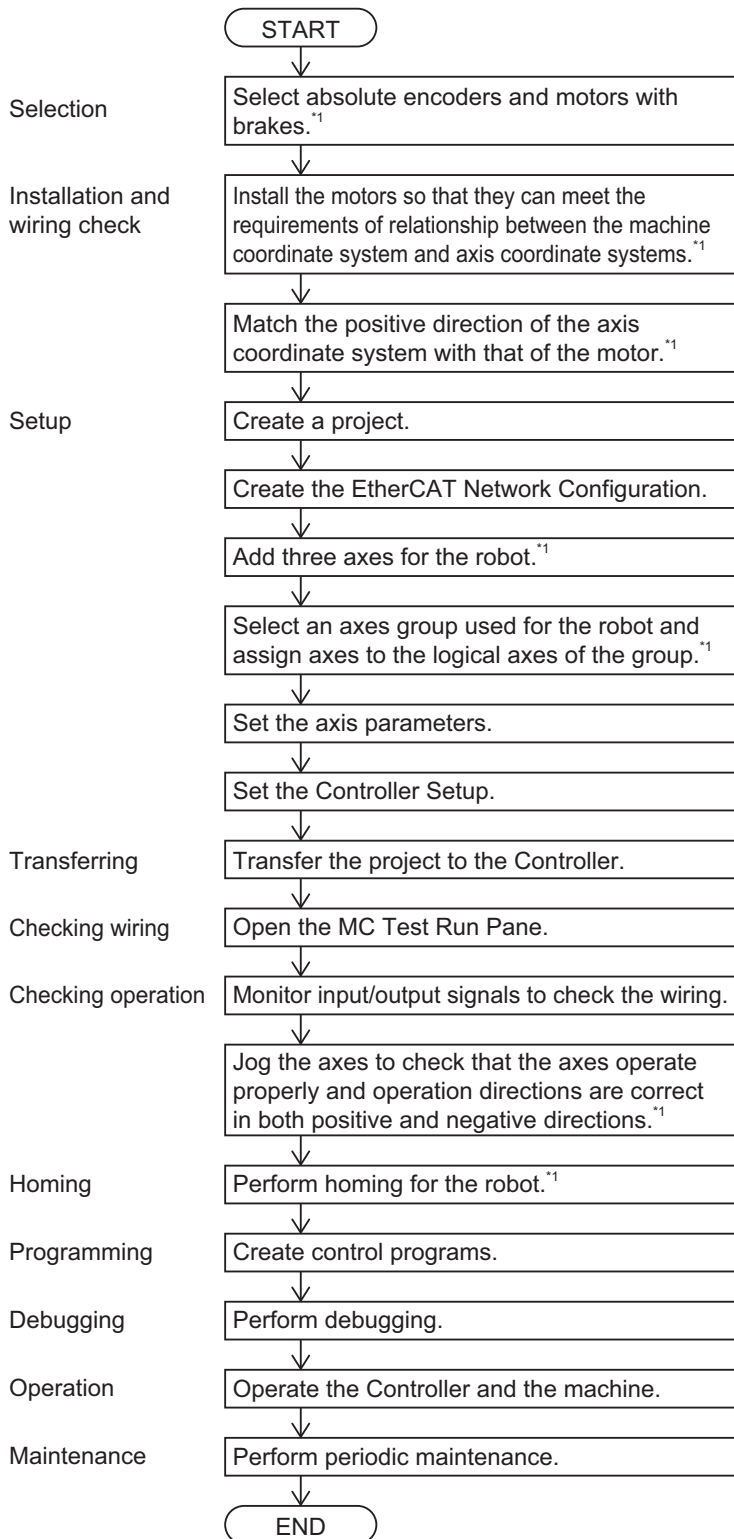
## ● Alignment

The robots align workpieces conveyed randomly, and line them up on another conveyor.



# 1-2 Operation Procedure of Unit

This section provides the procedure to operate a robot.



\*1. This procedure is different from that for NJ501-1□□□ Units. Refer to the *NJ/NX-series CPU Unit Motion Control User's Manual* (Cat. No. W507) for information on other procedures.

## 1-3 Specifications of Unit

This section gives the performance specifications and function specifications of an NJ Robotics CPU Unit.

### 1-3-1 Performance Specifications

The following robot functions are added to the MC Function Module for NJ Robotics CPU Units with model numbers NJ501-1□□□.

Item	Overview
Coordinate System	The two types of systems, the machine coordinate system (MCS) and user coordinate system (UCS), are provided for robot operation.
Robot Parameter Settings	Sets the robot parameters including the kinematics type and link length.
Workspace Check	Checks whether the robot operates within the range of motion (workspace).
User Coordinate System	You can set the user coordinate systems for each robot.
Robot Tool	You can set multiple robot tools for each robot.
Monitoring	Reads the current position and current velocity of the robot.
Time-specified Absolute Positioning	Moves the robot to a set position in a set time period.
Conveyor Synchronization	Makes the robot to track the workpiece on the conveyor.
Inverse Kinematics	Transforms the coordinates (X, Y, Z, Rx, Ry, Rz) of the robot to the axis coordinates.
Robot Jogging	Jogs the robot.
Transition	You can select the method to make a smooth transition trajectory between robot operation instructions.
Robot Velocity and Acceleration check	Checks robot maximum velocity and acceleration (pre-check and online check).
Synchronized Offset functionality	You can set offsets during conveyor synchronization.

## 1-3-2 Function Specifications

The following table shows the function specifications of an NJ Robotics CPU Unit.

Item		Specification				
		NJ501-4500	NJ501-4400	NJ501-4300	NJ501-4320	NJ501-4310
Applicable robots		Delta3, Delta3R, Delta2, Cartesian 3D, Cartesian 3D Gantry, Cartesian 2D, Cartesian 2D Gantry, H-Bot <sup>*7</sup> , SCARA RRP, SCARA RRP+R, SCARA PRR, SCARA PRR+R and Expansion1 <sup>*8</sup>				
Controllable Servo Drives		OMRON G5-series Servo Drives with built-in EtherCAT communications <sup>*1</sup>				
Controllable encoder input terminals		OMRON GX-series EtherCAT Remote I/O Terminals GX-EC0211/-EC0241 <sup>*2</sup> OMRON NX-series Incremental Encoder Units NX-EC0122/-EC0222/-EC0142 <sup>*3</sup> OMRON NX-series SSI Input Units NX-ECS112/-ECS212 <sup>*3</sup>				
Control method		Control commands using EtherCAT communications				
Control modes		Position control (Cyclic Synchronous Position Control Mode) Velocity control (Cyclic Synchronous Velocity Control Mode) Torque control (Cyclic Synchronous Torque Control Mode)				
Number of controlled axes	Maximum number of controlled axes	64 axes	32 axes	16 axes		
	Single-axis control	64 axes max.	32 axes max.	16 axes max.		
	Linear interpolation control <sup>*4</sup>	4 axes max. per axes group				
	Circular interpolation control <sup>*4</sup>	2 axes per axes group				
Number of axes groups		32 groups max.				
Unit conversion	Unit of display	pulse, mm, nm, degree and inch				
	Electronic gear ratio	Pulse per motor rotation/travel distance per motor rotation				
Positions that can be managed		Command positions and actual positions				
Position command values		Negative or positive long reals (LREAL) or 0 (unit: command units <sup>*5</sup> )				
Velocity command values		Negative or positive long reals (LREAL) or 0 (unit: command units/s) <sup>*6</sup>				
Acceleration command values and deceleration command values		Positive long reals (LREAL) or 0 (unit: command units/s <sup>2</sup> )				
Jerk command values		Positive long reals (LREAL) or 0 (unit: command units/s <sup>2</sup> )				
Override factors		0.00% or 0.01% to 500.00%				
Axis type		Servo axes, Virtual servo axes, Encoder axes, and Virtual encoder axes				
Motion control period		The same control period as that is used for the process data communications cycle for EtherCAT. Use the robot functions with the period of 1 ms, 2 ms or 4 ms.				
Cams	Number of cam data	65,535 points max. per cam table 1,048,560 points max. for all cam tables				
	Number of cam tables	640 tables max.				
	Cam profile curve	Created with the Cam Editor.				
	Overwriting cam data	Can be overwritten from the user program.				

## 1 Overview of NJ Robotics CPU Unit

Item		Specification				
		NJ501-4500	NJ501-4400	NJ501-4300	NJ501-4320	NJ501-4310
Coordinate System	Axis coordinate system (ACS)	1 system per axis				
	Machine coordinate system (MCS)	1 system per group				
	User coordinate system (UCS)	16 systems per group				
	Tool coordinate system (TCS)	17 systems per group (default + defined robot tools)				
Robot tool		16 tools group				
Maximum number of robots		8	8	8		1

\*1. Unit version 2.1 or later is recommended.

\*2. Unit version 1.1 or later is recommended.

\*3. Attach the unit to a NX-ECC201 EtherCAT Coupler Unit.

\*4. You cannot use this function for robot control.

\*5. Positions can be set within a 40-bit signed integer range when converted to pulses.

\*6. The maximum velocity command value is 400 Mpps (in pulses) when you use OMRON G5-series Servo Drives.

\*7. Cartesian 3D, Cartesian 3D Gantry, Cartesian 2D, Cartesian 2D Gantry and H-Bot are applicable only in Robot Version 1.03 or later.

\*8. SCARA RRP, SCARA RRP+R, SCARA PRR, SCARA PRR+R and Expansion1 are applicable only in Robot Version 1.04 or later.



#### Version Information

SCARA RRP, SCARA RRP+R, SCARA PRR, SCARA PRR+R and Expansion1 are applicable only in Robot Version 1.04 or later.

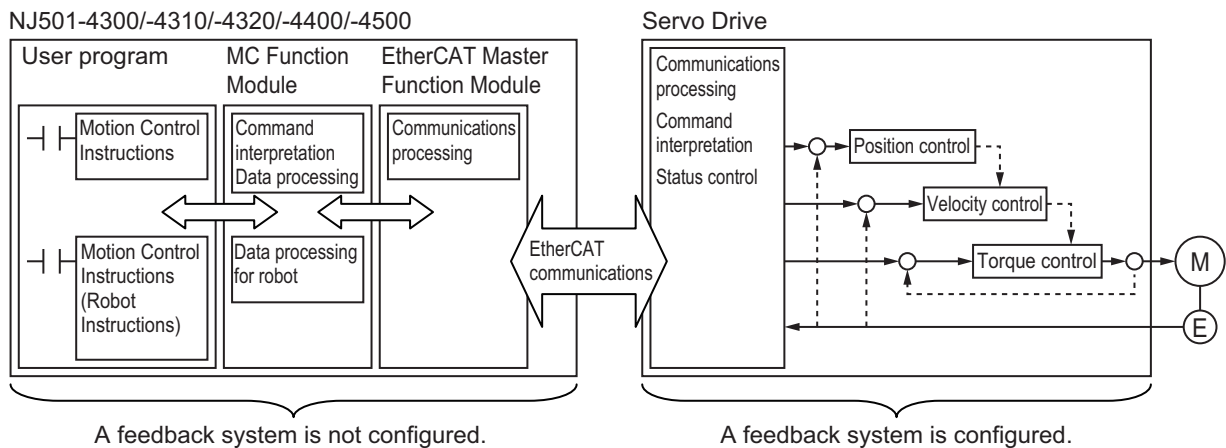
## 1-4 Robot Control Configuration

A control system built with Servo Drives generally controls motor operation with a semi-closed loop. The semi-closed loop uses an encoder attached to the motor to detect the amount of rotation that has been performed by the motor in response to the command value. This is provided as feedback of the machine's travel distance. The following error between the command value and actual motor rotation is calculated and control is performed to bring the following error to zero.

In a machine configuration that uses the MC Function Module, no feedback information is provided for the commands from the user program in the CPU Unit. A feedback system is built into the Servo Drive.

The NJ Robotics CPU Units are the NJ501-1□□□ Units to which the instructions and data processing for robot operation were added, as shown in the following figure.

With these instructions and data processing, the NJ Robotics CPU Unit can perform robot control in the same way as other NJ501-1□□□ Units' motion control, using function blocks based on IEC 61131-3.



For details on the configuration and principles of the motion control system, refer to the *NJ/NX-series CPU Unit Motion Control User's Manual* (Cat. No. W507).

# 1 Overview of NJ Robotics CPU Unit



# Features of NJ Robotics CPU Unit

This section describes the types of robots you can control, their coordinate systems and functions.

---

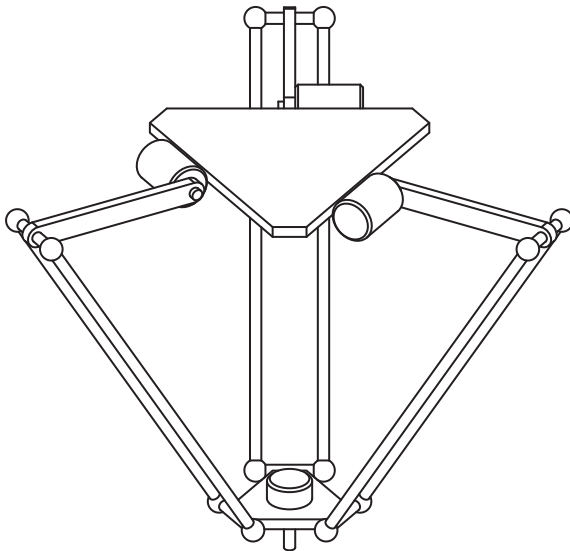
<b>2-1</b>	<b>Controllable Robot Types</b>	<b>2-2</b>
<b>2-2</b>	<b>Coordinate System</b>	<b>2-5</b>
2-2-1	Overview of Coordinate Systems	2-5
2-2-2	Coordinate Systems	2-19
<b>2-3</b>	<b>Robot Functions</b>	<b>2-20</b>
2-3-1	Kinematics Setting	2-20
2-3-2	Workspace Check	2-30
2-3-3	User Coordinate System (UCS)	2-38
2-3-4	Monitoring	2-39
2-3-5	Time-specified Absolute Positioning	2-39
2-3-6	Conveyor Synchronization	2-40
2-3-7	Inverse Kinematics	2-41
2-3-8	Robot Tool	2-41
2-3-9	Robot Jogging	2-44
2-3-10	Checking Maximum Interpolation Velocity and Maximum Interpolation Acceleration/Deceleration	2-45
2-3-11	Multi-execution of Instructions with Buffered Mode	2-47
2-3-12	Multi-execution of Instructions with Blending Mode	2-47
2-3-13	Trajectory Types for Time-specified Motion	2-48
2-3-14	Joint Interpolated Point-to-Point Movement	2-50
2-3-15	Management of Robots with Multiple Arm Configurations	2-51

## 2-1 Controllable Robot Types

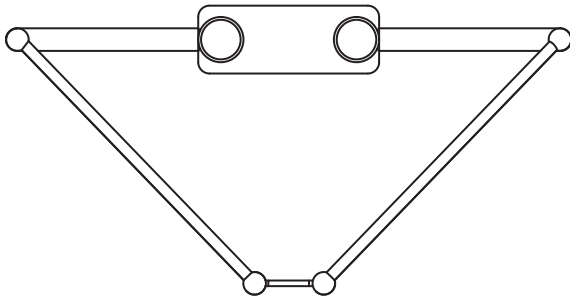
---

The robot types connectable to the NJ Robotics CPU Unit are Delta3, Delta3R, Delta2, Cartesian 3D, Cartesian 3D Gantry, Cartesian 2D, Cartesian 2D Gantry, H-Bot, SCARA RRP, SCARA RRP+R, SCARA PRR, SCARA PRR+R and Expansion1.

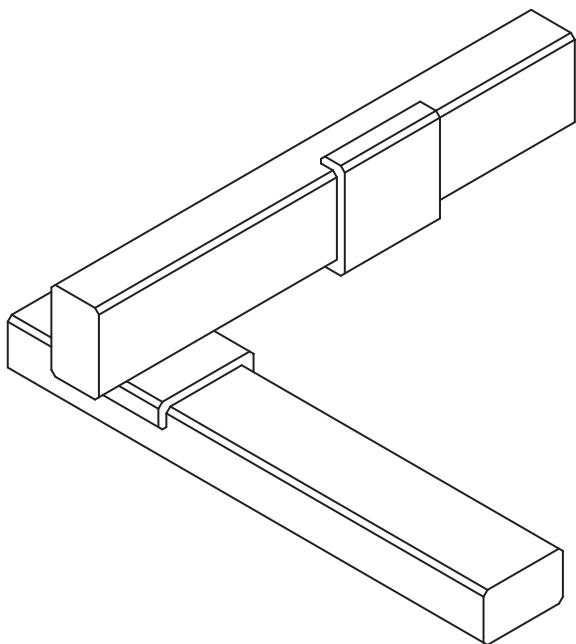
The following figure is an example of Delta3.



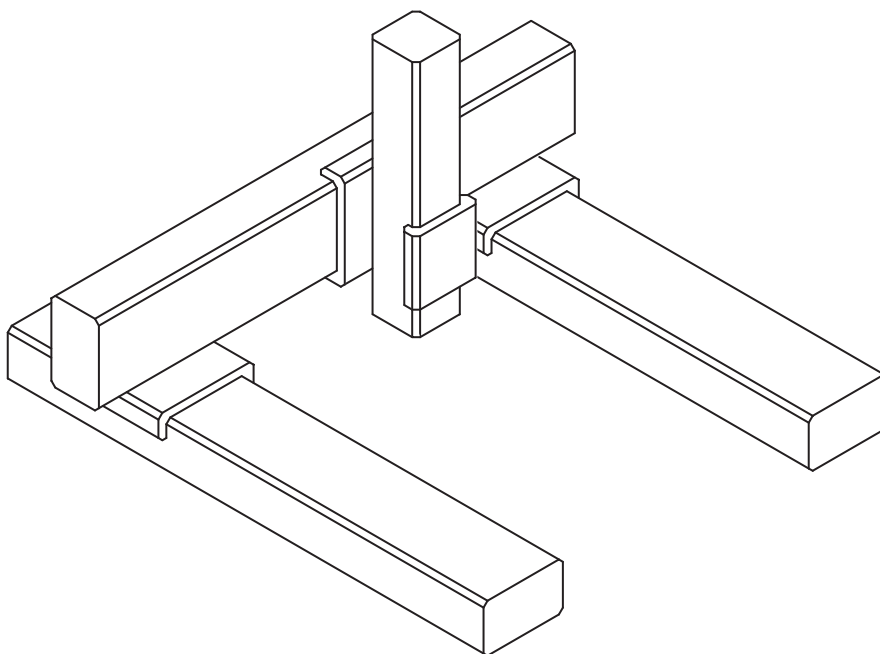
The following figure is an example of Delta2.



The following figure is an example of Cartesian 2D.

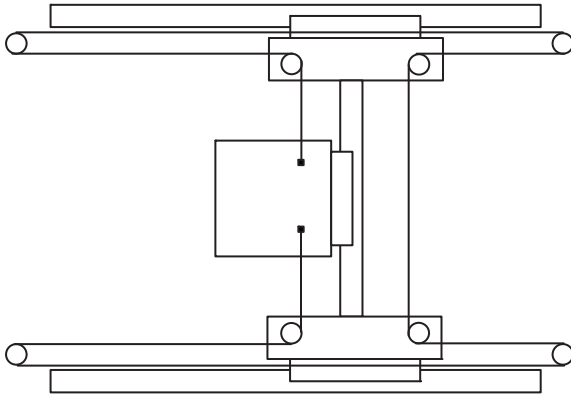


The following figure is an example of Cartesian 3D Gantry.

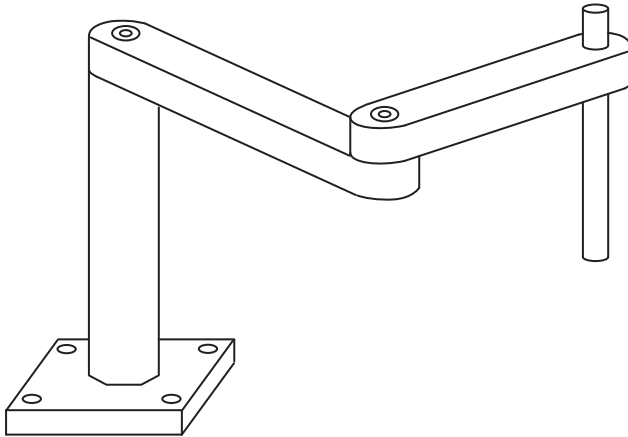


## 2 Features of NJ Robotics CPU Unit

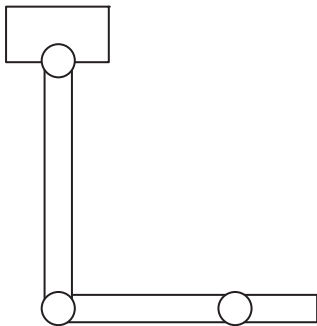
The following figure is an example of H-Bot.



The following image is an example of SCARA.



The following figure is an example of Expansion1.



## 2-2 Coordinate System

This section describes the coordinate systems used by an NJ Robotics CPU Unit.

### 2-2-1 Overview of Coordinate Systems

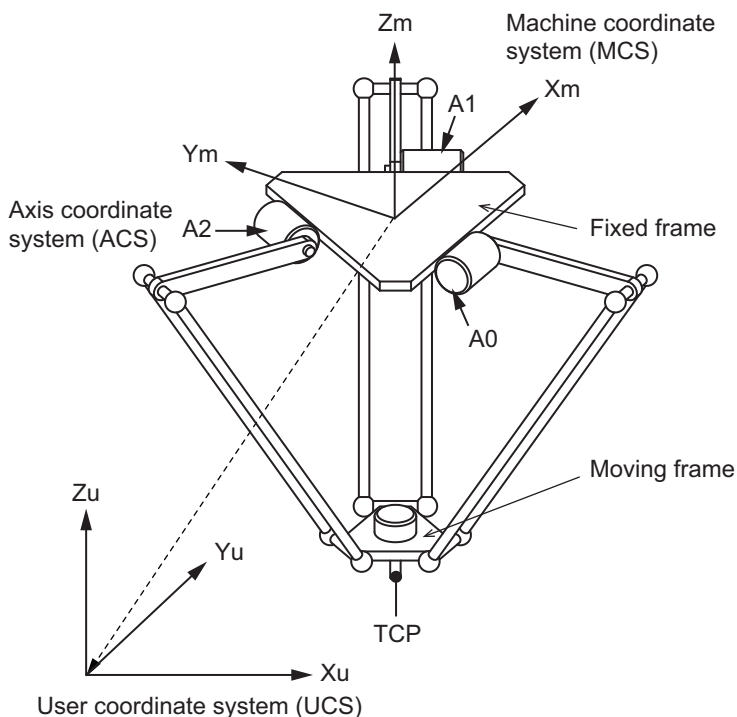
This section describes the types of coordinate systems, the directions of axis coordinate systems (ACS), and the positional relationship between the axis coordinate systems (ACS) and the machine coordinate system (MCS).

#### Types of Coordinate Systems

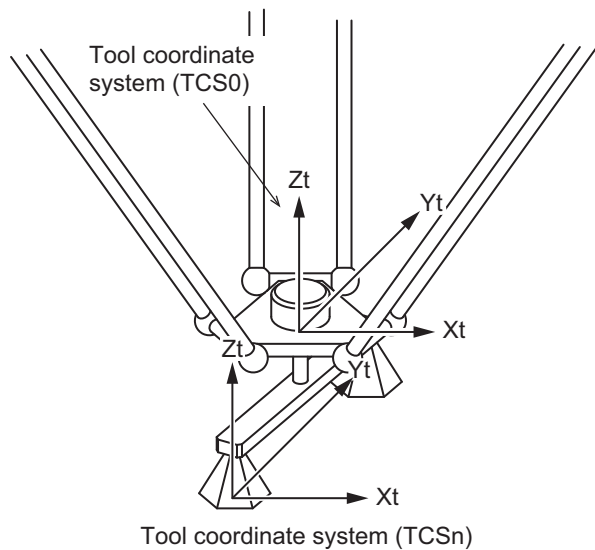
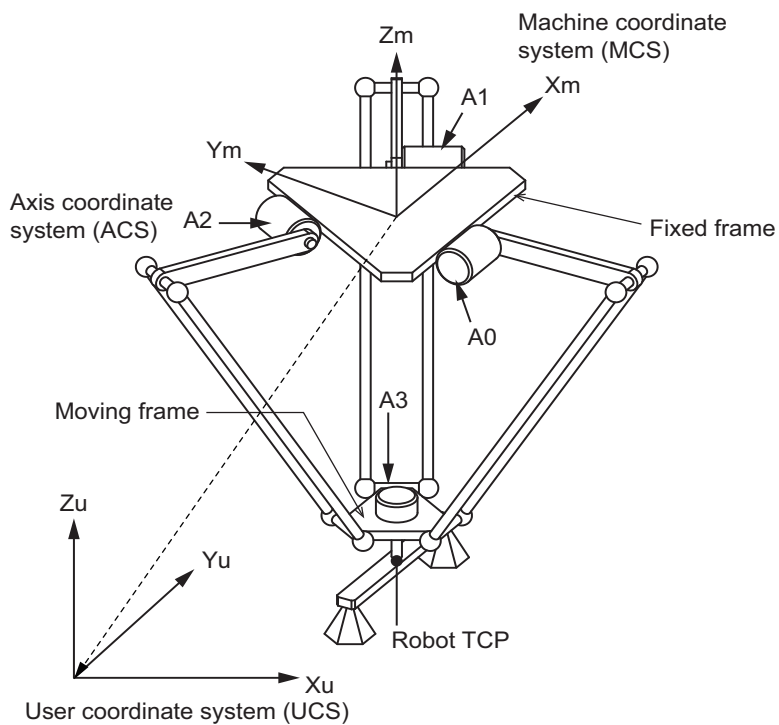
An NJ Robotics CPU Unit uses the following coordinate systems to control the robots.

Coordinate System	Description	Reference
Axis coordinate system (ACS)	The coordinate systems shown with A0, A1 and A2 in the following figure	P. 2-19
Machine coordinate system (MCS)	The coordinate system shown with X <sub>m</sub> , Y <sub>m</sub> and Z <sub>m</sub> in the following figure	P. 2-19
User coordinate system (UCS)	The coordinate system shown with X <sub>u</sub> , Y <sub>u</sub> and Z <sub>u</sub> in the following figure	P. 2-19
Tool coordinate system (TCS)	The coordinate system shown with X <sub>t</sub> , Y <sub>t</sub> and Z <sub>t</sub> in the following figure	P. 2-19

#### ● Coordinate Systems for Delta3

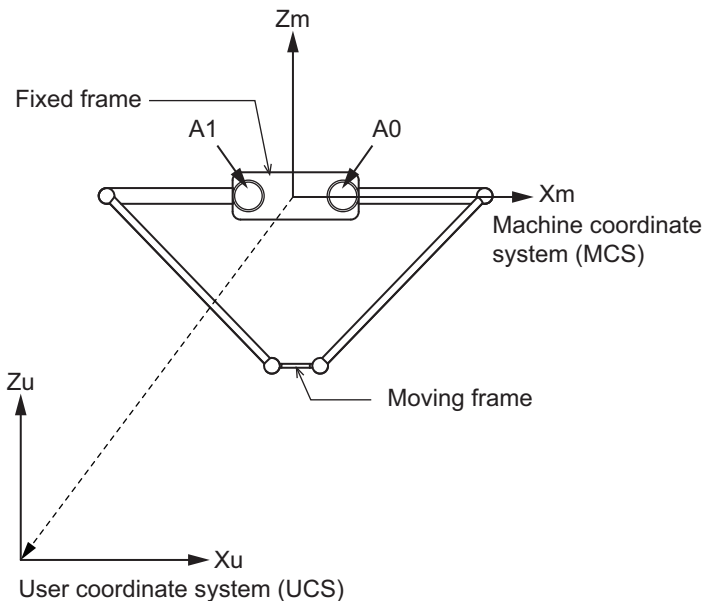


● Coordinate Systems for Delta3R

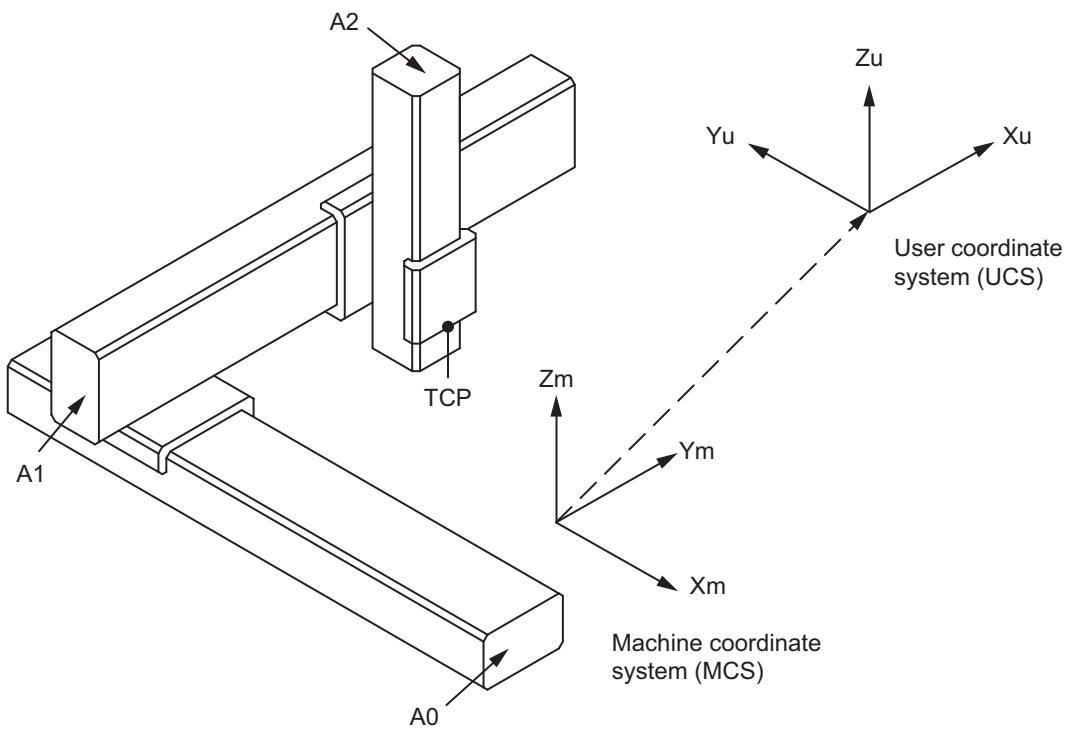


● **Coordinate Systems for Delta2**

The figure shows Delta2 viewed from  $Y_m$  direction.

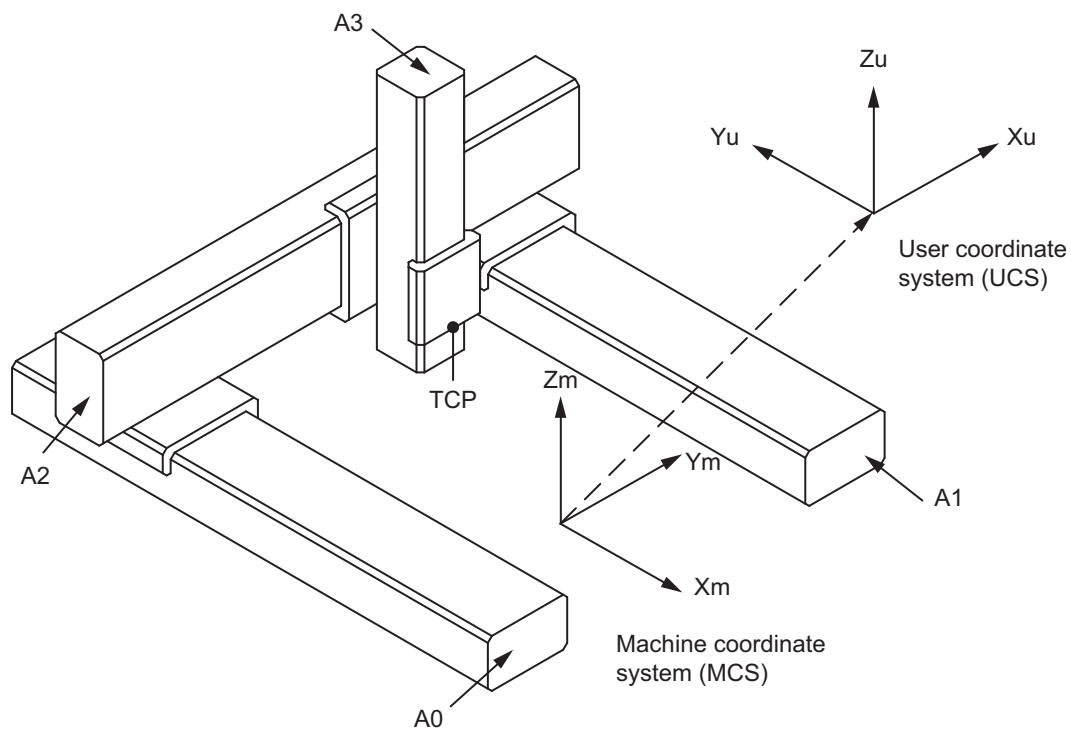


● **Coordinate Systems for Cartesian 3D**

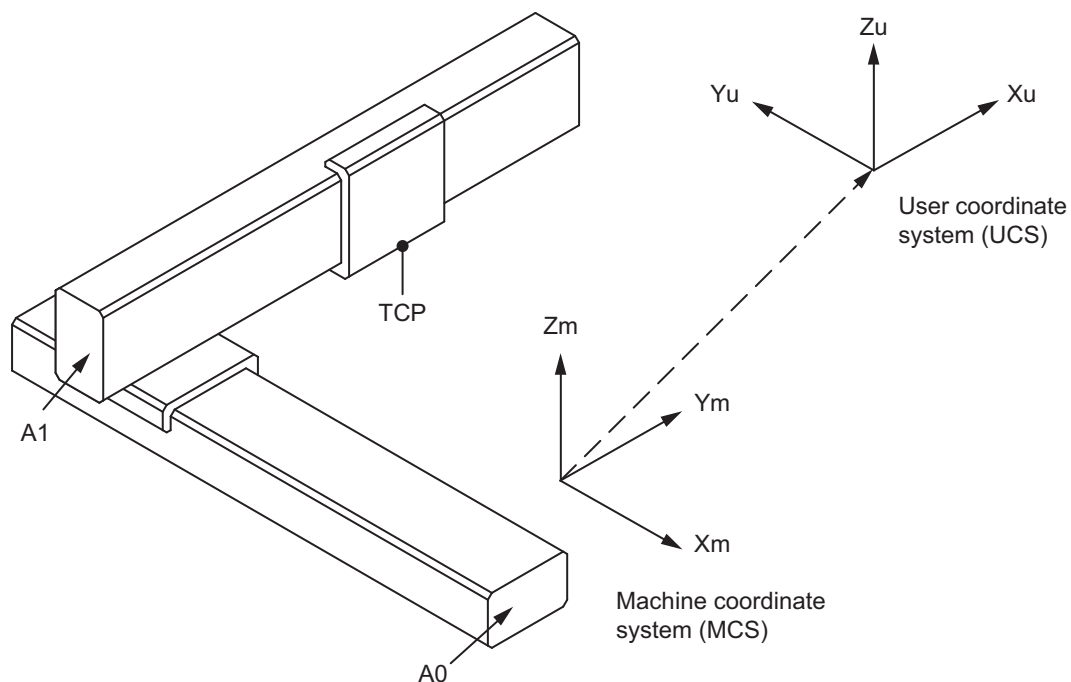


## 2 Features of NJ Robotics CPU Unit

### ● Coordinate Systems for Cartesian 3D Gantry



### ● Coordinate Systems for Cartesian 2D

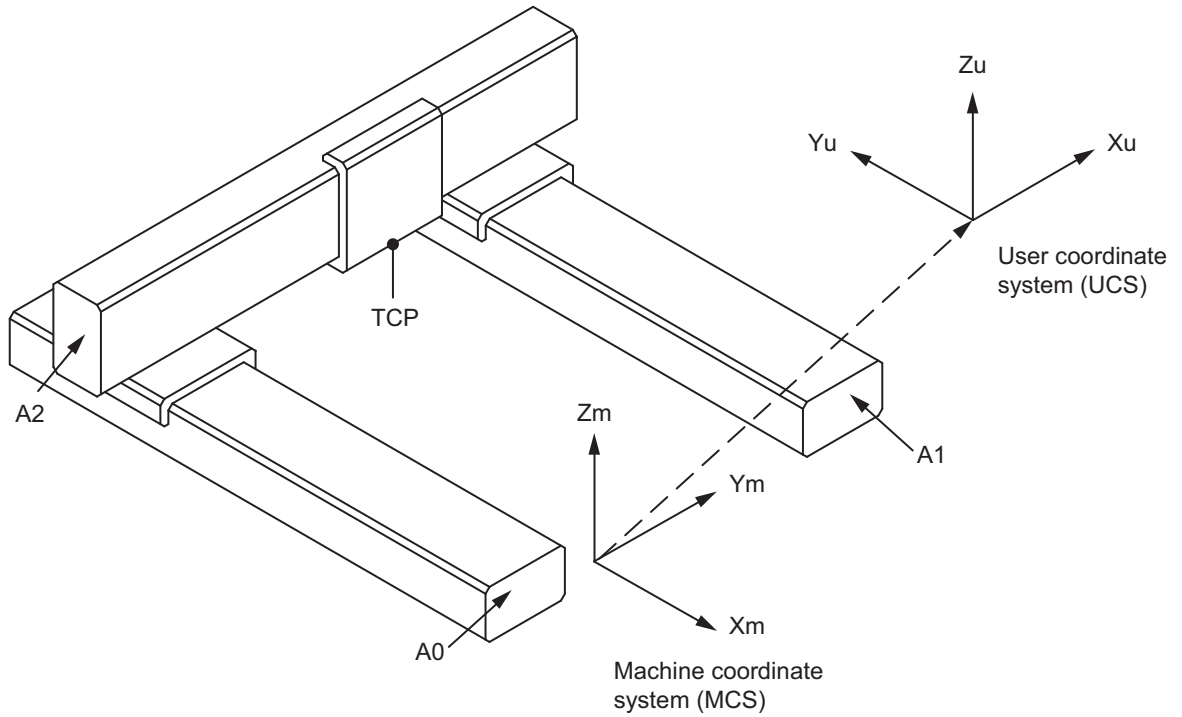


The Cartesian 2D kinematics can be arranged in the three coordinate planes:

- Cartesian XY kinematics
- Cartesian XZ kinematics
- Cartesian YZ kinematics

The figure is an example of Cartesian XY kinematics.

● **Coordinate Systems for Cartesian 2D Gantry**



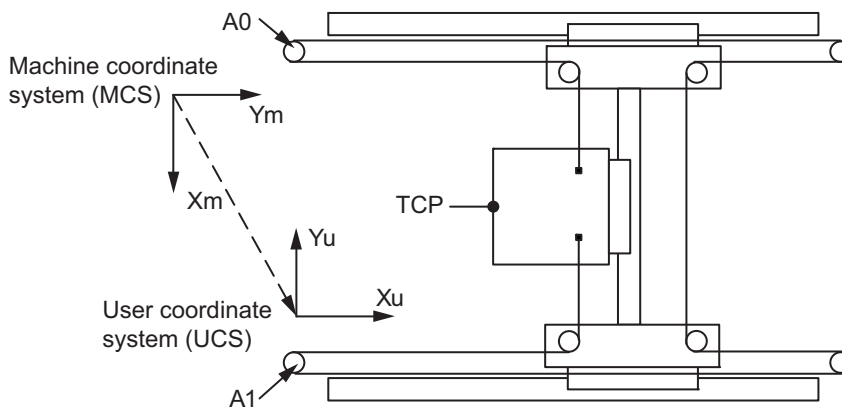
The Cartesian 2D Gantry kinematics can be arranged only in XY coordinate plane.

● **Coordinate Systems for H-Bot**

The H-Bot kinematics has 4 sub types.

Direction of TCP movement depends on direction of the axes A0 and A1, and depends on H-Bot type.

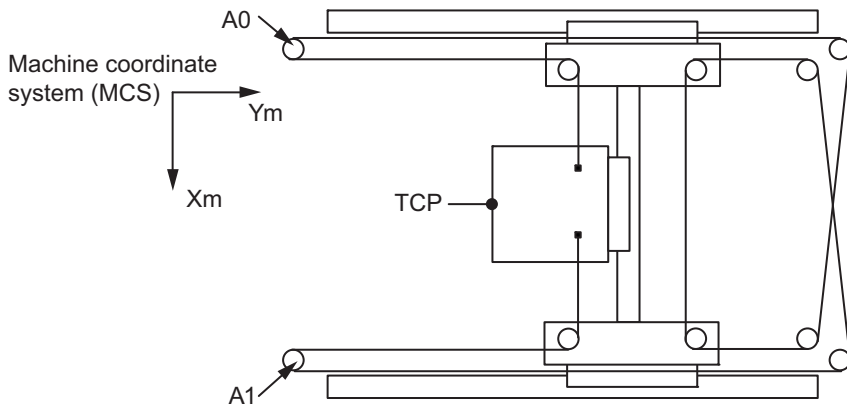
- H-Bot type 1



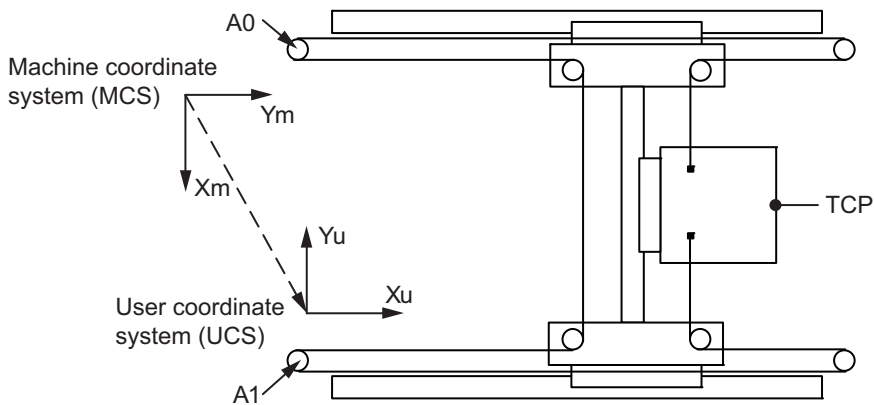
A0 movement direction	A1 movement direction	X movement direction	Y movement direction
Positive	Positive	Positive	No movement
Positive	Negative	No movement	Positive
Negative	Positive	No movement	Negative
Negative	Negative	Negative	No movement

## 2 Features of NJ Robotics CPU Unit

H-Bot1 kinematics can be used also with CoreXY robot.

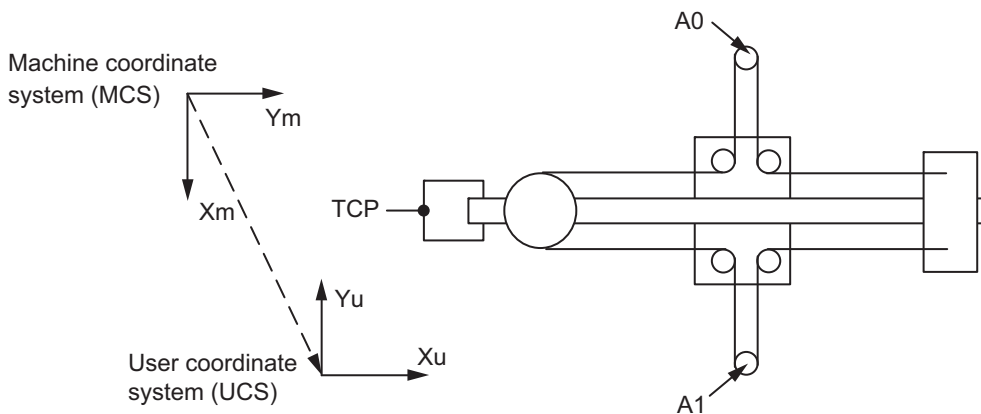


- H-Bot type 2



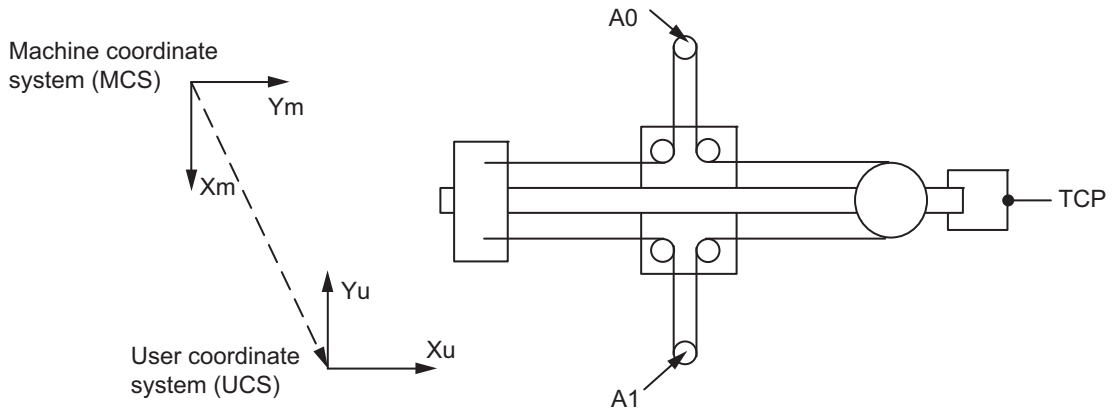
A0 movement direction	A1 movement direction	X movement direction	Y movement direction
Positive	Positive	Negative	No movement
Positive	Negative	No movement	Positive
Negative	Positive	No movement	Negative
Negative	Negative	Positive	No movement

- T-Bot type 1



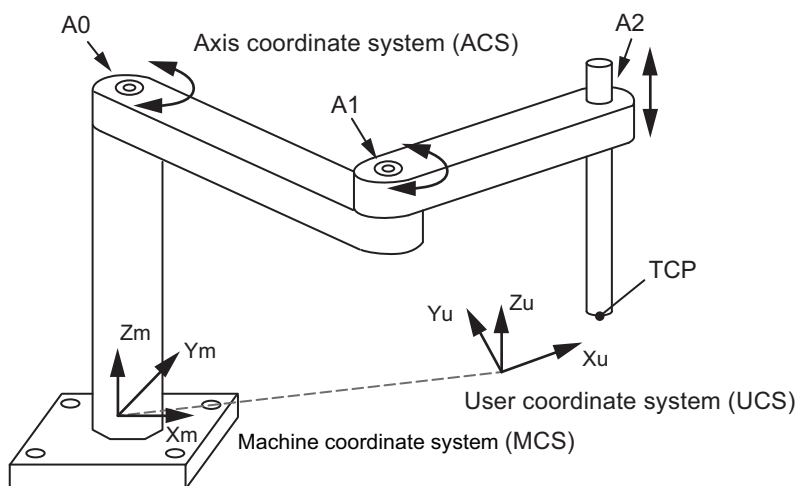
A0 movement direction	A1 movement direction	X movement direction	Y movement direction
Positive	Positive	Negative	No movement
Positive	Negative	No movement	Negative
Negative	Positive	No movement	Positive
Negative	Negative	Positive	No movement

- T-Bot type 2



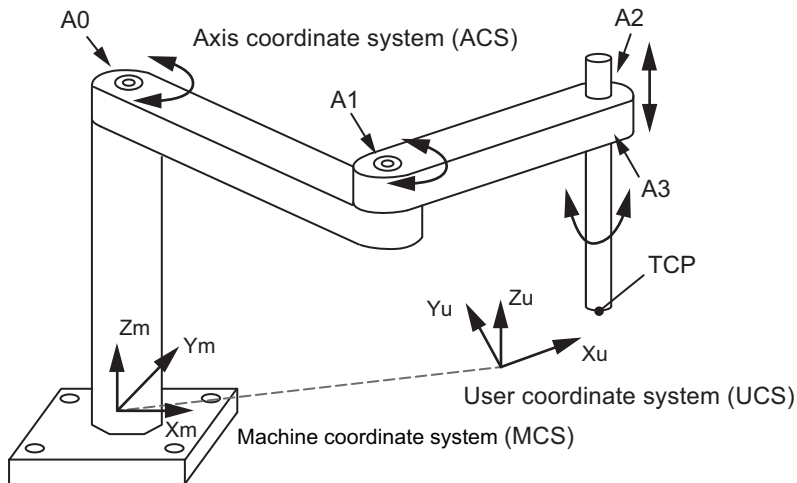
A0 movement direction	A1 movement direction	X movement direction	Y movement direction
Positive	Positive	Positive	No movement
Positive	Negative	No movement	Negative
Negative	Positive	No movement	Positive
Negative	Negative	Negative	No movement

● Coordinate Systems for SCARA RRP

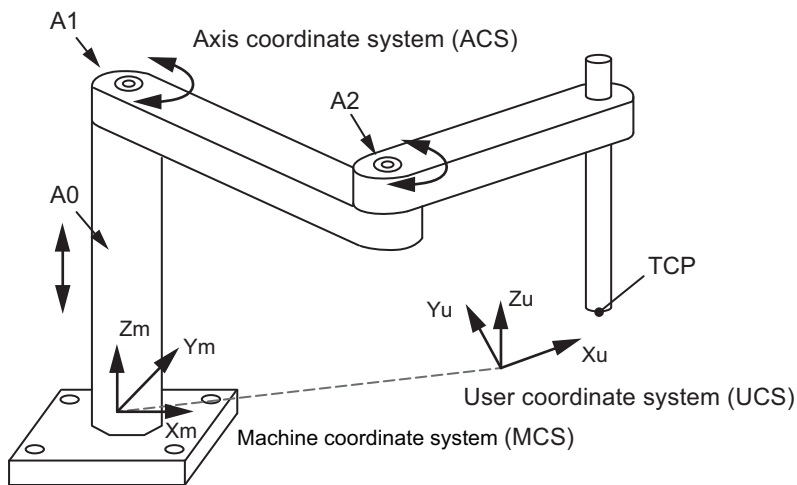


## 2 Features of NJ Robotics CPU Unit

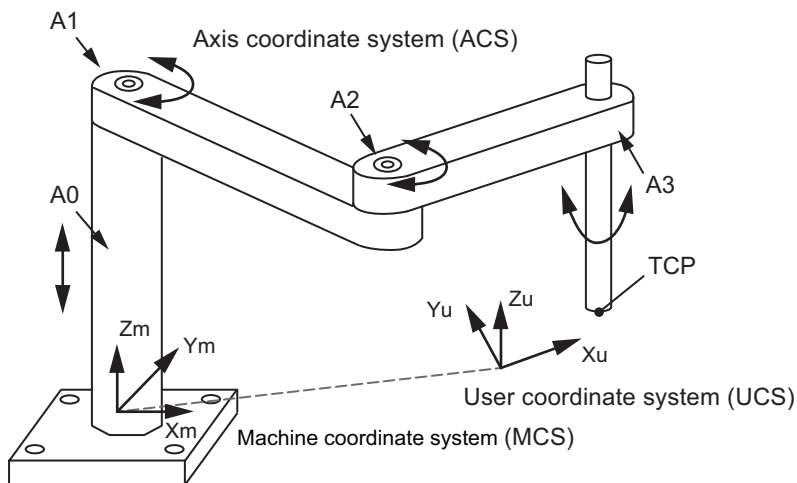
### ● Coordinate Systems for SCARA RRP+R



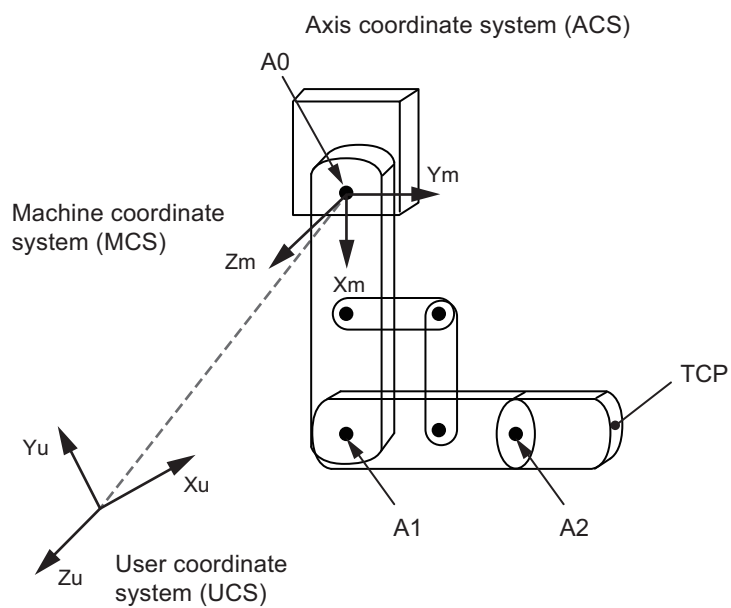
### ● Coordinate Systems for SCARA PRR



### ● Coordinate Systems for SCARA PRR+R



## ● Coordinate Systems for Expansion1



## Directions of Axis Coordinate System (ACS)

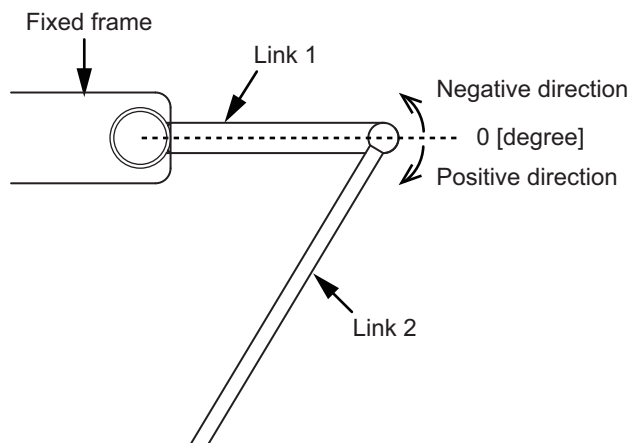
### ● Directions of ACS for Delta Robots

This coordinate system is applied to three axes, A0 to A2, of Delta3 and Delta3R, and two axes, A0 and A1, of Delta2.

"0 Degree" refers to an axis position at which the fixed frame and link 1 form a straight line. The direction below the 0 degree line is called Positive Direction, and the direction above the 0 degree line is called Negative Direction. The settings of axes (Servo Drives) must conform with this axis coordinate system (ACS).

You must wire so that an axis operates in a downward direction if a positive command velocity is given, and the axis operates in an upward direction if a negative command velocity is given.

Wire a Servomotor so that the motor rotates in the directions shown in the following figure.



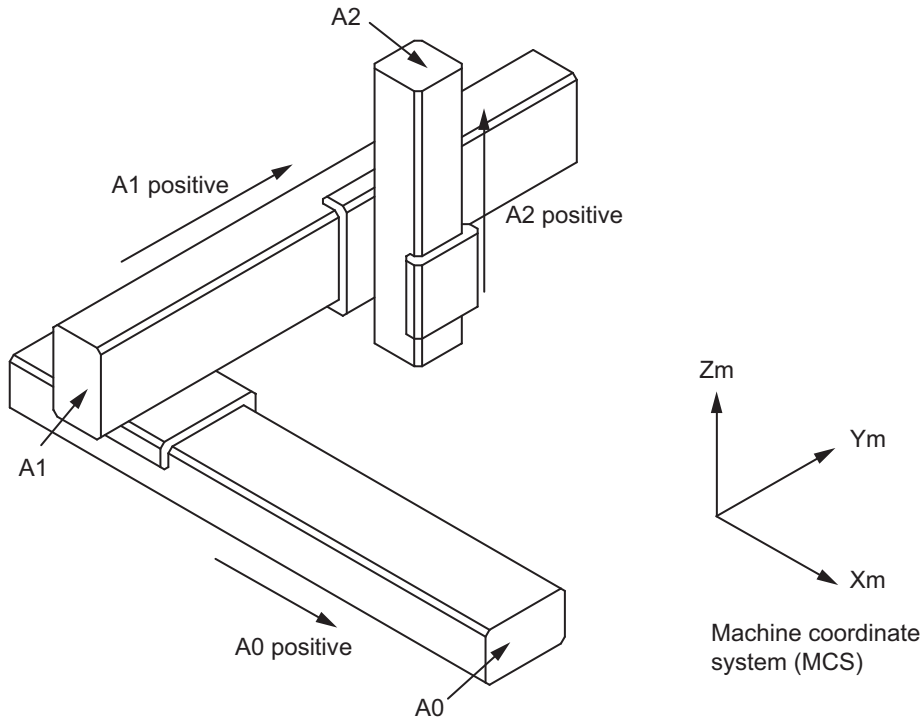
### ● Directions of ACS for Cartesian Robots and Gantry Robots

In these robots, each axis is corresponding to one of coordinate X, Y or Z. See the following table.

Axes	Kinematics type					
	Cartesian 3D	Cartesian 3D Gantry	Cartesian 2D			Cartesian 2D Gantry
			Plane XY	Plane XZ	Plane YZ	
A0	X	X	X	X	Y	X
A1	Y	X	Y	Z	Z	X
A2	Z	Y	Not used			Y
A3	Not used	Z	Not used			Not used

Then, positive direction of each axis must follow the positive direction of the corresponding coordinate.

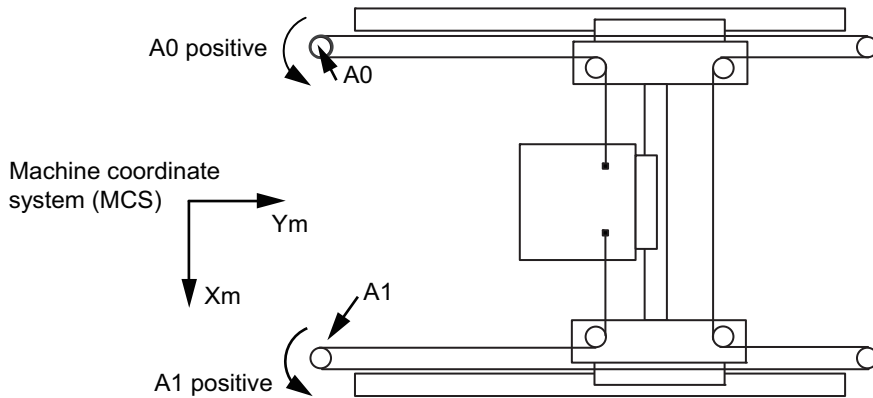
The following figure is an example of Cartesian 3D.



● **Directions of ACS for H-Bots and T-Bots**

In these robots, positive direction of each axis is counter-clockwise in plane XY.

The following figure is an example of H-Bot type 1.



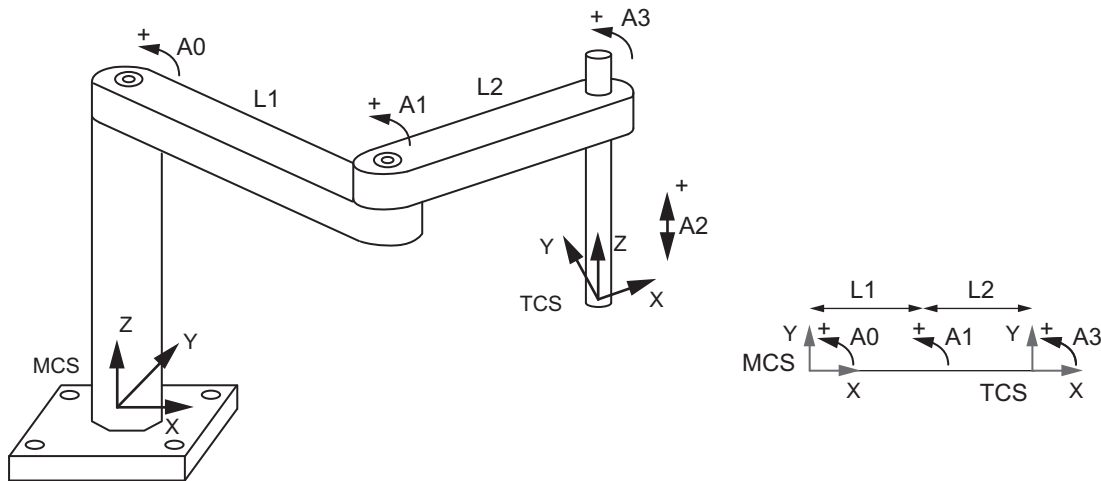
## 2 Features of NJ Robotics CPU Unit

### ● Directions of ACS for SCARA

For a SCARA robot, the position when link L1 is aligned with link L2 (L1 aligned with  $X_m$ ) is defined as "0 degree". Positive direction of each axis is counter-clockwise in plane XY.

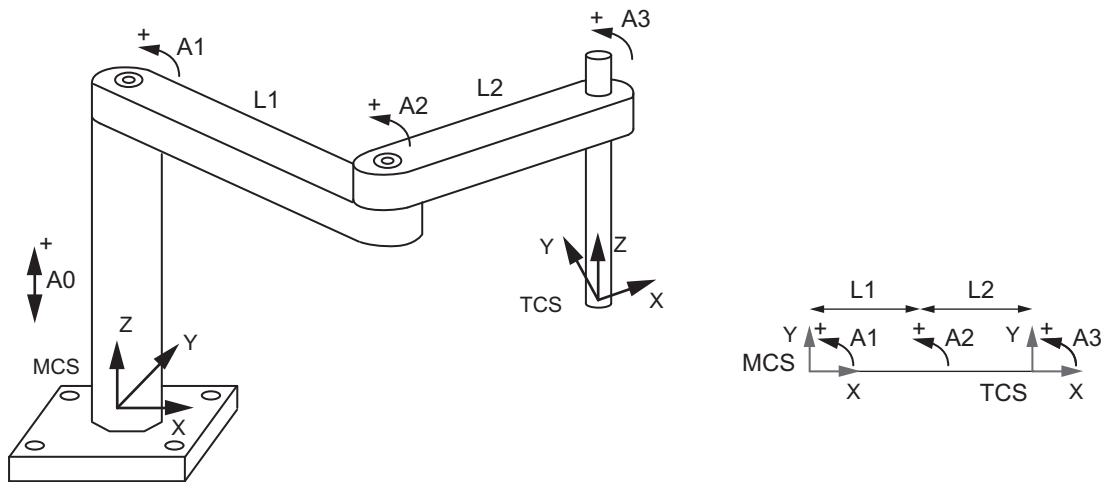
Directions in SCARA RRP+R

RRP type keeps the same directions, but it is without A3.



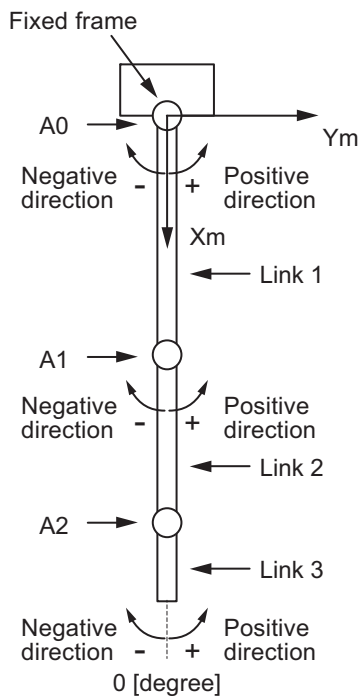
Directions in SCARA PRR+R

PRR type keeps the same directions, but it is without A3.



### ● Directions of ACS for Expansion1

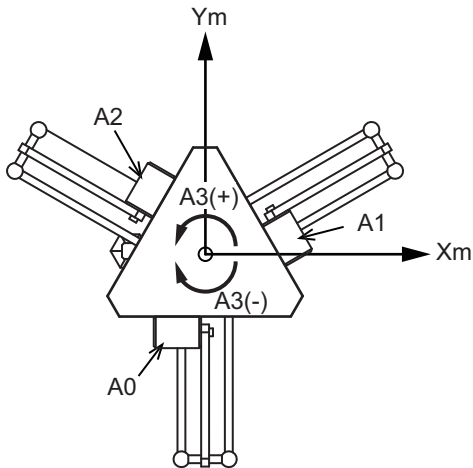
For the Expansion1 robot, the position when link  $i$  is aligned with link  $i-1$  (in case of first axis link 1 aligned with  $X_m$ ) is defined as "0 degree" for the axis  $A_{i-1}$ .



## Positional Relationship between Axis Coordinate System (ACS) and Machine Coordinate System (MCS)

The following figure shows the directions of the axis coordinate system (ACS) of each axis (A0 to A2 and A3) and the directions of the machine coordinate system (MCS).

The figure shows the fixed frame of a Delta3 robot viewed from Zm direction.



### Precautions for Safe Use

Set a correct relationship between the direction of commands given by the Controller and the motor rotation direction of the Servo Drive.

Otherwise, the robot may operate unexpectedly.



### Precautions for Correct Use

The positional relationship between axes, and the directions of their coordinate systems are fixed and cannot be changed.

## 2-2-2 Coordinate Systems

This section describes the axis coordinate system (ACS), the machine coordinate system (MCS), the user coordinate system (UCS), and the tool coordinate system (TCS).

### Axis Coordinate System (ACS)

The axis coordinate system (ACS) is a rotation or linear motion coordinate system specified for each axis.

This system is also called the Joint Coordinate System or the Link Coordinate System.

Each axis has one coordinate system.

### Machine Coordinate System (MCS)

The machine coordinate system (MCS) is a Cartesian coordinate system specified for each robot.

Use the position of the robot TCP viewed from the origin of the machine coordinate system to set the target position for robot operation.

Each robot (one axes group) has one machine coordinate system.

### User Coordinate System (UCS)

The user coordinate system is a Cartesian coordinate system defined by the user.

Use the rotation and the position relative to the origin of the machine coordinate system to set the origin position (Tx, Ty, Tz, Rx, Ry, Rz) of the user coordinate systems. The rotation is also called Pose.

You can set up to 16 user coordinate systems for each robot.

### Tool Coordinate System (TCS)

The tool coordinate system is a Cartesian coordinate system set by the user according to the tool to be used.

The reference position of the tool coordinate system is the position of the tool TCP. You can set up to 16 tool coordinate systems for each robot, and select from the total of 17 systems that include the default tool coordinate system.

## 2-3 Robot Functions

This section gives information related to the robot functions that include the kinematics, workspace and coordinate systems.

### 2-3-1 Kinematics Setting

To operate a robot, you need to set the kinematics type and the transform parameters for the axes group.

This section describes the settings of the kinematics type and kinematics parameters.

To set an axes group, use the `MC_SetKinTransform` (Set Kinematics Transformation) instruction.

Refer to `MC_SetKinTransform` on page 4-6 for details on the `MC_SetKinTransform` (Set Kinematics Transformation) instruction.

### Setting Kinematics Type

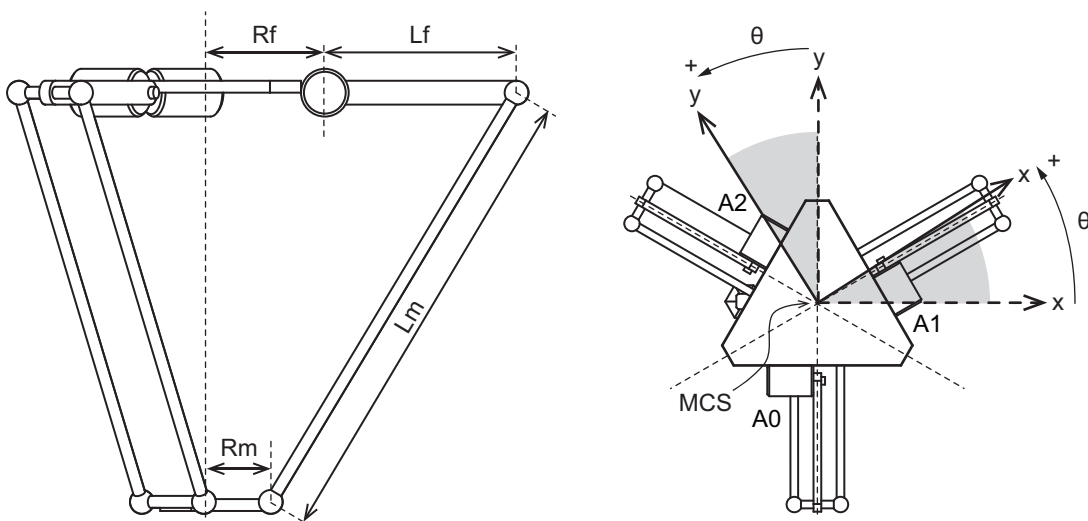
Select the kinematics type from the following options: Delta3, Delta3R, Delta2, Cartesian 3D, Cartesian 3D Gantry, Cartesian 2D, Cartesian 2D Gantry, H-Bot, SCARA RRP, SCARA RRP+R, SCARA PRR, SCARA PRR+R and Expansion1.

### Setting Kinematics Parameters

In this section, the settings for each kinematics are described. The restriction on kinematics is also explained here.

#### ● Setting for Delta3 and Delta3R

Set the following parameters for Delta3 and Delta3R.



Parameter	Description
Rf	The distance between the center of fixed frame and the axis motor Radius (unit: mm)
Rm	The distance between the center of moving frame and the joint of link 2 Radius (unit: mm)
Lf	The length of link 1 (unit: mm)
Lm	The length of link 2 (unit: mm)
$\theta$	The origin position for rotation around Z axis of the machine coordinate system. This parameter is used for axis adjustment. (unit: degree)

- The kinematics parameters for Delta3R are the same as Delta3.
- A wrist axis must be controlled as a single axis in Delta3 kinematics.  
In Delta3R kinematics, the robot instructions control the wrist axis as a part of the axes group (4 axes).
- $\theta$  parameter (Angle Offset) enables to rotate (around Z) default MCS of the robot to be aligned with the desired Coordinate System to be used when defining robot kinematics.



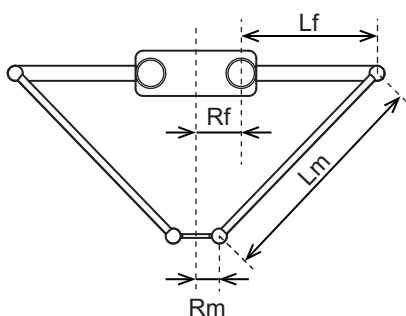
#### Precautions for Correct Use

If you select a robot tool in Delta3R kinematics, the wrist axis is controlled as a part of the axes group. This means that the wrist axis may operate unexpectedly.

The  $0^\circ$  for the wrist axis is located in the direction of X axis of the machine coordinate system (MCS). Note that if you rotate the machine coordinate system around Z axis with the parameter  $\theta$ , the  $0^\circ$  position of the wrist axis will also be rotated.

#### ● Setting for Delta2

Set the following parameters for Delta2.



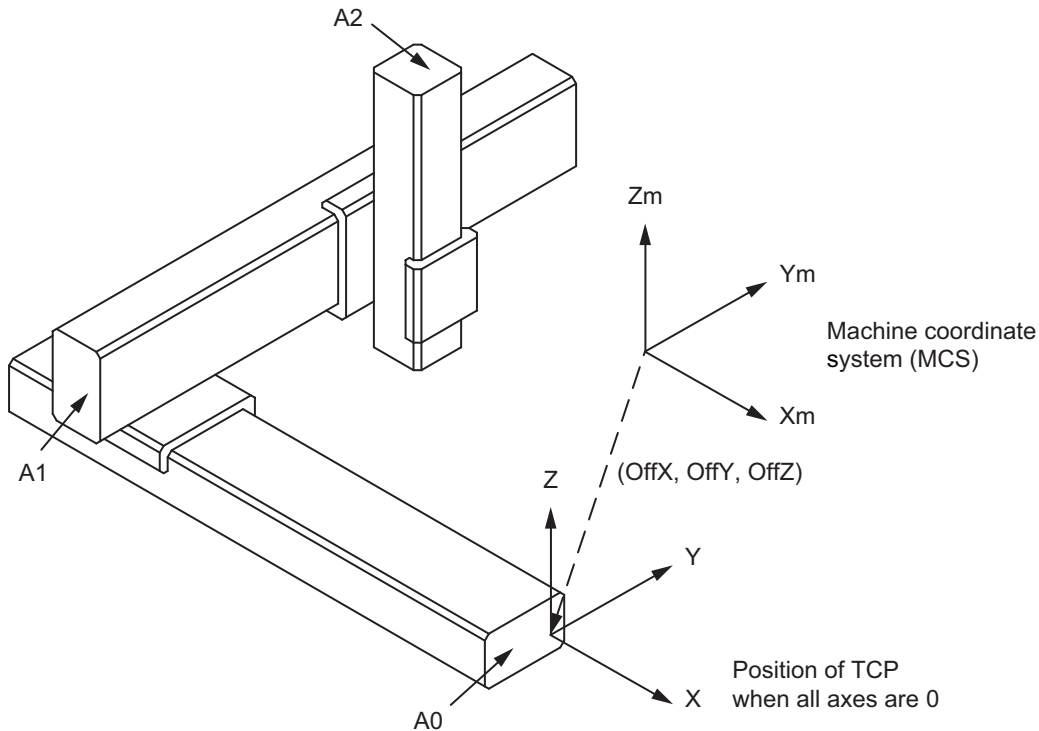
Parameter	Description
Rf	The distance between the center of fixed frame and the axis motor Radius (unit: mm)
Rm	The distance between the center of moving frame and the joint of link 2 Radius (unit: mm)
Lf	The length of link 1 (unit: mm)
Lm	The length of link 2 (unit: mm)

- You cannot use the wrist axis in Delta2.

## 2 Features of NJ Robotics CPU Unit

### ● Setting for Cartesian 3D

Set the following parameters for Cartesian 3D.



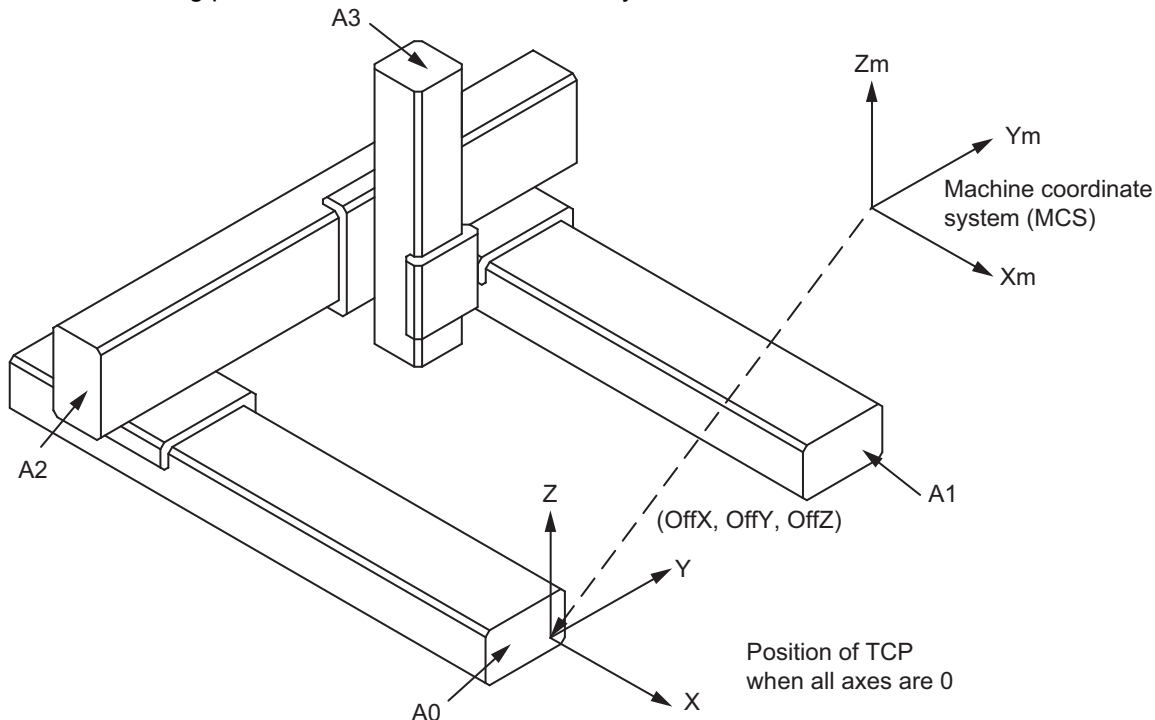
Parameter	Description
OffX	The distance on X between MCS origin and TCP when all axes are 0 (unit: mm)
OffY	The distance on Y between MCS origin and TCP when all axes are 0 (unit: mm)
OffZ	The distance on Z between MCS origin and TCP when all axes are 0 (unit: mm)

- This kinematics does not mean `_eMC_GROUP_TYPE#_mcXYZ` for `_MC_GRP[0-31].Kinematics.GrpType`.

When this kinematics is set, some functionalities are limited. Refer to *4-1 Overview of Robot Instructions* on page 4-2.

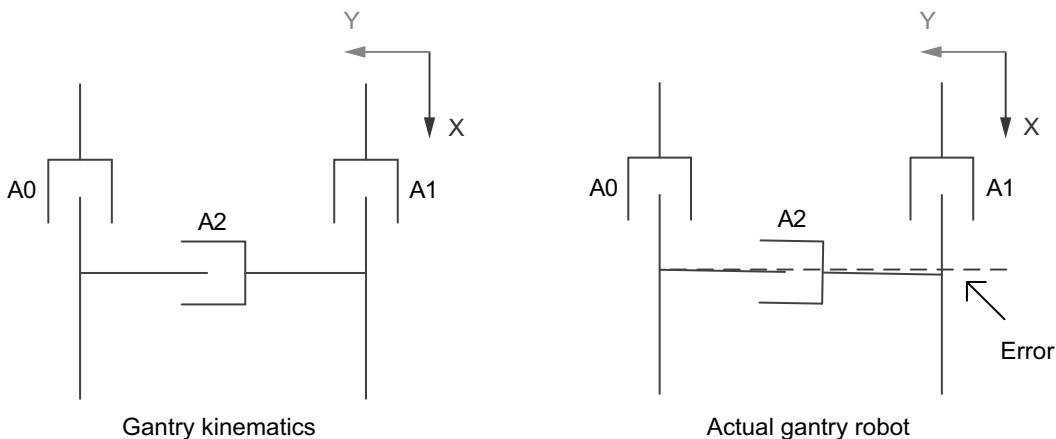
● **Setting for Cartesian 3D Gantry**

Set the following parameters for Cartesian 3D Gantry.



Parameter	Description
OffX	The distance on X between MCS origin and TCP when all axes are 0 (unit: mm)
OffY	The distance on Y between MCS origin and TCP when all axes are 0 (unit: mm)
OffZ	The distance on Z between MCS origin and TCP when all axes are 0 (unit: mm)
Err1	The maximum acceptable error of commanded positions between axes A0 and A1 (absolute value) (unit: mm)
Err2	The maximum acceptable error of actual positions between axes A0 and A1 (absolute value) (unit: mm)

Gantry kinematics is based on the assumption that two linked axes (A0 and A1) are perfectly aligned and equal. But in the actual robot, there is always a difference in positions between the two linked axes of the gantry, A0 and A1.



## 2 Features of NJ Robotics CPU Unit

When there is this difference (between axes A0 and A1) and a robotics movement is executed, there could be a position jump at the beginning of the movement because same position is output to axes A0 and A1. This might cause a mechanical shock and damage the robot when there is a high mis-alignment between the axes A0 and A1.

The system implements two checks in order to prevent possible problems:

- a) Commanded axes position error (A0 and A1) is checked against parameter **Err1** before executing robotics movement commands. In the case commanded position error  $|A0-A1|$  is higher than **Err1**, the error “570F Cannot Calculate Kinematics” occurs.
- b) Actual axes position error (A0 and A1) is checked against parameter **Err2**. When actual position error  $|A0-A1|$  is higher than **Err2**, Cartesian position is not valid. In this case *Valid* output variable of the MC\_GroupMon instruction changes to FALSE.

If the error 570F occurs and the commanded axes position error is higher than **Err1**, user can compensate the axes position error using a motion movement instruction to align the position of A0 and A1 (e.g. MC\_GroupSyncMoveAbsolute).



### Precautions for Correct Use

---

- When using Cartesian 3D Gantry Kinematics, you have to configure parameters **Err1** and **Err2**. Those default values are 0, but cannot be used.
  - Program to stop the robot if MC\_GroupMon returns FALSE in *Valid* output because of actual position error  $|A0-A1|$ .
- 

In case our gantry robot mechanics cannot work in this simple solution, an alternative solution is used to command the gantry axes (A0 and A1) by using a gearing of the two axes with a torque feedback of A1 to ensure same position of the axes and equal distribution of torque.

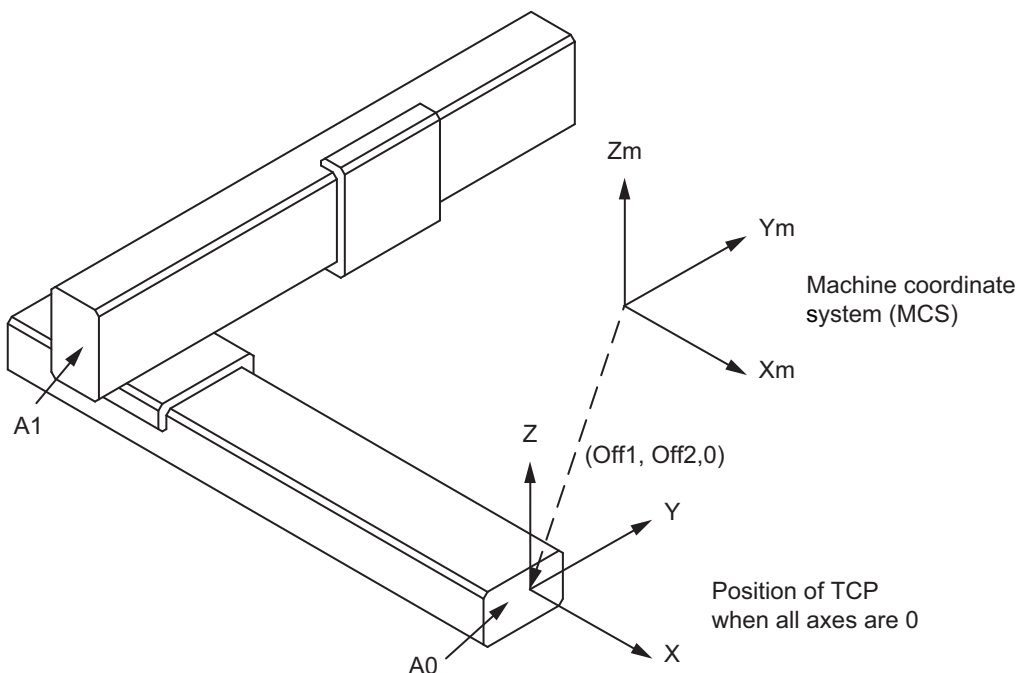
For example, user could implement this solution by:

- Using the Cartesian 3D kinematics (`_mcCartesian3DType1`) with the axis A1 excluded from the axes group (A0 as X coordinate, A2 as Y coordinate and A3 as Z coordinate). Applying its own compensation algorithm to control A1 based on A0 commanded position.
- Using the Cartesian 3D kinematics (`_mcCartesian3DType1`) with two axes A0, A1 excluded from the axes group (A4 as X virtual coordinate, A2 as Y coordinate and A3 as Z coordinate). Applying its own compensation algorithm to control A0 and A1 based on A4 commanded position.

When monitoring Cartesian position of an AxesGroup with Cartesian 3D Gantry Kinematics set by using the MC\_GroupMon instruction, it is calculated assuming that A0 is equal to A1 in terms of commanded and actual position.

## ● Setting for Cartesian 2D

Set the following parameters for Cartesian 2D.



Parameter	Description
Off1	The distance of the first coordinate between MCS origin and TCP when all axes are 0 (unit: mm)
Off2	The distance of the second coordinate between MCS origin and TCP when all axes are 0 (unit: mm)
Plane	Index to defined planes: 0: plane XY – the first coordinate is X and the second is Y 1: plane XZ – the first coordinate is X and the second is Z 2: plane YZ – the first coordinate is Y and the second is Z

- This kinematics does not mean `_eMC_GROUP_TYPE#_mcXY` for `_MC_GRP[0-31].Kinematics.GrpType`.

When this kinematics is set, some functionalities are limited. Refer to *4-1 Overview of Robot Instructions* on page 4-2.

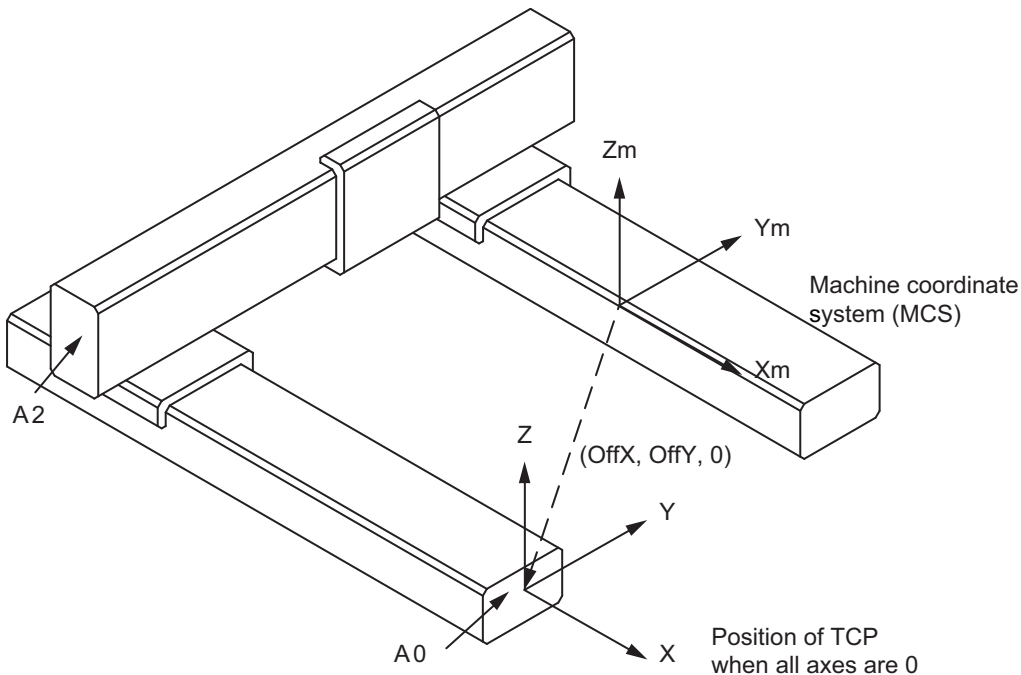


### Precautions for Safe Use

When using Cartesian 2D kinematics, configure Plane correctly. Otherwise, a robot may operate unexpectedly.

## ● Setting for Cartesian 2D Gantry

Set the following parameters for Cartesian 2D Gantry.



Parameter	Description
OffX	The distance on X between MCS origin and TCP when all axes are 0 (unit: mm)
OffY	The distance on Y between MCS origin and TCP when all axes are 0 (unit: mm)
Err1	The maximum acceptable error of commanded positions between axes A0 and A1 (absolute value) (unit: mm)
Err2	The maximum acceptable error of actual positions between axes A0 and A1 (absolute value) (unit: mm)

Gantry kinematics is based on the assumption that two linked axes (A0 and A1) are perfectly aligned and equal. But in the actual robot, there is always a difference in positions between the two linked axes of the gantry, A0 and A1.

Refer to *Setting for Cartesian 3D Gantry* on page 2-23 to prevent possible problems.

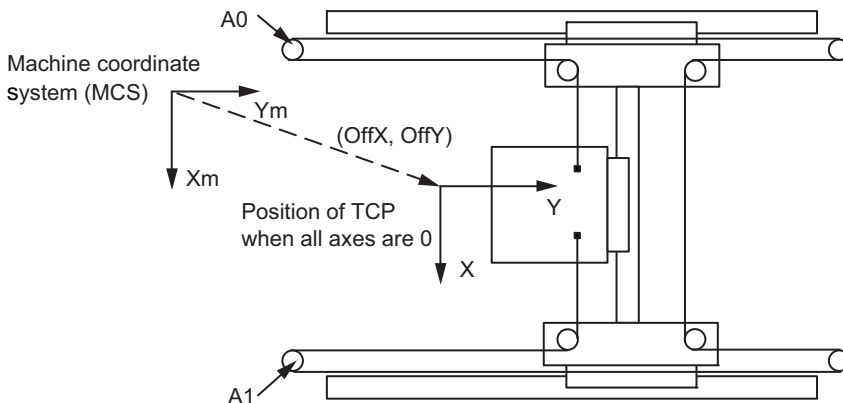


### Precautions for Correct Use

- When using Cartesian 2D Gantry Kinematics, you have to configure parameters **Err1** and **Err2**. Those default values are 0, but cannot be used.
- Program to stop the robot if MC\_GroupMon returns FALSE in *Valid* output because of actual position error  $|A0-A1|$ .

● **Setting for H-Bot**

Set the following parameters for H-Bot.



Parameter	Description
OffX	The distance on X between MCS origin and TCP when all axes are 0 (unit: mm)
OffY	The distance on Y between MCS origin and TCP when all axes are 0 (unit: mm)
Type	Type of H-Bot: 0: H-Bot type 1 1: H-Bot type 2 2: T-Bot type 1 3: T-Bot type 2

● **Restriction on Kinematics**

The Delta3, Delta3R and Delta2 kinematics must satisfy all the following conditions for operation with an NJ Robotics CPU Unit.

Item	Restriction
Maximum length of kinematics parameters	$R_f$ must be 2,000.0 [mm] or less and $L_m$ must be 10,000.0 [mm] or less.
Maximum angle between link 1 and link 2	The angle between link 1 and link 2 must be 0.0 to 180.0 [degree].
Relationship between $R_f$ and $R_m$ length	$R_m$ must be shorter than $R_f$ .

## 2 Features of NJ Robotics CPU Unit

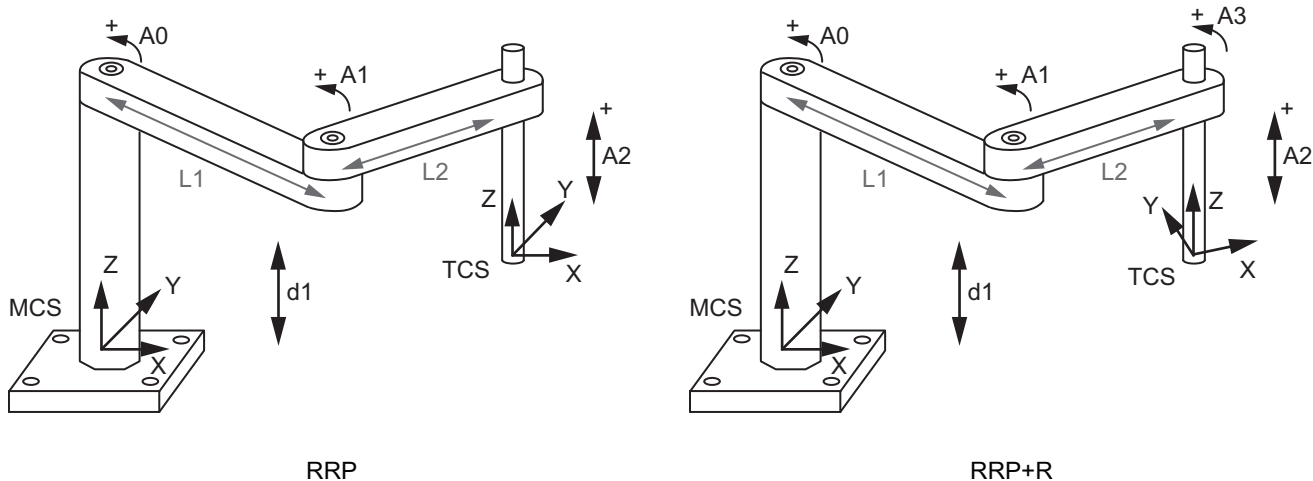
### ● Setting for SCARA

SCARA kinematics consists of 3 or 4 axes, depending of if wrist is part of it or not.

In all types of SCARA kinematics, the origin of the robot default Machine Coordinate System (MCS) is fixed on the base of robot.

All SCARA kinematics types support the both, right- and left-arm positions. There is no kinematics parameter for setting arm configuration. Robotics instructions detect and keep operation in the current arm configuration of robot.

Set the following parameters for SCARA RRP and RRP+R.

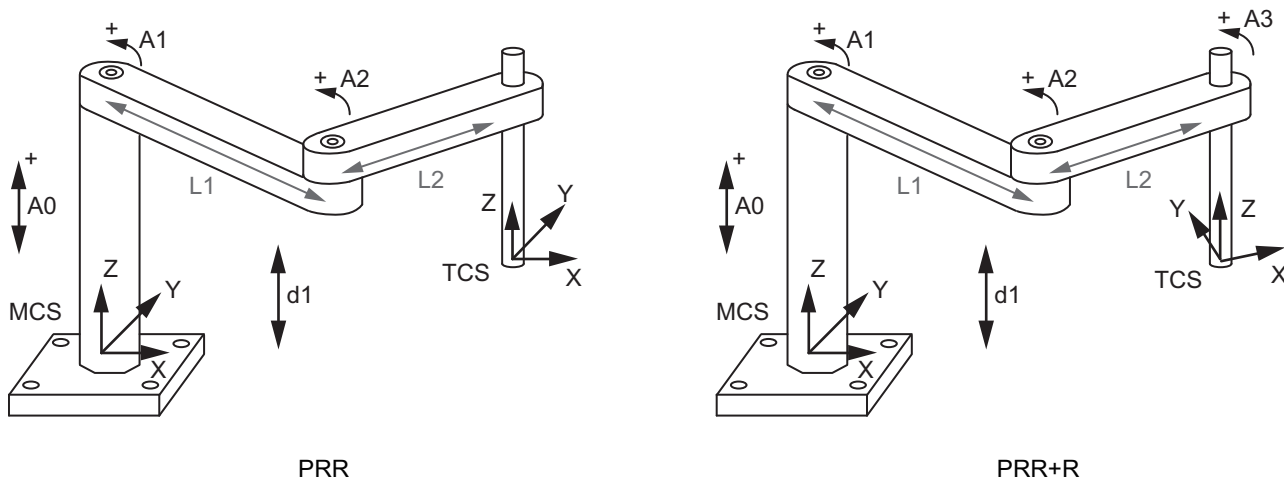


Parameter	Description
L1	First link length
L2	Second link length
d1	Distance of end effector from base in Z when axis A2=0

The kinematics parameters for SCARA RRP are the same as SCARA RRP+R.

A wrist axis must be controlled as a single axis in SCARA RRP kinematics. In SCARA RRP+R kinematics, the robot instructions control the wrist axis as a part of the axes group (4 axes).

Set the following parameters for SCARA PRR and PRR+R.



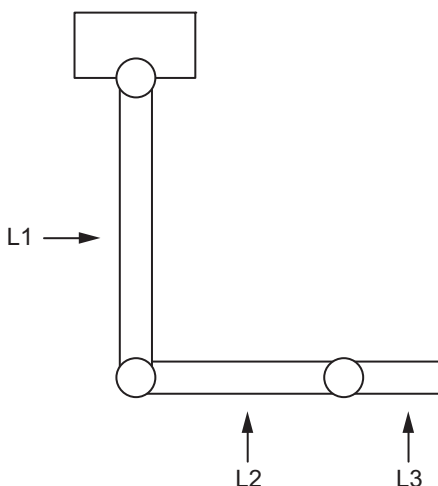
Parameter	Description
L1	First link length
L2	Second link length
d1	Distance of end effector from base in Z when axis A0=0

The kinematics parameters for SCARA PRR are the same as SCARA PRR+R.

A wrist axis must be controlled as a single axis in SCARA PRR kinematics. In SCARA PRR+R kinematics, the robot instructions control the wrist axis as a part of the axes group (4 axes).

### ● Setting for Expansion1

Set the following parameters for Expansion1.



Parameter	Description
L1	Length of Link 1 (unit: mm)
L2	Length of Link 2 (unit: mm)
L3	Length of Link 3 (unit: mm)

The Expansion1 robot is a serial manipulator composed by three revolute joints and three links. Expansion1 robot has the characteristic of having two inverse kinematics solutions. In current implementation only one solution is allowed (right elbow).

This robot requires an axis group composed of just 3 axes. The robot default Machine Coordinate System (MCS) origin is placed on the first axis. It means that the valid range of A1 axis is  $0^\circ \leq \theta \leq 180^\circ$ . The valid range of the other 2 axes (A0, A2) is  $-180^\circ < \theta \leq 180^\circ$ . When commanded position of one of the axes is  $180^\circ$ , the actual position might go over the limit. In that case MC\_GroupMon will set the output Valid to False and will not show the position.



### Precautions for Safe Use

---

Expansion1 kinematics supports a right-arm Planar robot only. Do not use the kinematics if a robot is in a left-arm position.

If the robot was in a left-arm position and you used the kinematics and MC\_InverseKin instruction to calculate target joint positions on ACS and you provided the positions to the joints, the robot might break or one or more axis might go in an error state.

If a Planar robot is in a left-arm position, it has to be moved first to a right-arm position before using the Expansion1 kinematics to control it. There are 2 ways to do that:

- After the servo off is applied to each axis of the Robot, ensure your safety and manually move the robot to a right-arm position.
  - Reset the Axis group of the robot and move the “elbow” axis by a single-axis motion command to form right-arm robot position.
- 

## 2-3-2 Workspace Check

The workspace check refers to a function that checks the range of motion of the robot TCP.

If the robot TCP is placed outside the workspace, an error occurs and operation of the axes group stops.

When a robot operation instruction is executed, the check function checks whether the TCP position meets the following four conditions.

- Target position of robot TCP when starting robot instruction
- Command position of robot TCP each motion cycle when robot instruction is ongoing
- Current command position of robot TCP when starting robot instruction

To use the workspace check, you need to set the workspace check parameters for the axes group. This section describes the settings of the workspace parameters.

To set an axes group, use the MC\_SetKinTransform (Set Kinematics Transformation) instruction.

Refer to *MC\_SetKinTransform* on page 4-6 for details on the MC\_SetKinTransform (Set Kinematics Transformation) instruction.

## Types of Workspaces

---

There are seven types of workspaces: Delta3Workspace, Delta2Workspace, Cartesian3DWorkspace, Cartesian2DWorkspace, HBotWorkspace, ScaraWorkspace and Expansion1Workspace.

Select the workspace according to the robot type.



### Precautions for Safe Use

---

Be sure to select a correct combination of *KinType* (Kinematics Type Selection) and *WorkspaceType* (Workspace Type Selection) variables for the MC\_SetKinTransform (Set Kinematics Transformation) instruction.

Check that the workspace check function is performed as intended.

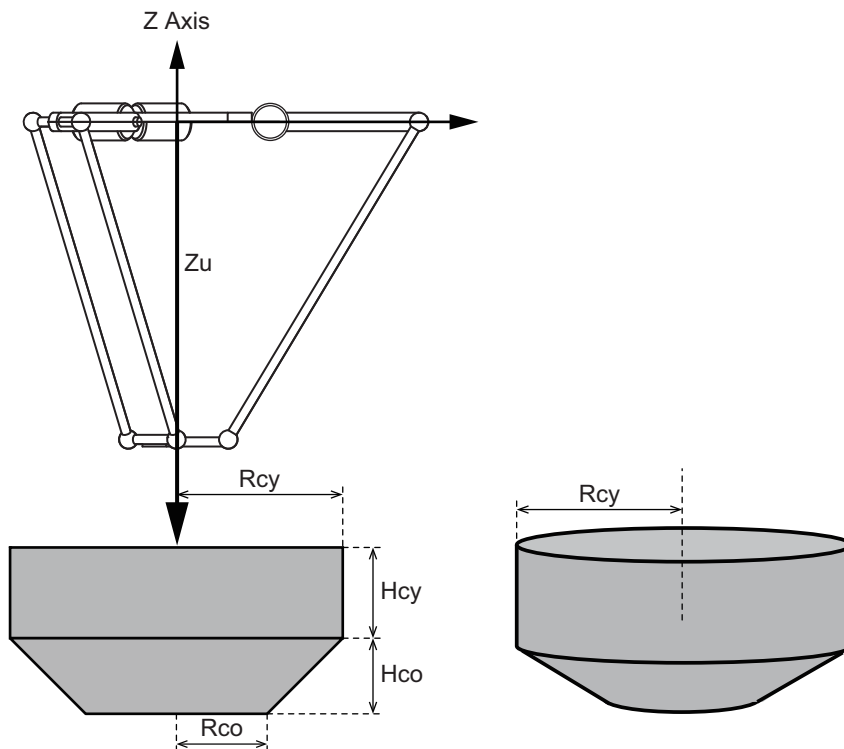
---

## Setting Workspace Parameters

This section describes details on parameter settings for each workspaces.

### ● Delta3Workspace

The shape of Delta3Workspace consists of a combination of a cylinder and a truncated cone.



Set the following parameters.

Parameter	Description
$Z_u$	The front face of the cylinder. The position relative to the origin of Z axis (unit: mm)
$R_{cy}$	The radius of the cylinder (unit: mm)
$H_{cy}$	The height of the cylinder (unit: mm)
$H_{co}$	The height of the truncated cone (unit: mm)
$R_{co}$	The radius of the bottom of the truncated cone (unit: mm)



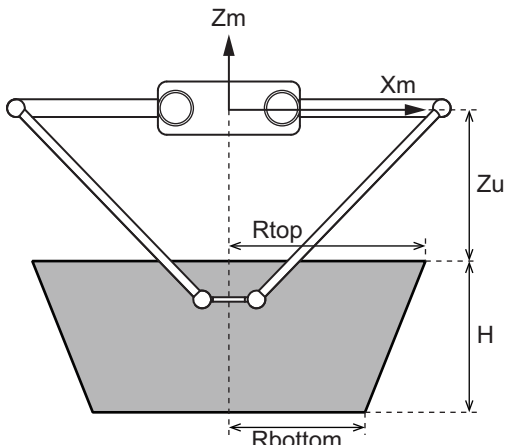
### Precautions for Correct Use

A Delta3R with the motor of the fourth axis attached to the fixed frame generally has a smaller range of motion than other Delta3 robots. In this case, you need to adjust workspace parameters to fit the smaller range of motion.

## 2 Features of NJ Robotics CPU Unit

### ● Delta2Workspace

The shape of Delta2Workspace is a trapezoid.

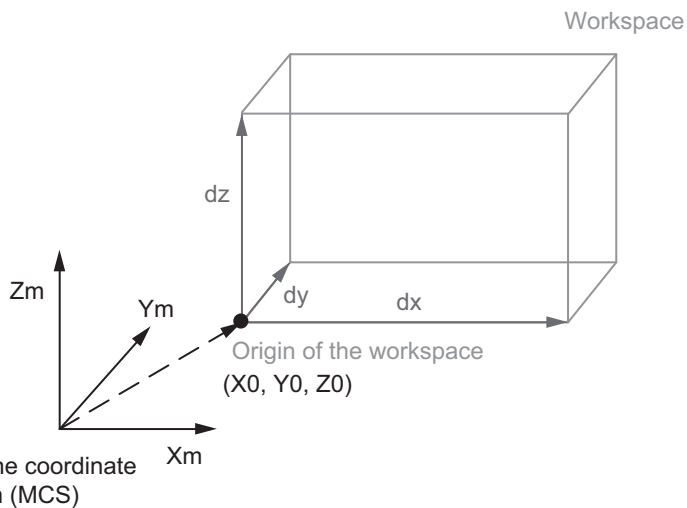


Set the following parameters.

Parameter	Description
$Z_u$	The top of the trapezoid. The position relative to the origin of Z axis (unit: mm)
$H$	The height of the trapezoid (unit: mm)
$R_{top}$	The radius of the top of the trapezoid (unit: mm)
$R_{bottom}$	The radius of the bottom of the trapezoid (unit: mm)

### ● Cartesian3DWorkspace

The shape of Cartesian3D workspace is a rectangular solid.

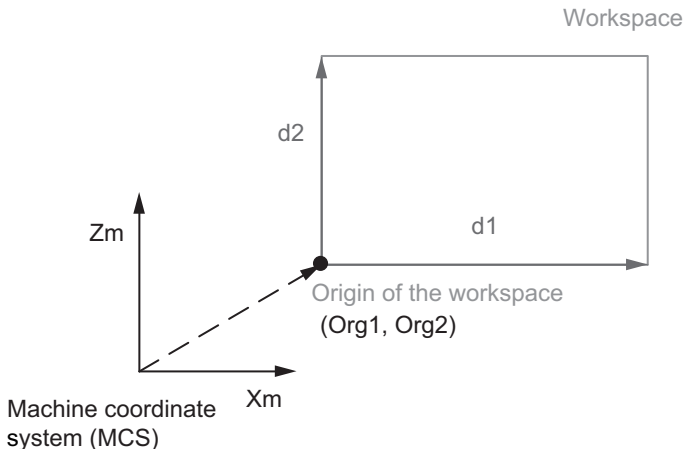


Set the following parameters.

Parameter	Description
$X_0$	Origin of the workspace: X coordinate (unit: mm)
$Y_0$	Origin of the workspace: Y coordinate (unit: mm)
$Z_0$	Origin of the workspace: Z coordinate (unit: mm)
$dx$	Workspace dimension on the X direction (unit: mm)
$dy$	Workspace dimension on the Y direction (unit: mm)
$dz$	Workspace dimension on the Z direction (unit: mm)

● **Cartesian2DWorkspace**

The shape of Cartesian2D workspace is a rectangular in specified plane.

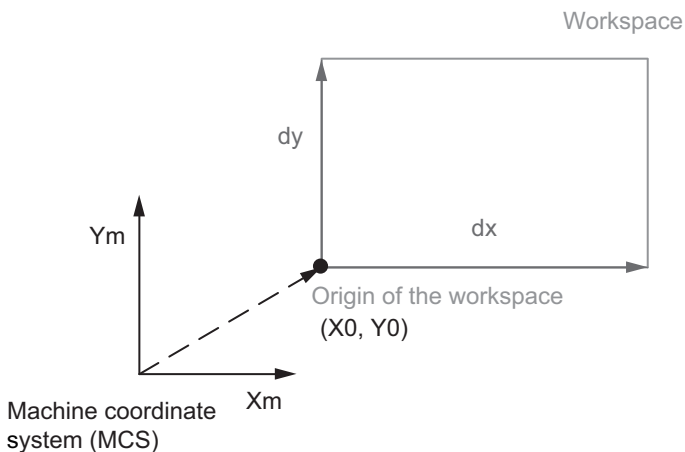


Set the following parameters.

Parameter	Description
Org1	Origin of the workspace: the first coordinate (unit: mm)
Org2	Origin of the workspace: the second coordinate (unit: mm)
d1	Workspace dimension on the first coordinate direction (unit: mm)
d2	Workspace dimension on the second coordinate direction (unit: mm)
Plane	Index to defined planes: 0: plane XY – the first coordinate is X and the second is Y 1: plane XZ – the first coordinate is X and the second is Z 2: plane YZ – the first coordinate is Y and the second is Z

● **HBotWorkspace**

The shape of H-Bot workspace is a rectangular in plane XY.



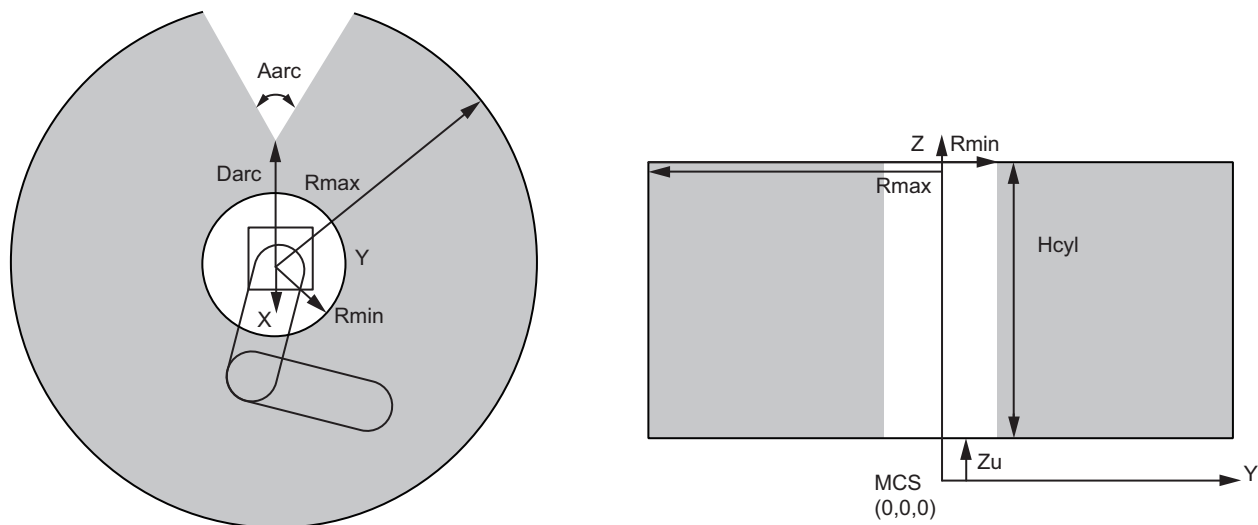
Set the following parameters.

Parameter	Description
X0	Origin of the workspace: X coordinate (unit: mm)
Y0	Origin of the workspace: Y coordinate (unit: mm)
dx	Workspace dimension on the X direction (unit: mm)
dy	Workspace dimension on the Y direction (unit: mm)

## 2 Features of NJ Robotics CPU Unit

### ● ScaraWorkspace

The shape is a volume bounded by two concentric cylinders and two planes intersecting the outer cylinder.

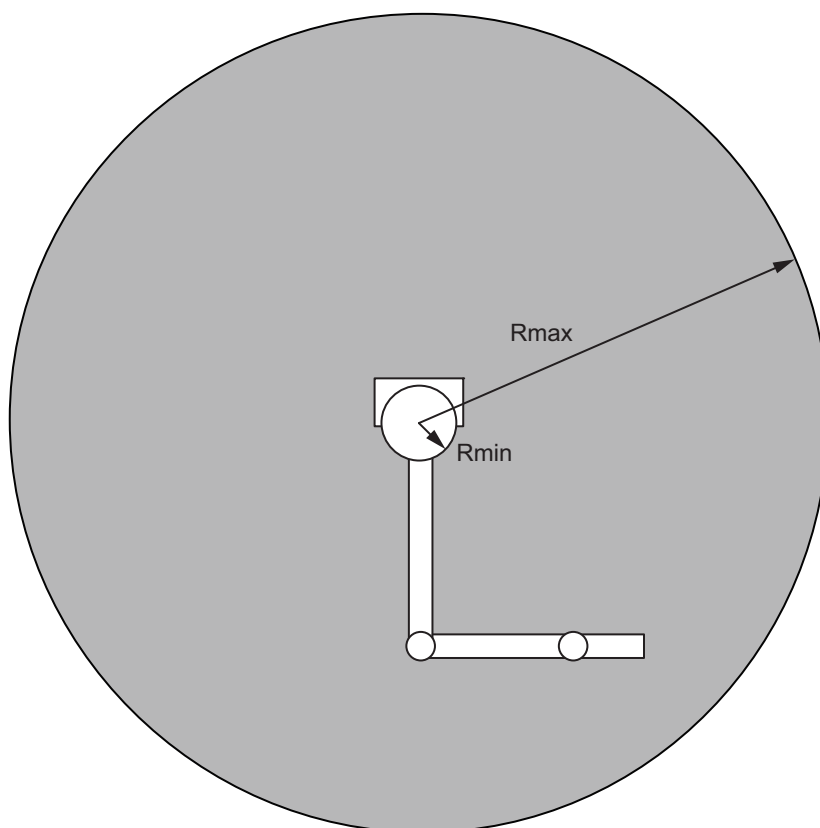


Set the following parameters.

Parameter	Description
Rmin	Internal radius
Rmax	External radius
Zu	Z coordinate of the bottom part of the cylinder
Hcyl	Height of the cylinder
Darc	Arc origin distance (Distance from the center of the cylinder to the origin of the arc along X negative axis)
Aarc	Arc angle

## ● Expansion1Workspace

The Expansion1Workspace is an annulus that is a ring-shaped object, especially a region bounded by two concentric circles.



Set the following parameters.

Parameter	Description
Rmin	Radius of the internal circle (unit: mm)
Rmax	Radius of the external circle (unit: mm)



### Precautions for Safe Use

Choose the correct combination of Kinematics type and Workspace type which are set by MC\_SetKinTransform.

Confirm if the workspace check works as expected.

## Robot Motion Range

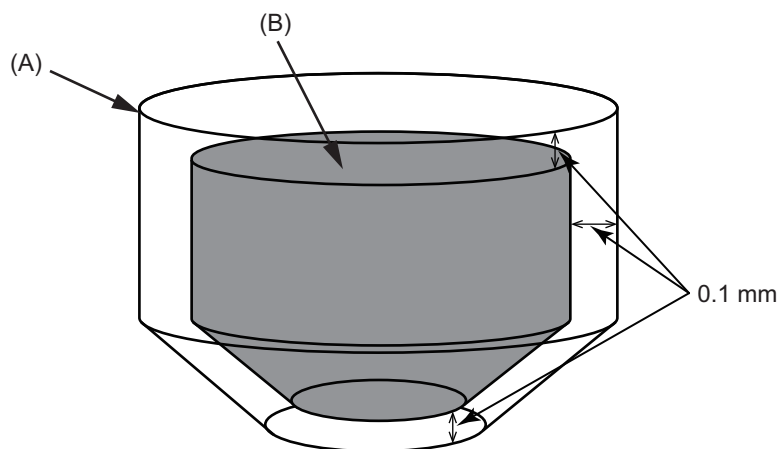
The space of the robot motion range exists 0.1 mm back from each edge of the workspace set by the MC\_SetKinTransform (Set Kinematics Transformation) instruction.

For the following kinematics: Delta-3, Delta 3R, Delta-2, Cartesian 3D, Cartesian 3D Gantry, Cartesian 2D, Cartesian 2D Gantry and H-Bot, the robot motion range is 0.1 mm smaller in all dimensions from the Workspace that is set by the MC\_SetKinTransform.

For SCARA and Expansion1 kinematics, the robot motion range is same in all dimensions as the Workspace that is set by the MC\_SetKinTransform.

If the TCP target position or the TCP actual current position is outside the robot motion range shown below, an error is detected and the robot stops immediately.

The following figure shows an example of Delta3Workspace.



Symbol	Description
(A)	The workspace set by the MC_SetKinTransform (Set Kinematics Transformation) instruction
(B)	The robot motion range

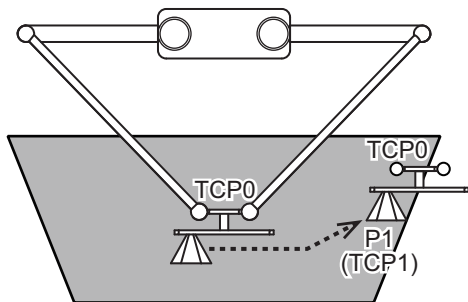
## Precautions on Using Robot Tool

When a robot tool is used, the robot TCP is always used for the workspace check.

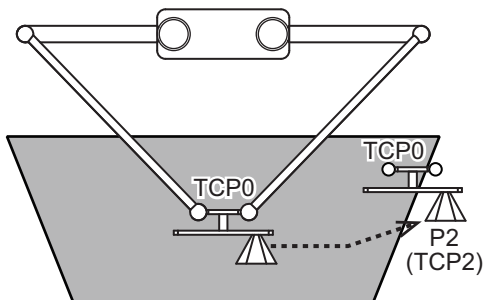


### Precautions for Correct Use

If you select a robot tool other than the robot TCP, the robot TCP (TCP0) placed outside the workspace may cause an error, even if the robot tool TCP (TCP1) is within the workspace. Confirm safety before use.



The robot tool TCP placed outside the workspace does not cause an error if the robot TCP is within the workspace.



### 2-3-3 User Coordinate System (UCS)

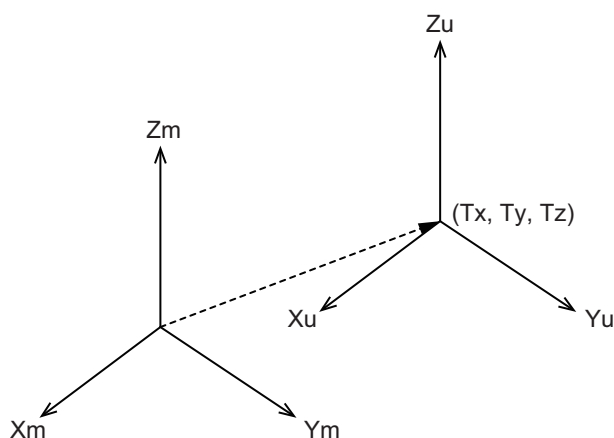
Use the `MC_DefineCoordSystem` (Define Coordinate) instruction to set origin positions ( $T_x$ ,  $T_y$ ,  $T_z$ ,  $R_x$ ,  $R_y$ ,  $R_z$ ) of user coordinate systems (UCS).

This section describes the origin positions you set.

Refer to *MC\_DefineCoordSystem* on page 4-17 for details on the `MC_DefineCoordSystem` (Define Coordinate) instruction.

#### Position Setting ( $T_x$ , $T_y$ , $T_z$ )

First you must set the origin of the user coordinate system (UCS) with the distance ( $T_x$ ,  $T_y$ ,  $T_z$ ) from the origin of the machine coordinate system (MCS).



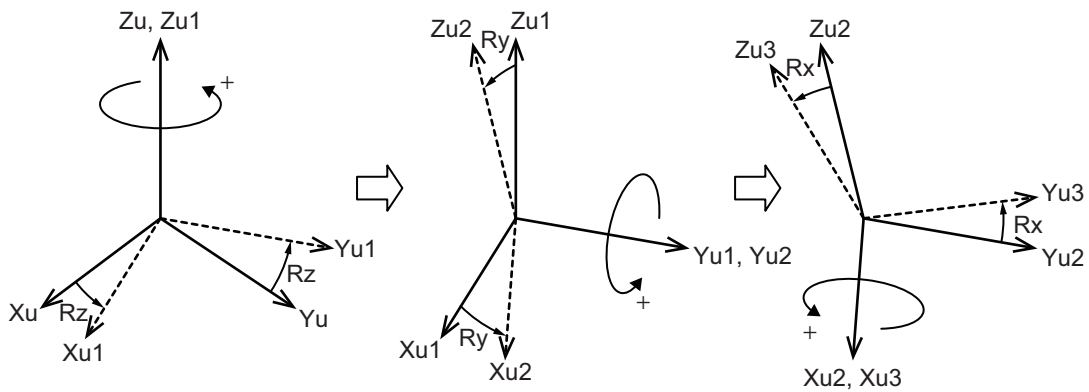
Coordinate axis	Coordinate System
$X_m$ - $Y_m$ - $Z_m$	Machine coordinate system (MCS)
$X_u$ - $Y_u$ - $Z_u$	User coordinate system (UCS)

#### Rotation Setting ( $R_x$ , $R_y$ , $R_z$ )

Set the rotation  $R_z$  around Z axis. The positive direction of rotation  $R_z$  is a clockwise direction from  $X_u$  axis to  $Y_u$  axis when Z axis positive direction is viewed from the origin.

Set the rotation  $R_y$  around  $Y_{u1}$  axis.  $Y_{u1}$  is an axis of the user coordinate system  $X_{u1}$ - $Y_{u1}$ - $Z_{u1}$  created by the rotation  $R_z$ . The positive direction of rotation  $R_y$  is a clockwise direction from  $Z_{u1}$  axis to  $X_{u1}$  axis when  $Y_{u1}$  axis positive direction is viewed from the origin.

Set the rotation  $R_x$  around  $X_{u2}$  axis.  $X_{u2}$  is an axis of the user coordinate system  $X_{u2}$ - $Y_{u2}$ - $Z_{u2}$  created by the rotation  $R_y$ . The positive direction of rotation  $R_x$  is a clockwise direction from  $Y_{u2}$  axis to  $Z_{u2}$  axis when  $X_{u2}$  axis positive direction is viewed from the origin.



### Precautions for Correct Use

User coordinate systems (UCS) are set in relation with a machine coordinate system (MCS).

If you re-execute the MC\_SetKinTransform (Set Kinematics Transformation) instruction for a robot, you must re-define all user coordinate systems for the robot.

Use the MC\_DefineCoordSystem (Define Coordinate) instruction to set user coordinate systems.

## 2-3-4 Monitoring

You can monitor the following robot information.

- The command current position and the actual current position of TCP in the selected coordinate system for the selected robot tool
- The actual current velocity of TCP in the selected coordinate system for the selected robot tool, and the actual current velocity in the directions of X axis, Y axis and Z axis

Use the MC\_GroupMon (Group Monitor) instruction to monitor the information. Refer to *MC\_GroupMon* on page 4-25 for details on the MC\_GroupMon (Group Monitor) instruction.

## 2-3-5 Time-specified Absolute Positioning

This function moves the robot from the current position to the set target position (X, Y, Z, Rx, Ry, Rz) in the specified time.

A robot could be moved using a polynomial function (3rd degree) or an Universal CAM Curve function for each Cartesian coordinate; it generates a linear movement in 3D Cartesian space.

- Polynomial 3 curve. It uses a simple 3rd degree polynomial to interpolate the path between initial position and final position.
- Modified sine curve. It is based on a combination of sinusoidal trajectories. It allows controlling acceleration and belongs to Universal CAM Curve family.
- Modified constant velocity curve. It is based on combination of sinusoidal and linear trajectories. It has a constant velocity during one half of trajectory time. It belongs to Universal CAM Curve family.

Use this function to move the robot to a desired position.

Use the MC\_MoveTimeAbsolute (Time-specified Absolute Positioning) instruction to move the robot with this function.

Refer to *MC\_MoveTimeAbsolute* on page 4-31 for details on the MC\_MoveTimeAbsolute (Time-specified Absolute Positioning) instruction.

## 2-3-6 Conveyor Synchronization

Conveyor tracking is a process in which an object (workpiece) laying on a moving or stationary conveyor belt is followed-up by a robot. Robot can perform action when is synchronized with the object, it depends on the application. In the most typical application the robot picks-up the object and places it in another place (box, conveyor...). Objects laying on the conveyor could be detected by a vision system or by a sensing device (magnetic sensor, etc.).

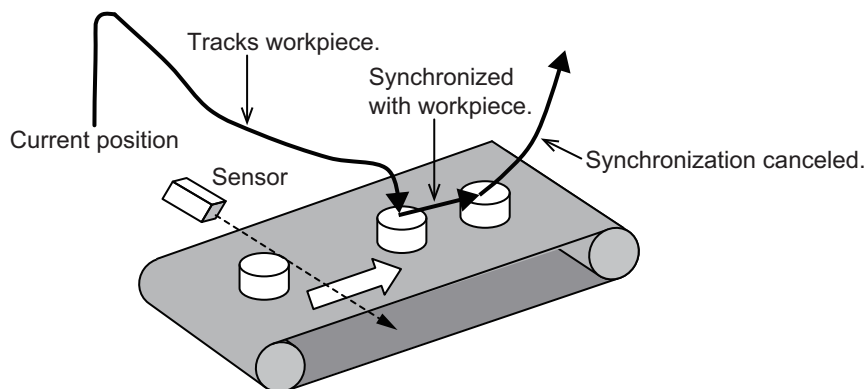
By using robotics functions you can make the robot to move from the current position and track the specified workpiece on the conveyor.

After catching up, the robot moves in synchronization with the workpiece.

The synchronization is canceled when the robot moves to the next target position.

Use the `MC_SyncLinearConveyor` (Start Conveyor Synchronization) instruction to start synchronization with the conveyor. To cancel the synchronization, use the `MC_SyncOut` (End Synchronization) instruction.

Refer to `MC_MoveDirectAbsolute` on page 4-41 for details on the `MC_SyncLinearConveyor` (Start Conveyor Synchronization) instruction, and `MC_SyncOut` on page 4-73 for the `MC_SyncOut` (End Synchronization) instruction.



## Offset Setting

You can set offsets during conveyor synchronization.

If an offset is enabled, the offset value is added to the position of the workpiece on the conveyor, and TCP moves to the calculated position.

The offset value is applied in every control period. Therefore, the robot can operate in different ways while synchronization with the conveyor is maintained.

Refer to *Setting Offsets* on page 4-67 for details.



### Precautions for Correct Use

- You can enable offsets only when TCP is in synchronization with the workpiece. If you enable offsets when synchronization is not performed, an error will occur.
- If an offset is changed from Disabled to Enabled, the offset value of the moment is applied. Note that the motion of TCP may change suddenly depending on the offset value.
- If an offset is changed from Enabled to Disabled, the offset value of the moment is maintained.

## 2-3-7 Inverse Kinematics

This function transforms a robot TCP position in the machine coordinate system (MCS) to a position in the axis coordinate system (ACS) of each robot axis.

When you want to move the robot along a desired trajectory, use this function together with the `MC_GroupSyncMoveAbsolute` (Axes Group Cyclic Synchronous Absolute Positioning) instruction.

For the inverse kinematics, use the `MC_InverseKin` (Inverse Kinematics) instruction.

Refer to the *NJ/NX-series Motion Control Instructions Reference Manual* (Cat. No. W508) for details on the `MC_GroupSyncMoveAbsolute` (Axes Group Cyclic Synchronous Absolute Positioning) instruction, and `MC_InverseKin` on page 4-82 for the `MC_InverseKin` (Inverse Kinematics) instruction.

## 2-3-8 Robot Tool

You can define/select robot tools for a robot for which the kinematics is already set.

Use the `MC_DefineToolTransform` (Define Tool Coordinate) instruction to make axes group settings.

Refer to `MC_DefineToolTransform` on page 4-21 for details on the `MC_DefineToolTransform` (Define Tool Coordinate) instruction.

## Robot Tool Definition

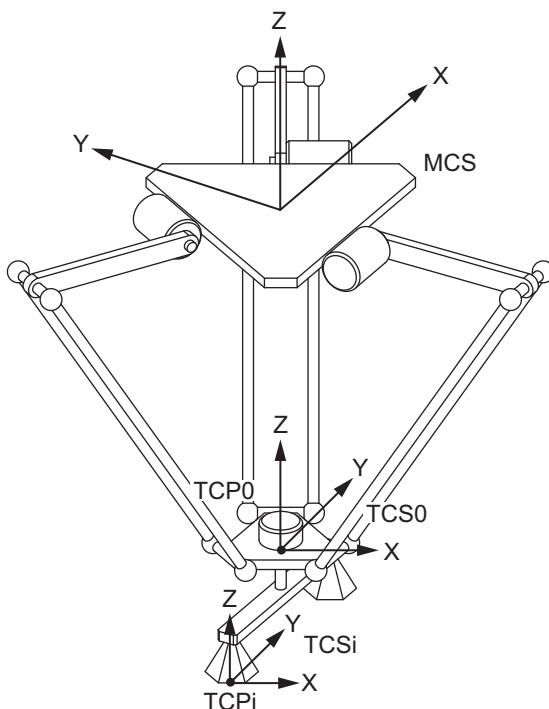
You can define up to 16 robot tools with IDs 1 to 16 for each robot.

Together with the default TCP, the total number of robot tools is 17.

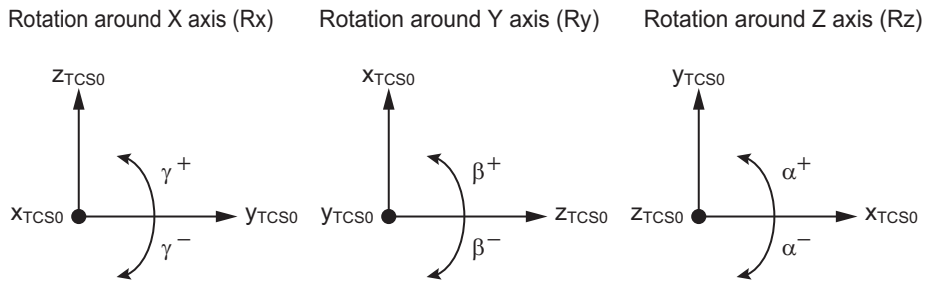
$TCP_i$  denotes a TCP with ID  $i$ . You can select  $TCP_i$  after you define a robot tool.

To define a robot tool, use the coordinates ( $T_x$ ,  $T_y$ ,  $T_z$ ,  $R_x$ ,  $R_y$ ,  $R_z$ ) with the reference point  $TCS_0$  of the tool coordinate system (TCS).

The coordinates ( $T_x$ ,  $T_y$ ,  $T_z$ ) represent a relative position of  $TCP_i$  to  $TCP_0$ .



## 2 Features of NJ Robotics CPU Unit



Some parameters for robot tool definition are invalid depending on the kinematics type. In such a case, the values set for invalid parameters are ignored.

Kinematics type		Target position on X axis: Tx [mm]	Target position on Y axis: Ty [mm]	Target position on Z axis: Tz [mm]	Target rotational position around X axis: Rx [degree]	Target rotational position around Y axis: Ry [degree]	Target rotational position around Z axis: Rz [degree]
Delta3		Valid	Valid	Valid	Invalid	Invalid	Invalid
Delta3R		Valid	Valid	Valid	Invalid	Invalid	Valid
Delta2		Valid	Invalid	Valid	Invalid	Invalid	Invalid
Expansion1		Valid	Valid	Invalid	Invalid	Invalid	Valid
Cartesian 3D Cartesian 3D-Gantry		Valid	Valid	Valid	Invalid	Invalid	Invalid
Cartesian 2D	Plane XY	Valid	Valid	Invalid	Invalid	Invalid	Invalid
	Plane XZ	Valid	Invalid	Valid	Invalid	Invalid	Invalid
	Plane YZ	Invalid	Valid	Valid	Invalid	Invalid	Invalid
Cartesian 2D-Gantry H-Bot		Valid	Valid	Invalid	Invalid	Invalid	Invalid
SCARA RRP SCARA PRR		Valid	Valid	Valid	Invalid	Invalid	Invalid
SCARA RRP+R SCARA PRR+R		Valid	Valid	Valid	Invalid	Invalid	Valid

## Example of Robot Tool Operation

If you select a robot tool, TCS0 of the selected tool coordinate system (TCS) becomes the base point for the robot.

In this example, the robot tool is defined as follows, and the selected robot tool ID1 moves to the target position P2.

### ● Robot Tool Definition

$$\text{TCP1} = (20.0, 0.0, -5.0, 0.0, 0.0, 0.0)$$

The unit for X, Y, Z is mm, and the unit for Rx, Ry, Rz is degree.

### ● Coordinate Position on MCS

The current position P0 of TCP0 is set as follows.

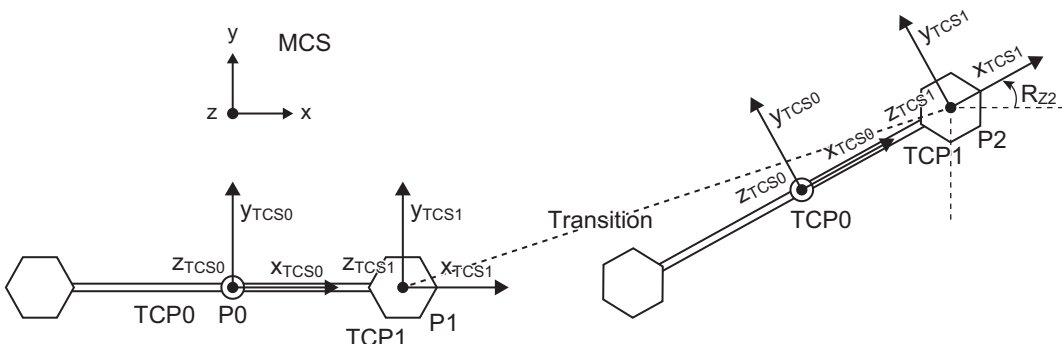
$$\text{P0} = (0.0, -20.0, -750.0, 0.0, 0.0, 0.0)$$

The following is the current position of TCP1.

$$\text{P1} = (20.0, -20.0, -755.0, 0.0, 0.0, 0.0)$$

The target position P2 is set as follows.

$$\text{P2} = (70.0, 0.0, -750.0, 0.0, 0.0, 30.0)$$



$$\text{P0} = (X_0, Y_0, Z_0, R_{x_0}, R_{y_0}, R_{z_0})$$

$$\text{P1} = (X_1, Y_1, Z_1, R_{x_1}, R_{y_1}, R_{z_1})$$

$$\text{P2} = (X_2, Y_2, Z_2, R_{x_2}, R_{y_2}, R_{z_2})$$



### Precautions for Correct Use

Robot tools are set in relation with a machine coordinate system (MCS).

If you re-execute the MC\_SetKinTransform (Set Kinematics Transformation) instruction for a robot, you must re-define all robot tools for the robot.

Use the MC\_DefineToolTransform (Define Tool Coordinate) instruction to define robot tools.

### 2-3-9 Robot Jogging

This function performs jogging in a specified direction.

Jogging is performed in the specified coordinate system.

You can select machine coordinate system (MCS), user coordinate system (UCS), or tool coordinate system (TCS).

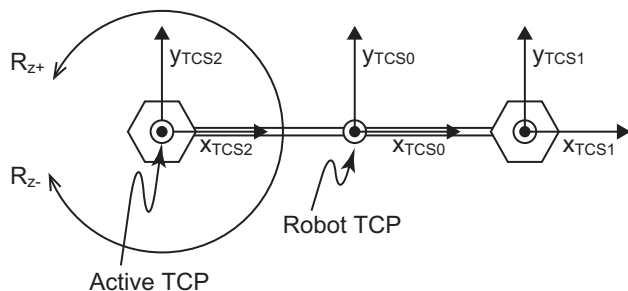
Use the MC\_RobotJog (Axes Group Jog) instruction to execute jogging.

Refer to *MC\_RobotJog* on page 4-87 for details on the MC\_RobotJog (Axes Group Jog) instruction.

### Rotational Axis Jogging

During jogging (rotation) of a rotational axis (Rx, Ry, Rz), the position of the selected tool is maintained.

The following figure shows an example of jogging executed for Rz axis of the active TCP.



## 2-3-10 Checking Maximum Interpolation Velocity and Maximum Interpolation Acceleration/Deceleration

The maximum interpolation velocity check and maximum interpolation acceleration/deceleration check are performed for the robot TCP and the active TCP.

The maximum interpolation velocity and maximum interpolation acceleration/deceleration for the robot TCP are set to protect the robot.

The maximum interpolation velocity and maximum interpolation acceleration/deceleration for the active TCP are set to protect workpieces.

The following table shows the frequency of maximum interpolation velocity check and maximum interpolation acceleration/deceleration check.

Details are provided below.

B/I: Background and instruction, B: Background only, I: Instruction only, NP: Not performed

Timing	Robot TCP is selected				Active TCP is selected			
	Maximum Interpolation Velocity		Maximum Interpolation Acceleration/Deceleration		Maximum Interpolation Velocity		Maximum Interpolation Acceleration/Deceleration	
	Robot TCP	Active TCP	Robot TCP	Active TCP	Robot TCP	Active TCP	Robot TCP	Active TCP
At instruction execution	I	NP	I	NP	NP	NP	NP	NP
In every control period	B/I	NP	B/I	NP	B	I	B	I



### Precautions for Correct Use

The checks are disabled if the maximum interpolation velocity and maximum interpolation acceleration/deceleration are set to 0.0.

## Checks in Background

You can set the values detected as the velocity error and acceleration error of the robot TCP.

The maximum interpolation velocity check and maximum interpolation acceleration/deceleration check are enabled when the following conditions are met.

- The kinematics are set and the axes group is enabled.
- The maximum interpolation velocity and maximum interpolation acceleration/deceleration are set to the values other than 0.0.
- The background velocity error check and acceleration error check are enabled only for command values. They are disabled for actual values.
- If the command value exceeds the maximum interpolation velocity or maximum interpolation acceleration/deceleration, an axes group error occurs and instruction execution stops.
- The velocity error check and acceleration error check are executed for the robot TCP.
- Use the MC\_SetKinTransform (Set Kinematics Transformation) instruction to set the maximum interpolation velocity and maximum interpolation acceleration/deceleration. The checks are disabled if the values are set to 0.0.

## 2 Features of NJ Robotics CPU Unit

- The factory default value is set to 0.0 (disabled).  
To enable the checks, set the values other than 0.0.
- Depending on the instruction, the background checks are enabled/disabled. See the following instruction list.

Check-enabled instructions	Check-disabled instructions
<ul style="list-style-type: none"> <li>• MC_MoveTimeAbsolute</li> <li>• MC_SyncLinearConveyor</li> <li>• MC_SyncOut *1</li> <li>• MC_RobotJog</li> <li>• MC_MoveDirectAbsolute</li> </ul>	<ul style="list-style-type: none"> <li>• MC_SyncOut *2</li> <li>• MC_GroupImmediateStop</li> <li>• MC_GroupStop</li> <li>• MC_GroupSyncMoveAbsolute</li> </ul>

\*1. When the Sync Stop Type Selection is set to *Synchronized stop* or *Synchronized stop with Deceleration*

\*2. When the Sync Stop Type Selection is set to *Immediate stop*

### Checks in Instructions

The maximum interpolation velocity and maximum interpolation acceleration/deceleration are checked in the following timings in instructions.

- At an instruction execution. The checks are enabled only when the robot TCP is selected.
- In every control period
- While execution of the instruction is in progress, the command velocity and command acceleration of the selected TCP are checked in every control period.
- If the command velocity exceeds the maximum interpolation velocity, or if the command acceleration/deceleration exceeds the maximum interpolation acceleration/deceleration, an axes group error occurs and instruction execution stops.
- The following are the instructions for which the checks in instructions are enabled. The checks are disabled if the set value is 0 or instruction execution is completed.

Check-enabled instructions
<ul style="list-style-type: none"> <li>• MC_MoveTimeAbsolute</li> <li>• MC_SyncLinearConveyor</li> <li>• MC_SyncOut *1</li> <li>• MC_MoveDirectAbsolute*2</li> </ul>

\*1. When the Sync Stop Type Selection is set to *Synchronized stop* or *Synchronized stop with Deceleration*

\*2. Only the check "In every control period" is performed but not at "At an instruction execution".

### ● Checks at Instruction Execution

The following two types of checks are performed at instruction execution.

- Sets the values detected as the velocity error and acceleration error for the trajectory data.
- Calculates the command velocity and command acceleration/deceleration during instruction execution. If the calculated value exceeds the maximum interpolation velocity or maximum interpolation acceleration/deceleration, an axes group error occurs and instruction execution stops.



### Precautions for Correct Use

The checks at instruction execution are enabled only when the robot TCP is specified.

If a TCP other than robot TCP is specified, the checks are disabled.

### ● Checks in Every Control Period

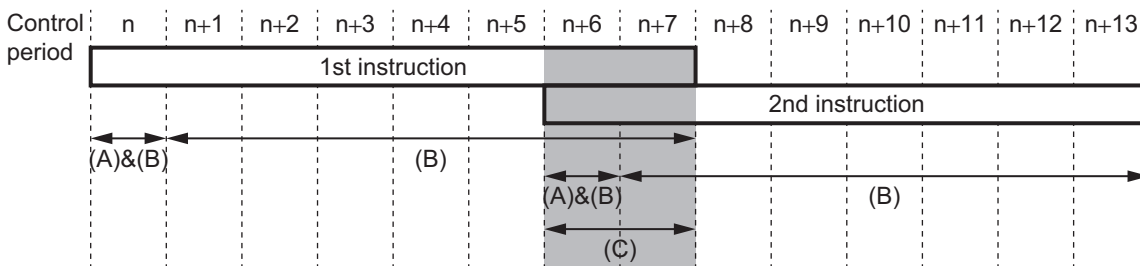
The following two types of checks are performed in every control period.

- Checks for the velocity error and acceleration error during instruction execution.
- If the command velocity or command acceleration/deceleration exceeds the maximum interpolation velocity or maximum interpolation acceleration/deceleration, an axes group error occurs and instruction execution stops.

In the first control period after instruction execution, only the velocity error is checked.

### ● Checks During Transition

During transition between two instructions, the velocity error and acceleration error are checked in both instructions.



Symbol	Meaning
(A)	Checks at instruction execution
(B)	Checks in every control period
(C)	Transition period

## 2-3-11 Multi-execution of Instructions with Buffered Mode

The NJ Robotics CPU Units support multi-execution of instructions with the buffered (standby) mode.

Refer to 4-2-2 *Multi-execution in Buffered Mode* on page 4-99 and the *NJ/NX-series CPU Unit Motion Control User's Manual* (Cat. No. W507) for details.

## 2-3-12 Multi-execution of Instructions with Blending Mode

The NJ Robotics CPU Units support multi-execution of instructions with the blending (mixing) mode.

Because the robot control requires smooth transitions, you need to select the *TransitionMode* (Transition Mode) input variable to blend robot instructions.

You can select the type of transition between instructions using the *TransitionMode* (Transition Mode) input variable in robot instructions.

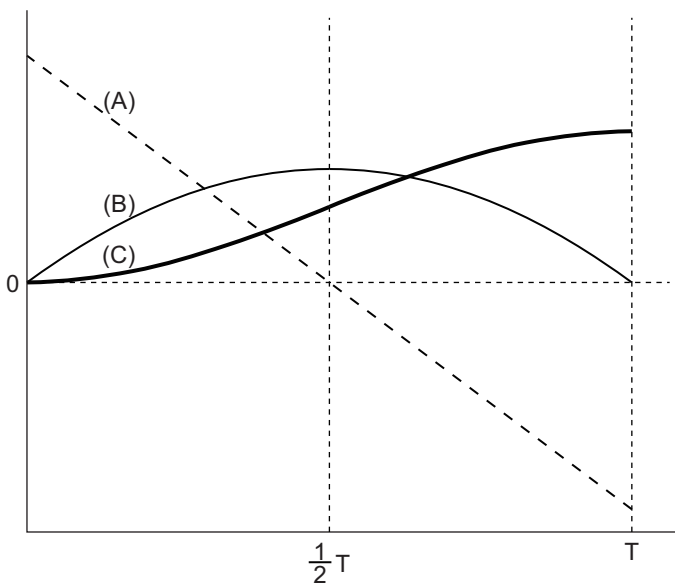
Refer to 4-2-3 *Multi-execution in Blending Mode* on page 4-100 and the *NJ/NX-series CPU Unit Motion Control User's Manual* (Cat. No. W507) for details.

### 2-3-13 Trajectory Types for Time-specified Motion

There are three options for the trajectory type of time-specified motion: the polynomial 3 curve, modified sine curve, and modified constant velocity curve.

#### Polynomial 3 Curve

A polynomial 3 curve takes 0.0 as the start velocity and end velocity.



Symbol	Meaning
(A)	Acceleration/deceleration
(B)	Velocity
(C)	Position

#### Modified Sine Curve

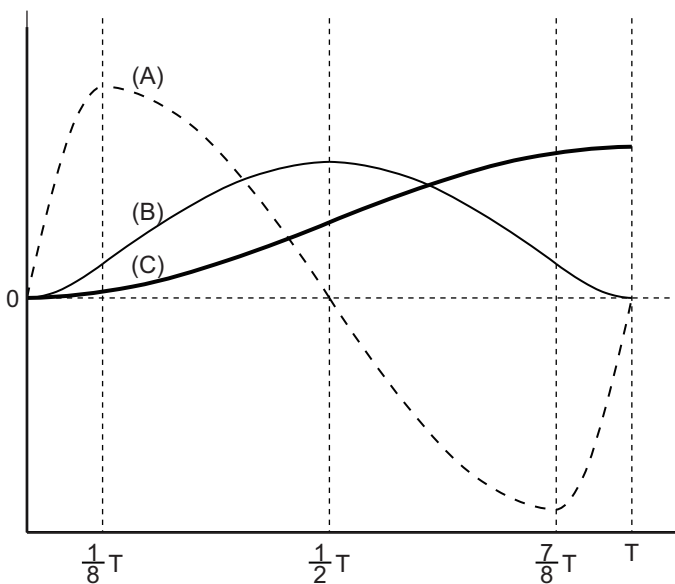
A modified sine curve consists of a combination of sine waves.

This is a type of universal cam curves to that can control acceleration.

It is recommended to select the modified sine curve to use the transition function.

Also use this curve for high-velocity, medium-load equipment.

The modified sine curve is effective in reducing the impact of load variations on the motor.



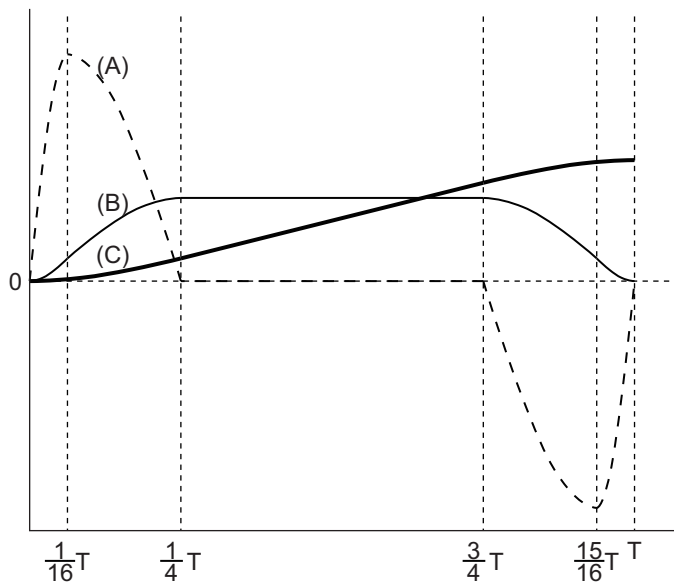
Symbol	Meaning
(A)	Acceleration/deceleration
(B)	Velocity
(C)	Position

## Modified Constant Velocity Curve

A modified constant velocity curve consists of a combination of sine waves and straight lines.

This is a type of universal cam curves in which the velocity is constant for the half of the trajectory time period.

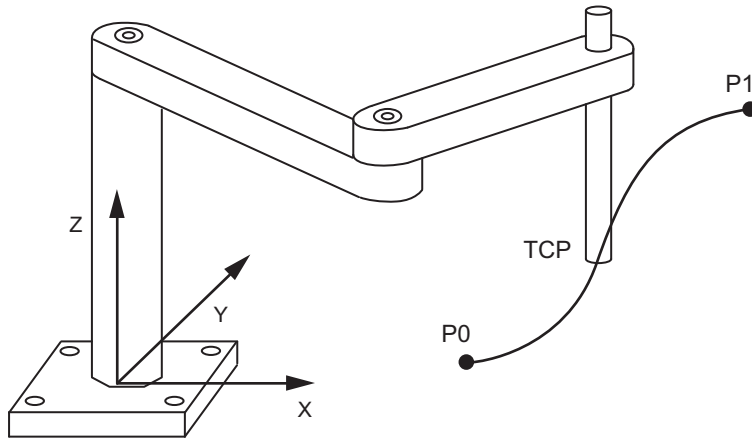
Use this curve for middle-velocity, high-load equipment.



Symbol	Meaning
(A)	Acceleration/deceleration
(B)	Velocity
(C)	Position

### 2-3-14 Joint Interpolated Point-to-Point Movement

This function moves the robot from the current position to the set target position (X, Y, Z, Rx, Ry, Rz), by doing multi-synchronized movements of its joints in joint space. Robot path between the points depends on the robot posture and is not always a straight line.



Target velocity of each joint is percentage (Velocity Ratio) of maximum velocity of each of them. Target acceleration and deceleration of each joint is determined in the same way by using Acceleration Ratio/Deceleration Ratio and maximum acceleration/maximum deceleration. The axis that needs the longest time to complete its movement is the limiting axis. Velocities, accelerations and decelerations of the other joints will be calculated in a way to complete their movements at the same time with the limiting axis. The function will create the shortest movement in joint space.

The function uses Trapezoidal and DoubleS profiles. It can make acceleration and deceleration “softer” by setting Jerk.

For a given value of Jerk and some combinations of path distance and axes dynamics (velocity, acceleration and deceleration), it might happen that axes-synchronization could not be achieved. In that case, the Jerk value will be respected and trajectory will be executed without scaling down values of some of the dynamics (velocity, acceleration, and deceleration) of an axis. The axis that is not scaled down will finish before the scaled ones.

The following are the good practices to avoid such situation:

- Avoid big difference in distance travel among axes.
- Review A/V (Acceleration-Velocity) relation in the axes configuration, ratio should be as similar as possible among robot axes.
- Avoid low ratios of J/A (Jerk-Acceleration) and very low values of Jerk. For example,  $J/A < 10$  or  $Jerk < 1000$ .

Due to robot settings and/or MC\_MoveDirectAbsolute instruction inputs, it is possible that good practices could not be applied and axes-synchronization not achieved for certain parameters.

Note Joint interpolated movement of robot is different than a movement in Cartesian space (e.g. MC\_MoveTimeAbsolute) because acceleration times of each separate axis are not equal, neither deceleration times, but only the total movement times.

Use this function to move the robot to a desired position by doing shortest movement in joint space.

Use this function also to change arm-configuration of SCARA robots.

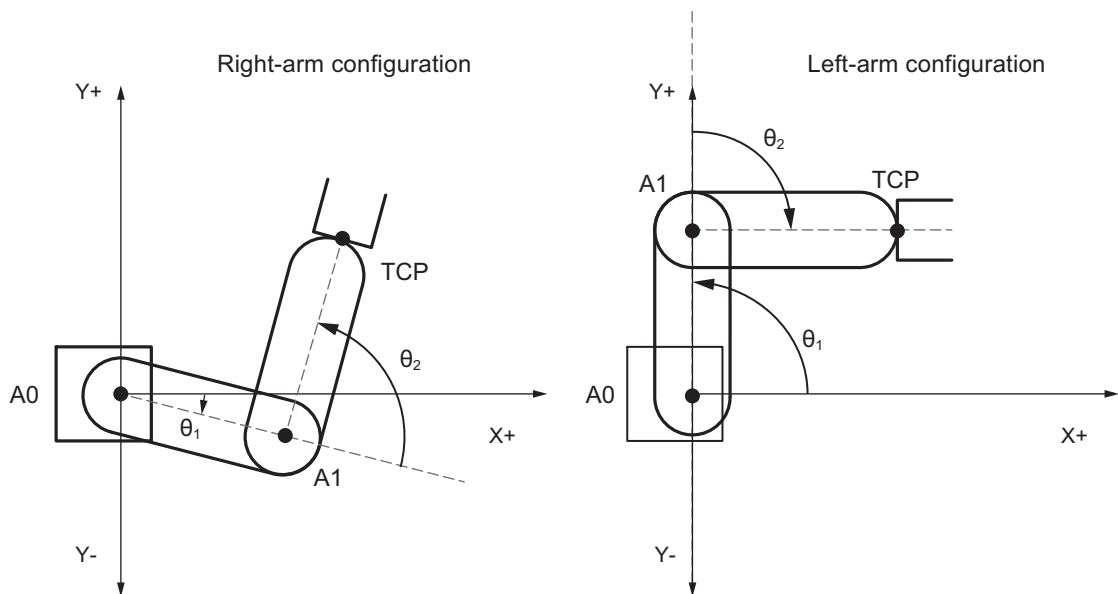
Use the MC\_MoveDirectAbsolute instruction to move the robot with this function.

Refer to *MC\_MoveDirectAbsolute* on page 4-41 for the details of the instruction.

## 2-3-15 Management of Robots with Multiple Arm Configurations

Some robots could have more than one arm configuration (axes position) to achieve a Cartesian pose. The one to be used must be selected by user or automatically by control system.

An example of a robot with multiple configurations is SCARA, which has left and right arm configuration.



For SCARA RRP:

- Right-arm configuration:  $\theta_2(A1) \geq 0.0^\circ$
- Left-arm configuration:  $\theta_2(A1) < 0.0^\circ$

For SCARA PRR:

- Right-arm configuration:  $\theta_2(A2) \geq 0.0^\circ$
- Left-arm configuration:  $\theta_2(A2) < 0.0^\circ$

Arm configuration of a SCARA robot can only be changed by user by using the following operations:

- Joint interpolated point-to-point movement with MC\_MoveDirectAbsolute by selecting Arm Configuration.
- Synchronized axes movement with MC\_GroupSyncMoveAbsolute by using points in ACS provided by MC\_InverseKin that also supports selecting Arm Configuration.
- Single axis movements by reset Axes group and moving the “elbow” axis by using a single-axis motion movement.

The following robot movements detect and keep the current arm configuration of robot.

- MC\_MoveTimeAbsolute
- MC\_SyncLinearConveyor
- MC\_SyncOut
- MC\_RobotJog

i.e. they do not have an input to select an arm configuration and they do not change the current arm configuration when the movement is performed, even when they start or finish movement in singularity configuration.

A singularity exists for SCARA when the rotary joints are aligned i.e. in case of RRP when  $A1 = 0^\circ \pm 360^\circ$  or  $A1 = 180^\circ \pm 360^\circ$  and in case of SCARA PRR when  $A2$  has one of these values. In a case of singularity, the robot loses 1 degree of freedom and the detected arm configuration is the right-arm one

## 2 Features of NJ Robotics CPU Unit

(default), if the previous command was not a robot movement instruction with left-arm configuration. In that case, the left-arm configuration will be kept.

In case of singularity, the following 2 situations require special attention though:

- (a) After executing a Home operation of SCARA robot axes, the robot needs to be moved with a point-to-point movement (MC\_MoveDirectAbsolute) where the arm configuration is set by the instruction.
- (b) After commanding a SCARA robot to a singularity position and re-starting the controller, the robot needs to be moved with a point-to-point movement (MC\_MoveDirectAbsolute) where the arm configuration is set by the instruction.

Arm configuration in the system is reset by executing MC\_SetKinTransfrom, MC\_GroupDisable or a command with target position in ACS e.g. MC\_GroupSyncMoveAbsolute and MC\_MoveDirectAbsolute.

Monitoring robot pose doesn't depend on arm configuration because direct kinematics has always solution and it is unique. Therefore for any set of axes position in ACS (A0, A1, A2) there is always solution in MCS (X, Y, Z).



# Setting Robot Functions

This section describes the settings required to use robots.

---

<b>3-1</b>	<b>Setting Group for Robot</b> .....	<b>3-2</b>
3-1-1	Group Setting Procedures .....	3-2
3-1-2	Axes Group Settings .....	3-3
3-1-3	Axis Settings .....	3-4
<b>3-2</b>	<b>Robot Kinematics Settings</b> .....	<b>3-9</b>
<b>3-3</b>	<b>Homing</b> .....	<b>3-10</b>
<b>3-4</b>	<b>Target Position and Direction Setting Method</b> .....	<b>3-11</b>
3-4-1	Position Settings .....	3-11
3-4-2	Wrist Control .....	3-12
<b>3-5</b>	<b>Checking Wiring from the Sysmac Studio</b> .....	<b>3-15</b>

## 3-1 Setting Group for Robot

This section describes how to set the group assigned to the robot and the specific items to set with the Sysmac Studio.

In this section, Delta3 is used as the example for the group setting procedure.

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



### Version Information

For the Sysmac Studio version 1.13 or lower, you need to enable the robot options. Refer to *A-2 Sysmac Studio Robot Options* on page A-21 for how to enable the Sysmac Studio robot options.

### 3-1-1 Group Setting Procedures

Use the following procedures.

- 1** Create a project.
- 2** Select NJ501-4□□□ for **CPU**.
- 3** In the **EtherCAT Network** field, register Servo Drives for three axes that are used for the robot.
- 4** Add three axes under **Axis Settings**.
- 5** Assign a Servo Drive to each axis added in the previous step.
- 6** Add an axes group for the robot.
- 7** Set **Axes Group Settings**.  
Refer to *3-1-2 Axes Group Settings* on page 3-3 for specific settings.
- 8** Set **Axis Settings** according to the mechanical specifications of each axis.  
Refer to *3-1-3 Axis Settings* on page 3-4 for specific settings.

This completes the settings of the axes group for the robot.

If necessary, set items for the Controller, save the project and transfer the data to the Controller.

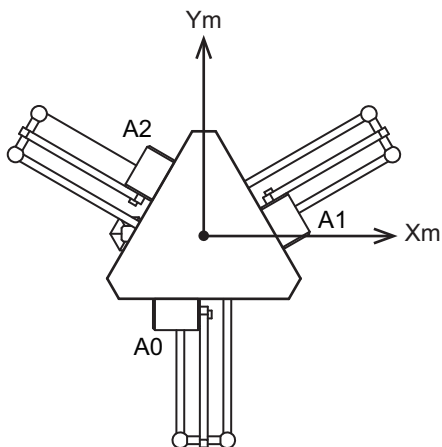
### 3-1-2 Axes Group Settings

Use the Sysmac Studio to set the axes group parameters for each axes group.

#### Axes Group Basic Settings

Set whether to use the axes group. If you use the axes group, set the axis configuration and the axes to use.

Parameter name	Setting	Set value
Axes Group Use	Set whether to enable or disable the axes group.	Select "Used axes group".
Composition	Set the axis composition of the axes group.	Select "3 axes" for using Delta3 robot. For using other robots, see the following table.
Composition Axes	Sets the axis number to assign it to the axes group.	
	Axis A0	Assign the axis located at A0 position in the following figure.
	Axis A1	Assign the axis located at A1 position in the following figure.
	Axis A2	Assign the axis located at A2 position in the following figure.



Parameter name	Kinematics type							
	Delta3	Delta3R	Delta2	Cartesian 3D	Cartesian 3D-Gantry	Cartesian 2D	Cartesian 2D-Gantry	H-Bot
Composition	3 axes	4 axes	2 axes	3 axes	4 axes	2 axes	3 axes	2 axes

Parameter name	Kinematics type		
	SCARA RRP SCARA PRR	SCARA RRP+R SCARA PRR+R	Expansion1
Composition	3 axes	4 axes	3 axes

#### Axes Group Operation Settings

Use these parameters to set the items related to the axes group operation such as the maximum interpolation velocity and axes group stopping method. Set these items according to the specifications of the devices you control.

You set only the Axes Group Stop Method among these operation parameters. As for other operation parameters, their default values are used.

## 3 Setting Robot Functions

Parameter name	Setting	Set value
Maximum Interpolation Velocity	Set the maximum interpolation velocity for the trajectory.	Use the default value.
Maximum Interpolation Acceleration	Set the maximum interpolation acceleration for the trajectory.	
Maximum Interpolation Deceleration	Set the maximum interpolation deceleration for the trajectory.	
Interpolation Acceleration/Deceleration Over	Select the operation for when the maximum interpolation acceleration/deceleration may be exceeded after excessive deceleration during the axes group acceleration/deceleration control because stopping at the target position is given priority.	
Interpolation Velocity Warning Value	Set the percentage of the maximum interpolation velocity at which to output an interpolation velocity warning.	
Interpolation Acceleration Warning Value	Set the percentage of the maximum interpolation acceleration at which to output an interpolation acceleration warning.	
Interpolation Deceleration Warning Value	Set the percentage of the maximum interpolation deceleration rate at which to output an interpolation deceleration warning.	
Axes Group Stop Method	Set how to stop the composition axes that have no error when an error that forces an immediate stop occurs in an axis during a multi-axes coordinated motion.	
Correction Allowance Ratio	This parameter applies when the center designation method is used for a circular interpolation instruction. It compensates the distance when the distance between the start point and the center point does not equal the distance between the end point and the center point.	Use the default value.



### Precautions for Safe Use

Select *Immediate stop* or *Immediate stop and Servo OFF* when using Delta 3, Delta 3R, Delta 2, Cartesian 2D Gantry or Cartesian 3D Gantry robot.

### 3-1-3 Axis Settings

Use the Sysmac Studio to set the axis parameters for each axis.

When axes of the robot operates separately, each axis operation follows its **Axis Settings**. The settings of three axes must be the same.

If you use the MC\_SyncLinearConveyor (Start Conveyor Synchronization) instruction, you must also set the conveyor axes.

## Axis Basic Settings

The Axis Basic Settings are used to set whether to use the axis. If you use the axis, set the axis type and the node address of the EtherCAT slave device.

Parameter name	Setting	Set value
Axis Use	Set whether to enable or disable the axis.	Select <i>Used axis</i> .
Axis type	Set the axis type.	Select <i>Servo axis</i> or <i>Virtual servo axis</i> .
Input device	Specify the node address of the EtherCAT slave device that is assigned to the axis.	Select the node address of the servo axis.

## Unit Conversion Settings

These parameters set position units.

Parameter name	Setting	Set value
Unit of Display	Select the display unit.	Select as the following table, according to the intended kinematics type.
Command Pulse Count Per Motor Rotation	Set the number of command pulses per motor rotation according to the encoder resolution. *1	Set the value according to the robot specification.
Work Travel Distance Per Motor Rotation	Set the workpiece travel distance per motor rotation according to the machine specifications. *2	

\*1. For example, if the encoder resolution is 10,000 pulses per rotation, set 10,000.

\*2. When using H-Bot kinematics type, the radius of the pulley must be considered.

Axis in the AxesGroup	Kinematics type							
	Delta3	Delta3R	Delta2	Cartesian 3D	Cartesian 3D-Gantry	Cartesian 2D	Cartesian 2D-Gantry	H-Bot
A0	degree	degree	degree	mm	mm	mm	mm	mm
A1	degree	degree	degree	mm	mm	mm	mm	mm
A2	degree	degree	-	mm	mm	-	mm	-
A3	-	degree	-	-	mm	-	-	-

Axis in the AxesGroup	Kinematics type				
	SCARA RRP	SCARA PRR	SCARA RRP+R	SCARA PRR+R	Expansion1
A0	degree	mm	degree	mm	degree
A1	degree	degree	degree	degree	degree
A2	mm	degree	mm	degree	degree
A3	-	-	degree	-	-

## Operation Settings

These parameters set items for axis operation, such as the maximum velocity and maximum acceleration/deceleration rate. Set them according to the specifications of the device you control.

The following table shows the list of parameters used for robot control.

### 3 Setting Robot Functions

Parameter name	Setting	Set value
Maximum Velocity	Set the maximum velocity for each axis.* <sup>1</sup>	Set the value according to the robot specification.
Maximum Jog Velocity	Set the maximum jog velocity for the axis.* <sup>2</sup>	
Maximum Acceleration	Set the maximum acceleration rate for an axis operation command.	
Maximum Deceleration	Set the maximum deceleration rate for an axis operation command.* <sup>3</sup>	Set the value according to the robot specification.
Acceleration/Deceleration Over	Select the operation for when the maximum acceleration/deceleration may be exceeded after excessive deceleration during the axis acceleration/deceleration control because stopping at the target position is given priority.	
Operation Selection at Reversing	Specify the operation for reversing rotation for multi-execution/re-execution of instructions and interrupt feeding.	
Velocity Warning Value	Set the percentage of the maximum velocity at which to output a velocity warning for the axis.	
Acceleration Warning Value	Set the percentage of the maximum acceleration rate at which to output an acceleration warning for the axis.	
Deceleration Warning Value	Set the percentage of the maximum deceleration rate at which to output a deceleration warning for the axis.	
Positive Torque Warning Value	Set the torque command value at which to output a positive torque warning for the axis.	
Negative Torque Warning Value	Set the torque command value at which to output a negative torque warning for the axis.	
Actual Velocity Filter Time Constant	Set the time period in milliseconds for calculating the average travel of the actual velocity.	
In-position Range* <sup>4</sup>	Set the in-position width.	
In-position Check Time* <sup>4</sup>	Set the in-position check time in milliseconds.	
Zero Position Range	Set the home position detection width.	

\*1. The operation stops immediately if you specify a velocity command value that is greater than the maximum velocity.

\*2. 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.

\*3. When the AxesGroup for the robot is disabled during the robot moving, each Axis in the group starts deceleration-stop with its maximum deceleration independently.  
This causes when MC\_GroupDisable (Disable Axes Group) instruction is executed, when the operating mode of the CPU Unit changes to PROGRAM mode, or when MC Test Run is started.  
With the default value "0", the Axis immediately stops without deceleration ramp.

\*4. The in-position check is processed by the MC Function Module. The function in the Servo Drive is not used.



#### Precautions for Safe Use

Set "0" for Axes which are linked mechanically, when using Delta 3, Delta 3R, Delta 2, Cartesian 2D Gantry or Cartesian 3D Gantry robot.

## Other Operation Settings

These parameters set the items such as the stopping methods at input signals ON and the torque limits.

Parameter name	Setting	Set value
Immediate Stop Input Stop Method	Set the stopping method in the MC Function Module when the immediate stop input signal turns ON.	Set the value according to the robot specification.
Limit Input Stop Method	Set the stopping method in the MC Function Module when the positive limit input or negative limit input signal turns ON.	
Drive Error Reset Monitoring Time	Set the monitor time for a drive error reset.	
Maximum Positive Torque Limit	Set the maximum value of the positive torque limit.	
Maximum Negative Torque Limit	Set the maximum value of the negative torque limit.	
Immediate Stop Input Logic Inversion	Set whether to reverse the logic of the immediate stop input signal.	
Positive Limit Input Logic Inversion	Set whether to reverse the logic of the positive limit input signal.	
Negative Limit Input Logic Inversion	Set whether to reverse the logic of the negative limit input signal.	
Home Proximity Input Logic Inversion	Set whether to reverse the logic of the home proximity input signal.	

## Limit Settings

Use the following parameters to select functions for limiting the following error and for software limits.

Parameter name	Setting	Set value
Software Limits	Select the software limit function.	Set the value between -180.0 and 180.0°.
Positive Software Limit	Set the software limit in the positive direction.	
Negative Software Limit	Set the software limit in the negative direction.	
Following Error Over Value	Set the excessive following error check value.	
Following Error Warning Value	Set the following error warning check value.	

## Position Count Settings

Set the count mode for the position.

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

Parameter name	Setting	Set value
Count Mode	Set the count mode for the position.	Select <i>Linear Mode (finite length)</i> .
Encoder Type	Set the encoder type.	Set the encoder type to <i>Absolute encoder (ABS)</i> .

Set the Count Mode to "Rotary Mode" if you use the MC\_SyncLinearConveyor (Start Conveyor Synchronization) instruction. Use the unit of mm.

## Homing Settings

Set the motor operation to use to determine home.

Parameter name	Setting	Set value
Homing Method	Set the homing operation.	Select <i>Zero position preset</i> .

## 3-2 Robot Kinematics Settings

---

You need to set the following items in user program to operate robot.

- To set items such as the robot link length and workspace, use the MC\_SetKinTransform (Set Kinematics Transformation) instruction while all homes for axes are already determined.

Refer to *MC\_SetKinTransform* on page 4-6 for details on the MC\_SetKinTransform (Set Kinematics Transformation) instruction.

## 3-3 Homing

---

This section describes how to define the home of the robot.

For example, you need to take the following steps to define the home of Delta3.

- 1** Move the robot axis to the home and fix it.  
Refer to the manuals and other references related to your robot, and carry out this step.
- 2** Select **Absolute Encoder** in the Sysmac Studio and set the multiple rotation to 0.
- 3** Execute homing with the MC Test Run of the Sysmac Studio to set the current position to 0.0. During this operation, the Homing Method must be set to *Zero position preset*.  
By doing this operation, the absolute encoder compensation value, which is used to set the current position to 0.0, is saved in the NJ-series CPU Unit's non-volatile memory.  
It is also possible to set the current position to 0.0 with the MC\_Home (Homing) instruction in the user program instead of using the MC Test Run for homing.  
Refer to the *NJ/NX-series CPU Unit Motion Control User's Manual* (Cat. No. W507) for how to use the MC Test Run.
- 4** Repeat the above steps until the homes of all robot axes are determined.

## 3-4 Target Position and Direction Setting Method

This section describes how to set the positions for different types of robots and how to control the rotational axis of Delta3R.

### 3-4-1 Position Settings

The target position values set with the *Position* (Target Position [X, Y, Z, Rx, Ry, Rz]) input variables vary depending on the selected robot type.

Position		Kinematics type		
Name	Meaning	Delta3	Delta3R	Delta2
Position [0]	Target position on X axis [mm]	Used	Used	Used
Position [1]	Target position on Y axis [mm]	Used	Used	Not used
Position [2]	Target position on Z axis [mm]	Used	Used	Used
Position [3]	Target rotational position around X axis [degree]	Not used	Not used	Not used
Position [4]	Target rotational position around Y axis [degree]	Not used	Not used	Not used
Position [5]	Target rotational position around Z axis [degree]	Not used	Used	Not used

Position	Kinematics type				
	Cartesian 3D Cartesian 3D-Gantry	Cartesian 2D Plane XY    Plane XZ    Plane YZ			Cartesian 2D-Gantry H-Bot
Position [0]	Used	Used	Used	Not used	Used
Position [1]	Used	Used	Not used	Used	Used
Position [2]	Used	Not used	Used	Used	Not used
Position [3]	Not used	Not used			Not used
Position [4]	Not used	Not used			Not used
Position [5]	Not used	Not used			Not used

Position	Kinematics type		
	SCARA RRP SCARA PRR	SCARA RRP+R SCARA PRR+R	Expansion1
Position [0]	Used	Used	Used
Position [1]	Used	Used	Used
Position [2]	Used	Used	Not used
Position [3]	Not used	Not used	Not used
Position [4]	Not used	Not used	Not used
Position [5]	Not used	Used	Used

## 3 Setting Robot Functions

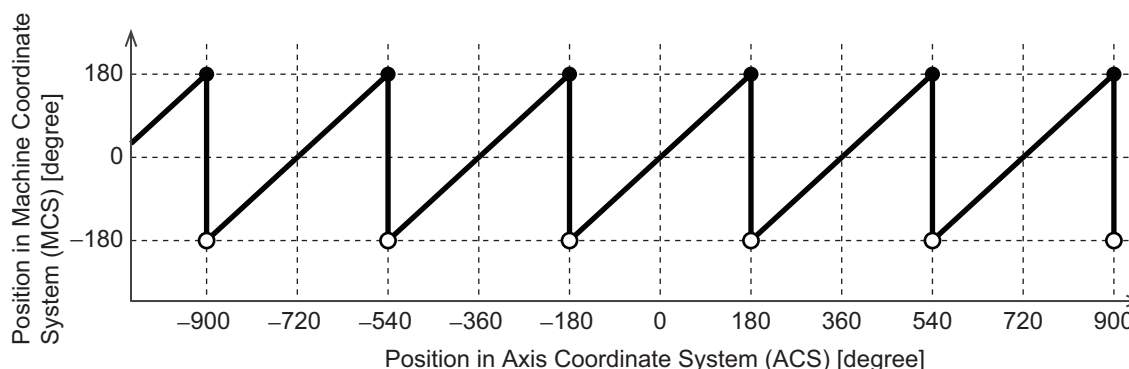
### 3-4-2 Wrist Control

For the robots with one additional axis to control the wrist, the rotational axis control is determined with the rotation (angle) around Z axis and the rotation direction.

#### Setting Rotation around Z Axis

The setting range of the rotation around Z axis is  $-180.0 < \theta \leq 180.0$ .

The following figure shows the relationship with the positions in the axis coordinate system (ACS).



The range of command position for rotation around Z axis is limited. It is the same as the range of command position for the Linear Mode in the Motion Control Function Module.

If a command position exceeds the range, an undefined home error occurs. To avoid the undefined home error, set the software limit for the axis.

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

For the robots with the additional rotational axis (A3) to control wrist, a Cartesian pose is defined by (X, Y, Z, 0, 0, Rz); position (X, Y, Z) and rotation around Z coordinate:

Joint Space (ACS)	→ Direct Kinematics →	Cartesian Space (MCS)
	← Inverse Kinematics ←	
A0 (°)		X (mm)
A1 (°)		Y (mm)
A2 (°)		Z (mm)
A3 (°)		Rz (°)

The range of the rotation around Z (Rz) of the robot is angular position within one turn; from  $-180.0^\circ$  to  $180.0^\circ$ . The lower limit ( $-180^\circ$ ) is not reachable, therefore the kinematics range of rotation around Z (Rz) is  $-180.0^\circ < Rz \leq 180.0^\circ$ .

## Setting Rotation Direction

*Direction* input specifies the direction followed by the rotational axes, which control the wrist when positioning the robot to the target position.

*Direction* input is only applicable to the kinematics that provides orientation (Rx, Ry, Rz).

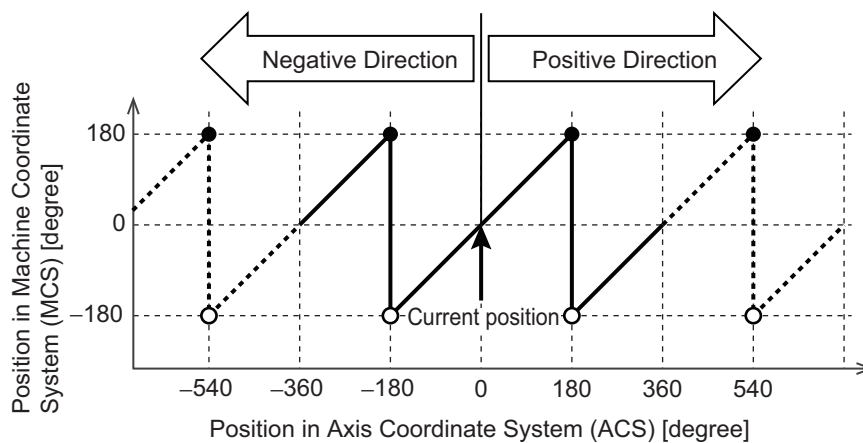
Set value	Setting Methods
Positive direction	_mcPositiveDirection
Negative direction	_mcNegativeDirection
No direction specified	_mcNoDirection

### ● **\_mcPositiveDirection**

Rotation axis is set to move in the positive direction (counter clockwise) so that the robot achieves the target position set in the Position input.

### ● **\_mcNegativeDirection**

Rotation axis is set to move in the negative direction (clockwise) so that the robot achieves the target position set in the Position input.



### 3 Setting Robot Functions

#### ● **\_mcNoDirection**

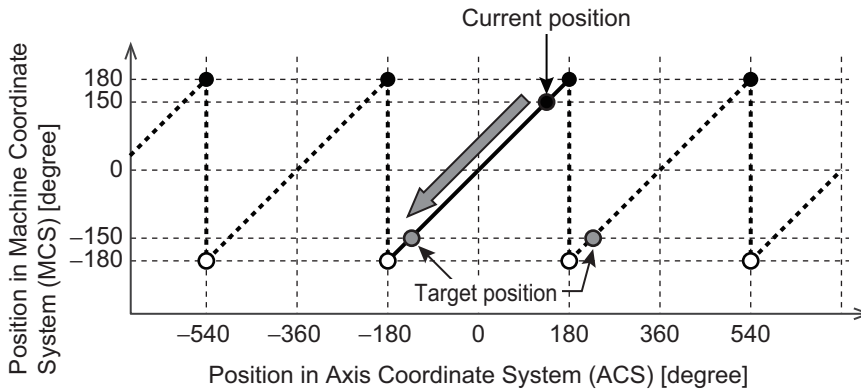
Direction of rotation axis is not set. Robot will move the rotation axis so that it achieves the target position keeping the rotation around Z (Rz) position in the same turn of its kinematics range (ring).

The robot operates in the ring which is the same as that the current position belongs.

Therefore, the robot may take a longer way.

The following is an example of settings which causes a longer way operation when controlling a Delta-3R robot.

Item	Set value
Current position	150.0
Target position	-150.0
Rotation direction	_mcNoDirection



In the same example from above, a SCARA with wrist would have the same behavior, with the only difference of the range of the Rz rotation. Instead of the  $\pm 180$  degrees, the Rz range in that case is in line with the link L2 of the robot arm in the moment of starting the movement.

For example:

Robot SCARA RRP+R with start joints in  $[\theta_1=45^\circ, \theta_2=30^\circ, \theta_3=0^\circ]$

Kinematics range	Offset (body joints)	Rz range
-180 to 180	$45^\circ+30^\circ=75^\circ$	$-105^\circ$ to $255^\circ$

Robot SCARA PRR+R with start joints in  $[\theta_1=30^\circ, \theta_2=-20^\circ, \theta_3=0^\circ]$

Kinematics range	Offset (body joints)	Rz range
-180 to 180	$30^\circ+(-20^\circ)=10^\circ$	$-170^\circ$ to $190^\circ$

Where  $\theta_1$  is the angle between the link L1 and the X axis of MCS,  $\theta_2$  is the angle between the link L2 and the link L1,  $\theta_3$  is the angle between the wrist and the link L2 axis.

## 3-5 Checking Wiring from the Sysmac Studio

This section describes how to check the wiring and electronic gear ratio setting of robot axis.

Check item	What to check
Check wiring	The wiring for devices such as limit sensors are all correct.
	If you give the axis a positive command value, the axis operates in a downward direction.
Checking electronic gear settings	The axes moves correctly according to the set angle.
	For example, 90°.

Use the MC Test Run of the Sysmac Studio or the user program to check the operation.

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

Take the following steps to check the operation.

- 1** Remove the moving frame if it is attached.  
This ensures free movement of each axis.
- 2** Check the operation.  
See the above table for check items. Refer to *Directions of Axis Coordinate System (ACS)* on page 2-14 for details on axis operation direction.
- 3** After you check that all operations are performed as shown in the above table, turn OFF the Servo for all axes. Attach the moving frame.

### 3 Setting Robot Functions



# Robot Instructions

This section describes the robot instructions.

4

<b>4-1</b>	<b>Overview of Robot Instructions</b>	<b>4-2</b>
4-1-1	List of Robot Instructions	4-2
4-1-2	Axes Group Instructions Enabled/disabled	4-2
4-1-3	Robot Execution and Stop Instructions	4-4
4-1-4	Unusable Output Variables of Axes Group	4-4
	<b>Details on Robot Instructions</b>	<b>4-5</b>
	MC_SetKinTransform	4-6
	MC_DefineCoordSystem	4-17
	MC_DefineToolTransform	4-21
	MC_GroupMon	4-25
	MC_MoveTimeAbsolute	4-31
	MC_MoveDirectAbsolute	4-41
	MC_SyncLinearConveyor	4-52
	MC_SyncOut	4-73
	MC_InverseKin	4-82
	MC_RobotJog	4-87
	MC_GroupSyncMoveAbsolute	4-96
<b>4-2</b>	<b>Multi-execution of Motion Control Instructions</b>	<b>4-98</b>
4-2-1	Multi-execution in Aborting Mode	4-98
4-2-2	Multi-execution in Buffered Mode	4-99
4-2-3	Multi-execution in Blending Mode	4-100
<b>4-3</b>	<b>State Transitions of Robot Instructions</b>	<b>4-105</b>
<b>4-4</b>	<b>Sample Programming</b>	<b>4-106</b>

## 4-1 Overview of Robot Instructions

This section gives the list of the robot instructions, and the information on the axes group instructions that you can execute with kinematics settings and on the instructions to start or stop robot operation.

### 4-1-1 List of Robot Instructions

In addition to the instructions that you can use with an NJ501-1□□□ Unit, you can use the following robot instructions with an NJ501-4□□□ Unit.

The robot instructions are categorized as the axes group instructions.

Refer to the *NJ/NX-series Motion Control Instructions Reference Manual* (Cat. No. W508) for the instructions that you can use with an NJ501-1□□□ Unit.

Classifications	Motion control instruction	
	Instruction	Name
Robot Instructions	MC_SetKinTransform	Set Kinematics Transformation
	MC_DefineCoordSystem	Define Coordinate System
	MC_DefineToolTransform	Define Tool Transformation
	MC_GroupMon	Group Monitor
	MC_MoveTimeAbsolute	Time-specified Absolute Positioning
	MC_SyncLinearConveyor	Start Conveyor Synchronization
	MC_SyncOut	End Conveyor Synchronization
	MC_InverseKin	Inverse Kinematics
	MC_RobotJog	Robot Jog
	MC_MoveDirectAbsolute	Move Direct Absolute

### 4-1-2 Axes Group Instructions Enabled/disabled

The following table shows the axes group instructions that you can execute with the kinematics settings. The axes group instructions that you cannot execute with the kinematics settings are also provided.

OK: Executable ---: An error occurs if you execute

Motion control instruction	Name	With kinematics setting	Without kinematics setting
MC_GroupEnable	Enable Axes Group	OK	OK
MC_GroupDisable	Disable Axes Group	OK	OK
MC_MoveLinear	Linear Interpolation	---	OK
MC_MoveLinearAbsolute	Absolute Linear Interpolation	---	OK
MC_MoveLinearRelative	Relative Linear Interpolation	---	OK
MC_MoveCircular2D	Circular 2D Interpolation	---	OK
MC_GroupStop	Group Stop	OK	OK
MC_GroupImmediateStop	Axes Group Immediate Stop	OK	OK
MC_GroupSetOverride	Set Group Overrides	OK <sup>*1</sup>	OK
MC_GroupReadPosition	Read Axes Group Position	OK	OK
MC_ChangeAxesInGroup	Change Axes in Group	---	OK
MC_GroupSyncMoveAbsolute	Axes Group Cyclic Synchronous Absolute Positioning	OK	OK
MC_GroupReset	Group Reset	OK	OK
MC_SetKinTransform	Set Kinematics Transformation	*2	OK
MC_DefineCoodSystem	Define Coordinate	*2	---

Motion control instruction	Name	With kinematics setting	Without kinematics setting
MC_DefineToolTransform	Define Tool Coordinate	*2	---
MC_GroupMon	Group Monitor	*2	---
MC_MoveTimeAbsolute	Time-specified Absolute Positioning	*2	---
MC_SyncLinearConveyor	Start Conveyor Synchronization	*2	---
MC_SyncOut	End Synchronization	*2	---
MC_InverseKin	Inverse Kinematics	*2	---
MC_RobotJog	Axes Group Jog	*2	---
MC_MoveDirectAbsolute	Move Direct Absolute	*2	---

\*1. You can executable the instruction, but the set value will be invalid. The robot operation is not affected.

\*2. Refer to the following table. The support of instructions depend on the setting of KinType.

Motion control instruction	Delta3	Delta3R	Delta2	Cartesian 2D 2D Gantry	Cartesian 3D 3D Gantry
MC_SetkinTransform	OK	OK	OK	OK	OK
MC_DefineCoordSystem	OK	OK	OK	OK	OK
MC_DefineToolTransform	OK	OK	OK	OK	OK
MC_GroupMon	OK	OK	OK	OK	OK
MC_MoveTimeAbsolute	OK	OK	OK	OK	OK
MC_SyncLinearConveyor	OK	OK	OK	OK	OK
MC_SyncOut	OK	OK	OK	OK	OK
MC_InverseKin	OK	OK	OK	OK	OK
MC_RobotJog	OK	OK	OK	OK	OK
MC_MoveDirectAbsolute	---	---	---	OK	OK

Motion control instruction	H-Bot T-Bot	SCARA RRP SCARA RRP+R	SCARA PRR SCARA PRR+R	Expansion1
MC_SetkinTransform	OK	OK	OK	OK
MC_DefineCoordSystem	OK	OK	OK	OK
MC_DefineToolTransform	OK	OK	OK	OK
MC_GroupMon	OK	OK	OK	OK
MC_MoveTimeAbsolute	OK	OK	OK	---
MC_SyncLinearConveyor	OK	OK	OK	---
MC_SyncOut	OK	OK	OK	---
MC_InverseKin	OK	OK	OK	OK
MC_RobotJog	OK	OK	OK	---
MC_MoveDirectAbsolute	OK	OK	OK	---



#### Precautions for Correct Use

- If you execute instructions other than MC\_GroupSetOverride (Set Group Overrides) instruction, an error occurs and the axes operation stops immediately. Refer to 4-2 *Multi-execution of Motion Control Instructions* on page 4-98 for details.
- You cannot use the MC\_WriteAxisParameter (Write Axis Parameters) instruction if the kinematics is set for the axes group.

### 4-1-3 Robot Execution and Stop Instructions

Use the following instructions to execute or stop robot operation.

#### Robot Execution Instructions

Use the following four instructions to start robot operation.

- MC\_MoveTimeAbsolute (Time-specified Absolute Positioning) instruction
- MC\_SyncLinearConveyor (Start Conveyor Synchronization) instruction
- MC\_RobotJog (Axes Group Jog) instruction
- MC\_GroupSyncMoveAbsolute (Axes Group Cyclic Synchronous Absolute Positioning) instruction
- MC\_MoveDirectAbsolute (Joint Interpolated Point-to-Point Movement) instruction

Note Not all instructions are supported by each robot type, refer to *4-1-2 Axes Group Instructions Enabled/disabled* on page 4-2.

#### Robot Stop Instructions

Use the following three instructions to stop robot operation.

- MC\_SyncOut (End Synchronization) instruction
- MC\_GroupStop (Group Stop) instruction
- MC\_GroupImmediateStop (Axes Group Immediate Stop) instruction

If you execute the MC\_GroupStop (Group Stop) instruction, the robot stops immediately regardless of the input variable settings.



#### Precautions for Correct Use

The robot operation stops if any of the following errors occurs during robot operation.

Note that the robot stops immediately in such a case.

- An error that occurs in the MC Function Module.
- An error that occurs in the EtherCAT Master Function Module.
- An error that occurs in the Built-in EtherCAT communications port (hardware).
- An error that occurs in an EtherCAT slave.

### 4-1-4 Unusable Output Variables of Axes Group

While axes operates for a robot instruction, the value of the following output variables of the axes group is 0.0.

You cannot use these variables for monitoring.

- `_MC_GRP[0-31].Cmd.Vel`
- `_MC_GRP[0-31].Cmd.AccDec`



#### Additional Information

When the robot stops for the MC\_Stop (Stop) instruction or an error, the command value for stopping operation is output to the output variable.

# Details on Robot Instructions

**4**

This section describes the robot instructions and the MC\_GroupSyncMoveAbsolute (Axes Group Cyclic Synchronous Absolute Positioning) instruction, which is a group motion instruction that you can use for robot operation.

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

# MC\_SetKinTransform

The MC\_SetKinTransform instruction makes the axes group settings required for robot control. The settings include the kinematics setting, workspace check setting, and maximum interpolation velocity setting.

Instruction	Name	FB/ FUN	Graphic expression	ST expression
MC_SetKinTransform	Set Kinematics Transformation	FB		<pre>MC_SetKinTransform_instance (   AxesGroup :=parameter,   Execute :=parameter,   KinTransform :=parameter,   Workspace :=parameter,   EnableWorkspace :=parameter,   MaxVelocityTCP :=parameter,   MaxAccelerationTCP :=parameter,   MaxDecelerationTCP :=parameter,   Done =&gt;parameter,   Busy =&gt;parameter,   CommandAborted =&gt;parameter,   Error =&gt;parameter,   ErrorID =&gt;parameter );</pre>

## Variables

This section describes the input variables, output variables and in-out variables for the MC\_SetKinTransform (Set Kinematics Transformation) instruction.

## Input Variables

Name	Meaning	Data type	Valid range	Default	Description
Execute	Execute	BOOL	TRUE or FALSE	FALSE	The instruction is executed when <i>Execute</i> changes to TRUE.
KinTransform	Kinematics Transformation	<code>_sMC_KIN_REF</code>	---	---	Set the parameters including the kinematics type and robot link length.* <sup>1</sup>
Workspace	Workspace Parameters	<code>_sMC_WORKSPACE_REF</code>	---	---	Set the range of the workspace.* <sup>2</sup>
EnableWorkspace	Enable Workspace	BOOL	TRUE or FALSE	TRUE	Select whether to enable (TRUE) or to disable (FALSE) the workspace check.
MaxVelocityTCP	Maximum Interpolation Velocity	LREAL	Non-negative number	0* <sup>3</sup>	Set a velocity for which maximum interpolation velocity errors of TCP* <sup>4</sup> are detected. (unit: mm/s)
MaxAccelerationTCP	Maximum Interpolation Acceleration	LREAL	Non-negative number	0* <sup>5</sup>	Set an acceleration rate for which maximum interpolation acceleration errors of TCP* <sup>6</sup> are detected. (unit: mm/s <sup>2</sup> )
MaxDecelerationTCP	Maximum Interpolation Deceleration	LREAL	Non-negative number	0* <sup>7</sup>	Set a deceleration rate for which maximum interpolation deceleration errors of TCP* <sup>8</sup> are detected. (unit: mm/s <sup>2</sup> )

\*1. Refer to `_sMC_KIN_REF` on page 4-7 for details.

\*2. Refer to `_sMC_WORKSPACE_REF` on page 4-8 for details.

\*3. If *MaxVelocityTCP* (Maximum Interpolation Velocity) is set to 0, interpolation velocity errors of TCP are not detected.

\*4. A maximum interpolation velocity error indicates that the maximum interpolation velocity is exceeded.

\*5. If *MaxAccelerationTCP* (Maximum Interpolation Acceleration) is set to 0, interpolation acceleration errors of TCP are not detected.

\*6. A maximum interpolation acceleration error indicates that the maximum interpolation acceleration is exceeded.

\*7. If *MaxDecelerationTCP* (Maximum Interpolation Deceleration) is set to 0, interpolation deceleration errors of TCP are not detected.

\*8. A maximum interpolation deceleration error indicates that the maximum interpolation deceleration is exceeded.

### ● `_sMC_KIN_REF`

Name	Meaning	Data type	Valid range	Description
KinType	Kinematics Type Selection	<code>_eMC_KIN_TYPE</code>	100: <code>_mcDelta3Type1</code> 101: <code>_mcDelta3Type2</code> 102: <code>_mcDelta2Type1</code> 104: <code>_mcCartesian3DType1</code> * <sup>2</sup> 105: <code>_mcCartesian3DType2</code> * <sup>2</sup> 106: <code>_mcCartesian2DType1</code> * <sup>2</sup> 107: <code>_mcCartesian2DType2</code> * <sup>2</sup> 108: <code>_mcHBotType1</code> * <sup>2</sup> 109: <code>_mcScaraRRPType1</code> * <sup>3</sup> 110: <code>_mcScaraRRPType2</code> * <sup>3</sup> 111: <code>_mcScaraPRRType1</code> * <sup>3</sup> 112: <code>_mcScaraPRRType2</code> * <sup>3</sup> 500: <code>_mcExpansion1</code> * <sup>3</sup>	Select the kinematics type. 100: Delta3 101: Delta3R 102: Delta2 104: Cartesian 3D Robot 105: Cartesian 3D Gantry Robot 106: Cartesian 2D Robot 107: Cartesian 2D Gantry Robot 108: H-Bot Robot 109: SCARA RRP Robot 110: SCARA RRP+R Robot 111: SCARA PRR Robot 112: SCARA PRR+R Robot 500: Expansion1 Robot
KinParam	Kinematics Parameter	ARRAY[0..7] OF LREAL	---	Set the robot arm length and other parameters.* <sup>1</sup>

## 4 Robot Instructions

Name	Meaning	Data type	Valid range	Description
Expansion Param	Expansion Parameter	ARRAY[0..7] OF LREAL	---	Set a value for adjusting home position of the machine coordinate system (MCS). <sup>*1</sup>

\*1. Refer to *Details on \_sMC\_KIN\_REF* on page 4-11 for details.

\*2. Those kinematics types are applicable only in Robot Version 1.03 or later.

\*3. Those kinematics types are applicable only in Robot Version 1.04 or later.

### ● **\_sMC\_WORKSPACE\_REF**

Name	Meaning	Data type	Valid range	Description
WorkspaceType	Workspace Type Selection	_eMC_ WORKSPACE_ TYPE	100: _mcDelta3 Workspace	Select the workspace type.
			101: _mcDelta2 Workspace	100: Delta3 or Delta3R
			102: _mcCartesian3Dworkspace <sup>*2</sup>	101: Delta2
			103: _mcCartesian2Dworkspace <sup>*2</sup>	102: Cartesian 3D
			104: _mcHBotWorkspace <sup>*2</sup>	103: Cartesian 2D
			105: _mcScaraWorkspace <sup>*3</sup>	104: H-Bot
WorkspaceParam	Workspace Parameters	ARRAY[0..7] OF LREAL	500: _mcExpansion1Workspace <sup>*3</sup>	105: SCARA robot volume
				500: Expansion1 robot volume
			---	Set the range of the workspace. <sup>*1</sup>

\*1. Refer to *Details on \_sMC\_WORKSPACE\_REF* on page 4-14 for details.

\*2. Those workspace types are applicable only in Robot Version 1.03 or later.

\*3. Those workspace types are applicable only in Robot Version 1.04 or later.

## Output Variables

Name	Meaning	Data type	Valid range	Description
Done	Done	BOOL	TRUE or FALSE	TRUE when the settings are completed.
Busy	Executing	BOOL	TRUE or FALSE	TRUE when the instruction is acknowledged.
CommandAborted	Command Aborted	BOOL	TRUE or FALSE	TRUE when the instruction is aborted.
Error	Error	BOOL	TRUE or FALSE	TRUE while there is an error.
ErrorID	Error Code	WORD	*1	Contains the error code when an error occurs. A value of 16#0000 indicates normal execution.

\*1. Refer to *Section 6 Troubleshooting* for details.

### ● Output Variable Update Timing

Name	Timing for changing to TRUE	Timing for changing to FALSE
Done	When the settings are completed successfully.	<ul style="list-style-type: none"> <li>When <i>Execute</i> is TRUE and changes to FALSE.</li> <li>After one period when <i>Execute</i> is FALSE.</li> </ul>
Busy	When <i>Execute</i> changes to TRUE.	<ul style="list-style-type: none"> <li>When <i>Done</i> changes to TRUE.</li> <li>When <i>Error</i> changes to TRUE.</li> </ul>
CommandAborted	<ul style="list-style-type: none"> <li>When this instruction is canceled due to an error.</li> <li>When this instruction is executed while there is an error.</li> </ul>	<ul style="list-style-type: none"> <li>When <i>Execute</i> is TRUE and changes to FALSE.</li> <li>After one period when <i>Execute</i> is FALSE.</li> </ul>
Error	When there is an error in the execution conditions or input parameters for the instruction.	When the error is cleared.

## In-Out Variables

Name	Meaning	Data type	Valid range	Description
AxesGroup	Axes Group	_sGROUP_REF	---	Specify the axes group.

## Function

- Set parameters for the kinematics, workspace check, maximum interpolation velocity check and other functions for the specified axis group.
- If you turn off the power supply to the CPU Unit, the values of kinematics, workspace check, *MaxVelocityTCP* (Maximum Interpolation Velocity), *MaxAccelerationTCP* (Maximum Interpolation Acceleration) and *MaxDecelerationTCP* (Maximum Interpolation Deceleration) that you set with this instruction are initialized. When you use the robot function, or you turn on the power supply, be sure to execute this instruction to set the kinematics, workspace check, maximum interpolation velocity and others functions.
- The values that you set are retained when the operating mode of the CPU Unit changes to PROGRAM mode.
- You can use the following robot instructions from one period after *Done* of this instruction changes to TRUE.

Instruction	Name
MC_DefineCoordSystem	Define Coordinate
MC_DefineToolTransform	Define Tool Coordinate
MC_GroupMon	Group Monitor
MC_MoveTimeAbsolute	Time-specified Absolute Positioning
MC_SyncLinearConveyor	Start Conveyor Synchronization
MC_SyncOut	End Synchronization
MC_InverseKin	Inverse Kinematics
MC_RobotJog	Axes Group Jog
MC_MoveDirectAbsolute	Joint Interpolated Point-to Point Movement

- When the axes group is disabled, the axes group instructions are also disabled even if you set the kinematics with this instruction. However, it is possible to operate each axis with single axis instructions.
- The parameters are set in the *Workspace* (Workspace Parameters) input variable even when the *EnableWorkspace* (Enable Workspace) input variable is disabled.
- If a Kinematics Initialization Error (571F hex) occurs, the kinematics parameters previously registered are deleted. When this error occurs, you must set the kinematics parameters again.
- You can use this instruction only when all the following conditions are met.
  - a) The number of axes registered in the axes group matches the number of axes of the kinematics (robot).
  - b) The display unit of each axis which is registered in the axes group is set according to the kinematics type. Refer to *Unit Conversion Settings* on page 3-5 of 3-1-3 Axis Settings.
  - c) The count mode for all axes registered in the axes group is Linear Mode.
  - d) Home for all axes registered in the axes group are already defined.
  - e) The axes group is disabled.



### Precautions for Correct Use

If you perform positioning with the MC\_GroupSyncMoveAbsolute (Axes Group Cyclic Synchronous Absolute Positioning) instruction, the workspace check is not performed. Also, checks for the following variables that are set with the MC\_SetKinTransform (Set Kinematics Transformation) instruction are not performed: *MaxVelocityTCP* (Maximum Interpolation Velocity), *MaxAccelerationTCP* (Maximum Interpolation Acceleration) and *MaxDecelerationTCP* (Maximum Interpolation Deceleration).

Consider that velocity and acceleration checks and workspace check are not performed when using MC\_GroupSyncMoveAbsolute (Axes Group Cyclic Synchronous Absolute Positioning) instruction. Therefore, make sure you take sufficient safety measures.

## Details on \_sMC\_KIN\_REF

The following are the member variables of a \_sMC\_KIN\_REF data type variable.

Refer to 2-3-1 *Kinematics Setting* on page 2-20 as well.

### ● KinType

Select the kinematics type.

Name	Data type	Valid range	Description
Kinematics Type Selection	_eMC_KIN_TYPE	100: _mcDelta3Type1	Select the kinematics type.
		101: _mcDelta3Type2	100: Delta3
		102: _mcDelta2Type1	101: Delta3R
		104: _mcCartesian3DType1 *1	102: Delta2
		105: _mcCartesian3DType2 *1	104: Cartesian 3D Robot
		106: _mcCartesian2DType1 *1	105: Cartesian 3D Gantry Robot
		107: _mcCartesian2DType2 *1	106: Cartesian 2D Robot
		108: _mcHBotType1 *1	107: Cartesian 2D Gantry Robot
		109: _mcScaraRRPType1 *2	108: H-Bot Robot
		110: _mcScaraRRPType2 *2	109: SCARA RRP Robot
		111: _mcScaraPRRType1 *2	110: SCARA RRP+R Robot
		112: _mcScaraPRRType2 *2	111: SCARA PRR Robot
		500: _mcExpansion1 *2	112: SCARA PRR+R Robot
		500: Expansion1 Robot	

\*1. Those kinematics types are applicable only in Robot Version 1.03 or later.

\*2. Those kinematics types are applicable only in Robot Version 1.04 or later.

### ● KinParam

Set the robot parameters depending on robot type.

Name	Data type	Delta3, Delta3R and Delta2	
		Valid range	Description
KinParam[0]	LREAL	Positive number	The radius of the fixed frame: Rf [mm]
KinParam[1]	LREAL	Positive number	Link 1: Lf [mm]
KinParam[2]	LREAL	Positive number	The radius of the moving frame: Rm [mm]
KinParam[3]	LREAL	Positive number	Link 2: Lm [mm]
KinParam[4] to [7] (Reserved)	LREAL	0	(Reserved)

Name	Data type	Cartesian3D and Cartesian3D Gantry	
		Valid range	Description
KinParam[0]	LREAL	Full range	The distance on X between MCS origin and TCP when all axes are 0: OffX [mm]
KinParam[1]	LREAL	Full range	The distance on Y between MCS origin and TCP when all axes are 0: OffY [mm]
KinParam[2]	LREAL	Full range	The distance on Z between MCS origin and TCP when all axes are 0: OffZ [mm]
KinParam[3] to [7] (Reserved)	LREAL	0	(Reserved)

## 4 Robot Instructions

Name	Data type	Cartesian2D	
		Valid range	Description
KinParam[0]	LREAL	Full range	The distance of the first coordinate between MCS origin and TCP when all axes are 0: Off1 [mm]
KinParam[1]	LREAL	Full range	The distance of the second coordinate between MCS origin and TCP when all axes are 0: Off2 [mm]
KinParam[2] to [7] (Reserved)	LREAL	0	(Reserved)

Name	Data type	Cartesian2D Gantry and H-Bot	
		Valid range	Description
KinParam[0]	LREAL	Full range	The distance on X between MCS origin and TCP when all axes are 0: OffX [mm]
KinParam[1]	LREAL	Full range	The distance on Y between MCS origin and TCP when all axes are 0: OffY [mm]
KinParam[2] to [7] (Reserved)	LREAL	0	(Reserved)

Name	Data type	SCARA	
		Valid range	Description
KinParam[0]	LREAL	Positive Number	L1: the first link length
KinParam[1]	LREAL	Positive Number	L2: the second link length
KinParam[2]	LREAL	Full Range	d1: distance of end effector from base in Z when axis A2=0
KinParam[3]	LREAL	0	(Reserved)
KinParam[4]	LREAL	0	(Reserved)
KinParam[5]	LREAL	0	(Reserved)
KinParam[6]	LREAL	0	(Reserved)
KinParam[7]	LREAL	0	(Reserved)

Name	Data type	Expansion1	
		Valid range	Description
KinParam[0]	LREAL	Positive Number	L1: the first link length
KinParam[1]	LREAL	Positive Number	L2: the second link length
KinParam[2]	LREAL	Positive Number	L3: the third link length
KinParam[3]	LREAL	0	(Reserved)
KinParam[4]	LREAL	0	(Reserved)
KinParam[5]	LREAL	0	(Reserved)
KinParam[6]	LREAL	0	(Reserved)
KinParam[7]	LREAL	0	(Reserved)

## ● ExpansionParam

Set the expansion parameters depending on robot type.

Name	Data type	Delta3 and Delta3R	
		Valid range	Description
ExpansionParam[0]	LREAL	0.0 to 360.0	Z axis home position: $\theta$ [degree]
ExpansionParam[1] to [7] (Reserved)	LREAL	0	(Reserved)

Name	Data type	Cartesian2D	
		Valid range	Description
ExpansionParam[0]	LREAL	[0, 1, 2]	Index to defined planes: 0: plane XY – the first coordinate is X and the second is Y 1: plane XZ – the first coordinate is X and the second is Z 2: plane YZ – the first coordinate is Y and the second is Z
ExpansionParam[1] to [7] (Reserved)	LREAL	0	(Reserved)

Name	Data type	Cartesian3D Gantry and Cartesian2D Gantry	
		Valid range	Description
ExpansionParam[0]	LREAL	-1.0 or Positive Number *1	The maximum acceptable error of commanded positions between axes A0 and A1 (absolute value)
ExpansionParam[1]	LREAL	-1.0 or Positive Number *1	The maximum acceptable error of actual positions between axes A0 and A1 (absolute value)
ExpansionParam[2] to [7] (Reserved)	LREAL	0	(Reserved)

\*1. When input is -1.0, the axes position error check is disabled.

Name	Data type	H-Bot	
		Valid range	Description
ExpansionParam[0]	LREAL	[0, 1, 2, 3]	Type of H-Bot: 0: H-Bot type 1 1: H-Bot type 2 2: T-Bot type 1 3: T-Bot type 2
ExpansionParam[1] to [7] (Reserved)	LREAL	0	(Reserved)



### Precautions for Correct Use

Robot types which are not in above tables do not use *ExpansionParam*. The values you set will be ignored.

## Details on `_sMC_WORKSPACE_REF`

The following are the member variables of a `_sMC_WORKSPACE_REF` data type variable.

Refer to *2-3-1 Kinematics Setting* on page 2-20 as well.

### ● WorkspaceType

Select the workspace type.

Name	Data type	Valid range	Description
Workspace Type Selection	<code>_eMC_WORKSPACE_TYPE</code>	100: <code>_mcDelta3</code> Workspace	Select the workspace type.
		101: <code>_mcDelta2</code> Workspace	100: Delta3 or Delta3R
		102: <code>_mcCartesian3Dworkspace</code> *1	101: Delta2
		103: <code>_mcCartesian2Dworkspace</code> *1	102: Cartesian 3D
		104: <code>_mcHBotWorkspace</code> *1	103: Cartesian 2D
		105: <code>_mcScaraWorkspace</code> *2	104: H-Bot
		500: <code>_mcExpansion1Workspace</code> *2	105: SCARA robot volume 500: Expansion1 robot volume

\*1. Those workspace types are applicable only in Robot Version 1.03 or later.

\*2. Those workspace types are applicable only in Robot Version 1.04 or later.

### ● WorkspaceParam

Set the range of the workspace.

The setting values depending on Workspace Type Selection as shown below.

Name	Data type	Delta3 or Delta3R	
		Valid range	Description
<code>WorkspaceParam[0]</code>	LREAL	Negative number	The cylinder top face position: $Z_u$ [mm]
<code>WorkspaceParam[1]</code>	LREAL	Positive number	The radius of cylinder: $R_{cy}$ [mm]
<code>WorkspaceParam[2]</code>	LREAL	Positive number	The height of cylinder: $H_{cy}$ [mm]
<code>WorkspaceParam[3]</code>	LREAL	Non-negative number	The radius of truncated cone: $R_{co}$ [mm]
<code>WorkspaceParam[4]</code>	LREAL	Non-negative number	The height of truncated cone: $H_{co}$ [mm]
<code>WorkspaceParam[5]</code> to <code>[7]</code> (Reserved)	LREAL	0	(Reserved)

Name	Data type	Delta2	
		Valid range	Description
<code>WorkspaceParam[0]</code>	LREAL	Negative number	Type trapezoid top face position: $Z_u$ [mm]
<code>WorkspaceParam[1]</code>	LREAL	Positive number	Half the length of the trapezoid top face: $R_{top}$ [mm]
<code>WorkspaceParam[2]</code>	LREAL	Non-negative number	Half the length of the trapezoid bottom face: $R_{bottom}$ [mm]
<code>WorkspaceParam[3]</code>	LREAL	Positive number	The height of trapezoid: $H$ [mm]
<code>WorkspaceParam[4]</code> to <code>[7]</code> (Reserved)	LREAL	0	(Reserved)

Name	Data type	Cartesian3D	
		Valid range	Description
<code>WorkspaceParam[0]</code>	LREAL	Full Range	Origin of the workspace: X coordinate: $X_0$ [mm]
<code>WorkspaceParam[1]</code>	LREAL	Full Range	Origin of the workspace: Y coordinate: $Y_0$ [mm]

Name	Data type	Cartesian3D	
		Valid range	Description
WorkspaceParam[2]	LREAL	Full Range	Origin of the workspace: Z coordinate: Z0 [mm]
WorkspaceParam[3]	LREAL	Positive number	Workspace dimension on the X direction: DX [mm]
WorkspaceParam[4]	LREAL	Positive number	Workspace dimension on the Y direction: DY [mm]
WorkspaceParam[5]	LREAL	Positive number	Workspace dimension on the Z direction: DZ [mm]
WorkspaceParam[6] to [7] (Reserved)	LREAL	0	(Reserved)

Name	Data type	Cartesian2D	
		Valid range	Description
WorkspaceParam[0]	LREAL	Full Range	Origin of the workspace: the first coordinate: Org1 [mm]
WorkspaceParam[1]	LREAL	Full Range	Origin of the workspace: the second coordinate: Org2 [mm]
WorkspaceParam[2]	LREAL	Positive number	Workspace dimension on the first coordinate direction: d1 [mm]
WorkspaceParam[3]	LREAL	Positive number	Workspace dimension on the second coordinate direction: d2 [mm]
WorkspaceParam[4]	LREAL	[0, 1, 2]	Index to defined planes: 0: plane XY – the first coordinate is X and the second is Y 1: plane XZ – the first coordinate is X and the second is Z 2: plane YZ – the first coordinate is Y and the second is Z
WorkspaceParam[5] to [7] (Reserved)	LREAL	0	(Reserved)

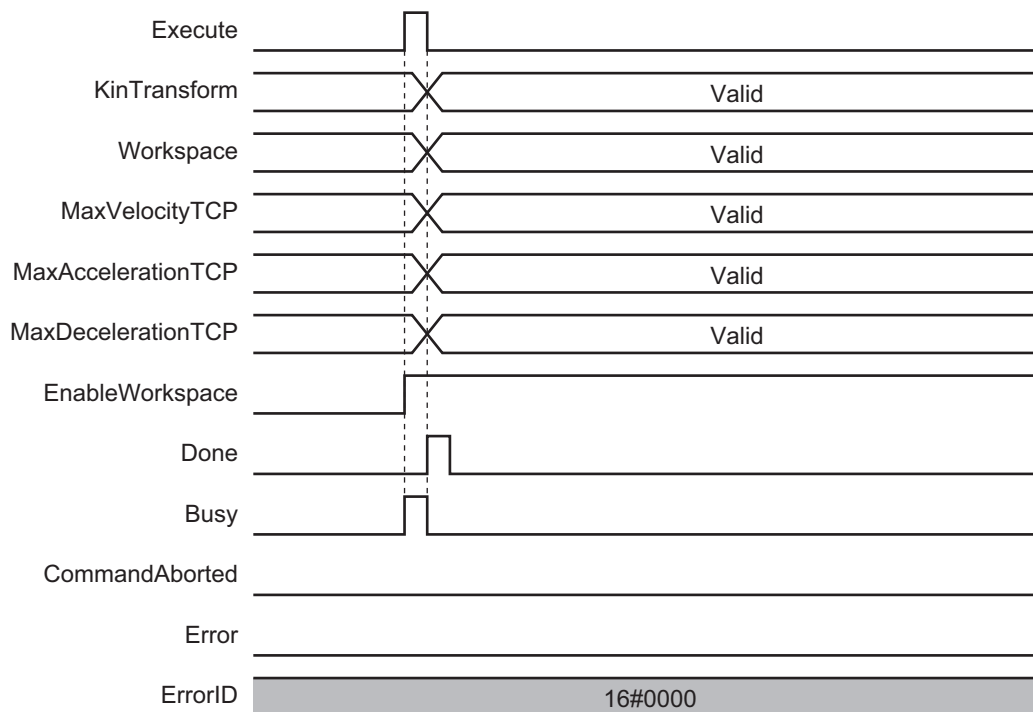
Name	Data type	H-Bot	
		Valid range	Description
WorkspaceParam[0]	LREAL	Full Range	Origin of the workspace: X coordinate: X0 [mm]
WorkspaceParam[1]	LREAL	Full Range	Origin of the workspace: Y coordinate: Y0 [mm]
WorkspaceParam[2]	LREAL	Positive number	Workspace dimension on the X direction: DX [mm]
WorkspaceParam[3]	LREAL	Positive number	Workspace dimension on the Y direction: DY [mm]
WorkspaceParam[4] to [7] (Reserved)	LREAL	0	(Reserved)

Name	Data type	SCARA	
		Valid range	Description
WorkspaceParam[0]	LREAL	Zero or Positive number	Rmin: Internal radius of the cylinder [mm]
WorkspaceParam[1]	LREAL	Positive number	Rmax: External radius of the cylinder [mm]
WorkspaceParam[2]	LREAL	Full Range	Zu: Z coordinate of the bottom part of the cylinder [mm]
WorkspaceParam[3]	LREAL	Positive number	Hcyl: Height of the cylinder [mm]
WorkspaceParam[4]	LREAL	Zero or Positive number	Darc: arc origin distance (Distance from the centre of the cylinder to the origin of the arc along X negative axis) [mm]
WorkspaceParam[5]	LREAL	[0, 180)	Aarc: arc angle [deg]
WorkspaceParam[6]	LREAL	0	(Reserved)
WorkspaceParam[7]	LREAL	0	(Reserved)

4 Robot Instructions

Name	Data type	Expansion1	
		Valid range	Description
WorkspaceParam[0]	LREAL	Zero or Positive number	Rmin: Internal radius of the circle [mm]
WorkspaceParam[1]	LREAL	Positive number	Rmax: External radius of circle [mm]
WorkspaceParam[2]	LREAL	0	(Reserved)
WorkspaceParam[3]	LREAL	0	(Reserved)
WorkspaceParam[4]	LREAL	0	(Reserved)
WorkspaceParam[5]	LREAL	0	(Reserved)
WorkspaceParam[6]	LREAL	0	(Reserved)
WorkspaceParam[7]	LREAL	0	(Reserved)

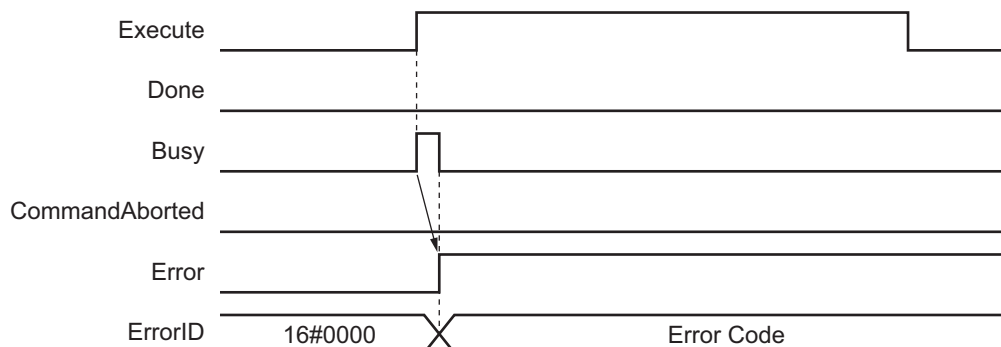
### Timing Charts



### Errors

If an error occurs during instruction execution, *Error* will change to TRUE.

You can find out the cause of the error by referring to the value output by *ErrorID* (Error Code).



# MC\_DefineCoordSystem

The MC\_DefineCoordSystem instruction sets user coordinate systems (UCS) for the specified robot (axes group).

Instruction	Name	FB/ FUN	Graphic expression	ST expression
MC_DefineCoordSystem	Define Coordinate	FB		<pre>MC_DefineCoordSystem_instance (   AxesGroup :=parameter,   Execute :=parameter,   CoordTransform :=parameter,   Done =&gt;parameter,   Busy =&gt;parameter,   CommandAborted =&gt;parameter,   Error =&gt;parameter,   ErrorID =&gt;parameter );</pre>

MC\_DefineCoordSystem

4

Variables

## Variables

This section describes the input variables, output variables and in-out variables for the MC\_DefineCoordSystem (Define Coordinate) instruction.

### Input Variables

Name	Meaning	Data type	Valid range	Default	Description
Execute	Execute	BOOL	TRUE or FALSE	FALSE	The instruction is executed when <i>Execute</i> changes to TRUE.
CoordTransform	Coordinate Transformation	_sMC_COORD- D_REF	---	---	Set the user coordinate system (UCS).

#### ● \_sMC\_COORD\_REF

Name	Meaning	Data type	Valid range	Description
CSID	Coordinate System ID	UINT	0 to 15	Select the user coordinate system (UCS) number.
Pose	Pose	ARRAY[0..5] OF LREAL	---	Set the pose of the user coordinate system (UCS). <sup>*1</sup>

\*1. Refer to *Details on \_sMC\_COORD\_REF* on page 4-19 for details.

## Output Variables

Name	Meaning	Data type	Valid range	Description
Done	Done	BOOL	TRUE or FALSE	TRUE when the settings are completed.
Busy	Executing	BOOL	TRUE or FALSE	TRUE when the instruction is acknowledged.
CommandAborted	Command Aborted	BOOL	TRUE or FALSE	TRUE when the instruction is aborted.
Error	Error	BOOL	TRUE or FALSE	TRUE while there is an error.
ErrorID	Error Code	WORD	*1	Contains the error code when an error occurs. A value of 16#0000 indicates normal execution.

\*1. Refer to *Section 6 Troubleshooting* for details.

### ● Output Variable Update Timing

Name	Timing for changing to TRUE	Timing for changing to FALSE
Done	When the settings are completed successfully.	<ul style="list-style-type: none"> <li>When <i>Execute</i> is TRUE and changes to FALSE.</li> <li>After one period when <i>Execute</i> is FALSE.</li> </ul>
Busy	When <i>Execute</i> changes to TRUE.	<ul style="list-style-type: none"> <li>When <i>Done</i> changes to TRUE.</li> <li>When <i>Error</i> changes to TRUE.</li> </ul>
CommandAborted	<ul style="list-style-type: none"> <li>When this instruction is canceled due to an error.</li> <li>When this instruction is executed while there is an error.</li> </ul>	<ul style="list-style-type: none"> <li>When <i>Execute</i> is TRUE and changes to FALSE.</li> <li>After one period when <i>Execute</i> is FALSE.</li> </ul>
Error	When there is an error in the execution conditions or input parameters for the instruction.	When the error is cleared.

## In-Out Variables

Name	Meaning	Data type	Valid range	Description
AxesGroup	Axes Group	_sGROUP_REF	---	Specify the axes group.

## Function

- This instruction sets user coordinate systems (UCS) for the specified robot (axes group).
- You can set up to 16 user coordinate systems (UCS) for each robot with *CSID* (Coordinate System ID).
- The values that you set are retained when the operating mode of the CPU Unit changes to PROGRAM mode.
- You can use the following robot instructions from one period after *Done* of this instruction changes to TRUE.

Instruction	Name
MC_GroupMon	Group Monitor
MC_MoveTimeAbsolute	Time-specified Absolute Positioning
MC_SyncLinearConveyor	Start Conveyor Synchronization
MC_RobotJog	Axes Group Jog

- You can use this instruction only when the following condition is met.  
The kinematics is already set with MC\_SetKinTransform (Set Kinematics Transformation) instruction.



### Precautions for Correct Use

User coordinate systems (UCS) are set in relation with a machine coordinate system (MCS).

If you re-execute the MC\_SetKinTransform (Set Kinematics Transformation) instruction for a robot, you must re-define all user coordinate systems for the robot.

Use the MC\_DefineCoordSystem (Define Coordinate) instruction to set user coordinate systems.

## Details on \_sMC\_COORD\_REF

The following are the member variables of a \_sMC\_COORD\_REF data type variable.

Refer to 2-3-3 *User Coordinate System (UCS)* on page 2-38 as well.

### ● CSID

Select the user coordinate system (UCS) number.

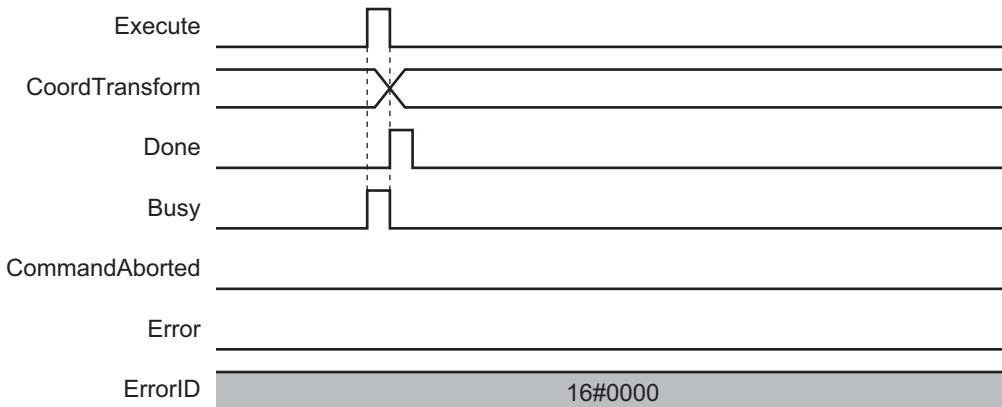
Name	Data type	Valid range	Description
Coordinate System ID	UINT	0 to 15	---

● Pose

Set the pose of the user coordinate system (UCS).

Name	Data type	Valid range	Description
Pose [0]	LREAL	Positive number, negative number or 0	The origin position of the user coordinate system viewed from the origin of X axis of the machine coordinate system: Tx [mm]
Pose [1]	LREAL	Positive number, negative number or 0	The origin position of the user coordinate system viewed from the origin of Y axis of the machine coordinate system: Ty [mm]
Pose [2]	LREAL	Positive number, negative number or 0	The origin position of the user coordinate system viewed from the origin of Z axis of the machine coordinate system: Tz [mm]
Pose [3]	LREAL	-180.0 to 180.0	The rotation around X axis of the user coordinate system: Rx [degree]
Pose [4]	LREAL	-90.0 to 90.0	The rotation around Y axis of the user coordinate system: Ry [degree]
Pose [5]	LREAL	-180.0 to 180.0	The rotation around Z axis of the user coordinate system: Rz [degree]

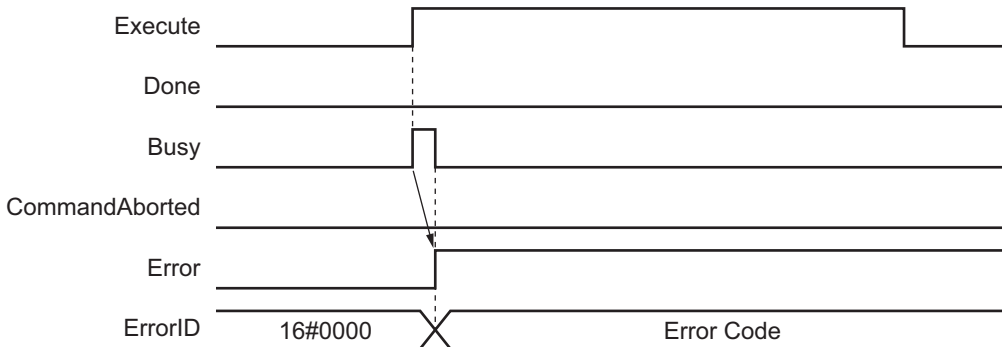
**Timing Charts**



**Errors**

If an error occurs during instruction execution, *Error* will change to TRUE.

You can find out the cause of the error by referring to the value output by *ErrorID* (Error Code).



# MC\_DefineToolTransform

The MC\_DefineToolTransform instruction sets tool coordinate systems (TCS) for the specified robot (axes group).

Instruction	Name	FB/ FUN	Graphic expression	ST expression
MC_DefineToolTransform	Define Tool Coordinate	FB		<pre>MC_DefineToolTransform_instance (   AxesGroup :=parameter,   Execute :=parameter,   ToolTransform :=parameter,   Done =&gt;parameter,   Busy =&gt;parameter,   CommandAborted =&gt;parameter,   Error =&gt;parameter,   ErrorID =&gt;parameter );</pre>

MC\_DefineToolTransform

4

Variables

## Variables

This section describes the input variables, output variables and in-out variables for the MC\_DefineToolTransform (Define Tool Coordinate) instruction.

### Input Variables

Name	Meaning	Data type	Valid range	Default	Description
Execute	Execute	BOOL	TRUE or FALSE	FALSE	The instruction is executed when <i>Execute</i> changes to TRUE.
ToolTransform	Transform Tool Coordinate System	_sMC_TOOL_COORD_REF	---	---	Make the settings of the tool coordinate system (TCS).

#### ● \_sMC\_TOOL\_COORD\_REF

Name	Meaning	Data type	Valid range	Description
ToolID	Tool ID	UINT	1 to 16	Select the ID of the tool.
Pose	Pose	ARRAY[0..5] OF LREAL	---	Set the pose of the tool coordinate system (TCS). <sup>*1</sup>

\*1. Refer to *Details on \_sMC\_TOOL\_COORD\_REF* on page 4-23 for details.

## Output Variables

Name	Meaning	Data type	Valid range	Description
Done	Done	BOOL	TRUE or FALSE	TRUE when the settings are completed.
Busy	Executing	BOOL	TRUE or FALSE	TRUE when the instruction is acknowledged.
CommandAborted	Command Aborted	BOOL	TRUE or FALSE	TRUE when the instruction is aborted.
Error	Error	BOOL	TRUE or FALSE	TRUE while there is an error.
ErrorID	Error Code	WORD	*1	Contains the error code when an error occurs. A value of 16#0000 indicates normal execution.

\*1. Refer to *Section 6 Troubleshooting* for details.

### ● Output Variable Update Timing

Name	Timing for changing to TRUE	Timing for changing to FALSE
Done	When the settings are completed successfully.	<ul style="list-style-type: none"> <li>When <i>Execute</i> is TRUE and changes to FALSE.</li> <li>After one period when <i>Execute</i> is FALSE.</li> </ul>
Busy	When <i>Execute</i> changes to TRUE.	<ul style="list-style-type: none"> <li>When <i>Done</i> changes to TRUE.</li> <li>When <i>Error</i> changes to TRUE.</li> </ul>
CommandAborted	<ul style="list-style-type: none"> <li>When this instruction is canceled due to an error.</li> <li>When this instruction is executed while there is an error.</li> </ul>	<ul style="list-style-type: none"> <li>When <i>Execute</i> is TRUE and changes to FALSE.</li> <li>After one period when <i>Execute</i> is FALSE.</li> </ul>
Error	When there is an error in the execution conditions or input parameters for the instruction.	When the error is cleared.

## In-Out Variables

Name	Meaning	Data type	Valid range	Description
AxesGroup	Axes Group	_sGROUP_REF	---	Specify the axes group.

## Function

- This instruction sets tool coordinate systems (TCS) for the specified robot (axes group).
- You can set up to 16 tool coordinate systems (TCS) for each robot with *ToolID* (Tool ID).
- The values that you set are retained when the operating mode of the CPU Unit changes to PROGRAM mode.
- You can use the set tool coordinate systems for the following robot instructions from one period after *Done* of this instruction changes to TRUE.

Instruction	Name
MC_GroupMon	Group Monitor
MC_MoveTimeAbsolute	Time-specified Absolute Positioning
MC_SyncLinearConveyor	Start Conveyor Synchronization
MC_RobotJog	Axes Group Jog

- You can use this instruction only when the following condition is met.  
The kinematics is already set with MC\_SetKinTransform (Set Kinematics Transformation) instruction.



### Precautions for Correct Use

Robot tools are set in relation with a machine coordinate system (MCS).

If you re-execute the MC\_SetKinTransform (Set Kinematics Transformation) instruction for a robot, you must re-define all robot tools for the robot.

Use the MC\_DefineToolTransform (Define Tool Coordinate) instruction to define robot tools.

## Details on \_sMC\_TOOL\_COORD\_REF

The following are the member variables of a \_sMC\_TOOL\_COORD\_REF data type variable.  
Refer to 2-3-8 Robot Tool on page 2-41 as well.

### ● ToolID

Select the ID of the tool.

Name	Data type	Valid range	Description
Tool ID	UINT	1 to 16	---

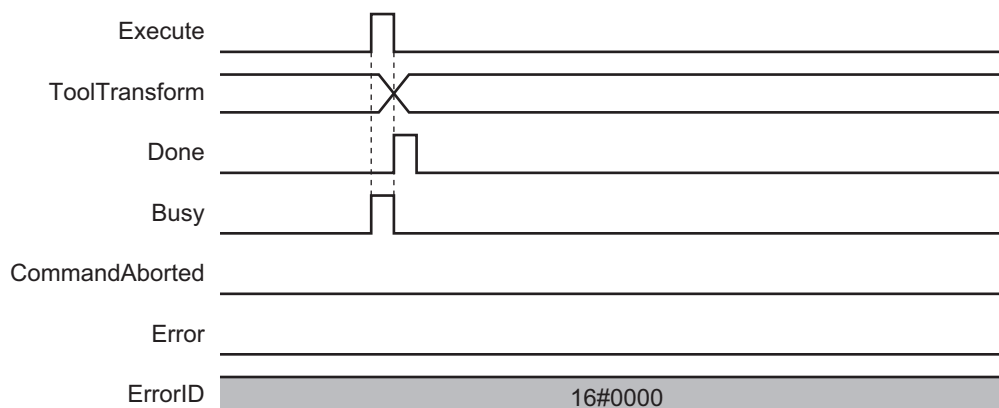
## 4 Robot Instructions

## ● Pose

Set the pose of the tool coordinate system (TCS).

Name	Data type	Valid range	Description
Pose [0]	LREAL	Positive number, negative number or 0	The origin position of the tool coordinate system viewed from the robot TCP: Tx [mm]
Pose [1]	LREAL	Positive number, negative number or 0	The origin position of the tool coordinate system viewed from the robot TCP: Ty [mm]
Pose [2]	LREAL	Positive number, negative number or 0	The origin position of the tool coordinate system viewed from the robot TCP: Tz [mm]
Pose [3]	LREAL	-180.0 to 180.0	The rotation around X axis of the tool coordinate system: Rx [degree]
Pose [4]	LREAL	-90.0 to 90.0	The rotation around Y axis of the tool coordinate system: Ry [degree]
Pose [5]	LREAL	-180.0 to 180.0	The rotation around Z axis of the tool coordinate system: Rz [degree]

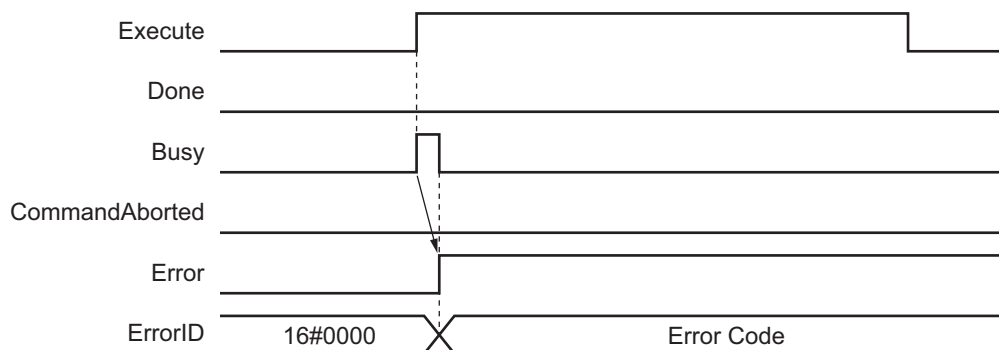
## Timing Charts



## Errors

If an error occurs during instruction execution, *Error* will change to TRUE.

You can find out the cause of the error by referring to the value output by *ErrorID* (Error Code).



# MC\_GroupMon

The MC\_GroupMon instruction reads the current position and current velocity of the robot.

Instruction	Name	FB/ FUN	Graphic expression	ST expression
MC_GroupMon	Group Monitor	FB		<pre>MC_GroupMon_instance (   AxesGroup :=parameter,   Enable :=parameter,   CoordSystem :=parameter,   CSID :=parameter,   ToolID :=parameter,   Valid =&gt;parameter,   Busy =&gt;parameter,   Error =&gt;parameter,   ErrorID =&gt;parameter,   Position =&gt;parameter,   Velocity =&gt;parameter );</pre>

MC\_GroupMon

4

Variables

## Variables

This section describes the input variables, output variables and in-out variables for the MC\_GroupMon (Group Monitor) instruction.

### Input Variables

Name	Meaning	Data type	Valid range	Default	Description
Enable	Enable	BOOL	TRUE or FALSE	FALSE	Starts reading the current value when this variable changes to TRUE.
CoordSystem	Coordinate System	_eMC_RBT_COORD_SYSTEM	1: _mcRBT_MCS 3: _mcRBT_UCS	1	Select the coordinate system. 1: Machine coordinate system (MCS) 3: User coordinate system (UCS)
CSID	Coordinate System ID	UINT	0 to 15	0	Select the user coordinate system number if you select the user coordinate system (UCS).
ToolID	Tool ID	UINT	0 to 16	0	Select the ID of the robot tool that is monitored.

## Output Variables

Name	Meaning	Data type	Valid range	Description
Valid	Enabled	BOOL	TRUE or FALSE	<i>Position</i> (Current Position) and <i>Velocity</i> (Current velocity) are valid when this variable is TRUE.
Busy	Executing	BOOL	TRUE or FALSE	TRUE when the instruction is acknowledged.
Error	Error	BOOL	TRUE or FALSE	TRUE while there is an error.
ErrorID	Error Code	WORD	*1	Contains the error code when an error occurs. A value of 16#0000 indicates normal execution.
Position	Current Position	_sMC_POSITION_REF	---	Contains the values of the command current position and actual current position. The current positions are updated each period.
Velocity	Current Velocity	_sMC_VELOCITY_REF	---	Contains the actual current velocity of TCP and the axes.

\*1. Refer to *Section 6 Troubleshooting* for details.

### ● \_sMC\_POSITION\_REF

Name	Meaning	Data type	Valid range	Description
Command Position	Command Current Position	ARRAY[0..5] OF LREAL	Positive number, negative number or 0	The command current position of TCP in the specified coordinate system.*1
Actual Position	Actual Current Position	ARRAY[0..5] OF LREAL	Positive number, negative number or 0	The actual current position of TCP in the specified coordinate system.*1

\*1. Refer to *Details on \_sMC\_POSITION\_REF* on page 4-28 for details.

### ● \_sMC\_VELOCITY\_REF

Name	Meaning	Data type	Valid range	Description
Actual VelocityTCP	Actual TCP Current Velocity	LREAL	Non-negative number	The actual current velocity of TCP.
Actual Velocity	Actual Current Velocity	ARRAY[0..5] OF LREAL	Positive number, negative number or 0	The actual current velocity of TCP on axes in the specified coordinate system.*1

\*1. Refer to *Details on \_sMC\_VELOCITY\_REF* on page 4-29 for details.

### ● Output Variable Update Timing

Name	Timing for changing to TRUE	Timing for changing to FALSE
Valid	When the current values are read successfully.	<ul style="list-style-type: none"> <li>When <i>Enable</i> changes to FALSE.</li> <li>When <i>Error</i> changes to TRUE.</li> </ul>
Busy	When <i>Execute</i> changes to TRUE.	<ul style="list-style-type: none"> <li>When <i>Done</i> changes to TRUE.</li> <li>When <i>Error</i> changes to TRUE.</li> </ul>
Error	When there is an error in the execution conditions or input parameters for the instruction.	When the error is cleared.

## In-Out Variables

Name	Meaning	Data type	Valid range	Description
AxesGroup	Axes Group	_sGROUP_REF	---	Specify the axes group.

## Function

- This instruction outputs the current position and current velocity in the selected coordinate system for the specified robot (axes group) by performing the kinematics calculation with the current position and current velocity in the axis coordinate system (ACS).
- The values of the *Position* (Current Position) and *Velocity* (Current Velocity) output variables are retained when *Enable* changes from TRUE to FALSE.
- When the *Valid* (Enabled) output variable is TRUE, *CommandPosition* (Command Current Position) varies as shown below depending on whether the axes group is enabled or disabled.

Axes group	CommandPosition (Command Current Position)
Enabled	Updates data each period.
Disabled	Retains the last data output before the axes group is disabled.

- When the *Valid* (Enabled) output variables is TRUE, the data of *ActualPosition* (Actual Current Position), *ActualVelocityTCP* (Actual TCP Current Velocity) and *ActualVelocity* (Actual Current Velocity) are updated each period regardless of whether the axes group is enabled or disabled.
- You can use this instruction only when all the following conditions are met.
  - a) The kinematics is already set with MC\_SetKinTransform (Set Kinematics Transformation) instruction.
  - b) Home for all axes registered in the axes group are already defined.
  - c) When you select the user coordinate system (UCS), you must previously set *CSID* (Coordinate System ID) used by the MC\_DefineCoordSystem (Define Coordinate) instruction.
  - d) When you use the robot tool, you must previously set *ToolID* (Tool ID) used by the MC\_DefineToolTransform (Define Tool Coordinate) instruction.

## Details on `_sMC_POSITION_REF`

The following are the member variables of a `_sMC_POSITION_REF` data type variable.

### ● CommandPosition

This variable contains the command current position of TCP in the specified coordinate system.

The setting varies depending on *KinType* (Kinematics Type Selection) as shown below.

Name	Data type	Valid range	Description
CommandPosition [0]	LREAL	Positive number, negative number or 0	The command current position on axis X [mm]
CommandPosition [1]	LREAL	Positive number, negative number or 0	The command current position on axis Y [mm]
CommandPosition [2]	LREAL	Positive number, negative number or 0	The command current position on axis Z [mm]
CommandPosition [3]	LREAL	Positive number, negative number or 0	The command current rotational position around X axis [degree]
CommandPosition [4]	LREAL	Positive number, negative number or 0	The command current rotational position around Y axis [degree]
CommandPosition [5]	LREAL	Positive number, negative number or 0	The command current rotational position around Z axis [degree]

### ● ActualPosition

This variable contains the actual current position of TCP in the specified coordinate system.

The setting varies depending on *KinType* (Kinematics Type Selection) as shown below.

Name	Data type	Valid range	Description
ActualPosition [0]	LREAL	Positive number, negative number or 0	The actual current position on X axis [mm]
ActualPosition [1]	LREAL	Positive number, negative number or 0	The actual current position on Y axis [mm]
ActualPosition [2]	LREAL	Positive number, negative number or 0	The actual current position on Z axis [mm]
ActualPosition [3]	LREAL	Positive number, negative number or 0	The actual current rotational position around X axis [degree]
ActualPosition [4]	LREAL	Positive number, negative number or 0	The actual current rotational position around Y axis [degree]
ActualPosition [5]	LREAL	Positive number, negative number or 0	The actual current rotational position around Z axis [degree]

## Details on `_SMC_VELOCITY_REF`

The following are the member variables of a `_SMC_VELOCITY_REF` data type variable.

### ● ActualVelocityTCP

This variable contains the actual current velocity of TCP.

Name	Data type	Valid range	Description	
			Delta3/Delta2	Delta3R
ActualVelocityTCP	LREAL	Non-negative number	Unit: mm/s	

The TCP velocity is obtained from the following equation.

$$V_{TCP} = \sqrt{V_x^2 + V_y^2 + V_z^2}$$

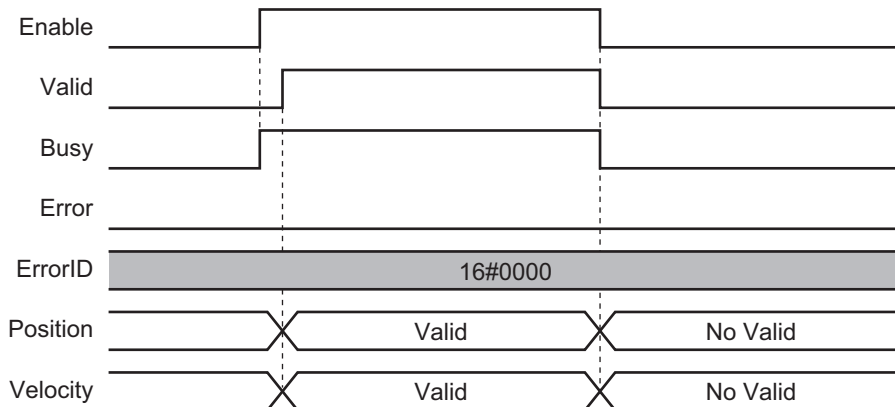
### ● ActualVelocity

This variable contains the actual current velocity of TCP on axes in the specified coordinate system.

The setting varies depending on *KinType* (Kinematics Type Selection) as shown below.

Name	Data type	Valid range	Description
ActualVelocity [0]	LREAL	Positive number, negative number or 0	The actual current velocity on X axis: Vx [mm/s]
ActualVelocity [1]	LREAL	Positive number, negative number or 0	The actual current velocity on Y axis: Vy [mm/s]
ActualVelocity [2]	LREAL	Positive number, negative number or 0	The actual current velocity on Z axis: Vz [mm/s]
ActualVelocity [3]	LREAL	Positive number, negative number or 0	The actual current rotational velocity around X axis: VRx [degree/s]
ActualVelocity [4]	LREAL	Positive number, negative number or 0	The actual current rotational velocity around Y axis: VRy [degree/s]
ActualVelocity [5]	LREAL	Positive number, negative number or 0	The actual current rotational velocity around Z axis: VRz [degree/s]

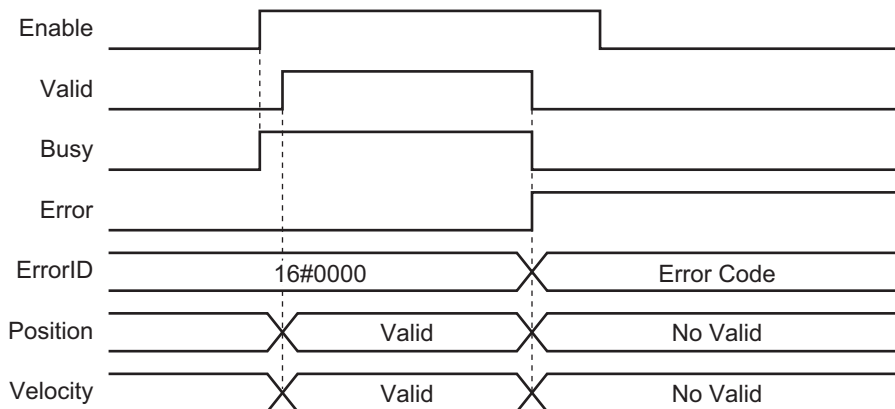
## Timing Charts



## Errors

If an error occurs during instruction execution, *Error* will change to TRUE.

You can find out the cause of the error by referring to the value output by *ErrorID* (Error Code).



# MC\_MoveTimeAbsolute

The MC\_MoveTimeAbsolute instruction moves the robot to a specified absolute target position in the specified time period.

Instruction	Name	FB/ FUN	Graphic expression	ST expression
MC_MoveTimeAbsolute	Time-specified Absolute Positioning	FB		<pre>MC_MoveTimeAbsolute_instance (   AxesGroup :=parameter,   Execute :=parameter,   Position :=parameter,   Direction :=parameter,   TrajData :=parameter,   CoordSystem :=parameter,   CSID :=parameter,   ToolID :=parameter,   BufferMode :=parameter,   TransitionMode :=parameter,   TransitionParameter :=parameter,   Done =&gt;parameter,   Busy =&gt;parameter,   Active =&gt;parameter,   CommandAborted =&gt;parameter,   Error =&gt;parameter,   ErrorID =&gt;parameter );</pre>

MC\_MoveTimeAbsolute

4

Variables

## Variables

This section describes the input variables, output variables and in-out variables for the MC\_MoveTimeAbsolute (Time-specified Absolute Positioning) instruction.

### Input Variables

Name	Meaning	Data type	Valid range	Default	Description
Execute	Execute	BOOL	TRUE or FALSE	FALSE	The instruction is executed when <i>Execute</i> changes to TRUE.
Position	Target Position	ARRAY[0..5] OF LREAL	Positive number, negative number or 0	0	Set the target position.
Direction	Direction	ARRAY[0..3] OF _eMC_DIRECTION	0: _mcPositive Direction 2: _mcNegative Direction 4: _mcNoDirection	4	Set the travel direction. 0: Positive direction 2: Negative direction 4: No direction specified
TrajData	Trajectory Data	_sMC_MOVE_TRAJ_REF	---	---	Set the parameters for creating the command value used to move the robot to a specified target position.

## 4 Robot Instructions

Name	Meaning	Data type	Valid range	Default	Description
CoordSystem	Coordinate System	_eMC_RBT_ COORD_SYSTEM	1: _mcRBT_MCS 3: _mcRBT_UCS	1	Select the coordinate system. 1: Machine coordinate system (MCS) 3: User coordinate system (UCS)
CSID	Coordinate System ID	UINT	0 to 15	0	Select the user coordinate system number if you select the user coordinate system (UCS).
ToolID	Tool ID	UINT	0 to 16	0	Select the ID of the robot tool.
BufferMode <sup>*1</sup>	Buffer Mode Selection	_eMC_BUFFER_ MODE	0: _mcAborting 1: _mcBuffered 2: _mcBlending Low 3: _mcBlending Previous	0	Select the operation that is performed when more than one motion instruction are executed. 0: Aborting 1: Buffered 2: Blending low 3: Blending previous
Transition Mode <sup>*2</sup>	Transition Mode	_eMC_ TRANSITION_ MODE	0: _mcTMNone 11: _mcTMStart Height 12: _mcTMStart RemainingTime	0	Select the trajectory of motion. 0: Transition disabled 11: Start height 12: Start remaining time
Transition Parameter	Transition Parameters	ARRAY[0..7] OF LREAL	Positive number, negative number or 0	0	Set the transition parameters. <sup>*3</sup>

\*1. Refer to *BufferMode (Buffer Mode Selection)* on page 4-38 for details.

\*2. Refer to *TransitionMode (Transition Mode)* on page 4-38 for details.

\*3. Refer to *TransitionParameter (Transition Parameters)* on page 4-39 for details.

### ● **\_sMC\_MOVE\_TRAJ\_REF**

Name	Meaning	Data type	Valid range	Description
MoveTraj Type	Motion Trajectory Type Selection	_eMC_MOVE_ TRAJ_TYPE	0: _mcPolynomial3 1: _mcRBT_ ModifiedSine	Select the trajectory type used to create the command position for each axis. 0: Polynomic 3 Curve 1: Modified Sine Curve
TrajTime	Trajectory Target Time	UINT	Positive number	Set the time for creating the command position used to move the robot from the current position to the target position. (unit: ms)
MaxVelocity	Velocity Error Detection Value	LREAL	Non-negative number	Set a velocity for which velocity errors are detected <sup>*1</sup> during command position creation. (unit: mm/s)
Max Acceleration	Acceleration Error Detection Value	LREAL	Non-negative number	Set an acceleration rate for which acceleration errors are detected <sup>*2</sup> during command position creation. (unit: mm/s <sup>2</sup> )

\*1. A velocity error indicates that the command velocity exceeded the specified value. If you set this variable to 0, velocity errors are not detected.

\*2. An acceleration error indicates that the command acceleration/deceleration rate exceeded the specified value. If you set this variable to 0, acceleration errors are not detected.

## Output Variables

Name	Meaning	Data type	Valid range	Description
Done	Done	BOOL	TRUE or FALSE	TRUE when the instruction is completed.
Busy	Executing	BOOL	TRUE or FALSE	TRUE when the instruction is acknowledged.
Active	Controlling	BOOL	TRUE or FALSE	TRUE while control is in progress.
CommandAborted	Command Aborted	BOOL	TRUE or FALSE	TRUE when the instruction is aborted.
Error	Error	BOOL	TRUE or FALSE	TRUE while there is an error.
ErrorID	Error Code	WORD	*1	Contains the error code when an error occurs. A value of 16#0000 indicates normal execution.

\*1. Refer to *Section 6 Troubleshooting* for details.

### ● Output Variable Update Timing

Name	Timing for changing to TRUE	Timing for changing to FALSE
Done	When the set target time elapsed.	<ul style="list-style-type: none"> <li>When <i>Execute</i> is TRUE and changes to FALSE.</li> <li>After one period when <i>Execute</i> is FALSE.</li> </ul>
Busy	When <i>Execute</i> changes to TRUE.	<ul style="list-style-type: none"> <li>When <i>Done</i> changes to TRUE.</li> <li>When <i>Error</i> changes to TRUE.</li> <li>When <i>CommandAborted</i> changes to TRUE.</li> </ul>
Active	When the axes move.	<ul style="list-style-type: none"> <li>When <i>Done</i> changes to TRUE.</li> <li>When <i>Error</i> changes to TRUE.</li> <li>When <i>CommandAborted</i> changes to TRUE.</li> </ul>
CommandAborted	<ul style="list-style-type: none"> <li>When this instruction is aborted because another motion control instruction was executed with the Buffer Mode set to <i>Aborting</i>.</li> <li>When this instruction is canceled due to an error.</li> <li>When this instruction is executed while there is an error.</li> <li>When you start this instruction during MC_GroupStop instruction execution.</li> </ul>	<ul style="list-style-type: none"> <li>When <i>Execute</i> is TRUE and changes to FALSE.</li> <li>After one period when <i>Execute</i> is FALSE.</li> </ul>
Error	When there is an error in the execution conditions or input parameters for the instruction.	When the error is cleared.

## In-Out Variables

Name	Meaning	Data type	Valid range	Description
AxesGroup	Axes Group	_sGROUP_REF	---	Specify the axes group.

## Function

- This instruction moves the robot (axes group) from the current position to the target position specified in the *Position* (Target Position) input variable.
- This instruction is completed when the set target time elapsed. An in-position check is not performed.
- You can use this instruction only when all the following conditions are met.
  - a) The kinematics is already set with MC\_SetKinTransform (Set Kinematics Transformation) instruction.
  - b) Home for all axes registered in the axes group are already defined.
  - c) The Servo is already ON for all axes registered in the axes group.
  - d) The limit input is OFF for all axes registered in the axes group.
  - e) The axes group is enabled.



### Precautions for Safe Use

- Adjust the home of the robot before you start robot operation.
- Do not operate the robot outside the workspace when the workspace check function is disabled. If you do so, the robot may be damaged.
- If you set 0 for the *MaxVelocity* (Velocity Error Detection Value) or *MaxAcceleration* (Acceleration Error Detection Value) parameter for input variables, or if you use the default (0) of these variables, the velocity error check or acceleration error check is not performed. Properly set *MaxVelocity* (Velocity Error Detection Value) and *MaxAcceleration* (Acceleration Error Detection Value) to avoid an unexpected velocity and acceleration.
- If a robot tool (*ToolID*: 1 to 16) other than TCP0 is selected, checks for *MaxVelocity* (Velocity Error Detection Value) and *MaxAcceleration* (Acceleration Error Detection Value) are not performed at instruction execution.

## Setting the Target Position

- For *Position* (Target Position), positions in the machine coordinate system (MCS) or positions in the user coordinate system (UCS) must be specified.
- If you select the user coordinate system (UCS), set the user coordinate system number in *CSID* (Coordinate System ID).
- The workspace check is performed if it is enabled. In case of an error, the axes group operation stops.

For details on the workspace check, refer to *2-3-2 Workspace Check* on page 2-30.

### ● Position (Target Position)

Name	Data type	Valid range	Description
Position [0]	LREAL	Positive number, negative number or 0	The target position on X axis [mm]
Position [1]	LREAL	Positive number, negative number or 0	The target position on Y axis [mm]
Position [2]	LREAL	Positive number, negative number or 0	The target position on Z axis [mm]
Position [3]	LREAL	-180.0 to 180.0	The target rotational position around X axis: Rx [degree]
Position [4]	LREAL	-180.0 to 180.0	The target rotation position around Y axis: Ry [degree]
Position [5]	LREAL	-180.0 to 180.0	The target rotational position around Z axis: Rz [degree]

The target position you can set in *Position* (Target Position) varies depending on *KinType* (Kinematics Type Selection) as shown below.

Position		Kinematics type		
Name	Description	Delta3	Delta3R	Delta2
Position [0]	The target position on X axis [mm]	Valid	Valid	Valid
Position [1]	The target position on Y axis [mm]	Valid	Valid	Invalid
Position [2]	The target position on Z axis [mm]	Valid	Valid	Valid
Position [3]	The target rotational position around X axis: Rx [degree]	Invalid	Invalid	Invalid
Position [4]	The target rotation position around Y axis: Ry [degree]	Invalid	Invalid	Invalid
Position [5]	The target rotational position around Z axis: Rz [degree]	Invalid	Valid	Invalid

Position	Kinematics type				
	Cartesian 3D Cartesian 3D-Gantry	Cartesian 2D			Cartesian 2D-Gantry H-Bot
Name		Plane XY	Plane XZ	Plane YZ	
Position [0]	Valid	Valid	Valid	Invalid	Valid
Position [1]	Valid	Valid	Invalid	Valid	Valid
Position [2]	Valid	Invalid	Valid	Valid	Invalid
Position [3]	Invalid	Invalid			Invalid
Position [4]	Invalid	Invalid			Invalid
Position [5]	Invalid	Invalid			Invalid

Position	Kinematics type	
	SCARA RRP SCARA PRR	SCARA RRP+R SCARA PRR+R
Position [0]	Valid	Valid
Position [1]	Valid	Valid
Position [2]	Valid	Valid
Position [3]	Invalid	Invalid
Position [4]	Invalid	Invalid
Position [5]	Invalid	Valid

Table above is based on machine coordinate system (MCS). When a user coordinate system is used, the specified *Position* (Target Position) is transformed into a position in the machine coordinate system. After the transformation, invalid values that are determined based on this table are ignored.

## Setting Direction

- Direction input specifies the direction followed by the wrist axes when positioning the robot to the target position.
- Direction input is only applicable to the kinematics that provides orientation (Rx, Ry, Rz).
- Direction is referred to the wrist axes, so in case there is only one wrist axis for the robot, the first element of the array will be used (e.g. Delta-3R, SCARA RRP+R and SCARA PRR+R). In case two wrist axes existed for a kinematics, the first two elements of the array would be used. Maximum number is 4, maximum axes of an Axes Group.
- For Delta-3R, SCARA RRP+R and SCARA PRR+R, the meaning of the Direction input is below:

Name	Data type	Valid range	Description
Direction [0]	_eMC_DIRECTION	0: _mcPositiveDirection 2: _mcNegativeDirection 4: _mcNoDirection	Direction of rotation of the wrist axis (e.g. A3 of Delta-3R)
Direction [1]	_eMC_DIRECTION	Full range	Reserved
Direction [2]	_eMC_DIRECTION	Full range	Reserved
Direction [3]	_eMC_DIRECTION	Full range	Reserved

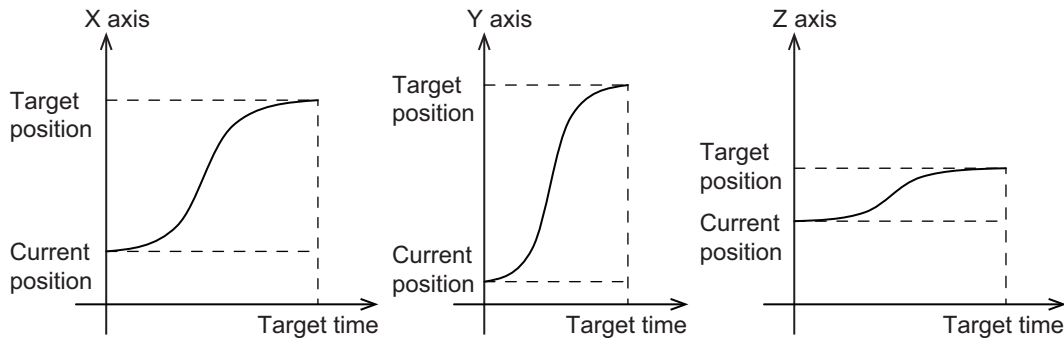
- If another kinematics is set, the Direction input is not to be used.

## Setting Trajectory

- Use *TrajData* (Trajectory Data) to set the parameters for creating the trajectory toward the target position.

*TrajData* (Trajectory Data) consists of the following variables: *MoveTrajType* (Motion Trajectory Type Selection), *TrajTime* (Trajectory Target Time), *MaxVelocity* (Velocity Error Detection Value), and *MaxAcceleration* (Acceleration Error Detection Value).

- Select the polynomial 3 curve or modified sine curve (recommended) for *MoveTrajType* (Motion Trajectory Type Selection).



- The workspace check is performed if it is enabled. If the created command position is outside the workspace, an error occurs and the axis stops.
- When this instruction is executed, the maximum values of the command velocity and command acceleration rate are created. The instruction checks whether or not these values exceed *MaxVelocity* (Velocity Error Detection Value) or *MaxAcceleration* (Acceleration Error Detection Value).  
When a command position is created, a command velocity and command acceleration rate are created at same time. These values are also checked in the same way.  
If a specified limit is exceeded, an error occurs and the axis stops.
- If *MaxVelocity* (Velocity Error Detection Value) or *MaxAcceleration* (Acceleration Error Detection Value) is set to 0, check is not performed for the variable which is set to 0.

## BufferMode (Buffer Mode Selection)

- A different instruction instance can be executed during axis motion.
- *BufferMode* specifies how to join the axis motions for this interpolation instruction and the previous interpolation instruction.
- The following Buffer Modes are supported for *BufferMode* (Buffer Mode Selection).

Buffer Mode Selection	Description
Aborting	Cancels the instruction being executed and switches to this instruction. If the direction of axis motion is reversed by switching instructions, the motion will be reversed according to the Operation Selection at Reversing axis parameter.
Buffered	Buffers this instruction and executes it automatically after the current instruction is completed.
Blending	Starts the buffered instruction at the velocity (transit velocity) at which the current instruction reaches the target position. The operation of the current instruction is changed so that the axes reach the target position at the transit velocity. There are four methods to specify the transit velocity. These are described below.
Blending low	The lower of the target velocities of the current instruction and the buffered instruction is used as the transit velocity.
Blending previous	The target velocity of the current instruction is used as the transit velocity.
Blending next	The target velocity of the buffered instruction is used as the transit velocity.
Blending high	The transit velocity is set to the target velocity of the current instruction or the buffered instruction, whichever is highest.

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

## TransitionMode (Transition Mode)

- *TransitionMode* specifies how to combine the trajectories created by the previous interpolation operation and the next interpolation operation.
- For *TransitionMode*, select *\_mcTMNone* (Transition disabled), *\_mcTMStartHeight* (Start height) or *\_mcTMStartRemainingTime* (Start remaining time).
- *TransitionMode* (Transition Mode) is valid when blending is specified for *BufferMode* (Buffer Mode Selection).
- An error will occur if you do not set *TransitionMode* to *\_mcTMNone* (Transition disabled) when blending is not used.

## TransitionParameter (Transition Parameters)

The setting of *TransitionParameter* (Transition Parameters) varies depending on whether *Transition-Mode* (Transition Mode) is set to *\_mcTMStartHeight* (Start height) or *\_mcTMStartRemainingTime* (Start remaining time).

### ● *\_mcTMStartHeight* (Start height)

Name	Data type	Valid range	Description
TransitionParameter [0]	LREAL	Positive number or 0	Transition Start Height (The height at which the transition is started) [mm]
TransitionParameter [1] to [7] (Reserved)	LREAL	0	(Reserved)

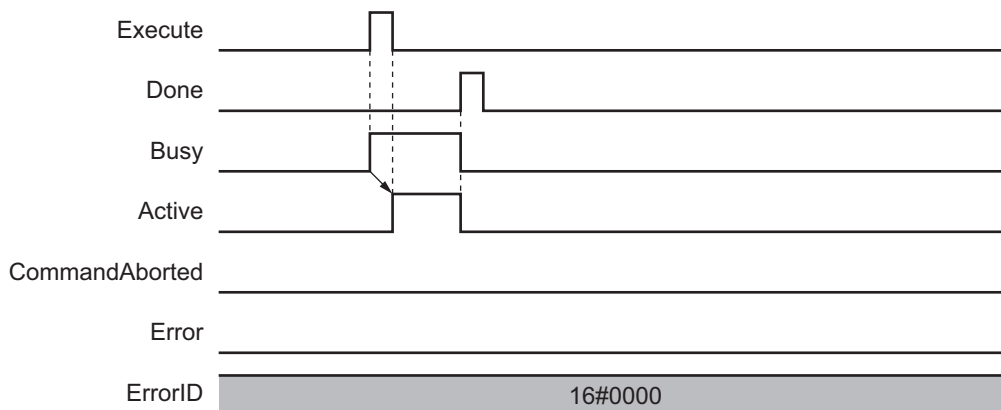
Refer to *\_mcTMStartHeight* (Start Height) on page 4-100 for details on *\_mcTMStartHeight* (Start height).

### ● *\_mcTMStartRemainingTime* (Start remaining time)

Name	Data type	Valid range	Description
TransitionParameter [0]	LREAL	Positive number or 0	Transition Start Time (The time at which the transition is started) [ms]
TransitionParameter [1] to [7] (Reserved)	LREAL	0	(Reserved)

Refer to *\_mcTMStartRemainingTime* (Start Remaining Time) on page 4-102 for details on *\_mcTMStartRemainingTime* (Start remaining time).

## Timing Charts



## Re-execution of Motion Control Instructions

This instruction cannot be re-executed.

A Motion Control Instruction Re-execution Disabled error (error code: 543B hex) occurs if re-execution is attempted, and all axes in motion stop.

## Multi-execution of Motion Control Instructions

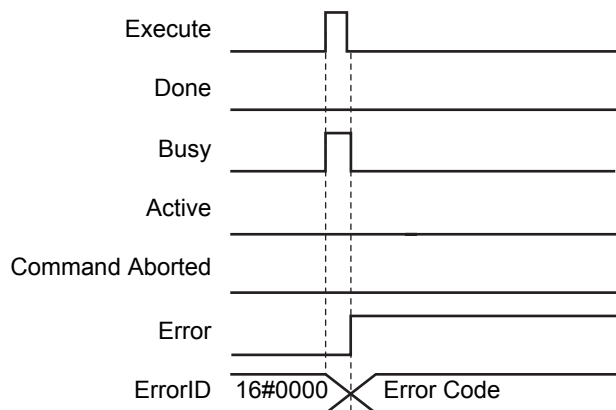
There are restrictions on the instructions that you can execute while this instruction is in execution.

Refer to *4-2 Multi-execution of Motion Control Instructions* on page 4-98 for details on multi-execution of motion control instructions.

## Errors

If an error occurs during instruction execution, *Error* will change to TRUE.

You can find out the cause of the error by referring to the value output by *ErrorID* (Error Code).



# MC\_MoveDirectAbsolute

The MC\_MoveDirectAbsolute instruction moves the robot from current to target position using Joint Interpolated Point-to-Point Movement.

Instruction	Name	FB/ FUN	Graphic expression	ST expression
MC_MoveDirectAbsolute	Point-to-Point Joint Interpolated Positioning	FB		<pre>MC_MoveDirectAbsolute_instance (   AxesGroup :=parameter,   Execute :=parameter,   Position :=parameter,   ArmConfig :=parameter,   TrajData :=parameter,   CoordSystem :=parameter,   CSID :=parameter,   ToolID :=parameter,   BufferMode :=parameter,   TransitionMode :=parameter,   TransitionParameter :=parameter,   Done =&gt;parameter,   Busy =&gt;parameter,   Active =&gt;parameter,   CommandAborted =&gt;parameter,   Error =&gt;parameter,   ErrorID =&gt;parameter );</pre>

MC\_MoveDirectAbsolute

4

Variables

## Variables

This section describes the input variables, output variables and in-out variables for the MC\_MoveDirectAbsolute (Point-to-Point Joint Interpolated Positioning) instruction.

### Input Variables

Name	Meaning	Data type	Valid range	Default	Description
Execute	Execute	BOOL	TRUE or FALSE	FALSE	The instruction is executed when <i>Execute</i> changes to TRUE.
Position	Target Position	ARRAY [0..5] OF LREAL		0	Set the target position.
ArmConfig <sup>*1</sup>	Arm Configuration	_eMC_ARM_CONFIG	0: _mcCurrentArmConfig 1: _mcRightArmConfig 2: _mcLeftArmConfig	0	Set the robot arm configuration: 0: Current configuration 1: Right-arm 2: Left-arm
TrajData	Trajectory Data	_sMC_MOVEP TP_TRAJ_REF	---	---	Set the parameters for creating the command value used to move the robot to a specified target position.

## 4 Robot Instructions

Name	Meaning	Data type	Valid range	Default	Description
CoordSystem	Coordinate System	_eM-C_RBT_COORD_SYSTEM	0: _mcRBT_ACS 1: _mcRBT_MCS 3: _mcRBT_UCS	1	Select the coordinate system. 0: Axis coordinate system (ACS) 1: Machine coordinate system (MCS) 3: User coordinate system (UCS)
CSID	Coordinate System ID	UINT	0 to 15	0	Select the user coordinate system number if you select the user coordinate system (UCS).
ToolID	Tool ID	UINT	0 to 16	0	Select the ID of the robot tool.
BufferMode <sup>*2</sup>	Buffer Mode Selection	_eM-C_BUFFER_MODE	0: _mcAborting 1: _mcBuffered	0	Select the operation that is performed when more than one motion instruction are executed. 0: Aborting 1: Buffered
Transition-Mode	Transition Mode	_eMC_TRANSITION_MODE	0: _mcTMNone	0	Select the trajectory of motion. 0: Transition disabled
Transition-Parameter	Transition Parameters	ARRAY [0..7] OF LREAL	Full Range	0	None of the parameters is used, because of Transition disabled.

\*1. *ArmConfig* will be ignored if selected *CoordSystem* is Axis Coordinate System (ACS).

\*2. Refer to *BufferMode (Buffer Mode Selection)* on page 4-50 for details.

### ● **\_sMC\_MOVEPTP\_TRAJ\_REF**

Name	Meaning	Data type	Valid range	Default	Description
PTPTraj-Type	Motion Trajectory Type Selection	_eM-C_MOVEPTP_TRAJ_TYPE	0: _mcTrapezoidal 1: _mcDoubleSType1 2: _mcDoubleSType2 3: _mcDoubleSType3 4: _mcDoubleSType4 5: _mcDoubleSType5	0	Select the trajectory type used to create the command position for each axis. 0: Trapezoidal 1: DoubleSType1 2: DoubleSType2 3: DoubleSType3 4: DoubleSType4 5: DoubleSType5
VelocityRatio	Velocity Ratio	UINT	0 to 100	0	Set percentage that will be multiplied <sup>*1</sup> with maximum velocity of each axis to calculate target velocity of each of them. (unit: %)
AccelerationRatio	Acceleration Ratio	UINT	0 to 100	0	Set percentage that will be multiplied <sup>*2</sup> with maximum acceleration of each axis to calculate target acceleration of each of them. (unit: %)
DecelerationRatio	Deceleration Ratio	UINT	0 to 100	0	Set percentage that will be multiplied <sup>*3</sup> with maximum deceleration of each axis to calculate target deceleration of each of them. (unit: %)

Name	Meaning	Data type	Valid range	Default	Description
Jerk	Jerk	LREAL	Non-negative number	0	Set the jerk to set the rate of change <sup>*4</sup> of acceleration and deceleration in time. (unit: command units/s <sup>3</sup> )
MaxVelocity	Velocity Error Detection Value	LREAL	Non-negative number	0.0	Set a velocity for which velocity errors are detected <sup>*5</sup> during command position creation. (unit: mm/s)
MaxAcceleration	Acceleration Error Detection Value	LREAL	Non-negative number	0.0	Set an acceleration rate for which acceleration errors are detected <sup>*6</sup> during command position creation. (unit: mm/s <sup>2</sup> )

\*1. Velocity Ratio Setting Out of Range error will occur if *VelocityRatio* is set to 0.

\*2. Maximum axis acceleration that is configured in the Operation settings will be used, if *AccelerationRatio* is set to 0.

\*3. Maximum axis deceleration that is configured in the Operation settings will be used, if *DecelerationRatio* is set to 0.

\*4. Trapezoidal will be used instead of DoubleS profile if *Jerk* is set to 0.

\*5. A velocity error indicates that the command velocity exceeded the specified value. If you set this variable to 0, velocity errors are not detected.

\*6. An acceleration error indicates that the command acceleration/deceleration rate exceeded the specified value. If you set this variable to 0, acceleration errors are not detected.

## Output Variables

Name	Meaning	Data type	Valid range	Description
Done	Done	BOOL	TRUE or FALSE	TRUE when the instruction is completed.
Busy	Executing	BOOL	TRUE or FALSE	TRUE when the instruction is acknowledged.
Active	Controlling	BOOL	TRUE or FALSE	TRUE while control is in progress.
CommandAborted	Command Aborted	BOOL	TRUE or FALSE	TRUE when the instruction is aborted.
Error	Error	BOOL	TRUE or FALSE	TRUE while there is an error.
ErrorID	Error Code	WORD	*1	Contains the error code when an error occurs. A value of 16#0000 indicates normal execution.

\*1. Refer to *Section 6 Troubleshooting* for details.

### ● Output Variable Update Timing

Name	Timing for changing to TRUE	Timing for changing to FALSE
Done	Timing for changing to FALSE.	<ul style="list-style-type: none"> <li>When <i>Execute</i> is TRUE and changes to FALSE.</li> <li>After one period when <i>Execute</i> is FALSE.</li> </ul>
Busy	When <i>Execute</i> changes to TRUE.	<ul style="list-style-type: none"> <li>When <i>Done</i> changes to TRUE.</li> <li>When <i>Error</i> changes to TRUE.</li> <li>When <i>CommandAborted</i> changes to TRUE.</li> </ul>
Active	When the axes move.	<ul style="list-style-type: none"> <li>When <i>Done</i> changes to TRUE.</li> <li>When <i>Error</i> changes to TRUE.</li> <li>When <i>CommandAborted</i> changes to TRUE.</li> </ul>

## 4 Robot Instructions

Name	Timing for changing to TRUE	Timing for changing to FALSE
CommandAborted	<ul style="list-style-type: none"> <li>When this instruction is aborted because another motion control instruction was executed with the Buffer Mode set to <i>Aborting</i>.</li> <li>When this instruction is canceled due to an error.</li> <li>When this instruction is executed while there is an error.</li> <li>When you start this instruction during MC_GroupStop instruction execution.</li> </ul>	<ul style="list-style-type: none"> <li>When <i>Execute</i> is TRUE and changes to FALSE.</li> <li>After one period when <i>Execute</i> is FALSE.</li> </ul>
Error	When there is an error in the execution conditions or input parameters for the instruction.	When the error is cleared.

## In-Out Variables

Name	Meaning	Data type	Valid range	Description
AxesGroup	Axes Group	_sGROUP_REF	---	Specify the axes group.

## Function

- This instruction moves the robot (axes group) from the current position to the target position specified in the *Position* (Target Position) input variable.
- This instruction is completed when robot commanded position has reached the target position. An in-position check is not performed.
- You can use this instruction only when all the following conditions are met.
  - a) The kinematics is already set with MC\_SetKinTransform (Set Kinematics Transformation) instruction.
  - b) Home for all axes registered in the axes group are already defined.
  - c) The Servo is already ON for all axes registered in the axes group.
  - d) The limit input is OFF for all axes registered in the axes group.
  - e) The axes group is enabled.



### Precautions for Safe Use

- Adjust the home of the robot before you start robot operation.
- Do not operate the robot outside the workspace when the workspace check function is disabled. If you do so, the robot may be damaged.
- If you set 0 for the *MaxVelocity* (Velocity Error Detection Value) or *MaxAcceleration* (Acceleration Error Detection Value) parameter for input variables, or if you use the default (0) of these variables, the velocity error check or acceleration error check is not performed. Properly set *MaxVelocity* (Velocity Error Detection Value) and *MaxAcceleration* (Acceleration Error Detection Value) to avoid an unexpected velocity and acceleration.
- If a robot tool (*ToolID*: 1 to 16) other than TCP0 is selected, checks for *MaxVelocity* (Velocity Error Detection Value) and *MaxAcceleration* (Acceleration Error Detection Value) are not performed at instruction execution.
- Always set the Velocity Ratio. Target Velocity Ratio Setting Out of Range error will occur if Velocity Ratio is set to 0.
- Axes Group velocity settings do not affect groups with kinematics attached.
- If Acceleration Ratio or Deceleration Ratio is set to 0 (default value), maximum axis acceleration/deceleration configured in the Operation settings is used.
- If Jerk is set to 0, Trapezoidal profile will be used.
- When working in ACS, read the actual positions directly from axis information, but not from MC\_GroupMon because the instruction works in a Cartesian coordinate system only, MCS or UCS.

## Setting the Target Position

- For *Position* (Target Position), positions in the axis coordinate system (ACS), positions in the machine coordinate system (MCS) or positions in the user coordinate system (UCS) must be specified.
- If you select the axis coordinate system (ACS), you are setting each axis position instead of working in a Cartesian coordinate system. In this way, the robot arm configuration can be freely changed. In case of using ACS, the *ArmConfig* input will be ignored.
- Robot arm configuration can be also change when working in machine coordinate system (MCS) or user coordinate system (UCS) by using the *ArmConfig* input. *ArmConfig* (Arm Configuration) selection will make the robot reaching the target position by keeping the current arm configuration or changing it to right-arm or left-arm configuration.
- If you select the user coordinate system (UCS), set the user coordinate system number in *CSID* (Coordinate System ID).
- The workspace check is performed if it is enabled. In case of an error, the axes group operation stops.  
For details on the workspace check, refer to 2-3-2 *Workspace Check* on page 2-30.

### ● Position (Target Position) in ACS (Axis Coordinate System)

Name	Data type	Valid range	Description
Position [0]	LREAL	Positive number, negative number or 0	The target position of A0 axis
Position [1]	LREAL	Positive number, negative number or 0	The target position of A1 axis
Position [2]	LREAL	Positive number, negative number or 0	The target position of A2 axis
Position [3]	LREAL	Positive number, negative number or 0	The target position of A3 axis
Position [4]			Reserved
Position [5]			Reserved

The target position you can set in *Position* (Target Position) varies depending on *KinType* (Kinematics Type Selection) as shown below.

Position		Kinematics type			
Name	Description	SCARA RRP	SCARA RRP+R	SCARA PRR	SCARA PRR+R
Position [0]	The target position of A0 axis	Valid	Valid	Valid	Valid
Position [1]	The target position of A1 axis	Valid	Valid	Valid	Valid
Position [2]	The target position of A2 axis	Valid	Valid	Valid	Valid
Position [3]	The target position of A3 axis	Invalid	Valid	Invalid	Valid
Position [4]	Reserved				
Position [5]	Reserved				

Position	Kinematics type			
Name	Cartesian 3D	Cartesian 3D-Gantry *1	Cartesian 2D (Plane XY/XZ/YZ) H-Bot	Cartesian 2D-Gantry *1
Position [0]	Valid	Valid	Valid	Valid
Position [1]	Valid	Valid	Valid	Valid
Position [2]	Valid	Valid	Invalid	Valid
Position [3]	Invalid	Valid	Invalid	Invalid
Position [4]				
Position [5]				

\*1. In case of working with a Gantry type of robot, make sure that you provide the ACS Position [0] equal to Position [1], to keep the mechanically linked axes A0 and A1 aligned.

When working in ACS, read the actual positions directly from axis information, but not from MC\_GroupMon because the instruction works in a Cartesian coordinate system only, MCS or UCS.

#### ● Position (Target Position) in MCS (Machine Coordinate System) or UCS (User Coordinate System)

Name	Data type	Valid range	Description
Position [0]	LREAL	Positive number, negative number or 0	The target position on X axis [mm]
Position [1]	LREAL	Positive number, negative number or 0	The target position on Y axis [mm]
Position [2]	LREAL	Positive number, negative number or 0	The target position on Z axis [mm]
Position [3]	LREAL	-180 to 180	The target rotational position around X axis: Rx [degree]
Position [4]	LREAL	-180 to 180	The target rotational position around Y axis: Ry [degree]
Position [5]	LREAL	-180 to 180	The target rotational position around Z axis: Rz [degree]

The target position you can set in *Position* (Target Position) varies depending on *KinType* (Kinematics Type Selection) as shown below.

Position		Kinematics type			
Name	Description	SCARA RRP	SCARA RRP+R	SCARA PRR	SCARA PRR+R
Position [0]	The target position on X axis [mm]	Valid	Valid	Valid	Valid
Position [1]	The target position on Y axis [mm]	Valid	Valid	Valid	Valid
Position [2]	The target position on Z axis [mm]	Valid	Valid	Valid	Valid
Position [3]	The target rotational position around X axis: Rx [degree]	Invalid	Invalid	Invalid	Invalid
Position [4]	The target rotational position around Y axis: Ry [degree]	Invalid	Invalid	Invalid	Invalid
Position [5]	The target rotational position around Z axis: Rz [degree]	Invalid	Valid	Invalid	Valid

Position	Kinematics type				
	Cartesian 3D Cartesian 3D-Gantry	Cartesian 2D			Cartesian 2D-Gantry H-Bot
Name		Plane XY	Plane XZ	Plane YZ	
Position [0]	Valid	Valid	Valid	Invalid	Valid
Position [1]	Valid	Valid	Invalid	Valid	Valid
Position [2]	Valid	Invalid	Valid	Valid	Invalid
Position [3]	Invalid	Invalid			Invalid
Position [4]	Invalid	Invalid			Invalid
Position [5]	Invalid	Invalid			Invalid

Tables above are based on machine coordinate system (MCS). When a user coordinate system is used, the specified *Position* (Target Position) is transformed into a position in the machine coordinate system. After the transformation, invalid values that are determined based on this table are ignored.

## Setting the Trajectory

- Use *TrajData* (Trajectory Data) to set the parameters for creating the trajectory toward the target position.  
*TrajData* (Trajectory Data) consists of the following variables: *PTPTrajType* (Move Trajectory Type), *VelocityRatio* (Axis Velocity Ratio), *AccelerationRatio* (Axis Acceleration Ratio), *DecelerationRatio* (Axis Deceleration Ratio), *Jerk* (Axis Jerk), *MaxVelocity* (Velocity Error Detection Value) and *MaxAcceleration* (Acceleration Error Detection Value).
- Trajectory of each axis is calculated by using trapezoidal or DoubleS profile (depending on value of *PTPTrajType*).
- Trapezoidal Profile is using the following parameters that are set by user: *VelocityRatio*, *AccelerationRatio* and *DecelerationRatio*. *Jerk* is not used.
- DoubleS Profile (S-Curve) is using the following parameters that are set by user: *VelocityRatio*, *AccelerationRatio*, *DecelerationRatio* and *Jerk*. Depending on *PTPTrajType*, there are 4 profile types with predefined jerk and 1 with custom jerk, see the following table:

PTP Trajectory Type	Description
_mcTrapezoidal	Trapezoidal profile
_mcDoubleSType1	DoubleS with predefined jerk 1
_mcDoubleSType2	DoubleS with predefined jerk 2
_mcDoubleSType3	DoubleS with predefined jerk 3
_mcDoubleSType4	DoubleS with predefined jerk 4
_mcDoubleSType5	DoubleS with custom jerk

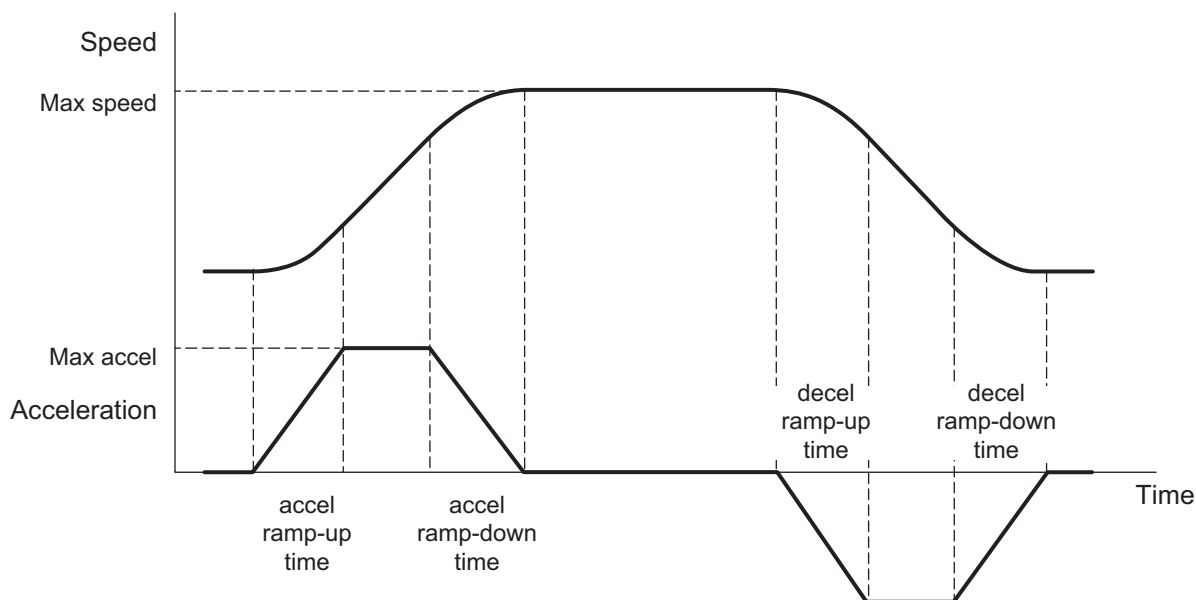
There are 4 DoubleS profile types depending on the configuration of the acceleration and deceleration ramp times. The time is the number of seconds required to ramp up or down to 100% of the acceleration or deceleration:

	Accel Ramp Up Time [s] – Tj1	Accel Ramp Down Time [s] – Tj2	Decel Ramp Up Time [s] – Tj3	Decel Ramp Up Time [s] – Tj4
DoubleS Type1	0.01	0.01	0.01	0.01
DoubleS Type2	0.03	0.03	0.03	0.03
DoubleS Type3	0.05	0.05	0.05	0.05
DoubleS Type4	0.1	0.1	0.1	0.1
DoubleS Type5	Defined by Jerk input	Defined by Jerk input	Defined by Jerk input	Defined by Jerk input

The values in DoubleS Type 5 will provide the relation between acceleration/deceleration set and jerk considering that:

$$Tj1=Tj2=Max\ acceleration/Jerk$$

$$Tj3=Tj4=Max\ deceleration/Jerk$$



- Velocity, acceleration and deceleration of each axis are calculated as explained in 2-3-14 *Joint Interpolated Point-to-Point Movement* on page 2-50.
- Axes Group velocity settings do not affect groups with kinematics attached.
- If Acceleration Ratio or Deceleration Ratio is set to 0 (default value), maximum axis acceleration/deceleration configured in the Operation settings is used.
- If Jerk is set to 0, Trapezoidal profile will be used.
- The workspace check is performed if it is enabled. If the created command position is outside the workspace, an error occurs and the axis stops.
- When a command position is created, a command velocity and command acceleration rate are created at same time. The instruction checks whether or not these values exceed *MaxVelocity* (Velocity Error Detection Value) or *MaxAcceleration* (Acceleration Error Detection Value). If a specified limit is exceeded, an error occurs and the axis stops.
- If *MaxVelocity* (Velocity Error Detection Value) or *MaxAcceleration* (Acceleration Error Detection Value) is set to 0, check is not performed for the variable which is set to 0.
- As explained in 2-3-14 *Joint Interpolated Point-to-Point Movement* on page 2-50, it is possible that trajectory parameters of an axis cannot be scaled down, so this axis will finish its movement before than the others. To avoid such situation, please see the good practices in the same chapter.

## BufferMode (Buffer Mode Selection)

- A different instruction instance can be executed during axis motion.
- BufferMode specifies how to join the axis motions for this interpolation instruction and the previous interpolation instruction.
- The following Buffer Modes are supported for *BufferMode* (Buffer Mode Selection).

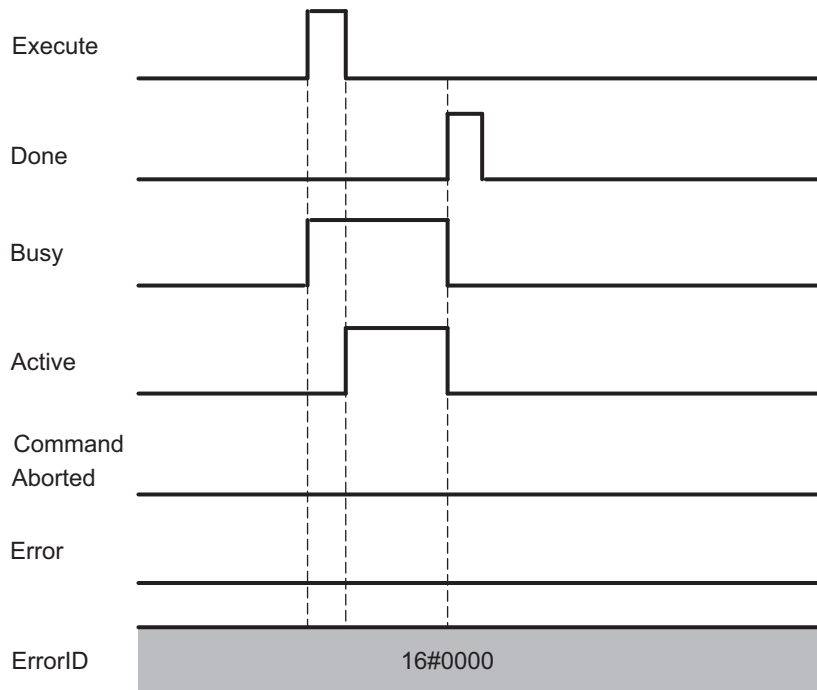
Buffer Mode Selection	Description
Aborting	Cancels the instruction being executed and switches to this instruction. If the direction of axis motion is reversed by switching instructions, the motion will be reversed according to the Operation Selection at Reversing axis parameter.
Buffered	Buffers this instruction and executes it automatically after the current instruction is completed. The instruction can only be buffered if the current instruction is also MC_MoveDirectAbsolute.
Blending	Not supported by this instruction.

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

## TransitionMode (Transition Mode)

Not supported by this instruction.

## Timing Charts



## Re-execution of Motion Control Instructions

This instruction cannot be re-executed.

A Motion Control Instruction Re-execution Disabled error (error code: 543B hex) occurs if re-execution is attempted, and all axes in motion stop.

## Multi-execution of Motion Control Instructions

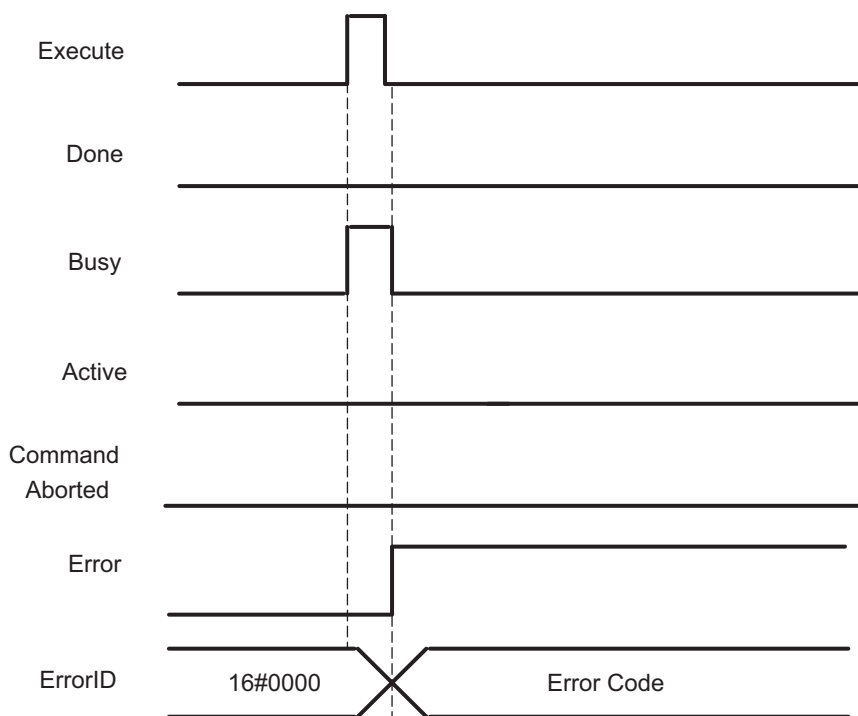
There are restrictions on the instructions that you can execute while this instruction is in execution.

Refer to 4-2 *Multi-execution of Motion Control Instructions* on page 4-98 for details on multi-execution of motion control instructions.

## Errors

If an error occurs during instruction execution, *Error* will change to TRUE.

You can find out the cause of the error by referring to the value output by *ErrorID* (Error Code).



# MC\_SyncLinearConveyor

The MC\_SyncLinearConveyor instruction moves the robot to track the specified workpiece on the conveyor. After catching up, the robot moves in synchronization with the workpiece.

Instruction	Name	FB/ FUN	Graphic expression	ST expression
MC_SyncLinearConveyor	Start Conveyor Synchronization	FB		<pre>MC_SyncLinearConveyor_instance (   AxesGroup :=parameter,   ConveyorAxis :=parameter,   Execute :=parameter,   InitWorkpiecePosition :=parameter,   Direction :=parameter,   TrajData :=parameter,   EnableOffset :=parameter,   PositionOffset :=parameter,   CoordSystem :=parameter,   CSID :=parameter,   ToolID :=parameter,   BufferMode :=parameter,   TransitionMode :=parameter,   TransitionParameter :=parameter,   InSync =&gt;parameter,   Busy =&gt;parameter,   Active =&gt;parameter,   CommandAborted =&gt;parameter,   Error =&gt;parameter,   ErrorID =&gt;parameter,   TrajStatus =&gt;parameter );</pre>

## Variables

This section describes the input variables, output variables and in-out variables for the MC\_SyncLinearConveyor (Start Conveyor Synchronization) instruction.

### Input Variables

Name	Meaning	Data type	Valid range	Default	Description
Execute	Execute	BOOL	TRUE or FALSE	FALSE	The instruction is executed when <i>Execute</i> changes to TRUE.
InitWorkpiece-Position	Initial Workpiece Position	ARRAY[0..5] OF LREAL	Positive number, negative number or 0	0	Set the target workpiece position at the point in time when synchronization with the conveyor is started.*1
Direction	Direction	ARRAY[0..3] OF _eMC_DIRECTION	0: _mcPositive Direction 2: _mcNegative Direction 4: _mcNoDirection	4	Set the travel direction. 0: Positive direction 2: Negative direction 4: No direction specified

Name	Meaning	Data type	Valid range	Default	Description
TrajData	Trajectory Data	_sMC_SYNC_ TRAJ_REF	---	---	Set the parameters for creating the command value used to move the robot to a specified target position.
EnableOffset	Offset Enabled	BOOL	TRUE or FALSE	FALSE	Enables offset value setting.
Position Offset	Offset	ARRAY[0..5] OF LREAL	Positive number, negative number or 0	0	Set the offset value for the position in the coordinate system.
CoordSystem	Coordinate System	_eMC_RBT_ COORD_SYSTEM	1: _mcRBT_MCS 3: _mcRBT_UCS	1	Select the coordinate system. 1: Machine coordinate system (MCS) 3: User coordinate system (UCS)
CSID	Coordinate System ID	UINT	0 to 15	0	Select the user coordinate system number if you select the user coordinate system (UCS).
ToolID	Tool ID	UINT	0 to 16	0	Select the ID of the robot tool.
BufferMode	Buffer Mode Selection	_eMC_BUFFER_ MODE	0: _mcAborting 1: _mcBuffered 2: _mcBlending Low 3: _mcBlending Previous 4: _mcBlending Next 5: _mcBlending High	0	Select the operation that is performed when more than one motion instruction are executed. 0: Aborting 1: Buffered 2: Blending low 3: Blending previous 4: Blending next 5: Blending high
Transition Mode <sup>*2</sup>	Transition Mode	_eMC_ TRANSITION_ MODE	0: _mcTMNone 11: _mcTMStart Height 12: _mcTMStart RemainingTime	0	Select the trajectory of motion. 0: Transition disabled 11: Start height 12: Start remaining time
Transition Parameter	Transition Parameters	ARRAY[0..7] OF LREAL	Positive number, negative number or 0	0	Set the transition parameters. <sup>*3</sup>

\*1. Refer to *Workpiece Position Setting* on page 4-58 for details.

\*2. Refer to *TransitionMode (Transition Mode)* on page 4-70 for details.

\*3. Refer to *TransitionParameter (Transition Parameters)* on page 4-71 for details.

## ● **\_sMC\_SYNC\_TRAJ\_REF**

Name	Meaning	Data type	Valid range	Description
SyncTraj Type	Synchronization Trajectory Type Selection	_eMC_SYNC_TRAJ_TYPE	0: _mcFlatTraj 2: _mcFlatTrajMod Sine 3: _mcFlatTrajMod ConstVel	Select the trajectory type that is used to create the command position for synchronization with the conveyor. 0: Polynomial 3 Curve 2: Modified Sine Curve 3: Modified Constant Velocity Curve
TrajTime	Trajectory Target Time	ARRAY [0..7] OF UINT	---	Set the time for creating the command position used to move the robot from the current position to the target position. *1
Traj Transition	Trajectory Transition	ARRAY [0..7] OF UINT	---	Set the percent of the time used for transition to the next motion. *1
Traj Distance	Trajectory Travel Distance	ARRAY[0..7] OF LREAL	Positive number, negative number or 0	Set the travel distance between the current position and the target position. *1
MaxVelocity	Velocity Error Detection Value	LREAL	Non-negative number	Set a velocity for which velocity errors are detected during command position creation. *2
Max Acceleration	Acceleration Error Detection Value	LREAL	Non-negative number	Set an acceleration rate for which acceleration errors are detected during command position creation. *3

\*1. Refer to *Details on \_sMC\_SYNC\_TRAJ\_REF* on page 4-64 for details.

\*2. A velocity error indicates that the command velocity exceeded the specified value. If you set this variable to 0, velocity errors are not detected.

\*3. An acceleration error indicates that the command acceleration/deceleration rate exceeded the specified value. If you set this variable to 0, acceleration errors are not detected.

## Output Variables

Name	Meaning	Data type	Valid range	Description
InSync	In Sync	BOOL	TRUE or FALSE	TRUE when the robot TCP catches up the workpiece on the conveyor.
Busy	Executing	BOOL	TRUE or FALSE	TRUE when the instruction is acknowledged.
Active	Controlling	BOOL	TRUE or FALSE	TRUE while control is in progress.
CommandAborted	Command Aborted	BOOL	TRUE or FALSE	TRUE when the instruction is aborted.
Error	Error	BOOL	TRUE or FALSE	TRUE while there is an error.
ErrorID	Error Code	WORD	*1	Contains the error code when an error occurs. A value of 16#0000 indicates normal execution.
TrajStatus	Trajectory Status	_sMC_SYNC_TRAJ_STATUS	---	Contains the phase and the time remaining for creating the command value.

\*1. Refer to *Section 6 Troubleshooting* for details.

### ● **\_sMC\_SYNC\_TRAJ\_STATUS**

Name	Meaning	Data type	Valid range	Description
RemainTime	Remaining Time	UINT	Non-negative number	Contains the remaining time for creating the command value.
Phase	Phase	UINT	0 to 6	Contains the command position expressed as a value of a position on the trajectory.

Note Refer to *Details on \_sMC\_SYNC\_TRAJ\_STATUS* on page 4-66 for details.

### ● **Output Variable Update Timing**

Name	Timing for changing to TRUE	Timing for changing to FALSE
InSync	When the command current position matches <sup>*1</sup> the actual current position of the workpiece after the elapse of the trajectory target time.	<ul style="list-style-type: none"> <li>When <i>Error</i> changes to TRUE.</li> <li>When <i>CommandAborted</i> changes to TRUE.</li> </ul>
Busy	When <i>Execute</i> changes to TRUE.	<ul style="list-style-type: none"> <li>When <i>Done</i> changes to TRUE.</li> <li>When <i>Error</i> changes to TRUE.</li> <li>When <i>CommandAborted</i> changes to TRUE.</li> </ul>
Active	When the axes move.	<ul style="list-style-type: none"> <li>When <i>Done</i> changes to TRUE.</li> <li>When <i>Error</i> changes to TRUE.</li> <li>When <i>CommandAborted</i> changes to TRUE.</li> </ul>
CommandAborted	<ul style="list-style-type: none"> <li>When this instruction is aborted because another motion control instruction was executed with the Buffer Mode set to <i>Aborting</i>.</li> <li>When this instruction is canceled due to an error.</li> <li>When this instruction is executed while there is an error.</li> <li>When you start this instruction during MC_Group-Stop instruction execution.</li> </ul>	<ul style="list-style-type: none"> <li>When <i>Execute</i> is TRUE and changes to FALSE.</li> <li>After one period when <i>Execute</i> is FALSE.</li> </ul>
Error	When there is an error in the execution conditions or input parameters for the instruction.	When the error is cleared.

\*1. Refer to *Setting Trajectory* on page 4-59 for details on position matching.

## In-Out Variables

Name	Meaning	Data type	Valid range	Description
AxesGroup	Axes Group	_sGROUP_REF	---	Specify the axes group.
ConveyorAxis	Conveyor Axis	_sAxis_REF	---	Specify the axis.

## Function

- The robot (axes group) moves from the current position and tracks the workpiece specified in *InitWorkpiecePosition* (Initial Workpiece Position).
- When the robot catches up the workpiece, the *InSync* (In Sync) output variable changes to TRUE. After this, synchronization is maintained.
- To cancel synchronization, use the MC\_SyncOut (End Synchronization) instruction in normal operation.
- When defining and selecting UCS for MC\_SyncLinearConveyor, Z Axis must be parallel to the MCS Z Axis because the conveyor tracking is only performed in X, Y and not in Z.
- The workspace check is performed if it is enabled. If an error is detected, the axes group operation stops.  
Refer to 2-3-2 *Workspace Check* on page 2-30 for details on the workspace.
- While the command position is created, the command velocity and command acceleration rate that are adjusted for the conveyor velocity are also created. The instruction checks whether or not these values exceed *MaxVelocity* (Velocity Error Detection Value) or *MaxAcceleration* (Acceleration Error Detection Value). If a specified limit is exceeded, an error occurs and the axis stops.
- Checks for *MaxVelocity* (Velocity Error Detection Value) and *MaxAcceleration* (Acceleration Error Detection Value) at instruction execution is performed for a trajectory for the workpiece which is not moved. For this reason, a velocity error or an acceleration error may be detected in position ranges after Phase2, depending on the conveyor velocity.
- If a Motion Control Period Exceeded error occurs, it may cause a velocity error or an acceleration error. Create the program and make the setting so that they do not cause a Motion Control Period Exceeded error.
- You can use this instruction only when all the following conditions are met.
  - a) The kinematics is already set with MC\_SetKinTransform (Set Kinematics Transformation) instruction.
  - b) Home for all axes registered in the axes group are already defined.
  - c) The Servo is already ON for all axes registered in the axes group.
  - d) The limit input is OFF for all axes registered in the axes group.
  - e) The axes group is enabled.



### Precautions for Safe Use

---

- Adjust the home of the robot before you start robot operation.
  - Do not operate the robot outside the workspace when the workspace check function is disabled. If you do so, the robot may be damaged.
  - Use the MC\_SyncOut (End Synchronization) instruction to stop the robot operated by the MC\_SyncLinearConveyor (Start Conveyor Synchronization) instruction.
  - If you set 0 for the *MaxVelocity* (Velocity Error Detection Value) or *MaxAcceleration* (Acceleration Error Detection Value) parameter for input variables, or if you use the default (0) of these variables, the velocity error check or acceleration error check is not performed. Properly set *MaxVelocity* (Velocity Error Detection Value) and *MaxAcceleration* (Acceleration Error Detection Value) to avoid an unexpected velocity and acceleration.
  - If a robot tool (*ToolID*: 1 to 16) other than TCP0 is selected, checks for *MaxVelocity* (Velocity Error Detection Value) and *MaxAcceleration* (Acceleration Error Detection Value) are not performed at instruction execution.
- 



### Additional Information

---

Velocity errors and acceleration errors may not be detected at instruction execution.

Even if velocity errors and acceleration errors are not detected at instruction execution, they may be detected while Phase 2 or Phase 4 is executed.

This is because the velocity that is detected is a blend of velocities in horizontal (X and Y axes) direction and vertical (Z axis) direction.

---

## Workpiece Position Setting

- *InitWorkpiecePosition* (Initial Workpiece Position) of the workpiece that is tracked by the robot is set in the position in the machine coordinate system (MCS) or user coordinate system (UCS).
- If you select the user coordinate system (UCS), set the user coordinate system number in *CSID* (Coordinate System ID).
- If this instruction is executed successfully, the workpiece position is updated as the conveyor axis moves, and the command position is created for the updated workpiece position.
- If the workspace check is enabled and *InitWorkpiecePosition* (Initial Workpiece Position) is set outside the workspace, an error occurs and the axis stops. In the same way, the axis stops if the current position is found outside the workspace.

### ● InitWorkpiecePosition

In this variable, set the workpiece current position at the point in time when the Conveyor Synchronization Start instruction is executed.

Name	Data type	Valid range	Description
InitWorkpiecePosition [0]	LREAL	Positive number, negative number or 0	The current position of workpiece on X axis [mm]
InitWorkpiecePosition [1]	LREAL	Positive number, negative number or 0	The current position of workpiece on Y axis [mm]
InitWorkpiecePosition [2]	LREAL	Positive number, negative number or 0	The current position of workpiece on Z axis [mm]
InitWorkpiecePosition [3]	LREAL	-180.0 to 180.0	The current rotational position of workpiece around X axis [degree]
InitWorkpiecePosition [4]	LREAL	-180.0 to 180.0	The current rotational position of workpiece around Y axis [degree]
InitWorkpiecePosition [5]	LREAL	-180.0 to 180.0	The current rotational position of workpiece around Z axis [degree]

The current position you can set in *InitWorkpiecePosition* (Initial Workpiece Position) varies depending on *KinType* (Kinematics Type Selection) as shown below.

Name	Position Description	Kinematics type		
		Delta3	Delta3R	Delta2
InitWorkpiecePosition [0]	The current position of workpiece on X axis [mm]	Valid	Valid	Valid
InitWorkpiecePosition [1]	The current position of workpiece on Y axis [mm]	Valid	Valid	Invalid
InitWorkpiecePosition [2]	The current position of workpiece on Z axis [mm]	Valid	Valid	Valid
InitWorkpiecePosition [3]	The current rotational position of workpiece around X axis [degree]	Invalid	Invalid	Invalid
InitWorkpiecePosition [4]	The current rotational position of workpiece around Y axis [degree]	Invalid	Invalid	Invalid
InitWorkpiecePosition [5]	The current rotational position of workpiece around Z axis [degree]	Invalid	Valid	Invalid

Position	Kinematics type				
Name	Cartesian 3D Cartesian 3D-Gantry	Cartesian 2D			Cartesian 2D-Gantry H-Bot
		Plane XY	Plane XZ	Plane YZ	
InitWorkpiecePosition [0]	Valid	Valid	Valid	Invalid	Valid
InitWorkpiecePosition [1]	Valid	Valid	Invalid	Valid	Valid
InitWorkpiecePosition [2]	Valid	Invalid	Valid	Valid	Invalid
InitWorkpiecePosition [3]	Invalid	Invalid			Invalid
InitWorkpiecePosition [4]	Invalid	Invalid			Invalid
InitWorkpiecePosition [5]	Invalid	Invalid			Invalid

Position	Kinematics type	
Name	SCARA RRP SCARA PRR	SCARA RRP+R SCARA PRR+R
	InitWorkpiecePosition [0]	Valid
InitWorkpiecePosition [1]	Valid	Valid
InitWorkpiecePosition [2]	Valid	Valid
InitWorkpiecePosition [3]	Invalid	Invalid
InitWorkpiecePosition [4]	Invalid	Invalid
InitWorkpiecePosition [5]	Invalid	Valid

Table above is based on machine coordinate system (MCS). When a user coordinate system is used, the specified *InitWorkpiecePosition* (Initial Workpiece Position) is transformed into a position in the machine coordinate system. After the transformation, invalid values that are determined based on this table are ignored.

## Setting Direction

- Direction input specifies the direction followed by the rotational axes when positioning the robot to the target position.
- Direction input is only applicable to the kinematics that provides orientation (Rx, Ry, Rz).
- Direction is referred to the rotational axes, so in case there is only one rotational axis for the robot, the first element of the array will be used (Delta-3R). In case two rotational axes existed for a kinematics, the first two elements of the array would be used. Maximum number is 4, maximum axes of an Axes Group.
- If other kinematics different than Delta-3R is set, Direction input is not used.

For Delta-3R meaning of Direction input is below:

Name	Data type	Valid range	Description
Direction [0]	_eMC_DIRECTION	0: _mcPositiveDirection 2: _mcNegativeDirection 4: _mcNoDirection	Direction of rotation of the rotational axis for Delta-3R (A3 of Delta-3R)
Direction [1]	_eMC_DIRECTION	Full range	Reserved
Direction [2]	_eMC_DIRECTION	Full range	Reserved
Direction [3]	_eMC_DIRECTION	Full range	Reserved

## Setting Trajectory

- Use *TrajData* (Trajectory Data) to set the parameters for creating the trajectory along which the robot tracks the workpiece.

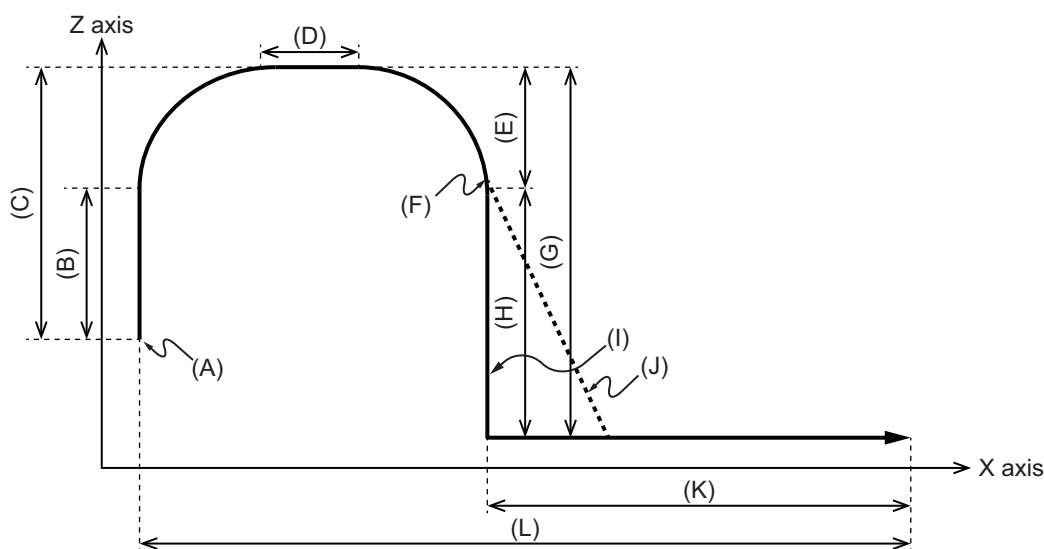
## 4 Robot Instructions

This section describes the relationship between the trajectory and the setting values for a polynomial 3 curve.

The following symbols are used in this description.

Symbol	Description
T1	Trajectory target time (unit: ms)
T2	
T3	
Ttrans1	Trajectory transition (unit: %)
Ttrans2	
H1	The travel distance on the trajectory in Z axis direction in the machine coordinate system (unit: mm)

● If  $H1 > 0.0$  and  $T1 > 0$ :



Symbol	Description
(A)	The current position of the robot
(B)	$T1 \times Ttrans1$
(C)	$T1, H1$
(D)	$T2$
(E)	$T3 - T3 \times Ttrans2$
(F)	<i>InSync</i> changes to TRUE when the position of the robot and the position of the workpiece match.
(G)	$T3$
(H)	$T3 \times Ttrans2$
(I)	The trajectory for the workpiece which is not moved.
(J)	The trajectory for the workpiece which is in motion.
(K)	The range in which <i>InSync</i> is TRUE.
(L)	The range of MC_SyncLinearConveyor (Start Conveyor Synchronization) instruction execution.

- 1** The instruction creates command positions according to which the robot moves away from the current position and reaches the (current position + H1) position on Z axis after the elapse of T1.
- 2** After the elapse of  $(T1 \times Ttrans1)$ , the instruction creates command positions according to which the robot tracks the workpiece.  
At this moment, the robot does not stop.

**3** After the elapse of  $(T1 + T2)$ , the instruction creates command positions according to which the robot moves down along Z axis.

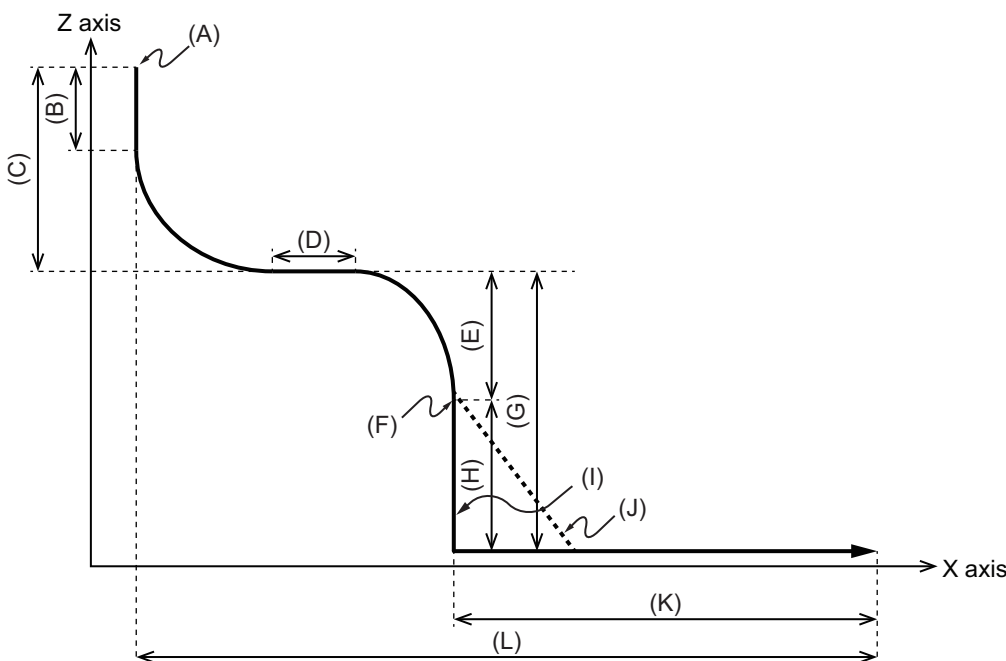
At this moment, the robot does not stop.

**4** After the elapse of  $(T1 + T2 + (T3 - T3 \times Ttrans2))$ , the instruction compares the command position  $(X, Y)$  and the actual current position  $(X, Y)$  of the workpiece.

If the difference is within the following range, *InSync* changes to TRUE. If the difference exceeds the following range, an error occurs and the axis stops. At this moment, the Workpiece Synchronization Excessive Following error (error code: 6702 hex) occurs.

Motion control period	Range of difference
500 $\mu$ s	0.1 mm max.
1 ms	0.1 mm max.
2 ms	0.2 mm max.
4 ms	0.4 mm max.

● If  $H1 < 0.0$  and  $T1 > 0$ :



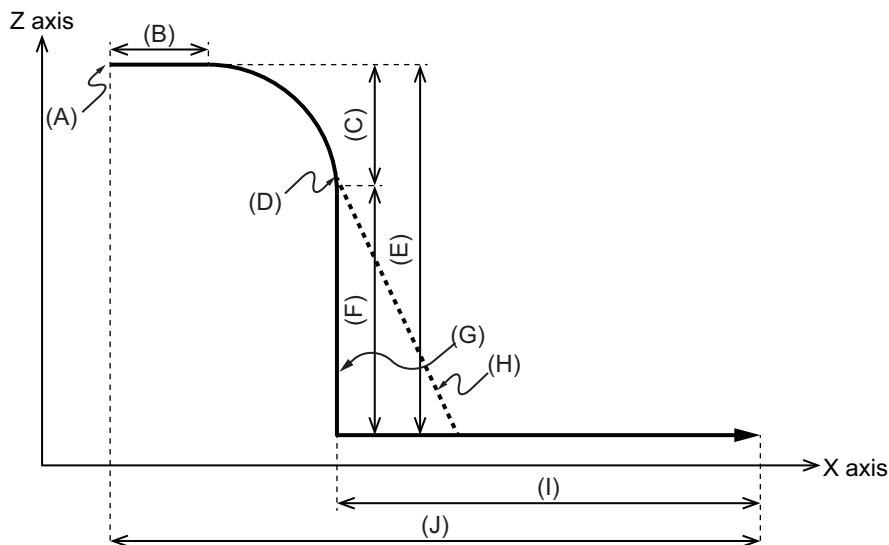
Symbol	Description
(A)	The current position of the robot
(B)	$T1 \times Ttrans1$
(C)	$T1, H1$
(D)	$T2$
(E)	$T3 - T3 \times Ttrans2$
(F)	<i>InSync</i> changes to TRUE when the position of the robot and the position of the workpiece match.
(G)	$T3$
(H)	$T3 \times Ttrans2$
(I)	The trajectory for the workpiece which is not moved.
(J)	The trajectory for the workpiece which is in motion.
(K)	The range in which <i>InSync</i> is TRUE.
(L)	The range of MC_SyncLinearConveyor (Start Conveyor Synchronization) instruction execution.

## 4 Robot Instructions

- 1** The instruction creates command positions according to which the robot moves away from the current position and goes down along Z axis during the T1 period.
- 2** After the elapse of  $(T1 \times Ttrans1)$ , the instruction creates command positions according to which the robot tracks the workpiece.  
At this moment, the robot does not stop.
- 3** After the elapse of  $(T1 + T2)$ , the instruction creates command positions according to which the robot moves down along Z axis.  
At this moment, the robot does not stop.
- 4** After the elapse of  $(T1 + T2 + (T3 - T3 \times Ttrans2))$ , the instruction compares the command position (X, Y) and the actual current position (X, Y) of the workpiece.  
If the difference is within the following range, *InSync* changes to TRUE. If the difference exceeds the following range, an error occurs and the axis stops. At this moment, the Workpiece Synchronization Excessive Following error (error code: 6702 hex) occurs.

Motion control period	Range of difference
500 $\mu$ s	0.1 mm max.
1 ms	0.1 mm max.
2 ms	0.2 mm max.
4 ms	0.4 mm max.

● When T1 = 0:



Symbol	Description
(A)	The current position of the robot
(B)	T2
(C)	$T3 - T3 \times Ttrans2$
(D)	<i>InSync</i> changes to TRUE when the position of the robot and the position of the workpiece match.
(E)	T3
(F)	$T3 \times Ttrans2$
(G)	The trajectory for the workpiece which is not moved.
(H)	The trajectory for the workpiece which is in motion.
(I)	The range in which <i>InSync</i> is TRUE.
(J)	The range of MC_SyncLinearConveyor (Start Conveyor Synchronization) instruction execution.

- 1 After the elapse of T2, the instruction creates command positions according to which the robot moves down along Z axis.

At this moment, the robot does not stop.

- 2 After the elapse of  $(T2 + (T3 - T3 \times Ttrans2))$ , the instruction compares the command position (X, Y) and the actual current position (X, Y) of the workpiece.

If the difference is within the following range, *InSync* changes to TRUE. If the difference exceeds the following range, an error occurs and the axis stops. At this moment, the Workpiece Synchronization Excessive Following error (error code: 6702 hex) occurs.

Motion control period	Range of difference
500 $\mu$ s	0.1 mm max.
1 ms	0.1 mm max.
2 ms	0.2 mm max.
4 ms	0.4 mm max.

### ● Setting Rotation:

In case of kinematics that includes the rotation of the wrist (Delta-3R), the final orientation of the selected TCP is set in the Initial Workpiece Position and the direction to follow in the rotation axis in the Direction input. The trajectory described in MC\_SyncLinearConveyor (Start Conveyor Synchronization) instruction is divided in 6 phases and it is possible to set when rotation is starting and when is finishing through the Trajectory Transitions.

- 1 Rotation starts at the beginning of the phase which is specified as Initial Phase. The parameter Rotation Transition 1 sets the Initial Phase.
- 2 Rotation finishes at the end of the phase which is specified as Final Phase. The parameter Rotation Transition 2 sets the Final Phase.

It is only possible to set phases 0 to 4, because at the end of phase 4, synchronization is reached and orientation of the robot must be already achieved.

The following are the limitations of Rotation Transitions.

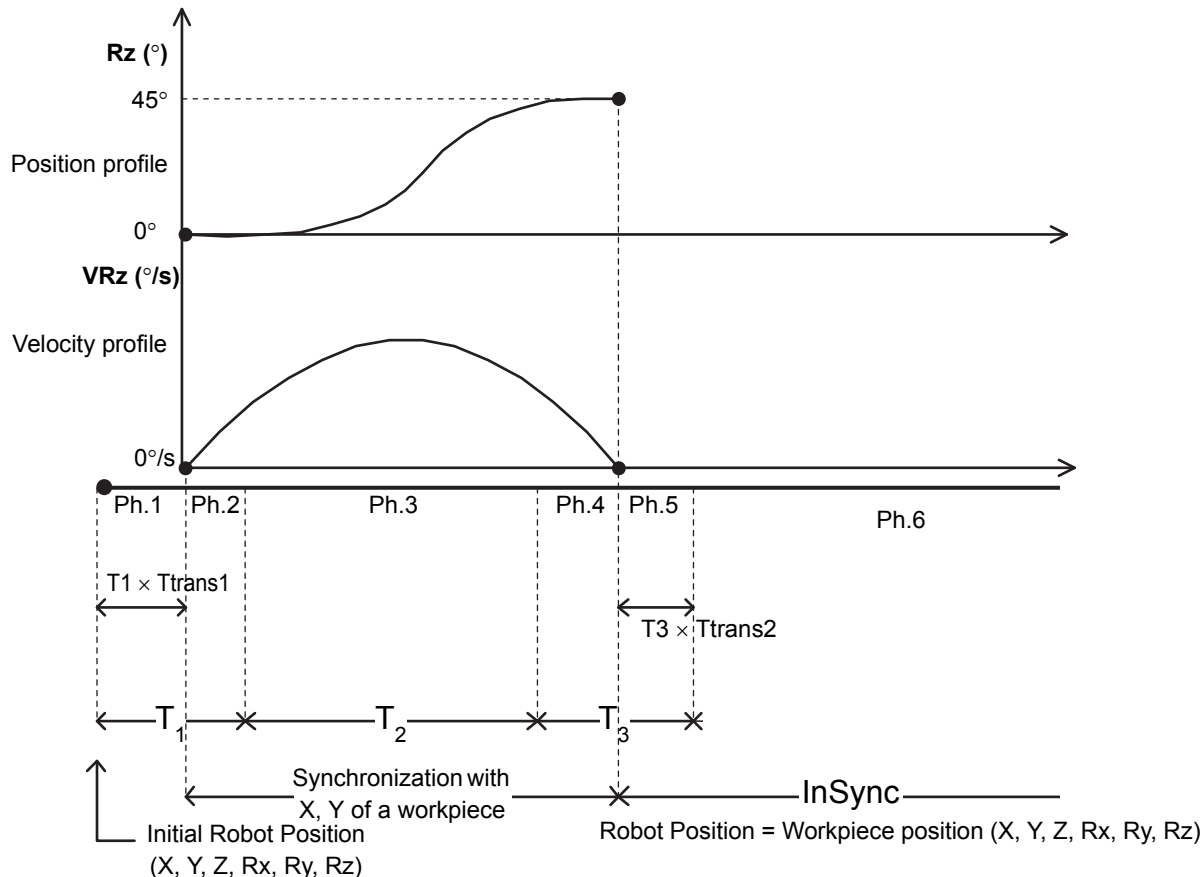
- In case Rotation Transition 1 and Rotation Transition 2 are set to zero, rotation is disabled.
- In case Rotation Transition 1 is zero and Rotation Transition 2 is different than zero error will occur.
- In case Rotation Transition 1 is different than zero and Rotation Transition 2 is zero error will occur.
- In case Rotation Transition 1 > Rotation Transition 2 error will occur.
- In case Rotation Transition 1 or Rotation Transition 2 are out of valid range error will occur.
- In case phase 1 is not performed ( $T1 = 0$ ) and Rotation Transition 1 = 1 or Rotation Transition 1 = 2 an error will occur.
- In case phase 2 is not performed (" $T1 - T1 \times Ttrans1$ " is less than Primary Cycle time), Rotation Transition 1 is set to 2 and Rotation Transition 2 is also set to 2 an error will occur.
- In case phase 4 is not performed (" $T3 - T3 \times Ttrans2$ " is less than Primary Cycle time), Rotation Transition 1 is set to 4 and Rotation Transition 2 is also set to 4 an error will occur.

In all cases above error identifier is Illegal Trajectory Transitions (error code: 5716 hex).

## 4 Robot Instructions

The following are the example of using rotation.

- Rotation transition1 (Initial Phase) = 2
- Rotation transition2 (Final Phase) = 4
- Rotation Rz from 0° to 45°
- Synchronization Trajectory Type Selection = polynomial 3 curve



### Details on `_sMC_SYNC_TRAJ_REF`

The following are the member variables of a `_sMC_SYNC_TRAJ_REF` data type variable.

#### ● SyncTrajType

Select the trajectory type that is used to create the command position for synchronization with the conveyor.

Name	Data type	Valid range	Description
SyncTrajType	<code>_eMC_SYNC_TRAJ_TYPE</code>	0: <code>_mcFlatTraj</code> 2: <code>_mcFlatTrajModSine</code> 3: <code>_mcFlatTrajModConstVel</code>	0: Polynomial 3 Curve 2: Modified Sine Curve 3: Modified Constant Velocity Curve

### ● TrajTime

Set the time for creating the command position used to move the robot from the current position to the target position.

Name	Data type	Valid range	Description
TrajTime [0]	UINT	Non-negative number	T1 [ms]
TrajTime [1]	UINT	Positive number	T2 [ms]
TrajTime [2]	UINT	Positive number	T3 [ms]
TrajTime [3] to [7] (Reserved)	UINT	0	(Reserved)

### ● TrajTransition

Set the percent of the time used for transition to the next motion.

Name	Data type	Valid range	Description
TrajTransition [0]	UINT	1 to 100	Ttrans1 [%]
TrajTransition [1]	UINT	1 to 100	Ttrans2 [%]
TrajTransition [2]	UINT	0 to 4	Rotation transition1 (Initial Phase)
TrajTransition [3]	UINT	0 to 4	Rotation transition2 (Final Phase)
TrajTransition [4] to [7] (Reserved)	UINT	0	(Reserved)



#### Precautions for Correct Use

When you use the Delta3R, set both *TrajTransition [2]* and *TrajTransition [3]* to the values other than 0 to enable the rotation of the Delta3R wrist axis.

If both values are set to 0, the rotation of the wrist axis is disabled. If one of the values is set to 0, an error occurs.

### ● TrajDistance

Set the travel distance between the current position and the target position.

Name	Data type	Valid range	Description
TrajDistance [0]	LREAL	Positive number, negative number or 0	The travel distance on the trajectory in Z axis direction in the machine coordinate system: H1 [mm]
TrajDistance [1] to [7] (Reserved)	LREAL	0	(Reserved)

### ● MaxVelocity

Set a velocity for which velocity errors are detected during command position creation.

A velocity error indicates that the command velocity exceeded the specified value. If you set this variable to 0, velocity errors are not detected.

Name	Data type	Valid range	Description
MaxVelocity	LREAL	Non-negative number	Unit: mm/s

### ● MaxAcceleration

Set an acceleration rate for which acceleration errors are detected during command position creation.

An acceleration error indicates that the command acceleration/deceleration rate exceeded the specified value. If you set this variable to 0, acceleration errors are not detected.

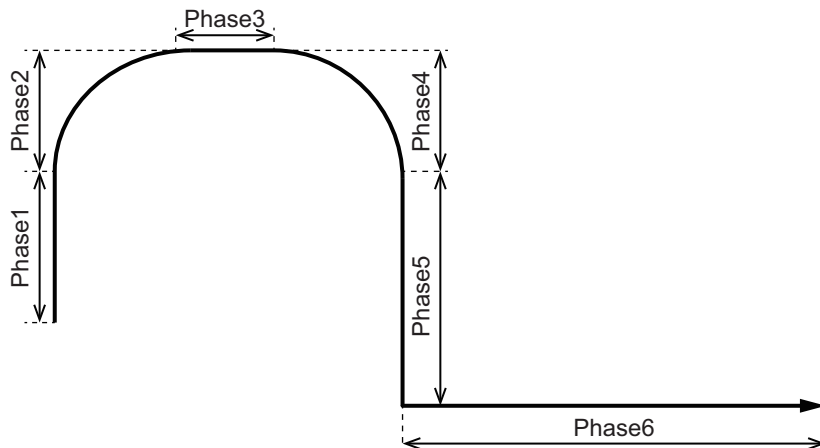
Name	Data type	Valid range	Description
MaxAcceleration	LREAL	Non-negative number	Unit: mm/s <sup>2</sup>

## Details on `_sMC_SYNC_TRAJ_STATUS`

- While the command position is created, the status information is output in the member variables of `_sMC_SYNC_TRAJ_STATUS` (Trajectory Status).

*InSync* is TRUE if Phase is 5 or 6.

Name	Description
RemainTime	Gives the remaining of the Trajectory Target Time ( $T1 + T2 + T3$ ). This value is calculated by a subtraction from the Trajectory Target Time ( $T1 + T2 + T3$ ). (unit: ms)
Phase	Phase0 to Phase6. Shows the phase in which trajectory calculation is in progress.
Phase0	The instruction is not executed.
Phase1	$t \leq T_{trans1} \times T1$
Phase2	$(T_{trans1} \times T1) < t \leq T1$
Phase3	$T1 < t \leq (T1 + T2)$
Phase4	$(T1 + T2) < t \leq (T1 + T2 + (T3 - T3 \times T_{trans2}))$
Phase5	$(T1 + T2 + (T3 - T3 \times T_{trans2})) < t \leq (T1 + T2 + T3)$
Phase6	$(T1 + T2 + T3) < t$



- In the case of that  $T1 - T1 \times T_{trans1}$  is less than Primary Task cycle time, phase 2 is not performed. The same happens for  $T3 - T3 \times T_{trans2}$ .

See the phase of `MC_SyncLinearConveyor` with transition at 100% on the following table.

Phase	Description
Phase1	$t \leq T1$
Phase2	(Not performed)
Phase3	$(T1) < t \leq (T1 + T2)$
Phase4	(Not performed)
Phase5	$(T1 + T2) < t \leq (T1 + T2 + T3)$
Phase6	$(T1 + T2 + T3) < t$

## Setting Offsets

The function adds an offset to the robot position and orientation when it is synchronized with a conveyor.

The offset is defined in a mixed format: Cartesian position for X, Y and Z and joint position for Orientation part. Offset positions are referenced to its own local coordinate system, attached to the TCP0 of the robot and aligned with the coordinate system set in the MC\_SyncLinearConveyor (MCS or UCS).

Offsets are applied each control cycle to the output of the MC\_SyncLinearConveyor. Therefore it is possible to keep synchronization with conveyor while other movements are performed.

### ● EnableOffset (Offset Enabled)

The offset functionality is enabled with the *EnableOffset* (Offset Enabled) input variable. Therefore, the offset is referenced to the position of the TCP0 in the moment when EnableOffset is set to TRUE.

When EnableOffset input is changed from TRUE to FALSE the offset is not subtracted. The output is the same as the last *PositionOffset* is kept.

Offset functionality can only be used during Phase 6 of MC\_SyncLinearConveyor. If the EnableOffset input is set to TRUE in a phase different than 6, error occurs.



### Precautions for Correct Use

- You can enable offsets only when TCP is in synchronization with the workpiece. If you enable offsets when synchronization is not performed, an error will occur.
- If the offset is changed from Disabled to Enabled, TCP may move suddenly depending on the offset value.

### ● PositionOffset (Offset Value)

To set the offset value, set the position (X, Y, Z) in the coordinate system in which the workpiece position is specified, and set the A3 rotational angle in the axis coordinate system (ACS).

Name	Data type	Valid range	Description
PositionOffset [0]	LREAL	Positive number, negative number or 0	Offset X coordinate [mm]
PositionOffset [1]	LREAL	Positive number, negative number or 0	Offset Y coordinate [mm]
PositionOffset [2]	LREAL	Positive number, negative number or 0	Offset Z coordinate [mm]
PositionOffset [3]	LREAL	Positive number, negative number or 0	The offset A3 in the axis coordinate system [degree]
PositionOffset [4] to [5] (Reserved)	LREAL	0	(Reserved)

The parameters you can set in *PositionOffset* (Offset Value) vary depending on *KinType* (Kinematics Type Selection) as shown below.

The invalid set parameters are ignored.

Name	Kinematics type		
	Delta3	Delta3R	Delta2
PositionOffset [0]	Valid	Valid	Valid
PositionOffset [1]	Valid	Valid	Invalid
PositionOffset [2]	Valid	Valid	Valid
PositionOffset [3]	Invalid	Valid	Invalid
PositionOffset [4] to [5] (Reserved)	Invalid	Invalid	Invalid

4 Robot Instructions

Name	Kinematics type				
	Cartesian 3D Cartesian 3D-Gantry	Cartesian 2D			Cartesian 2D-Gantry H-Bot
		Plane XY	Plane XZ	Plane YZ	
PositionOffset [0]	Valid	Valid	Valid	Invalid	Valid
PositionOffset [1]	Valid	Valid	Invalid	Valid	Valid
PositionOffset [2]	Valid	Invalid	Valid	Valid	Invalid
PositionOffset [3]	Invalid	Invalid			Invalid
PositionOffset [4] to [5] (Reserved)	Invalid	Invalid			Invalid

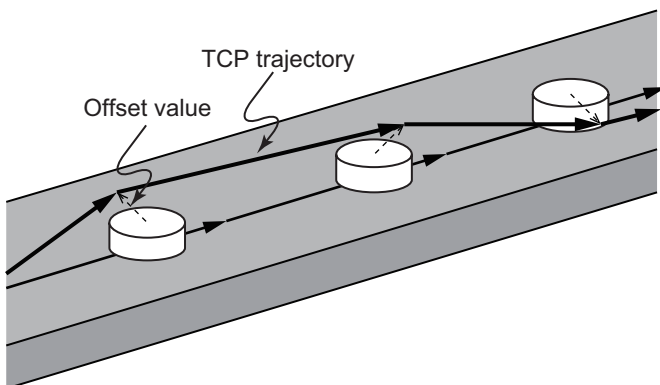
Name	Kinematics type	
	SCARA RRP SCARA PRR	SCARA RRP+R SCARA PRR+R
PositionOffset [0]	Valid	Valid
PositionOffset [1]	Valid	Valid
PositionOffset [2]	Valid	Valid
PositionOffset [3]	Invalid	Valid
PositionOffset [4]	Invalid	Invalid
PositionOffset [5]	Invalid	Invalid

● Operation Example

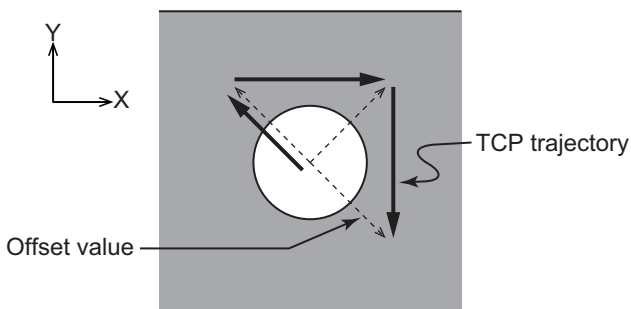
This example shows the operation with the following offset table.

Name	Offset Enabled/Disabled				
	Disabled	Enabled	Enabled	Enabled	Disabled
PositionOffset [0]	-10.0	-10.0	10.0	10.0	0.0
PositionOffset [1]	10.0	10.0	10.0	-10.0	0.0
PositionOffset [2]	0.0	0.0	0.0	0.0	0.0

The following figure shows the TCP operation with the offset value performed on the conveyor.



The following figure shows the TCP operation with the offset value viewed from above.



## Setting Coordinate System and Conveyor Axis

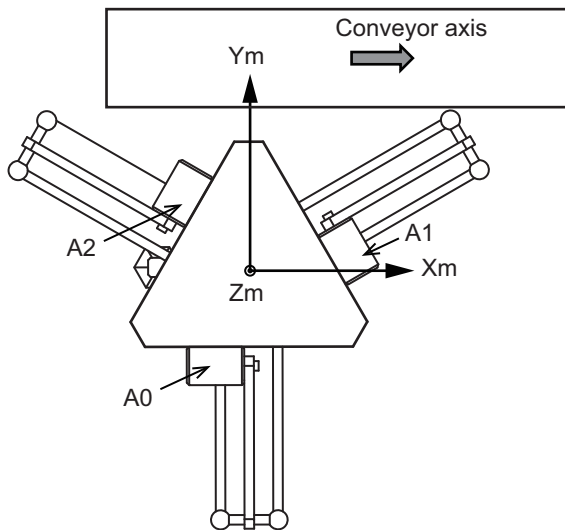
To make the robot track the workpiece specified in *InitWorkpiecePosition* (Initial Workpiece Position), you must align X axis in the coordinate system specified in *CoordSystem* (Coordinate System) with the conveyor axis direction.

In this example, the machine coordinate system (MCS) and user coordinate system (UCS) are specified.

### ● Specifying Machine Coordinate System (MCS)

Align  $X_m$  and the conveyor axis in the same direction as shown in the following figure.

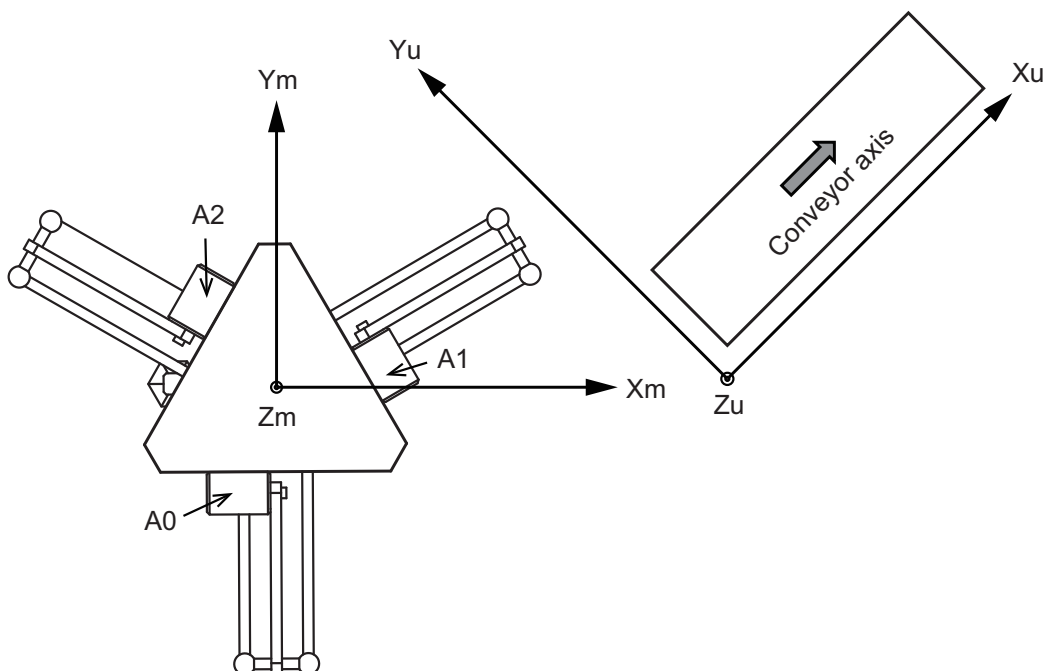
$X_m$ ,  $Y_m$  and  $Z_m$  represent the machine coordinate system (MCS).



### ● Specifying User Coordinate System (UCS)

Align  $X_u$  and the conveyor axis in the same direction as shown in the following figure.

$X_m$ ,  $Y_m$  and  $Z_m$  represent the machine coordinate system (MCS), and  $X_u$ ,  $Y_u$  and  $Z_u$  represent the user coordinate system (UCS).



## BufferMode (Buffer Mode Selection)

- A different instruction instance can be executed during axis motion.
- *BufferMode* specifies how to join the axis motions for this interpolation instruction and the previous interpolation instruction.
- The following Buffer Modes are supported for *BufferMode*.

Buffer Mode Selection	Description
Aborting	Cancels the instruction being executed and switches to this instruction. If the direction of axis motion is reversed by switching instructions, the motion will be reversed according to the Operation Selection at Reversing axis parameter.
Buffered	Buffers this instruction and executes it automatically after the current instruction is completed.
Blending	Starts the buffered instruction at the velocity (transit velocity) at which the current instruction reaches the target position. The operation of the current instruction is changed so that the axes reach the target position at the transit velocity. There are four methods to specify the transit velocity. These are described below.
Blending low	The lower of the target velocities of the current instruction and the buffered instruction is used as the transit velocity.
Blending previous	The target velocity of the current instruction is used as the transit velocity.
Blending next	The target velocity of the buffered instruction is used as the transit velocity.
Blending high	The higher of the target velocities of the current instruction and the buffered instruction is used as the transit velocity.

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

## TransitionMode (Transition Mode)

- *TransitionMode* specifies how to combine the trajectories created by the previous interpolation operation and the next interpolation operation.
- For *TransitionMode*, select *\_mcTMNone* (Transition disabled), *\_mcTMStartHeight* (Start height) or *\_mcTMStartRemainingTime* (Start remaining time).
- *TransitionMode* is valid when blending is specified for *BufferMode* (Buffer Mode Selection).
- An error will occur if you do not set *TransitionMode* to *\_mcTMNone* (Transition disabled) when blending is not used.

## TransitionParameter (Transition Parameters)

The setting of *TransitionParameter* (Transition Parameters) varies depending on whether *Transition-Mode* (Transition Mode) is set to *\_mcTMStartHeight* (Start height) or *\_mcTMStartRemainingTime* (Start remaining time).

### ● *\_mcTMStartHeight* (Start height)

Name	Data type	Valid range	Description
TransitionParameter [0]	LREAL	Positive number or 0	Transition Start Height (The height at which the transition is started) [mm]
TransitionParameter [1] to [7] (Reserved)	LREAL	0	(Reserved)

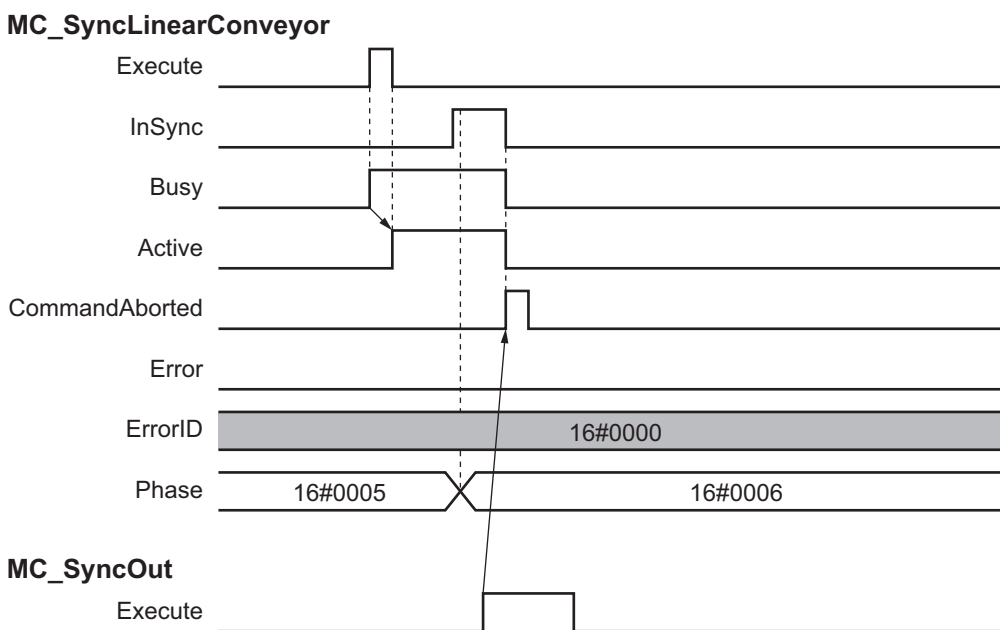
Refer to *\_mcTMStartHeight* (Start Height) on page 4-100 for details on *\_mcTMStartHeight* (Start height).

### ● *\_mcTMStartRemainingTime* (Start remaining time)

Name	Data type	Valid range	Description
TransitionParameter [0]	LREAL	Positive number or 0	Transition Start Time (The time at which the transition is started) [ms]
TransitionParameter [1] to [7] (Reserved)	LREAL	0	(Reserved)

Refer to *\_mcTMStartRemainingTime* (Start Remaining Time) on page 4-102 for details on *\_mcTM-StartRemainingTime* (Start remaining time).

## Timing Charts



## Re-execution of Motion Control Instructions

This instruction cannot be re-executed.

A Motion Control Instruction Re-execution Disabled error (error code: 543B hex) occurs if re-execution is attempted, and all axes in motion stop.

## Multi-execution of Motion Control Instructions

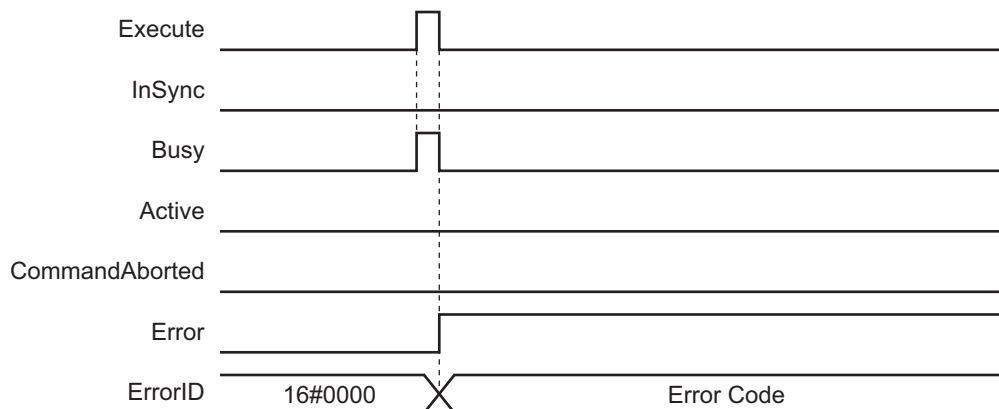
There are restrictions on the instructions that you can execute while this instruction is in execution.

Refer to *4-2 Multi-execution of Motion Control Instructions* on page 4-98 for details on multi-execution of motion control instructions.

## Errors

If an error occurs during instruction execution, *Error* will change to TRUE.

You can find out the cause of the error by referring to the value output by *ErrorID* (Error Code).



# MC\_SyncOut

The MC\_SyncOut instruction cancels synchronization with the conveyor.

Instruction	Name	FB/ FUN	Graphic expression	ST expression
MC_SyncOut	End Syn- chronization	FB		<pre>MC_SyncOut_instance (   AxesGroup :=parameter,   Execute :=parameter,   TrajData :=parameter,   DecelerationTrigger :=parameter, *1   Done =&gt;parameter,   Busy =&gt;parameter,   CommandAborted =&gt;parameter,   Error =&gt;parameter,   ErrorID =&gt;parameter );</pre>

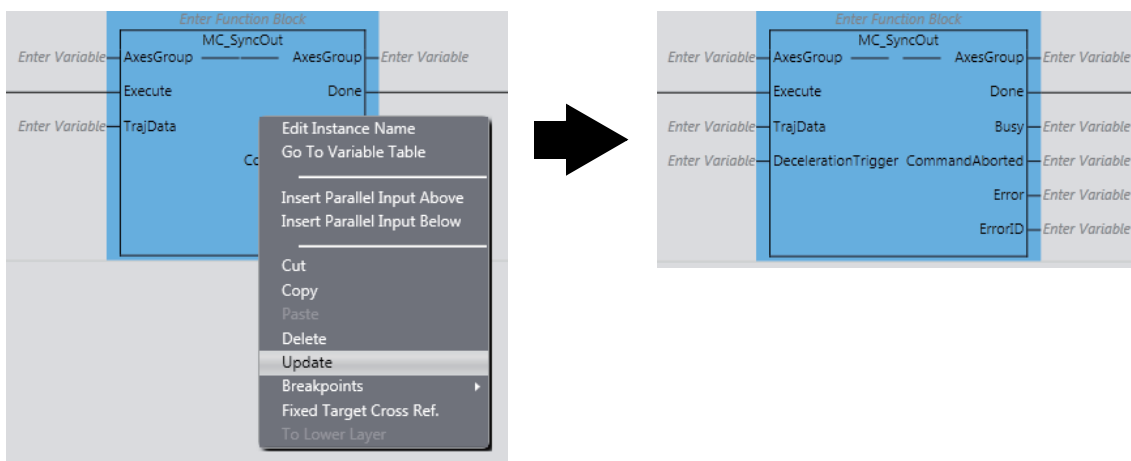
\*1. A CPU Unit with robot version 1.03 or later and Sysmac Studio version 1.15 or higher are required to use this variable. For any previous version combinations, this variable does not exist.



## Version Information

To use this instruction in CPU Unit version 1.11 or later with an existing project whose CPU Unit version is 1.10 or before, user has to update the ladder programs as follows:

- Operate **Change Device** to change Version of CPU Unit to 1.11 or later.
- In the ladder program, user has to right-click the existing MC\_SyncOut instruction and select **Update** to display *DecelerationTrigger*. Without this operation, Sysmac Studio will report errors in building the project.



- In ST program, user does not need to do anything. If *DecelerationTrigger* is not specified, it is maintained to the default value FALSE, so deceleration is never triggered.

Conversely, to use this instruction in CPU Unit version 1.10 or before with an existing project whose CPU Unit version is 1.11 or later, user has to update the ladder programs to remove *DecelerationTrigger* input and rebuild them from Sysmac Studio.

## Variables

This section describes the input variables, output variables and in-out variables for the MC\_SyncOut (End Synchronization) instruction.

### Input Variables

Name	Meaning	Data type	Valid range	Default	Description
Execute	Execute	BOOL	TRUE or FALSE	FALSE	The instruction is executed when <i>Execute</i> changes to TRUE.
TrajData	Trajectory Data	_sMC_SYNCOUT_TRAJ_REF	---	---	Set the parameters for canceling synchronization.
DecelerationTrigger <sup>*1</sup>	Deceleration Trigger	BOOL	TRUE or FALSE	FALSE	Finish synchronization and start deceleration when <i>DecelerationTrigger</i> changes to TRUE.  This variable is valid only when <i>Synchronization stop with Deceleration</i> is specified.

\*1. A CPU Unit with robot version 1.03 or later and Sysmac Studio version 1.15 or higher are required to use this variable. For any previous version combinations, this variable does not exist.

#### ● \_sMC\_SYNCOUT\_TRAJ\_REF

Name	Meaning	Data type	Valid range	Description
SyncStop Type	Sync Stop Type Selection	_eMC_SYNC_STOP_TYPE	0: _mcSyncStop 1: _mcRBT_ImmediateStop 2: _mcSyncDecelerationStop <sup>*4</sup>	Select the trajectory type that is used to create the command position for cancellation of synchronization with the conveyor.  0: Synchronized stop 1: Immediate stop 2: Synchronized stop with Deceleration
TrajTime	Trajectory Target Time	ARRAY [0..7] OF UINT	---	Set the target time for a synchronized stop. <sup>*1</sup>
Traj Transition (Reserved)	Trajectory Transition	ARRAY [0..7] OF UINT	0	(Reserved)
Traj Distance	Trajectory Travel Distance	ARRAY[0..7] OF LREAL	Positive number, negative number or 0	Set the travel distance between the current position and the target position. <sup>*1</sup>
MaxVelocity	Velocity Error Detection Value	LREAL	Non-negative number	Set a velocity for which velocity errors are detected <sup>*2</sup> during command position creation. (unit: mm/s)
Max Acceleration	Acceleration Error Detection Value	LREAL	Non-negative number	Set an acceleration rate for which acceleration errors are detected <sup>*3</sup> during command position creation. (unit: mm/s <sup>2</sup> )

\*1. Refer to *Details on \_sMC\_SYNCOUT\_TRAJ\_REF* on page 4-79 for details.

\*2. A velocity error indicates that the command velocity exceeded the specified value. If you set this variable to 0, velocity errors are not detected.

\*3. An acceleration error indicates that the command acceleration/deceleration rate exceeded the specified value. If you set this variable to 0, acceleration errors are not detected.

\*4. A CPU Unit with robot version 1.03 or later and Sysmac Studio version 1.15 or higher are required to use this sync stop type. For any previous version combinations, this type does not exist.

## Output Variables

Name	Meaning	Data type	Valid range	Description
Done	Done	BOOL	TRUE or FALSE	TRUE when the instruction is completed.
Busy	Executing	BOOL	TRUE or FALSE	TRUE when the instruction is acknowledged.
CommandAborted	Command Aborted	BOOL	TRUE or FALSE	TRUE when the instruction is aborted.
Error	Error	BOOL	TRUE or FALSE	TRUE while there is an error.
ErrorID	Error Code	WORD	*1	Contains the error code when an error occurs. A value of 16#0000 indicates normal execution.

\*1. Refer to *Section 6 Troubleshooting* for details.

### ● Output Variable Update Timing

Name	Timing for changing to TRUE	Timing for changing to FALSE
Done	When the set target time elapsed.	<ul style="list-style-type: none"> <li>When <i>Execute</i> is TRUE and changes to FALSE.</li> <li>After one period when <i>Execute</i> is FALSE.</li> </ul>
Busy	When <i>Execute</i> changes to TRUE.	<ul style="list-style-type: none"> <li>When <i>Done</i> changes to TRUE.</li> <li>When <i>Error</i> changes to TRUE.</li> <li>When <i>CommandAborted</i> changes to TRUE.</li> </ul>
CommandAborted	<ul style="list-style-type: none"> <li>When this instruction is aborted because another motion control instruction was executed with the Buffer Mode set to <i>Aborting</i>.</li> <li>When this instruction is canceled due to an error.</li> <li>When this instruction is executed while there is an error.</li> <li>When you start this instruction during MC_GroupStop instruction execution.</li> </ul>	<ul style="list-style-type: none"> <li>When <i>Execute</i> is TRUE and changes to FALSE.</li> <li>After one period when <i>Execute</i> is FALSE.</li> </ul>
Error	When there is an error in the execution conditions or input parameters for the instruction.	When the error is cleared.

## In-Out Variables

Name	Meaning	Data type	Valid range	Description
AxesGroup	Axes Group	_sGROUP_REF	---	Specify the axes group.

## Function

- This instruction cancels ongoing robot (axes group) synchronization which is started by the MC\_SyncLinearConveyor (Start Conveyor Synchronization) instruction.
- This instruction is completed when the set target time elapsed. An in-position check is not performed.
- The workspace check is performed if it is enabled. If an error is detected, the axes group operation stops.

Refer to 2-3-2 *Workspace Check* on page 2-30 for details on the workspace.

- While the command position is created, the command velocity and command acceleration rate that are adjusted for the conveyor velocity are also created. The instruction checks whether or not these values exceed *MaxVelocity* (Velocity Error Detection Value) or *MaxAcceleration* (Acceleration Error Detection Value). If a specified limit is exceeded, an error occurs and the axis stops.
- You can use this instruction only when all the following conditions are met.
  - a) The kinematics is already set with MC\_SetKinTransform (Set Kinematics Transformation) instruction.
  - b) Home for all axes registered in the axes group are already defined.
  - c) The Servo is already ON for all axes registered in the axes group.
  - d) The limit input is OFF for all axes registered in the axes group.
  - e) The axes group is enabled.
  - f) If you select a synchronized stop, *Phase* of the trajectory status must be 6.  
This is checked only when the instruction is executed.
  - g) If you select an immediate stop, MC\_SyncLinearConveyor (Start Conveyor Synchronization) instruction execution must be in progress.



### Precautions for Safe Use

- Adjust the home of the robot before you start robot operation.
- Do not operate the robot outside the workspace when the workspace check function is disabled. If you do so, the robot may be damaged.
- Use the MC\_SyncOut (End Synchronization) instruction to stop the robot operated by the MC\_SyncLinearConveyor (Start Conveyor Synchronization) instruction. If you use a different instruction to stop, the robot does not decelerate smoothly.
- If you set 0 for the *MaxVelocity* (Velocity Error Detection Value) or *MaxAcceleration* (Acceleration Error Detection Value) parameter for input variables, or if you use the default (0) of these variables, the velocity error check or acceleration error check is not performed. Properly set *MaxVelocity* (Velocity Error Detection Value) and *MaxAcceleration* (Acceleration Error Detection Value) to avoid an unexpected velocity and acceleration.
- If a robot tool (*ToolID*: 1 to 16) other than TCP0 is selected, checks for *MaxVelocity* (Velocity Error Detection Value) and *MaxAcceleration* (Acceleration Error Detection Value) are not performed at instruction execution.

## Setting Trajectory

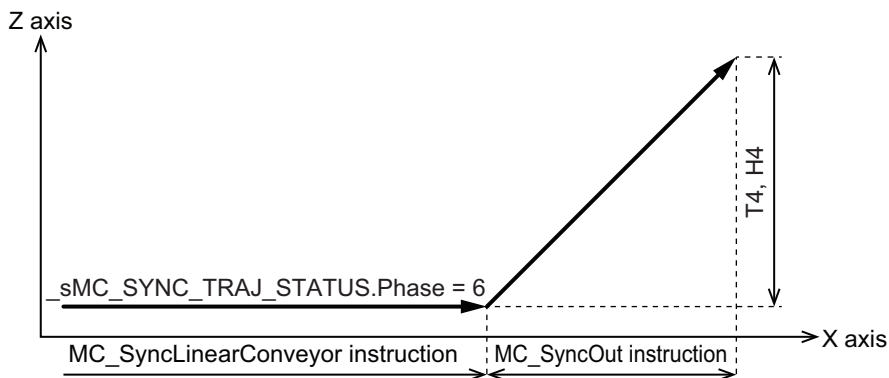
- Use *TrajData* (Trajectory Data) to set the parameters for creating the trajectory for synchronization cancellation.
- There are three types of *SyncStopType* (Synch Stop Type Selection): *Immediate stop*, *Synchronized stop*, and *Synchronized stop with Deceleration*.
- When this instruction is executed with *SyncStopType* set to *Immediate stop*, the MC\_SyncLinearConveyor (Start Conveyor Synchronization) instruction operation is aborted and the subsequent command position creation is stopped.
- If you select *Synchronized stop*, you can execute this instruction only when *Phase* of the trajectory status is 6.

The details on stopping operation and parameter setting are described below.

The following symbols are used in this description.

`_sMC_SYNC_TRAJ_STATUS.Phase=6` means that *Phase* of the trajectory status is 6.

Symbol	Description
T4	Trajectory target time (unit: ms)
H4	The travel distance on the trajectory in Z axis direction in the machine coordinate system (unit: mm)



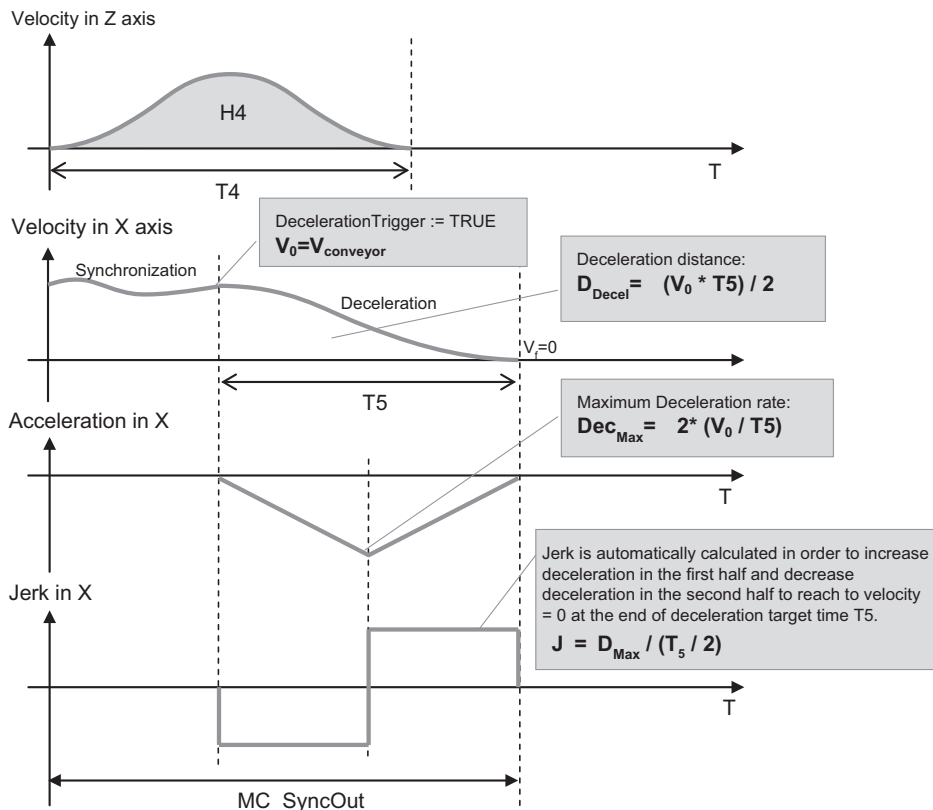
- The instruction creates command positions according to which the robot moves away from the current position and reaches the (current position + H4) position on Z axis after the elapse of T4.
- When the operation switches from the MC\_SyncLinearConveyor (Start Conveyor Synchronization) instruction to the MC\_SyncOut (End Synchronization) instruction, transition velocities are created with command positions.
- If you select *Synchronized stop with Deceleration*, you can execute this instruction only when *Phase* of the trajectory status is 6 too.

The details on stopping operation and parameter setting are described below.

Symbol	Description
T4	Trajectory target time on Z axis (unit: ms)
T5	Deceleration target time (unit: ms)
H4	The travel distance on the trajectory in Z axis direction in the machine coordinate system (unit: mm)

- From the beginning of the execution, the instruction works as *Synchronized Stop*.
- When *DecelerationTrigger* is changed from FALSE to TRUE, the robot will finish synchronization and start deceleration in X axis. The movement in X axis will be decelerated to stop, in the specified deceleration target time T5. The trajectory shape of deceleration is a Double-S with fixed Jerk.

## 4 Robot Instructions



- When velocity in X axis is 0 and the trigger is detected, the movement in X axis is not performed, but the duration of T5 is waited to end the MC\_SyncOut instruction.



#### Precautions for Correct Use

When Synchronized stop with Deceleration is selected and DecelerationTrigger remains FALSE till the end of Trajectory target time T4, the movement in X axis stops without deceleration.



#### Precautions for Safe Use

When Synchronized stop with Deceleration is selected and Deceleration target time T5 is set to unexpected long time, the robot may reach out of the workspace during deceleration. Do not disable workspace check.



#### Additional Information

If you select `_mcSyncStop` or `_mcSyncDecelerationStop` for `SyncStopType` and set the value of H4 to 0, the robot performs synchronized motion according to the value of T4 and then stops.

## Details on `_sMC_SYNCOUT_TRAJ_REF`

The following are the member variables of the `_sMC_SYNCOUT_TRAJ_REF` data type variable.

### ● SyncStopType

Select the trajectory type that is used to create the command position for cancellation of synchronization with the conveyor.

Name	Data type	Valid range	Description
SyncStopType	<code>_eMC_SYNC_STOP_TYPE</code>	0: <code>_mcSyncStop</code> 1: <code>_mcRBT_ImmediateStop</code> 2: <code>_mcSyncDecelerationStop</code> *1	Select the trajectory type that is used to create the command position for cancellation of synchronization with the conveyor. 0: Synchronized stop 1: Immediate stop 2: Synchronized stop with Deceleration

\*1. A CPU Unit with robot version 1.03 or later and Sysmac Studio version 1.15 or higher are required to use this sync stop type. For any previous version combinations, this type does not exist.

### ● TrajTime

Set the target time for a synchronized stop.

Name	Data type	Valid range	Description
TrajTime [0]	UINT	Non-negative number	T4 [ms]
TrajTime [1]	UINT	Non-negative number	T5 [ms] Used only when <i>Synchronized stop with Deceleration</i> is specified
TrajTime [2] to [7] (Reserved)	UINT	0	(Reserved)

### ● TrajTransition (Reserved)

Name	Data type	Valid range	Description
TrajTransition [0] to [7] (Reserved)	UINT	0	(Reserved)

### ● TrajDistance

Set the travel distance between the current position and the target position.

Name	Data type	Valid range	Description
TrajDistance [0]	LREAL	Non-negative number	The travel distance on the trajectory in Z axis direction in the machine coordinate system: H4 [mm]
TrajDistance [1] to [7] (Reserved)	LREAL	0	(Reserved)

### ● MaxVelocity

Set a velocity for which velocity errors are detected during command position creation.

A velocity error indicates that the specified velocity is exceeded.

Name	Data type	Valid range	Description
MaxVelocity	LREAL	Non-negative number	Unit: mm/s

## ● MaxAcceleration

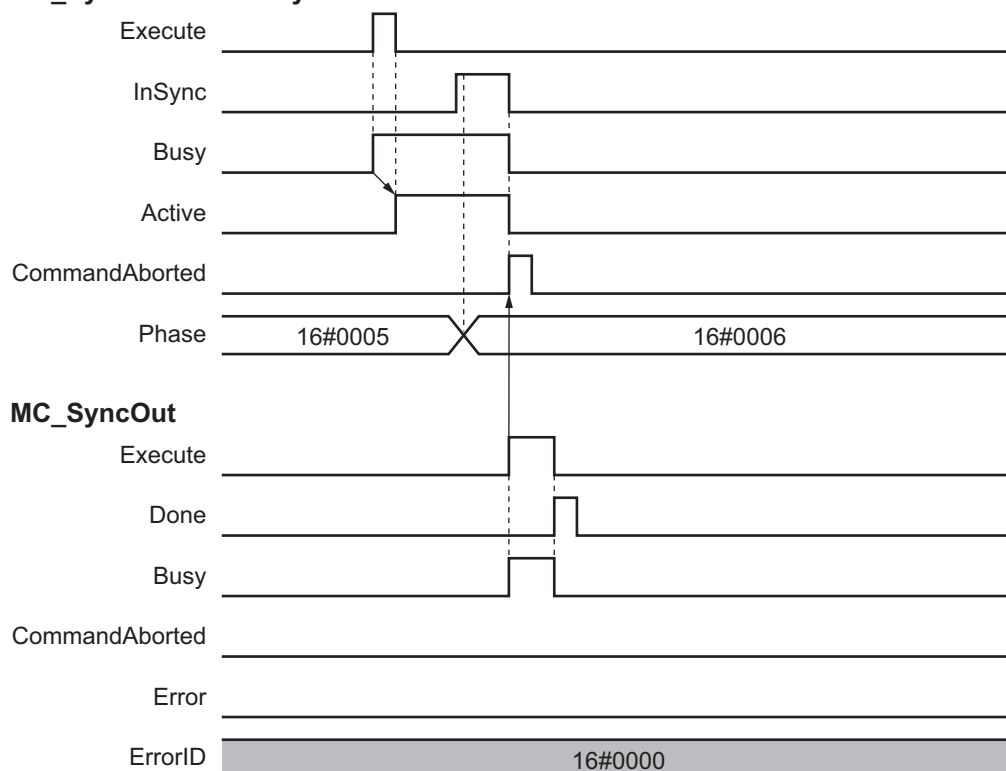
Set an acceleration rate for which acceleration errors are detected during command position creation.

An acceleration error indicates that the specified acceleration or deceleration rate is exceeded.

Name	Data type	Valid range	Description
MaxAcceleration	LREAL	Non-negative number	Unit: mm/s <sup>2</sup>

## Timing Charts

### MC\_SyncLinearConveyor



## Re-execution of Motion Control Instructions

This instruction cannot be re-executed.

A Motion Control Instruction Re-execution Disabled error (error code: 543B hex) occurs if re-execution is attempted, and all axes in motion stop.

## Multi-execution of Motion Control Instructions

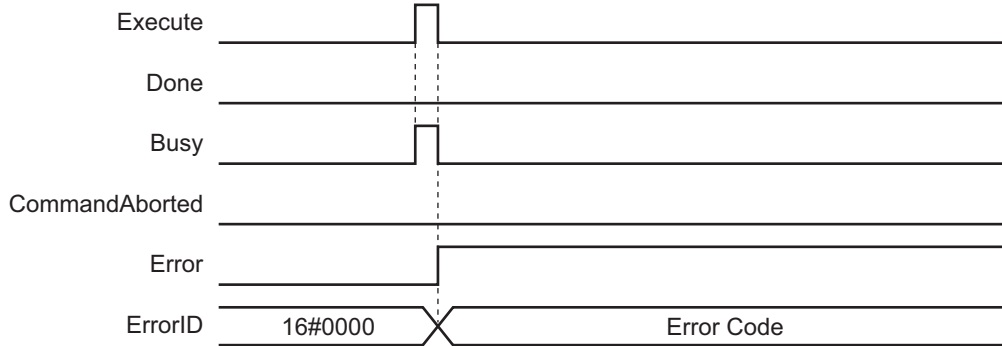
There are restrictions on the instructions that you can execute while this instruction is in execution.

Refer to 4-2 *Multi-execution of Motion Control Instructions* on page 4-98 for details on multi-execution of motion control instructions.

## Errors

If an error occurs during instruction execution, *Error* will change to TRUE.

You can find out the cause of the error by referring to the value output by *ErrorID* (Error Code).



# MC\_InverseKin

The MC\_InverseKin instruction performs inverse kinematics transformation.

Instruction	Name	FB/ FUN	Graphic expression	ST expression
MC_InverseKin	Inverse Kinematics	FB		<pre>MC_InverseKin_instance (   AxesGroup :=parameter,   Enable :=parameter,   Position :=parameter,   ArmConfig :=parameter,   ToolID :=parameter,   Valid =&gt;parameter,   Busy =&gt;parameter,   Error =&gt;parameter,   ErrorID =&gt;parameter,   AxesPosition =&gt;parameter,   OutWorkspace =&gt;parameter );</pre>

## Variables

This section describes the input variables, output variables and in-out variables for the MC\_InverseKin (Inverse Kinematics) instruction.

### Input Variables

Name	Meaning	Data type	Valid range	Default	Description
Enable	Enable	BOOL	TRUE or FALSE	FALSE	Transforms a target position into axes target positions when <i>Enable</i> changes to TRUE.
Position	Target Position	ARRAY[0..5] OF LREAL	Positive number, negative number or 0	0	Set the target position in the machine coordinate system (MCS). <sup>*1</sup>
ArmConfig <sup>*2</sup>	Arm Configuration	_eMC_ARM_CONFIG	_mcCurrentArmConfig _mcRightArmConfig or _mcLeftArmConfig	0	For the robots that have more than one way to reach a Cartesian position, the inverse kinematics calculation will be done for the current, left or right option: 0: _mcCurrentArmConfig 1: _mcRightArmConfig 2: _mcLeftArmConfig
ToolID	Tool ID	UINT	0 to 16	0	Select the ID of the robot tool.

\*1. Refer to *Details on Position and AxesPosition* on page 4-84 for details.

\*2. A CPU Unit with robot version 1.04 or later and Sysmac Studio version 1.18 or higher are required to use this input. For any previous version combinations, this type does not exist.

The input can be used only in combinations with SCARA RRP, RRP+R, PRR and PRR+R kinematics and it will be ignored with the other kinematics.

## Output Variables

Name	Meaning	Data type	Valid range	Description
Valid	Enabled	BOOL	TRUE or FALSE	<i>AxesPosition</i> (Axes Target Position) is valid when this variable is TRUE. Axes target positions are updated each period.
Busy	Executing	BOOL	TRUE or FALSE	TRUE when the instruction is acknowledged.
Error	Error	BOOL	TRUE or FALSE	TRUE while there is an error.
ErrorID	Error Code	WORD	*1	Contains the error code when an error occurs. A value of 16#0000 indicates normal execution.
AxesPosition	Axes Target Position	ARRAY[0..5] OF LREAL	Positive number, negative number or 0	Contains axes target positions that are transformed from a target position in the machine coordinate system (MCS). <sup>*2</sup>
OutWorkspace	Outside Workspace	BOOL	TRUE or FALSE	TRUE when the target position is outside the workspace.

\*1. Refer to *Section 6 Troubleshooting* for details.

\*2. Refer to *Details on Position and AxesPosition* on page 4-84 for details.

### ● Output Variable Update Timing

Name	Timing for changing to TRUE	Timing for changing to FALSE
Valid	When kinematics transformation is completed successfully.	<ul style="list-style-type: none"> <li>When <i>Error</i> changes to TRUE.</li> <li>When <i>Enable</i> is FALSE.</li> </ul>
Busy	When <i>Execute</i> changes to TRUE.	<ul style="list-style-type: none"> <li>When <i>Enable</i> changes to FALSE.</li> <li>When <i>Error</i> changes to TRUE.</li> </ul>
Error	When there is an error in the execution conditions or input parameters for the instruction.	When the error is cleared.
OutWorkspace	When the target position is outside the workspace.	<ul style="list-style-type: none"> <li>When the target position is within the workspace.</li> <li>When <i>Enable</i> is FALSE.</li> <li>When <i>Error</i> changes to TRUE.</li> </ul>

## In-Out Variables

Name	Meaning	Data type	Valid range	Description
AxesGroup	Axes Group	_sGROUP_REF	---	Specify the axes group.

## Function

- This instruction transforms *Position* (Target Position) in the machine coordinate system (MCS) into axes target positions in the axis coordinate systems (ACS).
- If *Position* (Target Position) is outside the workspace, *OutWorkspace* (Outside Workspace) changes to TRUE regardless of workspace is enabled or disabled.
- Valid* (Enabled) changes to TRUE when transformation of *Position* (Target Position) into axes target positions in the axis coordinate systems (ACS) is completed.
- You can use this instruction only when the following condition is met.

The kinematics is already set with *MC\_SetKinTransform* (Set Kinematics Transformation) instruction.

## Details on Position and AxesPosition

The following are the details on *Position* (Target Position) which you set and *AxesPosition* (Axes Target Position) which is the result of transformation.

### ● Position

Set the target position in the machine coordinate system (MCS).

Name	Data type	Valid range	Description
Position [0]	LREAL	Positive number, negative number or 0	The target position on X axis in the machine coordinate system [mm]
Position [1]	LREAL	Positive number, negative number or 0	The target position on Y axis in the machine coordinate system [mm]
Position [2]	LREAL	Positive number, negative number or 0	The target position on Z axis in the machine coordinate system [mm]
Position [3]	LREAL	Greater than -180.0, less than or equal to 180.0	The target rotational position around X axis in the machine coordinate system [degree]
Position [4]	LREAL	Greater than -180.0, less than or equal to 180.0	The target rotational position around Y axis in the machine coordinate system [degree]
Position [5]	LREAL	Greater than -180.0, less than or equal to 180.0	The target rotational position around Z axis in the machine coordinate system [degree]

The target position you can set in *Position* (Target Position) varies depending on *KinType* (Kinematics Type Selection) as shown below.

Position		Kinematics type		
Name	Description	Delta3	Delta3R	Delta2
Position [0]	The target position on X axis [mm]	Valid	Valid	Valid
Position [1]	The target position on Y axis [mm]	Valid	Valid	Invalid
Position [2]	The target position on Z axis [mm]	Valid	Valid	Valid
Position [3]	The target rotational position around X axis [degree]	Invalid	Invalid	Invalid
Position [4]	The target rotational position around Y axis [degree]	Invalid	Invalid	Invalid
Position [5]	The target rotational position around Z axis [degree]	Invalid	Valid	Invalid

Position	Kinematics type				
	Cartesian 3D Cartesian 3D-Gantry	Cartesian 2D			Cartesian 2D-Gantry H-Bot
Plane XY		Plane XZ	Plane YZ		
Position [0]	Valid	Valid	Valid	Invalid	Valid
Position [1]	Valid	Valid	Invalid	Valid	Valid
Position [2]	Valid	Invalid	Valid	Valid	Invalid
Position [3]	Invalid	Invalid			Invalid
Position [4]	Invalid	Invalid			Invalid
Position [5]	Invalid	Invalid			Invalid

Position	Kinematics type		
	SCARA RRP SCARA PRR	SCARA RRP+R SCARA PRR+R	Expansion1
Position [0]	Valid	Valid	Valid
Position [1]	Valid	Valid	Valid
Position [2]	Valid	Valid	Invalid
Position [3]	Invalid	Invalid	Invalid
Position [4]	Invalid	Invalid	Invalid
Position [5]	Invalid	Valid	Valid

### ● AxesPosition

This variable contains axes target positions that are transformed from a target position in the machine coordinate system (MCS).

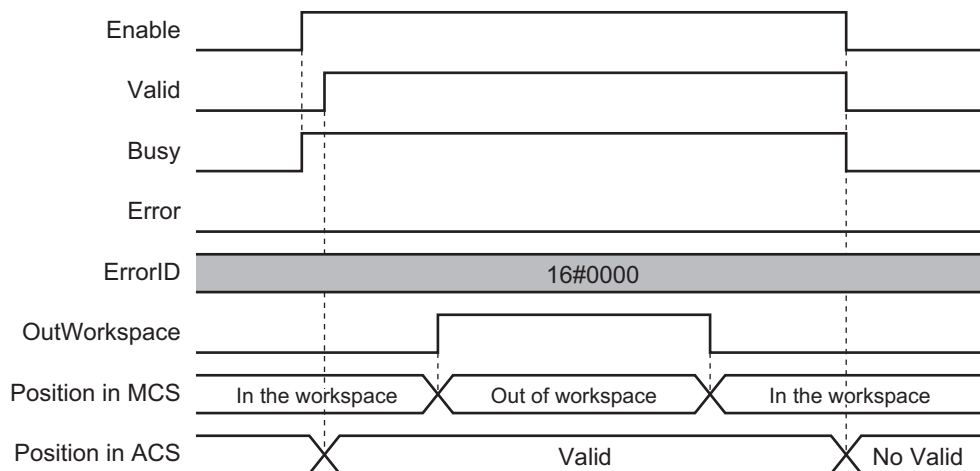
Name	Data type	Valid range	Description
AxesPosition [0]	LREAL	Positive number, negative number or 0	The target position in the axis coordinate system for A0 axis ([degree] or [mm], depending on <i>KinType</i> )
AxesPosition [1]	LREAL	Positive number, negative number or 0	The target position in the axis coordinate system for A1 axis ([degree] or [mm], depending on <i>KinType</i> )
AxesPosition [2]	LREAL	Positive number, negative number or 0	The target position in the axis coordinate system for A2 axis ([degree] or [mm], depending on <i>KinType</i> )
AxesPosition [3]	LREAL	Positive number, negative number or 0	The target position in the axis coordinate system for A3 axis ([degree] or [mm], depending on <i>KinType</i> )
AxesPosition [4] to [5] (Reserved)	LREAL	0	(Reserved)

*AxesPosition* (Axes Target Position) varies depending on *KinType* (Kinematics Type Selection) as shown below.

AxesPosition		Kinematics type		
Name	Description	Delta3	Delta3R	Delta2
AxesPosition [0]	The target position in the axis coordinate system for A0 axis ([degree] or [mm], depending on <i>KinType</i> )	Used	Used	Used
AxesPosition [1]	The target position in the axis coordinate system for A1 axis ([degree] or [mm], depending on <i>KinType</i> )	Used	Used	Used
AxesPosition [2]	The target position in the axis coordinate system for A2 axis ([degree] or [mm], depending on <i>KinType</i> )	Used	Used	Not used
AxesPosition [3]	The target position in the axis coordinate system for A3 axis ([degree] or [mm], depending on <i>KinType</i> )	Not used	Used	Not used
AxesPosition [4] to [5] (Reserved)	(Reserved)	Not used	Not used	Not used

AxesPosition	Kinematics type					
	Cartesian 3D-Gantry	Cartesian 3D Cartesian 2D-Gantry	Cartesian 2D H-Bot	SCARA RRP SCARA PRR	SCARA RRP+R SCARA PRR+R	Expansion1
AxesPosition [0]	Used	Used	Used	Used	Used	Used
AxesPosition [1]	Used	Used	Used	Used	Used	Used
AxesPosition [2]	Used	Used	Not used	Used	Used	Used
AxesPosition [3]	Used	Not used	Not used	Not used	Used	Not used
AxesPosition [4] to [5] (Reserved)	Not used	Not used	Not used	Not used	Not used	Not used

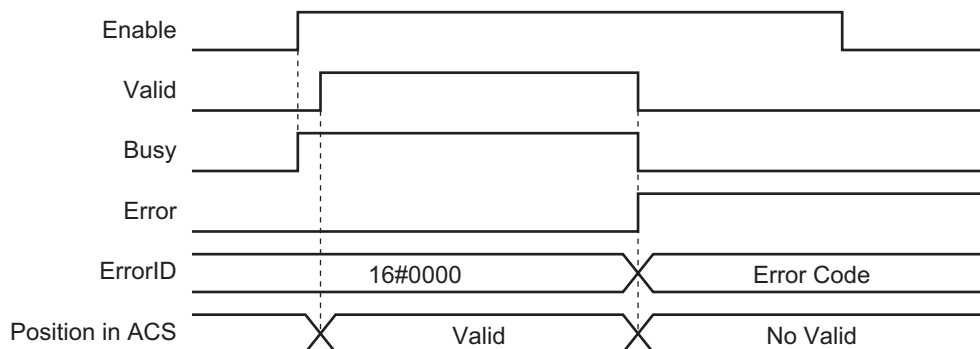
### Timing Charts



### Errors

If an error occurs during instruction execution, *Error* will change to TRUE.

You can find out the cause of the error by referring to the value output by *ErrorID* (Error Code).



# MC\_RobotJog

The MC\_RobotJog instruction jogs the axes group according to the specified target velocity.

Instruction	Name	FB/ FUN	Graphic expression	ST expression
MC_RobotJog	Axes Group Jog	FB		<pre>MC_RobotJog_instance (   AxesGroup :=parameter,   PositiveEnable_X :=parameter,   NegativeEnable_X :=parameter,   PositiveEnable_Y :=parameter,   NegativeEnable_Y :=parameter,   PositiveEnable_Z :=parameter,   NegativeEnable_Z :=parameter,   PositiveEnable_RX :=parameter,   NegativeEnable_RX :=parameter,   PositiveEnable_RY :=parameter,   NegativeEnable_RY :=parameter,   PositiveEnable_RZ :=parameter,   NegativeEnable_RZ :=parameter,   Velocity :=parameter,   Acceleration :=parameter,   Deceleration :=parameter,   JogMode :=parameter,   CSID :=parameter,   ToolID :=parameter,   Busy =&gt;parameter,   CommandAborted =&gt;parameter,   Error =&gt;parameter,   ErrorID =&gt;parameter );</pre>

MC\_RobotJog

4

Variables

## Variables

This section describes the input variables, output variables and in-out variables for the MC\_RobotJog (Axes Group Jog) instruction.

### Input Variables

Name	Meaning	Data type	Valid range	Default	Description
Positive Enable_X	X Positive Enable	BOOL	TRUE or FALSE	FALSE	When this variable changes to TRUE, the robot starts moving in the X axis positive direction. When it changes to FALSE, the robot stops moving.
Negative Enable_X	X Negative Enable	BOOL	TRUE or FALSE	FALSE	When this variable changes to TRUE, the robot starts moving in the X axis negative direction. When it changes to FALSE, the robot stops moving.

## 4 Robot Instructions

Name	Meaning	Data type	Valid range	Default	Description
Positive Enable_Y	Y Positive Enable	BOOL	TRUE or FALSE	FALSE	When this variable changes to TRUE, the robot starts moving in the Y axis positive direction. When it changes to FALSE, the robot stops moving.
Negative Enable_Y	Y Negative Enable	BOOL	TRUE or FALSE	FALSE	When this variable changes to TRUE, the robot starts moving in the Y axis negative direction. When it changes to FALSE, the robot stops moving.
Positive Enable_Z	Z Positive Enable	BOOL	TRUE or FALSE	FALSE	When this variable changes to TRUE, the robot starts moving in the Z axis positive direction. When it changes to FALSE, the robot stops moving.
Negative Enable_Z	Z Negative Enable	BOOL	TRUE or FALSE	FALSE	When this variable changes to TRUE, the robot starts moving in the Z axis negative direction. When it changes to FALSE, the robot stops moving.
Positive Enable_RX	X Rotate Positive Enable	BOOL	TRUE or FALSE	FALSE	When this variable changes to TRUE, the robot starts moving in the positive rotation direction around X axis. When it changes to FALSE, the robot stops moving.
Negative Enable_RX	X Rotate Negative Enable	BOOL	TRUE or FALSE	FALSE	When this variable changes to TRUE, the robot starts moving in the negative rotation direction around X axis. When it changes to FALSE, the robot stops moving.
Positive Enable_RY	Y Rotate Positive Enable	BOOL	TRUE or FALSE	FALSE	When this variable changes to TRUE, the robot starts moving in the positive rotation direction around Y axis. When it changes to FALSE, the robot stops moving.
Negative Enable_RY	Y Rotate Negative Enable	BOOL	TRUE or FALSE	FALSE	When this variable changes to TRUE, the robot starts moving in the negative rotation direction around Y axis. When it changes to FALSE, the robot stops moving.
Positive Enable_RZ	Z Rotate Positive Enable	BOOL	TRUE or FALSE	FALSE	When this variable changes to TRUE, the robot starts moving in the positive rotation direction around Z axis. When it changes to FALSE, the robot stops moving.
Negative Enable_RZ	Z Rotate Negative Enable	BOOL	TRUE or FALSE	FALSE	When this variable changes to TRUE, the robot starts moving in the negative rotation direction around Z axis. When it changes to FALSE, the robot stops moving.
Velocity	Target Velocity	ARRAY[0..1] OF LREAL	Non-negative number	0	Set the target velocity. (unit: mm/s or degree/s)
Acceleration	Acceleration Rate	ARRAY[0..1] OF LREAL	Non-negative number	0	Set the target acceleration rate. (unit: mm/s <sup>2</sup> or degree/s <sup>2</sup> )
Deceleration	Deceleration Rate	ARRAY[0..1] OF LREAL	Non-negative number	0	Set the target deceleration rate. (unit: mm/s <sup>2</sup> or degree/s <sup>2</sup> )
JogMode	Jog Mode Selection	_eMC_JOG_MODE	1: _mcMachine Mode 2: _mcUserMode 3: _mcToolMode	1	Specify the Jog Mode. 1: TCPi of the machine coordinate system (MCS) 2: TCPi of the user coordinate system (UCS) 3: TCPi of the tool coordinate system (TCS)
CSID	Coordinate System ID	UINT	0 to 15	0	Select the user coordinate system number if you select the user coordinate system (UCS).

Name	Meaning	Data type	Valid range	Default	Description
ToolID	Tool ID	UINT	0 to 16	0	Select the ID of the robot tool.

## Output Variables

Name	Meaning	Data type	Valid range	Description
Busy	Executing	BOOL	TRUE or FALSE	TRUE when the instruction is acknowledged.
CommandAborted	Command Aborted	BOOL	TRUE or FALSE	TRUE when the instruction is aborted.
Error	Error	BOOL	TRUE or FALSE	TRUE while there is an error.
ErrorID	Error Code	WORD	*1	Contains the error code when an error occurs. A value of 16#0000 indicates normal execution.

\*1. Refer to Section 6 Troubleshooting for details.

### ● Output Variable Update Timing

Name	Timing for changing to TRUE	Timing for changing to FALSE
Busy	When <i>Enable</i> changes to TRUE.	<ul style="list-style-type: none"> <li>When <i>Error</i> changes to TRUE.</li> <li>When all <i>Enable</i> variables change to FALSE.</li> </ul>
CommandAborted	<ul style="list-style-type: none"> <li>When this instruction is aborted because another motion control instruction was executed with the Buffer Mode set to <i>Aborting</i>.</li> <li>When this instruction is canceled due to an error.</li> <li>When this instruction is executed while there is an error.</li> <li>When you start this instruction during MC_GroupStop instruction execution.</li> </ul>	<ul style="list-style-type: none"> <li>When <i>Enable</i> is TRUE and changes to FALSE.</li> <li>After one period when <i>Enable</i> is FALSE.</li> </ul>
Error	When there is an error in the execution conditions or input parameters for the instruction.	When the error is cleared.

## In-Out Variables

Name	Meaning	Data type	Valid range	Description
AxesGroup	Axes Group	_sGROUP_REF	---	Specify the axes group.

## Function

- This instruction jogs the specified robot (axes group) at the velocity set in *Velocity* (Target Velocity).
- To jog the robot in the positive direction, change *PositiveEnable* (Positive Direction Enable) variables to TRUE. To jog the robot in the negative direction, change *NegativeEnable* (Negative Direction Enable) variables to TRUE.
- The robot can move in the directions along the axes (X, Y, Z) in the Cartesian coordinate system and also in the rotation directions around the axes (Rx, Ry, Rz).
- Use Jog Mode Selection to select a coordinate system and an active TCP. Refer to *2-3-9 Robot Jogging* on page 2-44 for details on robot jogging.
- Robot jogging is a trapezoidal control which is performed according to the set velocity, acceleration rate and deceleration rate.
- The *Enable* input variables vary depending on *KinType* (Kinematics Type Selection) as shown below.

PositiveEnable/NegativeEnable		Kinematics type		
Name	Meaning	Delta3	Delta3R	Delta2
X	X Enable	Used	Used	Used
Y	Y Enable	Used	Used	Not used
Z	Z Enable	Used	Used	Used
RX	X Rotate Enable	Not used	Not used	Not used
RY	Y Rotate Enable	Not used	Not used	Not used
RZ	Z Rotate Enable	Not used	Used	Not used

PositiveEnable/ NegativeEnable	Kinematics type					
	Name	Cartesian 3D Cartesian 3D-Gantry	Cartesian 2D			Cartesian 2D-Gantry H-Bot
			Plane XY	Plane XZ	Plane YZ	
	X	Used	Used	Used	Not used	Used
	Y	Used	Used	Not used	Used	Used
	Z	Used	Not used	Used	Used	Not used
	RX	Not used	Not used			Not used
	RY	Not used	Not used			Not used
	RZ	Not used	Not used			Not used

PositiveEnable/ NegativeEnable	Kinematics type		
	Name	SCARA RRP SCARA PRR	SCARA RRP+R SCARA PRR+R
	X	Used	Used
	Y	Used	Used
	Z	Used	Used
	RX	Not used	Not used
	RY	Not used	Not used
	RZ	Not used	Used

Table above is based on machine coordinate system (MCS). When a user coordinate system is used, positions of the axis for which Rotate Enable is set are transformed into positions in the machine coordinate system. After the transformation, invalid values that are determined based on this table are ignored.



#### Precautions for Correct Use

- If a *PositiveEnable* (Positive Direction Enable) variable and a *NegativeEnable* (Negative Direction Enable) variable change from FALSE to TRUE at the same time, the *PositiveEnable* (Positive Direction Enable) variable takes priority and jogging is performed in the positive direction.
- If 180° or more is added to the axis command position in a control period, the calculation is not performed correctly according to the kinematics algorithm. In such a cases, an error occurs and the MC\_RobotJog (Axes Group Jog) instruction is aborted.

## Target Velocity

It sets the target velocity for the selected robot components. It is divided in Velocity for X, Y and Z (translation) and Velocity for RX, RY, and RZ (rotation):

Name	Data type	Valid range	Description
Velocity[0]	LREAL	Non-negative value	Target velocity for components X, Y and Z. [mm/s]
Velocity[1]	LREAL	Non-negative value	Target velocity for components RX, RY and RZ. [°/s]



#### Additional Information

If multiple TCP operations along each axes are enabled simultaneously, the velocity of TCP will be higher than the target velocity set in *Velocity*.

For example, if X, Y and Z axes are enabled simultaneously, the velocity of TCP will be Velocity[0] multiplied by  $\sqrt{3}$ .

## Acceleration Rate and Deceleration Rate

Acceleration input sets the acceleration rate to reach target velocity for the selected robot components. It is divided in Acceleration for X, Y and Z (translation) and Acceleration for RX, RY, and RZ (rotation):

Name	Data type	Valid range	Description
Acceleration[0]	LREAL	Non-negative value	Linear Acceleration rate for components X, Y and Z. [mm/s <sup>2</sup> ]
Acceleration [1]	LREAL	Non-negative value	Rotation Acceleration rate for components RX, RY and RZ. [°/s <sup>2</sup> ]

Deceleration input sets the deceleration rate to stop from target velocity for the selected robot components. It is divided in Deceleration for X, Y and Z (translation) and Deceleration for RX, RY, and RZ (rotation):

Name	Data type	Valid range	Description
Deceleration[0]	LREAL	Non-negative value	Linear Deceleration rate for components X, Y and Z. [mm/s <sup>2</sup> ]
Deceleration [1]	LREAL	Non-negative value	Rotation Deceleration rate for components RX, RY and RZ. [°/s <sup>2</sup> ]

When the *Acceleration* (Acceleration Rate) or *Deceleration* (Deceleration Rate) is 0 and you start jogging the selected component of the robot, the robot will reach the target velocity without accelerating or decelerating.

## Jog Mode Selection

Three jog modes are defined in the current specification (MACHINE, USER, and TOOL).

### ● MACHINE (*\_mcMachineMode*)

The selected TCP is moving along the Cartesian Coordinates of the machine coordinate system (MCS). Rotations are performed around the tool coordinate system (TCS) of the selected tool. It is the default jog mode.

### ● USER (*\_mcUserMode*)

The selected TCP is moving along the Cartesian Coordinates of the selected user coordinate system (UCS). Rotations are performed around the tool coordinate system (TCS) of the selected robot tool.

### ● TOOL (*\_mcToolMode*)

The selected TCP is moving along the Cartesian coordinates of tool coordinate system (TCS). Rotations are performed around the tool coordinate system (TCS) of the selected robot tool.

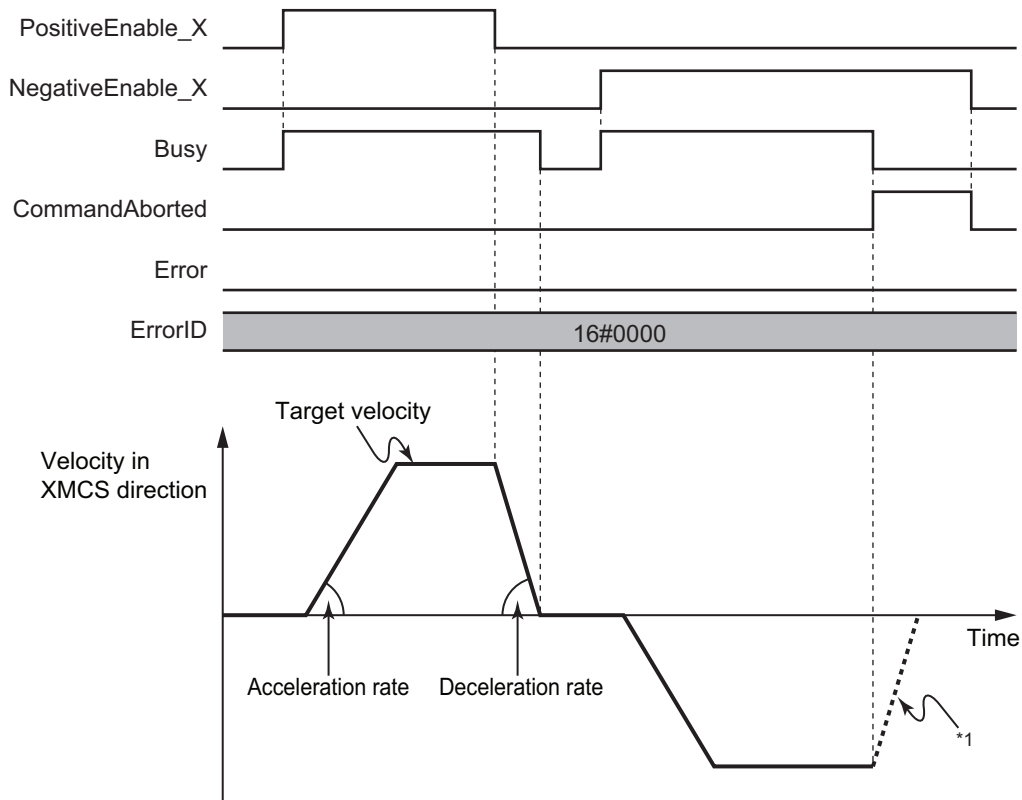
## Coordinate System ID

- If you select *\_mcUserMode* (user mode) for *JogMode* (Jog Mode Selection), you need to set the user coordinate system you use in *CSID* (Coordinate System ID). If you do not select *\_mcUserMode* (user mode) for *JogMode* (Jog Mode Selection), the variable *CSID* (Coordinate System ID) is ignored.
- The value that was set in *CSID* (Coordinate System ID) when *Enable* changed to TRUE is used while this instruction is enabled. Operation is not affected even if a different value is set in *CSID* (Coordinate System ID) while this instruction is enabled.

## Tool ID

- The TCP which is operated with this instruction is a selected tool TCP in each JogMode.
- If ToolID is set to 0, default TCP (TCP0) is used.
- The value that was set in *ToolID* (Tool ID) when *Enable* changed to TRUE is used while this instruction is enabled. Operation is not affected even if you change the input value of *ToolID* (Tool ID) to select a different tool while this instruction is enabled.

## Timing Charts



\*1. Decelerates to a stop due to an error in another instruction.

## Re-execution of Motion Control Instructions

This section describes re-execution in the same direction and in a different direction.

### ● Restarting with *Enable* in the Same Direction

If you change a *PositiveEnable* (Positive Direction Enable) variable to TRUE when it is FALSE and the axes are decelerating, the axes will begin to accelerate towards the target velocity.

This also applies to a *NegativeEnable* (Negative Direction Enable) variable.

If you change the value of *Velocity* (Target Velocity), *Acceleration* (Acceleration Rate), or *Deceleration* (Deceleration Rate) when you re-execute the instruction, operation is performed with the new value of the input variable.

Note that *Busy* (Executing) does not change to FALSE because the axes do not stop.

### ● Restarting with *Enable* in the Opposite Direction

- If you change a *NegativeEnable* (Negative Direction Enable) variable to TRUE when the robot is jogging in the positive direction, the robot reverse its direction and starts jogging in the negative direction. In this case, jogging is performed with the current input variables when a *NegativeEnable* (Negative Direction Enable) variable changed to TRUE. The input variables are *Velocity* (Target Velocity), *Acceleration* (Acceleration Rate), and *Deceleration* (Deceleration Rate).

The deceleration rate for reversing and the acceleration rate after reversing are determined by the variables that are input when *NegativeEnable* (Negative Direction Enable) changed to TRUE, regardless of the value of the Operation Selection at Reversing axis parameter.

If a *PositiveEnable* (Positive Direction Enable) variable changes to TRUE during jogging in the negative direction, the same operation is performed.

- If you change a *NegativeEnable* (Negative Direction Enable) variable to TRUE while a *PositiveEnable* (Positive Direction Enable) variable is TRUE, the robot start jogging in the negative direction. If you change the *NegativeEnable* (Negative Direction Enable) variable to FALSE after the previous operation, the axes will not jog in the positive direction even if the *PositiveEnable* (Positive Direction Enable) variable is TRUE.

To jog the axes in the positive direction, you need to change the *PositiveEnable* (Positive Direction Enable) variable to FALSE and then change it to TRUE again.

The same operation applies to the opposite case.

## Multi-execution of Motion Control Instructions

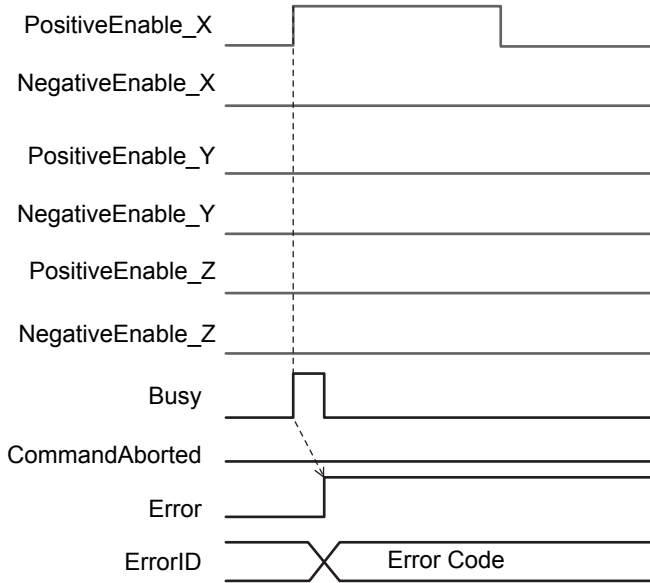
There are restrictions on the instructions that you can execute while this instruction is in execution.

Refer to 4-2 *Multi-execution of Motion Control Instructions* on page 4-98 for details on multi-execution of motion control instructions.

## Errors

If an error occurs during instruction execution, *Error* will change to TRUE.

You can find out the cause of the error by referring to the value output by *ErrorID* (Error Code).



# MC\_GroupSyncMoveAbsolute

The MC\_GroupSyncMoveAbsolute instruction cyclically outputs target positions in the axis coordinate systems set for the axes.

Instruction	Name	FB/ FUN	Graphic expression	ST expression
MC_GroupSyncMove Absolute	Axes Group Cyclic Syn- chronous Absolute Positioning	FB		<pre>MC_GroupSyncMoveAbsolute_instance (   AxesGroup :=parameter,   Execute :=parameter,   Position :=parameter,   CoordSystem :=parameter,   BufferMode :=parameter,   InPosition =&gt;parameter,   Busy =&gt;parameter,   Active =&gt;parameter,   CommandAborted =&gt;parameter,   Error =&gt;parameter,   ErrorID =&gt;parameter );</pre>

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

## Function

- This instruction is executed independently of the kinematics settings of the axes group.
- You can create an original trajectory for robot operation by using this instruction together with the MC\_InverseKin (Inverse Kinematics) instruction.

Use the following procedure to create the program.

- 1** Execute the MC\_Power (Power Servo) instruction.  
Turn ON the Servo for the robot axes.
- 2** Execute the MC\_SetKinTransform (Set Kinematics Transformation) instruction.  
Set the kinematics for the axes group.
- 3** Execute the MC\_GroupEnable (Enable Axes Group) instruction.  
Enable the axes group.
- 4** Calculate the trajectory.  
Create the target positions (Xi, Yi, Zi) in the machine coordinate system (MCS).  
After this, execute the calculation each period.
- 5** Execute the MC\_InverseKin (Inverse Kinematics) instruction.  
Transform the target position (Xi, Yi, Zi) into the target value (A0i, A1i, A2i) in the axis coordinate system (ACS).

- 6** Execute the MC\_GroupSyncMoveAbsolute (Axes Group Cyclic Synchronous Absolute Positioning) instruction.

The instruction outputs target positions (A0i, A1i, A2i) to the Servo Drive.



#### Precautions for Correct Use

---

If you perform positioning with the MC\_GroupSyncMoveAbsolute (Axes Group Cyclic Synchronous Absolute Positioning) instruction, the workspace check is not performed. Also, checks for the following variables that are set with the MC\_SetKinTransform (Set Kinematics Transformation) instruction are not performed: *MaxVelocityTCP* (Maximum Interpolation Velocity), *MaxAccelerationTCP* (Maximum Interpolation Acceleration) and *MaxDecelerationTCP* (Maximum Interpolation Deceleration).

Take sufficient care when you use the MC\_GroupSyncMoveAbsolute (Axes Group Cyclic Synchronous Absolute Positioning) instruction.

---

## 4-2 Multi-execution of Motion Control Instructions

You can execute some robot instructions during execution of another robot instruction.

The robot instructions you can execute also vary depending on their *BufferMode* (Buffer Mode Selection).

This section describes the robot instructions that you can execute and that you cannot execute for each mode.

Note that the MC\_GroupSyncMoveAbsolute (Axes Group Cyclic Synchronous Absolute Positioning) instruction is not a robot instruction. This instruction is additionally described here because you can use it to operate the robot.

### 4-2-1 Multi-execution in Aborting Mode

The following table shows the instructions executed with *BufferMode* (Buffer Mode Selection) set to *Aborting*.

The current instruction	MC_Move TimeAbsolute	MC_Sync Linear Conveyor <sup>*1</sup>	MC_Sync Linear Conveyor <sup>*2</sup>	MC_Sync Out <sup>*3</sup>	MC_Group SyncMove Absolute	MC_Robot Jog	MC_Move DirectAbsolute
Next instruction							
MC_MoveTime Absolute <sup>*4</sup>	Conditional	Conditional	Conditional	Conditional	Conditional	Conditional	Conditional
MC_SyncLinear Conveyor <sup>*4</sup>	Conditional	Conditional	Conditional	Conditional	Conditional	Conditional	Conditional
MC_SyncOut <sup>*3</sup>	Error	Error	Executable	Error	Error	Error	Error
MC_SyncOut <sup>*5</sup>	Error	Executable	Executable	Error	Error	Error	Error
MC_GroupSync MoveAbsolute	Executable	Executable	Executable	Executable	Executable	Executable	Executable
MC_RobotJog	Error	Error	Error	Error	Error	Error	Error
MC_Move DirectAbsolute	Conditional	Conditional	Conditional	Error	Conditional	Conditional	Conditional

\*1. When *Phase* is not 6.

\*2. When *Phase* is 6.

\*3. When *SyncStopType* (Synch Stop Type Selection) is set to 0 (Synchronized stop) or 2 (Synchronized stop with Deceleration).

\*4. When *BufferMode* (Buffer Mode Selection) is set to *Aborting*.

\*5. When *SyncStopType* (Synch Stop Type Selection) is set to 1 (Immediate stop).

Note 1. Executable: You can execute the next instruction. The instruction which is currently executed is aborted.

2. Conditional: You can execute the next instruction. The instruction which is currently executed is aborted. Note that command velocity output starts from the initial value (0).

3. Error: An error is detected when the next instruction is executed and the axes stop immediately.

## 4-2-2 Multi-execution in Buffered Mode

The following table shows the instructions executed with *BufferMode* (Buffer Mode Selection) set to *Buffered* (standby).

The robot instructions that are not listed in the table are the instructions for which you cannot select *Buffered* in *BufferMode* (Buffer Mode Selection).

The current instruction	MC_Move TimeAbsolute	MC_Sync Linear Conveyor <sup>*1</sup>	MC_Sync Linear Conveyor <sup>*2</sup>	MC_Sync Out <sup>*3</sup>	MC_Group SyncMove Absolute	MC_Robot Jog	MC_Move DirectAbsolute
Next instruction							
MC_MoveTime Absolute <sup>*4</sup>	Executable	Error	Error	Executable	Error	Error	Executable
MC_SyncLinear Conveyor <sup>*4</sup>	Executable	Error	Error	Executable	Error	Error	Executable
MC_Move DirectAbsolute	Executable	Error	Error	Executable	Error	Error	Executable

\*1. When *Phase* is not 6.

\*2. When *Phase* is 6.

\*3. When *SyncStopType* (Synch Stop Type Selection) is set to 0: Synchronized stop or 2: Synchronized stop with Deceleration.

\*4. When *BufferMode* (Buffer Mode Selection) is set to *Buffered* (standby).

Note 1. Executable: You can execute the next instruction. The instruction which is currently executed is aborted.

2. Error: An error is detected when the next instruction is executed and the axes stop immediately.

### 4-2-3 Multi-execution in Blending Mode

The following table shows the instructions executed with *BufferMode* (Buffer Mode Selection) set to *Blending* (mixing).

The robot instructions that are not listed in the table are the instructions for which you cannot select *Blending* in *BufferMode* (Buffer Mode Selection).

The current instruction	MC_MoveTimeAbsolute	MC_SyncLinearConveyor <sup>*1</sup>	MC_SyncLinearConveyor <sup>*2</sup>	MC_SyncOut <sup>*3</sup>	MC_GroupSyncMoveAbsolute	MC_RobotJog	MC_MoveDirectAbsolute
Next instruction							
MC_MoveTimeAbsolute <sup>*4</sup>	Conditional 2	Error	Error	Conditional 1	Error	Error	Error
MC_SyncLinearConveyor <sup>*4</sup>	Conditional 2	Error	Error	Conditional 1	Error	Error	Error

\*1. When *Phase* is not 6.

\*2. When *Phase* is 6.

\*3. When *SyncStopType* (Synch Stop Type Selection) is set to 0: Synchronized stop or 2: Synchronized stop with Deceleration.

\*4. When *BufferMode* (Buffer Mode Selection) is set to *Blending* (mixing).

Note 1. Conditional 1: You can execute the next instruction when *TransitionMode* (Transition Mode) is set to *\_mcTMStartHeight* (Start height) or *\_mcTMStartRemainingTime* (Start remaining time).

2. Conditional 2: You can execute the next instruction only when *TransitionMode* (Transition Mode) is set to *\_mcTMStartRemainingTime* (Start remaining time).

3. Error: An error is detected when the next instruction is executed and the axes stop immediately.

## **\_mcTMStartHeight (Start Height)**

If the *TransitionMode* (Transition Mode) input variable is set to *\_mcTMStartHeight* (Start height), the next instruction is blended with the current instruction.

The value of Transition Start Height set in the *TransitionParameter* (Transition Parameters) input variable determines the blended part of the instruction.

Transition starts when the travel distance in Z axis direction exceeds the Transition Start Height in the current instruction. The transition ends at the same time as the current instruction is completed only by itself.

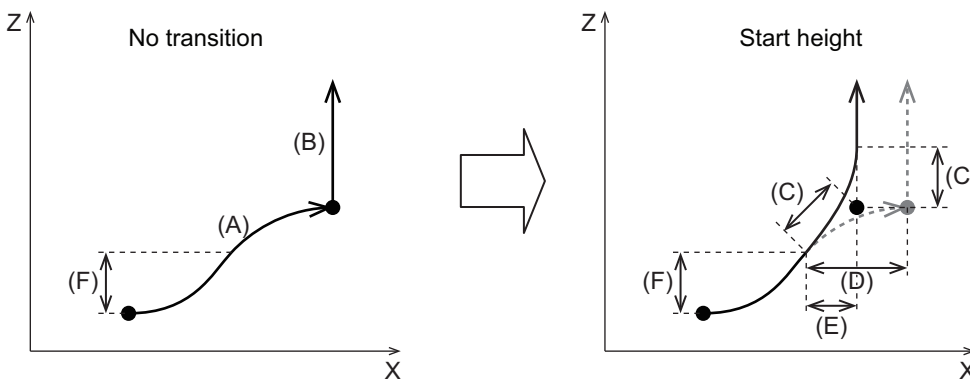


**Precautions for Correct Use**

- If the current position in the current instruction is already higher than the Transition Start Height, transition starts immediately and an observation occurs.
- If duration of the buffered instruction is the same as or shorter than the transition time, an error occurs. When MC\_SyncLinearConveyor is executed as the buffered instruction, the duration of buffered instruction is equal to  $(T1 + T2 + T3 - T3 \times Ttrans2)$ .
- When *\_mcTMStartHeight* (Start height) is used, the values of velocity, acceleration rate and deceleration rate may be greater than that of the current and next instructions that are executed separately.  
If they must be used, be careful of the operation.
- An error occurs if the next instruction is the MC\_SyncLinearConveyor (Start Conveyor Synchronization) instruction and all the following conditions are met.  
The operation direction along Z axis in the machine coordinate system (MCS) is different for the current and buffered instructions and operation that meets the above conditions is completed earlier than the transition.

● **Trajectory in Conveyor Synchronization**

The motion of current instruction in the conveyor direction is decelerated so that the instruction is blended with the next instruction.



Symbol	Description
(A)	The conveyor synchronization instruction which is currently executed
(B)	The next instruction
(C)	The time at which transition is started
(D)	Operation distance
(E)	Half operation distance
(F)	Transition Start Height

Duration of deceleration in the conveyor direction is determined by the following table.

Synchronized stop	Synchronized stop with Deceleration		
	Deceleration by DecelerationTrigger has not started yet	is on going	has already finished
Remaining time of current instruction	Specified deceleration target time	(Continues current deceleration profile)	Do not move in the conveyor direction

## **\_mcTMStartRemainingTime (Start Remaining Time)**

If the *TransitionMode* (Transition Mode) input variable is set to *\_mcTMStartRemainingTime* (Start remaining time), the next instruction is blended with the current instruction.

The value of the Transition Start Time set in the *TransitionParameter* (Transition Parameters) input variable determines the blended part of the instruction.

The transition starts a certain milliseconds earlier than the time at which the current instruction is completed. The number of milliseconds is set in Transition Start Time. The transition ends when the time set in the Transition Start Time has elapsed after the transition started.



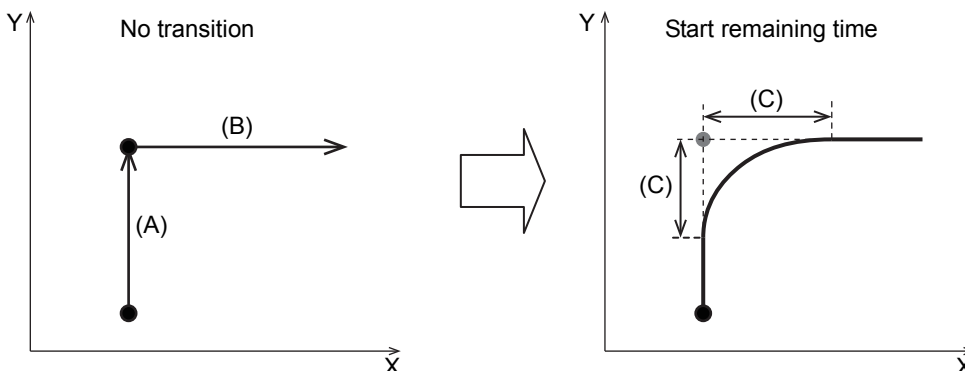
### **Precautions for Correct Use**

- If the Transition Start Time cannot be divided evenly by the primary task in the control period of the Controller, the transition time is extended to a multiple number of the control period. For example, if the Transition Start Time is set to 5 ms and the control period is 2 ms, the transition time of 5 ms is adjusted to 6 ms.
- If the remaining time for executing the current instruction is shorter than the Transition Start Time, the transition is immediately started and an observation occurs.
- If duration of the buffered instruction is the same as or shorter than the Transition Start Time, an error occurs. When MC\_SyncLinearConveyor (Start Conveyor Synchronization) is executed as the buffered instruction, the duration of buffered instruction is equal to  $(T1 + T2 + T3 - T3 \times Ttrans2)$ .
- When *\_mcTMStartRemainingTime* (Start remaining time) is used, the values of velocity, acceleration rate and deceleration rate may be greater than that of the current and next instructions that are executed separately.  
If they must be used, be careful of the operation.
- When MC\_SyncLinearConveyor (Start Conveyor Synchronization) is executed as the buffered instruction, an error (error code: 5734 hex) occurs in following conditions. In each case, the error occurs only when all conditions are met.

The current instruction	Operation status or conditions for executing next instruction
MC_SyncOut	<ul style="list-style-type: none"> <li>• TrajDistance[0] (H1) is a negative number.</li> <li>• TrajTime[0] (T1) is not 0 and is smaller than the Transition Start Time.</li> </ul>
MC_MoveTimeAbsolute	<ul style="list-style-type: none"> <li>• The robot is moving in the Z axis positive direction in the MCS.</li> <li>• TrajDistance[0] (H1) is a negative number.</li> <li>• TrajTime[0] (T1) is not 0 and is smaller than the Transition Start Time.</li> <li>• The robot is moving in the Z axis negative direction in the MCS.</li> <li>• TrajDistance[0] (H1) is a positive number.</li> <li>• TrajTime[0] (T1) is not 0 and is smaller than the Transition Start Time.</li> <li>• <i>_mcPositiveDirection</i> or <i>_mcNegativeDirection</i> is set in <i>Direction</i>. (The error does not occur if <i>_mcNoDirection</i> is set.)</li> <li>• The rotation direction of axis (A3 of Delta3R) for the current instruction is the opposite to that selected from above variables.</li> <li>• The rotation operation for the next instruction is completed before the transition ends.</li> </ul>

● **Trajectory for Time-specified Absolute Positioning**

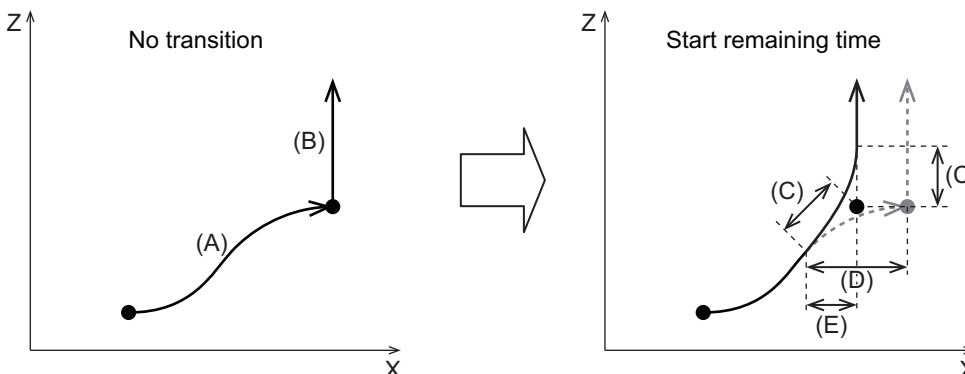
The following figure shows the TCP trajectory for two Time-specified Absolute Positioning instructions.



Symbol	Description
(A)	The current instruction
(B)	The next instruction
(C)	The time at which transition is started.

● **Trajectory in Conveyor Synchronization**

The motion of current instruction in the conveyor direction is decelerated so that the instruction is blended with the next instruction.



Symbol	Description
(A)	The conveyor synchronization instruction which is currently executed
(B)	The next instruction
(C)	The time at which transition is started
(D)	Operation distance
(E)	Half operation distance

## 4 Robot Instructions

Transition Start Time is compared with the values in the following table:

<b>Synchronized stop</b>	<b>Synchronized stop with Deceleration</b>		
	<b>Deceleration by DecelerationTrigger has not started yet</b>	<b>is on going</b>	<b>has already finished</b>
Remaining time of current instruction	Maximum value of remaining time of current instruction and specified deceleration target time	Maximum value of remaining times of movement in Z axis and deceleration	Remaining time in Z axis

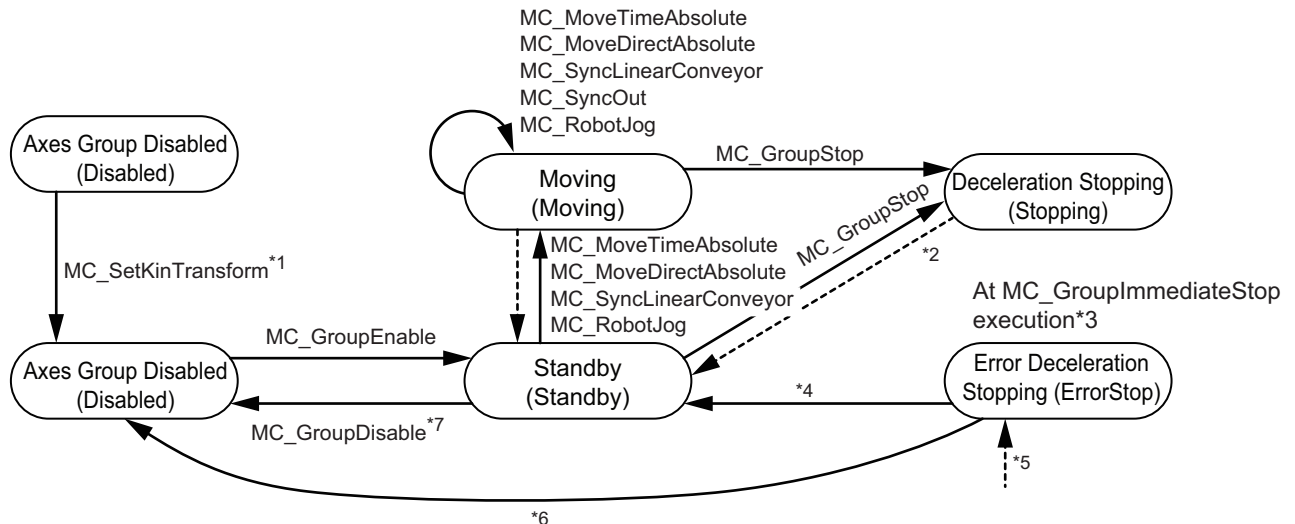
Duration of deceleration in the conveyor direction is determined by the following table.

<b>Synchronized stop</b>	<b>Synchronized stop with Deceleration</b>		
	<b>Deceleration by DecelerationTrigger has not started yet</b>	<b>is on going</b>	<b>has already finished</b>
Remaining time of current instruction	Specified deceleration target time	(Continues current deceleration profile)	Do not move in the conveyor direction

## 4-3 State Transitions of Robot Instructions

The following figure shows the axes group status (`_MC_GRP[0-31].Status`) during robot instruction execution.

The state transition of robot instructions are the same as that of the instructions for NJ501-1□□□ Units.



- \*1. Sets the kinematics transformation data for the axes group. You can use the robot instruction after this step.
- \*2. The axes group enters a *Standby* state if the *Done* output variable from the MC\_GroupStop (Group Stop) instruction is TRUE and the *Execute* input variable to the same instruction is FALSE.
- \*3. If the MC\_GroupImmediateStop (Axes Group Immediate Stop) instruction is executed, the axes group enters an *ErrorStop* state regardless of its current state.
- \*4. The axes group enters a *Standby* state if the MC\_GroupReset (Group Reset) instruction or the ResetMCErr (Reset Motion Control Error) instruction is executed while the axes group is in a *GroupEnable* state.
- \*5. An *ErrorStop* state is entered from any other state. It is entered even if an error occurs while the axes group is in a *GroupDisable* state.
- \*6. The axes group enters a *Disabled* state if the MC\_GroupReset (Group Reset) instruction or the ResetMCErr (Reset Motion Control Error) instruction is executed while the axes group is in a *GroupDisabled* state.
- \*7. The axes group stays in the same state even if the MC\_GroupDisable (Disable Axes Group) instruction is executed while in a *ErrorStop* state.

# 4-4 Sample Programming

In this section, the program for conveyor tracking is given as an example.

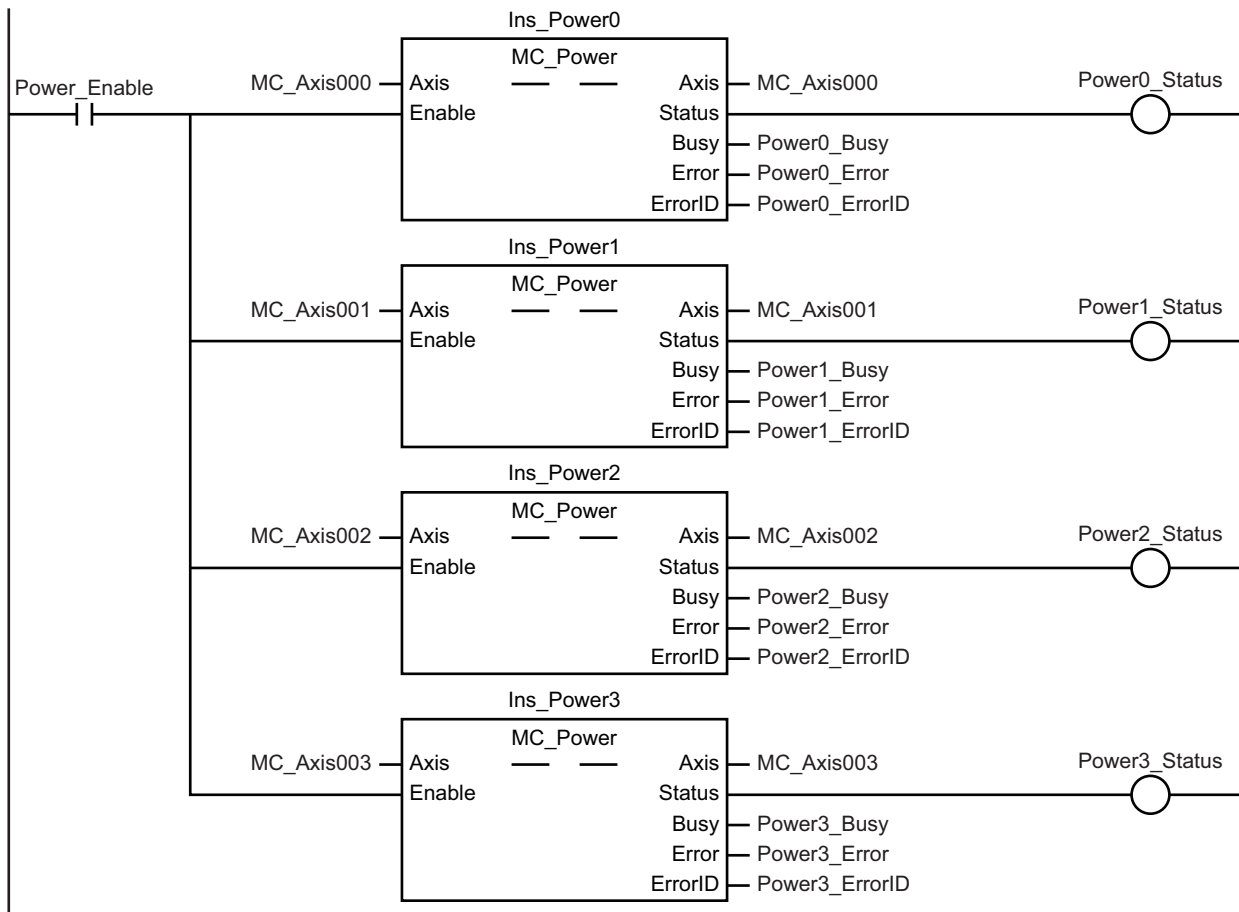
The system consists of five axes including Delta3R and a conveyor with one axis.

This example assumes that the axis settings and axes group settings are already completed on the Sysmac Studio.

System configuration		Setting	
Delta3R	Axes group	MC_Group000	
	Logical axis	A0	MC_Axis000
		A1	MC_Axis001
		A2	MC_Axis002
Conveyor axis	A3	MC_Axis003	
		MC_Axis004	

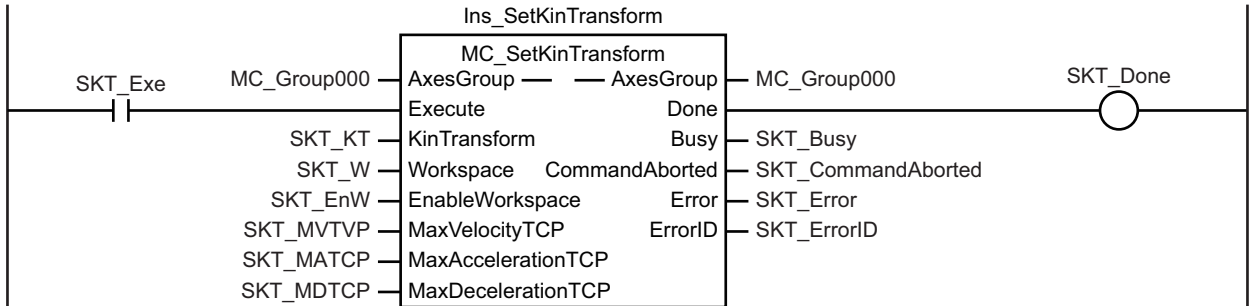
**1** Make all axes to be ready for operation.

Servo ON

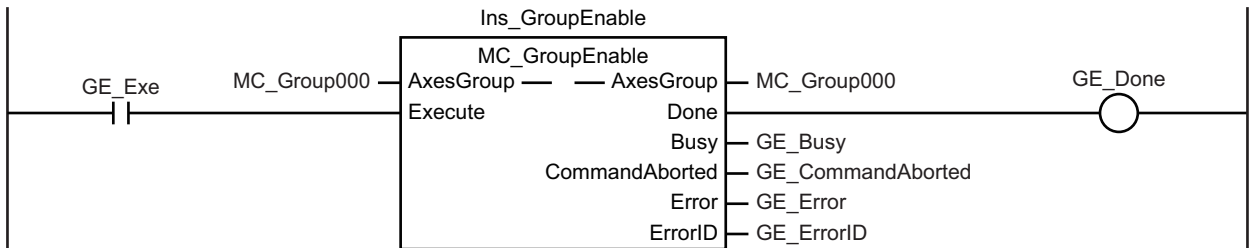


**2** Set the Delta3R kinematics transform data for the axes group 0 to enable the axes group.

Set kinematics transform



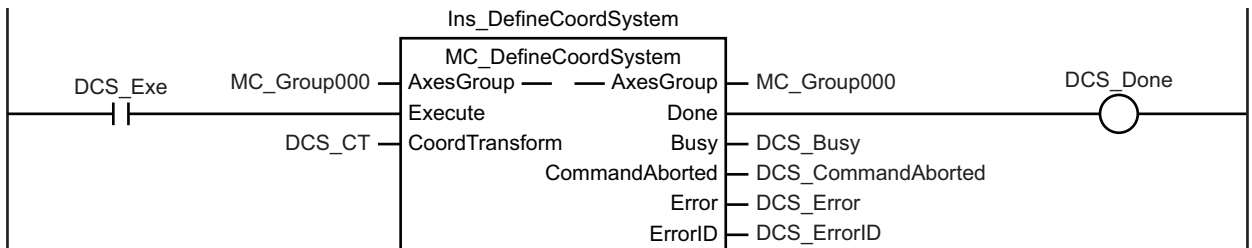
Enable axes group



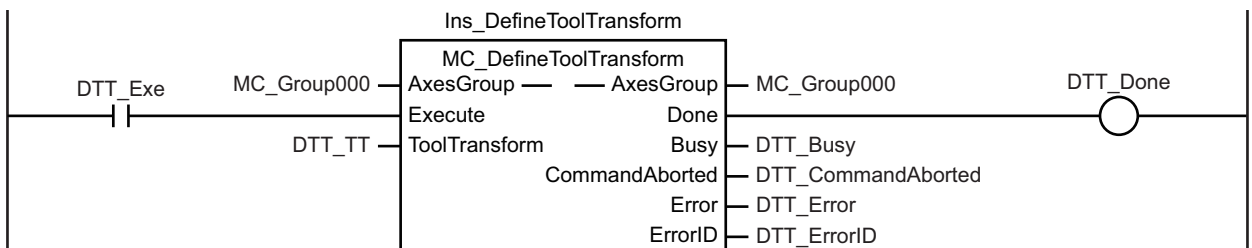
**3** Set tools and user coordinate systems (UCS) for the robot.

Change the settings as required.

Define UCS



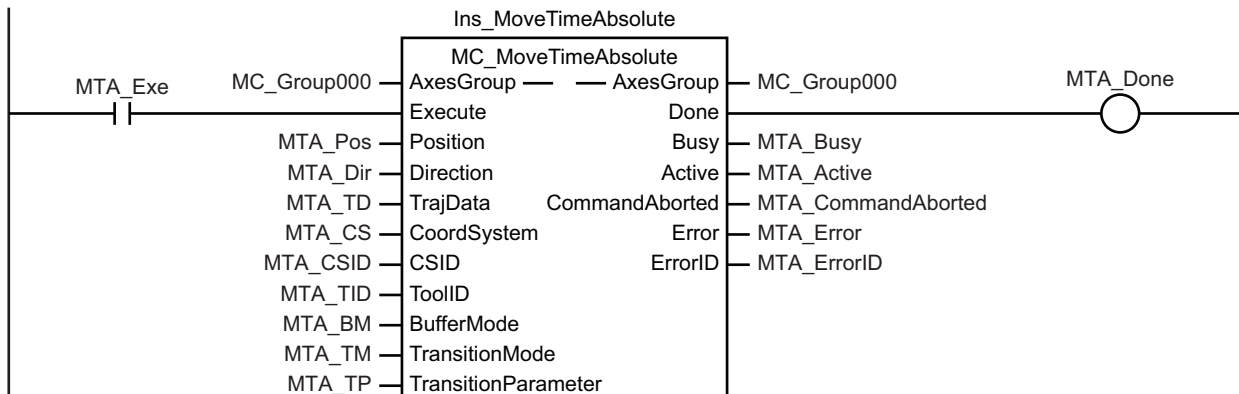
Define tool



4 Robot Instructions

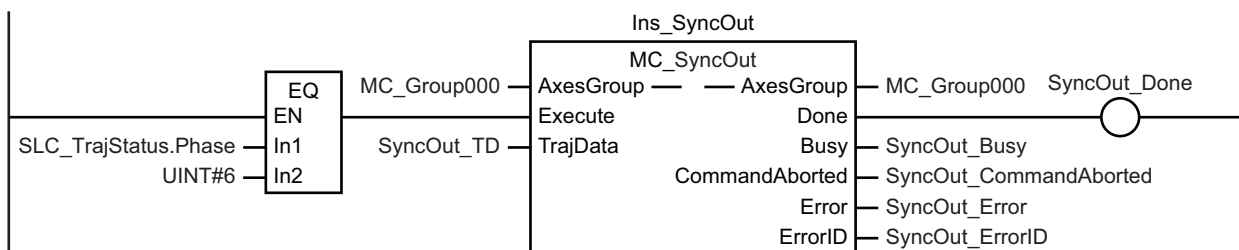
**4** Move the robot to the standby position.

Execute MC\_MoveTimeAbsolute

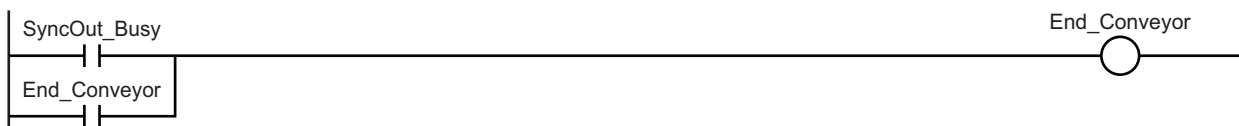


**5** Start workpiece tracking.

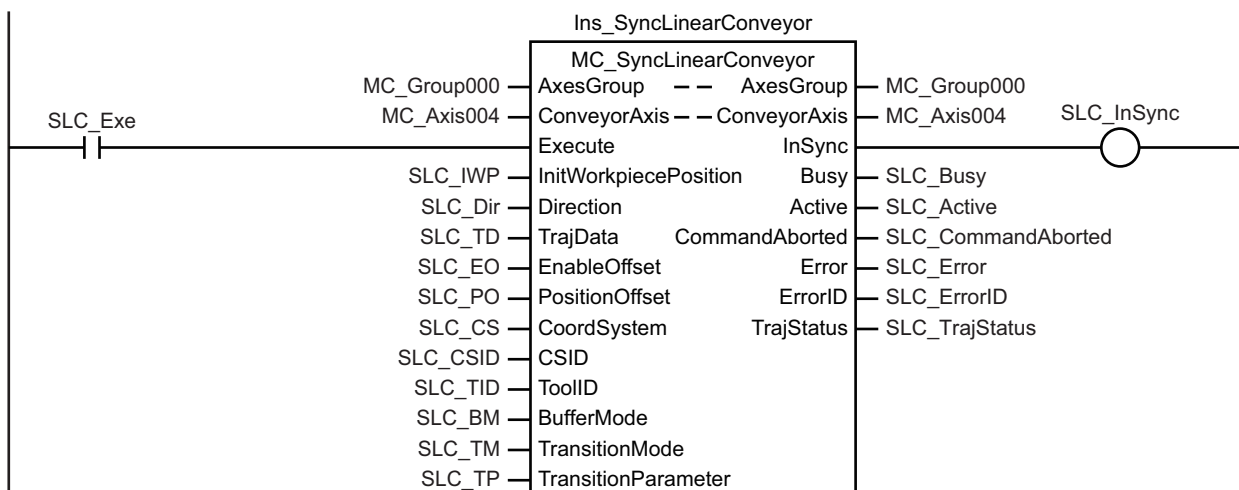
Execute MC\_SyncOut



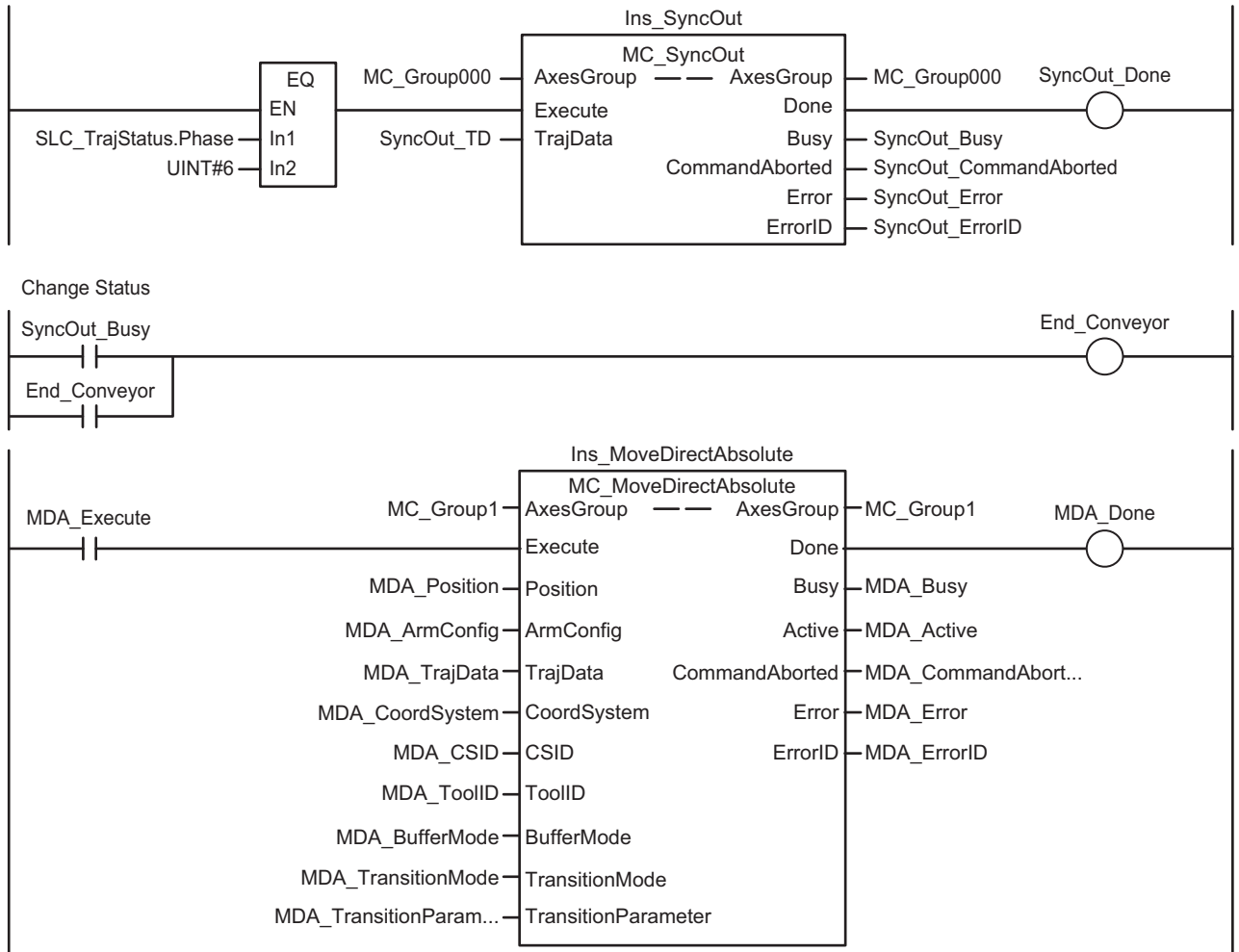
Change Status



Move Target Group



**6** Finish workpiece tracking.







# 5

## Vision & Robot Integrated Simulation

---

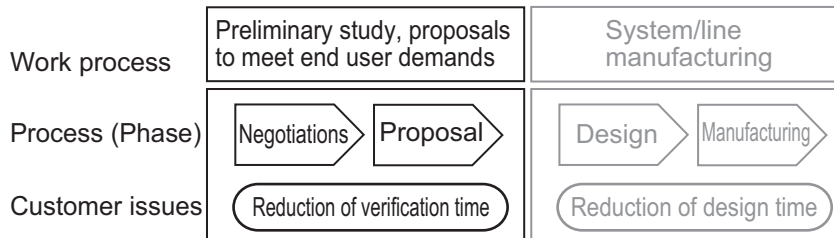
This chapter describes the Vision & Robot integrated simulation, a function to simulate the pick-and-place application where this Controller is used in combination with one or more vision sensors and robots.

---

<b>5-1 Overview of Simulation</b> .....	<b>5-2</b>
<b>5-2 Models that Support Simulation</b> .....	<b>5-3</b>
<b>5-3 Simulation Procedures</b> .....	<b>5-5</b>

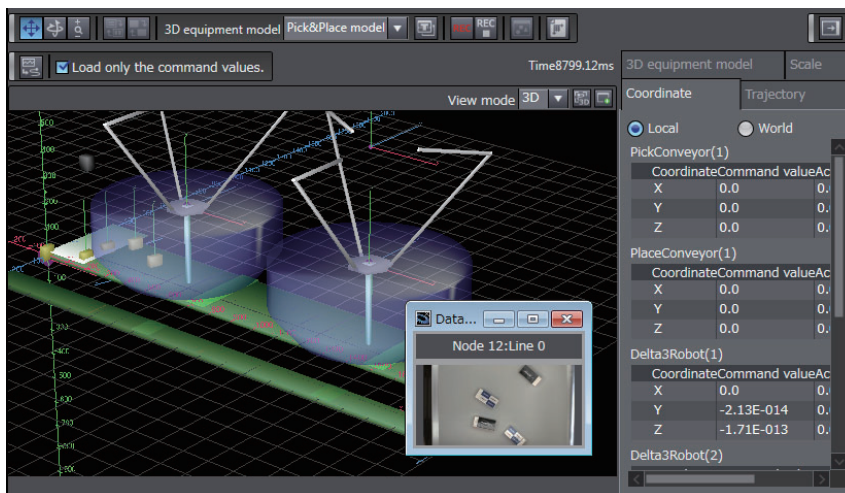
## 5-1 Overview of Simulation

When you consider newly introducing a pick-and-place application device, you need to perform verification based on the actual device environment.



You can use the Sysmac Studio's simulation function to perform this verification in the most upstream. This simulation is called "Vision & Robot integrated simulation."

In the Vision & Robot integrated simulation, you can simulate the operation of the pick-and-place application in 3D. This leads to the reduction of time for the verification in the most upstream because the actual device environment is no longer necessary.

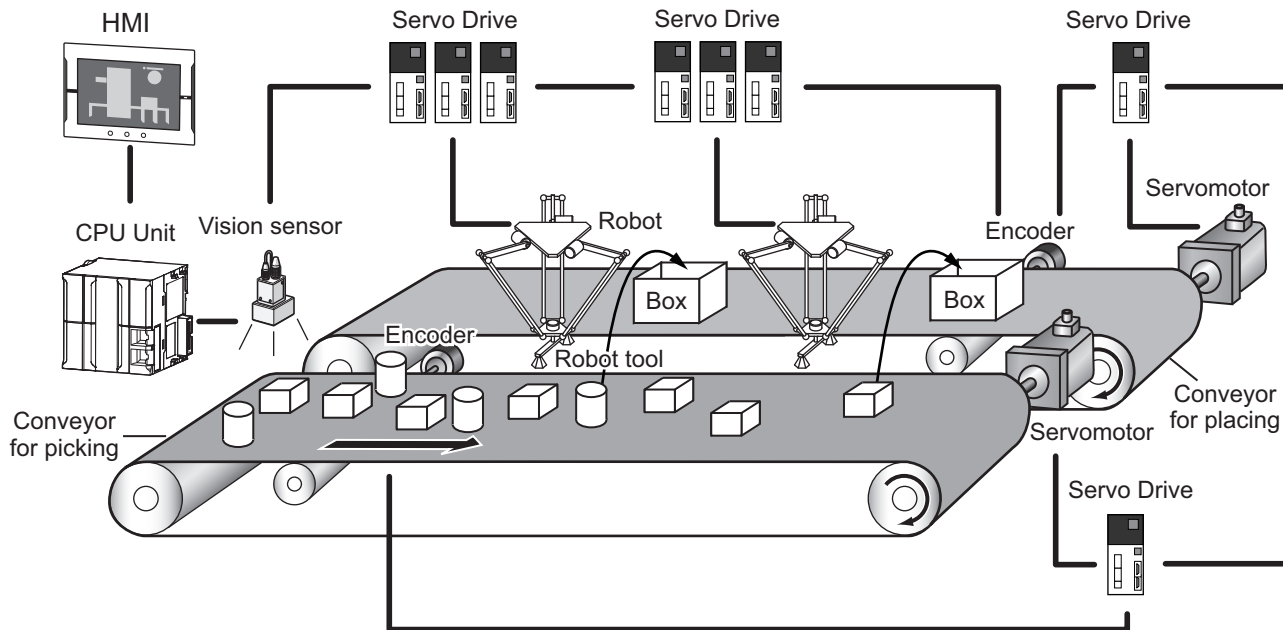


To perform the Vision & Robot integrated simulation, you need to purchase the "Robot Additional Option," a Sysmac Studio option.

Refer to *A-1 Sysmac Studio Robot Additional Option* on page A-2 for details.

# 5-2 Models that Support Simulation

This section provides an example of system configuration that enables you to perform a simulation in the pick-and-place application, where this Controller is used in combination with one or more vision sensors and robots.



5-2 Models that Support Simulation

5

### ● CPU Units

The compatible models are listed in the following table.

Series	Product	Model	Version
NJ5	Robotics	NJ501-4300/-4400/-4500	CPU Units: Version 1.10 or later Robots: Version 1.02 or later
		NJ501-4310	
	Robotics/DB Connection	NJ501-4320	

### ● Vision Sensors

The compatible models and the number of supported units are listed in the following table.

Series	Model	Qty
FH	FH-1□□□	1 sensor for each conveyor for picking
	FH-3□□□	

### ● Servo for Driving Robots

The compatible models are listed in the following table.

Series	Product	Model
G5	AC Servo Drive	R88D-KN□-ECT
	AC Servomotor	R88M-K□

### ● Servo for Driving Conveyors

A conveyor drive system consists of a combination of two elements, the conveyor drive source and the device to detect the travel distance of the conveyor. The following table shows the possible combinations of the conveyor drive source and the device to detect the travel distance of the conveyor.

	Conveyor drive source	Device to detect the conveyor travel distance
(1)	G5-series Servo Drive and Servomotor	G5-series Servomotor encoder
(2)	Any	Encoder connected to an encoder input slave (GX-EC□□□□)
(3)	Any	Encoder connected to an NX-series Position Interface Unit (NX-EC□□□□)

### ● Conveyor for Picking and Conveyor for Placing

You can specify up to the following number of conveyors for picking and conveyors for placing.

- At least one conveyor for picking
- Up to six conveyors for picking and conveyors for placing in total

### ● Robot

You can specify one to eight Delta3/Delta3R robots.

### ● Robot Tool

You can specify the vacuum-type robot tool only. The number of the robot tools that you can specify depends on the number of robots.

### ● HMI

You can perform a simulation of HMI projects along with a Vision & Robot integrated simulation. The compatible models are listed in the following table.

Series	Model
NA	NA5-15W101□
	NA5-12W101□
	NA5-9W001□
	NA5-7W001□



## Caution

Although the Sysmac Studio's simulation function simulates the operations of the Controller and vision sensors, there are differences from the Controller and vision sensors in operation and timing.

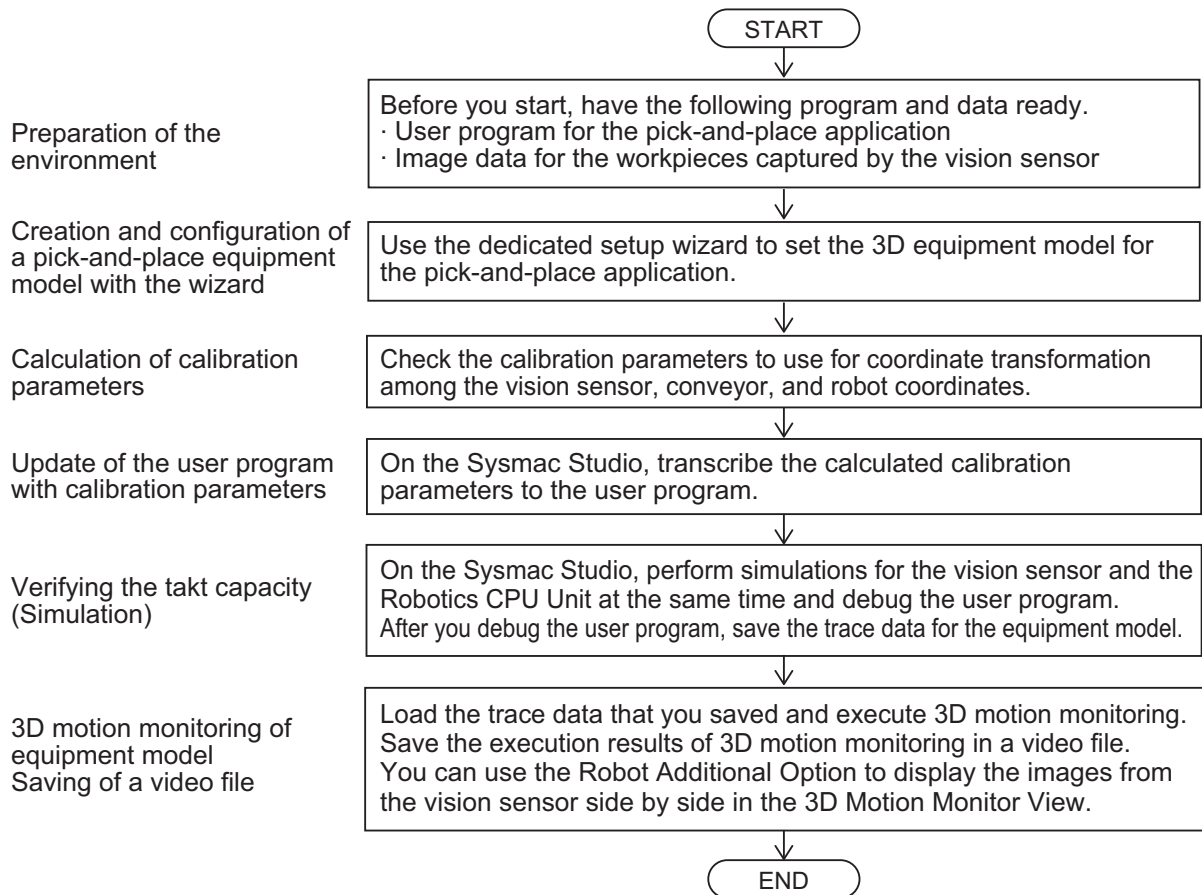
After you use the simulation function to debug the user program, always check operation and perform adjustments on the physical Controller and vision sensors before you use the user program to operate the controlled system. Accidents may occur if the controlled system performs unexpected operation.



## 5-3 Simulation Procedures

Use the following flow chart to perform a Vision & Robot integrated simulation.

For details on the procedure for each operation in the flow chart, refer to the *Vision & Robot Integrated Simulation Startup Guide* (Cat. No. Y128).



## 5 Vision & Robot Integrated Simulation



# Troubleshooting

This section describes the errors (events) that can occur and the corrections for them.

---

<b>6-1</b>	<b>Error Table</b> .....	<b>6-2</b>
<b>6-2</b>	<b>Error Description</b> .....	<b>6-23</b>
6-2-1	Interpreting Error Description .....	6-23
6-2-2	Error Descriptions .....	6-24

## 6-1 Error Table

The following table lists the errors (events) that can occur when a robot instruction is executed or when a robot instruction execution is in progress. Only the errors (events) that are directly attributed to the robot instructions are listed in this table.

The robot instructions can also be a new cause of some errors related to NJ501-1□□□ Units. For those errors, the descriptions were added to the Meaning and Assumed causes columns.

Refer to 6-2 *Error Description* on page 6-23 for details on the error codes.

Refer to the *NJ/NX-series Troubleshooting Manual* (Cat. No. W503) for codes of all events that can occur in relation with NJ-series Controllers.

The following abbreviations are used in the Level column.

Abbreviation	Name
Maj	Major fault level
Prt	Partial fault level
Min	Minor fault level
Obs	Observation
Info	Information

### Errors Related to Robot Control Function

The upper four digits of the event codes are output as the error code to *ErrorID*.

Event code	Event name	Meaning	Assumed cause	Level					Reference
				Maj	Prt	Min	Obs	Info	
54220000 hex	Target Velocity Setting Out of Range	The parameter specified for the <i>Velocity</i> input variable to a motion control instruction is out of range.	<ul style="list-style-type: none"> <li>Instruction input parameter exceeded the valid range of the input variable.</li> </ul>			√			P. 6-24
54230000 hex	Acceleration Setting Out of Range	The parameter specified for the <i>Acceleration</i> input variable to a motion control instruction is out of range.	<ul style="list-style-type: none"> <li>Instruction input parameter exceeded the valid range of the input variable.</li> </ul>			√			P. 6-25
54240000 hex	Deceleration Setting Out of Range	The parameter specified for the <i>Deceleration</i> input variable to a motion control instruction is out of range.	<ul style="list-style-type: none"> <li>Instruction input parameter exceeded the valid range of the input variable.</li> </ul>			√			P. 6-25
54250000 hex	Jerk Setting Out of Range	The parameter specified for the <i>Jerk</i> input variable to a motion control instruction is out of range.	<ul style="list-style-type: none"> <li>Instruction input parameter exceeded the valid range of the input variable.</li> </ul>			√			P. 6-26

Event code	Event name	Meaning	Assumed cause	Level					Reference
				Maj	Prt	Min	Obs	Info	
542B0000 hex	Buffer Mode Selection Out of Range	The parameter specified for the <i>BufferMode</i> input variable to a motion control instruction is out of range.	<ul style="list-style-type: none"> <li>Instruction input parameter exceeded the valid range of the input variable.</li> </ul>			√			P. 6-27
542C0000 hex	Coordinate System Selection Out of Range	The parameter specified for the <i>CoordSystem</i> input variable to a motion control instruction is out of range.	<ul style="list-style-type: none"> <li>Instruction input parameter exceeded the valid range of the input variable.</li> </ul>			√			P. 6-27
542E0000 hex	Direction Selection Out of Range	The parameter specified for the <i>Direction</i> input variable to a motion control instruction is out of range.	<ul style="list-style-type: none"> <li>Instruction input parameter exceeded the valid range of the input variable.</li> </ul>			√			P. 6-28
54320000 hex	Transition Mode Selection Out of Range	The parameter specified for the <i>TransitionMode</i> input variable to a motion control instruction is out of range.	<ul style="list-style-type: none"> <li>Instruction input parameter exceeded the valid range of the input variable.</li> <li><i>_mcAborting</i> or <i>_mcBuffered</i> was specified for <i>BufferMode</i> and <i>_mcTMNone</i> was not specified for <i>TransitionMode</i>.</li> </ul>			√			P. 6-28
543B0000 hex	Motion Control Instruction Re-execution Disabled	An attempt was made to re-execute a motion control instruction that cannot be re-executed.	<ul style="list-style-type: none"> <li>A motion control instruction that cannot be re-executed was re-executed.</li> </ul>			√			P. 6-29
543C0000 hex	Motion Control Instruction Multi-execution Disabled	Multiple functions that cannot be executed simultaneously were executed for the same target (MC common, axis, or axes group).	<ul style="list-style-type: none"> <li>Multiple functions that cannot be executed simultaneously were executed for the same target (MC common, axis, or axes group).</li> </ul>			√			P. 6-30
543E0000 hex	Instruction Cannot Be Executed during Multi-axes Coordinated Control	<ul style="list-style-type: none"> <li>A motion instruction was executed for an axis or an axes group that was in a coordinated multi-axes motion.</li> <li>A robot instruction that you cannot use for an axes group in a <i>GroupEnable</i> state was executed.</li> </ul>	<ul style="list-style-type: none"> <li>A motion instruction was executed for an axis or an axes group that was in a coordinated multi-axes motion.</li> <li>The <i>MC_SetKinTransform</i> instruction was executed for an axes group in a <i>GroupEnable</i> state.</li> </ul>			√			P. 6-31

## 6 Troubleshooting

Event code	Event name	Meaning	Assumed cause	Level					Reference
				Maj	Prt	Min	Obs	Info	
543F0000 hex	Multi-axes Coordinated Control Instruction Executed for Disabled Axes Group	A multi-axes coordinated control instruction was executed for an axes group that was in a <i>GroupDisable</i> state.	<ul style="list-style-type: none"> <li>A multi-axes coordinated control instruction was executed for an axes group that was in a <i>GroupDisable</i> state.</li> <li>One of the following instructions was executed for an axes group that was in a <i>GroupDisable</i> state. MC_MoveTimeAbsolute MC_SyncLinearConveyor MC_SyncOut MC_RobotJog</li> </ul>			√			P. 6-32
54410000 hex	Impossible Axis Operation Specified when the Servo is OFF	A motion instruction was executed for an axis for which the Servo is OFF.	<ul style="list-style-type: none"> <li>A motion instruction was executed for an axis for which the Servo is OFF.</li> <li>A zero position preset was performed with the MC_Home or MC_HomeWithParameter instruction for an axis for which EtherCAT process data communications are not established.</li> </ul>			√			P. 6-33
54420000 hex	Composition Axis Stopped Error	A motion instruction was executed for an axes group while the MC_Stop instruction was being executed for a composition axis.	<ul style="list-style-type: none"> <li>A motion instruction was executed for an axes group while the MC_Stop instruction was being executed for a composition axis.</li> </ul>			√			P. 6-34
54430000 hex	Motion Control Instruction Multi-execution Buffer Limit Exceeded	The number of motion control instructions that is buffered for Buffered or Blending Buffer Modes exceeded the buffer limit.	<ul style="list-style-type: none"> <li>An axis instruction was executed when there was already a current instruction and a buffered instruction for the same axis.</li> <li>An axes group instruction was executed when there was already eight current instructions and buffered instructions for the same axis.</li> </ul>			√			P. 6-34
54570000 hex	Motion Control Instruction Re-execution Disabled (Axes Group Specification)	An attempt was made to change the parameter for the <i>AxesGroup</i> input variable when re-executing a motion control instruction. (This input variable cannot be changed when re-executing an instruction.)	<ul style="list-style-type: none"> <li>A parameter for an input variable that cannot be changed for re-execution was changed.</li> </ul>			√			P. 6-35

Event code	Event name	Meaning	Assumed cause	Level					Reference
				Maj	Prt	Min	Obs	Info	
54610000 hex	Illegal Axes Group Specification	The axes group specified for the <i>AxesGroup</i> input variable to a motion control instruction does not exist or is not a used axes group.	<ul style="list-style-type: none"> <li>An axes group does not exist for the variable specified for the <i>AxesGroup</i> input variable to the instruction.</li> <li>The axes group specified for the <i>AxesGroup</i> input variable to the instruction is not specified as a used axes group.</li> </ul>			√			P. 6-35
54660000 hex	Instruction Execution Error with Undefined Home	High-speed homing, an interpolation instruction, or a robot instruction was executed when home was undefined.	<ul style="list-style-type: none"> <li>High-speed homing was executed when home was undefined.</li> <li>An interpolation instruction was executed for an axes group that includes an axis with no defined home.</li> <li>One of the following robot instructions was executed for an axes group that includes a logical axis with no defined home. MC_SetKinTransform MC_MoveTimeAbsolute MC_SyncLinearConveyor MC_SyncOut MC_GroupMon MC_RobotJog MC_MoveDirectAbsolute</li> </ul>			√			P. 6-36
54780000 hex	Target Position Setting Out of Range	The parameter specified for the <i>Position</i> input variable to a motion control instruction is out of range.	<ul style="list-style-type: none"> <li>Instruction input parameter exceeded the valid range of the input variable.</li> <li>The target position of a Rotary Mode axis is not within the ring setting range.</li> </ul>			√			P. 6-37
57050000 hex	Kinematics Unsupported Instruction	An attempt was made to execute an instruction that cannot be used for an axes group for which the kinematics transform was set.	<ul style="list-style-type: none"> <li>One of the following instructions was executed for an axes group for which the kinematics transform was set. MC_MoveLinear MC_MoveLinearAbsolute MC_MoveLinearRelative MC_MoveCircular2D MC_ChangeAxesInGroup</li> </ul>			√			P. 6-38

## 6 Troubleshooting

Event code	Event name	Meaning	Assumed cause	Level					Reference
				Maj	Prt	Min	Obs	Info	
57060000 hex	Axes Group Mismatch with Kinematics	The configuration elements of the specified axes group and the specified kinematics do not match.	<ul style="list-style-type: none"> <li>The number of axes in the axes group and the number of axes in the specified robot (kinematics type) do not match.</li> <li>The count mode for axes in the axes group and the count mode for the specified robot (kinematics type) do not match.</li> <li>The display unit for axes in the axes group and the display unit for the specified robot (kinematics type) do not match.</li> </ul>			√			P. 6-39
57070000 hex	Kinematics Type Out of Range	<i>KinType</i> specified for the <i>KinTransform</i> input variable to a motion control instruction is out of range.	<ul style="list-style-type: none"> <li><i>KinType</i> is outside the setting range.</li> </ul>			√			P. 6-39
57080000 hex	Kinematics Parameter Out of Range	<i>KinParam</i> or <i>ExpansionParam</i> specified for the <i>KinTransform</i> input variable to a motion control instruction is out of range.	<ul style="list-style-type: none"> <li><i>KinParam</i> is outside the setting range.</li> <li><i>ExpansionParam</i> is outside the setting range.</li> </ul>			√			P. 6-40
57090000 hex	Workspace Type Out of Range	<i>WorkspaceType</i> specified for the <i>Workspace</i> input variable to a motion control instruction is out of range.	<ul style="list-style-type: none"> <li><i>WorkspaceType</i> is outside the setting range.</li> </ul>			√			P. 6-40
570A0000 hex	Workspace Parameter Out of Range	<i>WorkspaceParam</i> specified for the <i>Workspace</i> input variable to a motion control instruction is out of range.	<ul style="list-style-type: none"> <li><i>WorkspaceParam</i> is outside the setting range.</li> </ul>			√			P. 6-41
570B0000 hex	Invalid Coordinate System Number	The coordinate system ID specified for the <i>CSID</i> input variable to a motion control instruction is out of range or not defined.	<ul style="list-style-type: none"> <li>The coordinate system ID is outside the setting range.</li> <li>The specified coordinate system ID is not defined by the <i>MC_DefineCoordSystem</i> instruction.</li> </ul>			√			P. 6-41

Event code	Event name	Meaning	Assumed cause	Level					Reference
				Maj	Prt	Min	Obs	Info	
570C0000 hex	Coordinate Transformation Parameter Out of Range	The coordinate transformation parameter <i>Pose</i> , which was specified for the <i>CoordTransform</i> input variable to a motion control instruction, is out of range.	<ul style="list-style-type: none"> <li>The coordinate transformation parameter <i>Pose</i> is outside the setting range.</li> </ul>			√			P. 6-42
570D0000 hex	Transition parameters out of range	The Transition Parameters specified for the <i>TransitionParameter</i> input variable to a motion control instruction is out of range.	<ul style="list-style-type: none"> <li><i>TransitionParameter</i> is outside the setting range.</li> </ul>			√			P. 6-42
570F0000 hex	Cannot Calculate Kinematics	The inverse kinematics or direct kinematics cannot be calculated.	<ul style="list-style-type: none"> <li>The kinematics parameter is outside the setting range.</li> <li>One of the robot axes (A0 to A2) is stopped at the position for which direct kinematics calculation cannot be performed.</li> <li>Inverse kinematics calculation cannot be performed for the target position specified in the machine coordinate system.</li> </ul>			√			P. 6-43
57100000 hex	Kinematics Transform Not Set	The kinematics transform is not set for the specified axes group.	<ul style="list-style-type: none"> <li>The kinematics transform is not set for the axes group.</li> </ul>			√			P. 6-43
57110000 hex	Target Position Out of Range	The position parameter specified as variable for the Position is out of range.	<ul style="list-style-type: none"> <li>Specified wrong value of Position input.</li> </ul>			√			P. 6-44
57120000 hex	Velocity Error Detection Value Out of Range	<i>MaxVelocity</i> specified for the <i>TrajData</i> input variable to a motion control instruction is out of range.	<ul style="list-style-type: none"> <li><i>MaxVelocity</i> is outside the setting range.</li> </ul>			√			P. 6-44
57130000 hex	Acceleration Error Detection Value Out of Range	<i>MaxAcceleration</i> specified for the <i>TrajData</i> input variable to a motion control instruction is out of range.	<ul style="list-style-type: none"> <li><i>MaxAcceleration</i> is outside the setting range.</li> </ul>			√			P. 6-45

## 6 Troubleshooting

Event code	Event name	Meaning	Assumed cause	Level					Reference
				Maj	Prt	Min	Obs	Info	
57140000 hex	Trajectory Target Time Out of Range	<i>TrajTime</i> specified for the <i>TrajData</i> input variable to a motion control instruction is out of range.	<ul style="list-style-type: none"> <li><i>TrajTime</i> is outside the setting range.</li> </ul>			√			P. 6-45
57150000 hex	Trajectory Type Out of Range	<i>MoveTrajType</i> specified for the <i>TrajData</i> input variable to a motion control instruction is out of range.	<ul style="list-style-type: none"> <li><i>MoveTrajType</i> or <i>SyncTrajType</i> is outside the setting range.</li> </ul>			√			P. 6-46
57160000 hex	Trajectory Transition Out of Range	<i>TrajTransition</i> specified for the <i>TrajData</i> input variable to a motion control instruction is out of range.	<ul style="list-style-type: none"> <li><i>TrajTransition</i> is outside the setting range.</li> </ul>			√			P. 6-46
57170000 hex	Trajectory Travel Distance Out of Range	<i>TrajDistance</i> specified for the <i>TrajData</i> input variable to a motion control instruction is out of range.	<ul style="list-style-type: none"> <li><i>TrajDistance</i> is outside the setting range.</li> </ul>			√			P. 6-47
57190000 hex	Initial Workpiece Position Outside Workspace	The position of the workpiece specified for the <i>InitWorkpiecePosition</i> input variable to a motion control instruction is outside the workspace.	<ul style="list-style-type: none"> <li><i>InitWorkpiecePosition</i> is outside the workspace.</li> </ul>			√			P. 6-47
571A0000 hex	Invalid Conveyor Axis Specified	The axis specified for the <i>ConveyorAxis</i> in-out variable to a motion control instruction is not correct.	<ul style="list-style-type: none"> <li>The specified axis is registered in the axes group which is specified for <i>AxesGroup</i>.</li> <li>The specified axis is an unused axis.</li> <li>The unit of the conveyor axis is incorrect.</li> </ul>			√			P. 6-48
571B0000 hex	Target Position Outside Workspace	The target position specified for the <i>Position</i> input variable to a motion control instruction is outside the workspace.	<ul style="list-style-type: none"> <li><i>Position</i> is outside the workspace.</li> </ul>			√			P. 6-48

Event code	Event name	Meaning	Assumed cause	Level					Reference
				Maj	Prt	Min	Obs	Info	
571C0000 hex	Cannot Cancel Synchronization	The MC_SyncOut instruction cannot be executed.	<ul style="list-style-type: none"> <li>The MC_SyncLinearConveyor instruction was not executed.</li> <li>The MC_SyncLinearConveyor instruction execution is in progress, but synchronization is not currently performed.</li> </ul>			√			P. 6-49
571E0000 hex	Too Many Kinematics	The number of kinematics exceeded the limit.	<ul style="list-style-type: none"> <li>The number kinematics set by the MC_SetKinTransform instruction exceeded the limit.</li> </ul>			√			P. 6-49
571F0000 hex	Kinematics Initialization Error	Kinematics initialization failed.	<ul style="list-style-type: none"> <li>One of the robot axes (A0 to A3) is stopped at the position for which direct kinematics calculation cannot be performed.</li> </ul>			√			P. 6-50
57320000 hex	Invalid Tool Number	The tool ID specified for the <i>ToolID</i> input variable to a motion control instruction is out of range or not defined.	<ul style="list-style-type: none"> <li><i>ToolID</i> is outside the setting range.</li> <li>The specified tool ID is not defined by the MC_DefineToolTransform instruction.</li> </ul>			√			P. 6-51
57330000 hex	Tool Parameter Out of Range	The transform parameter specified for the <i>ToolTransform</i> input variable to a motion control instruction is out of range.	<ul style="list-style-type: none"> <li>The transform parameter is outside the setting range.</li> </ul>			√			P. 6-51
57340000 hex	Unsupported Transition Data	The transition mode specified for the <i>TransitionMode</i> input variable to a motion control instruction does not support the current instruction or the buffered instruction.	<ul style="list-style-type: none"> <li>The parameters specified for the <i>TransitionParameter</i> input variable cannot be used for transition in the mode specified for the <i>TransitionMode</i> input variable.</li> <li>The trajectory data which is set for the buffered instruction cannot be used for transition in the mode specified for the <i>TransitionMode</i> input variable.</li> </ul>			√			P. 6-52
57360000 hex	Offset Not Allowed	During the MC_SyncLinearConveyor instruction execution, the offset function can be used only in Phase6.	<ul style="list-style-type: none"> <li>The <i>EnableOffset</i> input variable changed to TRUE before the MC_SyncLinearConveyor instruction operation entered Phase6.</li> </ul>			√			P. 6-52

## 6 Troubleshooting

Event code	Event name	Meaning	Assumed cause	Level					Reference
				Maj	Prt	Min	Obs	Info	
57370000 hex	Motion Control Instruction Multi-execution Disabled (Trajectory Type)	An input variable that cannot be changed was changed during multi-execution of instructions.	<ul style="list-style-type: none"> <li>• <i>MoveTrajType</i> or <i>SyncTrajType</i> for the current instruction do not match <i>MoveTrajType</i> or <i>SyncTrajType</i> for the next instruction executed with <i>Buffered</i> or <i>Blending</i>.</li> </ul>			√			P. 6-53
57390000 hex	Unsupported Transition Mode	The next instruction was executed with <i>TransitionMode</i> which does not support the combination of the current and next instructions.	<ul style="list-style-type: none"> <li>• The transition mode specified for the <i>TransitionMode</i> input variable to a motion control instruction does not support the combination of the current instruction and buffered instruction.</li> </ul>			√			P. 6-53
57440000 hex	Jog Mode Out of Range	The <i>JogMode</i> input variable to the MC_RobotJog instruction is out of range.	<ul style="list-style-type: none"> <li>• The <i>JogMode</i> input variable to the MC_RobotJog instruction is outside the setting range.</li> </ul>			√			P. 6-54
57450000 hex	Initial Workpiece Position Out of Range	The <i>InitWorkpiecePosition</i> input variable to the MC_SyncLinearConveyor instruction is out of range.	<ul style="list-style-type: none"> <li>• The <i>InitWorkpiecePosition</i> input variable to the MC_SyncLinearConveyor instruction is outside the setting range.</li> </ul>			√			P. 6-54
57460000 hex	Maximum Interpolation Velocity Out of Range	The <i>MaxVelocityTCP</i> input variable to the MC_SetKinTransform instruction is out of range.	<ul style="list-style-type: none"> <li>• The <i>MaxVelocityTCP</i> input variable to the MC_SetKinTransform instruction is outside the setting range.</li> </ul>			√			P. 6-55
57470000 hex	Maximum Interpolation Acceleration Out of Range	The <i>MaxAccelerationTCP</i> input variable to the MC_SetKinTransform instruction is out of range.	<ul style="list-style-type: none"> <li>• The <i>MaxAccelerationTCP</i> input variable to the MC_SetKinTransform instruction is outside the setting range.</li> </ul>			√			P. 6-55
57480000 hex	Maximum Interpolation Deceleration Out of Range	The <i>MaxDecelerationTCP</i> input variable to the MC_SetKinTransform instruction is out of range.	<ul style="list-style-type: none"> <li>• The <i>MaxDecelerationTCP</i> input variable to the MC_SetKinTransform instruction is outside the setting range.</li> </ul>			√			P. 6-56
574D0000 hex	Target Velocity Ratio Setting Out of Range	The parameter specified for the <i>VelocityRatio</i> input variable to a motion control instruction is out of range.	<ul style="list-style-type: none"> <li>• Instruction input parameter exceeded the valid range of the input variable.</li> </ul>			√			P. 6-56

Event code	Event name	Meaning	Assumed cause	Level					Reference
				Maj	Prt	Min	Obs	Info	
574E0000 hex	Acceleration Ratio Setting Out of Range	The parameter specified for the <i>AccelerationRatio</i> input variable to a motion control instruction is out of range.	<ul style="list-style-type: none"> <li>Instruction input parameter exceeded the valid range of the input variable.</li> </ul>			√			P. 6-57
574F0000 hex	Deceleration Ratio Setting Out of Range	The parameter specified for the <i>DecelerationRatio</i> input variable to a motion control instruction is out of range.	<ul style="list-style-type: none"> <li>Instruction input parameter exceeded the valid range of the input variable.</li> </ul>			√			P. 6-57
57500000 hex	Arm Configuration Setting Out of Range	The parameter specified for the <i>ArmConfig</i> input variable to a motion control instruction is out of range.	<ul style="list-style-type: none"> <li>Instruction input parameter exceeded the valid range of the input variable.</li> </ul>			√			P. 6-58
64430000 hex	Positive Limit Input	An instruction was executed for a motion in the positive direction when the positive limit input was ON.	<ul style="list-style-type: none"> <li>An instruction for a motion in the positive direction was executed when the positive limit input was ON, or an instruction for a motion with no direction specification was executed when the positive limit input was ON. An axes group motion control instruction was executed when the positive limit input was ON.</li> </ul>			√			P. 6-59
64440000 hex	Negative Limit Input	An instruction for a motion in the negative direction was executed when the negative limit input was ON.	<ul style="list-style-type: none"> <li>An instruction for a motion in the negative direction was executed when the negative limit input was ON, or an instruction for a motion with no direction specification was executed when the negative limit input was ON. An axes group motion control instruction was executed when the negative limit input was ON.</li> </ul>			√			P. 6-60

## 6 Troubleshooting

Event code	Event name	Meaning	Assumed cause	Level					Reference
				Maj	Prt	Min	Obs	Info	
64590000 hex	Home Undefined during Coordinated Motion	Home of the logical axis became undefined during axes group motion or while decelerating to a stop.	<ul style="list-style-type: none"> <li>The command position or actual position overflowed or underflowed for a logical axis in an axes group motion or a logical axis that was decelerating to a stop and the home definition was lost.</li> <li>A slave communications error occurred for a logical axis and home became undefined during axes group motion or while decelerating to a stop.</li> <li>A slave for a logical axis left the network or was disabled and home became undefined during axes group motion or while decelerating to a stop.</li> </ul>			√			P. 6-61
645A0000 hex	Maximum Interpolation Velocity Error	The command velocity exceeded the maximum interpolation velocity specified for the <i>MaxVelocityTCP</i> input variable to the MC_SetKinTransform instruction.	<ul style="list-style-type: none"> <li>The trajectory data for the current instruction is incorrect.</li> <li>The maximum interpolation velocity specified for the <i>MaxVelocityTCP</i> input variable to the MC_SetKinTransform instruction is incorrect or too low.</li> </ul>			√			P. 6-62
645B0000 hex	Maximum Interpolation Acceleration Error	The command acceleration rate exceeded the maximum interpolation acceleration specified for the <i>MaxAccelerationTCP</i> input variable to the MC_SetKinTransform instruction.	<ul style="list-style-type: none"> <li>The trajectory data for the current instruction is incorrect.</li> <li>The maximum interpolation acceleration specified for the <i>MaxAccelerationTCP</i> input variable to the MC_SetKinTransform instruction is incorrect or too low.</li> </ul>			√			P. 6-63
645C0000 hex	Maximum Interpolation Deceleration Error	The command deceleration rate exceeded the maximum interpolation deceleration specified for the <i>MaxDecelerationTCP</i> input variable to the MC_SetKinTransform instruction.	<ul style="list-style-type: none"> <li>The trajectory data for the current instruction is incorrect.</li> <li>The maximum interpolation deceleration specified for the <i>MaxDecelerationTCP</i> input variable to the MC_SetKinTransform instruction is incorrect or too low.</li> </ul>			√			P. 6-64

Event code	Event name	Meaning	Assumed cause	Level					Reference
				Maj	Prt	Min	Obs	Info	
67000000 hex	Command Position Outside Workspace	The command position is outside the workspace.	<ul style="list-style-type: none"> <li>The specified trajectory data is incorrect.</li> </ul>			√			P. 6-64
67010000 hex	Current Position Outside Workspace	The current position was outside the workspace when an instruction was executed.	<ul style="list-style-type: none"> <li>The current position was outside the workspace when one of the following instructions was executed. MC_MoveTimeAbsolute MC_SyncLinearConveyor MC_SyncOut MC_RobotJog MC_MoveDirectAbsolute</li> </ul>			√			P. 6-65
67020000 hex	Workpiece Synchronization Excessive Following Error	The robot could not catch up the workpiece on the conveyor.	<ul style="list-style-type: none"> <li>The conveyor position was changed before the robot reached the workpiece.</li> <li>The conveyor current position is incorrect.</li> </ul>			√			P. 6-66
67030000 hex	Velocity Error Detection	The command velocity exceeded <i>MaxVelocity</i> specified for the <i>TrajData</i> input variable to a motion control instruction.	<ul style="list-style-type: none"> <li>The combination of parameters specified for the <i>TrajData</i> input variable caused a too high command velocity.</li> <li>The parameters specified for the <i>TrajData</i> input variable are incorrect.</li> <li><i>MaxVelocity</i> is too low.</li> </ul>			√			P. 6-66
67040000 hex	Acceleration Error Detection	The command acceleration rate exceeded the <i>MaxAcceleration</i> specified for the <i>TrajData</i> input variable to a motion control instruction.	<ul style="list-style-type: none"> <li>The combination of parameters specified for the <i>TrajData</i> input variable caused a too high command acceleration rate.</li> <li>The parameters specified for the <i>TrajData</i> input variable are incorrect.</li> <li><i>MaxAcceleration</i> is too low.</li> </ul>			√			P. 6-67
67050000 hex	Command Current Velocity Limit Exceeded	The command current velocity exceeded the axis maximum velocity.	<ul style="list-style-type: none"> <li>The axis maximum velocity was exceeded.</li> </ul>			√			P. 6-67
74300000 hex	Axes Group Composition Axis Error	An error occurred for an axis in an axes group.	<ul style="list-style-type: none"> <li>An error occurred for an axis in an axes group that was in motion.</li> </ul>			√			P. 6-68

## 6 Troubleshooting

Event code	Event name	Meaning	Assumed cause	Level					Reference
				Maj	Prt	Min	Obs	Info	
77000000 hex	Conveyor Axis Position Read Error	The MC_SyncLinearConveyor instruction cannot be executed due to a conveyor axis position error.	<ul style="list-style-type: none"> <li>EtherCAT process data communications are not established for the conveyor axis.</li> <li>The slave of the conveyor axis was disconnected.</li> <li>An Absolute Encoder Current Position Calculation Failed error (64580000 hex) was detected for the conveyor axis.</li> </ul>			√			P. 6-69
94230000 hex	Transition Parameter Adjusted	The specified <i>TransitionParameter</i> input variable was adjusted before transition started.	<ul style="list-style-type: none"> <li>The transition start condition specified for <i>TransitionParameter</i> input variable for multi-execution of instructions was already exceeded by the current instruction.</li> </ul>				√		P. 6-70

## Errors Related to Robot Instructions

Event code	Event name	Meaning	Assumed cause	Level					Reference
				Maj	Prt	Min	Obs	Info	
54015422 hex	Target Velocity Setting Out of Range	The parameter specified for the <i>Velocity</i> input variable to a motion control instruction is out of range.	<ul style="list-style-type: none"> <li>Instruction input parameter exceeded the valid range of the input variable.</li> </ul>				√		P. 6-70
54015423 hex	Acceleration Setting Out of Range	The parameter specified for the <i>Acceleration</i> input variable to a motion control instruction is out of range.	<ul style="list-style-type: none"> <li>Instruction input parameter exceeded the valid range of the input variable.</li> </ul>				√		P. 6-71
54015424 hex	Deceleration Setting Out of Range	The parameter specified for the <i>Deceleration</i> input variable to a motion control instruction is out of range.	<ul style="list-style-type: none"> <li>Instruction input parameter exceeded the valid range of the input variable.</li> </ul>				√		P. 6-71
54015425 hex	Jerk Setting Out of Range	The parameter specified for the <i>Jerk</i> input variable to a motion control instruction is out of range.	<ul style="list-style-type: none"> <li>Instruction input parameter exceeded the valid range of the input variable.</li> </ul>				√		P. 6-72
5401542B hex	Buffer Mode Selection Out of Range	The parameter specified for the <i>BufferMode</i> input variable to a motion control instruction is out of range.	<ul style="list-style-type: none"> <li>Instruction input parameter exceeded the valid range of the input variable.</li> </ul>				√		P. 6-72

Event code	Event name	Meaning	Assumed cause	Level					Reference
				Maj	Prt	Min	Obs	Info	
5401542C hex	Coordinate System Selection Out of Range	The parameter specified for the <i>CoordSystem</i> input variable to a motion control instruction is out of range.	<ul style="list-style-type: none"> <li>Instruction input parameter exceeded the valid range of the input variable.</li> </ul>				√		P. 6-73
5401542E hex	Direction Selection Out of Range	The parameter specified for the <i>Direction</i> input variable to a motion control instruction is out of range.	<ul style="list-style-type: none"> <li>Instruction input parameter exceeded the valid range of the input variable.</li> </ul>				√		P. 6-73
54015432 hex	Transition Mode Selection Out of Range	The parameter specified for the <i>TransitionMode</i> input variable to a motion control instruction is out of range.	<ul style="list-style-type: none"> <li>Instruction input parameter exceeded the valid range of the input variable.</li> <li><i>_mcAborting</i> or <i>_mcBuffered</i> was specified for <i>BufferMode</i> and <i>_mcTMNone</i> was not specified for <i>TransitionMode</i>.</li> </ul>				√		P. 6-74
5401543B hex	Motion Control Instruction Re-execution Disabled	An attempt was made to re-execute a motion control instruction that cannot be re-executed.	<ul style="list-style-type: none"> <li>A motion control instruction that cannot be re-executed was re-executed.</li> </ul>				√		P. 6-75
5401543C hex	Motion Control Instruction Multi-execution Disabled	Multiple functions that cannot be executed simultaneously were executed for the same target (MC common, axis, or axes group).	<ul style="list-style-type: none"> <li>Multiple functions that cannot be executed simultaneously were executed for the same target (MC common, axis, or axes group).</li> </ul>				√		P. 6-76
5401543E hex	Instruction Cannot Be Executed during Multi-axes Coordinated Control	<ul style="list-style-type: none"> <li>A motion instruction was executed for an axis or an axes group that was in a coordinated multi-axes motion.</li> <li>A robot instruction that you cannot use for an axes group in a <i>GroupEnable</i> state was executed.</li> </ul>	<ul style="list-style-type: none"> <li>A motion instruction was executed for an axis or an axes group that was in a coordinated multi-axes motion.</li> <li>The <i>MC_SetKinTransform</i> instruction was executed for an axes group in a <i>GroupEnable</i> state.</li> </ul>				√		P. 6-77

## 6 Troubleshooting

Event code	Event name	Meaning	Assumed cause	Level					Reference
				Maj	Prt	Min	Obs	Info	
5401543F hex	Multi-axes Coordinated Control Instruction Executed for Disabled Axes Group	A multi-axes coordinated control instruction was executed for an axes group that was in a <i>GroupDisable</i> state.	<ul style="list-style-type: none"> <li>A multi-axes coordinated control instruction was executed for an axes group that was in a <i>GroupDisable</i> state.</li> <li>One of the following instructions was executed for an axes group that was in a <i>GroupDisable</i> state. MC_MoveTimeAbsolute MC_SyncLinearConveyor MC_SyncOut MC_RobotJog</li> </ul>				√		P. 6-78
54015441 hex	Impossible Axis Operation Specified when the Servo is OFF	A motion instruction was executed for an axis for which the Servo is OFF.	<ul style="list-style-type: none"> <li>A motion instruction was executed for an axis for which the Servo is OFF.</li> <li>A zero position preset was performed with the MC_Home or MC_HomeWithParameter instruction for an axis for which EtherCAT process data communications are not established.</li> </ul>				√		P. 6-79
54015442 hex	Composition Axis Stopped Error	A motion instruction was executed for an axes group while the MC_Stop instruction was being executed for a composition axis.	<ul style="list-style-type: none"> <li>A motion instruction was executed for an axes group while the MC_Stop instruction was being executed for a composition axis.</li> </ul>				√		P. 6-80
54015443 hex	Motion Control Instruction Multi-execution Buffer Limit Exceeded	The number of motion control instructions that is buffered for Buffered or Blending Buffer Modes exceeded the buffer limit.	<ul style="list-style-type: none"> <li>An axis instruction was executed when there was already a current instruction and a buffered instruction for the same axis.</li> <li>An axes group instruction was executed when there was already eight current instructions and buffered instructions for the same axis.</li> </ul>				√		P. 6-81
54015461 hex	Illegal Axes Group Specification	The axes group specified for the <i>AxesGroup</i> input variable to a motion control instruction does not exist or is not a used axes group.	<ul style="list-style-type: none"> <li>An axes group does not exist for the variable specified for the <i>AxesGroup</i> input variable to the instruction.</li> <li>The axes group specified for the <i>AxesGroup</i> input variable to the instruction is not specified as a used axes group.</li> </ul>				√		P. 6-82

Event code	Event name	Meaning	Assumed cause	Level					Reference
				Maj	Prt	Min	Obs	Info	
54015466 hex	Instruction Execution Error with Undefined Home	High-speed homing, an interpolation instruction, or a robot instruction was executed when home was undefined.	<ul style="list-style-type: none"> <li>High-speed homing was executed when home was undefined.</li> <li>An interpolation instruction was executed for an axes group that includes an axis with no defined home.</li> <li>One of the following robot instructions was executed for an axes group that includes a logical axis with no defined home. MC_SetKinTransform MC_MoveTimeAbsolute MC_SyncLinearConveyor MC_SyncOut MC_GroupMon MC_RobotJog</li> </ul>				√		P. 6-83
54015478 hex	Target Position Setting Out of Range	The parameter specified for the <i>Position</i> input variable to a motion control instruction is out of range.	<ul style="list-style-type: none"> <li>Instruction input parameter exceeded the valid range of the input variable.</li> <li>The target position of a Rotary Mode axis is not within the ring setting range.</li> </ul>				√		P. 6-84
54015706 hex	Axes Group Mismatch with Kinematics	The configuration elements of the specified axes group and the specified kinematics do not match.	<ul style="list-style-type: none"> <li>The number of axes in the axes group and the number of axes in the specified robot (kinematics type) do not match.</li> <li>The count mode for axes in the axes group and the count mode for the specified robot (kinematics type) do not match.</li> <li>The display unit for axes in the axes group and the display unit for the specified robot (kinematics type) do not match.</li> </ul>				√		P. 6-85
54015707 hex	Kinematics Type Out of Range	<i>KinType</i> specified for the <i>KinTransform</i> input variable to a motion control instruction is out of range.	<ul style="list-style-type: none"> <li><i>KinType</i> is outside the setting range.</li> </ul>				√		P. 6-86
54015708 hex	Kinematics Parameter Out of Range	<i>KinParam</i> or <i>ExpansionParam</i> specified for the <i>KinTransform</i> input variable to a motion control instruction is out of range.	<ul style="list-style-type: none"> <li><i>KinParam</i> is outside the setting range.</li> <li><i>ExpansionParam</i> is outside the setting range.</li> </ul>				√		P. 6-86

## 6 Troubleshooting

Event code	Event name	Meaning	Assumed cause	Level					Reference
				Maj	Prt	Min	Obs	Info	
54015709 hex	Workspace Type Out of Range	<i>WorkspaceType</i> specified for the <i>Workspace</i> input variable to a motion control instruction is out of range.	<ul style="list-style-type: none"> <li><i>WorkspaceType</i> is outside the setting range.</li> </ul>				√		P. 6-87
5401570A hex	Workspace Parameter Out of Range	<i>WorkspaceParam</i> specified for the <i>Workspace</i> input variable to a motion control instruction is out of range.	<ul style="list-style-type: none"> <li><i>WorkspaceParam</i> is outside the setting range.</li> </ul>				√		P. 6-87
5401570B hex	Invalid Coordinate System Number	The coordinate system ID specified for the <i>CSID</i> input variable to a motion control instruction is out of range or not defined.	<ul style="list-style-type: none"> <li>The coordinate system ID is outside the setting range.</li> <li>The specified coordinate system ID is not defined by the <i>MC_DefineCoordSystem</i> instruction.</li> </ul>				√		P. 6-88
5401570C hex	Coordinate Transformation Parameter Out of Range	The coordinate transformation parameter <i>Pose</i> , which was specified for the <i>CoordTransform</i> input variable to a motion control instruction, is out of range.	<ul style="list-style-type: none"> <li>The coordinate transformation parameter <i>Pose</i> is outside the setting range.</li> </ul>				√		P. 6-89
5401570D hex	Transition parameters out of range	The Transition Parameters specified for the <i>TransitionParameter</i> input variable to a motion control instruction is out of range.	<ul style="list-style-type: none"> <li><i>TransitionParameter</i> is outside the setting range.</li> </ul>				√		P. 6-89
54015710 hex	Kinematics Transform Not Set	The kinematics transform is not set for the specified axes group.	<ul style="list-style-type: none"> <li>The kinematics transform is not set for the axes group.</li> </ul>				√		P. 6-90
54015711 hex	Target Position Out of Range	The position parameter specified as variable for the Position is out of range.	<ul style="list-style-type: none"> <li>Specified wrong value of Position input.</li> </ul>				√		P. 6-90
54015712 hex	Velocity Error Detection Value Out of Range	<i>MaxVelocity</i> specified for the <i>TrajData</i> input variable to a motion control instruction is out of range.	<ul style="list-style-type: none"> <li><i>MaxVelocity</i> is outside the setting range.</li> </ul>				√		P. 6-91

Event code	Event name	Meaning	Assumed cause	Level					Reference
				Maj	Prt	Min	Obs	Info	
54015713 hex	Acceleration Error Detection Value Out of Range	<i>MaxAcceleration</i> specified for the <i>TrajData</i> input variable to a motion control instruction is out of range.	<ul style="list-style-type: none"> <li><i>MaxAcceleration</i> is outside the setting range.</li> </ul>				√		P. 6-91
54015714 hex	Trajectory Target Time Out of Range	<i>TrajTime</i> specified for the <i>TrajData</i> input variable to a motion control instruction is out of range.	<ul style="list-style-type: none"> <li><i>TrajTime</i> is outside the setting range.</li> </ul>				√		P. 6-92
54015715 hex	Trajectory Type Out of Range	<i>MoveTrajType</i> specified for the <i>TrajData</i> input variable to a motion control instruction is out of range.	<ul style="list-style-type: none"> <li><i>MoveTrajType</i> or <i>SyncTrajType</i> is outside the setting range.</li> </ul>				√		P. 6-92
54015716 hex	Trajectory Transition Out of Range	<i>TrajTransition</i> specified for the <i>TrajData</i> input variable to a motion control instruction is out of range.	<ul style="list-style-type: none"> <li><i>TrajTransition</i> is outside the setting range.</li> </ul>				√		P. 6-93
54015717 hex	Trajectory Travel Distance Out of Range	<i>TrajDistance</i> specified for the <i>TrajData</i> input variable to a motion control instruction is out of range.	<ul style="list-style-type: none"> <li><i>TrajDistance</i> is outside the setting range.</li> </ul>				√		P. 6-93
54015719 hex	Initial Workpiece Position Outside Workspace	The position of the workpiece specified for the <i>InitWorkpiecePosition</i> input variable to a motion control instruction is outside the workspace.	<ul style="list-style-type: none"> <li><i>InitWorkpiecePosition</i> is outside the workspace.</li> </ul>				√		P. 6-94
5401571A hex	Invalid Conveyor Axis Specified	The axis specified for the <i>ConveyorAxis</i> in-out variable to a motion control instruction is not correct.	<ul style="list-style-type: none"> <li>The specified axis is registered in the axes group which is specified for <i>AxesGroup</i>.</li> <li>The specified axis is an unused axis.</li> <li>The unit of the conveyor axis is incorrect.</li> </ul>				√		P. 6-94
5401571B hex	Target Position Outside Workspace	The target position specified for the <i>Position</i> input variable to a motion control instruction is outside the workspace.	<ul style="list-style-type: none"> <li><i>Position</i> is outside the workspace.</li> </ul>				√		P. 6-95

## 6 Troubleshooting

Event code	Event name	Meaning	Assumed cause	Level					Reference
				Maj	Prt	Min	Obs	Info	
5401571C hex	Cannot Cancel Synchronization	The MC_SyncOut instruction cannot be executed.	<ul style="list-style-type: none"> <li>The MC_SyncLinearConveyor instruction was not executed.</li> <li>The MC_SyncLinearConveyor instruction execution is in progress, but synchronization is not currently performed.</li> </ul>				√		P. 6-96
5401571E hex	Too Many Kinematics	The number of kinematics exceeded the limit.	<ul style="list-style-type: none"> <li>The number kinematics set by the MC_SetKinTransform instruction exceeded the limit.</li> </ul>				√		P. 6-97
5401571F hex	Kinematics Initialization Error	Kinematics initialization failed.	<ul style="list-style-type: none"> <li>One of the robot axes (A0 to A3) is stopped at the position for which direct kinematics calculation cannot be performed.</li> </ul>				√		P. 6-98
54015732 hex	Invalid Tool Number	The tool ID specified for the <i>ToolID</i> input variable to a motion control instruction is out of range or not defined.	<ul style="list-style-type: none"> <li><i>ToolID</i> is outside the setting range.</li> <li>The specified tool ID is not defined by the MC_DefineToolTransform instruction.</li> </ul>				√		P. 6-99
54015733 hex	Tool Parameter Out of Range	The transform parameter specified for the <i>ToolTransform</i> input variable to a motion control instruction is out of range.	<ul style="list-style-type: none"> <li>The transform parameter is outside the setting range.</li> </ul>				√		P. 6-99
54015736 hex	Offset Not Allowed	During the MC_SyncLinearConveyor instruction execution, the offset function can be used only in Phase6.	<ul style="list-style-type: none"> <li>The <i>EnableOffset</i> input variable changed to TRUE before the MC_SyncLinearConveyor instruction operation entered Phase6.</li> </ul>				√		P. 6-100
54015744 hex	Jog Mode Out of Range	The <i>JogMode</i> input variable to the MC_RobotJog instruction is out of range.	<ul style="list-style-type: none"> <li>The <i>JogMode</i> input variable to the MC_RobotJog instruction is outside the setting range.</li> </ul>				√		P. 6-100
54015745 hex	Initial Workpiece Position Out of Range	The <i>InitWorkpiecePosition</i> input variable to the MC_SyncLinearConveyor instruction is out of range.	<ul style="list-style-type: none"> <li>The <i>InitWorkpiecePosition</i> input variable to the MC_SyncLinearConveyor instruction is outside the setting range.</li> </ul>				√		P. 6-101

Event code	Event name	Meaning	Assumed cause	Level					Reference
				Maj	Prt	Min	Obs	Info	
54015746 hex	Maximum Interpolation Velocity Out of Range	The <i>MaxVelocityTCP</i> input variable to the MC_SetKinTransform instruction is out of range.	<ul style="list-style-type: none"> <li>The <i>MaxVelocityTCP</i> input variable to the MC_SetKinTransform instruction is outside the setting range.</li> </ul>				√		P. 6-102
54015747 hex	Maximum Interpolation Acceleration Out of Range	The <i>MaxAccelerationTCP</i> input variable to the MC_SetKinTransform instruction is out of range.	<ul style="list-style-type: none"> <li>The <i>MaxAccelerationTCP</i> input variable to the MC_SetKinTransform instruction is outside the setting range.</li> </ul>				√		P. 6-103
54015748 hex	Maximum Interpolation Deceleration Out of Range	The <i>MaxDecelerationTCP</i> input variable to the MC_SetKinTransform instruction is out of range.	<ul style="list-style-type: none"> <li>The <i>MaxDecelerationTCP</i> input variable to the MC_SetKinTransform instruction is outside the setting range.</li> </ul>				√		P. 6-104
5401574D hex	Target Velocity Ratio Setting Out of Range	The parameter specified for the <i>VelocityRatio</i> input variable to a motion control instruction is out of range.	<ul style="list-style-type: none"> <li>Instruction input parameter exceeded the valid range of the input variable.</li> </ul>				√		P. 6-104
5401574E hex	Acceleration Ratio Setting Out of Range	The parameter specified for the <i>AccelerationRatio</i> input variable to a motion control instruction is out of range.	<ul style="list-style-type: none"> <li>Instruction input parameter exceeded the valid range of the input variable.</li> </ul>				√		P. 6-105
5401574F hex	Deceleration Ratio Setting Out of Range	The parameter specified for the <i>DecelerationRatio</i> input variable to a motion control instruction is out of range.	<ul style="list-style-type: none"> <li>Instruction input parameter exceeded the valid range of the input variable.</li> </ul>				√		P. 6-105
54015750 hex	Arm Configuration Setting Out of Range	The parameter specified for the <i>ArmConfig</i> input variable to a motion control instruction is out of range.	<ul style="list-style-type: none"> <li>Instruction input parameter exceeded the valid range of the input variable.</li> </ul>				√		P. 6-106

## 6 Troubleshooting

Event code	Event name	Meaning	Assumed cause	Level					Reference
				Maj	Prt	Min	Obs	Info	
54016443 hex	Positive Limit Input	An instruction was executed for a motion in the positive direction when the positive limit input was ON.	<ul style="list-style-type: none"> <li>An instruction for a motion in the positive direction was executed when the positive limit input was ON, or an instruction for a motion with no direction specification was executed when the positive limit input was ON. An axes group motion control instruction was executed when the positive limit input was ON.</li> </ul>				√		P. 6-107
54016444 hex	Negative Limit Input	An instruction for a motion in the negative direction was executed when the negative limit input was ON.	<ul style="list-style-type: none"> <li>An instruction for a motion in the negative direction was executed when the negative limit input was ON, or an instruction for a motion with no direction specification was executed when the negative limit input was ON. An axes group motion control instruction was executed when the negative limit input was ON.</li> </ul>				√		P. 6-108
54016701 hex	Current Position Outside Workspace	The current position was outside the workspace when an instruction was executed.	<ul style="list-style-type: none"> <li>The current position was outside the workspace when one of the following instructions was executed. MC_MoveTimeAbsolute MC_SyncLinearConveyor MC_SyncOut MC_RobotJog</li> </ul>				√		P. 6-109

## 6-2 Error Description

This section describes the information that is given for individual errors.

### 6-2-1 Interpreting Error Description

The items that are used to describe individual errors (events) are given in the following copy of an error table.

<b>Event name</b>	Gives the name of the error.		<b>Event code</b>	Gives the code of the error.		
<b>Meaning</b>	Gives a short description of the error.					
<b>Source</b>	Gives the source of the error.		<b>Source details</b>	Gives details on the source of the error.	<b>Detection timing</b>	Tells when the error is detected.
	<b>Level</b>	Tells the level of influence on control.*1				
<b>Error attributes</b>	<b>Level</b>	Tells the level of influence on control.*1	<b>Recovery</b>	Gives the recovery method.*2	<b>Log category</b>	Tells which log the error is saved in.*3
	<b>User program</b>	Tells what will happen to execution of the user program.*4				
<b>Effects</b>	<b>User program</b>	Tells what will happen to execution of the user program.*4	<b>Operation</b>	Provides special information on the operation that results from the error.		
<b>Indicators</b>	Gives the status of the built-in EtherNet/IP port and built-in EtherCAT port indicators. Indicator status is given only for errors in the EtherCAT Master Function Module and the EtherNet/IP Function Module.					
<b>System-defined variables</b>	<b>Variable</b>	<b>Data type</b>	<b>Name</b>			
	Lists the variable names, data types, and meanings for system-defined variables that provide direct error notification, that are directly affected by the error, or that contain settings that cause the error.					
<b>Cause and correction</b>	<b>Assumed cause</b>	<b>Correction</b>	<b>Prevention</b>			
	Lists the possible causes, corrections, and preventive measures for the error.					
<b>Attached information</b>	This is the attached information that is displayed by the Sysmac Studio or an HMI.*5					
<b>Precautions/Remarks</b>	Provides precautions, restrictions, and supplemental information. If the user can set the event level, the event levels that can be set, the recovery method, operational information, and other information is also provided.					

\*1. One of the following:

- Major fault: Major fault level
- Partial fault: Partial fault level
- Minor fault: Minor fault level
- Observation
- Information

\*2. One of the following:

- Automatic recovery: Normal status is restored automatically when the cause of the error is removed.
- Error reset: Normal status is restored when the error is reset after the cause of the error is removed.
- Cycle the power supply: Normal status is restored when the power supply to the Controller is turned OFF and then back ON after the cause of the error is removed.
- Controller reset: Normal status is restored when the Controller is reset after the cause of the error is removed.
- Depends on cause: The recovery method depends on the cause of the error.

\*3. One of the following:

- System: System event log
- Access: Access event log

\*4. One of the following:

- Continues: Execution of the user program will continue.
- Stops: Execution of the user program stops.
- Starts: Execution of the user program starts.

\*5. If "System Information" is displayed, the information is for OMRON.

## 6-2-2 Error Descriptions

### Errors Related to Robot Control Function

<b>Event name</b>	Target Velocity Setting Out of Range		<b>Event code</b>	54220000 hex	
<b>Meaning</b>	The parameter specified for the <i>Velocity</i> input variable to a motion control instruction is out of range.				
<b>Source</b>	Motion Control Function Module		<b>Source details</b>	Axis/axes group	<b>Detection timing</b>
					At instruction execution
<b>Error attributes</b>	<b>Level</b>	Minor fault	<b>Recovery</b>	Error reset	<b>Log category</b>
					System
<b>Effects</b>	<b>User program</b>	Continues.	<b>Operation</b>	<p>If "axis" is given for the source details, operation is not possible for relevant axis. Relevant axis decelerates to a stop if it is in motion.</p> <p>If "axes group" is given for the source details, operation is not possible for relevant axes group. Relevant axes group decelerates to a stop if it is in motion.</p>	
<b>System-defined variables</b>	<b>Variable</b>		<b>Data type</b>		<b>Name</b>
	_MC_AX[*].MFaultLvl.Active		BOOL		Axis Minor Fault Occurrence
	_MC_GRP[*].MFaultLvl.Active		BOOL		Axes Group Minor Fault Occurrence
<b>Cause and correction</b>	<b>Assumed cause</b>		<b>Correction</b>		<b>Prevention</b>
	Instruction input parameter exceeded the valid range of the input variable.		Correct the parameter so that the valid range of the input variable is not exceeded for the relevant instruction.		Set the input parameter to the instruction so that the valid range of the input variable is not exceeded.
<b>Attached information</b>	None				
<b>Precautions/Remarks</b>	None				

<b>Event name</b>	Acceleration Setting Out of Range		<b>Event code</b>	54230000 hex	
<b>Meaning</b>	The parameter specified for the <i>Acceleration</i> input variable to a motion control instruction is out of range.				
<b>Source</b>	Motion Control Function Module		<b>Source details</b>	Axis/axes group	<b>Detection timing</b> At instruction execution
<b>Error attributes</b>	<b>Level</b>	Minor fault	<b>Recovery</b>	Error reset	<b>Log category</b> System
<b>Effects</b>	<b>User program</b>	Continues.	<b>Operation</b>	<p>If "axis" is given for the source details, operation is not possible for relevant axis. Relevant axis decelerates to a stop if it is in motion.</p> <p>If "axes group" is given for the source details, operation is not possible for relevant axes group. Relevant axes group decelerates to a stop if it is in motion.</p>	
<b>System-defined variables</b>	<b>Variable</b>		<b>Data type</b>		<b>Name</b>
	_MC_AX[*].MFaultLvl.Active		BOOL		Axis Minor Fault Occurrence
	_MC_GRP[*].MFaultLvl.Active		BOOL		Axes Group Minor Fault Occurrence
<b>Cause and correction</b>	<b>Assumed cause</b>		<b>Correction</b>		<b>Prevention</b>
	Instruction input parameter exceeded the valid range of the input variable.		Correct the parameter so that the valid range of the input variable is not exceeded for the relevant instruction.		Set the input parameter to the instruction so that the valid range of the input variable is not exceeded.
<b>Attached information</b>	None				
<b>Precautions/Remarks</b>	None				

<b>Event name</b>	Deceleration Setting Out of Range		<b>Event code</b>	54240000 hex	
<b>Meaning</b>	The parameter specified for the <i>Deceleration</i> input variable to a motion control instruction is out of range.				
<b>Source</b>	Motion Control Function Module		<b>Source details</b>	Axis/axes group	<b>Detection timing</b> At instruction execution
<b>Error attributes</b>	<b>Level</b>	Minor fault	<b>Recovery</b>	Error reset	<b>Log category</b> System
<b>Effects</b>	<b>User program</b>	Continues.	<b>Operation</b>	<p>If "axis" is given for the source details, operation is not possible for relevant axis. Relevant axis decelerates to a stop if it is in motion.</p> <p>If "axes group" is given for the source details, operation is not possible for relevant axes group. Relevant axes group decelerates to a stop if it is in motion.</p>	
<b>System-defined variables</b>	<b>Variable</b>		<b>Data type</b>		<b>Name</b>
	_MC_AX[*].MFaultLvl.Active		BOOL		Axis Minor Fault Occurrence
	_MC_GRP[*].MFaultLvl.Active		BOOL		Axes Group Minor Fault Occurrence
<b>Cause and correction</b>	<b>Assumed cause</b>		<b>Correction</b>		<b>Prevention</b>
	Instruction input parameter exceeded the valid range of the input variable.		Correct the parameter so that the valid range of the input variable is not exceeded for the relevant instruction.		Set the input parameter to the instruction so that the valid range of the input variable is not exceeded.
<b>Attached information</b>	None				
<b>Precautions/Remarks</b>	None				

## 6 Troubleshooting

<b>Event name</b>	Jerk Setting Out of Range		<b>Event code</b>	54250000 hex	
<b>Meaning</b>	The parameter specified for the <i>Jerk</i> input variable to a motion control instruction is out of range.				
<b>Source</b>	PLC Function Module		<b>Source details</b>	Instruction	<b>Detection timing</b> At instruction execution
<b>Error attributes</b>	<b>Level</b>	Observation	<b>Recovery</b>	---	<b>Log category</b> System
<b>Effects</b>	<b>User program</b>	Continues.	<b>Operation</b>	The relevant instruction will end according to specifications.	
<b>System-defined variables</b>	<b>Variable</b>		<b>Data type</b>		<b>Name</b>
	_MC_AX[*].MFaultLvl.Active		BOOL		Axis Minor Fault Occurrence
	_MC_GRP[*].MFaultLvl.Active		BOOL		Axes Group Minor Fault Occurrence
<b>Cause and correction</b>	<b>Assumed cause</b>		<b>Correction</b>		<b>Prevention</b>
	Instruction input parameter exceeded the valid range of the input variable.		Correct the parameter so that the valid range of the input variable is not exceeded for the relevant instruction.		Set the input parameter to the instruction so that the valid range of the input variable is not exceeded.
<b>Attached information</b>	<p>Attached Information 1: Error Location</p> <p>Attached Information 2: Error Location Details (Rung Number). For a program section, the rung number from the start of the section is given. For ST, the line number is given.</p> <p>Attached Information 3: Names of the Instruction and Instruction Instance Where the Error Occurred. If there is more than one possible instruction, information is given on all of them. Nothing is given if the instruction cannot be identified.</p> <p>Attached Information 4: Expansion Error Code (<i>ErrorIDEx</i>)</p>				
<b>Precautions/Remarks</b>	If a program is changed after an error occurs, the attached information that is displayed may not be correct.				

<b>Event name</b>	Buffer Mode Selection Out of Range		<b>Event code</b>	542B0000 hex	
<b>Meaning</b>	The parameter specified for the <i>BufferMode</i> input variable to a motion control instruction is out of range.				
<b>Source</b>	Motion Control Function Module		<b>Source details</b>	Axis/axes group	<b>Detection timing</b> At instruction execution
<b>Error attributes</b>	<b>Level</b>	Minor fault	<b>Recovery</b>	Error reset	<b>Log category</b> System
<b>Effects</b>	<b>User program</b>	Continues.	<b>Operation</b>	<p>If "axis" is given for the source details, operation is not possible for relevant axis. Relevant axis decelerates to a stop if it is in motion.</p> <p>If "axes group" is given for the source details, operation is not possible for relevant axes group.</p> <p>Relevant axes group decelerates to a stop if it is in motion.</p> <p>If the axes group is in motion for a robot instruction, it stops immediately.</p>	
<b>System-defined variables</b>	<b>Variable</b>		<b>Data type</b>		<b>Name</b>
	_MC_AX[*].MFAultLvl.Active		BOOL		Axis Minor Fault Occurrence
		_MC_GRP[*].MFAultLvl.Active		BOOL	
		Axes Group Minor Fault Occurrence			
<b>Cause and correction</b>	<b>Assumed cause</b>		<b>Correction</b>		<b>Prevention</b>
	Instruction input parameter exceeded the valid range of the input variable.		Correct the parameter so that the valid range of the input variable is not exceeded for the instruction.		Set the input parameter to the instruction so that the valid range of the input variable is not exceeded.
<b>Attached information</b>	None				
<b>Precautions/Remarks</b>	None				

<b>Event name</b>	Coordinate System Selection Out of Range		<b>Event code</b>	542C0000 hex	
<b>Meaning</b>	The parameter specified for the <i>CoordSystem</i> input variable to a motion control instruction is out of range.				
<b>Source</b>	Motion Control Function Module		<b>Source details</b>	Axes group	<b>Detection timing</b> At instruction execution
<b>Error attributes</b>	<b>Level</b>	Minor fault	<b>Recovery</b>	Error reset	<b>Log category</b> System
<b>Effects</b>	<b>User program</b>	Continues.	<b>Operation</b>	<p>Operation is not possible for relevant axes group.</p> <p>Relevant axes group decelerates to a stop if it is in motion.</p> <p>If the axes group is in motion for a robot instruction, it stops immediately.</p>	
<b>System-defined variables</b>	<b>Variable</b>		<b>Data type</b>		<b>Name</b>
	_MC_GRP[*].MFAultLvl.Active		BOOL		Axes Group Minor Fault Occurrence
<b>Cause and correction</b>	<b>Assumed cause</b>		<b>Correction</b>		<b>Prevention</b>
	Instruction input parameter exceeded the valid range of the input variable.		Correct the parameter so that the valid range of the input variable is not exceeded for the instruction.		Set the input parameter to the instruction so that the valid range of the input variable is not exceeded.
<b>Attached information</b>	None				
<b>Precautions/Remarks</b>	None				

## 6 Troubleshooting

<b>Event name</b>	Direction Selection Out of Range		<b>Event code</b>	542E0000 hex	
<b>Meaning</b>	The parameter specified for the <i>Direction</i> input variable to a motion control instruction is out of range.				
<b>Source</b>	Motion Control Function Module		<b>Source details</b>	Axis	<b>Detection timing</b> At instruction execution
<b>Error attributes</b>	<b>Level</b>	Minor fault	<b>Recovery</b>	Error reset	<b>Log category</b> System
<b>Effects</b>	<b>User program</b>	Continues.	<b>Operation</b>	Operation is not possible for relevant axis. Relevant axis decelerates to a stop if it is in motion.	
<b>System-defined variables</b>	<b>Variable</b>		<b>Data type</b>		<b>Name</b>
	_MC_AX[*].MFAultLvl.Active		BOOL		Axis Minor Fault Occurrence
<b>Cause and correction</b>	<b>Assumed cause</b>		<b>Correction</b>		<b>Prevention</b>
	Instruction input parameter exceeded the valid range of the input variable.		Correct the parameter so that the valid range of the input variable is not exceeded for the relevant instruction.		Set the input parameter to the instruction so that the valid range of the input variable is not exceeded.
<b>Attached information</b>	None				
<b>Precautions/Remarks</b>	None				

<b>Event name</b>	Transition Mode Selection Out of Range		<b>Event code</b>	54320000 hex	
<b>Meaning</b>	The parameter specified for the <i>TransitionMode</i> input variable to a motion control instruction is out of range.				
<b>Source</b>	Motion Control Function Module		<b>Source details</b>	Axes group	<b>Detection timing</b> At instruction execution
<b>Error attributes</b>	<b>Level</b>	Minor fault	<b>Recovery</b>	Error reset	<b>Log category</b> System
<b>Effects</b>	<b>User program</b>	Continues.	<b>Operation</b>	Operation is not possible for relevant axes group. Relevant axes group decelerates to a stop if it is in motion. If the axes group is in motion for a robot instruction, it stops immediately.	
<b>System-defined variables</b>	<b>Variable</b>		<b>Data type</b>		<b>Name</b>
	_MC_GRP[*].MFAultLvl.Active		BOOL		Axes Group Minor Fault Occurrence
<b>Cause and correction</b>	<b>Assumed cause</b>		<b>Correction</b>		<b>Prevention</b>
	Instruction input parameter exceeded the valid range of the input variable.		Correct the parameter so that the valid range of the input variable is not exceeded for the instruction.		Set the input parameter to the instruction so that the valid range of the input variable is not exceeded.
	_mcAborting or _mcBuffered was specified for <i>BufferMode</i> and _mcTMNone was not specified for <i>TransitionMode</i> .		If you specify _mcAborting or _mcBuffered for <i>BufferMode</i> , specify _mcTMNone for <i>TransitionMode</i> . If you specify the values other than _mcTMNone for <i>TransitionMode</i> , specify _mcBlendingLow, _mcBlendingPrevious, _mcBlendingNext, or _mcBlendingHigh for <i>BufferMode</i> .		If you specify _mcAborting or _mcBuffered for <i>BufferMode</i> , specify _mcTMNone for <i>TransitionMode</i> . If you specify the values other than _mcTMNone for <i>TransitionMode</i> , specify _mcBlendingLow, _mcBlendingPrevious, _mcBlendingNext, or _mcBlendingHigh for <i>BufferMode</i> .
<b>Attached information</b>	None				
<b>Precautions/Remarks</b>	None				

<b>Event name</b>	Motion Control Instruction Re-execution Disabled		<b>Event code</b>	543B0000 hex		
<b>Meaning</b>	An attempt was made to re-execute a motion control instruction that cannot be re-executed.					
<b>Source</b>	Motion Control Function Module		<b>Source details</b>	MC common, axis, or axes group	<b>Detection timing</b>	At instruction execution
<b>Error attributes</b>	<b>Level</b>	Minor fault	<b>Recovery</b>	Error reset	<b>Log category</b>	System
<b>Effects</b>	<b>User program</b>	Continues.	<b>Operation</b>	<p>If "MC Common" is given for the source details, operation is not affected.</p> <p>If "axis" is given for the source details, operation is not possible for relevant axis. Relevant axis decelerates to a stop if it is in motion.</p> <p>If "axes group" is given for the source details, operation is not possible for relevant axes group. Relevant axes group decelerates to a stop if it is in motion.</p> <p>If the axes group is in motion for a robot instruction, it stops immediately.</p>		
<b>System-defined variables</b>	<b>Variable</b>		<b>Data type</b>		<b>Name</b>	
	_MC_COM.MFaultLvl.Active		BOOL		MC Common Minor Fault Occurrence	
	_MC_AX[*].MFaultLvl.Active		BOOL		Axis Minor Fault Occurrence	
	_MC_GRP[*].MFaultLvl.Active		BOOL		Axes Group Minor Fault Occurrence	
<b>Cause and correction</b>	<b>Assumed cause</b>		<b>Correction</b>		<b>Prevention</b>	
	A motion control instruction that cannot be re-executed was re-executed.		Correct the program so that the <i>Execute</i> input variable does not change to TRUE until the <i>Busy</i> output variable from the instruction changes to FALSE.		When you use instructions that cannot be re-executed, include a condition for the <i>Execute</i> input variable so that it does not change to TRUE unless the <i>Busy</i> output variable for the previous instruction is FALSE. Or, stop the instruction before executing it again.	
<b>Attached information</b>	None					
<b>Precautions/Remarks</b>	None					

## 6 Troubleshooting

<b>Event name</b>	Motion Control Instruction Multi-execution Disabled		<b>Event code</b>	543C0000 hex			
<b>Meaning</b>	Multiple functions that cannot be executed simultaneously were executed for the same target (MC common, axis, or axes group).						
<b>Source</b>	Motion Control Function Module		<b>Source details</b>	MC common, axis, or axes group	<b>Detection timing</b>	At instruction execution or at multi-execution of instructions	
<b>Error attributes</b>	<b>Level</b>	Minor fault	<b>Recovery</b>	Error reset	<b>Log category</b>	System	
<b>Effects</b>	<b>User program</b>	Continues.	<b>Operation</b>	<p>If "MC Common" is given for the source details, operation is not affected.</p> <p>If "axis" is given for the source details, operation is not possible for relevant axis. Relevant axis decelerates to a stop if it is in motion.</p> <p>If "axes group" is given for the source details, operation is not possible for relevant axes group.</p> <p>Relevant axes group decelerates to a stop if it is in motion.</p> <p>If the axes group is in motion for a robot instruction, it stops immediately.</p>			
<b>System-defined variables</b>	<b>Variable</b>		<b>Data type</b>		<b>Name</b>		
	_MC_COM.MFaultLvl.Active		BOOL		MC Common Minor Fault Occurrence		
	_MC_AX[*].MFaultLvl.Active		BOOL		Axis Minor Fault Occurrence		
		_MC_GRP[*].MFaultLvl.Active		BOOL		Axes Group Minor Fault Occurrence	
<b>Cause and correction</b>	<b>Assumed cause</b>		<b>Correction</b>		<b>Prevention</b>		
	Multiple functions that cannot be executed simultaneously were executed for the same target (MC common, axis, or axes group).		Check the specifications of multi-execution of instructions for this instruction and do not execute instructions that cannot be executed at the same time.		Check the specifications for multi-execution of instructions for the instruction and do not execute instructions that cannot be executed at the same time.		
<b>Attached information</b>	None						
<b>Precautions/Remarks</b>	None						

<b>Event name</b>	Instruction Cannot Be Executed during Multi-axes Coordinated Control		<b>Event code</b>	543E0000 hex	
<b>Meaning</b>	<ul style="list-style-type: none"> <li>A motion instruction was executed for an axis or an axes group that was in a coordinated multi-axes motion.</li> <li>A robot instruction that you cannot use for an axes group in a <i>GroupEnable</i> state was executed.</li> </ul>				
<b>Source</b>	Motion Control Function Module	<b>Source details</b>	Axis/axes group	<b>Detection timing</b>	At instruction execution or at multi-execution of instructions
<b>Error attributes</b>	<b>Level</b>	Minor fault	<b>Recovery</b>	Error reset	<b>Log category</b> System
<b>Effects</b>	<b>User program</b>	Continues.	<b>Operation</b>	<p>If "axis" is given for the source details, operation is not possible for relevant axis. Relevant axis decelerates to a stop if it is in motion.</p> <p>If "axes group" is given for the source details, operation is not possible for relevant axes group. Relevant axes group decelerates to a stop if it is in motion.</p> <p>If the axes group is in motion for a robot instruction, it stops immediately.</p>	
<b>System-defined variables</b>	<b>Variable</b>		<b>Data type</b>		<b>Name</b>
	_MC_AX[*].MFAultLvl.Active		BOOL		Axis Minor Fault Occurrence
		_MC_GRP[*].MFAultLvl.Active		BOOL Axes Group Minor Fault Occurrence	
<b>Cause and correction</b>	<b>Assumed cause</b>		<b>Correction</b>		<b>Prevention</b>
	A motion instruction was executed for an axis or an axes group that was in a coordinated multi-axes motion.		Correct the program so that axis operation instructions are executed only for axes or axes groups that are not in coordinated multi-axes motion.		Execute axis operation instructions only for axes or axes groups that are not in coordinated multi-axes motion.
		The MC_SetKinTransform (Set Kinematics Transformation) instruction was executed for an axes group in a <i>GroupEnable</i> state.		Correct the program so that the instruction is executed only when the axes group is in a <i>GroupDisable</i> state.	
				Execute the instruction only when the axes group is in a <i>GroupDisable</i> state.	
<b>Attached information</b>	None				
<b>Precautions/Remarks</b>	None				

## 6 Troubleshooting

<b>Event name</b>	Multi-axes Coordinated Control Instruction Executed for Disabled Axes Group		<b>Event code</b>	543F0000 hex		
<b>Meaning</b>	A multi-axes coordinated control instruction was executed for an axes group that was in a <i>GroupDisable</i> state.					
<b>Source</b>	Motion Control Function Module		<b>Source details</b>	Axes group	<b>Detection timing</b>	At instruction execution
<b>Error attributes</b>	<b>Level</b>	Minor fault	<b>Recovery</b>	Error reset	<b>Log category</b>	System
<b>Effects</b>	<b>User program</b>	Continues.	<b>Operation</b>	Operation is not possible for relevant axes group. Relevant axes group decelerates to a stop if it is in motion. If the axes group is in motion for a robot instruction, it stops immediately.		
<b>System-defined variables</b>	<b>Variable</b>	<b>Data type</b>		<b>Name</b>		
	_MC_GRP[*].MFaultLvl.Active	BOOL		Axes Group Minor Fault Occurrence		
<b>Cause and correction</b>	<b>Assumed cause</b>		<b>Correction</b>		<b>Prevention</b>	
	<p>A multi-axes coordinated control instruction was executed for an axes group that was in a <i>GroupDisable</i> state.</p> <p>One of the following instructions was executed for an axes group that was in a <i>GroupDisable</i> state.</p> <ul style="list-style-type: none"> <li>• MC_MoveTimeAbsolute (Time-specified Absolute Positioning) instruction</li> <li>• MC_SyncLinearConveyor (Start Conveyor Synchronization) instruction</li> <li>• MC_SyncOut (End Synchronization) instruction</li> <li>• MC_RobotJog (Axes Group Jog) instruction</li> </ul>		Correct the program so that the instruction is executed only after the axes group state changed to <i>GroupEnable</i> . Execute the MC_GroupEnable (Enable Axes Group) instruction to change the axes group state to <i>GroupEnable</i> .		Execute multi-axes coordinated control instructions only after enabling the axes group. Execute the MC_GroupEnable (Enable Axes Group) instruction to change the axes group state to <i>GroupEnable</i> .	
<b>Attached information</b>	None					
<b>Precautions/Remarks</b>	None					

<b>Event name</b>	Impossible Axis Operation Specified when the Servo is OFF		<b>Event code</b>	54410000 hex	
<b>Meaning</b>	A motion instruction was executed for an axis for which the Servo is OFF.				
<b>Source</b>	Motion Control Function Module		<b>Source details</b>	Axis/axes group	<b>Detection timing</b> At instruction execution
<b>Error attributes</b>	<b>Level</b>	Minor fault	<b>Recovery</b>	Error reset	<b>Log category</b> System
<b>Effects</b>	<b>User program</b>	Continues.	<b>Operation</b>	The motion instruction will not start.	
<b>System defined variables</b>	<b>Variable</b>		<b>Data type</b>		<b>Name</b>
	_MC_AX[*].MFAultLvl.Active		BOOL		Axis Minor Fault Occurrence
	_MC_GRP[*].MFAultLvl.Active		BOOL		Axes Group Minor Fault Occurrence
<b>Cause and correction</b>	<b>Assumed cause</b>		<b>Correction</b>		<b>Prevention</b>
	A motion instruction was executed for an axis for which the Servo is OFF.		Correct the program so that the instruction is executed after the Servo is turned ON.		Execute an axis motion instruction only after the Servo is turned ON.
	A zero position preset was performed with the MC_Home (Home) instruction or MC_HomeWithParameter (Home with Parameters) instruction for an axis for which EtherCAT process data communications are not established.		If the <i>_EC_PDSlavTbl</i> (Process Data Communicating Slave Table) system-defined variable for the EtherCAT master of the master axis is FALSE, remove its cause and execute the MC_Home or MC_HomeWithParameter instruction to preset home after <i>_EC_PD-SlavTbl</i> changes to TRUE.		Write the program to make sure that the <i>_EC_PDSlavTbl</i> (Process Data Communicating Slave Table) system-defined variable for the EtherCAT master is TRUE if you execute the MC_Home or MC_HomeWithParameter instruction to preset home immediately after you turn ON the power supply to the Controller, download data, reset a slave communications error, disconnect the slave, reconnect the slave, enable the slave, or disable the slave.
<b>Attached information</b>	Attached Information 1: Depends on the source details. Axis: 0 Axes group: Number of the logical axis where the error occurred				
<b>Precautions/Remarks</b>	None				

## 6 Troubleshooting

<b>Event name</b>	Composition Axis Stopped Error		<b>Event code</b>	54420000 hex		
<b>Meaning</b>	A motion instruction was executed for an axes group while the MC_Stop instruction was being executed for a composition axis.					
<b>Source</b>	Motion Control Function Module		<b>Source details</b>	Axes group	<b>Detection timing</b>	At instruction execution
<b>Error attributes</b>	<b>Level</b>	Minor fault	<b>Recovery</b>	Error reset	<b>Log category</b>	System
<b>Effects</b>	<b>User program</b>	Continues.	<b>Operation</b>	Operation is not possible for relevant axes group.		
<b>System defined variables</b>	<b>Variable</b>		<b>Data type</b>		<b>Name</b>	
	_MC_GRP[*].MFAultLvl.Active		BOOL		Axes Group Minor Fault Occurrence	
<b>Cause and correction</b>	<b>Assumed cause</b>		<b>Correction</b>		<b>Prevention</b>	
	A motion instruction was executed for an axes group while the MC_Stop instruction was being executed for a composition axis.		Change the <i>Execute</i> input variable to the MC_Stop instruction for the composition axis to FALSE, reset the error, and then execute the motion control instruction.		Change the <i>Execute</i> input variables to the MC_Stop instructions for all of the composition axes to FALSE before you execute motion control instruction.	
<b>Attached information</b>	Attached information 1: Number of the logical axis that was stopped.					
<b>Precautions/Remarks</b>	None					

<b>Event name</b>	Motion Control Instruction Multi-execution Buffer Limit Exceeded		<b>Event code</b>	54430000 hex		
<b>Meaning</b>	The number of motion control instructions that is buffered for Buffered or Blending Buffer Modes exceeded the buffer limit.					
<b>Source</b>	Motion Control Function Module		<b>Source details</b>	Axis/axes group	<b>Detection timing</b>	Controller
<b>Error attributes</b>	<b>Level</b>	Minor fault	<b>Recovery</b>	Error reset	<b>Log category</b>	System
<b>Effects</b>	<b>User program</b>	Continues.	<b>Operation</b>	If "axis" is given for the source details, operation is not possible for relevant axis. Relevant axis decelerates to a stop if it is in motion. If "axes group" is given for the source details, operation is not possible for relevant axes group. Relevant axes group decelerates to a stop if it is in motion.		
<b>System defined variables</b>	<b>Variable</b>		<b>Data type</b>		<b>Name</b>	
	_MC_AX[*].MFAultLvl.Active		BOOL		Axis Minor Fault Occurrence	
	_MC_GRP[*].MFAultLvl.Active		BOOL		Axes Group Minor Fault Occurrence	
<b>Cause and correction</b>	<b>Assumed cause</b>		<b>Correction</b>		<b>Prevention</b>	
	An axis instruction was executed when there was already a current instruction and a buffered instruction for the same axis.		Correct the program so that the number of executed instructions does not exceed the buffer limit.		Do not execute an axis instruction when there is already a current instruction and a buffered instruction for the same axis.	
An axes group instruction was executed when there was already eight current instructions and buffered instructions for the same axis.		Do not execute an axes group instruction when there are already eight current and buffered instructions for the same axis.				
<b>Attached information</b>	None					
<b>Precautions/Remarks</b>	None					

<b>Event name</b>	Motion Control Instruction Re-execution Disabled (Axes Group Specification)		<b>Event code</b>	54570000 hex		
<b>Meaning</b>	An attempt was made to change the parameter for the <i>AxesGroup</i> input variable when re-executing a motion control instruction. (This input variable cannot be changed when re-executing an instruction.)					
<b>Source</b>	Motion Control Function Module		<b>Source details</b>	Axes group	<b>Detection timing</b>	At instruction re-execution
<b>Error attributes</b>	<b>Level</b>	Minor fault	<b>Recovery</b>	Error reset	<b>Log category</b>	System
<b>Effects</b>	<b>User program</b>	Continues.	<b>Operation</b>	Operation is not possible for relevant axes group. Relevant axes group decelerates to a stop if it is in motion.		
<b>System defined variables</b>	<b>Variable</b>		<b>Data type</b>		<b>Name</b>	
	_MC_GRP[*].MFAultLvl.Active		BOOL		Axes Group Minor Fault Occurrence	
<b>Cause and correction</b>	<b>Assumed cause</b>		<b>Correction</b>		<b>Prevention</b>	
	A parameter for an input variable that cannot be changed for re-execution was changed.		Correct the program so that the parameter for the relevant input variable does not change when the relevant instruction is re-executed.		Check the manual to see if the input variables to the relevant motion control instruction can be changed by re-execution. Write the program so that the input parameters for any input variable that cannot be changed do not change upon re-execution.	
<b>Attached information</b>	None					
<b>Precautions/Remarks</b>	None					

<b>Event name</b>	Illegal Axes Group Specification		<b>Event code</b>	54610000 hex		
<b>Meaning</b>	The axes group specified for the <i>AxesGroup</i> input variable to a motion control instruction does not exist or is not a used axes group.					
<b>Source</b>	Motion Control Function Module		<b>Source details</b>	MC common	<b>Detection timing</b>	At instruction execution
<b>Error attributes</b>	<b>Level</b>	Minor fault	<b>Recovery</b>	Error reset	<b>Log category</b>	System
<b>Effects</b>	<b>User program</b>	Continues.	<b>Operation</b>	The relevant instruction is not executed.		
<b>System-defined variables</b>	<b>Variable</b>		<b>Data type</b>		<b>Name</b>	
	_MC_COM.MFAultLvl.Active		BOOL		MC Common Minor Fault Occurrence	
<b>Cause and correction</b>	<b>Assumed cause</b>		<b>Correction</b>		<b>Prevention</b>	
	An axes group does not exist for the variable specified for the <i>AxesGroup</i> input variable to the instruction.		Correct the specification for the instruction so that the specified axes group exists.		Specify a variable that exists when you specify a variable for an input parameter to an instruction.	
	The axes group specified for the <i>AxesGroup</i> input variable to the instruction is not specified as a used axes group.		Set a used axes group for the axes group that is specified for the instruction.		Set a used axes group for the <i>AxesGroup</i> input variable to the instruction.	
<b>Attached information</b>	None					
<b>Precautions/Remarks</b>	None					

## 6 Troubleshooting

<b>Event name</b>	Instruction Execution Error with Undefined Home		<b>Event code</b>	54660000 hex		
<b>Meaning</b>	High-speed homing, an interpolation instruction, or a robot instruction was executed when home was undefined.					
<b>Source</b>	Motion Control Function Module		<b>Source details</b>	Axis/axes group	<b>Detection timing</b>	At instruction execution
<b>Error attributes</b>	<b>Level</b>	Minor fault	<b>Recovery</b>	Error reset	<b>Log category</b>	System
<b>Effects</b>	<b>User program</b>	Continues.	<b>Operation</b>	<p>If "axis" is given for the source details, operation is not possible for relevant axis. Relevant axis decelerates to a stop if it is in motion.</p> <p>If "axes group" is given for the source details, operation is not possible for relevant axes group.</p> <p>Relevant axes group decelerates to a stop if it is in motion.</p> <p>If the axes group is in motion for a robot instruction, it stops immediately.</p>		
<b>System-defined variables</b>	<b>Variable</b>		<b>Data type</b>		<b>Name</b>	
	_MC_AX[*].MFaultLvl.Active		BOOL		Axis Minor Fault Occurrence	
	_MC_GRP[*].MFaultLvl.Active		BOOL		Axes Group Minor Fault Occurrence	
<b>Cause and correction</b>	<b>Assumed cause</b>		<b>Correction</b>		<b>Prevention</b>	
	High-speed homing was executed when home was undefined.		Execute the high-speed homing operation only after home is defined by homing.		Execute the high-speed homing instruction only after home is defined by homing.	
	<p>An interpolation instruction was executed for an axes group that includes an axis with no defined home.</p> <p>One of the following robot instructions was executed for an axes group that includes a logical axis with no defined home.</p> <ul style="list-style-type: none"> <li>• MC_SetKinTransform (Set Kinematics Transformation) instruction</li> <li>• MC_MoveTimeAbsolute (Time-specified Absolute Positioning) instruction</li> <li>• MC_SyncLinearConveyor (Start Conveyor Synchronization) instruction</li> <li>• MC_SyncOut (End Synchronization) instruction</li> <li>• MC_GroupMon (Group Monitor) instruction</li> <li>• MC_RobotJog (Axes Group Jog) instruction</li> </ul>		Perform homing to define home for all axes in the axes group before you execute the interpolation instruction.		Perform homing to define home for all axes in the axes group before you execute an interpolation instruction.	
<b>Attached information</b>	<p>Attached Information 1: Depends on the source details.</p> <p style="padding-left: 40px;">Axis: 0</p> <p style="padding-left: 40px;">Axes group: Logical axis number</p>					
<b>Precautions/Remarks</b>	If you execute the Set Position instruction for axes which are in a ready state or for which homing is already performed, home will again be undefined. In this case, you must perform homing again to define home or make the axes ready for operation.					

<b>Event name</b>	Target Position Setting Out of Range		<b>Event code</b>	54780000 hex	
<b>Meaning</b>	The parameter specified for the <i>Position</i> input variable to a motion control instruction is out of range.				
<b>Source</b>	Motion Control Function Module		<b>Source details</b>	Axis/axes group	<b>Detection timing</b> At instruction execution
<b>Error attributes</b>	<b>Level</b>	Minor fault	<b>Recovery</b>	Error reset	<b>Log category</b> System
<b>Effects</b>	<b>User program</b>	Continues.	<b>Operation</b>	<p>If "axis" is given for the source details, operation is not possible for relevant axis. Relevant axis decelerates to a stop if it is in motion.</p> <p>If "axes group" is given for the source details, operation is not possible for relevant axes group. Relevant axes group decelerates to a stop if it is in motion.</p>	
<b>System-defined variables</b>	<b>Variable</b>		<b>Data type</b>		<b>Name</b>
	_MC_AX[*].MFAultLvl.Active		BOOL		Axis Minor Fault Occurrence
	_MC_GRP[*].MFAultLvl.Active		BOOL		Axes Group Minor Fault Occurrence
<b>Cause and correction</b>	<b>Assumed cause</b>		<b>Correction</b>		<b>Prevention</b>
	Instruction input parameter exceeded the valid range of the input variable.		Correct the parameter so that the valid range of the input variable is not exceeded for the relevant instruction.		Set the input parameter to the instruction so that the valid range of the input variable is not exceeded.
	The target position of a Rotary Mode axis is not within the ring setting range.		Correct the target position of the Rotary Mode axis to within the ring setting range.		Set the target position of the Rotary Mode axis to within the ring setting range.
<b>Attached information</b>	Depends on the source details. Axis: None Axes group: Element number that is out of range in the <i>Position</i> input variable to the instruction.				
<b>Precautions/Remarks</b>	None				

## 6 Troubleshooting

<b>Event name</b>	Kinematics Unsupported Instruction		<b>Event code</b>	57050000 hex	
<b>Meaning</b>	An attempt was made to execute an instruction that cannot be used for an axes group for which the kinematics transform was set.				
<b>Source</b>	Motion Control Function Module		<b>Source details</b>	Axes group	<b>Detection timing</b> At instruction execution
<b>Error attributes</b>	<b>Level</b>	Minor fault	<b>Recovery</b>	Error reset	<b>Log category</b> System
<b>Effects</b>	<b>User program</b>	Continues.	<b>Operation</b>	Operation is not possible for relevant axes group. Relevant axes group stops immediately if it is in motion.	
<b>System-defined variables</b>	<b>Variable</b>	<b>Data type</b>		<b>Name</b>	
	_MC_GRP[*].MFaultLvl.Active	BOOL		Axes Group Minor Fault Occurrence	
<b>Cause and correction</b>	<b>Assumed cause</b>		<b>Correction</b>		<b>Prevention</b>
	<p>One of the following instructions was executed for an axes group for which the kinematics transform was set.</p> <ul style="list-style-type: none"> <li>• MC_MoveLinear (Linear Interpolation) instruction</li> <li>• MC_MoveLinearAbsolute (Absolute Linear Interpolation) instruction</li> <li>• MC_MoveLinearRelative (Relative Linear Interpolation) instruction</li> <li>• MC_MoveCircular2D (Circular 2D Interpolation) instruction</li> <li>• MC_ChangeAxesInGroup (Change Axes in Group) instruction</li> </ul>		Check and correct the program so that the instruction that caused the error is not executed for a group for which the kinematics transform is set.		Do not execute these instructions for an axes group for which the kinematics transform is set.
<b>Attached information</b>	None				
<b>Precautions/Remarks</b>	None				

<b>Event name</b>	Axes Group Mismatch with Kinematics		<b>Event code</b>	57060000 hex	
<b>Meaning</b>	The configuration elements of the specified axes group and the specified kinematics do not match.				
<b>Source</b>	Motion Control Function Module		<b>Source details</b>	Axes group	<b>Detection timing</b> At instruction execution
<b>Error attributes</b>	<b>Level</b>	Minor fault	<b>Recovery</b>	Error reset	<b>Log category</b> System
<b>Effects</b>	<b>User program</b>	Continues.	<b>Operation</b>	Operation is not possible for relevant axes group.	
<b>System-defined variables</b>	<b>Variable</b>		<b>Data type</b>		<b>Name</b>
	_MC_GRP[*].MFaultLvl.Active		BOOL		Axes Group Minor Fault Occurrence
<b>Cause and correction</b>	<b>Assumed cause</b>		<b>Correction</b>		<b>Prevention</b>
	The number of axes in the axes group and the number of axes in the specified robot (kinematics type) do not match.		Correct the setting so that the number of axes in the axes group and the number of axes in the specified robot (kinematics type) match.		Make sure that the number of axes in the axes group and the number of axes in the specified robot (kinematics type) match.
	The count mode for axes in the axes group and the count mode for the specified robot (kinematics type) do not match.		Correct the count mode for axes in the axes group so that it can match the count mode for the robot (kinematics type).		Make sure that the count mode for axes in the axes group is the same as the count mode specified for the robot (kinematics type).
	The display unit for axes in the axes group and the display unit for the specified robot (kinematics type) do not match.		Correct the display unit for axes in the axes group so that it can match the display unit for the robot (kinematics type).		Make sure that the display unit for axes in the axes group is the same as the display unit specified for the robot (kinematics type).
<b>Attached information</b>	None				
<b>Precautions/Remarks</b>	None				

<b>Event name</b>	Kinematics Type Out of Range		<b>Event code</b>	57070000 hex	
<b>Meaning</b>	<i>KinType</i> specified for the <i>KinTransform</i> input variable to a motion control instruction is out of range.				
<b>Source</b>	Motion Control Function Module		<b>Source details</b>	Axes group	<b>Detection timing</b> At instruction execution
<b>Error attributes</b>	<b>Level</b>	Minor fault	<b>Recovery</b>	Error reset	<b>Log category</b> System
<b>Effects</b>	<b>User program</b>	Continues.	<b>Operation</b>	Operation is not possible for relevant axes group. Relevant axes group stops immediately if it is in motion.	
<b>System-defined variables</b>	<b>Variable</b>		<b>Data type</b>		<b>Name</b>
	_MC_GRP[*].MFaultLvl.Active		BOOL		Axes Group Minor Fault Occurrence
<b>Cause and correction</b>	<b>Assumed cause</b>		<b>Correction</b>		<b>Prevention</b>
	<i>KinType</i> is outside the setting range.		Correct <i>KinType</i> of the <i>KinTransform</i> input variable.		Set the kinematics type correctly for the <i>KinTransform</i> input variable.
<b>Attached information</b>	None				
<b>Precautions/Remarks</b>	None				

## 6 Troubleshooting

<b>Event name</b>	Kinematics Parameter Out of Range		<b>Event code</b>	57080000 hex		
<b>Meaning</b>	<i>KinParam</i> or <i>ExpansionParam</i> specified for the <i>KinTransform</i> input variable to a motion control instruction is out of range.					
<b>Source</b>	Motion Control Function Module		<b>Source details</b>	Axes group	<b>Detection timing</b>	At instruction execution
<b>Error attributes</b>	<b>Level</b>	Minor fault	<b>Recovery</b>	Error reset	<b>Log category</b>	System
<b>Effects</b>	<b>User program</b>	Continues.	<b>Operation</b>	Operation is not possible for relevant axes group.		
<b>System-defined variables</b>	<b>Variable</b>		<b>Data type</b>		<b>Name</b>	
	_MC_GRP[*].MFaultLvl.Active		BOOL		Axes Group Minor Fault Occurrence	
<b>Cause and correction</b>	<b>Assumed cause</b>		<b>Correction</b>		<b>Prevention</b>	
	<i>KinParam</i> is outside the setting range.		Correct <i>KinParam</i> or <i>ExpansionParam</i> of the <i>KinTransform</i> input variable.		Set the correct kinematics parameter and expansion parameter for the <i>KinTransform</i> input variable.	
<b>Attached information</b>	None					
<b>Precautions/Remarks</b>	None					

<b>Event name</b>	Workspace Type Out of Range		<b>Event code</b>	57090000 hex		
<b>Meaning</b>	<i>WorkspaceType</i> specified for the <i>Workspace</i> input variable to a motion control instruction is out of range.					
<b>Source</b>	Motion Control Function Module		<b>Source details</b>	Axes group	<b>Detection timing</b>	At instruction execution
<b>Error attributes</b>	<b>Level</b>	Minor fault	<b>Recovery</b>	Error reset	<b>Log category</b>	System
<b>Effects</b>	<b>User program</b>	Continues.	<b>Operation</b>	Operation is not possible for relevant axes group.		
<b>System-defined variables</b>	<b>Variable</b>		<b>Data type</b>		<b>Name</b>	
	_MC_GRP[*].MFaultLvl.Active		BOOL		Axes Group Minor Fault Occurrence	
<b>Cause and correction</b>	<b>Assumed cause</b>		<b>Correction</b>		<b>Prevention</b>	
	<i>WorkspaceType</i> is outside the setting range.		Correct <i>WorkspaceType</i> of the <i>Workspace</i> input variable.		Set the correct workspace type for the <i>Workspace</i> input variable.	
<b>Attached information</b>	None					
<b>Precautions/Remarks</b>	None					

<b>Event name</b>	Workspace Parameter Out of Range		<b>Event code</b>	570A0000 hex	
<b>Meaning</b>	<i>WorkspaceParam</i> specified for the <i>Workspace</i> input variable to a motion control instruction is out of range.				
<b>Source</b>	Motion Control Function Module		<b>Source details</b>	Axes group	<b>Detection timing</b>
<b>Error attributes</b>	<b>Level</b>	Minor fault	<b>Recovery</b>	Error reset	<b>Log category</b>
<b>Effects</b>	<b>User program</b>	Continues.	<b>Operation</b>	Operation is not possible for relevant axes group.	
<b>System-defined variables</b>	<b>Variable</b>		<b>Data type</b>		<b>Name</b>
	_MC_GRP[*].MFAultLvl.Active		BOOL		Axes Group Minor Fault Occurrence
<b>Cause and correction</b>	<b>Assumed cause</b>		<b>Correction</b>		<b>Prevention</b>
	<i>WorkspaceParam</i> is outside the setting range.		Correct <i>WorkspaceParam</i> of the <i>Workspace</i> input variable.		Set the correct workspace parameter for the <i>Workspace</i> input variable.
<b>Attached information</b>	None				
<b>Precautions/Remarks</b>	None				

<b>Event name</b>	Invalid Coordinate System Number		<b>Event code</b>	570B0000 hex	
<b>Meaning</b>	The coordinate system ID specified for the <i>CSID</i> input variable to a motion control instruction is out of range or not defined.				
<b>Source</b>	Motion Control Function Module		<b>Source details</b>	Axes group	<b>Detection timing</b>
<b>Error attributes</b>	<b>Level</b>	Minor fault	<b>Recovery</b>	Error reset	<b>Log category</b>
<b>Effects</b>	<b>User program</b>	Continues.	<b>Operation</b>	Operation is not possible for relevant axes group. Relevant axes group stops immediately if it is in motion.	
<b>System-defined variables</b>	<b>Variable</b>		<b>Data type</b>		<b>Name</b>
	_MC_GRP[*].MFAultLvl.Active		BOOL		Axes Group Minor Fault Occurrence
<b>Cause and correction</b>	<b>Assumed cause</b>		<b>Correction</b>		<b>Prevention</b>
	The coordinate system ID is outside the setting range.		Change the value of the <i>CSID</i> input variable to a correct coordinate system ID.		Set the coordinate system ID correctly for the <i>CSID</i> input variable.
	The specified coordinate system ID is not defined by the MC_DefineCoordSystem (Define Coordinate) instruction.		Define the coordinate system with the MC_DefineCoordSystem (Define Coordinate) instruction.		Define the coordinate system and use it.
<b>Attached information</b>	None				
<b>Precautions/Remarks</b>	None				

## 6 Troubleshooting

<b>Event name</b>	Coordinate Transformation Parameter Out of Range		<b>Event code</b>	570C0000 hex		
<b>Meaning</b>	The coordinate transformation parameter <i>Pose</i> , which was specified for the <i>CoordTransform</i> input variable to a motion control instruction, is out of range.					
<b>Source</b>	Motion Control Function Module		<b>Source details</b>	MC common	<b>Detection timing</b>	At instruction execution
<b>Error attributes</b>	<b>Level</b>	Minor fault	<b>Recovery</b>	---	<b>Log category</b>	System
<b>Effects</b>	<b>User program</b>	Continues.	<b>Operation</b>	Not affected.		
<b>System-defined variables</b>	<b>Variable</b>		<b>Data type</b>		<b>Name</b>	
	_MC_COM.MFaultLvl.Active		BOOL		MC Common Minor Fault Occurrence	
<b>Cause and correction</b>	<b>Assumed cause</b>		<b>Correction</b>		<b>Prevention</b>	
	The coordinate transformation parameter <i>Pose</i> is outside the setting range.		Correct the coordinate transformation parameter <i>Pose</i> of the <i>CoordTransform</i> input variable.		Set the coordinate transformation parameter <i>Pose</i> correctly for the <i>CoordTransform</i> input variable.	
<b>Attached information</b>	None					
<b>Precautions/Remarks</b>	None					

<b>Event name</b>	Transition parameters out of range		<b>Event code</b>	570D0000 hex		
<b>Meaning</b>	The Transition Parameters specified for the <i>TransitionParameter</i> input variable to a motion control instruction is out of range.					
<b>Source</b>	Motion Control Function Module		<b>Source details</b>	Axes group	<b>Detection timing</b>	At instruction execution
<b>Error attributes</b>	<b>Level</b>	Minor fault	<b>Recovery</b>	Error reset	<b>Log category</b>	System
<b>Effects</b>	<b>User program</b>	Continues.	<b>Operation</b>	Operation is not possible for relevant axes group. Relevant axes group stops immediately if it is in motion.		
<b>System-defined variables</b>	<b>Variable</b>		<b>Data type</b>		<b>Name</b>	
	_MC_GRP[*].MFaultLvl.Active		BOOL		Axes Group Minor Fault Occurrence	
<b>Cause and correction</b>	<b>Assumed cause</b>		<b>Correction</b>		<b>Prevention</b>	
	<i>TransitionParameter</i> is outside the setting range.		Correct the parameters of the <i>TransitionParameter</i> input variable.		Set the correct parameters for the <i>TransitionParameter</i> input variable.	
<b>Attached information</b>	None					
<b>Precautions/Remarks</b>	None					

<b>Event name</b>	Cannot Calculate Kinematics		<b>Event code</b>	570F0000 hex		
<b>Meaning</b>	The inverse kinematics or direct kinematics cannot be calculated.					
<b>Source</b>	Motion Control Function Module		<b>Source details</b>	Axes group	<b>Detection timing</b>	At or during instruction execution
<b>Error attributes</b>	<b>Level</b>	Minor fault	<b>Recovery</b>	Error reset	<b>Log category</b>	System
<b>Effects</b>	<b>User program</b>	Continues.	<b>Operation</b>	Operation is not possible for relevant axes group. Relevant axes group stops immediately if it is in motion.		
<b>System-defined variables</b>	<b>Variable</b>	<b>Data type</b>		<b>Name</b>		
	_MC_GRP[*].MFAultLvl.Active	BOOL		Axes Group Minor Fault Occurrence		
<b>Cause and correction</b>	<b>Assumed cause</b>		<b>Correction</b>		<b>Prevention</b>	
	The kinematics parameter is outside the setting range.		Correct the kinematics parameter.		Set the kinematics parameter correctly.	
	One of the robot axes (A0 to A2) is stopped at the position for which direct kinematics calculation cannot be performed.		Turn OFF the Servo for the robot axes. Ensure safety and move the axis inside the robot moving range.		Review the program and make sure that the robot moves within its moving range.	
	Inverse kinematics calculation cannot be performed for the target position specified in the machine coordinate system.		Correct the target position so that it is inside the robot moving range.		Set the target position inside the robot moving range.	
<b>Attached information</b>	None					
<b>Precautions/Remarks</b>	None					

<b>Event name</b>	Kinematics Transform Not Set		<b>Event code</b>	57100000 hex		
<b>Meaning</b>	The kinematics transform is not set for the specified axes group.					
<b>Source</b>	Motion Control Function Module		<b>Source details</b>	Axes group	<b>Detection timing</b>	At instruction execution
<b>Error attributes</b>	<b>Level</b>	Minor fault	<b>Recovery</b>	Error reset	<b>Log category</b>	System
<b>Effects</b>	<b>User program</b>	Continues.	<b>Operation</b>	Operation is not possible for relevant axes group. Relevant axes group stops immediately if it is in motion.		
<b>System-defined variables</b>	<b>Variable</b>	<b>Data type</b>		<b>Name</b>		
	_MC_GRP[*].MFAultLvl.Active	BOOL		Axes Group Minor Fault Occurrence		
<b>Cause and correction</b>	<b>Assumed cause</b>		<b>Correction</b>		<b>Prevention</b>	
	The kinematics transform is not set for the axes group.		Use the MC_SetKinTransform (Set Kinematics Transformation) instruction to set the kinematics transform for the axes group.		Execute the instruction after you set the kinematics transform for the axes group.	
<b>Attached information</b>	None					
<b>Precautions/Remarks</b>	None					

## 6 Troubleshooting

<b>Event name</b>	Target Position Out of Range		<b>Event code</b>	57110000 hex		
<b>Meaning</b>	The position parameter specified as variable for the <i>Position</i> is out of range					
<b>Source</b>	Motion Control Function Module		<b>Source details</b>	MC common	<b>Detection timing</b>	At instruction execution
<b>Error attributes</b>	<b>Level</b>	Minor fault	<b>Recovery</b>	---	<b>Log category</b>	System
<b>Effects</b>	<b>User program</b>	Continues.	<b>Operation</b>	Not affected.		
<b>System-defined variables</b>	<b>Variable</b>		<b>Data type</b>		<b>Name</b>	
	_MC_COM.MFaultLvl.Active		BOOL		MC Common Minor Fault Occurrence	
<b>Cause and correction</b>	<b>Assumed cause</b>		<b>Correction</b>		<b>Prevention</b>	
	<i>Positoin</i> is outside the setting range.		Correct the target position inside maximum movement volume of the robot.		Set the target position inside maximum movement volume of the robot.	
<b>Attached information</b>	None					
<b>Precautions/Remarks</b>	None					

<b>Event name</b>	Velocity Error Detection Value Out of Range		<b>Event code</b>	57120000 hex		
<b>Meaning</b>	<i>MaxVelocity</i> specified for the <i>TrajData</i> input variable to a motion control instruction is out of range.					
<b>Source</b>	Motion Control Function Module		<b>Source details</b>	Axes group	<b>Detection timing</b>	At instruction execution
<b>Error attributes</b>	<b>Level</b>	Minor fault	<b>Recovery</b>	Error reset	<b>Log category</b>	System
<b>Effects</b>	<b>User program</b>	Continues.	<b>Operation</b>	Operation is not possible for relevant axes group. Relevant axes group stops immediately if it is in motion.		
<b>System-defined variables</b>	<b>Variable</b>		<b>Data type</b>		<b>Name</b>	
	_MC_GRP[*].MFaultLvl.Active		BOOL		Axes Group Minor Fault Occurrence	
<b>Cause and correction</b>	<b>Assumed cause</b>		<b>Correction</b>		<b>Prevention</b>	
	<i>MaxVelocity</i> is outside the setting range.		Correct the value of <i>MaxVelocity</i> of the <i>TrajData</i> input variable.		Set <i>MaxVelocity</i> for the <i>TrajData</i> input variable correctly.	
<b>Attached information</b>	None					
<b>Precautions/Remarks</b>	None					

<b>Event name</b>	Acceleration Error Detection Value Out of Range		<b>Event code</b>	57130000 hex	
<b>Meaning</b>	<i>MaxAcceleration</i> specified for the <i>TrajData</i> input variable to a motion control instruction is out of range.				
<b>Source</b>	Motion Control Function Module		<b>Source details</b>	Axes group	<b>Detection timing</b> At instruction execution
<b>Error attributes</b>	<b>Level</b>	Minor fault	<b>Recovery</b>	Error reset	<b>Log category</b> System
<b>Effects</b>	<b>User program</b>	Continues.	<b>Operation</b>	Operation is not possible for relevant axes group. Relevant axes group stops immediately if it is in motion.	
<b>System-defined variables</b>	<b>Variable</b>		<b>Data type</b>		<b>Name</b>
	_MC_GRP[*].MFAultLvl.Active		BOOL		Axes Group Minor Fault Occurrence
<b>Cause and correction</b>	<b>Assumed cause</b>		<b>Correction</b>		<b>Prevention</b>
	<i>MaxAcceleration</i> is outside the setting range.		Correct the value of <i>MaxAcceleration</i> of the <i>TrajData</i> input variable.		Set <i>MaxAcceleration</i> correctly for the <i>TrajData</i> input variable.
<b>Attached information</b>	None				
<b>Precautions/Remarks</b>	None				

<b>Event name</b>	Trajectory Target Time Out of Range		<b>Event code</b>	57140000 hex	
<b>Meaning</b>	<i>TrajTime</i> specified for the <i>TrajData</i> input variable to a motion control instruction is out of range.				
<b>Source</b>	Motion Control Function Module		<b>Source details</b>	Axes group	<b>Detection timing</b> At instruction execution
<b>Error attributes</b>	<b>Level</b>	Minor fault	<b>Recovery</b>	Error reset	<b>Log category</b> System
<b>Effects</b>	<b>User program</b>	Continues.	<b>Operation</b>	Operation is not possible for relevant axes group. Relevant axes group stops immediately if it is in motion.	
<b>System-defined variables</b>	<b>Variable</b>		<b>Data type</b>		<b>Name</b>
	_MC_GRP[*].MFAultLvl.Active		BOOL		Axes Group Minor Fault Occurrence
<b>Cause and correction</b>	<b>Assumed cause</b>		<b>Correction</b>		<b>Prevention</b>
	<i>TrajTime</i> is outside the setting range.		Correct the value of <i>TrajTime</i> of the <i>TrajData</i> input variable.		Set <i>TrajTime</i> correctly for the <i>TrajData</i> input variable.
<b>Attached information</b>	None				
<b>Precautions/Remarks</b>	None				

## 6 Troubleshooting

<b>Event name</b>	Trajectory Type Out of Range		<b>Event code</b>	57150000 hex		
<b>Meaning</b>	<i>MoveTrajType</i> or <i>SyncTrajType</i> specified for the <i>TrajData</i> input variable to a motion control instruction is out of range.					
<b>Source</b>	Motion Control Function Module		<b>Source details</b>	Axes group	<b>Detection timing</b>	At instruction execution
<b>Error attributes</b>	<b>Level</b>	Minor fault	<b>Recovery</b>	Error reset	<b>Log category</b>	System
<b>Effects</b>	<b>User program</b>	Continues.	<b>Operation</b>	Operation is not possible for relevant axes group. Relevant axes group stops immediately if it is in motion.		
<b>System-defined variables</b>	<b>Variable</b>	<b>Data type</b>		<b>Name</b>		
	_MC_GRP[*].MFAultLvl.Active	BOOL		Axes Group Minor Fault Occurrence		
<b>Cause and correction</b>	<b>Assumed cause</b>		<b>Correction</b>		<b>Prevention</b>	
	<i>MoveTrajType</i> or <i>SyncTrajType</i> is outside the setting range.		Correct the value of <i>MoveTrajType</i> or <i>SyncTrajType</i> of the <i>TrajData</i> input variable.		Set <i>MoveTrajType</i> or <i>SyncTrajType</i> correctly for the <i>TrajData</i> input variable.	
<b>Attached information</b>	None					
<b>Precautions/Remarks</b>	None					

<b>Event name</b>	Trajectory Transition Out of Range		<b>Event code</b>	57160000 hex		
<b>Meaning</b>	<i>TrajTransition</i> specified for the <i>TrajData</i> input variable to a motion control instruction is out of range.					
<b>Source</b>	Motion Control Function Module		<b>Source details</b>	Axes group	<b>Detection timing</b>	At instruction execution
<b>Error attributes</b>	<b>Level</b>	Minor fault	<b>Recovery</b>	Error reset	<b>Log category</b>	System
<b>Effects</b>	<b>User program</b>	Continues.	<b>Operation</b>	Operation is not possible for relevant axes group. Relevant axes group stops immediately if it is in motion.		
<b>System-defined variables</b>	<b>Variable</b>	<b>Data type</b>		<b>Name</b>		
	_MC_GRP[*].MFAultLvl.Active	BOOL		Axes Group Minor Fault Occurrence		
<b>Cause and correction</b>	<b>Assumed cause</b>		<b>Correction</b>		<b>Prevention</b>	
	<i>TrajTransition</i> is outside the setting range.		Correct the value of <i>TrajTransition</i> of the <i>TrajData</i> input variable.		Set the trajectory transition correctly.	
<b>Attached information</b>	None					
<b>Precautions/Remarks</b>	None					

<b>Event name</b>	Trajectory Travel Distance Out of Range		<b>Event code</b>	57170000 hex	
<b>Meaning</b>	<i>TrajDistance</i> specified for the <i>TrajData</i> input variable to a motion control instruction is out of range.				
<b>Source</b>	Motion Control Function Module		<b>Source details</b>	Axes group	<b>Detection timing</b> At instruction execution
<b>Error attributes</b>	<b>Level</b>	Minor fault	<b>Recovery</b>	Error reset	<b>Log category</b> System
<b>Effects</b>	<b>User program</b>	Continues.	<b>Operation</b>	Operation is not possible for relevant axes group. Relevant axes group stops immediately if it is in motion.	
<b>System-defined variables</b>	<b>Variable</b>		<b>Data type</b>		<b>Name</b>
	_MC_GRP[*].MFAultLvl.Active		BOOL		Axes Group Minor Fault Occurrence
<b>Cause and correction</b>	<b>Assumed cause</b>		<b>Correction</b>		<b>Prevention</b>
	<i>TrajDistance</i> is outside the setting range.		Correct the value of <i>TrajDistance</i> of the <i>TrajData</i> input variable.		Set the trajectory travel distance correctly.
<b>Attached information</b>	None				
<b>Precautions/Remarks</b>	None				

<b>Event name</b>	Initial Workpiece Position Outside Workspace		<b>Event code</b>	57190000 hex	
<b>Meaning</b>	The position of the workpiece specified for the <i>InitWorkpiecePosition</i> input variable to a motion control instruction is outside the workspace.				
<b>Source</b>	Motion Control Function Module		<b>Source details</b>	Axes group	<b>Detection timing</b> At instruction execution
<b>Error attributes</b>	<b>Level</b>	Minor fault	<b>Recovery</b>	Error reset	<b>Log category</b> System
<b>Effects</b>	<b>User program</b>	Continues.	<b>Operation</b>	Operation is not possible for relevant axes group. Relevant axes group stops immediately if it is in motion.	
<b>System-defined variables</b>	<b>Variable</b>		<b>Data type</b>		<b>Name</b>
	_MC_GRP[*].MFAultLvl.Active		BOOL		Axes Group Minor Fault Occurrence
<b>Cause and correction</b>	<b>Assumed cause</b>		<b>Correction</b>		<b>Prevention</b>
	<i>InitWorkpiecePosition</i> is outside the workspace.		Move the workpiece inside the workspace.		Do not allow the initial position of the workpiece to exceed the range of the workspace.
<b>Attached information</b>	None				
<b>Precautions/Remarks</b>	None				

## 6 Troubleshooting

<b>Event name</b>	Invalid Conveyor Axis Specified		<b>Event code</b>	571A0000 hex		
<b>Meaning</b>	The axis specified for the <i>ConveyorAxis</i> in-out variable to a motion control instruction is not correct.					
<b>Source</b>	Motion Control Function Module		<b>Source details</b>	Axes group	<b>Detection timing</b>	At instruction execution
<b>Error attributes</b>	<b>Level</b>	Minor fault	<b>Recovery</b>	Error reset	<b>Log category</b>	System
<b>Effects</b>	<b>User program</b>	Continues.	<b>Operation</b>	Operation is not possible for relevant axes group. Relevant axes group stops immediately if it is in motion.		
<b>System-defined variables</b>	<b>Variable</b>	<b>Data type</b>		<b>Name</b>		
	_MC_GRP[*].MFAultLvl.Active	BOOL		Axes Group Minor Fault Occurrence		
<b>Cause and correction</b>	<b>Assumed cause</b>		<b>Correction</b>		<b>Prevention</b>	
	The specified axis is registered in the axes group which is specified for <i>AxesGroup</i> .		Correct the axis setting.		Set the conveyor axis correctly.	
	The specified axis is an unused axis.		Change the axis setting to Used Axis.		Set Used Axis for the axis.	
	The unit of the conveyor axis is incorrect.		Change the unit of axis to mm.		Set the unit of axis to mm.	
<b>Attached information</b>	None					
<b>Precautions/Remarks</b>	None					

<b>Event name</b>	Target Position Outside Workspace		<b>Event code</b>	571B0000 hex		
<b>Meaning</b>	The target position specified for the <i>Position</i> input variable to a motion control instruction is outside the workspace.					
<b>Source</b>	Motion Control Function Module		<b>Source details</b>	Axes group	<b>Detection timing</b>	At instruction execution
<b>Error attributes</b>	<b>Level</b>	Minor fault	<b>Recovery</b>	Error reset	<b>Log category</b>	System
<b>Effects</b>	<b>User program</b>	Continues.	<b>Operation</b>	Operation is not possible for relevant axes group. Relevant axes group stops immediately if it is in motion.		
<b>System-defined variables</b>	<b>Variable</b>	<b>Data type</b>		<b>Name</b>		
	_MC_GRP[*].MFAultLvl.Active	BOOL		Axes Group Minor Fault Occurrence		
<b>Cause and correction</b>	<b>Assumed cause</b>		<b>Correction</b>		<b>Prevention</b>	
	<i>Position</i> is outside the workspace.		Correct the value of the <i>Position</i> input variable.		Do not allow the target position to exceed the range of the workspace.	
<b>Attached information</b>	None					
<b>Precautions/Remarks</b>	None					

<b>Event name</b>	Cannot Cancel Synchronization		<b>Event code</b>	571C0000 hex	
<b>Meaning</b>	The MC_SyncOut (End Synchronization) instruction cannot be executed.				
<b>Source</b>	Motion Control Function Module		<b>Source details</b>	Axes group	<b>Detection timing</b> At instruction execution
<b>Error attributes</b>	<b>Level</b>	Minor fault	<b>Recovery</b>	Error reset	<b>Log category</b> System
<b>Effects</b>	<b>User program</b>	Continues.	<b>Operation</b>	Operation is not possible for relevant axes group. Relevant axes group stops immediately if it is in motion.	
<b>System-defined variables</b>	<b>Variable</b>		<b>Data type</b>		<b>Name</b>
	_MC_GRP[*].MFAultLvl.Active		BOOL		Axes Group Minor Fault Occurrence
<b>Cause and correction</b>	<b>Assumed cause</b>		<b>Correction</b>		<b>Prevention</b>
	The MC_SyncLinearConveyor (Start Conveyor Synchronization) instruction execution is in progress, but <i>Phase</i> of the <i>TrajStatus</i> output variable is not 6.		Execute the MC_SyncOut (End Synchronization) instruction when <i>Phase</i> of the <i>TrajStatus</i> output variable is 6.		Execute the MC_SyncOut (End Synchronization) instruction only when <i>Phase</i> of the <i>TrajStatus</i> output variable from the MC_SyncLinearConveyor (Start Conveyor Synchronization) instruction is 6.
<b>Attached information</b>	None				
<b>Precautions/Remarks</b>	None				

<b>Event name</b>	Too Many Kinematics		<b>Event code</b>	571E0000 hex	
<b>Meaning</b>	The number of kinematics exceeded the limit.				
<b>Source</b>	Motion Control Function Module		<b>Source details</b>	Axes group	<b>Detection timing</b> At instruction execution
<b>Error attributes</b>	<b>Level</b>	Minor fault	<b>Recovery</b>	Error reset	<b>Log category</b> System
<b>Effects</b>	<b>User program</b>	Continues.	<b>Operation</b>	Operation is not possible for relevant axes group. Relevant axes group stops immediately if it is in motion.	
<b>System-defined variables</b>	<b>Variable</b>		<b>Data type</b>		<b>Name</b>
	_MC_GRP[*].MFAultLvl.Active		BOOL		Axes Group Minor Fault Occurrence
<b>Cause and correction</b>	<b>Assumed cause</b>		<b>Correction</b>		<b>Prevention</b>
	The number of the kinematics set by the MC_SetKinTransform (Set Kinematics Transformation) instruction exceeded the limit.		Review the ladder diagrams and make sure that the limit is not exceeded.		Do not allow the number of the kinematics to exceed the following limit: 8 for NJ501-4400/-4500/-4300/-4320 Units, 1 for an NJ501-4310 Unit.
<b>Attached information</b>	None				
<b>Precautions/Remarks</b>	None				

## 6 Troubleshooting

<b>Event name</b>	Kinematics Initialization Error		<b>Event code</b>	571F0000 hex		
<b>Meaning</b>	Kinematics initialization failed.					
<b>Source</b>	Motion Control Function Module		<b>Source details</b>	Axes group	<b>Detection timing</b>	At instruction execution
<b>Error attributes</b>	<b>Level</b>	Minor fault	<b>Recovery</b>	Error reset	<b>Log category</b>	System
<b>Effects</b>	<b>User program</b>	Continues.	<b>Operation</b>	Operation is not possible for relevant axes group. Relevant axes group stops immediately if it is in motion.		
<b>System-defined variables</b>	<b>Variable</b>	<b>Data type</b>		<b>Name</b>		
	_MC_GRP[*].MFaultLvl.Active	BOOL		Axes Group Minor Fault Occurrence		
<b>Cause and correction</b>	<b>Assumed cause</b>	<b>Correction</b>		<b>Prevention</b>		
	One of the robot axes (A0 to A3) is stopped at the position for which direct kinematics calculation cannot be performed.	<p>Turn OFF the Servo for the robot axes. Ensure safety and move the axis inside the robot moving range.</p> <p>If this error is detected, the settings of the kinematics, workspace, user coordinate systems and tool coordinate system for the relevant axes group are cleared.</p> <p>To operate the robot, you need to make these settings again.</p>		Review the program and make sure that the robot moves within its moving range.		
	Kinematics parameters set are incorrect.	<p>Review and set kinematics parameters correctly according the existing robot.</p> <p>If this error is detected, the settings of the kinematics, workspace, user coordinate systems and tool coordinate system for the relevant axes group are cleared.</p> <p>To operate the robot, you need to make these settings again.</p>		Review kinematics parameters of your robot.		
<b>Attached information</b>	None					
<b>Precautions/Remarks</b>	None					

<b>Event name</b>	Invalid Tool Number		<b>Event code</b>	57320000 hex	
<b>Meaning</b>	The tool ID specified for the <i>ToolID</i> input variable to a motion control instruction is out of range or not defined.				
<b>Source</b>	Motion Control Function Module		<b>Source details</b>	Axes group	<b>Detection timing</b> At instruction execution
<b>Error attributes</b>	<b>Level</b>	Minor fault	<b>Recovery</b>	Error reset	<b>Log category</b> System
<b>Effects</b>	<b>User program</b>	Continues.	<b>Operation</b>	Operation is not possible for relevant axes group. Relevant axes group stops immediately if it is in motion.	
<b>System-defined variables</b>	<b>Variable</b>		<b>Data type</b>		<b>Name</b>
	_MC_GRP[*].MFAultLvl.Active		BOOL		Axes Group Minor Fault Occurrence
<b>Cause and correction</b>	<b>Assumed cause</b>		<b>Correction</b>		<b>Prevention</b>
	<i>ToolID</i> is outside the setting range.		Change the value of the <i>ToolID</i> input variable to a correct tool ID.		Set the tool ID correctly for the <i>ToolID</i> input variable.
	The specified tool ID is not defined by the MC_DefineToolTransform (Define Tool Coordinate) instruction.		Define the tool with the MC_DefineToolTransform (Define Tool Coordinate) instruction.		Define the tool and use it.
<b>Attached information</b>	None				
<b>Precautions/Remarks</b>	None				

<b>Event name</b>	Tool Parameter Out of Range		<b>Event code</b>	57330000 hex	
<b>Meaning</b>	The transform parameter specified for the <i>ToolTransform</i> input variable to a motion control instruction is out of range.				
<b>Source</b>	Motion Control Function Module		<b>Source details</b>	Axes group	<b>Detection timing</b> At instruction execution
<b>Error attributes</b>	<b>Level</b>	Minor fault	<b>Recovery</b>	Error reset	<b>Log category</b> System
<b>Effects</b>	<b>User program</b>	Continues.	<b>Operation</b>	Operation is not possible for relevant axes group. Relevant axes group stops immediately if it is in motion.	
<b>System-defined variables</b>	<b>Variable</b>		<b>Data type</b>		<b>Name</b>
	MC_GRP[*].MFAultLvl.Active		BOOL		Axes Group Minor Fault Occurrence
<b>Cause and correction</b>	<b>Assumed cause</b>		<b>Correction</b>		<b>Prevention</b>
	The transform parameter is outside the setting range.		Change the value of the <i>ToolTransform</i> input variable to a correct transform parameter.		Set the transform parameter correctly for the <i>ToolTransform</i> input variable.
<b>Attached information</b>	None				
<b>Precautions/Remarks</b>	None				

## 6 Troubleshooting

<b>Event name</b>	Unsupported Transition Data		<b>Event code</b>	57340000 hex		
<b>Meaning</b>	The transition mode specified for the <i>TransitionMode</i> input variable to a motion control instruction does not support the current instruction or the buffered instruction.					
<b>Source</b>	Motion Control Function Module		<b>Source details</b>	Axes group	<b>Detection timing</b>	At instruction execution
<b>Error attributes</b>	<b>Level</b>	Minor fault	<b>Recovery</b>	Error reset	<b>Log category</b>	System
<b>Effects</b>	<b>User program</b>	Continues.	<b>Operation</b>	Operation is not possible for relevant axes group. Relevant axes group stops immediately if it is in motion.		
<b>System-defined variables</b>	<b>Variable</b>	<b>Data type</b>		<b>Name</b>		
	_MC_GRP[*].MFAultLvl.Active	BOOL		Axes Group Minor Fault Occurrence		
<b>Cause and correction</b>	<b>Assumed cause</b>	<b>Correction</b>		<b>Prevention</b>		
	The parameters specified for the <i>TransitionParameter</i> input variable cannot be used for transition in the mode specified for the <i>TransitionMode</i> input variable.	Correct the parameter of the <i>TransitionParameter</i> input variable so that it can be used for transition.		Set the <i>TransitionParameter</i> input variable to the value of parameter that can be used for transition.		
	The trajectory data which is set for the buffered instruction cannot be used for transition in the mode specified for the <i>TransitionMode</i> input variable.	Correct the trajectory data of the buffered instruction so that it can be used as the parameter for transition.		Set the trajectory data of the buffered instruction to the value that can be used as the parameter for transition.		
<b>Attached information</b>	None					
<b>Precautions/Remarks</b>	None					

<b>Event name</b>	Offset Not Allowed		<b>Event code</b>	57360000 hex		
<b>Meaning</b>	During the MC_SyncLinearConveyor (Start Conveyor Synchronization) instruction execution, the offset function can be used only in Phase6.					
<b>Source</b>	Motion Control Function Module		<b>Source details</b>	Axes group	<b>Detection timing</b>	At or during instruction execution
<b>Error attributes</b>	<b>Level</b>	Minor fault	<b>Recovery</b>	Error reset	<b>Log category</b>	System
<b>Effects</b>	<b>User program</b>	Continues.	<b>Operation</b>	Operation is not possible for relevant axes group. Relevant axes group stops immediately if it is in motion.		
<b>System-defined variables</b>	<b>Variable</b>	<b>Data type</b>		<b>Name</b>		
	_MC_GRP[*].MFAultLvl.Active	BOOL		Axes Group Minor Fault Occurrence		
<b>Cause and correction</b>	<b>Assumed cause</b>	<b>Correction</b>		<b>Prevention</b>		
	The <i>EnableOffset</i> input variable changed to TRUE before the MC_SyncLinearConveyor (Start Conveyor Synchronization) instruction operation entered Phase6.	Change the <i>EnableOffset</i> input variable to TRUE only when <i>Phase</i> of the <i>TrajStatus</i> output variable from the MC_SyncLinearConveyor (Start Conveyor Synchronization) instruction is 6.		Change the <i>EnableOffset</i> input variable to TRUE only when <i>Phase</i> of the <i>TrajStatus</i> output variable from the MC_SyncLinearConveyor (Start Conveyor Synchronization) instruction is 6.		
<b>Attached information</b>	None					
<b>Precautions/Remarks</b>	None					

<b>Event name</b>	Motion Control Instruction Multi-execution Disabled (Trajectory Type)		<b>Event code</b>	57370000 hex	
<b>Meaning</b>	An input variable that cannot be changed was changed during multi-execution of instructions.				
<b>Source</b>	Motion Control Function Module		<b>Source details</b>	Axes group	<b>Detection timing</b>
<b>Error attributes</b>	<b>Level</b>	Minor fault	<b>Recovery</b>	Error reset	<b>Log category</b>
<b>Effects</b>	<b>User program</b>	Continues.	<b>Operation</b>	Operation is not possible for relevant axes group. Relevant axes group stops immediately if it is in motion.	
<b>System-defined variables</b>	<b>Variable</b>	<b>Data type</b>		<b>Name</b>	
	_MC_GRP[*].MFAultLvl.Active	BOOL		Axes Group Minor Fault Occurrence	
<b>Cause and correction</b>	<b>Assumed cause</b>		<b>Correction</b>		<b>Prevention</b>
	MoveTrajType or SyncTrajType for the current instruction do not match MoveTrajType or SyncTrajType for the next instruction executed with Buffered or Blending.		You cannot change these input variables when you execute the next instruction for multi-execution of instructions. Correct the program.		Set MoveTrajType and SyncTrajType that are the same as that for the current instruction.
<b>Attached information</b>	None				
<b>Precautions/Remarks</b>	None				

<b>Event name</b>	Unsupported Transition Mode		<b>Event code</b>	57390000 hex	
<b>Meaning</b>	The next instruction was executed with <i>TransitionMode</i> which does not support the combination of the current and next instructions.				
<b>Source</b>	Motion Control Function Module		<b>Source details</b>	Axes group	<b>Detection timing</b>
<b>Error attributes</b>	<b>Level</b>	Minor fault	<b>Recovery</b>	Error reset	<b>Log category</b>
<b>Effects</b>	<b>User program</b>	Continues.	<b>Operation</b>	Operation is not possible for relevant axes group. Relevant axes group stops immediately if it is in motion.	
<b>System-defined variables</b>	<b>Variable</b>	<b>Data type</b>		<b>Name</b>	
	_MC_GRP[*].MFAultLvl.Active	BOOL		Axes Group Minor Fault Occurrence	
<b>Cause and correction</b>	<b>Assumed cause</b>		<b>Correction</b>		<b>Prevention</b>
	The transition mode specified for the <i>TransitionMode</i> input variable to a motion control instruction does not support the combination of the current instruction and buffered instruction.		Change the <i>TransitionMode</i> input variable to a correct transition mode.		Set the <i>TransitionMode</i> input variable correctly.
<b>Attached information</b>	None				
<b>Precautions/Remarks</b>	None				

## 6 Troubleshooting

<b>Event name</b>	Jog Mode Out of Range		<b>Event code</b>	57440000 hex		
<b>Meaning</b>	The <i>JogMode</i> input variable to the MC_RobotJog (Axes Group Jog) instruction is out of range.					
<b>Source</b>	Motion Control Function Module		<b>Source details</b>	Axes group	<b>Detection timing</b>	At instruction execution
<b>Error attributes</b>	<b>Level</b>	Minor fault	<b>Recovery</b>	Error reset	<b>Log category</b>	System
<b>Effects</b>	<b>User program</b>	Continues.	<b>Operation</b>	Operation is not possible for relevant axes group. Relevant axes group stops immediately if it is in motion.		
<b>System-defined variables</b>	<b>Variable</b>	<b>Data type</b>		<b>Name</b>		
	_MC_GRP[*].MFAultLvl.Active	BOOL		Axes Group Minor Fault Occurrence		
<b>Cause and correction</b>	<b>Assumed cause</b>		<b>Correction</b>		<b>Prevention</b>	
	The <i>JogMode</i> input variable to the MC_RobotJog (Axes Group Jog) instruction is outside the setting range.		Correct the value of the <i>JogMode</i> input variable.		Select the jog mode correctly.	
<b>Attached information</b>	None					
<b>Precautions/Remarks</b>	None					

<b>Event name</b>	Initial Workpiece Position Out of Range		<b>Event code</b>	57450000 hex		
<b>Meaning</b>	The <i>InitWorkpiecePosition</i> input variable to the MC_SyncLinearConveyor (Start Conveyor Synchronization) instruction is out of range.					
<b>Source</b>	Motion Control Function Module		<b>Source details</b>	Axes group	<b>Detection timing</b>	At instruction execution
<b>Error attributes</b>	<b>Level</b>	Minor fault	<b>Recovery</b>	Error reset	<b>Log category</b>	System
<b>Effects</b>	<b>User program</b>	Continues.	<b>Operation</b>	Operation is not possible for relevant axes group. Relevant axes group stops immediately if it is in motion.		
<b>System-defined variables</b>	<b>Variable</b>	<b>Data type</b>		<b>Name</b>		
	_MC_GRP[*].MFAultLvl.Active	BOOL		Axes Group Minor Fault Occurrence		
<b>Cause and correction</b>	<b>Assumed cause</b>		<b>Correction</b>		<b>Prevention</b>	
	The <i>InitWorkpiecePosition</i> input variable to the MC_SyncLinearConveyor (Start Conveyor Synchronization) instruction is outside the setting range.		Correct the value of the <i>InitWorkpiecePosition</i> input variable.		Review the program and make sure that the value of <i>InitWorkpiecePosition</i> input variable to the MC_SyncLinearConveyor (Start Conveyor Synchronization) instruction is within the setting range.	
<b>Attached information</b>	Attached Information 1: Element number of the <i>InitWorkpiecePosition</i> input variable for which the error occurred.					
<b>Precautions/Remarks</b>	None					

<b>Event name</b>	Maximum Interpolation Velocity Out of Range		<b>Event code</b>	57460000 hex	
<b>Meaning</b>	The <i>MaxVelocityTCP</i> input variable to the MC_SetKinTransform (Set Kinematics Transformation) instruction is out of range.				
<b>Source</b>	Motion Control Function Module		<b>Source details</b>	Axes group	<b>Detection timing</b>
<b>Error attributes</b>	<b>Level</b>	Minor fault	<b>Recovery</b>	Error reset	<b>Log category</b>
<b>Effects</b>	<b>User program</b>	Continues.	<b>Operation</b>	Operation is not possible for relevant axes group. Relevant axes group stops immediately if it is in motion.	
<b>System-defined variables</b>	<b>Variable</b>		<b>Data type</b>		<b>Name</b>
	_MC_GRP[*].MFaultLvl.Active		BOOL		Axes Group Minor Fault Occurrence
<b>Cause and correction</b>	<b>Assumed cause</b>		<b>Correction</b>		<b>Prevention</b>
	The <i>MaxVelocityTCP</i> input variable to the MC_SetKinTransform (Set Kinematics Transformation) instruction is outside the setting range.		Correct the value of the <i>MaxVelocityTCP</i> input variable.		Set the maximum interpolation velocity correctly.
<b>Attached information</b>	None				
<b>Precautions/Remarks</b>	None				

<b>Event name</b>	Maximum Interpolation Acceleration Out of Range		<b>Event code</b>	57470000 hex	
<b>Meaning</b>	The <i>MaxAccelerationTCP</i> input variable to the MC_SetKinTransform (Set Kinematics Transformation) instruction is out of range.				
<b>Source</b>	Motion Control Function Module		<b>Source details</b>	Axes group	<b>Detection timing</b>
<b>Error attributes</b>	<b>Level</b>	Minor fault	<b>Recovery</b>	Error reset	<b>Log category</b>
<b>Effects</b>	<b>User program</b>	Continues.	<b>Operation</b>	Operation is not possible for relevant axes group. Relevant axes group stops immediately if it is in motion.	
<b>System-defined variables</b>	<b>Variable</b>		<b>Data type</b>		<b>Name</b>
	_MC_GRP[*].MFaultLvl.Active		BOOL		Axes Group Minor Fault Occurrence
<b>Cause and correction</b>	<b>Assumed cause</b>		<b>Correction</b>		<b>Prevention</b>
	The <i>MaxAccelerationTCP</i> input variable to the MC_SetKinTransform (Set Kinematics Transformation) instruction is outside the setting range.		Correct the value of the <i>MaxAccelerationTCP</i> input variable.		Set the maximum interpolation acceleration correctly.
<b>Attached information</b>	None				
<b>Precautions/Remarks</b>	None				

## 6 Troubleshooting

<b>Event name</b>	Maximum Interpolation Deceleration Out of Range		<b>Event code</b>	57480000 hex	
<b>Meaning</b>	The <i>MaxDecelerationTCP</i> input variable to the MC_SetKinTransform (Set Kinematics Transformation) instruction is out of range.				
<b>Source</b>	Motion Control Function Module		<b>Source details</b>	Axes group	<b>Detection timing</b>
<b>Error attributes</b>	<b>Level</b>	Minor fault	<b>Recovery</b>	Error reset	<b>Log category</b>
<b>Effects</b>	<b>User program</b>	Continues.	<b>Operation</b>	Operation is not possible for relevant axes group. Relevant axes group stops immediately if it is in motion.	
<b>System-defined variables</b>	<b>Variable</b>	<b>Data type</b>		<b>Name</b>	
	_MC_GRP[*].MFaultLvl.Active	BOOL		Axes Group Minor Fault Occurrence	
<b>Cause and correction</b>	<b>Assumed cause</b>		<b>Correction</b>		<b>Prevention</b>
	The <i>MaxDecelerationTCP</i> input variable to the MC_SetKinTransform (Set Kinematics Transformation) instruction is outside the setting range.		Correct the value of the <i>MaxDecelerationTCP</i> input variable.		Set the maximum interpolation deceleration correctly.
<b>Attached information</b>	None				
<b>Precautions/Remarks</b>	None				

<b>Event name</b>	Target Velocity Ratio Setting Out of Range		<b>Event code</b>	574D0000 hex	
<b>Meaning</b>	The parameter specified for the <i>VelocityRatio</i> input variable to a motion control instruction is out of range.				
<b>Source</b>	Motion Control Function Module		<b>Source details</b>	Axes group	<b>Detection timing</b>
<b>Error attributes</b>	<b>Level</b>	Minor fault	<b>Recovery</b>	Error reset	<b>Log category</b>
<b>Effects</b>	<b>User program</b>	Continues.	<b>Operation</b>	Operation is not possible for relevant axes group. Relevant axes group decelerates to a stop if it is in motion.	
<b>System-defined variables</b>	<b>Variable</b>	<b>Data type</b>		<b>Name</b>	
	_MC_GRP[*].MFaultLvl.Active	BOOL		Axes Group Minor Fault Occurrence	
<b>Cause and correction</b>	<b>Assumed cause</b>		<b>Correction</b>		<b>Prevention</b>
	Instruction input parameter exceeded the valid range of the input variable.		Correct the parameter so that the valid range of the input variable is not exceeded for the relevant instruction.		Set the input parameter to the instruction so that the valid range of the input variable is not exceeded.
<b>Attached information</b>	None				
<b>Precautions/Remarks</b>	None				

<b>Event name</b>	Acceleration Ratio Setting Out of Range		<b>Event code</b>	574E0000 hex	
<b>Meaning</b>	The parameter specified for the <i>AccelerationRatio</i> input variable to a motion control instruction is out of range.				
<b>Source</b>	Motion Control Function Module		<b>Source details</b>	Axes group	<b>Detection timing</b> At instruction execution
<b>Error attributes</b>	<b>Level</b>	Minor fault	<b>Recovery</b>	Error reset	<b>Log category</b> System
<b>Effects</b>	<b>User program</b>	Continues.	<b>Operation</b>	Operation is not possible for relevant axes group. Relevant axes group decelerates to a stop if it is in motion.	
<b>System-defined variables</b>	<b>Variable</b>		<b>Data type</b>		<b>Name</b>
	_MC_GRP[*].MFAultLvl.Active		BOOL		Axes Group Minor Fault Occurrence
<b>Cause and correction</b>	<b>Assumed cause</b>		<b>Correction</b>		<b>Prevention</b>
	Instruction input parameter exceeded the valid range of the input variable.		Correct the parameter so that the valid range of the input variable is not exceeded for the relevant instruction.		Set the input parameter to the instruction so that the valid range of the input variable is not exceeded.
<b>Attached information</b>	None				
<b>Precautions/Remarks</b>	None				

<b>Event name</b>	Deceleration Ratio Setting Out of Range		<b>Event code</b>	574F0000 hex	
<b>Meaning</b>	The parameter specified for the <i>DecelerationRatio</i> input variable to a motion control instruction is out of range.				
<b>Source</b>	Motion Control Function Module		<b>Source details</b>	Axes group	<b>Detection timing</b> At instruction execution
<b>Error attributes</b>	<b>Level</b>	Minor fault	<b>Recovery</b>	Error reset	<b>Log category</b> System
<b>Effects</b>	<b>User program</b>	Continues.	<b>Operation</b>	Operation is not possible for relevant axes group. Relevant axes group decelerates to a stop if it is in motion.	
<b>System-defined variables</b>	<b>Variable</b>		<b>Data type</b>		<b>Name</b>
	_MC_GRP[*].MFAultLvl.Active		BOOL		Axes Group Minor Fault Occurrence
<b>Cause and correction</b>	<b>Assumed cause</b>		<b>Correction</b>		<b>Prevention</b>
	Instruction input parameter exceeded the valid range of the input variable.		Correct the parameter so that the valid range of the input variable is not exceeded for the relevant instruction.		Set the input parameter to the instruction so that the valid range of the input variable is not exceeded.
<b>Attached information</b>	None				
<b>Precautions/Remarks</b>	None				

## 6 Troubleshooting

<b>Event name</b>	Arm Configuration Setting Out of Range		<b>Event code</b>	57500000 hex		
<b>Meaning</b>	The parameter specified for the <i>ArmConfig</i> input variable to a motion control instruction is out of range.					
<b>Source</b>	Motion Control Function Module		<b>Source details</b>	Axes group	<b>Detection timing</b>	At instruction execution
<b>Error attributes</b>	<b>Level</b>	Minor fault	<b>Recovery</b>	Error reset	<b>Log category</b>	System
<b>Effects</b>	<b>User program</b>	Continues.	<b>Operation</b>	Operation is not possible for relevant axes group. Relevant axes group decelerates to a stop if it is in motion.		
<b>System-defined variables</b>	<b>Variable</b>		<b>Data type</b>		<b>Name</b>	
	_MC_GRP[*].MFAultLvl.Active		BOOL		Axes Group Minor Fault Occurrence	
<b>Cause and correction</b>	<b>Assumed cause</b>		<b>Correction</b>		<b>Prevention</b>	
	Instruction input parameter exceeded the valid range of the input variable.		Correct the parameter so that the valid range of the input variable is not exceeded for the relevant instruction.		Set the input parameter to the instruction so that the valid range of the input variable is not exceeded.	
<b>Attached information</b>	None					
<b>Precautions/Remarks</b>	None					

<b>Event name</b>	Positive Limit Input		<b>Event code</b>	64430000 hex	
<b>Meaning</b>	An instruction was executed for a motion in the positive direction when the positive limit input was ON.				
<b>Source</b>	Motion Control Function Module		<b>Source details</b>	Axis/axes group	<b>Detection timing</b> At instruction execution
<b>Error attributes</b>	<b>Level</b>	Minor fault	<b>Recovery</b>	Error reset	<b>Log category</b> System
<b>Effects</b>	<b>User program</b>	Continues.	<b>Operation</b>	If "axis" is given for the source details, operation is not possible for relevant axis. If "axes group" is given for the source details, operation is not possible for relevant axes group.	
<b>System-defined variables</b>	<b>Variable</b>		<b>Data type</b>		<b>Name</b>
	_MC_AX[*].MFAultLvl.Active		BOOL		Axis Minor Fault Occurrence
	_MC_GRP[*].MFAultLvl.Active		BOOL		Axes Group Minor Fault Occurrence
<b>Cause and correction</b>	<b>Assumed cause</b>		<b>Correction</b>		<b>Prevention</b>
	An instruction for a motion in the positive direction was executed when the positive limit input was ON, or an instruction for a motion with no direction specification was executed when the positive limit input was ON. An axes group motion control instruction was executed when the positive limit input was ON.		Execute an error reset and then perform a recovery operation in the negative direction. If the error occurred during an axes group motion control instruction, disable the axes group and then perform the above operation. If this error occurs again, check the connection of the positive limit signal, the logic setting for the positive limit input, and the execution conditions for the start command, and correct any mistakes. Check the logic settings both in the axis parameters and in the slave settings.		Check to make sure there are no problems with the positive limit signal connection, the logic setting for the positive limit input, and the execute conditions for the instruction. Check the logic settings both in the axis parameters and in the slave settings.
<b>Attached information</b>	Depends on the source details. Axis: 0 Axes group: Logical axis number				
<b>Precautions/Remarks</b>	None				

## 6 Troubleshooting

<b>Event name</b>	Negative Limit Input		<b>Event code</b>	64440000 hex	
<b>Meaning</b>	An instruction for a motion in the negative direction was executed when the negative limit input was ON.				
<b>Source</b>	Motion Control Function Module		<b>Source details</b>	Axis/axes group	<b>Detection timing</b> At instruction execution
<b>Error attributes</b>	<b>Level</b>	Minor fault	<b>Recovery</b>	Error reset	<b>Log category</b> System
<b>Effects</b>	<b>User program</b>	Continues.	<b>Operation</b>	If "axis" is given for the source details, operation is not possible for relevant axis. If "axes group" is given for the source details, operation is not possible for relevant axes group.	
<b>System-defined variables</b>	<b>Variable</b>		<b>Data type</b>		<b>Name</b>
	_MC_AX[*].MFAultLvl.Active		BOOL		Axis Minor Fault Occurrence
	_MC_GRP[*].MFAultLvl.Active		BOOL		Axes Group Minor Fault Occurrence
<b>Cause and correction</b>	<b>Assumed cause</b>		<b>Correction</b>		<b>Prevention</b>
	An instruction for a motion in the negative direction was executed when the negative limit input was ON, or an instruction for a motion with no direction specification was executed when the negative limit input was ON. An axes group motion control instruction was executed when the negative limit input was ON.		Execute an error reset and then perform a recovery operation in the positive direction. If the error occurred during an axes group motion control instruction, disable the axes group and then perform the above operation. If this error occurs again, check the connection of the negative limit signal, the logic setting for the negative limit input, and the execution conditions for the start command, and correct any mistakes. Check the logic settings both in the axis parameters and in the slave settings.		Check to make sure there are no problems with the negative limit signal connection, the logic setting for the negative limit input, and the execute conditions for the instruction. Check the logic settings both in the axis parameters and in the slave settings.
<b>Attached information</b>	Depends on the source details. Axis: 0 Axes group: Logical axis number				
<b>Precautions/Remarks</b>	None				

<b>Event name</b>	Home Undefined during Coordinated Motion		<b>Event code</b>	64590000 hex	
<b>Meaning</b>	Home of the logical axis became undefined during axes group motion or while decelerating to a stop.				
<b>Source</b>	Motion Control Function Module		<b>Source details</b>	Axes group	<b>Detection timing</b> During instruction execution
<b>Error attributes</b>	<b>Level</b>	Minor fault	<b>Recovery</b>	Error reset	<b>Log category</b> System
<b>Effects</b>	<b>User program</b>	Continues.	<b>Operation</b>	The axes group decelerates to a stop.	
<b>System-defined variables</b>	<b>Variable</b>		<b>Data type</b>	<b>Name</b>	
		_MC_GRP[*].MFaultLvl.Active	BOOL	Axes Group Minor Fault Occurrence	
<b>Cause and correction</b>	<b>Assumed cause</b>		<b>Correction</b>	<b>Prevention</b>	
		The command position or actual position overflowed or underflowed for a logical axis in an axes group motion or a logical axis that was decelerating to a stop and the home definition was lost.	Correct the program so that the axis operates within ranges that do not cause overflows or underflows in the command position or actual position.	Write the program so that the axis operates within ranges that do not cause overflows or underflows in the command position or actual position.	
		A slave communications error occurred for a logical axis and home became undefined during axes group motion or while decelerating to a stop.	Correct the slave communications error and define home.	None	
		A slave for a logical axis left the network or was disabled and home became undefined during axes group motion or while decelerating to a stop.	Connect the disconnected or disabled Slave to the network again and define home.	Do not disconnect or disable the slave of a logical axis during axes group motion or while decelerating to a stop.	
<b>Attached information</b>	None				
<b>Precautions/Remarks</b>	None				

## 6 Troubleshooting

<b>Event name</b>	Maximum Interpolation Velocity Error		<b>Event code</b>	645A0000 hex		
<b>Meaning</b>	The command velocity exceeded the maximum interpolation velocity specified for the <i>MaxVelocityTCP</i> input variable to the MC_SetKinTransform (Set Kinematics Transformation) instruction.					
<b>Source</b>	Motion Control Function Module		<b>Source details</b>	Axes group	<b>Detection timing</b>	During instruction execution
<b>Error attributes</b>	<b>Level</b>	Minor fault	<b>Recovery</b>	Error reset	<b>Log category</b>	System
<b>Effects</b>	<b>User program</b>	Continues.	<b>Operation</b>	Operation is not possible for relevant axes group. Relevant axes group stops immediately if it is in motion.		
<b>System-defined variables</b>	<b>Variable</b>	<b>Data type</b>		<b>Name</b>		
	_MC_GRP[*].MFaultLvl.Active	BOOL		Axes Group Minor Fault Occurrence		
<b>Cause and correction</b>	<b>Assumed cause</b>	<b>Correction</b>		<b>Prevention</b>		
	The trajectory data for the current instruction is incorrect.	Change the trajectory data so that the command velocity is lowered.		Set the parameters so that the command velocity for trajectory does not exceed the maximum interpolation velocity specified for the MC_SetKinTransform (Set Kinematics Transformation) instruction.		
	The maximum interpolation velocity specified for the <i>MaxVelocityTCP</i> input variable to the MC_SetKinTransform (Set Kinematics Transformation) instruction is incorrect or too low.	Change the value of the <i>MaxVelocityTCP</i> input variable to an appropriate value.		Set the maximum interpolation velocity correctly.		
<b>Attached information</b>	None					
<b>Precautions/Remarks</b>	None					

<b>Event name</b>	Maximum Interpolation Acceleration Error		<b>Event code</b>	645B0000 hex	
<b>Meaning</b>	The command acceleration rate exceeded the maximum interpolation acceleration specified for the <i>MaxAccelerationTCP</i> input variable to the MC_SetKinTransform (Set Kinematics Transformation) instruction.				
<b>Source</b>	Motion Control Function Module		<b>Source details</b>	Axes group	<b>Detection timing</b> During instruction execution
<b>Error attributes</b>	<b>Level</b>	Minor fault	<b>Recovery</b>	Error reset	<b>Log category</b> System
<b>Effects</b>	<b>User program</b>	Continues.	<b>Operation</b>	Operation is not possible for relevant axes group. Relevant axes group stops immediately if it is in motion.	
<b>System-defined variables</b>	<b>Variable</b>		<b>Data type</b>	<b>Name</b>	
		_MC_GRP[*].MFAultLvl.Active	BOOL	Axes Group Minor Fault Occurrence	
<b>Cause and correction</b>	<b>Assumed cause</b>		<b>Correction</b>	<b>Prevention</b>	
		The trajectory data for the current instruction is incorrect.	Change the parameters so that the command acceleration rate is lowered.	Set the parameters so that the command acceleration rate for trajectory does not exceed the maximum interpolation acceleration specified for the MC_SetKinTransform (Set Kinematics Transformation) instruction.	
		The maximum interpolation acceleration specified for the <i>MaxAccelerationTCP</i> input variable to the MC_SetKinTransform (Set Kinematics Transformation) instruction is incorrect or too low.	Change the value of the <i>MaxAccelerationTCP</i> input variable to an appropriate value.	Set the maximum interpolation acceleration correctly.	
<b>Attached information</b>	None				
<b>Precautions/Remarks</b>	None				

## 6 Troubleshooting

<b>Event name</b>	Maximum Interpolation Deceleration Error		<b>Event code</b>	645C0000 hex			
<b>Meaning</b>	The command deceleration rate exceeded the maximum interpolation deceleration specified for the <i>MaxDecelerationTCP</i> input variable to the MC_SetKinTransform (Set Kinematics Transformation) instruction.						
<b>Source</b>	Motion Control Function Module		<b>Source details</b>	Axes group	<b>Detection timing</b>	During instruction execution	
<b>Error attributes</b>	<b>Level</b>	Minor fault	<b>Recovery</b>	Error reset	<b>Log category</b>	System	
<b>Effects</b>	<b>User program</b>	Continues.	<b>Operation</b>	Operation is not possible for relevant axes group. Relevant axes group stops immediately if it is in motion.			
<b>System-defined variables</b>	<b>Variable</b>		<b>Data type</b>		<b>Name</b>		
	_MC_GRP[*].MFaultLvl.Active		BOOL		Axes Group Minor Fault Occurrence		
<b>Cause and correction</b>	<b>Assumed cause</b>		<b>Correction</b>		<b>Prevention</b>		
	The trajectory data for the current instruction is incorrect.		Change the parameters so that the command deceleration rate is lowered.		Set the parameters so that the command deceleration rate for trajectory does not exceed the maximum interpolation deceleration specified for the MC_SetKinTransform (Set Kinematics Transformation) instruction.		
		The maximum interpolation deceleration specified for the <i>MaxDecelerationTCP</i> input variable to the MC_SetKinTransform (Set Kinematics Transformation) instruction is incorrect or too low.		Change the value of the <i>MaxDecelerationTCP</i> input variable to an appropriate value.		Set the maximum interpolation deceleration correctly.	
<b>Attached information</b>	None						
<b>Precautions/Remarks</b>	None						

<b>Event name</b>	Command Position Outside Workspace		<b>Event code</b>	67000000 hex		
<b>Meaning</b>	The command position is outside the workspace.					
<b>Source</b>	Motion Control Function Module		<b>Source details</b>	Axes group	<b>Detection timing</b>	At or during instruction execution
<b>Error attributes</b>	<b>Level</b>	Minor fault	<b>Recovery</b>	Error reset	<b>Log category</b>	System
<b>Effects</b>	<b>User program</b>	Continues.	<b>Operation</b>	Operation is not possible for relevant axes group. Relevant axes group stops immediately if it is in motion.		
<b>System-defined variables</b>	<b>Variable</b>		<b>Data type</b>		<b>Name</b>	
	_MC_GRP[*].MFaultLvl.Active		BOOL		Axes Group Minor Fault Occurrence	
<b>Cause and correction</b>	<b>Assumed cause</b>		<b>Correction</b>		<b>Prevention</b>	
	The specified trajectory data is incorrect.		Correct the trajectory data so that the operation is performed inside the workspace.		Set the trajectory data so that the command position for trajectory does not exceed the range of the workspace.	
<b>Attached information</b>	None					
<b>Precautions/Remarks</b>	None					

<b>Event name</b>	Current Position Outside Workspace		<b>Event code</b>	67010000 hex	
<b>Meaning</b>	The current position was outside the workspace when an instruction was executed.				
<b>Source</b>	Motion Control Function Module		<b>Source details</b>	Axes group	<b>Detection timing</b> At instruction execution
<b>Error attributes</b>	<b>Level</b>	Minor fault	<b>Recovery</b>	Error reset	<b>Log category</b> System
<b>Effects</b>	<b>User program</b>	Continues.	<b>Operation</b>	Operation is not possible for relevant axes group. Relevant axes group stops immediately if it is in motion.	
<b>System-defined variables</b>	<b>Variable</b>	<b>Data type</b>		<b>Name</b>	
	_MC_GRP[*].MFAultLvl.Active	BOOL		Axes Group Minor Fault Occurrence	
<b>Cause and correction</b>	<b>Assumed cause</b>		<b>Correction</b>		<b>Prevention</b>
	<p>The current position was outside the workspace when one of the following robot instructions was executed.</p> <ul style="list-style-type: none"> <li>• MC_MoveTimeAbsolute (Time-specified Absolute Positioning) instruction</li> <li>• MC_SyncLinearConveyor (Start Conveyor Synchronization) instruction</li> <li>• MC_SyncOut (End Synchronization) instruction</li> <li>• MC_RobotJog (Axes Group Jog) instruction</li> </ul>		Move the robot inside the workspace.		Review the program and make sure that the robot moves within the range of the workspace.
<b>Attached information</b>	None				
<b>Precautions/Remarks</b>	None				

## 6 Troubleshooting

<b>Event name</b>	Workpiece Synchronization Excessive Following Error		<b>Event code</b>	67020000 hex		
<b>Meaning</b>	The robot could not catch up the workpiece on the conveyor.					
<b>Source</b>	Motion Control Function Module		<b>Source details</b>	Axes group	<b>Detection timing</b>	During instruction execution
<b>Error attributes</b>	<b>Level</b>	Minor fault	<b>Recovery</b>	Error reset	<b>Log category</b>	System
<b>Effects</b>	<b>User program</b>	Continues.	<b>Operation</b>	Operation is not possible for relevant axes group. Relevant axes group stops immediately if it is in motion.		
<b>System-defined variables</b>	<b>Variable</b>	<b>Data type</b>		<b>Name</b>		
	_MC_GRP[*].MFaultLvl.Active	BOOL		Axes Group Minor Fault Occurrence		
<b>Cause and correction</b>	<b>Assumed cause</b>		<b>Correction</b>		<b>Prevention</b>	
	The conveyor position was changed before the robot reached the workpiece.		Operate the conveyor at a constant velocity. Or, do not change the velocity suddenly.		Operate the conveyor at a constant velocity.	
	The current position of the conveyor is incorrect. (Out of range, encoder overflow, etc.)		Correct the current position of the conveyor.		Make the setting so that the current position of the conveyor is updated correctly.	
<b>Attached information</b>	None					
<b>Precautions/Remarks</b>	None					

<b>Event name</b>	Velocity Error Detection		<b>Event code</b>	67030000 hex		
<b>Meaning</b>	The command velocity exceeded <i>MaxVelocity</i> specified for the <i>TrajData</i> input variable to a motion control instruction.					
<b>Source</b>	Motion Control Function Module		<b>Source details</b>	Axes group	<b>Detection timing</b>	At or during instruction execution
<b>Error attributes</b>	<b>Level</b>	Minor fault	<b>Recovery</b>	Error reset	<b>Log category</b>	System
<b>Effects</b>	<b>User program</b>	Continues.	<b>Operation</b>	Operation is not possible for relevant axes group. Relevant axes group stops immediately if it is in motion.		
<b>System-defined variables</b>	<b>Variable</b>	<b>Data type</b>		<b>Name</b>		
	_MC_GRP[*].MFaultLvl.Active	BOOL		Axes Group Minor Fault Occurrence		
<b>Cause and correction</b>	<b>Assumed cause</b>		<b>Correction</b>		<b>Prevention</b>	
	The combination of parameters specified for the <i>TrajData</i> input variable caused a too high command velocity.		Correct the parameters so that the command velocity is lower than <i>MaxVelocity</i> .		Set the parameter so that the command velocity for trajectory is lower than the velocity error detection value.	
	The parameters specified for the <i>TrajData</i> input variable are incorrect.		Correct the parameters.		Set the <i>TrajData</i> input variable correctly.	
	<i>MaxVelocity</i> is too low.		Change the value of the <i>MaxVelocity</i> of the <i>TrajData</i> input variable to an appropriate value.		Set the velocity error detection value correctly.	
<b>Attached information</b>	None					
<b>Precautions/Remarks</b>	None					

<b>Event name</b>	Acceleration Error Detection		<b>Event code</b>	67040000 hex		
<b>Meaning</b>	The command acceleration rate exceeded the <i>MaxAcceleration</i> specified for the <i>TrajData</i> input variable to a motion control instruction.					
<b>Source</b>	Motion Control Function Module		<b>Source details</b>	Axes group	<b>Detection timing</b>	At or during instruction execution
<b>Error attributes</b>	<b>Level</b>	Minor fault	<b>Recovery</b>	Error reset	<b>Log category</b>	System
<b>Effects</b>	<b>User program</b>	Continues.	<b>Operation</b>	Operation is not possible for relevant axes group. Relevant axes group stops immediately if it is in motion.		
<b>System-defined variables</b>	<b>Variable</b>	<b>Data type</b>		<b>Name</b>		
	_MC_GRP[*].MFaultLvl.Active	BOOL		Axes Group Minor Fault Occurrence		
<b>Cause and correction</b>	<b>Assumed cause</b>		<b>Correction</b>		<b>Prevention</b>	
	The combination of parameters specified for the <i>TrajData</i> input variable caused a too high command acceleration rate.		Correct the parameters so that the command acceleration rate is lower than <i>MaxAcceleration</i> .		Set the parameters so that the command velocity for trajectory is lower than the acceleration error detection value.	
	The parameters specified for the <i>TrajData</i> input variable are incorrect.		Correct the parameters.		Set the <i>TrajData</i> input variable correctly.	
	<i>MaxAcceleration</i> is too low.		Change the value of the <i>MaxAcceleration</i> of the <i>TrajData</i> input variable to an appropriate value.		Set the acceleration error detection value correctly.	
<b>Attached information</b>	None					
<b>Precautions/Remarks</b>	None					

<b>Event name</b>	Command Current Velocity Limit Exceeded		<b>Event code</b>	67050000 hex		
<b>Meaning</b>	The command current velocity exceeded the axis maximum velocity.					
<b>Source</b>	Motion Control Function Module		<b>Source details</b>	Axes group	<b>Detection timing</b>	During instruction execution
<b>Error attributes</b>	<b>Level</b>	Minor fault	<b>Recovery</b>	Error reset	<b>Log category</b>	System
<b>Effects</b>	<b>User program</b>	Continues.	<b>Operation</b>	Operation is not possible for relevant axes group. Relevant axes group stops immediately if it is in motion.		
<b>System-defined variables</b>	<b>Variable</b>	<b>Data type</b>		<b>Name</b>		
	_MC_GRP[*].MFaultLvl.Active	BOOL		Axes Group Minor Fault Occurrence		
<b>Cause and correction</b>	<b>Assumed cause</b>		<b>Correction</b>		<b>Prevention</b>	
	The axis maximum velocity was exceeded.		Reduce the velocity for robot operation.		Set the input parameters so that they do not exceed the axes maximum velocity.	
<b>Attached information</b>	Attached Information 1: Depends on the source details. Axis: 0 Axes group: Number of the logical axis where the error occurred					
<b>Precautions/Remarks</b>	None					

## 6 Troubleshooting

<b>Event name</b>	Axes Group Composition Axis Error		<b>Event code</b>	74300000 hex	
<b>Meaning</b>	An error occurred for an axis in an axes group.				
<b>Source</b>	Motion Control Function Module		<b>Source details</b>	Axes group	<b>Detection timing</b>
<b>Error attributes</b>	<b>Level</b>	Minor fault	<b>Recovery</b>	Error reset	<b>Log category</b>
<b>Effects</b>	<b>User program</b>	Continues.	<b>Operation</b>	If one of the composition axes stops immediately, the axes group operation will follow the setting of the Axes Group Stop Method Selection. Otherwise, an interpolated trajectory stop is performed.	
<b>System-defined variables</b>	<b>Variable</b>	<b>Data type</b>		<b>Name</b>	
	_MC_GRP[*].MFaultLvl.Active	BOOL		Axes Group Minor Fault Occurrence	
<b>Cause and correction</b>	<b>Assumed cause</b>		<b>Correction</b>		<b>Prevention</b>
	An error occurred for an axis in an axes group that was in motion.		Check the error code of the axes in the axes group and remove the cause of the error.		None
<b>Attached information</b>	None				
<b>Precautions/Remarks</b>	None				

<b>Event name</b>	Conveyor Axis Position Read Error		<b>Event code</b>	77000000 hex	
<b>Meaning</b>	The MC_SyncLinearConveyor (Start Conveyor Synchronization) instruction cannot be executed due to a conveyor axis position error.				
<b>Source</b>	Motion Control Function Module		<b>Source details</b>	Axes group	<b>Detection timing</b> At instruction execution
<b>Error attributes</b>	<b>Level</b>	Minor fault	<b>Recovery</b>	Error reset	<b>Log category</b> System
<b>Effects</b>	<b>User program</b>	Continues.	<b>Operation</b>	Relevant axes group stops immediately.	
<b>System-defined variables</b>	<b>Variable</b>		<b>Data type</b>		<b>Name</b>
	_MC_GRP[*].MFAultLvl.Active		BOOL		Axes Group Minor Fault Occurrence
<b>Cause and correction</b>	<b>Assumed cause</b>		<b>Correction</b>		<b>Prevention</b>
	EtherCAT process data communications are not established for the conveyor axis.		If the <code>_EC_PDSlavTbl</code> (Process Data Communicating Slave Table) system-defined variable for the EtherCAT master of the conveyor axis is FALSE, investigate the error in the conveyor axis and remove the cause.		If you execute the MC_SyncLinearConveyor (Start Conveyor Synchronization) instruction right after you turn ON the power supply, download data, or reset slave communications error, make sure that the <code>_EC_PDSlavTbl</code> (Process Data Communicating Slave Table) system-defined variable for the EtherCAT master is TRUE for the node of the conveyor axis before you execute the instruction.
	The slave of the conveyor axis was disconnected.		See if a slave of the conveyor axis is disconnected.		Do not disconnect a slave from the master axis during execution of the MC_SyncLinearConveyor (Start Conveyor Synchronization) instruction.
An Absolute Encoder Current Position Calculation Failed error (64580000 hex) was detected for the conveyor axis.		See if an Absolute Encoder Current Position Calculation Failed error (64580000 hex) occurred for the conveyor axis and make suitable corrections to restore operation.		Do not execute the MC_SyncLinearConveyor (Start Conveyor Synchronization) instruction for a conveyor axis in an Absolute Encoder Current Position Calculation Failed error (64580000 hex).	
<b>Attached information</b>	None				
<b>Precautions/Remarks</b>	None				

## 6 Troubleshooting

<b>Event name</b>	Transition Parameter Adjusted		<b>Event code</b>	94230000 hex		
<b>Meaning</b>	The specified <i>TransitionParameter</i> input variable was adjusted before transition started.					
<b>Source</b>	Motion Control Function Module		<b>Source details</b>	Axes group	<b>Detection timing</b>	At multi-execution of instructions
<b>Error attributes</b>	<b>Level</b>	Observation	<b>Recovery</b>	Error reset	<b>Log category</b>	System
<b>Effects</b>	<b>User program</b>	Continues.	<b>Operation</b>	Not affected.		
<b>System-defined variables</b>	<b>Variable</b>		<b>Data type</b>		<b>Name</b>	
	_MC_GRP[*].Obsr.Active		BOOL		Axes Group Observation Occurrence	
<b>Cause and correction</b>	<b>Assumed cause</b>		<b>Correction</b>		<b>Prevention</b>	
	The transition start condition specified for <i>TransitionParameter</i> input variable for multi-execution of instructions was already exceeded by the current instruction.		Correct the transition start condition so that the operation time or operation distance for the current instruction is not exceeded. Correct the timing for multi-execution of instructions so that the remaining time and operation distance for the current instruction meet the transition start condition.		Set the transition start condition so that the operation time or operation distance for the current instruction is not exceeded. Start multi-execution of instructions only when the remaining time and operation distance for the current instruction meet the transition start condition.	
<b>Attached information</b>	None					
<b>Precautions/Remarks</b>	None					

## Errors Related to Robot Instructions

<b>Event name</b>	Target Velocity Setting Out of Range		<b>Event code</b>	54015422 hex		
<b>Meaning</b>	The parameter specified for the <i>Velocity</i> input variable to a motion control instruction is out of range.					
<b>Source</b>	PLC Function Module		<b>Source details</b>	Instruction	<b>Detection timing</b>	At instruction execution
<b>Error attributes</b>	<b>Level</b>	Observation	<b>Recovery</b>	---	<b>Log category</b>	System
<b>Effects</b>	<b>User program</b>	Continues	<b>Operation</b>	The relevant instruction will end according to specifications.		
<b>System-defined variables</b>	<b>Variable</b>		<b>Data type</b>		<b>Name</b>	
	None		---		---	
<b>Cause and correction</b>	<b>Assumed cause</b>		<b>Correction</b>		<b>Prevention</b>	
	Instruction input parameter exceeded the valid range of the input variable.		Correct the parameter so that the valid range of the input variable is not exceeded for the relevant instruction.		Set the input parameter to the instruction so that the valid range of the input variable is not exceeded.	
<b>Attached information</b>	Attached Information 1: Error Location Attached Information 2: Error Location Details (Rung Number). For a program section, the rung number from the start of the section is given. For ST, the line number is given. Attached Information 3: Names of the Instruction and Instruction Instance Where the Error Occurred. If there is more than one possible instruction, information is given on all of them. Nothing is given if the instruction cannot be identified. Attached Information 4: Expansion Error Code ( <i>ErrorIDEx</i> )					
<b>Precautions/Remarks</b>	If a program is changed after an error occurs, the attached information that is displayed may not be correct.					

<b>Event name</b>	Acceleration Setting Out of Range		<b>Event code</b>	54015423 hex	
<b>Meaning</b>	The parameter specified for the <i>Acceleration</i> input variable to a motion control instruction is out of range.				
<b>Source</b>	PLC Function Module		<b>Source details</b>	Instruction	<b>Detection timing</b> At instruction execution
<b>Error attributes</b>	<b>Level</b>	Observation	<b>Recovery</b>	---	<b>Log category</b> System
<b>Effects</b>	<b>User program</b>	Continues	<b>Operation</b>	The relevant instruction will end according to specifications.	
<b>System-defined variables</b>	<b>Variable</b>		<b>Data type</b>		<b>Name</b>
	None		---		---
<b>Cause and correction</b>	<b>Assumed cause</b>		<b>Correction</b>		<b>Prevention</b>
	Instruction input parameter exceeded the valid range of the input variable.		Correct the parameter so that the valid range of the input variable is not exceeded for the relevant instruction.		Set the input parameter to the instruction so that the valid range of the input variable is not exceeded.
<b>Attached information</b>	<p>Attached Information 1: Error Location</p> <p>Attached Information 2: Error Location Details (Rung Number). For a program section, the rung number from the start of the section is given. For ST, the line number is given.</p> <p>Attached Information 3: Names of the Instruction and Instruction Instance Where the Error Occurred. If there is more than one possible instruction, information is given on all of them. Nothing is given if the instruction cannot be identified.</p> <p>Attached Information 4: Expansion Error Code (<i>ErrorIDEx</i>)</p>				
<b>Precautions/Remarks</b>	If a program is changed after an error occurs, the attached information that is displayed may not be correct.				

<b>Event name</b>	Deceleration Setting Out of Range		<b>Event code</b>	54015424 hex	
<b>Meaning</b>	The parameter specified for the <i>Deceleration</i> input variable to a motion control instruction is out of range.				
<b>Source</b>	PLC Function Module		<b>Source details</b>	Instruction	<b>Detection timing</b> At instruction execution
<b>Error attributes</b>	<b>Level</b>	Observation	<b>Recovery</b>	---	<b>Log category</b> System
<b>Effects</b>	<b>User program</b>	Continues	<b>Operation</b>	The relevant instruction will end according to specifications.	
<b>System-defined variables</b>	<b>Variable</b>		<b>Data type</b>		<b>Name</b>
	None		---		---
<b>Cause and correction</b>	<b>Assumed cause</b>		<b>Correction</b>		<b>Prevention</b>
	Instruction input parameter exceeded the valid range of the input variable.		Correct the parameter so that the valid range of the input variable is not exceeded for the relevant instruction.		Set the input parameter to the instruction so that the valid range of the input variable is not exceeded.
<b>Attached information</b>	<p>Attached Information 1: Error Location</p> <p>Attached Information 2: Error Location Details (Rung Number). For a program section, the rung number from the start of the section is given. For ST, the line number is given.</p> <p>Attached Information 3: Names of the Instruction and Instruction Instance Where the Error Occurred. If there is more than one possible instruction, information is given on all of them. Nothing is given if the instruction cannot be identified.</p> <p>Attached Information 4: Expansion Error Code (<i>ErrorIDEx</i>)</p>				
<b>Precautions/Remarks</b>	If a program is changed after an error occurs, the attached information that is displayed may not be correct.				

## 6 Troubleshooting

<b>Event name</b>	Jerk Setting Out of Range		<b>Event code</b>	54015425 hex	
<b>Meaning</b>	The parameter specified for the <i>Jerk</i> input variable to a motion control instruction is out of range.				
<b>Source</b>	PLC Function Module		<b>Source details</b>	Instruction	<b>Detection timing</b> At instruction execution
<b>Error attributes</b>	<b>Level</b>	Observation	<b>Recovery</b>	---	<b>Log category</b> System
<b>Effects</b>	<b>User program</b>	Continues.	<b>Operation</b>	The relevant instruction will end according to specifications.	
<b>System-defined variables</b>	<b>Variable</b>		<b>Data type</b>		<b>Name</b>
	_MC_AX[*].MFaultLvl.Active		BOOL		Axis Minor Fault Occurrence
	_MC_GRP[*].MFaultLvl.Active		BOOL		Axes Group Minor Fault Occurrence
<b>Cause and correction</b>	<b>Assumed cause</b>		<b>Correction</b>		<b>Prevention</b>
	Instruction input parameter exceeded the valid range of the input variable.		Correct the parameter so that the valid range of the input variable is not exceeded for the relevant instruction.		Set the input parameter to the instruction so that the valid range of the input variable is not exceeded.
<b>Attached information</b>	<p>Attached Information 1: Error Location</p> <p>Attached Information 2: Error Location Details (Rung Number). For a program section, the rung number from the start of the section is given. For ST, the line number is given.</p> <p>Attached Information 3: Names of the Instruction and Instruction Instance Where the Error Occurred. If there is more than one possible instruction, information is given on all of them. Nothing is given if the instruction cannot be identified.</p> <p>Attached Information 4: Expansion Error Code (<i>ErrorIDEx</i>)</p>				
<b>Precautions/Remarks</b>	If a program is changed after an error occurs, the attached information that is displayed may not be correct.				

<b>Event name</b>	Buffer Mode Selection Out of Range		<b>Event code</b>	5401542B hex	
<b>Meaning</b>	The parameter specified for the <i>BufferMode</i> input variable to a motion control instruction is out of range.				
<b>Source</b>	PLC Function Module		<b>Source details</b>	Instruction	<b>Detection timing</b> At instruction execution
<b>Error attributes</b>	<b>Level</b>	Observation	<b>Recovery</b>	---	<b>Log category</b> System
<b>Effects</b>	<b>User program</b>	Continues	<b>Operation</b>	The relevant instruction will end according to specifications.	
<b>System-defined variables</b>	<b>Variable</b>		<b>Data type</b>		<b>Name</b>
	None		---		---
<b>Cause and correction</b>	<b>Assumed cause</b>		<b>Correction</b>		<b>Prevention</b>
	Instruction input parameter exceeded the valid range of the input variable.		Correct the parameter so that the valid range of the input variable is not exceeded for the instruction.		Set the input parameter to the instruction so that the valid range of the input variable is not exceeded.
<b>Attached information</b>	<p>Attached Information 1: Error Location</p> <p>Attached Information 2: Error Location Details (Rung Number). For a program section, the rung number from the start of the section is given. For ST, the line number is given.</p> <p>Attached Information 3: Names of the Instruction and Instruction Instance Where the Error Occurred. If there is more than one possible instruction, information is given on all of them. Nothing is given if the instruction cannot be identified.</p> <p>Attached Information 4: Expansion Error Code (<i>ErrorIDEx</i>)</p>				
<b>Precautions/Remarks</b>	If a program is changed after an error occurs, the attached information that is displayed may not be correct.				

<b>Event name</b>	Coordinate System Selection Out of Range		<b>Event code</b>	5401542C hex	
<b>Meaning</b>	The parameter specified for the <i>CoordSystem</i> input variable to a motion control instruction is out of range.				
<b>Source</b>	PLC Function Module		<b>Source details</b>	Instruction	<b>Detection timing</b> At instruction execution
<b>Error attributes</b>	<b>Level</b>	Observation	<b>Recovery</b>	---	<b>Log category</b> System
<b>Effects</b>	<b>User program</b>	Continues	<b>Operation</b>	The relevant instruction will end according to specifications.	
<b>System-defined variables</b>	<b>Variable</b>		<b>Data type</b>		<b>Name</b>
	None		---		---
<b>Cause and correction</b>	<b>Assumed cause</b>		<b>Correction</b>		<b>Prevention</b>
	Instruction input parameter exceeded the valid range of the input variable.		Correct the parameter so that the valid range of the input variable is not exceeded for the instruction.		Set the input parameter to the instruction so that the valid range of the input variable is not exceeded.
<b>Attached information</b>	<p>Attached Information 1: Error Location</p> <p>Attached Information 2: Error Location Details (Rung Number). For a program section, the rung number from the start of the section is given. For ST, the line number is given.</p> <p>Attached Information 3: Names of the Instruction and Instruction Instance Where the Error Occurred. If there is more than one possible instruction, information is given on all of them. Nothing is given if the instruction cannot be identified.</p> <p>Attached Information 4: Expansion Error Code (<i>ErrorIDEx</i>)</p>				
<b>Precautions/Remarks</b>	If a program is changed after an error occurs, the attached information that is displayed may not be correct.				

<b>Event name</b>	Direction Selection Out of Range		<b>Event code</b>	5401542E hex	
<b>Meaning</b>	The parameter specified for the <i>Direction</i> input variable to a motion control instruction is out of range.				
<b>Source</b>	PLC Function Module		<b>Source details</b>	Instruction	<b>Detection timing</b> At instruction execution
<b>Error attributes</b>	<b>Level</b>	Observation	<b>Recovery</b>	---	<b>Log category</b> System
<b>Effects</b>	<b>User program</b>	Continues	<b>Operation</b>	The relevant instruction will end according to specifications.	
<b>System-defined variables</b>	<b>Variable</b>		<b>Data type</b>		<b>Name</b>
	None		---		---
<b>Cause and correction</b>	<b>Assumed cause</b>		<b>Correction</b>		<b>Prevention</b>
	Instruction input parameter exceeded the valid range of the input variable.		Correct the parameter so that the valid range of the input variable is not exceeded for the relevant instruction.		Set the input parameter to the instruction so that the valid range of the input variable is not exceeded.
<b>Attached information</b>	<p>Attached Information 1: Error Location</p> <p>Attached Information 2: Error Location Details (Rung Number). For a program section, the rung number from the start of the section is given. For ST, the line number is given.</p> <p>Attached Information 3: Names of the Instruction and Instruction Instance Where the Error Occurred. If there is more than one possible instruction, information is given on all of them. Nothing is given if the instruction cannot be identified.</p> <p>Attached Information 4: Expansion Error Code (<i>ErrorIDEx</i>)</p>				
<b>Precautions/Remarks</b>	If a program is changed after an error occurs, the attached information that is displayed may not be correct.				

## 6 Troubleshooting

<b>Event name</b>	Transition Mode Selection Out of Range		<b>Event code</b>	54015432 hex	
<b>Meaning</b>	The parameter specified for the <i>TransitionMode</i> input variable to a motion control instruction is out of range.				
<b>Source</b>	PLC Function Module		<b>Source details</b>	Instruction	<b>Detection timing</b> At instruction execution
<b>Error attributes</b>	<b>Level</b>	Observation	<b>Recovery</b>	---	<b>Log category</b> System
<b>Effects</b>	<b>User program</b>	Continues	<b>Operation</b>	The relevant instruction will end according to specifications.	
<b>System-defined variables</b>	<b>Variable</b>	<b>Data type</b>		<b>Name</b>	
	None	---		---	
<b>Cause and correction</b>	<b>Assumed cause</b>	<b>Correction</b>		<b>Prevention</b>	
	Instruction input parameter exceeded the valid range of the input variable.	Correct the parameter so that the valid range of the input variable is not exceeded for the instruction.		Set the input parameter to the instruction so that the valid range of the input variable is not exceeded.	
	<i>_mcAborting</i> or <i>_mcBuffered</i> was specified for <i>BufferMode</i> and <i>_mcTMNone</i> was not specified for <i>TransitionMode</i> .	If you specify <i>_mcAborting</i> or <i>_mcBuffered</i> for <i>BufferMode</i> , specify <i>_mcTMNone</i> for <i>TransitionMode</i> . If you specify the values other than <i>_mcTMNone</i> for <i>TransitionMode</i> , specify <i>_mcBlendingLow</i> , <i>_mcBlendingPrevious</i> , <i>_mcBlendingNext</i> , or <i>_mcBlendingHigh</i> for <i>BufferMode</i> .		If you specify <i>_mcAborting</i> or <i>_mcBuffered</i> for <i>BufferMode</i> , specify <i>_mcTMNone</i> for <i>TransitionMode</i> . If you specify the values other than <i>_mcTMNone</i> for <i>TransitionMode</i> , specify <i>_mcBlendingLow</i> , <i>_mcBlendingPrevious</i> , <i>_mcBlendingNext</i> , or <i>_mcBlendingHigh</i> for <i>BufferMode</i> .	
<b>Attached information</b>	<p>Attached Information 1: Error Location</p> <p>Attached Information 2: Error Location Details (Rung Number). For a program section, the rung number from the start of the section is given. For ST, the line number is given.</p> <p>Attached Information 3: Names of the Instruction and Instruction Instance Where the Error Occurred. If there is more than one possible instruction, information is given on all of them. Nothing is given if the instruction cannot be identified.</p> <p>Attached Information 4: Expansion Error Code (<i>ErrorIDEx</i>)</p>				
<b>Precautions/Remarks</b>	If a program is changed after an error occurs, the attached information that is displayed may not be correct.				

<b>Event name</b>	Motion Control Instruction Re-execution Disabled		<b>Event code</b>	5401543B hex	
<b>Meaning</b>	An attempt was made to re-execute a motion control instruction that cannot be re-executed.				
<b>Source</b>	PLC Function Module		<b>Source details</b>	Instruction	<b>Detection timing</b> At instruction execution
<b>Error attributes</b>	<b>Level</b>	Observation	<b>Recovery</b>	---	<b>Log category</b> System
<b>Effects</b>	<b>User program</b>	Continues	<b>Operation</b>	The relevant instruction will end according to specifications.	
<b>System-defined variables</b>	<b>Variable</b>	<b>Data type</b>		<b>Name</b>	
	None	---		---	
<b>Cause and correction</b>	<b>Assumed cause</b>		<b>Correction</b>		<b>Prevention</b>
	A motion control instruction that cannot be re-executed was re-executed.		Correct the program so that the <i>Execute</i> input variable does not change to TRUE until the <i>Busy</i> output variable from the instruction changes to FALSE.		When you use instructions that cannot be re-executed, include a condition for the <i>Execute</i> input variable so that it does not change to TRUE unless the <i>Busy</i> output variable for the previous instruction is FALSE. Or, stop the instruction before executing it again.
<b>Attached information</b>	<p>Attached Information 1: Error Location</p> <p>Attached Information 2: Error Location Details (Rung Number). For a program section, the rung number from the start of the section is given. For ST, the line number is given.</p> <p>Attached Information 3: Names of the Instruction and Instruction Instance Where the Error Occurred. If there is more than one possible instruction, information is given on all of them. Nothing is given if the instruction cannot be identified.</p> <p>Attached Information 4: Expansion Error Code (<i>ErrorIDEx</i>)</p>				
<b>Precautions/Remarks</b>	If a program is changed after an error occurs, the attached information that is displayed may not be correct.				

## 6 Troubleshooting

<b>Event name</b>	Motion Control Instruction Multi-execution Disabled		<b>Event code</b>	5401543C hex	
<b>Meaning</b>	Multiple functions that cannot be executed simultaneously were executed for the same target (MC common, axis, or axes group).				
<b>Source</b>	PLC Function Module		<b>Source details</b>	Instruction	<b>Detection timing</b> At instruction execution
<b>Error attributes</b>	<b>Level</b>	Observation	<b>Recovery</b>	---	<b>Log category</b> System
<b>Effects</b>	<b>User program</b>	Continues	<b>Operation</b>	The relevant instruction will end according to specifications.	
<b>System-defined variables</b>	<b>Variable</b>		<b>Data type</b>		<b>Name</b>
	None		---		---
<b>Cause and correction</b>	<b>Assumed cause</b>		<b>Correction</b>		<b>Prevention</b>
	Multiple functions that cannot be executed simultaneously were executed for the same target (MC common, axis, or axes group).		Check the specifications of multi-execution of instructions for this instruction and do not execute instructions that cannot be executed at the same time.		Check the specifications for multi-execution of instructions for the instruction and do not execute instructions that cannot be executed at the same time.
<b>Attached information</b>	<p>Attached Information 1: Error Location</p> <p>Attached Information 2: Error Location Details (Rung Number). For a program section, the rung number from the start of the section is given. For ST, the line number is given.</p> <p>Attached Information 3: Names of the Instruction and Instruction Instance Where the Error Occurred. If there is more than one possible instruction, information is given on all of them. Nothing is given if the instruction cannot be identified.</p> <p>Attached Information 4: Expansion Error Code (<i>ErrorIDEx</i>)</p>				
<b>Precautions/Remarks</b>	If a program is changed after an error occurs, the attached information that is displayed may not be correct.				

<b>Event name</b>	Instruction Cannot Be Executed during Multi-axes Coordinated Control		<b>Event code</b>	5401543E hex			
<b>Meaning</b>	<ul style="list-style-type: none"> <li>A motion instruction was executed for an axis or an axes group that was in a coordinated multi-axes motion.</li> <li>A robot instruction that you cannot use for an axes group in a <i>GroupEnable</i> state was executed.</li> </ul>						
<b>Source</b>	PLC Function Module		<b>Source details</b>	Instruction	<b>Detection timing</b>	At instruction execution	
<b>Error attributes</b>	<b>Level</b>	Observation	<b>Recovery</b>	---	<b>Log category</b>	System	
<b>Effects</b>	<b>User program</b>	Continues	<b>Operation</b>	The relevant instruction will end according to specifications.			
<b>System-defined variables</b>	<b>Variable</b>		<b>Data type</b>		<b>Name</b>		
	None		---		---		
<b>Cause and correction</b>	<b>Assumed cause</b>		<b>Correction</b>		<b>Prevention</b>		
	A motion instruction was executed for an axis or an axes group that was in a coordinated multi-axes motion.		Correct the program so that axis operation instructions are executed only for axes or axes groups that are not in coordinated multi-axes motion.		Execute axis operation instructions only for axes or axes groups that are not in coordinated multi-axes motion.		
		The MC_SetKinTransform (Set Kinematics Transformation) instruction was executed for an axes group in a <i>GroupEnable</i> state.		Correct the program so that the instruction is executed only when the axes group is in a <i>GroupDisable</i> state.		Execute the instruction only when the axes group is in a <i>GroupDisable</i> state.	
<b>Attached information</b>	<p>Attached Information 1: Error Location</p> <p>Attached Information 2: Error Location Details (Rung Number). For a program section, the rung number from the start of the section is given. For ST, the line number is given.</p> <p>Attached Information 3: Names of the Instruction and Instruction Instance Where the Error Occurred. If there is more than one possible instruction, information is given on all of them. Nothing is given if the instruction cannot be identified.</p> <p>Attached Information 4: Expansion Error Code (<i>ErrorIDEx</i>)</p>						
<b>Precautions/Remarks</b>	If a program is changed after an error occurs, the attached information that is displayed may not be correct.						

## 6 Troubleshooting

<b>Event name</b>	Multi-axes Coordinated Control Instruction Executed for Disabled Axes Group		<b>Event code</b>	5401543F hex	
<b>Meaning</b>	A multi-axes coordinated control instruction was executed for an axes group that was in a <i>GroupDisable</i> state.				
<b>Source</b>	PLC Function Module		<b>Source details</b>	Instruction	<b>Detection timing</b> At instruction execution
<b>Error attributes</b>	<b>Level</b>	Observation	<b>Recovery</b>	---	<b>Log category</b> System
<b>Effects</b>	<b>User program</b>	Continues	<b>Operation</b>	The relevant instruction will end according to specifications.	
<b>System-defined variables</b>	<b>Variable</b>	<b>Data type</b>		<b>Name</b>	
	None	---		---	
<b>Cause and correction</b>	<b>Assumed cause</b>		<b>Correction</b>		<b>Prevention</b>
	<p>A multi-axes coordinated control instruction was executed for an axes group that was in a <i>GroupDisable</i> state.</p> <p>One of the following instructions was executed for an axes group that was in a <i>GroupDisable</i> state.</p> <ul style="list-style-type: none"> <li>• MC_MoveTimeAbsolute (Time-specified Absolute Positioning) instruction</li> <li>• MC_SyncLinearConveyor (Start Conveyor Synchronization) instruction</li> <li>• MC_SyncOut (End Synchronization) instruction</li> <li>• MC_RobotJog (Axes Group Jog) instruction</li> </ul>		<p>Correct the program so that the instruction is executed only after the axes group state changed to <i>GroupEnable</i>. Execute the MC_GroupEnable (Enable Axes Group) instruction to change the axes group state to <i>GroupEnable</i>.</p>		<p>Execute multi-axes coordinated control instructions only after enabling the axes group. Execute the MC_GroupEnable (Enable Axes Group) instruction to change the axes group state to <i>GroupEnable</i>.</p>
<b>Attached information</b>	<p>Attached Information 1: Error Location</p> <p>Attached Information 2: Error Location Details (Rung Number). For a program section, the rung number from the start of the section is given. For ST, the line number is given.</p> <p>Attached Information 3: Names of the Instruction and Instruction Instance Where the Error Occurred. If there is more than one possible instruction, information is given on all of them. Nothing is given if the instruction cannot be identified.</p> <p>Attached Information 4: Expansion Error Code (<i>ErrorIDEx</i>)</p>				
<b>Precautions/Remarks</b>	If a program is changed after an error occurs, the attached information that is displayed may not be correct.				

<b>Event name</b>	Impossible Axis Operation Specified when the Servo is OFF		<b>Event code</b>	54015441 hex	
<b>Meaning</b>	A motion instruction was executed for an axis for which the Servo is OFF.				
<b>Source</b>	PLC Function Module		<b>Source details</b>	Instruction	<b>Detection timing</b> At instruction execution
<b>Error attributes</b>	<b>Level</b>	Observation	<b>Recovery</b>	---	<b>Log category</b> System
<b>Effects</b>	<b>User program</b>	Continues	<b>Operation</b>	The relevant instruction will end according to specifications.	
<b>System-defined variables</b>	<b>Variable</b>	<b>Data type</b>		<b>Name</b>	
	None	---		---	
<b>Cause and correction</b>	<b>Assumed cause</b>		<b>Correction</b>		<b>Prevention</b>
	A motion instruction was executed for an axis for which the Servo is OFF.		Correct the program so that the instruction is executed after the Servo is turned ON.		Execute an axis motion instruction only after the Servo is turned ON.
	A zero position preset was performed with the MC_Home (Home) instruction or MC_HomeWithParameter (Home with Parameters) instruction for an axis for which EtherCAT process data communications are not established.		If the <code>_EC_PDSlavTbl</code> (Process Data Communicating Slave Table) system-defined variable for the EtherCAT master of the master axis is FALSE, remove its cause and execute the MC_Home or MC_HomeWithParameter instruction to preset home after <code>_EC_PD-SlavTbl</code> changes to TRUE.		Write the program to make sure that the <code>_EC_PDSlavTbl</code> (Process Data Communicating Slave Table) system-defined variable for the EtherCAT master is TRUE if you execute the MC_Home or MC_HomeWithParameter instruction to preset home immediately after you turn ON the power supply to the Controller, download data, reset a slave communications error, disconnect the slave, reconnect the slave, enable the slave, or disable the slave.
<b>Attached information</b>	<p>Attached Information 1: Error Location</p> <p>Attached Information 2: Error Location Details (Rung Number). For a program section, the rung number from the start of the section is given. For ST, the line number is given.</p> <p>Attached Information 3: Names of the Instruction and Instruction Instance Where the Error Occurred. If there is more than one possible instruction, information is given on all of them. Nothing is given if the instruction cannot be identified.</p> <p>Attached Information 4: Expansion Error Code (<i>ErrorIDEx</i>)</p>				
<b>Precautions/Remarks</b>	If a program is changed after an error occurs, the attached information that is displayed may not be correct.				

## 6 Troubleshooting

<b>Event name</b>	Composition Axis Stopped Error		<b>Event code</b>	54015442 hex	
<b>Meaning</b>	A motion instruction was executed for an axes group while the MC_Stop instruction was being executed for a composition axis.				
<b>Source</b>	PLC Function Module		<b>Source details</b>	Instruction	<b>Detection timing</b> At instruction execution
<b>Error attributes</b>	<b>Level</b>	Observation	<b>Recovery</b>	---	<b>Log category</b> System
<b>Effects</b>	<b>User program</b>	Continues	<b>Operation</b>	The relevant instruction will end according to specifications.	
<b>System-defined variables</b>	<b>Variable</b>		<b>Data type</b>		<b>Name</b>
	None		---		---
<b>Cause and correction</b>	<b>Assumed cause</b>		<b>Correction</b>		<b>Prevention</b>
	A motion instruction was executed for an axes group while the MC_Stop instruction was being executed for a composition axis.		Change the <i>Execute</i> input variable to the MC_Stop instruction for the composition axis to FALSE, reset the error, and then execute the motion control instruction.		Change the <i>Execute</i> input variables to the MC_Stop instructions for all of the composition axes to FALSE before you execute motion control instruction.
<b>Attached information</b>	<p>Attached Information 1: Error Location</p> <p>Attached Information 2: Error Location Details (Rung Number). For a program section, the rung number from the start of the section is given. For ST, the line number is given.</p> <p>Attached Information 3: Names of the Instruction and Instruction Instance Where the Error Occurred. If there is more than one possible instruction, information is given on all of them. Nothing is given if the instruction cannot be identified.</p> <p>Attached Information 4: Expansion Error Code (<i>ErrorIDEx</i>)</p>				
<b>Precautions/Remarks</b>	If a program is changed after an error occurs, the attached information that is displayed may not be correct.				

<b>Event name</b>	Motion Control Instruction Multi-execution Buffer Limit Exceeded		<b>Event code</b>	54015443 hex	
<b>Meaning</b>	The number of motion control instructions that is buffered for Buffered or Blending Buffer Modes exceeded the buffer limit.				
<b>Source</b>	PLC Function Module		<b>Source details</b>	Instruction	<b>Detection timing</b> At instruction execution
<b>Error attributes</b>	<b>Level</b>	Observation	<b>Recovery</b>	---	<b>Log category</b> System
<b>Effects</b>	<b>User program</b>	Continues	<b>Operation</b>	The relevant instruction will end according to specifications.	
<b>System-defined variables</b>	<b>Variable</b>	<b>Data type</b>		<b>Name</b>	
	None	---		---	
<b>Cause and correction</b>	<b>Assumed cause</b>		<b>Correction</b>		<b>Prevention</b>
	An axis instruction was executed when there was already a current instruction and a buffered instruction for the same axis.		Correct the program so that the number of executed instructions does not exceed the buffer limit.		Do not execute an axis instruction when there is already a current instruction and a buffered instruction for the same axis.
<b>Attached information</b>	An axes group instruction was executed when there was already eight current instructions and buffered instructions for the same axis.				Do not execute an axes group instruction when there are already eight current and buffered instructions for the same axis.
	Attached Information 1: Error Location Attached Information 2: Error Location Details (Rung Number). For a program section, the rung number from the start of the section is given. For ST, the line number is given. Attached Information 3: Names of the Instruction and Instruction Instance Where the Error Occurred. If there is more than one possible instruction, information is given on all of them. Nothing is given if the instruction cannot be identified. Attached Information 4: Expansion Error Code ( <i>ErrorIDEx</i> )				
<b>Precautions/Remarks</b>	If a program is changed after an error occurs, the attached information that is displayed may not be correct.				

## 6 Troubleshooting

<b>Event name</b>	Illegal Axes Group Specification		<b>Event code</b>	54015461 hex	
<b>Meaning</b>	The axes group specified for the <i>AxesGroup</i> input variable to a motion control instruction does not exist or is not a used axes group.				
<b>Source</b>	PLC Function Module		<b>Source details</b>	Instruction	<b>Detection timing</b> At instruction execution
<b>Error attributes</b>	<b>Level</b>	Observation	<b>Recovery</b>	---	<b>Log category</b> System
<b>Effects</b>	<b>User program</b>	Continues	<b>Operation</b>	The relevant instruction will end according to specifications.	
<b>System-defined variables</b>	<b>Variable</b>	<b>Data type</b>		<b>Name</b>	
	None	---		---	
<b>Cause and correction</b>	<b>Assumed cause</b>		<b>Correction</b>		<b>Prevention</b>
	An axes group does not exist for the variable specified for the <i>AxesGroup</i> input variable to the instruction.		Correct the specification for the instruction so that the specified axes group exists.		Specify a variable that exists when you specify a variable for an input parameter to an instruction.
	The axes group specified for the <i>AxesGroup</i> input variable to the instruction is not specified as a used axes group.		Set a used axes group for the axes group that is specified for the instruction.		Set a used axes group for the <i>AxesGroup</i> input variable to the instruction.
<b>Attached information</b>	<p>Attached Information 1: Error Location</p> <p>Attached Information 2: Error Location Details (Rung Number). For a program section, the rung number from the start of the section is given. For ST, the line number is given.</p> <p>Attached Information 3: Names of the Instruction and Instruction Instance Where the Error Occurred. If there is more than one possible instruction, information is given on all of them. Nothing is given if the instruction cannot be identified.</p> <p>Attached Information 4: Expansion Error Code (<i>ErrorIDEx</i>)</p>				
<b>Precautions/Remarks</b>	If a program is changed after an error occurs, the attached information that is displayed may not be correct.				

<b>Event name</b>	Instruction Execution Error with Undefined Home		<b>Event code</b>	54015466 hex	
<b>Meaning</b>	High-speed homing, an interpolation instruction, or a robot instruction was executed when home was undefined.				
<b>Source</b>	PLC Function Module		<b>Source details</b>	Instruction	<b>Detection timing</b> At instruction execution
<b>Error attributes</b>	<b>Level</b>	Observation	<b>Recovery</b>	---	<b>Log category</b> System
<b>Effects</b>	<b>User program</b>	Continues	<b>Operation</b>	The relevant instruction will end according to specifications.	
<b>System-defined variables</b>	<b>Variable</b>	<b>Data type</b>		<b>Name</b>	
	None	---		---	
<b>Cause and correction</b>	<b>Assumed cause</b>		<b>Correction</b>		<b>Prevention</b>
	High-speed homing was executed when home was undefined.		Execute the high-speed homing operation only after home is defined by homing.		Execute the high-speed homing instruction only after home is defined by homing.
	An interpolation instruction was executed for an axes group that includes an axis with no defined home. One of the following robot instructions was executed for an axes group that includes a logical axis with no defined home. <ul style="list-style-type: none"> <li>• MC_SetKinTransform (Set Kinematics Transformation) instruction</li> <li>• MC_MoveTimeAbsolute (Time-specified Absolute Positioning) instruction</li> <li>• MC_SyncLinearConveyor (Start Conveyor Synchronization) instruction</li> <li>• MC_SyncOut (End Synchronization) instruction</li> <li>• MC_GroupMon (Group Monitor) instruction</li> <li>• MC_RobotJog (Axes Group Jog) instruction</li> </ul>		Perform homing to define home for all axes in the axes group before you execute the interpolation instruction.		Perform homing to define home for all axes in the axes group before you execute an interpolation instruction.
<b>Attached information</b>	<p>Attached Information 1: Error Location</p> <p>Attached Information 2: Error Location Details (Rung Number). For a program section, the rung number from the start of the section is given. For ST, the line number is given.</p> <p>Attached Information 3: Names of the Instruction and Instruction Instance Where the Error Occurred. If there is more than one possible instruction, information is given on all of them. Nothing is given if the instruction cannot be identified.</p> <p>Attached Information 4: Expansion Error Code (<i>ErrorIDEx</i>)</p>				
<b>Precautions/Remarks</b>	If a program is changed after an error occurs, the attached information that is displayed may not be correct.				

## 6 Troubleshooting

<b>Event name</b>	Target Position Setting Out of Range		<b>Event code</b>	54015478 hex	
<b>Meaning</b>	The parameter specified for the <i>Position</i> input variable to a motion control instruction is out of range.				
<b>Source</b>	PLC Function Module		<b>Source details</b>	Instruction	<b>Detection timing</b> At instruction execution
<b>Error attributes</b>	<b>Level</b>	Observation	<b>Recovery</b>	---	<b>Log category</b> System
<b>Effects</b>	<b>User program</b>	Continues	<b>Operation</b>	The relevant instruction will end according to specifications.	
<b>System-defined variables</b>	<b>Variable</b>	<b>Data type</b>		<b>Name</b>	
	None	---		---	
<b>Cause and correction</b>	<b>Assumed cause</b>		<b>Correction</b>		<b>Prevention</b>
	Instruction input parameter exceeded the valid range of the input variable.		Correct the parameter so that the valid range of the input variable is not exceeded for the relevant instruction.		Set the input parameter to the instruction so that the valid range of the input variable is not exceeded.
	The target position of a Rotary Mode axis is not within the ring setting range.		Correct the target position of the Rotary Mode axis to within the ring setting range.		Set the target position of the Rotary Mode axis to within the ring setting range.
<b>Attached information</b>	<p>Attached Information 1: Error Location</p> <p>Attached Information 2: Error Location Details (Rung Number). For a program section, the rung number from the start of the section is given. For ST, the line number is given.</p> <p>Attached Information 3: Names of the Instruction and Instruction Instance Where the Error Occurred. If there is more than one possible instruction, information is given on all of them. Nothing is given if the instruction cannot be identified.</p> <p>Attached Information 4: Expansion Error Code (<i>ErrorIDEx</i>)</p>				
<b>Precautions/Remarks</b>	If a program is changed after an error occurs, the attached information that is displayed may not be correct.				

<b>Event name</b>	Axes Group Mismatch with Kinematics		<b>Event code</b>	54015706 hex	
<b>Meaning</b>	The configuration elements of the specified axes group and the specified kinematics do not match.				
<b>Source</b>	PLC Function Module		<b>Source details</b>	Instruction	<b>Detection timing</b> At instruction execution
<b>Error attributes</b>	<b>Level</b>	Observation	<b>Recovery</b>	---	<b>Log category</b> System
<b>Effects</b>	<b>User program</b>	Continues	<b>Operation</b>	The relevant instruction will end according to specifications.	
<b>System-defined variables</b>	<b>Variable</b>		<b>Data type</b>		<b>Name</b>
	None		---		---
<b>Cause and correction</b>	<b>Assumed cause</b>		<b>Correction</b>		<b>Prevention</b>
	The number of axes in the axes group and the number of axes in the specified robot (kinematics type) do not match.		Correct the setting so that the number of axes in the axes group and the number of axes in the specified robot (kinematics type) match.		Make sure that the number of axes in the axes group and the number of axes in the specified robot (kinematics type) match.
	The count mode for axes in the axes group and the count mode for the specified robot (kinematics type) do not match.		Correct the count mode for axes in the axes group so that it can match the count mode for the robot (kinematics type).		Make sure that the count mode for axes in the axes group is the same as the count mode specified for the robot (kinematics type).
<b>Attached information</b>	The display unit for axes in the axes group and the display unit for the specified robot (kinematics type) do not match.		Correct the display unit for axes in the axes group so that it can match the display unit for the robot (kinematics type).		Make sure that the display unit for axes in the axes group is the same as the display unit specified for the robot (kinematics type).
	Attached Information 1: Error Location Attached Information 2: Error Location Details (Rung Number). For a program section, the rung number from the start of the section is given. For ST, the line number is given. Attached Information 3: Names of the Instruction and Instruction Instance Where the Error Occurred. If there is more than one possible instruction, information is given on all of them. Nothing is given if the instruction cannot be identified. Attached Information 4: Expansion Error Code ( <i>ErrorIDEx</i> )				
<b>Precautions/Remarks</b>	If a program is changed after an error occurs, the attached information that is displayed may not be correct.				

## 6 Troubleshooting

<b>Event name</b>	Kinematics Type Out of Range		<b>Event code</b>	54015707 hex	
<b>Meaning</b>	<i>KinType</i> specified for the <i>KinTransform</i> input variable to a motion control instruction is out of range.				
<b>Source</b>	PLC Function Module		<b>Source details</b>	Instruction	<b>Detection timing</b> At instruction execution
<b>Error attributes</b>	<b>Level</b>	Observation	<b>Recovery</b>	---	<b>Log category</b> System
<b>Effects</b>	<b>User program</b>	Continues	<b>Operation</b>	The relevant instruction will end according to specifications.	
<b>System-defined variables</b>	<b>Variable</b>	<b>Data type</b>		<b>Name</b>	
	None	---		---	
<b>Cause and correction</b>	<b>Assumed cause</b>		<b>Correction</b>		<b>Prevention</b>
	<i>KinType</i> is outside the setting range.		Correct <i>KinType</i> of the <i>KinTransform</i> input variable.		Set the kinematics type correctly for the <i>KinTransform</i> input variable.
<b>Attached information</b>	<p>Attached Information 1: Error Location</p> <p>Attached Information 2: Error Location Details (Rung Number). For a program section, the rung number from the start of the section is given. For ST, the line number is given.</p> <p>Attached Information 3: Names of the Instruction and Instruction Instance Where the Error Occurred. If there is more than one possible instruction, information is given on all of them. Nothing is given if the instruction cannot be identified.</p> <p>Attached Information 4: Expansion Error Code (<i>ErrorIDEx</i>)</p>				
<b>Precautions/Remarks</b>	If a program is changed after an error occurs, the attached information that is displayed may not be correct.				

<b>Event name</b>	Kinematics Parameter Out of Range		<b>Event code</b>	54015708 hex	
<b>Meaning</b>	<i>KinParam</i> or <i>ExpansionParam</i> specified for the <i>KinTransform</i> input variable to a motion control instruction is out of range.				
<b>Source</b>	PLC Function Module		<b>Source details</b>	Instruction	<b>Detection timing</b> At instruction execution
<b>Error attributes</b>	<b>Level</b>	Observation	<b>Recovery</b>	---	<b>Log category</b> System
<b>Effects</b>	<b>User program</b>	Continues	<b>Operation</b>	The relevant instruction will end according to specifications.	
<b>System-defined variables</b>	<b>Variable</b>	<b>Data type</b>		<b>Name</b>	
	None	---		---	
<b>Cause and correction</b>	<b>Assumed cause</b>		<b>Correction</b>		<b>Prevention</b>
	<i>KinParam</i> is outside the setting range.		Correct <i>KinParam</i> or <i>ExpansionParam</i> of the <i>KinTransform</i> input variable.		Set the correct kinematics parameter and expansion parameter for the <i>KinTransform</i> input variable.
	<i>ExpansionParam</i> is outside the setting range.				
<b>Attached information</b>	<p>Attached Information 1: Error Location</p> <p>Attached Information 2: Error Location Details (Rung Number). For a program section, the rung number from the start of the section is given. For ST, the line number is given.</p> <p>Attached Information 3: Names of the Instruction and Instruction Instance Where the Error Occurred. If there is more than one possible instruction, information is given on all of them. Nothing is given if the instruction cannot be identified.</p> <p>Attached Information 4: Expansion Error Code (<i>ErrorIDEx</i>)</p>				
<b>Precautions/Remarks</b>	If a program is changed after an error occurs, the attached information that is displayed may not be correct.				

<b>Event name</b>	Workspace Type Out of Range		<b>Event code</b>	54015709 hex	
<b>Meaning</b>	<i>WorkspaceType</i> specified for the <i>Workspace</i> input variable to a motion control instruction is out of range.				
<b>Source</b>	PLC Function Module		<b>Source details</b>	Instruction	<b>Detection timing</b> At instruction execution
<b>Error attributes</b>	<b>Level</b>	Observation	<b>Recovery</b>	---	<b>Log category</b> System
<b>Effects</b>	<b>User program</b>	Continues	<b>Operation</b>	The relevant instruction will end according to specifications.	
<b>System-defined variables</b>	<b>Variable</b>	<b>Data type</b>		<b>Name</b>	
	None	---		---	
<b>Cause and correction</b>	<b>Assumed cause</b>		<b>Correction</b>		<b>Prevention</b>
	<i>WorkspaceType</i> is outside the setting range.		Correct <i>WorkspaceType</i> of the <i>Workspace</i> input variable.		Set the correct workspace type for the <i>Workspace</i> input variable.
<b>Attached information</b>	<p>Attached Information 1: Error Location</p> <p>Attached Information 2: Error Location Details (Rung Number). For a program section, the rung number from the start of the section is given. For ST, the line number is given.</p> <p>Attached Information 3: Names of the Instruction and Instruction Instance Where the Error Occurred. If there is more than one possible instruction, information is given on all of them. Nothing is given if the instruction cannot be identified.</p> <p>Attached Information 4: Expansion Error Code (<i>ErrorIDEx</i>)</p>				
<b>Precautions/Remarks</b>	If a program is changed after an error occurs, the attached information that is displayed may not be correct.				

<b>Event name</b>	Workspace Parameter Out of Range		<b>Event code</b>	5401570A hex	
<b>Meaning</b>	<i>WorkspaceParam</i> specified for the <i>Workspace</i> input variable to a motion control instruction is out of range.				
<b>Source</b>	PLC Function Module		<b>Source details</b>	Instruction	<b>Detection timing</b> At instruction execution
<b>Error attributes</b>	<b>Level</b>	Observation	<b>Recovery</b>	---	<b>Log category</b> System
<b>Effects</b>	<b>User program</b>	Continues	<b>Operation</b>	The relevant instruction will end according to specifications.	
<b>System-defined variables</b>	<b>Variable</b>	<b>Data type</b>		<b>Name</b>	
	None	---		---	
<b>Cause and correction</b>	<b>Assumed cause</b>		<b>Correction</b>		<b>Prevention</b>
	<i>WorkspaceParam</i> is outside the setting range.		Correct <i>WorkspaceParam</i> of the <i>Workspace</i> input variable.		Set the correct workspace parameter for the <i>Workspace</i> input variable.
<b>Attached information</b>	<p>Attached Information 1: Error Location</p> <p>Attached Information 2: Error Location Details (Rung Number). For a program section, the rung number from the start of the section is given. For ST, the line number is given.</p> <p>Attached Information 3: Names of the Instruction and Instruction Instance Where the Error Occurred. If there is more than one possible instruction, information is given on all of them. Nothing is given if the instruction cannot be identified.</p> <p>Attached Information 4: Expansion Error Code (<i>ErrorIDEx</i>)</p>				
<b>Precautions/Remarks</b>	If a program is changed after an error occurs, the attached information that is displayed may not be correct.				

## 6 Troubleshooting

<b>Event name</b>	Invalid Coordinate System Number		<b>Event code</b>	5401570B hex	
<b>Meaning</b>	The coordinate system ID specified for the <i>CSID</i> input variable to a motion control instruction is out of range or not defined.				
<b>Source</b>	PLC Function Module		<b>Source details</b>	Instruction	<b>Detection timing</b> At instruction execution
<b>Error attributes</b>	<b>Level</b>	Observation	<b>Recovery</b>	---	<b>Log category</b> System
<b>Effects</b>	<b>User program</b>	Continues	<b>Operation</b>	The relevant instruction will end according to specifications.	
<b>System-defined variables</b>	<b>Variable</b>	<b>Data type</b>		<b>Name</b>	
	None	---		---	
<b>Cause and correction</b>	<b>Assumed cause</b>		<b>Correction</b>		<b>Prevention</b>
	The coordinate system ID is outside the setting range.		Change the value of the <i>CSID</i> input variable to a correct coordinate system ID.		Set the coordinate system ID correctly for the <i>CSID</i> input variable.
	The specified coordinate system ID is not defined by the MC_DefineCoordSystem (Define Coordinate) instruction.		Define the coordinate system with the MC_DefineCoordSystem (Define Coordinate) instruction.		Define the coordinate system and use it.
<b>Attached information</b>	<p>Attached Information 1: Error Location</p> <p>Attached Information 2: Error Location Details (Rung Number). For a program section, the rung number from the start of the section is given. For ST, the line number is given.</p> <p>Attached Information 3: Names of the Instruction and Instruction Instance Where the Error Occurred. If there is more than one possible instruction, information is given on all of them. Nothing is given if the instruction cannot be identified.</p> <p>Attached Information 4: Expansion Error Code (<i>ErrorIDEx</i>)</p>				
<b>Precautions/Remarks</b>	If a program is changed after an error occurs, the attached information that is displayed may not be correct.				

<b>Event name</b>	Coordinate Transformation Parameter Out of Range		<b>Event code</b>	5401570C hex	
<b>Meaning</b>	The coordinate transformation parameter <i>Pose</i> , which was specified for the <i>CoordTransform</i> input variable to a motion control instruction, is out of range.				
<b>Source</b>	PLC Function Module		<b>Source details</b>	Instruction	<b>Detection timing</b> At instruction execution
<b>Error attributes</b>	<b>Level</b>	Observation	<b>Recovery</b>	---	<b>Log category</b> System
<b>Effects</b>	<b>User program</b>	Continues	<b>Operation</b>	The relevant instruction will end according to specifications.	
<b>System-defined variables</b>	<b>Variable</b>		<b>Data type</b>		<b>Name</b>
	None		---		---
<b>Cause and correction</b>	<b>Assumed cause</b>		<b>Correction</b>		<b>Prevention</b>
	The coordinate transformation parameter <i>Pose</i> is outside the setting range.		Correct the coordinate transformation parameter <i>Pose</i> of the <i>CoordTransform</i> input variable.		Set the coordinate transformation parameter <i>Pose</i> correctly for the <i>CoordTransform</i> input variable.
<b>Attached information</b>	<p>Attached Information 1: Error Location</p> <p>Attached Information 2: Error Location Details (Rung Number). For a program section, the rung number from the start of the section is given. For ST, the line number is given.</p> <p>Attached Information 3: Names of the Instruction and Instruction Instance Where the Error Occurred. If there is more than one possible instruction, information is given on all of them. Nothing is given if the instruction cannot be identified.</p> <p>Attached Information 4: Expansion Error Code (<i>ErrorIDEx</i>)</p>				
<b>Precautions/Remarks</b>	If a program is changed after an error occurs, the attached information that is displayed may not be correct.				

<b>Event name</b>	Transition parameters out of range		<b>Event code</b>	5401570D hex	
<b>Meaning</b>	The Transition Parameters specified for the <i>TransitionParameter</i> input variable to a motion control instruction is out of range.				
<b>Source</b>	PLC Function Module		<b>Source details</b>	Instruction	<b>Detection timing</b> At instruction execution
<b>Error attributes</b>	<b>Level</b>	Observation	<b>Recovery</b>	---	<b>Log category</b> System
<b>Effects</b>	<b>User program</b>	Continues	<b>Operation</b>	The relevant instruction will end according to specifications.	
<b>System-defined variables</b>	<b>Variable</b>		<b>Data type</b>		<b>Name</b>
	None		---		---
<b>Cause and correction</b>	<b>Assumed cause</b>		<b>Correction</b>		<b>Prevention</b>
	<i>TransitionParameter</i> is outside the setting range.		Correct the parameters of the <i>TransitionParameter</i> input variable.		Set the correct parameters for the <i>TransitionParameter</i> input variable.
<b>Attached information</b>	<p>Attached Information 1: Error Location</p> <p>Attached Information 2: Error Location Details (Rung Number). For a program section, the rung number from the start of the section is given. For ST, the line number is given.</p> <p>Attached Information 3: Names of the Instruction and Instruction Instance Where the Error Occurred. If there is more than one possible instruction, information is given on all of them. Nothing is given if the instruction cannot be identified.</p> <p>Attached Information 4: Expansion Error Code (<i>ErrorIDEx</i>)</p>				
<b>Precautions/Remarks</b>	If a program is changed after an error occurs, the attached information that is displayed may not be correct.				

## 6 Troubleshooting

<b>Event name</b>	Kinematics Transform Not Set		<b>Event code</b>	54015710 hex	
<b>Meaning</b>	The kinematics transform is not set for the specified axes group.				
<b>Source</b>	PLC Function Module		<b>Source details</b>	Instruction	<b>Detection timing</b> At instruction execution
<b>Error attributes</b>	<b>Level</b>	Observation	<b>Recovery</b>	---	<b>Log category</b> System
<b>Effects</b>	<b>User program</b>	Continues	<b>Operation</b>	The relevant instruction will end according to specifications.	
<b>System-defined variables</b>	<b>Variable</b>	<b>Data type</b>		<b>Name</b>	
	None	---		---	
<b>Cause and correction</b>	<b>Assumed cause</b>		<b>Correction</b>		<b>Prevention</b>
	The kinematics transform is not set for the axes group.		Use the MC_SetKinTransform (Set Kinematics Transformation) instruction to set the kinematics transform for the axes group.		Execute the instruction after you set the kinematics transform for the axes group.
<b>Attached information</b>	<p>Attached Information 1: Error Location</p> <p>Attached Information 2: Error Location Details (Rung Number). For a program section, the rung number from the start of the section is given. For ST, the line number is given.</p> <p>Attached Information 3: Names of the Instruction and Instruction Instance Where the Error Occurred. If there is more than one possible instruction, information is given on all of them. Nothing is given if the instruction cannot be identified.</p> <p>Attached Information 4: Expansion Error Code (<i>ErrorIDEx</i>)</p>				
<b>Precautions/Remarks</b>	If a program is changed after an error occurs, the attached information that is displayed may not be correct.				

<b>Event name</b>	Target Position Out of Range		<b>Event code</b>	54015711 hex	
<b>Meaning</b>	The position parameter specified as variable for the <i>Position</i> is out of range				
<b>Source</b>	PLC Function Module		<b>Source details</b>	Instruction	<b>Detection timing</b> At instruction execution
<b>Error attributes</b>	<b>Level</b>	Observation	<b>Recovery</b>	---	<b>Log category</b> System
<b>Effects</b>	<b>User program</b>	Continues	<b>Operation</b>	The relevant instruction will end according to specifications.	
<b>System-defined variables</b>	<b>Variable</b>	<b>Data type</b>		<b>Name</b>	
	None	---		---	
<b>Cause and correction</b>	<b>Assumed cause</b>		<b>Correction</b>		<b>Prevention</b>
	<i>Positoin</i> is outside the setting range.		Correct the target position inside maximum movement volume of the robot.		Set the target position inside maximum movement volume of the robot.
<b>Attached information</b>	<p>Attached Information 1: Error Location</p> <p>Attached Information 2: Error Location Details (Rung Number). For a program section, the rung number from the start of the section is given. For ST, the line number is given.</p> <p>Attached Information 3: Names of the Instruction and Instruction Instance Where the Error Occurred. If there is more than one possible instruction, information is given on all of them. Nothing is given if the instruction cannot be identified.</p> <p>Attached Information 4: Expansion Error Code (<i>ErrorIDEx</i>)</p>				
<b>Precautions/Remarks</b>	If a program is changed after an error occurs, the attached information that is displayed may not be correct.				

<b>Event name</b>	Velocity Error Detection Value Out of Range		<b>Event code</b>	54015712 hex	
<b>Meaning</b>	<i>MaxVelocity</i> specified for the <i>TrajData</i> input variable to a motion control instruction is out of range.				
<b>Source</b>	PLC Function Module		<b>Source details</b>	Instruction	<b>Detection timing</b> At instruction execution
<b>Error attributes</b>	<b>Level</b>	Observation	<b>Recovery</b>	---	<b>Log category</b> System
<b>Effects</b>	<b>User program</b>	Continues	<b>Operation</b>	The relevant instruction will end according to specifications.	
<b>System-defined variables</b>	<b>Variable</b>	<b>Data type</b>		<b>Name</b>	
	None	---		---	
<b>Cause and correction</b>	<b>Assumed cause</b>		<b>Correction</b>		<b>Prevention</b>
	<i>MaxVelocity</i> is outside the setting range.		Correct the value of <i>MaxVelocity</i> of the <i>TrajData</i> input variable.		Set <i>MaxVelocity</i> for the <i>TrajData</i> input variable correctly.
<b>Attached information</b>	<p>Attached Information 1: Error Location</p> <p>Attached Information 2: Error Location Details (Rung Number). For a program section, the rung number from the start of the section is given. For ST, the line number is given.</p> <p>Attached Information 3: Names of the Instruction and Instruction Instance Where the Error Occurred. If there is more than one possible instruction, information is given on all of them. Nothing is given if the instruction cannot be identified.</p> <p>Attached Information 4: Expansion Error Code (<i>ErrorIDEx</i>)</p>				
<b>Precautions/Remarks</b>	If a program is changed after an error occurs, the attached information that is displayed may not be correct.				

<b>Event name</b>	Acceleration Error Detection Value Out of Range		<b>Event code</b>	54015713 hex	
<b>Meaning</b>	<i>MaxAcceleration</i> specified for the <i>TrajData</i> input variable to a motion control instruction is out of range.				
<b>Source</b>	PLC Function Module		<b>Source details</b>	Instruction	<b>Detection timing</b> At instruction execution
<b>Error attributes</b>	<b>Level</b>	Observation	<b>Recovery</b>	---	<b>Log category</b> System
<b>Effects</b>	<b>User program</b>	Continues	<b>Operation</b>	The relevant instruction will end according to specifications.	
<b>System-defined variables</b>	<b>Variable</b>	<b>Data type</b>		<b>Name</b>	
	None	---		---	
<b>Cause and correction</b>	<b>Assumed cause</b>		<b>Correction</b>		<b>Prevention</b>
	<i>MaxAcceleration</i> is outside the setting range.		Correct the value of <i>MaxAcceleration</i> of the <i>TrajData</i> input variable.		Set <i>MaxAcceleration</i> correctly for the <i>TrajData</i> input variable.
<b>Attached information</b>	<p>Attached Information 1: Error Location</p> <p>Attached Information 2: Error Location Details (Rung Number). For a program section, the rung number from the start of the section is given. For ST, the line number is given.</p> <p>Attached Information 3: Names of the Instruction and Instruction Instance Where the Error Occurred. If there is more than one possible instruction, information is given on all of them. Nothing is given if the instruction cannot be identified.</p> <p>Attached Information 4: Expansion Error Code (<i>ErrorIDEx</i>)</p>				
<b>Precautions/Remarks</b>	If a program is changed after an error occurs, the attached information that is displayed may not be correct.				

## 6 Troubleshooting

<b>Event name</b>	Trajectory Target Time Out of Range		<b>Event code</b>	54015714 hex	
<b>Meaning</b>	<i>TrajTime</i> specified for the <i>TrajData</i> input variable to a motion control instruction is out of range.				
<b>Source</b>	PLC Function Module		<b>Source details</b>	Instruction	<b>Detection timing</b> At instruction execution
<b>Error attributes</b>	<b>Level</b>	Observation	<b>Recovery</b>	---	<b>Log category</b> System
<b>Effects</b>	<b>User program</b>	Continues	<b>Operation</b>	The relevant instruction will end according to specifications.	
<b>System-defined variables</b>	<b>Variable</b>	<b>Data type</b>		<b>Name</b>	
	None	---		---	
<b>Cause and correction</b>	<b>Assumed cause</b>		<b>Correction</b>		<b>Prevention</b>
	<i>TrajTime</i> is outside the setting range.		Correct the value of <i>TrajTime</i> of the <i>TrajData</i> input variable.		Set <i>TrajTime</i> correctly for the <i>TrajData</i> input variable.
<b>Attached information</b>	<p>Attached Information 1: Error Location</p> <p>Attached Information 2: Error Location Details (Rung Number). For a program section, the rung number from the start of the section is given. For ST, the line number is given.</p> <p>Attached Information 3: Names of the Instruction and Instruction Instance Where the Error Occurred. If there is more than one possible instruction, information is given on all of them. Nothing is given if the instruction cannot be identified.</p> <p>Attached Information 4: Expansion Error Code (<i>ErrorIDEx</i>)</p>				
<b>Precautions/Remarks</b>	If a program is changed after an error occurs, the attached information that is displayed may not be correct.				

<b>Event name</b>	Trajectory Type Out of Range		<b>Event code</b>	54015715 hex	
<b>Meaning</b>	<i>MoveTrajType</i> or <i>SyncTrajType</i> specified for the <i>TrajData</i> input variable to a motion control instruction is out of range.				
<b>Source</b>	PLC Function Module		<b>Source details</b>	Instruction	<b>Detection timing</b> At instruction execution
<b>Error attributes</b>	<b>Level</b>	Observation	<b>Recovery</b>	---	<b>Log category</b> System
<b>Effects</b>	<b>User program</b>	Continues	<b>Operation</b>	The relevant instruction will end according to specifications.	
<b>System-defined variables</b>	<b>Variable</b>	<b>Data type</b>		<b>Name</b>	
	None	---		---	
<b>Cause and correction</b>	<b>Assumed cause</b>		<b>Correction</b>		<b>Prevention</b>
	<i>MoveTrajType</i> or <i>SyncTrajType</i> is outside the setting range.		Correct the value of <i>MoveTrajType</i> or <i>SyncTrajType</i> of the <i>TrajData</i> input variable.		Set <i>MoveTrajType</i> or <i>SyncTrajType</i> correctly for the <i>TrajData</i> input variable.
<b>Attached information</b>	<p>Attached Information 1: Error Location</p> <p>Attached Information 2: Error Location Details (Rung Number). For a program section, the rung number from the start of the section is given. For ST, the line number is given.</p> <p>Attached Information 3: Names of the Instruction and Instruction Instance Where the Error Occurred. If there is more than one possible instruction, information is given on all of them. Nothing is given if the instruction cannot be identified.</p> <p>Attached Information 4: Expansion Error Code (<i>ErrorIDEx</i>)</p>				
<b>Precautions/Remarks</b>	If a program is changed after an error occurs, the attached information that is displayed may not be correct.				

<b>Event name</b>	Trajectory Transition Out of Range		<b>Event code</b>	54015716 hex	
<b>Meaning</b>	<i>TrajTransition</i> specified for the <i>TrajData</i> input variable to a motion control instruction is out of range.				
<b>Source</b>	PLC Function Module		<b>Source details</b>	Instruction	<b>Detection timing</b> At instruction execution
<b>Error attributes</b>	<b>Level</b>	Observation	<b>Recovery</b>	---	<b>Log category</b> System
<b>Effects</b>	<b>User program</b>	Continues	<b>Operation</b>	The relevant instruction will end according to specifications.	
<b>System-defined variables</b>	<b>Variable</b>	<b>Data type</b>		<b>Name</b>	
	None	---		---	
<b>Cause and correction</b>	<b>Assumed cause</b>		<b>Correction</b>		<b>Prevention</b>
	<i>TrajTransition</i> is outside the setting range.		Correct the value of <i>TrajTransition</i> of the <i>TrajData</i> input variable.		Set the trajectory transition correctly.
<b>Attached information</b>	<p>Attached Information 1: Error Location</p> <p>Attached Information 2: Error Location Details (Rung Number). For a program section, the rung number from the start of the section is given. For ST, the line number is given.</p> <p>Attached Information 3: Names of the Instruction and Instruction Instance Where the Error Occurred. If there is more than one possible instruction, information is given on all of them. Nothing is given if the instruction cannot be identified.</p> <p>Attached Information 4: Expansion Error Code (<i>ErrorIDEx</i>)</p>				
<b>Precautions/Remarks</b>	If a program is changed after an error occurs, the attached information that is displayed may not be correct.				

<b>Event name</b>	Trajectory Travel Distance Out of Range		<b>Event code</b>	54015717 hex	
<b>Meaning</b>	<i>TrajDistance</i> specified for the <i>TrajData</i> input variable to a motion control instruction is out of range.				
<b>Source</b>	PLC Function Module		<b>Source details</b>	Instruction	<b>Detection timing</b> At instruction execution
<b>Error attributes</b>	<b>Level</b>	Observation	<b>Recovery</b>	---	<b>Log category</b> System
<b>Effects</b>	<b>User program</b>	Continues	<b>Operation</b>	The relevant instruction will end according to specifications.	
<b>System-defined variables</b>	<b>Variable</b>	<b>Data type</b>		<b>Name</b>	
	None	---		---	
<b>Cause and correction</b>	<b>Assumed cause</b>		<b>Correction</b>		<b>Prevention</b>
	<i>TrajDistance</i> is outside the setting range.		Correct the value of <i>TrajDistance</i> of the <i>TrajData</i> input variable.		Set the trajectory travel distance correctly.
<b>Attached information</b>	<p>Attached Information 1: Error Location</p> <p>Attached Information 2: Error Location Details (Rung Number). For a program section, the rung number from the start of the section is given. For ST, the line number is given.</p> <p>Attached Information 3: Names of the Instruction and Instruction Instance Where the Error Occurred. If there is more than one possible instruction, information is given on all of them. Nothing is given if the instruction cannot be identified.</p> <p>Attached Information 4: Expansion Error Code (<i>ErrorIDEx</i>)</p>				
<b>Precautions/Remarks</b>	If a program is changed after an error occurs, the attached information that is displayed may not be correct.				

## 6 Troubleshooting

<b>Event name</b>	Initial Workpiece Position Outside Workspace		<b>Event code</b>	54015719 hex	
<b>Meaning</b>	The position of the workpiece specified for the <i>InitWorkpiecePosition</i> input variable to a motion control instruction is outside the workspace.				
<b>Source</b>	PLC Function Module		<b>Source details</b>	Instruction	<b>Detection timing</b> At instruction execution
<b>Error attributes</b>	<b>Level</b>	Observation	<b>Recovery</b>	---	<b>Log category</b> System
<b>Effects</b>	<b>User program</b>	Continues	<b>Operation</b>	The relevant instruction will end according to specifications.	
<b>System-defined variables</b>	<b>Variable</b>	<b>Data type</b>		<b>Name</b>	
	None	---		---	
<b>Cause and correction</b>	<b>Assumed cause</b>		<b>Correction</b>		<b>Prevention</b>
	<i>InitWorkpiecePosition</i> is outside the workspace.		Move the workpiece inside the workspace.		Do not allow the initial position of the workpiece to exceed the range of the workspace.
<b>Attached information</b>	<p>Attached Information 1: Error Location</p> <p>Attached Information 2: Error Location Details (Rung Number). For a program section, the rung number from the start of the section is given. For ST, the line number is given.</p> <p>Attached Information 3: Names of the Instruction and Instruction Instance Where the Error Occurred. If there is more than one possible instruction, information is given on all of them. Nothing is given if the instruction cannot be identified.</p> <p>Attached Information 4: Expansion Error Code (<i>ErrorIDEx</i>)</p>				
<b>Precautions/Remarks</b>	If a program is changed after an error occurs, the attached information that is displayed may not be correct.				

<b>Event name</b>	Invalid Conveyor Axis Specified		<b>Event code</b>	5401571A hex	
<b>Meaning</b>	The axis specified for the <i>ConveyorAxis</i> in-out variable to a motion control instruction is not correct.				
<b>Source</b>	PLC Function Module		<b>Source details</b>	Instruction	<b>Detection timing</b> At instruction execution
<b>Error attributes</b>	<b>Level</b>	Observation	<b>Recovery</b>	---	<b>Log category</b> System
<b>Effects</b>	<b>User program</b>	Continues	<b>Operation</b>	The relevant instruction will end according to specifications.	
<b>System-defined variables</b>	<b>Variable</b>	<b>Data type</b>		<b>Name</b>	
	None	---		---	
<b>Cause and correction</b>	<b>Assumed cause</b>		<b>Correction</b>		<b>Prevention</b>
	The specified axis is registered in the axes group which is specified for <i>AxesGroup</i> .		Correct the axis setting.		Set the conveyor axis correctly.
	The specified axis is an unused axis.		Change the axis setting to Used Axis.		Set Used Axis for the axis.
	The unit of the conveyor axis is incorrect.		Change the unit of axis to mm.		Set the unit of axis to mm.
<b>Attached information</b>	<p>Attached Information 1: Error Location</p> <p>Attached Information 2: Error Location Details (Rung Number). For a program section, the rung number from the start of the section is given. For ST, the line number is given.</p> <p>Attached Information 3: Names of the Instruction and Instruction Instance Where the Error Occurred. If there is more than one possible instruction, information is given on all of them. Nothing is given if the instruction cannot be identified.</p> <p>Attached Information 4: Expansion Error Code (<i>ErrorIDEx</i>)</p>				
<b>Precautions/Remarks</b>	If a program is changed after an error occurs, the attached information that is displayed may not be correct.				

<b>Event name</b>	Target Position Outside Workspace		<b>Event code</b>	5401571B hex	
<b>Meaning</b>	The target position specified for the <i>Position</i> input variable to a motion control instruction is outside the workspace.				
<b>Source</b>	PLC Function Module		<b>Source details</b>	Instruction	<b>Detection timing</b> At instruction execution
<b>Error attributes</b>	<b>Level</b>	Observation	<b>Recovery</b>	---	<b>Log category</b> System
<b>Effects</b>	<b>User program</b>	Continues	<b>Operation</b>	The relevant instruction will end according to specifications.	
<b>System-defined variables</b>	<b>Variable</b>	<b>Data type</b>		<b>Name</b>	
	None	---		---	
<b>Cause and correction</b>	<b>Assumed cause</b>		<b>Correction</b>		<b>Prevention</b>
	<i>Position</i> is outside the workspace.		Correct the value of the <i>Position</i> input variable.		Do not allow the target position to exceed the range of the workspace.
<b>Attached information</b>	<p>Attached Information 1: Error Location</p> <p>Attached Information 2: Error Location Details (Rung Number). For a program section, the rung number from the start of the section is given. For ST, the line number is given.</p> <p>Attached Information 3: Names of the Instruction and Instruction Instance Where the Error Occurred. If there is more than one possible instruction, information is given on all of them. Nothing is given if the instruction cannot be identified.</p> <p>Attached Information 4: Expansion Error Code (<i>ErrorIDEx</i>)</p>				
<b>Precautions/Remarks</b>	If a program is changed after an error occurs, the attached information that is displayed may not be correct.				

## 6 Troubleshooting

<b>Event name</b>	Cannot Cancel Synchronization		<b>Event code</b>	5401571C hex	
<b>Meaning</b>	The MC_SyncOut (End Synchronization) instruction cannot be executed.				
<b>Source</b>	PLC Function Module		<b>Source details</b>	Instruction	<b>Detection timing</b> At instruction execution
<b>Error attributes</b>	<b>Level</b>	Observation	<b>Recovery</b>	---	<b>Log category</b> System
<b>Effects</b>	<b>User program</b>	Continues	<b>Operation</b>	The relevant instruction will end according to specifications.	
<b>System-defined variables</b>	<b>Variable</b>	<b>Data type</b>		<b>Name</b>	
	None	---		---	
<b>Cause and correction</b>	<b>Assumed cause</b>		<b>Correction</b>		<b>Prevention</b>
	The MC_SyncLinearConveyor (Start Conveyor Synchronization) instruction execution is in progress, but <i>Phase</i> of the <i>TrajStatus</i> output variable is not 6.		Execute the MC_SyncOut (End Synchronization) instruction when <i>Phase</i> of the <i>TrajStatus</i> output variable is 6.		Execute the MC_SyncOut (End Synchronization) instruction only when <i>Phase</i> of the <i>TrajStatus</i> output variable from the MC_SyncLinearConveyor (Start Conveyor Synchronization) instruction is 6.
<b>Attached information</b>	<p>Attached Information 1: Error Location</p> <p>Attached Information 2: Error Location Details (Rung Number). For a program section, the rung number from the start of the section is given. For ST, the line number is given.</p> <p>Attached Information 3: Names of the Instruction and Instruction Instance Where the Error Occurred. If there is more than one possible instruction, information is given on all of them. Nothing is given if the instruction cannot be identified.</p> <p>Attached Information 4: Expansion Error Code (<i>ErrorIDEx</i>)</p>				
<b>Precautions/Remarks</b>	If a program is changed after an error occurs, the attached information that is displayed may not be correct.				

<b>Event name</b>	Too Many Kinematics		<b>Event code</b>	5401571E hex	
<b>Meaning</b>	The number of kinematics exceeded the limit.				
<b>Source</b>	PLC Function Module		<b>Source details</b>	Instruction	<b>Detection timing</b> At instruction execution
<b>Error attributes</b>	<b>Level</b>	Observation	<b>Recovery</b>	---	<b>Log category</b> System
<b>Effects</b>	<b>User program</b>	Continues	<b>Operation</b>	The relevant instruction will end according to specifications.	
<b>System-defined variables</b>	<b>Variable</b>	<b>Data type</b>		<b>Name</b>	
	None	---		---	
<b>Cause and correction</b>	<b>Assumed cause</b>		<b>Correction</b>		<b>Prevention</b>
	The number of the kinematics set by the MC_SetKinTransform (Set Kinematics Transformation) instruction exceeded the limit.		Review the ladder diagrams and make sure that the limit is not exceeded.		Do not allow the number of the kinematics to exceed the following limit: 8 for NJ501-4400/-4500/-4300/-4320 Units, 1 for an NJ501-4310 Unit.
<b>Attached information</b>	<p>Attached Information 1: Error Location</p> <p>Attached Information 2: Error Location Details (Rung Number). For a program section, the rung number from the start of the section is given. For ST, the line number is given.</p> <p>Attached Information 3: Names of the Instruction and Instruction Instance Where the Error Occurred. If there is more than one possible instruction, information is given on all of them. Nothing is given if the instruction cannot be identified.</p> <p>Attached Information 4: Expansion Error Code (<i>ErrorIDEx</i>)</p>				
<b>Precautions/Remarks</b>	If a program is changed after an error occurs, the attached information that is displayed may not be correct.				

## 6 Troubleshooting

<b>Event name</b>	Kinematics Initialization Error		<b>Event code</b>	5401571F hex	
<b>Meaning</b>	Kinematics initialization failed.				
<b>Source</b>	PLC Function Module		<b>Source details</b>	Instruction	<b>Detection timing</b> At instruction execution
<b>Error attributes</b>	<b>Level</b>	Observation	<b>Recovery</b>	---	<b>Log category</b> System
<b>Effects</b>	<b>User program</b>	Continues	<b>Operation</b>	The relevant instruction will end according to specifications.	
<b>System-defined variables</b>	<b>Variable</b>	<b>Data type</b>		<b>Name</b>	
	None	---		---	
<b>Cause and correction</b>	<b>Assumed cause</b>	<b>Correction</b>	<b>Prevention</b>		
	One of the robot axes (A0 to A3) is stopped at the position for which direct kinematics calculation cannot be performed.	Turn OFF the Servo for the robot axes. Ensure safety and move the axis inside the robot moving range.  If this error is detected, the settings of the kinematics, workspace, user coordinate systems and tool coordinate system for the relevant axes group are cleared.  To operate the robot, you need to make these settings again.	Review the program and make sure that the robot moves within its moving range.		
	Kinematics parameters set are incorrect.	Review and set kinematics parameters correctly according the existing robot.  If this error is detected, the settings of the kinematics, workspace, user coordinate systems and tool coordinate system for the relevant axes group are cleared.  To operate the robot, you need to make these settings again.	Review kinematics parameters of your robot.		
<b>Attached information</b>	<p>Attached Information 1: Error Location</p> <p>Attached Information 2: Error Location Details (Rung Number). For a program section, the rung number from the start of the section is given. For ST, the line number is given.</p> <p>Attached Information 3: Names of the Instruction and Instruction Instance Where the Error Occurred. If there is more than one possible instruction, information is given on all of them. Nothing is given if the instruction cannot be identified.</p> <p>Attached Information 4: Expansion Error Code (<i>ErrorIDEx</i>)</p>				
<b>Precautions/Remarks</b>	If a program is changed after an error occurs, the attached information that is displayed may not be correct.				

<b>Event name</b>	Invalid Tool Number		<b>Event code</b>	54015732 hex	
<b>Meaning</b>	The tool ID specified for the <i>ToolID</i> input variable to a motion control instruction is out of range or not defined.				
<b>Source</b>	PLC Function Module		<b>Source details</b>	Instruction	<b>Detection timing</b> At instruction execution
<b>Error attributes</b>	<b>Level</b>	Observation	<b>Recovery</b>	---	<b>Log category</b> System
<b>Effects</b>	<b>User program</b>	Continues	<b>Operation</b>	The relevant instruction will end according to specifications.	
<b>System-defined variables</b>	<b>Variable</b>	<b>Data type</b>		<b>Name</b>	
	None	---		---	
<b>Cause and correction</b>	<b>Assumed cause</b>		<b>Correction</b>		<b>Prevention</b>
	<p><i>ToolID</i> is outside the setting range.</p> <p>The specified tool ID is not defined by the MC_DefineToolTransform (Define Tool Coordinate) instruction.</p>		<p>Change the value of the <i>ToolID</i> input variable to a correct tool ID.</p> <p>Define the tool with the MC_DefineToolTransform (Define Tool Coordinate) instruction.</p>		<p>Set the tool ID correctly for the <i>ToolID</i> input variable.</p> <p>Define the tool and use it.</p>
<b>Attached information</b>	<p>Attached Information 1: Error Location</p> <p>Attached Information 2: Error Location Details (Rung Number). For a program section, the rung number from the start of the section is given. For ST, the line number is given.</p> <p>Attached Information 3: Names of the Instruction and Instruction Instance Where the Error Occurred. If there is more than one possible instruction, information is given on all of them. Nothing is given if the instruction cannot be identified.</p> <p>Attached Information 4: Expansion Error Code (<i>ErrorIDEx</i>)</p>				
<b>Precautions/Remarks</b>	If a program is changed after an error occurs, the attached information that is displayed may not be correct.				

<b>Event name</b>	Tool Parameter Out of Range		<b>Event code</b>	54015733 hex	
<b>Meaning</b>	The transform parameter specified for the <i>ToolTransform</i> input variable to a motion control instruction is out of range.				
<b>Source</b>	PLC Function Module		<b>Source details</b>	Instruction	<b>Detection timing</b> At instruction execution
<b>Error attributes</b>	<b>Level</b>	Observation	<b>Recovery</b>	---	<b>Log category</b> System
<b>Effects</b>	<b>User program</b>	Continues	<b>Operation</b>	The relevant instruction will end according to specifications.	
<b>System-defined variables</b>	<b>Variable</b>	<b>Data type</b>		<b>Name</b>	
	None	---		---	
<b>Cause and correction</b>	<b>Assumed cause</b>		<b>Correction</b>		<b>Prevention</b>
	The transform parameter is outside the setting range.		Change the value of the <i>ToolTransform</i> input variable to a correct transform parameter.		Set the transform parameter correctly for the <i>ToolTransform</i> input variable.
<b>Attached information</b>	<p>Attached Information 1: Error Location</p> <p>Attached Information 2: Error Location Details (Rung Number). For a program section, the rung number from the start of the section is given. For ST, the line number is given.</p> <p>Attached Information 3: Names of the Instruction and Instruction Instance Where the Error Occurred. If there is more than one possible instruction, information is given on all of them. Nothing is given if the instruction cannot be identified.</p> <p>Attached Information 4: Expansion Error Code (<i>ErrorIDEx</i>)</p>				
<b>Precautions/Remarks</b>	If a program is changed after an error occurs, the attached information that is displayed may not be correct.				

## 6 Troubleshooting

<b>Event name</b>	Offset Not Allowed		<b>Event code</b>	54015736 hex	
<b>Meaning</b>	During the MC_SyncLinearConveyor (Start Conveyor Synchronization) instruction execution, the offset function can be used only in Phase6.				
<b>Source</b>	PLC Function Module		<b>Source details</b>	Instruction	<b>Detection timing</b> At instruction execution
<b>Error attributes</b>	<b>Level</b>	Observation	<b>Recovery</b>	---	<b>Log category</b> System
<b>Effects</b>	<b>User program</b>	Continues	<b>Operation</b>	The relevant instruction will end according to specifications.	
<b>System-defined variables</b>	<b>Variable</b>	<b>Data type</b>		<b>Name</b>	
	None	---		---	
<b>Cause and correction</b>	<b>Assumed cause</b>		<b>Correction</b>		<b>Prevention</b>
	The <i>EnableOffset</i> input variable changed to TRUE before the MC_SyncLinearConveyor (Start Conveyor Synchronization) instruction operation entered Phase6.		Change the <i>EnableOffset</i> input variable to TRUE only when <i>Phase</i> of the <i>TrajStatus</i> output variable from the MC_SyncLinearConveyor (Start Conveyor Synchronization) instruction is 6.		Change the <i>EnableOffset</i> input variable to TRUE only when <i>Phase</i> of the <i>TrajStatus</i> output variable from the MC_SyncLinearConveyor (Start Conveyor Synchronization) instruction is 6.
<b>Attached information</b>	<p>Attached Information 1: Error Location</p> <p>Attached Information 2: Error Location Details (Rung Number). For a program section, the rung number from the start of the section is given. For ST, the line number is given.</p> <p>Attached Information 3: Names of the Instruction and Instruction Instance Where the Error Occurred. If there is more than one possible instruction, information is given on all of them. Nothing is given if the instruction cannot be identified.</p> <p>Attached Information 4: Expansion Error Code (<i>ErrorIDEx</i>)</p>				
<b>Precautions/Remarks</b>	If a program is changed after an error occurs, the attached information that is displayed may not be correct.				

<b>Event name</b>	Jog Mode Out of Range		<b>Event code</b>	54015744 hex	
<b>Meaning</b>	The <i>JogMode</i> input variable to the MC_RobotJog (Axes Group Jog) instruction is out of range.				
<b>Source</b>	PLC Function Module		<b>Source details</b>	Instruction	<b>Detection timing</b> At instruction execution
<b>Error attributes</b>	<b>Level</b>	Observation	<b>Recovery</b>	---	<b>Log category</b> System
<b>Effects</b>	<b>User program</b>	Continues	<b>Operation</b>	The relevant instruction will end according to specifications.	
<b>System-defined variables</b>	<b>Variable</b>	<b>Data type</b>		<b>Name</b>	
	None	---		---	
<b>Cause and correction</b>	<b>Assumed cause</b>		<b>Correction</b>		<b>Prevention</b>
	The <i>JogMode</i> input variable to the MC_RobotJog (Axes Group Jog) instruction is outside the setting range.		Correct the value of the <i>JogMode</i> input variable.		Select the jog mode correctly.
<b>Attached information</b>	<p>Attached Information 1: Error Location</p> <p>Attached Information 2: Error Location Details (Rung Number). For a program section, the rung number from the start of the section is given. For ST, the line number is given.</p> <p>Attached Information 3: Names of the Instruction and Instruction Instance Where the Error Occurred. If there is more than one possible instruction, information is given on all of them. Nothing is given if the instruction cannot be identified.</p> <p>Attached Information 4: Expansion Error Code (<i>ErrorIDEx</i>)</p>				
<b>Precautions/Remarks</b>	If a program is changed after an error occurs, the attached information that is displayed may not be correct.				

<b>Event name</b>	Initial Workpiece Position Out of Range		<b>Event code</b>	54015745 hex	
<b>Meaning</b>	The <i>InitWorkpiecePosition</i> input variable to the MC_SyncLinearConveyor (Start Conveyor Synchronization) instruction is out of range.				
<b>Source</b>	PLC Function Module		<b>Source details</b>	Instruction	<b>Detection timing</b> At instruction execution
<b>Error attributes</b>	<b>Level</b>	Observation	<b>Recovery</b>	---	<b>Log category</b> System
<b>Effects</b>	<b>User program</b>	Continues	<b>Operation</b>	The relevant instruction will end according to specifications.	
<b>System-defined variables</b>	<b>Variable</b>	<b>Data type</b>		<b>Name</b>	
	None	---		---	
<b>Cause and correction</b>	<b>Assumed cause</b>		<b>Correction</b>		<b>Prevention</b>
	The <i>InitWorkpiecePosition</i> input variable to the MC_SyncLinearConveyor (Start Conveyor Synchronization) instruction is outside the setting range.		Correct the value of the <i>InitWorkpiecePosition</i> input variable.		Review the program and make sure that the value of <i>InitWorkpiecePosition</i> input variable to the MC_SyncLinearConveyor (Start Conveyor Synchronization) instruction is within the setting range.
<b>Attached information</b>	<p>Attached Information 1: Error Location</p> <p>Attached Information 2: Error Location Details (Rung Number). For a program section, the rung number from the start of the section is given. For ST, the line number is given.</p> <p>Attached Information 3: Names of the Instruction and Instruction Instance Where the Error Occurred. If there is more than one possible instruction, information is given on all of them. Nothing is given if the instruction cannot be identified.</p> <p>Attached Information 4: Expansion Error Code (<i>ErrorIDEx</i>)</p>				
<b>Precautions/Remarks</b>	If a program is changed after an error occurs, the attached information that is displayed may not be correct.				

## 6 Troubleshooting

<b>Event name</b>	Maximum Interpolation Velocity Out of Range		<b>Event code</b>	54015746 hex	
<b>Meaning</b>	The <i>MaxVelocityTCP</i> input variable to the MC_SetKinTransform (Set Kinematics Transformation) instruction is out of range.				
<b>Source</b>	PLC Function Module		<b>Source details</b>	Instruction	<b>Detection timing</b> At instruction execution
<b>Error attributes</b>	<b>Level</b>	Observation	<b>Recovery</b>	---	<b>Log category</b> System
<b>Effects</b>	<b>User program</b>	Continues	<b>Operation</b>	The relevant instruction will end according to specifications.	
<b>System-defined variables</b>	<b>Variable</b>	<b>Data type</b>		<b>Name</b>	
	None	---		---	
<b>Cause and correction</b>	<b>Assumed cause</b>		<b>Correction</b>		<b>Prevention</b>
	The <i>MaxVelocityTCP</i> input variable to the MC_SetKinTransform (Set Kinematics Transformation) instruction is outside the setting range.		Correct the value of the <i>MaxVelocityTCP</i> input variable.		Set the maximum interpolation velocity correctly.
<b>Attached information</b>	<p>Attached Information 1: Error Location</p> <p>Attached Information 2: Error Location Details (Rung Number). For a program section, the rung number from the start of the section is given. For ST, the line number is given.</p> <p>Attached Information 3: Names of the Instruction and Instruction Instance Where the Error Occurred. If there is more than one possible instruction, information is given on all of them. Nothing is given if the instruction cannot be identified.</p> <p>Attached Information 4: Expansion Error Code (<i>ErrorIDEx</i>)</p>				
<b>Precautions/Remarks</b>	If a program is changed after an error occurs, the attached information that is displayed may not be correct.				

<b>Event name</b>	Maximum Interpolation Acceleration Out of Range		<b>Event code</b>	54015747 hex	
<b>Meaning</b>	The <i>MaxAccelerationTCP</i> input variable to the MC_SetKinTransform (Set Kinematics Transformation) instruction is out of range.				
<b>Source</b>	PLC Function Module		<b>Source details</b>	Instruction	<b>Detection timing</b> At instruction execution
<b>Error attributes</b>	<b>Level</b>	Observation	<b>Recovery</b>	---	<b>Log category</b> System
<b>Effects</b>	<b>User program</b>	Continues	<b>Operation</b>	The relevant instruction will end according to specifications.	
<b>System-defined variables</b>	<b>Variable</b>	<b>Data type</b>		<b>Name</b>	
	None	---		---	
<b>Cause and correction</b>	<b>Assumed cause</b>		<b>Correction</b>		<b>Prevention</b>
	The <i>MaxAccelerationTCP</i> input variable to the MC_SetKinTransform (Set Kinematics Transformation) instruction is outside the setting range.		Correct the value of the <i>MaxAccelerationTCP</i> input variable.		Set the maximum interpolation acceleration correctly.
<b>Attached information</b>	<p>Attached Information 1: Error Location</p> <p>Attached Information 2: Error Location Details (Rung Number). For a program section, the rung number from the start of the section is given. For ST, the line number is given.</p> <p>Attached Information 3: Names of the Instruction and Instruction Instance Where the Error Occurred. If there is more than one possible instruction, information is given on all of them. Nothing is given if the instruction cannot be identified.</p> <p>Attached Information 4: Expansion Error Code (<i>ErrorIDEx</i>)</p>				
<b>Precautions/Remarks</b>	If a program is changed after an error occurs, the attached information that is displayed may not be correct.				

## 6 Troubleshooting

<b>Event name</b>	Maximum Interpolation Deceleration Out of Range		<b>Event code</b>	54015748 hex	
<b>Meaning</b>	The <i>MaxDecelerationTCP</i> input variable to the MC_SetKinTransform (Set Kinematics Transformation) instruction is out of range.				
<b>Source</b>	PLC Function Module		<b>Source details</b>	Instruction	<b>Detection timing</b> At instruction execution
<b>Error attributes</b>	<b>Level</b>	Observation	<b>Recovery</b>	---	<b>Log category</b> System
<b>Effects</b>	<b>User program</b>	Continues	<b>Operation</b>	The relevant instruction will end according to specifications.	
<b>System-defined variables</b>	<b>Variable</b>	<b>Data type</b>		<b>Name</b>	
	None	---		---	
<b>Cause and correction</b>	<b>Assumed cause</b>		<b>Correction</b>		<b>Prevention</b>
	The <i>MaxDecelerationTCP</i> input variable to the MC_SetKinTransform (Set Kinematics Transformation) instruction is outside the setting range.		Correct the value of the <i>MaxDecelerationTCP</i> input variable.		Set the maximum interpolation deceleration correctly.
<b>Attached information</b>	<p>Attached Information 1: Error Location</p> <p>Attached Information 2: Error Location Details (Rung Number). For a program section, the rung number from the start of the section is given. For ST, the line number is given.</p> <p>Attached Information 3: Names of the Instruction and Instruction Instance Where the Error Occurred. If there is more than one possible instruction, information is given on all of them. Nothing is given if the instruction cannot be identified.</p> <p>Attached Information 4: Expansion Error Code (<i>ErrorIDEx</i>)</p>				
<b>Precautions/Remarks</b>	If a program is changed after an error occurs, the attached information that is displayed may not be correct.				

<b>Event name</b>	Target Velocity Ratio Setting Out of Range		<b>Event code</b>	5401574D hex	
<b>Meaning</b>	The parameter specified for the <i>VelocityRatio</i> input variable to a motion control instruction is out of range.				
<b>Source</b>	PLC Function Module		<b>Source details</b>	Instruction	<b>Detection timing</b> At instruction execution
<b>Error attributes</b>	<b>Level</b>	Observation	<b>Recovery</b>	---	<b>Log category</b> System
<b>Effects</b>	<b>User program</b>	Continues	<b>Operation</b>	The relevant instruction will end according to specifications.	
<b>System-defined variables</b>	<b>Variable</b>	<b>Data type</b>		<b>Name</b>	
	None	---		---	
<b>Cause and correction</b>	<b>Assumed cause</b>		<b>Correction</b>		<b>Prevention</b>
	Instruction input parameter exceeded the valid range of the input variable.		Correct the parameter so that the valid range of the input variable is not exceeded for the relevant instruction.		Set the input parameter to the instruction so that the valid range of the input variable is not exceeded.
<b>Attached information</b>	<p>Attached Information 1: Error Location</p> <p>Attached Information 2: Error Location Details (Rung Number). For a program section, the rung number from the start of the section is given. For ST, the line number is given.</p> <p>Attached Information 3: Names of the Instruction and Instruction Instance Where the Error Occurred. If there is more than one possible instruction, information is given on all of them. Nothing is given if the instruction cannot be identified.</p> <p>Attached Information 4: Expansion Error Code (<i>ErrorIDEx</i>)</p>				
<b>Precautions/Remarks</b>	If a program is changed after an error occurs, the attached information that is displayed may not be correct.				

<b>Event name</b>	Acceleration Ratio Setting Out of Range		<b>Event code</b>	5401574E hex	
<b>Meaning</b>	The parameter specified for the <i>AccelerationRatio</i> input variable to a motion control instruction is out of range.				
<b>Source</b>	PLC Function Module		<b>Source details</b>	Instruction	<b>Detection timing</b> At instruction execution
<b>Error attributes</b>	<b>Level</b>	Observation	<b>Recovery</b>	---	<b>Log category</b> System
<b>Effects</b>	<b>User program</b>	Continues	<b>Operation</b>	The relevant instruction will end according to specifications.	
<b>System-defined variables</b>	<b>Variable</b>		<b>Data type</b>		<b>Name</b>
	None		---		---
<b>Cause and correction</b>	<b>Assumed cause</b>		<b>Correction</b>		<b>Prevention</b>
	Instruction input parameter exceeded the valid range of the input variable.		Correct the parameter so that the valid range of the input variable is not exceeded for the relevant instruction.		Set the input parameter to the instruction so that the valid range of the input variable is not exceeded.
<b>Attached information</b>	<p>Attached Information 1: Error Location</p> <p>Attached Information 2: Error Location Details (Rung Number). For a program section, the rung number from the start of the section is given. For ST, the line number is given.</p> <p>Attached Information 3: Names of the Instruction and Instruction Instance Where the Error Occurred. If there is more than one possible instruction, information is given on all of them. Nothing is given if the instruction cannot be identified.</p> <p>Attached Information 4: Expansion Error Code (<i>ErrorIDEx</i>)</p>				
<b>Precautions/Remarks</b>	If a program is changed after an error occurs, the attached information that is displayed may not be correct.				

<b>Event name</b>	Deceleration Ratio Setting Out of Range		<b>Event code</b>	5401574F hex	
<b>Meaning</b>	The parameter specified for the <i>DecelerationRatio</i> input variable to a motion control instruction is out of range.				
<b>Source</b>	PLC Function Module		<b>Source details</b>	Instruction	<b>Detection timing</b> At instruction execution
<b>Error attributes</b>	<b>Level</b>	Observation	<b>Recovery</b>	---	<b>Log category</b> System
<b>Effects</b>	<b>User program</b>	Continues	<b>Operation</b>	The relevant instruction will end according to specifications.	
<b>System-defined variables</b>	<b>Variable</b>		<b>Data type</b>		<b>Name</b>
	None		---		---
<b>Cause and correction</b>	<b>Assumed cause</b>		<b>Correction</b>		<b>Prevention</b>
	Instruction input parameter exceeded the valid range of the input variable.		Correct the parameter so that the valid range of the input variable is not exceeded for the relevant instruction.		Set the input parameter to the instruction so that the valid range of the input variable is not exceeded.
<b>Attached information</b>	<p>Attached Information 1: Error Location</p> <p>Attached Information 2: Error Location Details (Rung Number). For a program section, the rung number from the start of the section is given. For ST, the line number is given.</p> <p>Attached Information 3: Names of the Instruction and Instruction Instance Where the Error Occurred. If there is more than one possible instruction, information is given on all of them. Nothing is given if the instruction cannot be identified.</p> <p>Attached Information 4: Expansion Error Code (<i>ErrorIDEx</i>)</p>				
<b>Precautions/Remarks</b>	If a program is changed after an error occurs, the attached information that is displayed may not be correct.				

## 6 Troubleshooting

<b>Event name</b>	Arm Configuration Setting Out of Range		<b>Event code</b>	54015750 hex	
<b>Meaning</b>	The parameter specified for the <i>ArmConfig</i> input variable to a motion control instruction is out of range.				
<b>Source</b>	PLC Function Module		<b>Source details</b>	Instruction	<b>Detection timing</b> At instruction execution
<b>Error attributes</b>	<b>Level</b>	Observation	<b>Recovery</b>	---	<b>Log category</b> System
<b>Effects</b>	<b>User program</b>	Continues	<b>Operation</b>	The relevant instruction will end according to specifications.	
<b>System-defined variables</b>	<b>Variable</b>	<b>Data type</b>		<b>Name</b>	
	None	---		---	
<b>Cause and correction</b>	<b>Assumed cause</b>		<b>Correction</b>		<b>Prevention</b>
	Instruction input parameter exceeded the valid range of the input variable.		Correct the parameter so that the valid range of the input variable is not exceeded for the relevant instruction.		Set the input parameter to the instruction so that the valid range of the input variable is not exceeded.
<b>Attached information</b>	<p>Attached Information 1: Error Location</p> <p>Attached Information 2: Error Location Details (Rung Number). For a program section, the rung number from the start of the section is given. For ST, the line number is given.</p> <p>Attached Information 3: Names of the Instruction and Instruction Instance Where the Error Occurred. If there is more than one possible instruction, information is given on all of them. Nothing is given if the instruction cannot be identified.</p> <p>Attached Information 4: Expansion Error Code (<i>ErrorIDEx</i>)</p>				
<b>Precautions/Remarks</b>	If a program is changed after an error occurs, the attached information that is displayed may not be correct.				

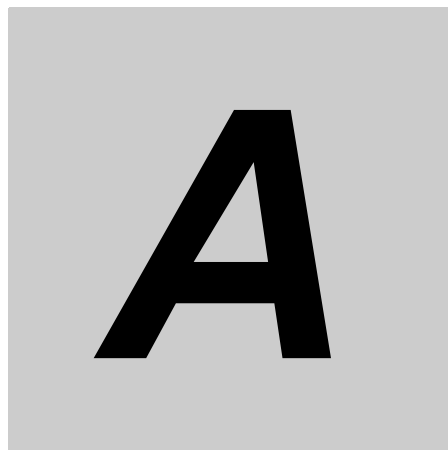
<b>Event name</b>	Positive Limit Input		<b>Event code</b>	54016443 hex	
<b>Meaning</b>	An instruction was executed for a motion in the positive direction when the positive limit input was ON.				
<b>Source</b>	PLC Function Module		<b>Source details</b>	Instruction	<b>Detection timing</b> At instruction execution
<b>Error attributes</b>	<b>Level</b>	Observation	<b>Recovery</b>	---	<b>Log category</b> System
<b>Effects</b>	<b>User program</b>	Continues	<b>Operation</b>	The relevant instruction will end according to specifications.	
<b>System-defined variables</b>	<b>Variable</b>		<b>Data type</b>		<b>Name</b>
	None		---		---
<b>Cause and correction</b>	<b>Assumed cause</b>		<b>Correction</b>		<b>Prevention</b>
	An instruction for a motion in the positive direction was executed when the positive limit input was ON, or an instruction for a motion with no direction specification was executed when the positive limit input was ON. An axes group motion control instruction was executed when the positive limit input was ON.		Execute an error reset and then perform a recovery operation in the negative direction. If the error occurred during an axes group motion control instruction, disable the axes group and then perform the above operation. If this error occurs again, check the connection of the positive limit signal, the logic setting for the positive limit input, and the execution conditions for the start command, and correct any mistakes. Check the logic settings both in the axis parameters and in the slave settings.		Check to make sure there are no problems with the positive limit signal connection, the logic setting for the positive limit input, and the execute conditions for the instruction. Check the logic settings both in the axis parameters and in the slave settings.
<b>Attached information</b>	<p>Attached Information 1: Error Location</p> <p>Attached Information 2: Error Location Details (Rung Number). For a program section, the rung number from the start of the section is given. For ST, the line number is given.</p> <p>Attached Information 3: Names of the Instruction and Instruction Instance Where the Error Occurred. If there is more than one possible instruction, information is given on all of them. Nothing is given if the instruction cannot be identified.</p> <p>Attached Information 4: Expansion Error Code (<i>ErrorIDEx</i>)</p>				
<b>Precautions/Remarks</b>	If a program is changed after an error occurs, the attached information that is displayed may not be correct.				

## 6 Troubleshooting

<b>Event name</b>	Negative Limit Input		<b>Event code</b>	54016444 hex	
<b>Meaning</b>	An instruction for a motion in the negative direction was executed when the negative limit input was ON.				
<b>Source</b>	PLC Function Module		<b>Source details</b>	Instruction	<b>Detection timing</b> At instruction execution
<b>Error attributes</b>	<b>Level</b>	Observation	<b>Recovery</b>	---	<b>Log category</b> System
<b>Effects</b>	<b>User program</b>	Continues	<b>Operation</b>	The relevant instruction will end according to specifications.	
<b>System-defined variables</b>	<b>Variable</b>	<b>Data type</b>		<b>Name</b>	
	None	---		---	
<b>Cause and correction</b>	<b>Assumed cause</b>		<b>Correction</b>		<b>Prevention</b>
	An instruction for a motion in the negative direction was executed when the negative limit input was ON, or an instruction for a motion with no direction specification was executed when the negative limit input was ON. An axes group motion control instruction was executed when the negative limit input was ON.		Execute an error reset and then perform a recovery operation in the positive direction. If the error occurred during an axes group motion control instruction, disable the axes group and then perform the above operation. If this error occurs again, check the connection of the negative limit signal, the logic setting for the negative limit input, and the execution conditions for the start command, and correct any mistakes. Check the logic settings both in the axis parameters and in the slave settings.		Check to make sure there are no problems with the negative limit signal connection, the logic setting for the negative limit input, and the execute conditions for the instruction. Check the logic settings both in the axis parameters and in the slave settings.
<b>Attached information</b>	<p>Attached Information 1: Error Location</p> <p>Attached Information 2: Error Location Details (Rung Number). For a program section, the rung number from the start of the section is given. For ST, the line number is given.</p> <p>Attached Information 3: Names of the Instruction and Instruction Instance Where the Error Occurred. If there is more than one possible instruction, information is given on all of them. Nothing is given if the instruction cannot be identified.</p> <p>Attached Information 4: Expansion Error Code (<i>ErrorIDEx</i>)</p>				
<b>Precautions/Remarks</b>	If a program is changed after an error occurs, the attached information that is displayed may not be correct.				

<b>Event name</b>	Current Position Outside Workspace		<b>Event code</b>	54016701 hex	
<b>Meaning</b>	The current position was outside the workspace when an instruction was executed.				
<b>Source</b>	PLC Function Module		<b>Source details</b>	Instruction	<b>Detection timing</b> At instruction execution
<b>Error attributes</b>	<b>Level</b>	Observation	<b>Recovery</b>	---	<b>Log category</b> System
<b>Effects</b>	<b>User program</b>	Continues	<b>Operation</b>	The relevant instruction will end according to specifications.	
<b>System-defined variables</b>	<b>Variable</b>	<b>Data type</b>		<b>Name</b>	
	None	---		---	
<b>Cause and correction</b>	<b>Assumed cause</b>		<b>Correction</b>		<b>Prevention</b>
	<p>The current position was outside the workspace when one of the following robot instructions was executed.</p> <ul style="list-style-type: none"> <li>• MC_MoveTimeAbsolute (Time-specified Absolute Positioning) instruction</li> <li>• MC_SyncLinearConveyor (Start Conveyor Synchronization) instruction</li> <li>• MC_SyncOut (End Synchronization) instruction</li> <li>• MC_RobotJog (Axes Group Jog) instruction</li> </ul>		Move the robot inside the workspace.		Review the program and make sure that the robot moves within the range of the workspace.
<b>Attached information</b>	<p>Attached Information 1: Error Location</p> <p>Attached Information 2: Error Location Details (Rung Number). For a program section, the rung number from the start of the section is given. For ST, the line number is given.</p> <p>Attached Information 3: Names of the Instruction and Instruction Instance Where the Error Occurred. If there is more than one possible instruction, information is given on all of them. Nothing is given if the instruction cannot be identified.</p> <p>Attached Information 4: Expansion Error Code (<i>ErrorIDEx</i>)</p>				
<b>Precautions/Remarks</b>	If a program is changed after an error occurs, the attached information that is displayed may not be correct.				





# Appendices

This chapter describes how to enable/disable the Sysmac Studio Robot Additional Option and provides information on each function offered by this option. It also describes how to enable/disable the Sysmac Studio robot options.

---

<b>A-1 Sysmac Studio Robot Additional Option</b> .....	<b>A-2</b>
A-1-1 Enabling the Sysmac Studio Robot Additional Option .....	A-2
A-1-2 Disabling the Sysmac Studio Robot Additional Option .....	A-4
A-1-3 3D Equipment Model Creation Wizard .....	A-6
A-1-4 3D Machine Models .....	A-8
A-1-5 Calculation of Calibration Parameters .....	A-19
A-1-6 Display of Images in 3D Motion Monitoring .....	A-20
<b>A-2 Sysmac Studio Robot Options</b> .....	<b>A-21</b>
A-2-1 Enabling Sysmac Studio Robot Options .....	A-21
A-2-2 Disabling Sysmac Studio Robot Options .....	A-23
<b>A-3 Version Information</b> .....	<b>A-25</b>

A

# A-1 Sysmac Studio Robot Additional Option

To perform a Vision & Robot integrated simulation, you need to purchase the “Robot Additional Option,” a Sysmac Studio option.

To enable the Robot Additional Option, you need to register your license in the Sysmac Studio version 1.14 or higher. You cannot register your license in the Sysmac Studio version 1.13 or lower.

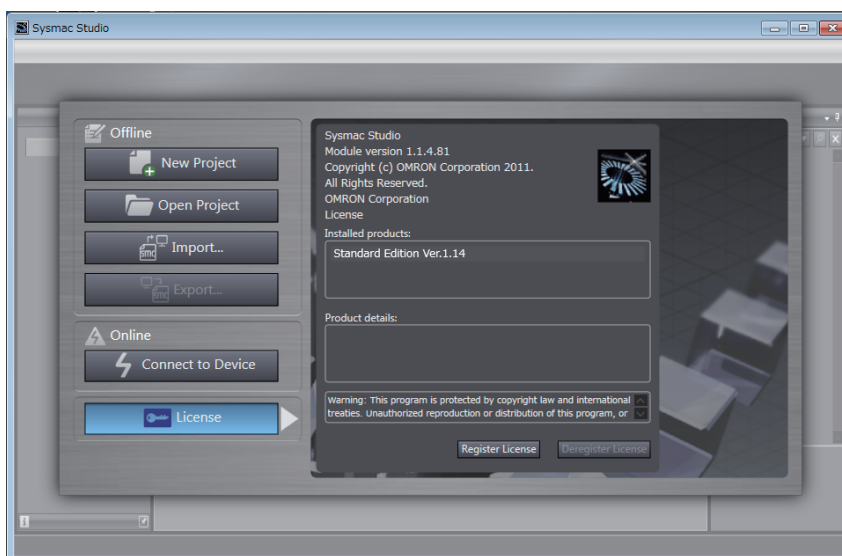
The Robot Additional Option provides the following functions necessary to perform a Vision & Robot integrated simulation.

- 3D Equipment Model Creation Wizard
- Addition of 3D machine models for the pick-and-place application, including the conveyors for picking
- Calculation of calibration parameters
- Display of images in 3D motion monitoring

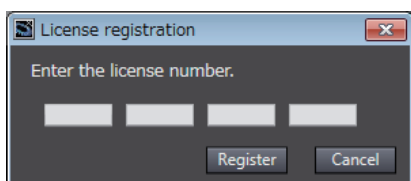
## A-1-1 Enabling the Sysmac Studio Robot Additional Option

You can use the Robot Additional Option after you take the following procedure on the Sysmac Studio.

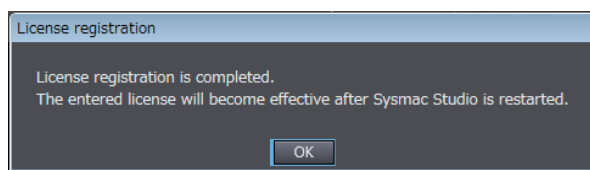
- 1 Install the Sysmac Studio Standard Edition.  
Refer to the *Sysmac Studio Version 1 Operation Manual* (Cat. No. W504) for how to install.
- 2 Start the Sysmac Studio and click the **License** Button on the left side of the startup window. Then, click the **Register License** Button.



- 3 Enter the Robot Additional Option license number, and click the **Register** Button.

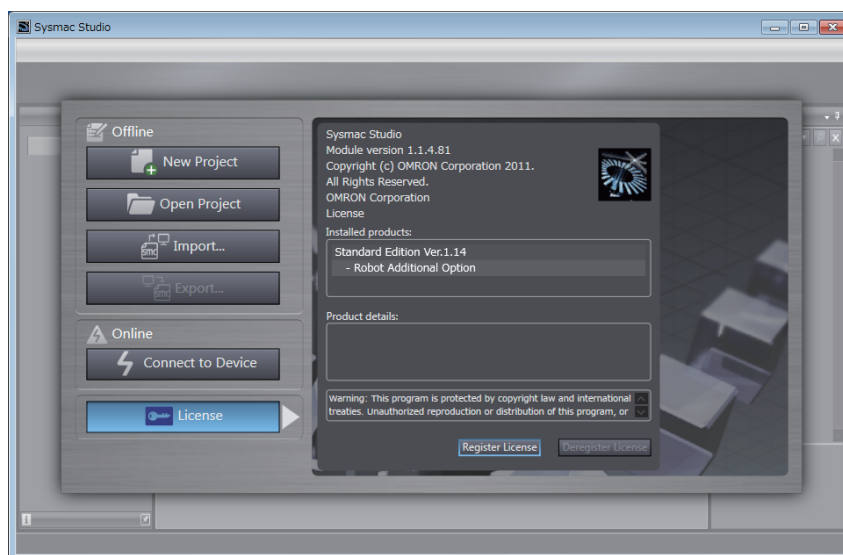


- 4** If the license is registered successfully, the following message appears.



Click the **OK** Button to close the window.

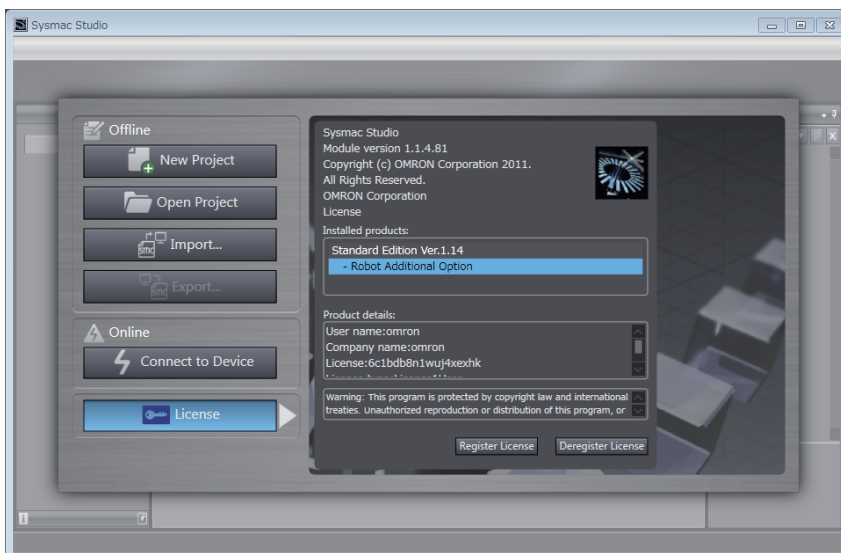
- 5** Confirm that **Robot Additional Option** is displayed under **Installed products** as shown below. Restart the Sysmac Studio.



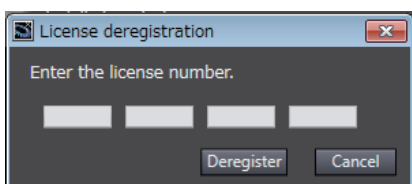
## A-1-2 Disabling the Sysmac Studio Robot Additional Option

Use the following procedure to disable the Sysmac Studio Robot Additional Option or to move the license of the Sysmac Studio Robot Additional Option from a computer to another.

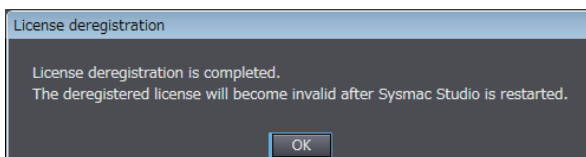
- 1 Start the Sysmac Studio and click the **License** Button on the left side of the startup window. Then, click the **Deregister License** Button.



- 2 Enter the Robot Additional Option license number, and click the **Deregister** Button.

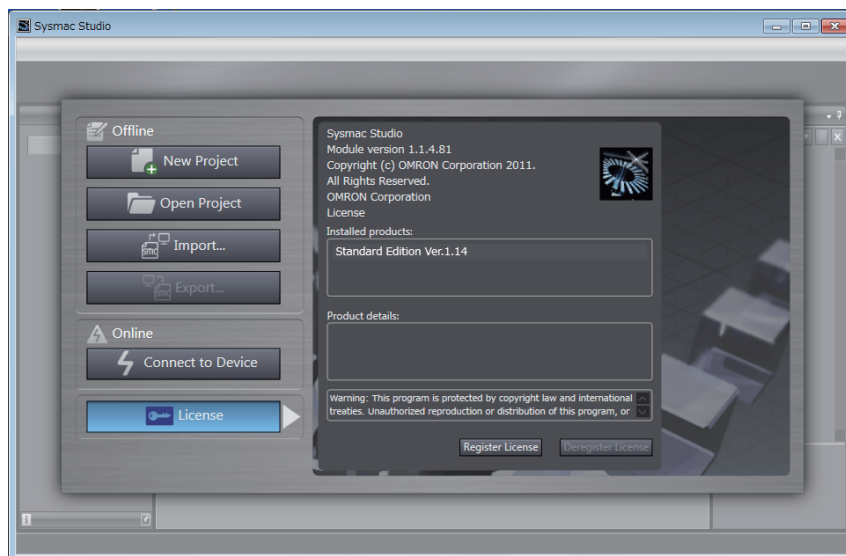


- 3 If the license is deregistered successfully, the following message appears.



Click the **OK** Button to close the window.

- 4 Confirm that **Robot Additional Option** is not displayed under **Installed products** as shown below. Restart the Sysmac Studio.



### A-1-3 3D Equipment Model Creation Wizard

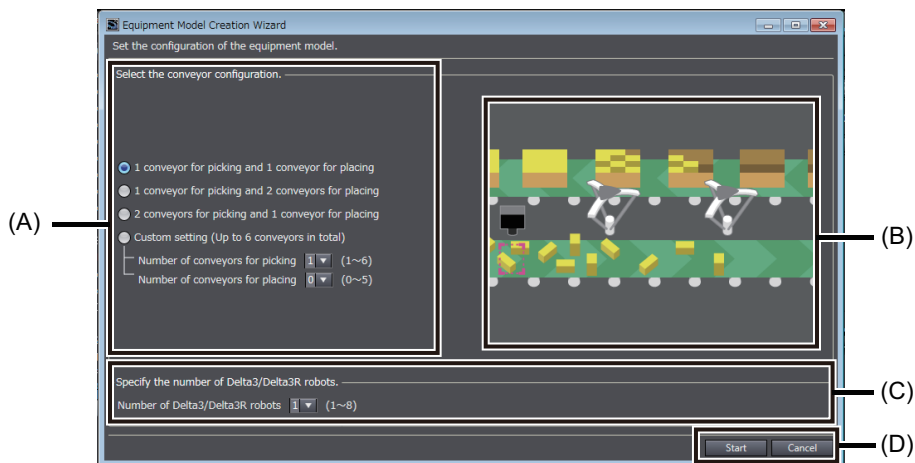
The 3D Equipment Model Creation Wizard allows you to select how many conveyors for picking, how many conveyors for placing, and how many robots to use for the target pick-and-place application to configure 3D machine models for conveyors and robots.

After you complete the settings in this wizard, calibration parameters that represent the coordinate transformation data for each 3D machine model is generated automatically.

For detailed operating procedures, refer to the *Vision & Robot Integrated Simulation Startup Guide* (Cat. No. Y128).

### 3D Equipment Model Settings

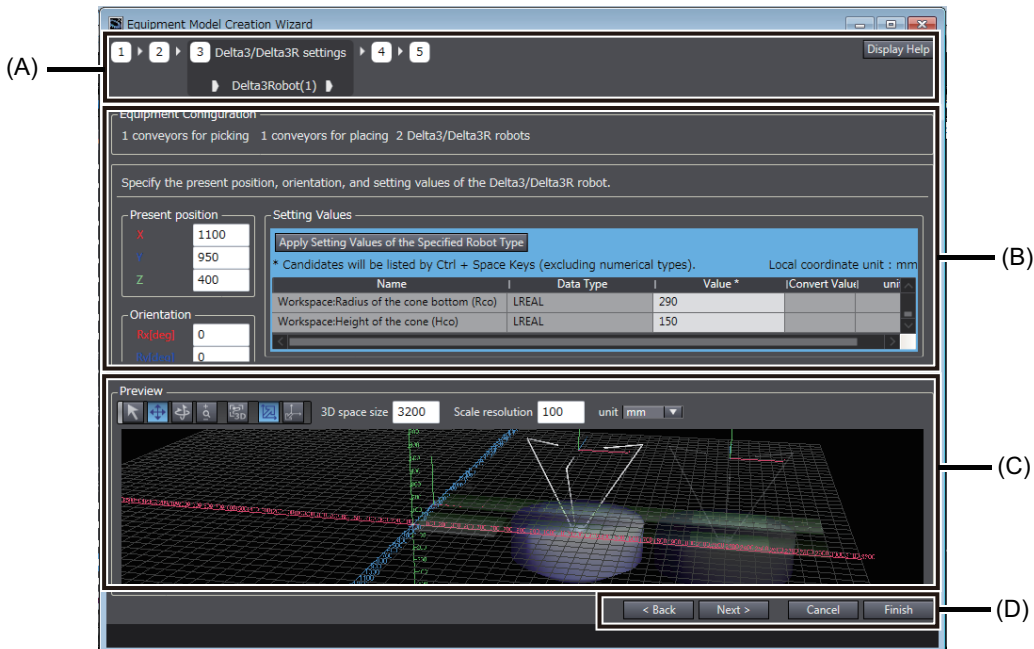
Use this window to set the configuration of a 3D equipment model that consists of conveyor settings, the number of robots, and other information.



Symbol	Name	Description
(A)	3D equipment model configuration pattern selection	Select the conveyor configuration from among <b>1 conveyor for picking and 1 conveyor for placing</b> , <b>1 conveyor for picking and 2 conveyors for placing</b> , <b>2 conveyors for picking and 1 conveyor for placing</b> , and <b>Custom setting</b> .
(B)	3D equipment model image display	This area displays the image of the pattern selected in the 3D equipment model configuration pattern selection area.
(C)	Number of robots specification	Select the number of robots.
(D)	<b>Start</b> Button and <b>Cancel</b> Button	Click the <b>Start</b> Button to start the wizard with the settings of the selected conveyor configuration and the specified number of Delta3/Delta3R robots. Click the <b>Cancel</b> Button to close the window.

## 3D Machine Model Settings

Use this window to configure the position and size parameters of each 3D machine model. Refer to *A-1-4 3D Machine Models* on page A-8 for details on each 3D machine model.



Symbol	Name	Description
(A)	Navigation area	This area displays the step numbers of the wizard and the name of the current 3D machine model.  If you click the <b>Display Help</b> Button that is enabled only when you configure the robot settings, the user's manual that provides a description of each setting appears in an auto display window.
(B)	Setup area	Set the parameters, position, and orientation of each 3D machine model.
(C)	Preview area	This area shows a preview of the 3D machine model that you configured in the setup area.  If you change these settings, the preview will be updated to reflect the changes.
(D)	Action area	Click the <b>Finish</b> Button to display the finish setup page.  Click the <b>Next</b> Button to display the next page of the 3D machine model setup window.  Click the <b>Back</b> Button to display the previous page of the 3D machine model setup window.  If you click the <b>Cancel</b> Button, a confirmation dialog box appears to ask you if you want to discard your edits. Follow the instruction in the dialog box.

A-1 Sysmac Studio Robot Additional Option

**A**

A-1-3 3D Equipment Model Creation Wizard

## A-1-4 3D Machine Models

This section describes the parameters of the 3D machine models that you can use with the Robot Additional Option.

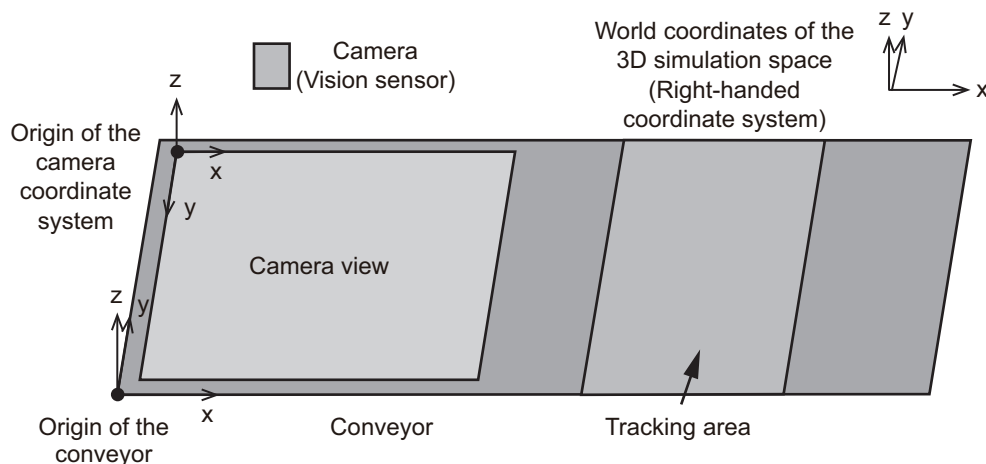
Use the 3D machine models described here in a Vision & Robot integrated simulation.

### Conveyor for Picking

This model represents a conveyor for picking.

#### ● Components of this 3D Machine Model

The 3D machine model for a conveyor for picking has the components shown in the figure below.



Component name	Description
Camera view	The area in which the vision sensor captures images. The workpieces detected by the vision sensor's simulator are displayed in this area.
Conveyor	The conveyor for picking. It moves the workpieces displayed in the camera view.
Tracking area	Robots can pick the workpieces located in this area. This area has the same width as the width of the conveyor.
Camera	The model of the camera used for the simulation of the pick-and-place application.

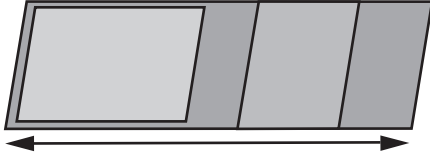
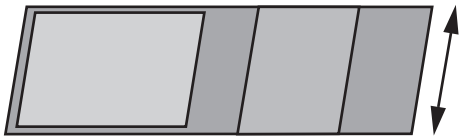
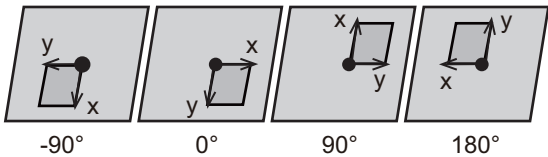
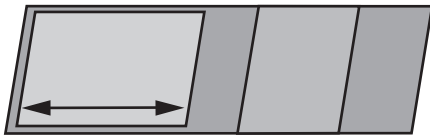
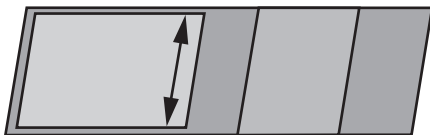
#### ● Operation of this 3D Machine Model


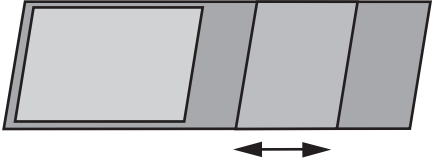
This 3D machine model performs the following two operations.

- Displaying workpieces in the initial position of the camera view
- Moving workpieces from the initial position according to the conveyor's encoder value

## ● Settings

The 3D machine model parameter setting area has the following settings.

Name	Data type	Description
Conveyor: Length	LREAL	Set the length of the conveyor's belt. 
Conveyor: Width	LREAL	Set the width of the conveyor's belt. 
Conveyor: Corresponding variable	_sAXIS_REF	Set the conveyor's encoder axis. The actual values will be loaded.
Camera: Display	---	Select whether you want to show or hide the camera. If you select to hide the camera, the camera and the camera view will be hidden.
Camera: Orientation	---	Select the orientation of the camera from among $-90^\circ$ , $0^\circ$ , $90^\circ$ , and $180^\circ$ . 
Camera view: Origin (X)	LREAL	Set the X-coordinate of the origin of the camera view.
Camera view: Origin (Y)	LREAL	Set the Y-coordinate of the origin of the camera view.
Camera view: X view	LREAL	Set the length of the camera view. 
Camera view: Y view	LREAL	Set the width of the camera view. 
Workpieces: Quantity	USINT	Set the number of workpieces to pick to between 1 and 200. The set value then appears on the conveyor image.
Workpiece: Height	LREAL	Set the height of the workpiece to pick.
Workpiece: Length	LREAL	Set the length of the workpiece to pick.
Workpiece: Width	LREAL	Set the width of the workpiece to pick.

Name	Data type	Description
Tracking area: Position (X)	LREAL	<p>Set the position of the conveyor area where robots can pick workpieces.</p> <p>This position must be set as an X-coordinate in the local coordinate system of the conveyor. You can set this for up to eight robots.</p> 
Tracking area: Length	LREAL	<p>Set the length of the conveyor area where robots can pick workpieces. You can set this for up to eight robots.</p> 
Active robot	---	<p>Set the 3D machine model name for each Delta3/Delta3R robot that uses the conveyor. You can set up to eight robots.</p>
Vision sensor: Node address	UINT	<p>Set the node address of the vision sensor in the EtherCAT network.</p>
Vision sensor: Line No.	USINT	<p>Set the line number of the vision sensor that appears in the I/O Map.</p>

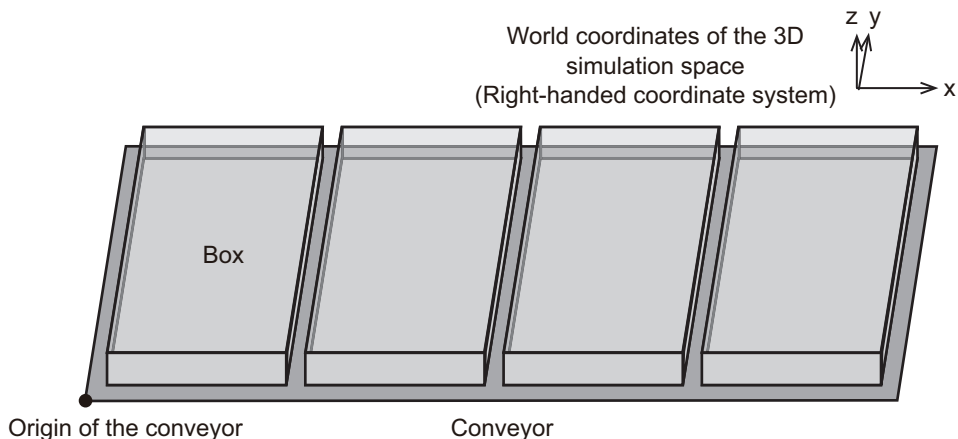
Name	Data type	Description
Vision sensor: Variables of detected workpiece information	sSimWorkInitial-Data (User-defined)	<p>Get device variables for the vision sensor.</p> <p>From among these device variables, assign the necessary variables at once to the data type, which is a user-defined structure.</p> <p>The necessary variables are given below.</p> <ul style="list-style-type: none"> <li>• Data output completion flag</li> <li>• Number of detected workpieces</li> <li>• Encoder value when triggered</li> <li>• Measured (X-/Y-) coordinates of workpieces</li> <li>• Measured angles of workpieces</li> </ul> <p>You can set these variables individually.</p> <p>The following variables are set automatically when you correctly set variables of detected workpiece information.</p>
Data output completion flag: Corresponding variable	BOOL	This flag indicates that data was output from the vision sensor. The workpiece image is generated when this flag changes to TRUE.
Encoder value when triggered: Corresponding variable	LREAL	The encoder count when the vision sensor captured the image.
Number of detected workpieces: Corresponding variable	LREAL	The number of workpieces detected in the single image.
Index of workpiece 1: Corresponding variable	LREAL	The number to indicate the type of the first workpiece detected.
Measured X-coordinate of workpiece 1: Corresponding variable	LREAL	The X-coordinate of the position in which the first workpiece was detected.
Measured Y-coordinate of workpiece 1: Corresponding variable	LREAL	The Y-coordinate of the position in which the first workpiece was detected.
Measured angle of workpiece 1: Corresponding variable	LREAL	The pose of the first workpiece detected.
...		
Index of workpiece 7: Corresponding variable	LREAL	The number to indicate the type of the seventh workpiece detected.
Measured X-coordinate of workpiece 7: Corresponding variable	LREAL	The X-coordinate of the position in which the seventh workpiece was detected.
Measured Y-coordinate of workpiece 7: Corresponding variable	LREAL	The Y-coordinate of the position in which the seventh workpiece was detected.
Measured angle of workpiece 7: Corresponding variable	LREAL	The pose of the seventh workpiece detected.

## Conveyor for Placing (Box)

This model represents a conveyor for placing with boxes placed on it. It allows you to check whether workpieces are packed in boxes.

### ● Components of this 3D Machine Model

The 3D machine model for a Conveyor for placing (Box) has the components shown in the figure below.



### ● Operation of this 3D Machine Model

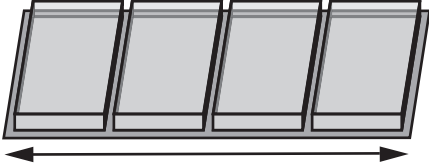
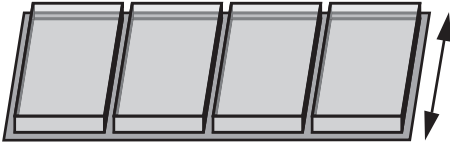
The boxes displayed on the conveyor are all in the same size and placed at equal intervals.


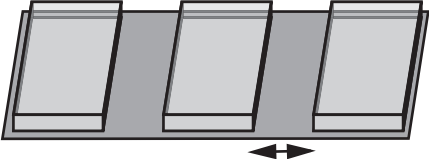
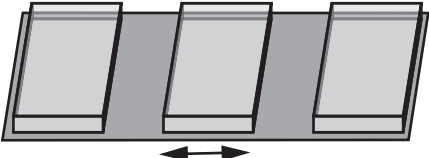
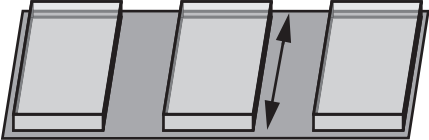
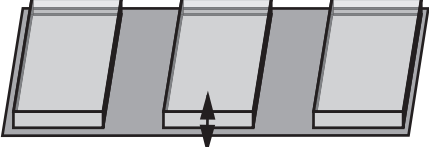
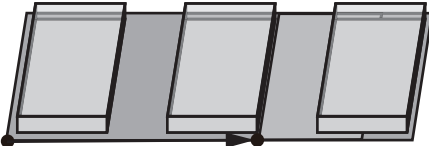
These boxes move according to the value of the encoder axis when the trigger variable that corresponds with the conveyor is turned ON. Each box disappears when it reaches the end of the conveyor.

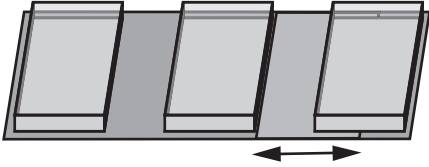
As with the case of the boxes, the workpieces placed either on the conveyor or in the boxes move as the encoder value increases. Each workpiece disappears when it reaches the end of the conveyor.

### ● Settings

The 3D machine model parameter setting area has the following settings.

Name	Data type	Description
Conveyor: Length	LREAL	Set the length of the conveyor's belt. 
Conveyor: Width	LREAL	Set the width of the conveyor's belt. 

Name	Data type	Description
Conveyor: Workpiece acceptance height	LREAL	<p>Set the height at which workpieces are considered to come in contact with the conveyor. Any workpiece placed at this height or lower is considered to be in contact with the conveyor and moved in conjunction with the conveyor.</p> 
Conveyor: Corresponding variable	_sAXIS_REF	<p>Set the conveyor's encoder axis. The actual values will be loaded.</p>
Box: Interval	LREAL	<p>Set the interval between two adjacent boxes.</p> 
Box: Length	LREAL	<p>Set the length of the boxes.</p> 
Box: Width	LREAL	<p>Set the width of the boxes.</p> 
Box: Height	LREAL	<p>Set the height of the boxes.</p> 
Box: Transparency of side [%]	USINT	<p>Set the transparency of the side faces of the boxes to between 0 and 100 (transparent), so that you can easily view how robots place workpieces.</p>
Tracking area: Position (X)	LREAL	<p>Set the position of the conveyor area where robots can place workpieces.</p> <p>This position must be set as an X-coordinate in the local coordinate system of the conveyor. You can set this for up to eight robots.</p> 

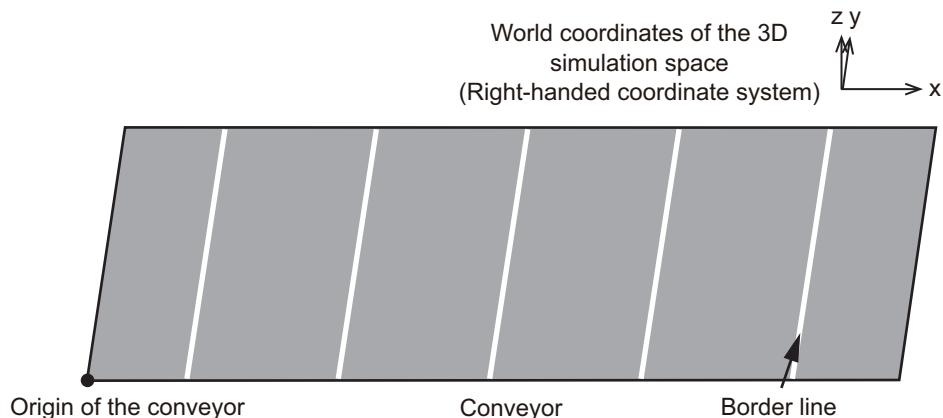
Name	Data type	Description
Tracking area: Length	LREAL	Set the length of the conveyor area where robots can place workpieces. You can set this for up to eight robots. 
Active robot	---	Set the 3D machine model name for each Delta3/Delta3R robot that uses the conveyor. You can set up to eight robots.
Trigger variable: Corresponding variable	BOOL	Set the trigger variable that changes to TRUE when a box is detected. The boxes do not move until this trigger variable changes to TRUE. The trigger for the detection sensor is located at the starting point of the conveyor.

## Conveyor for Placing (with Borders)

This model represents a conveyor for placing with equally spaced border lines. It allows you to view how robots place workpieces in the predetermined range.

### ● Components of this 3D Machine Model

The 3D machine model for a Conveyor for placing (with borders) has the components shown in the figure below.



### ● Operation of this 3D Machine Model

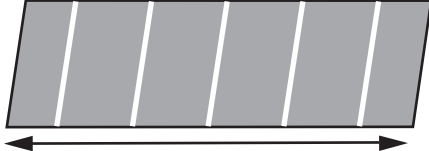


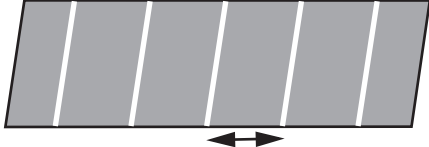

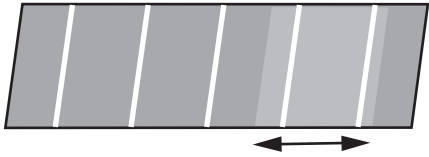
The border lines displayed on the conveyor are spaced at equal intervals.

These border lines move according to the value of the encoder axis when the trigger variable that corresponds with the conveyor is turned ON. Each border line disappears when it reaches the end of the conveyor.

As with the case of the border lines, the workpieces placed on the conveyor move as the encoder value increases. Each workpiece disappears when it reaches the end of the conveyor.

● **Settings**

The 3D machine model parameter setting area has the following settings.

Name	Data type	Description
Conveyor: Length	LREAL	Set the length of the conveyor's belt. 
Conveyor: Width	LREAL	Set the width of the conveyor's belt. 
Conveyor: Workpiece acceptance height	LREAL	Set the height at which workpieces are considered to come in contact with the conveyor. Any workpiece placed at this height or lower is considered to be in contact with the conveyor and moved in conjunction with the conveyor. 
Conveyor: Corresponding variable	_sAXIS_REF	Set the conveyor's encoder axis. The actual values will be loaded.
Border line: Interval	LREAL	Set the display interval between two adjacent border lines. 
Tracking area: Position (X)	LREAL	Set the position of the conveyor area where robots can place workpieces. This position must be set as an X-coordinate in the local coordinate system of the conveyor. You can set this for up to eight robots. 
Tracking area: Length	LREAL	Set the length of the conveyor area where robots can place workpieces. You can set this for up to eight robots. 
Active robot	---	Set the 3D machine model name for each Delta3/Delta3R robot that uses the conveyor. You can set up to eight robots.

Name	Data type	Description
Trigger variable: Corresponding variable	BOOL	Set the trigger variable that changes to TRUE when a border line is detected. The border lines do not move until this trigger variable changes to TRUE.  The trigger for the detection sensor is located at the starting point of the conveyor.

## Delta3/Delta3R Robot

This model represents a Delta3 robot with three parallel links.

### ● Components of this 3D Machine Model

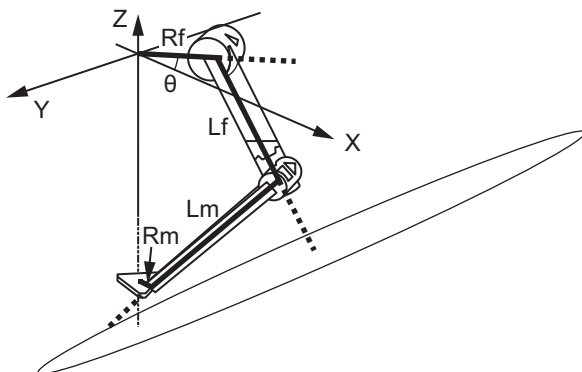
Refer to the description for Delta3/Delta3R in *2-3 Robot Functions* on page 2-20 for details.

### ● Operation of this 3D Machine Model

Refer to the description for Delta3/Delta3R in *2-3 Robot Functions* on page 2-20 for details.

### ● Settings

The 3D machine model parameter setting area has the following settings.



Name	Data type	Description
<b>Apply Setting Values of the Specified Robot Type</b> Button	---	<p>Use this button to apply setting values automatically based on the robot type that you select.</p> <p>Clicking the <b>Apply Setting Values of the Specified Robot Type</b> Button opens the following dialog box, in which you can automatically apply setting values for the supported robot types.</p> <div data-bbox="798 1563 1364 1787" data-label="Image"> </div> <p>In the <b>Robot Type List</b> Box, select the robot type of which you want to apply the setting values.</p> <ul style="list-style-type: none"> <li>The robot types displayed in this list vary depending on the regional setting of the computer on which the Sysmac Studio is installed.</li> <li>Click the <b>Apply</b> Button to apply the setting values of the selected robot type to the 3D machine model.</li> </ul>

Name	Data type	Description
Fixed frame: Radius (Rf)	LREAL	Set the distance between the center of the fixed frame and the axis motor. (Unit: mm)
Link 1: Length (Lf)	LREAL	Set the length of link 1. (Unit: mm)
Link 2: Length (Lm)	LREAL	Set the length of link 2. (Unit: mm)
Arm: Thickness	LREAL	Set the thickness of arm 1/2/3, and the size of base rotary axis 1/2/3 and intermediate rotary axis 1/2/3.  The size of the joint changes according to the zoom factor.
Moving frame: Radius (Rm)	LREAL	Set the distance between the center of moving frame and the joint of link 2. (Unit: mm)
Axes Group: Corresponding variable	_sGROUP_REF	The axes group that composes a Delta3 robot. Setting a variable automatically sets the corresponding variables for the axis coordinate systems (A0, A1, A2, and A3) within the axes group.
Axis Coordinate System for Axis A0: Corresponding variable	_sAXIS_REF	The corresponding variable for arm 1.
Axis Coordinate System for Axis A1: Corresponding variable	_sAXIS_REF	The corresponding variable for arm 2.
Axis Coordinate System for Axis A2: Corresponding variable	_sAXIS_REF	The corresponding variable for arm 3.
Axis Coordinate System for Axis A3: Corresponding variable	_sAXIS_REF	The corresponding variable for the rotary axis attached to the end effector .
Axis Coordinate System for Axis A3: Rotation direction	---	You can select <i>CW</i> or <i>CCW</i> .
Axis Coordinate System for Axis A0: Initial angle	LREAL	The angle of arm 1 in its initial state.
Axis Coordinate System for Axis A1: Initial angle	LREAL	The angle of arm 2 in its initial state.
Axis Coordinate System for Axis A2: Initial angle	LREAL	The angle of arm 3 in its initial state.
Origin position for rotation around Z axis of machine coordinate system: Theta [deg]	LREAL	The parameter to rotate the machine coordinate system around the Z axis.  You can set this to between $0^\circ$ and $360^\circ$ .
Workspace: Show	---	Select whether to show or hide the workspace.
Workspace: Transparency [%]	USINT	Set the transparency of the workspace polygon to between $0$ and $100$ (transparent).
Workspace: Front position of the cylinder (Zu)	LREAL	Refer to the description of Delta3/Delta3R workspace in <i>2-3 Robot Functions</i> on page 2-20 for details.
Workspace: Radius of the cylinder (Rcy)	LREAL	
Workspace: Height of the cylinder (Hcy)	LREAL	
Workspace: Height of the cone (Hco)	LREAL	
Workspace: Radius of the cone bottom (Rco)	LREAL	

## Vacuum-type Robot Tool

This model represents a robot tool that uses the suction of air to perform pick-and-place operation.

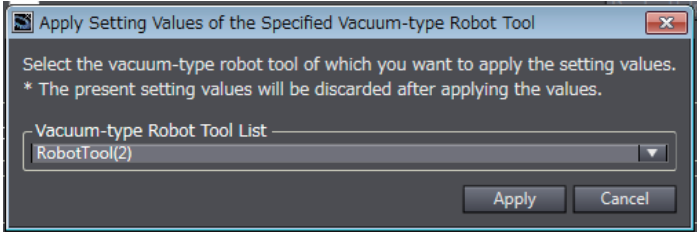
### ● Components of this 3D Machine Model

Refer to *2-3-8 Robot Tool* on page 2-41 for details.

### ● Operation of this 3D Machine Model

- The vacuum-type robot tool picks and holds a workpiece when its vacuum pads come in contact with the workpiece only when the input of the corresponding variable for the robot is ON.
- The vacuum-type robot tool stops holding the workpiece by its vacuum pads as soon as the input of the corresponding variable for the robot turns OFF.

### ● Settings

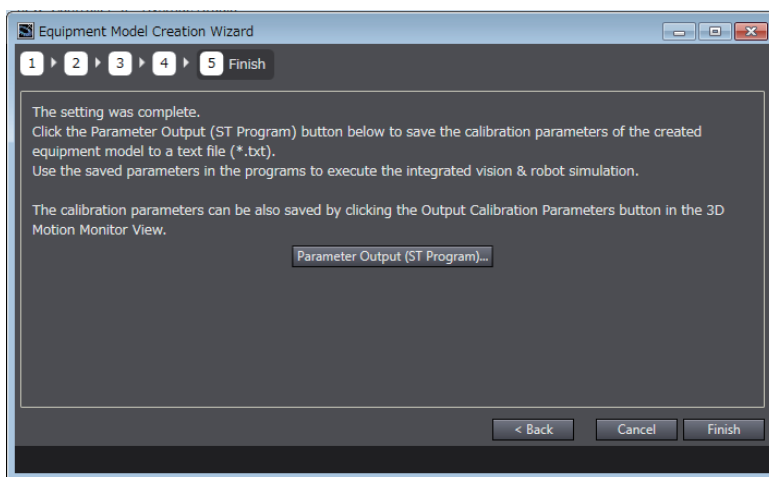
Item	Data type	Description
Apply Setting Values of the Specified Vacuum-type Robot Tool Button	---	<p>Clicking the <b>Apply Setting Values of the Specified Vacuum-type Robot Tool</b> Button opens the following dialog box, in which you can automatically apply setting values for the supported vacuum-type robot tools.</p>  <p>In the <b>Vacuum-type Robot Tool List</b> Box, select the vacuum-type robot tool of which you want to apply the setting values.</p> <ul style="list-style-type: none"> <li>• Click the <b>Apply</b> Button to apply the setting values of the selected vacuum-type robot tool to the 3D machine model.</li> <li>• This function does not apply the setting values for the items <b>Connected 3D machine model</b> and <b>Robot vacuum tool: Corresponding variable</b>.</li> </ul>
Robot vacuum tool: Home position in Tool Coordinate System (Tz)	LREAL	Refer to <i>2-3-8 Robot Tool</i> on page 2-41 for details.
Robot vacuum tool: Don't display the picked workpieces.	---	<p>Show or hide the picked workpieces.</p> <p>TRUE: Hides the picked workpieces.</p> <p>FALSE: Shows the picked workpieces.</p> <p>This function is effective when you check only whether the workpieces are picked.</p>
Connected 3D machine model	---	Specify the 3D machine model (robot) on which you want to mount the robot tool.
Robot vacuum tool: Number of pieces	USINT	Set the number of vacuum pads to between 1 and 16.
Robot vacuum tool: Corresponding variable	BOOL	A variable that corresponds with the vacuum ON/OFF signal. Workpieces are picked when this variable is TRUE and released when it changes to FALSE.

Item	Data type	Description
Robot vacuum tool: Home position in Tool Coordinate System (Tx)	LREAL	Refer to 2-3-8 <i>Robot Tool</i> on page 2-41 for details.
Robot vacuum tool: Home position in Tool Coordinate System (Ty)	LREAL	

## A-1-5 Calculation of Calibration Parameters

To perform a Vision & Robot integrated simulation, you need to set the parameters to transform the coordinate system (calibration parameters) in the user program.

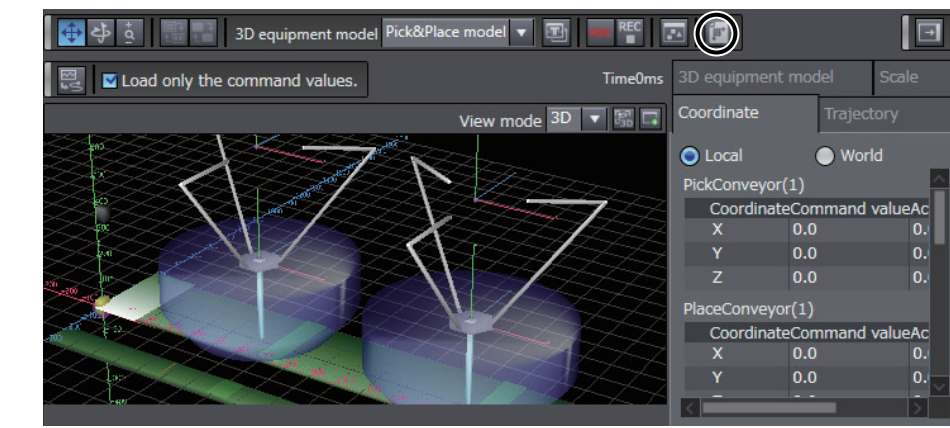
Once you complete setting all the 3D machine models with the 3D Equipment Model Creation Wizard, you can output the calibration parameters to a file.



For details on how to use and operate these parameters, refer to the *Vision & Robot Integrated Simulation Startup Guide* (Cat. No. Y128).

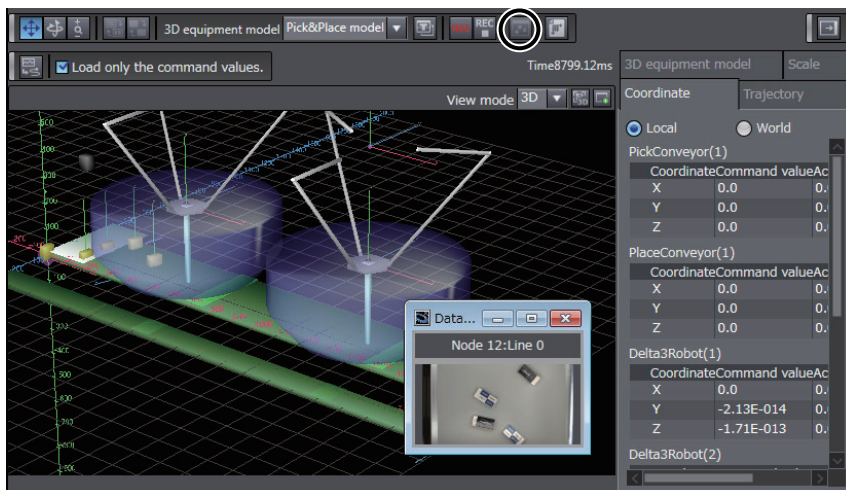
### Additional Information

If the selected 3D equipment model includes the conveyor for picking, you can save the calibration parameters in a file by clicking the **Output Calibration Parameters** Button in the 3D Motion Monitor View.



## A-1-6 Display of Images in 3D Motion Monitoring

In a Vision & Robot integrated simulation, when you execute 3D motion monitoring after loading the trace data for a 3D equipment model for the pick-and-place application, it is possible to display the images of workpieces captured by the vision sensor in accordance with the operation timing of the 3D equipment model.

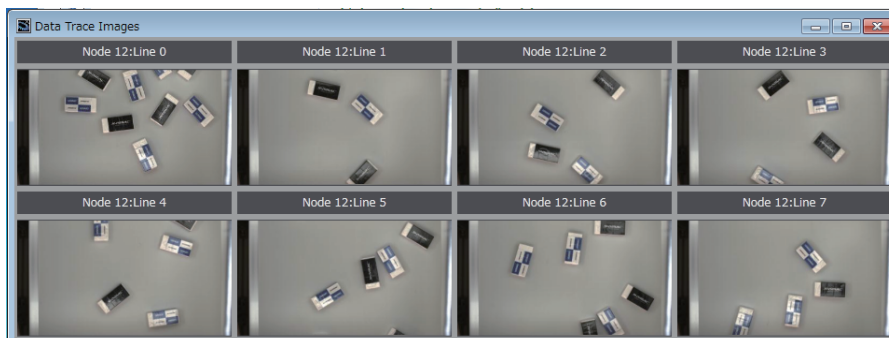


For detailed operating procedures, refer to the *Vision & Robot Integrated Simulation Startup Guide* (Cat. No. Y128).



### Additional Information

In a 3D equipment model for the pick-and-place application, you can use up to eight vision sensors. The following is a display example where eight vision sensors are used.



## A-2 Sysmac Studio Robot Options

This section describes how to enable/disable the Sysmac Studio robot options.

### Version Information

For the Sysmac Studio version 1.13 or lower, you need to enable the Sysmac Studio robot options to use the robot functions with an NJ Robotics CPU Unit (Model: NJ501-4□□□).

For the Sysmac Studio version 1.14 or higher, you can use the robot functions without enabling the robot options.

After you enable the robot options, take the following two steps to set the required parameters.

- Set the axes group and axes for the robot.
- Set the items such as link length of the robot.

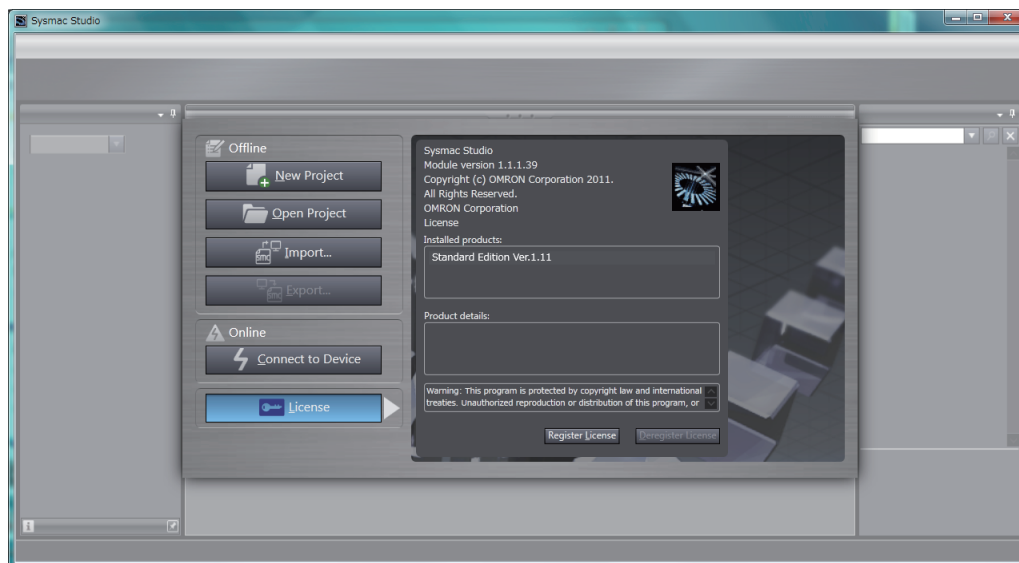
Refer to *3-1 Setting Group for Robot* on page 3-2 for the settings of the axes group and axes for the robot, and *3-2 Robot Kinematics Settings* on page 3-9 for the settings of the link length of the robot and other items.

Also refer to the *Sysmac Studio Version 1 Operation Manual* (Cat. No. W504) for details on operation and parameters to be set.

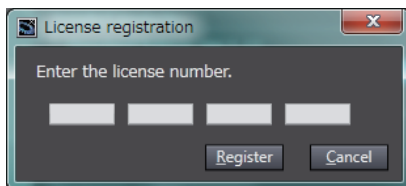
### A-2-1 Enabling Sysmac Studio Robot Options

You can use the robot options after you take the following procedure on the Sysmac Studio.

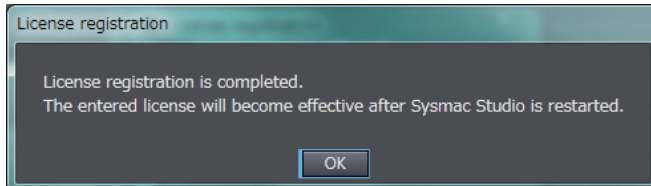
- 1 Install the Sysmac Studio Standard Edition.  
Refer to the *Sysmac Studio Version 1 Operation Manual* (Cat. No. W504) for how to install.
- 2 Start the Sysmac Studio and click the **License** Button on the left side of the startup window. Then, click the **Register License** Button.



- 3** Enter the robot options license number, and click the **Register** Button.

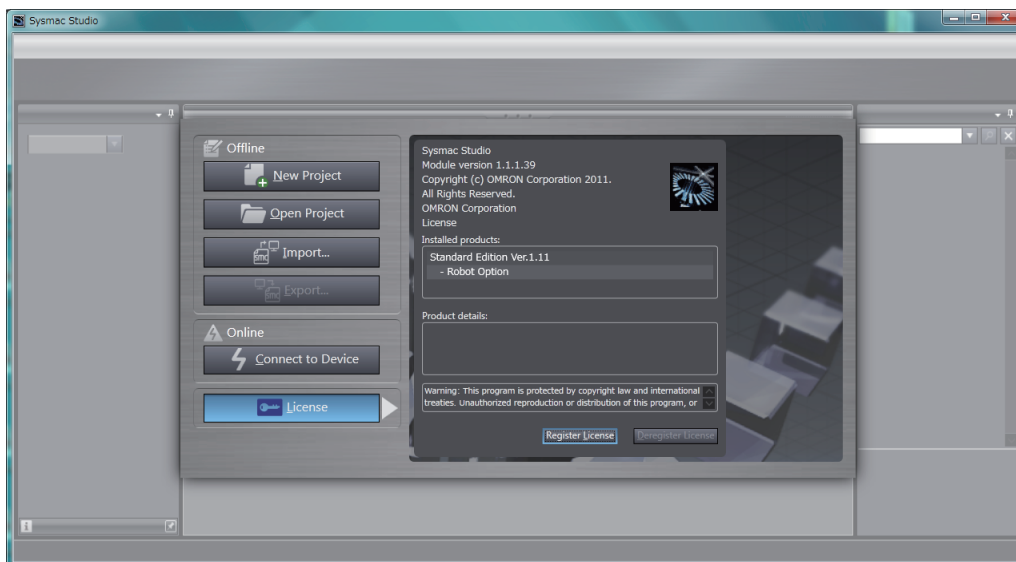


- 4** If the license is registered successfully, the following message appears.



Click the **OK** Button to close the window.

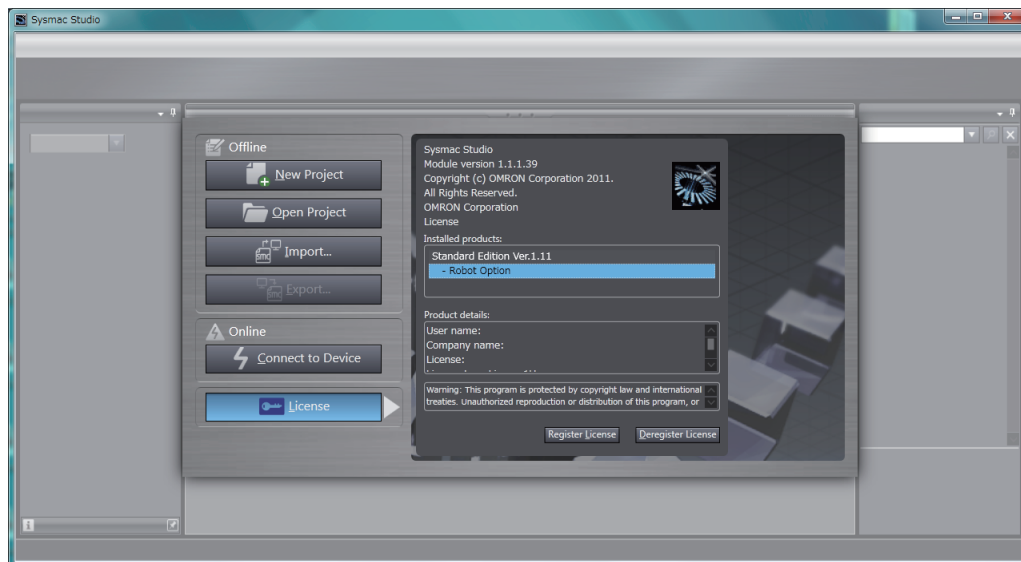
- 5** Confirm that **Robot Option** is displayed under **Installed products** as shown below. Restart the Sysmac Studio.



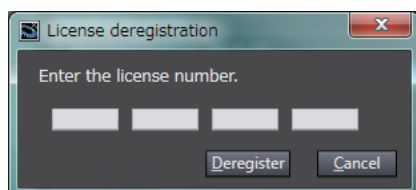
## A-2-2 Disabling Sysmac Studio Robot Options

Use the following procedure to disable the Sysmac Studio robot options or to move the license of the Sysmac Studio robot options from a computer to another.

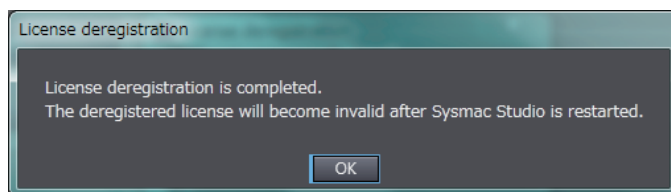
- 1 Start the Sysmac Studio and click the **License** Button on the left side of the startup window. Then, click the **Deregister License** Button.



- 2 Enter the robot options license number, and click the **Deregister** Button.

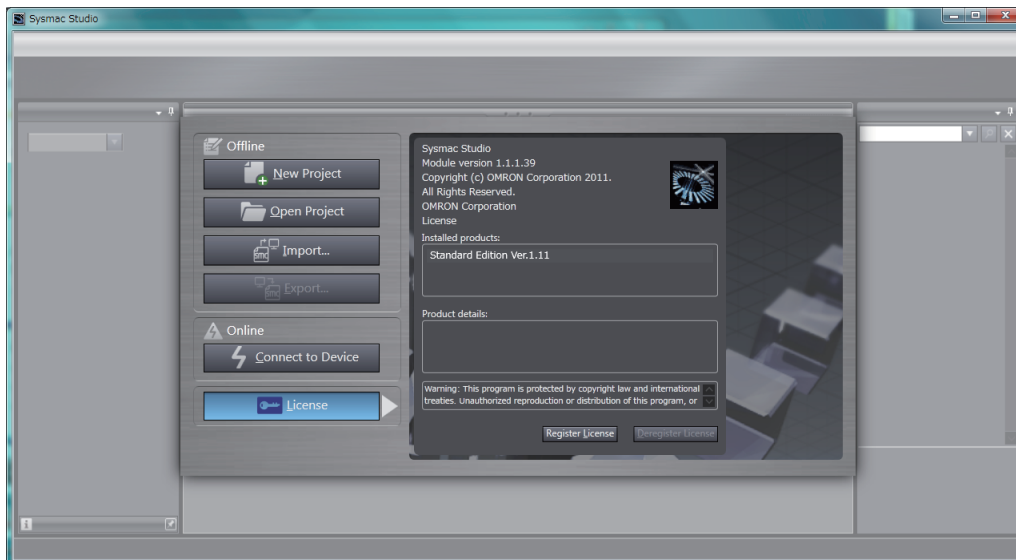


- 3 If the license is deregistered successfully, the following message appears.



Click the **OK** Button to close the window.

- 4** Confirm that **Robot Option** is not displayed under **Installed products** as shown below. Restart the Sysmac Studio.



## A-3 Version Information

This section describes the functions that are supported for each robot version.

Refer to the *NJ/NX-series Motion Control Instructions Reference Manual* (Cat. No. W508) for version information on the motion control instructions.

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

### Robot Control Functions That Were Added for Robot Version 1.00

Version 1.03 or higher of the Sysmac Studio is required to use the functions that were added for robot version 1.00 of the CPU Unit.

Function	Outline
Delta3 robot	<p>The control functions for the Delta3 robot were supported.</p> <p>The Delta3 robot can be controlled or that behaviors can be monitored by using the following robot instructions.</p> <ul style="list-style-type: none"> <li>• MC_SetKintransform</li> <li>• MC_SyncLinearCoveyor</li> <li>• MC_SyncOut</li> <li>• MC_DefineCoordinateSystem</li> <li>• MC_MoveTimeAbsolute</li> <li>• MC_GroupMon</li> <li>• MC_InverseKin</li> </ul>

### Robot Control Functions That Were Added for Robot Version 1.01

Version 1.09 or higher of the Sysmac Studio is required to use the functions that were added for robot version 1.01 of the CPU Unit.

Function	Outline
Delta3R robot and Delta2 robot	<p>The control functions for the Delta3R robot and Delta2 robot were supported. These robots can be controlled or that behavior can be monitored by using same robot instructions as Delta3 robot.</p>
Function of the tool define coordinate system	<p>The MC_DefineCoordSystem instruction was supported.</p> <p>Tool coordinate system can be set for the robot by using this instruction.</p>
Function of the axes group jog	<p>The MC_RobotJog instruction was supported.</p> <p>The axes group can be jog by using this instructions.</p>
Transition mode	<p>The Transition mode was supported for the MC_MoveTimeAbsolute instruction and the MC_SyncLinearConveyor instruction.</p>
Buffer mode selection	<p>The Buffered mode was supported for the MC_MoveTimeAbsolute instruction and the MC_SyncLinearConveyor instruction.</p>

## Robot Control Functions That Were Added for Robot Version 1.02

Version 1.10 or higher of the Sysmac Studio is required to use the functions that were added for robot version 1.02 of the CPU Unit.

Function	Outline
Vision & Robot Integrated Simulation	Vision & Robot Integrated Simulation was supported in Sysmac Studio.

## Robot Control Functions That Were Added for Robot Version 1.03

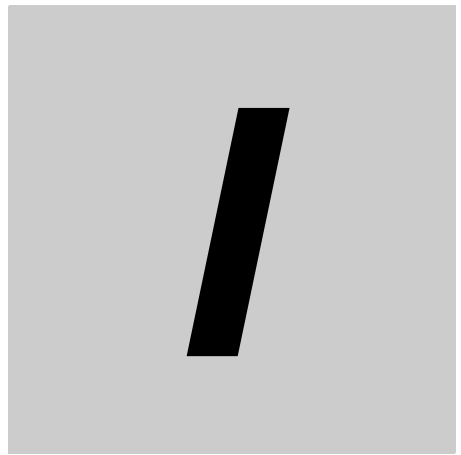
Version 1.15 or higher of the Sysmac Studio is required to use the functions that were added for robot version 1.03 of the CPU Unit.

Function	Outline
Cartesian robot	<p>The control functions for the Cartesian robot were supported.</p> <p>The Cartesian robot can be controlled and that behavior can be monitored by using same robot instruction as the Delta3 robot.</p> <p>The types of Robot that were supported are Cartesian 3D, Cartesian 3D Gantry, Cartesian 2D, Cartesian Gantry and H-Bot.</p>
The deceleration stop for MC_SyncOut	The deceleration stop for MC_SyncOut was supported.

## Robot Control Functions That Were Added for Robot Version 1.04

Version 1.18 or higher of the Sysmac Studio is required to use the functions that were added for robot version 1.04 of the CPU Unit.

Function	Outline
SCARA robot	<p>The control functions for SCARA robots were supported.</p> <p>The SCARA robot can be controlled and that behavior can be monitored by using same robot instruction as the Delta3 robot.</p> <p>The types of Robot that were supported are SCARA RPP, SCARA PRR, SCARA RRP+R and SCARA PRR+R.</p>
Expansion1 robot	<p>The control functions for Expansion1 robots were supported.</p> <p>The Expansion1 robot can be controlled by using ACS that were calculated by the MC_InverseKin instruction.</p>
Point-to-Point Joint Interpolated Positioning	<p>The MC_MoveDirectAbsolute instruction was supported.</p> <p>The current position can be moved to the target position by using this instructions.</p>



# Index

---



# Index

---

## Numerics

---

3D Equipment Model Creation Wizard .....A-6

## A

---

ACS ..... 2-5, 2-19

Assumed cause ..... 6-2

Axes group

    Basic settings ..... 3-3

    Operation settings ..... 3-3

Axis

    Basic settings ..... 3-5

Axis coordinate system ..... 2-5, 2-19

    Directions ..... 2-14

    Positional relationship with machine

        coordinate system ..... 2-18

## C

---

Calibration parameter .....A-19

Coordinate system

    Cartesian 2D ..... 2-8

    Cartesian 2D Gantry ..... 2-9

    Cartesian 3D ..... 2-7

    Cartesian 3D Gantry ..... 2-8

    Delta2 ..... 2-7

    Delta3 ..... 2-5

    Delta3R ..... 2-6

    Expansion1 ..... 2-13

    H-Bot ..... 2-9

    SCARA PRR ..... 2-12

    SCARA PRR+R ..... 2-12

    SCARA RRP ..... 2-11

    SCARA RRP+R ..... 2-12

    Types ..... 2-5

## E

---

Electronic gear ratio setting

    Check ..... 3-15

Event code ..... 6-2

Event name ..... 6-2

Execution instructions ..... 4-4

## H

---

Homing settings ..... 3-8

## K

---

Kinematics

    Parameter setting ..... 2-20

    Type setting ..... 2-20

## L

---

Limit settings ..... 3-7

## M

---

Machine coordinate system ..... 2-5, 2-19

MC Test Run ..... 3-15

MCS ..... 2-5, 2-19

## O

---

Offset setting ..... 2-40

Operation settings ..... 3-5

Other operation settings ..... 3-7

## P

---

Pick-and-place ..... 2-40, 5-2, 5-3, A-2

Position count settings ..... 3-8

## R

---

Robot motion range ..... 2-36

Robot system configuration ..... 1-5

## S

---

Start remaining time ..... 4-102

Stop instructions ..... 4-4

## T

---

TCS ..... 2-5, 2-19

Tool coordinate system ..... 2-5, 2-19

## U

---

UCS ..... 2-5, 2-19

Unit conversion settings ..... 3-5

User coordinate system ..... 2-5, 2-19

## V

---

Vision sensor ..... 1-5, 5-3, A-8

**W**

---

## Workspace

Parameter setting .....	2-31
Types .....	2-30





**OMRON Corporation Industrial Automation Company**  
Kyoto, JAPAN

Contact: [www.ia.omron.com](http://www.ia.omron.com)

**Regional Headquarters**

**OMRON EUROPE B.V.**

Wegalaan 67-69, 2132 JD Hoofddorp  
The Netherlands  
Tel: (31)2356-81-300/Fax: (31)2356-81-388

**OMRON ELECTRONICS LLC**

2895 Greenspoint Parkway, Suite 200  
Hoffman Estates, IL 60169 U.S.A  
Tel: (1) 847-843-7900/Fax: (1) 847-843-7787

**OMRON ASIA PACIFIC PTE. LTD.**

No. 438A Alexandra Road # 05-05/08 (Lobby 2),  
Alexandra Technopark,  
Singapore 119967  
Tel: (65) 6835-3011/Fax: (65) 6835-2711

**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 2015-2017 All Rights Reserved.  
In the interest of product improvement,  
specifications are subject to change without notice.

**Cat. No. W539-E1-05**

0417



# SCATTERGOOD & JOHNSON LTD

ELECTRICAL ENGINEERING & FLUID CONTROL DISTRIBUTORS

Est.1899

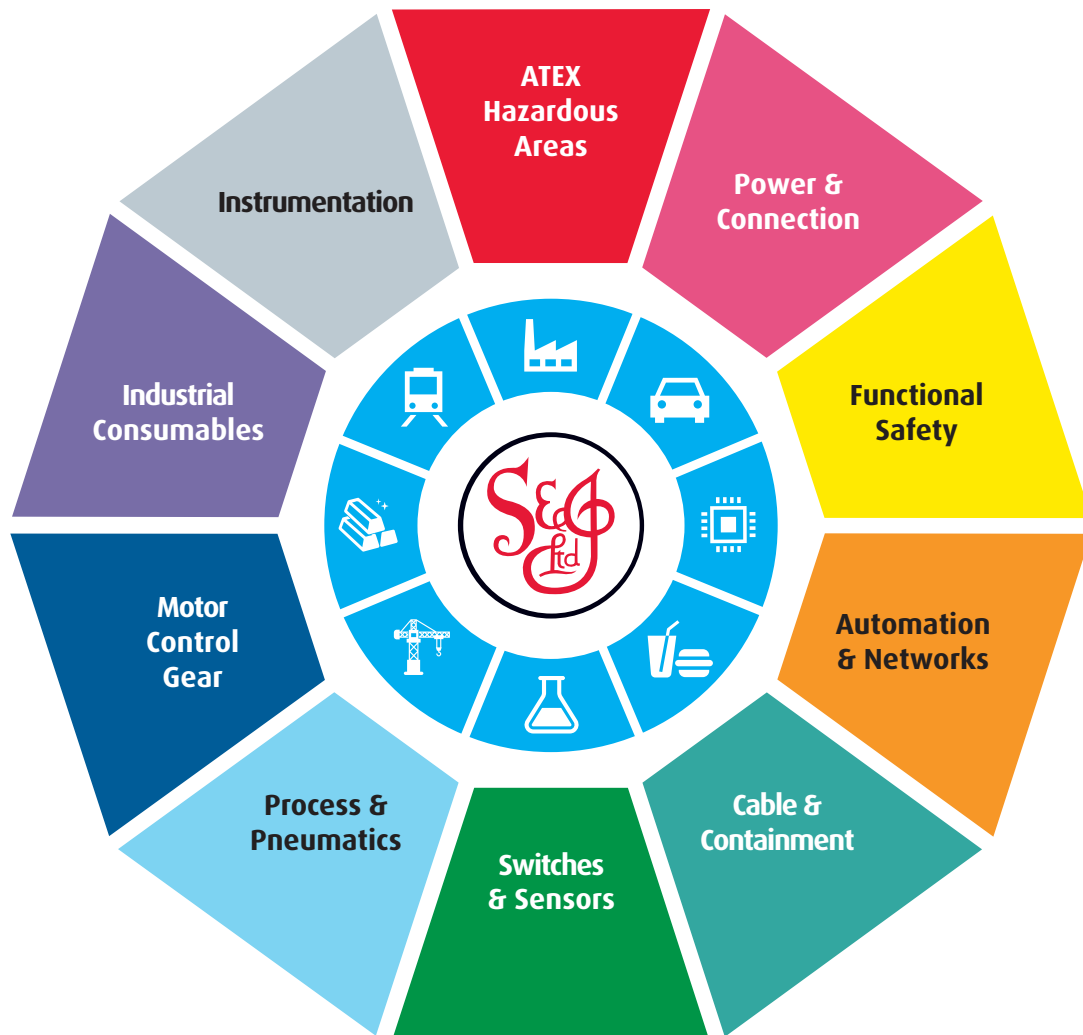
At Scattergood & Johnson Ltd, we pride ourselves on being a technical distributor to specialist industries.

Working with a range of quality product suppliers across a number of specialist markets, we are not your average 'box shifter' - we are your technical and supply chain partner.

We fully support every product we sell - for free! Our internal team and external sales engineers can answer any product or application question, no matter the complexity.

Backing up this technical ability is a range of 50,000+ products available from stock for nationwide next day delivery (same day if required!), or you can collect what you need from any of our trade counters around the UK.

Select your specialist interest below to learn more about how we can help.



Online, In Branch and On the Road - Scattergood & Johnson Ltd, there when you need us.

# [www.scatts.co.uk](http://www.scatts.co.uk)