OMRON

Machine Automation Controller

NJ-series

Robot Integrated CPU Unit

User's Manual

NJ501-R520
NJ501-R500
NJ501-R420
NJ501-R400
NJ501-R320
NJ501-R300

CPU Unit

Sysmac
always in control

O037-E1-03

Introduction

Thank you for purchasing an NJ-series Robot Integrated CPU Unit.

This manual contains information that is necessary to use the robot control function of the NJ-series CPU Unit. Please read this manual and make sure you understand the functionality and performance of the NJ-series CPU Unit before you attempt to use it in a control system.
Keep this manual in a safe place where it will be available for reference during operation.
This manual describes the functions added to the NJ501-R□□□ CPU Unit.
Refer to the *NJ/NX-series CPU Unit Software User's Manual (Cat. No. W501)* and the *NJ/NX-series Database Connection CPU Units User's Manual (Cat. No. W527)* for information on the common functions for the NJ501-□□□□ CPU Unit and the functions for the NJ501-□□20 CPU Unit.

Intended Audience

This manual is intended for the following personnel, who must also have knowledge of electrical systems (an electrical engineer or the equivalent).
- Personnel in charge of introducing FA systems.
- Personnel in charge of designing FA systems.
- Personnel in charge of installing and maintaining FA systems.
- Personnel in charge of managing FA systems and facilities.

Also, this manual is intended for the personnel, who understand the following contents.
- Personnel who understand the programming language specifications in international standard IEC 61131-3 or Japanese standard JIS B 3503, for programming.
- Personnel in charge of working with a robot and well knowing how to handle the robot.

Applicable Products

This manual covers the following products.
- NJ-series Robot Integrated CPU Unit
  NJ501-R520
  NJ501-R500
  NJ501-R420
  NJ501-R400
  NJ501-R320
  NJ501-R300

Relevant Manuals

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

Most operations are performed from the Sysmac Studio Automation Software. Refer to the *Sysmac Studio Version 1 Operation Manual (Cat. No. W504)* and the *Sysmac Studio Robot Integrated System Building Function with Robot Integrated CPU Unit Operation Manual (Cat. No. W595)* for information on the Sysmac Studio.

Purpose of use	Manual												
	Basic information												
	NJ-series CPU Unit Hardware User's Manual	NJ/NX-series CPU Unit Software User's Manual	NJ/NX-series Instructions Reference Manual	NJ/NX-series CPU Unit Motion Control User's Manual	NJ/NX-series Motion Control Instructions Reference Manual	NJ/NX-series CPU Unit Built-in EtherCAT Port User's Manual	NJ/NX-series CPU Unit Built-in EtherNet/IP Port User's Manual	NJ/NX-series Database Connection CPU Units User's Manual	NJ-series Robot Integrated CPU Unit User's Manual	eV+3 User's Manual	eV+3 Keyword Reference Manual	NJ-series NJ Robotics CPU Unit User's Manual	NJ/NX-series Troubleshooting Manual
Introduction to NJ-series Controllers	○												
Setting devices and hardware													
Using motion control				○									
Using EtherCAT	○					○							
Using EtherNet/IP							○						
Using robot control for OMRON robots									○				
Software settings													
Using motion control				○									
Using EtherCAT						○							
Using EtherNet/IP		○					○						
Using the database connection service								○					
Using robot control for OMRON robots									○	○	○		
Using robot control with NJ Robotics function												○	
Writing the user program													
Using motion control				○	○								
Using EtherCAT						○							
Using EtherNet/IP		○	○				○						
Using the database connection service								○					
Using robot control for OMRON robots									○	○	○		
Using robot control with NJ Robotics function												○	
Programming error processing									○	○	○		○

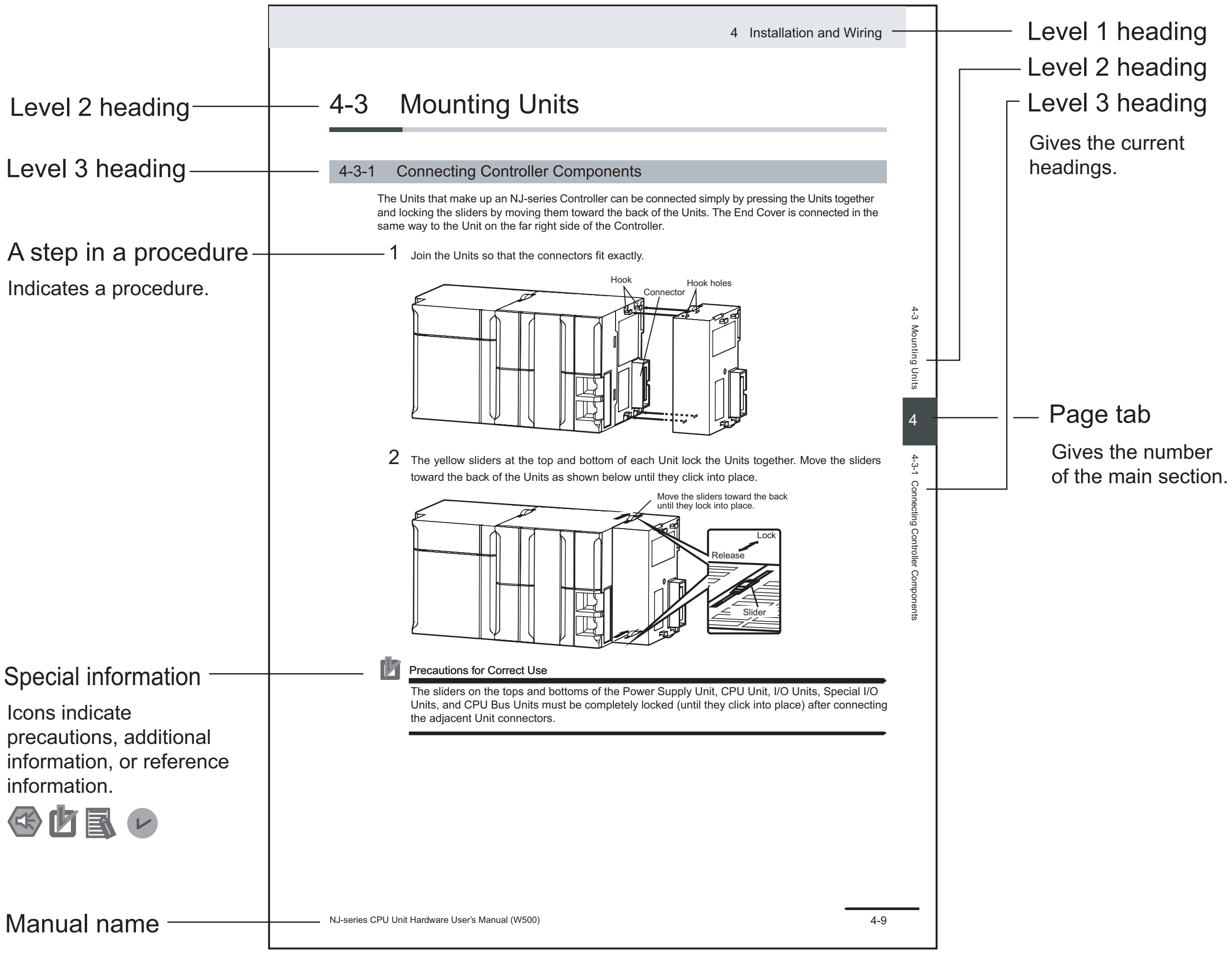
Purpose of use	Manual												
	Basic information												
	NJ-series CPU Unit Hardware User's Manual	**NJ/NX-series CPU Unit Software User's Manual**	**NJ/NX-series Instructions Reference Manual**	**NJ/NX-series CPU Unit Motion Control User's Manual**	**NJ/NX-series Motion Control Instructions Reference Manual**	**NJ/NX-series CPU Unit Built-in EtherCAT Port User's Manual**	**NJ/NX-series CPU Unit Built-in EtherNet/IP Port User's Manual**	**NJ/NX-series Database Connection CPU Units User's Manual**	**NJ-series Robot Integrated CPU Unit User's Manual**	**eV+3 User's Manual**	**eV+3 Keyword Reference Manual**	**NJ-series NJ Robotics CPU Unit User's Manual**	**NJ/NX-series Troubleshooting Manual**
Testing operation and debugging		○											
Using motion control				○									
Using EtherCAT						○							
Using EtherNet/IP							○						
Using the database connection service								○					
Using robot control for OMRON robots									○	○	○		
Using robot control with NJ Robotics function												○	
Learning about error management functions and corrections*1								△	△	△	△	△	○
Maintenance	○												
Using motion control				○									
Using EtherCAT						○							
Using EtherNet/IP							○						

*1. Refer to the *NJ/NX-series Troubleshooting Manual (Cat. No. W503)* for the error management concepts and the error items. However, refer to the manuals that are indicated with triangles for details on errors corresponding to the products with the manuals that are indicated with triangles.

Manual Structure

Page Structure

The following page structure is used in this manual.



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

Special Information

Special information in this manual is classified as follows:

Precautions for Safe Use

Precautions on what to do and what not to do to ensure safe usage of the product.

Precautions for Correct Use

Precautions on what to do and what not to do to ensure proper operation and performance.

Additional Information

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

Version Information

Information on differences in specifications and functionality for Controller with different unit versions and for different versions of the Sysmac Studio is given.

Precaution on Terminology

In this manual, "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

CONTENTS

Section 2 Robot Control System Configuration and Functions

Section 3 Robot Control Parameters

Section 4 Program Design of Robot Control

Section 5 Robot Control Function

Section 6 Robot Control Instructions

Section 7 Variables and Instructions

Section 8 Common Command Instructions

Section 9 Robot Command Instructions

Section 10 System Control Instructions

Section 11 Troubleshooting

Appendices

Index

Terms and Conditions Agreement

Warranty, Limitations of Liability

Warranties

- **Exclusive Warranty**

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

- **Limitations**

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

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

- **Buyer Remedy**

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

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

Limitation on Liability; Etc

OMRON COMPANIES SHALL NOT BE LIABLE FOR SPECIAL, INDIRECT, INCIDENTAL, OR CONSEQUENTIAL DAMAGES, LOSS OF PROFITS OR PRODUCTION OR COMMERCIAL LOSS IN ANY

WAY CONNECTED WITH THE PRODUCTS, WHETHER SUCH CLAIM IS BASED IN CONTRACT, WARRANTY, NEGLIGENCE OR STRICT LIABILITY.

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

Application Considerations

Suitability of Use

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

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

Programmable Products

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

Disclaimers

Performance Data

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

Change in Specifications

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

Errors and Omissions

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

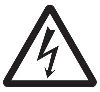
Safety Precautions

Definition of Precautionary Information

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

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

Symbols

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

WARNING

⚠ WARNING

Refer to the following manuals for warnings.

- *NJ-series CPU Unit Hardware User's Manual (Cat. No. W500)*

Designing Systems

When you build a robot system including this CPU Unit and OMRON robots, be sure to comply with laws and regulations relating to the safety of industrial robot application in a country or region where the robots are used to design and operate the system.
Refer to the *Robot Safety Guide (Cat. No. I590)* for details.

Operation

If you change the operating mode of this CPU Unit from RUN Mode to PROGRAM Mode, the sequence control program stops, but the current V+ program continues. If necessary, monitor the operating mode of the CPU Unit from the V+ program and stop the V+ program.
Always confirm safety when you change the operating mode of the CPU Unit during execution of the V+ program. If you cannot confirm safety, stop the V+ program and then change the operating mode.

Cautions

⚠ Caution

Refer to the following manuals for cautions.

- *NJ-series CPU Unit Hardware User's Manual (Cat. No. W500)*
- *NJ-series NJ Robotics CPU Unit User's Manual (Cat. No. W539)*

Designing Programs

There are different methods to attach a robot from the sequence control program and the V+ program. In addition, when a robot is attached from a program, the robot cannot be attached from another program without detaching the robot.
If the same OMRON robot is controlled by switching the sequence control program or the V+ program respectively, make sure to detach the robot from the program that the robot is attached, and then change the control program to attach the robot.

Operation

Do not remove the SD Memory Card during operation when you use the robot control function with this product.
Doing so causes the robot control function to stop due to an error.

The V+ program files and the robot setting files in the SD Memory Card are required for the operation of the Robot Control Function Module. Do not edit or delete the files if you are not sure that the operation is not affected even when the files are edited and deleted.
Always confirm how the file operations affect the control before you perform file operations in the SD Memory Card.

Precautions for Safe Use

Refer to the following manuals for precautions for safe use.

- *NJ-series CPU Unit Hardware User's Manual (Cat. No. W500)*

Motion Control

- The coordinate system used by the Robot Control Function Module have different specifications from the coordinate system used by the NJ Robotics function.
  If you use both functions simultaneously, use the RC_ConvertCoordSystem (Convert Coordinate System) instruction to set the same coordinate system before performing the robot control.
- If the OMRON robot is operated with tracking control, set the parameters for the master machine so that the velocity of the master machine does not change rapidly.
- When the current position for a motion control axis is changed with the MC_Home (Home), MC_HomeWithParameter (Home with Parameters), or MC_SetPosition (Set Position) instruction or the **Axis Use** axis parameter setting is changed with the MC_ChangeAxisUse (Change Axis Use) instruction during tracking control, the tracking operation may change rapidly.
- When the EtherCAT slave assigned to a motion control axis is disconnected or connected, or enabled or disabled during tracking control, the tracking operation may change rapidly.

Precautions for Correct Use

Refer to the following manuals for precautions for correct use.

- *NJ-series CPU Unit Hardware User’s Manual (Cat. No. W500)*
- *NJ/NX-series Database Connection CPU Units User's Manual (Cat. No. W527)*

Designing Programs

- If you create the program to use with the sequence control program and the V+ program, design the interlocks between the programs with shared variables.

Regulations and Standards

Refer to the following manuals for regulations and standards.

- *NJ-series CPU Unit Hardware User's Manual (Cat. No. W500)*

Additional Information

The Robot Integrated CPU Unit is not a robot control device that is defined in ISO 10218-1. Therefore, the Robot Integrated CPU Unit does not comply with the robot regulations and standards.
Refer to the OMRON robot manuals for information on the OMRON robot itself.

Versions

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

Checking Versions

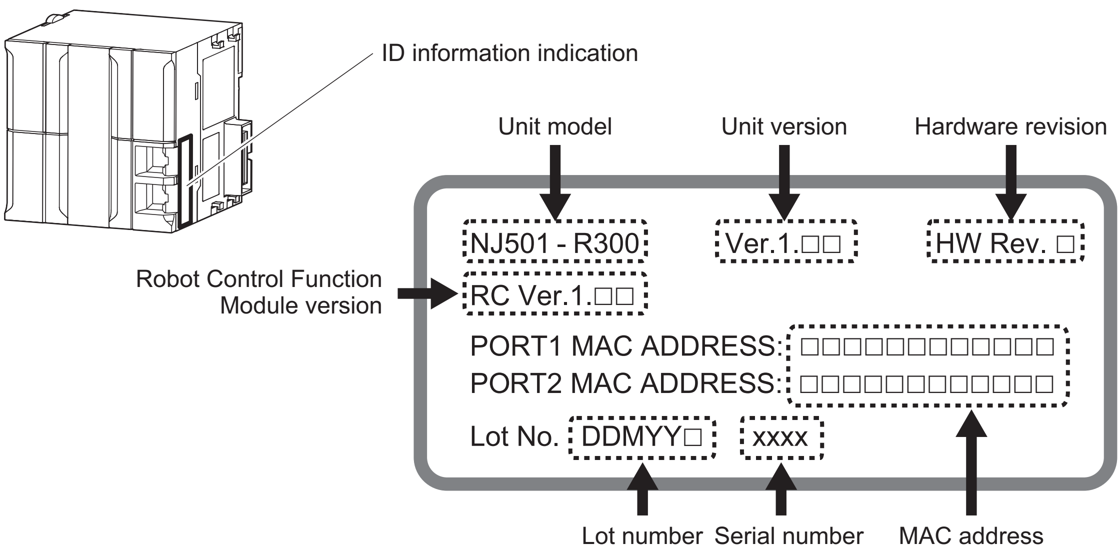
You can check versions on the ID information indications or with the Sysmac Studio.

Checking Unit Versions on ID Information Indications

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

● **NJ501-R□00**

The ID information on the NJ-series NJ501-R300 CPU Unit is shown below.



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

Checking Unit Versions with the Sysmac Studio

You can use the Sysmac Studio to check unit versions.

● Checking the Unit Version of an NJ-series CPU Unit

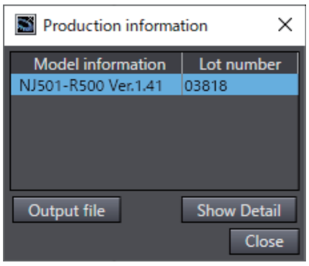
You can use the Production Information while the Sysmac Studio is online to check the unit version of a Unit. You can do this for the CPU Unit, CJ-series Special I/O Units, and CJ-series CPU Bus Units. You cannot check the unit versions of CJ-series Basic I/O Units with the Sysmac Studio.

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

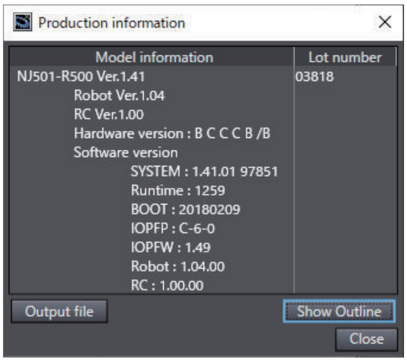
2 Right-click any open space in the Unit Editor and select **Production Information**.
The Production Information Dialog Box is displayed.

● Changing Information Displayed in Production Information Dialog Box

1 Click the **Show Detail** or **Show Outline** Button at the lower right of the **Production Information** Dialog Box.
The view will change between the **production information** details and outline.



Outline View



Detailed View

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

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

Related Manuals

The following are the manuals related to this manual. 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.	An introduction to the entire NJ-series system is provided along with the following information on the CPU Unit. • Features and system configuration • Introduction • Part names and functions • General specifications • Installation and wiring • Maintenance and inspection
NJ/NX-series CPU Unit Software User's Manual	W501	NX701-□□□□ NX102-□□□□ NX1P2-□□□□ NJ501-□□□□ NJ301-□□□□ NJ101-□□□□	Learning how to program and set up an NJ/NX-series CPU Unit. Mainly software information is provided.	The following information is provided on a Controller built with an NJ/NX-series CPU Unit. • CPU Unit operation • CPU Unit features • Initial settings • Programming based on IEC 61131-3 language specifications
NJ/NX-series Instructions Reference Manual	W502	NX701-□□□□ NX102-□□□□ NX1P2-□□□□ NJ501-□□□□ NJ301-□□□□ NJ101-□□□□	Learning detailed specifications on the basic instructions of an NJ/NX-series CPU Unit.	The instructions in the instruction set (IEC 61131-3 specifications) are described.
NJ/NX-series CPU Unit Motion Control User's Manual	W507	NX701-□□□□ NX102-□□□□ NX1P2-□□□□ NJ501-□□□□ NJ301-□□□□ NJ101-□□□□	Learning about motion control settings and programming concepts.	The settings and operation of the CPU Unit and programming concepts for motion control are described.
NJ/NX-series Motion Control Instructions Reference Manual	W508	NX701-□□□□ NX102-□□□□ NX1P2-□□□□ NJ501-□□□□ NJ301-□□□□ NJ101-□□□□	Learning about the specifications of the motion control instructions.	The motion control instructions are described.
NJ/NX-series CPU Unit Built-in EtherCAT® Port User's Manual	W505	NX701-□□□□ NX102-□□□□ NX1P2-□□□□ NJ501-□□□□ NJ301-□□□□ NJ101-□□□□	Using the built-in EtherCAT port on an NJ/NX-series CPU Unit.	Information on the built-in EtherCAT port is provided. This manual provides an introduction and provides information on the configuration, features, and setup.
NJ/NX-series CPU Unit Built-in EtherNet/IP™ Port User's Manual	W506	NX701-□□□□ NX102-□□□□ NX1P2-□□□□ NJ501-□□□□ NJ301-□□□□ NJ101-□□□□	Using the built-in EtherNet/IP port on an NJ/NX-series CPU Unit.	Information on the built-in EtherNet/IP port is provided. Information is provided on the basic setup, tag data links, and other features.
NJ-series Robot Integrated CPU Unit User's Manual	O037	NJ501-R□□□	Using the NJ-series Robot Integrated CPU Unit.	Describes the settings and operation of the CPU Unit and programming concepts for OMRON robot control.

Manual name	Cat. No.	Model numbers	Application	Description
Sysmac Studio Robot Integrated System Building Function with Robot Integrated CPU Unit Operation Manual	W595	SYSMAC-SE2□□□ SYSMAC-SE200D-64	Learning about the operating procedures and functions of the Sysmac Studio to configure Robot Integrated System using Robot Integrated CPU Unit.	Describes the operating procedures of the Sysmac Studio for Robot Integrated CPU Unit.
Sysmac Studio Robot Integrated System Building Function with IPC Application Controller Operation Manual	W621	SYSMAC-SE2□□□ SYSMAC-SE200D-64	Learning about the operating procedures and functions of the Sysmac Studio to configure Robot Integrated System using IPC Application Controller.	Describes the operating procedures of the Sysmac Studio for IPC Application Controller.
Sysmac Studio 3D Simulation Function Operation Manual	W618	SYSMAC-SE2□□□ SYSMAC-SA4□□□-64	Learning about an outline of the 3D simulation function of the Sysmac Studio and how to use the function.	Describes an outline, execution procedures, and operating procedures for the 3D simulation function of the Sysmac Studio.
eV+3 User's Manual	I651	NJ501-R□□□	Operating the OMRON robot with the V+ program.	Describes the V+ language to control the OMRON robots.
eV+3 Keyword Reference Manual	I652	NJ501-R□□□	Operating the OMRON robot with the V+ program.	Describes V+ keywords that are used in the V+ language.
eCobra 600 and 800 Robot with EtherCAT User's Guide	I653	RL4-□□□□□□□	Using the eCobra.	Describes the eCobra.
i4H Robots with EtherCAT User's Manual	I661	RS4-□□□□□□□	Using the i4H.	Describes the i4H.
Viper 650 and 850 Robot with EtherCAT User's Guide	I654	RL6-□□□□□□□	Using the Viper.	Describes the Viper.
iX3 565 Robot with EtherCAT User's Guide	I655	RX3-□□□□□□□	Using the iX3.	Describes the iX3.
iX4 650 H/HS and 800 H/HS Robot with EtherCAT User's Guide	I656	RX4-□□□□□□□	Using the iX4.	Describes the iX4.
Robot Safety Guide	I590	RL4-□□□□□□□ RS4-□□□□□□□ RL6-□□□□□□□ RX3-□□□□□□□ RX4-□□□□□□□	Learning how to use the OMRON robot safely.	Describes how to use the OMRON robot safely.
Teaching Pendant T20 User's Manual	I601	10046-010	Operating the OMRON robot with a teaching pendant.	Describes the setup, operation, and user maintenance for the Teaching Pendant T20.
IPC Application Controller User's Manual	I632	AC1-152000	Using the IPC Application Controller.	Describes the IPC Application Controller.
NJ/NX-series Database Connection CPU Units User's Manual	W527	NX701-□□20 NX102-□□20 NJ501-□□20 NJ101-□□20	Using the database connection service with NJ/NX-series Controllers.	Describes the database connection service.
NJ-series NJ Robotics CPU Unit User's Manual	W539	NJ501-4□□□ NJ501-R□□□	Controlling robots with NJ-series CPU Units.	Describes the functionality to control robots.

Manual name	Cat. No.	Model numbers	Application	Description
NJ/NX-series Troubleshooting Manual	W503	NX701-□□□□ NX102-□□□□ NX1P2-□□□□ NJ501-□□□□ NJ301-□□□□ NJ101-□□□□	Learning about the errors that may be detected in an NJ/NX-series Controller.	Concepts on managing errors that may be detected in an NJ/NX-series Controller and information on individual errors are described.
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.	The hardware, setup, and functions for the NX-series Incremental Encoder Input Units, SSI Input Units, and Pulse Output Unit are described.
AC Servomotors/Servo Drives 1S-series with Built-in EtherCAT® Communications User's Manual	I586	R88M-1□ R88D-1SN□-ECT	Learning how to use the Servomotors/ Servo Drives with built-in EtherCAT Communications.	Describes the hardware, setup methods and functions of the Servomotors/Servo Drives with built-in EtherCAT Communications.
	I621	R88M-1AL□/ -1AM□ R88D-1SAN□-ECT		
AC Servomotors/Servo Drives G5 Series with Built-in EtherCAT® Communications User's Manual	I576	R88M-K□ R88D-KN□-ECT	Learning how to use the AC Servomotors/ Servo Drives with built-in EtherCAT Communications.	Describes the hardware, setup methods and functions of the AC Servomotors/ Servo Drives with built-in EtherCAT Communications. The Linear Motor Type models and dedicated models for position control are available in G5-series.
	I577	R88L-EC-□ R88D-KN□-ECT-L		

Terminology

This section describes the terms that are used in this manual.

Term	Description
continuous-path motion	A motion to move continuous operations smoothly without stopping motion of the OMRON robot.
IEC 61131-3 language	A programming language to write a sequence control program.
robots controllable by NJ Robotics function	Specify the controllable robots by the data processing for robot in the Motion Control Function Module of the NJ-series CPU Unit. The controllable robot consists of the 1S-series or G5-series Servomotor/Servo Drive with built-in EtherCAT communications and the robot arm that is prepared by the customer.
TCP	A tip (Tool Center Point) defined in each OMRON robot. The target position or path can be specified based on the TCP.
V+ keyword	A generic term for instructions that are used during a V+ program and monitoring command.
V+ language	A programming language for OMRON robot control.
V+ task	A task that can execute a V+ program.
V+ program	A control program written in the V+ language.
OMRON robot	Specifies the OMRON robot controllable from the Robot Integrated CPU Unit. The robot consists of the robot amplifier and the robot arm connected to the robot amplifier.
shared variable	A variable that can be shared between the sequence control program and V+ program.
sequence control program	A control program written in IEC 61131-3 language including the motion control.
hardware servo	A servo system built into the robot amplifier.
user program	A generic term for the collection of programs written in the ladder diagram, ST, and V+ languages.
remote encoder	Specifies the encoder which set the motion control axis as the external encoder for robot control.
recipe	A set of product type data in the customer’s system.
recipe change	Specifies that the product data and information (recipe) related to the production process are changed. The target recipe for the Robot Integrated CPU Unit is a property from the present values of variables and a vision sensor.
local encoder	Specifies the encoder connected to the encoder input port on the OMRON robot.
Robot Control Function Module	A software to perform robot control that is installed in the Robot Integrated CPU Unit.
robot control instructions	FB instructions written in the sequence control program to control the OMRON robots. They include an instruction to directly control the OMRON robots and an instruction to execute or abort V+ programs assigned to the V+ tasks.
Robot Integrated CPU Unit	A CPU Unit that supports control function for the OMRON robot with the NJ-series CPU Unit.

Revision History

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

Cat. No. O037-E1-03

Revision code

Revision code	Date	Revised content
01	August 2020	Original production
02	December 2020	• Made changes accompanying the addition of NJ501-R□20. • Made changes accompanying the addition of parallel robots (iX3 and iX4) and a SCARA robot (i4H). • Corrected mistakes.
03	April 2021	• Made changes accompanying addition of functions. • Corrected mistakes.

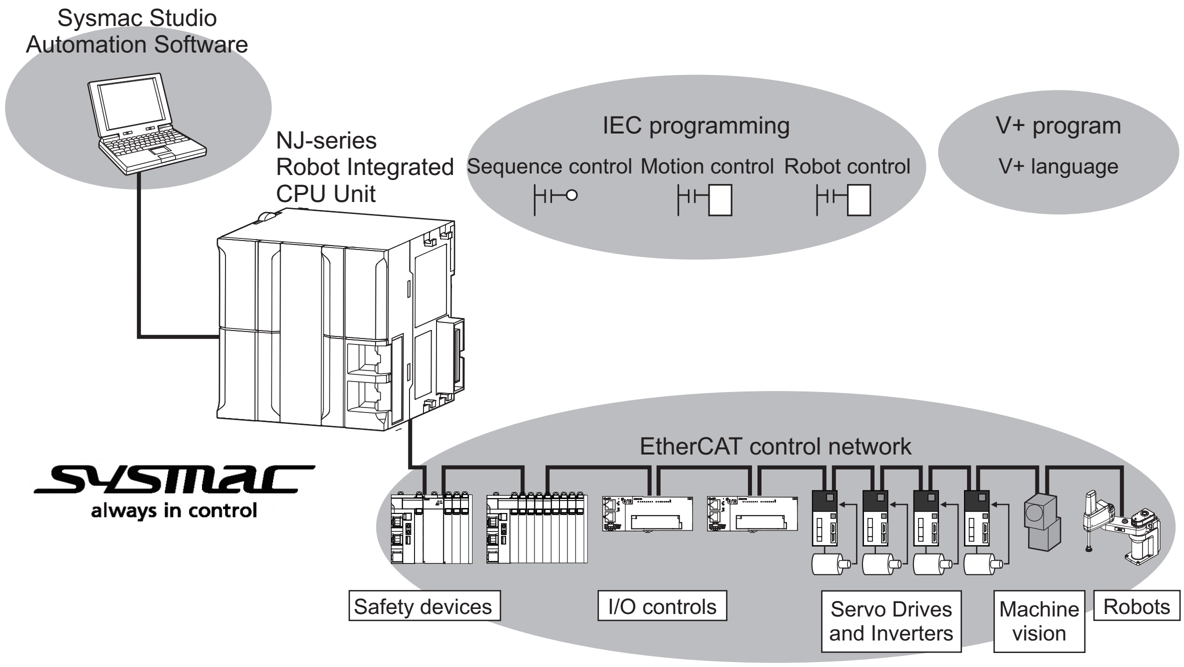
1

Introduction to Robot Integrated CPU Unit

This section describes the features, basic system configuration, specifications, and overall operating procedure of an NJ-series Robot Integrated CPU Unit.

1-1 Features

The NJ-series Robot Integrated CPU Units 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 robot control. Synchronized control of I/O devices on high-speed EtherCAT can be applied to robots, 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 Robot Integrated CPU Units 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 commission the system through unified concepts and usability.



Robot Control Function Module

The Robot Control Function Module (sometimes abbreviated to “RC Function Module”) is a software function module that is built into the Robot Integrated CPU Unit.
The RC Function Module can perform robot control for up to 8 OMRON robots through the built-in EtherCAT port on the Robot Integrated CPU Unit.
Cyclic communications are performed with OMRON robots, Servo Drives, and other devices that are connected to the EtherCAT port to enable tracking control with OMRON robots and axes.

Sequence Control Program

The OMRON robots are controlled with robot control instructions in the sequence control program. The sequence control program includes function blocks that are operated directly to the OMRON robots such as the robot joint interpolation, robot linear interpolation, stopping a robot, and other operations. The program also includes function blocks that controls V+ programs such as executing V+ tasks, aborting V+ tasks, and other operations.

V+ Program

The OMRON robots are controlled using V+ programs. The V+ programs are written in the V+ language that is a special language for the robot control. You can easily realize various operation of the OMRON robot using V+ programs.
In addition, the V+ program can use the interlock with a sequence control program (ladder diagram and ST language) using shared variables.

Integrated Sequence Control and Motion Control

A 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.

Programming Languages Based on the IEC 61131-3 International Standard

The 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.

Kinematics Function Supported

The kinematics function (NJ Robotics function) can perform data processing for robot in the Motion Control Function Module (sometimes abbreviated to “MC Function Module”) to control robots that use parallel link mechanism, Cartesian robots, and SCARA robots that are prepared by the customer.
Refer to the *NJ-series NJ Robotics CPU Unit User's Manual (Cat. No. W539)* for details on the kinematics function.
Note that the "kinematics function" is written as the "NJ Robotics function" in this manual.

Database Connection Service Supported (NJ501-R□20 Only)

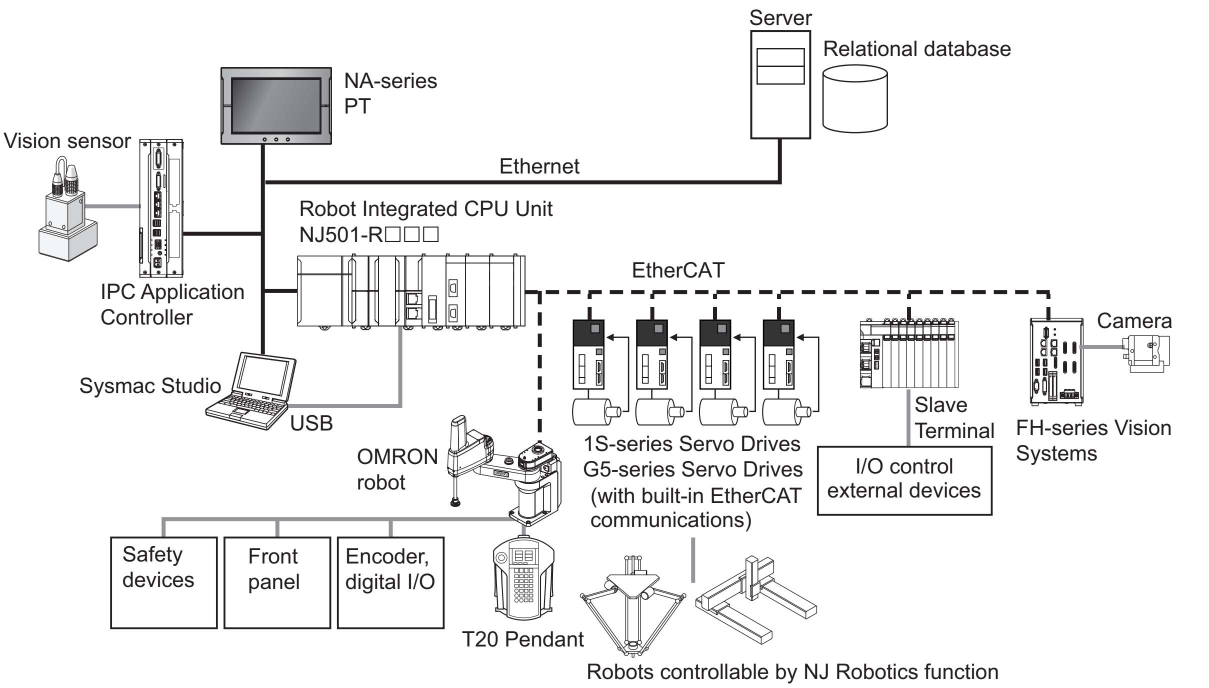
The database connection service provides the functionality to insert, update, read or delete records for the relational database on the server connected to the built-in EtherNet/IP ports. To do so, a special instruction must be executed in the Robot Integrated CPU Unit.
Moreover, secure communications are possible using encryption communications.
You are not required to use a unique Unit or tool for the database connection function. The Robot Integrated CPU Unit and Sysmac Studio are used for this function.
In addition, a middleware for the connection to NJ-series Controllers with a server is not required.
Refer to the *NJ/NX-series Database Connection CPU Units User's Manual (Cat. No. W527)* for details on the database connection service.

Data Transmission Using EtherCAT Communications

The OMRON robots are connected with EtherCAT communications to enable exchange of all control information with high-speed data communications.
In addition, cyclic communications are performed with OMRON robots, Servo Drives and other devices with EtherCAT communications, and the performance for the entire equipment is maximized.

1-2 System Configuration

This section describes the system configuration and components related to the Robot Integrated CPU Unit.



The function of each component in the system is given below.

Component	Function in the system
EtherCAT	Controls for Servo Drives and OMRON robots through the EtherCAT master port that is built into the Robot Integrated CPU Unit. It enables precise control in a fixed period with very little deviation.
OMRON robot	Consists of the robot amplifier and the robot arm connected to the robot amplifier. It connects with a Robot Integrated CPU Unit through EtherCAT communications. It has digital I/O ports to enable control for the external devices.
T20 Pendant *1	A teaching pendant for the OMRON robot. It connects to the OMRON robot and performs a test run for the OMRON robot or teaching.
Sysmac Studio	An integrated development software for use with the Robot Integrated CPU Unit that allows you to create programming and make device settings.
NA-series PT	Displays various information and performs operation as required. It is used when you instruct a recipe change to the Robot Integrated CPU Unit.
IPC Application Controller	A Controller to manage a recipe and more than one OMRON robot controlled by the Robot Integrated CPU Unit. It can perform image processing by connecting a vision sensor.
1S-series Servo Drives G5-series Servo Drives	Servo Drives with built-in EtherCAT communications.
Robots controllable by NJ Robotics function	Robots that can be controlled from the Robot Integrated CPU Unit that controls Servomotors/Servo Drives with built-in EtherCAT communications.

Component	Function in the system
Slave Terminal	Consists of the NX-ECC20□ Communications Coupler Unit and NX Units that are connected to EtherCAT communications. It exchanges I/O data with a Robot Integrated CPU Unit through EtherCAT communications. Various Units such as digital I/O, analog I/O are covered, therefore, you can use the NX Units depending on the system demand.
Relational database	A database management system. For the Robot Integrated CPU Unit (NJ501-R□20), it supports connection with Oracle Database, Microsoft SQL Server, MySQL, FireBird, DB2, and PostgreSQL.
FH-series Vision Systems	Vision systems connected to the EtherCAT communications.
Front panel	Changes the operating mode of OMRON robot and executes a emergency stop.
I/O control external devices	External devices to control from I/O ports of the NX Units or the robot. They include a photoelectric sensor, an air cylinder, a robot hand, and other devices.
Safety devices	Safety devices such as a Safety Controller, safety sensor, and safety relay.

*1. Refer to the *Teaching Pendant T20 User's Manual (Cat. No. I601)* for details.

Precautions for Correct Use

- Always insert the included SD Memory Card when you use the robot control function with the Robot Integrated CPU Unit.
- Do not remove the SD Memory Card while power is supplied when you use the robot control function with the Robot Integrated CPU Unit.
  Doing so causes the robot control function to stop due to an error.
- The robot setting files and V+ program files in the SD Memory Card are required for the operation of the Robot Control Function Module.
  Be careful not to overwrite or delete the robot setting files and V+ program files.

1-3 Specifications

This section describes the general specifications, performance specifications, and function specifications for the Robot Integrated CPU Unit.

1-3-1 General Specifications

General specifications conform to the general specifications of the CPU Unit.
Refer to the *NJ-series CPU Unit Hardware User's Manual (Cat. No. W500)* for details.

1-3-2 Performance Specifications

Performance specifications conform to the performance specifications of the NJ-series Controllers.
Refer to the *NJ-series CPU Unit Hardware User's Manual (Cat. No. W500)* for details.

The performance specifications for the Robot Integrated CPU Unit are described below.

(○: Supported, ×: Not supported)

Item	NJ501-					
	R520	R500	R420	R400	R320	R300
Maximum number of controlled axes*1	64 axes		32 axes		16 axes	
Maximum number of controllable OMRON robots *2	8 max.					
Maximum number of robots controllable by NJ Robotics function *2*3	8 max.					
Database connection service *4	○	×	○	×	○	×

*1. This is the total for all axis types in Motion Control Function Module. Refer to the *NJ/NX-series CPU Unit Motion Control User's Manual (Cat. No. W507)* for details on axis types.
This includes the number of axes used for the robots that are controlled with the NJ Robotics function.

*2. When the number of connected devices increases, the number of devices that can be connected to EtherCAT network is limited due to the restrictions for process data size of EtherCAT communications.

*3. The number of robots controllable by NJ Robotics function varies depending on the number of controlled axes that you use. The number of controlled axes depends on the robot type. Refer to the *NJ-series NJ Robotics CPU Unit User's Manual (Cat. No. W539)* for details.

*4. Refer to the *NJ/NX-series Database Connection CPU Units User's Manual (Cat. No. W527)* for information on the specifications of the database connection service.

1-3-3 Function Specifications

This section describes the functions that are specific to the NJ-series Robot Integrated CPU Units.

Basic Function

Category	Function	Specification	Reference
Tasks	Task management	Adds V+ tasks along with the function supported by the NJ-series CPU Units.	page 2-12
Programming	V+ program execution	Executes the V+ program.	page 1-9

Category	Function	Specification	Reference
	Shared variables for sequence control program and V+ program	Shares variables between the sequence control program and the V+ program.	page 4-7
	V+ task control from sequence control program	Controls V+ tasks such as execution, end, and other operations from the sequence control program.	page 4-6
	System-defined variables for robot control	Can use the variables for monitoring the state of the RC Function Module or each OMRON robot and variables for reading and writing the built-in I/O in the each OMRON robot.	page 2-8
	I/O controls	Read and write data for the digital I/O of the EtherCAT slave devices connected with EtherCAT communications or NX Units on the EtherCAT Coupler Unit from the V+ program.	page 3-4
Motion control	Control with NJ Robotics function	Controls Servomotors/Servo Drives connected with the EtherCAT communications to control robots. The function can realize the robot operation, single-axis operation, and synchronized operation with the Motion Control Function Module.	page 2-4
Troubleshooting	Event log	Manages the event log for the OMRON robot along with the function supported by the NJ-series CPU Units.	page 11-1
	Error management	Manages the OMRON robot errors along with the function supported by the NJ-series CPU Units.	
	Error reset	Clears the OMRON robot errors along with the function supported by the NJ-series CPU Units.	
Security	Robot System Operation Authority	Adds functions along with the function supported by the NJ-series CPU Units.	page 2-29
	CPU Unit write-protection		page 2-29
SD Memory Card	File explorer	Uses the V+ File Browser in the Sysmac Studio to access V+ programs and other data in the Robot Integrated CPU Unit.	page 2-17
Memory management	Clear All Memory	Includes robot setting files and V+ programs along with the function supported by the NJ-series CPU Units.	page 2-22
Backup	Backing up and restoring data for the Robot Integrated CPU Unit	Includes robot setting files and V+ programs along with the function supported by the NJ-series CPU Units.	page 2-24
	Backing up and restoring data for the OMRON robot	Backs up or restores data related to the OMRON robot.	page 2-28

Function for Robot Control Function Module

Category	Function		Specification	Reference
Robot control	Control program for OMRON robot		Controls the OMRON robots with the sequence control program and V+ program.	page 5-2
	Tracking control	Local encoder (Master) - Robot (Slave)	Performs the tracking control for the local encoder as the master machine and the same OMRON robot as the slave machine. The function executes in the V+ program.	page 5-5

Category	Function		Specification	Reference
		Axis (Master) - Robot (Slave)	Performs the tracking control for the motion control axis in the MC Function Module as the master machine and the OMRON robot as the slave machine. The function executes in the V+ program.	page 5-5
	Latching	Robot position latching	Inputs an external trigger signal to the latch input of the OMRON robot to output the current position.	page 5-6
		Local encoder latching	Uses the latch function of the OMRON robot to identify the encoder counter value at imaging by a vision sensor.	page 5-7
		Remote encoder latching	Uses the digital signal assigned to I/O in the V+ program as shown in *3-2-2 V+Digital I/O Settings* on page 3-4 as the latch signal and reads the position information of the motion control axis from the V+ program when the latch signal occurs.	page 5-9
	Path compensation	Cyclic path compensation	Compensates a path of the robot that is controlled in the V+ program.	page 5-10

Additional Information

Refer to the *Sysmac Studio Robot Integrated System Building Function with Robot Integrated CPU Unit Operation Manual (Cat. No. W595)* for information on the debugging that is specific to the Robot Control Function Module.
Refer to the *Sysmac Studio Version 1 Operation Manual (Cat. No. W504)* for information on the debugging that is common to the NJ-series CPU Unit.

● Communications Port List

The following table shows the list of communications ports that are used by the Robot Control Function Module.

Service	Type	Port number	Remark
Vision Client	UDP	1989	Fixed value
Controller scan	UDP	1992	Fixed value
ActiveV	TCP/UDP	1997	Fixed value
TFTP Client	TCP	2345	Fixed value
ACE communication	TCP	43234	Fixed value

IPC Application Controller Cooperation Function

Function	Specification	Reference
Image processing with a vision sensor	Detects and inspects a workpiece using a vision sensor.	page 2-6
Conveyor tracking	Synchronizes the OMRON robot and the belt conveyor using the encoder that is mounted to the belt conveyor.	page 2-7
Recipe change from CPU Unit	Requests a recipe change from the Robot Integrated CPU Unit to the IPC Application Controller.	page 2-7

1-3-4 V+ Program Specifications

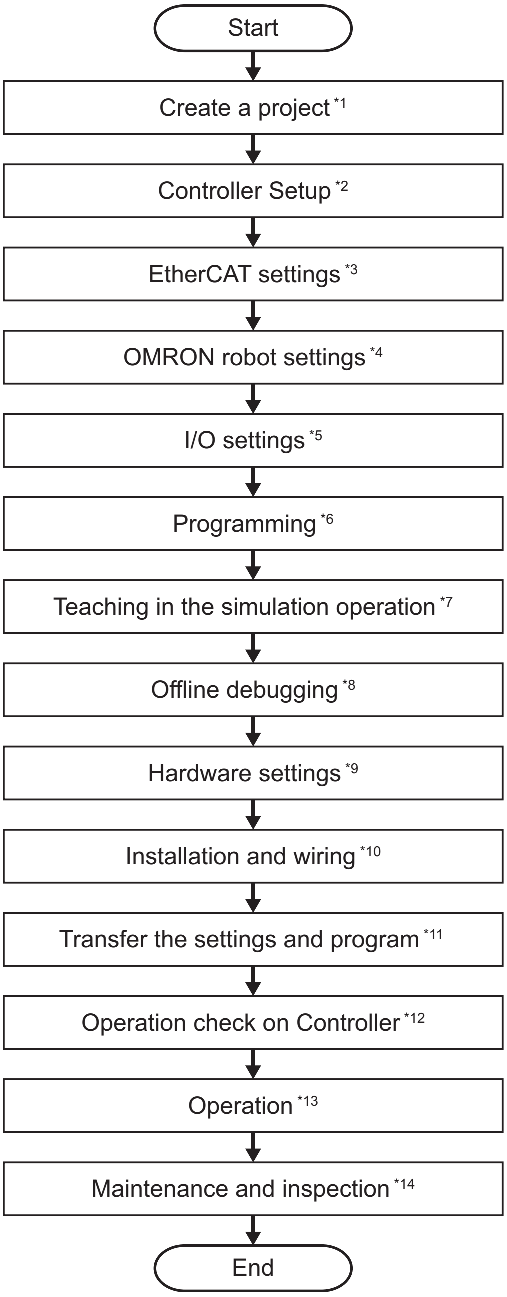
Refer to the *eV+3 User's Manual (Cat. No. I651)* and the *eV+3 Keyword Reference Manual (Cat. No. I652)* for information on the V+ program specifications.

Precautions for Correct Use

The V+ keywords using for a hardware that does not support in the Robot Integrated CPU Unit, such as DeviceNet, graphics, and IEEE1394, are not supported.

1-4 Basic Procedure of Operation

This section describes an example of the procedure to realize applications using the OMRON robots.



Symbol	Description	Reference
*1	Create a new project for the Robot Integrated CPU Unit in the Sysmac Studio.	Sysmac Studio Version 1 Operation Manual (Cat. No. W504)
*2	Make the settings for the Robot Integrated CPU Unit.	*2-5 Tasks* on page 2-12
*3	Add the EtherCAT slaves including the OMRON robots on the EtherCAT, and make the settings.	*2-6 EtherCAT Communications and Robot Control* on page 2-15

Sym-bol	Description	Reference
*4	Make the robot control parameter settings.	*3-3 Robot Setting Parameters* on page 3-9
*5	Make the setting to control I/O from the user program.	*3-2 Robot Common Parameters* on page 3-4
*6	Write the user program for OMRON robots and peripheral devices with the sequence control program and the V+ program.	• *Section 4 Program Design of Robot Control* on page 4-1 • *Section 5 Robot Control Function* on page 5-1 • *Section 6 Robot Control Instructions* on page 6-1 • *Section 7 Variables and Instructions* on page 7-1 • *Section 8 Common Command Instructions* on page 8-1 • *Section 9 Robot Command Instructions* on page 9-1 • *Section 10 System Control Instructions* on page 10-1
*7	Perform teaching in the simulation operation.	• Sysmac Studio 3D Simulation Function Operation Manual (Cat. No. W618) • Sysmac Studio Robot Integrated System Building Function with Robot Integrated CPU Unit Operation Manual (Cat. No. W595)
*8	Verify operation in the simulation operation. Make changes as required.	• *4-4-1 Offline Debugging* on page 4-16 • Sysmac Studio 3D Simulation Function Operation Manual (Cat. No. W618) • Sysmac Studio Robot Integrated System Building Function with Robot Integrated CPU Unit Operation Manual (Cat. No. W595)
*9	Make the settings for the hardware switches on the devices and other settings. If you operate a robot in the simulation operation, you can make the hardware settings later.	Manuals for the OMRON robots that you use and the EtherCAT slaves
*10	Install the devices. Wire the network and the I/O. If you operate a robot in the simulation operation, you can perform installation and wiring later.	• NJ-series CPU Unit Hardware User's Manual (Cat. No. W500) • Manuals for the OMRON robots that you use
*11	Transfer the settings and program from the Sysmac Studio to the Robot Integrated CPU Unit and the EtherCAT slaves.	Sysmac Studio Robot Integrated System Building Function with Robot Integrated CPU Unit Operation Manual (Cat. No. W595)
*12	Check the operations for the program and the teaching position on the physical Controller. Make changes as required.	• *4-4-3 Online Debugging* on page 4-17 • Sysmac Studio Robot Integrated System Building Function with Robot Integrated CPU Unit Operation Manual (Cat. No. W595) • Teaching Pendant T20 User's Manual (Cat. No. I601)
*13	Operate the Controller and the machine.	---
*14	Perform the troubleshooting for the error, periodic inspections, and maintenance.	• *Section 11 Troubleshooting* on page 11-1 • *2-9 Backup and Restore Operations* on page 2-24 • NJ/NX-series Troubleshooting Manual (Cat. No. W503)

Additional Information

Refer to the following manuals for information on the NJ-series common procedure and the procedure and connection with peripheral devices other than the OMRON robot.

- NJ-series CPU Unit Hardware User’s Manual (Cat. No. W500)
- NJ/NX-series CPU Unit Software User’s Manual (Cat. No. W501)
- NJ/NX-series CPU Unit Motion Control User’s Manual (Cat. No. W507)
- NJ-series NJ Robotics CPU Unit User's Manual (Cat. No. W539)
- NJ/NX-series Database Connection CPU Units User's Manual (Cat. No. W527)

2

2

Robot Control System Configuration and Functions

This section outlines the internal structure of the Robot Integrated CPU Unit and describes the configuration and functions of the Robot Control Function Module.

2-1 Internal Configuration for the Robot Integrated CPU Unit

This section provides information about the internal mechanisms of the NJ-series Robot Integrated CPU Unit.

The Robot Integrated CPU Unit has the following software configuration.

The Robot Control Function Module is a software module that performs robot control for the OMRON robots.

Robot Control Function Module	Motion Control Function Module	EtherCAT Master Function Module	Other function modules *1
PLC Function Module			
OS			

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

The PLC Function Module resides on the OS, and each function module is executed. A description of each function module is given in the following table.

Function module	Abbreviation	Description
PLC Function Module	PLC	This module manages overall scheduling, executes the user program, sends commands to the Robot Control Function Module, and provides interfaces to database connection, USB and the SD Memory Card.
Robot Control Function Module*1	RC	This module performs robot control for the OMRON robot according to the commands from robot control instructions that are executed in the user program and the commands from V+ programs. It sends data to the EtherCAT Master Function Module.
Motion Control Function Module *2	MC	This module performs motion control according to the commands from motion control instructions that are executed in the user program. It sends data to the EtherCAT Master Function Module. It is used for the robot control with the NJ Robotics function and the motion control for a transfer equipment and a press machine.
EtherCAT Master Function Module	ECAT	This module communicates with the EtherCAT slaves as the EtherCAT master.

*1. The Robot Control Function Module is sometimes abbreviated to "RC Function Module".

*2. The Motion Control Function Module is sometimes abbreviated to "MC Function Module".

2-2 Relationship between Robot Integrated CPU Unit and Robot

The Robot Integrated CPU Unit supports the RC Function Module and the MC Function Module.

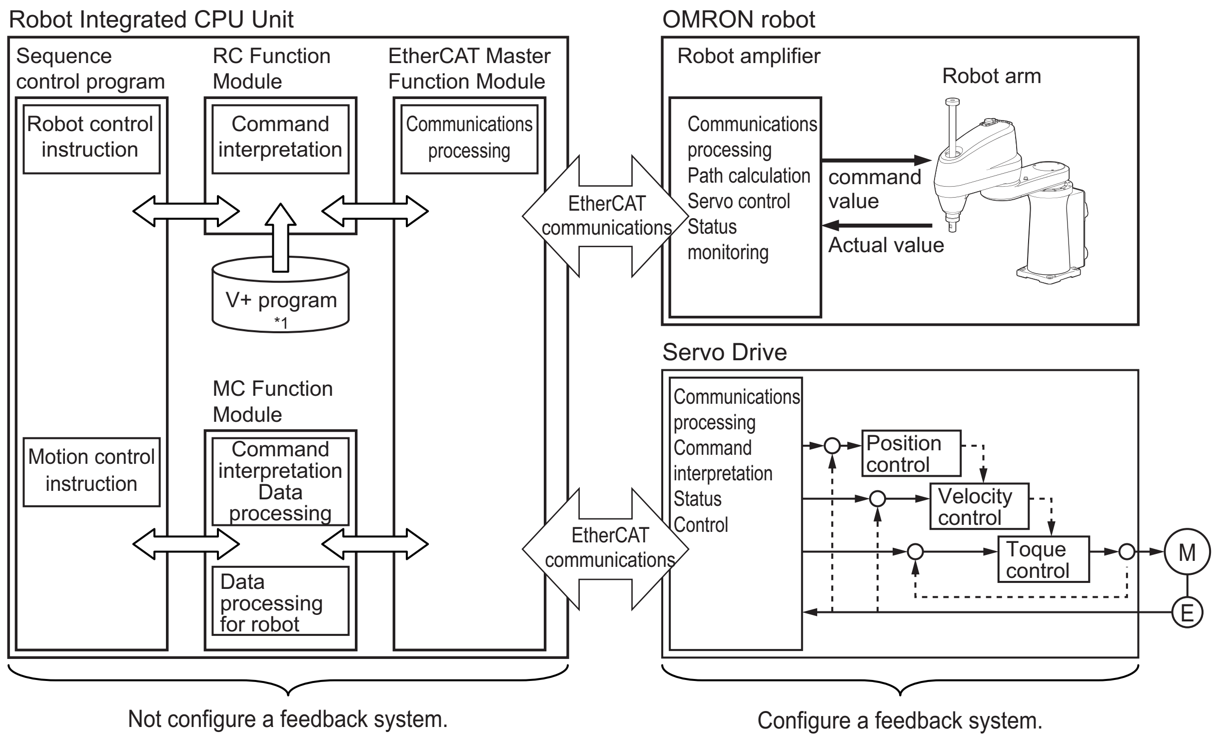
- The RC Function Module can control the OMRON robot through the built-in EtherCAT port from the robot control instructions and V+ programs.
- The MC Function Module can perform motion control through the built-in EtherCAT port on the CPU Unit. Cyclic communications are performed with Servo Drives and other devices that are connected to the EtherCAT port to enable high-speed and high-precision machine control.
  In addition, the NJ Robotics function can perform data processing for robot in the MC Function Module to control robots that use parallel link mechanism, Cartesian robots, and SCARA robots that are prepared by the customer.

A Robot Integrated CPU Unit can control the OMRON robot and the robots that are prepared by the customer.

(○: Controllable, ×: Not controllable)

Robot to control	Robot control instruction	V+ program	Motion control instruction
OMRON robot[*1]	○	○	×
Robots controlled by NJ Robotics function[*2]	×	×	○

*1. Refer to *Controllable OMRON Robots* on page 2-4 for information on the OMRON robot.
*2. Refer to *Robot Types Controllable by NJ Robotics Function* on page 2-5 for information on the robot controllable by the NJ-Robotics function.



*1. The V+ program using the RC Function Module and robot setting files are saved in an SD Memory Card.

Refer to *2-7-1 Included SD Memory Card Functions* on page 2-18 for details.

Robot Control Function Module

A program that controls the OMRON robot is written with robot control instructions and V+ programs.

- The sequence control program can control OMRON robots with robot control instructions only. The program can also control the OMRON robot by starting and stopping the V+ tasks with the robot control instructions.
  You can use robot variables if the current position and status of the OMRON robot are monitored in the sequence control program.
  In the same way, the robot built-in I/O of OMRON robots perform the control with robot I/O variables in the sequence control program.
- In the V+ program, the RC Function Module analyzes the V+ program line by line and sends instructions (V+ keywords) and command values to the OMRON robot connected to the EtherCAT.
  After receipt of an instruction (V+ keyword) or a command value, the OMRON robot performs path calculation and controls the robot arm.
  The present value and status of the OMRON robot are sent through the EtherCAT to the RC Function Module.

● Controllable OMRON Robots

The following OMRON robots that support EtherCAT communications can be controlled.

OMRON robot *1	Product name	Model
Parallel robots	iX3	RX3-□□□□□□□
	iX4	RX4-□□□□□□□
SCARA robot	eCobra *2	RL4-□□□□□□□
	i4H	RS4-□□□□□□□
Articulated robot	Viper *2	RL6-□□□□□□□

*1. Refer to the OMRON robot manuals for information on the specifications of the OMRON robot.
*2. Only the models that support EtherCAT communications are available.

Motion Control Function Module

The MC Function Module interprets commands, performs data processing based on the commands from the sequence control program in the CPU Unit, and performs Servo Drive control.
The NJ Robotics function performs data processing for robot in the MC Function Module and controls Servomotors/Servo Drives with built-in EtherCAT communications to perform robot control.

- When motion control instructions are executed in the sequence control program, the MC Function Module interprets the resulting commands.
- The MC Function Module then performs motion control processing at a fixed period based on the results of the command interpretation. It generates command values to send to the Servo Drive. The following command values are generated: target position, target velocity, and target torque.
- The command values are sent by using PDO communications during each process data communications cycle of EtherCAT communications.

- The Servo Drive performs position loop control, velocity loop control, and torque loop control based on the command values received during each process data communications cycle of EtherCAT communications.
- The encoder's current value and the Servo Drive status are sent to the CPU Unit during each process data communications cycle of EtherCAT communications.

Additional Information

- Motion control processing and process data communications in EtherCAT communications are performed during the same time period.
- The MC Function Module controls the Servo Drive, which contains the position control loop, velocity control loop, and torque control loop.
- Refer to the *NX-series Position Interface Units User’s Manual (Cat. No. W524)* for information on the configuration to use the NX-series Position Interface Units.

● Robot Types Controllable by NJ Robotics Function

The robot types that can be controlled by NJ Robotics function are as follows: Delta3, Delta3R, Delta2, Cartesian 3D, Cartesian 3D Gantry, Cartesian 2D, Cartesian 2D Gantry, H-Bot, T-Bot, Expansion1, SCARA RRP, SCARA RRP+R, SCARA PRR, and SCARA PRR+R

Refer to the *NJ-series NJ Robotics CPU Unit User's Manual (Cat. No. W539)* for details on robot types.

● Motion Control Instructions for NJ Robotics Function

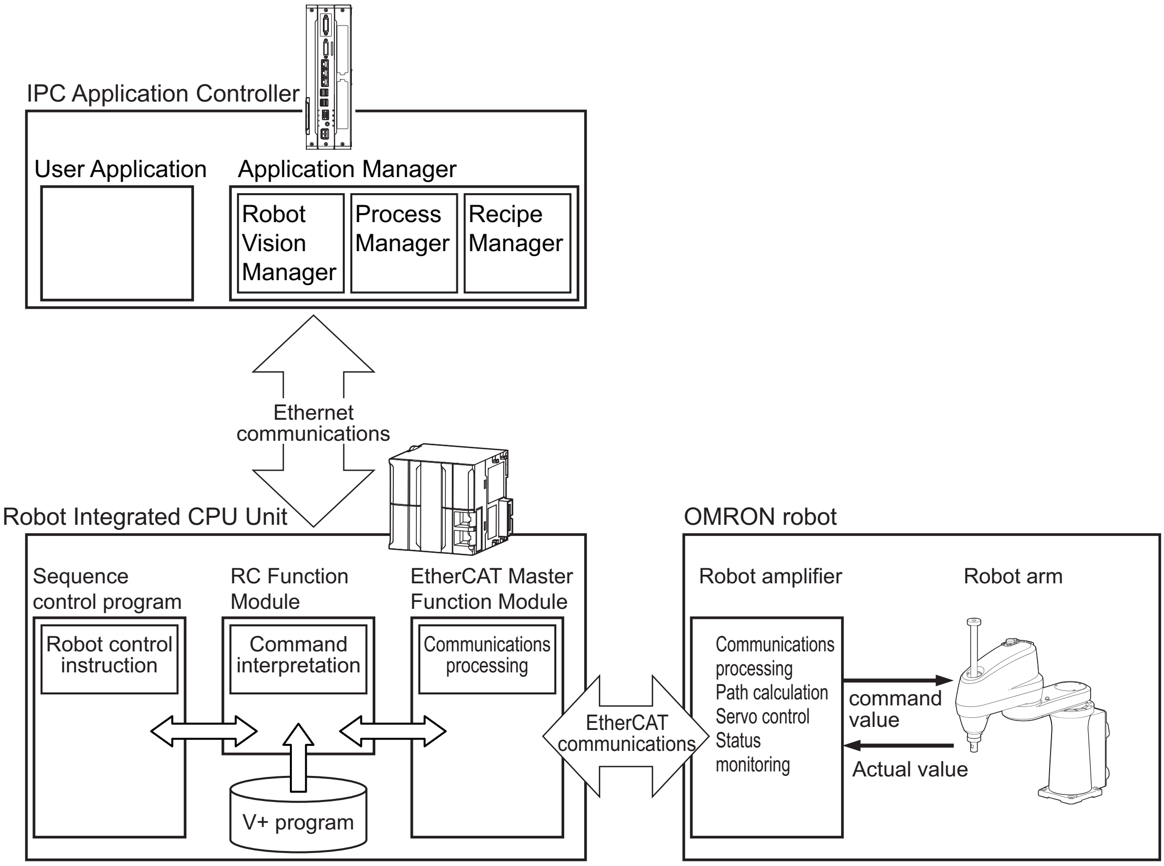
In addition to the motion control instructions that you can use with an NJ501-1□□□ Unit, you can use the following motion control instructions with the NJ Robotics function.
The motion control instructions for the NJ Robotics function are classified as the instructions for axes group commands.

Classification	Motion control instruction	
	Instruction	Name
Instructions for axes group commands	MC_SetKinTransform	Set Kinematics Transformation
	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

Refer to the *NJ-series NJ Robotics CPU Unit User's Manual (Cat. No. W539)* for information on the motion control instructions for NJ Robotics functions.
Refer to the *NJ/NX-series Motion Control Instructions Reference Manual (Cat. No. W508)* for information on the motion control instructions that can be used for the NJ501-1□□□.

2-3 Relationship between Robot Integrated CPU Unit and IPC Application Controller

This section describes the cooperation of the Robot Integrated CPU Unit and a software that is installed in the IPC Application Controller.



Refer to the *Sysmac Studio Robot Integrated System Building Function with IPC Application Controller Operation Manual (Cat. No. W621)* for information on the software that is installed in the IPC Application Controller.

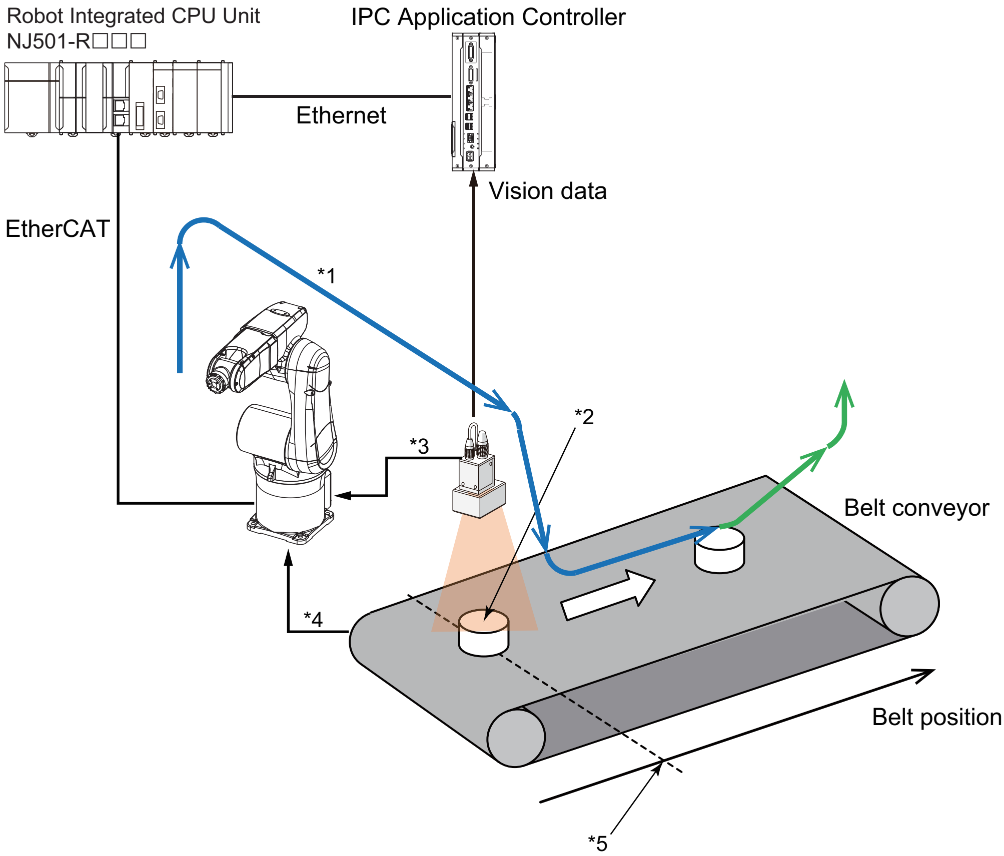
Image Processing with Vision Sensor

You can detect and inspect a workpiece with the vision sensor connected to the IPC Application Controller.
The image processing uses the Robot Vision Manager function in the IPC Application Controller.

Refer to the *Sysmac Studio Robot Integrated System Building Function with IPC Application Controller Operation Manual (Cat. No. W621)* for details.

Conveyor Tracking

You can make the operation of OMRON robot follow the operation of belt conveyor using the encoder that is mounted to the belt conveyor to configure applications for conveyor tracking.



*1. Robot path (conveyor tracking)
*2. Workpiece position detected by the image processing
*3. Input a shutter signal as a latch signal
*4. Pulse of encoder mounted to the belt conveyor
*5. Latched belt position

The conveyor tracking uses the Pack Manager function and Process Manager function in the IPC Application Controller.

Refer to the *Sysmac Studio Robot Integrated System Building Function with IPC Application Controller Operation Manual (Cat. No. W621)* for details.

Request for Recipe Change

You can request a recipe change from the Robot Integrated CPU Unit to the IPC Application Controller.
The recipe change uses the Recipe Manager function and Robot Vision Manager function in the IPC Application Controller.

Refer to *5-6 Changing Recipe* on page 5-14 for information on how to create the user program to request a recipe change.

2-4 System-defined Variables for Robot Control

This section describes the system-defined variables for robot control that belong to the RC Function Module.

In a Robot Integrated CPU Unit, in the same way for an NJ-series CPU Unit, you use variables in the sequence control program to access I/O and memory in the CPU Unit.
You also use the "system-defined variables for robot control" to access the state of OMRON robot and Robot built-in I/O.

2-4-1 Overview of System-defined Variables for Robot Control

System-defined variables are provided in advance in an NJ-series Controller. The variables and all attributes are defined by the system. They have specific functions. You cannot change the variable names or any other attributes.

Of these, system-defined variables that belong to the RC Function Module are called "system-defined variables for robot control".
Of these, system-defined variables that belong to current errors (events) in the RC Function Module are called "error status variables".

Refer to *7-1 System-defined variables for Robot Control* on page 7-2 for details on the list of system-defined variables for robot control.
Refer to *Error Status Variables* on page 11-6 for information on the error status variables.

Level 1	Level 2	Level 3	Description
System-defined variables	System-defined variables for robot control	Robot control common variable	Monitor the common status of the RC Function Module.
		Robot variables	Monitor the status of each OMRON robot.
		Robot I/O variables	Read and write the robot built-in I/O of each OMRON robot.
	Error status variables	Robot Control Error Status	Gives the collective error status of all error status for the RC Function Module.
		Robot Control Common Error Status	Gives the collective error status of all errors that occur for common processing in the RC Function Module.
		Robot Error Status	Gives the collective error status of all error status for each OMRON robot.

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

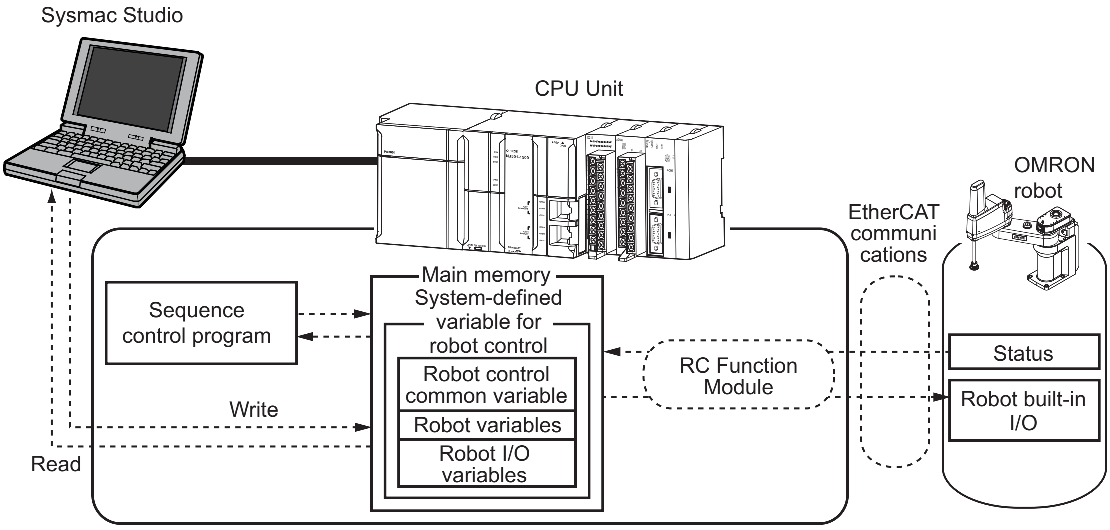
2-4-2 System of System-defined Variables for Robot Control

System-defined variables for robot control consist of information representing the status of the RC Function Module, control command to the OMRON robots connected to EtherCAT communications, status information, and the portion of the robot control parameter settings used to perform robot control.

You can access system-defined variables for robot control from the sequence control program, and read and write them from the Sysmac Studio.
Among the values of system-defined variables for robot control, the *TCPActPos* (Actual Position of TCP for OMRON Robot) and *JointActPos* (Actual Position of Each Joint of OMRON Robot) robot variables retain the previous values even if EtherCAT communications have been disconnected or not established.

Precautions for Correct Use

Access the *_EC_PDSlavTbl* (Process Data Communicating Slave Table) system-defined variable and check that the process data of the slave is enabled (operational) before the robot variables that are not *TCPActPos* (Actual Position of TCP for OMRON Robot) and *JointActPos* (Actual Position of Each Joint of OMRON Robot) and robot I/ I/O variables are used.
Refer to the *NJ/NX-series CPU Unit Built-in EtherCAT Port User's Manual (Cat. No. W505)* for information on the *_EC_PDSlavTbl* (Process Data Communicating Slave Table) system-defined variable.



The system-defined variables for robot control are updated every primary period.
Information from the OMRON robots, which is obtained via EtherCAT communications, is reflected in the system-defined variables for robot control within the control period and can be accessed when the user program is executed.

Robot Control Common Variable

The robot control common variable can monitor the common status of the RC Function Module.
The robot control common variable is a system-defined variable. The variable name _RC_COM is used for the robot control common variable. The data type is _sRC_COMMON_REF, which is a structure.
Refer to *7-1-1 Robot Control Common Variable* on page 7-2 for details on the robot control common variable.

Robot Variables

The robot variables can monitor the current position and state of the OMRON robot.

The robot variable is a system-defined variable. The variable names _RC_RBT[0..7] are used for the robot variables. The data type is _sRC_RBT_REF, which is a structure.
You can use robot variables to access status and position of OMRON robots from the sequence control program.

Each robot variable has two variable names: One is the system-defined variable name and the other is the variable name that is assigned when you add an OMRON robot on the Sysmac Studio.
You can change the variable names that you created on the Sysmac Studio to any variable names for each OMRON robot.

- The system-defined variables have the following variable names.
  _RC_RBT[0] to _RC_RBT[7]
- The default variable names that are assigned when you add an OMRON robot on the Sysmac Studio are shown below.
  You can change these variable names to any variable names for each OMRON robot from the Sysmac Studio.
  RC_Robot001 to RC_Robot008 (default)

You can use either the system-defined variable names or the variable names of the OMRON robots that you added on the Sysmac Studio in the user program.
The relationship between system-defined variable names and robot numbers is shown below.

Robot variable name in the system-defined variables (AT specification in global variable table)	Robot variable name when the robot is added on the Sysmac Studio (Default)	Robot number *1
_RC_RBT[0]	RC_Robot001	1
_RC_RBT[1]	RC_Robot002	2
:	:	:
_RC_RBT[7]	RC_Robot008	8

*1. Refer to *3-3-2 Robot Basic Settings* on page 3-9 for information on the robot number.

Refer to *Operation States of the OMRON Robots* on page 4-22 for information on the definition of each robot status that is stored in robot variables.
Refer to *7-1-2 Robot Variables* on page 7-3 for details on robot variables.

Robot I/O Variables

The robot I/O variables are system-defined variables for handling the robot built-in I/O. The variable names _RC_RBT_IO[0..7] are used for the robot I/O variables. The data type is _sRC_RBT_IO_REF, which is a structure.
You can use robot I/O variables to access the robot built-in I/O of OMRON robots from the sequence control program.

Each robot I/O variable has two variable names: One is the system-defined variable name and the other is the variable name that is assigned when you add a robot on the Sysmac Studio.

- The system-defined variables have the following variable names.
  _RC_RBT_IO[0] to _RC_RBT_IO[7]
- The variable names that are created on the Sysmac Studio are as follows.
  You can change these variable names to any variable names for each OMRON robot from the Sysmac Studio.
  RC_Robot001_IO to RC_Robot008_IO (default)

The relationship between system-defined variable names and robot numbers is shown below.

Robot I/O variable name in the system-defined variables (AT specification in global variable table)	Robot I/O variable name when the robot I/O is added on Sysmac Studio (Default)	Robot number [*1]
_RC_RBT_IO[0]	RC_Robot001_IO	1
_RC_RBT_IO[1]	RC_Robot002_IO	2
:	:	:
_RC_RBT_IO[7]	RC_Robot008_IO	8

*1. Refer to *3-3-2 Robot Basic Settings* on page 3-9 for information on the robot number.

Refer to *7-1-3 Robot I/O Variables* on page 7-8 for details on robot I/O variables.

Precautions for Correct Use

The robot built-in I/O of OMRON robots can be read and written to the same robot built-in I/O device from both the sequence control program and V+ program.
However, write from only one of the programs to the output device of the same robot built-in I/O because exclusive processing is complicated.

2-4-3 Attributes of System-defined Variables for Robot Control

The attributes of system-defined variables for robot control are summarized as follows.

Attribute	System-defined variables for robot control		
	Robot control common variable	Robot variables	Robot I/O variables
Global/local	Global		
R/W access	Read only		Read and write are possible.
Retain	Not retained.		
Network Publish	Published.[*1]		
Name	Fixed.	Different names can be created on the Sysmac Studio.	

*1. Variables are published on the network with the variable names of the system-defined variables. Different variable names that you created on the Sysmac Studio are not published on the network.

2-5 Tasks

This section provides information on tasks for the Robot Integrated CPU Unit and task operation.

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

2-5-1 Tasks and Services for Robot Integrated CPU Unit

Tasks are used to specify an execution condition and execution order to a series of processes, such as I/O refreshing and user program execution.
The NJ-series Robot Integrated CPU Unit support the following tasks.

Type of task	Task name	Program that can be executed
Tasks that execute programs at a fixed period	Primary periodic task	Sequence control program
	Priority-16, 17, or 18 periodic task	
Tasks that execute programs only once when the execution conditions for the tasks are met	Priority-8 or 48 event task	
Tasks that execute V+ programs operated on the system services	V+ task	V+ program

The V+ task can set a maximum of 64 tasks (0 to 63).
Refer to the *eV+3 User's Manual (Cat. No. I651)* for details on the V+ tasks.
Refer to the *NJ/NX-series CPU Unit Software User's Manual (Cat. No. W501)* for information on the tasks that are not V+ tasks.

Precautions for Correct Use

The robot control instructions can be used only for the primary periodic task.
If robot control instructions are used in any other tasks, an error will occur when the user program is built on the Sysmac Studio.
Refer to *6-1 Overview of Robot Control Instructions* on page 6-2 for information on the robot control instructions.

Task Period of Primary Periodic Task

The task period of primary periodic task for the Robot Integrated CPU Unit is given below.

Item	Specification	Initial value
Task period of primary periodic task	1 ms, 2 ms, or 4 ms	2 ms

Precautions for Correct Use

If the OMRON robot is connected, set the EtherCAT communications cycle that is supported by the OMRON robot to the task period of the Robot Integrated CPU Unit.
Refer to the OMRON robot manuals for information on EtherCAT communications specifications of the OMRON robot.

2-5-2 Basic Operation of Tasks

The Robot Integrated CPU Unit cannot execute more than one periodic task and event task at the same time.
The order in which tasks are executed depends on the execution priority that is set for each task.
However, the V+ tasks are operated on the system services and up to 64 tasks are executed in sequence with time slicing according to the priority of the V+ tasks.
To operate a robot with V+ tasks as designed in advance, you must obtain the the system service execution time to design tasks.
Refer to *A-2 Guideline for System Service Execution Time Ratio* on page A-3 for information on the guideline for the system service execution time.

This section explains the relationship between the primary periodic task for the Robot Integrated CPU Unit and the V+ task.
Refer to the *NJ/NX-series CPU Unit Software User's Manual (Cat. No. W501)* and the *eV+3 User's Manual (Cat. No. I651)* for information on the task operation other than above, task execution priority and system services.

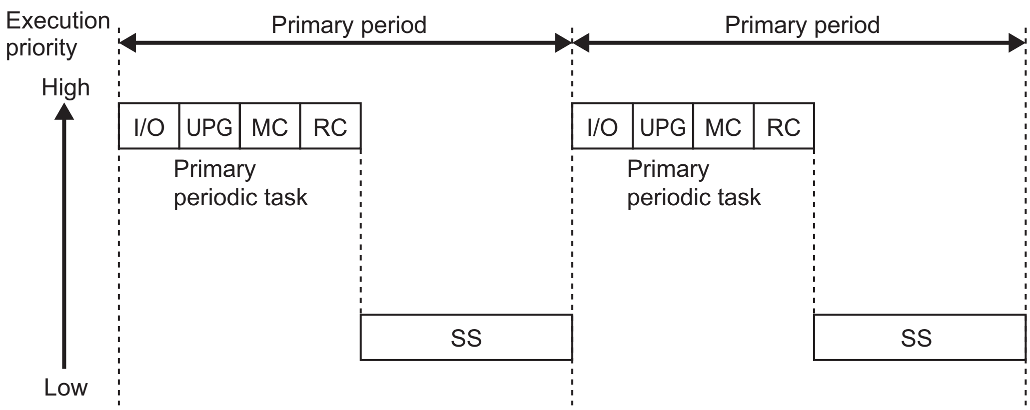
Precautions for Correct Use

If sufficient system service times for execution of V+ tasks cannot be allocated due to the processing time of the user program or the system services other than V+ tasks, the V+ tasks may be executed during more than one period.
Always confirm the operation of V+ tasks in the actual operating conditions.
If the intended operations are not performed, make unused time by extending the primary task period or reviewing the processing.

Operation of Primary Periodic Task

The primary periodic task includes operations such as system common processing, motion control, and robot control in addition to I/O refreshing and user program execution.
After the execution of the primary periodic task, the robot control instruction written in the sequence control program executes in the robot control processing of the next primary periodic task.
The V+ task is executed on the system services.

The operation for the NJ501-R☐☐☐ is given below.
The V+ task, database connection service, and other services are executed on the system services.

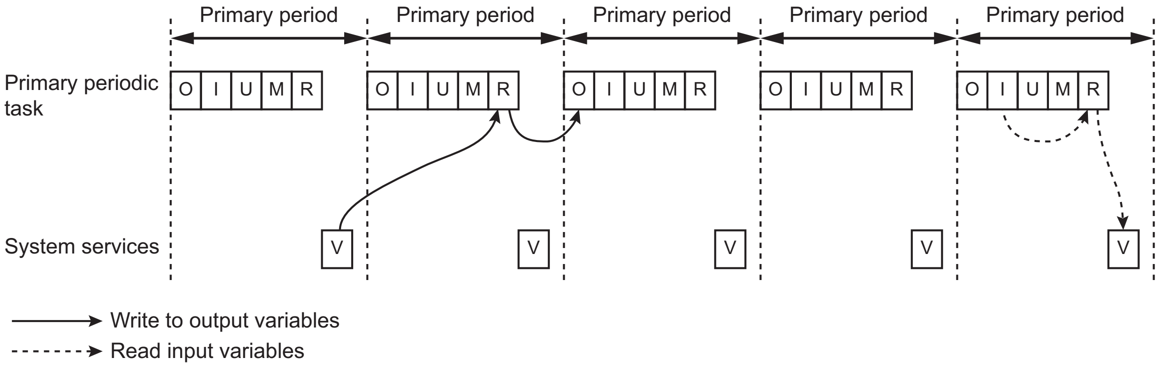


Abbreviation	Description
I/O	I/O refreshing
UPG	User program execution
MC	Motion control
RC	Robot control
SS	System service execution including V+ task and database connection service

2-5-3 Relationship between V+ Task and I/O Refreshing

The digital I/O of the EtherCAT slave devices connected to the Robot Integrated CPU Unit or the NX Units on the EtherCAT Coupler Unit can be read and written from the V+ program assigned to the V+ task.
However, the digital I/O of the EtherCAT slave devices connected to the Robot Integrated CPU Unit or the NX Units on the EtherCAT Coupler Unit cannot be read and written directly from the V+ program, so they are read and written through the RC Function Module.

The V+ program and the timing of I/O refreshing are shown in the following figure.
When output variables are written in the V+ program, data is output at the timing of solid line in the following figure.
When input variables are read in the V+ program, the value input at the timing of dotted line in the following figure is read.



Abbreviation	Description
O	Output refreshing
I	Input refreshing
U	User program execution
M	Motion control
R	Robot control
V	V+ program execution

Precautions for Correct Use

If the digital I/O of the EtherCAT slave devices or the NX Units on the EtherCAT Coupler Unit are read and written in the V+ program, it takes more than one I/O refresh timing and the concurrency of data is not ensured.
If ensuring concurrency is required in the V+ program, read and write I/O data in the sequence control program and use shared variables.

2-6 EtherCAT Communications and Robot Control

The RC Function Module controls OMRON robots through PDO communications of the EtherCAT Master Function Module that is built into the Robot Integrated CPU Unit.

This section describes EtherCAT communications and other items related to the RC Function Module.

EtherCAT Communications Method

Select DC Mode for the EtherCAT communications method between the Robot Integrated CPU Unit and OMRON robots.
If Free-Run Mode is selected for the EtherCAT communications method of the OMRON robot, the OMRON robot cannot be assigned to the RC Function Module.

Precautions for Correct Use

To control the OMRON robot, the process data must be enabled (operational) in the EtherCAT communications.
If a teaching pendant is used, the process data must be enabled (operational) in the EtherCAT communications.
Access the _EC_PDSlavTbl (Process Data Communicating Slave Table) system-defined variable to confirm that the process data is enabled (operational).
Refer to the *NJ/NX-series CPU Unit Built-in EtherCAT Port User's Manual (Cat. No. W505)* for information on the _EC_PDSlavTbl (Process Data Communicating Slave Table) system-defined variable.

Fail-soft Operation

The fail-soft operation can continue or stop communications with EtherCAT slaves that can operate normally when a communications error occurred.
Continuous operation only for the EtherCAT slaves that can operate normally is called fail-soft operation.
The communications between the CPU Unit and EtherCAT slaves can continue until they stop safety by the user program or user operation.

If the fail-soft operation is performed, set **Fail-soft Operation Setting** to **Fail-soft**.
If the **Fail-soft Operation Setting** parameter is set to **Stop**, the Robot Integrated CPU Unit will stop process data communications for all slaves when an EtherCAT communications error is detected in a slave. The OMRON robots also stop.
The operation when the OMRON robot stops depends on the specifications of OMRON robot.

Refer to the *NJ/NX-series CPU Unit Software User's Manual (Cat. No. W501)* and the *NJ/NX-series CPU Unit Built-in EtherCAT Port User's Manual (Cat. No. W505)* for information on the fail-soft operation.
Refer to the OMRON robot manuals for information on the specifications of OMRON robot.

Load Rejection

If the following level Controller error occurred, the Robot Integrated CPU Unit stops output to all of the EtherCAT slaves including the OMRON robots.

- Major fault
- Partial fault for the EtherCAT Function Module
- Partial fault for the RC Function Module

The operation of the OMRON robot and the robot built-in I/O when the output is stopped depends on the specifications of OMRON robot.
Refer to the OMRON robot manuals for information on the specifications of OMRON robot.

Disconnecting and Connecting EtherCAT Slave

The Robot Integrated CPU Unit does not have functionality for the OMRON robot to disconnect or connect EtherCAT slaves and enable or disable the slaves.
Therefore, an EtherCAT Slave Disconnection Error (75020000 hex) occurs when the OMRON robot is disconnected from EtherCAT network or disabled by the following operations.

- If the OMRON robot is disconnected from EtherCAT network with a command from the Sysmac Studio during operation of the Robot Integrated CPU Unit.
- If the OMRON robot is disconnected from EtherCAT network with the EC_DisconnectSlave (Disconnect EtherCAT Slave) instruction.
- If the OMRON robot is disabled with the EC_ChangeEnableSetting (Enable/Disable EtherCAT Slave) instruction.

The operation when the OMRON robot stops depends on the specifications of OMRON robot.

Refer to the *NJ/NX-series Instructions Reference Manual (Cat. No. W502)* for information on the EC_DisconnectSlave (Disconnect EtherCAT Slave) instruction and the EC_ChangeEnableSetting (Enable/Disable EtherCAT Slave) instruction.
Refer to the OMRON robot manuals for information on the specifications of OMRON robot.

2-7 SD Memory Card Operations

This section describes the functions to access to an SD Memory Card.
The Robot Integrated CPU Unit provides functionality to access from a V+ program along with the SD Memory Card function supported by the NJ-series CPU Units.
The following table shows the functions to access to the SD Memory Card and the range that can be accessed.

Function	Description
Downloading and uploading from/to the Sysmac Studio	The Sysmac Studio can download and upload the robot control parameters, robot setting files, and V+ programs to the Robot Integrated CPU Unit.
V+ File Browser in the Sysmac Studio*1	Use the V+ File Browser in the Sysmac Studio to access V+ programs and files created in the V+ programs in the Robot Integrated CPU Unit.
SD Memory Card function in the Sysmac Studio	Use the SD Memory Card function in the Sysmac Studio to access files in the SD Memory Card.
SD Memory Card access from the FTP client	Enable the FTP server for the NJ-series CPU Units to access files in the SD Memory Card from the FTP client.
File access from the V+ program*1	The V+ programs can use V+ keywords such as FCOPY to create, edit, and delete files in the SD Memory Card.
File access from the sequence control program	The sequence control program can use SD Memory Card instructions or FTP client instructions to create, edit, and delete files in the SD Memory Card.
SD Memory Card backups	Back up, restore, or verify data between the files under the D folder and backup files in the same SD Memory Card.
Sysmac Studio Controller backups	Back up, restore, or verify data between the files under the D folder in the SD Memory Card and backup files in the computer that the Sysmac Studio is installed.
Save configuration setting and auto startup setting on the Sysmac Studio	When the save configuration setting is enabled, write the data in V+ memory into the SD Memory Card. When the auto startup is enabled, read the data in the SD Memory Card to the V+ memory.
Clear All Memory in the Sysmac Studio	Initialize the files under the D folder in the SD Memory Card.
Monitor Window on the Sysmac Studio	Use the Monitor Window on the Sysmac Studio to access files in the SD Memory Card.

*1. Only the D folder under the root directory can be accessed.

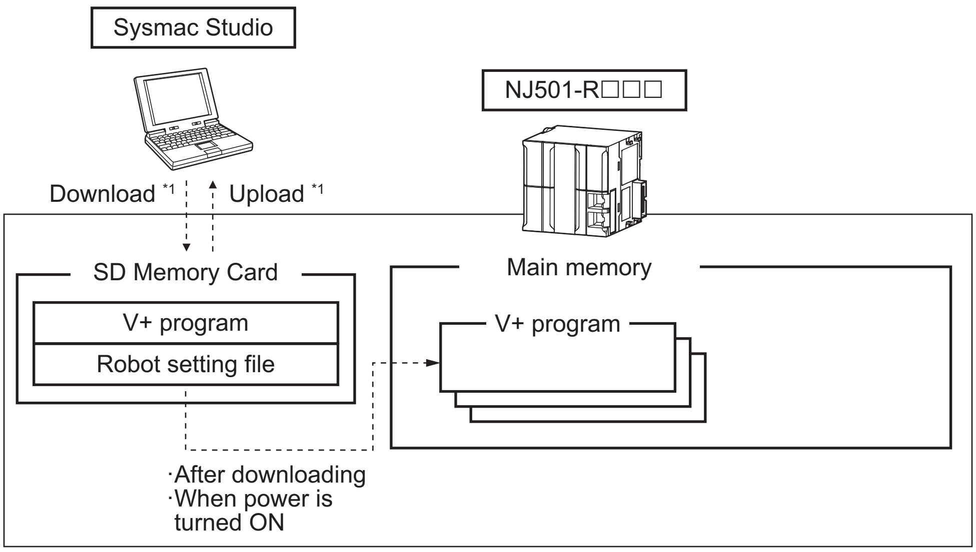
Refer to the *NJ/NX-series CPU Unit Software User’s Manual (Cat. No. W501)* for information on the SD Memory Card function of the NJ-series CPU Unit.

Precautions for Correct Use

- Always insert an SD Memory Card when you use the robot control function with the Robot Integrated CPU Unit.
- Do not remove the SD Memory Card while power is supplied when you use the robot control function with the Robot Integrated CPU Unit.
  Doing so causes the robot control function to stop due to an error.
- The robot setting files and V+ program files in the SD Memory Card are required for the operation of the RC Function Module.
  Be careful not to overwrite or delite the robot setting files and V+ program files.
- Do not set write protection for the SD Memory Card.
  If the SD Memory Card is set to write protection, you cannot connect online.
- When you use downloading and uploading operations from the Sysmac Studio, an error occurs if data consistency is not ensured. This will occur because the SD Memory Card is faulty or writing data failed due to insufficient space available on the SD Memory Card.
  In the same way, if the power supply to the CPU Unit is turned OFF during transfer to the CPU Unit, when data consistency is not ensured, an error is detected.

2-7-1 Included SD Memory Card Functions

The V+ program using the Robot Integrated CPU Unit and robot setting files are saved in an SD Memory Card.
Therefore, if you use the robot control function, the included SD Memory Card is required to insert to the Controller.



*1. Use the Synchronize Menu of the Sysmac Studio to "upload" and "download" the data.

Precautions for Correct Use

- If you insert an SD Memory Card, always turn OFF the power supply to the Robot Integrated CPU Unit.
- If you press the power supply switch, the PWR indicator is not turned off and the power is not stopped.
- The V+ program using the RC Function Module and robot setting files are saved in an SD Memory Card.
  Do not insert and remove an SD Memory Card while power is supplied to the Robot Integrated CPU Unit.
  When the SD Memory Card is removed, a Remove SD Memory Card with Robot Control Function Enabled error (17C10000 hex) occurs.
- Do not format the SD Memory Card, edit or delete data related to the robot control function such as the robot setting files and V+ programs while the Robot Integrated CPU Unit is operating.
  When these operations are performed, a Remove SD Memory Card with Robot Control Function Enabled error (17C10000 hex) occurs.
- Of the online operations in the Sysmac Studio, the following operations overwrite data in the SD Memory Card.
  a) Downloading from the Sysmac Studio
  b) Restoring in the Sysmac Studio
  c) Clear All Memory in the Sysmac Studio
  d) Save configuration setting on the Sysmac Studio
  e) Monitor Window on the Sysmac Studio
  Because a Robot Integrated CPU Unit can connect online with more than one Sysmac Studio, if the above operations are performed at the same time, these operations are not ensured.
  If a Robot Integrated CPU Unit connects online with more than one Sysmac Studio, pay careful attention during operation.

The differences of operation whether the SD Memory Card is inserted or not at startup are described below.

	SD Memory Card at startup	
	Inserted	Not inserted
Connection with the Sysmac Studio	Possible.	Possible.
Troubleshooting function of the Sysmac Studio	Possible.	Possible.
Robot integrated system control function of the Sysmac Studio	Possible.	Not possible.
Downloading and uploading from/to the Sysmac Studio	Possible.	Not possible.
Backing up and restoring data in the Sysmac Studio	Possible.	Not possible.
Clear All Memory in the Sysmac Studio	Possible.	Not possible.
Using the RC Function Module	Possible.	Not possible.
Using functions other than the RC Function Module	Possible.	Not possible.*1

*1. If you remove the SD Memory Card after you download data from the Sysmac Studio or restore data from the SD Memory Card, the function can be operated.

Refer to *A-1 Differences in Functions between Robot Integrated CPU Unit and NJ-series CPU Unit* on page A-2 for information on differences in functions for the Robot Integrated CPU Unit and the NJ-series CPU Unit.

SD Memory Card Specifications

The SD Memory Card is inserted into the Robot Integrated CPU Unit when it is shipped.

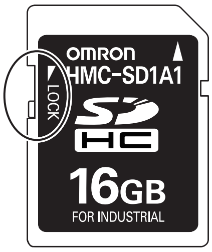
SDHC cards are supported, but use one of the following OMRON Cards.
OMRON is not responsible for the operation, performance, or write life of any other SDHC card.

Model	Card type	Capacity [GB]	Formatting	Number of overwrites	Weight
HMC-SD491[*1]	SDHC card	4	FAT32	100,000 overwrites	2 g max.
HMC-SD1A1		16			

*1. This is the SD Memory Card inserted when the CPU Unit is shipped.

Additional Information

Write Protection Key
You will not be able to write to the SD Memory Card if the key is set to the LOCK position. Use this setting to prevent overwriting.



State of Included SD Memory Card at Factory Setting

Nothing is recorded in the included SD Memory Card at the factory setting.
The Robot Integrated CPU Unit checks whether the D folder exists in the SD Memory Card when the power is turned ON.
If the folder does not exist, the Robot Integrated CPU Unit creates a new folder in the SD Memory Card when the power is turned ON and copies the robot setting files from the built-in non-volatile memory in the CPU Unit to the SD Memory Card.

Precautions for Correct Use

If the copy to the SD Memory Card fails, a Robot Control Function Enabled without SD Memory Card error (17C20000 hex) occurs.

2-7-2 Exclusive Control of File Access in the SD Memory Card

If the same file on the SD Memory Card is accessed from different sources, unintended operations such as reading a file while it is being written or writing a file while it is being read may occur.
Therefore, it is necessary to perform exclusive controls in order to prevent multiple accesses ("reading and writing data" or "writing and writing data") to the same file simultaneously. It is not necessary to perform exclusive controls for "reading and reading data".
When you use a combination of operations that requires exclusive controls, always execute the later processing only after checking that the first processing is finished.
Note that the exclusive controls are performed automatically for the file accesses with more than one instruction in the sequence control program.

When the following functions are used, an access to files on the SD Memory Card will occur.

- Some V+ keywords to access the SD Memory Card
- SD Memory Card operation instructions and FTP client instructions in the sequence control program
- Backup, restore and verification operations with the SD Memory Card
- File operations in the SD Memory Card from the Sysmac Studio
- Downloading, uploading, and verification from the Sysmac Studio
- V+ Edit from the Sysmac Studio
- Backup, restore and verification operations from the Sysmac Studio
- FTP server

Refer to the *eV+3 Keyword Reference Manual (Cat. No. I652)* for information on some V+ keywords to access the SD Memory Card.

Precautions for Correct Use

- Do not newly add, edit, or delete a folder or file under the D folder in the SD Memory Card during the backup or verification operation.
  If a folder or file is newly added, edited, or deleted under the D folder during the backup or verification operation, the uncompleted folder or file may be backed up or verified, so the operation is not ensured.
- If the restore, automatic transfer, program transfer, or verification operation is executed, the data under the D folder from a backup file is temporarily expanded under the ~~D folder in the SD Memory Card.
  Do not use the same folder name as a temporary folder name because the temporary folder that is existed in advance is cleared.

2-8 Memory Management

This section describes the memory management for the Robot Integrated CPU Unit.

2-8-1 Data and File Locations

The data and files related to the robot control function are located in the built-in non-volatile memory in the CPU Unit or SD Memory Card as described below.
The data placed in the SD Memory Card is located in the D folder under the root directory.

Data or file	Description	Location
V+ program	A program file that includes the V+ programs	SD Memory Card
Global variable data of the V+ program	Saved data for global variable values of the V+ program This data includes the teaching location data.	SD Memory Card
Robot setting files	Setting files that include the OMRON robot settings	SD Memory Card
Robot control parameters	Parameters that are used by the RC Function Module	Built-in non-volatile memory
V+ program creation file	Files created with a file creation instruction or copy instruction during V+ program.	SD Memory Card

Refer to *Specifications of Supported SD Memory Cards, Folders, and Files* in the *NJ/NX-series CPU Unit Software User's Manual (Cat. No. W501)* for information on the folders in the SD Memory Card and file specifications.

2-8-2 Clear All Memory

You can perform the Clear All Memory operation from the Sysmac Studio to return to the factory settings for the user program, configuration, settings, and variables in the CPU Unit including the SD Memory Card.
The data for the Clear All Memory operation is given below.

Type of data		Description
User programs	Sequence control program	Cleared.
	V+ programs	Cleared.
CPU Unit configuration and settings		Cleared. *1
Variables	System-defined variables	Cleared.
	Device variables	
	User-defined variables	
	Variables for V+ programs	
Event logs		Cleared if the user selected.
Files under the D folder in the SD Memory Card *2		Cleared.

*1. The robot control parameters, system setting files and robot setting files are also cleared.
*2. The files include the V+ program creation file.

Refer to the *Sysmac Studio Version 1 Operation Manual (Cat. No. W504)* for information on the Clear All Memory operation.
Refer to the *NJ/NX-series Database Connection CPU Units User's Manual (Cat. No. W527)* for information on the Clear All Memory operation for the database connection service.

Precautions for Correct Use

- The Clear All Memory operation can be performed only in PROGRAM mode.
- When the operating mode is changed to the PROGRAM mode, the OMRON robot that is controlled with the sequence control program stops.
- If the Clear All Memory operation is performed, when V+ programs are running or OMRON robots are moving, the Sysmac Studio displays a confirmation dialog box, stops the V+ programs or OMRON robots, and clears the memory.

2-9 Backup and Restore Operations

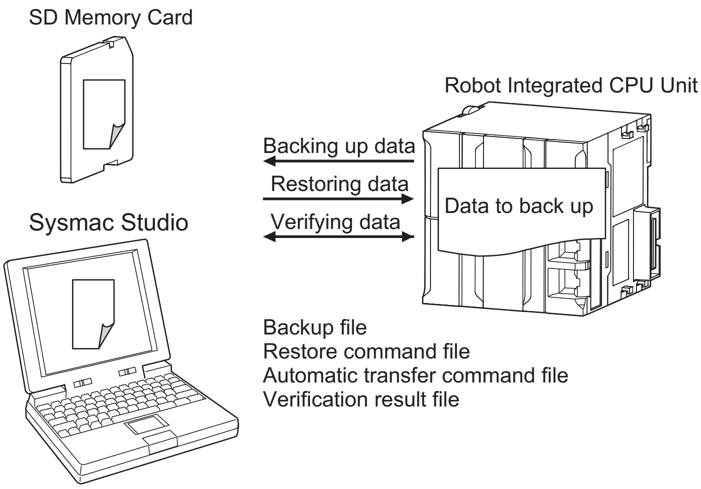
This section provides information on backup and restore operations for the Robot Integrated CPU Unit and the OMRON robot.

Additional Information

There are restrictions on the combination of the unit version of the Robot Integrated CPU Unit and the version of the OMRON robot.
If the version of the device to create a backup data and the version of the device to restore do not meet these restrictions, the backup and restore operations cannot be used.
Refer to *A-3-1 Correspondence between Unit Versions of CPU Units and Versions of Sysmac Studio and OMRON Robots* on page A-4 for details.

2-9-1 Backup and Restore Operations for Robot Integrated CPU Unit

Use the SD Memory Card or the Sysmac Studio to perform backup and restore operations for the Robot Integrated CPU Unit.



The following data is included for the buckup operation, in addition to the backup operation for the NJ-series CPU Units.

- Robot control parameters
- System setting files
- V+ Programs
- Global variable data of the V+ Program
- Robot setting files
- V+ Program creation file

The data that is saved in the built-in non-volatile memory in the CPU Unit and all folders and files under the D folder that is saved in the SD Memory Card are backed up.

Refer to the *NJ/NX-series CPU Unit Software User's Manual (Cat. No. W501)* for information on the NJ-series common specifications related to the backup and restore operations.

Operating Procedure When an SD Memory Card is Used

This section describes the procedure for backup, restore, and verification operations to an SD Memory Card.

● Procedure for Backup Operations to the SD Memory Card

The Robot Integrated CPU Unit saves V+ Programs and robot setting files in the SD Memory Card. If you buck up data, the V+ Programs and robot setting files in the SD Memory Card are copied to the backup file. Therefore, perform the backup operation while the SD Memory Card used during operation is inserted into the Robot Integrated CPU Unit.

Precautions for Correct Use

The SD Memory Card used for backup is required to insert to the Robot Integrated CPU Unit in order to restart the Controller.
Copy the backup-related files from the SD Memory Card that is used for backup, and then prepare a new SD Memory Card for restore operation.
Refer to the *NJ/NX-series CPU Unit Software User's Manual (Cat. No. W501)* for information on the backup-related files.

1. Set pins 1 to 4 on the DIP switch on the CPU Unit as follows: 1: OFF, 2: OFF, 3: ON, and 4: OFF.
2. Press the SD Memory Card power supply switch for 3 seconds.
   The backup is started. The SD PWR indicator will flash, lighting for 3 seconds and going out for 0.5 seconds.
   When the backup operation is completed, the SD PWR indicator will stop flashing and remain lit.
3. Set all of pins 1 to 4 on the DIP switch on the CPU Unit to OFF.
4. Turn OFF the power supply to the CPU Unit and to the EtherCAT slaves.
5. Remove the SD Memory Card from the CPU Unit.
6. Copy the backup-related files from the SD Memory Card that is used for backup, and then prepare a new SD Memory Card for restore operation.
7. Insert the SD Memory Card used for backup into the CPU Unit.
8. Turn ON the power supply to the CPU Unit and to the EtherCAT slaves.

Additional Information

If you copy the backup-related files without turning OFF the power supply, use the Sysmac Studio or FTP client.

● Procedure for Restore Operations from the SD Memory Card

The Robot Integrated CPU Unit must operate while the SD Memory Card that is used for restore operation is inserted.
Therefore, we recommend that the SD Memory Card for restore operation is copied before you perform the operation.

1. Turn OFF the power supply to the CPU Unit and to the EtherCAT slaves.
2. Insert the SD Memory Card for restore operation that stores the backup file into the CPU Unit.
3. Set pins 1 to 4 on the DIP switch on the CPU Unit as follows: 1: OFF, 2: OFF, 3: ON, and 4: ON.
4. Turn ON the power supply to the CPU Unit and to the EtherCAT slaves.
   The restore operation is started. The SD PWR indicator will flash, lighting for 3 seconds and going out for 0.5 seconds.
   When the restore operation is completed, the SD PWR indicator will stop flashing and remain lit.
5. Turn OFF the power supply to the CPU Unit and to the EtherCAT slaves.
6. Set all of pins 1 to 4 on the DIP switch on the CPU Unit to OFF.
7. Turn ON the power supply to the CPU Unit and to the EtherCAT slaves.

Precautions for Correct Use

- To restore data with EtherCAT slaves connected, always cycle the power supply to the CPU Unit and the EtherCAT slaves after completion of the restore operation.
  If you start operation without cycling the power supply, the Controller may perform unintended operation.
- To verify the data after you restore data with EtherCAT slaves connected, first turn OFF the power supply to the CPU Unit and EtherCAT slaves, and then start the Robot Integrated CPU Unit in Safe Mode before you perform the verification procedure.
  If you cycle the power supply normally, the Controller will start operation before you perform the verification procedure. That means that operation could be started with data that is not correct.
  For information on Safe Mode, refer to the *NJ/NX-series Troubleshooting Manual (Cat. No. W503)*.

● Procedure for Verification Operations of the SD Memory Card

The Robot Integrated CPU Unit saves V+ Programs and robot setting files in the SD Memory Card.
When you verify data, the programs and settings in the V+ Programs and robot setting files are compared with the backup file.
Therefore, perform the verification operation using the SD Memory Card for restore operation.

1. Set all of pins 1 to 4 on the DIP switch on the CPU Unit to OFF.
2. Press the SD Memory Card power supply switch for 3 seconds.

Data comparison is started. The SD PWR indicator will flash, lighting for 3 seconds and going out for 0.5 seconds.

If the verification operation is completed and the data is the same, the SD PWR indicator will stop flashing and remain lit.
If the verification operation is completed and differences were found in the data, the SD PWR indicator will flash, lighting for 0.5 seconds and going out for 0.5 seconds.

Operating Procedure When the Sysmac Studio is Used

Refer to the *NJ/NX-series CPU Unit Software User's Manual (Cat. No. W501)* and the *Sysmac Studio Version 1 Operation Manual (Cat. No. W504)* for information on the procedures for backup, restore, and verification operations to the Sysmac Studio.

The Robot Integrated CPU Unit saves V+ programs and robot setting files in the SD Memory Card. When you back up, restore, or verify data, write or access V+ programs and robot setting files. Therefore, perform the backup, restore, and verification operations while the SD Memory Card used during operation is inserted.

Required Available Space of SD Memory Card

If you use the SD Memory Card for the backup and restore operations of the Robot Integrated CPU Unit, the backup file is saved in the SD Memory Card.
If the backup is executed using an SD Memory Card without sufficient available capacity, a Backup Failed (102A0000 hex) occurs.

The capacity of data to back up vary depending on the operating conditions of the Robot Integrated CPU Unit. Therefore, confirm that the backup operation is normally ended before the Robot Integrated CPU Unit is started.

Folders in SD Memory Card

If the D folder does not exist in the SD Memory Card when data is backed up, a Backup Failed (102A0000 hex) will occur.
The data placed in the D folder is included in the *User program and settings* under Data group. Therefore, be sure to back up the data group.
Refer to the *NJ/NX-series CPU Unit Software User's Manual (Cat. No. W501)* for information on the data group.

Precautions for Correct Use

- Do not newly add, edit, or delete a folder or file under the D folder in the SD Memory Card during the backup or verification operation.
  If a folder or file is newly added, edited, or deleted under the D folder during the backup or verification operation, the uncompleted folder or file may be backed up or verified, so the operation is not ensured.
- If the restore, automatic transfer, program transfer, or verification operation is executed, the data under the D folder from a backup file is temporarily expanded under the ~~D folder in the SD Memory Card.
  Do not use the same folder name as a temporary folder name because the temporary folder that is existed in advance is cleared.

2-9-2 Backup and Restore Operations for OMRON Robot

The OMRON robot settings are stored in the SD Memory Card in the Robot Integrated CPU Unit. The settings are automatically transferred from the Robot Integrated CPU Unit to the OMRON robot. Therefore, use the SD Memory Card or Sysmac Studio to perform backup and restore operations for the Robot Integrated CPU Unit. This can also back up and restore the OMRON robot settings.

If you need to replace the OMRON robot, the OMRON robot settings are automatically transferred, so you can quickly replace the OMRON robot.

Replacement Procedure for OMRON Robot

This section describes the procedure to replace an OMRON robot.

1. Turn OFF the power supply to the CPU Unit, OMRON robot, and other EtherCAT slaves.
2. Replace the OMRON robot.
3. Turn ON the power supply to the CPU Unit, OMRON robot, and other EtherCAT slaves.

Refer to the OMRON robot manuals for details on the replacement procedure for OMRON robot.

2-10 Security

To protect your assets, you can use security functions to protect the user program and various data in the Controller. To prevent incorrect operation, you can also use security functions to restrict operations on the Sysmac Studio.
The Robot Integrated CPU Unit adds V+ programs and robot control parameters along with the security function supported by the NJ-series CPU Units.
The operation authority verification is the same function supported by the NJ-series CPU Units.
Refer to the *NJ/NX-series CPU Unit Software User's Manual (Cat. No. W501)* for details on the operation authority verification.
Refer to the *Sysmac Studio Version 1 Operation Manual (Cat. No. W504)* for specific operating procedures for operation authorities.

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

2-10-1 Robot System Operation Authority

Online operations are restricted by operation rights to prevent damage to equipment or injuries that may be caused by operating mistakes.
The online operations that are allowed for each operation authority in the Robot Integrated CPU Unit are given below.
You can use online operations with the Robot System Operation Authority to set the operation authorities.

The access level in default setting is the design engineer access level.

Refer to the *Sysmac Studio Robot Integrated System Building Function with Robot Integrated CPU Unit Operation Manual (Cat. No. W595)* for information on the Robot System Operation Authority.

(OK: Operation possible, NP: Operation not possible)

Function	Operator	Teaching engineer	Design engineer
Robot built-in I/O	NP	OK	OK
Variables for V+ programs	NP	OK	OK
Online editing of V+ programs	NP	NP	OK
Monitoring window of V+ programs	NP	NP	OK
Restarting V+OS	NP	NP	OK
File explorer related to V+	OK	OK	OK
Robot test run	NP	OK	OK
Virtual front panel	NP	NP	OK
V+ task status control	NP	NP	OK
Vision sensor window	NP	OK	OK

2-10-2 CPU Unit Write Protection

This function disables the ability to write data to the CPU Unit to protect user program assets and prevent misuse.
The following protect data added for the Robot Integrated CPU Unit is given below.

CPU Unit data	Writing data to the CPU Unit during write protection	Data location
Robot control parameters	Not possible.	Built-in non-volatile memory

Refer to the *NJ/NX-series CPU Unit Software User's Manual (Cat. No. W501)* for information on the CPU Unit write protection.

3

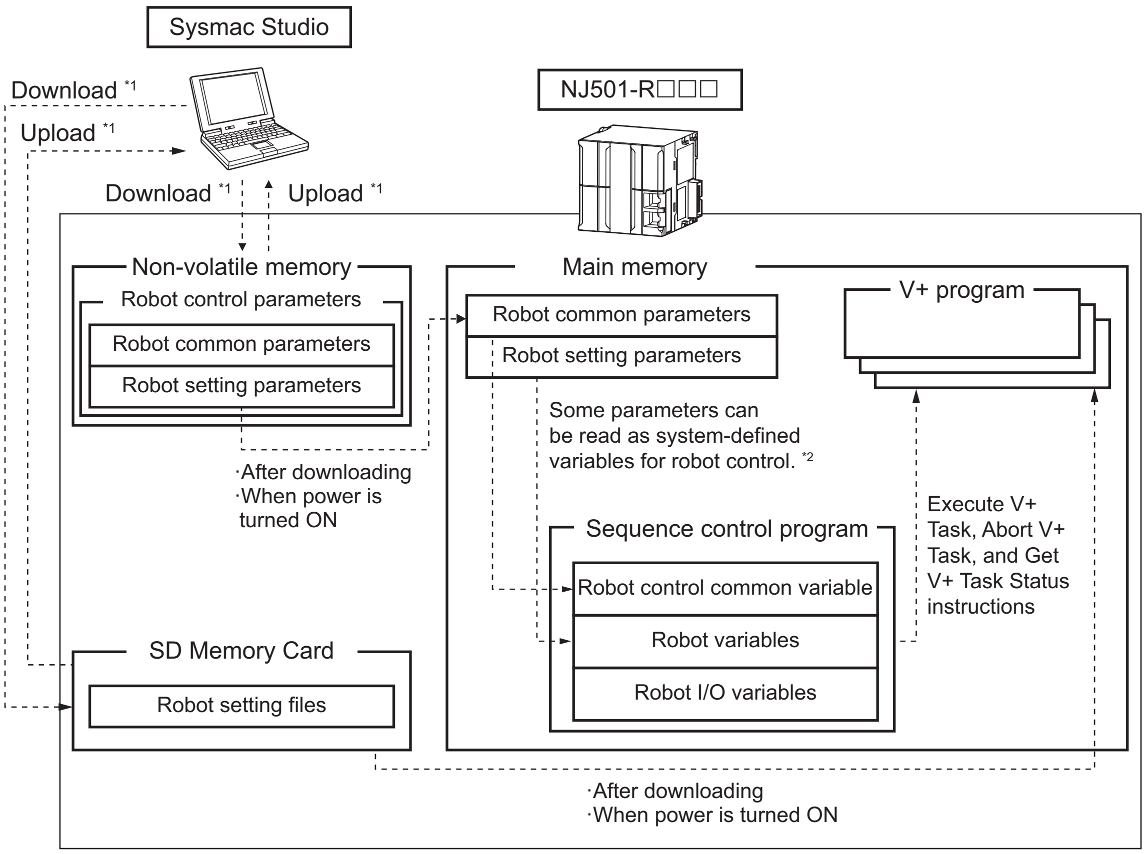
Robot Control Parameters

This section describes the parameters that are set in the Robot Control Function Module.

3-1 Introduction to Robot Control Parameters

You can use the sequence control program or V+ program to control OMRON robots with the Robot Control Function Module of the Robot Integrated CPU Unit.
To set the motion of each OMRON robot, robot control parameters and robot setting files are used.
Robot control parameters are stored in the non-volatile memory, whereas robot setting files are stored in an SD Memory Card.

Robot control parameters are a generic term for parameters including following:
- Robot common parameters
- Robot setting parameters



*1. Use the Synchronize Menu of the Sysmac Studio to "upload" and "download" the project.
*2. System-defined variables for robot control corresponding to robot control parameters can be read only. They cannot be written.

3-1-1 Data Flow for Robot Control Parameters

- Download the robot control parameters that you set using the Sysmac Studio to the Robot Integrated CPU Unit to save the parameter settings in the built-in non-volatile memory in the Robot Integrated CPU Unit. When you upload the robot control parameters to the Sysmac Studio, the robot control parameters that were saved in the non-volatile memory are uploaded.

- The settings that were saved in the non-volatile memory are applied to the main memory after you download them or when the power is turned ON.
- If the settings are correct, the RC Function Module executes control based on the settings in the main memory.
  If there is a mistake or error in the settings, the RC Function Module causes a partial fault.
- The settings of some of the parameters can be accessed as system-defined variables for robot control with the user program.
- You can upload and download robot control parameters regardless of the Robot Integrated CPU Unit's mode or the status of the RC Function Module.

3-1-2 Relationship between V+ Program and Robot Control Parameters

- You cannot read or change robot control parameters directly from the V+ program.
- To read robot control parameters, you need to share variables in the V+ program and the sequence control program to read them indirectly. However, some parameters can be read.
  Refer to *4-2-5 Shared Variables with V+ Program* on page 4-7 and *4-2-6 Using Shared Variables with V+ Programs* on page 4-12 for details on the shared variables.

3-2 Robot Common Parameters

The robot common parameters provide settings that are required for OMRON robots to the RC Function Module.
Even if more than one OMRON robot is connected to the Robot Integrated CPU Unit, there is only one set of robot common parameters per Robot Integrated CPU Unit.

3-2-1 Robot Common Parameters

Use the Sysmac Studio to set the robot common parameters for each Robot Integrated CPU Unit.
Refer to the *Sysmac Studio Robot Integrated System Building Function with Robot Integrated CPU Unit Operation Manual (Cat. No. W595)* for details on how to set the robot common parameters.

Classification	Parameter name	Reference
V+Digital I/O Setting 1	Device	*3-2-2 V+Digital I/O Settings* on page 3-4
	Port	
	V+Digital I/O	
V+Digital I/O Setting 2	Device	
	Port	
	V+Digital I/O	
⁝ (Each element above can be repeated up to 999 times)		
V+Digital I/O Setting 999	Device	
	Port	
	V+Digital I/O	
Remote Encoder Latch Setting 1	Used Encoder ID	*3-2-3 Remote Encoder Latch Settings* on page 3-6
	Motion Control Axis	
	Position Type Selection	
	Latch Signal Number	
⁝ (Each element above can be repeated up to 16 times)		
Remote Encoder Latch Setting 16	Used Encoder ID	
	Motion Control Axis	
	Position Type Selection	
	Latch Signal Number	

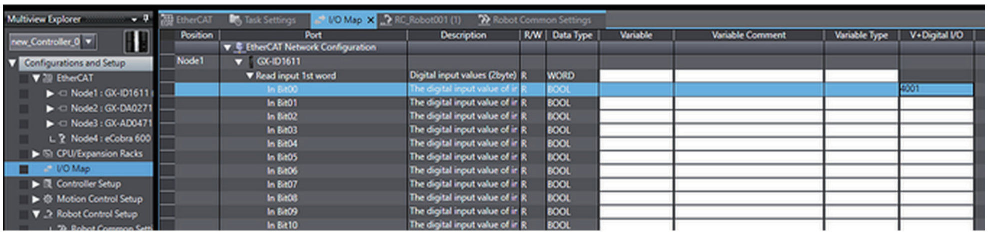
3-2-2 V+Digital I/O Settings

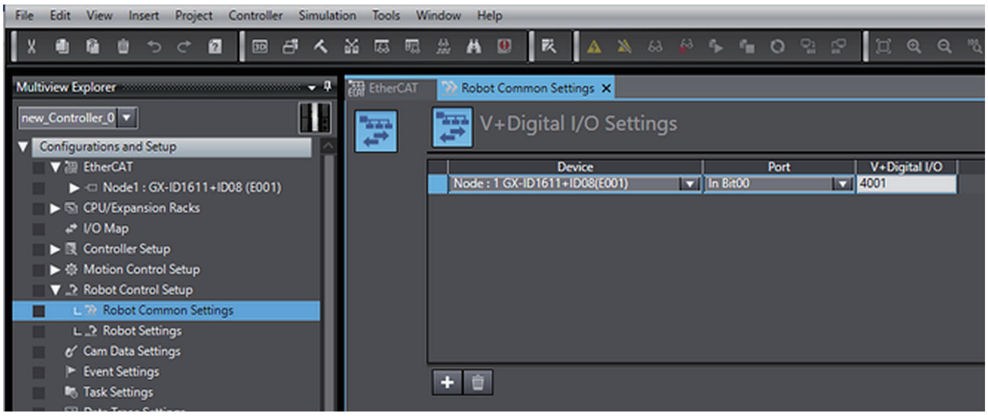
Make the **V+Digital I/O Settings** to use I/O ports of the Robot Integrated CPU Unit as I/O signals in the V+ program.

Refer to the *Sysmac Studio Robot Integrated System Building Function with Robot Integrated CPU Unit Operation Manual (Cat. No. W595)* for how to set the **V+Digital I/O Settings**.
Refer to the *NJ/NX-series CPU Unit Software User's Manual (Cat. No. W501)* for information on the I/O ports.

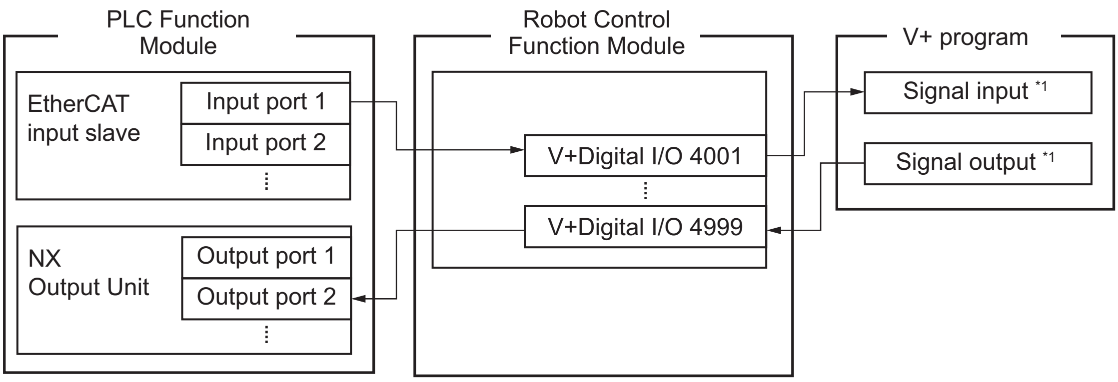
Parameter name	Function	Setting range
Device	Select an EtherCAT slave device or an NX Unit on the EtherCAT Coupler Unit that is connected to the Robot Integrated CPU Unit for which the V +Digital I/O are assigned.	---
Port	Select the I/O ports of the above device to assign to the V+Digital I/O.	
V+Digital I/O	Set the V+Digital I/O that is used when a device is accessed from the V+ program.	4001 to 4999

The images of the setting tab pages are shown below.
The following figures are the setting tab pages for I/O Map (upper figure) and Robot Common Settings (lower figure).





The following figure shows an overview of V+Digital I/O settings.
In an example below, the input port 1 of input slave in the EtherCAT slave device is assigned to the V +Digital I/O 4001. The output port 2 of the Digital Output Unit in NX Units on the EtherCAT Coupler Unit is assigned to the V+Digital I/O 4999.



*1. Can be used from the V+ program in the same way as the robot built-in I/O.

Precautions for Correct Use

- The assignment of the V+Digital I/O and device variables to the same output ports is a exclusive relationship. Therefore, the output ports to which the V+Digital I/O are assigned cannot output signals from the sequence control program.
  In the same way, the output ports that are assigned to the device variables cannot output signals from the V+ program.
- The V+Digital I/O and device variables can be assigned to the same input ports. Therefore, the same input ports can be accessed in either the V+ program or the sequence control program.

Device

Select the device to assign to the V+Digital I/O.

The following table shows the EtherCAT slave device and NX Units that can be selected.

Device	Type
EtherCAT slave	Digital I/O
NX Units	NX Unit with input refreshing with input changed times Digital Input Unit NX Unit with output refreshing with specified time stamps Digital Output Unit Digital Mixed I/O Unit

Precautions for Correct Use

You cannot specify CJ-series Units that are connected to the Robot Integrated CPU Unit.

Port

Select the I/O ports of the device to assign to the V+Digital I/O.

Select the I/O ports of the device that you selected in *Device* on page 3-6.

V+Digital I/O

Set the number in order to access to the I/O ports for digital I/O of the EtherCAT slave devices or the NX Units on the EtherCAT Coupler Unit from the V+ program.

In the V+ program, any signal is input or output by specifying V+Digital I/O.

For the input ports or output ports of a device connected to the Robot Integrated CPU Unit, you can assign V+Digital I/O between 4001 and 4999.

3-2-3 Remote Encoder Latch Settings

Set the latching of the motion control axis used from the V+ program.

For each prepared motion control axis, set **Motion Control Axis** and other parameters for a used encoder ID so that it can be accessed from the V+ program as a remote encoder.

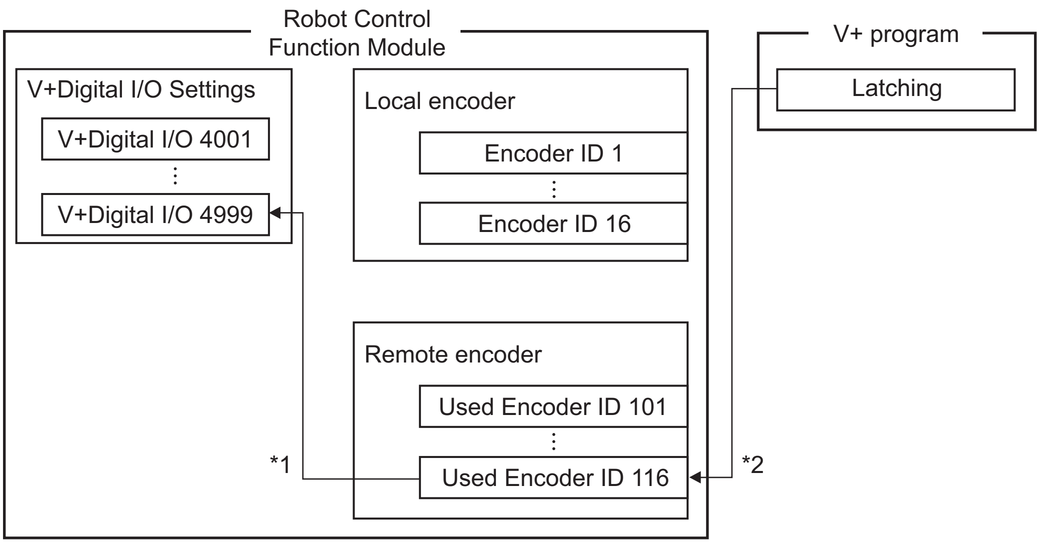
Parameter name	Function	Setting range
Used Encoder ID	Set the encoder ID that is used when a remote encoder is accessed from the V+ program.	101 to 116

Parameter name	Function	Setting range
Motion Control Axis	Select the motion control axis that is accessed as a remote encoder.	0 to 63*1
Position Type Selection	Select the type of position that is accessed as a remote encoder.	Actual position or command position
Latch Signal Number	The V+Digital I/O used as an external trigger when a latch occurs. *2	±4001 to 4999 +: Latching when a signal changes to TRUE. -: Latching when a signal changes to FALSE.

*1. The setting range is limited by the maximum number of controlled axes for each CPU Unit model. Refer to *1-3-2 Performance Specifications* on page 1-7 for details.

*2. Up to eight signals that is set in **V+Digital I/O Settings** can be specified for latch signal numbers. In addition, the latch trigger condition can be specified depending on the sign.
Refer to *3-2-2 V+Digital I/O Settings* on page 3-4 for information on the **V+ Digital I/O Settings**.

The following figure shows an overview of remote encoder latching.



*1. Specify the V+Digital I/O that is set in **V+Digital I/O Settings** for the latch signal.
*2. Can be used from the V+ program in the same way as local encoders.

Used Encoder ID

Set the encoder ID that is used when a remote encoder is accessed from the V+ program.
When you access an encoder value of the remote encoder or acquire latch results from the V+ program, set an used encoder ID as an argument for the V+ keyword to select the encoder that you use.
The used encoder ID is in the range of 101 and 116.

Motion Control Axis

Specify the motion control axis accessed from the V+ program as a remote encoder. Specify the **Axis Number** axis parameter.
You can use all axis types including servo axes, virtual servo axes, encoder axes, and virtual encoder axes.

Position Type Selection

Select the position type of motion control axis accessed from the V+ program as a remote encoder.
The following table shows the applicable position type that depends on the axis type of motion control axis.

Axis type	Position type	
	Command position	Actual position
Servo axis	Applicable	Applicable
Virtual servo axis	Applicable	Applicable*1
Encoder axis	Not applicable	Applicable
Virtual encoder axis	Not applicable	Applicable*2

*1. For a virtual servo axis, the actual position is the same as the command position.
However, there is sometimes calculation error because processing is performed with long reals in the MC Function Module.

*2. This is used when there is no actual encoder.

Latch Signal Number

Set the latch signal number and latch trigger conditions when you acquire the latch result of a remote encoder from the V+ program.

The V+Digital I/O that are set in **V+Digital I/O Settings** are used as the latch numbers for the remote encoder .
Whether the latch signal changes to TRUE or FALSE can be selected for latch trigger conditions by adding the positive or negative sign to a latch signal number.
Up to eight latch signal numbers can be set for a encoder that you use, but they cannot be omitted.
Refer to *3-2-2 V+Digital I/O Settings* on page 3-4 for information on the **V+Digital I/O Settings**.

Precautions for Correct Use

If the **Count Mode** of a motion control axis is set to **Rotary Mode**, make the ring counter setting as follows.

- Modulo maximum position setting value: 8,388,608
- Modulo minimum position setting value: -8,388,608

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

3-3 Robot Setting Parameters

The robot setting parameters provide settings for OMRON robots controlled in the RC Function Module.
There are robot setting parameters for each OMRON robot to control.

3-3-1 Robot Setting Parameters

Use the Sysmac Studio to set the robot setting parameters for each OMRON robot.
Refer to the *Sysmac Studio Robot Integrated System Building Function with Robot Integrated CPU Unit Operation Manual (Cat. No. W595)* for details on how to set the robot setting parameters.

Classification	Parameter name	Reference
Robot Basic Settings	Robot Number	*3-3-2 Robot Basic Settings* on page 3-9
	Robot Device Assignment	

3-3-2 Robot Basic Settings

Set the selection of the EtherCAT slave device for the OMRON robot.

Parameter name	Function	Setting range	Default
Robot Number *1	Set the number that is used when a OMRON robot is accessed from the V+ program.	1 to 8	1
Robot Device Assignment	Select the EtherCAT slave device corresponding to the OMRON robot.	*2	---

*1. You cannot change the robot number.
*2. The OMRON robot that is not assigned, among the OMRON robots that exist on the EtherCAT, can be set.

Robot Number

The robot numbers are automatically set in the order that the OMRON robots are created, and you cannot change the numbers.

Refer to *Robot Variables* on page 2-9 for information on the relationship between the robot numbers and _RC_RBT[0] to _RC_RBT[7], the robot variable names in the system-defined variables.
Also, refer to *Robot I/O Variables* on page 2-10 for information on the relationship between the robot numbers and _RC_RBT_IO[0] to _RC_RBT_IO[7], the robot I/O variable names in the system-defined variables.

Robot Device Assignment

Select the EtherCAT slave device corresponding to the OMRON robot.

4

Program Design of Robot Control

This section describes the program design of the robot control.

4-1 Introduction

The Robot Integrated CPU Unit can perform robot control, in addition to sequence control and motion control.

There are two methods of robot control as described below.

- Write robot control instructions in a sequence control program to perform robot control.
  Outputs from robot control instructions and system-defined variables for robot control are refreshed in the same control period as that of I/O control or motion control.
  This method is suitable for controlling an OMRON robot with other devices.
- Write a V+ program and execute it to perform robot control.
  The V+ language provides more functions related to the robot control than robot control instructions of the sequence control program.
  In addition, because V+ programs are executed line by line, it is easy to write a sequence of OMRON robot motions.
  This method is suitable mainly for programming OMRON robot motions.

Moreover, you can start and stop program execution and share variables between a sequence control program and a V+ program.

4-2 Sequence Control Program

To control a robot from a sequence control program, use robot control instructions defined as function blocks.
Robot control instructions are also used to start and stop execution of a V+ Task.

In addition, to read information from the OMRON robot and control the robot built-in I/O, use system-defined variables for robot control defined as system-defined variables.
It is possible to share the interlock and variable values between the programs using the shared variables between a sequence control program and a V+ program.

4-2-1 Robot Control Instructions

The following table lists robot control instructions.

Type	Function
Common commands	Instructions to start/abort the V+ Task execution and read status information
	Instructions to convert a coordinate system for use with NJ Robotics functions
Robot commands	Instructions to directly control the OMRON robot

Refer to *Section 6 Robot Control Instructions* on page 6-1 for details on the robot control instructions.

In addition, the following system control instructions are used to reset errors and read status information from the RC Function Module.

Type	Function
System control instructions	Instructions to reset all current errors from the RC Function Module
	Instructions to read the highest level current errors from the RC Function Module

Refer to *Section 10 System Control Instructions* on page 10-1 for details on the system control instructions.

4-2-2 Timing Charts for Robot Control Instructions

This section describes the basic timing charts for the robot control instructions.
Refer to the individual instruction for details on the unique input variables and output variables for each instruction.

Execute-type Instructions

The instruction starts when *Execute* changes to TRUE.
Busy (Executing) changes to TRUE when the instruction is acknowledged.
If the processing is completed normally, *Busy* changes to FALSE and *Done* changes to TRUE.
The processing completed normally means that the command from the instruction to RC Function Module is completed.

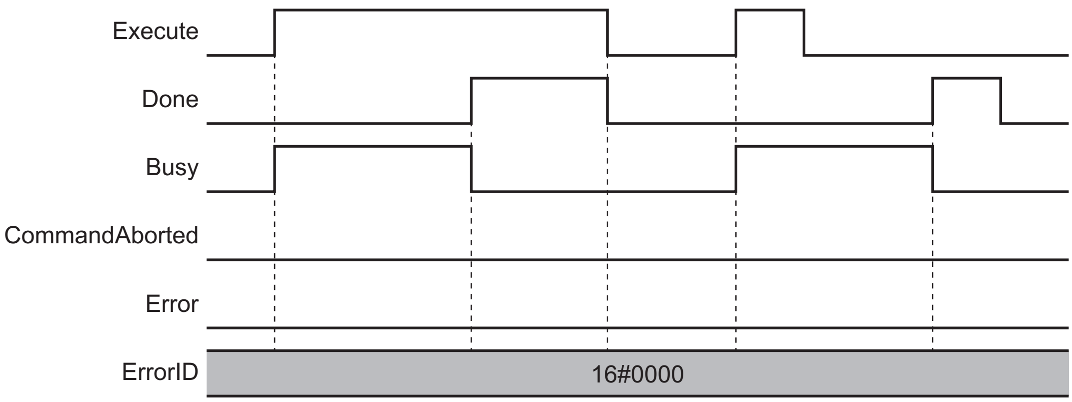
If the same instances of instructions are executed consecutively, wait for more than one task period after *Done* for previous execution changes to FALSE, and then *Execute* for next instruction changes to TRUE.

If the processing is interrupted, *Busy* changes to FALSE.
For the instructions with *CommandAborted* output variable, *CommandAborted* changes to TRUE at the same time.

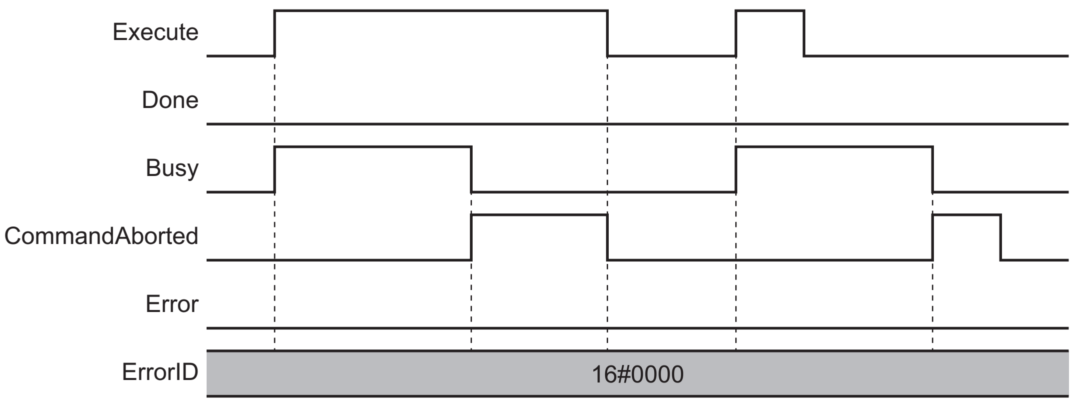
When an error occurs, *Error* changes to TRUE and *Busy* changes to FALSE.
If either *Done* or *CommandAborted* changes to TRUE while *Execute* is TRUE, both *Done* and *CommandAborted* change to FALSE when *Execute* changes to FALSE.
If *Done* and *CommandAborted* change to TRUE while *Execute* is FALSE, both *Done* and *CommandAborted* change to TRUE for one task period.

When an error occurs, the error code for *ErrorID* (Error Code) is set.
The *Error* and *ErrorID* (Error Code) are retained after *Execute* changes to FALSE. When the error is reset, *Error* changes to FALSE and 16#0000 is set in *ErrorID* (Error Code).

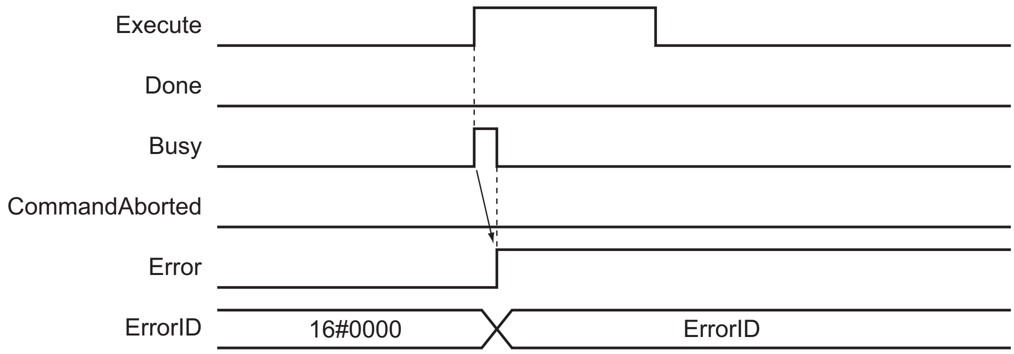
● When the Instruction Ended Normally



● When the Instruction is Aborted



● **When an Error Occurred**



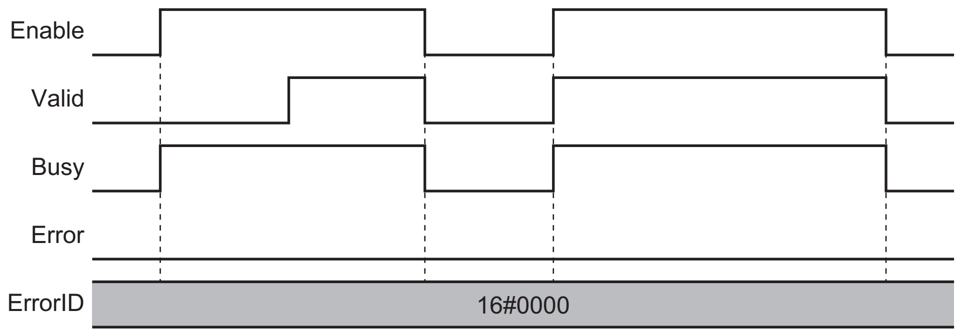
Enable-type Instructions

The instruction is executed while *Enable* is TRUE.
When the instruction is acknowledged, *Busy* (Executing) changes to TRUE, and then *Valid* (Enabled) changes to TRUE while output value is calculated.

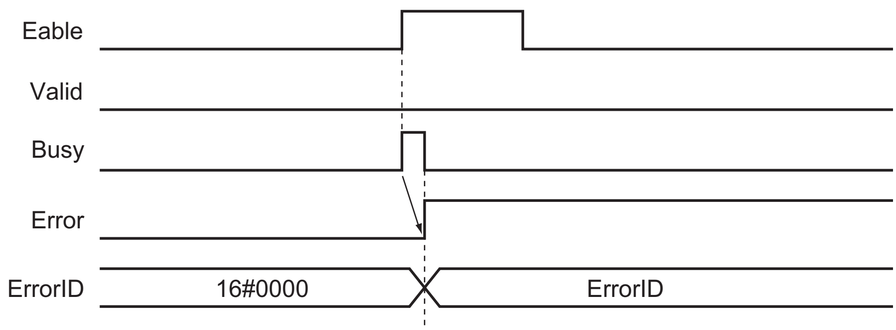
When an error occurs, *Error* changes to TRUE and *Busy* (Executing) and *Valid* change to FALSE.
Even if *Enable* changes from TRUE to FALSE, *Error* remains TRUE.

When an error occurs, the error code for *ErrorID* (Error Code) is set.
When *Error* changes from TRUE to FALSE, 16#0000 is set in *ErrorID* (Error Code).

● **When the Instruction Ended Normally**



● **When an Error Occurred**



4-2-3 System-defined Variables for Robot Control

The following table lists system-defined variables for robot control.
Each of these variables is updated at a frequency of the control period of primary periodic task for the Robot Integrated CPU Unit.

Variable	Function
Robot control common variable	Monitors the common status of the RC Function Module.
Robot variables	Monitor the status of each OMRON robot.
	Monitor the actual position of the TCP and the actual position of the joint for each OMRON robot.
Robot I/O variables	Monitor the built-in I/O of each OMRON robot.
	Output to the output port of the built-in I/O of each OMRON robot.

Refer to *2-4 System-defined Variables for Robot Control* on page 2-8 for details on each variable.

4-2-4 Execution Control for V+ Program

You can execute robot control instructions in a sequence control program to execute V+ tasks to which V+ programs are assigned.
You can also assign V+ programs to V+ tasks to use V+ programs that are written with OMRON robot motions as subroutines in a sequence control program.

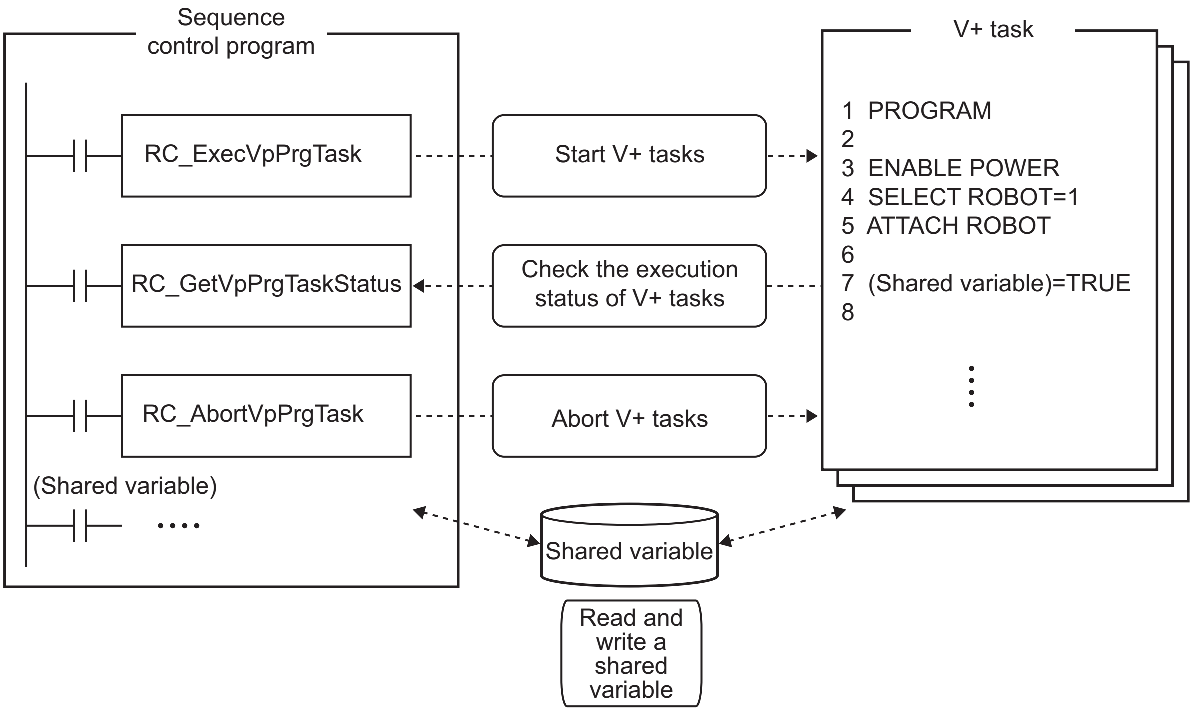
You can also use robot control instructions to obtain the status of a V+ task or abort execution of a V+ task.

V+ Task Control from Sequence Control Program

This section describes the V+ task control from the sequence control program in the Robot Integrated CPU Unit.

You can control the following V+ tasks from a sequence control program. Refer to *Section 7 Variables and Instructions* on page 7-1 for details on each instruction.

Function	Description	Instruction	Reference
Execute V+ Task	Starts execution of the specified V+ task.	RC_ExecVpPrgTask	page 8-2
Abort V+ Task	Aborts execution of the specified V+ task.	RC_AbortVpPrgTask	page 8-6
Get V+ Task Status	Reads the specified V+ task status.	RC_GetVpPrgTaskStatus	page 8-8



● Starting V+ Tasks

The sequence control program executes the RC_ExecVpPrgTask (Execute V+ Task) instruction to start execution of a V+ task.

Refer to *RC_ExecVpPrgTask* on page 8-2 for details on the instruction.

● Aborting V+ Tasks

The sequence control program executes the RC_AbortVpPrgTask (Abort V+ Task) instruction to request to abort a V+ task.

Refer to *RC_AbortVpPrgTask* on page 8-6 for details on the instruction.

● Checking Execution Status of V+ Tasks

The sequence control program executes the RC_GetVpPrgTaskStatus (Get V+ Task Status) instruction to check execution status of a V+ task.

Refer to *RC_GetVpPrgTaskStatus* on page 8-8 for details on the instruction.

4-2-5 Shared Variables with V+ Program

The shared variables refer to the function that shares global variables in a sequence control program with a V+ program.

The shared variables can be read and written from either the sequence control program or the V+ program.

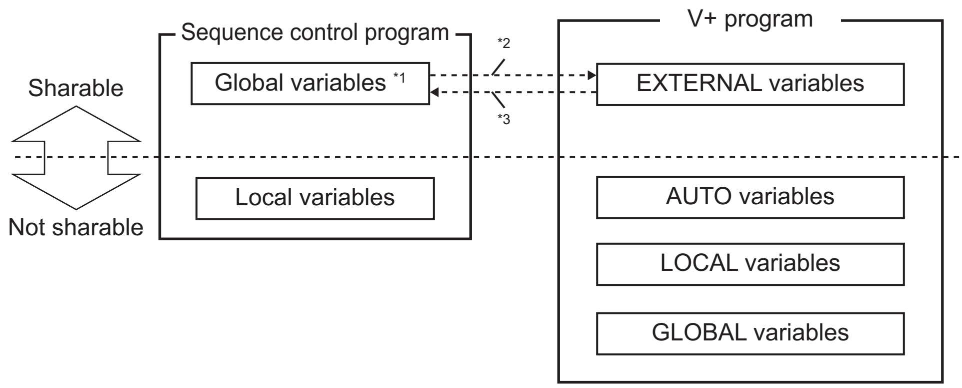
This allows you to exchange information and synchronize the timing of processing between the sequence control program and V+ program in execution.

EXTERNAL variables are external variables in the V+ program. The EXTERNAL variables can be read and written for the global variables in the sequence control program.

Shared variables can be used by making two settings as described below.

- Define a shared variable as a global variable in the sequence control program in the global variable table of the Sysmac Studio.
- Define the variable with the same name in the V+ program as the EXTERNAL variable.

The mechanism of shared variables is given below.



*1. The system-defined variables cannot be shared.
*2. Reading a value: Access to an external variable with the same name
*3. Writing a value: Update of an external variable with the same name

Refer to the *eV+3 User's Manual (Cat. No. I651)* and the *eV+3 Keyword Reference Manual (Cat. No. I652)* for information on the EXTERNAL variable.

Precautions for Correct Use

- A variable name in the V+ program can be used with lowercase letters only.
  A variable name in the sequence control program can be used with uppercase and lowercase letters. However, it is not case sensitive.
  Therefore, even if the global variable name that is defined in the sequence control program includes uppercase letters, the variable can be shared with the variable, whose name is used with lowercase letters, in the V+ program.
- The same variable name cannot be used between the variable defined as the EXTERNAL variable and the GLOBAL variable in the V+ program.

Shared Variable Setting Procedure

The variable to share is registered in the global variable table of the sequence control program. Then, you can use variables by sharing them when you make the EXTERNAL declaration to a variable with the same name in the V+ program.

Concurrency of Shared Variables

The concurrency of variable values is not ensured if more than one shared variable is read or written continuously in the V+ program.
To ensure concurrency of shared variables, create a shared variable to use as a flag separately and control the timing to read and write shared variables.

Example: When values of array variables are accessed by a V+ program with concurrency maintained

Sequence control program

```
IF NOT gisupdated THEN
    gsharedvar[1]: = var1;
    gsharedvar[2]: = var2;
    gisupdated: = TRUE;
END_IF;
```

Used as a flag

Global variables

gsharedvar	Array[0..10] OF LREAL
gisupdated	BOOL

V+ program

```
EXTERNAL gsharedvar[]
EXTERNAL gisupdated
LOCAL var1, var2

IF gisupdated THEN
    var1=gsharedvar[1]
    var2=gsharedvar[2]
    gisupdated=FALSE
END
```

Data Type of Shared Variables

The shared variables have sharable or not sharable data type.

The following table gives whether a variable is sharable or not depending on the data type. Be sure to check that the variable data is within the valid range. Exceeding the valid range results in an undefined value.

(○: Sharable, ---: Not sharable)

Data type		Data size	Sharable or not	Remarks
Basic data types	BOOL	16 bits	○	
	BYTE	8 bits	---	
	WORD	16 bits	---	
	DWORD	32 bits	---	
	LWORD	64 bits	---	
	SINT	8 bits	---	
	INT	16 bits	○	Valid range of data: -32768 to 32767
	DINT	32 bits	---	
	LINT	64 bits	---	
	USINT	8 bits	---	
	UINT	16 bits	○	Valid range of data: 0 to 65535
	UDINT	32 bits	---	
	ULINT	64 bits	---	
	REAL	32 bits	○	Valid range of data: -3.402823×10^{38} to 3.402823×10^{38} (7 significant digits)
	LREAL	64 bits	○	Valid range of data: $-1.79769313486231 \times 10^{308}$ to $1.79769313486231 \times 10^{308}$ (15 significant digits)
	TIME	64 bits	---	
	DATE	64 bits	---	
	TIME_OF_DAY	64 bits	---	
	DATE_AND_TIME	64 bits	---	

Data type		Data size	Sharable or not	Remarks
	STRING	Number of single-byte characters × 8 bits	---	
Other data types	Structure	---	---	
	Union	---	---	
	Enumeration	---	---	
	Fixed length array	Number of elements × Data size of basic data type	○	A variable can be shared if all of the following conditions are met. • Basic data type is sharable • One-dimensional array only • Array with 100 elements or less

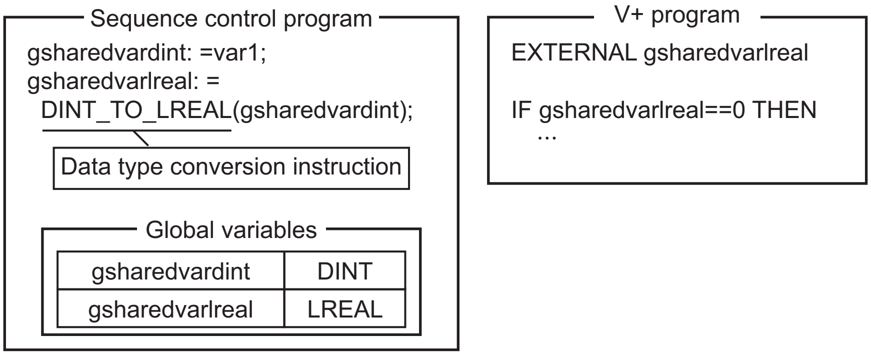
You can use the Data type conversion instructions in the sequence control program to share variables of data types that cannot be shared.
To share a DINT variable, for example, use the DINT_TO_REAL instruction to convert it to REAL data, so the variable can be shared. Note that the Data type conversion instructions have specific valid ranges for both the data types before and after conversion. Check the specifications of the Data type conversion instruction that you use before use.
Refer to *Data Type Conversion Instructions* in the *NJ/NX-series Instructions Reference Manual (Cat. No. W502)* for details on the Data type conversion instruction.

For shared variables, a V+ Program Error (96040000 hex) will occur in the following cases.
- A variable that is not defined in the sequence control program is accessed from the V+ program.
- A variable defined with the data type that cannot be shared is accessed from the V+ program.
- An attempt was made to write the variable that is declared as *Read Only* in the sequence control program from the V+ program.
- A subscript (index) for an array is within the range defined in the V+ program, but it is out of range defined in the sequence control program.
- A subscript (index) that is out of range of which the array is declared in the sequence control program is accessed from the V+ program.
  For example, for the array that ARRAY[10..100] is declared in the sequence control program, access to ARRAY[0] in the V+ program.

● If Data Type That Cannot be Shared was Defined as EXTERNAL Variable

If the global variable in the sequence control program is declared with a data type that cannot be shared and you define it as the EXTERNAL variable in the V+ program, a system event will occur during execution of the V+ program. In this case, it does not affect other function modules and execution of the sequence control program.
When you share variables of data types that cannot be shared, use the Data type conversion instructions in the sequence control program to convert to variables of data types that can be shared.

Example: When the value of a DINT variable is accessed by a V+ program



Deleting Shared Variables

When you delete a shared variable after you set the variable to be shared, delete both the global variable in the sequence control program and the EXTERNAL declaration.

Precautions for Correct Use

- If you delete only the global variable in the sequence control program, a system event will occur when the EXTERNAL variable is defined during execution of the V+ program.
- If you delete only the EXTERNAL declaration, the global variable in the sequence control program can be used only in the sequence control program.
  In addition, the variable is used for an independent variable that is not related in the global variable in the sequence control program within the V+ program.
  In other words, the variables with the same variable name are used as different variables individually in the sequence control program and in the V+ program.

Restrictions on Variable Names of Shared Variables

The variable names of shared variables are subject to the following restrictions.

Item	Restriction
Maximum length of variable name	15 characters
Applicable characters for variable names	0 to 9, a to z, ".", and "," The first character must be a to z.

Precautions for Correct Use

- If you define the EXTERNAL variable in the V+ program, do not define a variable with the same name as the AUTO, LOCAL, or GLOBAL variable in the same V+ program.
  When the same name as the EXTERNAL variable is defined, a system event occurs in the RC Function Module.

V+ program

```
EXTERNAL gprogramstate
GLOBAL gprogramstate

gprogramstep=0

WHILE TRUE DO
   IF (gprogramstep==1) THEN
      MOVE p1
      gprogramstep=2
   END
END
```

A variable with the same name already exists.*1

*1. A system event occurs.

- If you define the EXTERNAL variable in the V+ program, define a variable with the same name in the global variable table of the sequence control program.
  When a variable with the same name does not exist in the global variable table of the sequence control program, a system event occurs in the RC Function Module.
  Even if the system event occurred, it does not affect other function modules and execution of the sequence control program.

4-2-6 Using Shared Variables with V+ Programs

This section describes how to use the shared variables that share variables between a sequence control program and a V+ program.

Using Shared Variables

The methods to share variables between a sequence control program and a V+ program are described below.

- Synchronizing the control timing with interlock between programs
- Controlling the processing sequence by sharing values of variables between programs

● Synchronizing the Control Timing with Interlock Between Programs

You can use the shared variables if a sensor input is detected with a sequence control program and the V+ program operation is started.

An example of a program is provided below.
In this example, when data E001_In_Bit00, which has been input to bit 0 of the Digital Input Unit (EtherCAT slave), turns ON in the sequence control program, the value of shared variable *gsensoron* changes to TRUE. Then, the V+ program starts control operation.

Sequence control program

```
IF(E001_In_Bit00=TRUE) THEN
  // Interlock
  gsensoron: =TRUE;
END_IF;
```

V+ program

```
EXTERNAL gsensoron

gsensoron=FALSE

WHILE TRUE DO
    ; Interlock
IF gsensoron THEN
    MOVE p1
    gsensoron=FALSE
END
```

● Controlling the Processing Sequence by Sharing Values of Variables Between Programs

The values of variables can be shared between a sequence control program and a V+ program. This allows you to control the sequence of processing in each process.

An example of a program is provided below.

Sequence control program

```
CASE (gprogramstep) OF
  0:
    // Initialize

    gprogramstep: =1;
  1:
    // Wait Robot
  2:
    // Finalize

    gprogramstep: =3;
END_CASE;
```

V+ program

```
EXTERNAL gprogramstep

gprogramstep=0

WHILE TRUE DO
    IF (gprogramstep==1) THEN
        MOVE p1
        gprogramstep=2
    END
END
```

4-3 V+ Program

A control program that is written in the V+ language is called V+ program.
The Robot Integrated CPU Unit executes V+ programs to assign to V+ tasks.
It is also possible to control the OMRON robots only with V+ programs.
This section provides information related to the V+ program in the Robot Integrated CPU Unit.

Refer to the *eV+3 User's Manual (Cat. No. I651)* and the *eV+3 Keyword Reference Manual (Cat. No. I652)* for details on the descriptions in V+ language.
Refer to the *Sysmac Studio Robot Integrated System Building Function with Robot Integrated CPU Unit Operation Manual (Cat. No. W595)* for how to assign V+ programs to V+ Tasks.

Precautions for Correct Use

If the digital I/O control is performed for the EtherCAT slave devices that are not OMRON robots or the NX Units on the EtherCAT Coupler Unit in the V+ program, confirm that the *_EC_PDSlavTbl* (Process Data Communicating Slave Table) system-defined variable changes to TRUE in the sequence control program.
Confirm that the process data communications of the EtherCAT slaves is enabled before the digital I/O control is performed for the EtherCAT slave devices or the NX Units on the EtherCAT Coupler Unit in the V+ program.

4-3-1 Overview of V+ Programs

The Robot Integrated CPU Unit can start or stop V+ tasks from a sequence control program, a V+ program, or the Sysmac Studio.
The RC Function Module analyzes the V+ program assigned to V+ tasks line by line and sends instructions (V+ keywords) and command values to the OMRON robot connected to the EtherCAT network.
After receipt of an instruction (V+ keyword) or a command value, the OMRON robot performs path calculation and controls the robot arm. The present value and status of the OMRON robot are sent through the EtherCAT network to the RC Function Module.
It is possible to share the interlock and variable values between the programs using the shared variables between a sequence control program and a V+ program.

V+ programs can automatically start when the power supply to the Robot Integrated CPU Unit is turned ON.
You can control the OMRON robots only with V+ programs using the automatic start.
Refer to the *eV+3 User's Manual (Cat. No. I651)* and the *eV+3 Keyword Reference Manual (Cat. No. I652)* for details on the automatic start function.

4-3-2 Control of V+ Tasks

Refer to the *eV+3 User's Manual (Cat. No. I651)* and the *eV+3 Keyword Reference Manual (Cat. No. I652)* for how to control V+ Tasks with V+ programs.

4-3-3 V+Digital I/O Settings from V+ Program

To perform digital I/O control for the EtherCAT slave devices or the NX Units on the EtherCAT Coupler Unit in the V+ program, you need to make the settings in the Robot Common Parameter Settings Tab Page on the Sysmac Studio.

Refer to *3-2-2 V+Digital I/O Settings* on page 3-4 for details on the V+Digital I/O Settings.

4-3-4 Latching from V+ Program

Refer to *5-3 Latching* on page 5-6 for information on the latching from the V+ program.

4-4 Debugging Program

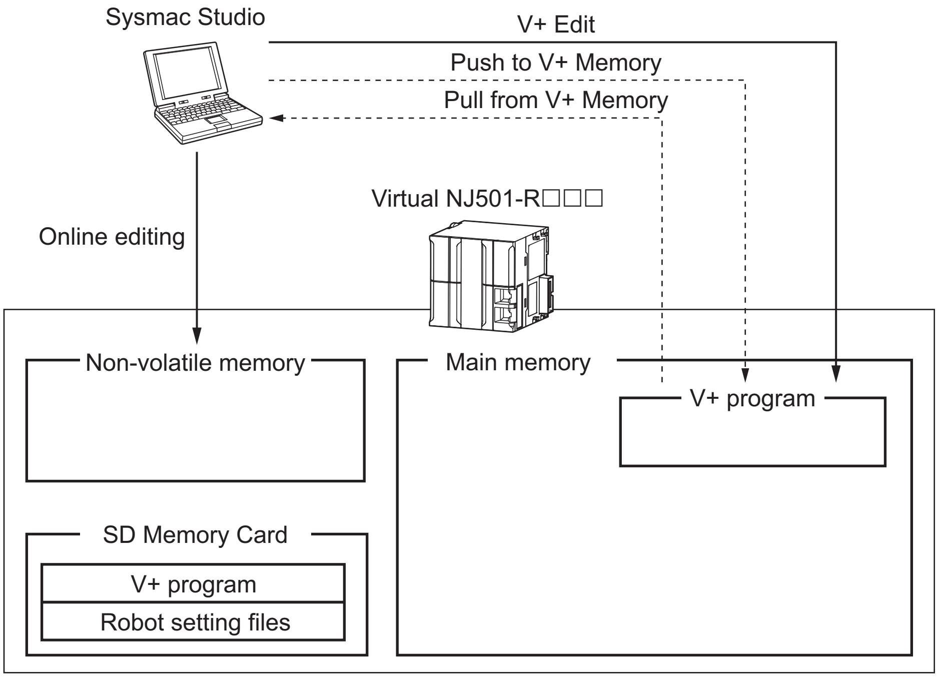
This section describes the offline debugging from the simulation operation and the online debugging from the online connection.

The simulation operation for the Robot Integrated CPU Unit can be performed only in EMULATION mode.

Refer to the *NJ/NX-series CPU Unit Software User's Manual (Cat. No. W501)* for information on the offline debugging in the sequence control program.

4-4-1 Offline Debugging

This section describes the offline debugging from the simulation operation.
When the simulation operation starts, the V+ emulator is executed on the Sysmac Studio.
When the Sysmac Studio connects to the V+ emulator, the following operations are possible.



Function	Description
Online editing	Edits directly to the sequence control program in the built-in non-volatile memory.
V+ Edit	Edits directly to the V+ program in the main memory.
Push to V+ Memory	Transfers the V+ program on the Sysmac Studio to the main memory.
Pull from V+ Memory	Transfers the V+ program in the main memory to the Sysmac Studio.
Check V+ Memory	Compares the V+ program in the main memory and the V+ program on the Sysmac Studio.

The simulation operation for the Robot Integrated CPU Unit can be performed only in EMULATION mode.
If the EMULATION mode is started, click the **Enable/Disable EMULATION mode** icon in the toolbar on the main window.

Refer to the *Sysmac Studio Robot Integrated System Building Function with Robot Integrated CPU Unit Operation Manual (Cat. No. W595)* and the *Sysmac Studio 3D Simulation Function Operation Manual (Cat. No. W618)* for details on the simulation.

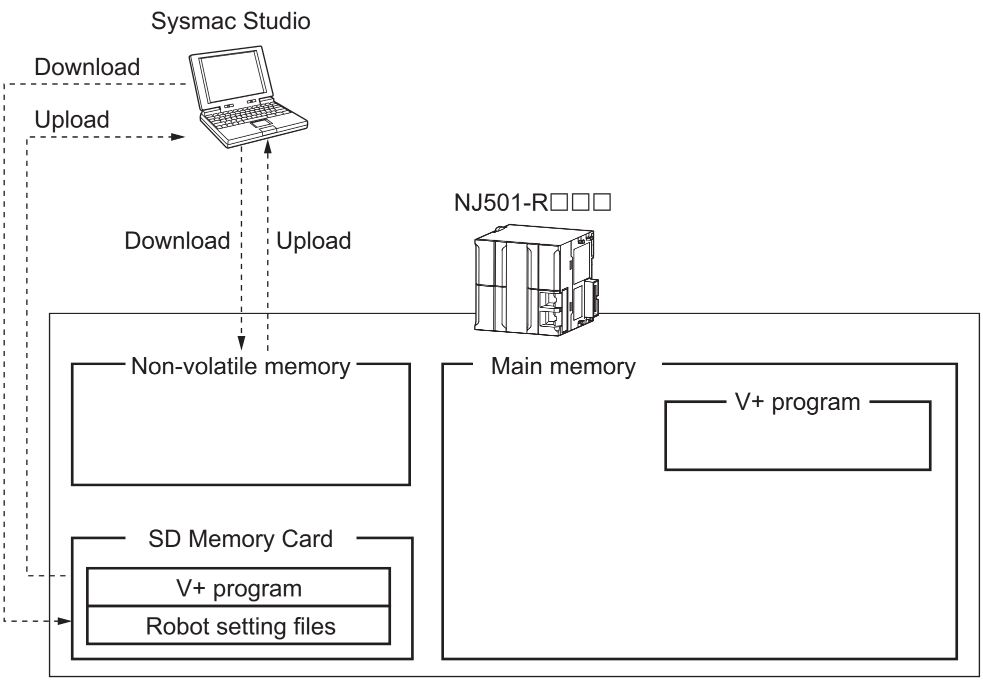
Precautions for Correct Use

V+ Edit is disabled during execution of V+ program. Stop the V+ program before executing the V+ Edit.

4-4-2 Transferring Settings and Programs

This section explains the functionality to transfer the settings and programs between the Sysmac Studio and the built-in non-volatile memory in the Robot Integrated CPU Unit or SD Memory Card.

When the Sysmac Studio connects online to the Robot Integrated CPU Unit, the following operations are possible.



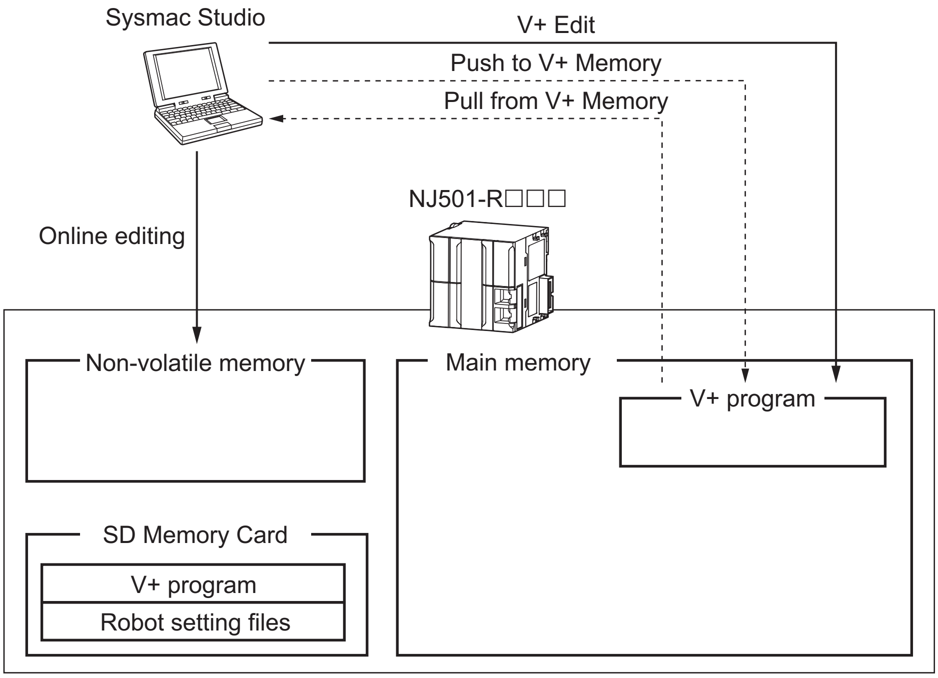
Function	Description
Downloading data	Transfers the settings and programs from the Sysmac Studio to the built-in non-volatile memory in the Robot Integrated CPU Unit or SD Memory Card.
Uploading data	Transfers the settings and programs from the built-in non-volatile memory in the Robot Integrated CPU Unit or SD Memory Card to the Sysmac Studio.
Verifying data	Compares the settings and programs between the built-in non-volatile memory in the Robot Integrated CPU Unit or SD Memory Card and the Sysmac Studio.

Refer to the *NJ/NX-series CPU Unit Software User's Manual (Cat. No. W501)* for information on the downloading and uploading data.

4-4-3 Online Debugging

This section describes the online debugging.

When the Sysmac Studio connects online to the Robot Integrated CPU Unit, the following operations are possible.



Function	Description
Online editing	Edits directly to the sequence control program in the built-in non-volatile memory.
V+ Edit	Edits directly to the V+ program in the main memory. *1
Push to V+ Memory	Transfers the V+ program on the Sysmac Studio to the main memory.
Pull from V+ Memory	Transfers the V+ program in the main memory to the Sysmac Studio.
Check V+ Memory	Compares the V+ program in the main memory and the V+ program on the Sysmac Studio.

*1. When you perform online debugging, set **V+ Edit Mode** to **Start** in the Sysmac Studio and the V+ Program Editor Tab Page can be edited.

The online connection to the Robot Integrated CPU Unit is not possible in EMULATION mode.

Refer to the *Sysmac Studio Version 1 Operation Manual (Cat. No. W504)* and the *Sysmac Studio Robot Integrated System Building Function with Robot Integrated CPU Unit Operation Manual (Cat. No. W595)* for details on the online connection.

Precautions for Correct Use

- The settings are required to save the V+ program in the SD Memory Card.
  Refer to the *Sysmac Studio Robot Integrated System Building Function with Robot Integrated CPU Unit Operation Manual (Cat. No. W595)* for details on the save settings.
- V+ Edit is disabled during execution of V+ program. Stop the V+ program before executing the V+ Edit.
- Unintended operation may occur for the online editing in the robot control instruction. Confirm safety before use.

4-5 States and State Transition

This section describes the states and state transition of the Robot Integrated CPU Unit and OMRON robots.

4-5-1 States of the Robot Integrated CPU Unit

This section describes the operation for states of the Robot Integrated CPU Unit when the operating mode of the OMRON robot is in Auto mode.

Refer to *Operating Mode of the OMRON Robot* on page 4-21 for information on operating mode of the OMRON robot.
Refer to *CPU Unit Status* in the *NJ/NX-series CPU Unit Software User's Manual (Cat. No. W501)* for information on each state of the CPU Unit.

Precautions for Correct Use

If the operating mode of the Robot Integrated CPU Unit is changed to PROGRAM mode, the operations and control for the OMRON robot that is controlled with V+ program and I/O continue.
The devices that are controlled with the sequence control program and other robots stop. Always confirm safety before you change the operating mode to prevent interference between devices.

Additional Information

When the following operation is executed during operation of the V+ task assigned to the V+ program in which EXTERNAL V+ keyword is executed, the V+ task is stopped.
- Download operation with enabled _DeviceOutHoldCfg (Device Output Hold Configuration) system-defined variable

However, it does not affect for _DeviceOutHoldCfg (Device Output Hold Configuration) system-defined variable.

Operations in Sequence Control Program

The sequence control program is executed in RUN mode.

If there are no errors in the RC Function Module, the control for all of OMRON robots attached from the sequence control program and inputs to and outputs from the robot built-in I/O are possible.
Even if an error occurred in the RC Function Module, the execution of the sequence control program continues.

● Differences in Event Levels

The operation differs depending on the event level that occurs in the RC Function Module as described below.

- If a partial fault level error occurs in the RC Function Module, the control for all of OMRON robots and inputs to and outputs from the robot built-in I/O are not possible.

- If a minor fault level error that "robot" is given for the source details occurs in the RC Function Module, the control for the OMRON robot and outputs from the robot built-in I/O are not possible. Only inputs to the robot built-in I/O are possible.
- If an observation or information occurs in the RC Function Module, the control for all of OMRON robots attached from the sequence control program including the OMRON robot and inputs to and outputs from the robot built-in I/O are possible.

The operation when the control for the OMRON robots and inputs to and outputs from the built-in I/O are not possible depends on the specifications of OMRON robot. Refer to the OMRON robot manuals for details.

Refer to the *NJ/NX-series CPU Unit Software User's Manual (Cat. No. W501)* and the *NJ/NX-series CPU Unit Built-in EtherCAT Port User's Manual (Cat. No. W505)* for information on the EtherCAT slaves that are not OMRON robots.

Operations in V+ Program

The V+ programs are executed in both RUN mode and PROGRAM mode.
The OMRON robot continues to move when the operating mode changes from RUN mode to PROGRAM mode.

If there are no errors in the RC Function Module, the control for the OMRON robots attached from the V+ program and inputs to and outputs from the robot built-in I/O are possible.

● Differences in Event Levels

The operation differs depending on the event level that occurs in the RC Function Module as described below.

- If a partial fault level error occurs in the RC Function Module, all of V+ programs stop.
  The control for all of OMRON robots and inputs to and outputs from the robot built-in I/O are not possible.
  Note that the values of system-defined variables can be changed, but inputs to and outputs from the robot built-in I/O are not reflected.
- If a minor fault level error occurs in the RC Function Module, the operation continues for the V+ programs.
  Therefore, the control for the OMRON robot and inputs to and outputs from the robot built-in I/O are possible.
- If an observation level error or an information level error that is not V+ Program Error (96040000 hex) occurs in the RC Function Module, the operation continues for the V+ programs.
  Therefore, the control for the OMRON robot and inputs to and outputs from the robot built-in I/O are possible.
- If the V+ Program Error (96040000 hex) in an information level occurs, the V+ program in which the OMRON robot is attached stops.
  Therefore, the control for the OMRON robot and inputs to and outputs from the robot built-in I/O are not possible.

The operation when the control for the OMRON robots and inputs to and outputs from the built-in I/O are not possible depends on the specifications of OMRON robot. Refer to the OMRON robot manuals for details.

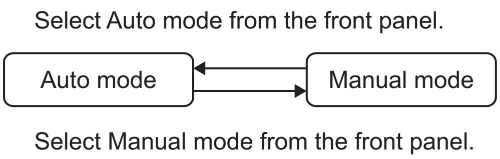
Refer to the *NJ/NX-series CPU Unit Software User's Manual (Cat. No. W501)* and the *NJ/NX-series CPU Unit Built-in EtherCAT Port User's Manual (Cat. No. W505)* for information on the EtherCAT slaves that are not OMRON robots.

4-5-2 States of the OMRON Robots

This section describes the types of operating mode and operations for the OMRON robots, the control status of power supply, and the transition of operation states.

Operating Mode of the OMRON Robot

The OMRON robot has two operating modes, Auto mode and Manual mode, each of which represents its control status.
The operating mode of the OMRON robot can be changed on the front panel that is connected to the OMRON robot.



When the operating mode of the OMRON robot is changed to either Auto mode or Manual mode, the OMRON robot high power turns OFF.

Additional Information

You can monitor the operating mode of the OMRON robot from the Sysmac Studio. Refer to the *Sysmac Studio Robot Integrated System Building Function with Robot Integrated CPU Unit Operation Manual (Cat. No. W595)* for details.

● Manual Mode

In Manual mode, the velocity and torque of the OMRON robots are restricted by the system in order to reduce the risk of operators who work within the range of motion for the OMRON robots.
Therefore, in Manual mode, you cannot control the OMRON robots by the sequence control program and V+ program.
Only the teaching pendant connected to the OMRON robot can be used.

The velocity of the OMRON robots is restricted to 250 mm/s or less in Manual mode.
Refer to the OMRON robot manuals for details.
In addition, under the responsibility of the user, determine whether the specifications of Manual mode are allowed under the regulations of the region where the OMRON robots are used.

Refer to the *Teaching Pendant T20 User's Manual (Cat. No. I601)* for information on the teaching pendant.

● Auto Mode

In Auto mode, the OMRON robots are controlled by the sequence control program and V+ program and can move at full speed.

Additional Information

The OMRON robots can be controlled by a teaching pendant and the jog operation in the Sysmac Studio even in Auto mode only when the OMRON robots are not controlled by the sequence control program or the V+ program.
Refer to the *Teaching Pendant T20 User's Manual (Cat. No. I601)* for information on the teaching pendant.
Refer to the *Sysmac Studio Robot Integrated System Building Function with Robot Integrated CPU Unit Operation Manual (Cat. No. W595)* for information on the jog operation in the Sysmac Studio.

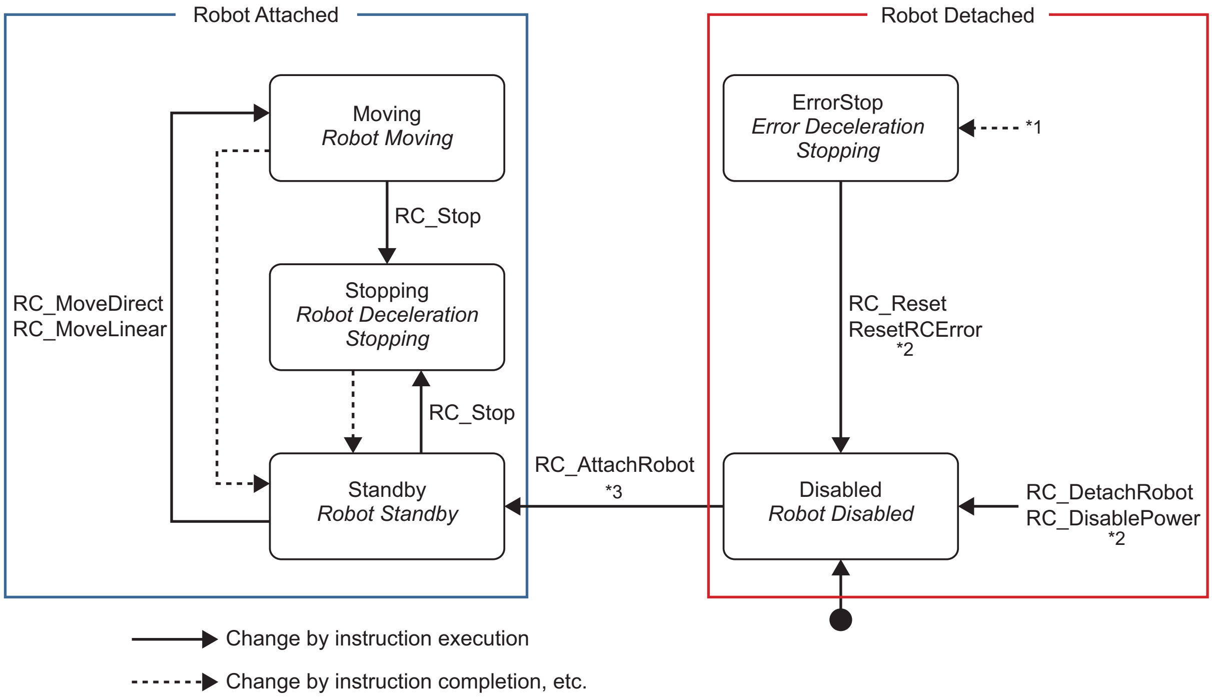
Control Status of Power Supply to OMRON Robot's Motor

High power ON/OFF controls the power supply to the OMRON robot's motor.
When you turn high power ON, the power is supplied to the OMRON robot's motor and the robot is enabled to move. When you turn high power OFF, the power is not supplied to the OMRON robot's motor and the robot does not move.
According to safety regulations, it is mandatory to follow a series of steps to turn high power ON. Refer to the OMRON robot manuals for information on the procedure.

Refer to *RC_EnablePower* on page 9-2 and *RC_DisablePower* on page 9-5 for information on the robot control instructions for high power ON/OFF.

Operation States of the OMRON Robots

While the OMRON robot is moving in Auto mode, when the robot control instruction is executed to the OMRON robot from a sequence control program, the operation state of the OMRON robot is updated.
The following table shows the transition of the OMRON robot's operation state.
Robot control instructions are executed in sequence and the OMRON robot enters one of the states listed in the following table.
Note that the OMRON robot's operation state below shows the transition of a sequence control program. It does not show the transition of a V+ program.



*1. Transition to this state occurs from any state if an OMRON robot error occurs.
For *ErrorStop* (Error Deceleration Stopping) state, the OMRON robot high power turns OFF. The calibration is not completed for the eCobra.

*2. Transition to *Disabled* (Robot Disabled) state occurs from *ErrorStop* (Error Deceleration Stopping) state if an error is reset with the RC_Reset (Reset Robot Error) or ResetRCError (Reset Robot Control Error) instruction. Transition to *Disabled* (Robot Disabled) state with the RC_DetachRobot (Detach Robot) or RC_DisablePower (Disable Robot High Power) instruction occurs from the state other than *ErrorStop* (Error Deceleration Stopping) state.

*3. Transition to *Standby* (Robot Standby) state occurs only if the robot can be attached.
Refer to *RC_AttachRobot* on page 9-10 for details.

State name	Definition
Robot Detached	OMRON robot control by robot control instructions is disabled.
Disabled (Robot Disabled)	A robot can be attached.
ErrorStop (Error Deceleration Stopping)	There is an error in the OMRON robot.
Robot Attached	OMRON robot control by robot control instructions is enabled.
Standby (Robot Standby)	The OMRON robot is attached and stopped.
Moving (Robot Moving)	The RC_MoveDirect (Robot Joint Interpolation) or RC_MoveLinear (Robot Linear Interpolation), which are instructions to control an OMRON robot, is in execution.
Stopping (Robot Deceleration Stopping)	The RC_Stop (Stop Robot) instruction is in execution. This includes when *Execute* is TRUE after the robot is stopped with the RC_Stop (Stop Robot) instruction.

4-5-3 Changing the Operating Mode

An Robot Integrated CPU Unit has two operating modes: PROGRAM mode and RUN mode.
The OMRON robot has two operating modes: Auto mode and Manual mode.
This section describes the operation when each operating mode changes.

Changing Operating Mode of Robot Integrated CPU Unit

This section describes changing operating mode from RUN Mode to PROGRAM Mode, or vice versa.

● Changing from RUN Mode to PROGRAM Mode

When the operating mode of the Robot Integrated CPU Unit is changed from RUN mode to PROGRAM mode, the following will occur.

- The OMRON robot during control in the sequence control program stops immediately.
  The Robot Integrated CPU Unit detaches a robot, and the high power for the OMRON robot will turn OFF.
- The OMRON robot during control in the V+ program continues to move.
- The sequence control program that is in progress is aborted and all of output variables in the robot control instructions are set to their initial values.
  The *CommandAborted* output variable remains FALSE.

● Changing from PROGRAM Mode to RUN Mode

All of output variables in the robot control instructions are set to their initial values.

Changing Operating Mode of OMRON Robot

While the robot has been controlled by a sequence control program or V+ program, when the operating mode of the OMRON robot is changed from Auto mode to Manual mode, the OMRON robot high power turns OFF.

If the operating mode of the OMRON robot is changed to Manual mode, stop the OMRON robot motion before changing.

Additional Information

When the operating mode of the OMRON robot is changed to either Auto mode or Manual mode, the OMRON robot high power turns OFF.

You can monitor the operating mode of the OMRON robot with robot variables and V+ keywords in the sequence control program and the V+ program.

For example, while the Robot Integrated CPU Unit is in RUN mode, the operating mode of the OMRON robot is changed to Manual mode. In this case, you can monitor the operating mode of the OMRON robot in the sequence control program to stop the sequence control program.

Refer to *2-4 System-defined Variables for Robot Control* on page 2-8 for details on how to use robot variables.
Refer to the *eV+3 User's Manual (Cat. No. I651)* for details on how to use the V+ keyword.

● Changing to Manual Mode

In a system configuration where the Robot Integrated CPU Unit controls more than one OMRON robot, even if one OMRON robot is changed to Manual mode, the operating mode of the other OMRON robots are not changed.
To stop the motion of the other OMRON robots or motion control axes when one OMRON robot is changed to Manual mode, you need to program it in the user program.

Whether each OMRON robot is in Auto mode or Manual mode can be accessed with a robot variable in the sequence control program or with a V+ keyword in the V+ program.

Refer to *Section 3 Robot Control Parameters* on page 3-1 for details on how to use robot variables.
Refer to the *eV+3 User's Manual (Cat. No. I651)* for details on how to use the V+ keyword.

4-5-4 Operation of Events

Events that occurred during execution of robot control instructions from the sequence control program are either minor fault events or observation events of which the source is an OMRON robot.
Events that occurred during execution of the V+ program are information events of which the source is a robot control common error.
Refer to *Section 11 Troubleshooting* on page 11-1 for details.

In addition, these events are recorded as “information” in the event log when you change the operating mode of the OMRON robot.

5

Robot Control Function

This section describes the functionality to control the OMRON robots from the Robot Integrated CPU Unit.

5-1 Robot Control Common Function

The Robot Integrated CPU Unit controls the OMRON robots with the sequence control program and the V+ program.
The common functions to control the OMRON robots provides the state monitoring and state transition for the OMRON robot, motion operation, motion modifier that specifies parameters for motion operation, and other functions.

A list of the robot control function that can be used in the sequence control program and the V+ program is given below.

Item			Description	Sequence control program	V+ program
Robot state transition	High power	ON/OFF	Turns ON or OFF the power supply to the OMRON robot's motor.	○	○
		Auto power off	Disables high power for the OMRON robot automatically if an error occurred by a specific motion.	---	○
	Calibration	Calibration status	Enables to confirm that the calibration of the OMRON robot is completed.	○	○
		Execute calibration	Initializes the the OMRON robot and makes the robot controllable from a program.	○	○
	Enable/Disable	Robot enabled/disabled	Enables or disables to send a motion instruction to the OMRON robot.	---	○
	Robot system status	Robot status	Reads and changes the state of the OMRON robot.	○	○
		Information related to motion	Reads information for OMRON robot motion.	○	○
		Manual mode	Reads the state when the OMRON robot is manually moved by a teaching pendant.	○	○
		Hardware status	Reads the state of input circuits for the hardware of the OMRON robot.	---	○
		Front panel switch setting	Reads whether the front panel switch is in Manual mode or Auto mode.	---	○
Motion operation	Basic operation	Joint interpolation operation	Moves each joint of the OMRON robot to the specified position at a constant velocity.	○	○
		Linear interpolation operation	Moves TCP to follow a straight line to the specified position.	○	○
		Circular interpolation operation	Moves TCP to follow a circle or circular arc to the specified position.	---	○
		Jog operation	Operates the specified joint of the OMRON robot or moves TCP along the direction to cartesian coordinates.	---	○

Item			Description	Sequence control program	V+ program
	Tool coordinate system operation	Rise or lower	Rises or lowers TPC along the Z axis of the tool coordinate system.	---	○
		Align tool	Operates the OMRON robot to align the Z axis of the robot tool to the nearest axis of the world coordinate system.	---	○
	Joint operation	Individual joint operation	Operates the specified joint of the OMRON robot.	---	○
	Continuous-path motion	ON/OFF	Specifies the continuous operation for the OMRON robot to transfer smoothly or wait until the target position is reached.	○	○
	Deceleration stop	Stop the current operation	Stops the current operation of the OMRON robot to cause a deceleration stop.	○	○
	Standard position	Move to standard position	Moves to the standard position that is defined by each model of the OMRON robot.	---	○
Motion modifier	Robot velocity	Velocity profile	Selects the velocity profile that is defined in the Sysmac Studio. The velocity profile consists of the jerk up or down range of acceleration or deceleration.	○	○
		Operation velocity	Specifies the velocity of the robot motion.	○	○
		Acceleration	Specifies the acceleration of the robot motion.	○	○
		Deceleration	Specifies the deceleration of the robot motion.	○	○
		Shortest operation time	Specifies the shortest time of a operation for the OMRON robot.	---	○
		Unit of velocity	A percentage of the maximum velocity	○	○
			mm/s	○	○
			inch/s	---	○
	Arm configuration	ABOVE/BELOW, LEFTY/RIGHTY, FLIP/NOFLIP	Specifies the posture of OMRON robot if the OMRON robot has reached the target position.	○	○
	Hardware servo	High accuracy/low accuracy	Selects the checking methods for reaching to the target position: prioritize a high accuracy or reduce the operation time as a low accuracy	○	○
	Rotational axis	Rotation range limitation	Restricts the rotational range of the wrist axis for the OMRON robot.	○	○
		Rotation range exceeded error	Detects an error if the instruction that exceeds the rotational range of the wrist axis for the OMRON robot is received.	○	○
	Following error	Wait for cancellation of following error	Specifies whether to wait for the cancellation of the following error at the end of the OMRON robot motion.	○	○

Item			Description	Sequence control program	V+ program
Other functions	Coordinate system	World coordinate system	Changes the world coordinate system for the OMRON robot.	○	○
		Tool coordinate system	Sets the tool offset of the OMRON robot and changes the tool coordinate system.	○	○
		Conversion from/to NJ Robotics function coordinate system	Converts the coordinate system used by the RC Function Module into the coordinate system for the NJ Robotics function, or vice versa.	○	---
	Position variables	Conversion	Specifies the target position or other position with the position and direction in absolute coordinates space.	○	○
		Relative conversion	Specifies a combination of relative conversion for a conversion.	---	○
		High-accuracy positioning point	Specifies the target position or other position with the position for each joint of the OMRON robot.	---	○
	Stop	Specified time stop	Stops the OMRON robot motion for the specified time.	---	○
	Robot tool	Tool offset setting	Specifies the offset value for TCP of the OMRON robot to the tool tip as a relative position to the world coordinate system.	○	○
	Conveyor tracking	Belt variables	Data types to handle a belt conveyor.	---	○
		Nominal transformation	Defines the position, direction, and direction of operation for the belt conveyor.	---	○
		Encoder scaling factor	Converts the encoder count that is mounted to the belt conveyor into millimeters.	---	○
		Encoder offset	Defines the reference position for the encoder that is mounted to the belt conveyor.	---	○
		Belt window	Restricts the area to move the OMRON robot on the belt conveyor.	---	○
		Belt relative operation	Performs tracking the OMRON robot to the target position that is relatively specified for the belt conveyor.	---	○
	Latching	Robot position latching	Obtains the position of the OMRON robot when a latch input signal occurs.	---	○
		Local encoder latching	Obtains the value of encoder connected to the encoder input port of the OMRON robot when a latch input signal occurs.	---	○
		Remote encoder latching	Obtains the value of encoder which set a motion control axis as an external encoder for robot control when a latch input signal occurs.	---	○

5-2 Tracking Control

The Robot Integrated CPU Unit provides the tracking control to follow the relative movement distance of the slave machine for the relative movement distance of the master machine.
The relative movement distance is a differential distance between the position in the previous control period and the position in the current control period.

The following table shows the combination of master machine and slave machines that can specified in the Robot Integrated CPU Unit.

Master machine	Slave machines	Applicable program
Local encoder	OMRON robots	V+ program
Motion control axis	OMRON robots	V+ program

The encoder that is used as the master machine is specified with DEFBELT V+ keyword in the V+ program.
Refer to the *eV+3 User's Manual (Cat. No. I651)* and the *eV+3 Keyword Reference Manual (Cat. No. I652)* for details on the V+ programs.

If the tracking control is used, the master machine is defined for each slave machine (OMRON robot). When the tracking control starts, the slave machine operates to follow the defined master machine.

Precautions for Safe Use

- If the OMRON robot is operated with tracking control, set the parameters for the master machine so that the velocity of the master machine does not change rapidly.
- When the current position for a motion control axis is changed with the MC_Home (Home), MC_HomeWithParameter (Home with Parameters), or MC_SetPosition (Set Position) instruction or the **Axis Use** axis parameter setting is changed with the MC_ChangeAxisUse (Change Axis Use) instruction during tracking control, the tracking operation may change rapidly.
- When the EtherCAT slave assigned to a motion control axis is disconnected or connected, or enabled or disabled during tracking control, the tracking operation may change rapidly.

5-3 Latching

The Robot Integrated CPU Unit provides the functionality to latch the positions of OMRON robots and encoder counter values that assume to use a belt conveyor.
The latch function saves the position when a latch signal such as a sensor input occurred and reads the position from the V+ program.

The following table shows the object for latching, latch signal, applicable program, and the maximum number.

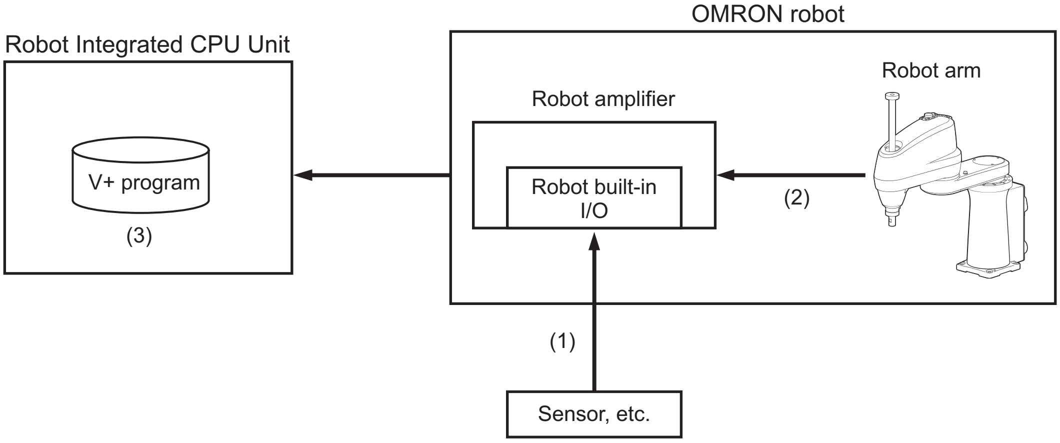
Object for latching	Maximum number of applicable devices	Latch signal	Maximum number of latch signal	Applicable program
Robot position	---	Robot built-in I/O	8 signals per robot<K04>A	V+ program
Local encoder	2 devices per robot	Robot built-in I/O	8 signals per encoder	V+ program
Remote encoder*1	16 devices*2	I/O for latch input of the device	8 signals per encoder	V+ program

*1. The remote encoder uses an EtherCAT servo axis or encoder axis that is assigned to an "axis" for the motion control function. Refer to the *NJ/NX-series CPU Unit Motion Control User’s Manual (Cat. No. W507)* for details.

*2. The number includes the virtual servo axes and virtual encoder axes.

5-3-1 Robot Position Latching

The robot position latching reads the robot position from the V+ program when a latch signal occurred.
The robot built-in I/O must be used for the latch signal.
The robot position latching can be used with LATCH V+ keyword in the V+ program.



1. Input the latch signal.
2. Latch the robot position.
3. Enable to read the latch data with the V+ program.

An example of a V+ program is provided below.

```
;Robot position
SigNo = LATCHED(robot)
IF SigNo<>0 THEN
```

```
robot.pos = LATCH(robot) ;Read the latch data of the robot position
END
```

Refer to the *Sysmac Studio Robot Integrated System Building Function with Robot Integrated CPU Unit Operation Manual (Cat. No. W595)* for information on the settings of the robot position latching.

5-3-2 Local Encoder Latching

The local encoder latching reads the local encoder value from the V+ program when a latch signal is input.
The robot built-in I/O must be used for the latch signal.

The local encoder latching can be used with DEVICE V+ keyword in the V+ program.

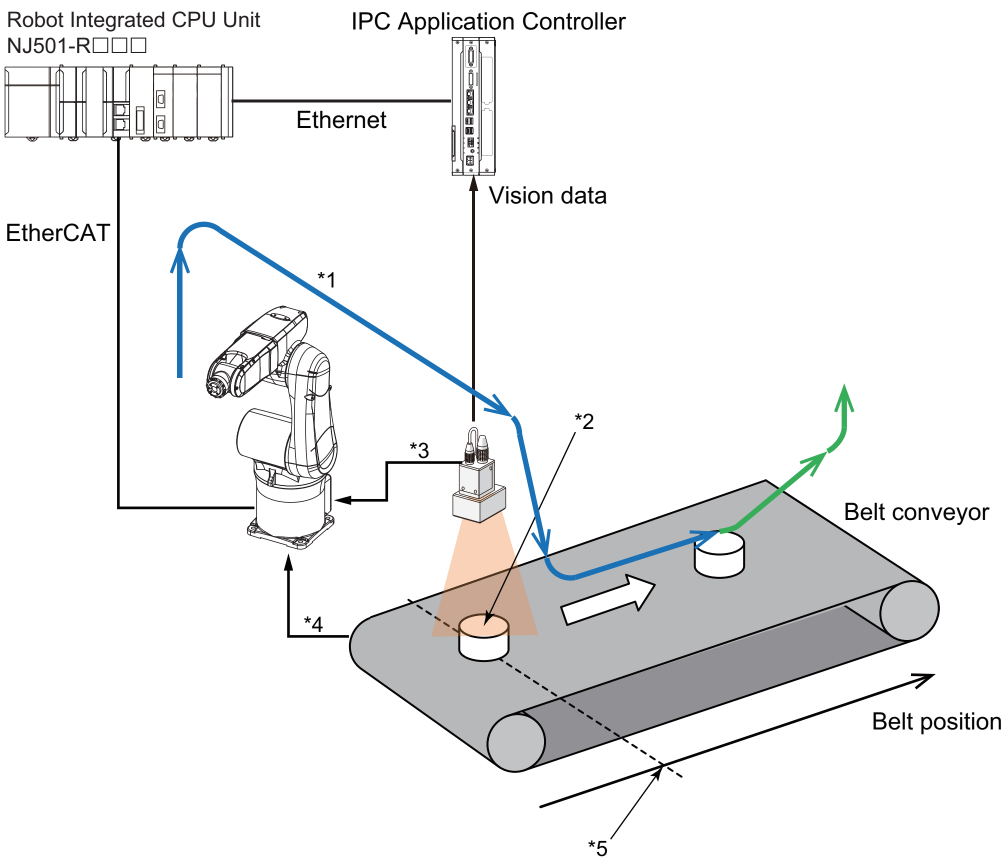
The configuration to use the local encoder latching is given below.

Encoder device	Digital I/O
Encoder connected to the encoder input port on the OMRON robot	Robot built-in I/O

The latched position that is obtained with this function is used as a belt reference position in case of the conveyor tracking.

The configuration example for conveyor tracking is given below.

- The latch signal is connected to an input of the robot built-in I/O.
- Connect the encoder that is mounted to the belt conveyor to the encoder input of the OMRON robot.
- Use the Sysmac Studio to make the settings for linking the latch signal and the encoder that is mounted to the belt conveyor.
- Assign the latch signal number of the robot built-in I/O that is used for the latch signal to the encoder that is mounted to the belt conveyor.



*1. Robot path (conveyor tracking)
*2. Workpiece position detected by the image processing
*3. Input a shutter signal as a latch signal
*4. Pulse of encoder mounted to the belt conveyor
*5. Latched belt position

Refer to the *Sysmac Studio Robot Integrated System Building Function with Robot Integrated CPU Unit Operation Manual (Cat. No. W595)* for information on the settings of the local encoder latching.

An example of a V+ program is provided below.

```
DEFBELT %belt = nom_trans, belt_num, … ;Belt definition
…
WHILE SigNo <> 0 DO ; Wait until it latches
SigNo = LATCHED(-belt_num)
END
b = DEVICE(0, belt_num-1, , 4) ;Obtain the latched belt  position
SETBELT %belt = b ;Set the latched belt  position
...
SET pick.loc = VLOCATION($ip, sequence, tool, instance, result, index, frame) ; Obt
ain the results of vision sensor
…
trans := %belt: pick.loc ;Multiple conversion (workpiece position calculated by a v
ision system)
MOVES trans ;Move to the pick position (relative movement for the belt)
;Application processing
```

5-3-3 Remote Encoder Latching

The remote encoder latching uses the digital signal assigned to I/O in the V+ program as shown in *3-2-2 V+Digital I/O Settings* on page 3-4 as the latch signal and reads the position information of the motion control axis from the V+ program when the latch signal occurs.

The remote encoder latching can use the latch function with V+ keywords by specifying an encoder ID in the same way for using a local encoder.
To use the remote encoder latching, set a motion control axis as an external encoder for robot control.

Refer to *3-2-3 Remote Encoder Latch Settings* on page 3-6 for information on the settings of the remote encoder latching.

Precautions for Safe Use

- When the current position for a motion control axis is changed with the MC_Home (Home), MC_HomeWithParameter (Home with Parameters), or MC_SetPosition (Set Position) instruction or the **Axis Use** axis parameter setting is changed with the MC_ChangeAxisUse (Change Axis Use) instruction during tracking control, the tracking operation may change rapidly.
- When the EtherCAT slave assigned to a motion control axis is disconnected or connected, or enabled or disabled during tracking control, the tracking operation may change rapidly.
- Refer to the *NJ/NX-series Motion Control Instructions Reference Manual (Cat. No. W508)* for details on the instructions.

Additional Information

- When time stamps are assigned to both of the followings, more precise latch position compensation can be performed.
  a) Process data for the device that is assigned to a motion control axis for which **Position Type Selection** is set to **Actual position**.
  b) Process data for the device that contains signals set in *3-2-2 V+Digital I/O Settings* on page 3-4.

  Refer to the user's manual for each device for information on the assignment of time stamp to process data for each device.
- If any of the following conditions is met for the motion control axis assigned to the encoder ID as a remote encoder, the position information is not latched. We recommend that **Count Mode** for a motion control axis is set to **Rotary Mode**.
  a) **Axis Use** is not set to **Used axis**.
  b) Process data communications are not established with the device assigned to **Input Device**.
  c) The latch position is overflowed or underflowed.
  d) As a result of latch position compensation by a time stamp, latched position is overflowed or underflowed.

5-4 Cyclic Path Compensation

This section describes the cyclic path compensation.

The Robot Integrated CPU Unit provides the functionality to compensate the path with ALTER V+ keyword for the OMRON robot that is controlled with a V+ program.
The cyclic path compensation realizes the compensation with higher accuracy than with the ALTER V + keyword.

The cyclic path compensation enables realtime OMRON robot path compensation by using the feedback information that is calculated based on the information collected from each type of sensor such as a pressure sensor, force sensor and vision device, etc. which is connected to the Robot Integrated CPU Unit.
Therefore, the path control that is appropriate for the environment in which the OMRON robot is used is enabled.

Overview of Cyclic Path Compensation

For the OMRON robot that is executing the operation specified with the V+ program which will be used as the reference, you can give offset values for the target position to enable compensating tracking control.
Or, for the OMRON robot that does not have the operation which will be used as the reference, you can control the robot path by directly giving the target position for the robot based on the path determined from any information.

Unlike the ALTER V+ keyword in the V+ program, the RC_SyncPathOffset (Robot Path Compensation) instruction in the robot control instructions can periodically give the offset values or target position to the OMRON robot.
Note that the offset values or target position need to be calculated with an user-defined algorithm.

Refer to *RC_SyncPathOffset* on page 9-33 for details on the RC_SyncPathOffset (Robot Path Compensation) instruction.

Precautions for Correct Use

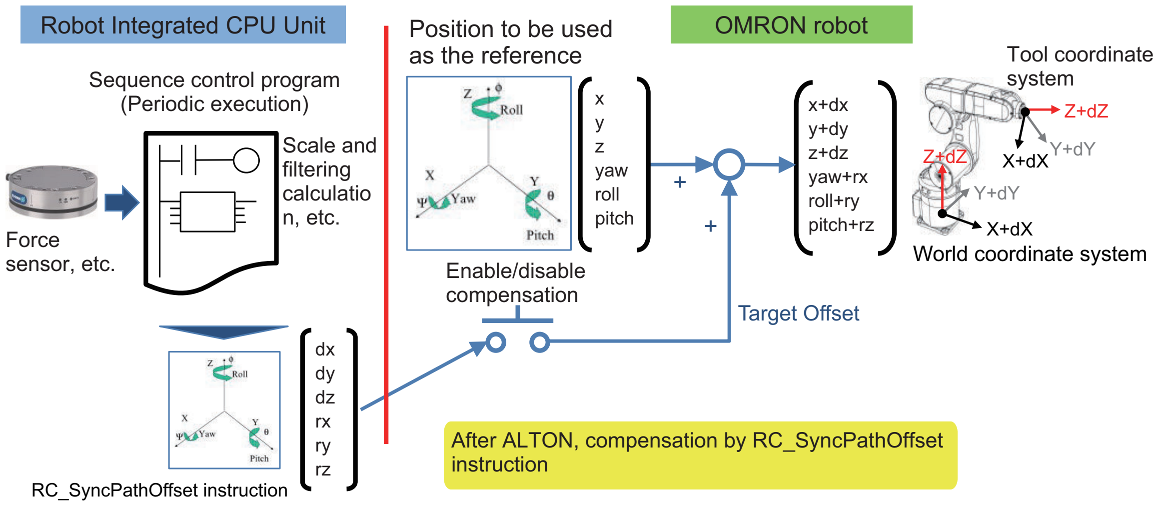
You cannot use this function for the robot that is executing the conveyor tracking of the V+ program.

Example of Compensation Control with Cyclic Path Compensation

You can compensate the robot path by giving the user-defined compensation values that are calculated based on the external input information from a pressure sensor or force sensor to the OMRON robot that is executing the operation which is used as the reference.

An example is shown below.
The operation of the OMRON robot which will be used as the reference is realized with the V+ program.

You must define the coordinate system with ALTON V+ keyword in the V+ program in advance to accept the compensation values.
The RC_SyncPathOffset (Robot Path Compensation) instruction is executed periodically to send compensation values that are calculated based on the information from the force sensor to the OMRON robot and it enables highly accurate compensation processing.
You can control to enable/disable the compensation processing with execution/stop of the RC_SyncPathOffset (Robot Path Compensation) instruction.



Refer to *RC_SyncPathOffset* on page 9-33 for information on the RC_SyncPathOffset (Robot Path Compensation) instruction.
Refer to the *eV+3 Keyword Reference Manual (Cat. No. I652)* for information on the ALTON V+ keyword.

Example of Path Control with Cyclic Path Compensation

For the OMRON robot that does not have the operation which will be used as the reference in advance, you can give the target position of the robot directly based on the path determined from any input information such as a vision sensor, etc.

An example is shown below.
You must define the coordinate system with the ALTON V+ keyword in the V+ program in advance to accept the values of target position.
Then, the path of the robot to move is calculated based on the information from the vision sensor, etc. with the RC_SyncPathOffset (Robot Path Compensation) instruction in the sequence control program that is executed periodically, and it is sent to the OMRON robot to realize highly accurate path control.

5-5 Coordinate System Integration with NJ Robotics Function

This function converts the coordinates used by the RC Function Module into the coordinate system specified for the NJ Robotics function, or vice versa.
You can control the OMRON robots controlled in the RC Function Module and the robots controlled by the NJ Robotics function in the same coordinate system with this function.
The conversion can be performed with the RC_ConvertCoordSystem (Convert Coordinate System) instruction.

Robot Control Function Module coordinate system (Yaw-Pitch-Roll system)

Index	Description
0	X value
1	Y value
2	Z value
3	Yaw angle
4	Pitch angle
5	Roll angle

_rcRCToNJRobotics →

← _rcNJRoboticsToRC

NJ Robotics function coordinate system (Roll-Pitch-Yaw system)

Index	Description
0	X value
1	Y value
2	Z value
3	Roll angle
4	Pitch angle
5	Yaw angle

An example of using the coordinates created for the program in the Robot Control Function Module to the program for NJ Robotics is given below.

ST Language Program for Robot Control Function Module

```
Pos_YPR[0] := pos_x;
Pos_YPR[1] := pos_y;
Pos_YPR[2] := pos_z;
Pos_YPR[3] := pos_yaw;
Pos_YPR[4] := pos_pitch;
Pos_YPR[5] := pos_roll;
Move_Exec := trigger;
RC_MoveLinear_instance(
Robot := _RC_RBT[0],
Execute := Move_Exec,
Position := Pos_YPR,
Done => Move_Done
);
```

ST Language Program for NJ Robotics

```
Pos_YPR[0] := pos_x;
Pos_YPR[1] := pos_y;
Pos_YPR[2] := pos_z;
Pos_YPR[3] := pos_yaw;
Pos_YPR[4] := pos_pitch;
Pos_YPR[5] := pos_roll;
//Convert the Robot Control Function Module coordinate system into the NJ Robotics
function coordinate system
```

```
RC_ConvertCoordSystem(_eRC_CONVERT_FORM#_rcRCToNJRobotics, Pos_YPR, Pos_RPY);
Move_Exec := trigger;
MC_MoveTimeAbsolute_instance(
AxesGroup := _MC_GRP [0],
Execute := Move_Exec,
Position := Pos_RPY, //Set the position after conversion as a target position
Done => Move_Done
);
```

Refer to *RC_ConvertCoordSystem* on page 8-11 for details.

Precautions for Safe Use

The coordinate system used by the Robot Control Function Module have different specifications from the coordinate system used by the NJ Robotics function.
If you use both functions simultaneously, use the RC_ConvertCoordSystem (Convert Coordinate System) instruction to set the same coordinate system before performing the robot control.

5-6 Changing Recipe

You can request a recipe change from the Robot Integrated CPU Unit to the IPC Application Controller.
The recipe change uses the Recipe Manager function and Robot Vision Manager function in the IPC Application Controller.

Refer to the *Sysmac Studio Robot Integrated System Building Function with IPC Application Controller Operation Manual (Cat. No. W621)* for information on the software that is installed in the IPC Application Controller.

Refer to the *Sysmac Studio Robot Integrated System Building Function with IPC Application Controller Operation Manual (Cat. No. W621)* for information on creating and monitoring a recipe.

Recipe Change from V+ program

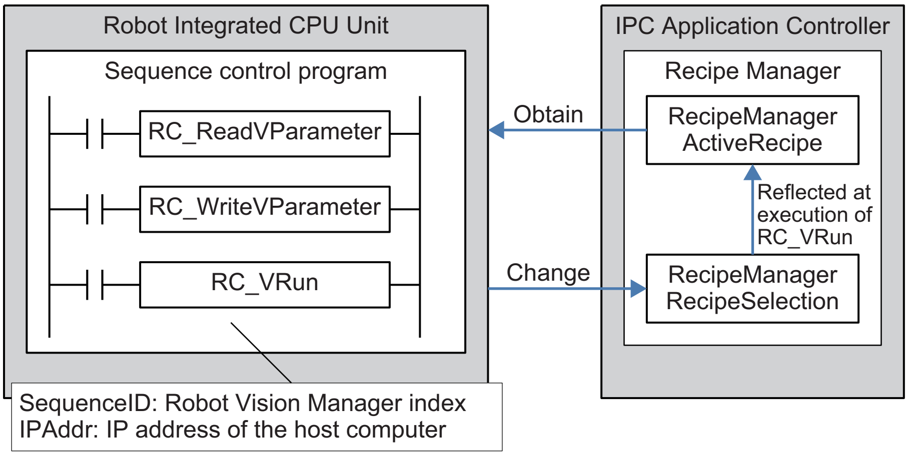
You can change or obtain the recipe number that is currently selected using VPARAMETER V+ keyword from the V+ program.
To obtain a recipe number, get the **RecipeManagerActiveRecipe** (Parameter ID 8001).
To change a recipe number, execute the VRUN V+ keyword after the setting of **RecipeManagerRecipeSelection** (Parameter ID 8002).

Refer to the **ACE Reference Guide** help file in the Sysmac Studio for details.

Recipe Change from Sequence Control Program

You can obtain or change the recipe number that is currently selected using the RC_ReadVParameter (Read Vision Parameters) instruction or the RC_WriteVParameter (Write Vision Parameters) instruction from the sequence control program.
To obtain a recipe number, get the **RecipeManagerActiveRecipe** (Parameter ID 8001).
To change a recipe number, execute the RC_VRun instruction of the function block after the setting of **RecipeManagerRecipeSelection** (Parameter ID 8002).
Use the RC_ReadVParameter (Read Vision Parameters) instruction to obtain the recipe number in order to check the completion of recipe reflection by the RC_VRun (Run Vision Sequence) instruction.
Refer to the **ACE Reference Guide** help file in the Sysmac Studio for details.

An example is shown below.



If you change a recipe from the HMI, design the screen for HMI and a sequence control program so that the recipe number is changed in the sequence control program on the screen for HMI. Refer to the user's manual for HMI for details.

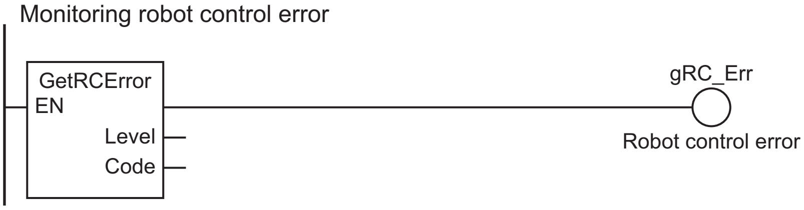
● Sample Programming

Internal variables	Variable	Data type	Initial value	Comment
	Retry	BOOL		Retry to obtain
	RC_RVP	RC_ReadVParameter		
	IPAddr	STRING[256]	'127.0.0.1'	IP Address
	RC_VR	RC_VRun		
	RC_WVP	RC_WriteVParameter		
	WriteOK	BOOL		Changing completed
	ReadOK	BOOL		Obtaining completed
	UpdateOK	BOOL		Reflection completed

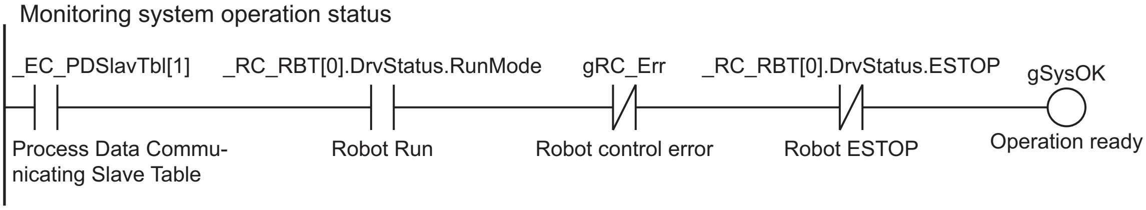
External variables	Variable	Data type	Constant	Comment
	_RC_RBT	ARRAY[0..7] OF _sRC_RBT_REF	☑	
	_EC_PDSlavTbl	ARRAY[1..192] OF BOOL	☑	
	gRC_Err	BOOL	☐	Robot control error
	gSysOK	BOOL	☐	Operation ready
	gChangeRecipeNo	LREAL	☐	Changed recipe number
	gCurrentRecipeNo	LREAL	☐	Current recipe number
	gSequenceID	DINT	☐	Sequence ID
	gStartChangeRecipe	BOOL	☐	Start recipe change

External variables	Variable	Data type	Constant	Comment
	gCompleteChangeRecipe	BOOL	☐	Recipe change completed

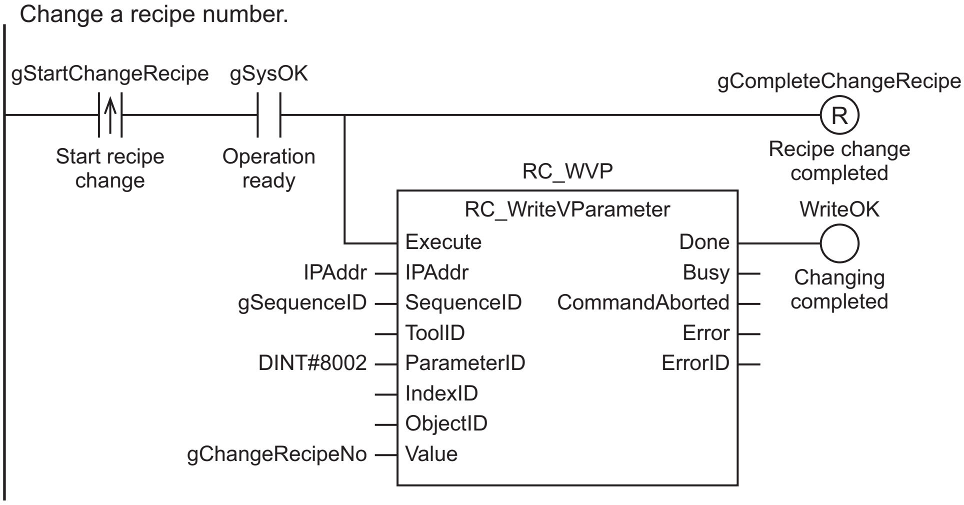
Monitoring robot control error



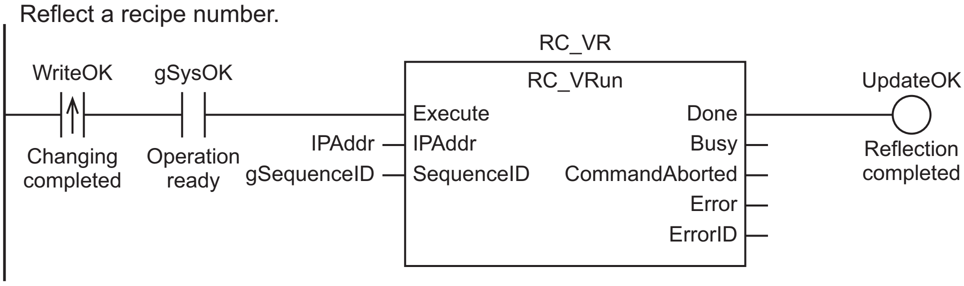
Monitoring system operation status



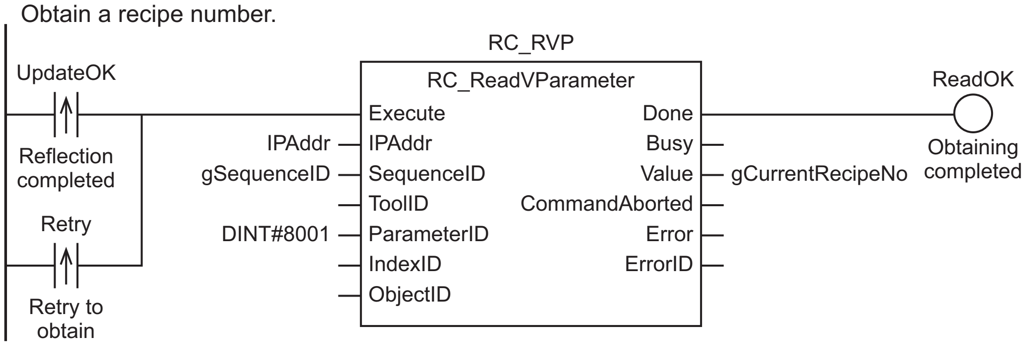
Change a recipe number.



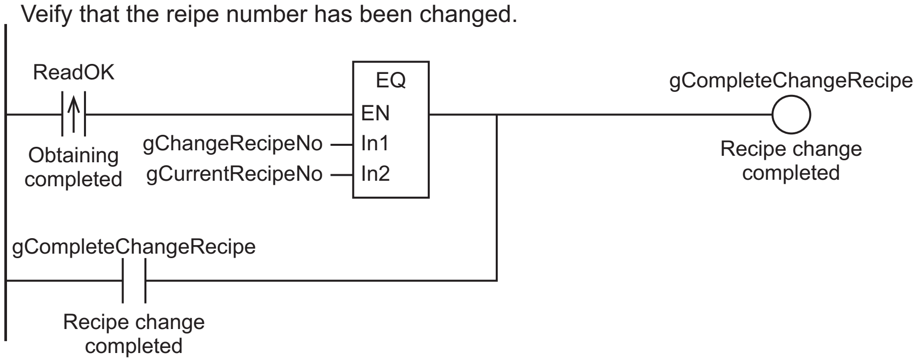
Reflect a recipe number.



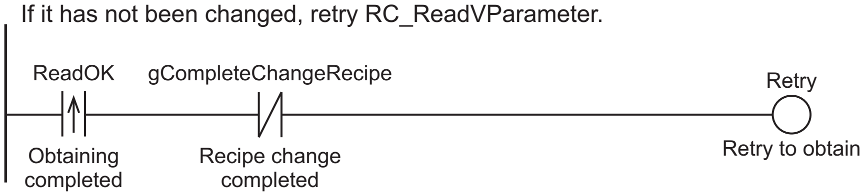
Obtain a recipe number.



Veify that the reipe number has been changed.



If it has not been changed, retry RC_ReadVParameter.



6

Robot Control Instructions

This section describes the overview of robot control instructions and basic understanding of the instructions.

6

6-1 Overview of Robot Control Instructions

This section describes the overview of robot control instructions.
The robot control instruction is an instruction of the sequence control program to control the OMRON robots with the Robot Integrated CPU Unit.
Robot control instructions are used to control an OMRON robot from the sequence control program with the Robot Integrated CPU Unit. These instructions are defined as function blocks.
The instructions include instructions that directly control the OMRON robots and instructions to execute or abort V+ programs assigned to the V+ tasks.
Any type of OMRON robots such as parallel robots, SCARA robots, and articulated robots can directly perform robot control with the same instructions and programming methods.

6-1-1 Types of Robot Control Instructions

The following table shows the different types of robot control instructions.

Type	Description	Reference
Common commands	Common instructions for the RC Function Module.	page 8-1
Robot commands	Instructions for the RC Function Module to perform robot control.	page 9-1

The instructions for the RC Function Module include robot control instructions and system control instructions. Refer to *Section 10 System Control Instructions* on page 10-1 for information on the system control instructions in the RC Function Module.

6-1-2 Execution and Status of Robot Control Instructions

Variables that start instruction execution or indicate the execution status of an instruction are defined as common rules for the instructions.
There are two input variables that start instruction execution: *Execute* and *Enable*.
The output variables that indicate the execution status of an instruction include *Busy*, *Done*, *CommandAborted*, and *Error*.

Refer to the *NJ/NX-series CPU Unit Motion Control User’s Manual (Cat. No. W507)* for details on the above input variables and output variables.

6-1-3 Error Processing

When robot control instructions in the RC Function Module are executed, input parameters and instruction processing are checked for errors.
If an error occurs in an instruction, the *Error* output variable from the instruction changes to TRUE and an event code for the error is output to *ErrorID* output variable.
The upper four digits of the event code give the error code for *ErrorID*.

If there is no error in an instruction and the processing is aborted due to external factors, *Command Aborted* output variable in the instruction changes to TRUE.

Error Processing for Individual Instruction

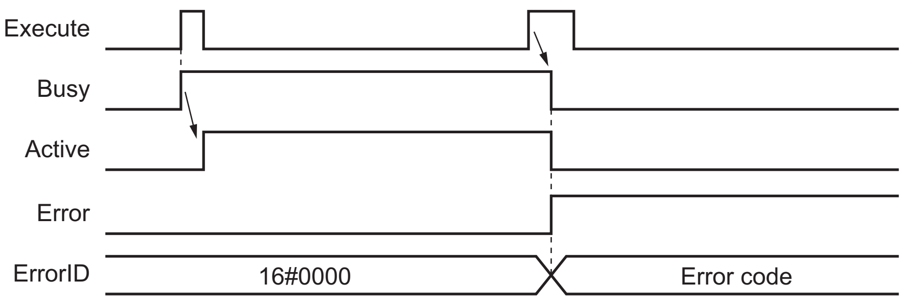
You can use the *Error* and *ErrorID* output variables from the instruction to process errors that occur for each instruction.

Error Processing for Different Types of Errors

You can use the error status that is provided by the system-defined variables for robot control to process each type of error separately.

6-1-4 Changing Input Variables during Execution of Robot Control Instructions (Instruction Re-execution)

If the values of the input variables to the same instance are changed while the robot control instruction is under execution and *Execute* is changed to TRUE again, the instruction ended in an error.



6-1-5 Multi-execution of Instructions with BufferMode

Another robot control instruction instance can be executed for the OMRON robot during motion with the robot control instruction.
You can use multi-execution of instructions to execute each robot control instruction in sequence without waiting for the motion completion.
You can specify when the instruction instance that is executed last starts by setting an input variable called *BufferMode*.
The number of multi-execution of instructions that is buffered is up to eight.

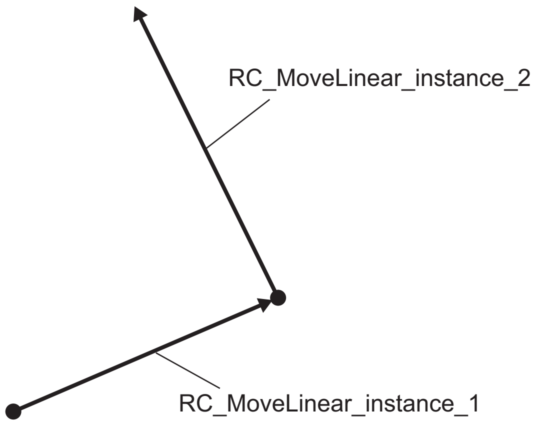
The following Buffer Modes are supported for *BufferMode*.
- Continuous-path motion OFF: _rcBuffered
- Continuous-path motion ON: _rcBlending

BufferMode is Set to Continuous-path Motion OFF

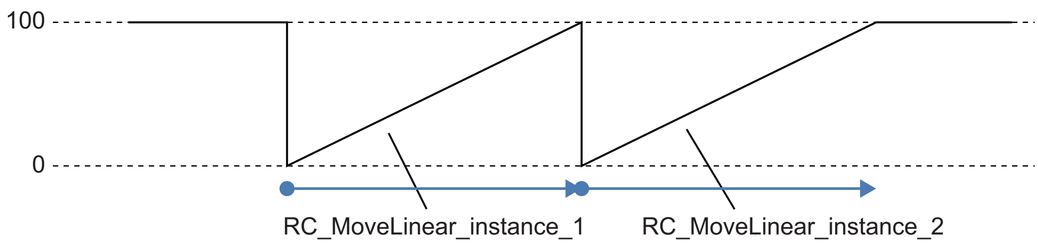
If *BufferMode* for the instruction instance executed second is set to Continuous-path motion OFF, the continuous-path motion is not performed.
After the OMRON robot reached the target position of the first instruction, the robot starts motion for the target position of the second instruction.

The OMRON robot reaches the given target position accurately. However, the robot stops once, so the total operation time becomes longer.

The following figure shows the path with *BufferMode* (Buffer Mode Selection) set to Continuous-path motion OFF.

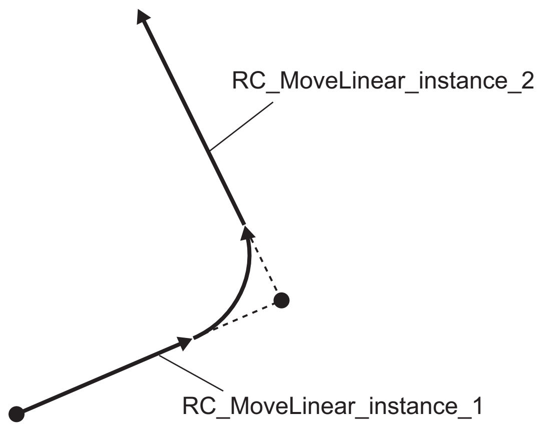


The following figure shows the change of *MotionCompleted* (Robot Motion Progress) robot variable in this motion.



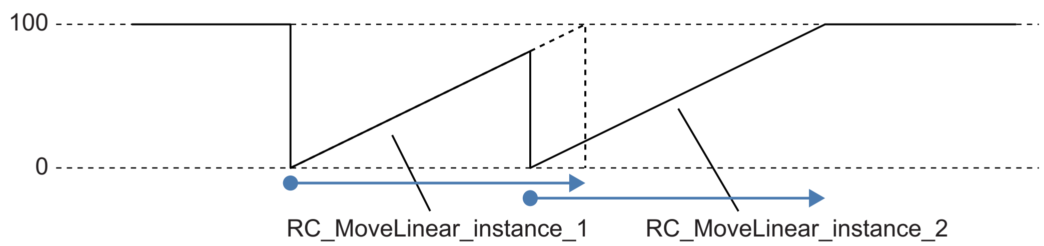
BufferMode is Set to Continuous-path Motion ON

If *BufferMode* for the instruction instance executed second is set to Continuous-path motion ON, the continuous-path motion is performed.
Before the OMRON robot reaches the target position of the first instruction, the robot starts motion for the second instruction. These motions are combined.
The OMRON robot moves smoothly without stopping, and the total operation time is reduced. However, the target position of the first instruction may not be reached.
The following figure shows the path with *BufferMode* (Buffer Mode Selection) set to Continuous-path motion ON.



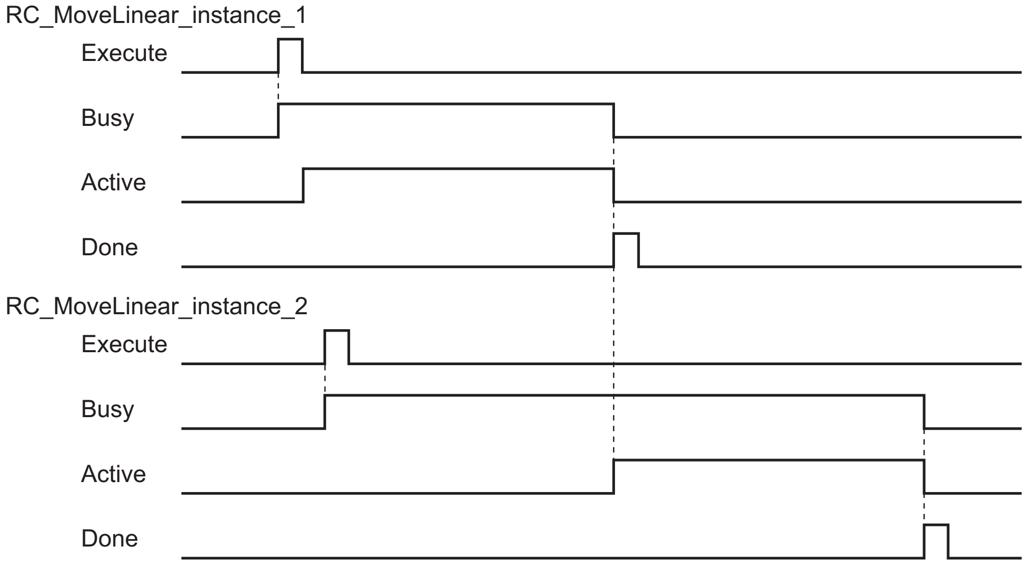
If _rcBlending (Continuous-path motion ON) is specified, you cannot specify the size of curved line that changes a direction smoothly with parameters.
Continuous-path motion controls the OMRON robot motion from the settings of velocity, acceleration, and deceleration so that the robot performs the optimum motion.

The following figure shows the change of *MotionCompleted* (Robot Motion Progress) robot variable in this motion.
When the motion of the second instruction starts, *MotionCompleted* (Robot Motion Progress) is reset to 0.



Timing Charts for Multi-execution of Instructions

During multi-execution of instructions, *Active* for the second instruction changes to TRUE when *Done* for the first instruction changes to TRUE.
The same operation is performed regardless of the selection of *BufferMode* (Buffer Mode Selection).


RC_MoveLinear_instance_1
Execute
Busy
Active
Done
RC_MoveLinear_instance_2
Execute
Busy
Active
Done

6-2 Basic Understanding of Robot Control Instructions

This section describes basic specifications and restrictions for programming with robot control instructions.
Refer to *Section 8 Common Command Instructions* on page 8-1, *Section 9 Robot Command Instructions* on page 9-1, and *Section 10 System Control Instructions* on page 10-1 for details on the robot control instructions.

6-2-1 Names of Robot Control Instructions

All robot control instructions for the Robot Control Function Module begin with "RC_".

6-2-2 Languages of Robot Control Instructions

Robot control instructions of the RC Function Module can be used in the following programming languages.

- Ladder Diagram Language
- Structured Text Language

6-2-3 Locations of Robot Control Instructions

This section describes the difference in operation depending on the tasks assigned from robot control instructions and locations of robot control instructions.

Task Types

The robot control instructions can be used for the primary periodic task. If robot control instructions are used in any other tasks, an error will occur when the user program is built on the Sysmac Studio.

Task Types	Applicability
Primary periodic task	Applicable
Priority-16 periodic task	Not Applicable
Priority-17 periodic task	Not Applicable
Priority-18 periodic task	Not Applicable
Priority-8 event task	Not Applicable
Priority-48 event task	Not Applicable

Additional Information

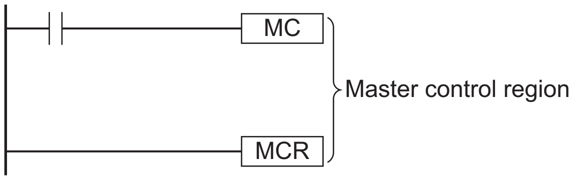
System control instructions can be used in the primary periodic task, or in a priority-16, 17, or 18 periodic task.

Function Block Definitions

You can also use robot control instructions in user-defined function block definitions.

Master Control Regions

The area in a ladder diagram between the Master Control Start instruction (MC) and the Master Control End instruction (MCR) is the master control region.



If a robot control instruction is located in the master control region, when the MC input condition is FALSE, the following will occur.

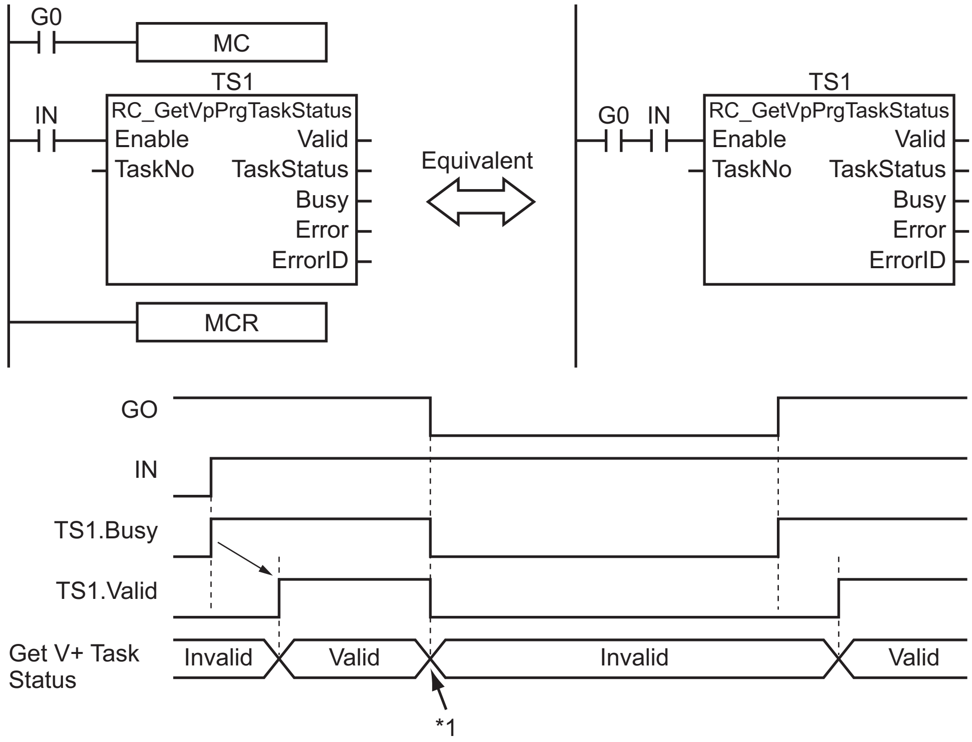
- Robot control instructions for which an input variable, *Enable* or *Execute*, is connected directly to the left bus bar are executed with a FALSE value for the input value.
- The values of the output parameters are updated as normal even when the *Enable* or *Execute* input variables to the robot control instructions are FALSE.

Precautions for Correct Use

You cannot use master control for Structured Text Language and inline ST inside a ladder diagram.
Refer to the *NJ/NX-series CPU Unit Software User's Manual (Cat. No. W501)* for information on the master control and the *NJ/NX-series Instructions Reference Manual (Cat. No. W502)* for information on the MC and MCR instructions.

● Enable-type Robot Control Instructions

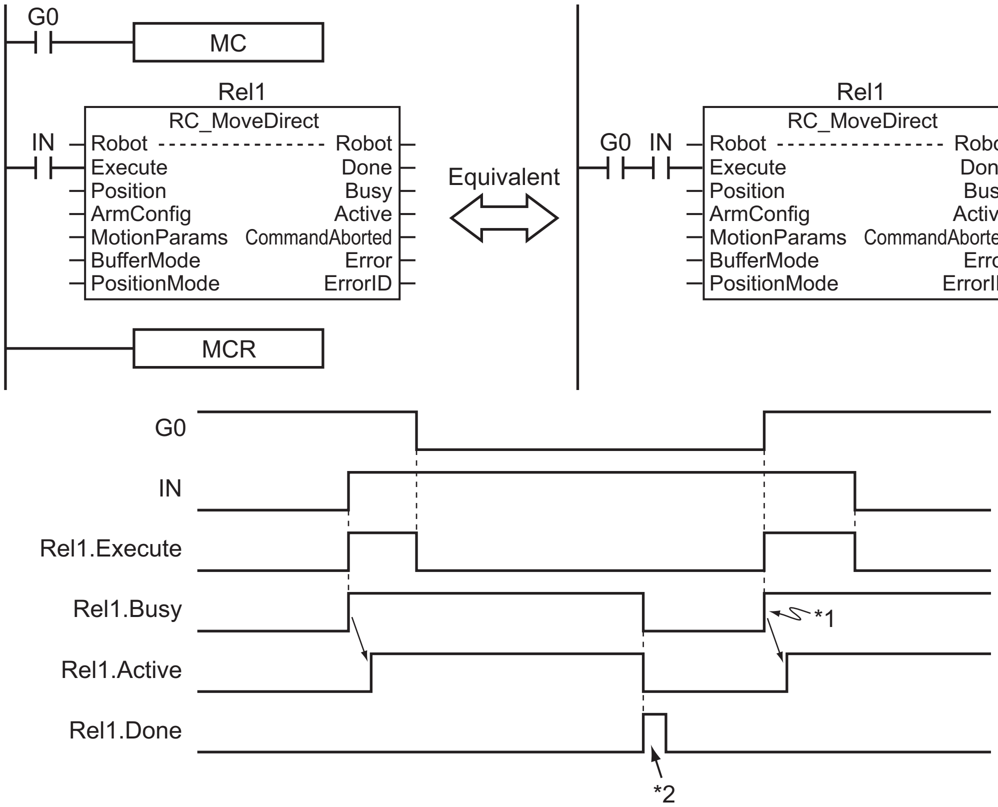
- Instructions located in master control regions are equivalent to the programming shown on the right in the following figure.
- When G0 is TRUE, RC_GetVpPrgTaskStatus is executed normally.
- When G0 is FALSE, RC_GetVpPrgTaskStatus is executed as if the *Enable* input variable was FALSE.
- Instructions executed when G0 is TRUE abort operation if G0 changes to FALSE during operation.



*1. *Enable* of the robot control instruction changes to FALSE and the data to obtain is invalid.

● Execute-type Robot Control Instructions

- Instructions located in master control regions are equivalent to the programming shown on the right in the following figure.
- When G0 is TRUE, RC_MoveDirect is executed normally.
- When G0 is FALSE, RC_MoveDirect is executed as if the *Execute* input variable was FALSE.
- Instructions executed when G0 is TRUE continue operation until completion, even if G0 changes to FALSE during operation. The values of output parameters are also updated in the normal way.



*1. Positioning starts when *Execute* changes to TRUE.

*2. Positioning is completed when *Execute* changes to FALSE, so Rel1.Done changes to TRUE for only one period.

Precautions for Correct Use

Execute-type robot control instructions are executed when G0 changes to TRUE. It is not recommended to use them in the master control region.
If they must be used, be careful of the operation.

Robot Control Instructions in ST Structure Instructions

This section describes the operation of robot control instructions when they are located in ST structures, such as IF, CASE, WHILE, or REPEAT structures.
When the evaluation result for the condition expression of an ST structure instruction is FALSE, the robot control instructions within the structure are not executed. Also, the values of the output variables are not updated.
If an execute-type instruction is executed and then the evaluation result changes to FALSE, processing is continued until it is completed. In that case, however, the values of the output variables are not updated.

Precautions for Correct Use

The execution status of an execute-type instruction in an ST structure will not be clear if the evaluation result of the condition expression changes to FALSE during execution of the instruction. We therefore do not recommend using execution-type instructions in ST structures.
If they must be used, be careful of the operation.

Additional Information

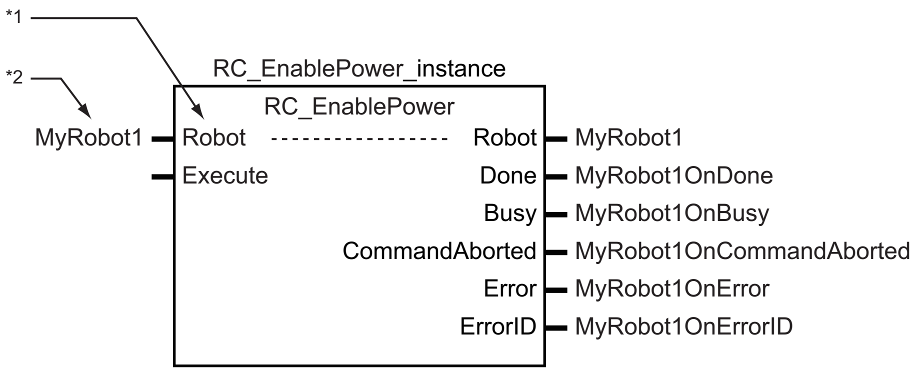
To switch the execution of an execute-type instruction with the condition expression, place only the *Execute* input parameter in the ST structure. Place the execute-type instruction itself outside of the ST structure. Refer to the *NJ/NX-series Instructions Reference Manual (Cat. No. W502)* for details on the ST structure instruction.

Treatment of REAL and LREAL Data

Refer to the *NJ/NX-series Motion Control Instructions Reference Manual (Cat. No. W508)* for details.

6-2-4 OMRON Robot Specification Method in Sequence Control Program

In the sequence control program, a robot variable name is specified for the in-out variable *Robot* in robot control instructions.
In the following example, the robot variable name for the OMRON robot that was added for the system-defined robot variable name of *_RC_RBT[0]* has been changed to *MyRobot1* in the Sysmac Studio.



*1. *Robot* in-out variable
*2. Specify with a variable name of robot variable.

You can also use the *_RC_RBT[0]* system-defined variable in place of *MyRobot1*.

Refer to *7-2 Instructions* on page 7-9 for details on the robot control instructions.

6-2-5 Multi-execution of Robot Control Instructions

This section describes executing multiple robot control instructions for the same OMRON robot within the same task period.

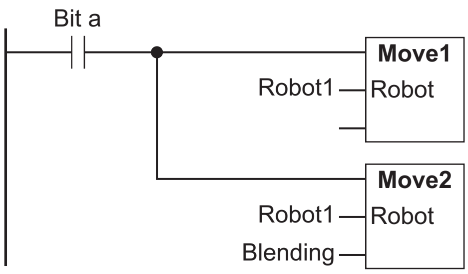
Instructions that are expressed in the user program are actually executed in order from the top. In other words, the instruction that is placed at the top of a ladder diagram or ST program is executed first, and then the subsequent instructions are executed. Therefore, the operations in order of execution for more than one instruction to the same OMRON robot within the same task period is the same operations as if the instruction are executed with multi-execution of instructions in order of the user program.

There is a maximum communications data size per task period of primary periodic task for an OMRON robot.

Therefore, if instructions are executed for an OMRON robot within the same task period and the communications data size for a task period exceeds upper limit, instructions for the different OMRON robot may be executed even if the order in the user program is later operation.
If you control the order of operation for the OMRON robots, perform a necessary interlock control using output variables that indicate the execution status for robot control instructions.

In the following programming, the instruction instances, Move1 and Move2, start in the same task period when bit *a* turns ON.

- Instructions in a program are executed from the top. Therefore Move1 is started first, and then Move2 is started before Move1 is finished.
- This is considered multi-execution of the robot control instructions. In this example, *Blending* is used to execute Move2 in relation to Move1.



6-2-6 Executing Robot Control Instructions to Uncreated Robots

If the robot control instruction is executed for the specified uncreated robot, the instruction will not be executed.
However, the *Busy* output variable from the instruction changes to TRUE. *Busy* changes to FALSE when *Execute* or *Enable* input variable changes to FALSE.

7

Variables and Instructions

This section describes the variables and instructions for the Robot Control Function Module.

7-1 System-defined variables for Robot Control

Of these system-defined variables, the system-defined variables that belong to the RC Function Module are called "system-defined variables for robot control".
The system-defined variables for robot control are used only when the OMRON robot is controlled by the sequence control program.
They cannot be used in V+ program.

This section provides lists of various variables.
Refer to *2-4-2 System of System-defined Variables for Robot Control* on page 2-8 for information on the system for variables.

● System-defined Variables for Robot Control

Level 1	Level 2	Level 3	Description
System-defined variables	System-defined variables for robot control	Robot control common variable	Monitor the common status of the RC Function Module.
		Robot variables	Monitor the status of each OMRON robot.
		Robot I/O variables	Read and write the robot built-in I/O of each OMRON robot.

7-1-1 Robot Control Common Variable

The variable name _RC_COM is used for the robot control common variable. The data type is _sCOMMON_REF, which is a structure.
This section describes the configuration of the robot control common variable and provides details on the members.

Variable			Data type	Name	Function
_RC_COM			_sRC_COMMON_REF	Robot Control Common Variable	
	Status		_sCOMMON_REF_STA	Robot Control Common Status	
		RunMode	BOOL	Robot Control Common Run	TRUE during RC Function Module operation.
	PFaultLvl		_sRC_REF_EVENT	Robot Control Common Partial Fault	
		Active	BOOL	Robot Control Common Partial Fault Occurrence	TRUE while there is a robot control common partial fault.
		Code	WORD	Robot Control Common Partial Fault Code	Contains the code for a robot control common partial fault. The upper four digits of the event code have the same value.
	MFaultLvl		_sRC_REF_EVENT	Robot Control Common Minor Fault	
		Active	BOOL	Robot Control Common Minor Fault Occurrence	TRUE while there is a robot control common minor fault.

Variable			Data type	Name	Function
		Code	WORD	Robot Control Common Minor Fault Code	Contains the code for a robot control common minor fault. The upper four digits of the event code have the same value.
	Obsr		_sRC_REF_EVENT	Robot Control Common Observation	
		Active	BOOL	Robot Control Common Observation Occurrence	TRUE while there is a robot control common observation.
		Code	WORD	Robot Control Common Observation Code	Contains the code for a robot control common observation. The upper four digits of the event code have the same value.
	Reserved		ARRAY[0..7] OF BYTE	---	Reserved

7-1-2 Robot Variables

The system-defined variable names _RC_RBT[0..7] are used for the robot variables. The data type is _sRC_RBT_REF, which is a structure.

This section describes the configuration of the robot variables and provides details on the members.

Variable			Data type	Name	Function
_RC_RBT[0..7]			_sRC_RBT_REF	Robot Variables	
	Status		_sRC_RBT_STA	Robot Status	
		Ready	BOOL	Robot Ready	TRUE when preparations for the OMRON robot execution are finished and the robot is stopped. The condition for being ready to execute is an AND of the following conditions. • The OMRON robot is in a Power Enabled state. • The OMRON robot is in Auto mode. • The OMRON robot is in a Calibrated state. • _RC_RBT[*].Status.Standby (Robot Standby) is TRUE. • _RC_RBT[*].DrvStatus.RunMode (Robot Run) is TRUE.
		Disabled	BOOL	Robot Disabled	TRUE while the OMRON robot is detached and stopped. The following Robot Status variables are mutually exclusive. Only one of them can be TRUE at a time. Disabled, Standby, Moving, Synchronized, Stopping, or ErrorStop
		Standby	BOOL	Robot Standby	TRUE while the OMRON robot is attached and stopped. The following Robot Status variables are mutually exclusive. Only one of them can be TRUE at a time. Disabled, Standby, Moving, Synchronized, Stopping, or ErrorStop

Variable		Data type	Name	Function
	Moving	BOOL	Robot Moving	TRUE during execution of the instructions to control an OMRON robot. The following Robot Status variables are mutually exclusive. Only one of them can be TRUE at a time. Disabled, Standby, Moving, Synchronized, Stopping, or ErrorStop
	Synchronized	BOOL	Robot Synchronizing	Always FALSE. The following Robot Status variables are mutually exclusive. Only one of them can be TRUE at a time. Disabled, Standby, Moving, Synchronized, Stopping, or ErrorStop
	Stopping	BOOL	Robot Deceleration Stopping	TRUE until the OMRON robot stops for an RC_Stop instruction. This includes when *Execute* is TRUE after the OMRON robot stops for an RC_Stop instruction. Instructions to control an OMRON robot are not executed in this state. *CommandAborted* changes to TRUE. The following Robot Status variables are mutually exclusive. Only one of them can be TRUE at a time. Disabled, Standby, Moving, Synchronized, Stopping, or ErrorStop
	ErrorStop	BOOL	Error Deceleration Stopping	This status exists when the OMRON robot controlled by the sequence control program is stopping or stopped for a robot minor fault (_RC_RBT[*].MFaultLvl.Active is TRUE). Instructions to control an OMRON robot are not executed in this state. *CommandAborted* changes to TRUE. The following Robot Status variables are mutually exclusive. Only one of them can be TRUE at a time. Disabled, Standby, Moving, Synchronized, Stopping, or ErrorStop
Details		_sRC_RBT_DET	Robot Control Status	
	Idle	BOOL	Standby	TRUE while the instructions to control an OMRON robot and the stop control are not executed.

Variable		Data type	Name	Function
DrvStatus		_sRC_RBT_STA_DRV	Robot Drive Status	
	RunMode	BOOL	Robot Run	TRUE when the OMRON robot is ready for operation. FALSE immediately after the power supply to the Robot Integrated CPU Unit and to the OMRON robot is turned ON. The Robot Integrated CPU Unit automatically starts operation preparations for the OMRON robot once communications with the OMRON robot are established after the power is turned ON, and the value changes to TRUE when the operation preparations are completed. If the communications between the Robot Integrated CPU Unit and the OMRON robot are disconnected or a communications error occurs, the value changes to FALSE. If the robot control instructions to control the OMRON robot is executed during FALSE, an error will occur.
	PowerEnabled	BOOL	Power Enabled	TRUE when high power of the OMRON robot is enabled.
	Enabled	BOOL	Robot Enabled	TRUE when the OMRON robot is enabled.
	Calibrated	BOOL	Calibrated	TRUE if the calibration of the OMRON robot is completed.
	DryRun	BOOL	DRY.RUN Enabled	TRUE when sending of motion control instructions to the OMRON robot is disabled.
	ESTOP	BOOL	Robot ESTOP	TRUE while the OMRON robot is during execution of the emergency stop processing. *1 The emergency stop processing is executed in any of the followings. • Execute the ESTOP V+ keyword. • Stop the OMRON robot due to a minor fault. • Turn ON the ESTOP hardware switch on the OMRON robot. This variable changes to FALSE when the emergency stop processing is completed or the ESTOP hardware switch is turned OFF.
	Comp	BOOL	COMP Mode	TRUE when the OMRON robot is set to **COMP (Computer) Mode**. The following Robot Status variables are mutually exclusive. Only one of them can be TRUE at a time. Comp, Joint, World, Tool, or Free
	Joint	BOOL	Joint Mode	TRUE if the the OMRON robot is set to **Joint Mode**. The following Robot Status variables are mutually exclusive. Only one of them can be TRUE at a time. Comp, Joint, World, Tool, or Free

Variable			Data type	Name	Function
		World	BOOL	World Mode	TRUE if the the OMRON robot is set to **World Mode**. The following Robot Status variables are mutually exclusive. Only one of them can be TRUE at a time. Comp, Joint, World, Tool, or Free
		Tool	BOOL	Tool Mode	TRUE if the the OMRON robot is set to **Tool Mode**. The following Robot Status variables are mutually exclusive. Only one of them can be TRUE at a time. Comp, Joint, World, Tool, or Free
		Free	BOOL	Free Mode	TRUE if the the OMRON robot is set to **Free Mode**. The following Robot Status variables are mutually exclusive. Only one of them can be TRUE at a time. Comp, Joint, World, Tool, or Free
		Manual	BOOL	Manual Mode	TRUE if the front panel switch is set to **Manual mode**. FALSE if the switch is set to **Auto mode**.
		Righty	BOOL	Right Arm Setting	Shows if the right arm setting or left arm setting is enabled for the OMRON robot. TRUE: Right Arm Setting FALSE: Left Arm Setting
		Below	BOOL	Below Setting	Shows if the below setting or above setting is enabled for the OMRON robot. TRUE: Below Setting FALSE: Above Setting
		Flip	BOOL	Flip Setting	Indicates the flip setting of the OMRON robot. TRUE: Enabled FALSE: Disabled
		MotionCompleted	UINT	Robot Motion Progress	Shows the current progress of robot motion. The unit is %. When a robot motion is complete or a robot is not moving, the value is 100. While continuous-path motions are applied to multiple robot motions, the robot motion progress increases till the next motion is started, and when the next motion is started, outputs the progress from 0 again . During continuous-path motion, the next motion starts before the completion of the previous motion, therefore the value returns to 0 before reaching 100. *2

Variable			Data type	Name	Function
	TCPActPos		_sRC_TCP_POS	Actual Position of TCP for OMRON Robot	
		X	LREAL	X-axis Position	Actual X-axis position of TCP for the OMRON robot
		Y	LREAL	Y-axis Position	Actual Y-axis position of TCP for the OMRON robot
		Z	LREAL	Z-axis Position	Actual Z-axis position of TCP for the OMRON robot
		Yaw	LREAL	Yaw Angle	Actual yaw angle position of TCP for the OMRON robot
		Pitch	LREAL	Pitch Angle	Actual pitch angle position of TCP for the OMRON robot
		Roll	LREAL	Roll Angle	Actual roll angle position of TCP for the OMRON robot
	JointActPos		_sRC_JOINT_POS	Actual Position of Each Joint of OMRON Robot	
		J1	LREAL	Joint1 Axis Position	Actual position of Joint1 for the OMRON robot
		J2	LREAL	Joint2 Axis Position	Actual position of Joint2 for the OMRON robot
		J3	LREAL	Joint3 Axis Position	Actual position of Joint3 for the OMRON robot
		J4	LREAL	Joint4 Axis Position	Actual position of Joint4 for the OMRON robot
		J5	LREAL	Joint5 Axis Position	Actual position of Joint5 for the OMRON robot
		J6	LREAL	Joint6 Axis Position	Actual position of Joint6 for the OMRON robot
	MFaultLvl		_sRC_REF_EVENT	Robot Minor Fault	
		Active	BOOL	Robot Minor Fault Occurrence	TRUE while there is a robot minor fault.
		Code	WORD	Robot Minor Fault Code	Contains the code for a robot minor fault. The upper four digits of the event code have the same value.
	Obsr		_sRC_REF_EVENT	Robot Observation	
		Active	BOOL	Robot Observation Occurrence	TRUE while there is a robot observation.
		Code	WORD	Robot Observation Code	Contains the code for a robot observation. The upper four digits of the event code have the same value.
	Cfg		_sRC_RBT_CFG	Robot Basic Settings	
		RobotNo	UINT	Robot Number	Contains the logical number of the OMRON robot in the RobotControlSetting window. A robot number is used for a logical number of a robot in the V+ program. The range is from 1 to 8.
		RobotEnable	_eRC_ROBOT_USE	Robot Use	Shows if the OMRON robot is used or unused. 0: _rcNoneRobot (Uncreated robot) 2: _rcUsedRobot (Used robot)
		Reserved	ARRAY[0..7] OF BYTE	---	Reserved
	Reserved		ARRAY[0..31] OF BYTE	---	Reserved

*1. The emergency stop processing means that the OMRON robot is detached and the hight power turns OFF.

*2. Refer to *6-1-5 Multi-execution of Instructions with BufferMode* on page 6-3 for details.

7-1-3 Robot I/O Variables

The system-defined variable names _RC_RBT_IO[0..7] are used for the robot I/O variables. The data type is _sRC_RBT_IO_REF, which is a structure.
This section describes the configuration of the robot I/O variables and provides details on the members.

Variable			Data type	Name	R/W	Function
_RC_RBT_IO[0..7]			_sRC_RBT_IO_REF	Robot I/O Variables		
	XIO		_sRC_RBT_IO_XIO	XIO		
		Input	ARRAY[0..11] OF BOOL	XIO Input	R	Input of XIO.
		Output	ARRAY[0..7] OF BOOL	XIO Output	RW	Output of XIO.
	RO		ARRAY[0..3] OF BOOL	Robot Output	RW	Output related to the end effector.
	IOBlox1		ARRAY[0..3] OF _sRC_RBT_IO_IOBlox	IO Blox1*1		
		Input	ARRAY[0..7] OF BOOL	IO Blox1 Input	R	Input of IO Blox1[0..3].
		Output	ARRAY[0..7] OF BOOL	IO Blox1 Output	RW	Output of IO Blox1[0..3].
	IOBlox2		ARRAY[0..3] OF _sRC_RBT_IO_IOBlox	IO Blox2*1		
		Input	ARRAY[0..7] OF BOOL	IO Blox2 Input	R	Input of IO Blox2[0..3].
		Output	ARRAY[0..7] OF BOOL	IO Blox2 Output	RW	Output of IO Blox2[0..3].
	Reserved		ARRAY[0..7] OF BYTE	---	R	Reserved

*1. The maximum number of IOBlox is 4 x 2 groups.
If there is no IOBlox, the information is not mapped.

Refer to the OMRON robot manuals for information on the IO Blox.

7-2 Instructions

The instructions related to the RC Function Module are listed below.

Classification	Type	Description
Robot Control Instructions	Common commands	Common instructions for the RC Function Module
	Robot commands	Instructions for the RC Function Module to perform robot control
System Control Instructions	---	Instructions to control a system. The RC Function Module uses instructions to manage errors.

7-2-1 Common Commands

This section describes the common instructions for the RC Function Module.
The Administration/Motion Column gives "Administration" for non-motion instructions and "Motion" for motion instructions.

Instruction	Instruction name	Description	Administration/Motion	Reference
RC_ExecVpPrgTask	Execute V+ Task	Starts execution of the specified V+ task.	Administration	page 8-2
RC_AbortVpPrgTask	Abort V+ Task	Aborts execution of the specified V+ task.	Administration	page 8-6
RC_GetVpPrgTaskStatus	Get V+ Task Status	Reads the specified V+ task status.	Administration	page 8-8
RC_ConvertCoordSystem	Convert Coordinate System	Converts the coordinates used by the RC Function Module into the NJ Robotics function coordinate system, or vice versa.	Administration	page 8-11
RC_WriteVParameter	Write Vision Parameters	Sets the vision parameters of the specified IPC Application Controller.	Administration	page 8-14
RC_ReadVParameter	Read Vision Parameters	Obtains the vision parameters of the specified IPC Application Controller.	Administration	page 8-17
RC_VRun	Run Vision Sequence	Runs the vision sequence of the specified IPC Application Controller.	Administration	page 8-19

7-2-2 Robot Commands

This section describes the instructions for the RC Function Module to perform robot control.
The Administration/Motion Column gives "Administration" for non-motion instructions and "Motion" for motion instructions.

Instruction	Instruction name	Description	Administration/Motion	Reference
RC_EnablePower	Enable Robot High Power	Turns ON (Enables) high power for the OMRON robot.	Administration	page 9-2
RC_DisablePower	Disable Robot High Power	Turns OFF (Disables) high power for the OMRON robot.	Administration	page 9-5
RC_Calibrate	Robot Calibration	Executes calibration of the OMRON robot. Executing this instruction after you turn ON the power supply to the OMRON robot to make it controllable from the program.	Motion	page 9-8

Instruction	Instruction name	Description	Administra-tion/Motion	Reference
RC_AttachRobot	Attach Robot	Makes the specified OMRON robot controllable from the sequence control program.	Administra-tion	page 9-10
RC_DetachRobot	Detach Robot	Detaches the specified OMRON robot from the sequence control program and makes the robot controllable from other programs that are not the sequence control program.	Administra-tion	page 9-13
RC_SetTool-Transform	Set Tool Conversion Coordinates	Sets or releases the tool conversion coordinates for the OMRON robot.	Administra-tion	page 9-15
RC_MoveDirect	Robot Joint Interpolation	Enables joint interpolation operation of the OMRON robot.	Motion	page 9-20
RC_MoveLinear	Robot Linear Interpolation	Enables linear interpolation operation of the OMRON robot.	Motion	page 9-27
RC_SyncPa-thOffset	Robot Path Compensation	Periodically sends robot path target offset to the specified OMRON robot.	Motion	page 9-33
RC_Stop	Stop Robot	Stops the current operation of the OMRON robot to cause a deceleration stop.	Motion	page 9-42
RC_Reset	Reset Robot Error	Clears an OMRON robot error.	Administra-tion	page 9-45

7-2-3 System Control Instructions

This section describes the instructions to control a system, which are related to the RC Function Module.

The Administration/Motion Column gives "Administration" for non-motion instructions and "Motion" for motion instructions.

Instruction	Instruction name	Description	Administration/ Motion	Reference
ResetRCEr-ror	Reset Robot Control Error	Resets current Controller errors in the Robot Control Function Module.	Administration	page 10-2
GetRCError	Get Robot Control Error Status	Gets the highest event level (partial fault or minor fault) and highest level event code of the current Controller errors in the Robot Control Function Module.	Administration	page 10-4

8

Common Command Instructions

This section describes the common instructions of the robot control instructions.

RC_ExecVpPrgTask

The RC_ExecVpPrgTask instruction starts execution of the specified V+ task.

Instruction	Name	FB/FUN	Graphic expression	ST expression
RC_ExecVpPrgTask	Execute V+ Task	FB	RC_ExecVpPrgTask_instance RC_ExecVpPrgTask Execute — Done TaskNo — Busy PrgName — CommandAborted PrgParam — Error CycleNumber — ErrorID StartStep	RC_ExecVpPrgTask_instance(Execute :=*parameter*, TaskNo :=*parameter*, PrgName :=*parameter*, PrgParam :=*parameter*, CycleNumber :=*parameter*, StartStep :=*parameter*, Done =>*parameter*, Busy =>*parameter*, CommandAborted =>*parameter*, Error =>*parameter*, ErrorID =>*parameter*);

Variables

Input Variables

Input variable	Name	Data type	Valid range	Default	Description
Execute	Execute	BOOL	TRUE or FALSE	FALSE	The instruction is executed when Execute changes to TRUE.
TaskNo	Task Number	UINT	0 to 63	0	Specifies the V+ task number.
PrgName	Program Name	STRING	---	“”	Specifies the name of the V+ program. *1
PrgParam	Parameter List	STRING	---	“”	This is a comma-delimited list of constants, variables, or expressions used as arguments in the V+ program.
CycleNumber	Cycle Number	INT	-32,768 to 32,767	1	Specifies how many times to execute the program. When a negative value is specified, the program is executed an unlimited number of times.
StartStep	Start Step	UDINT	0 to 32,767	0	Specifies the step to start executing the program. *2

*1. There are restrictions on the name of the V+ program. Refer to *Restrictions on V+ Program Names* on page 8-4 for details.

*2. A step is a line number in the V+ Program Editor Tab Page. If *StartStep* (Start Step) is omitted, the first enabled step in the program becomes the start step.

Output Variables

Output variable	Name	Data type	Valid range	Description
Done	Done	BOOL	TRUE or FALSE	TRUE when the command from the instruction to the RC Function Module 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. The lower four digits of the event code give the error code for ErrorID. Refer to *11-3 Error Table* on page 11-8 for details.

● Output Variable Update Timing

Output variable	Timing for changing to TRUE	Timing for changing to FALSE
Done	When the command from the instruction to the RC Function Module is completed.	• When *Execute* is TRUE and changes to FALSE. • After one period when *Execute* is FALSE.
Busy	When *Execute* changes to TRUE.	• When *Done* changes to TRUE. • When *Error* changes to TRUE.
CommandAborted	When the instruction is executed before the relevant V+ task is started while the RC_ExecVpPrgTask instruction was already executed.	When *Execute* changes to FALSE.
Error	When there is an error in the execution conditions or input variables for the instruction.	• When *Execute* is TRUE and changes to FALSE. • After one period when *Execute* is FALSE.

Function

- The RC_ExecVpPrgTask instruction assigns the V+ program specified by *PrgName* (Program Name) to the V+ task specified by the *TaskNo* (Task Number) input variable and executes the program.
- The program is executed from the step specified by *StartStep* (Start Step) as many as the number of times specified by *CycleNumber* (Cycle Number).
  If *StartStep* (Start Step) is omitted, the first enabled step in the program becomes the start step.
  A step is a line number in the V+ Program Editor Tab Page.
- The V+ task is started and *Done* changes to TRUE.
- You can check the status of the executed V+ task with the RC_GetVpPrgTaskStatus (Get V+ Task Status) instruction.
  In addition, you can abort the V+ task with the RC_AbortVpPrgTask (Abort V+ Task) instruction.

Instruction Details

● Task Number and Program Name

Specify the task number (0 to 63) of the V+ task to execute and the name of the V+ program.
The name of the V+ program can be omitted. When the name is omitted, the instruction executes the V+ program that is assigned to the specified V+ task number.

A V+ Program Error (96040000 hex) will occur in the following cases.

- The specified V+ task is already executed.
- The specified V+ task number is incorrect.
- The V+ program with the specified program name does not exist.
- The program name is omitted, but the V+ program is not assigned to the V+ task.

Refer to the *eV+3 User's Manual (Cat. No. I651)* for information on how to name the V+ program.

● Restrictions on V+ Program Names

The V+ program names are subject to the following restrictions.

Item	Restriction
Maximum length of the V+ program names	15 characters
Applicable characters for V+ program names	0 to 9, a to z, ".", and "_" The first character must be a to z.

● Parameter List

Specify as a text string the comma-delimited list of constants, variables, or expressions corresponding to the arguments of the **.PROGRAM** statement in the V+ program to execute. The variables or expressions are interpreted by the V+ program.
If the parameter list disagrees with the arguments in V+ program that is downloaded, a V+ Program Error (96040000 hex) will occur.

Refer to the *eV+3 Keyword Reference Manual (Cat. No. I652)* for details on the format when arguments are omitted.

● Cycle Number

Specify how many times execute the V+ program repeatedly. When this variable is omitted or *0* is specified, the program will be executed only once.
You can specify to repeat the program up to 32,767 times.
When a negative value is specified, the program will be executed an unlimited number of times repeatedly. The conditions for stopping the repeated program execution are given below.

- The V+ program is stopped with the RC_AbortVpPrgTask (Abort V+ Task) instruction.
- The HALT instruction is executed in the V+ program.
- A V+ Program Error (96040000 hex) occurred.

● Start Step

Specify in which step to start executing the V+ program. When this variable is omitted or *0* is specified, the program will be executed from the first step.
You can specify the number of steps up to 32,767.

A Starting Step Setting Out of Range (55140000 hex) will occur if you specify a value that is out of the valid range.
If you specify a value that exceeds the number of steps in the V+ program to execute, the instruction will start executing the program from the last step of the program.

Error

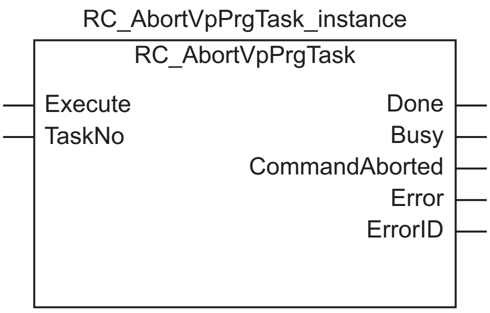
If an error occurs during instruction execution, *Error* will change to TRUE.
You can find out the cause of the error by referring to the value output by *ErrorID* (Error Code).

● Error Code

Refer to *11-3 Error Table* on page 11-8 for details on the error codes.

RC_AbortVpPrgTask

The RC_AbortVpPrgTask instruction aborts execution of the specified V+ task.

Instruction	Name	FB/FUN	Graphic expression	ST expression
RC_AbortVpPrg-Task	Abort V+ Task	FB		RC_AbortVpPrgTask_instance(Execute :=*parameter*, TaskNo :=*parameter*, Done =>*parameter*, Busy =>*parameter*, CommandAborted =>*parameter*, Error =>*parameter*, ErrorID =>*parameter*);

Variables

Input Variables

Input variable	Name	Data type	Valid range	Default	Description
Execute	Execute	BOOL	TRUE or FALSE	FALSE	The instruction is executed when Execute changes to TRUE.
TaskNo	Task Number	UINT	0 to 63	0	Specifies the V+ task number.

Output Variables

Output variable	Name	Data type	Valid range	Description
Done	Done	BOOL	TRUE or FALSE	TRUE when the command from the instruction to the RC Function Module 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. The lower four digits of the event code give the error code for ErrorID. Refer to *11-3 Error Table* on page 11-8 for details.

Output Variable Update Timing

Output variable	Timing for changing to TRUE	Timing for changing to FALSE
Done	When the command from the instruction to the RC Function Module is completed.	• When *Execute* is TRUE and changes to FALSE. • After one period when *Execute* is FALSE.
Busy	When *Execute* changes to TRUE.	• When *Done* changes to TRUE. • When *Error* changes to TRUE.
CommandAborted	When the instruction is executed before the relevant V+ task is started while the RC_ExecVpPrgTask instruction was already executed.	When *Execute* changes to FALSE.
Error	When there is an error in the execution conditions or input variables for the instruction.	• When *Execute* is TRUE and changes to FALSE. • After one period when *Execute* is FALSE.

Function

- The RC_AbortVpPrgTask instruction aborts execution of the specified V+ task.
  Use this instruction to abort a V+ task that is executed by the RC_ExecVpPrgTask (Execute V+ Task) instruction.
- If this instruction was executed, the OMRON robot stops after completion of the current operation.
  During continuous-path motion, the robot stops on completion of the operation immediately after the current operation.

Task Number

Specify the task number (0 to 63) of the V+ task to abort.

A V+ Program Error (96040000 hex) will occur in the following cases.

- The specified V+ task is not executed.
- The specified V+ task number is incorrect.
- The V+ program is not assigned to the specified V+ task number.

RC_GetVpPrgTaskStatus

The RC_GetVpPrgTaskStatus instruction reads the specified V+ task status.

Instruction	Name	FB/FUN	Graphic expression	ST expression
RC_GetVpPrg-TaskStatus	Get V+ Task Status	FB	RC_GetVpPrgTaskStatus_instance RC_GetVpPrgTaskStatus Enable — Valid TaskNo — TaskStatus Busy Error ErrorID	RC_GetVpPrgTaskStatus_instance(Enable :=*parameter*, TaskNo :=*parameter*, Valid =>*parameter*, TaskStatus=>*parameter*, Busy =>*parameter*, Error =>*parameter*, ErrorID =>*parameter*);

Variables

Input Variables

Input variable	Name	Data type	Valid range	Default	Description
Enable	Enable	BOOL	TRUE or FALSE	FALSE	The instruction is executed while the value of this variable is TRUE.
TaskNo	Task Number	UINT	0 to 63	0	Specifies the V+ task number.

Output Variables

Output variable	Name	Data type	Valid range	Description
Valid	Valid	BOOL	TRUE or FALSE	TRUE when the V+ task status is read.
TaskStatus	V+ Task Status	INT	-1 to +7	Outputs the value of the V+ task status.
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.

*1. The lower four digits of the event code give the error code for ErrorID. Refer to *11-3 Error Table* on page 11-8 for details.

● Output Variable Update Timing

Output variable	Timing for changing to TRUE	Timing for changing to FALSE
Valid	When the instruction was enabled and read the V+ task status for the first time.	• When *Enable* changes to FALSE. • When *Error* changes to TRUE.
Busy	When *Enable* changes to TRUE.	• When *Enable* changes to FALSE. • When *Error* changes to TRUE.
Error	When there is an error in the execution conditions or input variables for the instruction.	When *Enable* changes to FALSE.

Function

The RC_GetVpPrgTaskStatus instruction continues reading the specified V+ task status while the *Enable* input variable is TRUE.
It may take more than one control cycle until the instruction reads V+ task status for the first time after the *Enable* input variable changes to TRUE. After the *Valid* output variable changes to TRUE, reference the value of the *TaskStatus* (V+ Task Status) output variable.

● Task Number

Specify the task number (0 to 63) of the V+ task from which you want to read the status.
An error will occur if the specified V+ task number is incorrect.

● V+ Task Status

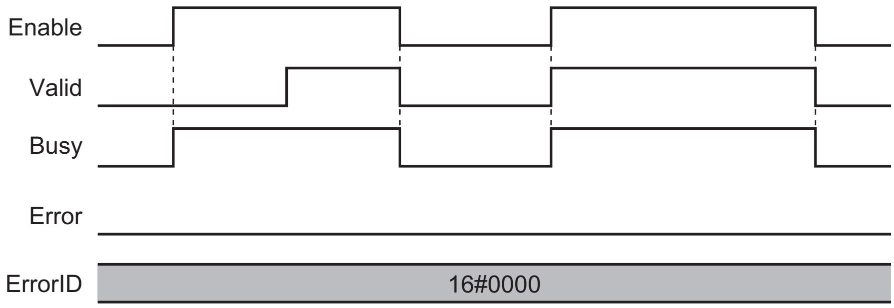
The following table shows the values that you can read as the V+ task status and their meaning.

Status value	State of V+ task
0	Idle. The V+ program is not assigned to the V+ task.
1	Stop due to program completion.
2	Stop due to a program execution error. For example, an undefined variable is referenced.
3	Stop due to an ABORT, the RC_AbortVpPrgTask (Abort V+ Task) instruction, a press of the E-Stop Button, a robot error, or a watch point.
4	Executing
5	Stop due to a PAUSE or breakpoint.
7	Stop due to single step execution.

Timing Charts

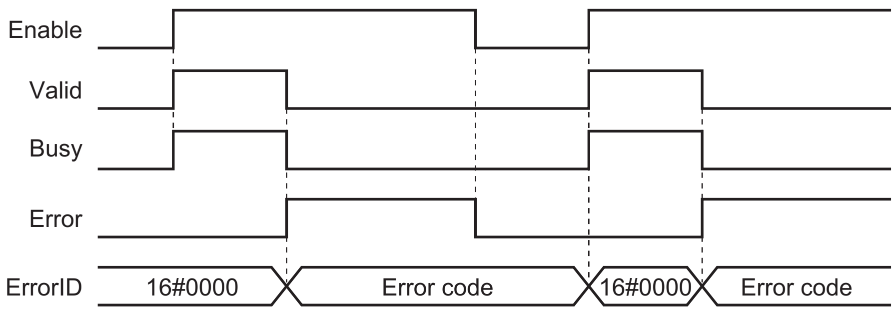
Busy (Executing) changes to TRUE when *Enable* changes to TRUE.

● When the Instruction Ended Normally



● When an Error Occurred

When an error occurs, *Error* changes to TRUE and *Busy* (Executing) and *Valid* (Enabled) change to FALSE.
When *Enable* changes from TRUE to FALSE, *Busy* (Executing), *Valid* (Enabled), and *Error* change to FALSE.
If an error occurred, the error code for *ErrorID* (Error Code) is set.
The *ErrorID* (Error Code) is retained after *Error* changes to FALSE. 16#0000 is set in *ErrorID* (Error Code) when *Enable* changes from FALSE to TRUE.



Error

If an error occurred during instruction execution, *Error* changes to TRUE and *TaskStatus* (V+ Task Status) is not read.
You can find out the cause of the error by referring to the value output by *ErrorID* (Error Code).

● Error Code

Refer to *11-3 Error Table* on page 11-8 for details on the error codes.

Precautions for Correct Use

Make sure that you use this instruction only once for one V+ task in order to avoid inconsistency of V+ task status in the same control cycle.
If you execute two of this instruction for the same task number at the same time, the second instruction is not executed until the *Enable* input variable of the first instruction turns FALSE.

RC_ConvertCoordSystem

The RC_ConvertCoordSystem instruction converts the coordinates used by the Robot Control Function Module to the NJ Robotics function coordinate system, or vice versa.

Instruction	Name	FB/FUN	Graphic expression	ST expression
RC_ConvertCoord-System	Convert Coordinate System	FUN	(@)RC_ConvertCoordSystem EN ENO ConvertForm Out InCoord OutCoord	RC_ConvertCoordSystem (ConvertForm := *parameter*, InCoord := *parameter*, ENO => *parameter*, OutCoord => *parameter*);

Variables

Input Variables

Input variable	Name	Data type	Valid range	Default	Description
EN	Enable (Execution Condition)	BOOL	TRUE or FALSE	TRUE	TRUE: Instruction is executed. *1 FALSE: Instruction is not executed.
ConvertForm	Conversion Format	_eRC_CONVERT_FORM	1: _rcRCToNJRobotics 2: _rcNJRoboticsToRC	1	Specifies the coordinate system specification format. 1: Conversion to the NJ Robotics function coordinate system 2: Conversion from the NJ Robotics function coordinate system
InCoord	Data to Convert	ARRAY[0..5] OF LREAL	Negative number, positive number, or 0	0	Position of the input coordinate system

*1. If input upward differentiation (@) is specified as an instruction option, the execution condition is when the value of *EN* changes from FALSE to TRUE.

Output Variables

Output variable	Name	Data type	Valid range	Description
ENO	Enable Output	BOOL	TRUE	Always TRUE
Out	Return Value	BOOL	TRUE or FALSE	TRUE: Conversion succeeded FALSE: Conversion failed
OutCoord	Conversion Result	ARRAY[0..5] OF LREAL	Negative number, positive number, or 0	Position of the output coordinate system

Function

The RC_ConvertCoordSystem instruction is used if the coordinate position of the OMRON robot is converted to the coordinate position of the NJ Robotics or the coordinate position of the NJ Robotics is converted to the coordinate position of the OMRON robot.

If the conversion is successful, *ENO* (Enable Out) output variable will change to TRUE. If the conversion fails, the variable will change to FALSE.

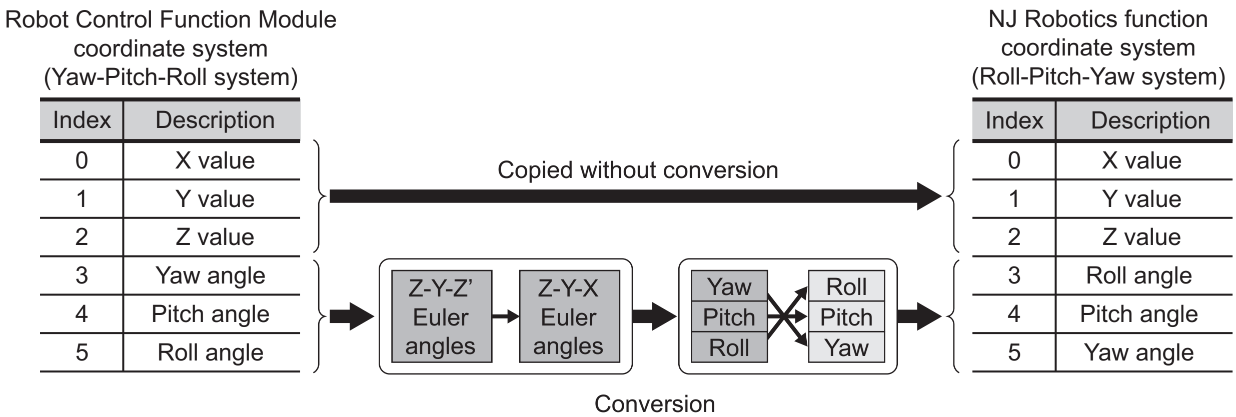
In the following cases, the conversion fails.

- *ConvertForm* input variable is outside of the valid range
- *InCoord* (Data to Convert) is nonnumeric data.

Conversion to NJ Robotics Function Coordinate System

To convert the RC Function Module coordinate system to the NJ Robotics function coordinate system, set *ConvertForm* (Conversion Format) to **1: _rcToNJRobotics**. Then, the position in the Yaw-Pitch-Roll coordinate system is converted to the position in the Roll-Pitch-Yaw coordinate system and output to *OutCoord* (Conversion Result).

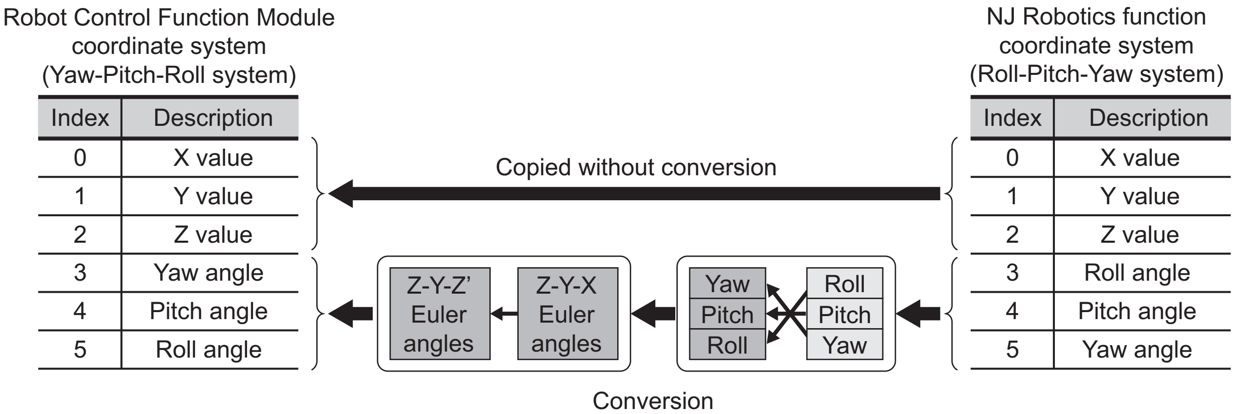
If Roll, Pitch, or Yaw is ±180°, the output value will be +180°. For a rotating coordinate system, the instruction converts Z-Y-Z’ Euler angles to Z-Y-X Euler angles.



Conversion to Robot Control Function Module Coordinate System

To convert the NJ Robotics function coordinate system to the RC Function Module coordinate system, set *ConvertForm* (Conversion Format) to **2: _rcFromNJRobotics**. Then, the position in the Roll-Pitch-Yaw coordinate system is converted to the position in the Yaw-Pitch-Roll coordinate system and output to *OutCoord* (Conversion Result).

For a rotating coordinate system, the instruction converts Z-Y-X Euler angles to Z-Y-Z’ Euler angles.


Robot Control Function Module
coordinate system
(Yaw-Pitch-Roll system)
Index
Description
0
X value
1
Y value
2
Z value
3
Yaw angle
4
Pitch angle
5
Roll angle
Copied without conversion
Z-Y-Z'
Euler
angles
Z-Y-X
Euler
angles
Yaw
Pitch
Roll
Roll
Pitch
Yaw
Conversion
NJ Robotics function
coordinate system
(Roll-Pitch-Yaw system)
Index
Description
0
X value
1
Y value
2
Z value
3
Roll angle
4
Pitch angle
5
Yaw angle

RC_WriteVParameter

The RC_WriteVParameter instruction sets the vision parameters of the specified IPC Application Controller.

Instruction	Name	FB/FUN	Graphic expression	ST expression
RC_WriteVParameter	Write Vision Parameters	FB	RC_WriteVParameter_instance RC_WriteVParameter Execute — Done IPAddr — Busy SequenceID — CommandAborted ToolID — Error ParameterID — ErrorID IndexID ObjectID Value	RC_WriteVParameter_instance(Execute :=*parameter*, IPAddr :=*parameter*, SequenceID :=*parameter*, ToolID :=*parameter*, ParameterID :=*parameter*, IndexID :=*parameter*, ObjectID :=*parameter*, Value :=*parameter*, Done =>*parameter*, Busy =>*parameter*, CommandAborted =>*parameter*, Error =>*parameter*, ErrorID =>*parameter*,);

Variables

Input Variables

Input variable	Name	Data type	Valid range	Default	Description
Execute	Execute	BOOL	TRUE or FALSE	FALSE	The instruction is executed when Execute changes to TRUE.
IPAddr	IP Address	STRING	*1	“”	Specifies the IP address of the IPC Application Controller. Specifies in the standard IP address format. *2
SequenceID	Sequence ID	DINT	Full range	1	Specifies the index of a vision sequence. The first sequence is *1*.
ToolID	Tool ID	DINT	Full range	1	Specifies the tool index of a vision sequence as an option. The first tool is *1*.
ParameterID	Parameter ID	DINT	Full range	1	Specifies the parameter identifier (ID) as an option.
IndexID	Index ID	DINT	Full range	1	Specifies the index ID as an option.
ObjectID	Object ID	DINT	Full range	1	Specifies the index of an object. Some parameters require the object index to access a specific value in an array.

Input variable	Name	Data type	Valid range	Default	Description
Value	Setting Value	LREAL	Full range	0	Specifies the value to write.

*1. 16 bytes max. (15 bytes + NULL)

*2. An example of the standard IP address format is "192.168.1.120".

Output Variables

Output variable	Name	Data type	Valid range	Description
Done	Done	BOOL	TRUE or FALSE	TRUE when the command from the instruction to the RC Function Module 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. The lower four digits of the event code give the error code for ErrorID. Refer to *11-3 Error Table* on page 11-8 for details.

● Output Variable Update Timing

Output variable	Timing for changing to TRUE	Timing for changing to FALSE
Done	When the command from the instruction to the RC Function Module is completed.	• When *Execute* is TRUE and changes to FALSE. • After one period when *Execute* is FALSE.
Busy	When *Execute* changes to TRUE.	• When *Done* changes to TRUE. • When *Error* changes to TRUE.
CommandAborted	When another instruction causes an error and aborts this instruction.	When *Execute* changes to FALSE.
Error	When there is an error in the execution conditions or input variables for the instruction.	• When *Execute* is TRUE and changes to FALSE. • After one period when *Execute* is FALSE.

Function

- The RC_WriteVParameter instruction sets the vision parameters specified for *SequenceID* (Sequence ID), *ToolID* (Tool ID), *ParameterID* (Parameter ID), *IndexID* (Index ID) and *ObjectID* (Object ID) to the IPC Application Controller specified for the *IPAddr* (IP Address) input variable.
- Setting the vision parameters is used for recipe change.
  Refer to *5-6 Changing Recipe* on page 5-14 for details.
- The RC_WriteVParameter (Write Vision Parameters) instruction has the same functionality as the VPARAMETER V+ program command keyword.
  Refer to the *eV+3 Keyword Reference Manual (Cat. No. I652)* for details.

Sample Programming

Refer to *Sample Programming* on page 5-15 for sample programming for changing recipe.

RC_ReadVParameter

The RC_ReadVParameter instruction obtains the vision parameters of the specified IPC Application Controller.

Instruction	Name	FB/FUN	Graphic expression	ST expression
RC_ReadVParameter	Read Vision Parameters	FB	RC_ReadVParameter_instance RC_ReadVParameter Execute — Done IPAddr — Busy SequenceID — Value ToolID — CommandAborted ParameterID — Error IndexID — ErrorID ObjectID	RC_ReadVParameter_instance(Execute :=*parameter*, IPAddr :=*parameter*, SequenceID :=*parameter*, ToolID :=*parameter*, ParameterID :=*parameter*, IndexID :=*parameter*, ObjectID :=*parameter*, Done =>*parameter*, Busy =>*parameter*, Value =>*parameter*, CommandAborted =>*parameter*, Error =>*parameter*, ErrorID =>*parameter*,);

Variables

Input Variables

Input variable	Name	Data type	Valid range	Default	Description
Execute	Execute	BOOL	TRUE or FALSE	FALSE	The instruction is executed when Execute changes to TRUE.
IPAddr	IP Address	STRING	*1	“”	Specifies the IP address of the IPC Application Controller. Specifies in the standard IP address format. *2
SequenceID	Sequence ID	DINT	Full range	1	Specifies the index of a vision sequence. The first sequence is *1*.
ToolID	Tool ID	DINT	Full range	1	Specifies the tool index of a vision sequence as an option. The first tool is *1*.
ParameterID	Parameter ID	DINT	Full range	1	Specifies the parameter identifier (ID) as an option.
IndexID	Index ID	DINT	Full range	1	Specifies the index ID as an option.
ObjectID	Object ID	DINT	Full range	1	Specifies the index of an object. Some parameters require the object index to access a specific value in an array.

*1. 16 bytes max. (15 bytes + NULL)
*2. An example of the standard IP address format is "192.168.1.120".

Output Variables

Output variable	Name	Data type	Valid range	Description
Done	Done	BOOL	TRUE or FALSE	TRUE when the command from the instruction to the RC Function Module is completed.
Busy	Executing	BOOL	TRUE or FALSE	TRUE when the instruction is acknowledged.
Value	Setting Value	LREAL	---	TRUE when the command from the instruction to the RC Function Module is completed.
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. The lower four digits of the event code give the error code for ErrorID. Refer to *11-3 Error Table* on page 11-8 for details.

● Output Variable Update Timing

Output variable	Timing for changing to TRUE	Timing for changing to FALSE
Done	When the command from the instruction to the RC Function Module is completed.	• When *Execute* is TRUE and changes to FALSE. • After one period when *Execute* is FALSE.
Busy	When *Execute* changes to TRUE.	• When *Done* changes to TRUE. • When *Error* changes to TRUE.
CommandAborted	When another instruction causes an error and aborts this instruction.	When *Execute* changes to FALSE.
Error	When there is an error in the execution conditions or input variables for the instruction.	• When *Execute* is TRUE and changes to FALSE. • After one period when *Execute* is FALSE.

Function

- The RC_ReadVParameter instruction obtains the vision parameters specified for *SequenceID* (Sequence ID), *ToolID* (Tool ID), *ParameterID* (Parameter ID), *IndexID* (Index ID) and *ObjectID* (Object ID) from the IPC Application Controller specified for the *IPAddr* (IP Address) input variable.
- Obtaining the vision parameters is used for recipe change.
  Refer to *5-6 Changing Recipe* on page 5-14 for details.
- The RC_ReadVParameter (Read Vision Parameters) instruction has the same functionality as the VPARAMETER V+ function keyword.
  Refer to the *eV+3 Keyword Reference Manual (Cat. No. I652)* for details.

Sample Programming

Refer to *Sample Programming* on page 5-15 for sample programming for changing recipe.

RC_VRun

The RC_VRun instruction runs the vision sequence of the specified IPC Application Controller.

Instruction	Name	FB/FUN	Graphic expression	ST expression
RC_VRun	Run Vision Sequence	FB	RC_VRun_instance RC_VRun Execute — Done IPAddr — Busy SequenceID — CommandAborted Error ErrorID	RC_VRun_instance(Execute :=*parameter*, IPAddr :=*parameter*, SequenceID :=*parameter*, Done =>*parameter*, Busy =>*parameter*, Value =>*parameter*, CommandAborted =>*parameter*, Error =>*parameter*, ErrorID =>*parameter*,);

Variables

Input Variables

Input variable	Name	Data type	Valid range	Default	Description
Execute	Execute	BOOL	TRUE or FALSE	FALSE	The instruction is executed when Execute changes to TRUE.
IPAddr	IP Address	STRING	*1	“”	Specifies the IP address of the IPC Application Controller. Specifies in the standard IP address format. *2
SequenceID	Sequence ID	DINT	Full range	1	Specifies the index of a vision sequence. The first sequence is *1*.

*1. 16 bytes max. (15 bytes + NULL)

*2. An example of the standard IP address format is "192.168.1.120".

Output Variables

Output variable	Name	Data type	Valid range	Description
Done	Done	BOOL	TRUE or FALSE	TRUE when the command from the instruction to the RC Function Module 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.

Output variable	Name	Data type	Valid range	Description
ErrorID	Error Code	WORD	*1	Contains the error code when an error occurs. A value of 16#0000 indicates normal execution.

*1. The lower four digits of the event code give the error code for ErrorID. Refer to *11-3 Error Table* on page 11-8 for details.

● Output Variable Update Timing

Output variable	Timing for changing to TRUE	Timing for changing to FALSE
Done	When the command from the instruction to the RC Function Module is completed.	• When *Execute* is TRUE and changes to FALSE. • After one period when *Execute* is FALSE.
Busy	When *Execute* changes to TRUE.	• When *Done* changes to TRUE. • When *Error* changes to TRUE.
CommandAborted	When another instruction causes an error and aborts this instruction.	When *Execute* changes to FALSE.
Error	When there is an error in the execution conditions or input variables for the instruction.	• When *Execute* is TRUE and changes to FALSE. • After one period when *Execute* is FALSE.

Function

- The RC_VRun instruction starts running the vision sequence specified for *SequenceID* (Sequence ID) to the IPC Application Controller specified for the *IPAddr* (IP Address) input variable.
- Running the vision sequence is used for recipe change.
  Refer to *5-6 Changing Recipe* on page 5-14 for details.
- The RC_VRun (Run Vision Sequence) instruction has the same functionality as the VRUN V+ program command keyword.
  Refer to the *eV+3 Keyword Reference Manual (Cat. No. I652)* for details.

Sample Programming

Refer to *Sample Programming* on page 5-15 for sample programming for changing recipe.

9

Robot Command Instructions

This section describes the instructions for the Robot Control Function Module to perform robot control.

RC_EnablePower

The RC_EnablePower instruction turns ON (enables) high power for the OMRON robot.

Instruction	Name	FB/FUN	Graphic expression	ST expression
RC_EnablePower	Enable Robot High Power	FB	RC_EnablePower_instance RC_EnablePower Robot ----------------- Robot Execute Done Busy CommandAborted Error ErrorID	RC_EnablePower_instance(Robot :=*parameter*, Execute :=*parameter*, Done =>*parameter*, Busy =>*parameter*, CommandAborted => *parameter*, Error =>*parameter*, ErrorID =>*parameter*);

Variables

Input Variables

Input variable	Name	Data type	Valid range	Default	Description
Execute	Execute	BOOL	TRUE or FALSE	FALSE	The instruction is executed when Execute changes to TRUE.

Output Variables

Output variable	Name	Data type	Valid range	Description
Done	Done	BOOL	TRUE or FALSE	TRUE when the command from the instruction to the RC Function Module is completed. *1
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	*2	Contains the error code when an error occurs. A value of 16#0000 indicates normal execution.

*1. Check the instruction completion including the OMRON robot with *_RC_RBT[*].DrvStatus.PowerEnabled* robot variable because the RC Function Module sends a command to the OMRON robot after the instruction sends a command to the RC Function Module. Do not check the instruction completion with *Done* output variable.

*2. The lower four digits of the event code give the error code for ErrorID. Refer to *11-3 Error Table* on page 11-8 for details.

● Output Variable Update Timing

Output variable	Timing for changing to TRUE	Timing for changing to FALSE
Done	When the command from the instruction to the RC Function Module is completed.	• When *Execute* is TRUE and changes to FALSE. • After one period when *Execute* is FALSE.
Busy	When *Execute* changes to TRUE.	• When *Done* changes to TRUE. • When *CommandAborted* changes to TRUE. • When *Error* changes to TRUE.
CommandAborted	• When another instruction causes an error and aborts this instruction. • When the instruction is executed during *ErrorStop* (Error Deceleration Stopping) or *Stopping* (Robot Deceleration Stopping).	• When *Execute* is TRUE and changes to FALSE. • After one period when *Execute* is FALSE.
Error	When there is an error in the execution conditions or input variables for the instruction.	• When *Execute* is TRUE and changes to FALSE. • After one period when *Execute* is FALSE.

In-out Variables

In-out variable	Name	Data type	Valid range	Description
Robot	Robot	_sRC_RBT_REF	---	Specifies the OMRON robot.

Function

The RC_EnablePower instruction sends a request to turn ON (enable) high power for the specified OMRON robot to the robot.
When the ROBOT POWER Button on the front panel of the OMRON robot to which a request is received is pressed while the button is flashing, the high power is turned ON and the power is supplied to the OMRON robot's motor.

You can use any of the following operation means to request the high power ON for the OMRON robot.

- Button on the teaching pendant connected to the OMRON robot
- Button in the Sysmac Studio
- V+ keyword (ENABLE/DISABLE POWER)

Since these operations are not performed exclusive processing mutually, you will get the result of the last operation performed.
You can check the high power for the OMRON robot with *_RC_RBT[*].DrvStatus.PowerEnabled* robot variable.

Precautions for Correct Use

Although the high power of the OMRON robot is maintained even if the operating mode of the Robot Integrated CPU Unit is changed to PROGRAM mode, it is disabled if data is downloaded.

Operation When the Instruction Is Executed More Than Once

The following gives the operation when the RC_EnablePower (Enable Robot High Power) or RC_DisablePower (Disable Robot High Power) instruction is executed more than once for the same OMRON robot in the sequence control program.

● Multi-execution of Instructions

If a different instance of the RC_EnablePower (Enable Robot High Power) or RC_DisablePower (Disable Robot High Power) instruction is executed during execution of the RC_EnablePower (Enable Robot High Power) instruction, the latter instruction will cause an error.

● Simultaneous Execution of Instructions

The following describes what will happen if instructions are executed in the same task period.
Instructions that are expressed in a program are actually executed in order from the top. In other words, the instruction that is placed at the top of a ladder diagram or ST program is executed first, and then the subsequent instructions are executed. In this case, an error will occur in the second and later instructions.
This is the same operation as in multi-execution of instructions.

RC_DisablePower

The RC_DisablePower instruction turns OFF (disables) high power for the OMRON robot.

Instruction	Name	FB/FUN	Graphic expression	ST expression
RC_DisablePower	Disable Robot High Power	FB	RC_DisablePower_instance RC_DisablePower Robot ---------------- Robot Execute Done Busy CommandAborted Error ErrorID	RC_DisablePower_instance(Robot :=*parameter*, Execute :=*parameter*, Done =>*parameter*, Busy =>*parameter*, CommandAborted =>*parameter*, Error =>*parameter*, ErrorID =>*parameter*);

Variables

Input Variables

Input variable	Name	Data type	Valid range	Default	Description
Execute	Execute	BOOL	TRUE or FALSE	FALSE	The instruction is executed when Execute changes to TRUE.

Output Variables

Output variable	Name	Data type	Valid range	Description
Done	Done	BOOL	TRUE or FALSE	TRUE when the command from the instruction to the RC Function Module is completed. *1
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	*2	Contains the error code when an error occurs. A value of 16#0000 indicates normal execution.

*1. Check the instruction completion including the OMRON robot with *_RC_RBT[*].DrvStatus.PowerEnabled* robot variable because the RC Function Module sends a command to the OMRON robot after the instruction sends a command to the RC Function Module. Do not check the instruction completion with *Done* output variable.

*2. The lower four digits of the event code give the error code for ErrorID. Refer to *11-3 Error Table* on page 11-8 for details.

Output Variable Update Timing

Output variable	Timing for changing to TRUE	Timing for changing to FALSE
Done	When the command from the instruction to the RC Function Module is completed.	• When *Execute* is TRUE and changes to FALSE. • After one period when *Execute* is FALSE.
Busy	When *Execute* changes to TRUE.	• When *Execute* changes to FALSE. • When *Error* changes to TRUE.
CommandAborted	When the instruction is executed during *ErrorStop* (Error Deceleration Stopping).	When *Execute* changes to FALSE.
Error	When there is an error in the execution conditions or input variables for the instruction.	• When *Execute* is TRUE and changes to FALSE. • After one period when *Execute* is FALSE.

In-out Variables

In-out variable	Name	Data type	Valid range	Description
Robot	Robot	_sRC_RBT_REF	---	Specifies the OMRON robot.

Function

The RC_DisablePower instruction turns OFF (disables) high power for the specified OMRON robot to stop the power supply to the OMRON robot's motor.

When the power supply is stopped, the OMRON robot performs the following sequence of steps.

1. The OMRON robot decelerates to a stop according to the user-specified parameter.
2. The brake for each joint is turned ON.
3. The robot waits until the user-specified brake delay interval expires.
4. The power supply to the OMRON robot is turned OFF.
5. *_RC_RBT[*].DrvStatus.PowerEnabled* robot variable changes to FALSE.

You can use any of the following operation means to change the high power status for the OMRON robot.

- Button on the teaching pendant connected to the OMRON robot
- Button in the Sysmac Studio
- V+ keyword (ENABLE/DISABLE POWER)

Since these operations are not performed exclusive processing mutually, you will get the result of the last operation performed.
You can check the high power for the OMRON robot with *_RC_RBT[*].DrvStatus.PowerEnabled* robot variable.

Precautions for Correct Use

Although the high power of the OMRON robot is maintained even if the operating mode of the Robot Integrated CPU Unit is changed to PROGRAM mode, it is disabled if data is downloaded.

Operation When the Instruction Is Executed More Than Once

The following gives the operation when the RC_EnablePower (Enable Robot High Power) or RC_DisablePower (Disable Robot High Power) instruction is executed more than once for the same OMRON robot in the sequence control program.

● Multi-execution of Instructions

If a different instance of the RC_DisablePower (Disable Robot High Power) or RC_EnablePower (Enable Robot High Power) instruction is executed during execution of the RC_DisablePower (Disable Robot High Power) instruction, the latter instruction will cause an error.

● Simultaneous Execution of Instructions

The following describes what will happen if instructions are executed in the same task period.
Instructions that are expressed in a program are actually executed in order from the top. In other words, the instruction that is placed at the top of a ladder diagram or ST program is executed first, and then the subsequent instructions are executed. In this case, an error will occur in the second and later instructions.
This is the same operation as in multi-execution of instructions.

RC_Calibrate

The RC_Calibrate instruction executes calibration of the OMRON robot.

Instruction	Name	FB/FUN	Graphic expression	ST expression
RC_Calibrate	Robot Calibration	FB	RC_Calibrate_instance RC_Calibrate Robot ---------------- Robot Execute Done Busy CommandAborted Error ErrorID	RC_Calibrate_instance(Robot :=*parameter*, Execute :=*parameter*, Done =>*parameter*, Busy =>*parameter*, CommandAborted =>*parameter*, Error =>*parameter*, ErrorID =>*parameter*);

Variables

Input Variables

Input variable	Name	Data type	Valid range	Default	Description
Execute	Execute	BOOL	TRUE or FALSE	FALSE	The instruction is executed when Execute changes to TRUE.

Output Variables

Output variable	Name	Data type	Valid range	Description
Done	Done	BOOL	TRUE or FALSE	TRUE when the command from the instruction to the RC Function Module is completed. *1
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	*2	Contains the error code when an error occurs. A value of 16#0000 indicates normal execution.

*1. Check the instruction completion including the OMRON robot with *_RC_RBT[*].DrvStatus.Calibrated* robot variable because the RC Function Module sends a command to the OMRON robot after the instruction sends a command to the RC Function Module. Do not check the instruction completion with *Done* output variable.

*2. The lower four digits of the event code give the error code for ErrorID. Refer to *11-3 Error Table* on page 11-8 for details.

● Output Variable Update Timing

Output variable	Timing for changing to TRUE	Timing for changing to FALSE
Done	When the command from the instruction to the RC Function Module is completed.	• When *Execute* is TRUE and changes to FALSE. • After one period when *Execute* is FALSE.
Busy	When *Execute* changes to TRUE.	• When *Done* changes to TRUE. • When *CommandAborted* changes to TRUE. • When *Error* changes to TRUE.
CommandAborted	• When another instruction causes an error and aborts this instruction. • When the instruction is executed during *ErrorStop* (Error Deceleration Stopping) or *Stopping* (Robot Deceleration Stopping).	• When *Execute* is TRUE and changes to FALSE. • After one period when *Execute* is FALSE.
Error	When there is an error in the execution conditions or input variables for the instruction.	• When *Execute* is TRUE and changes to FALSE. • After one period when *Execute* is FALSE.

In-out Variables

In-out variable	Name	Data type	Valid range	Description
Robot	Robot	_sRC_RBT_REF	---	Specifies the OMRON robot.

Function

- The RC_Calibrate instruction executes calibration of the specified OMRON robot.
- This calibration makes the OMRON robot controllable from a user program.
- You can check the calibration completion with *_RC_RBT[*].DrvStatus.Calibrated* robot variable.
- If this instruction is executed for the OMRON robot whose calibration was completed, execute the calibration again.
- In the following case, executing the instruction causes an error.
  a) The high power for the specified OMRON robot is not turned ON.
  b) The specified OMRON robot is attached from a sequence control program.
  c) The specified OMRON robot is attached from a V+ program.

Additional Information

- Depending on the OMRON robot model, the robot may perform automatic calibration when the power is turned ON or you must execute calibration after the power is turned ON.
  The calibration execution is required for the eCobra. However, other robots that are not the eCobra perform calibration automatically.
  Refer to the relevant manuals for the OMRON robot for details.
  Even if the calibration information for the OMRON robot was lost, the calibration is completed by executing this instruction.
- The calibration completion is maintained if the operating mode of the Robot Integrated CPU Unit is changed to PROGRAM mode. If data is downloaded, the calibration is required again.

RC_AttachRobot

The RC_AttachRobot instruction makes the specified OMRON robot controllable from the sequence control program.

Instruction	Name	FB/FUN	Graphic expression	ST expression
RC_AttachRobot	Attach Robot	FB	RC_AttachRobot_instance RC_AttachRobot Robot ----------------- Robot Execute Done Busy CommandAborted Error ErrorID	RC_AttachRobot_instance(Robot :=*parameter*, Execute :=*parameter*, Done =>*parameter*, Busy =>*parameter*, CommandAborted =>*parameter*, Error =>*parameter*, ErrorID =>*parameter*);

Variables

Input Variables

Input variable	Name	Data type	Valid range	Default	Description
Execute	Execute	BOOL	TRUE or FALSE	FALSE	The instruction is executed when Execute changes to TRUE.

Output Variables

Output variable	Name	Data type	Valid range	Description
Done	Done	BOOL	TRUE or FALSE	TRUE when the command from the instruction to the RC Function Module is completed. *1
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	*2	Contains the error code when an error occurs. A value of 16#0000 indicates normal execution.

*1. Check the instruction completion including the OMRON robot with *_RC_RBT[*].Status.Standby* robot variable because the RC Function Module sends a command to the OMRON robot after the instruction sends a command to the RC Function Module. Do not check the instruction completion with *Done* output variable.

*2. The lower four digits of the event code give the error code for ErrorID. Refer to *11-3 Error Table* on page 11-8 for details.

Output Variable Update Timing

Output variable	Timing for changing to TRUE	Timing for changing to FALSE
Done	When the command from the instruction to the RC Function Module is completed.	• When *Execute* is TRUE and changes to FALSE. • After one period when *Execute* is FALSE.
Busy	When *Execute* changes to TRUE.	• When *Done* changes to TRUE. • When *CommandAborted* changes to TRUE. • When *Error* changes to TRUE.
CommandAborted	• When another instruction causes an error and aborts this instruction. • When the instruction is executed during *ErrorStop* (Error Deceleration Stopping) or *Stopping* (Robot Deceleration Stopping).	• When *Execute* is TRUE and changes to FALSE. • After one period when *Execute* is FALSE.
Error	When there is an error in the execution conditions or input variables for the instruction.	• When *Execute* is TRUE and changes to FALSE. • After one period when *Execute* is FALSE.

In-out Variables

In-out variable	Name	Data type	Valid range	Description
Robot	Robot	_sRC_RBT_REF	---	Specifies the OMRON robot.

Function

The Robot Integrated CPU Unit allows the sequence control program or a V+ program to control a robot.
Executing this instruction enables the sequence control program to control the specified OMRON robot.
This is called that the robot is attached.
If you use the RC_AttachRobot (Attach Robot) instruction, always check the completion of the instruction with *_RC_RBT[*].Status.Standby* robot variable.

If the program cannot attach the specified OMRON robot, the instruction ends in a failure and *Error* changes to TRUE.

The program cannot attach the robot in the following cases.
- The specified OMRON robot has already been attached by a sequence control program.
- The robot drive status of the specified OMRON robot is Joint Mode, World Mode, Tool Mode, or Manual Mode.
- The high power for the specified OMRON robot is not turned ON.
- The calibration of the specified OMRON robot is not completed.

If the program fails to attach the specified OMRON robot, the instruction is aborted and *CommandAborted* changes to TRUE.
The program fails to attach the robot in the following case.
- The specified OMRON robot is controlled from a V+ program.

If the program cannot attach or fails to attach the specified OMRON robot, eliminate the cause, reset the error, and then execute the instruction again.

To detach the OMRON robot that is already attached with this instruction, execute the RC_DetachRobot (Detach Robot) instruction. Then, the OMRON robot can be detached from the sequence control program.
Refer to *RC_DetachRobot* on page 9-13 for details.

Precautions for Correct Use

- After the robot was attached with this instruction, when the operating mode of the Robot Integrated CPU Unit is changed to PROGRAM mode, the robot is detached.
- If the robot drive status of the specified OMRON robot is Joint Mode, World Mode, Tool Mode, or Manual Mode, the robot is also detached.

RC_DetachRobot

The RC_DetachRobot instruction detaches the specified OMRON robot from the sequence control program and makes the robot controllable from other programs that are not the sequence control program.

Instruction	Name	FB/FUN	Graphic expression	ST expression
RC_DetachRobot	Detach Robot	FB	RC_DetachRobot_instance RC_DetachRobot Robot ---------------- Robot Execute Done Busy CommandAborted Error ErrorID	RC_DetachRobot_instance(Robot :=*parameter*, Execute :=*parameter*, Done =>*parameter*, Busy =>*parameter*, CommandAborted =>*parameter*, Error =>*parameter*, ErrorID =>*parameter*);

Variables

Input Variables

Input variable	Name	Data type	Valid range	Default	Description
Execute	Execute	BOOL	TRUE or FALSE	FALSE	The instruction is executed when Execute changes to TRUE.

Output Variables

Output variable	Name	Data type	Valid range	Description
Done	Done	BOOL	TRUE or FALSE	TRUE when the command from the instruction to the RC Function Module is completed. *1
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	*2	Contains the error code when an error occurs. A value of 16#0000 indicates normal execution.

*1. Check the instruction completion including the OMRON robot with *_RC_RBT[*].Status.Disabled* robot variable because the RC Function Module sends a command to the OMRON robot after the instruction sends a command to the RC Function Module. Do not check the instruction completion with *Done* output variable.

*2. The lower four digits of the event code give the error code for ErrorID. Refer to *11-3 Error Table* on page 11-8 for details.

● Output Variable Update Timing

Output variable	Timing for changing to TRUE	Timing for changing to FALSE
Done	When the command from the instruction to the RC Function Module is completed.	• When *Execute* is TRUE and changes to FALSE. • After one period when *Execute* is FALSE.
Busy	When *Execute* changes to TRUE.	• When *Done* changes to TRUE. • When *CommandAborted* changes to TRUE. • When *Error* changes to TRUE.
CommandAborted	When the instruction is executed during *ErrorStop* (Error Deceleration Stopping) or *Stopping* (Robot Deceleration Stopping).	• When *Execute* is TRUE and changes to FALSE. • After one period when *Execute* is FALSE.
Error	When there is an error in the execution conditions or input variables for the instruction.	• When *Execute* is TRUE and changes to FALSE. • After one period when *Execute* is FALSE.

In-out Variables

In-out variable	Name	Data type	Valid range	Description
Robot	Robot	_sRC_RBT_REF	---	Specifies the OMRON robot.

Function

The Robot Integrated CPU Unit allows the sequence control program or a V+ program to control a robot.
The RC_DetachRobot instruction detaches the specified OMRON robot from the sequence control program to make it usable from the V+ program or teaching pendant.
This is called that the robot is detached.
If you use the RC_DetachRobot (Detach Robot) instruction, always check the completion of the instruction with *_RC_RBT[*].Status.Disabled* robot variable.

RC_SetToolTransform

The RC_SetToolTransform instruction sets or releases the tool conversion coordinates for the OMRON robot.

Instruction	Name	FB/FUN	Graphic expression	ST expression
RC_SetToolTransform	Set Tool Conversion Coordinates	FB	RC_SetToolTransform_instance RC_SetToolTransform Robot ---------------- Robot Execute Done ToolCoordTransform Busy CommandAborted Error ErrorID	RC_SetToolTransform_instance(Robot := *parameter*, Execute := *parameter*, ToolCoordTransform := *parameter*, Done => *parameter*, Busy => *parameter*, CommandAborted => *parameter*, Error => *parameter*, ErrorID => *parameter*);

Variables

Input Variables

Input variable	Name	Data type	Valid range	Default	Description
Execute	Execute	BOOL	TRUE or FALSE	FALSE	The instruction is executed when Execute changes to TRUE.
ToolCoordTransform	Tool Conversion Coordinates	ARRAY[0..5] OF LREAL	Negative number, positive number, or 0	0	Specifies the offset value for the tip of the robot arm to the tool center point as a relative position (X, Y, Z, Yaw, Pitch, and Roll) to the world coordinate system. The default setting “0,0,0,0,0,0” specifies to cancel the tool offset.

Output Variables

Output variable	Name	Data type	Valid range	Description
Done	Done	BOOL	TRUE or FALSE	TRUE when the command from the instruction to the RC Function Module 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.

Output variable	Name	Data type	Valid range	Description
ErrorID	Error Code	WORD	*1	Contains the error code when an error occurs. A value of 16#0000 indicates normal execution.

*1. The lower four digits of the event code give the error code for ErrorID. Refer to *11-3 Error Table* on page 11-8 for details.

● Output Variable Update Timing

Output variable	Timing for changing to TRUE	Timing for changing to FALSE
Done	When the command from the instruction to the RC Function Module is completed.	• When *Execute* is TRUE and changes to FALSE. • After one period when *Execute* is FALSE.
Busy	When *Execute* changes to TRUE.	• When *Done* changes to TRUE. • When *CommandAborted* changes to TRUE. • When *Error* changes to TRUE.
CommandAborted	• When another instruction causes an error and aborts this instruction. • When the instruction is executed during *ErrorStop* (Error Deceleration Stopping) or *Stopping* (Robot Deceleration Stopping).	• When *Execute* is TRUE and changes to FALSE. • After one period when *Execute* is FALSE.
Error	When there is an error in the execution conditions or input variables for the instruction.	• When *Execute* is TRUE and changes to FALSE. • After one period when *Execute* is FALSE.

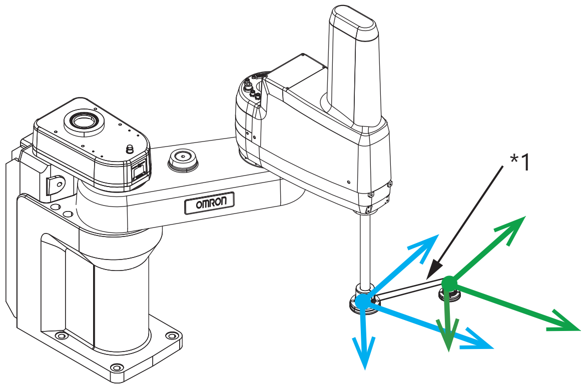
In-out Variables

In-out variable	Name	Data type	Valid range	Description
Robot	Robot	_sRC_RBT_REF	---	Specifies the OMRON robot.

Function

The RC_SetToolTransform instruction sets tool conversion coordinates for the specified OMRON robot.

- The instruction sets the tool conversion coordinates for the world coordinate system of the specified OMRON robot as a relative position (X, Y, Z, Yaw, Pitch, and Roll).
- After *Done* (Done) from this instruction changes to TRUE, the following robot control instructions operate in the tool conversion coordinates that you set.
  a) RC_MoveDirect (Robot Joint Interpolation) instruction
  b) RC_MoveLinear (Robot Linear Interpolation) instruction



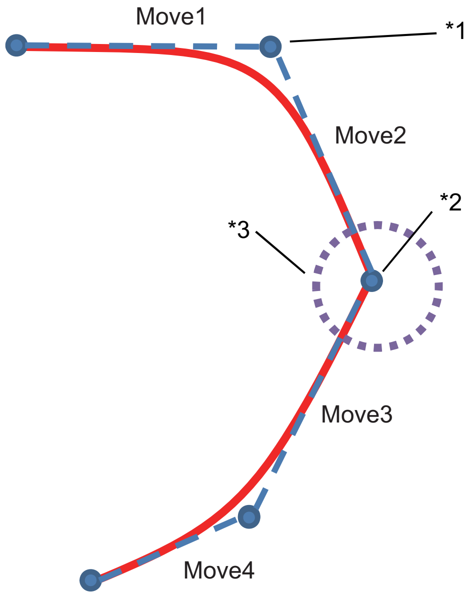
*1. If the tool is mounted to the tip of the robot arm, when you want to command the position of the tool center point as coordinates, set the tool conversion coordinates.
For the tool conversion coordinates, set the relative position from the tip of the robot arm.

When the instruction is executed while the OMRON robot is moving, the tool coordinate system is set after the current instruction is completed.

A instruction to operate the OMRON robot can perform multi-execution for up to 8 instruction instances.
When this instruction is executed during multi-execution of instructions, if the Continuous-path motion ON or OFF is set in the current instruction to operate the OMRON robot, the instruction is executed as described below.

- If the Continuous-path motion ON is set in the current instruction to operate the OMRON robot, the next instruction to operate the OMRON robot is completed, and then this instruction is executed.
- If the Continuous-path motion OFF is set in the current instruction to operate the OMRON robot, the current instruction is completed, and then this instruction is executed.

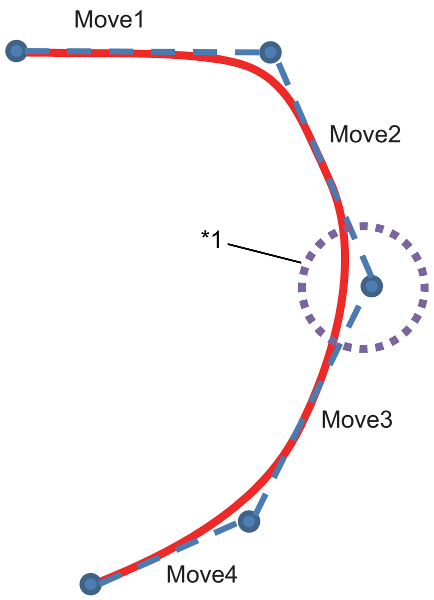
When the instruction is executed during execution of an instruction that specifies a continuous-path motion ON, a continuous-path motion does not take place.
For example, if the multi-execution of instructions is used for the instruction with Continuous-path motion ON, the current instruction with Continuous-path motion ON does not change the continuous-path motion before execution of this instruction.
If the RC_SetToolTransform (Set Tool Conversion Coordinates) instruction is executed while Move1 is operating, the following operations are performed.
The solid line in the figure represents a path when the Continuous-path motion ON is set, and the dotted line represents a path when the Continuous-path motion OFF is set.



*1. If the Continuous-path motion OFF is set, it is the timing to execute the RC_SetToolTransform (Set Tool Conversion Coordinates) instruction.
*2. If the Continuous-path motion ON is set, it is the timing to execute the RC_SetToolTransform (Set Tool Conversion Coordinates) instruction.
*3. Even if the Continuous-path motion ON is set, the path between Move2 and Move3 does not become a continuous-path motion.

Refer to *BufferMode is Set to Continuous-path Motion OFF* on page 6-3 for information on the motion when the continuous-path motion does not take place.

If the RC_SetToolTransform (Set Tool Conversion Coordinates) instruction is not executed with the same operation, the following operations are performed.



*1. If the Continuous-path motion ON is set, the path between Move2 and Move3 becomes a continuous-path motion.

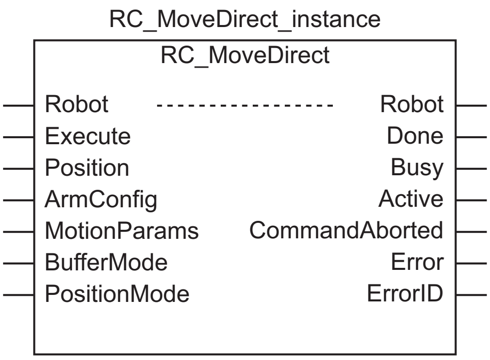
Precautions for Correct Use

In the following case, executing the instruction causes an error.

- The specified OMRON robot is not attached with the RC_AttachRobot (Attach Robot) instruction.
- The specified OMRON robot is Manual mode.

RC_MoveDirect

The RC_MoveDirect instruction enables joint interpolation operation of the OMRON robot.

Instruction	Name	FB/FUN	Graphic expression	ST expression
RC_MoveDirect	Robot Joint Interpolation	FB		RC_MoveDirect_instance(Robot :=*parameter*, Execute :=*parameter*, Position :=*parameter*, ArmConfig :=*parameter*, MotionParams :=*parameter*, BufferMode :=*parameter*, PositionMode :=*parameter*, Done =>*parameter*, Busy =>*parameter*, Active =>*parameter*, CommandAborted =>*parameter*, Error =>*parameter*, ErrorID =>*parameter*);

Variables

Input Variables

Input variable	Name	Data type	Valid range	Default	Description
Execute	Execute	BOOL	TRUE or FALSE	FALSE	The instruction is executed when Execute changes to TRUE.
Position	Target Position	ARRAY[0..5] OF LREAL	Negative number, positive number, or 0	0	Specifies the target position of joint interpolation in the OMRON robot's world coordinate system.
ArmConfig	Arm Configuration	_sRC_ARM_CONFIG	---	---	Specifies various parameters for configuration of the robot arm.
MotionParams	Motion Parameters	_sRC_JOINT_MOTION_REF	---	---	Specifies various parameters for the motion velocity of the OMRON robot.
BufferMode	Buffer Mode Selection	_eRC_BUFFER_MODE	1: _rcBuffered 2: _rcBlending	1	Specifies a continuous-path motion from the previous operation instruction. 1: Continuous-path motion OFF 2: Continuous-path motion ON
PositionMode	Target Position Specification Method	_eRC_POSITION_MODE	0: _rcAbsolute	0	Selects the target position specification method. 0: Absolute position

● Structure (_sRC_ARM_CONFIG Data Type)

Input variable	Name	Data type	Valid range	Default	Description
LeftyRighty	Lefty/Righty Setting	_eRC_LEFTY_RIGHTY	1: _rcLefty 2: _rcRighty	1	Specifies the LEFTY/RIGHTY setting of the robot arm. 1: LEFTY 2: RIGHTY
AboveBelow	Above/Below Setting	_eRC_ABOVE_BELOW	1: _rcAbove 2: _rcBelow	1	Specifies the ABOVE/BELOW setting of the robot arm. 1: ABOVE 2: BELOW
Flip	Flip Setting	_eRC_FLIP	1: _rcNoFlip 2: _rcFlip	1	Specifies the FLIP/NOFLIP setting of the robot arm. 1: NOFLIP 2: FLIP

● Structure (_sRC_JOINT_MOTION_REF Data Type)

Input variable	Name	Data type	Valid range	Default	Description
VelocityProfile	Velocity Profile	_eRC_VELOCITY_PROFILE	0: _rcTrapezoidal 1: _rcSCurve1 2: _rcSCurve2 3: _rcSCurve3 4: _rcSCurve4 5: _rcSCurve5 6: _rcSCurve6 7: _rcSCurve7 8: _rcSCurve8	0	Specifies the shape of the velocity profile. 0: Trapezoid (Parabolic/Constant deceleration rate) 1 to 8: Index 1 to 8 of the S-curve profile set in the Sysmac Studio*1
VelocityRatio	Velocity Ratio	REAL	0.000001 to 120*2	0*3	Specifies the operation velocity of each joint as a percentage of the maximum velocity. The unit is %.
Acceleration-Ratio	Acceleration Ratio	REAL	1 to 1000	0*3	Specifies the acceleration rate of each joint as a percentage of the maximum acceleration rate. The unit is %.
Deceleration-Ratio	Deceleration Ratio	REAL	1 to 1000	0*3	Specifies the deceleration rate of each joint as a percentage of the maximum deceleration rate. The unit is %.
NullingTolerance	Positioning Accuracy	_eRC_NULLING_TOLERANCE	1: _rcCoarseTolerance 2: _rcFineTolerance	2	Defines the zero tolerance applied to the end of positioning motion. For coarse positioning, the OMRON robot operation time is shorter. 1: Coarse positioning 2: High-accuracy positioning
SingleTurn	Rotation Limit	_eRC_SINGLE_TURN	1: _rcSingleTurn 2: _rcMultipleTurn	2	Specifies the restriction on the wrist joint angle of the OMRON robot. 1: Restricted to -180° to 180° 2: No angle restriction

*1. If you specify the S-curve profile that is not enabled in the Sysmac Studio, the robot moves as *0* is specified.
*2. You can set to *120*, however, the maximum velocity is *100*.
*3. Default value is *0*, however, *0* cannot be set. If this instruction is executed with the default value, an error will occur.

Refer to the OMRON robot manuals for information on the OMRON robot settings.

Output Variables

Output variable	Name	Data type	Valid range	Description
Done	Done	BOOL	TRUE or FALSE	TRUE when the target position is reached.
Busy	Executing	BOOL	TRUE or FALSE	TRUE when the instruction is acknowledged.
Active	Controlling	BOOL	TRUE or FALSE	TRUE while the axis is being controlled.
CommandAborted	Command Aborted	BOOL	TRUE or FALSE	TRUE when the instruction is aborted. TRUE if the coordinates that are out of the range of the OMRON robot motion are specified for the target position.
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. The lower four digits of the event code give the error code for ErrorID. Refer to *11-3 Error Table* on page 11-8 for details.

● Output Variable Update Timing

Output variable	Timing for changing to TRUE	Timing for changing to FALSE
Done	When the target position is reached.	• When *Execute* is TRUE and changes to FALSE. • After one period when *Execute* is FALSE.
Busy	When *Execute* changes to TRUE.	• When *Done* changes to TRUE. • When *CommandAborted* changes to TRUE. • When *Error* changes to TRUE.
Active	When the OMRON robot starts moving.	• When *Done* changes to TRUE. • When *CommandAborted* changes to TRUE. • When *Error* changes to TRUE.
CommandAborted	• When another instruction causes an error and aborts this instruction. • When the instruction is executed during *ErrorStop* (Error Deceleration Stopping) or *Stopping* (Robot Deceleration Stopping).	• When *Execute* is TRUE and changes to FALSE. • After one period when *Execute* is FALSE.
Error	When there is an error in the execution conditions or input variables for the instruction.	• When *Execute* is TRUE and changes to FALSE. • After one period when *Execute* is FALSE.

In-out Variables

In-out variable	Name	Data type	Valid range	Description
Robot	Robot	_sRC_RBT_REF	---	Specifies the OMRON robot.

Function

- The RC_MoveDirect instruction moves the tool center point of the specified OMRON robot to the position specified by *Position* (Target Position) in joint interpolation operation.
- The arm configuration when the OMRON robot has reached the target position can be specified.
- Parallel robots, such as iX3 and iX4, do not support joint interpolation operation with this instruction. If the instruction is used for a parallel robot, the robot performs linear interpolation operation.
- In the following cases, executing the instruction causes an error.
  a) The high power for the specified OMRON robot is not turned ON.
  b) The specified OMRON robot is not attached with the RC_AttachRobot (Attach Robot) instruction.
  c) The specified OMRON robot is Manual mode.

Target Position

Specify the position that the tool center point of the OMRON robot reaches, as the position and angle in the world coordinate system.
Whether elements are enabled or disabled and what they mean vary depending on the type of the target OMRON robot.

Variables	Valid range*1	Description	
		Parallel robot (iX3 or iX4) SCARA robot (eCobra or i4H)	Articulated robot (Viper)
Position[0]	Positive number, negative number, or 0	X-axis target position (mm)	
Position[1]	Positive number, negative number, or 0	Y-axis target position (mm)	
Position[2]	Positive number, negative number, or 0	Z-axis target position (mm)	
Position[3]	Positive number, negative number, or 0	Specify 0(deg).	Yaw (deg)
Position[4]	Positive number, negative number, or 0	Specify 180(deg).	Pitch (deg) *180* when the tool center point is facing downward (negative direction of the Z axis)
Position[5]	Positive number, negative number, or 0	End effector rotation (deg)	Roll (deg)

*1. Refer to the OMRON robot manuals for details on the actual travel range.
If the restricted range for the OMRON robot is exceeded, an error will occur at the time of which the OMRON robot is tried to travel over the restricted range.

Arm Configuration

Specify the posture of the robot arm when the tool center point of the OMRON robot has reached the target position.
Whether elements are enabled or disabled and what they mean vary depending on the type of the target OMRON robot.

Name	Meaning	Valid range	Description		
			Parallel robot (iX3 or iX4)	SCARA robot (eCobra or i4H)	Articulated robot (Viper)
LeftyRighty	Lefty/ Righty Setting	1: _rcLefty 2: _rcRighty	Disabled	Sets the OMRON robot so that the first two links operate like a human left arm (**1: _rcLefty**) or right arm (**2: _rcRighty**). [*1]	
AboveBelow	Above/ Below Setting	1: _rcAbove 2: _rcBelow	Disabled		Sets the OMRON robot so that the elbow joint is above (**1: _rcAbove**) or below (**2: _rcBelow**) the line that connects the shoulder and the wrist. [*2]
Flip	Flip Setting	1: _rcNoFlip 2: _rcFlip	Disabled		Sets the OMRON robot so that the pitch angle of the wrist is indicated as a positive value (**1: _rcNoFlip**) or a negative value (**2: _rcFlip**). [*3]

*1. Current setting can be checked with _RC_RBT[*].DrvStatus.Righty.
*2. Current setting can be checked with _RC_RBT[*].DrvStatus.Below.
*3. Current setting can be checked with _RC_RBT[*].DrvStatus.Flip.

Motion Parameters

Specify the acceleration/deceleration and end conditions for the tool center point of the OMRON robot that moves to the target position in the path.

Name	Meaning	Valid range	Description
VelocityProfile	Velocity Profile	0: _rcTrapezoidal 1: _rcSCurve1 2: _rcSCurve2 3: _rcSCurve3 4: _rcSCurve4 5: _rcSCurve5 6: _rcSCurve6 7: _rcSCurve7 8: _rcSCurve8	Specifies the shape of the velocity profile. When **0: _rcTrapezoidal** is specified, the robot accelerates/decelerates at a constant acceleration/deceleration rate and therefore the velocity curve has a trapezoidal shape. When **1 to 8** is specified, the increase or decrease time of acceleration or deceleration rate during acceleration or deceleration is applied so that the acceleration/deceleration rate curve has a trapezoidal shape according to the S-curve profile set in the Sysmac Studio. The specified value **1 to 8** corresponds to Index 1 to 8 of the S-curve profile. [*1]
VelocityRatio	Velocity Ratio	0.000001 to 120[*2]	Sets the target velocity at each joint during movement to the target position, which is the maximum velocity (full speed) set for each joint multiplied by the value specified here. The unit is %. [*3]
AccelerationRatio	Acceleration Ratio	1 to 1,000	Sets the maximum acceleration rate at each joint during acceleration to the target velocity, which is the maximum acceleration (full acceleration rate) set for each joint multiplied by the value specified here. The unit is %. [*3]
DecelerationRatio	Deceleration Ratio	1 to 1,000	Sets the maximum deceleration rate at each joint during deceleration to the target position, which is the maximum deceleration (full deceleration rate) set for each joint multiplied by the value specified here. The unit is %. [*3]

Name	Meaning	Valid range	Description
NullingTolerance	Position-ing Accu-racy	1: _rcCoarseTolerance 2: _rcFineTolerance	Select positioning accuracy when the motion is completed. The setting is selected for each OMRON robot. When **1: _rcCoarseTolerance** is specified, the hardware servo of the OMRON robot is set to low accuracy. This setting reduces the time to completion of the operation in return for low accuracy. When **2: _rcFineTolerance** is specified, the hardware servo is set to high accuracy. This settings increases the time to completion of the operation in return for high accuracy.
SingleTurn	Rotation Limit	1: _rcSingleTurn 2: _rcMultipleTurn	When **1: _rcSingleTurn** is specified, the wrist joint angle of the OMRON robot is restricted in the -180° to 180° range and therefore the joint never reaches the limit stop. When **2: _rcMultipleTurn** is specified, the restricted range of the wrist joint angle is within the range of motion of the OMRON robot, so the joint moves the shortest distance. However, it may reach the limit stop in the next linear interpolation operation.

*1. If you specify the S-curve profile that is not enabled in the Sysmac Studio, the robot moves as *0* is specified.

*2. You can set to *120*, however, the maximum velocity is *100*.

*3. If this instruction is executed with the default value, an error will occur. Refer to the OMRON robot manuals for information on the OMRON robot settings.

Buffer Mode Selection

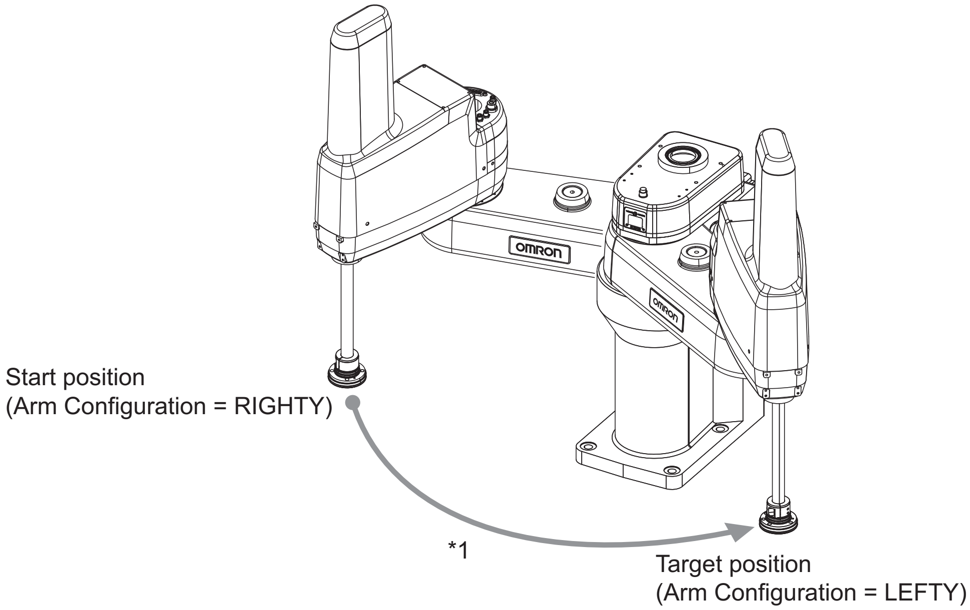
Specify how to continue from the previous motion when the instruction is executed while the OMRON robot is moving.

If **1: _rcBuffered** is specified for *BufferMode* (Buffer Mode Selection), the robot starts the motion executed by the instruction after the previous instruction reaches the target position. This motion is not a continuous-path motion.

If **2: _rcBlending** is specified, the robot starts the motion executed by the instruction in an overlapping manner before the previous instruction reaches the target position. This is called continuous-path motion.

Operation Example

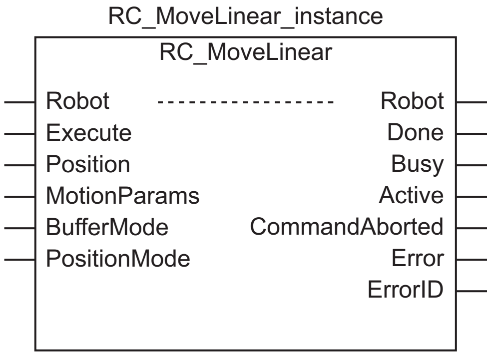
An example of operating a SCARA robot by changing the arm configuration is given below.



*1. The path is a curved line according to the joint structure. You cannot specify what kind of path the robot should follow.

RC_MoveLinear

The RC_MoveLinear instruction enables linear interpolation operation of the OMRON robot.

Instruction	Name	FB/FUN	Graphic expression	ST expression
RC_MoveLinear	Robot Linear Interpolation	FB		RC_MoveLinear_instance(Robot :=*parameter*, Execute :=*parameter*, Position :=*parameter*, MotionParams :=*parameter*, BufferMode :=*parameter*, PositionMode :=*parameter*, Done =>*parameter*, Busy =>*parameter*, Active =>*parameter*, CommandAborted =>*parameter*, Error =>*parameter*, ErrorID =>*parameter*);

Variables

Input Variables

Input variable	Name	Data type	Valid range	Default	Description
Execute	Execute	BOOL	TRUE or FALSE	FALSE	The instruction is executed when Execute changes to TRUE.
Position	Target Position	ARRAY[0..5] OF LREAL	Negative number, positive number, or 0	0	Specifies the target position of linear interpolation in the OMRON robot's world coordinate system.
MotionParams	Motion Parameters	_sRC_LINEAR_MOTION_REF	---	---	Specifies various parameters for the motion velocity of the OMRON robot.
BufferMode	Buffer Mode Selection	_eRC_BUFFER_MODE	1: _rcBuffered 2: _rcBlending	1	Specifies a continuous-path motion from the previous operation instruction. 1: Continuous-path motion OFF 2: Continuous-path motion ON
PositionMode	Target Position Specification Method	_eRC_POSITION_MODE	0: _rcAbsolute	0	Selects the target position specification method. 0: Absolute position

● Structure (_sRC_LINEAR_MOTION_REF Data Type)

Input variable	Name	Data type	Valid range	Default	Description
VelocityProfile	Velocity Profile	_eRC_VELOCITY_PROFILE	0: _rcTrapezoidal 1: _rcSCurve1 2: _rcSCurve2 3: _rcSCurve3 4: _rcSCurve4 5: _rcSCurve5 6: _rcSCurve6 7: _rcSCurve7 8: _rcSCurve8	0	Specifies the shape of the velocity profile. 0: Trapezoid (Parabolic/Constant deceleration rate) 1 to 8: Index 1 to 8 of the S-curve profile set in the Sysmac Studio*1
VelocityMode	Velocity Selection	_eRC_VELOCITY_MODE	0: _rcRatio 1: _rcMMPS	0	Selects the velocity specification method. When **0: _rcRatio** is specified, the velocity is specified as the maximum velocity multiplied by *VelocityRatio* (Velocity Ratio). When **1: _rcMMPS** is specified, the velocity that is specified in *Velocity* is used.
VelocityRatio	Velocity Ratio	REAL	0.000001 to 120*2	0*3	Applicable if **0: _rcRatio** is specified for *VelocityMode* (Velocity Selection). Specifies the travel velocity of TCP as a percentage of the maximum travel velocity in the cartesian coordinate system. The unit is %.
RotationVelocityRatio	Rotation Velocity Ratio	REAL	0.000001 to 120*2	0*3	Specifies the rotation velocity of the tool as a percentage of the maximum rotation velocity. The unit is %.
Velocity	Velocity	REAL	Positive number	0*3	Applicable if **1: _rcMMPS** is specified for *VelocityMode* (Velocity Selection). Specifies the travel velocity of TCP as an absolute value of the velocity in the cartesian coordinate system. The unit is mm/s.
AccelerationRatio	Acceleration Ratio	REAL	1 to 1,000	0*3	Specifies the acceleration rate of TCP as a percentage of the maximum acceleration rate. The unit is %.
DecelerationRatio	Deceleration Ratio	REAL	1 to 1,000	0*3	Specifies the deceleration rate of TCP as a percentage of the maximum deceleration rate. The unit is %.

Input variable	Name	Data type	Valid range	Default	Description
NullingToler-ance	Positioning Accuracy	_eRC_NUL-LING_TOL-ERANCE	1: _rcCoarseTolerance 2: _rcFineTolerance	2	Defines the zero tolerance applied to the end of positioning motion. For coarse positioning, the OM-RON robot operation time is shorter. 1: Coarse positioning 2: High-accuracy positioning
SingleTurn	Rotation Limit	_eRC_SIN-GLE_TURN	1: _rcSingleTurn 2: _rcMultipleTurn	2	Specifies the restriction on the wrist joint angle of the OMRON robot. 1: Restricted to -180° to 180° 2: No angle restriction

*1. If you specify the S-curve profile that is not enabled in the Sysmac Studio, the robot moves as *0* is specified.

*2. You can set to *120*, however, the maximum velocity is *100*.

*3. Default value is *0*, however, *0* cannot be set. If this instruction is executed with the default value, an error will occur. Refer to the OMRON robot manuals for information on the OMRON robot settings.

Output Variables

Output variable	Name	Data type	Valid range	Description
Done	Done	BOOL	TRUE or FALSE	TRUE when the target position is reached.
Busy	Executing	BOOL	TRUE or FALSE	TRUE when the instruction is acknowledged.
Active	Controlling	BOOL	TRUE or FALSE	TRUE while the axis is being controlled.
CommandAborted	Command Abort-ed	BOOL	TRUE or FALSE	TRUE when the instruction is aborted. TRUE if the coordinates that are out of the range of the OMRON robot motion are specified for the target position.
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. The lower four digits of the event code give the error code for ErrorID. Refer to *11-3 Error Table* on page 11-8 for details.

● Output Variable Update Timing

Output variable	Timing for changing to TRUE	Timing for changing to FALSE
Done	When the target position is reached.	• When *Execute* is TRUE and changes to FALSE. • After one period when *Execute* is FALSE.
Busy	When *Execute* changes to TRUE.	• When *Done* changes to TRUE. • When *CommandAborted* changes to TRUE. • When *Error* changes to TRUE.
Active	When the OMRON robot starts moving.	• When *Done* changes to TRUE. • When *CommandAborted* changes to TRUE. • When *Error* changes to TRUE.

Output variable	Timing for changing to TRUE	Timing for changing to FALSE
CommandAborted	• When another instruction causes an error and aborts this instruction. • When the instruction is executed during *ErrorStop* (Error Deceleration Stopping) or *Stopping* (Robot Deceleration Stopping).	• When *Execute* is TRUE and changes to FALSE. • After one period when *Execute* is FALSE.
Error	When there is an error in the execution conditions or input variables for the instruction.	• When *Execute* is TRUE and changes to FALSE. • After one period when *Execute* is FALSE.

In-out Variables

In-out variable	Name	Data type	Valid range	Description
Robot	Robot	_sRC_RBT_REF	---	Specifies the OMRON robot.

Function

- The RC_MoveLinear instruction moves the tool center point of the specified OMRON robot to the position specified by *Position* (Target Position) in linear interpolation.
  The tool center point moves in a linear path, whereas the tool rotates smoothly to the target angle.
- The arm configuration remains unchanged during this operation and maintains the status before execution of the instruction.
- In the following cases, executing the instruction causes an error.
  a) The high power for the specified OMRON robot is not turned ON.
  b) The specified OMRON robot is not attached with the RC_AttachRobot (Attach Robot) instruction.
  c) The specified OMRON robot is Manual mode.

Target Position

Specify the position that the tool center point of the OMRON robot reaches, as the position and angle in the world coordinate system.
Refer to *Target Position* on page 9-23 of the RC_MoveDirect (Robot Joint Interpolation) instruction for details.

Motion Parameters

Specify the acceleration/deceleration and end conditions for TCP of the OMRON robot that moves to the target position in the path.
Motion parameters are almost the same as the RC_MoveDirect (Robot Joint Interpolation) instruction, except for the difference in velocity and acceleration/deceleration specification.

Name	Meaning	Valid range	Description
VelocityMode	Velocity Selection	0: _rcRatio 1: _rcMMPS	Selects the specification method for the travel velocity of TCP. When **0: _rcRatio** is specified, the velocity is specified as the maximum velocity multiplied by *VelocityRatio* (Velocity Ratio). When **1: _rcMMPS** is specified, the velocity that is specified in *Velocity* is used.
VelocityRatio	Velocity Ratio	0.000001 to 120*1	Applicable if **0: _rcRatio** is specified for *VelocityMode* (Velocity Selection). Specifies the travel velocity of TCP as a percentage of the maximum travel velocity in the cartesian coordinate system. The unit is %. *2
RotationVelocityRatio	Rotation Velocity Ratio	0.000001 to 120*1	Specifies the rotation velocity of the tool as a percentage of the maximum rotation velocity in the cartesian coordinate system. The unit is %. *2
Velocity	Velocity	Positive number	Applicable if **1: _rcMMPS** is specified for *VelocityMode* (Velocity Selection). Specifies the travel velocity of TCP as an absolute value of the velocity in the cartesian coordinate system. The unit is mm/s. *2
AccelerationRatio	Acceleration Ratio	1 to 1,000	Specifies the acceleration rate of TCP in the cartesian coordinate system as a percentage of the maximum travel acceleration rate and a percentage of the maximum rotational acceleration rate. The unit is %. *2 The same percentage is applied for travel and rotation.
DecelerationRatio	Deceleration Ratio	1 to 1,000	Specifies the deceleration rate of TCP in the cartesian coordinate system as a percentage of the maximum travel deceleration rate and a percentage of the maximum rotational deceleration rate. The unit is %. *2 The same percentage is applied for travel and rotation.

*1. You can set to *120*, however, the maximum velocity is *100*.

*2. If this instruction is executed with the default value, an error will occur. Refer to the OMRON robot manuals for information on the OMRON robot settings.

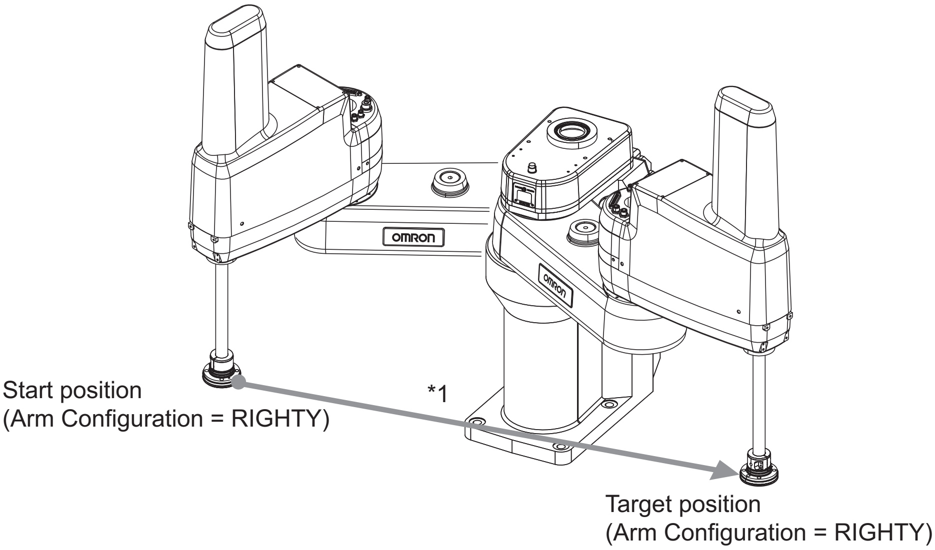
Buffer Mode Selection

Specify how to continue from the previous motion when the instruction is executed while the OMRON robot is moving.

Refer to *Buffer Mode Selection* on page 9-25 of the RC_MoveDirect (Robot Joint Interpolation) instruction for details.

Operation Example

An example of operating a SCARA robot using linear interpolation is given below.



*1. The path is a straight line in a Cartesian space. The arm configuration cannot be changed.

RC_SyncPathOffset

The RC_SyncPathOffset instruction periodically sends robot path target offset to the specified OMRON robot.

Instruction	Name	FB/FUN	Graphic expression	ST expression
RC_SyncPathOffset	Robot Path Compensation	FB	RC_SyncPathOffset_instance RC_SyncPathOffset Robot — Robot Execute — CompensationActive Offset — Busy CommandAborted Error ErrorID	RC_SyncPathOffset_instance(Robot :=*parameter*, Execute :=*parameter*, Offset :=*parameter*, CompensationActive =>*parameter*, Busy =>*parameter*, CommandAborted =>*parameter*, Error =>*parameter*, ErrorID =>*parameter*);

Precautions for Correct Use

If you use this instruction, make sure that the specified OMRON robot is attached from a V+ program.

Variables

Input Variables

Input variable	Name	Data type	Valid range	Default	Description
Execute	Execute	BOOL	TRUE or FALSE	FALSE	The instruction is executed when Execute changes to TRUE.
Offset	Target Offset	ARRAY[0..5] OF LREAL	Negative number, positive number, or 0	0	Sets the path target offset. [*1]

*1. Refer to *Target Offset* on page 9-36 for details.

Output Variables

Output variable	Name	Data type	Valid range	Description
CompensationActive	Compensating	BOOL	TRUE or FALSE	TRUE when the path compensation is active.
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.

Output variable	Name	Data type	Valid range	Description
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. The lower four digits of the event code give the error code for ErrorID. Refer to *11-3 Error Table* on page 11-8 for details.

● Output Variable Update Timing

Output variable	Timing for changing to TRUE	Timing for changing to FALSE
CompensationActive	• After one period when the instruction is acknowledged if the ALTER mode of the robot is enabled*1. • When the ALTER mode is enabled*1 during instruction execution if the ALTER mode of the robot is disabled.	• When *CommandAborted* changes to TRUE. • When *Error* changes to TRUE.
Busy	When *Execute* changes to TRUE.	• When *CommandAborted* changes to TRUE. • When *Error* changes to TRUE.
CommandAborted	• When the instruction is executed during *ErrorStop* (Error Deceleration Stopping). • When the ALTER mode is disabled in the V+ program.	• When *Execute* is TRUE and changes to FALSE. • After one period when *Execute* is FALSE.
Error	When there is an error in the execution conditions or input variables for the instruction.	When the error is cleared.

*1. The ALTER mode is enabled when the interpolation operation is executed at least once after the ALTON V+ keyword is executed. Refer to the *eV+3 Keyword Reference Manual (Cat. No. I652)* for details.

In-out Variables

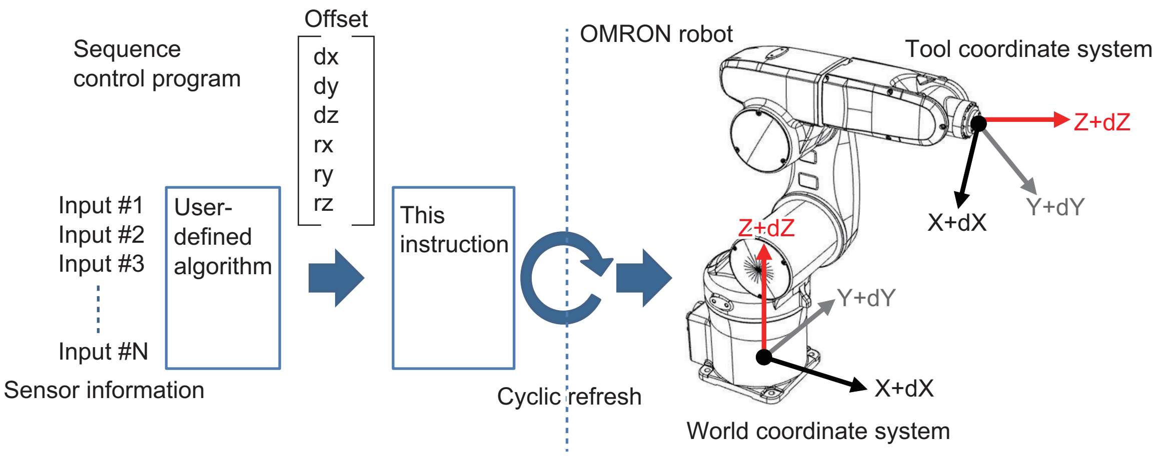
In-out Variables	Name	Data type	Valid range	Description
Robot	Robot	_sRC_RBT_REF	---	Specifies the OMRON robot.

Function

The RC_SyncPathOffset instruction periodically sends robot path target offset to the specified OMRON robot.
If you use this instruction, you need to attach the specified OMRON robot from a V+ program.

You can change the value of *Offset* (Target Offset) input variable while *Busy* (Executing) output variable is TRUE.
The *Offset* (Target Offset) is added to the path of OMRON robot while *CompensationActive* (Compensating) output variable is TRUE. In addition, the compensated value by this instruction is output to *TCPActPos* (Actual Position of TCP for OMRON Robot) member in the robot variable.

For example, you can use this instruction to establish a system that calculate a user-defined path compensation algorithm in the sequence control program and periodically compensate the OMRON robot path by utilizing the sensor information from a vision sensor or force sensor, etc.



In the following cases, executing the instruction causes an error. In addition, if the OMRON robot changes to the following states, the instruction is aborted.

- The robot drive status of the specified OMRON robot is Joint Mode, World Mode, Tool Mode, or Manual Mode.
- The high power for the specified OMRON robot is not turned ON.
- The calibration of the specified OMRON robot is not completed.
- The specified OMRON robot is attached from a sequence control program.
- The version of the specified OMRON robot does not support this function.

If *CommandAborted* or *Error* of this instruction changes to TRUE, the offset that was previously added to the the posture of OMRON robot is retained.

Operating Procedure for Robot Path Compensation

This instruction updates the path target offset as with the ALTER V+ keyword. Therefore, use this instruction in the same way as with the ALTER V+ keyword, that is, use the ALTON V+ keyword and enable path compensation from the sequence control program in advance.
Use the following procedures.

1. Execute the ALTON V+ keyword to the target OMRON robot in the V+ program.
2. Execute the interpolation operation for the target OMRON robot in the V+ program.
3. Execute this instruction from the sequence control program and compensate the path.
4. If you complete the path compensation, execute ALTOFF V+ keyword or DETACH V+ keyword to the target OMRON robot in the V+ program and abort this instruction.

Target Offset

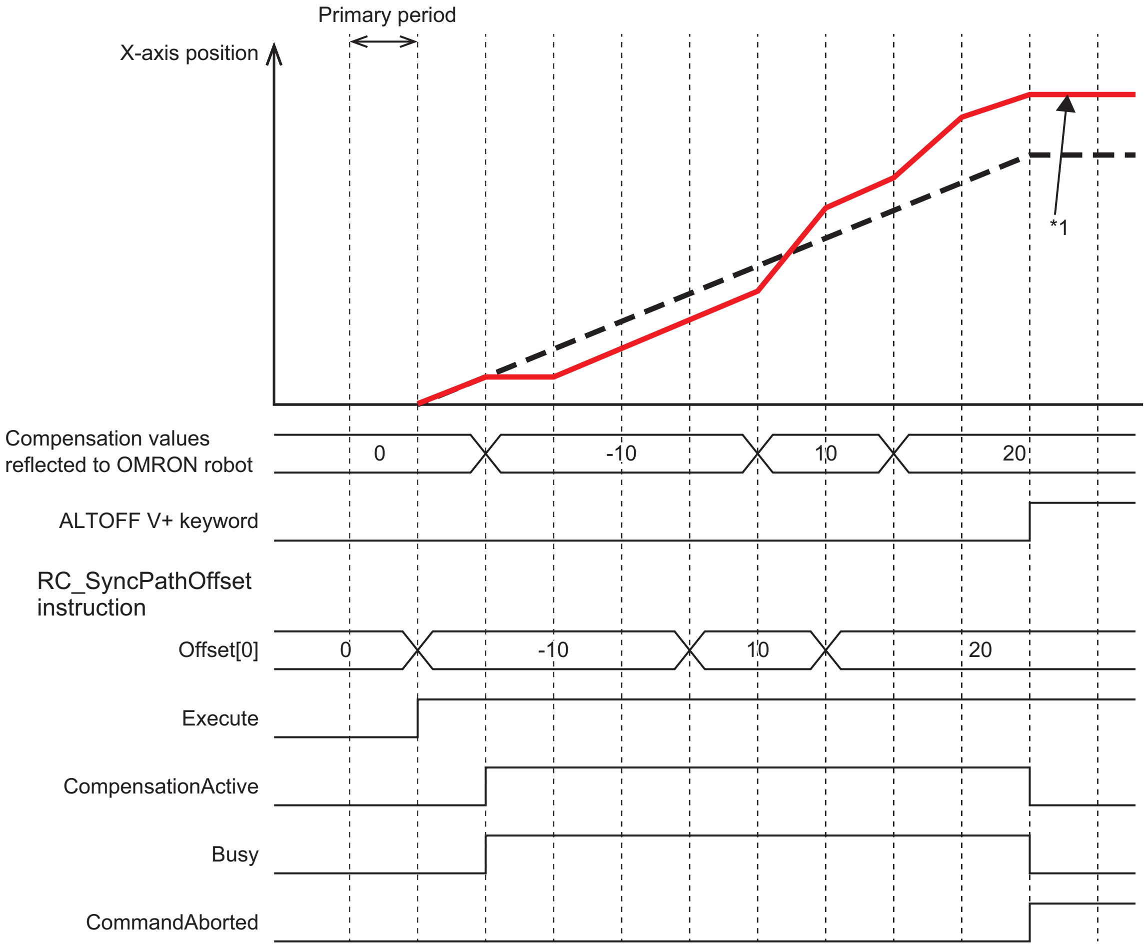
Specify the path target offset with position and angle.
It is added to the world coordinate system or tool coordinate system depending on the value of mode that is specified with the ALTON V+ keyword.

Name	Valid range*1	Description
Offset[0]	Positive number, negative number, or 0	X-axis target offset (mm)
Offset[1]	Positive number, negative number, or 0	Y-axis target offset (mm)
Offset[2]	Positive number, negative number, or 0	Z-axis target offset (mm)
Offset[3]	Positive number, negative number, or 0	X-axis target offset (deg)
Offset[4]	Positive number, negative number, or 0	Y-axis target offset (deg)
Offset[5]	Positive number, negative number, or 0	Z-axis target offset (deg)

*1. Refer to the OMRON robot manuals for details on the actual valid range. If the valid range for the OMRON robot is exceeded, an error will occur at the time of travel.

Timing Chart

As an example, a timing chart is shown below for the period from when this instruction is executed for the X-axis in execution of the liner interpolation (MOVES V+ keyword) to when the instruction is aborted with the ALTOFF V+ keyword.
The black dotted line represents before compensation and the red solid line represents after compensation.



*1. The OMRON robot stops at the position where the previous compensation value was added.

Sample Programming

The following programming example adds the target offset to Z-axis direction by using the RC_SyncPathOffset (Robot Path Compensation) instruction when the OMRON robot reciprocates between the loc.1 and the loc.p2.

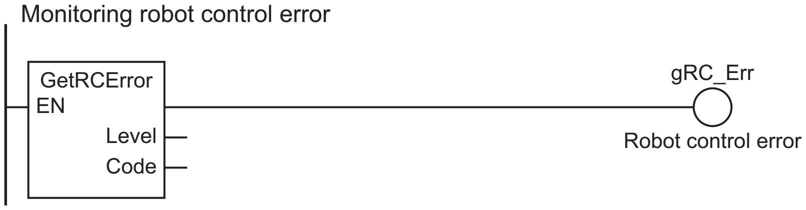
The motion completes after 10 rounds.

LD

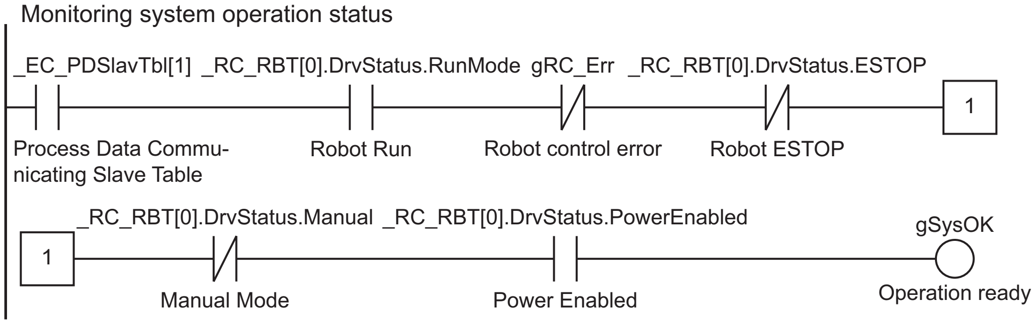
Internal variables	Variable	Data type	Initial value	Comment
	RC_ET1	RC_ExecVpPrgTask		
	RC_GT1	RC_GetVpPrgTaskStatus		
	RC_SP1	RC_SyncPathOffset		
	Offset	ARRAY[0..5] OF LREAL		Path target offset

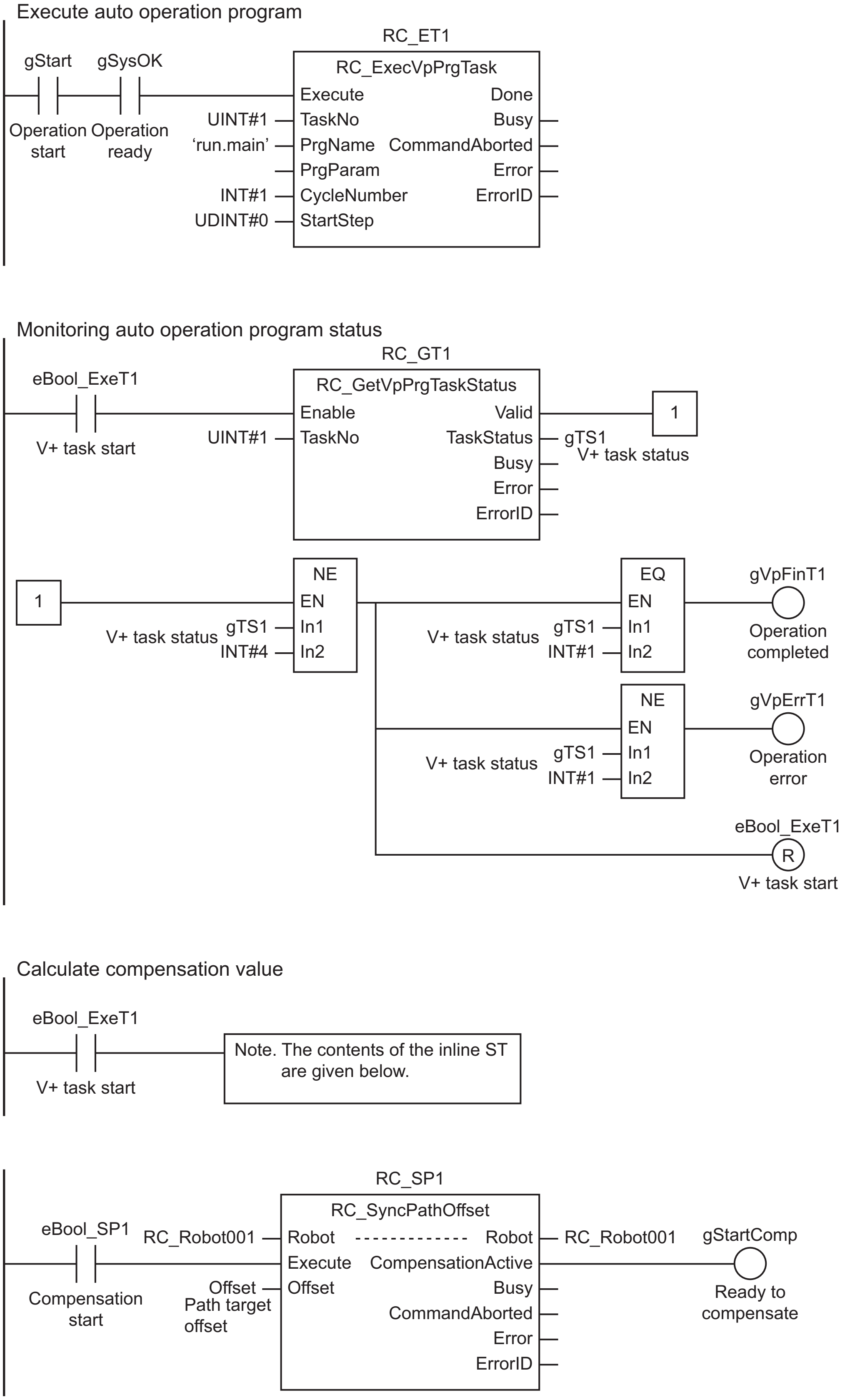
External variables	Variable	Data type	Constant	Comment
	_EC_PDSlavTbl	ARRAY[1..192] OF BOOL	☑	Process Data Communicating Slave Table
	_RC_RBT	ARRAY[0..7] OF _sRC_RBT_REF	☑	
	gRC_Err	BOOL	☐	Robot control error
	gSysOK	BOOL	☐	Operation ready
	gStart	BOOL	☐	Operation start
	gTS1	INT	☐	V+ task status
	gVpFinT1	BOOL	☐	Operation completed
	gVpErrT1	BOOL	☐	Operation error
	eBool_ExeT1	BOOL	☐	V+ task start
	RC_Robot001	_sRC_RBT_REF	☑	
	gStartComp	BOOL	☐	Ready to compensate
	eBool_SP1	BOOL	☐	Compensation start
	gRobo_State	INT	☐	Operation status

Monitoring robot control error



Monitoring system operation status





● Contents of Inline ST

```
// Calculate compensation values based on the information from sensor or robot.
CASE gRobo_State OF
```

```
        100:
            Offset[2]:=Offset[2]+0.1;
        200:
            Offset[2]:=Offset[2]-0.1;
        300:
            Offset[2]:=0.0;
    END_CASE;
```

V+ Program

```
.PROGRAM run.main()
;

        EXTERNAL ebool_exet1
        EXTERNAL ebool_sp1
        EXTERNAL gstartcomp
        EXTERNAL grobo_state
        AUTO cnt, loc.p1, loc.p2, loc.init


    ;set Target position
        SET loc.init = TRANS(400,0,350,0,180,0)
        SET loc.p1   = TRANS(400,-200,300,0,180,0)
        SET loc.p2   = TRANS(400, 200,300,0,180,0)

        cnt = 0
        ebool_exet1 = TRUE

        ENABLE POWER
        CALIBRATE
        ATTACH ()

        ;enable compensation from IEC program
        ALTON 6
        ebool_sp1 = TRUE

        ;move to initial position
        MOVES loc.init
        BREAK

        ;waiting for completion for preparation of RC_SyncPathOffset.
        DO
            WAIT.EVENT , 2E-03
        UNTIL gstartcomp == TRUE

        DO
            grobo_state = 100
```

```
            MOVES loc.p1
            BREAK

            grobo_state = 200
            MOVES loc.p2
            BREAK
            cnt = cnt + 1
        UNTIL cnt>10

        ;disable compensation from IEC program
        ALTOFF
        grobo_state = 300

        ebool_sp1 = FALSE
.END
```

RC_Stop

The RC_Stop instruction stops the current operation of the OMRON robot to cause a deceleration stop.

Instruction	Name	FB/FUN	Graphic expression	ST expression
RC_Stop	Stop Robot	FB	RC_Stop_instance RC_Stop Robot ----------------- Robot Execute Done Busy CommandAborted Error ErrorID	RC_Stop_instance(Robot :=*parameter*, Execute :=*parameter*, Done =>*parameter*, Busy =>*parameter*, CommandAborted =>*parameter*, Error =>*parameter*, ErrorID =>*parameter*);

Variables

Input Variables

Input variable	Name	Data type	Valid range	Default	Description
Execute	Execute	BOOL	TRUE or FALSE	FALSE	The instruction is executed when Execute changes to TRUE.

Output Variables

Output variable	Name	Data type	Valid range	Description
Done	Done	BOOL	TRUE or FALSE	TRUE when the command from the instruction to the RC Function Module 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. The lower four digits of the event code give the error code for ErrorID. Refer to *11-3 Error Table* on page 11-8 for details.

● Output Variable Update Timing

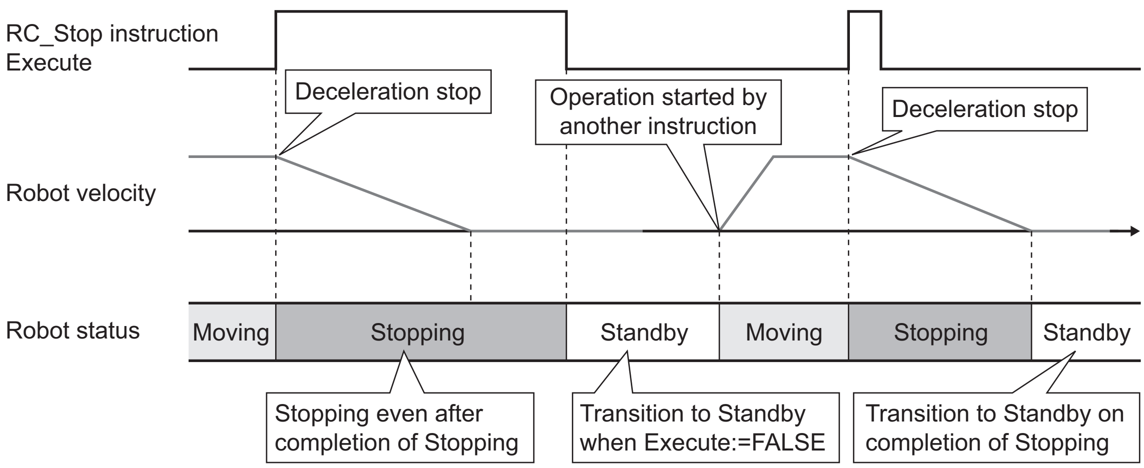
Output variable	Timing for changing to TRUE	Timing for changing to FALSE
Done	When the command from the instruction to the RC Function Module is completed.	• When *Execute* is TRUE and changes to FALSE. • After one period when *Execute* is FALSE.
Busy	When *Execute* changes to TRUE.	• When *Done* changes to TRUE. • When *CommandAborted* changes to TRUE. • When *Error* changes to TRUE.
CommandAborted	• When another instruction causes an error and aborts this instruction. • When the instruction is executed during *ErrorStop* (Error Deceleration Stopping) or *Stopping* (Robot Deceleration Stopping).	• When *Execute* is TRUE and changes to FALSE. • After one period when *Execute* is FALSE.
Error	When there is an error in the execution conditions or input variables for the instruction.	• When *Execute* is TRUE and changes to FALSE. • After one period when *Execute* is FALSE.

In-out Variables

In-out variable	Name	Data type	Valid range	Description
Robot	Robot	_sRC_RBT_REF	---	Specifies the OMRON robot.

Function

- The RC_Stop instruction stops the current operation of the specified OMRON robot to cause a deceleration stop.
  The instruction can be executed even if the OMRON robot is not moving.
- The OMRON robot status is Stopping if at least one of the following conditions is met: The OMRON robot is decelerating to a stop according to the instruction or *Execute* input variable to the instruction is TRUE.
  During the Stopping status, other operation commands are not acknowledged by the OMRON robot.



- In the following case, the operation is not stopped even if the instruction is executed, *CommandAborted* will change to TRUE.

RC_Stop

9

Function

a) The specified OMRON robot is not attached with the RC_AttachRobot (Attach Robot) instruction.
b) The specified OMRON robot is Manual mode.

RC_Reset

The RC_Reset instruction clears an OMRON robot error.

Instruction	Name	FB/FUN	Graphic expression	ST expression
RC_Reset	Reset Robot Error	FB	RC_Reset_instance RC_Reset Robot ---------------- Robot Execute Done Busy Failure Error ErrorID	RC_Reset_instance(Robot :=*parameter*, Execute :=*parameter*, Done =>*parameter*, Busy =>*parameter*, Failure =>*parameter*, Error =>*parameter*, ErrorID =>*parameter*);

Variables

Input Variables

Input variable	Name	Data type	Valid range	Default	Description
Execute	Execute	BOOL	TRUE or FALSE	FALSE	The instruction is executed when Execute changes to TRUE.

Output Variables

Output variable	Name	Data type	Valid range	Description
Done	Done	BOOL	TRUE or FALSE	TRUE when the command from the instruction to the RC Function Module is completed.
Busy	Executing	BOOL	TRUE or FALSE	TRUE when the instruction is acknowledged.
Failure	Failure End	BOOL	TRUE or FALSE	TRUE when the instruction was not executed correctly.
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. The lower four digits of the event code give the error code for ErrorID. Refer to *11-3 Error Table* on page 11-8 for details.

● Output Variable Update Timing

Output variable	Timing for changing to TRUE	Timing for changing to FALSE
Done	When the command from the instruction to the RC Function Module is completed.	• When *Execute* is TRUE and changes to FALSE. • After one period when *Execute* is FALSE.
Busy	When *Execute* changes to TRUE.	• When *Done* changes to TRUE. • When *Error* changes to TRUE.
Failure	• When an error reset is executed while deceleration stop or high power OFF is processed for an OMRON robot error. • When an error reset is executed during a robot error caused by a robot control common error .	• When *Execute* is TRUE and changes to FALSE. • After one period when *Execute* is FALSE.
Error	When there is an error in the execution conditions or input variables for the instruction.	• When *Execute* is TRUE and changes to FALSE. • After one period when *Execute* is FALSE.

In-out Variables

In-out variable	Name	Data type	Valid range	Description
Robot	Robot	_sRC_RBT_REF	---	Specifies the OMRON robot for which to clear the error.

Function

- The RC_Reset instruction clears an error for which "robot" is given for source details of the event occurred in the specified OMRON robot.
- The instruction can be executed even if there is no error in the OMRON robot. In this case, the instruction is completed without changing the OMRON robot status.
  When there is no error in the OMRON robot, no status change occurs even when the V+ program is aborted (with the ProgramError or Pause status).
- If this instruction is executed while deceleration stop or high power OFF is processed for an OMRON robot error, *Failure* (Failure End) will change to TRUE and the error will not be cleared.
  An OMRON robot error that results from an error for which "robot control common" is given for source details of the event cannot be cleared with this instruction, so *Failure* will also change to TRUE and the error will not be cleared.
- Network errors, such as EtherCAT Slave Communications Error, are not cleared by executing this instruction.
  To clear network errors, execute the ResetECError (Reset EtherCAT Error) instruction.
  Refer to the *NJ/NX-series Instructions Reference Manual (Cat. No. W502)* for information on the ResetECError (Reset EtherCAT Error) instruction.

Additional Information

When you reset errors common to the robot control and errors that occurred in all OMRON robots and V+ programs, refer to *ResetRCError* on page 10-2 for details.

10

System Control Instructions

This section describes the system control instructions.

ResetRCError

The ResetRCError instruction resets current Controller errors in the Robot Control Function Module.

Instruction	Name	FB/FUN	Graphic expression	ST expression
ResetRCError	Reset Robot Control Error	FB	ResetRCError_instance ResetRCError Execute — Done Busy Failure Error ErrorID	ResetRCError_instance(Execute :=*parameter*, Done =>*parameter*, Busy =>*parameter*, Failure =>*parameter*, Error =>*parameter*, ErrorID =>*parameter*);

Variables

Input Variables

Input variable	Name	Data type	Valid range	Default	Description
Execute	Execute	BOOL	TRUE or FALSE	FALSE	The instruction is executed when Execute changes to TRUE.

Output Variables

Output variable	Name	Data type	Valid range	Description
Done	Done	BOOL	TRUE or FALSE	TRUE when the command from the instruction to the RC Function Module is completed.
Busy	Executing	BOOL	TRUE or FALSE	TRUE when the instruction is acknowledged.
Failure	Failure End	BOOL	TRUE or FALSE	TRUE when the instruction was not executed correctly.
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. The lower four digits of the event code give the error code for ErrorID. Refer to *11-3 Error Table* on page 11-8 for details.

● Output Variable Update Timing

Output variable	Timing for changing to TRUE	Timing for changing to FALSE
Done	When the command from the instruction to the RC Function Module is completed.	• When *Execute* is TRUE and changes to FALSE. • After one period when *Execute* is FALSE.

Output variable	Timing for changing to TRUE	Timing for changing to FALSE
Busy	When *Execute* changes to TRUE.	• When *Done* changes to TRUE. • When *Error* changes to TRUE.
Failure	• When the errors are not reset. • When another instruction causes an error and aborts this instruction. • When the instruction is executed during *ErrorStop* (Error Deceleration Stopping) or *Stopping* (Robot Deceleration Stopping).	• When *Execute* is TRUE and changes to FALSE. • After one period when *Execute* is FALSE.
Error	When there is an error in the execution conditions or input variables for the instruction.	• When *Execute* is TRUE and changes to FALSE. • After one period when *Execute* is FALSE.

Function

- The ResetRCError instruction resets errors common to the robot control and errors that occurred in all OMRON robots and V+ programs. This allows you to execute a sequence control program or V+ program.
- A current error can be reset by this instruction if "Recovery" is "Error reset" in the error description for the error.
  In the following cases, *Failure* (Failure End) output variable will change to TRUE and an error will not be reset.
  a) When "Recovery" is "Cycle the power supply" or "Reset the Controller", which means that error reset is not possible
  b) When this instruction is executed while deceleration stop or high power OFF is processed for an OMRON robot error
  For details on events, refer to *11-4-2 Error Descriptions* on page 11-22.

Additional Information

When you reset errors for the individual OMRON robot, refer to *RC_Reset* on page 9-45 for details.

GetRCError

The GetRCError instruction gets the highest event level (partial fault or minor fault) and highest level event code of the current Controller errors in the Robot Control Function Module.

Instruction	Name	FB/FUN	Graphic expression	ST expression
GetRCError	Get Robot Control Error Status	FUN	(@)GetRCError EN — Out Level Code	Out: =GetRCError(Level =>*parameter*, Code =>*parameter*);

Variables

Output Variables

Output variable	Name	Data type	Valid range	Description
Out	Error Flag	BOOL	TRUE or FALSE	TRUE: Controller error FALSE: No Controller error
Level	Highest Level Status	UINT	0, 2, or 3	Highest event level of all current Controller errors in the RC Function Module 0: No Controller error 2: Partial fault level 3: Minor fault level
Code	Highest Level Event Code	DWORD	16#00000000, or 16#00070000 to 16#FFFFFFFF	Highest level event code of all current Controller errors in the RC Function Module 16#0000_0000: No Controller error 16#0007_0000 to 16#FFFF_FFFF: Event code

Function

- The GetRCError instruction gets *Level* (Highest Level Status) and *Code* (Highest Level Event Code) of current Controller errors in the RC Function Module.
- If there are no Controller errors, the value of *Out* (Error Flag) is FALSE.
- If there is more than one Controller error, the value of *Code* (Highest Level Event Code) is the event code for the Controller error that occurred first.

11

Troubleshooting

This section provides details of the errors (events) that may occur in the Robot Control Function Module, including how to troubleshoot them.

11-1 Errors

This section describes errors related to the RC Function Module.

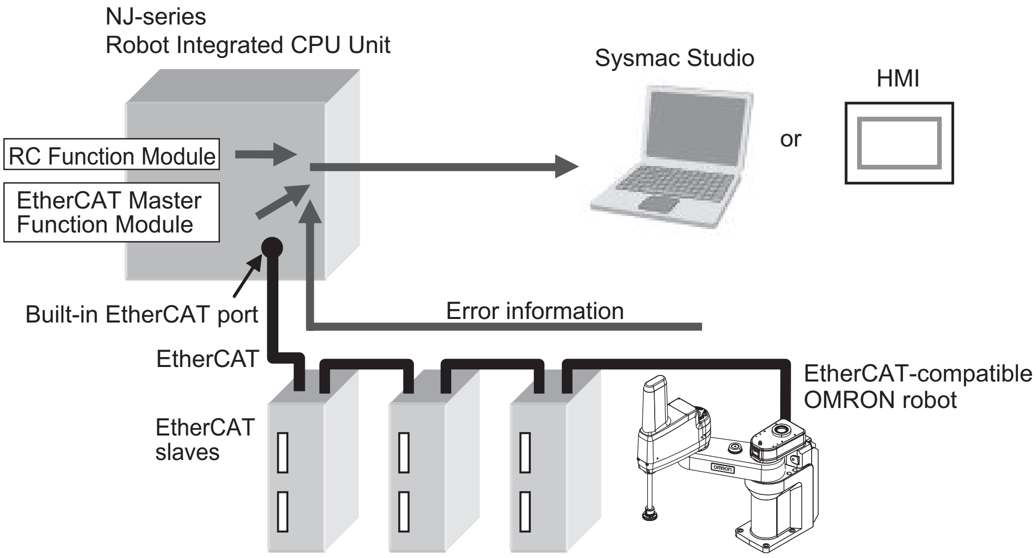
Additional Information

Refer to the *NJ/NX-series Troubleshooting Manual (Cat. No. W503)* for details on how to identify and reset errors in the NJ-series Controllers.

11-1-1 Sources of Errors Related to the Robot Control Function Module

Some errors may occur inside the RC Function Module, and others may occur due to some problem with EtherCAT communications, which are used for connection to OMRON robots and Servo Drives.

- Inside the RC Function Module
- EtherCAT Master Function Module
- Built-in EtherCAT communications port (hardware)
- EtherCAT slaves
- EtherCAT-compatible OMRON robots



You can identify the source and cause of an error by checking system-defined variables or using the Sysmac Studio or an HMI.

Precautions for Correct Use

Refer to the appendices of the *NJ/NX-series Troubleshooting Manual (Cat. No. W503)* for the applicable range of the HMI Troubleshooter.

11-1-2 Error Sources

RC Function Module errors can be categorized into the following two types according to the error source.

Error source	Description
Robot Control Common Error	Errors in the RC Function Module or errors whose sources cannot be identified • Initialization errors, control period exceeded errors, etc. • Errors that occur when the maximum robot number is exceeded by a robot number specified for execution of a robot control instruction • Errors that occur while V+ program is being executed
Robot Error	Errors that occur when the sequence control program executes an instruction to control OMRON robots

Additional Information

While **Use** is selected for **Event Log Settings** - **Instruction Error Output**, any errors related to robot control instructions will show "PLC Function Module" for the error source and "Instruction" for the source detail if detected.
For details on **Event Log Settings**, refer to the *NJ/NX-series CPU Unit Software User's Manual (Cat. No. W501)*.

11-1-3 Error Levels

The following table describes each error level.

Error level	Operation
Major fault	All control operations of the NJ-series Controller stop for a major fault level error.
Partial fault	All control operations of a certain function module in the NJ-series Controller stop for a partial fault level error. If a partial fault level error occurs in the RC Function Module, all functions of the RC Function Module, including robot operations, stop.
Minor fault	Some control operations of a certain function module in the NJ-series Controller stop for a minor fault level error. If a minor fault level error occurs in the RC Function Module, the relevant robots stop.
Observation	Observation level errors do not affect NJ-series Controller control operations. They are indicated to report potential problems before they develop into a minor fault level error or worse.
Information	Users are provided with information that do not indicate errors. In case of the V+ Program Error (96040000 hex), the robot that is controlled with the V+ program stops. Otherwise, the operation continues.

11-1-4 Errors Related to EtherCAT Communications and EtherCAT Slaves

The following RC Function Module errors can occur due to errors in EtherCAT communications or EtherCAT slaves.

Error name	Event code	Cause	Operation for error
EtherCAT Slave Disconnection Error	75020000 hex	One of the following occurred for the EtherCAT slave that is allocated to a robot. • Disconnection or replacement • Disablement	The OMRON robot where the error occurred stops. *1

Error name	Event code	Cause	Operation for error
EtherCAT Slave Communications Error	85800000 hex	A communications error occurred for the EtherCAT slave that is allocated to a robot.	The OMRON robot where the error occurred stops and does not acknowledge any operation other than error reset. If a V+ program is used to control the robot, the V+ program stops.

*1. Refer to *EtherCAT Slave Disconnection Error (page 11-45)* for the recovery.

11-1-5 OMRON Robot Events

Events which may occur during control operation for OMRON robots will have different event levels and event sources depending on how the control program is executed for the OMRON robots. You need to understand the differences before you modify the control program and design operations to respond to errors.

Controlling OMRON Robots with Sequence Control Program

If an error for which robot control cannot be continued occurred, the OMRON robot is stopped, but the sequence control program execution continues.
Consider the operation for errors including peripheral devices and write the user program with system-defined variables for robot control or output variables from the robot control instructions.
In addition, the above error is registered as a minor fault level event, so you can check it as one of the Controller errors on the Sysmac Studio or HMI. You can also check event logs for minor fault level events.
To recover operation from an error status, it is necessary to reset the error from the V+ program, sequence control program, or user program.

Event level	Event name	Error source
Minor fault	Robot Error Occurred	Robot

Controlling OMRON Robots with V+ Programs

If an error for which robot control cannot be continued occurred, the OMRON robot and the V+ program execution are stopped.
Consider the operation for errors including peripheral devices and write the user program with REACTE/ RETURNE V+ keyword or the RC_GetVpPrgTaskStatus (Get V+ Task Status) instruction.
In addition, the above error is registered as an information level event, so you cannot check it as one of the Controller errors on the Sysmac Studio or HMI.
You can check event logs for the information level events.
If you check the error as one of the Controller errors on the Sysmac Studio or HMI, write the user program so that a user-defined error is generated with the SetAlarm (Create User-defined Error) instruction.
To recover operation from an error status, you need to restart the V+task from the V+ program or sequence control program.
If a user-defined error occurred, it is necessary to reset the error from the V+ program, sequence control program, or user program.

Refer to the *eV+3 Keyword Reference Manual (Cat. No. I652)* for information on the REACTE/ RE-TURNE V+ keyword.
Refer to the *NJ/NX-series Instructions Reference Manual (Cat. No. W502)* for information on the SetAlarm (Create User-defined Error) instruction.

Event level	Event name	Error source
Information	V+ Program Error	Robot control common
	V+ Program Warning	
	V+ Program Information	

11-2 Identifying and Resetting Errors

This section describes how to identify and reset errors in the RC Function Module.
Errors in the RC Function Module will remain until they are reset.
To reset a Controller error, it is necessary to eliminate the cause of the error. The same error will occur again if you just reset it without eliminating the cause.

11-2-1 How to Check for Errors

You can use the following to check for errors in the RC Function Module.
- LEDs on the Controller Unit
- Troubleshooting functions of the Sysmac Studio
- Troubleshooter on an HMI
- Instructions for reading error status
- System-defined variables

Refer to the *NJ/NX-series Troubleshooting Manual (Cat. No. W503)* for details on how to check for errors using the LEDs on a Controller Unit, the troubleshooting functions of the Sysmac Studio, or the Troubleshooter on an HMI.

Instruction to Get Current Error Status in the Robot Control Function Module

You can use the following instruction to obtain information on current errors (events) in the RC Function Module. Refer to the description of the instruction for details.

Instruction name	Instruction	Function	Reference
Get Robot Control Error Status	GetRCError	Gets the highest event level (partial fault or minor fault) and highest level event code of the current Controller errors in the RC Function Module.	page 10-4

System-defined Variables Related to Current Error Status in the Robot Control Function Module

You can use the following system-defined variables to obtain information on current errors (events) in the RC Function Module. Refer to the description of each variable for details.

● Error Status Variables

The following table describes error status variables for the RC Function Module.

Name	Variable	Function
Robot Control Error Status	_RC_ErrSta	Gives the collective error status of all error status for the RC Function Module.
Robot Control Common Error Status	_RC_ComErrSta	Gives the collective error status of all errors that occur for common processing in the RC Function Module.
Robot Error Status	_RC_RBT_ErrSta	Gives the collective error status of all error status for each OMRON robot.

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

Bit	Name	Description	Value	Meaning
15	Master Detection*1	Indicates whether the master detected an error in the slaves that it manages.	TRUE	Error
			FALSE	No error
14	Slave Summary*2	Gives the collective error status of EtherCAT slaves that are assigned to OMRON robots in the RC Function Module.	TRUE	Error
			FALSE	No error
13 to 8	Reserved			
7	Major Fault	Indicates if there is a major fault level error.	TRUE	Error
			FALSE	No error
6	Partial Fault	Indicates if there is a partial fault level error.	TRUE	Error
			FALSE	No error
5	Minor Fault	Indicates if there is a minor fault level error.	TRUE	Error
			FALSE	No error
4	Observation	Indicates if there is an observation level error.	TRUE	Error
			FALSE	No error
3 to 0	Reserved			

*1. This bit is not used in Robot Control Error Status, Robot Control Common Error Status, or Robot Error Status for the RC Function Module.

*2. For the RC Function Module, this bit is used only in _RC_ErrSta (Robot Control Error Status).

● System-defined Variables for Robot Control

You can monitor the Robot Control Common Variable and Robot Variables of the system-defined variables for robot control to see if any errors have occurred in the RC Function Module.

Refer to the *7-1 System-defined variables for Robot Control* on page 7-2 for information on system-defined variables for robot control.

11-2-2 How to Reset Errors

You can use the following to reset errors in the RC Function Module.

- Commands from the Sysmac Studio
- Commands from an HMI
- Execution of ResetRCError (Reset Robot Control Error) or RC_Reset (Reset Robot Error)

Refer to the *NJ/NX-series Troubleshooting Manual (Cat. No. W503)* for details on how to reset errors from the Sysmac Studio or an HMI.

You can use the following instructions to reset errors (events) in the RC Function Module. Refer to the descriptions of the instructions for details.

Instruction name	Instruction	Function	Reference
Reset Robot Control Error	ResetRCError	Resets current Controller errors in the RC Function Module.	page 10-2
Reset Robot Error	RC_Reset	Clears an OMRON robot error.	page 9-45

11-3 Error Table

This section provides tables of errors (events) that may occur in the RC Function Module.
Refer to the *NJ/NX-series Troubleshooting Manual (Cat. No. W503)* for information on all event codes of the NJ-series Controllers.

11-3-1 How to Read Error Tables

The contents of the error tables are described below.

Item	Description
Event code	The event code of the error in the NJ-series CPU Unit is given. The codes are given in eight hexadecimal digits. *1
Event name	The name of the error is given
Meaning	A short description of the error is given.
Assumed cause	The assumed cause of the error is given
Level	The level of influence on control is given. The abbreviations have the following meanings. Maj: Major fault level Prt: Partial fault level Min: Minor fault level Obs: Observation Info: Information The symbols have the following meanings. ○: Event levels that are defined by the system. ⦿: Event levels that can be changed by the user.*2
Reference	The catalog number of the manual that provides details on the event is given. The manual name that corresponds to the manual number is given before each error table.

*1. A version in parentheses in the Event code column is the unit version of the CPU Unit when the event occurs for only specific unit versions of the CPU Unit.
*2. This symbol appears only for events for which the user can change the event level.

11-3-2 Error Tables

Errors Related to Robot Control Function

The following table lists errors related to the common portion of the RC Function Module and OMRON robots.

Event code	Event name	Meaning	Assumed cause	Level: Maj	Level: Prt	Level: Min	Level: Obs	Level: Info	Reference
17C00000 hex	Robot Control Parameter Setting Error	The robot control parameter settings that were saved in non-volatile memory are missing.	• The power supply to the Controller was interrupted or communications with the Sysmac Studio were disconnected while downloading the robot control parameter settings or clearing memory. • Built-in non-volatile memory in the CPU Unit failed.		○				page 11-22
17C10000 hex	Remove SD Memory Card with Robot Control Function Enabled	The SD Memory Card was removed when the robot control function was enabled.	• The SD Memory Card was removed when the robot control function was enabled. • The files related to the robot control function in the SD Memory Card were edited or deleted when the robot control function was enabled.		○				page 11-23
17C20000 hex	Robot Control Function Enabled without SD Memory Card	The robot control function was enabled without inserting an SD Memory Card.	• More than one robot device was assigned to the Robot Basic Settings and the robot control function was enabled without inserting an SD Memory Card. • An SD Memory Card was inserted, however, it cannot be written due to write-protection, insufficient memory, or damage.		○				page 11-24
37C20000 hex	Robot Setting Mismatch	A mismatch was detected for the robot settings in the non-volatile memory and the SD Memory Card.	• An SD Memory Card was replaced. • The files related to the robot control function in the SD Memory Card were overwritten without using the Sysmac Studio.		○				page 11-25
47C00000 hex	Robot Control Initialization Error	Initialization of the Robot Control Function Module failed.	The CPU Unit has failed.		○				page 11-26
47C10000 hex	Robot Control Function Module System Error	A fatal error was detected in the Robot Control Function Module.	A fatal error was detected in the Robot Control Function Module.		○				page 11-26

Event code	Event name	Meaning	Assumed cause	Level					Reference
				Maj	Prt	Min	Obs	Info	
75000000 hex	Robot Control Period Exceeded	The robot control processing failures occurred two consecutive times during task period of primary periodic task.	• The task period of primary periodic task is too short for the amount of the user program that is executed in the primary periodic task. • Too many robot control instructions are executed for the task period of primary periodic task.		○				page 11-27
47C20000 hex	Robot Control Function Module Processing Error	An unexpected error occurred in the Robot Control Function Module.	An unexpected error was detected in the Robot Control Function Module.			○			page 11-27
55100000 hex	Robot Control Instruction Re-execution Disabled	An attempt was made to re-execute a robot control instruction that cannot be re-executed.	The *Execute* (Execute) input variable was re-executed during execution of the robot control instruction that has the *Execute* (Execute) input variable.			○			page 11-28
55110000 hex	V+ Task Number Setting Out of Range	The value of *TaskNo* (Task Number) input variable to a robot control instruction is out of range.	Instruction input parameter exceeded the valid range of the input variable.			○			page 11-28
55120000 hex	Illegal Robot Specification	The robot specified for the *Robot* (Robot) in-out variable to a robot control instruction does not exist.	The value of variable that is used for the subscript for array of _RC_RBT[ ] robot variable specified for the *Robot* (Robot) in-out variable to an instruction, is specified to the robot that does not exist.			○			page 11-29
55130000 hex	Illegal Parameter List Specification	The value of *PrgParam* (Parameter List) input variable to a robot control instruction is not correct.	The length of the parameter list specified for the *PrgParam* (Parameter List) input variable to an instruction was outside of the valid range.			○			page 11-29
55140000 hex	Starting Step Setting Out of Range	The parameter specified in the *StartStep* (Start Step) input variable to a robot control instruction is out of range.	Instruction input parameter exceeded the valid range of the input variable.			○			page 11-30
55150000 hex	Target Position Setting Out of Range	The value of *Position* (Target Position) input variable to a robot control instruction is out of range.	Instruction input parameter exceeded the valid range of the input variable.			○			page 11-30

Event code	Event name	Meaning	Assumed cause	Level					Reference
				Maj	Prt	Min	Obs	Info	
55160000 hex	Lefty and Righty Setting Out of Range	The value of *LeftyRighty* (Lefty/Righty Setting) member in the *ArmConfig* (Arm Configuration) input variable to a robot control instruction is out of range.	Instruction input parameter exceeded the valid range of the input variable.			○			page 11-31
55170000 hex	Above and Below Setting Out of Range	The value of *AboveBelow* (Above/ Below Setting) member in the *ArmConfig* (Arm Configuration) input variable to a robot control instruction is out of range.	Instruction input parameter exceeded the valid range of the input variable.			○			page 11-31
55180000 hex	Flip Setting Out of Range	The value of *Flip* (Flip Setting) member in the *ArmConfig* (Arm Configuration) input variable to a robot control instruction is out of range.	Instruction input parameter exceeded the valid range of the input variable.			○			page 11-32
55190000 hex	Velocity Profile Selection Out of Range	The value of *VelocityProfile* (Velocity Profile) member in the *MotionParams* (Motion Parameters) input variable to a robot control instruction is out of range.	Instruction input parameter exceeded the valid range of the input variable.			○			page 11-32
551A0000 hex	Velocity Mode Selection Out of Range	The value of *VelocityMode* (Velocity Selection) member in the *MotionParams* (Motion Parameters) input variable to a robot control instruction is out of range.	Instruction input parameter exceeded the valid range of the input variable.			○			page 11-33
551B0000 hex	Velocity Ratio Setting Out of Range	The value of *VelocityRatio* (Velocity Ratio) member in the *MotionParams* (Motion Parameters) input variable to a robot control instruction is out of range.	Instruction input parameter exceeded the valid range of the input variable.			○			page 11-33
551C0000 hex	Rotation Velocity Ratio Setting Out of Range	The value of *RotationVelocityRatio* (Rotation Velocity Ratio) member in the *MotionParams* (Motion Parameters) input variable to a robot control instruction is out of range.	Instruction input parameter exceeded the valid range of the input variable.			○			page 11-34

Event code	Event name	Meaning	Assumed cause	Level					Reference
				Maj	Prt	Min	Obs	Info	
551D0000 hex	Velocity Setting Out of Range	The value of *Velocity* (Velocity) member in the *MotionParams* (Motion Parameters) input variable to a robot control instruction is out of range.	Instruction input parameter exceeded the valid range of the input variable.			○			page 11-34
551E0000 hex	Acceleration Ratio Setting Out of Range	The value of *AccelerationRatio* (Acceleration Ratio) member in the *MotionParams* (Motion Parameters) input variable to a robot control instruction is out of range.	Instruction input parameter exceeded the valid range of the input variable.			○			page 11-35
551F0000 hex	Deceleration Ratio Setting Out of Range	The value of *DecelerationRatio* (Deceleration Ratio) member in the *MotionParams* (Motion Parameters) input variable to a robot control instruction is out of range.	Instruction input parameter exceeded the valid range of the input variable.			○			page 11-35
55200000 hex	Positioning Accuracy Selection Out of Range	The value of *NullingTolerance* (Positioning Accuracy) member in the *MotionParams* (Motion Parameters) input variable to a robot control instruction is out of range.	Instruction input parameter exceeded the valid range of the input variable.			○			page 11-36
55210000 hex	Rotation Limit Selection Out of Range	The value of *SingleTurn* (Rotation Limit) member in the *MotionParams* (Motion Parameters) input variable to a robot control instruction is out of range.	Instruction input parameter exceeded the valid range of the input variable.			○			page 11-36
55220000 hex	Buffer Mode Selection Out of Range	The value of *BufferMode* (Buffer Mode Selection) input variable to a robot control instruction is out of range.	Instruction input parameter exceeded the valid range of the input variable.			○			page 11-37
55230000 hex	Target Position Specification Method Setting Out of Range	The value of *PositionMode* (Target Position Specification Method) input variable to a robot control instruction is out of range.	Instruction input parameter exceeded the valid range of the input variable.			○			page 11-37

Event code	Event name	Meaning	Assumed cause	Level					Reference
				Maj	Prt	Min	Obs	Info	
552F0000 hex	Target Offset Setting Out of Range	The value of *Offset* (Target Offset) input variable to a robot control instruction is out of range.	Instruction input parameter exceeded the valid range of the input variable.			○			page 11-38
55330000 hex	Robot Control Instruction Executed while Robot is not Attached	An instruction required that a robot has been attached was executed for the robot that has not been attached.	An instruction that controls a robot was executed for the robot that has not been attached.			○			page 11-38
55350000 hex	Tool Coordination Transform Setting Out of Range	The value of *ToolCoordTransform* (Tool Conversion Coordinates) input variable to a robot control instruction is out of range.	Instruction input parameter exceeded the valid range of the input variable.			○			page 11-39
55360000 hex	Robot Control Instruction Multi-execution Disabled	Multiple robot control instructions that cannot be executed simultaneously were executed.	Multiple robot control instructions that cannot be executed simultaneously were executed.			○			page 11-39
553C0000 hex	Robot Control Instruction Multi-execution Buffer Limit Exceeded	The number of multi-execution for the robot control instructions exceeded the upper limit.	The total number of current robot control instructions and buffered robot control instructions exceeded eight.			○			page 11-40
553D0000 hex	Robot Control Instruction Executed with Calibration Not Completed	An instruction that is required for the calibration completion was executed for a robot whose calibration was not completed.	• An instruction that controls a robot was executed for the robot that the calibration has not been completed. • An instruction to synchronize the master machine and the robot was executed for a robot whose calibration was not completed.			○			page 11-40
553E0000 hex	Robot Control Instruction Executed while Robot High Power is OFF	An instruction required for the robot in a Power Enabled state was executed for the robot in which high power turns OFF.	An instruction that controls a robot was executed for the robot in which high power turns OFF.			○			page 11-41
553F0000 hex	Robot Already Attached	An attempt was made to attach a robot again or execute calibration for the robot that was already attached.	The target robot was already attached in the sequence control program.			○			page 11-41

Event code	Event name	Meaning	Assumed cause	Level					Reference
				Maj	Prt	Min	Obs	Info	
55400000 hex	Robot Control Instruction Executed while Robot is MANUAL Mode or is not COMP Mode	A robot control instruction for which the robot is MANUAL mode or is not COMP mode was executed.	• The robot that you control is MANUAL mode. • The robot in Auto mode that you control is not COMP mode.			○			page 11-42
55440000 hex	Cannot Execute Robot Control Instruction	The Robot Control Function Module is not running.	The robot control instruction was executed while the Robot Control Function Module was not running.			○			page 11-42
55480000 hex	Illegal Program Name Specification	The program name specified for the *PrgName* (Program Name) input variable to a robot control instruction is incorrect.	The length of the program name specified for the *PrgName* (Program Name) input variable to the instruction was outside of the valid range.			○			page 11-43
554A0000 hex	Unsupported Function Executed	The function that is not supported was executed.	The function that is not supported by the robot was executed.			○			page 11-43
554B0000 hex	Illegal IP Address Specification	The value of *IPAddr* input variable to a robot control instruction is not correct.	The length of the parameter list specified for the *IPAddr* input variable to an instruction was outside of the valid range.			○			page 11-44
554C0000 hex	Vision Error Occurred	An error occurred during execution of the vision function in the IPC Application Controller from a sequence control program.	An error occurred during execution of the vision function in the IPC Application Controller.			○			page 11-44
75010000 hex	Robot Control Common Error Occurred	A robot control common error occurred.	A partial fault level robot control common error occurred.			○			page 11-45
75020000 hex	EtherCAT Slave Disconnection Error	One of the following occurred for the EtherCAT slave that is allocated to a robot. • Disconnect or replace the slave • Disable the slave	One of the following occurred for the EtherCAT slave that is allocated to a robot. • Disconnection or replacement • Disablement			○			page 11-45
75030000 hex	Robot Error Occurred	An error occurred in the robot that the robot control instruction execution is in progress.	An error occurred in the robot that the robot control instruction execution is in progress.			○			page 11-46

Event code	Event name	Meaning	Assumed cause	Level					Reference
				Maj	Prt	Min	Obs	Info	
75040000 hex	Robot Control Period Exceeded	The robot control processing failures occurred two consecutive times during task period of primary periodic task.	• The task period of primary periodic task is too short for the amount of the user program that is executed in the primary periodic task. • Too many robot control instructions are executed for the task period of primary periodic task.			○			page 11-46
85800000 hex	EtherCAT Slave Communications Error	A communications error occurred for the EtherCAT slave that is allocated to a robot.	A communications error occurred for the EtherCAT slave that is allocated to a robot.			○			page 11-47
96040000 hex	V+ Program Error	An error occurred in the V+ program.	An error occurred in the V+ program that was being executed.					○	page 11-47
96050000 hex	V+ Program Warning	The V+ program issued a warning message.	The V+ program that was being executed issued a warning message.					○	page 11-48
96060000 hex	V+ Program Information	The V+ program issued an information message.	The V+ program that was being executed issued an information message.					○	page 11-48
96090000 hex	Robot Manual Mode Started	The robot was set to Manual mode.	The robot was set to Manual mode.					○	page 11-49
960A0000 hex	Robot Auto Mode Started	The robot was set to Auto mode.	The robot was set to Auto mode.					○	page 11-49

Errors Related to Robot Control Instructions

The following table lists errors related to robot control instructions for the RC Function Module.

Event code	Event name	Meaning	Assumed cause	Level					Reference
				Maj	Prt	Min	Obs	Info	
54015510 hex	Robot Control Instruction Re-execution Disabled	An attempt was made to re-execute a robot control instruction that cannot be re-executed.	The *Execute* (Execute) input variable was re-executed during execution of the robot control instruction that has the *Execute* (Execute) input variable.				○		page 11-50
54015511 hex	V+ Task Number Setting Out of Range	The value of *TaskNo* (Task Number) input variable to a robot control instruction is out of range.	Instruction input parameter exceeded the valid range of the input variable.				○		page 11-51

Event code	Event name	Meaning	Assumed cause	Level					Refer-ence
				Maj	Prt	Min	Obs	Info	
54015512 hex	Illegal Robot Specification	The robot specified for the *Robot* (Robot) in-out variable to a robot control instruction does not exist.	The value of variable that is used for the subscript for array of _RC_RBT[ ] robot variable specified for the *Robot* (Robot) in-out variable to a instruction, is specified to the robot that does not exist.				○		page 11-52
54015513 hex	Illegal Parameter List Specification	The value of *PrgParam* (Parameter List) input variable to a robot control instruction is not correct.	The length of the parameter list specified for the *PrgParam* (Parameter List) input variable to an instruction was outside of the valid range.				○		page 11-53
54015514 hex	Starting Step Setting Out of Range	The value specified in the *StartStep* (Start Step) input variable to a robot control instruction is out of range.	Instruction input parameter exceeded the valid range of the input variable.				○		page 11-54
54015515 hex	Target Position Setting Out of Range	The value of *Position* (Target Position) input variable to a robot control instruction is out of range.	Instruction input parameter exceeded the valid range of the input variable.				○		page 11-54
54015516 hex	Lefty and Righty Setting Out of Range	The value of *LeftyRighty* (Lefty/Righty Setting) member in the *ArmConfig* (Arm Configuration) input variable to a robot control instruction is out of range.	Instruction input parameter exceeded the valid range of the input variable.				○		page 11-55
54015517 hex	Above and Below Setting Out of Range	The value of *AboveBelow* (Above/Below Setting) member in the *ArmConfig* (Arm Configuration) input variable to a robot control instruction is out of range.	Instruction input parameter exceeded the valid range of the input variable.				○		page 11-56
54015518 hex	Flip Setting Out of Range	The value of *Flip* (Flip Setting) member in the *ArmConfig* (Arm Configuration) input variable to a robot control instruction is out of range.	Instruction input parameter exceeded the valid range of the input variable.				○		page 11-57

Event code	Event name	Meaning	Assumed cause	Level					Refer-ence
				Maj	Prt	Min	Obs	Info	
54015519 hex	Velocity Profile Selection Out of Range	The value of *VelocityProfile* (Velocity Profile) member in the *MotionParams* (Motion Parameters) input variable to a robot control instruction is out of range.	Instruction input parameter exceeded the valid range of the input variable.				○		page 11-58
5401551A hex	Velocity Mode Selection Out of Range	The value of *VelocityMode* (Velocity Selection) member in the *MotionParams* (Motion Parameters) input variable to a robot control instruction is out of range.	Instruction input parameter exceeded the valid range of the input variable.				○		page 11-59
5401551B hex	Velocity Ratio Setting Out of Range	The value of *VelocityRatio* (Velocity Ratio) member in the *MotionParams* (Motion Parameters) input variable to a robot control instruction is out of range.	Instruction input parameter exceeded the valid range of the input variable.				○		page 11-60
5401551C hex	Rotation Velocity Ratio Setting Out of Range	The value of *RotationVelocityRatio* (Rotation Velocity Ratio) member in the *MotionParams* (Motion Parameters) input variable to a robot control instruction is out of range.	Instruction input parameter exceeded the valid range of the input variable.				○		page 11-61
5401551D hex	Velocity Setting Out of Range	The value of *Velocity* (Velocity) member in the *MotionParams* (Motion Parameters) input variable to a robot control instruction is out of range.	Instruction input parameter exceeded the valid range of the input variable.				○		page 11-62
5401551E hex	Acceleration Ratio Setting Out of Range	The value of *AccelerationRatio* (Acceleration Ratio) member in the *MotionParams* (Motion Parameters) input variable to a robot control instruction is out of range.	Instruction input parameter exceeded the valid range of the input variable.				○		page 11-63

Event code	Event name	Meaning	Assumed cause	Level					Refer-ence
				Maj	Prt	Min	Obs	Info	
5401551F hex	Deceleration Ratio Setting Out of Range	The value of *DecelerationRatio* (Deceleration Ratio) member in the *MotionParams* (Motion Parameters) input variable to a robot control instruction is out of range.	Instruction input parameter exceeded the valid range of the input variable.				○		page 11-64
54015520 hex	Positioning Accuracy Selection Out of Range	The value of *NullingTolerance* (Positioning Accuracy) member in the *MotionParams* (Motion Parameters) input variable to a robot control instruction is out of range.	Instruction input parameter exceeded the valid range of the input variable.				○		page 11-65
54015521 hex	Rotation Limit Selection Out of Range	The value of *SingleTurn* (Rotation Limit) member in the *MotionParams* (Motion Parameters) input variable to a robot control instruction is out of range.	Instruction input parameter exceeded the valid range of the input variable.				○		page 11-66
54015522 hex	Buffer Mode Selection Out of Range	The value of *BufferMode* (Buffer Mode Selection) input variable to a robot control instruction is out of range.	Instruction input parameter exceeded the valid range of the input variable.				○		page 11-67
54015523 hex	Target Position Specification Method Setting Out of Range	The value of *PositionMode* (Target Position Specification Method) input variable to a robot control instruction is out of range.	Instruction input parameter exceeded the valid range of the input variable.				○		page 11-68
5401552F hex	Target Offset Setting Out of Range	The value of *Offset* (Target Offset) input variable to a robot control instruction is out of range.	Instruction input parameter exceeded the valid range of the input variable.				○		page 11-69
54015533 hex	Robot Control Instruction Executed while Robot is not Attached	An instruction required that a robot has been attached was executed for the robot that has not been attached.	An instruction that controls a robot was executed for the robot that has not been attached.				○		page 11-70
54015535 hex	Tool Coordination Transform Setting Out of Range	The value of *ToolCoordTransform* (Tool Conversion Coordinates) input variable to a robot control instruction is out of range.	Instruction input parameter exceeded the valid range of the input variable.				○		page 11-71

Event code	Event name	Meaning	Assumed cause	Level: Maj	Level: Prt	Level: Min	Level: Obs	Level: Info	Reference
54015536 hex	Robot Control Instruction Multi-execution Disabled	Multiple robot control instructions that cannot be executed simultaneously were executed.	Multiple robot control instructions that cannot be executed simultaneously were executed.				○		page 11-72
5401553C hex	Robot Control Instruction Multi-execution Buffer Limit Exceeded	The number of multi-execution for the robot control instructions exceeded the upper limit.	The total number of current robot control instructions and buffered robot control instructions exceeded eight.				○		page 11-73
5401553D hex	Robot Control Instruction Executed with Calibration Not Completed	An instruction that is required for the calibration completion was executed for a robot whose calibration was not completed.	• An instruction that controls a robot was executed for the robot that the calibration has not been completed. • An instruction to synchronize the master machine and the robot was executed for a robot whose calibration was not completed.				○		page 11-74
5401553E hex	Robot Control Instruction Executed while Robot High Power is OFF	An instruction required for the robot in a Power Enabled state was executed for the robot in which high power turns OFF.	An instruction that controls a robot was executed for the robot in which high power turns OFF.				○		page 11-75
5401553F hex	Robot Already Attached	An attempt was made to attach a robot again or execute calibration for the robot that was already attached.	The target robot was already attached in the sequence control program.				○		page 11-76
54015540 hex	Robot Control Instruction Executed while Robot is MANUAL Mode or is not COMP Mode	A robot control instruction for which the robot is MANUAL mode or is not COMP mode was executed.	• The robot that you control is MANUAL mode. • The robot in Auto mode that you control is not COMP mode.				○		page 11-77
54015544 hex	Cannot Execute Robot Control Instruction	The Robot Control Function Module is not running.	The robot control instruction was executed while the Robot Control Function Module was not running.				○		page 11-78

Event code	Event name	Meaning	Assumed cause	Level					Refer-ence
				Maj	Prt	Min	Obs	Info	
54015548 hex	Illegal Program Name Specification	The program name specified for the *PrgName* (Program Name) input variable to a robot control instruction is incorrect.	The length of the program name specified for the *PrgName* (Program Name) input variable to the instruction was outside of the valid range.				○		page 11-79
5401554A hex	Unsupported Function Executed	The function that is not supported was executed.	The function that is not supported by the robot was executed.				○		page 11-79
5401554B hex	Illegal IP Address Specification	The value of *IPAddr* input variable to a robot control instruction is not correct.	The length of the parameter list specified for the *IPAddr* input variable to an instruction was outside of the valid range.				○		page 11-80

11-4 Error Descriptions

11-4-1 How to Read Error Descriptions

The items that are used to describe individual errors (events) are described in the following copy of an error table.

<table>
<tr><td>Event name</td><td colspan="4">Gives the name of the error.</td><td>Event code</td><td colspan="2">Gives the code of the error.</td></tr>
<tr><td>Meaning</td><td colspan="7">Gives a short description of the error.</td></tr>
<tr><td>Source</td><td colspan="3">Gives the source of the error.</td><td>Source details</td><td>Gives details on the source of the error.</td><td>Detection timing</td><td>Tells when the error is detected.</td></tr>
<tr><td>Error attributes</td><td>Level</td><td colspan="2">Tells the level of influence on control. [*1]</td><td>Recovery</td><td>Gives the recovery method. [*2]</td><td>Log category</td><td>Tells which log the error is saved in. [*3]</td></tr>
<tr><td>Effects</td><td>User program</td><td colspan="2">Tells what will happen to execution of the user program. [*4]</td><td>Operation</td><td colspan="3">Provides special information on the operation that results from the error.</td></tr>
<tr><td>Indicators</td><td colspan="7">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.</td></tr>
<tr><td rowspan="2">System-defined variables</td><td colspan="3">Variable</td><td colspan="2">Data type</td><td colspan="2">Name</td></tr>
<tr><td colspan="7">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.</td></tr>
<tr><td rowspan="2">Cause and correction</td><td colspan="3">Assumed cause</td><td colspan="2">Correction</td><td colspan="2">Prevention</td></tr>
<tr><td colspan="7">Lists the possible causes, corrections, and preventive measures for the error.</td></tr>
<tr><td>Attached information</td><td colspan="7">This is the attached information that is displayed by the Sysmac Studio or an HMI. [*5]</td></tr>
<tr><td>Precautions/ Remarks</td><td colspan="7">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.</td></tr>
</table>

*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. Refer to the appendices of the *NJ/NX-series Troubleshooting Manual (Cat. No. W503)* for the applicable range of the HMI Troubleshooter.

11-4-2 Error Descriptions

This section describes the error details for the robot control function and robot control instructions.

Errors in Robot Control Function

This section describes the details of the errors related to the common portion of the RC Function Module and OMRON robots.

<table>
<tr><td>Event name</td><td colspan="3">Robot Control Parameter Setting Error</td><td>Event code</td><td colspan="2">17C00000 hex</td></tr>
<tr><td>Meaning</td><td colspan="6">The robot control parameter settings that were saved in non-volatile memory are missing.</td></tr>
<tr><td>Source</td><td colspan="2">Robot Control Function Module</td><td>Source details</td><td>Robot control common</td><td>Detection timing</td><td>At power ON, at Controller reset, or when downloading</td></tr>
<tr><td>Error attributes</td><td>Level</td><td>Partial fault</td><td>Recovery</td><td>Cycle the power supply or reset the Controller.</td><td>Log category</td><td>System</td></tr>
<tr><td>Effects</td><td>User program</td><td>Continues.</td><td>Operation</td><td colspan="3">It will not be possible to perform robot control.
The V+ program stops.
All OMRON robots stop.</td></tr>
<tr><td rowspan="2">System-defined variables</td><td colspan="2">Variable</td><td colspan="2">Data type</td><td colspan="2">Name</td></tr>
<tr><td colspan="2">_RC_COM.PFaultLvl.Active</td><td colspan="2">BOOL</td><td colspan="2">Robot Control Common Partial Fault Occurrence</td></tr>
<tr><td rowspan="3">Cause and correction</td><td colspan="2">Assumed cause</td><td colspan="2">Correction</td><td colspan="2">Prevention</td></tr>
<tr><td colspan="2">The power supply to the Controller was interrupted or communications with the Sysmac Studio were disconnected while downloading the robot control parameter settings or clearing memory.</td><td colspan="2">Download the robot control parameter settings from the Sysmac Studio again.</td><td colspan="2">Do not turn OFF the power supply to the CPU Unit during save processing for the parameters.</td></tr>
<tr><td colspan="2">Built-in non-volatile memory in the CPU Unit failed.</td><td colspan="2">If the error occurs even after the above correction is performed, non-volatile memory may have failed. After you replace the CPU Unit, download all settings including the robot settings from the Sysmac Studio to the new CPU Unit.</td><td colspan="2">None</td></tr>
<tr><td>Attached information</td><td colspan="6">None</td></tr>
<tr><td>Precautions/ Remarks</td><td colspan="6">None</td></tr>
</table>

<table>
<tr><td>Event name</td><td colspan="3">Remove SD Memory Card with Robot Control Function Enabled</td><td>Event code</td><td colspan="2">17C10000 hex</td></tr>
<tr><td>Meaning</td><td colspan="6">The SD Memory Card was removed when the robot control function was enabled.</td></tr>
<tr><td>Source</td><td colspan="2">Robot Control Function Module</td><td>Source details</td><td>Robot control common</td><td>Detection timing</td><td>Continuously</td></tr>
<tr><td>Error attributes</td><td>Level</td><td>Partial fault</td><td>Recovery</td><td>Cycle the power supply or reset the Controller.</td><td>Log category</td><td>System</td></tr>
<tr><td>Effects</td><td>User program</td><td>Continues.</td><td>Operation</td><td colspan="3">It will not be possible to perform robot control.
The V+ program stops.
All OMRON robots stop.</td></tr>
<tr><td rowspan="2">System-defined variables</td><td colspan="2">Variable</td><td colspan="2">Data type</td><td colspan="2">Name</td></tr>
<tr><td colspan="2">_RC_COM.PFaultLvl.Active</td><td colspan="2">BOOL</td><td colspan="2">Robot Control Common Partial Fault Occurrence</td></tr>
<tr><td rowspan="3">Cause and correction</td><td colspan="2">Assumed cause</td><td colspan="2">Correction</td><td colspan="2">Prevention</td></tr>
<tr><td colspan="2">The SD Memory Card was removed when the robot control function was enabled.</td><td colspan="2">Insert the SD Memory Card again, and then cycle the power supply to the CPU Unit or reset.</td><td colspan="2">Do not remove the SD Memory Card when the robot control function is enabled.</td></tr>
<tr><td colspan="2">The files related to the robot control function in the SD Memory Card were edited or deleted when the robot control function was enabled.</td><td colspan="2">Download the files related to the robot control function from the Sysmac Studio again.</td><td colspan="2">Do not edit or delete the files related to the robot control function on the SD Memory Card.</td></tr>
<tr><td>Attached information</td><td colspan="6">None</td></tr>
<tr><td>Precautions/ Remarks</td><td colspan="6">None</td></tr>
</table>

<table>
<tr><td>Event name</td><td colspan="3">Robot Control Function Enabled without SD Memory Card</td><td>Event code</td><td colspan="2">17C20000 hex</td></tr>
<tr><td>Meaning</td><td colspan="6">The robot control function was enabled without inserting an SD Memory Card.</td></tr>
<tr><td>Source</td><td colspan="2">Robot Control Function Module</td><td>Source details</td><td>Robot control common</td><td>Detection timing</td><td>At power ON, at Controller reset, or when downloading</td></tr>
<tr><td>Error attributes</td><td>Level</td><td>Partial fault</td><td>Recovery</td><td>Cycle the power supply or reset the Controller.</td><td>Log category</td><td>System</td></tr>
<tr><td>Effects</td><td>User program</td><td>Continues.</td><td>Operation</td><td colspan="3">It will not be possible to perform robot control.
The V+ program stops.
All OMRON robots stop.</td></tr>
<tr><td rowspan="2">System-defined variables</td><td colspan="2">Variable</td><td colspan="2">Data type</td><td colspan="2">Name</td></tr>
<tr><td colspan="2">_RC_COM.PFaultLvl.Active</td><td colspan="2">BOOL</td><td colspan="2">Robot Control Common Partial Fault Occurrence</td></tr>
<tr><td rowspan="3">Cause and correction</td><td colspan="2">Assumed cause</td><td colspan="2">Correction</td><td colspan="2">Prevention</td></tr>
<tr><td colspan="2">More than one robot device was assigned to the Robot Basic Settings and the robot control function was enabled without inserting an SD Memory Card.</td><td colspan="2">Insert an SD Memory Card into the CPU Unit if it is not inserted.</td><td colspan="2">None</td></tr>
<tr><td colspan="2">An SD Memory Card was inserted, however, it cannot be written due to write-protection, insufficient memory, or damage.</td><td colspan="2">Change the settings for the SD Memory Card or increase available space.
If this error occurs again even after this correction, replace the SD Memory Card.</td><td colspan="2">None</td></tr>
<tr><td>Attached information</td><td colspan="6">None</td></tr>
<tr><td>Precautions/ Remarks</td><td colspan="6">None</td></tr>
</table>

<table>
<tr><th>Event name</th><td colspan="3">Robot Setting Mismatch</td><th>Event code</th><td colspan="2">37C20000 hex</td></tr>
<tr><th>Meaning</th><td colspan="6">A mismatch was detected for the robot settings in the non-volatile memory and the SD Memory Card.</td></tr>
<tr><th>Source</th><td colspan="2">Robot Control Function Module</td><th>Source details</th><td>Robot control common</td><th>Detection timing</th><td>At power ON, at Controller reset, or when downloading</td></tr>
<tr><th>Error attributes</th><th>Level</th><td>Partial fault</td><th>Recovery</th><td>Cycle the power supply or reset the Controller.</td><th>Log category</th><td>System</td></tr>
<tr><th>Effects</th><th>User program</th><td>Continues.</td><th>Operation</th><td colspan="3">It will not be possible to perform robot control.
The V+ program stops.
All OMRON robots stop.</td></tr>
<tr><th rowspan="2">System-defined variables</th><th colspan="2">Variable</th><th colspan="2">Data type</th><th colspan="2">Name</th></tr>
<tr><td colspan="2">_RC_COM.PFaultLvl.Active</td><td colspan="2">BOOL</td><td colspan="2">Robot Control Common Partial Fault Occurrence</td></tr>
<tr><th rowspan="3">Cause and correction</th><th colspan="2">Assumed cause</th><th colspan="2">Correction</th><th colspan="2">Prevention</th></tr>
<tr><td colspan="2">An SD Memory Card was replaced.</td><td colspan="2" rowspan="2">Perform the Clear All Memory operation to clear all settings including the robot settings from the Sysmac Studio, cycle the power supply to the CPU Unit, and then download the settings. Or, use the SD Memory Card that you used.</td><td colspan="2">None</td></tr>
<tr><td colspan="2">The files related to the robot control function in the SD Memory Card were overwritten without using the Sysmac Studio.</td><td colspan="2">Do not delete or edit the files related to the robot control function in the SD Memory Card.</td></tr>
<tr><th>Attached information</th><td colspan="6">None</td></tr>
<tr><th>Precautions/ Remarks</th><td colspan="6">None</td></tr>
</table>

Event name	Robot Control Initialization Error			**Event code**	47C00000 hex	
Meaning	Initialization of the Robot Control Function Module failed.					
Source	Robot Control Function Module		**Source details**	Robot control common	**Detection timing**	At power ON, at Controller reset, or when downloading
Error attributes	**Level**	Partial fault	**Recovery**	Cycle the power supply or reset the Controller.	**Log category**	System
Effects	**User program**	Continues.	**Operation**	It will not be possible to perform robot control. The V+ program stops. All OMRON robots stop.		
System-defined variables	**Variable**		**Data type**		**Name**	
	_RC_COM.PFaultLvl.Active		BOOL		Robot Control Common Partial Fault Occurrence	
Cause and correction	**Assumed cause**		**Correction**		**Prevention**	
	The CPU Unit has failed.		Replace the CPU Unit.		None	
Attached information	Attached information 1: System information					
Precautions/ Remarks	None					

Event name	Robot Control Function Module System Error			**Event code**	47C10000 hex	
Meaning	A fatal error was detected in the Robot Control Function Module.					
Source	Robot Control Function Module		**Source details**	Robot control common	**Detection timing**	Continuously
Error attributes	**Level**	Partial fault	**Recovery**	Cycle the power supply or reset the Controller.	**Log category**	System
Effects	**User program**	Continues.	**Operation**	It will not be possible to perform robot control. The V+ program stops. All OMRON robots stop.		
System-defined variables	**Variable**		**Data type**		**Name**	
	_RC_COM.PFaultLvl.Active		BOOL		Robot Control Common Partial Fault Occurrence	
Cause and correction	**Assumed cause**		**Correction**		**Prevention**	
	A fatal error was detected in the Robot Control Function Module.		Use the Sysmac Controller Log Upload Tool to obtain the log file, and contact your OMRON representative.		None	
Attached information	Attached information 1: System information Attached information 2: System information Attached information 3: System information Attached information 4: System information					
Precautions/ Remarks	None					

Event name	Robot Control Period Exceeded			**Event code**	75000000 hex	
Meaning	The robot control processing failures occurred two consecutive times during task period of primary periodic task.					
Source	Robot Control Function Module		**Source details**	Robot control common	**Detection timing**	Continuously
Error attributes	**Level**	Partial fault	**Recovery**	Error reset	**Log category**	System
Effects	**User program**	Continues.	**Operation**	It will not be possible to perform robot control. The V+ program stops. All OMRON robots stop.		
System-defined variables	**Variable**		**Data type**		**Name**	
	_RC_COM.PFaultLvl.Active		BOOL		Robot Control Common Partial Fault Occurrence	
Cause and correction	**Assumed cause**		**Correction**		**Prevention**	
	The task period of primary periodic task is too short for the amount of the user program that is executed in the primary periodic task.		Check the task period of primary periodic task in the **Task Period Monitor** on the Sysmac Studio, and set the task period of the primary periodic task to be long enough to complete all processing.		Set the task period of the primary periodic task to be long enough to complete all processing.	
	Too many robot control instructions are executed for the task period of primary periodic task.		Decrease the robot control instructions that are executed.		Assign the user program that is executed in a certain period to the primary periodic task.	
Attached information	None					
Precautions/ Remarks	None					

Event name	Robot Control Function Module Processing Error			**Event code**	47C20000 hex	
Meaning	An unexpected error occurred in the Robot Control Function Module.					
Source	Robot Control Function Module		**Source details**	Robot control common	**Detection timing**	Continuously
Error attributes	**Level**	Minor fault	**Recovery**	Error reset	**Log category**	System
Effects	**User program**	Continues.	**Operation**	Not affected.		
System-defined variables	**Variable**		**Data type**		**Name**	
	_RC_COM.MFaultLvl.Active		BOOL		Robot Control Common Minor Fault Occurrence	
Cause and correction	**Assumed cause**		**Correction**		**Prevention**	
	An unexpected error was detected in the Robot Control Function Module.		Use the Sysmac Controller Log Upload Tool to obtain the log file, and contact your OMRON representative.		None	
Attached information	Attached information 1: System information Attached information 2: System information Attached information 3: System information Attached information 4: System information					
Precautions/ Remarks	None					

Event name	Robot Control Instruction Re-execution Disabled			**Event code**	55100000 hex	
Meaning	An attempt was made to re-execute a robot control instruction that cannot be re-executed.					
Source	Robot Control Function Module		**Source details**	Robot control common or robot	**Detection timing**	At instruction execution
Error attributes	**Level**	Minor fault	**Recovery**	Error reset	**Log category**	System
Effects	**User program**	Continues.	**Operation**	The relevant robot stops.		
System-defined variables	**Variable**		**Data type**		**Name**	
	_RC_COM.MFaultLvl.Active		BOOL		Robot Control Common Minor Fault Occurrence	
	_RC_RBT[*].MFaultLvl.Active		BOOL		Robot Minor Fault Occurrence	
Cause and correction	**Assumed cause**		**Correction**		**Prevention**	
	The *Execute* (Execute) input variable was re-executed during execution of the robot control instruction that has the *Execute* (Execute) input variable.		Correct the program so that the *Execute* (Execute) input variable does not change to TRUE until the *Busy* (Executing) output variable to the instruction changes to FALSE.		When using instructions that cannot be re-executed, include a condition for the *Execute* (Execute) input variable so that it does not change to TRUE unless the *Busy* (Executing) output variable for the instruction to be used is FALSE. Or, stop the instruction before executing it again.	
Attached information	None					
Precautions/ Remarks	The robot control instruction cannot change the current operation by the re-execution.					

Event name	V+ Task Number Setting Out of Range			**Event code**	55110000 hex	
Meaning	The value of *TaskNo* (Task Number) input variable to a robot control instruction is out of range.					
Source	Robot Control Function Module		**Source details**	Robot control common	**Detection timing**	At instruction execution
Error attributes	**Level**	Minor fault	**Recovery**	Error reset	**Log category**	System
Effects	**User program**	Continues.	**Operation**	Not affected.		
System-defined variables	**Variable**		**Data type**		**Name**	
	_RC_COM.MFaultLvl.Active		BOOL		Robot Control Common Minor Fault Occurrence	
Cause and correction	**Assumed cause**		**Correction**		**Prevention**	
	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.	
Attached information	None					
Precautions/ Remarks	None					

Event name	Illegal Robot Specification			**Event code**	55120000 hex	
Meaning	The robot specified for the *Robot* (Robot) in-out variable to a robot control instruction does not exist.					
Source	Robot Control Function Module		**Source details**	Robot control common	**Detection timing**	At instruction execution
Error attributes	**Level**	Minor fault	**Recovery**	Error reset	**Log category**	System
Effects	**User program**	Continues.	**Operation**	Not affected.		
System-defined variables	**Variable**		**Data type**		**Name**	
	_RC_COM.MFaultLvl.Active		BOOL		Robot Control Common Minor Fault Occurrence	
Cause and correction	**Assumed cause**		**Correction**		**Prevention**	
	The value of variable that is used for the subscript for array of _RC_RBT[ ] robot variable specified for the *Robot* (Robot) in-out variable to an instruction, is specified to the robot that does not exist.		Correct the value of variable so that existing robot is specified.		If a variables is used for an input parameter to the instruction, check that the value of variable is correct.	
Attached information	None					
Precautions/ Remarks	None					

Event name	Illegal Parameter List Specification			**Event code**	55130000 hex	
Meaning	The value of *PrgParam* (Parameter List) input variable to a robot control instruction is not correct.					
Source	Robot Control Function Module		**Source details**	Robot control common	**Detection timing**	At instruction execution
Error attributes	**Level**	Minor fault	**Recovery**	Error reset	**Log category**	System
Effects	**User program**	Continues.	**Operation**	Not affected.		
System-defined variables	**Variable**		**Data type**		**Name**	
	_RC_COM.MFaultLvl.Active		BOOL		Robot Control Common Minor Fault Occurrence	
Cause and correction	**Assumed cause**		**Correction**		**Prevention**	
	The length of the parameter list specified for the *PrgParam* (Parameter List) input variable to an instruction was outside of the valid range.		Correct the length of the string in the parameter list specified for the *PrgParam* (Parameter List) input variable to the instruction so that the length of the string is in the valid range.		Make sure that the length of the string in the parameter list specified for the *PrgParam* (Parameter List) input variable to the instruction is in the valid range.	
Attached information	None					
Precautions/ Remarks	None					

Event name	Starting Step Setting Out of Range			**Event code**	55140000 hex	
Meaning	The value specified in the *StartStep* (Start Step) input variable to a robot control instruction is out of range.					
Source	Robot Control Function Module		**Source details**	Robot control common	**Detection timing**	At instruction execution
Error attributes	**Level**	Minor fault	**Recovery**	Error reset	**Log category**	System
Effects	**User program**	Continues.	**Operation**	Not affected.		
System-defined variables	**Variable**		**Data type**		**Name**	
	_RC_COM.MFaultLvl.Active		BOOL		Robot Control Common Minor Fault Occurrence	
Cause and correction	**Assumed cause**		**Correction**		**Prevention**	
	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.	
Attached information	None					
Precautions/ Remarks	None					

Event name	Target Position Setting Out of Range			**Event code**	55150000 hex	
Meaning	The value of *Position* (Target Position) input variable to a robot control instruction is out of range.					
Source	Robot Control Function Module		**Source details**	Robot	**Detection timing**	At instruction execution
Error attributes	**Level**	Minor fault	**Recovery**	Error reset	**Log category**	System
Effects	**User program**	Continues.	**Operation**	The relevant robot stops.		
System-defined variables	**Variable**		**Data type**		**Name**	
	_RC_RBT[*].MFaultLvl.Active		BOOL		Robot Minor Fault Occurrence	
Cause and correction	**Assumed cause**		**Correction**		**Prevention**	
	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.	
Attached information	Attached information 1: Element number that is out of range in the *Position* (Target Position) input variable.					
Precautions/ Remarks	None					

Event name	Lefty and Righty Setting Out of Range			Event code	55160000 hex	
Meaning	The value of *LeftyRighty* (Lefty/Righty Setting) member in the *ArmConfig* (Arm Configuration) input variable to a robot control instruction is out of range.					
Source	Robot Control Function Module		Source details	Robot	Detection timing	At instruction execution
Error attributes	Level	Minor fault	Recovery	Error reset	Log category	System
Effects	User program	Continues.	Operation	The relevant robot stops.		
System-defined variables	Variable		Data type		Name	
	_RC_RBT[*].MFaultLvl.Active		BOOL		Robot Minor Fault Occurrence	
Cause and correction	Assumed cause		Correction		Prevention	
	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.	
Attached information	None					
Precautions/ Remarks	None					

Event name	Above and Below Setting Out of Range			Event code	55170000 hex	
Meaning	The value of *AboveBelow* (Above/Below Setting) member in the *ArmConfig* (Arm Configuration) input variable to a robot control instruction is out of range.					
Source	Robot Control Function Module		Source details	Robot	Detection timing	At instruction execution
Error attributes	Level	Minor fault	Recovery	Error reset	Log category	System
Effects	User program	Continues.	Operation	The relevant robot stops.		
System-defined variables	Variable		Data type		Name	
	_RC_RBT[*].MFaultLvl.Active		BOOL		Robot Minor Fault Occurrence	
Cause and correction	Assumed cause		Correction		Prevention	
	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.	
Attached information	None					
Precautions/ Remarks	None					

Event name	Flip Setting Out of Range			Event code	55180000 hex	
Meaning	The value of *Flip* (Flip Setting) member in the *ArmConfig* (Arm Configuration) input variable to a robot control instruction is out of range.					
Source	Robot Control Function Module		Source details	Robot	Detection timing	At instruction execution
Error attributes	Level	Minor fault	Recovery	Error reset	Log category	System
Effects	User program	Continues.	Operation	The relevant robot stops.		
System-defined variables	Variable		Data type		Name	
	_RC_RBT[*].MFaultLvl.Active		BOOL		Robot Minor Fault Occurrence	
Cause and correction	Assumed cause		Correction		Prevention	
	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.	
Attached information	None					
Precautions/ Remarks	None					

Event name	Velocity Profile Selection Out of Range			Event code	55190000 hex	
Meaning	The value of *VelocityProfile* (Velocity Profile) member in the *MotionParams* (Motion Parameters) input variable to a robot control instruction is out of range.					
Source	Robot Control Function Module		Source details	Robot	Detection timing	At instruction execution
Error attributes	Level	Minor fault	Recovery	Error reset	Log category	System
Effects	User program	Continues.	Operation	The relevant robot stops.		
System-defined variables	Variable		Data type		Name	
	_RC_RBT[*].MFaultLvl.Active		BOOL		Robot Minor Fault Occurrence	
Cause and correction	Assumed cause		Correction		Prevention	
	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.	
Attached information	None					
Precautions/ Remarks	None					

Event name	Velocity Mode Selection Out of Range			**Event code**	551A0000 hex	
Meaning	The value of *VelocityMode* (Velocity Selection) member in the *MotionParams* (Motion Parameters) input variable to a robot control instruction is out of range.					
Source	Robot Control Function Module		**Source details**	Robot	**Detection timing**	At instruction execution
Error attributes	**Level**	Minor fault	**Recovery**	Error reset	**Log category**	System
Effects	**User program**	Continues.	**Operation**	The relevant robot stops.		
System-defined variables	**Variable**		**Data type**		**Name**	
	_RC_RBT[*].MFaultLvl.Active		BOOL		Robot Minor Fault Occurrence	
Cause and correction	**Assumed cause**		**Correction**		**Prevention**	
	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.	
Attached information	None					
Precautions/ Remarks	None					

Event name	Velocity Ratio Setting Out of Range			**Event code**	551B0000 hex	
Meaning	The value of *VelocityRatio* (Velocity Ratio) member in the *MotionParams* (Motion Parameters) input variable to a robot control instruction is out of range.					
Source	Robot Control Function Module		**Source details**	Robot	**Detection timing**	At instruction execution
Error attributes	**Level**	Minor fault	**Recovery**	Error reset	**Log category**	System
Effects	**User program**	Continues.	**Operation**	The relevant robot stops.		
System-defined variables	**Variable**		**Data type**		**Name**	
	_RC_RBT[*].MFaultLvl.Active		BOOL		Robot Minor Fault Occurrence	
Cause and correction	**Assumed cause**		**Correction**		**Prevention**	
	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.	
Attached information	None					
Precautions/ Remarks	None					

Event name	Rotation Velocity Ratio Setting Out of Range			Event code	551C0000 hex	
Meaning	The value of *RotationVelocityRatio* (Rotation Velocity Ratio) member in the *MotionParams* (Motion Parameters) input variable to a robot control instruction is out of range.					
Source	Robot Control Function Module		Source details	Robot	Detection timing	At instruction execution
Error attributes	Level	Minor fault	Recovery	Error reset	Log category	System
Effects	User program	Continues.	Operation	The relevant robot stops.		
System-defined variables	Variable		Data type		Name	
	_RC_RBT[*].MFaultLvl.Active		BOOL		Robot Minor Fault Occurrence	
Cause and correction	Assumed cause		Correction		Prevention	
	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.	
Attached information	None					
Precautions/ Remarks	None					

Event name	Velocity Setting Out of Range			Event code	551D0000 hex	
Meaning	The value of *Velocity* (Velocity) member in the *MotionParams* (Motion Parameters) input variable to a robot control instruction is out of range.					
Source	Robot Control Function Module		Source details	Robot	Detection timing	At instruction execution
Error attributes	Level	Minor fault	Recovery	Error reset	Log category	System
Effects	User program	Continues.	Operation	The relevant robot stops.		
System-defined variables	Variable		Data type		Name	
	_RC_RBT[*].MFaultLvl.Active		BOOL		Robot Minor Fault Occurrence	
Cause and correction	Assumed cause		Correction		Prevention	
	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.	
Attached information	None					
Precautions/ Remarks	None					

<table>
<tr><th>Event name</th><td colspan="3">Acceleration Ratio Setting Out of Range</td><th>Event code</th><td colspan="2">551E0000 hex</td></tr>
<tr><th>Meaning</th><td colspan="6">The value of AccelerationRatio (Acceleration Ratio) member in the MotionParams (Motion Parameters) input variable to a robot control instruction is out of range.</td></tr>
<tr><th>Source</th><td colspan="2">Robot Control Function Module</td><th>Source details</th><td>Robot</td><th>Detection timing</th><td>At instruction execution</td></tr>
<tr><th>Error attributes</th><th>Level</th><td>Minor fault</td><th>Recovery</th><td>Error reset</td><th>Log category</th><td>System</td></tr>
<tr><th>Effects</th><th>User program</th><td>Continues.</td><th>Operation</th><td colspan="3">The relevant robot stops.</td></tr>
<tr><th rowspan="2">System-defined variables</th><th colspan="2">Variable</th><th colspan="2">Data type</th><th colspan="2">Name</th></tr>
<tr><td colspan="2">_RC_RBT[*].MFaultLvl.Active</td><td colspan="2">BOOL</td><td colspan="2">Robot Minor Fault Occurrence</td></tr>
<tr><th rowspan="2">Cause and correction</th><th colspan="2">Assumed cause</th><th colspan="2">Correction</th><th colspan="2">Prevention</th></tr>
<tr><td colspan="2">Instruction input parameter exceeded the valid range of the input variable.</td><td colspan="2">Correct the parameter so that the valid range of the input variable is not exceeded for the relevant instruction.</td><td colspan="2">Set the input parameter to the instruction so that the valid range of the input variable is not exceeded.</td></tr>
<tr><th>Attached information</th><td colspan="6">None</td></tr>
<tr><th>Precautions/ Remarks</th><td colspan="6">None</td></tr>
</table>

<table>
<tr><th>Event name</th><td colspan="3">Deceleration Ratio Setting Out of Range</td><th>Event code</th><td colspan="2">551F0000 hex</td></tr>
<tr><th>Meaning</th><td colspan="6">The value of DecelerationRatio (Deceleration Ratio) member in the MotionParams (Motion Parameters) input variable to a robot control instruction is out of range.</td></tr>
<tr><th>Source</th><td colspan="2">Robot Control Function Module</td><th>Source details</th><td>Robot</td><th>Detection timing</th><td>At instruction execution</td></tr>
<tr><th>Error attributes</th><th>Level</th><td>Minor fault</td><th>Recovery</th><td>Error reset</td><th>Log category</th><td>System</td></tr>
<tr><th>Effects</th><th>User program</th><td>Continues.</td><th>Operation</th><td colspan="3">The relevant robot stops.</td></tr>
<tr><th rowspan="2">System-defined variables</th><th colspan="2">Variable</th><th colspan="2">Data type</th><th colspan="2">Name</th></tr>
<tr><td colspan="2">_RC_RBT[*].MFaultLvl.Active</td><td colspan="2">BOOL</td><td colspan="2">Robot Minor Fault Occurrence</td></tr>
<tr><th rowspan="2">Cause and correction</th><th colspan="2">Assumed cause</th><th colspan="2">Correction</th><th colspan="2">Prevention</th></tr>
<tr><td colspan="2">Instruction input parameter exceeded the valid range of the input variable.</td><td colspan="2">Correct the parameter so that the valid range of the input variable is not exceeded for the relevant instruction.</td><td colspan="2">Set the input parameter to the instruction so that the valid range of the input variable is not exceeded.</td></tr>
<tr><th>Attached information</th><td colspan="6">None</td></tr>
<tr><th>Precautions/ Remarks</th><td colspan="6">None</td></tr>
</table>

Event name	Positioning Accuracy Selection Out of Range			**Event code**	55200000 hex	
Meaning	The value of *NullingTolerance* (Positioning Accuracy) member in the *MotionParams* (Motion Parameters) input variable to a robot control instruction is out of range.					
Source	Robot Control Function Module		**Source details**	Robot	**Detection timing**	At instruction execution
Error attributes	**Level**	Minor fault	**Recovery**	Error reset	**Log category**	System
Effects	**User program**	Continues.	**Operation**	The relevant robot stops.		
System-defined variables	**Variable**		**Data type**		**Name**	
	_RC_RBT[*].MFaultLvl.Active		BOOL		Robot Minor Fault Occurrence	
Cause and correction	**Assumed cause**		**Correction**		**Prevention**	
	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.	
Attached information	None					
Precautions/ Remarks	None					

Event name	Rotation Limit Selection Out of Range			**Event code**	55210000 hex	
Meaning	The value of *SingleTurn* (Rotation Limit) member in the *MotionParams* (Motion Parameters) input variable to a robot control instruction is out of range.					
Source	Robot Control Function Module		**Source details**	Robot	**Detection timing**	At instruction execution
Error attributes	**Level**	Minor fault	**Recovery**	Error reset	**Log category**	System
Effects	**User program**	Continues.	**Operation**	The relevant robot stops.		
System-defined variables	**Variable**		**Data type**		**Name**	
	_RC_RBT[*].MFaultLvl.Active		BOOL		Robot Minor Fault Occurrence	
Cause and correction	**Assumed cause**		**Correction**		**Prevention**	
	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.	
Attached information	None					
Precautions/ Remarks	None					

Event name	Buffer Mode Selection Out of Range			**Event code**	55220000 hex	
Meaning	The value of *BufferMode* (Buffer Mode Selection) input variable to a robot control instruction is out of range.					
Source	Robot Control Function Module		**Source details**	Robot	**Detection timing**	At instruction execution
Error attributes	**Level**	Minor fault	**Recovery**	Error reset	**Log category**	System
Effects	**User program**	Continues.	**Operation**	The relevant robot stops.		
System-defined variables	**Variable**		**Data type**		**Name**	
	_RC_RBT[*].MFaultLvl.Active		BOOL		Robot Minor Fault Occurrence	
Cause and correction	**Assumed cause**		**Correction**		**Prevention**	
	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.	
Attached information	None					
Precautions/ Remarks	None					

Event name	Target Position Specification Method Setting Out of Range			**Event code**	55230000 hex	
Meaning	The value of *PositionMode* (Target Position Specification Method) input variable to a robot control instruction is out of range.					
Source	Robot Control Function Module		**Source details**	Robot	**Detection timing**	At instruction execution
Error attributes	**Level**	Minor fault	**Recovery**	Error reset	**Log category**	System
Effects	**User program**	Continues.	**Operation**	The relevant robot stops.		
System-defined variables	**Variable**		**Data type**		**Name**	
	_RC_RBT[*].MFaultLvl.Active		BOOL		Robot Minor Fault Occurrence	
Cause and correction	**Assumed cause**		**Correction**		**Prevention**	
	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.	
Attached information	None					
Precautions/ Remarks	None					

Event name	Target Offset Setting Out of Range			**Event code**	552F0000 hex	
Meaning	The value of *Offset* (Target Offset) input variable to a robot control instruction is out of range.					
Source	Robot Control Function Module		**Source details**	Robot	**Detection timing**	At instruction execution
Error attributes	**Level**	Minor fault	**Recovery**	Error reset	**Log category**	System
Effects	**User program**	Continues.	**Operation**	The relevant robot stops.		
System-defined variables	**Variable**		**Data type**		**Name**	
	_RC_RBT[*].MFaultLvl.Active		BOOL		Robot Minor Fault Occurrence	
Cause and correction	**Assumed cause**		**Correction**		**Prevention**	
	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.	
Attached information	Attached information 1: Element number that is out of range in the *Offset* (Target Offset) input variable.					
Precautions/ Remarks	None					

Event name	Robot Control Instruction Executed while Robot is not Attached			**Event code**	55330000 hex	
Meaning	An instruction required that a robot has been attached was executed for the robot that has not been attached.					
Source	Robot Control Function Module		**Source details**	Robot	**Detection timing**	At instruction execution
Error attributes	**Level**	Minor fault	**Recovery**	Error reset	**Log category**	System
Effects	**User program**	Continues.	**Operation**	The relevant robot stops.		
System-defined variables	**Variable**		**Data type**		**Name**	
	_RC_RBT[*].MFaultLvl.Active		BOOL		Robot Minor Fault Occurrence	
Cause and correction	**Assumed cause**		**Correction**		**Prevention**	
	An instruction that controls a robot was executed for the robot that has not been attached.		Correct the program so that the instruction is executed while the robot has been attached.		Make sure that the robot has been attached before executing the robot control instruction. To attach the robot, change the operating mode of the robot to Auto mode and COMP mode, and then attach the robot with the RC_AttachRobot (Attach Robot) instruction.	
Attached information	None					
Precautions/ Remarks	None					

<table>
<tr><td>Event name</td><td colspan="3">Tool Coordination Transform Setting Out of Range</td><td>Event code</td><td colspan="2">55350000 hex</td></tr>
<tr><td>Meaning</td><td colspan="6">The value of ToolCoordTransform (Tool Conversion Coordinates) input variable to a robot control instruction is out of range.</td></tr>
<tr><td>Source</td><td colspan="2">Robot Control Function Module</td><td>Source details</td><td>Robot</td><td>Detection timing</td><td>At instruction execution</td></tr>
<tr><td>Error attributes</td><td>Level</td><td>Minor fault</td><td>Recovery</td><td>Error reset</td><td>Log category</td><td>System</td></tr>
<tr><td>Effects</td><td>User program</td><td>Continues.</td><td>Operation</td><td colspan="3">The relevant robot stops.</td></tr>
<tr><td rowspan="2">System-defined variables</td><td colspan="2">Variable</td><td colspan="2">Data type</td><td colspan="2">Name</td></tr>
<tr><td colspan="2">_RC_RBT[*].MFaultLvl.Active</td><td colspan="2">BOOL</td><td colspan="2">Robot Minor Fault Occurrence</td></tr>
<tr><td rowspan="2">Cause and correction</td><td colspan="2">Assumed cause</td><td colspan="2">Correction</td><td colspan="2">Prevention</td></tr>
<tr><td colspan="2">Instruction input parameter exceeded the valid range of the input variable.</td><td colspan="2">Correct the parameter so that the valid range of the input variable is not exceeded for the relevant instruction.</td><td colspan="2">Set the input parameter to the instruction so that the valid range of the input variable is not exceeded.</td></tr>
<tr><td>Attached information</td><td colspan="6">Attached information 1: Element number that is out of range in the ToolCoordTransform (Tool Conversion Coordinates) input variable.</td></tr>
<tr><td>Precautions/ Remarks</td><td colspan="6">None</td></tr>
</table>

<table>
<tr><td>Event name</td><td colspan="3">Robot Control Instruction Multi-execution Disabled</td><td>Event code</td><td colspan="2">55360000 hex</td></tr>
<tr><td>Meaning</td><td colspan="6">Multiple robot control instructions that cannot be executed simultaneously were executed.</td></tr>
<tr><td>Source</td><td colspan="2">Robot Control Function Module</td><td>Source details</td><td>Robot control common or robot</td><td>Detection timing</td><td>At instruction execution</td></tr>
<tr><td>Error attributes</td><td>Level</td><td>Minor fault</td><td>Recovery</td><td>Error reset</td><td>Log category</td><td>System</td></tr>
<tr><td>Effects</td><td>User program</td><td>Continues.</td><td>Operation</td><td colspan="3">The relevant robot stops.
Not affected when this event occurs from the robot control common.</td></tr>
<tr><td rowspan="3">System-defined variables</td><td colspan="2">Variable</td><td colspan="2">Data type</td><td colspan="2">Name</td></tr>
<tr><td colspan="2">_RC_COM.MFaultLvl.Active</td><td colspan="2">BOOL</td><td colspan="2">Robot Control Common Minor Fault Occurrence</td></tr>
<tr><td colspan="2">_RC_RBT[*].MFaultLvl.Active</td><td colspan="2">BOOL</td><td colspan="2">Robot Minor Fault Occurrence</td></tr>
<tr><td rowspan="2">Cause and correction</td><td colspan="2">Assumed cause</td><td colspan="2">Correction</td><td colspan="2">Prevention</td></tr>
<tr><td colspan="2">Multiple robot control instructions that cannot be executed simultaneously were executed.</td><td colspan="2">Check the specifications of multi-execution of instructions for relevant instruction and correct the program so that instructions that cannot be executed at the same time are not executed simultaneously.</td><td colspan="2">Check the specifications for multi-execution of instructions for the instruction and do not execute instructions that cannot be executed at the same time.</td></tr>
<tr><td>Attached information</td><td colspan="6">None</td></tr>
<tr><td>Precautions/ Remarks</td><td colspan="6">None</td></tr>
</table>

<table>
<tr><td>Event name</td><td colspan="3">Robot Control Instruction Multi-execution Buffer Limit Exceeded</td><td>Event code</td><td colspan="2">553C0000 hex</td></tr>
<tr><td>Meaning</td><td colspan="6">The number of multi-execution for the robot control instructions exceeded the upper limit.</td></tr>
<tr><td>Source</td><td colspan="2">Robot Control Function Module</td><td>Source details</td><td>Robot</td><td>Detection timing</td><td>At instruction execution</td></tr>
<tr><td>Error attributes</td><td>Level</td><td>Minor fault</td><td>Recovery</td><td>Error reset</td><td>Log category</td><td>System</td></tr>
<tr><td>Effects</td><td>User program</td><td>Continues.</td><td>Operation</td><td colspan="3">The relevant robot stops.</td></tr>
<tr><td rowspan="2">System-defined variables</td><td colspan="2">Variable</td><td colspan="2">Data type</td><td colspan="2">Name</td></tr>
<tr><td colspan="2">_RC_RBT[*].MFaultLvl.Active</td><td colspan="2">BOOL</td><td colspan="2">Robot Minor Fault Occurrence</td></tr>
<tr><td rowspan="2">Cause and correction</td><td colspan="2">Assumed cause</td><td colspan="2">Correction</td><td colspan="2">Prevention</td></tr>
<tr><td colspan="2">The total number of current robot control instructions and buffered robot control instructions exceeded eight.</td><td colspan="2">Correct the program so that the number of multi-execution for the robot control instructions does not exceed the upper limit.</td><td colspan="2">Make sure that the total number of current robot control instructions and buffered robot control instructions does not exceed eight.</td></tr>
<tr><td>Attached information</td><td colspan="6">None</td></tr>
<tr><td>Precautions/ Remarks</td><td colspan="6">None</td></tr>
</table>

<table>
<tr><td>Event name</td><td colspan="3">Robot Control Instruction Executed with Calibration Not Completed</td><td>Event code</td><td colspan="2">553D0000 hex</td></tr>
<tr><td>Meaning</td><td colspan="6">An instruction that is required for the calibration completion was executed for a robot whose calibration was not completed.</td></tr>
<tr><td>Source</td><td colspan="2">Robot Control Function Module</td><td>Source details</td><td>Robot</td><td>Detection timing</td><td>At instruction execution</td></tr>
<tr><td>Error attributes</td><td>Level</td><td>Minor fault</td><td>Recovery</td><td>Error reset</td><td>Log category</td><td>System</td></tr>
<tr><td>Effects</td><td>User program</td><td>Continues.</td><td>Operation</td><td colspan="3">The relevant robot stops.</td></tr>
<tr><td rowspan="2">System-defined variables</td><td colspan="2">Variable</td><td colspan="2">Data type</td><td colspan="2">Name</td></tr>
<tr><td colspan="2">_RC_RBT[*].MFaultLvl.Active</td><td colspan="2">BOOL</td><td colspan="2">Robot Minor Fault Occurrence</td></tr>
<tr><td rowspan="3">Cause and correction</td><td colspan="2">Assumed cause</td><td colspan="2">Correction</td><td colspan="2">Prevention</td></tr>
<tr><td colspan="2">An instruction that controls a robot was executed for the robot that the calibration has not been completed.</td><td colspan="2" rowspan="2">Correct the program so that the relevant instruction is executed after the RC_Calibrate (Robot Calibration) instruction is executed and the calibration is completed.</td><td colspan="2" rowspan="2">If the DrvStatus.Calibrated member in the robot variable is FALSE, execute the RC_Calibrate (Robot Calibration) instruction.</td></tr>
<tr><td colspan="2">An instruction to synchronize the master machine and the robot was executed for a robot whose calibration was not completed.</td></tr>
<tr><td>Attached information</td><td colspan="6">None</td></tr>
<tr><td>Precautions/ Remarks</td><td colspan="6">None</td></tr>
</table>

Event name	Robot Control Instruction Executed while Robot High Power is OFF		**Event code**	553E0000 hex		
Meaning	An instruction required for the robot in a Power Enabled state was executed for the robot in which high power turns OFF.					
Source	Robot Control Function Module		**Source details**	Robot	**Detection timing**	At instruction execution
Error attributes	**Level**	Minor fault	**Recovery**	Error reset	**Log category**	System
Effects	**User program**	Continues.	**Operation**	The relevant robot stops.		
System-defined variables	**Variable**		**Data type**		**Name**	
	_RC_RBT[*].MFaultLvl.Active		BOOL		Robot Minor Fault Occurrence	
Cause and correction	**Assumed cause**		**Correction**		**Prevention**	
	An instruction that controls a robot was executed for the robot in which high power turns OFF.		Turn the robot high power ON.		Check that the *DrvStatus.PowerEnabled* member in the robot variable is TRUE before executing the robot control instruction.	
Attached information	None					
Precautions/ Remarks	None					

Event name	Robot Already Attached		**Event code**	553F0000 hex		
Meaning	An attempt was made to attach a robot again or execute calibration for the robot that was already attached.					
Source	Robot Control Function Module		**Source details**	Robot	**Detection timing**	At instruction execution
Error attributes	**Level**	Minor fault	**Recovery**	Error reset	**Log category**	System
Effects	**User program**	Continues.	**Operation**	The relevant robot stops.		
System-defined variables	**Variable**		**Data type**		**Name**	
	_RC_RBT[*].MFaultLvl.Active		BOOL		Robot Minor Fault Occurrence	
Cause and correction	**Assumed cause**		**Correction**		**Prevention**	
	The target robot was already attached in the sequence control program.		Correct the program so that the relevant instruction is executed if the robot has not already been attached.		Make sure that the RC_AttachRobot (Attach Robot) instruction or the RC_Calibrate (Robot Calibration) instruction is executed if the robot has not already been attached.	
Attached information	None					
Precautions/ Remarks	None					

<table>
<tr><td>Event name</td><td colspan="3">Robot Control Instruction Executed while Robot is MANUAL Mode or is not COMP Mode</td><td>Event code</td><td colspan="2">55400000 hex</td></tr>
<tr><td>Meaning</td><td colspan="6">A robot control instruction for which the robot is MANUAL mode or is not COMP mode was executed.</td></tr>
<tr><td>Source</td><td colspan="2">Robot Control Function Module</td><td>Source details</td><td>Robot</td><td>Detection timing</td><td>At instruction execution</td></tr>
<tr><td>Error attributes</td><td>Level</td><td>Minor fault</td><td>Recovery</td><td>Error reset</td><td>Log category</td><td>System</td></tr>
<tr><td>Effects</td><td>User program</td><td>Continues.</td><td>Operation</td><td colspan="3">The relevant robot stops.</td></tr>
<tr><td rowspan="2">System-defined variables</td><td colspan="2">Variable</td><td colspan="2">Data type</td><td colspan="2">Name</td></tr>
<tr><td colspan="2">_RC_RBT[*].MFaultLvl.Active</td><td colspan="2">BOOL</td><td colspan="2">Robot Minor Fault Occurrence</td></tr>
<tr><td rowspan="3">Cause and correction</td><td colspan="2">Assumed cause</td><td colspan="2">Correction</td><td colspan="2">Prevention</td></tr>
<tr><td colspan="2">The robot that you control is MANUAL mode.</td><td colspan="2">Change the robot that you control to Auto mode on the front panel.</td><td colspan="2" rowspan="2">Check that the DrvStatus.Manual member in the robot variable is FALSE and the DrvStatus.Comp member in the robot variable is TRUE before executing the robot control instruction.</td></tr>
<tr><td colspan="2">The robot in Auto mode that you control is not COMP mode.</td><td colspan="2">Change the robot in Auto mode that you control to COMP mode with a teaching pendant or the Support Software.</td></tr>
<tr><td>Attached information</td><td colspan="6">None</td></tr>
<tr><td>Precautions/ Remarks</td><td colspan="6">None</td></tr>
</table>

<table>
<tr><td>Event name</td><td colspan="3">Cannot Execute Robot Control Instruction</td><td>Event code</td><td colspan="2">55440000 hex</td></tr>
<tr><td>Meaning</td><td colspan="6">The Robot Control Function Module is not running.</td></tr>
<tr><td>Source</td><td colspan="2">Robot Control Function Module</td><td>Source details</td><td>Robot control common or robot</td><td>Detection timing</td><td>At instruction execution</td></tr>
<tr><td>Error attributes</td><td>Level</td><td>Minor fault</td><td>Recovery</td><td>Error reset</td><td>Log category</td><td>System</td></tr>
<tr><td>Effects</td><td>User program</td><td>Continues.</td><td>Operation</td><td colspan="3">The relevant robot stops.</td></tr>
<tr><td rowspan="3">System-defined variables</td><td colspan="2">Variable</td><td colspan="2">Data type</td><td colspan="2">Name</td></tr>
<tr><td colspan="2">_RC_COM.MFaultLvl.Active</td><td colspan="2">BOOL</td><td colspan="2">Robot Control Common Minor Fault Occurrence</td></tr>
<tr><td colspan="2">_RC_RBT[*].MFaultLvl.Active</td><td colspan="2">BOOL</td><td colspan="2">Robot Minor Fault Occurrence</td></tr>
<tr><td rowspan="2">Cause and correction</td><td colspan="2">Assumed cause</td><td colspan="2">Correction</td><td colspan="2">Prevention</td></tr>
<tr><td colspan="2">The robot control instruction was executed while the Robot Control Function Module was not running.</td><td colspan="2">Correct the program so that the relevant instruction is executed after waiting for _RC_COM.Status.RunMode or _RC_RBT[*].DrvStatus.RunMode to change to TRUE.</td><td colspan="2">Make sure that the robot control instruction is executed after waiting for _RC_COM.Status.RunMode or _RC_RBT[*].DrvStatus.RunMode to change to TRUE.</td></tr>
<tr><td>Attached information</td><td colspan="6">None</td></tr>
<tr><td>Precautions/ Remarks</td><td colspan="6">None</td></tr>
</table>

Event name	Illegal Program Name Specification			**Event code**	55480000 hex	
Meaning	The program name specified for the *PrgName* (Program Name) input variable to a robot control instruction is incorrect.					
Source	Robot Control Function Module		**Source details**	Robot control common	**Detection timing**	At instruction execution
Error attributes	**Level**	Minor fault	**Recovery**	Error reset	**Log category**	System
Effects	**User program**	Continues.	**Operation**	Not affected.		
System-defined variables	**Variable**		**Data type**		**Name**	
	_RC_COM.MFaultLvl.Active		BOOL		Robot Control Common Minor Fault Occurrence	
Cause and correction	**Assumed cause**		**Correction**		**Prevention**	
	The length of the program name specified for the *PrgName* (Program Name) input variable to the instruction was outside of the valid range.		Correct the length of text strings of the program name that is specified in the *PrgName* (Program Name) input variable to the instruction within the valid range.		Make sure that the length of text strings of the program name that is specified in the *PrgName* (Program Name) input variable to the instruction is in the valid range.	
Attached information	None					
Precautions/ Remarks	None					

Event name	Unsupported Function Executed			**Event code**	554A0000 hex	
Meaning	The function that is not supported was executed.					
Source	Robot Control Function Module		**Source details**	Robot	**Detection timing**	At instruction execution
Error attributes	**Level**	Minor fault	**Recovery**	Error reset	**Log category**	System
Effects	**User program**	Continues.	**Operation**	Not affected.		
System-defined variables	**Variable**		**Data type**		**Name**	
	_RC_RBT[*].MFaultLvl.Active		BOOL		Robot Minor Fault Occurrence	
Cause and correction	**Assumed cause**		**Correction**		**Prevention**	
	The function that is not supported by the robot was executed.		Check the version of the robot.		Use the robot with version that supports the function.	
Attached information	None					
Precautions/ Remarks	None					

Event name	Illegal IP Address Specification			Event code	554B0000 hex	
Meaning	The value of *IPAddr* input variable to a robot control instruction is not correct.					
Source	Robot Control Function Module		Source details	Robot control common	Detection timing	At instruction execution
Error attributes	Level	Minor fault	Recovery	Error reset	Log category	System
Effects	User program	Continues.	Operation	Not affected.		
System-defined variables	Variable		Data type		Name	
	_RC_COM.MFaultLvl.Active		BOOL		Robot Control Common Minor Fault Occurrence	
Cause and correction	Assumed cause		Correction		Prevention	
	The length of the parameter list specified for the *IPAddr* input variable to an instruction was outside of the valid range.		Correct the length of the string in the parameter list specified for the *IPAddr* input variable to the instruction so that the length of the string is in the valid range.		Correct the length of the string in the parameter list specified for the *IPAddr* input variable to the instruction so that the length of the string is in the valid range.	
Attached information	None					
Precautions/ Remarks	None					

Event name	Vision Error Occurred			Event code	554C0000 hex	
Meaning	An error occurred during execution of the vision function in the IPC Application Controller from a sequence control program.					
Source	Robot Control Function Module		Source details	Robot control common	Detection timing	Continuously
Error attributes	Level	Minor fault	Recovery	Error reset	Log category	System
Effects	User program	Continues.	Operation	Not affected.		
System-defined variables	Variable		Data type		Name	
	_RC_COM.MFaultLvl.Active		BOOL		Robot Control Common Minor Fault Occurrence	
Cause and correction	Assumed cause		Correction		Prevention	
	An error occurred during execution of the vision function in the IPC Application Controller.		Identify the cause of error from the attached information, and correct the sequence control program.		Write the sequence control program so that the error does not occur.	
Attached information	Attached information 1: eV+ event number					
Precautions/ Remarks	Refer to the *eV+3 User's Manual (Cat. No. I651)* for information on the eV+ event number.					

Event name	Robot Control Common Error Occurred			Event code	75010000 hex	
Meaning	A robot control common error occurred.					
Source	Robot Control Function Module		Source details	Robot	Detection timing	Continuously
Error attributes	Level	Minor fault	Recovery	Error reset	Log category	System
Effects	User program	Continues.	Operation	The relevant robot stops.		
System-defined variables	Variable		Data type		Name	
	_RC_RBT[*].MFaultLvl.Active		BOOL		Robot Minor Fault Occurrence	
Cause and correction	Assumed cause		Correction		Prevention	
	A partial fault level robot control common error occurred.		Check the robot control common error that occurred and remove the cause of the error.		None	
Attached information	None					
Precautions/ Remarks	When a partial fault level robot control common error occurs, the robot and V+ task do not operate.					

Event name	EtherCAT Slave Disconnection Error			Event code	75020000 hex	
Meaning	One of the following occurred for the EtherCAT slave that is allocated to a robot. • Disconnect or replace the slave. • Disable the slave.					
Source	Robot Control Function Module		Source details	Robot	Detection timing	Continuously
Error attributes	Level	Minor fault	Recovery	Error reset	Log category	System
Effects	User program	Continues.	Operation	The relevant robot stops.		
System-defined variables	Variable		Data type		Name	
	_RC_RBT[*].MFaultLvl.Active		BOOL		Robot Minor Fault Occurrence	
Cause and correction	Assumed cause		Correction		Prevention	
	One of the following occurred for the EtherCAT slave that is allocated to a robot. • Disconnection or replacement • Disablement		Reconnect or enable the EtherCAT slave that is allocated to the relevant robot without cycling the power supply to the EtherCAT slave.		None	
Attached information	None					
Precautions/ Remarks	None					

Event name	Robot Error Occurred			Event code	75030000 hex	
Meaning	An error occurred in the robot that the robot control instruction execution is in progress.					
Source	Robot Control Function Module		Source details	Robot	Detection timing	Continuously
Error attributes	Level	Minor fault	Recovery	Error reset	Log category	System
Effects	User program	Continues.	Operation	The relevant robot stops.		
System-defined variables	Variable		Data type		Name	
	_RC_RBT[*].MFaultLvl.Active		BOOL		Robot Minor Fault Occurrence	
Cause and correction	Assumed cause		Correction		Prevention	
	An error occurred in the robot that the robot control instruction execution is in progress.		Identify the cause of error in the robot from the attached information, and correct the sequence control program.		Write the sequence control program so that the error does not occur.	
Attached information	Attached information 1: eV+ event number Attached information 2: System information					
Precautions/ Remarks	Refer to the *eV+3 User's Manual (Cat. No. I651)* for information on the eV+ event number.					

Event name	Robot Control Period Exceeded			Event code	75040000 hex	
Meaning	The robot control processing failures occurred two consecutive times during task period of primary periodic task.					
Source	Robot Control Function Module		Source details	Robot	Detection timing	Continuously
Error attributes	Level	Minor fault	Recovery	Error reset	Log category	System
Effects	User program	Continues.	Operation	The relevant robot stops.		
System-defined variables	Variable		Data type		Name	
	_RC_RBT[*].MFaultLvl.Active		BOOL		Robot Minor Fault Occurrence	
Cause and correction	Assumed cause		Correction		Prevention	
	The task period of primary periodic task is too short for the amount of the user program that is executed in the primary periodic task.		Check the task period of primary periodic task in the **Task Period Monitor** on the Sysmac Studio, and set the task period of the primary periodic task to be long enough to complete all processing.		Set the task period of the primary periodic task to be long enough to complete all processing.	
	Too many robot control instructions are executed for the task period of primary periodic task.		Decrease the robot control instructions that are executed.		Assign the user program that is executed in a certain period to the primary periodic task.	
Attached information	None					
Precautions/ Remarks	For the CPU Unit with unit version 1.43 or later, this event will occur instead of 75000000 hex.					

Event name	EtherCAT Slave Communications Error			**Event code**	85800000 hex	
Meaning	A communications error occurred for the EtherCAT slave that is allocated to a robot.					
Source	Robot Control Function Module		**Source details**	Robot	**Detection timing**	Continuously
Error attributes	**Level**	Minor fault	**Recovery**	Error reset	**Log category**	System
Effects	**User program**	Continues.	**Operation**	The relevant robot stops.		
System-defined variables	**Variable**		**Data type**		**Name**	
	_RC_RBT[*].MFaultLvl.Active		BOOL		Robot Minor Fault Occurrence	
Cause and correction	**Assumed cause**		**Correction**		**Prevention**	
	A communications error occurred for the EtherCAT slave that is allocated to a robot.		Check the event log for the error that occurred in the EtherCAT Master Function Module. Remove the cause of the error and clear the relevant error.		None	
Attached information	None					
Precautions/ Remarks	Even if this error is reset, the error in the slave allocated to a robot is not reset.					

Event name	V+ Program Error			**Event code**	96040000 hex	
Meaning	An error occurred in the V+ program.					
Source	Robot Control Function Module		**Source details**	Robot control common	**Detection timing**	Continuously
Error attributes	**Level**	Information	**Recovery**	---	**Log category**	System
Effects	**User program**	Continues.	**Operation**	The relevant V+ task is stopped. The relevant robot stops.		
System-defined variables	**Variable**		**Data type**		**Name**	
	---		---		---	
Cause and correction	**Assumed cause**		**Correction**		**Prevention**	
	An error occurred in the V+ program that was being executed.		Identify the cause of error in the V+ program from the attached information, and correct the V+ program.		Write the V+ program so that the error does not occur.	
Attached information	Attached information 1: eV+ event number Attached information 2: Event Sources Attached Information 3: V+ task number where an error occurred Attached information 4: Robot number where an error occurred					
Precautions/ Remarks	Refer to the *eV+3 User's Manual (Cat. No. I651)* for information on the eV+ event number.					

Event name	V+ Program Warning			Event code	96050000 hex	
Meaning	The V+ program issued a warning message.					
Source	Robot Control Function Module		Source details	Robot control common	Detection timing	Continuously
Error attributes	Level	Information	Recovery	---	Log category	System
Effects	User program	Continues.	Operation	Not affected.		
System-defined variables	Variable		Data type		Name	
	---		---		---	
Cause and correction	Assumed cause		Correction		Prevention	
	The V+ program that was being executed issued a warning message.		Identify the cause of warning in the V+ program from the attached information, and correct the V + program if necessary.		Write the V+ program so that the warning is not detected.	
Attached information	Attached information 1: eV+ event number Attached information 2: Event Sources Attached Information 3: V+ task number where an error occurred Attached information 4: Robot number where an error occurred					
Precautions/ Remarks	Refer to the *eV+3 User's Manual (Cat. No. I651)* for information on the eV+ event number.					

Event name	V+ Program Information			Event code	96060000 hex	
Meaning	The V+ program issued an information message.					
Source	Robot Control Function Module		Source details	Robot control common	Detection timing	Continuously
Error attributes	Level	Information	Recovery	---	Log category	System
Effects	User program	Continues.	Operation	Not affected.		
System-defined variables	Variable		Data type		Name	
	---		---		---	
Cause and correction	Assumed cause		Correction		Prevention	
	The V+ program that was being executed issued an information message.		None		None	
Attached information	Attached information 1: eV+ event number Attached information 2: Event Sources Attached Information 3: V+ task number where an error occurred Attached information 4: Robot number where an error occurred					
Precautions/ Remarks	Refer to the *eV+3 User's Manual (Cat. No. I651)* for information on the eV+ event number.					

Event name	Robot Manual Mode Started			**Event code**	96090000 hex	
Meaning	The robot was set to Manual mode.					
Source	Robot Control Function Module		**Source details**	Robot	**Detection timing**	Continuously
Error attributes	**Level**	Information	**Recovery**	---	**Log category**	Access
Effects	**User program**	Continues.	**Operation**	The relevant robot stops.		
System-defined variables	**Variable**		**Data type**		**Name**	
	---		---		---	
Cause and correction	**Assumed cause**		**Correction**		**Prevention**	
	The robot was set to Manual mode.		None		None	
Attached information	None					
Precautions/ Remarks	None					

Event name	Robot Auto Mode Started			**Event code**	960A0000 hex	
Meaning	The robot was set to Auto mode.					
Source	Robot Control Function Module		**Source details**	Robot	**Detection timing**	Continuously
Error attributes	**Level**	Information	**Recovery**	---	**Log category**	Access
Effects	**User program**	Continues.	**Operation**	The relevant robot stops.		
System-defined variables	**Variable**		**Data type**		**Name**	
	---		---		---	
Cause and correction	**Assumed cause**		**Correction**		**Prevention**	
	The robot was set to Auto mode.		None		None	
Attached information	None					
Precautions/ Remarks	None					

Errors Related to Robot Control Instructions

This section describes the details of the errors related to robot control instructions for the RC Function Module.

Event name	Robot Control Instruction Re-execution Disabled			**Event code**	54015510 hex	
Meaning	An attempt was made to re-execute a robot control instruction that cannot be re-executed.					
Source	PLC Function Module		**Source details**	Instruction	**Detection timing**	At instruction execution
Error attributes	**Level**	Observation	**Recovery**	---	**Log category**	System
Effects	**User program**	Continues.	**Operation**	The relevant instruction will end according to specifications.		
System-defined variables	**Variable**		**Data type**		**Name**	
	_RC_COM.MFaultLvl.Active		BOOL		Robot Control Common Minor Fault Occurrence	
	_RC_RBT[*].MFaultLvl.Active		BOOL		Robot Minor Fault Occurrence	
Cause and correction	**Assumed cause**		**Correction**		**Prevention**	
	The *Execute* (Execute) input variable was re-executed during execution of the robot control instruction that has the *Execute* (Execute) input variable.		Correct the program so that the *Execute* (Execute) input variable does not change to TRUE until the *Busy* (Executing) output variable to the instruction changes to FALSE.		When using instructions that cannot be re-executed, include a condition for the *Execute* (Execute) input variable so that it does not change to TRUE unless the *Busy* (Executing) output variable for the instruction to be used is FALSE. Or, stop the instruction before executing it again.	
Attached information	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 (ErrorIDEx)					
Precautions/ Remarks	• The robot control instruction cannot change the current operation by the re-execution. • If a program is changed after an error occurs, the attached information that is displayed may not be correct.					

Event name	V+ Task Number Setting Out of Range			Event code	54015511 hex	
Meaning	The value of *TaskNo* (Task Number) input variable to a robot control instruction is out of range.					
Source	PLC Function Module		Source details	Instruction	Detection timing	At instruction execution
Error attributes	Level	Observation	Recovery	---	Log category	System
Effects	User program	Continues.	Operation	The relevant instruction will end according to specifications.		
System-defined variables	Variable		Data type		Name	
	_RC_COM.MFaultLvl.Active		BOOL		Robot Control Common Minor Fault Occurrence	
Cause and correction	Assumed cause		Correction		Prevention	
	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.	
Attached information	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 (ErrorIDEx)					
Precautions/ Remarks	If a program is changed after an error occurs, the attached information that is displayed may not be correct.					

<table>
<tr><td>Event name</td><td colspan="3">Illegal Robot Specification</td><td>Event code</td><td colspan="2">54015512 hex</td></tr>
<tr><td>Meaning</td><td colspan="6">The robot specified for the Robot (Robot) in-out variable to a robot control instruction does not exist.</td></tr>
<tr><td>Source</td><td colspan="2">PLC Function Module</td><td>Source details</td><td>Instruction</td><td>Detection timing</td><td>At instruction execution</td></tr>
<tr><td>Error attributes</td><td>Level</td><td>Observation</td><td>Recovery</td><td>---</td><td>Log category</td><td>System</td></tr>
<tr><td>Effects</td><td>User program</td><td>Continues.</td><td>Operation</td><td colspan="3">The relevant instruction will end according to specifications.</td></tr>
<tr><td rowspan="2">System-defined variables</td><td colspan="2">Variable</td><td colspan="2">Data type</td><td colspan="2">Name</td></tr>
<tr><td colspan="2">_RC_COM.MFaultLvl.Active</td><td colspan="2">BOOL</td><td colspan="2">Robot Control Common Minor Fault Occurrence</td></tr>
<tr><td rowspan="2">Cause and correction</td><td colspan="2">Assumed cause</td><td colspan="2">Correction</td><td colspan="2">Prevention</td></tr>
<tr><td colspan="2">The value of variable that is used for the subscript for array of _RC_RBT[ ] robot variable specified for the Robot (Robot) in-out variable to an instruction, is specified to the robot that does not exist.</td><td colspan="2">Correct the value of variable so that existing robot is specified.</td><td colspan="2">If a variables is used for an input parameter to the instruction, check that the value of variable is correct.</td></tr>
<tr><td>Attached information</td><td colspan="6">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 (ErrorIDEx)</td></tr>
<tr><td>Precautions/ Remarks</td><td colspan="6">If a program is changed after an error occurs, the attached information that is displayed may not be correct.</td></tr>
</table>

Event name	Illegal Parameter List Specification			**Event code**	54015513 hex	
Meaning	The value of *PrgParam* (Parameter List) input variable to a robot control instruction is not correct.					
Source	PLC Function Module		**Source details**	Instruction	**Detection timing**	At instruction execution
Error attributes	**Level**	Observation	**Recovery**	---	**Log category**	System
Effects	**User program**	Continues.	**Operation**	The relevant instruction will end according to specifications.		
System-defined variables	**Variable**		**Data type**		**Name**	
	_RC_COM.MFaultLvl.Active		BOOL		Robot Control Common Minor Fault Occurrence	
Cause and correction	**Assumed cause**		**Correction**		**Prevention**	
	The length of the parameter list specified for the *PrgParam* (Parameter List) input variable to an instruction was outside of the valid range.		Correct the length of the string in the parameter list specified for the *PrgParam* (Parameter List) input variable to the instruction so that the length of the string is in the valid range.		Make sure that the length of the string in the parameter list specified for the *PrgParam* (Parameter List) input variable to the instruction is in the valid range.	
Attached information	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 (ErrorIDEx)					
Precautions/ Remarks	If a program is changed after an error occurs, the attached information that is displayed may not be correct.					

Event name	Starting Step Setting Out of Range			Event code	54015514 hex	
Meaning	The value specified in the *StartStep* (Start Step) input variable to a robot control instruction is out of range.					
Source	PLC Function Module		Source details	Instruction	Detection timing	At instruction execution
Error attributes	Level	Observation	Recovery	---	Log category	System
Effects	User program	Continues.	Operation	The relevant instruction will end according to specifications.		
System-defined variables	Variable		Data type		Name	
	_RC_COM.MFaultLvl.Active		BOOL		Robot Control Common Minor Fault Occurrence	
Cause and correction	Assumed cause		Correction		Prevention	
	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.	
Attached information	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 (ErrorIDEx)					
Precautions/ Remarks	If a program is changed after an error occurs, the attached information that is displayed may not be correct.					

Event name	Target Position Setting Out of Range			Event code	54015515 hex	
Meaning	The value of *Position* (Target Position) input variable to a robot control instruction is out of range.					
Source	PLC Function Module		Source details	Instruction	Detection timing	At instruction execution
Error attributes	Level	Observation	Recovery	---	Log category	System
Effects	User program	Continues.	Operation	The relevant instruction will end according to specifications.		
System-defined variables	Variable		Data type		Name	
	_RC_RBT[*].MFaultLvl.Active		BOOL		Robot Minor Fault Occurrence	
Cause and correction	Assumed cause		Correction		Prevention	
	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.	
Attached information	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 (ErrorIDEx)					
Precautions/ Remarks	If a program is changed after an error occurs, the attached information that is displayed may not be correct.					

<table>
<tr><td>Event name</td><td colspan="3">Lefty and Righty Setting Out of Range</td><td>Event code</td><td colspan="2">54015516 hex</td></tr>
<tr><td>Meaning</td><td colspan="6">The value of LeftyRighty (Lefty/Righty Setting) member in the ArmConfig (Arm Configuration) input variable to a robot control instruction is out of range.</td></tr>
<tr><td>Source</td><td colspan="2">PLC Function Module</td><td>Source details</td><td>Instruction</td><td>Detection timing</td><td>At instruction execution</td></tr>
<tr><td>Error attributes</td><td>Level</td><td>Observation</td><td>Recovery</td><td>---</td><td>Log category</td><td>System</td></tr>
<tr><td>Effects</td><td>User program</td><td>Continues.</td><td>Operation</td><td colspan="3">The relevant instruction will end according to specifications.</td></tr>
<tr><td rowspan="2">System-defined variables</td><td colspan="2">Variable</td><td colspan="2">Data type</td><td colspan="2">Name</td></tr>
<tr><td colspan="2">_RC_RBT[*].MFaultLvl.Active</td><td colspan="2">BOOL</td><td colspan="2">Robot Minor Fault Occurrence</td></tr>
<tr><td rowspan="2">Cause and correction</td><td colspan="2">Assumed cause</td><td colspan="2">Correction</td><td colspan="2">Prevention</td></tr>
<tr><td colspan="2">Instruction input parameter exceeded the valid range of the input variable.</td><td colspan="2">Correct the parameter so that the valid range of the input variable is not exceeded for the relevant instruction.</td><td colspan="2">Set the input parameter to the instruction so that the valid range of the input variable is not exceeded.</td></tr>
<tr><td>Attached information</td><td colspan="6">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 (ErrorIDEx)</td></tr>
<tr><td>Precautions/ Remarks</td><td colspan="6">If a program is changed after an error occurs, the attached information that is displayed may not be correct.</td></tr>
</table>

<table>
<tr><td>Event name</td><td colspan="3">Above and Below Setting Out of Range</td><td>Event code</td><td colspan="2">54015517 hex</td></tr>
<tr><td>Meaning</td><td colspan="6">The value of AboveBelow (Above/Below Setting) member in the ArmConfig (Arm Configuration) input variable to a robot control instruction is out of range.</td></tr>
<tr><td>Source</td><td colspan="2">PLC Function Module</td><td>Source details</td><td>Instruction</td><td>Detection timing</td><td>At instruction execution</td></tr>
<tr><td>Error attributes</td><td>Level</td><td>Observation</td><td>Recovery</td><td>---</td><td>Log category</td><td>System</td></tr>
<tr><td>Effects</td><td>User program</td><td>Continues.</td><td>Operation</td><td colspan="3">The relevant instruction will end according to specifications.</td></tr>
<tr><td rowspan="2">System-defined variables</td><td colspan="2">Variable</td><td colspan="2">Data type</td><td colspan="2">Name</td></tr>
<tr><td colspan="2">_RC_RBT[*].MFaultLvl.Active</td><td colspan="2">BOOL</td><td colspan="2">Robot Minor Fault Occurrence</td></tr>
<tr><td rowspan="2">Cause and correction</td><td colspan="2">Assumed cause</td><td colspan="2">Correction</td><td colspan="2">Prevention</td></tr>
<tr><td colspan="2">Instruction input parameter exceeded the valid range of the input variable.</td><td colspan="2">Correct the parameter so that the valid range of the input variable is not exceeded for the relevant instruction.</td><td colspan="2">Set the input parameter to the instruction so that the valid range of the input variable is not exceeded.</td></tr>
<tr><td>Attached information</td><td colspan="6">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 (ErrorIDEx)</td></tr>
<tr><td>Precautions/ Remarks</td><td colspan="6">If a program is changed after an error occurs, the attached information that is displayed may not be correct.</td></tr>
</table>

Event name	Flip Setting Out of Range			**Event code**	54015518 hex	
Meaning	The value of *Flip* (Flip Setting) member in the *ArmConfig* (Arm Configuration) input variable to a robot control instruction is out of range.					
Source	PLC Function Module		**Source details**	Instruction	**Detection timing**	At instruction execution
Error attributes	**Level**	Observation	**Recovery**	---	**Log category**	System
Effects	**User program**	Continues.	**Operation**	The relevant instruction will end according to specifications.		
System-defined variables	**Variable**		**Data type**		**Name**	
	_RC_RBT[*].MFaultLvl.Active		BOOL		Robot Minor Fault Occurrence	
Cause and correction	**Assumed cause**		**Correction**		**Prevention**	
	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.	
Attached information	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 (ErrorIDEx)					
Precautions/ Remarks	If a program is changed after an error occurs, the attached information that is displayed may not be correct.					

Event name	Velocity Profile Selection Out of Range			**Event code**	54015519 hex	
Meaning	The value of *VelocityProfile* (Velocity Profile) member in the *MotionParams* (Motion Parameters) input variable to a robot control instruction is out of range.					
Source	PLC Function Module		**Source details**	Instruction	**Detection timing**	At instruction execution
Error attributes	**Level**	Observation	**Recovery**	---	**Log category**	System
Effects	**User program**	Continues.	**Operation**	The relevant instruction will end according to specifications.		
System-defined variables	**Variable**		**Data type**		**Name**	
	_RC_RBT[*].MFaultLvl.Active		BOOL		Robot Minor Fault Occurrence	
Cause and correction	**Assumed cause**		**Correction**		**Prevention**	
	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.	
Attached information	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 (ErrorIDEx)					
Precautions/ Remarks	If a program is changed after an error occurs, the attached information that is displayed may not be correct.					

Event name	Velocity Mode Selection Out of Range			**Event code**	5401551A hex	
Meaning	The value of *VelocityMode* (Velocity Selection) member in the *MotionParams* (Motion Parameters) input variable to a robot control instruction is out of range.					
Source	PLC Function Module		**Source details**	Instruction	**Detection timing**	At instruction execution
Error attributes	**Level**	Observation	**Recovery**	---	**Log category**	System
Effects	**User program**	Continues.	**Operation**	The relevant instruction will end according to specifications.		
System-defined variables	**Variable**		**Data type**		**Name**	
	_RC_RBT[*].MFaultLvl.Active		BOOL		Robot Minor Fault Occurrence	
Cause and correction	**Assumed cause**		**Correction**		**Prevention**	
	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.	
Attached information	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 (ErrorIDEx)					
Precautions/ Remarks	If a program is changed after an error occurs, the attached information that is displayed may not be correct.					

Event name	Velocity Ratio Setting Out of Range			**Event code**	5401551B hex	
Meaning	The value of *VelocityRatio* (Velocity Ratio) member in the *MotionParams* (Motion Parameters) input variable to a robot control instruction is out of range.					
Source	PLC Function Module		**Source details**	Instruction	**Detection timing**	At instruction execution
Error attributes	**Level**	Observation	**Recovery**	---	**Log category**	System
Effects	**User program**	Continues.	**Operation**	The relevant instruction will end according to specifications.		
System-defined variables	**Variable**		**Data type**		**Name**	
	_RC_RBT[*].MFaultLvl.Active		BOOL		Robot Minor Fault Occurrence	
Cause and correction	**Assumed cause**		**Correction**		**Prevention**	
	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.	
Attached information	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 (ErrorIDEx)					
Precautions/ Remarks	If a program is changed after an error occurs, the attached information that is displayed may not be correct.					

Event name	Rotation Velocity Ratio Setting Out of Range			Event code	5401551C hex	
Meaning	The value of *RotationVelocityRatio* (Rotation Velocity Ratio) member in the *MotionParams* (Motion Parameters) input variable to a robot control instruction is out of range.					
Source	PLC Function Module		Source details	Instruction	Detection timing	At instruction execution
Error attributes	Level	Observation	Recovery	---	Log category	System
Effects	User program	Continues.	Operation	The relevant instruction will end according to specifications.		
System-defined variables	Variable		Data type		Name	
	_RC_RBT[*].MFaultLvl.Active		BOOL		Robot Minor Fault Occurrence	
Cause and correction	Assumed cause		Correction		Prevention	
	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.	
Attached information	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 (ErrorIDEx)					
Precautions/ Remarks	If a program is changed after an error occurs, the attached information that is displayed may not be correct.					

<table>
<tr><td>Event name</td><td colspan="3">Velocity Setting Out of Range</td><td>Event code</td><td colspan="2">5401551D hex</td></tr>
<tr><td>Meaning</td><td colspan="6">The value of Velocity (Velocity) member in the MotionParams (Motion Parameters) input variable to a robot control instruction is out of range.</td></tr>
<tr><td>Source</td><td colspan="2">PLC Function Module</td><td>Source details</td><td>Instruction</td><td>Detection timing</td><td>At instruction execution</td></tr>
<tr><td>Error attributes</td><td>Level</td><td>Observation</td><td>Recovery</td><td>---</td><td>Log category</td><td>System</td></tr>
<tr><td>Effects</td><td>User program</td><td>Continues.</td><td>Operation</td><td colspan="3">The relevant instruction will end according to specifications.</td></tr>
<tr><td rowspan="2">System-defined variables</td><td colspan="2">Variable</td><td colspan="2">Data type</td><td colspan="2">Name</td></tr>
<tr><td colspan="2">_RC_RBT[*].MFaultLvl.Active</td><td colspan="2">BOOL</td><td colspan="2">Robot Minor Fault Occurrence</td></tr>
<tr><td rowspan="2">Cause and correction</td><td colspan="2">Assumed cause</td><td colspan="2">Correction</td><td colspan="2">Prevention</td></tr>
<tr><td colspan="2">Instruction input parameter exceeded the valid range of the input variable.</td><td colspan="2">Correct the parameter so that the valid range of the input variable is not exceeded for the relevant instruction.</td><td colspan="2">Set the input parameter to the instruction so that the valid range of the input variable is not exceeded.</td></tr>
<tr><td>Attached information</td><td colspan="6">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 (ErrorIDEx)</td></tr>
<tr><td>Precautions/ Remarks</td><td colspan="6">If a program is changed after an error occurs, the attached information that is displayed may not be correct.</td></tr>
</table>

Event name	Acceleration Ratio Setting Out of Range			Event code	5401551E hex	
Meaning	The value of *AccelerationRatio* (Acceleration Ratio) member in the *MotionParams* (Motion Parameters) input variable to a robot control instruction is out of range.					
Source	PLC Function Module		Source details	Instruction	Detection timing	At instruction execution
Error attributes	Level	Observation	Recovery	---	Log category	System
Effects	User program	Continues.	Operation	The relevant instruction will end according to specifications.		
System-defined variables	Variable		Data type		Name	
	_RC_RBT[*].MFaultLvl.Active		BOOL		Robot Minor Fault Occurrence	
Cause and correction	Assumed cause		Correction		Prevention	
	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.	
Attached information	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 (ErrorIDEx)					
Precautions/ Remarks	If a program is changed after an error occurs, the attached information that is displayed may not be correct.					

<table>
<tr><td>Event name</td><td colspan="3">Deceleration Ratio Setting Out of Range</td><td>Event code</td><td colspan="2">5401551F hex</td></tr>
<tr><td>Meaning</td><td colspan="6">The value of DecelerationRatio (Deceleration Ratio) member in the MotionParams (Motion Parameters) input variable to a robot control instruction is out of range.</td></tr>
<tr><td>Source</td><td colspan="2">PLC Function Module</td><td>Source details</td><td>Instruction</td><td>Detection timing</td><td>At instruction execution</td></tr>
<tr><td>Error attributes</td><td>Level</td><td>Observation</td><td>Recovery</td><td>---</td><td>Log category</td><td>System</td></tr>
<tr><td>Effects</td><td>User program</td><td>Continues.</td><td>Operation</td><td colspan="3">The relevant instruction will end according to specifications.</td></tr>
<tr><td rowspan="2">System-defined variables</td><td colspan="2">Variable</td><td colspan="2">Data type</td><td colspan="2">Name</td></tr>
<tr><td colspan="2">_RC_RBT[*].MFaultLvl.Active</td><td colspan="2">BOOL</td><td colspan="2">Robot Minor Fault Occurrence</td></tr>
<tr><td rowspan="2">Cause and correction</td><td colspan="2">Assumed cause</td><td colspan="2">Correction</td><td colspan="2">Prevention</td></tr>
<tr><td colspan="2">Instruction input parameter exceeded the valid range of the input variable.</td><td colspan="2">Correct the parameter so that the valid range of the input variable is not exceeded for the relevant instruction.</td><td colspan="2">Set the input parameter to the instruction so that the valid range of the input variable is not exceeded.</td></tr>
<tr><td>Attached information</td><td colspan="6">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 (ErrorIDEx)</td></tr>
<tr><td>Precautions/ Remarks</td><td colspan="6">If a program is changed after an error occurs, the attached information that is displayed may not be correct.</td></tr>
</table>

<table>
<tr><td>Event name</td><td colspan="3">Positioning Accuracy Selection Out of Range</td><td>Event code</td><td colspan="2">54015520 hex</td></tr>
<tr><td>Meaning</td><td colspan="6">The value of NullingTolerance (Positioning Accuracy) member in the MotionParams (Motion Parameters) input variable to a robot control instruction is out of range.</td></tr>
<tr><td>Source</td><td colspan="2">PLC Function Module</td><td>Source details</td><td>Instruction</td><td>Detection timing</td><td>At instruction execution</td></tr>
<tr><td>Error attributes</td><td>Level</td><td>Observation</td><td>Recovery</td><td>---</td><td>Log category</td><td>System</td></tr>
<tr><td>Effects</td><td>User program</td><td>Continues.</td><td>Operation</td><td colspan="3">The relevant instruction will end according to specifications.</td></tr>
<tr><td rowspan="2">System-defined variables</td><td colspan="2">Variable</td><td colspan="2">Data type</td><td colspan="2">Name</td></tr>
<tr><td colspan="2">_RC_RBT[*].MFaultLvl.Active</td><td colspan="2">BOOL</td><td colspan="2">Robot Minor Fault Occurrence</td></tr>
<tr><td rowspan="2">Cause and correction</td><td colspan="2">Assumed cause</td><td colspan="2">Correction</td><td colspan="2">Prevention</td></tr>
<tr><td colspan="2">Instruction input parameter exceeded the valid range of the input variable.</td><td colspan="2">Correct the parameter so that the valid range of the input variable is not exceeded for the relevant instruction.</td><td colspan="2">Set the input parameter to the instruction so that the valid range of the input variable is not exceeded.</td></tr>
<tr><td>Attached information</td><td colspan="6">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 (ErrorIDEx)</td></tr>
<tr><td>Precautions/ Remarks</td><td colspan="6">If a program is changed after an error occurs, the attached information that is displayed may not be correct.</td></tr>
</table>

<table>
<tr><td>Event name</td><td colspan="3">Rotation Limit Selection Out of Range</td><td>Event code</td><td colspan="2">54015521 hex</td></tr>
<tr><td>Meaning</td><td colspan="6">The value of SingleTurn (Rotation Limit) member in the MotionParams (Motion Parameters) input variable to a robot control instruction is out of range.</td></tr>
<tr><td>Source</td><td colspan="2">PLC Function Module</td><td>Source details</td><td>Instruction</td><td>Detection timing</td><td>At instruction execution</td></tr>
<tr><td>Error attributes</td><td>Level</td><td>Observation</td><td>Recovery</td><td>---</td><td>Log category</td><td>System</td></tr>
<tr><td>Effects</td><td>User program</td><td>Continues.</td><td>Operation</td><td colspan="3">The relevant instruction will end according to specifications.</td></tr>
<tr><td rowspan="2">System-defined variables</td><td colspan="2">Variable</td><td colspan="2">Data type</td><td colspan="2">Name</td></tr>
<tr><td colspan="2">_RC_RBT[*].MFaultLvl.Active</td><td colspan="2">BOOL</td><td colspan="2">Robot Minor Fault Occurrence</td></tr>
<tr><td rowspan="2">Cause and correction</td><td colspan="2">Assumed cause</td><td colspan="2">Correction</td><td colspan="2">Prevention</td></tr>
<tr><td colspan="2">Instruction input parameter exceeded the valid range of the input variable.</td><td colspan="2">Correct the parameter so that the valid range of the input variable is not exceeded for the relevant instruction.</td><td colspan="2">Set the input parameter to the instruction so that the valid range of the input variable is not exceeded.</td></tr>
<tr><td>Attached information</td><td colspan="6">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 (ErrorIDEx)</td></tr>
<tr><td>Precautions/ Remarks</td><td colspan="6">If a program is changed after an error occurs, the attached information that is displayed may not be correct.</td></tr>
</table>

<table>
<tr><td>Event name</td><td colspan="3">Buffer Mode Selection Out of Range</td><td>Event code</td><td colspan="2">54015522 hex</td></tr>
<tr><td>Meaning</td><td colspan="6">The value of BufferMode (Buffer Mode Selection) input variable to a robot control instruction is out of range.</td></tr>
<tr><td>Source</td><td colspan="2">PLC Function Module</td><td>Source details</td><td>Instruction</td><td>Detection timing</td><td>At instruction execution</td></tr>
<tr><td>Error attributes</td><td>Level</td><td>Observation</td><td>Recovery</td><td>---</td><td>Log category</td><td>System</td></tr>
<tr><td>Effects</td><td>User program</td><td>Continues.</td><td>Operation</td><td colspan="3">The relevant instruction will end according to specifications.</td></tr>
<tr><td rowspan="2">System-defined variables</td><td colspan="2">Variable</td><td colspan="2">Data type</td><td colspan="2">Name</td></tr>
<tr><td colspan="2">_RC_RBT[*].MFaultLvl.Active</td><td colspan="2">BOOL</td><td colspan="2">Robot Minor Fault Occurrence</td></tr>
<tr><td rowspan="2">Cause and correction</td><td colspan="2">Assumed cause</td><td colspan="2">Correction</td><td colspan="2">Prevention</td></tr>
<tr><td colspan="2">Instruction input parameter exceeded the valid range of the input variable.</td><td colspan="2">Correct the parameter so that the valid range of the input variable is not exceeded for the relevant instruction.</td><td colspan="2">Set the input parameter to the instruction so that the valid range of the input variable is not exceeded.</td></tr>
<tr><td>Attached information</td><td colspan="6">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 (ErrorIDEx)</td></tr>
<tr><td>Precautions/ Remarks</td><td colspan="6">If a program is changed after an error occurs, the attached information that is displayed may not be correct.</td></tr>
</table>

Event name	Target Position Specification Method Setting Out of Range		Event code	54015523 hex	
Meaning	The value of *PositionMode* (Target Position Specification Method) input variable to a robot control instruction is out of range.				
Source	PLC Function Module	Source details	Instruction	Detection timing	At instruction execution
Error attributes	Level: Observation	Recovery	---	Log category	System
Effects	User program: Continues.	Operation	The relevant instruction will end according to specifications.		
System-defined variables	**Variable**	**Data type**		**Name**	
	_RC_RBT[*].MFaultLvl.Active	BOOL		Robot Minor Fault Occurrence	
Cause and correction	**Assumed cause**	**Correction**		**Prevention**	
	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.	
Attached information	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 (ErrorIDEx)				
Precautions/ Remarks	If a program is changed after an error occurs, the attached information that is displayed may not be correct.				

Event name	Target Offset Setting Out of Range			**Event code**	5401552F hex	
Meaning	The value of *Offset* (Target Offset) input variable to a robot control instruction is out of range.					
Source	PLC Function Module		**Source details**	Instruction	**Detection timing**	At instruction execution
Error attributes	**Level**	Observation	**Recovery**	---	**Log category**	System
Effects	**User program**	Continues.	**Operation**	The relevant instruction will end according to specifications.		
System-defined variables	**Variable**		**Data type**		**Name**	
	_RC_RBT[*].MFaultLvl.Active		BOOL		Robot Minor Fault Occurrence	
Cause and correction	**Assumed cause**		**Correction**		**Prevention**	
	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.	
Attached information	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 (ErrorIDEx)					
Precautions/ Remarks	If a program is changed after an error occurs, the attached information that is displayed may not be correct.					

Event name	Robot Control Instruction Executed while Robot is not Attached		Event code	54015533 hex	
Meaning	An instruction required that a robot has been attached was executed for the robot that has not been attached.				
Source	PLC Function Module	Source details	Instruction	Detection timing	At instruction execution
Error attributes	Level	Observation	Recovery	---	Log category	System
Effects	User program	Continues.	Operation	The relevant instruction will end according to specifications.	
System-defined variables	Variable	Data type	Name		
	_RC_RBT[*].MFaultLvl.Active	BOOL	Robot Minor Fault Occurrence		
Cause and correction	Assumed cause	Correction	Prevention		
	An instruction that controls a robot was executed for the robot that has not been attached.	Correct the program so that the instruction is executed while the robot has been attached.	Make sure that the robot has been attached before executing the robot control instruction. To attach the robot, change the operating mode of the robot to Auto mode and COMP mode, and then attach the robot with the RC_AttachRobot (Attach Robot) instruction.		
Attached information	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 (ErrorIDEx)				
Precautions/ Remarks	If a program is changed after an error occurs, the attached information that is displayed may not be correct.				

Event name	Tool Coordination Transform Setting Out of Range			Event code	54015535 hex	
Meaning	The value of *ToolCoordTransform* (Tool Conversion Coordinates) input variable to a robot control instruction is out of range.					
Source	PLC Function Module		Source details	Instruction	Detection timing	At instruction execution
Error attributes	Level	Observation	Recovery	---	Log category	System
Effects	User program	Continues.	Operation	The relevant instruction will end according to specifications.		
System-defined variables	Variable		Data type		Name	
	_RC_RBT[*].MFaultLvl.Active		BOOL		Robot Minor Fault Occurrence	
Cause and correction	Assumed cause		Correction		Prevention	
	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.	
Attached information	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 (ErrorIDEx)					
Precautions/ Remarks	If a program is changed after an error occurs, the attached information that is displayed may not be correct.					

<table>
<tr><td>Event name</td><td colspan="3">Robot Control Instruction Multi-execution Disabled</td><td>Event code</td><td colspan="2">54015536 hex</td></tr>
<tr><td>Meaning</td><td colspan="6">Multiple robot control instructions that cannot be executed simultaneously were executed.</td></tr>
<tr><td>Source</td><td colspan="2">PLC Function Module</td><td>Source details</td><td>Instruction</td><td>Detection timing</td><td>At instruction execution</td></tr>
<tr><td>Error attributes</td><td>Level</td><td>Observation</td><td>Recovery</td><td>---</td><td>Log category</td><td>System</td></tr>
<tr><td>Effects</td><td>User program</td><td>Continues.</td><td>Operation</td><td colspan="3">The relevant instruction will end according to specifications.</td></tr>
<tr><td rowspan="3">System-defined variables</td><td colspan="2">Variable</td><td colspan="2">Data type</td><td colspan="2">Name</td></tr>
<tr><td colspan="2">_RC_COM.MFaultLvl.Active</td><td colspan="2">BOOL</td><td colspan="2">Robot Control Common Minor Fault Occurrence</td></tr>
<tr><td colspan="2">_RC_RBT[*].MFaultLvl.Active</td><td colspan="2">BOOL</td><td colspan="2">Robot Minor Fault Occurrence</td></tr>
<tr><td rowspan="2">Cause and correction</td><td colspan="2">Assumed cause</td><td colspan="2">Correction</td><td colspan="2">Prevention</td></tr>
<tr><td colspan="2">Multiple robot control instructions that cannot be executed simultaneously were executed.</td><td colspan="2">Check the specifications of multi-execution of instructions for relevant instruction and correct the program so that instructions that cannot be executed at the same time are not executed simultaneously.</td><td colspan="2">Check the specifications for multi-execution of instructions for the instruction and do not execute instructions that cannot be executed at the same time.</td></tr>
<tr><td>Attached information</td><td colspan="6">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 (ErrorIDEx)</td></tr>
<tr><td>Precautions/ Remarks</td><td colspan="6">If a program is changed after an error occurs, the attached information that is displayed may not be correct.</td></tr>
</table>

Event name	Robot Control Instruction Multi-execution Buffer Limit Exceeded			**Event code**	5401553C hex	
Meaning	The number of multi-execution for the robot control instructions exceeded the upper limit.					
Source	PLC Function Module		**Source details**	Instruction	**Detection timing**	At instruction execution
Error attributes	**Level**	Observation	**Recovery**	---	**Log category**	System
Effects	**User program**	Continues.	**Operation**	The relevant instruction will end according to specifications.		
System-defined variables	**Variable**		**Data type**		**Name**	
	_RC_RBT[*].MFaultLvl.Active		BOOL		Robot Minor Fault Occurrence	
Cause and correction	**Assumed cause**		**Correction**		**Prevention**	
	The total number of current robot control instructions and buffered robot control instructions exceeded eight.		Correct the program so that the number of multi-execution for the robot control instructions does not exceed the upper limit.		Make sure that the total number of current robot control instructions and buffered robot control instructions does not exceed eight.	
Attached information	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 (ErrorIDEx)					
Precautions/ Remarks	If a program is changed after an error occurs, the attached information that is displayed may not be correct.					

<table>
<tr><td>Event name</td><td colspan="3">Robot Control Instruction Executed with Calibration Not Completed</td><td>Event code</td><td colspan="2">5401553D hex</td></tr>
<tr><td>Meaning</td><td colspan="6">An instruction that is required for the calibration completion was executed for a robot whose calibration was not completed.</td></tr>
<tr><td>Source</td><td colspan="2">PLC Function Module</td><td>Source details</td><td>Instruction</td><td>Detection timing</td><td>At instruction execution</td></tr>
<tr><td>Error attributes</td><td>Level</td><td>Observation</td><td>Recovery</td><td>---</td><td>Log category</td><td>System</td></tr>
<tr><td>Effects</td><td>User program</td><td>Continues.</td><td>Operation</td><td colspan="3">The relevant instruction will end according to specifications.</td></tr>
<tr><td rowspan="2">System-defined variables</td><td colspan="2">Variable</td><td colspan="2">Data type</td><td colspan="2">Name</td></tr>
<tr><td colspan="2">_RC_RBT[*].MFaultLvl.Active</td><td colspan="2">BOOL</td><td colspan="2">Robot Minor Fault Occurrence</td></tr>
<tr><td rowspan="3">Cause and correction</td><td colspan="2">Assumed cause</td><td colspan="2">Correction</td><td colspan="2">Prevention</td></tr>
<tr><td colspan="2">An instruction that controls a robot was executed for the robot that the calibration has not been completed.</td><td colspan="2" rowspan="2">Correct the program so that the relevant instruction is executed after the RC_Calibrate (Robot Calibration) instruction is executed and the calibration is completed.</td><td colspan="2" rowspan="2">If the DrvStatus.Calibrated member in the robot variable is FALSE, execute the RC_Calibrate (Robot Calibration) instruction.</td></tr>
<tr><td colspan="2">An instruction to synchronize the master machine and the robot was executed for a robot whose calibration was not completed.</td></tr>
<tr><td>Attached information</td><td colspan="6">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 (ErrorIDEx)</td></tr>
<tr><td>Precautions/ Remarks</td><td colspan="6">If a program is changed after an error occurs, the attached information that is displayed may not be correct.</td></tr>
</table>

Event name	Robot Control Instruction Executed while Robot High Power is OFF			**Event code**	5401553E hex	
Meaning	An instruction required for the robot in a Power Enabled state was executed for the robot in which high power turns OFF.					
Source	PLC Function Module		**Source details**	Instruction	**Detection timing**	At instruction execution
Error attributes	**Level**	Observation	**Recovery**	---	**Log category**	System
Effects	**User program**	Continues.	**Operation**	The relevant instruction will end according to specifications.		
System-defined variables	**Variable**		**Data type**		**Name**	
	_RC_RBT[*].MFaultLvl.Active		BOOL		Robot Minor Fault Occurrence	
Cause and correction	**Assumed cause**		**Correction**		**Prevention**	
	An instruction that controls a robot was executed for the robot in which high power turns OFF.		Turn the robot high power ON.		Check that the *DrvStatus.PowerEnabled* member in the robot variable is TRUE before executing the robot control instruction.	
Attached information	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 (ErrorIDEx)					
Precautions/ Remarks	If a program is changed after an error occurs, the attached information that is displayed may not be correct.					

<table>
<tr><td>Event name</td><td colspan="3">Robot Already Attached</td><td>Event code</td><td colspan="2">5401553F hex</td></tr>
<tr><td>Meaning</td><td colspan="6">An attempt was made to attach a robot again or execute calibration for the robot that was already attached.</td></tr>
<tr><td>Source</td><td colspan="2">PLC Function Module</td><td>Source details</td><td>Instruction</td><td>Detection timing</td><td>At instruction execution</td></tr>
<tr><td>Error attributes</td><td>Level</td><td>Observation</td><td>Recovery</td><td>---</td><td>Log category</td><td>System</td></tr>
<tr><td>Effects</td><td>User program</td><td>Continues.</td><td>Operation</td><td colspan="3">The relevant instruction will end according to specifications.</td></tr>
<tr><td rowspan="2">System-defined variables</td><td colspan="2">Variable</td><td colspan="2">Data type</td><td colspan="2">Name</td></tr>
<tr><td colspan="2">_RC_RBT[*].MFaultLvl.Active</td><td colspan="2">BOOL</td><td colspan="2">Robot Minor Fault Occurrence</td></tr>
<tr><td rowspan="2">Cause and correction</td><td colspan="2">Assumed cause</td><td colspan="2">Correction</td><td colspan="2">Prevention</td></tr>
<tr><td colspan="2">The target robot was already attached in the sequence control program.</td><td colspan="2">Correct the program so that the relevant instruction is executed if the robot has not already been attached.</td><td colspan="2">Make sure that the RC_AttachRobot (Attach Robot) instruction or the RC_Calibrate (Robot Calibration) instruction is executed if the robot has not already been attached.</td></tr>
<tr><td>Attached information</td><td colspan="6">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 (ErrorIDEx)</td></tr>
<tr><td>Precautions/ Remarks</td><td colspan="6">If a program is changed after an error occurs, the attached information that is displayed may not be correct.</td></tr>
</table>

<table>
<tr><td>Event name</td><td colspan="3">Robot Control Instruction Executed while Robot is MANUAL Mode or is not COMP Mode</td><td>Event code</td><td colspan="2">54015540 hex</td></tr>
<tr><td>Meaning</td><td colspan="6">A robot control instruction for which the robot is MANUAL mode or is not COMP mode was executed.</td></tr>
<tr><td>Source</td><td colspan="2">PLC Function Module</td><td>Source details</td><td>Instruction</td><td>Detection timing</td><td>At instruction execution</td></tr>
<tr><td>Error attributes</td><td>Level</td><td>Observation</td><td>Recovery</td><td>---</td><td>Log category</td><td>System</td></tr>
<tr><td>Effects</td><td>User program</td><td>Continues.</td><td>Operation</td><td colspan="3">The relevant instruction will end according to specifications.</td></tr>
<tr><td rowspan="2">System-defined variables</td><td colspan="2">Variable</td><td colspan="2">Data type</td><td colspan="2">Name</td></tr>
<tr><td colspan="2">_RC_RBT[*].MFaultLvl.Active</td><td colspan="2">BOOL</td><td colspan="2">Robot Minor Fault Occurrence</td></tr>
<tr><td rowspan="3">Cause and correction</td><td colspan="2">Assumed cause</td><td colspan="2">Correction</td><td colspan="2">Prevention</td></tr>
<tr><td colspan="2">The robot that you control is MANUAL mode.</td><td colspan="2">Change the robot that you control to Auto mode on the front panel.</td><td colspan="2" rowspan="2">Check that the DrvStatus.Manual member in the robot variable is FALSE and the DrvStatus.Comp member in the robot variable is TRUE before executing the robot control instruction.</td></tr>
<tr><td colspan="2">The robot in Auto mode that you control is not COMP mode.</td><td colspan="2">Change the robot in Auto mode that you control to COMP mode with a teaching pendant or the Support Software.</td></tr>
<tr><td>Attached information</td><td colspan="6">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 (ErrorIDEx)</td></tr>
<tr><td>Precautions/ Remarks</td><td colspan="6">If a program is changed after an error occurs, the attached information that is displayed may not be correct.</td></tr>
</table>

<table>
<tr><td>Event name</td><td colspan="4">Cannot Execute Robot Control Instruction</td><td>Event code</td><td colspan="2">54015544 hex</td></tr>
<tr><td>Meaning</td><td colspan="7">The Robot Control Function Module is not running.</td></tr>
<tr><td>Source</td><td colspan="2">PLC Function Module</td><td>Source details</td><td>Instruction</td><td>Detection timing</td><td colspan="2">At instruction execution</td></tr>
<tr><td>Error attributes</td><td>Level</td><td>Observation</td><td>Recovery</td><td>---</td><td>Log category</td><td colspan="2">System</td></tr>
<tr><td>Effects</td><td>User program</td><td>Continues.</td><td>Operation</td><td colspan="4">The relevant instruction will end according to specifications.</td></tr>
<tr><td rowspan="3">System-defined variables</td><td colspan="2">Variable</td><td colspan="2">Data type</td><td colspan="3">Name</td></tr>
<tr><td colspan="2">_RC_COM.MFaultLvl.Active</td><td colspan="2">BOOL</td><td colspan="3">Robot Control Common Minor Fault Occurrence</td></tr>
<tr><td colspan="2">_RC_RBT[*].MFaultLvl.Active</td><td colspan="2">BOOL</td><td colspan="3">Robot Minor Fault Occurrence</td></tr>
<tr><td rowspan="2">Cause and correction</td><td colspan="2">Assumed cause</td><td colspan="2">Correction</td><td colspan="3">Prevention</td></tr>
<tr><td colspan="2">The robot control instruction was executed while the Robot Control Function Module was not running.</td><td colspan="2">Correct the program so that the relevant instruction is executed after waiting for _RC_COM.Status.RunMode or _RC_RBT[*].DrvStatus.RunMode to change to TRUE.</td><td colspan="3">Make sure that the robot control instruction is executed after waiting for _RC_COM.Status.RunMode or _RC_RBT[*].DrvStatus.RunMode to change to TRUE.</td></tr>
<tr><td>Attached information</td><td colspan="7">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 (ErrorIDEx)</td></tr>
<tr><td>Precautions/ Remarks</td><td colspan="7">If a program is changed after an error occurs, the attached information that is displayed may not be correct.</td></tr>
</table>

Event name	Illegal Program Name Specification			**Event code**	54015548 hex	
Meaning	The program name specified for the *PrgName* (Program Name) input variable to a robot control instruction is incorrect.					
Source	PLC Function Module		**Source details**	Instruction	**Detection timing**	At instruction execution
Error attributes	**Level**	Observation	**Recovery**	---	**Log category**	System
Effects	**User program**	Continues.	**Operation**	The relevant instruction will end according to specifications.		
System-defined variables	**Variable**		**Data type**		**Name**	
	_RC_COM.MFaultLvl.Active		BOOL		Robot Control Common Minor Fault Occurrence	
Cause and correction	**Assumed cause**		**Correction**		**Prevention**	
	The length of the program name specified for the *PrgName* (Program Name) input variable to the instruction was outside of the valid range.		Correct the length of text strings of the program name that is specified in the *PrgName* (Program Name) input variable to the instruction within the valid range.		Make sure that the length of text strings of the program name that is specified in the *PrgName* (Program Name) input variable to the instruction is in the valid range.	
Attached information	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 (ErrorIDEx)					
Precautions/ Remarks	If a program is changed after an error occurs, the attached information that is displayed may not be correct.					

Event name	Unsupported Function Executed			**Event code**	5401554A hex	
Meaning	The function that is not supported was executed.					
Source	PLC Function Module		**Source details**	Instruction	**Detection timing**	At instruction execution
Error attributes	**Level**	Observation	**Recovery**	---	**Log category**	System
Effects	**User program**	Continues.	**Operation**	The relevant instruction will end according to specifications.		
System-defined variables	**Variable**		**Data type**		**Name**	
	_RC_RBT[*].MFaultLvl.Active		BOOL		Robot Minor Fault Occurrence	
Cause and correction	**Assumed cause**		**Correction**		**Prevention**	
	The function that is not supported by the robot was executed.		Check the version of the robot.		Use the robot with version that supports the function.	
Attached information	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 (ErrorIDEx)					
Precautions/ Remarks	If a program is changed after an error occurs, the attached information that is displayed may not be correct.					

<table>
<tr><td>Event name</td><td colspan="4">Illegal IP Address Specification</td><td>Event code</td><td colspan="2">5401554B hex</td></tr>
<tr><td>Meaning</td><td colspan="7">The value of IPAddr input variable to a robot control instruction is not correct.</td></tr>
<tr><td>Source</td><td colspan="2">PLC Function Module</td><td>Source details</td><td>Instruction</td><td>Detection timing</td><td colspan="2">At instruction execution</td></tr>
<tr><td>Error attributes</td><td>Level</td><td>Observation</td><td>Recovery</td><td>---</td><td>Log category</td><td colspan="2">System</td></tr>
<tr><td>Effects</td><td>User program</td><td>Continues.</td><td>Operation</td><td colspan="4">The relevant instruction will end according to specifications.</td></tr>
<tr><td rowspan="2">System-defined variables</td><td colspan="2">Variable</td><td colspan="2">Data type</td><td colspan="3">Name</td></tr>
<tr><td colspan="2">_RC_COM.MFaultLvl.Active</td><td colspan="2">BOOL</td><td colspan="3">Robot Control Common Minor Fault Occurrence</td></tr>
<tr><td rowspan="2">Cause and correction</td><td colspan="2">Assumed cause</td><td colspan="2">Correction</td><td colspan="3">Prevention</td></tr>
<tr><td colspan="2">The length of the parameter list specified for the IPAddr input variable to an instruction was outside of the valid range.</td><td colspan="2">Correct the length of the string in the parameter list specified for the IPAddr input variable to the instruction so that the length of the string is in the valid range.</td><td colspan="3">Correct the length of the string in the parameter list specified for the IPAddr input variable to the instruction so that the length of the string is in the valid range.</td></tr>
<tr><td>Attached information</td><td colspan="7">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 (ErrorIDEx)</td></tr>
<tr><td>Precautions/ Remarks</td><td colspan="7">If a program is changed after an error occurs, the attached information that is displayed may not be correct.</td></tr>
</table>

A

Appendices

A-1 Differences in Functions between Robot Integrated CPU Unit and NJ-series CPU Unit

The Robot Integrated CPU Unit cannot be used while the SD Memory Card is removed.
When an SD Memory Card is used, the differences between Robot Integrated CPU Unit and NJ-series CPU Unit are given below.
The unit version of the Robot Integrated CPU Unit is 1.41 or later.

Function	Whether to change or not	Description
Mounting or unmounting SD Memory Card with power supply switch	Changed	You cannot stop the power supply with the power supply switch on the Robot Integrated CPU Unit.
SD Memory Card function in the Sysmac Studio	Changed	The operation is not ensured when the data related in the robot control function is overwritten. *1
FTP client communications instructions	Changed	The operation is not ensured when the data related in the robot control function is overwritten. *1
FTP server	Changed	The operation is not ensured when the data related in the robot control function is overwritten. *1
SD Memory Card instructions in the sequence control program	Changed	The operation is not ensured when the data related in the robot control function is overwritten. *1
SD Memory Card life expiration detection	No change	---
System-defined variables related to SD Memory Cards	No change	---
SD Memory Card self-diagnostic functions	No change	---
SD Memory Card backups	Changed	To use this function for copying a system, you must copy the data in the SD Memory Card.
Automatic transfer from SD Memory Card	No change	---
Program transfer from SD Memory Card	Changed	If the program transfer from SD Memory Card is used for recipe change, you must design and program the operations with the Recipe Manager that changes various parameters for the V+ system.
EtherCAT diagnosis/statistics log	No change	---
Database connection Operation log	Changed	You cannot confirm the log after the SD Memory Card for the database connection is removed.
Getting internal log of the SD Memory Card	No change	---

*1. Refer to *2-8-1 Data and File Locations* on page 2-22 for information on the data related to the robot control function.

A-2 Guideline for System Service Execution Time Ratio

The V+ task is executed in the system services.
If the sufficient system service execution time is not allocated in the Robot Integrated CPU Unit, the V+ task execution time may be longer.
To operate a robot with V+ tasks as designed in advance, design the tasks so that the system service execution time ratio (CPU usage) becomes as shown in the following table.

CPU Unit model	Guideline for system service execution time ratio
NJ501-R□00	35% or more
NJ501-R□20*1	35% or more

*1. The DB connection service is also executed in the system services.
If the sufficient system service execution time is not allocated, the execution time of DB connection instruction or V+ task may be longer.

Additional Information

You can check the following in the Task Execution Time Monitor of the Sysmac Studio.
- Task execution time ratio (execution time ratio except for system services) when a Controller is connected.
- System service execution time ratio when a simulator is connected.

A-3 Version Information

This section provides information related to the different unit versions of the CPU Units and different versions of the Sysmac Studio and OMRON robots.

A-3-1 Correspondence between Unit Versions of CPU Units and Versions of Sysmac Studio and OMRON Robots

This section describes the relationship between the unit versions of the CPU Units and the versions of Sysmac Studio and OMRON robots.

Unit version of CPU Unit	Corresponding version of Sysmac Studio	eV+ version of OMRON robot
Ver.1.44	Ver.1.45	5.0C1
Ver.1.43	Ver.1.44	4.0C1
Ver.1.41	Ver.1.42	3.0C1

Precautions for Correct Use

The combination of the unit version of CPU Unit and the eV+ version of OMRON robot can be used only with the combinations listed in the above table.
For example, you cannot use the combination of the CPU Unit with unit version 1.43 and the OMRON robot with eV+ version 3.0C1.
Similarly, you cannot use the combination of the OMRON robot with eV+ version 4.0C1 and the CPU Unit with unit version 1.41.

A-3-2 Functions Supported by Each Unit Version of CPU Units

This section describes the functions that are supported for each unit version of the CPU Units and robot control instructions.

Robot Control Function

This section describes the functions that are added for each unit version of the CPU Units.

● Functions That Were Added for Unit Version 1.44

The functions that were added for CPU Units with unit version 1.44 operate in the combination with the OMRON robot with eV+ version 5.0C1.

Function	Description
Tracking control Axis (Master) – Robot (Slave)	Performs the tracking control for the motion control axis in the MC Function Module as the master machine and the OMRON robot as the slave machine.
Remote encoder latching	Uses the digital signal assigned to I/O in the V+ program as the latch signal and reads the position information of the motion control axis from the V+ program when the latch signal occurs.
Cyclic path compensation	Compensates a path of the robot that is controlled in the V+ program.

Functions That Were Added for Unit Version 1.43

No function is added for CPU Units with unit version 1.43, however the controllable OMRON robots were added.

Robot Control Instructions

The instructions that are supported and their specifications depend on the unit version of the CPU Unit and the version of the Sysmac Studio.
These are given in the following table.

Instruction	Name	New/ Change d	Version		Refer- ence
			CPU Unit	Sysmac Studio	
RC_WriteVParameter	Write Vision Parameters	New	Ver.1.44	Ver.1.45	page 8-14
RC_ReadVParameter	Read Vision Parameters	New	Ver.1.44	Ver.1.45	page 8-17
RC_VRun	Run Vision Sequence	New	Ver.1.44	Ver.1.45	page 8-19
RC_SyncPathOffset	Robot Path Compensation	New	Ver.1.44	Ver.1.45	page 9-33

I

Index

I

Index

OMRON Corporation **Industrial Automation Company**
Kyoto, JAPAN

Contact: www.ia.omron.com

Regional Headquarters

OMRON EUROPE B.V.
Wegalaan 67-69, 2132 JD Hoofddorp
The Netherlands
Tel: (31)2356-81-300/Fax: (31)2356-81-388

OMRON ELECTRONICS LLC
2895 Greenspoint Parkway, Suite 200
Hoffman Estates, IL 60169 U.S.A.
Tel: (1) 847-843-7900/Fax: (1) 847-843-7787

OMRON ASIA PACIFIC PTE. LTD.
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:

Cat. No. O037-E1-03

0421