MITSUBISHI ELECTRIC
Changes for the Better

Mitsubishi Electric Servo System Motion Module Quick Start Guide

Let's Start!

Quick Start Guide

MELSEC iQ-R Series Motion Module

[PLC CPU Ladder Program]

MITSUBISHI ELECTRIC SERVO SYSTEM MELSERVO-J5

MELSEC iQ-R series

Applicable Model

- RD78G
- RD78GH
- MR-J5-G

SAFETY PRECAUTIONS

(Read these precautions before using this product.)

Before using this product, please read this manual and the relevant manuals carefully and pay full attention to safety to handle the product correctly.

The precautions given in this manual are concerned with this product only. Refer to the MELSEC iQ-R Module Configuration Manual for a description of the PLC system safety precautions.

In this manual, the safety precautions are classified into two levels " ⚠WARNING" and " ⚠CAUTION".

⚠ **WARNING**	Indicates that incorrect handling may cause hazardous conditions, resulting in death or severe injury.
⚠ **CAUTION**	Indicates that incorrect handling may cause hazardous conditions, resulting in minor or moderate injury or property damage.

Under some circumstances, failure to observe the precautions given under " ⚠CAUTION" may lead to serious consequences.

Observe the precautions of both levels because they are important for personal and system safety.

Make sure that the end users read this manual and then keep the manual in a safe place for future reference.

[Design Precautions]

⚠WARNING

- Configure safety circuits external to the programmable controller to ensure that the entire system operates safely even when a fault occurs in the external power supply or the programmable controller. Failure to do so may result in an accident due to an incorrect output or malfunction.
  (1) Emergency stop circuits, protection circuits, and protective interlock circuits for conflicting operations (such as forward/reverse rotations or upper/lower limit positioning) must be configured external to the programmable controller.
  (2) When the programmable controller detects an abnormal condition, it stops the operation and all outputs are:
    - Turned off if the overcurrent or overvoltage protection of the power supply module is activated.
    - Held or turned off according to the parameter setting if the self-diagnostic function of the CPU module detects an error such as a watchdog timer error.
  (3) All outputs may be turned on if an error occurs in a part, such as an I/O control part, where the CPU module cannot detect any error. To ensure safety operation in such a case, provide a safety mechanism or a fail-safe circuit external to the programmable controller. For a fail-safe circuit example, refer to "General Safety Requirements" in the MELSEC iQ-R Module Configuration Manual.
  (4) Outputs may remain on or off due to a failure of a component such as a relay and transistor in an output circuit. Configure an external circuit for monitoring output signals that could cause a serious accident.
- In an output circuit, when a load current exceeding the rated current or an overcurrent caused by a load short-circuit flows for a long time, it may cause smoke and fire. To prevent this, configure an external safety circuit, such as a fuse.
- Configure a circuit so that the programmable controller is turned on first and then the external power supply. If the external power supply is turned on first, an accident may occur due to an incorrect output or malfunction.
- For the operating status of each station after a communication failure, refer to manuals for the network used. For the manuals, please consult your local Mitsubishi representative. Incorrect output or malfunction due to a communication failure may result in an accident.
- When connecting an external device with a CPU module or intelligent function module to modify data of a running programmable controller, configure an interlock circuit in the program to ensure that the entire system will always operate safely. For other forms of control (such as program modification, parameter change, forced output, or operating status change) of a running programmable controller, read the relevant manuals carefully and ensure that the operation is safe before proceeding. Improper operation may damage machines or cause accidents. When a Safety CPU is used, data cannot be modified while the Safety CPU is in SAFETY MODE.
- Especially, when a remote programmable controller is controlled by an external device, immediate action cannot be taken if a problem occurs in the programmable controller due to a communication failure. To prevent this, configure an interlock circuit in the program, and determine corrective actions to be taken between the external device and CPU module in case of a communication failure.

[Design Precautions]

⚠WARNING

- Do not write any data to the "system area" and "write-protect area" of the buffer memory in the module. Also, do not use any "use prohibited" signals as an output signal from the CPU module to each module. Doing so may cause malfunction of the programmable controller system. For the "system area", "write-protect area", and the "use prohibited" signals, refer to the user's manual for the module used. For areas used for safety communications, they are protected from being written by users, and thus safety communications failure caused by data writing does not occur.
- If a communication cable is disconnected, the network may be unstable, resulting in a communication failure of multiple stations. Configure an interlock circuit in the program to ensure that the entire system will always operate safely even if communications fail. Incorrect output or malfunction due to a communication failure may result in an accident. When safety communications are used, an interlock by the safety station interlock function protects the system from an incorrect output or malfunction.
- Configure safety circuits external to the programmable controller to ensure that the entire system operates safely even when a fault occurs in the external power supply or the programmable controller. Failure to do so may result in an accident due to an incorrect output or malfunction.
  (1) Machine homing is controlled by two kinds of data: a homing direction and a homing speed. Deceleration starts when the proximity dog signal turns on. If an incorrect homing direction is set, motion control may continue without deceleration. To prevent machine damage caused by this, configure an interlock circuit external to the programmable controller.
  (2) When the module detects an error, the motion slows down and stops or the motion rapidly stops, depending on the stop group setting in parameter. Set the parameter to meet the specifications of a positioning control system. In addition, set the homing parameter and positioning data within the specified setting range.
  (3) Outputs may remain on or off, or become undefined due to a failure of a component such as an insulation element and transistor in an output circuit, where the module cannot detect any error. In a system that the incorrect output could cause a serious accident, configure an external circuit for monitoring output signals.
- If safety standards (ex., robot safety rules, etc.,) apply to the system using the module, drive unit and servomotor, make sure that the safety standards are satisfied.
- Construct a safety circuit externally of the module or drive unit if the abnormal operation of the module or drive unit differs from the safety directive operation in the system.

[Design Precautions]

⚠CAUTION

- Do not install the control lines or communication cables together with the main circuit lines or power cables. Keep a distance of 100 mm or more between them. Failure to do so may result in malfunction due to noise.
- During control of an inductive load such as a lamp, heater, or solenoid valve, a large current (approximately ten times greater than normal) may flow when the output is turned from off to on. Therefore, use a module that has a sufficient current rating.
- After the CPU module is powered on or is reset, the time taken to enter the RUN status varies depending on the system configuration, parameter settings, and/or program size. Design circuits so that the entire system will always operate safely, regardless of the time.
- Do not power off the programmable controller or reset the CPU module while the settings are being written. Doing so will make the data in the flash ROM and SD memory card undefined. The values need to be set in the buffer memory and written to the flash ROM and SD memory card again. Doing so also may cause malfunction or failure of the module.

[Security Precautions]

⚠WARNING

- To maintain the security (confidentiality, integrity, and availability) of the programmable controller and the system against unauthorized access, denial-of-service (DoS) attacks, computer viruses, and other cyberattacks from external devices via the network, take appropriate measures such as firewalls, virtual private networks (VPNs), and antivirus solutions.

[Installation Precautions]

⚠WARNING

- Shut off the external power supply (all phases) used in the system before mounting or removing the module. Failure to do so may result in electric shock or cause the module to fail or malfunction.

[Installation Precautions]

⚠CAUTION

- Use the programmable controller in an environment that meets the general specifications in the Safety Guidelines included with the base unit. Failure to do so may result in electric shock, fire, malfunction, or damage to or deterioration of the product.
- To mount a module, place the concave part(s) located at the bottom onto the guide(s) of the base unit, and push in the module, and fix it with screw(s). Incorrect interconnection may cause malfunction, failure, or drop of the module.
- To mount a module with no module fixing hook, place the concave part(s) located at the bottom onto the guide(s) of the base unit, push in the module, and fix it with screw(s). Incorrect interconnection may cause malfunction, failure, or drop of the module.
- Tighten the screws within the specified torque range. Undertightening can cause drop of the screw, short circuit, or malfunction. Overtightening can damage the screw and/or module, resulting in drop, short circuit, or malfunction.
- When using an extension cable, connect it to the extension cable connector of the base unit securely. Check the connection for looseness. Poor contact may cause malfunction.
- When using an SD memory card, fully insert it into the SD memory card slot. Check that it is inserted completely. Poor contact may cause malfunction.
- Securely insert an extended SRAM cassette or a battery-less option cassette into the cassette connector of the CPU module. After insertion, close the cassette cover and check that the cassette is inserted completely. Poor contact may cause malfunction.
- Do not directly touch any conductive parts and electronic components of the module, SD memory card, extended SRAM cassette, battery-less option cassette, or connector. Doing so can cause malfunction or failure of the module.

[Wiring Precautions]

⚠WARNING

- Shut off the external power supply (all phases) used in the system before installation and wiring. Failure to do so may result in electric shock or cause the module to fail or malfunction.
- After installation and wiring, attach a blank cover module (RG60) to each empty slot and an included extension connector protective cover to the unused extension cable connector before powering on the system for operation. Failure to do so may result in electric shock.

[Wiring Precautions]

⚠CAUTION

- Individually ground the FG and LG terminals of the programmable controller with a ground resistance of 100 ohms or less. Failure to do so may result in electric shock or malfunction.
- Use applicable solderless terminals and tighten them within the specified torque range. If any spade solderless terminal is used, it may be disconnected when the terminal screw comes loose, resulting in failure.
- Check the rated voltage and signal layout before wiring to the module, and connect the cables correctly. Connecting a power supply with a different voltage rating or incorrect wiring may cause fire or failure.
- Connectors for external devices must be crimped or pressed with the tool specified by the manufacturer, or must be correctly soldered. Incomplete connections may cause short circuit, fire, or malfunction.
- Securely connect the connector to the module. Poor contact may cause malfunction.
- Do not install the control lines or communication cables together with the main circuit lines or power cables. Keep a distance of 100 mm or more between them. Failure to do so may result in malfunction due to noise.
- Place the cables in a duct or clamp them. If not, dangling cables may swing or inadvertently be pulled, resulting in malfunction or damage to modules or cables. In addition, the weight of the cables may put stress on modules in an environment of strong vibrations and shocks. Do not clamp the extension cables with the jacket stripped. Doing so may change the characteristics of the cables, resulting in malfunction.
- Check the interface type and correctly connect the cable. Incorrect wiring (connecting the cable to an incorrect interface) may cause failure of the module and external device.
- Tighten the terminal screws or connector screws within the specified torque range. Undertightening can cause drop of the screw, short circuit, fire, or malfunction. Overtightening can damage the screw and/or module, resulting in drop, short circuit, fire, or malfunction.
- When disconnecting the cable from the module, do not pull the cable by the cable part. For the cable with connector, hold the connector part of the cable. For the cable connected to the terminal block, loosen the terminal screw. Pulling the cable connected to the module may result in malfunction or damage to the module or cable.
- Prevent foreign matter such as dust or wire chips from entering the module. Such foreign matter can cause a fire, failure, or malfunction.
- A protective film is attached to the top of the module to prevent foreign matter, such as wire chips, from entering the module during wiring. Do not remove the film during wiring. Remove it for heat dissipation before system operation.
- Programmable controllers must be installed in control panels. Connect the main power supply to the power supply module in the control panel through a relay terminal block. Wiring and replacement of a power supply module must be performed by qualified maintenance personnel with knowledge of protection against electric shock. For wiring, refer to the MELSEC iQ-R Module Configuration Manual.
- For Ethernet cables to be used in the system, select the ones that meet the specifications in this manual. If not, normal data transmission is not guaranteed.

[Startup and Maintenance Precautions]

⚠WARNING

- Do not touch any terminal while power is on. Doing so will cause electric shock or malfunction.
- Correctly connect the battery connector. Do not charge, disassemble, heat, short-circuit, solder, or throw the battery into the fire. Also, do not expose it to liquid or strong shock. Doing so will cause the battery to produce heat, explode, ignite, or leak, resulting in injury and fire.
- Shut off the external power supply (all phases) used in the system before cleaning the module or retightening the terminal screws, connector screws, or module fixing screws. Failure to do so may result in electric shock.

[Startup and Maintenance Precautions]

⚠CAUTION

- When connecting an external device with a CPU module or intelligent function module to modify data of a running programmable controller, configure an interlock circuit in the program to ensure that the entire system will always operate safely. For other forms of control (such as program modification, parameter change, forced output, or operating status change) of a running programmable controller, read the relevant manuals carefully and ensure that the operation is safe before proceeding. Improper operation may damage machines or cause accidents.
- Especially, when a remote programmable controller is controlled by an external device, immediate action cannot be taken if a problem occurs in the programmable controller due to a communication failure. To prevent this, configure an interlock circuit in the program, and determine corrective actions to be taken between the external device and CPU module in case of a communication failure.
- Do not disassemble or modify the modules. Doing so may cause failure, malfunction, injury, or a fire.
- Use any radio communication device such as a cellular phone or PHS (Personal Handy-phone System) more than 25 cm away in all directions from the programmable controller. Failure to do so may cause malfunction.
- Shut off the external power supply (all phases) used in the system before mounting or removing the module. Failure to do so may cause the module to fail or malfunction.
- Tighten the screws within the specified torque range. Undertightening can cause drop of the component or wire, short circuit, or malfunction. Overtightening can damage the screw and/or module, resulting in drop, short circuit, or malfunction.
- After the first use of the product, do not perform each of the following operations more than 50 times (IEC 61131-2/JIS B 3502 compliant).
  Exceeding the limit may cause malfunction.
  - Mounting/removing the module to/from the base unit
  - Inserting/removing the extended SRAM cassette or battery-less option cassette to/from the CPU module
  - Mounting/removing the terminal block to/from the module
- After the first use of the product, do not insert/remove the SD memory card to/from the CPU module more than 500 times. Exceeding the limit may cause malfunction.

[Startup and Maintenance Precautions]

⚠CAUTION

- Do not touch the metal terminals on the back side of the SD memory card. Doing so may cause malfunction or failure of the module.
- Do not touch the integrated circuits on the circuit board of an extended SRAM cassette or a battery-less option cassette. Doing so may cause malfunction or failure of the module.
- Do not drop or apply shock to the battery to be installed in the module. Doing so may damage the battery, causing the battery fluid to leak inside the battery. If the battery is dropped or any shock is applied to it, dispose of it without using.
- Startup and maintenance of a control panel must be performed by qualified maintenance personnel with knowledge of protection against electric shock. Lock the control panel so that only qualified maintenance personnel can operate it.
- Before handling the module, touch a conducting object such as a grounded metal to discharge the static electricity from the human body. Failure to do so may cause the module to fail or malfunction.
- Before testing the operation, set a low speed value for the speed limit parameter so that the operation can be stopped immediately upon occurrence of a hazardous condition.
- Confirm and adjust the program and each parameter before operation. Unpredictable movements may occur depending on the machine.
- When using the absolute position system function, on starting up, and when the module or absolute position motor has been replaced, always perform a homing.
- Before starting the operation, confirm the brake function.
- Do not perform a megger test (insulation resistance measurement) during inspection.
- After maintenance and inspections are completed, confirm that the position detection of the absolute position detection function is correct.
- Lock the control panel and prevent access to those who are not certified to handle or install electric equipment.

[Operating Precautions]

⚠CAUTION

- When changing data and operating status, and modifying program of the running programmable controller from an external device such as a personal computer connected to an intelligent function module, read relevant manuals carefully and ensure the safety before operation. Incorrect change or modification may cause system malfunction, damage to the machines, or accidents.
- Do not power off the programmable controller or reset the CPU module while the setting values in the buffer memory are being written to the flash ROM in the module. Doing so will make the data in the flash ROM and SD memory card undefined. The values need to be set in the buffer memory and written to the flash ROM and SD memory card again. Doing so can cause malfunction or failure of the module.
- Note that when the reference axis speed is specified for interpolation operation, the speed of the partner axis (2nd, 3rd, or 4th axis) may exceed the speed limit value.
- Do not go near the machine during test operations or during operations such as teaching. Doing so may lead to injuries.

[Disposal Precautions]

⚠CAUTION

- When disposing of this product, treat it as industrial waste.
- When disposing of batteries, separate them from other wastes according to the local regulations. For details on battery regulations in EU member states, refer to the MELSEC iQ-R Module Configuration Manual.

[Transportation Precautions]

⚠CAUTION

- When transporting lithium batteries, follow the transportation regulations. For details on the regulated models, refer to the MELSEC iQ-R Module Configuration Manual.
- The halogens (such as fluorine, chlorine, bromine, and iodine), which are contained in a fumigant used for disinfection and pest control of wood packaging materials, may cause failure of the product. Prevent the entry of fumigant residues into the product or consider other methods (such as heat treatment) instead of fumigation. The disinfection and pest control measures must be applied to unprocessed raw wood.

INTRODUCTION

Thank you for purchasing the Mitsubishi Electric MELSEC iQ-R series programmable controllers.
This manual explains system startup, parameter settings, and the programming methods necessary to use the Motion module.
Before using this product, please read this manual and the relevant manuals carefully and develop familiarity with the functions and performance of the MELSEC iQ-R series programmable controller to handle the product correctly.
When applying the program examples provided in this manual to an actual system, ensure the applicability and confirm that it will not cause system control problems.
Please make sure that the end users read this manual.

Relevant products

RD78G4, RD78G8, RD78G16, RD78G32, RD78G64, RD78GHV, RD78GHW

CONTENTS

CONTENTS

RELEVANT MANUALS

• Motion module

Manual name [manual number]	Description	Available form
Motion Module Quick Start Guide (Programmable Controller CPU Ladder Programs) [L03194ENG] (This manual)	Describes system startup, parameter settings, and programming methods for first-time users of the Motion module	e-Manual PDF
Motion Module Quick Start Guide [L03191ENG]	Describes system startup, parameter settings, and programming methods for first-time users of the Motion module	e-Manual PDF
MELSEC iQ-R Motion Module User's Manual (Startup) [IB-0300406ENG]	Specifications, procedures before operation, system configuration, and wiring of the Motion module	Print book
		e-Manual PDF
MELSEC iQ-R Motion Module User's Manual (Application) [IB-0300411ENG]	Functions, I/O signals, variables, labels, programming, and troubleshooting of the Motion module	Print book
		e-Manual PDF
MELSEC iQ-R Motion Module User's Manual (Network) [IB-0300426ENG]	Functions, parameter settings, troubleshooting, and buffer memory of CC-Link IE TSN	Print book
		e-Manual PDF
MELSEC iQ-R Programming Manual (Motion Module Instructions, Standard Functions/Function Blocks) [IB-0300431ENG]	Instructions for the Motion module and standard functions/function blocks	Print book
		e-Manual PDF
MELSEC iQ-R Programming Manual (Motion Control Function Blocks) [IB-0300533ENG]	Motion control function blocks, variables, and programming	Print book
		e-Manual PDF

• Programmable controller

Manual name [manual number]	Description	Available form
MELSEC iQ-R CPU Module User's Manual (Startup) [SH-081263ENG]	Specifications, procedures before operation, and troubleshooting of the CPU module	Print book
		e-Manual PDF
MELSEC iQ-R CPU Module User's Manual (Application) [SH-081264ENG]	Memory, functions, devices, and parameters of the CPU module	Print book
		e-Manual PDF
MELSEC iQ-R Ethernet/CC-Link IE User's Manual (Startup) [SH-081256ENG]	Specifications, procedures before operation, system configuration, wiring, and communication examples of Ethernet, CC-Link IE Controller Network, and CC-Link IE Field Network	Print book
		e-Manual PDF
MELSEC iQ-R Ethernet User's Manual (Application) [SH-081257ENG]	Functions, parameter settings, programming, troubleshooting, I/O signals, and buffer memory of Ethernet	Print book
		e-Manual PDF
MELSEC iQ-R Structured Text (ST) Programming Guide Book [SH-081483ENG]	Explains the programming method using Structured Text (ST) in GX Works3. Fundamental operations and functions are explained using sample programs.	Print book
		e-Manual PDF
MELSEC iQ-R Programming Manual (Program Design) [SH-081265ENG]	Program specifications (ladder, ST, FBD/LD, and SFC programs)	e-Manual PDF
MELSEC iQ-R Programming Manual (CPU Module Instructions/ Standard Functions/Function Blocks) [SH-081266ENG]	Instructions for the CPU module and standard functions/function blocks	e-Manual PDF

• Servo amplifier

Manual name [manual number]	Description	Available form
MR-J5-G/MR-J5W-G User's Manual (Introduction) [SH-030294ENG]	Describes startup of the servo amplifier	Print book e-Manual PDF
MR-J5 User's Manual (Hardware) [SH-030298ENG]	Describes installation, wiring, usage of options, etc.	Print book e-Manual PDF
MR-J5 User's Manual (Function) [SH-030300ENG]	Describes the functions necessary for operating the servo amplifier	Print book e-Manual PDF
MR-J5 User's Manual (Communication Function) [SH-030302ENG]	Describes the functions necessary for using the communication function	Print book e-Manual PDF
MR-J5 User's Manual (Object Dictionary) [SH-030304ENG]	Describes the object dictionary used in the servo amplifier	Print book e-Manual PDF
MR-J5-G/MR-J5W-G User's Manual (Parameters) [SH-030308ENG]	Describes the parameters used in the servo amplifier	Print book e-Manual PDF
MR-J5 User's Manual (Troubleshooting) [SH-030312ENG]	Describes the troubleshooting used to identify the cause of alarm and warning occurrences	Print book e-Manual PDF

Point

e-Manual refers to the Mitsubishi Electric FA electronic book manuals that can be browsed using a dedicated tool.

e-Manual has the following features:

- Required information can be cross-searched in multiple manuals.
- Other manuals can be accessed from the links in the manual.
- The hardware specifications of each part can be found from the product figures.
- Pages that users often browse can be bookmarked.

1 OVERVIEW

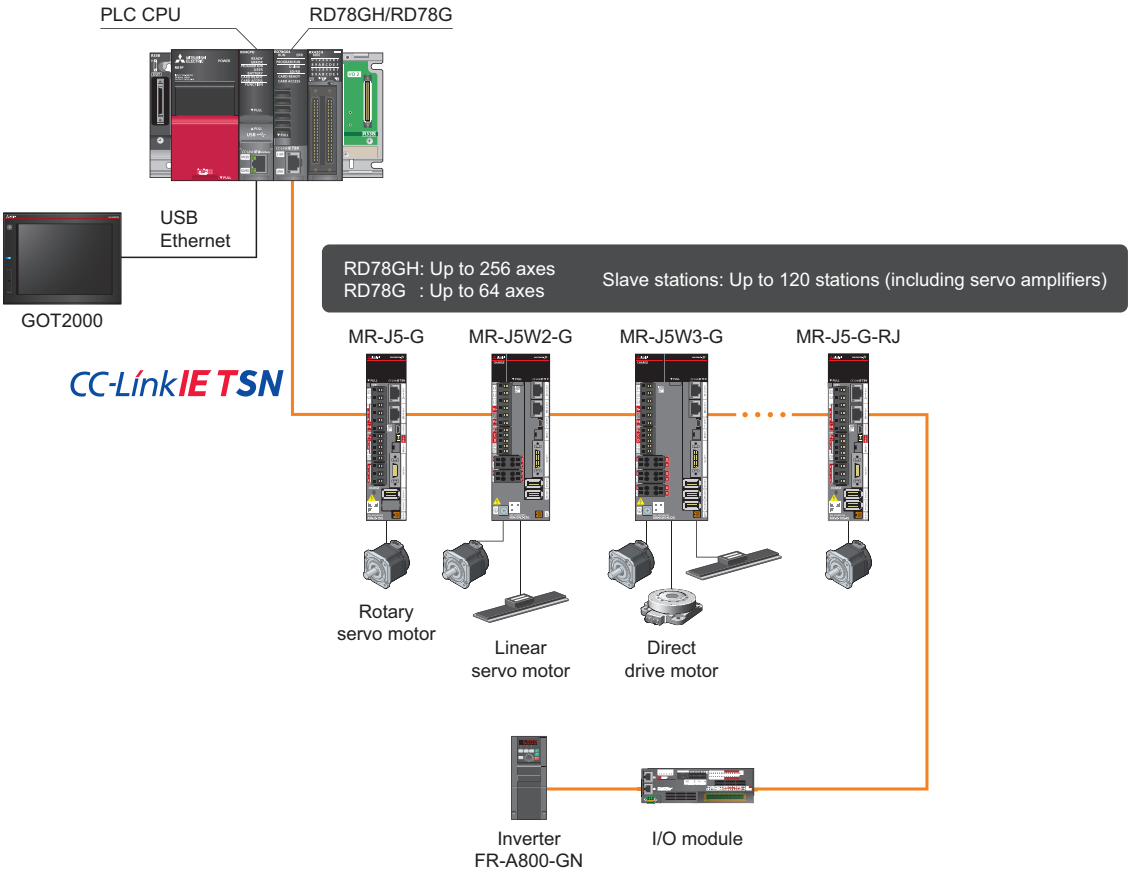
This document is intended to teach first-time users of the Mitsubishi Electric Motion module RD78G(H) about the hardware and software necessary to configure a servo system.

1.1 Features of MELSEC iQ-R Series Motion Module RD78G(H)

MELSEC iQ-R series Motion module RD78G(H) is a servo system controller that supports CC-Link IE TSN to enable mixing of the real-time control required for Motion control and the information communication utilized by IT systems. By supporting the CC-Link IE TSN network, the Motion module can freely connect with various devices including servo amplifiers, I/O modules, and high-speed counters.
This allows for a flexible system configuration that utilizes various devices.

The main features of the RD78G(H) are shown below.

- Contains a multi-core processor that enables high-speed processing
- Supports servo motor control for up to 256 axes
- Supports programming for positioning, etc., by utilizing PLCopen® Motion Control Function Blocks (FB).

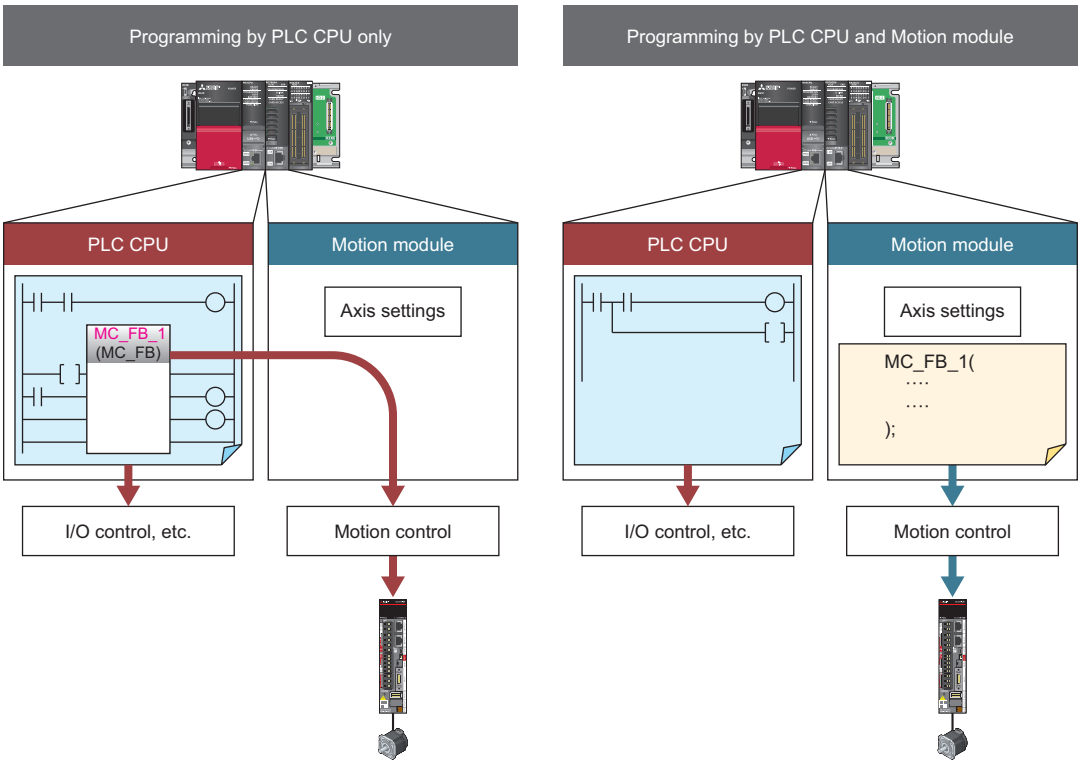


1.2 Programming with the Motion module

There are two programming methods for a servo system using RD78G(H): Programming by PLC CPU only and programming by PLC CPU and Motion module.

For the PLC CPU, program using the ladder diagram (Ladder), function block diagram (FBD), sequential function chart (SFC), and structured text (ST) languages as set forth in IEC 61131-3. For the Motion module, program using the structured text (ST) language.

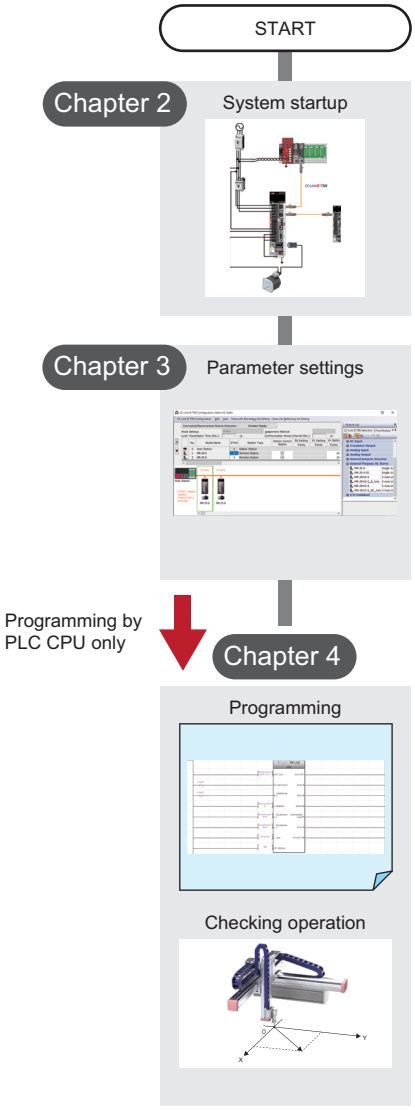
The method described in this document uses the ladder language to program on the PLC CPU.



Programming method	Advantages	Disadvantages
Programming by PLC CPU only	Easy program management by programming with only one CPU.	Longer scan time due to processing with one CPU.
Programming by PLC CPU and Motion module	Reduced scan time because Motion control is programmed to the Motion module, not affecting processing times in other sequence programs.	Necessary to manage both PLC CPU programs and Motion module programs.

1.3 Organization of This Document

This document describes procedures from startup to debugging by using a two-axis XY table as an example.
The programming part describes how to program on the PLC CPU using the ladder language.



■ System startup
· System components preparation
· Firmware/software preparation
· Profile registration
· Motion module FB library registration
· Module installation
· Wiring and cable connection
· Rotary switch setting

■ Parameter setting
· Creating a project
· Network configuration
· Parameter settings
· Motion Control Setting
· Axis group setting
· Labels
· Public labels

■ Programming
· JOG, homing
· Positioning
· Linear interpolation control
· Synchronous control

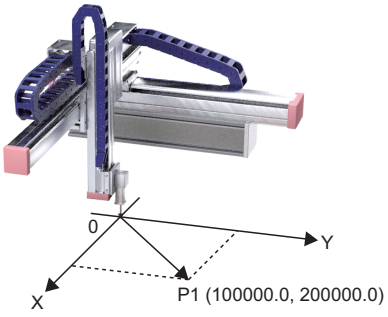
■ Checking operation
· Axis monitor
· Program monitor
· Program watch
· Event history

2 SYSTEM STARTUP

2.1 System Overview

The example in this document uses a 2-axis system that uses ball screws.

Equipment



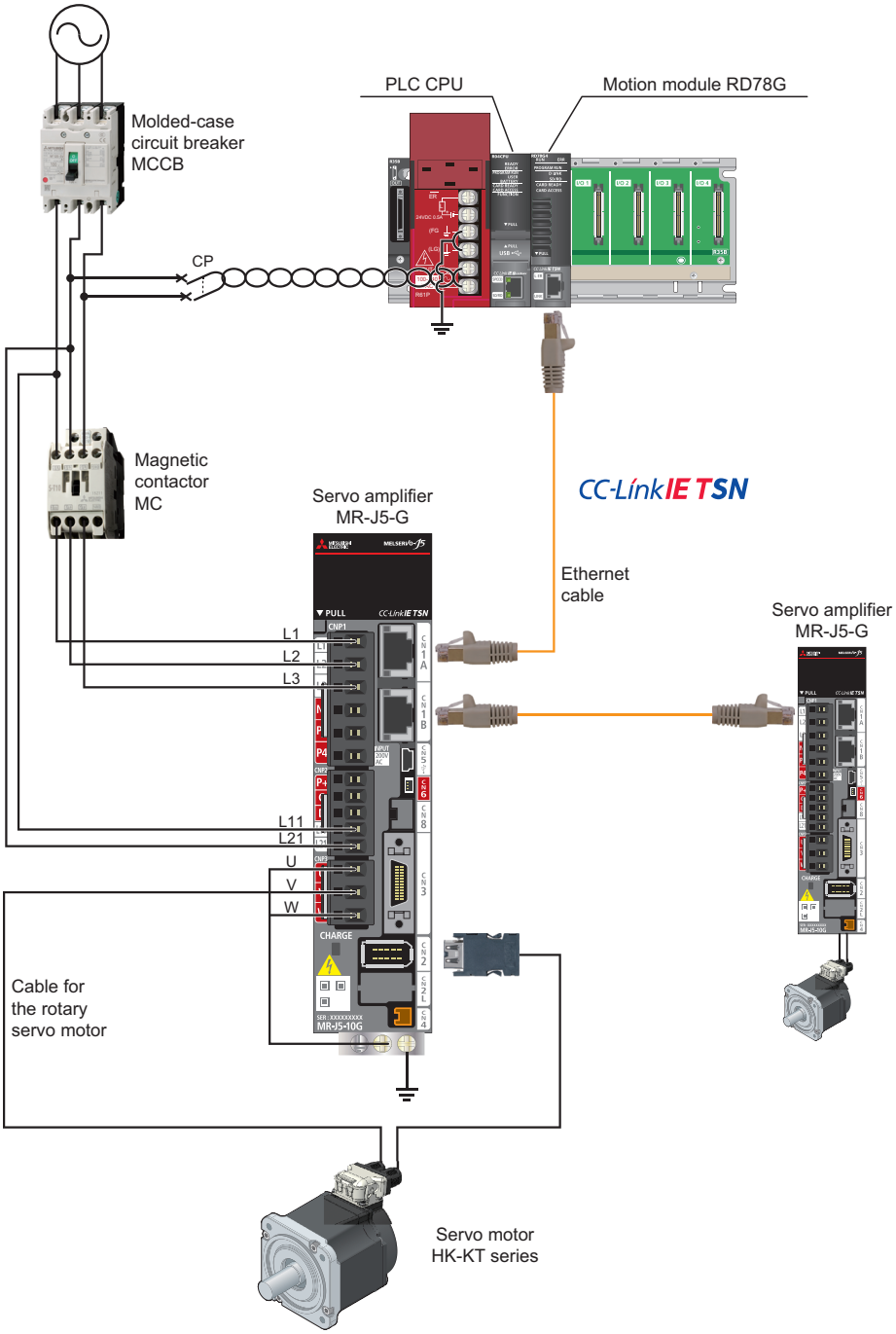
[Unit: µm]

■Specifications

Item	Description
Lead of the ball screw (PB)	10.0 [mm]
Reduction ratio (NL/NM)	1/2 (Load side [NL]/Motor side [NM]) Each time the servo motor rotates twice, the ball screw of the load side rotates once.
Encoder resolution	26 bits (67108864 [pulse])

2.2 System Configuration

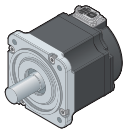
The following shows a system configuration example that uses a Motion module (RD78G), a servo amplifier (MR-J5-G), and a servo motor (HK-KT series).

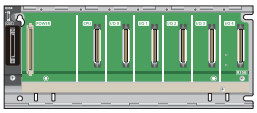


2.3 System Components Preparation

This document describes a project that uses the following components.
Prepare modules, cables, and software according to the user's system.

Motion module	Engineering environment
RD78G4 	MELSOFT GX Works3 

Servo amplifier	Servo motor
MR-J5-10G 	HK-KT13W 

Main base unit	Power supply module	CPU module
R35B 	R61P 	R04CPU 

Cable for the rotary servo motor	Ethernet cable	USB cable
MR-AEP2CBL2M-A1-L 	Category 5e or higher (double shielded/STP) straight cable 	MR-J3USBCBL3M 

Molded-case circuit breaker (MCCB)	Magnetic contactor (MC)	Circuit protector (CP)
		

2.4 Firmware/Software Installation

Firmware/software preparation

The contents of this document are based on the firmware, software, and engineering environment specifications listed in the following tables.

The screen displays may differ depending on the version of the firmware/software being used, so confirm that the latest version of the firmware/software is being used.

Please contact your local sales office for the latest version of the firmware/software.

Firmware/software

Module	Description	Module model	Version
PLC CPU module	Firmware update information file	R00CPU/R01CPU/R02CPU	14 or later
		R04CPU/R08CPU/R16CPU/R32CPU/R120CPU*1	46 or later
		R□ENCPU*1	46 or later
Motion module	Network boot software	RD78G/RD78GH	04 or later
	Boot software		04 or later
	Basic system software		09 or later
Servo amplifier	Firmware for CC-Link IE TSN-compatible servo amplifier MR-J5-G		A7 or later

*1 This version can be installed to modules that are ver. 23 or later.

Engineering environment

Item	Description	Version
MELSOFT GX Works3	PLC engineering software	1.065T or later
Motion Control Setting*1	Software to set and monitor the Motion module	1.011M or later
MELSOFT MR Configurator2	Servo amplifier setup software	1.105K or later
MELSOFT GX LogViewer	An easy-to-use tool for displaying and analyzing large-capacity data collected by modules with the logging function	1.106K or later

*1 Required to use the Motion module. For installing Motion Control Setting, refer to the following.
☞ Page 23 Motion Control Setting installation

Profile

Item	Description	Version
Profile for MR-J5-G	Profiles for MELSERVO-J5 series CC-Link IE TSN-compatible servo amplifiers	04 or later

FB library

Item	Description	Version
MELSEC iQ-R Motion module FB library*1	An FB library for systems using a MELSEC iQ-R series Motion module	03D or later

*1 Use when programming by PLC CPU only.

MELSOFT GX Works3 installation

Pre-installation check

- Log on to the personal computer as an administrator.
- Close all running applications before installation. If the product is installed while other applications are running, it may not operate normally.

Installation procedure

■Installing by using the MELSOFT GX Works3 DVD-ROM

1. Insert the MELSOFT GX Works3 DVD-ROM into the DVD-ROM drive, then double-click "setup.exe" in the Disk1 folder of the DVD-ROM.
2. Select or enter the necessary information while following the on-screen instructions.*1

*1 The product ID is written on the "License certificate" included with the product. Enter the 12-digit number divided into 3 and 9 digits.

■Installing by downloading the update module (update version)

1. Please contact your local sales office for downloading the newest version of MELSOFT GXWorks3.
2. After unzipping the downloaded zip file to a location of the user's choosing, double click "sw1dnd-gxw3-e_****.exe*1" in the "sw1dnd-gxw3-e_****"*1 folder.

*1 Displays the MELSOFT GXWorks3 version.

3. Select or enter the necessary information while following the on-screen instructions.

Point

Motion Control Setting and GX LogViewer are not included in the MELSOFT GX Works3 update module. When updating MELSOFT GX Works3 by using the update module, also update Motion Control Setting and GX LogViewer to the latest version separately. Please contact your local sales office for the update module.

Motion Control Setting installation

Motion Control Setting is required for the Motion module.
When using the Motion module, install Motion Control Setting.

Pre-installation check

- Log on to the personal computer as an administrator.
- Close all running applications before installation. If the product is installed while other applications are running, it may not operate normally.

Installation procedure

1. Please contact your local sales office for downloading Motion Control Setting.
2. After unzipping the downloaded zip file to a location of the user's choosing, double click "sw1dnn-mucnf-e_****.exe*1" in the "sw1dnn-mucnf-e_****"*1 folder.

*1 Displays the Motion Control Setting version.

3. Select or enter the necessary information while following the on-screen instructions.

Profile registration

Register the profiles for the devices to be used, such as servo amplifiers or remote I/O modules. (The profiles for MR-J5-G/MR-J5W□-G servo amplifiers are automatically registered when MELSOFT GX Works3 is installed.)

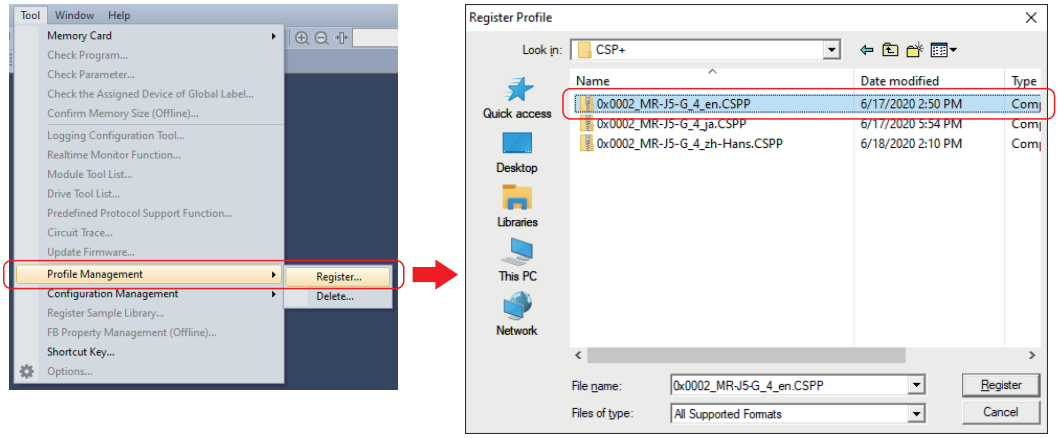
Downloading the profiles

Please contact your local sales office for downloading the profiles.

Profile registration procedure

After unzipping the downloaded zip file to a location of the user's choosing, register the CSP+ file (zip file).

1. Start MELSOFT GX Works3.
2. Select [Tool]⇨[Profile Management]⇨[Register] from the menu to display the "Register Profile" screen. Select the file for the language to use from the unzipped profile, then click the [Register] button.

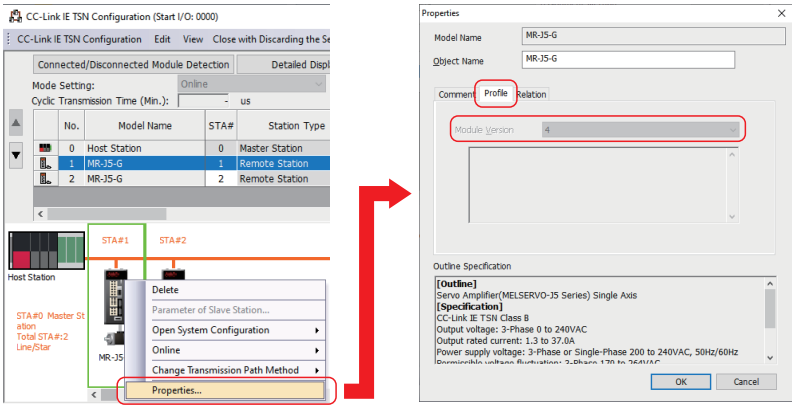


■Confirming the profile version

The profile version of the module being used can be confirmed by checking the "Module Version" displayed in the "Profile" tab of the "Properties" screen.

To display the "Properties" screen, right click the module being used (Example: Servo amplifier) in the "CC-Link IE TSN Configuration" window, then select "Properties" from the shortcut menu. For the "CC-Link IE TSN Configuration" window, refer to the following.

☞ Page 37 Network Configuration



MELSEC iQ-R series Motion module FB library

To create programs with the PLC CPU, register the FB library using the following procedure.

Downloading the Motion module FB library

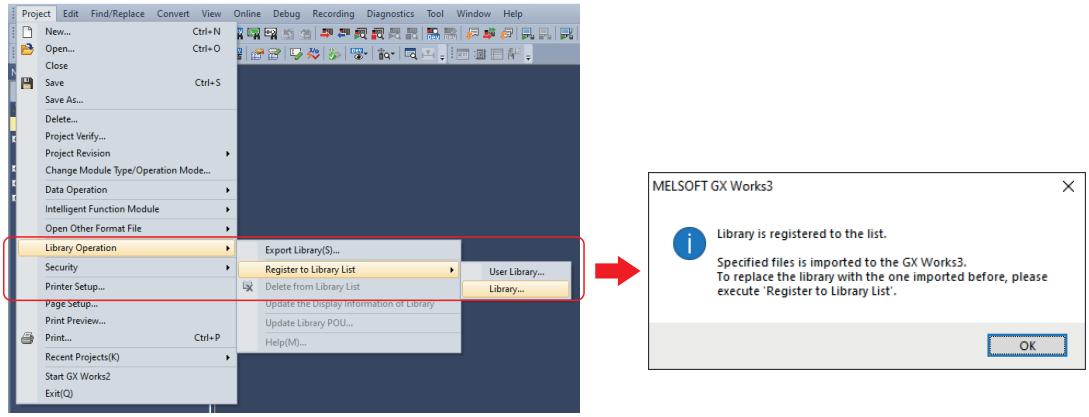
Please contact your local sales office for downloading the Motion module FB library.

Registration procedure for the Motion module FB library

After unzipping the downloaded zip file to a location of the user's choosing, register the Motion module FB library file.

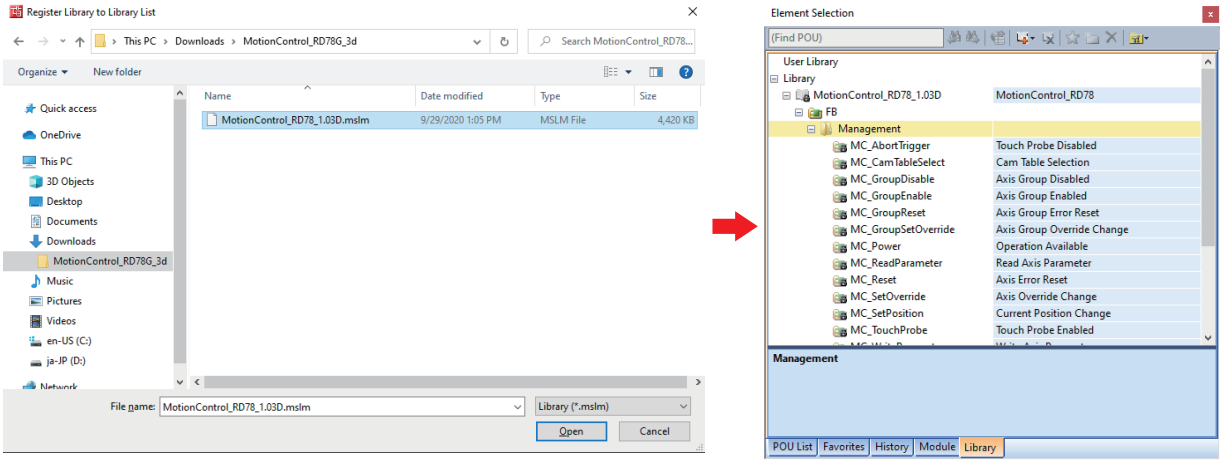
1. Start MELSOFT GX Works3.

2. Open a project of the user's choosing, then select [Project]⇨[Library Operation]⇨[Register to Library List]⇨[Library] from the menu to display the confirmation message. Click the [OK] button.



3. On the "Register Library to Library List" screen, select the Motion module FB library file (MotionControl_RD78_****.mslm[*1]) to register, then click the [Open] button.
Confirm that the Motion module FBs are registered in the [Library] tab of the element selection window.

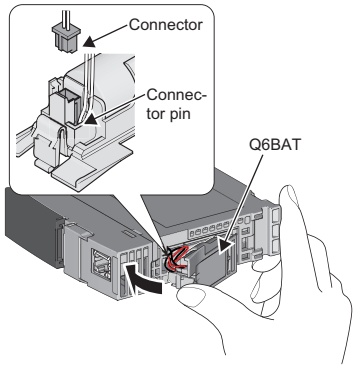
*1 **** = Motion module FB library version



2.5 Module Installation

Installing the battery

The Q6BAT installed in the CPU module is shipped with the connector disconnected. Connect the connector using the following procedure.



1. Open the battery cover located on the bottom of the CPU module.
2. Check that the Q6BAT is correctly installed.
3. Check the direction of the connector attached to the Q6BAT, then insert it into the connector pins on the cover. Firmly push the connector in all the way.
4. Close the bottom cover.

Installing the module

Install each module onto the main base unit.

2.6 Wiring and Cable Connection

The following shows a wiring and cable connection example for the Motion module and the servo amplifier.
The listed wire sizes are applicable when the MR-J5-10G servo amplifier is used.
If servo amplifiers with other capacities are used, refer to the relevant user's manual.

Wiring the power supply module

The following shows the applicable size of the power and grounding wires for the power supply module.
To reduce noise such as lightning surges, connect an isolation transformer.

Item	Applicable wire size
Power wires	AWG18 to AWG14
Grounding wires	AWG18 to AWG14

Wiring the power supply of the servo amplifier

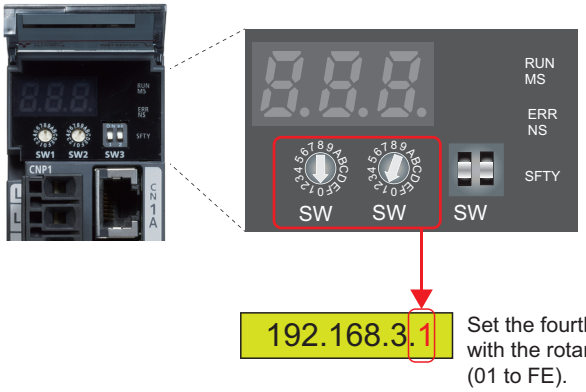
Wire the control circuit power supply (L11 and L21) and the main circuit power supply (L1, L2, and L3) of the servo amplifier.

Item	Applicable wire size
Control circuit power supply (L11 and L21)	1.25mm^2 to 2mm^2 (AWG16 to AWG14)
Main circuit power supply (L1, L2, and L3)	2mm^2 (AWG14)
Grounding wires	AWG18 to AWG14

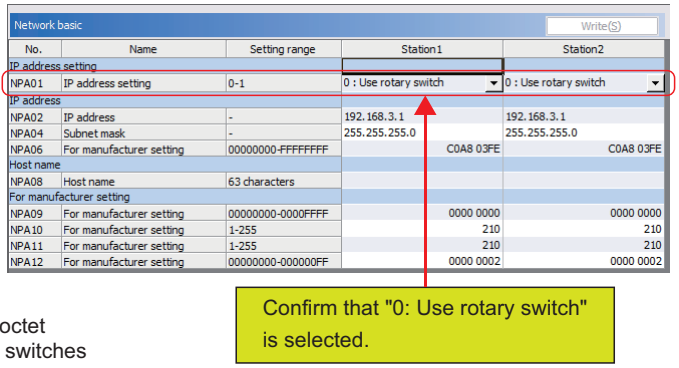
Rotary switch setting of the servo amplifier

The settings of the rotary switches (SW1 and SW2) of the servo amplifier correspond to the fourth octet of the IP address in the "CC-Link IE TSN Configuration" window.

<Rotary switches of the servo amplifier>



<"Network basic" screen for the servo amplifier>



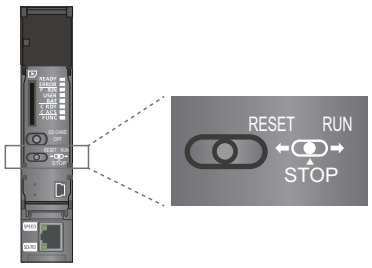
- For IP address settings for the servo amplifier, refer to the following.

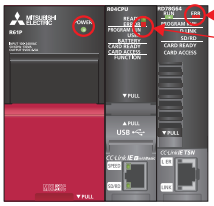
☞ Page 37 Network Configuration
☞ Page 40 Servo Parameter Setting

Connecting the cables

Connect the Ethernet cables and the cables for the rotary servo motor.
Connect the USB cable between the personal computer and the CPU module.

Powering on the system





1. Check the wiring of the power supply module.
2. Make sure that the PLC CPU module is in the STOP status.
3. Turn ON the power supply.

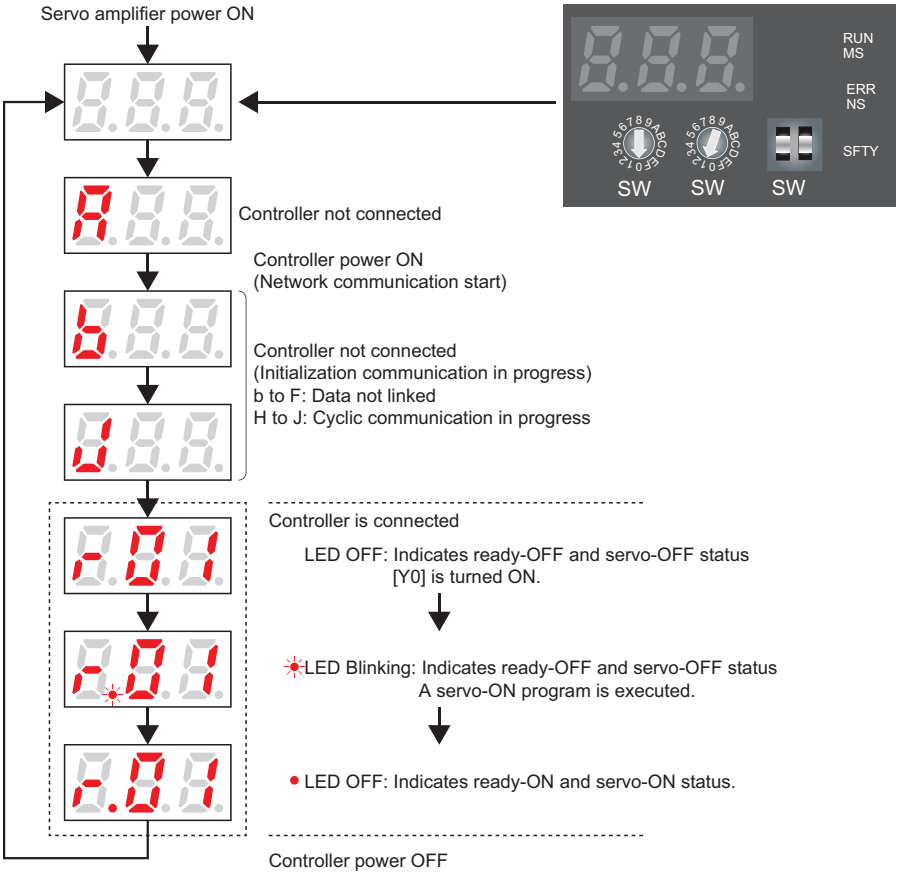
Confirm that the LEDs of the modules are ON.

Module	LED display	Description
Power supply module	LED (green) ON	—
PLC CPU module	READY LED (green) ON	The ERROR LED flashes red if parameters and programs have not been written to the CPU module, but this does not indicate that an error is occurring. After the parameters and programs are written and the power supply is cycled, the ERROR LED turns OFF.
Motion module	RUN LED (green) ON	The ERROR LED flashes red if parameters and programs have not been written to the Motion module, but this does not indicate that an error is occurring. When communication with the slave station is performed normally after the parameters and programs are written and the power supply is cycled, the ERR LED turns off.

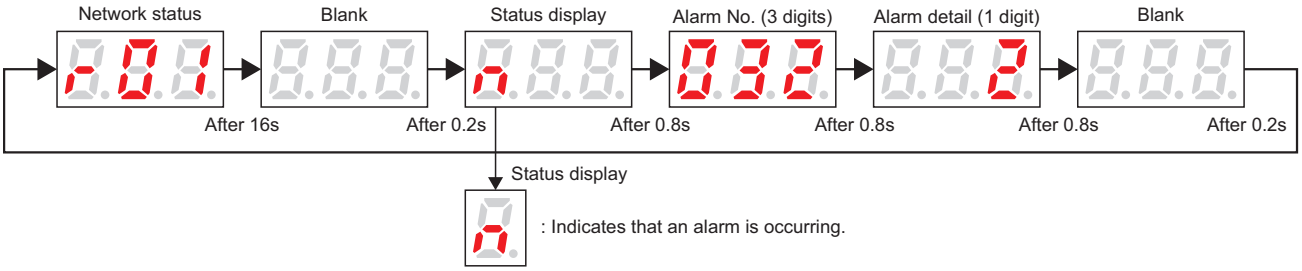
Powering on the servo amplifier

Check the wiring of the servo amplifier, then turn ON the control circuit power supply of the servo amplifier.
The communication status of the servo amplifier and the Motion module can be checked on the servo amplifier display.

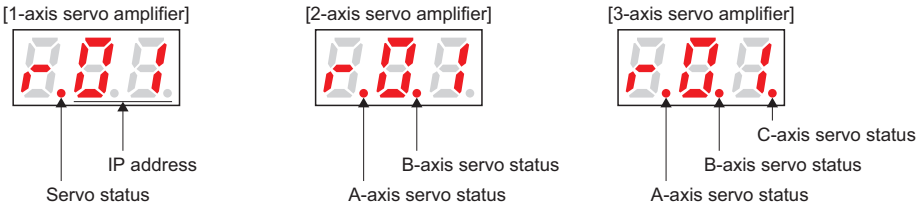
■The 7-segment LED display of the servo amplifier (normal status)



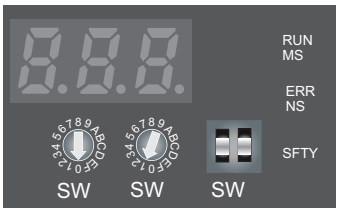
■Alarm occurrence of a 1-axis servo amplifier



■7-segment LED display during a network connection



Status LEDs

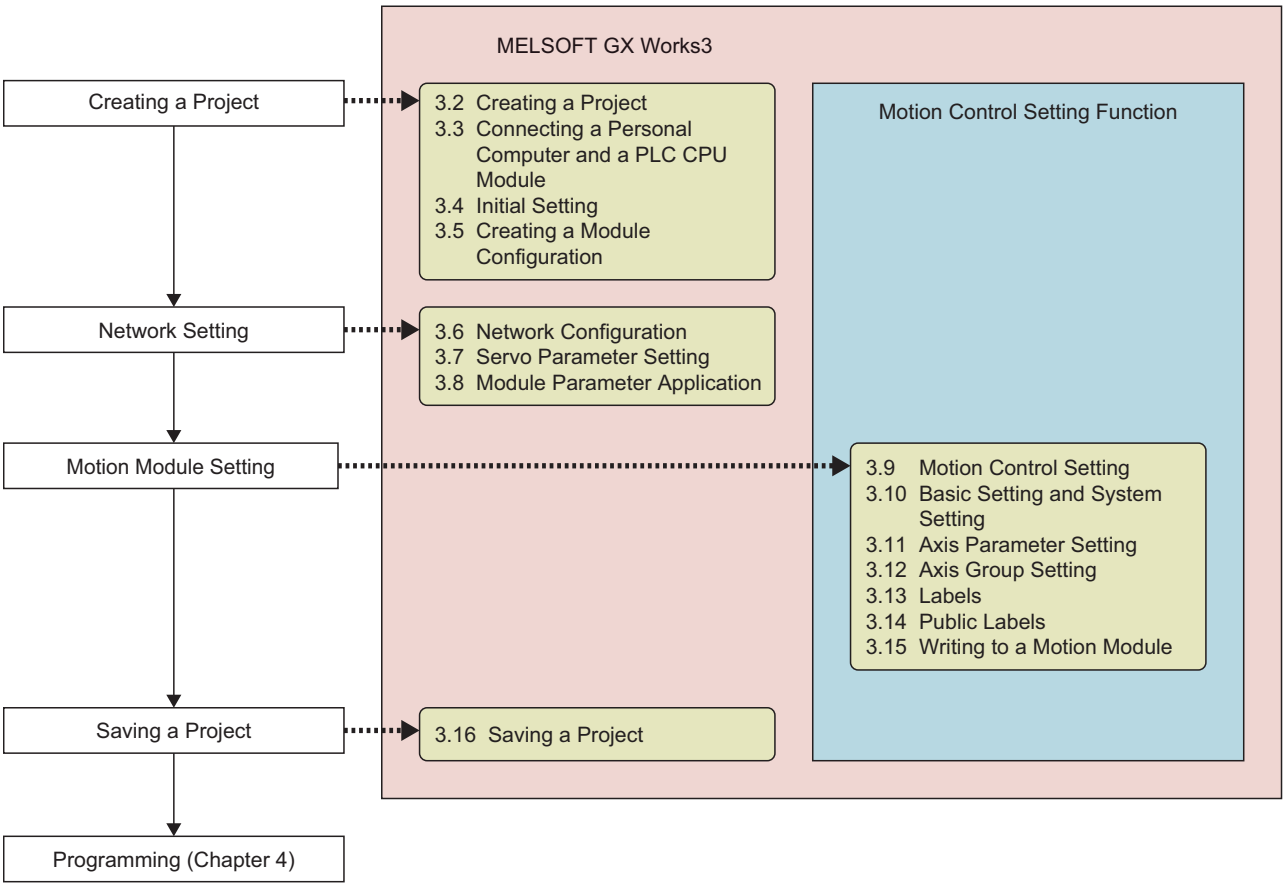


Name	LED display color	Description
RUN MS	Green	OFF: Indicates that an alarm is occurring. ON: Indicates that the servo amplifier is ON.
ERR NS	Red	OFF: Indicates that no alarm or warning is occurring. Blinking: Indicates that a warning is occurring. ON: Indicates that an alarm is occurring.
SFTY	Green	OFF: Indicates that the functional safety cannot be activated. ON: Indicates that the functional safety can be activated.

3 PARAMETER SETTING

3.1 Parameter Setting Procedure

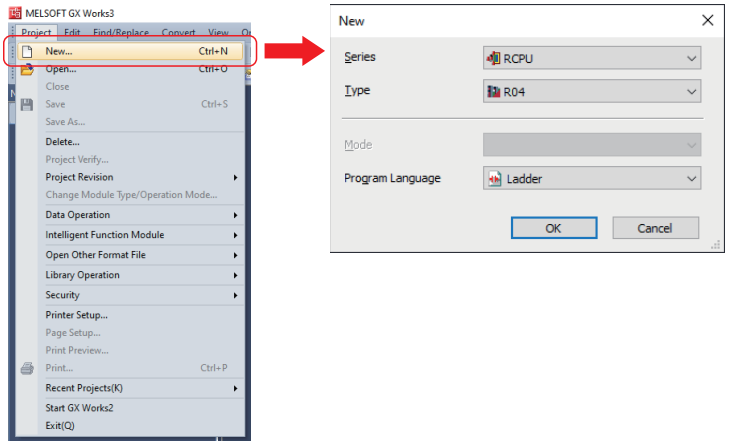
The following diagram shows the flow of the parameter setting procedure.



3.2 Creating a Project

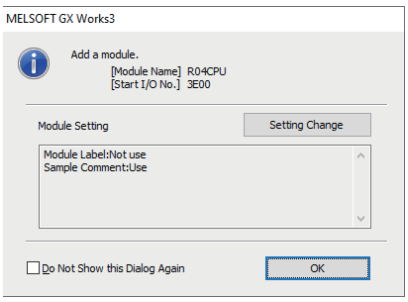
Create a new project.

1. Start MELSOFT GX Works3, then select [Project]⇨[New] from the menu.
2. The "New" screen is displayed. Select the series, the type, and the program language, then click the [OK] button.

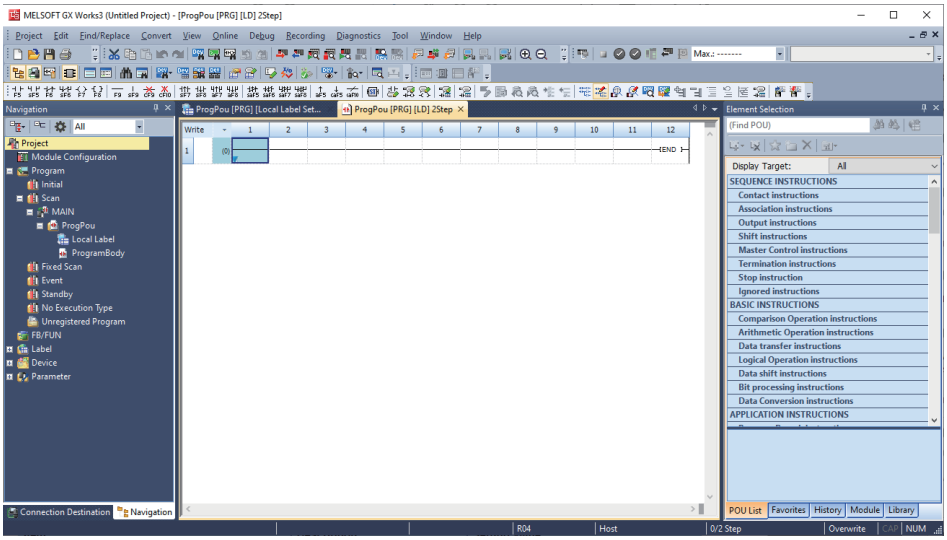


■Setting example
- Series: RCPU
- Type: Select the PLC CPU to be used
- Program language: Ladder

3. Confirm the contents of the displayed message, then click the [OK] button.



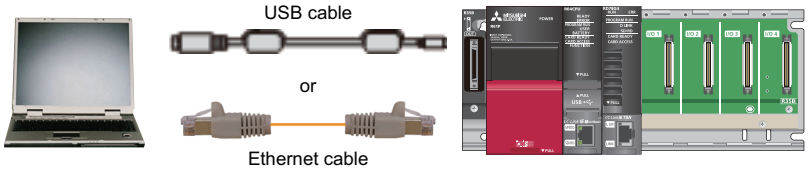
4. The project is created.



3.3 Connecting a Personal Computer and a PLC CPU Module

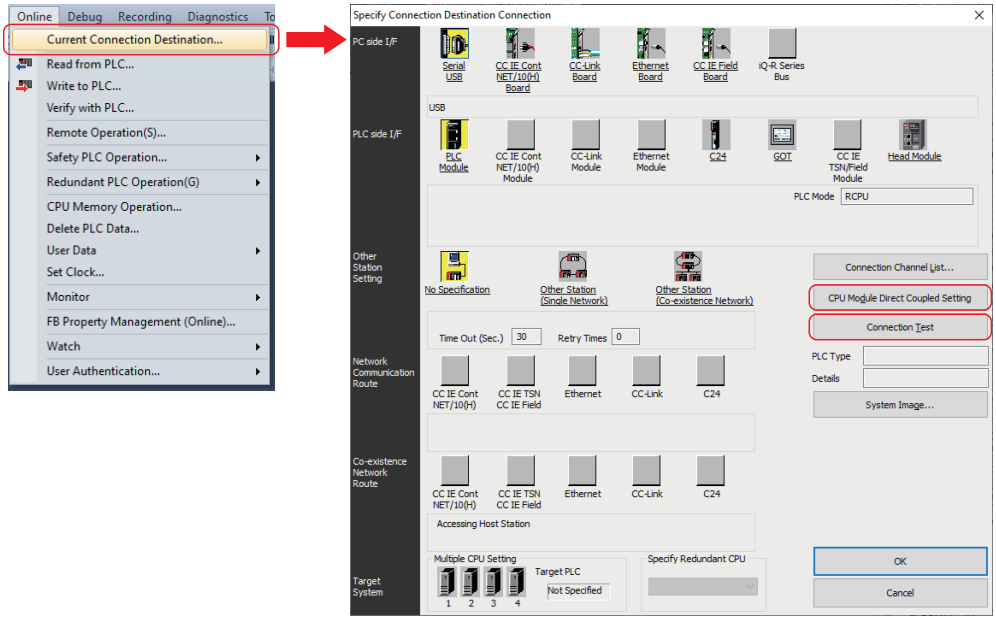
Check the connection between the personal computer and the PLC CPU module.

1. Connect the personal computer and the PLC CPU module with a USB or Ethernet cable.



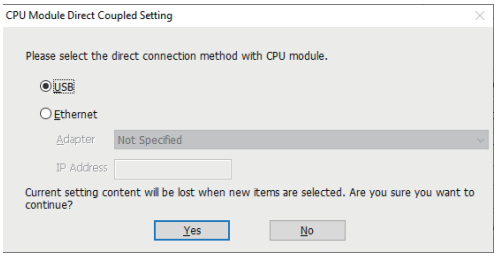
2. Select [Online]⇨[Current Connection Destination] from the menu to display the "Specify Connection Destination Connection" screen.

Click the [CPU Module Direct Coupled Setting] button.



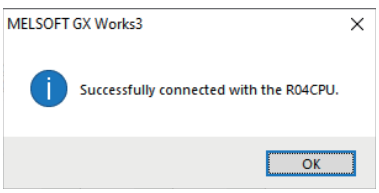
3. The "CPU Module Direct Coupled Setting" screen is displayed. Select the connection method to use, then click the [Yes] button.

*1 When selecting "Ethernet", specify the adapter.



4. Return to the "Specify Connection Destination Connection" screen. Click the [Connection Test] button on the "Specify Connection Destination Connection" screen to perform a connection test with the personal computer.

If the connection was successful, the successfully connected message is displayed. Click the [OK] button.



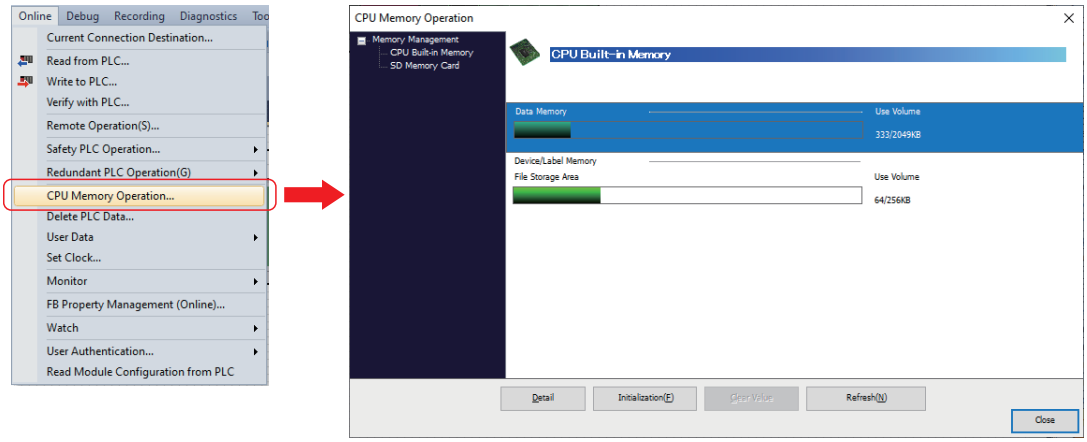
5. Return to the "Specify Connection Destination Connection" screen. Click the [OK] button to complete the specification of the connection destination.

3.4 Initial Setting

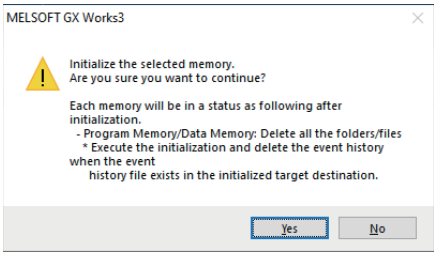
Initializing a PLC CPU module

The battery in the PLC CPU module is shipped with the connector disconnected. Therefore, the data in the memory is unstable. Initialize the PLC CPU module to clear the memory.

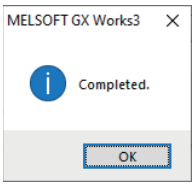
1. Select [Online]⇨[CPU Memory Operation] from the menu. The "CPU Memory Operation" screen is displayed. Click the [Initialization] button.



2. When the confirmation message appears, click the [Yes] button to start initialization.



3. When initialization is complete, the completed message is displayed. Click the [OK] button.

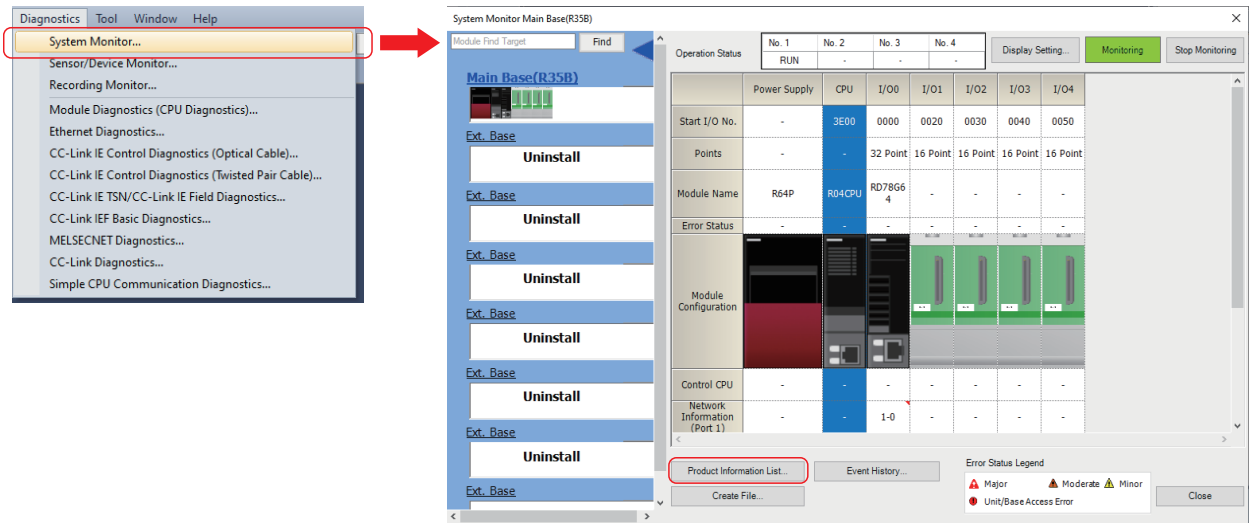


4. Return to the "CPU Memory Operation" screen. Click the [Close] button to complete the initialization.

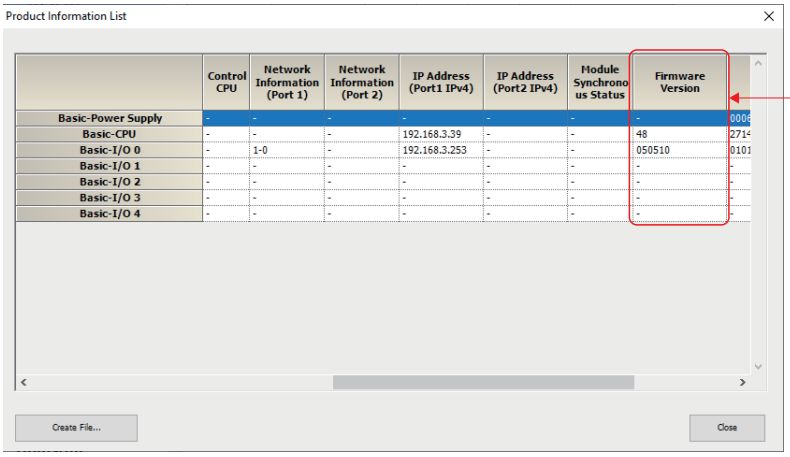
Checking the firmware version

Check the firmware version of the PLC CPU module and the Motion module.

1. Select [Diagnostics]⇨[System Monitor] from the menu to display the "System Monitor Main Base(R35B)" screen. Click the [Product Information List] button.



2. The "Product Information List" screen is displayed. Check the firmware version.



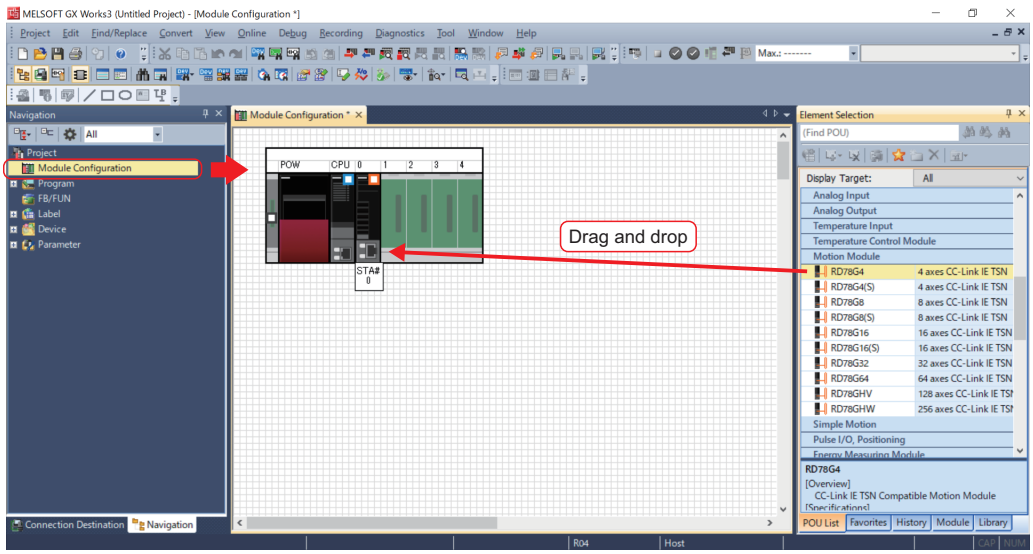
*1 Motion module version designation

050510
- Basic system software
- Boot software
- Network boot software

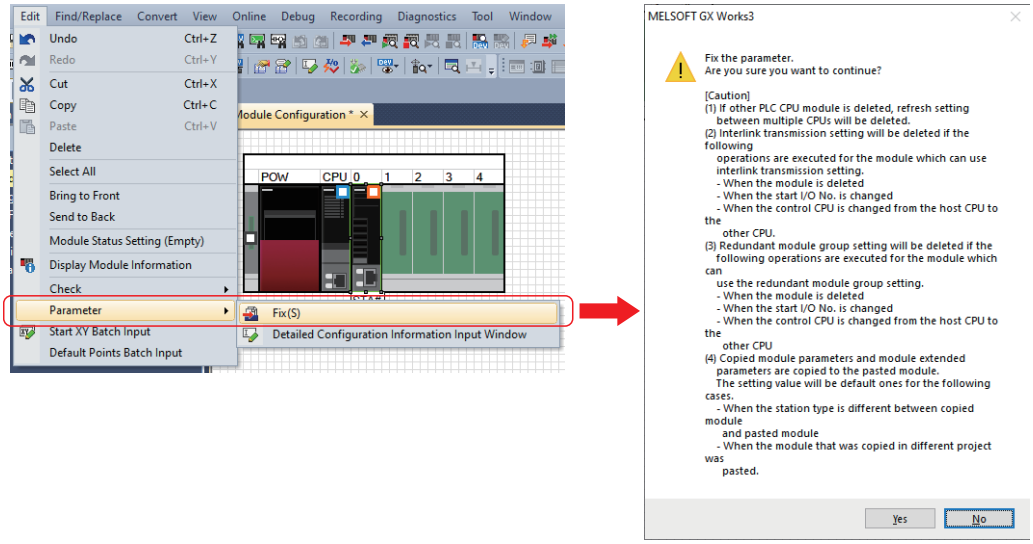
3.5 Creating a Module Configuration

Create a module configuration.

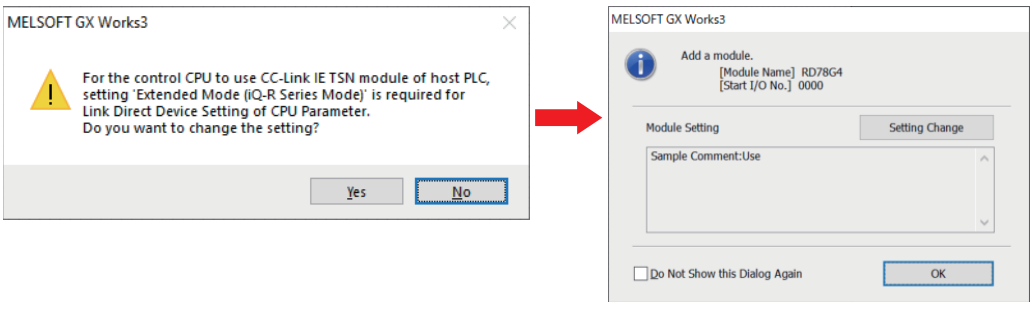
1. Double-click "Module Configuration" in the navigation window to display the "Module Configuration" window.
Click the [POU List] tab in the element selection window. Select the modules to be used (Main base unit, Power supply module, PLC CPU module, and Motion module), then drag and drop them into position on the module configuration.



2. Select [Edit]⇨[Parameter]⇨[Fix] from the menu. When the confirmation message appears, click the [Yes] button.



3. When the confirmation message appears, click the [Yes] button.
A message for the module to be set appears. Confirm the contents of the displayed message, then click the [OK] button to fix the parameters (initial values) for each module.

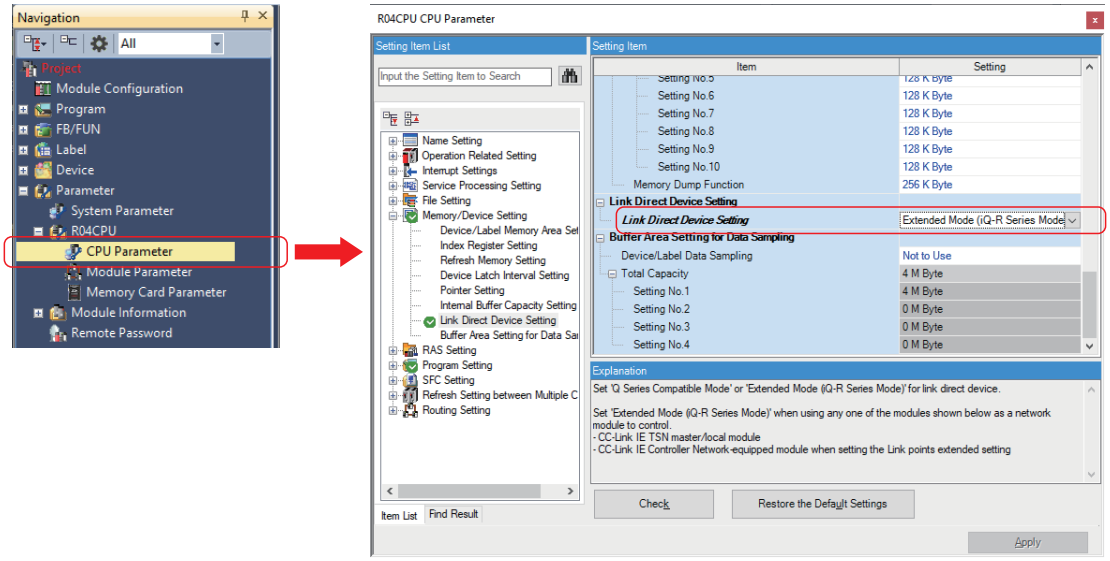


Point

When using a Motion module, "Memory/Device Setting"⇨"Link Direct Device Setting" must be set to "Extended Mode (iQ-R Series Mode)" in the parameter editor (CPU parameter).

Double-click [Parameter]⇨[CPU module model*1]⇨[CPU Parameter] in the navigation window to display the parameter editor (CPU parameter).

*1 : The model name of the set CPU module is displayed.



3.6 Network Configuration

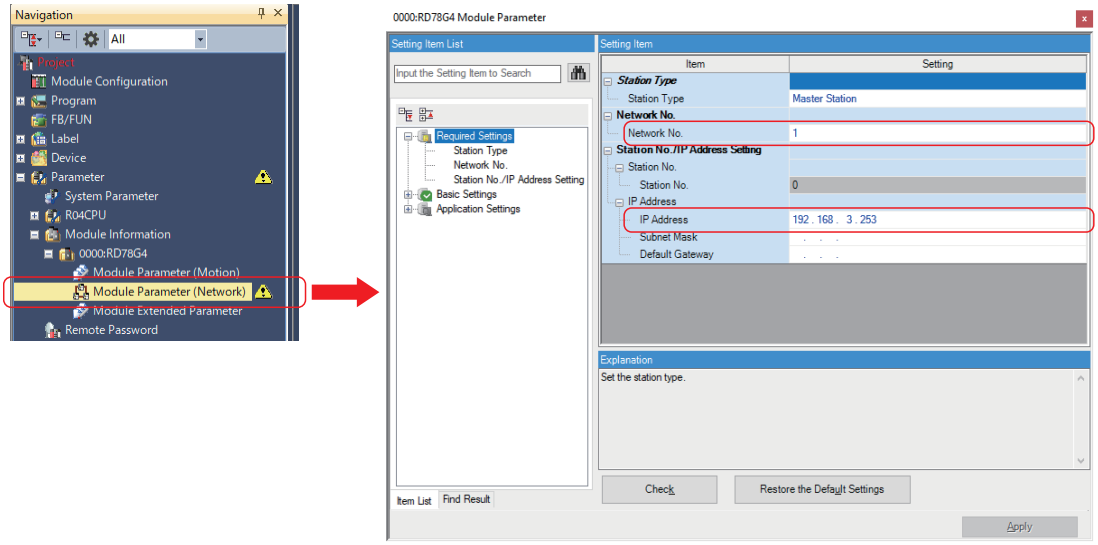
Set slave devices, such as servo amplifiers, to be connected to CC-Link IE TSN.

Setting a network configuration

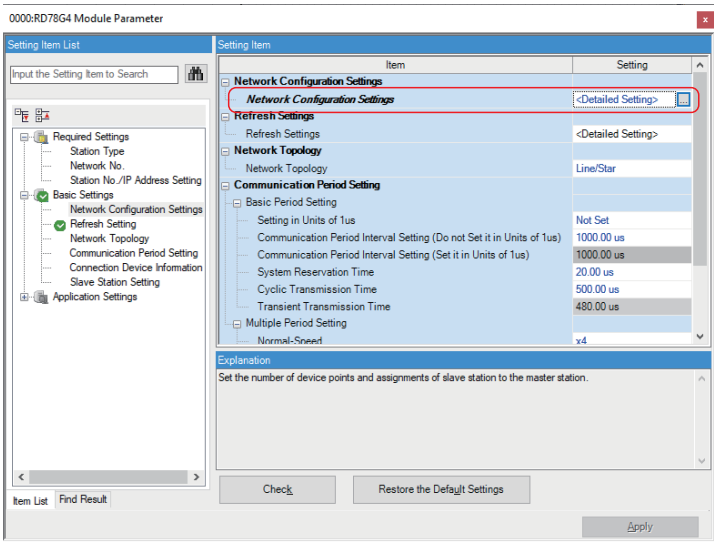
1. In the navigation window, double-click "Parameter"⇨"Module Information"⇨"Module model name*1"⇨"Module Parameter (Network)" to display the parameter editor (Module Parameter). In "Required Settings", set "Network No." and "IP Address".

The example in this document uses the initial value. (Network No.: 1, IP Address: 192.168.3.253)

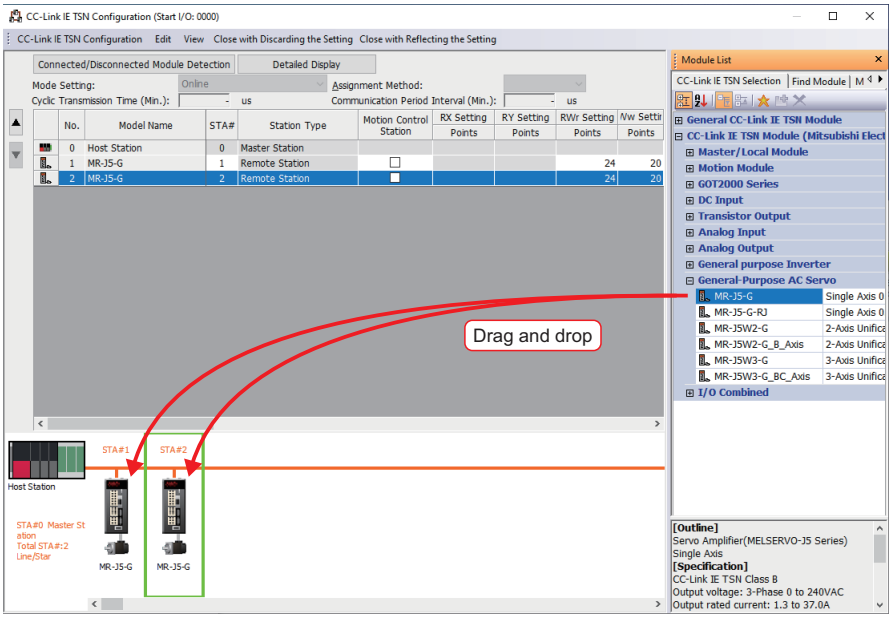
*1 The model name of the set Motion module is displayed.



2. Double-click "Basic Settings"⇨"Network Configuration Settings"⇨"<Detailed Setting>" in the parameter editor (Module Parameter).



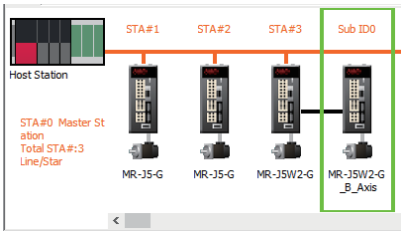
3. The "CC-Link IE TSN Configuration" window is displayed. In the module list, click the [CC-Link IE TSN Selection] tab, then select the devices to use and drag and drop them into their assigned position.
The initial values are displayed in the station No. and the I/O points.
The IP address is assigned automatically based on the master station setting, the master station No., and the order the module is dropped.
The setting example in this document uses two servo amplifiers. Drag and drop MR-J5-G twice.



*1 When adding devices which are not listed in the module list, register the profiles. (☞ Page 24 Profile registration)

Point

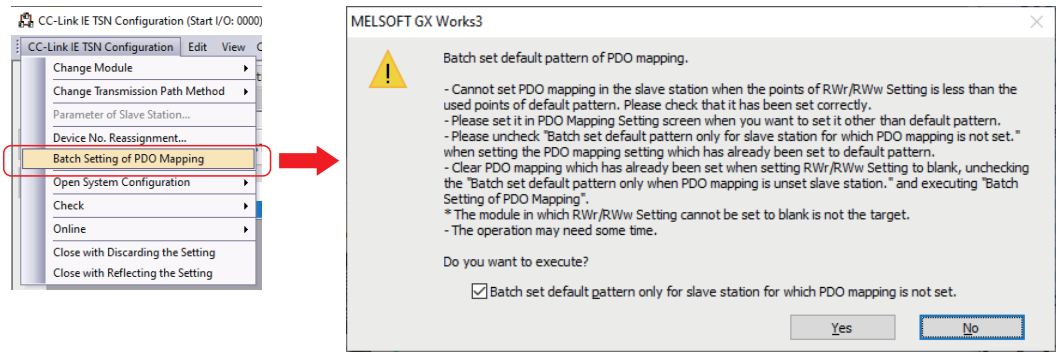
When the user drag and drops a multi-axis servo amplifier (MR-J5W□-G), the B-axis and C-axis are automatically set along with the A-axis.



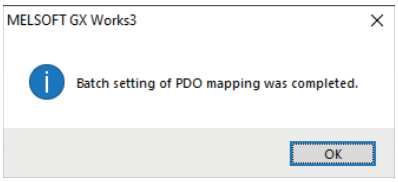
PDO mapping

Set default data of the PDO mapping in a batch.

1. Select [CC-Link IE TSN Configuration]⇨[Batch Setting of PDO Mapping] from the menu.
   When the configuration message appears, click [Yes] to set the default pattern of the PDO mapping in a batch.



2. When the PDO mapping batch setting completes, the completed message appears. Click the [OK] button.

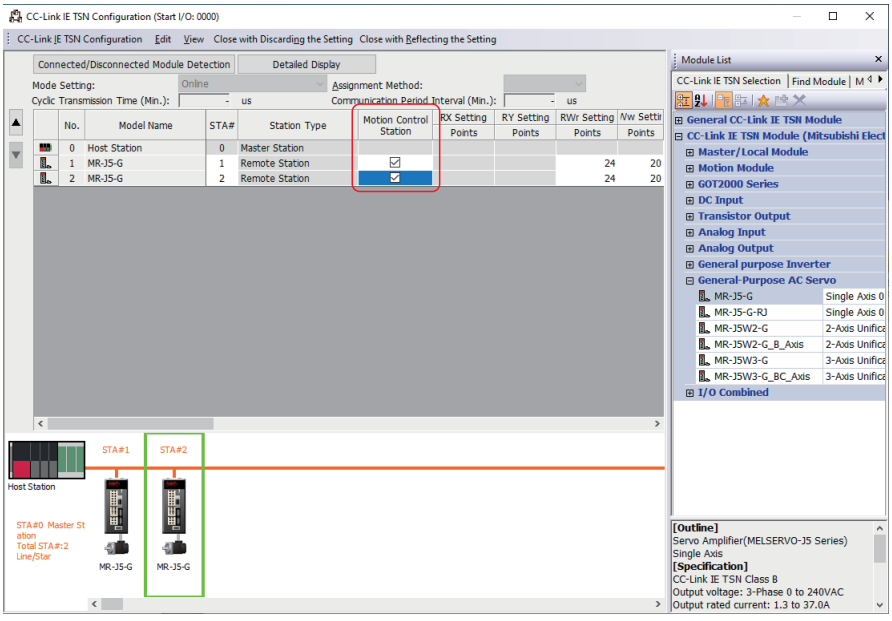


Point

PDO is an abbreviation for "Process Data Object".
PDO communication is equivalent to the conventional CC-Link cyclic communication.
PDO mapping performs mapping (associating) of the data (objects) sent/received between the controller and the slaves in cyclic communication (PDO communication).

Setting a Motion control station

Place a check in the "Motion Control Station" checkbox for modules controlled by the Motion module (servo amplifiers and I/O modules, etc).



3.7 Servo Parameter Setting

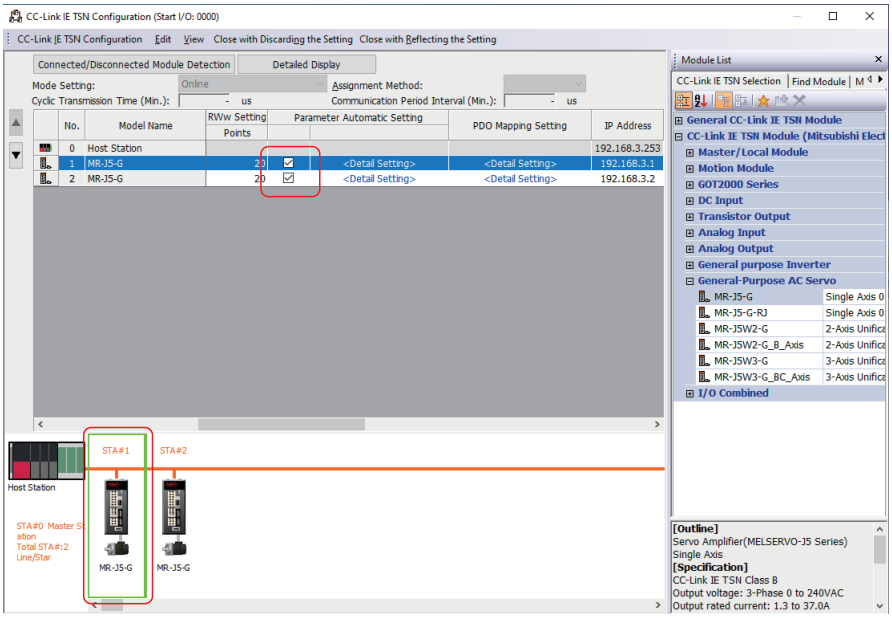
There are two methods for writing parameters to servo amplifiers.

Writing method	Advantage	Disadvantage
Writing to a PLC CPU from MELSOFT GX Works3	• Batch-management of parameters • Setting parameters without connecting servo amplifiers	• Taking time at the initial communication
Writing to servo amplifiers from MR Configurator2	• Quick startup of servo amplifiers since transmission of parameters is not necessary at power-on	• Necessary to write parameters to each servo amplifier

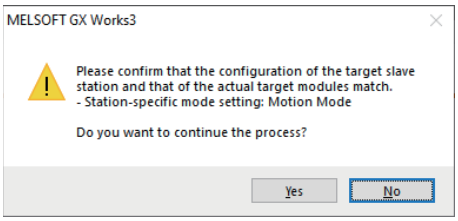
The following shows the procedure for writing servo parameters to a PLC CPU from MELSOFT GX Works3.

Procedure for writing to a PLC CPU from MELSOFT GX Works3

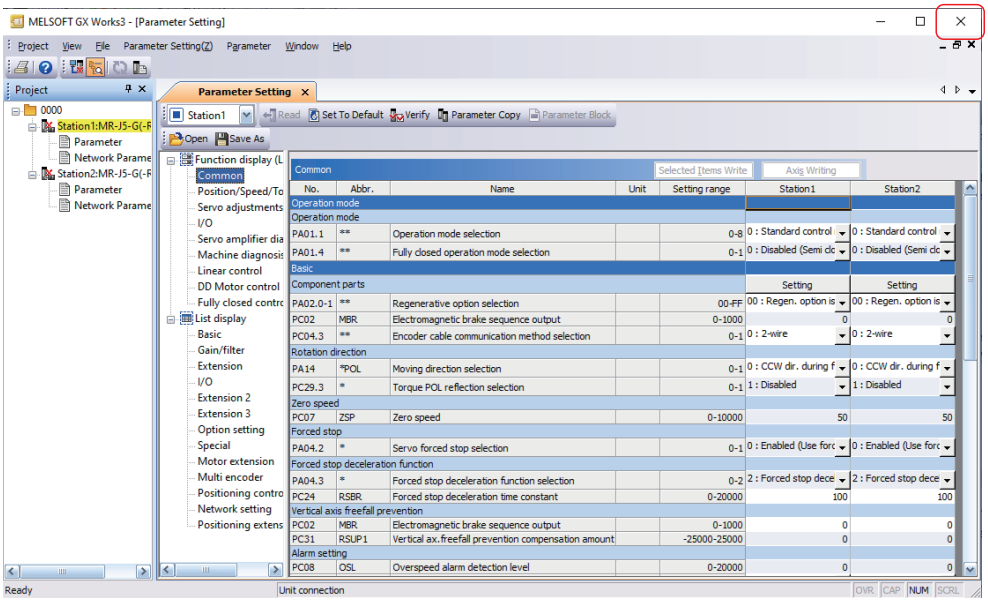
1. Place a check in the "Parameter Automatic Setting" checkbox in the "CC-Link IE TSN Configuration" window to write parameters to the slave stations from the master station at the initial communication.
2. Double-click the illustration of the servo amplifier to be set.



3. When the confirmation message appears, click the [Yes] button.



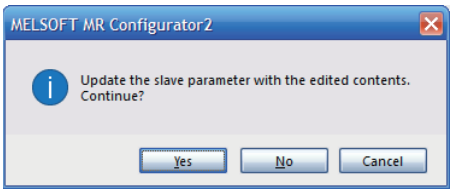
4. The "Parameter Setting" screen is displayed. Set the servo parameters. When the setting is complete, click the [×] button in the upper right corner of the "Parameter Setting" screen.



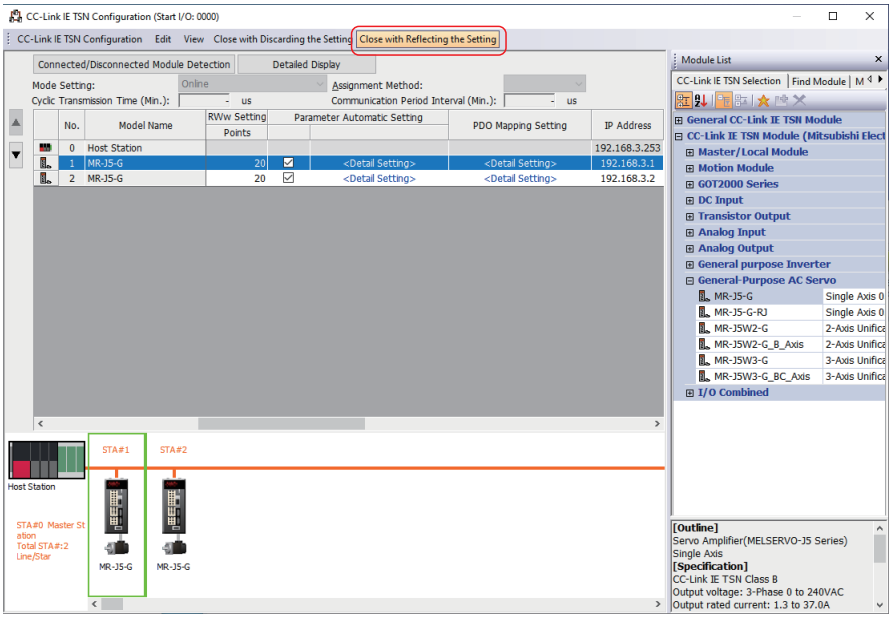
The example in this document changes the following parameters.

Operation	No.	Item	Setting value
[Common]⇨[Basic]⇨[Forced stop]	PA04.2	Servo forced stop selection	0: Enabled (Use forced stop input EM1 or EM2) → 1: Disabled (The forced stop input EM1 and EM2 are not used)
[I/O]⇨[Stroke limit function]⇨[Stroke limit function]	PC19.0	[AL.099 Stroke limit warning] selection	0: Enabled → 1: Disabled
	PD41.2	Limit switch enabled status selection	0: Limit switch always enabled → 1: Enabled only for homing mode

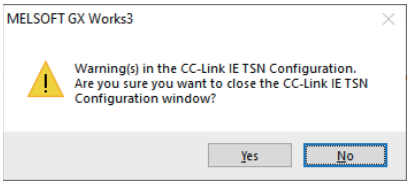
5. When the confirmation message appears, click the [Yes] button to update the parameters.



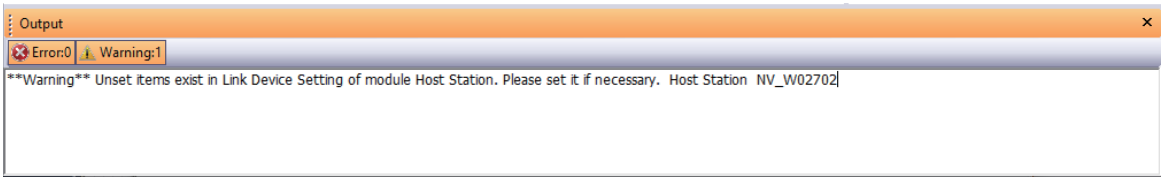
6. Return to the "CC-Link IE TSN Configuration" window. Select [Close with Reflecting the Setting] from the menu. Reflect the data, then close the "CC-Link IE TSN Configuration" window.



A warning message appears on some versions. If a warning message appears, click the [Yes] button.

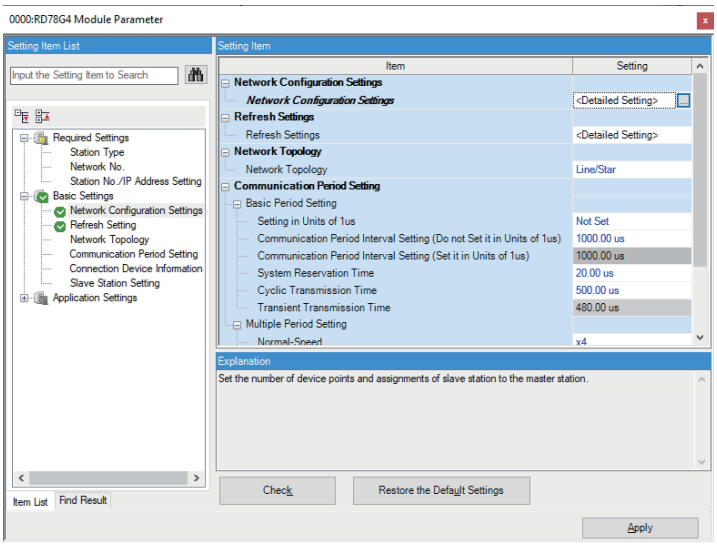


A description of the displayed warning message is shown in the output window.
If the description matches the one shown below, the displayed warning does not indicate a problem.



3.8 Module Parameter Application

Click the [Apply] button in the parameter editor (Module Parameter) to reflect the parameters of the Motion module.



Point

Note that changes made to parameters are not reflected unless the [Apply] button is clicked.

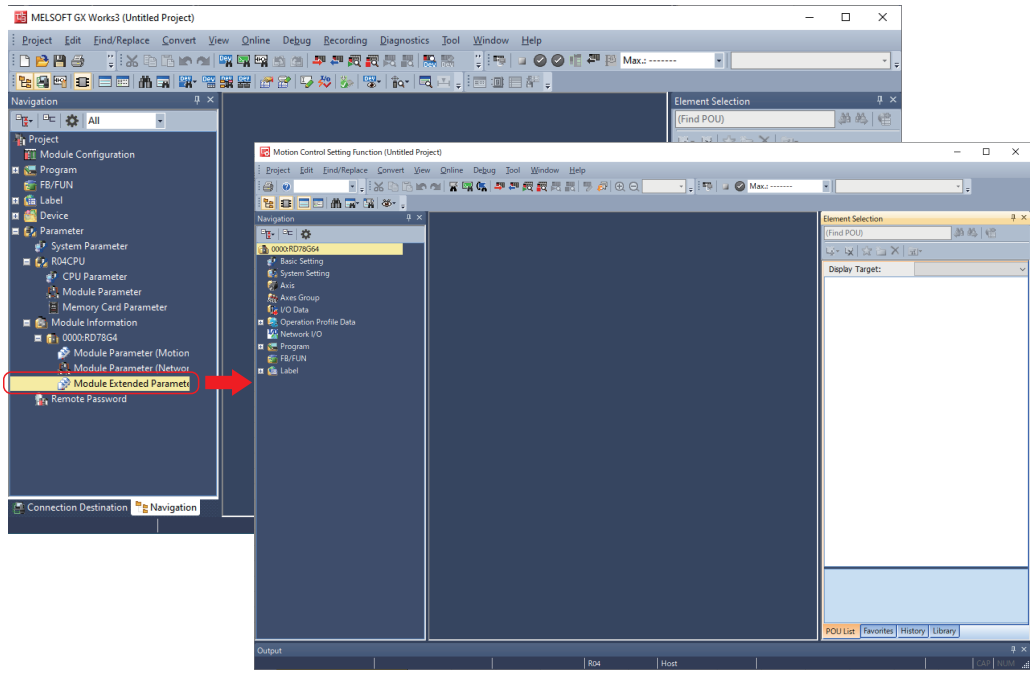
3.9 Motion Control Setting

Set parameters and create programs for the Motion module on the "Motion Control Setting Function" screen.

Double-click "Parameter"⇨"Module Information"⇨"Module model name*1"⇨"Module Extended Parameter" in the MELSOFT GX Works3 navigation window to display the "Motion Control Setting Function" screen.

The set items are saved in the MELSOFT GX Works3 project.

*1 The name of the set Motion module is displayed.



Point

If double-clicking "Module Extended Parameter" does not display the "Motion Control Setting Function" screen, there is a high probability that Motion Control Setting still needs to be installed. For installation, refer to the following.

☞ Page 23 Motion Control Setting installation

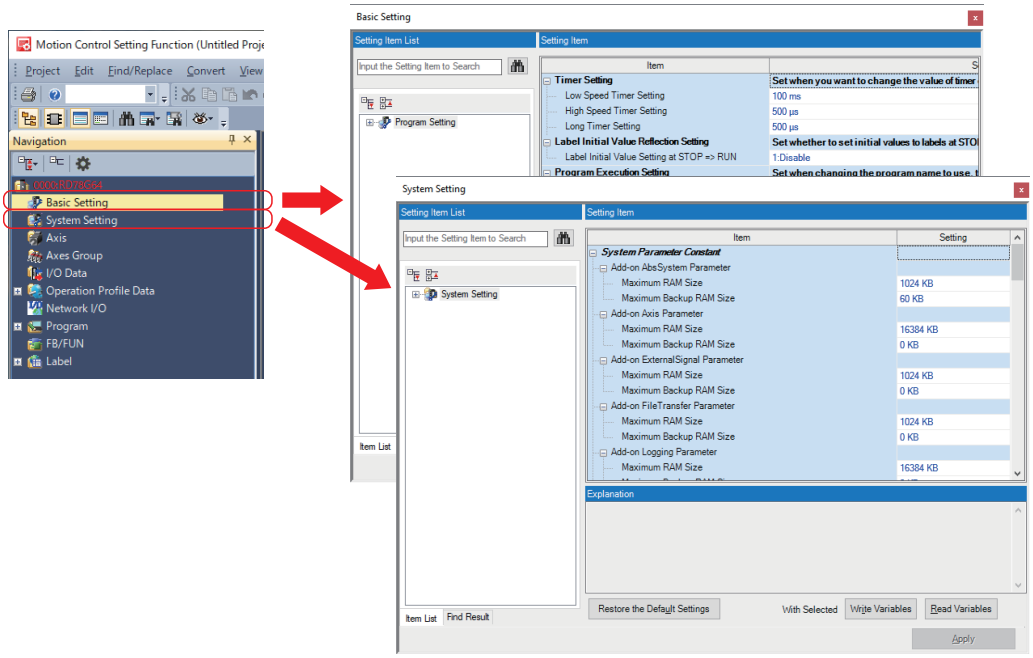
3.10 Basic Setting and System Setting

In the basic setting, set each parameter for the programs.

In the system setting, set the all axes forced stop and each parameter for the add-ons.

In Motion Control Setting Function, double-click "Basic Setting" or "System Setting" in the navigation window to display the "Basic Setting" screen or "System Setting" screen.

Change the parameters as necessary. (The example in this document uses the initial values.)



3.11 Axis Parameter Setting

Setting a station address and an axis type

The following five axis types are available.

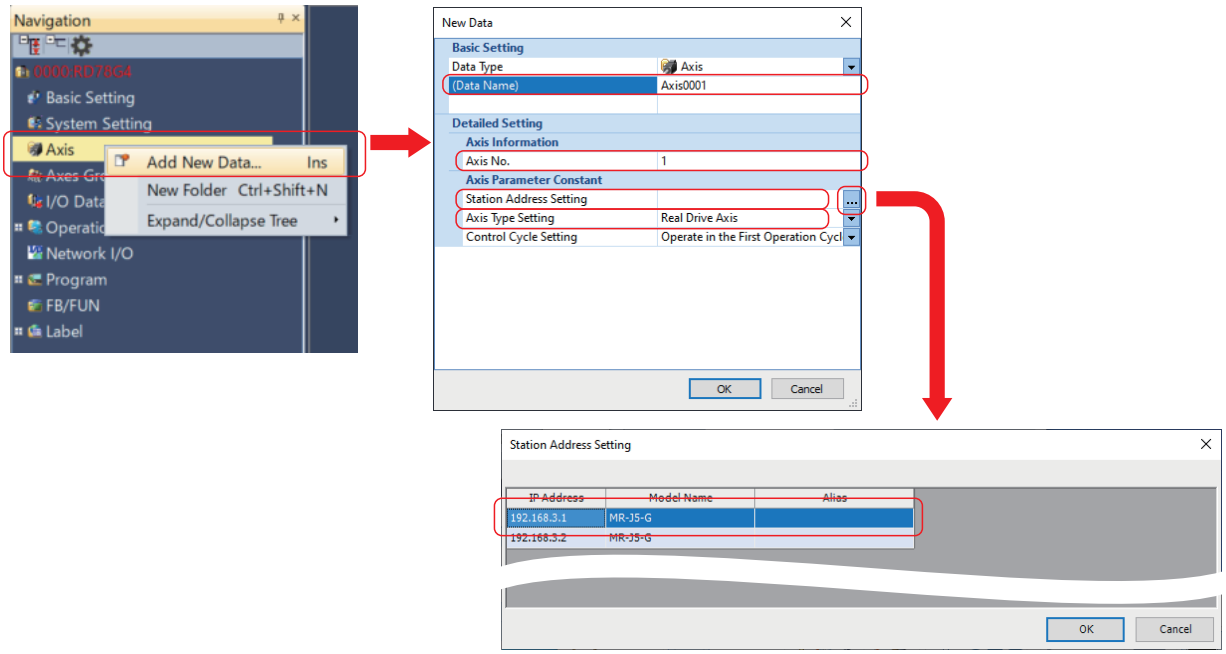
Axis type	Description
Real drive axis	Outputs commands using a servo amplifier connected to CC-Link IE TSN.
Real encoder axis	Generates the current position from the output pulses of the synchronous encoder which is connected to a servo amplifier on CC-Link IE TSN.
Virtual drive axis	Virtually generates commands.
Virtual encoder axis	Virtually generates the current position from the variables.
Virtual linked axis	Virtually connects the single axis synchronous control FBs.

The example in this document registers two real drive axes, one virtual drive axis, and two virtual linked axes.

Real drive axis

Set the station address to link the axis information in the axis parameter to the servo amplifier set in the network configuration.

1. Right-click "Axis" in the navigation window of Motion Control Setting Function, then select [Add New Data].
2. The "New Data" screen is displayed. Enter the data name, the Axis No., the station address setting, and the axis type setting, then click the [OK] button. Click the station address setting [...] button to display the "Station Address Setting" screen. The station address can be set by selecting one of the IP addresses displayed on the "Station Address Setting" screen and clicking the [OK] button.



■Real drive axis setting description

Setting item	First axis	Second axis
Data name	Axis0001	Axis0002
Axis No.	1	2
Station address setting	192.168.3.1	192.168.3.2
Axis type setting	Real drive axis	Real drive axis

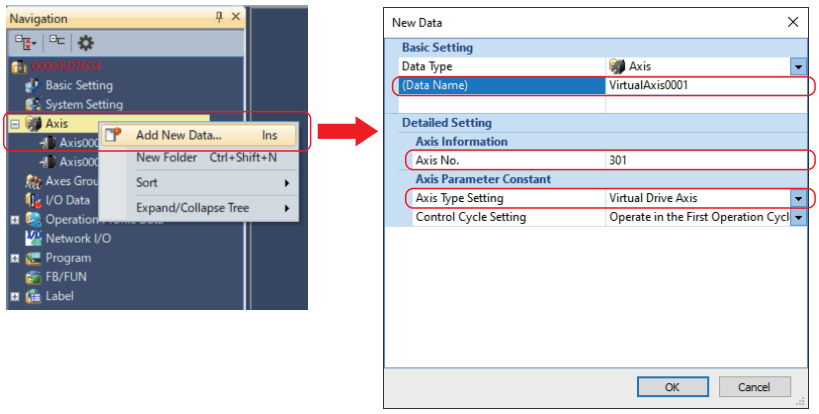
Point

For the station address setting, use the IP address set in the "CC-Link IE TSN Configuration" window.
A multi-drop No. is shown when the station address for the second or third axis of a multi-axis servo amplifier is specified.
(Example) For B-axis of MR-J5-W3 with "IP address: 192.168.3.1"

192.168.3.1#1
- Multi-drop No. (#0: A-axis, #1: B-axis, #2: C-axis)
- IP address

Virtual drive axis

1. Right-click "Axis" in the navigation window of Motion Control Setting Function, then select [Add New Data].
2. The "New Data" screen is displayed. Enter the data name, the Axis No., and the axis type setting, then click the [OK] button.

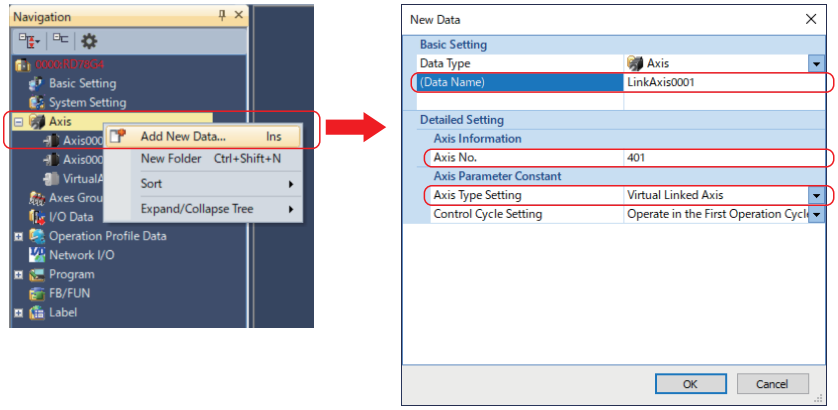


■Virtual drive axis setting description

Setting item	First axis
Data name	VirtualAxis0001
Axis No.	301
Axis type setting	Virtual drive axis

Virtual linked axis

1. Right-click "Axis" in the navigation window of Motion Control Setting Function, then select [Add New Data].
2. The "New Data" screen is displayed. Enter the data name, the Axis No., and the axis type setting, then click the [OK] button.



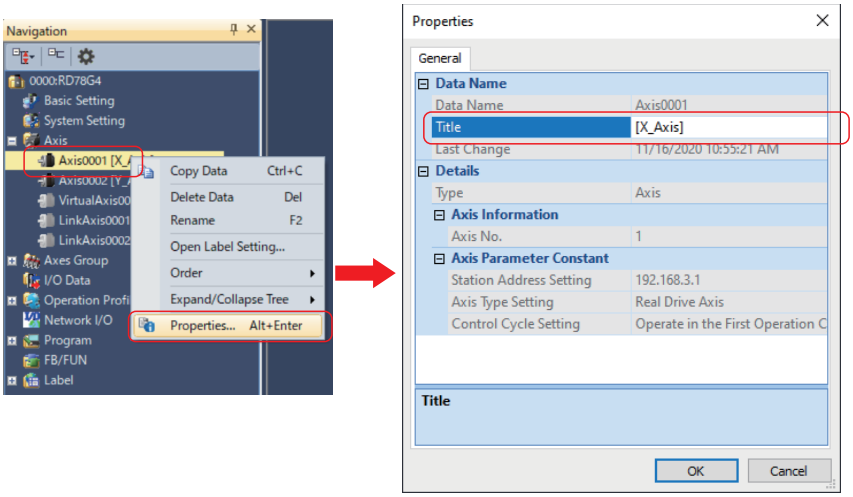
■Virtual linked axis setting description

Setting item	First axis	Second axis
Data name	LinkAxis0001	LinkAxis0002
Axis No.	401	402
Axis type setting	Virtual linked axis	Virtual linked axis

Axis comment setting

Set a comment for each registered axis. Set the necessary comments.

1. Right-click "Axis"⇨"Axis0001" in the navigation window of Motion Control Setting Function, then select [Properties].
2. The "Properties" screen is displayed. Enter the comment in Title, then click the [OK] button.



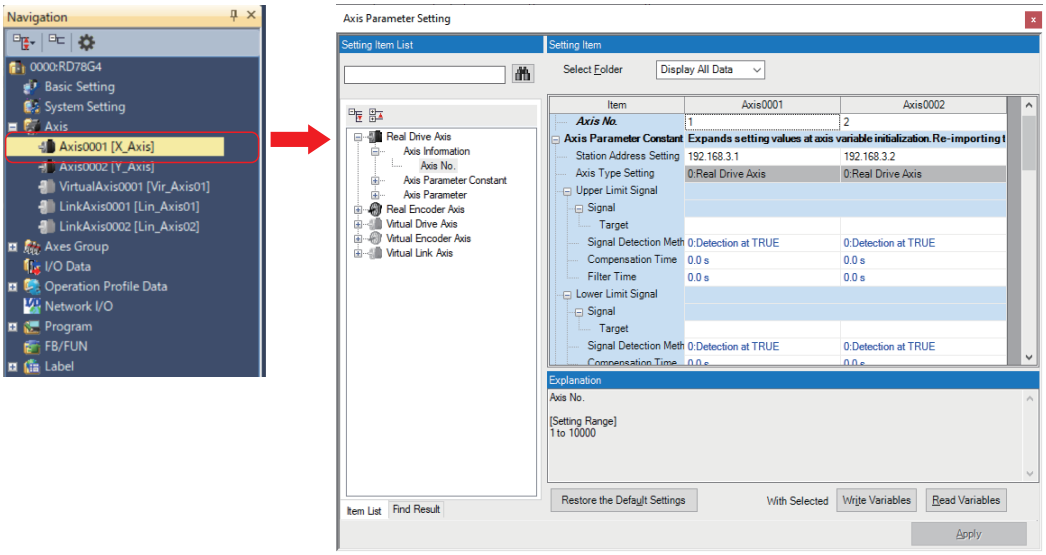
3. In the same manner, right-click "Axis"⇨"Axis0002", "VirtualAxis0001", "LinkAxis0001", and "LinkAxis0002" in the navigation window, select [Properties] to display the "Properties" screen, then enter the comment for the registered axis.

The example in this document sets the following comments for each axis.

Axis type setting	Data name	Title
Real drive axis	Axis0001	[X_Axis]
	Axis0002	[Y_Axis]
Virtual drive axis	VirtualAxis0001	[Vir_Axis01]
Virtual linked axis	LinkAxis0001	[Lin_Axis01]
	LinkAxis0002	[Lin_Axis02]

Real drive axis

1. Double-click "Axis"⇨"Axis0001" in the navigation window to open the "Axis Parameter Setting" window.
Set the axis parameters of Axis0001 and Axis0002.



The setting example for each real drive axis (Axis0001 and Axis0002) in this document is shown below.

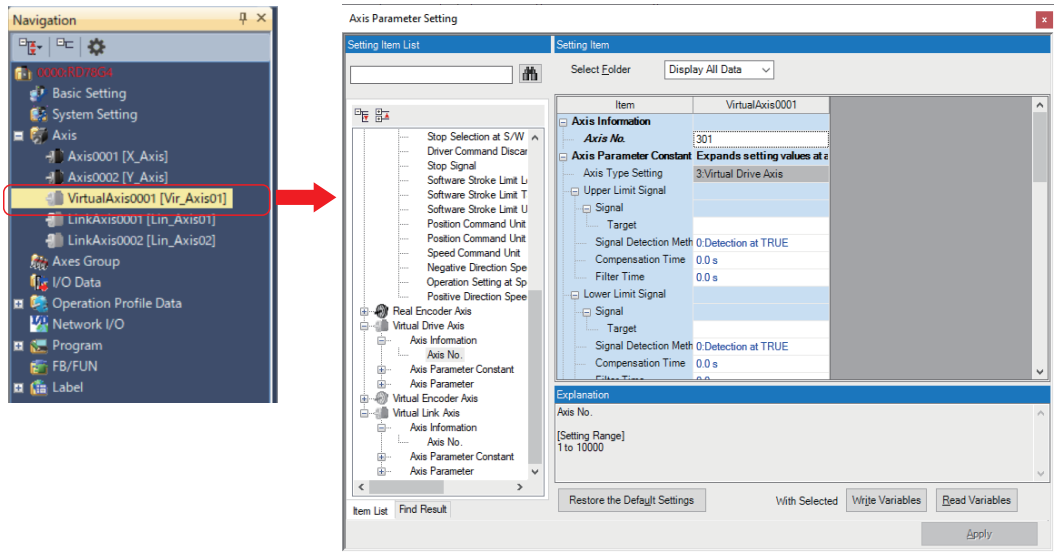
Item				Description	Setting value	
					Axis0001	Axis0002
Axis parameter constant	Station address setting			Set the IP address of the driver.	192.168.3.1	192.168.3.2
	Upper limit signal			Set the hardware stroke limit switches at the upper/lower limit of the movable range by assigning external signals.	Initial value *: Refer to the following for the setting method. ☞ Page 149 Using External Input Signals	
	Lower limit signal					
Axis parameter	Driver unit conversion numerator			Convert units of the target position and the feedback position between the controller and the driver. *: Refer to the following for the setting method. ☞ Page 52 Driver unit conversion (electronic gear)	67108864	
	Driver unit conversion denominator				5000	
	Forced stop signal			Set the forced stop signal.	Initial value	
	Stop signal	Signal	Target	Set the stop signal.	[VAR]G_bStopSignalX*1	[VAR]G_bStopSignalY*1
		Signal detection method			Initial value	
		Compensation time				
		Filter time				
	Start permission at homing uncompleted			Set whether axis start is allowed or not for when the homing request is TRUE.	0: Disabled	
	Software stroke limit lower limit value			Set the range and the target of the software stroke limit.	Initial value	
	Software stroke limit target					
	Software stroke limit upper limit value					
	Position command unit			Set the position command unit and the velocity command unit used for the Motion control. *: The units used in this document are as follows: [um] for the position command unit and [um/s] for the velocity command unit.	um*2	
	Velocity command unit				U/s	

*1 A global label is registered for the stop signal. For setting the global labels, refer to the following.
☞ Page 82 Global label list/settings for public labels

*2 "um" indicates micrometer (μm).

Virtual drive axis

1. Double-click "Axis"⇨"VirtualAxis0001" in the navigation window to open the "Axis Parameter Setting" window. Set the axis parameters of VirtualAxis0001.



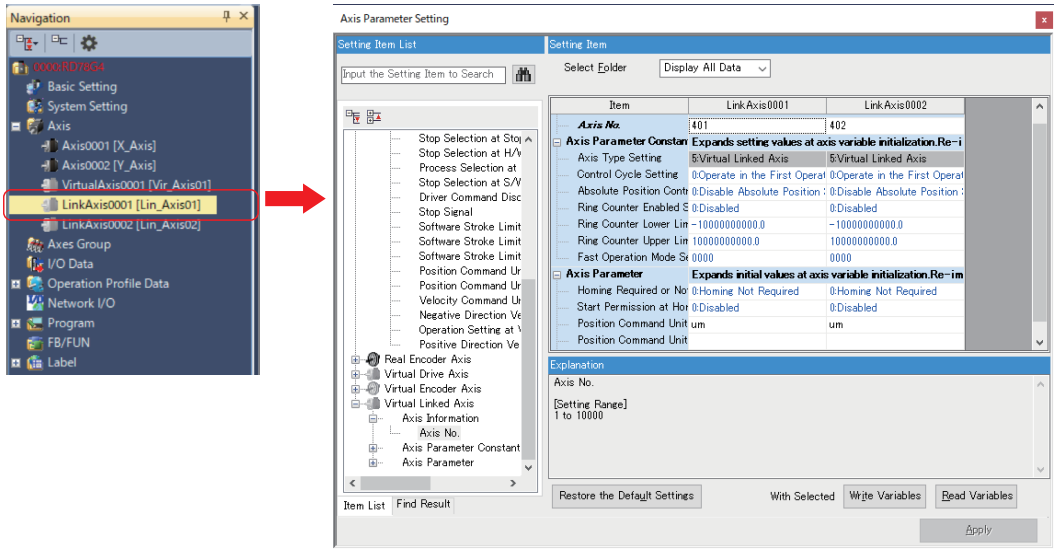
The setting example for the virtual drive axis (VirtualAxis0001) in this document is shown below.

Item		Description	Setting value
			VirtualAxis0001
Axis parameter	Position command unit	Set the position command unit and the velocity command unit used for the Motion control. *: In this example, the same unit as that of the real drive axis is set for the virtual drive axis ([um] for the position command unit and [um/s] for the velocity command unit).	um*1
	Velocity command unit		U/s

*1 "um" indicates micrometer (μm).

Virtual linked axis

1. Double-click "Axis"⇨"LinkAxis0001" in the navigation window to open the "Axis Parameter Setting" window. Set the axis parameters of LinkAxis0001 and LinkAxis0002.



The setting example for each virtual linked axis (LinkAxis0001 and LinkAxis0002) in this document is shown below.

Item		Description	Setting value	
			LinkAxis0001	LinkAxis0002
Axis parameter	Position command unit	Set the position command unit and the velocity command unit used for the Motion control. *: In this example, the same unit as that of the real drive axis is set for the virtual linked axes ([um] for the position command unit and [um/s] for the velocity command unit).	um*1	um*1
	Velocity command unit		U/s	U/s

*1 "um" indicates micrometer (μm).

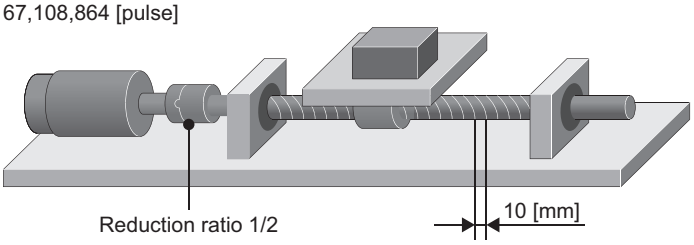
Driver unit conversion (electronic gear)

A setting example of the driver unit conversion numerator/driver unit conversion denominator is shown below.

About the electronic gear

■Ball screw

67,108,864 [pulse]



Item	Setting value
Servo motor encoder resolution	67,108,864 [pulse] (26 bits)
Lead of the ball screw	10000 [um]
Reduction ratio	1/2 (Load side [NL]/Motor side [NM]) Each time the servo motor rotates twice, the ball screw of the load side rotates once.

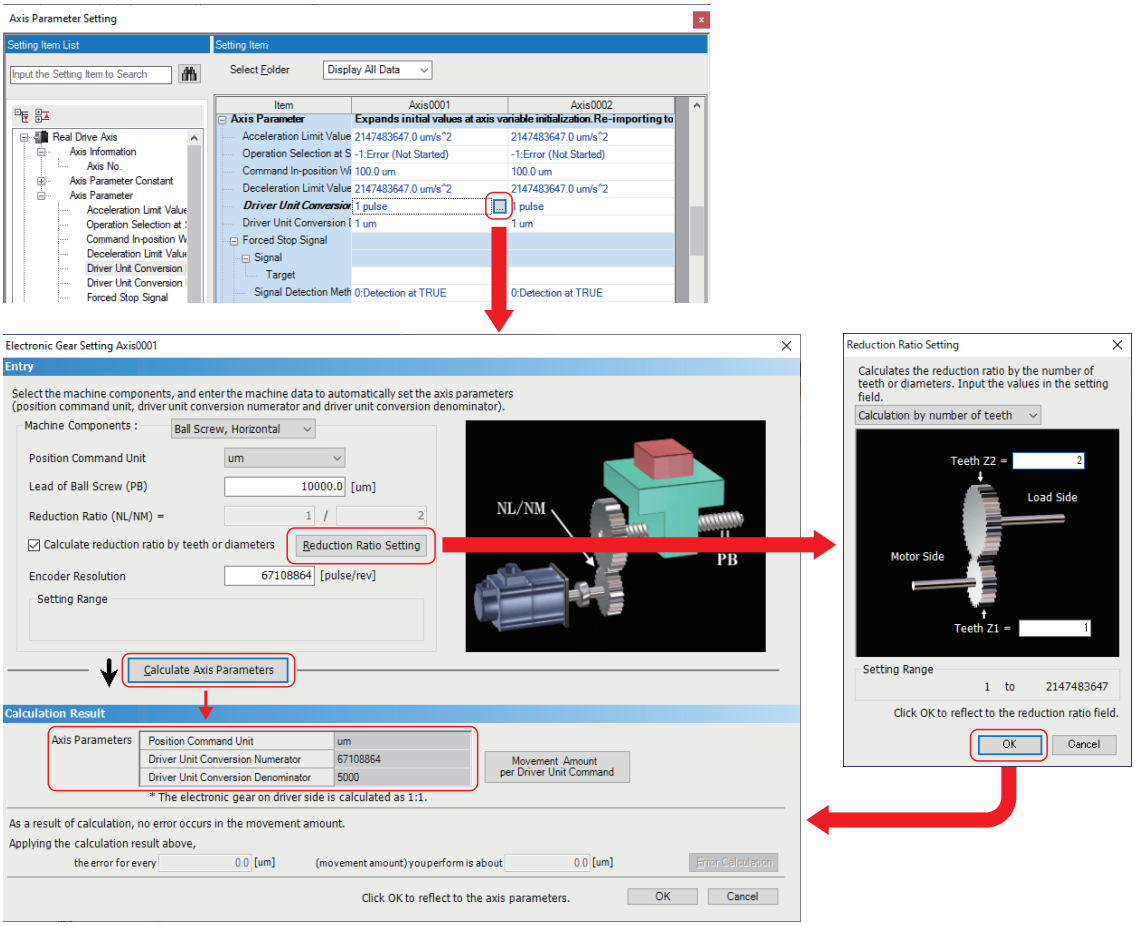
$$\frac{\textit{Driver unit conversion numerator}}{\textit{Driver unit conversion denominator}} = \frac{\textit{Number of encoder pulses}}{\textit{Movement amount} \times \textit{Reduction ratio}} = \frac{67108864}{10000 \times 1/2} = \frac{67108864}{5000}$$

- Driver unit conversion numerator = Number of pulses per rotation 67,108,864
- Driver unit conversion denominator = Movement amount per rotation 5000

Electronic gear setting

The procedure for setting the electronic gear is shown below.

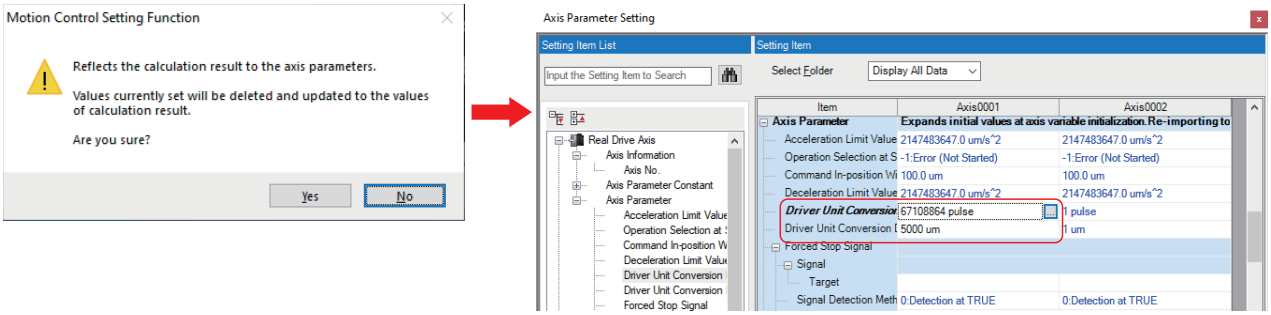
1. "In the "Axis Parameter Setting" window, select "Axis Parameter"⇨"Driver Unit Conversion Numerator" or "Driver Unit Conversion Denominator". Click the [...] button to display the "Electronic Gear Setting" screen.



2. Enter data of the machine components. Click the "Reduction Ratio Setting" button to open the "Reduction Ratio Setting" screen. Set the number of teeth, then click the [OK] button.

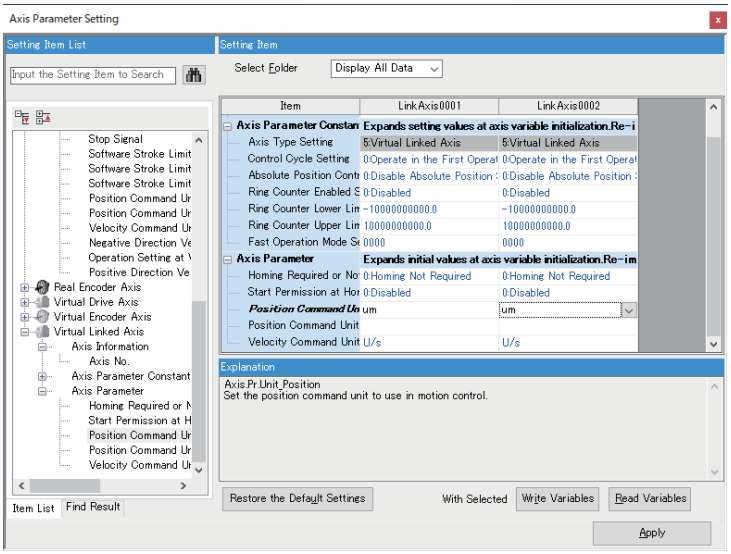
3. Return to the "Electronic Gear Setting". Click the [Calculate Axis Parameters] button to display the calculation result. Confirm the calculation result, then click the [OK] button.

4. When the confirmation message appears, click the [Yes] button to reflect the calculation result to the axis parameters.



Reflecting axis parameters

After setting each item, click the [Apply] button to fix the parameters.



Point

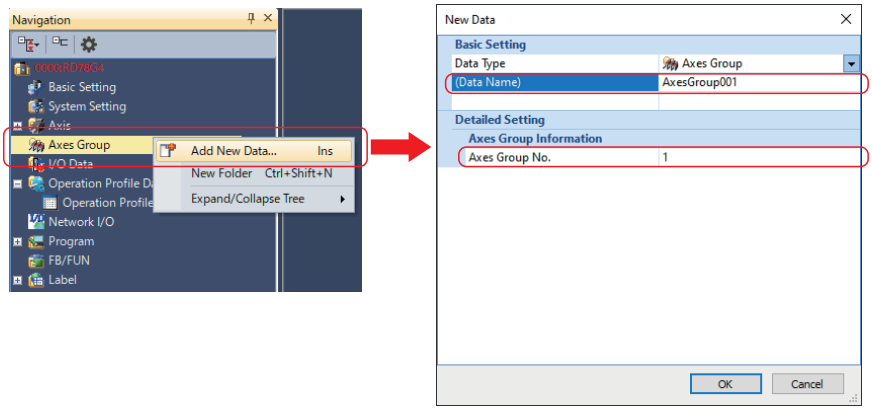
Note that changes made to parameters are not reflected unless the [Apply] button is clicked.

3.12 Axes Group Setting

An axes group is used for multiple axes control such as linear interpolation and circular interpolation.
An axes group does not need to be created for axes used for synchronous control.
The X axis (Axis0001) and Y axis (Axis0002) of the two-axis machine example in this document (☞ Page 19 System Overview) are registered in an axes group.

Axes group

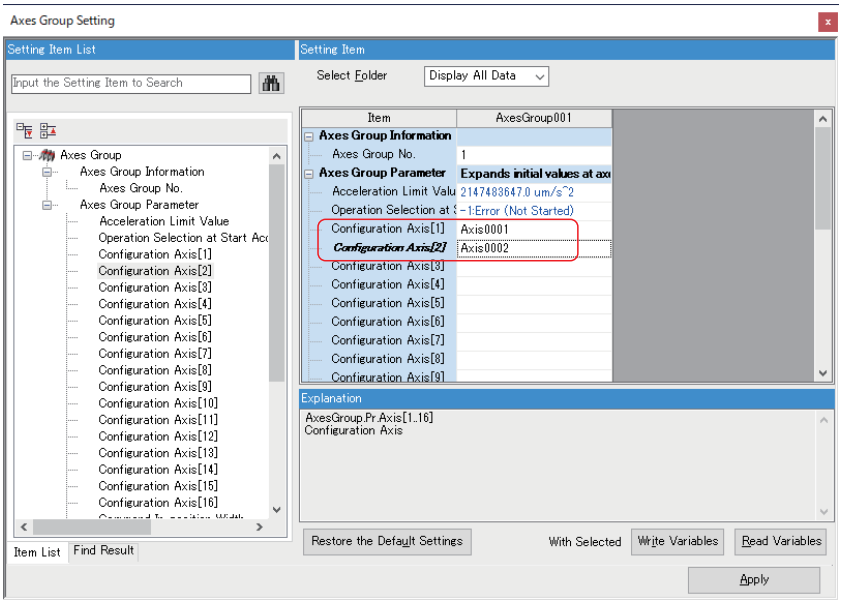
1. Right-click "Axes Group" in the navigation window of Motion Control Setting Function, then select [Add New Data].
2. The "New Data" screen is displayed. Enter the data name and the axes group No., then click the [OK] button.



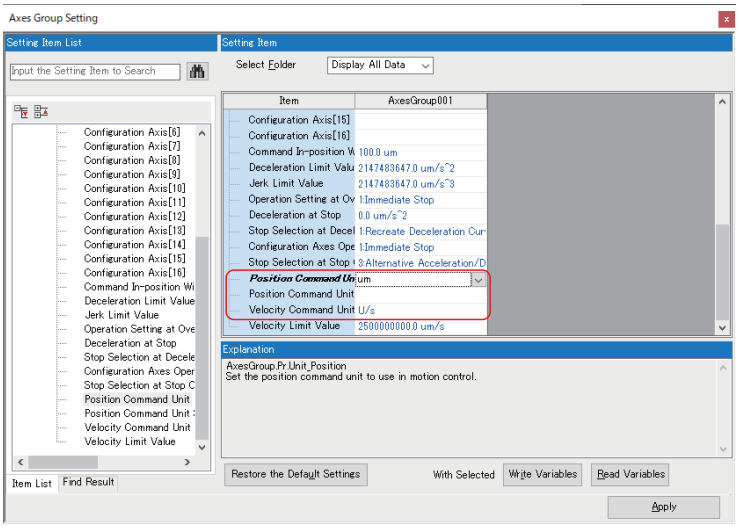
Setting item	Setting value
Data name	AxesGroup001
Axes group No.	1

3. The "Axes Group Setting" window is displayed. Set the axis group parameters of AxesGroup001.

- Setting example of axes group configuration axes (Configuration axis[1]: Axis0001, Configuration axis[2]: Axis0002)



• Setting example of axes group units (Position command unit: [um]*1, Velocity command unit: [U/s])



*1 "um" indicates micrometer (μm).

4. After setting the items for each axes group, click the [Apply] button to fix the axes group parameter setting.

Point

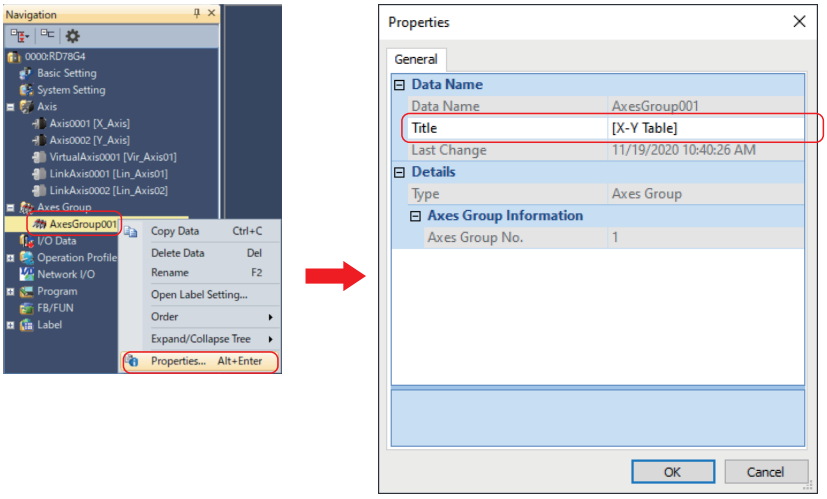
Note that changes made to parameters are not reflected unless the [Apply] button is clicked.

Axes group comment setting

Set a comment for each registered axis group. Set the necessary comments.

1. Right-click "Axes Group"⇨"AxesGroup001" in the navigation window of Motion Control Setting Function, then select [Properties].

2. The "Properties" screen is displayed. Enter the comment in Title, then click the [OK] button.



The example in this document sets the following comment for the axis group.

Data name	Title
AxesGroup001	[X-Y Table]

3.13 Labels

Labels are classified into local labels and global labels.

Select whether the label is registered as a local label or a global label according to your application.

Label	Description
Local label	• Labels that can only be used in one program
Global label	• Labels that can be used in all programs • Labels that can be set to public labels (☞ Page 59 Public Labels)

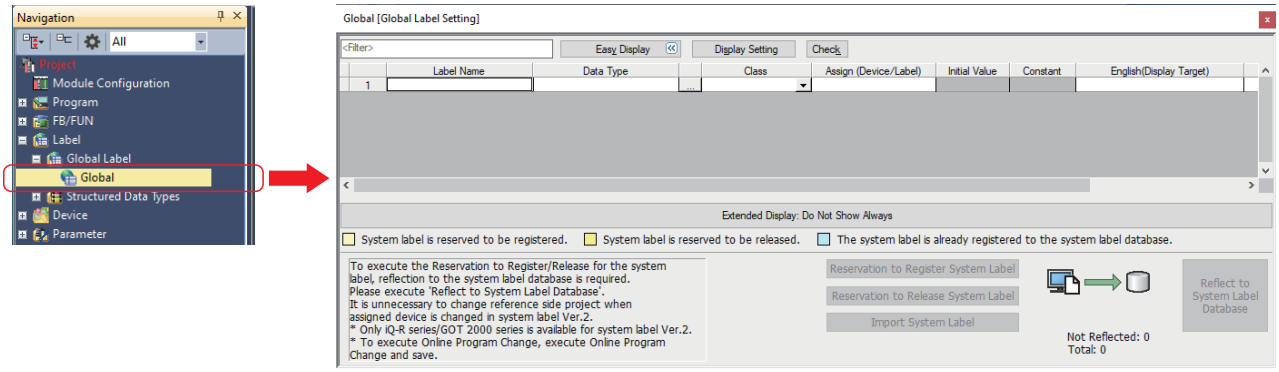
Creating labels

This section describes how to create a label.

For registration examples of labels, refer to the following.

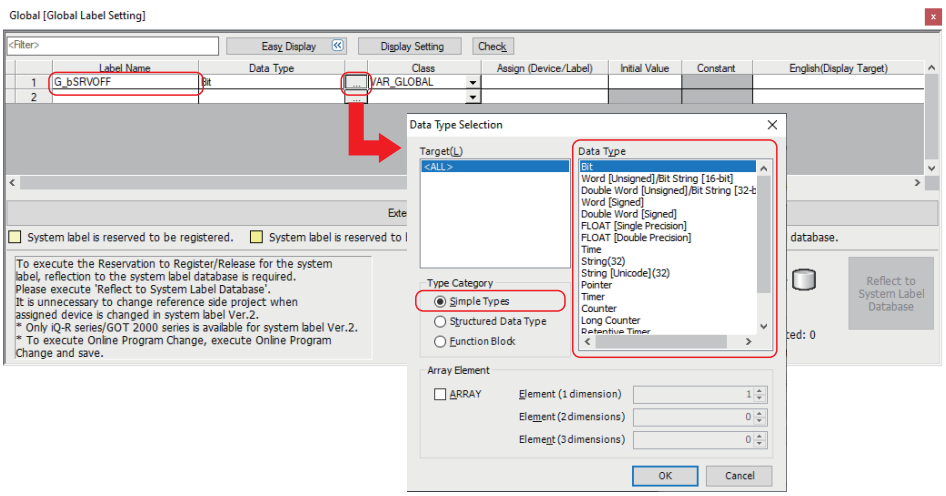
☞ Page 82 Global label list/settings for public labels

1. Double-click "Label"⇨"Global Label"⇨"Global" in the navigation window to display the "Global [Global Label Setting]" screen.



2. Set a global label.

- Enter the label name. (setting example: G_bSRVOFF)
- Select the data type. (Click the [...] button to display the "Data Type Selection" screen.)
  Select "Simple Types" in the type category, select the data type, then click the [OK] button.



■Setting example

- Type category: Simple types
- Data type: Bit

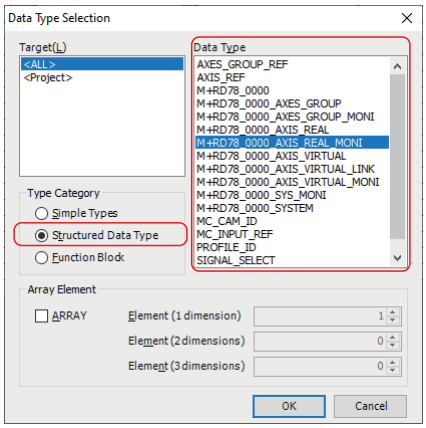
• Set the structured data types in the same manner as the simple types.

If setting arrays, place a check in the ARRAY checkbox, then enter the number of elements in Element.

■Setting example of structured data types

• Type category: Structured data type

• Data type: M+RD78_0000_AXIS_REAL_MONI*1



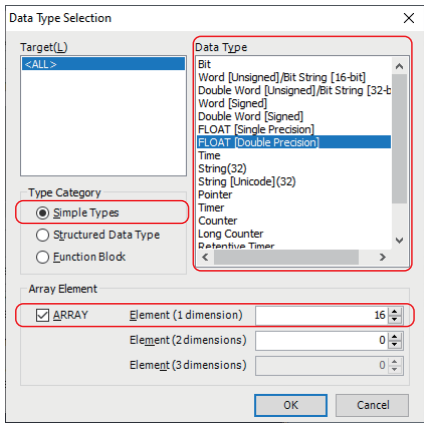
■Setting example of arrays

• Type category: Simple types

• Data type: FLOAT (double precision)

• ARRAY: Checked

• Number of elements: 16



*1 "M+RD78_0000_*****" data types are structured data types on the Motion module side. To use these data types, set them as public labels. For details of public labels, refer to the following.

☞ Page 59 Public Labels

Registration example of global labels

Global labels for the programs used in Chapter 4 are shown below.

For details, refer to the following.

☞ Page 82 Global label list/settings for public labels

■PLC CPU side global labels

Global [Global Label Setting]

<Filter> | Easy Display | Display Setting | Check

	Label Name	Data Type		Class	Assign (Device/Label)	Initial Value	Constant	English(Display Target)	Remark	System L
1	G_bSRVOFF	Bit		VAR_GLOBAL				Servo OFF		
2	G_leJogVelocity	FLOAT [Double Precision]		VAR_GLOBAL				JOG Velocity		
3	G_bJogF1	Bit		VAR_GLOBAL				JOG Positive rotation command Axis0001		
4	G_bJogR1	Bit		VAR_GLOBAL				JOG Reverse rotation command Axis0001		
5	G_bJogF2	Bit		VAR_GLOBAL				JOG Positive rotation command Axis0002		
6	G_bJogR2	Bit		VAR_GLOBAL				JOG Reverse rotation command Axis0002		
7	G_bHoming1CMD	Bit		VAR_GLOBAL				Homing command Axis0001		
8	G_bHoming2CMD	Bit		VAR_GLOBAL				Homing command Axis0002		
9	G_bHoming3CMD	Bit		VAR_GLOBAL				Homing command VirtualAxis0001		
10	G_bHoming1Req	Bit		VAR_GLOBAL				Homing start request Axis0001		
11	G_bHoming2Req	Bit		VAR_GLOBAL				Homing start request Axis0002		
12	G_bHoming3Req	Bit		VAR_GLOBAL				Homing start request VirtualAxis0001		
13	G_bPosReq	Bit		VAR_GLOBAL				Single axis positioning start request		
14	G_bContPosReq	Bit		VAR_GLOBAL				Single axis continuous positioning start request		
15	G_bInterpolationReq	Bit		VAR_GLOBAL				2-axis linear interpolation control start request		
16	G_bSyncReq	Bit		VAR_GLOBAL				Synchronous control start request		
17	G_bPosCMD	Bit		VAR_GLOBAL				Single axis positioning start		
18	G_bContPosCMD	Bit		VAR_GLOBAL				Single axis continuous positioning start		
19	G_bInterpolationCMD	Bit		VAR_GLOBAL				2-axis linear interpolation control start		
20	G_bSyncCMD	Bit		VAR_GLOBAL				Synchronous control start		
21	G_bErrorReset	Bit		VAR_GLOBAL				Error reset		
22	G_bSysErrorReset	Bit		VAR_GLOBAL				System error reset		
23	G_bAxis0001Status	Bit		VAR_GLOBAL				Standby Axis0001		
24	G_bAxis0002Status	Bit		VAR_GLOBAL				Standby Axis0002		
25										

■Motion module side global labels

Global [Global Label Setting]

	Label Name	Data Type	Class	Initial Value	Constant	English(Display Target)	Remark	Public Label	Motion Control Attribute
1	G_bStopSignalX	Bit	VAR_GLOBAL	0		Stop command Axis0001		Enable	WRITE (=> Motion)
2	G_bStopSignalY	Bit	VAR_GLOBAL	0		Stop command Axis0002		Enable	WRITE (=> Motion)
3									

To use Motion module side global labels, set them as public labels. For details of public labels, refer to the following.

☞ Page 59 Public Labels

3.14 Public Labels

A PLC CPU program can read global labels and structure members from the Motion module and write data to global labels and structure members in the Motion module bywhen they are set as public labels.
Monitor data for the position and velocity and other types of data are defined as variables with specified names (labels) in the Motion module. These labels can be used on the PLC CPU when they are set as public labels.
This section describes how to create a public label. For registration examples of actual labels, refer to the following.
☞ Page 82 Global label list/settings for public labels

Public label registration

The following shows how to register Axis information (AxisRef), Axis status (AxisStatus), and the command in-position of the real drive axis as public labels.

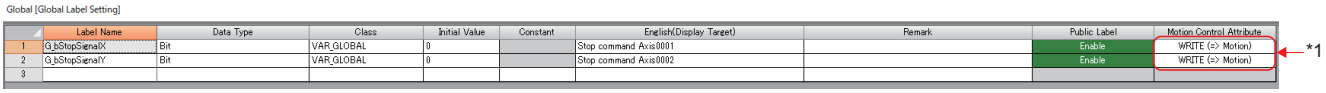
Registering Motion module structures as public labels

1. Double-click "Label"⇨"Global Label"⇨"Ax+Global" in the navigation window of Motion Control Setting Function to display the Ax+Global [Global Label Setting] screen. Set the axis labels being registered as public labels to "Enable".
2. Double-click "Structured Data Types"⇨"AXIS_REAL" to display the "AXIS_REAL [Structure Setting]" screen. Set Axis information (AxisRef) and Axis monitor data (Md) to "Enable".
3. Double-click "Structured Data Types"⇨"AXIS_REAL_MONI" to display the "AXIS_REAL_MONI [Structure Setting]" screen. Set Axis status (AxisStatus) and Command in-position (CmdInPos) to "Enable".

*1 In the sample progam, the virtual drive axis and virtual link axes are also set as public labels.

Setting user-created global labels as public labels

A PLC CPU can read user-created global labels from the Motion module and write data to user-created global labels labels in the Motion module when they are set as public labels.



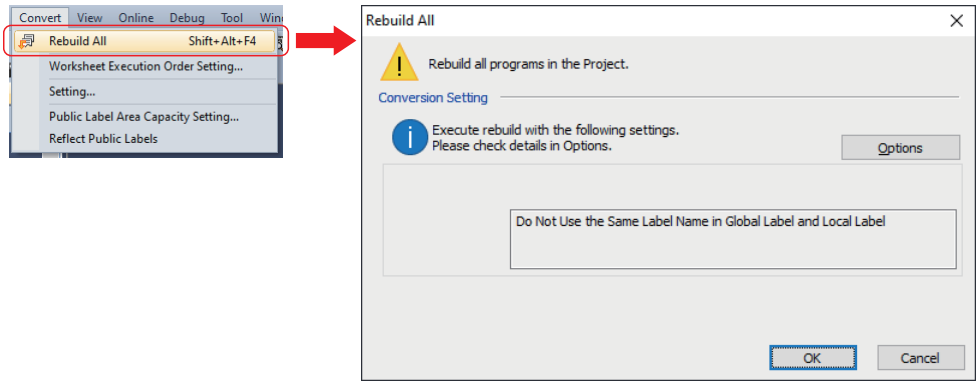
*1 The motion control attribute must be set for user-created global labels.
· Read label data from the Motion module with the PLC CPU: READ (Motion =>)
· Write data from the PLC CPU to labels in the Motion module: WRITE (=> Motion)

Public label reflection

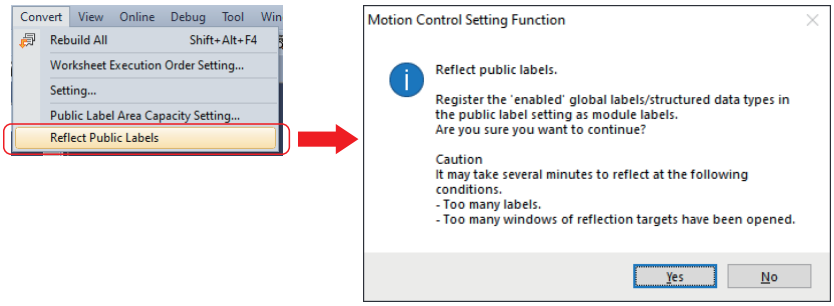
Reflect the registered public labels.

After the public labels are reflected, the PLC CPU can reference the members that were registered as public labels.

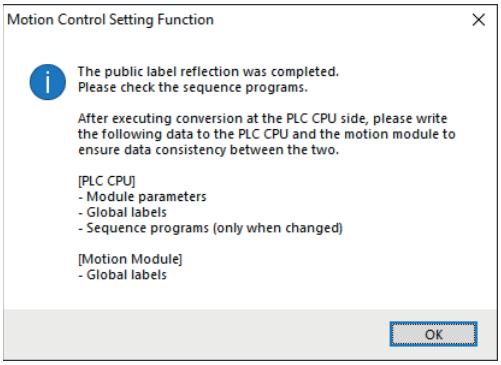
1. In Motion Control Setting Function, select [Convert]⇨[Rebuild All] in the menu. When the confirmation message appears, click the [OK] button to convert all the public labels that were set.



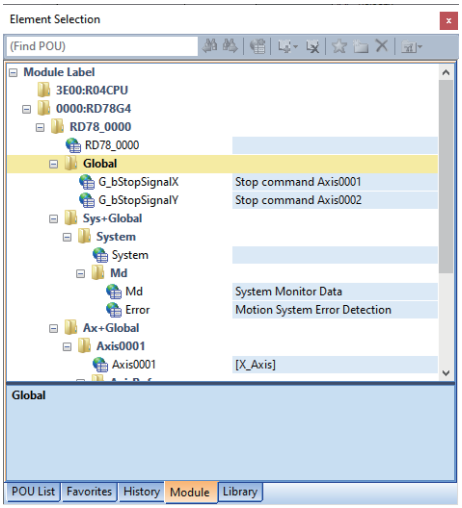
2. Select [Convert]⇨[Reflect Public Labels] in the menu. When the confirmation message appears, click the [Yes] button.



3. When reflection of public labels is complete, a completed message is displayed. Click the [OK] button.



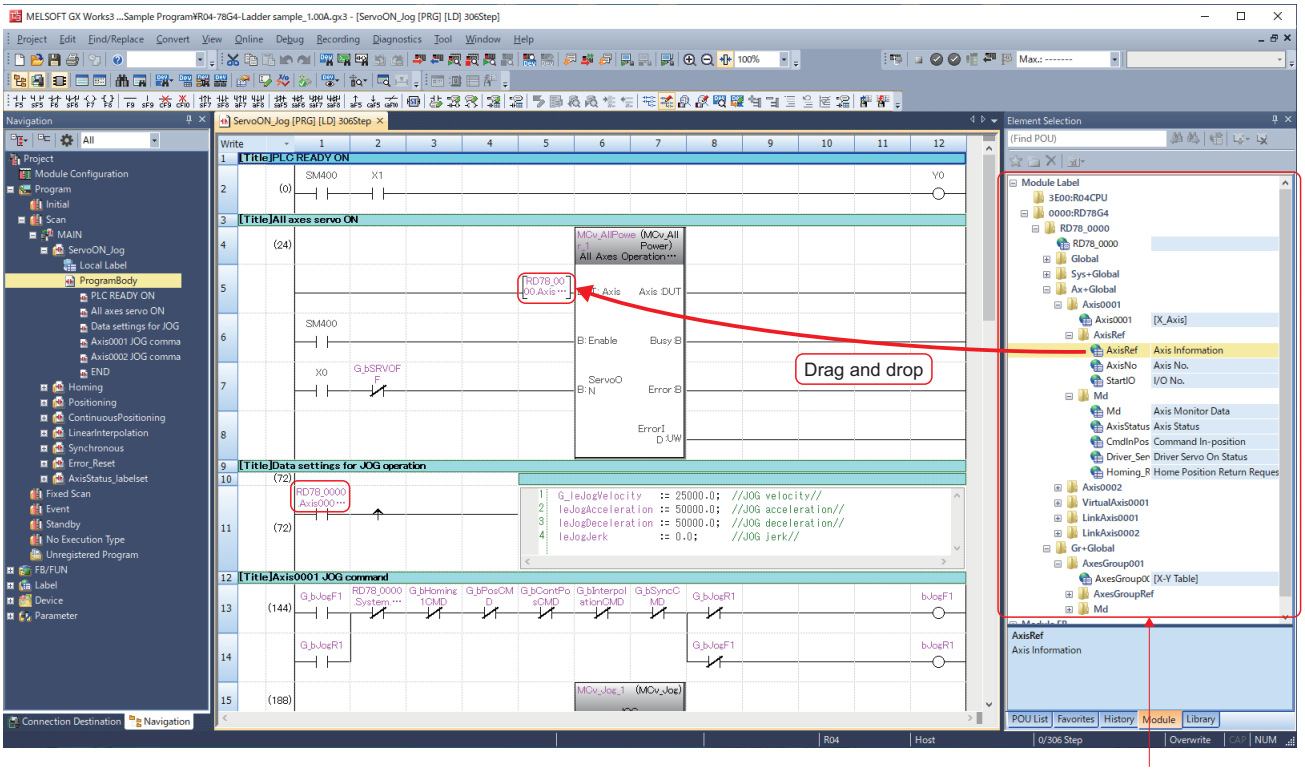
4. To use the public labels on the PLC CPU, convert all the programs by selecting [Convert]⇨[Rebuild All] in the menu on the MELSOFT GX Works3 side.
The reflected public labels are registered as module labels in the element selection window of MELSOFT GX Works3.



A program example using public labels

A program example using public labels is shown below.

1. Double-click "Module Label" from the [Module] tab in the element selection window to display the public labels. To use the public labels, drag and drop them into their assigned position.



Point

When a public label from the Motion module is used on the PLC CPU, "Module + Start I/O No." is added at the start of the name.

(Example) When specifying the label for Axis 1 (Axis0001)

RD78_0000.Axis0001.AxisRef

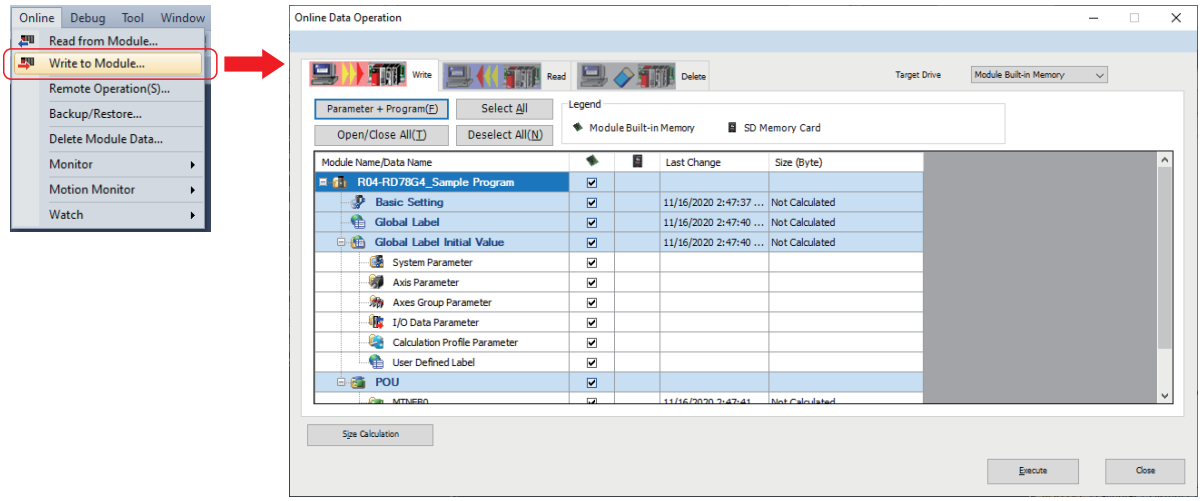
- AxisRef → Member
- Axis0001 → Axis label name
- RD78_0000 → Module + Start I/O No.

3.15 Writing to a Motion Module

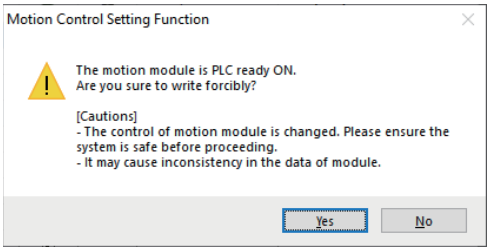
After writing the module parameters to the PLC CPU, reset the PLC CPU, then turn the PLC CPU power ON again to communicate with the Motion module.

After that, write the items set in Motion Control Setting to the Motion module.

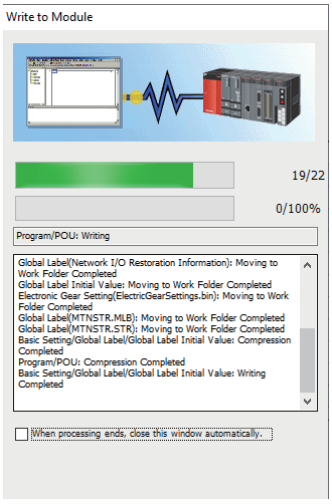
1. Select [Online]⇨[Write to Module] to open the "Online Data Operation" screen. Check the items to be written on the screen, then click the [Execute] button.



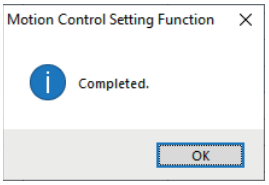
2. When the Motion module is in PLC Ready ON status, the following confirmation message is displayed.
When the Motion module is in PLC Ready OFF status, no confirmation message is displayed.



3. Write the parameters to the Motion module.



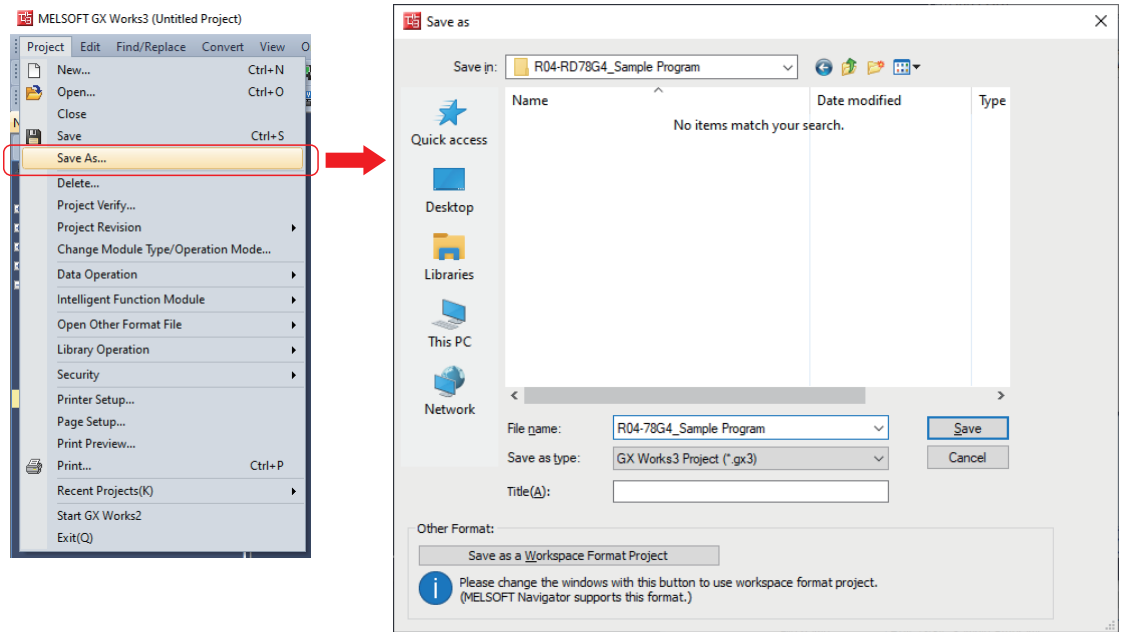
4. When the writing is complete, a completed message is displayed. Click the [OK] button.
Turn the PLC CPU power OFF→ON again and confirm that no error occurs. The error status can be checked with the LED display of the PLC CPU and the Motion module, or with the system monitor in MELSOFT GX Works3.



3.16 Saving a Project

Save the created project.

1. Select [Project]⇨[Save As] on MELSOFT GX Works3 to display the "Save as" screen. Enter a file name, then click the [Save] button.



3.17 Parameter List

The following shows a list of the parameters which are set in this chapter.

For the public label setting, refer to the following.

☞ Page 82 Global label list/settings for public labels

Axis parameter setting

■Real drive axis

Item	Axis0001	Axis0002
Axis Information		
Axis No.	1	2
Axis Parameter Constant	**Expands setting values at axis variable initialization.Re-importing to the control data is not executed after axis variable initializatic**	
Station Address Setting	192.168.3.1	192.168.3.2
Axis Type Setting	0:Real Drive Axis	0:Real Drive Axis
Upper Limit Signal		
Signal		
Target		
Signal Detection Method	0:Detection at TRUE	0:Detection at TRUE
Compensation Time	0.0 s	0.0 s
Filter Time	0.0 s	0.0 s
Lower Limit Signal		
Signal		
Target		
Signal Detection Method	0:Detection at TRUE	0:Detection at TRUE
Compensation Time	0.0 s	0.0 s
Filter Time	0.0 s	0.0 s
Control Cycle Setting	0:Operate in the First Operation Cycle	0:Operate in the First Operation Cycle
Absolute Position Reference Setting	3:Machine Feed Value	3:Machine Feed Value
Absolute Position Control Setting	-1:Automatic Setting (Acquire from Connected Device)	-1:Automatic Setting (Acquire from Connected Device)
Ring Counter Enabled Selection	0:Disabled	0:Disabled
Ring Counter Lower Limit Value	-10000000000.0	-10000000000.0
Ring Counter Upper Limit Value	10000000000.0	10000000000.0
Slave Emulation Enabled	0:Disabled	0:Disabled
Torque Limit Maximum Value	1000.0 %	1000.0 %
Negative Direction Torque Limit Initial Value	300.0 %	300.0 %
Positive Direction Torque Limit Initial Value	300.0 %	300.0 %
Fast Operation Mode Setting	0000	0000
Axis Parameter	**Expands initial values at axis variable initialization.Re-importing to the control data is also executed after axis variable initializatio**	
Acceleration Limit Value	2147483647.0 um/s^2	2147483647.0 um/s^2
Operation Selection at Start Acceleration/Deceleration 0	-1:Error (Not Started)	-1:Error (Not Started)
Command In-position Width	100.0 um	100.0 um
Deceleration Limit Value	2147483647.0 um/s^2	2147483647.0 um/s^2
Driver Unit Conversion Numerator	67108864 pulse	67108864 pulse
Driver Unit Conversion Denominator	5000 um	5000 um
Forced Stop Signal		
Signal		
Target		
Signal Detection Method	0:Detection at TRUE	0:Detection at TRUE
Compensation Time	0.0 s	0.0 s
Filter Time	0.0 s	0.0 s
Homing Required or Not	1:Homing Required	1:Homing Required
Jerk Limit Value	2147483647.0 um/s^3	2147483647.0 um/s^3
Operation Setting at Overrun	1:Immediate Stop	1:Immediate Stop
Start Permission at Homing Uncompleted	0:Disabled	0:Disabled
Deceleration at Stop	0.0 um/s^2	0.0 um/s^2
Stop Selection at Deceleration to Stop	1:Recreate Deceleration Curve	1:Recreate Deceleration Curve
Stop Selection at Stop Cause Occurrence	3:Alternative Acceleration/Deceleration	3:Alternative Acceleration/Deceleration
Stop Selection at H/W Stroke Limit Error Occurrence	1:Immediate Stop	1:Immediate Stop
Process Selection at Servo OFF Command During Operation	0:Ignore	0:Ignore
Stop Selection at S/W Stroke Limit Error Occurrence	1:Immediate Stop	1:Immediate Stop
Driver Command Discard Detection Setting	1:Detection Enabled	1:Detection Enabled
Stop Signal		
Signal		
Target	[VAR]G_bStopSignalX	[VAR]G_bStopSignalY
Signal Detection Method	0:Detection at TRUE	0:Detection at TRUE
Compensation Time	0.0 s	0.0 s
Filter Time	0.0 s	0.0 s
Software Stroke Limit Lower Value	-10000000000.0 um	-10000000000.0 um
Software Stroke Limit Target	-1:Invalid	-1:Invalid
Software Stroke Limit Upper Value	10000000000.0 um	10000000000.0 um
Position Command Unit	um	um
Position Command Unit String		
Velocity Command Unit	U/s	U/s
Negative Direction Velocity Limit Value	2500000000.0 um/s	2500000000.0 um/s
Operation Setting at Velocity Limit Value Exceeded	0:Ignore	0:Ignore
Positive Direction Velocity Limit Value	2500000000.0 um/s	2500000000.0 um/s

■Virtual drive axis

Item	VirtualAxis0001
Axis Information	
Axis No.	301
Axis Parameter Constant	**Expands setting values at axis variable initialization.Re-importing to the**
Axis Type Setting	3:Virtual Drive Axis
Upper Limit Signal	
Signal	
Target	
Signal Detection Method	0:Detection at TRUE
Compensation Time	0.0 s
Filter Time	0.0 s
Lower Limit Signal	
Signal	
Target	
Signal Detection Method	0:Detection at TRUE
Compensation Time	0.0 s
Filter Time	0.0 s
Control Cycle Setting	0:Operate in the First Operation Cycle
Absolute Position Control Setting	0:Disable Absolute Position System
Ring Counter Enabled Selection	0:Disabled
Ring Counter Lower Limit Value	-10000000000.0
Ring Counter Upper Limit Value	10000000000.0
Fast Operation Mode Setting	0000
Axis Parameter	**Expands initial values at axis variable initialization.Re-importing to the c**
Acceleration Limit Value	2147483647.0 um/s^2
Operation Selection at Start Acceleration/Deceleration 0	-1:Error (Not Started)
Command In-position Width	100.0 um
Deceleration Limit Value	2147483647.0 um/s^2
Forced Stop Signal	
Signal	
Target	
Signal Detection Method	0:Detection at TRUE
Compensation Time	0.0 s
Filter Time	0.0 s
Homing Required or Not	1:Homing Required
Jerk Limit Value	2147483647.0 um/s^3
Operation Setting at Overrun	1:Immediate Stop
Start Permission at Homing Uncompleted	0:Disabled
Deceleration at Stop	0.0 um/s^2
Stop Selection at Deceleration to Stop	1:Recreate Deceleration Curve
Stop Selection at Stop Cause Occurrence	3:Alternative Acceleration/Deceleration
Stop Selection at H/W Stroke Limit Error Occurrence	1:Immediate Stop
Process Selection at Servo OFF Command During Operation	0:Ignore
Stop Selection at S/W Stroke Limit Error Occurrence	1:Immediate Stop
Stop Signal	
Signal	
Target	
Signal Detection Method	0:Detection at TRUE
Compensation Time	0.0 s
Filter Time	0.0 s
Software Stroke Limit Lower Value	-10000000000.0 um
Software Stroke Limit Target	-1:Invalid
Software Stroke Limit Upper Value	10000000000.0 um
Position Command Unit	um
Position Command Unit String	
Velocity Command Unit	U/s
Negative Direction Velocity Limit Value	2500000000.0 um/s
Operation Setting at Velocity Limit Value Exceeded	0:Ignore
Positive Direction Velocity Limit Value	2500000000.0 um/s

■Virtual linked axis

Item	LinkAxis0001	LinkAxis0002
Axis Information		
Axis No.	401	402
Axis Parameter Constant	**Expands setting values at axis variable initialization.Re-importing to the control data is not executed after axis variabl**	
Axis Type Setting	5:Virtual Linked Axis	5:Virtual Linked Axis
Control Cycle Setting	0:Operate in the First Operation Cycle	0:Operate in the First Operation Cycle
Absolute Position Control Setting	0:Disable Absolute Position System	0:Disable Absolute Position System
Ring Counter Enabled Selection	0:Disabled	0:Disabled
Ring Counter Lower Limit Value	-10000000000.0	-10000000000.0
Ring Counter Upper Limit Value	10000000000.0	10000000000.0
Fast Operation Mode Setting	0000	0000
Axis Parameter	**Expands initial values at axis variable initialization.Re-importing to the control data is also executed after axis variable**	
Homing Required or Not	0:Homing Not Required	0:Homing Not Required
Start Permission at Homing Uncompleted	0:Disabled	0:Disabled
Position Command Unit	um	um
Position Command Unit String		
Velocity Command Unit	U/s	U/s

Axes group setting

Item	AxesGroup001
Axes Group Information	
Axes Group No.	1
Axes Group Parameter	**Expands initial values at axes group variable generation.Re-impor**
Acceleration Limit Value	2147483647.0 um/s^2
Operation Selection at Start Acceleration/Deceleration 0	-1:Error (Not Started)
Configuration Axis[1]	Axis0001
Configuration Axis[2]	Axis0002
Configuration Axis[3]	
Configuration Axis[4]	
Configuration Axis[5]	
Configuration Axis[6]	
Configuration Axis[7]	
Configuration Axis[8]	
Configuration Axis[9]	
Configuration Axis[10]	
Configuration Axis[11]	
Configuration Axis[12]	
Configuration Axis[13]	
Configuration Axis[14]	
Configuration Axis[15]	
Configuration Axis[16]	
Command In-position Width	100.0 um
Deceleration Limit Value	2147483647.0 um/s^2
Jerk Limit Value	2147483647.0 um/s^3
Operation Setting at Overrun	1:Immediate Stop
Deceleration at Stop	0.0 um/s^2
Stop Selection at Deceleration to Stop	1:Recreate Deceleration Curve
Configuration Axes Operation Selection at Axis Stop Cause Occurrence	1:Immediate Stop
Stop Selection at Stop Cause Occurrence	3:Alternative Acceleration/Deceleration
Position Command Unit	um
Position Command Unit String	
Velocity Command Unit	U/s
Velocity Limit Value	2500000000.0 um/s

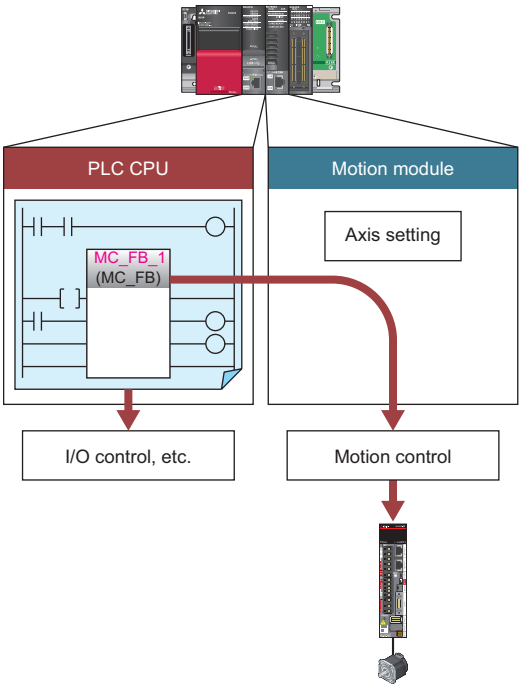
4 PROGRAMMING BY PLC CPU ONLY

Programming can be done on a PLC CPU by using the FBD/LD, ladder, or ST languages.
The example in this chapter uses the ladder language.
To obtain the sample program used in this chapter, contact your local sales office.

- Sample program

File name
R04-78G4-Ladder sample_*****.gx3

*1 ***** = indicates the sample program version



Point

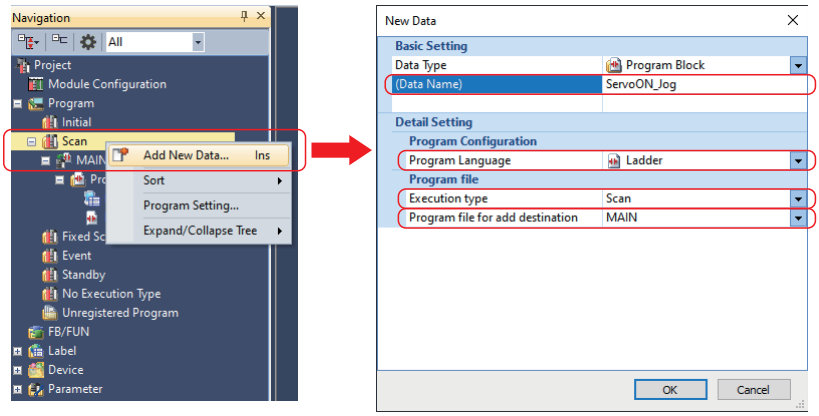
When applying the sample program to an actual system, ensure the applicability and confirm that it will not cause system control problems.
Consider and add interlock conditions according to the user's system.

4.1 Creation Procedure for PLC CPU Programs

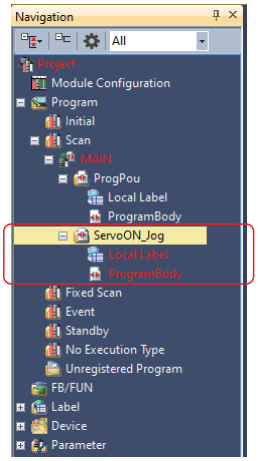
MELSOFT GX Works3 is used to create a program for a PLC CPU.

Creating a program block

1. In the navigation window of MELSOFT GX Works3, right-click "Program"⇨Program execution type ("Scan" in this example), then select "Add New Data".
2. The "New Data" screen is displayed. Set the data name, the program language, and the program file for add destination. Then click the [OK] button. Only alphanumeric characters can be used for the data name.



3. A program block (setting example: ServoON_Jog) is added to the navigation window.



Program execution type

The following five execution types are available.

Execution type	Description
Initial	This type of program is executed only once when the CPU module has been powered OFF→ON or switched from the STOP state to the RUN state.
Scan	This type of program is executed only once per scan. The execution starts from the next scan right after the initial execution type program is executed.
Fixed scan	This is an interrupt program which is executed at a specified time interval. Different from a normal interrupt program, this type of program does not require the interrupt pointer (I) or IRET instruction and is executed for each program file unit.
Event	This type of program starts execution when triggered by a specified event.
Standby	This type of program is executed only when its execution is requested.

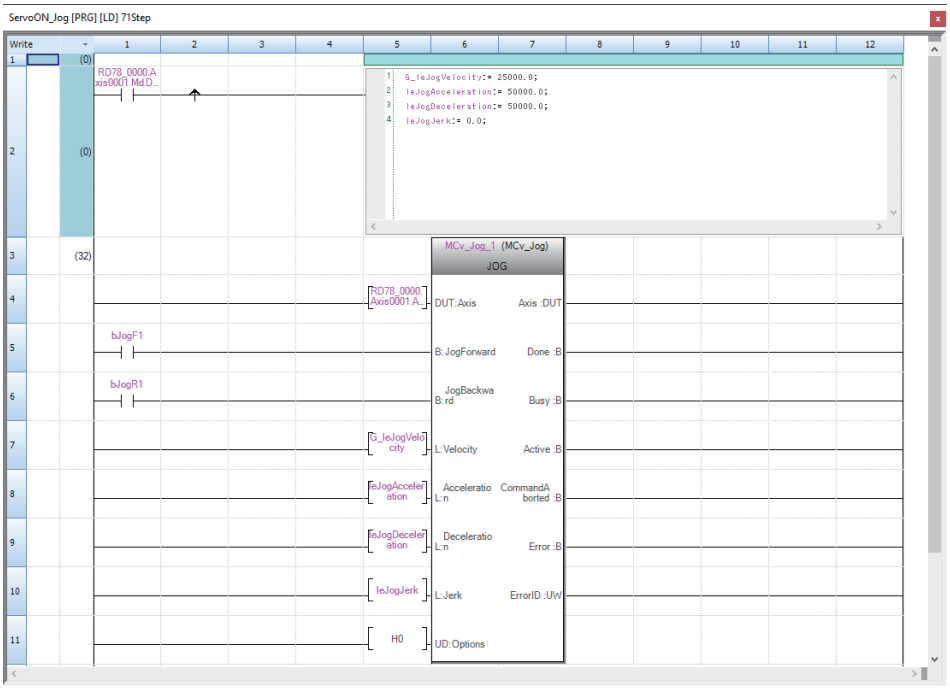
To set the execution type, perform the following operations in the navigation window.

The set execution type will be applied to "Program Setting" in "CPU Parameter".

- Right-click on a target program in the navigation window, then select [Register Program] from the shortcut menu.
- Drag and drop the program onto the target execution type.

Inputting a FB

This section explains the procedure for setting the following JOG operation FB (MCv_Jog).



Point

To create a program using FBs on the PLC CPU, a Motion module FB library must be registered on MELSOFT GX Works3. For details, refer to the following.

☞ Page 25 MELSEC iQ-R series Motion module FB library

Preparing labels

Prepare labels for input/output signals of the FB. This example uses labels for the Velocity, Acceleration, Deceleration, and Jerk inputs of MCv_Jog. When registering a new label, consider whether the label is registered as a global or a local label.

In this example, the labels for JOG command and JOG velocity are registered as global labels, considering that JOG velocity will be operated from an external device such as a GOT (HMI). The labels for JOG acceleration, JOG deceleration, and JOG jerk are registered as local labels of the "ServoON_Jog" program created in Creating a program block (☞ Page 69 Creating a program block) since these labels are only used within this program.

For naming rules for labels, refer to the following.

☞ Page 81 Naming rules for labels

■Global labels

Global [Global Label Setting]

<Filter> | Easy Display | Display Setting | Check

	Label Name	Data Type		Class	Assign (Device/Label)	Initial Value	Constant	English(Display Target)
1	G_IeJogVelocity	FLOAT [Double Precision]	...	VAR_GLOBAL				JOG Velocity
2			...					

■Local labels

ServoON_Jog [PRG] [Local Label Setting]

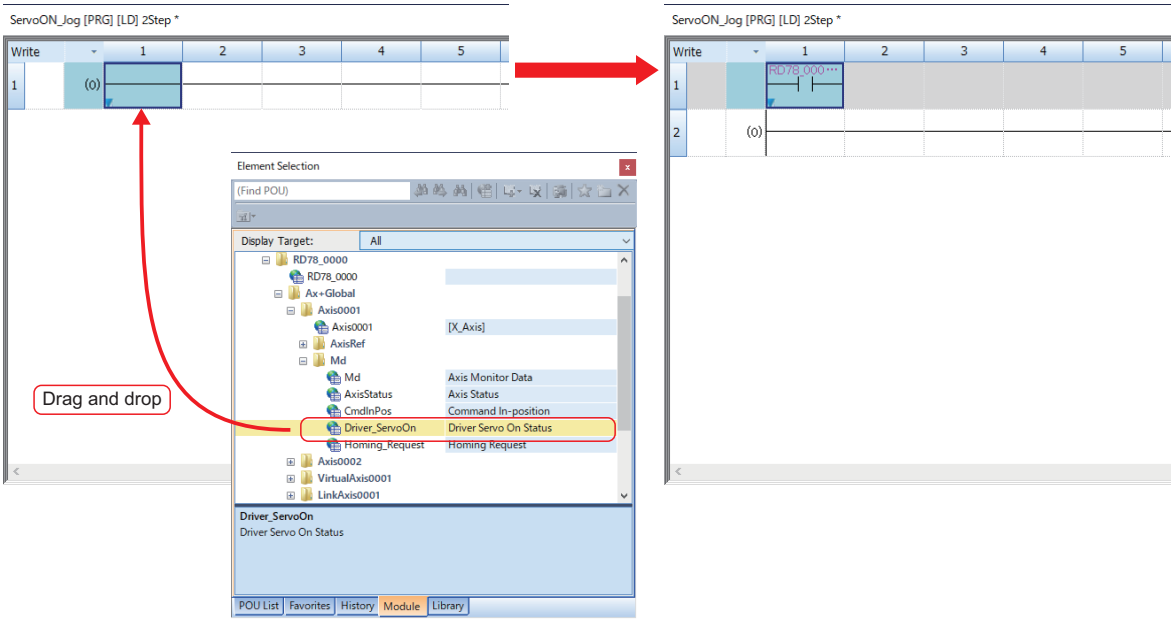
<Filter> | Easy Display | Display Setting | Check

	Label Name	Data Type		Class	Initial Value	Constant	English(Display Target)
1	bJogF1	Bit	...	VAR			JOG Positive rotation Axis0001
2	bJogR1	Bit	...	VAR			JOG Reverse rotation Axis0001
3	IeJogAcceleration	FLOAT [Double Precision]	...	VAR			JOG acceleration
4	IeJogDeceleration	FLOAT [Double Precision]	...	VAR			JOG deceleration
5	IeJogJerk	FLOAT [Double Precision]	...	VAR			JOG jerk
6			...				

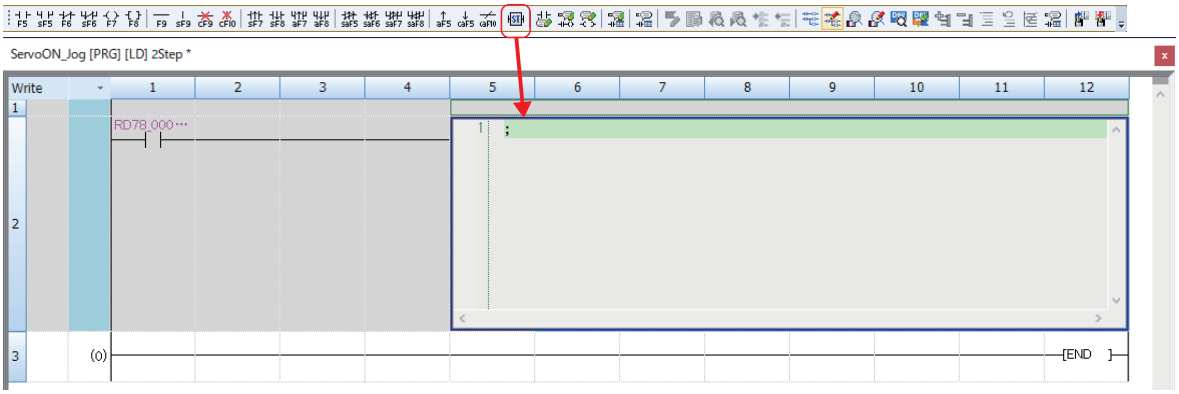
Inputting inline structured text

In this program, the values stored for the JOG velocity, acceleration, deceleration, and jerk are used in inline sructured text. This allows value calculations to be created easily within the ladder program.

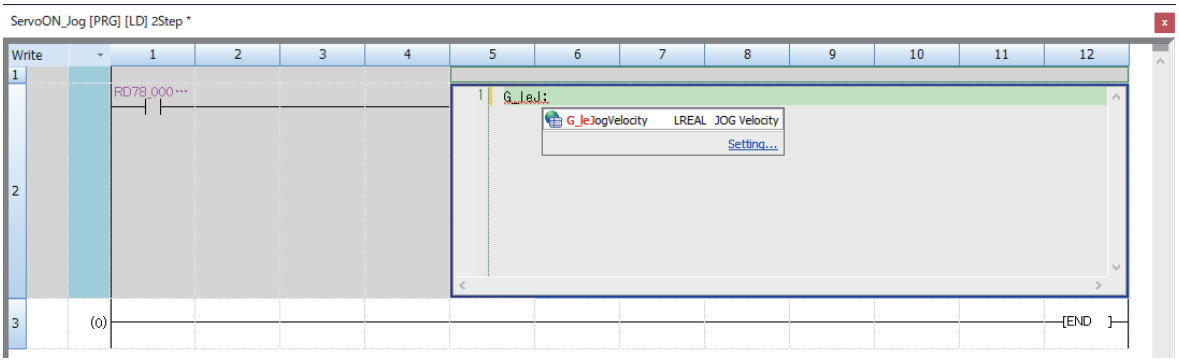
1. Click the "Module" tab in the element selection window. Select "Module Label"⇨"0000:RD78G4"⇨"RD78_0000"⇨"Ax+Global"⇨"Axis0001"⇨"Md" to display the labels for Md. Select "Driver_ServoOn (Driver Servo ON Status)", then drag and drop it into its assigned position on the program editor. "RD78G_0000.Axis0001.MD.Driver_ServoOn" is assigned to the ladder symbol (open contact).



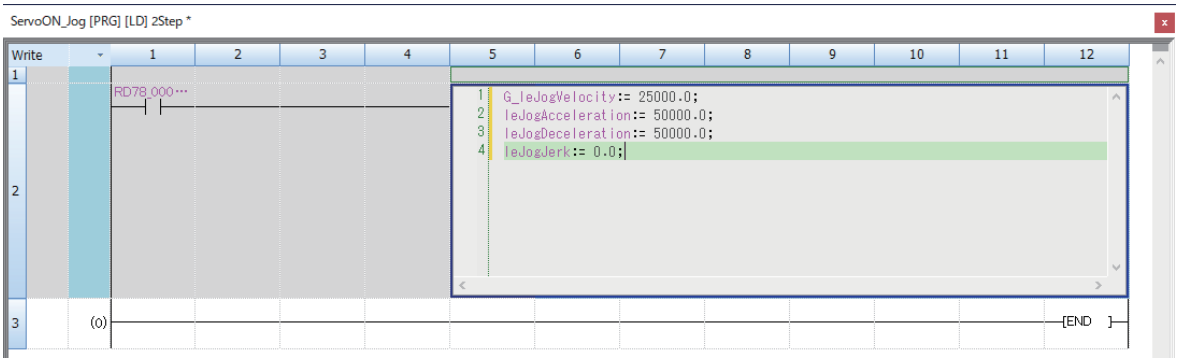
2. Click the insert inline structured text box icon in the toolbar to assign the "Inline Structured Text Box".



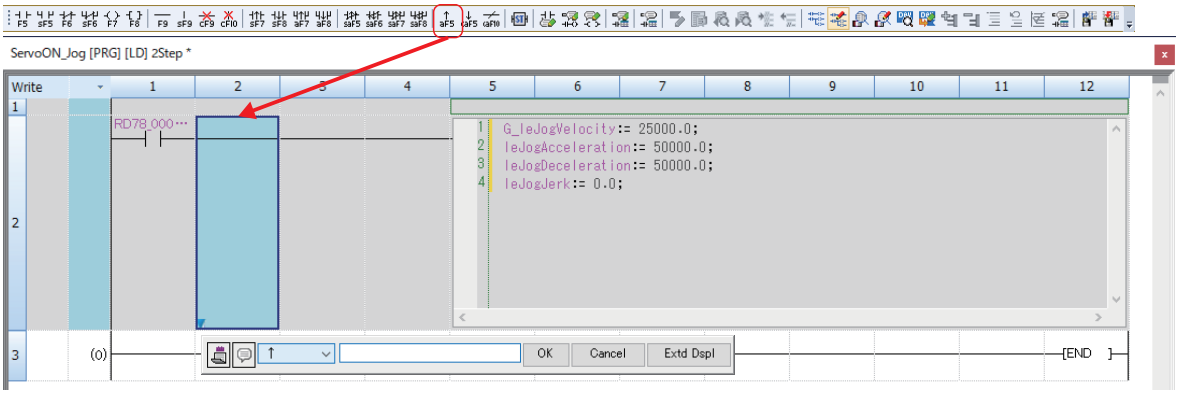
3. Enter "G_lej" to display the registered labels. Select "G_leJogVelocity". After that, enter ":=", then enter the value (25000.0) to store in "G_leJogVelocity". Be sure to include ";" at the end of the statement.



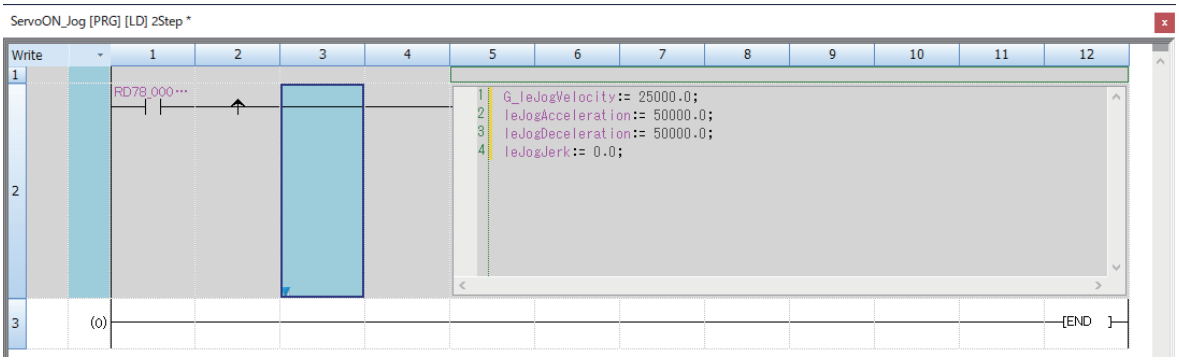
4. In the same manner, enter the "leJogAcceleration", "leJogDeceleration", and "leJogJerk" local labels.



5. On the program editor, click the position to assign the ladder symbol (operation result rising pulse) to, then click the ladder symbol (operation result rising pulse) icon in the toolbar.
When the "Enter Ladder" screen appears, click the [OK] button without entering anything.



6. The ladder symbol (operation result rising pulse) is assigned.

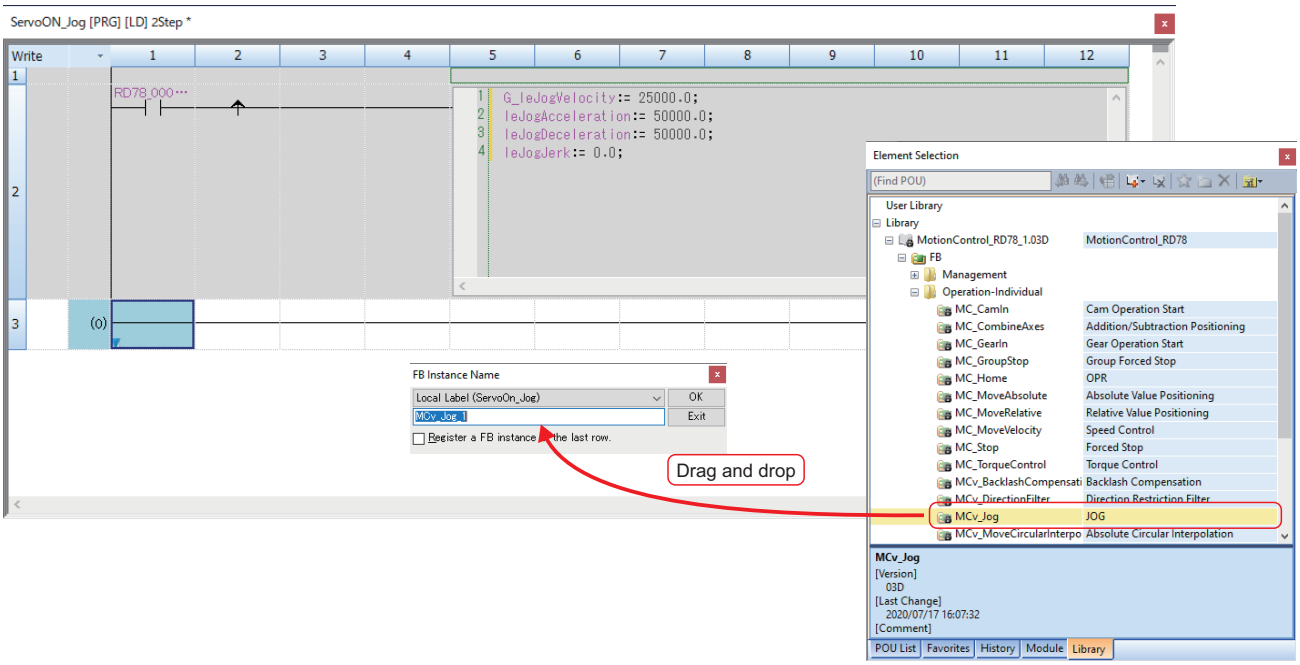


Inputting an FB

1. Click the [Library] tab in the element selection window, then select "Library"⇨"MotionControl_RD78_****[*1]"⇨"FB" to display the Motion module FB library.
The FB for JOG operation (MCv_Jog) is located in the "Operation-Individual" tree.

2. Drag and drop the FB to use (MCv_Jog) onto the program editor to display the "FB Instance Name" screen.
Enter the FB instance name, then click the [OK] button. This example uses the initial settings.

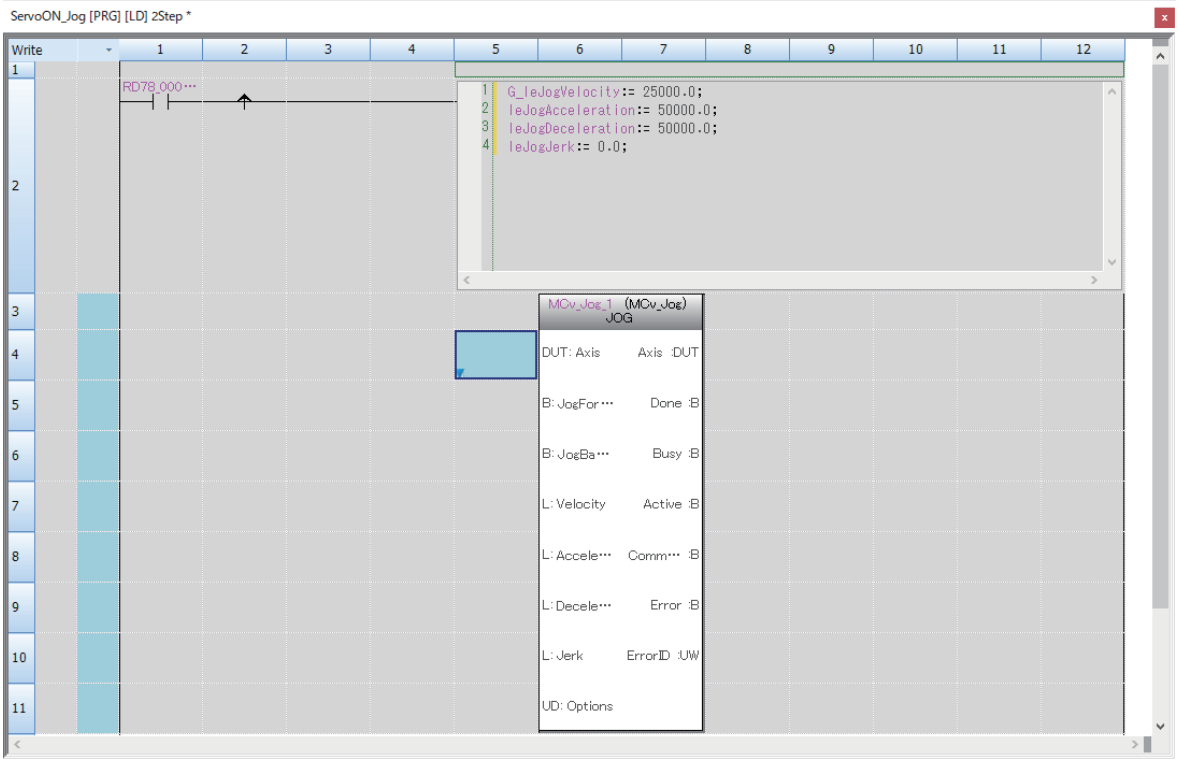
*1 **** = Motion module FB library version



Motion module FBs are categorized into the following three groups.

Group	Description
Management	A Motion control FB that takes an axis or an axes group for the argument and does not change the axis status or the axes group status by execution. (There are some exceptions.)
Operation-Individual	A Motion control FB that takes an axis or an axes group for the argument and changes the axis status or the axes status by execution.
StandardFB	A Motion control FB that does not take an axis or an axes group for the argument.

3. "MCv_Jog" is assigned on the program editor.



Point

When using the ladder language, the labels of assigned FBs are registered as local labels in the program the FBs are assigned to.

Inputting I/O signals

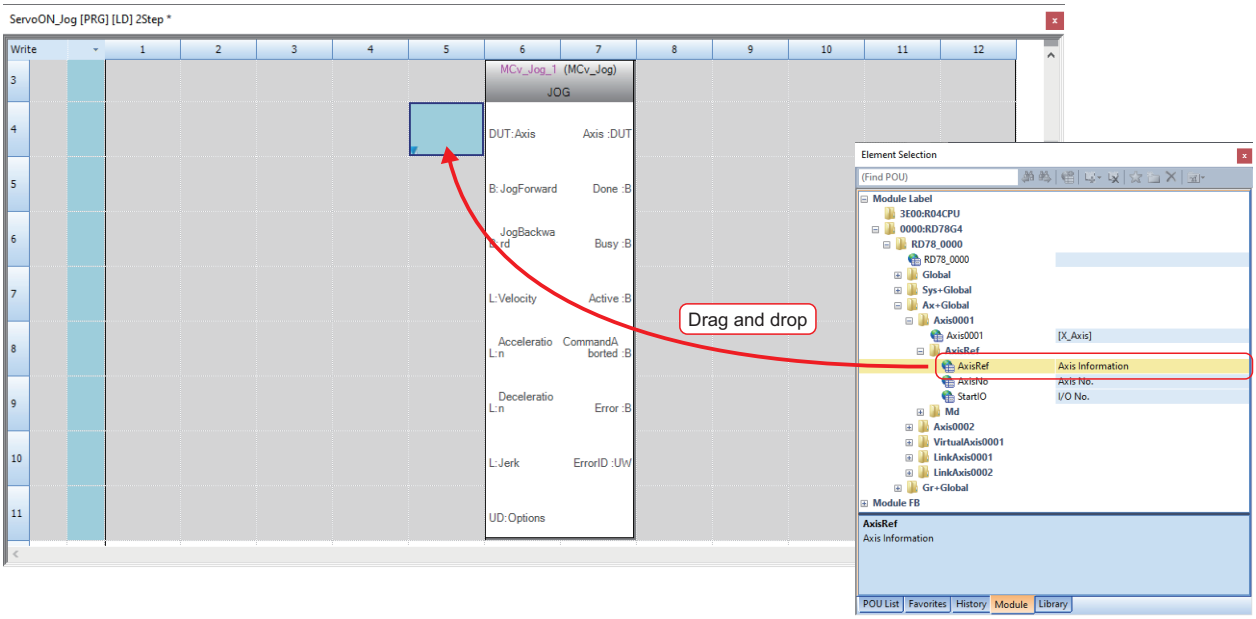
Create a ladder symbol in the FB input circuit.

■Axis input (public label)

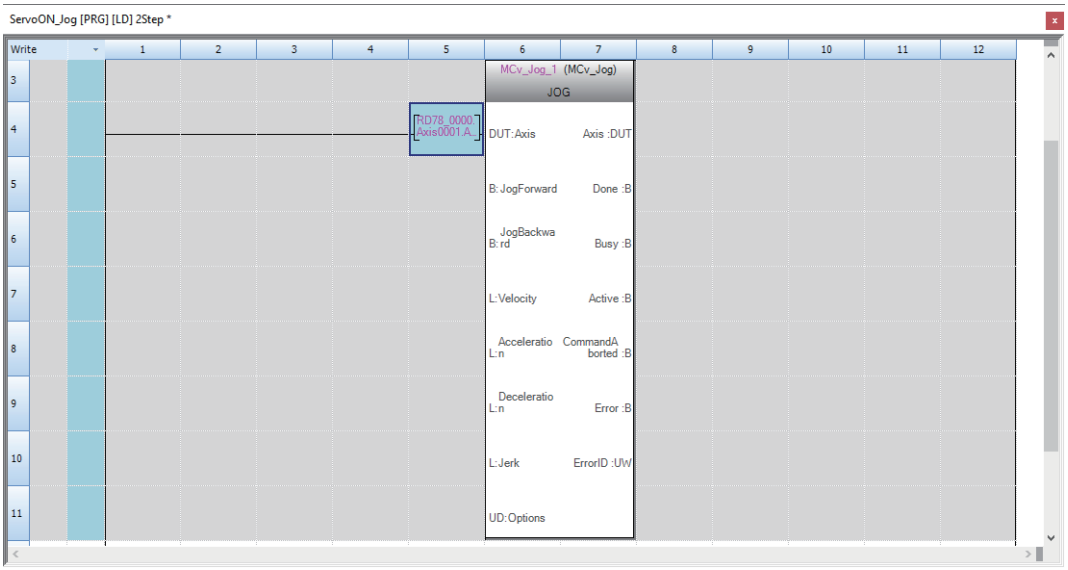
Connect the AxisRef structure registered as a public label to the Axis input by using the label as a circuit symbol (application instruction). For details of public label registration, refer to the following.

☞ Page 59 Public label registration

1. Click the [Module] tab in the element selection window. Select "Module Label"⇨"0000:RD78G4"⇨"RD78_0000"⇨"Ax+Global"⇨"Axis0001"⇨"AxisRef" to display the AxisRef labels. Select "AxisRef (Axis Information)", then drag and drop the label to the left of Axis in the program editor.



2. "RD78G_0000.Axis0001.AxisRef" connects to the Axis input of the FB.



Point

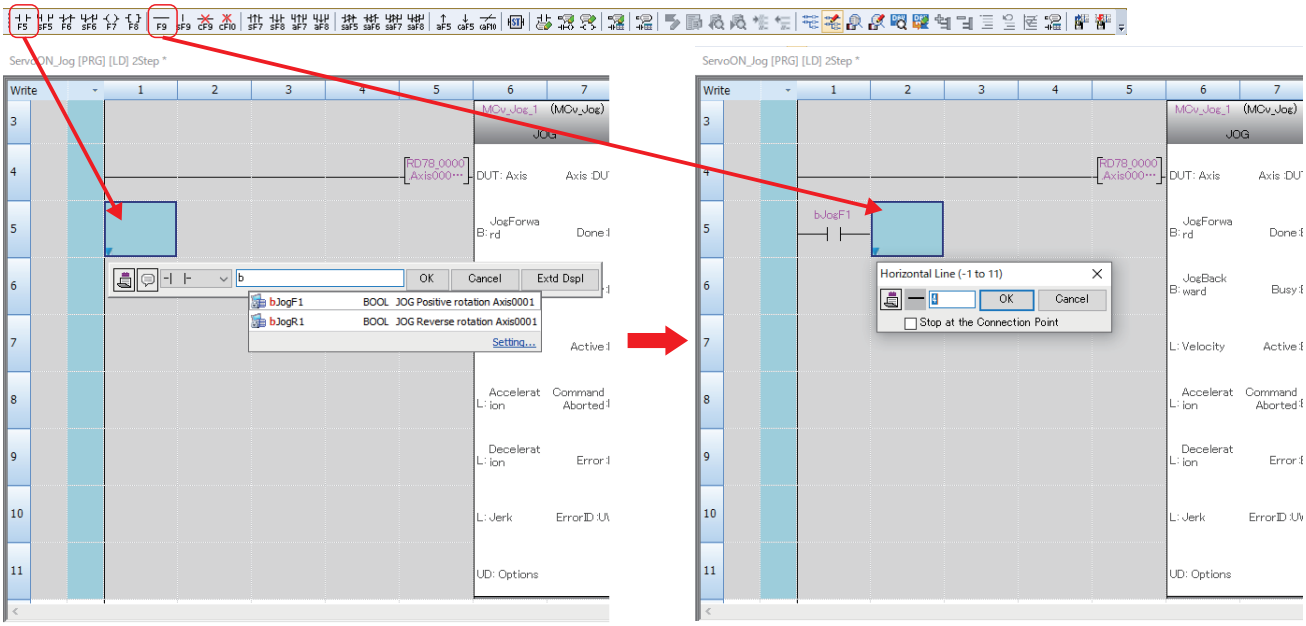
The name of public label "RD78_0000.Axis0001.AxisRef" can be directly entered on the editor.
When you enter "RD78", "RD78_0000" is displayed as a suggestion. When you select "RD78_0000" and enter ".", the structure members, such as "Axis0001", will be displayed.
When you select "Axis0001" and enter ".", the structure members, such as "AxisRef", will be displayed.
Select "AxisRef" to complete the input.

RD78_0000.Axis0001.AxisRef
- Member
- Axis label name
- Module + Start I/O No.

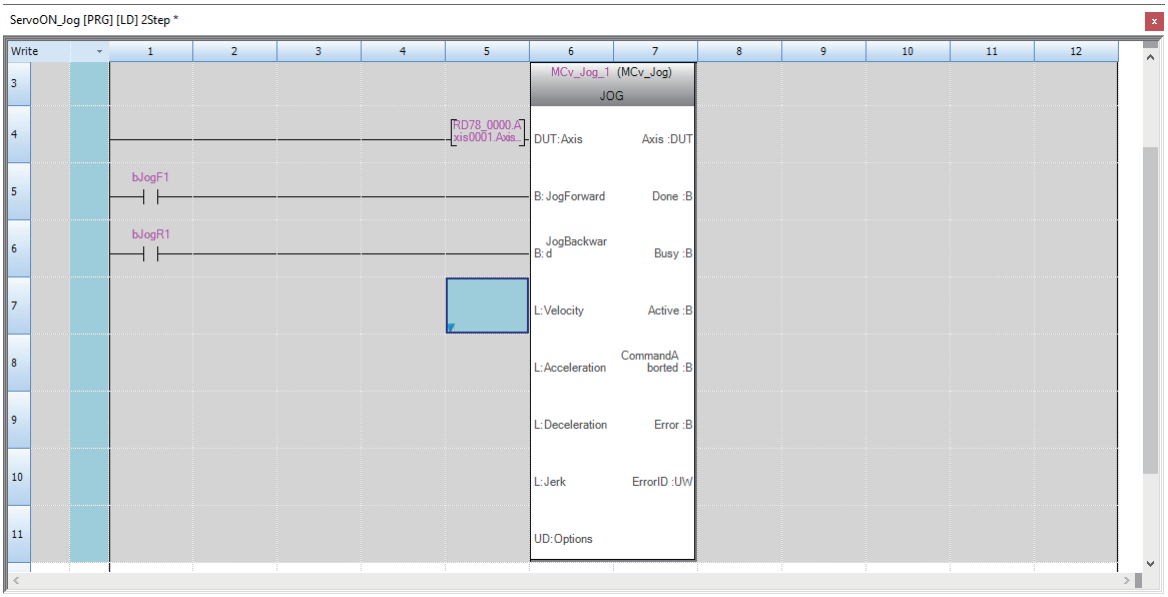
■Ladder symbol (contacts)

Click the position on the program editor on which to assign the ladder symbol, enter the device No. or label name, then connect the ladder symbol.
In the example below, enter the local label (bJogF1) in an open contact ladder symbol, then connect the ladder symbol.

1. Click the position on the program editor on which to assign the ladder symbol (open contact), then click the ladder symbol (open contact) icon in the toolbar to display the "Enter Ladder" screen. Enter "bJogF1", then click the [OK] button. For the label name, the registered global labels and local labels are displayed as a suggestion and can be selected.
2. Assign the ladder symbol (open contact) into which "bJogF1" was entered. Click the horizontal line icon to display the "Horizontal Line" screen. Click the [OK] button to connect the line.



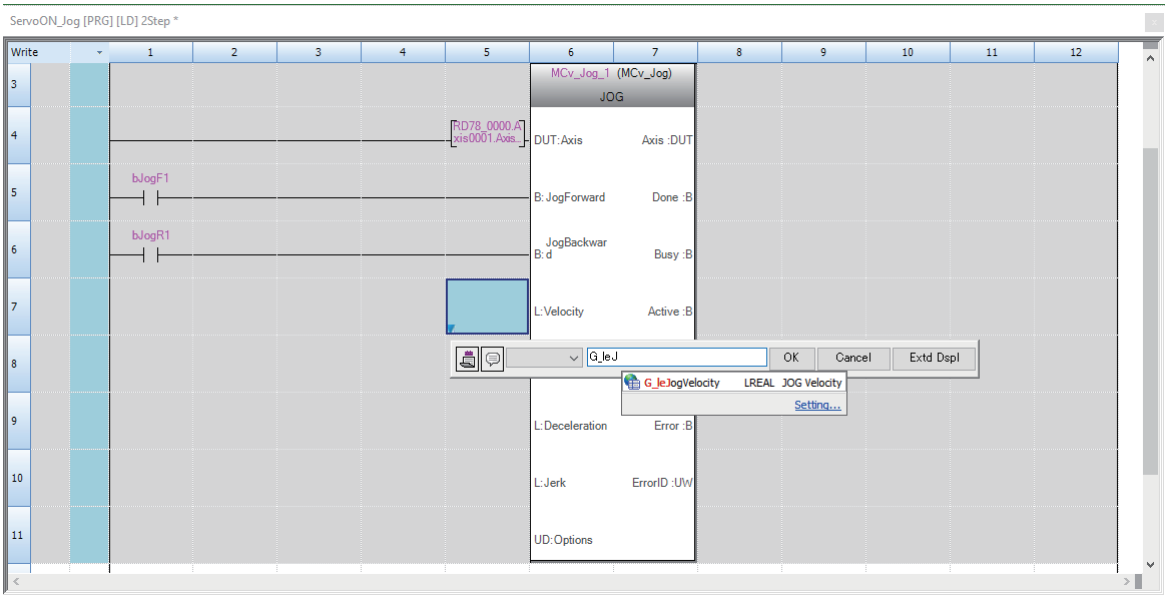
3. In the same manner, connect a ladder symbol (open contact) to the JogBackward input.



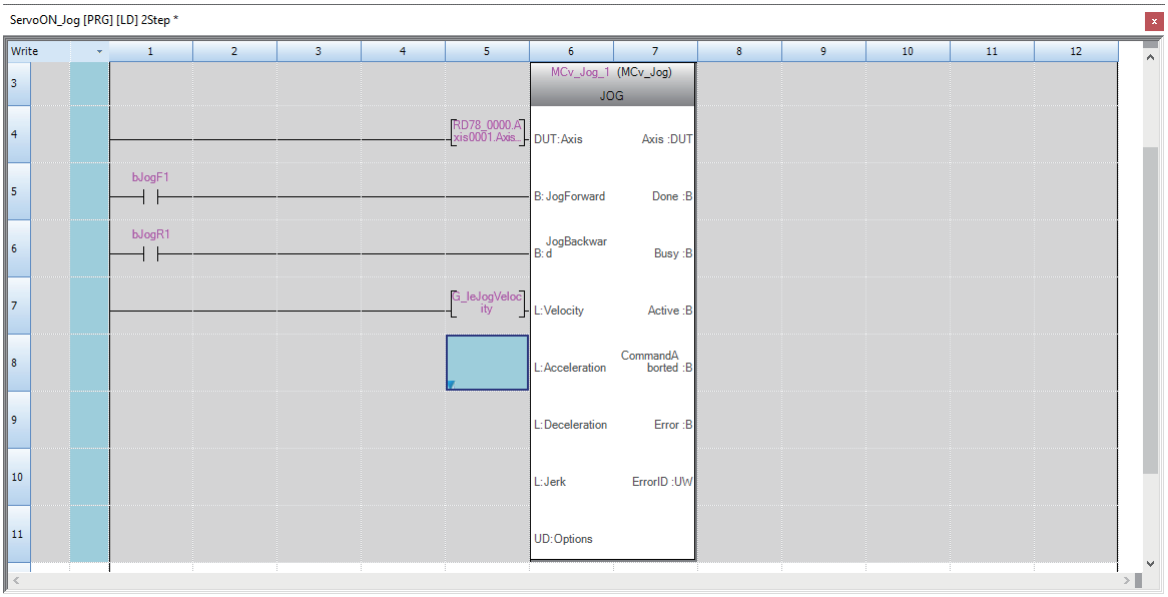
■Ladder symbol (application instruction)

Click the position on the program editor on which to assign the ladder symbol, then directly enter the device No. or label name to assign it there. In the following example, a global label (G_leJogVelocity) is entered.

1. Double-click to the left of Velocity to display the "Enter Ladder" screen. Enter "G_leJogVelocity", then click the [OK] button. For the label name, the registered global labels and local labels are displayed as a suggestion and can be selected.

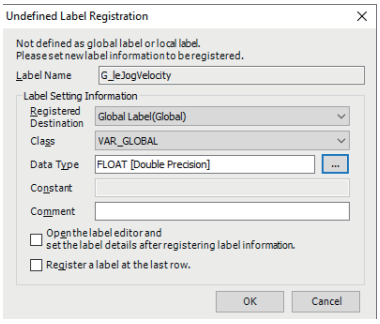


2. "G_leJogVelocity" connects to the Velocity input of the FB.

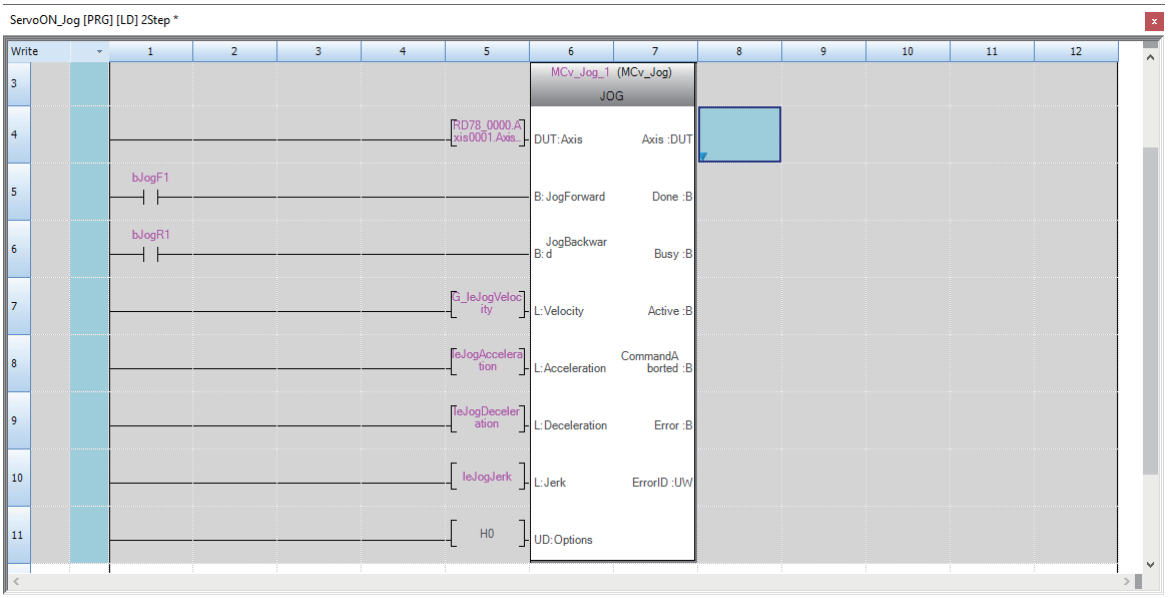


Point

When an undefined label is entered, the "Undefined Label Registration" screen appears.
Enter the data type and the registered destination, then click the [OK] button to register the label in the registered destination.



3. In the same manner, connect local labels "leJogAcceleration", "leJogDeceleration", and "leJogJerk" to the corresponding inputs (Acceleration, Deceleration, and Jerk inputs). For the Options input, connect the constant value (hexadecimal) of "H0".



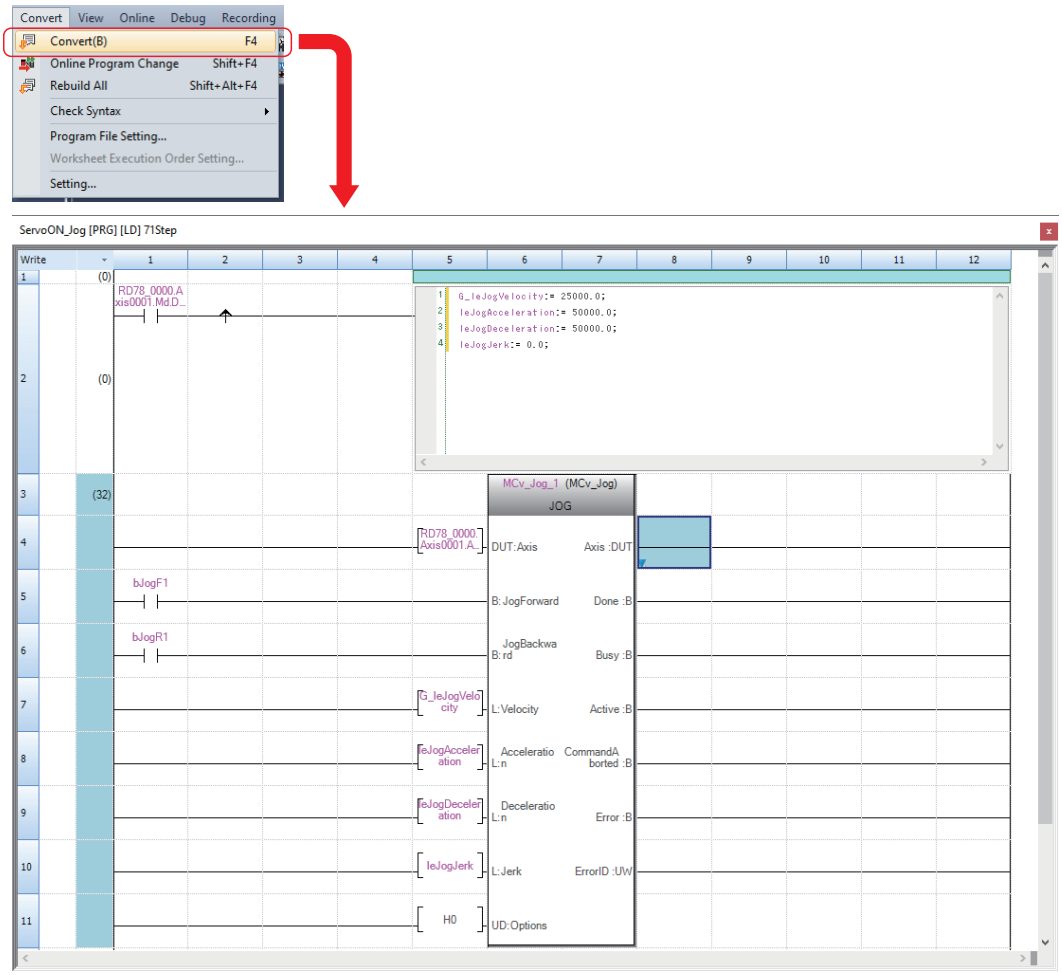
Point

When using an initial value for FB input signals, the setting can be omitted.
The Options input in this example can be omitted since it sets the initial value of "H0".

■Omitting an output signal

I/O signals using an initial value and unused I/O signals can be omitted.
In the following example, the output signals "Done", "Busy", "Active", "CommandAborted", "Error", and "ErrorID" are omitted because they are not used.

1. Select [Convert]⇨[Convert] in the menu to fix the created program. A line is connected on the output variable side.



2. The program is completed.

Point

Performing [Convert]⇨[Convert] in the menu connects a line to all input variables and output variables that are not connected to the FB. A line can be connected in a batch to all areas to be omitted.

4.2 Labels

Naming rules for labels

This document uses the following prefix rules for labels to distinguish their data type.

Data type		Value range	Prefix	
			Local	Global
Bit	BOOL	FALSE, TRUE	b	G_b
Word [unsigned]/bit string [16 bits]	WORD	0 to 65535	u	G_u
Double word [unsigned]/bit string [32 bits]	DWORD	0 to 4294967295	ud	G_ud
Word [signed]	INT	-32468 to 32767	w	G_w
Double word [signed]	DINT	-2147483648 to 2147483647	d	G_d
Single-precision real number	REAL	-2^{128} to -2^{-126}, 0, 2^{-126} to 2^{128}	e	G_e
Double-precision real number	LREAL	-2^{1024} to -2^{-1022}, 0, 2^{-1022} to 2^{1024}	le	G_le
Time	TIME	T#-24d20h31m23s648ms to T#24d20h31m23s647ms	tm	G_tm
Timer	TIMER	TIMER structure • S: contact • C: coil • N: current value	td	G_td

Ex.

For local labels

Data type	Label
Bit	bMoveCMD
Double-precision real number	lePosition
Word [signed]+array	wAxes[16]
Timer	tdTimer1

Ex.

For global labels

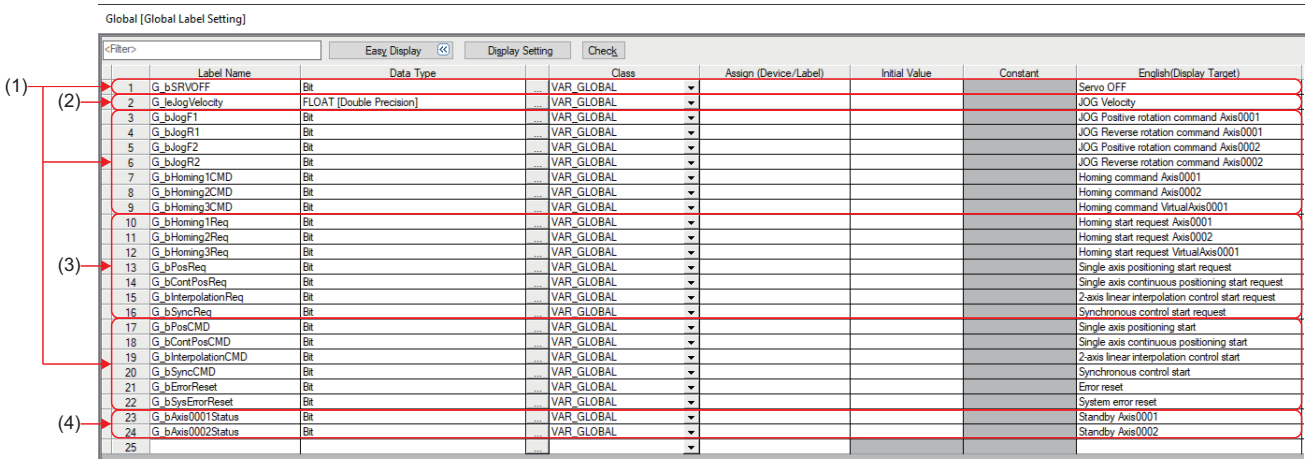
Data type	Label
Bit	G_bMoveCMD
Double-precision real number	G_leVelocity

Global label list/settings for public labels

The following shows setting examples for the global labels in the sample program.
For the local labels, refer to each program.

PLC CPU

The following are the settings for the global labels on the PLC CPU side.



Global [Global Label Setting]

<Filter> | Easy Display | Display Setting | Check

	Label Name	Data Type	Class	Assign (Device/Label)	Initial Value	Constant	English(Display Target)
1	G_bSRVOFF	Bit	VAR_GLOBAL				Servo OFF
2	G_leJogVelocity	FLOAT [Double Precision]	VAR_GLOBAL				JOG Velocity
3	G_bJogF1	Bit	VAR_GLOBAL				JOG Positive rotation command Axis0001
4	G_bJogR1	Bit	VAR_GLOBAL				JOG Reverse rotation command Axis0001
5	G_bJogF2	Bit	VAR_GLOBAL				JOG Positive rotation command Axis0002
6	G_bJogR2	Bit	VAR_GLOBAL				JOG Reverse rotation command Axis0002
7	G_bHoming1CMD	Bit	VAR_GLOBAL				Homing command Axis0001
8	G_bHoming2CMD	Bit	VAR_GLOBAL				Homing command Axis0002
9	G_bHoming3CMD	Bit	VAR_GLOBAL				Homing command VirtualAxis0001
10	G_bHoming1Req	Bit	VAR_GLOBAL				Homing start request Axis0001
11	G_bHoming2Req	Bit	VAR_GLOBAL				Homing start request Axis0002
12	G_bHoming3Req	Bit	VAR_GLOBAL				Homing start request VirtualAxis0001
13	G_bPosReq	Bit	VAR_GLOBAL				Single axis positioning start request
14	G_bContPosReq	Bit	VAR_GLOBAL				Single axis continuous positioning start request
15	G_bInterpolationReq	Bit	VAR_GLOBAL				2-axis linear interpolation control start request
16	G_bSyncReq	Bit	VAR_GLOBAL				Synchronous control start request
17	G_bPosCMD	Bit	VAR_GLOBAL				Single axis positioning start
18	G_bContPosCMD	Bit	VAR_GLOBAL				Single axis continuous positioning start
19	G_bInterpolationCMD	Bit	VAR_GLOBAL				2-axis linear interpolation control start
20	G_bSyncCMD	Bit	VAR_GLOBAL				Synchronous control start
21	G_bErrorReset	Bit	VAR_GLOBAL				Error reset
22	G_bSysErrorReset	Bit	VAR_GLOBAL				System error reset
23	G_bAxis0001Status	Bit	VAR_GLOBAL				Standby Axis0001
24	G_bAxis0002Status	Bit	VAR_GLOBAL				Standby Axis0002
25							

No.	Description
(1)	These labels are used to execute operation of each program.
(2)	This double-precision real number label stores the JOG velocity. By registering this label as a global label, the JOG velocity can be changed from a GOT (HMI) or other programs.
(3)	These labels turn ON while the program is being executed. They are used as an interlock condition to prevent simultaneous execution of multiple programs.
(4)	These labels turn ON when Axis status (AxisStatus) is "4: Standby (Standstill)". They are used to check whether the axis status is standby status or not.

Motion module

The following shows the global and public label settings of the Motion module.
Set these labels in Motion Control Setting Function of MELSOFT GX Works3. When the settings are finished, reflect the public labels.

■Ax+Global

The following labels are registered automatically when each axis is set in Motion Control Setting Function. Set "Public Label" to "Enable".

Ax+Global [Global Label Setting]

	Label Name	Data Type	Class	Initial Value	Constant	English(Display Target)	Remark	Public Label	Motion Control Attribute
1	Axis0001	AXIS_REAL	VAR_GLOBAL	<Detailed Setting>		[X_Axis]		Enable	-
2	Axis0002	AXIS_REAL	VAR_GLOBAL	<Detailed Setting>		[Y_Axis]		Enable	-
3	VirtualAxis0001	AXIS_VIRTUAL	VAR_GLOBAL	<Detailed Setting>		[Vir_Axis01]		Enable	-
4	LinkAxis0001	AXIS_VIRTUAL_LINK	VAR_GLOBAL	<Detailed Setting>		[Lin_Axis01]		Enable	-
5	LinkAxis0002	AXIS_VIRTUAL_LINK	VAR_GLOBAL	<Detailed Setting>		[Lin_Axis02]		Enable	-
6									

■Gr+Global

The following label is registered automatically when the axes group is set in Motion Control Setting Function. Set "Public Label" to "Enable".

Gr+Global [Global Label Setting]

	Label Name	Data Type	Class	Initial Value	Constant	English(Display Target)	Remark	Public Label	Motion Control Attribute
1	AxesGroup001	AXES_GROUP	VAR_GLOBAL	<Detailed Setting>		[X-Y Table]		Enable	-
2									

■Global

The following global labels are set on the Motion module side. Set "Public Label" to "Enable".

Global [Global Label Setting]

	Label Name	Data Type	Class	Initial Value	Constant	English(Display Target)	Remark	Public Label	Motion Control Attribute
1	G_bStopSignalX	Bit	VAR_GLOBAL	0		Stop command Axis0001		Enable	WRITE (=> Motion)
2	G_bStopSignalY	Bit	VAR_GLOBAL	0		Stop command Axis0002		Enable	WRITE (=> Motion)
3									

■Sys+Global

The following structure stores data related to the system. Set "Public Label" to "Enable".

Sys+Global [Global Label Setting]

	Label Name	Data Type	Class	Initial Value	Constant	English(Display Target)	Remark	Public Label	Motion Control Attribute
1	System	SYSTEM	VAR_GLOBAL	<Detailed Setting>				Enable	-
2									

■AXIS_REAL structure

AXIS_REAL [Structure Setting]

	Label Name	Data Type	Class	Initial Value	Constant	English(Display Target)	Public Label	Motion Control Attribute
1	AxisRef	AXIS_REF				Axis Information	Enable	READ (Motion =>): MdConst
2	PrConst	AXIS_REAL_PRM_CONST				Axis Parameter Constant	Disable	WRITE (=> Motion): PrConst
3	Pr	AXIS_REAL_PRM				Axis Parameter	Disable	WRITE (=> Motion): Pr
4	Md	AXIS_REAL_MONI				Axis Monitor Data	Enable	READ (Motion =>): Md
5	Cd	AXIS_REAL_CMD				Axis Control Data	Disable	WRITE (=> Motion): Cd
6								

■AXIS_REAL_MONI structure

AXIS_REAL_MONI [Structure Setting]

	Label Name	Data Type	Class	Initial Value	Constant	English(Display Target)	Public Label	Motion Control Attribute
1	AccelerationLimit	FLOAT [Double Precision]		0.0		Acceleration Limit Value	Disable	-
2	AccelerationOverride	FLOAT [Double Precision]		1.0		Acceleration Override Factor	Disable	-
3	AccelerationZeroBehavior	Word [Signed]		-1		Operation Selection at Start Acceleration/Deceleration 0	Disable	-
4	ActualPosition	FLOAT [Double Precision]		0.0		Actual Position	Disable	-
5	ActualVelocity	FLOAT [Double Precision]		0.0		Actual Velocity	Disable	-
6	Analyzing	Bit		0		Analyzing	Disable	-
7	AutoDeceleration	Bit		0		Automatically Decelerating	Disable	-
8	AxisName	String [Unicode](127)		""		Axis Name	Disable	-
9	AxisStatus	Word [Signed]		0		Axis Status	Enable	-
10	BufferingFBs	Word [Signed]		0		Number of Buffering FBs	Disable	-
11	CmdInPos	Bit		0		Command In-position	Enable	-
12	CmdInPos_Width	FLOAT [Double Precision]		100.0		Command In-position Width	Disable	-
13							Disable	
		Word [Signed]		0				-
27	Driver_ReadyOn	Bit		0		Driver Ready On Status	Disable	-
28	Driver_ServoOn	Bit		0		Driver Servo On Status	Enable	-
29	Driver_State	Word [Signed]		0		Driver Status	Disable	-
30							Disable	
		Bit		0				-
40	Homing_Request	Bit		1		Homing Request	Enable	-
41	Homing_Required	Bit		0		Homing Required or Not	Disable	-
42						Homing Operation Status	Disable	

■AXES_GROUP structure

AXES_GROUP [Structure Setting]

	Label Name	Data Type	Class	Initial Value	Constant	English(Display Target)	Public Label	Motion Control Attribute
1	AxesGroupRef	AXES_GROUP_REF				Axes Group Information	Enable	READ (Motion =>): MdConst
2	Pr	AXES_GROUP_PRM				Axes Group Parameter	Disable	WRITE (=> Motion): Pr
3	Md	AXES_GROUP_MONI				Axes Group Monitor Data	Enable	READ (Motion =>): Md
4	Cd	AXES_GROUP_CMD				Axes Group Control Data	Disable	WRITE (=> Motion): Cd
5								

■AXES_GROUP_MONI structure

AXES_GROUP_MONI [Structure Setting]

	Label Name	Data Type	Class	Initial Value	Constant	English(Display Target)	Public Label	Motion Control Attribute
1	AccelerationLimit	FLOAT [Double Precision]		0.0		Acceleration Limit Value	Disable	-
2	AccelerationOverride	FLOAT [Double Precision]		1.0		Acceleration Override Factor	Disable	-
3	AccelerationZeroBehavior	Word [Signed]		-1		Operation Selection at Start Acceleration/Deceleration 0	Disable	-
4	ActualVelocity	FLOAT [Double Precision]		0.0		Actual Velocity	Disable	-
5	Analyzing	Bit		0		Analyzing	Disable	-
6	AutoDeceleration	Bit		0		Automatically Decelerating	Disable	-
7	Axis	AXIS_REF(1..16)				Configuration Axis	Disable	-
8	BufferingFBs	Word [Signed]		0		Number of Buffering FBs	Disable	-
9	CmdInPos	Bit		0		Command In-position	Enable	-
10	CmdInPos_Width	FLOAT [Double Precision]		100.0		Command In-position Width	Disable	-
11	CommandedAcceleration	FLOAT [Double Precision]		0.0		Commanded Acceleration	Disable	-
12	CommandedDeceleration	FLOAT [Double Precision]		0.0		Commanded Deceleration	Disable	-
13	CommandedJerk	FLOAT [Double Precision]		0.0		Commanded Jerk	Disable	-
14	CommandedVelocity	FLOAT [Double Precision]		0.0		Commanded Velocity	Disable	-
15	DecelerationLimit	FLOAT [Double Precision]		0.0		Deceleration Limit Value	Disable	-
16	Error	Bit		0		Axes Group Error Detection	Disable	-
17	ErrorID	Word [Unsigned]/Bit String [16-bit]		0		Axes Group Error Code	Disable	-
18	GroupName	String [Unicode](127)		""		Axes Group Name	Disable	-
19	GroupStatus	Word [Signed]		0		Axes Group Status	Enable	-
20	InterpolationAxes	Double Word [Unsigned]/Bit String...		0		Interpolation Axes	Disable	-
21	[illegible]					[illegible]	Disable	

■AXIS_VIRTUAL structure

AXIS_VIRTUAL [Structure Setting]

	Label Name	Data Type	Class	Initial Value	Constant	English(Display Target)	Public Label	Motion Control Attribute
1	AxisRef	AXIS_REF				Axis Information	Enable	READ (Motion =>): MdConst
2	PrConst	AXIS_VIRTUAL_PRM_CONST				Axis Parameter Constant	Disable	WRITE (=> Motion): PrConst
3	Pr	AXIS_VIRTUAL_PRM				Axis Parameter	Disable	WRITE (=> Motion): Pr
4	Md	AXIS_VIRTUAL_MONI				Axis Monitor Data	Enable	READ (Motion =>): Md
5	Cd	AXIS_VIRTUAL_CMD				Axis Control Data	Disable	WRITE (=> Motion): Cd
6								

■AXIS_VIRTUAL_MONI structure

AXIS_VIRTUAL_MONI [Structure Setting]

	Label Name	Data Type	Class	Initial Value	Constant	English(Display Target)	Public Label	Motion Control Attribute
1	AccelerationLimit	FLOAT [Double Precision]		0.0		Acceleration Limit Value	Disable	-
2	AccelerationOverride	FLOAT [Double Precision]		1.0		Acceleration Override Factor	Disable	-
3	AccelerationZeroBehavior	Word [Signed]		-1		Operation Selection at Start Acceleration/Deceleration 0	Disable	-
4	Analyzing	Bit		0		Analyzing	Disable	-
5	AutoDeceleration	Bit		0		Automatically Decelerating	Disable	-
6	AxisName	String [Unicode](127)		""		Axis Name	Disable	-
7	AxisStatus	Word [Signed]		0		Axis Status	Enable	-
8	BufferingFBs	Word [Signed]		0		Number of Buffering FBs	Disable	-
9	CmdInPos	Bit		0		Command In-position	Enable	-
10	CmdInPos_Width	FLOAT [Double Precision]		100.0		Command In-position Width	Disable	-
11	[illegible]					Commanded Acceleration	Disable	
[illegible]	[illegible]	SIGNAL_SELECT				[illegible]	[illegible]	
22	Homing_Complete	Bit		0		Homing Completed	Disable	-
23	Homing_Request	Bit		1		Homing Request	Enable	-
24	Homing_Required	Bit		0		Homing Required or Not	Disable	-
25	[illegible]					[illegible]	Disable	

■AXIS_VIRTUAL_LINK structure

AXIS_VIRTUAL_LINK [Structure Setting]

	Label Name	Data Type	Class	Initial Value	Constant	English(Display Target)	Public Label	Motion Control Attribute
1	AxisRef	AXIS_REF				Axis Information	Enable	READ (Motion =>): MdConst
2	PrConst	AXIS_VIRTUAL_LINK_PRM_CONST				Axis Parameter Constant	Disable	WRITE (=> Motion): PrConst
3	Pr	AXIS_VIRTUAL_LINK_PRM				Axis Parameter	Disable	WRITE (=> Motion): Pr
4	Md	AXIS_VIRTUAL_LINK_MONI				Axis Monitor Data	Enable	READ (Motion =>): Md
5	Cd	AXIS_VIRTUAL_LINK_CMD				Axis Control Data	Disable	WRITE (=> Motion): Cd
6								

■SYS_MONI structure

SYS_MONI [Structure Setting]

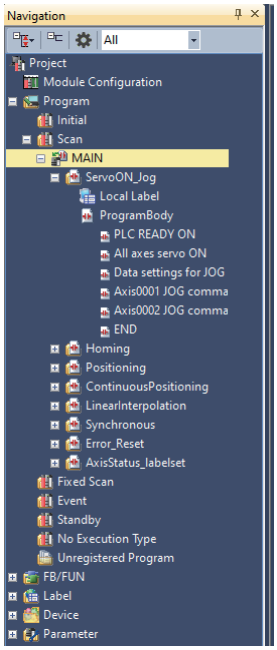
	Label Name	Data Type	Class	Initial Value	Constant	English(Display Target)	Public Label	Motion Control Attribute
1	Addon_AbsSystem	ADDON_MONI				Add-on AbsSystem Monitor	Disable	-
2	Addon_Axis	ADDON_MONI				Add-on Axis Monitor	Disable	-
3	Addon_ExternalSignal	ADDON_MONI				Add-on ExternalSignal Monitor	Disable	-
4	[illegible]					Add-on FileTransfer Monitor	Disable	
[illegible]	[illegible]	CYCLE_MONI				[illegible]	[illegible]	
24	Environment_UserRootPath	String(127)		''			Disable	-
25	Error	Bit		0		Motion Area System Error Detection	Enable	-
26	ErrorHistory	Word [Unsigned]/Bit String [16-bit...		0		Error History Information	Disable	-
27	[illegible]					Latest Error History Data No.	Disable	

4.3 Project Creation

Project name

The following shows the program examples created in this chapter.

Programs are created in the "Program"⇨"Scan"⇨"MAIN" tree.

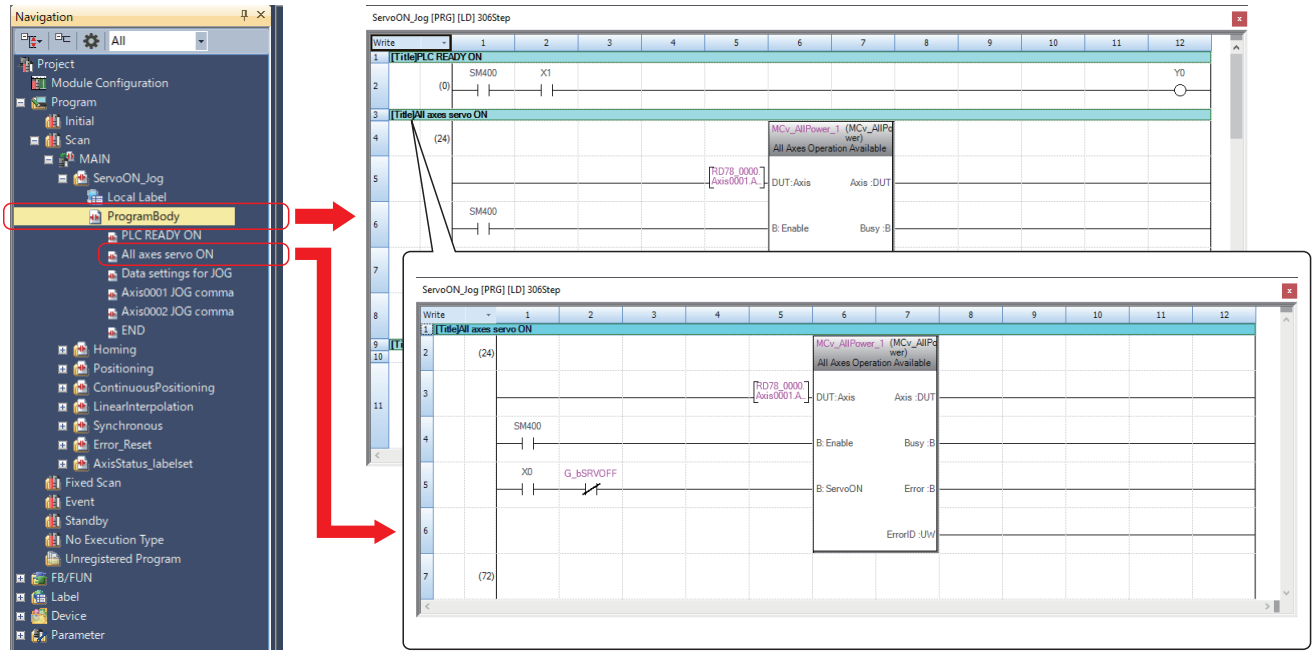
Navigation window display	Program name	Description	Reference
	ServoON_Jog	PLC READY	☞ Page 86 PLC READY (Program Name: ServoON_Jog)
		All axes servo ON	☞ Page 87 Servo ON (Program Name: ServoON_Jog)
		JOG operation	☞ Page 88 JOG Operation (Program Name: ServoON_Jog)
	Homing	Homing	☞ Page 93 Homing (Program Name: Homing)
	Positioning	Single axis positioning	☞ Page 97 Single Axis Positioning Control (Program Name: Positioning)
	ContinuousPositioning	Single axis continuous positioning	☞ Page 100 Single Axis Continuous Positioning (Program Name: ContinuousPositioning)
	LinearInterpolation	Two-axis linear interpolation control	☞ Page 105 Interpolation Control (Program Name: LinearInterpolation)
	Synchronous	Synchronous control	☞ Page 114 Synchronous Control (Program Name: Synchronous)
	Error_Reset	Error reset	☞ Page 129 Error Reset (Program Name: ErrorReset)
	AxisStatus_labelset	Axis status label setting	☞ Page 133 Axis Status Label Setting (Program Name: AxisStatus_labelset)

Showing line statements in the navigation window

Line statements are set in the "Program"⇨"Scan"⇨"MAIN"⇨"Program name"[*1]⇨"ProgramBody" tree in the navigation window.

Double-click the line statement to display only the appropriate circuit block.

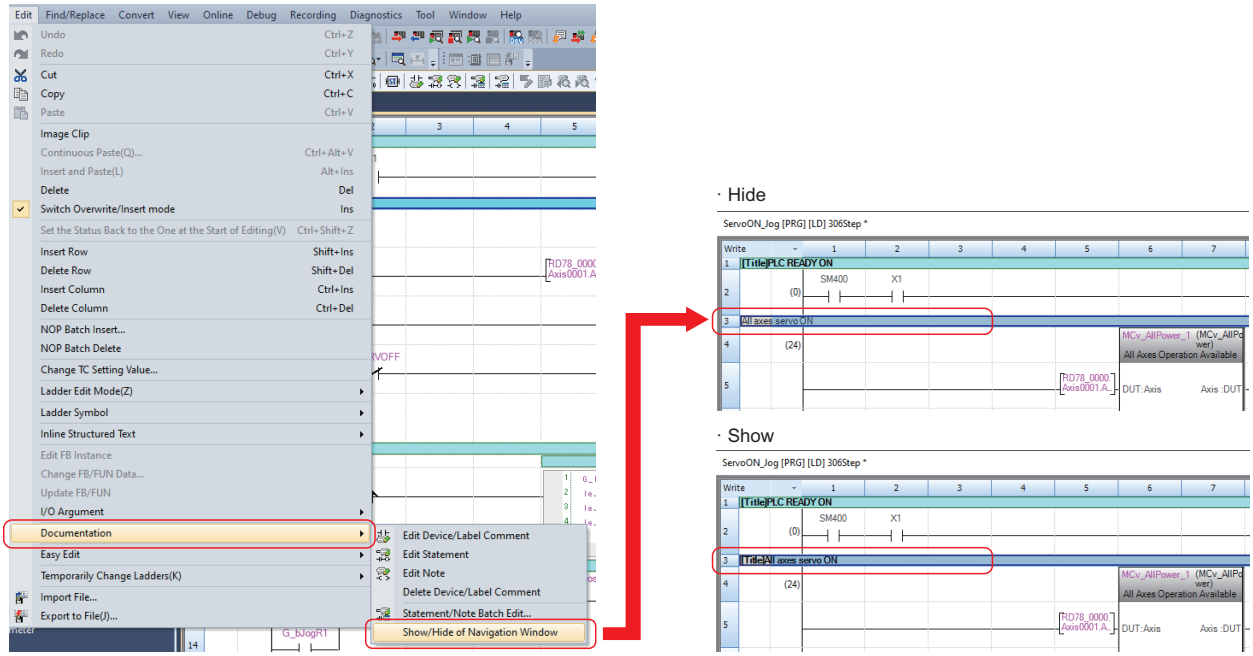
*1 The name of the created program is displayed.



■Showing/hiding line statements

Select the line statement on the program editor, then select [Edit]⇨[Documentation]⇨[Show/Hide of Navigation Window] in the menu to switch between showing and hiding the line statement.

When the line statement is shown in the navigation window, "[Title]" is added before the line statement.



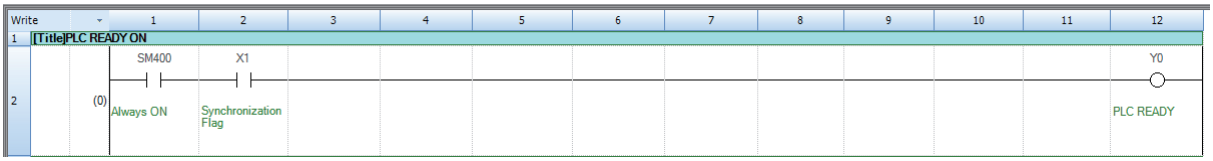
4.4 PLC READY (Program Name: ServoON_Jog)

The RD78G Motion module uses 32 points each for inputs and outputs to send/receive data to/from a PLC CPU. Regardless of the programming language, PLC READY [Y0] needs to be turned ON.

Turn on PLC READY [Y0] with the PLC CPU to start operation of the Motion module.

Program example

PLC READY [Y0] is turned ON after the PLC CPU is powered ON and the synchronization flag [X1] is turned ON.



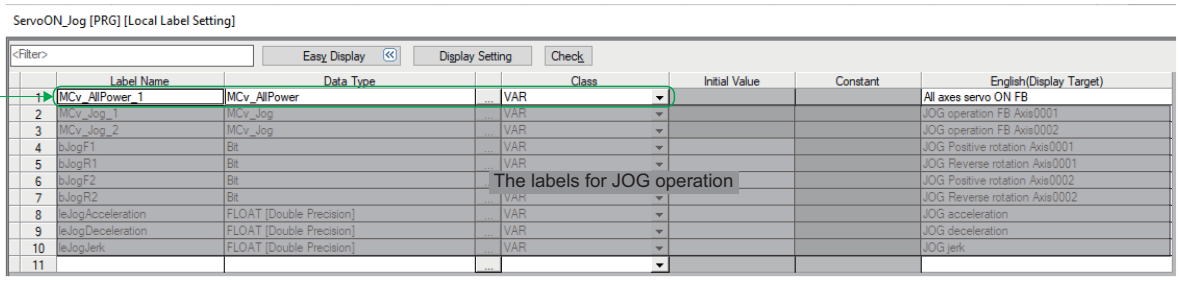
4.5 Servo ON (Program Name: ServoON_Jog)

This program performs servo ON for the real drive axes connected to the servo system.
There are two FBs that are used for servo ON: MCv_AllPower, which performs servo ON for all axes and MC_Power, which performs servo ON for each individual axis.

FBs used

Type	FB	Description
Management	MC_Power	Switches a specified axis to the operation possible status.
	MCv_AllPower	Switches all axes to the operation possible status.

Local labels

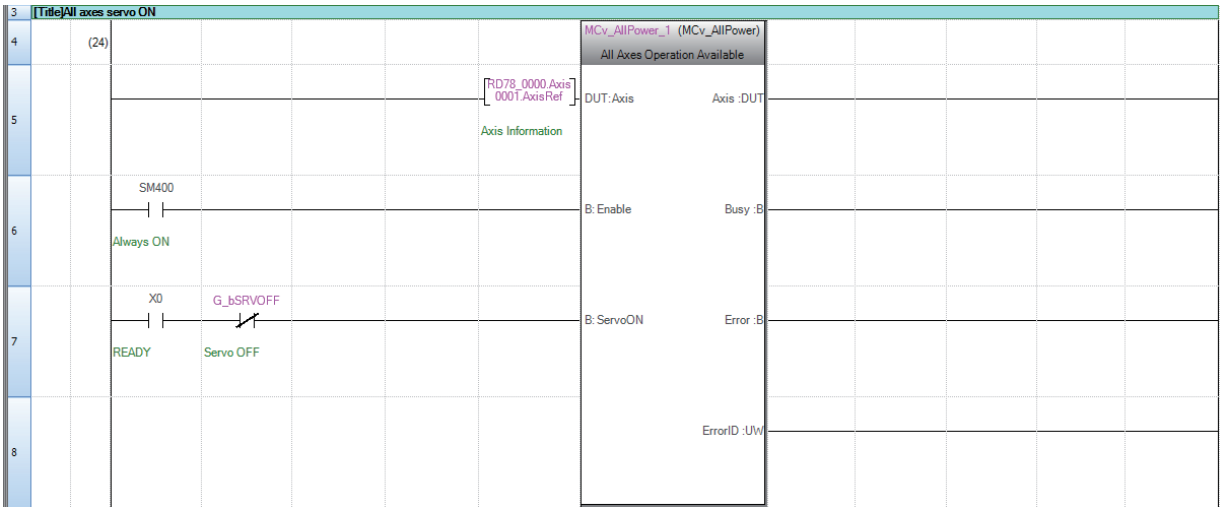


Label Name	Data Type	Class	Initial Value	Constant	English(Display Target)
MCv_AllPower_1	MCv_AllPower	VAR			All axes servo ON FB
MCv_Jog_1	MCv_Jog	VAR			JOG operation FB Axis0001
MCv_Jog_2	MCv_Jog	VAR			JOG operation FB Axis0002
bJogF1	Bit	VAR			JOG Positive rotation Axis0001
bJogR1	Bit	VAR			JOG Reverse rotation Axis0001
bJogF2	Bit	VAR			JOG Positive rotation Axis0002
bJogR2	Bit	VAR			JOG Reverse rotation Axis0002
eJogAcceleration	FLOAT [Double Precision]	VAR			JOG acceleration
eJogDeceleration	FLOAT [Double Precision]	VAR			JOG deceleration
eJogJerk	FLOAT [Double Precision]	VAR			JOG jerk

No.	Description
(1)	This label is automatically added when the user drags and drops the FB (MCv_AllPower) onto the program editor.

Program example

When PLC READY [Y0] is turned ON, Ready [X0] is turned ON. X0 turning ON is used as the condition for executing all axes servo ON. When executing servo OFF, turn ON "G_bSRVOFF".
For MCv_AllPower, connect the chosen AxisRef type structure to the Axis input.



Point

The I/O No. of the Motion module must be specified in the AxisRef type structure being input to Axis of the FB.
When setting the I/O number, use the "I/O No. (StartIO)" member of the AxisRef type structure.
When the AxisRef type structure is made public as a Motion module label the I/O No. will already be set, so the I/O No. can be set by just setting the axis label (setting example: RD78_0000.Axis0001.AxisRef).
When directly setting the "I/O No. (StartIO)" member of the AxisRef type structure, store the first 3 digits of the Motion module start I/O No. when expressed in 4-digit hexadecimal as the "AxisRef.StartIO" value.
(Example) When the I/O No. is "0010"
Store "H1" for AxisRef.StartIO

4.6 JOG Operation (Program Name: ServoON_Jog)

The servo system outputs commands to a target axis and operates the axis in a specified direction while the JOG positive/reverse rotation commands are ON.

FBs used

Type	FB	Description
Operation	MCv_Jog	Executes JOG operation according to the set velocity.

Acceleration/deceleration processing function

The acceleration/deceleration function is used to adjust the acceleration/deceleration curve of the Motion control according to the user's machine.

■Overview

For the acceleration/deceleration method, the following methods are available.

Acceleration/deceleration method	Description
Acceleration/deceleration specification method (Initial value)	Accelerates/decelerates the axis by the acceleration, deceleration, and jerk values specified in the FB.
Acceleration/deceleration time-fixed method	Accelerates/decelerates the axis based on the time specified in the FB regardless of the velocity.

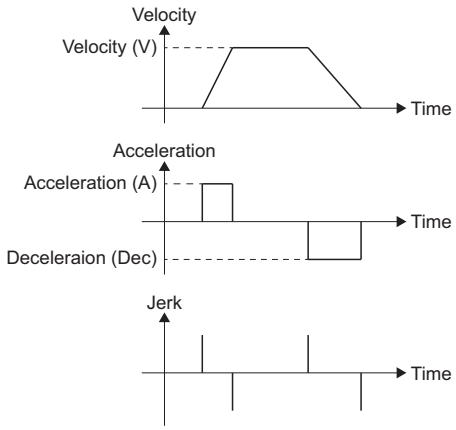
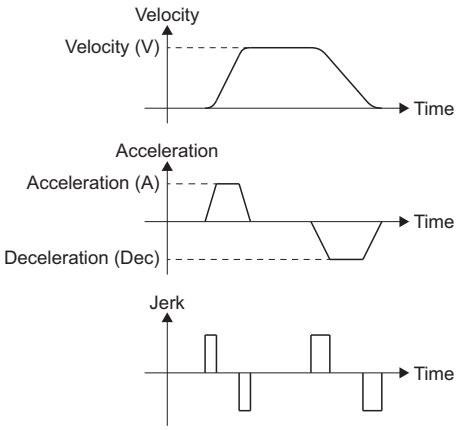
■Setting method

The acceleration/deceleration method can be set with the Options input of operation Motion control FBs, such as MCv_Jog.

Bit	Name	Description
0 to 2	Acceleration/deceleration method setting	0: Acceleration/deceleration specification method (mcAccDec) 1: Acceleration/deceleration time-fixed method (mcFixedTime)
3 to 31	The function varies depending on the FB.	

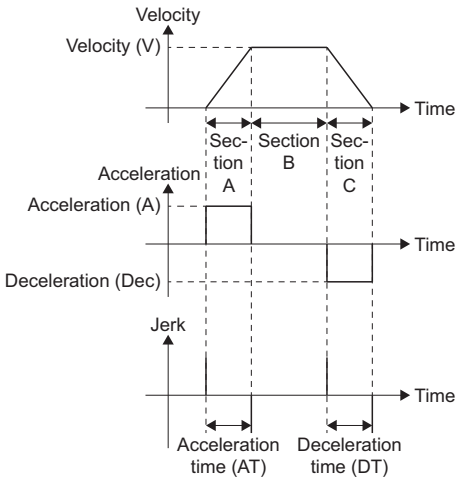
■Acceleration/deceleration specification method

Select "0: Acceleration/deceleration specification method (mcAccDec)" in Acceleration/deceleration method setting (Options: Bit 0 to 2) of the FB, then set the Acceleration, Deceleration, and Jerk inputs.

Trapezoidal acceleration/deceleration	Jerk acceleration/deceleration
When "0.0" is set for the Jerk input of the FB, the operation is referred to as trapezoidal acceleration/deceleration. The velocity creates a trapezoidal shape. 	When a value other than "0.0" is set for the Jerk input of the FB, the operation is referred to as jerk acceleration/deceleration. The velocity creates an S-curve shape. 

■Acceleration/deceleration time-fixed method

Select "1: Acceleration/deceleration time-fixed method (mcFixedTime)" in Acceleration/deceleration method setting (Options: Bit 0 to 2), then set the acceleration time in the Acceleration input of the FB. The Deceleration and Jerk inputs are not used.



■FB input variables

The following table explains each FB input variable.

Input variable	Name	Description
Velocity	Velocity	Specify the velocity used for the FB.
Acceleration	Acceleration	Specify the acceleration used for the FB. • The unit is [U/s^2] when the method is set to "0: Acceleration/deceleration specification method (mcAccDec)". • The unit is [s] when the method is set to "1: Acceleration/deceleration time-fixed method (mcFixedTime)".
Deceleration	Deceleration	Specify the deceleration used for the FB. • The unit is [U/s^2] when the method is set to "0: Acceleration/deceleration specification method (mcAccDec)". • Not used when the method is set to "1: Acceleration/deceleration time-fixed method (mcFixedTime)".
Jerk	Jerk	Specify the jerk used for the FB. • The unit is [U/s^3] when the method is set to "0: Acceleration/deceleration specification method (mcAccDec)". • Not used when the method is set to "1: Acceleration/deceleration time-fixed method (mcFixedTime)".
Options	Options	Specify the acceleration/deceleration method using Acceleration/deceleration method setting (Bit 0 to 2). • 0: Acceleration/deceleration specification method (mcAccDec) • 1: Acceleration/deceleration time-fixed method (mcFixedTime)

Point

When "0: Acceleration/deceleration specification method (McAccDec)" is selected, the values for Acceleration and Deceleration are calculated using the acceleration/deceleration time as follows.

<For velocity V [U/s], acceleration time [s], and deceleration time [s]>

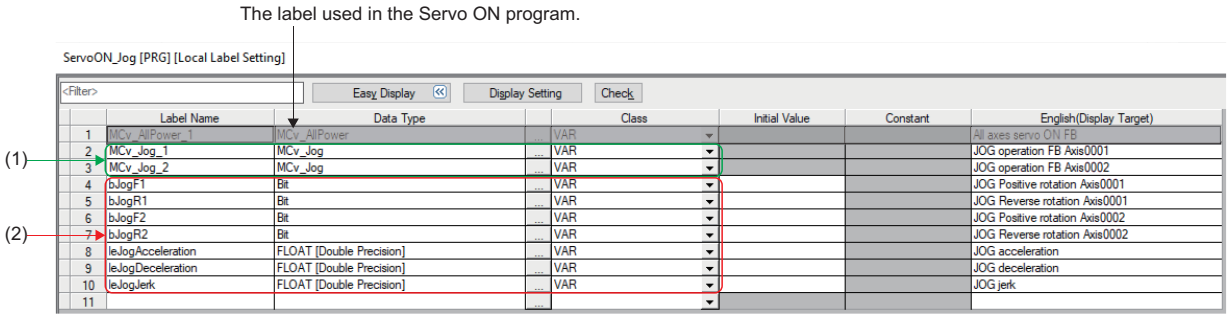
Velocity := (velocity V);

Acceleration := (velocity V/acceleration time);

Deceleration := (velocity V/deceleration time);

Options := (mcAccDec);

Local labels



Label Name	Data Type	Class	Initial Value	Constant	English(Display Target)
MCv_AllPower_1	MCv_AllPower	VAR			All axes servo ON FB
MCv_Jog_1	MCv_Jog	VAR			JOG operation FB Axis0001
MCv_Jog_2	MCv_Jog	VAR			JOG operation FB Axis0002
bJogF1	Bit	VAR			JOG Positive rotation Axis0001
bJogR1	Bit	VAR			JOG Reverse rotation Axis0001
bJogF2	Bit	VAR			JOG Positive rotation Axis0002
bJogR2	Bit	VAR			JOG Reverse rotation Axis0002
leJogAcceleration	FLOAT [Double Precision]	VAR			JOG acceleration
leJogDeceleration	FLOAT [Double Precision]	VAR			JOG deceleration
leJogJerk	FLOAT [Double Precision]	VAR			JOG jerk

No.	Description
(1)	These labels are automatically added when the user drags and drops the FB (MCv_Jog) onto the program editor.
(2)	These labels are registered manually.

Program example

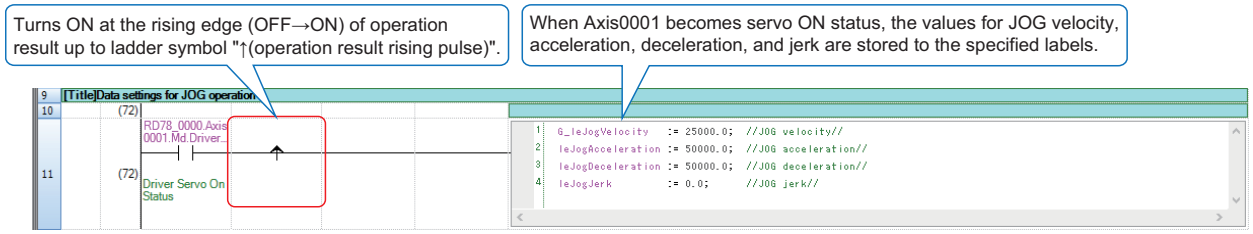
Inline structured text is used to store values for the JOG velocity, acceleration, deceleration, and jerk.

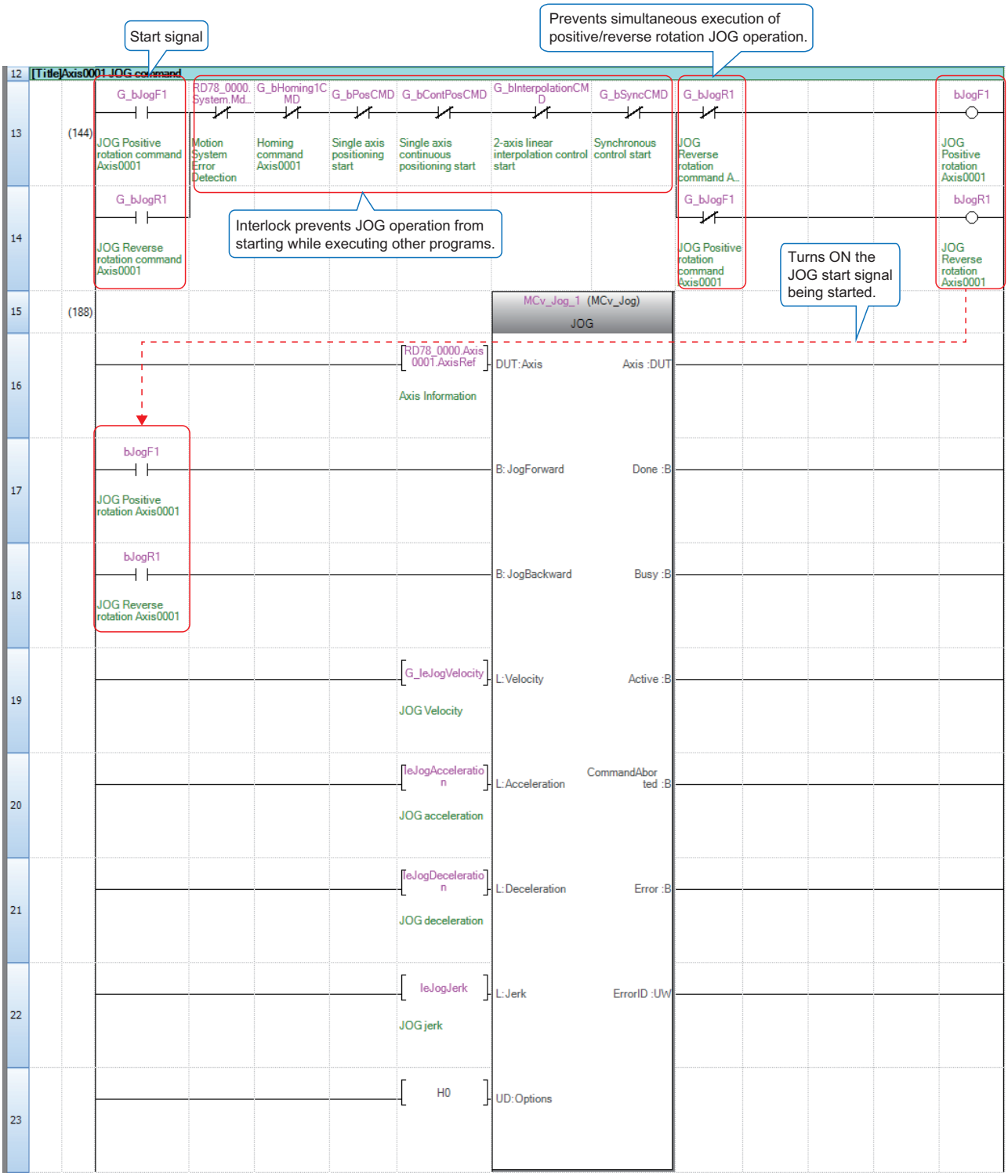
The labels for the JOG positive/reverse rotation commands are connected to the corresponding command inputs (JogForward and JogBackward inputs) of the FB (MCv_Jog).

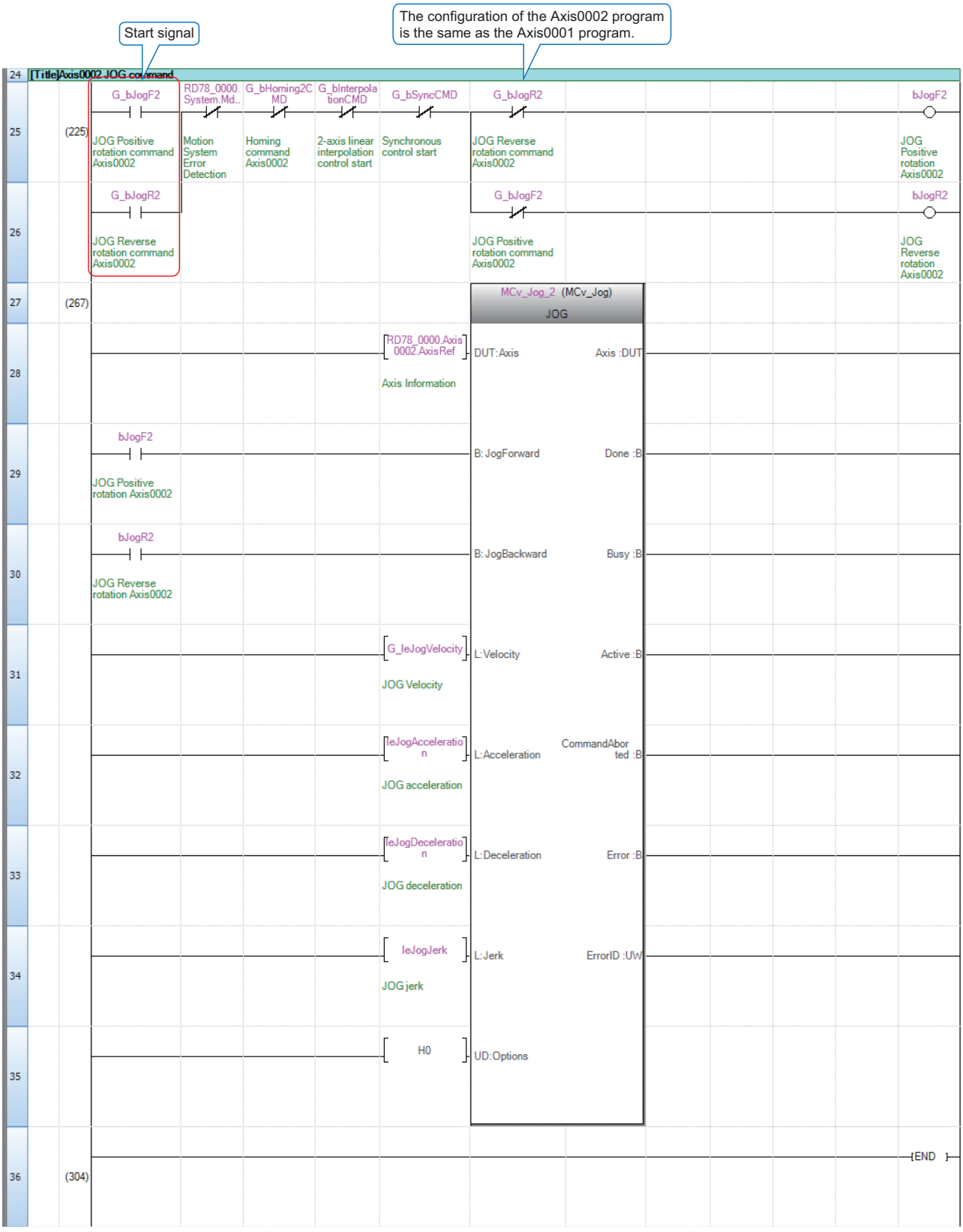
The interlock conditions are added to prevent execution of the JOG operation while other programs are running.

- Start signal

Signal name	Label name	
	Axis0001	Axis0002
JOG positive rotation command	G_bJogF1	G_bJogF2
JOG reverse rotation command	G_bJogR1	G_bJogR2




Start signal
Prevents simultaneous execution of positive/reverse rotation JOG operation.
12 [Title]Axis0001 JOG command
G_bJogF1
RD78_0000.System.Md...
G_bHoming1CMD
G_bPosCMD
G_bContPosCMD
G_bInterpolationCMD
G_bSyncCMD
G_bJogR1
bJogF1
13
(144)
JOG Positive rotation command Axis0001
Motion System Error Detection
Homing command Axis0001
Single axis positioning start
Single axis continuous positioning start
2-axis linear interpolation control start
Synchronous control start
JOG Reverse rotation command A...
JOG Positive rotation Axis0001
G_bJogR1
G_bJogF1
bJogR1
14
JOG Reverse rotation command Axis0001
JOG Positive rotation command Axis0001
JOG Reverse rotation Axis0001
Interlock prevents JOG operation from starting while executing other programs.
Turns ON the JOG start signal being started.
15
(188)
MCv_Jog_1 (MCv_Jog)
JOG
RD78_0000.Axis0001.AxisRef
DUT:Axis
Axis :DUT
16
Axis Information
bJogF1
B:JogForward
Done :B
17
JOG Positive rotation Axis0001
bJogR1
B:JogBackward
Busy :B
18
JOG Reverse rotation Axis0001
G_leJogVelocity
L:Velocity
Active :B
19
JOG Velocity
leJogAcceleration
L:Acceleration
CommandAborted :B
20
JOG acceleration
leJogDeceleration
L:Deceleration
Error :B
21
JOG deceleration
leJogJerk
L:Jerk
ErrorID :UW
22
JOG jerk
H0
UD:Options
23


Start signal
The configuration of the Axis0002 program is the same as the Axis0001 program.
[Title]Axis0002 JOG command
G_bJogF2
JOG Positive rotation command Axis0002
RD78_0000.System.Md..
Motion System Error Detection
G_bHoming2CMD
Homing command Axis0002
G_bInterpolationCMD
2-axis linear interpolation control start
G_bSyncCMD
Synchronous control start
G_bJogR2
JOG Reverse rotation command Axis0002
bJogF2
JOG Positive rotation Axis0002
bJogR2
JOG Reverse rotation Axis0002
MCv_Jog_2 (MCv_Jog)
JOG
RD78_0000.Axis0002.AxisRef
Axis Information
DUT:Axis
Axis :DUT
B:JogForward
Done :B
B:JogBackward
Busy :B
G_leJogVelocity
JOG Velocity
L:Velocity
Active :B
leJogAcceleration
JOG acceleration
L:Acceleration
CommandAborted :B
leJogDeceleration
JOG deceleration
L:Deceleration
Error :B
leJogJerk
JOG jerk
L:Jerk
ErrorID :UW
H0
UD:Options
END

4.7 Homing (Program Name: Homing)

The homing method is set using the parameters of the servo amplifier.

This section explains the data set method homing.

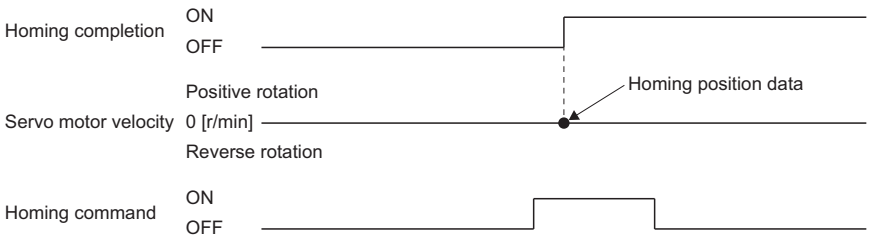
For the dog type homing, refer to the following.

☞ Page 149 Using External Input Signals

Overview

The data set type homing sets the current position as the home position.

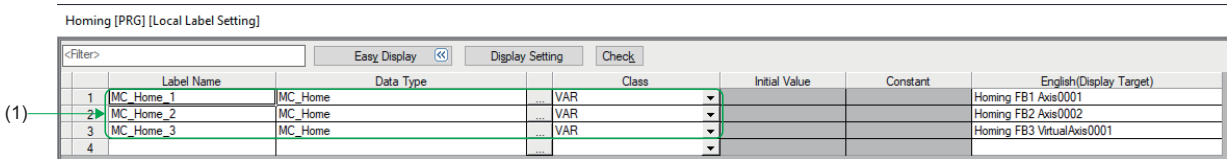
■Time chart of data set type homing (Method37)



FBs used

Type	FB	Description
Operation	MC_Home	Executes homing for the specified axis.

Local labels



No.	Description
(1)	These labels are automatically added when the user drags and drops the FB (MC_Home) onto the program editor.

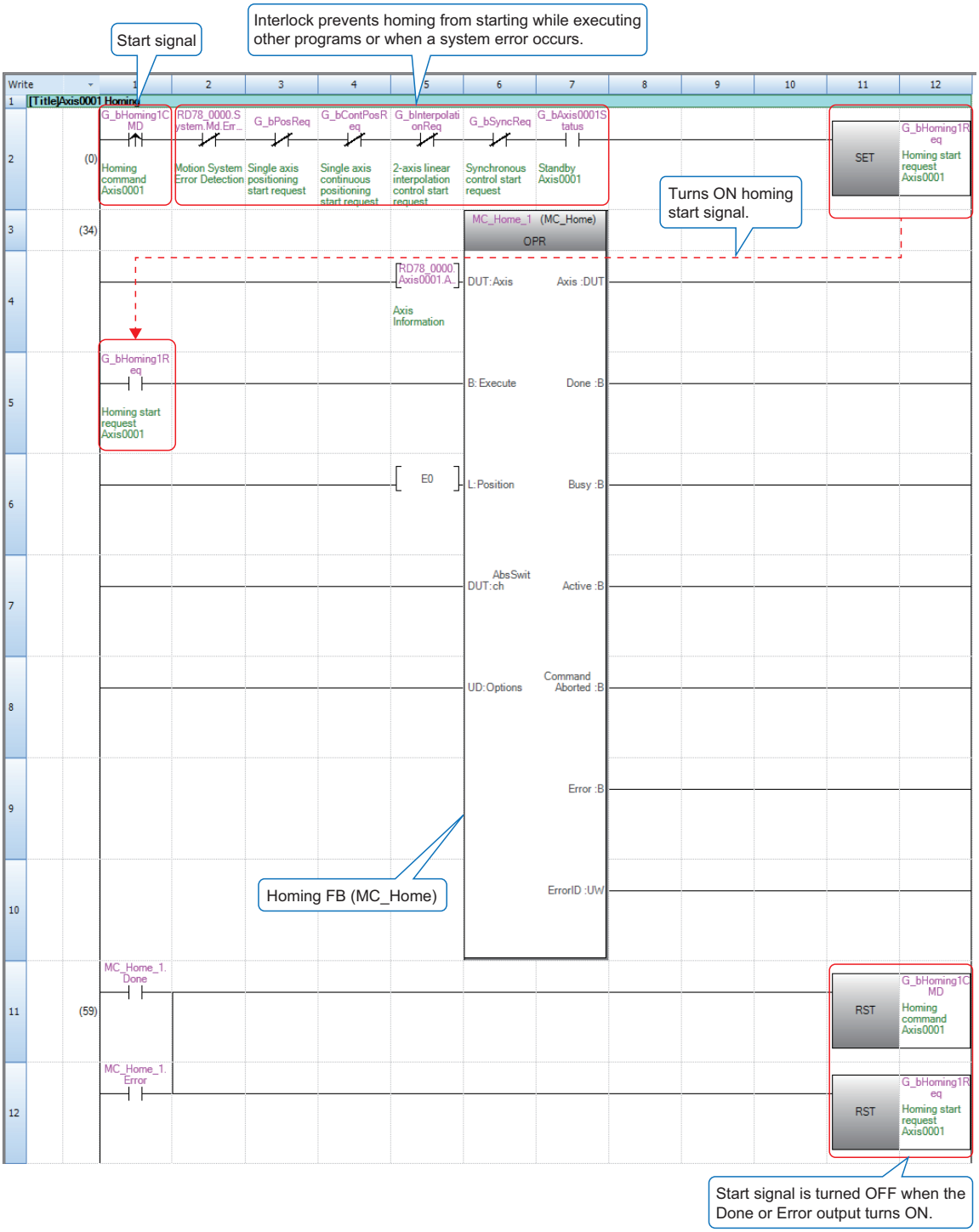
Program example

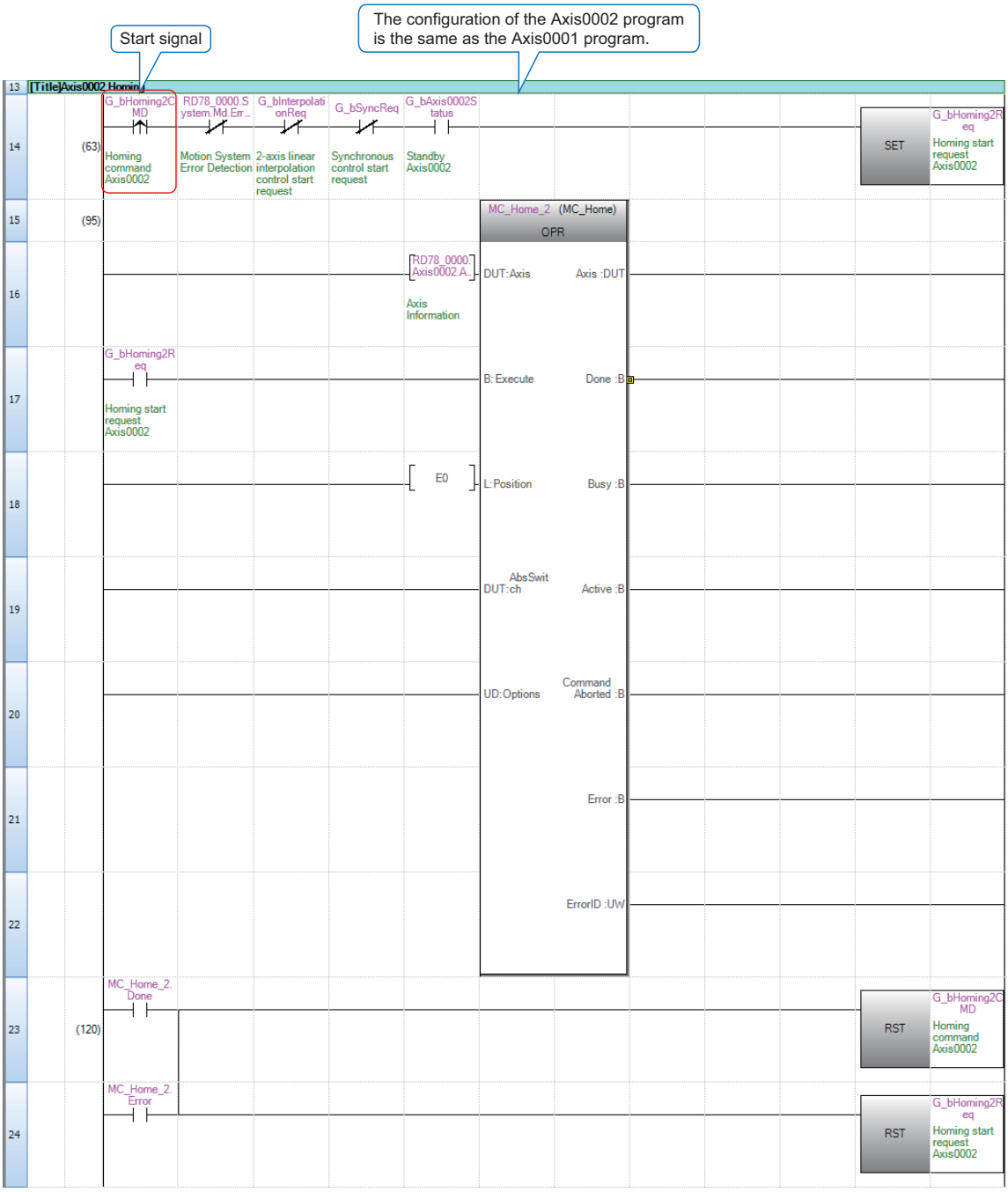
This program executes homing for real drive axes Axis 1 (Axis0001) and Axis 2 (Axis0002) and the virtual drive axis (VirtualAxis0001). The homing request label is kept ON by a self-holding circuit and is connected to the Execute command (Execute) input of the FB (MC_Home). The start signal is turned OFF when the homing completion signal (the Done output of MC_Home) is turned ON or when an error occurs.

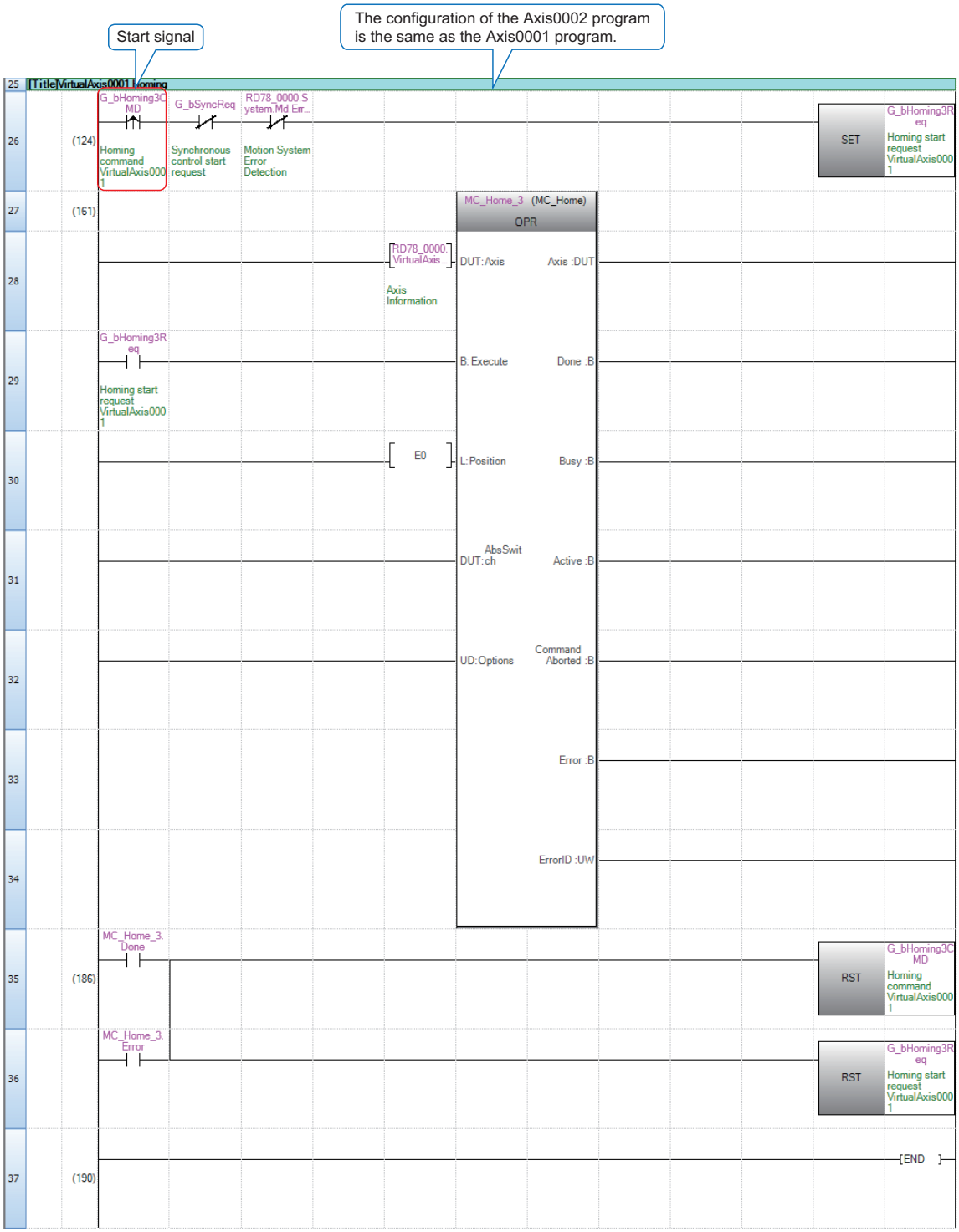
The interlock conditions are added to prevent execution of the homing while other programs are running or when a system error occurs.

• Start signal

Signal name	Label name		
	Axis0001	Axis0002	VirtualAxis0001
Homing command	G_bHoming1CMD	G_bHoming2CMD	G_bHoming3CMD




Start signal
The configuration of the Axis0002 program is the same as the Axis0001 program.
13 [Title]Axis0002 Homing
14 (63)
G_bHoming2CMD
Homing command Axis0002
RD78_0000.System.Md.Err...
Motion System Error Detection
G_bInterpolationReq
2-axis linear interpolation control start request
G_bSyncReq
Synchronous control start request
G_bAxis0002Status
Standby Axis0002
SET
G_bHoming2Req
Homing start request Axis0002
15 (95)
MC_Home_2 (MC_Home)
OPR
16
RD78_0000.Axis0002.A...
Axis Information
DUT:Axis
Axis :DUT
17
G_bHoming2Req
Homing start request Axis0002
B:Execute
Done :B
18
E0
L:Position
Busy :B
19
AbsSwitDUT:ch
Active :B
20
UD:Options
Command Aborted :B
21
Error :B
22
ErrorID :UW
23 (120)
MC_Home_2.Done
RST
G_bHoming2CMD
Homing command Axis0002
24
MC_Home_2.Error
RST
G_bHoming2Req
Homing start request Axis0002


Start signal
The configuration of the Axis0002 program is the same as the Axis0001 program.
25 [Title]VirtualAxis0001 Homing
G_bHoming3CMD
G_bSyncReq
RD78_0000.System.Md.Err...
26 (124)
Homing command VirtualAxis0001
Synchronous control start request
Motion System Error Detection
SET
G_bHoming3Req
Homing start request VirtualAxis0001
27 (161)
MC_Home_3 (MC_Home)
OPR
RD78_0000.VirtualAxis...
28
DUT:Axis
Axis :DUT
Axis Information
G_bHoming3Req
29
B:Execute
Done :B
Homing start request VirtualAxis0001
E0
30
L:Position
Busy :B
AbsSwit
31
DUT:ch
Active :B
32
UD:Options
Command Aborted :B
33
Error :B
34
ErrorID :UW
MC_Home_3.Done
35 (186)
RST
G_bHoming3CMD
Homing command VirtualAxis0001
MC_Home_3.Error
36
RST
G_bHoming3Req
Homing start request VirtualAxis0001
37 (190)
END

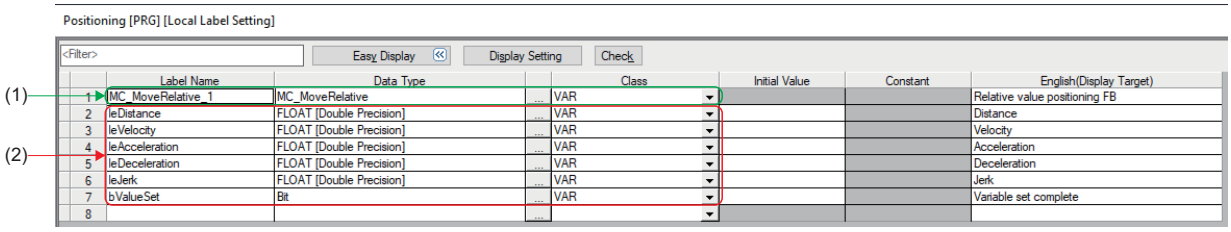
4.8 Single Axis Positioning Control (Program Name: Positioning)

The single axis positioning control executes positioning to a specified position by using the address information.

FBs used

Type	FB	Description
Operation	MC_MoveAbsolute	Executes positioning using a specified absolute position.
	MC_MoveRelative	Executes positioning using a specified relative distance.

Local labels

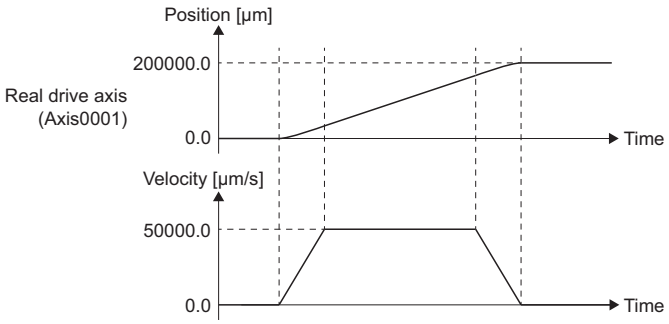


Label Name	Data Type	Class	Initial Value	Constant	English(Display Target)
1 MC_MoveRelative_1	MC_MoveRelative	VAR			Relative value positioning FB
2 leDistance	FLOAT [Double Precision]	VAR			Distance
3 leVelocity	FLOAT [Double Precision]	VAR			Velocity
4 leAcceleration	FLOAT [Double Precision]	VAR			Acceleration
5 leDeceleration	FLOAT [Double Precision]	VAR			Deceleration
6 leJerk	FLOAT [Double Precision]	VAR			Jerk
7 bValueSet	Bit	VAR			Variable set complete
8					

No.	Description
(1)	This label is automatically added when the user drags and drops the FB (MC_MoveRelative) onto the program editor.
(2)	These labels are registered manually.

Program example

This program executes relative positioning in the following operation pattern.

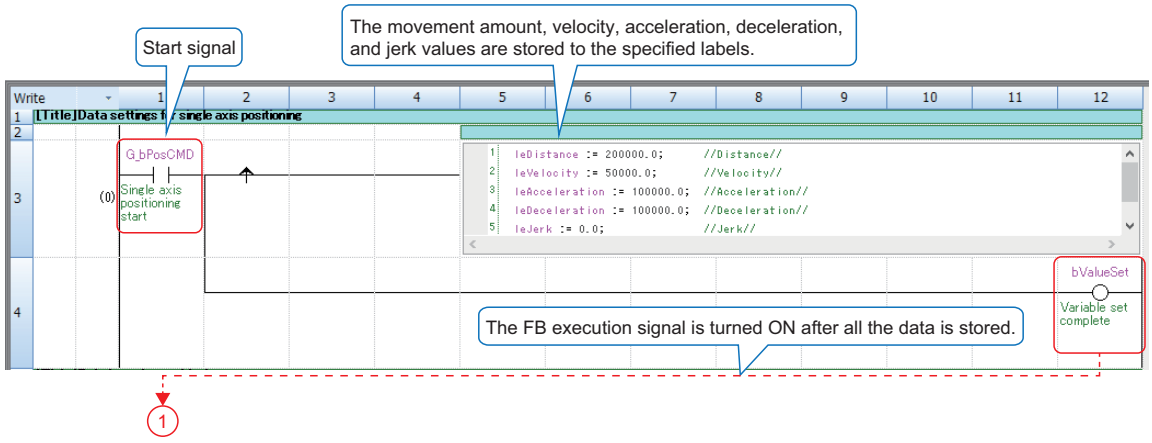


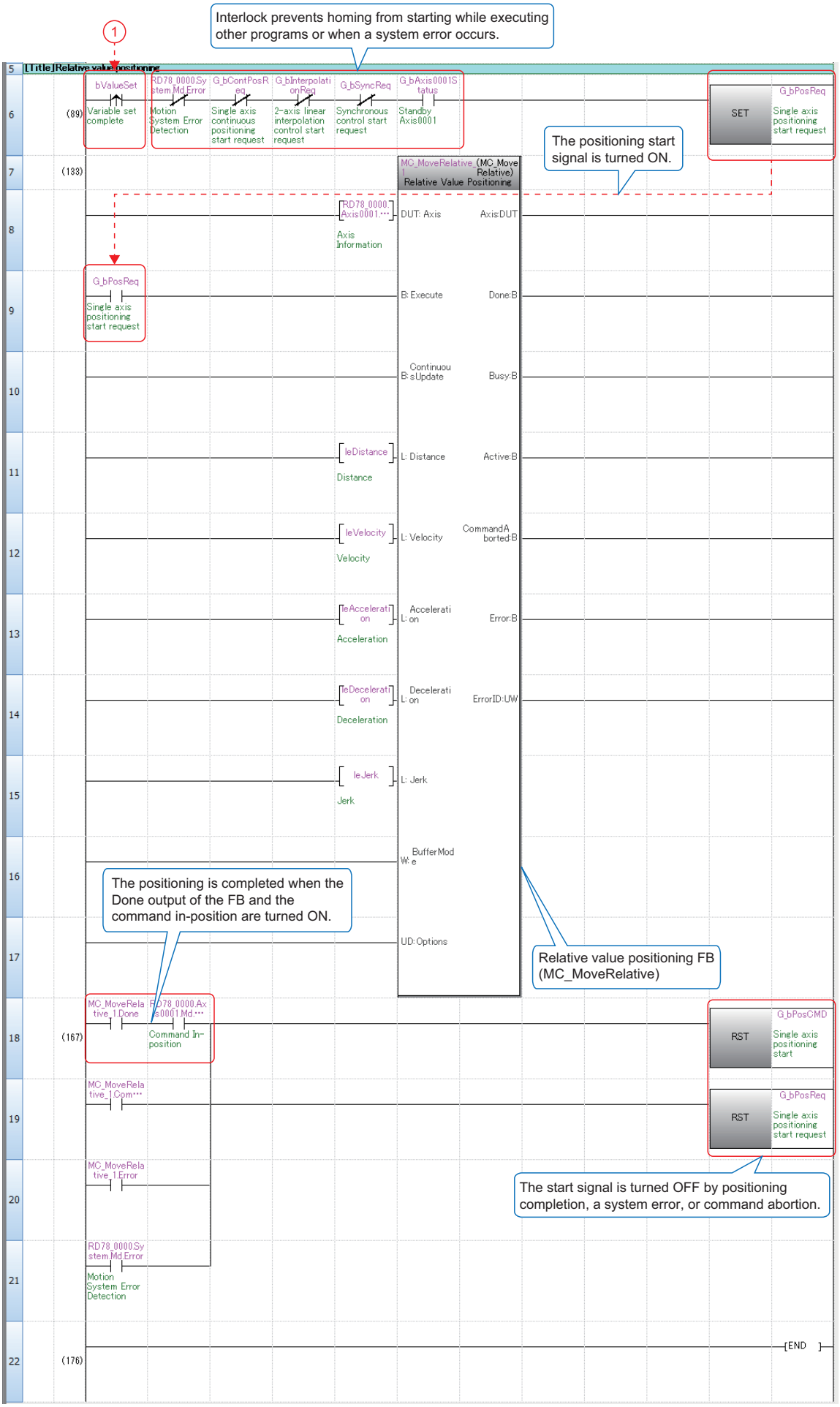
When the single axis positioning command is turned ON, the positioning data is stored to each specified label. When all the data is stored, the execution signal of the relative value positioning FB (MC_MoveRelative) is kept ON by a self-holding circuit. The start signal is turned OFF when any of the following conditions are satisfied: the positioning is completed (the Done output of MC_MoveRelative and command in-position); an error occurs; or the execution is aborted.

The interlock conditions are added to prevent execution of the single axis positioning while other programs are running or when a system error occurs.

- Start signal

Signal name	Label name
Single axis positioning command	G_bPosCMD




Interlock prevents homing from starting while executing other programs or when a system error occurs.
1
[Title]Relative value positioning
bValueSet
Variable set complete
RD78_0000.System_Md.Error
Motion System Error Detection
G_bContPosReq
Single axis continuous positioning start request
G_bInterpolationReq
2-axis linear interpolation control start request
G_bSyncReq
Synchronous control start request
G_bAxis0001Status
Standby Axis0001
SET
G_bPosReq
Single axis positioning start request
The positioning start signal is turned ON.
MC_MoveRelative_1
(MC_MoveRelative)
Relative Value Positioning
RD78_0000.Axis0001...
DUT: Axis
Axis Information
AxisDUT
G_bPosReq
Single axis positioning start request
B: Execute
Done:B
Continuous Update
Busy:B
leDistance
L: Distance
Distance
Active:B
leVelocity
Velocity
CommandAborted:B
leAcceleration
Acceleration
Error:B
leDeceleration
Deceleration
ErrorID:UW
leJerk
Jerk
BufferMode
UD: Options
The positioning is completed when the Done output of the FB and the command in-position are turned ON.
Relative value positioning FB (MC_MoveRelative)
MC_MoveRelative_1.Done
RD78_0000.Axis0001.Md...
Command In-position
RST
G_bPosCMD
Single axis positioning start
MC_MoveRelative_1.Com...
RST
G_bPosReq
Single axis positioning start request
MC_MoveRelative_1.Error
RD78_0000.System_Md.Error
Motion System Error Detection
The start signal is turned OFF by positioning completion, a system error, or command abortion.
END

4.9 Single Axis Continuous Positioning (Program Name: ContinuousPositioning)

Overview

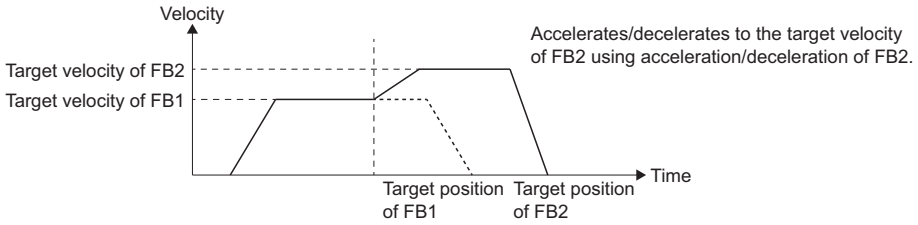
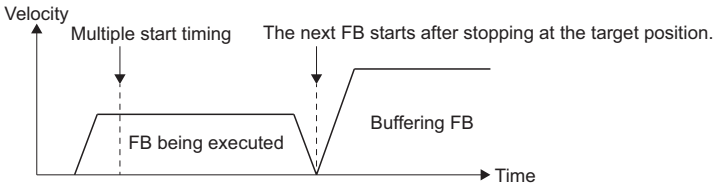
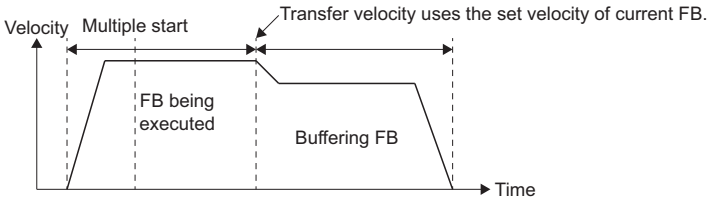
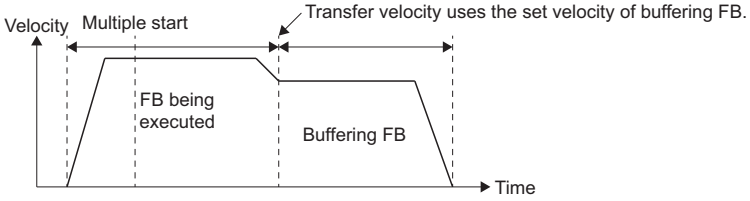
An axis can continuously execute multiple Motion control FBs without stopping by buffering the next operation FB of another instance while executing the first Motion control FB.

Select the buffer mode through the BufferMode input of a Motion control FB.

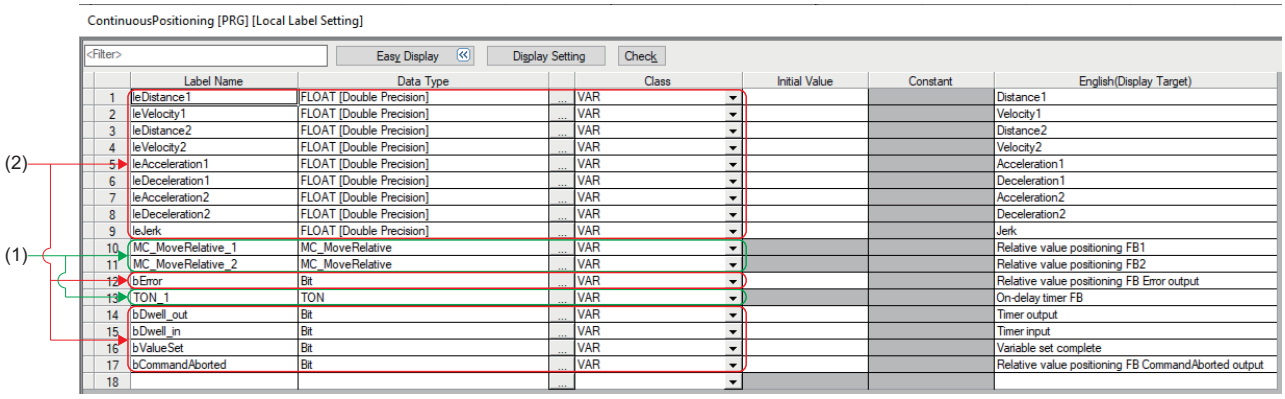
Up to two operation FBs can be buffered on one axis.

The example in this section uses an FB (MC_MoveRelative) to perform relative value positioning.

Operation pattern of buffer mode

Buffer mode	Operation
0: mcAborting	The FB being executed is iaborted (canceled), and the next FB is immediately executed. 
1: mcBuffered	The next FB is executed after the FB being executed completes (the axis decelerates to a stop). 
2: mcBlendingLow	The lower target velocity between the FB being executed and the next FB is used as the switching velocity.
3: mcBlendingPrevious	The next FB is executed after the target position of the FB being executed is reached. The target velocity is the set velocity of the next FB. 
4: mcBlendingNext	The next FB is executed after the target position of the FB being executed is reached. When the target position is reached, the velocity is switched to the set velocity of the next FB. 
5: mcBlendingHigh	The higher target velocity between the FB being executed and the next FB is used as the switching velocity.

Local labels

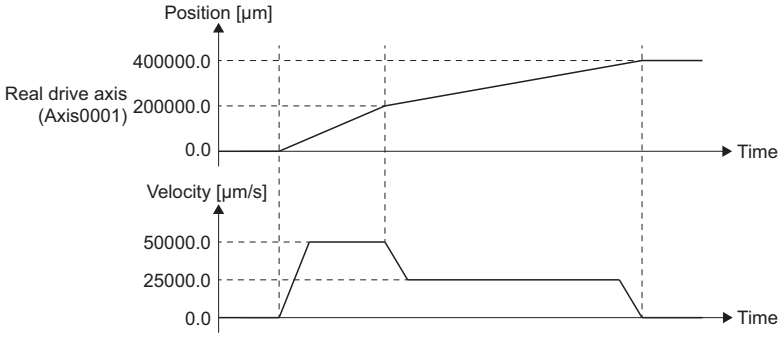


	Label Name	Data Type		Class	Initial Value	Constant	English(Display Target)
1	leDistance1	FLOAT [Double Precision]	...	VAR			Distance1
2	leVelocity1	FLOAT [Double Precision]	...	VAR			Velocity1
3	leDistance2	FLOAT [Double Precision]	...	VAR			Distance2
4	leVelocity2	FLOAT [Double Precision]	...	VAR			Velocity2
5	leAcceleration1	FLOAT [Double Precision]	...	VAR			Acceleration1
6	leDeceleration1	FLOAT [Double Precision]	...	VAR			Deceleration1
7	leAcceleration2	FLOAT [Double Precision]	...	VAR			Acceleration2
8	leDeceleration2	FLOAT [Double Precision]	...	VAR			Deceleration2
9	leJerk	FLOAT [Double Precision]	...	VAR			Jerk
10	MC_MoveRelative_1	MC_MoveRelative	...	VAR			Relative value positioning FB1
11	MC_MoveRelative_2	MC_MoveRelative	...	VAR			Relative value positioning FB2
12	bError	Bit	...	VAR			Relative value positioning FB Error output
13	TON_1	TON	...	VAR			On-delay timer FB
14	bDwell_out	Bit	...	VAR			Timer output
15	bDwell_in	Bit	...	VAR			Timer input
16	bValueSet	Bit	...	VAR			Variable set complete
17	bCommandAborted	Bit	...	VAR			Relative value positioning FB CommandAborted output
18			...				

No.	Description
(1)	These labels are automatically added when the user drags and drops the FBs (MC_MoveRelative and TON) onto the program editor.
(2)	These labels are registered manually.

Program example

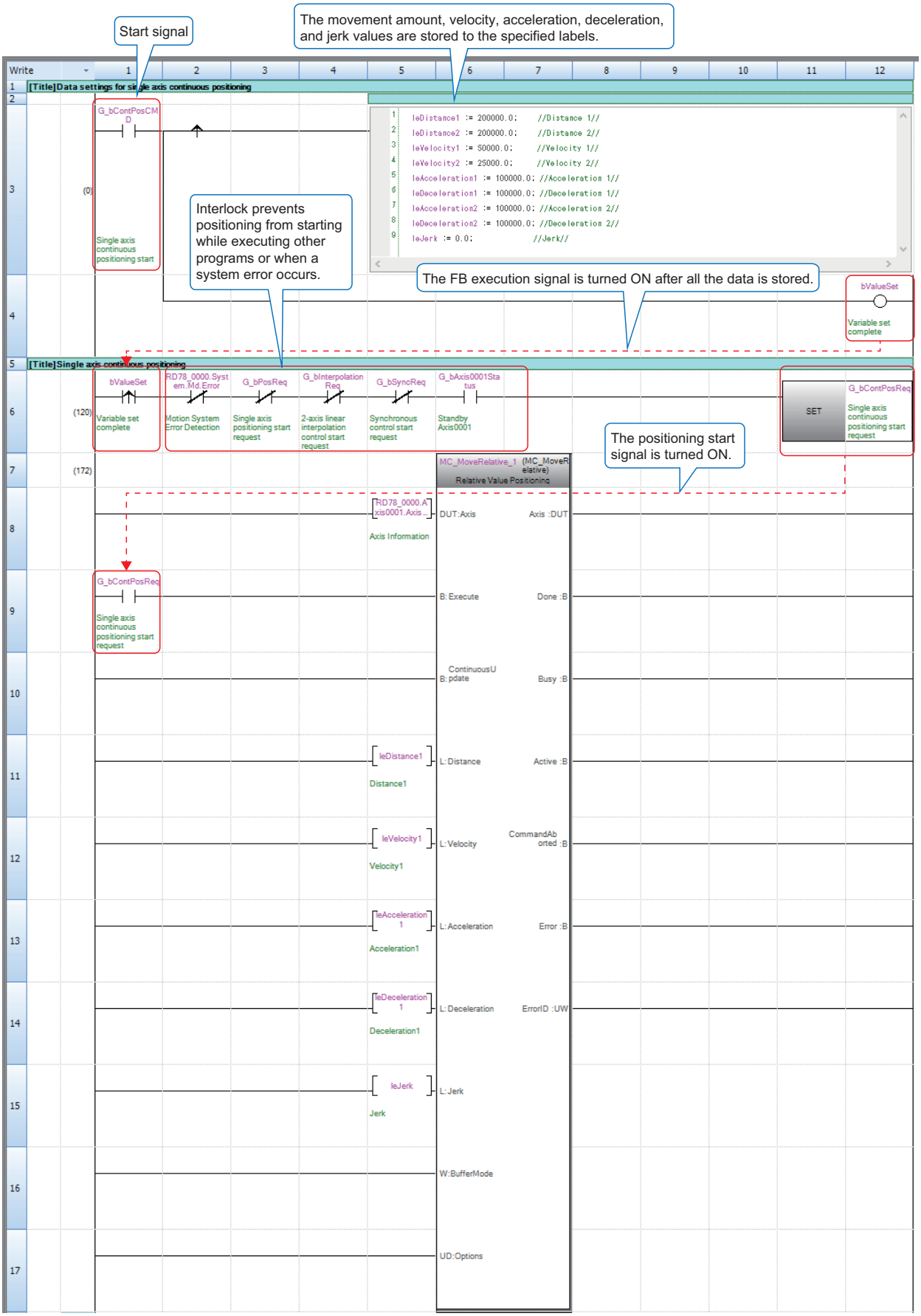
This program executes relative positioning in the following operation pattern.

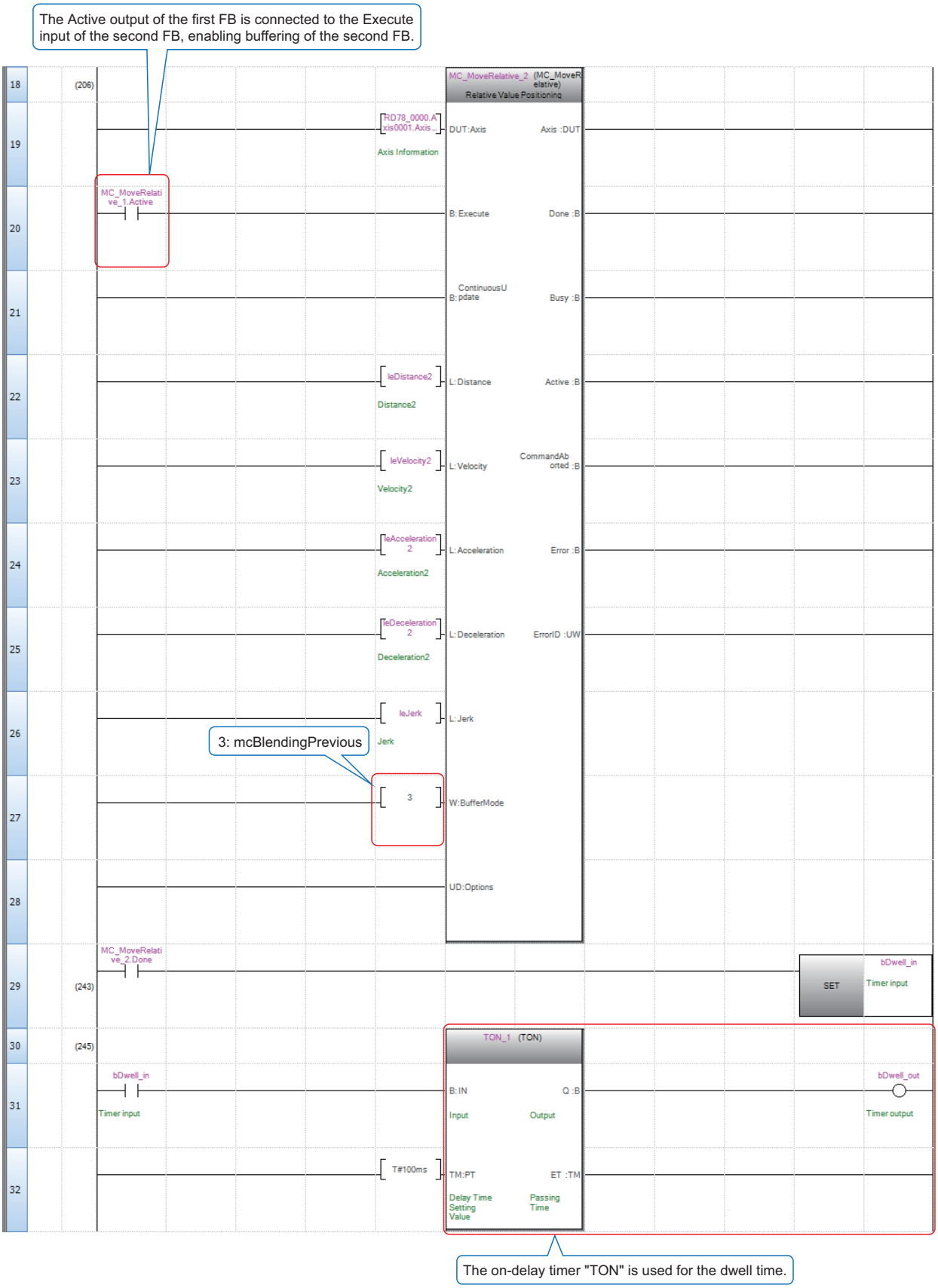


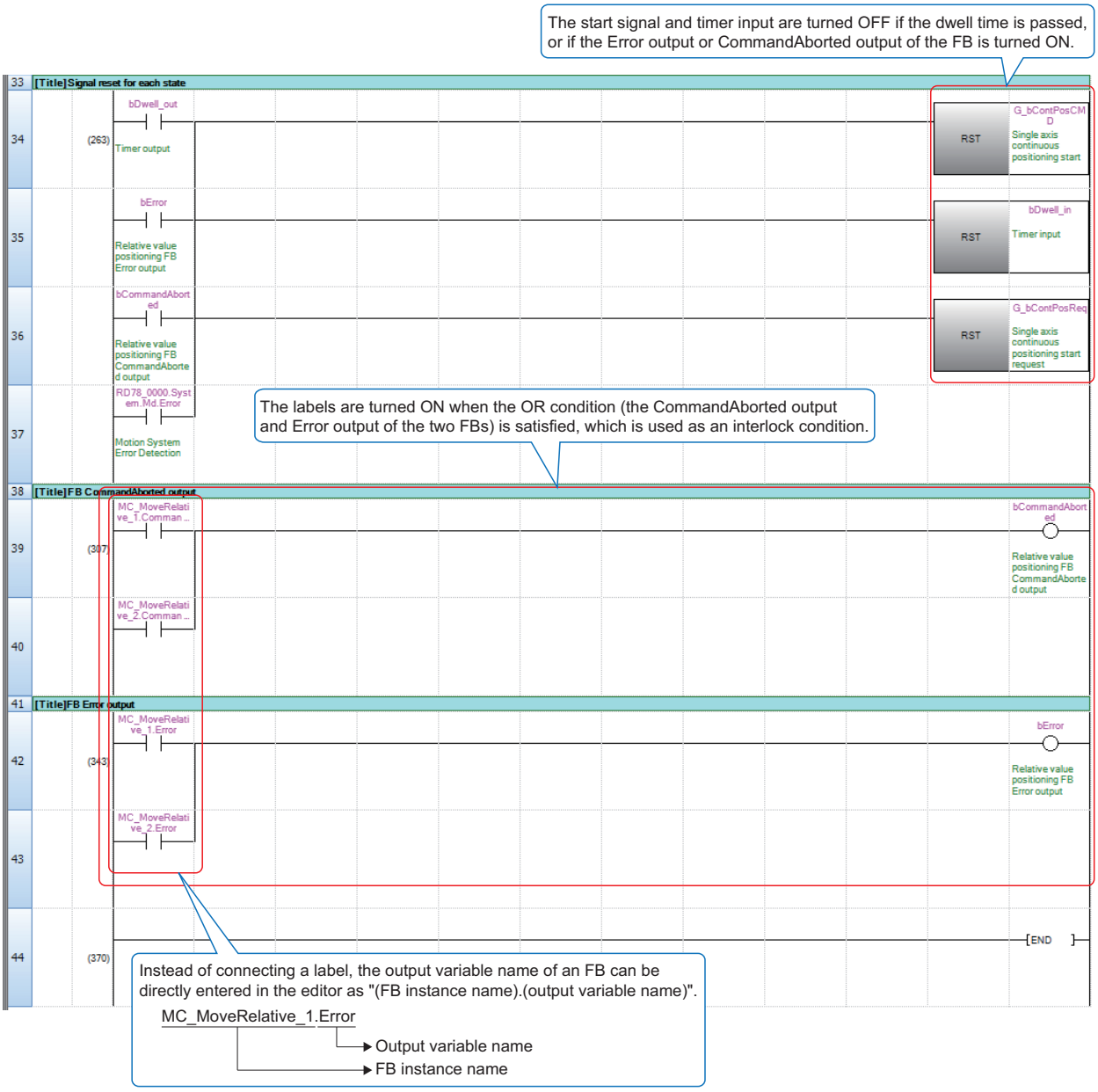
When the single axis positioning command is turned ON, the positioning data is stored to each specified label.
When all the data is stored, the execution signal of the FB (MC_MoveRelative_1) is kept ON by a self-holding circuit.
The Active output of the first FB (MC_MoveRelative_1) is connected to the Execute input of the second FB (MC_MoveRelative_2), which enables buffering of the second FB while executing the first FB.
When the first FB is finished, the second FB is continuously executed. To set the dwell time, the on-delay timer (100 [ms]) is used. The on-delay timer input and the start signal are reset when any of the following conditions are satisfied: the dwell time is passed; an error occurs; or the execution is aborted.
The interlock conditions are added to prevent execution of the single axis continuous positioning while other programs are running or when a system error occurs.

- Start signal

Signal name	Label name
Single axis continuous positioning command	G_bContPosCMD


Start signal
The movement amount, velocity, acceleration, deceleration, and jerk values are stored to the specified labels.
Interlock prevents positioning from starting while executing other programs or when a system error occurs.
The FB execution signal is turned ON after all the data is stored.
The positioning start signal is turned ON.
[Title]Data settings for single axis continuous positioning
[Title]Single axis continuous positioning
G_bContPosCMD
Single axis continuous positioning start
leDistance1 := 200000.0; //Distance 1//
leDistance2 := 200000.0; //Distance 2//
leVelocity1 := 50000.0; //Velocity 1//
leVelocity2 := 25000.0; //Velocity 2//
leAcceleration1 := 100000.0; //Acceleration 1//
leDeceleration1 := 100000.0; //Deceleration 1//
leAcceleration2 := 100000.0; //Acceleration 2//
leDeceleration2 := 100000.0; //Deceleration 2//
leJerk := 0.0; //Jerk//
bValueSet
Variable set complete
RD78_0000.System.Md.Error
Motion System Error Detection
G_bPosReq
Single axis positioning start request
G_bInterpolationReq
2-axis linear interpolation control start request
G_bSyncReq
Synchronous control start request
G_bAxis0001Status
Standby Axis0001
SET
G_bContPosReq
Single axis continuous positioning start request
MC_MoveRelative_1
(MC_MoveRelative)
Relative Value Positioning
RD78_0000.Axis0001.Axis...
Axis Information
DUT:Axis
Axis :DUT
B: Execute
Done :B
ContinuousUB: pdate
Busy :B
leDistance1
Distance1
L: Distance
Active :B
leVelocity1
Velocity1
L: Velocity
CommandAborted :B
leAcceleration1
Acceleration1
L: Acceleration
Error :B
leDeceleration1
Deceleration1
L: Deceleration
ErrorID :UW
leJerk
Jerk
L: Jerk
W:BufferMode
UD:Options

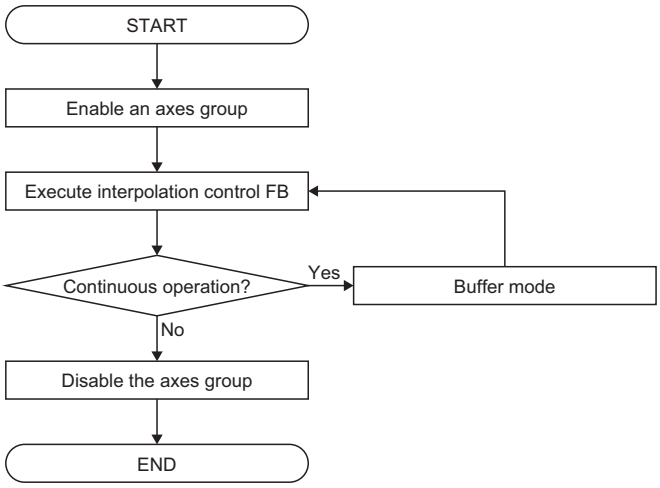

The Active output of the first FB is connected to the Execute input of the second FB, enabling buffering of the second FB.
MC_MoveRelative_2 (MC_MoveRelative)
Relative Value Positioning
RD78_0000.Axis0001.Axis
Axis Information
DUT:Axis
Axis :DUT
MC_MoveRelative_1.Active
B: Execute
Done :B
ContinuousUpdate
Busy :B
leDistance2
Distance2
L: Distance
Active :B
leVelocity2
Velocity2
L: Velocity
CommandAborted :B
leAcceleration2
Acceleration2
L: Acceleration
Error :B
leDeceleration2
Deceleration2
L: Deceleration
ErrorID :UW
leJerk
Jerk
L: Jerk
3: mcBlendingPrevious
3
W:BufferMode
UD:Options
MC_MoveRelative_2.Done
SET
bDwell_in
Timer input
TON_1 (TON)
bDwell_in
Timer input
B: IN
Input
Q :B
Output
bDwell_out
Timer output
T#100ms
TM:PT
Delay Time Setting Value
ET :TM
Passing Time
The on-delay timer "TON" is used for the dwell time.


The start signal and timer input are turned OFF if the dwell time is passed, or if the Error output or CommandAborted output of the FB is turned ON.
33 [Title]Signal reset for each state
34 bDwell_out
(263)
Timer output
35 bError
Relative value positioning FB Error output
36 bCommandAborted
Relative value positioning FB CommandAborted output
37 RD78_0000.System.Md.Error
Motion System Error Detection
RST
G_bContPosCMD
Single axis continuous positioning start
RST
bDwell_in
Timer input
RST
G_bContPosReq
Single axis continuous positioning start request
The labels are turned ON when the OR condition (the CommandAborted output and Error output of the two FBs) is satisfied, which is used as an interlock condition.
38 [Title]FB CommandAborted output
39 MC_MoveRelative_1.Command...
(307)
40 MC_MoveRelative_2.Command...
bCommandAborted
Relative value positioning FB CommandAborted output
41 [Title]FB Error output
42 MC_MoveRelative_1.Error
(343)
43 MC_MoveRelative_2.Error
bError
Relative value positioning FB Error output
44 (370)
[END]
Instead of connecting a label, the output variable name of an FB can be directly entered in the editor as "(FB instance name).(output variable name)".
MC_MoveRelative_1.Error
Output variable name
FB instance name

4.10 Interpolation Control (Program Name: LinearInterpolation)

Procedure for interpolation control

The following shows the procedure flow for executing interpolation control of two or more axes.



Enabling/disabling an axes group

For axes group setting, refer to the following.

☞ Page 55 Axes Group Setting

To execute interpolation control, transit the axes group status to "4: Standby (GroupStandby)" and enable the axes group.

FBs used

Type	FB	Description
Management	MC_GroupEnable	Transits the specified axes group status from "0: Axes group disabled (GroupDisabled)" to "4: Standby (GroupStandby)".
	MC_GroupDisable	Transits the specified axes group status to "0: Axes group disabled (GroupDisabled)".

Interpolation control

For FBs that perform interpolation control, FBs for linear and circular interpolation control are available. Execute the following FBs after the axes group is enabled.

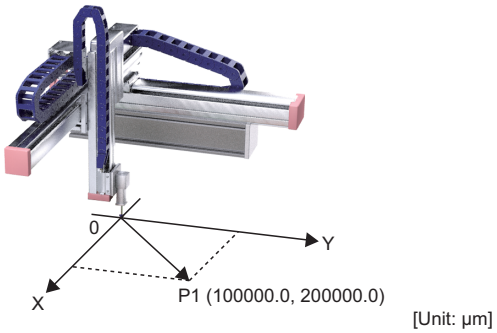
FBs used

Type	FB	Description
Operation	MCv_MoveLinearInterpolateAbsolute	Specifies the target position of the absolute position of the set axes group, then executes positioning by linear interpolation control.
	MCv_MoveLinearInterpolateRelative	Specifies the movement amount of the relative position of the set axes group, then executes positioning by linear interpolation control.
	MCv_MoveCircularInterpolateAbsolute	Executes positioning with 2-axis circular interpolation using the structuring axes of the set axes group by setting the end point and sub point of the absolute position.
	MCv_MoveCircularInterpolateRelative	Executes positioning with 2-axis circular interpolation using the structuring axes of the set axes group by setting the relative position from the current position at start to the end point and sub point.

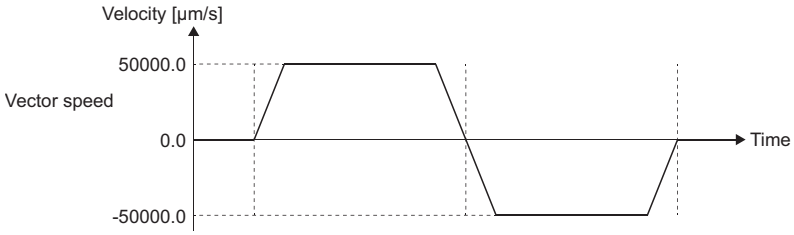
Program example for linear interpolation

Operation pattern

The machine goes back and forth between the origin point (0.0, 0.0) [μm] and point P1 (100000.0, 200000.0) [μm].



[Unit: μm]



Linear interpolation axis and movement amount settings

■LinearAxes input of MCv_MoveLinearInterpolateRelative

Set the data type of the label to connect to the LinearAxes input as INT type (Word [signed]) with "16" elements in the array.

In the sample program, the labels for wAxes[0] to wAxes[15] are used.

For the AxesGroup input, the interpolation control axes are selected from configuration axis[1] to [16] set in the axes group parameter. Specify the configuration axis No. to be used for the interpolation control by wAxes. (Note that the wAxes must be set in order starting from wAxes[0].)

■Distance input of MCv_MoveLinearInterpolateRelative

Set the data type of the label to connect to the LinearAxes input as LREAL type (double-precision real number) with "16" elements in the array.

In the sample program, the labels for lePosition[0] to lePosition[15] are used.

For AxesGroup, specify the movement amount of configuration axis[1] to [16] set in the axes group parameter in lePosition[0] to lePosition[15].

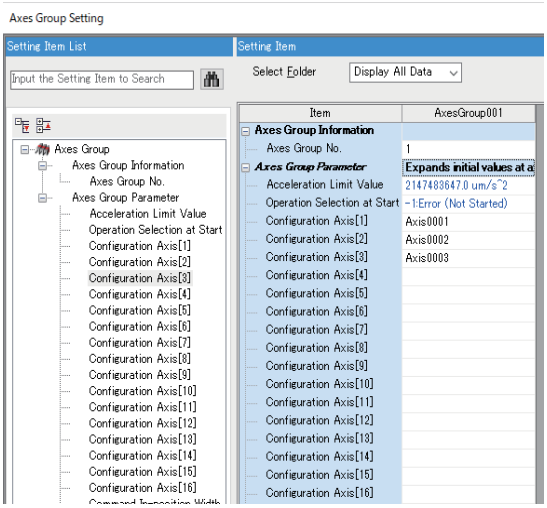
Point

Regardless of the number of the interpolation control axes, the number of elements must be "16" for the arrays used in both the LinearAxes input data type (INT type) and the Distance input data type (LREAL type).

■Setting examples

When setting the following axes group

Item	Data name
Configuration Axis[1]	Axis0001
Configuration Axis[2]	Axis0002
Configuration Axis[3]	Axis0003



Ex.

When using the following 2 axes as linear interpolation axes: configuration axis 1 (Axis0001) and configuration axis 2 (Axis0002)

```
wAxes[0]:= 1; //configuration axis[1]
wAxes[1]:= 2; //configuration axis[2]

lePosition[0]:= (movement amount of configuration axis 1 (Axis0001));
lePosition[1]:= (movement amount of configuration axis 2 (Axis0002));
```

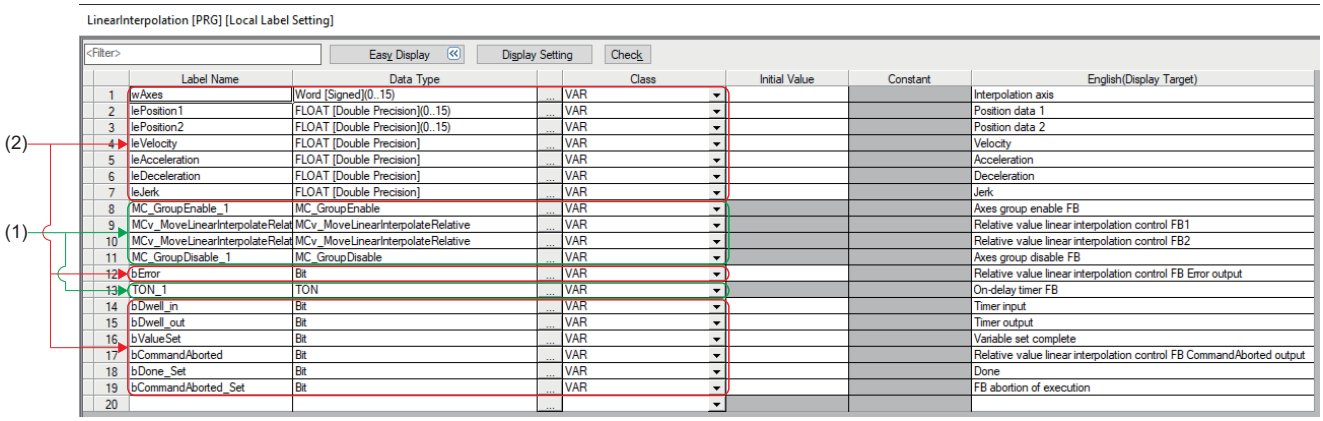
Ex.

When using the following 2 axes as linear interpolation axes: configuration axis 2 (Axis0002) and configuration axis 3 (Axis0003)

```
wAxes[0]:= 2; //configuration axis[2]
wAxes[1]:= 3; //configuration axis[3]

lePosition[0]:= 0.0;
lePosition[1]:= (movement amount of configuration axis 2 (Axis0002));
lePosition[2]:= (movement amount of configuration axis 3 (Axis0003));
```

Local labels



<Filter> | Easy Display | Display Setting | Check

	Label Name	Data Type	Class	Initial Value	Constant	English(Display Target)
1	iwAxes	Word [Signed](0..15)	VAR			Interpolation axis
2	iePosition1	FLOAT [Double Precision](0..15)	VAR			Position data 1
3	iePosition2	FLOAT [Double Precision](0..15)	VAR			Position data 2
4	ieVelocity	FLOAT [Double Precision]	VAR			Velocity
5	ieAcceleration	FLOAT [Double Precision]	VAR			Acceleration
6	ieDeceleration	FLOAT [Double Precision]	VAR			Deceleration
7	ieJerk	FLOAT [Double Precision]	VAR			Jerk
8	MC_GroupEnable_1	MC_GroupEnable	VAR			Axes group enable FB
9	MCv_MoveLinearInterpolateRelat	MCv_MoveLinearInterpolateRelative	VAR			Relative value linear interpolation control FB1
10	MCv_MoveLinearInterpolateRelat	MCv_MoveLinearInterpolateRelative	VAR			Relative value linear interpolation control FB2
11	MC_GroupDisable_1	MC_GroupDisable	VAR			Axes group disable FB
12	bError	Bit	VAR			Relative value linear interpolation control FB Error output
13	TON_1	TON	VAR			On-delay timer FB
14	bDwell_in	Bit	VAR			Timer input
15	bDwell_out	Bit	VAR			Timer output
16	bValueSet	Bit	VAR			Variable set complete
17	bCommandAborted	Bit	VAR			Relative value linear interpolation control FB CommandAborted output
18	bDone_Set	Bit	VAR			Done
19	bCommandAborted_Set	Bit	VAR			FB abortion of execution
20						

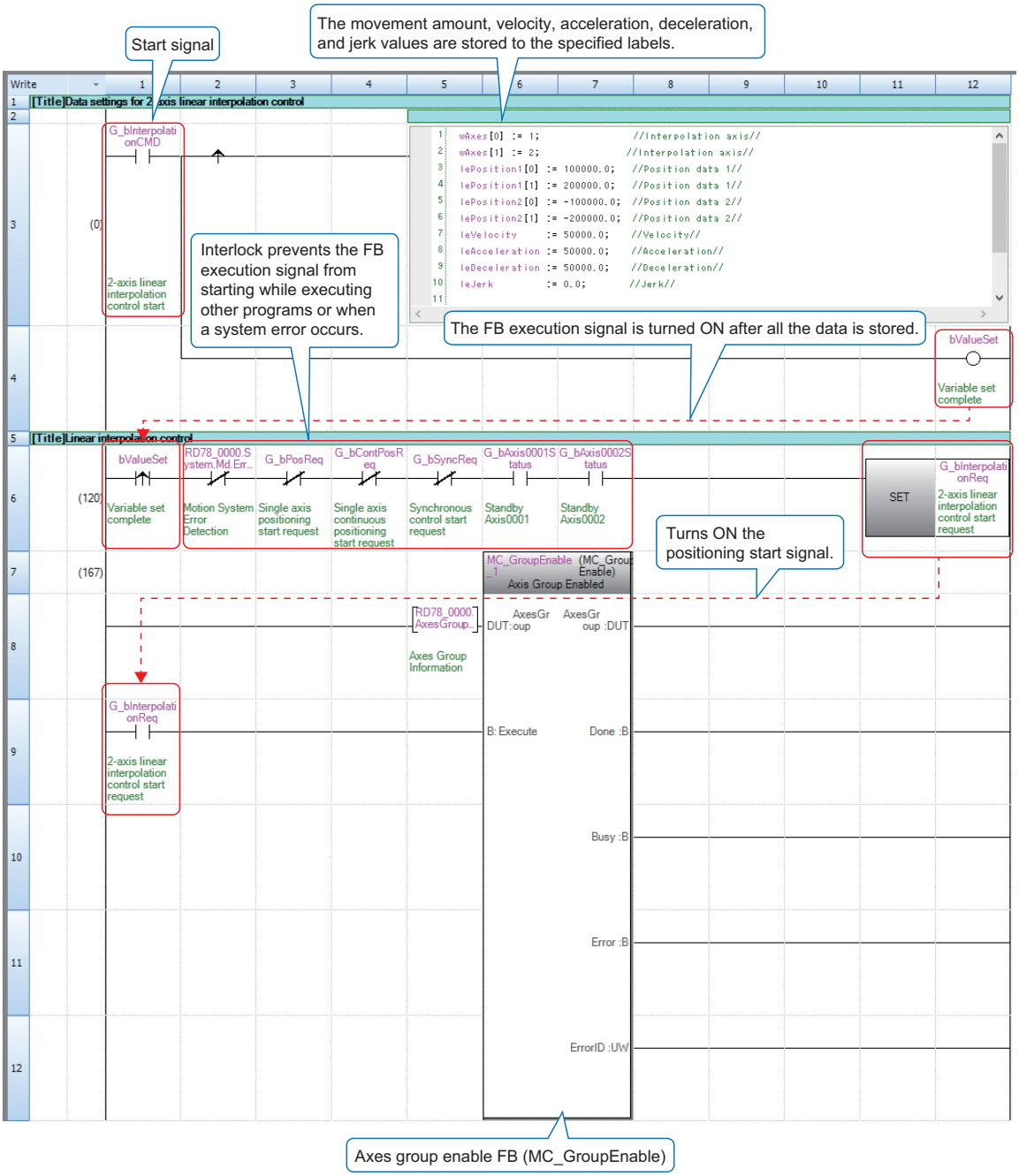
No.	Description
(1)	These labels are automatically added when the user drags and drops the FBs (MC_GroupEnable, MCv_MoveLinearInterpolateRelative, MC_GroupDisable, and TON) onto the program editor.
(2)	These labels are registered manually.

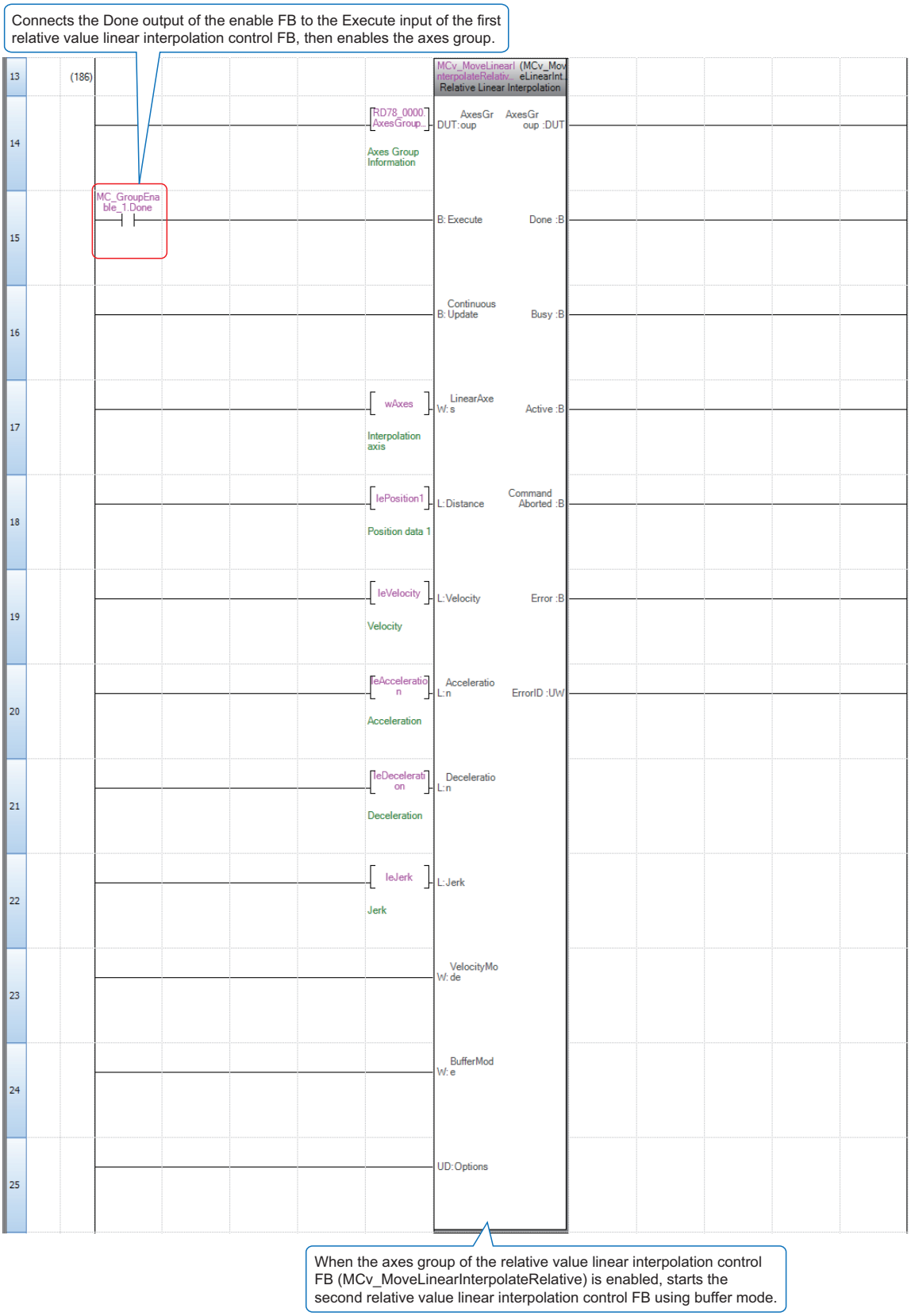
Program example

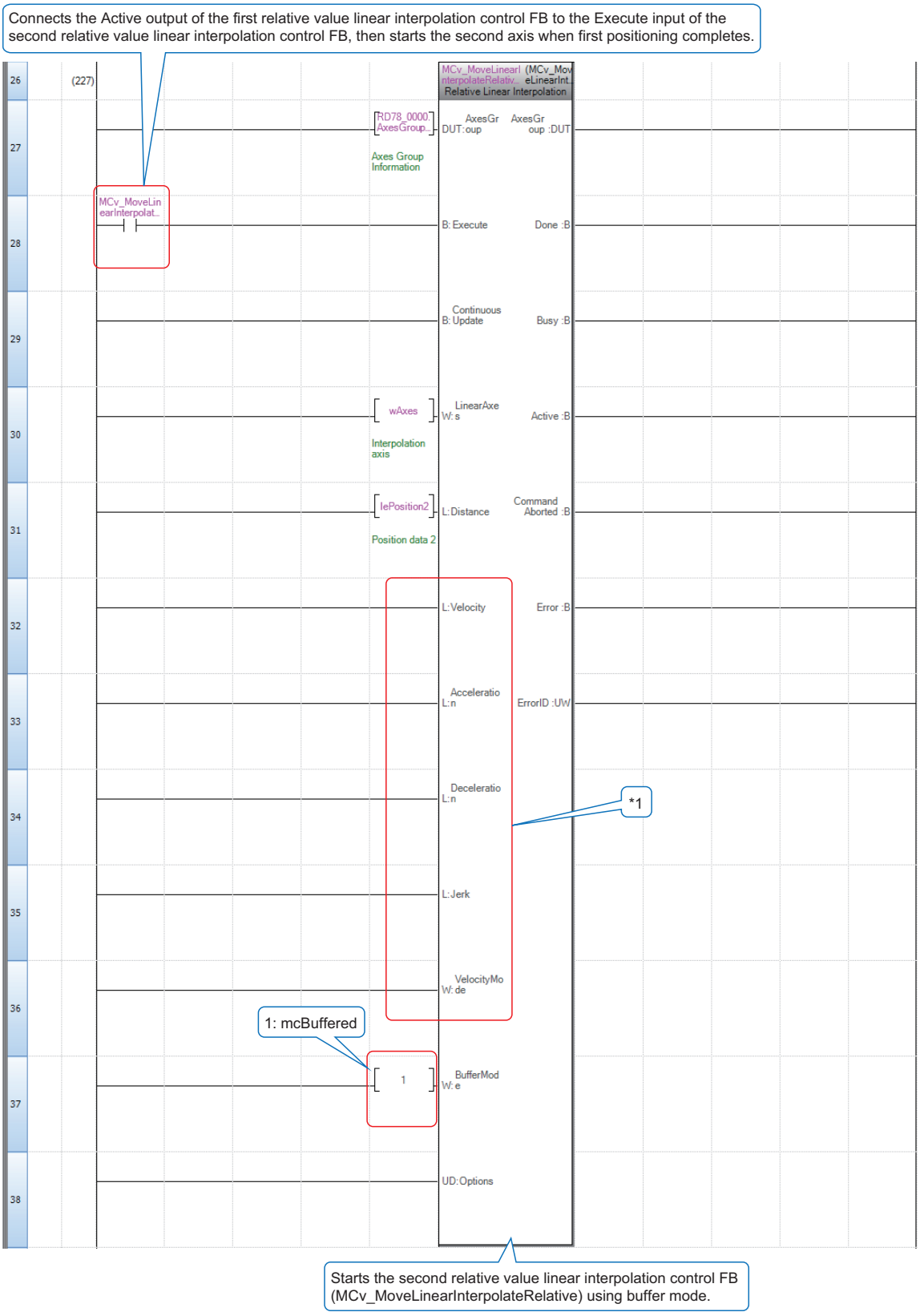
When the two-axis interpolation control start is turned ON, the positioning data is stored to each specified label.
When all the data is stored, the execution signal of the FB (MC_GroupEnable_1) is kept ON through a self-holding circuit.
After the axes group is enabled by the FB (MC_GroupEnable), the two FBs for relative value linear interpolation control (MCv_MoveLinearInterpolateRelative) are started in buffer mode.
The axes group is disabled when any of the following conditions are satisfied: the dwell time is passed after the positioning is completed; an error occurs; or the execution is aborted. The on-delay timer and the start signal are reset when the axes group is disabled.
The interlock conditions are added to prevent execution of the two-axis interpolation control while other programs are running or when a system error occurs.

- Start signal

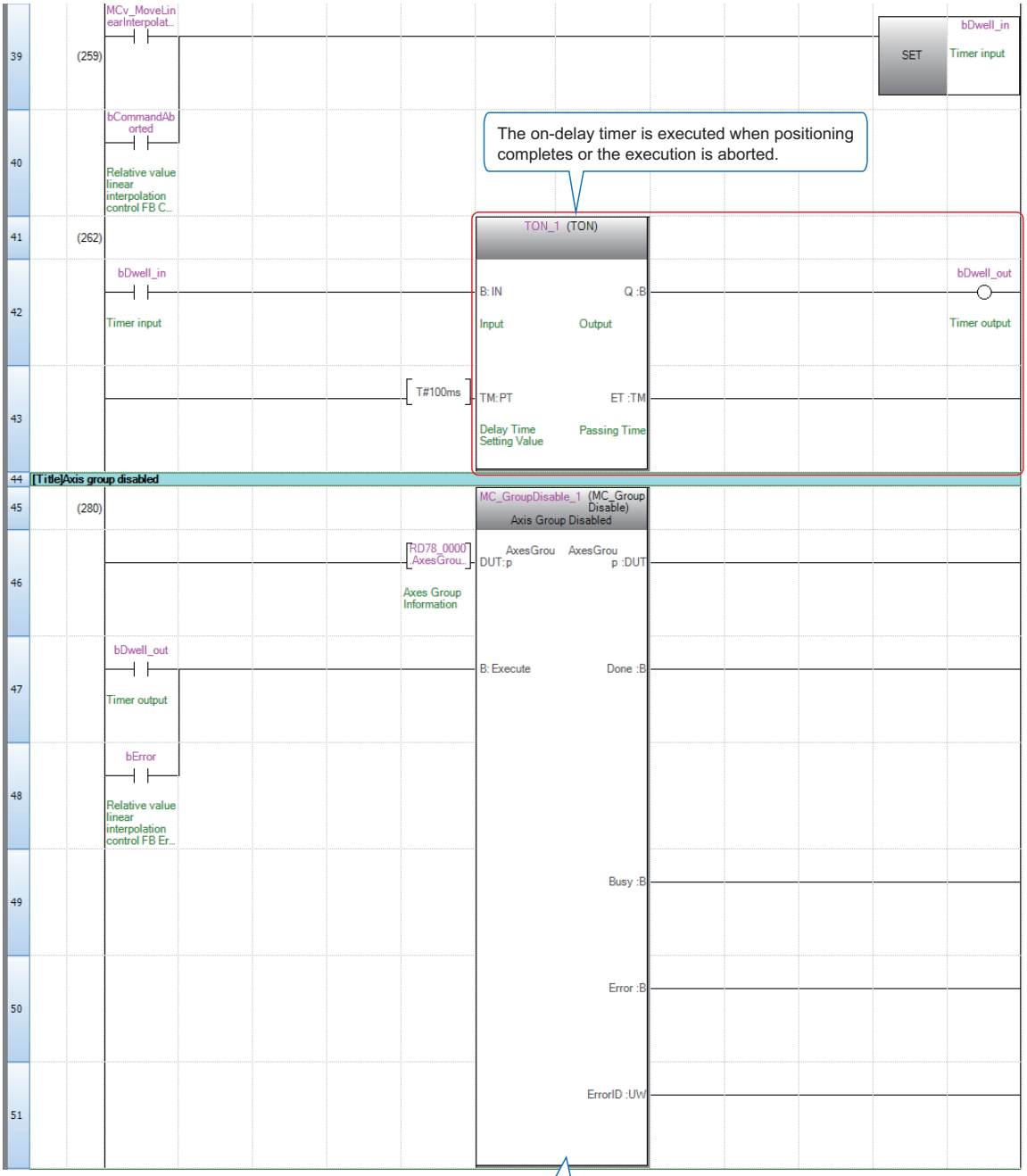
Signal name	Label name
Two-axis linear interpolation control start	G_bInterpolationCMD




Connects the Done output of the enable FB to the Execute input of the first relative value linear interpolation control FB, then enables the axes group.
13
(186)
MCv_MoveLinearI nterpolateRelativ...
(MCv_Mov eLinearInt...
Relative Linear Interpolation
14
D78_0000: AxesGroup...
AxesGr DUT:oup
AxesGr oup :DUT
Axes Group Information
15
MC_GroupEna ble_1.Done
B: Execute
Done :B
16
Continuous B: Update
Busy :B
17
wAxes
LinearAxe W: s
Active :B
Interpolation axis
18
ePosition1
L: Distance
Command Aborted :B
Position data 1
19
eVelocity
L: Velocity
Error :B
Velocity
20
eAcceleratio n
Acceleratio L:n
ErrorID :UW
Acceleration
21
eDeceleratio n
Deceleratio L:n
Deceleration
22
eJerk
L: Jerk
Jerk
23
VelocityMo W: de
24
BufferMod W: e
25
UD: Options
When the axes group of the relative value linear interpolation control FB (MCv_MoveLinearInterpolateRelative) is enabled, starts the second relative value linear interpolation control FB using buffer mode.

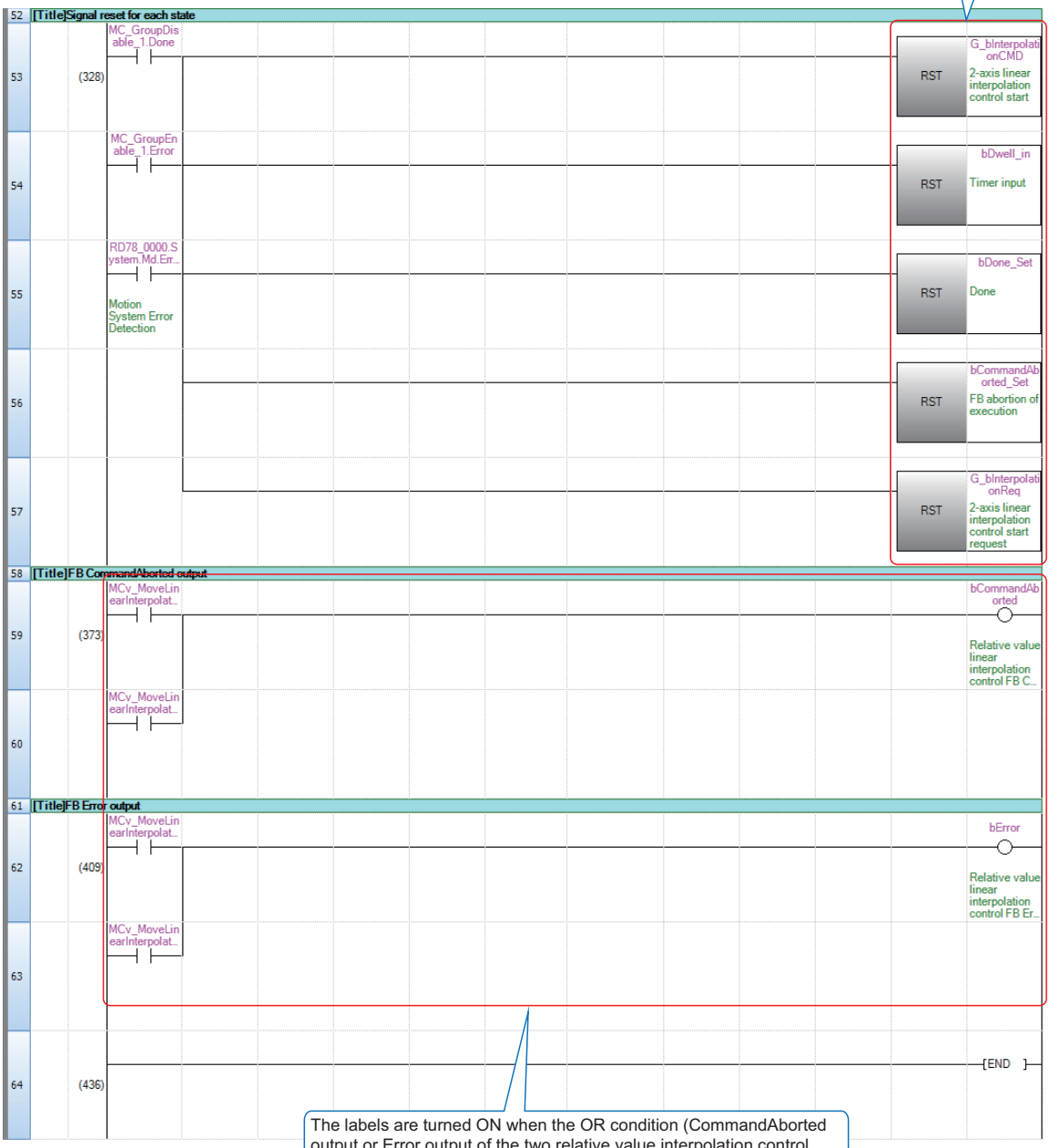


*1 When the values for the Velocity, Acceleration, and Deceleration inputs are omitted for the buffering FB, those of the previous FB are applied.



Axes group disable FB (MC_GroupDisable) disables the axes group when the dwell time is passed or an error occurs in linear interpolation control.

The start signal and input signal, etc. are turned OFF when one of the following conditions occurs:
· A group disable completes
· A group enable error occurs
· A system error occurs

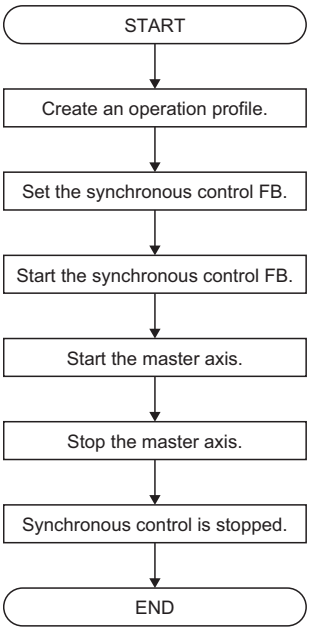


The labels are turned ON when the OR condition (CommandAborted output or Error output of the two relative value interpolation control FBs) is satisfied, which is used as an interlock condition.

4.11 Synchronous Control (Program Name: Synchronous)

Procedures for executing synchronous control

The following shows the procedure flow for executing synchronous control.



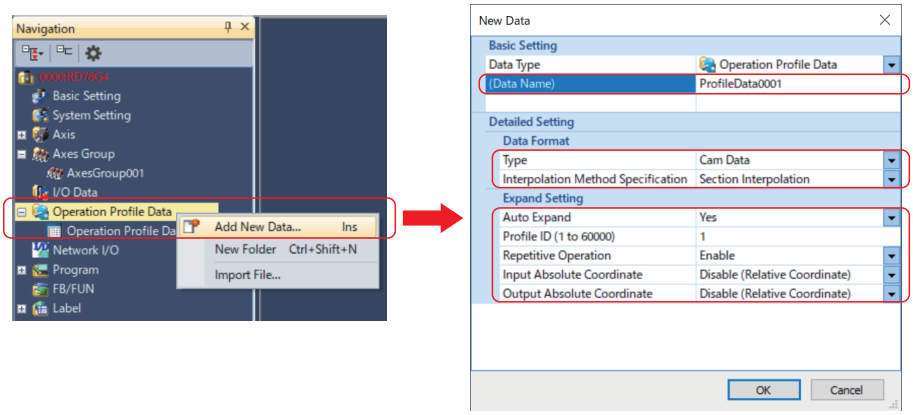
Operation profile

Waveform data used for control are collectively called operation profiles.

This section explains how to create cam data.

Create a new operation profile

1. In the navigation window of Motion Control Setting Function, right-click "Operation Profile Data" and select "Add New Data". The "New Data" screen is displayed.



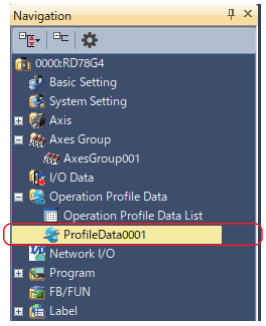
2. Set the data name, type, interpolation method specification, etc., then click the [OK] button.

The example in this document uses the initial value.

The following shows the items set in the expand setting.

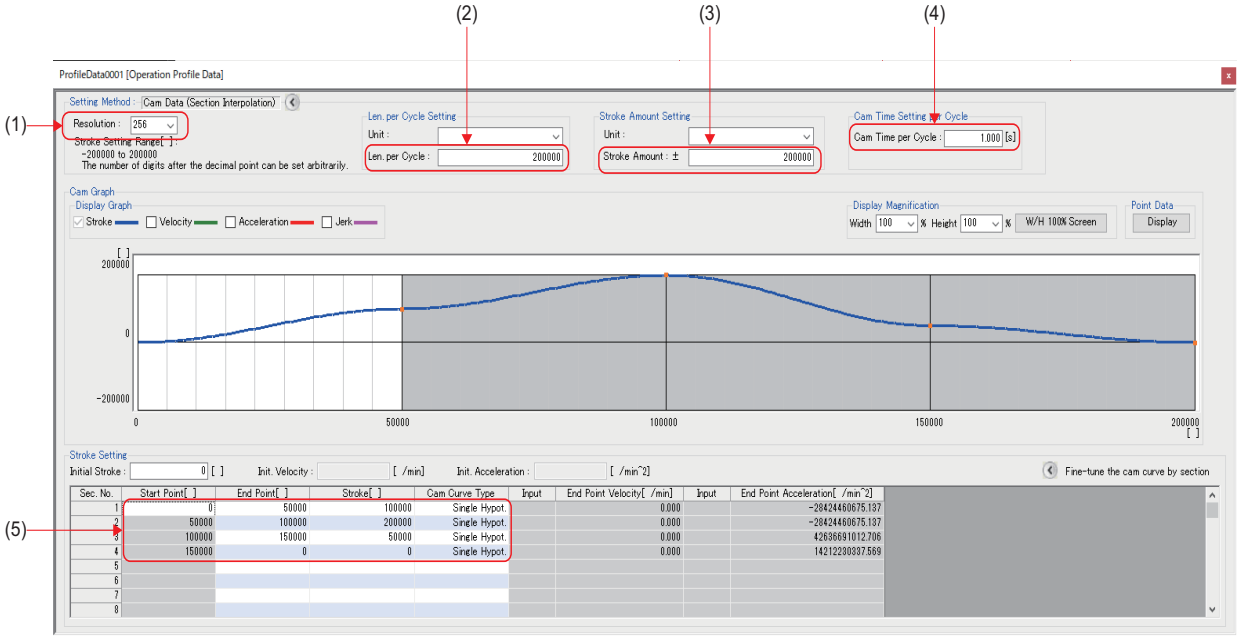
Setting item	Description
Auto expand	• Yes: The operation profile data is automatically opened at power-on. • No: The open FB for operation profile data needs to be executed.
Repetitive operation	• Disable: The control ends when it executes until the end of operation profile data. • Enable: The execution of operation profile data is continuously repeated.
Output absolute coordinate	• Disable (relative coordinate): When the operation profile (cam data) is started, an output value is calculated based on the current value. When executing a feed cam operation, select this setting. • Enable (absolute coordinate): The output value at the time the operation profile (cam data) is started is calculated to be always the start point for one cycle of the operation profile data. When the start point and the end point of the operation profile data are different, the command is output in one operation cycle in order to return to the first output value at the next one cycle start.

3. The operation profile is added to the navigation window.



Creating cam data

Set the operation profile waveform.



The following shows the items set with the operation profile.

In this section, the following settings are used.

No.	Item	Setting value	Description
(1)	Resolution	256	Set the resolution of the cam data.
(2)	Length per cycle setting	200000 (blank for the unit setting)	Set the length per cycle and its unit. (Set the movement amount of the master axis for one cam cycle.)
(3)	Stroke amount setting	200000 (blank for the unit setting)	Set the stroke amount and its unit. (Set the movement amount of the slave axis for one cam cycle.)
(4)	Cam time setting per cycle	1.000	Set the time for one cam cycle. This setting is used when velocity, acceleration, and jerk are calculated.
(5)	Stroke setting	*1	Set the stroke.

*1 Setting values for the stroke setting

Section No.	Start point	End point	Stroke	Cam curve type
1	0	50000	100000	Single hypotenuse
2	50000	100000	200000	Single hypotenuse
3	100000	150000	50000	Single hypotenuse
4	150000	0	0	Single hypotenuse

Point

When executing linear cam operation (the same operation as the master axis, or an operation that changes the master axis velocity based on a specified velocity ratio), create an operation profile for the linear cam, or use MC_GearIn.

The operation profile data for the linear cam is not provided in the system.

Single axis synchronous control FBs

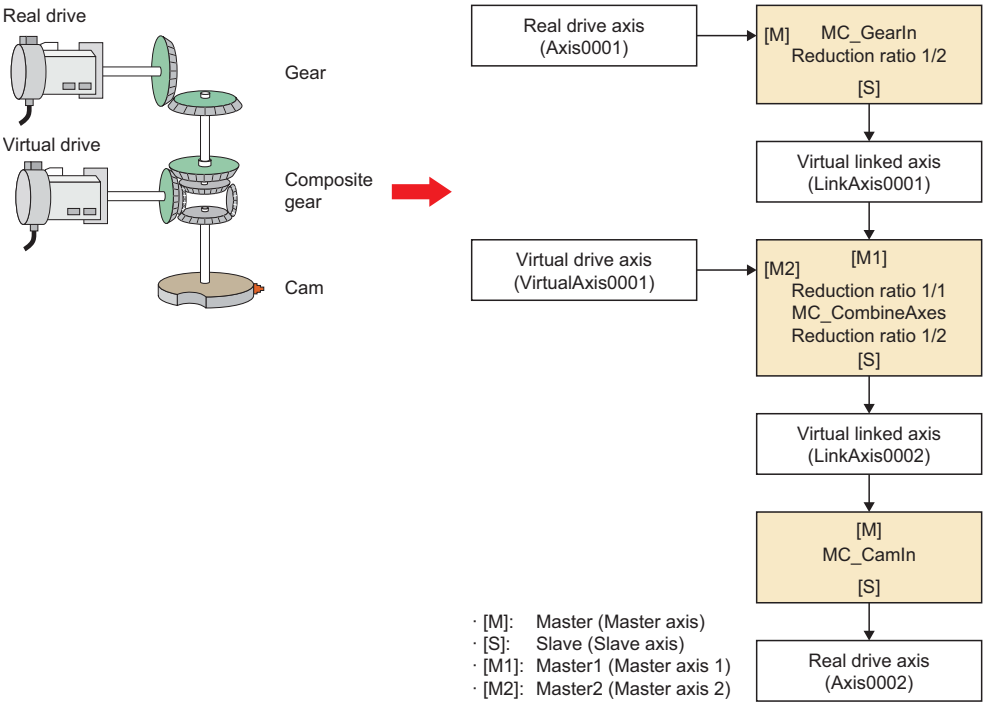
The single axis synchronous control FBs operate as software-based mechanical modules such as gears, speed change gears, and cams. These FBs transmit the position information (command) of Slave that is synchronized with Master.

FBs used

Type	FB	Description
Operation	MC_CamIn	Executes cam operation.
	MC_GearIn	Executes gear operation based on the specified velocity ratio between the master axis and the slave axis.
	MC_CombineAxes	Combines the motion of two axes by a selectable combination method and outputs the result to the third axis.
	MCv_BacklashCompensationFilter	Executes the specific filter processing to the input of Master and outputs the result to Slave.
	MCv_SmoothingFilter	
	MCv_DirectionFilter	
	MCv_SpeedLimitFilter	
	MC_Stop	Decelerates the specified axis to a stop. Used to stop the synchronous control.
Management	MCv_ChangeCycle	Changes the cam current value per cycle to the specified value during MC_CamIn control. It is used to compensate the cam current value per cycle into an arbitrary value.

Axes configuration

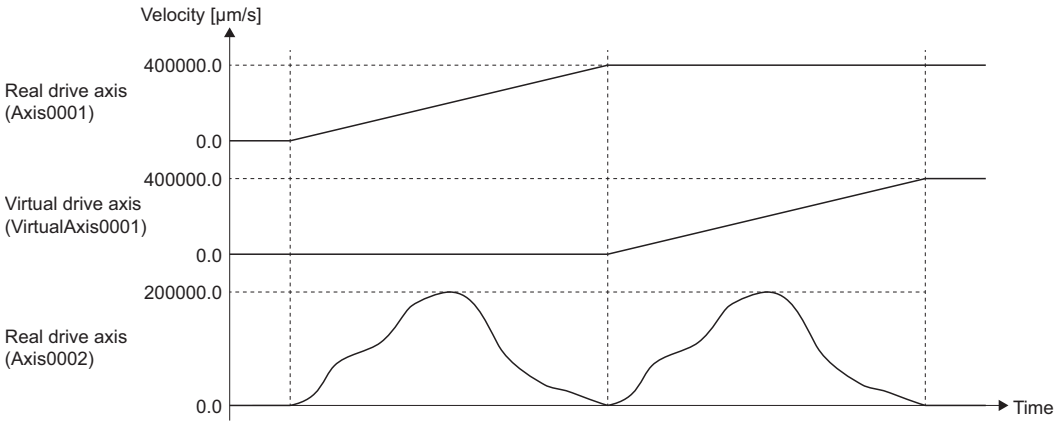
This chapter explains the following cam system.



Program example for synchronous control

Operation pattern

The X-axis (Axis0001) moves from the origin point for 400000.0 [um] while the Y-axis (Axis0002) is operated according to the cam pattern created in Operation profile (☞ Page 115 Operation profile). When the X-axis reaches the target position (400000.0 [um]), the virtual drive axis starts operation and the Y-axis repeats the cam operation. At this time, the X-axis is stopped and only the Y-axis is operated.



Virtual drive axis and virtual linked axis

This program uses the virtual drive axis (VirtualAxis0001) and the virtual linked axes (LinkAxis0001, LinkAxis0002) in addition to the real drive axes (Axis0001, Axis0002).

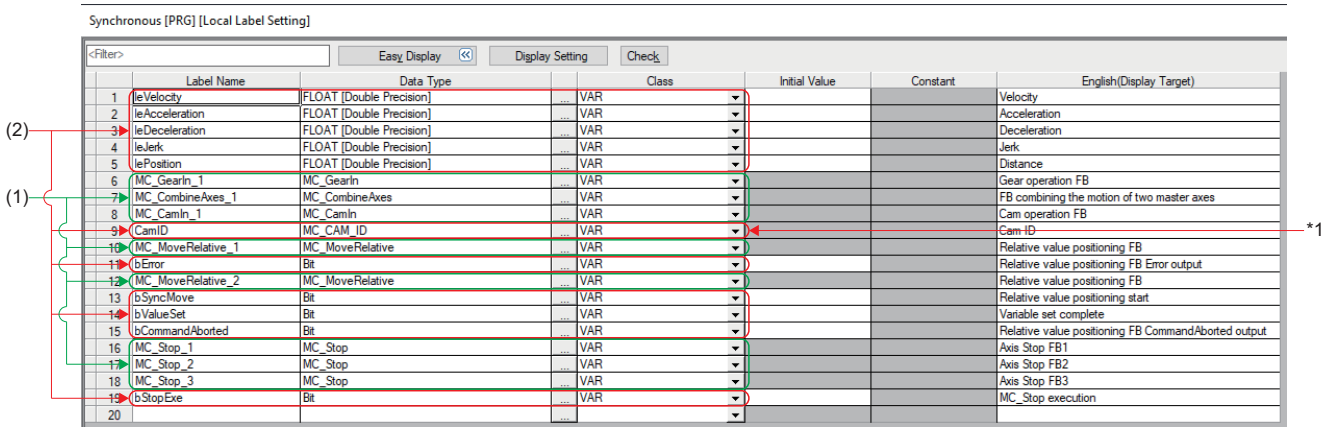
The AxisRef type structure is used in the PLC CPU program that drives the virtual drive axis and the virtual linked axes.

Set each of the AxisRef structures as public labels.

For the public label setting, refer to the following.

☞ Page 82 Motion module

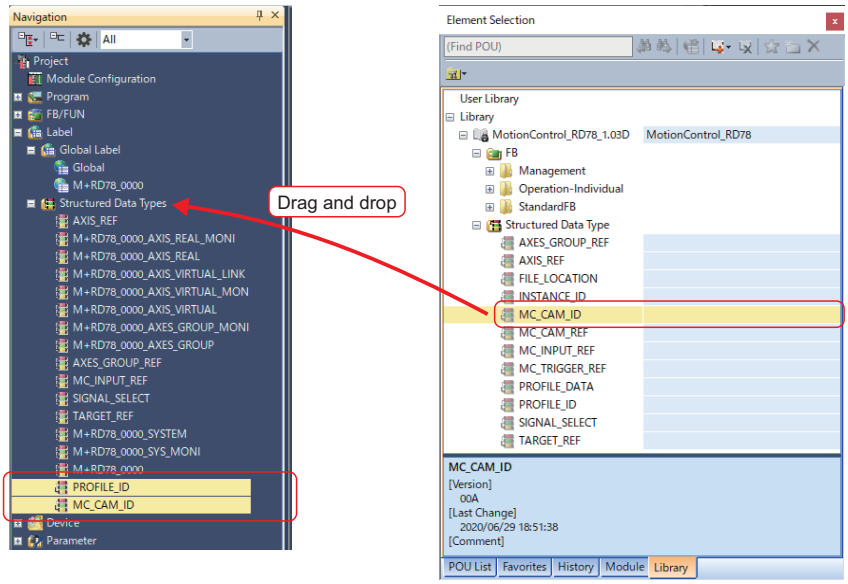
Local labels



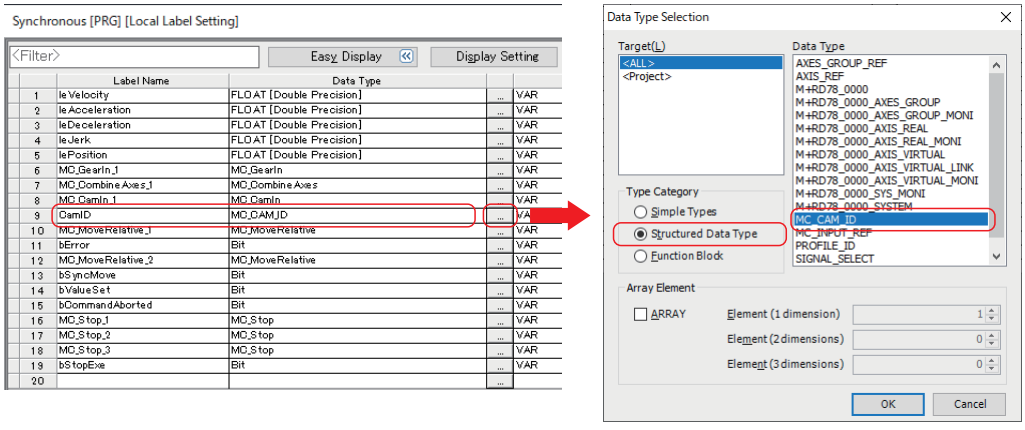
No.	Description
(1)	These labels are automatically added when the user drags and drops the FBs (MC_GearIn, MC_CombineAxes, MC_CamIn, MC_MoveRelative, and MC_Stop) onto the program editor.
(2)	These labels are registered manually.

*1 The method for registering the MC_CAM_ID structure is shown below.

1. Click the [Library] tab in the element selection window, then select "Library"⇨"MotionControl_RD78_****"*2"⇨"Structured Data Type" to display the list of structured data types.
2. Select "MC_CAM_ID", then drag and drop it onto "Label"⇨"Structured Data Types" in the navigation window.
3. "MC_CAM_ID" is registered in the "Structured Data Types" tree and becomes available as a data type on the label editor.



4. On the "Local Label Setting" screen, click the "..." button next to the label (setting example: CamID) to set MC_CAM_ID data structure for.
5. Selecting "Structured Data Type" as the type category on the "Data Type Setting" screen will display "MC_CAM_ID" as the data type. Select "MC_CAM_ID", then click the [OK] button. The MC_CAM_ID data type structure can now be used.



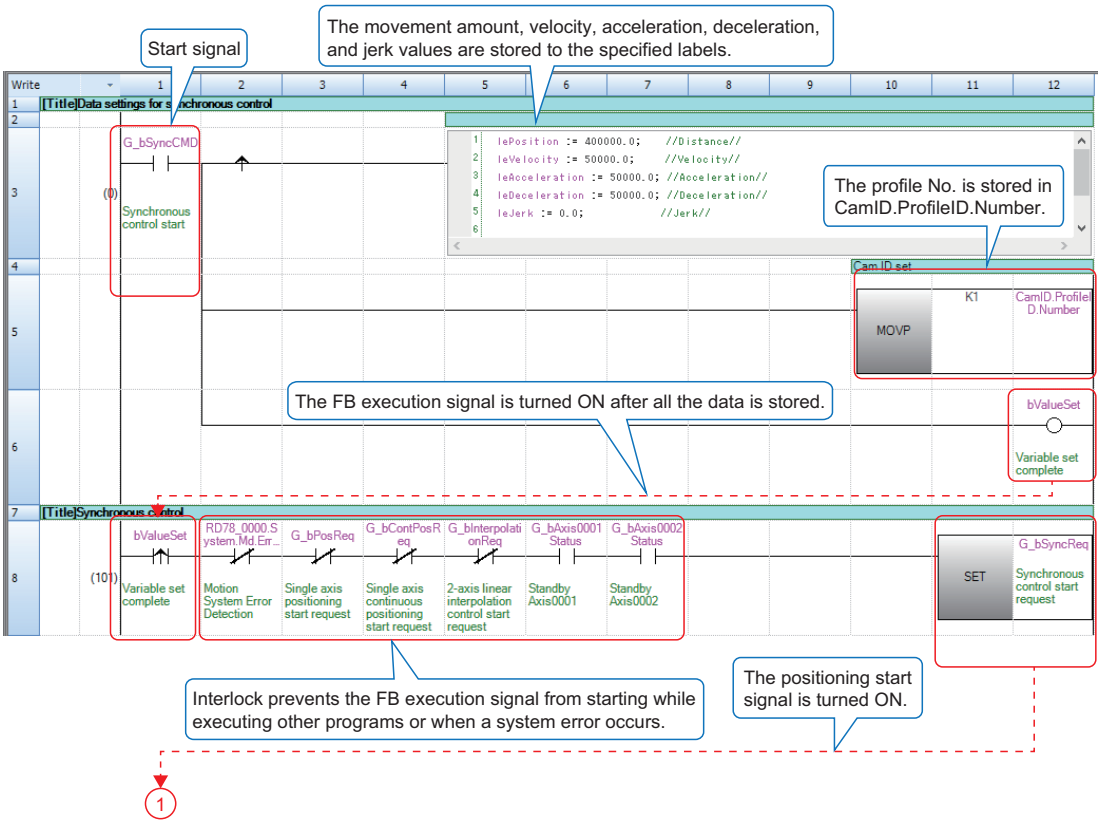
*2 **** = Motion module FB library version

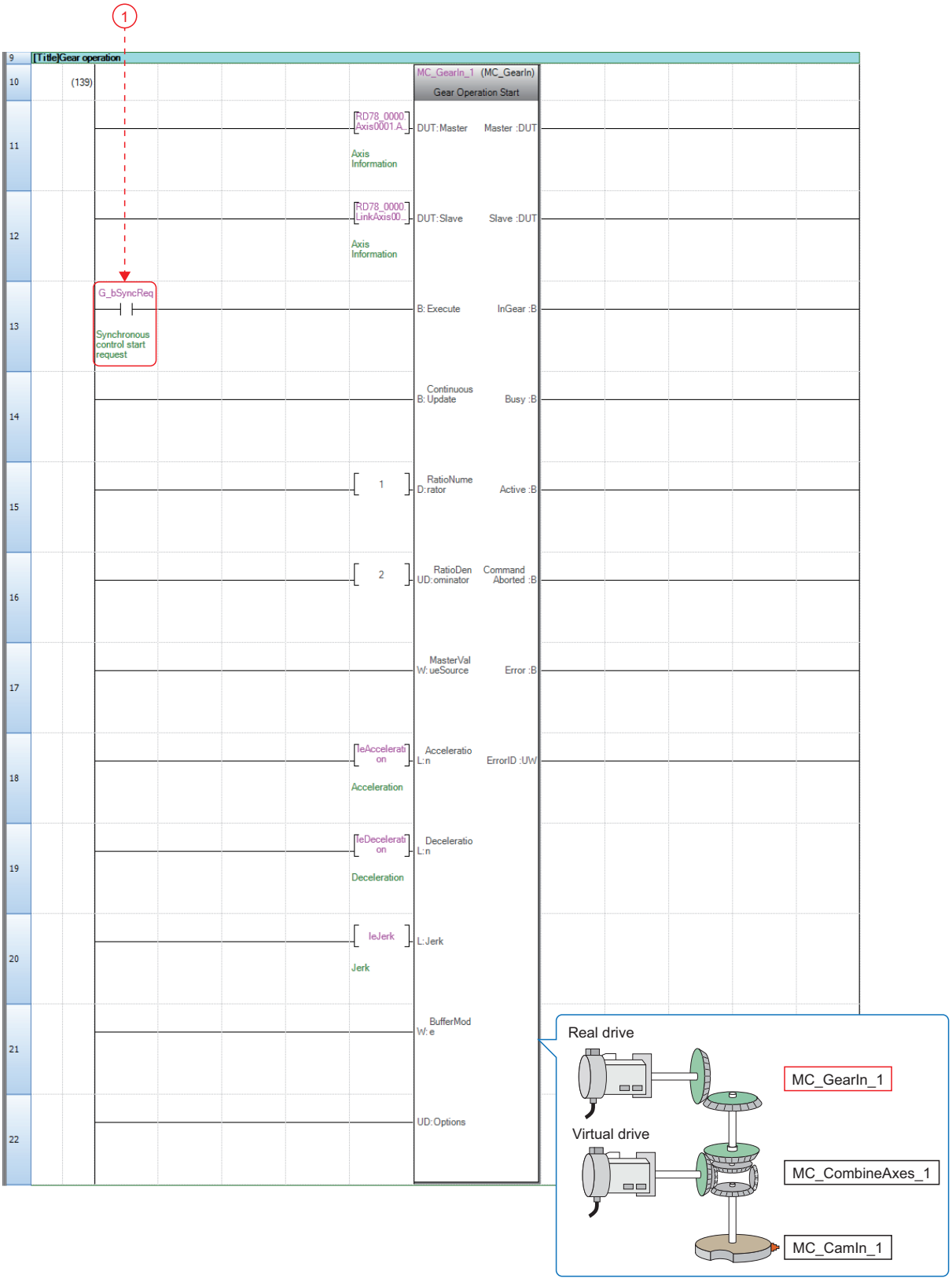
Program example

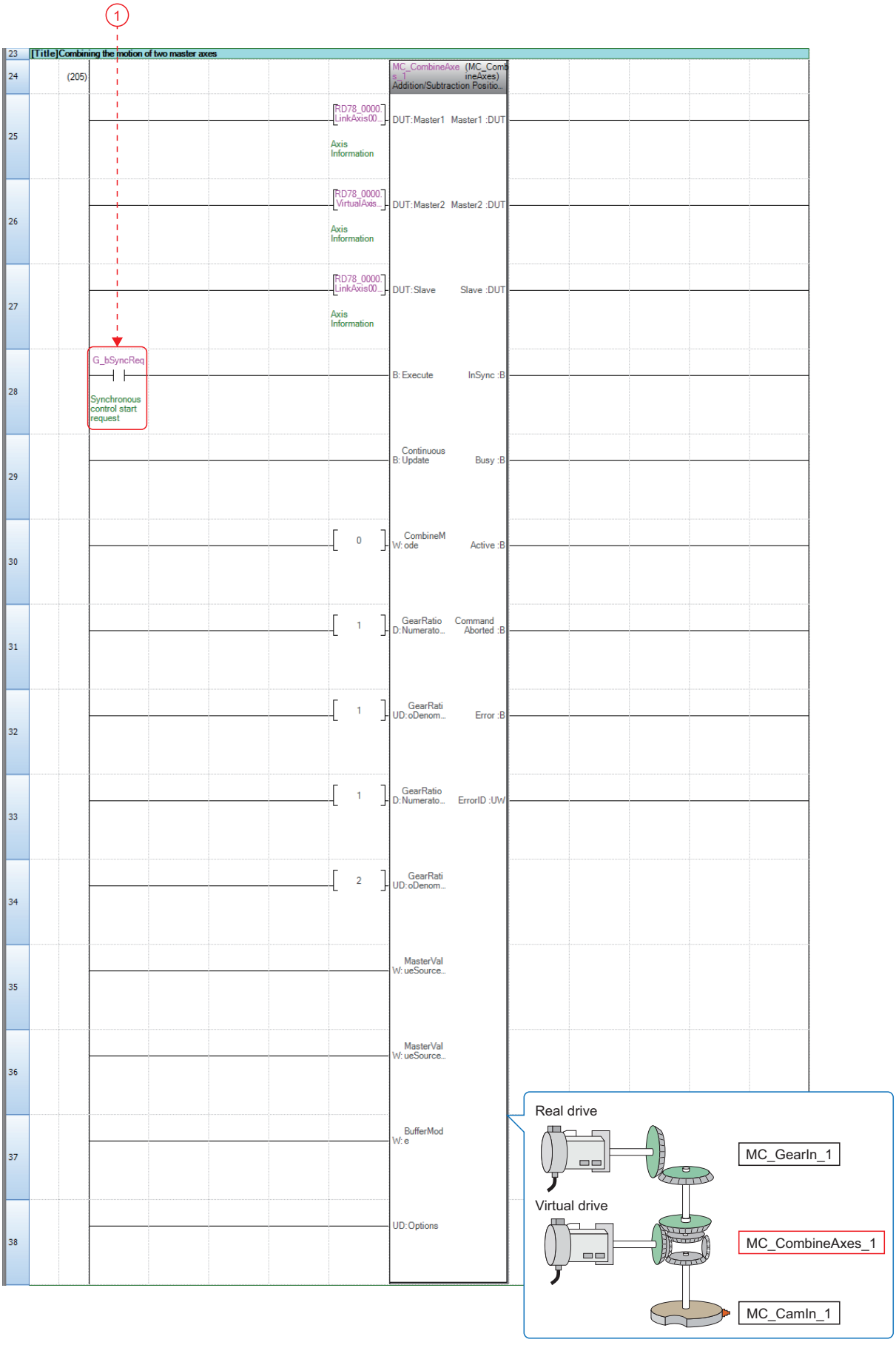
When the synchronous control start is turned ON, the positioning data is stored to each specified label.
When all the data is stored, the execution signal of the FB (MC_GearIn, MC_CombineAxes, and MC_CamIn) is kept ON by a self-holding circuit.
When the axis status of Axis0002 is turned to "7: During synchronous operation (SynchronizedMotion)", Axis0001 (Master axis) starts positioning. At this time, Axis0002 is operated according to the specified operation profile. When positioning of Axis0001 is completed, the virtual drive axis (VirtualAxis0001) starts positioning.
At this time, Axis0001 is stopped, but Axis0002 is operated according to the specified operationc profile.
The synchronous control is stopped by executing the FB (MC_Stop) on the real drive axis (Axis0002) and the virtual link axes (LinkAxis0001 and LinkAxis0002) when any of the following conditions are satisfied: the positioning of the virtual drive axis (VirtualAxis0001) is completed; an error occurs; or the execution is aborted. The start signal is reset when the real drive axis (Axis0002) status is out of synchronization.
The interlock conditions are added to prevent execution of the synchronous control while other programs are running or when a system error occurs.

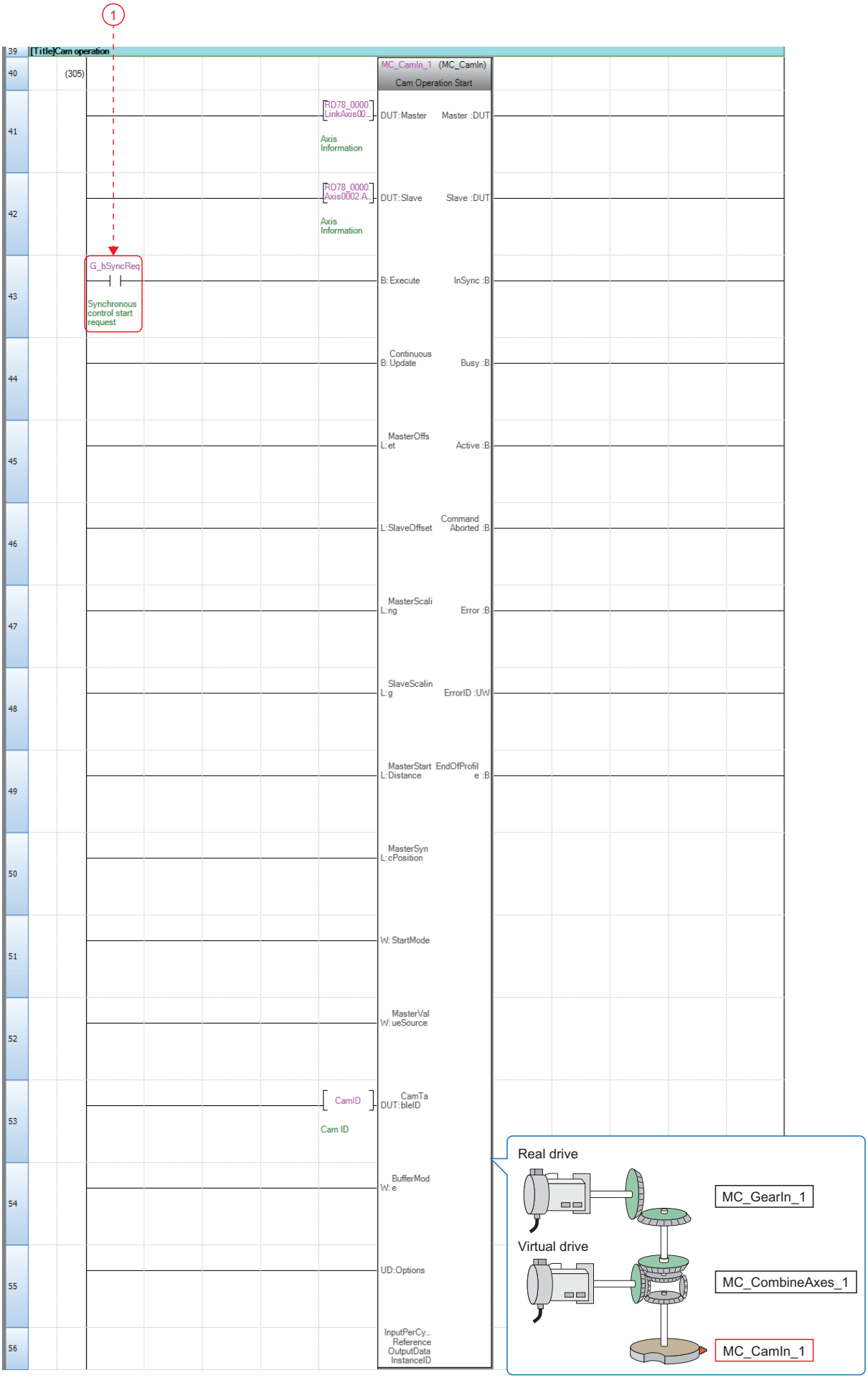
- Start signal

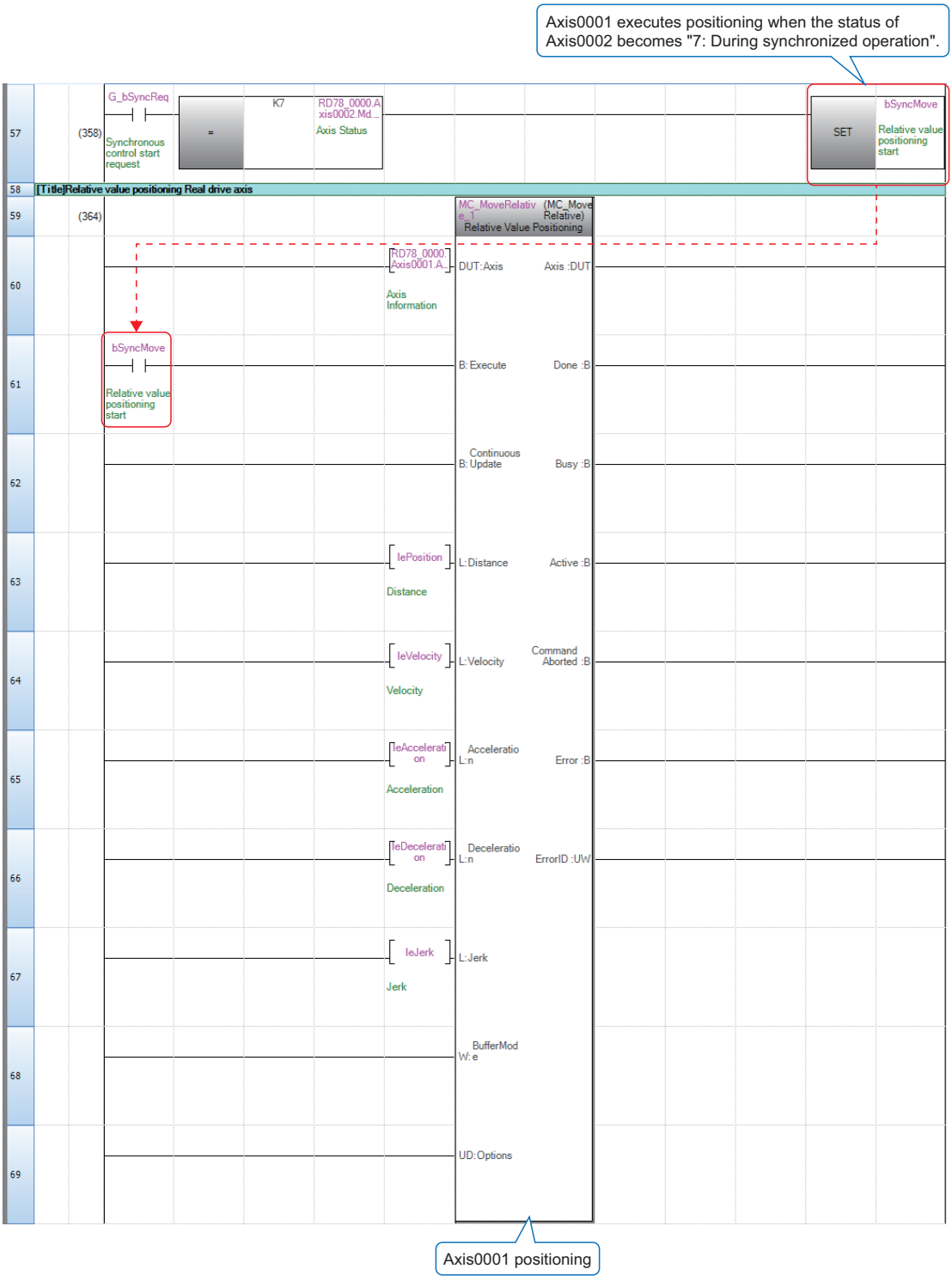
Signal name	Label name
Synchronous control start	G_bSyncCMD

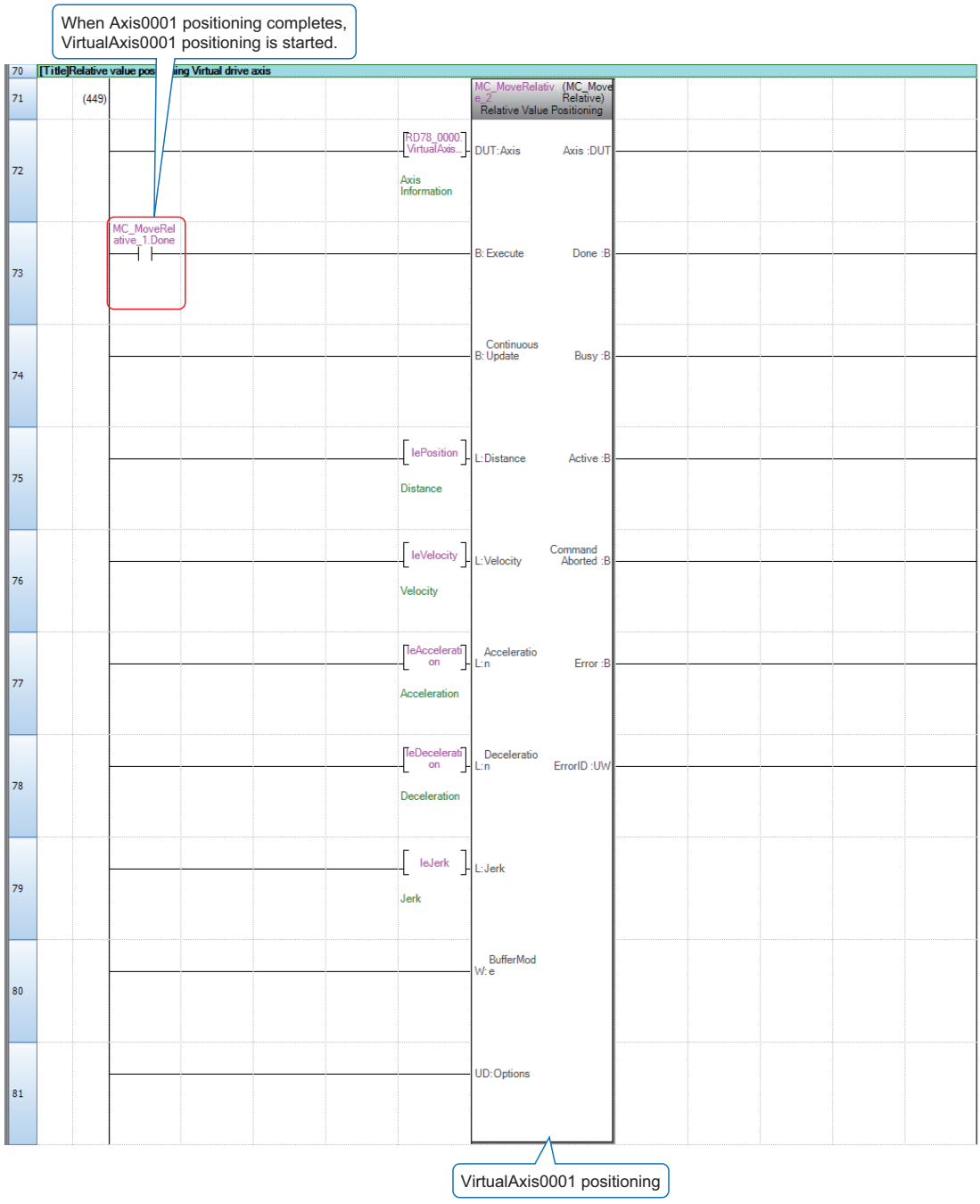


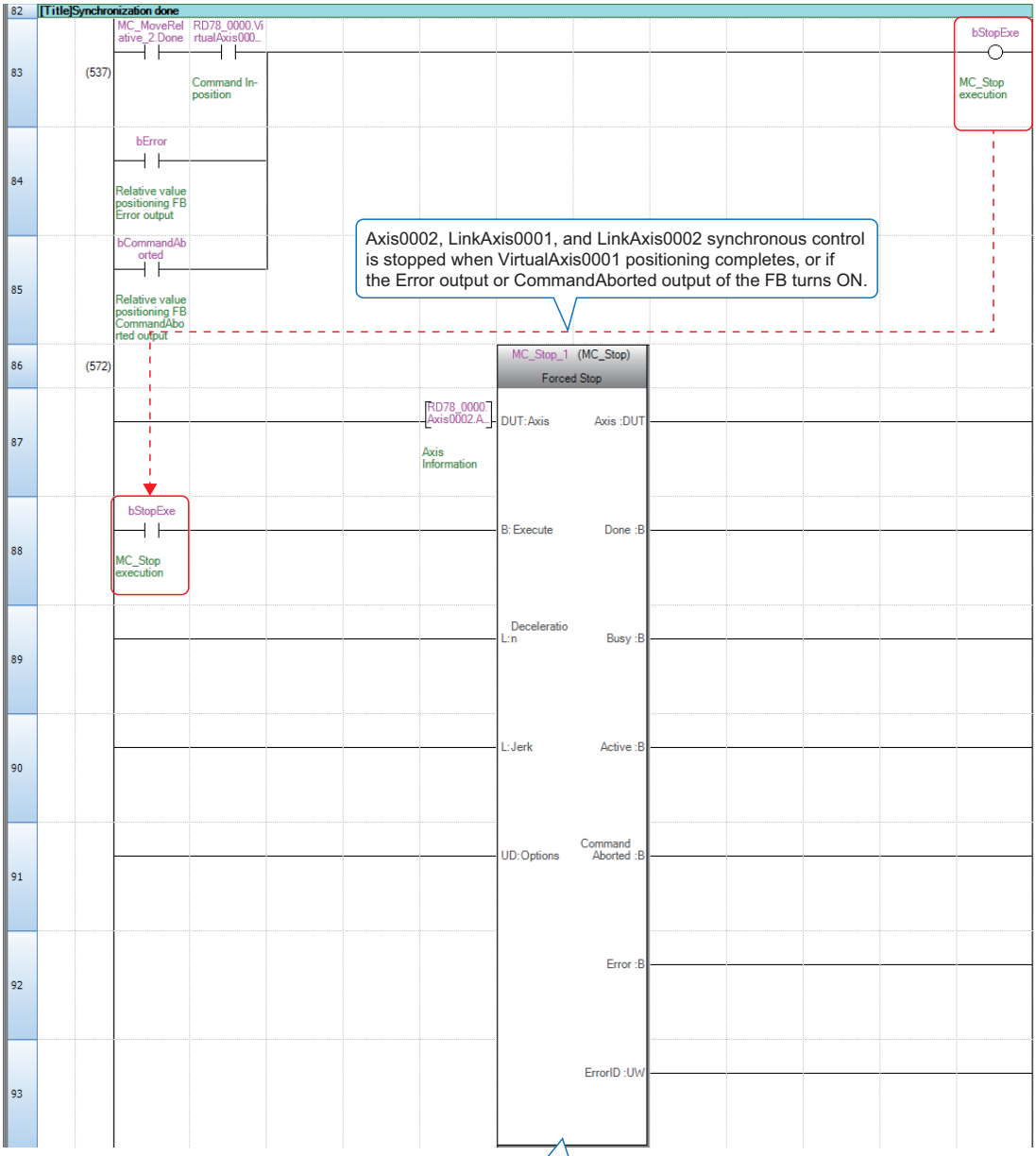

1
9
[Title]Gear operation
10
(139)
MC_GearIn_1 (MC_GearIn)
Gear Operation Start
11
RD78_0000
Axis0001.A_
DUT:Master
Master :DUT
Axis
Information
12
RD78_0000
LinkAxis00_
DUT:Slave
Slave :DUT
Axis
Information
13
G_bSyncReq
B:Execute
InGear :B
Synchronous
control start
request
14
Continuous
B:Update
Busy :B
15
1
RatioNume
D:rator
Active :B
16
2
RatioDen
UD:ominator
Command
Aborted :B
17
MasterVal
W:ueSource
Error :B
18
leAccelerati
on
Acceleratio
L:n
ErrorID :UW
Acceleration
19
leDeceleratl
on
Deceleratio
L:n
Deceleration
20
leJerk
L:Jerk
Jerk
21
BufferMod
W:e
22
UD:Options
Real drive
MC_GearIn_1
Virtual drive
MC_CombineAxes_1
MC_CamIn_1

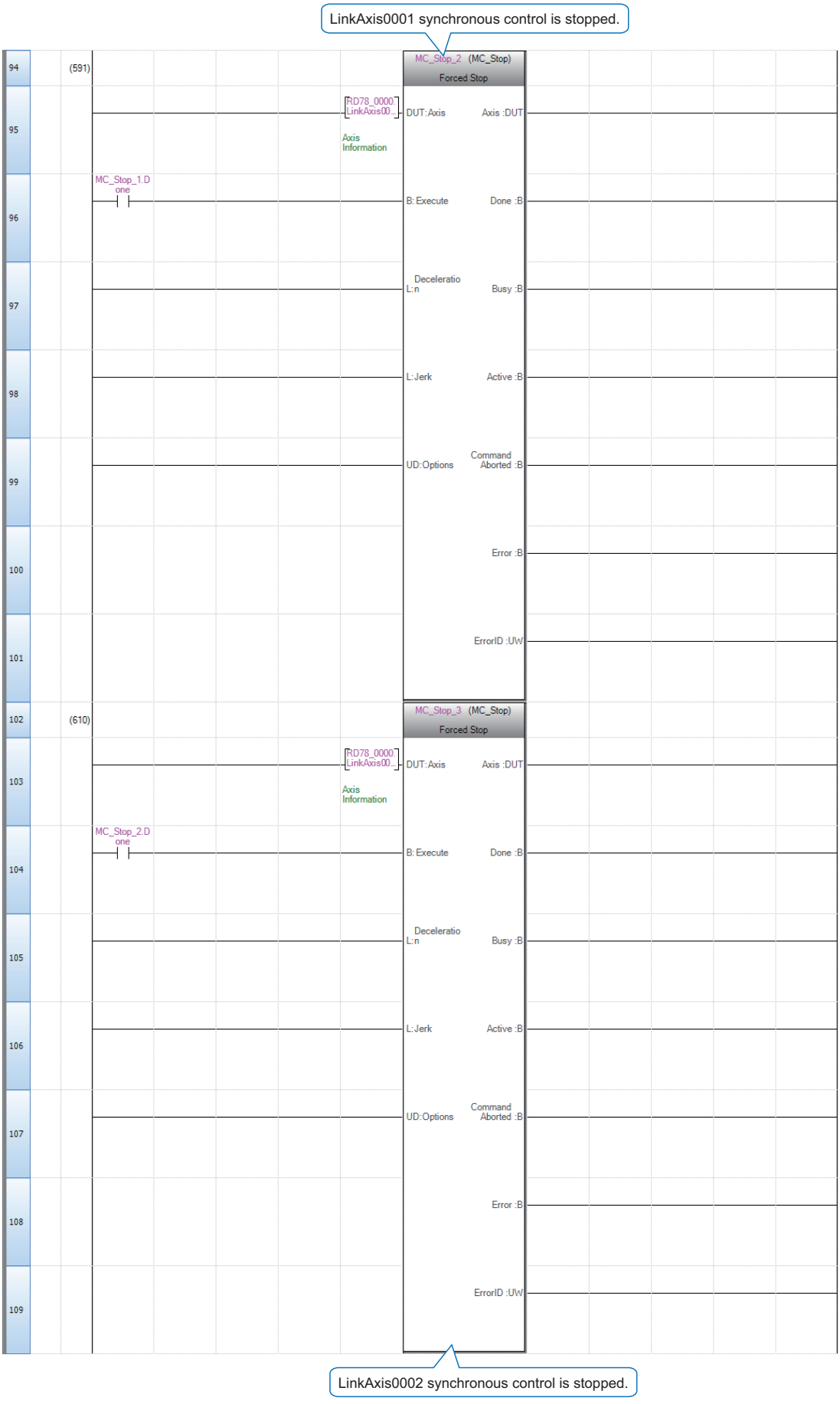

1
23
[Title]Combining the motion of two master axes
24
(205)
MC_CombineAxes_1
(MC_CombineAxes)
Addition/Subtraction Positio...
25
RD78_0000
LinkAxis00
Axis Information
DUT:Master1
Master1 :DUT
26
RD78_0000
VirtualAxis...
Axis Information
DUT:Master2
Master2 :DUT
27
RD78_0000
LinkAxis00
Axis Information
DUT:Slave
Slave :DUT
28
G_bSyncReq
Synchronous control start request
B:Execute
InSync :B
29
Continuous
B:Update
Busy :B
30
0
CombineMW:ode
Active :B
31
1
GearRatioD:Numerato...
Command Aborted :B
32
1
GearRatiUD:oDenom...
Error :B
33
1
GearRatioD:Numerato...
ErrorID :UW
34
2
GearRatiUD:oDenom...
35
MasterValW:ueSource...
36
MasterValW:ueSource...
37
BufferModW:e
38
UD:Options
Real drive
Virtual drive
MC_GearIn_1
MC_CombineAxes_1
MC_CamIn_1

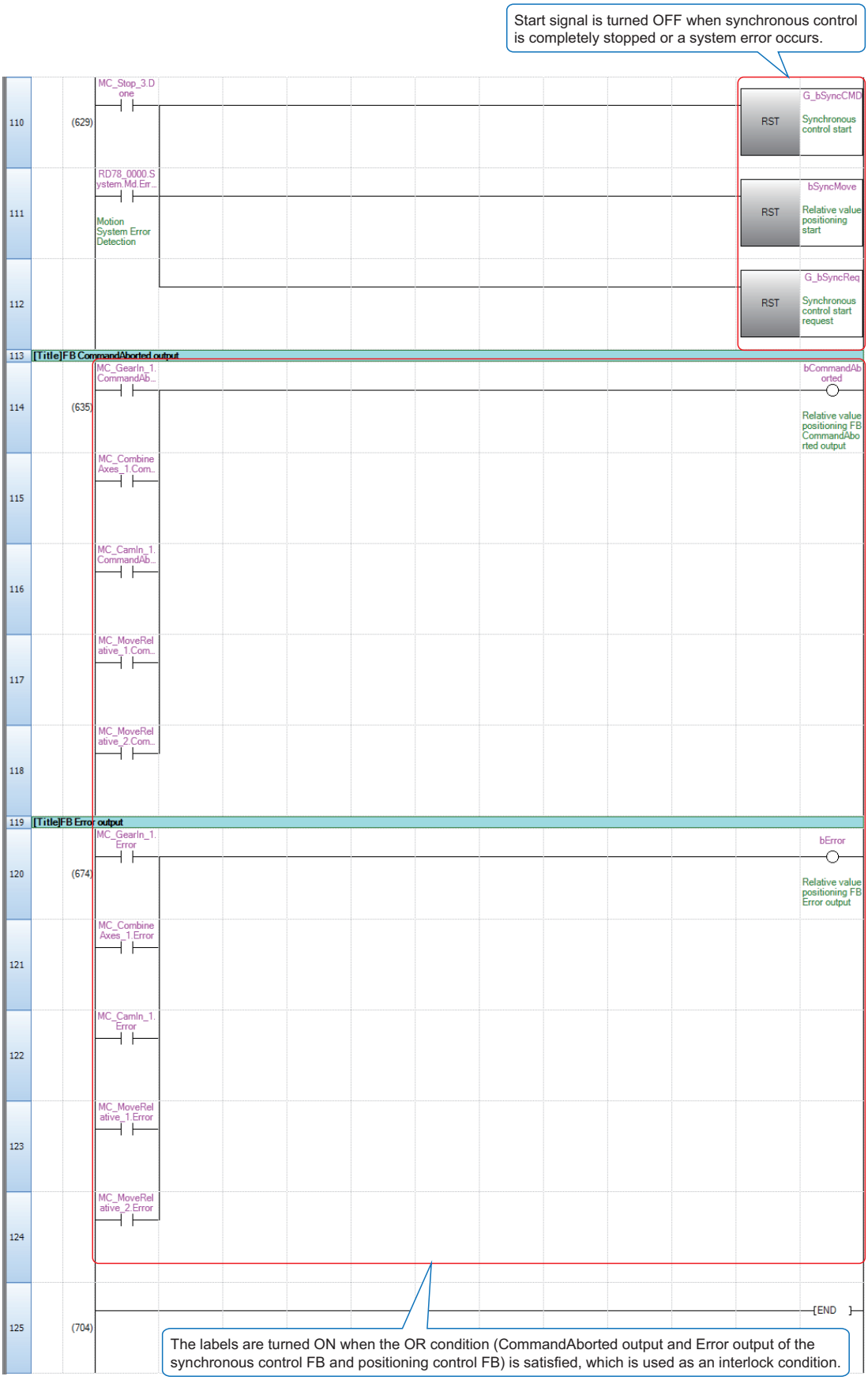

1
39 [Title]Cam operation
40 (305)
MC_CamIn_1 (MC_CamIn)
Cam Operation Start
41 RD78_0000 LinkAxis00
Axis Information
DUT:Master
Master :DUT
42 RD78_0000 Axis0002.A
Axis Information
DUT:Slave
Slave :DUT
43 G_bSyncReq
Synchronous control start request
B:Execute
InSync :B
44 Continuous B:Update
Busy :B
45 MasterOffs L:et
Active :B
46 L:SlaveOffset
Command Aborted :B
47 MasterScali L:ng
Error :B
48 SlaveScalin L:g
ErrorID :UW
49 MasterStart L:Distance
EndOfProfil e :B
50 MasterSyn L:cPosition
51 W:StartMode
52 MasterVal W:ueSource
53 CamID
Cam ID
CamTa DUT:bleID
54 BufferMod W:e
55 UD:Options
56 InputPerCy...
Reference
OutputData
InstanceID
Real drive
MC_GearIn_1
Virtual drive
MC_CombineAxes_1
MC_CamIn_1


Axis0001 executes positioning when the status of Axis0002 becomes "7: During synchronized operation".
57
(358)
G_bSyncReq
Synchronous control start request
=
K7
RD78_0000.Axis0002.Md...
Axis Status
SET
bSyncMove
Relative value positioning start
58
[Title]Relative value positioning Real drive axis
59
(364)
MC_MoveRelative_1
(MC_MoveRelative)
Relative Value Positioning
60
RD78_0000.Axis0001.A...
Axis Information
DUT:Axis
Axis :DUT
61
bSyncMove
Relative value positioning start
B:Execute
Done :B
62
Continuous B:Update
Busy :B
63
lePosition
Distance
L:Distance
Active :B
64
leVelocity
Velocity
L:Velocity
Command Aborted :B
65
leAcceleration
Acceleration
L:n
Error :B
66
leDeceleration
Deceleration
L:n
ErrorID :UW
67
leJerk
Jerk
L:Jerk
68
BufferMode
W:e
69
UD:Options
Axis0001 positioning


When Axis0001 positioning completes, VirtualAxis0001 positioning is started.
70
[Title]Relative value pos ing Virtual drive axis
71
(449)
MC_MoveRelativ e_2
(MC_Move Relative)
Relative Value Positioning
72
RD78_0000 VirtualAxis_
DUT: Axis
Axis :DUT
Axis Information
73
MC_MoveRel ative_1.Done
B: Execute
Done :B
74
Continuous B: Update
Busy :B
75
lePosition
L: Distance
Active :B
Distance
76
leVelocity
L: Velocity
Command Aborted :B
Velocity
77
leAccelerati on
Acceleratio n: L
Error :B
Acceleration
78
leDeceleratio n
Deceleratio n: L
ErrorID :UW
Deceleration
79
leJerk
L: Jerk
Jerk
80
BufferMod e: W
81
UD: Options
VirtualAxis0001 positioning


82
[Title]Synchronization done
MC_MoveRelative_2.Done
RD78_0000.VirtualAxis000...
bStopExe
83
(537)
Command In-position
MC_Stop execution
bError
84
Relative value positioning FB Error output
bCommandAborted
85
Relative value positioning FB CommandAborted output
Axis0002, LinkAxis0001, and LinkAxis0002 synchronous control is stopped when VirtualAxis0001 positioning completes, or if the Error output or CommandAborted output of the FB turns ON.
86
(572)
MC_Stop_1 (MC_Stop)
Forced Stop
87
RD78_0000 Axis0002.A...
Axis Information
DUT:Axis
Axis :DUT
bStopExe
88
MC_Stop execution
B:Execute
Done :B
Deceleratio
L:n
Busy :B
89
L:Jerk
Active :B
90
UD:Options
Command Aborted :B
91
Error :B
92
ErrorID :UW
93
Axis0002 synchronous control is stopped.


LinkAxis0001 synchronous control is stopped.
94
(591)
MC_Stop_2 (MC_Stop)
Forced Stop
RD78_0000
LinkAxis00
95
Axis
Information
DUT:Axis
Axis :DUT
MC_Stop_1.D
one
96
B:Execute
Done :B
Deceleratio
L:n
97
Busy :B
98
L:Jerk
Active :B
99
UD:Options
Command
Aborted :B
100
Error :B
101
ErrorID :UW
102
(610)
MC_Stop_3 (MC_Stop)
Forced Stop
RD78_0000
LinkAxis00
103
Axis
Information
DUT:Axis
Axis :DUT
MC_Stop_2.D
one
104
B:Execute
Done :B
Deceleratio
L:n
105
Busy :B
106
L:Jerk
Active :B
107
UD:Options
Command
Aborted :B
108
Error :B
109
ErrorID :UW
LinkAxis0002 synchronous control is stopped.


Start signal is turned OFF when synchronous control is completely stopped or a system error occurs.
MC_Stop_3.Done
(629)
RST
G_bSyncCMD
Synchronous control start
RD78_0000.System.Md.Err...
Motion System Error Detection
RST
bSyncMove
Relative value positioning start
RST
G_bSyncReq
Synchronous control start request
[Title]FB CommandAborted output
MC_GearIn_1.CommandAb...
(635)
bCommandAborted
Relative value positioning FB CommandAborted output
MC_CombineAxes_1.Com...
MC_CamIn_1.CommandAb...
MC_MoveRelative_1.Com...
MC_MoveRelative_2.Com...
[Title]FB Error output
MC_GearIn_1.Error
(674)
bError
Relative value positioning FB Error output
MC_CombineAxes_1.Error
MC_CamIn_1.Error
MC_MoveRelative_1.Error
MC_MoveRelative_2.Error
(704)
[END]
The labels are turned ON when the OR condition (CommandAborted output and Error output of the synchronous control FB and positioning control FB) is satisfied, which is used as an interlock condition.

4.12 Error Reset (Program Name: ErrorReset)

This program resets the error on each axis.

FBs used

Type	FB	Description
Management	MC_Reset	Resets errors and warnings of the axis.
	MC_GroupReset	Resets errors and warnings of the axes group and each axis in the axes group.
	MCv_MotionErrorReset	Reset all errors and warnings of the Motion system.

Local labels

(1)

Error_Reset [PRG] [Local Label Setting]

<Filter> | Easy Display | Display Setting | Check

	Label Name	Data Type		Class	Initial Value	Constant	English(Display Target)
1	MC_Reset_1	MC_Reset	...	VAR			Axis error reset FB1 Axis0001
2	MC_Reset_2	MC_Reset	...	VAR			Axis error reset FB2 Axis0002
3	MC_Reset_3	MC_Reset	...	VAR			Axis error reset FB3 VirtualAxis0001
4	MC_Reset_4	MC_Reset	...	VAR			Axis error reset FB4 LinkAxis0001
5	MC_Reset_5	MC_Reset	...	VAR			Axis error reset FB5 LinkAxis0002
6	MC_GroupReset_1	MC_GroupReset	...	VAR			Axes group error reset FB
7	MCv_MotionErrorReset_1	MCv_MotionErrorReset	...	VAR			System error reset FB
8			...				

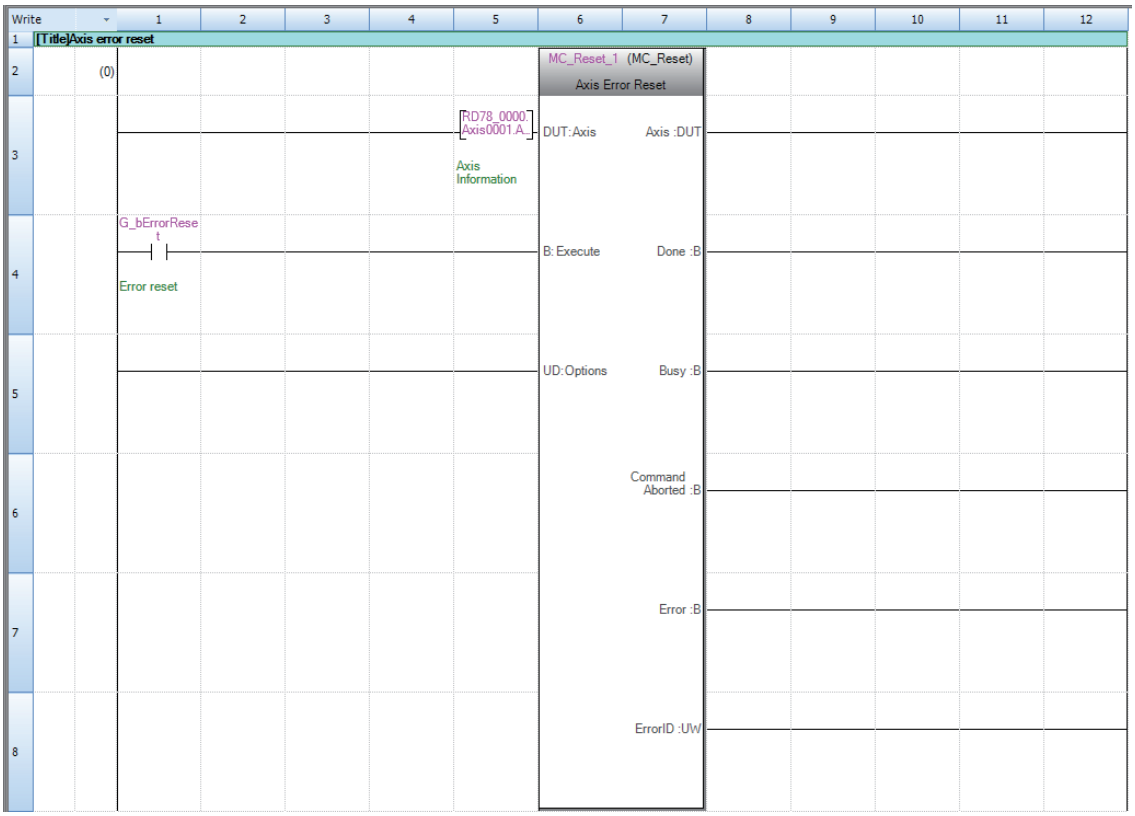
No.	Description
(1)	These labels are automatically added when the user drags and drops the FB (MC_MoveRelative) onto the program editor.

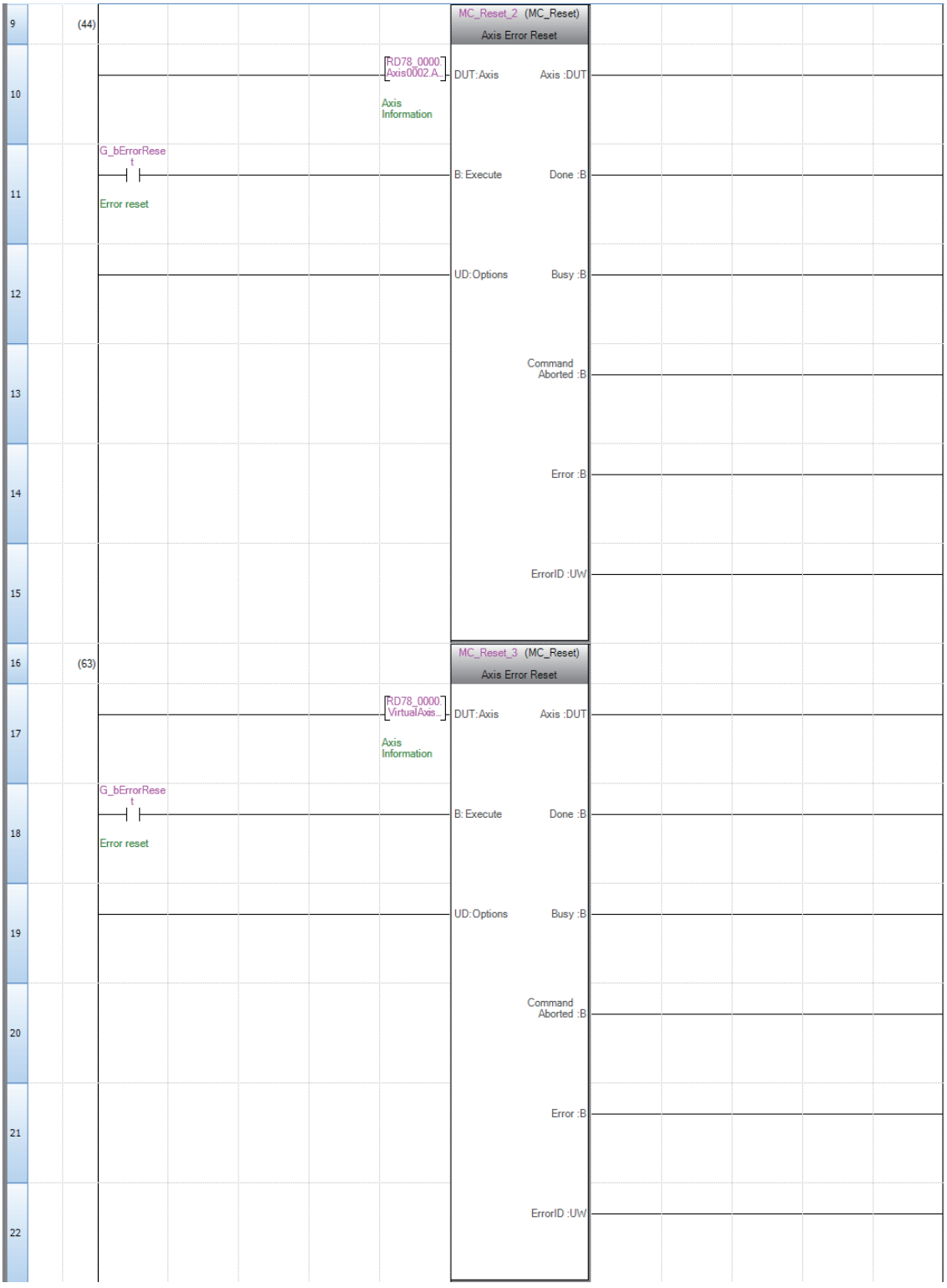
Program example

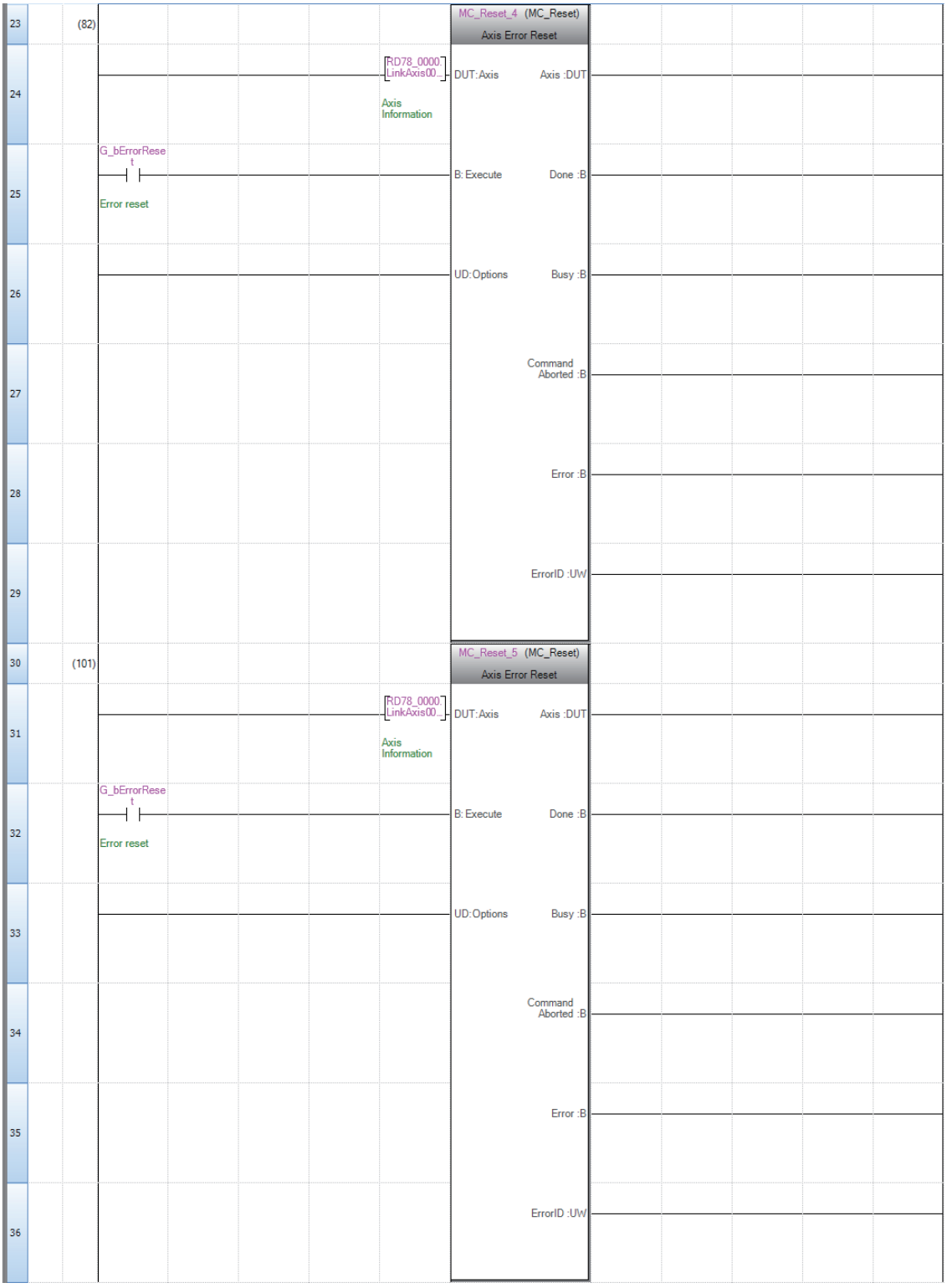
The following shows the program example for error reset.

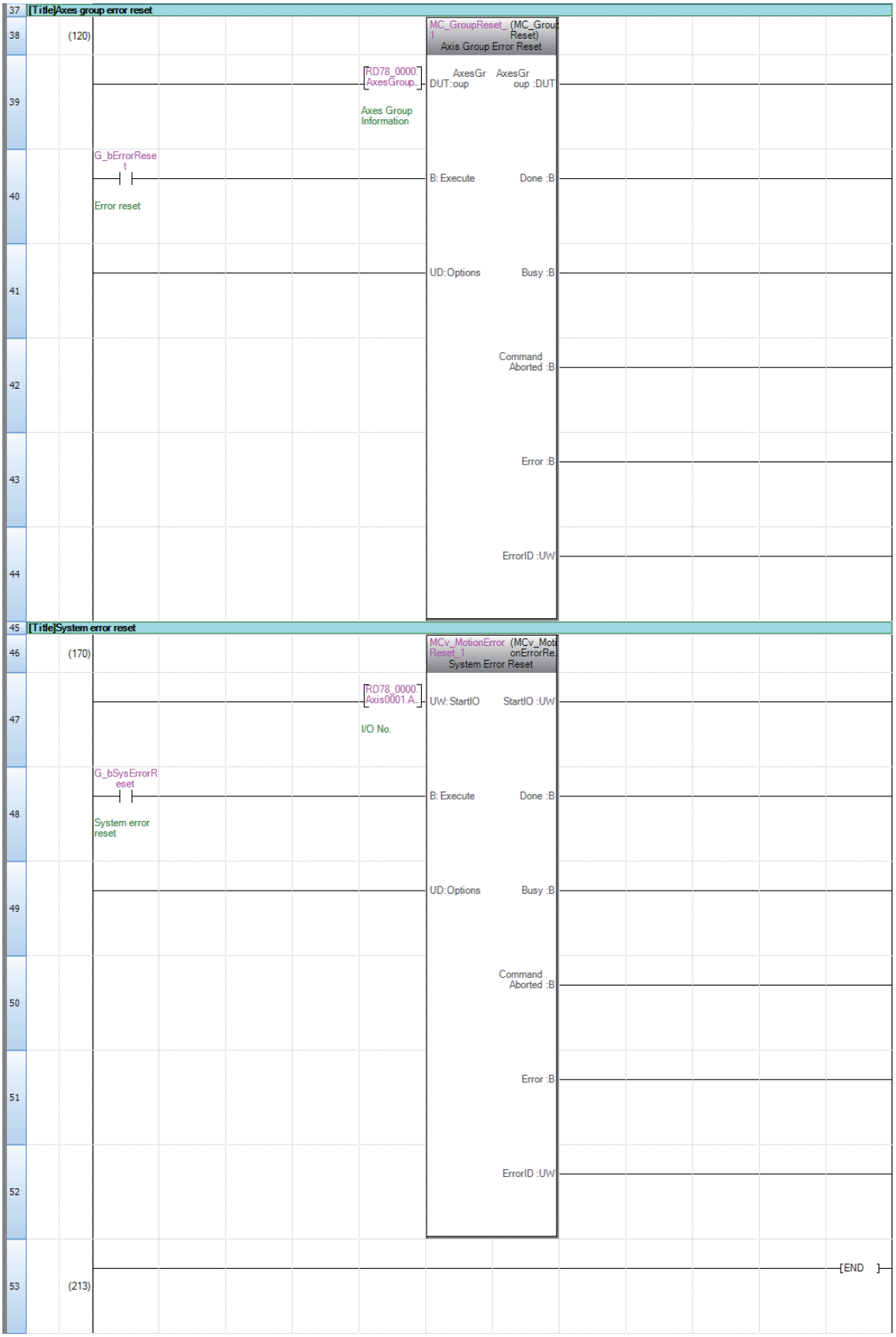
When the error reset label is turned ON, MC_Reset and MC_GroupReset are executed.

When the system error reset label is turned ON, MCv_MotionErrorReset is executed.




9
(44)
MC_Reset_2 (MC_Reset)
Axis Error Reset
10
RD78_0000
Axis0002.A...
DUT:Axis
Axis :DUT
Axis
Information
G_bErrorRese
t
11
B:Execute
Done :B
Error reset
12
UD:Options
Busy :B
13
Command
Aborted :B
14
Error :B
15
ErrorID :UW
16
(63)
MC_Reset_3 (MC_Reset)
Axis Error Reset
17
RD78_0000
VirtualAxis...
DUT:Axis
Axis :DUT
Axis
Information
G_bErrorRese
t
18
B:Execute
Done :B
Error reset
19
UD:Options
Busy :B
20
Command
Aborted :B
21
Error :B
22
ErrorID :UW


23
(82)
MC_Reset_4 (MC_Reset)
Axis Error Reset
RD78_0000
LinkAxis00..
DUT:Axis
Axis :DUT
24
Axis
Information
G_bErrorRese
t
25
B:Execute
Done :B
Error reset
26
UD:Options
Busy :B
27
Command
Aborted :B
28
Error :B
29
ErrorID :UW
30
(101)
MC_Reset_5 (MC_Reset)
Axis Error Reset
RD78_0000
LinkAxis00..
DUT:Axis
Axis :DUT
31
Axis
Information
G_bErrorRese
t
32
B:Execute
Done :B
Error reset
33
UD:Options
Busy :B
34
Command
Aborted :B
35
Error :B
36
ErrorID :UW

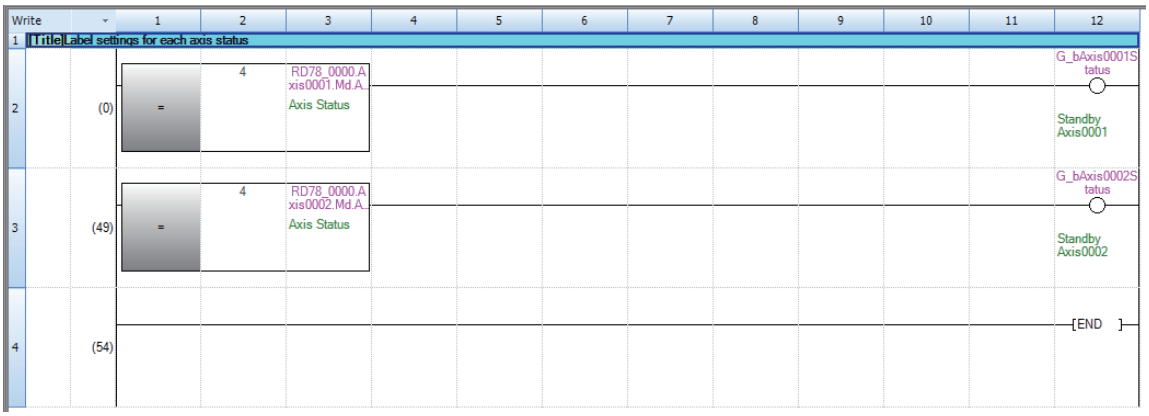

37 [Title]Axes group error reset
38 (120)
MC_GroupReset_1 (MC_GroupReset)
Axis Group Error Reset
39
RD78_0000 AxesGroup...
Axes Group Information
AxesGr DUT:oup
AxesGr oup :DUT
40
G_bErrorReset
Error reset
B: Execute
Done :B
41
UD: Options
Busy :B
42
Command Aborted :B
43
Error :B
44
ErrorID :UW
45 [Title]System error reset
46 (170)
MCv_MotionErrorReset_1 (MCv_MotionErrorRe...
System Error Reset
47
RD78_0000 Axis0001.A...
I/O No.
UW: StartIO
StartIO :UW
48
G_bSysErrorReset
System error reset
B: Execute
Done :B
49
UD: Options
Busy :B
50
Command Aborted :B
51
Error :B
52
ErrorID :UW
53 (213)
[END]

4.13 Axis Status Label Setting (Program Name: AxisStatus_labelset)

Creates bit-type labels that turn ON when the axis status (AxisStatus) of the real drive axes (Axis0001 and Axis0002) transits to "4: Standby (Standstill)". These labels are then used as a start condition interlock for each program.

Program example

After the axis status (AxisStatus) of the real drive axes (Axis0001 and Axis0002) transits to "4: Standby (Standstill)", the standby signal turns ON.

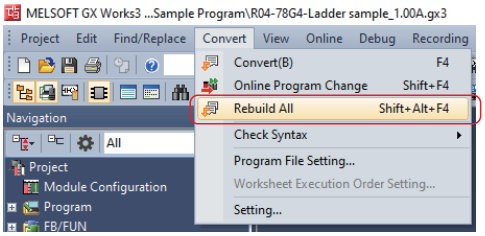


4.14 Checking Operation

Conversion and writing of programs

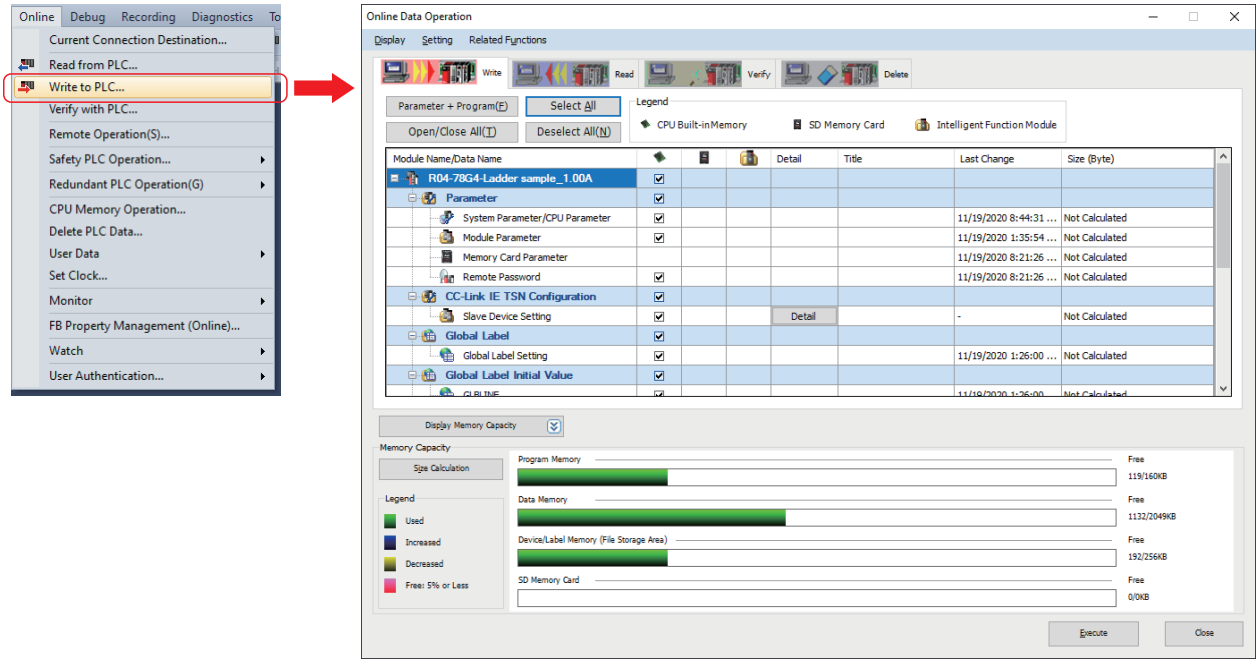
Write the program to the PLC CPU.

1. After creating the programs, select [Convert]⇨[Rebuild All] in the menu to convert all the programs.



2. Confirm that no error occurs after executing "Rebuild All". Select [Online]⇨[Write to PLC] in the menu, then write the programs to the PLC CPU.

When the parameters and the public labels of the Motion module have been changed, write the programs after first converting all the programs and reflecting the public labels in Motion Control Setting Function.



Axis monitor

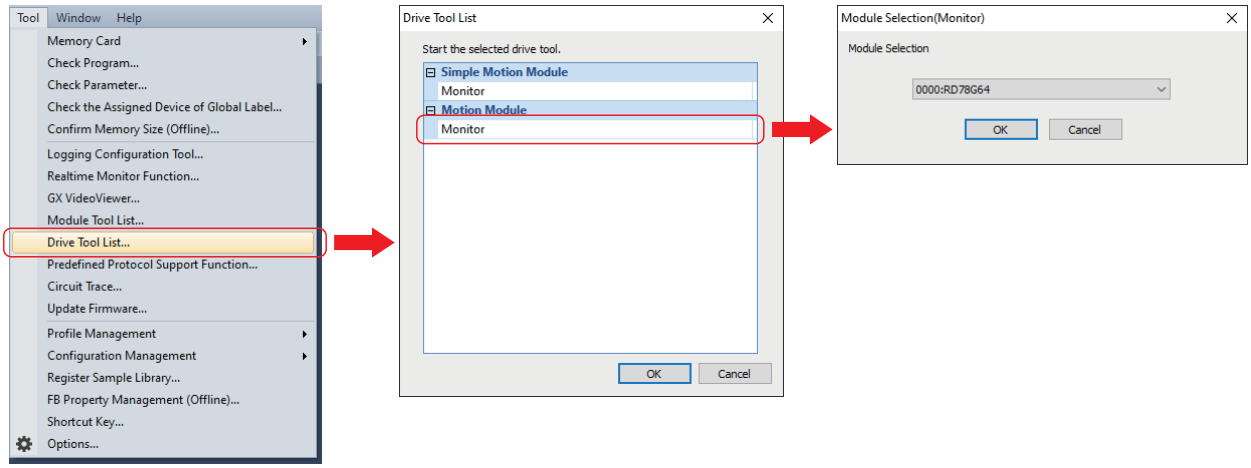
The axis monitor screen displays the current values and the error codes of all axes in operation all at once.
Users can check the current values and whether any error occurs during the operation.

Displaying the axis monitor screen

The axis monitor screen can be displayed by using the following methods.

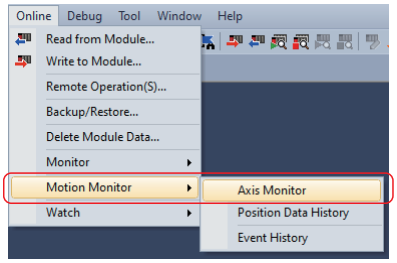
■Displaying the axis monitor screen from MELSOFT GX Works3

1. Select [Tool]⇨[Drive Tool List] from the menu in MELSOFT GX Works3 to display the "Drive Tool List" screen.
2. Double-click [Monitor] in "Motion Module" to display the "Module Selection(Monitor)" screen. Select the Motion module (setting example: 0000:RD78G64), then click the [OK] button.



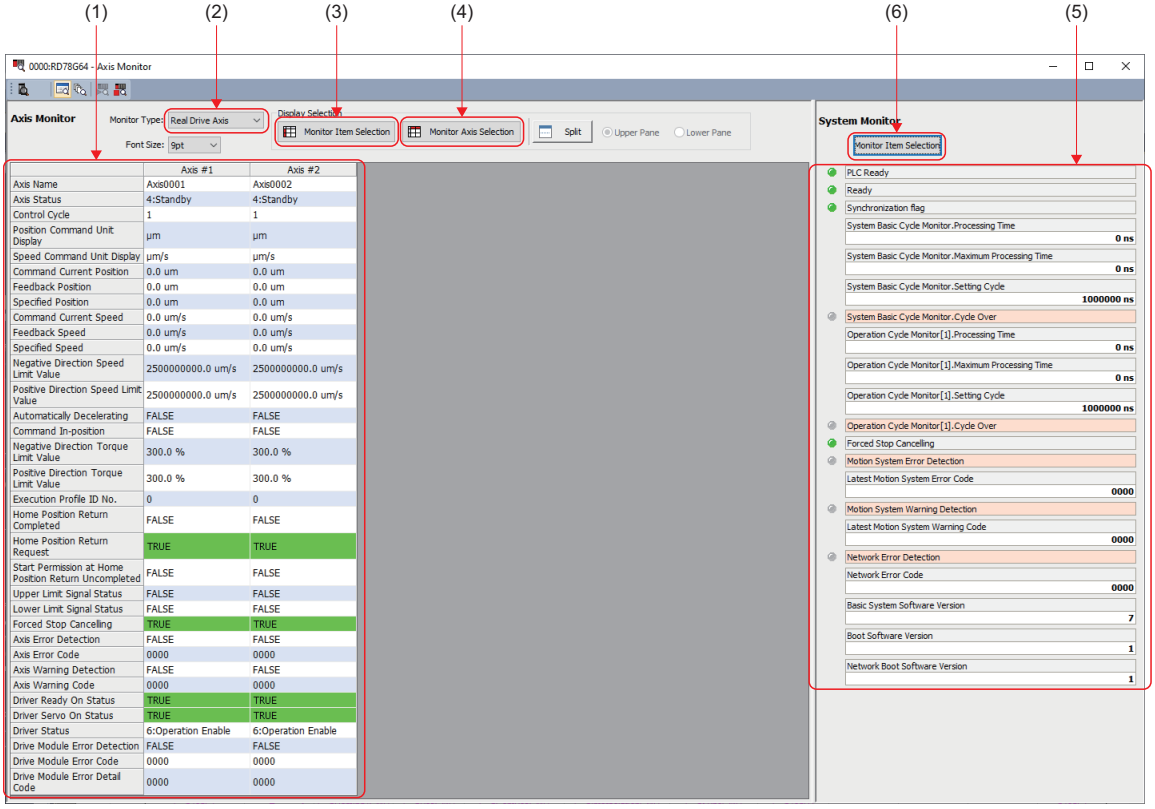
■Displaying the axis monitor screen from Motion Control Setting Function

1. Select [Online]⇨[Motion Monitor]⇨[Axis Monitor] from the menu in Motion Control Setting Function.



Axis monitor screen

The following contents are displayed on the axis monitor.



No.	Name	Description
(1)	Axis monitor display items	The monitor items for each axis selected in monitor type are displayed.
(2)	Monitor type	The axis type to monitor can be selected.
(3)	Monitor item selection	Add/delete items to monitor in the axis monitor display items.
(4)	Monitor axis selection	Add/delete axes to monitor in the axis monitor display items.
(5)	System monitor display items	The system monitor items are displayed.
(6)	Monitor item selection	Add/delete items to monitor in the system monitor display items.

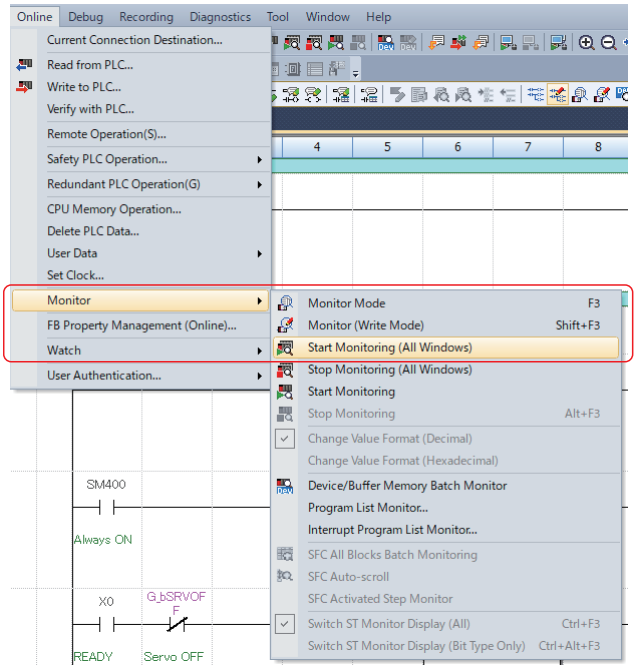
Program monitor

The program monitor enables the user to check the current status of the program in execution on the program editor.

Displaying the program monitor screen

The program monitor screen can be displayed by using the following methods.

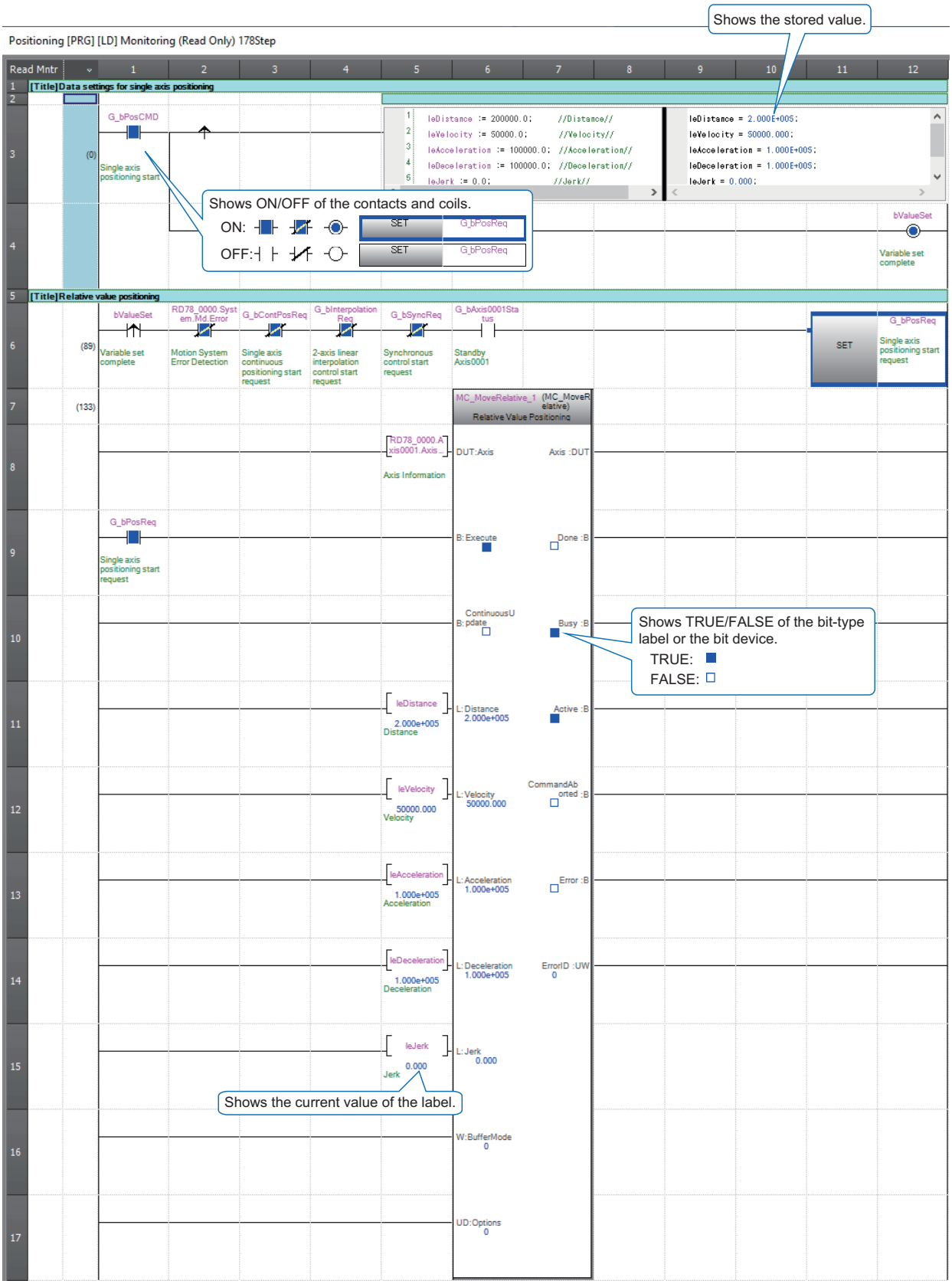
- Select [Online]⇨[Monitor]⇨[Start Monitoring (All Windows)] from the menu in MELSOFT GX Works3.



- Click the [Start Monitoring (All Windows)] icon on the tool bar.

Program monitor screen

The following contents are displayed on the program monitor.

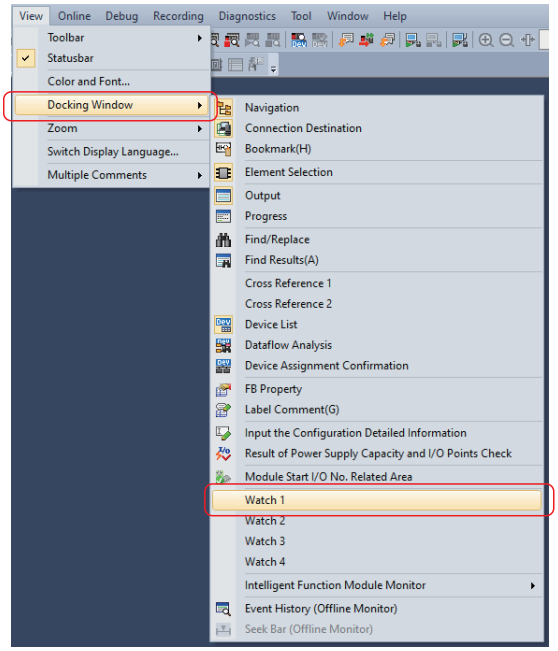


Watch

The watch function enables the user to check the current values of registered devices and labels.
Register the devices and labels to be monitored to a watch window.

Displaying the watch window

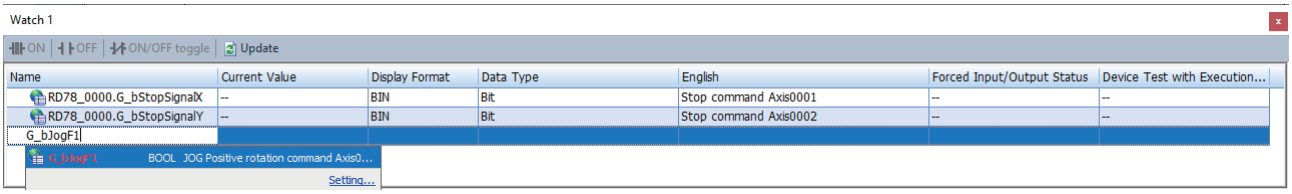
Select [View]⇨[Docking Window]⇨[Watch 1] to [Watch 4] from the menu in MELSOFT GX Works3.



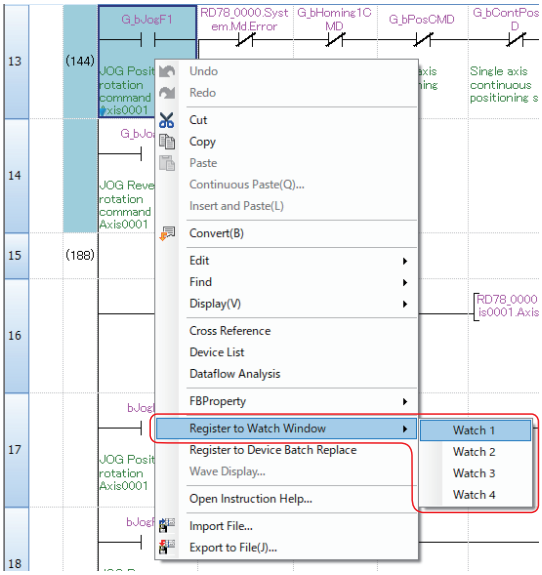
Registration of devices/labels/structures to a watch window

Register devices/labels/structures by using the following methods.

- Enter the device No., label, or structure in the "Name" column of a watch window.

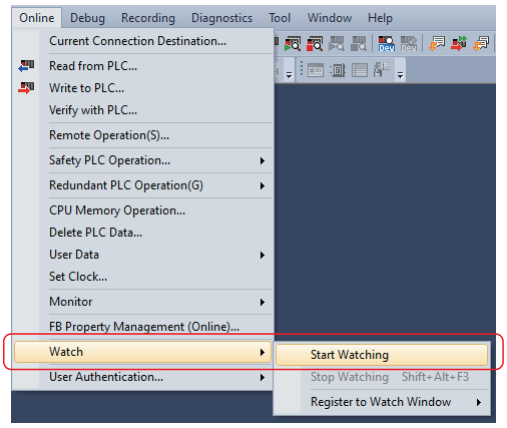


- Select the label or structure to register on the program editor, right-click the label or structure, then select [Register to Watch Window]⇨[Watch 1] to [Watch 4].



Monitoring start

Select [Online]⇨[Watch]⇨[Start Watching] from the menu in MELSOFT GX Works3.



Changing current values

Directly enter a value in "Current Value" during monitoring.

To switch bit devices and bit type labels ON/OFF (to TRUE/FALSE), select the row, then double-click while holding Shift or press Shift + Enter.

Watch 1[Watching]

ON | OFF | ON/OFF toggle | Update

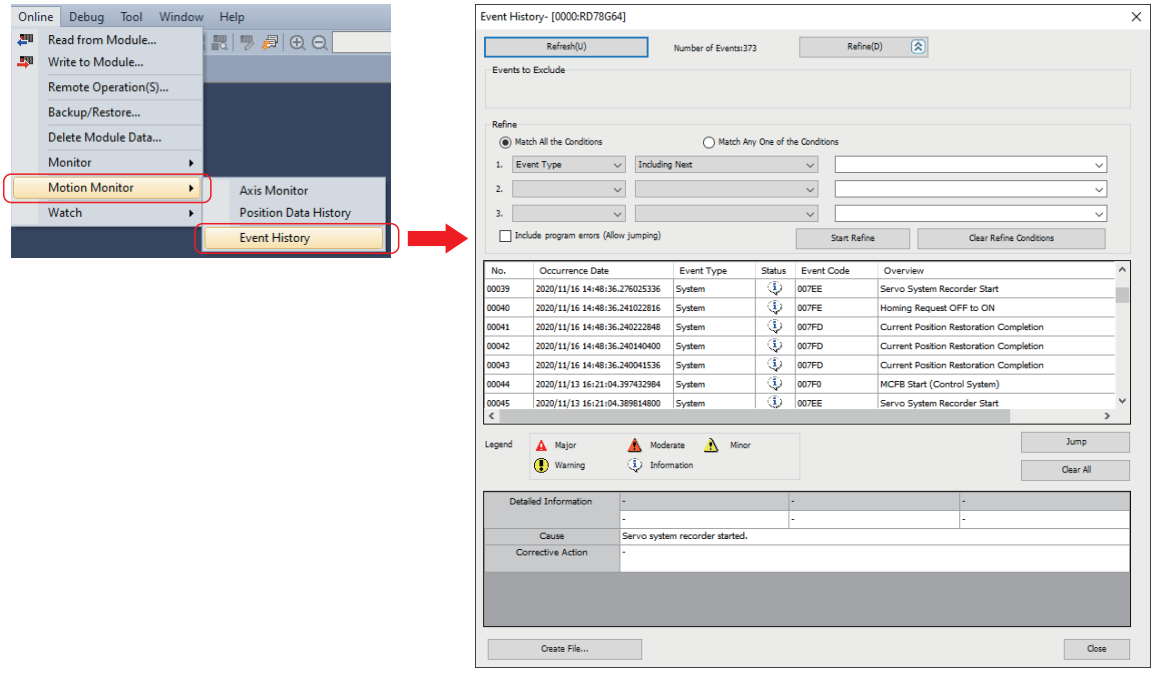
Name	Current Value	Display Format	Data Type	English	Forced Input/Output Status	Device Test with Execution...
RD78_0000.G_bStopSignalX	FALSE	BIN	Bit	Stop command Axis0001	--	--
RD78_0000.G_bStopSignalY	FALSE	BIN	Bit	Stop command Axis0002	--	--
G_bSRVOFF	FALSE	BIN	Bit	Servo OFF	--	--
G_bJogF1	FALSE	BIN	Bit	JOG Positive rotation command Axis0001	--	--
G_bJogR1	FALSE	BIN	Bit	JOG Reverse rotation command Axis0001	--	--
G_bJogF2	FALSE	BIN	Bit	JOG Positive rotation command Axis0002	--	--
G_bJogR2	FALSE	BIN	Bit	JOG Reverse rotation command Axis0002	--	--
G_bHoming1CMD	FALSE	BIN	Bit	Homing command Axis0001	--	--
G_bHoming2CMD	FALSE	BIN	Bit	Homing command Axis0002	--	--
G_bHoming3CMD	FALSE	BIN	Bit	Homing command VirtualAxis0001	--	--
G_bPosCMD	FALSE	BIN	Bit	Single axis positioning start	--	--
G_bContPosCMD	FALSE	BIN	Bit	Single axis continuous positioning start	--	--
G_bInterpolationCMD	FALSE	BIN	Bit	2-axis linear interpolation control start	--	--
G_bSyncCMD	FALSE	BIN	Bit	Synchronous control start	--	--
G_bErrorReset	FALSE	BIN	Bit	Error reset	--	--
G_leJogVelocity	25000.00000000000000	--	FLOAT [Double Precision]	JOG Velocity	--	--

Point

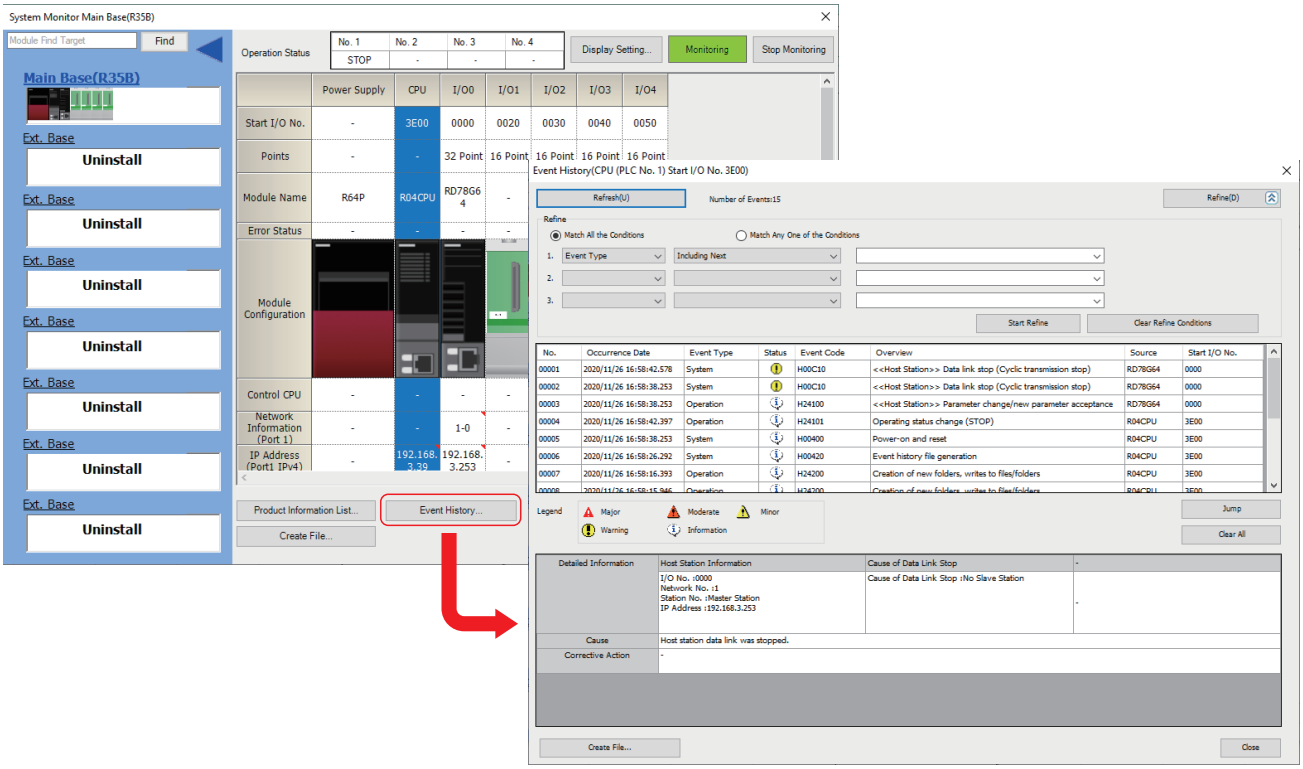
In the sample program, the labels "RD78_0000.G_bStopSignalX" and "RD78_0000.G_bStopSignalY" are registered in watch window 1. Turning these signals ON (to TRUE) while operating a program aborts the program.

Event history

The event history of the Motion module can be checked from [Online]⇨[Motion Monitor]⇨[Event History] in Motion Control Setting Function. If an error occurs, check the details of the error. The error occurrence time in the event history of the Motion module synchronizes with those recorded in the servo amplifiers. Check the error details of the event history together with the data in the servo amplifier.



To check the event history of the PLC CPU, click [Diagnostics]⇨[System Monitor] in MELSOFT GX Works3, then click "Event History" on the "System Monitor" screen.



APPENDICES

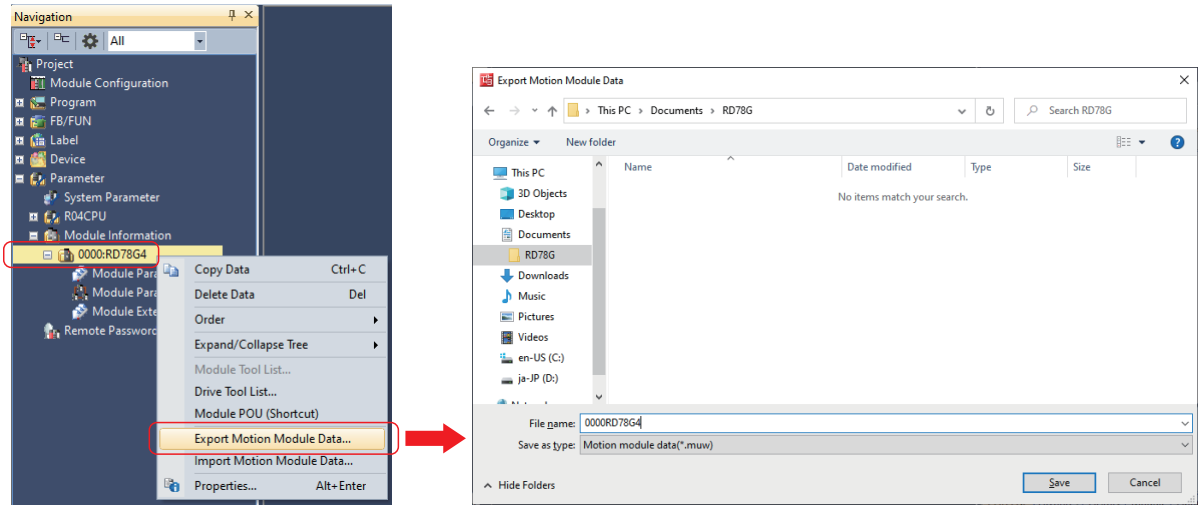
Appendix 1 Reusing Programs

This chapter explains how to reuse the parameters and programs of the Motion module from a sample program in another project. The following procedure can also be used when the Motion module is replaced.
This chapter shows an example where the RD78G4 is replaced with RD78GHV.

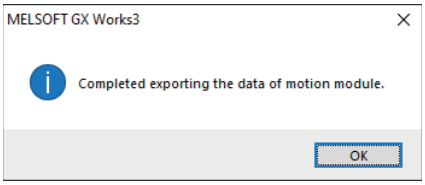
Exporting the Motion module data

Output the parameters and programs of the Motion module to one file.

1. Start MELSOFT GX Works3, then open the data to reuse. In the navigation window, double-click "Parameter"⇨"Module information". Right-click "0000_RD78G4", then select [Export Motion Module Data].
2. The "Export Motion Module Data" screen is displayed. Select the folder to export to, then click the [Save] button.



3. When the exporting is complete, a completed message appears.



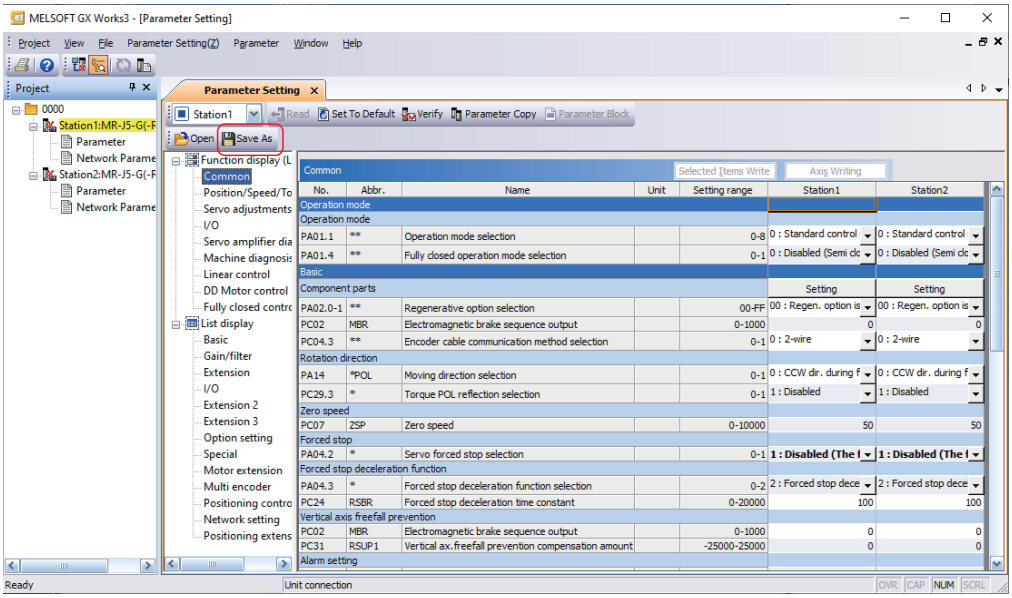
Saving servo parameters

Servo parameters are not included in the muw file outputted in Exporting the Motion module data (☞ Page 142 Exporting the Motion module data).

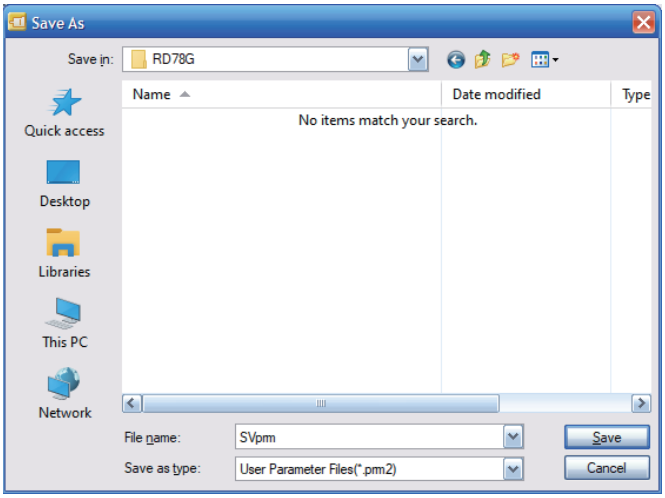
When writing the servo parameters from the Motion module to the servo amplifier, save the servo parameters of the program being reused in a separate file.

1. In the "CC-Link IE TSN Configuration" window, double-click the illustration of the set servo amplifier to display the "Parameter Setting" screen. For the "CC-Link IE TSN Configuration" window, refer to the following.
   ☞ Page 37 Network Configuration, ☞ Page 40 Servo Parameter Setting

2. On the "Parameter Setting" screen, click the [Save As] button.



3. The "Save As" screen is displayed. Select a folder to save to, then click the [Save] button. The servo parameters are outputted to a prm2 file.



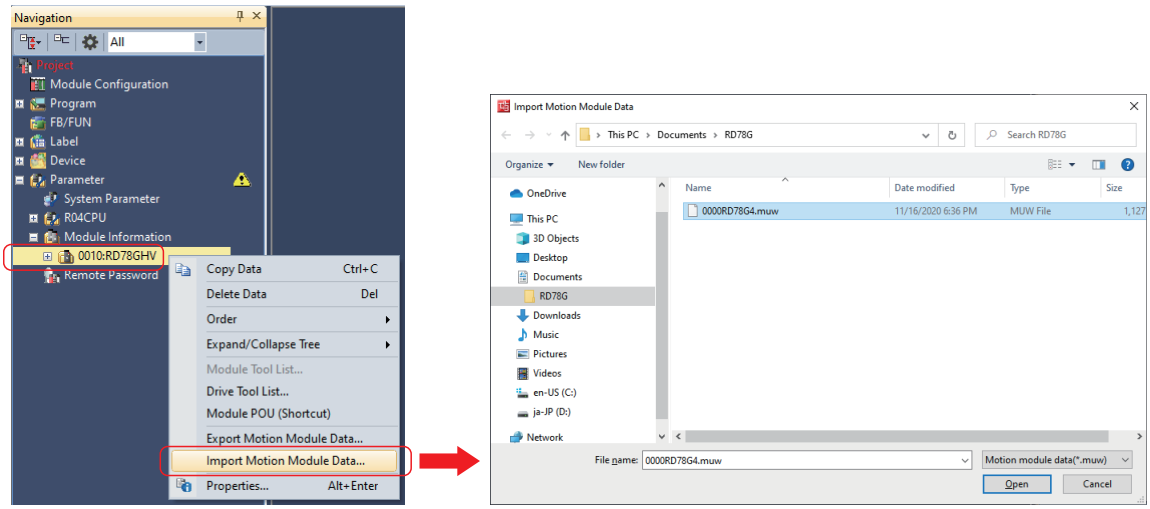
A

Reflection to the editing program

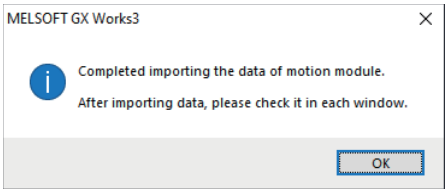
Start MELSOFT GX Works3 and create a module configuration for the editing program.

Importing the Motion module data

1. In the navigation window, double-click "Parameter"⇨"Module information". Right-click "0010:RD78GHV", then select [Import Motion Module Data].
2. The "Import Motion Module Data" screen is displayed. Select the muw file outputted in Exporting the Motion module data (☞ Page 142 Exporting the Motion module data), then click the [Open] button.



3. When the importing is complete, a completed message appears.

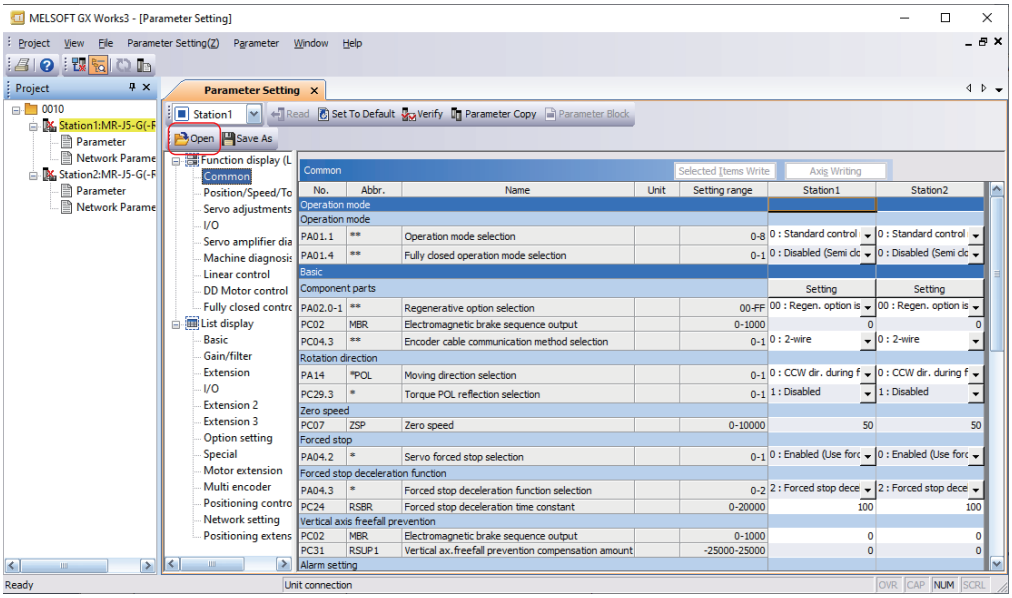


Reading servo parameters

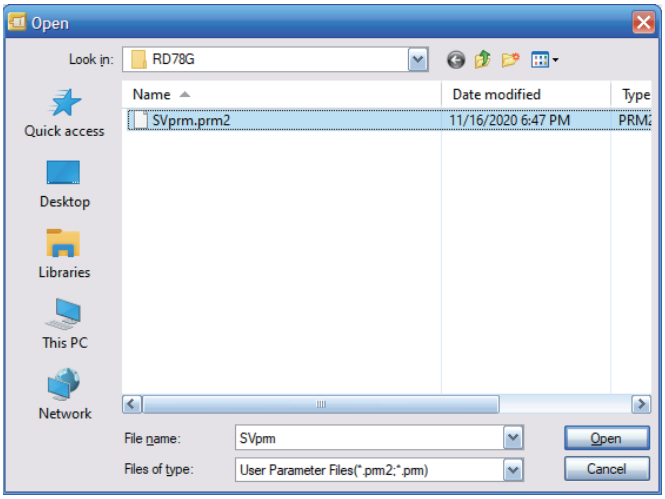
To read the servo parameters from the Motion module, read the servo parameter file saved in Saving servo parameters (☞ Page 143 Saving servo parameters).

1. Double-click the illustration of the set servo amplifier in the "CC-Link IE TSN Configuration" window to display the "Parameter Setting" screen. For the "CC-Link IE TSN Configuration" window, refer to the following.
☞ Page 37 Network Configuration, ☞ Page 40 Servo Parameter Setting

2. On the "Parameter Setting" screen of the editing program, click the [Open] button.

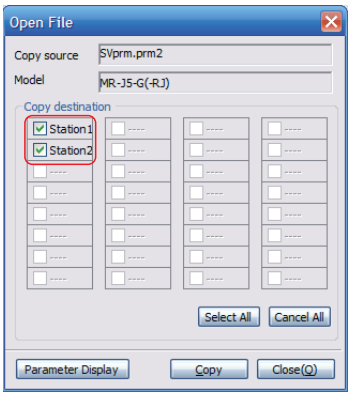


3. The "Open" screen is displayed. Select the prm2 file saved in Saving servo parameters (☞ Page 143 Saving servo parameters), then click the [Open] button.

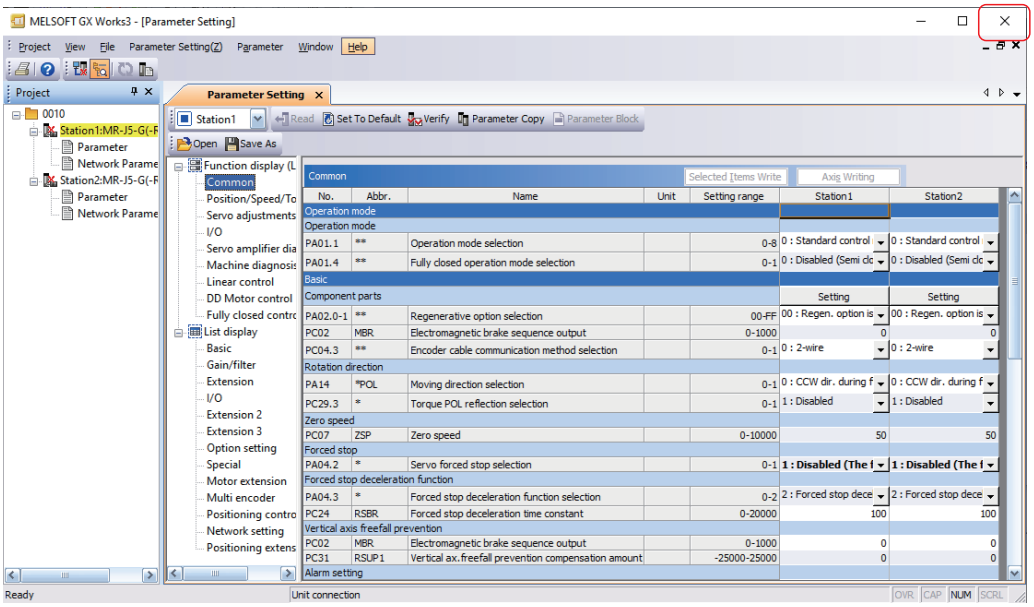


A

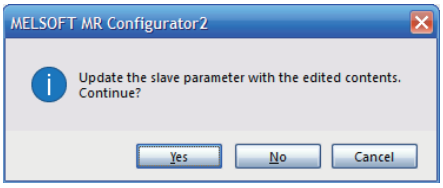
4. The "Open File" screen is displayed. Select the stations to read the servo parameters, then click the [Copy] button.



5. Make sure that the servo parameters are read, then click the [×] button at the top right of the "Parameter Setting" screen.



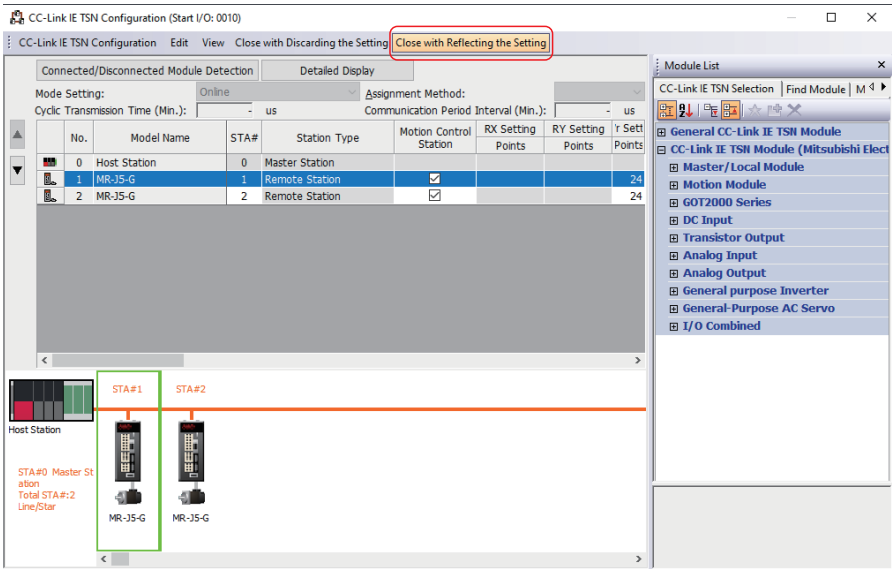
6. When the confirmation message appears, click the [Yes] button to update the parameters.



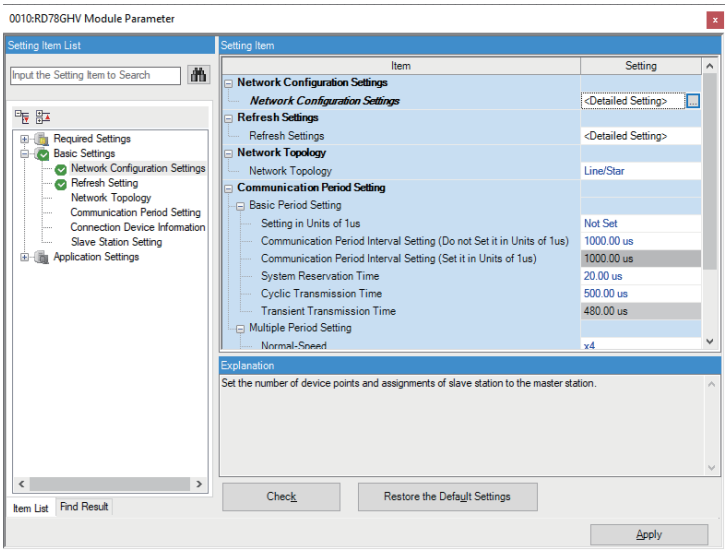
Fixing parameters

Fix the CC-Link IE TSN configuration and the module parameter (network).

1. Click [Close with Reflecting the Setting] in the menu on the "CC-Link IE TSN Configuration" window. Reflect the data, then close the "CC-Link IE TSN Configuration" window.



2. In the parameter editor (Module Parameter), click the [Apply] button to reflect the parameters of the Motion module.



Correcting programs

If the start I/O number of the program being reused and the editing program are different, change the following:

- PLC READY [Y0] of the sequence program
- READY [X0]
- The device No. of Synchronization flag [X1]
- The module No. of the buffer memory address (from U**\G)
- Public labels being used

Be sure to change the above especially when the module is changed to RD78GHV or RD78GHW. The RD78GHV or RD78GHW has reserved 16 points and available 32 points for input and output, and when the module is installed to the slot No. 0 to 1, the device of the PLC READY is [Y10].

Additionally, the start of the public label name changes from [RD78_0000.****] to [RD78_0010.****].

After reflecting the public labels in Conversion of all programs and reflection of public labels (☞ Page 148 Conversion of all programs and reflection of public labels), change all of the label names in the program using character string replacement.

Conversion of all programs and reflection of public labels

First convert all the programs of the Motion module, then reflect the public labels. After the public labels are reflected, convert all the programs of the PLC CPU.
If the programs of the PLC CPU are converted first, the public labels are not defined, and an error will occur.

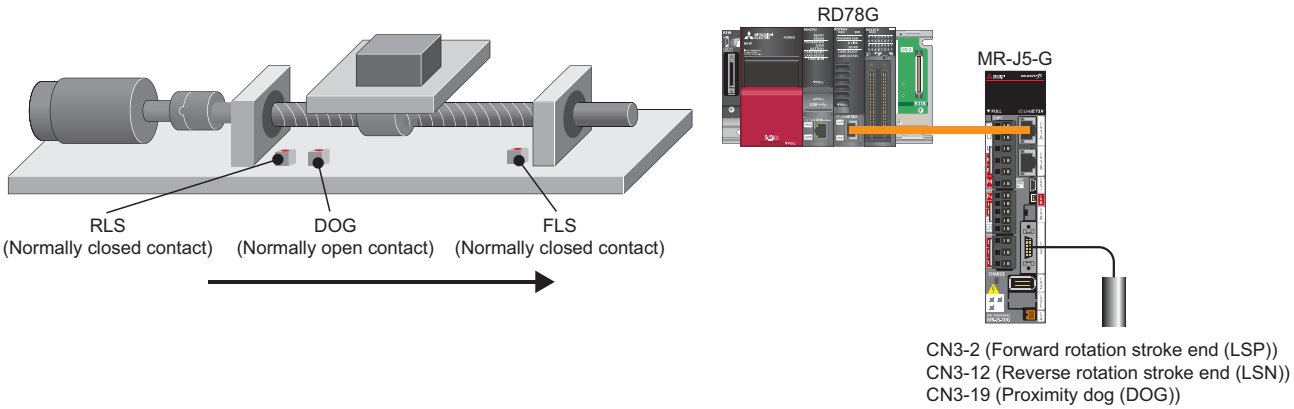
Appendix 2 Using External Input Signals

External input signals can be used for the hardware stroke limit signals and the proximity dog signal.

Usable external signals	Reference
DI signal of the servo amplifier	☞ Page 149 Using the DI signal of the servo amplifier
Input signal of the PLC CPU	☞ Page 156 Using the input signal of the PLC CPU
Input signal of the remote input module	☞ Page 161 Using the input signal of the remote input module

Using the DI signal of the servo amplifier

This section explains the case where the LSP/LSN signals of the servo amplifier are used for the hardware stroke limit signals and the case where homing is executed with the DOG signal of the servo amplifier.



Hardware stroke limit

The negative logic (normally closed contact) is recommended for wiring of the hardware stroke limit.

⚠CAUTION

• If the positive logic (normally open contact) is used for wiring of the hardware stroke limit, it may cause a serious accident at sensor failure or disconnection.

■The connection destination of the servo amplifier signals and the location of the sensors

The setting location of the sensor varies according to the setting of the servo parameter [Pr. PA14 (Travel direction selection)].

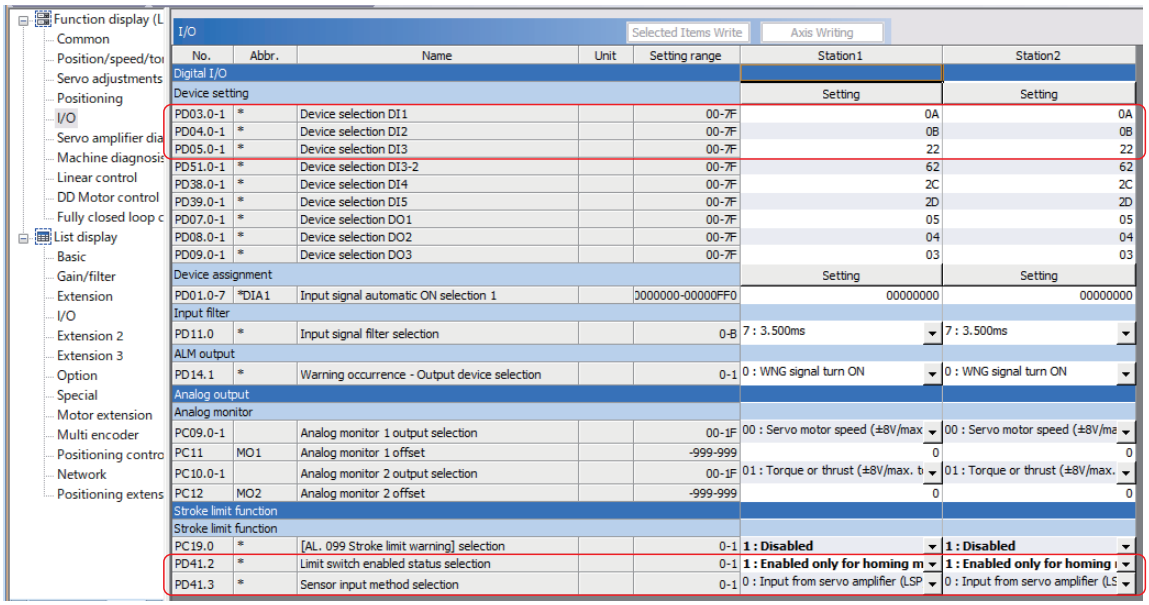
Set the limit switches connected to the servo amplifier as follows:

[Pr. PA14 (Travel direction selection)]	LSP signal	LSN signal
0 (CCW or positive direction when the positioning address increases)	Upper stroke limit of the positioning address increasing side (FLS)	Lower stroke limit of the positioning address decreasing side (RLS)
1 (CW or negative direction when the positioning address increases)	Lower stroke limit of the positioning address decreasing side (RLS)	Upper stroke limit of the positioning address increasing side (FLS)

■Servo parameters

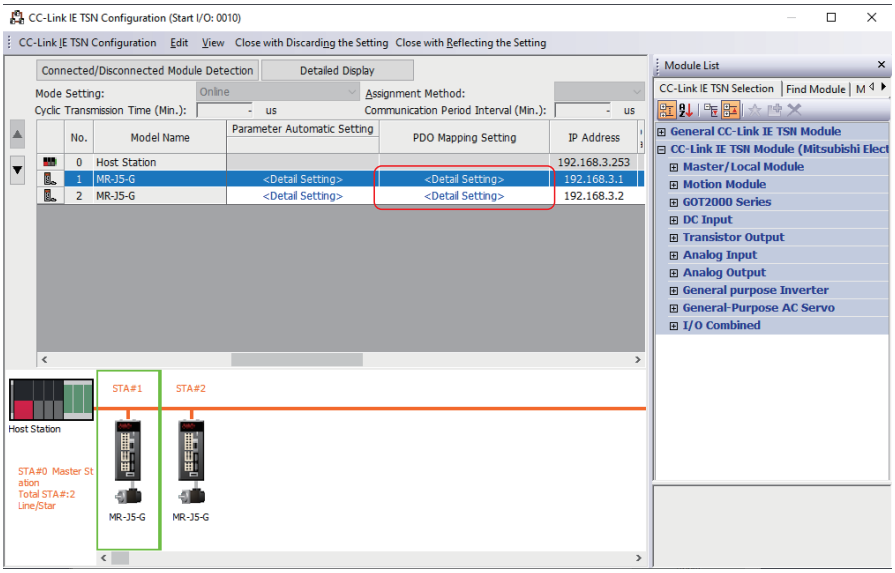
Make sure that the servo parameters are set as follows:

No.	Name	Setting value
Pr.PD03.0-1	Device selection DI1	0A (LSP)
Pr.PD04.0-1	Device selection DI2	0B (LSN)
Pr.PD05.0-1	Device selection DI3	22 (DOG)
Pr.PD41.2	Limit switch enabled status selection	1: Enabled only for homing mode
Pr.PD41.3	Sensor input method selection	0: Input from servo amplifier (LSP/LSN/DOG)



■Changing the PDO mapping

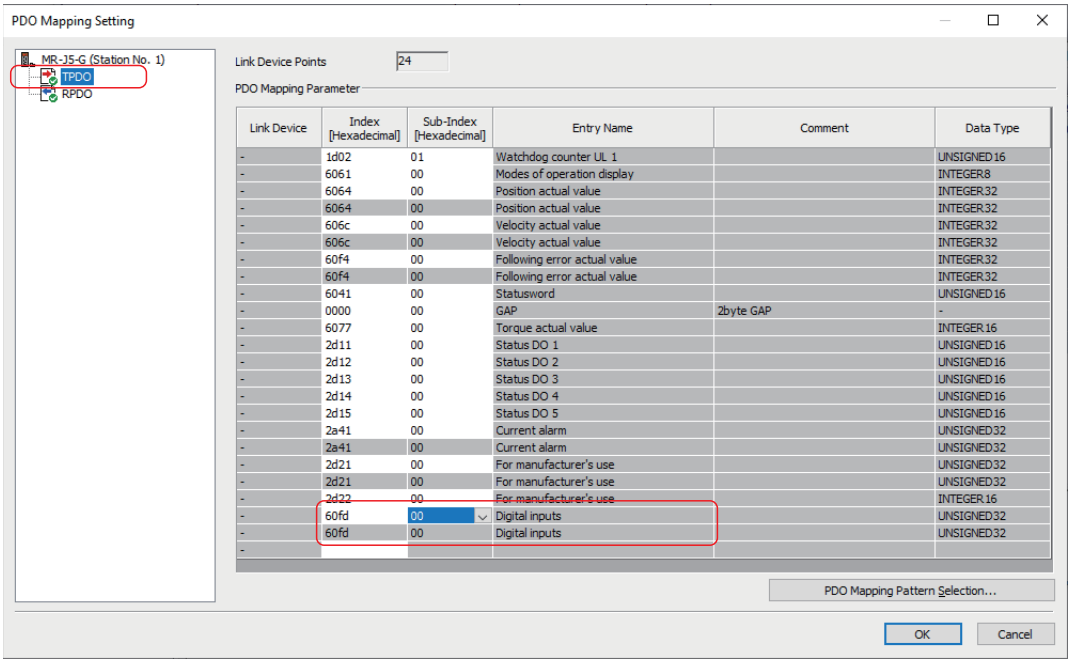
1. In the "CC-Link IE TSN Configuration" window, double-click "<Detail Setting>" in "PDO Mapping Setting".



2. The "PDO Mapping Setting" screen is displayed. The PDO mapping is set in TPDO. Select TPDO to display the TPDO list. Set the end of the PDO mapping parameter as follows, then click the [OK] button.

Index (hexadecimal)	Sub-index (hexadecimal)
60fd	00

After the parameter is set, the entry name is displayed as "Digital inputs". (The data is double word type, so it is displayed in two lines.)



Set as above for all axes that use hardware limit signals connected to the servo amplifier.

Point

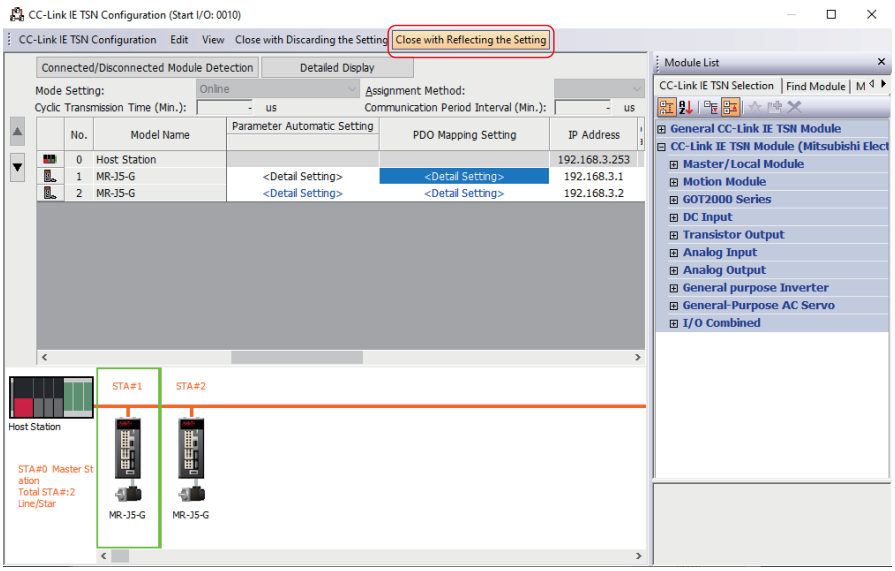
"Digital inputs" is a label that contains the status of the servo amplifier input signal.

- bit0: Negative limit switch
- bit1: Positive limit switch

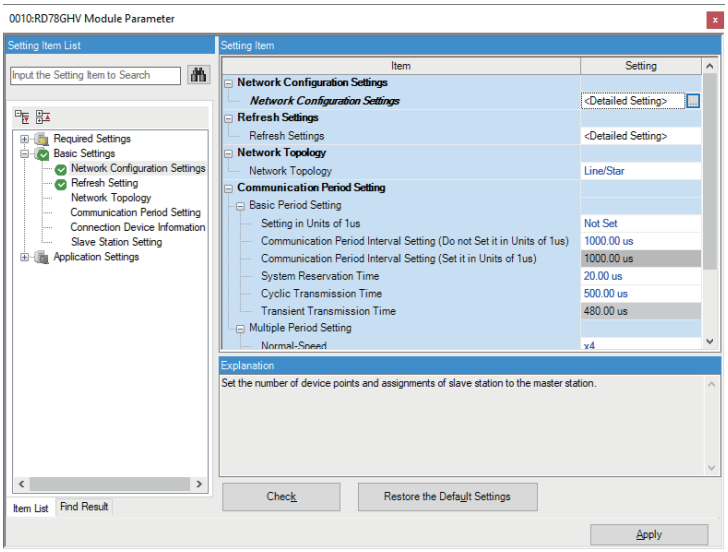
For details, refer to "[Digital inputs (Obj. 60FDh)]" in the following manual.

MR-J5-G/MR-J5W-G User's Manual (Object Dictionary)

3. After the setting is completed, select "Close with Reflecting the Setting" from the menu in the "CC-Link IE TSN Configuration" window. Reflect the data, then close the "CC-Link IE TSN Configuration" window.

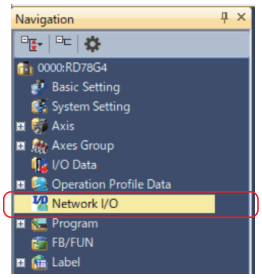


4. In the parameter editor (Module Parameter), click the [Apply] button to reflect the parameters of the Motion module.

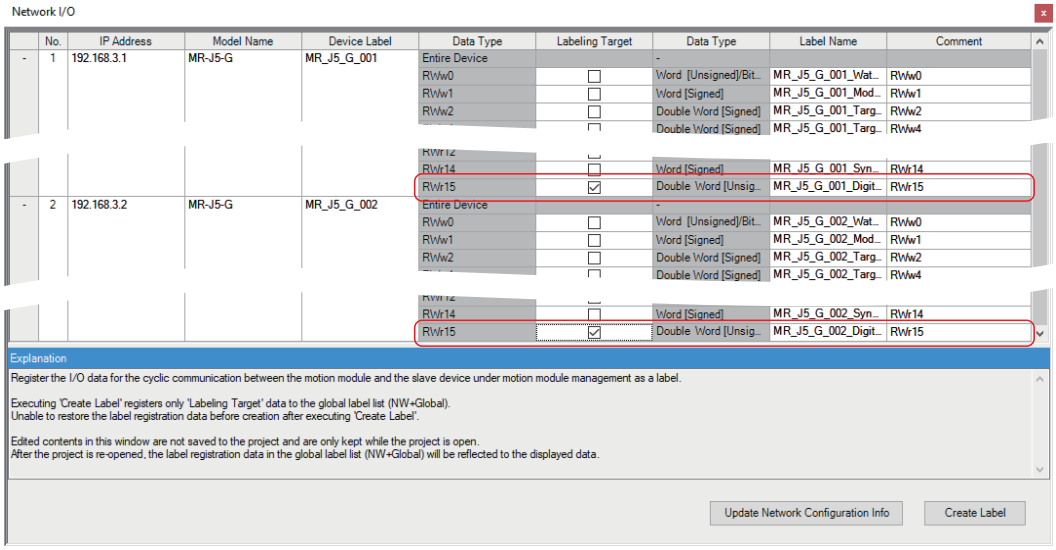


■Changing parameters of the Motion module

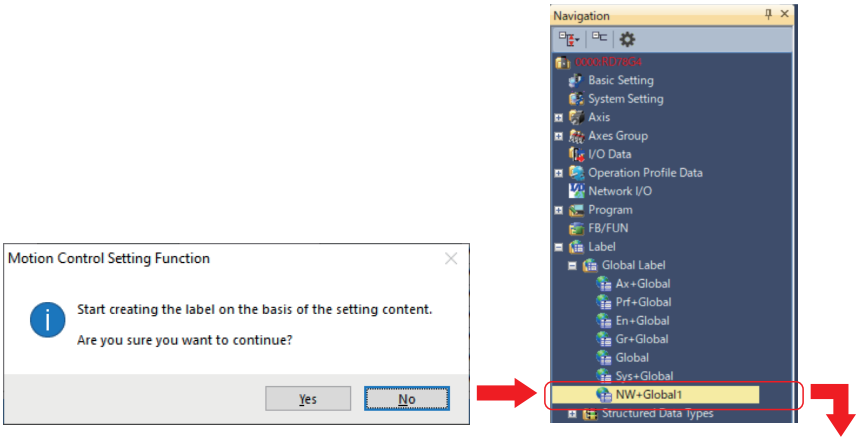
1. Double-click "Network I/O" in the navigation window of Motion Control Setting Function.



2. The devices controlled by the Motion module are displayed. Place a check in the "Labeling Target" checkbox of [MR_J5_G_***_DigitalInputs] (the label of the servo amplifier with the hardware stroke limit connected), then click the [Create Label] button.



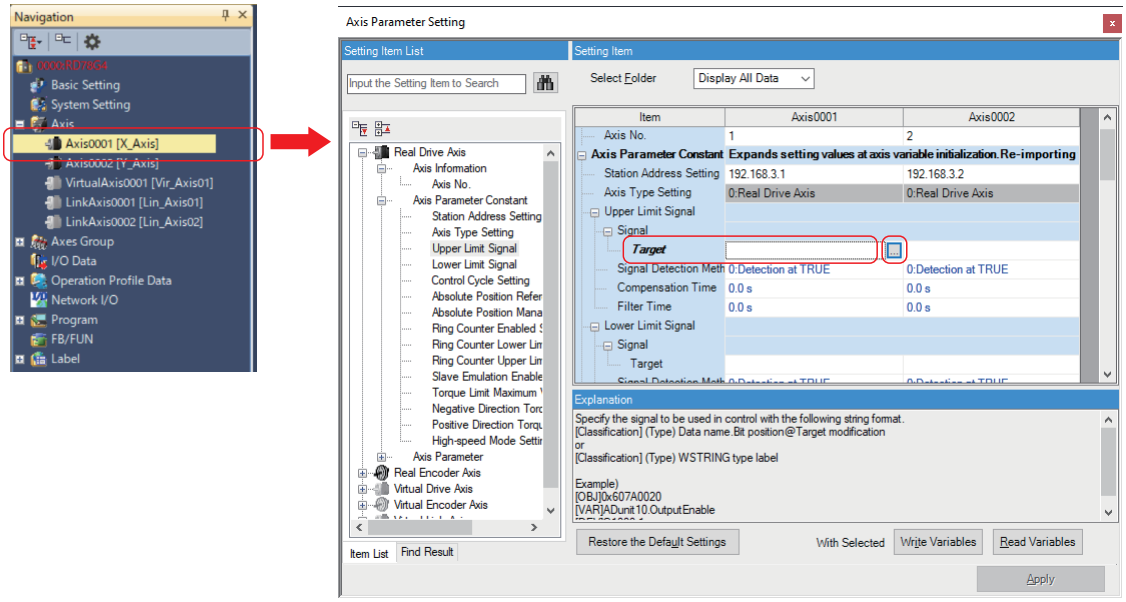
3. When the confirmation message appears, click the [Yes] button. "NW+Global1" is added to "Global Label" in the navigation window. The network I/O label is generated and stored in "NW+Global1".



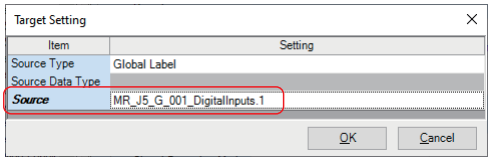
NW+Global1 [Global Label Setting]

	Label Name	Data Type	Class	Initial Value	Constant	Comment	Remark	Public Label	Motion Control Attribute
1	MR_J5_G_001_DigitalInputs	Double Word [Unsigned]/Bit String [3...	VAR_GLOBAL			RWr15		Disable	READ (Motion =>): RWr
2	MR_J5_G_002_DigitalInputs	Double Word [Unsigned]/Bit String [3...	VAR_GLOBAL			RWr15		Disable	READ (Motion =>): RWr
3									

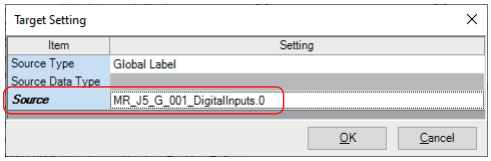
4. Open the axis parameters of the real drive axis.
From the navigation window, double-click "Axis"⇨name of the axis to be used (initial name: "Axis0001").
The "Axis Parameter Setting" window is displayed. Select "Real Drive Axis"⇨"Upper Limit Signal", then click the [...] button in "Target".



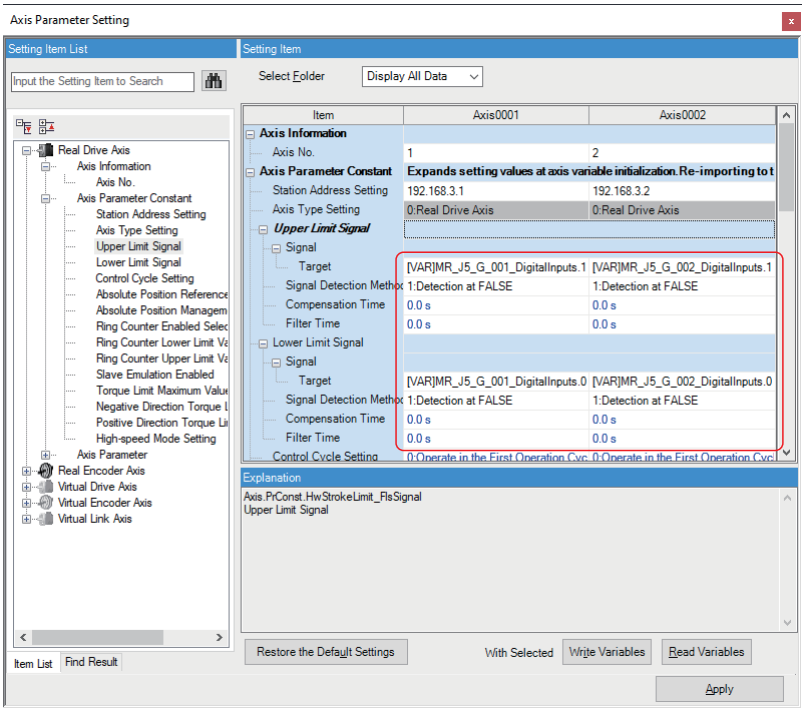
5. The "Target Setting" screen is displayed. Enter "MR_J5_G_***_DigitalInputs.1" in "Source".



6. In the same manner, enter "MR_J5_G_***_DigitalInputs.0" in "Source" on the "Target Setting" screen of the lower limit signal.



7. Perform steps 5. to 6. for all axes that use hardware limit signals connected to the servo amplifier. Make sure that the relation between the axis label and the actual servo amplifier is correct.
In addition, change the signal detection method to "1: Detection at FALSE" because the normally closed contact is used.



8. The setting is completed. Convert all the programs.

■Operation check

Write the program, then check whether ON/OFF of each signal can be monitored with "Upper Limit Signal Status" and "Lower Limit Signal Status" on the axis monitor.

To restore the value within the limit range, perform an error reset once, then move the axis to the direction within the range with JOG operation, etc.

Homing with the DOG signal

This section explains how to use the DOG signal of the servo amplifier for Homing Switch, such as using the proximity dog type homing.

■Servo parameters

Set the homing method and the polarity of the proximity dog signal with the following parameters.

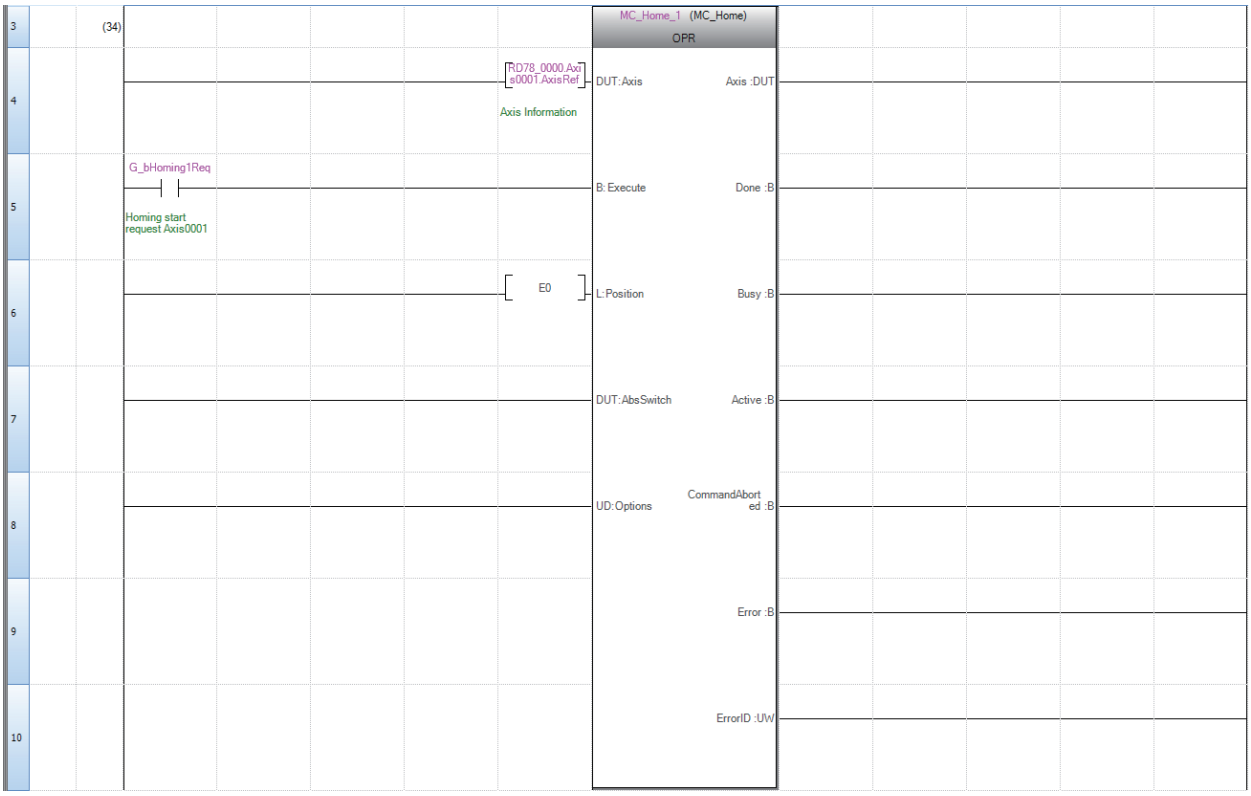
No.	Name
Pr. PT29.0	Device input polarity 1
Pr. PT45	Homing method

Set parameters from the following table as necessary according to the homing method.
(The parameters that require setting vary by the homing method.)

No.	Name
Pr. PT05	Homing speed
Pr. PT06	Creep speed
Pr. PT07	Home position shift distance
Pr. PT08	Homing position data
Pr. PT09	Travel distance after proximity dog
Pr. PT55.0	Homing deceleration time constant selection
Pr. PT56	Homing acceleration time constant
Pr. PT57	Homing deceleration time constant

■Program

For programs that use the DOG signal of the servo amplifier, the AbsSwitch input of MC_Home is omitted.

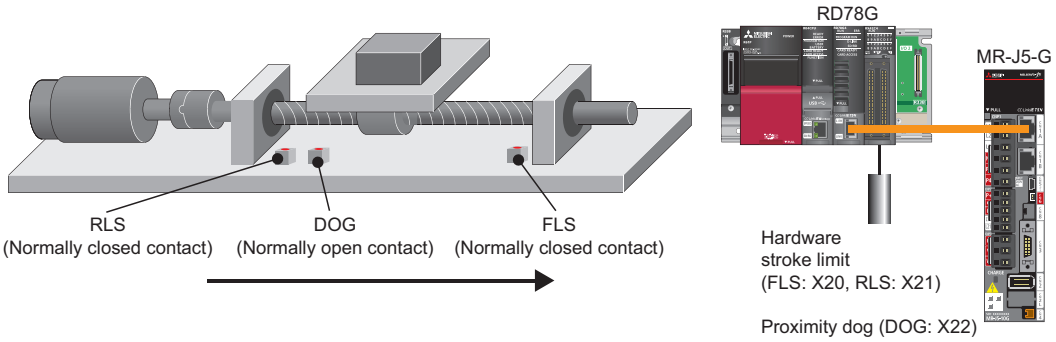


■Operation check

Write the program, then make sure that homing is correctly executed with the specified homing method.

Using the input signal of the PLC CPU

This section explains the case where the PLC CPU input signal is used for the hardware stroke limit signals and the proximity dog signal.



Hardware stroke limit

The negative logic (normally closed contact) is recommended for wiring of the hardware stroke limit.

⚠CAUTION

- If the positive logic (normally open contact) is used for the wiring of the hardware stroke limit, it may cause a serious accident at sensor failure or disconnection.

In this example, X20 is used as the FLS signal, and X21 is used as the RLS signal.

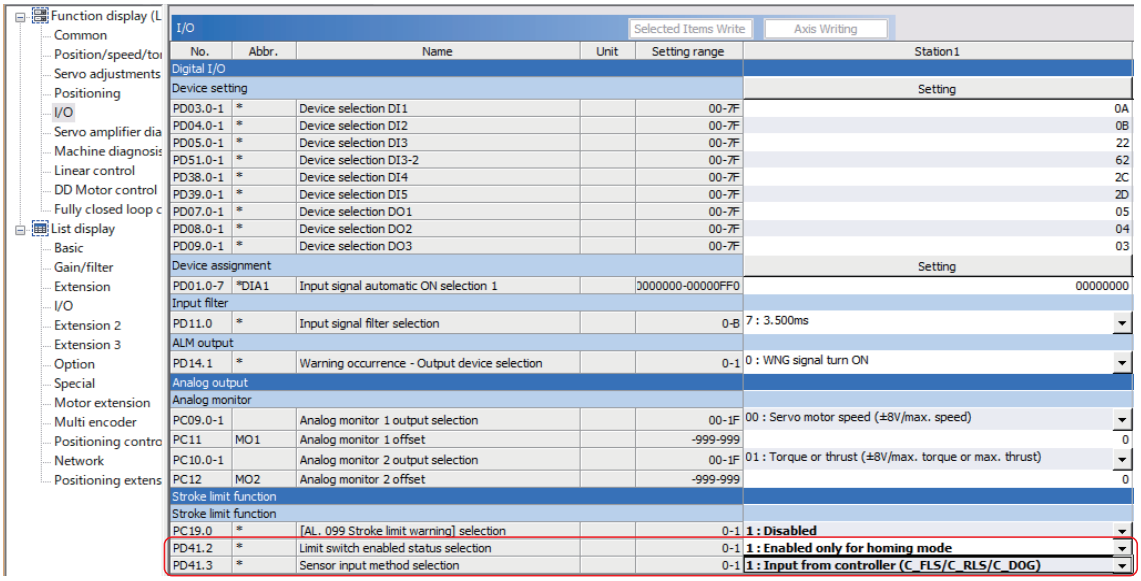
■The location of the sensors

Set the upper stroke limit (FLS) to the positioning address increasing side, and set the lower stroke limit (RLS) to the positioning address decreasing side.

■Servo parameters

Set the following servo parameters as shown below:

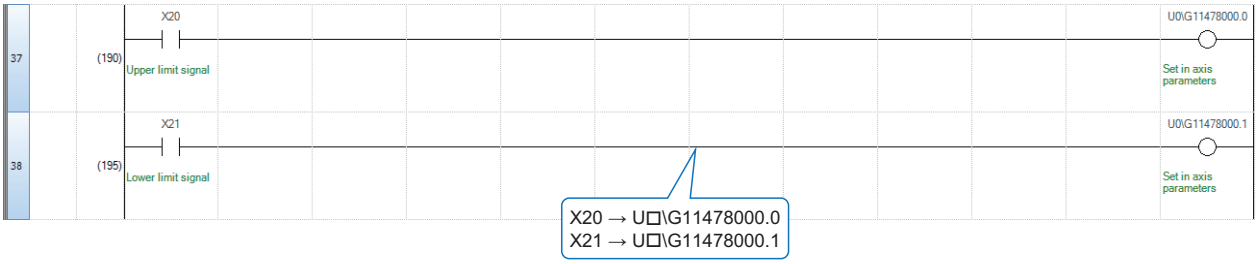
No.	Name	Setting value
Pr. PD41.2	Limit switch enabled status selection	1: Enabled only for homing mode
Pr. PD41.3	Sensor input method selection	1: Input from controller (C_FLS/C_RLS/C_DOG)



■Program

The ON/OFF status of X20 and X21 is copied to the public label or the buffer memory and transmitted to the Motion module.

The following shows a program when the ON/OFF status is copied to the buffer memory.

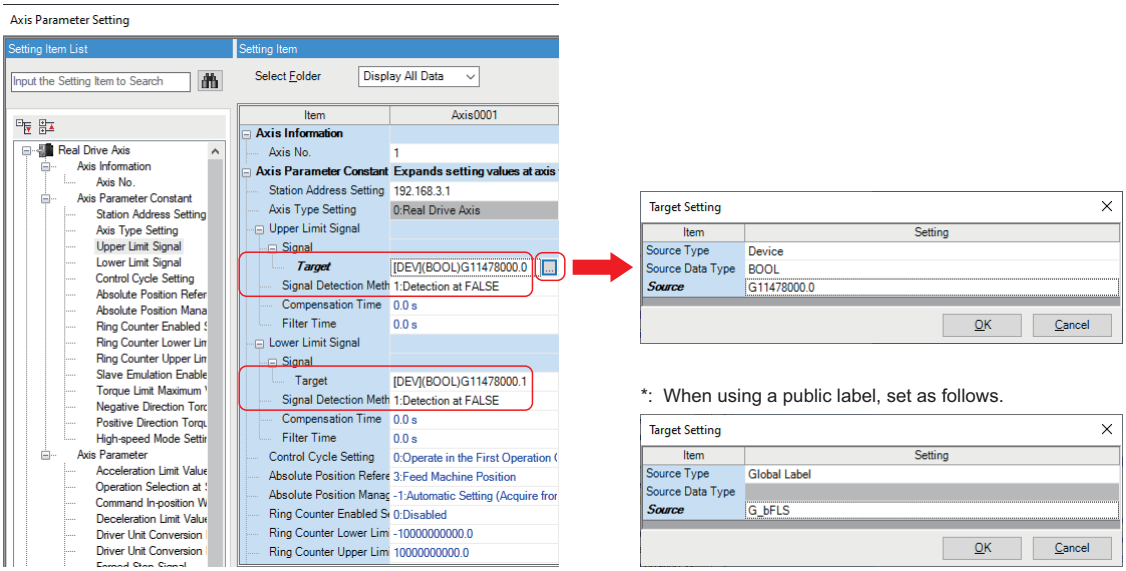


Point

The user setting area of the buffer memory is "11478000 to 11997999".

■Parameters of the Motion module

In the "Axis Parameter Setting" window, set "Upper Limit Signal" and "Lower Limit Signal" as follows:



Set as above for all axes that use hardware limit signals.

Change the signal detection method to "1: Detection at FALSE" because the normally closed contact is used.

■Operation check

Write the program, then check whether ON/OFF of each signal can be monitored with "Upper limit signal status" and "Lower limit signal status" of the axis monitor.

To restore the value within the limit range, perform an error reset once, then move the axis to the direction within the range with JOG operation, etc.

Homing with the DOG signal

In this example, X22 is used as a DOG signal.

■Servo parameters

Set the homing method and the polarity of the proximity dog signal with the following parameters.

No.	Name
Pr. PT29.0	Device input polarity 1
Pr. PT45	Homing method

Set parameters from the following table as necessary according to the homing method.
(The parameters that require setting vary by the homing method.)

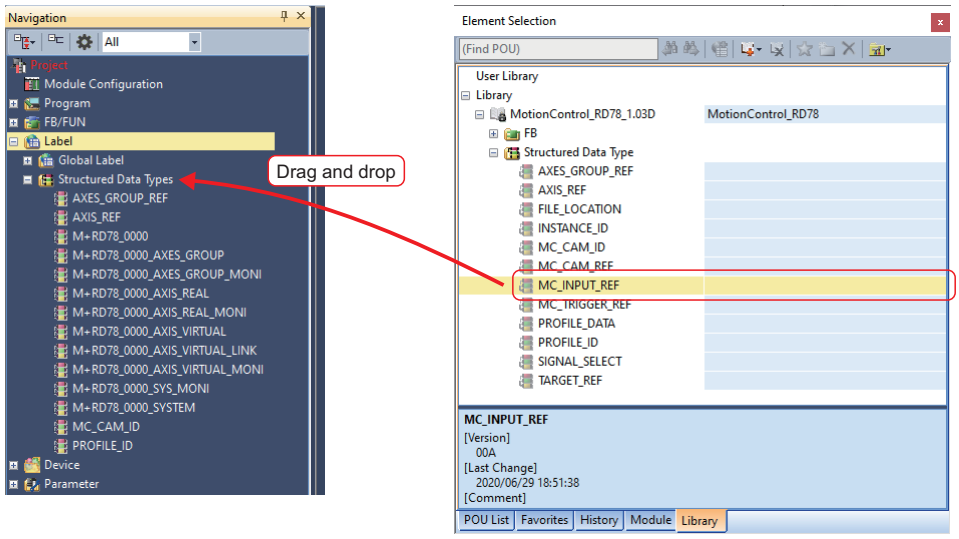
No.	Name
Pr. PT05	Homing speed
Pr. PT06	Creep speed
Pr. PT07	Home position shift distance
Pr. PT08	Homing position data
Pr. PT09	Travel distance after proximity dog
Pr. PT55.0	Homing deceleration time constant selection
Pr. PT56	Homing acceleration time constant
Pr. PT57	Homing deceleration time constant

■Structures

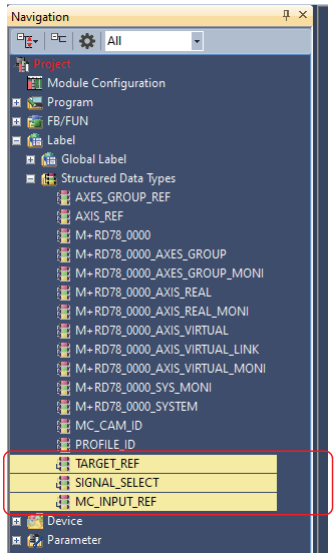
It is necessary to define an MC_INPUT_REF type structure to the PLC CPU.

1. Click the [Library] tab in the element selection window on MELSOFT GX Works3, then select "Library"⇨"MotionControl_RD78_****[*1]"⇨"Structured Data Type" to display the list of structured data types.
Select “MC_INPUT_REF”, then drag and drop it onto "Label"⇨"Structured Data Types" in the navigation window.

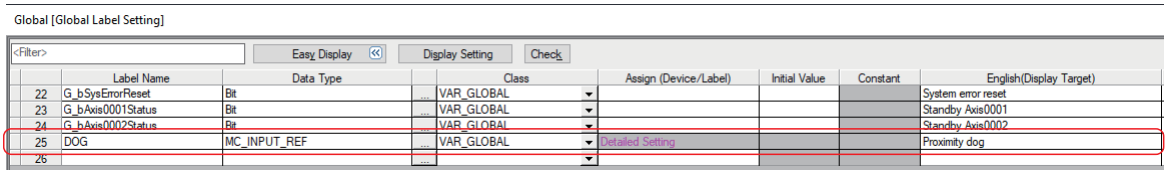
*1 **** = Motion module FB library version



2. "MC_INPUT_REF" and the structures of the members ("TARGET_REF" and "SIGNAL_SELECT") are registered.



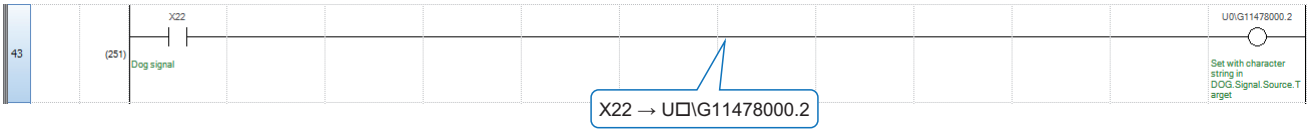
3. Prepare an MC_INPUT_REF type structure in the labels of the PLC CPU.
The label can be registered as either a global or public label. The label name is "DOG" in this example.



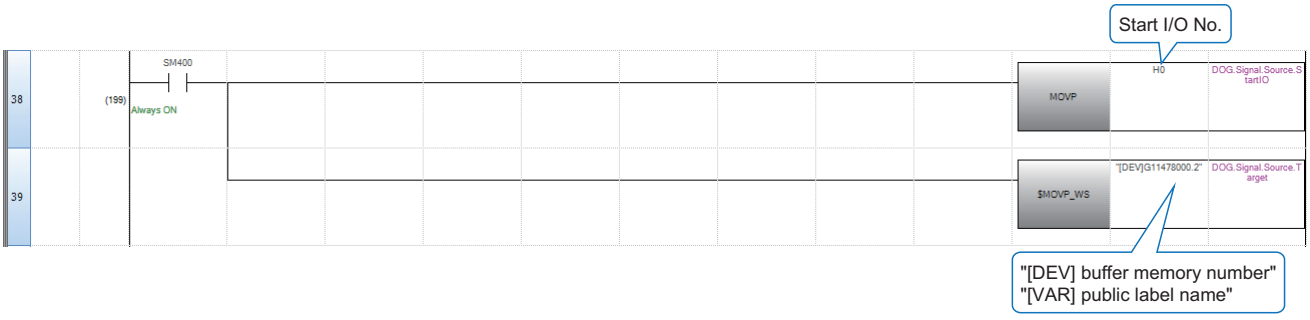
■Program

1. Set the members of the MC_INPUT_REF type structure with the program.

- Copy the ON/OFF status of "X22" to the buffer memory or the public label and transmit it to the Motion module.
  The following program is for when the status is copied to the buffer memory.



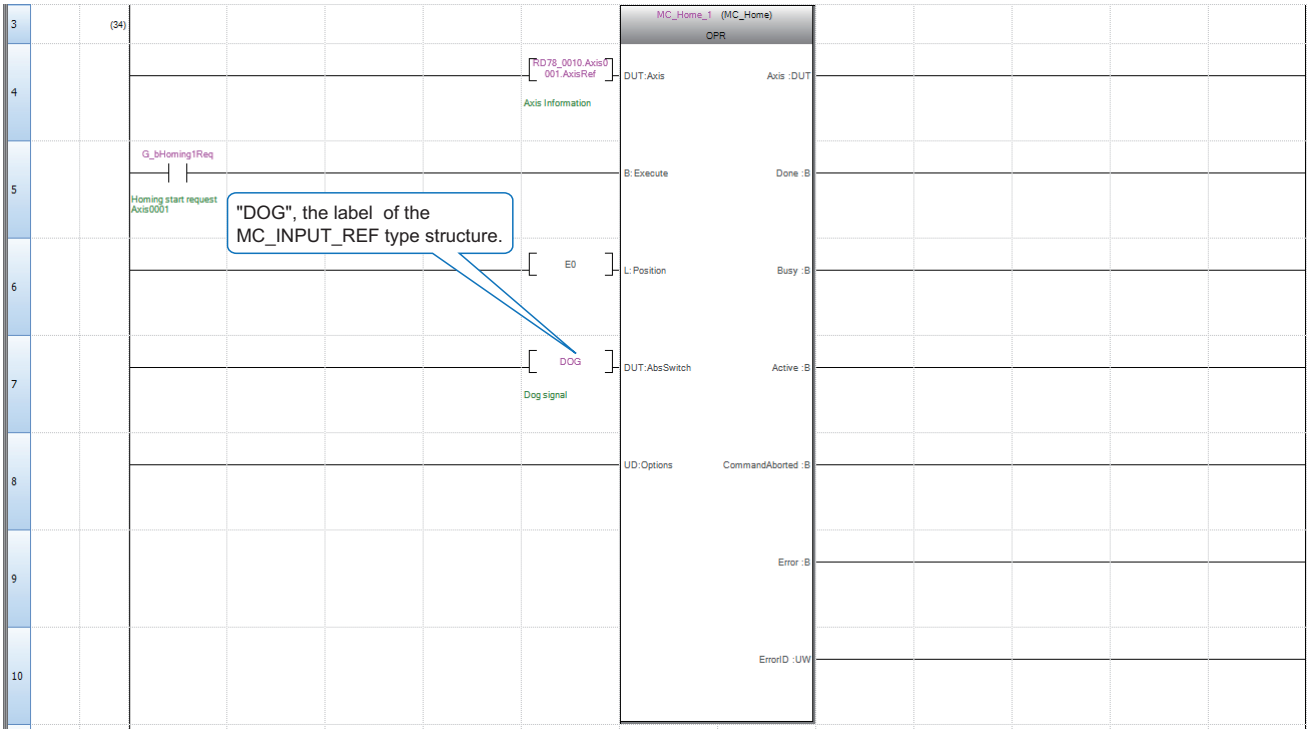
- Set the start I/O number of the Motion module in "DOG.Signal.Source.StartIO".
  Set the buffer memory number or the public label name with a Unicode character string in "DOG.Signal.Source.Target".



A

2. Write MC_Home to the program.

- Enter the label "DOG" (the MC_INPUT_REF type structure set above) in the AbsSwitch input of MC_Home.

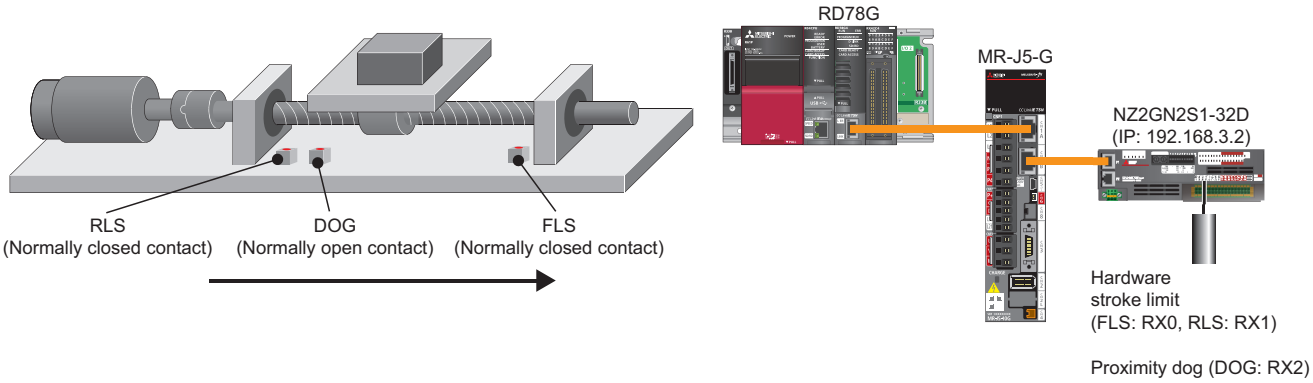


■Operation check

Write the program, then make sure that homing is correctly executed with the specified homing method.

Using the input signal of the remote input module

This section explains the case where the remote input module of the Motion control station is used for the hardware stroke limit signal and the proximity dog signal.



Hardware stroke limit

The negative logic (normally closed contact) is recommended for the wiring of the hardware stroke limit.

⚠CAUTION

- If the positive logic (normally open contact) is used for the wiring of the hardware stroke limit, it may cause a serious accident at sensor failure or disconnection.

In this example, "RX0" is used as the FLS signal, and "RX1" is used as the RLS signal.

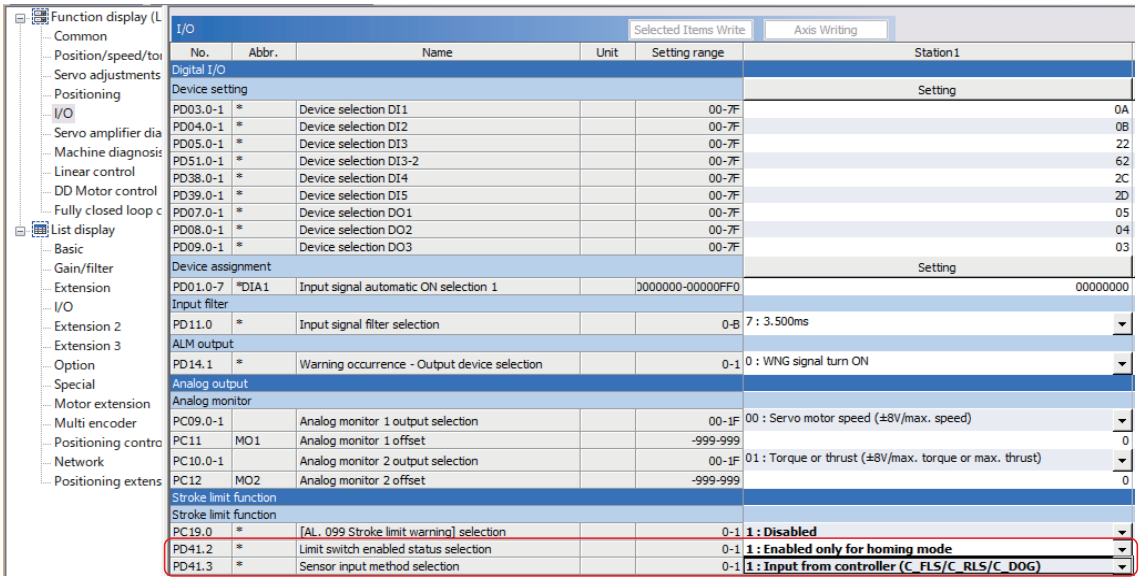
■The location of the sensors

Set the upper stroke limit (FLS) to the positioning address increasing side, and set the lower stroke limit (RLS) to the positioning address decreasing side.

■Servo parameters

Set the following servo parameters as shown below:

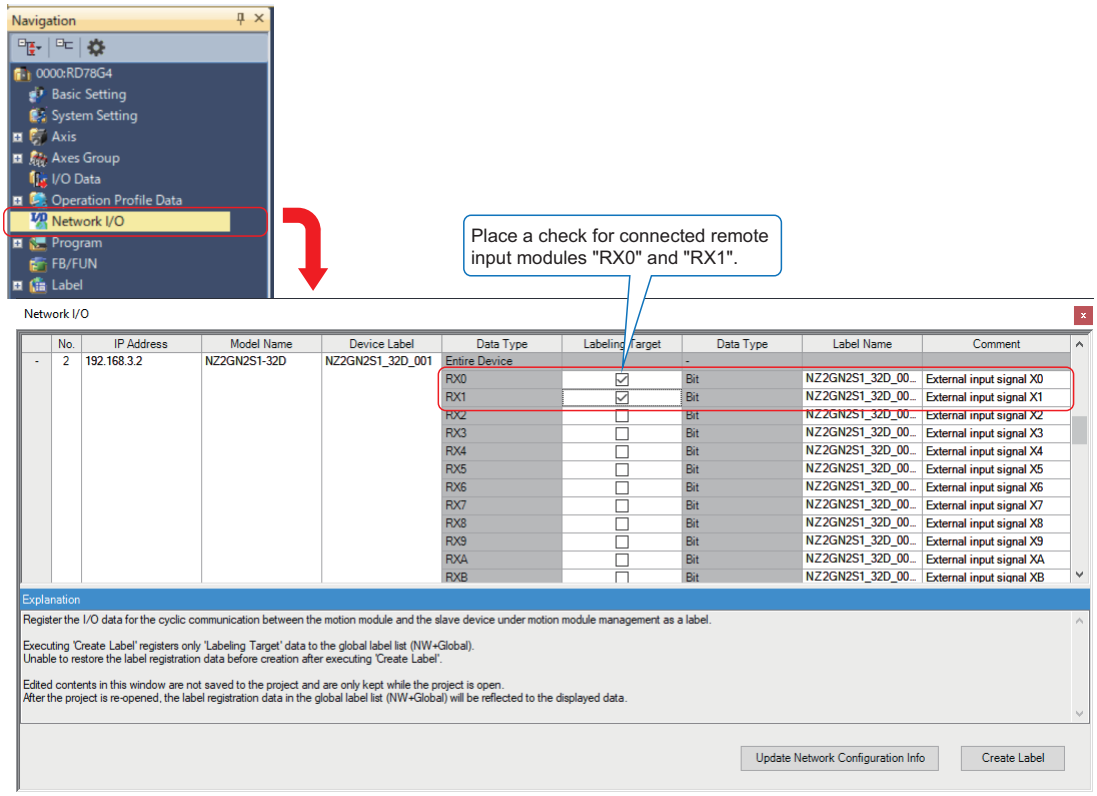
No.	Name	Setting value
Pr. PD41.2	Limit switch enabled status selection	1: Enabled only for homing mode
Pr. PD41.3	Sensor input method selection	1: Input from controller (C_FLS/C_RLS/C_DOG)



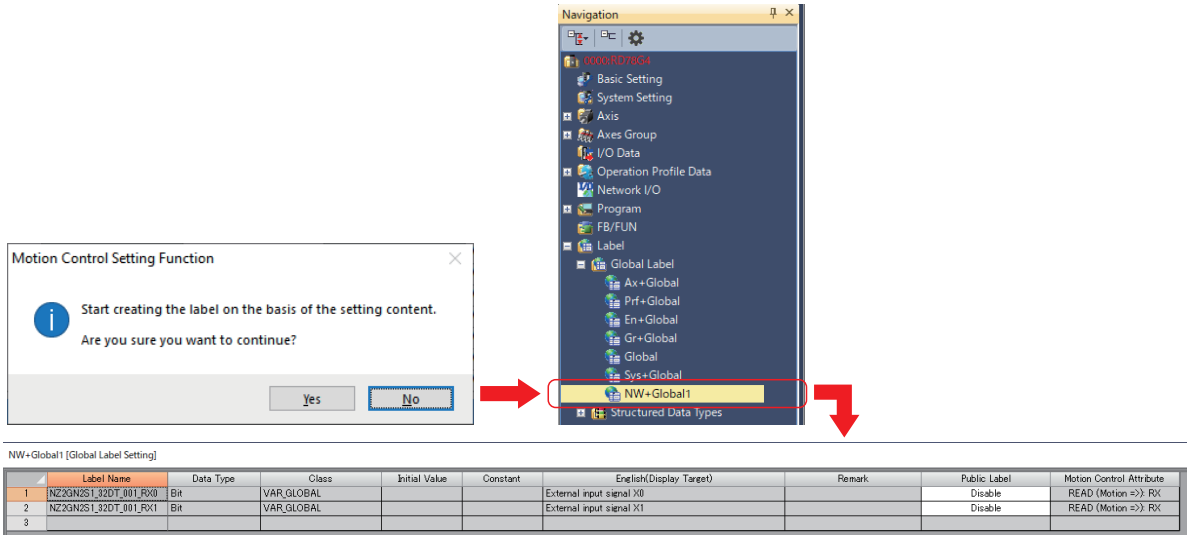
■Parameters of the Motion module

- Setting of the network I/O

1. Double-click "Network I/O" in the navigation window of Motion Control Setting Function. On the displayed "Network I/O" screen, label the input signal of the remote input module. Place a check in the "Labeling Target" checkbox of "RX0" and "RX1", then click the [Create Label] button.

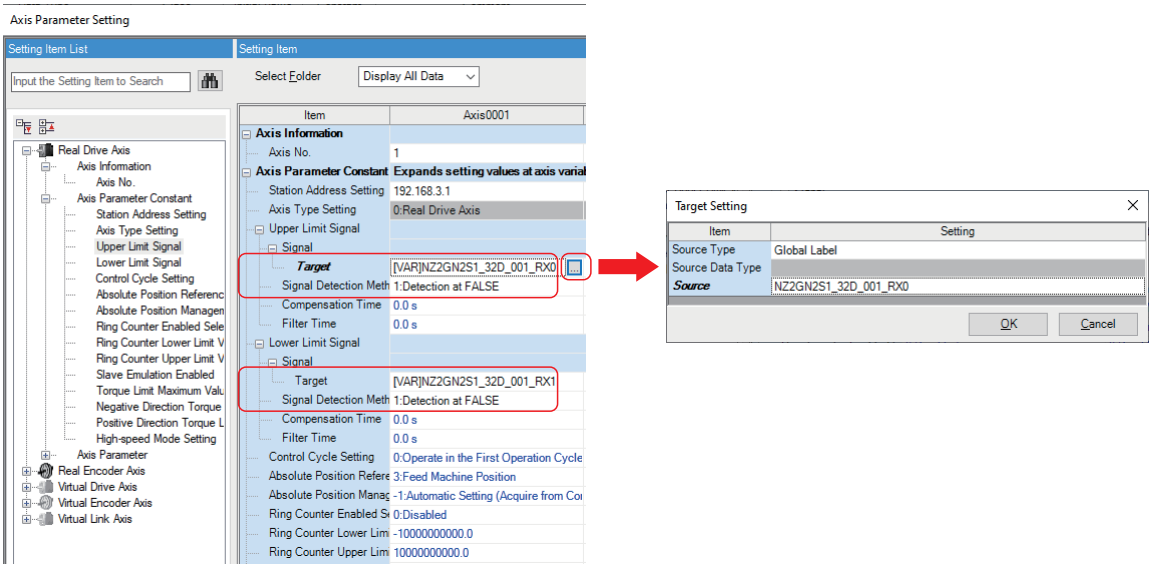


2. After the label is generated, "NW+Global1" will be added in "Global Label" in the navigation window. The network I/O label is generated and stored in "NW+Global1".



• Axis parameters

In the "Axis Parameter Setting" window, set the target of "Upper Limit Signal" and "Lower Limit Signal" as follows:



Perform the settings shown above for all axes that use hardware stroke limit signals.

In addition, change the signal detection method to "1: Detection at FALSE" because the normally closed contact is used.

■Operation check

Write the program and check whether ON/OFF of each signal can be monitored with "Upper limit signal status" and "Lower limit signal status" of the axis monitor.

To restore the value within the limit range, perform an error reset once, then move the axis to the direction within the range with JOG operation, etc.

Homing with the DOG signal

In this example, "RX2" is used as a DOG signal.

■Servo parameters

Set the homing method and the polarity of the proximity dog signal with the following parameters.

No.	Name
Pr. PT29.0	Device input polarity 1
Pr. PT45	Homing method

Set parameters from the following table as necessary according to the homing method.

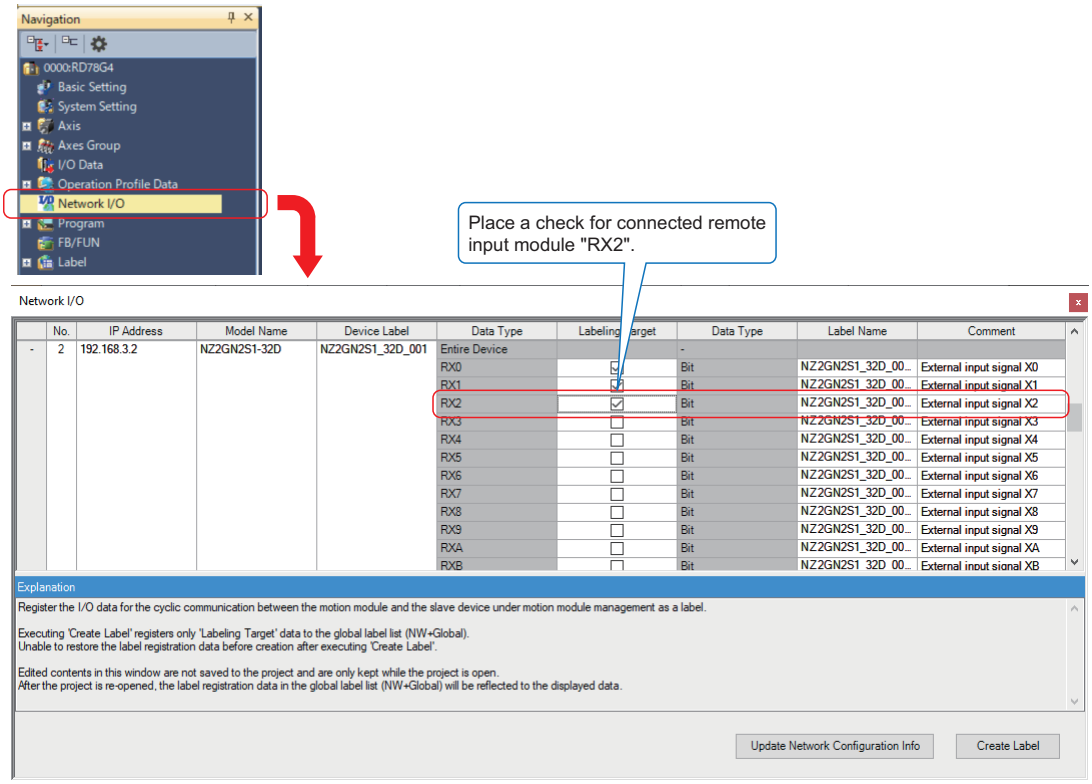
(The parameters that require setting vary by the homing method.)

No.	Name
Pr. PT05	Homing speed
Pr. PT06	Creep speed
Pr. PT07	Home position shift distance
Pr. PT08	Homing position data
Pr. PT09	Travel distance after proximity dog
Pr. PT55.0	Homing deceleration time constant selection
Pr. PT56	Homing acceleration time constant
Pr. PT57	Homing deceleration time constant

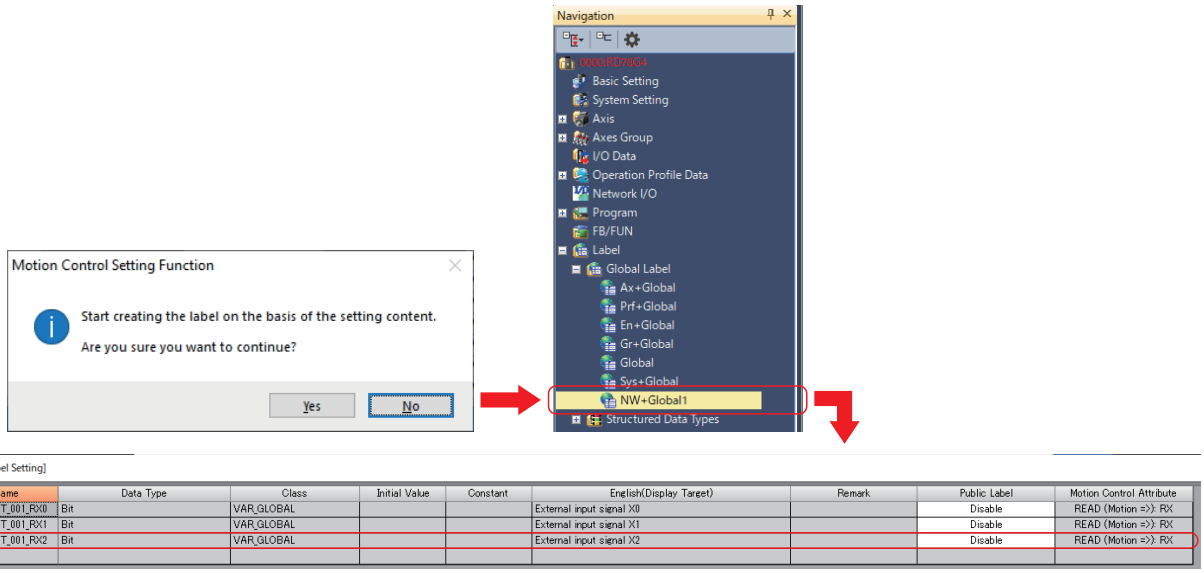
■Parameters of the Motion module

- Setting of the network I/O

1. Double-click "Network I/O" in the navigation window of Motion Control Setting Function. On the displayed "Network I/ O" screen, label the input signal of the remote input module. Place a check in the "Labeling Target" checkbox of "RX2", then click the [Create Label] button.



2. After the label is generated, "NW+Global1" will be added in "Global Label" in the navigation window. The network I/O label is generated and stored in "NW+Global1".

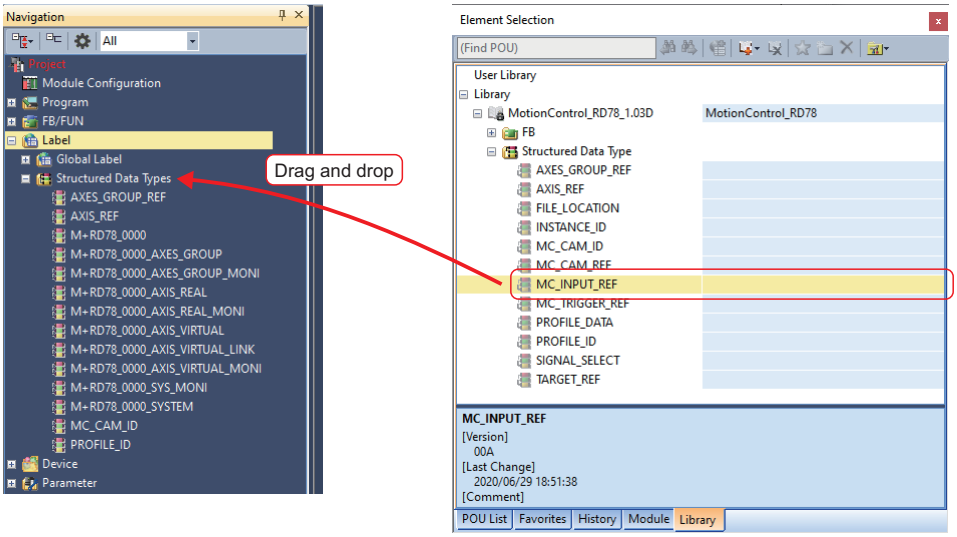


■Structures

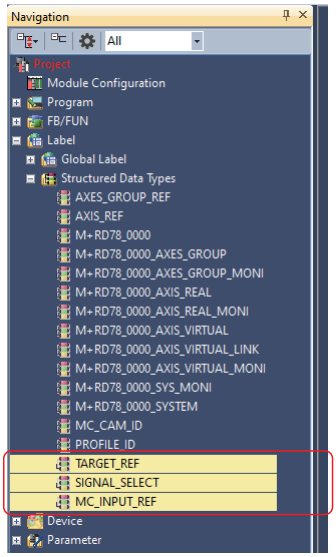
It is necessary to define an MC_INPUT_REF type structure to the PLC CPU.

1. Click the [Library] tab in the element selection window on MELSOFT GX Works3, then select "Library"⇨"MotionControl_RD78_****[*1]"⇨"Structured Data Type" to display the list of structured data.
Select "MC_INPUT_REF", then drag and drop it onto "Label"⇨"Structured Data Types" in the navigation window.

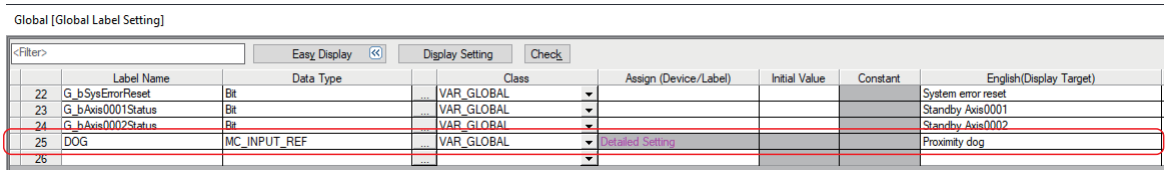
*1 **** = Motion module FB library version



2. "MC_INPUT_REF" and the structures of the members ("TARGET_REF" and "SIGNAL_SELECT") are registered.



3. Prepare an MC_INPUT_REF type structure in the labels of the PLC CPU.
The label can be registered as either a global or public label. The label name is "DOG" in this example.

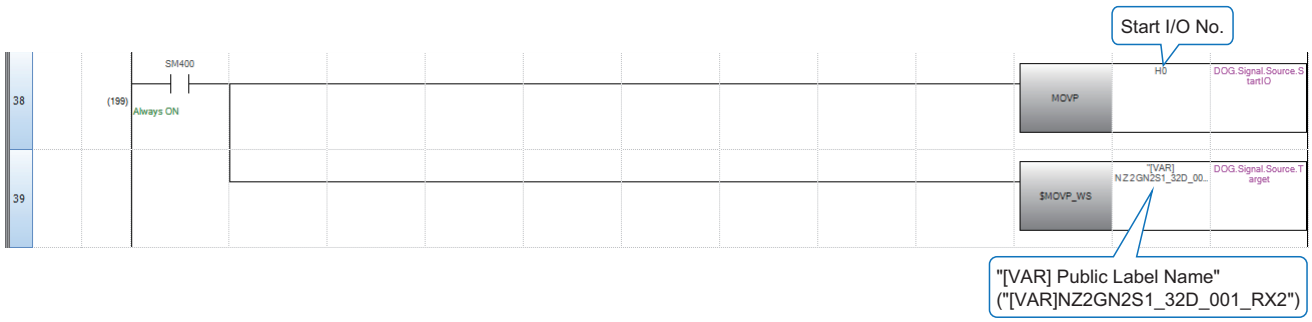


A

■Program

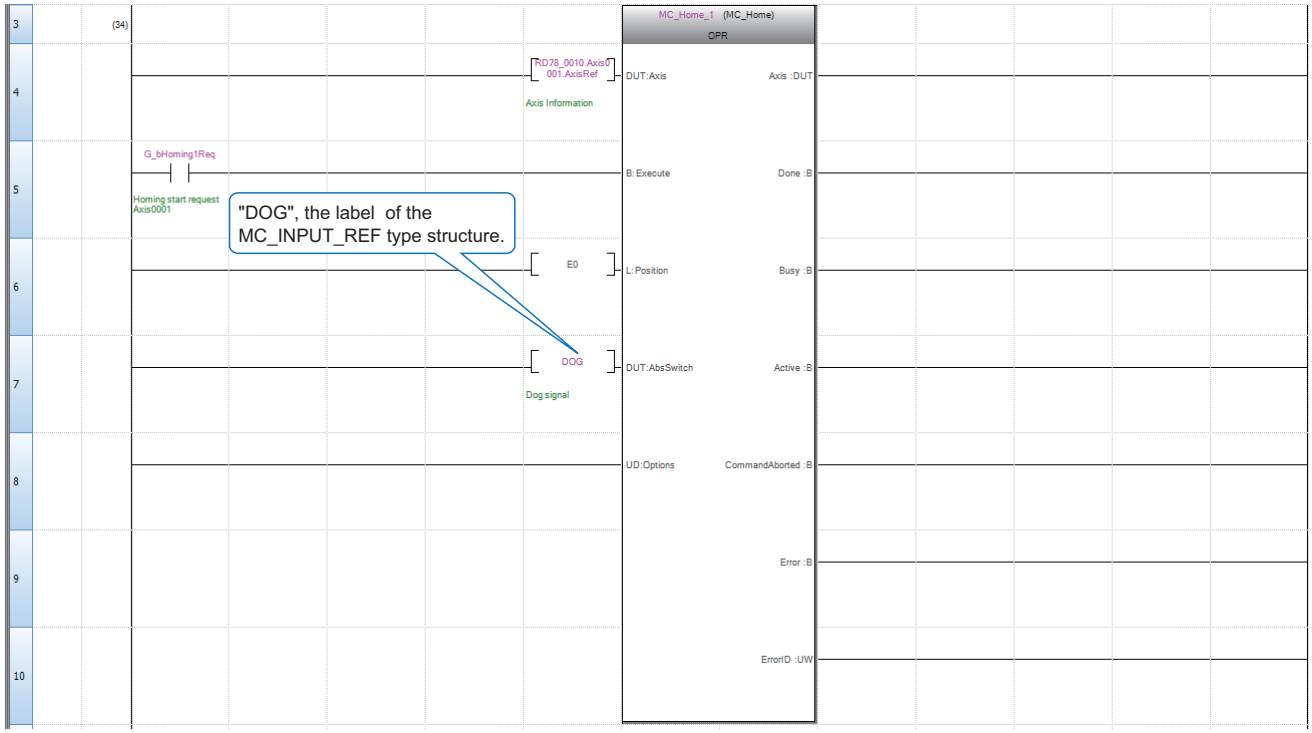
1. Set the members of the MC_INPUT_REF type structure with the program.

- Set the start I/O number of the Motion module in "DOG.Signal.Source.StartIO".
  Set the label name of "RX2" with a Unicode character string in "DOG.Signal.Source.Target".



2. Write MC_Home to the program.

- Enter the label "DOG" (the MC_INPUT_REF type structure set above) in the AbsSwitch input of MC_Home.



■Operation check

Write the program, then make sure that homing is correctly executed with the specified homing method.

Appendix 3 Precautions When the Absolute Position Detection System is Used

When the absolute position detection system is used, be careful of the following items.

Setting of the servo parameters

Change the following two servo parameters.

No.	Name	Setting value
PA03.0	Absolute position detection system selection	"0: Disabled (incremental system)" → "1: Enabled (absolute position detection system)"
PC29.5	[AL. 0E3 Absolute position counter warning] selection	"1: Enabled" → "0: Disabled"

At the first power-on

After the absolute position detection system is enabled, [AL. 025.1 Servo motor encoder absolute position erased] occurs on the servo amplifier at the first power-on.

Wait for 5 s as the alarm occurs, then cycle the power.

After that, execute homing.

REVISIONS

*The manual number is given on the bottom left of the back cover.

Revision date	*Manual number	Description
September 2021	L(NA)03194ENG-A	First edition

Japanese manual number: L03193-A

Mitsubishi Electric Servo System Motion Module Quick Start Guide [PLC CPU Ladder Program]

Country/Region	Sales office	
USA	Mitsubishi Electric Automation, Inc. 500 Corporate Woods Parkway, Vernon Hills, IL 60061, U.S.A.	Tel : +1-847-478-2100
Mexico	Mitsubishi Electric Automation, Inc. Mexico Branch Boulevard Miguel de Cervantes Saavedra 301, Torre Norte Piso 5, Ampliacion Granada, Miguel Hidalgo, Ciudad de Mexico, Mexico, C.P.11520	Tel : +52-55-3067-7512
Brazil	Mitsubishi Electric do Brasil Comercio e Servicos Ltda. Avenida Adelino Cardana, 293, 21 andar, Bethaville, Barueri SP, Brazil	Tel : +55-11-4689-3000
Germany	Mitsubishi Electric Europe B.V. German Branch Mitsubishi-Electric-Platz 1, 40882 Ratingen, Germany	Tel : +49-2102-486-0
UK	Mitsubishi Electric Europe B.V. UK Branch Travellers Lane, UK-Hatfield, Hertfordshire, AL10 8XB, U.K.	Tel : +44-1707-28-8780
Italy	Mitsubishi Electric Europe B.V. Italian Branch Centro Direzionale Colleoni - Palazzo Sirio, Viale Colleoni 7, 20864 Agrate Brianza (MB), Italy	Tel : +39-039-60531
Spain	Mitsubishi Electric Europe B.V. Spanish Branch Carretera de Rubi, 76-80-Apdo. 420, E-08174 Sant Cugat del Valles (Barcelona), Spain	Tel : +34-935-65-3131
France	Mitsubishi Electric Europe B.V. French Branch 25, Boulevard des Bouvets, 92741 Nanterre Cedex, France	Tel : +33-1-55-68-55-68
Czech Republic	Mitsubishi Electric Europe B.V. Czech Branch, Prague Office Pekarska 621/7, 155 00 Praha 5, Czech Republic	Tel : +420-255-719-200
Poland	Mitsubishi Electric Europe B.V. Polish Branch ul. Krakowska 48, 32-083 Balice, Poland	Tel : +48-12-347-65-00
Russia	Mitsubishi Electric (Russia) LLC St. Petersburg Branch Startovaya street, 8, BC "Aeroplaza", office 607; 196210, St. Petersburg, Russia	Tel : +7-812-449-51-34
Sweden	Mitsubishi Electric Europe B.V. (Scandinavia) Hedvig Mollersgata 6, 223 55 Lund, Sweden	Tel : +46-8-625-10-00
Turkey	Mitsubishi Electric Turkey A.S. Umraniye Branch Serifali Mah. Kale Sok. No:41 34775 Umraniye - Istanbul, Turkey	Tel : +90-216-969-2500
UAE	Mitsubishi Electric Europe B.V. Dubai Branch Dubai Silicon Oasis, P.O.BOX 341241, Dubai, U.A.E.	Tel : +971-4-3724716
South Africa	Adroit Technologies 20 Waterford Office Park, 189 Witkoppen Road, Fourways, South Africa	Tel : +27-11-658-8100
China	Mitsubishi Electric Automation (China) Ltd. Mitsubishi Electric Automation Center, No.1386 Hongqiao Road, Shanghai, China	Tel : +86-21-2322-3030
Taiwan	SETSUYO ENTERPRISE CO., LTD. 6F, No.105, Wugong 3rd Road, Wugu District, New Taipei City 24889, Taiwan	Tel : +886-2-2299-2499
Korea	Mitsubishi Electric Automation Korea Co., Ltd. 7F to 9F, Gangseo Hangang Xi-tower A, 401, Yangcheon-ro, Gangseo-Gu, Seoul 07528, Korea	Tel : +82-2-3660-9529
Singapore	Mitsubishi Electric Asia Pte. Ltd. 307 Alexandra Road, Mitsubishi Electric Building, Singapore 159943	Tel : +65-6473-2486
Thailand	Mitsubishi Electric Factory Automation (Thailand) Co., Ltd. True Digital Park Building Sukhumvit 101 Sukhumvit Road, Bang Chak, Prakanong, Bangkok, Thailand	Tel : +66-2092-8600
Indonesia	PT. Mitsubishi Electric Indonesia Gedung Jaya 8th Floor, JL. MH. Thamrin No.12, Jakarta Pusat 10340, Indonesia	Tel : +62-21-3192-6461
Vietnam	Mitsubishi Electric Vietnam Company Limited 11th & 12th Floor, Viettel Tower B, 285 Cach Mang Thang 8 Street, Ward 12, District 10, Ho Chi Minh City, Vietnam	Tel : +84-28-3910-5945
India	Mitsubishi Electric India Pvt. Ltd. Pune Branch Emerald House, EL-3, J Block, M.I.D.C., Bhosari, Pune - 411026, Maharashtra, India	Tel : +91-20-2710-2000
Australia	Mitsubishi Electric Australia Pty. Ltd. 348 Victoria Road, P.O. Box 11, Rydalmere, N.S.W 2116, Australia	Tel : +61-2-9684-7777

MITSUBISHI ELECTRIC CORPORATION

HEAD OFFICE: TOKYO BUILDING, 2-7-3, MARUNOUCHI, CHIYODA-KU, TOKYO 100-8310, JAPAN
NAGOYA WORKS: 1-14, YADA-MINAMI 5, HIGASHI-KU, NAGOYA, JAPAN

L(NA)03194ENG-A 2109 (IP)

New publication, effective September 2021