

Mitsubishi Electric Servo System Motion Module Quick Start Guide

Let's Start! Quick Start Guide

MELSEC iQ-R Series Motion Module







Applicable Model

- -RD78G
- -RD78GH
- -MR-J5-G



(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: "A WARNING" and "A CAUTION".



Indicates that incorrect handling may cause hazardous conditions, resulting in death or severe injury.

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 "ACAUTION" 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]

MWARNING

- 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.

[Design Precautions]

≜WARNING

- 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 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]

- 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]

- 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]

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.

REVISIONS

Revision Date	Manual No.	Description
Oct. 2020	L(NA)03191ENG-A	First edition

This manual confers no industrial property rights or any rights of any other kind, nor does it confer any patent licenses. Mitsubishi Electric Corporation cannot be held responsible for any problems involving industrial property rights which may occur as a result of using the contents noted in this manual.

© 2020 MITSUBISHI ELECTRIC CORPORATION

INTRODUCTION

Please read this manual carefully so that equipment is used to its optimum.

CONTENTS

SAFETY PRECAUTIONS	A-	1
REVISIONS	A-	10
CONTENTS	A-	11

1. OVERVIEW

1.1 What Is A Servo System?	
1.2 Application Examples of Servo Systems	
1.3 Role of Servo System Controllers	
1.4 Features of MELSEC iQ-R Series Motion Module RD78G(H)	
1.5 Programming Methods	
1.6 Organization of This Document	
1.7 Relevant Manuals	

2. SYSTEM STARTUP

2.1 System Overview	. 2-	1
2.2 System Configuration	2-	2
2.3 System Components Preparation	2-	3
2.4 Firmware/Software Installation	2-	4
2.4.1 Firmware/software preparation	2-	4
2.4.2 MELSOFT GX Works3 installation	2-	5
2.4.3 Profile registration	2-	6
2.4.4 MELSOFT iQ-R Series Motion Module FB library	2-	7
2.5 Module Installation	2-	8
2.6 Wiring and Cable Connection	2-	9

3. PARAMETER SETTING

3- 1 to 3-38

1- 1 to 1- 6

2- 1 to 2-12

3.1 Parameter Setting Procedure	
3.2 Creating a Project	
3.3 Connecting a Personal Computer and a PLC CPU Module	
3.4 Initial Setting	
3.4.1 Initializing a PLC CPU module	
3.4.2 Checking firmware version	
3.5 Creating a Module Configuration Diagram	
3.6 Network Configuration	
3.6.1 Setting a network configuration	
3.6.2 PDO mapping	
3.6.3 Setting a motion control station	
3.7 Servo Parameter Setting	
3.8 Module Parameter Application	
3.9 Motion Control Setting	
3.10 Basic Setting and System Setting	3-16

3.11 Axis Parameter Setting	
3.11.1 Setting a station address and an axis type	
3.11.2 Setting each item	
3.11.3 Driver unit conversion (electronic gear)	
3.11.4 Reflecting axis parameters	
3.12 Axes Group Setting	
3.13 Labels	
3.13.1 Creating a label	
3.14 Public Labels	
3.14.1 Public label registration	
3.14.2 Public label reflection	
3.14.3 A program example using public labels	
3.15 Writing to a Motion Module	
3.16 Saving a Project	
3.17 Parameter List	

4. PROGRAMMING BY A PLC CPU ONLY

4- 1 to 4-56

4.1 Programming Procedure for PLC CPU	
4.1.1 Creating a program block	
4.1.2 Program execution type	
4.1.3 Inputting a FB	
4.2 Naming Rules of Labels	4-11
4.3 Projects	4-12
4.3.1 Program names	4-12
4.3.2 Settings for global and public labels	4-13
4.4 PLC READY (Program Name: ServoON_Jog)	4-17
4.5 Servo ON (Program Name: ServoON_Jog)	4-18
4.6 JOG Operation (Program Name: ServoON_Jog)	4-19
4.7 Homing (Program Name: Homing)	4-24
4.8 Single Axis Positioning Control (Program Name: Positioning)	4-27
4.9 Single Axis Continuous Positioning (Program Name: ContinuousPositioning)	
4.10 Interpolation Control (Program Name: LinearInterpolation)	
4.10.1 Procedure for interpolation control	
4.10.2 Enabling/disabling an axes group	
4.10.3 Interpolation control	
4.10.4 Program example for linear interpolation	
4.11 Synchronous Control (Program Name: Synchronous)	
4.11.1 Procedures for executing synchronous control	
4.11.2 Calculation profile	4-40
4.11.3 Single axis synchronous control FBs	
4.11.4 Axes configuration	
4.11.5 Program example	
4.12 Error Reset (Program Name: ErrorReset)	
4.13 Checking Operation	4-49
4.13.1 Conversion and Writing of Programs	
4.13.2 Axis monitor	
4.13.3 Program monitor	4-52
4.13.4 Watch	4-53
4.13.5 Event History	

5. PROGRAMMING BY A PLC CPU AND THE MOTION MODULE	5- 1 to 5-56
5.1 Programming Procedure for Motion Module	
5.1.1 Creating a program block	
5.1.2 Program execution type	
5.1.3 Inputting a FB	
5.1.4 ENUM enumerator	5-10
5.1.5 Conversion of all the programs and reflection of public labels	5-11
5.2 Naming Rules of Labels	5-12
5.3 Projects	5-13
5.3.1 Program names	5-13
5.3.2 Settings for global and public labels	5-15
5.4 PLC READY (Program Name: ServoON_Jog)	5-17
5.5 Servo ON (Program Name: ServoON_Jog)	5-18
5.6 JOG Operation (Program Name: ServoON_Jog)	5-19
5.7 Homing (Program Name: Homing)	5-24
5.8 Single Axis Positioning Control (Program Name: Positioning)	5-27
5.9 Single Axis Continuous Positioning (Program Name: ContinuousPositioning)	5-29
5.10 Interpolation Control (Program Name: LinearInterpolation)	5-33
5.10.1 Procedure for interpolation control	
5.10.2 Enabling/disabling an axes group	5-33
5.10.3 Interpolation control	
5.10.4 Program example for linear interpolation	
5.11 Synchronous Control (Program Name: Synchronous)	
5.11.1 Procedures for executing synchronous control	
5.11.2 Calculation profile	
5.11.3 Single axis synchronous control FBs	
5.11.4 Axes configuration	
5.11.5 Program example	
5.12 Error Reset (Program Name: ErrorReset)	
5.13 Checking Operation	
5.13.1 Conversion and Writing of Programs	
5.13.2 Axis monitor	5-51
5.13.3 Program monitor	
5.13.4 Watch	
5.13.5 Event History	

APPENDICES

APP-1 to APP-34

APPENDIX 1 Diverting Programs	APP-	1
APPENDIX 1.1 Exporting the Motion module data	APP-	1
APPENDIX 1.2 Saving servo parameters	APP-	2
APPENDIX 1.3 Reflection to the editing program	APP-	3
APPENDIX 1.4 Conversion of All the Programs and Reflection of Public Labels	APP-	7
APPENDIX 2 Using External Input Signals	APP-	8
APPENDIX 2.1 Using the DI signal of the servo amplifier	APP-	8
APPENDIX 2.2 Using the input signal of the PLC CPU	APP-1	16
APPENDIX 2.3 Using the input signal of the remote input module	APP-2	23
APPENDIX 3 Precautions When the Absolute Position Detection is Used	APP-3	30
APPENDIX 4 Program Example in Ladder	APP-3	31

MEMO

1. OVERVIEW

This document is intended for first-time users of Mitsubishi Electric's servo system controller RD78G(H) to learn about necessary system components, software, parameter settings, and programming methods to configure a servo system.

1.1 What Is A Servo System?

A servo system controls an object following the input command values, such as the position, direction, attitude, etc. of the controlled object.

The Mitsubishi Electric's servo system, as shown below, controls a servo motor by transmitting position and speed commands from a controller (a programmable controller and a servo system controller) to a servo amplifier.

An encoder is mounted on the servo motor to detect the position and speed. The detected data is fed back to the servo amplifier so that the deviation between the position/speed commands and the actual position/speed is minimized.



1.2 Application Examples of Servo Systems

A servo system is used in a wide range of machines such as ones that require high-accuracy positioning, ones that require accurate synchronization such as packing machines and printing machines, and ones that require tension control (torque control) and speed control such as unwinding/winding machines, etc.



[A machine that requires position control] [A machine that requires synchronous control] [A machine that requires torque control]

1.3 Role of Servo System Controllers

The servo system controller outputs position/speed commands to servo amplifiers and generates commands to execute interpolation and synchronous control using multiple axes. In addition, the controller changes the position/speed commands according to the value of the sensors and other devices controlled by a programmable controller.

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

MELSEC iQ-R series Motion module RD78G(H) is a servo system controller supporting CC-Link IE TSN that enables mixing of real-time control communication required for motion control and non-real time information communication utilized by the IT systems. By supporting the network, the Motion module can flexibly connect various devices including servo amplifiers, I/O modules, and high-speed counters.

Main features of RD78G(H)

- · High-speed processing with multi-core processor
- Up to 256 control axes
- Programming with the PLCopen® Motion Control FB (function block) for positioning control, etc.



1.5 Programming Methods

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

Selectable programming languages vary depending on the controller:

- Motion module : structured text language (ST)
- PLC CPU
 - CPU : ladder diagram (Ladder), function block diagram/ladder diagram (FBD/LD), sequential function chart (SFC), and structured text language (ST) (These languages are compliant with IEC 61131-3.)



- (1) Programming by a PLC CPU only
 - Advantages : Easy program management by programming with only one CPU.
 - Disadvantages : Longer scan time due to processing by one CPU.
- (2) Programming by a PLC CPU and the Motion module
 - Advantages : Reduced scan time because motion control is programmed to the Motion module, not affecting other processing in a sequence program.
 - Disadvantages : Necessary to manage programs of both the PLC CPU and the Motion module.

1.6 Organization of This Document

This document describes procedures from startup to debugging by taking a two-axis XY table as an example.

The programming part consists of two chapters: programming by a PLC CPU only (Chapter 4), and programming by a PLC CPU and the Motion module (Chapter 5).



1.7 Relevant Manuals

(1) Motion module

Manual title	Manual No.
MELSEC iQ-R Motion Module User's Manual (Startup)	IB-0300406ENG
MELSEC iQ-R Motion Module User's Manual (Application)	IB-0300411ENG
MELSEC iQ-R Motion Module User's Manual (Network)	IB-0300426ENG
MELSEC iQ-R Programming Manual (Motion Control Function Blocks)	IB-0300533ENG
MELSEC iQ-R Programming Manual (Motion Module Instructions, Standard Functions/Function Blocks)	IB-0300431ENG

(2) Programmable controller

Manual title	Manual No.
MELSEC iQ-R CPU Module User's Manual (Startup)	SH-081263ENG
MELSEC iQ-R CPU Module User's Manual (Application)	SH-081264ENG
MELSEC iQ-R Ethernet/CC-Link IE User's Manual (Startup)	SH-081256ENG
MELSEC iQ-R Ethernet/CC-Link IE User's Manual (Application)	SH-081257ENG
MELSEC iQ-R Programming Manual (Program Design)	SH-081265ENG
MELSEC iQ-R Structured Text (ST) Programming Guide Book	SH-081483ENG
MELSEC iQ-R Programming Manual (CPU Module Instructions, Standard Functions/Function Blocks)	SH-081266ENG

(3) Servo amplifier

Manual title	Manual No.
MR-J5 User's Manual (Hardware)	SH-030298ENG
MR-J5-G/MR-J5W-G User's Manual (Introduction)	SH-030294ENG
MR-J5-G/MR-J5W-G User's Manual (Parameters)	SH-030308ENG
MR-J5 User's Manual (Function)	SH-030300ENG
MR-J5 User's Manual (Communication Function)	SH-030302ENG
MR-J5 User's Manual (Trouble Shooting)	SH-030312ENG
MR-J5 User's Manual (Object Dictionary)	SH-030304ENG

MEMO

2. SYSTEM STARTUP

2.1 System Overview

The following shows an example of a two-axis machine that uses ball screws.

(1) Machine



(2) Specifications	
	Lead of the ball screw (PB)	: 10.0 [mm]
	Reduction ratio (NL/NM)	: 1/2 (Load side [NL]/Motor side [NM])
		When 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 using 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 using the following components. Prepare modules, devices, cables, and software according to the user's system.

Motion module	Engineering environment	
RD78G4	MELSOFT GX Works3 Engineering environment Programmable Controller Engineering Software MELSOFT GX Works3	
Servo amplifier	Servo motor	
MR-J5-10G	НК-КТ13W	
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)
		T-T

2.4 Firmware/Software Installation

2.4.1 Firmware/software preparation

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

The screen displays may differ depending on the version of the firmware/software to be used. Please contact your local sales office for the latest version.

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

[Firmware/software]

(Note-1): This version can be installed to a controller with ver. 23 or later.

[Engineering environment]

Item	Description	Version
MELSOFT GX Works3	PLC engineering software	1.065T or later
Motion control setting	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 to display and analyze large-capacity data collected by the logging function of a module	1.106K or later

[Profile]

Item	Description	Version
Profile for MR-J5-G	Profiles for MELSERVO-J5 series CC-Link IE TSN-compatible servo amplifier	02 or later

[FB library] *Use it when programming with a PLC CPU.

Item	Description	Version
MELSEC iQ-R Motion module FB library	An FB library for the systems using the MELSEC iQ-R series Motion module	01B or later

2.4.2 MELSOFT GX Works3 installation

- (1) Check before installation
 - 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.
- (2) Installation procedure
 - 1) Insert the MELSOFT GX Works3 DVD-ROM to the DVD-ROM drive, and double-click "setup.exe" in the Disk1 folder on the DVD-ROM.
 - 2) Select or enter the necessary information by following the on-screen instructions. (Note-1)

[POINTS]

Motion control setting and GX LogViewer can be installed from the MELSOFT GX Works3 DVD, but they are not included in MELSOFT GX Works3 update module.

When updating MELSOFT GX Works3 by using the update module, also update Motion control setting and GX LogViewer separately.

Please contact your local sales office for the update module.

⁽Note-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.

2.4.3 Profile registration

Register the profiles for the devices such as a servo amplifier and a remote I/O module to be used. (The profiles for MR-J5-G/MR-J5W_-G servo amplifiers are automatically registered when MELSOFT GX Works3 is installed.)

(1) Downloading the profiles

Please contact your local sales office for downloading the profiles.

(2) Installing the profiles

After unzipping the downloaded file in any location, register a CSP+ file (zip file).

1) Start MELSOFT GX Works3, click [Tool] → [Profile Management] → [Register], and display the "Register Profile" screen.

Select the file of the language to be used from the unzipped profile, and click the [Register] button.



2.4.4 MELSOFT iQ-R Series Motion Module FB library

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

- (1) Downloading the Motion module FBPlease contact your local sales office for downloading the Motion module FBs.
- (2) Registration procedure of the Motion module FB library1) Start MELSOFT GX Works3.
 - 2) Open any project.

Select [Project] \rightarrow [Library Operation] \rightarrow [Register to Library List] \rightarrow [Library], and then click the [OK] button on the confirmation screen.



 On the "Register Library to Library List" screen, select [MotionControl_****.mslm], and click the [Open] button.

Make sure that the Motion module FB is registered in the [Library] tab of the element selection window.



2.5 Module Installation

(1) Installing a battery

The Q6BAT installed in the CPU module is shipped with the connector disconnected. Connect the connector with 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, and insert it into the connector pins on the cover. Firmly push the connector all the way.
- 4) Close the bottom cover.
- (2) Installing the modules

Install each module to 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 size is applicable when the MR-J5-10G servo amplifier is used.

If the servo amplifiers with other capacities are used, refer to the relevant user's manual.

(1) Wiring the power supply module

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

Item	Applicable wire size
Power wires	AWG 18 to AWG 14
Grounding wires	AWG 18 to AWG 14

(2) 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)	1.25 mm ² to 2 mm ² (AWG 16 to AWG 14)
Main circuit power supply (L1, L2, and L3)	2 mm² (AWG 14)
Grounding wires	AWG 18 to AWG 14

(3) 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

(4) Connecting the cables

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

(5) Power-on of 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.

- Power supply module: LED (green) ON
- PLC CPU module: READY LED (green) ON
- The ERROR LED flashes red when parameters and programs are not written to the CPU module, but no immediate 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

When the 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.

(6) Power-on of the servo amplifier

Check the wiring of the servo amplifier, and 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.

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



(b) Alarm occurrence of a 1-axis servo amplifier



(c) 7-segment LED display during a network connection



(7) Status LEDs



Name	LED 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. 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, and select [Project] \rightarrow [NEW].



2) Select the series, the type, and the program language, and click the [OK] button.

New		\times
Series	📲 RCPU	\sim
<u>T</u> ype	12 R04	\sim
Mode		\sim
Program Language	Do not Specify	\sim
	OK Cancel	

[Setting example] Series: RCPU Type: Select the PLC CPU to be used Program language: Select the language to be used

3) Read the displayed message, and click the [OK] button.

MELSOFT GX Works3	
Add a module. [Module Name] R04CPU [Start I/O No.] 3E00	
Module Setting	Setting Change
Module Label:Not use Sample Comment:Use	^
	~
Do Not Show this Dialog Again	ОК
3.3 Connecting a Personal Computer and a PLC CPU Module

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

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



2) Select [Online] → [Current Connection Destination] to open the "Specify Connection Destination" screen.

Click the [CPU Module Direct Coupled Setting] button, and set the connection destination for the personal computer.



3) Select the connection method to be used, and click the [Yes] button to return to the "Specify Connection Destination" screen.

CPU Module Direct Co	upled Setting	×
Please select the	direct connection method with CPU module.	
€ <u>U</u> SB		
○ <u>E</u> thernet		
<u>A</u> dapter	Not Specified	\sim
IP Address		
Current setting co continue?	ontent will be lost when new items are selected. Are you sure you want to)
	Yes <u>N</u> o	

When selecting "Ethernet", specify the adapter.

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

MELSOFT	GX Works3	×
1	Successfully connected with the R04CPU.	
	ОК	

3.4 Initial Setting

3.4.1 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]. Click the [Initialization] button on the [CPU Memory Operation] screen.

Onli	ne Debug Recording Diagnostics		CPU Memory Operation		×
	Current Connection Destination		Memory Management CPU Built-in Memory	CPU Built-in Memory	
2 10	Read from PLC		SD Memory Card	· · · · · · · · · · · · · · · · · · ·	
-	Write to PLC			Data Memory Use Volume	
	Verify with PLC			305/2049кВ	
	Remote Operation(S)			Device/Label Memory	
	Safety PLC Operation	•		File Storage Area Use Volume	
	Redundant PLC Operation(G)	•		64/256KB	
	CPU Memory Operation	ᡝ			
	Delete PLC Data				
	User Data	•			
	Set Clock				
	Monitor	•			
	FB Property Management (Online)				
	Watch	۲.		Detail Initialization(E) Gen Value Refresh(N)	
	User Authentication	•			Close

2) Click the [Yes] button on the confirmation screen to start initialization.

MELSOFT GX Works3 ×	MELSOFT GX Works3 X
Initialize the selected memory. Are you sure you want to continue? Each memory will be in a status as following after initialization.	Completed.
	ОК
<u>Y</u> es <u>N</u> o	

3.4.2 Checking firmware version

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

Select [Diagnostics] → [System Monitor] to open the "System Monitor Main Base (R35B)" screen.
 Click the [Product Information List] button to open the "Product Information List" screen.



2) Check the firmware version.

	Contro I CPU	Network Informatio n (Port 1)	Network Informatio n (Port 2)	IP Address (Port1 IPv4)	IP Address (Port2 IPv4)	Module Synchrono us Status	Firmware Version
Basic-Power Supply	-	-	-	-	-	-	-
Basic-CPU	-	-	-	192.168.3.39	-	-	46
Basic-I/O 0	-	1-0	-	192.168.3.253	-	-	040409
Basic-I/O 1	-	-	-	-	-	-	-
Basic-I/O 2	-	-	-	-	-	-	-
Basic-I/O 3	-	-	-	-	-	-	-
Basic-I/O 4	-	-	-	-	-	-	-

[Motion module version designation]



3.5 Creating a Module Configuration Diagram

Create a module configuration diagram.

 Double-click "Module Configuration" in the navigation window to open 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), and drag and drop them to the module configuration diagram.



2) Select [Edit] \rightarrow [Parameter] \rightarrow [Fix], and click the [Yes] button on the confirmation screen.



3) Click the [Yes] button on the confirmation screen. Read the displayed message, and click the [OK] button to fix the parameters (initial values) for each module.

MELSOFT	GX Works3	\times	MELSOFT G	GX Works3	
	For the control CPU to use CC-Link IE TSN module of host PLC, setting 'Extended Mode (iQ-R Series Mode)' is required for Link Direct Device Setting of CPU Parameter. Do you want to change the setting?	_	 1	Add a module. [Module Name] RD78G4 [Start I/O No.] 0000	
	Vac		Modul	le Setting	Setting Change
	Tes 10		Sam	ple Comment:Use	^
					~
			<u>D</u> o N	lot Show this Dialog Again	ОК

3.6 Network Configuration

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

3.6.1 Setting a network configuration

 Double-click "Module Parameter (Network)" of the Motion module in the navigation window. Set "Required Settings" → "Network No." and "IP Address" in the parameter editor (Module Parameter).

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

Navigation P ×	0000:RD78G4 Module Parameter		×
	Setting Item List	Setting Item	
 Module Configuration Program FB/FUN Label Device System Parameter System Parameter R04CPU Module Information Module Parameter (Motion) Module Parameter (Notion) Module Parameter (Network) Module Parameter (Network) Module Password 	Setting Item List	Item Station Type Station Type Network No. Station No /IP Address Settine Station No. IP Address IP Address Station No. Default Gateway	Setting Master Station 1 0 192.168 . 3.253
		Set the station type.	^
	Item List Find Result	Check_ Restore	the Default Settings

2) Double-click "Basic Settings" → "Network Configuration Settings" → "<Detailed Setting>" in the parameter editor (Module Parameter) to open the "CC-Link IE TSN Configuration" window.

0000:RD78G4 Module Parameter		×		
Setting Item List	Setting Item			
Transfer Contrine Transfer Consult	Item	Setting		
Input the Setting Item to Search	Network Configuration Settings			
	Network Configuration Settings	<detailed setting=""></detailed>		
	😑 Refresh Settings			
Bequired Settings	Refresh Settings	<detailed setting=""></detailed>		
Basic Settings	Network Topology			
Network Configuration Settings	Network Topology	Line/Star		
🔤 😋 Refresh Setting	Communication Period Setting			
Network Topology	Basic Period Setting			
Communication Period Setting	Setting in Units of 1us	Not Set		
Connection Device Information	Communication Period Interval Setting (Do not Set it in U	1000.00 us		
Slave Station Setting	Communication Period Interval Setting (Set it in Units of	1 1000.00 us		
H-m Application Settings	System Reservation Time	20.00 us		
	Cyclic Transmission Time	500.00 us		
	Transient Transmission Time	480.00 us		
	- Multiple Period Setting			
	Normal-Speed	×4		
	Low-Speed	×16		
	Connection Device Information			
	Authentication Class Setting	Authentication Class B Only		
	Slave Station Setting			
	Disconnection Detection Setting	4 times		
	Explanation			
	Set the number of device points and assignments of slave station	n to the master station.		
		×		
< >>	Check Restore the Default Settings	1		
Item List Find Result				
		Apply		

3) Select the devices to be used in the module list, and drag and drop them.

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 units of the servo amplifiers. Drag and drop MR-J5-G twice.



(Note): When adding devices which are not listed in the module list, register the profiles. (Refer to Section 2.4.3.)



3.6.2 PDO mapping

Set default data of the PDO mapping in a batch.

1) Select [CC-Link IE TSN Configuration] → [Batch Setting of PDO Mapping]. Click the [Yes] button on the confirmation screen to set the default pattern of the PDO mapping in a batch.



[POINTS]

PDO is an abbreviation for Process Data Object.

The PDO communication is equivalent to the conventional CC-Link cyclic communication.

The PDO mapping performs mapping (associating) of the data (object) which is sent/received between the controller and the slaves in cyclic communication (PDO communication).

3.6.3 Setting a motion control station

Select the checkboxes of "Motion Control Station" of the modules controlled by the Motion module such as servo amplifiers and I/O modules.

B (C-Link	IE TSN	I Configuration	n (Start I/O: 00)00)								×
; cc	-Link <u>I</u> E	TSN (Configuration	<u>E</u> dit <u>V</u> iew	Close	with Discardi <u>ng</u> the Setting	Close with <u>R</u> eflecti	ing the Setting					
	Conn	ected	l/Disconnected	d Module Det	Module List			×					
	Mode	Settin	ıg:	Onlin	е	<u>A</u> ssign	nment Method:			CC-Link IE TSN Selection Find	Module	My Favo	orites]
	Cyclic	Transr	mission Time (I	Min.):	-	us Comn	nunication Period	Interval (Min.)	-	記 원 階 記 ☆ 啓 ★			
		No.	Model	Name	STA#	Station Type	Motion Control	RX Setting	RY Setting				^
							Station	Points	Points	Transistor Output			
v		0	Host Station		0	Master Station				Analog Input			
	86	2	MR-15-G		2	Remote Station				Analog Output			
	Be	~	MIC-33-G	_		Kembee Station			_	General purpose Inver	ter		
										General-Purpose AC S	ervo		
										B. MR-J5-G	Single	Axis 0.1	to
				_						MR-JS-G-RJ	Single	Axis 0.1	to
	<		_						>	MR-J5W2-G	2-AXIS	Unificatio	oni
1			STA#1	STA#2						MR-JOWZ-G_B_AXIS	2-Axis	Unificatio	oni
										MR-JSW3-G RC Avie	3-Axis	Unificatio	on i
Host	Statio	n								I/O Combined	J ANIS	onincada	
ST St Lin	A#0 N ation tal STA e/Star	laster \#:2	MR-J5-G	MR-J5-G									•

3.7 Servo Parameter Setting

There are two methods for writing parameters to servo amplifiers.

Writing method	Advantage	Disadvantage
Writing to a PLC CPU by MELSOFT GX Works3	 Batch-management of parameters Setting parameters without connecting servo amplifiers 	 Taking time at the initial communication
Writing to servo amplifiers by 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 by MELSOFT GX Works3.

1) Select the checkboxes of "Parameter Automatic Setting" in the "CC-Link IE TSN Configuration" window to write parameters to the slave stations from the master station at the initial communication.

CC-Link IE TSN Configuration (Start I/O: 0000)									- 0	Х	
i co	CC-Link JE TSN Configuration Edit View Close with Discarding the Setting Close with Reflecting the Setting										
	Connected/Disconnected Module Detection Detailed Display										
	Mode 9	Settin	g:	Online	2	\sim	<u>A</u> ssignm	ent Method:	\sim		
	Cyclic 1	Transn	nission Time (I	Min.):	- US		Commun	nication Period Interval (Min.):	- US		
				Name	RWr Settin	g RWw Setting	Pa	rameter Automatic Setting	DDO Manaira Cattina	TD Address	
		NO.	Model	Name	Points	Points			PDO Mapping Setting	IP Addres	s
		0	Host Station							192.168.3.2	253
		1	MR-J5-G		2	4 20		<detail setting=""></detail>	<detail setting=""></detail>	192.168.3	.1
		2	MR-J5-G		2	4 20		<detail setting=""></detail>	<detail setting=""></detail>	192.168.3	.2
	<	-	_				_			_	>
		\square	STA#1	STA#2							
自局 ST S To Lir	TA#0 M tation otal STA ne/Star	laster #:2	MR-J5-G								>

2) Double-click the illustration of the servo amplifier to be set, and click the [Yes] button on the confirmation screen.



💶 MELSOFT GX Works3 - [Para	ameter Setting]							- 0	×
: Project View Eile Paramet	ter Setting(Z) Parameter	<u>M</u> indow <u>H</u> el	p						_ @ ×
i 🔏 i 😧 i 🐯 📷 i 🖏 🖿									
Project 🛛 🕂 🗙	Parameter Setting	J X							4 ♦ 🕶
	Station 1 X CR	ad 🔊 Set To	o Default	Serverify Deparameter Copy Parameter Block					
Station 1:MR-J5-G(-F					1				
Parameter	Save As	Copy	ste Mau	ndo 🚰 Redo					
Network Parame	😑 🚟 Function display (L	1/0				Coloris data and	A . / - 141 / //		
🖻 隆 Station2:MR-J5-G(-F	- Common	1/0				Selected Items write	Axis writing		
- Parameter	— Position/Speed/To	No.	Abbr.	Name	Unit	Setting range	Station 1	Station2	^^
Network Parame	 Servo adjustments 	Device setting	g				Setting	Setting	
	— I/O	PD03.0-1 *		Device selection DI1		00-7F	0A		0A
	 Servo amplifier dia 	PD04.0-1 *		Device selection DI2		00-7F	OB		OB
	 Machine diagnosis 	PD05.0-1 *		Device selection D13		00-7	22		22
	- Linear control	PD51.0-1 *		Device selection D13-2		00-7	62		62
	DD Motor control	PD38.0-1 *		Device selection D14		00-/F	20		20
	- Fully closed contro	PD39.0-1		Device selection DIS		00-7F	20		20
	🖬 🗐 list display	PD07.0-1 *		Device selection DO1		00-7F	03		04
	Basic	PD09.0-1 *		Device selection DO2		00-7F	03		03
	- Gain/filter	Device assign	ment				Setting	Setting	
	Extension	PD01.0-7 *		Input signal automatic on selection 1		2000000-0000EE0	0000000	00000	1000
	-1/0	Input filter	2101	Input agrial automatic on selection 1		500000000000000000000000000000000000000	0000000	00000	
	Extension 2	PD11.0 *		Input signal filter selection		0.8	7:3.500ms 🚽	7:3.500ms	
	Extension 2	ALM subsut		Input signal litter selection		0-0			<u> </u>
	- Extension 5	ALMOUDUL					0 + WNG cigoal turn (0 : WNG cional turn	
	- Option setting	PD14.1 *		Selection of output device at warning occurrence		0-1	0. WWG signal turrit 🗸	o . www.signar.com	<u> </u>
	- Special	Analog outpu	it						
	- Motor extension	Analog monit	or				00 - Come motor and	00 - Come motor of	
	- Multi encoder	PC09.0-1		Analog monitor 1 output selection		00-1F	00 : Servo motor spe 👻	00 : Servo motor sp	~
	 Positioning contro 	PC11 M	01	Analog monitor 1 offset		-999-999	0		0
	 Network setting 	PC10.0-1		Analog monitor 2 output selection		00-1F	01 : Torque or thrust	01 : Torque or thru:	st 👻
	Positioning extens	PC12 M	02	Analog monitor 2 offset		-999-999	0		0
		Stroke limit fu	unction						
		Stroke limit fu	unction						
		PC19.0 *		[AL. 099 Stroke limit warning] selection		0-1	1 : Disabled 🔹	1 : Disabled	-
		PD41.2 *		Limit switch enabled status selection		0-1	1: Only enabled ir 🔻	1 : Only enabled	ir 🔻
< >	<	PD41.3 *		Sensor input method selection		0-1	0 : Input from servo 👻	0 : Input from serve) <u> </u>
Ready	Uni	t connection					0	OVR CAP NUM S	CRL /

3) Set servo parameters on the "Parameter Setting" screen.

The example in this document changes the following parameters.

Operation	No.	ltem	Setting value
$\begin{array}{l} [Common] \rightarrow [Basic] \\ \rightarrow [Forced stop] \end{array}$	PA04.2	Servo forced stop selection	0: Enabled \rightarrow 1: Disabled
$[I/O] \rightarrow [Stroke limit function] \rightarrow [Stroke limit function]$	PC19.0	[AL. 099 Stroke limit warning] selection	0: Enabled \rightarrow 1: Disabled
	PD41.2	Limit switch enabled status selection	0: Always enabled → 1: Only enabled in home position return mode

4) Click the [×] button at the upper right in the window.

Click the [Yes] button on the confirmation screen to update the parameters.

MELSOF	T MR Configurator2	×
1	Update the slave parameter with the edited contents. Continue?	
	Yes <u>N</u> o Cancel	

5) Click the [Close with Reflecting the Setting] button to reflect the data.

— D	×
Module List	×
CC-Link IE TSN Selection Find Modu	ı
計 別 唱 📴 🔭 🗠 🗙 👘	
DC Input	^
Transistor Output	
Analog Input	
Analog Output	
General purpose Inverter	
General-Purpose AC Servo	
MR-JS-G Single Axis	30.
MR-15W2-C 2-Avis Unit	s U. fica
MR-15W2-G B 2-Axis Unit	fica
MR-J5W3-G 3-Axis Unif	lica
MR-J5W3-G_BC 3-Axis Unif	lica
I/O Combined	
	~
	×
	Module List CC-Link IE TSN Selection Find Modu DC Input Transistor Output Analog Input Analog Output General Purpose Inverter General Purpose Inverter General-Purpose AC Servo MR-J5-G Single Axis MR-J5-G-RJ Single Axis MR-J5-W2-G 2-Axis Unif MR-J5W2-G 2-Axis Unif MR-J5W2-G 2-Axis Unif MR-J5W2-G 2-Axis Unif MR-J5W2-G 2-Axis Unif MR-J5W2-G 2-Axis Unif

Although the warning window appears depending on the version, click the [Yes] button.



3.8 Module Parameter Application

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

0000:RD78G64 Module Parameter		x		
Setting Item List	Setting Item			
Tonut the Setting Item to Search	Item	Setting 🔨		
Input the Setting Item to Search	Network Configuration Settings			
	Network Configuration Settings	<detailed setting=""></detailed>		
	😑 Refresh Settings			
H- Bequired Settings	Refresh Settings	<detailed setting=""></detailed>		
Basic Settings	Network Topology			
Network Configuration Settings	Network Topology	Line/Star		
🔤 🗠 🗠 Refresh Setting	Communication Period Setting			
Network Topology	Basic Period Setting			
Communication Period Setting	Setting in Units of 1us	Not Set		
Connection Device Information	Communication Period Interval Setting (Do not Set it in Units of 1us)	1000.00 us		
Application Settings	 Communication Period Interval Setting (Set it in Units of 1us) 	1000.00 us		
	System Reservation Time	20.00 us		
	Cyclic Transmission Time	500.00 us		
	Transient Transmission Time	480.00 us		
	Multiple Period Setting			
	Normal-Speed	×4		
	Low-Speed	×16		
	Gonnection Device Information	¥		
	Explanation			
	Set the number of device points and assignments of slave station to the mast	ter station.		
Them Link Find Recult	Chec <u>k</u> Restore the Defa <u>u</u> lt Settings			
Item List Find result				
		Apply		
1				

[POINTS]

Note that changes in the parameters are not reflected unless clicking the [Apply] button.

3.9 Motion Control Setting

Set parameters and create programs for the Motion module on the motion control setting screen.

Double-click "Module Extended Parameter" of the Motion module in the navigation window to open the motion control setting function.

The set items are saved as a project of MELSOFT GX Works3.



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.

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



3.11 Axis Parameter Setting

3.11.1 Setting a station address and an axis type

Axes are classified into five types as follows.

- 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 FBs of the single axis synchronous control.

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

1) 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.

Right-click "Axis" in the navigation window of the motion control setting function, and select [Add New Data].

Enter the data name, the axis No., the station address setting, and the axis type setting as the following table shows, and click the [OK] button.



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

[POINTS]

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 a station address of the second or the third axis of a multi-axis servo amplifier is specified.

Example: For B-axis of MR-J5-W3 with an IP address of 192.168.3.1

<u>192.168.3.1#1</u> Multidrop No. (#0: A-axis, #1: B-axis, #2: C-axis) IP address

2) Virtual drive axis

Right-click "Axis" in the navigation window of the motion control setting function, and select [Add New Data].

Enter the data name, the axis No., and the axis type setting as the following table shows, and click the [OK] button.

Ne	w Data		Х
В	asic Setting		
D	Data Type	🐖 Axis	-
([Data Name)	VirtualAxis0001	
D	etailed Setting		
	Axis Information		
	Axis No.	301	
	Axis Parameter Constant		
	Axis Type Setting	Virtual Drive Axis	•
	Axis Type Setting Control Cycle Setting	Operate in the First Operation Cycle	• •
	Axis Type Setting Control Cycle Setting	Virtual Drive Axis Operate in the First Operation Cycle	• •
	Axis Type Setting Control Cycle Setting	Virtual Drive Axis Operate in the First Operation Cycle	•
	Axis Type Setting Control Cycle Setting	Virtual Drive Axis Operate in the First Operation Cycle	•
	Axis Type Setting Control Cycle Setting	Virtual Drive Axis Operate in the First Operation Cycle	•
	Axis Type Setting Control Cycle Setting	Virtual Drive Axis Operate in the First Operation Cycle	•
	Axis Type Setting Control Cycle Setting	Virtual Drive Axis Operate in the First Operation Cycle	•
	Axis Type Setting Control Cycle Setting	Virtual Drive Axis Operate in the First Operation Cycle	* *
	Axis lype Setting Control Cycle Setting	Virtual Drive Axis Operate in the First Operation Cycle	•
	Axis lype Setting Control Cycle Setting	Virtual Drive Axis Operate in the First Operation Cycle	•

Setting item	First axis
Date Name	VirtualAxis0001
Axis No.	301
Axis Type Setting	Virtual Drive Axis

3) Virtual linked axis

Right-click "Axis" in the navigation window of the motion control setting function, and select [Add New Data].

Enter the data name, the axis No., and the axis type setting as the following table shows, and click the [OK] button.

New Data		×
Basic Setting		
Data Type	💓 Axis	-
(Data Name)	LinkAxis0001	
Detailed Setting		
Axis Information		
Axis No.	401	
Axis Parameter Constant		
Axis Type Setting	Virtual Link Axis	•
Control Cycle Setting	Operate in the First Operat	tion Cycle 🔻
	ок с	ancel

Setting item	First axis	Second axis
Data Name	LinkAxis0001	LinkAxis0002
Axis No.	401	402
Axis Type Setting	Virtual Link Axis	Virtual Link Axis

3.11.2 Setting each item

- 1) Real drive axis
 - Double-click "Axis" \rightarrow "Axis0001" in the navigation window to open the "Axis Parameter Setting" window.



Item		Description	Axis0001 setting value	Axis0002 setting value
Arria	Station Address Setting	Set the IP address of the driver.	192.168.3.1	192.168.3.2
Axis Parameter Constant	Upper Limit Signal Lower 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 Appendix 2.)	
	Driver Unit Conversion Numerator	Convert units of the target position and the feedback position between the	6710	8864
	Driver Unit Conversion Denominator	controller and the driver. (Refer to Section 3.11.3 for the setting methods.)	5000	
	Stop Signal \rightarrow Signal \rightarrow Target	Set the stop signal.	[VAR]G_bStopSignalX (Note-1)	[VAR]G_bStopSignalX (Note-1)
	Forced Stop Signal, Stop Signal	Set the forced stop signal and the stop signal.	Initial value	
Axis Parameter	Start Permission at Home Position Return Uncompleted	Set whether axis start is allowed or not for when the homing request is true.	0: Not allowed	
	Software Stroke Limit Upper Limit Value/Lower Limit Value/Target	Set the range and the target of the software stroke limit	Initial value	
	Position Command Unit	Set the position command unit and the speed command unit used for the motion control.	um ^(Note-2)	
	Speed Command Unit	follows: [um] for the position command unit and [um/s] for the speed command unit.	U	/s

The setting example in this document is shown below.
--

(Note-1): The stop signal is registered as a global label. Refer to Section 4.3.2 or Section 5.3.2 for setting the global labels. (Note-2): "um" indicates micrometer.

2) Virtual drive axis

Double-click "Axis" \rightarrow "VirtualAxis0001" in the navigation window to open the "Axis Parameter Setting" window.

Navigation	Axis Parameter Setting		×
₽⊑- ₽⊏ ✿	Setting Item List	Setting Item	
60000:RD78G4 Basic Setting	Input the Setting Item to Search	Select <u>Fo</u> lder Display All Data ~	
System Setting		Item VirtualAxis0001	^
 Axis Axiso001 [X_Axis] Axis0002 [Y_Axis] VirtualAxis0001 [Vir_Axis01] LinkAxis0001 [Lin_Axis01] LinkAxis0002 [Lin_Axis02] Axes Group (O Data Calculation Profile Network I/O Program FB/FUN Label 		Axis Information Axis Information Axis Nameter Constant Expands setting values at an Axis Type Setting Withus Diversion Withus Diversion Withus Diversion Withus Diversion Withus Diversion Withus Diversion Signal Compensation Time 0.0 s Filter Time Opersation Time 0.0 s Filter Time Opersation Time 0.0 s Filter Time Operation Absolute Position Mana Objection Ring Counter Enabled S Disable Absolute Position Ring Counter Enabled S String Range] Ito 10000 Restore the Defagit Settings With Selected With Variables Reace	- Variables
		Ap	ply

The setting example in this document is shown below.

Item		Description	Setting value
	Position Command Unit	Set the position command unit and the speed command unit used for the motion control. In this example, the same unit as that of the real	um
	Speed Command Unit	drive axis is set for the virtual drive axis ([um] for the position command unit and [um/s] for the speed command unit).	U/s

3) Virtual linked axis

Double-click "Axis" \rightarrow "LinkAxis0001" in the navigation window to open the "Axis Parameter Setting" window.

Navigation	Axis Parameter Setting	×
	Setting Item List	Setting Item
10000:RD78G4 Basic Setting	Input the Setting Item to Search	Select Folder Display All Data
 Basic Setting System Setting System Setting Axis Axiso001 [X_Axis] Axis0002 [Y_Axis] VirtualAxis0001 [Vir_Axis01] LinkAxis0001 [Lin_Axis01] Chlexis0002 [Lin_Axis02] Axes Group (V Data Calculation Profile Network I/O Reprogram FB/FUN Label 	Real Drive Axis Real Encoder Axis Virtual Link Axis Virtual Link Axis Virtual Link Axis Axis Parameter Constant Axis Parameter Home Position Return Required Start Permission at Home Position Command Unit Position Command Unit Speed Command Unit	Item Link Axis 0001 Link Axis 0002 Axis Information
	< >> Item List Find Result	Restore the Default Settings With Selected Write Variables Read Variables
		Арріу

The setting example in this document is shown below.

Item		Description	Setting value
	Position Command Unit	Set the position command unit and the speed command unit used for the motion control. In this example, the same unit as that of the real	um
	Speed Command Unit	drive axis is set for the virtual linked axes ([um] for the position command unit and [um/s] for the speed command unit).	U/s

3.11.3 Driver unit conversion (electronic gear)

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

(1) About an electronic gear

[Ball screw]



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]) When the servo motor rotates twice, the ball screw of the load side rotates once.

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

Driver unit conversion numerator = Number of pulses per rotation 67,108,864 Driver unit conversion denominator = Movement amount per rotation 5000 (2) Electronic gear setting

The procedure of the electronic gear setting is shown below.

 Click the []] button in the "Axis Parameter" → "Driver Unit Conversion Numerator" or "Driver Unit Conversion Denominator" to open 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.
- 3) Click the [Calculate Axis Parameters] button.
- 4) Click the [OK] button to reflect the calculation results to the axis parameter.

3.11.4 Reflecting axis parameters

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

Axis Parameter Setting				×
Setting Item List	Setting Item			
Input the Setting Item to Search	Select <u>F</u> older Displ	ay All Data 🗸 🗸		
	Item	Axis0001	Axis0002	^
	Axis Information			
🖃 📲 Real Drive Axis	Axis No.	1	2	
- Axis Information	😑 Axis Parameter Constan	Expands setting values at a	xis variable initialization.Re-	-i
Axis No.	Station Address Setting	192.168.3.1	192.168.3.2	
Axis Parameter Constant	Axis Type Setting	0:Real Drive Axis	0:Real Drive Axis	
Station Address Setting				
Axis Type Setting				
Upper Limit Signal	Target			
Control Cycle Setting	Signal Detection Met	0:Detection at TRUE	0:Detection at TRUE	
Absolute Position Reference Se	Compensation Time	0.0 s	0.0 s	
Absolute Position Management	Filter Time	0.0 s	0.0 s	
Ring Counter Enabled Selection				
Ring Counter Lower Limit Value	- Signal			
Ring Counter Upper Limit Value	Target			
Slave Emulation Enabled	Signal Detection Met	0:Detection at TRUE	0:Detection at TRUE	
Iorque Limit Maximum Value	Compensation Time	0.0 s	0.0 s	
Positive Direction Torque Limit	Filter Time	0.0 s	0.0 s	
High-speed Mode Setting				
	Explanation			
🖅 🌒 Real Encoder Axis	Axis.PrConst.SlaveEmulate_E	inable ave emulation axis		^
😟 🗐 Virtual Drive Axis	If set enabled, it will operate	virtually assuming the slave	e station is connected even	in an
😟 🛷 Virtual Encoder Axis	unconnected state.			
⊕∰ Virtual Link Axis	[Setting Range] 0:Disabled 1:Enabled			~
Item List Find Result	Restore the Default Setting	s With Selec	ted Write Variables Re	ad Variables
			Арр	ly

3.12 Axes Group Setting

An axes group is used for multiple axes control such as linear interpolation and circular interpolation. An axes group is not used for synchronous control.

Right-click "Axes Group" in the navigation window of the motion control setting function, and select [Add New Data]. Set the data name and the axes group No., and click the [OK] button.

Navigation	т ×	New	/ Data		Х
□ 📴 - 🔤 🌣		Ba	isic Setting		
🚹 0000:RD78G4		Da	ata Type	豴 Axes Group	-
Basic Setting		(D	ata Name)	AxesGroup001	
System Setting				-	
🖽 🐖 Axis		De	etailed Setting		
🕀 🦓 Axes Group	Add New Data	▶	Axes Group Information		
🛛 🚺 🖉 I/O Data 📃 🖳	New Folder Ctrl+Shift+N		Axes Group No.	1	
E Calculation Pro					
	Sort •				
E CR/ELIN	Expand/Collapse Tree				
🖬 🛗 Label					
				OK Cancel	

The X axis (Axis0001) and Y axis (Axis0002) of the following two-axis machine example (refer to Section 2.1) are registered in an axes group.



The setting example of the axes group is shown below.

[Setting example of axes group structuring axes]

Axes Group Setting				x
Setting Item List	Setting Item			
Input the Setting Item to Search	Select <u>F</u> older Displ	ay All Data 🗸 🧹		
Axes Group Axes Group Information Axes Group Information Axes Group No Structurine Axis[1] Structurine Axis[1] Structurine Axis[10] Structurine Axis[10] Structurine Axis[10] Structurine Axis[11]	Item Axes Group No. Acceleration Limit Valu Operation Selection at: Structuring Axis[1] Structuring Axis[2] Structuring Axis[3] Structuring Axis[4] Structuring Axis[5] Structuring Axis[6] Structuring Axis[6] Structuring Axis[7] Structuring Axis[9] Structuring Axis[10] Structuring Axis[10] Structuring Axis[11] Structuring Axis[12] Structuring A	AxesGroup001 1 Expands initial values at ax 2147483647.0 um/s ⁻² - I:Error (Not Started) Axis0001 Axis0002		*
Structuring Axis[13]	Explanation			
Structuring Axis[14] Structuring Axis[15] Structuring Axis[16] Command In-position Deceleration Limit Va Jerk Limit Value ∨	Axes Group No. [Setting Range] 1 to 10000			<
Item List Find Result	Restore the Default Setting	s With Selecte	Write Variables Read Variables	
			Apply	

[Setting example of axes group units] (Position command unit: [um], Speed command unit: [U/s])

Axes Group Setting		x
Setting Item List	Setting Item	
Input the Setting Item to Search	Select <u>F</u> older Display All Data ~	
Big Big - Structuring Axis[1] - Structuring Axis[2] - Structuring Axis[3] - Structuring Axis[6] - Structuring Axis[6] - Structuring Axis[6] - Structuring Axis[6] - Structuring Axis[7] - Structuring Axis[8] - Structuring Axis[10] - Structuring Axis[11] - Structuring Axis[12] - Structuring Axis[13] - Structuring Axis[16] - Command In-position - Deceleration Limit Value - Operation Setting at (- Deceleration at Stop - Stop Selection at Stop - Stop Selection at Stop - Stop Selection at Stop -	Item AxesGroup001 Structuring Axis[13] Structuring Axis[14] Structuring Axis[15] Structuring Axis[15] Structuring Axis[16] Command In-position Width 100.0 um Deceleration Limit Value 2147483647.0 um/s ⁻² Jerk Limit Value 2147483647.0 um/s ⁻² Jerk Limit Value 2147483647.0 um/s ⁻² Structuring Axis Operation Setting at Overrul IImmediate Stop 0.0 um/s ⁻² Stop Selection at Stop 0.0 um/s ⁻² Stop Selection at Stop by F 3.Alternative Acceleration/D Position Command Unit Strum um Speed Command Unit Strum nm Speed Limit Value Revolution inch Arbitrary Unit	< >
Them List Find Result	Restore the Default Settings With Selected Write Variables Read Variables	
Rein List Production	Apply	

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

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 be used in only one program				
Global label	 Labels that can be used in all programs Labels that can be set to public labels (Refer to Section 3.14.) 				

3.13.1 Creating a label

This section describes how to create a label in the motion control setting function. Refer to Section 4.3.2 or Section 5.3.2 for registration examples of labels.

1) Double-click "Label" → "Global Label" → "Global" in the navigation window of the motion control setting function to open the "Global [Global Label Setting]" window.

Navigation 4	× Glob	Global [Global Label Setting]							
PE- 🌣			Label Name	Data 1	уре	Class		Initial Value	Constant
(1) 0000/RD78G4		1							
🤹 Basic Setting									
🛃 System Setting									
🗉 🐖 Axis	<								>
🗉 🎆 Axes Group		1							~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
👔 I/O Data	•								~
🗉 🥰 Calculation Profile			Label Name		Dat	a Type	Initial		
Wetwork I/O			Labor Hamo		but	u typo	Value		
🖽 🔚 Program									
📅 FB/FUN									
🔳 🛗 Label									
🗖 🕼 Global Label									
😭 Global	Ext	d Denl							
😭 Sys+Global		a bopi							
Ax+Global									

2) Set a global label.

- (a) Enter the label name. (Setting example: G_bSRVONCMD)
- (b) Select the data type. (Click the []] button to open the "Data Type Selection" window.) Select "Simple Types" in the type category, and select the data type.

Global	[Global Label Setting]				x
	Label Name	Data Type	Class	Initial Value	Constant
1	G_bSRVONCMD	Bit	. VAR_GLOBAL		
2		Data Type Selection			×
<	Label Name	Target(L) CALL> Type Category ③ Simple Types ③ Structured Data Type ③ Eunction Block Array Element □ ARRAY Element	Data Type Data Type Bit Word [Unsigned],Bit Double Word [Signed] Double Word [Signed] FLOAT [Double Preci Time String(22) String [Unicode](32) Timer Long Counter Long Counter Inne Retentive Timer Inne Retentive Timer Counter Long Retentive Timer Conter Long Retenti Long Retentive Timer Conter Long Retenti Long Re	String [16-bit] hed]/Bit String [32-b d] ion] ision] sion]	

Set the structured data type variables and the array variables in the same manner as Simple Types, and click the [OK] button.

Data Type Selection	×	Data Type Selection	Х
Target(L) <all> <project> Type Category Simple Types Structured Data Type Function Block</project></all>	Data Type AXES_GROUP_REF AXIS_ENCODER_OND AXIS_ENCODER_OND AXIS_ENCODER_PRM AXIS_ENCODER_PRM AXIS_ENCODER_PRM AXIS_REAL_OND AXIS_REAL_OND AXIS_REAL_PRM_CONST AXIS_REAL_PRM AXIS_REAL_PRM_CONST AXIS_REF AXIS_VIRTUAL AXIS_IREF AXIS_VIRTUAL_CMD AXI	Target(L)	Data Type Bit Word [Unsigned]/Bit String [16-bit] Double Word [Unsigned]/Bit String [32-b Word [Signed] Double Word [Signed] FLOAT [Single Precision] FLOAT [Double Precision] Time String(32) String [Unicode](32) Timer Counter Long Counter Long Counter Long Counter Long Retentive Timer Long Retentive Timer Long Retentive Timer
Array Element	1	Array Element ☑ ARRAY <u>E</u> lement	16 文
	OK Cancel		OK Cancel

[Registration example of global labels]

Global labels for the programs used in Chapter 5 are shown below. (Refer to Chapter 5 for details.)

	Label Name	Data Type	Class	Initial	Constant	Comment	Remark	Public Label	Motion Control Attribute
1	G_bSRVONCMD	Bit	VAR_GLOBAL			サーボオンオフ / Servo ON/OFF		Enable	WRITE (=> Motion)
2	G_bHoming1CMD	Bit	VAR_GLOBAL			原点復帰指令 Axis0001 / Homing command Axis0001		Enable	WRITE (=> Motion)
3	G_bHoming2CMD	Bit	VAR_GLOBAL			原点復帰指令 Axis0002 / Homing command Axis0002		Enable	WRITE (=> Motion)
4	G_bHoming3CMD	Bit	VAR_GLOBAL			原点復帰指令 VirtualAxis0001 / Homing command VirtualAxis001		Enable	WRITE (=> Motion)
5	G_bPosCMD	Bit	VAR_GLOBAL			単軸位置決め始動 / Single axis positioning start		Enable	WRITE (=> Motion)
6	G_bContPosCMD	Bit	VAR_GLOBAL			単軸連続位置決め始動 / Single axis continuous positioning start		Enable	WRITE (=> Motion)
7	G_bInterpolationCMD	Bit	VAR_GLOBAL			2軸直線補間制御始動 / 2-axis linear interpolation control start		Enable	WRITE (=> Motion)
8	G_bSyncCMD	Bit	VAR_GLOBAL			同期制御始動 / Synchronous control start		Enable	WRITE (=> Motion)
9	G_bResetCMD	Bit	VAR_GLOBAL			エラーリゼット / Error reset		Enable	WRITE (=> Motion)
10	G_bMotionResetCMD	Bit	VAR_GLOBAL			システムエラーリセット / System error reset		Enable	WRITE (=> Motion)
11	G_bJogF1CMD	Bit	VAR_GLOBAL			JOG正転指令 Axis0001 / JOG Positive rotation command Axis0001		Enable	WRITE (=> Motion)
12	G_bJogR1CMD	Bit	VAR_GLOBAL			JOG逆転指令 Axis0001 / JOG Reverse rotation command Axis0001		Enable	WRITE (=> Motion)
13	G_bJogF2CMD	Bit	VAR_GLOBAL			JOG正転指令 Axis0002 / JOG Positive rotation command Axis0002		Enable	WRITE (=> Motion)
14	G_bJogR2CMD	Bit	VAR_GLOBAL			JOG逆転指令 Axis0002 / JOG Reverse rotation command Axis0002		Enable	WRITE (=> Motion)
15	G_leJogVelocity	FLOAT [Double Precision]	VAR_GLOBAL			JOG速度 / JOG Velocity		Enable	WRITE (=> Motion)
16	G_bHoming1Done	Bit	VAR_GLOBAL			原点復帰完了 Axis0001 / Homing complete Axis0001		Enable	READ (Motion =>)
17	G_bHoming2Done	Bit	VAR_GLOBAL			原点復帰完了 Axis0002 / Homing complete Axis0002		Enable	READ (Motion =>)
18	G_bHoming3Done	Bit	VAR_GLOBAL			原点復帰完了 VirtualAxis0001 / Homing complete VirtualAxis0001		Enable	READ (Motion =>)
19	G_bPosDone	Bit	VAR_GLOBAL			単軸位置決め完了 / Single axis positioning complete		Enable	READ (Motion =>)
20	G_bContPosDone	Bit	VAR_GLOBAL			単軸連続位置決め完了 / Single axis continuous positioning complete		Enable	READ (Motion =>)
21	G_bInterpolationDone	Bit	VAR_GLOBAL			2軸直線補間制御完了 / 2-axis linear interpolation control complete		Enable	READ (Motion =>)
22	G_bSyncDone	Bit	VAR_GLOBAL			同期制御完了/ Synchronous control complete		Enable	READ (Motion =>)
23	G_bJog1Busy	Bit	VAR_GLOBAL			JOG運転中 Axis0001 / JOG operation in progress Axis0001		Enable	READ (Motion =>)
24	G_bJog2Busy	Bit	VAR_GLOBAL			JOG運転中 Axis0002 / JOG operation in progress Axis0002		Enable	READ (Motion =>)
25	G_bStopSignalX	Bit	VAR_GLOBAL			停止指令 Axis0001 / Stop command Axis0001		Enable	READ (Motion =>)
26	G_bStopSignalY	Bit	VAR_GLOBAL			停止指令 Axis0002 / Stop command Axis0002		Enable	READ (Motion =>)

(Note): Refer to Section 3.14 for details of public labels.

3.14 Public Labels

A PLC CPU can read and write global labels and structure members in the Motion modules by setting them as public labels.

Data such as monitor data of position and speed is defined as a variable with a specified name (label) in the Motion module. The data can be used from the PLC CPU by setting the labels as public labels.

This section describes how to register a public label. Refer to Section 4.3.2 or Section 5.3.2 for registration examples of public labels.

3.14.1 Public label registration

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

[Registering Motion module structures as a public label]

- Double-click "Label" → "Global Label" → "Ax+Global" in the navigation window of the motion control setting function, and set the "Public Label" column of the target labels to "Enable".
- 2) Double-click "Structured Data Types" → "AXIS_REAL", and set the "Public Label" column of the AxisRef (axis information) and the Md (axis monitor data) to "Enable".
- 3) Double-click "Structured Data Types" → "AXIS_REAL_MONI", and set the "Public Label" column of the AxisStatus (axis status) and the CmdInPos (command in-position) to "Enable".

🚍 🍈 Label											
🖃 🌆 Global Label			Label Name	Data Type	Class	Initial	Constant	Comment	Remark	Public Labe	Motion Control Attribute
Global		1	Axis0001	AXIS_REAL	VAR_GLOBAL	<detail•••< td=""><td>•</td><td>[X_Axis]</td><td></td><td>Enable</td><td>-</td></detail•••<>	•	[X_Axis]		Enable	-
	1	2	Axis0002	AXIS_REAL	VAR_GLOBAL	<detail…< td=""><td></td><td>[Y_Axis]</td><td></td><td>Enable</td><td></td></detail…<>		[Y_Axis]		Enable	
Sys+Global	. '/	3	VirtualAxis0001	AXIS_VIRTUAL	VAR_GLOBAL	<detail…< td=""><td>•</td><td>[Vir_Axis01]</td><td></td><td>Enable</td><td>(Note)</td></detail…<>	•	[Vir_Axis01]		Enable	(Note)
👔 Ax+Global	H	4	Link Axis0001	AXIS_VIRTUAL_LINK	VAR_GLOBAL	< Detail…		[Lin_Axis01]		Enable	
👫 Gr+Global		5	Link Axis0002	AXIS_VIRTUAL_LINK	VAR_GLOBAL	<detail•••< th=""><th>•</th><th>[Lin_Axis02]</th><th></th><th>Enable</th><th>-</th></detail•••<>	•	[Lin_Axis02]		Enable	-
🚰 Prf+Global		6									
👫 Prg+Global					(Nete). The sin	برشام امريه	avia and th	امىيانىلى م	انمادهما ميرنم	are est so nublic labels
En+Global	En+Global (Note): The virtual drive axis and the virtual linked axis are set as public labels									are set as public labels.	
🗏 🔠 Structured Data Types	1		Label Name	Data Type	Class	Initial	Constant	Comm	ent	Public Labe	Motion Control Attribute
ADDON_MONI		7	AutoDeceleration	Bit		0	1	Automatically	Deceleratin	e Disable	-
ADDON_PARAM		8	AxisName	String [Unicode](127)	,			Axis Name	Axis Name		-
	-	9	AxisStatus	Word [Signed]		0		Axis Status		Enable	-
•		10	BufferingEBs	Word [Signed]		0		Number of Buffering EB		Disable	-
•		11	CmdInPos	Bit		0		Command In-n	nsition	Enable	-
AXIS_REAL_CMD	3)	12	CmdInPos Width	ELOAT [Double Prec	is	100.0		Command In-n	nsition Wi•	• Disable	-
AXIS_REAL_MONI	H	13	CommandedAccel	FLOAT [Double Prec	is	0.0		Specified Acce	leration	Disable	-
AXIS_REAL_PRM											
AXIS_REAL_PRM_CONST	2)		Label Name	Data Type	Class	Initial G	onstant	Comment		Public Label	Motion Control Attribute
AXIS REAL	Ь	1	AxisBef	AXIS REF	0.000		A	is Information		Enable	READ (Motion =>): MdConst
AXIS ENCODER		2	PrConst	AXIS REAL PRM CO	INST		A	dis Parameter G	onstant	Disable	WRITE (=> Motion): PrConst
		3	Pr	AXIS REAL PRM			A	kis Parameter		Disable	WRITE (=> Motion): Pr
		4	Md	AXIS REAL MONI			A	ris Monitor Data	_	Enable	READ (Motion =>): Md
		5	Cd	AVIS REAL CMD	_		A	ric Control Data	_	Disable	WRITE (=> Motion): Cd
AXIS_VIRTUAL_LINK		6		in as a concomp				as control bata		0100010	mare (-> motion)- od

Reading label data in the Motion module: READ (Motion \rightarrow) Writing data to labels in the Motion module: WRITE (\rightarrow Motion)

[Setting user-created global labels as public labels]

A PLC CPU can read and write user-created global labels in the Motion module by setting them as public labels.

The motion control attribute is required to be set for the user-created global labels.

	Label Name	Data Type	Class	Initial	Constant	Comment	Remark	Public Label	Motion Control Attribute
8	G_bSyncCMD	Bit	VAR_GLOBAL			同期制御始動 / Synchronous control start		Enable	WRITE (=> Motion)
9	G_bResetCMD	Bit	VAR_GLOBAL			エラーリセット / Error reset		Enable	WRITE (=> Motion)
10	G_bMotionResetCMD	Bit	VAR_GLOBAL			システムエラーリセット / System error reset		En able	WRITE (=> Motion)
11	G_bJogF1 CMD	Bit	VAR_GLOBAL			JOG正転指令 Axis0001 / JOG Positive rotation command Axis0001		Enable	WRITE (=> Motion)
12	G_bJogR1 CMD	Bit	VAR_GLOBAL			JOG逆転指令 Axis0001 / JOG Reverse rotation command Axis0001		Enable	WRITE (=> Motion)
13	G_bJogF2CMD	Bit	VAR_GLOBAL			JOG正転指令 Axis0002 / JOG Positive rotation command Axis0002		Enable	WRITE (=> Motion)
14	G_bJogR2CMD	Bit	VAR_GLOBAL			JOG逆転指令 Axis0002 / JOG Reverse rotation command Axis0002		Enable	WRITE (=> Motion)
15	GjeJogVelocity	FLOAT [Double Precision]	VAR_GLOBAL			JOG遠度 / JOG Velocity		En able	WRITE (=> Motion)
16	G_bHoming1 Done	Bit	VAR_GLOBAL			原点復帰完了 Avis0001 / Homing complete Avis0001		Enable	READ (Motion =>)
17	G_bHoming2Done	Bit	VAR_GLOBAL			原点復帰完了 Avis0002 / Homing complete Avis0002		Enable	READ (Motion =>)
18	G_bHoming3Done	Bit	VAR_GLOBAL			原点復帰完了 VirtualAxis0001 / Homing complete VirtualAxis0001		En able	READ (Motion =>)
19	G_bPosDone	Bit	VAR_GLOBAL			単軸位置決め完了 / Single axis positioning complete		Enable	READ (Motion =>)
20	G_bContPosDone	Bit	VAR_GLOBAL			単軸連続位置決め完了 / Single axis continuous positioning complete		En able	READ (Motion =>)
21	G_bInterpolationDone	Bit	VAR_GLOBAL			2軸直線補間制御完了 / 2-axis linear interpolation control complete		Enable	READ (Motion =>)
22	G_bSyncDone	Bit	VAR_GLOBAL			同期制御完了/ Synchronous control complete		Enable	READ (Motion =>)
23	G_bJog1 Busy	Bit	VAR_GLOBAL			JOG運転中 Axis0001 / JOG operation in progress Axis0001		En able	READ (Motion =>)
		•							

3.14.2 Public label reflection

Reflect the registered public labels.

After the labels are reflected, the PLC CPU can use the members registered as public labels.

1) In the motion control setting function, select [Convert] \rightarrow [Rebuild All] to convert all public labels.

2) Select [Convert] \rightarrow [Reflect Public Labels], and click the [Yes] button on the confirmation screen.



3) Convert all the program in MELSOFT GX Works3 for the PLC CPU to use the public labels. The reflected public labels are registered in the element selection window of MELSOFT GX Works3

ecuon			Ý 4
)		4	◎◎ 噫 呩-以 ☆ ≧ × ⊇-
Label E00:R04Cl 000:RD78 RD78_00	PU G4		
Global	0000		
	SRVONCM Diaming1 Diaming2 Diaming2 Diaming3 Diam	AD CMD CMD CMD CMD ionCMD ionCMD co seetCMD D	サーボオンオフ / Servo ON/OFF 原点復帰指令 Axis0001 / Homing command Axis0001 原点復帰指令 Axis0002 / Homing command Axis0002 尾直復帰指令 NitualAxis0001 / Homing command Virtuz 単範直接機管制的容量。 準範連接使置決め始動 / Single axis continuous positioni 2個直接標準間制的容量。 24直接標準間制的容量。 24直接標準間制的容量。 25月上5日-UPT / System error reset JOG正転指令 Axis0001 / JOG Positive rotation command
	Label E00:R04Cl 000:RD78 RD78:00 Color	Label E00:R04CPU 200:R07864 RD78_0000 ← RD78_0000 ← RD78_0000 ← G_bHoming1 ← G_bHoming2 ← G	Label BOR:RD4CPU BOR:RD764 RD78,0000 RD78,0000 Colored Colo

3.14.3 A program example using public labels

A program example using public labels is shown below.

1) Double-click "Module Label" on the [Module] tab in the element selection window to open the public labels. Drag and drop the public labels.

MELSOFT GX Works3 ¥R04-78G4_FBDsa	mple_1.001B(forCapture).gx3 - [ProgramBody : LinearInterpola	tion [PRG] [FBD/LD] 362Step]			– a ×
i 🗅 📂 💾 🚑 😒 🕡	X 🗈 🗅 🖛 🖬 🖼 🖬 🖄 🖉 🚚 🎮	👧 🛃 🗮 🐘 🎇 🞜 🗳 🐬 🛃 🛼 🛞 🕀 🔾 🕂 70%	🔹 🛫 💷 🥥 🥥 🛍 🏝 🔛 Max.: 🔹		
Project Edit Find/Replace Conver	t View Online Debug Recording Diagnostics To	ol Window Help			_ 8 ×
1 24 GA 100 III III III III III III III III III	a 📾 📖 🖛 🖬 🚓 🐅 💷 🛯 🚛 🗔 🗍 📖	· · · · · · · · · · · · · · · · · · ·			
	W or or or or or or care as as as an or or or or or			d b = Element Selection	
	Programbody - Emean Responder.			(Sind POL)	35 45 [49] 12-14
	輪グループ有効化	● 東線構築制御1 Dear internetition control 1	直接转替利约2 Linear Internetition control 2	FORHER?	da ab 1 - 1 - 5 1 - 2
Module Configuration					
E Se Program	Axes group enable FB MC.GroupEnable, 1	Relative value linear interpolation control FB1 MCv.Movel.ineprinterpolate Relative.1	Relative value linear interpolation control FB2 MOV.MoveLinear/interpolate/Relative.2	Module Label	
t Initial	Z aus Incer interpole MC,GraupEnable O,binterpolationProc	MOv.MoveLinearInterpolateRelative	MOv.MoveLinear/interpolateRelative Relative volue linear	SET 3E00:R04CPU	
🔳 🏨 Scan	Execute Darve	Execute Done II	Execute Dane 100re2 DN	END 0 0000:RD78G4	
E 🖓 MAIN	Busy	ContinuousUpdate Bury	CentinuousUpdate Busy	d C ■ 1075_0000	
🖬 🙆 ServoON_Jog	Error -	- IGotherer LinearAses Active	univer LinearAses Active D	🛞 🍶 Sys+Global	
E 🔛 Homing	ErrorID	Postion data I Distance CommandAbarted II	Paston data 2 Distance CommandAborted	🖬 🗐 🗟 Ax+Global	
T 🕰 ContinuousPositioni	Ares Group Information TO 11,0000 Ares Group 201 Ares Group Tel Ares Group	Velocity Velocity Drar B	Velocity Error	🖂 🎍 Axis0001	
🖬 🙍 Synchronous		Acceleration	Reation Relation	e value inearr	n [X_Axis]
🖬 🚵 ErrorReset		Deceleration		AxisRef	his before the
🖬 🚵 LinearInterpolation		Jork Deceleration	Decaleration	Aust Aust	No Axis Information
The Local Label		derk derk	Jerk	🖬 Start	IO I/O No.
ProgramBody		VelocityMode	VelocityMode	🖂 🎽 Md	
the Event		BufferMode	BufferMode	🔂 Md	Axis Monitor Data
di Standby		Options	Options	Axis	Status Axis Status
No Execution Type		Autors	AverGroup	TEIR Dia	InPos Command In-position
Unregistered Program					ing Request Home Position Return Request
🖬 🚎 FB/FUN				H Axis0002	
D 🛄 Label				🗉 🎍 VirtualAxis	0001
E C. Parameter				🗉 🎍 LinkAxis0	001
				E 👍 LinkAxis00	002
				B Gr+Global	-001
	10/7 /L 7 M St 41	時間指令, 9イマ人	カビリロット AFB資行中断出力	Axeson	pup01 DX-V Table1
		Axes group disable FD MOLGANA AND AND AND AND AND AND AND AND AND		E 🗎 AxesGro	pupRef
	10vel.ed	MC_GroupDisable Ares group disable	RST MOV, MoveLine ar Internet	Relative value linea	GroupRef Axes Group Information
		Execute Date Coupositions	EN ENO 2-asis linear interpre-	Economical Grou	pNo Axes Group No.
	Relative value Stear in	Bury D Area group enable F	MCv_MoveLinear/Harr	ter Start	IO I/O No.
		Drar B	-•	🗉 🏨 Md	
		DrorD D Metion System Erro	RST	E Module Fo	
	R078_0000_AreaGroup001_AreaGroup1ef	Assetion a	EN END B		
			d Novel in 0		
				AxesGroupRef	
				Axes Group Information	
			RST		
			EN ENO Dorm		
The Company into Desting			d Constant	POILList Favorites Histo	v Modulo Library
Connection Destina				Pool List Tavontes Tristo	The start

Registered public label name



3.15 Writing to a Motion Module

After writing the module parameters to the PLC CPU, reset the PLC CPU, and turn the PLC CPU power ON again to communicate with the Motion module. Then, write the items set in the motion control setting to the Motion module.

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

View	Onlin	ne Debug Tool	Window	Online Data Operation						-		×
		Read from Module Write to Module Remote Operation(S Backup/Restore Delete Module Data. Monitor Motion Monitor Watch		Parameter + Program(E) Parameter + Program(E) Select All Open/Close All Deselect Al((y) Mode Hame/Data Name Basic Setting Control Data Control Data Control Data Colobal Label Colobal Label Colobal Label Colobal Label Colobal Label Colobal Label Colobal Label Colobal Label Colobal Control Parameter Colobal Con Parameter Col	Ad Legend Modulation V V V V V V V V V V V V V V V V V V V	Deins Merrory II SD /1 Last Change 2020/04/20 15:25:34 2020/06/20 19:24:15 2020/06/20 19:24:15 2020/06/20 19:24:15	femory Card Size (byte) Not Calculated Not Calculated Not Calculated	Target Drive	Module Bulkin Memory	×		~
				Size Calculation					Evecute) [Cose	

2) Write the parameters to the Motion module.

Write to Module						
	43/48					
	100/100%					
Calculation Profile: Compressing						
Global Label(Network I/O Restoration Information): 1 Work Folder Completed Global Label Initial Value: Moving to Work Folder Co Electronic Gear Setting(ElectricGearSettings.bin): Mov Folder Completed Global Label(MTNSTR.MLB): Moving to Work Folder Global Label(MTNSTR.STR): Moving to Work Folder Basic Setting/Global Label/Global Label Initial Value: Completed	Moving to mpleted ving to Work Completed Completed Compression					
Program/POU: Compression Completed Calculation Profile: Compression Completed						
	*					
When processing ends, close this window automat	ically.					

3) Turn the PLC CPU power OFF to ON again, and check that no error occurs. The error status can be checked with the LED display of the PLC CPU and the Motion module, or the system monitor of MELSOFT GX Works3.

3.16 Saving a Project

Save the created project.

1) Select [Project] \rightarrow [Save As] on MELSOFT GX Works3 to open the "Save As" screen. Enter a file name, and click the [Save] button.

Proj	ect Edit Find/Replace Con	nvert View (🕫 Save as					×
D 2	New Open	Ctrl+N Ctrl+O	Save in:	Documents	~	G 🤌 📂 🛄 -		
B	Close Save	Ctrl+S	*	Name	BDsample 1.000A	Date modified	Type GX3 File	
	Save As		Quick access					
	Delete Project Verify Project Revision Change Module Type/Operatio	► n Mode	Desktop Libraries This PC					
			Other Format: Save e Please of MELSO	< File name: Save as type: Title(A): as a <u>Workspace Fo</u> thange the window FT Navigator supp	R04-78G4_FBDsample_1.000A GX Works3 Project (*.gx3) mmat Project s with this button to use workspace fo orts this format.)	rmat project.	> Save Cancel	

3.17 Parameter List

The following shows a list of the parameters which are set in this chapter. Refer to Chapter 4 and Chapter 5 for the public label setting.

[Axis parameter setting]

1) Real Drive Axis

	ltem	Axis0001	A	xis	0002			
	Axis Information							
<u> </u>	Axis No.	1	2					
	Axis Parameter Constant	Expands setting values at a	×is variable	ini	tialization Re-i			
	Station Address Setting	192 168 3 1	192 168 3 3	2				
	Avis Type Setting	0-Real Drive Avie	0:Real Driv	- I	vie			
	Unner Limit Signal	U.Nedi Drive Akis	U.Near Din		W15			
	G Signal							
	Target							
	Signal Detection Meth	0.Detection at TRUE	0.Detection		TRUE			
	Compensation Time		0.Delection	ıaı	TRUE			
	Eilter Time	0.0 s	0.0 s					
	Lower Limit Signal	0.0 5	0.0 5					
	Tarast							
	Signal Detection Math	0.Detection of TDUE	0.0-1		TOUE			
	Signal Detection Meth	U:Detection at TRUE	U:Detection	1 at	TRUE			
	Compensation Time	0.0 s	0.0 s					
	Castral Curela Catting	0.0 s	0.0 s					
	Abashda Dasitian Defense	U:Operate in the First Operati	U:Operate i	in ti	e First Operati			
	Absolute Position Refere	3:Feed Machine Position	3:Feed Ma	chir	ne Position			
2	Absolute Position Manag	-1:Automatic Setting (Acquire	-1:Automat	IC S	etting (Acquire			
	Ring Counter Enabled S	U:Disabled	U:Disabled		•			
5	Ring Counter Lower Lim	-1000000000.0	-10000000	000	.0			
	Ring Counter Upper Lim	1000000000.0	10000000	00.	U			
3	Slave Emulation Enabled	0:Disabled	0:Disabled					
	I orque Limit Maximum \	1000.0 %	1000.0 %	-6	Forced Stop S	bignal		
3	Negative Direction Torq	300.0 %	300.0 %		🕞 Signal			
	Positive Direction Forqu	300.0 %	300.0 %		Target			
	High-speed Mode Setting	0000	0000		Signal Dete	ection Meth	0:Detection at TRUE	0:Detection at TRUE
	Axis Parameter	Expands initial values at ax	is vanable		Compensa	tion Time	0.0 s	0.0 s
)	Acceleration Limit Value	2147483647.0 um/s"2	214748364		Filter Time		0.0 s	0.0 s
	Operation Selection at S	-1:Error (Not Started)	-1:Error (N		Home Positio	n Return R	1:Home Position Return Requ	1:Home Position Return Requ
5	Command In-position Wi	100.0 um	100.0 um		Jerk Limit Val	ue	2147483647.0 um/s^3	2147483647.0 um/s^3
	Deceleration Limit Value	214/48364/.0 um/s 2	214/4836		Operation Set	ting at Ove	1:Immediate Stop	1:Immediate Stop
2	Driver Unit Conversion I	67108864 pulse	67108864		Start Permiss	ion at Hom	0:Disabled	0:Disabled
	Driver Unit Conversion I	5000 um	5000 um		Deceleration	at Stop	0.0 um/s^2	0.0 um/s^2
					Stop Selection	at Decele	1:Recreate Deceleration Curv	1:Recreate Deceleration Curv
					Stop Selection	at Stop by	3:Alternative Acceleration/De	3:Alternative Acceleration/De
					Stop Selection	n at H/W St	1:Immediate Stop	1:Immediate Stop
					Process Sele	ction at Ser	0:Ignore	0:Ignore
					Stop Selection	n at S/W St	1:Immediate Stop	1:Immediate Stop
					Driver Comm	and Discar	1:Detection Enabled	1:Detection Enabled
					Stop Signal			
					Signal			
					Target		[VAR]G bStopSignalX	[VAR]G bStopSignalY
					Signal Dete	ction Meth	0:Detection at TRUE	0:Detection at TRUE
					Compensa	tion Time	0.0 s	0.0 s
					Filter Time		0.0 s	0.0 s
					Software Stro	ke Limit Lo	-1000000000.0 um	-1000000000.0 um
					Software Stro	ke Limit Ta	-1:Invalid	-1:Invalid
					Software Stro	ke Limit Ur	1000000000 0 um	1000000000 0 um
					Position Com	mand Unit	100000000000000	100

 Negative Direction Spee 250000000.0 um/s
 250000000.0 um/s

 Operation Setting at Spe 0:Ignore
 0:Ignore

 Positive Direction Speec 250000000.0 um/s
 250000000.0 um/s

U/s

Position Command Unit Speed Command Unit U/s

2) Virtual Drive Axis

	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	1000000000	0.um		
	Position Command Unit		o um		
	Position Command Unit String	um	Iter	n	VirtualAxis0001
	Fostion Command Unit		Axis Information		
	Negative Disastice Const List Value	U/S	Axis No.		301
	Negative Direction Speed Limit value	250000000.0	Axis Parameter C	onstant	Expands setting values at axis var
	Operation Setting at Speed Limit Valu	0:Ignore	Axis Type Setting	1	3:Virtual Drive Axis
-	Positive Direction Speed Limit Value	250000000.0	Upper Limit Sign	al	
			Signal	-	
			Tarnet		
			Signal Detection	on Method	0:Detection at TRUE
			Componention	Time	
			Compensation	Time	0.0 5
			- I ritter Time		0.0 s
			E Lower Limit Sign	31	
			- Signal		
			Iarget		
			Signal Detection	on Method	0:Detection at TRUE
			Compensation	Time	0.0 s
			Filter Time		0.0 s
			Control Cycle Se	tting	0:Operate in the First Operation Cyc
			Absolute Position	n Management Setting	0:Disable Absolute Position System
			Ring Counter Ena	abled Selection	0:Disabled
			Ring Counter Lov	ver Limit Value	-1000000000.0
			Ring Counter Up	per Limit Value	1000000000.0
			High-speed Mode	e Setting	0000
			🗆 Axis Parameter		Expands initial values at axis varia
			Acceleration Lim	it Value	2147483647.0 um/s^2
			Operation Select	ion at Start Accelerat	-1:Error (Not Started)
			Command In-pos	ition Width	100.0 um
			Deceleration Lim	it Value	2147483647.0 um/s^2
			- Forced Stop Sign	al	
			Signal		
			Target		
			Signal Detection	on Method	0:Detection at TRUE
			Compensation	Time	0.0 c
			Filter Time		0.00
			Home Position D	eturn Required or No	1:Home Position Poture Posuized
			lock Limit Value	etann nequired or No	2147492647.0 um/- ^2
			Operation Settion	at Ouerra	214/40304/.0 UTVS 3
			Operation Setting	at Users Parities D	
			Start Permission	at Home Position Re	U:Disabled
			Deceleration at S	otop	0.0 um/s 2
			Stop Selection at	Deceleration to Stop	1:Recreate Deceleration Curve
			Stop Selection at	Stop by Factors	3:Alternative Acceleration/Decelerati
			Stop Selection at	H/W Stroke Limit Er	1:Immediate Stop
			Process Selection	n at Servo OFF Com	0:Ignore
			Stop Selection at	S/W Stroke Limit Err	1:Immediate Stop

3) Virtual Linked Axis

Item	LinkAxis0001	LinkAxis0002					
Axis Information							
Axis No.	401	402					
Axis Parameter Constant	Expands setting values at a	Expands setting values at axis variable initialization. Re-					
Axis Type Setting	5:Virtual Link Axis	5:Virtual Link Axis					
Control Cycle Setting	0:Operate in the First Operati	0:Operate in the First Operati					
Absolute Position Manag	0:Disable Absolute Position S	0:Disable Absolute Position S					
Ring Counter Enabled Se	0:Disabled	0:Disabled					
Ring Counter Lower Limi	-1000000000.0	-1000000000.0					
Ring Counter Upper Limi	1000000000.0	1000000000.0					
High-speed Mode Setting	0000	0000					
Axis Parameter	Expands initial values at ax	is variable initialization. Re-in					
Home Position Return Re	0:Home Position Return Not F	0:Home Position Return Not F					
Start Permission at Hom	0:Disabled	0:Disabled					
Position Command Unit	um	um					
Position Command Unit							
Speed 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 a
Acceleration Limit Value	2147483647.0 um/s^2
Operation Selection at S	-1:Error (Not Started)
Structuring Axis[1]	Axis0001
Structuring Axis[2]	Axis0002
Structuring Axis[3]	
Structuring Axis[4]	
Structuring Axis[5]	
Structuring Axis[6]	
Structuring Axis[7]	
Structuring Axis[8]	
Structuring Axis[9]	
Structuring Axis[10]	
Structuring Axis[11]	
Structuring Axis[12]	
Structuring Axis[13]	
Structuring Axis[14]	
Structuring Axis[15]	
Structuring Axis[16]	
Command In-position Wi	100.0 um
Deceleration Limit Value	2147483647.0 um/s^2
Jerk Limit Value	2147483647.0 um/s^3
Operation Setting at Ove	1:Immediate Stop
Deceleration at Stop	0.0 um/s^2
Stop Selection at Decele	1:Recreate Deceleration Curv
Structuring Axis Operation	1:Immediate Stop
Stop Selection at Stop by	3:Alternative Acceleration/De
Position Command Unit	um
Position Command Unit	
Speed Command Unit	U/s
Speed Limit Value	2500000000.0 um/s
4. PROGRAMMING BY A PLC CPU ONLY

A PLC CPU is programmed by any of the following languages: FBD/LD, ladder, or ST language. This chapter explains programming with FBD/LD that enables users to easily understand the relationship between FBs.

To obtain the sample program used in this chapter, contact your local sales office.



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

4.1 Programming Procedure for PLC CPU

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

4.1.1 Creating a program block

- 1) In the navigation window of MELSOFT GX Works3, right-click an execution type ("Scan" in this example) under "Program", and click "Add New Data".
- 2) On the "New Data" screen, set the data name, the program language, and the program file for add destination. Only alphanumeric characters can be used to for the data name.

Navigation	д×	New Data	×
-œ- ∞= 🇱 All 🔹		Basic Setting	
Ap Project		Data Type	😬 Program Block 📃 👻
Module Configuration		(Data Name)	ServoON_Jog
重 🔚 Program			
🚻 Initial		Detail Setting	
🗆 🕕 Scan 📃		Program Configuration	
😑 🖓 MAIN 🔤 📪	Add New Data Ins	Program Language	📸 FBD/LD 👻
🕌 MAIN	Sort	Program file	
🔲 🦳 ProgPou		Execution type	Scan 🗸
Local Label	Program Setting	Program file for add destine	ation MAIN 🔻
😤 ProgramBody	Expand/Collapse Tree 🔹 🕨		
🚻 Fixed Scan			
🚻 Event			
🚻 Standby			
🚻 No Execution Type			
🛅 Unregistered Program			
🚰 FB/FUN			
🗉 🏥 Label			OK Cancel
🗉 🛗 Device			ii.
🖬 🛃 Parameter			

3) A program block is added to the navigation window.



4.1.2 Program execution type

There are following five execution types.

Execution type	Description
Initial	This type of program is executed only once when the CPU module has been powered OFF and 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. Differently from the normal interrupt program, this type of program does not require the interrupt pointer (I) and the 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, right-click on a target program in the navigation window, and select [Register Program] from the shortcut menu, or drag and drop the program onto the target execution type.

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

4.1.3 Inputting a FB

Register the Motion module FBs on MELSOFT GX Works3. (Refer to Section 2.4.4.) This section explains the procedure for setting the following JOG operation FB (MCv_Jog). The sample program adds some interlock conditions to this program.



(1) 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 a global label, considering the JOG velocity is operated from an external device, such as GOT (HMI). The labels for JOG acceleration, JOG deceleration, and JOG jerk are as local labels of ServoON_Jog (refer to Section 4.1.1) since these labels are used only within the program. Refer to Section 4.2 for naming rules of labels.

[Global label]

🔓 Global [Global Label Setting] 🛛 🗙										
	Label Name	Data Type		English(Display Target)	Access from External Device					
1	G_bJogF1	Bit		JOG Positive rotation command Axis0001						
2	G_bJogR1	Bit		JOG Reverse rotation command Axis0001						
3	G_leJogVelocity	FLOAT [Double Precision]		JOG Velocity						
4			l		Γ					

[Local label]

	🔒 ServoO	N_Jog [PRG] [Local Labe ×			
ľ	<filter></filter>		Show Details(Y) 📎 Display	Setti	ng Chec <u>k</u>
		Label Name	Data Type		English(Display Target)
	1	leJogAcceleration	FLOAT [Double Precision]		JOG acceleration
Γ	2	leJogDeceleration	FLOAT [Double Precision]		JOG deceleration
	3	leJogJerk	FLOAT [Double Precision]		JOG jerk
	4				

(2) Inputting a FB

Select "library" \rightarrow "MotionControl_RD78G_****" \rightarrow "FB" on the [Library] tab in the element selection window, and you will find the Motion module FB library is registered. Drag and drop a desired FB to the program editor.



Motion module FBs are categorized in the following three groups.

Group	Description
Management	A motion control FB that takes an axis or an axes group for the argument
(administrative FB)	and does not change the axis status or the axes group status by execution
Operation-Individual	A motion control FB that takes an axis or an axes group for the argument
(motion FB)	and changes the axis status or the axes status by execution
StandardFB (general FB)	A motion control FB that does not take an axis or an axes group for the argument

The FB for JOG operation (MCv_Jog) is under the "Operation-Individual" tree.

Drag and drop MCv_Jog to the editor. When an undefined label is entered, the "Undefined Label Registration" screen appears. Enter the label name, the registered destination, and, if necessary, the comment. This example uses the initial settings.

Undefined Label Registration								
Not defined as of Please set new li	global label or local label. abel information to be registered.							
Label Name MCv_Jog_1								
Label Setting I	nformation							
<u>R</u> egistered Destination	Local Label(ServoON_Jog)	~						
Cla <u>s</u> s	VAR	\sim						
Data T <u>v</u> pe	MCv_Jog							
Co <u>n</u> stant								
Co <u>m</u> ment								
Op <u>e</u> nthe set the la	label editor and bel details after registering label information.							
	ОК	Cancel						

ProgramBody : ServoON_Jog [P ×	4 ۵ 🗸	Element Selection 📮 🗄
	~	(Find POU)
	- 1	Ģ• Ģ ☆ ⊑ × <u>a</u> •
	- 1	🖃 🧰 FB 🔗
		🕀 🌗 Management
MOv_Jos_1		🖃 퉲 Operation-Individu
MUV_dog		MC_CamIn Cam Operation Sta
JogForward Done		📴 MC_CombineAx+ Addition/Subtracti
Jor Backward Busy		📷 MC_GearIn Gear Operation Sta
		MC_GroupStop Group Forced Stop
Velocity Active		E MC_Home OPR
		💼 MC_MoveAbsolt Absolute Value Pos
Acceleration CommandAborted		💼 MC_MoveRelativ Relative Value Posi
Deceleration Error		MC_MoveVelocit Speed Control
		📷 MC_Stop Forced Stop
Jerk ErroriD		📷 MC_TorqueConti Torque Control
Ontions		MCv_BacklashCc Backlash Compens
		MCv_DirectionFi Direction Restrictic
Axis D		📴 MCv_Jog 🛛 JOG
1		🕞 MCv_MoveCircu Absolute Circular lı
		📴 MCv_MoveCircu Relative Circular In
		📴 MCv_MoveLinea Absolute Linear Int 🤊
		MCv. log
		[Version]
		01B
		[Last Change]
		2019/09/19 15:09:27
	$$ \sim	[Comment]
	\rightarrow	POU List Favorites History Module Library

When the setting is completed, click the [OK] button. The selected FB (MCv_Jog) is displayed on the program editor.

(3) Connecting I/O signals

Connect FBD and LD elements to a target input connection point of the FB. Place an element on the editor, and move it through drag and drop.

(a) LD element

Click where an element is to be placed on the program editor, and then click the element icon on the tool bar to place the element.

When a contact element is placed, it shows "???". Double-click the element, and enter a bit type label or a bit device number. For the label name, the registered global labels are displayed as a suggestion. Connect the LD elements to the input connection point of the FB.



The following shows the diagram that connects the LD elements to the JogForward and JogBackward input connection points.



(b) FBD elements

Click where the FBD element is to be placed, and then directly enter the device No. or label name there.

When entering "G_leJogVelocity" that is registered as a global label in (1), click the left side of the Velocity input of MCv_Jog and enter "G_leJogVelocity". The registered global labels are displayed as a suggestion. Connect the FBD element to the input connection point of the FB.



When an undefined label is entered, the "Undefined Label Registration" screen appears. Enter the data type and the registered destination.

U	ndefined Label F	Registration	×					
	Not defined as g Please set new la	lobal label or local label. bel information to be registered.						
	Label Name G_leJogVelocity							
	Label Setting Ir	formation						
	Registered Destination	Global Label(Global) ~						
	Cla <u>s</u> s	VAR_GLOBAL ~						
	Data T <u>v</u> pe	FLOAT [Double Precision]						
	Co <u>n</u> stant							
	Co <u>m</u> ment							
	Op <u>e</u> nthel set the lab	abel editor and el details after registering label information.						
		OK Cancel						

Connect the local labels (leJogAcceleration, leJogDeceleration, and leJogJerk) registered in (1) to the corresponding input connection point (Acceleration, Deceleration, and Jerk inputs) of MCv_Jog. For the Options input, connect the constant value of "0".

[NOTES]

Note that values need to be stored to G_leJogVelocity, leJogAccelertion, leJogDeceleration, and leJogJerk by another program (using EDMOVP instruction, etc.) (For a setting example, refer to Section 4.6.)



(c) Axis input (public label)

Connect the AxisRef structure registered as a public label to the Axis input connection point. (Refer to Section 3.14.1.)

Click [Module] tab in the element selection window. Select "Module label" \rightarrow "0000:RD78G4" \rightarrow "RD78_0000" \rightarrow "Ax+Global" \rightarrow "Axis0001" \rightarrow "AxisRef", and then drag and drop the AxisRef (axis information) to the program editor.



When "RD78G_0000.Axis0001.AxisRef" appears on the editor, connect it to the Axis input connection point of the FB.

[POINTS]

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 axis label name, such as "Axis0001", will be displayed. When you select "Axis0001" and enter ".", the structure member, such as "AxisRef", will be displayed. Select "AxisRef", and the input is completed.





The program is completed.

[POINTS]

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 "0".

4.2 Naming Rules of Labels

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

Data tara			Prefix		
Data type		Value range	Local	Global	
Bit	BOOL	FALSE(0), TRUE(1)	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	

[Examples of local labels] Bit: bMoveCMD Double-precision real number: lePosition Arrays: wAxes[16] Timer: tdTimer1

[Examples of global labels] Bit: G_bJogF1 Double-precision real number: G_leVelocity

4.3 Projects

4.3.1 Program names

The following shows the program examples explained in this chapter.



4.3.2 Settings for global and public labels

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

(1) PLC CPU

		Label Name	Data Type	Class		Assign (Device/Label)	Initial Value	Constant	English(Display Target)
	1	G_bSRVOFF	Bit	 VAR_GLOBAL	•				Servo OFF
	2	G_bJogF1	Bit	 VAR_GLOBAL	-				JOG Positive rotation command Axis0001
	3	G_bJogR1	Bit	 VAR_GLOBAL	-				JOG Reverse rotation command Axis0001
	4	G_bJogF2	Bit	 VAR_GLOBAL	-				JOG Positive rotation command Axis0002
	5	G_bJogR2	Bit	 VAR_GLOBAL	-				JOG Reverse rotation command Axis0002
	6	G_bHoming1CMD	Bit	 VAR_GLOBAL	-				Homing command Axis0001
4)	7	G_bHoming2CMD	Bit	 VAR GLOBAL	-				Homing command Axis0002
·) ·	8	G_bHoming3CMD	Bit	 VAR_GLOBAL	-				Homing command VirtualAxis0001
	9	G_bPosCMD	Bit	 VAR_GLOBAL	-				Single axis positioning start
	10	G_bContPosCMD	Bit	 VAR_GLOBAL	-			in a service	Single axis continuous positioning start
	11	G_bInterpolationCMD	Bit	 VAR_GLOBAL	-				2-axis linear interpolation control start
	12	G_bSyncCMD	Bit	 VAR GLOBAL	-				Synchronous control start
	13	G bErrorReset	Bit	VAR GLOBAL	-				Error reset
	14	G bSvsErrorReset	Bit	VAR GLOBAL	-			ann ann assann	System error reset
2)	15	G_leJogVelocity	FLOAT [Double Precision]	 VAR_GLOBAL	-				JOG Velocity
	16	G_bHoming1Req	Bit	 VAR_GLOBAL	-				Homing start request Axis0001
	17	G_bHoming2Reg	Bit	 VAR GLOBAL	*				Homing start request Axis0002
	18	G_bHoming3Req	Bit	 VAR_GLOBAL	-				Homing start request VirtualAxis0001
2)	19	G_bPosReg	Bit	 VAR_GLOBAL	-				Single axis positioning start request
3)	20	G_bContPosReg	Bit	 VAR_GLOBAL	-				Single axis continuous positioning start request
	21	G_bInterpolationReg	Bit	 VAR GLOBAL	-				2-axis linear interpolation control start request
	22	G_bSyncReg	Bit	 VAR GLOBAL	-				Synchronous control start request
	23	G_bJog1Busy	Bit	 VAR_GLOBAL	-				JOG operation in progress Axis0001
	24	G_bJog2Busy	Bit	 VAR_GLOBAL	+				JOG operation in progress Axis0002
	25				*				

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.

(2) Motion module

The following shows the global and public label settings of the Motion module. Set these labels on the motion control setting function of MELSOFT GX Works3. When the settings are finished, reflect the public labels. (Refer to Section 3.14.2.)

[AX+Global]

4	Label Name	Data Type	Class	Initial Value	Constant	Comment	Public Label	Motion Control
1	Axis0001	AXIS_REAL	VAR_GLO BAL	<detailed…< td=""><td></td><td>[X_Axis]</td><td>Enable</td><td>-</td></detailed…<>		[X_Axis]	Enable	-
2	Axis0002	AXIS_REAL	VAR_GLOBAL	<detailed…< td=""><td></td><td>[Y_Axis]</td><td>Enable</td><td>-</td></detailed…<>		[Y_Axis]	Enable	-
3	VirtualAxis0001	AXIS_VIRTUAL	VAR_GLOBAL	<detailed…< td=""><td></td><td>[Vir_Axis01]</td><td>Enable</td><td>-</td></detailed…<>		[Vir_Axis01]	Enable	-
4	LinkAxis0001	AXIS_VIRTUAL_LINK	VAR_GLOBAL	<detailed…< td=""><td></td><td>[Lin_Axis01]</td><td>Enable</td><td>-</td></detailed…<>		[Lin_Axis01]	Enable	-
5	LinkAxis0002	AXIS_VIRTUAL_LINK	VAR_GLOBAL	<detailed…< td=""><td></td><td>[Lin_Axis02]</td><td>Enable</td><td>-</td></detailed…<>		[Lin_Axis02]	Enable	-
6								

[Gr+Global]

	Label Name	Data Type	Class	Initial Value	Constant	Comment	Remark	Public Label	Motion Control Attribute
1	AxesGroup001	AXES_GROUP	VAR_GLOBAL	<detailed setti…<="" td=""><td></td><td>[X-Y Table]</td><td></td><td>Enable</td><td>-</td></detailed>		[X-Y Table]		Enable	-
2									

[Global]

The following are the structures that store data related to the system. Set "Public label" to "Enable".

_	Label Name	Data Type	Class	Initial Value	Constant	Comment	Remark	Public Label	Motion Control Attribute
1	G_bStopSignalX	Bit	VAR_GLOBAL			停止指令 Axis0001 / Stop command Axis0001		Enable	WRITE (=> Motion)
2	G_bStopSignalY	Bit	VAR_GLOBAL			停止指令 Axis0002 / Stop command Axis0002		Enable	WRITE (=> Motion)
3									

≈

≈

[Sys+Global]

The following is the structure that stores data related to the system. Set "Public label" to "Enable".

_	Label Name	Data Type	Class	Initial Value	Constant	Comment	Remark	Public Label	Motion Control Attribute
1	System	SYSTEM	VAR_GLOBAL	<detailed setti<="" td=""><td></td><td></td><td></td><td>Enable</td><td>-</td></detailed>				Enable	-
2									

[AXIS_REAL structure]

	Label Name	Data Type	Class	Initial Value	Constant	Comment	Public Label	Motion Control Attribute
1	AxisRef	AXIS_REF				Axis Information	Enable	READ (Motion =>): MdConst
2	PrConst	AXIS_REAL_PRM_CO····				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]

1	Label Name	Data Type	Class	Initial Value	Constant	Comment	Public Label	Motion Control
1	AccelerationLimit	FLOAT [Double Precim		0.0		Acceleration Limit Value	Disable	-
2	Acceleration0 verride	FLOAT [Double Precim		1.0		Acceleration Override Coefficient	Disable	-
3	AccelerationZeroBehavior	Word [Signed]		-1		Operation Selection at Start Accel····	Disable	-
4	ActualPosition	FLOAT [Double Precim		0.0		Feedback Position	Disable	-
5	ActualVelocity	FLOAT [Double Precim		0.0		Feedback Speed	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 Precim		100.0		Command In–position Width	Disable	-

≈

26	Driver_Mode	Word [Signed]	0	Driver Control Mode	Disable	-
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	DriverError	Bit	0	Drive Module Error Detection	Disable	-
31	DriverErrorID	Word [Unsigned]/Bit…	0	Drive Module Error Code	Disable	-
32	DriverErrorDetailID	Word [Unsigned]/Bit…	0	Drive Module Error Detail Code	Disable	-
33	Error	Bit	0	Axis Error Detection	Disable	-
34	ErrorID	Word [Unsigned]/Bit…	0	Axis Error Code	Disable	-
35	FeedMachinePosition	FLOAT [Double Precim	0.0	Feed Machine Position	Disable	-
36	FollowupDisable	Bit	0	Follow-up Disabled	Disable	-
37	ForcedStop_Released	Bit	0	Forced Stop Cancelling	Disable	-
38	ForcedStop_Signal	SIGNAL_SELECT		Forced Stop Signal	Disable	-
39	Homing_Complete	Bit	0	Home Position Return Completed	Disable	-
40	Homing_Request	Bit	1	Home Position Return Request	Enable	-
41	Homing Required	Bit	0	Home Position Return Required or	Disable	-

≈

[AXES_GROUP structure]

	Label Name	Data Type	Class	Initial Value	Constant	Comment	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]

_	Label Name	Data Type	Class	Initial Value	Constant	Comment	Public Label	Motion Control
1	AccelerationLimit	FLOAT [Double Precision]		0.0		Acceleration Limit Value	Disable	-
2	Acceleration0 verride	FLOAT [Double Precision]		1.0		Acceleration Override Coefficient	Disable	-
3	AccelerationZeroBehavior	Word [Signed]		-1		Operation Selection at Start Accel…	Disable	-
4	ActualVelocity	FLOAT [Double Precision]		0.0		Feedback Speed	Disable	-
5	Analyzing	Bit		0		Analyzing	Disable	-
6	AutoDeceleration	Bit		0		Automatically Decelerating	Disable	-
7	Axis	AXIS_REF(116)				Structuring Axis	Disable	-
8	BufferingFBs	Word [Signed]		0		Number of Buffering FBs	Disable	-
9	CmdInPos	Bit		0		Command In–position		-
10	CmdInPos_Width	FLOAT [Double Precision]		100.0		Command In–position Width	Disable	-
11	CommandedAcceleration	FLOAT [Double Precision]		0.0		Specified Acceleration	Disable	-
12	CommandedDeceleration	FLOAT [Double Precision]		0.0		Specified Deceleration	Disable	-
13	CommandedJerk	FLOAT [Double Precision]		0.0		Specified Jerk	Disable	-
14	CommandedVelocity	FLOAT [Double Precision]		0.0		Specified Speed	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 Stri…		0		Axes Group Error Code	Disable	-
18	GroupName	String [Unicode](127)		""		Axes Group Name	Disable	-
19	GroupStatus	Word [Signed]		0		Axes Group Status		-
20	InterpolationAxes	Double Word [Unsigned]…		0		Interpolation Axis	Disable	-
21	InVelocity	Bit		0		Target Speed Reached	Disable	-

≈

[AXIS_VIRTUAL structure]

1		Label Name	Data Type	Class	Initial Value	Constant	Comment	Public Label	Motion Control Attribute
	1	AxisRef	AXIS_REF				Axis Information	Enable	READ (Motion =>): MdConst
	2	PrConst	AXIS_REAL_PRM_CO····				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_VIRTUAL_MONI structure]

1	Label Name	Data Type	Class	Initial	Constant	Comment	Public Label	Motion Control
1	AccelerationLimit	FLOAT [Double Precision]		0.0		Acceleration Limit Value	Disable	
2	AccelerationO verride	FLOAT [Double Precision]		1.0		Acceleration Override Coefficient	Disable	-
3	AccelerationZeroBehavior	Word [Signed]		-1		Operation Selection at Start Accel…	Disable	-
4	Analyzing	Bit		0		Analyzing	Disable	
5	AutoDeceleration	Bit		0		Automatically Decelerating	Disable	-
6	AxisName	String [Unicode](127)		1111		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	CommandedAcceleration	FLOAT [Double Precision]		0.0		Specified Acceleration	Disable	-

≈

≈

20	ForcedStop_Released	Bit	0	Forced Stop Cancelling	Disable	-
21	ForcedStop_Signal	SIGNAL_SELECT		Forced Stop Signal	Disable	-
22	Homing_Complete	Bit	0	Home Position Return Completed	Disable	-
23	Homing_Request	Bít	1	Home Position Return Request	Enable	-
24	Homing_Required	Bit	0	Home Position Return Required or…	Disable	-
OF	THE AVER A PROPERTY AND A PROPERTY A	Pro-	0	la la siona con s	S: 11	

≈

≈

≈

[AXIS_VIRTUA_LINK structure]

	Label Name	Data Type	Class	Initial Value	Constant	Comment	Public Label	Motion Control Attribute
1	AxisRef	AXIS_REF				Axis Information	Enable	READ (Motion =>): Md…
2	PrConst	AXIS_VIRTUAL_LINK_PRM_CONST				Axis Parameter Constant	Disable	WRITE (=> Motion): Pr···
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]

	Label Name	Data Type	Class	Initial Value	Constant	Comment	Public Label	Motion Control
1	Addon_AbsSystem	ADDO N_MO NI				Add-on AbsSystem Monitor	Disable	-
2	Addon_Axis	ADDO N_MO NI				Add-on Axis Monitor	Disable	-
3	Addon_ExternalSignal	ADDO N_MO NI				Add-on ExternalSignal Monitor	Disable	-
4	Addon_FileTransfer	ADDON_MONI				Add-on FileTransfer Monitor	Disable	-

≈

23	BuffermemoryRefreshCycle	CYCLE_MONI		Buffer Memory Refresh Cycle Moni…	Disable	-
24	Environment_UserRootPath	String(127)	"	User Root Path	Disable	-
25	Error	Bit	0	Motion System Error Detection	Enable	-
26	ErrorHistory	Word [Unsigned]…	0	Error History Information	Disable	-
27	ErrorHistory_Latest	Word [Signed]	0	Latest Error History Data No.	Disable	-

4.4 PLC READY (Program Name: ServoON_Jog)

The 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 by a PLC CPU to start operation of the Motion module.

(1) Program example

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



4.5 Servo ON (Program Name: ServoON_Jog)

This program turns ON the real drive axes connected to the servo system. The following two FBs are used for executing servo ON.

(1) FBs

Type Command		Description		
MCFB	MC_Power	Switches a specified axis to the operation possible status.		
(administrative)	MCv_AllPower	Switches all axes to the operation possible status.		

(2) Local labels

	Label Name	Data Type		English(Display Target)
1	MCv_AllPower_1	MCv_AllPower		All axes servo ON FB
2	MCv_Jog_1	MCv_Jog		JOG operation FB Axis0001
3	MCv_Jog_2	MCv_Jog		JOG operation FB Axis0002
4	leJogAcceleration	FLOAT The labels for JOG ope	rati	on G acceleration
5	leJogDeceleration	FLOAT [Double Precision]		JOG deceleration
6	leJogJerk	FLOAT [Double Precision]		JOG jerk
7				

Drag and drop MCv_AllPower to the program editor. The FB (MCv_AllPower_1) is added as a local label. (Refer to Section 4.1.3.)

(3) Program example

When PLC READY [Y0] is turned ON, Ready [X0] is turned ON.

Turning ON of X0 is used as the condition for executing all axes servo ON.

When executing servo OFF, turn ON G_bSRVOFF.

For MCv_AllPower, connect AxisRef type structure to the Axis input connection point.



[POINTS]

The I/O No. of the Motion module must be specified in the AxisRef type structure. When a user specifies the I/O No. in the PLC CPU program, store the numerical value that deletes the lowest digit of the Motion module I/O No. to the AxisRef.StartIO.

For example, when the I/O No. is "0010", store "1" 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.

(1) FB

Туре	Command	Description
MCFB (motion)	MCv_Jog	Executes a JOG operation according to the target velocity.

(2) 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.

(a) Overview

Select the acceleration/deceleration methods from the following.

Method	Description
Acceleration/deceleration specification method (Initial)	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 speed.

(b) Setting method

The acceleration/deceleration method can be specified by the Options input of motionFBs, such as MCv_Jog.

Bit	Description	
0 to 2	Acceleration/deceleration method setting 0: mcAccDec (acceleration/deceleration specification method) 1: mcFixedTime (acceleration/deceleration time-fixed method)	
3 to 15	The function vertice depending on the EP	
16 to 31	The function varies depending on the FB.	

(c) Acceleration/deceleration specification method

Select "0: mcAccDec" in the Options input (using bit 0 to 2 for acceleration/deceleration method setting) of the FB, and set acceleration, deceleration, and jerk.

1) Trapezoidal acceleration/deceleration

When "0.0" is set for the Jerk input of the FB, the operation is referred to as the trapezoidal acceleration/deceleration.

The velocity creates a trapezoidal shape.



2) Jerk acceleration/deceleration

When values other than "0.0" is set for the Jerk input of the FB, the operation is referred to as the jerk acceleration/deceleration.

The velocity creates an S-curve shape.



(d) Acceleration/deceleration time-fixed method

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



(e) FB input variables

The following table lists the input variables explained in (b) to (d).

Input variable	Name	Details
Velocity	Target velocity	Specify the target velocity.
Acceleration	Acceleration or acceleration/ deceleration time	 Specify the acceleration or the acceleration/deceleration time. The unit is [U/s²] when the method is set to "0: mcAccDec". The unit is [s] when the method is set to "1: mcFixedTime".
Deceleration	Deceleration	Specify the deceleration. • The unit is [U/s²] when the method is set to "0: mcAccDec". • Not used when the method is set to "1: mcFixedTime"
Jerk	Jerk	Specify the jerk. • The unit is [U/s³] when the method is set to "0: mcAccDec". • Not used when the method is set to "1: mcFixedTime".
Options	Options	Specify the acceleration/deceleration method using bit 0 to2. 0: mcAccDec Acceleration/deceleration specification method 1: mcFixedTime Acceleration/deceleration time-fixed method

[POINTS]

When the "Acceleration/deceleration specification method" (Options: 0) is selected, the acceleration/deceleration can be calculated using the acceleration/deceleration time as follows. [Target velocity: V [U/s], acceleration time [s], deceleration time [s]],

Velocity := (target velocity V);

Acceleration := (target velocity V/acceleration time);

Deceleration := (target velocity V/deceleration time);

Options := (mcAccDec);

(3) Local labels

The label used in the Servo ON program

				-
		Label Name	Data Type	English(Display Target)
	1	MCv_AllPower_1	MCv_AllPower	 All axes servo ON FB
1)	 2	MCv_Jog_1	MCv_Jog	 JOG operation FB Axis0001
1)	3	MCv_Jog_2	MCv_Jog	 JOG operation FB Axis0002
2)	4	leJogAcceleration	FLOAT [Double Precision]	 JOG acceleration
2) ·	5	leJogDeceleration	FLOAT [Double Precision]	 JOG deceleration
	6	leJogJerk	FLOAT [Double Precision]	 JOG jerk
	7			

1) These labels are automatically added when the user drags and drops the FB "MCv_Jog" to the program editor.

2) These labels are registered manually.

(4) Program example

The EDMOVP instruction is used to store values to 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]

	Axis0001	Axis0002
JOG positive rotation command	G_bJogF1	G_bJogF2
JOG reverse rotation command	G_bJogR1	G_bJogR2

JOG運転用データ設定 Data setting for JOG operation	When Axis0001 becomes servo ON status, the values for JOG velocity, acceleration, deceleration, and jerk are stored to the specified labels.
Driver Servo On Status	EDMOVP EN ENO s d Gle.bgVelocity 500000 s d electeration
Connected	
	EDMOVP
	EN ENO JOG deceleration s d le.bgDeceleration 0.0 s d le.bgJerk

[POINTS]

To enter the EDMOVP instruction, directly enter "EDMOVP" on the editor, or drag and drop "EDMOVP" to the editor (select "APPLICATION INSTRUCTIONS" \rightarrow "Real number instructions" on the [POU List] tab in the element selection window.)



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 Appendix.

(1) Overview

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

[Time chart of data set type homing (Method37)]

Homing completion	ON	
Servo motor speed	Positive rotation 0 r/min Reverse rotation	Homing position data
Homing operation start	ON OFF	

(2) FB

Type Command		Description		
MCFB (motion)	MC_Home	Executes homing of the specified axis.		

(3) Local labels

		Label Name	Data Type	English(Display Target)
1)	 1	MC_Home_1	MC_Home	 Homing FB1 Axis0001
1)	 2	MC_Home_2	MC_Home	 Homing FB2 Axis0002
	3	MC Home 3	MC Home	Homing FB3 VirtualAxis0001
	4	bHoming1Done	Bit	 Homing FB1 Done output
2)	5	bHoming1 Error	Bit	 Homing FB1 Error output
	 6	bHoming2Done	Bit	 Homing FB2 Done output
	7	bHoming2Error	Bit	 Homing FB2 Error output
	8	bHoming3Done	Bit	 Homing FB3 Done output
	9	bHoming3Error	Bit	 Homing FB3 Error output
	10			

1) These labels are automatically added when the user drags and drops the FB (MC_Home) to the program editor.

2) These labels are registered manually.

(4) Program example

This program executes homing for the axis 1 (Axis0001), the 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 input of MC_Home. The start signal is turned OFF when the 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 an error occurs.

[Start signal]

	Axis0001	Axis0002	VirtualAxis0001
Homing command	G_bHoming1CMD	G_bHoming2CMD	G_bHoming3CMD



ning FB2 Error

1

The configuration of the axis 2 program is

the same as that of the axis 1 program.

ndAb orted

Error ErrorID

Cor

Asis

di F



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.

(1) FBs

Туре	Command	Description		
MCFB	MC_MoveAbsolute	Executes positioning using a specified absolute position.		
(motion)	MC_MoveRelative	Executes positioning using a specified relative distance.		

(2) Local labels

4			Label Name	Data Type	English(Display Target)
1)		1	MC_MoveRelative_1	MC_MoveRelative	 Relative value positioning FB
		2	leDistance	FLOAT [Double Precision]	 Distance
		3	leVelocity	FLOAT [Double Precision]	 Velocity
		4	leAcceleration	FLOAT [Double Precision]	 Acceleration
		5	leDeceleration	FLOAT [Double Precision]	 Deceleration
2)		6	leJerk	FLOAT [Double Precision]	 Jerk
2)		7	bDone	Bit	 Relative value positioning FB Done output
		8	bError	Bit	 Relative value positioning FB Error output
		9	bValueSet	Bit	 Variable set complete
		10	bCommandAborted	Bit	 Relative value positioning FB CommandAborted output
		11			

1) This label is automatically added when the user drags and drops the FB (MC_MoveRelative) to the program editor.

2) These labels are registered manually.

(3) Program example

This program executes relative positioning in the following operation pattern.



When the single axis positioning command is turned ON, each positioning data is stored to the 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 three conditions is 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.



4 - 28

4.9 Single Axis Continuous Positioning (Program Name: ContinuousPositioning)

(1) Overview

An axis can continuously execute multiple motion FBs without stopping by buffering the next motion FB of another instance while executing the first motion FB. Select the buffer mode through the BufferMode input of a motion control FB. Up to two motion FBs can be buffered on one axis.

(2) Operation pattern of buffer mode



(3) Local labels

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

1) These labels are automatically added when the user drags and drops the FB (MC_MoveRelative and TON) to the program editor.

2) These labels are registered manually.

(4) Program example

This program executes relative positioning in the following operation pattern.



When the single axis positioning command is turned ON, each positioning data is stored to the 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 next FB (MC_MoveRelative_2), which enables buffering of the next FB while executing the first FB. When the first FB is finished, the next FB is continuously executed.

To set 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 three conditions is 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]

Single axis continuous positioning command	G_bContPosCMD
--	---------------

4. PROGRAMMING BY A PLC CPU ONLY







[POINTS]

Instead of connecting a label, output signals of a FB can be directly entered by describing "(FB name). (output signal name)" in the editor.

4.10 Interpolation Control (Program Name: LinearInterpolation)

4.10.1 Procedure for interpolation control

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



4.10.2 Enabling/disabling an axes group

Refer to Section 3.12 for axes group setting. To execute interpolation control, set the axes group status to "4: GroupStandby".

(1) FBs

Туре	Command	Description
MCFB	MC_GroupEnable	Transits the specified axes group status from "0: GroupDisabled" to "4: GroupStandby".
(administrative)	MC_GroupDisable	Transits the specified axes group status to "0: GroupDisabled".

4.10.3 Interpolation control

The FBs for linear and circular interpolation control are as follows. Execute the FBs after the axes group is enabled.

(1)	FBs
٠.		

Туре	Command	Description	
	MCv_MoveLinearInterpolateAbsolute	Absolute value linear interpolation control	
MCFB	MCv_MoveLinearInterpolateRelative	Relative value linear interpolation control	
(motion)	MCv_MoveCircularInterpolateAbsolute	Absolute value circular interpolation control	
	MCv_MoveCircularInterpolateRelative	Relative value circular interpolation control	

4.10.4 Program example for linear interpolation

(1) Operation pattern

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



- (2) Axis No. and movement amount settings
 - (a) LinearAxes input of MCv_MoveLinearInterpolateRelative

Use the array of INT (signed word) type with 16 elements.

In the sample program, wAxes[0..15] (label) is used.

The interpolation control axes are selected from the structuring axis [1] to [16] set in the axes group (in Section 3.12). Specify the structuring axis No. to be used for the interpolation control by wAxes. (Note that the wAxes must be set from the index No. [0] in order.)

(b) Distance input

Use the array of LREAL (double-precision real number) type with 16 elements In the sample program, lePosition[0..15] is used.

Define the movement amount of the structuring axis [1] to [16] in lePosition[0] to lePosition[15].

[POINTS]

Regardless of the number of the interpolation control axes, the number of elements must be 16 for the INT-type array for the LinearAxes input and the LREAL-type array for the Distance input.

[Setting example 1] • Axes group Structuring axis [1]: Axis0001, structuring axis [2]: Axis0002, and structuring axis [3]: Axis0003

Linear interpolation
 Axis0001 and Axis0002

Select <u>F</u> older Disp	olay All Data 🗸
Item	AxesGroup001
Axes Group No.	1
Axes Group Parameter	Expands initial values at ax
Acceleration Limit Val	u 2147483647.0 pulse/s^2
Operation Selection at	(-1:Error (Not Started)
Structuring Axis[1]	Axis0001
Structuring Axis[2]	Axis0002
Structuring Axis[3]	Axis0003
Structuring Axis[4]	
Structuring Axis[5]	
Structuring Axis[6]	
Structuring Axis[7]	
Structuring Axis[8]	
Structuring Axis[9]	
Structuring Axis[10]	
Structuring Axis[11]	
Structuring Axis[12]	
Structuring Axis[13]	
Structuring Axis[14]	
Structuring Axis[15]	
Structuring Axis[16]	

[Setting example 2]

Axes group

Structuring axis [1]: Axis0001, structuring axis [2]: Axis0002,

- and structuring axis [3]: Axis0003
- Linear interpolation
 Axis0002 and Axis0003

etting Item		wAxes[0] := 2; (\leftarrow structuring axis [2])
Select Folder Display All Data 🗸		wAxes[1] := 3; (← structuring axis [3])
Item	AxesGroup001	
Axes Group No.	1	lePosition[0] = 0.0
Axes Group Parameter	Expands initial values at ax	
Acceleration Limit Val	a 2147483647.0 pulse/s^2	lePosition[1] := (movement amount of structuri
 Operation Selection at 	(-1:Error (Not Started)	axis 2 $(=Axis0002)$).
Structuring Axis[1]	Axis0001	
Structuring Axis[2]	Axis0002	lePosition[2] := (movement amount of structuri
Structuring Axis[3]	Axis0003	axis 3 $(=Axis0003)$
Structuring Axis[4]		
Structuring Axis[5]		
Structuring Axis[6]		
Structuring Axis[7]		
Structuring Axis[8]		
Structuring Axis[9]		
Structuring Axis[10]		
Structuring Axis[11]		
Structuring Axis[12]		
Structuring Axis[13]		
Structuring Axis[14]		
Structuring Axis[15]		
Structuring Axis[16]		

(3) Local labels

			Label Name	Data Type	English(Display Target)
		1	wAxes	Word [Signed](015)	 Interpolation axis
		2	lePosition1	FLOAT [Double Precision](015)	 Position data 1
		3	lePosition2	FLOAT [Double Precision](015)	 Position data 2
2)		4	leVelocity	FLOAT [Double Precision]	 Velocity
Z)		5	leAcceleration	FLOAT [Double Precision]	 Acceleration
		6	leDeceleration	FLOAT [Double Precision]	 Deceleration
		7	leJerk	FLOAT [Double Precision]	Jerk
		8	MC_GroupEnable_1	MC_GroupEnable	 Axes group enable FB
1)		9	MCv_MoveLinearInterpolateRelative_1	MCv_MoveLinearInterpolateRelative	 Relative value linear interpolation control FB1
1)		10	MCv_MoveLinearInterpolateRelative_2	MCv_MoveLinearInterpolateRelative	 Relative value linear interpolation control FB2
		11	MC GroupDisable 1	MC GroupDisable	 Axes group disable FB
		12	bDone2	Bit	 Relative value linear interpolation control FB2 Done output
		13	bGrpEnError	Bit	 Axes group enable FB Error output
		14	bError	Bit	 Relative value linear interpolation control FB Error output
		15	bGrpDsblDone	Bit	Axes group disable FB Done output
	·	16	TON_1	TON	 On-delay timer FB
		17	bDwell_in	Bit	 Timer input
		18	bDwell_out	Bit	 Timer output
		19	bValueSet	Bit	 Variable set complete
		20	bCommandAborted	Bit	 Relative value linear interpolation control FB CommandAborted output
		21	bDone_Set	Bit	 Done
		22	bCommandAborted_Set	Bit	 FB abortion of execution
		23			

1) These labels are automatically added when the user drags and drops the corresponding FB to the program editor.

2) These labels are registered manually.

(4) Program example

When the two-axis interpolation control start is turned ON, each positioning data is stored to the 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 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 is 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 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]

Two-axis linear interpolation control start	G_bInterpolationCMD
---	---------------------


(Note) When the values for the Velocity, Acceleration, and Deceleration inputs are omitted, those of the previous FB are applied.

MOV_Mo

4 1

軸グループ無効化 Axes group disabled	Axes group disable FB	結動指令。タイマ入力をリセット Resets the start command and timer input
Timer output	MC_GroupDisable_1 MC_GroupDisable	Aves group disable FB··· RST
bDwell_out	Axes group disable	
Relative value linear in…	Busy	Axes group enable FB··· d
bError	Error	bGrpEnError
1	ErrorID	Motion System Error
Axes Group Information	AvecGroup	RD78_0000.System.Md····
	, and any p	Timer input
	\sim	
- "MC_GroupDisable" disal	oles the axes group	
when the dwell time is pa	assed or when the	RST
linear interpolation control	error is turned ON.	EN END Done
		d bDone,Set
		RST
		EN END EB electrico of execution
		d bDommand Aborted_Set
		The start signal and the timer input, etc.
		are turned OFF when any of the following
		conditions is satisfied.
		Group disable done
		System error
		- Gystem enter
各FB実行中斷出力	参FBI5~	- 出力
FB Command Aborted output	FB Error o	putput
MCv_MoveLinearInter···· Relation	ve value linea MDv. Move	eLinearinter···· Relative value linea····
	nmand Aborted	bError

MOV_Mo

The labels are turned ON when the OR condition (CommandoAborted and Error outputs of the two linear interpolation control FBs) is satisfied, which is used as an interlock condition or when releasing the self-holding circuit.

- 4.11 Synchronous Control (Program Name: Synchronous)
- 4.11.1 Procedures for executing synchronous control

The following shows the procedure flow for executing synchronous control.



4.11.2 Calculation profile

Waveform data used for control is collectively called calculation profile data. This section explains how to create a cam data.

(1) Creating a new calculation profile

In the navigation window of the motion control setting function, right-click "Calculation Profile" and select "Add New Data".

Calculation Profile ProfileData0001 [Cam #1]	P Add New Da	ta Ins	Ne	ew Data		×
Wetwork I/O	New Folder	Ctrl+Shift+N		Basic Setting		
🗉 🔚 Program	Import File			Data Type	Calculation Profile	-
📅 FB/FUN	Sort	+		(Data Name)	ProfileData0001	
E NE Label						
			1	Detailed Setting		
				Data Format		
				Туре	Cam Data	-
				Interpolation Method Specification	Section Interpolation	-
				Expand Setting		
				Auto Expand	Yes	-
				Profile ID (1 to 60000)	1	
				Repetitive Operation	Enable	-
				Input Absolute Coordinate	Disable (Relative Coordinate)	-
				Output Absolute Coordinate	Disable (Relative Coordinate)	-
					OK Cance	

The following shows the setting items on the "New data" screen.

Item	Description
Automatic open	Yes: The calculation profile data is automatically opened at power-on. No: The open FB for calculation profile data needs to be executed.
Periodic	Invalid: The control ends when it executes until the end of calculation profile data. Valid: The execution of calculation profile data is continuously repeated.
Output absolute coordinate	Invalid (relative): When the calculation profile (cam) is started, an output value is calculated based on the current value. When executing a feed cam operation, select this setting. Valid (absolute): The output value at the time the calculation profile (cam) is started is calculated to be always the start point for one cycle of the calculation profile data. When the start point and the end point of the calculation 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.

The example in this document uses the initial value. Click the [OK] button.

(2) Creating a cam data

Set the calculation profile waveform.



The following shows the necessary setting items.

No.	Item	Description
1	Resolution	Set the resolution of the cam data.
2	Longth por Cyclo	Set the length per cycle and its unit.
2	Lengin per Cycle	(Set the movement amount of the master axis for one cam cycle)
2	Stroke Amount	Set the stroke amount and its unit.
3		(Set the movement amount of the slave axis for one cam cycle)
		Set the time for one cam cycle.
4	Cam Time per Cycle	This setting is used when velocity, acceleration, and jerk are
		calculated.
5	Stroke Setting	Set the stroke.

The following shows the example settings.

No.	Item	Setting value				
1 Resolution		256				
2	Length per Cycle	200000 (blank for the unit setting)				
3	Stroke Amount	200000 (blank for the unit setting)				
4	Cam Time per Cycle	1.000				
5	Stroke Setting	Refer to the following table.				

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	

[POINTS]

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

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

4.11.3 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.

(1) FBs

Туре	Command	Description
	MC_CamIn	Executes cam operation.
	MC_GearIn	Executes gear operation based on the specified speed ratio between the master axis and the slave axis.
	MC_CombineAxes	Combines motion of two axes by a selectable combination method, and outputs the result to the third axis.
MCFB (motion)	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.
	MCv_*****Filter	Executes the specific filter processing to the input of Master, and outputs the result to Slave.
	MC_Stop	Stops the synchronous control.

4.11.4 Axes configuration

This chapter explains the following cam system.



4.11.5 Program example

(1) Operation pattern

The X-axis moves from the origin point for 400000.0 [um] while the Y-axis (Axis0002) is operated according to the cam pattern created in Section 4.11.2. 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.



(2) Virtual drive axis and virtual linked axis

This program uses the virtual drive axis (VirtualAxis0001) and the virtual liked 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 VirtualAxis and LinkAxis. Set each of the AxisRef structures to the public label setting. (Refer to Section 4.3.2.)

- Data Type FLOAT [Double Precision] Label Nam English(Display Target) leVelocity Velocity FLOAT [Double Precision Acceleration Acceleration 2) 0AT leDeceleration Double Precision Deceleration [Double Precision 4 leJerk Jerk)istance AC_GearIn_ iear operation FB GearIn 1) FB combining the motion of two master axes Cam operation FB <u>/C Combine</u> MC CombineA Note VIC_CamIn_1 CamID MC_CamIn MC CAM ID am ID Jam ID Cam operation FB inSync output Relative value positioning FB Relative value positioning FB Error output Relative value positioning FBI Done output Relative value positioning FB 10 bIns olnSync VC_MoveRelative_1 MC_MoveRelative 11 13 MC MoveRelativ 14 MC MoveRel elative value positioning F 15 elative value positioning FB2 Done output ne 16 StopDone Xis Stop complete Relative value positioning start Variable set complete SyncMove ValueSet 18 Relative value positioning FB CommandAborted output 19 omman MC_Stop_1 MC_Stop_2 MC_Stop MC_Stop lxis Stop FB1 20 21 Axis Stop FB 22 23 MC Ston MC Axis Stop FB3
- (3) Local labels

1) These labels are automatically added when the user drags and drops the corresponding FB to the program editor.

2) These labels are registered manually.

(Note): Registering MC_CAM_ID type structure

Click the [Library] tab in the element selection window, and select "Library" \rightarrow "MotionControl_RD78_****" \rightarrow "Structured Data Type". Drag and drop "MC_CAM_ID" to "Structured Data Types" under "Label" in the navigation window. The structure is registered under the "Structured Data Types" tree and available as a data type on the label editor.



(4) Program example

When the synchronous control start is turned ON, each positioning data is stored to the 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 Axis0002 status is turned to "7: SynchronizedMotion", Axis0001 (Master axis) starts positioning. At this time, Axis0002 is operated according to the specified calculation 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 calculation profile. The synchronous control is stopped by executing MC_Stop on the real drive axis (Axis0002) and the virtual linked axes (LinkAxis0001 and LinkAxis0002) when any of the following conditions is 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]

Synchronous control start G_bSyncCMD



(To the next page)



(To the next page)

(Continued)





The labels are turned ON when the OR condition (the CommandoAborted and Error outputs of the synchronous control FBs and the positioning FBs) is satisfied, which is used as an interlock condition or when releasing the self-holding circuit.

4.12 Error Reset (Program Name: ErrorReset)

This program resets the error on each axis.

(1) FBs

Туре	Command	Description
	MC_Reset	Resets errors and warnings of the axis.
MCFB (administrative)	MC_GroupReset	Resets errors and warnings of the axes group.
(duministrative)	MCv_MotionErrorReset	Reset all errors and warnings of the motion system.

(2) 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.



軸グループエラーリセット Axes group error reset	Arms marin error reset EP	システムエラーリセット System error reset	Sustan array const EP
Error reset G. bError Reset	MC_GroupReset	System error reset G.bSysErrorReset	MCv_MotionError Reset
	Execute Done	•	Execute Done
	Options Busy	•	Options Busy
	CommandAborted		CommandAborted
	Error	•	Error 🛑
	ErrorID	10.1	ErrorID 💼
RD78_000 0 AxesGroup001.AxesGroupRef	AxesGroup	PO No. RD78_0000 Axis0001 Axis Ref.StartIO	Start10

4.13 Checking Operation

4.13.1 Conversion and Writing of Programs

Write the program to the PLC CPU. Select [Convert] \rightarrow [Rebuild All] to convert all the program.



Confirm that no error occurs after executing "Rebuild All". Select [Online] \rightarrow [Write to PLC], and write the program to the PLC CPU.

When the parameters and the public labels of the Motion module are changed, write the program after converting all the program and reflecting the public labels in the motion control setting function.

ay <u>S</u> etting Related	I Functions									
😃 🌔 🇊 Write		9	1	👔 Verity	. 🔜 🧳	Delete				
Parameter + Program(E) Select All	Legend								
Open/Close All(T)	Deselect All(<u>N</u>)	CPU 6	Built-in Me	mory	SD M	1emory Card	Intelligent Function Module			
Iodule Name/Data Name			5		Detail	Title	Last Change	Size (Byte)		
📲 R04-78G4_FB	Dsample_1.001B									
😑 🐼 Parameter										
- 🦑 System P	Parameter/CPU Parameter						2020/06/29 14:27:42	Not Calculat	ed	
- 🙆 Module P	arameter						2020/08/28 15:25:16	Not Calculat	ed	_
Memory	Card Parameter						2020/06/29 12:06:10	Not Calculat	ed	_
Remote F	Password						2020/06/29 12:06:10	Not Calculat	ed	-1
D 🔂 CC-Link II	E TSN Configuration									
Slave De	vice Setting				Detail	_	-	Not Calculat	ed	_
🖯 🛅 Global Lab	el									
Gobal La	bel Setting						2020/08/28 15:24:41	Not Calculat	ed	_
Global Lab	iel Initial Value									-
GLBLINF							2020/08/28 15:24:41	Not Calculat	ed	
Display Memory Cap nory Capacity	acity 😴									
Size Calculation	Program Memory								Free 121/160KB	
gend	Data Memory								Free	
Used									1732/2049KB	
Increased	Device/Label Memory (File Stor	age Area) —							Free	
Decreased									192/256KB	
Free: 5% or Less	SD Memory Card								Free	
									0/0KB	

4.13.2 Axis monitor

The 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.

(1) Displaying monitor screens

The monitor screen can be displayed by the following two methods.

 Select [Tool] → [Drive Tool List] in MELSOFT GX Works3, and double-click "Monitor" on the "Drive Tool List" screen. Select the Motion module on the "Module Selection (Monitor)" screen, and click the [OK] button.

Tool	Window Help	Drive Tool List	×
*	Memory Card Image: Check Program Check Program Check Parameter Check Parameter Confirm Memory Size (Offline) Confirm Memory Size (Offline) Eoging Configuration Tool Realtime Monitor Function Module Tool List Drive Tool List Predefined Protocol Support Function Circuit Trace Update Firmware Profile Management Imagement Register Sample Library FB Property Management (Offline) Shortcut Key Options	Start the selected drive tool. Start the selected drive tool. Monitor Monitor OK OK	Cancel
		Module Selection(Monitor) Module Selection 0000:RD78G4 OK Cancel	×

2) In the motion control setting function, select [Online] \rightarrow [Motion Monitor] \rightarrow [Axis Monitor].



(2) Monitor screen

0000:RD78G4 - Axis Monito	or					
: 🖪 🕞 🔊 📖 📖	3				2	
	<u> </u>		4			,
Axis Monitor	Type: Real Drive Axis	Display Selection			Syst	em Monitor
	.,,	Monitor Item	election 🔠 Monitor Axis Selection 📃 Split 💿 Upper Pane	O Lower Pane	oy st	
Font	t Size: 9pt 🗸 🗸	L			4	Monitor Item Selection
	Axis #1	Axis #2		^		PLC Ready
Axis Name	Axis0001	Axis0002				Peady
Axis Status	0:Axis Disabled	0:Axis Disabled				
Control Cycle	1	1			•	Synchronization flag
Position Command Unit Display	μm	μm				System Basic Cycle Monitor.Proces 3800 n
Speed Command Unit Display	µm/s	µm/s				System Basic Cycle Monitor.Maxim
Command Current Position	0.0 um	0.0 um				4760 n
Feedback Position	0.0 um	0.0 um				System Basic Cycle Monitor.Settin
Specified Position	0.0 um	0.0 um				1000000 n
Command Current Speed	0.0 um/s	0.0 um/s				System Basic Cycle Monitor.Cycle
Feedback Speed	0.0 um/s	0.0 um/s				Operation Cycle Monitor[1].Proces.
Specified Speed	0.0 um/s	0.0 um/s				62440 r
Negative Direction Speed Limit Value	2500000000.0 um/s	2500000000.0 um/s				Operation Cycle Monitor[1].Maxim 67840 n
Positive Direction Speed Limit Value	2500000000.0 um/s	250000000.0 um/s				Operation Cycle Monitor[1].Settin
Automatically Decelerating	FALSE	FALSE				
Command In-position	FALSE	FALSE				Operation Cycle Monitor[1].Cycle
Negative Direction Torque Limit Value	300.0 %	300.0 %				Forced Stop Cancelling Motion System Error Detection
Positive Direction Torque Limit Value	300.0 %	300.0 %				Latest Motion System Error Code
Execution Profile ID No.	0	0				000
Home Position Return Completed	FALSE	FALSE			•	Motion System Warning Detection
Home Position Return Request	FALSE	FALSE				000
Start Permission at Home Position Return Uncompleted	FALSE	FALSE				Network Error Detection
Upper Limit Signal Status	FALSE	FALSE				000
Lower Limit Signal Status	FALSE	FALSE				Basic System Software Version
Forced Stop Cancelling	TRUE	TRUE				
Axis Error Detection	FALSE	FALSE				Boot Software Version
Axis Error Code	0000	0000				
Axis Warning Detection	FALSE	FALSE				Network Boot Software Version
Axis Warning Code	0000	0000				
Driver Ready On Status	FALSE	FALSE		~	1	

No.	Description				
1	The monitor items for each axis are displayed.				
2	The system monitor items are displayed.				
3	The axis type to be monitored can be changed.				
4	The monitor items can be added/deleted.				

4.13.3 Program monitor

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

(1) Displaying monitor screen

Select [Online] \rightarrow [Monitor] \rightarrow [Start Monitoring (All Windows)], or click the icon [\underline{m}] on the tool bar.

(2) Monitor screen



4.13.4 Watch

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

(1) Displaying monitor screens

Select [View] \rightarrow [Docking Window] \rightarrow [Watch 1 to Watch 4] in MELSOFT GX Works3.



(2) Registration of devices/labels/structures to a watch window

Enter the device No./label/structure to be registered in the "Name" column of a watch window or right-click a label or a structure to be registered on the program editor, and select [Register to Watch Window] \rightarrow [Watch 1 to 4].

L	G_bJogF1	G_bHoming1Req	G_bPosReq	G_bContPosReq	G_bInterpolation
	JOG Reverse rota G_bJogR1	Undo Redo			VI.
F		Cut Copy Paste			
		Select FBD Network	Block	-	
		Convert(B)			
		Edit Find View	, , ,		
		Cross Reference Device List Dataflow Analysis			
G Axis0002		FB Property		•	
	JOG Positive rota	Register to Watch W	ndow	Watch 1	Reverse rotat
⊢	G_bJogF2	Register to Device Ba Wave Display(W)	tch Replace	Watch 2 Watch 3	G_bJogR2
	JOG Reverse rota G bJocR2	Process Control Exter	nsion	Watch 4	Positive rotat
		Open Help(G)			

(3) Monitoring start

Select [Online] \rightarrow [Watch] \rightarrow [Start Watching] in MELSOFT GX Works3.

(4) Changing current values

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

The bit device can be switched to ON/OFF with double-click while holding the [Shift] key or with [Shift] + [Enter] after selecting the row.

/atch 1 🕴 🖓							
HIII-ON HI-OFF HAON/OFF toggle	2 Update						
Name	Current Value	Display Format	Data Type	English	Forced Input/Output Status	Device Test with Execution	^
<pre>G_bSRVOFF</pre>		BIN	Bit	Servo OFF		-	
😭 G_bJogF1		BIN	Bit	JOG Positive rotation command Axis0001	-		
😭 G_bJogR1		BIN	Bit	JOG Reverse rotation command Axis0001			
😭 G_bJogF2		BIN	Bit	JOG Positive rotation command Axis0002			
e_bJogR2		BIN	Bit	JOG Reverse rotation command Axis0002			
G_bHoming1CMD		BIN	Bit	Homing command Axis0001			
G_bHoming2CMD		BIN	Bit	Homing command Axis0002			
G_bHoming3CMD		BIN	Bit	Homing command VirtualAxis0001			
G_bPosCMD		BIN	Bit	Single axis positioning start			
<pre> @ G_bContPosCMD </pre>		BIN	Bit	Single axis continuous positioning start			
G_bInterpolationCMD		BIN	Bit	2-axis linear interpolation control start			
G_bSyncCMD		BIN	Bit	Synchronous control start			
<pre>G_bErrorReset</pre>		BIN	Bit	Error reset			
							~
🚍 Output 📰 Progress 🚟 Watch 1							

[POINTS]

In the sample program, "RD78_0000.G_bStopSignalX" and "RD78_0000.G_bStopSignalY" are registered in the watch window 1. Turning these signals ON aborts the operation.

4.13.5 Event History

The event history of the Motion module can be checked from [Online] \rightarrow [Motion Monitor] \rightarrow [Event History] in the 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.

Event	History- [0000:RD78G4]							×
Ever	Refresh(U) Its to Exclude		Number of Events:	158	Ref	ìne(D)	8		
Refin	e Match All the Conditions		O Match /	Any One of t	he Conditions				
1.	Event Type	/ Includin	ig Next		\sim			~	
2.		/			\sim			~	
з.		/			~			~	
	Include program errors (Allo	w jumping)			Start Re	fine	Clear Refine	Conditions	
No.	Occurrence Date		Event Type	Status	Event Code	Overvi	ew		^
00001	2020/09/07 07:33:19	402538560	System	4	007F0	MCFB S	tart (Control System)		-
00002	2020/09/07 07:33:19	391777656	System	•	007EE	Servo S	ystem Recorder Start		
00003	2020/09/07 07:33:19	289555680	System	((007FE	Homing	Request OFF to ON		
00004	2020/09/07 07:33:19	289115064	System		007FD	Current	Position Restoration Com	pletion	
00005	2020/09/07 07:33:19	289032336	System	•	007FD	Current	Position Restoration Com	pletion	_
00006	2020/09/07 07:33:19	288934016	System	•	007FD	Current	Position Restoration Com	pletion	_
00007	2020/09/04 08:46:03	947703512	System	0	007F0	MCFB S	tart (Control System)		× 1
Legend	A Major Warning	💧 Mod (i) Infor	erate 🔥 Mind	or				Jump Clear All	
l	Detailed Information	MCFB Star	t/Stop Information		-		-		
		Detail Code FB Type :N FB Instance Instance Na	e :0 MCv_AllPower e ID :45430288 ame :MCv_AllPowe	r	-		-		
	Cause	MCFB was	started.				I		
	Corrective Action	-							
	Create File							Close	

To check the event history of the PLC CPU, click [Diagnosis] \rightarrow [System monitor] in MELSOFT GX Works3, and click "Event history" on the "System monitor" screen.

MEMO

5. PROGRAMMING BY A PLC CPU AND THE MOTION MODULE

This chapter explains programming by a PLC CPU and the Motion module. To obtain the sample program used in this chapter, contact your local sales office.

■ File name: **<u>R04-78G4</u> <u>STsample</u> *****.gx3** (***** indicates the version)



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.

5.1 Programming Procedure for Motion Module

The motion control setting function is used to create a program for the Motion module.

5.1.1 Creating a program block

- 1) In the navigation window of the motion control setting function, right-click an execution type ("Normal" in this example) under "Program", and click "Add New Data".
- 2) On the "New Data" screen, set the data name. Only alphanumeric characters can be used to for the data name.

Navigation	Р×		New Data		×
🖳 🗠 🔅			Basic Setting		
85-0000-RD78G4			Data Type	💼 Program Block	-
Basic Setting			(Data Name)	ServoON_Jog	
System Setting			Detailed Setting		
🖽 🎆 Axis			Program Configuration		
🖽 ò Axes Group			Program Language	ब्रा ST	
🚺 I/O Data			Program File		
🖽 🧖 Calculation Profile			Execution Type	Normal	-
Wetwork I/O			Program File of Add Destination	<same as="" data="" name=""></same>	
🗏 🚰 Program					
🚻 Initial					
🕀 🚻 Normal 📃					
🚻 Fixed Scan 📑	Add New Data Ir	ns			
🚻 Standby	Sort	•			
🚻 No Executio	Drogram Setting				
🚰 FB/FUN	Flogram Setting			01	
🗉 🏥 Label	Expand/Collapse Tree	•		OK	ancei .::

3) A program block is added to the navigation window.



5.1.2 Program execution type

There are following four execution types.

Execution type	Description
0: Standby	This type of program is executed only when its execution is requested. The program is executed after converted to a normal execution type program by using a PSCAN instruction from another running program.
1: Normal	This type of program is executed by normal task of the motion system.
2: Initial	This type of program is executed only once when PLC READY [Y0] changes from OFF to ON.
3: Fixed scan	This type of program is executed at each specified time. (The first operation cycle: 62.5 [us] ^(Note-1) to 6000 [ms]) Note that the time shorter than the operation cycle cannot be set.

(Note-1): "us" indicates microsecond.

To set the execution type, drag and drop the program block onto the target execution type, or change the type from "Basic Setting" \rightarrow "Program Execution Setting".

Navigation 🛛 🕂 🗙	Pasic Setting ×						4 ▷ 🗕
• E • •= 🔅	Setting Item List	Setting Item	1				
Constant Section Constant Section		Item Image: Search Image: Search			nen you want to ch s nether to set initia de nen changing the p	lange the value of ti al values to labels a program name to us	
E 🥵 Program ft Initial R ft Normal ft Fixed Scan ft Standby	Program Execution Se B- Operation Setting at Error	Operati Invalid File N Opera	Pragram Execution Setting Operation Setting at Error Invalid Device/Label/Buffer Memory Specifi File Name Specification Operation Error			led Setting> t whether to stop t	the user program vi
			Execution Order	Program Name	Execution Type	Program Exe Fixed Scan	cution Condition Interval Setting
			1	ServoON_Jog	1:Normal	~	
			2		0:Standby		
			3		1:Normal		
			4		2:Initial		
			5		3:Fixed Scan		
			6			_	

5.1.3 Inputting a FB

This section explains the procedure for setting the following JOG operation FB (MCv_Jog). The sample program adds some interlock conditions and public label settings to this program.

```
//Axis0001 JOG
2
   MCv_Jog_1(
3
                    := Axis0001.AxisRef ,
       Axis
4
       JogForward := G_bJogF1CMD ,
5
       JogBackward := G_bJogR1CMD ,
6
       Velocity
                   := G_leJogVelocity ,
7
       Acceleration: = leJogAcceleration ,
8
       Deceleration = leJogDeceleration ,
             := leJogJerk ,
=> @ blactb
9
       Jerk
10
       Busy
                   => G_bJog1Busy
11
   );
12
```

(1) 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, JOG signal during operation, and JOG velocity are registered as a global label, considering the JOG velocity is operated from an external device, such as GOT (HMI). The labels for JOG acceleration, JOG deceleration, and JOG jerk are as local labels of ServoON_Jog (refer to Section 5.1.1) since these labels are used only within the program. Refer to Section 5.2 for naming rules of labels.

[Global label]

_	Label Name	Data Type	Class	Ini	Co	Comment	Re 🔺	Public Label	Motion
1	G_bJogF1CMD	Bit	VAR_GLOBAL			JOG正転指令 Axis0001 / JOG Positive rotation command Axis0001		Disable	-
2	G_bJogR1CMD	Bit	VAR_GLOBAL			JOG逆転指令 Axis0001 / JOG Reverse rotation command Axis0001		Disable	-
3	G_bJog1 Busy	Bit	VAR_GLOBAL			JOG運転中 Axis0001 / JOG operation in progress Axis0001		Disable	-
4	G_leJogVelocity	FLOAT [Double···	VAR_GLOBAL			JOG速度 / JOG Velocity		Disable	-
5									

[Local label]

	4	Label Name	Data Type	Class	Initial	Constant	Comment
1		leJogAcceleration	FLOAT [Double Precision]	VAR			JOG加速度 / JOG acceleration
2	2	leJogDeceleration	FLOAT [Double Precision]	VAR			JOG減速度 / JOG deceleration
3	}	leJogJerk	FLOAT [Double Precision]	VAR			JOGジャーク / JOG jerk
4	ł						

(2) Inputting a FB

Drag and drop "MCv_Jog" under "Operation-Individual" tree on the [POU list] tab of the element selection window.

sr ServoON_Jog [PRG] [ST] * ×	4 ▷ 🗕	Element Selection	Ψ×
1 :		(Find POU)	約44 倍
	~	u- u <mark>☆</mark> ⊇ × ⊒-	
		Display Target: Al	l ~
		Bistable	^
		Control Syntax	
		Control Syntax	
		Operator	
		Operator	
		Motion Control Function/Func	tion Block
		Management	
		Operation - Individual	
		MC_Home	OPR
		The MC_Stop	Forced Stop
		MC_GroupStop	Group Forced Stop
	Drag & drop	MC_MoveAbsolute	Absolute Value Position
	Brug a drop	MC_MoveRelative	Relative Value Position
		Sector MCv_Jog	JOG
	-	MC_MoveVelocity	Speed Control
		MC_TorqueControl	Torque Control
		MCv_SpeedControl	Speed Control (Includii
		MCv_MoveLinearInterpo	: Absolute Value Linear I
		MCv_MoveLinearInterpo	: Relative Value Linear In
		MCv_MoveCircularInter	Absolute Value Circular
		MCv_MoveCircularinter	Relative Value Circular
		MC_Camin	Cam Operation Start
		MC_Gearin	Gear Operation Start
<	v	MCv_Jog POULList Favorites History	Library

Motion module FBs are categorized in the following three groups.

Group	Description
Management	A motion control FB that takes an axis or an axes group for the argument
(administrative FB)	and does not change the axis status or the axes group status by execution
Operation-Individual	A motion control FB that takes an axis or an axes group for the argument
(motion FB)	and changes the axis status or the axes status by execution
StandardFB (general FB)	A motion control FB that does not take an axis or an axes group for the argument

When an undefined label is entered, the "Undefined Label Registration" screen appears. Enter the label name, the registered destination, and, if necessary, the comment. This example uses the initial settings.

Undefined Label Re	egistration	×
Not defined as glo Please set new lab	obal label or local label. el information to be registered.	
<u>L</u> abel Name	MCv_Jog_1	
Label Setting Info	ormation	
Registered Destination	Local Label(ServoON_Jog) ~	
Cla <u>s</u> s	VAR ~	
Data T <u>y</u> pe	MCv_Jog	
Co <u>n</u> stant		
Co <u>m</u> ment		
Op <u>e</u> nthe lal set the labe	bel editor and I details after registering label information.	
	OK Cancel	

When the setting is completed, click the [OK] button. The selected FB (MCv_Jog) is displayed on the program editor.

51	Serv	voON_JOG [PRG] [ST] * ×	4 Þ 🗕
Γ	1	:, MCv_Jog_1(Axis:= <u>?AXIS_REF?</u> , JogForward:= <u>?BOOL?</u> , JogBackward:= <u>?BOOL?</u> , Velocity:= <u>?LREAL?</u> ,	Acc
	2	. <u>PBOOL?</u> ,Busy=> <u>PBOOL?</u> ,Deceleration:= <u>PLWORD?</u> ,Done=> <u>PBOOL?</u> ,Busy=> <u>PBOOL?</u>	
	3	Active=> <u>?B00L?</u> ,CommandAborted=> <u>?B00L?</u> ,Error=> <u>?B00L?</u> ,ErrorID=> <u>?W0RD?</u>);	

Delete the semicolon at the head of the FB.

(Depending on where a FB is added, there may be two semicolons at the end of the FB. In that case, delete one.)



Users can break a line or adjust indents of a FB.

The following program example breaks the line at every I/O signal and adds indents as follows.



(3) Inputting I/O signals

Enter labels for the undefined variables which display ?AXIS_REF?, ?BOOL?, etc.

(a) AxisRef type structure

The following describes how to input AxisRef type structure (axis information) of the axis 1 in the input variable of "Axis".

1) When entering "ax" instead of ?AXIS_REF?, the registered labels are displayed as a suggestion.

2) Select Axis0001 and enter ".". Then, the suggestions of the structure member for Axis0001 are displayed.

1	MCv_Jog_1(
2	Axis:= Axis0001.	e.		
3	JogForward:= ?BO	AxisRef	AXIS_REF	Axis Information
4	JogBackward:= ?B	🖽 Cd	AXIS_REAL_CMD	Axis Control Data
5	Velocity:= ?LREA	🖽 Md	AXIS_REAL_MONI	Axis Monitor Data
6	Acceleration:= ?	🛅 Pr	AXIS_REAL_PRM	Axis Parameter
7	Deceleration:= ?	PrConst	AXIS_REAL_PRM_CONST	Axis Parameter Constant
8	Jerk:= ?LREAL? .			Setting
9	Options:= ?DWORD	?.		
10	Done=> ?BOOL? .			
11	Busy=> ?BOOL? .			
12	Active=> ?BOOL?			
13	CommandAborted=>	?BOOL? .		
14	Error=> ?BOOL? .			
15	ErrorID=> ?WORD?			
16):	~~		

3) Select "AxisRef".

1	MCv Jog 1(
2	Axis:= Axis0001.AxisRef ,
-3	JogForward:= <u>?BOOL</u> ? ,
4	JogBackward:= <u>?BOOL?</u> ,
-5	Velocity:= <u>?LREAL?</u> ,
6	Acceleration:= <u>?LREAL?</u> ,
- 7	<pre>Deceleration:= ?LREAL? ,</pre>
8	Jerk:= <u>?LREAL?</u> ,
9	Options:= <u>?DWORD?</u> ,
10	Done=> <u>?BOOL?</u> ,
11	Busy=> <u>?BOOL?</u> ,
12	Active=> <u>?BOOL?</u> ,
13	CommandAborted=> <u>?BOOL?</u> ,
14	Error=> <u>?BOOL?</u> ,
15	ErrorID=> <u>?WORD?</u>
16);
17	

- (b) Global and local labels
 - Enter G_bJogF1CMD (global label) for the JogForward input. When entering "G_b" instead of ?BOOL?, the registered labels are displayed as a suggestion.

1	MC∨_Jog_1(
2	Axis:= Axis0001.AxisRef ,	
3	JogForward:= <u>G_b</u> ,	
4	JogBackward:= ?В 😭 с_bјод визу	BOOL JOG運転中 Axis0001 / JOG operatio
5	Velocity:= <u>?LREA</u> 😭 G_bjogFiCMD	BOOL JOG正転指令 Axis0001 / JOG Positiv
6	Acceleration:= ? 😭 G_bjogRiCMD	BOOL JOG逆転指令 Axis0001 / JOG Rever
7	Deceleration:= ?	Setting
8	Jerk:= <u>?LREAL?</u> ,	
9	Options:= ?DWORD? ,	
10	Done=> ?BOOL? ,	
11	Busy=> ?BOOL? ,	
12	Active=> ?BOOL? .	
13	CommandAborted=> ?BOOL? ,	
14	Error=> ?BOOL? ,	
15	ErrorID=> ?WORD?	
16);	

2) Select "G_bJogF1CMD".

1	MCv_Jog_1(
2	Axis:= Axis0001.AxisRef ,
- 3	JogForward:= G bJogF1CMD ,
- 4	JogBackward:= <u>?BOOL?</u> ,
5	Velocity:= ?LREAL? ,
6	Acceleration:= ?LREAL? ,
- 7	Deceleration:= <u>?LREAL?</u> ,
8	Jerk:= ?LREAL? .
9	Options:= <u>?DWORD?</u> ,
10	Done=> ?BOOL?
11	Busy=> ?BOOL? ,
12	Active=> ?BOOL? .
13	CommandAborted=> ?BOOL? ,
14	Error=> ?BOOL? .
15	ErrorID=> ?WORD?
16);
17	

3) Enter the labels registered in (1) to the corresponding input signals (JogBackward, Velocity, Acceleration, Deceleration, Jerk, and Busy) in the same manner.



(c) Omitting I/O signals

I/O signals of a FB can be omitted when the initial value is used, or the signal is not used. In this example, the initial value (0) is used for the Options input, and the Done, Active, CommandAborted, Error, and ErrorID outputs are not used. Therefore, these signals can be omitted.

Delete the unnecessary "," at the end of the FB.



(d) Adding comments and editing for appearance Add comments or indents as necessary.

1	//A×isOOO1 JOG	
2	MC∨ Jog 1(
3	Axis :=	AxisOOO1.AxisRef ,
4	JogForward :=	G bJogF1CMD ,
5	JogBackward :=	G_bJogR1CMD ,
6	Velocity :=	G leJogVelocity ,
7	Acceleration:=	leJogAcceleration ,
8	Deceleration:=	leJogDeceleration ,
9	Jerk :=	leJogJerk ,
10	Busy =>	G bJog1Busy
11);	
12		

The program creation is completed.

5.1.4 ENUM enumerator

The constant values of the axis status (Axis0001.Md.AxisStatus), the buffer mode (BufferMode), etc. are defined as an ENUM enumerator. Programming using the ENUM enumerator is possible.

(1) Inputting ENUM enumerator

[Example of ENUM enumerators for the FB argument] The enumerator for BufferMode is defined by "MC_BUFFER_MODE__mc*****" (label). (For details of the BufferMode, refer to Section 5.9.)

17								
18	//框	1対値位置決め,R	elat	tive va	lue positio	ning		
19	MC_N	MoveRelative_1(
20		Axis	:=	Axis00	01.AxisRef	,		
21		Execute	:=	bExecu	ite_P ,			
22		Distance	:=	leDist	ance ,			
23		Velocity	:=	leVelo	ocity ,			
24		Acceleration	:=	TeAcce	eleration ,			
25		Deceleration	:=	leDece	eleration ,			
26		Jerk	:=	leJerk	· ,			
27		BufferMode	:=	mc_buf				
28		Done	=>	bDone	MC_BUFFER_MODE	_mcAborting	INT	Aborting
29		Busy	=>	bBusy	MC_BUFFER_MODE	mcBlendingHigh	INT	BlendingHigh
30		CommandAborted	=>	bComma	MC_BUFFER_MODE	mcBlendingLow	INT	BlendingLow
31		Error	=>	bError	MC_BUFFER_MODE	_mcBlandingNext	INT	BlendingNext
32);				MC_BUFFER_MODE	mcBlendingPrev	INT	BlendingPrevious
33					MC_BUFFER_MODE	mcBuffered	INT	Buffered
34								setting

[Example of ENUM enumerator for axis variable] The enumerator for axis status is defined by "MC_AXIS_STATUS__*****" (label).

66					
67	//入力軸始動, Ir	iput axis start			
68	SET(G_bSyncCMD &	i (AxisOOO2.Md.AxisStatus	= MC_AXIS_STATUS_	_SynchronizedMotion),bSyncMoveCMD);
69					

(2) Checking values of ENUM enumerators

- 1) Select [Tool] \rightarrow [Option] to display the option screen.
- 2) Set "Constant" of the ST editor to "Yes", and click the [OK] button.

3) Check the value of an ENUM enumerator by moving the cursor on it in the program.

Tool	Window Help	Options		×
	Check Parameter Confirm Memory Size	Project	Tool Hint Display Item in Tool Hint	^
	Add-on Management	Common Item	Data Type Class	No 🔽
	Install Motion Software	ST Editor	Constant	Yes 🗸
	Logging Function	Label Editor Common	Monitor Value	Yes
*	Shortcut Key	🔏 Edit	Instruction Format Instruction Explanation	Yes Ves
*	options	H Find/Replace	Instruction Argument Explanation	Yes 👻
		Monitor	Display in Multiline	Yes 👻
		I Online	Editor Display Items	~
		Back to Default Back to User	Default Set as User Default	Import Export OK Cancel
		BufferMode := Done => CommandAborted => Frror =>	MC_BUFFER_MODEmcAborti	ng

5.1.5 Conversion of all the programs and reflection of public labels

(1) Conversion of all the programs

Select [Convert] \rightarrow [Rebuilt All] in the motion control setting function, and then a message window appears. Click the [OK] button to convert all the programs.

Project Edit Find/Replace Co	nvert View Online Debug Tool Wir Rebuild All	×
: () () () () () () () () () () () () ()	Rebuild All Shift+Alt+F4 Worksheet Execution Order Setting Conversion Setting	
Navigation	Setting Description Options Options Public Label Area Capacity Setting Reflect Public Labels Do Not Use the Same Label Name in Global Label and Local Label Options	
 Basic Setting System Setting System Setting Axis Axes Group U O Data 		el
Calculation Profile Network I/O Program		
FB/FUN E (habel	Checking (PROFILE_DATA) structure definition	
	50 /	100
	Cancel	

(2) Reflection of public labels

When some labels are registered as a public label, reflect the public labels. Select [Convert] \rightarrow [Reflect Public Labels], and then a message window appears. Click the [OK] button to reflect the public labels to the PLC CPU. (Refer to Section 3.14.2.)

Conv	vert View	Online	Debug	Tool	Win
	Rebuild Al Worksheet Setting Public Lab Reflect Pu	l Execution el Area Ca <mark>plic Labels</mark>	Shift Order Set pacity Sett	t+Alt+F tting	4

Edit the PLC CPU program after reflecting the public labels.

5.2 Naming Rules of Labels

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

			Prefix		
Data type		Value range	Local	Global	
Bit	BOOL	FALSE(0), TRUE(1)	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	

[Examples of local labels] Bit: bMoveCMD Double-precision real number: lePosition Arrays: wAxes[16] Timer: tdTimer1

[Examples of global labels] Bit: G_bJogF1 Double-precision real number: G_leVelocity

5.3 Projects

5.3.1 Program names

The following shows the program examples explained in this chapter.



[Motion module]	
Navigation 4 ×	Program name: ServoON_Jog
	(PLC READY, all axes servo ON, JOG operation)
👔 0000:RD78G4 🧈 Basic Setting	Program name: Homing
System Setting	
🗉 쯝 Axis	Program name: Positioning
🖩 🎰 Axes Group	(single axis positioning)
🅼 I/O Data	
🖪 🧟 Calculation Profile	Program name: ContinuousPositioning
🐕 Network I/O	(single axis continuous positioning)
🗏 🔚 Program	
initial	Program name: LinearInterpolation
	(two evic linear interpolation
	(two-axis linear interpolation control)
si ProgramBody	Program name: Synchronous
I I I I I I I I I I I I I I I I I I I	(synchronous control)
🔳 🎬 Positioning	
🔳 🚳 Positioning	Program name: ErrorReset
🖀 Local Label	(error reset)
🚮 ProgramBody	
ContinuousPositioning	
ContinuousPositioning	
Te Local Label	
In ProgramBody	
III IIII ErrorReset	
Fixed Scan	
🚻 Standby	
🕕 No Execution Type	
💼 FB/FUN	
🗉 [Label	

5.3.2 Settings for global and public labels

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

(1) PLC CPU

The PLC CPU uses the public labels of the Motion module.

(2) Motion module

The following shows the global and public label settings of the Motion module.

[Global]



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 transmit the completion signal of each program to the PLC CPU.
- 4) These labels turn ON while the JOG operation is being executed.

They are used as an interlock condition to prevent execution of the JOG operation while other programs are running.

5) These labels stop the axis operation.

The operation is aborted when these labels are turned ON during the operation. (Refer to Section 5.13.4 for details.)

[Sys+Global]

The following is the structure that stores data related to the system. Set "Public label" to "Enable".

ſ	4	Label Name	Data Type	Class	Initial Value	Constant	Comment	Remark	Public Label	Motion Control
	1	System SYSTEM VAR_GLOBAL		<detailed setting=""></detailed>				Enable	-	
	2									

[AX+Global]

These labels are registered automatically when the axis is set in the motion control setting function. No operation is required on the label editor.

4	Label Name	Data Type	Class	Initial Value	Constan -	Comment	Remark	Public Label	Motion Control
1	Axis0001	AXIS_REAL	VAR_GLO BAL	<detailed setting=""></detailed>		[X_Axis]		Disable	-
2	Axis0002	AXIS_REAL	VAR_GLOBAL	<detailed setting=""></detailed>		[Y_Axis]		Disable	-
3	VirtualAxis0001	AXIS_VIRTUAL	VAR_GLO BAL	<detailed setting=""></detailed>		[Vir_Axis01]		Disable	-
4	LinkAxis0001	AXIS_VIRTUAL_LINK	VAR_GLO BAL	<detailed setting=""></detailed>		[Lin_Axis01]		Disable	-
5	LinkAxis0002	AXIS_VIRTUAL_LINK	VAR_GLOBAL	<detailed setting=""></detailed>		[Lin_Axis02]		Disable	-
6									

≈

[Gr+Global]

This label is registered automatically when the axes group is set in the motion control setting function. No operation is required on the label editor.

2	Label Name	Data Type	Class	Initial Value	Constant	Comment	Remark	Public Label	Motion Control
1	AxeeGroup001	AXES_GROUP	VAR GLOBAL	<detailed< th=""><th></th><th>[X-Y Table]</th><th></th><th>Disable</th><th>-</th></detailed<>		[X-Y Table]		Disable	-
2									

[Prf+Global]

These labels are registered automatically when the profile data is registered in the motion control setting function. No operation is required on the label editor.

	Label Name	Data Type	Class	Initial Value	Constant	Comment	Remark	Public Label	Motion Control
1	ProfileData0001	MC_CAM_REF	VAR_GLOBAL	<detailed< th=""><th></th><th>[Cam #1]</th><th></th><th>Disable</th><th>-</th></detailed<>		[Cam #1]		Disable	-
2									

[Prg+Global]

These labels are registered automatically when the program is created in the motion control setting function. No operation is required on the label editor.

	Label Name	Data Type	Class	Initial Value	Constant	Comment	Remark	Public Label	Motion Control
1	ServoON_Jog	PROGRAM_INFO	VAR_GLOBAL	<detailed…< td=""><td></td><td></td><td></td><td>Disable</td><td>-</td></detailed…<>				Disable	-
2	Homing	PROGRAM_INFO	VAR_GLOBAL	<detailed< td=""><td></td><td></td><td></td><td>Disable</td><td>-</td></detailed<>				Disable	-
3	Positioning	PROGRAM_INFO	VAR_GLOBAL	<detailed…< td=""><td></td><td></td><td></td><td>Disable</td><td>-</td></detailed…<>				Disable	-
4	ContinuousPosi…	PROGRAM_INFO	VAR_GLOBAL	<detailed< td=""><td></td><td></td><td></td><td>Disable</td><td>-</td></detailed<>				Disable	-
5	LinearInterpolat…	PROGRAM_INFO	VAR_GLOBAL	<detailed…< td=""><td></td><td></td><td></td><td>Disable</td><td>-</td></detailed…<>				Disable	-
6	Synchronous	PROGRAM_INFO	VAR_GLOBAL	<detailed< td=""><td></td><td></td><td></td><td>Disable</td><td>-</td></detailed<>				Disable	-
7	ErrorReset	PROGRAM_INFO	VAR_GLOBAL	<detailed< td=""><td></td><td></td><td></td><td>Disable</td><td>-</td></detailed<>				Disable	-
8									

[En+Global]

This group is for defining ENUM enumerators, which is automatically registered by the system. No operation is required on the user side.

[SYS_MONI structure]

_	Label Name	Data Type	Class	Initial Value	Constant	Comment	Public Label	Motion Control
1	Addon_AbsSystem	ADDO N_MO NI				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	Addon_FileTransfer	ADDON_MONI				Add-on FileTransfer Monitor	Disable	-

≈

23	BuffermemoryRefreshCycle	CYCLE_MONI		Buffer Memory Refresh Cycle Moni…	Disable	-
24	Environment_UserRootPath	String(127)		User Root Path	Disable	-
25	Error	Bit	0	Motion System Error Detection	Enable	-
26	ErrorHistory	Word [Unsigned]/Bit…	0	Error History Information	Disable	-
27	ErrorHistory_Latest	Word [Signed]	0	Latest Error History Data No.	Disable	-

[SYSTEM structure]

	Label Name	Data Type	Class	Initial Value	Constant	Comment	Public Label	Motion Control Attribute
1	PrConst	SYS_PRM_CONST				System Parameter Constant	Disable	WRITE (=> Motion): PrConst
2	Pr	SYS_PRM				System Parameter	Disable	WRITE (=> Motion): Pr
3	Md	SYS_MONI				System Monitor Data	Enable	READ (Motion =>): Md
4	Cd	Cd SYS_CMD		System Control Data	Disable	WRITE (=> Motion): Cd		
5	LO GGING_REALTIME	LOGGING_REALTIME				Logging Realtime Monitor	Disable	-
6	LoggingRef	LOGGING_REF(110)				Logging	Disable	-
7								

[POINTS]

When the public label settings of the Motion module are changed, reflect the public labels. (Refer to Section 3.14.2.)
5.4 PLC READY (Program Name: ServoON_Jog)

The 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 by a PLC CPU to start operation of the Motion module.

(1) Program example

PLC READY flag [Y0] is turned ON when the PLC CPU is powered ON and the synchronization flag [X1] for the Motion module is turned ON.

[PLC CPU]



[Motion module] None

5.5 Servo ON (Program Name: ServoON_Jog)

This program turns ON the real drive axes connected to the servo system. The following two FBs are used for executing servo ON.

(1) FBs

Type Command		Description		
MCFB	MC_Power	Switches a specified axis to the operation possible status.		
(administrative)	MCv_AllPower	Switches all axes to the operation possible status.		

(2) Local labels

_	Label Name	Data Type	Class	Initial Value	Constant	Comment
1	MCv_AllPower_1	MCv_AllPower	VAR			全軸サーボオンFB / All axes servo ON FB
2	MCv_Jog_1	MCv_Jog	VAR			JOG運転FB1 Axis0001 / JOG operation FB1 Axis0001
З	MCv_Jog_2	MCv_Jog	VAR			JOG運転FB2 Axis0002 / JOG operation FB2 Axis0002
4	leJogAcceleration	FLOAT [Double Precision]	VAR		~	- non速度 / JOG acceleration
5	leJogDeceleration	FLOAT [Double Precision]	VAR I he labe	els for JC	G opera	ation 或速度 / JOG deceleration
6	leJogJerk	FLOAT [Double Precision]	VAR			JOGジャーク / JOG jerk
7						

Drag and drop MCv_AllPower to the program editor. The FB (MCv_AllPower_1) is added as a local label. (Refer to Section 5.1.3.)

(3) Program example

When PLC READY [Y0] is turned ON, Ready [X0] is turned ON. Turning ON of X0 is used as the condition for executing all axes servo ON. When executing servo OFF, turn ON [M0].

[PLC CPU]



[Motion module]



5.6 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.

(1) FB

Туре	Command	Description
MCFB (motion)	MCv_Jog	Executes a JOG operation according to the target velocity.

(2) 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.

(a) Overview

Select the acceleration/deceleration methods from the following.

Method	Description
Acceleration/deceleration specification method (Initial)	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 speed.

(b) Setting method

The acceleration/deceleration method can be specified by the Options input of motion FBs, such as MCv_Jog.

Bit	Description			
0 to 2	Acceleration/deceleration method setting 0: mcAccDec (acceleration/deceleration specification method) 1: mcFixedTime (acceleration/deceleration time-fixed method)			
3 to 15	The function varies depending on the EP			
16 to 31	The function varies depending on the FB.			

(c) Acceleration/deceleration specification method

Select "0: mcAccDec" in the Options input (using bit 0 to 2 for acceleration/deceleration method setting) of the FB, and set acceleration, deceleration, and jerk.

1) Trapezoidal acceleration/deceleration

When "0.0" is set for the Jerk input of the FB, the operation is referred to as the trapezoidal acceleration/deceleration.

The velocity creates a trapezoidal shape.



2) Jerk acceleration/deceleration

When values other than "0.0" is set for the Jerk input of the FB, the operation is referred to as the jerk acceleration/deceleration.

The velocity creates an S-curve shape.



(d) Acceleration/deceleration time-fixed method

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



(e) FB input variables

The following table lists the input variables explained in (b) to (d).

Input variable	Name	Details
Velocity	Target velocity	Specify the target velocity.
Acceleration	Acceleration or acceleration/ deceleration time	 Specify the acceleration or the acceleration/deceleration time. The unit is [U/s²] when the method is set to "0: mcAccDec". The unit is [s] when the method is set to "1: mcFixedTime".
Deceleration	Deceleration	Specify the deceleration. • The unit is [U/s²] when the method is set to "0: mcAccDec". • Not used when the method is set to "1: mcFixedTime"
Jerk	Jerk	Specify the jerk. • The unit is [U/s³] when the method is set to "0: mcAccDec". • Not used when the method is set to "1: mcFixedTime".
Options	Options	Specify the acceleration/deceleration method using bit 0 to 2. 0: mcAccDec Acceleration/deceleration specification method 1: mcFixedTime Acceleration/deceleration time-fixed method

[POINTS] When the "Acceleration/deceleration specification method" (Options: 0) is selected, the acceleration/deceleration can be calculated using the acceleration/deceleration time as follows. [Target velocity: V [U/s], acceleration time [s], deceleration time [s]], Velocity := (target velocity V); Acceleration := (target velocity V/acceleration time); Deceleration := (target velocity V/deceleration time); Options := (mcAccDec);

(3) Local labels

					Tł	ne label i	used in the Servo ON program
		Label Name	Data Type	Class	Initial Value	Constant	Comment
	1	MCv_AllPower_1	MCv_AllPower	VAR			全軸サーボオンFB / All axes servo ON FB
1)	 2	MCv_Jog_1	MCv_Jog	VAR			JOG運転FB1 Axis0001 / JOG operation FB1 Axis0001
1)	3	MCv_Jog_2	MCv_Jog	VAR			JOG運転FB2 Axis0002 / JOG operation FB2 Axis0002
2)	 4	leJogAcceleration	FLOAT [Double Precision]	VAR			JOG加速度 / JOG acceleration
2)	5	leJogDeceleration	FLOAT [Double Precision]	VAR			JOG減速度 / JOG deceleration
	6	leJogJerk	FLOAT [Double Precision]	VAR			JOGジャーク / JOG jerk
	7						

1) These labels are automatically added when the user drags and drops the FB "MCv_Jog" to the program editor.

2) These labels are registered manually.

(4) Program example

The JOG velocity is set from the PLC CPU program since it is registered as a public label ("Write(=>Motion)" is selected). After the PLC CPU is turned to RUN state, the JOG velocity can be changed from the PLC CPU program or on GOT (HMI).

The ON/OFF status of the devices assigned as JOG operation signals (positive/reverse rotation) is transmitted to the Motion module through the public labels.

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

[Start signal]

	Axis0001	Axis0002
JOG positive rotation command	M1	M3
JOG reverse rotation command	M2	M4

Quick Start Guide

[PLC CPU] The public labels transmit the JOG velocity and the start The simultaneous execution of the positive/reverse rotation JOG operation is prevented. signals to the Motion module. The interlock conditions prevent the JOG operation from starting while other programs are running. oolty JOG速度 / JOG Velocity -17 Ň -~~ - + +—Й-V 数月19合 単軸位置決め始約 / 単敏速気位置決め始 の間間 001 / Homine Single axis positioning のが / Single axis positioning か / Single axis positioning start start JOG正転指令 Axis0001 / JOG Positive rotation commend Axis00 innesti . everbe rot Start signal V -0 JOGi豆転指令 Axis0001 / JOG Reverse rotation _____^M__ \neg -v-JOG正転指令 Axis0002 / JOG Positive rotation command Axis00 R078 0000 G bJ 5.版程序会 2种直体/相图影响会 同期影响的合新 / 30022 / Homing 動 / 2-axis linear Synchronous contro mand Ads0002 interpolation control start Revense notatio and Axis0002 Start signal -0 -j/-ΗH JOGIEtEI124 Axis0002 / JOO Reverse rotatic command Axis0 JOG Positive rotation commend Axis0002 Reverse rot The configuration of the axis 2 program is the same as that of the axis 1 program. [Motion module] //----JOG運転用データ設定-------//----Data setting for JOG operation----10 12 IF MCv_AllPower_1.Busy THEN leJogAcceleration := 50000.0; When Axis0001 is changed to servo ON status, 14 leJogDeceleration := 50000.0: the values for JOG acceleration, deceleration, 15 := 0.0; leJogJerk and jerk are stored to the specified labels. 16 END IF; 17 18 //----Axis0001 JOG運転------//----Axis0001 JOG operation----19 20 21 MCv_Jog_1(22 Axis := Axis0001.AxisRef = G_bJogF1CMD , 23 JogForward 24 G bJogR1CMD , JogBackward := These labels are operated from 25 leJogVelocity Velocity := the PLC CPU program. 26 Acceleration := TeJogAcceleration, 27 Deceleration := leJogDeceleration , 28 Jerk := leJogJerk 29 => G_bJog1Busy BUSY This signal is transmitted to the PLC CPU, which is 30); used as an interlock condition of other programs. 31 32 33 //----Axis0002 JOG運転------34 //----Axis0002 JOG operation----35 36 MCv_Jog_2(37 Axis := Axis0002.AxisRef , JogBackward := G_bJogF2CMD, Velocity := G_leJogVelocity, 38 The configuration of the axis 2 39 program is the same as that of the 40 axis 1 program. Acceleration := leJogAcceleration , 41 42 Deceleration := leJogDeceleration , := leJogJerk , => G_bJog2Busy 43 Jerk 44 BUSY 45);

5.7 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 Appendix.

(1) Overview

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

[Time chart of data set type homing (Method37)]

Homing completion	ON	
Servo motor speed	Positive rotation 0 r/min Reverse rotation	Homing position data
Homing operation start	ON OFF	

(2) FB

Туре	Command	Description
MCFB (motion)	MC_Home	Executes homing of the specified axis.

(3) Local labels

		Label Name	Data Type	Class	Initial Value	Constant	Comment
1)	1	MC_Home_1	MC_Home	VAR			原点復帰FB1 Axis0001 / Homing FB1 Axis0001
· (·	 2	MC_Home_2	MC_Home	VAR			原点復帰FB2 Axis0002 / Homing FB2 Axis0002
	3	MC_Home_3	MC_Home	VAR			原点復帰FB3 VirtualAxis0001 / Homing FB3 VirtualAxis0001
	4	bHoming1Done	Bit	VAR			原点復帰FB1 Done出力 / Homing FB1 Done output
	5	bHoming1 Error	Bit	VAR			原点復帰FB1 Error出力 / Homing FB1 Error output
2).	 6	bHoming2Done	Bit	VAR			原点復帰FE2 Done出力 / Homing FE2 Done output
2)	7	bHoming2Error	Bit	VAR			原点復帰FB2 Error出力 / Homing FB2 Error output
	8	bHoming3Done	Bit	VAR			原点復帰FB3 Done出力 / Homing FB3 Done output
	9	bHoming3Error	Bit	VAR			原点復帰FB3 Error出力 / Homing FB3 Error output
	10						

1) These labels are automatically added when the user drags and drops the FB (MC_Home) to the program editor.

2) These labels are registered manually.

(4) Program example

The start signal (the rising edge of the homing command device) is transmitted to the Motion module through the public label. When the homing is completed, the homing completion signal is transmitted to the PLC CPU to turn OFF the start signal.

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

[Start signal]

	Axis0001	Axis0002	VirtualAxis0001
Homing command	M10	M11	M12





[Motion module]

```
//----原点復帰 AxisOOO1----
   //----Homing Axis0001-----
 2
 3
 4
   //原点復帰, Homing
 5
   MC Home 1(
 6
                := Axis0001.AxisRef
       Axis
       Execute := G_bHoming1CMD ,
Position := 0.0 ,
 7
                                        А
 8
                => bHoming1Done,
 9
       Done
10
       Error
                => bHoming1Error
   );
11
12
   //シーケンサに送る原点復帰完了信号をONする
13
   //Transmits homing complete signal to the PLC CPU
14
15
   G_bHoming1Done := bHoming1Done OR bHoming1Error;
                                                          В
16
17
   //----原点復帰 Axis0002----
                                              The configuration of Axis0002 and
   //----Homing Axis0002-----
18
                                             VirtualAxis0001 programs is the same
19
                                                as that of Axis0001 program.
20
   //原点復帰,Homing
21
   MC_Home_2(
22
23
24
                := Axis0002.AxisRef ,
       Axis
       Execute := G_bHoming2CMD .
       Position := 0.0 ,
25
                => bHoming2Done,
       Done
26
                => bHoming2Error
       Error
27
   ):
28
   //シーケンサに送る原点復帰完了信号をONする
29
30
   //Transmits homing complete signal to the PLC CPU
31
   G bHoming2Done := bHoming2Done OR bHoming2Error;
32
33
   //----原点復帰 VirtualAxis0001----
   //----Homing VirtualAxis0001------
34
35
36
   //原点復帰, Homing
37
   MC_Home_3(
38
                := VirtualAxis0001.AxisRef ,
       Axis
       Execute := G_bHoming3CMD ,
39
       Position := 0.0 ,
40
                => bHoming3Done,
41
       Done
                => bHoming3Error
42
       Error
43
   ):
44
45
   //シーケンサに送る原点復帰完了信号をONする
   //Transmits homing complete signal to the PLC CPU
46
47
   G_bHoming3Done := bHoming3Done OR bHoming3Error;
48
```

5.8 Single Axis Positioning Control (Program Name: Positioning)

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

(1) FBs

Туре	Command	Description			
MCFB	MC_MoveAbsolute	Executes positioning using a specified absolute position.			
(motion)	MC_MoveRelative	Executes positioning using a specified relative distance.			

(2) Local labels

	4	Label Name	Data Type	Class	Initial Value	Constant	Comment
1)	 1	MC_MoveRelative_1	MC_MoveRelative	VAR			相対値位置決めFB / Relative value positioning FB
	2	leDistance	FLOAT [Double Precision]	VAR			移動量 / Distance
	3	leVelocity	FLOAT [Double Precision]	VAR			速度 / Velocity
	4	leAcceleration	FLOAT [Double Precision]	VAR			加速度 / Acceleration
	5	leDeceleration	FLOAT [Double Precision]	VAR			減速度 / Deceleration
2)	 6	leJerk	FLOAT [Double Precision]	VAR			ジャーク / Jerk
4)	 7	bDone	Bit	VAR			相対値位置決めFB Done出力 / Relative value positioning FB Done output
	8	bBusy	Bit	VAR			相対値位置決めFB Busy出力 / Relative value positioning FB Busy output
	9	bError	Bit	VAR			相対値位置決めFB Error出力 / Relative value positioning FB Error output
	10	bExecute_P	Bit	VAR			実行指令 / Start
	11	bCommandAborted	Bit	VAR			相対値位置決めFB 実行中断出力 / Relative value positioning FB CommandAborted output
	12						

1) This label is automatically added when the user drags and drops the FB (MC_MoveRelative) to the program editor.

2) These labels are registered manually.

(3) Program example

This program executes relative positioning in the following operation pattern.



The start signal (the rising edge of the single axis positioning command device) is transmitted to the Motion module through the public label. Responding to the start signal, the Motion module stores each positioning data to the specified local label. When all the data are stored, the positioning FB is executed. When the positioning is completed, the completion signal is transmitted to the PLC CPU to turn OFF the start signal.

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]

Single axis positioning command	M20

[PLC CPU]

Start signal	other	programs a	re running c	or when a s	system error of	curs.	transmitte throu	d to the Mo gh the pub	otion moo lic label.
29 (298) Single axis positionine start RD78_0000.G.UP	HD/8_0000.4 pJsp US JOG運動中 Avis0001 / JOG operation in progre Avis0001	B HD/8 0000 G BHomin HD. ののでは 原点復帰指令 単作 Actit0001 / Homine 動。 command Acts0001	as CARD a bit and a start a	Dispitterp RD 19 0000 G Inft(日) Inft(H) Inft(H	a JSynco RD19 0000 System dError b / Mation System Error s control Detection			SET	RD78 0000 G JP0 MD 単軸位置決助給費 Single axis positioning start
11 Carlor Heli GRAVE	7 / oto		(The s	start signal is re g completion si	eset when t gnal is turr	the led ON.	RST	MD 单键位面形的分数 Single axis positioning start M20 Single axis positioning start
[Motion mod 1] //単軸1 2] //Singl	ule] 立置決めま e axis p	判御用データ ositioning	?設定・実行 control(テ指令&リt data sett	セット ing, executio	on, and re	eset)		
4 = IF G_bPosCM 5 IeDista 6 IeVeloc 7 IeAccel 8 IeDecel 9 IeJerk 0 bExecut 1 ELSE 2 bExecut 3 END_IF;	DTHEN: nce ity eration eration e_P := T :e_P := F	<pre>:= 200000 := 50000.0 := 100000 := 0.0; RUE; ALSE;</pre>	.0; D; .0; .0;	When the sitioning da	start signal is t ata is stored to	urned ON, the specifi	each ed label.		
4 5 //単軸f 6 //相対値位f 9 MC_MoveRela 0 Axis 1 Execute 2 Distand 3 Velocit 4 Acceler 5 Deceler	立置決め- e axis p 置決め, R itive_1(ositioning elative va := AxisOO(:= bExecu := leDist; := leVelo; := leDece := leDece	Lue position Lue position Le_P ance sity leration , leration ,	oning	When all t position	he data is s ng FB is e:	stored, the xecuted.)	
		:= leJerk => bDone	, ,						
6 Jerk 7 Done 8 Busy 9 Command 0 Error 1); 2	Aborted	=> bBusy => bComman => bError	, ndAborted	,	The positionir and the co	ng is compl ommand in	eted when -position ar	the Done o e turned O	output N.

5.9 Single Axis Continuous Positioning (Program Name: ContinuousPositioning)

(1) Overview

An axis can continuously execute multiple motion FBs without stopping by buffering the next motion FB of another instance while executing the first motion FB. Select the buffer mode through the BufferMode input of a motion control FB. Up to two motion FBs can be buffered on one axis.

(2) Operation pattern of buffer mode



(3) Local labels



1) These labels are automatically added when the user drags and drops the FB (MC_MoveRelative and TON) to the program editor.

2) These labels are registered manually.

(4) Program example

This program executes relative positioning in the following operation pattern.



The start signal (the rising edge of the single axis continuous positioning command device) is transmitted to the Motion module through the public label. Responding to the start signal, the Motion module stores each positioning data to the specified local label. When all the data are stored, the relative value positioning FBs (MC_MoveRelative) are executed. The Active output of the first FB is connected to the Execute input of the next FB, which enables buffering of the next FB while executing the first FB. When the first FB is finished, the next FB is continuously executed. To set dwell time, the on-delay timer (100 [ms]) is used.

The completion signal is transmitted to the PLC CPU to turn OFF the start signal when any of the following conditions is 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]
--------	---------

Single axis continuous positioning command	M21
--	-----



[POINTS]

Instead of connecting a label, output signals of a FB can be directly entered by describing "(FB name). (output signal name)" in the editor.

5.10 Interpolation Control (Program Name: LinearInterpolation)

5.10.1 Procedure for interpolation control

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



5.10.2 Enabling/disabling an axes group

Refer to Section 3.12 for axes group setting. To execute interpolation control, change the axes group status to "4: GroupStandby".

(1) FBs

Туре	Command	Description
MCFB	MC_GroupEnable	Transits the specified axes group status from "0: GroupDisabled" to "4: GroupStandby".
(administrative)	MC_GroupDisable	Transits the specified axes group status to "0: GroupDisabled".

5.10.3 Interpolation control

The FBs for linear and circular interpolation control are as follows. Execute the FB after the axes group is enabled.

Туре	Command	Description		
	MCv_MoveLinearInterpolateAbsolute	Absolute value linear interpolation control		
MCFB	MCv_MoveLinearInterpolateRelative	Relative value linear interpolation control		
(motion)	MCv_MoveCircularInterpolateAbsolute	Absolute value circular interpolation control		
	MCv_MoveCircularInterpolateRelative	Relative value circular interpolation control		

5.10.4 Program example for linear interpolation

(1) Operation pattern

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



- (2) Axis No. and movement amount settings
 - (a) LinearAxes input of MCv_MoveLinearInterpolateRelative
 - Use the array of INT (signed word) type with 16 elements.

In the sample program, wAxes[0..15] (label) is used.

The interpolation control axes are selected from the structuring axis [1] to [16] set in the axes group (in Section 3.12). Specify the structuring axis No. to be used for the interpolation control by wAxes. (Note that the wAxes must be set from the index No. [0] in order.)

(b) Distance input

Use the array of LREAL (double-precision real number) type with 16 elements In the sample program, lePosition[0..15] is used.

Define the movement amount of the structuring axis [1] to [16] in lePosition[0] to lePosition[15].

[POINTS]

Regardless of the number of the interpolation control axes, the number of elements must be 16 for the INT-type array for the LinearAxes input and the LREAL-type array for the Distance input.

[Setting example 1]

Axes group

Structuring axis [1]: Axis0001, structuring axis [2]: Axis0002, and structuring axis [3]: Axis0003

Linear interpolation
 Axis0001 and Axis0002

Select <u>F</u> older Disp	lay All Data 🗸 🧹
Item	AxesGroup001
Axes Group No.	1
Axes Group Parameter	Expands initial values at ax
Acceleration Limit Valu	2147488647.0 pulse/s ²
Operation Selection at	-1:Error (Not Started)
Structuring Axis[1]	Axis0001
Structuring Axis[2]	Axis0002
Structuring Axis[3]	Axis0003
Structuring Axis[4]	
Structuring Axis[5]	
Structuring Axis[6]	
Structuring Axis[7]	
Structuring Axis[8]	
Structuring Axis[9]	
Structuring Axis[10]	
Structuring Axis[11]	
Structuring Axis[12]	
Structuring Axis[13]	
Structuring Axis[14]	
Structuring Axis[15]	
Structuring Axis[16]	

wAxes[0] := 1; (← structuring axis [1]) wAxes[1] := 2; (← structuring axis [2]) lePosition[0] := (movement amount of structuring axis 1 (=Axis0001)); lePosition[1] := (movement amount of structuring axis 2 (=Axis0002));

[Setting example 2]

Axes group

Structuring axis [1]: Axis0001, structuring axis [2]: Axis0002,

and structuring axis [3]: Axis0003

Linear interpolation
 Axis0002 and Axis0003

Select Folder Display All Data 🗸		wAxes[1] := 3; (\leftarrow structuring axis [3])
Item	AxesGroup001	
Axes Group No.	1	lePosition[0] := 0.0;
Axes Group Parameter	Expands initial values at ax	loDopition[1] = (movement emount of structuring
Acceleration Limit Valu	2147483647.0 pulse/s ²	iePosition[1] (movement amount of structuring
Operation Selection at \$	-1:Error (Not Started)	axis 2 (=Axis0002)):
Structuring Axis[1]	Axis0001	
Structuring Axis[2]	Axis0002	lePosition[2] := (movement amount of structuring
Structuring Axis[3]	Axis0003	axis 3 (= Δ xis 0003)).
Structuring Axis[4]		
Structuring Axis[5]		
Structuring Axis[6]		
Structuring Axis[7]		
Structuring Axis[8]		
Structuring Axis[9]		
Structuring Axis[10]		
Structuring Axis[11]		
Structuring Axis[12]		
Structuring Axis[13]		
Structuring Axis[14]		
Structuring Axis[15]		

(3) Local label

	[4	Label Name	Data Type	Class	Initia	Con	Comment
4		1	MCv_MoveLinearInterpolateRelative_1	MCv_MoveLinear	VAR			相対値直線補間制御FB1 / Relative value linear interpolation control FB1
I)		2	MCv_MoveLinearInterpolateRelative_2	MCv_MoveLinear…	VAR			相対値直線補間制御FB2 / Relative value linear interpolation control FB2
		3	MC_GroupEnable_1	MC GroupEnable	VAR			軸グループ有効FB / Axes group enable FB
		4	MC_GroupDisable_1	MC_GroupDisable	VAR			軸グループ無効FB / Axes group disable FB
		5	wAxes	Word [Signed](0	VAR			補間軸 / Interpolation axis
		6	lePosition1	FLOAT [Double…	VAR			位置データ1 / Position data 1
		7	lePosition2	FLOAT [Double…	VAR			位置デーダ2 / Position data 2
		8	leVelocity	FLOAT [Double…	VAR			速度 / Velocity
		9	leAcceleration	FLOAT [Double…	VAR			加速度 / Acceleration
		10	leDeceleration	FLOAT [Double…	VAR			減速度 / Deceleration
		11	leJerk	FLOAT [Double…	VAR			ジャーク / Jerk
		12	bGroupEnableDone	Bít	VAR			軸グループ有効完了 / Axes group enable done
2)		13	bGroupDisableDone	Bít	VAR			軸グループ無効完了 / Axes group disable done
Z)		14	bBusy1	Bit	VAR			相対値直線補間制卸FB1 Busy出力 / Relative value linear interpolation control FB1 Busy output
		15	bActive1	Bit	VAR			相対値直線補間制御FB1 Active出力 / Relative value linear interpolation control FB1 Active output
		16	bDone2	Bít	VAR			相対値直線補間制御FB2 Done出力 / Relative value linear interpolation control FB2 Done output
		17	bBusy2	Bít	VAR			相対値直線補間制御FB2 Busy出力 / Relative value linear interpolation control FB2 Busy output
		18	TON_1	TON	VAR			オンディレイタイマFB/On-delay timer FB
		19	bDwell_out	Bit	VAR			화구·미미· / Timer output
		20	bDwell_in	Bit	VAR			화구ス力 / Timer input
		21	bExcute_LP	Bit	VAR			実行指令 / Start
		22	bCommandAborted	Bít	VAR			相対値直線補間制御FB実行中断出力 / Relative value linear interpolation control FB CommandAborted output
		23	bError	Bít	VAR			相対値直線補間制卸FB Error出力 / Relative value linear interpolation control FB Error output
		24						

1) These labels are automatically added when the user drags and drops the corresponding FB to the program editor.

2) These labels are registered manually.

(4) Program example

	Program description	Module
1)	The start signal (the rising edge of the two-axis linear interpolation control start device) is transmitted to the Motion module through the public label.	PLC CPU
2)	 Responding to the start signal, the Motion module stores the positioning data to the specified local labels. When all the data are stored, the axes group is enabled. After the axes group is enabled, the two relative value positioning FBs (MCv_MoveLinearInterpolateRelative) are started in buffer mode. The axes group is disabled when any of the following conditions is satisfied: the dwell time is passed; an error occurs; or the execution is aborted. When the axes group is disabled, the start signal and the on-delay timer input are reset. 	Motion module
3)	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.	PLC CPU

[Start signal]

Two-axis linear interpolation control start	M22
	IVIZZ

[PLC CPU]





(Note): When the values for the Velocity, Acceleration, and Deceleration inputs are omitted, those of the previous FB are applied.

- 5.11 Synchronous Control (Program Name: Synchronous)
- 5.11.1 Procedures for executing synchronous control

The following shows the procedure flow for executing synchronous control.



5.11.2 Calculation profile

Waveform data used for control is collectively called calculation profile data. This section explains how to create a cam data.

(1) Creating a new calculation profile

In the navigation window of the motion control setting function, right-click "Calculation Profile" and select "Add New Data".

Calculation Profile Calculation Profile ProfileData0001 [Cam #1]	Add New Data	a Ins	 New Data		×
W Network I/O	New Folder	Ctrl+Shift+N	Basic Setting		
🗉 🚾 Program	Import File		Data Type	Calculation Profile	-
🚰 FB/FUN	Sort	•	(Data Name)	ProfileData0001	
🖬 👫 Label	3010				
			Detailed Setting		
			Data Format		
			Туре	Cam Data	-
			Interpolation Method Specification	Section Interpolation	-
			Expand Setting		
			Auto Expand	Yes	-
			Profile ID (1 to 60000)	1	
			Repetitive Operation	Enable	-
			Input Absolute Coordinate	Disable (Relative Coordinate)	-
			Output Absolute Coordinate	Disable (Relative Coordinate)	-
				OK Cancel	

The following shows the setting items on the "New data" screen.

Item	Description
Automatic open	Yes: The calculation profile data is automatically opened at power-on. No: The open FB for calculation profile data needs to be executed.
Periodic	Invalid: The control ends when it executes until the end of calculation profile data. Valid: The execution of calculation profile data is continuously repeated.
Output absolute coordinate	Invalid (relative): When the calculation profile (cam) is started, an output value is calculated based on the current value. When executing a feed cam operation, select this setting. Valid (absolute): The output value at the time the calculation profile (cam) is started is calculated to be always the start point for one cycle of the calculation profile data. When the start point and the end point of the calculation 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.

The example in this document uses the initial value. Click the [OK] button.

(2) Creating a cam data

Set the calculation profile waveform.



The following shows the necessary setting items.

No.	Item	Description
1	Resolution Set the resolution of the cam data.	
a Longth por Cyclo		Set the length per cycle and its unit.
2		(Set the movement amount of the master axis for one cam cycle)
2	Straka Amaunt	Set the stroke amount and its unit.
3 Stroke Amount		(Set the movement amount of the slave axis for one cam cycle)
		Set the time for one cam cycle.
4	Cam Time per Cycle	This setting is used when velocity, acceleration, and jerk are
		calculated.
5	Stroke Setting	Set the stroke.

The following shows the example settings.

No.	Item	Setting value
1	Resolution	256
2	Length per Cycle	200000 (blank for the unit setting)
3	Stroke Amount	200000 (blank for the unit setting)
4	Cam Time per Cycle	1.000
5	Stroke Setting	Refer to the following table.

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

[POINTS]

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

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

5.11.3 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.

(1) FBs

Туре	Command	Description
	MC_CamIn	Executes cam operation.
	MC_GearIn	Executes gear operation based on the specified speed ratio between the master axis and the slave axis.
	MC_CombineAxes	Combines motion of two axes by a selectable combination method, and outputs the result to the third axis.
MCFB (motion)	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.
	MCv_*****Filter	Executes the specific filter processing to the input of Master, and outputs the result to Slave.
	MC_Stop	Stops the synchronous control.

5.11.4 Axes configuration

This chapter explains the following cam system.



5.11.5 Program example

(1) Operation pattern

The X-axis moves from the origin point for 400000.0 [um] while the Y-axis (Axis0002) is operated according to the cam pattern created in Section 5.11.2. 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.



(2) Local labels

		Label Name	Data Type	Class	Initial	Const	Comment
	1	MC_GearIn_1	MC_GearIn	VAR			ギア動作FB / Gear operation FB
	2	MC CombineAxes 1	MC CombineAxes	VAR			加減算位置決めFB / FB combining the motion of two master axes
	3	MC_CamIn_1	MC_CamIn	VAR			力ム動作FB / Cam operation FB
	4	bGearInBusy	Bit	VAR			ギア動作FB Busy出力 / Gear operation FB Busy output
1)	5	bCombineAxesBusy	Bit	VAR			加減算位置決め動作FB Busy出力 / FB combining the motion of two master axes Busy output
•)	6	bCamInBusy	Bit	VAR			力ム動作FB Busy出力 / Cam operation FB Busy output
	7	MC_MoveRelative_1	MC_MoveRelative	VAR			相対値位置決めFB / Relative value positioning FB
	8	MC_MoveRelative_2	MC_MoveRelative	VAR			相対値位置決めFB / Relative value positioning FB
	9	MC_Stop_1	MC_Stop	VAR			軸停止FB1 / Axis Stop FB1
	10	MC_Stop_2	MC_Stop	VAR			軸停止FB2 / Axis Stop FB2
	11	MC_Stop_3	MC_Stop	VAR			軸停止FB3 / Axis Stop FB3
	12	leAcceleration	FLOAT [Double Pr…	VAR			加速度 / Acceleration
	13	leDeceleration	FLOAT [Double Pr···	VAR			減速度 / Deceleration
	14	leJerk	FLOAT [Double Pr…	VAR			ジャーク / Jerk
	15	lePosition1	FLOAT [Double Pr…	VAR			移動量 / Distance
	16	leVelocity	FLOAT [Double Pr···	VAR			速度 / Velocity
	17	blnSync	Bit	VAR			力ム動作FB inSync出力 / Cam operation FB inSync output
	18	bDone1	Bit	VAR			相対値位置決めFB1 Done出力 / Relative value positioning FB1 Done output
	19	bBusy1	Bit	VAR			相対値位置決めFB1 Busy出力 / Relative value positioning FB1 Busy output
2)	20	bDone2	Bit	VAR			相対値位置決めFB2 Done出力 / Relative value positioning FB2 Done output
4)	21	bBusy2	Bit	VAR			相対値位置決めFB2 Busy出力 / Relative value positioning FB2 Busy output
	22	bStopReq1	Bit	VAR			軸停止要求1 /Axis Stop request 1
	23	bStopDone1	Bit	VAR			軸停止完了1 / Axis Stop complete 1
	24	bStopDone2	Bit	VAR			軸停止完了2 /Axis Stop complete 2
	25	bStopDone3	Bit	VAR			軸停止完了3 / Axis Stop complete 3
	26	bSyncMoveCMD	Bit	VAR			相対位置決め始動 / Relative value positioning start
	27	bExecute_S	Bit	VAR			実行指令 / Start
	28	bCommandAborted	Bit	VAR			相対値位置決めFB 実行中断出力 / Relative value positioning FB CommandAborted output
	29	bError	Bit	VAR			相対値位置決めFB Error出力 / Relative value positioning FB Error output
	30						

1) These labels are automatically added when the user drags and drops the corresponding FB to the program editor.

2) These labels are registered manually.

(3) Program example

	Program description	Module
1)	The start signal (the rising edge of the synchronous control start device) is transmitted to the Motion module through the public label.	PLC CPU
2)	 Responding to the start signal, the Motion module stores the positioning data to the specified labels. When all the data are stored, the execution signals of the single axis synchronous control FBs (MC_GearIn, MC_CombineAxes, and MC_CamIn) are turned ON. When the Axis0002 status is turned to "7: SynchronizedMotion", Axis0001 (Master axis) starts positioning. At this time, Axis0002 is operated according to the specified calculation 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 calculation profile. The synchronous control is stopped by executing MC_Stop on the real drive axis (Axis0002) and the virtual linked axes (LinkAxis0001 and LinkAxis0002) when any of the following conditions is 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. 	Motion module
3)	The interlock conditions are added to prevent execution of the synchronous control while other programs are running or when a system error occurs.	PLC CPU

[Start signal]

Synchronous control start	M23
---------------------------	-----

[PLC CPU]



Quick Start Guide

[Motion module]



(To the next page)

(Continued)



(Continued)



5.12 Error Reset (Program Name: ErrorReset)

This program resets the error on each axis.

(1) FBs

Туре	Command	Description
	MC_Reset	Resets errors and warnings of the axis.
MCFB (administrative)	MC_GroupReset	Resets errors and warnings of the axes group.
(ddministrative)	MCv_MotionErrorReset	Reset all errors and warnings of the motion system.

(2) Program example

The following shows the program example for error reset.

The ON/OFF status of the device assigned as the start signal of the error reset is transmitted to the Motion module through the public label.

[PLC CPU]



14

5.13 Checking Operation

5.13.1 Conversion and Writing of Programs

Write the programs to the PLC CPU and the Motion module. After the programs are created in MELSOFT GX Works3 and the motion control setting function, convert all the programs from [Convert] \rightarrow [Rebuild All].

Confirm that no error occurs after executing "Rebuild All", and write the program to the PLC CPU and the Motion module.

- MELSOFT GX Works3: [Online] \rightarrow [Write to PLC]
- Motion control setting function: [Online] \rightarrow [Write to Module]



[MELSOFT GX Works3]

(Note): When the parameters and the public labels of the Motion module are changed, convert and write the programs to the PLC CPU after reflecting the public labels.

5.13.2 Axis monitor

The 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.

(1) Displaying monitor screens

Select [Online] \rightarrow [Motion Monitor] \rightarrow [Axis Monitor] in the motion control setting function.

(2) Monitor screen

💐 0000:RD78G4 - Axis Monito	or					_	- -
· 🔥 🛛 🗖 🗞 📖 🜉	3					2	
	<u> </u>		4			_	
Axis Monitor Monitor	Type: Real Drive Axis	Display Selection	≒_			Syste	em Monitor
Fon	t Size: 9pt 🗸 🗸	Monitor Item	election Monitor Axis Selection	n Split Opper Pane	O Lower Pane	4	Monitor Item Selection
	Axis #1	Axis #2			^	0	PLC Ready
Axis Name	Axis0001	Axis0002				0	Ready
Axis Status	0:Axis Disabled	0:Axis Disabled					Synchronization flag
Control Cycle	1	1					System Paris Cycle Monitor Proces
Position Command Unit Display	μm	μm					3800
Speed Command Unit Display	µm/s	µm/s					System Basic Cycle Monitor.Maxim.
Command Current Position	0.0 um	0.0 um					4760
Feedback Position	0.0 um	0.0 um					System Basic Cycle Monitor.Settin.
Specified Position	0.0 um	0.0 um					1000000
Command Current Speed	0.0 um/s	0.0 um/s				0	System Basic Cycle Monitor.Cycle .
Feedback Speed	0.0 um/s	0.0 um/s					Operation Cycle Monitor[1].Proces.
Specified Speed	0.0 um/s	0.0 um/s					62440
Negative Direction Speed Limit Value	2500000000.0 um/s	2500000000.0 um/s					Operation Cycle Monitor[1].Maxim. 67840
Positive Direction Speed Limit Value	2500000000.0 um/s	250000000.0 um/s					Operation Cycle Monitor[1].Settin
Automatically Decelerating	FALSE	FALSE					Occurrentian Curle Manifestial Curle
Command In-position	FALSE	FALSE					Operation Cycle Monitor[1].Cycle
Negative Direction Torque Limit Value	300.0 %	300.0 %				@ @	Forced Stop Cancelling Motion System Error Detection
Positive Direction Torque Limit Value	300.0 %	300.0 %					Latest Motion System Error Code
Execution Profile ID No.	0	0				_	000
Home Position Return Completed	FALSE	FALSE				0	Motion System Warning Detection Latest Motion System Warning Cod
Home Position Return Request	FALSE	FALSE				0	000
Start Permission at Home Position Return Uncompleted	FALSE	FALSE					Network Error Code
Upper Limit Signal Status	FALSE	FALSE					00
Lower Limit Signal Status	FALSE	FALSE					Basic System Software Version
Forced Stop Cancelling	TRUE	TRUE					
Axis Error Detection	FALSE	FALSE					Boot Software Version
Axis Error Code	0000	0000					
Axis Warning Detection	FALSE	FALSE					Network Boot Software Version
Axis Warning Code	0000	0000					
Driver Ready On Status	FALSE	FALSE			~	<	

No.	Description
1	The monitor items for each axis are displayed.
2	The system monitor items are displayed.
3	The axis type to be monitored can be changed.
4	The monitor items can be added/deleted.

5.13.3 Program monitor

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

(1) Displaying monitor screen

[PLC CPU]

Select [Online] \rightarrow [Monitor] \rightarrow [Start Monitoring (All Windows)] in MELSOFT GX Works3, or click the icon [🕎] on the tool bar.

[Motion module]

Select [Online] \rightarrow [Monitor] \rightarrow [Start Monitoring (All Windows)] in the motion control setting function, or click the icon [\mathbf{R}] on the tool bar.

(2) Monitor screen

The following shows the monitor screen of the motion control setting function.


5.13.4 Watch

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

- (1) Displaying monitor screens
 - [PLC CPU]

Select [View] \rightarrow [Docking Window] \rightarrow [Watch 1 to Watch 4] in MELSOFT GX Works3.

[Motion module]

Select [View] \rightarrow [Docking Window] \rightarrow [Watch 1 to Watch 4] in the motion control setting function.





[Motion control setting function]

[MELSOFT GX Works3]

(2) Registration of labels and structures to a watch window

Enter the label or the structure to be registered in the "Name" column of a watch window or right-click a label or a structure to be registered on the program editor, and select [Register to Watch Window] \rightarrow [Watch Window 1 to 4].



(3) Monitoring start

Select [Online] \rightarrow [Watch] \rightarrow [Start Watching] in MELSOFT GX Works3 or the motion control setting function.

(4) Changing current values

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

The bit device can be switched to ON/OFF with double-click while holding the [Shift] key or with [Shift] + [Enter] after selecting the row.

[POINTS]

In the sample program, "G_bStopSignalX" and "G_bStopSignalY" are registered in the watch window 1. Turning these signals ON stops the axis operation.

When "G_bStopSignalX" and "G_bStopSignalY" are operated from the PLC CPU, change the motion control attribute setting from "READ(Motion=>)" to "WRITE(=>Motion)", and reflect the public label.

5.13.5 Event History

The event history of the Motion module can be checked from [Online] \rightarrow [Motion Monitor] \rightarrow [Event History] in the 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.

Events t	Refresh(U)		Number of Events:1	.58	Refin	e(D)	8		
Refine	tch All the Conditions		O Match #	Iny One of t	he Conditions				
1. Ev	vent Type	✓ Includir	ig Next		~			~	
2.		~			~			~	
3.		~			\sim			~	
_ Inc	clude program errors (Al	low jumping)			Start Refi	ne	Clear Refine Con	ditions	
lo.	Occurrence Date		Event Type	Status	Event Code	Overv	/iew		
001	2020/09/07 07:33:1	9.402538560	System	4	007F0	MCFB :	Start (Control System)		
002	2020/09/07 07:33:1	9.391777656	System	4	007EE Servo System Recorder Start				
003	2020/09/07 07:33:1	9.289555680	System	4	007FE	Homing	Homing Request OFF to ON		
004	2020/09/07 07:33:1	9.289115064	System	•	007FD Current Position Restoration C			tion	
005	2020/09/07 07:33:1	9.289032336	System	•	007FD Current Position Restoration C			tion	
006	2020/09/07 07:33:1	9.288934016	System	۰	007FD	Curren	t Position Restoration Complet	tion	
007	2020/09/04 08:46:0	3.947703512	System	1	007F0	MCFB	Start (Control System)		
								>	
gend	🛕 Major	🔥 Mod	erate 🔥 Mino	vr				Jump	
	Uarning	🤹 🔅 Info	mation					Clear All	
Det	ailed Information	MCFB Star	t/Stop Information		-		-		
		Detail Code FB Type :N	e :0 4Cv AllPower						
		FB Instance	e ID :45430288	_	-		-		
		Instance N	ame mov_AliPowe						
	Cause	MCFB was	started.						
C	orrective Action								

To check the event history of the PLC CPU, click [Diagnosis] \rightarrow [System monitor] in MELSOFT GX Works3, and click "Event history" on the "System monitor" screen.

MEMO

APPENDICES

APPENDIX 1 Diverting Programs

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

APPENDIX 1.1 Exporting the Motion module data

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

 Start MELSOFT GX Works3, and open the diverting data. In the navigation window, double-click "Parameter" → "Module information". Right-click "0000_RD78G4", and select [Export Motion Module Data].



2) On the "Export Motion Module Data" screen, select the folder to export, and click the [Save] button.

On the confirmation screen, click the [OK] button to output the Motion module data to an muw file.



APPENDIX 1.2 Saving servo parameters

Servo parameters are not included in the muw file outputted in Appendix 1.1. When writing the servo parameters from the Motion module to the servo amplifier, save the servo parameters of the diverting program in a separate file.

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

MELSOFT GX Works3								-		×
: Project View Eile Paramet	ter Setting(Z) Parameter	Mindow He	elp							
: 🖉 🙆 : 📆 😓 🖪										
										4.5
; Project + A	Parameter Setting) ×								4 P 🗸
	Parameter Setting	:								
Parameter	Carting a state	land 🔊 Sat					1			
Network Parame			. To Delauit	Werny a Parameter Copy in Parameter block						
Station2:MR-J5-G(-F	🔁 Oper 🂾 Save As									
Parameter	🖃 🚟 Function display (L	C				a la stra sura				
🔤 🗎 Network Parame	Common	Common				Selected Items Write	Axis writing			
	Position/Speed/To	No.	Abbr.	Name	Unit	Setting range	Station 1	Station2	_ <u> </u>	
	Servo adjustments	Operation	mode							
	Eanie annalifier dia	PA01.1	**	Operation mode selection		0-8	0 : Standard control 👻	0 : Standard control	-	
	Machine diagnosi	PA01.4	**	Fully closed operation mode selection		0-1	0 : Disabled (Semi dd 👻	0 : Disabled (Semi clo		
	Linear control	Basic								
	DD Motor control	Componen	t parts				Setting	Setting		
	- Fully closed contro	PA02.0-1	**	Regenerative option selection		00-FF	00 : Regen. option is 👻	00 : Regen. option is	•	
	🖃 🥅 List display	PC02	MBR	Electromagnetic brake sequence output		0-1000	0		0	
	Basic	PC04.3	**	Encoder cable communication method selection		0-1	0:2-wire 👻	0 : 2-wire	•	
	Gain/filter	Rotation d	irection							
	Extension	PA14	*POL	Moving direction selection		0-1	0 : CCW dir. during f 🚽	0 : CCW dir. during f	-	
		PC29.3	*	Torque POL reflection selection		0-1	1 : Disabled	1 : Disabled	<u> </u>	
	Extension 2	Zero speed	4							
	Ontion setting	PC07	ZSP	Zero speed		0-10000	50		50	
	Special	PA04.2	*	Servo forced stop selection		0-1	1 : Disabled (The f 👻	1 : Disabled (The	-	l l
	Motor extension	Forced sto	p decelerati	on function						l l
	- Multi encoder	PA04.3	*	Forced stop deceleration function selection		0-2	2 : Forced stop dece 👻	2 : Forced stop dece	-	
	Positioning contro	PC24	RSBR	Forced stop deceleration time constant		0-20000	100		100	
	- Network setting	Vertical ax	is freefall pr	evention						l l
	Positioning extens	PC02	MBR DSLID1	Electromagnetic brake sequence output		-25000-25000	0		0	
		Alarm sett	ing	rendear axin certair prevention compensation anioant		23000 23000				l l
		PC08	OSL	Overspeed alarm detection level		0-20000	0		0	
		PC21.0	*	Alarm history clear selection		0-1	0 : Disabled	0 : Disabled	-	l l
	<	Encoder or	utput pulse j	phase setting			Setting	Setting	~	
<										2
Ready		Unit	t connection					OVR CAP	NUM	SCRL /

2) On the "Save As" screen, select a folder and click the [Save] button. The servo parameters are outputted to a prm2 file.

📶 Save As				$\mathbf{\overline{N}}$
Save in:	Documents	~	G 🤌 📂 🛄-	
3	Name 🔺		Date modified	Туре
Quick access		No items match your s	earch.	
Desktop				
-				
Libraries				
This PC				
1				
Network	<			>
	File <u>n</u> ame:	SVsystem	<u> </u>	<u>S</u> ave
	Save as type:	User Parameter Files(*.pm2)	~	Cancel

APPENDIX 1.3 Reflection to the editing program

Start MELSOFT GX Works3, and create a module configuration diagram of the editing program.

- (1) Importing the Motion module data
 - 1) In the navigation window, double-click "Parameter" → "Module information". Right-click "0010:RD78GHV", and select [Import Motion Module Data].



2) On the "Import Motion Module Data" screen, select the muw file outputted in Appendix 1.1, and click the [Open] button.

On the confirmation screen, click the [OK] button to complete the import of the Motion module data.



(2) Reading servo parameters

To read the servo parameters from the Motion module, read the servo parameter file outputted in Appendix 1.2.

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



2) On the "Open" screen, select the prm2 file outputted in Appendix 1.2, and click the [Open] button.

📶 Open				
Look in:	Documents	~	G 🤌 📂 🛄 -	
3	Name 🔺		Date modified	Туре
Quick access	SVsystem.prr	n2	2020/08/21 13:28	PRM2 File
Desktop				
-				
Libraries				
Lange Contract This PC				
۲				
Network	<			>
	File name:	SVsystem		Open
	Files of type:	User Parameter Files(*.pm2;*.pm)	✓	Cancel

Open File	X
Copy source	SVsystem.prm2
Model	MR-J5-G(-RJ)
Copy destinati	on
Station 1	
Station2	
	Select All Cancel All
Parameter Dis	play Copy Close(Q)

3) Select the stations to read the servo parameters, and click the [Copy] button.

4) Make sure that the servo parameters are read, and click the [×] button on the right top of the "Parameter Setting" screen.

MELSOFT GX Works3								-		×
Eroject View Ele Paramet	ter Setting(<u>Z)</u> P <u>a</u> rameter <u>V</u>	(indow H	elp							
🛃 🕜 🗄 🌄 🔂 🖪										
Project 4 ×	Parameter Setting	×							4	₽.
⊟ 0000	Parameter Setting									
Station1:MR-J5-G(-F		und ID co	t To Defeudt	- Varife - December Const December Black						
Network Parame			t to belaut	Soverny all Parameter copy a Parameter block						
🖃 陆 Station2:MR-J5-G(-F	; Den Save As									
Parameter	E Function display (L	Common				Selected Items Write	Axis Writing			
Network Parame	- Position/Speed/To	No.	Abbr.	Name	Unit	Setting range	Station1	Station2	^	
	- Servo adjustments	Operation	mode							
	I/O	Operation	mode	Occurrence and a selection			0 · Standard control -	0 · Standard control		
	 Servo amplifier dia 	PA01.1		Operation mode selection		8-0	0 : Disabled (Semi da	0 - Displied (Semi de	•	
	- Machine diagnosis	PA01.4	**	Fully closed operation mode selection		0-1	0 . Disabled (Senii dd 🗸	o . Disabled (Semi de	-	
	- Linear control	Compone	nt parts				Setting	Setting		
	- Fully closed control	PA02.0-1	**	Regenerative option selection		00-FF	00 : Regen. option is 🚽	00 : Regen. option is	-	
	🖃 🎆 List display	PC02	MBR	Electromagnetic brake sequence output		0-1000	0		0	
	- Basic	PC04.3	**	Encoder cable communication method selection		0-1	0 : 2-wire 👻	0 : 2-wire	-	
	Gain/filter	Rotation of	direction							
	Extension	PA14	*POL	Moving direction selection		0-1	0 : CCW dir. during f 🚽	0 : CCW dir. during f	•	
	I/O	PC29.3	*	Torque POL reflection selection		0-1	1 : Disabled	1 : Disabled	•	
	Extension 3	Zero spee	d	B		0.10000				
	- Option setting	Forced st	25P	Zero speed		0-10000	50		50	
	- Special	PA04.2	*	Servo forced stop selection		0-1	1 : Disabled (The f 🔻	1 : Disabled (The f	-	
	- Motor extension	Forced st	op decelerat	on function						
	- Multi encoder	PA04.3	•	Forced stop deceleration function selection		0-2	2 : Forced stop dece	2 : Forced stop dece	-	
	 Positioning contro 	PC24	RSBR	Forced stop deceleration time constant		0-20000	100	1	00	
	- Network setting	PC02	MBR	Electromagnetic brake sequence output		0-1000	0		0	
	- Positioning extens	PC31	RSUP1	Vertical ax.freefall prevention compensation amount		-25000-25000	0		0	
		Alarm set	ting							
		PC08	OSL	Overspeed alarm detection level		0-20000	0 · Disabled	0 · Disabled	0	
		PC21.0	-	Alarm history clear selection		0-1		C-W		
		Encoder o	utput pulse	phase setting			Setting	Setting	×	

Click the [Yes] button on the confirmation screen to update the parameters.



(3) Fixing parameters

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

1) Click [Close with Reflecting the Setting] on the "CC-Link IE TSN Configuration" window to reflect the data.

8	🖧 CC-Link IE TSN Configuration (Start I/O: 0000) — 🗆 🗙												-		×
: 0	2-Linl	k <u>i</u> e tsn c	Configuration	dit <u>V</u>	iew Close v	with Disca	ardi <u>ng</u> the Setting	Close with <u>F</u>	eflecting the	Setting					X
	Co	onnected	/Disconnected I	Module	Detection	(Detailed Display						Module List		
	Мо	de Settin	ig:	C	nline		<u>A</u> ssig	nment Meth	nod:		\sim		CC-Link IE TSN Selection	Find Me	odu 🌗
_	Cyc	lic Transr	mission Time (Mir	n.):	-	us	Com	munication P	eriod Interva	l (Min.):	- us		🛅 💱 🧏 📴 🖄 🛯		
		No.	Model Name	STA#	Station	Туре	Motion Control Station	RX Setting	RY Setting	RWr Setting	RWw Setting	Automa	General CC-Link IE	TSN Mo	dule
		0	Host Station	0	Master Sta	ition	ocación	FUILS	FUILS	Foints	Points		CC-Link IE ISN Mod	Jule (Mit adula	tsubishi
▼		o 1	MR-J5-G	1	Remote St	tation				24	1 20		Master/Local M Motion Module	baule	
		<mark>ь</mark> 2	MR-J5-G	2	Remote St	tation	\checkmark			24	4 20	\checkmark	FI GOT2000 Series		
自馬 S S T L L	TA#0 tation otal S ne/St) Master n TA#:2 car	STA#1	STA#2	2							>	DC Input Transistor Outp Analog Input Analog Output General Purpose Gift General-Purpose I/O Combined	ıt > Inverta > AC Ser	er rvo
			<									>			

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



(4) Correcting programs

If the start I/O number of the diverting program 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

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 as in Appendix 1.4, change all of the label names in the program using character string replacement.

APPENDIX 1.4 Conversion of All the Programs and Reflection of Public Labels

First convert all the programs of the Motion module, and then reflect 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	Appendix 2.1			
Input signal of the PLC CPU	Appendix 2.2			
Input signal of the remote input module	Appendix 2.3			

APPENDIX 2.1 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.



(1) 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.

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

(b) Servo parameters

Make sure that the servo parameters are set as follows:

No.	Item	Setting value
Pr. PD41.2	Limit switch enabled status selection	1: Only enabled in home position return mode
Pr. PD41.3	Sensor input method selection	0: Input from servo amplifier

I/O				Selected <u>I</u> tems Write	Axi <u>s</u> Writing	
No.	Abbr.	Name	Unit	Setting range	Station 1	Station2
Digital I/O						
Device set	ting				Setting	Setting
PD03.0-1	*	Device selection DI1		00-7F	0A	0A
PD04.0-1	*	Device selection DI2		00-7F	0B	0B
PD05.0-1	*	Device selection DI3		00-7F	22	22
PD51.0-1	*	Device selection DI3-2		00-7F	62	62
PD38.0-1	*	Device selection DI4		00-7F	2C	2C
PD39.0-1	*	Device selection DI5		00-7F	2D	2D
PD07.0-1	*	Device selection DO1		00-7F	05	05
PD08.0-1	*	Device selection DO2		00-7F	04	04
PD09.0-1	*	Device selection DO3		00-7F	03	03
Device ass	signment				Setting	Setting
PD01.0-7	*DIA1	Input signal automatic on selection 1		000000-00000FF0	00000000	00000000
Input filter	·					
PD11.0	*	Input signal filter selection		0-B	7:3.500ms 👻	7:3.500ms 👻
ALM outpu	ıt					
PD14.1	*	Selection of output device at warning occurrence		0-1	0 : WNG signal turn (🗸	0 : WNG signal turn (🚽
Analog out	tput					
Analog mo	nitor					
PC09.0-1		Analog monitor 1 output selection		00-1F	00 : Servo motor spe 👻	00 : Servo motor spe 👻
PC11	MO1	Analog monitor 1 offset		-999-999	0	0
PC10.0-1		Analog monitor 2 output selection		00-1F	01 : Torque or thrust 👻	01 : Torque or thrust 👻
PC12	MO2	Analog monitor 2 offset		-999-999	0	0
Stroke limit	t function	-				
Stroke limit	t function					
PC19.0	*	[AL. 099 Stroke limit warning] selection		0-1	1 : Disabled 🔹	1 : Disabled 🔹
PD41.2	*	Limit switch enabled status selection		0-1	1 : Only enabled ir 🔻	1 : Only enabled ir 🔻
PD41.3	*	Sensor input method selection		0-1	0 : Input from servo 👻	0 : Input from servo 👻

(c) Changing the PDO mapping

1) On the "CC-Link IE TSN Configuration" window, double-click "<Detail Setting>" in the "PDO Mapping Setting" to display the "PDO Mapping Setting" screen.

123	CC-	-Link	IE TSN	I Configuration	n (Start I/	O: 0000)							— 🗆 X
i c	C-Li	ink IE	TSN C	Configuration	Edit	View Close	with Di	scarding the Se	tting Clo	se with Reflecting the Sett	ing		
		Conn	ected	l/Disconnecte	d Module	Detection		Detailed Disp	lav				Module List ×
	Mada Satting: Online					Online			Assianm	ent Method:	\sim		CC-Link IE TSN Selection Find Modu 4 >
	0	Cyclic Transmission Time (Min.):			us		Commun	ication Period Interval (M	in.): - us				
	Ιſ					RWr S	Setting	RWw Setting	Param	eter Automatic Setting		maddana	General CC-Link IE TSN Module
			No.	Model	Name	Po	ints	Points			PDO Mapping Setting	IP Address	CC-Link IE TSN Module (Mitsubish
•			0	Host Station								192.168.3.253	Master/Local Module
	-	<u>.</u>	1	MR-J5-G			24	20		<detail setting=""></detail>	<detail setting=""></detail>	192.168.3.1	Motion Module
	h	ilo	2	MR-J5-G			24	20		<detail setting=""></detail>	<detail setting=""></detail>	192.168.3.2	GOT2000 Series
													DC Input
													Transistor Output
													Analog Input
													General nurnose Inverter
	•	<										>	General-Purpose AC Servo
				STA#1	STA#	2							■ I/O Combined
						-							
	2				-								
[_]	9			l∎									
	T 4 4		artar	L.	H								
	Stat	ion	aster										
	otal	l STA Star	#:2	-5	-10								
١	iiic)	Juan		MR-J5-G	MR-J5	-G							
				<								>	

2) Set the end of the PDO mapping parameter as follows, and click the [OK] button.

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

The entry name is displayed as "Digital inputs".

(The double word type data is displayed in two lines.)

C RFDO	r bo mapping re	i dille tel		1		
	Link Device	Index [Hexadecimal]	Sub-Index [Hexadecimal]	Entry Name	Comment	Data Type
	-	606c	00	Velocity actual value		INTEGER32
	-	60f4	00	Following error actual value		INTEGER32
	-	60f4	00	Following error actual value		INTEGER32
	-	6041	00	Statusword		UNSIGNED 16
	-	0000	00	GAP	2byte GAP	-
	-	6077	00	Torque actual value		INTEGER 16
	-	2d11	00	Status DO 1		UNSIGNED 16
	-	2d12	00	Status DO 2		UNSIGNED 16
	-	2d13	00	Status DO 3		UNSIGNED 16
	-	2d14	00	Status DO 4		UNSIGNED 16
	-	2d15	00	Status DO 5		UNSIGNED 16
	-	2a41	00	Current alarm		UNSIGNED 32
	-	2a41	00	Current alarm		UNSIGNED32
	-	2d21	00	Sync cycle counter		UNSIGNED32
	-	2d21	00	Sync cycle counter		UNSIGNED32
	-	2d22	00	Sync ABS counter		INTEGER 16
	-	60fd 🗸 🗸	00	Digital inputs		UNSIGNED32
	-	60fd	00	Digital inputs		UNSIGNED32
	-					

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

[POINTS]

"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 Section 17.1 in "MR-J5-G/MR-J5W-G User's Manual (Object Dictionary)".

 After the setting is completed, click "Close with Reflecting the Setting" on the "CC-Link IE TSN Configuration" window to reflect the data.

83	CC-L	ink IE TSN	V Configuration	n (Start l	/0: 0000)							— 🗆 X
÷ o	C-Lin	k IE TSN (Configuration	Edit	View Cl	ose with D	scarding the Se	tting Clo	ose with Reflecting the Setti	ng		
	C	onnected	l/Disconnecte	d Modul	e Detecti	on	Detailed Disp	lay				Module List ×
	Мо	de Settin	ng:		Online			Assignm	ent Method:	\sim		CC-Link IE TSN Selection Find Modu 4 +
	Cyc	lic Transr	mission Time (Min.):		- us		Commun	nication Period Interval (Mi	n.): - us		記 乳 階 📴 🏡 🖻 🗙
		No.	Model	Name	RV	Vr Setting	RWw Setting	Param	neter Automatic Setting	PDO Mapping Setting	IP Address	General CC-Link IE TSN Module
	-				_	Points	Points					CC-Link IE TSN Module (Mitsubish
V		0	Host Station						P. 1. 10. 11	0.1.10.10	192.168.3.253	Master/Local Module
	8	• 1 •	MR-J5-G			24	20		<detail setting=""></detail>	<detail setting=""></detail>	192.168.3.1	Motion Module
	E	o 2	MR-JD-G	_		24	20		<detail setting=""></detail>	<detail setting=""></detail>	192.108.3.2	GOT2000 Series
												DC Input
												Transistor Output
												Analog Input
												Analog Output
	<										>	General purpose Inverter
-	1.5		_									General-Purpose AC Servo
			STA#1	STA	#2							I/O Combined
自月 S T Li	TA#0 Statio otal S ne/Si) Master n STA#:2 tar										
			<	-in-Ju							>	

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



- (d) Changing parameters of the Motion module
 - 1) Double-click "Network I/O" in the navigation window of the motion control setting function to display the device controlled by the Motion module.

Navigation		×
🔂 0000 RD78G4		
🐢 Basic Setting		
🕵 System Setting		
🗉 😽 Axis		
🗉 🎆 Axes Group		
🕼 I/O Data		
🗉 🥵 Calculation Profile		
🗏 Network I/O		
🖽 🔚 Program		
📷 FB/FUN		
🗉 🎼 Label		

2) Check the "Labeling Target" checkbox of [MR_J5_G_***_DigitalInputs] (the label of the servo amplifier with the hardware stroke limit connected), and click the [Create Label] button.



3) Click the [Yes] button on the confirmation screen.

The generated network I/O label will be stored below "NW+Global1" in "Global Label".



4) Open the axis parameter of the real drive axis.

From the navigation window, select "Axis", and double-click the name of the axis to be used (initial name: Axis0001).

From the "Axis Parameter Setting" window, select "Real Drive Axis" \rightarrow "Upper Limit Signal", and click the [...] icon in "Target".



5) On the "Target Setting" screen, enter "MR_J5_G_***_DigitalInputs.1" in "Source".

Item		Setting		
Source Type	Global Label			
Source Data Type				
Saurce	MR_J5_G_001_DigitalInputs.1			

On the "Target Setting" screen of the lower stroke limit, enter "MR_J5_G_***_DigitalInputs.0" in "Source".

Target Setting	×
Item	Setting
Source Type	Global Label
Source Data Type	
Saurce	MR_J5_G_001_DigitalInputs.0
	<u>Q</u> K <u>Q</u> ancel

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

Setting Item		
Select <u>F</u> older Display All I	Data 🗸	
Item	Axis0001	Axis0002
📮 Axis Information		
Axis No.	1	2
📮 Axis Parameter Constant	Expands setting values at axis variab	le initialization.Re-importing to the
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		
Target	[VAR]MR_J5_G_001_DigitalInputs.1	[VAR]MR_J5_G_002_DigitalInputs.1
Signal Detection Method	1:Detection at FALSE	1:Detection at FALSE
Compensation Time	0.0 s	0.0 s
Filter Time	0.0 s	0.0 s
😑 Lower Limit Signal		
Target	[VAR]MR_J5_G_001_DigitalInputs.0	[VAR]MR_J5_G_002_DigitalInputs.0
Signal Detection Method	1:Detection at FALSE	1:Detection at FALSE
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 Cyc
Absolute Position Reference S	e 3:Feed Machine Position	3:Feed Machine Position

The setting is completed. Convert all the program.

(e) 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, and then move the axis to the direction within the range with JOG operation, etc.

(2) 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.

(a) Servo parameters

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

No.	Item
Pr. PT29.0	Device input polarity 1
Pr. PT45	Home position return method

Set the necessary parameters in the following table according to the homing method. (The parameters that require setting vary with the homing method.)

No.	Item
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

(b) Program

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



(c) Operation check

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

APPENDIX 2.2 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.



(1) 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 FLS, and X21 as RLS.

(a) The location of the sensors

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

(b) Servo parameters

Set the following servo parameters as shown below:

No.	Item	Setting value
Pr. PD41.2	Limit switch enabled status selection	1: Only enabled in home position return mode
Pr. PD41.3	Sensor input method selection	1: Input from controller

I/O				Selected Items Write	Axis Writing	
No.	Abbr.	Name	Unit	Setting range	Station1	_
Digital I/O						
Device set	ting				Setting	
PD03.0-1	*	Device selection DI1		00-7F		0A
PD04.0-1	*	Device selection DI2		00-7F		0B
PD05.0-1	*	Device selection DI3		00-7F		22
PD51.0-1	*	Device selection DI3-2		00-7F		62
PD38.0-1	*	Device selection DI4		00-7F		2C
PD39.0-1	*	Device selection DI5		00-7F		2D
PD07.0-1	*	Device selection DO1		00-7F		05
PD08.0-1	*	Device selection DO2		00-7F		04
PD09.0-1	*	Device selection DO3		00-7F		03
Device ass	ignment				Setting	
PD01.0-7	*DIA1	Input signal automatic on selection 1		000000-00000FF0	000000	000
Input filter						
PD11.0	*	Input signal filter selection		0-B	7:3.500ms	•
ALM outpu	it					
PD 14.1	*	Selection of output device at warning occurrence		0-1	0 : WNG signal turn ON	Ŧ
Analog ou	tput					
Analog mo	nitor					
PC09.0-1		Analog monitor 1 output selection		00-1F	00 : Servo motor speed (-
PC11	MO1	Analog monitor 1 offset		-999-999		0
PC10.0-1		Analog monitor 2 output selection		00-1F	01 : Torque or thrust (±8	•
PC12	MO2	Analog monitor 2 offset		-999-999		0
Stroke limi	t function					
Stroke limi	t function					
PC19.0	*	[AL. 099 Stroke limit warning] selection		0-1	0 : Enabled	-
PD41.2	*	Limit switch enabled status selection		0-1	1: Only enabled in ho	•
PD41.3	*	Sensor input method selection		0-1	1: Input from control	-

(c) Program of the PLC CPU

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.



[POINTS]

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

(d) Parameters of the Motion module

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

tting Item List		Setting Item										
put the Setting Item to Search	<i>i</i> h	Select <u>F</u> older	Display All Data 🔍									
		Item	Axis0001									
		Axis No.	1									
⊡~¶∎ Real Drive Axis ⊡∽ Axis Information		Axis Parameter Cor Station Address	nstan: Expands setting values at <i>Setti</i> r 19216831	taxis va								
Axis No.		Axis Type Setting	A:Real Drive Axis									
Axis Parameter Constant Station Address Setting Axis Type Setting		Upper Limit Signal										
Upper Limit Signal		Signal Detection	Met 1:Detection at EALSE									
Lower Limit Signal		Compensation T										
Ontrol Cycle Setting	noo Cotti	Filter Time	0.0 8									
Absolute Position Manag	ement Sei	□ Lower Limit Signal	0.0 8	_								
Ring Counter Enabled Se	lection	Signal										
Ring Counter Lower Limi	imit Value imit Value led	mit Value mit Value ed	mit Value mit Value ed	mit Value mit Value ed	mit Value mit Value ed	mit Value mit Value ed	mit Value mit Value ed	mit Value mit Value ed	it Value it Value I	Target Signal Detection	[DEV](BOOL)G11478000.1 Met 1:Detection at FALSE	
Torque Limit Maximum V	'alue	Compensation T	ime 0.0 s									
Negative Direction Torqu	e Limit In	Filter Time	0.0 s									
Positive Direction Torque	e Limit Ini	Control Cycle Sett	ing 0:Operate in the First Ope	ration C								
High-speed Mode Setting	5	Absolute Position	Refer 3:Feed Machine Position									
Axıs Parameter		Absolute Position	Mana – 1: Automatic Setting (Acc	uire fro								
How Windows Lincoder Axis		Ring Counter Enab	oled S 0:Disabled									
Wintual Drive Axis		Ring Counter Lowe	er Lin -10000000000.0									
Virtual Encouer Axis		Ring Counter Uppe	er Lin 10000000000.0									
		Slave Emulation E	nable 0:Disabled									
Ta	rget Setting			×								
			Catting									

Saurce Type	Device
Source Data Type	BOOL
Source	G11470000.0

(Note) When the public label is used, set as follows:

Target Setting	×
Item	Setting
Source Type	Global Label
Source Data Type	
Source	G_bFLS
	<u>Q</u> K <u>Q</u> ancel

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.

(e) 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, and then move the axis to the direction within the range with JOG operation, etc.

(2) Homing with the DOG signal

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

(a) Servo parameters

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

No.	ltem
Pr. PT29.0	Device input polarity 1
Pr. PT45	Home position return method

Set the necessary parameters in the following table according to the homing method. (The parameters that require setting vary with the homing method.)

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

(b) Programming with the Motion module

1) Program of the PLC CPU

The ON/OFF status of X22 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.



2) Parameters of the Motion module

Right-click "I/O Data" in the navigation window of the motion control setting function, and select [Add New Data].

Navigation	ų ×	
₽ੑੑੑੑੑੑੑੑੑੑੑੑੑੑੑੑੑੑੑ੶ ₽⊏ 🌣		
🔂 0000:RD78G4		
🧬 Basic Setting		
System Setting		
🎞 👹 Axis		
🖽 🊵 Axes Group		
1/O Data		
II 🦾 Calculation	Add New Data	Ins
- 😾 Network I/(New Folder Ctrl+S	Shift+N
🗉 🔚 Program	Expand/Collapse Tre	ee 🕨
📑 FB/FUN		
🗉 🛗 Label		

Set as follows:

New Data		×	
Basic Setting			
Data Type	🚺 I/O Data	-	
(Data Name)	DOG	•	 Set any data name.
Detailed Catting			
Detailed Setting			
External Signal Setting	1		Select "Device" for the buffer
Data lype	Input Signal	•	
Source Type	Device		memory, and "Global Label" f
Source Data Type			the public label.
Source	G11478000.2	→ ¬	· · ·
Signal Detection Method	HighLevel	-	
Compensation Time[s]	0.0		
Filter Time[s]	0.0		Set the buffer memory number
			for the buffer memory, and th
			public label name for the pub
			label.
	OK	Cancel	

3) Program of the Motion module

Enter the I/O data name set in 2) in the AbsSwitch input of MC_Home.

//Homing	
MC_Home_1(
Axis:= Axis0001.AxisRef,	
Execute:= bHoming ,	
Position:= 0.0 ,	
AbsSwitch:= DOG , 🗨 🛶	Enter the I/O data name.
Options:= 0 ,	
Done=> bHomeDone .	
Error=> bHomeError	
);	

(c) Programming with the PLC CPU

- 1) It is necessary to define an MC_INPUT_REF type structure to the PLC CPU.
 - From the [Library] tab in the element selection window on MELSOFT GX Works3, click "Library" \rightarrow "MotionControl_RD78_**** \rightarrow "Structured Data Type".

Select "MC_INPUT_REF", and drag and drop it onto "Structured Data Types" under "Label" in the navigation window.



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



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



4) Set the member of the MC_INPUT_REF type structure with the program.



5) Write MC_Home to the program.



(d) Operation check

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

APPENDIX 2.3 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.



(1) 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 FLS, and RX1 as RLS.

(a) The location of the sensors

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

(b) Servo parameters

Set the following servo parameters as shown below:

No.	Item	Setting value
Pr. PD41.2	Limit switch enabled status selection	1: Only enabled in home position return mode
Pr. PD41.3	Sensor input method selection	1: Input from controller

I/O				Selected Items Write	e Axis Writing	
No.	Abbr.	Name Unit S		Setting range	Station1	
Digital I/O						
Device set	ting				Setting	
PD03.0-1	*	Device selection DI1		00-7F		0A
PD04.0-1	*	Device selection DI2		00-7F		0B
PD05.0-1	*	Device selection DI3		00-7F		22
PD51.0-1	*	Device selection DI3-2		00-7F		62
PD38.0-1	*	Device selection DI4		00-7F		2C
PD39.0-1	*	Device selection DI5		00-7F		2D
PD07.0-1	*	Device selection DO1		00-7F		05
PD08.0-1	*	Device selection DO2		00-7F		04
PD09.0-1	*	Device selection DO3		00-7F		03
Device ass	signment				Setting	
PD01.0-7	PD01.0-7 *DIA1 Input signal automatic on selection 1 D000000-00000FF0					00
Input filter						
PD11.0	*	Input signal filter selection		0-B	7:3.500ms	•
ALM outpu	ıt					
PD14.1	*	Selection of output device at warning occurrence		0-1	0 : WNG signal turn ON	•
Analog ou	tput					
Analog mo	nitor					
PC09.0-1		Analog monitor 1 output selection		00-1F	00 : Servo motor speed (•
PC11	MO1	Analog monitor 1 offset		-999-999		0
PC10.0-1		Analog monitor 2 output selection		00-1F	01 : Torque or thrust (±8	•
PC12	MO2	Analog monitor 2 offset		-999-999		0
Stroke limi	t function					
Stroke limit	t function					
PC19.0	*	[AL. 099 Stroke limit warning] selection		0-1	0 : Enabled	-
PD41.2	*	Limit switch enabled status selection		0-1	1: Only enabled in ho	•
PD41.3	*	Sensor input method selection		0-1	1 : Input from control	•

(c) Parameters of the Motion module

1) Setting of the network I/O

Double-click "Network I/O" in the navigation window of the motion control setting function. On this screen, label the input signal of the remote input module. Check the checkbox of RX0 and RX1, and click the [Create Label] button.

Navigation 4 ×	N 28	etwor	k I/O ×							4	₽ ▼
P E - P⊂ ✿		No.	IP Address	Model Name	Device Label	Data Type	Labeling Target	Data Type	Label Name/Definition Name of***	Comment	^
1 0000:RD78G4	+	1	192.168.3.1	MR-J5-G	MR_J5_G_001	Entire Device					
Basic Setting	-	2	192.168.3.2	NZ2GN2S1-32DT	NZ2GN2S1_32DT_001	Entire Device		-			
System Setting						RX0	\checkmark	Bit	NZ2GN2S1_32DT_001_RX0	External input signal X0	
🖬 🐖 Axis						RX1		Bit	NZ2GN2S1_32DT_001_RX1	External input signal X1	
🖬 🌺 Axes Group						RX2		Bit	NZ2GN2S1_32DT_001_RX2	External input signal X2	
						RX3		Bit	NZ2GN2S1_32DT_001_RX3	External input signal X3	
Calculation Profile							_		NZ2GN2S1_32DT_001_RX4	External input signal X4	
Network 1/O				Ch		ad DV1	oftho		NZ2GN2S1_32DT_001_RX5	External input signal X5	
Network I/O					ieck RAU al		orthe		NZ2GN2S1_32DT_001_RX6	External input signal X6	
🖿 🔚 Program					nnoctod ror	noto inn	ut modul		NZ2GN2S1_32DT_001_RX7	External input signal X7	
B/FUN					inected lei	note inp	ut mouu	c .	NZ2GN2S1_32DT_001_RX8	External input signal X8	
🗏 🔚 Label									NZ2GN2S1_32DT_001_RX9	External input signal X9	
🗏 🔚 Global Label						RXA		Bit	NZ2GN2S1_32DT_001_RXA	External input signal XA	
🚰 Ax+Global						RXB		Bit	NZ2GN2S1_32DT_001_RXB	External input signal XB	
🚹 Prf+Global						RXC		Bit	NZ2GN2S1_32DT_001_RXC	External input signal XC	
👫 En+Global						RXD		Bit	NZ2GN2S1_32DT_001_RXD	External input signal XD	~
👫 Gr+Global	Expla	nation	i -								
👫 Global	Regi	ster th	he I/O data for the	cyclic communication b	etween the motion mod	ule and the slave	device under mo	tion module ma	nagement as a label.		~
📬 Sys+Global	Even	uting	'Overte Label' regi	tere only 'I sheling Targ	at' data to the global la	hal list (NW#Glab					
👫 NW+Global1	Unat	ble to r	restore the label regis	gistration data before ci	reation after executing	Create Label.	Jai).				
🕱 🚝 Structured Data Type	Edito		tanto in this windo	w we not exceed to the e	united and are only los	t while the project	at in ones				
	After	the p	project is re-opene	d, the label registration of	data in the global label	list (NW+Global)	will be reflected t	o the displayed	data.		
	-										× 1
									Update Network Configuration Inf	Create Label	

After the label is generated, the "NW+Global1" group will be added in "Global Label" in the navigation window.

2) Axis parameters

On the "Axis Parameter Setting" window, set "Target" of "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.

(d) 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, and then move the axis to the direction within the range with JOG operation, etc.

(2) Homing with the DOG signal

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

(a) Servo parameters

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

No.	Item
Pr. PT29.0	Device input polarity 1
Pr. PT45	Home position return method

Set the necessary parameters in the following table according to the homing method. (The parameters that require setting vary with the homing method.)

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

(b) Parameters of the Motion module (setting of the network I/O)

Double-click "Network I/O" in the navigation window of the motion control setting function. On this screen, label the input signal of the remote input module. Check the checkbox of RX2, and click the [Create Label] button.



After the label is generated, the "NW+Global1" group will be added in "Global Label" in the navigation window.

(c) Programming with the Motion module

1) Right-click "I/O Data" in the navigation window of the motion control setting function, and select [Add New Data].



Set as follows:

New Data		×	
Basic Setting			
Data Type	👫 I/O Data	-	
(Data Name)	DOG 🔶		Set any data name.
Detailed Setting			
External Signal Setting			
Data Type	Input Signal	-	
Source Type	Global Label		Select "Global Label".
Source Data Type			
Source	NZ2GN2S1_32D_001_RX2		7
Signal Detection Method	HighLevel	-	
Compensation Time[s]	0.0		
Filter Time[s]	0.0		
			Set the global label of RX2.
	OK Canc	el	

2) Enter the I/O data name set in (2) in the AbsSwitch input of MC_Home.

//Homing	
MC_Home_1(
Axis:= AxisOOO1.AxisRef,	
Execute:= bHoming ,	
Position:= 0.0 ,	[]
AbsSwitch:= DOG , 🛛 🛶 🛶	Enter the I/O data name.
Options:= 0 ,	
Done=> bHomeDone ,	
Error=> bHomeError	
);	

(d) Programming with the PLC CPU

- 1) It is necessary to define an MC_INPUT_REF type structure to the PLC CPU.
 - From the [Library] tab in the element selection window on MELSOFT GX Works3, click "Library" \rightarrow "MotionControl_RD78_****" \rightarrow "Structured Data Type".

Select "MC_INPUT_REF", and drag and drop it onto "Structured Data Types" under "Label" in the navigation window.



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



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



4) Set the member of the MC_INPUT_REF type structure with the program.



5) Write MC_Home to the program.



(e) Operation check

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

APPENDIX 3 Precautions When the Absolute Position Detection is Used

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

(1) Setting of the servo parameters

Change the following two servo parameters:

No.	Item	Setting value
Pr. PA03.0	Absolute position detection system selection	0: Disabled \rightarrow 1: Enabled
Pr. PC29.5	[AL. 0E3 Absolute position counter warning] selection	1: Enabled \rightarrow 0: Disabled

(2) At the first power-on

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

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

After that, execute homing.

APPENDIX 4 Program Example in Ladder

The following shows a [ServoON_Jog] program example created in ladder instead of FBD in Chapter 4.

Wr	ite -	1	2	3	4	5	6	7	8	9	10	11	12
1	*シーケンサレディロ	N / PLC Ready O	N										
		SM400	X1										YO
													\square
	(0)		11										\cup
1 ²	(0)	Always ON	Synchronization										PLC READY
			Flag										
3	*全軸サーボオン /	All axes servo Ol	N						-				
	0						MOv_AllPower_1	(MCv_AllPower)					
1	(4,						All Axes Opera	ation Available					
						RD78_0000.Axis0001.A							
							DUT: AXIS	Axis 1001					
5													
		SM400											
		0.0.400											
		\square					B: Enable	Busy B					1
6		Always ON											
		XO	G ISSNOEE										
							D: Same ON	E					
							B. Servoold	Error 6					
Ľ		READY	Servo OFF										
								EccortD 1 NV					
								21012-011					
l°.													
9	*JOG 連転用デー	久設定 / Data set	ting for JOG operati	ion	1					1	1	1	
		001 Md Driver										E25000	G leJozVelocity
		\vdash	-										
10	(27)										EDMOVP		JOG Velocity
												E50000	leJogAcceleratio
							-						n
11											EDMOVP		acceleration
												E50000	leJogDeceleratio
											501000		.10G
12											EDMOVP		deceleration
												EO	leJogJerk
1											EDMOVE		JOG jerk
13											CONOT		
		1											

14 *J	OG Axis0001									 			
		G_bJogF1	G_bJozR1	G_bHoming1Req	G_bPosReq	G_bContPosReq	G_bInterpolation Ren	G_bSyncReq			bJogF1		
		\vdash								 	-0-		
15	(54)	JOG Positive rotation command Axis0001	JOG Reverse rotation command Axis0001	Homing start request Axis0001	Single axis positioning start request	Single axis continuous positioning start request	2-axis linear interpolation control start request	Synchronous control start request			JOG Positive rotation command Axis0001		
		G_bJogF1	G_bJogR1	G_bHomins1Req	G_bPosReq	G_bContPosReq	G_bInterpolation Reg	G_bSyncReq			bJogR1		
16	(63)	JOG Positive rotation command Axis0001	JOG Reverse rotation command Axis0001	Homing start request Axis0001	Single axis positioning start request	Single axis continuous positioning start request	2-axis linear interpolation control start request	Synchronous control start request			JOG Reverse rotation command Axis0001		
17	(71)						MCv_Jog_1 J	(MCv_Jog) OG					
						RD78_0000 Axis0001 A xisRef	DUT: Axis	Axis DUT					
10													
		bJosF1					- B: JogForward	Done:B					
19		JOG Positive rotation command Axis0101											
		bJogR1					B: JogBackward	Busy B			G_bJog1Busy		
20		JOG Reverse rotation command Axis0001									JOG operation in progress Axis0001		
						G_leJogVelocity	L: Velocity	Active B		 			
21						JOG Velocity							
		-				leJogAcceleration]	L: Acceleration	CommandAbor ted:8	 				
22						JOG acceleration							
						leJosDeceleration	L: Deceleration	Error B					
23						JOG deceleration							
						leJogJerk	- L: Jerk	ErrorID :UW		 			
24						JOG jerk							
						Г но Т	LD: Ontione						
25							- op. op. op. op.						
L	26 *J	DG Axis0002											
-----	-------	-------------	--------------------------	---------------------	------------------	------------------------------	--------------------------------	-----------------	-----------------------------------	---	--	------	------------------------------
I			G_bJogF	2 G_bJogR2	G_bHomins2Req	G_bContPosReq	G_bSyncReq						bJogF2
1			\vdash									 	-0
1	27	(110	100 Positive	IOG Reverse	Homine start	Sinela avin	Synchronour control						100 Positive
1			rotation	rotation	request Axis0002	continuous	start request						rotation
I			Axis0002	Axis0002		request							Axis0002
I			G_bJosF	2 G_bJosR2	G_bHomins2Req	G_bContPosReq	GjbSyncReg						bJogR2
			\vdash		/_							 	-
I	28	(117		100 P		Sharela anda	Suchastan and						IOC Deverse
1			rotation	rotation	request Axis0002	continuous	start request						rotation
1			command Axis0002	Command Axis0002		positioning start request							command Axis0002
I	20	(100						MCv_Jos_2	(MCv_Jog)	I			
1	29	(123						JC	G				
I													
1			L				HD/8_0000.Axis0002.A xisRef	DUT: Axis	Axis DUT			 	
1	30												
1													
1													
ľ			bJogF2										
1								B: JogForward	Done:B			 	
1	31												
1			JOG Positive rotation										
1			command Axis0002										
ľ			b.logR2										G hJog2Busy
1								B: JosBackward	Busy B				
1	32												\cup
1			JOG Reverse rotation										JOG operation in progress
1			command Axis0002										Axis0002
ł			44100002					L: Velocity	Active B				
1							G_leJogVelocity					 	
1	33												
1							JOG Velocity						
1													
ł								L: Acceleration	CommandAbor ted [:] B				
1							leJogAcceleration						
1	34						L J						
1	-						JOG acceleration						
1													
ľ													
1	35						leJogDeceleration	L: Deceleration	Error B				
1													
1							JOG deceleration						
1													
ł							_	= L: Jerk	Error D UW				
1							leJogJerk					 	
1	36												
1							JOG jerk						
1													
I	37						-						
			L				но	UD: Options					
								op. options					
1													
	38	(162											1010
1													
- 1			1										

MEMO

Mitsubishi Electric Servo System Motion Module Quick Start Guide

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-08190 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 Piskarevsky pr. 2, bld 2, lit "Sch", BC "Benua", office 720; 195027 St. Petersburg, Russia	Tel : +7-812-633-3497
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 Mahallesi Nutuk Sokak No:5, TR-34775 Umraniye / Istanbul, Turkey	Tel : +90-216-526-3990
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 : +65-6473-2308
Thailand	Mitsubishi Electric Factory Automation (Thailand) Co., Ltd. 101, True Digital Park Office, 5th Floor, Sukhumvit Road, Bangchak, Phra Khanong, Bangkok	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