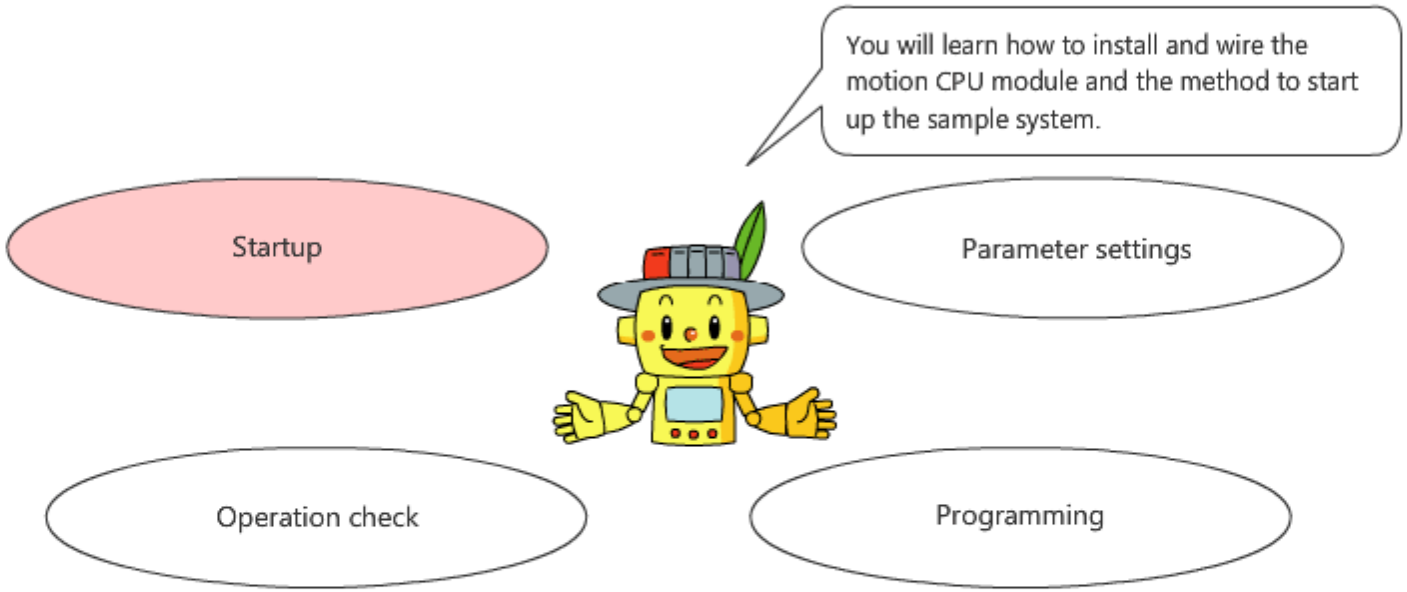
Servo System Controller

MELSEC iQ-R Series Motion Controller Basics (RnMTCPU)

This training course is for participants who will use a motion control system using the MELSEC iQ-R series motion CPU module for the first time.
Click the Next button at the upper right of the screen to proceed to the next page.

Introduction Purpose of the Course

This course is for participants who will configure a motion control system using a MELSEC iQ-R series motion CPU module for the first time, and for learning the system design, installation, wiring, configuration, and programming.



A basic knowledge of MELSEC iQ-R series programmable controller, AC servos, and positioning control are necessary to take this course.

Beginners are recommended to take the following courses:

- "MELSEC iQ-R Series Basics" course
- "GX Works3 (Ladder)" course
- "MELSERVO Basics (MR-J4)" course
- "FA Equipment for Beginners (Positioning)" course

Introduction Course Structure

The contents of this course are as follows.
We recommend that you start from Chapter 1.

Chapter 1 - Startup

Learn how to install and wire programmable controllers and servo amplifiers, wire external circuits, and other operations to start up the sample system in order.

Chapter 2 - Parameter Settings

Learn how to configure the motion CPU module system settings and various parameter settings.

Chapter 3 - Programming

Learn how to program the motion SFC programs using MT Developer2.

Chapter 4 - Operation Check

Learn how to perform operation checks using the sample programs.

Final Test

5 sections in total (14 questions) Passing grade: 60% or higher.

Introduction How to Use This e-Learning Tool

Go to the next page	>	Go to the next page.
Back to the previous page	<	Back to the previous page.
Move to the desired page	TOC	"Table of Contents" will be displayed, enabling you to navigate to the desired page.
Exit the learning	X	Exit the learning. Window such as "Contents" screen and the learning will be closed.

Introduction Cautions for Use

■Safety precautions

When you learn by using actual products, please fully read the safety precautions in the corresponding manuals.

■Precautions in this course

The displayed screens of the software version that you use may differ from those in this course.
The following shows the software used in this course and each software version.
For the latest version of each software, check the Mitsubishi Electric FA Website.

MELSOFT GX Works3	Ver.1.050C
MELSOFT MT Works2	Ver.1.146C

The icon indicates the reference manual. The contents of the manual described in this course are those of the following versions. If the versions differ, the section and content may be different.

Manual name	Manual No.	Version
MELSEC iQ-R Motion Controller User's Manual	IB-0300235	K
MELSEC iQ-R Motion Controller Programming Manual (Common)	IB-0300237	K
MELSEC iQ-R Motion Controller Programming Manual (Program Design)	IB-0300239	K
MELSEC iQ-R Motion Controller Programming Manual (Positioning Control)	IB-0300241	K

Introduction Cautions for Use

■Reference materials

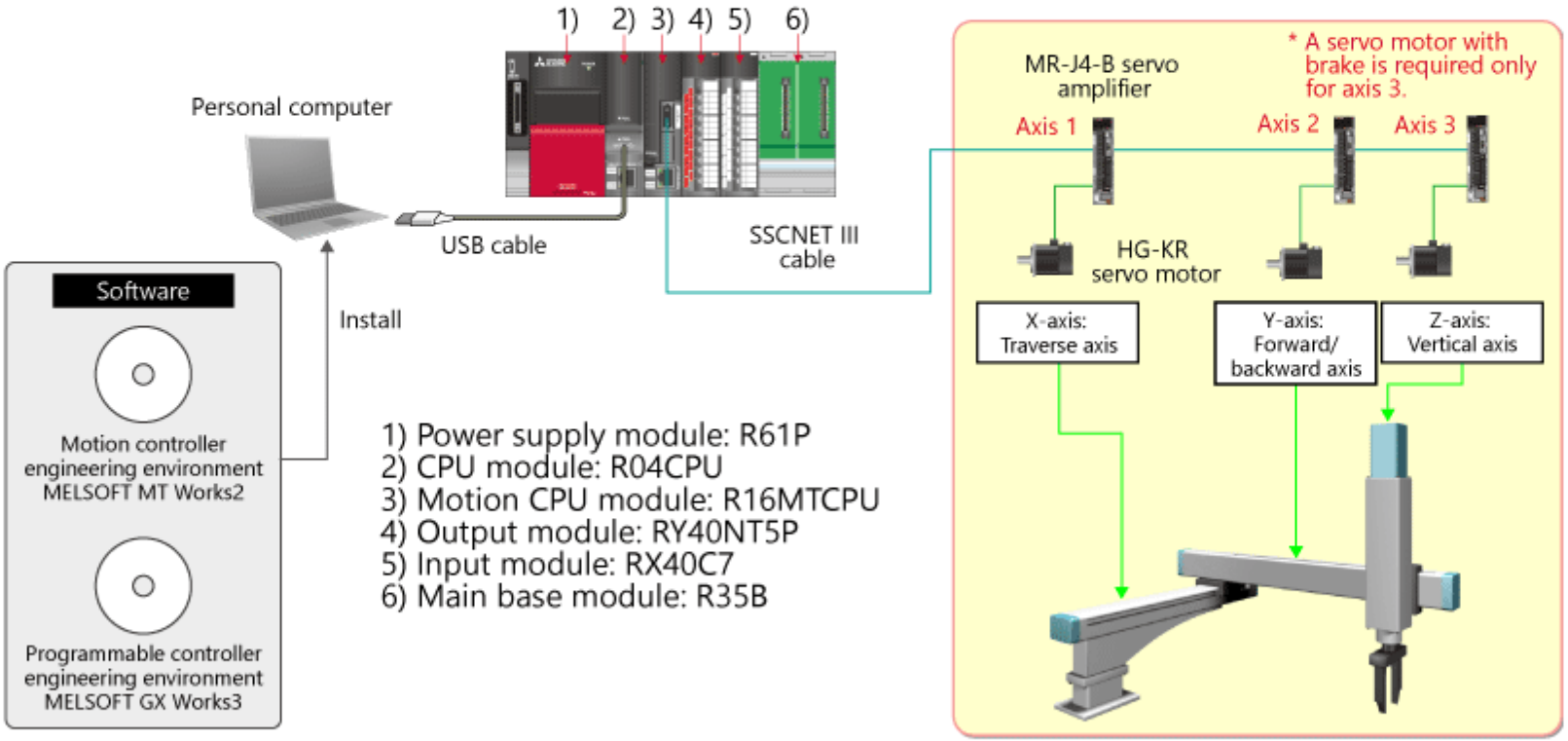
Below is a list of references related to the topics in this course. (Please note that these reference materials are not absolutely necessary as you can still complete this course without using them.)
Click the name of the reference file to download.

Name of reference	File format	File size
Recording paper	Compressed file	6.72 kB

Chapter 1 Startup

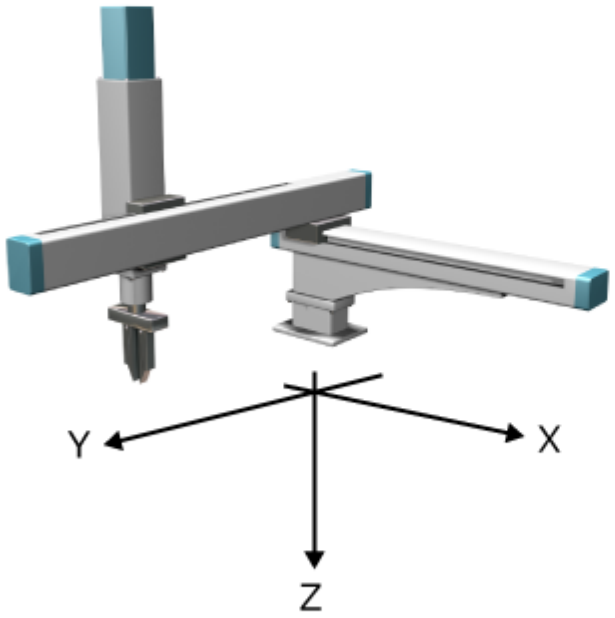
In this chapter, you will learn how to install and wire programmable controllers and servo amplifiers, wire external circuits, and perform the other work to start up the sample system in order.

1.1 System Configuration

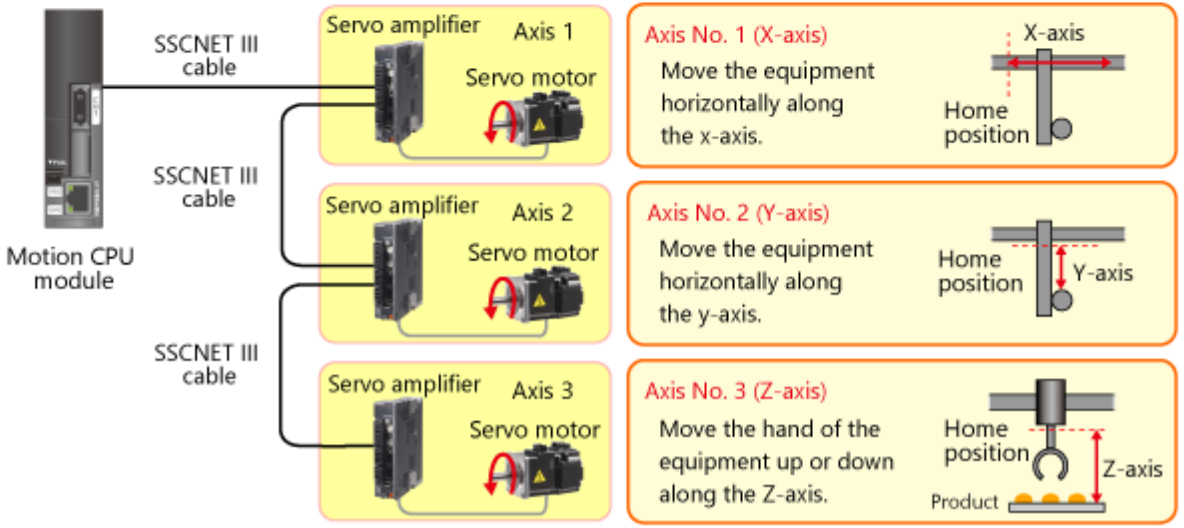


1.2 Sample System

The system to be operated in this course is a 3-axis X-Y-Z arm.
For the machine specifications, refer to the following table.



Axis		Mechanism	Reduction ratio	Operation range
Axis 1	X-axis: Traverse axis	Ball screw (Pitch: 10 mm)	1:2	-100.0 mm to 500.0 mm
Axis 2	Y-axis: Forward/backward axis	Ball screw (Pitch: 10 mm)	1:2	-100.0 mm to 500.0 mm
Axis 3	Z-axis: Vertical axis	Ball screw (Pitch: 10 mm)	1:2	-10.0 mm to 300.0 mm



1.2 Sample System

<Servo motor rotation direction>
From the machine specifications, consider the rotation direction of the servo motor when moving the machine in the forward rotation direction.
The rotation direction is either counterclockwise (CCW) or clockwise (CW) when seen from the load side (machine mounting side).
In the sample system, each axis is rotated counterclockwise (CCW) by the forward rotation command.

Counterclockwise (CCW)

Clockwise (CW)

<Consideration of home position return method>
Perform the home position return for each axis to eliminate the stop position errors.
There are multiple methods for returning to the home position. Select the method according to the machine specifications of the system.
In the sample system, the home position return is performed with the proximity dog method for each axis.

1.3 Wiring

This section explains the necessary wirings for the system.

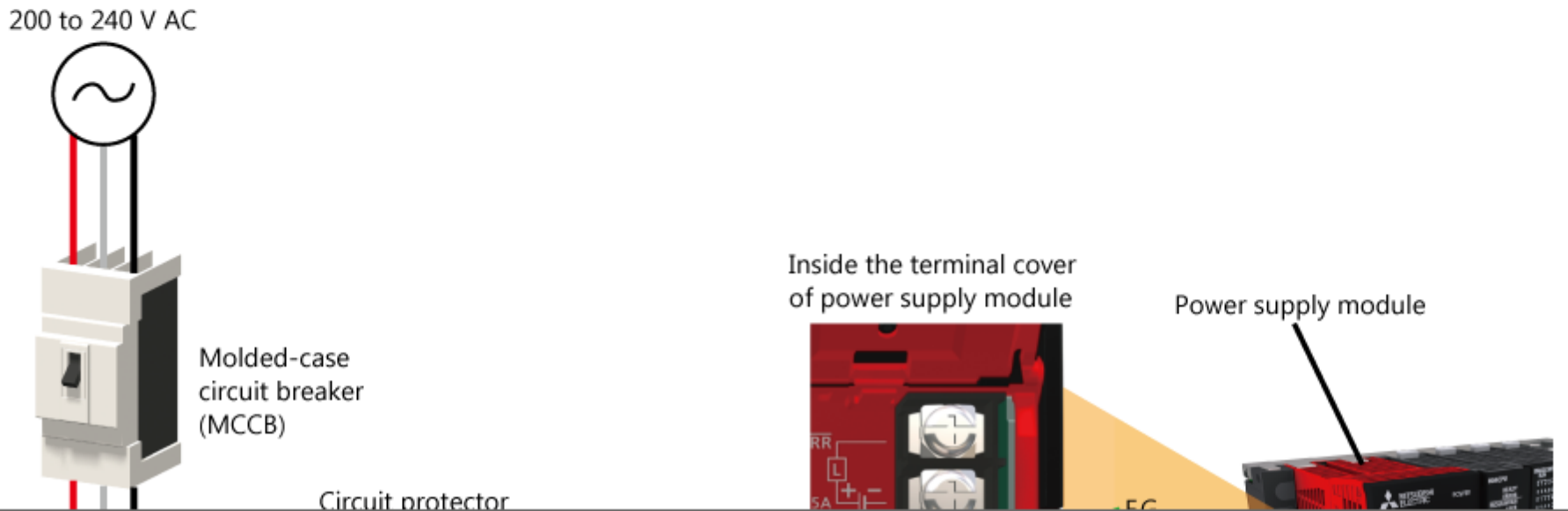
1.3.1 Wiring the Programmable Controller

(1) Wiring the power supply module

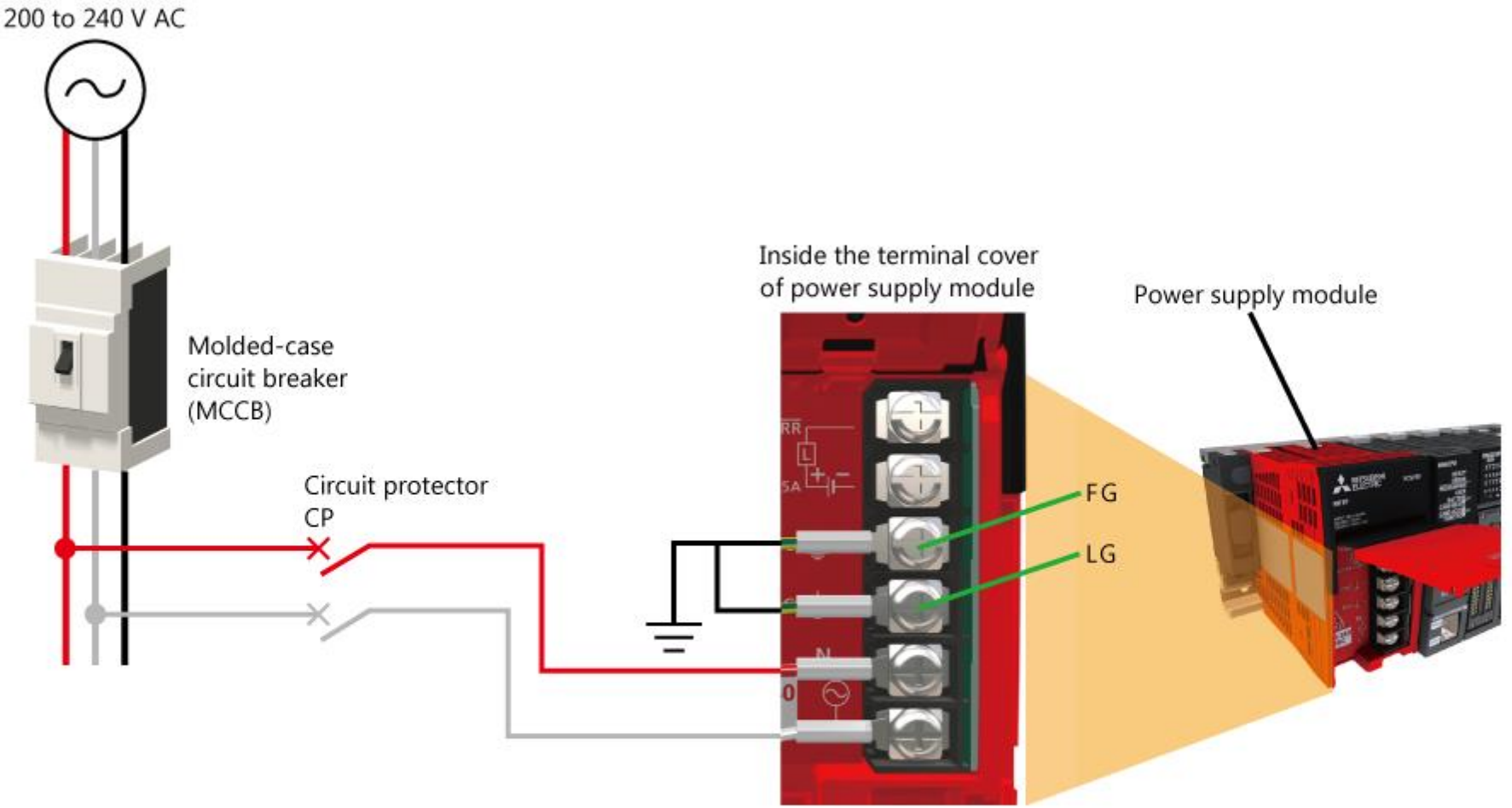
Connect the power supply wires to the power supply module of the programmable controller.

The following explains wiring of the power supply module.

- When performing the wiring, open the terminal cover on the front of the power supply module.
- Connect the AC power supply to be input to the power supply input terminals (L and N).
- Always ground the FG and LG terminals with D-class grounding (ground resistance of 100 Ω or less).



1.3.1 Wiring the Programmable Controller

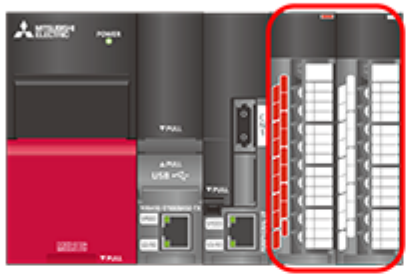


Applicable cable size: 18 to 14AWG

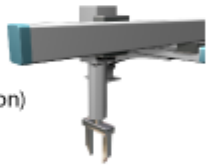
1.3.1 Wiring the Programmable Controller

(2) Wiring the I/O circuit

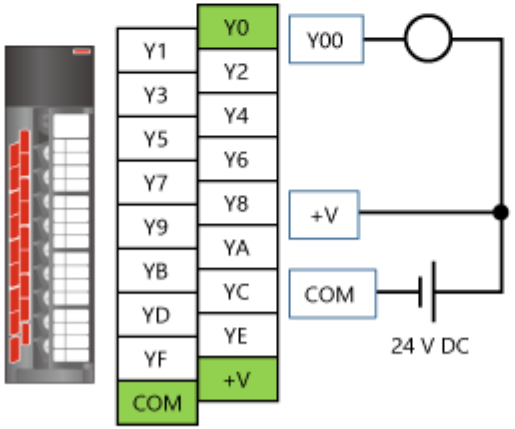
Connect the output module (RY40NT5P) and input module (RX40C7) to the external circuit.
The following figure shows the connection example of sink wiring.



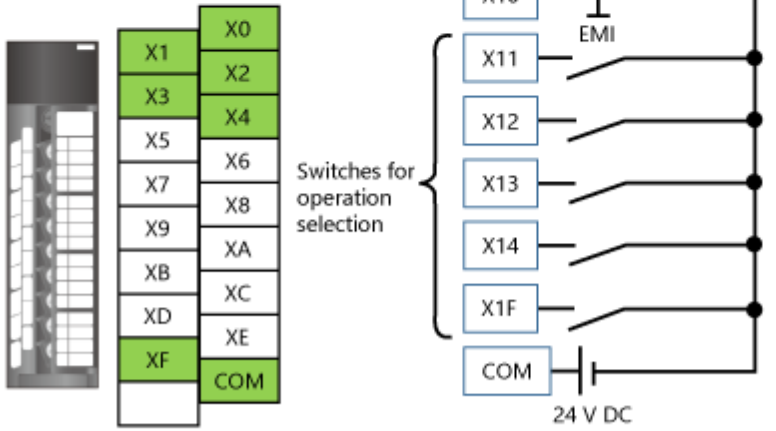
Output module (start XY: 0000)
RY40NT5P



Hand opening/closing part
(Hand closes when Y00 turns on)



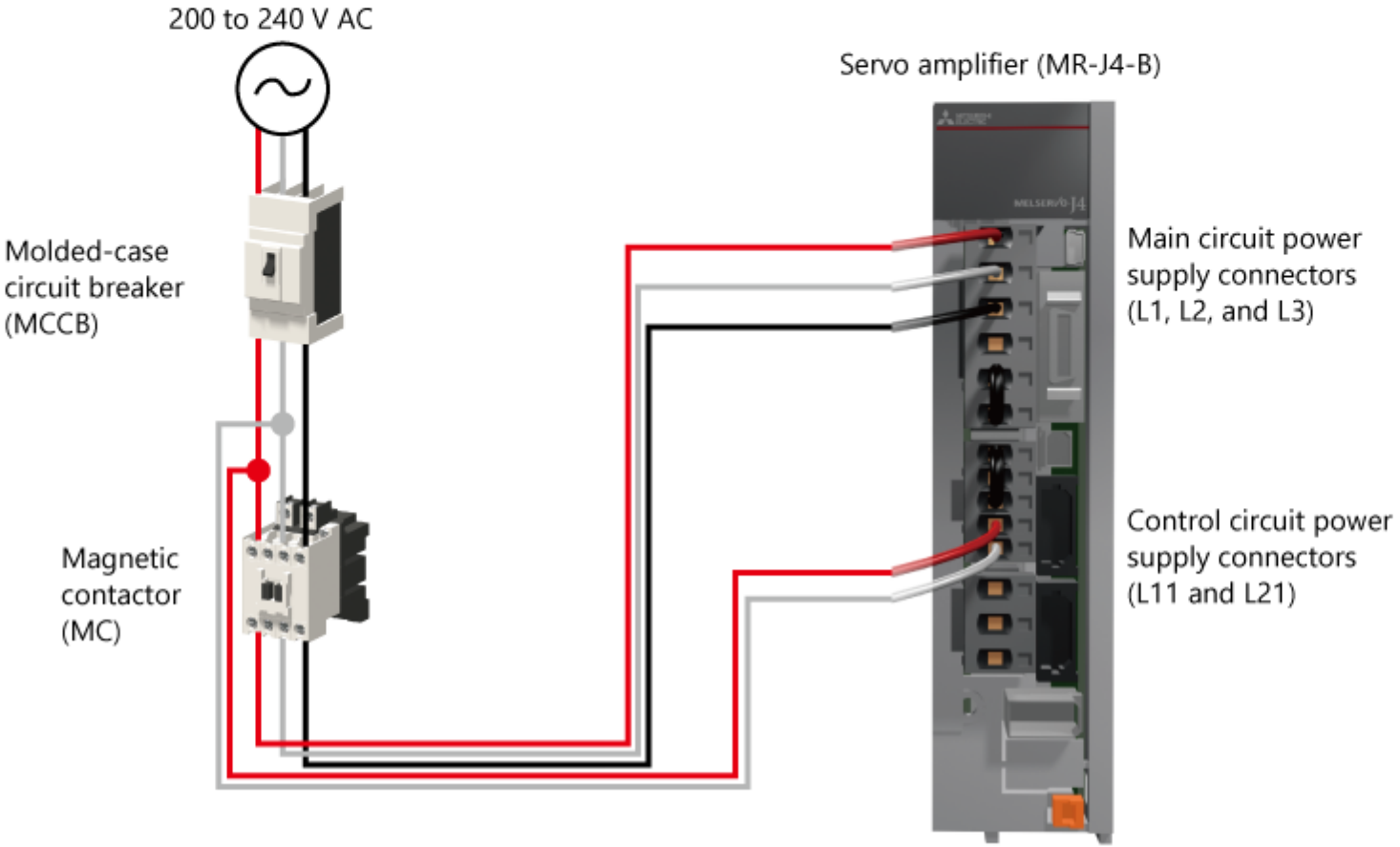
Input module (start XY: 0010)
RX40C7



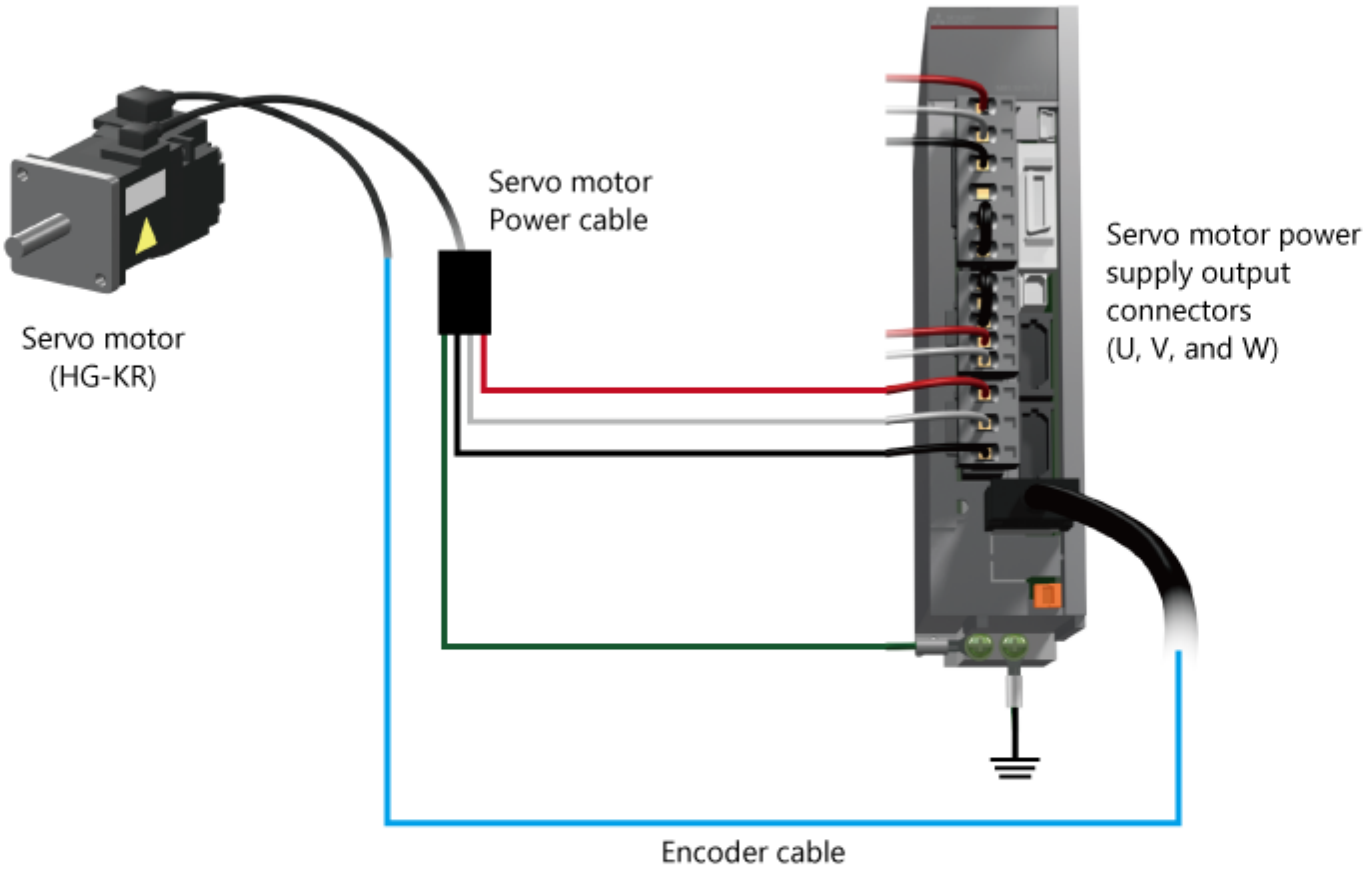
1.3.2 Wiring the Servo Amplifiers

(1) Connecting the power supply, motor power cable, and encoder cable

Connect the power supply to the main circuit power supply (L1, L2, and L3) and the control circuit power supply (L11 and L21) of the servo amplifier.
Connect the servo motor power cable and encoder cable.
The following figure is a schematic diagram. Since the actual wiring and applicable cable sizes differ depending on the capacity, refer to the Servo Amplifier Instruction Manual for details.



1.3.2 Wiring the Servo Amplifiers



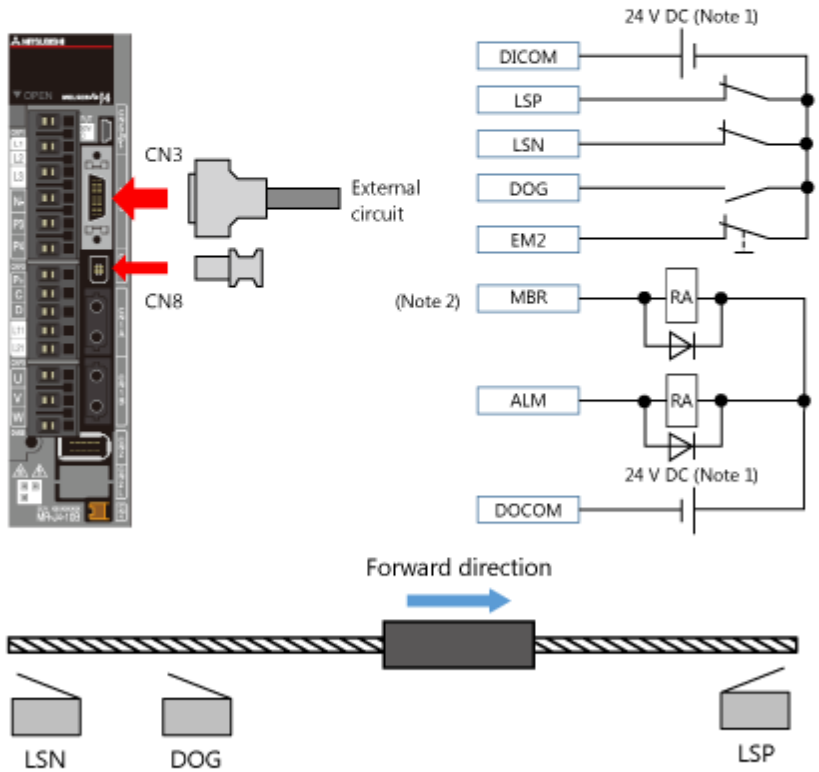
- Always use a molded-case circuit breaker (MCCB) for the input cables of the power supplies.
- Always connect a magnetic contactor (MC) between the main circuit power supply and the L1, L2, and L3 terminals of the servo amplifier.

1.3.2 Wiring the Servo Amplifiers

(2) Wiring the external circuits

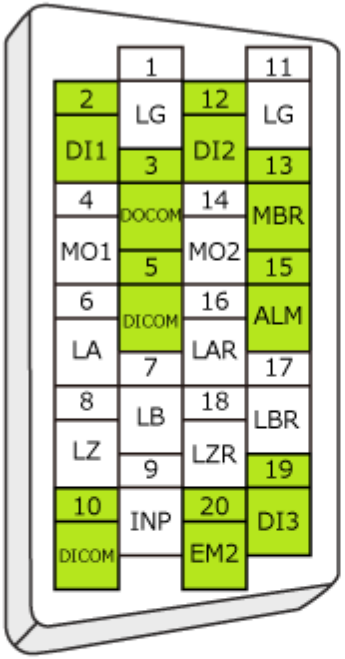
Connect the external circuits to the servo amplifier.

Connect the external circuits such as the figure shown below to CN3.
Each signal of the LSP, LSN, and DOG is set to input to the servo amplifier in section 2.4.4.
Always connect the short-circuit connector supplied with the servo amplifier to CN8.



CN3 pin arrangement

1.3.2 Wiring the Servo Amplifiers



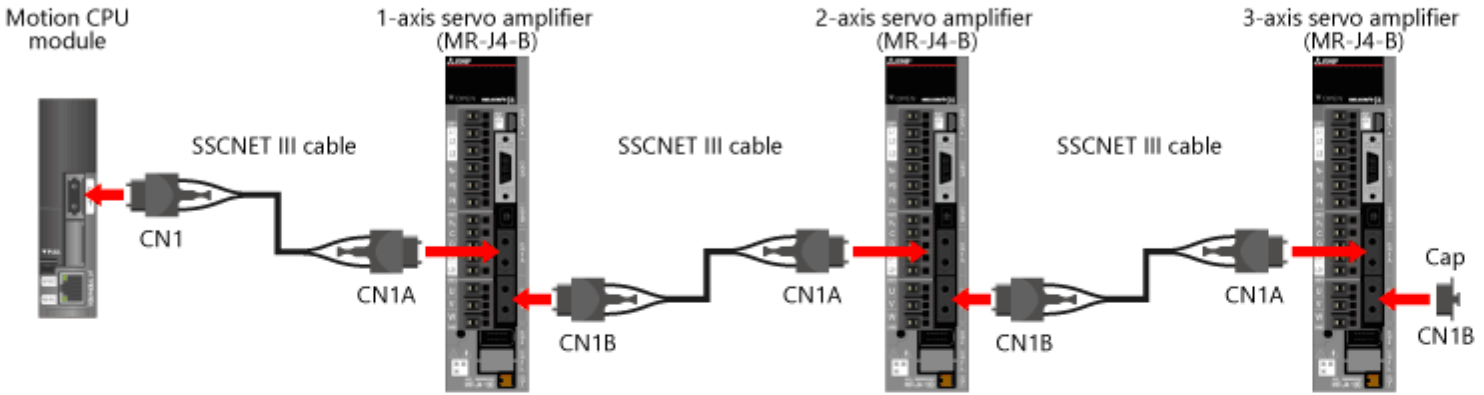
Pin No.	Abbreviation	Function/application
5 10	DICOM	Common terminals of input signal External connection to (+) of 24 V DC power supply
2	DI1 (LSP)	Hardware stroke limit switch on upper limit side
12	DI2 (LSN)	Hardware stroke limit switch on lower limit side
19	DI3 (DOG)	Proximity dog
20	EM2	Forced stop 2
13	MBR	Electromagnetic brake interlock
15	ALM	Alarm signal
3	DOCOM	Common terminals of output signal Connection to (-) of 24 V DC external power supply

(Note 1) The same power supply is used. This is a wiring example for sink I/O.

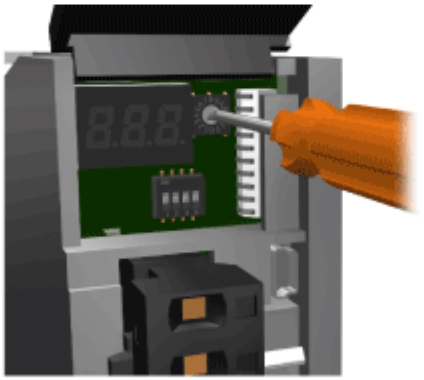
(Note 2) Use a servo motor with brake for the Z-axis, and provide an interlock circuit using the MBR output.
For the details, refer to the Servo Amplifier Instruction Manual.

1.3.3 Connecting the Communication Cables

Connect SSCNET III cables between the motion CPU module and a servo amplifier, and between the servo amplifiers.



Attach a cap to the last axis.



1-axis servo amplifier

Axis selection rotary switch (SW1)

Auxiliary axis number setting switch (SW2)

2-axis servo amplifier

Axis selection rotary switch (SW1)

Auxiliary axis number setting switch (SW2)

3-axis servo amplifier

Axis selection rotary switch (SW1)

Auxiliary axis number setting switch (SW2)

[CAUTION]
Turn off all "auxiliary axis number setting switch (SW2)" of the servo amplifiers.

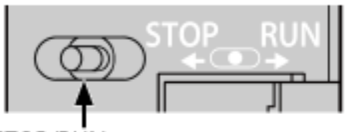
1.3.4 Turning on Power Supplies

1) Check that the RUN/STOP/RESET switches of the PLC CPU module and motion CPU module are set to STOP.

<PLC CPU module>

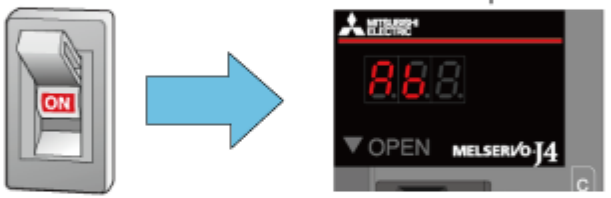
RESET/STOP/RUN

<Motion CPU module>

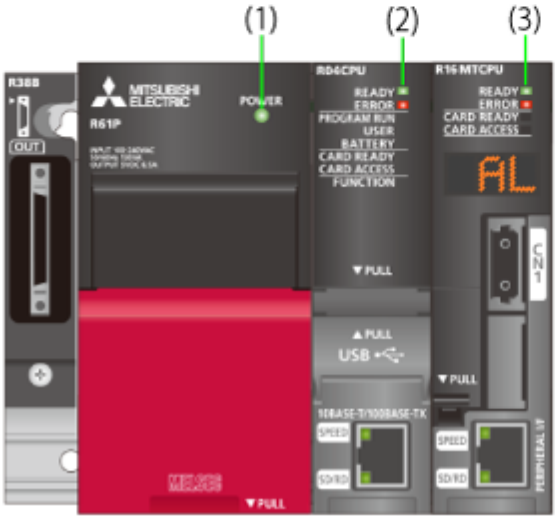


STOP/RUN

2) Turn on the power. When the servo amplifier is started up, "AA" (waiting to initialize) or "Ab" is displayed on the display.



3) LED status of programmable controller after powering on



(1)Power supply module: LED (green) ON

(2)PLC CPU module: READY LED (green) ON, ERROR LED (red) flashing

(3)Motion CPU module: READY LED (green) ON, ERROR LED (red) flashing, dot matrix LED display: AL2200H

If the parameters and programs are not written to the PLC CPU module and motion CPU module, the ERROR LED flashes red. The ERROR LED turns off when the power is turned off and on after the parameters and programs are written.

1.4 Summary of This Chapter

In this chapter, you have learned:

- System Configuration
- Sample System
- Wiring

Points

System Configuration	• Use the following modules of the MELSEC iQ-R series programmable controller. - PLC CPU: R04CPU - Motion controller: R16MTCPU - Output module: RY40NT5P - Input module: RX40C7 - Base module: R35B - Power supply module: R61P • Use the following software for the engineering environment. - GX Works3 (for PLC CPU) - MT Works2 (for motion CPU)
Sample System	• Use the servo for three axes to build a system to control the X-Y-Z arm.
Wiring	• Connect the hand opening/closing part to the output module. • Connect the controller emergency stop switch and operation selection switch to the input module. • Connect the external circuits such as the stroke limit and proximity dog to the servo amplifier. • Set the axis number with the rotary switch of the servo amplifier.

Chapter 2 Parameter Settings

In this chapter, you will learn about the parameter settings of the PLC CPU module, motion CPU module, and servo amplifiers in order.

2.1 Downloading the Sample Programs

Download the sample programs from the following table.
Open the zip file in an arbitrary location and check that each of the following project files are included.

Name of reference	File size
SampleProgram.zip	983kB

File name	Description	Software version
Sample_PLC.gx3	Project file for PLC CPU module	1.050C
Sample_Motion.mtw	Project file for motion CPU module	1.146C

2.2 PLC CPU Module Parameter Settings

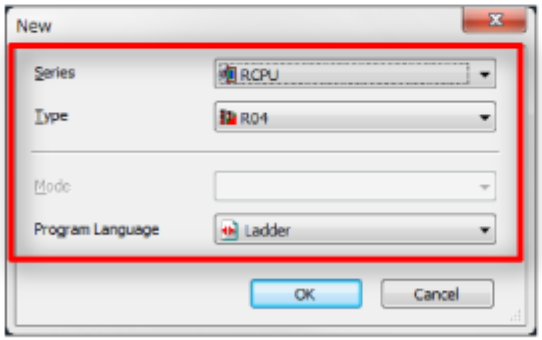
In this section, you will learn about the parameter settings of the PLC CPU module.
Create a project with the described procedure, or check that the sample project is as described.

2.2.1 Creating a GX Works3 Project

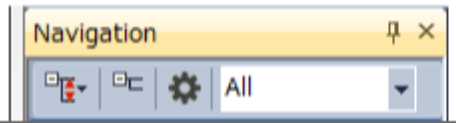
Create a GX Works3 project.

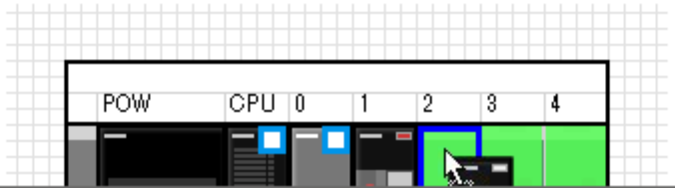
1) Start GX Works3, and select [Project] => [New].
   In the new window, configure the settings as shown in the figure below.
   Select [Module Configuration] from the project tree.
2) From the element selection window on the right side, drag and drop the same modules as in the system configuration diagram shown in section 1.1.
3) After creating a configuration diagram as for the programmable controller,
   select [Parameter] => [Fix] ( ) from [Edit] in the menu.

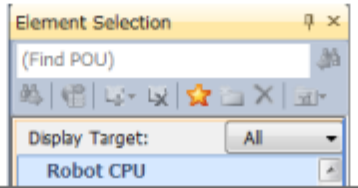
1)



2)

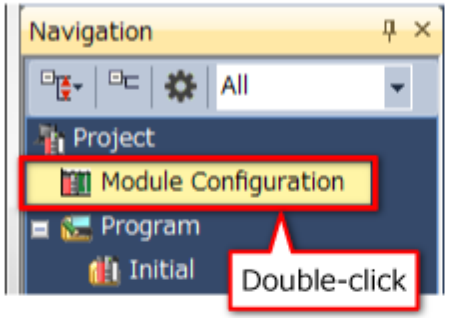


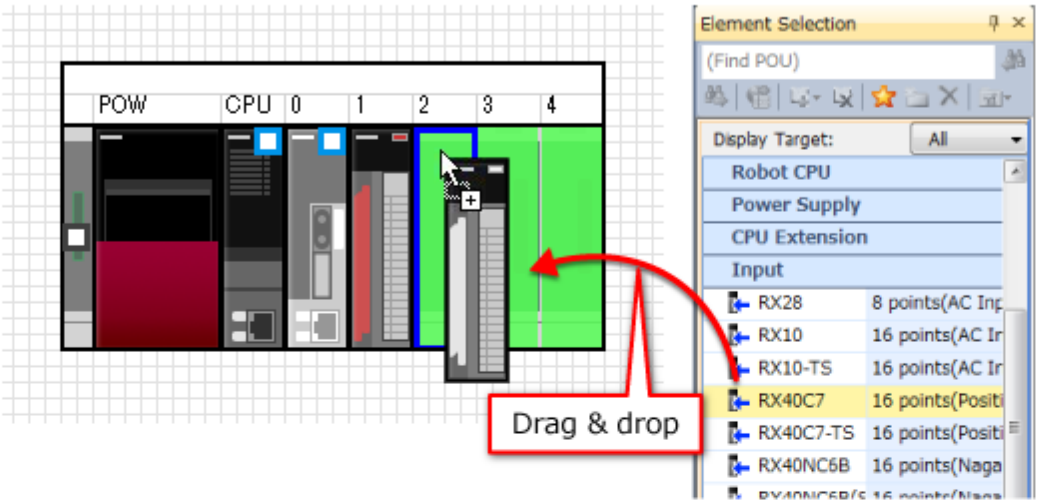




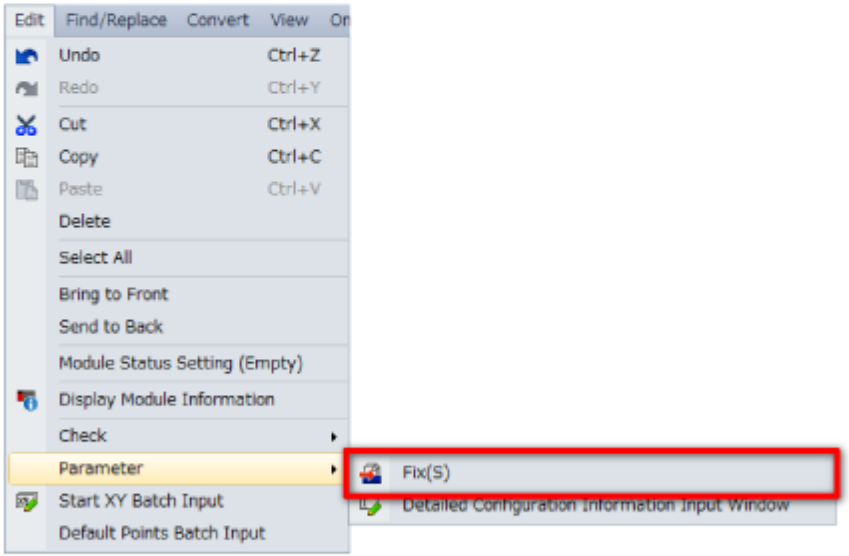
2.2.1 Creating a GX Works3 Project

2)





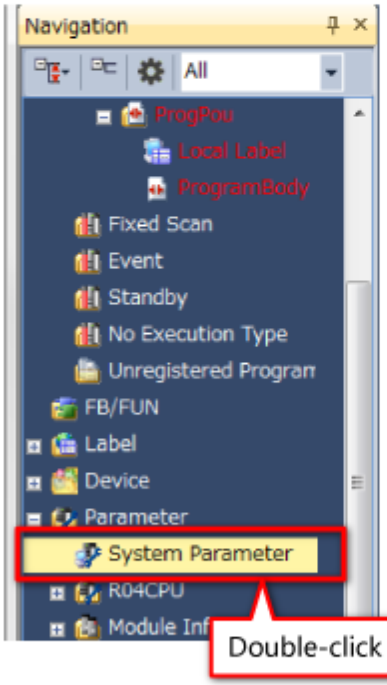
3)

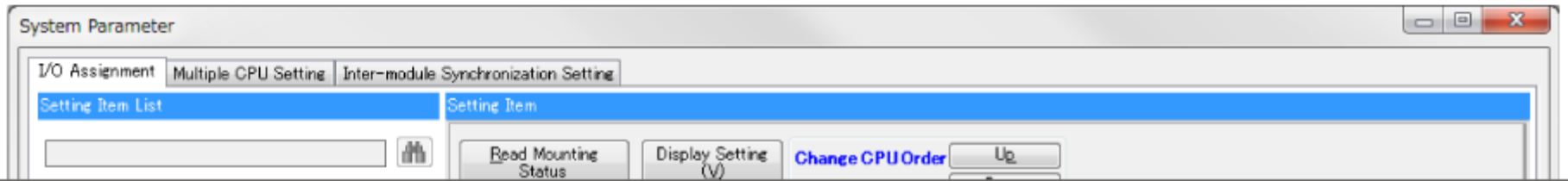


2.2.2 System Parameters

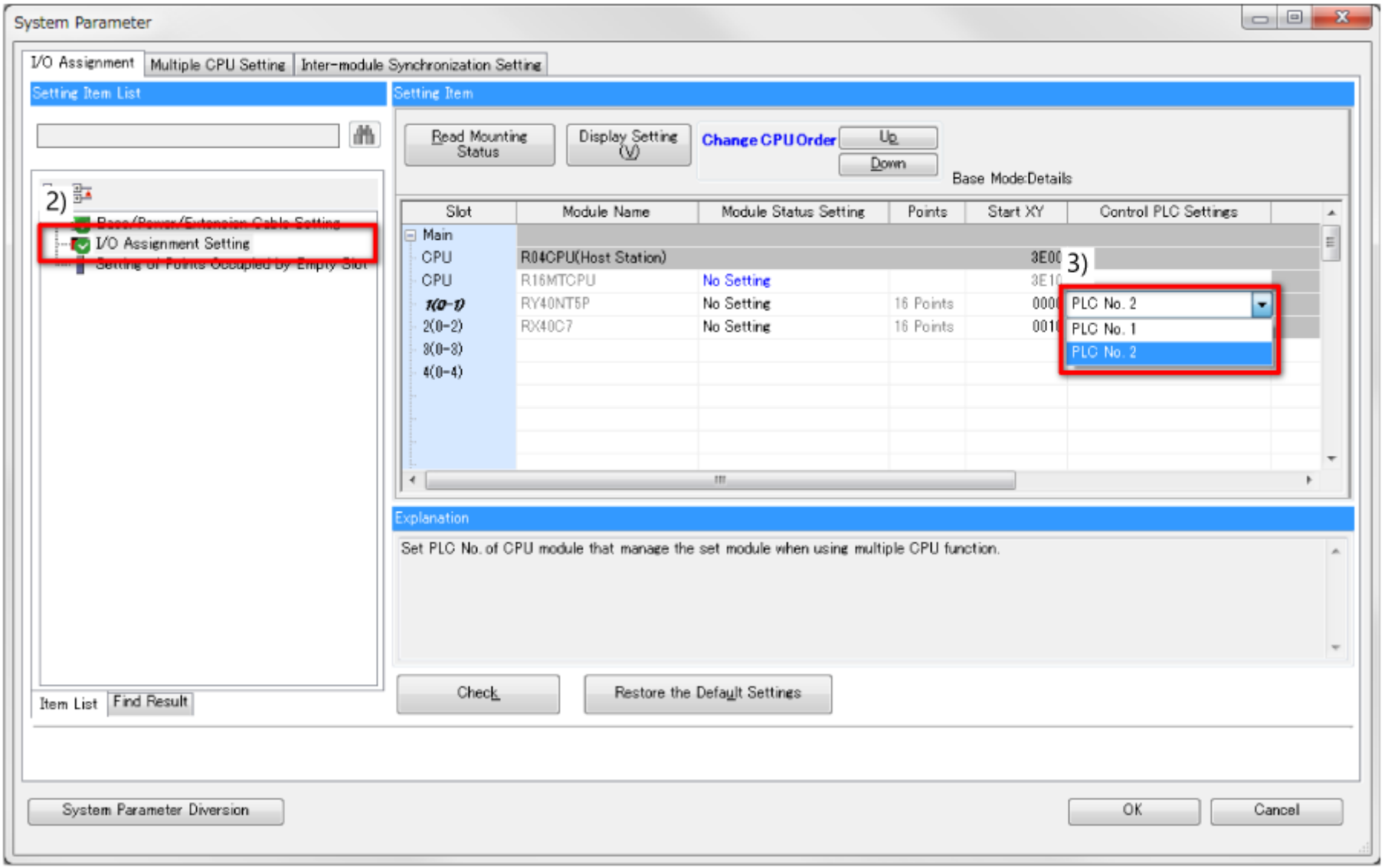
1) Select [Prameter] => [System Parameter] from the project tree in GX Works3.
   The system parameter window appears.
2) From [Setting Item List] on the left side of the window, select [I/O Assignment Setting].
3) Change control CPU settings of [RY40NT5P] output module and [RX40C7] input module to "PLC No.2".
   This will enable the output module and input module to be used in the program of the motion CPU module.
4) When the output module and input module are controlled with CPU No. 2, the colors of the output module and input module of the system configuration diagram are lightened.

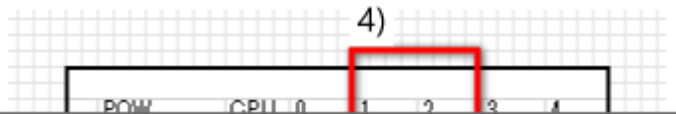
1)





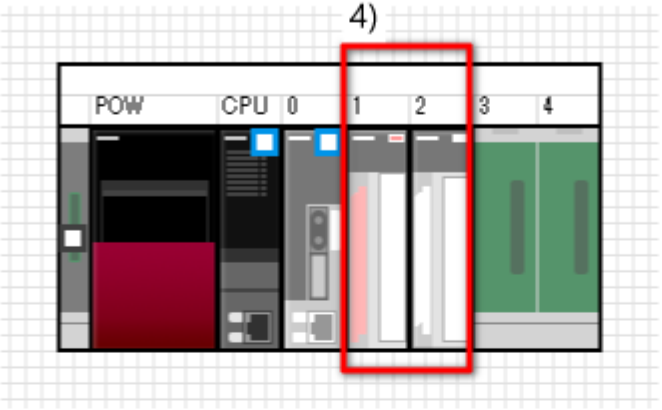
2.2.2 System Parameters





2.2.2 System Parameters

4)



2.3 Multiple CPU System

This section explains the data communication between CPU modules in a multiple CPU system.
For details of the multiple CPU system, refer to the MELSEC iQ-R Module Configuration Manual and MELSEC iQ-R CPU Module User's Manual (Application).

2.3.1 What is Multiple CPU System?

A multiple-CPU system is a system which multiple CPU modules are installed to control the I/O module and intelligent function module on each CPU module.
Furthermore, the communication between the CPU modules are performed.
When a motion CPU module is used, the system will always be the multiple CPU system.

A multiple-CPU system has the following advantages.

- The load on the processing can be distributed by assigning complicated servo control to the motion CPU module, and other controls such as the machine control and information control to the PLC CPU module.
- The number of controlling axes can be increased by using multiple motion CPU modules. Up to 192 axes can be controlled by using three R64MTCPU.
- The responsiveness of the entire system can be improved by distributing high-load processing to multiple CPU modules.

[CAUTION]
The motion CPU module cannot be set as CPU No. 1.
The PLC CPU module must be set as CPU No. 1.

2.3.2 Data Communication between CPU Modules

The data communication between the CPU modules are performed with the following two methods.

- Data communication using the CPU buffer memory area (Used for sending and receiving data at the timing of each CPU module.)
- Data communication using a fixed scan communication area (Used when matching the data sending and receiving timing between the CPU modules.)

Data communication using the CPU buffer memory is used in this course.

The refresh timing of the CPU buffer memory can be selected from two options: refreshing at END or with Q series compatible high-speed refreshing. Select refreshing at END in this course.
The refresh is performed at the END processing of the PLC CPU module side, and in the main cycle of the motion CPU module side.

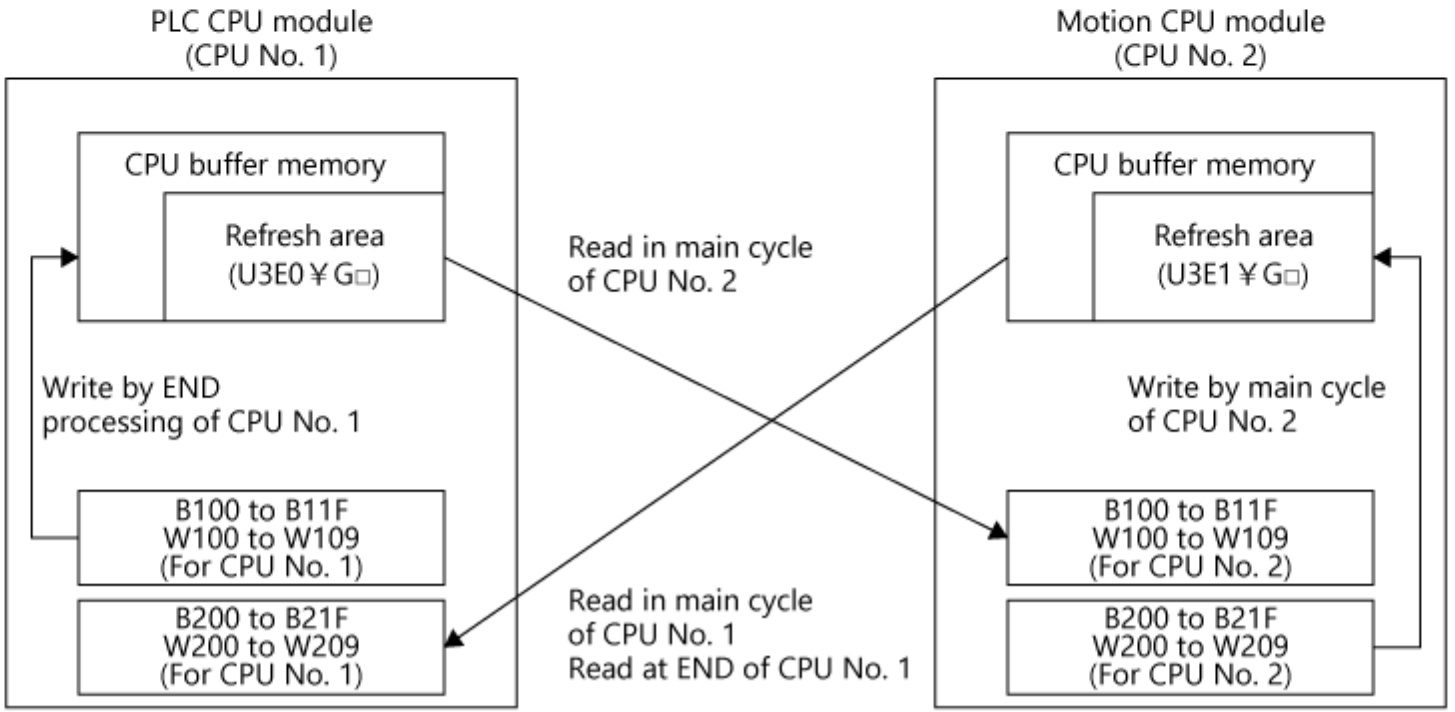
2.3.3 Data Communication Settings between CPU Modules of PLC CPU Module

(1) Operation image

The following shows the specifications for this course.
B100s and W100s are sent from CPU No. 1 to CPU No. 2 (device that is sent by the PLC CPU module)
B200s and W200s are sent from CPU No. 2 to CPU No. 1 (device that is received by the PLC CPU module)

The number of points of device is required to be set in units of 2-word.
In other words, the bit device is set in units of 32 points. When the starting device is a bit device, it is required to be specified in units of 16 points.

The following figure is an example when the number of points of a bit device is set as 2-word (= 32 points) and the number of points of a word device is set as 10-word for each CPU No. 1 and CPU No. 2.
These values are set in the sample programs.



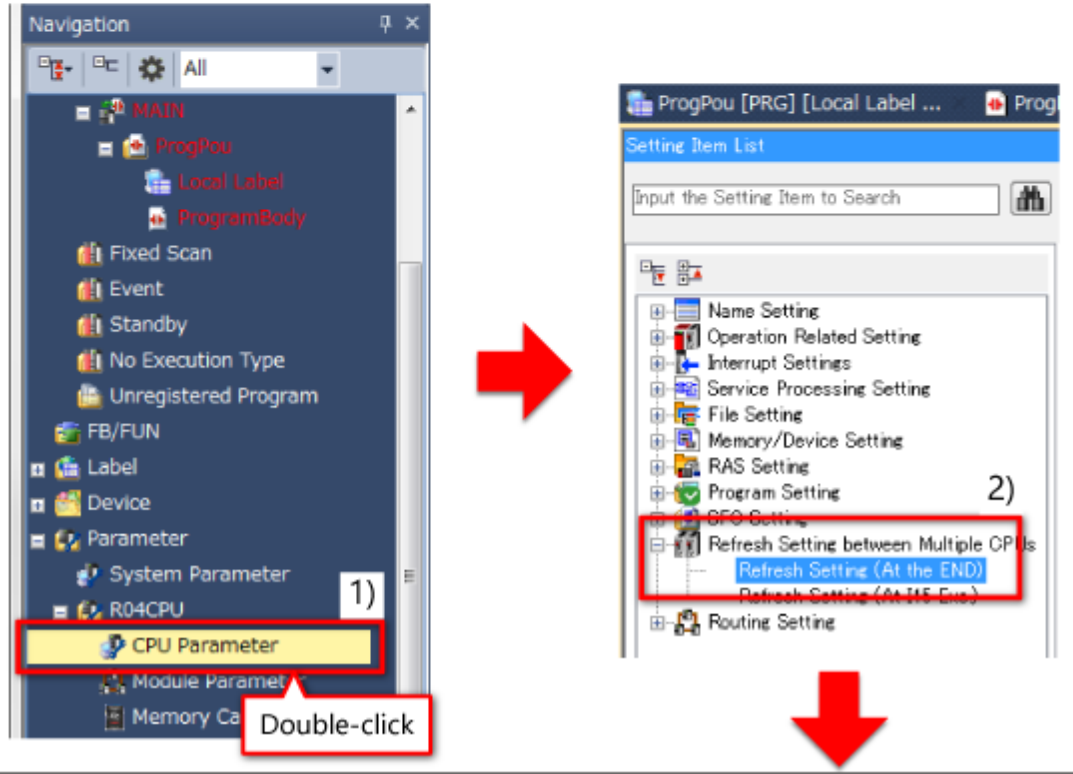
2.3.3 Data Communication Settings between CPU Modules of PLC CPU Module

(2) Setting method

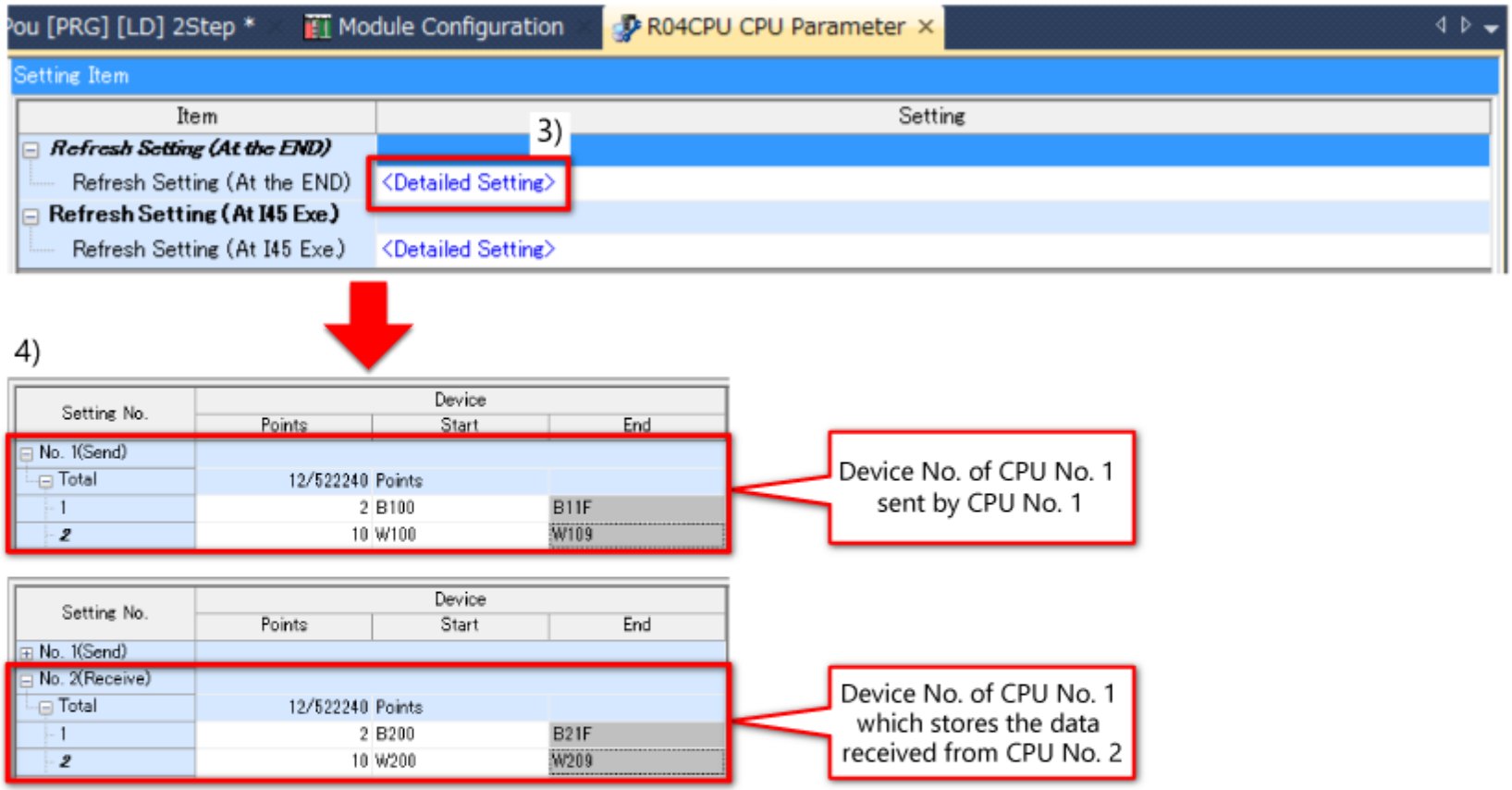
1) In the project tree, double-click [Parameter] => [R04CPU] => [CPU Parameter].
2) In the setting item list, click [Refresh Setting between Multiple CPUs] => [Refresh Setting (At the END)].
3) In setting item, double-click <Detailed Setting> of [Refresh Setting (At the END)].
4) Set the device No. sent by CPU No. 1, and the device No. of CPU No. 1 that receives and stores the data sent from CPU No. 2.

The memory offset can be displayed or hidden by clicking the [Detailed Setting] button in the [Refresh Setting (At the END)] window.

When these settings are completed, convert the project and save.



2.3.3 Data Communication Settings between CPU Modules of PLC CPU Module



2.4 Motion CPU Module Parameter Settings

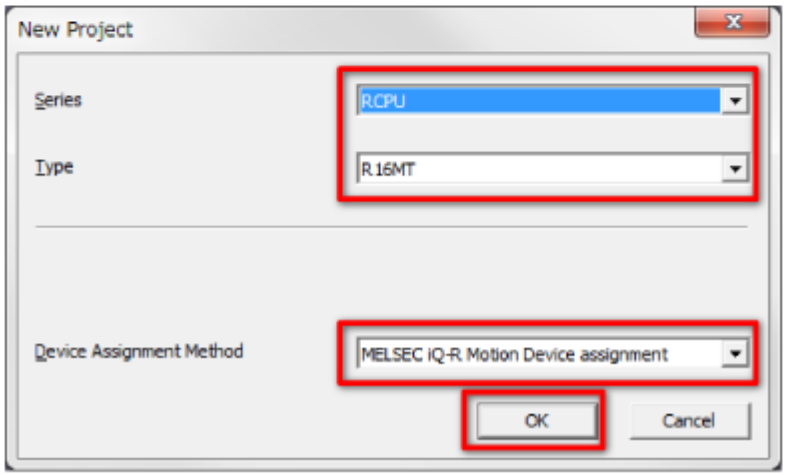
In this section, you will learn about the parameter settings of the motion CPU module.
Create a project with the described procedure, or check that the sample project is as described.

2.4.1 Creating an MT Works2 Project

Create an MT Developer2 project.

1) Start MT Developer2, and select [Project] => [New].
   In the new project window, configure the settings as shown in the figure below.
   The details of "Q series Motion compatible Device assignment" are explained in section 3.1.
   Click the [OK] button to confirm.
2) The [System Parameter Diversion] window appears.
   Click the [System Parameter Diversion] button.
   The R series common parameters can be diverted from the GX Works3 project that was created before.
3) In the [Open] window, select the project saved in section 2.3.3.
   Click the [OK] button to confirm.
4) The [Self CPU Selection] window appears.
   Set the CPU No. of the motion CPU module.
   Select "CPU2" in this course.
   Click the [OK] button to confirm.

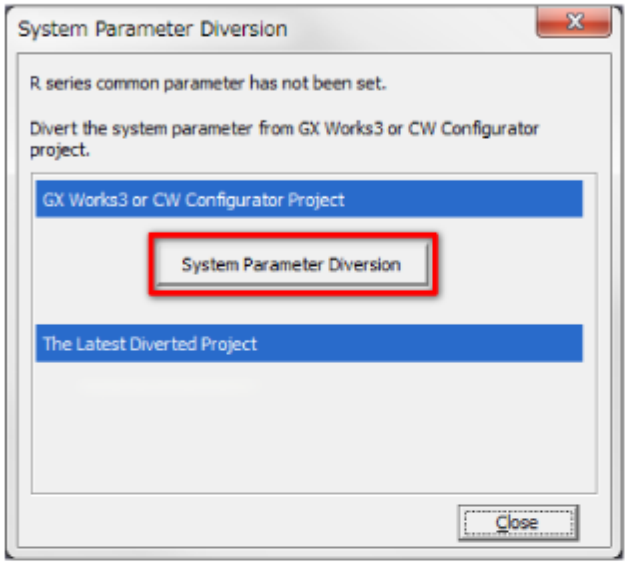
1)



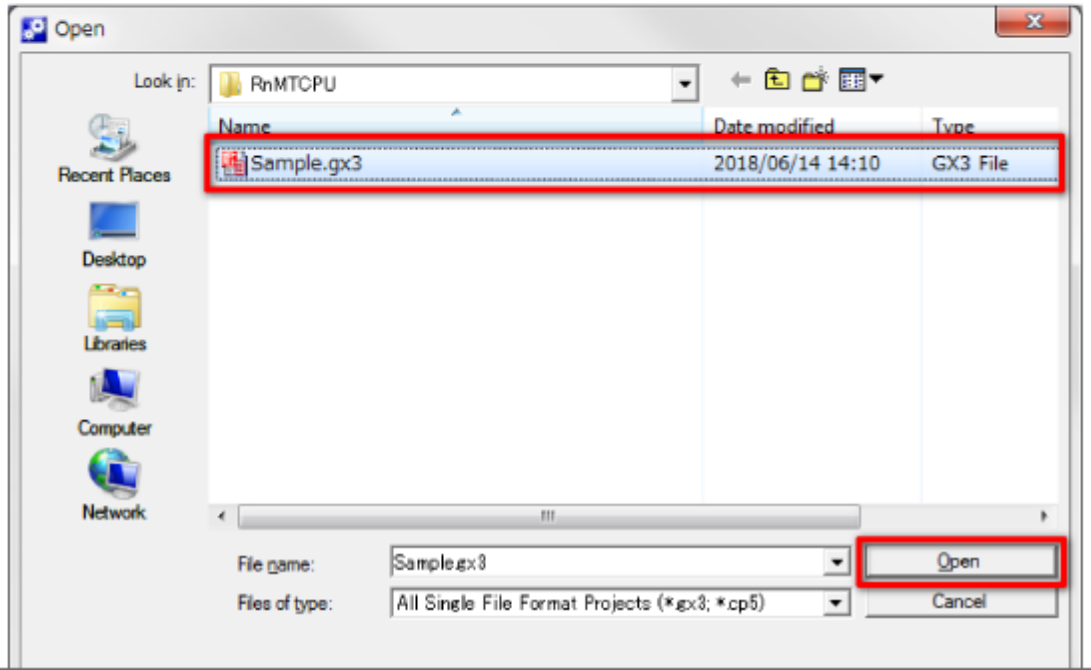
2)

2.4.1 Creating an MT Works2 Project

2)

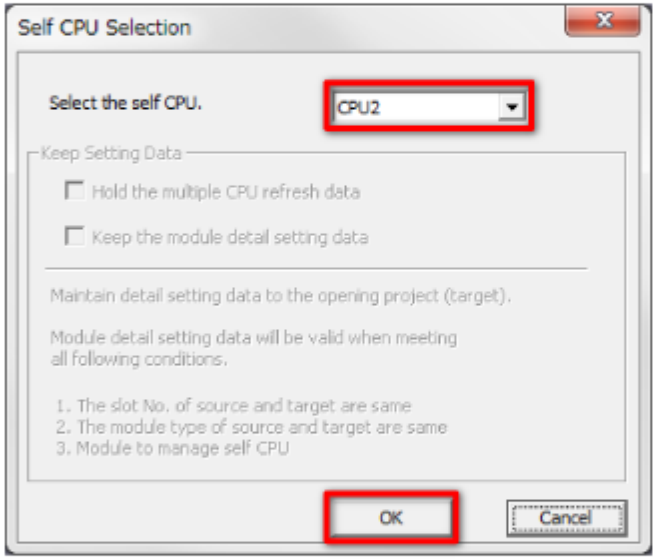


3)



2.4.1 Creating an MT Works2 Project

4)

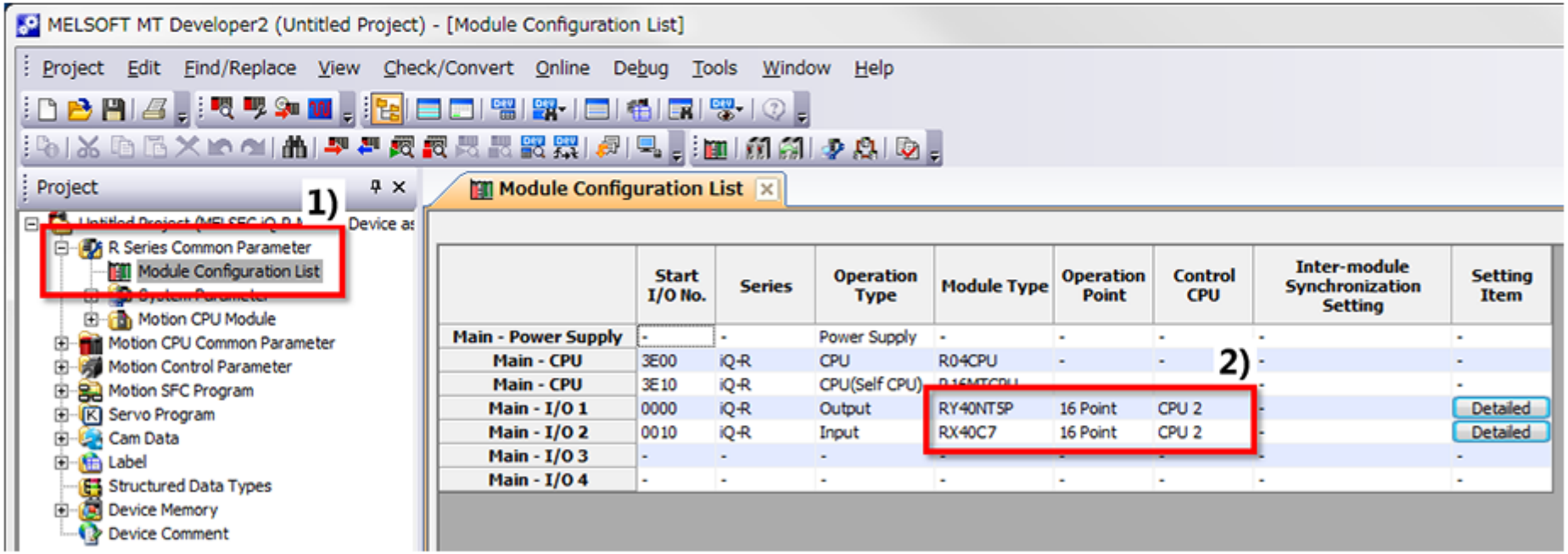


2.4.2 R Series Common Parameters

The necessary settings for the R series common parameters are set as a result of diverting the settings from GX Works3 project.
Check the parameters that have been changed from the default values. The procedures are explained in the order of the items in the project tree of MT Developer2.

(1) Module configuration list

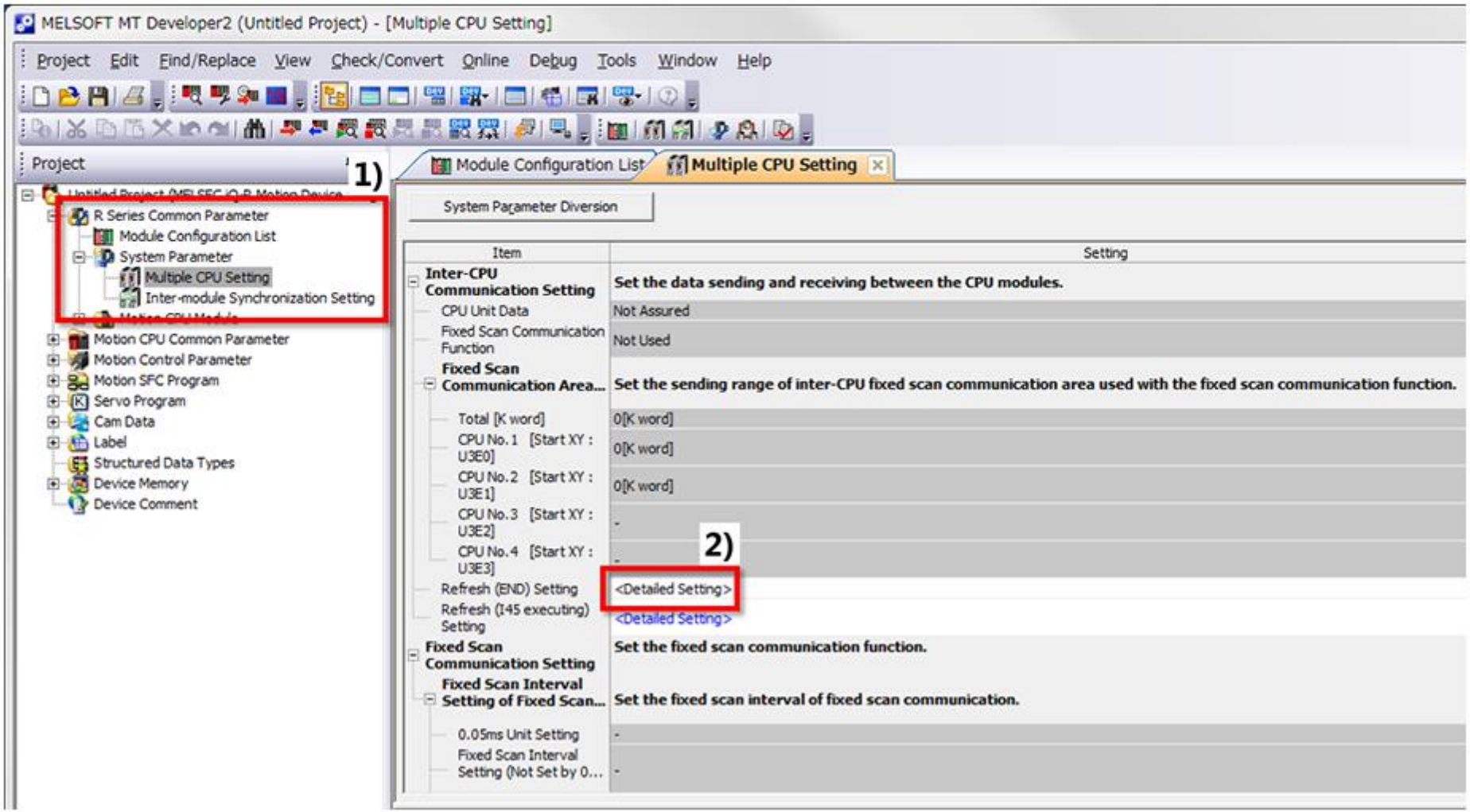
1) From the project tree, double-click [R Series Common Parameter] =>[Module Configuration List].
   The model names of the used input/output modules are displayed.
2) Check that the control CPU is "CPU2".



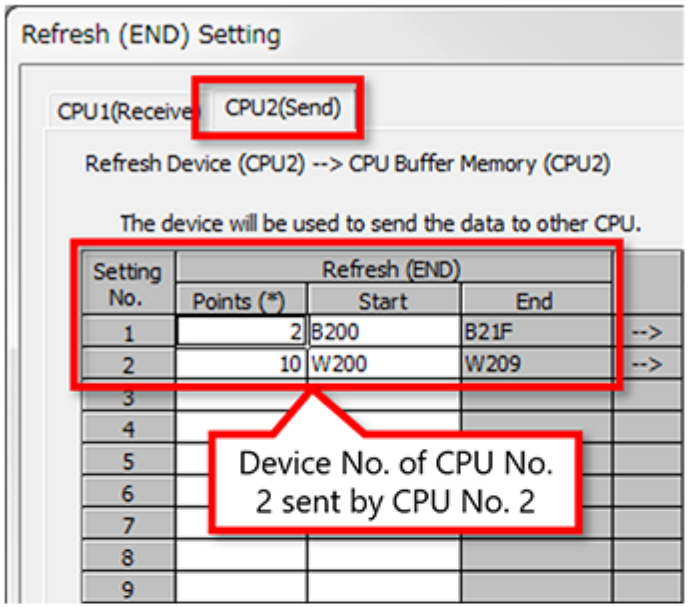
2.4.2 R Series Common Parameters

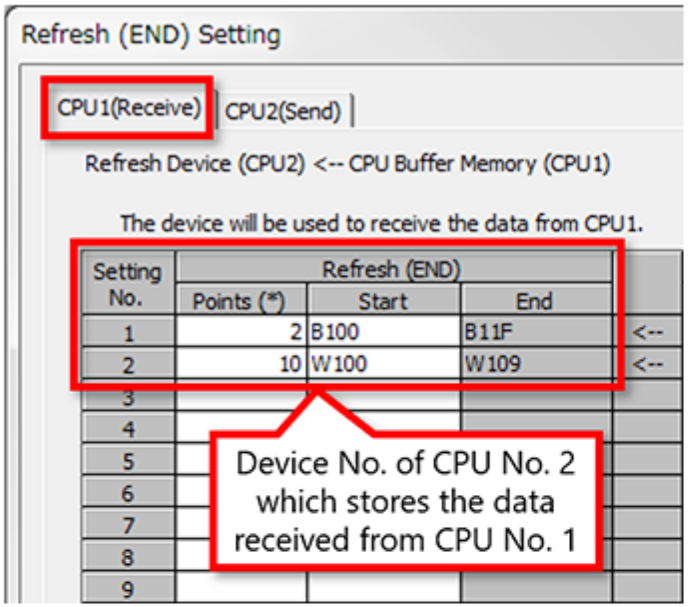
(2) Multiple CPU setting

1) From the project tree, double-click [R Series Common Parameter] => [System Parameter] => [Multiple CPU Setting].
2) Double-click <Detailed Setting> of [Inter-CPU Communication Setting] => [Refresh (END) Setting] in the multiple CPU setting window.
   Check that the refresh devices that are set in GX Works3 are registered.



2.4.2 R Series Common Parameters



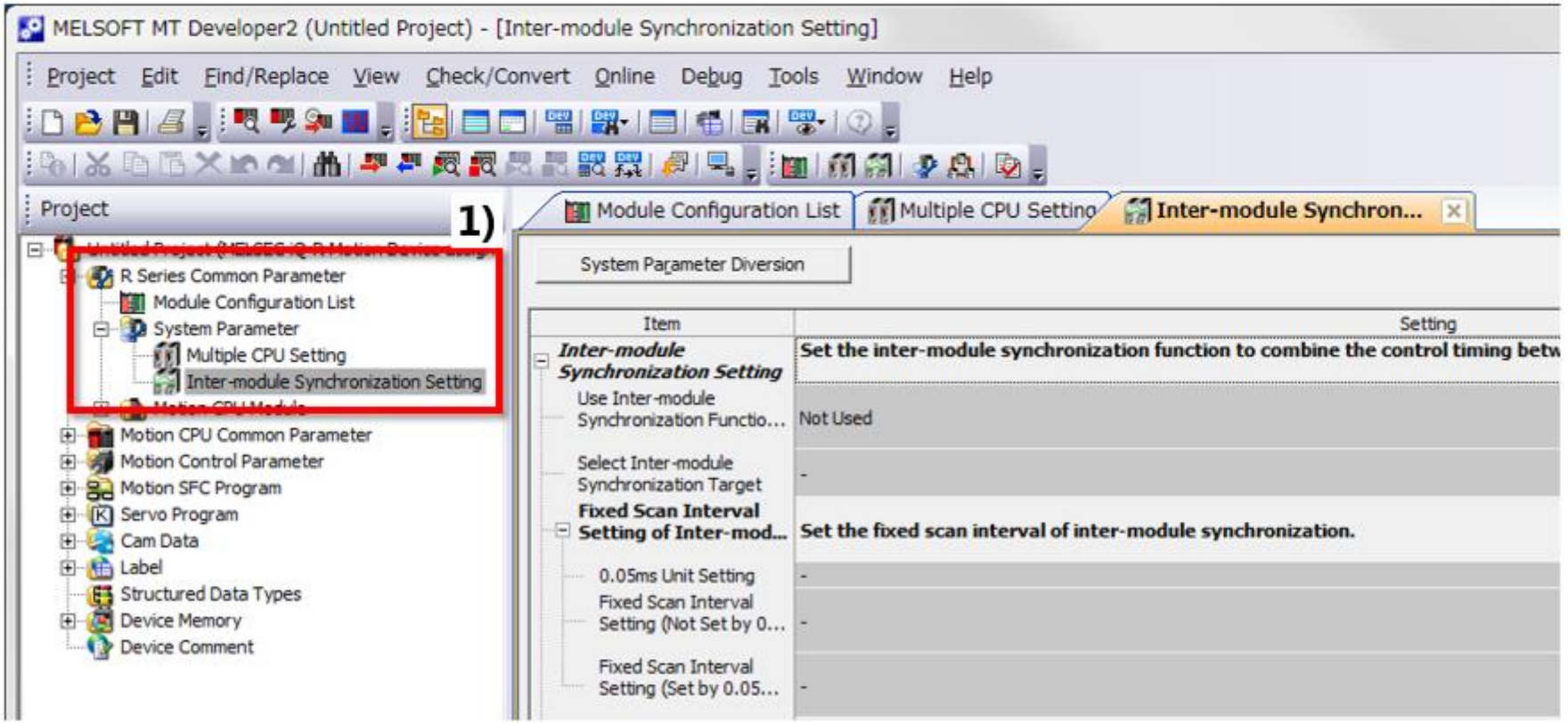


2.4.2 R Series Common Parameters

(3) Inter-module synchronization setting

1) From the project tree, double-click [R Series Common Parameter] => [System Parameter] => [Inter-module Synchronization Setting].

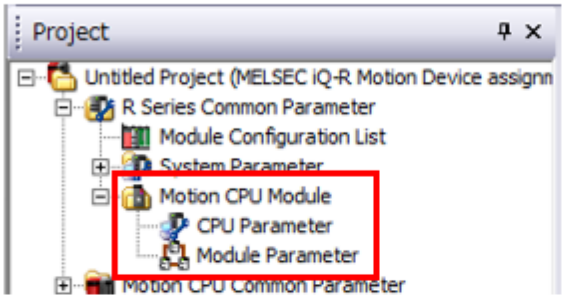
If the inter-module synchronization setting is changed in GX Works3, it is also changed in MT Developer2.
The inter-module synchronization setting is not changed in this course.



2.4.2 R Series Common Parameters

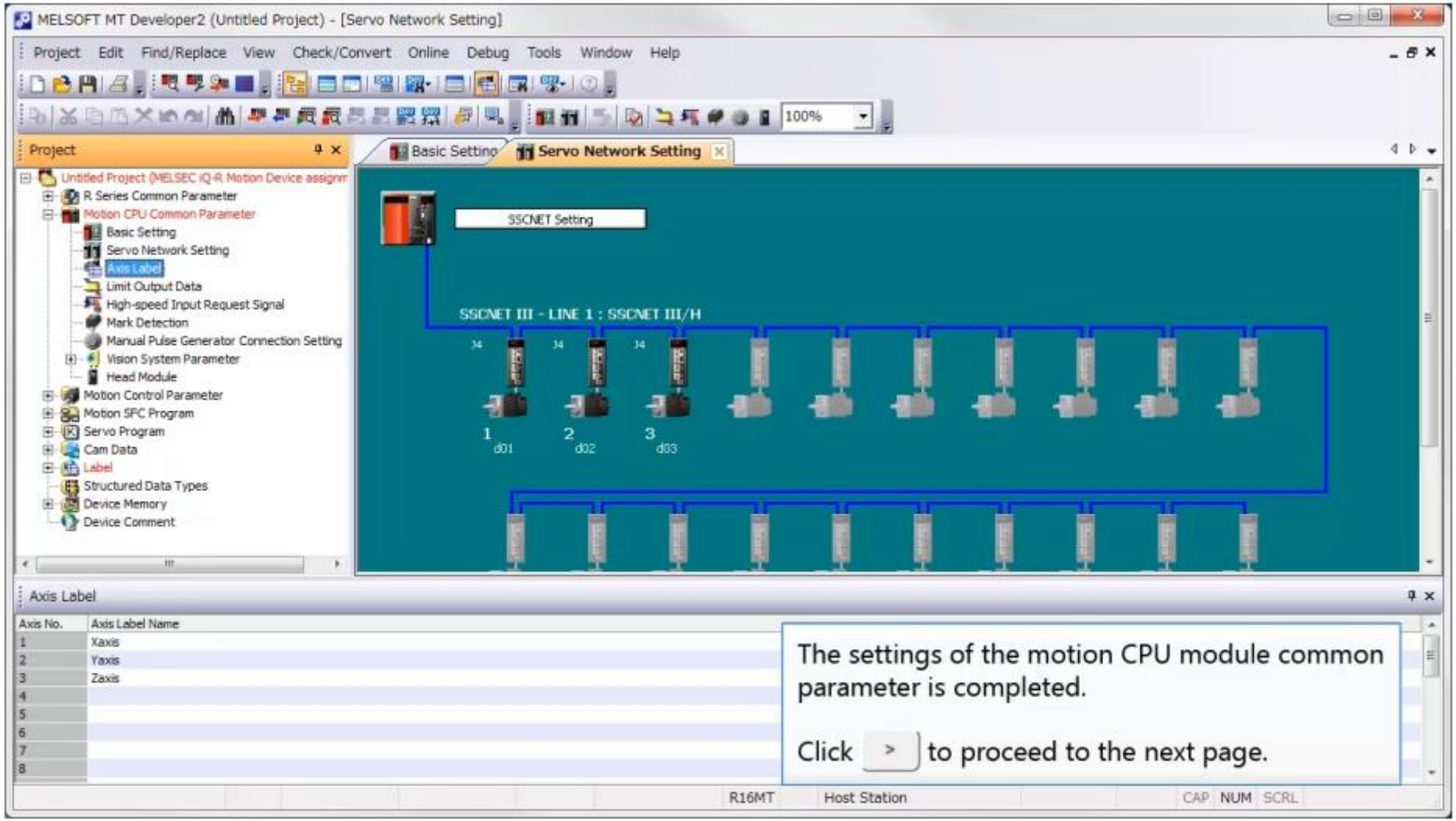
(4) Motion CPU module

The following functions are not used in this course.



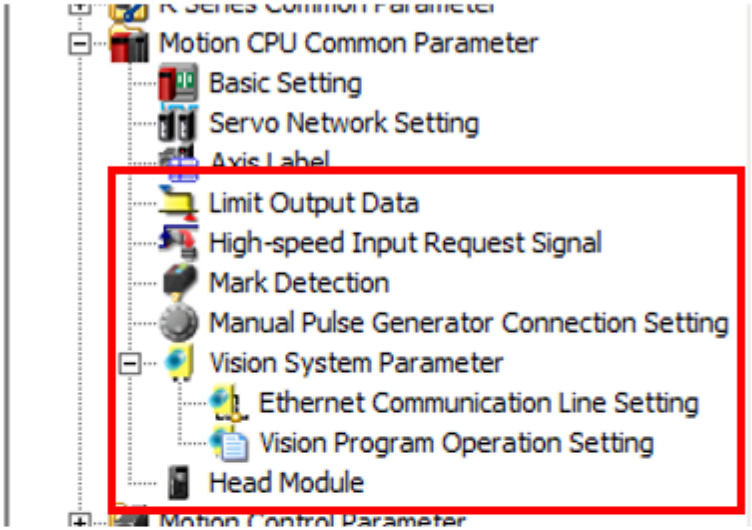
Function	Description
CPU parameter	The operation of the motion CPU module function is set in CPU parameter. Programming Manual (Common) Chapter 2 COMMON PARAMETERS 2.2 R Series Common Parameters
Module parameter	Securities and the own node settings to communicate with other devices by using PERIPHERAL interface of the motion CPU module are set in module parameter. Programming Manual (Common) Chapter 2 COMMON PARAMETERS 2.2 R Series Common Parameters

2.4.3 Motion CPU Module Common Parameters



2.4.3 Motion CPU Module Common Parameters

The following functions are not used in this course.



Function	Description
Limit output data	The limit output data setting is required to be set when the limit output function is used. Programming Manual (Common) Chapter 4 AUXILIARY AND APPLIED FUNCTIONS 4.1 Limit Switch Output Function
High-speed input request signal	The high-speed input request signal setting is required to be set when functions such as the mark detection function are used. Programming Manual (Common) Chapter 4 AUXILIARY AND APPLIED FUNCTIONS

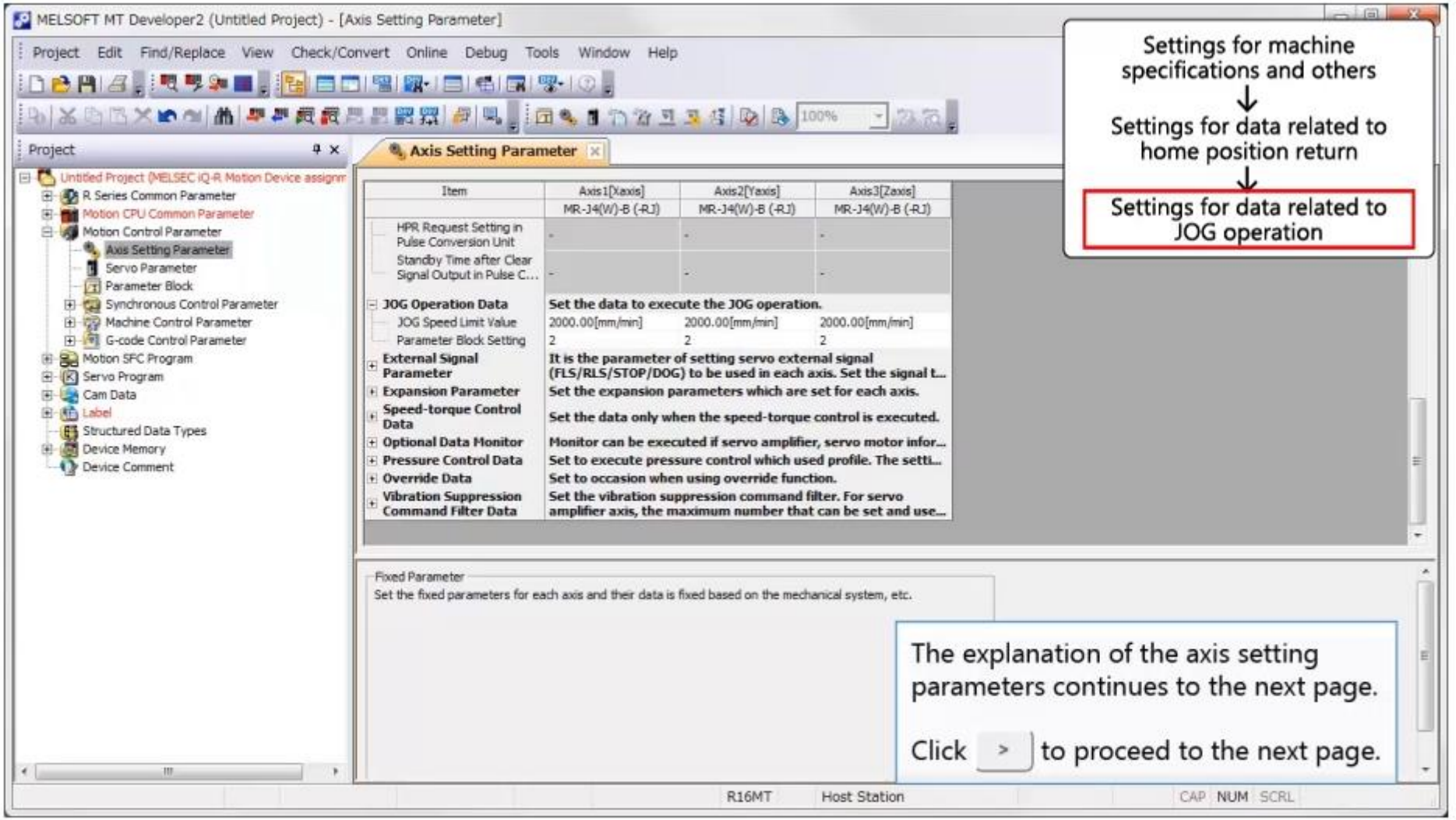
2.4.3 Motion CPU Module Common Parameters

	4.2 External Input Signal
Mark detection	The mark detection setting is required to be set when the mark detection function is used. Programming Manual (Common) Chapter 4 AUXILIARY AND APPLIED FUNCTIONS 4.3 Mark Detection Function
Manual pulse generator connection setting	The manual pulse generator connection setting is required to be set when the manual pulse is used. Programming Manual (Common) Chapter 2 COMMON PARAMETERS 2.3 Motion CPU Common Parameter
Vision system parameter	The vision system parameter is required to be set when the vision system is used. Programming Manual (Common) Chapter 6 COMMUNICATION FUNCTIONS 6.5 Vision System Connection Function
Head module	The head module is required to be set when the LJ72MS15 head module or MR-MT2010 sensing module is used. Programming Manual (Common) Chapter 5 FUNCTIONS USED WITH SSCNET COMMUNICATION 5.6 Connection of SSCNETIII/H Head Module

2.4.3 Motion CPU Module Common Parameters

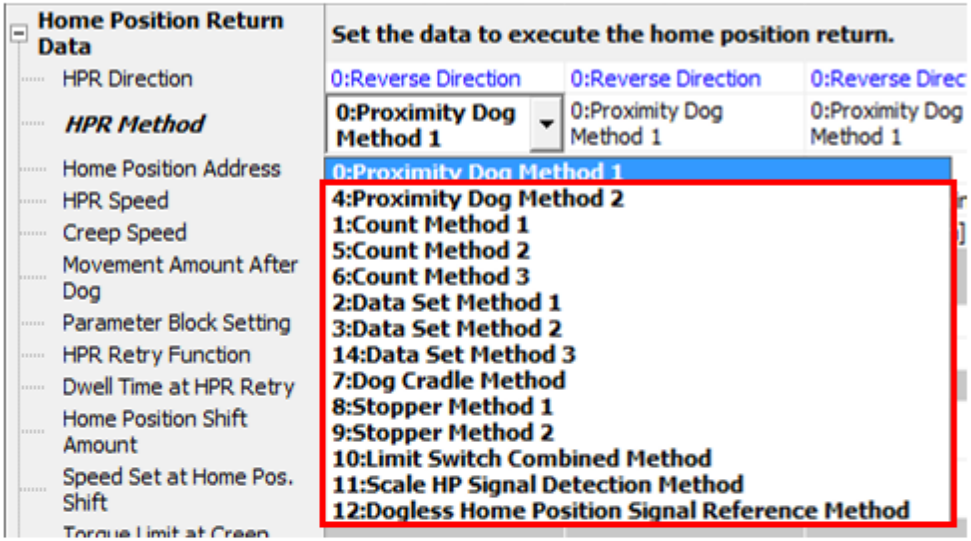
	5.6 Connection of SSCNETIII/H Head Module 5.7 Connection of Sensing Module

2.4.4 Motion Control Parameters (Axis Setting Parameters)



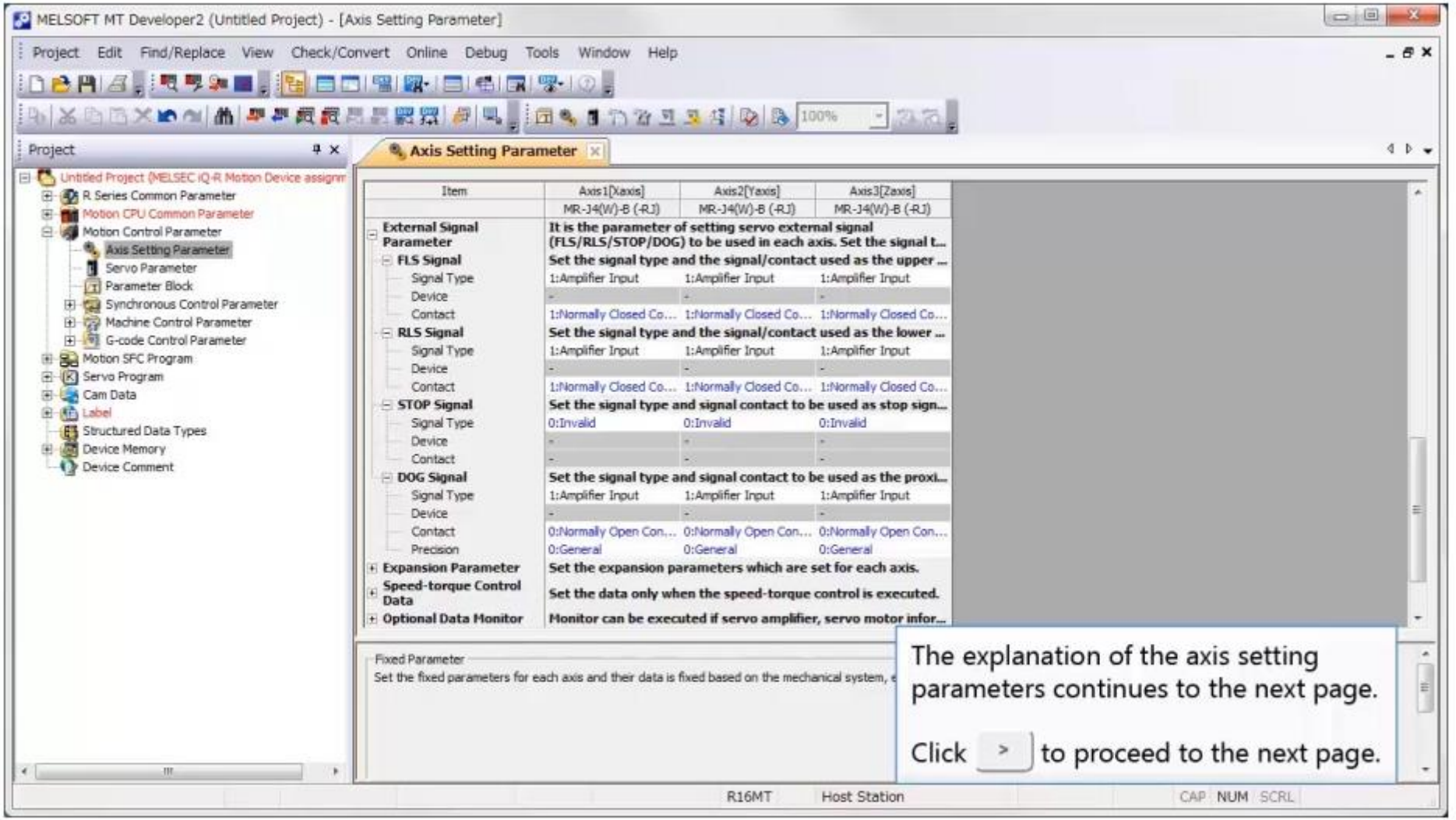
2.4.4 Motion Control Parameters (Axis Setting Parameters)

For details on the home position return method and other methods, refer to the following manual.



Programming Manual (Positioning Control)
Chapter 3 PARAMETERS FOR POSITIONING CONTROL
3.4 Home Position Return Data
Chapter 5 POSITIONING CONTROL
5.21 Home Position Return

2.4.4 Motion Control Parameters (Axis Setting Parameters)



2.4.4 Motion Control Parameters (Axis Setting Parameters)

The following functions are not used in this course.

Item	Axis1[Xaxis]	Axis2[Yaxis]	Axis3[Zaxis]
	MR-J4(W)-B (-RJ)	MR-J4(W)-B (-RJ)	MR-J4(W)-B (-RJ)
Fixed Parameter	Set the fixed parameters for each axis and their data is fixed...		
Home Position Return Data	Set the data to execute the home position return.		
JOG Operation Data	Set the data to execute the JOG operation.		
External Signal Parameter	It is the parameter of setting servo external signal (FLS/RLS/STOP/DOG) to be used in each axis. Set the signal t...		
Expansion Parameter	Set the expansion parameters which are set for each axis.		
Speed-torque Control Data	Set the data only when the speed-torque control is executed.		
Optional Data Monitor	Monitor can be executed if servo amplifier, servo motor infor...		
Pressure Control Data	Set to execute pressure control which used profile. The setti...		
Override Data	Set to occasion when using override function.		
Vibration Suppression Command Filter Data	Set the vibration suppression command filter. For servo amplifier axis, the maximum number that can be set and use...		

Function	Description
Expansion parameters	The expansion parameters are set when the following operation is performed with the parameters set in each axis. • Individually monitor the torque limit values of the positive direction and negative direction. • Change the acceleration/deceleration time when the speed is changed. • Specify the positioning direction when performing the positioning control in the absolute method with the degree axis. Programming Manual (Positioning Control) Chapter 3 PARAMETERS FOR POSITIONING CONTROL

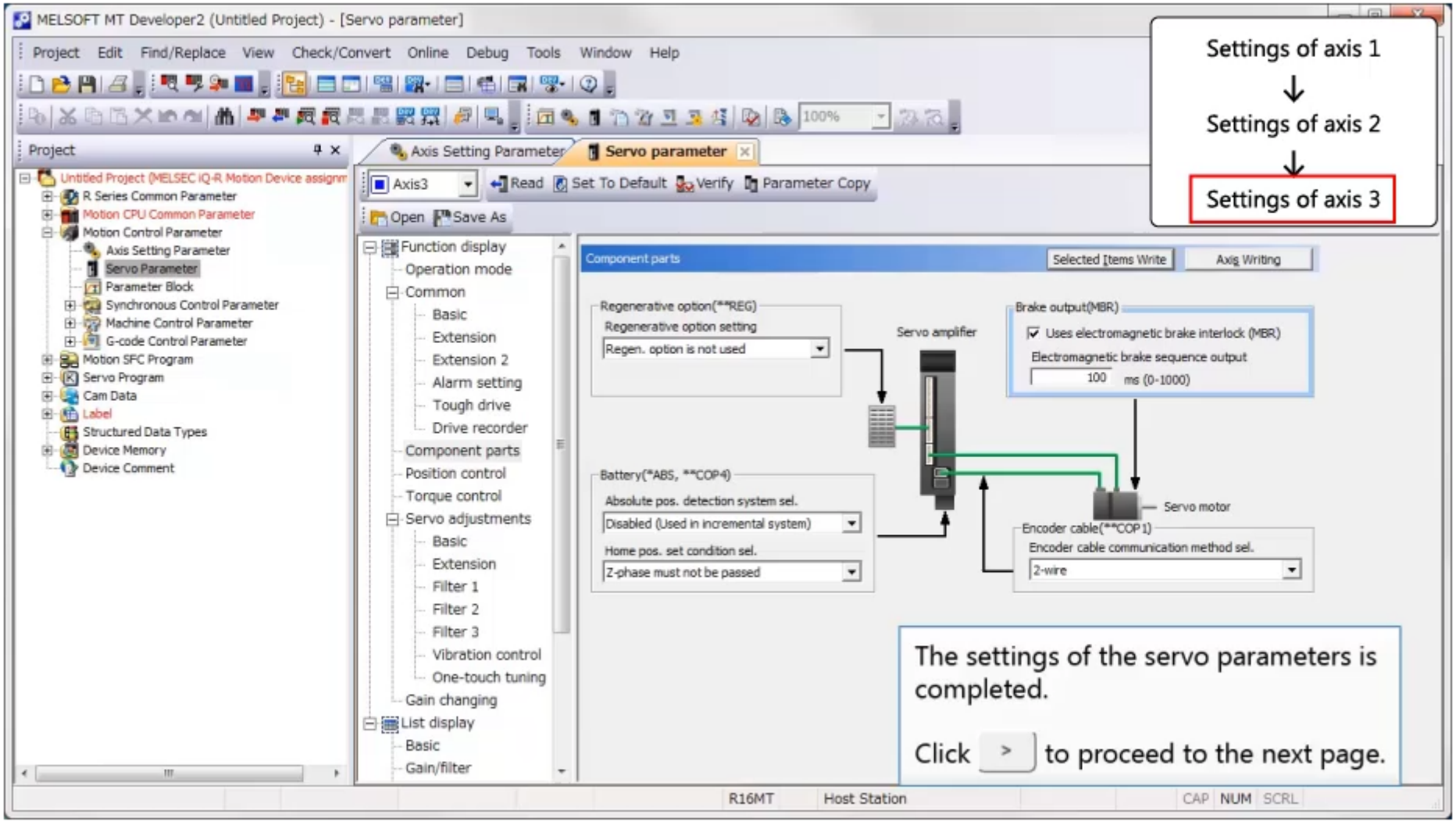
2.4.4 Motion Control Parameters (Axis Setting Parameters)

	3.7 Expansion Parameters
Speed-torque control data	Set the speed-torque control data when the speed-torque control is performed. Programming Manual (Positioning Control) Chapter 3 PARAMETERS FOR POSITIONING CONTROL 3.8 Speed-Torque Control Data
Optional data monitor	Set the optional data monitor items when the optional data monitor function is used. The optional data monitor function is used to store data in the servo amplifier to a specified word device and monitor the data. Programming Manual (Common) Chapter 5 FUNCTIONS USED WITH SSCNET COMMUNICATION 5.2 Optional Data Monitor
Pressure control data	Set the pressure control data when the pressure profile is used. Programming Manual (Positioning Control) Chapter 3 PARAMETERS FOR POSITIONING CONTROL 3.9 Pressure Control Data Chapter 7 AUXILIARY AND APPLIED FUNCTIONS 7.7 Pressure Control
	Set the override data when the override function is used. Set the override ratio of 0.0 to 300.0 [%] in 0.1 [%] increments for the command speed during positioning control.

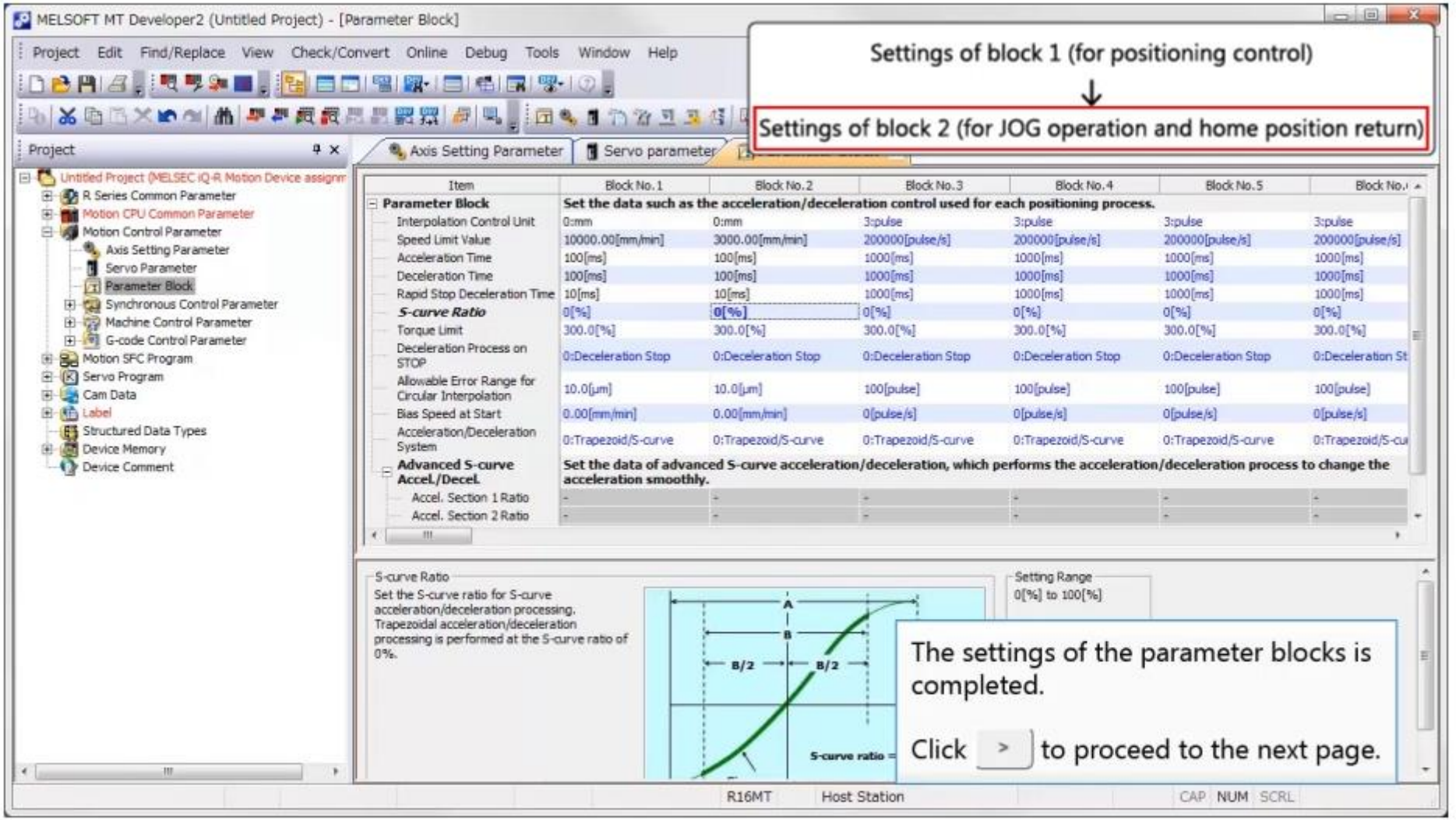
2.4.4 Motion Control Parameters (Axis Setting Parameters)

Override data	The value obtained by multiplying the speed command by the override ratio is the actual feed rate. Programming Manual (Positioning Control) Chapter 3 PARAMETERS FOR POSITIONING CONTROL 3.10 Override Data Chapter 7 AUXILIARY AND APPLIED FUNCTIONS 7.8 Override Function
Vibration suppression command filter data	Set the vibration suppression command filter data when the vibration suppression command filter is used. This function is used to suppress vibrations in position control on the load-side such as vibrations of the work platform and shaking of the machine frame. Programming Manual (Positioning Control) Chapter 3 PARAMETERS FOR POSITIONING CONTROL 3.11 Vibration Suppression Command Filter Data Chapter 7 AUXILIARY AND APPLIED FUNCTIONS 7.9 Vibration Suppression Command Filter

2.4.4 Motion Control Parameters (Servo Parameters)

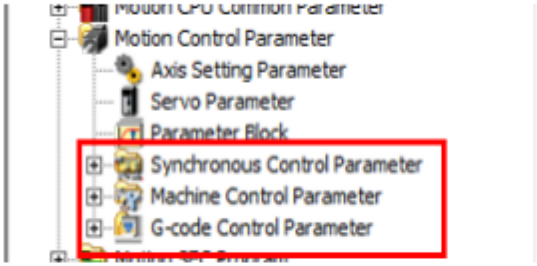


2.4.4 Motion Control Parameters (Parameter Blocks)



2.4.4 Motion Control Parameters (Others)

The following functions are not used in this course.



Function	Description
Synchronous control parameters	This function is used when the synchronous control is performed. Programming Manual (Advanced Synchronous Control)
Machine control parameters G-code control parameters	This function is used when the add-on library for iQ-R motion controller is used. Programming Manual (Machine Control) Programming Manual (G-code Control)

2.5 Summary of This Chapter

In this chapter, you have learned:

- Downloading the Sample Programs
- PLC CPU Parameter Settings
- Multiple CPU system
- Motion CPU Parameter Settings

Points

PLC CPU parameter settings	• Create a module configuration diagram in GX Works3. • Change the output module and input module to CPU No. 2 (motion CPU) control in the system parameters.
Multiple CPU system	• When a motion CPU is used, the system will always be the multiple-CPU system. • The motion CPU cannot be set as CPU No. 1. • The data communication between CPU modules are performed in two methods: data communication using the CPU buffer memory and data communication using a fixed scan communication area. • The data communication using the CPU buffer memory is refreshed at END or with Q compatible high-speed refreshing.
Motion CPU parameter settings	• The device assignment method of the motion CPU can be performed with Q series compatible assignment and MELSEC iQ-R Motion device assignment. • The system parameters can be diverted from a GX Works3 project file. • The basic settings (emergency stop input settings) and servo network settings are set in the motion CPU common parameters. • The parameters specific to each axis (such as machine specifications) are set in the motion control parameter.

Chapter 3 Motion CPU Module Programming

In this chapter, you will learn how to program the motion controller using a motion SFC program.

3.1 Devices

Motion CPU modules have devices such as inputs (X), outputs (Y), internal relays (M), link relays (B), annunciators (F), data registers (D), and link registers (W) just like PLC CPU modules.
In addition, motion CPU modules have their own special motion registers (#).

Some of the internal relays (M) and data registers (D and #) among the devices are assigned as the positioning dedicated signal.
The positioning dedicated signal can be assigned (device assignment method) with "MELSEC iQ-R Motion device assignment" and "Q series Motion compatible device assignment".
For the Q series motion compatible assignment method, the numbers and the motion CPU modules of the Q series are available, but the device numbers of up to axis 32 and after axis 33 are not consecutive.

It is recommended to assign the devices depending on the cases as follows:
Q series motion compatible assignment method: When diverting a program from MELSEC Q series motion CPU module
MELSEC iQ-R Motion device assignment method: When starting up a new system

The MELSEC iQ-R Motion device assignment is used in this course.

(Example) Assigning a device for each axis status

Assignment method	Axis 1	Axis 2	...	Axis 32	Axis 33	...
MELSEC iQ-R Motion device assignment	M32400 to M32431	M32432 to M32463	...	M33392 to M33423	M33424 to M33455	...
Q series motion compatible assignment	M2400 to M2419	M2420 to M2439	...	M3020 to M3039	M33424 to M33455	...

Same numbers as Q series motion

Both are the same from axis 33

For the details of the device numbers assigned to positioning dedicated signals, refer to the following manual.

Programming Manual (Positioning Control)
Chapter 2 POSITIONING DEDICATED SIGNALS

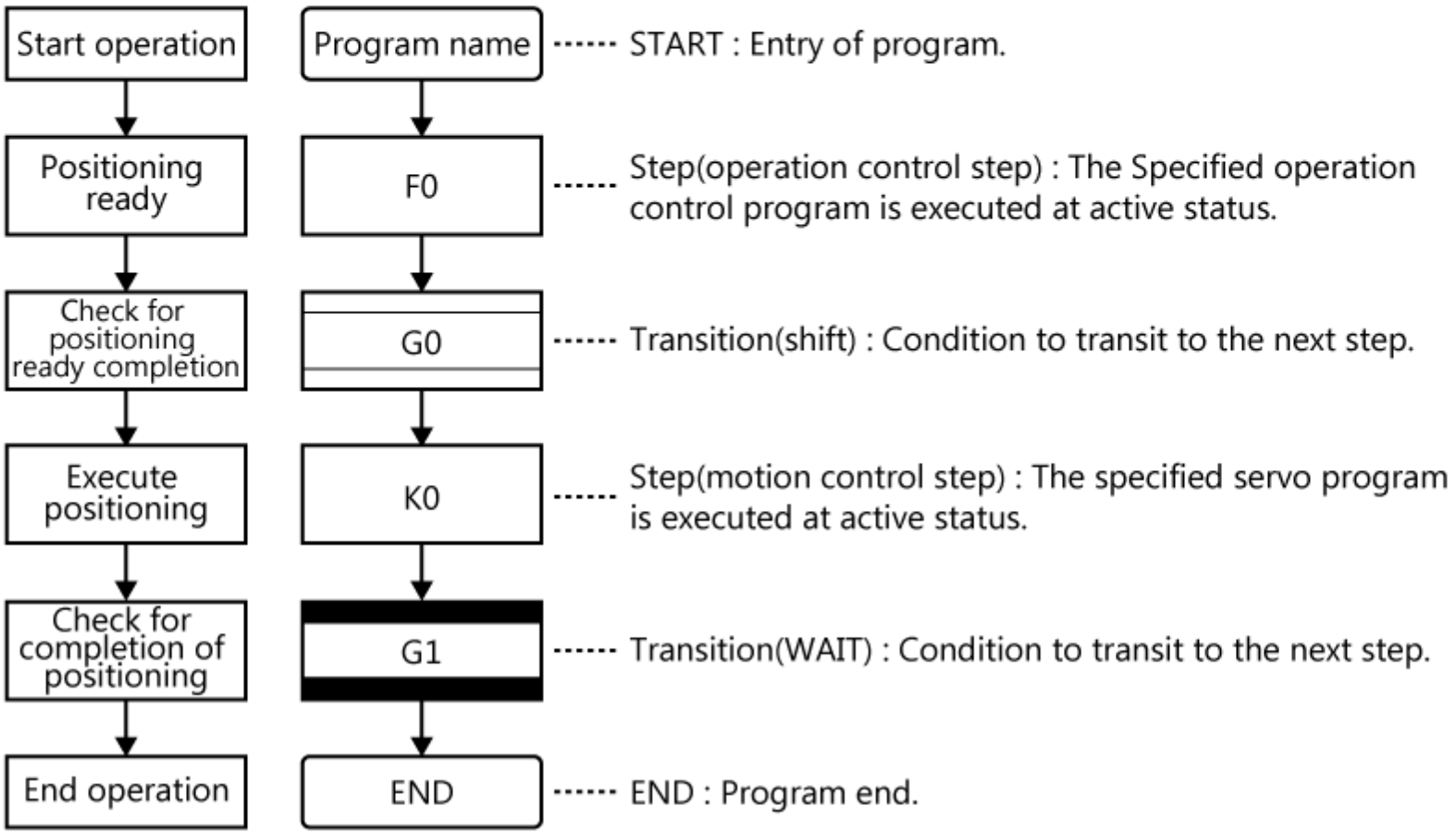
If the motion CPU module setting and MT Developer2 setting for the device assignment method are different, the communication cannot be performed.
In this case, select [Online] => [Change Device Assignment Method] from the toolbar of MT Developer2 to change the setting of the motion CPU module.

3.2 Motion SFC Program

In this section, you will learn about the meaning of the symbols in the chart of motion SFC program.

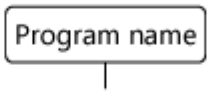
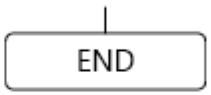
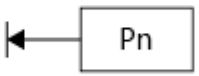
3.2.1 Motion SFC Program Configuration

A motion SFC program is created with the description similar to a flowchart.
As shown below, the basic description method is configured from a combination of elements, such as START, step, transition, and END.



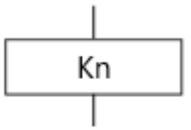
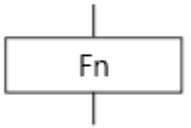
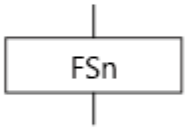
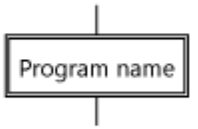
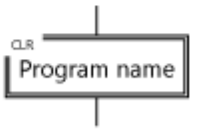
3.2.2 Motion SFC Program Symbols

(1) Basic elements

Name	Symbol	Description
START (Start program)		Indicates the entry point of the program with the program name. Limited to one element per program.
END (End program)		Indicates the end of the program. This can be placed multiple times in one program. It is not required to be placed.
Jump		Jumps to the specified pointer within its own program.
Pointer		Indicates the pointer of the jump destination.

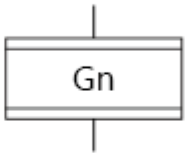
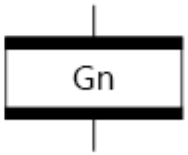
3.2.2 Motion SFC Program Symbols

(2) Steps

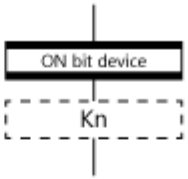
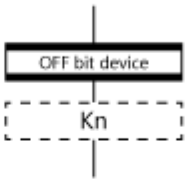
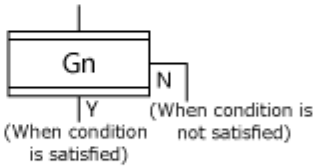
Name	Symbol	Description
Motion control step		Starts the specified servo program Kn. (Refer to section 3.4 for details.)
Once execution type operation control step		Executes the operation control program once.
Scan execution type operation control step		Executes an operation control program repeatedly until the next transition condition is satisfied.
Subroutine call/start step		Calls or starts the motion SFC program with the specified program name. The behavior changes depending on the subsequent transition is WAIT or not. (Refer to section 3.2.5 for details.)
Clear step		Stops the specified program being executed and ends the processing.

3.2.2 Motion SFC Program Symbols

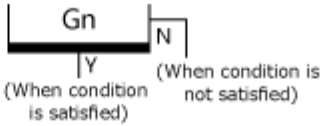
(3) Transitions

Name	Symbol	Description
Shift (Pre-read transition)		• If the processing right before is the motion control step, the processing shifts to the next step when the condition is satisfied without waiting for the motion operation to complete. • If the processing right before is the operation control step, the processing shifts to the next step when the condition is satisfied after the operation execution is completed. • If the processing right before is the sub routine call/start step, the processing shifts to the next step when the transition condition is satisfied without waiting for the subroutine operation to complete.
WAIT		• If the processing right before is the motion control step, the processing waits for the motion operation to complete and then shifts to the next step when the condition is satisfied. • If the processing right before is the operation control step, the processing shifts to the next step when the condition is satisfied after the operation execution is completed. (The operation is the same as shift.) • If the processing right before is the sub routine call/start step, the processing waits for the subroutine operation to complete and shifts to the next step when the transition condition is satisfied.
		Prepares to start the next motion control step and immediately output a

3.2.2 Motion SFC Program Symbols

WAITON		Prepares to start the next motion control step and immediately output a command if the specified bit device turns on.
WAITOFF		Prepares to start the next motion control step and immediately output a command if the specified bit device turns off.
Shift Y/N		• If the processing right before is the motion control step, the processing shifts to the step below when the condition is satisfied, and shifts to the step on the right when the condition is not satisfied without waiting for the motion to complete. • If the processing right before is the operation control step, the processing shifts to the step below after the operation execution is completed. The processing shifts to the step on the right when the condition is not satisfied. • If the processing right before is the sub routine call/start step, the processing transitions to the next step when the transition condition is satisfied, and shifts to the step connected from the right when the condition is not satisfied without waiting for the subroutine operation to complete.
		• If the processing right before is the motion control step, the processing waits for the motion to complete and shifts to the step below when the condition is satisfied, and shifts to the step on the right when the condition is not satisfied. • If the processing right before is the operation control step, the processing

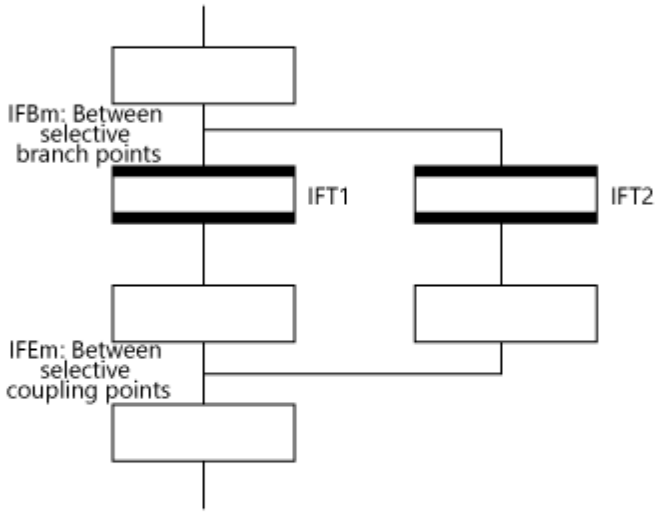
3.2.2 Motion SFC Program Symbols

WAIT Y/N		shifts to the step below after the operation execution is completed. The processing shifts to the step on the right when the condition is not satisfied. (The operation is the same as shift Y/N.) • If the processing right before is the sub routine call/start step, the processing waits for the subroutine operation to complete and shifts to the next step when the transition condition is satisfied, and shifts to the step connected from the right when the condition is not satisfied.

3.2.3 Branches and Couplings

This section explains the patterns of branch and coupling.

(1) Selective branches and couplings

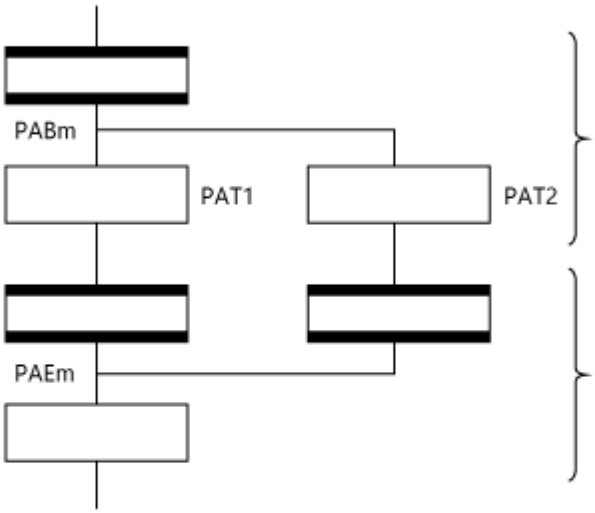


Selective branch
After the processing right before the branch is executed, the route whose condition is satisfied the first is executed.
All the start of selective branches is required to be shift transitions or WAIT transitions.
A mixture of transitions will result in a parallel branch.

Selective coupling
A selective coupling connects the route from the selective branch to single route.
The element before and after the coupling point can be either step or transition.

3.2.3 Branches and Couplings

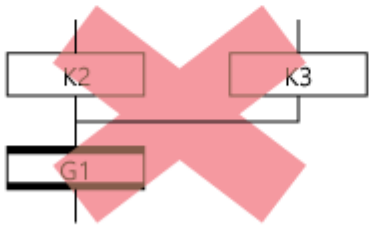
(2) Parallel branches and couplings



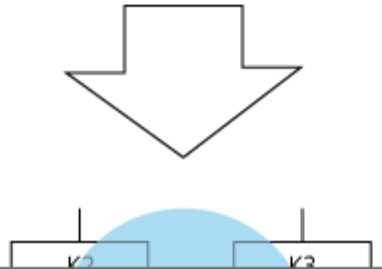
Parallel branch
After the processing right before a branch is executed,
all of the processing connected in parallel are executed simultaneously.
The start of a parallel branch may be either step or transition.
However, WAITON and WAITOFF cannot be set for the start.

Parallel coupling
A parallel coupling connects the route from the selective branch to single route.
The element before and after the coupling point can be either step or transition.

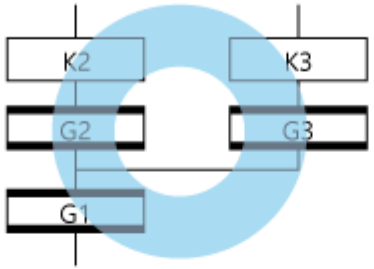
[CAUTION]



In the case of a coupling such as the figure on the left,
the completion of stopping the axes that started at K2 and
K3 will not be the condition for shifting to G1.

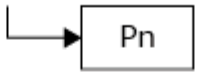
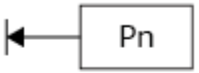


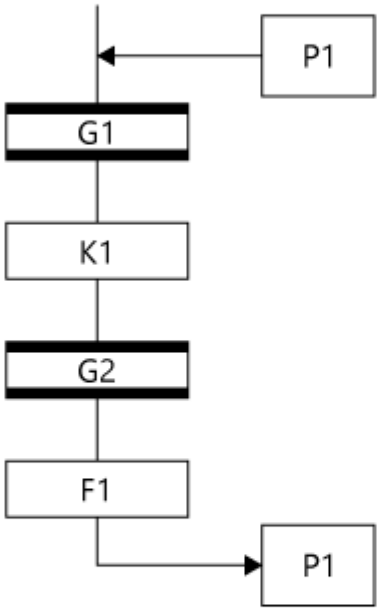
3.2.3 Branches and Couplings



To shift to G1 after the completion of stopping the axes that started at K2 and K3, set a WAIT transition for K2 and K3.

3.2.4 Jumps and Pointers

This section explains jumps () and pointers ().



- Set the jump for jumping to the specified pointer Pn in own program.
- Pointers can be set at steps, transitions, branch points, and coupling points.
- Up to 16384 (P0 to P16383) points of pointers can be set in one program.

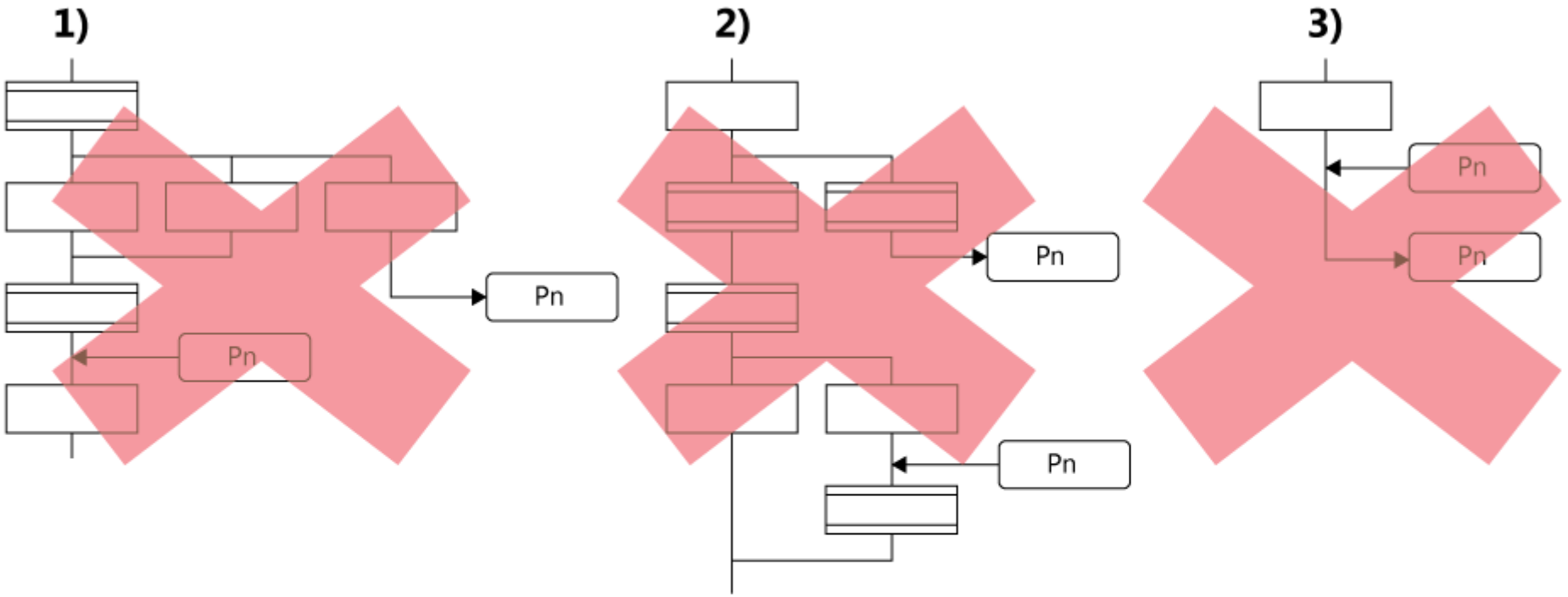
In the case of the figure on the left,
the processing loops as G1 => K1 => G2 => F1 => G1 => K1 => ... .

[CAUTION]

1) A jump to exit from the parallel branch - parallel coupling cannot be set.
2) A jump to enter inside the parallel branch - parallel coupling from the outside cannot be set.
3) The pointers and jumps that are processed consecutively cannot be set.

1) **2)** **3)**

3.2.4 Jumps and Pointers



3.2.5 Subroutine Call/Start Step

The control differs depending on the type of transition that is executed after the subroutine (Program name) call/start step.

(1)When WAIT transition is executed next: Call subroutine

As shown in Figure A below, when the subroutine call step is executed, the control shifts to the specified program, and when the called program executes END, the control is returned to the call source program.

(2)When the transition other than WAIT transition is executed next: Start subroutine

As shown in Figure B below, when the subroutine call step is executed, the specified program starts and continues to control the call source program. Two programs are executed simultaneously.

Figure A Subroutine call

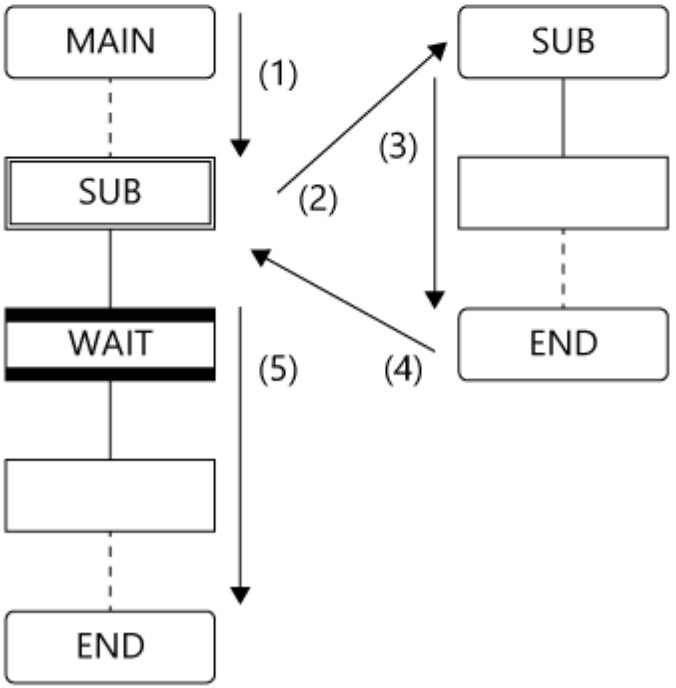
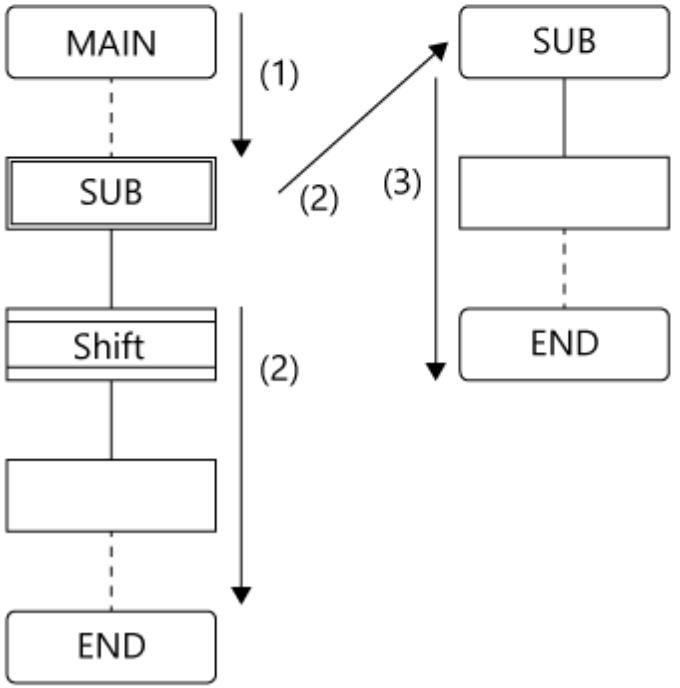


Figure B Subroutine start



3.3 Program Creation Method

In this section, you will learn how to create the programs that is described in the steps and transitions of motion SFC.

3.3.1 Devices and Constants

(1) Descriptions of bit devices

Device name	Device description
Input relay	Xn
Output relay	Yn
Internal relay	Mn
Link relay	Bn
Annunciator	Fn
Data register	Dn.m *1
Link register	Wn.m *1
Motion register	#n.m *1
Special relay	SMn
Special register	SDn.m *1
CPU buffer memory access device	U3E□¥Gn.m *1
CPU buffer memory access device (fixed scan communication area)	U3E□¥HGn.m *1
Module access device	U□¥Gn.m

*1 "m" indicates bit specification (bit number: 0 to F) of a word device.

3.3.1 Devices and Constants

(2) Descriptions of word devices

Device name	Device description		
	16-bit integer type	32-bit integer type (n is an even number)	64-bit floating point type (n is an even number)
Data register	Dn	DnL	DnF
Link register	Wn	WnL	Wn:F
Motion register	#n	#nL	#nF
Special register	SDn	SDnL	SDnF
CPU buffer memory access device	U3E□¥Gn	U3E□¥GnL	U3E□¥GnF
CPU buffer memory access device (fixed scan communication area)	U3E□¥HGn	U3E□¥HGnL	U3E□¥HGnF
Module access device	U□¥Gn	U□¥GnL	U□¥GnF

3.3.2 Operators and Functions

The priority of operator and function is as follows.
The operation sequence can be specified freely by using parentheses.

Priority		Item (Operator and function)
High ↑	1	Calculation in parentheses ((...))
	2	Standard function (SIN, COS, etc.), type conversion (USHORT, LONG, etc.)
	3	Bit inversion (~), logical negation (!), sign inversion (-)
	4	Multiplication (*), division (/), remainder (%)
	5	Addition (+), subtraction (-)
	6	Bit left shift (<<), bit right shift (>>)
	7	Comparison operators: Less than (<), less than or equal to (<=), more than (>), more than or equal to (>=)
	8	Comparison operators: Equal (==), unequal (!=)
	9	Bit logical AND (&)
↓ Low	10	Bit exclusive OR (^)
	11	Bit logical OR (|)
	12	Logical AND (*)

3.3.2 Operators and Functions

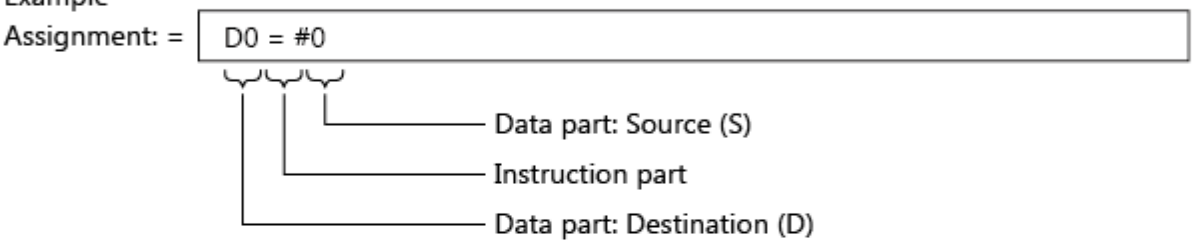
	13	Logical OR (+)
	14	Assignment (=)

3.3.3 Structure of Instruction

Many of the instructions that can be used in the operation control programs can be divided into instruction and data parts.
The instruction and data parts are used for the following purposes.

- Instruction part: Indicates the function of that instruction.
- Data part: Indicates the data used in the instruction.

Example



■Source (S)

- The source is the data used in the operation.
- The source varies as shown in the table below depending on the device specified in each command.

Devices	Description
Bit device, word device	Specify the device which stores the data used in the operation. The data is required to be stored in the specified device before the operation is executed. The data used in the command can be changed by changing the data stored in the specified device during the program execution.
Constant	Specify the numerical value used in the operation. Since the constant is set when creating the program, it cannot be changed during program execution.

■Destination (D)

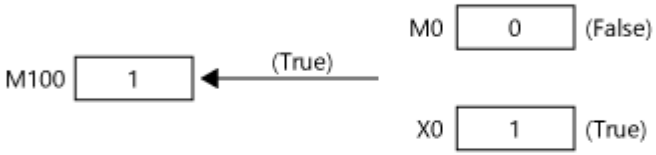
3.3.3 Structure of Instruction

- The data after the operation is stored as the destination data.
- Always set the device for storing the data in the destination data.

Program example

■ Program that sets M100 when either of M0 and X0 is ON (1)

```
SET M100 = M0 + X0
```



■ Program that sets M100 when #0 and D0 match

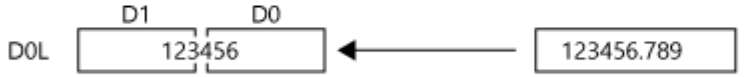
```
SET M100 = #0 == D0
```

■ Program that resets M100 when M0 is OFF (0)

```
RST M100 = !M0
```

■ Program that assigns K123456.789 to D0L

```
D0L = K123456.789
```



Assign by converting 64-bit floating point type to 32-bit integer type.

3.4 Servo Programs

This section explains about servo programs which consist of the servo motor rotation speed, target position address, and others.

3.4.1 Servo Program Configuration

One servo program consists of a program No., servo command, and positioning data.
When the program No. and target servo command are specified in MT Developer2, the positioning data required to execute the specified servo command can be set.

■Explanation of program

Program No.: Any number from 0 to 8191
(0 to 4095 if the OS software version is "09" or earlier)
can be specified as the number for specification
in the motion SFC program.

Servo command: Indicates the type of positioning control.

Positioning data: Data required to execute the servo command.

<K11>

ABS-3			Unit
Axis	1,	3000000.0	[μm]
Axis	2,	5500000.0	[μm]
Axis	3,	-2500000.0	[μm]
Vector speed		40000.00	[mm/min]

Servo program data	Setting
K11	Program No.
ABS-3	Servo command
Axis 1, 3000000.0	Axis to use
	Positioning address
Axis 2, 5500000.0	Axis to use
	Positioning address
Axis 3, -2500000.0	Axis to use
	Positioning address
Vector speed	Command speed of three axes (axis 1, axis 2, and axis 3) combined
Dwell	Dwell time
M code	M code
P.B.	Parameter block No.

For each servo command, there is the data required for execution.
For example, the data shown in the following table is

3.4.1 Servo Program Configuration

Dwell	2500	[ms]
M code	12	
P.B.	3	

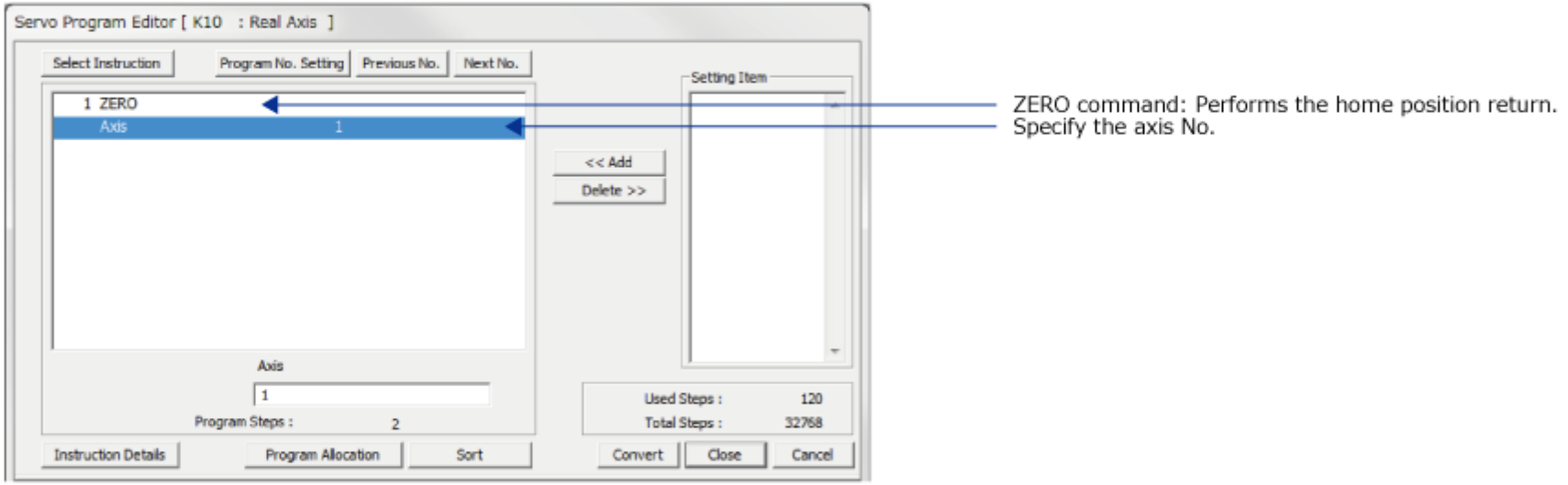
arequired for the ABS-3 command.

Setting condition	Setting item
Always set	・Axis to use and positioning address ・Instruction speed
Set as necessary	・Dwell time ・M code ・P.B. (parameter block) If this item is not set, control is performed with the initial value (parameter block 1).

3.4.2 Home Position Return

Use the ZERO command of the servo program to execute the home position return. Set the home position return method in [Motion Control Parameter] => [Axis Setting Parameter] => [Home Position Return Data]. For details of the home position return data, refer to section 2.4.4.

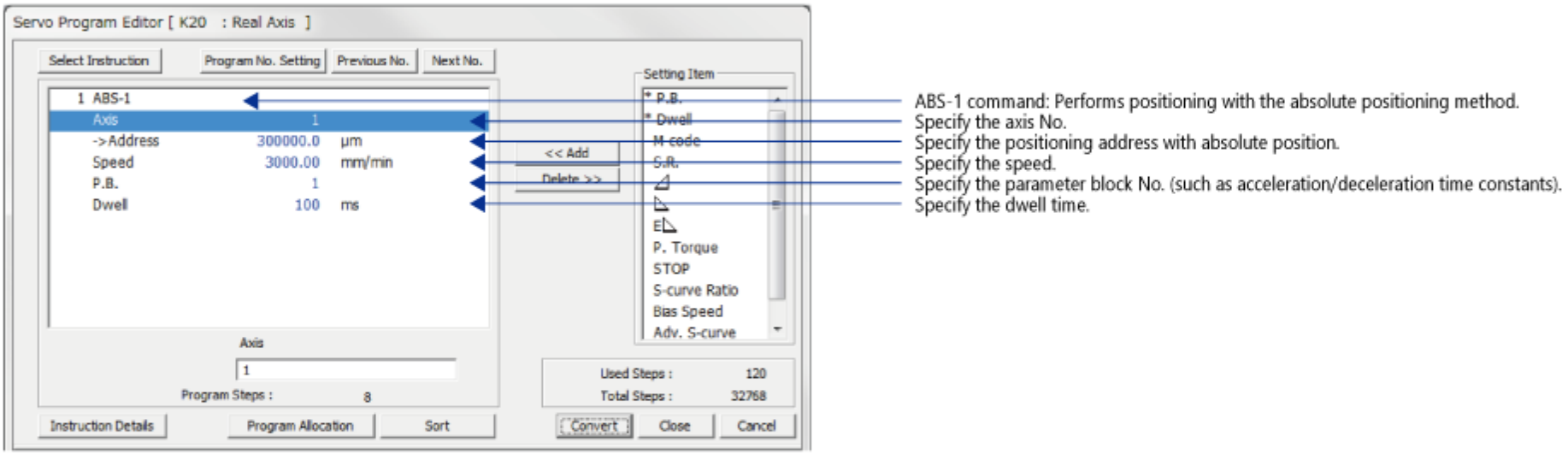
Example of setting ZERO command



3.4.3 Positioning of 1 Axis

Use the ABS-1 command of the servo program or INC-1 command to execute the positioning operation for 1 axis.
The home position return is required to be performed before the positioning.

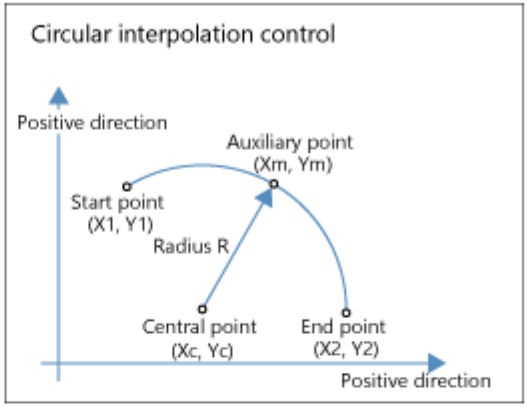
Example of setting ABS-1 command



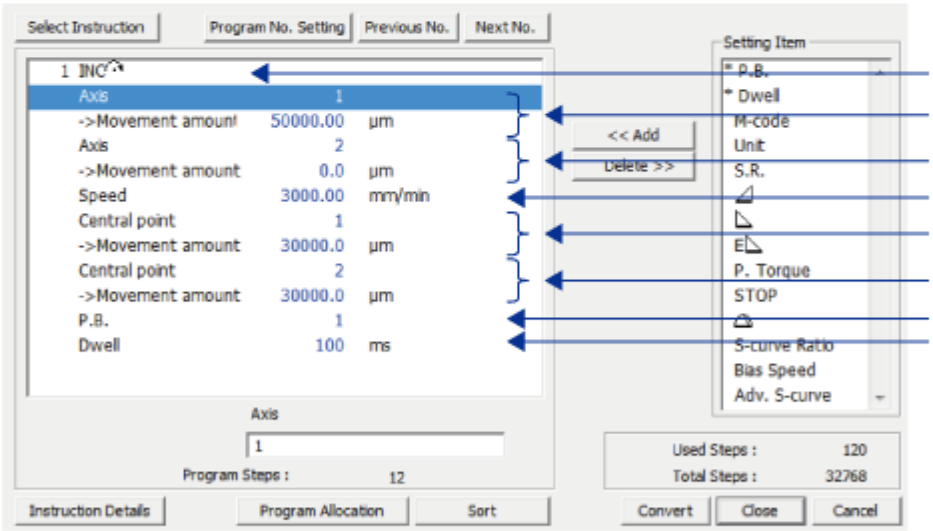
(Note) Select P.B. (parameter block) and dwell in [Setting Item] on the right side and click the [<<Add] button to add them to the servo program on the left side.

3.4.4 Interpolation Control

In the interpolation control, two to four axes to be used are specified to perform the positioning while tracing a linear or circular trajectory.
The home position return is required to be performed before the positioning.
In the case of circular interpolation, select a method from auxiliary point specification, radius specification, and central point specification.
Refer to the figure on the right for the concept of points in circular interpolation.



Example of setting INC ↷ command

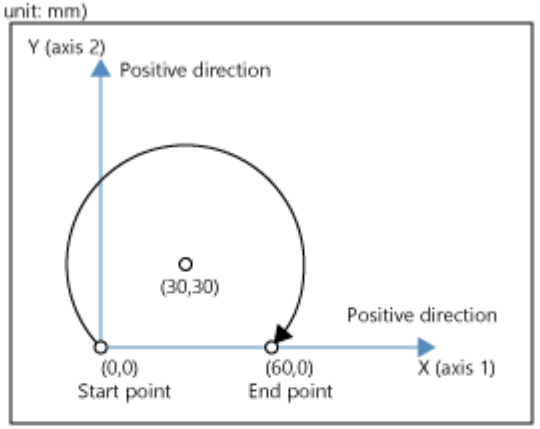


INC↷: Central point-specified circular interpolation control, incremental method, clockwise
Specify the axis No. of the X-axis and the X-coordinate of the end point.
Specify the axis No. of the Y-axis and the Y-coordinate of the end point.
Specify the vector speed.
Specify the X-coordinate of the central point.
Specify the Y-coordinate of the central point.
Specify the parameter block No.
Specify the dwell time.

The trajectory shown in the figure on the right is traced in this program.
(Movement amount unit: mm)

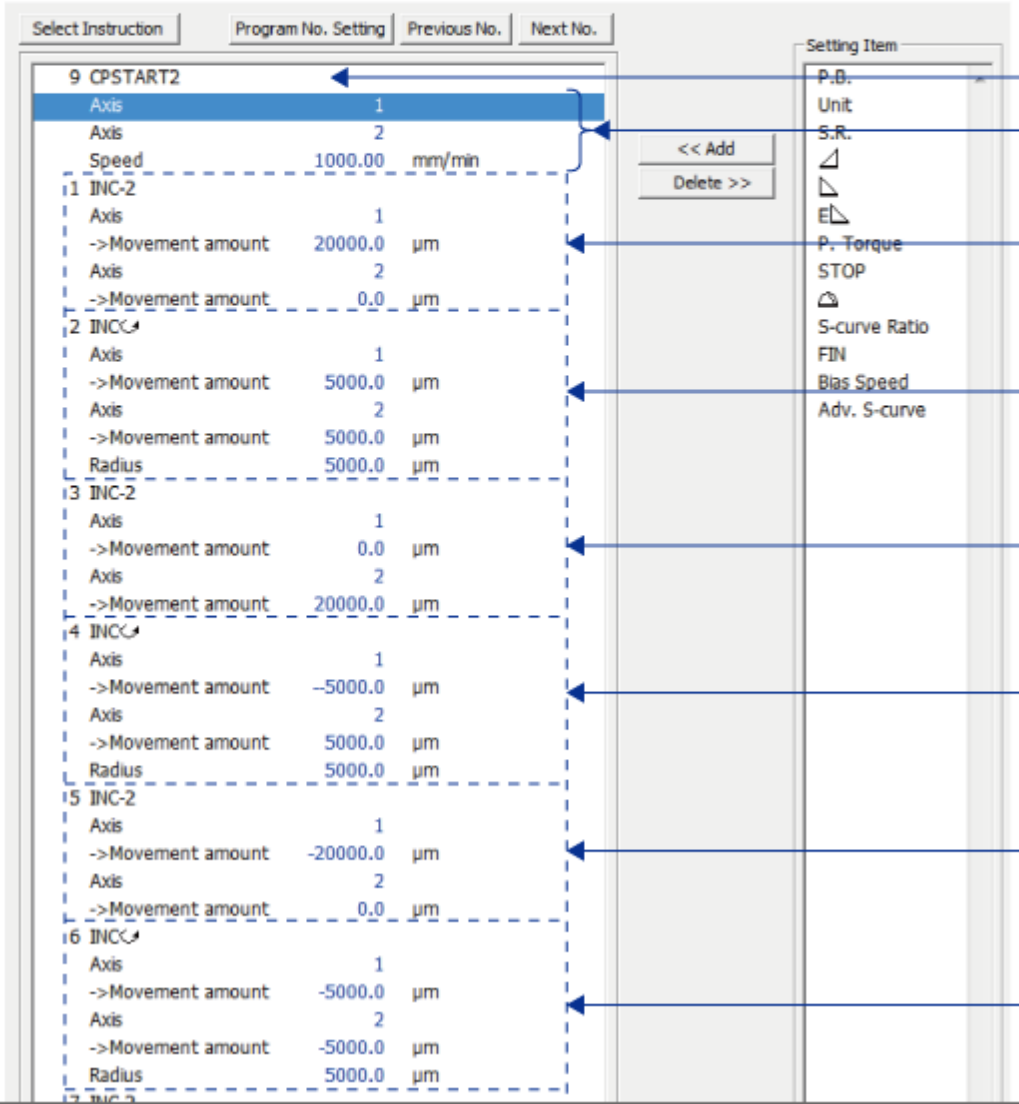
3.4.4 Interpolation Control

(Movement amount unit: mm)



3.4.5 Continuous Trajectory Control

In the continuous trajectory control, the positioning is performed continuously to a preset passing point with one start.
In addition, the control can be performed repeatedly between arbitrary points by using the command repeatedly.
The M code and torque limit values can be changed for each passing point.



CPSTART2: Continuous trajectory control using two axes

Two axes to be used and vector speed

First point
Control method: 2-axis linear control, incremental method
Set the movement amount of each axis.

Second point
Control method: 2-axis circular interpolation control, incremental method, radius specified counterclockwise rotation
Set the movement amount of each axis and the radius of circular interpolation.

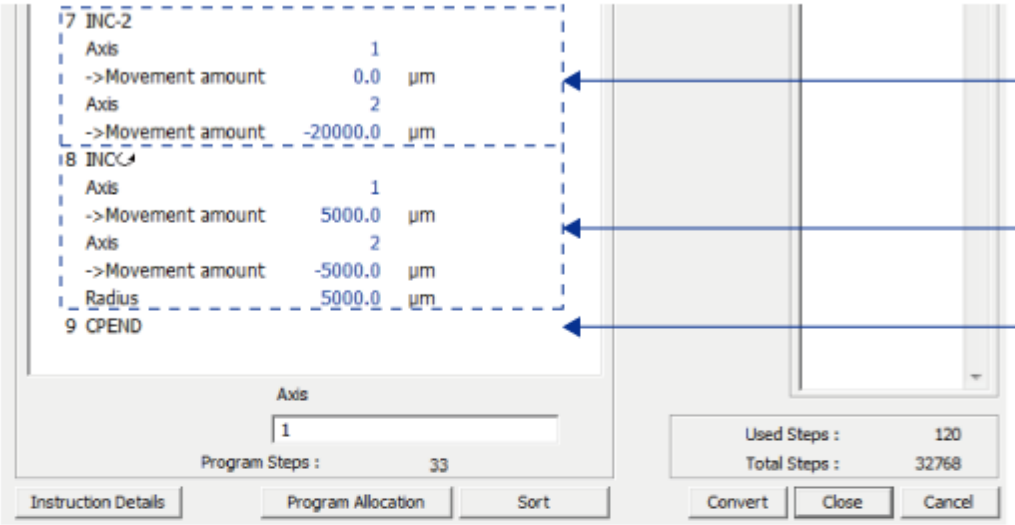
Third point
Control method: 2-axis linear control, incremental method
Set the movement amount of each axis.

Fourth point
Control method: 2-axis circular interpolation control, incremental method, radius specified counterclockwise rotation
Set the movement amount of each axis and the radius of circular interpolation.

Fifth point
Control method: 2-axis linear control, incremental method
Set the movement amount of each axis.

Sixth point
Control method: 2-axis circular interpolation control, incremental method, radius specified
Set the movement amount of each axis and the radius of circular interpolation.

3.4.5 Continuous Trajectory Control



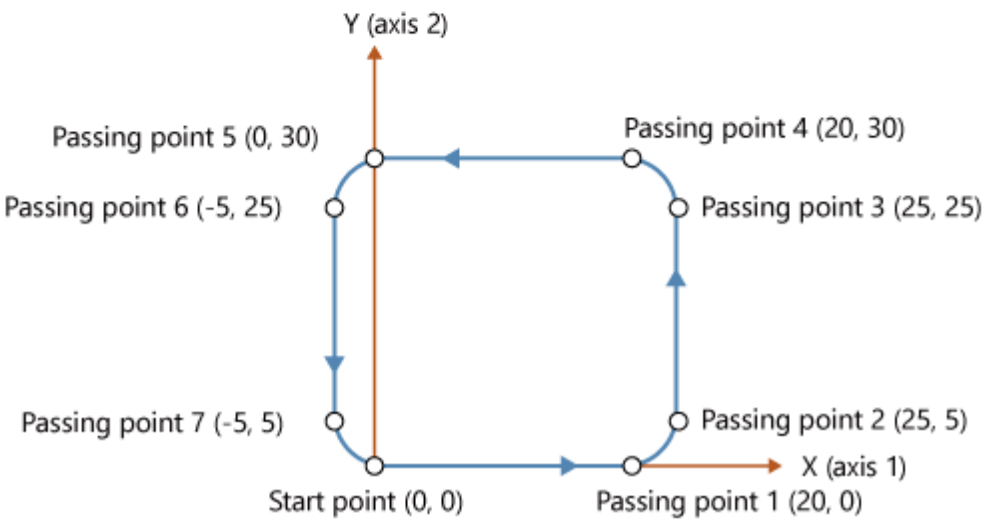
Seventh point
Control method: 2-axis linear control, incremental method
Set the movement amount of each axis.

Eighth point
Control method: 2-axis circular interpolation control, incremental method, radius specified
Set the movement amount of each axis and the radius of circular interpolation.

Always end with CPEND.

The trajectory shown in the figure below is traced in this program.

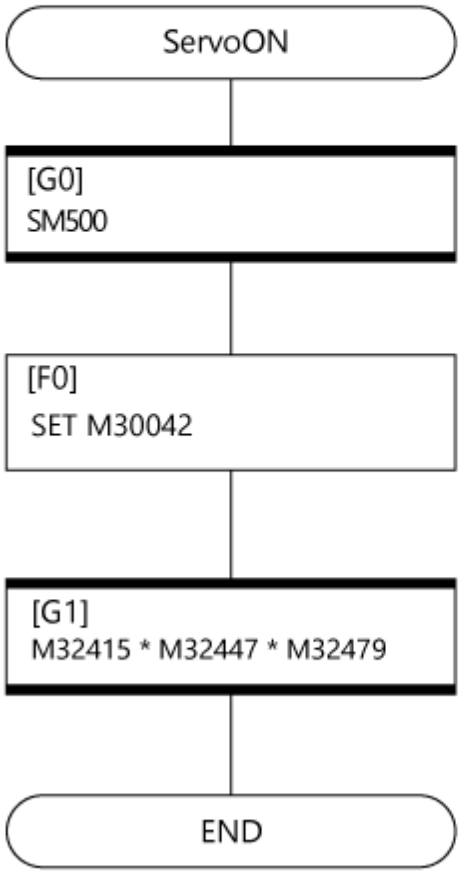
(Movement amount unit: mm)



3.5 MT Developer2 Operation

(1) How to create a motion SFC program

Videos are used in this section to explain how to create an SFC program in MT Developer2.
As the figure shown below, a program to turn on the servo of all axes is created as an example.



Waits until the motion CPU module starts up normally.
SM500: PCPU READY complete

Executes the all axes servo ON command.
M30042: [Rq.1123] All axes servo ON command

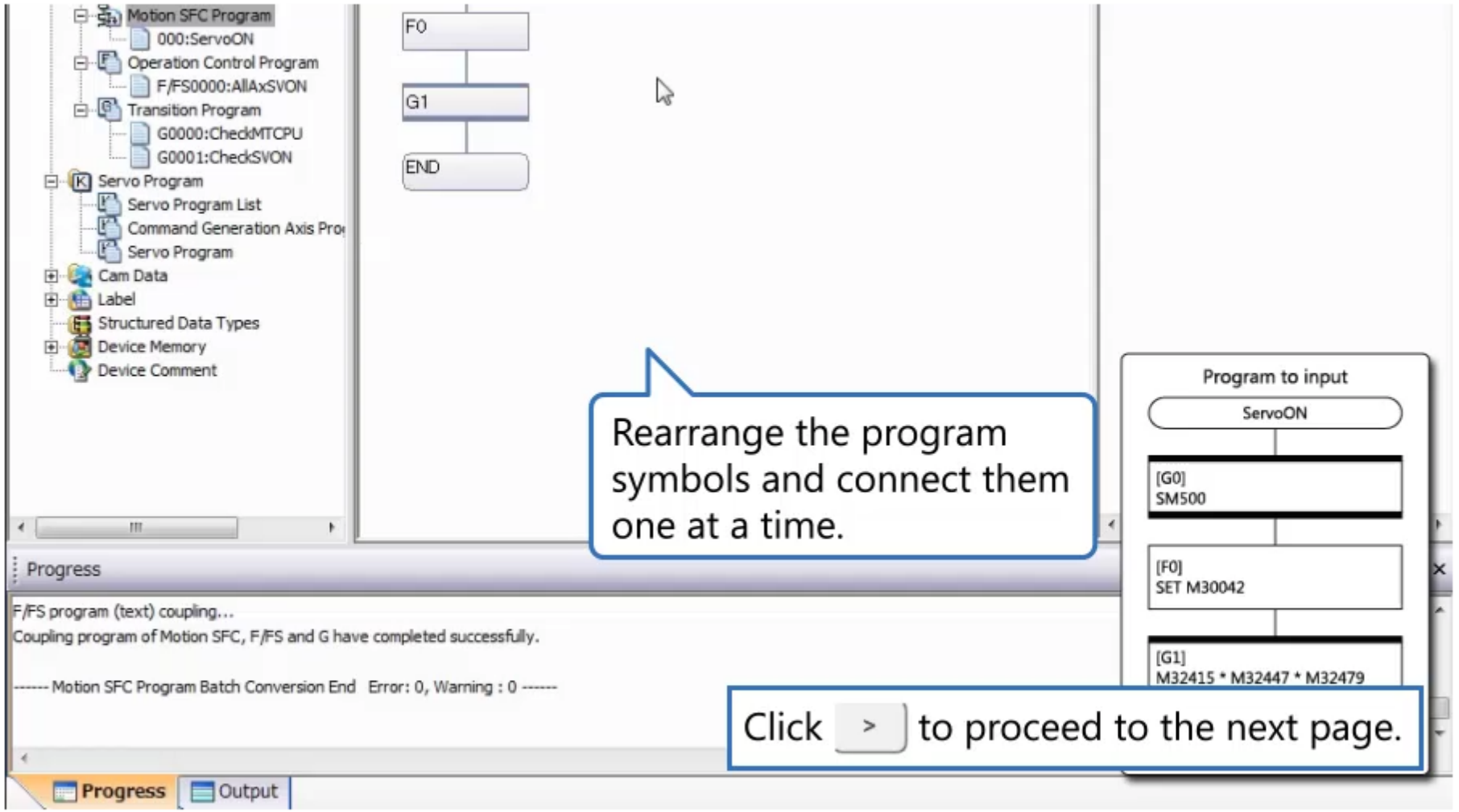
Waits until the servo amplifiers of axis 1 to axis 3 enters the servo ready state.
M32415+32n: [St.1075] Servo ready

3.5 MT Developer2 Operation

(Note) In the sample program, this program is registered in motion SFC program No. 200.
200 is added to each No. of operation control program and transition program.

3.5 MT Developer2 Operation

Previous

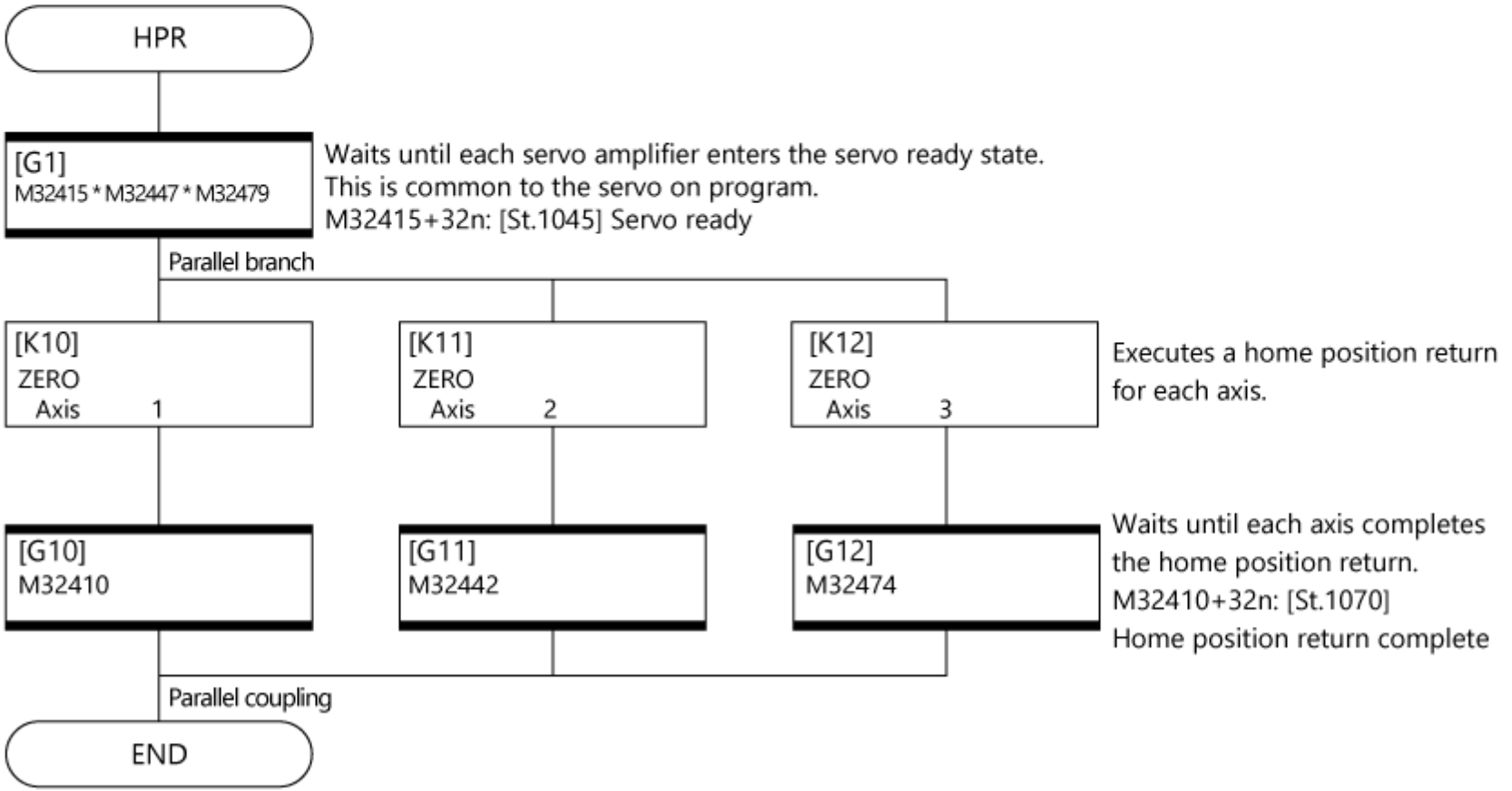


Click > to proceed to the next page.

3.5 MT Developer2 Operation

(2) How to create branches and couplings

Videos are used in this section to explain the operation when there are branches and servo programs.
As the figure shown below, a program to execute all axis home position return is created as an example.
This program is executed after all the axes of servos are turned on.

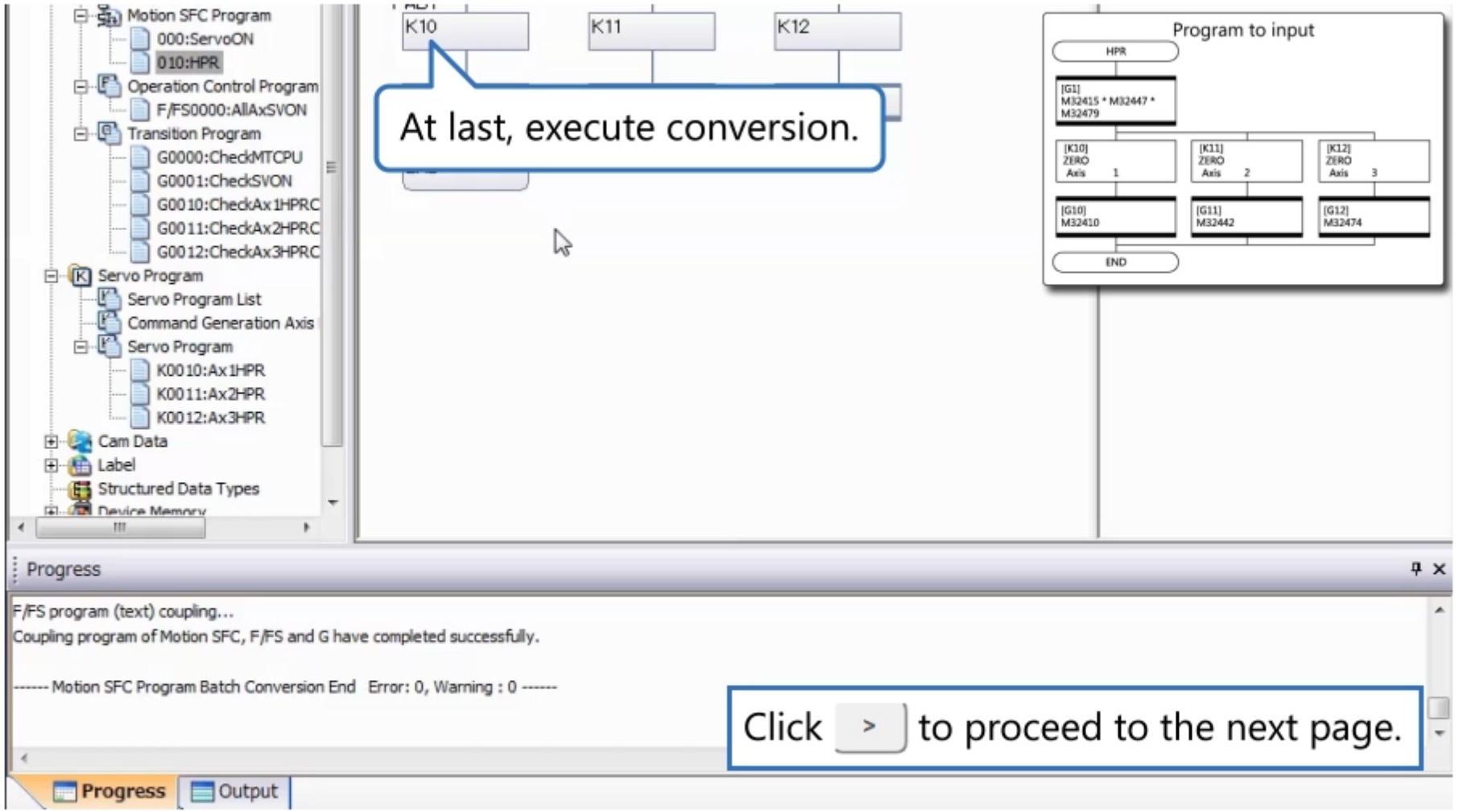


3.5 MT Developer2 Operation

(Note) In the sample program, this program is registered in motion SFC program No. 201.
200 is added to each No. of operation control program, transition program, and servo program.

3.5 MT Developer2 Operation

Previous

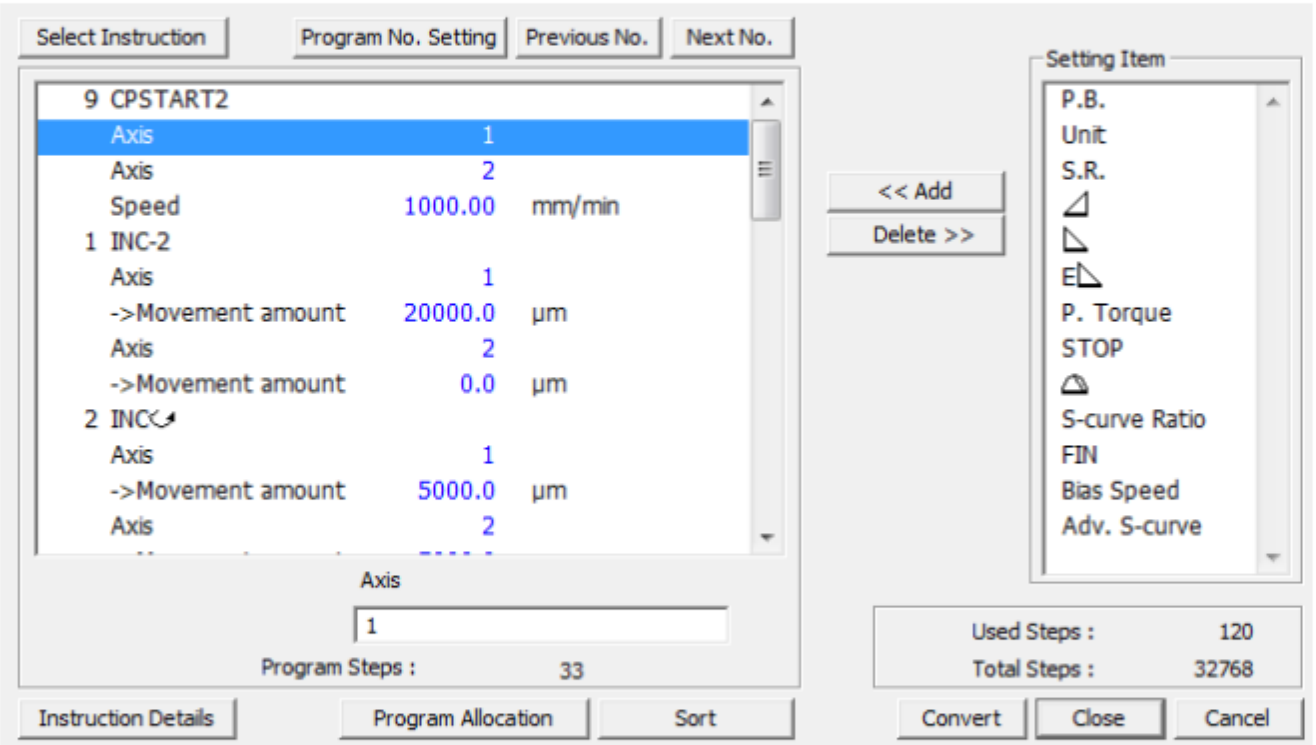


Click > to proceed to the next page.

3.5 MT Developer2 Operation

(3) How to create a servo program

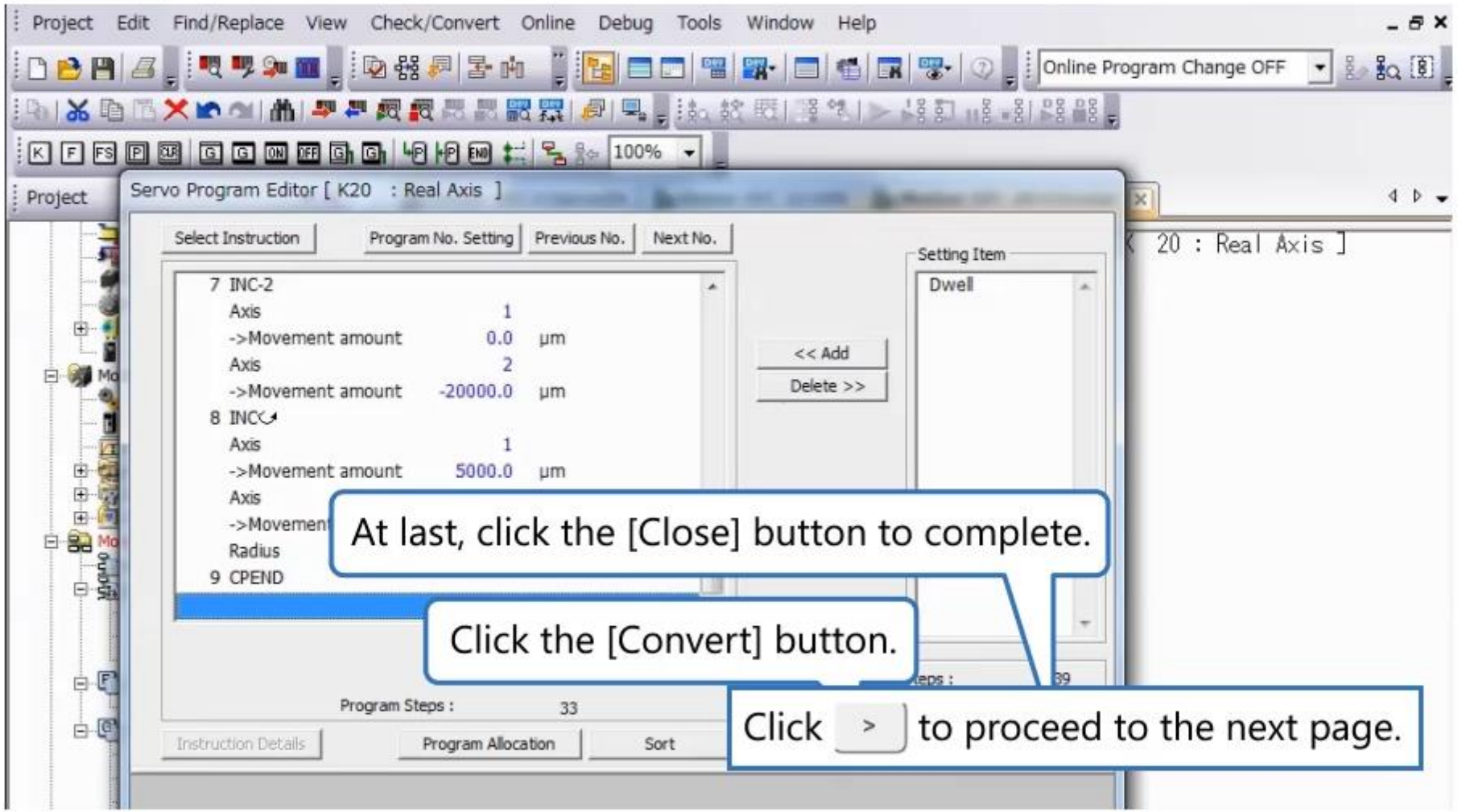
As an example of how to create a servo program, videos are used in this section to explain the describing method of the continuous trajectory control in section 3.4.5.



(Note) In the sample program, this program is registered in servo program No. 220.

3.5 MT Developer2 Operation

Previous

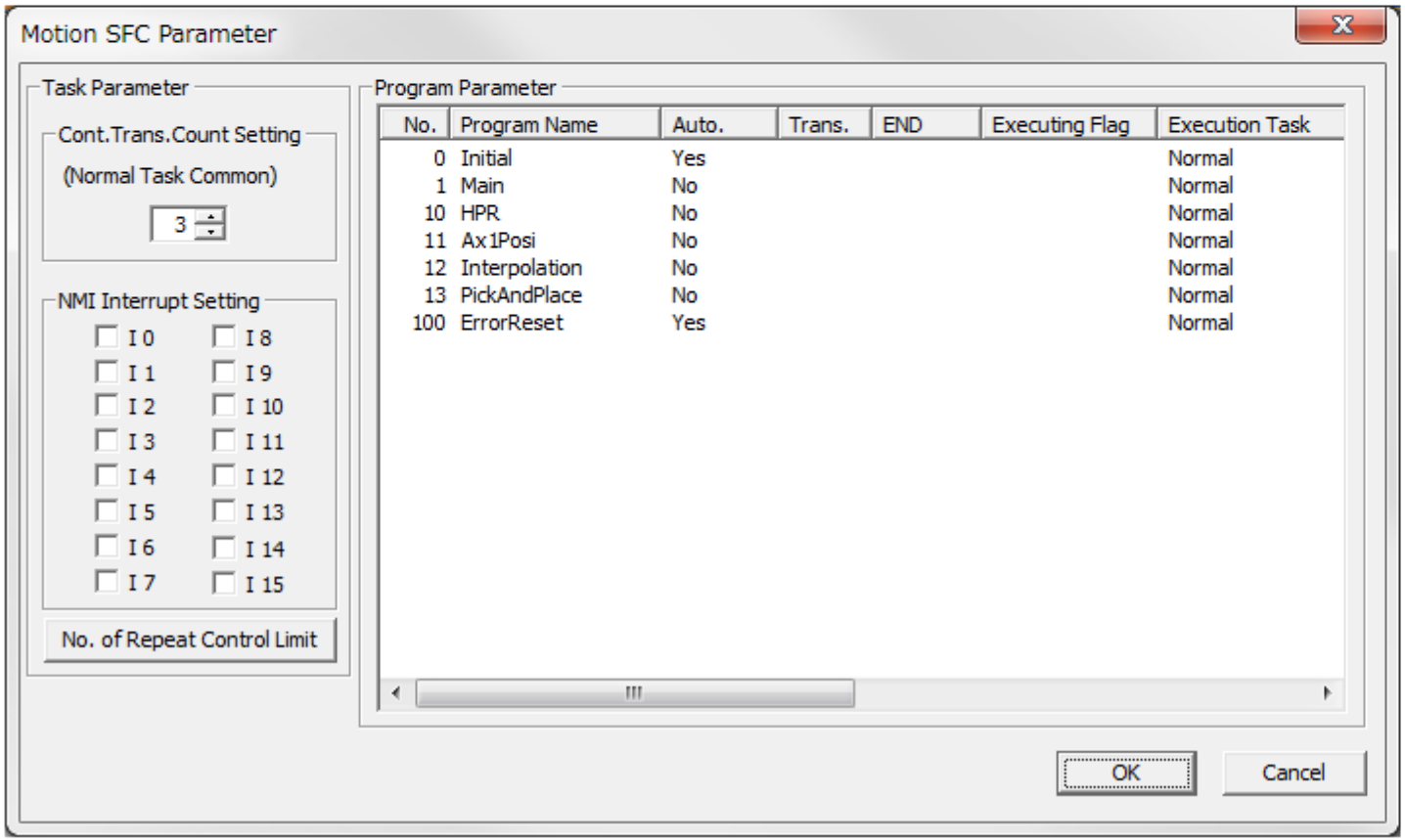


3.6 Motion SFC Parameters

The parameters related to a motion SFC program is set in the motion SFC parameter.
The motion SFC program can be executed automatically after the programmable controller enters the ready state by setting [Automatic Start] in the start settings.

For the details of other items, refer to the following manual.

Programming Manual (Program Design)
Chapter 6 MOTION SFC OPERATIONS AND PARAMETERS
6.9 Program Parameters



3.7 Summary of This Chapter

In this chapter, you have learned:

- Devices
- Motion SFC Program
- Program Creation Method
- Servo Programs
- MT Developer2 Operation
- Motion SFC Parameters

Points

Devices	• When the device assignment method is Q series motion compatible assignment method, the numbers up to axis 32 and numbers after axis 33 are not consecutive. • If the device assignment method registered to the motion CPU and the device assignment method of the project are different, the personal computer and motion CPU cannot be communicated.
Motion SFC Program	• A motion SFC program is created with the description similar to a flowchart. • The symbols used in a motion SFC program include start/end program, step, transition, jump, and pointer. • The selective branch, selective coupling, parallel branch, parallel coupling, and jump transition are available for the connection pattern.
Program Creation Method	• Learned the syntax of programs described in the step and transition.
Servo Programs	• The servo program consists of program No.s, servo commands, and positioning data. • Learned about the home position return command, 1 axis positioning command, interpolation control

3.7 Summary of This Chapter

	commands (linear interpolation and circular interpolation), and continuous trajectory control command.
MT Developer2 Operation	• Learned how to operate MT Developer2 in videos.
Motion SFC Parameters	• Automatic starting, task, type, and other settings can be configured in the motion SFC parameter.

Chapter 4 Operation Check of Sample Program

In this chapter, you will learn how to check the operation using the sample program.

4.1 Description of Sample Program

This section explains about the SFC program of the sample program.
Device assignment is as shown in the tables below.

• Input device

Device No.	Description	Device No.	Description
X10	Controller emergency stop	X13	2-axis interpolation control start
X11	All-axis home position return	X14	Continuous trajectory control start
X12	Axis-1 positioning start	X1F	Error reset

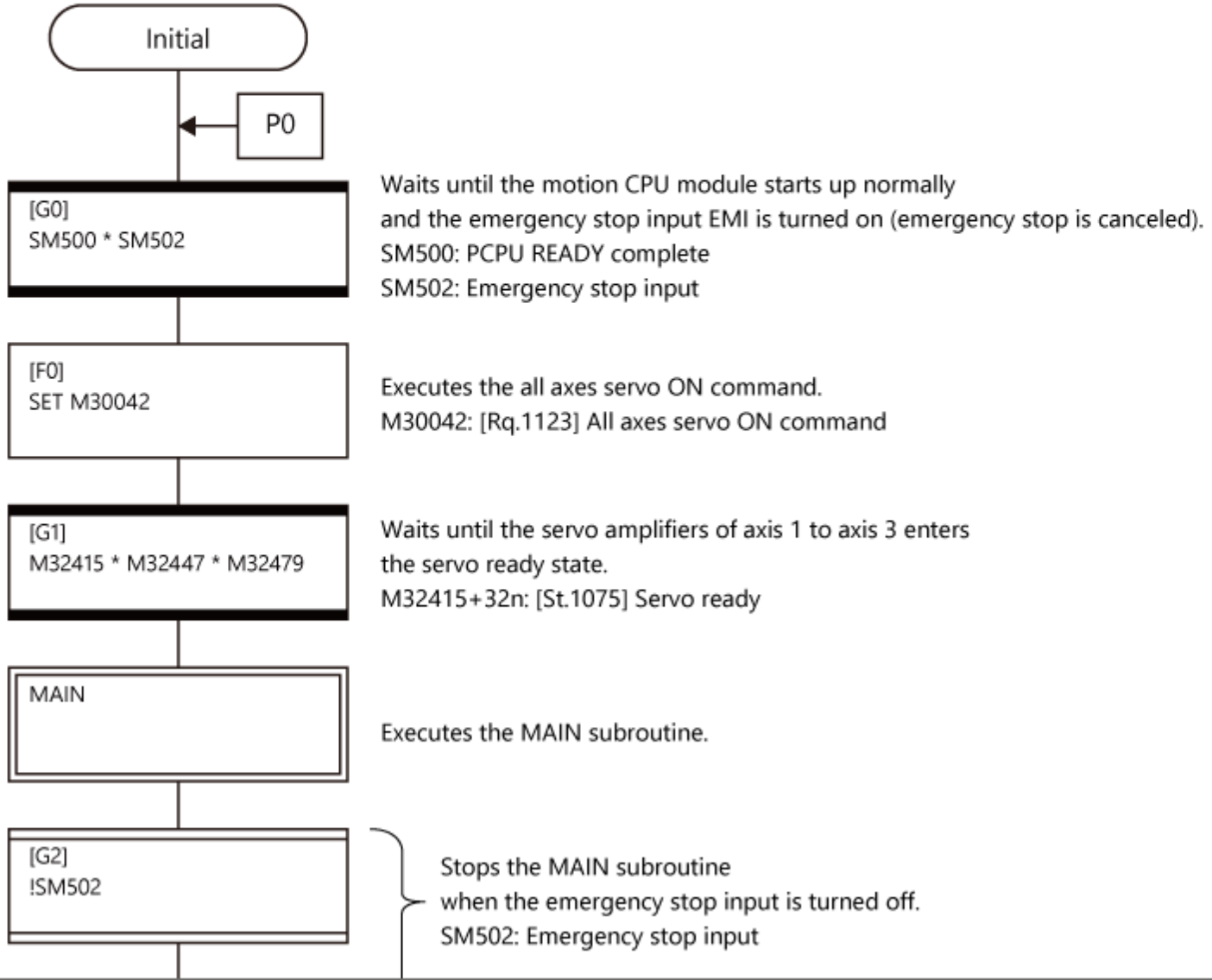
• Output device

Device No.	Description
Y00	Hand opening/closing command

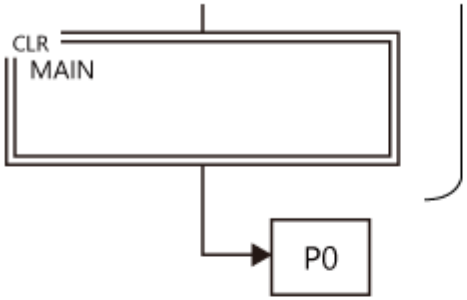
4.1 Description of Sample Program

(1)No. 000: Initial (automatic start)

Performs the initial settings when the motion CPU is started.



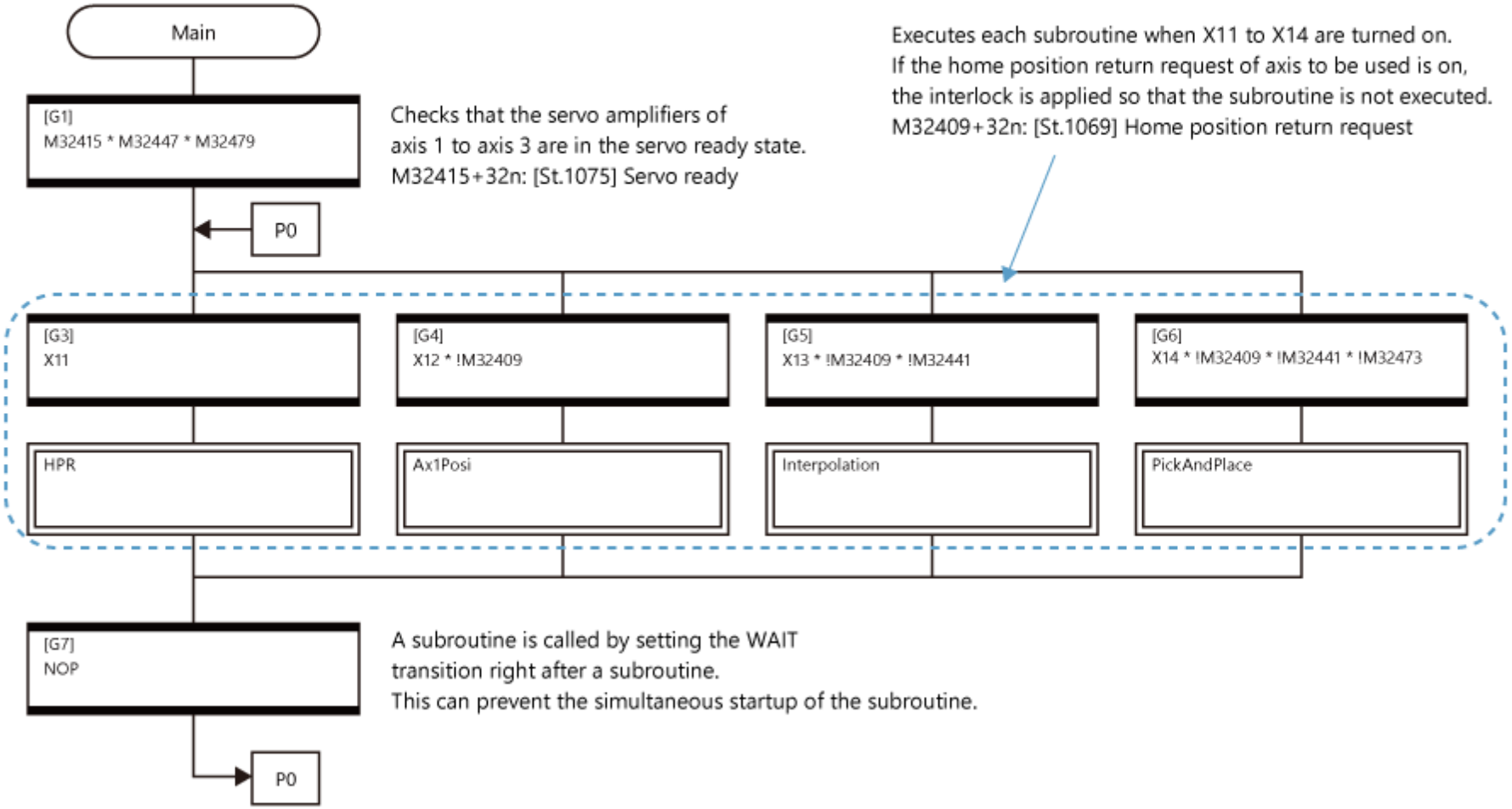
4.1 Description of Sample Program



4.1 Description of Sample Program

(2)No. 001: Main (no automatic start)

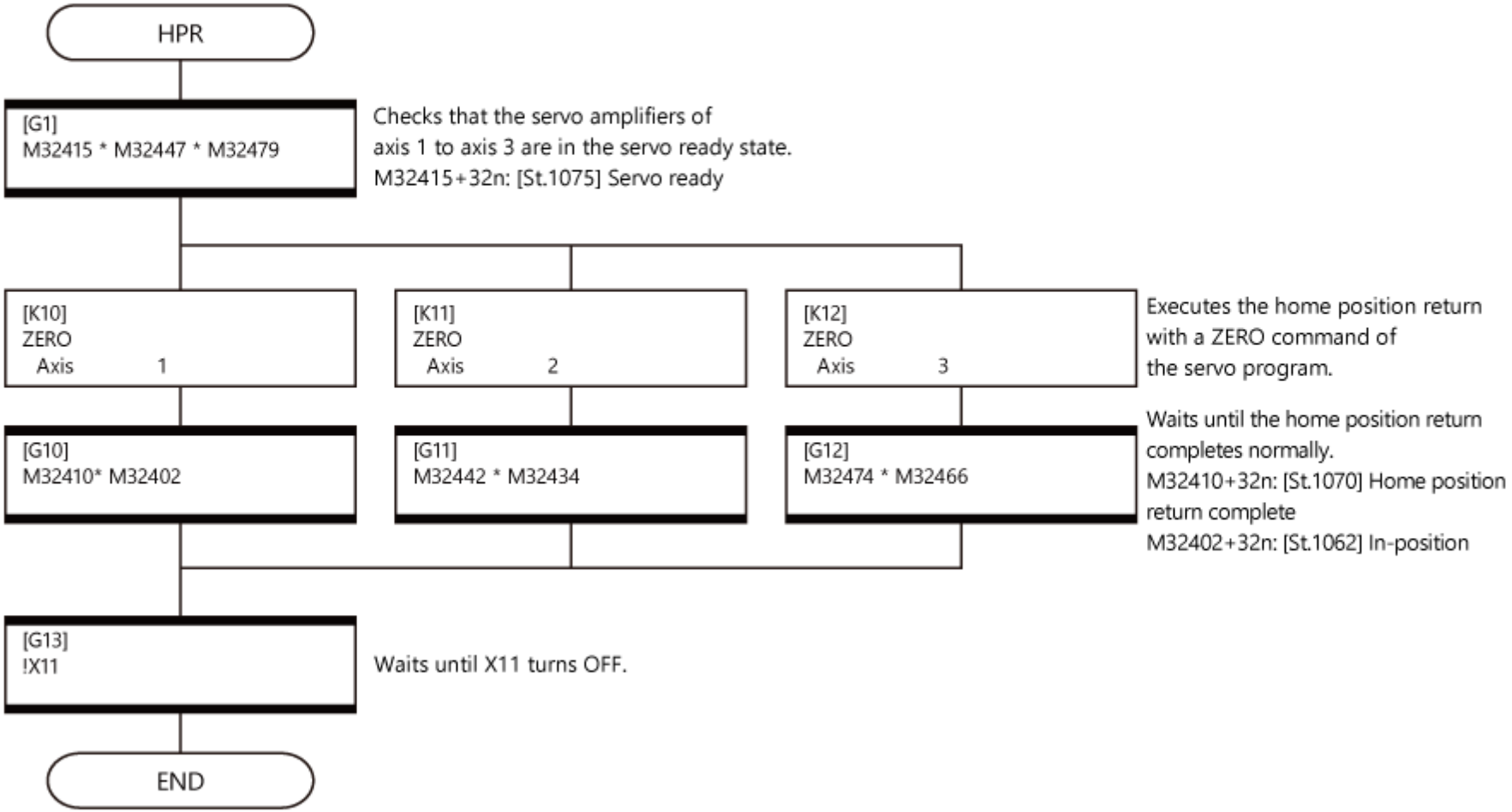
Switches the program that is executed by the input device.



4.1 Description of Sample Program

(3)No. 010: HPR (no automatic start)

This program performs the all-axis home position return.



4.1 Description of Sample Program

(4)No. 011: Ax1Posi (no automatic start)

This program performs the positioning control using only axis 1 (X-axis).

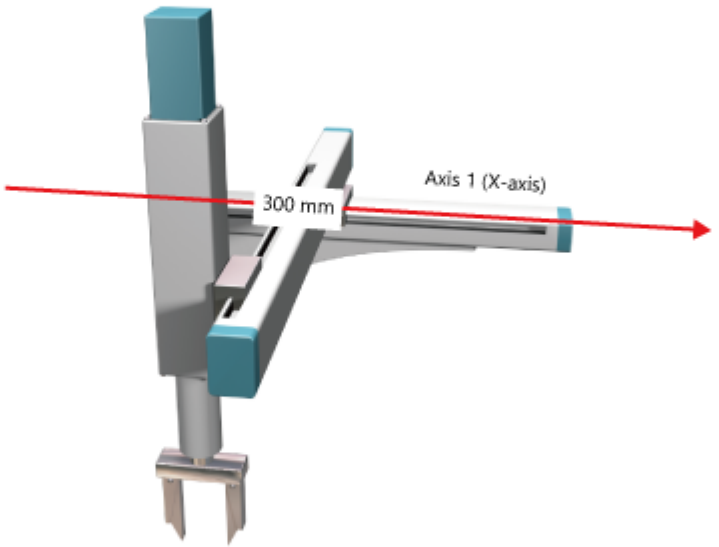
Ax1Posi

[G20]
!M32409

Checks that the home position return request of axis 1 is off.
M32409+32n: [St.1069] Home position return request

[K20]
ABS-1
Axis 1
└Address 300000.0 µm
Speed 3000.00 mm/min
P.B. 1
Dwell 100 ms

Checks that the home position return request of axis 1 is off.
M32409+32n: [St.1069] Home position return request



[G21]
M32401 * M32402

Waits until positioning completes.
M32401+32n: [St.1061] Positioning complete
M32402+32n: [St.1062] In-position

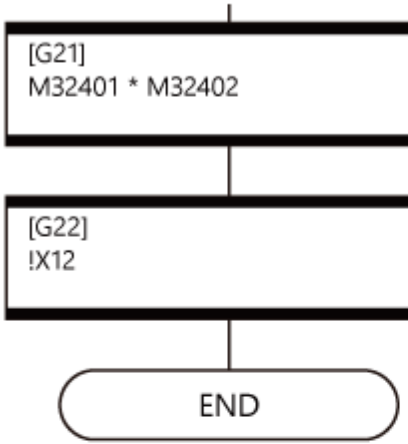
[K21]
ABS-1
Axis 1
└Address 0.0 µm
Speed 3000.00 mm/min
P.B. 1
Dwell 100 ms

Performs positioning with an ABS-1 command in the servo program.
Moves from the 300 mm position to the 0 mm position.

[G21]

Waits until positioning completes

4.1 Description of Sample Program



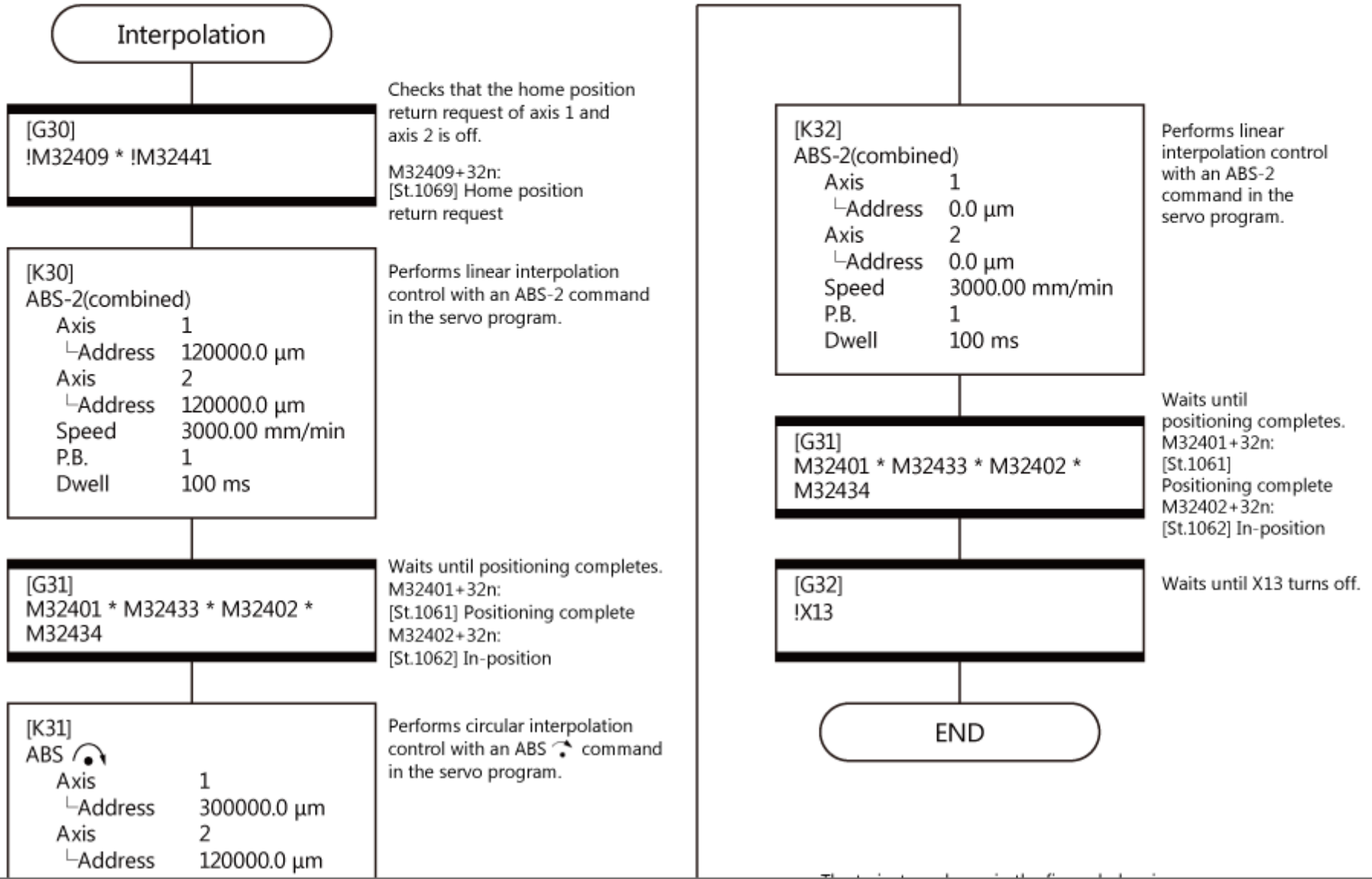
Waits until positioning completes.
M32401+32n: [St.1061] Positioning complete
M32402+32n: [St.1062] In-position

Waits until X12 turns OFF.

4.1 Description of Sample Program

(5)No. 012: Interpolation (no automatic start)

This program performs linear interpolation and circular interpolation using axis 1 (X-axis) and axis 2 (Y-axis).



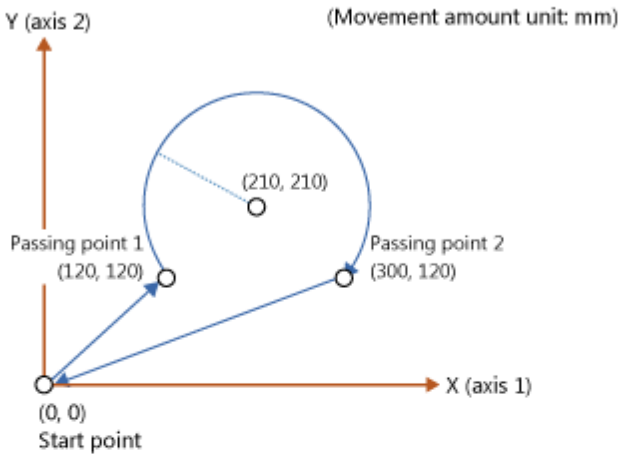
4.1 Description of Sample Program

Speed	3000.00 mm/min
Central point	1
└Address	210000.0 µm
Central point	2
└Address	210000.0 µm
P.B.	1
Dwell	100 ms

[G31]
M32401 * M32433 * M32402 * M32434

Waits until positioning completes.
M32401+32n:
[St.1061] Positioning complete
M32402+32n:
[St.1062] In-position

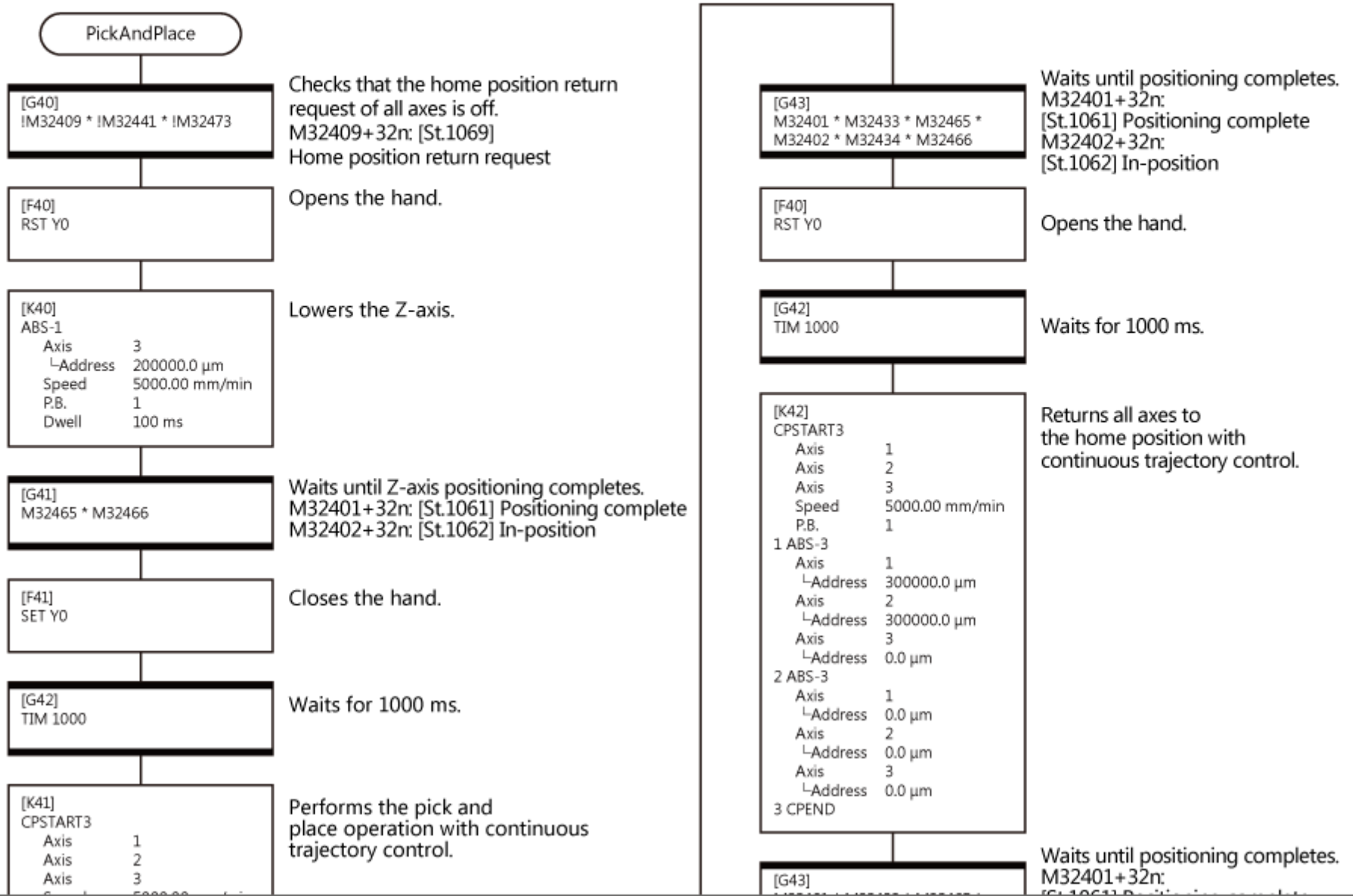
The trajectory shown in the figure below is traced in this program.



4.1 Description of Sample Program

(6)No. 013: PickAndPlace (no automatic start)

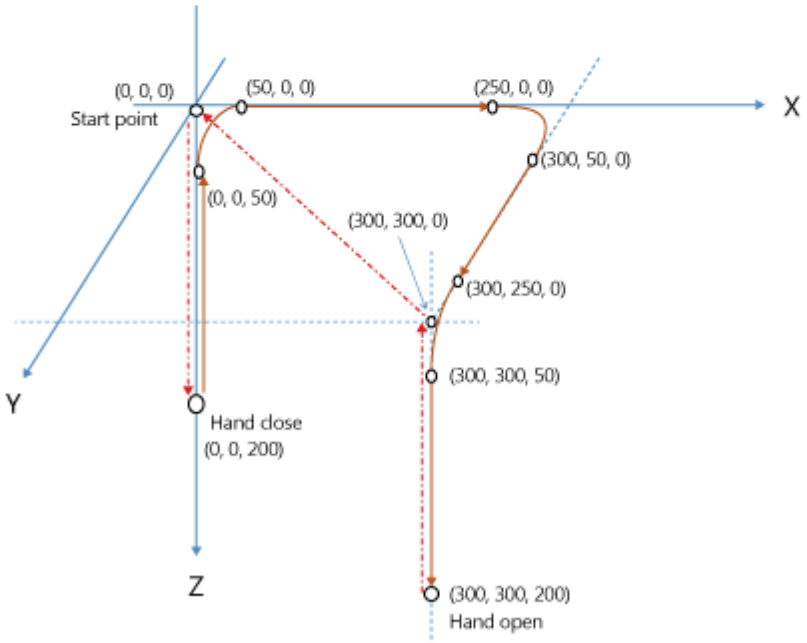
This program performs the pick and place operation using all axes.



4.1 Description of Sample Program

Speed	5000.00 mm/min
P.B.	1
1 ABS-3	
Axis	1
└Address	0.0 μm
Axis	2
└Address	0.0 μm
Axis	3
└Address	50000.0 μm
2 ABS	
Axis	1
└Address	50000.0 μm
Axis	3
└Address	0.0 μm
Radius	50000.0 ms
3 ABS-3	
Axis	1
└Address	250000.0 μm
Axis	2
└Address	0.0 μm
Axis	3
└Address	0.0 μm
4 ABS	
Axis	1
└Address	300000.0 μm
Axis	2
└Address	50000.0 μm
Radius	50000.0 ms
5 ABS-3	
Axis	1
└Address	300000.0 μm
Axis	2
└Address	250000.0 μm
Axis	3
└Address	0.0 μm
6 ABS	
Axis	2
└Address	300000.0 μm
Axis	3
└Address	50000.0 μm
Radius	50000.0 ms
7 ABS-3	
Axis	1
└Address	300000.0 μm
Axis	2
└Address	300000.0 μm
Axis	3
└Address	200000.0 μm
8 CPEND	

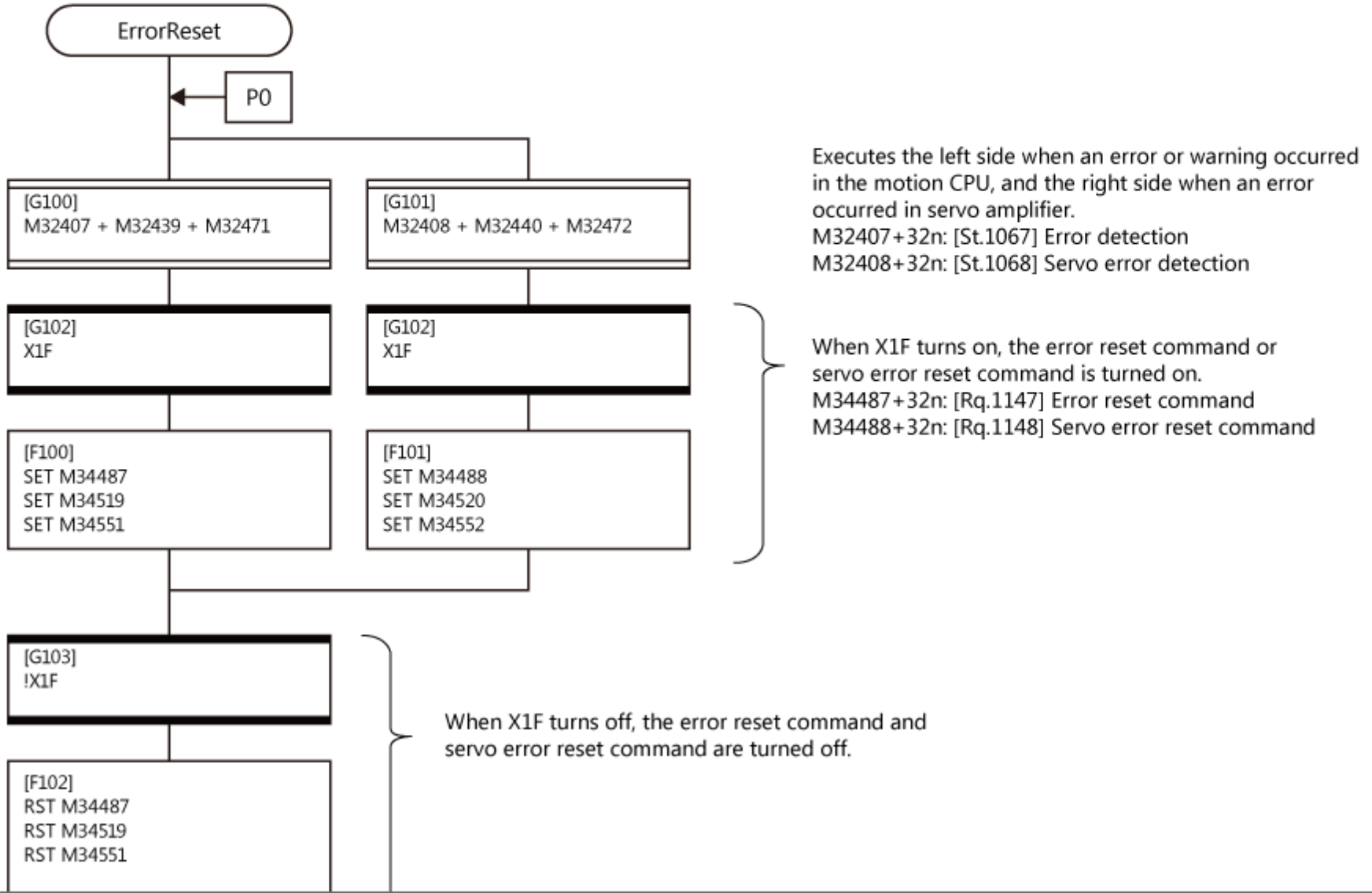
The trajectory shown in the figure below is traced in this program.



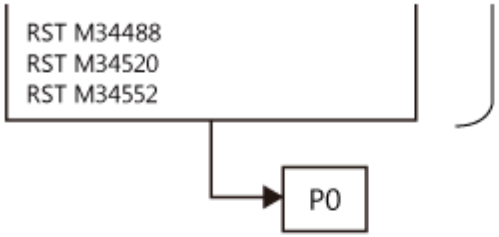
4.1 Description of Sample Program

(7)No. 100: ErrorReset (automatic start)

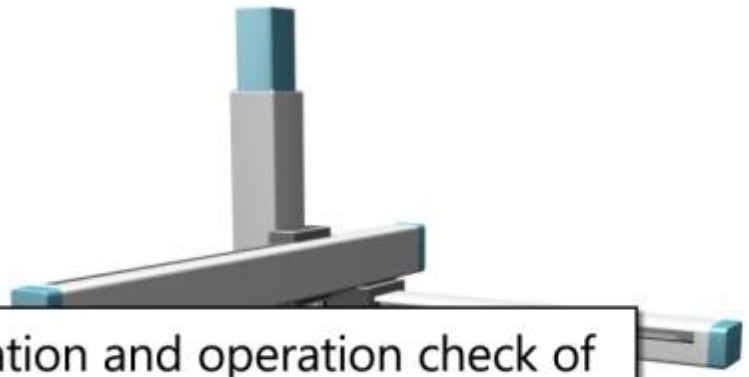
This program performs the error reset.



4.1 Description of Sample Program



4.2 Operation Check



This completes the explanation and operation check of the sample program.
Proceed to the next page.

4.3 Summary of This Chapter

In this chapter, you have learned:

- Description of Sample Program
- Operation Check of Sample Program

Points

Description of Sample Program	• The initial setting program and error reset are started automatically, and other programs are executed by calling subroutines. • Learned about the sample programs for the home position return, 1-axis positioning, 2-axis interpolation control, and continuous trajectory control program which you learned in Chapter 3.
Operation Check of Sample Program	• Learned how the sample system is operated by the sample programs in a video.

Test Final Test

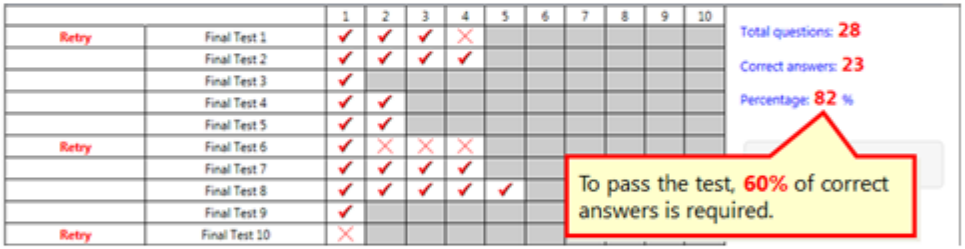
Now that you have completed all of the lessons of the **MELSEC iQ-R Series Motion Controller Basics (RnMTCPU)** Course, you are ready to take the final test. If you are unclear on any of the topics covered, please take this opportunity to review those topics.

There are a total of 5 questions (14 items) in this Final Test.

You can take the final test as many times as you like.

Score results

The number of correct answers, the number of questions, the percentage of correct answers, and the pass/fail result will appear on the score page.



TEST

Final Test 1

Select the correct word or phrase for ( ) in the following sentences.

- The engineering software for MELSEC iQ-R series programmable controller CPU is (Q1) and the engineering software for MELSEC iQ-R series motion CPU is (Q2).
- When a motion CPU is used, the system will always be (Q3).

Q1 Select the correct words

Q2 Select the correct words

Q3 Select the correct words

TEST

Final Test 2

Select the sentences below that are correct. (Multiple sentences can be selected.)

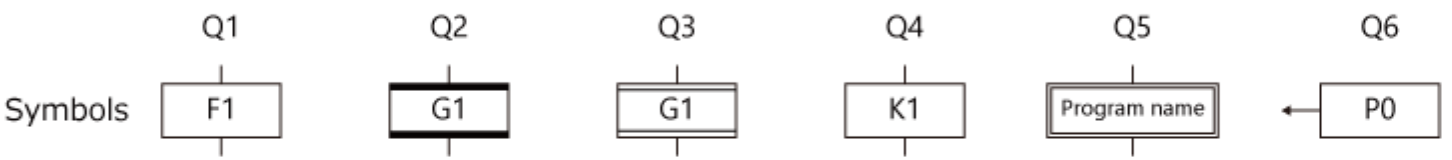
Q1

- [ ] The data communication between CPU modules are performed by data communication using the CPU buffer memory and data communication using a fixed scan communication area.
- [ ] There is no problem if the device assignment method in the project file and the device assignment method set in the motion CPU is different.
- [ ] The device assignment method of the motion CPU can be performed with Q series compatible assignment and MELSEC iQ-R Motion device assignment.
- [ ] The basic settings and servo network settings are configured in the system parameter of motion CPU.
- [ ] Steps, transitions, and function blocks are available for the motion SFC element.

TEST

Final Test 3

Select the names of the motion SFC program symbols from the following options.



Q1 Select the correct words

Q2 Select the correct words

Q3 Select the correct words

Q4 Select the correct words

Q5 Select the correct words

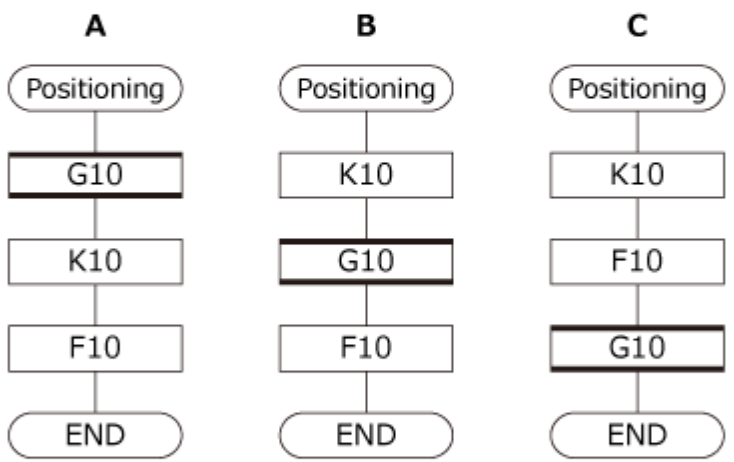
Q6 Select the correct words

TEST Final Test 4

From the following motion SFC programs, select the correct program that waits for motion control step movement to complete and then shifts to the next process.

Q1

- A
- B
- C



TEST

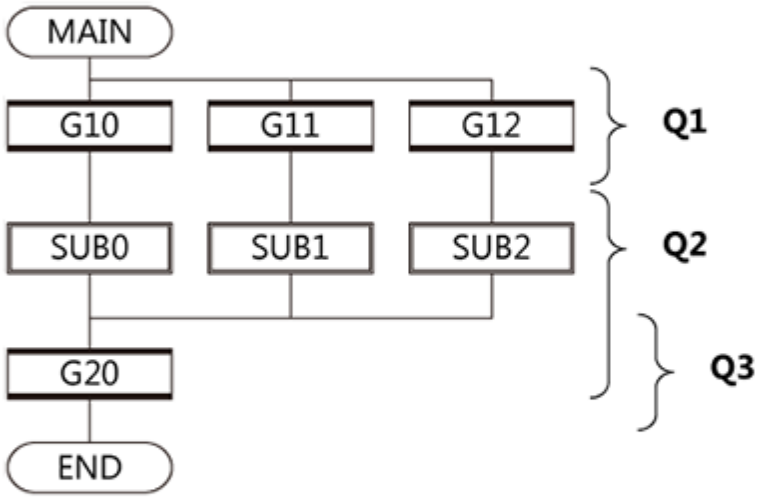
Final Test 5

Select the name of the type of each part in the following motion SFC program from the following options.

Q1 Select the correct words

Q2 Select the correct words

Q3 Select the correct words



Test Score

You have completed the Final Test. You results area as follows.
To end the Final Test, proceed to the next page

		1	2	3	4	5	6	7	8	9	10
	Final Test 1	✓	✓	✓							
	Final Test 2	✓									
	Final Test 3	✓	✓	✓	✓	✓	✓				
	Final Test 4	✓									
	Final Test 5	✓	✓	✓							

Total questions: **14**

Correct answers: **14**

Percentage: **100** %

Clear

You have completed the "MELSEC iQ-R Series Motion Controller Basics (RnMTCPU)" Course.

Thank you for taking this course.

We hope you enjoyed the lessons and the information you acquired in this course is useful for configuring systems in the future.

You can review the course as many times as you want.

Review

Close