Servo System Controller

MELSEC iQ-R Series Motion Module Basics (RD78G(H)/Startup)

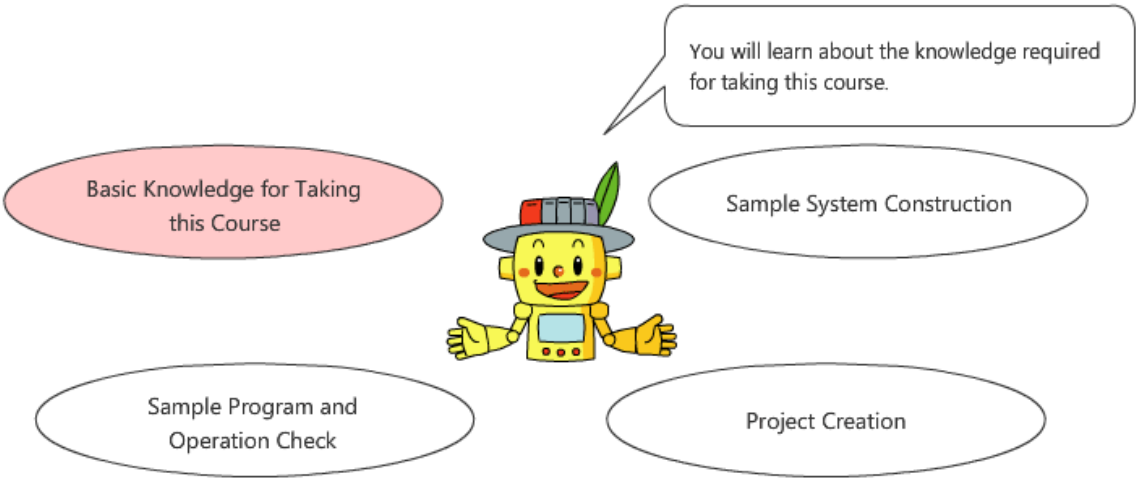
This course is for participants who will establish a motion control system using the MELSEC iQ-R series motion module for the first time.

Click the Forward button at the upper right of the screen to proceed to the next page.

L(NA)00238ENG-A

Introduction Purpose of the Course

This course is intended for those who will establish a motion control system using the MELSEC iQ-R series motion module for the first time and provides basic knowledge from system design to installation, wiring, setting, and programming.



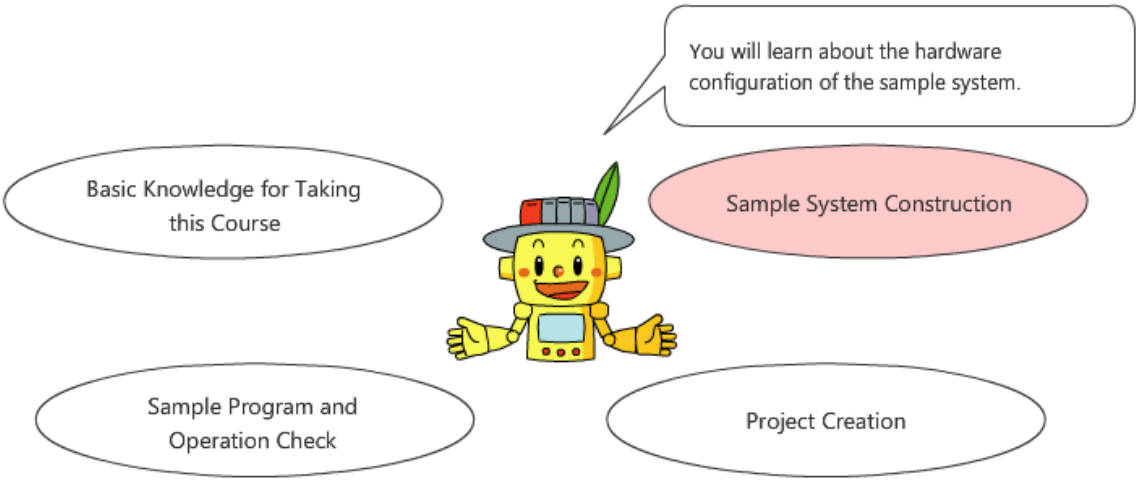
This course requires the basic knowledge of MELSEC iQ-R series PLCs, AC servos, and the positioning

control. For beginners, taking the following courses is recommended.

- "MELSEC iQ-R Series Basic" course
- "GX Works3 (Ladder)" course
- "Programming Basics (Structured Text)" course
- "FA Equipment for Beginners (Positioning)" course

Introduction Purpose of the Course

This course is intended for those who will establish a motion control system using the MELSEC iQ-R series motion module for the first time and provides basic knowledge from system design to installation, wiring, setting, and programming.



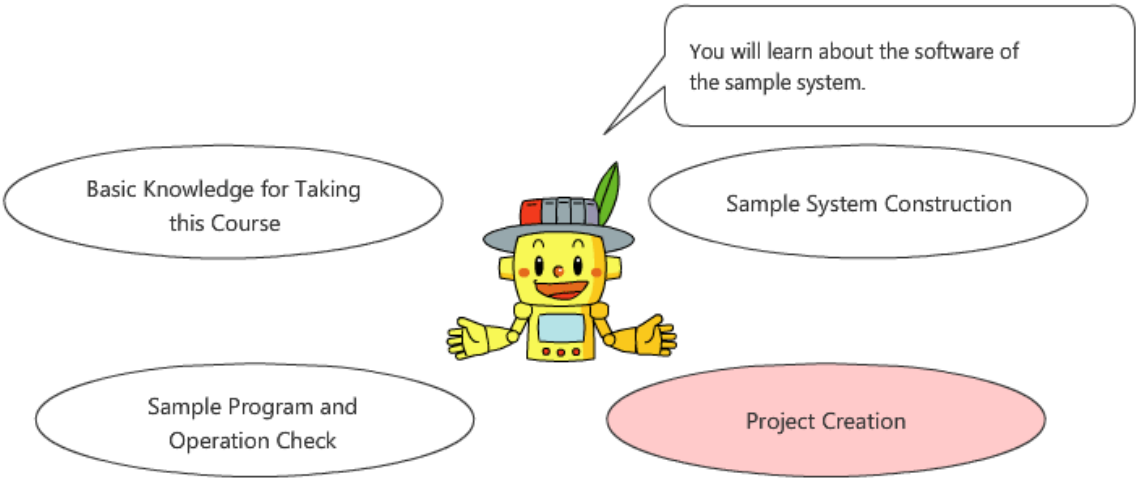
This course requires the basic knowledge of MELSEC iQ-R series PLCs, AC servos, and the positioning

control. For beginners, taking the following courses is recommended.

- "MELSEC iQ-R Series Basic" course
- "GX Works3 (Ladder)" course
- "Programming Basics (Structured Text)" course
- "FA Equipment for Beginners (Positioning)" course

Introduction Purpose of the Course

This course is intended for those who will establish a motion control system using the MELSEC iQ-R series motion module for the first time and provides basic knowledge from system design to installation, wiring, setting, and programming.



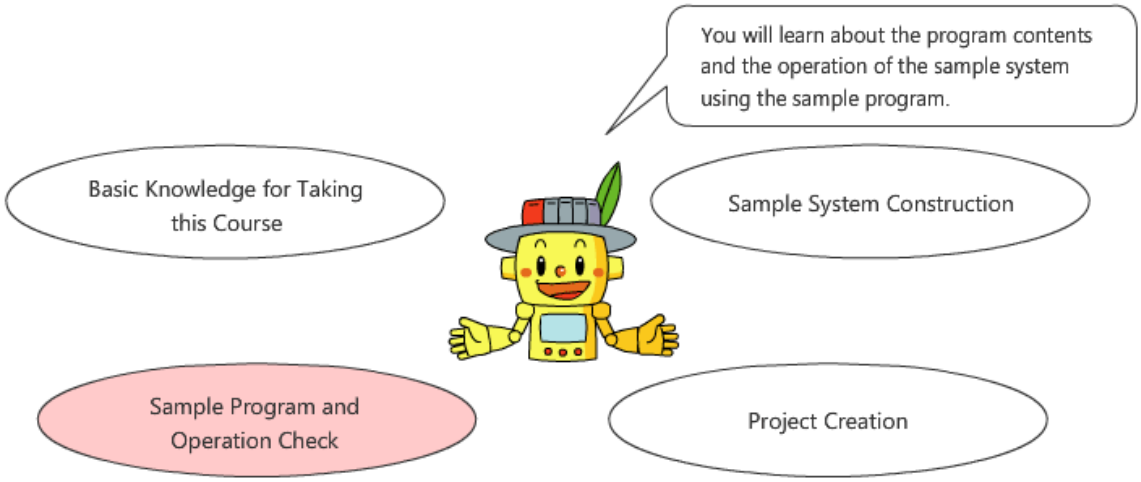
This course requires the basic knowledge of MELSEC iQ-R series PLCs, AC servos, and the positioning

control. For beginners, taking the following courses is recommended.

- "MELSEC iQ-R Series Basic" course
- "GX Works3 (Ladder)" course
- "Programming Basics (Structured Text)" course
- "FA Equipment for Beginners (Positioning)" course

Introduction Purpose of the Course

This course is intended for those who will establish a motion control system using the MELSEC iQ-R series motion module for the first time and provides basic knowledge from system design to installation, wiring, setting, and programming.



This course requires the basic knowledge of MELSEC iQ-R series PLCs, AC servos, and the positioning

control. For beginners, taking the following courses is recommended.
- "MELSEC iQ-R Series Basic" course
- "GX Works3 (Ladder)" course
- "Programming Basics (Structured Text)" 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 - Basic Knowledge for Taking this Course

This chapter describes the knowledge required for taking this course.

Chapter 2 - Sample System Construction

This chapter describes the hardware configuration of the sample system.

Chapter 3 - Project Creation

This chapter describes the software of the sample system.

Chapter 4 - Sample Program and Operation Check

This chapter describes the program contents and the operation of the sample system using the sample program.

Final Test

5 sections in total (7 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.

Cautions for Use

■Safety precautions

When you learn based on using actual products, please carefully 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.
This course is for the following software versions.
For the latest version of each software, check the Mitsubishi Electric FA Website.

MELSOFT GX Works3	Ver.1.072A	Motion Control Setting	Ver.1.015R
MELSOFT MR Configurator2	Ver.1.115V		

The firmware version of the PLC CPU must be 44 or later (46 or later for RD78GH).
The firmware version of the motion module must be 14 or later.
For how to update the firmware version, refer to module configuration manual.

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

Manual name	Manual No.	Version
MELSEC iQ-R Motion Module User's Manual (Startup)	IB-0300406	C
MELSEC iQ-R Motion Module User's Manual (Application)	IB-0300411	C
MELSEC iQ-R Motion Module User's Manual (Network)	IB-0300426	C
MELSEC iQ-R Programming Manual (Motion Module Instructions, Standard Functions/Function Blocks)	IB-0300431	C
MELSEC iQ-R Programming Manual (Motion Control Function Blocks)	IB-030533	A
MELSEC iQ-R Structured Text (ST) Programming Guide Book	SH-081483	E
MELSEC iQ-R Programming Manual (CPU Module Instructions, Standard Functions/Function Blocks)	SH-081266	W
MELSEC iQ-R CPU Module User's Manual (Application)	SH-081264	AF

Chapter 1 Basic Knowledge for Taking this Course

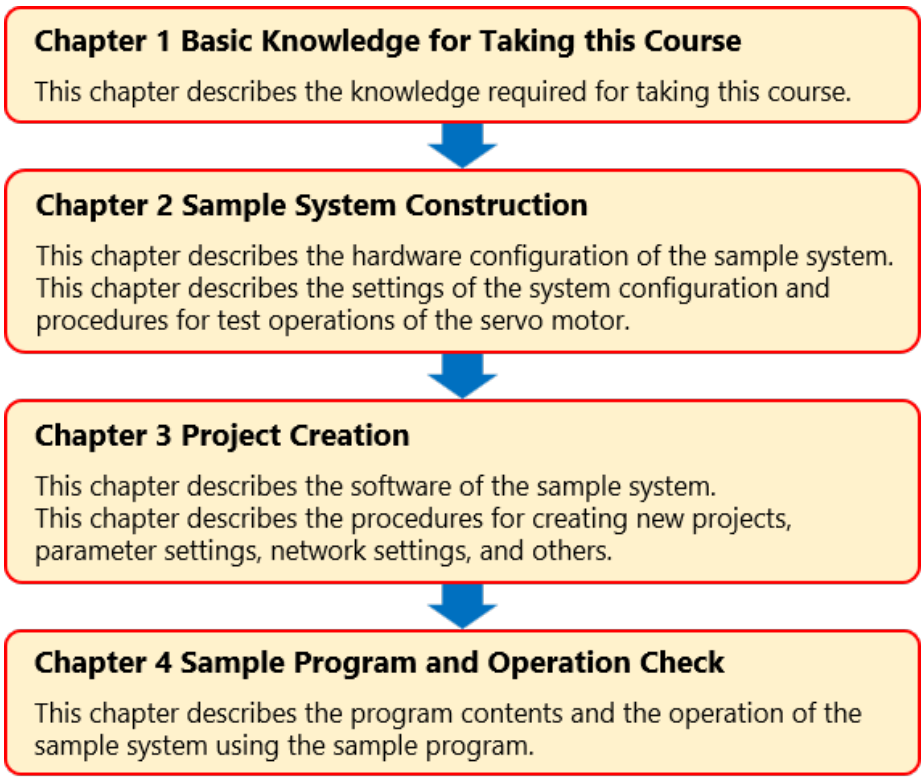
1.1 Subject of this Course

In this course, you will learn how to control the mechanism of a one-axis ball screw by using the motion module RD78G and AC servo of the MELSERVO-J5 series.

The following PTP operation is the subject of this course.

1.2 Flow of this Course

The following shows a flow of this course.



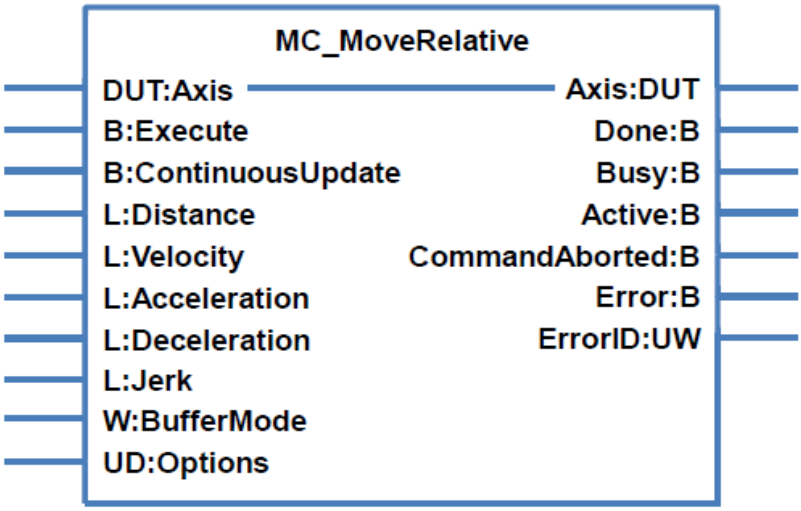
1.3 PLCopen® Motion Control FB

PLCopen®is a third-party organization, aiming at improving the development efficiency of PLC applications, promoting the international standard IEC 61131-3 for PLC programming, and creating and certificating the standard function block (FB) specifications that are independent of the vendor.

Using the FB specified by PLCopen®enables the programming independent of PLC manufacturers since the I/O and operation specifications of the FB are standardized.
This makes the program structured and improves the reusability, resulting in the reduction of engineering cost.
The motion control is defined as Motion Control FB.

The motion module is compatible with this Motion Control FB (hereafter, referred to as MCFB) and uses this FB for programming. (For details, refer to Chapter 4.)

Example) MC_MoveRelative (Relative value positioning control)



1.4 Programming Using ST

This section describes how to create ST programs and provides explanations of the structure of the ST.

(1) Reference manual
For the details of programing using the ST, refer to the following manuals.
Note that the commands that can be used differ between the PLC CPU module and the motion module.

Format of the ST

MELSEC iQ-R Structured Text (ST) Programming Guide Book

Commands that can be used in the ST

MELSEC iQ-R Programming Manual (CPU Module Instructions, Standard Functions/Function Blocks)

MELSEC iQ-R Programming Manual (Motion Module Instructions, Standard Functions/Function Blocks)

Labels and structures

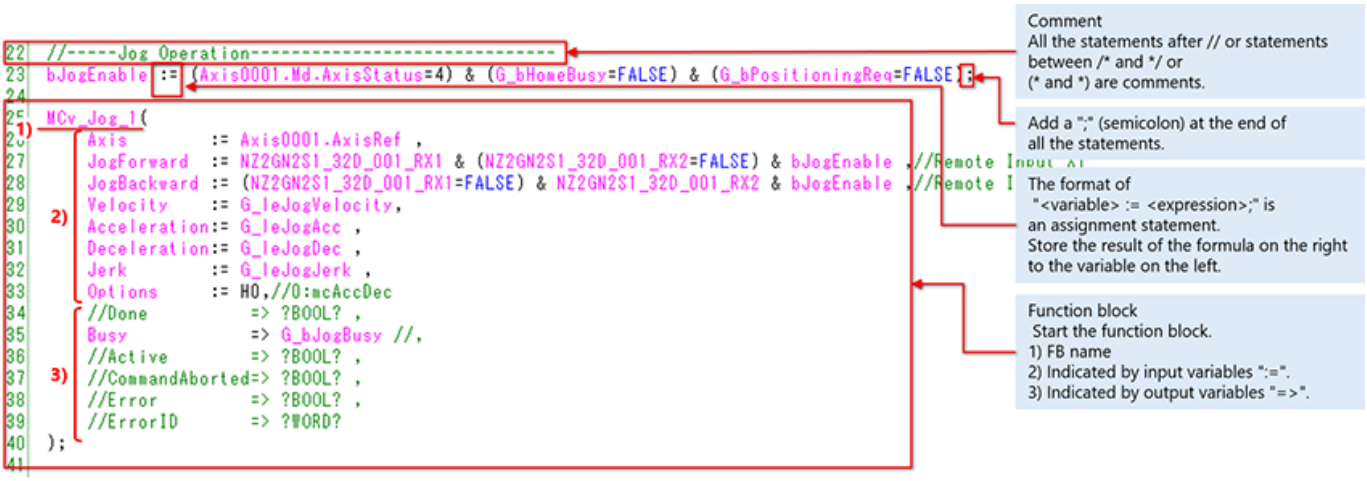
MELSEC iQ-R CPU Module User's Manual (Application)

Program example

MELSEC iQ-R Programming Manual (Motion Control Function Blocks)

(2) Basic rules of the ST (extract)

The following shows a part of the sample program.



1.5 Label, Arrangement, and Structure

(1) Label, arrangement, and structure

In programs of a motion module, labels are used instead of devices and buffer memory numbers.
A label is a variable consisting of a specified string used in I/O data or internal processing.
Using labels in programming enables creation of programs without being aware of devices and buffer memory sizes. For this reason, a program using labels can be reused easily even in a system having a different module configuration. Arrangement is a data type representing a collection of labels with the same data type using one name.
Structure is a data type representing a collection of labels with different formats using one name.

(2)Label type

- Local label ··· A local label is a label that can be used only in each POU. Local labels outside the POUs cannot be used.
  The settings of a local label include a label name, class, and data type.
- Global label ··· A global label is a label that provides the same data within a single project. It can be used in all programs in the project. (However, when using global labels of the motion module as those of the PLC CPU, the settings of the public labels are required. (NOTE))
  A global label can be used in program blocks and function blocks.
  The settings of a global label include a label name, class, and data type.
  In the CPU module, devices can be assigned to global labels.
- Module label ··· A module label is a label defined uniquely by each module. It is automatically generated by the engineering tool from the module used, and can be used as a global label.
- System label ··· A system label is a label that provides the same data in all projects compatible with iQ Works.
  It can be referenced from the GOT and the CPU modules on other stations, and used for monitoring and accessing data.
  (This label is not used in this course.)
- Slave label ··· For the public labels, refer to the following manual.

(NOTE) For the public labels, refer to the MELSEC iQ-R Series Motion Module Basics (RD78G(H)/Positioning Control), which is a course of a online training system, and the following manual.

MELSEC iQ-R Programming Manual (Motion Control Function Blocks)
4.2 Motion Module Program Creation

(3) Label data type

The following table shows the main label data types.

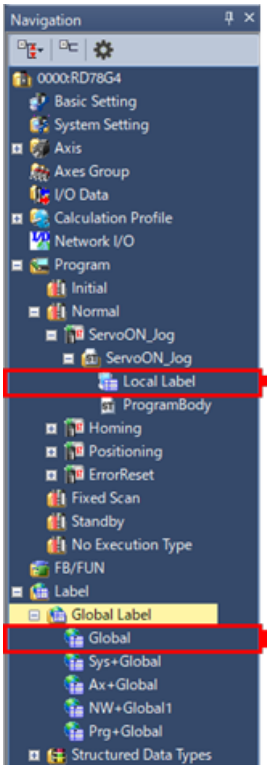
The sample program used in this course indicates the data type with the prefix of the label.

Data type		Range	Prefix
Bit	BOOL	FALSE(0), TRUE(1)	b
Word (unsigned)/bit string (16 bits)	WORD (UINT)	0 to 65535	u
Double word (unsigned)/bit string (32 bits)	DWORD (UDINT)	0 to 4294967295	ud
Word (signed)	INT	-32468 to 32767	w
Double word (signed)	DINT	-2147483648 to 2147483647	d
Single-precision real number	REAL	-2^{128} to -2^{-126}, 0, 2^{-126} to 2^{128}	e
Double-precision real number	LREAL	-2^{1024} to -2^{-1022}, 0, 2^{-1022} to 2^{1024}	le
Time	TIME	T#-24d20h31m23s648ms to T#24d20h31m23s647ms	tm
Timer	TIMER	TIMER is the structure. S (contact): BOOL C (coil): BOOL N (current value): WORD	td

In addition, for global labels, "G_" is added to the beginning of the label name.

(4) Label registration method

- Local label
  [Local Label] is provided for each program under [Program] in the project tree. Double-click here to open the local label editor.
- Global label
  Double-click [Label] → [Global Label] → [Global] in the project tree to open the global label editor.



	Label Name	Data Type	Class	Initial Value	Constant	Comment
1	MC_Power_1	MC_Power	VAR			
2	bPowerStatus	Bit	VAR			Servo ON/OFF status
3	bReadyStatus	Bit	VAR			Ready ON/OFF status
4	bPowerBusy	Bit	VAR			MC_Power Busy
5	bPowerError	Bit	VAR			MC_Power Error
6	uPowerErrorID	Word [Unsigned]/Bit String [16-bit]	VAR			MC_Power Error ID
7	bJogEnable	Bit	VAR			Jog Operation Enable
8	MCv_Jog_1	MCv_Jog	VAR			
9						

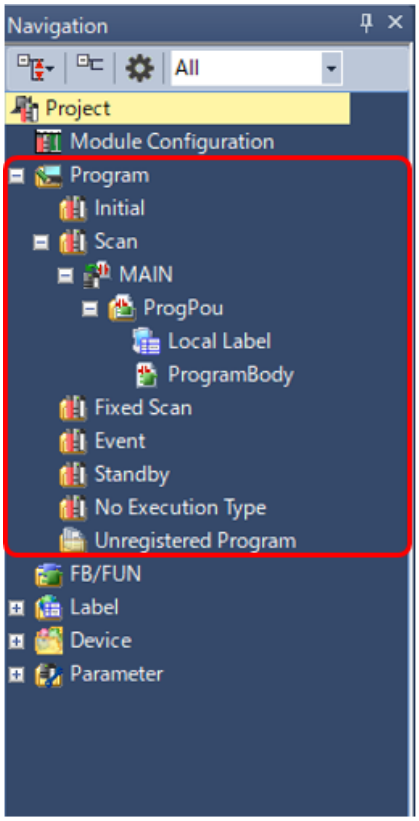
Example of the local label editor

	Label Name	Data Type	Class	Initial Value	Constant	Comment
1	G_leJogVelocity	FLOAT [Double Precision]	VAR_GLOBAL			JOG Velocity
2	G_leJogAcc	FLOAT [Double Precision]	VAR_GLOBAL			JOG Acceleration
3	G_leJogDec	FLOAT [Double Precision]	VAR_GLOBAL			JOG Deceleration
4	G_leJogJerk	FLOAT [Double Precision]	VAR_GLOBAL			JOG Jerk
5	G_bJogBusy	Bit	VAR_GLOBAL			MC_Jog Busy
6	G_bPositioningReq	Bit	VAR_GLOBAL			Positioning Request
7	G_lePoint0Address	FLOAT [Double Precision]	VAR_GLOBAL			Home Position Address
8	G_lePoint1Address	FLOAT [Double Precision]	VAR_GLOBAL			Positioning Address
9	G_bHomeBusy	Bit	VAR_GLOBAL			MC_Home Busy
10						

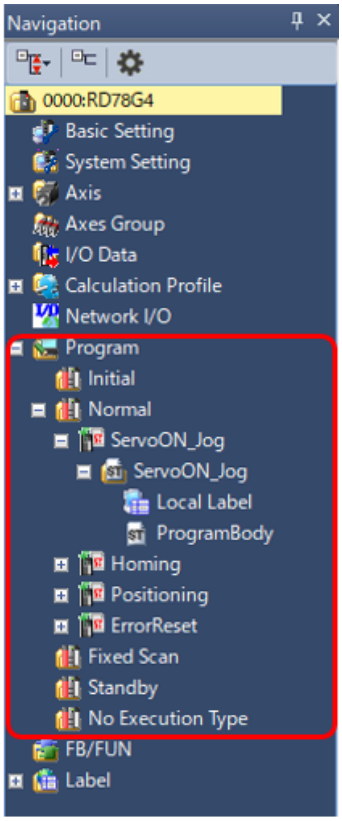
Example of the global label editor

1.6 Program Type

Programs of both PLC CPU and motion module are classified into the following program types.



<Project tree of GX Works3>



<Project tree of the motion control setting function>

Initial execution type program
This program type is executed only once when the CPU module powers ON or changes from the STOP status to the RUN status.

Scan execution type program (PLC CPU)/ normal execution type program (Motion module)
This program type is executed only once per scan from the scan following the scan where an initial execution type program was executed.

Fixed scan execution type program
An interrupt program which is executed at a specified time interval. Different from the normal interrupt program, this type of program does not require interrupt pointer (I) and IRET instruction to be written. Execution is performed by program file basis.

Event execution type program (PLC CPU)
This type of program starts execution when triggered by a specified event. The program is executed at the execution turn specified in program setting of the CPU parameters, and if execution conditions of specified trigger are met when the execution turn of the event execution type program comes, the program is executed.

Standby type program
This program is executed only when there is an execution request.

No execution type, unregistered program
This program type is not executed on the CPU module. Programs with no execution type specified (if selected) are written to the CPU. Unregistered programs are not written.

1.7 Summary of This Chapter

In this chapter, you have learned:

- PLCopen® Motion Control FB
- Programming Using ST
- Label, arrangement, and structure
- Program Type

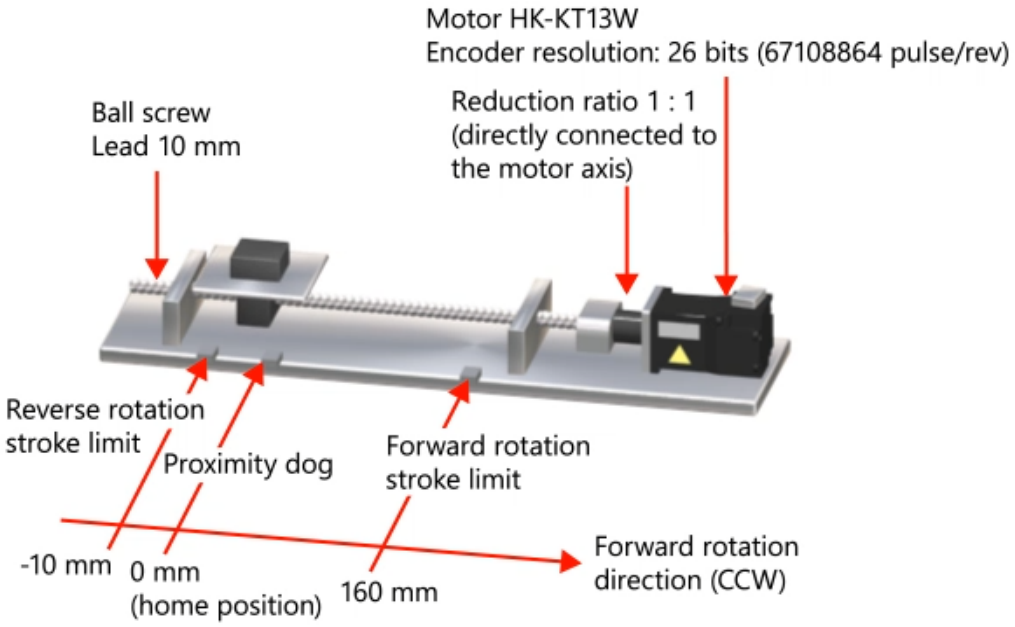
Important points

PLCopen® Motion Control FB	• PLCopen®, a third-party organization, develops the standard FB specifications that are independent of the vendor. • The motion control is defined as Motion Control FB.
Programming Using ST	• All statements end with ";"(semicolon). • The assignment statement is represented by <variable> := <expression>;. • The input variable of the FB is indicated by ":=",and the output variable is indicated by "=>"
Label, Arrangement, and Structure	• The label types include the local label, global label, module label, system label, and slave label. • Arrangement is a collection of labels with the same variable type. • Structure is a collection of labels with different variable types.
Program Type	• The program execution types include the initial execution type, scan execution type/normal execution type, fixed scan execution type, event execution type, standby execution type, and no execution type/unregistered program.

Chapter 2 Sample System Construction

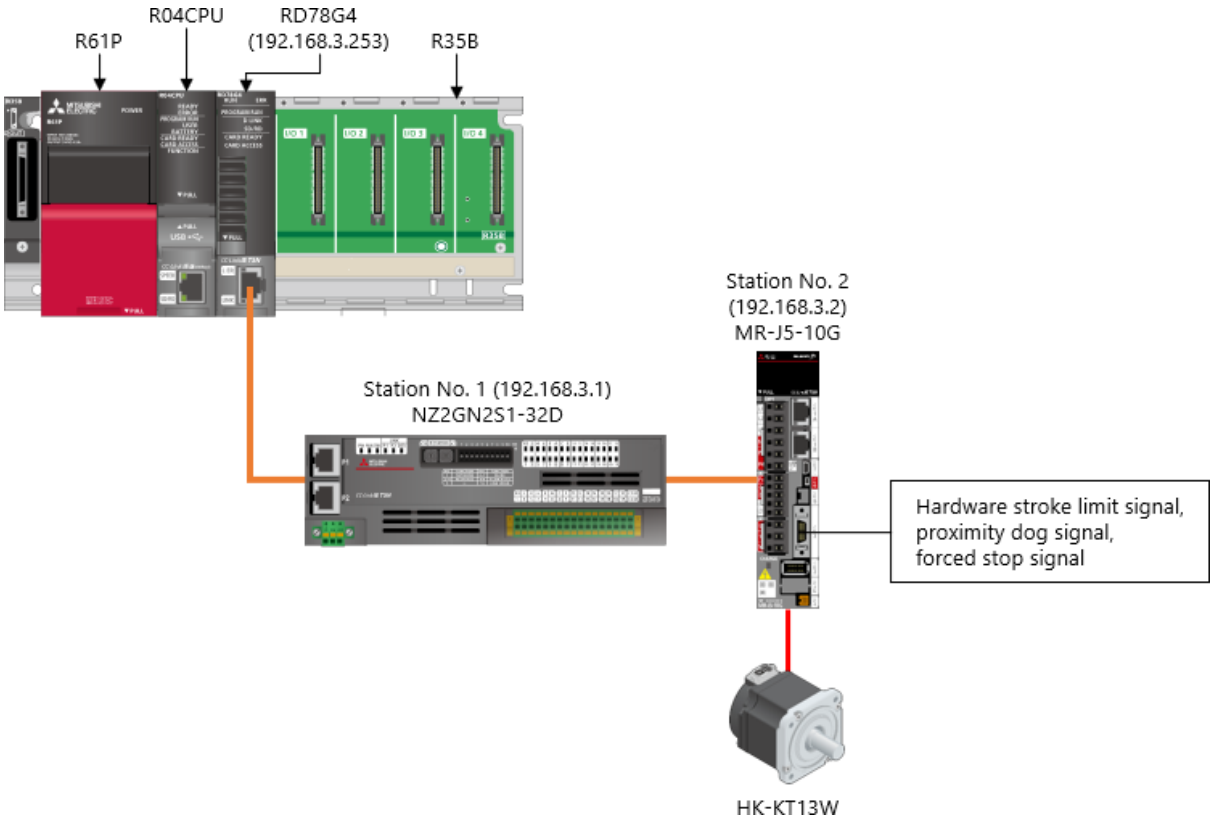
2.1 Device Configuration

Use the mechanism of a one-axis ball screw. The machine specifications are as follows.



2.2 System Configuration

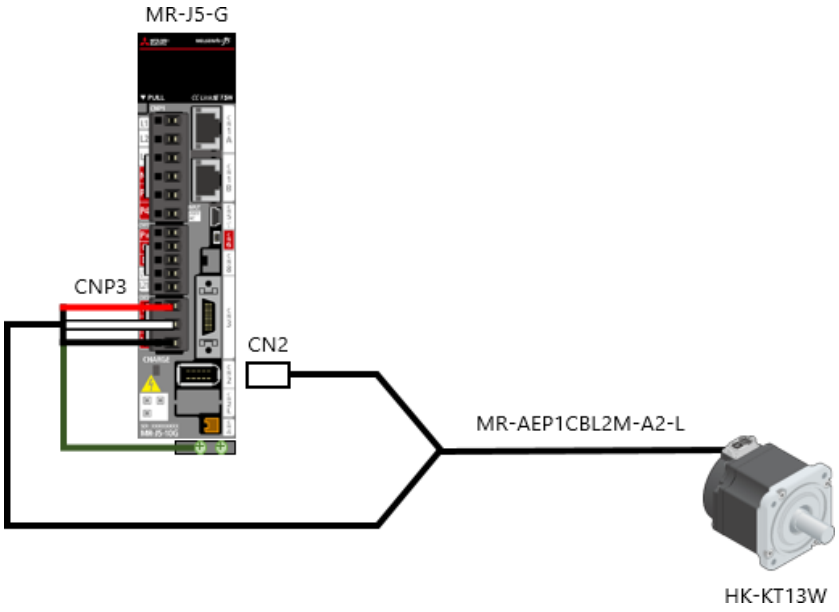
The system configuration of the sample system is as follows.



2.3 Wiring

2.3.1 Connecting a servo motor and a servo amplifier

For the servo motor power cable and encoder cable, use the 1-cable type option MR-AEP1CBL2M-A2-L.

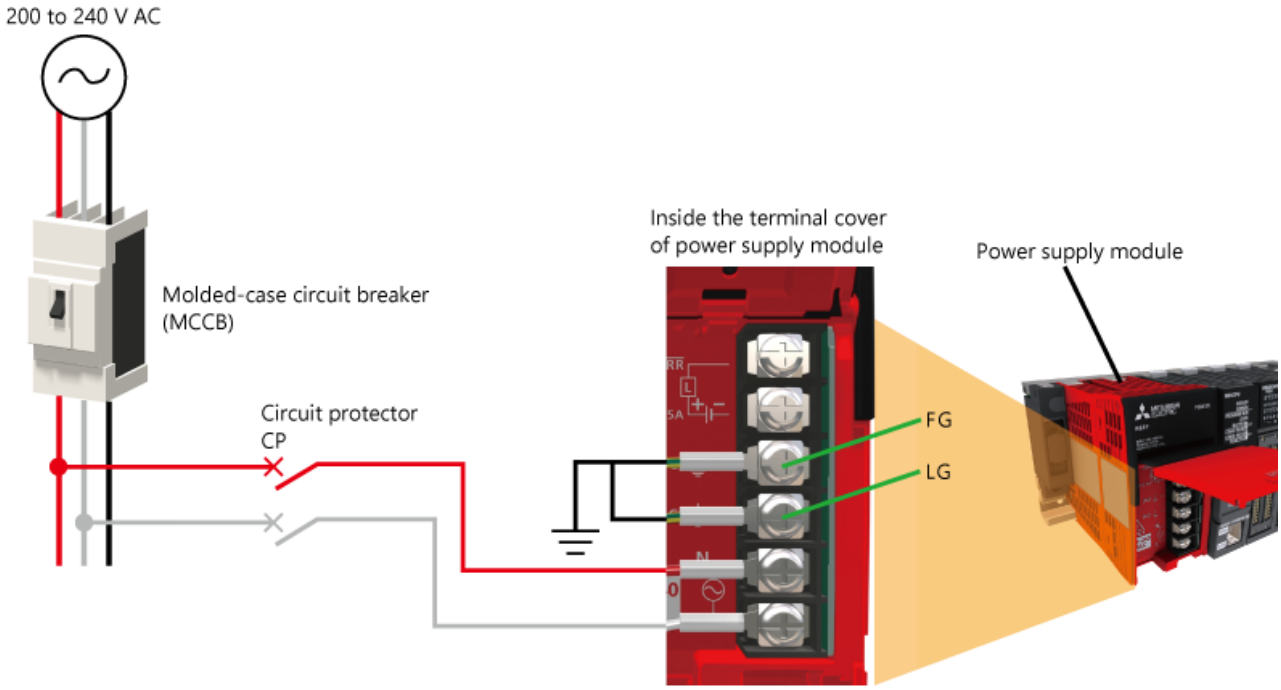


2.3.2 Wiring a power supply and network cables

(1) Wiring the PLC power supply
Wire the power supply to the power supply module of the PLC.

The following describes the wiring of the power supply module.

- Before 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 a ground resistance of 100 Ω or less.



Item	Applicable cable size	Tightening torque
Power supply cable	18 to 14 AWG	1.02 to 1.38 N·m
Grounding cable	18 to 14 AWG	1.02 to 1.38 N·m

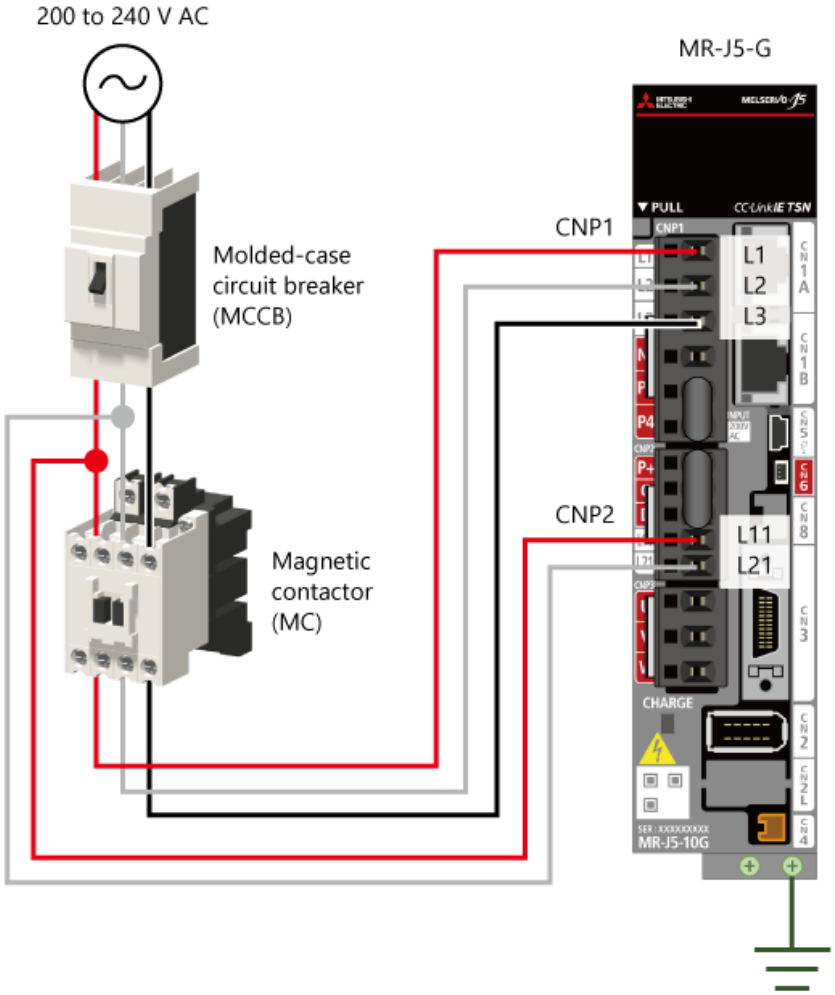
2.3.2 Wiring a power supply and network cables

(2) Wiring the power supply of the servo amplifier

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

The following shows the schematic diagram. The actual wiring and applicable cable sizes differ depending on the capacity. For details, refer to the Servo Amplifier User's Manual (Hardware).

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



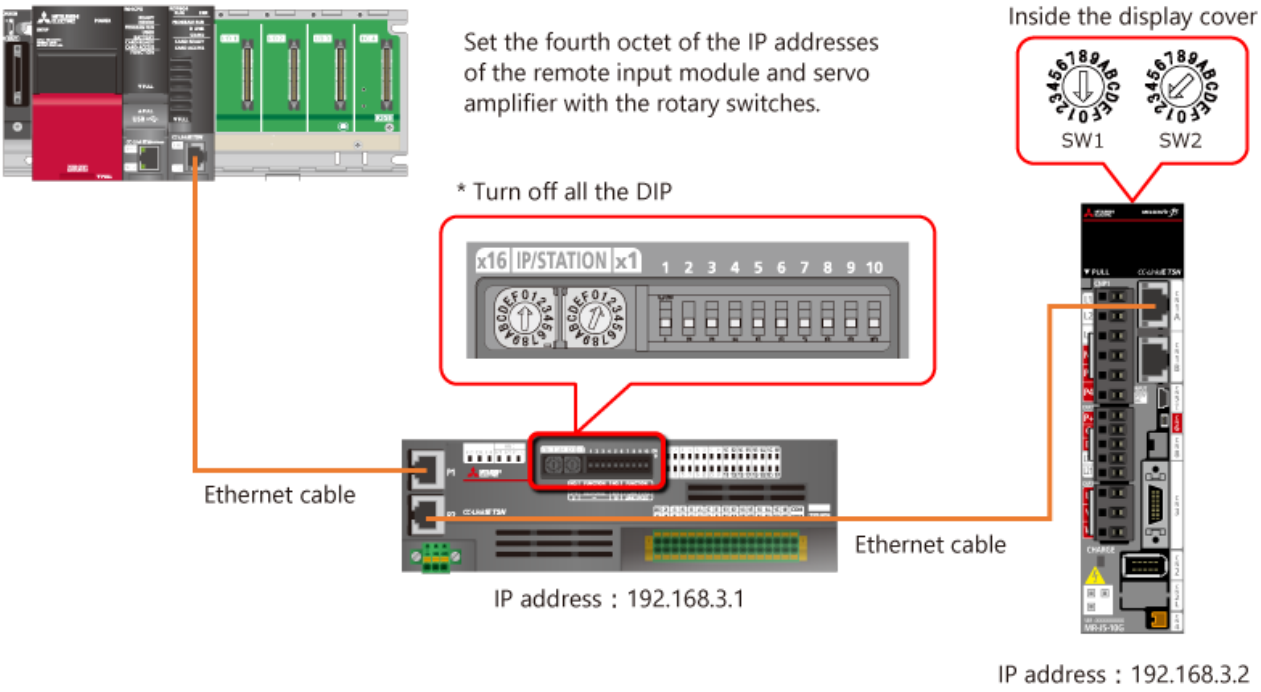
2.3.2 Wiring a power supply and network cables

(3) Wiring network cables

Wire the network cables (Ethernet cables).

Wire the Ethernet cables that satisfy the following standards.

Communication speed	Ethernet cable	Connector	Standard
1Gbps	Category 5e or higher, straight cable (double shielded, STP)	RJ45 connector	Cables that satisfy the following standards • IEEE802.3(1000BASE-T) • ANSI/TIA/EIA-568-B(Category 5e)



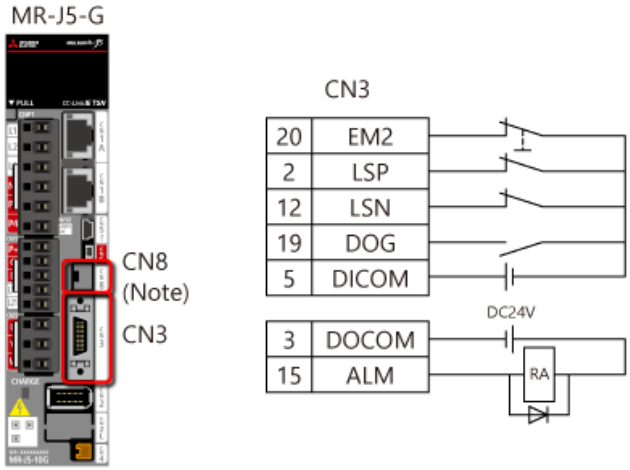
2.3.3 Wiring peripheral circuits

(1) I/O circuit of the amplifier

Wire the I/O circuit of the servo amplifier as follows.

Wire the proximity dog signal, forward/reverse rotation stroke limit switches, and forced stop switch.

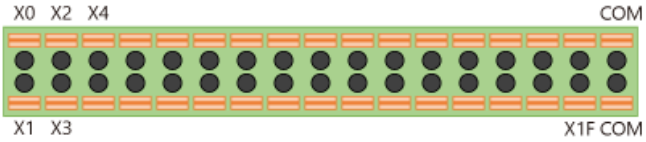
In addition, configure the circuit in which the magnetic contactor (MC) is turned off by ALM output.

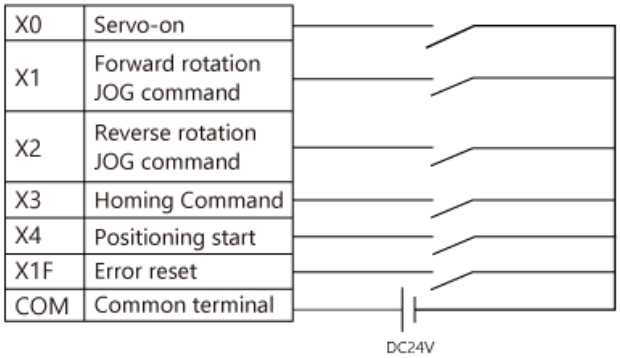


(Note) In this course, the STO function is not used. Thus, do not disconnect the short-circuit connector supplied with the servo amplifier from CN8.

(2) External circuit of the remote input module

Wire the external input circuit of the remote input module as follows.



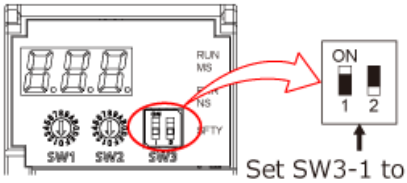


Use the alternate operation switch only for Servo-on (X0), and use the momentary operation switch for other signals.

2.4 Test Operation

After the wiring is completed, perform a test operation with a single servo amplifier to check the rotation direction and others. Follow the procedures below to perform the test operation.

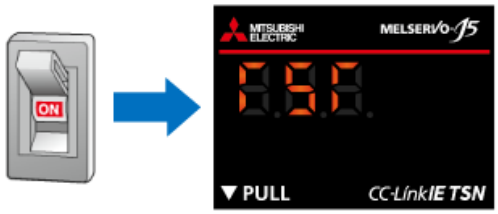
(1) Turn off the servo amplifier and PLC.
(2) Turn on the DIP switch (SW3-1) of the servo amplifier.



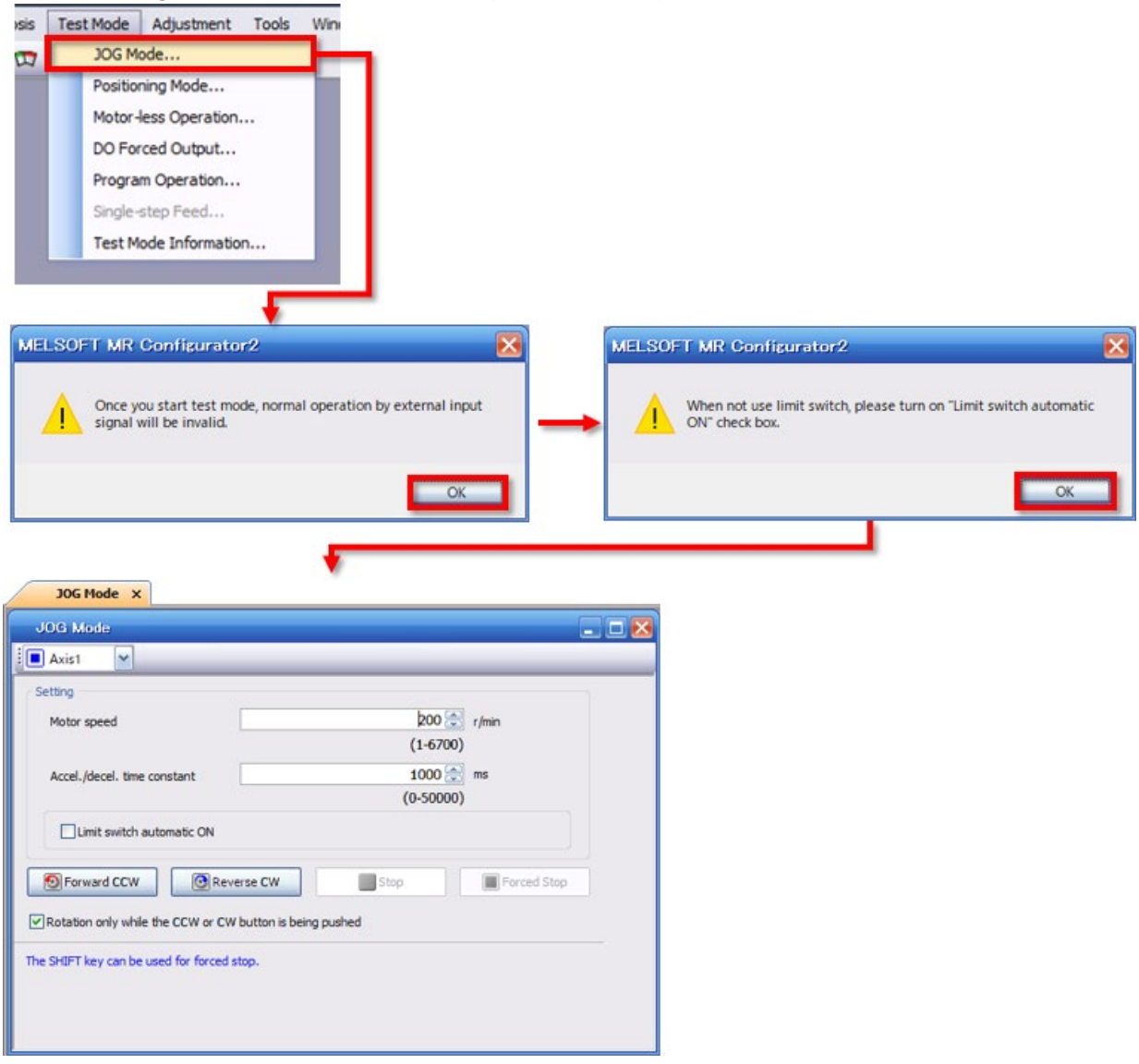
Set SW3-1 to the "ON" state.

(3) Connect the servo amplifier and a personal computer with a USB cable or Ethernet cable. (Note)
(4) Turn on the servo amplifier. "TST" is displayed on the display.

Servo amplifier



(5) Start MR Configurator2 and perform the test operation (JOG operation).



(6) Check the rotation direction and machine operation.

(7) After the test operation is completed, turn off the servo amplifier, and turn off the DIP switch (SW3-1).

(Note) When using a Ethernet cable, change the project of MR Configurator2 to a multi-axis project.

Tips
When using multiple servo amplifiers, the connection with Ethernet can eliminate the necessity of replacing cables.

2.5 Summary of This Chapter

In this chapter, you have learned:

- Device Configuration
- System Configuration
- Wiring
- Test Operation

Important points

Device Configuration	• Use a one-axis ball screw in the sample system.
System Configuration	• Connect the remote input module NZ2GN2S1-32D and servo amplifier MR-J5-G to the motion module RD78G4.
Wiring	• Use a one-cable type option cable for the servo motor. • Set the fourth octet of the IP addresses of the remote input module and servo amplifier with the rotary switches. • Connect the proximity dog signal, limit switches, and forced stop switch to the servo amplifier. • Connect operation command switches to the remote input module.
Test Operation	• Change the DIP switch of the servo amplifier, and connect it to a personal computer. • Check the rotation direction of the motor and machine operation using the test operation function of MR Configurator2.

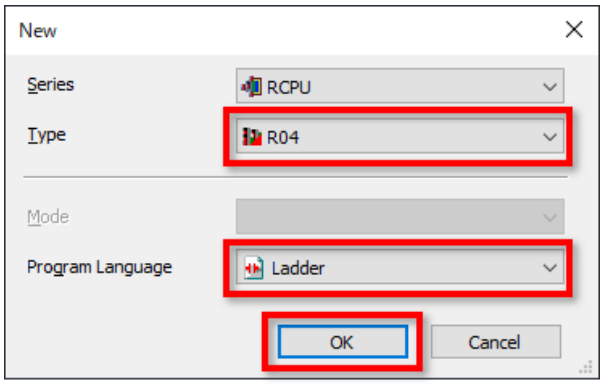
Chapter 3 Project Creation

In this chapter, you will learn how to create projects required for operating the motion module by using the sample program.
Start GX Works3 and operate it according to the screen.
Or, download the following sample program and check the settings.

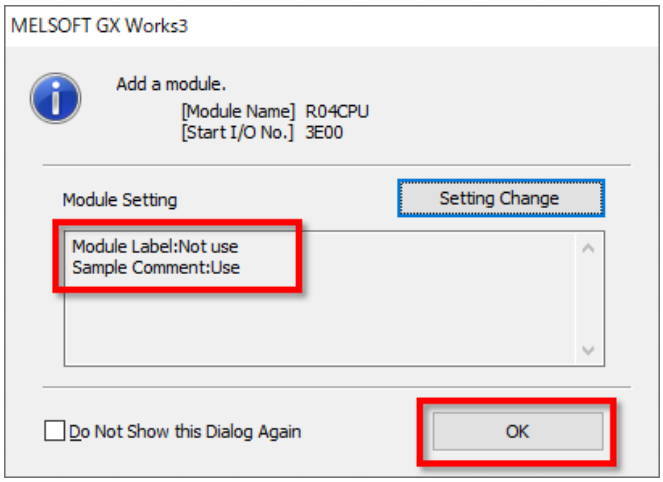
★Sample_RD78GBasic_en.zip(1.21MB) GX Works3 Ver.1.072A or later is required.

3.1 Creating a New Project

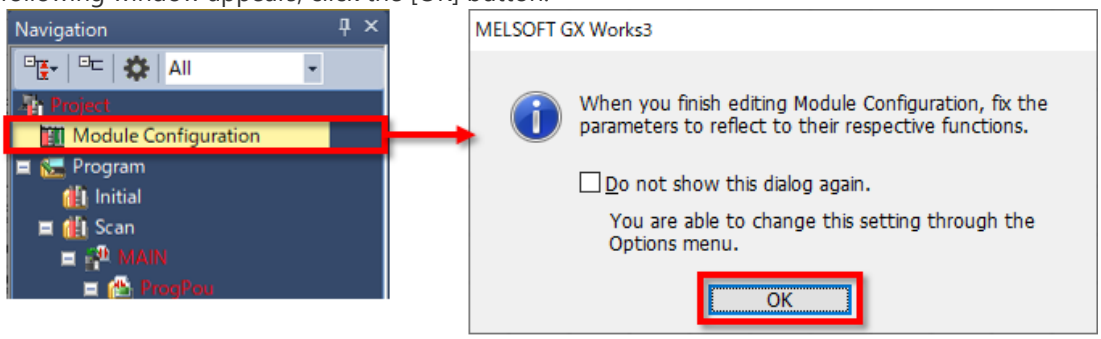
(1) Select [Project] → [New] in GX Works3.
Select the model of a PLC CPU to be used and a program language to be used in the PLC CPU on the following window.
In the sample program, the model is set to R04CPU, and the program language is set to Ladder.
After the selection is completed, click the [OK] button.



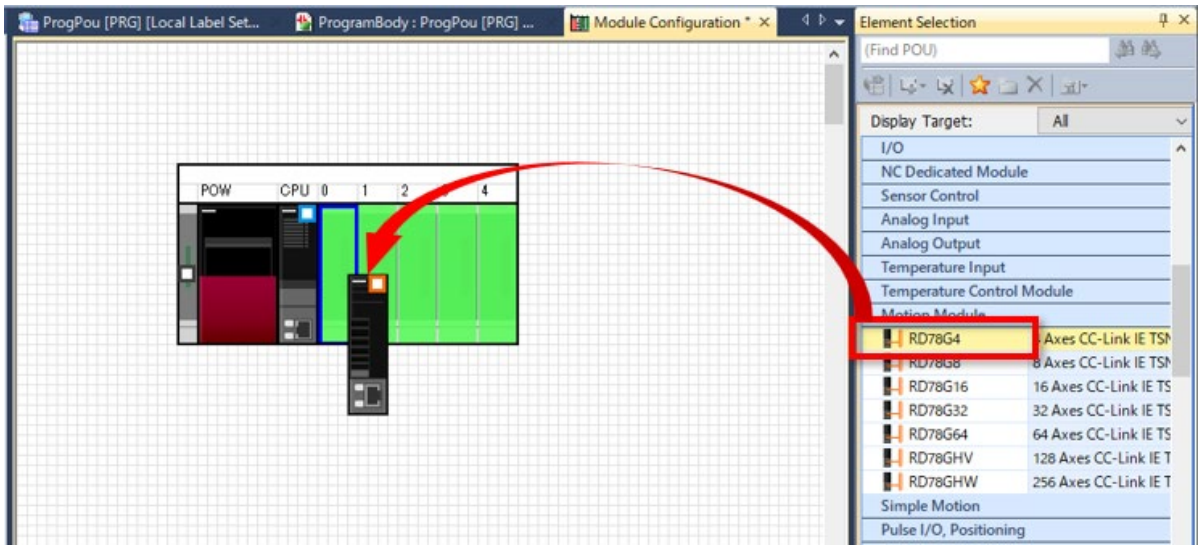
(2) When the following window appears, set whether to use the module label and sample comment.
To change the setting, click the [Setting Change] button.
Click the [OK] button to open the project.



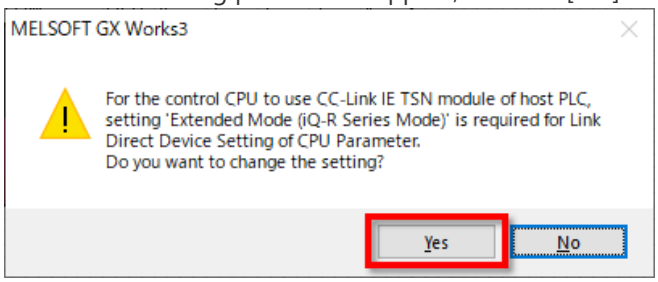
(3) Double-click [Module Configuration] in the project tree. When the following window appears, click the [OK] button.



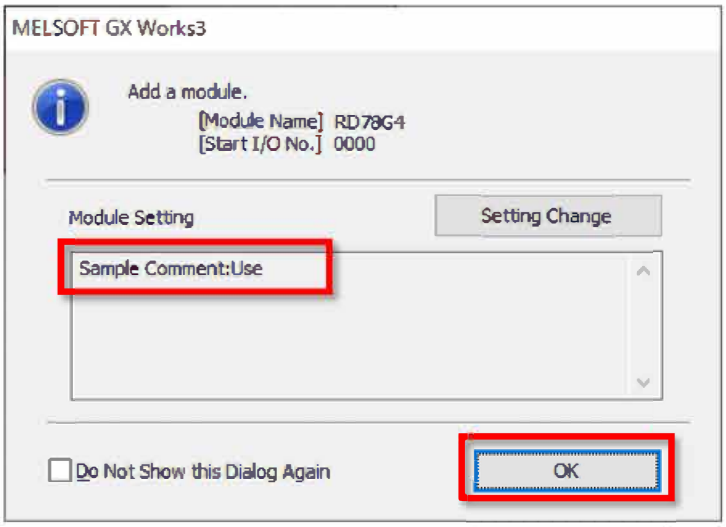
When the Module Configuration screen is opened, drag and drop a module to be used (base module, power supply module, and motion module) from the [Element Selection] window displayed on the right, and create a module configuration diagram as the one shown in Section 2.2.



After creating the module configuration diagram, right-click the screen, and select [Parameter] → [Fix].
When the following precautions appear, click the [Yes] button.

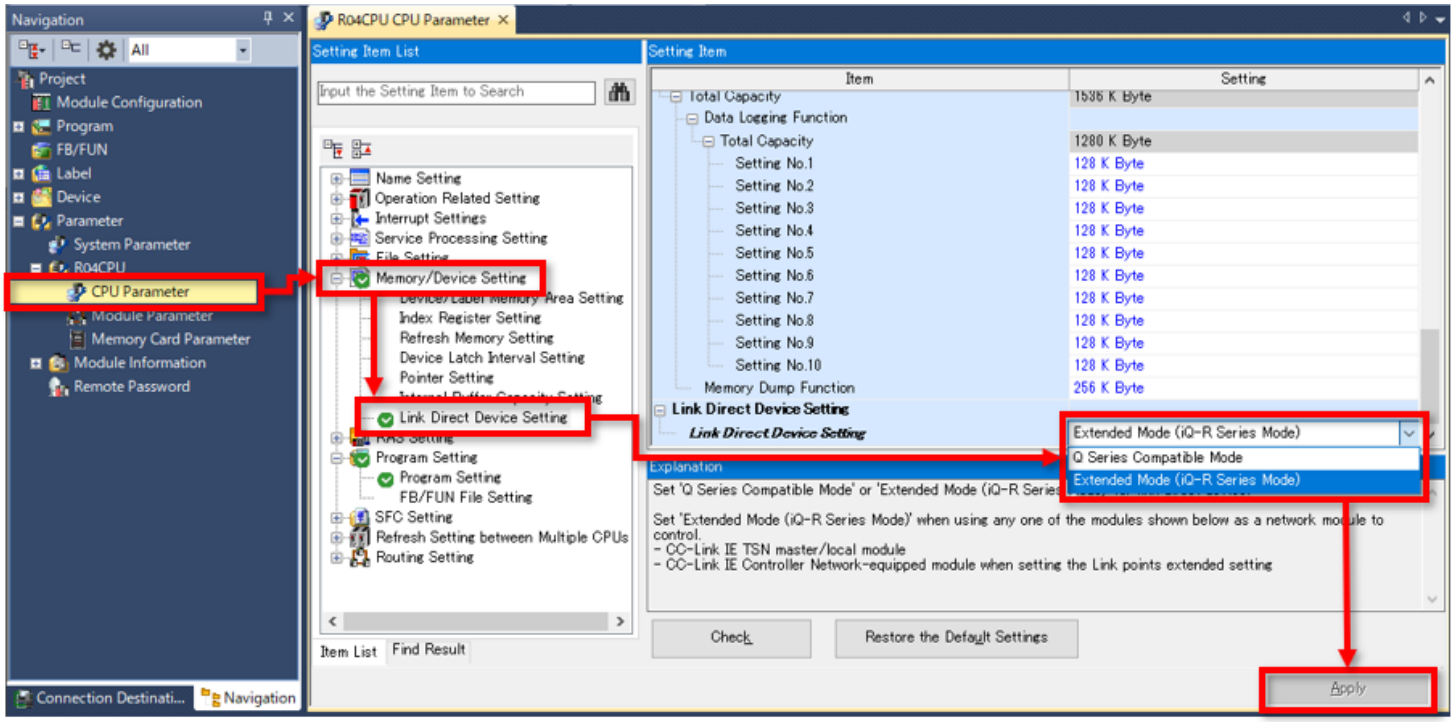


When the following window appears, check that the sample comment is set to [Use].
When [Not use] is set, click the [Setting Change] button, and change the setting on the displayed window.
Click the [OK] button to complete.


MELSOFT GX Works3
Add a module.
[Module Name] RD78G4
[Start I/O No.] 0000
Module Setting
Setting Change
Sample Comment:Use
Do Not Show this Dialog Again
OK

3.2 PLC CPU Settings

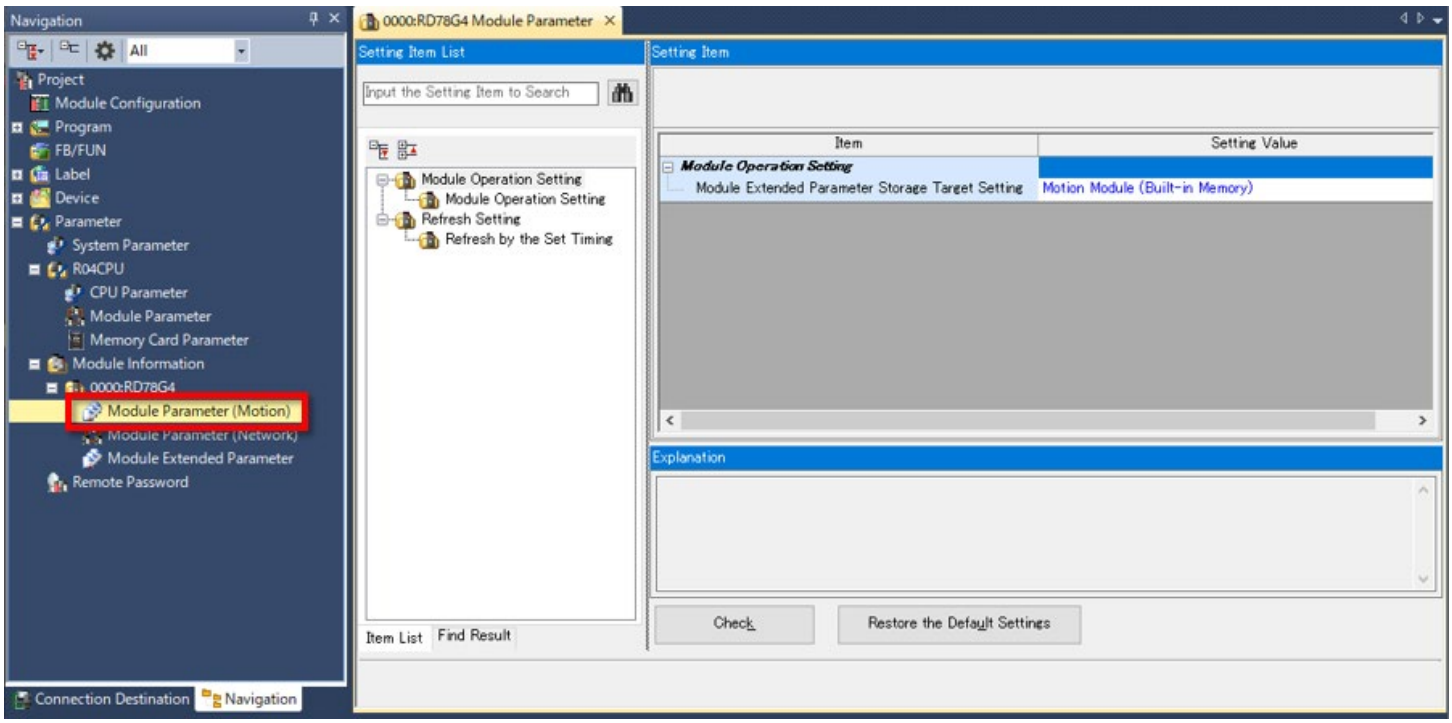
Double-click [Parameter] → [R04CPU] → [CPU Parameter] in the project tree.
Click [Link Direct Device Setting] in the setting item list.
Operate the drop-down list, and check that the link direct device setting is set to [Extended Mode (iQ-R Series Mode)].
If [Q Series Compatible Mode] is set, change it to [Extended Mode (iQ-R Series Mode)].
After the setting is completed, click the [Apply] button on the lower right.



3.3 Motion Module Setting

3.3.1 Module parameter (Motion)

Double-click [Parameter] → [Module Information] → [0000:RD78G4] → [Module Parameter (Motion)] in the project tree.
In the module operation setting, the storage destination of the module expansion parameters can be selected from a built-in memory or SD card (refer to 3.3.3 and 3.4).
In the refresh setting, set the timing to refresh the devices.
In this course, keep the default settings for both.

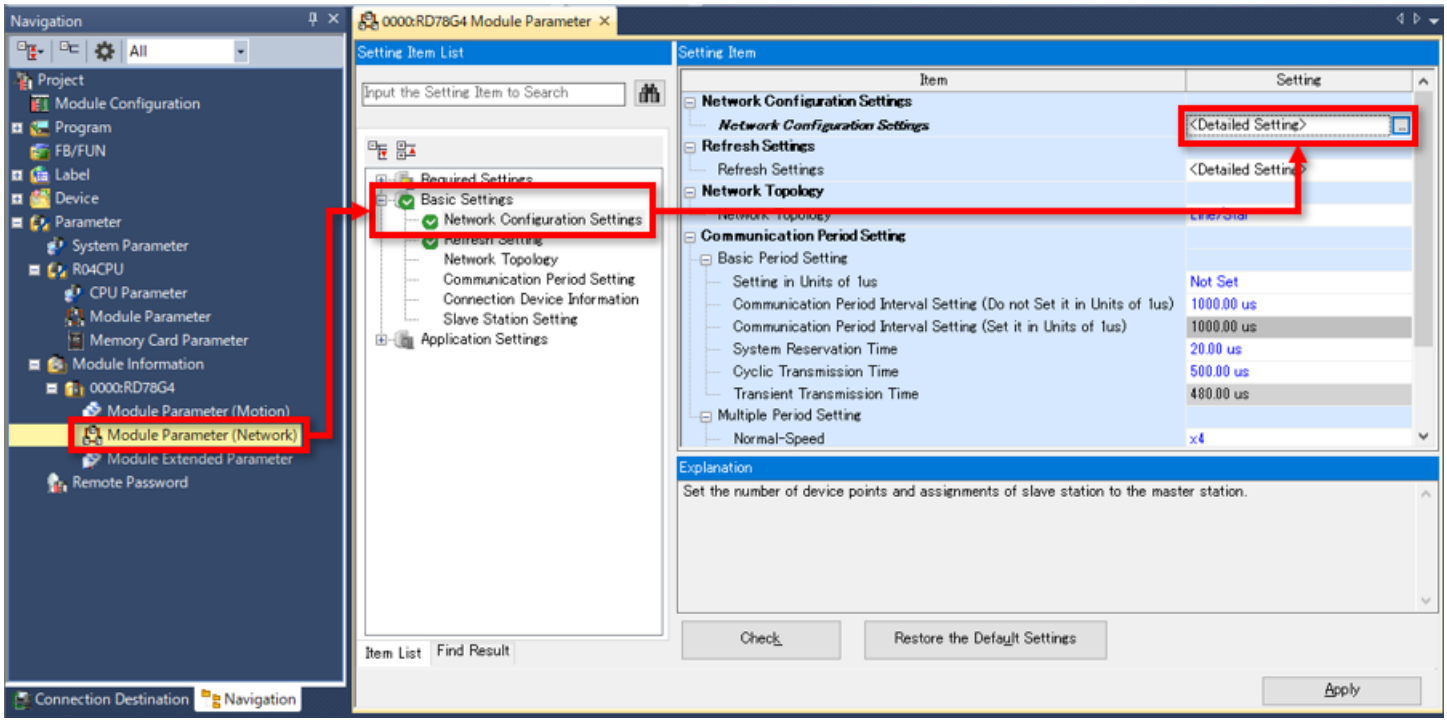


3.3.2 Module parameter (Network)

Double-click [Parameter] → [Module Information] → [0000:RD78G4] → [Module Parameter (Network)] in the project tree.
In this section, configure the settings for devices to be connected to the network and a link refresh.

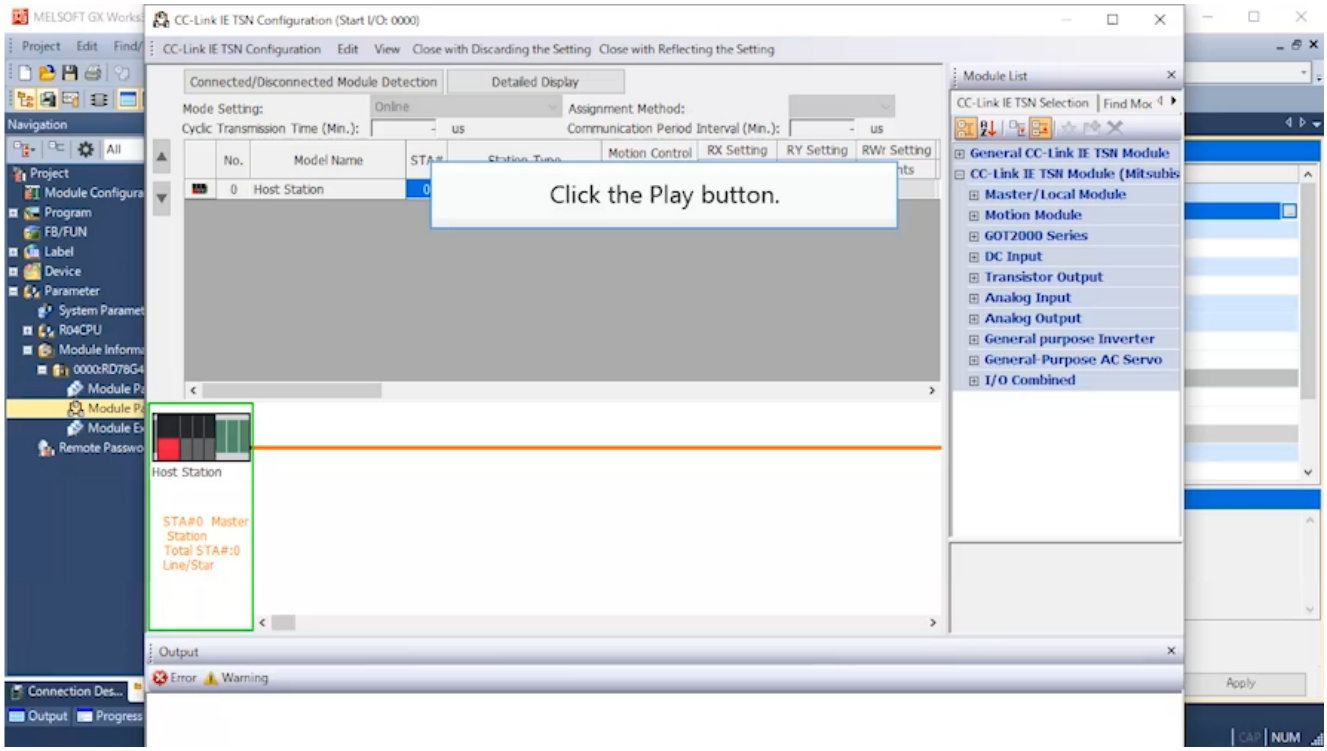
(1) Network configuration setting

Select [Basic Settings] in the setting item list, and double-click<Detailed Setting>in Network Configuration Settings.



3.3.2 Module parameter (Network)

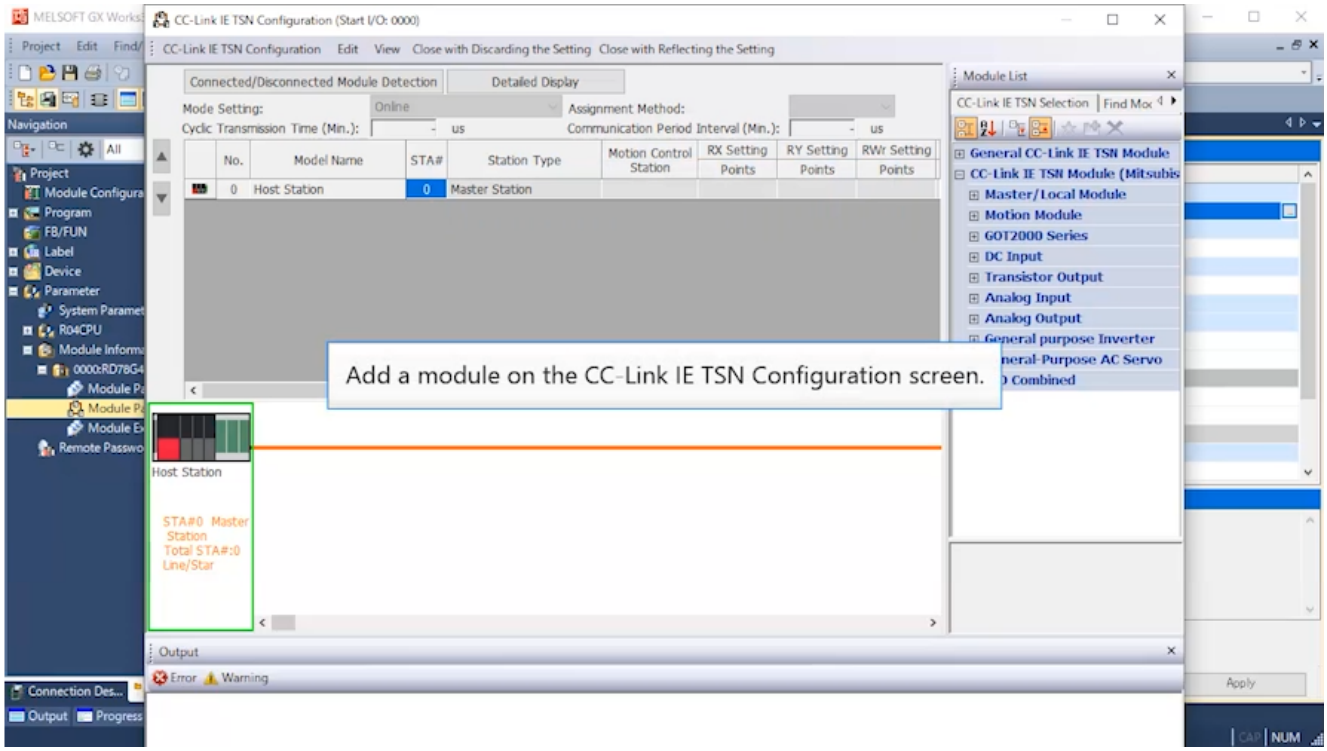
(2) Adding a module



* If NZ2GN2S1-32D or MR-J5-G is not displayed in the module list on the right side of the screen, download the profile data (CSP+ file) from **here**, and register it to GX Works3.

3.3.2 Module parameter (Network)

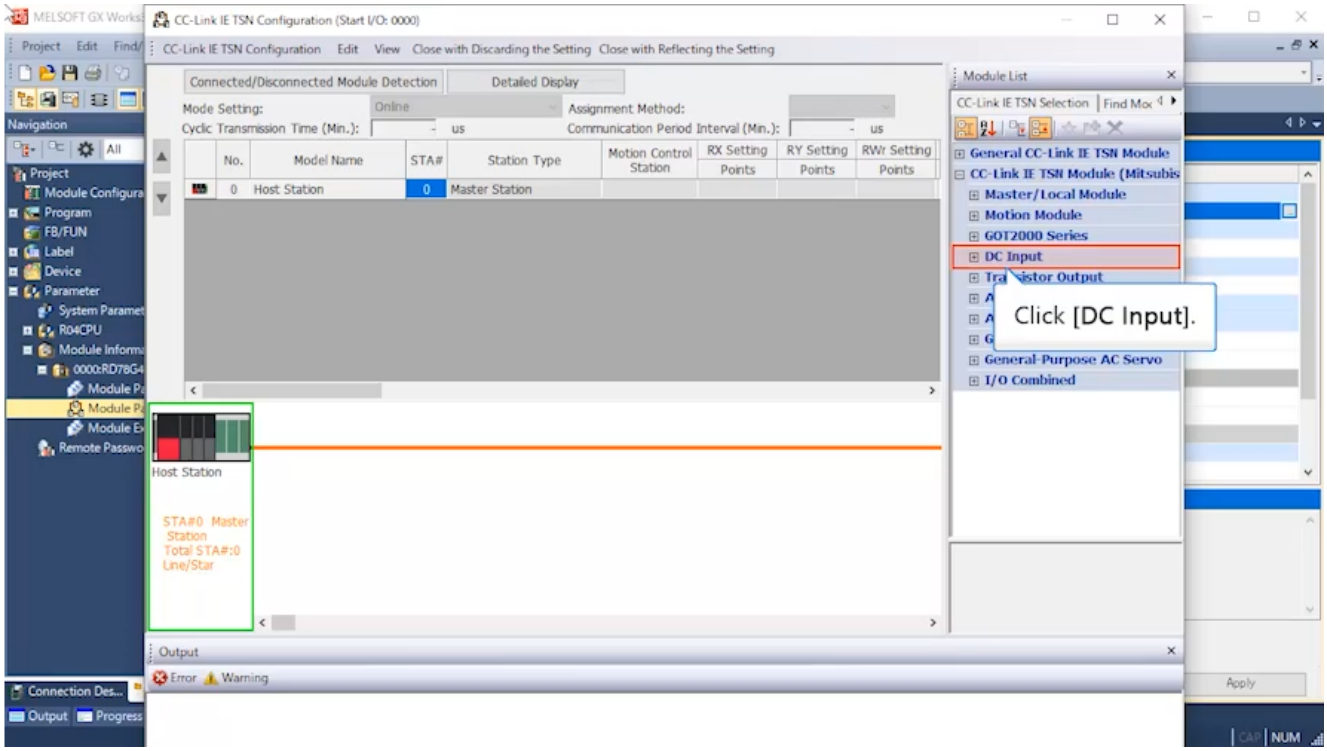
(2) Adding a module



* If NZ2GN2S1-32D or MR-J5-G is not displayed in the module list on the right side of the screen, download the profile data (CSP+ file) from **here**, and register it to GX Works3.

3.3.2 Module parameter (Network)

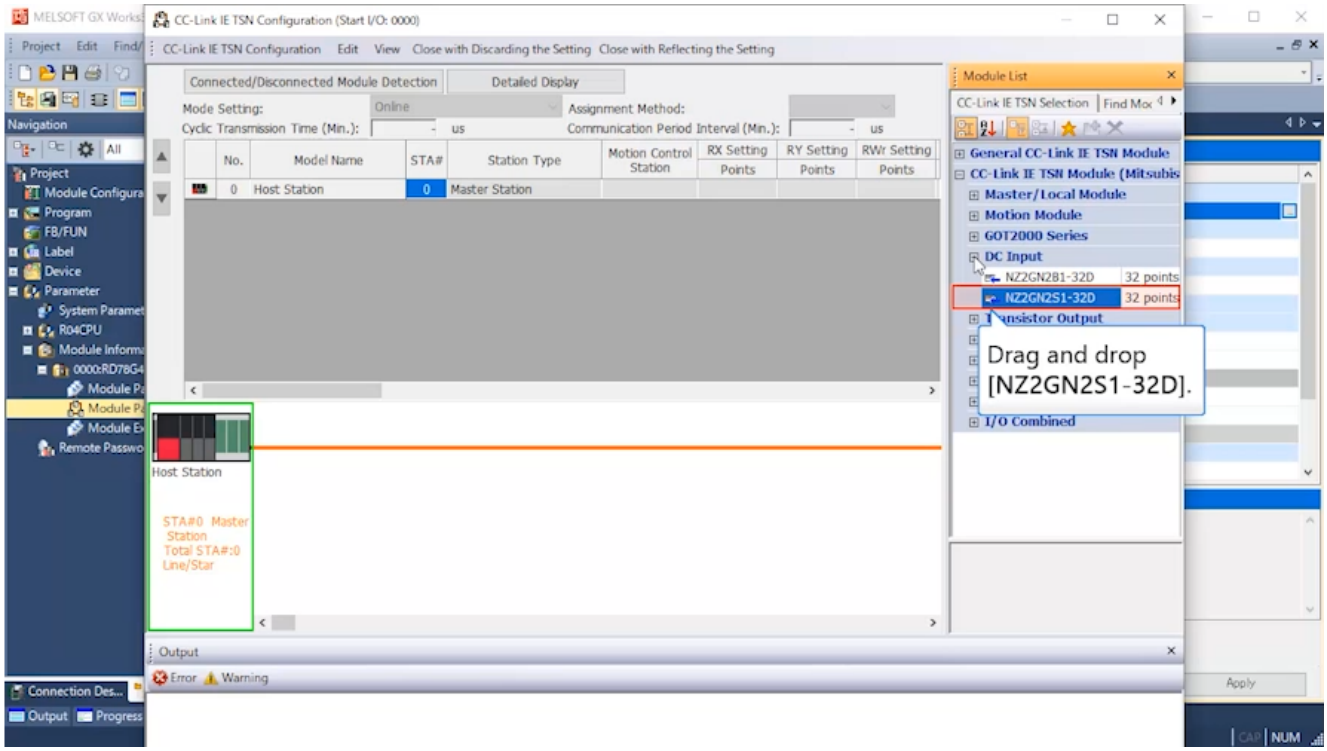
(2) Adding a module



* If NZ2GN2S1-32D or MR-J5-G is not displayed in the module list on the right side of the screen, download the profile data (CSP+ file) from **here**, and register it to GX Works3.

3.3.2 Module parameter (Network)

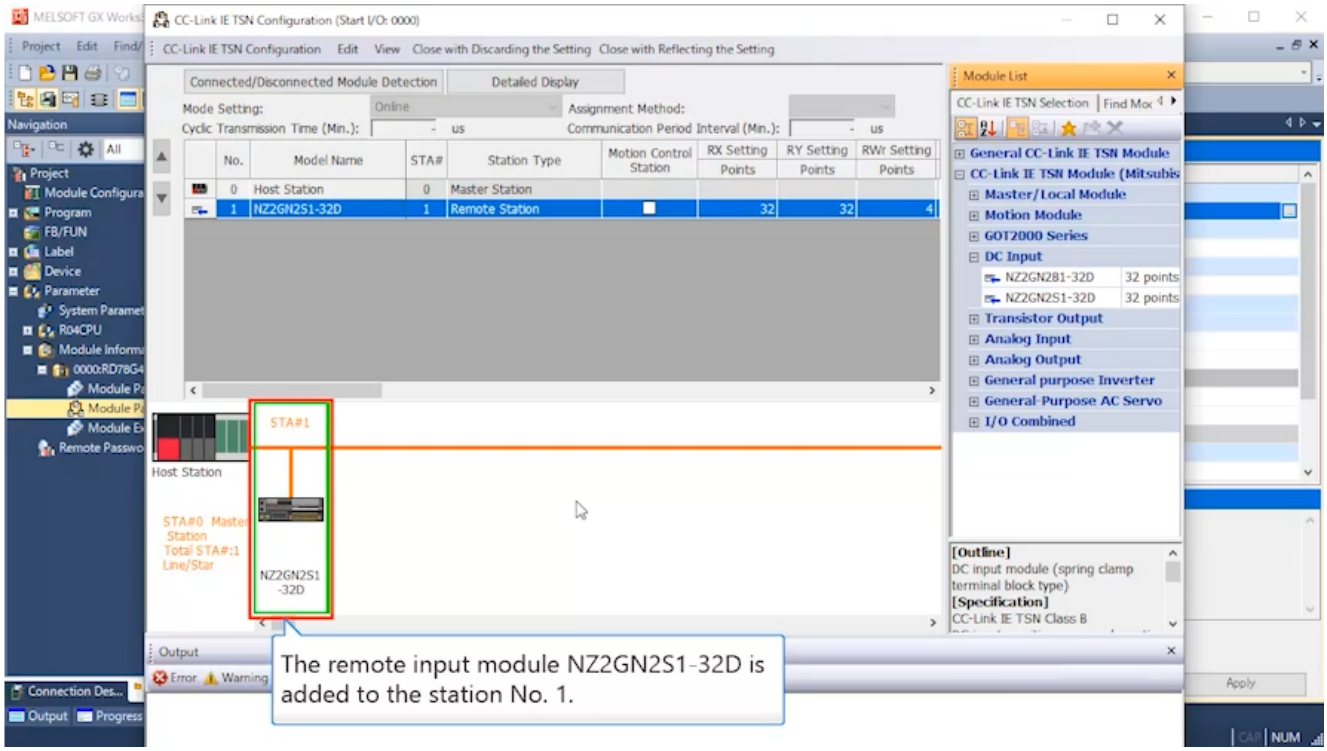
(2) Adding a module



* If NZ2GN2S1-32D or MR-J5-G is not displayed in the module list on the right side of the screen, download the profile data (CSP+ file) from **here**, and register it to GX Works3.

3.3.2 Module parameter (Network)

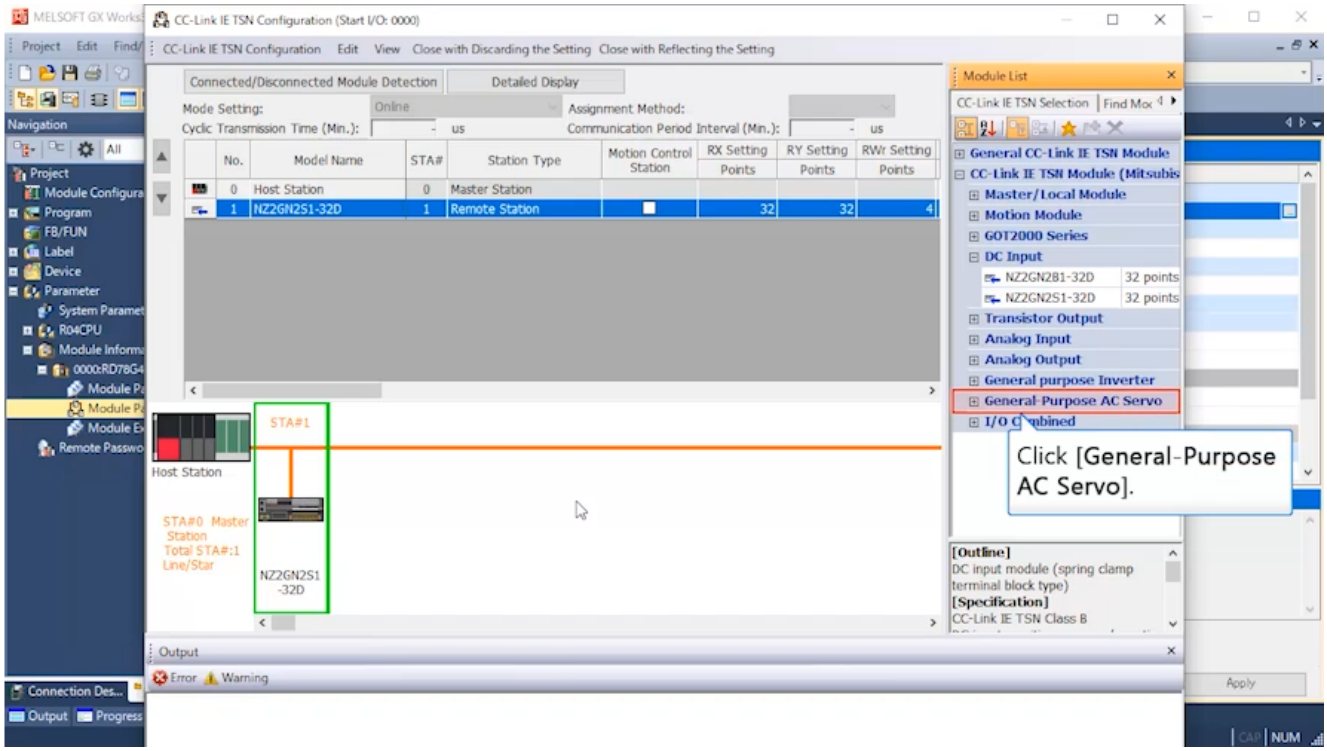
(2) Adding a module



* If NZ2GN2S1-32D or MR-J5-G is not displayed in the module list on the right side of the screen, download the profile data (CSP+ file) from **here**, and register it to GX Works3.

3.3.2 Module parameter (Network)

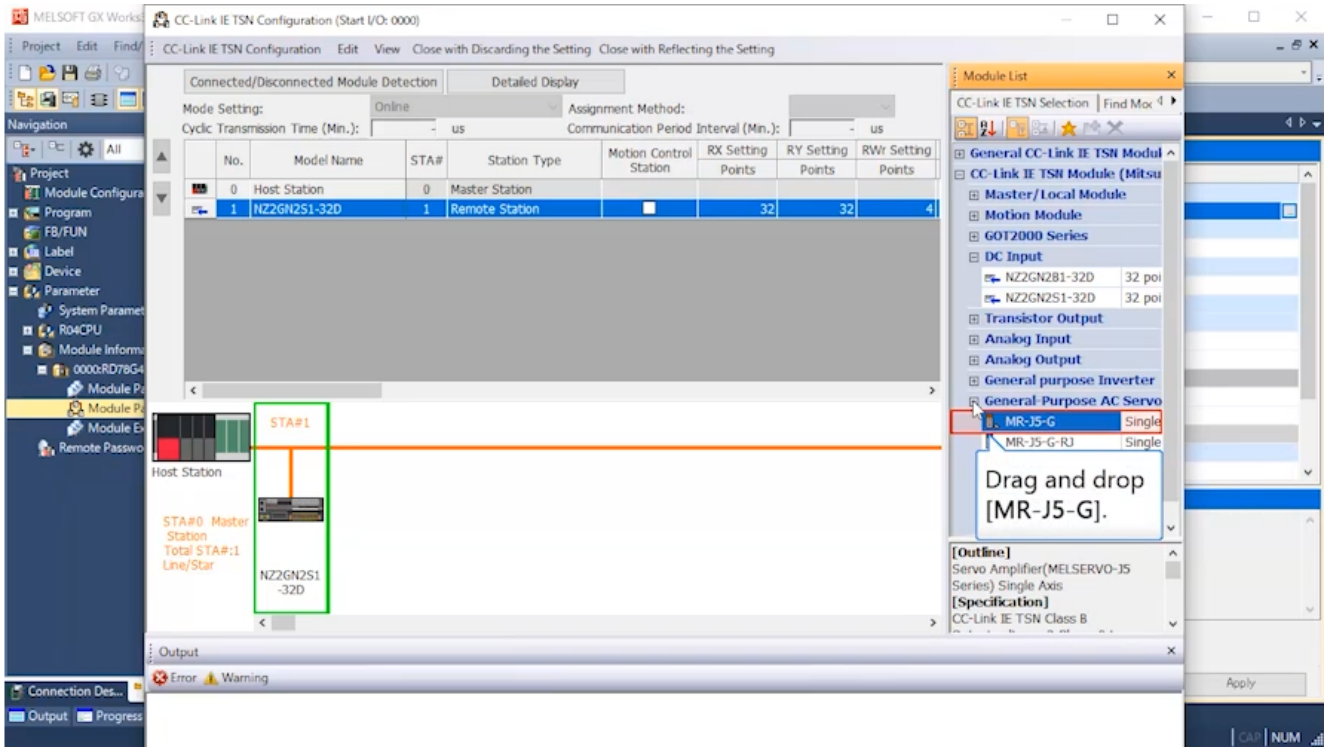
(2) Adding a module



* If NZ2GN2S1-32D or MR-J5-G is not displayed in the module list on the right side of the screen, download the profile data (CSP+ file) from **here**, and register it to GX Works3.

3.3.2 Module parameter (Network)

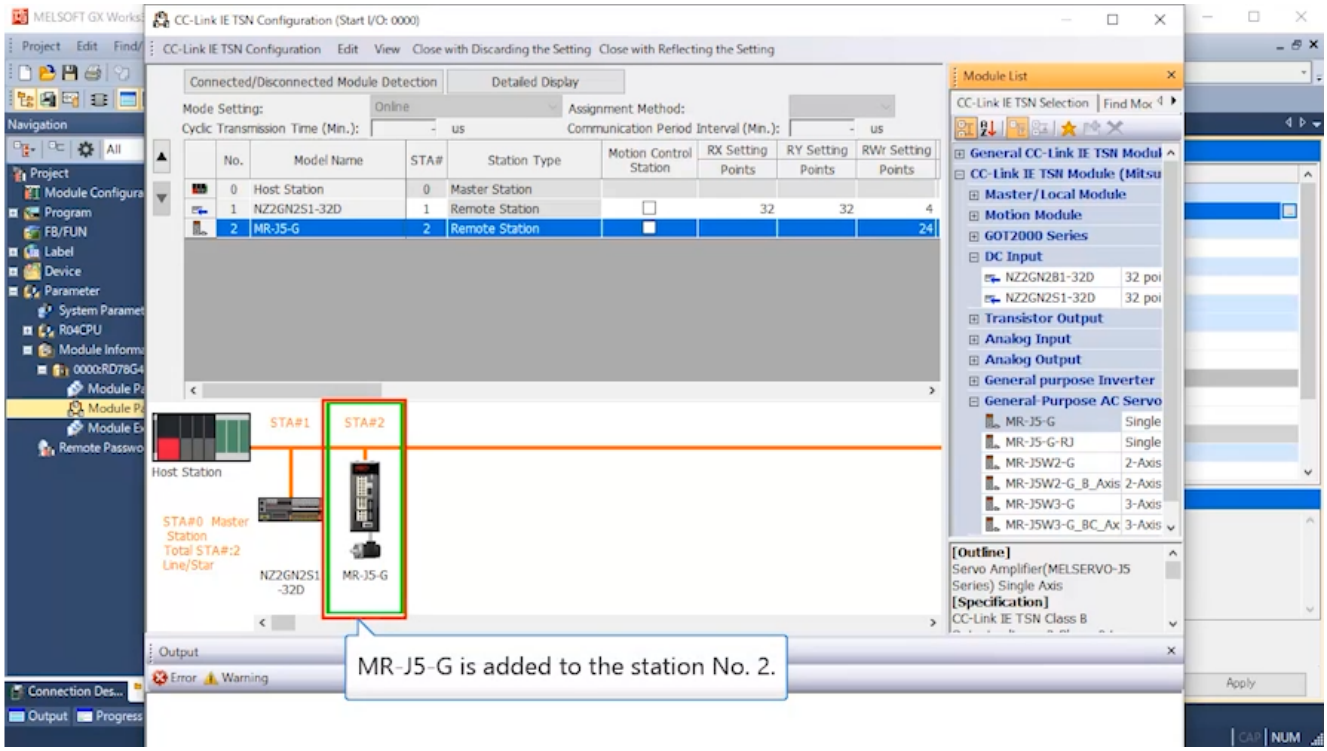
(2) Adding a module



* If NZ2GN2S1-32D or MR-J5-G is not displayed in the module list on the right side of the screen, download the profile data (CSP+ file) from **here**, and register it to GX Works3.

3.3.2 Module parameter (Network)

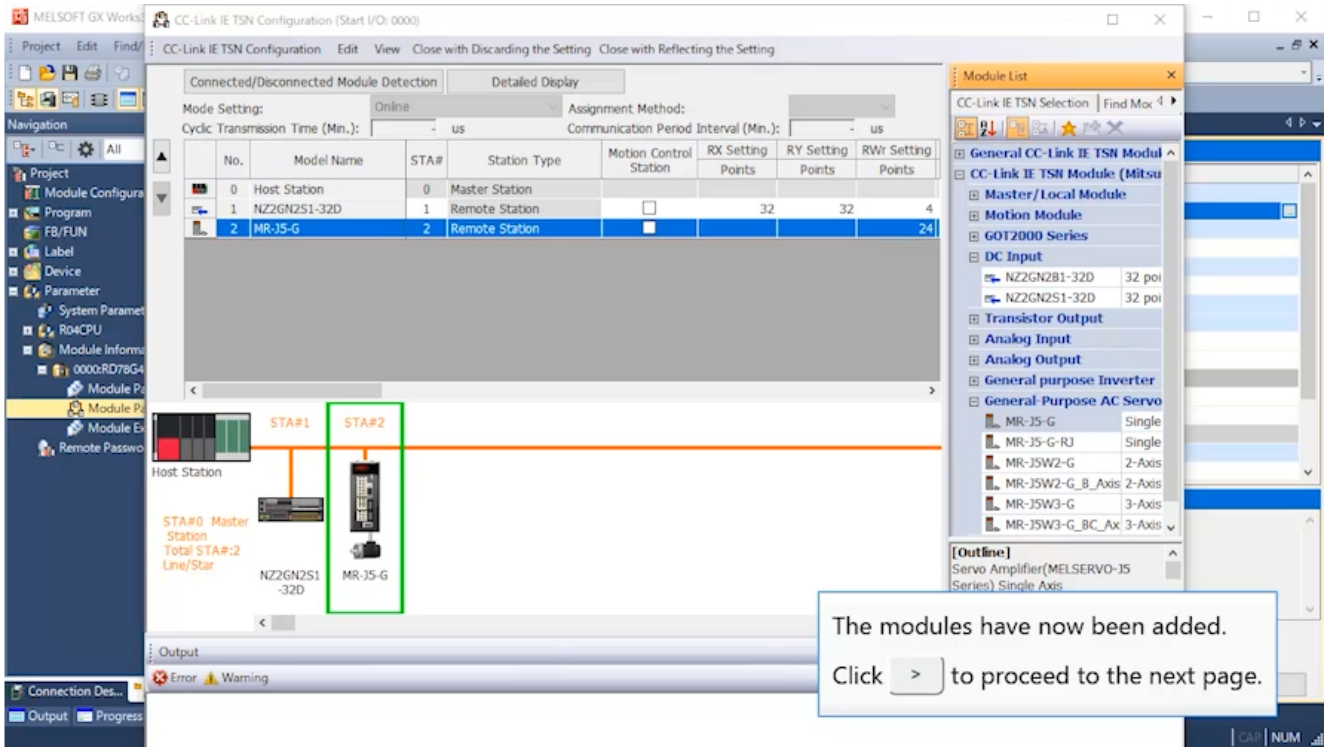
(2) Adding a module



* If NZ2GN2S1-32D or MR-J5-G is not displayed in the module list on the right side of the screen, download the profile data (CSP+ file) from **here**, and register it to GX Works3.

3.3.2 Module parameter (Network)

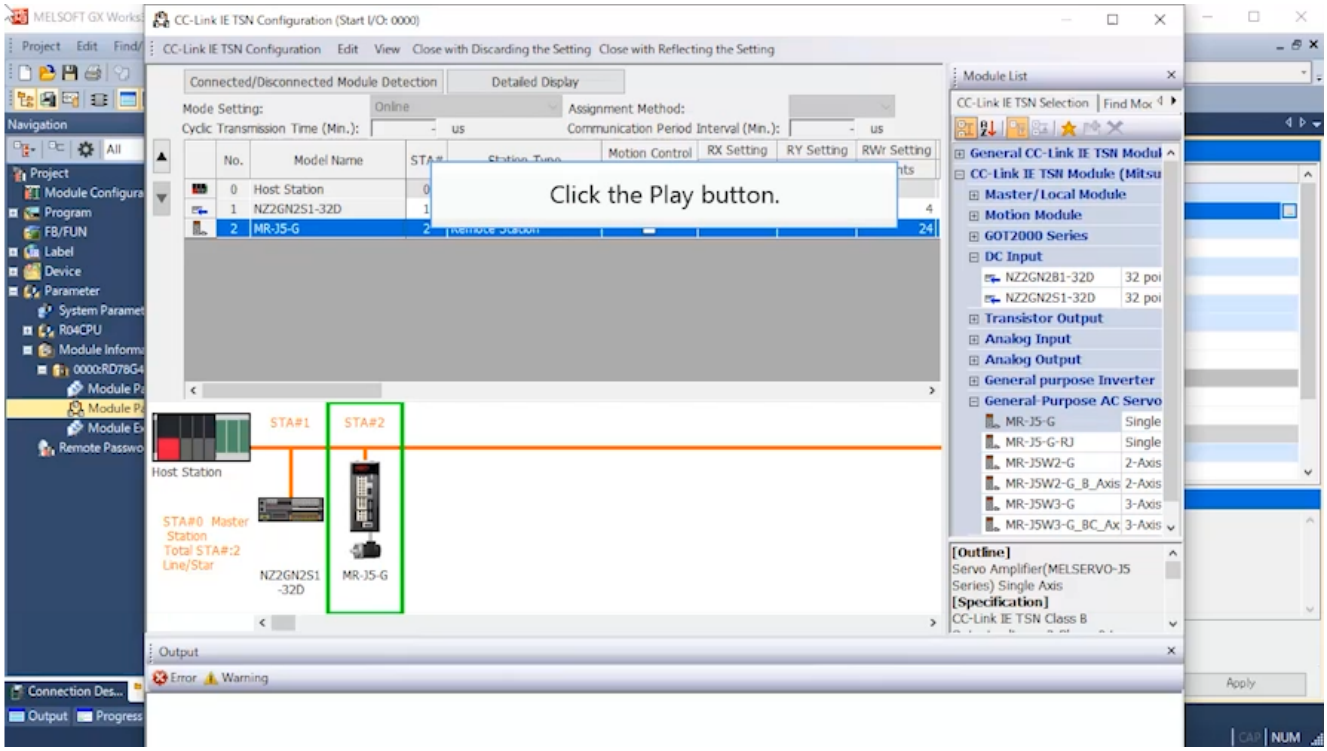
(2) Adding a module



* If NZ2GN2S1-32D or MR-J5-G is not displayed in the module list on the right side of the screen, download the profile data (CSP+ file) from **here**, and register it to GX Works3.

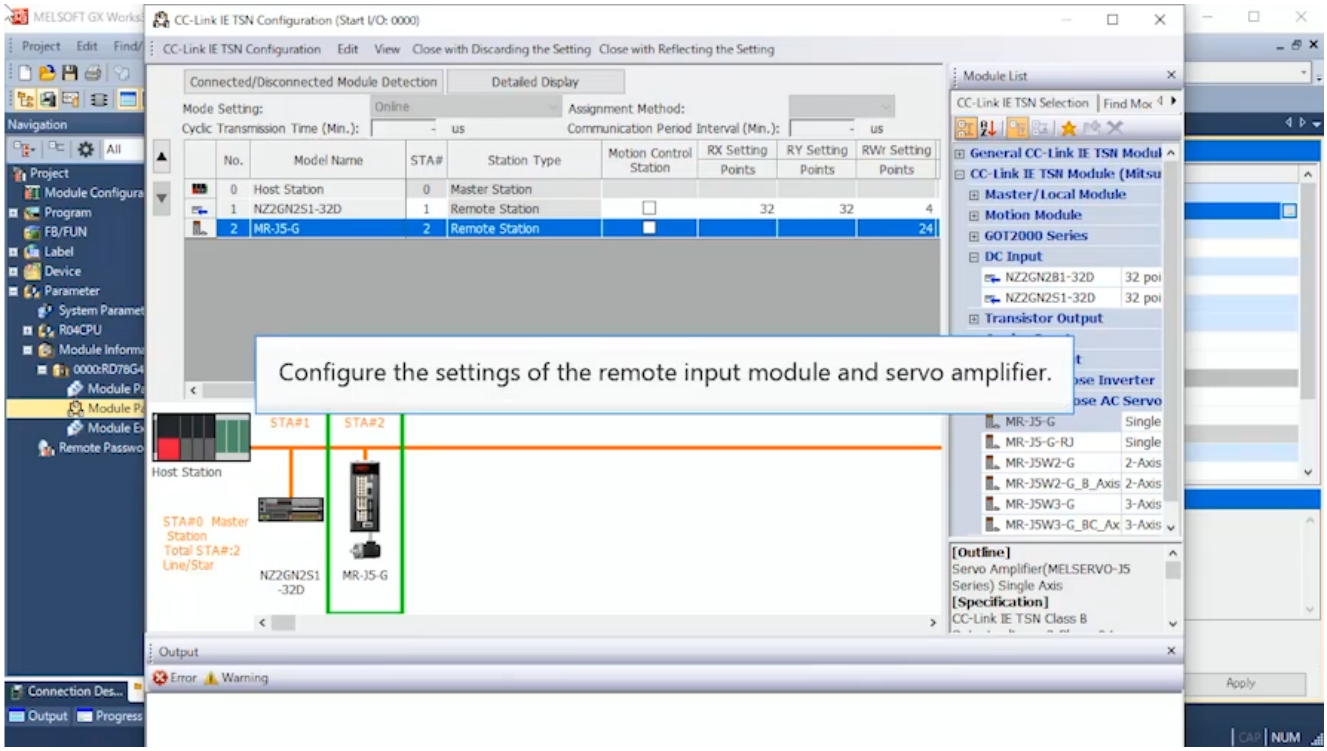
3.3.2 Module parameter (Network)

(3)Remote station setting



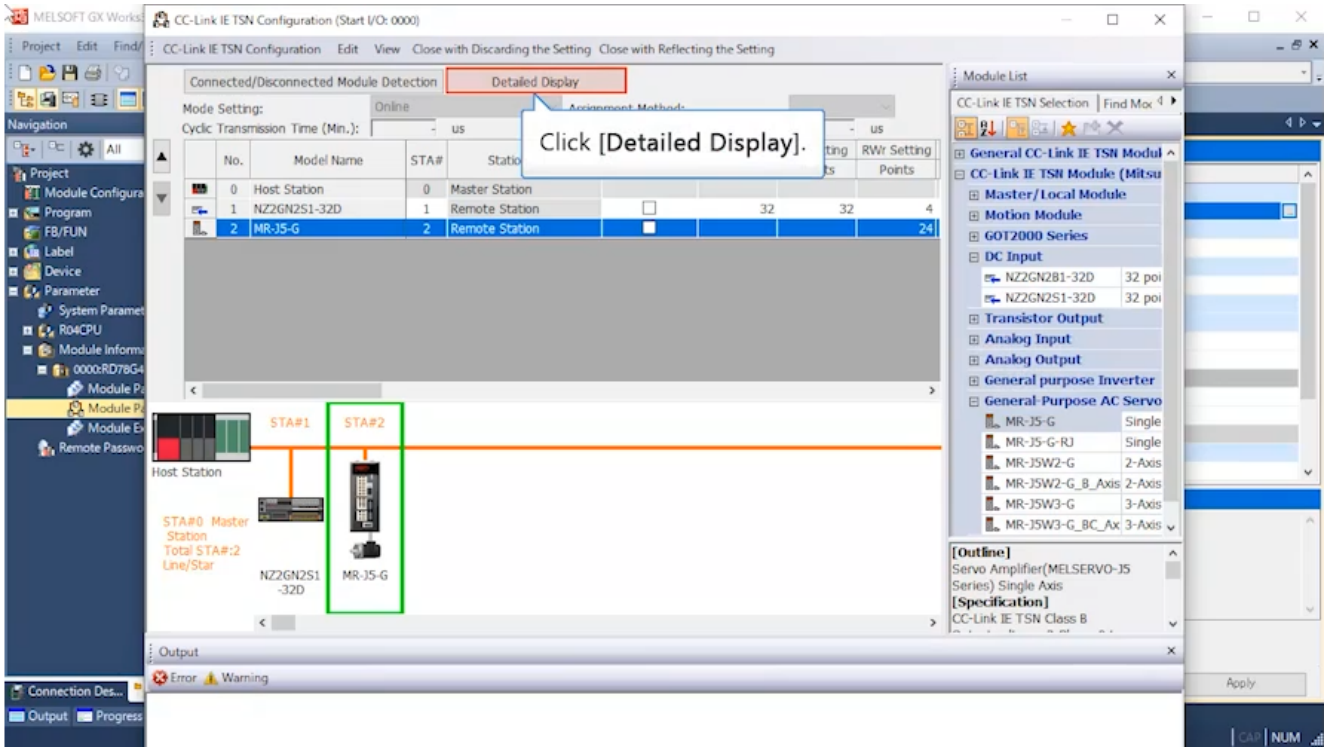
3.3.2 Module parameter (Network)

(3)Remote station setting



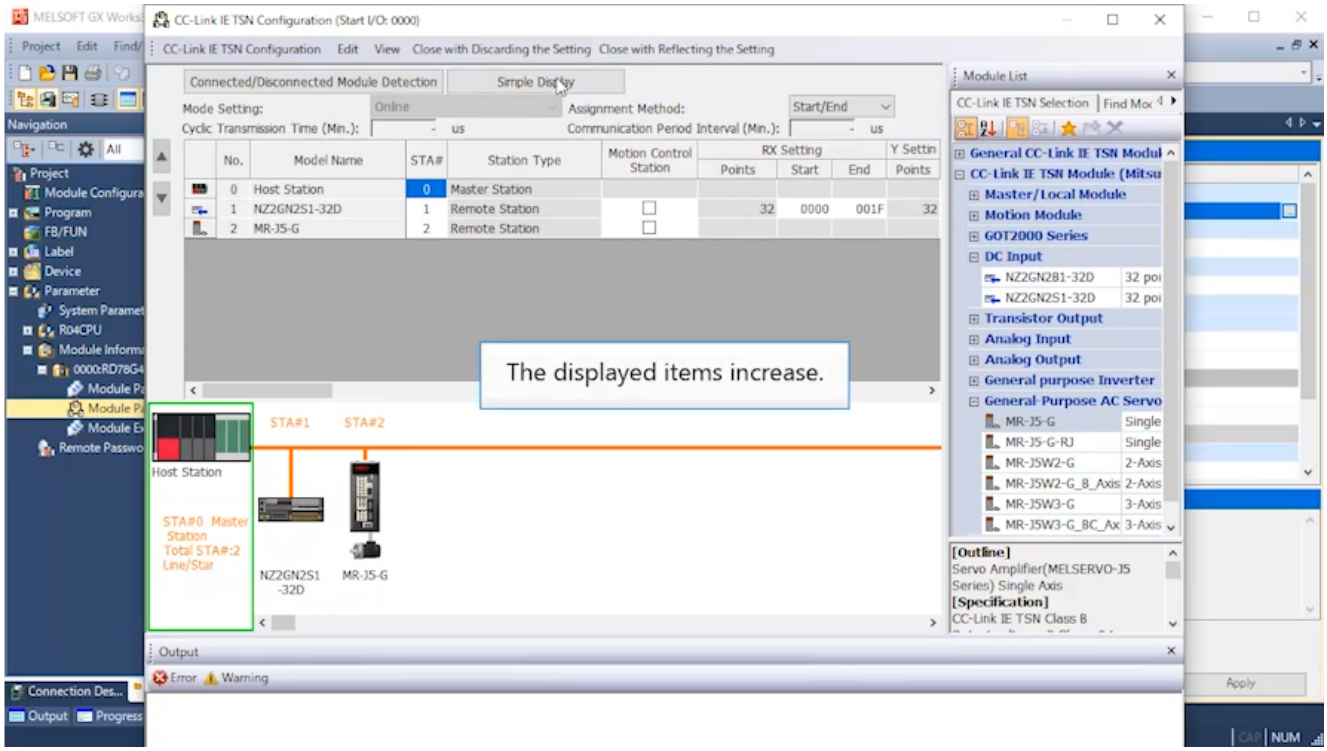
3.3.2 Module parameter (Network)

(3)Remote station setting



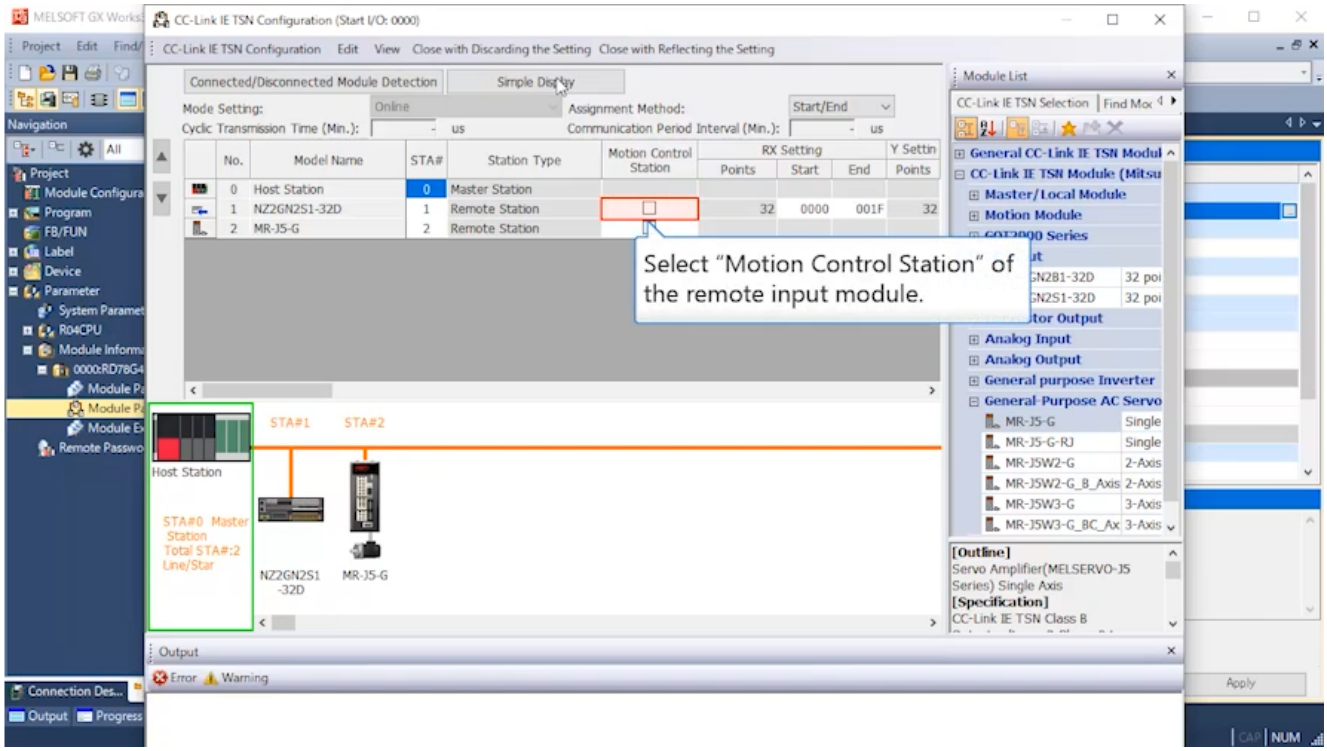
3.3.2 Module parameter (Network)

(3)Remote station setting



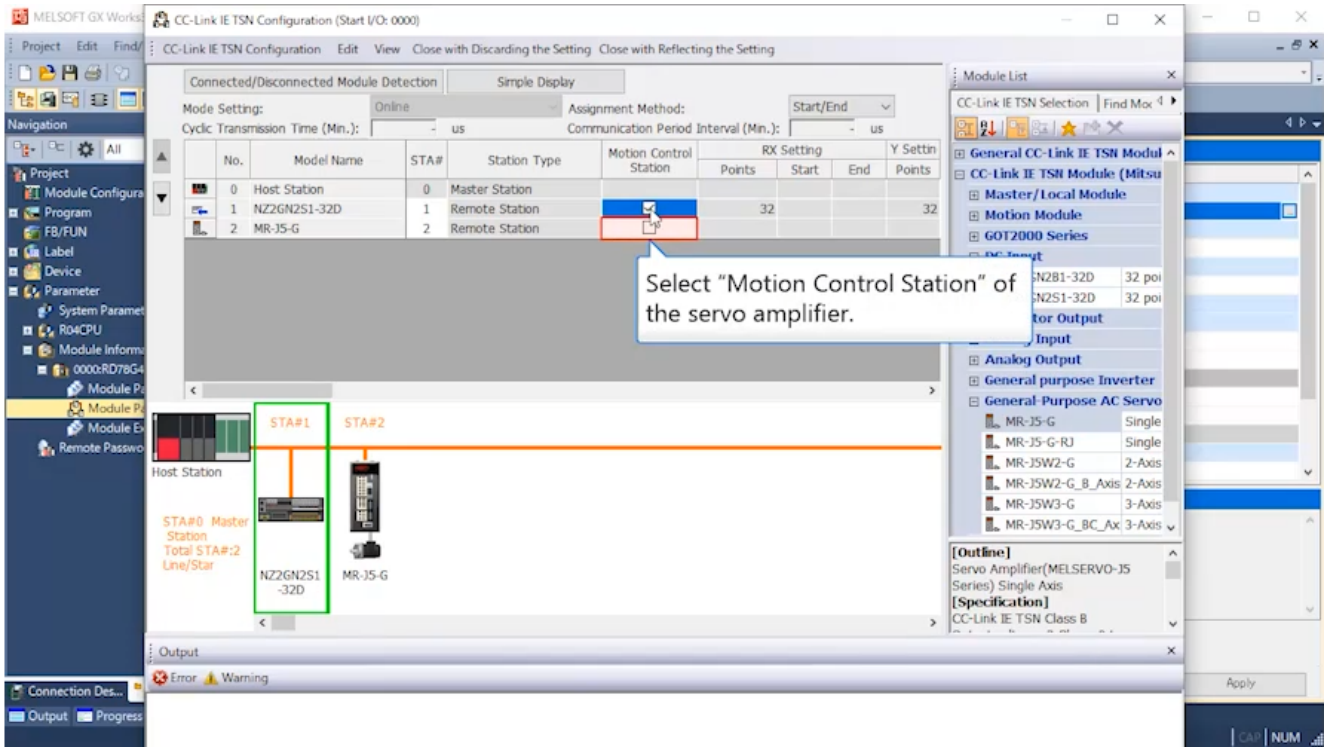
3.3.2 Module parameter (Network)

(3)Remote station setting



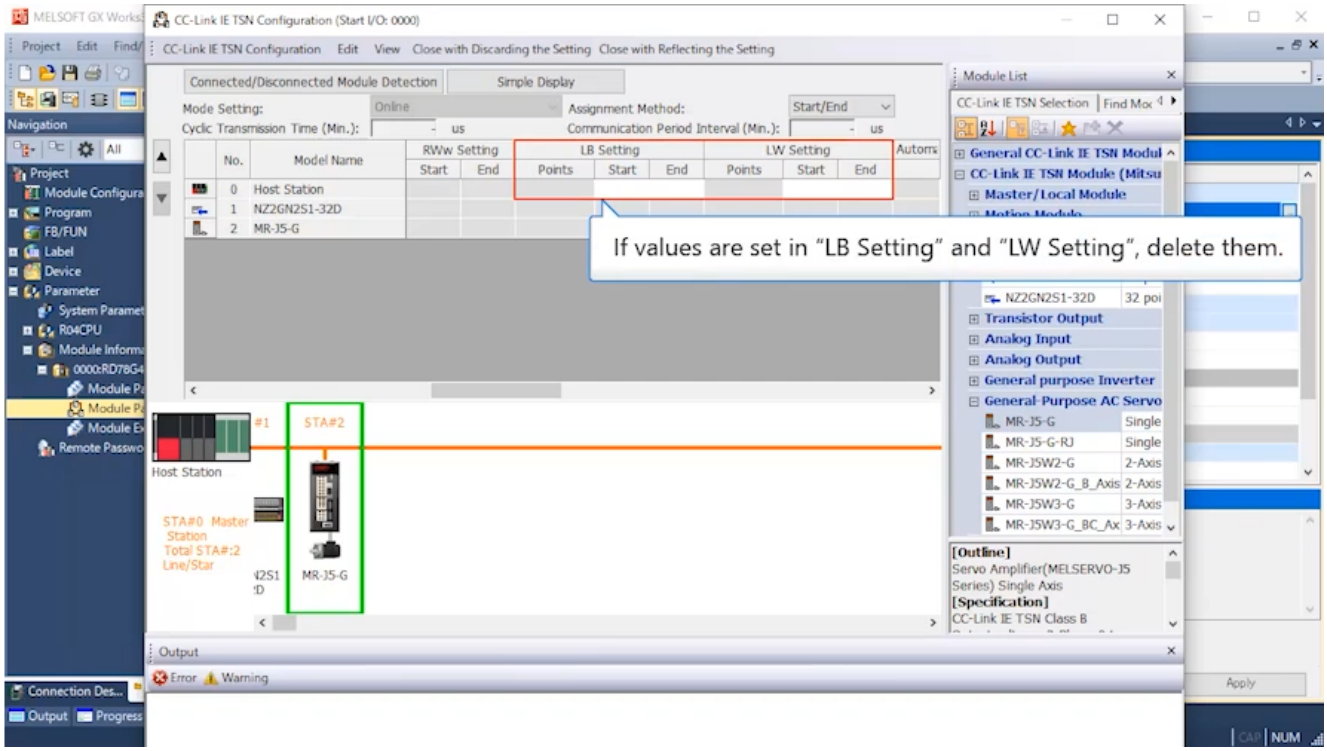
3.3.2 Module parameter (Network)

(3)Remote station setting



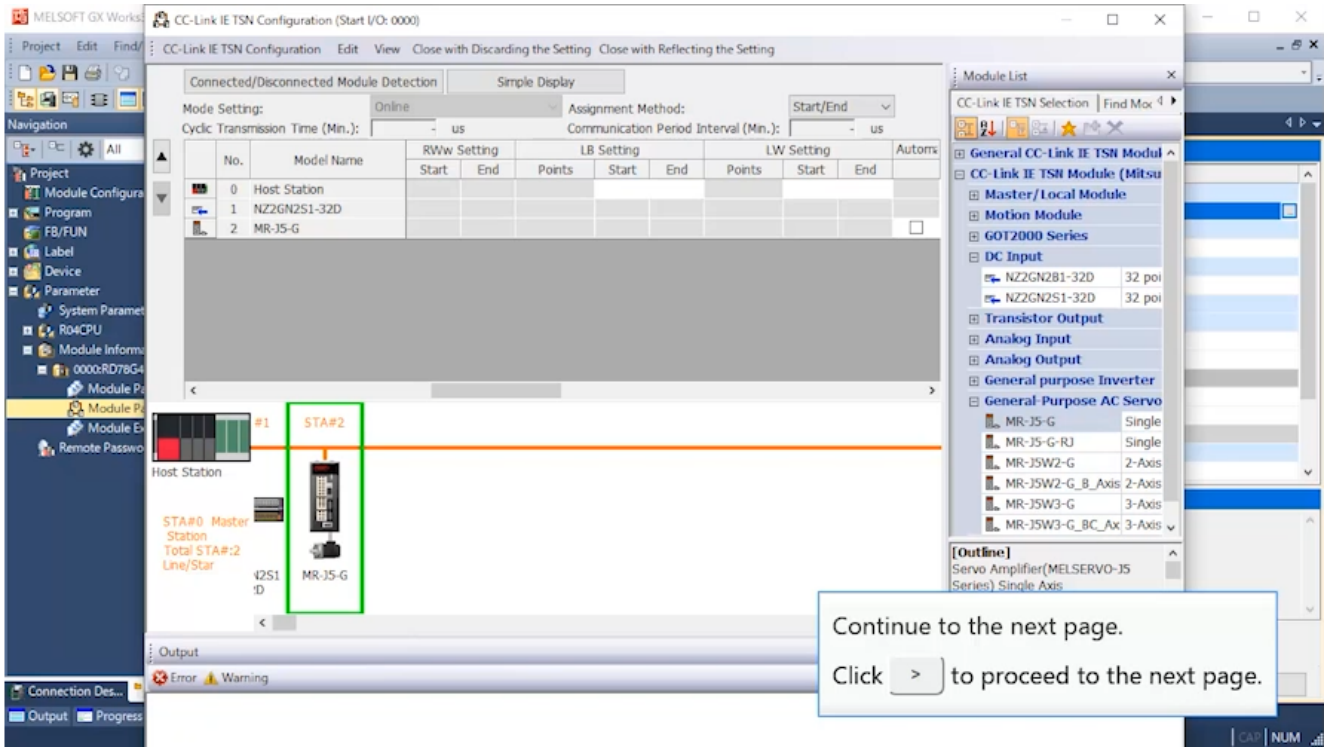
3.3.2 Module parameter (Network)

(3)Remote station setting



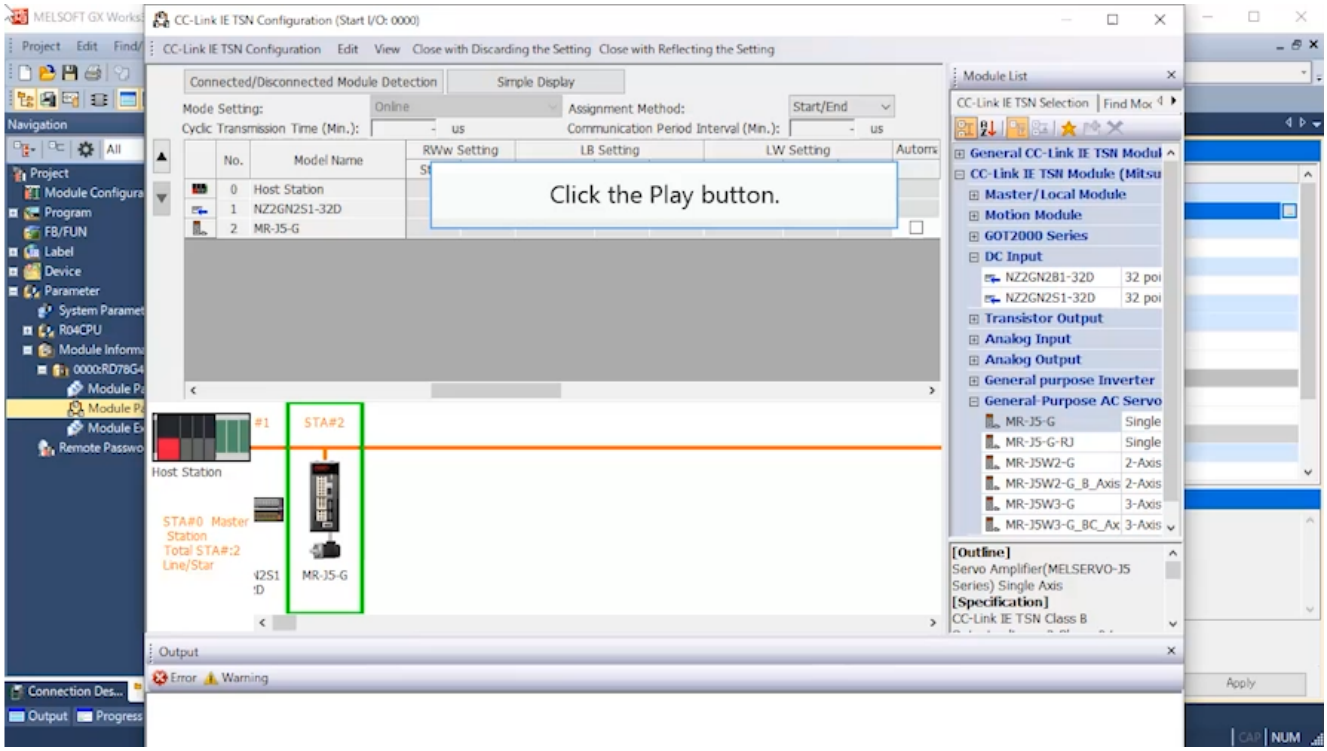
3.3.2 Module parameter (Network)

(3)Remote station setting



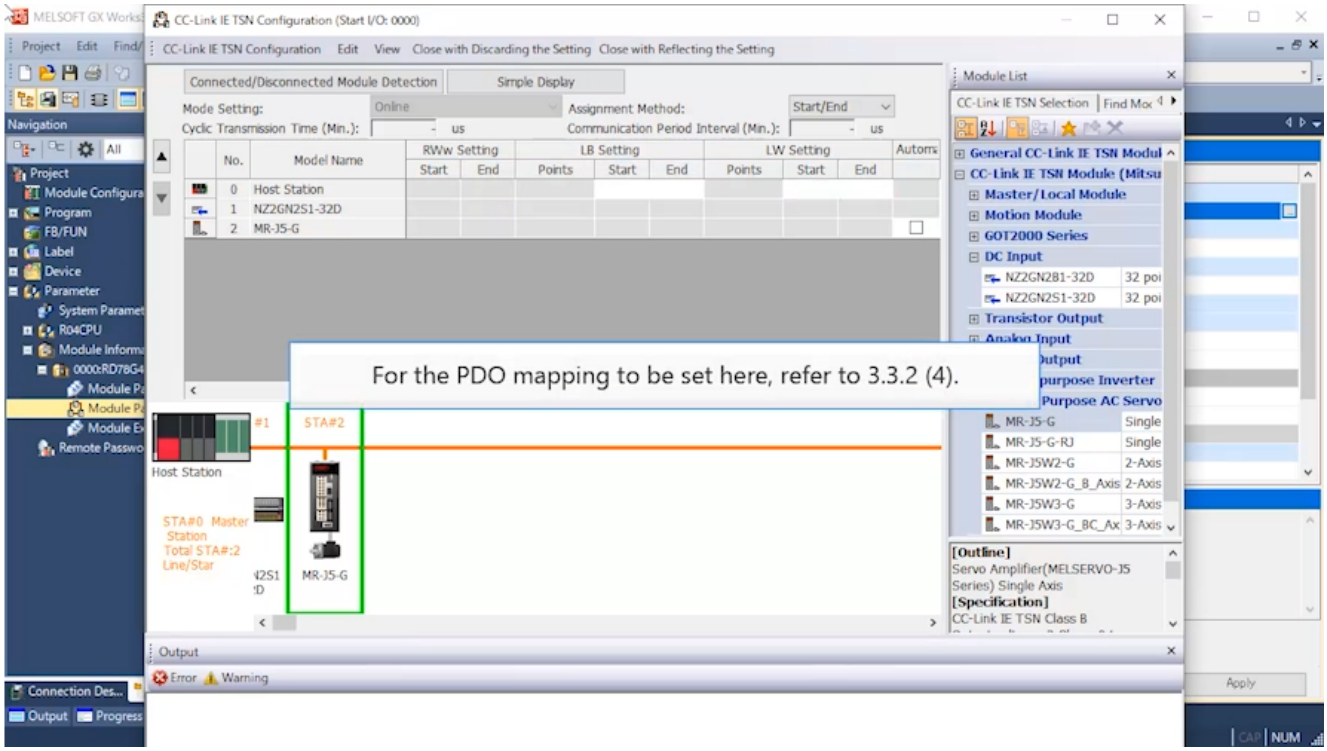
3.3.2 Module parameter (Network)

(3)Remote station setting (continue)



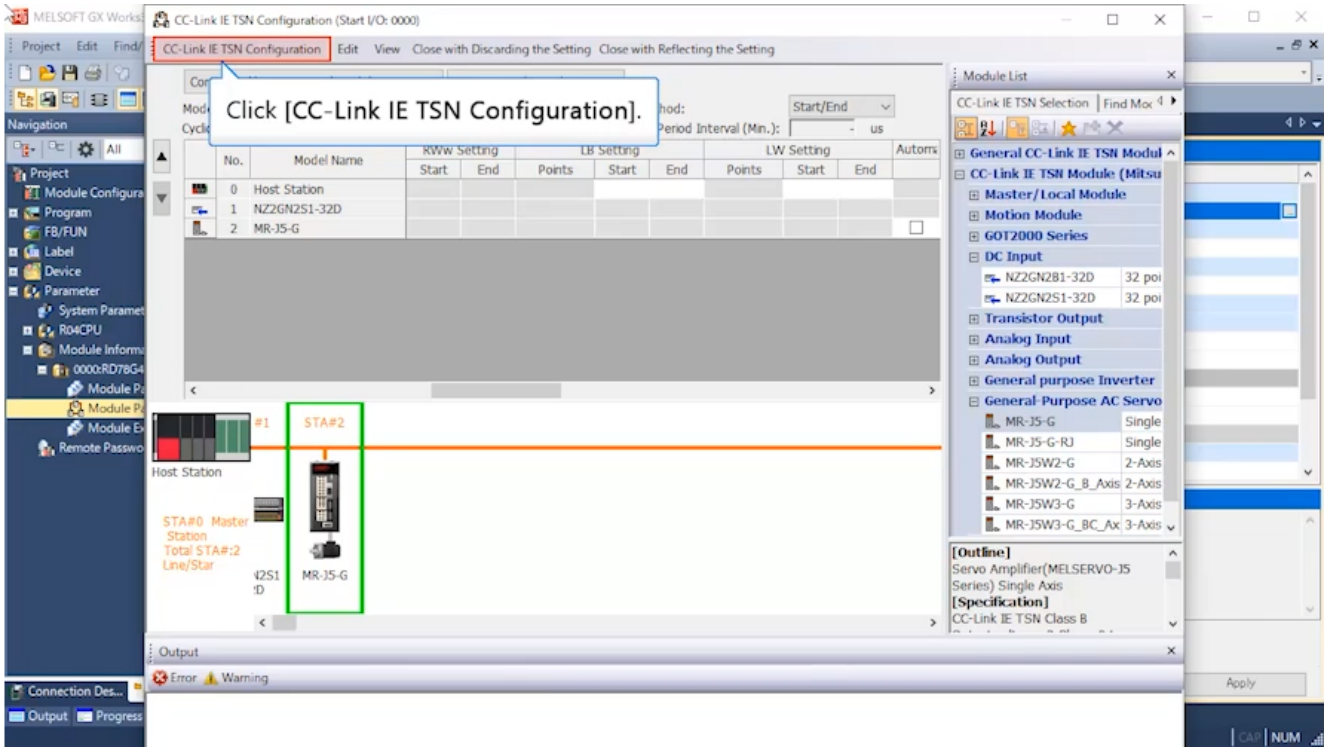
3.3.2 Module parameter (Network)

(3)Remote station setting (continue)



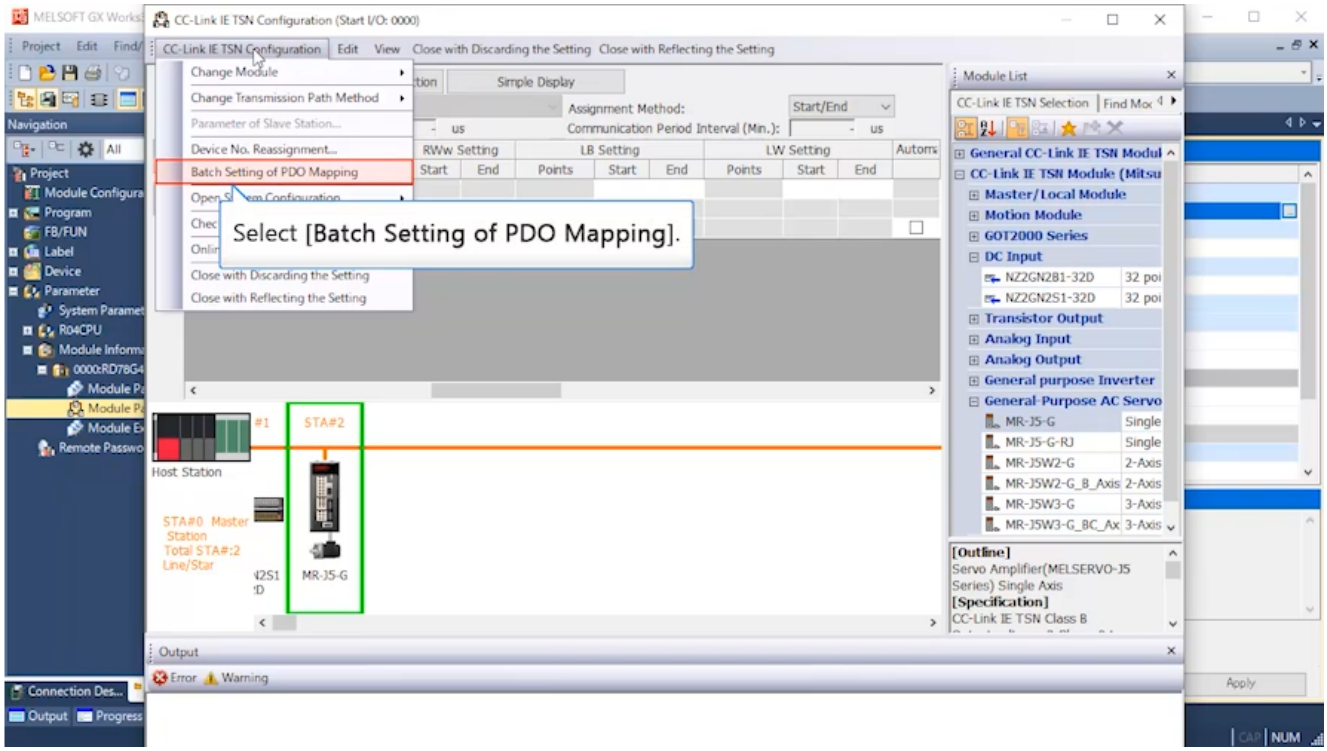
3.3.2 Module parameter (Network)

(3)Remote station setting (continue)



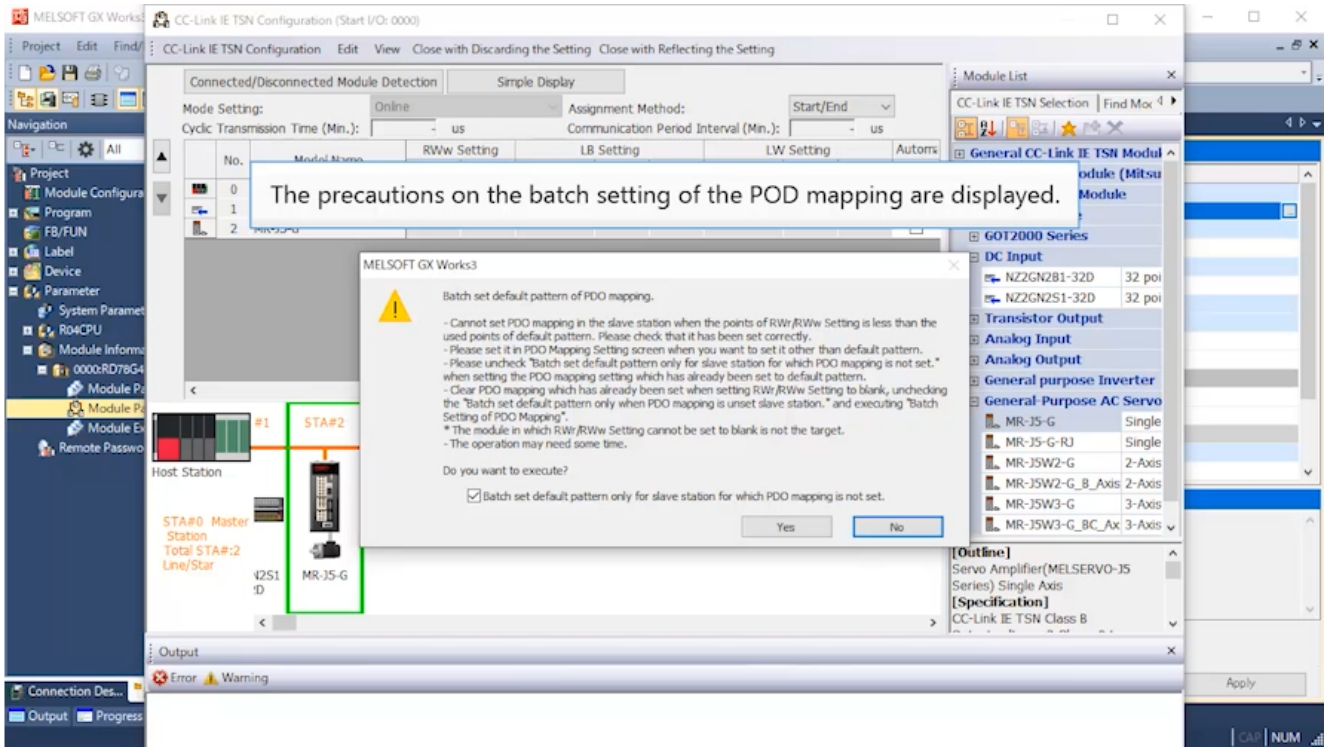
3.3.2 Module parameter (Network)

(3)Remote station setting (continue)



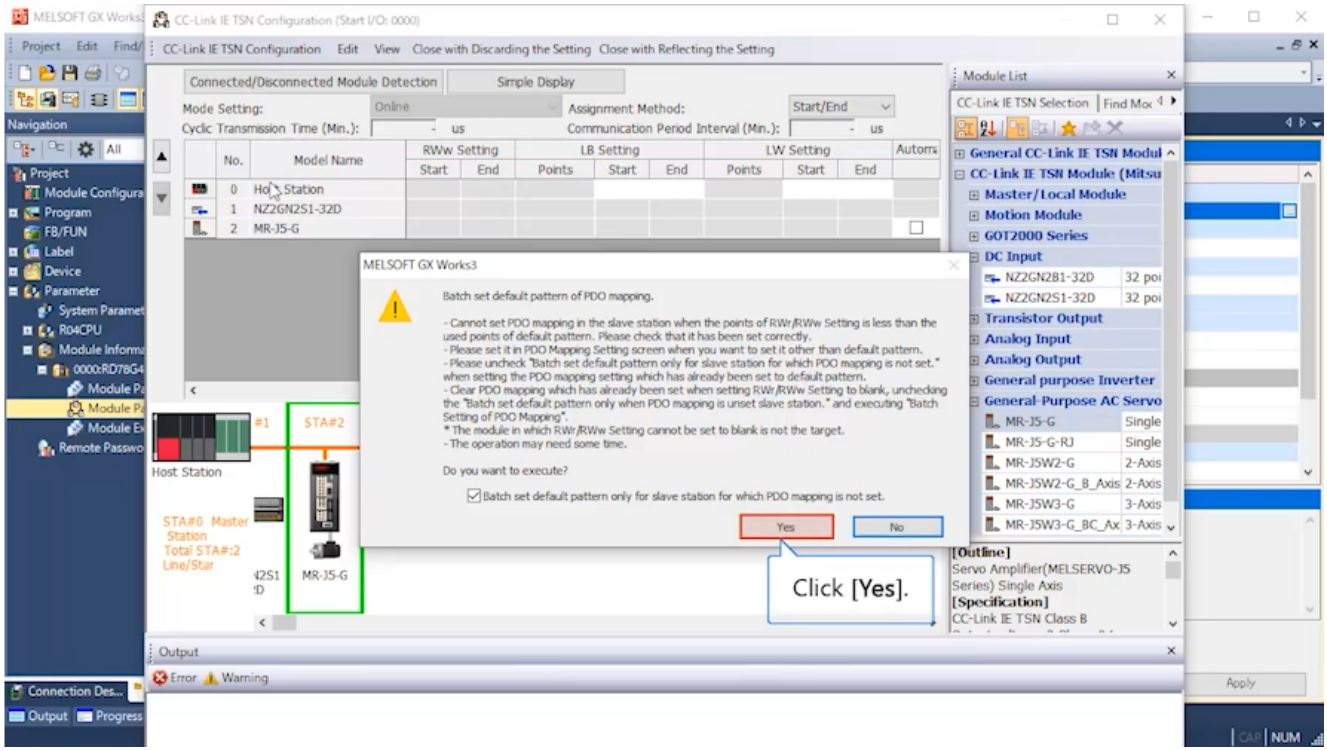
3.3.2 Module parameter (Network)

(3)Remote station setting (continue)



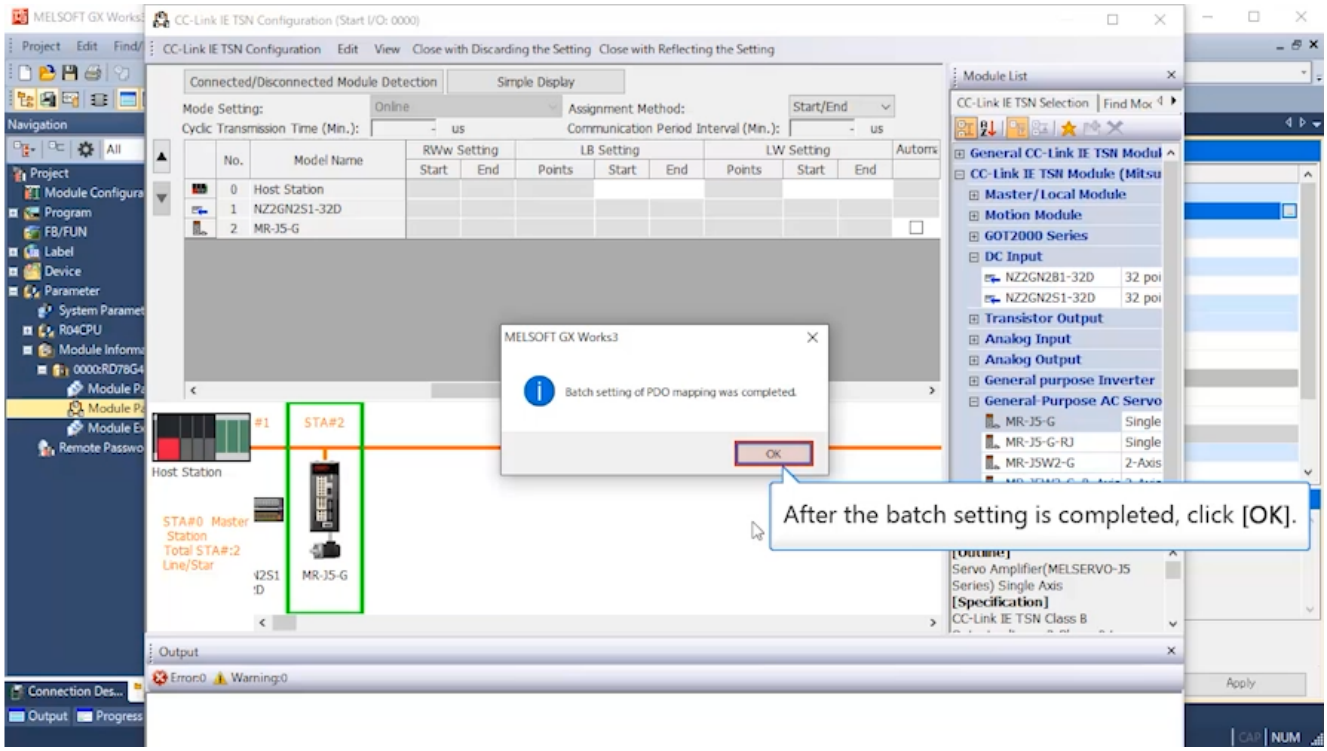
3.3.2 Module parameter (Network)

(3)Remote station setting (continue)



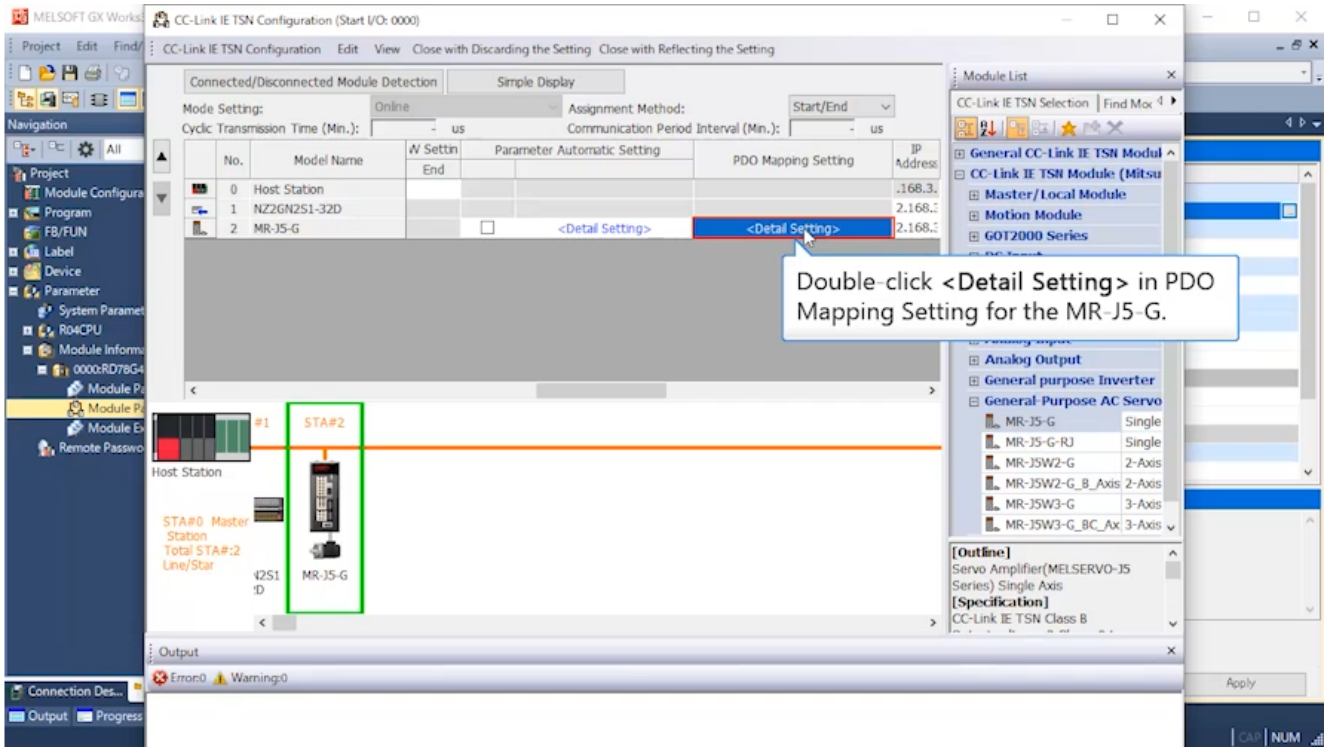
3.3.2 Module parameter (Network)

(3)Remote station setting (continue)



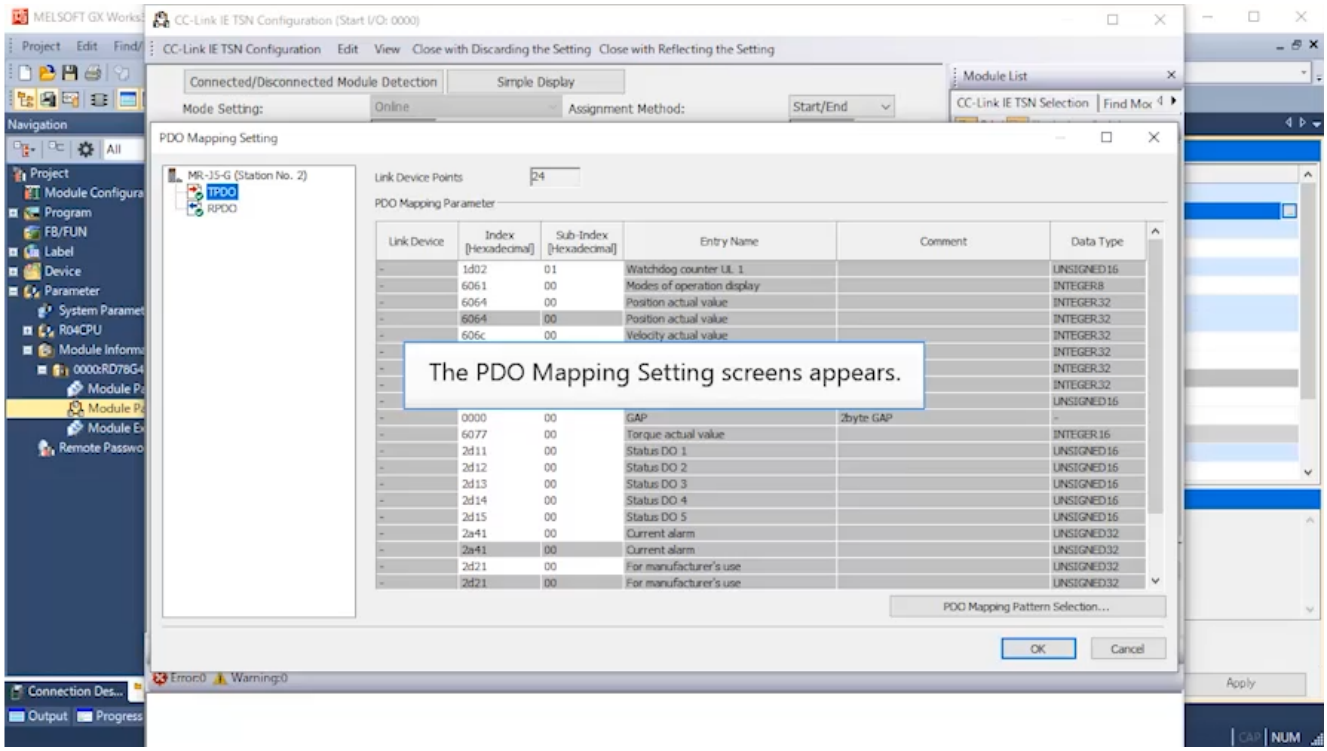
3.3.2 Module parameter (Network)

(3)Remote station setting (continue)



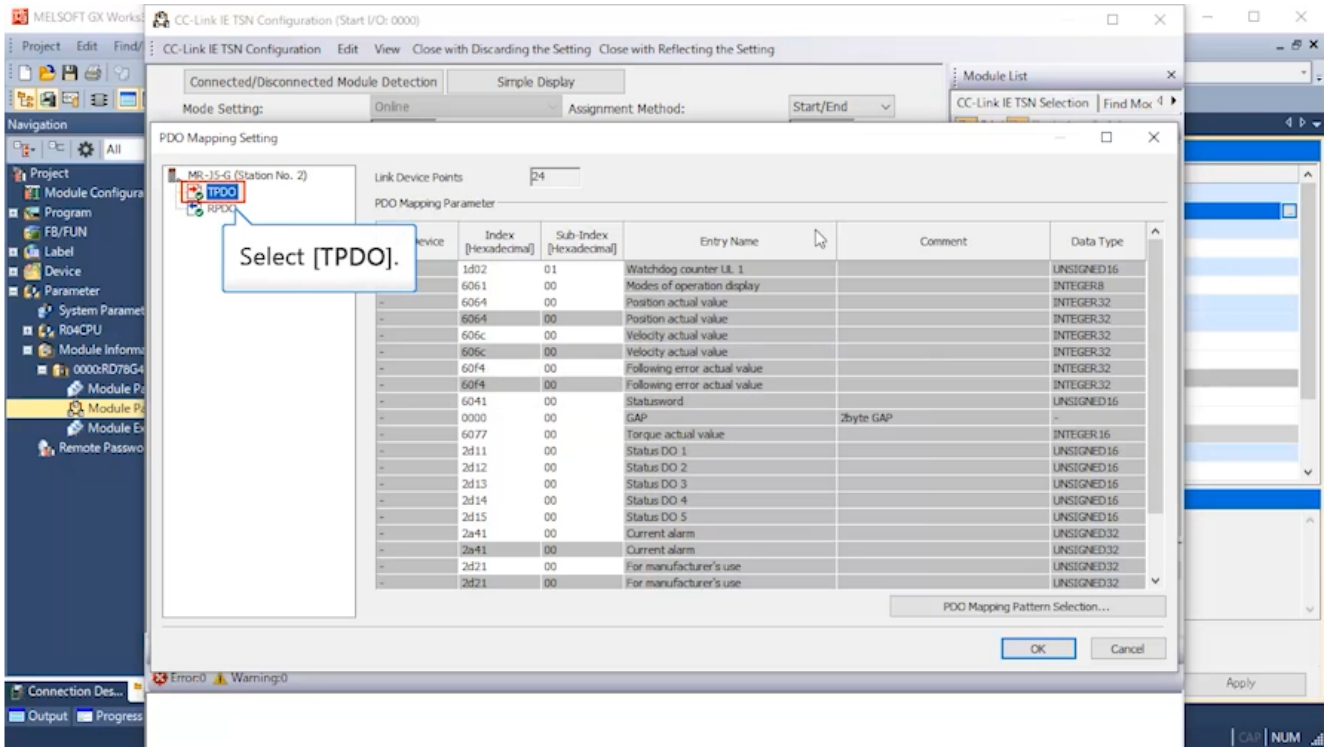
3.3.2 Module parameter (Network)

(3)Remote station setting (continue)



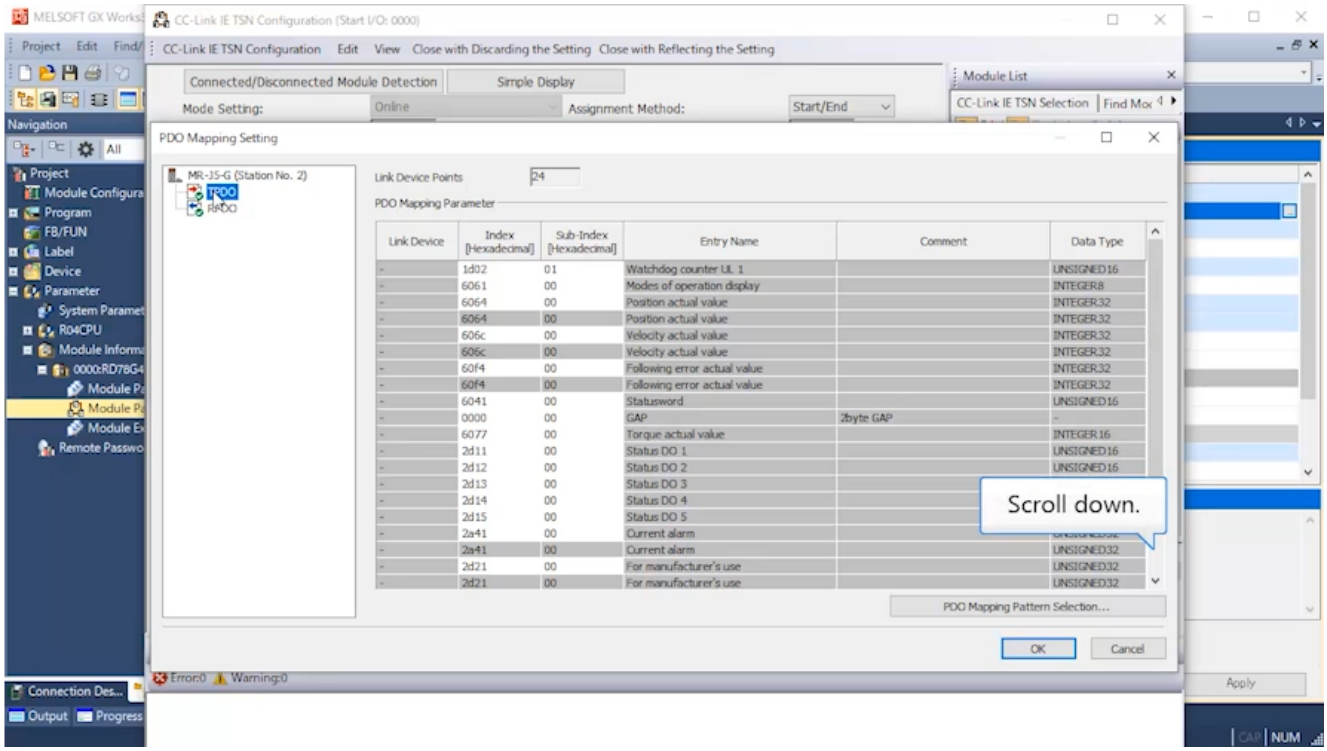
3.3.2 Module parameter (Network)

(3)Remote station setting (continue)



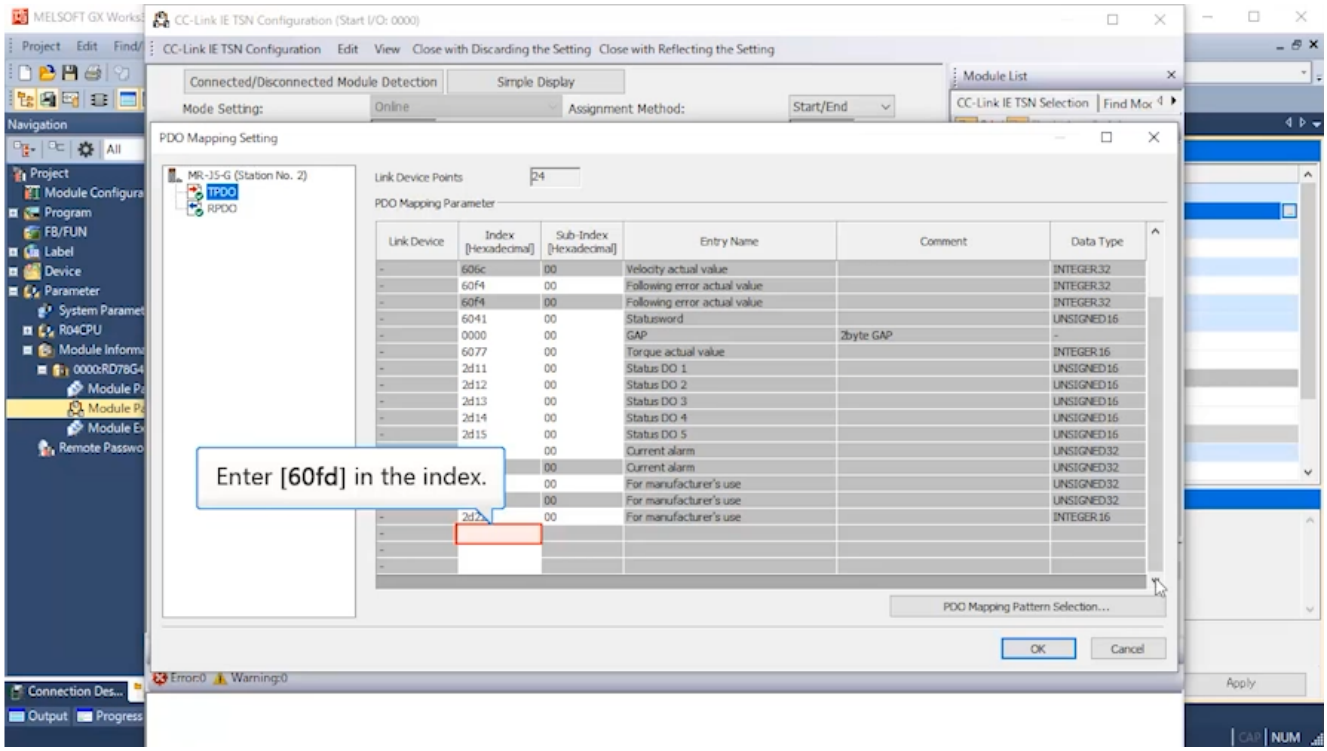
3.3.2 Module parameter (Network)

(3)Remote station setting (continue)



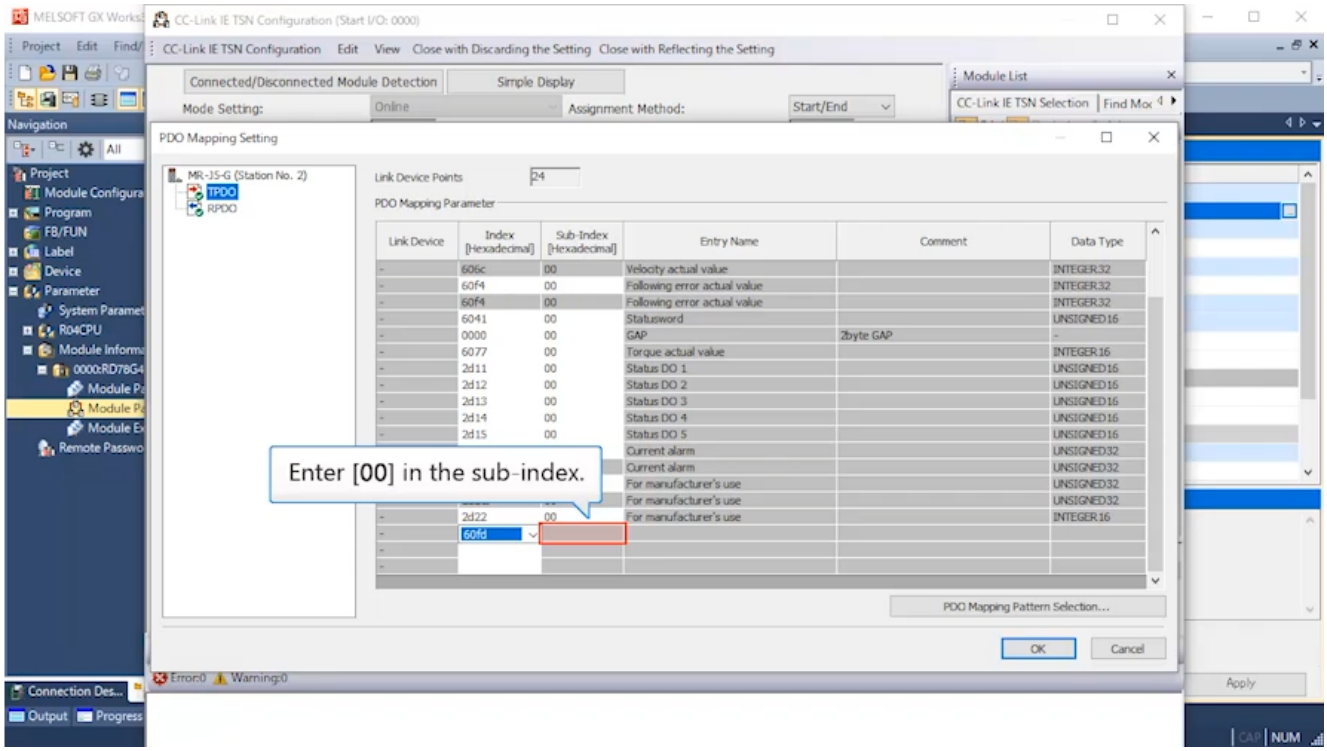
3.3.2 Module parameter (Network)

(3)Remote station setting (continue)



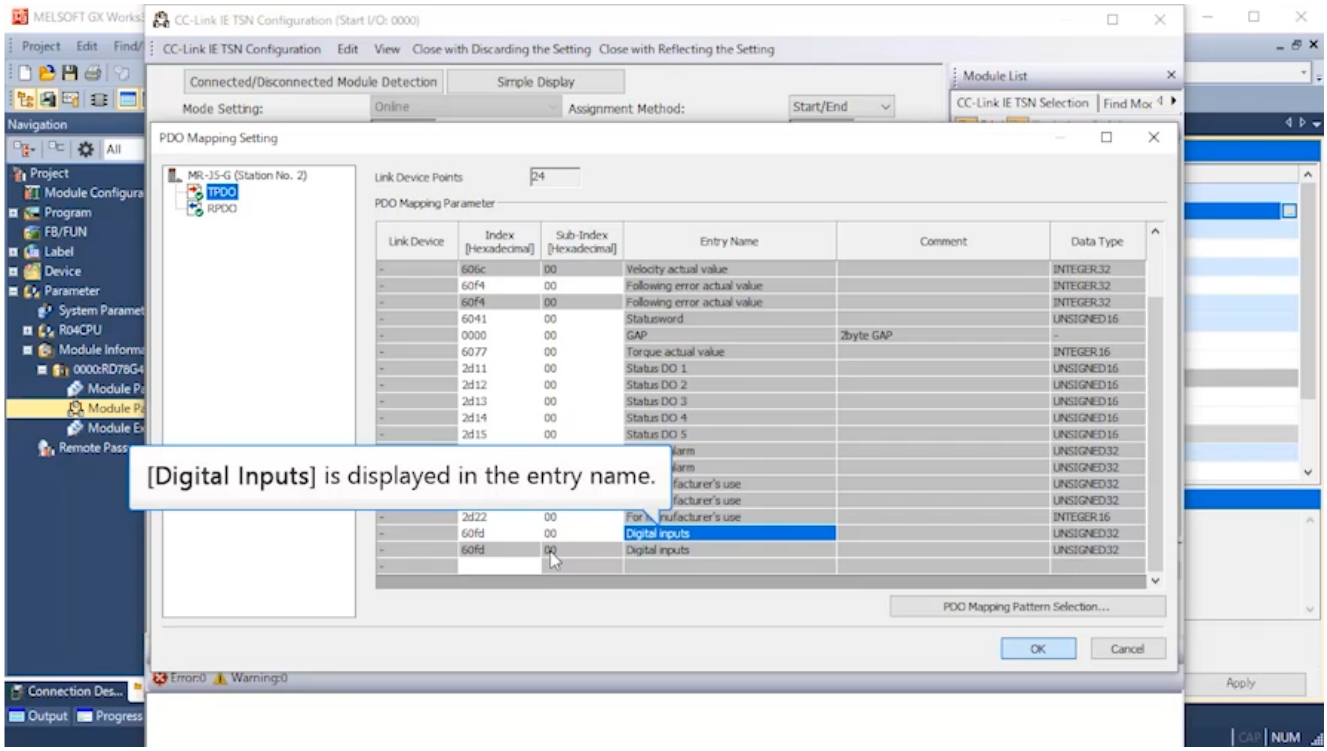
3.3.2 Module parameter (Network)

(3)Remote station setting (continue)



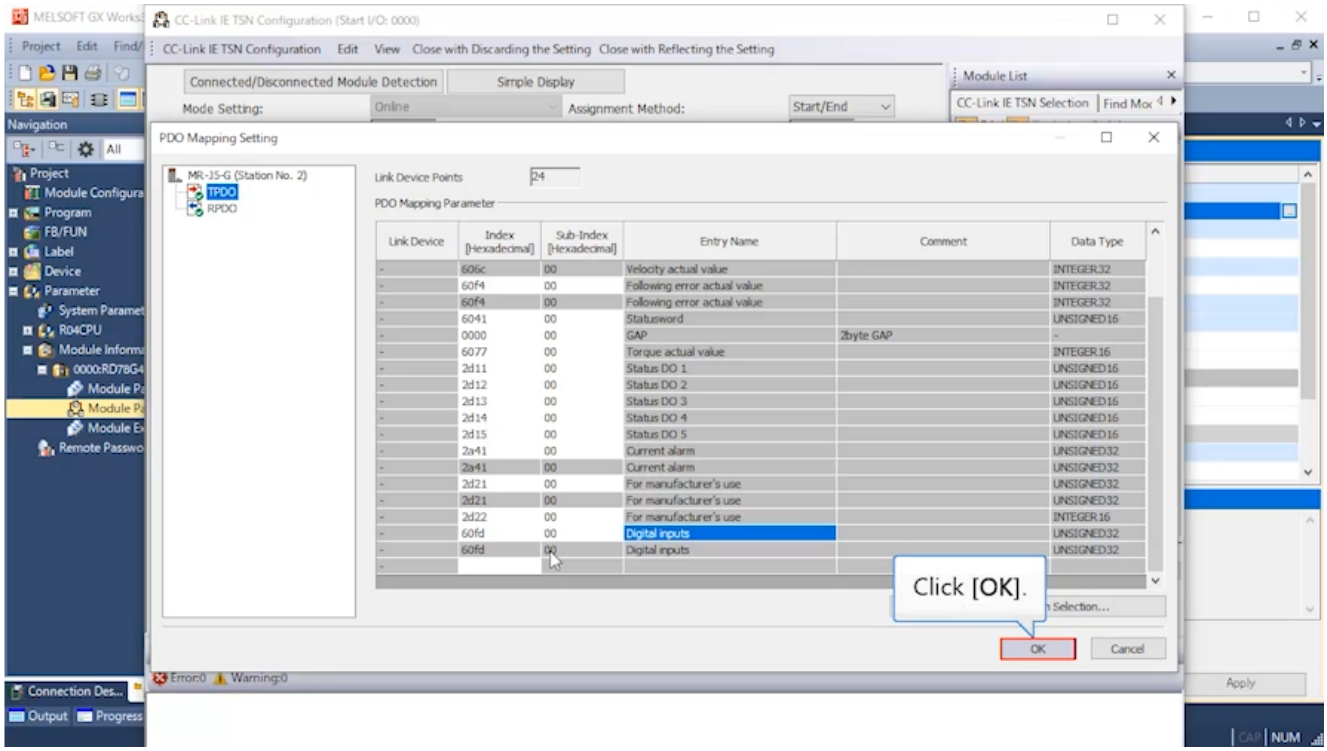
3.3.2 Module parameter (Network)

(3)Remote station setting (continue)



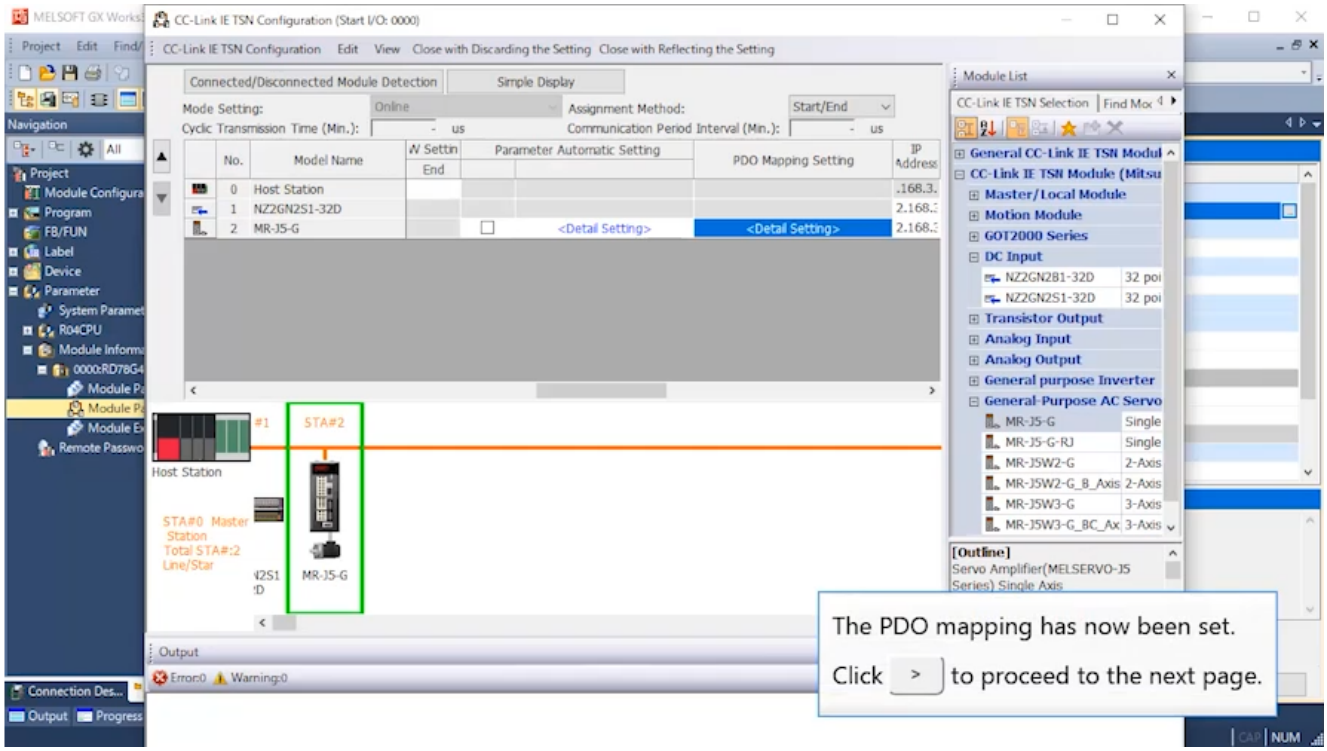
3.3.2 Module parameter (Network)

(3)Remote station setting (continue)



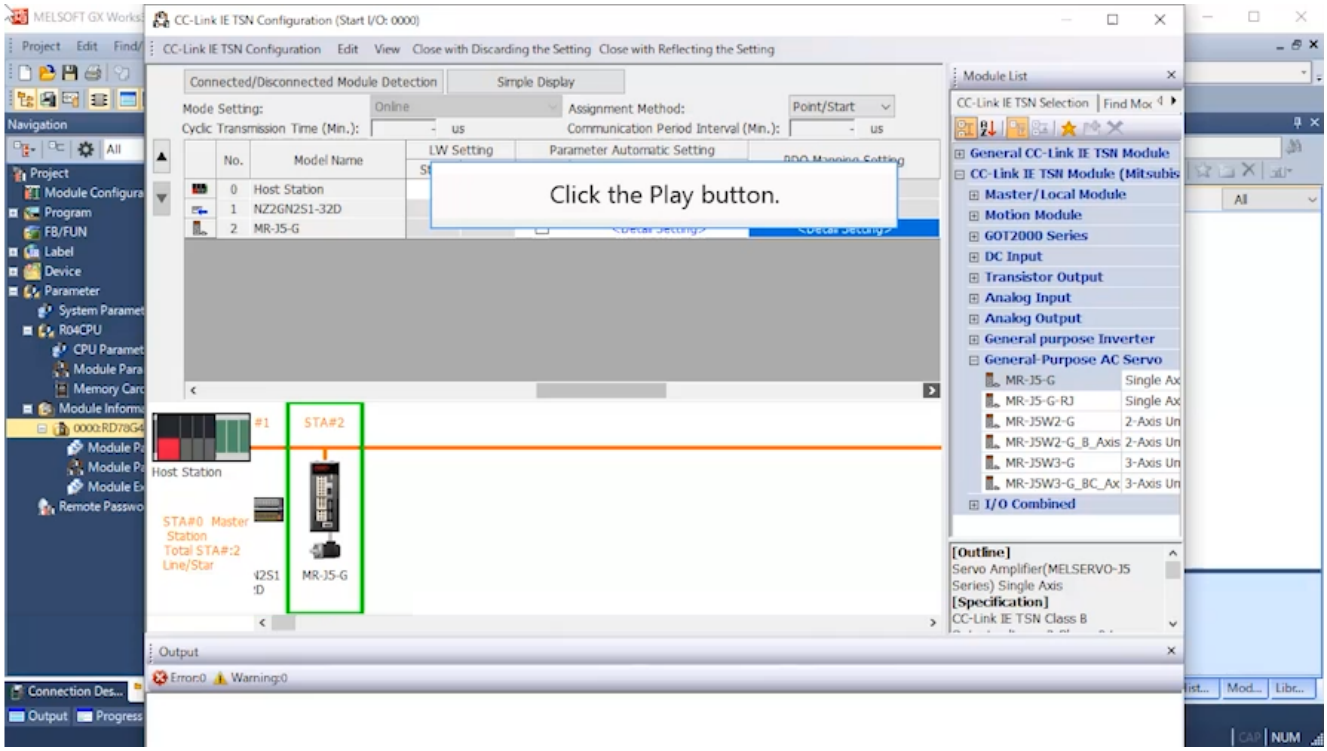
3.3.2 Module parameter (Network)

(3)Remote station setting (continue)



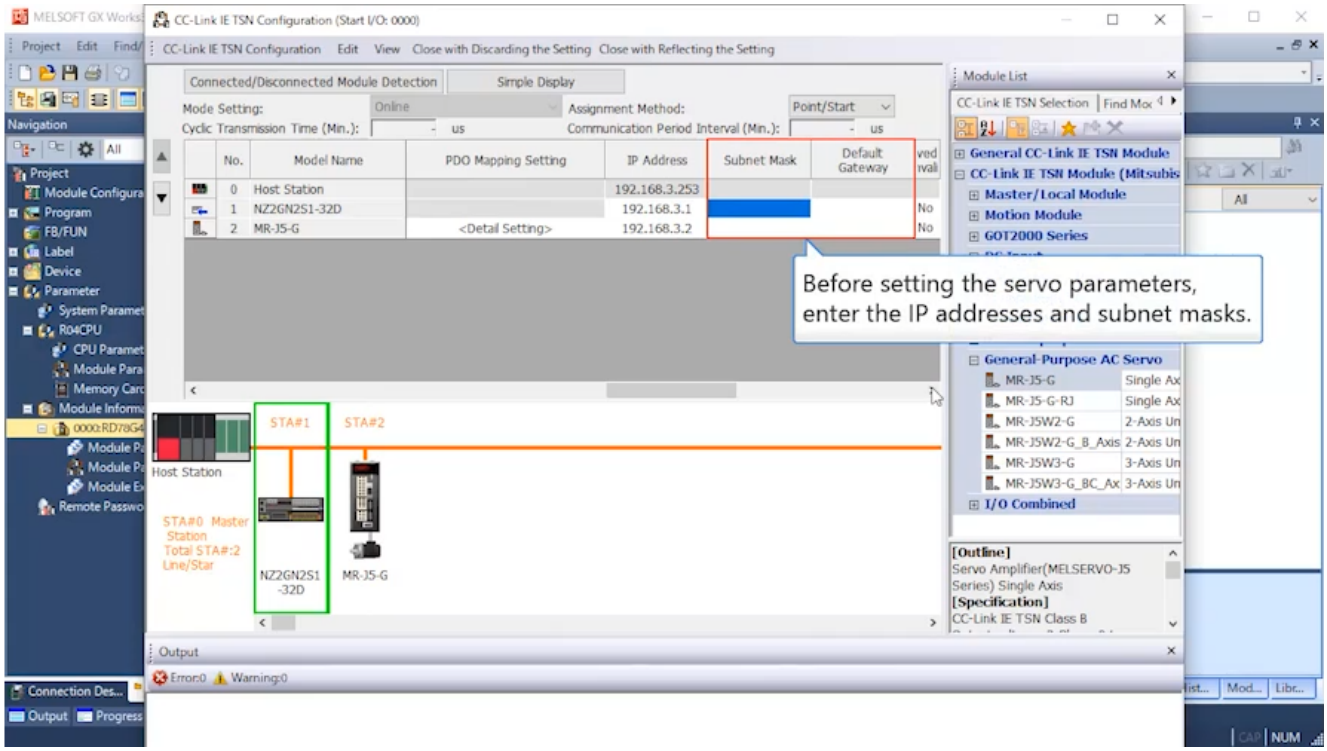
3.3.2 Module parameter (Network)

(3)Remote station setting (continue)



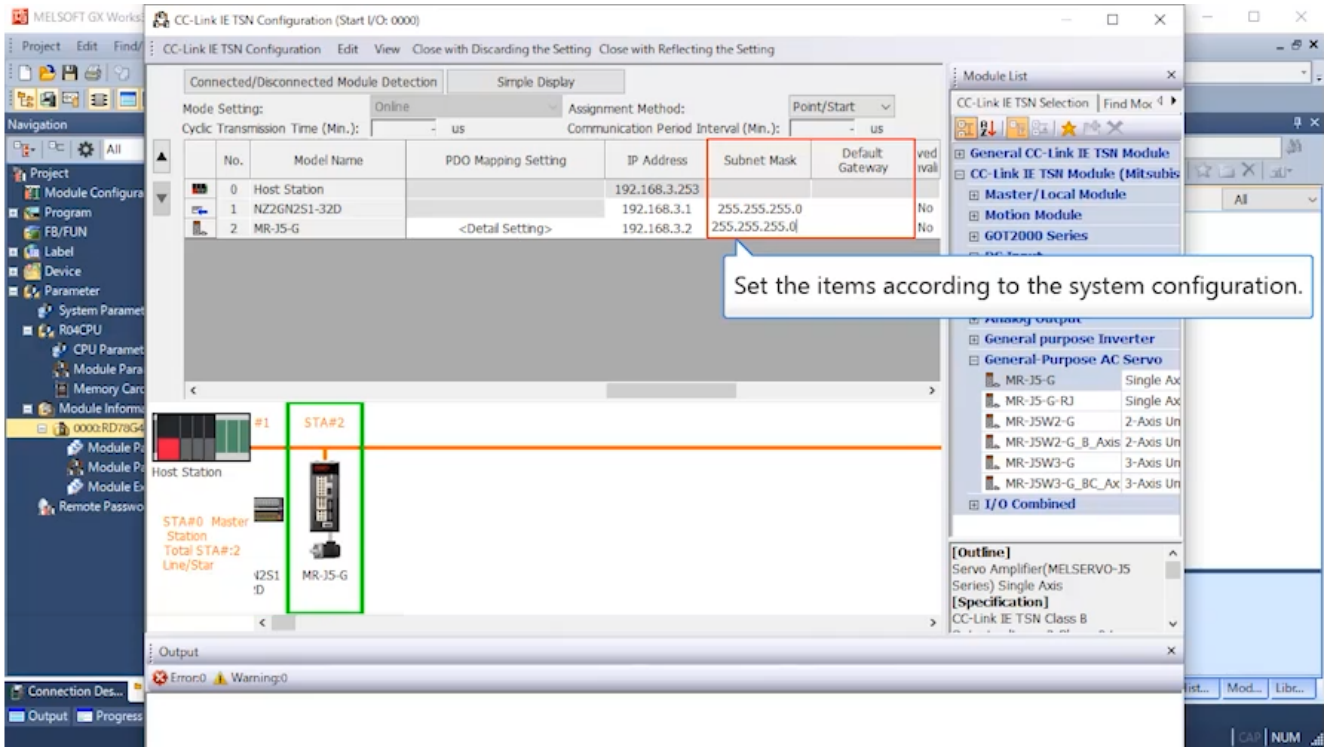
3.3.2 Module parameter (Network)

(3)Remote station setting (continue)



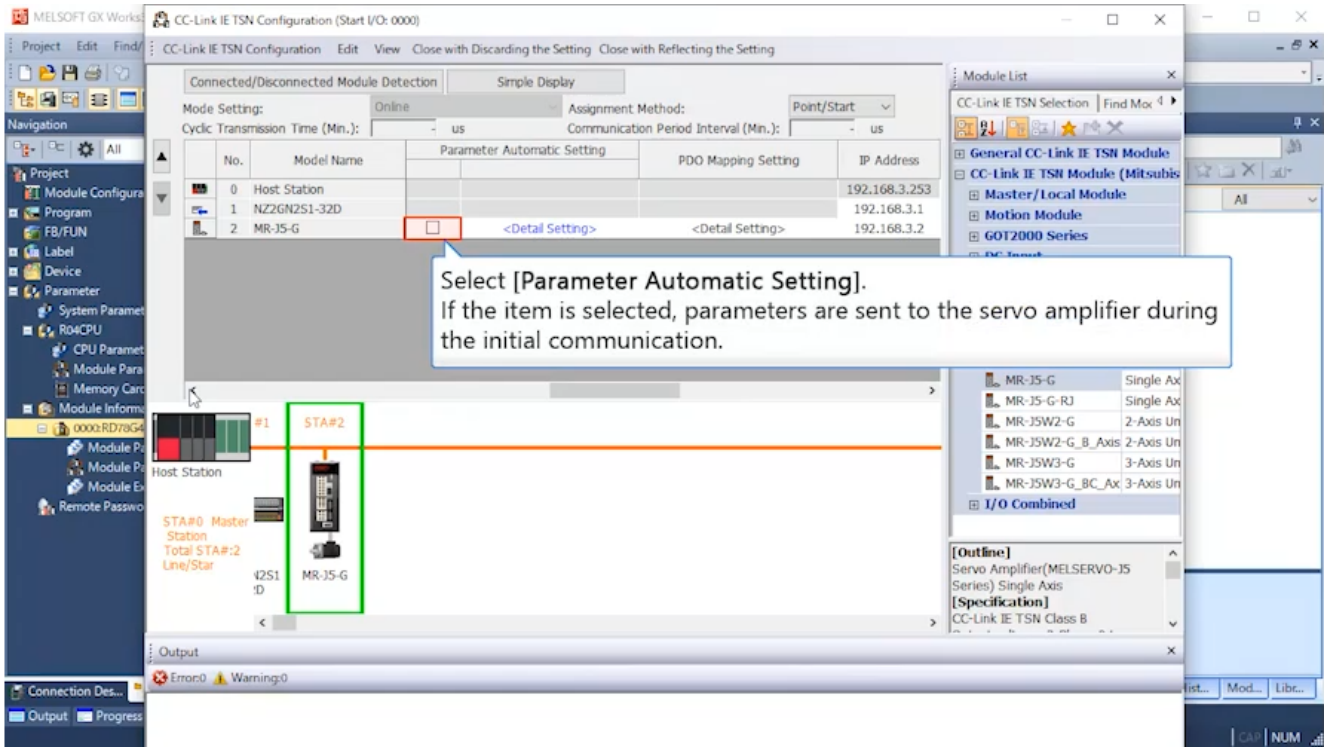
3.3.2 Module parameter (Network)

(3)Remote station setting (continue)



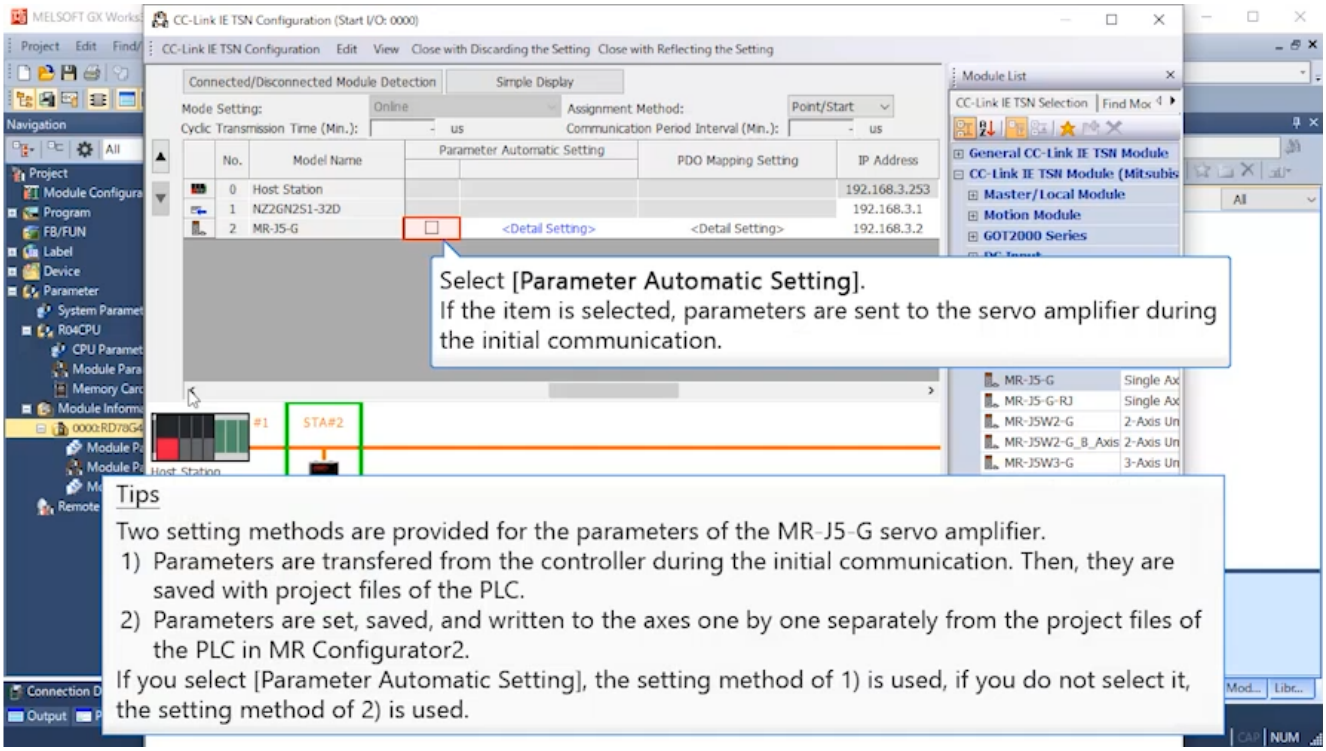
3.3.2 Module parameter (Network)

(3)Remote station setting (continue)



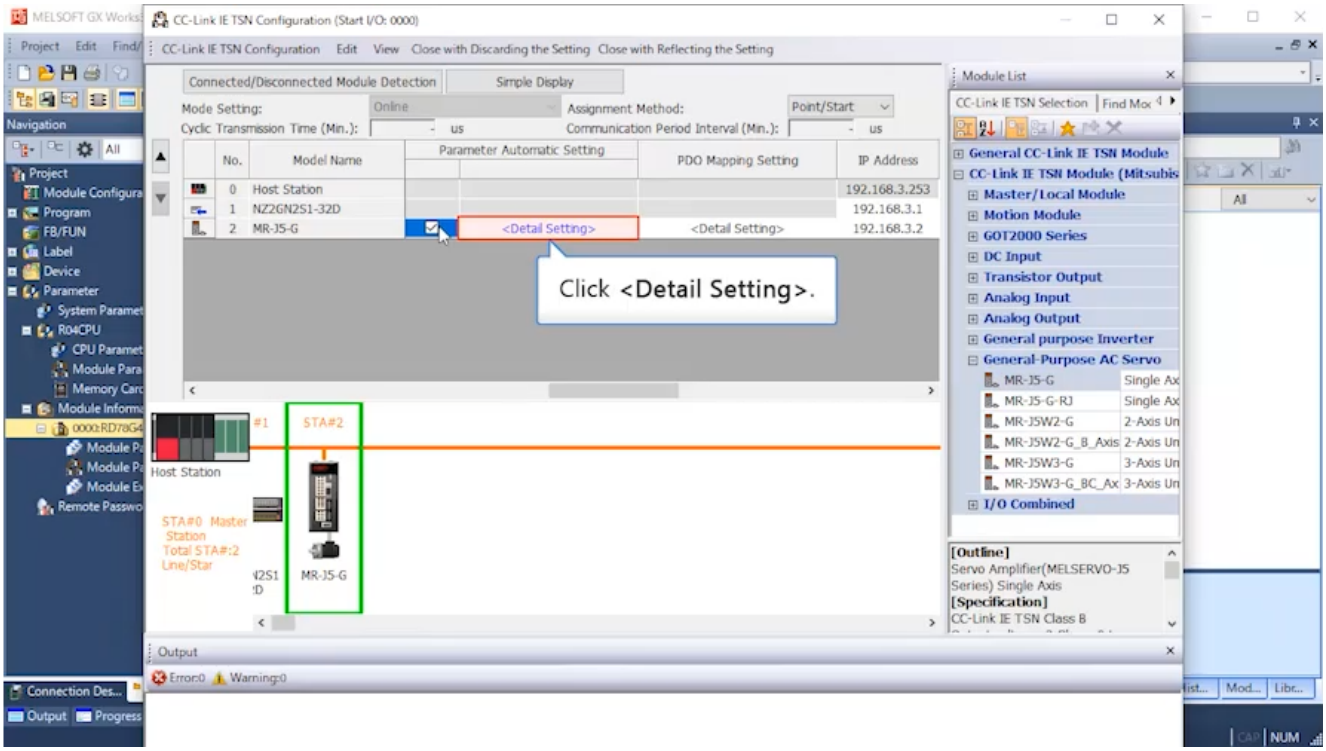
3.3.2 Module parameter (Network)

(3)Remote station setting (continue)



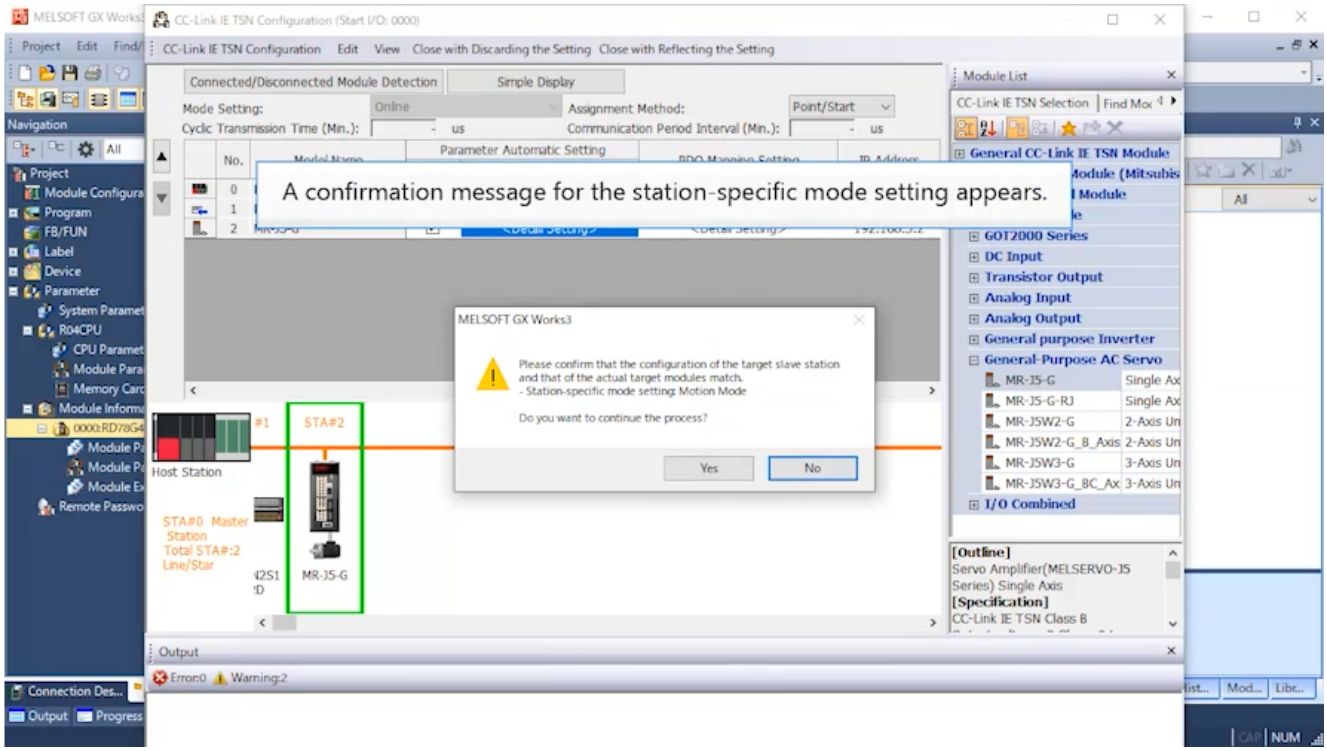
3.3.2 Module parameter (Network)

(3)Remote station setting (continue)



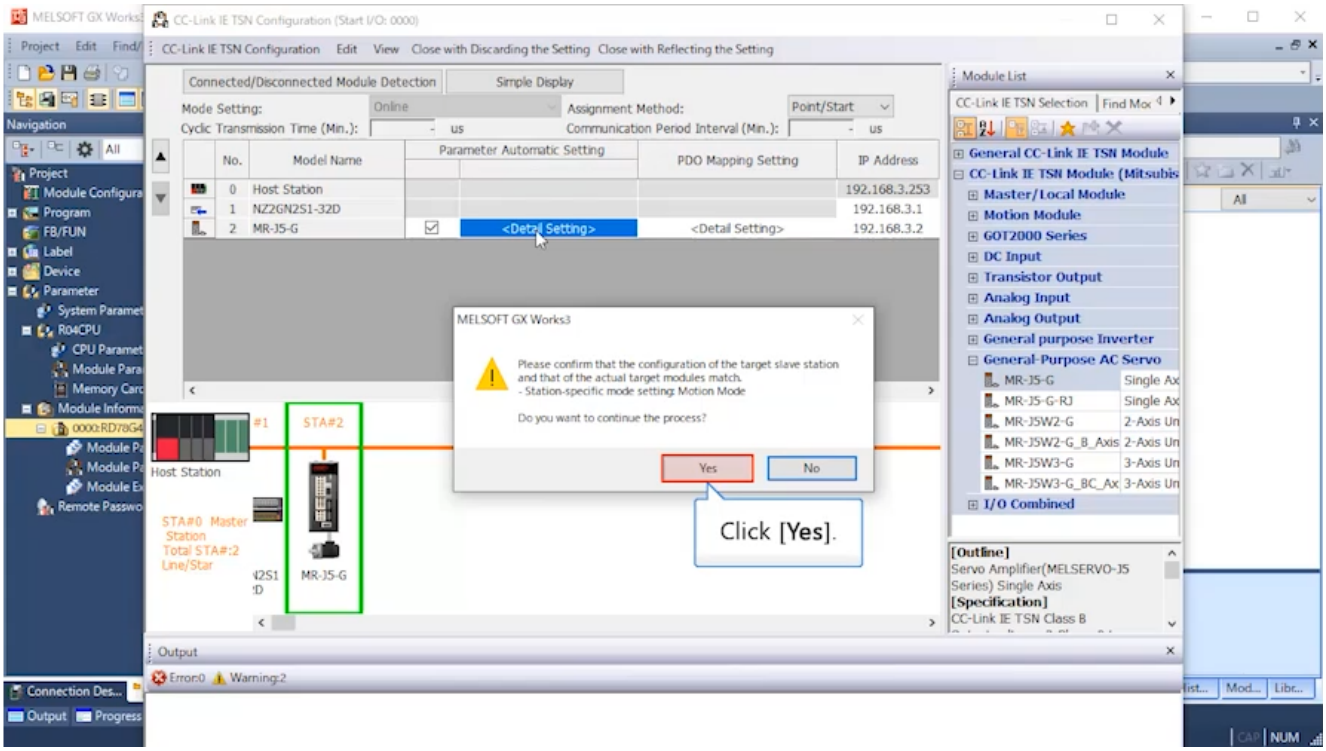
3.3.2 Module parameter (Network)

(3)Remote station setting (continue)



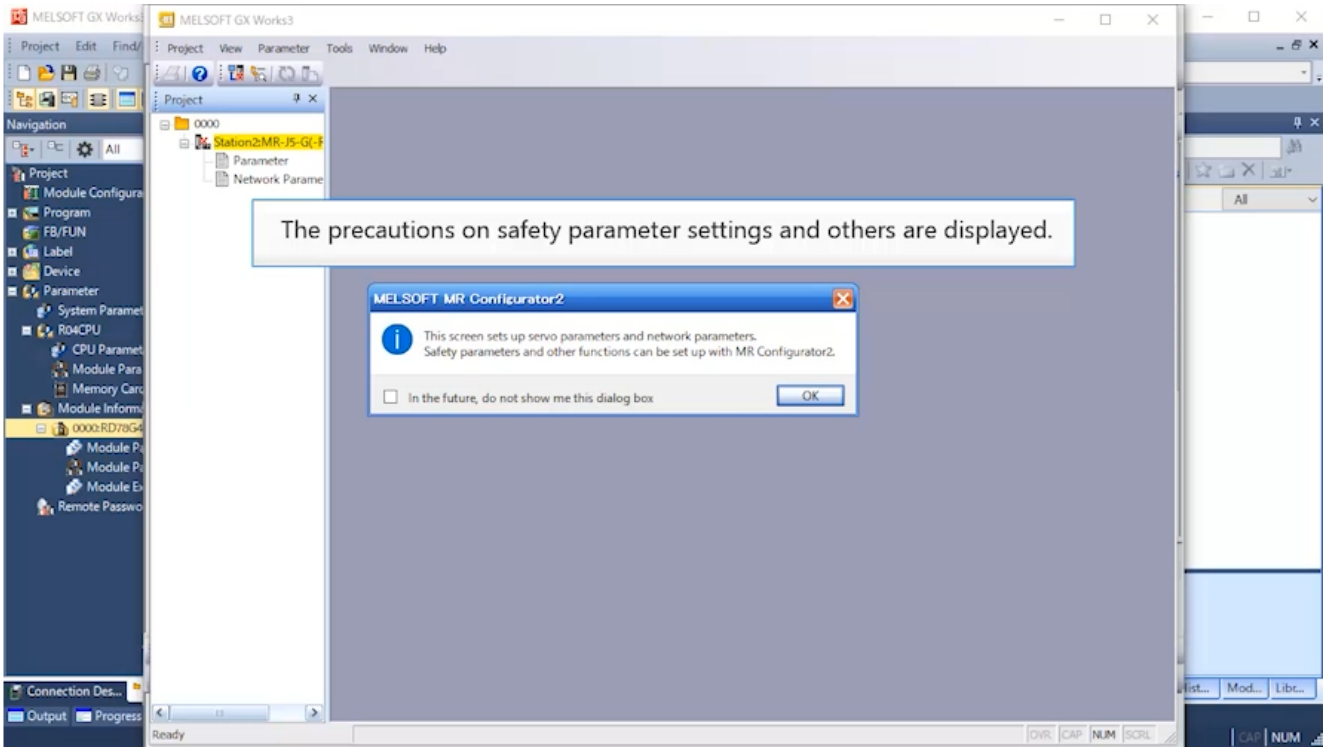
3.3.2 Module parameter (Network)

(3)Remote station setting (continue)



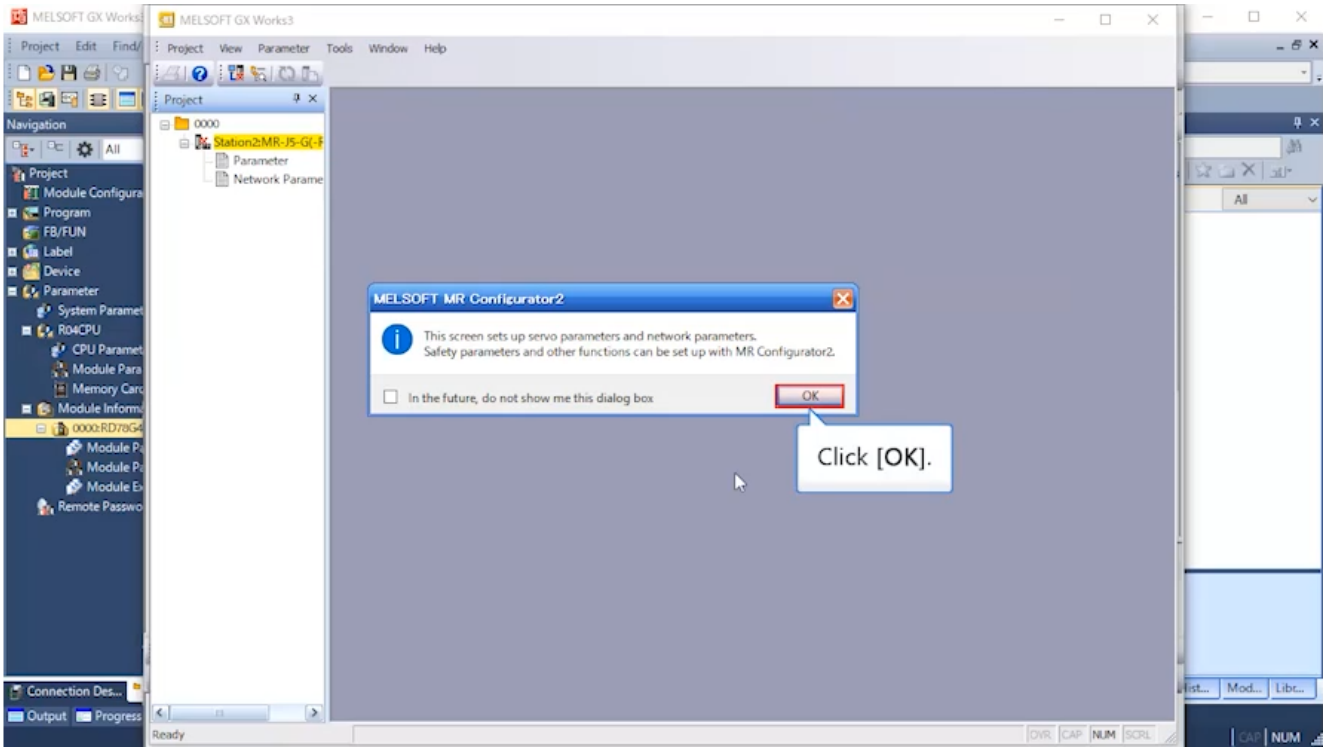
3.3.2 Module parameter (Network)

(3)Remote station setting (continue)



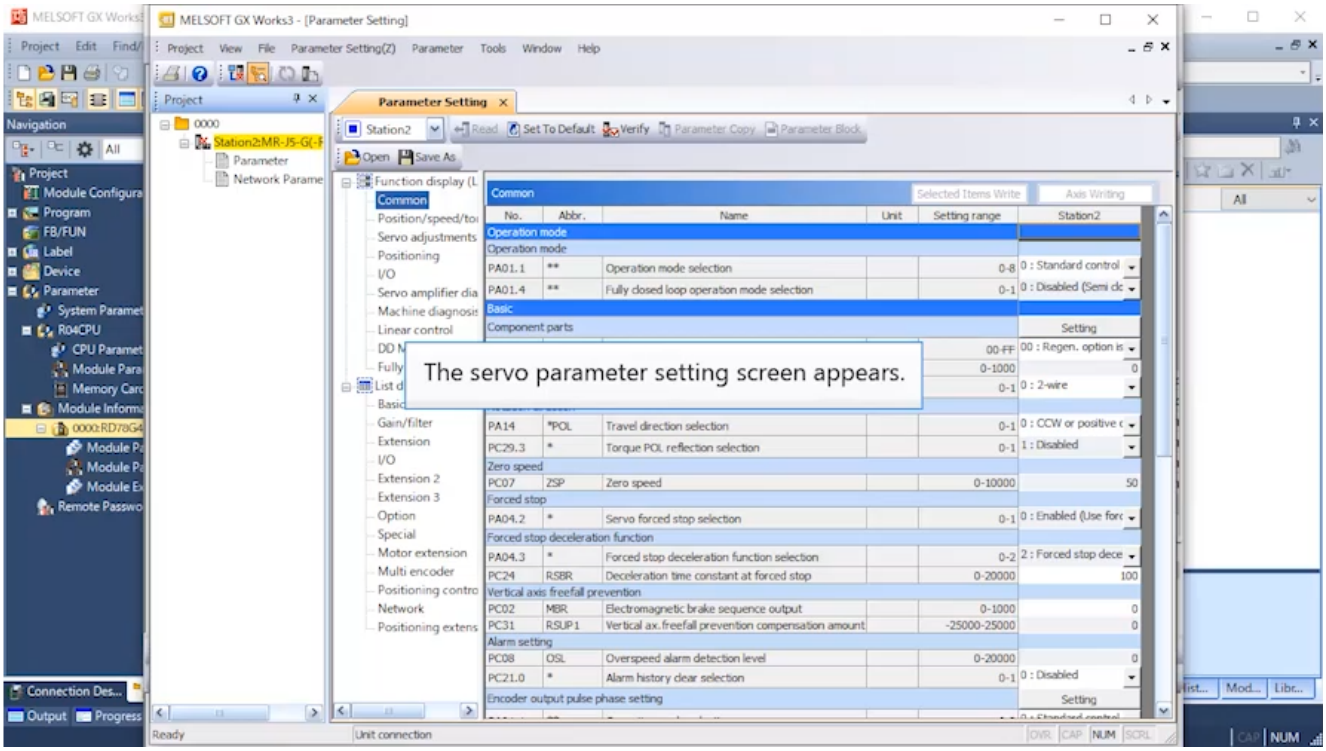
3.3.2 Module parameter (Network)

(3)Remote station setting (continue)



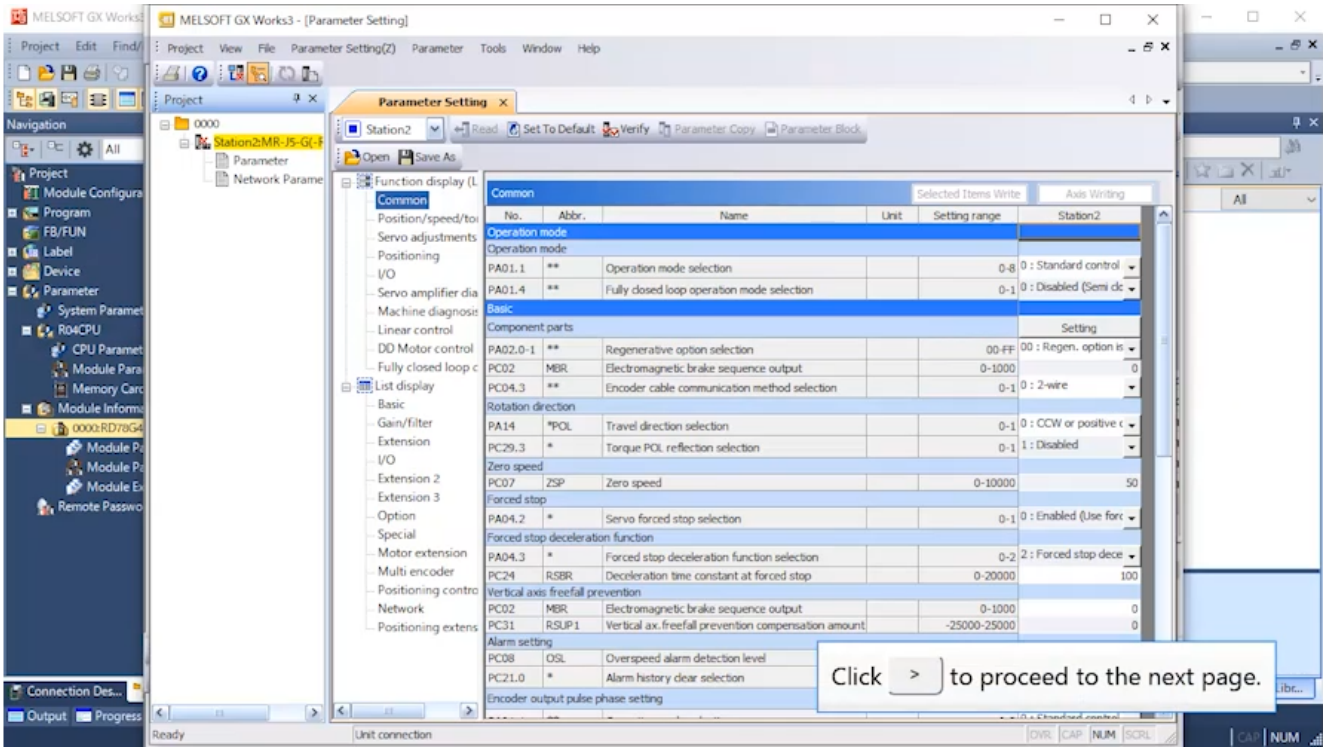
3.3.2 Module parameter (Network)

(3)Remote station setting (continue)



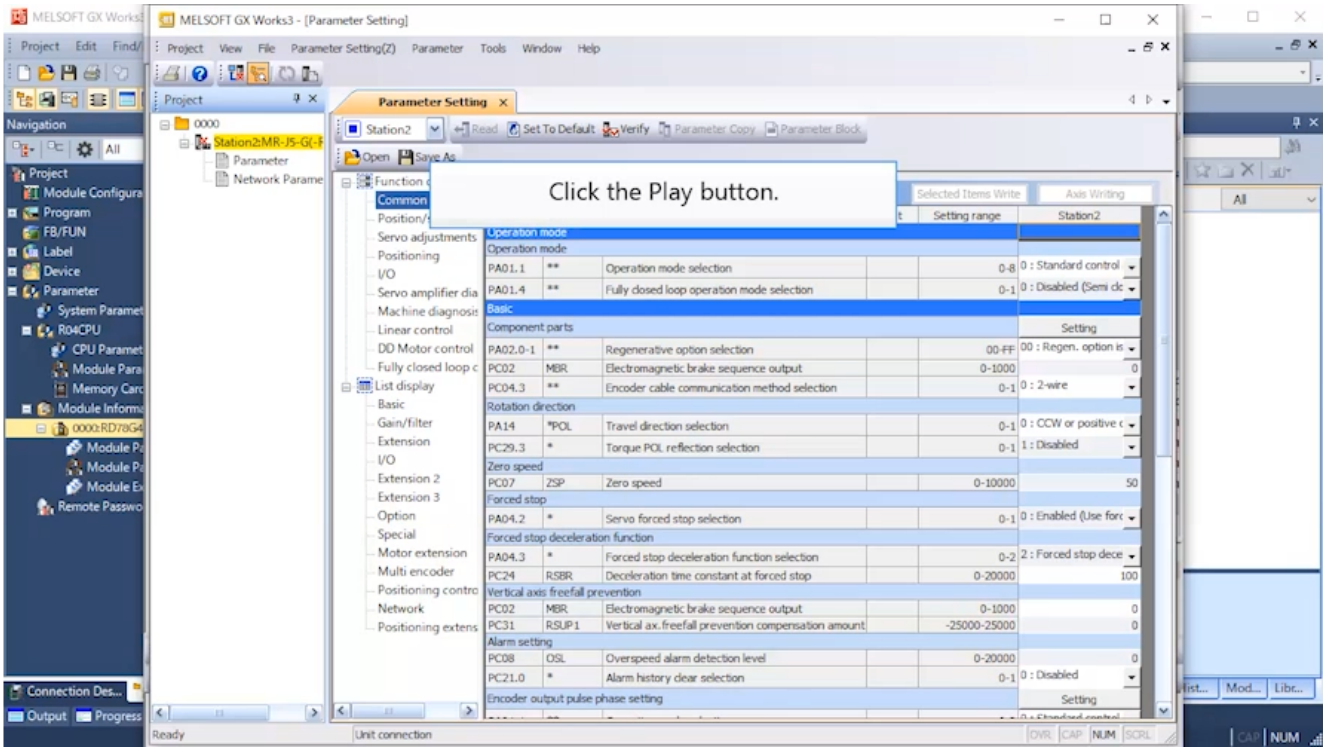
3.3.2 Module parameter (Network)

(3)Remote station setting (continue)



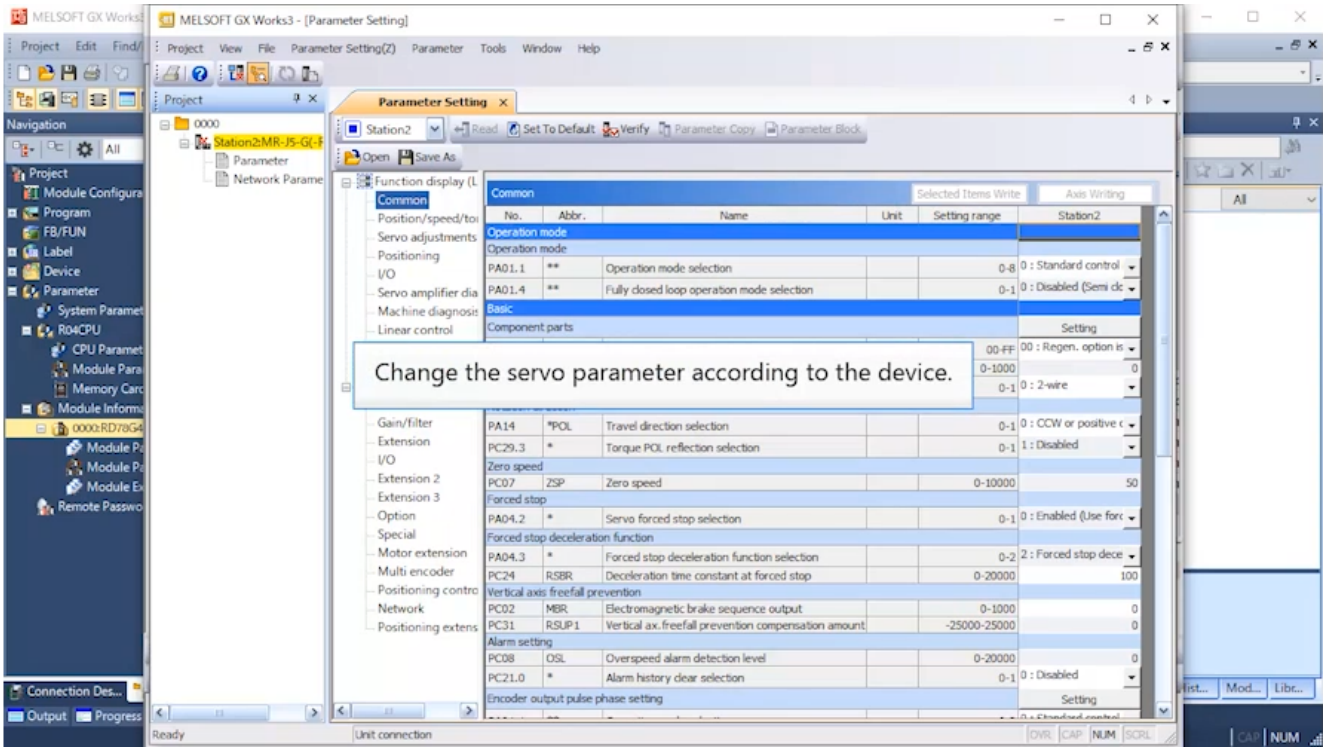
3.3.2 Module parameter (Network)

(3)Remote station setting (continue)



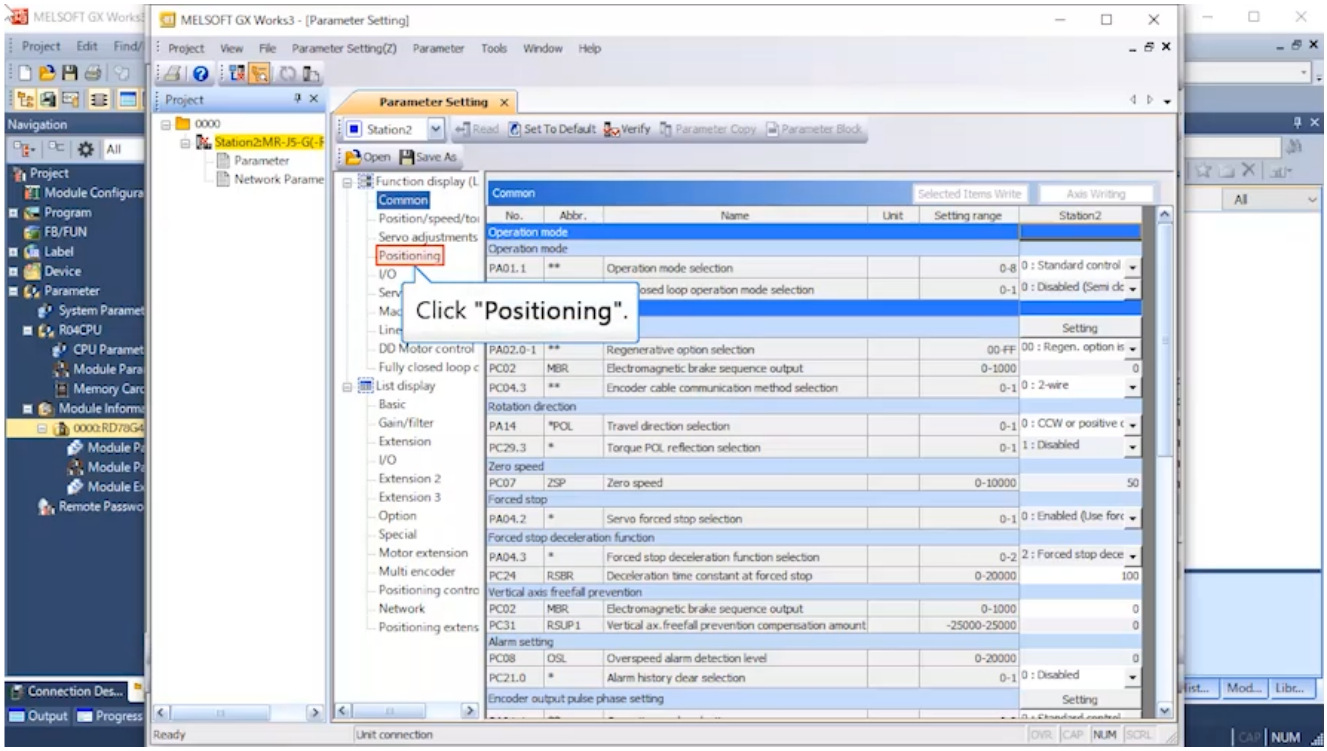
3.3.2 Module parameter (Network)

(3)Remote station setting (continue)



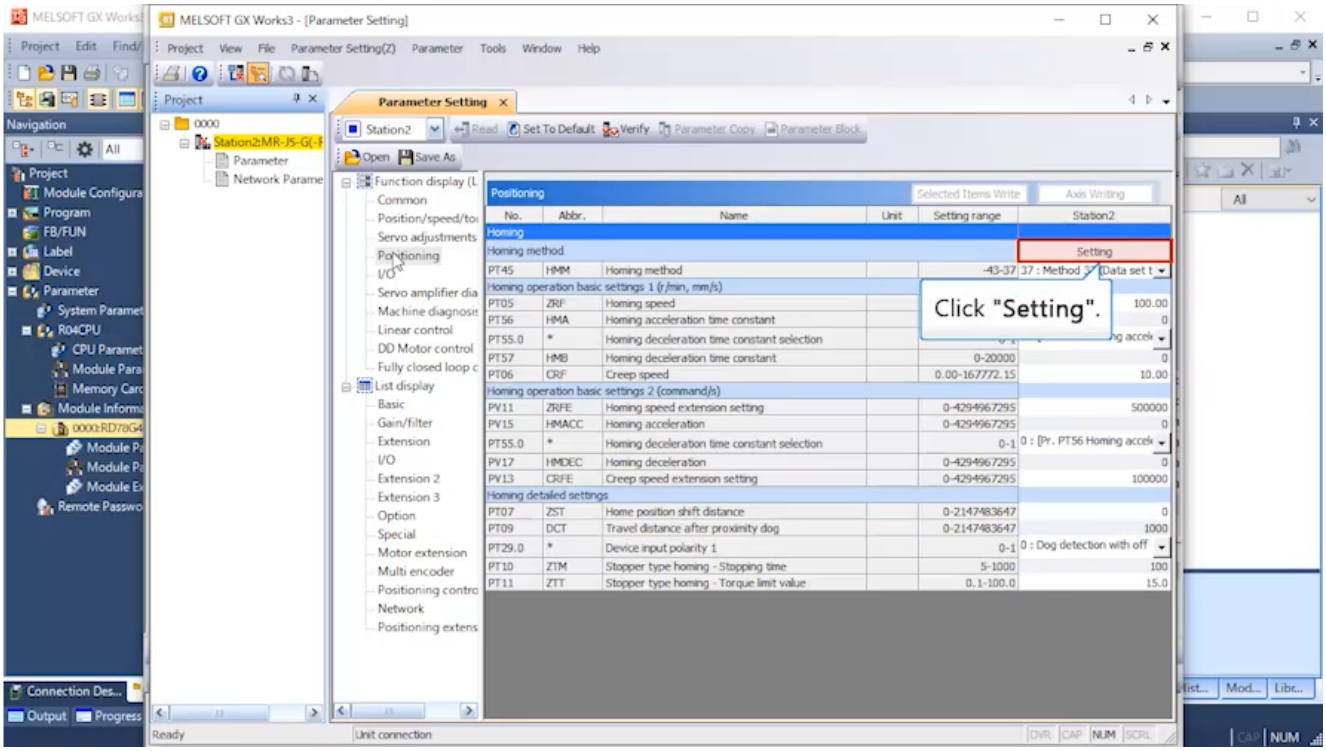
3.3.2 Module parameter (Network)

(3)Remote station setting (continue)



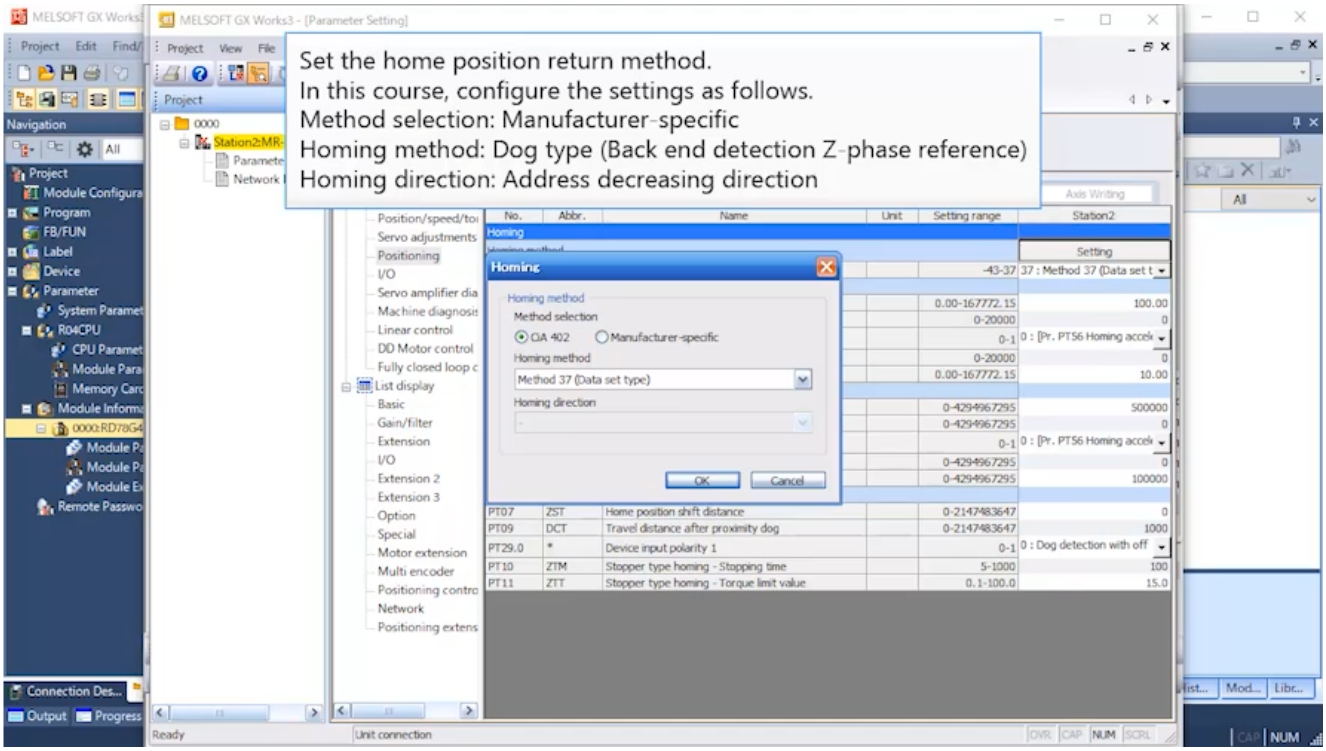
3.3.2 Module parameter (Network)

(3)Remote station setting (continue)



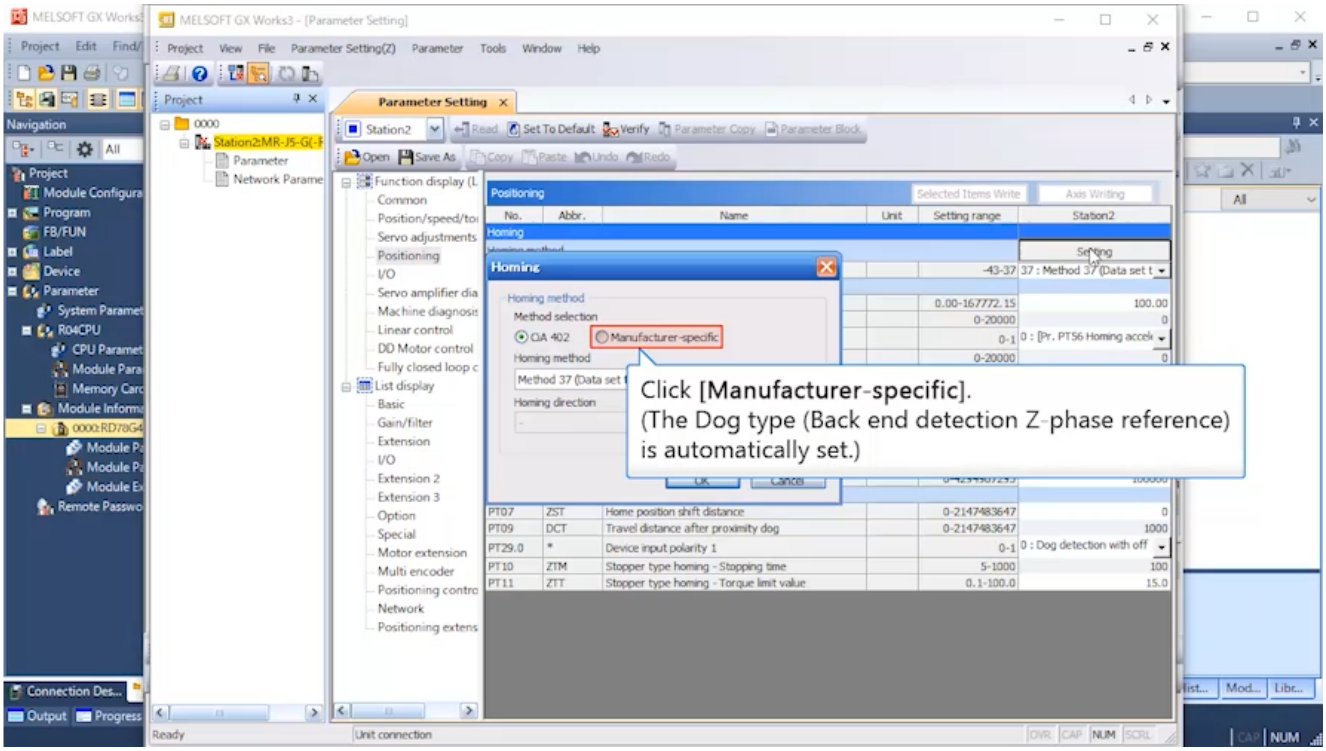
3.3.2 Module parameter (Network)

(3)Remote station setting (continue)



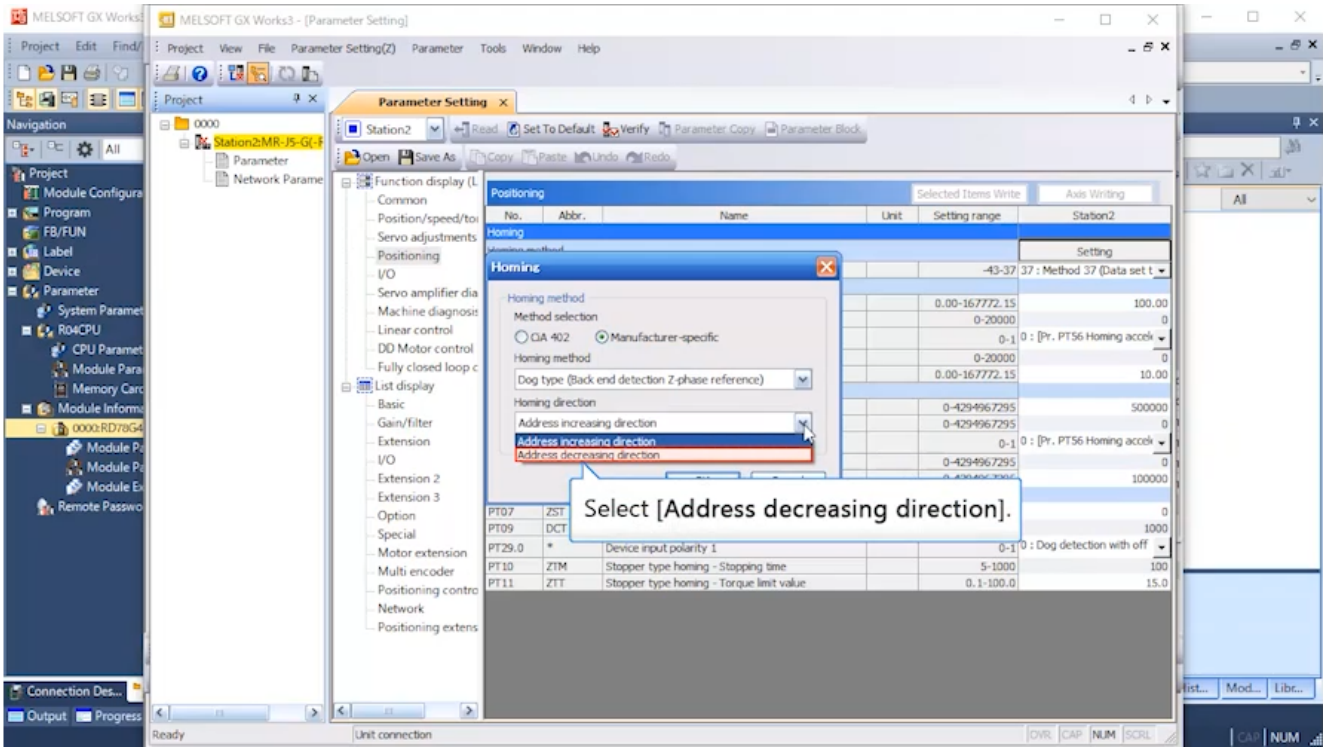
3.3.2 Module parameter (Network)

(3)Remote station setting (continue)



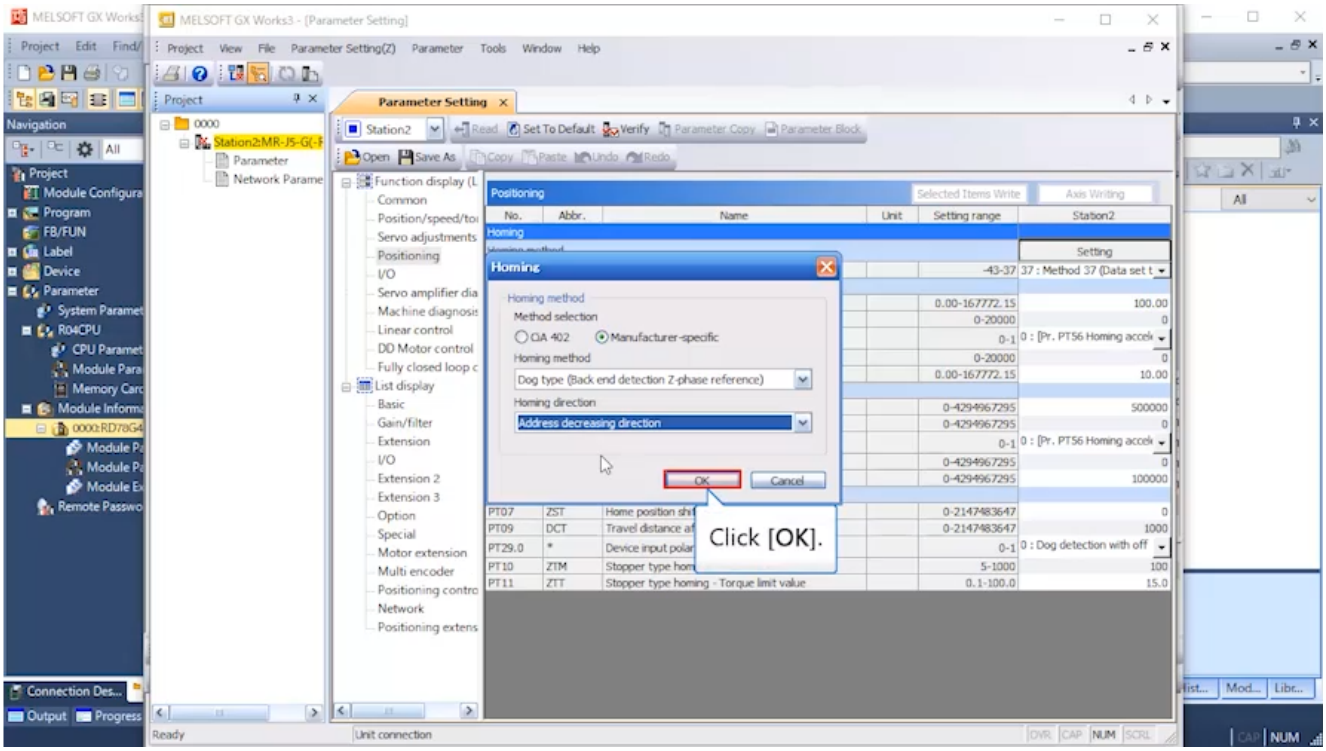
3.3.2 Module parameter (Network)

(3)Remote station setting (continue)



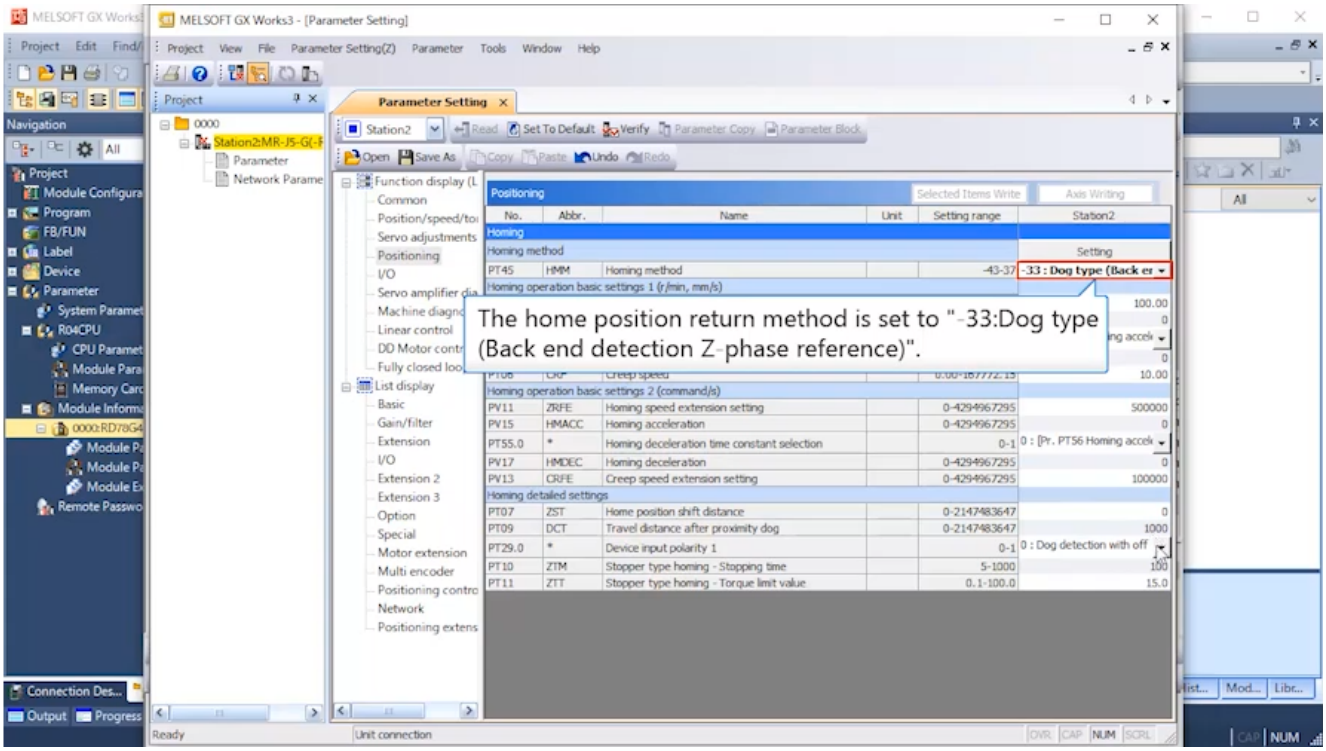
3.3.2 Module parameter (Network)

(3)Remote station setting (continue)



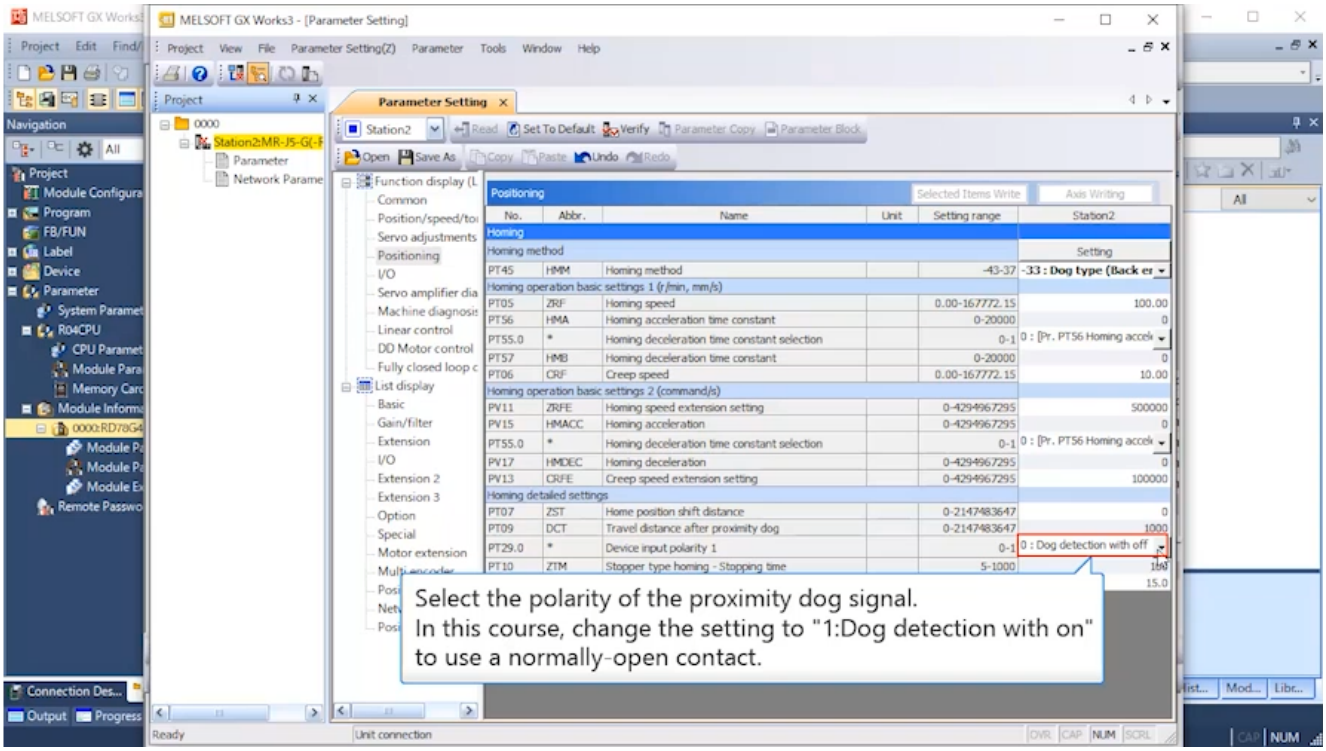
3.3.2 Module parameter (Network)

(3)Remote station setting (continue)



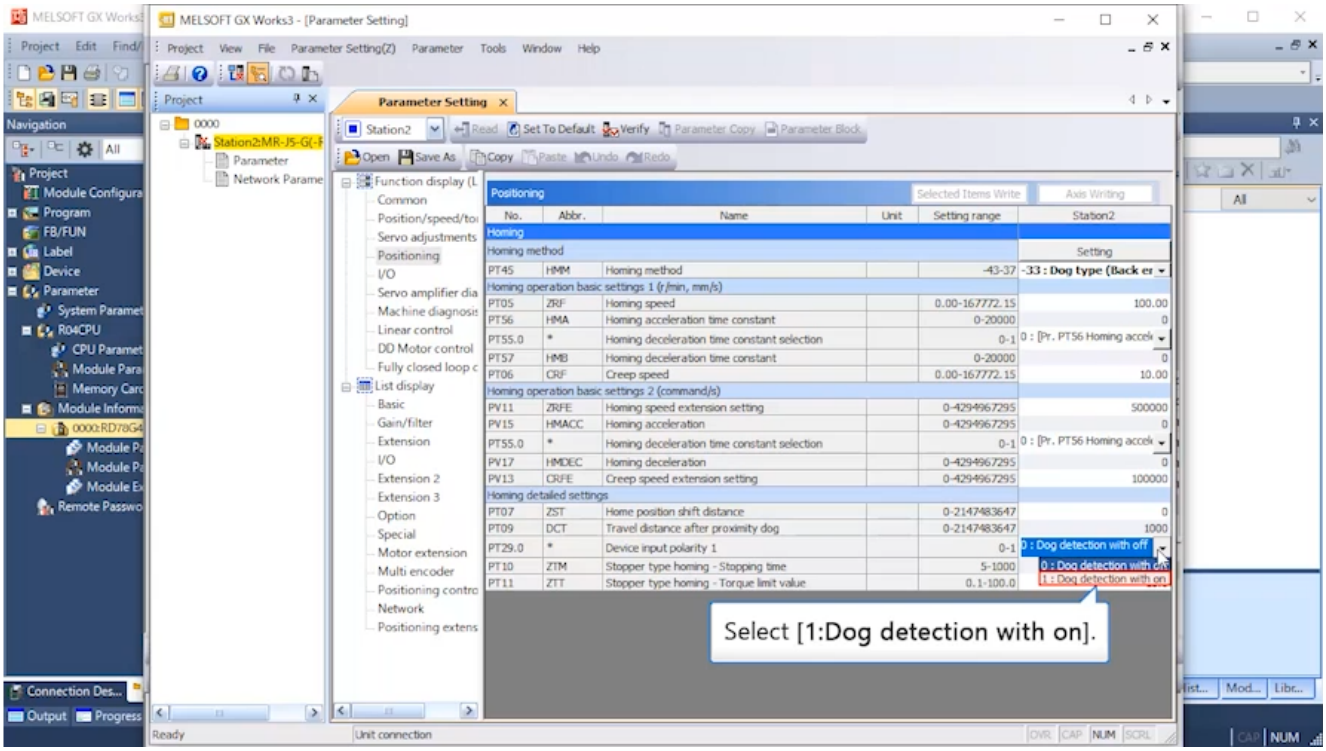
3.3.2 Module parameter (Network)

(3)Remote station setting (continue)



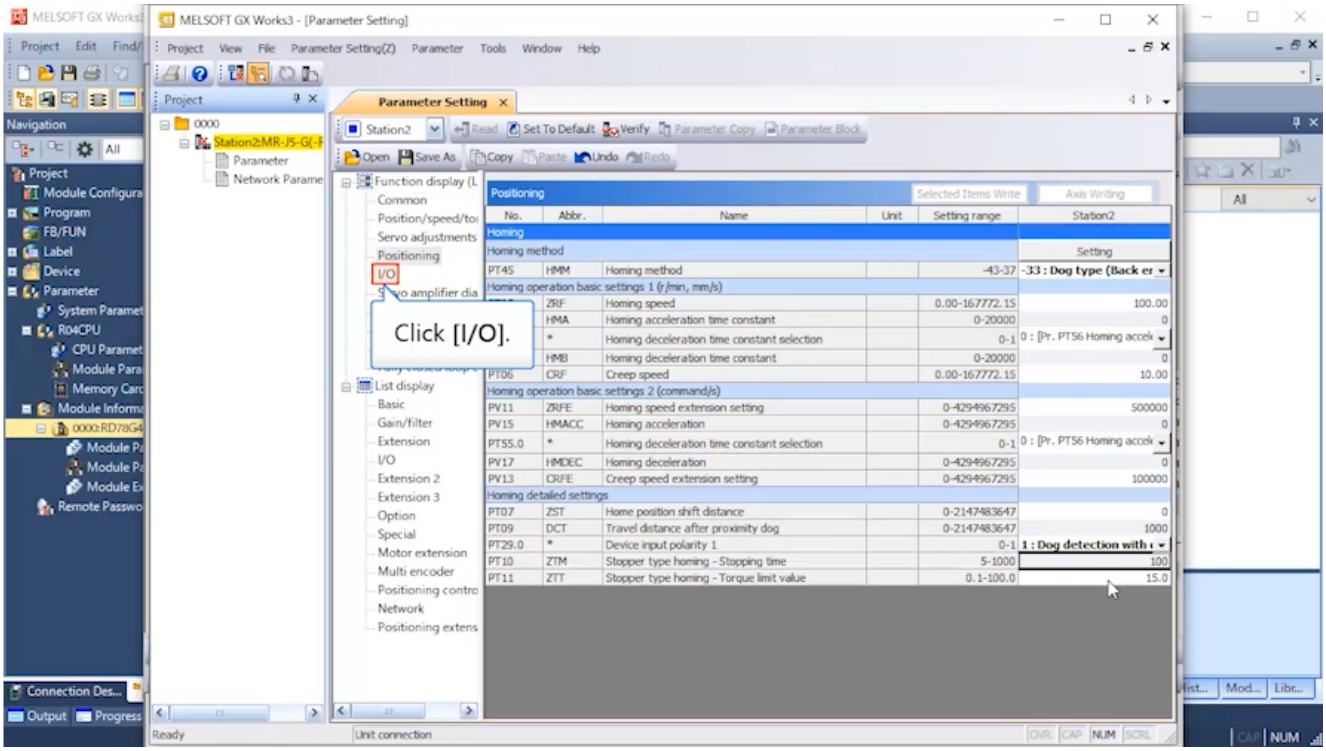
3.3.2 Module parameter (Network)

(3)Remote station setting (continue)



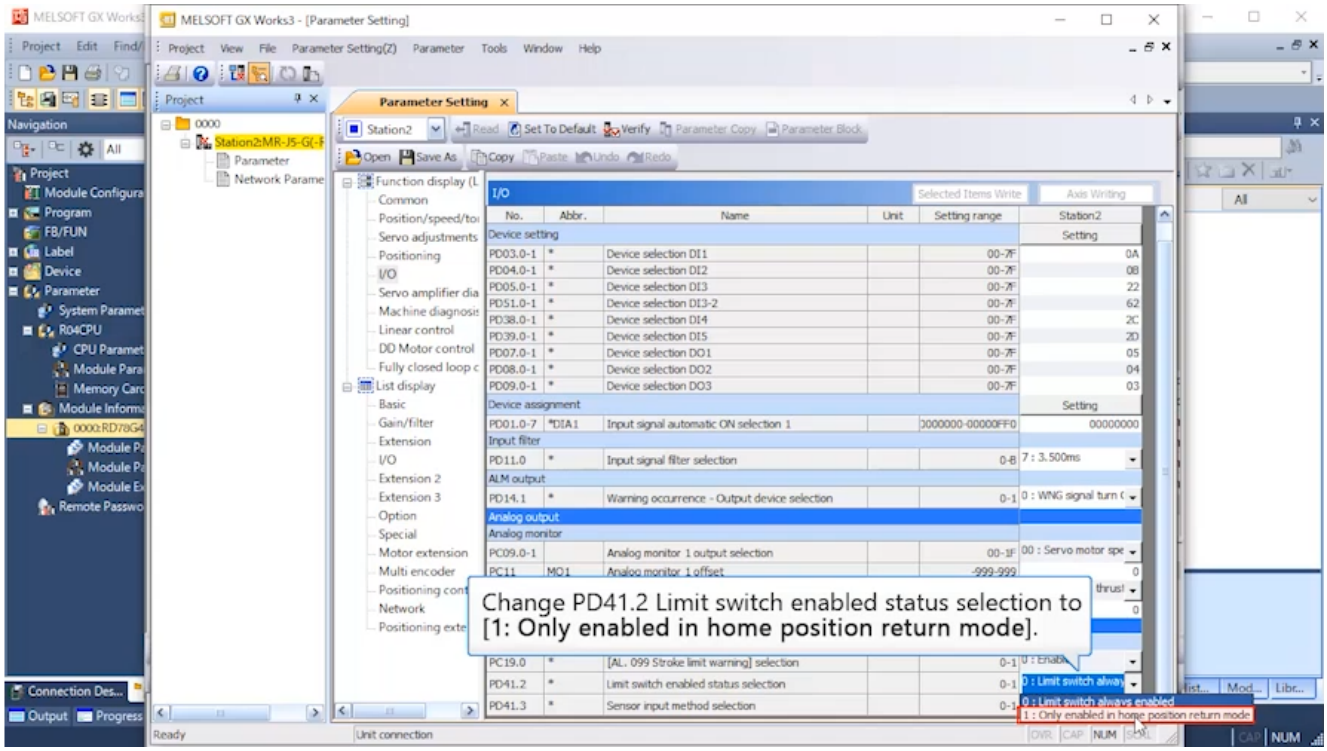
3.3.2 Module parameter (Network)

(3)Remote station setting (continue)



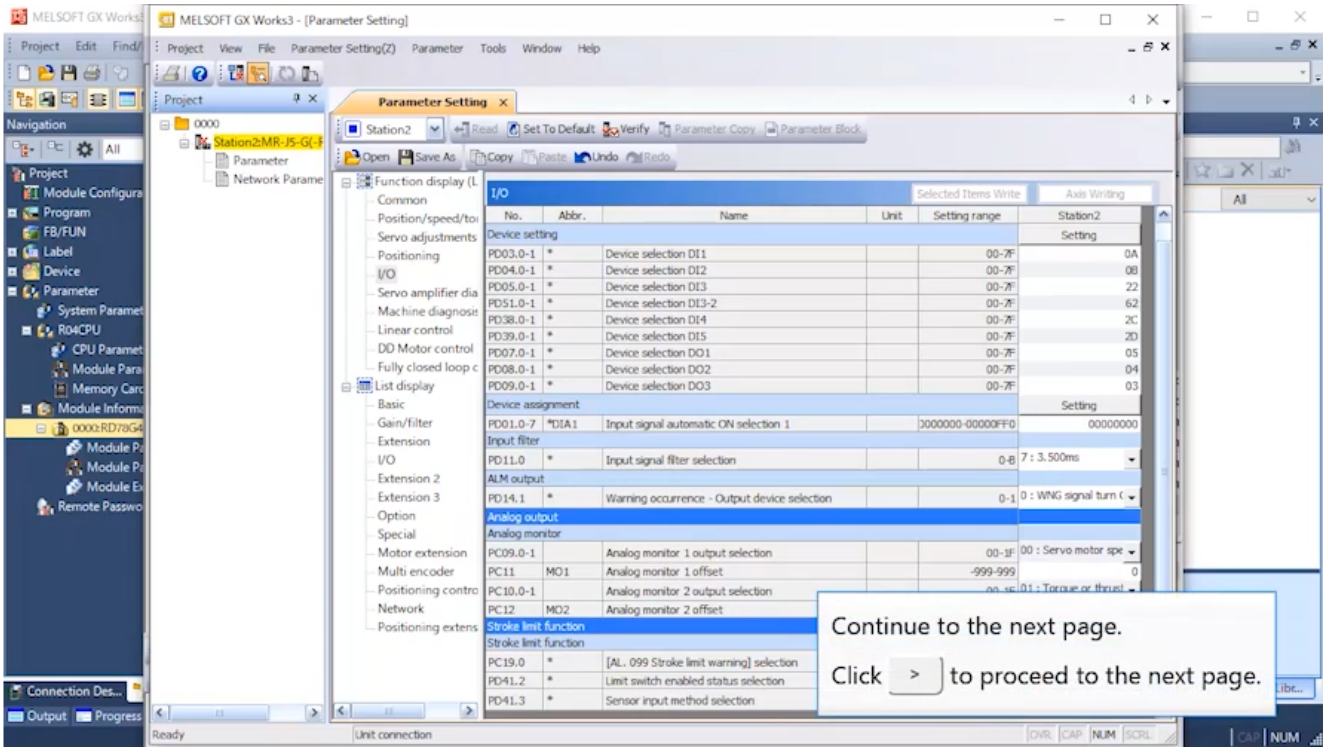
3.3.2 Module parameter (Network)

(3)Remote station setting (continue)



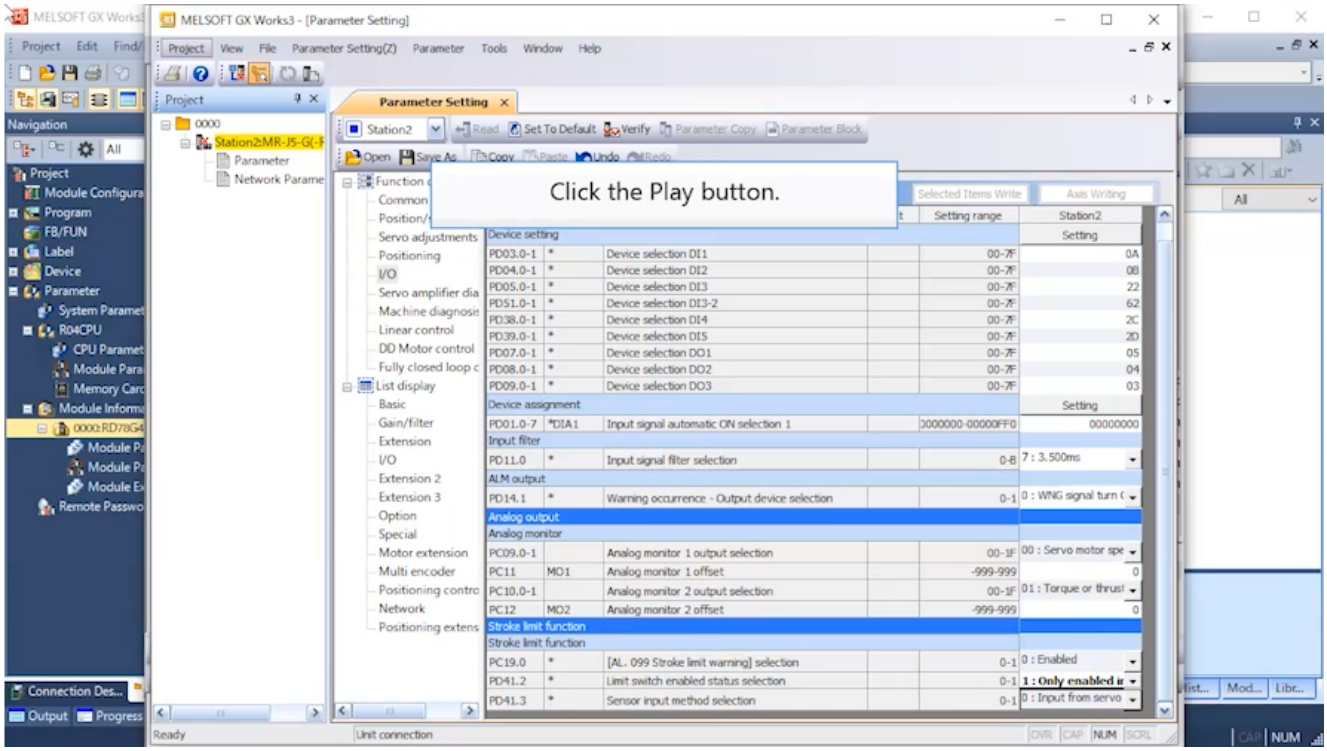
3.3.2 Module parameter (Network)

(3)Remote station setting (continue)



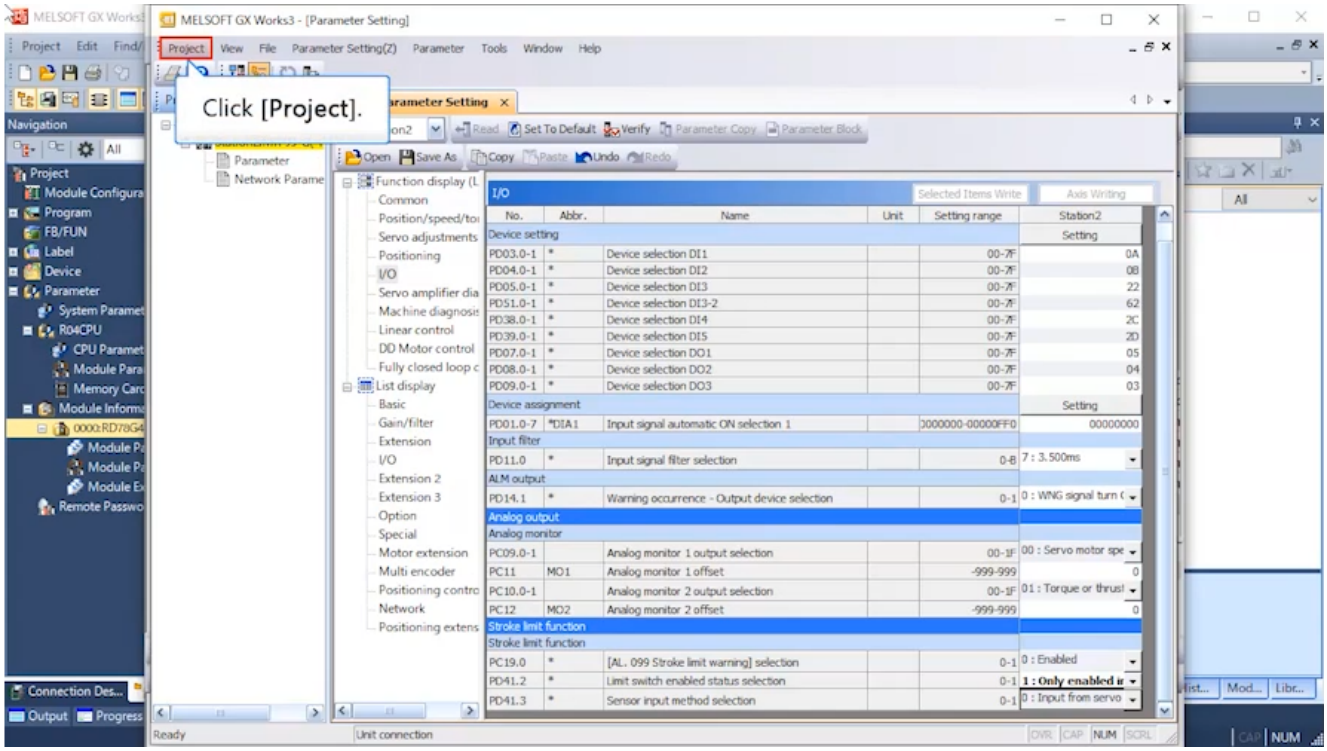
3.3.2 Module parameter (Network)

(3)Remote station setting (continue)



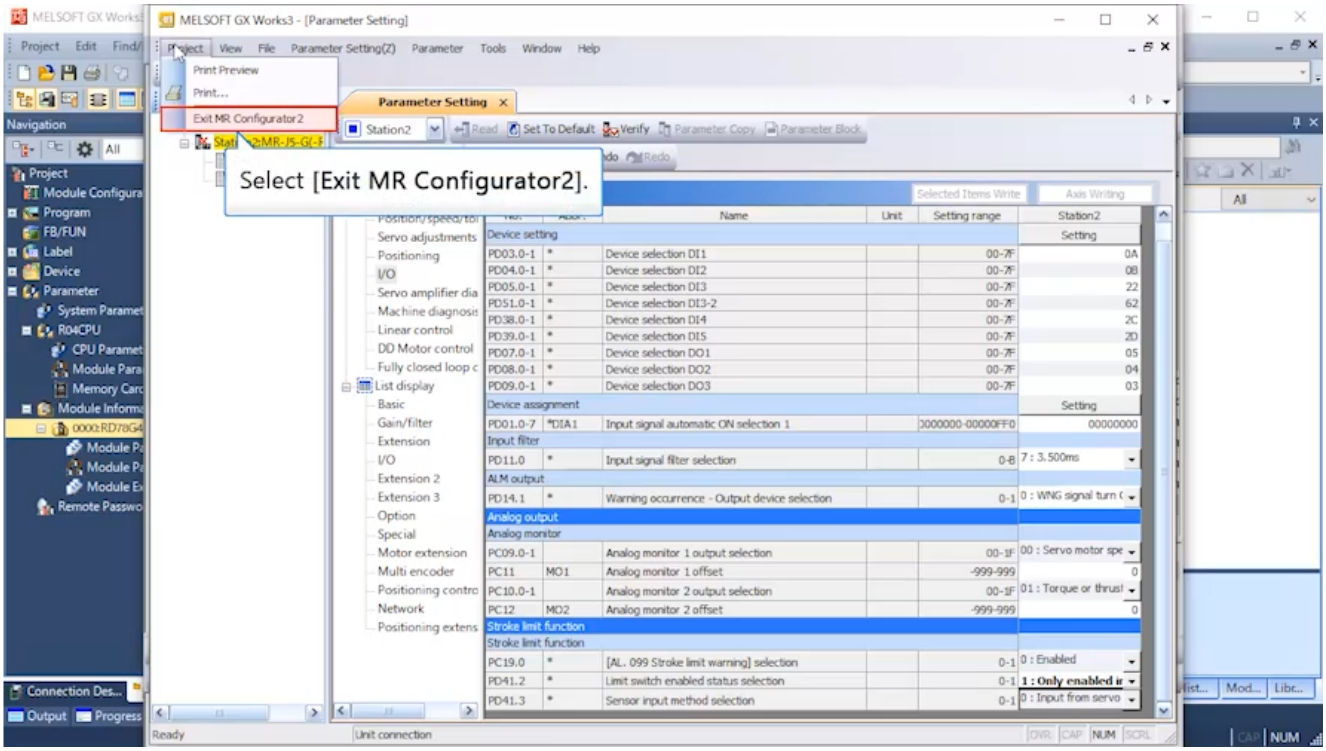
3.3.2 Module parameter (Network)

(3)Remote station setting (continue)



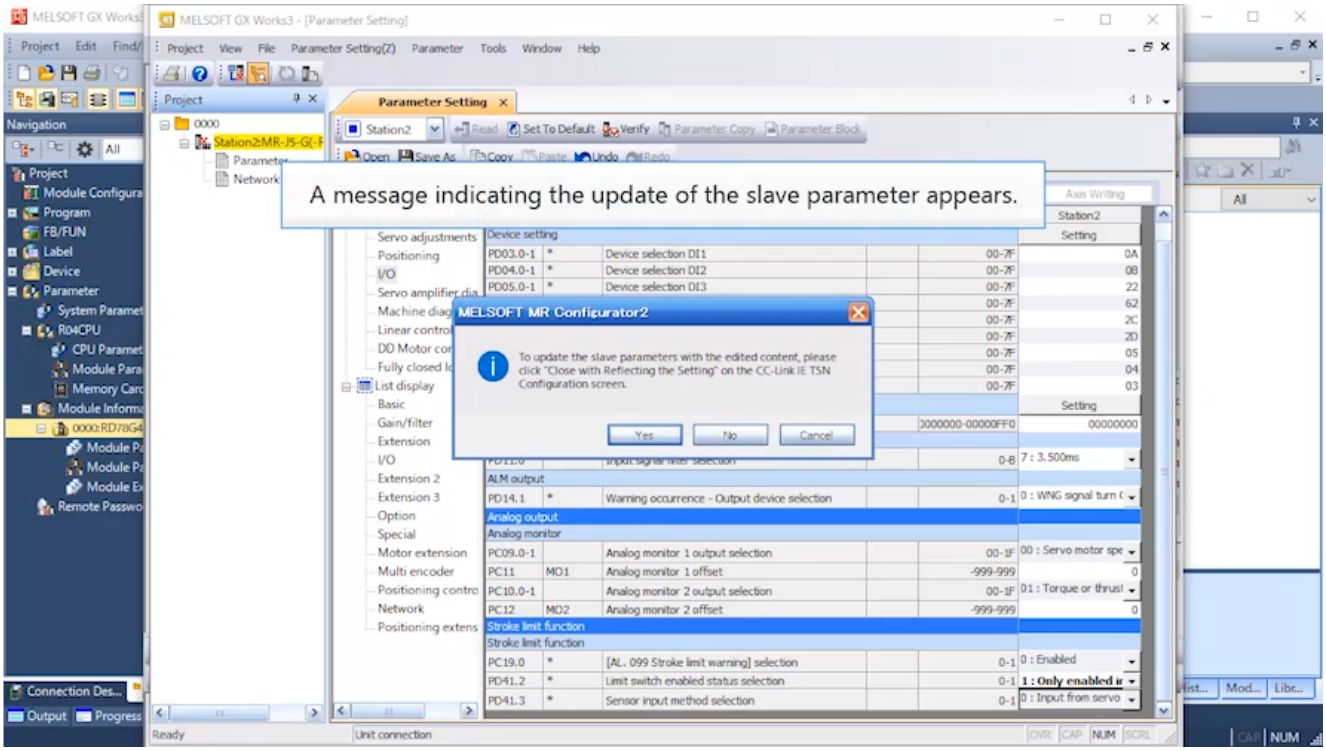
3.3.2 Module parameter (Network)

(3)Remote station setting (continue)



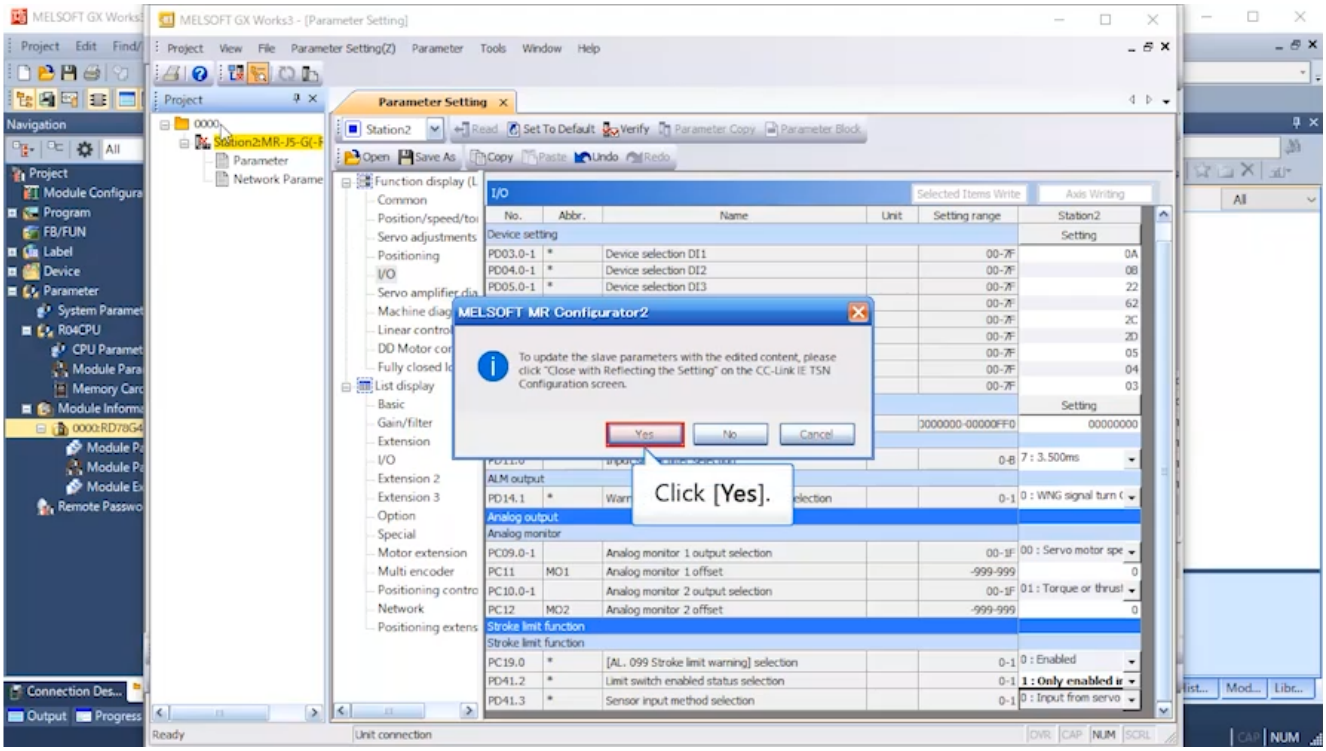
3.3.2 Module parameter (Network)

(3)Remote station setting (continue)



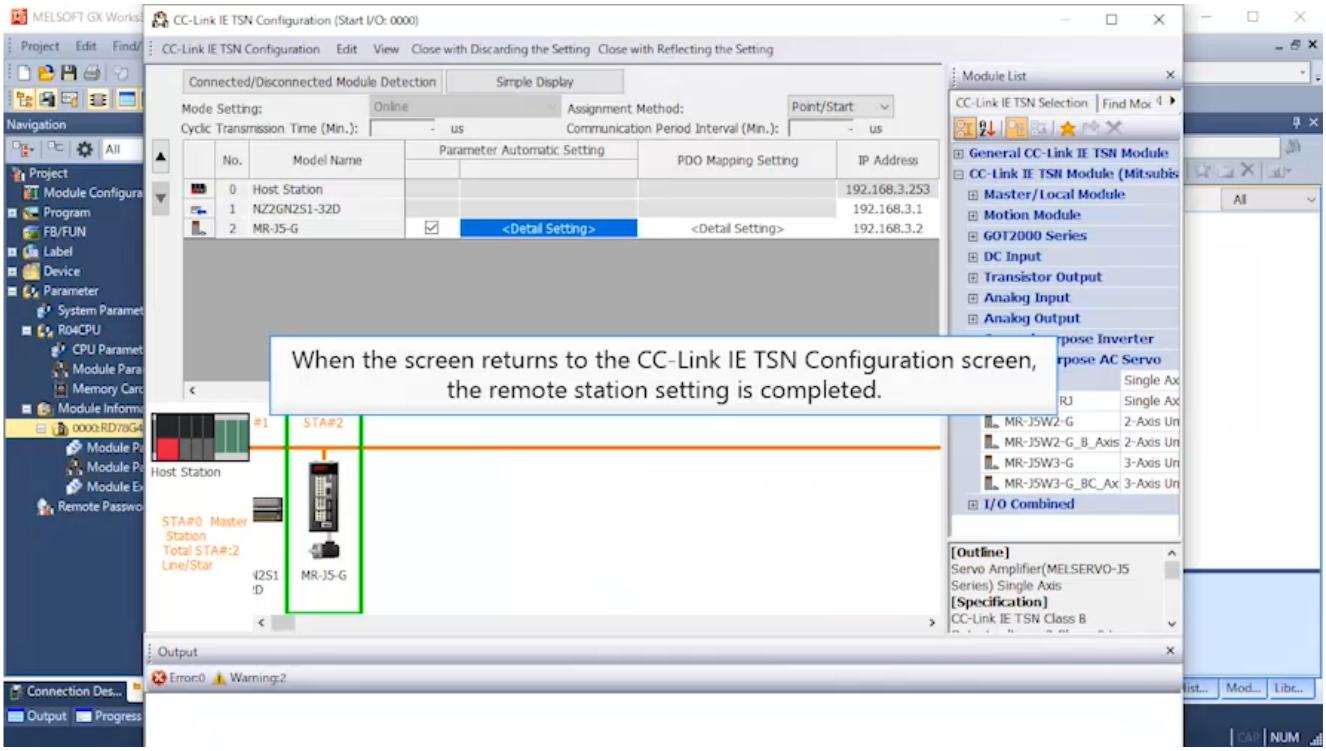
3.3.2 Module parameter (Network)

(3)Remote station setting (continue)



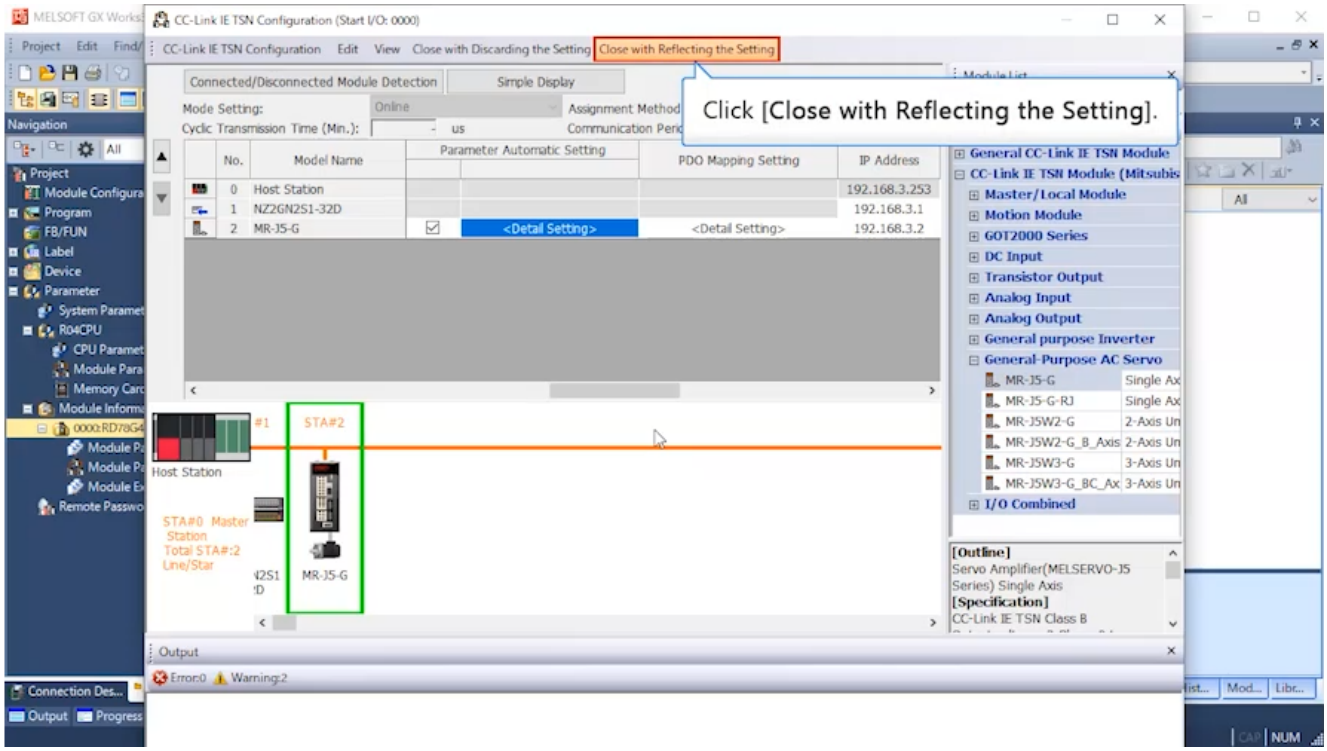
3.3.2 Module parameter (Network)

(3)Remote station setting (continue)



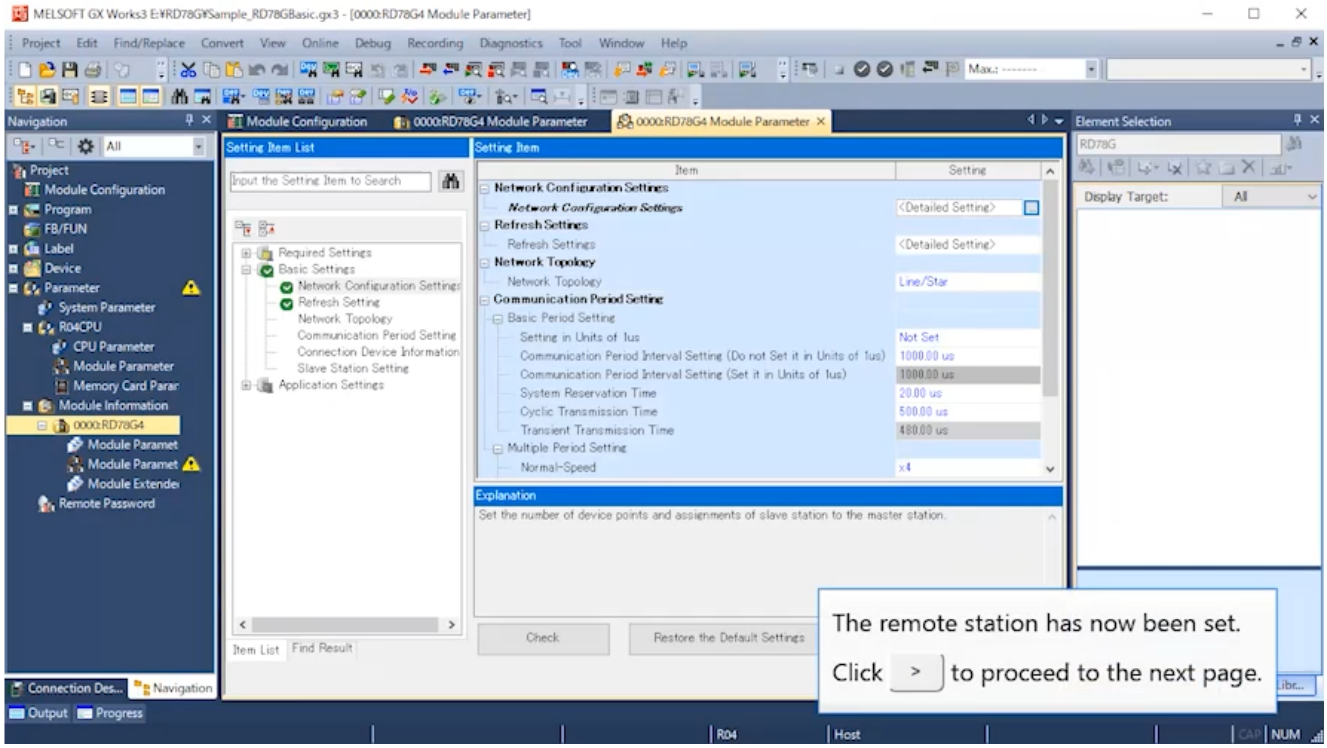
3.3.2 Module parameter (Network)

(3)Remote station setting (continue)



3.3.2 Module parameter (Network)

(3)Remote station setting (continue)



3.3.2 Module parameter (Network)

(4) PDO mapping

PDO is the abbreviation for Process Data Object, which is one of the communication profiles of CANopen objects.
The PDO communication is equivalent to the existing CC-Link cyclic communication.
It allows OD (Object Dictionary) to be directly operated.

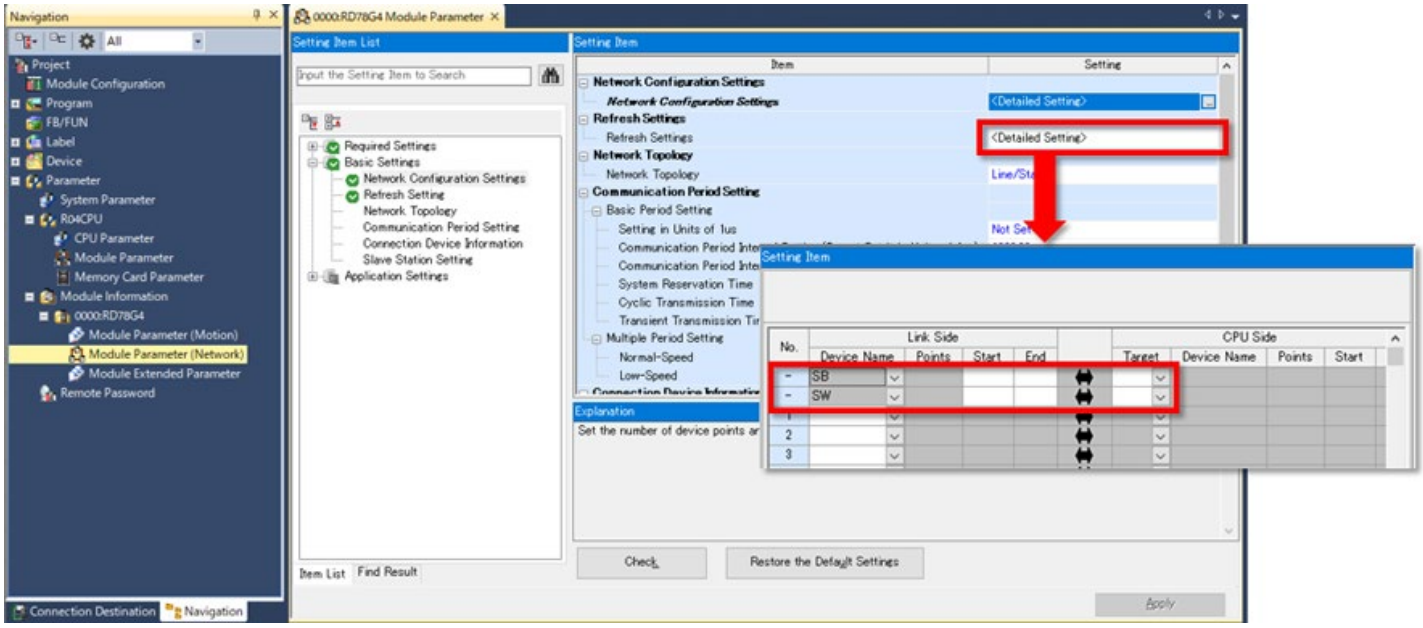
The PDO mapping means mapping (relating) the data to be exchanged between the controller and slave in the cyclic communication (PDO communication) in advance.

When adding sleeve stations or changing the IP address, perform the PDO mapping again.

Digital Inputs has been added to the PDO mapping. This sets the input signal status of the servo amplifier to be transmitted to the motion module by the cyclic communication.

3.3.2 Module parameter (Network)

(5) Refresh setting
Double-click [Refresh Settings] → <Detailed Setting>.
All the setting fields must be blank.



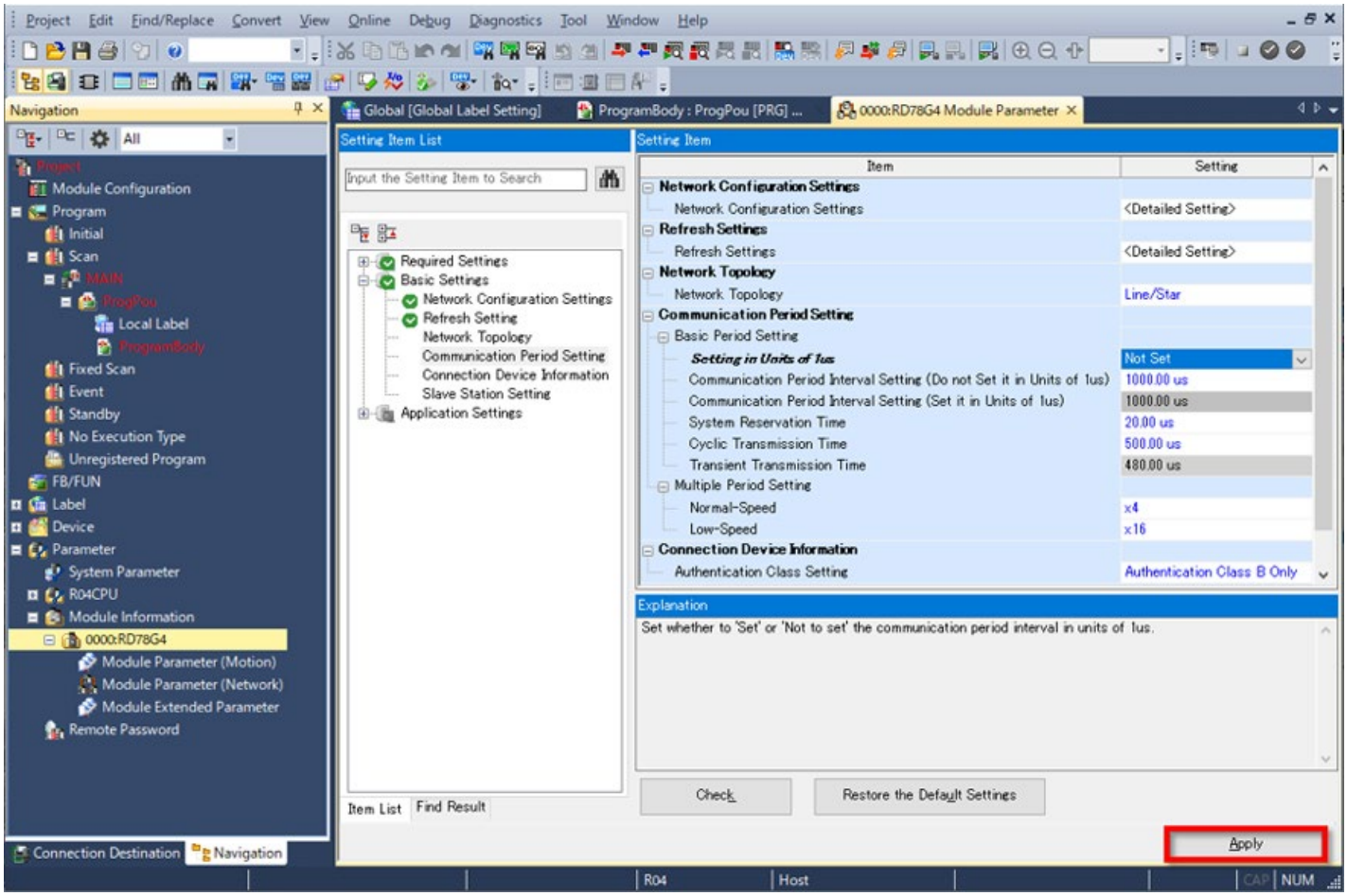
(Note) When the module label is set to [Not use], the setting field of the refresh target is blank from the beginning.

3.3.2 Module parameter (Network)

(6) Confirming module parameters

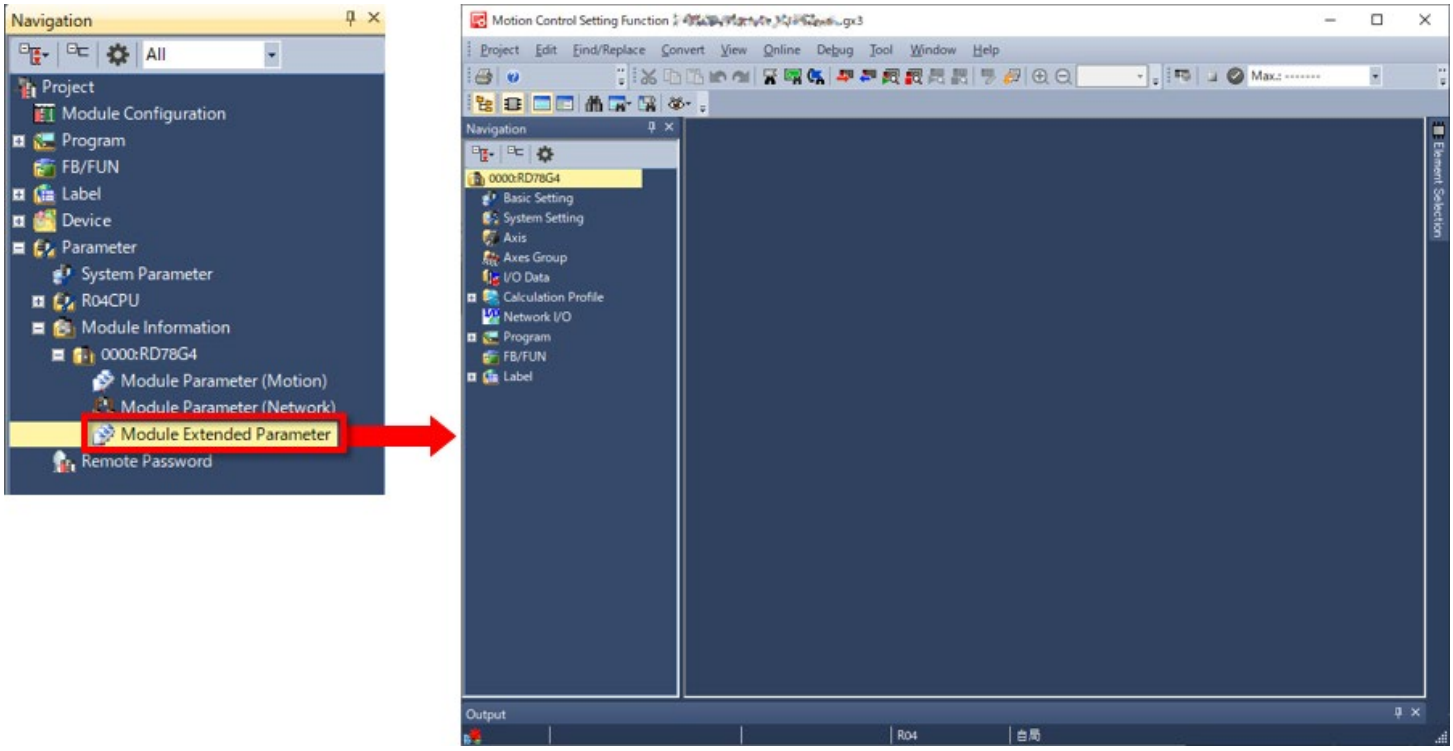
When the screen returns to the main screen of GX Works3, confirm the parameters that have been set.

Be sure to click the [Apply] button on the lower right of the screen.

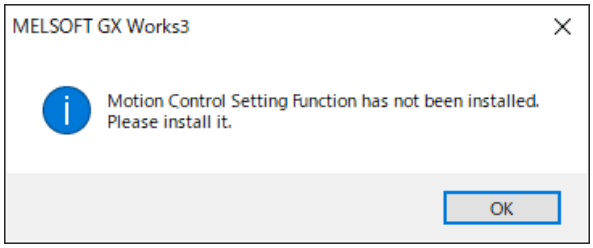


3.3.3 Module extended parameter

Double-click [Parameter] → [Module Information] → [0000:RD78G4] → [Module Extended Parameter]in the project tree.
The Motion Control Setting Function screen appears.
Program the motion module on this screen.
For the actual programs, refer to Chapter 4.



When Motion Control Setting is not started and the following message is displayed, Motion Control Setting is not installed in the personal computer used(*).

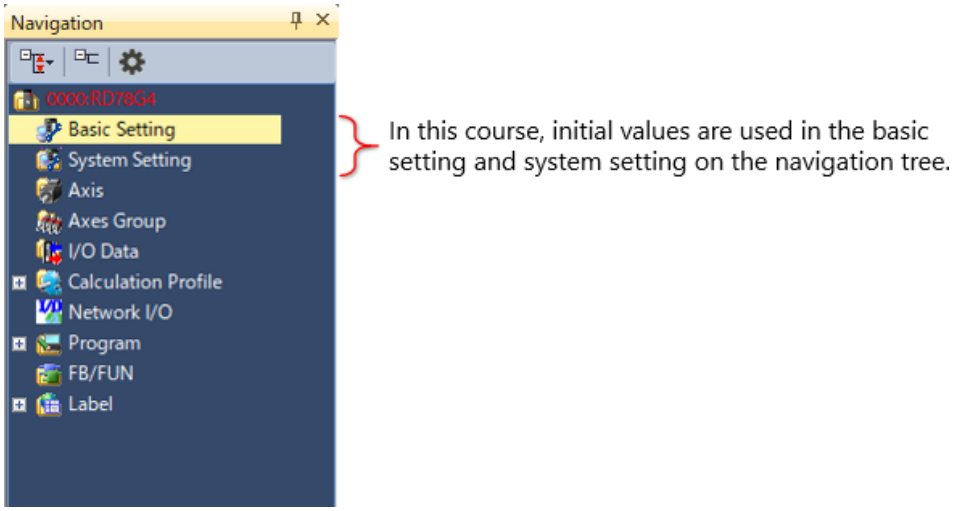


Please install Motion Control Setting Function.

(*) Indicates a personal computer running Windows®.

3.4 Motion Control Setting Function

This section describes the setting items required for the motion control setting function.

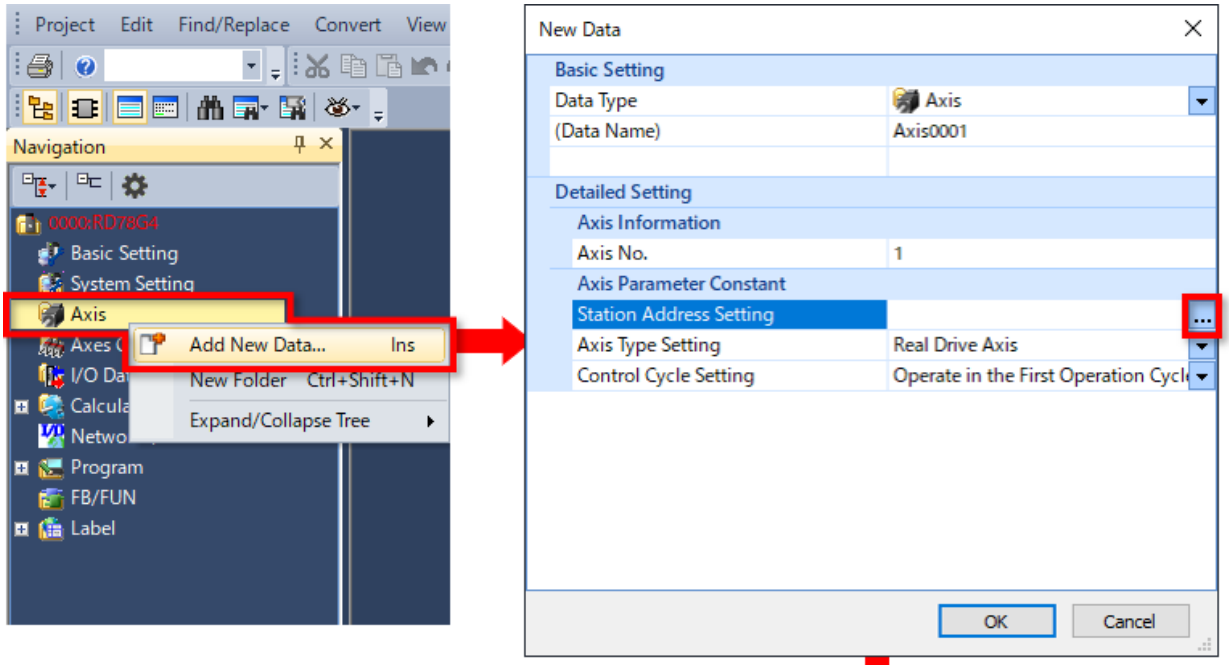


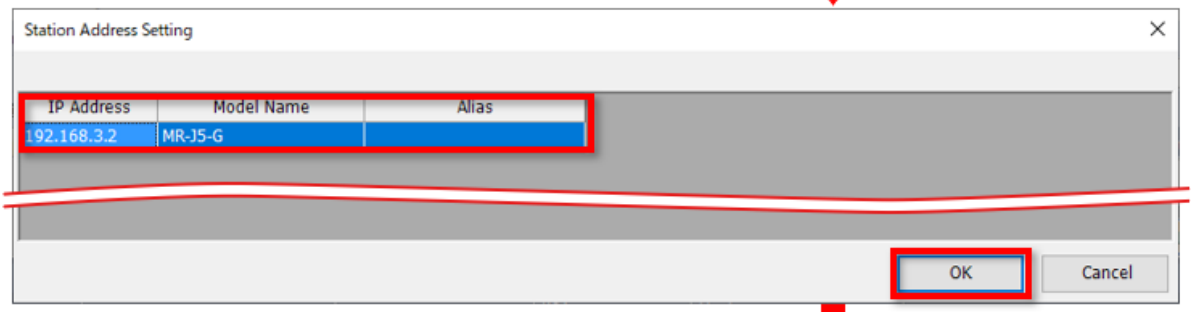
3.4.1 Axis registration

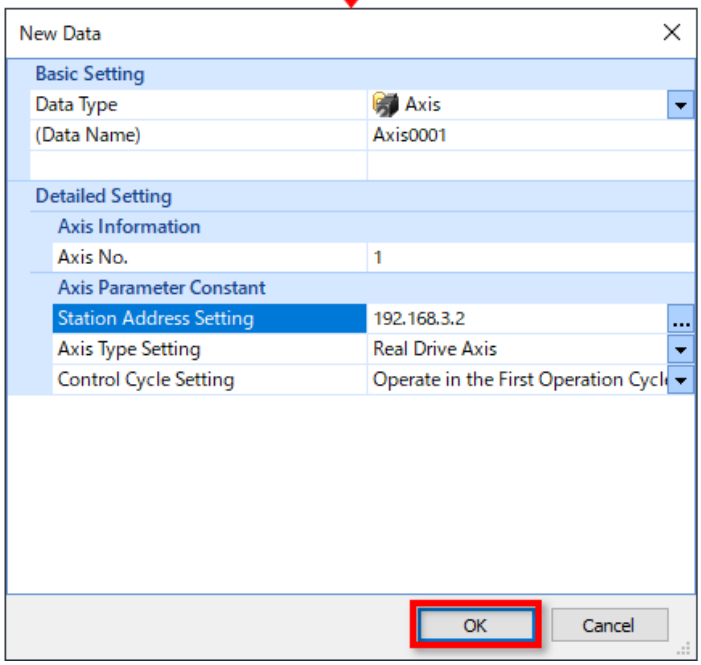
(1) Creating a new axis

Right-click [Axis] in the Navigation tree and select [Add New Data].

After the New Data window appears, set the items as shown below.





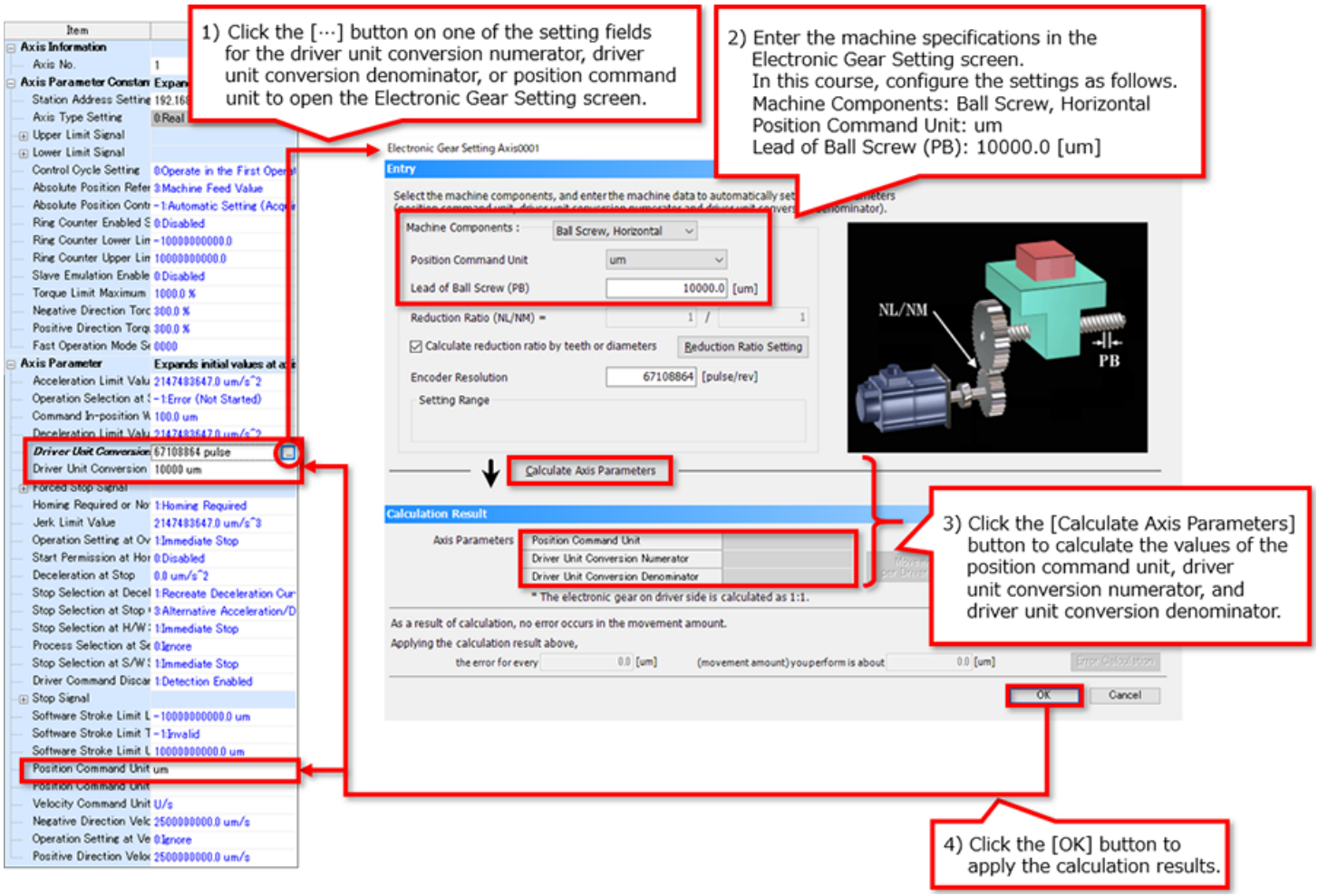


(2) Setting the driver unit conversion

The Axis Parameter Setting tab opens.

Mainly set the command unit, electronic gear, and limit values here.

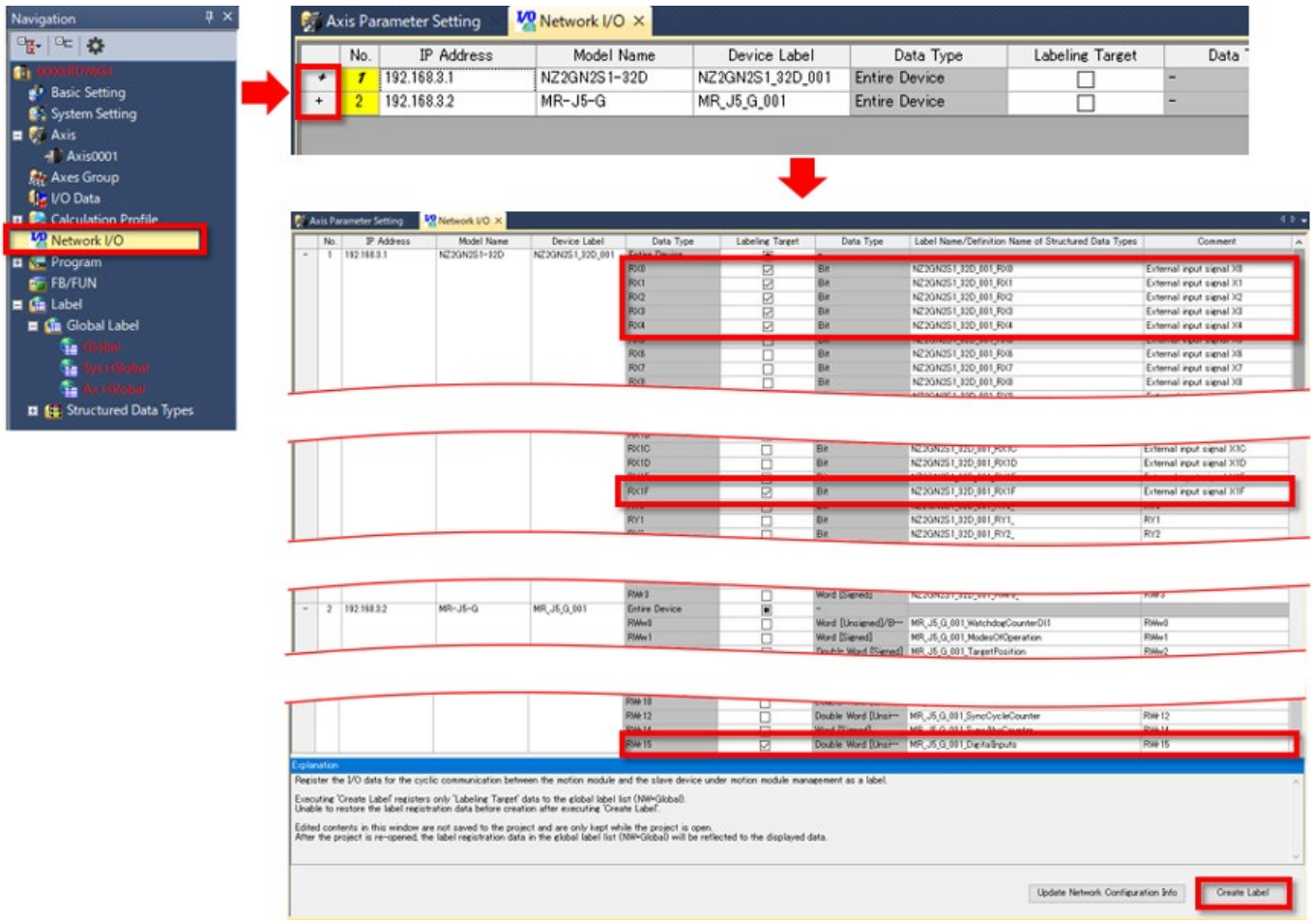
In this course, change the items in the red flame in the figure below.



3.4.2 Network I/O

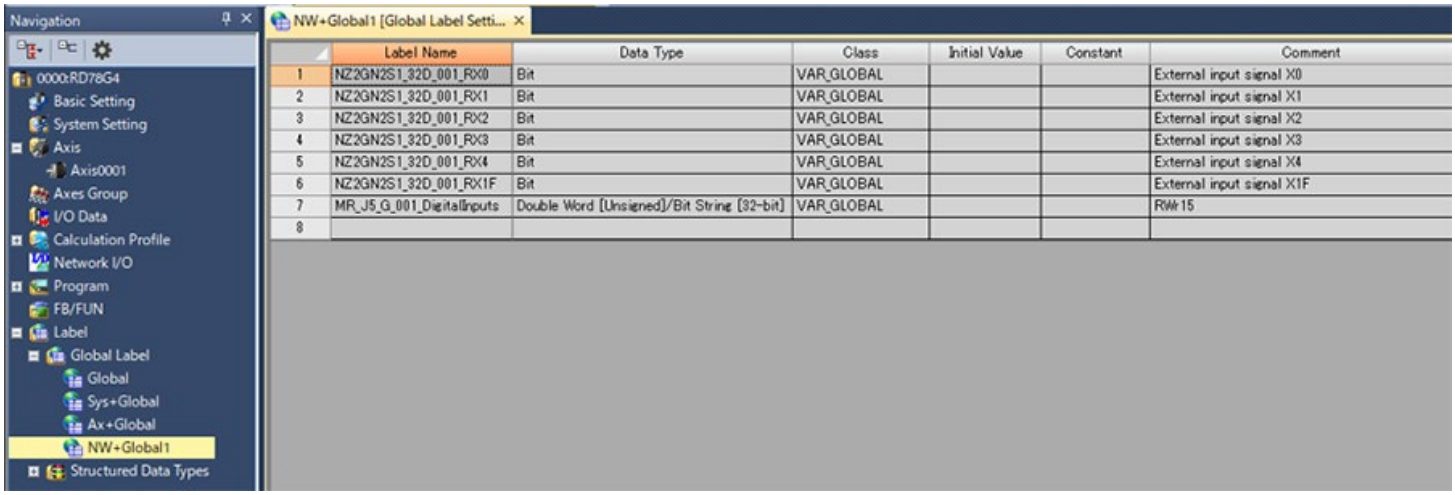
When using the remote I/O module, slave labels must be created from the network I/O.

1) Double-click [Network I/O] in the Navigation tree.
2) After the Network I/O tab opens, click the "+" marks on the left side of the lines of the remote input module and MR-J5-G.
3) Select the data for labeling. Select the following items in this course.
   - RX0 to RX4 and RX1F of the NZ2GN2S1-32D
   - RWr15 of the MR-J5-G
4) Click [Create Label] to create slave labels of the selected data.



3.4.2 Network I/O

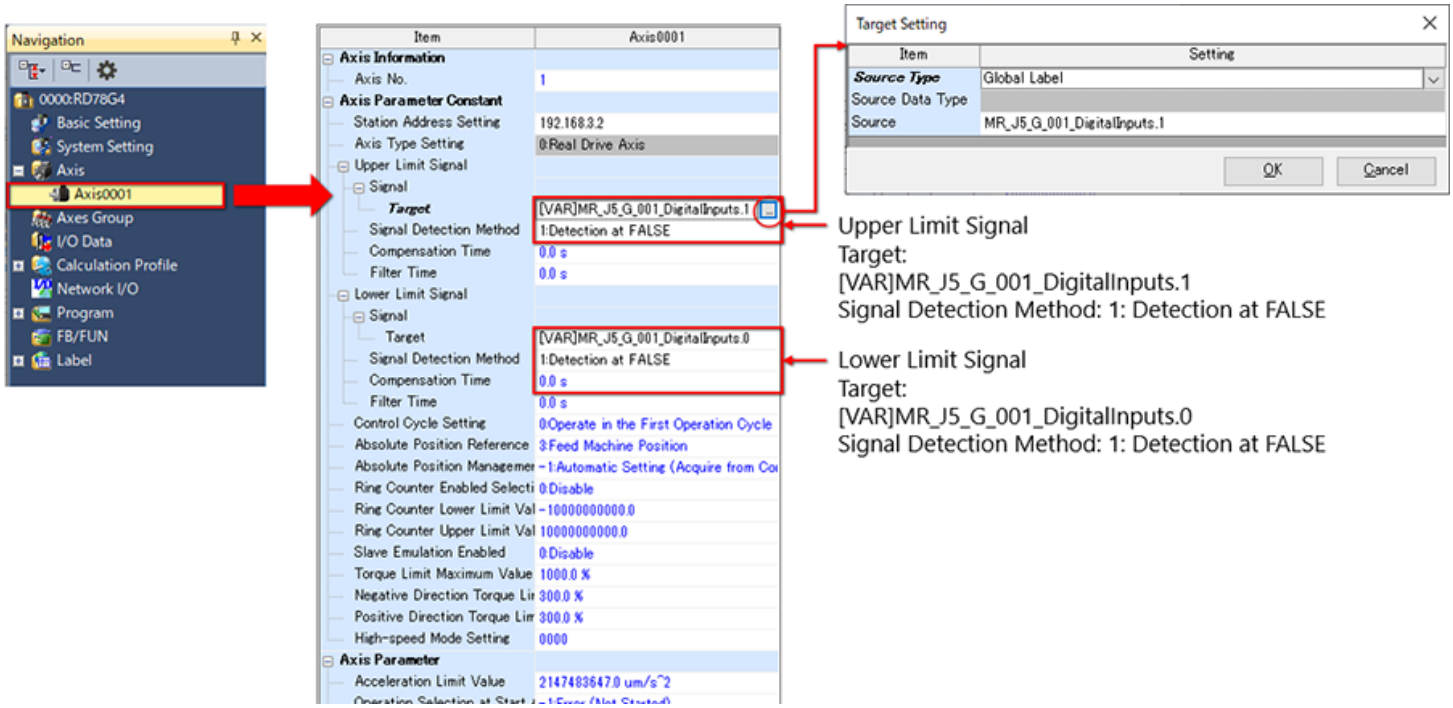
The created slave labels are registered in [Label] → [Global Label] → [NW+Global1]in the Navigation tree.



	Label Name	Data Type	Class	Initial Value	Constant	Comment
1	NZ2GN2S1_32D_001_RX0	Bit	VAR_GLOBAL			External input signal X0
2	NZ2GN2S1_32D_001_RX1	Bit	VAR_GLOBAL			External input signal X1
3	NZ2GN2S1_32D_001_RX2	Bit	VAR_GLOBAL			External input signal X2
4	NZ2GN2S1_32D_001_RX3	Bit	VAR_GLOBAL			External input signal X3
5	NZ2GN2S1_32D_001_RX4	Bit	VAR_GLOBAL			External input signal X4
6	NZ2GN2S1_32D_001_RX1F	Bit	VAR_GLOBAL			External input signal X1F
7	MR_J5_G_001_DigitalInputs	Double Word [Unsigned]/Bit String [32-bit]	VAR_GLOBAL			RWr15
8						

3.4.3 Limit signal setting

After the slave labels are created, double-click
"Axis0001" in the Navigation tree to display the Axis Parameter Setting tab again.
Set the lower limit and upper limit as shown in the figure below.
In the target field, the icon for displaying the input auxiliary window is displayed.



3.5 Summary of This Chapter

In this chapter, you have learned:

- Creating a New Project
- PLC CPU Settings
- Motion Module Setting
- Motion Control Setting Function

Important points

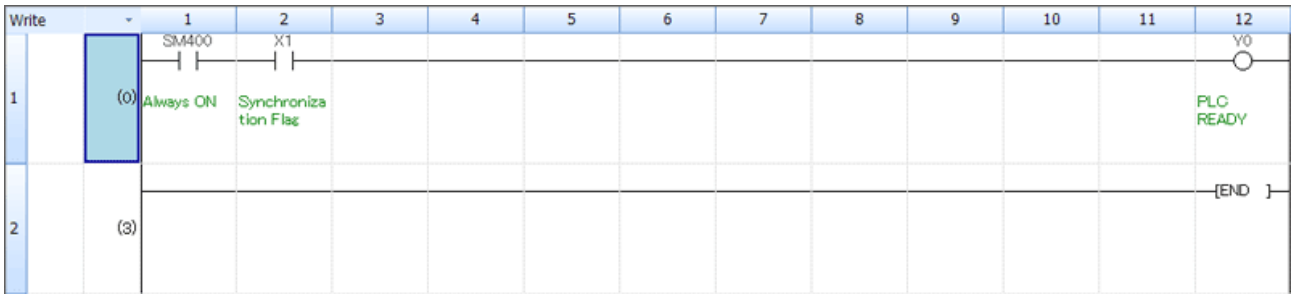
Creating a New Project	• Create a project of GX Works3 and create a module configuration diagram.
PLC CPU Settings	• Change the link direct device setting to the extended mode (iQ-R series mode).
Motion Module Setting	• In the Module Parameter (Network) screen, set the network configuration and parameters of the remote station. • In the network configuration setting, add a remote station, set the IP address, and perform the PDO mapping. • Delete all the link refresh settings.
Motion Control Setting Function	• In the Motion Control Setting Function screen, register the axes. • The slave labels are created from the network I/O.

Chapter 4 Sample Program and Operation Check

This chapter describes the sample program and how to use Motion Control FB.
Open the sample program file downloaded in Chapter 3.

4.1 PLC CPU Program

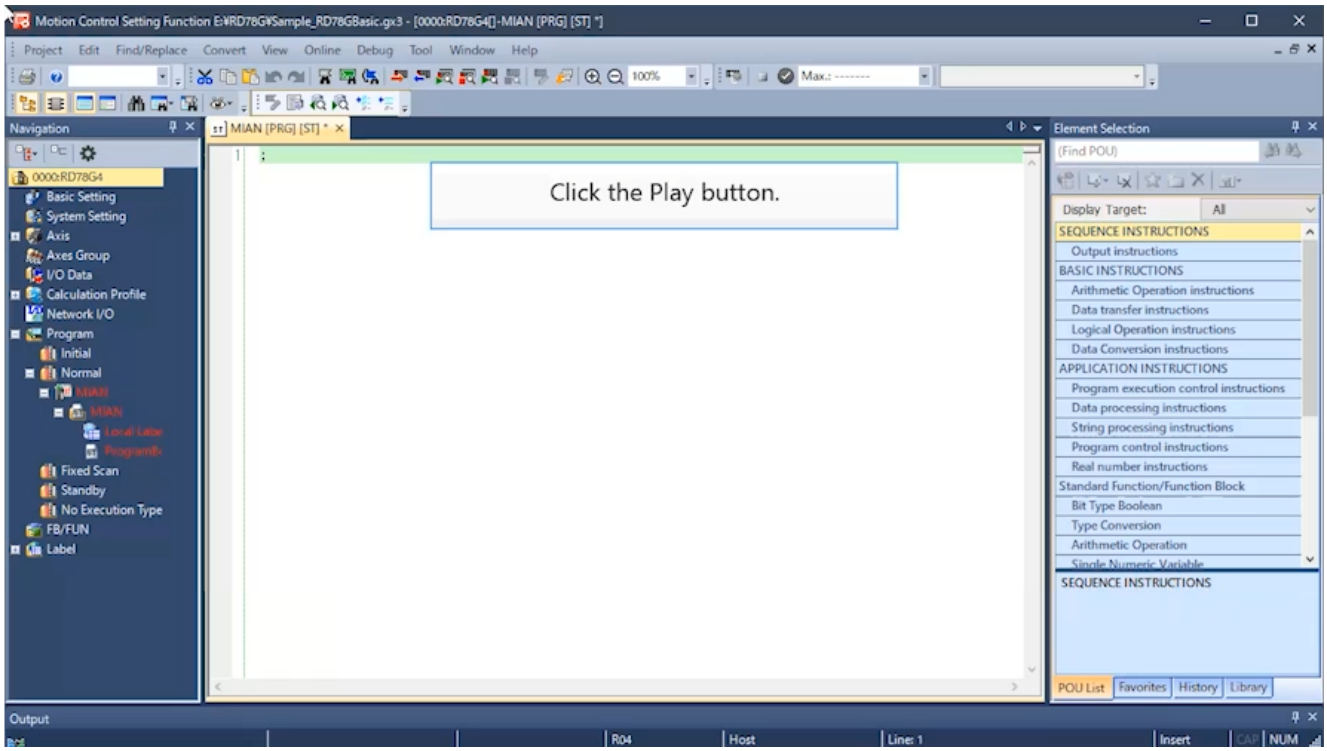
Double-click [Program] → [Scan] → [MAIN] → [ProgPou] → [Program Body] in the project tree to open the program.
Before using the RD78G motion module, always turn on [Y0: PLC READY] in the PLC CPU program.
The sample program uses Ladder.



4.2 Motion Module Program

Double-click the module extended parameter to display the [Motion Control Setting Function] screen.
Double-click [Program] → [Normal] → [MAIN] → [ProgramBody] in the project tree on the Motion Control Setting Function screen to open the program.

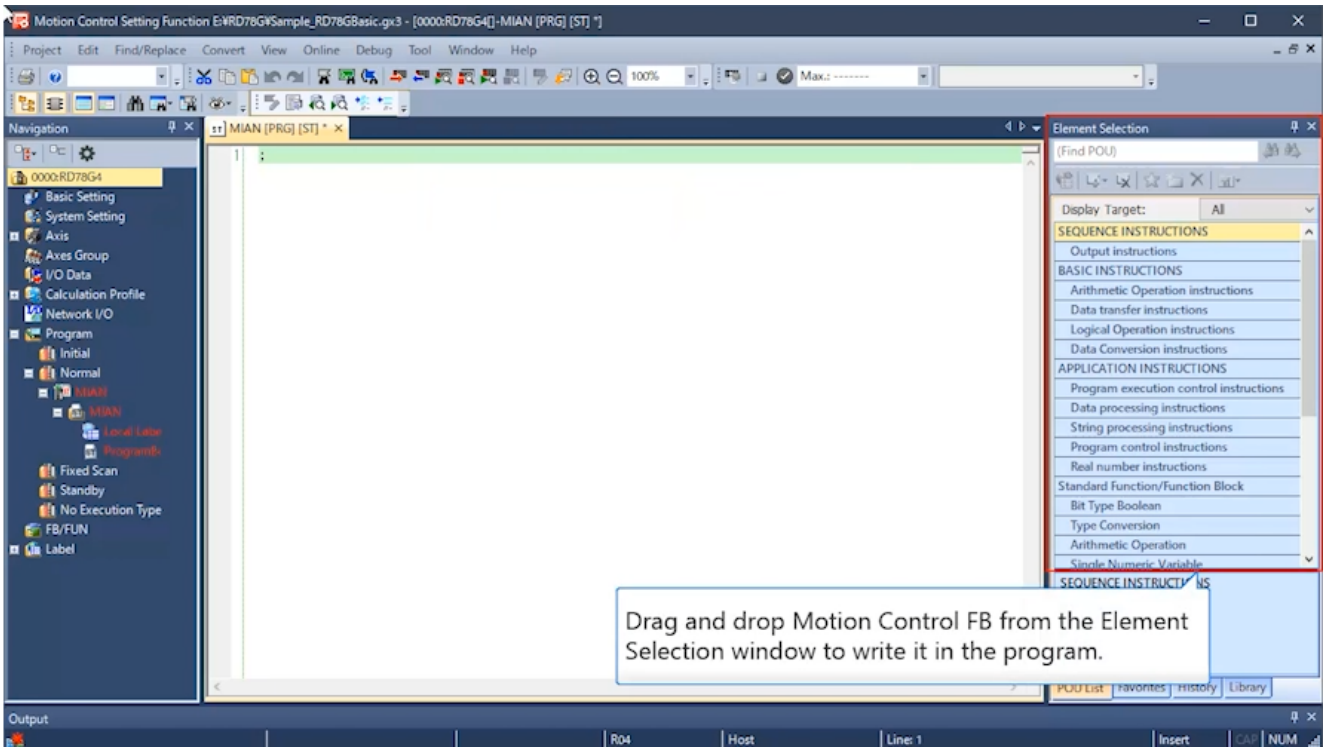
4.2.1 How to use Motion Control FB



4.2 Motion Module Program

Double-click the module extended parameter to display the [Motion Control Setting Function] screen.
Double-click [Program] → [Normal] → [MAIN] → [ProgramBody] in the project tree on the Motion Control Setting Function screen to open the program.

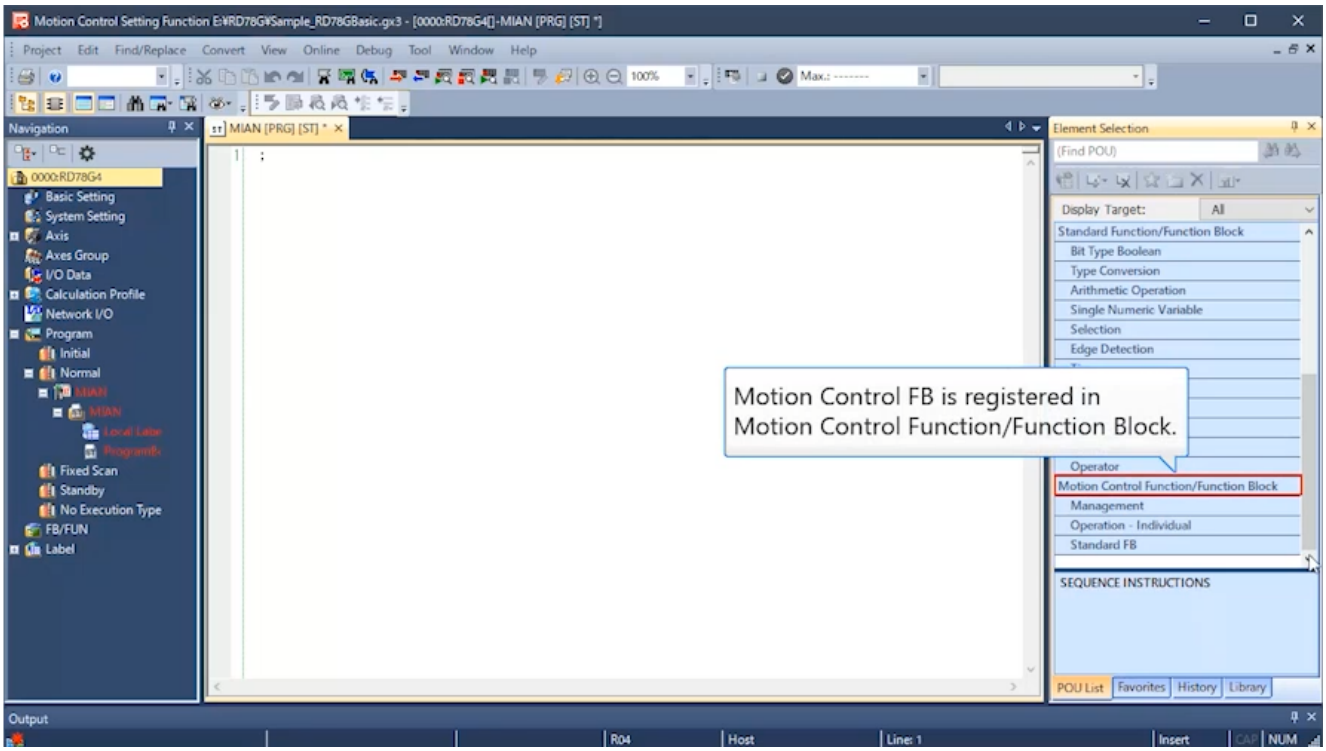
4.2.1 How to use Motion Control FB



4.2 Motion Module Program

Double-click the module extended parameter to display the [Motion Control Setting Function] screen.
Double-click [Program] → [Normal] → [MAIN] → [ProgramBody] in the project tree on the Motion Control Setting Function screen to open the program.

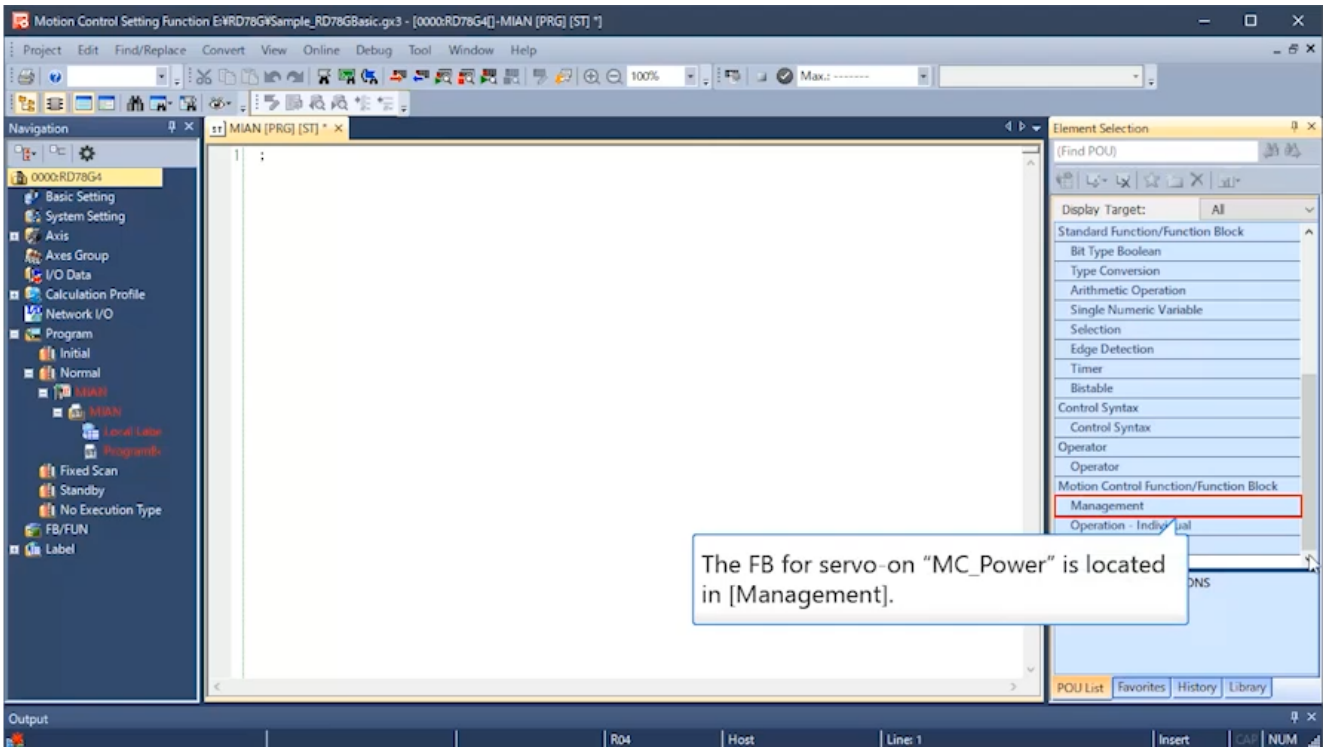
4.2.1 How to use Motion Control FB



4.2 Motion Module Program

Double-click the module extended parameter to display the [Motion Control Setting Function] screen.
Double-click [Program] → [Normal] → [MAIN] → [ProgramBody] in the project tree on the Motion Control Setting Function screen to open the program.

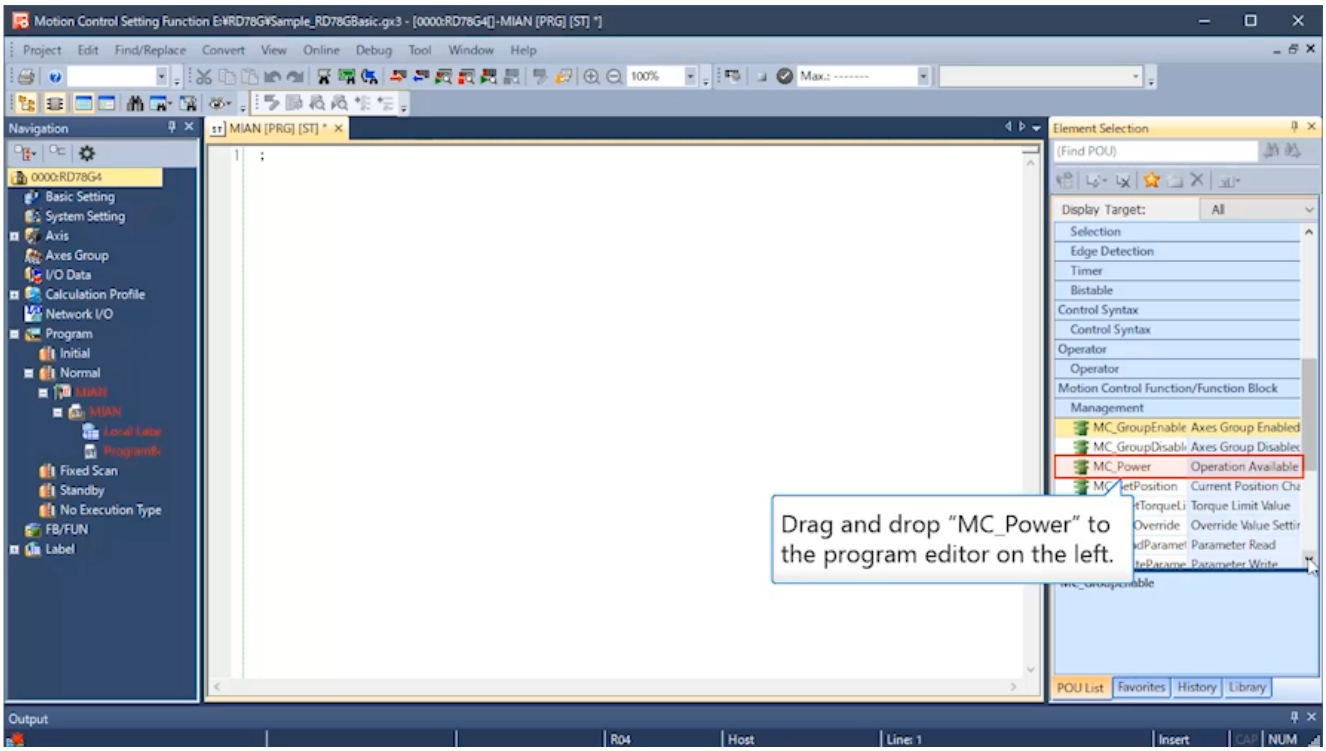
4.2.1 How to use Motion Control FB



4.2 Motion Module Program

Double-click the module extended parameter to display the [Motion Control Setting Function] screen.
Double-click [Program] → [Normal] → [MAIN] → [ProgramBody] in the project tree on the Motion Control Setting Function screen to open the program.

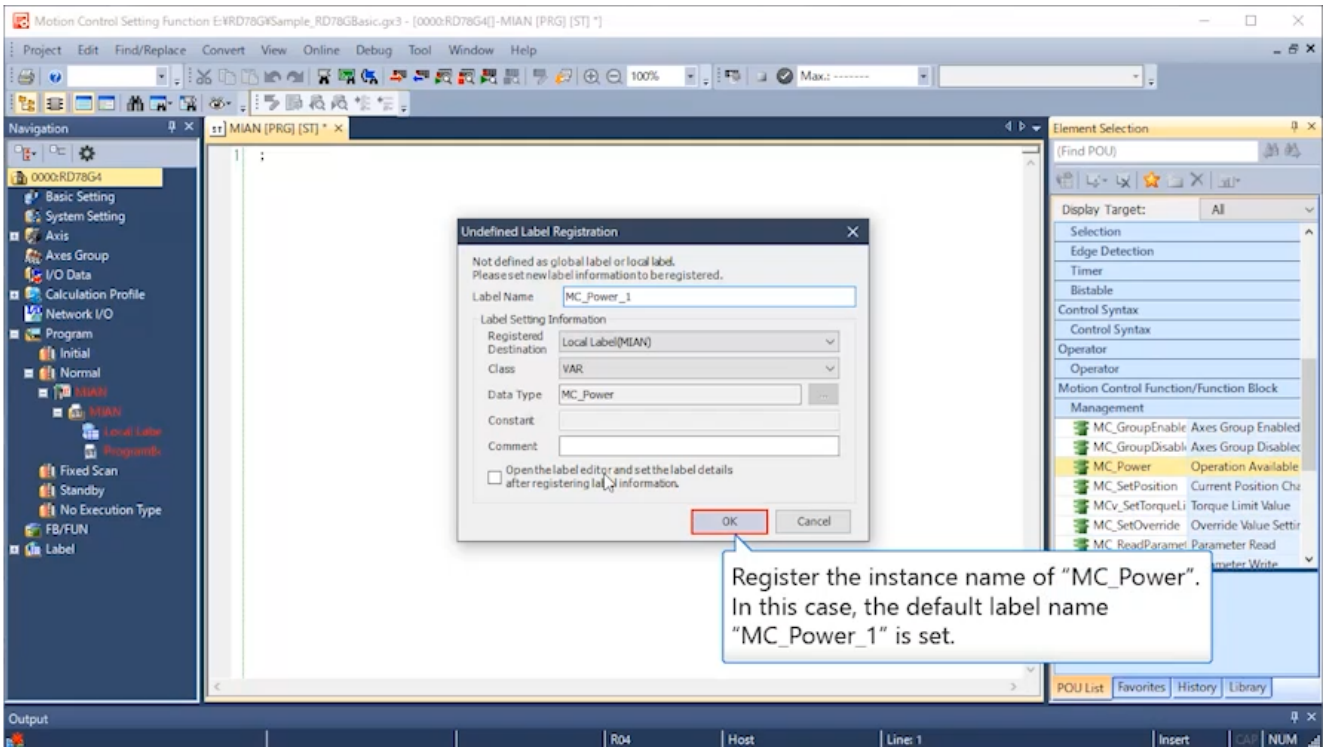
4.2.1 How to use Motion Control FB



4.2 Motion Module Program

Double-click the module extended parameter to display the [Motion Control Setting Function] screen.
Double-click [Program] → [Normal] → [MAIN] → [ProgramBody] in the project tree on the Motion Control Setting Function screen to open the program.

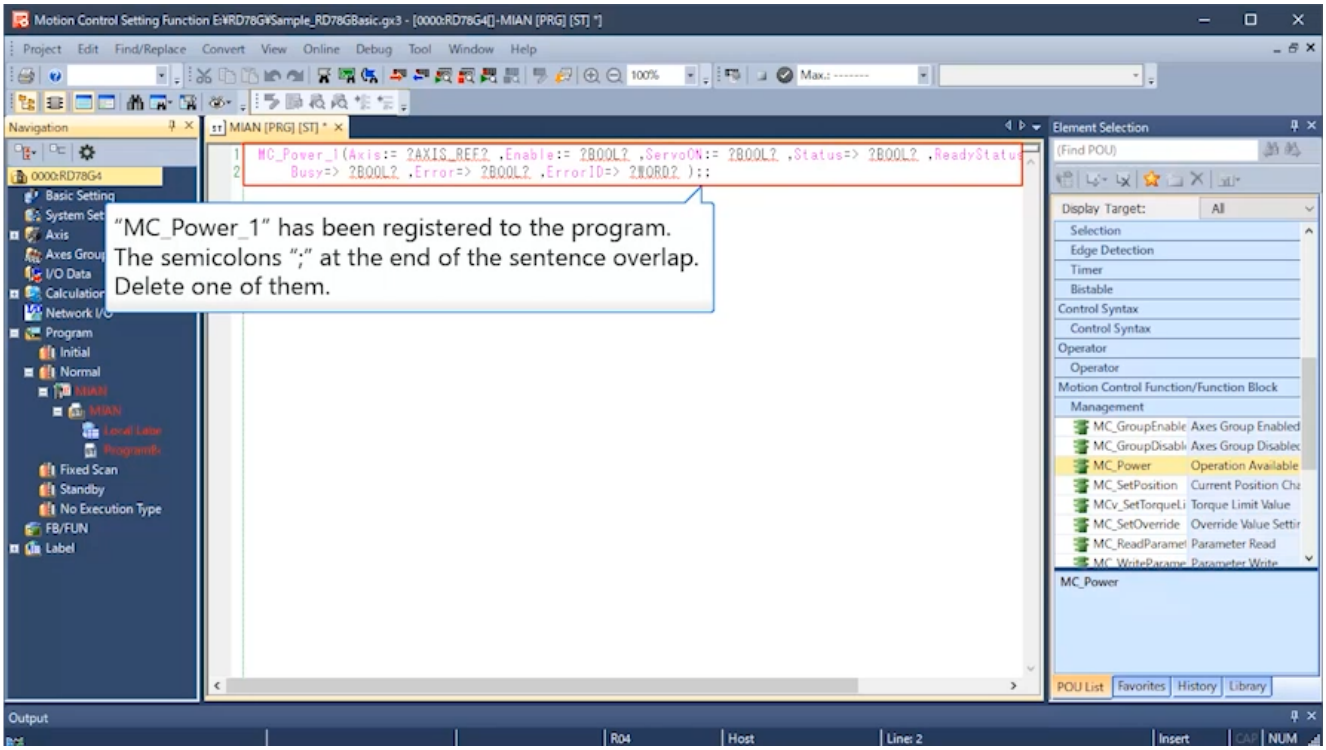
4.2.1 How to use Motion Control FB



4.2 Motion Module Program

Double-click the module extended parameter to display the [Motion Control Setting Function] screen.
Double-click [Program] → [Normal] → [MAIN] → [ProgramBody] in the project tree on the Motion Control Setting Function screen to open the program.

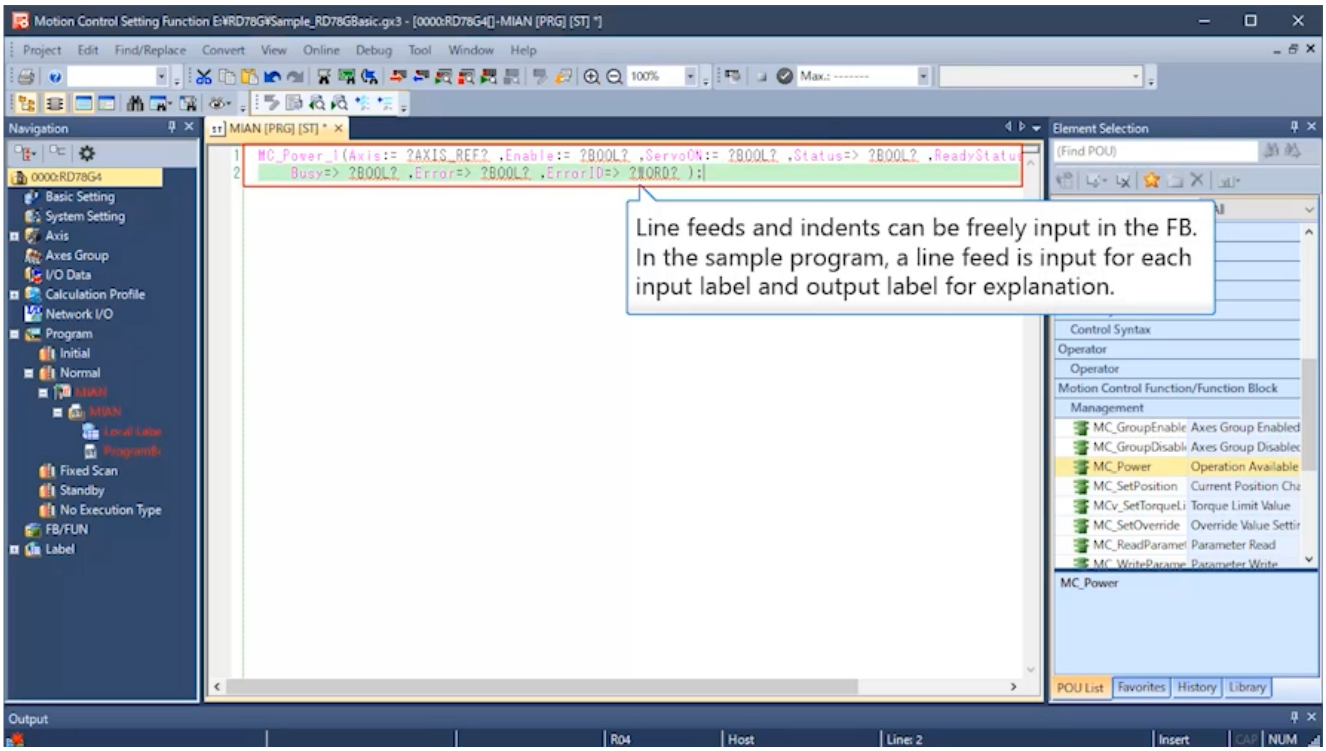
4.2.1 How to use Motion Control FB



4.2 Motion Module Program

Double-click the module extended parameter to display the [Motion Control Setting Function] screen.
Double-click [Program] → [Normal] → [MAIN] → [ProgramBody] in the project tree on the Motion Control Setting Function screen to open the program.

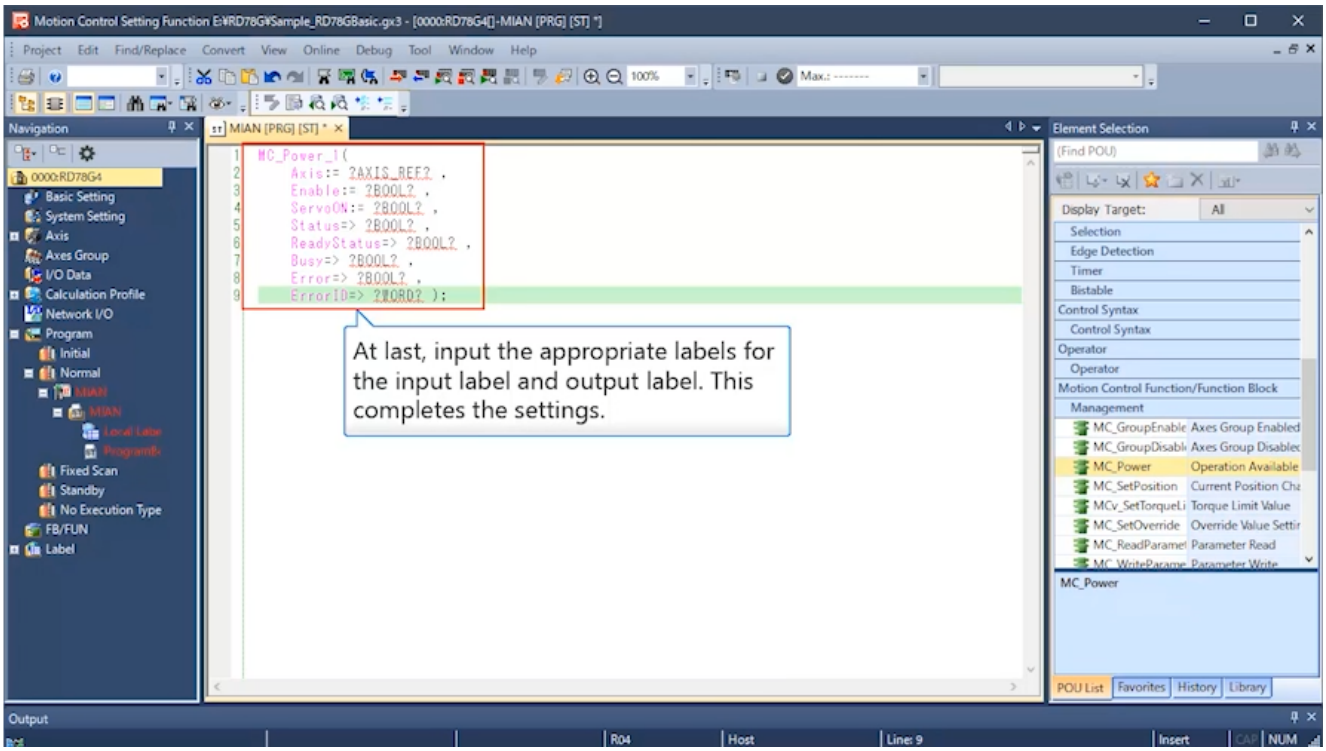
4.2.1 How to use Motion Control FB



4.2 Motion Module Program

Double-click the module extended parameter to display the [Motion Control Setting Function] screen.
Double-click [Program] → [Normal] → [MAIN] → [ProgramBody] in the project tree on the Motion Control Setting Function screen to open the program.

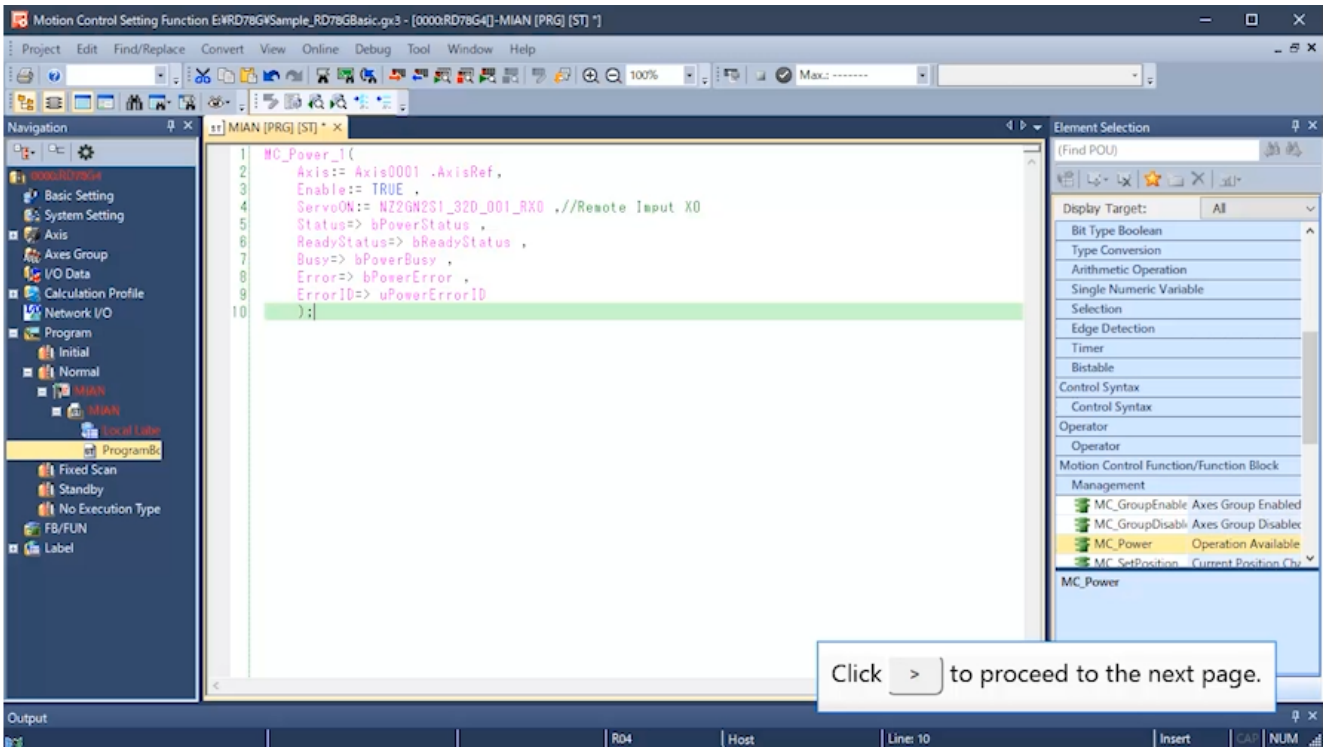
4.2.1 How to use Motion Control FB



4.2 Motion Module Program

Double-click the module extended parameter to display the [Motion Control Setting Function] screen.
Double-click [Program] → [Normal] → [MAIN] → [ProgramBody] in the project tree on the Motion Control Setting Function screen to open the program.

4.2.1 How to use Motion Control FB



4.2.2 Initial value setting, servo-on

Program name in the sample program: ServoON_JOG

Set the initial value and servo-on of the global label in this program.
Use MC_Power in Motion Control FB for servo-on.
Connecting X0 of the remote input module to the input of ServoON in the FB completes the program in which the servo turns on by turning on X0.

```
1  //-----Initial Value Setting-----------------------------
2  G_leJogVelocity := 20000.0;//20000um/s  = 1200mm/min
3  G_leJogAcc      := 20000.0;//20000um/s2 = 1200mm/min/s
4  G_leJogDec      := 20000.0;//20000um/s2 = 1200mm/min/s
5  G_leJogJerk     := 25000.0;//25000um/s3
6
7  G_lePoint0Address := 0.0;     //0.0mm
8  G_lePoint1Address := 150000.0;//150.0000mm
9
10 //-----Axis0001 Servo ON-----------------------------
11 MC_Power_1(
12     Axis     := Axis0001.AxisRef ,
13     Enable   := TRUE ,
14     ServoON  := NZ2GN2S1_32D_001_RX0 ,//Remote Input X0
15     Status       => bPowerStatus,
16     ReadyStatus  => bReadyStatus ,
17     Busy         => bPowerBusy ,
18     Error        => bPowerError ,
19     ErrorID      => uPowerErrorID
20 );
```

Set the velocity during JOG operation, acceleration/deceleration, jerk value, home position address during positioning operation, and positioning address assigned to the global label. Describing the meanings of the input values with comments beforehand enables the programs to be easily reviewed.

MCFB

Specify X0 of the remote input module for the servo-on request.

<MC_Power specification (extract)>

I/O variable name		Variable name	Data type	Description
Input	Enable	Enable	BOOL	While the Enable input is TRUE, the axis control is enabled.
	Servo-on request	ServoON	BOOL	Specifies the signal for the servo-on request.
Output	Ready	Status	BOOL	Indicates the operation ready status.
	Ready-on status	ReadyStatus	BOOL	Indicates the ready-on/off status.
	Executing	Busy	BOOL	Turns TRUE while the FB is being executed.
	Error	Error	BOOL	Turns TRUE when an error occurs in the FB.
	Error code	ErrorID	WORD (UINT)	Returns the error code occurred in the FB.

MELSEC iQ-R Motion Module User's Manual (Application)
2.4 Servo ON/OFF

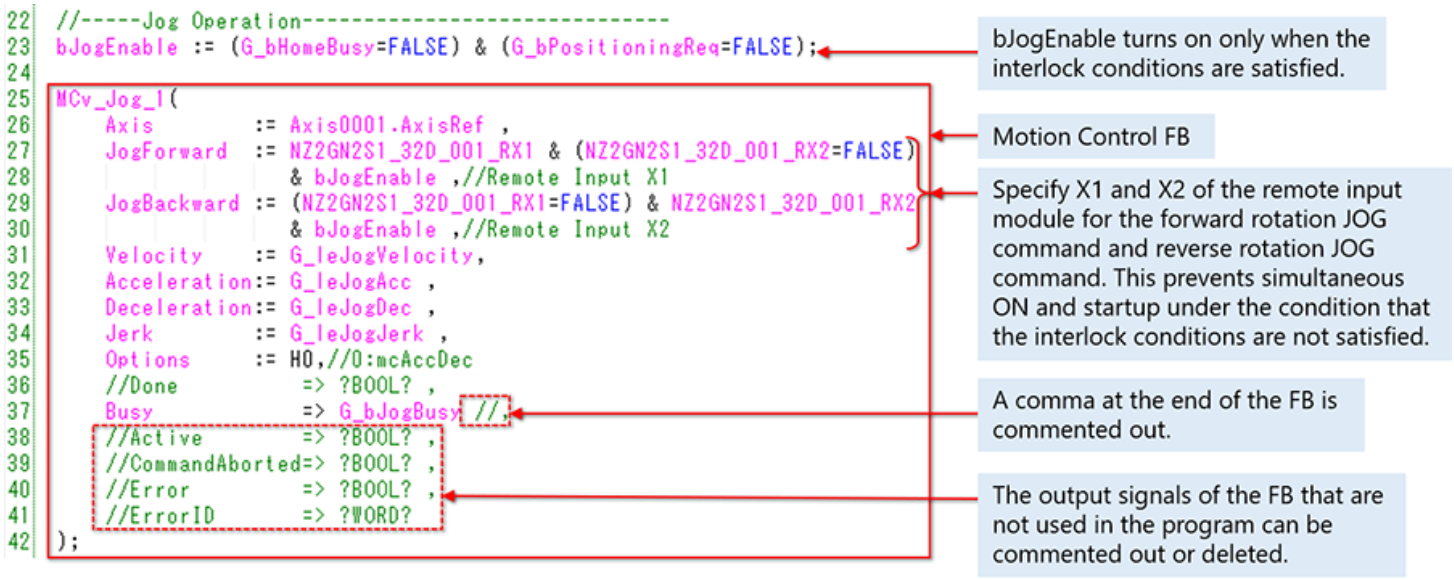
MELSEC iQ-R Programming Manual (Motion Control Function Blocks) 3.1 Management FBs
MC_Power

Tips
If each manual of the motion module has been downloaded in e-manual Viewer, pressing the F1 button with the text cursor pointed on the FB name can call the the pages in the manual where the FB specifications are described.

4.2.3 JOG operation

Program name in the sample program: ServoON_JOG

Use MCv_Jog in Motion Control FB.
To prevent MCv_Jog from being executed during the home position return and positioning operation, a bit called bJogEnable is provided to interlock.



<MCv_Jog specification (extract)>

I/O variable name		Variable name	Data type	Description
Input	Forward rotation JOG command	JogForward	BOOL	When TRUE is set, the forward rotation JOG is executed.
	Reverse rotation JOG command	JogBackWard	BOOL	When TRUE is set, the reverse rotation JOG is executed.
	Target velocity	Velocity	LREAL	Sets the command velocity.
	Acceleration	Acceleration	LREAL	Sets the acceleration.
	Deceleration	Deceleration	LREAL	Sets the deceleration.
	Jerk	Jerk	LREAL	Sets the jerk.
	Option	Options	DWORD(HEX) (Note)	Sets the function option with bit specification.(→Refer to the next page.)
Output	Execution completion	Done	BOOL	Turns TRUE for only one scan when the JOG command is turned off and the operation is decelerated to stop.
	Executing	Busy	BOOL	Turns TRUE while the FB is being executed.
	Controlling	Active	BOOL	Turns TRUE when the FB is controlling the axis.
	Abortion of execution	CommandAborted	BOOL	Turns TRUE when the execution is aborted.
	Error	Error	BOOL	Turns TRUE when an error occurs in the FB.
	Error code	ErrorID	WORD (UINT)	Returns the error code occurred in the FB.

(Note) A hexadecimal is written in the format of "H□" or "16#□".

MELSEC iQ-R Motion Module User's Manual (Application)

MELSEC iQ-R Programming Manual (Motion Control Function Blocks) 3.2 Operation FBs
MCv_Jog

4.2.3 JOG operation

The following shows the setting values and descriptions of the options for MCv_Jog.

Setting value	Acceleration/deceleration method setting
0h	mcAccDec · · · Acceleration/deceleration specification method (Jerk acceleration/deceleration method)
1h	mcFixedTime · · · Acceleration/deceleration time constant method (acceleration/deceleration time constant specification method)

When 0h: mcAccDec is specified, the acceleration/deceleration method is set to the jerk acceleration/deceleration method.

At this time, set Acceleration and Deceleration in a unit of [U/s^2], and Jerk in a unit of [U/s^3].
For the details of the jerk acceleration/deceleration (U: Axis command unit), refer to the next page.

When 1h: mcFixedTime is specified, the acceleration/deceleration method is set to the acceleration/deceleration time constant specification method.
At this time, set Acceleration in a unit of [s].
Deceleration and Jerk are not used.

MELSEC iQ-R Motion Module User's Manual (Application)

6.3 Single Axis Manual Control

MELSEC iQ-R Programming Manual (Motion Control Function Blocks)
3.2 Operation FBs
MCv_Jog

4.2.3 JOG operation

The following figures show the velocity waveforms during the jerk acceleration/deceleration.
The sum of the time to reach the target acceleration and the time to reach 0 from the target acceleration at the end of the acceleration is called the jerk application time. The ratio of the jerk application time in the acceleration (deceleration) time is called the jerk application ratio.

The following figures show the velocity waveforms and acceleration waveforms at a time of acceleration when the command velocity and command acceleration are constant and the jerk is changed.
The larger the jerk value becomes, the smaller the jerk application ratio becomes, and the velocity pattern changes to the trapezoidal acceleration/deceleration.
Additionally, the acceleration time and deceleration time will be shorter.

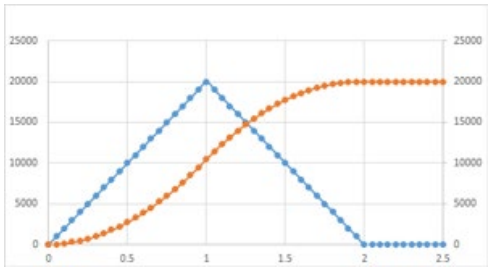
Command velocity: 20000 [µm/s] = 1200 [mm/min]

Command acceleration: 20000 [µm/s^2] = 1200 [mm/min/s]

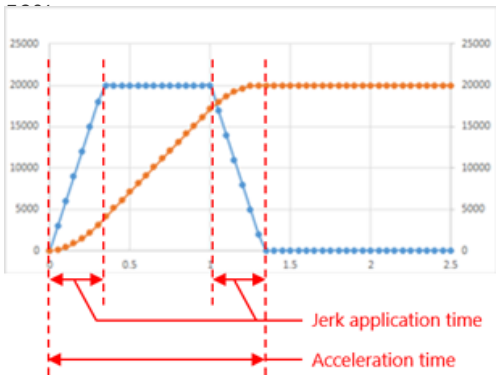
Blue line: Acceleration [µm/s^2] Vertical axis on the left

Orange line: Velocity [µm/s] Vertical axis on the right

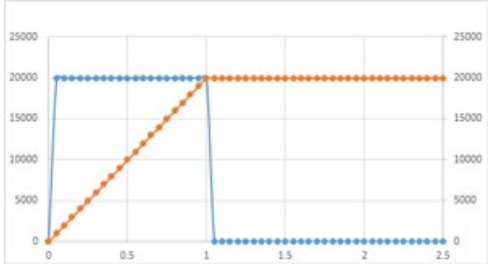
J = 20000 [µm/s^3]
Jerk application ratio: 100%



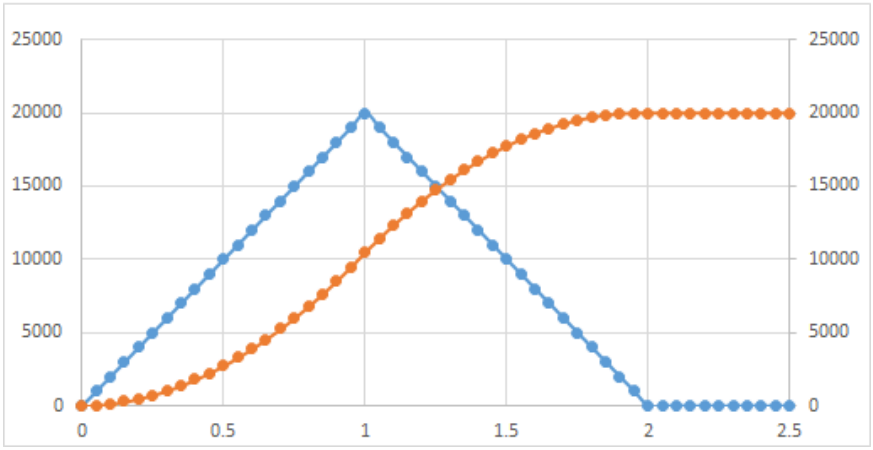
J = 60000 [µm/s^3]
Jerk application ratio:



When J = 0 [µm/s^3] is set,
the velocity pattern changes to the trapezoidal acceleration/deceleration.



J = 20000 [µm/s³] Jerk application ratio: 100%



4.2.3 JOG operation

The following figures show the velocity waveforms during the jerk acceleration/deceleration.
The sum of the time to reach the target acceleration and the time to reach 0 from the target acceleration at the end of the acceleration is called the jerk application time. The ratio of the jerk application time in the acceleration (deceleration) time is called the jerk application ratio.

The following figures show the velocity waveforms and acceleration waveforms at a time of acceleration when the command velocity and command acceleration are constant and the jerk is changed.
The larger the jerk value becomes, the smaller the jerk application ratio becomes, and the velocity pattern changes to the trapezoidal acceleration/deceleration.
Additionally, the acceleration time and deceleration time will be shorter.

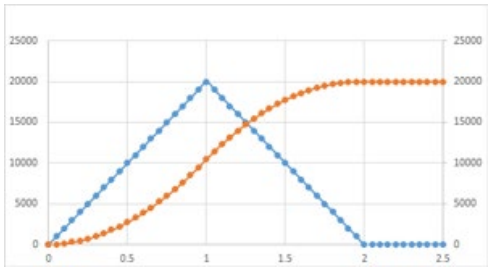
Command velocity: 20000 [µm/s] = 1200 [mm/min]

Command acceleration: 20000 [µm/s^2] = 1200 [mm/min/s]

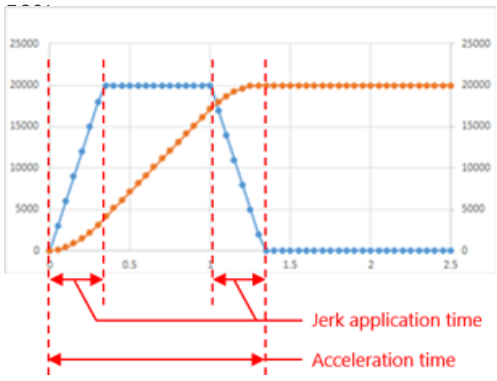
Blue line: Acceleration [µm/s^2] Vertical axis on the left

Orange line: Velocity [µm/s] Vertical axis on the right

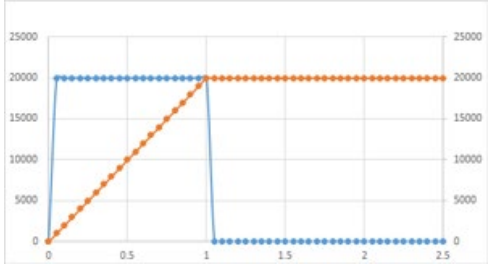
J = 20000 [µm/s^3]
Jerk application ratio: 100%



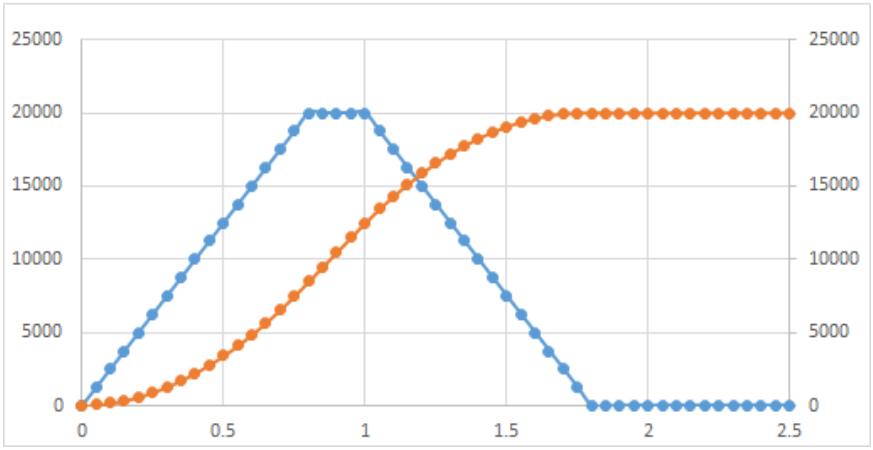
J = 60000 [µm/s^3]
Jerk application ratio:



When J = 0 [µm/s^3] is set,
the velocity pattern changes to the trapezoidal acceleration/deceleration.



J = 25000 [µm/s^3] Jerk application ratio: 88.8%



4.2.3 JOG operation

The following figures show the velocity waveforms during the jerk acceleration/deceleration.
The sum of the time to reach the target acceleration and the time to reach 0 from the target acceleration at the end of the acceleration is called the jerk application time. The ratio of the jerk application time in the acceleration (deceleration) time is called the jerk application ratio.

The following figures show the velocity waveforms and acceleration waveforms at a time of acceleration when the command velocity and command acceleration are constant and the jerk is changed.
The larger the jerk value becomes, the smaller the jerk application ratio becomes, and the velocity pattern changes to the trapezoidal acceleration/deceleration.
Additionally, the acceleration time and deceleration time will be shorter.

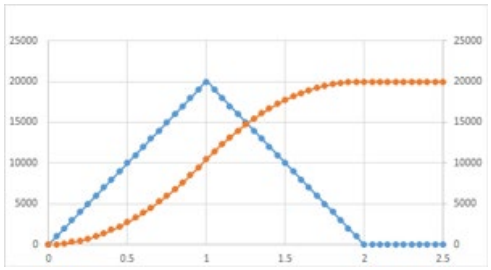
Command velocity: 20000 [µm/s] = 1200 [mm/min]

Command acceleration: 20000 [µm/s^2] = 1200 [mm/min/s]

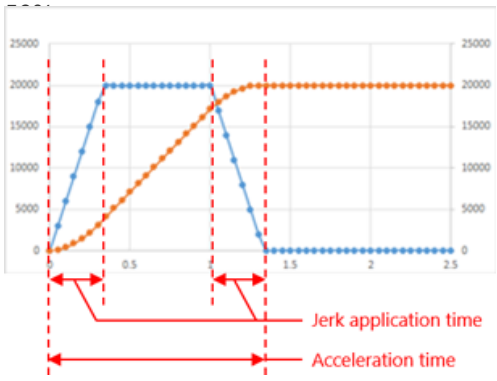
Blue line: Acceleration [µm/s^2] Vertical axis on the left

Orange line: Velocity [µm/s] Vertical axis on the right

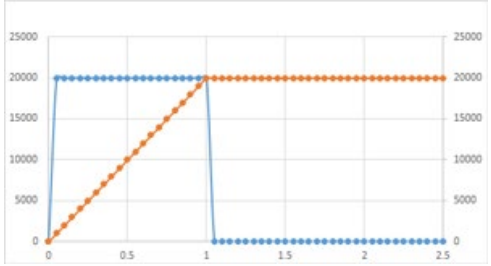
J = 20000 [µm/s^3]
Jerk application ratio: 100%



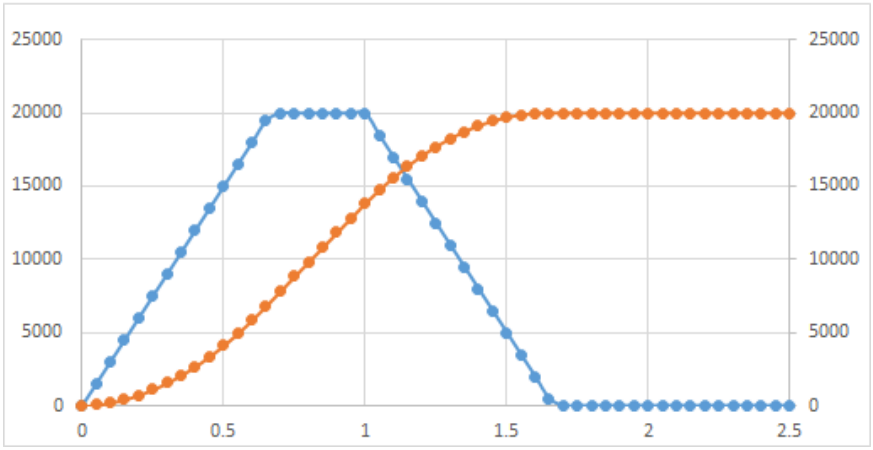
J = 60000 [µm/s^3]
Jerk application ratio:



When J = 0 [µm/s^3] is set,
the velocity pattern changes to the trapezoidal acceleration/deceleration.



J = 30000 [µm/s^3] Jerk application ratio: 80%



4.2.3 JOG operation

The following figures show the velocity waveforms during the jerk acceleration/deceleration.
The sum of the time to reach the target acceleration and the time to reach 0 from the target acceleration at the end of the acceleration is called the jerk application time. The ratio of the jerk application time in the acceleration (deceleration) time is called the jerk application ratio.

The following figures show the velocity waveforms and acceleration waveforms at a time of acceleration when the command velocity and command acceleration are constant and the jerk is changed.
The larger the jerk value becomes, the smaller the jerk application ratio becomes, and the velocity pattern changes to the trapezoidal acceleration/deceleration.
Additionally, the acceleration time and deceleration time will be shorter.

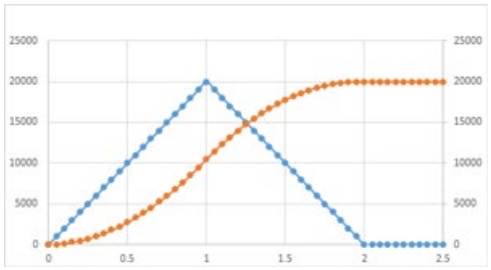
Command velocity: 20000 [µm/s] = 1200 [mm/min]

Command acceleration: 20000 [µm/s^2] = 1200 [mm/min/s]

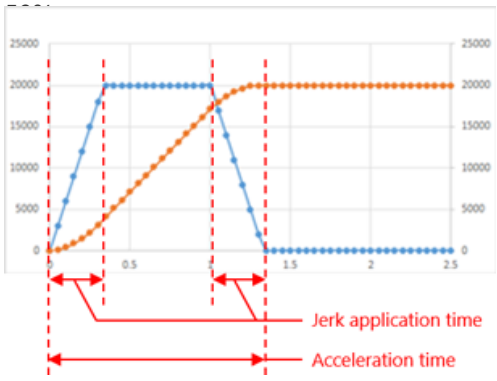
Blue line: Acceleration [µm/s^2] Vertical axis on the left

Orange line: Velocity [µm/s] Vertical axis on the right

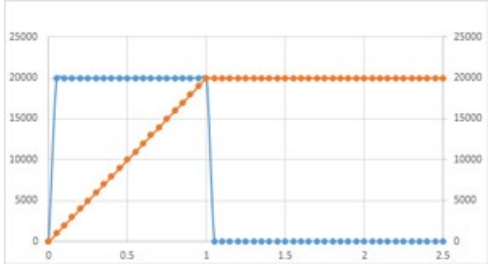
J = 20000 [µm/s^3]
Jerk application ratio: 100%



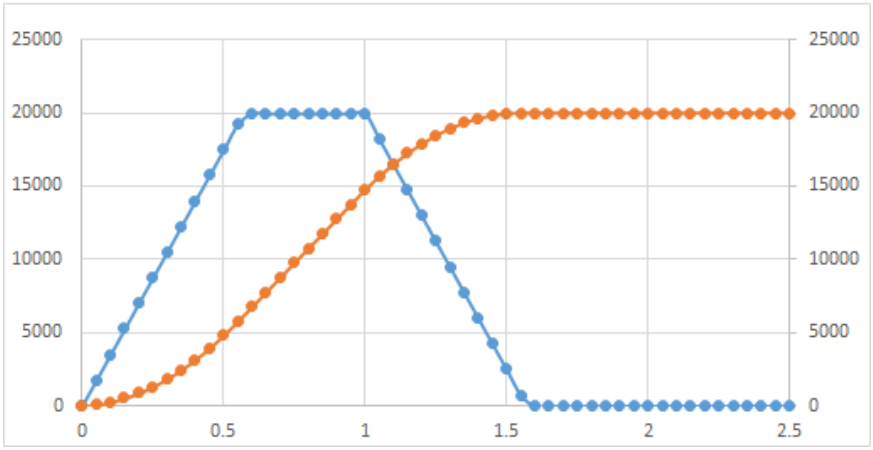
J = 60000 [µm/s^3]
Jerk application ratio:



When J = 0 [µm/s^3] is set,
the velocity pattern changes to the trapezoidal acceleration/deceleration.



J = 35000 [µm/s³] Jerk application ratio: 72.7%



4.2.3 JOG operation

The following figures show the velocity waveforms during the jerk acceleration/deceleration.
The sum of the time to reach the target acceleration and the time to reach 0 from the target acceleration at the end of the acceleration is called the jerk application time. The ratio of the jerk application time in the acceleration (deceleration) time is called the jerk application ratio.

The following figures show the velocity waveforms and acceleration waveforms at a time of acceleration when the command velocity and command acceleration are constant and the jerk is changed.
The larger the jerk value becomes, the smaller the jerk application ratio becomes, and the velocity pattern changes to the trapezoidal acceleration/deceleration.
Additionally, the acceleration time and deceleration time will be shorter.

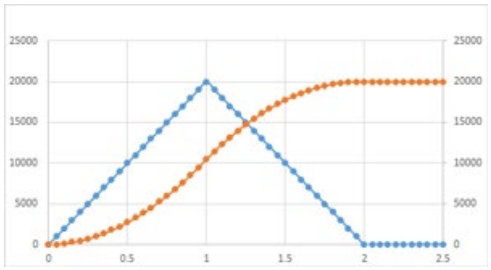
Command velocity: 20000 [µm/s] = 1200 [mm/min]

Command acceleration: 20000 [µm/s^2] = 1200 [mm/min/s]

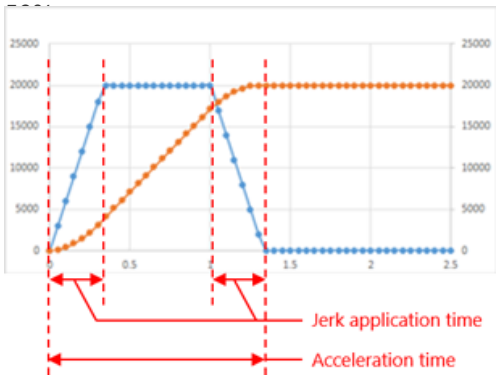
Blue line: Acceleration [µm/s^2] Vertical axis on the left

Orange line: Velocity [µm/s] Vertical axis on the right

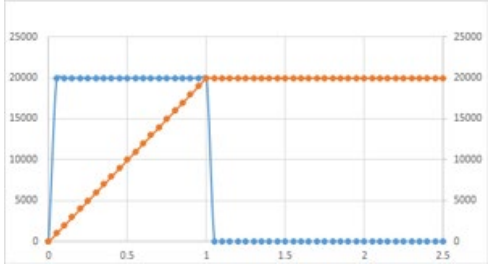
J = 20000 [µm/s^3]
Jerk application ratio: 100%



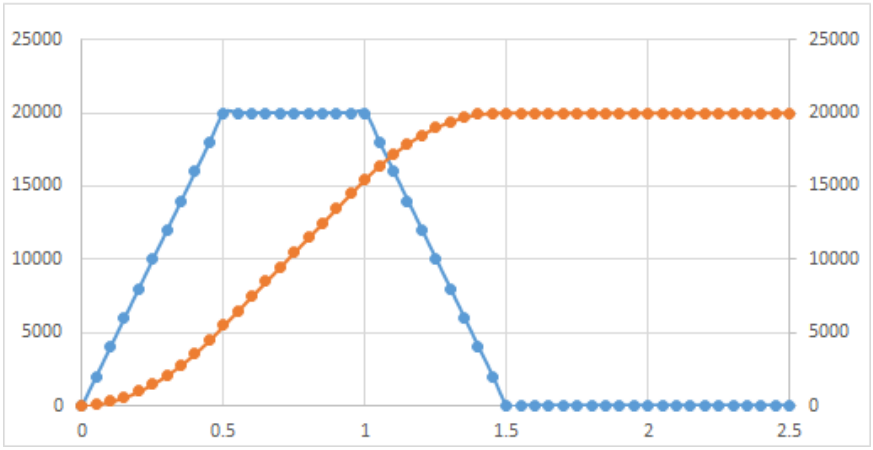
J = 60000 [µm/s^3]
Jerk application ratio:



When J = 0 [µm/s^3] is set,
the velocity pattern changes to the trapezoidal acceleration/deceleration.



J = 40000 [µm/s³] Jerk application ratio: 66.6%



4.2.3 JOG operation

The following figures show the velocity waveforms during the jerk acceleration/deceleration.
The sum of the time to reach the target acceleration and the time to reach 0 from the target acceleration at the end of the acceleration is called the jerk application time. The ratio of the jerk application time in the acceleration (deceleration) time is called the jerk application ratio.

The following figures show the velocity waveforms and acceleration waveforms at a time of acceleration when the command velocity and command acceleration are constant and the jerk is changed.
The larger the jerk value becomes, the smaller the jerk application ratio becomes, and the velocity pattern changes to the trapezoidal acceleration/deceleration.
Additionally, the acceleration time and deceleration time will be shorter.

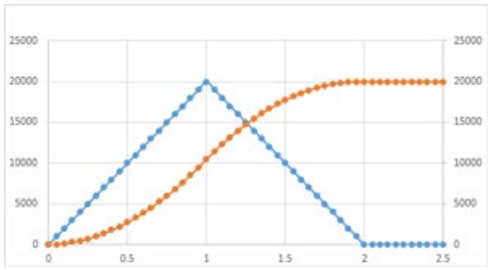
Command velocity: 20000 [µm/s] = 1200 [mm/min]

Command acceleration: 20000 [µm/s^2] = 1200 [mm/min/s]

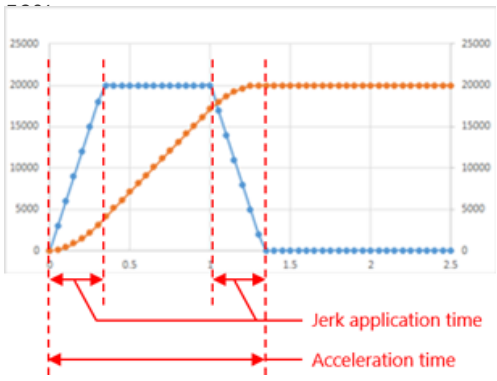
Blue line: Acceleration [µm/s^2] Vertical axis on the left

Orange line: Velocity [µm/s] Vertical axis on the right

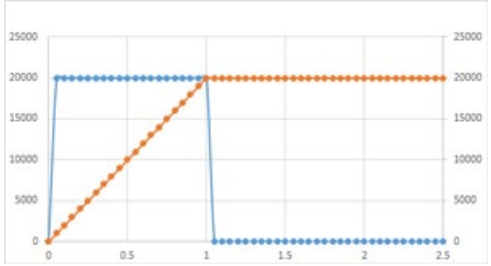
J = 20000 [µm/s^3]
Jerk application ratio: 100%



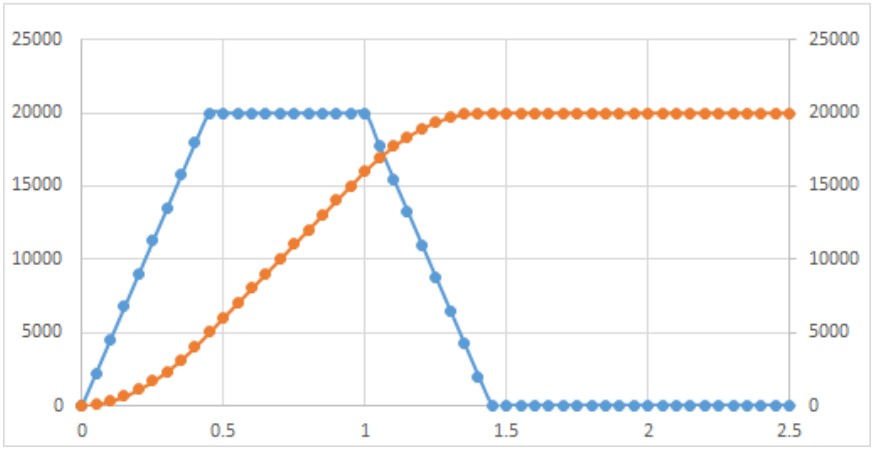
J = 60000 [µm/s^3]
Jerk application ratio:



When J = 0 [µm/s^3] is set,
the velocity pattern changes to the trapezoidal acceleration/deceleration.



J = 45000 [µm/s³] Jerk application ratio: 61.5%



4.2.3 JOG operation

The following figures show the velocity waveforms during the jerk acceleration/deceleration.
The sum of the time to reach the target acceleration and the time to reach 0 from the target acceleration at the end of the acceleration is called the jerk application time. The ratio of the jerk application time in the acceleration (deceleration) time is called the jerk application ratio.

The following figures show the velocity waveforms and acceleration waveforms at a time of acceleration when the command velocity and command acceleration are constant and the jerk is changed.
The larger the jerk value becomes, the smaller the jerk application ratio becomes, and the velocity pattern changes to the trapezoidal acceleration/deceleration.
Additionally, the acceleration time and deceleration time will be shorter.

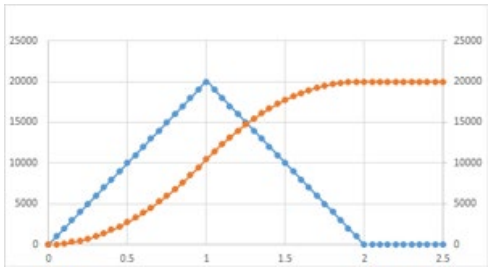
Command velocity: 20000 [μm/s] = 1200 [mm/min]

Command acceleration: 20000 [μm/s^2] = 1200 [mm/min/s]

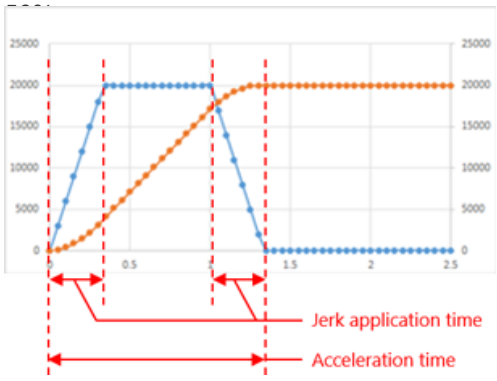
Blue line: Acceleration [μm/s^2] Vertical axis on the left

Orange line: Velocity [μm/s] Vertical axis on the right

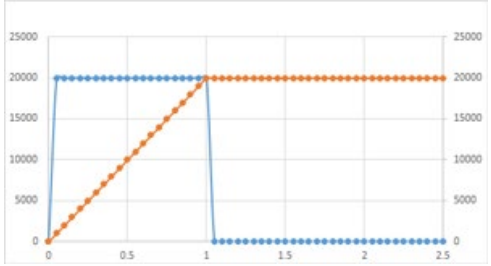
J = 20000 [μm/s^3]
Jerk application ratio: 100%



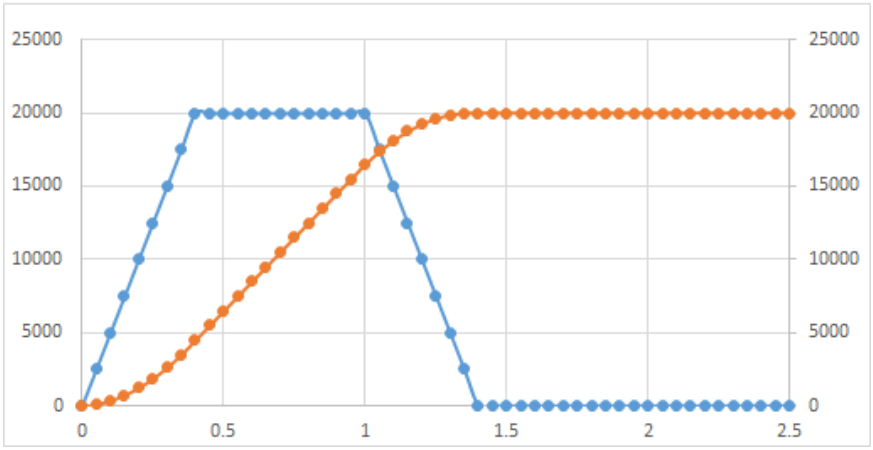
J = 60000 [μm/s^3]
Jerk application ratio:



When J = 0 [μm/s^3] is set,
the velocity pattern changes to the trapezoidal acceleration/deceleration.



J = 50000 [μm/s³] Jerk application ratio: 57.1%



4.2.3 JOG operation

The following figures show the velocity waveforms during the jerk acceleration/deceleration.
The sum of the time to reach the target acceleration and the time to reach 0 from the target acceleration at the end of the acceleration is called the jerk application time. The ratio of the jerk application time in the acceleration (deceleration) time is called the jerk application ratio.

The following figures show the velocity waveforms and acceleration waveforms at a time of acceleration when the command velocity and command acceleration are constant and the jerk is changed.
The larger the jerk value becomes, the smaller the jerk application ratio becomes, and the velocity pattern changes to the trapezoidal acceleration/deceleration.
Additionally, the acceleration time and deceleration time will be shorter.

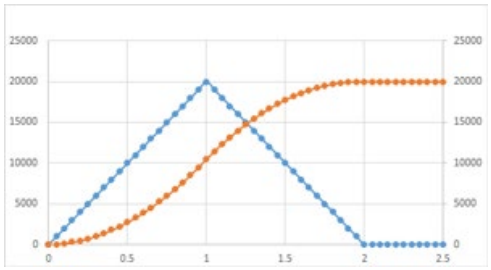
Command velocity: 20000 [µm/s] = 1200 [mm/min]

Command acceleration: 20000 [µm/s^2] = 1200 [mm/min/s]

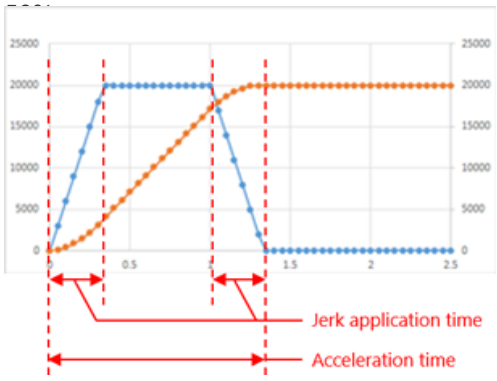
Blue line: Acceleration [µm/s^2] Vertical axis on the left

Orange line: Velocity [µm/s] Vertical axis on the right

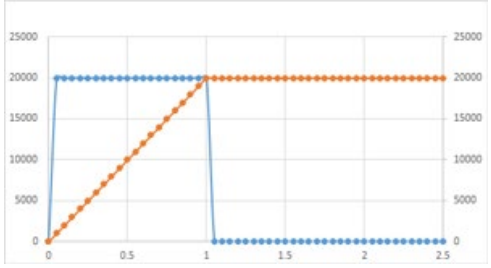
J = 20000 [µm/s^3]
Jerk application ratio: 100%



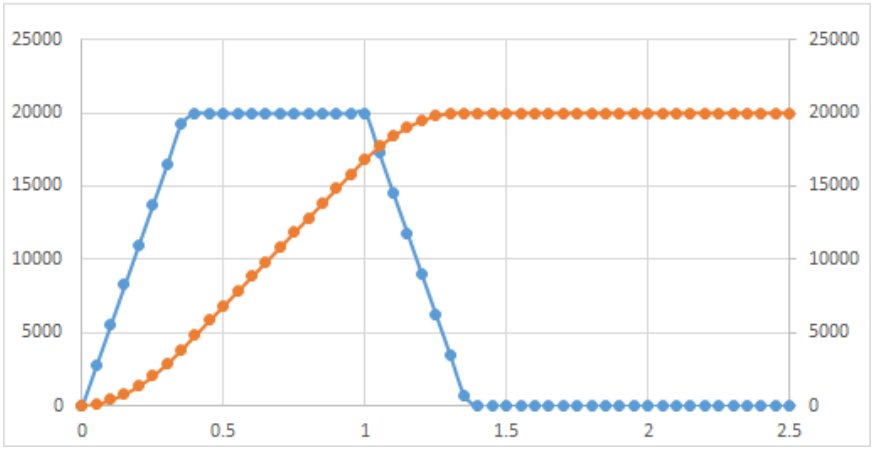
J = 60000 [µm/s^3]
Jerk application ratio:



When J = 0 [µm/s^3] is set,
the velocity pattern changes to the trapezoidal acceleration/deceleration.



J = 55000 [µm/s^3] Jerk application ratio: 53.3%



4.2.3 JOG operation

The following figures show the velocity waveforms during the jerk acceleration/deceleration.
The sum of the time to reach the target acceleration and the time to reach 0 from the target acceleration at the end of the acceleration is called the jerk application time. The ratio of the jerk application time in the acceleration (deceleration) time is called the jerk application ratio.

The following figures show the velocity waveforms and acceleration waveforms at a time of acceleration when the command velocity and command acceleration are constant and the jerk is changed.
The larger the jerk value becomes, the smaller the jerk application ratio becomes, and the velocity pattern changes to the trapezoidal acceleration/deceleration.
Additionally, the acceleration time and deceleration time will be shorter.

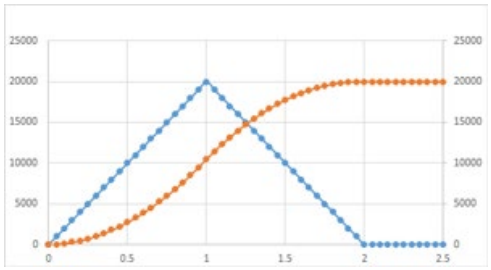
Command velocity: 20000 [µm/s] = 1200 [mm/min]

Command acceleration: 20000 [µm/s^2] = 1200 [mm/min/s]

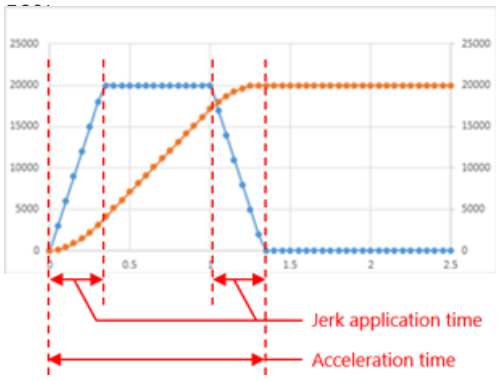
Blue line: Acceleration [µm/s^2] Vertical axis on the left

Orange line: Velocity [µm/s] Vertical axis on the right

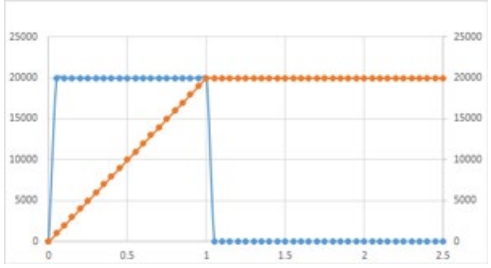
J = 20000 [µm/s^3]
Jerk application ratio: 100%



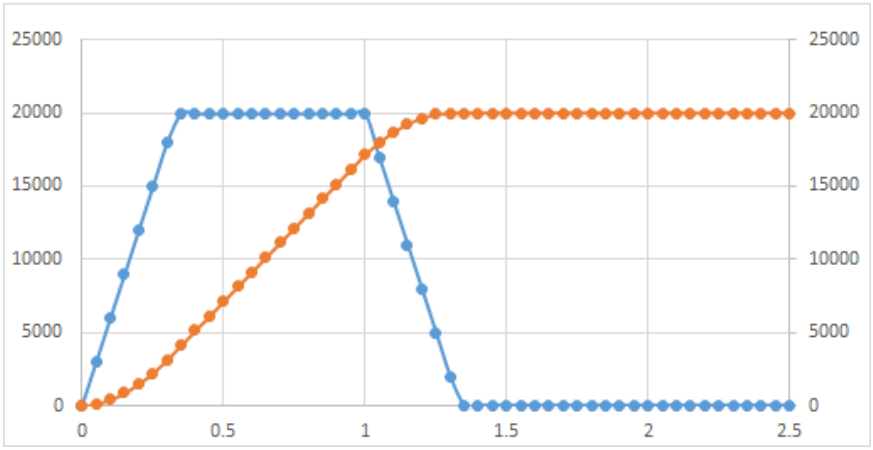
J = 60000 [µm/s^3]
Jerk application ratio:



When J = 0 [µm/s^3] is set,
the velocity pattern changes to the trapezoidal acceleration/deceleration.



J = 60000 [µm/s³] Jerk application ratio: 50%



4.2.3 JOG operation

The following figures show the velocity waveforms during the jerk acceleration/deceleration.
The sum of the time to reach the target acceleration and the time to reach 0 from the target acceleration at the end of the acceleration is called the jerk application time. The ratio of the jerk application time in the acceleration (deceleration) time is called the jerk application ratio.

The following figures show the velocity waveforms and acceleration waveforms at a time of acceleration when the command velocity and command acceleration are constant and the jerk is changed.
The larger the jerk value becomes, the smaller the jerk application ratio becomes, and the velocity pattern changes to the trapezoidal acceleration/deceleration.
Additionally, the acceleration time and deceleration time will be shorter.

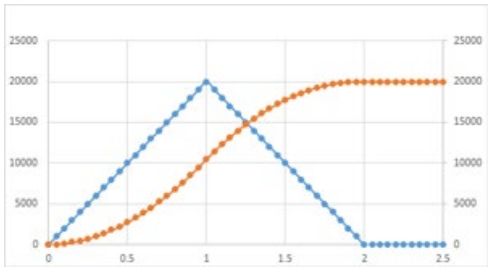
Command velocity: 20000 [µm/s] = 1200 [mm/min]

Command acceleration: 20000 [µm/s^2] = 1200 [mm/min/s]

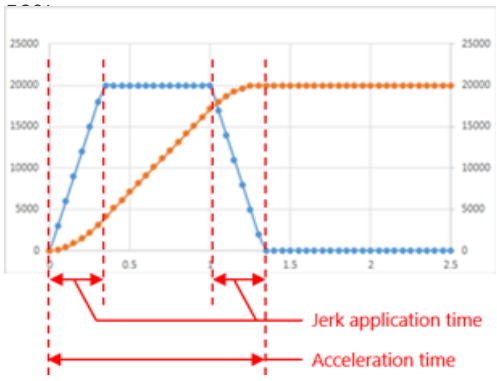
Blue line: Acceleration [µm/s^2] Vertical axis on the left

Orange line: Velocity [µm/s] Vertical axis on the right

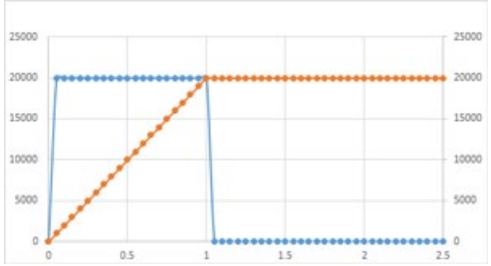
J = 20000 [µm/s^3]
Jerk application ratio: 100%



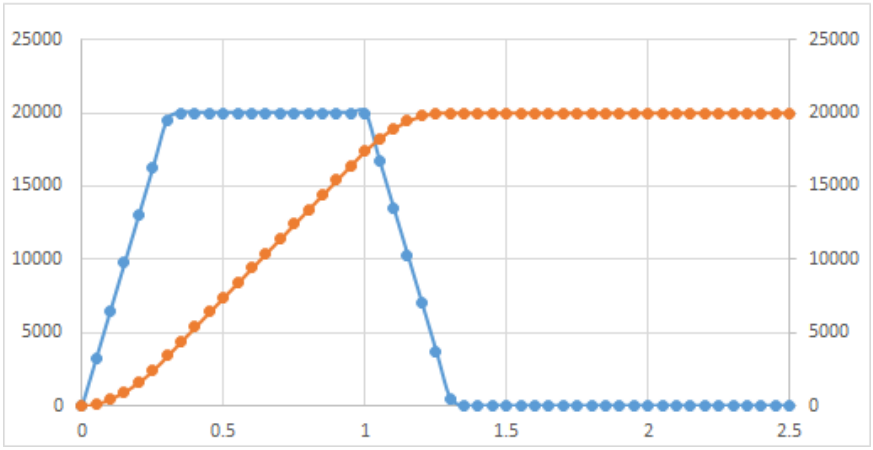
J = 60000 [µm/s^3]
Jerk application ratio:



When J = 0 [µm/s^3] is set,
the velocity pattern changes to the trapezoidal acceleration/deceleration.



J = 65000 [µm/s³] Jerk application ratio: 47%



4.2.3 JOG operation

The following figures show the velocity waveforms during the jerk acceleration/deceleration.
The sum of the time to reach the target acceleration and the time to reach 0 from the target acceleration at the end of the acceleration is called the jerk application time. The ratio of the jerk application time in the acceleration (deceleration) time is called the jerk application ratio.

The following figures show the velocity waveforms and acceleration waveforms at a time of acceleration when the command velocity and command acceleration are constant and the jerk is changed.
The larger the jerk value becomes, the smaller the jerk application ratio becomes, and the velocity pattern changes to the trapezoidal acceleration/deceleration.
Additionally, the acceleration time and deceleration time will be shorter.

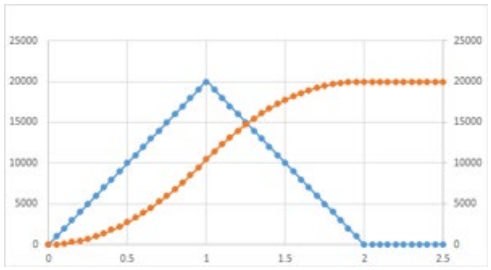
Command velocity: 20000 [µm/s] = 1200 [mm/min]

Command acceleration: 20000 [µm/s^2] = 1200 [mm/min/s]

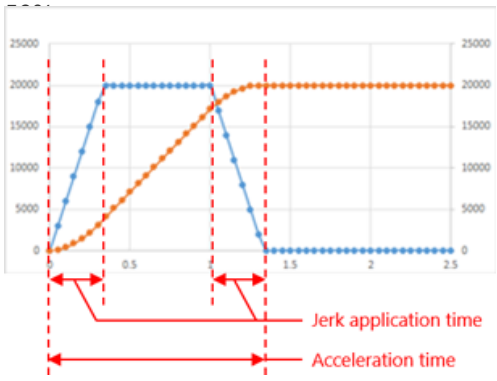
Blue line: Acceleration [µm/s^2] Vertical axis on the left

Orange line: Velocity [µm/s] Vertical axis on the right

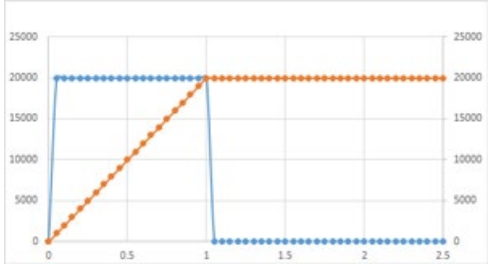
J = 20000 [µm/s^3]
Jerk application ratio: 100%



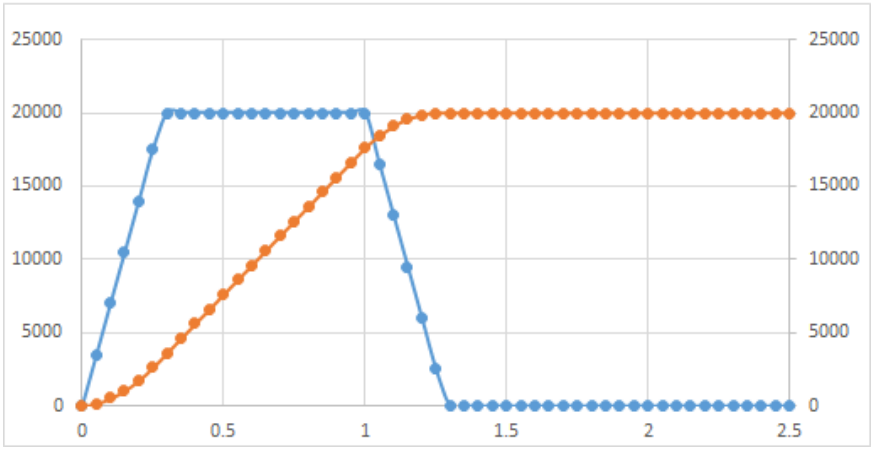
J = 60000 [µm/s^3]
Jerk application ratio:



When J = 0 [µm/s^3] is set,
the velocity pattern changes to the trapezoidal acceleration/deceleration.



J = 70000 [µm/s^3] Jerk application ratio: 44.4%



4.2.3 JOG operation

The following figures show the velocity waveforms during the jerk acceleration/deceleration.
The sum of the time to reach the target acceleration and the time to reach 0 from the target acceleration at the end of the acceleration is called the jerk application time. The ratio of the jerk application time in the acceleration (deceleration) time is called the jerk application ratio.

The following figures show the velocity waveforms and acceleration waveforms at a time of acceleration when the command velocity and command acceleration are constant and the jerk is changed.
The larger the jerk value becomes, the smaller the jerk application ratio becomes, and the velocity pattern changes to the trapezoidal acceleration/deceleration.
Additionally, the acceleration time and deceleration time will be shorter.

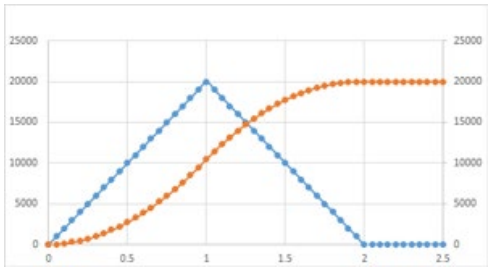
Command velocity: 20000 [µm/s] = 1200 [mm/min]

Command acceleration: 20000 [µm/s^2] = 1200 [mm/min/s]

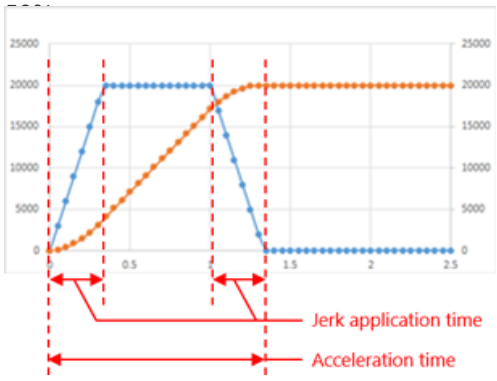
Blue line: Acceleration [µm/s^2] Vertical axis on the left

Orange line: Velocity [µm/s] Vertical axis on the right

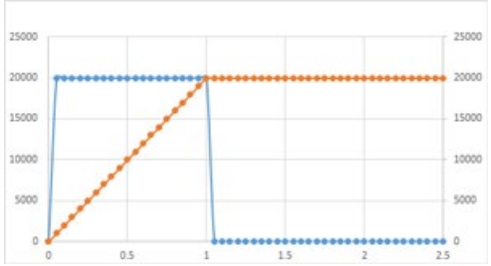
J = 20000 [µm/s^3]
Jerk application ratio: 100%



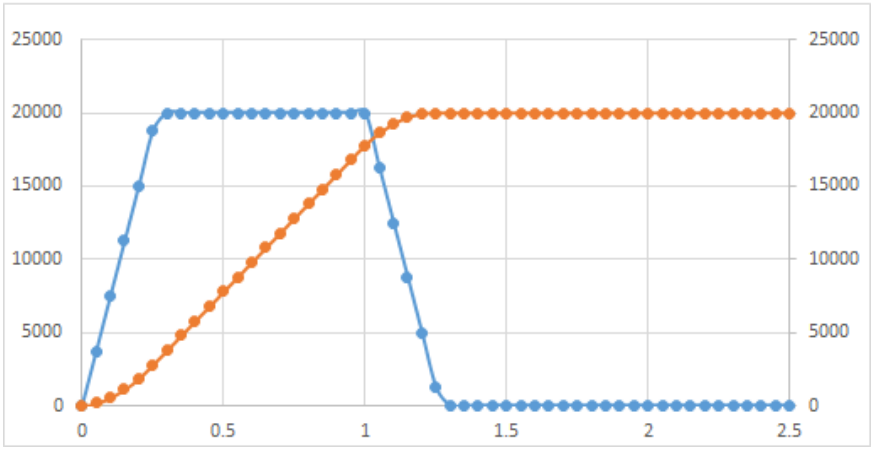
J = 60000 [µm/s^3]
Jerk application ratio:



When J = 0 [µm/s^3] is set,
the velocity pattern changes to the trapezoidal acceleration/deceleration.



J = 75000 [µm/s³] Jerk application ratio: 42.1%



4.2.3 JOG operation

The following figures show the velocity waveforms during the jerk acceleration/deceleration.
The sum of the time to reach the target acceleration and the time to reach 0 from the target acceleration at the end of the acceleration is called the jerk application time. The ratio of the jerk application time in the acceleration (deceleration) time is called the jerk application ratio.

The following figures show the velocity waveforms and acceleration waveforms at a time of acceleration when the command velocity and command acceleration are constant and the jerk is changed.
The larger the jerk value becomes, the smaller the jerk application ratio becomes, and the velocity pattern changes to the trapezoidal acceleration/deceleration.
Additionally, the acceleration time and deceleration time will be shorter.

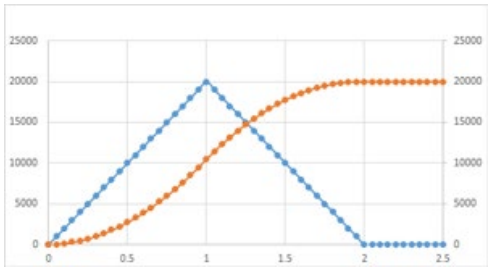
Command velocity: 20000 [μm/s] = 1200 [mm/min]

Command acceleration: 20000 [μm/s^2] = 1200 [mm/min/s]

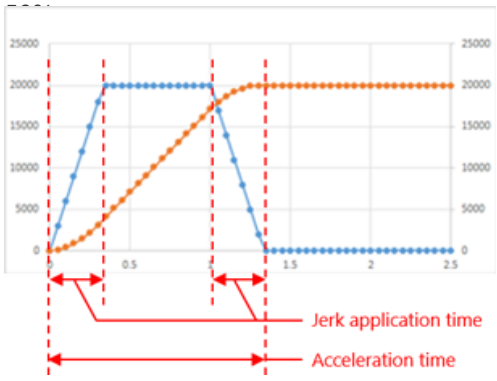
Blue line: Acceleration [μm/s^2] Vertical axis on the left

Orange line: Velocity [μm/s] Vertical axis on the right

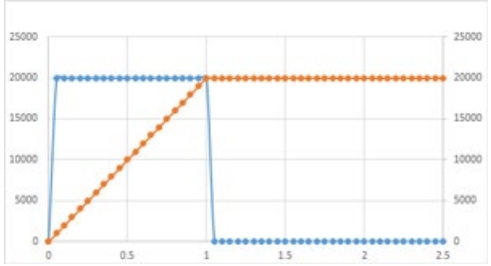
J = 20000 [μm/s^3]
Jerk application ratio: 100%



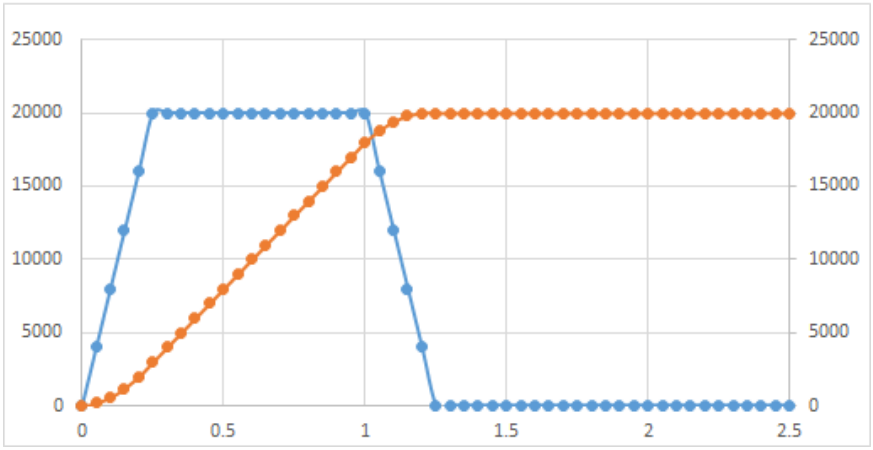
J = 60000 [μm/s^3]
Jerk application ratio:



When J = 0 [μm/s^3] is set,
the velocity pattern changes to the trapezoidal acceleration/deceleration.



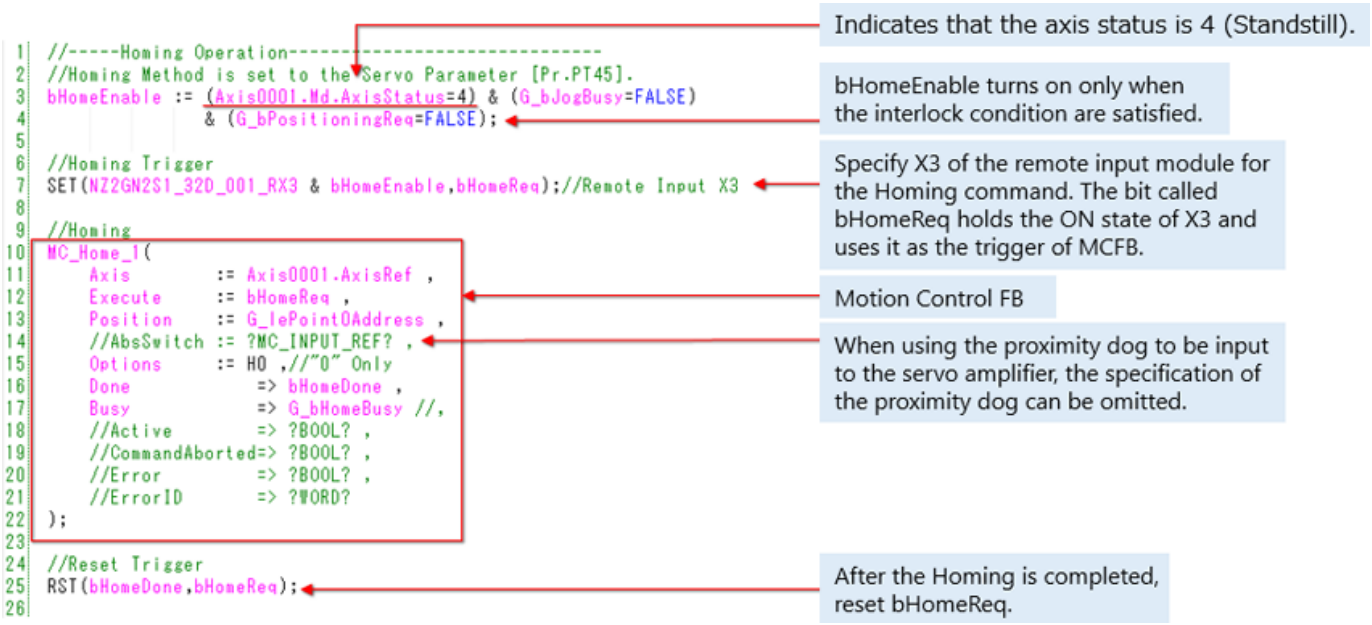
J = 80000 [μm/s^3] Jerk application ratio: 40%



4.2.4 Homing

Program name in the sample program: Homing

Use MC_Home in Motion Control FB.
To prevent MC_Home from being executed when the servo cannot be started, such as when the servo is off or when an error has occurred, or during JOG and positioning operation, a bit called bHomeEnable is provided to interlock.
Set the Homing method with the parameter [Pr.PT45] of the MR-J5-G servo amplifier.




```
//-----Homing Operation-------------------------
//Homing Method is set to the Servo Parameter [Pr.PT45].
bHomeEnable := (Axis0001.Md.AxisStatus=4) & (G_bJogBusy=FALSE)
             & (G_bPositioningReq=FALSE);

//Homing Trigger
SET(NZ2GN2S1_32D_001_RX3 & bHomeEnable,bHomeReq);//Remote Input X3

//Homing
MC_Home_1(
    Axis         := Axis0001.AxisRef ,
    Execute      := bHomeReq ,
    Position     := G_lePointOAddress ,
    //AbsSwitch  := ?MC_INPUT_REF? ,
    Options      := H0 ,//"0" Only
    Done             => bHomeDone ,
    Busy             => G_bHomeBusy //,
    //Active         => ?BOOL? ,
    //CommandAborted=> ?BOOL? ,
    //Error          => ?BOOL? ,
    //ErrorID        => ?WORD?
);

//Reset Trigger
RST(bHomeDone,bHomeReq);
```

<MC_Home specification (extract)>

I/O variable name		Variable name	Data type	Description
Input	Execution command	Execute	BOOL	Executes a home position return when TRUE is set.
	Target position	Position	LREAL	Specifies the home position address.
	Home position switch	AbsSwitch	MC_INPUT_REF	Specifies the proximity dog signal.
	Option	Options	DWORD(HEX)	Set "0".
Output	Execution completion	Done	BOOL	Turns TRUE after the home position return is completed.
	Executing	Busy	BOOL	Turns TRUE while the FB is being executed.
	Controlling	Active	BOOL	Turns TRUE when the FB is controlling the axis.
	Abortion of execution	CommandAborted	BOOL	Turns TRUE when the execution is aborted.
	Error	Error	BOOL	Turns TRUE when an error occurs in the FB.
	Error code	ErrorID	WORD (UINT)	Returns the error code occurred in the FB.

MELSEC iQ-R Motion Module User's Manual (Application)
5 HOMING

MELSEC iQ-R Programming Manual (Motion Control Function Blocks) 3.2 Operation FBs
MC_Home

4.2.5 Positioning control

Program name in the sample program: Positioning

Use MC_MoveRelative and MC_MoveAbsolute in Motion Control FB.
To prevent MC_Move from being executed when the servo cannot be started, such as when the servo is off or when an error has occurred, when the home position return is not completed, or during JOG operation and home position return operation, a bit called bMoveEnable is provided to interlock.

```
//-----Initial Value Setting----------------------------
lePosVelocity   := 20000.0 ;//20000um/s  = 1200mm/min
lePosAcc        := 20000.0 ;//20000um/s2 = 1200mm/min/s
lePosDec        := 20000.0 ;//20000um/s2 = 1200mm/min/s
lePosJerk       := 25000.0 ;//25000um/s3

//-----Positioning Operation----------------------------
bMoveEnable := (Axis0001.Md.AxisStatus=4) & (Axis0001.Md.Homing_Request=FALSE)
             & (G_bJogBusy=FALSE) & (G_bHomeBusy=FALSE);

//Start Trigger
SET(NZ2GN2S1_32D_001_RX4 & bMoveEnable,G_bPositioningReq); //Remote Input X4

//PTP1(Move Relative)
MC_MoveRelative_1(
    Axis            := Axis0001.AxisRef ,
    Execute         := G_bPositioningReq ,
    ContinuousUpdate:= FALSE ,
    Distance        := G_lePoint1Address ,
    Velocity        := lePosVelocity ,
    Acceleration    := lePosAcc ,
    Deceleration    := lePosDec ,
    Jerk            := lePosJerk ,
    BufferMode      := 0 ,//0:mcAborting
    Options         := H0 ,//0:mcAccDec
    Done            => bMove1Done ,
    Busy            => bMove1Busy //,
    //Active        => ?BOOL? ,
    //CommandAborted=> ?BOOL? ,
    //Error         => ?BOOL? ,
    //ErrorID       => ?WORD?
);

//Dwell
TON_1(
    IN:= bMove1Done ,
    PT:= T#500ms ,   //Dwell Time:500ms
    Q => bMove1Dwell //,
    //ET=> ?TIME?
);

//PTP2(Move Absolute)
MC_MoveAbsolute_1(
    Axis            := Axis0001.AxisRef ,
    Execute         := bMove1Dwell ,
    ContinuousUpdate:= FALSE ,
    Position        := G_lePoint0Address ,
    Velocity        := lePosVelocity ,
    Acceleration    := lePosAcc ,
    Deceleration    := lePosDec ,
    Jerk            := lePosJerk ,
    Direction       := 3 ,//3:mcShortestWay
    BufferMode      := 0 ,//0:mcAborting
    Options         := H0 ,//0:mcAccDec
    Done            => bMove2Done ,
    Busy            => bMove2Busy //,
    //Active        => ?BOOL? ,
    //CommandAborted=> ?BOOL? ,
    //Error         => ?BOOL? ,
    //ErrorID       => ?WORD?
);

//Dwell
TON_2(
    IN:= bMove2Done ,
    PT:= T#500ms ,   //Dwell Time:500ms
    Q => bMove2Dwell //,
    //ET=> ?TIME?
);

//Reset Trigger
RST(bMove2Dwell,G_bPositioningReq);
```

Going

Returning

Set the velocity, acceleration/deceleration, and jerk value during the positioning operation.

bMoveEnable turns on only when the interlock conditions are satisfied.

Specify X4 of the remote input module for the positioning start command. The bit called G_bPositioningReq holds the ON state of X4 and uses it as the trigger of Motion Control FB.

Motion Control FB

Standard FB for dwell (On-delay timer)

Motion Control FB

Standard FB for dwell (On-delay timer)

After the reciprocating operation is completed, reset G_bPositioningReq.

4.2.5 Positioning control

The following describes the I/O variables of MC_MoveRelative.

```
14  //PTP1(Move Relative)
15  MC_MoveRelative_1(
16      Axis             := Axis0001.AxisRef ,
17      Execute          := G_bPositioningReq ,
18      ContinuousUpdate:= FALSE ,
19      Distance         := G_lePoint1Address ,
20      Velocity         := lePosVelocity ,
21      Acceleration     := lePosAcc ,
22      Deceleration     := lePosDec ,
23      Jerk             := lePosJerk ,
24      BufferMode       := 0 ,//0:mcAborting
25      Options          := H0 ,//0:mcAccDec
26      Done             => bMove1Done ,
27      Busy             => bMove1Busy //,
28      //Active         => ?BOOL? ,
29      //CommandAborted=> ?BOOL? ,
30      //Error          => ?BOOL? ,
31      //ErrorID        => ?WORD?
32  );
33
```

<MC_MoveRelative specification (extract)>

I/O variable name		Variable name	Data type	Description
Input	Execution command	Execute	BOOL	Executes the positioning control when TRUE is set.
	Continuous update	ContinuousUpdate	BOOL	The movement distance, velocity, acceleration, and deceleration can be continuously changed while TRUE is set.
	Movement distance	Distance	LREAL	Sets the relative position according to the axis unit from the current position at start to the end point.
	Velocity	Velocity	LREAL	Sets the velocity according to the axis unit.
	Acceleration	Acceleration	LREAL	Sets the acceleration according to the axis unit.
	Deceleration	Deceleration	LREAL	Sets the deceleration according to the axis unit.
	Jerk	Jerk	LREAL	Sets the jerk according to the axis unit.
	Buffer mode	BufferMode	MC_BUFFER_MODE	Selects the buffer mode. →4.2.5-4 page
	Option	Options	DWORD(HEX)	Sets the function option. →4.2.5-6 page
Output	Execution completion	Done	BOOL	Turns TRUE after the positioning control is completed.
	Executing	Busy	BOOL	Turns TRUE while the FB is being executed.
	Controlling	Active	BOOL	Turns TRUE when the FB is controlling the axis.
	Abortion of execution	CommandAborted	BOOL	Turns TRUE when the execution is aborted.
	Error	Error	BOOL	Turns TRUE when an error occurs in the FB.
	Error code	ErrorID	WORD (UINT)	Returns the error code occurred in the FB.

Motion Module User's Manual (Application)
6.1 Single Axis Positioning Control

4.2.5 Positioning control

The following describes the I/O variables of MC_MoveAbsolute.

```
//PTP2(Move Absolute)
MC_MoveAbsolute_1(
    Axis             := Axis0001.AxisRef ,
    Execute          := bMove1Dwell ,
    ContinuousUpdate:= FALSE ,
    Position         := G_lePoint0Address ,
    Velocity         := lePosVelocity ,
    Acceleration     := lePosAcc ,
    Deceleration     := lePosDec ,
    Jerk             := lePosJerk ,
    Direction        := 3 ,//3:mcShortestWay
    BufferMode       := 0 ,//0:mcAborting
    Options          := H0 ,//0:mcAccDec
    Done             => bMove2Done ,
    Busy             => bMove2Busy //,
    //Active         => ?BOOL? ,
    //CommandAborted=> ?BOOL? ,
    //Error          => ?BOOL? ,
    //ErrorID        => ?WORD?
);
```

<MC_MoveAbsolute specification (extract)>

I/O variable name		Variable name	Data type	Description
Input	Execution command	Execute	BOOL	Executes the positioning control when TRUE is set.
	Continuous update	ContinuousUpdate	BOOL	The movement distance, velocity, acceleration, and deceleration can be continuously changed while TRUE is set.
	Target position	Position	LREAL	Sets the target position of the absolute position according to the axis unit.
	Velocity	Velocity	LREAL	Sets the velocity according to the axis unit.
	Acceleration	Acceleration	LREAL	Sets the acceleration according to the axis unit.
	Deceleration	Deceleration	LREAL	Sets the deceleration according to the axis unit.
	Jerk	Jerk	LREAL	Sets the jerk according to the axis unit.
	Direction selection	Direction	MC_DIRECTION	Selects the moving direction. →4.2.5-5 page
	Buffer mode	BufferMode	MC_BUFFER_MODE	Selects the buffer mode. →4.2.5-4 page
	Option	Options	DWORD(HEX)	Sets the function option. →4.2.5-6 page
Output	Execution completion	Done	BOOL	Turns TRUE after the positioning control is completed.
	Executing	Busy	BOOL	Turns TRUE while the FB is being executed.
	Controlling	Active	BOOL	Turns TRUE when the FB is controlling the axis.
	Abortion of execution	CommandAborted	BOOL	Turns TRUE when the execution is aborted.
	Error	Error	BOOL	Turns TRUE when an error occurs in the FB.
	Error code	ErrorID	WORD (UINT)	Returns the error code occurred in the FB.

Motion Module User's Manual (Application)
6.1 Single Axis Positioning Control
Absolute Positioning Control

MELSEC iQ-R Programming Manual (Motion Control Function Blocks)
3.2 Operation FBs
MC_MoveAbsolute

4.2.5 Positioning control

The following shows the setting values and descriptions of the buffer mode of MC_MoveAbsolute and MC_MoveRelative.

Setting value	Buffer mode type	Description
0:mcAborting	Aborting	Interrupts (cancels) the FB being executed and executes the next FB immediately.
1:mcBuffered	Buffered	Buffers the next FB on the FB being executed. If the FB being executed has already been buffered, the next FB is buffered on the previous FB. (Up to 2.) When the FB being executed is completed, the buffering FB is executed sequentially.
2:mcBlendingLow	BlendingLow	The lower target velocity between the FB being executed and the buffering FB is the switching speed.
3:mcBlendingPrevious	BlendingPrevious	The target velocity of the FB being executed is the switching speed.
4:mcBlendingNext	BlendingNext	The target velocity of the buffering FB is the switching speed.
5:mcBlendingHigh	BlendingHigh	The higher target velocity between the FB being executed and the buffering FB is the switching speed.

The buffer mode is a function that starts multiple motion control FBs simultaneously and performs the positioning continuously. For details, refer to the MELSEC iQ-R Series Motion Module Basics (RD78G(H)/Positioning Control), which is a course of a online training system, and the following manual.


Motion Module User's Manual (Application)
4.3 Multiple Start (Buffer Mode)

4.2.5 Positioning control

The following shows the setting values and descriptions for selecting the direction of MC_MoveAbsolute.

Ignore this setting when the software stroke limit is valid. Perform the positioning control in a direction in which the area outside the software stroke limit range is not crossed over. However, when both directions do not cross over the area outside the software stroke limit range, positioning control is performed in the direction closer to the target position (the one with the smaller absolute movement distance) based on the current position. If the distance is the same between the positive direction and the negative direction, the operation is performed in the current direction.
When the software stroke limit is invalid, the movement direction from the current position to the target position can be selected from the positive direction, negative direction, and shortest path.

Setting value	Direction selection	Description
1:mcPositiveDirection	Positive direction	Positioning is performed in the positive direction (address increase) from the current position to the target position.
2:mcNegativeDirection	Negative direction	Positioning is performed in the negative direction (address decrease) from the current position to the target position.
3:mcShortestWay	Shortest path	Positioning control is performed in the direction closer to the target position (the one with the smaller absolute movement distance) based on the current position.

For the details, refer to the following manual.

Motion Module User's Manual (Application)
6.1 Single Axis Positioning Control
Absolute Positioning Control

MELSEC iQ-R Programming Manual (Motion Control Function Blocks)
3.2 Operation FBs
MC_MoveAbsolute

4.2.5 Positioning control

The following shows the setting values and descriptions of the options for MC_MoveAbsolute and MC_MoveRelative.

Bit	Description
0 to 2	Acceleration/deceleration method specification (The contents are the same as MCv_Jog.) 0h:mcAccDec 1h:mcFixedTime
3	Only for MC_MoveRelative Position selection during buffer mode 0: Command current position 1: Actual current value For MC_MoveAbsolute, specify "0".
4	Empty (Specify "0".)
5	Reverse rotation permission selection 0: Permit 1: Not permit
6 to 15	Empty (Specify "0".)
16	Only for MC_MoveAbsolute Target position specification exceeding ring counter 0: Not permit 1: Permit For MC_MoveRelative, specify "0".
17 to 31	Empty (Specify "0".)

For the details of the settings on bit 3, bit 5, and bit 16, refer to the following manual.

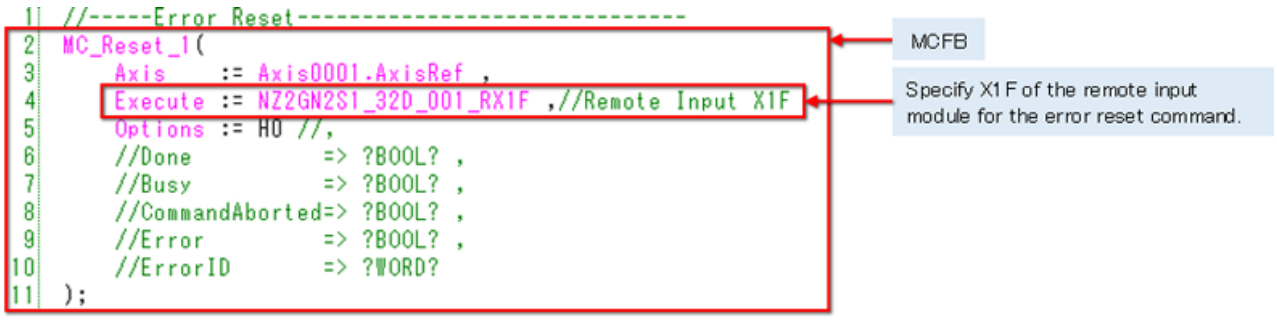
Motion Module User's Manual (Application)
6.1 Single Axis Positioning Control

MELSEC iQ-R Programming Manual (Motion Control Function Blocks)
3.2 Operation FBs
MC_MoveRelative or MC_MoveAbsolute

4.2.6 Error reset

Program name in the sample program:

ErrorReset Use MC_Reset of Motion Control FB.



<MC_Reset specification (extract)>

I/O variable name		Variable name	Data type	Description
Input	Execution command	Execute	BOOL	Executes the error reset when TRUE is set.
	Option	Options	DWORD(HEX)	Specify "0".
Output	Execution completion	Done	BOOL	Indicates that reset has been completed.
	Executing	Busy	BOOL	Turns TRUE while the FB is being executed.
	Abortion of execution	CommandAborted	BOOL	Indicates that the command has been aborted due to timeout. Turns TRUE by setting Execute to FALSE.
	Error	Error	BOOL	Turns TRUE when an error occurs in the FB.
	Error code	ErrorID	WORD (UINT)	Returns the error code occurred in the FB.

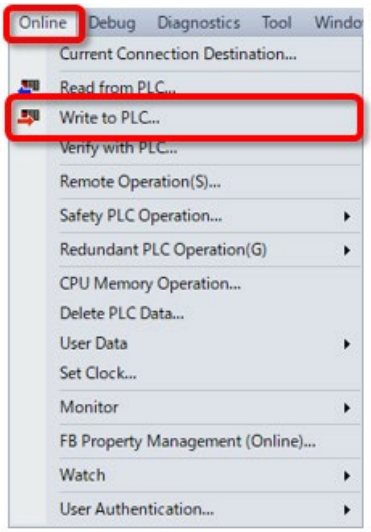
MELSEC iQ-R Programming Manual (Application)
22.3 Error and Warning Reset

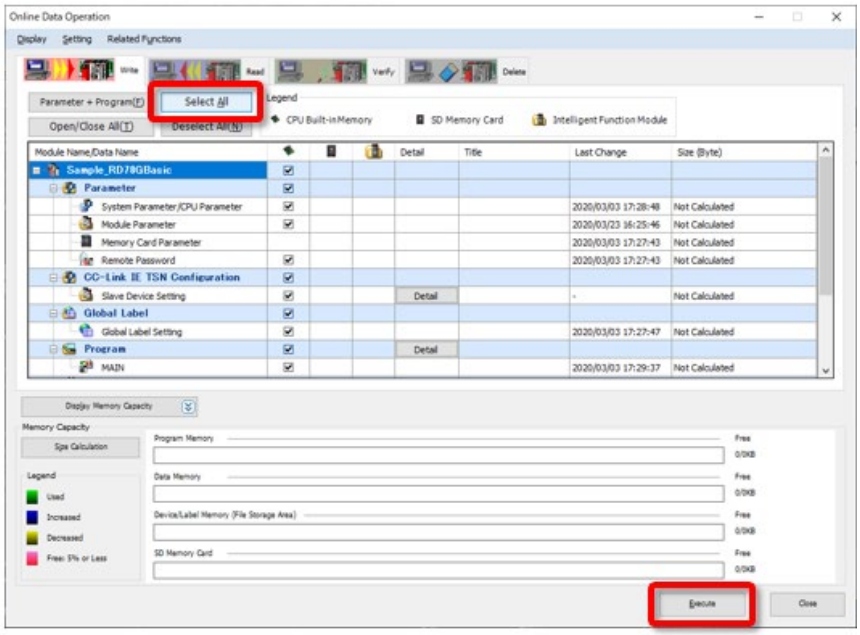
MELSEC iQ-R Programming Manual (Motion Control Function Blocks)
3.1 Management FBs
MC_Reset

4.3 Writing Programs

(1) PLC CPU program

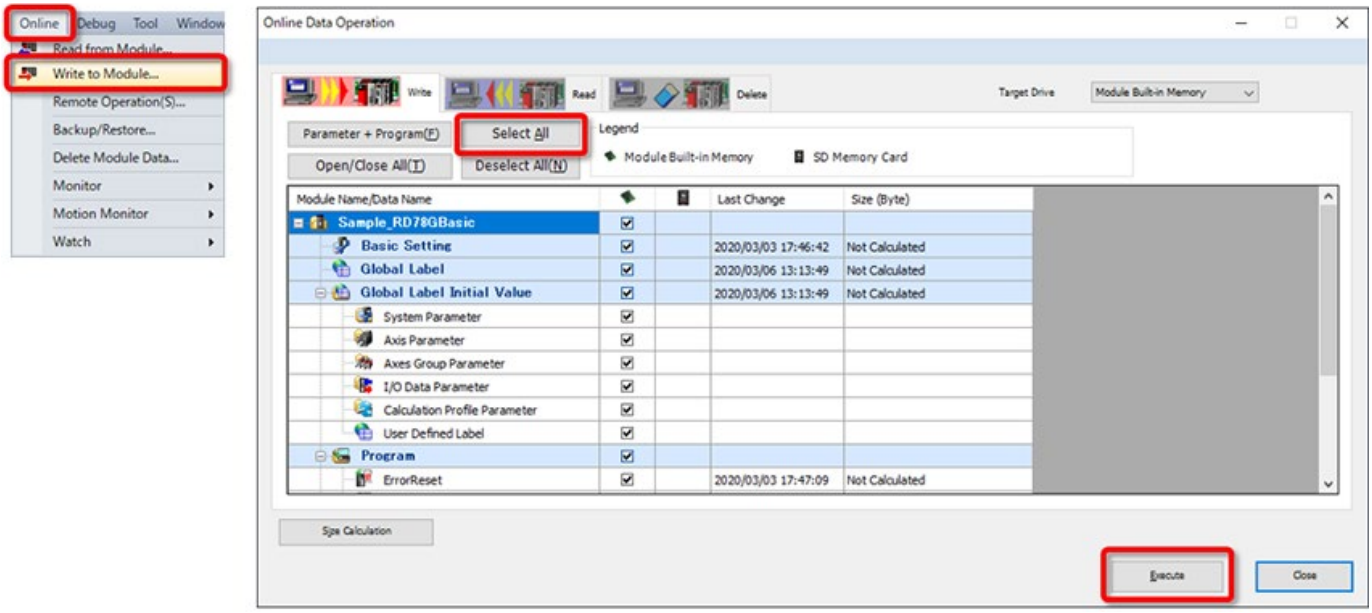
1) Convert all the PLC CPU programs.
2) Set the PLU CPU to the "STOP" state.
3) Select [Online] → [Write to PLC], and click [Select All] on the Write tab in the Online Data Operation screen.
4) Click [Execute] to write data.



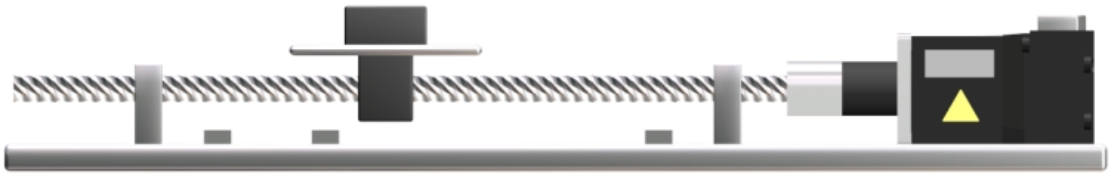


(2) Motion module program

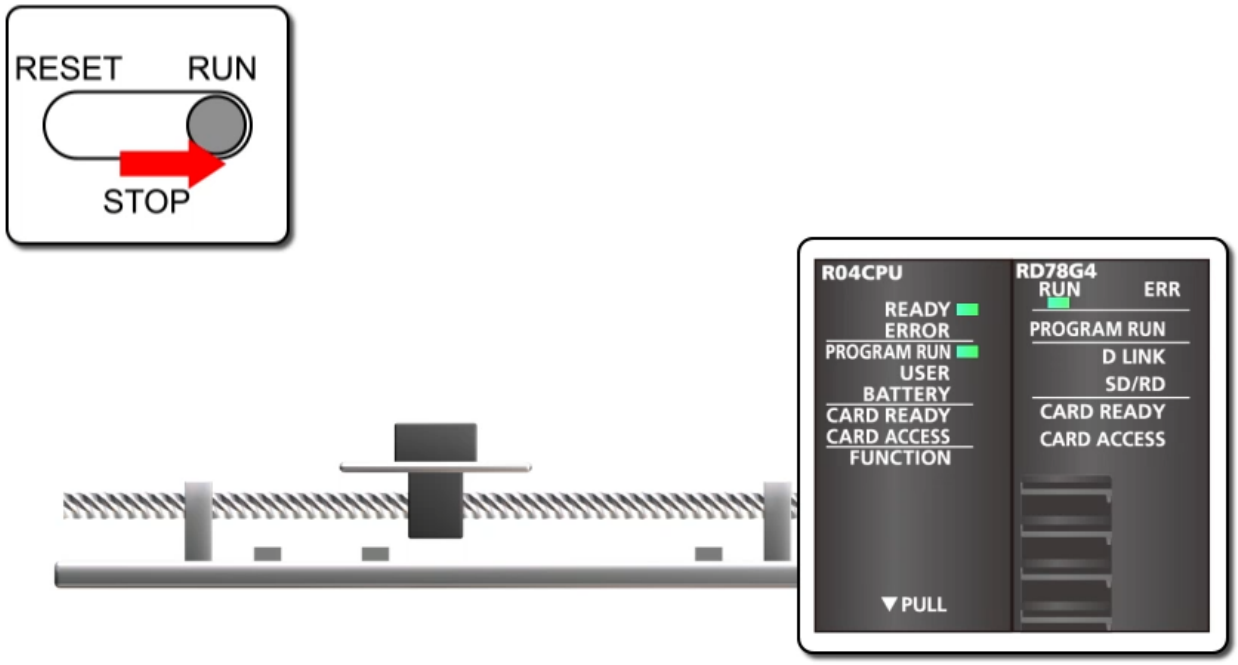
1) Convert all the programs of the motion module on the Motion Control Setting Function screen.
2) Check that the PLC CPU is set to the "STOP" state.
3) Select [Online] → [Write to Module], and click [Select All] on the Write tab in the Online Data Operation screen.
4) Click [Execute] to write data.



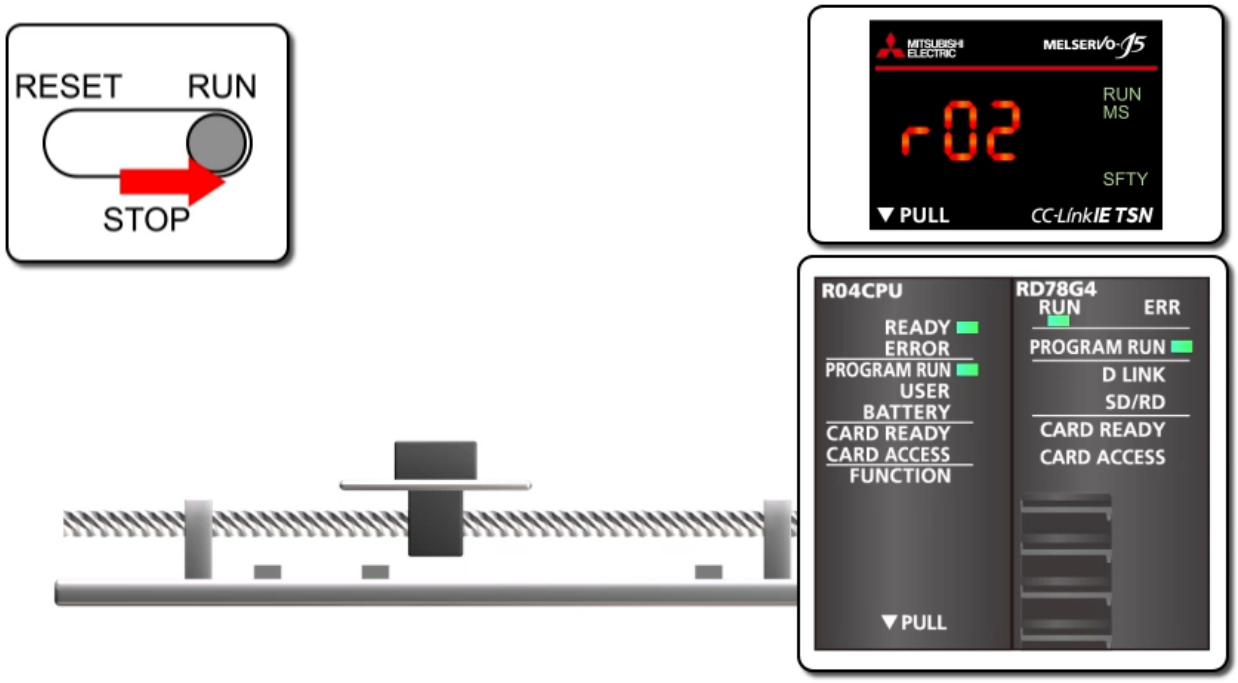
Click the play button at the lower left of the window.



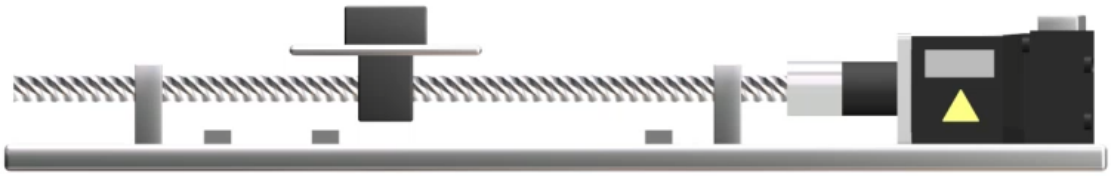
Check the sample program operation.
Start with the state where the programs of the PLC CPU and motion module are written.



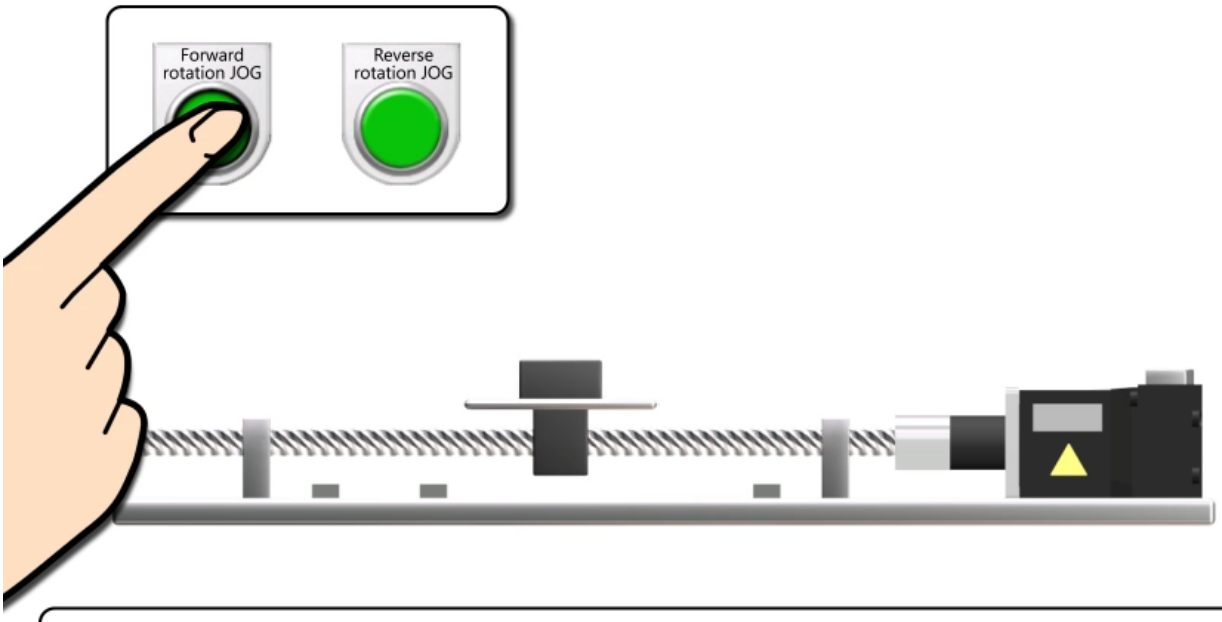
Set the RUN/STOP/RESET switch of the PLC CPU to RUN.
READY and PROGRAM RUN of the PLC CPU turn on.
RUN of the motion module turns on.



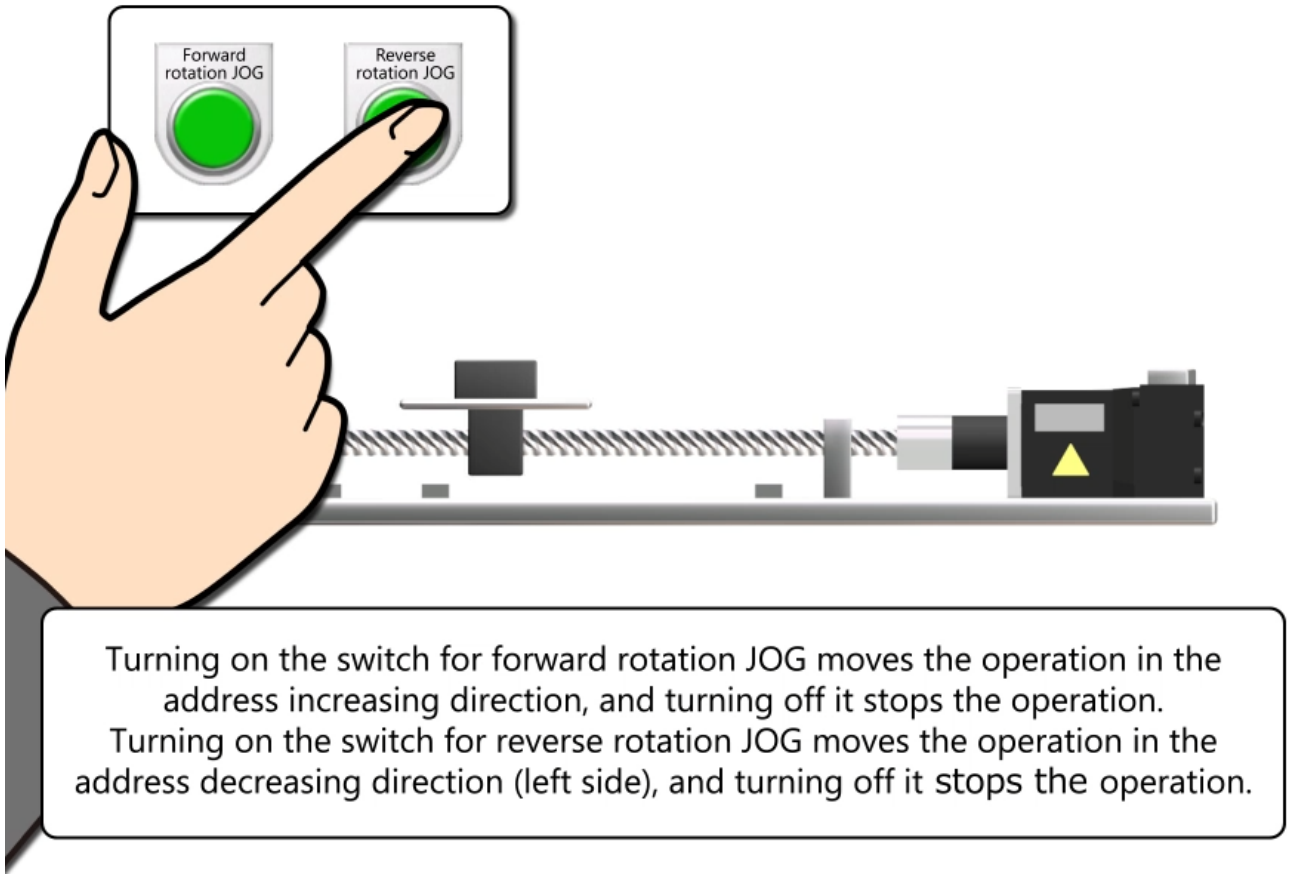
Wait until PROGRAM RUN of the motion module turns on.
"r.02" is displayed on the servo amplifier. (The dot blinks.)



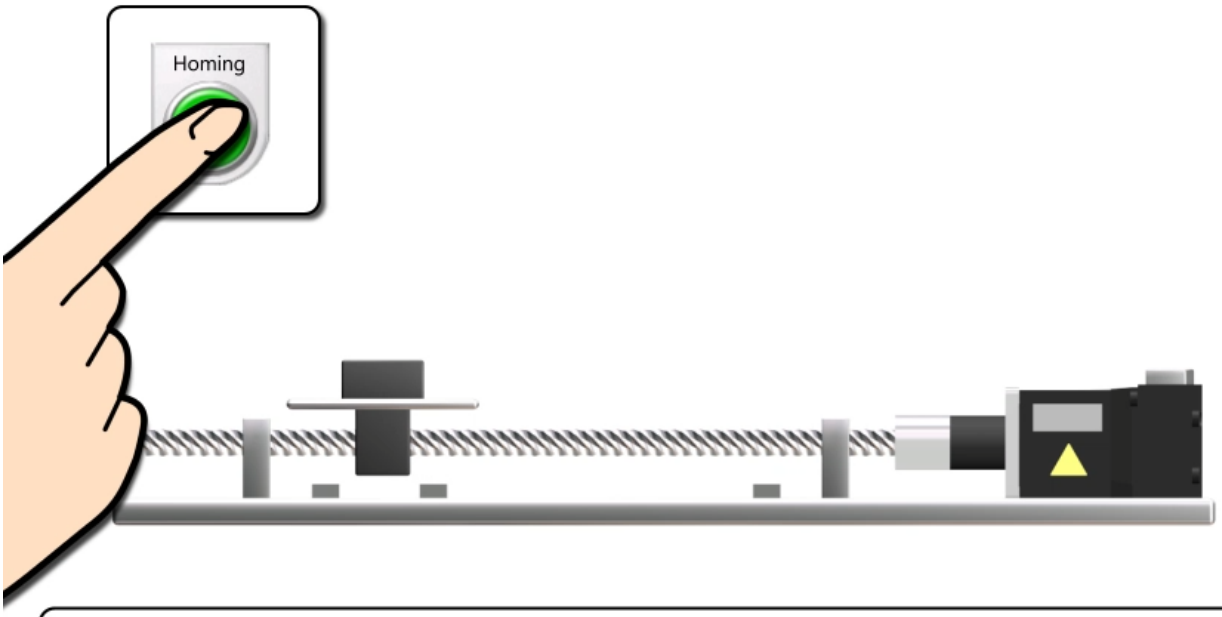
Turn on the servo ON switch (X0 of the remote input module).
"r.02" is displayed on the servo amplifier. (The dot turns on.)
The servo motor enters the servo-on state.



Turning on the switch for forward rotation JOG moves the operation in the address increasing direction, and turning off it stops the operation.
Turning on the switch for reverse rotation JOG moves the operation in the address decreasing direction (left side), and turning off it stops the operation.

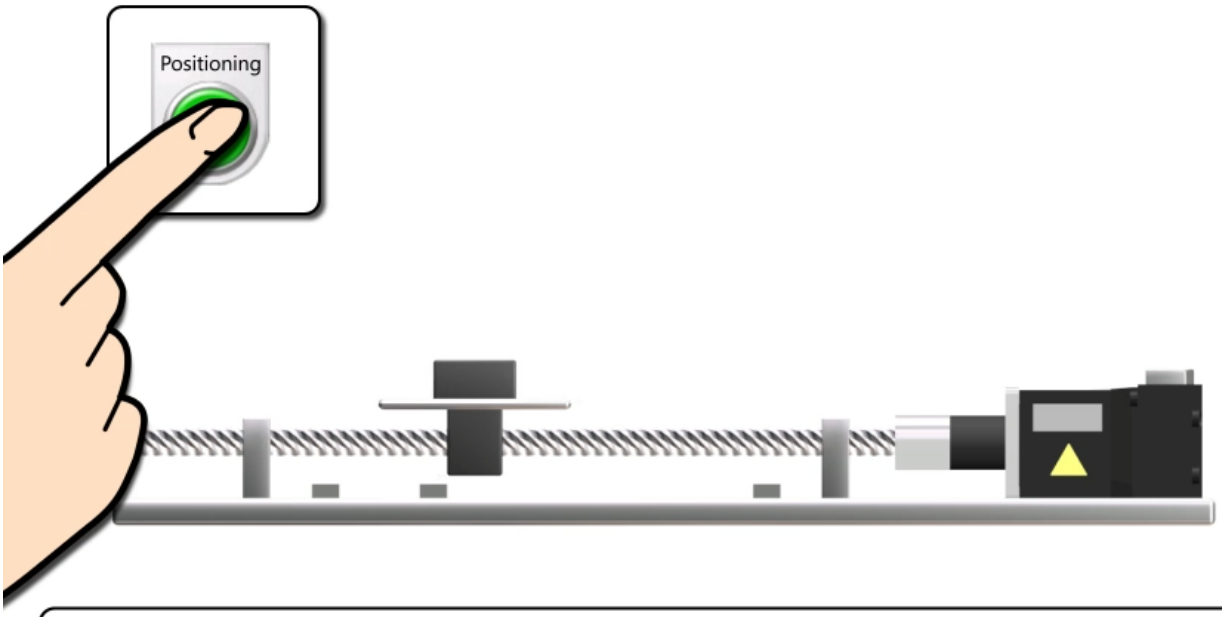


Turning on the switch for forward rotation JOG moves the operation in the address increasing direction, and turning off it stops the operation.
Turning on the switch for reverse rotation JOG moves the operation in the address decreasing direction (left side), and turning off it stops the operation.



Turning on the switch for homing starts the homing operation.
Execute the homing of the proximity dog method (Pr.PT45: -33).
The operation stops at a position where the dog is passed over a little, and the position is set as the home position.

4.4 Operation Check



Tuning on the switch for positioning start starts
the reciprocating operation.
The operation moves forward by 150 mm and stops for 0.5 seconds.
Then, it moves backward by 150 mm and stop for 0.5 seconds.

The operation check has been completed.
Go to the next page.

4.5 Summary of This Chapter

In this chapter, you have learned:

- PLC CPU program
- Motion Module Program
- Writing Programs
- Operation Check

Important points

PLC CPU program	• Always turn on Y0: PLC READY of the motion module on the PLC CPU.
Motion Module Program	• Drag and drop Motion Control FB from the Element Selection window to use. • Use MC_Power for servo-on, MCv_Jog for JOG operation, MC_Home for home position return, MC_MoveRelative for relative value positioning, MC_MoveAbsolute for absolute value positioning, and MC_Reset for error reset. • Set the home position return method with parameters of the servo amplifier.
Writing Programs	• Write the programs to the PLC CPU and motion module.
Operation Check	• The sample system operation is checked in the video.

Test Final Test

Now that you have completed all of the lessons of the MELSEC iQ-R Series Motion Module Basics (RD78G(H)/Startup) 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 (7 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 description(s) below. (Multiple selections are available)

Q1

- [ ] **Velocity waveforms during the jerk acceleration/deceleration change smoothly.**
- [ ] **If the jerk value is increased, the acceleration/deceleration time becomes longer.**
- [ ] **In the motion module, programs are created with the FBs created by Mitsubishi Electric.**
- [ ] **Statements must end with a ": (colon)" in the ST.**
- [ ] **Local labels can be used only in each POU.**

Test **Final Test 2**

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

- To perform the test operation, change the (Q1) of the servo amplifier before power-on.
- Check the motor rotation direction and machine operations with the test operation function of (Q2).
- Set the (Q3) with the rotary switches of the remote input module and servo amplifier.

Q1 Select the corresponding word.

Q2 Select the corresponding word.

Q3 Select the corresponding word.

Q1:
- DIP switch
- Rotary switch
- Command switch

Q2:
- GX Works3
- MR Configurator2
- Motion control setting function

Q3:
- IP address
- Station number

Test **Final Test 3**

Select the correct description(s) below. (Multiple selections are available)

Q1

- [ ] **Once the PDO mapping is performed, there is no problem even if the network configuration is changed.**
- [ ] **Parameters of the servo amplifier can be transferred from the controller at the time of initial communication or can be written to each axis using MR Configuretor2.**
- [ ] **The link direct device setting of the CPU parameters must be set to the extended mode (iQ-R series mode).**

Test **Final Test 4**

Select the correct description(s) about the program when using the motion module. (Multiple selections are available)

Q1

- [ ] **Always turn on Y0 of the motion module in the program of the PLC CPU.**
- [ ] **By turning on Y1 of the motion module, the servo turns on.**
- [ ] **Motion Control FB can be written to the program editor by drag-and-drop action.**
- [ ] **All the I/O signals of Motion Control FB must be set.**

Test **Final Test 5**

Select a correct description about the settings of the homing method.

Q1

- Set the homing method with the input variable "Options" in the FB "MC_Home".
- Set the homing method with the axis parameters on the Motion Control Setting Function screen.
- Set the homing method with the parameters of the servo amplifier MR-J5-G.

Test **Final Test 1**

Select the correct description(s) below. (Multiple selections are available)

Q1

- [x] **Velocity waveforms during the jerk acceleration/deceleration change smoothly.**
- [ ] **If the jerk value is increased, the acceleration/deceleration time becomes longer.**
- [ ] **In the motion module, programs are created with the FBs created by Mitsubishi Electric.**
- [ ] **Statements must end with a ": (colon)" in the ST.**
- [x] **Local labels can be used only in each POU.**

Test **Final Test 2**

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

- To perform the test operation, change the (Q1) of the servo amplifier before power-on.
- Check the motor rotation direction and machine operations with the test operation function of (Q2).
- Set the (Q3) with the rotary switches of the remote input module and servo amplifier.

Q1 1: DIP switch

Q2 2: MR Configurator2

Q3 1: IP address

Q1: • DIP switch
• Rotary switch
• Command switch

Q2: • GX Works3
• MR Configurator2
• Motion control setting function

Q3: • IP address
• Station number

Test Final Test 3

Select the correct description(s) below. (Multiple selections are available)

Q1

- [ ] **Once the PDO mapping is performed, there is no problem even if the network configuration is changed.**
- [x] **Parameters of the servo amplifier can be transferred from the controller at the time of initial communication or can be written to each axis using MR Configuretor2.**
- [x] **The link direct device setting of the CPU parameters must be set to the extended mode (iQ-R series mode).**

Test Final Test 4

Select the correct description(s) about the program when using the motion module. (Multiple selections are available)

Q1

- [x] **Always turn on Y0 of the motion module in the program of the PLC CPU.**
- [ ] **By turning on Y1 of the motion module, the servo turns on.**
- [x] **Motion Control FB can be written to the program editor by drag-and-drop action.**
- [ ] **All the I/O signals of Motion Control FB must be set.**

Test Final Test 5

Select a correct description about the settings of the homing method.

Q1

- Set the homing method with the input variable "Options" in the FB "MC_Home".
- Set the homing method with the axis parameters on the Motion Control Setting Function screen.
- **Set the homing method with the parameters of the servo amplifier MR-J5-G.**

Test **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: **7**

Correct answers: **7**

Percentage: **100** %

Clear

You have completed the "MELSEC iQ-R Series Motion Module Basics (RD78G(H)/Startup)" 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