MITSUBISHI ELECTRIC
Changes for the Better

Servo

MOTION CONTROLLER Basics (Real Mode:SFC)

This course is a training system for those who establish the motion control system using the motion CPU module of Mitsubishi motion controller Q series for the first time.

L(NA)00128ENG

Introduction Purpose of the Course

This course is for those who are going to establish the motion control system using the motion CPU module of Mitsubishi motion controller Q series for the first time to acquire the procedures including installing the operating system, setting the system, programming and debugging in the motion SFC language by using the motion controller engineering environment MELSOFT MT Works2.

The main contents of this course are for a person in charge of software.
The contents for a person in charge of hardware, such as the system design, installation, wiring and so on are prepared in the "SERVO MOTION CONTROLLER BASICS (HARDWARE)" course.

For this course, you are required to have knowledge about MELSEC-Q series PLC, AC servo and positioning control.
For those who take this course for the first time, we recommend to take
"MELSEC-Q SERIES BASICS" course,
"MELSERVO (MR-J3) BASICS" course,
"YOUR FIRST FACTORY AUTOMATION (POSITIONING CONTROL)" course.

Introduction Course Structure

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

Chapter 5 - BASICS OF MOTION CONTROL

You will learn the basics of the motion control system.

Chapter 6 - SELECTING AND INSTALLING THE OPERATING SYSTEM SOFTWARE

You will learn how to select and install the operating system software of the motion CPU module.

Chapter 7 - PARAMETER SETTING

You will learn how to set the system of the motion CPU module and each parameter.

Chapter 8 - OPERATION CHECK

You will learn how to check the operation of the servo motor and execute the home position return.

Chapter 9 - PROGRAM DESIGN

You will learn how to design a program.

Chapter 10 - MOTION SFC PROGRAM

You will learn the basics of the motion SFC program for the motion control.

Chapter 11 - PROGRAMING

You will learn how to program and debug the motion SFC program with MT Developer2.

Final Test

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		"Table of Contents" will be displayed, enabling you to navigate to the desired page.
Exit the learning		Exit the learning. Window such as "Contents" screen and the learning will be closed.

Introduction Cautions for Use

Safety precautions

When you learn by using actual products, please 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 version:

- MT Developer2 Version 1.18U
- MR Configurator2 Version 1.01B
- GX Works2 Version 1.55H

Reference

The following is the reference related to the learning. (You can learn without them.)
Click the reference name to download.

Reference name	File type	Size
Sample program	Compressed file	166.5 kB
Recording paper	Compressed file	5.57 kB

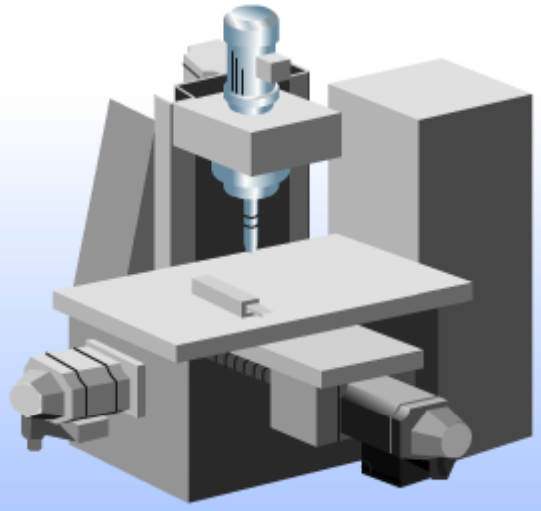
Chapter 5 BASICS OF MOTION CONTROL

The motion controller controls the multiple axes (servo motors) for a conveyor assembly, a processing machine etc. and performs the high-precision positioning control and speed control.
In this course, the system building and the program development of the established motion control system are prepared for a person in charge of software.

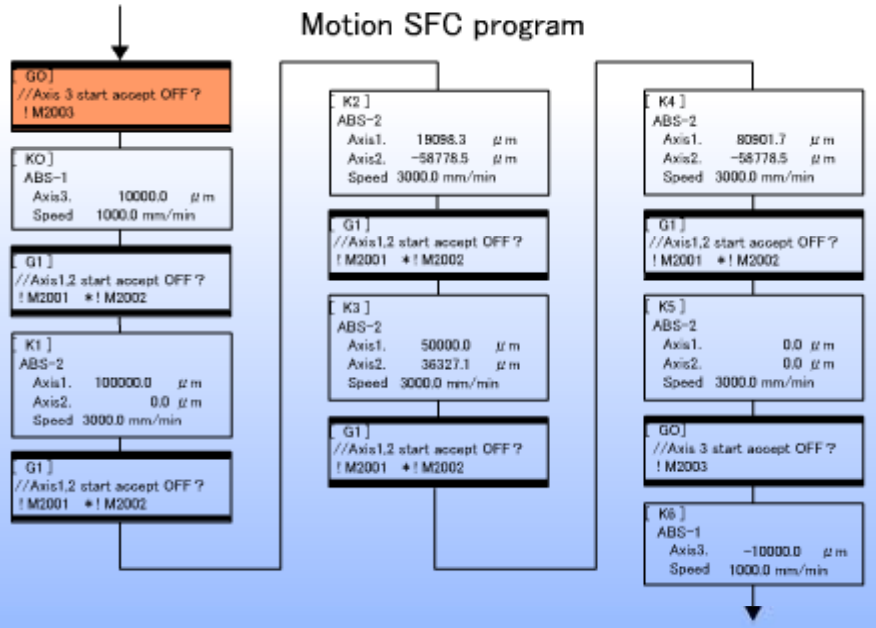
The application examples of the motion control are introduced in the following. Click the button of the application example which you would like to see.

X-Y table | Sealing | Spinner | Filling machine

■X-Y table



Motion SFC program



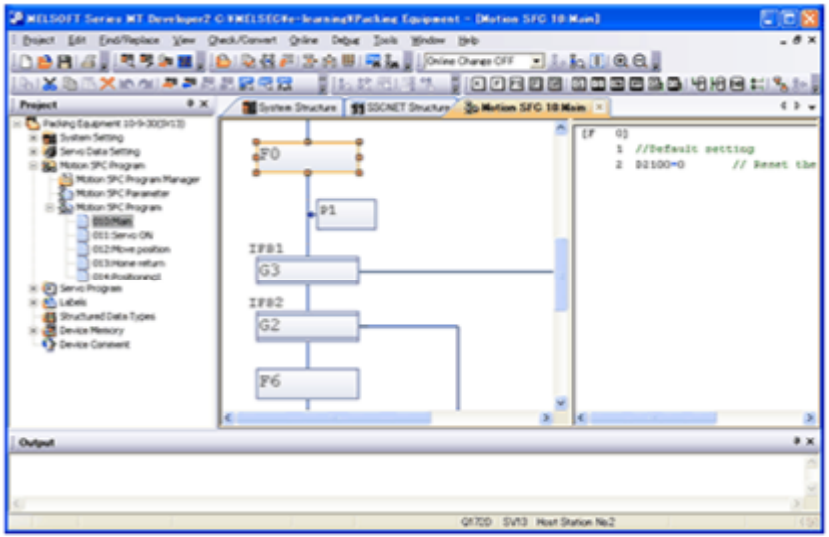
5.1 Development and Maintenance Environment of Motion Control System

For the development and maintenance environment of the motion control system, use the motion controller engineering environment MELSOFT MT Works2 and the servo setup software package MELSOFT MR Configurator2.
The following lists the main functions of each software.

- MELSOFT MT Works2
- MT Developer2

The development and maintenance environment of the motion control system

- Controlling the project
- Setting the system configuration
- Setting the servo data
- Testing the servo motor operation
- Creating a program in the motion SFC language
- Debugging and monitoring the program
- Writing or reading the program and the parameter
- Installing the operating system software



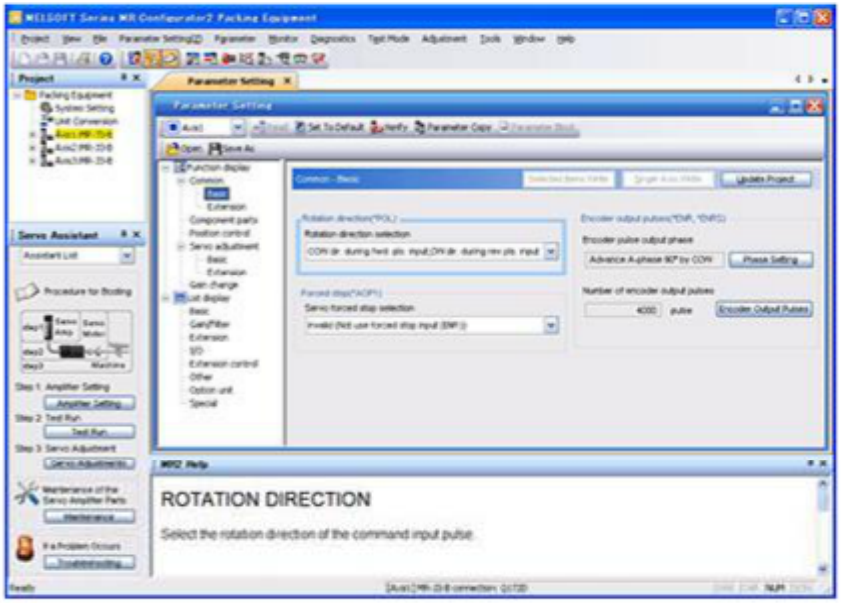
- MT Simulator2

The simulation environment of the motion SFC program

- MELSOFT MR Configurator2

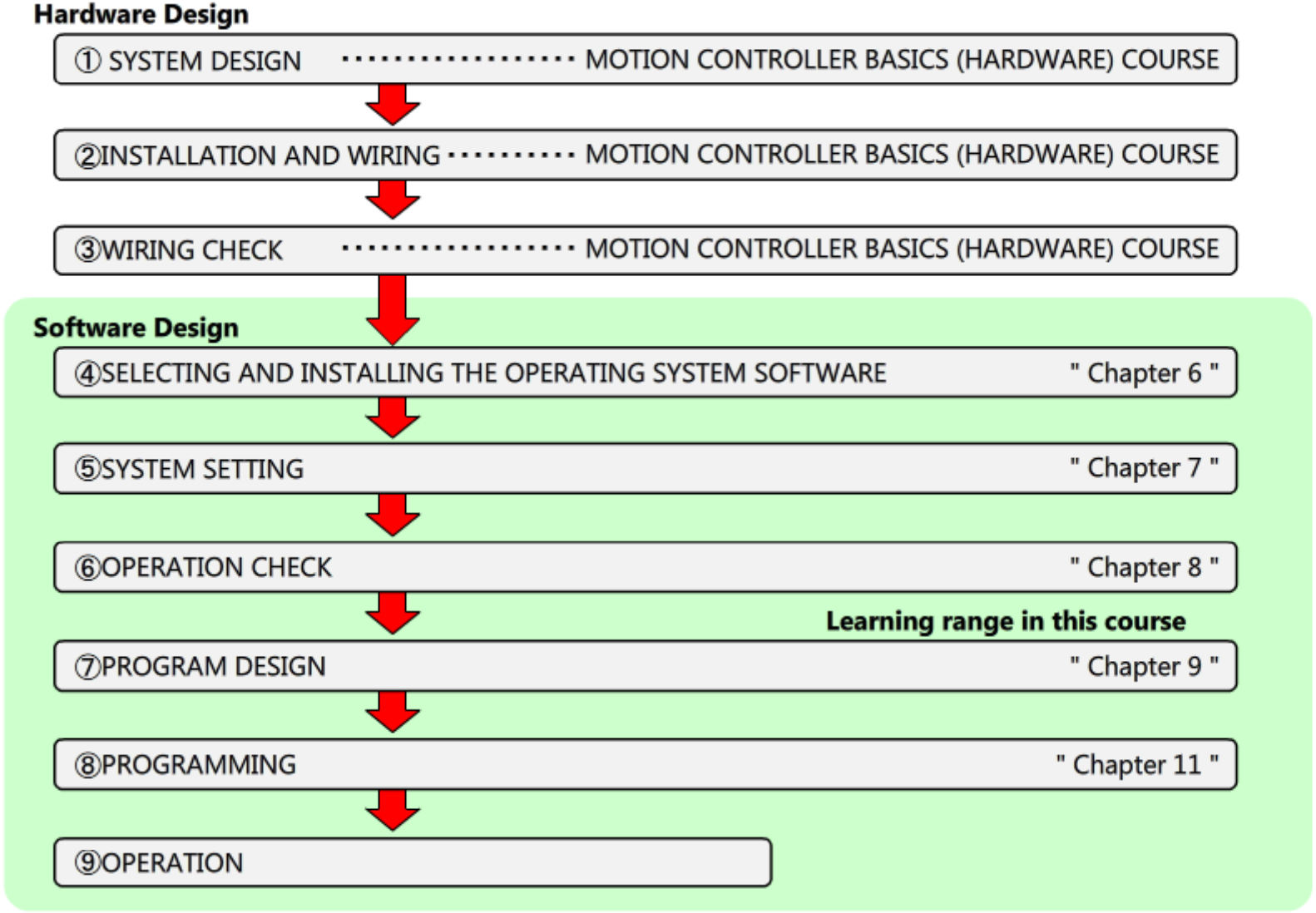
The setup environment of the servo amplifier and the servomotor

- Setting the servo parameter
- The operation test and gain adjustment of the servo amplifier



5.2 Procedures to Establish the Motion Control System

The following shows the procedure to establish the motion control system.
In this course, you will learn the process of the software design along with the establishment procedures.

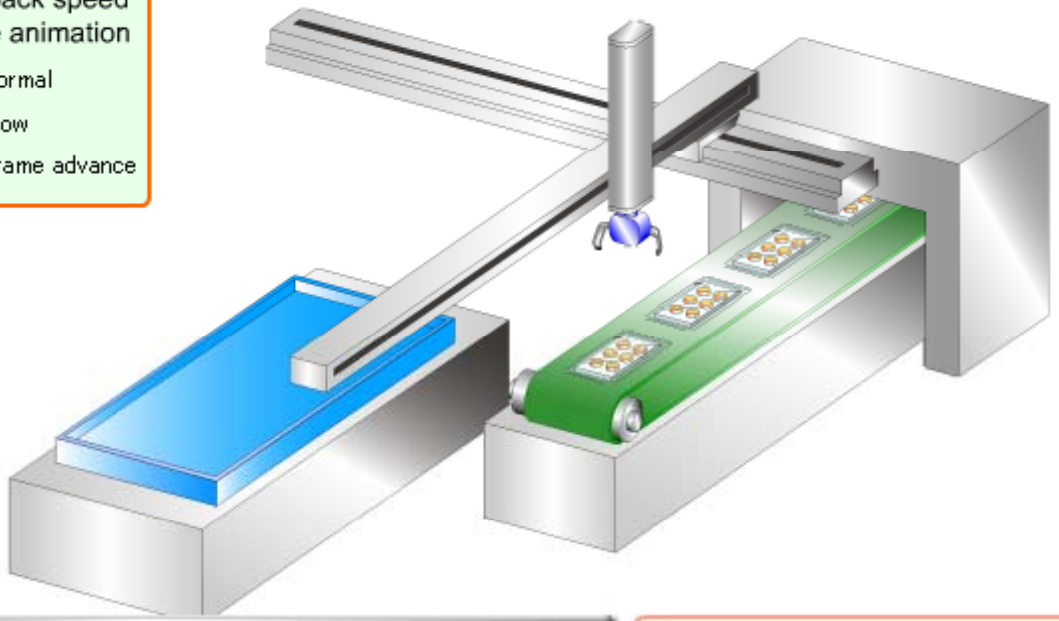


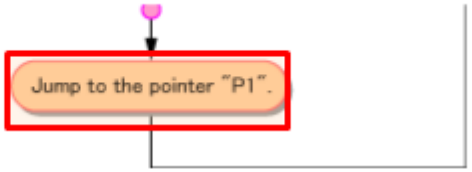
5.3 CONTROL FLOW

Check the control mode (control flow) in the sample system for this course by using the animation.

Operate the animation in the following sample system by a mouse according to the instruction of Click .

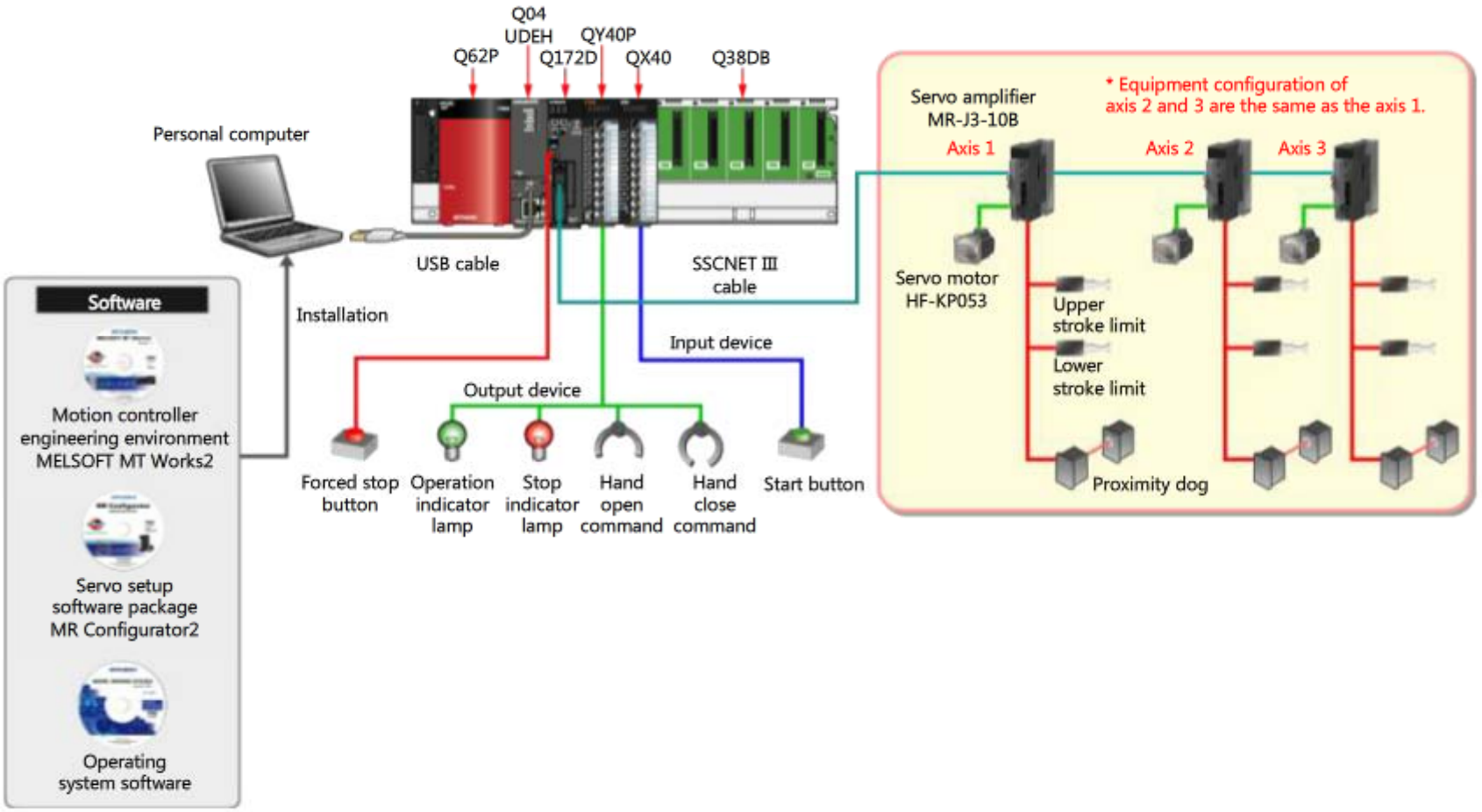
Playback speed of the animation
- (•) Normal
- ( ) Slow
- ( ) Frame advance





To arrange the next goods onto the pallet,
the control flow returns back to the pointer (P1).

5.3.1 Equipment configuration of the sample system for this course



5.3.1 Equipment configuration of the sample system for this course

The following table lists the equipment configuration of the sample system.

Item	Configuration component	Quantity	Model name	Description
Motion controller system	Base unit	1	Q38DB	a base unit which has 8 slots for mounting each module and supports multiple CPU
	Power supply module	1	Q62P	Supplies power to each module.
	PLC CPU module	1	Q04UDECPU	A CPU module which performs the sequence control * The battery (Q6BAT) is enclosed with the CPU module.
	Motion CPU module	1	Q172DCPU	A CPU module which performs the motion control * The battery (Q6BAT) and the battery holder (Q170DBATC) are enclosed with the CPU module.
	Input module	1	QX40	Inputs the ON/OFF signal from the start button. (16 points)
	Output module	1	QY40P	Outputs the ON/OFF signal to the indicator lamp and the device (the hand part). (16 points)
	External power supply	1	–	Supplies 24VDC power to the I/O devices and the forced stop input.
External I/O device	Start button	1	–	A push button switch to start the sample system
	Forced stop button	1	–	A push button switch to stop the servo motors of all the axes at an emergency.
	Cable for forced stop input	1	Q170EMICBL_M	Used for wiring the forced stop input to the motion CPU module.
	Hand part of the device	1	–	The hand part of the device for catching the goods
	Indicator lamp	2	–	The indicator lamps to inform if the system is in operation or stops.
	Servo amplifier	3	MR-J3-10B	A servo amplifiers for 3 axes
	Servo motor	2	HF-KP053	Servo motors for the axis 1 (X-axis) and the axis 2 (Y-axis)
		1	HF-KP053B	A servo motor with a brake for the axis 3 (Z-axis)
	Stroke limit	6	–	Sensors to detect the upper limit and the lower limit in the movable range of the device

5.3.1 Equipment configuration of the sample system for this course

Servo system	Proximity dog	3	–	Sensors to detect the starting position of deceleration at the home position return
	Motor power supply cable	3	MR-PWS1CBL2M-A1-L	A cable to conduct the power from the servo amplifier to the servomotor (Length: 2m)
	Encoder cable	3	MR-J3ENCBL2M-A1-L	A cable to connect the servo amplifier and the encoder of the servo motor (Length: 2m)
	SSCNET III cable	3	MR-J3BUS_M	A communication cable between the motion CPU module and the servo amplifier
Development environment	Personal computer	1	–	A personal computer to run the engineering environment software
	Engineering environment software	1	MT Works2	Software to set the motion CPU module, to program and so on
		1	GX Works2	Software to set the PLC CPU module, to program and so on
		1	MR Configurator2	Setup software to set the servo amplifier and the servo motor
	Operating system software	1	SW8DNC-SV13QD	Software to be installed to the motion CPU module
	USB cable	1	MR-J3USBCBL3M	

5.4 Summary

The following lists the contents you learned in Chapter 5.
The following points are very important, so please check them again.

Motion control overview	The motion controller controls the multiple axes (servo motors) for a conveyor assembly, a processing machine etc. and performs the high-precision positioning control and speed control.
Development and maintenance environment of motion control system	For the development and maintenance environment of the motion control system, use the motion controller engineering environment MELSOFT MT Works2 and the servo setup software package MELSOFT MR Configurator2.

Chapter 6 SELECTING AND INSTALLING THE OPERATING SYSTEM SOFTWARE

In Chapter 6, you will learn how to select and install the operating system software of the motion CPU module.

SELECTING AND INSTALLING THE OPERATING SYSTEM SOFTWARE ····· "Chapter 6"

↓

SYSTEM SETTING ····· "Chapter 7"

↓

OPERATION CHECK ····· "Chapter 8"

↓

PROGRAM DESIGN
····· "Chapter 9"(BASICS OF MOTION SFC: Chapter 10)

↓

PROGRAMMING ····· "Chapter 11"

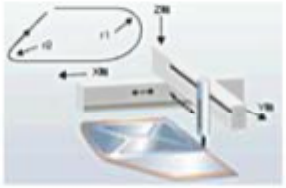
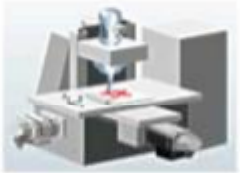
↓

OPERATION

Learning procedure of Chapter 6

6.1 Type and Selection of the Operating System Software

Select a motion CPU module and install the operating system software (control software) according to the application of a conveyor assembly,
a processing machine and so on.
There are the following 3 types of the operating system software according to the application.
In the sample system, select and install SV13 which is for a conveyor assembly.

Item	Conveyor assembly use (SV13)	Automatic machinery use(SV22)	Machine tool peripheral use (SV43)
Application			
Equipment example	Electronic components assembly equipment, conveying equipment, paint applicator, chip mounting, wafer slicer, loader and unloader, bonding machine, X-Y table	Food packaging machine, food processing machine, winding machine, spinning machine, textile machine, printing machine, book binder, press feeder, tire molder	Grinding machine, transfer machine, woodworker, loader and unloader
Positioning program	Dedicated language supporting the motion SFC	Mechanical support language supporting the motion SFC	EIA language (G code)
	Dedicated language Control method by the programming language suitable to the motion control such as the positioning control and other	Mechanical support language Method performing the synchronous control only by writing the configuration of the mechanical system	G code Method of using the normalized (coded) numerical value (00 to 101) which specifies the control function of the axis in the NC device

Precautions

- The operating system software is not installed at the time of the motion CPU module purchase.
  Please install the software following the procedures on the next screen.
- The operating system software is sold separately. Purchase the operating system software with the motion CPU module.

6.2 Installing the Operating System Software

Install the operating system software to the motion CPU module. Follow the procedure below.

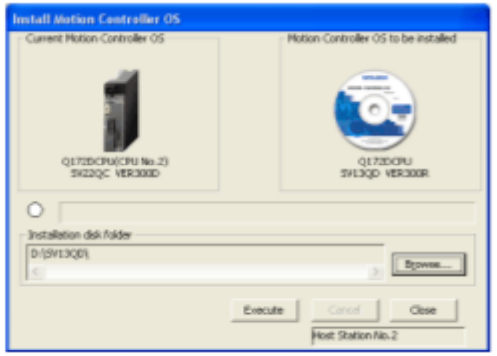
① Turn off the motion controller.
Switch the RUN/STOP switch of the motion CPU module to STOP.
Connect a personal computer and a PLC CPU module by a USB cable.

② Change the rotary switch for the function select of the motion CPU module to "Installation mode"
(Switch for function select 1: "A", Switch for function select 2: "0")

③ Turn on the motion controller.
LED display becomes "INS" (Installation mode).

④ Start the MT Developer2 and set the transfer setup.
(Install the USB driver as necessary .)

⑤ Set the CD-ROM of the operating system software to the personal computer and execute the installation from the MT Developer2.
After the installation, turn off the motion controller.



⑥ Change the rotary switch for the function select.
(Switch for function select 1: "0", Switch for function select 2: "0")

⑦ Turn on the motion controller.
LED display becomes "AL" (Motion error).
* "AL" is displayed because the parameter is not set at this moment, but it is not a problem.

6.3 Summary

The following lists the contents you learned in Chapter 6.
The following points are very important, so please check them again.

Type and selection of the operating system software	・Select a motion CPU module and install the operating system software (control software) according to the application of a conveyor assembly, processing machine and so on. Conveyor assembly use (SV13) Automatic machinery (SV22) Machine tool peripheral use (SV43) ・The operating system software is not installed at the time of the motion CPU module purchase. ・The operating system software is sold separetely. Purchase the operating system software with the motion CPU module.
Select and installing the operating system software	・Before the installation, change the rotary function select switch of the motion CPU module to the installation mode. (Switch for function select 1: "A", Switch for function select 2: "0") After the installation, return the rotary function select 1 switch to "0" the rotary function select 2 switch to "0". ・Execute the installation by the installation function of MT Developer2.

Chapter 7 SYSTEM SETTING

In Chapter 7, you will learn how to set the system of the motion CPU module and each parameter.

SELECTING AND INSTALLING THE OPERATING SYSTEM SOFTWARE ····· "Chapter 6"

↓

SYSTEM SETTING ····· "Chapter 7"

↓

OPERATION CHECK ····· "Chapter 8"

↓

PROGRAM DESIGN ····· "Chapter 9"(BASICS OF MOTION SFC: Chapter 10)

↓

PROGRAMMING ····· "Chapter 11"

↓

OPERATION

Learning procedure of Chapter 7

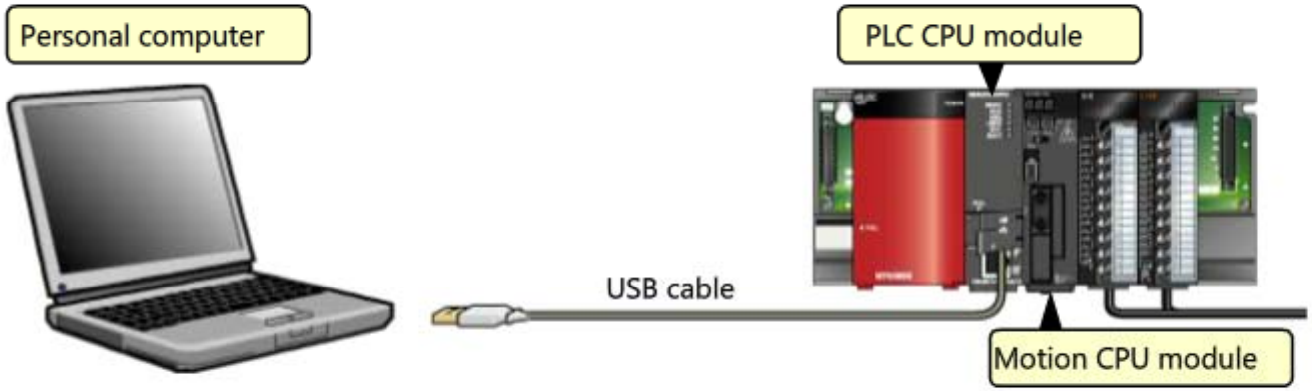
7.1 Transfer Setup
7.2 Creating a Project
7.3 System Setting
  7.3.1 Basic system setting
  7.3.2 System configuration setting
  7.3.3 SSCNET configuration setting
7.4 Servo Data Setting
  7.4.1 Fixed parameter setting
  7.4.2 Home position return data setting
  7.4.3 JOG operation data setting
7.5 Servo Parameter Setting
7.6 Parameter Block Setting
7.7 Saving a Project
7.8 Writing Parameters to the Motion CPU Module

7.1 Enabling Communications between the Personal Computer and the Motion CPU Module

Before setting parameters, enable communications between the personal computer to which MT Developer2 is installed and the motion CPU module, and apply setting data to the motion CPU module.

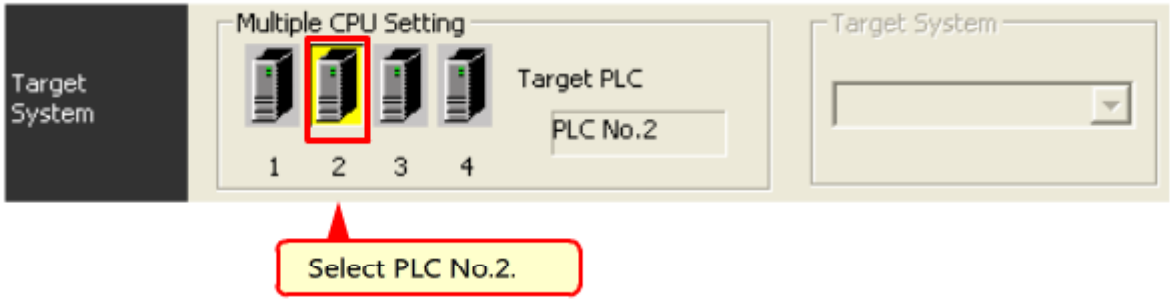
Setting procedure

- Connect the personal computer and the PLC CPU module by a USB cable.
- Set the transfer setup with MT Developer2.
  The screen of transfer setup and the operation are same as those of GX Works2.



Point of the transfer setup

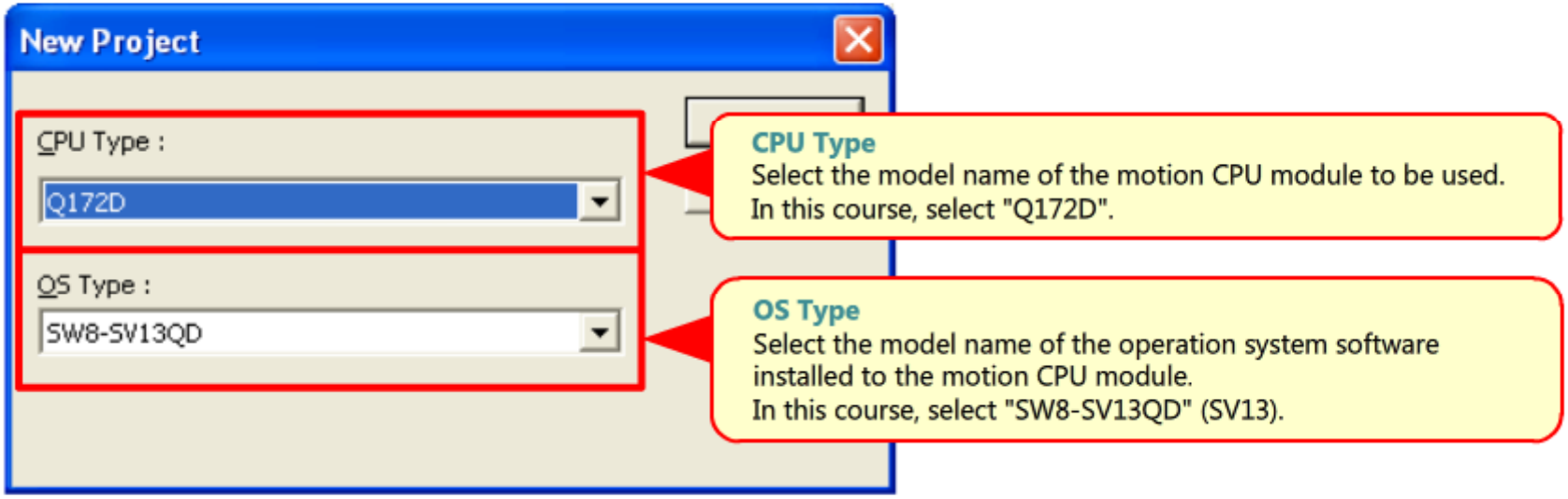
Since the communication target motion CPU module is mounted to the CPU slot 2 of the base unit, select PLC No.2 in the transfer setup.



7.2 Creating a Project

After completion of the transfer setup, create a new project.
A project is a unit that is used to control various parameters and programs by MT Developer2.

Set the following to create a project.
Select the motion CPU module type and the operation system software type.

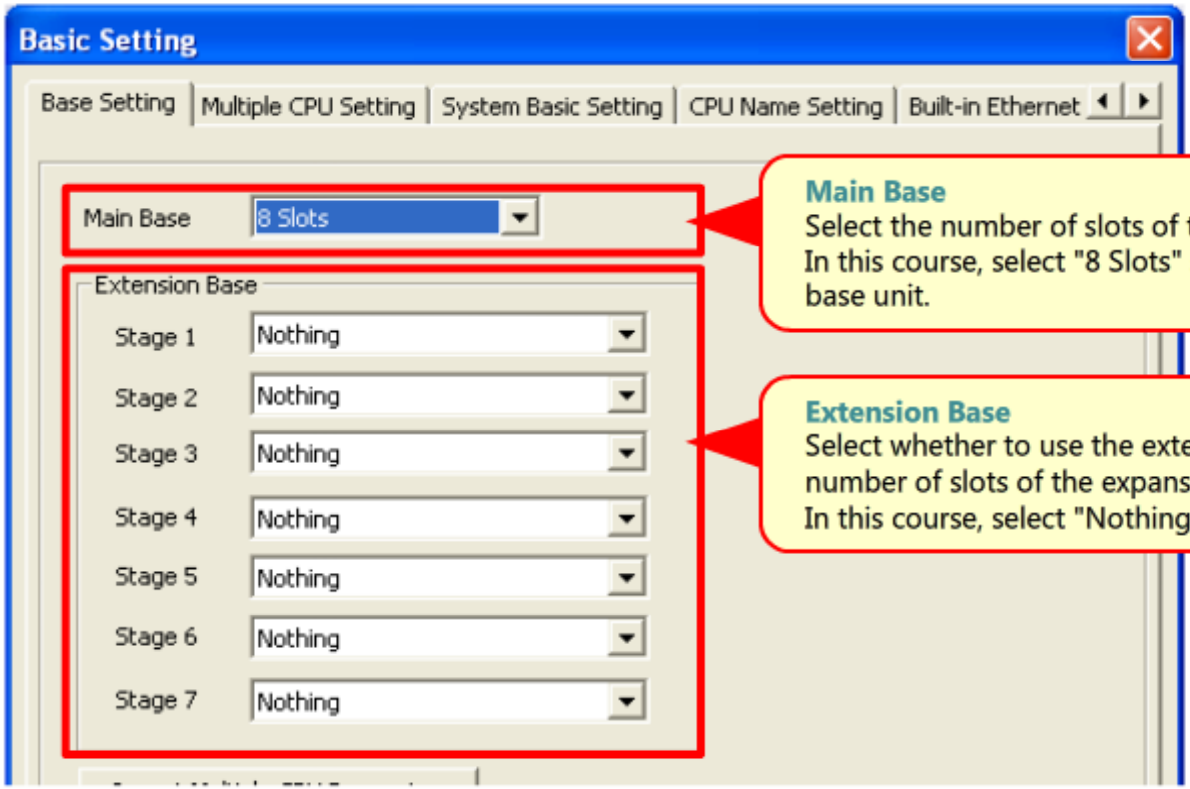


7.3 System Setting

After creating a project, set the system first.
Set the motion CPU module and servo according to the actual system configuration.

7.3.1 Basic system setting

First, set the Basic Setting. (After creating a project, a dialog box appears.)
The basic system setting includes such as base unit, multiple CPU, etc.
In this course, set the parameters in Base Setting. (For the other settings, use the default values.)



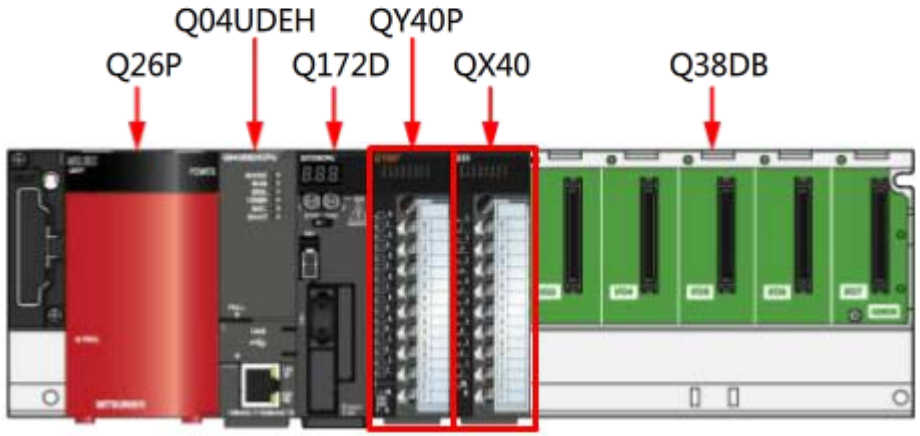
Main Base
Select the number of slots of the main base unit to be used.
In this course, select "8 Slots" since Q38DE is selected for the base unit.

Extension Base
Select whether to use the extension base unit or not and the number of slots of the expansion base unit.
In this course, select "Nothing" for all extension base stages.

7.3.2 System configuration setting

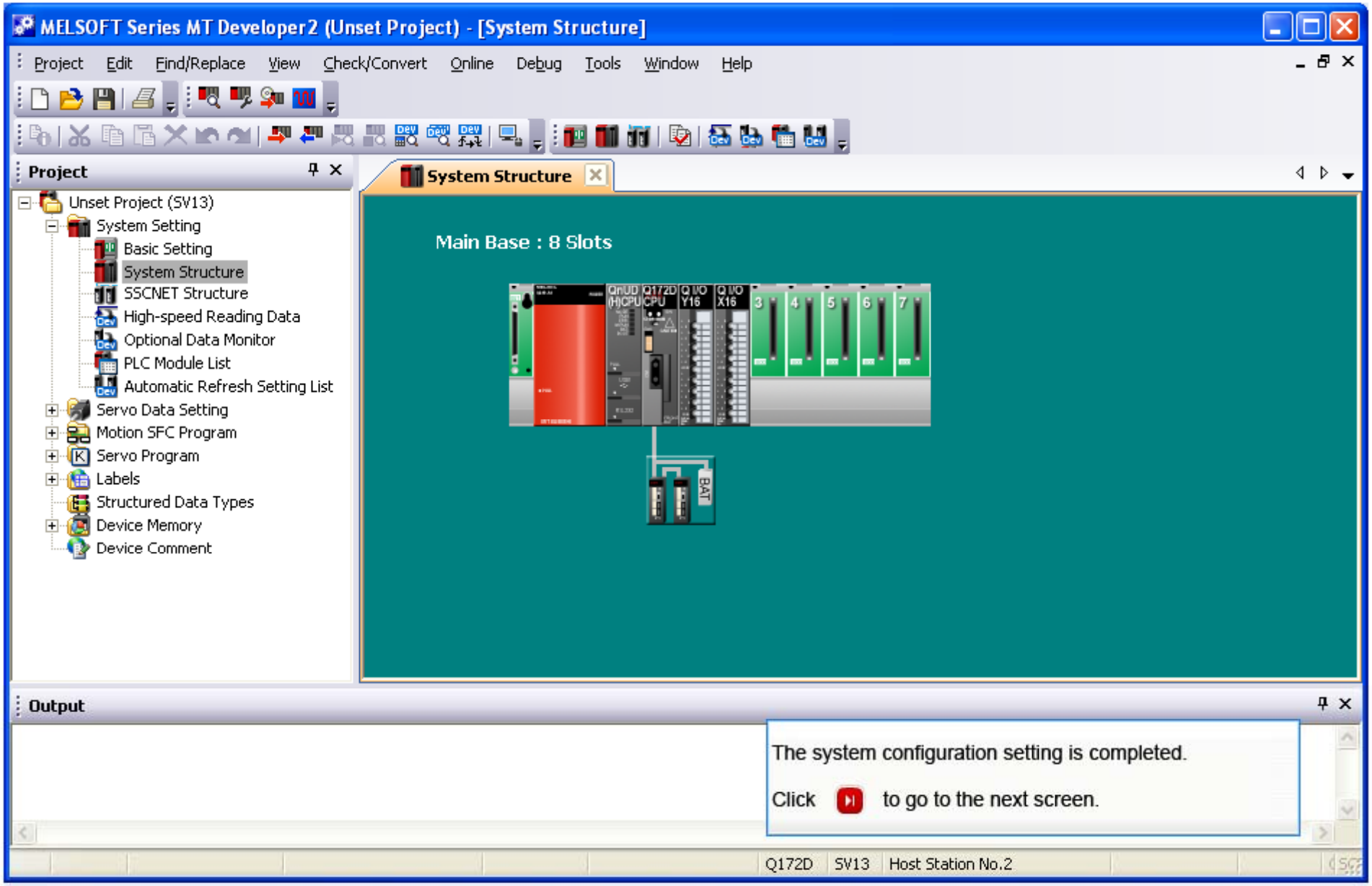
Next, set the module configuration used for the main base unit and the extension base unit.
Assign the motion module, I/O module, and other modules which are controlled by the motion CPU module to empty slots of the base unit.
In the sample system, assign an input module and an output module to the main base unit.

Slot No.	Module model name	I/O Type	Points	First I/O No.	High-Speed Read Setting	I/O Response Time Setting
Slot 1	QY40P	Output	16	0000	—	—
Slot 2	QX40	Input	16	0010	Not used	10ms



Let's set a system configuration in the next screen.

7.3.2 System configuration setting

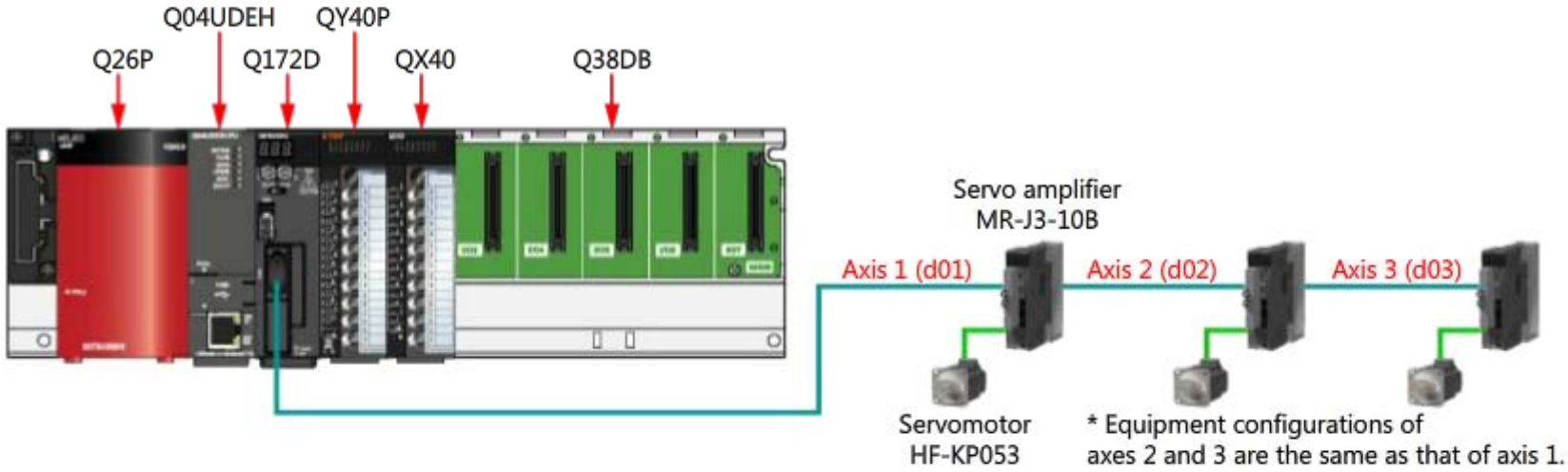


7.3.3 SSCNET structure setting

Next, set the servo amplifier configuration used for the system.
Assign a servo amplifier connected to the motion CPU module with SSCNET III cable according to each control axis number.

In the sample system, assign three servo amplifiers to three control axis numbers (d01 to d03).

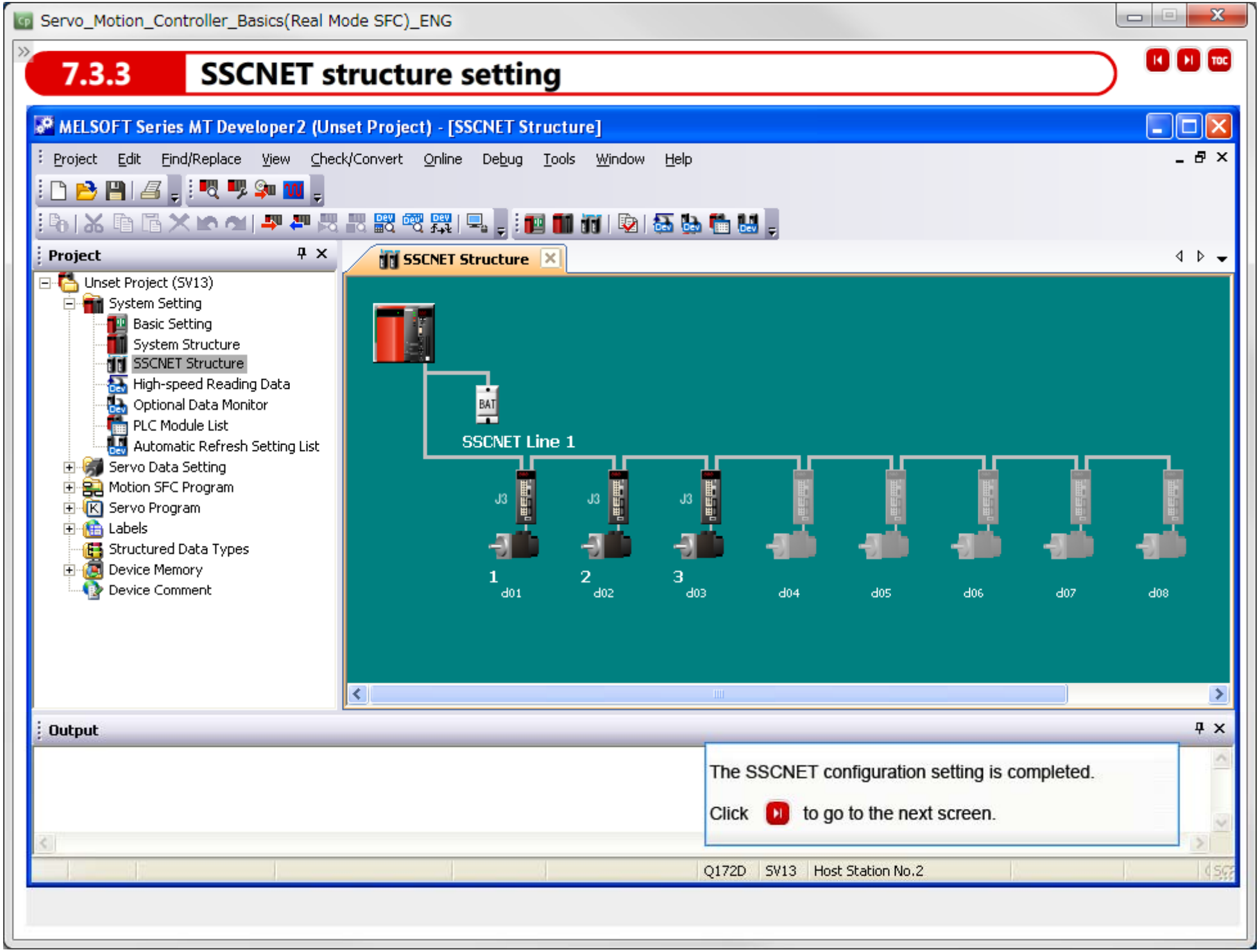
Control axis number at the servo amplifier side	Axis No.	Amplifier Type	External Signal Input Type	Allowable Travel during Power-Off
d01	1	MR-J3(W)-B	Amplifier Input Valid (Input Filter Setting: 3.5ms)	10 Revolution
d02	2			
d03	3			



Precautions

Axis No. set in the SSCNET III structure differs from the control axis number set using a rotary switch on a servo amplifier.
The axis No. set here is used to specify a control axis from the program.

Let's set the SSCNET III structure in the next screen.


Servo_Motion_Controller_Basics(Real Mode SFC)_ENG
7.3.3
SSCNET structure setting
MELSOFT Series MT Developer2 (Unset Project) - [SSCNET Structure]
Project
Edit
Find/Replace
View
Check/Convert
Online
Debug
Tools
Window
Help
Project
Unset Project (SV13)
System Setting
Basic Setting
System Structure
SSCNET Structure
High-speed Reading Data
Optional Data Monitor
PLC Module List
Automatic Refresh Setting List
Servo Data Setting
Motion SFC Program
Servo Program
Labels
Structured Data Types
Device Memory
Device Comment
SSCNET Structure
BAT
SSCNET Line 1
J3
J3
J3
1
2
3
d01
d02
d03
d04
d05
d06
d07
d08
Output
The SSCNET configuration setting is completed.
Click
to go to the next screen.
Q172D
SV13
Host Station No.2

7.4 Servo Data Setting

Next, set the servo data. Set the necessary data for positioning control to each axis which are set in the SSCNET configuration. Servo data are classified into the following three categories.

Classification	Description
Fixed Parameter	Refer to the section 7.4.1.
Home Position Return Data	Set the data necessary for executing the home position return. Home position return is a function that moves a machine to the home position and matches the home position addresses of the machine and the motion CPU module at the position.
JOG Operation Data	Set the data necessary for executing the JOG operation. JOG operation is a function that operates a servo motor manually in the forward or reverse rotation direction at the constant speed. It is used for the teaching or test operation when a system is when a system is installed.

7.4.1 Fixed parameter setting

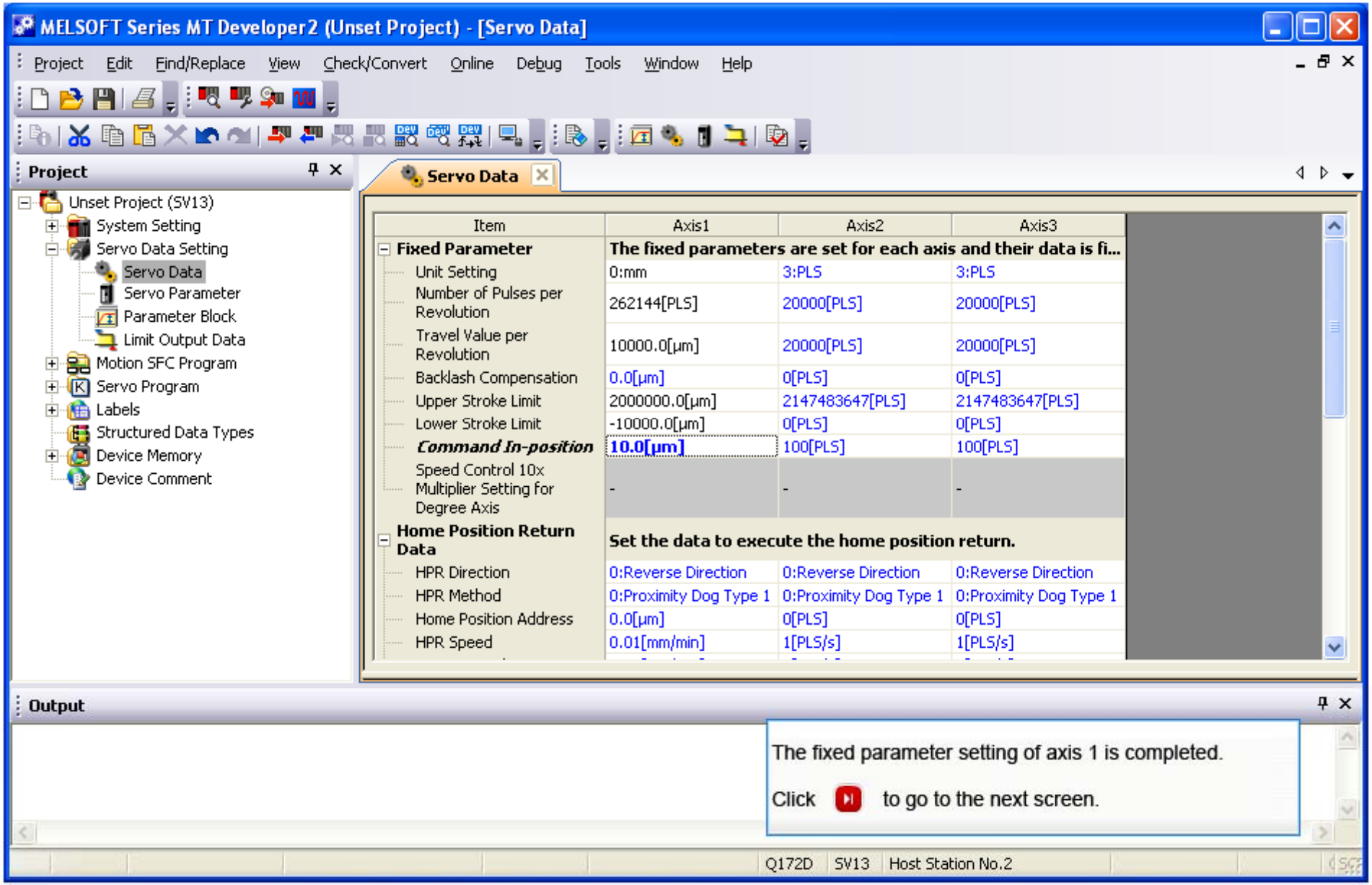
Set the characteristic value necessary for machine operation of the system. Set the data and movement range of the machine to convert
the command value of "address (travel value) and speed" which is called electric gear into the pulse unit.

In the sample system, set the following fixed parameters to axes 1 to 3.

Parameter item		Set value of axes 1 to 3	Remarks
Fixed Parameter	Unit Setting	0:mm	In the sample system, "mm" unit is used.
	Number of Pulses per Revolution	262144[PLS]	Usually, set the resolution value of the servo motor to be used.
	Travel Value per Revolution	10000.0[μm]	Ball screws (lead: 10mm) are used for the machine.
	Upper Stroke Limit	2000000.0[μm]	Set the movement range of the machine to prevent an overrun.
	Lower Stroke Limit	-10000.0[μm]	

Let's set the fixed parameters in the next screen.

7.4.1 Fixed parameter setting



7.4.2 Home position return data setting

Set the data necessary for executing the home position return. Home position return is a function that moves a machine to the home position and matches the home position addresses of the machine and the motion CPU module at the position.

In the sample system, set the following home position return data to axes 1 to 3.

Parameter item		Set value of axes 1 to 3	Remarks
Home Position Return Data	HPR Direction	0:Reverse Direction	—
	HPR Method	0:Proximity Dog Type 1	In the sample system, use " Proximity Dog Type 1 ".
	Home Position Address	0.0[μm]	—
	HPR Speed	20000.00[mm/min]	
	Creep Speed	100.00[mm/min]	
	Travel Value after Proximity Dog ON	—	
	Parameter Block Setting	1	For details, refer to Parameter Block Setting.
	HPR Retry Function	0:Invalid	—
	Dwell Time at the HPR Retry	—	
	Home Position Shift Amount	0.0[μm]	
	Speed Set at Home Position Shift	0:HPR Speed	
	Torque Limit Value at Creep Speed	—	
	Operation for HPR Incompletion	1:Not Execute Servo Program	

Let's set the home position return data in the next screen.

7.4.2 Home position return data setting

MELSOFT Series MT Developer2 (Unset Project) - [Servo Data]

Project Edit Find/Replace View Check/Convert Online Debug Tools Window Help

Project

- Unset Project (SV13)
  - System Setting
  - Servo Data Setting
    - Servo Data
    - Servo Parameter
    - Parameter Block
    - Limit Output Data
  - Motion SFC Program
  - Servo Program
  - Labels
  - Structured Data Types
  - Device Memory
  - Device Comment

Servo Data

Item	Axis1	Axis2	Axis3
Creep Speed	100.00[mm/min]	1[PLS/s]	1[PLS/s]
Travel Value after Proximity Dog ON	-	-	-
Parameter Block Setting	1	1	1
HPR Retry Function	0:Invalid	0:Invalid	0:Invalid
Dwell Time at the HPR Retry	-	-	-
Home Position Shift Amount	0.0[μm]	0[PLS]	0[PLS]
Speed Set at Home Position Shift	0:HPR Speed	0:HPR Speed	0:HPR Speed
Torque Limit Value at Creep Speed	-	-	-
Operation for HPR Incompletion	**1:Not Execute Servo Program**	1:Not Execute Servo Program	1:Not Execute Servo Program
Pulse Conversion Module Home Position Return Request Setting	-	-	-
Standby Time after Pulse Conversion Module Clear	-	-	-

Output

The home position return data setting of Axis 1 is completed.

Click to go to the next screen.

Q172D SV13 Host Station No.2

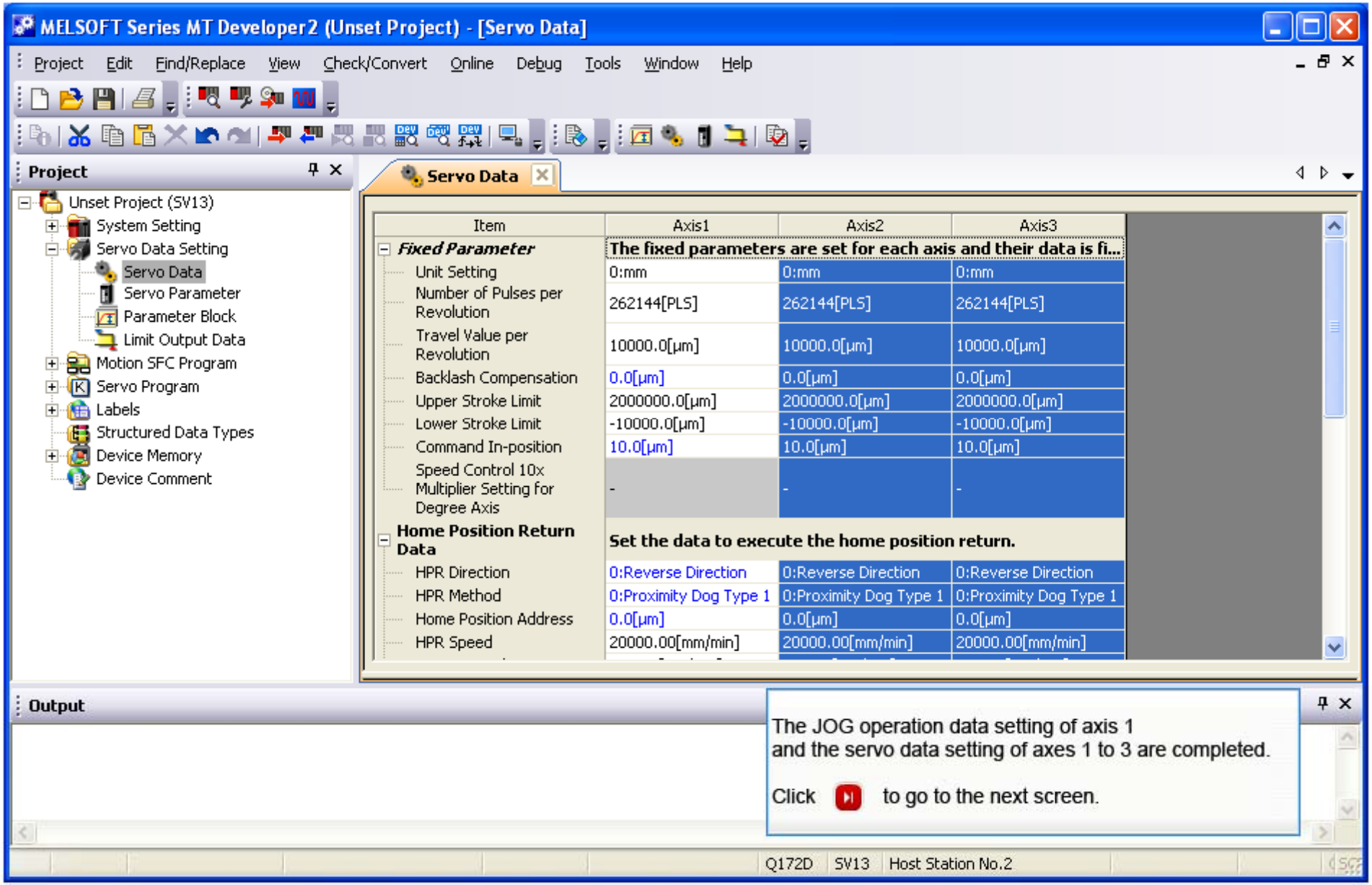
7.4.3 JOG operation data setting

Set the data necessary for executing the JOG operation.
JOG operation is a function that operates a servomotor manually in the forward or reverse rotation direction at the constant speed.
It is used for the teaching or test operation when a system is constructed.
In the sample system, set the following JOG operation data to axes 1 to 3.

Parameter item		Set value of axes 1 to 3	Remarks
JOG Operation Data	JOG Speed Limit Value	15000.00[mm/min]	—
	Parameter Block Setting	2	For details, refer to Parameter Block Setting.

Let's set the home position return data in the next screen.

7.4.3 JOG operation data setting



The JOG operation data setting of axis 1
and the servo data setting of axes 1 to 3 are completed.

Click ▶| to go to the next screen.

7.5 Servo Parameter Setting

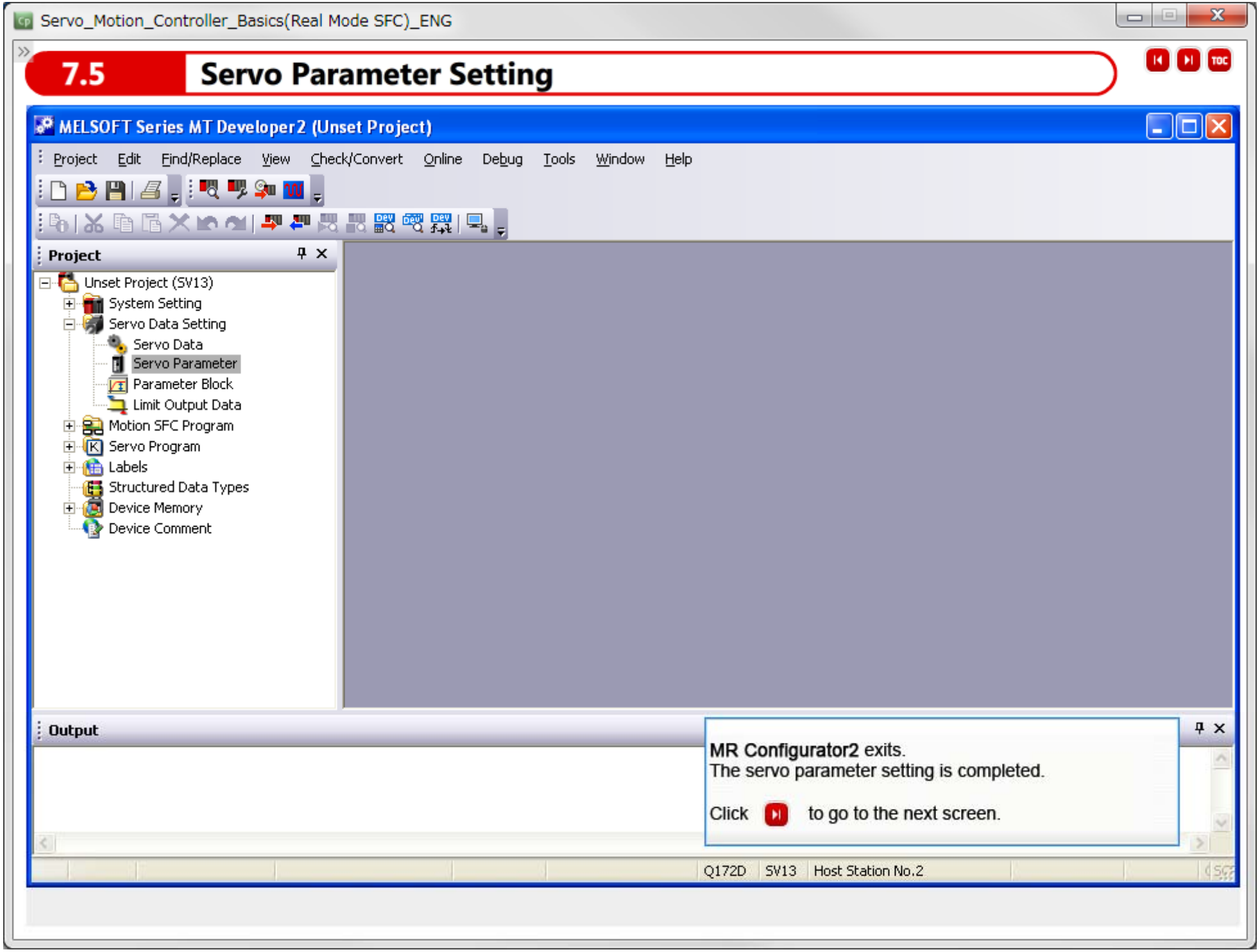
Next, set the servo specific parameters to each axis.
For the setting of servo parameter, servo setup software MELSOFT MR Configurator2 is needed separately.
Download and install MR Configurator2 before setting parameters.

In the sample system, set the following servo parameters to axes 1 to 3.

Parameter item	Set value
Rotation direction selection	CCW dir. during fwd. pls input, CW dir. during rev pls. input
Servo forced stop selection	Invalid (Not use forced stop input (EM1))
Absolute position detection system	Used in incremental system
Home position set condition selection	Z-phase must not be passed.
In-position range	100 [PLS]

* For parameters which are not used in this course, use the default values.

* Let's set the servo parameter in the next screen.


Servo_Motion_Controller_Basics(Real Mode SFC)_ENG
7.5
Servo Parameter Setting
MELSOFT Series MT Developer2 (Unset Project)
Project
Edit
Find/Replace
View
Check/Convert
Online
Debug
Tools
Window
Help
Project
Unset Project (SV13)
System Setting
Servo Data Setting
Servo Data
Servo Parameter
Parameter Block
Limit Output Data
Motion SFC Program
Servo Program
Labels
Structured Data Types
Device Memory
Device Comment
Output
MR Configurator2 exits.
The servo parameter setting is completed.
Click to go to the next screen.
Q172D
SV13
Host Station No.2

7.6 Parameter Block Setting

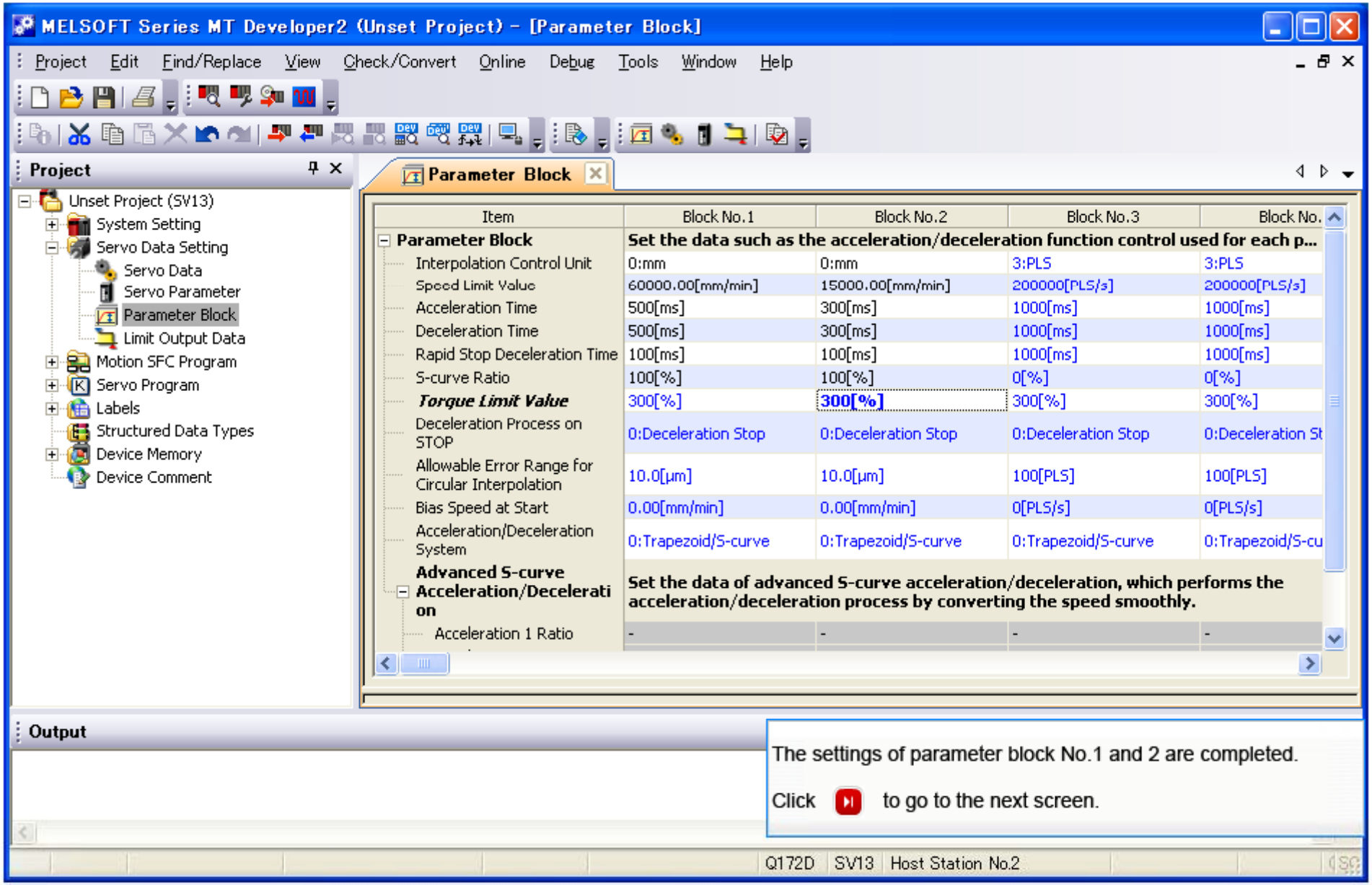
Set acceleration/ deceleration parameters for each control pattern.
Up to 64 acceleration/ deceleration patterns can be created.
Set an arbitrary parameter block No. to each control pattern in positioning control.

In the sample system, set the following parameters to the blocks No.1 and No. 2.

Parameter item	Block No. 1	Block No. 2
Control pattern	For positioning control and home position return	For JOG operation
Interpolation Control Unit	0:mm	0:mm
Speed Limit Value	60000.00[mm/min]	15000.00[mm/min]
Acceleration Time	500[ms]	300[ms]
Deceleration Time	500[ms]	300[ms]
Rapid Stop Deceleration Time	100[ms]	100[ms]
S-curve Ratio	100[%]	100[%]
Torque Limit Value	300[%]	300[%]
Deceleration Process on STOP	0:Deceleration Stop	0:Deceleration Stop
Allowable Error Range for Circular Interpolation	10.0[µm]	10.0[µm]
Acceleration/ Deceleration System	0:Trapezoid/ S-curve	0:Trapezoid/ S-curve

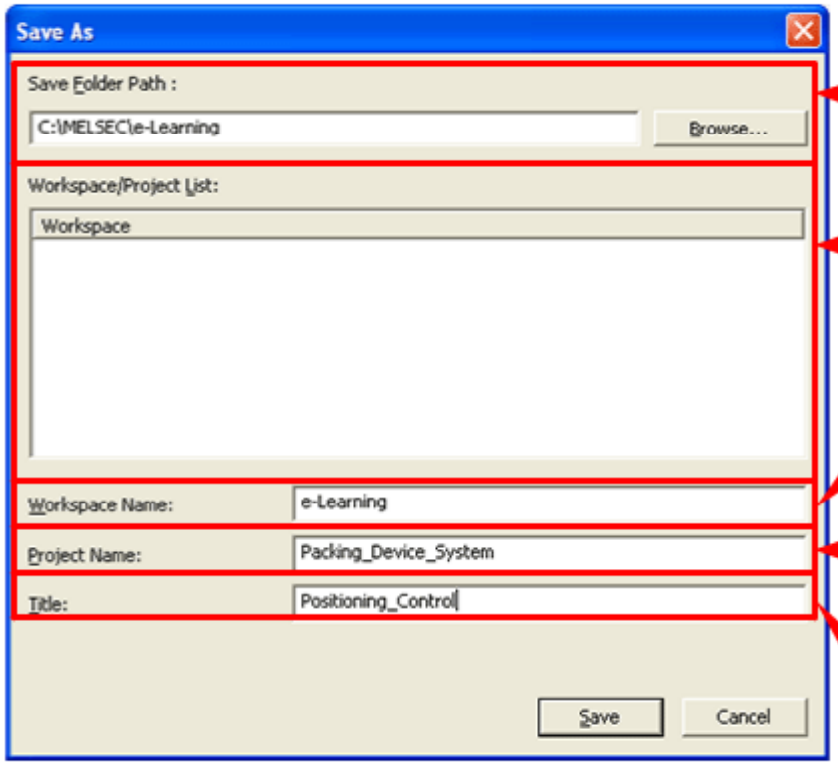
Let's set the parameter block setting in the next screen.

7.6 Parameter Block Setting



7.7 Saving a Project

Save a project including parameters after the parameter setting.
If you exit MT Developer2 without saving a project, the set parameters will be discarded.
If you save a new project, set the following project information.
It is recommended to give a name which you can easily recognize the contents of the project (control contents, system name, etc.).



Save Folder Path * Compulsory
Specify a folder to create a workspace.

Workspace/Project List
If one or more workspaces exist in the save folder path, they are displayed in list.
Double-clicking a workspace name displays a list of projects.

Workspace Name * Compulsory
Specify a workspace name. (up to 128 characters)

Project Name * Compulsory
Specify a project name. (up to 128 characters)

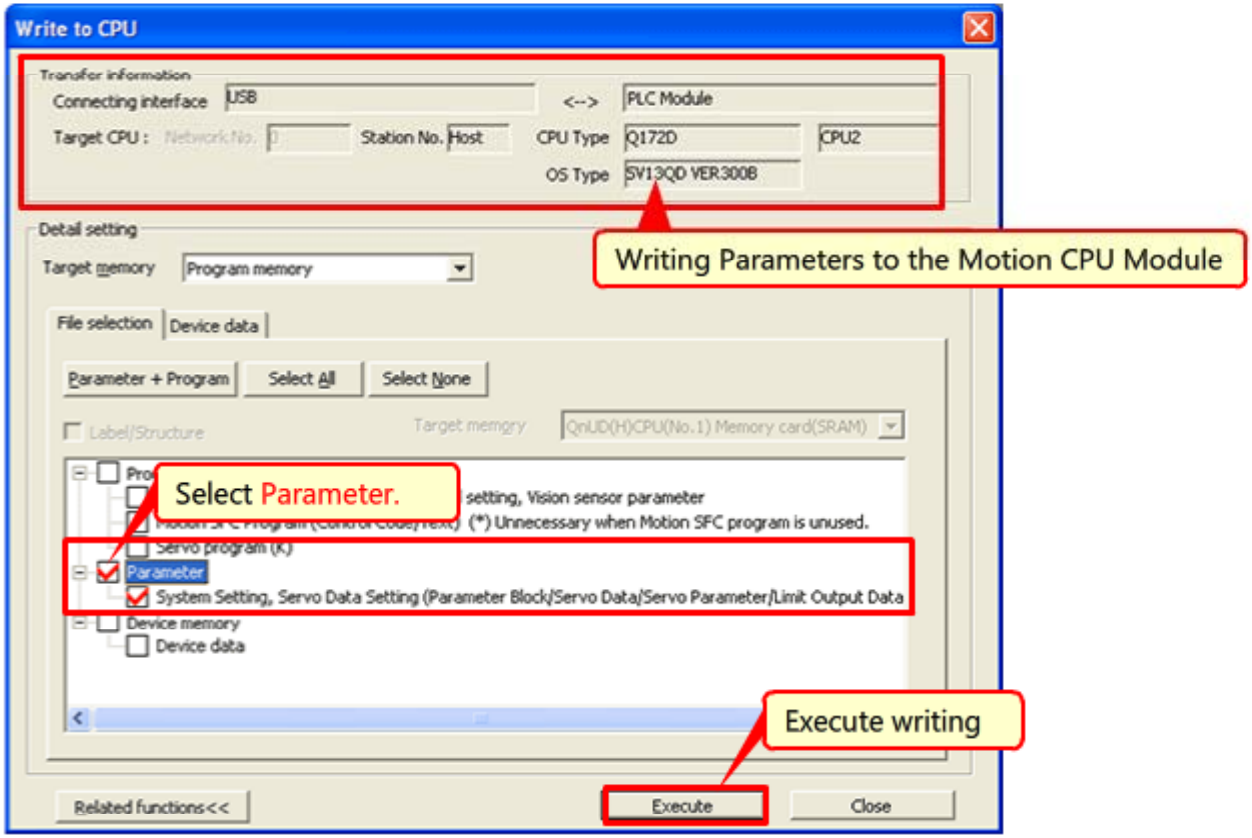
Title
Specify a title. (up to 128 characters)
Use here when you want to give a name which exceeds 128 characters. (Inputting title is not necessary.)

7.8 Writing Parameters to the Motion CPU Module

After saving the project, write parameters to the motion CPU module.
Before writing, check the following.

The power supplies of the motion controller and the servo amplifier are on.
The RUN/STOP switch of the motion CPU module is switched to STOP.
The personal computer and the PLC CPU module are connected correctly.

Check parameters in the Write to CPU screen and perform writing.



7.9 Summary

The following lists the contents you learned in Chapter 7.
The following points are very important, so please check them again.

Transfer setup	・Before setting parameters, enable communications between the personal computer and the motion CPU module. ・Since the communication target motion CPU module is mounted to the CPU slot 2 of the base unit, select PLC No.2 in the transfer setup.
Project	・A project is a unit that is used to control various parameters and programs by MT Developer2. ・Set an operating system type and the model name of a motion CPU module to be used for creating a project.
Basic system setting	The basic system setting includes such as base unit, multiple CPU, etc.
System configuration	Set the module configuration used for the main base unit and the extension base unit. Assign the motion module, I/O module, and other modules which are controlled by the motion CPU module to empty slots of the base unit.
SSCNET configuration	・Set the servo amplifier configuration used for the system. Assign a servo amplifier connected to the motion CPU module with SSCNET III cable according to each control axis number. ・Axis No. set in the SSCNET configuration differs from the control axis number set using a rotary switch on a servo amplifier. The axis No. is used to specify a control axis from the program.
Fixed parameter	Set the characteristic value necessary for machine operation of the system. Set the data and movement range of the machine to convert the command value of "address (travel value) and speed" which is called electric gear into the pulse unit.
Home position return data	Set the data necessary for executing the home position return. Home position return is a function that moves the machine to the home position and matches the home positions between the machine and the motion CPU module at the position.
JOG operation data	Set the data necessary for executing the JOG operation. JOG operation is a function that operates a servo motor manually in the forward or reverse rotation direction at the constant speed. It is used for the teaching or test operation when a system is constructed.
Servo parameter	Set the servo specific parameters to each axis. For the setting of servo parameter, servo setup software MELSOFT MR Configurator2 is needed separately.
Parameter block	Set the acceleration/deceleration process for each control pattern. Up to 64 acceleration/deceleration patterns can be created. Specify an arbitrary parameter block No. to each control pattern in positioning control.
Saving a project	・Save a project including parameters after the parameter setting.

7.9 Summary

Saving a project	・Save a project including parameters after the parameter setting. If MELSOFT MT Developer2 exits without saving a project, the set parameter contents will be discarded. ・Give a name which the contents of the project (control contents, system name, etc.) are easily recognized.
Writing parameters	Write parameters to the motion CPU module. Before writing, check the following. ・The power supplies of the motion controller and the servo amplifier are on. ・The RUN/STOP switch of the motion CPU module is STOP. ・A personal computer and the PLC CPU module are connected correctly.

Chapter 8 OPERATION CHECK

In Chapter 8, you will learn how to check the operation of a servo motor and how to perform the home position return. When turning on a servo amplifier and a servomotor for the first time, before installing the servomotor to a machine, make sure to check the operation to prevent an accident such as machine damage due to malfunction such as miswiring or incorrect parameter settings.

SELECTING AND INSTALLING THE OPERATING SYSTEM SOFTWARE ····· "Chapter 6"

↓

SYSTEM SETTING ····· "Chapter 7"

↓

OPERATION CHECK ····· "Chapter 8"

↓

PROGRAM DESIGN ····· "Chapter 9"(BASICS OF MOTION SFC: Chapter 10)

↓

PROGRAMMING ····· "Chapter 11"

↓

OPERATION

Learning procedure of Chapter 8

8.1 Checking the Operation of a Servo motor

8.2 Connection of a Servo motor with a Machine

8.3 Performing the Home Position Return

8.1 Checking the Operation of a Servo motor

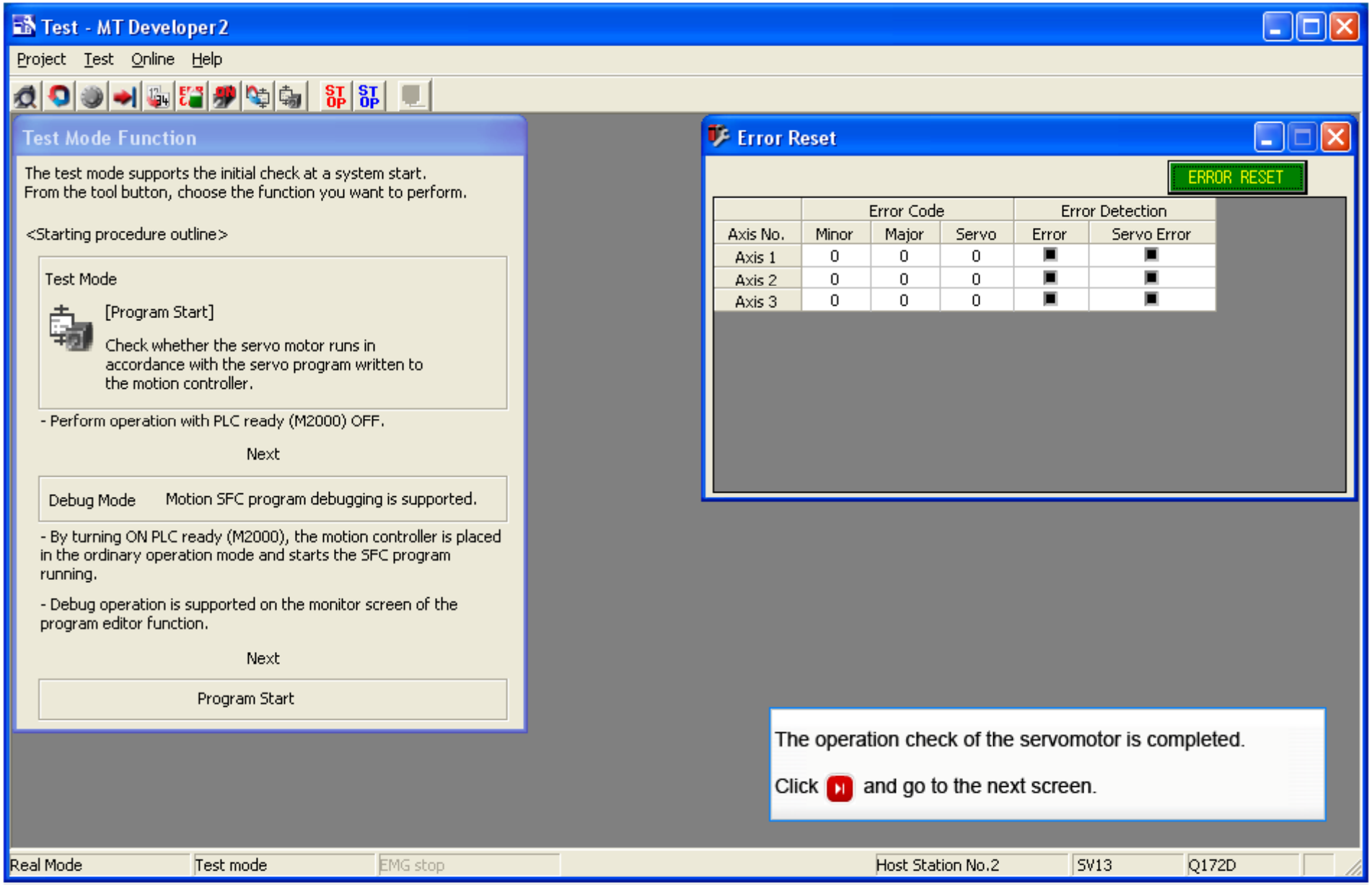
Check the status of a servo amplifier (for error), rotating direction of a servo motor, operation of the upper and lower stroke limits, and stopping accuracy of the home position return, by using the test function of MT Developer2.

The following shows the list of the test functions used in this course.

Name	Description
Servo on and off	Outputs the servo on or servo off command to all or desired axes of servo motors.
Initial check	Displays the status of a servo amplifier. If an error exists, the error code and the error name can be checked.
Upper and lower LS check	Performs JOG operation in a forward or reverse rotation to check if the upper or lower stroke limit operates normally.
JOG operation	Performs JOG operation of a connected motor. Before performing JOG operation, make sure to set the JOG operation data and set the data in the parameter blocks to be used.
Home position return test	Performs the home position return to check for an error between a stop position and a machine home position.

Let's check operation using the test function in the next screen.

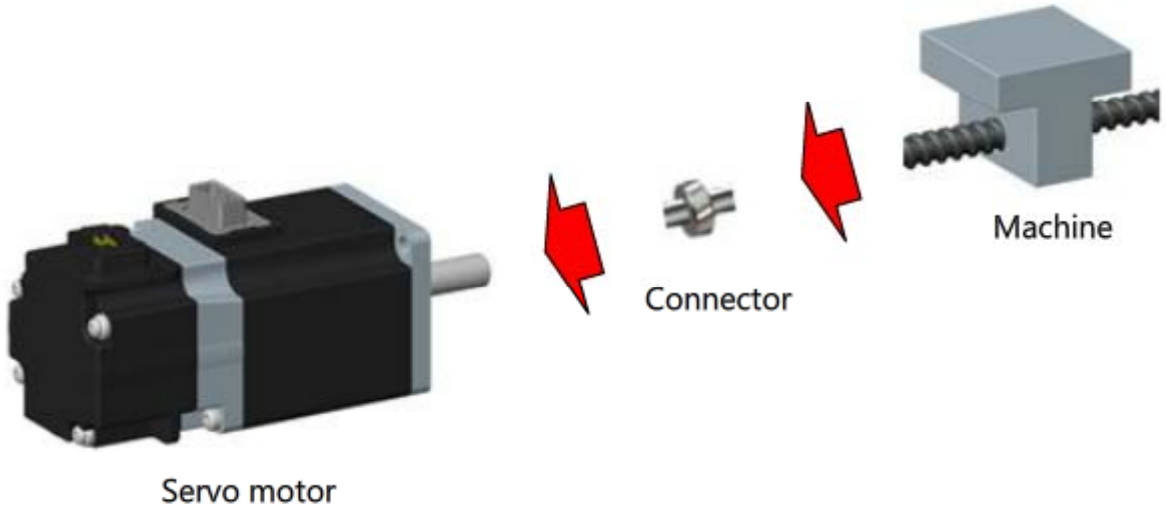
8.1 Checking the Operation of a Servo motor



8.2 Connection of a Servo motor with a Machine

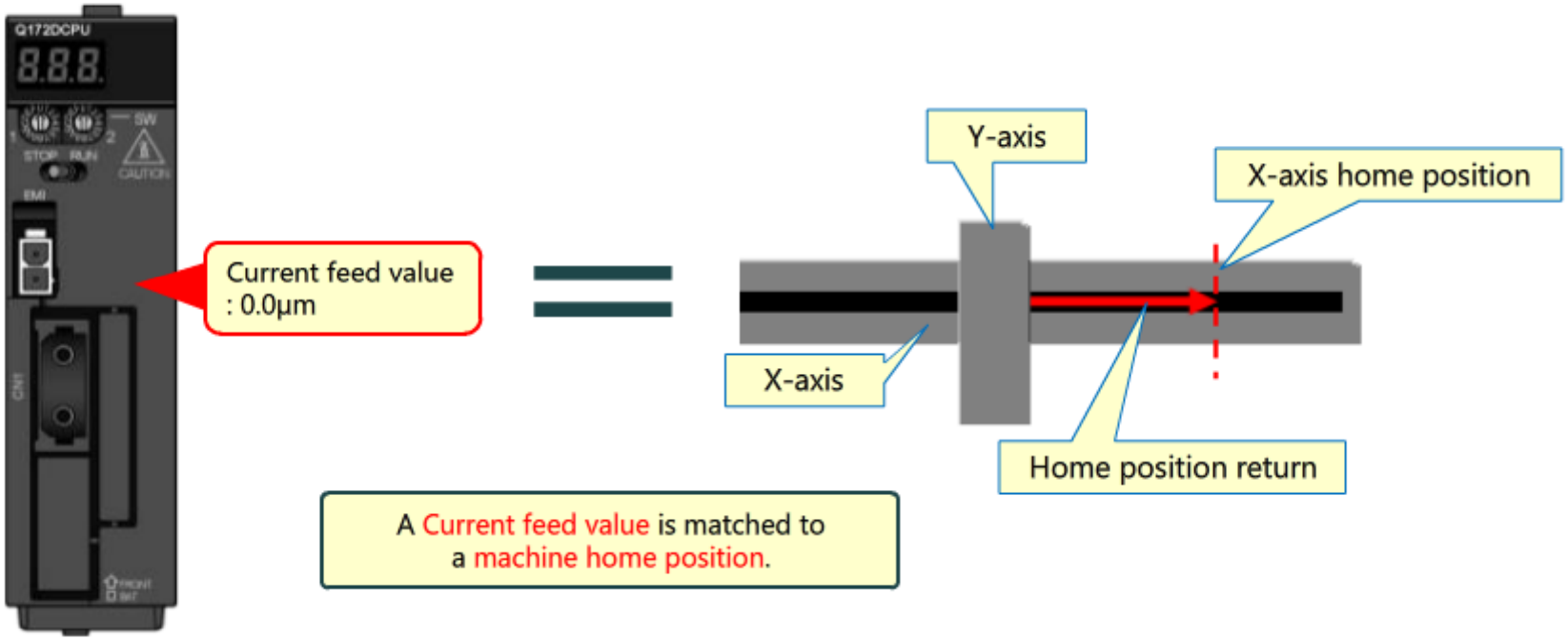
Next, install a machine to the rotational axis of a servo motor.

Before installation, check the operation of a servo motor without a machine in order to prevent machine damage due to malfunction of a servo system.
After completing the installation of a machine, check the normal operation of both a servo motor and a machine by using JOG operation again.



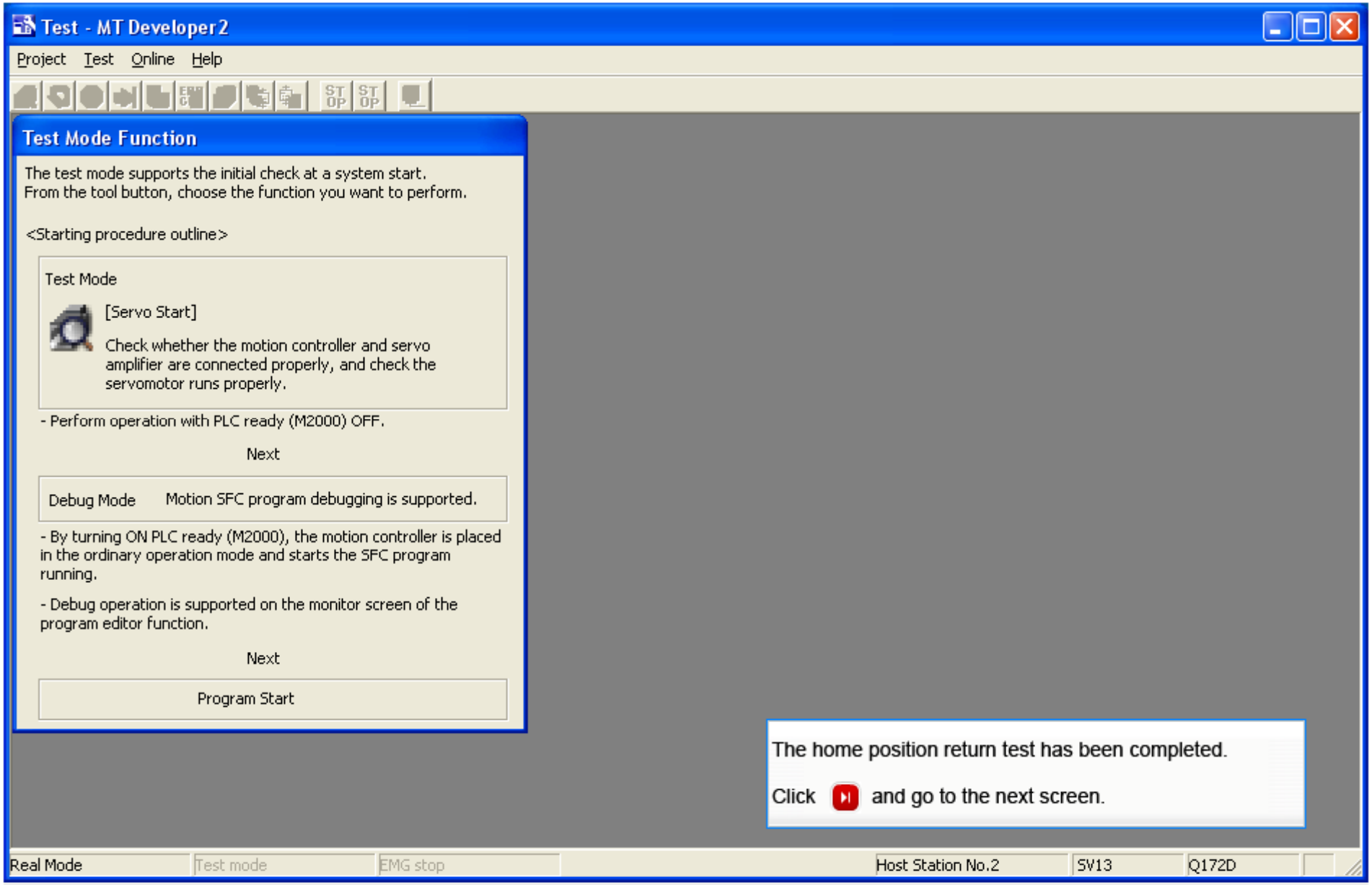
8.3 Performing the Home Position Return

After connecting a servo motor with a machine, check the normal operation of the home position return.
The home position return is an operation to match a home position stored in a motion CPU module to a machine home position.
The mismatched home positions cause an error of a stop position.
To prevent the error, perform the home position return test to confirm there is no error between a stop position and a machine home position.



Let's check the operation using the home position return test function in the next screen.

8.3 Performing the Home Position Return



8.4 Summary

The following lists the contents you learned in Chapter 8.
The following points are very important, so please check them again.

Operation check of servo motor	Check the status of a servo amplifier, rotating direction of a servo motor, operation of the upper and lower stroke limits by using the test function of MT Developer2..
Connection of a Servo motor with a Machine	・Before installation, check the operation of a servomotor without a machine in order to prevent machine damage due to malfunction of a servo system. ・After completing the installation of a machine, check the normal operation of both a servo motor and a machine by using JOG operation again.
Operation check of the home position return	After connecting a servo motor with a machine, check the normal operation of the home position return. After the home position return is executed in the home position return test, confirm there is no error between a stop position and a machine home position.

Chapter 9 PROGRAM DESIGN

In Chapter 9, you will learn how to design a program necessary for motion control.

SELECTING AND INSTALLING THE OPERATING SYSTEM SOFTWARE ····· "Chapter 6"

↓

SYSTEM SETTING ····· "Chapter 7"

↓

OPERATION CHECK ····· "Chapter 8"

↓

PROGRAM DESIGN ····· "Chapter 9"(BASICS OF MOTION SFC: Chapter 10)

↓

PROGRAMMING ····· "Chapter 11"

↓

OPERATION

Learning procedure of Chapter 9

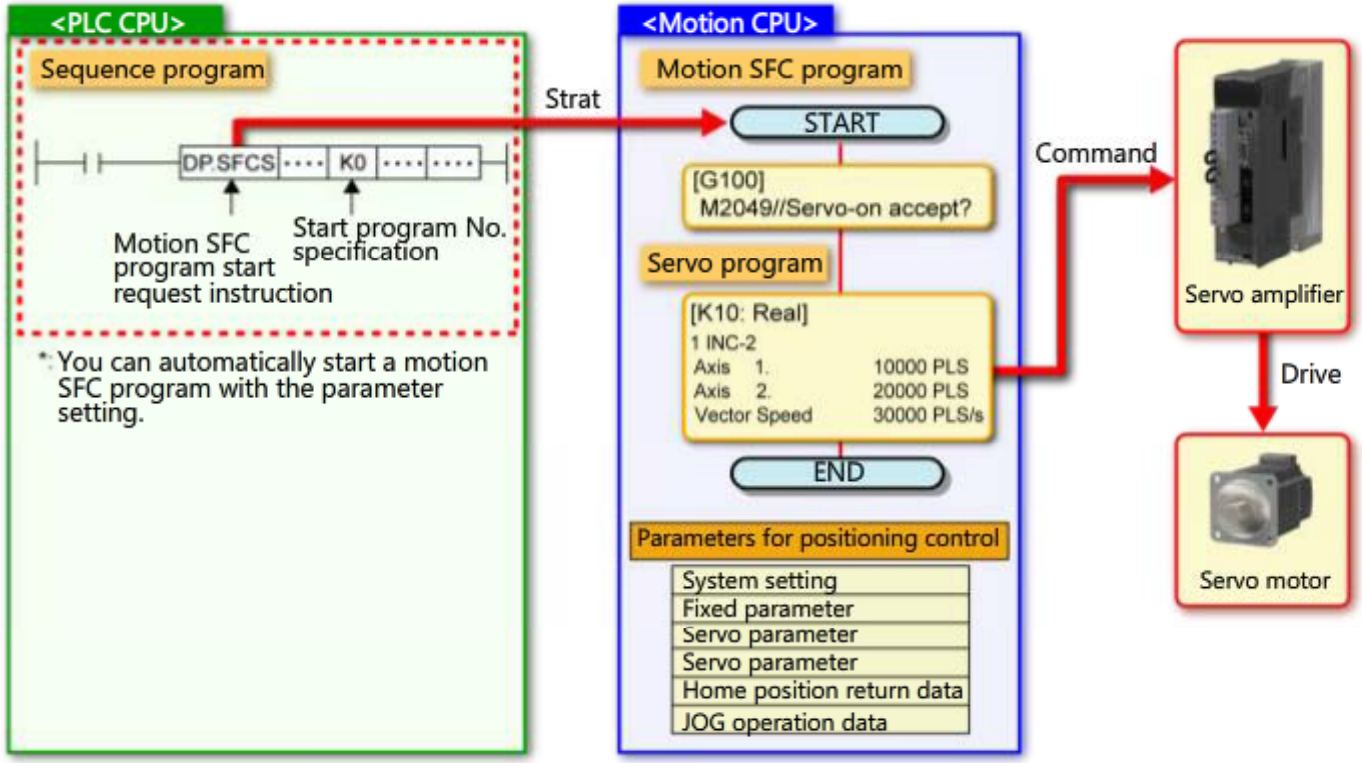
9.1 Programming Language for Motion Control
9.2 Flowcharting the Sequence of Control
9.3 Creating the Correspondence Table of I/O Devices and Device Nos.
9.4 Designing a Servo Program
  9.4.1 Servo instruction
  9.4.2 Positioning data
9.5 Creating a Servo Program

9.1 Programming Language for Motion Control

The following three types of the programming language enable motion control.

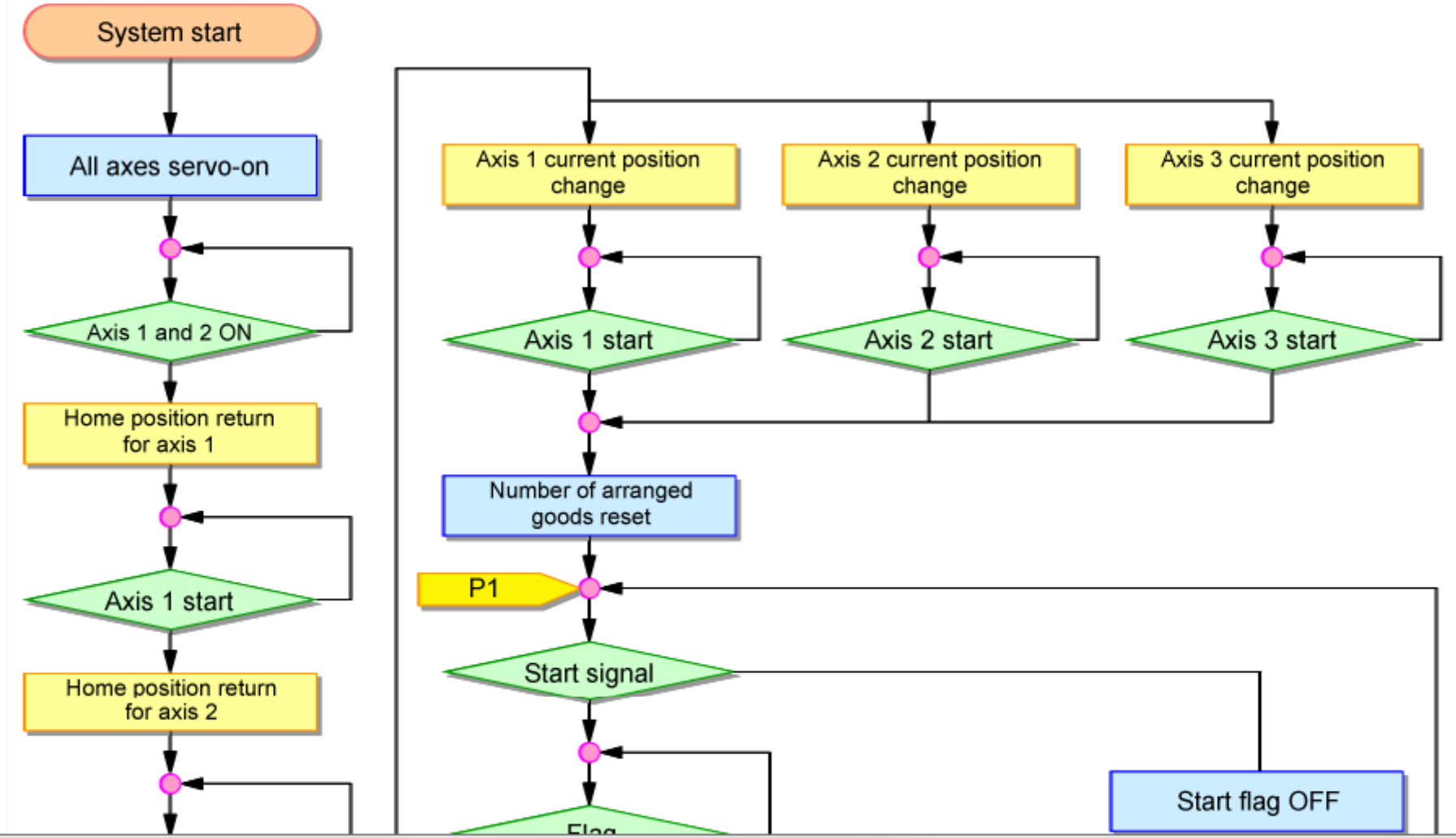
Programming language	Description
Sequence program	A motion SFC program is started with the motion dedicated seauence instruction**"D(P).SFCS"**. * When "Auto." is set to "Yes" in the parameter setting, a sequence program for start is unnecessary. * A specified servo program can be directly started with the motion dedicated sequence instruction **"D(P).SVST"**
Motion SFC program	The sequence of motion control is written in a flowchart-like format. In positioning control, a servo program is executed with the motion control step.
Servo program	The pattern of positioning control is written with servo instructions.

The following figure shows the relationship among the sequence program, motion SFC program, and servo program.

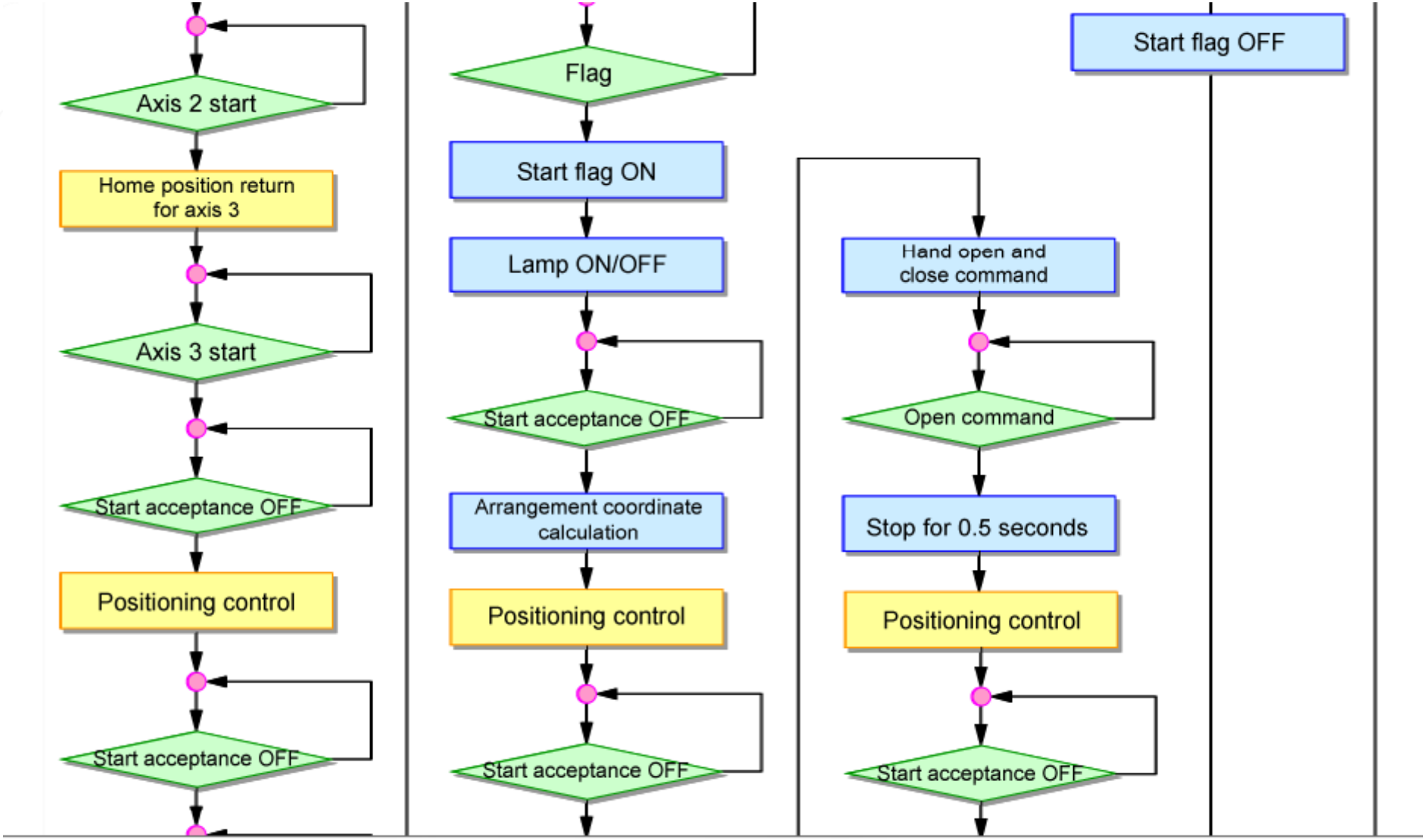


9.2 Flowcharting the Sequence of Control

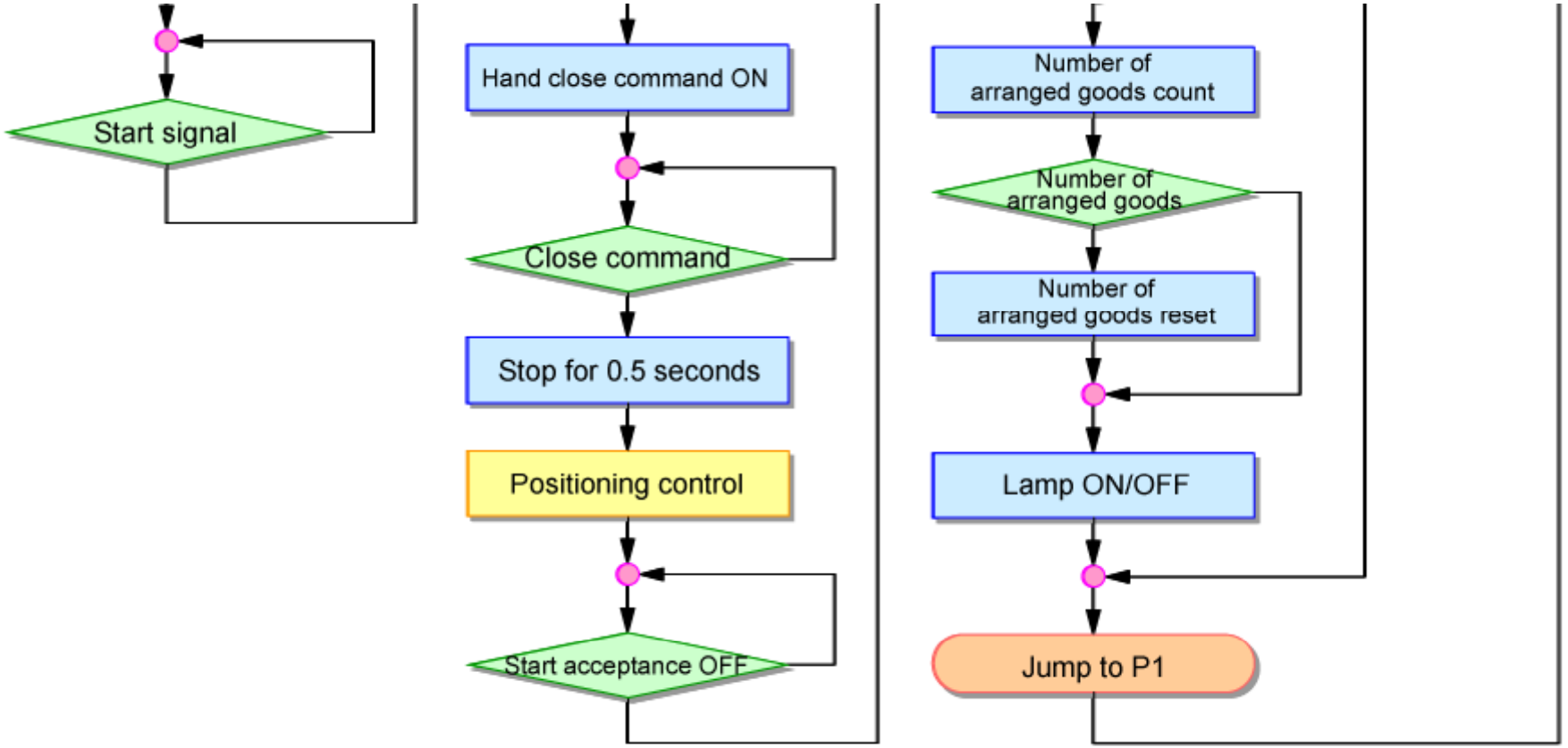
Motion SFC language is a programming language similar to a flowchart.
Expressing the sequence of control in a flowchart make design a motion SFC program easier.
The following shows the flowchart of the control of the sample system.
Put mouse pointer over the flowchart to display the details of each control.



9.2 Flowcharting the Sequence of Control



9.2 Flowcharting the Sequence of Control



9.3 Creating the Correspondence Table of I/O Devices and Device Nos.

Next, create the correspondence table of I/O devices and device Nos. to be used in the sample system.
Creating a correspondence table reduces programming glitches and streamlines your programming.

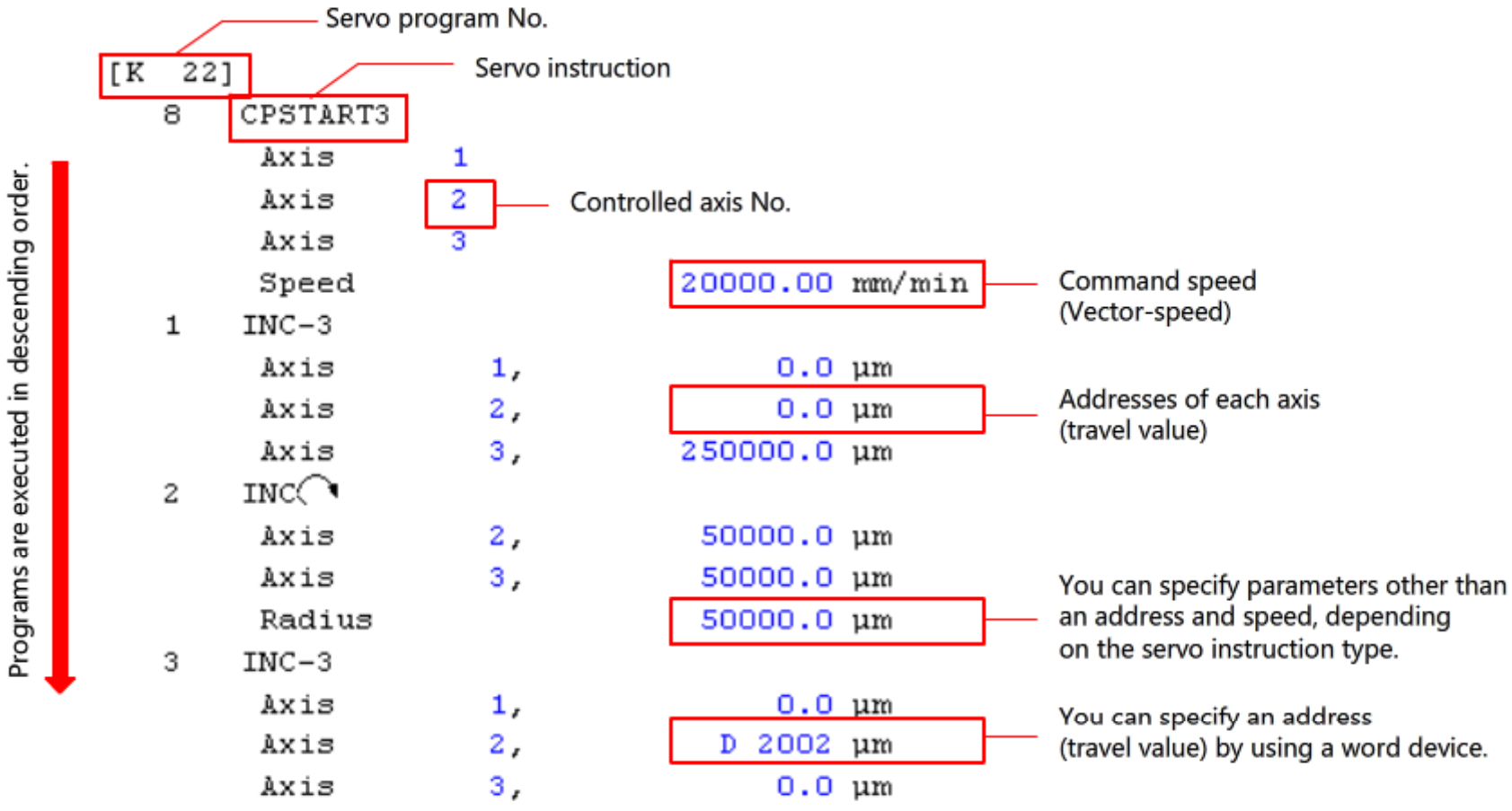
As an example, the following table shows the correspondence of I/O devices and device Nos in the sample system.

I/O device name	Device No.	Input or output	Type	Data type	Range	Initial value	Description
Start button	PX12	Input	Bit	—	—	OFF	A push button switch to start the system
Hand open command	PY0	Output	Bit	—	—	OFF	Output for controlling opening and closing the hand part of the device
Hand close command	PY1	Output	Bit				
Operation indicator lamp	PY2	Output	Bit	—	—	OFF	The lamp illuminates during system operation.
Stop indicator lamp	PY3	Output	Bit	—	—	OFF	The lamp illuminates during system stop.
Devices used in program	D2000	—	Word	16-bit integer	0～500	0	The travel value of the X-axis (axis 1) of the device is stored.
	D2002	—	Word	16-bit integer	0～1100	0	The travel value of the Y-axis (axis 2) of the device is stored.
	D2100	—	Word	16-bit integer	0～6	0	The number of goods arranged on the pallet is stored.
	M7100	—	Bit	—	—	OFF	Bit data to be output to the hand open command (PY0) is stored.
	M7101	—	Bit	—	—	OFF	Bit data to be output to the hand close command (PY1) is stored.
	M8001	—	Bit	—	—	OFF	Bit data input from the start button (PX12) is stored.

9.4 Designing a Servo Program

Next, design a servo program.
A servo program is a programmed pattern of positioning control.
The program consists of servo instructions, axis Nos., addresses (travel value), command speed, and acceleration pattern, etc.

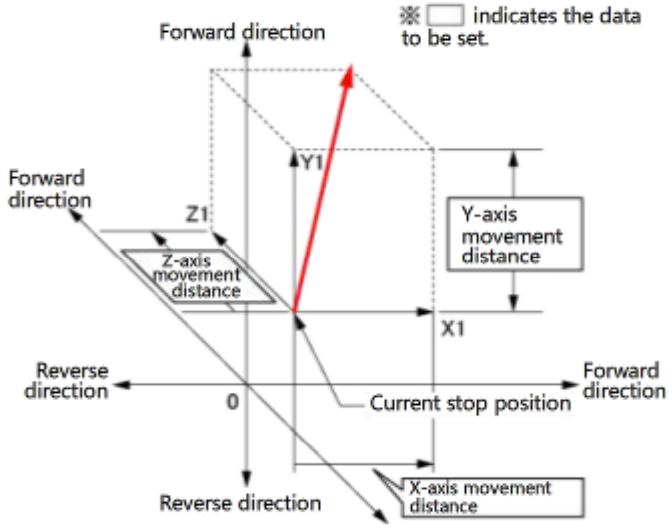
Register the pattern of positioning control as a servo program beforehand.
In positioning control with a Motion SFC program, a specified servo program No. is executed according to the control pattern.



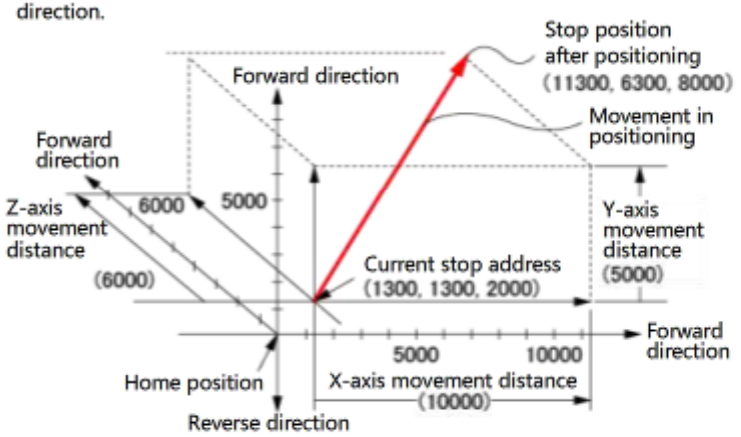
9.4.1 Servo instruction

Next, you will learn servo instructions used in a servo program.
Many servo instructions, such as simple linear positioning with one axis and linear and circular interpolations with two or more axes, are prepared. In this course, the six servo instructions used in the sample system is explained.

Symbol	Instruction name	Description
INC-3	Incremental 3-axes linear interpolation	• For three specified axes, 3-axes linear interpolation control is performed in the range of travel value specified for each axis, starting from a current stop position. • The movement directions of each axis depend on the sign (+ or -) of the travel value specified for each axis. ・When the travel value is positive: Positioning in forward direction (The address increases.) ・When the travel value is negative: Positioning in reverse direction (The address decreases.)



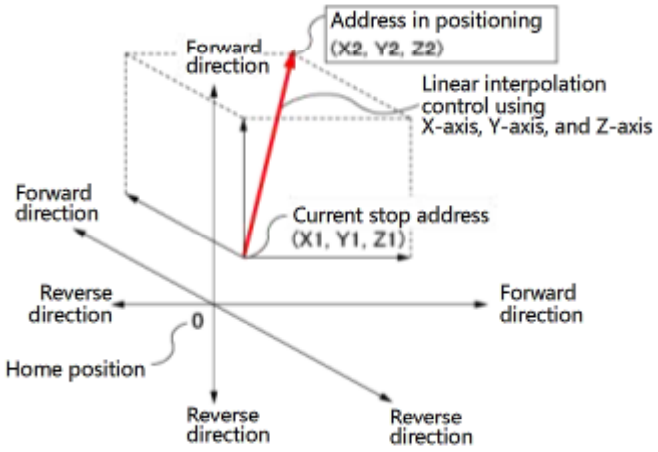
Example
Positioning is performed as follows when the movement distances are 10000 for X-axis direction, 5000 for Y-axis direction, and 6000 for Z-axis direction.



9.4.1 Servo instruction

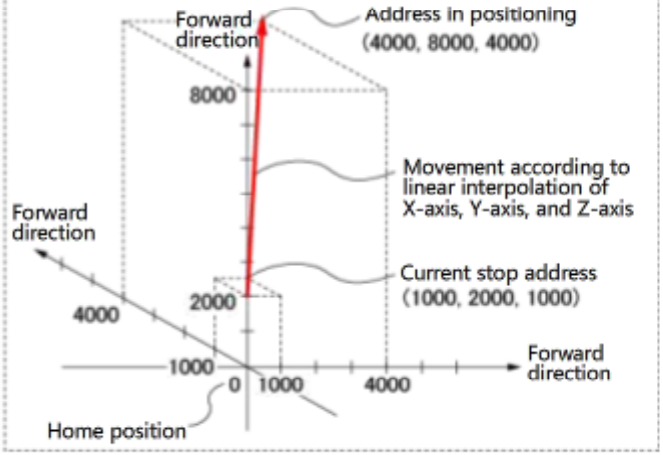
Symbol	Instruction name	Description
ABS-3	Absolute 3-axes linear interpolation	• By the absolute method, 3-axes linear interpolation control is performed for specified three axes in the range from a current stop position (X1, Y1, Z1) to a specified position (X2, Y2, Z2). • The movement directions of each axis depend on their stop addresses and specified addresses.

※ ☐ indicates the data to be set.



Example

Positioning is performed as follows when the current stop address is (1000, 2000, 1000) and the positioning address is (4000, 8000, 4000).

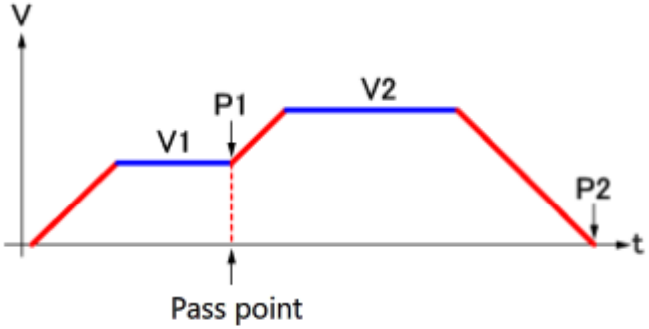


9.4.1 Servo instruction

Symbol	Instruction name	Description
CPSTART3 (CPEND)	Constant-speed control	• Constant-speed control of 3-axes linear interpolation control or 2-axis circular interpolation control is performed. Once started, constant-speed control is performed for a specified axis, with passing a preset pass point, until Constant-speed control end (CPEND). In interpolation control, a vector speed is used as a command speed. • As shown in the figures below, the control for changing to the specified speed at the pass point differs depending on ON/ OFF of "CP completion point specification flag (M2040)" at the time when the constant-speed control is started.

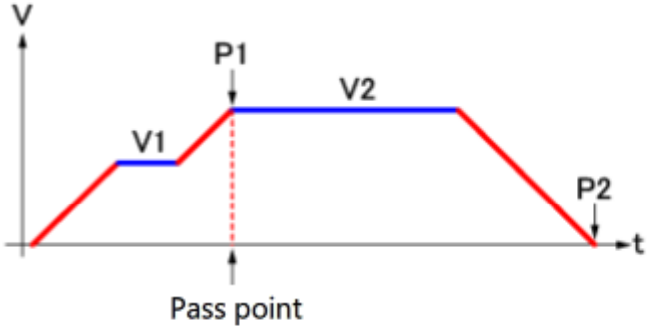
CP completion point specification flag: OFF
.....Specifies the point where the speed change is started.

The speed is started to change at the pass point.



CP completion point specification flag: ON
.....Specifies the point where the speed change is completed.

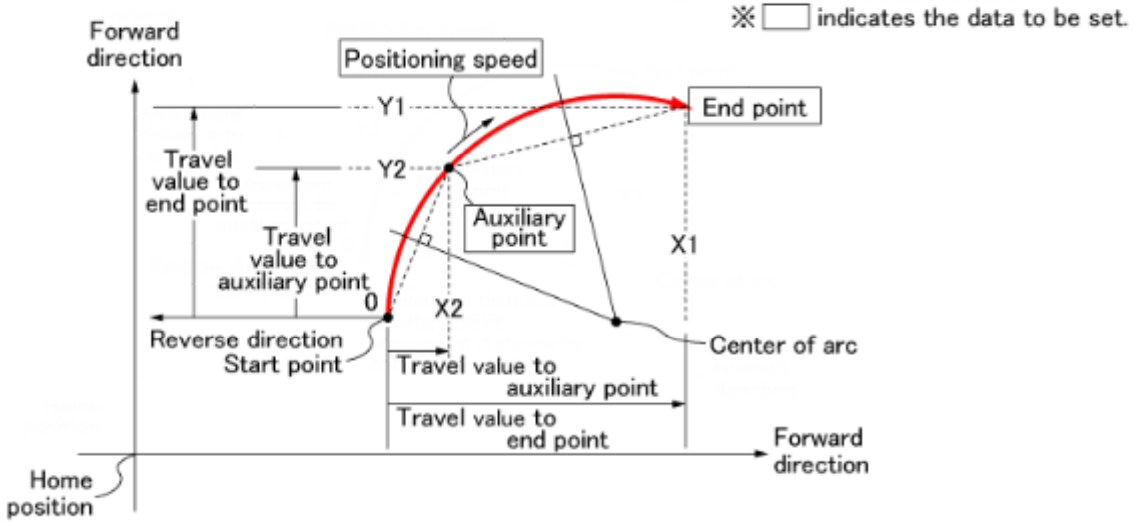
The speed is completely changed at the pass point.



9.4.1 Servo instruction

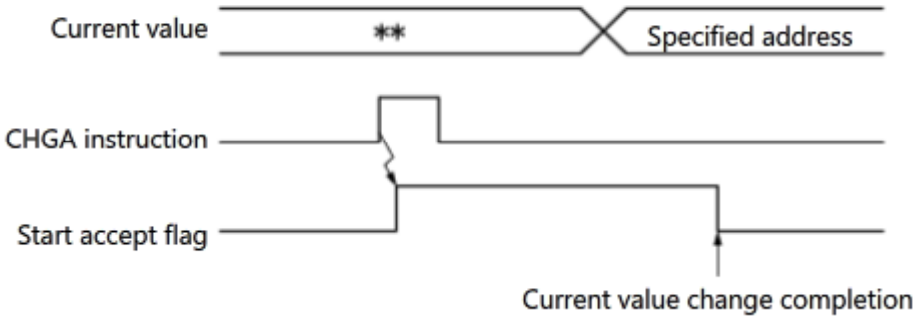
Symbol	Instruction name	Description
INC	Incremental auxiliary point-specified circular interpolation	• By the incremental method, 2-axis circular interpolation control is performed from a current stop position (start point) to a specified relative end address (X1, Y1) with an auxiliary address (pass point) (X2,Y2) passed. • The center of the arc is the point of intersection of the perpendicular bisectors of the start point (current stop) to the auxiliary point, and the auxiliary point to the end point .

※ □ indicates the data to be set.



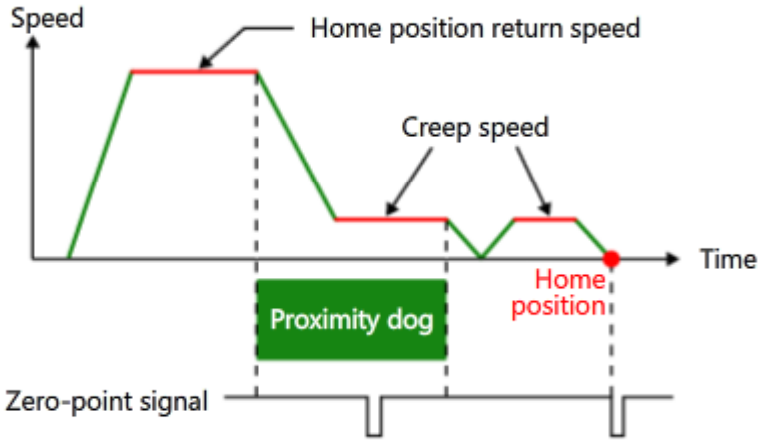
9.4.1 Servo instruction

Symbol	Instruction name	Description
CHGA	Current value change	• The current value of the specified axis is changed. • Only the current value of a stopped axis can be changed. A current value change for an operating axis causes a minor error 101. • A current value change is performed according to the following procedure. 1.Turn on "Start accept flag" corresponding to the specified axes. 2.Change the current values of the specified axes to the specified addresses. 3.Turn off "Start accept flag" on completion of the current value change. – Start accept flag: M200n (n: axis No.)



9.4.1 Servo instruction

Symbol	Instruction name	Description
ZERO	Home position return	• Perform the home position return when a machine home position needs to be checked, such as when a power supply is turned on. • Several methods are prepared for the home position return. Select a suitable method for system configuration or application. Specify a home position return method in the servo data setting screen. • In the sample system, "Proximity dog type 1" is used.



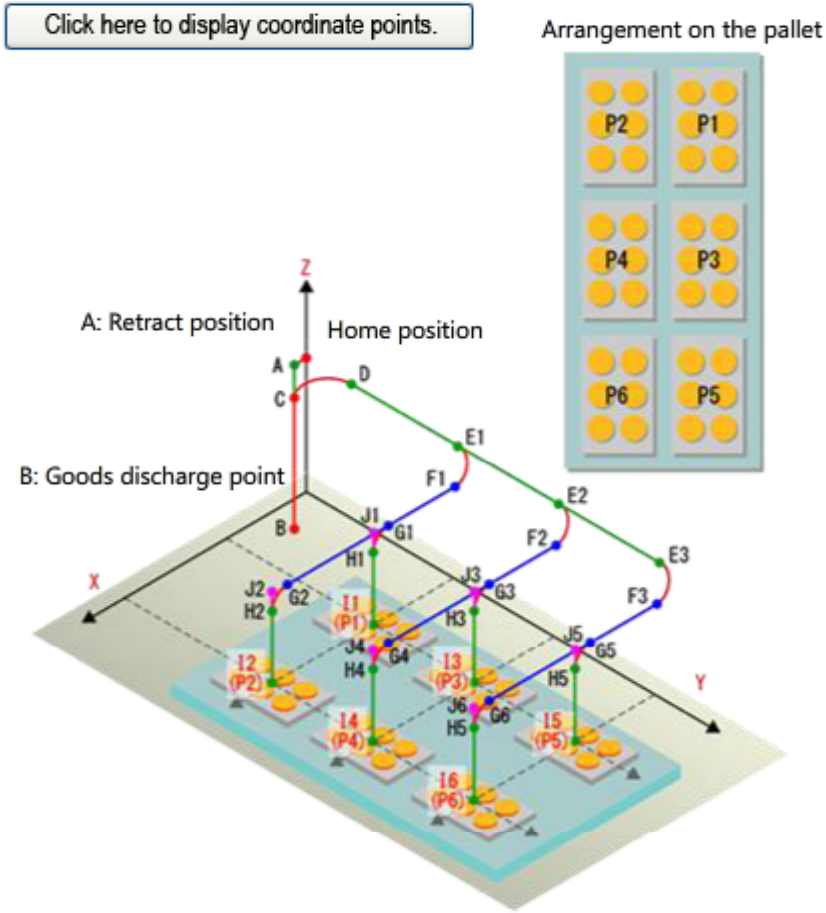
9.4.2 Designing the servo program of the sample system

Design a servo program based on the design specifications and the positioning control pattern of the sample system.

Goods arrangement points

The following figure and table show the positioning control pattern of the sample system and the servo instructions used for positioning control at each point.

Click here to display coordinate points.

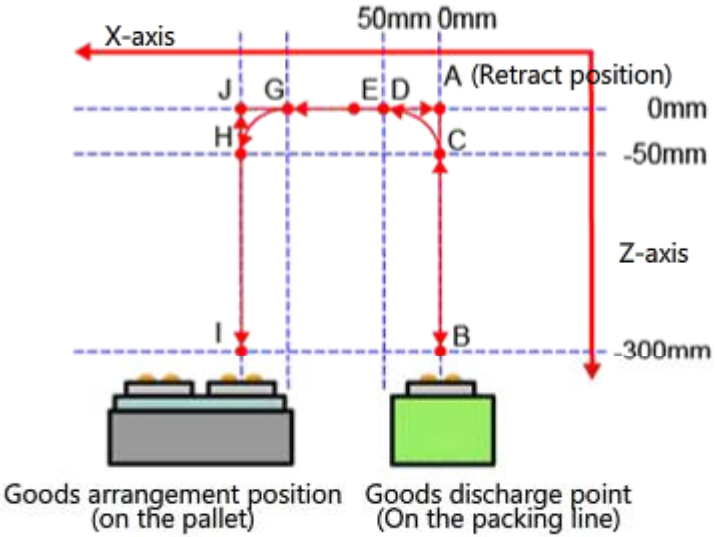


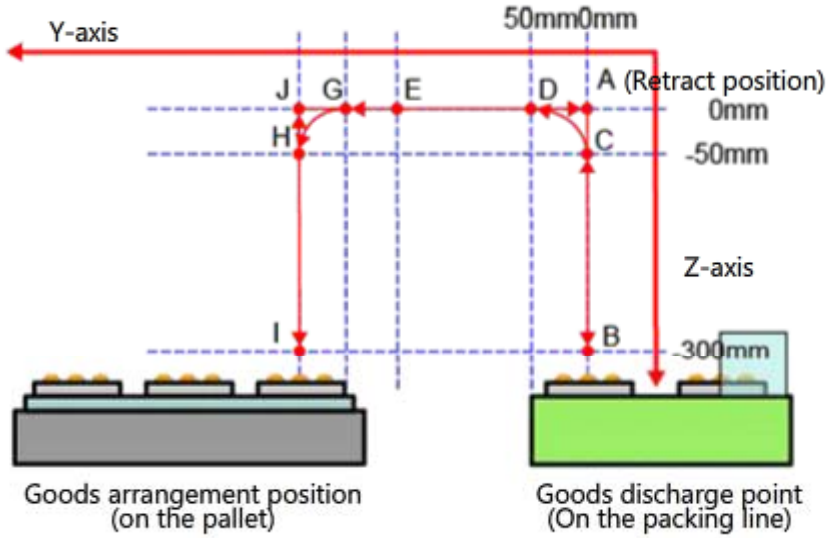
No.	Servo instruction			Movement range	Description
1	ZERO		Servo instruction	–	Home position return of axis 1 (X-axis)
2					Home position return of axis 2 (Y-axis)
3					Home position return of axis 3 (Z-axis)
10	INC-3		Incremental 3-axes linear interpolation	Home position -> A	The device is moved from the retract position to the offset position.
11	CHGA		Current value change	–	The current value of the offset position is changed to "0μm".
12					
13					
21	INC-3		Incremental 3-axes linear interpolation	A → B	The hand of the device (Z-axis) is lowered.
22	CPSTART3		3-axis constant-speed control start	–	Constant-speed control is started.
	1	INC-3	Incremental 3-axes linear interpolation	B → C	The hand of the device (Z-axis) is lifted.
	2	INC	Incremental radius-specified circular interpolation less than CW 180°	C → D	The device is moved to the arrangement position on the pallet.
	3	INC-3	Incremental 3-axes linear interpolation	D → E	
	4	INC	Incremental radius-specified circular interpolation less than CW 180°	E → F	
	5	INC-3	Incremental 3-axes linear interpolation	F → G	
	6	INC	Incremental radius-specified circular interpolation less than CW 180°	G → H	

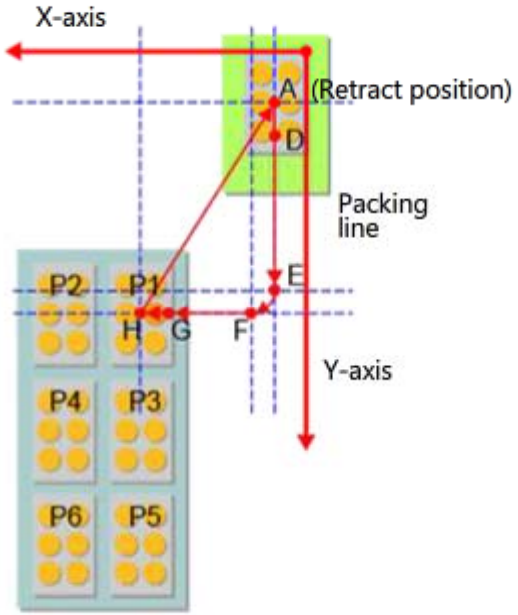
9.4.2 Designing the servo program of the sample system

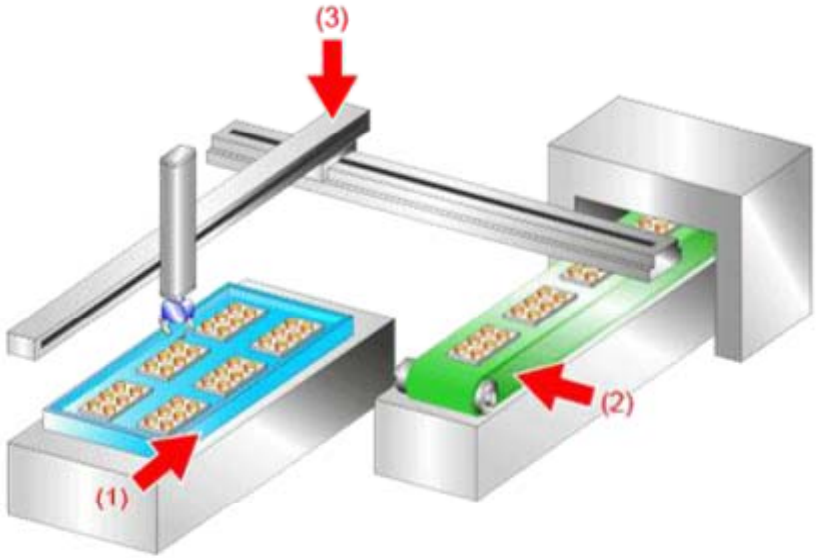
	7	INC-3	Incremental 3-axes linear interpolation	H→ I	The hand of the device (Z-axis) is lowered.
	CPEND		Constant-speed control end	–	Constant-speed control is completed.
23	CPSTART3		3-axis constant-speed control start	–	Constant-speed control is started.
	1	INC-3	Incremental 3-axes linear interpolation	I → J	The hand of the device (Z-axis) is lifted.
	2	ABS-3	Absolute 3-axes linear interpolation	J → A	The device is moved to the retract position.
	CPEND		Constant-speed control end	–	Constant-speed control is completed.

9.4.2 Designing the servo program of the sample system









9.5 Creating a Servo Program

Next, with MT Developer2, create the servo program you have designed.
In the sample system, you will create the following ten servo programs.
Let's create the servo programs in the next screen.

No.	Servo instruction			Movement range	Description
1	ZERO		Home position return	–	Home position return for axis 1 (X-axis)
2					Home position return for axis 2 (Y-axis)
3					Home position return for axis 3 (Z-axis)
10	INC-3		Incremental 3-axes linear interpolation	Home position -> A	The device is moved from the retract position to the offset position.
11	CHGA		Current value change	–	The current value of offset position of the axis 1 (X-axis) is changed to "0μm".
12					The current value of offset position of the axis 2 (Y-axis) is changed to "0μm".
13					The current value of offset position of the axis 3 (Z-axis) is changed to "0μm".
21	INC-3		Incremental 3-axes linear interpolation	A → B	The hand of the device (Z-axis) is lowered.
22	CPSTART3		3-axis constant-speed control start	–	Constant-speed control is started.
	1	INC-3	Incremental 3-axes linear interpolation	B → C	The hand of the device (Z-axis) is lifted.
	2	INC ↷	Incremental radius-specified circular interpolation less than CW 180°	C → D	The device is moved to the arrangement position on the pallet.
	3	INC-3	Incremental 3-axes linear interpolation	D → E	
	4	INC ↷	Incremental radius-specified circular interpolation less than CW 180°	E → F	
	5	INC-3	Incremental 3-axes linear interpolation	F → G	
	6	INC ↷	Incremental radius-specified circular interpolation less than CW 180°	G → H	
	7	INC-3	Incremental 3-axes linear interpolation	H → I	The hand of the device (Z-axis) is lowered.
	CPEND		Constant-speed control end	–	Constant-speed control is completed.

9.5 Creating a Servo Program

	CPEND		Constant-speed control end	–	Constant-speed control is completed.
23	CPSTART3		3-axis constant-speed control start	–	Constant-speed control is started.
	1	INC-3	Incremental 3-axes linear interpolation	I → J	The hand of the device (Z-axis) is lifted.
	2	ABS-3	Absolute 3-axes linear interpolation	J → A	The device is moved to the retract position.
	CPEND		Constant-speed control end	–	Constant-speed control is completed.

9.5 Creating a Servo Program

MELSOFT Series MT Developer2 C:¥MELSEC¥e-learning¥Packing Equipment - [Servo K23]

Project Edit Find/Replace View Check/Convert Online Debug Tools Window Help

Online Change OFF

Project

- Packing Equipment 10-9-30(SV13)
  - System Setting
  - Servo Data Setting
  - Motion SFC Program
  - Servo Program
    - Servo Program
      - K0001
      - K0002
      - K0003
      - K0010
      - K0011
      - K0012
      - K0013
      - K0021
      - K0022
      - K0023
  - Labels
  - Structured Data Types
  - Device Memory
  - Device Comment

K3 | Servo K10 | Servo K11 | Servo K12 | Servo K13 | Servo K21 | Servo K22 | Servo K23

```
[K  23]
   3    CPSTART3
         Axis        1
         Axis        2
         Axis        3
         Speed                    20000.00 mm/min
   1    INC-3
         Axis          1,              0.0 µm
         Axis          2,              0.0 µm
         Axis          3,         300000.0 µm
   2    ABS-3
         Axis          1,              0.0 µm
         Axis          2,              0.0 µm
         Axis          3,              0.0 µm
   3    CPEND
```

The servo programs have been created.

Click and go to the next screen.

Q172D | SV13 | Host Station No.2

9.6 Summary

The following lists the contents you learned in Chapter 9.
The following points are very important, so please check them again.

Flowcharting the Sequence of Control	Motion SFC language is a programming language similar to a flowchart. Expressing the sequence of control in a flowchart facilitates the design of a motion SFC program.
Servo program	• A servo program is a programmed pattern of positioning control. The program consists of servo instructions, axis Nos., addresses (travel value), movement speed, and acceleration pattern, etc. • In positioning control with a motion SFC program, specified servo programs are executed according to the control pattern.
Servo instruction	Instructions to command positioning control. Many servo instructions, such as simple linear positioning with one axis and linear and circular interpolations with two or more axes, are prepared.

Chapter 10 MOTION SFC PROGRAM

In Chapter 10, you will learn the basics of the motion SFC program.
In the end of the chapter, you will design a motion SFC program in the control procedure (flow chart) of the sample system.

SELECTING AND INSTALLING THE OPERATING SYSTEM SOFTWARE ····· "Chapter 6"

↓

SYSTEM SETTING ····· "Chapter 7"

↓

OPERATION CHECK ····· "Chapter 8"

↓

PROGRAM DESIGN ····· "Chapter 9"(BASICS OF MOTION SFC: Chapter 10)

↓

PROGRAMMING ····· "Chapter 11"

↓

OPERATION

Learning procedure of Chapter 10

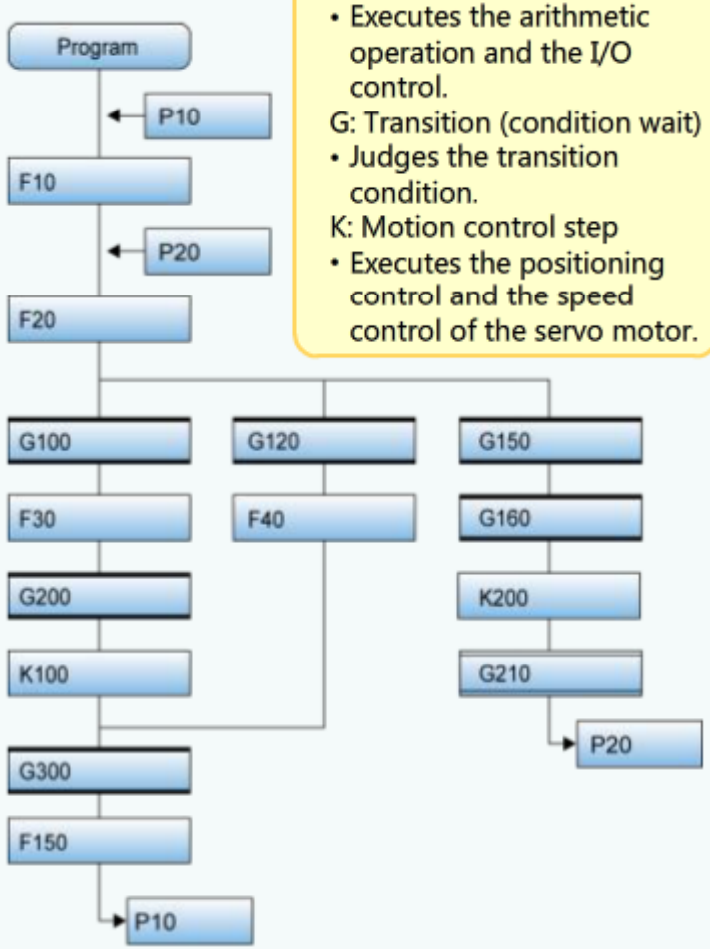
10.1 Features of the Motion SFC Program
10.2 Configuration Component of the Motion SFC Program
10.3 Available Device Type
10.4 Operation Sequence of the Motion SFC Program
10.5 Creating a Motion SFC Program of the Sample System
10.6 Start Method of the Motion SFC Program

10.1 Features of the Motion SFC Program

Motion SFC program

Motion SFC program is a program similar to the flow chart to diagrammatize a flow of a programming process.
It is also easy to handle for a person who learns the motion control programming for the first time.
The following shows the features of the motion SFC program.

Point	Feature
Program which can be used by anyone	By associating each operation of the machine with each operation step, a program which anyone can easily understand can be created using the flow chart format. Therefore, the maintainability is improved.
Not affected by the scan time	Since the motion CPU module determines the transition condition and starts positioning, there is no variation in the response time affected by the scan time at the PLC CPU module side.
Shortened tact time	The motion CPU module can process not only the positioning control but also the numerical operations, SET or RST of device, etc. This makes operations via a PLC CPU module unnecessary and shortens the tact time.
Transition condition description specific to the motion SFC	The transition condition description specific to the motion SFC allows a command be given to the servo amplifier after the start condition is satisfied. In addition, a transition to the next step can be performed after a positioning is started, without waiting for completion.



F: Operation control step
- Executes the arithmetic operation and the I/O control.

G: Transition (condition wait)
- Judges the transition condition.

K: Motion control step
- Executes the positioning control and the speed control of the servo motor.

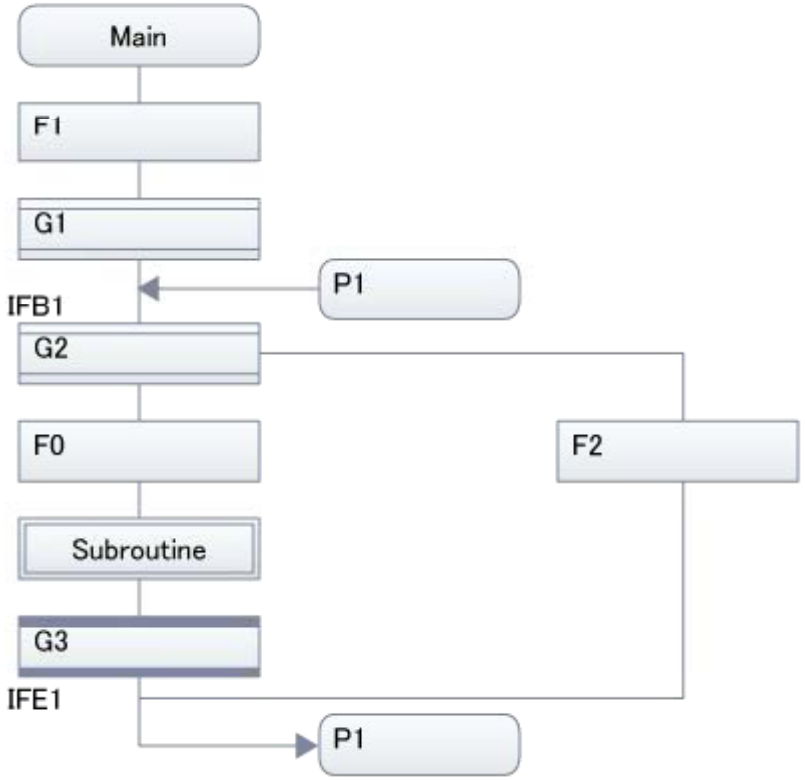
Supplement

The motion controller can be controlled from the sequence program, by using the motion dedicated sequence instruction.
For details, refer to the manuals.

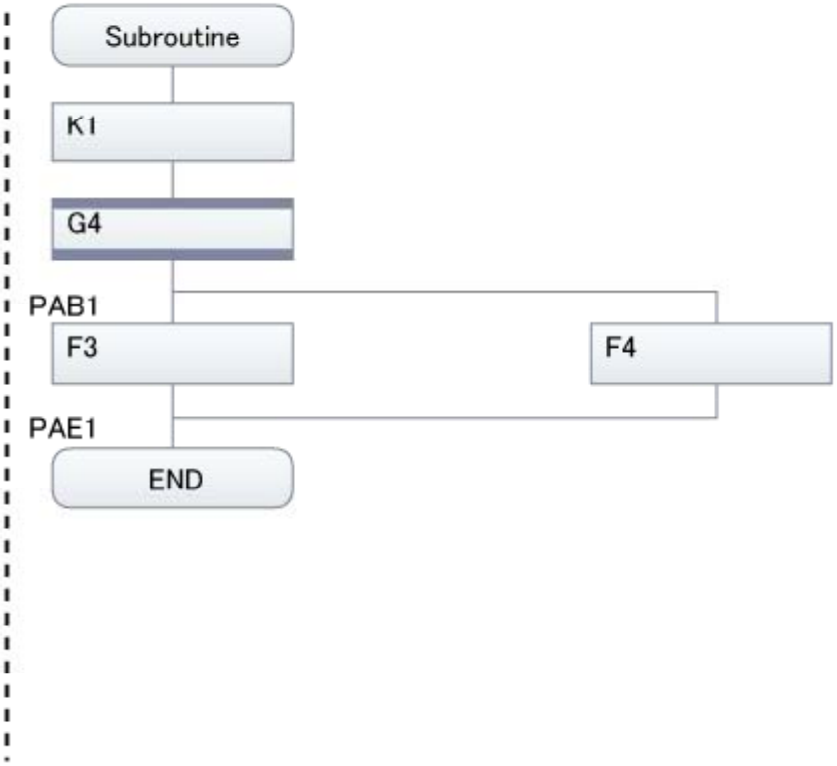
10.2 Configuration Component of the Motion SFC Program

The configuration components of the motion SFC program (SFC chart symbols) are explained by taking a sample program as an example.
Pointing the mouse cursor on the figure of each configuration component displays how to use these components.

Program example: Main program



Program example: Subroutine program



10.3 Available Device Type

You can use the following devices in the motion SFC program.

	Device		Symbol	Number of points	Read	Write	Remarks
Bit	Input or output	Input	X	8192 points	○	○	Note) A motion CPU module cannot access the I/O module with "X" and "Y". Use "PX" and "PY" instead.
		Output	Y		○	○	
		Input	PX	256 points	○	×	Devices for the I/O modules controlled by the motion CPU module Use "PX" and "PY" when accessing the I/O module.
		Output	PY		○	○	
	Internal relay		M	12288 points	○	○	This device can be used within the range of M0 to M8191.
	Link relay		B	8192 points	○	○	—
	Annunciator		F	2048 points	○	○	—
	Special relay		SM	2256 points	○	○	—
Word	Data register		D	8192 points	○	○	This device can be used within the range of D0 to D8191.
	Link register		W	8192 points	○	○	—
	Special register		SD	2256 points	○	○	—
	Motion register		#	12288 points	○	○	Use #8000 to #8639 as monitor devices and #8640 to #8735 as motion error history devices.

Multiple CPU shared device

CPU	Symbol	Number of points	Read	Write	Remarks
Self CPU	U□ G	Max. 14336 points*	○	○	You can share the device range assigned in the multiple CPU setting between the CPU modules and also access the devices controlled with the PLC CPU module. * The available points differ depending on the system setting.
Other CPU			○	×	

10.3 Available Device Type

Positioning dedicated device

This is a device which can access the status of the motion CPU module and each axis.
It uses a part of the range in the internal relay (M) and the data register (D).
For details, click the button below. You can browse a list of the devices in PDF format.

List of positioning-dedicated devices

In the sample system, use the following positioning dedicated devices.

Device No.	Application	Remarks
M2042	Set all the axes to the servo-on status.	—
M2415	Used to check the servo-on status for the axis 1.	The device is turned on in the servo-on status.
M2435	Used to check the servo-on status for the axis 2.	
M2001	Used to check the start acceptance status for the axis 1.	The device is turned on when the servo is operating.
M2002	Used to check the start acceptance status for the axis 2.	
M2003	Used to check the start acceptance status for the axis 3.	

Motion register

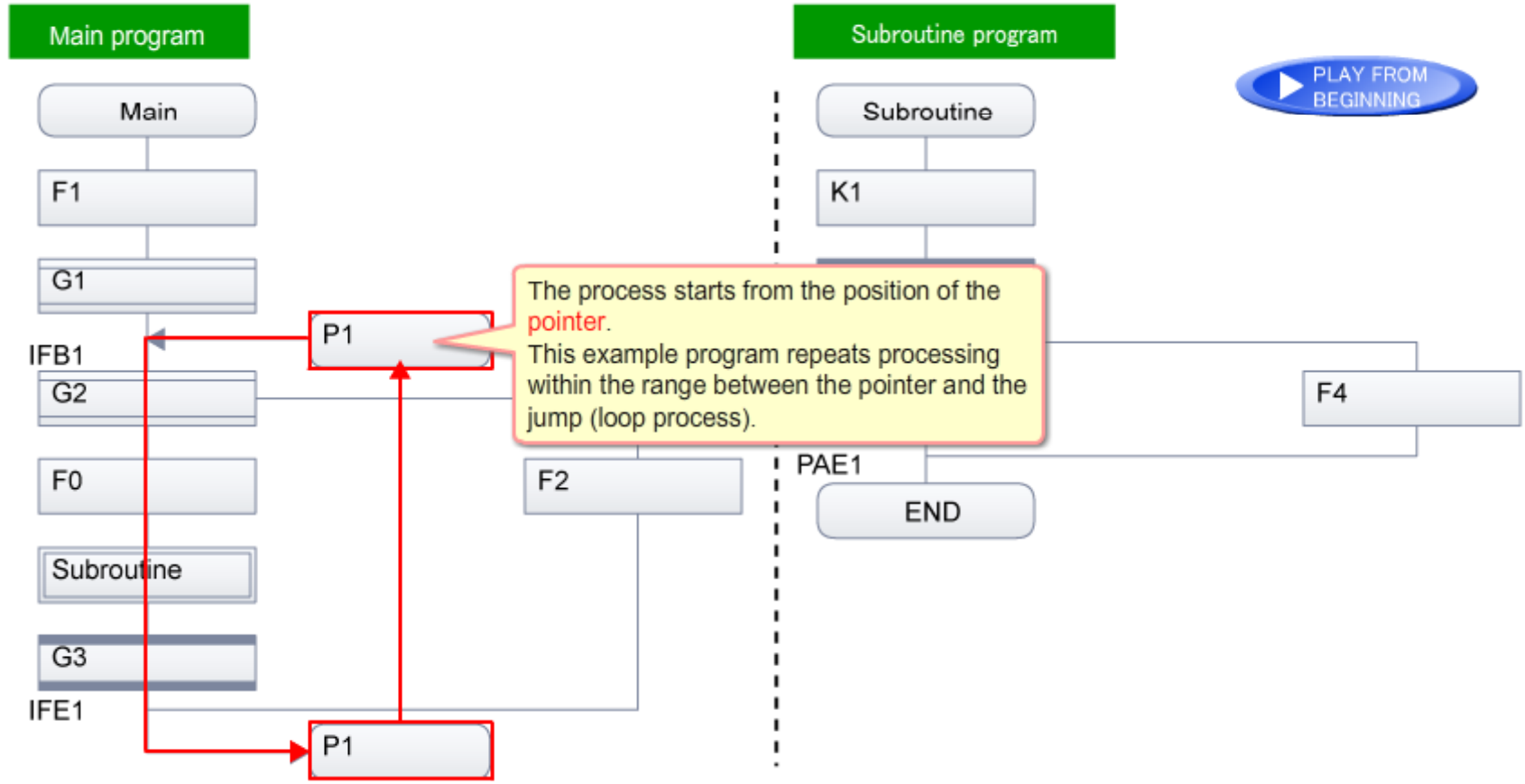
This is a device which can access the monitor value and the error history of each axis.
"#" is used for the device symbol. (In the sample system, the motion resister is not used.)
For details, click the button below. You can browse a list of the devices in PDF format.

List of motion resisters

10.4 Operation Sequence of the Motion SFC Program

The basic operation sequence of the motion SFC program starts from the "START" symbol and finishes with the "END" symbol.
If the program shifts to the transition during the operation, it does not shift to the next step until the conditions are satisfied. (Waiting for the conditions to be satisfied)
Also, the operation sequence changes if condition branches, jumps and calls of subroutine are included.

Let's check the operation sequence by taking a sample program as an example. Click the "PLAY" button to play the animation.



10.5 Creating a Motion SFC Program of the Sample System

Create a motion SFC program in the control procedure (flow chart) of the sample system.
The following shows motion SFC programs to be created.

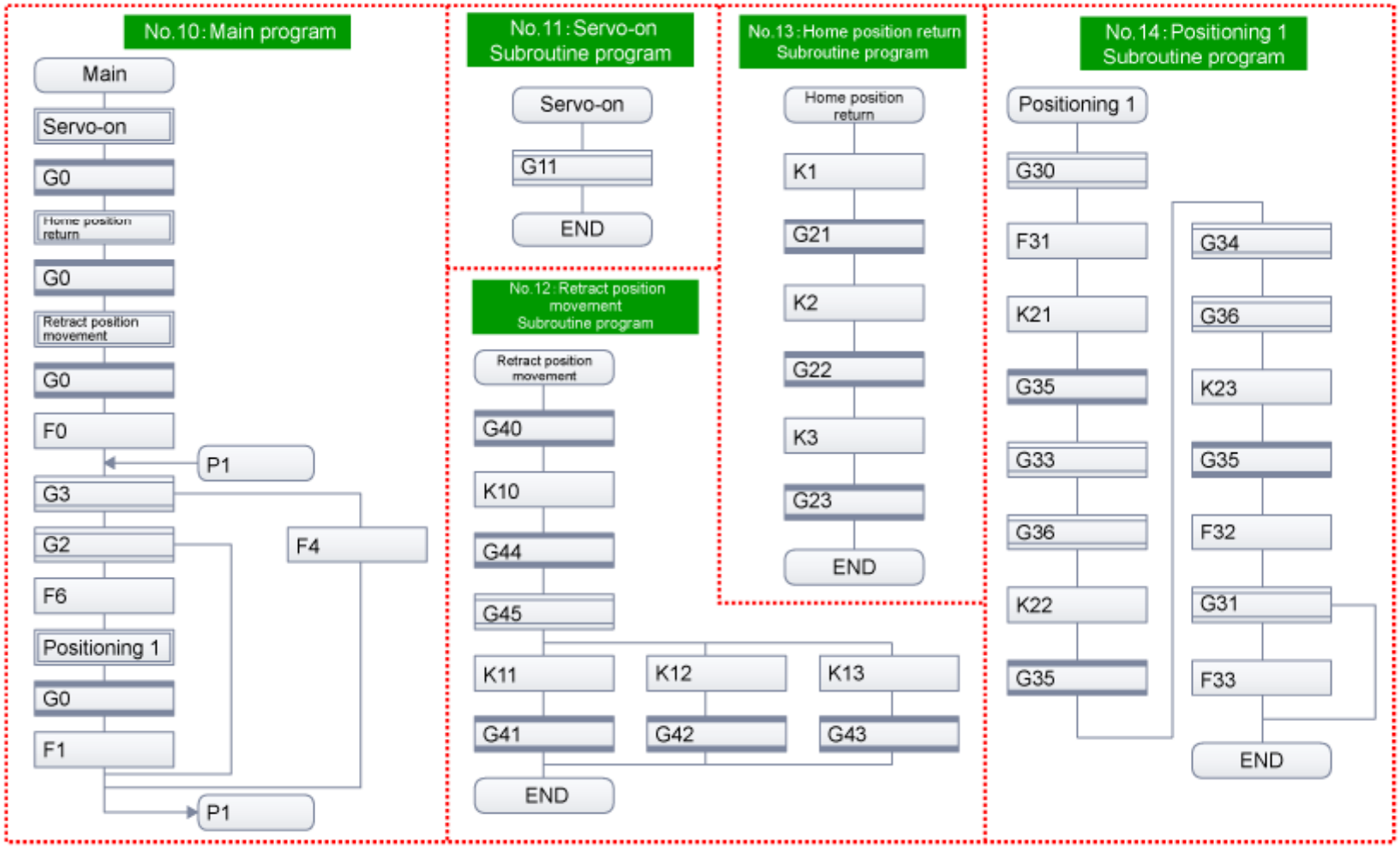
No.	Program name	Program description
10	Main	This is a main program to be executed from the sequence program for start. This program executes each subroutine of "Servo-on", "Retract position movement", "Home position return" when the system starts. After completion of the above subroutines, the program waits for the input of the start signal (PY12). When the start signal (PY12) is turned on, the "Positioning 1" subroutine is executed and the goods move onto the pallet. Until the number of arranged goods on the pallet becomes 6, the subroutine is executed repeatedly.
11	Servo-on	This is a subroutine program executed when the system starts from the main program. It sets all the axes to the servo-on status.
12	Retreat position movement	This is a subroutine program executed when the system starts from the main program. It moves the hand part of the device from the home position to the retreat position (the position where the hand part will be the center of the goods) and sets the position to "0mm" by using the current value change. By setting the retreat position to "0mm" by using the current value change, the address (travel value) can be easily obtained.
13	Home position return	This is a subroutine program executed when the system starts from the main program. It executes the home position return for all the axes by using the "Proximity dog type 1".
14	Positioning 1	This is a subroutine program for the positioning control executed when the start signal is turned on from the main program. It arranges six of the goods from the packing line onto the pallet. The arrangement coordinate is calculated from the number of arranged goods. When the number of arranged goods becomes 6, the number is reset to 0.

Point

Writing all the processes to a single program makes the program complicated and difficult to read when creating a program. Dividing the program according to the control contents (creating subroutines), calling and executing them from the main program makes the program simple and the program becomes easy to read.
In addition, the programming efficiency improves since there is no need to write the same processing details many times.

10.5 Creating a Motion SFC Program of the Sample System

The following shows the motion SFC programs in the sample system.
Pointing the mouse cursor to the figure of each configuration component displays each processing detail.



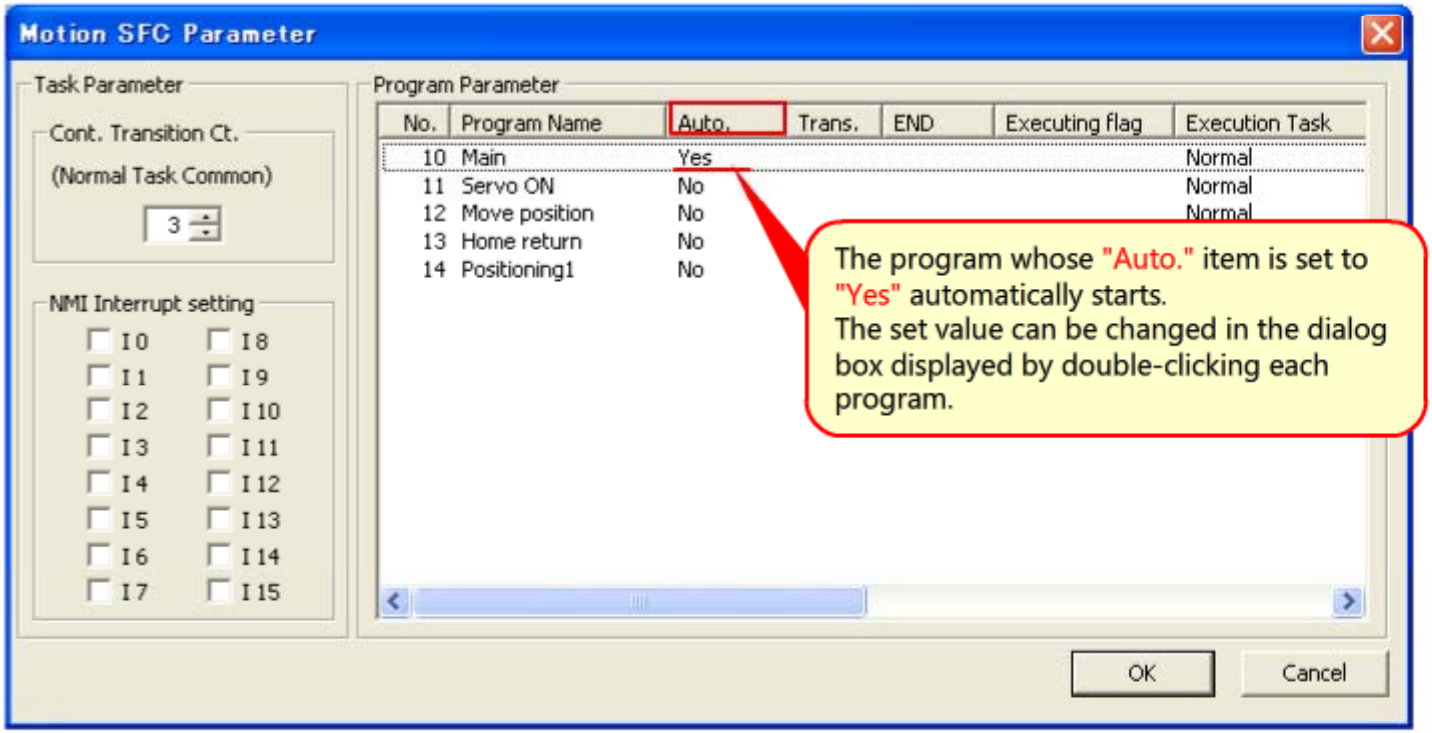
10.6 Start Method of the Motion SFC Program

The following two methods are available for starting the motion SFC program.

Automatically start when the motion CPU module is executed

By turning on the PLC READY flag (M2000) of the motion CPU module, any motion SFC program can be automatically started.
The sequence program to start the motion SFC program is unnecessary.
In the default setting, setting the STOP/RUN switch of the motion CPU module to RUN turns on M2000.

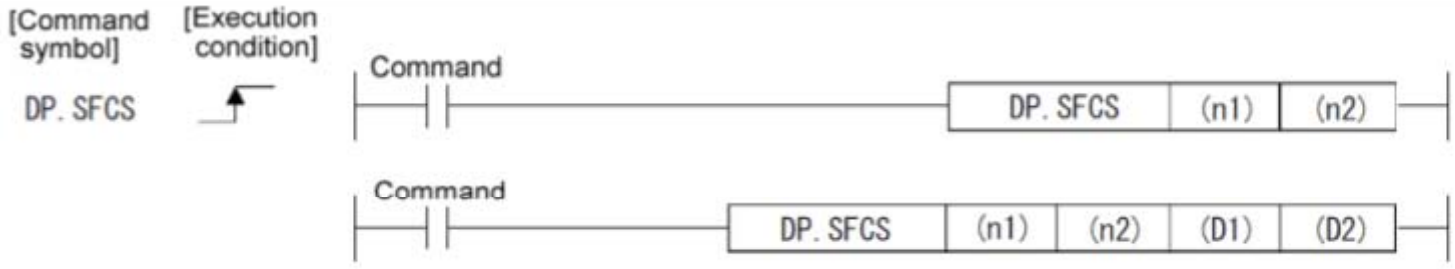
Set the program to be started automatically by the motion SFC parameter of MT Developer2.
Set "Auto." item of the program to "Yes". (The default setting is "No".)



10.6 Start Method of the Motion SFC Program

Start by using the motion dedicated sequence instruction from the sequence program

Any motion SFC program can be started by the motion SFC start request "D(P).SFCS" command from the sequence program.
The motion control system coupled with the sequence control can be established.



Setting data	Setting details	Set by	Data type
(n1)	(First I/O No. of the target CPU)/16 Value to specify actually is the following. CPU No.2 : 3E1H, CPU No.3 : 3E2H, CPU No.4 : 3E3H (Note): Motion CPU cannot be set as CPU No.1 in the Multiple CPU configuration.	User	16-bit binary
(n2)	Motion SFC program No. to start.	User	16-bit binary
(D1) (Note-1)	Completed devices (D1+0): Device which make turn on for one scan at accept completion of instruction. (D1+1): Device which make turn on for one scan at accept abnormal completion of instruction. (On completion of the error, D1 + 0 is turned on as well.)	System	bit
(D2) (Note-1)	Complete status storage Device	System	16-bit binary

Note-1: Omission possible with both of (D1) and (D2) omission.

10.7 Summary

The following lists the contents you learned in Chapter10.
The following points are very important, so please check them again.

Motion SFC program	This is a program similar to the flow chart for the motion control. It is also easy to handle for a person who learns the motion control programming for the first time.
Configuration component of the motion SFC	The motion SFC program is described by arranging and connecting the configuration component (SFC chart symbols). • START Starts a program. • Operation control step Executes the operation control program. • Motion control step Executes the servo program (positioning control). • Subroutine call/ start step Executes the subroutine program (other motion SFC programs). • Shift transition Without waiting for completion of the previous step, shifts to the next step when the transition conditions are satisfied. • WAIT transition After waiting for completion of the previous step, shifts to the next step when the transition conditions are satisfied. • Jump and pointer Moves a process to the specified pointer position. • END Finishes a program.
Operation sequence of the motion SFC	The basic operation sequence starts from "START" and finishes with "END". If a program shifts to the transition, it does not shift to the next step until the conditions are satisfied. Besides that, the sequence changes if the branch, subroutine, and others are executed.
Positioning dedicated device	This is a device which can access the status of the motion CPU module and each axis. A part of the range in the internal relay (M) and the data register (D) is assigned.
Motion register	With the motion CPU-dedicated device (Symbol: #), a motion controller can access the monitor values and error histories of each axis.

Chapter 11 PROGRAMMING

In Chapter 11, you will learn the method of programming and debugging the motion SFC program of the sample system with MT Developer2.

SELECTING AND INSTALLING THE OPERATING SYSTEM SOFTWARE ····· "Chapter 6"

↓

SYSTEM SETTING ····· "Chapter 7"

↓

OPERATION CHECK ····· "Chapter 8"

↓

PROGRAM DESIGN ·····"Chapter 9"(BASICS OF MOTION SFC: Chapter 10)

↓

PROGRAMMING ····· "Chapter 11"

↓

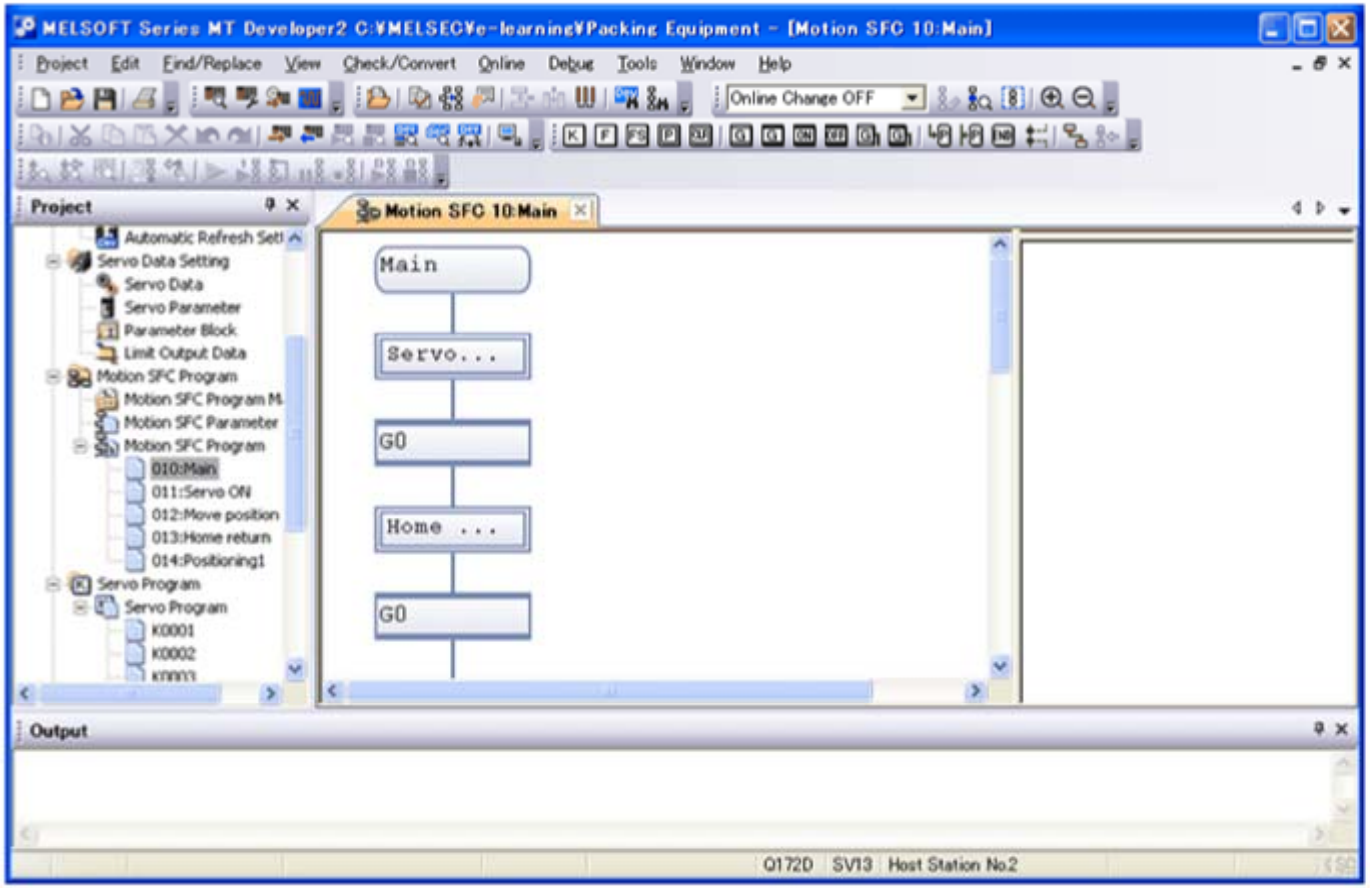
OPERATION

Learning procedure of Chapter 11

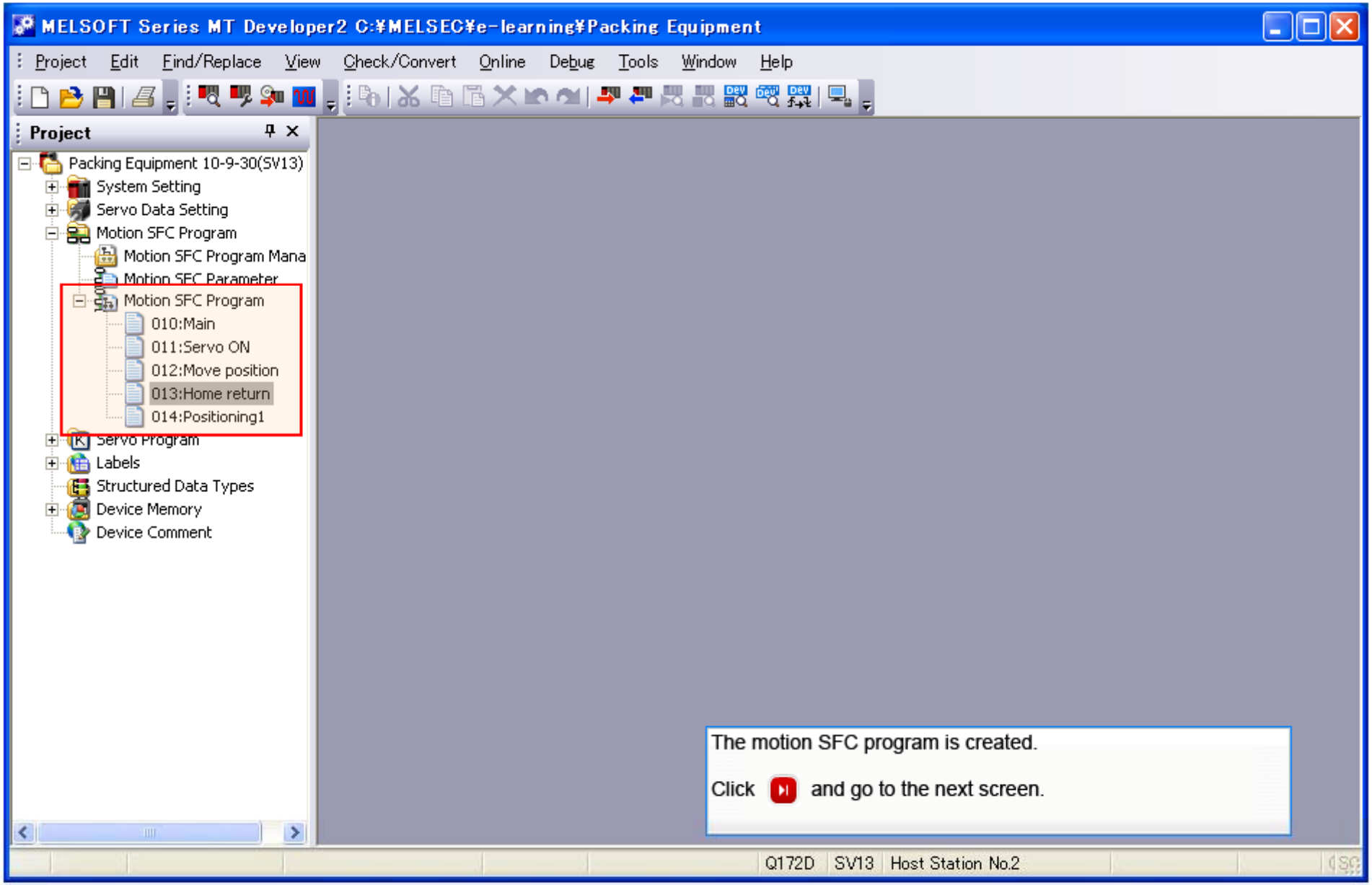
11.1 Creating a Motion SFC Program

Program the motion SFC program of the sample system with MT Developer2.
You will learn the basic operation such as selection, arrangement, alignment of figures and connection and disconnection of cables through programming.

Let's operate the programming of the motion SFC in the next screen.

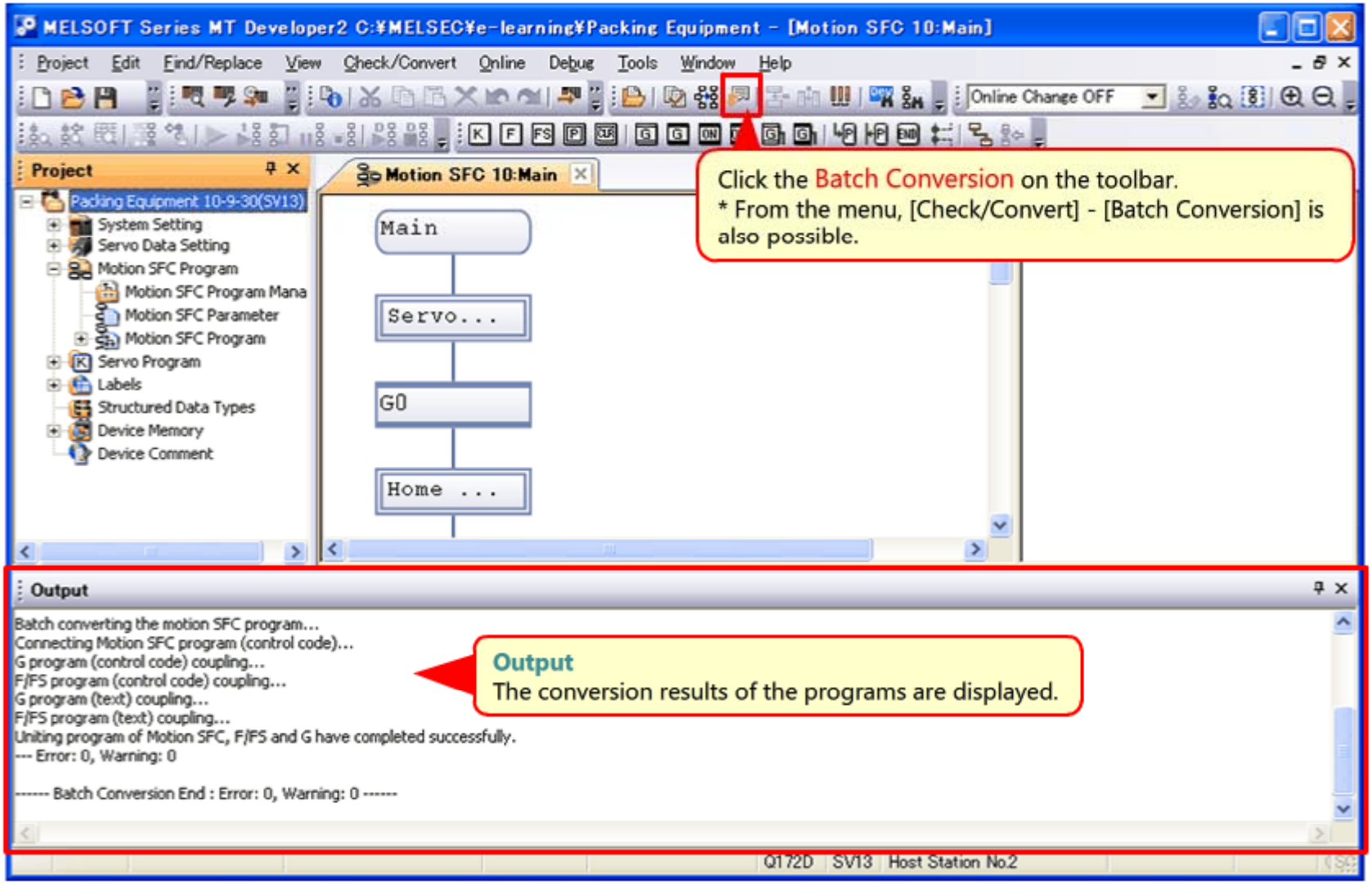


11.1 Creating a Motion SFC Program



11.2 Converting a Program into Practicable Format

After creating a program, convert it into a practicable format for the motion CPU module.
Programs which are not converted cannot be executed or stored.



11.3 Creating a Sequence Program for Start

Start the motion SFC program of the sample system with the motion dedicated PLC instruction "D（P）.SFCS" from the sequence program for start.

The following shows the sequence program for start of the sample system.
The motion SFC program No. 10 (Main) of the motion CPU module (No. 2) is started when M0 turns on.

* SM403 in the program is the special relay which turns off only at the first scan after the PLC CPU module has started.

Create the sequence program with GX Works2. (You cannot create sequence programs with MT Works2.)
Write the created programs to the PLC CPU module using Write to PLC of GX Works2.

11.4 Debugging a Motion SFC Program

After the completion of the programming, check that the program operates according to the design.
We call the cause of incorrect operation (fault point) Bug and research and correcting works Debug.

Debug is an important work. Do not execute programs in the actual system without debugging.
If bugs are left, they may cause abnormal stop, malfunctions or troubles.

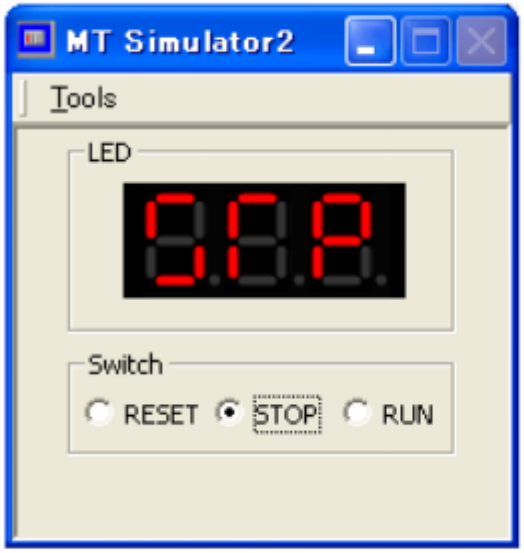
The functions to support debug are equipped in MT Developer2.

Name	Description
Simulator	A function to simulate the execution of the program without a motion CPU module. Use this in an environment where a motion CPU module cannot be provided for debug.
Monitor	A function to monitor the execution status and the status of the each device. There are various monitor functions such as the function to monitor only registered devices, to monitor the motion SFC program in execution.
Device test	A function to perform a test for checking the operations of created programs by turning on/off bit devices, writing the data of word device. Debug can be performed without connecting the I/O devices.

Precautions
Perform debugging without connecting I/O devices or machine to the motion controller system or servo motors.
Unintended operations may be caused by bugs.

11.4.1 Debugging without using a motion CPU module

Use a Simulation function when a motion CPU module cannot be provided for debug.
The operation of the program can be simulated on the virtual motion CPU module on the software.



Item	Status	Description
Switch	RUN	Executes a virtual motion CPU module.
	STOP	Stops a virtual motion CPU module. (initial status)
	RESET	Resets a virtual motion CPU module. (This can be selected only during STOP.)
LED		Displays the status of the motion CPU module or errors in 7-segment LED.

Precautions

- There is no guarantee that the motion SFC program operates according to the simulation after debug.
- Input or output of I/O modules are executed by using the memory for simulation.
- Therefore, the operation result of the simulation function may differ from the operation result of the actual motion CPU module.

11.4.2 Debugging the program of the sample system

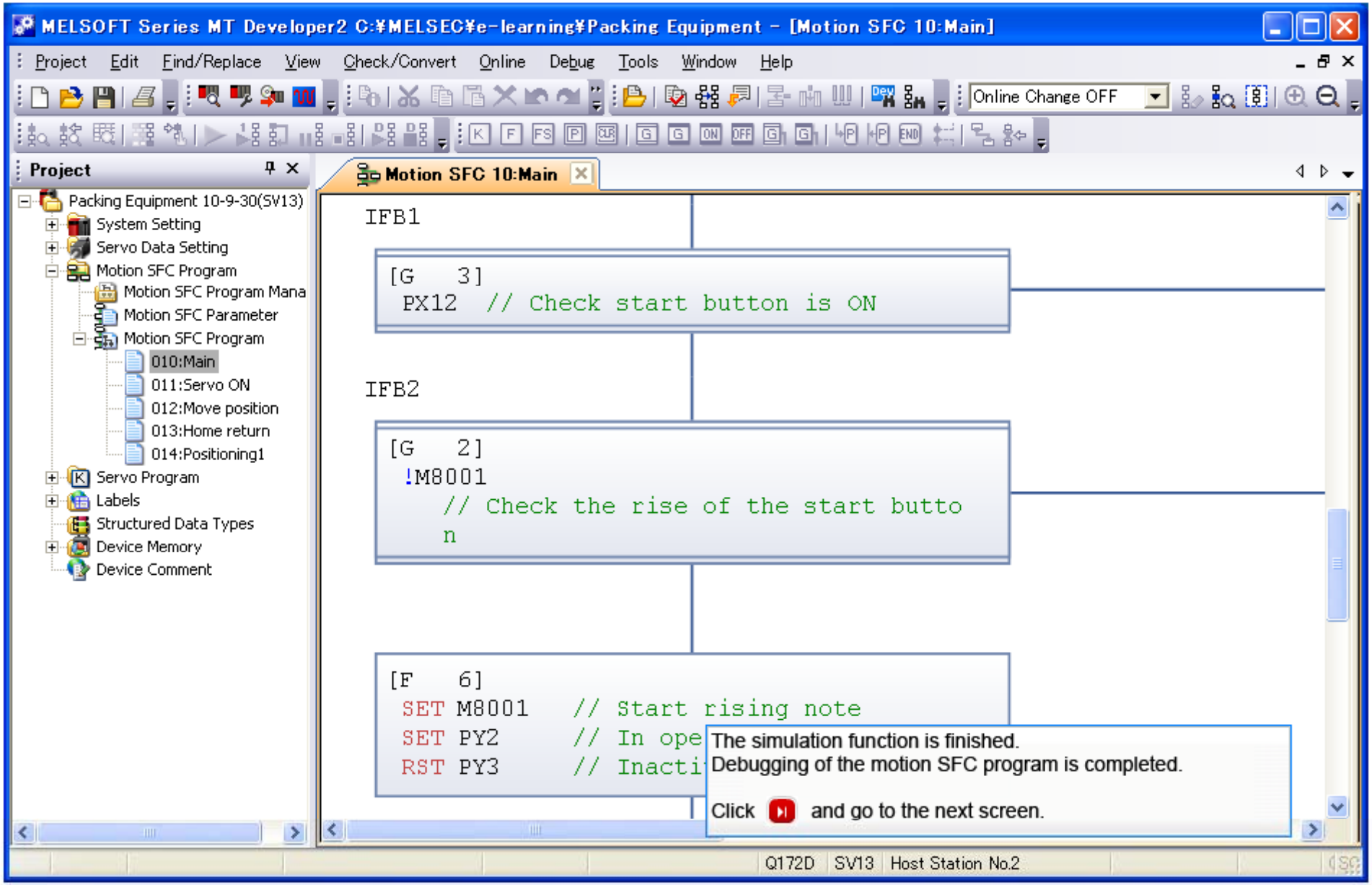
Debug the motion SFC program of the sample system with the simulation function.
First, shift an execution status of the program to debug mode.

In the debug mode, up to 4 stop positions of the program (which is called break point) can be specified.
The program stops when it shifts to the step specified as a break point.(This situation is called During break.)
During break, programs can be executed with the operation results or operations checked in one step at a time by using the following functions.

Function	Description
Request or cancel of debug mode	Shifts the execution of the program to the debug mode or releases the debug mode. When the program is shifted to the debug mode, the break function below can be used.
Execute or continue	Executes again or continues the program when the motion SFC program is during break or forced end.
One step execution	Shifts the motion SFC program during break from the break point to next step.
Forced shift	Shifts the program to the next step forcibly when it does not shift to the next step at the transition because the conditions are not satisfied.
Break	Ends the motion SFC program in execution or during break forcibly regardless of the break point.
Forced end	Ends the motion SFC program in execution or during break forcibly.

Let's operate the debug of the motion SFC program in the next screen.

11.4.2 Debugging the program of the sample system

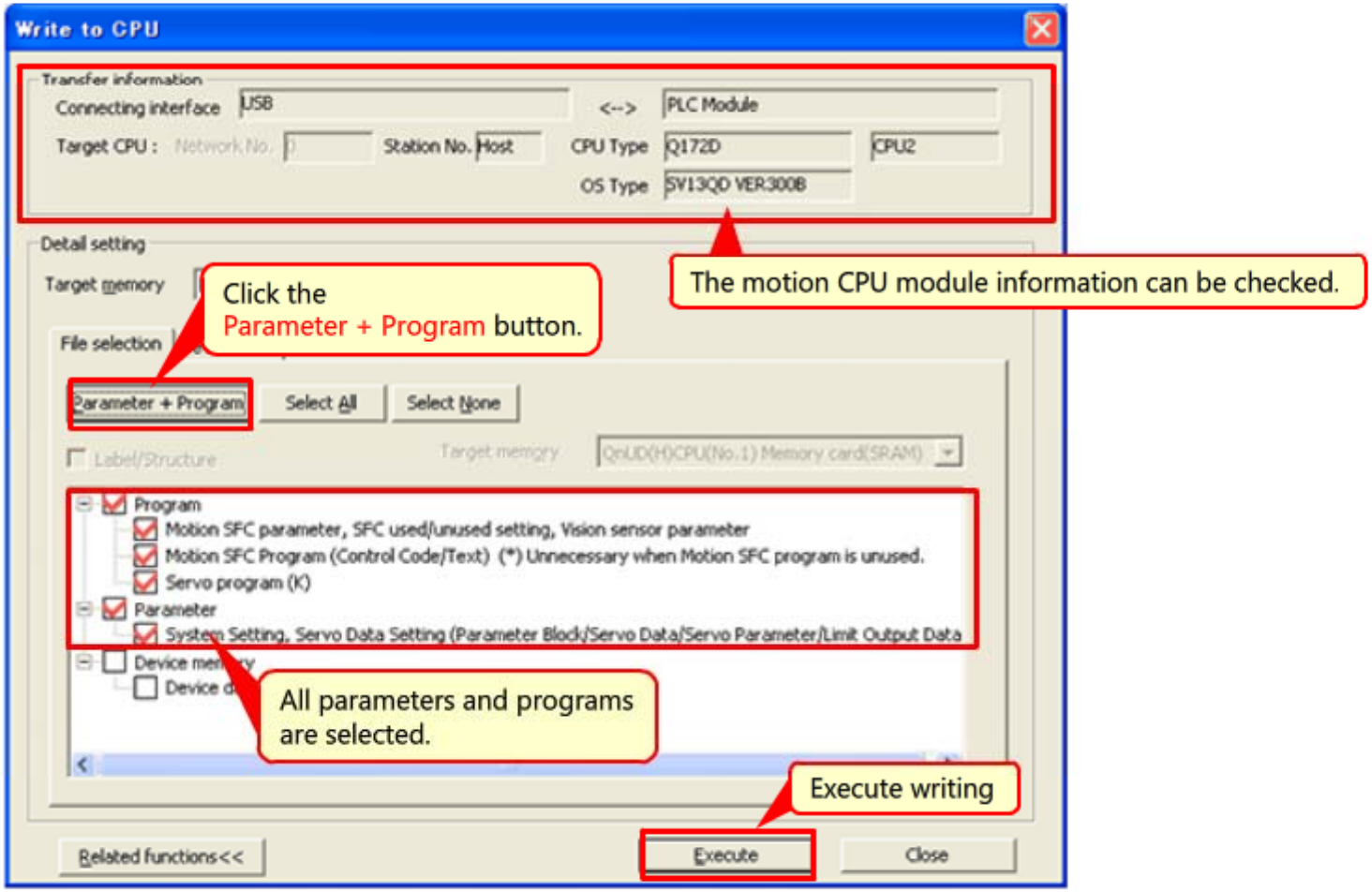


11.5 Writing a Motion SFC Program

Write a created motion SFC program to the motion CPU module.
Before writing, check the following.

- Power supplies of a motion controller and servo amplifier are turned on.
- The RUN/STOP switch of the motion CPU module is STOP.
- A personal computer and the PLC CPU module are connected correctly.

Click the Parameter + Program button in the following Write to CPU window and perform writing.



11.6 Executing the Motion SFC Program

Execute the motion SFC program wrote to the motion CPU module.
Operate the switches of the PLC CPU module and the motion CPU module in the following procedures.

1) Reset the PLC CPU module and the motion CPU module.
Set the RESET/ STOP/ RUN switch of PLC CPU to RESET.
Reset is performed by the No. 1 PLC CPU module.
All CPU modules, including the motion CPU module are reset.

2) Check error generation.

3) Execute the program.
Set the RESET/ STOP/ RUN switch of the PLC CPU module and the STOP/ RUN switch of the motion CPU module to RUN.

PLC CPU module

RESET/ STOP/ RUN

PLC CPU module

RESET/ STOP/ RUN

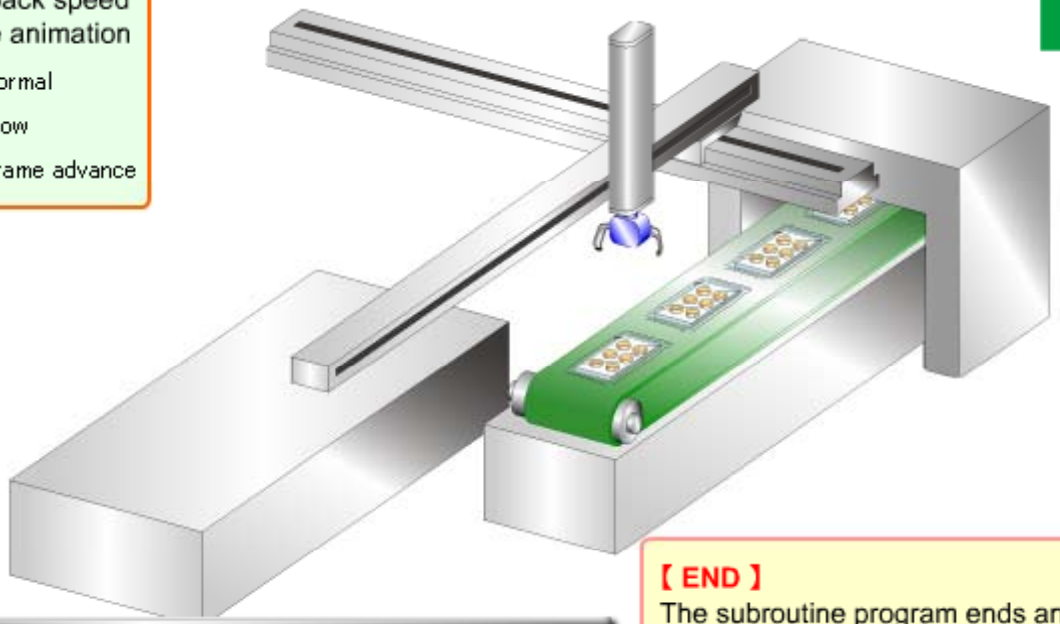
Motion CPU module

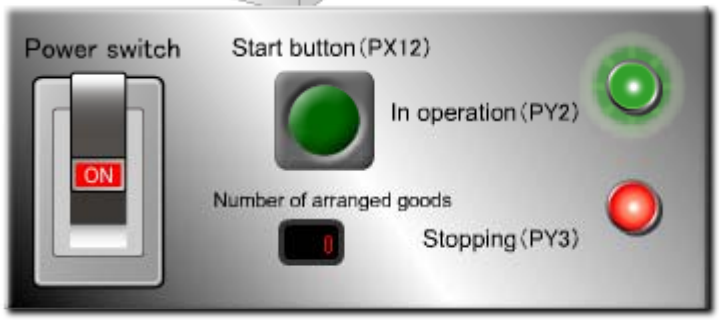
11.7 Completion of the Sample System

Finally, check the operation of completed sample system using the animation.

Operate the animation in the following sample system by a mouse according to the instruction of Click .

Playback speed of the animation
- (•) Normal
- ( ) Slow
- ( ) Frame advance





【 END 】
The subroutine program ends and the flow returns to the main program.

11.8 Summary

The following lists the contents you learned in Chapter11.
The following points are very important, so please check them again.

Convert a program	After creating a program, convert it into a practicable format for the motion CPU module. Programs which are not converted cannot be executed or stored.
Debug	After the completion of the programming, check that the program operates according to design. • We call the cause of incorrect operation (fault point) Bug and research and correcting works Debug. • Do not execute programs in the actual system without debugging. If bugs are left, they may cause abnormal stop, malfunctions or troubles.
Simulation function	Use a Simulation function when a motion CPU module cannot be provided. The operation of the program can be simulated on the virtual motion CPU module on the software.
Debug mode	Up to 4 stop positions of the program (which is called break point) can be specified. The program stops when it shifts to the step specified as a break point. (This situation is called During break.) During brake, the program can be executed using the following functions in one step at a time.
Execution of the motion SFC	1. Reset the PLC CPU module and the motion CPU module. Set the RESET/ STOP/ RUN switch of PLC CPU to RESET. Reset is performed by the No. 1 PLC CPU module. All CPU modules, including the motion CPU module are reset. 2. Checking error generation 3. Execute the program. Set the RESET/ STOP/ RUN switch of the PLC CPU module and the STOP/ RUN switch of the motion CPU module to RUN.

Test Final Test

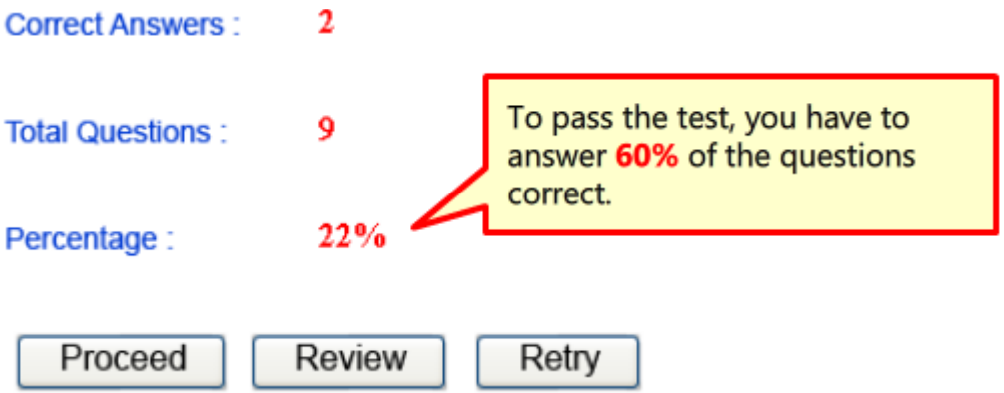
Now that you have completed all of the lessons of the MOTION CONTROLLER Basics (Real Mode:SFC) Course, you are ready to take the final test.
If you are unclear on any of the topics covered, please take this opportunity to review those topics.
There are a total of 5 questions (23 items) in this Final Test.
You can take the final test as many times as you like.

How to score the test

After selecting the answer, make sure to click the **Answer** button. Your answer will be lost if you proceed without clicking the Answer button. (Regarded as unanswered question.)

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.



- Click the **Proceed** button to exit the test.
- Click the **Review** button to review the test. (Correct answer check)
- Click the **Retry** button to retake the test again.

Test Final Test 1

Select the 3 correct features of the operating system software (described as Operationg system software hereinafter).

- [ ] Operationg system software is shipped with installed on a motion CPU module.
- [ ] Operationg system software is needed to be installed on a motion CPU module.
- [ ] Operationg system software is needed to be purchased separately from a motion CPU module.
- [ ] Operationg system software is enclosed with a motion CPU module.
- [ ] Set the motion CPU module to the install mode by the rotary switch before installing the Operationg system software.
- [ ] Operationg system software is already installed, so you can use the motion CPU module soon after the purchase.

Answer Back

Test Final Test 2

Select the functions of the configuration components (such as step, transition) used in a motion SFC program.

Configuration component		Processing detail
START	Main	
END	END	
Operation control step	F1	
Motion control step	K1	
Subroutine call/start step	Subroutine	
Shift transition	G1	
WAIT transition	G1	
Shift Y/N transition	G1	
Jump	P1	
Pointer	P1	

Answer Back

Processing detail

1. Executes the specified motion SFC program.
2. Without waiting for the completion of the previous step, shifts the program to the next step when the transition conditions are satisfied.
3. Jumps to the specified pointer Pn in the program.
4. Finishes the program or the subroutine program.
5. Branches to when the transition conditions are satisfied and not satisfied without waiting for the completion of the previous step.
6. Indicates the jump destination pointer (label).
7. When the previous step is the motion control step, waits for the completion of movement and shifts to the next step at the transition condition satisfied.
8. Executes the specified operation control program.
9. Starts the program or the subroutine program.
10. Executes the specified servo program.

Test Final Test 3

Select the correct program where the program shifts to the next step after the completion of movement of the motion control step.

○ Program example 1

○ Program example 2

○ Program example 3

Answer Back

Test Final Test 4

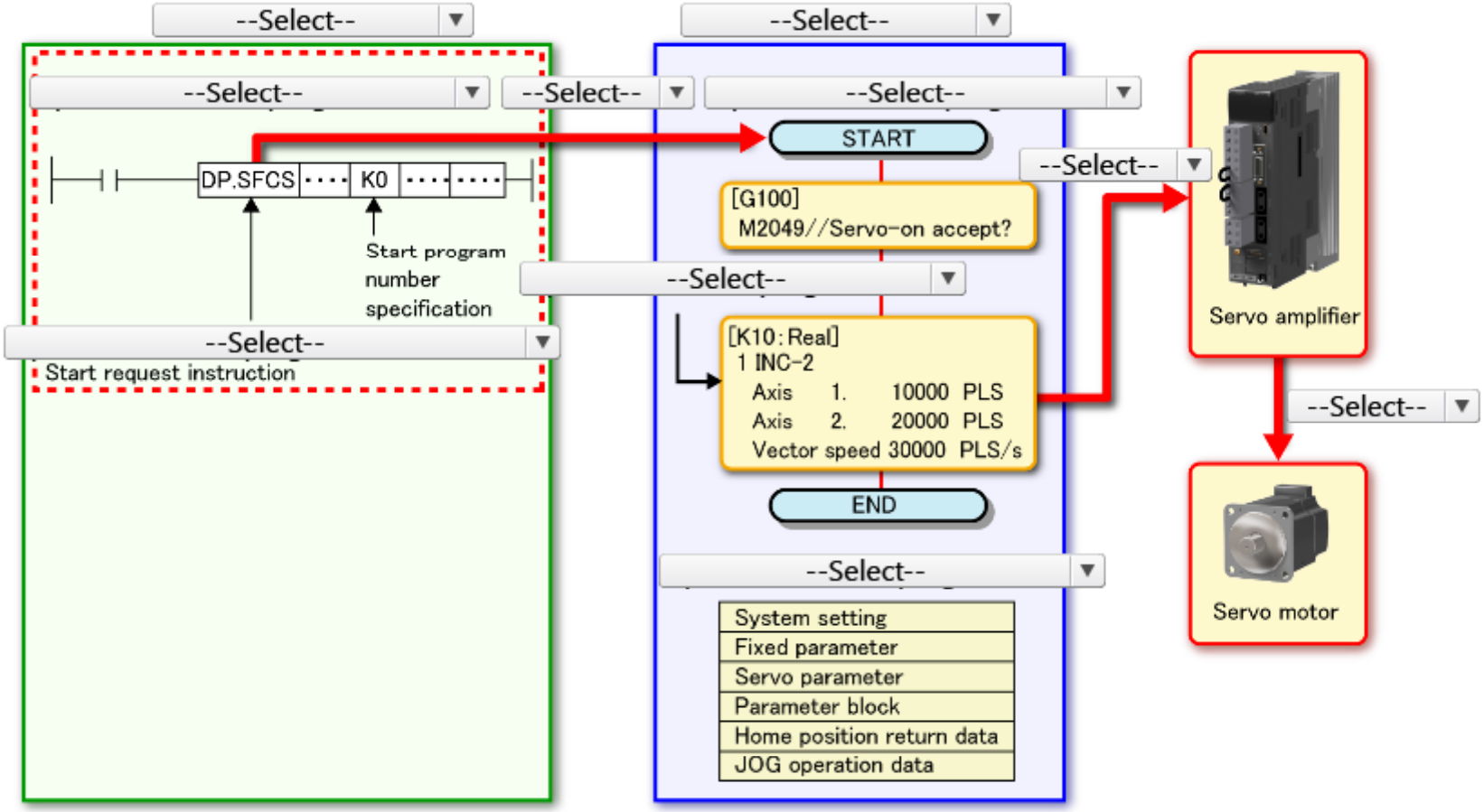
Select the 3 processes which should be executed before the positioning control when designing a motion SFC program.

- [ ] Servo-on
- [ ] Servo-off
- [ ] JOG operation
- [ ] Home position return
- [ ] Current value change
- [ ] Confirming that the start accept flag turns on
- [ ] Confirming that the start accept flag turns off

Answer Back

Test Final Test 5

Select the functions of the configuration components (such as step, transition) used in a motion SFC program.



Answer Back

Test Score

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

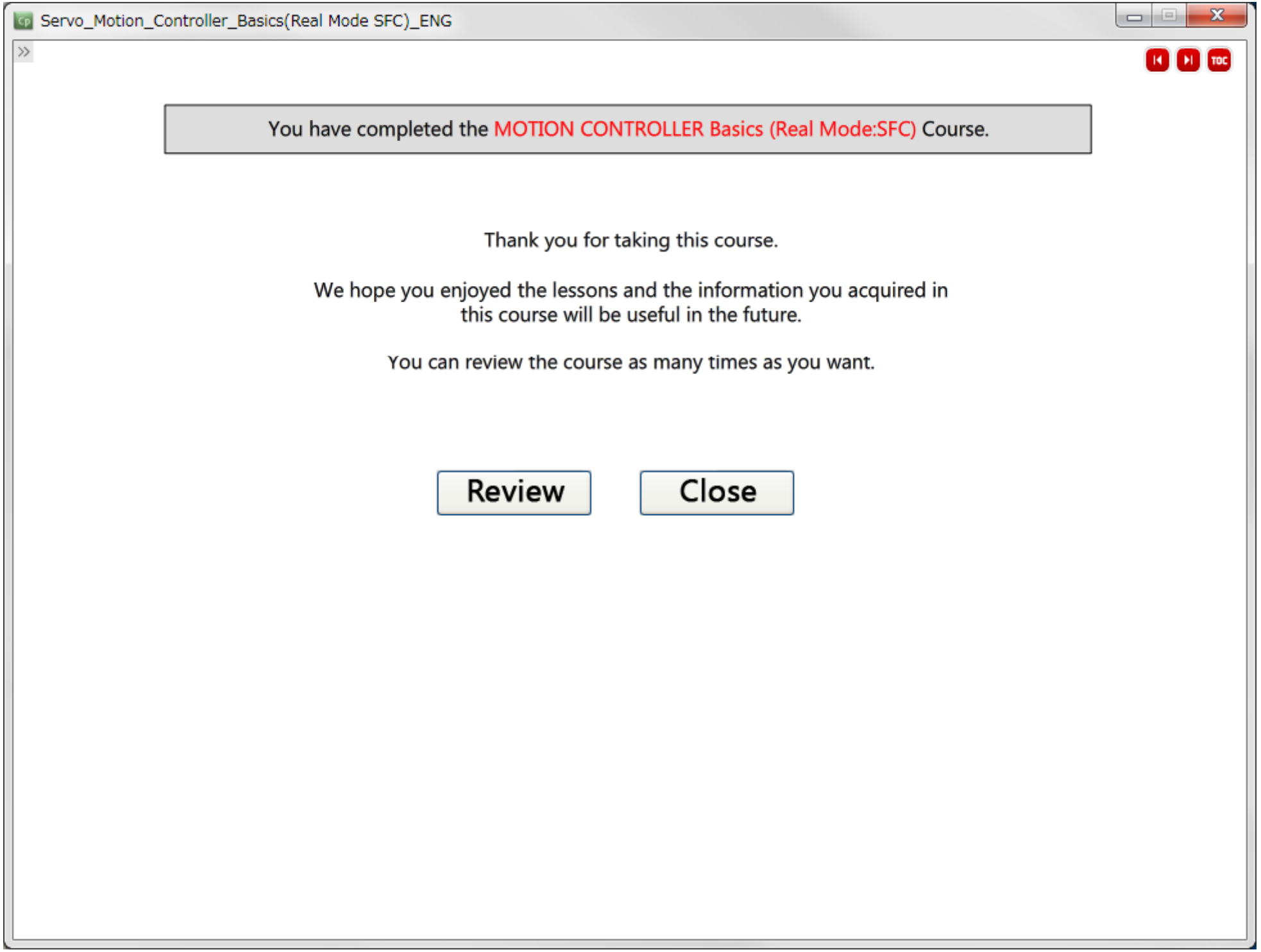
Correct answers : 0

Total questions : 5

Percentage : 0%

Proceed | Review | Retry

You failed the test.


Servo_Motion_Controller_Basics(Real Mode SFC)_ENG
TOC
You have completed the MOTION CONTROLLER Basics (Real Mode:SFC) Course.
Thank you for taking this course.
We hope you enjoyed the lessons and the information you acquired in this course will be useful in the future.
You can review the course as many times as you want.
Review
Close