

Programmable Automation Controller

MELSEC MX Controller Basics (Positioning)

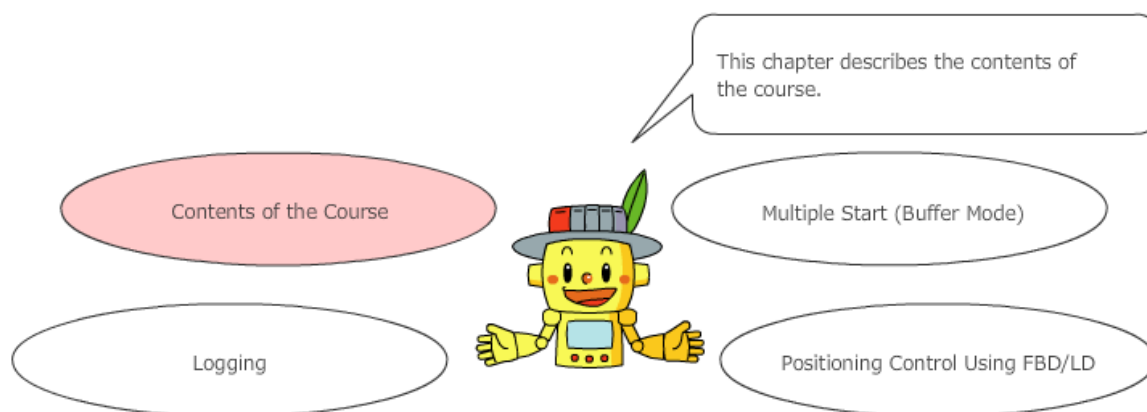
This course is for participants who will establish a motion control system using the MELSEC MX Controller MX-F Model for the first time.

* This course is also suitable for MX-R Model users, as it offers the equivalent explanations.

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

Introduction Purpose of the Course

This course aims to develop knowledge and understanding about positioning control of the motion control system using the MELSEC MX Controller MX-F Model.



This course is a continuation of the "MELSEC MX Controller Basics (Motion Control Startup)" course. Complete the above course before taking this course.

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

Chapter 1 - Contents of the Course

This chapter describes the contents of the course.

Chapter 2 - Multiple Start (Buffer Mode)

This chapter describes path control using buffer mode of motion control FBs.

Chapter 3 - Positioning Control Using FBD/LD

This chapter describes how to create a program using FBD/LD.

Chapter 4 - Logging

This chapter describes how to perform data logging using GX LogViewer.

Final Test

Passing grade: 60% or higher

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.

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 uses the following software version.

For the latest version of each software, check the Mitsubishi Electric FA Website.

MELSOFT GX Works3 Ver.1.125F

MELSOFT MR Configurator2 Ver.1.165X

The following table lists the titles and document numbers of manuals related to this course.
Download the manuals from the Mitsubishi Electric FA Website as needed and refer to them.
Each chapter indicates its relevant manuals.

Manual	Manual No.
MELSEC MX Controller MX-R Model User's Manual	SH-082641ENG
MELSEC MX Controller MX-F Model User's Manual	SH-082634ENG
MELSEC MX Controller MX-R Model Programming Manual	SH-082644ENG
MELSEC MX Controller MX-F Model Programming Manual	SH-082691ENG
MELSEC iQ-R/MELSEC iQ-F Structured Text (ST) Programming Guide Book	SH-081483ENG
MR-J5-G/MR-J5W-G User's Manual (Introduction)	SH-030294ENG
MR-J5 User's Manual (Hardware)	SH-030298ENG
MR-J5 User's Manual (Function)	SH-030300ENG
MR-J5-G/MR-J5W-G User's Manual (Parameters)	SH-030308ENG

The following shows a flow of this course.

Chapter 1 Contents of the Course

This chapter describes the contents of the course.

**Chapter 2 Multiple Start (Buffer Mode)**

This chapter describes path control using motion control FBs.

**Chapter 3 Positioning Control Using FBD/LD**

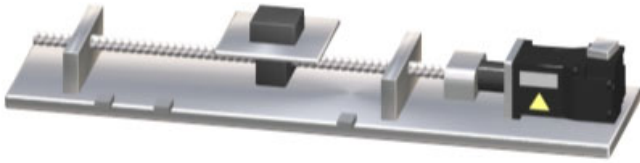
This chapter describes how to create programs using FBD/LD.

**Chapter 4 Logging**

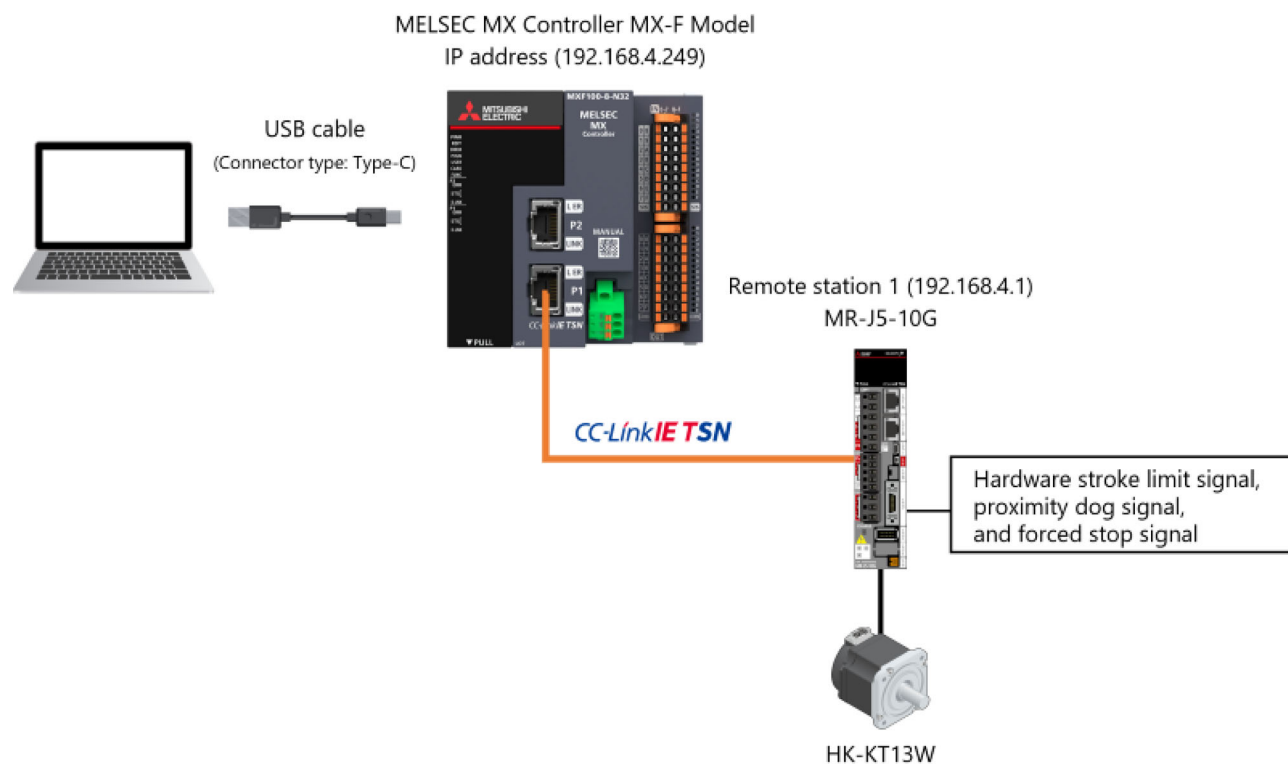
This chapter describes how to perform data logging using GX LogViewer to check operation.

This course uses the same one-axis ball screw mechanism as used in the "MELSEC MX Controller Basics (Motion Control Startup)" course.

The machine specifications are as follows.



The system configuration is as follows.

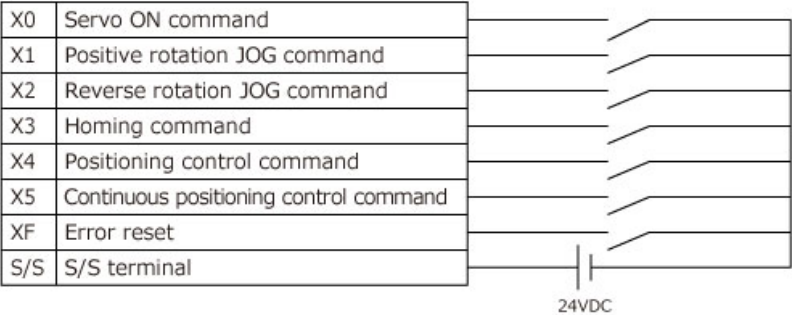
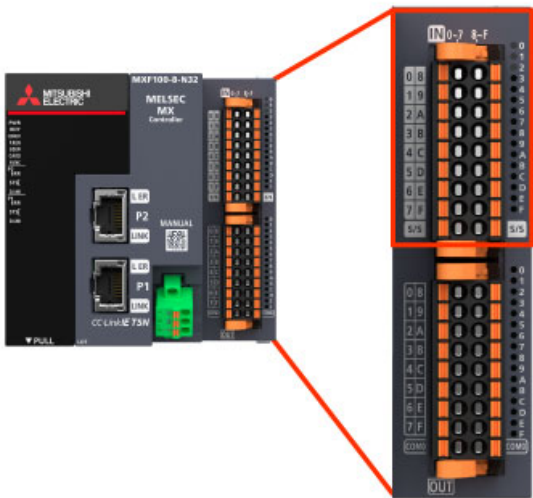


1.4

External Circuit Wiring

For details on the power supply wiring for the MELSEC MX Controller and servo amplifier and the connection of the servo motor, refer to the "MELSEC MX Controller Basics (Motion Control Startup)" course.

Wire the external circuit of the MELSEC MX Controller MX-F Model as shown below.



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

The "MELSEC MX Controller Basics (Motion Control Startup)" course covered the basics of positioning control. This course provides the detailed explanations.

This chapter describes buffer mode, which is one of the positioning functions.

To use the sample program for this chapter, download the following file.

[MX-F_positioning_1_en.zip \(1.99MB\)](#)

This function executes multiple motion control FBs continuously without stopping by executing the motion FB of another instance to the axis and the axes group that are executing the motion related motion control FB (hereinafter referred to as "motion FB").

The following table lists the buffer mode types.

Buffer mode setting value	Description
0 or mcAborting	Aborts (cancels) the FB being executed and executes the next FB immediately.
1 or mcBuffered	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 or mcBlendingLow	The lower target velocity between the FB being executed and the buffering FB is used as the switching velocity.
3 or mcBlendingPrevious	The target velocity of the FB being executed is used as the switching velocity.
4 or mcBlendingNext	The target velocity of the buffering FB is used as the switching velocity.
5 or mcBlendingHigh	The higher target velocity between the FB being executed and the buffering FB is used as the switching velocity.

Specify a buffer mode using the BufferMode input of a motion FB.

This section describes how to specify a buffer mode using one of the motion FBs, MC_MoveAbsolute, as an example.

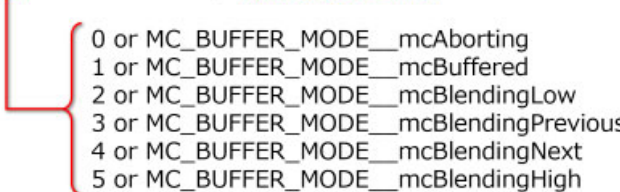
Note that the buffer modes that can be specified differs depending on the FB used.

For MC_MoveAbsolute, six buffer modes can be specified.

For details on the buffer modes available for each FB, refer to the manual.

Example: MC_MoveAbsolute

```
MC_MoveAbsolute_1(
  Axis      := Axis0001.AxisRef ,
  Execute   := G_bPositioningReq ,
  Position  := lePosition1 ,
  Velocity  := lePosVelocity ,
  Acceleration := lePosAcceleration ,
  Deceleration := lePosDeceleration ,
  Jerk      := lePosJerk ,
  Direction := MC_DIRECTION_mcShortestWay ,
  BufferMode := MC_BUFFER_MODE_mcAborting ,
  Done      => bMoveAbslDone ,
  CommandAborted => bMoveAbslAborted ,
  Error      => bMoveAbslError
);
```

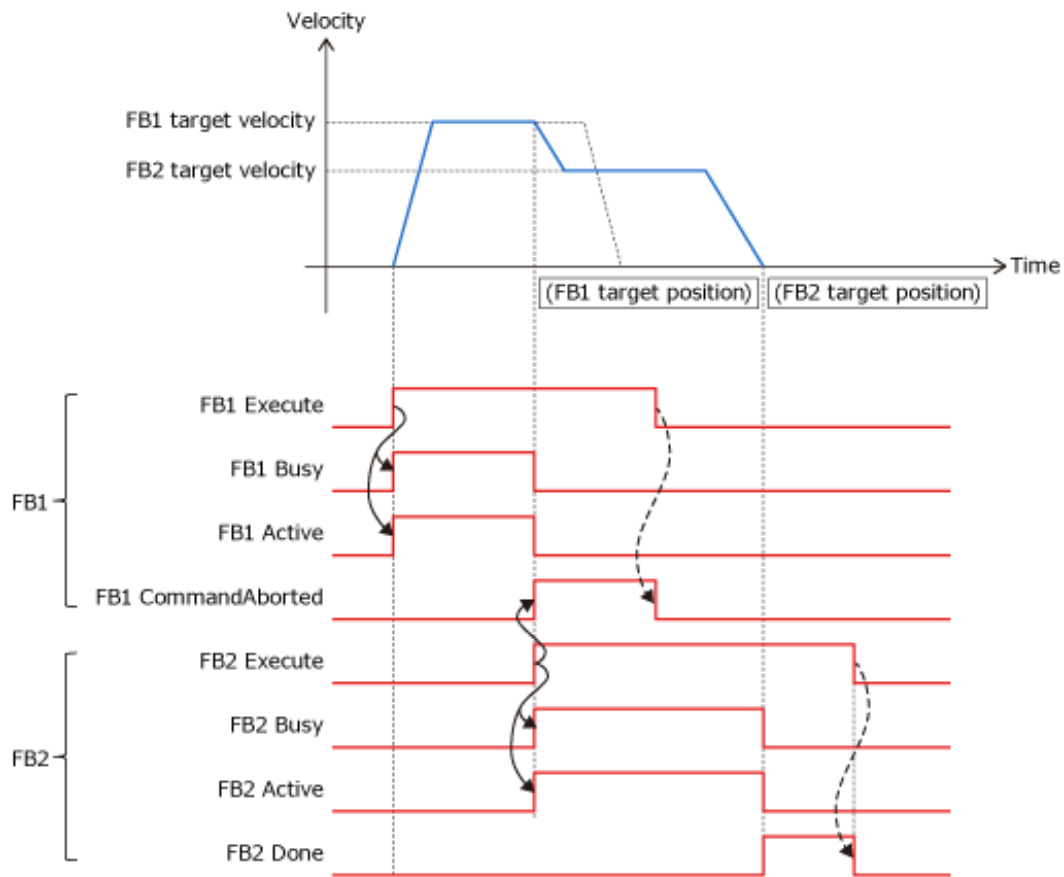


- 0 or MC_BUFFER_MODE__mcAborting
- 1 or MC_BUFFER_MODE__mcBuffered
- 2 or MC_BUFFER_MODE__mcBlendingLow
- 3 or MC_BUFFER_MODE__mcBlendingPrevious
- 4 or MC_BUFFER_MODE__mcBlendingNext
- 5 or MC_BUFFER_MODE__mcBlendingHigh

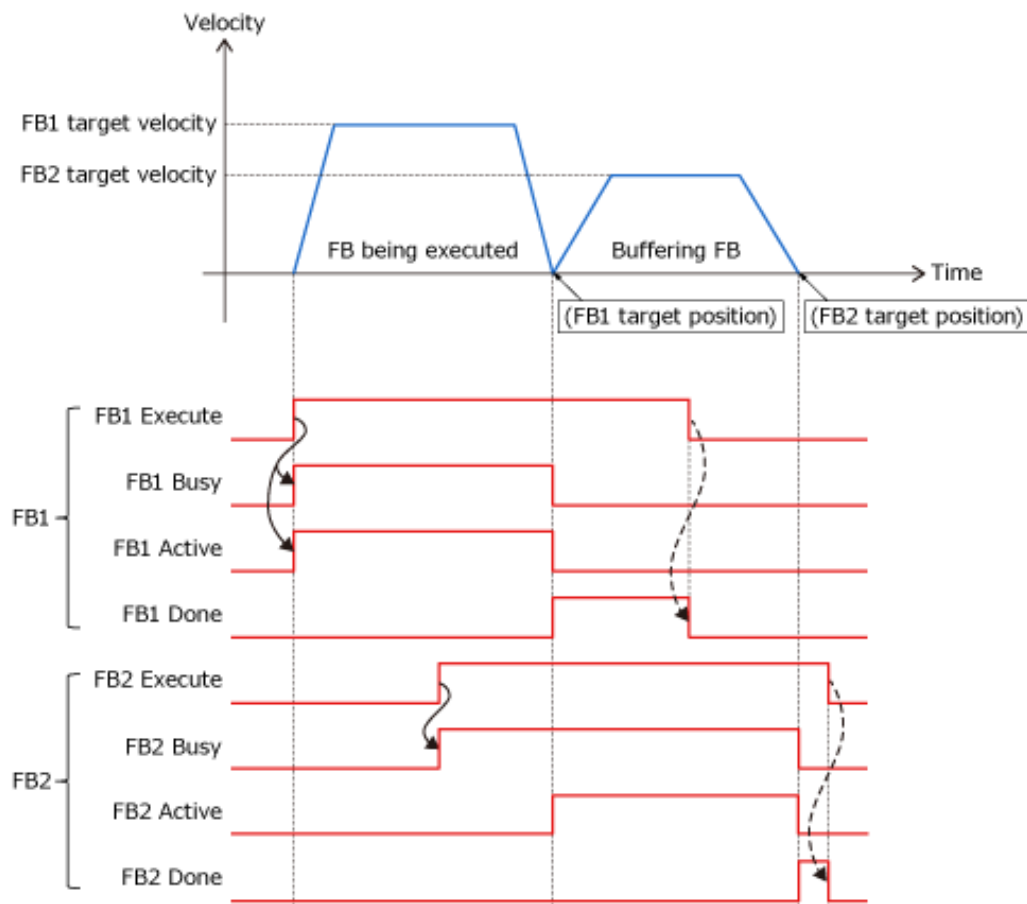
[Point]

For the Direction and BufferMode inputs, specify either a numeric value or an ENUM enumerator that starts with MC_BUFFER_MODE_ or MC_DIRECTION_.

The following diagram shows the operation when BufferMode is set to 0: mcAborting. This mode aborts the FB being executed and executes the next FB immediately.



The following diagram shows the operation when BufferMode is set to 1: mcBuffered.
This mode executes the next FB after the FB being executed is completed.



When BufferMode is set to mcBlending***, this mode executes the next FB continuously after the target position of the FB being executed is reached.

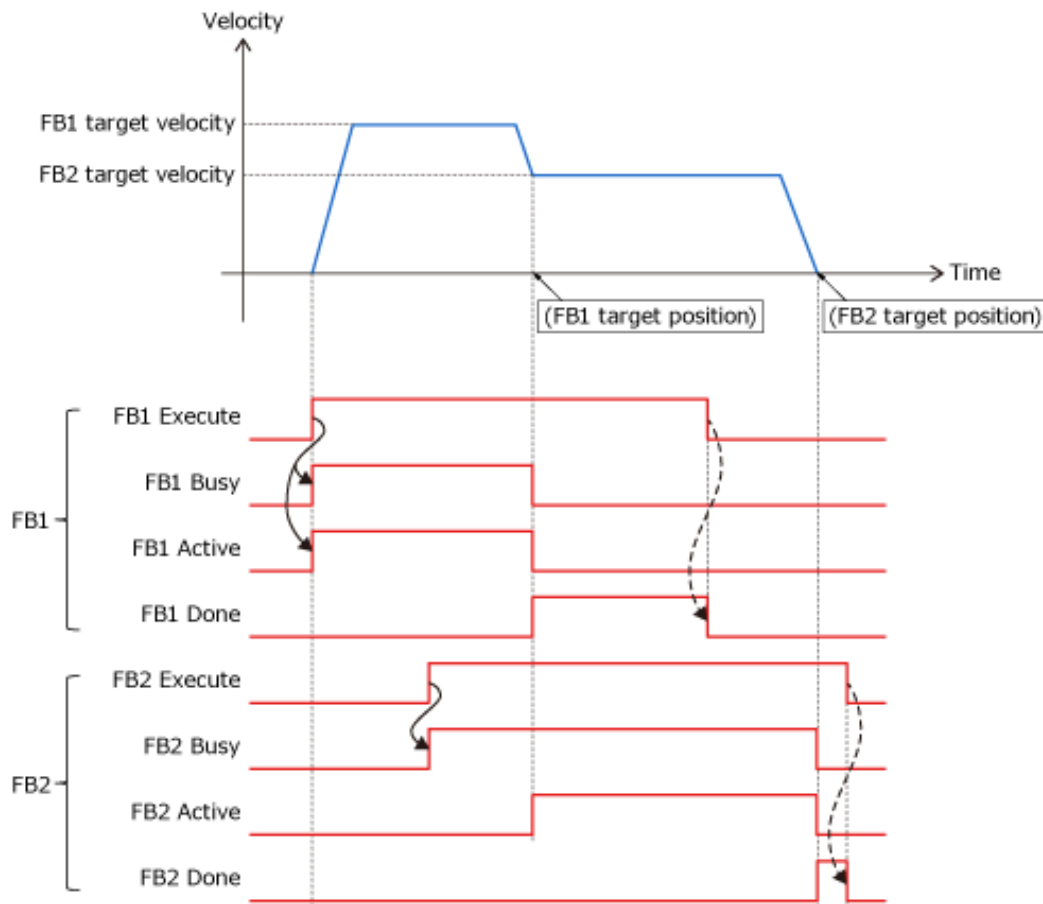
In the following description, the FB to be executed first is FB1, and FB to be buffered is FB2.

(1) BlendingPrevious

The following diagram shows the operation when BufferMode is set to 3: mcBlendingPrevious.

The axis moves at the target velocity of FB1 until it reaches the target position of FB1.

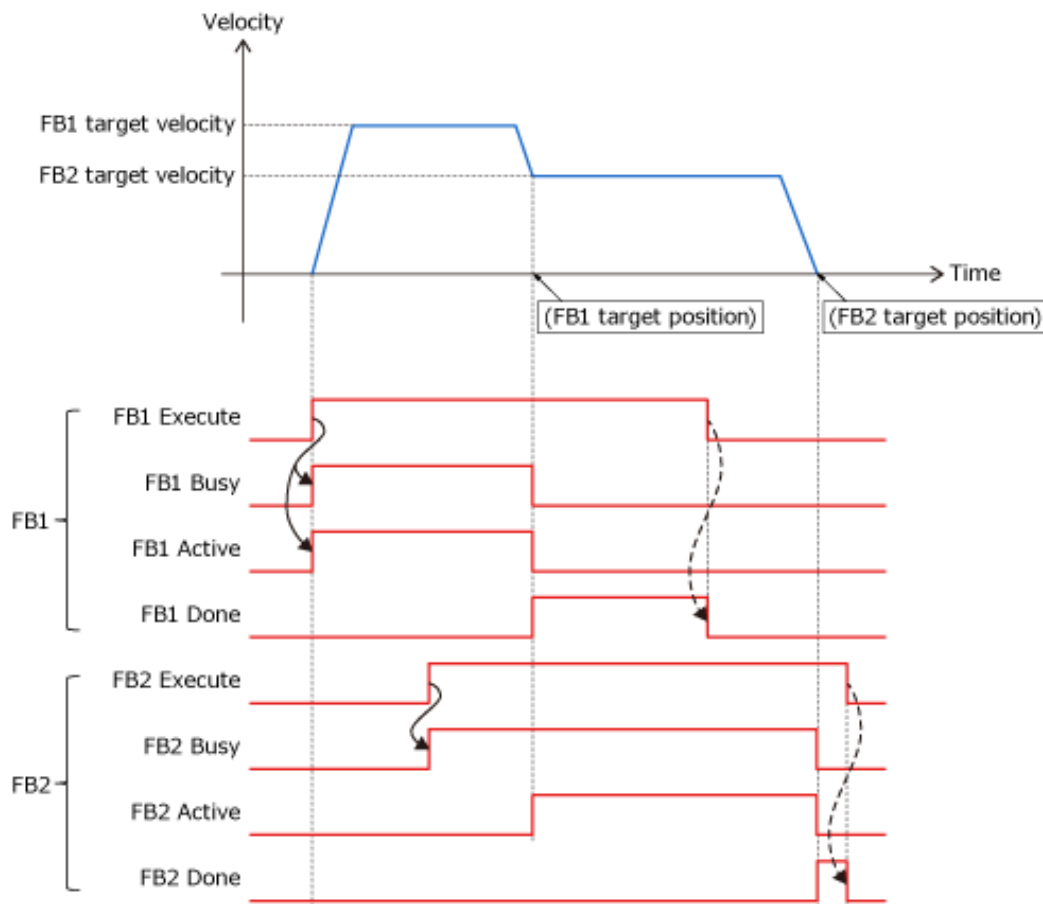
When operation switches to FB2, the velocity is changed to the target velocity of FB2, and the axis moves to the target position of FB2.



(2) BlendingNext

The following diagram shows the operation when BufferMode is set to 4: mcBlendingNext.

The axis is controlled so that it reaches the target velocity of FB2 when it reaches the target position of FB1.



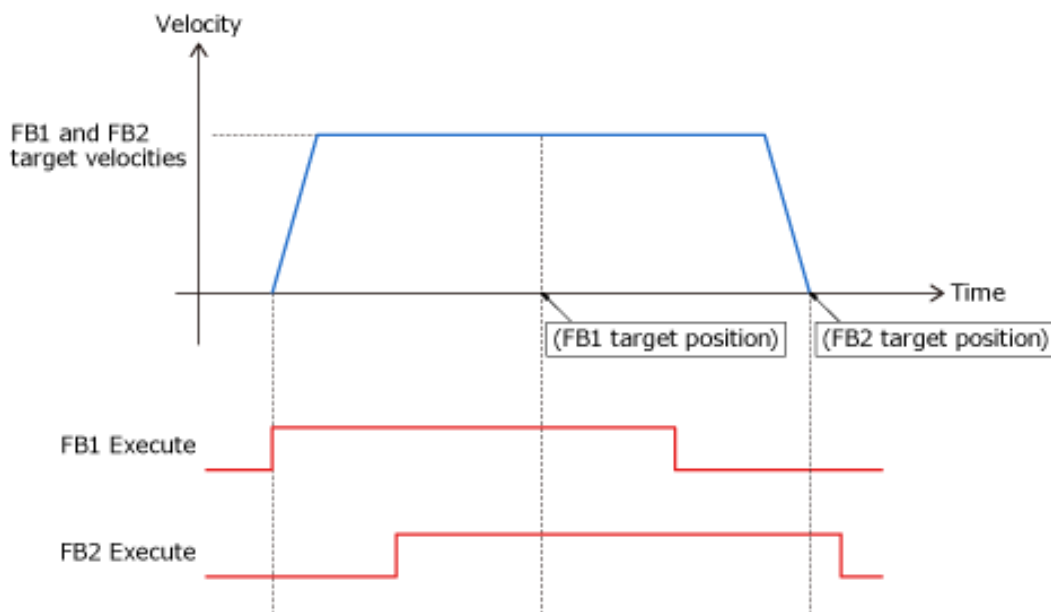
(3) BlendingLow, BlendingHigh

The operation when BufferMode is set to 2: mcBlendingLow or 5: mcBlendingHigh varies depending on the relative values of the target velocities of FB1 and FB2.

Setting value	FB1 target velocity > FB2 target velocity	FB1 target velocity < FB2 target velocity
2: mcBlendingLow	Same operation as BlendingPrevious	Same operation as BlendingNext
5: mcBlendingHigh	Same operation as BlendingNext	Same operation as BlendingPrevious

[Point]

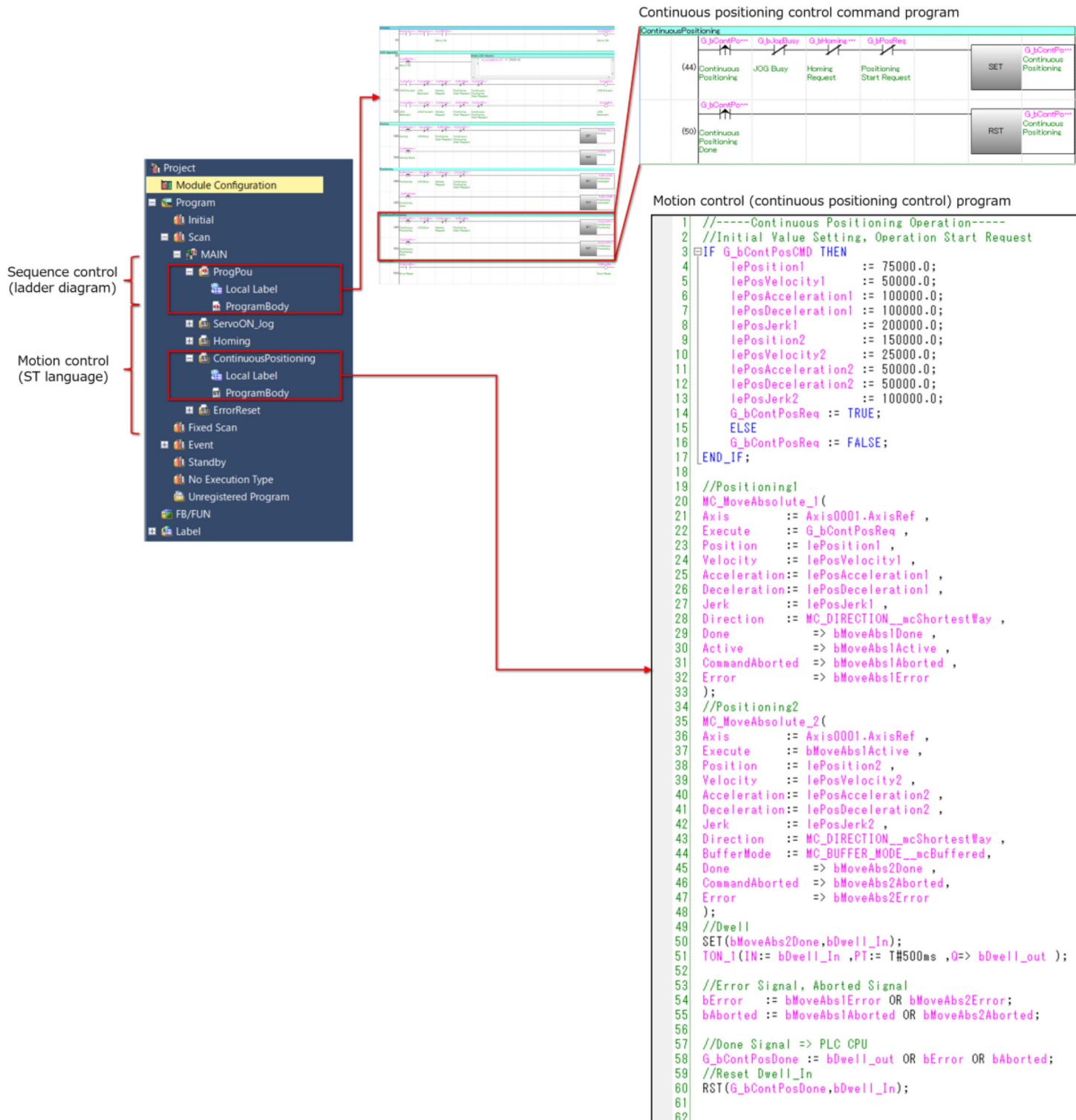
When the target velocities of FB1 and FB2 are the same, the velocity waveforms for BlendingPrevious, BlendingNext, BlendingHigh, and BlendingLow are as shown in the figure below.



This section describes the sample program configuration.

MX-F_positioning_1.gx3 provides the sequence control (ladder diagram) program and the motion control (ST language) program separately.

The programs shown below are used to explain the buffer modes in this chapter. The other programs operate in the same way as the sample program used in the "MELSEC MX Controller Basics (Motion Control Startup)" course, so their explanations are omitted in this course.



(1) Sample program operation

Use the program started by the continuous positioning control command (X5) to compare how each buffer mode operates.

Item	FB1 (MC_MoveAbsolute)	FB2 (MC_MoveAbsolute)
Positioning address	75000.0 [μm]	150000.0 [μm]
Velocity	50000.0 [$\mu\text{m}/\text{s}$]	25000.0 [$\mu\text{m}/\text{s}$]
Acceleration, deceleration	100000.0 [$\mu\text{m}/\text{s}^2$]	50000.0 [$\mu\text{m}/\text{s}^2$]
Jerk	200000.0 [$\mu\text{m}/\text{s}^3$]	100000.0 [$\mu\text{m}/\text{s}^3$]

(2) Continuous positioning control command program



The rising edge of the continuous positioning control command signal (G_bContPosCM: X5) is latched by G_bContPosCMD and sent to the motion control (continuous positioning control) program as the start condition for continuous positioning control. An interlock is applied to prevent positioning control from starting while other programs are running. When the motion control (continuous positioning control) program detects that the continuous positioning control completion signal turns on, G_bContPosCMD is reset on the rising edge of the signal.

(3) Motion control (continuous positioning control) program

```

1  //-----Continuous Positioning Operation-----
2  //Initial Value Setting, Operation Start Request
3  IF G_bContPosCMD THEN
4      lePosition1      := 75000.0;
5      lePosVelocity1   := 50000.0;
6      lePosAcceleration1 := 100000.0;
7      lePosDeceleration1 := 100000.0;
8      lePosJerk1       := 200000.0;
9      lePosition2      := 150000.0;
10     lePosVelocity2   := 25000.0;
11     lePosAcceleration2 := 50000.0;
12     lePosDeceleration2 := 50000.0;
13     lePosJerk2       := 100000.0;
14     G_bContPosReq := TRUE;
15 ELSE
16     G_bContPosReq := FALSE;
17 END_IF;
18
19 FB1
20 //Positioning1
21 MC_MoveAbsolute_1(
22     Axis      := Axis0001.AxisRef ,
23     Execute   := G_bContPosReq ,
24     Position  := lePosition1 ,
25     Velocity  := lePosVelocity1 ,
26     Acceleration:= lePosAcceleration1 ,
27     Deceleration:= lePosDeceleration1 ,
28     Jerk      := lePosJerk1 ,
29     Direction := MC_DIRECTION__mcShortestWay ,
30     Done      => bMoveAbs1Done ,
31     Active    => bMoveAbs1Active ,
32     CommandAborted => bMoveAbs1Aborted ,
33     Error     => bMoveAbs1Error
34 );
35
36 FB2
37 //Positioning2
38 MC_MoveAbsolute_2(
39     Axis      := Axis0001.AxisRef ,
40     Execute   := bMoveAbs1Active ,
41     Position  := lePosition2 ,
42     Velocity  := lePosVelocity2 ,
43     Acceleration:= lePosAcceleration2 ,
44     Deceleration:= lePosDeceleration2 ,
45     Jerk      := lePosJerk2 ,
46     Direction := MC_DIRECTION__mcShortestWay ,
47     BufferMode := MC_BUFFER_MODE_mcBuffered ,
48     Done      => bMoveAbs2Done ,
49     CommandAborted => bMoveAbs2Aborted ,
50     Error     => bMoveAbs2Error
51 );
52
53 //Dwell
54 SET bMoveAbs2Done, bDwell_In;
55 TON_T(IN:= bDwell_In ,PT:= T#100ms ,Q=> bDwell_out);
56
57 //Error Signal, Aborted Signal
58 bError := bMoveAbs1Error OR bMoveAbs2Error;
59 bAborted := bMoveAbs1Aborted OR bMoveAbs2Aborted;
60
61 //Done Signal => PLC CMD
62 G_bContPosDone := bDwell_out OR bError OR bAborted;
63 //Reset Dwell_In
64 RST(G_bContPosDone, bDwell_In);
65

```

The continuous positioning control start signal (G_bContPosCMD) is received from the continuous positioning control command program. The data required for positioning is stored in labels, and the continuous positioning control start request (G_bContPositioningReq) is turned on.

When the continuous positioning control command program turns off G_bContPosCMD, G_bContPositioningReq is turned off.

Positioning is performed by FB1 (MC_MoveAbsolute_1).

FB2 (MC_MoveAbsolute_2) is started by the Active output of FB1 (MC_MoveAbsolute_1). This allows FB2 to be buffered while FB1 is running.

Change the BufferMode input of FB2 (MC_MoveAbsolute_2) to check the buffer mode operation.

The on-delay timer used as the dwell timer is started by the Done output of FB2 (MC_MoveAbsolute_2).

After the dwell time has elapsed, or when either the Error output or CommandAborted output of MC_MoveAbsolute turns on, the execution completion signal is returned to the continuous positioning control command program. At the same time, the input of the on-delay timer is reset.

(1) How to convert the created programs into executable codes

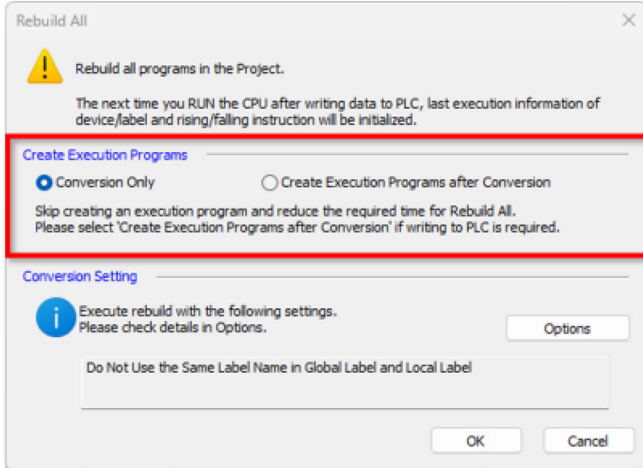
Data names displayed in red in the Navigation window have not yet been converted and must be converted.

(2) Creating an execution program

When converting any or all programs, whether to create an execution program can be selected.

To write data to the MELSEC MX Controller, select the checkbox of "Create Execution Programs after Conversion".

When selecting the checkbox of "Conversion Only", the required time for the conversion can be shortened.

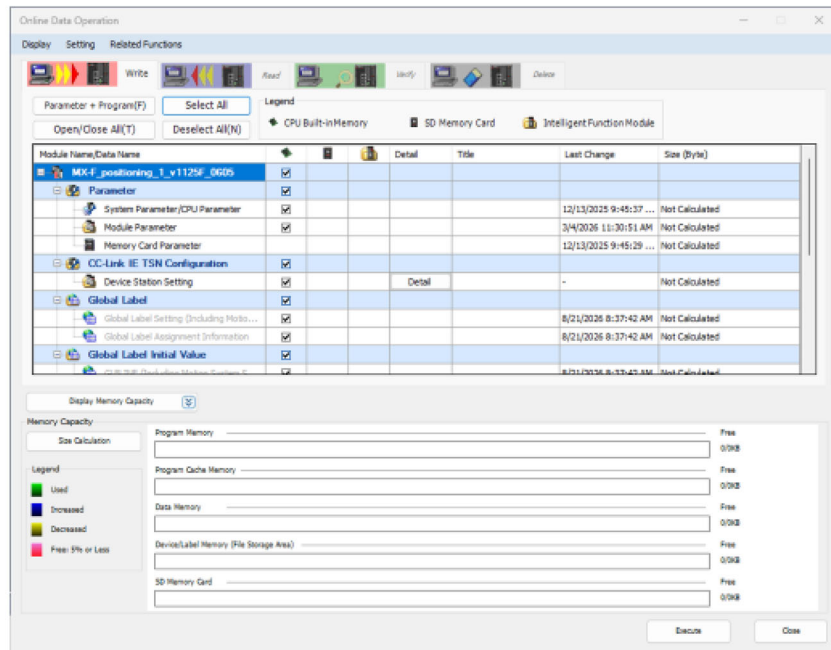
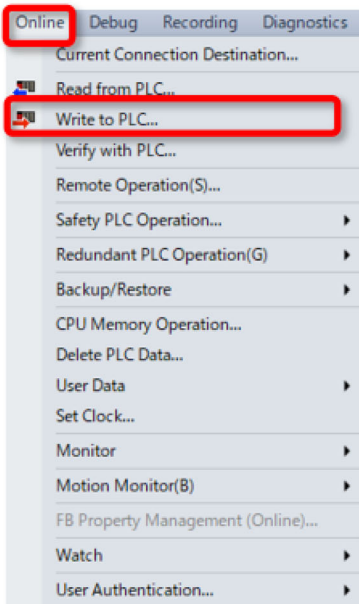


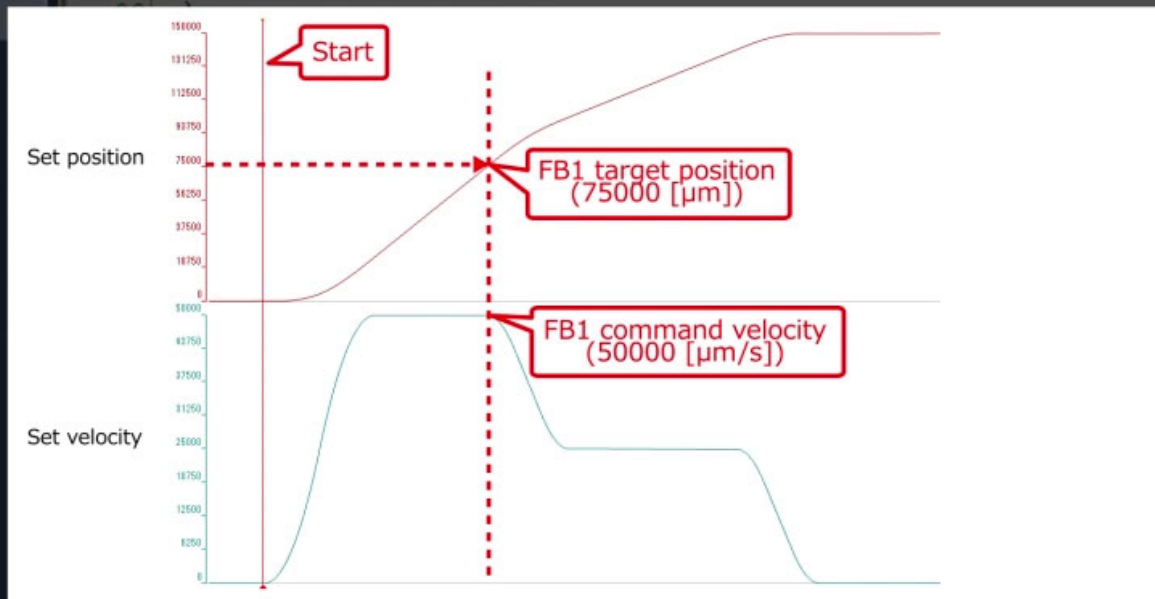
(3) Writing programs

1) Set the MELSEC MX Controller to the STOP state.

2) Select [Online] → [Write to PLC], and click [Select All] on the Write tab in the "Online Data Operation" window.

3) Click the [Execute] button to write data.





This completes the operation check for BufferMode.

```
50 SET(bMoveAbs2Done,bDwell_In);  
58 G_bContPosDone := bDwell_out OR bError OR bAborted;  
59 /DwellOut := bDwell_out;
```

In this chapter, you have learned:

- Multiple Start (Buffer Mode)
- Aborting
- Buffered
- Blending
- Example Programs
- Operation Check

Important points

Aborting	• When the next motion FB is executed while the previous motion FB is still running, this mode aborts the currently running FB and executes the next FB.
Buffered	• When the next motion FB is executed while the previous motion FB is still running, this mode waits until the currently running FB is complete, and then executes the next FB.
Blending	• When the next motion FB is executed while the previous motion FB is still running, this mode executes the next FB without stopping the previous motion FB. • There are four types of Blending, depending on the velocity switching method: BlendingLow, BlendingHigh, BlendingPrevious, and BlendingNext.
Example Programs	• Select a buffer mode using the BufferMode input of a motion FB.
Operation Check	• Watch the video to see how each buffer mode operates differently.

Download the sample program to be used in this chapter from the link below.

This program is created in the FBD/LD language, which makes it easier to understand the connections between FBs. It performs the same continuous positioning control described in Chapter 2.

[MX-F_positioning_2_en.zip \(1.85MB\)](#)

The procedure for creating a project is the same as the one described in the "MELSEC MX Controller Basics (Motion Control Startup)" course.

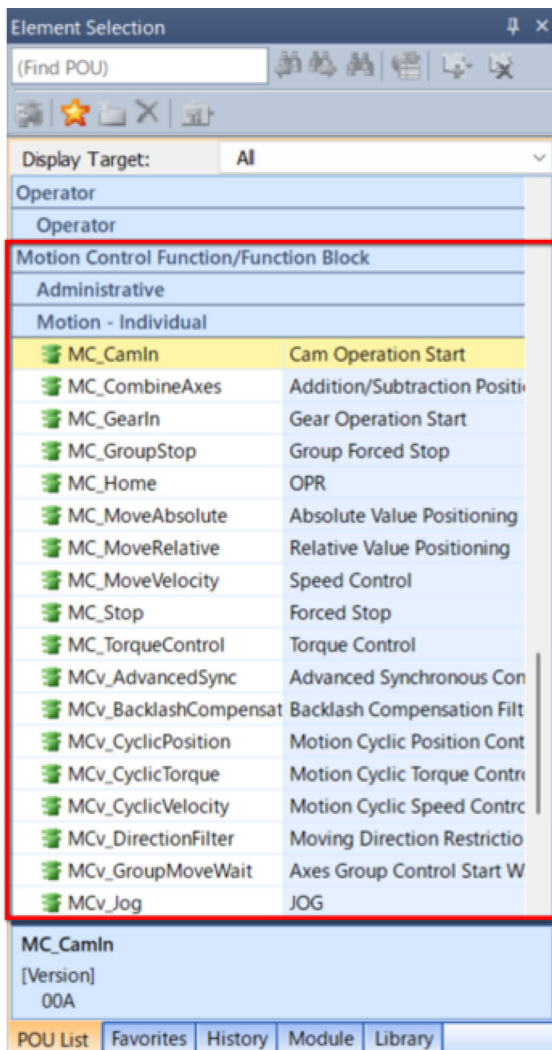
GX Works3

- Creating a module configuration
- Setting module parameters (networks)
 - Registering device stations (amplifiers)
- PDO mapping
- Setting servo parameters
- Registering axes
- Setting axis parameters

In this chapter, the MELSEC MX Controller program is created in the FBD/LD language which makes it easier to understand the connections between FBs.

(1) FB Library

The motion control FB library can be used by selecting it from the Element Selection window.



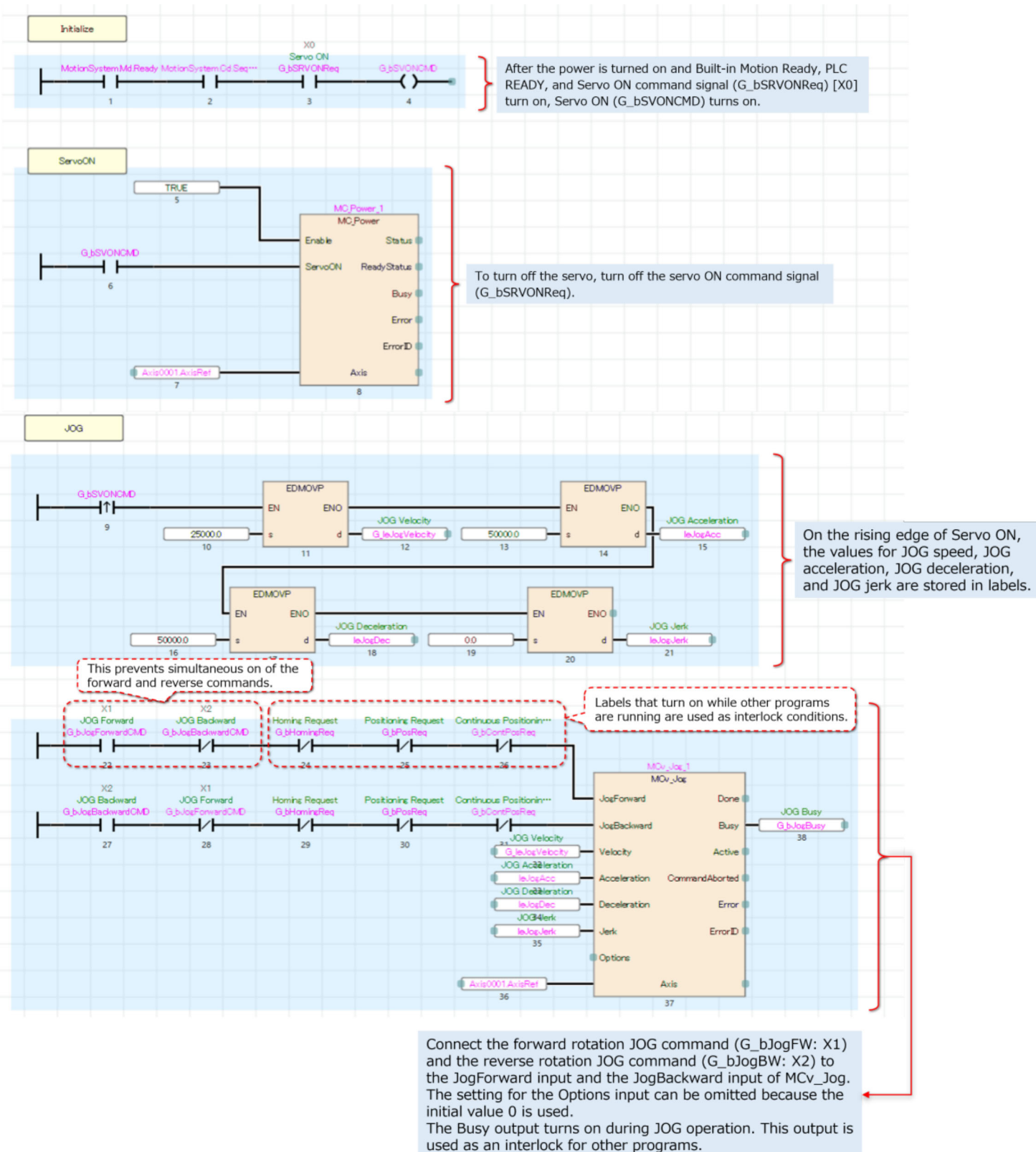
(2) The following table lists the FB libraries. FBs are classified into the following three groups.

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

The screenshot displays the MELSOFT GX Works3 interface. The left sidebar shows the project structure, including 'MAIN', 'ServoON_JOG', and 'ProgramBody'. The main workspace shows a ladder logic program with 16 steps. Steps 1-5 are connected in series, leading to a function block 'MOv_Jog'. Steps 6-10 are also connected in series, leading to the same function block. Steps 11-16 are connected in series, leading to the same function block. The function block 'MOv_Jog' has inputs for 'JOG Velocity' (11), 'JOG Acceleration' (12), 'JOG Deceleration' (13), 'JOG Jerk' (14), 'Options' (15), and 'Axis' (16). The output of the function block is 'JOG Busy' (18). A text box at the bottom right states: 'The program has been entered. Click [Next] to proceed to the next page.'

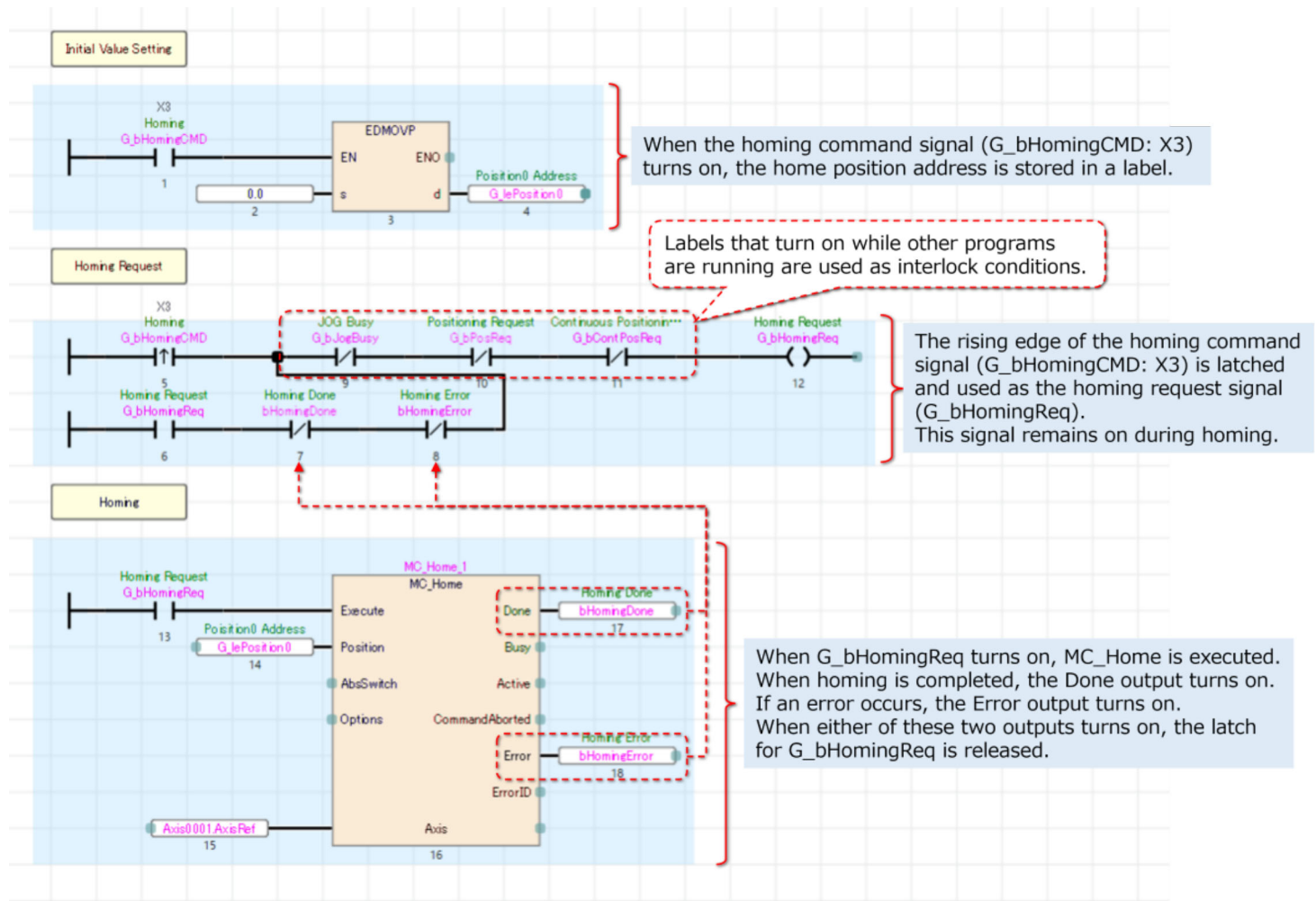
(1) ServoON_Jog

This program turns on the servo and performs JOG operation when PLC READY turns on.



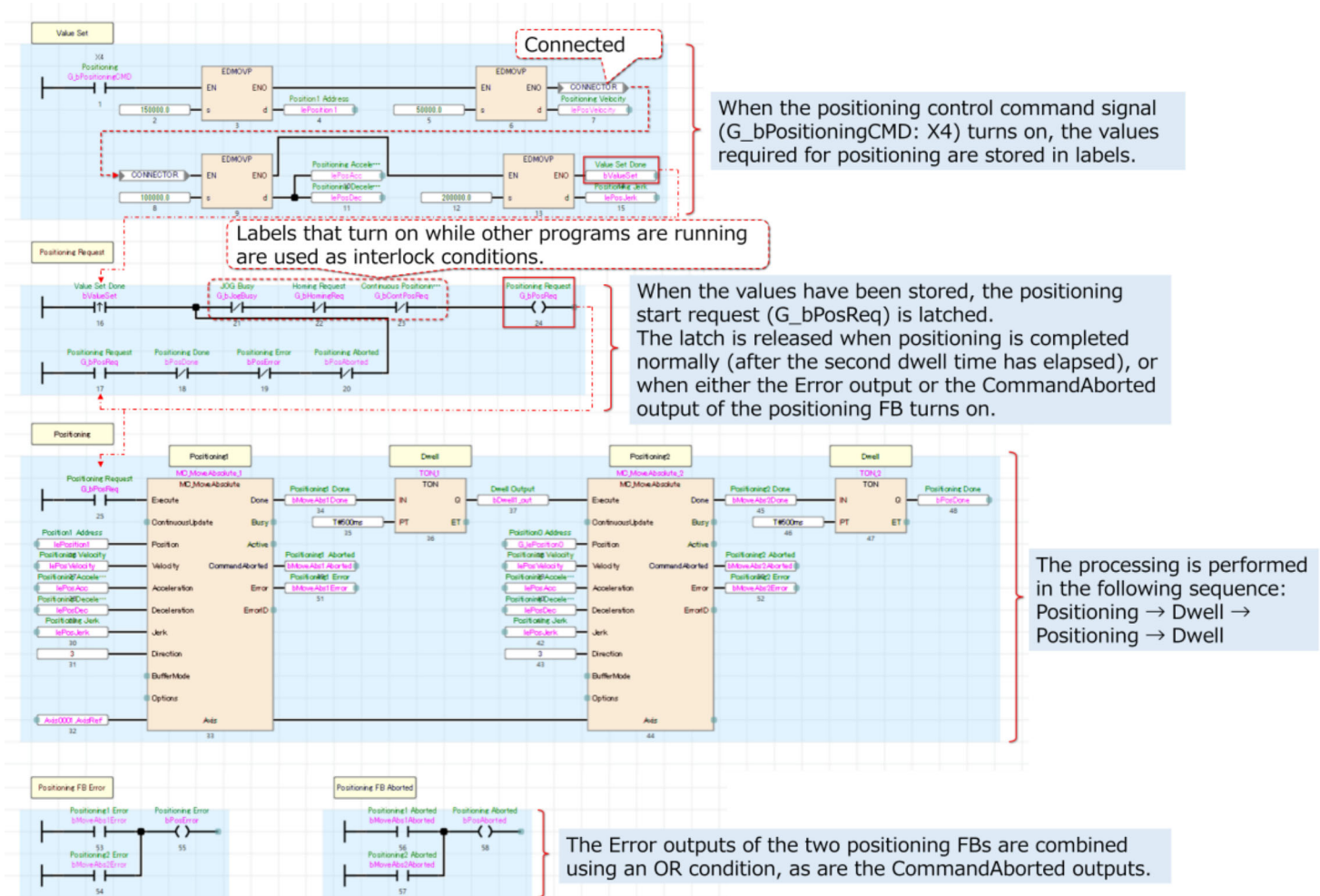
(2) Homing

This program performs homing.



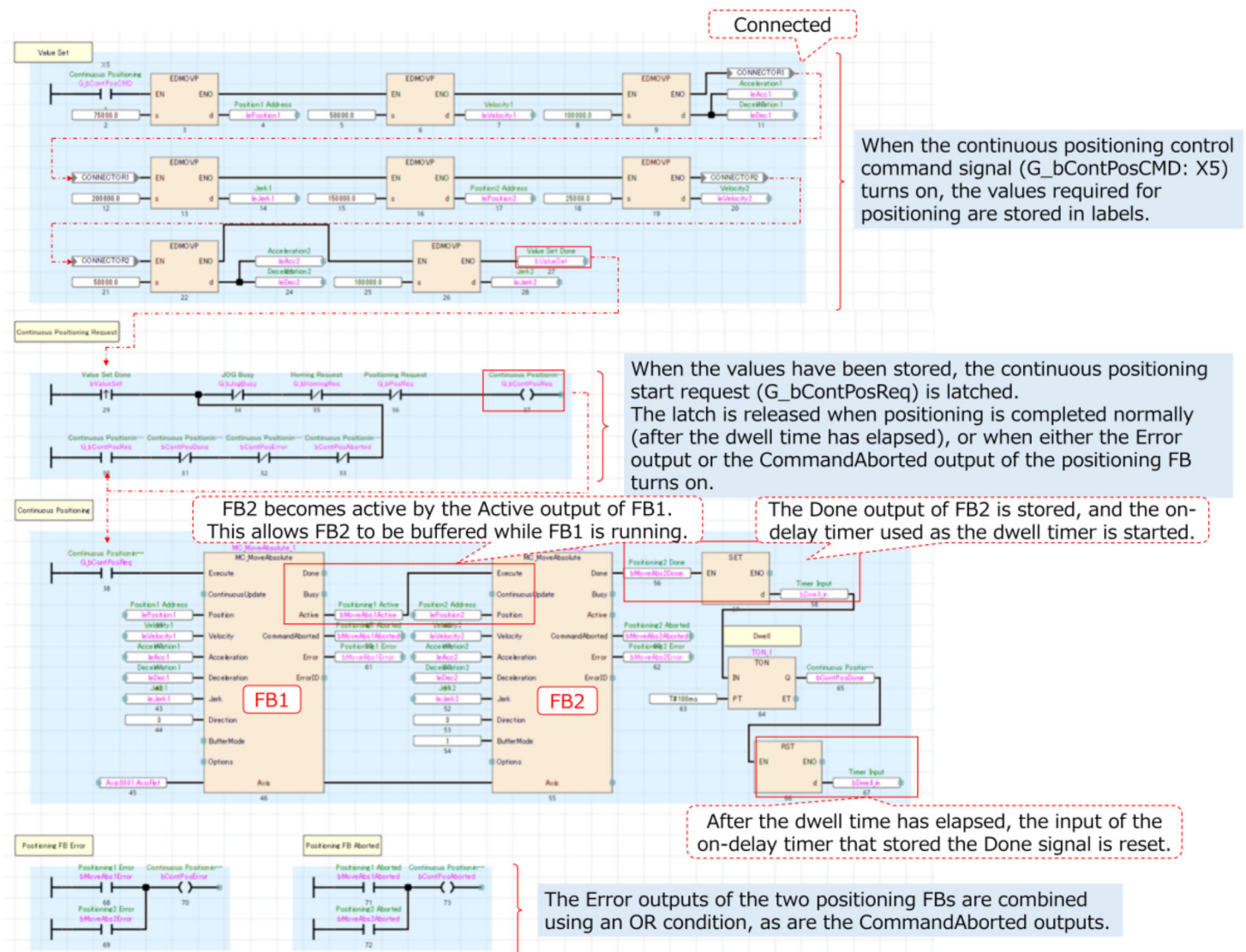
(3) Positioning

This program performs single axis positioning.



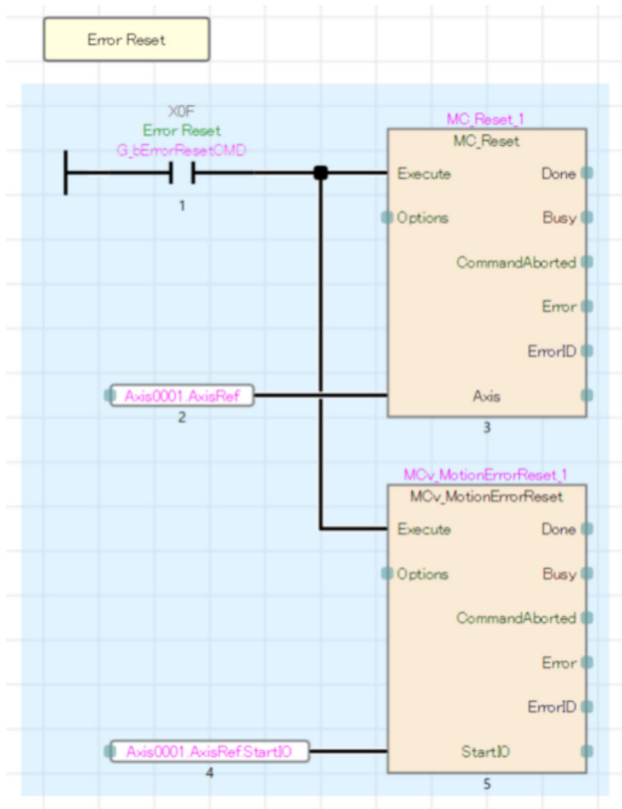
(4) ContinuousPositioning

This program performs continuous positioning in buffer mode.



(5) ErrorReset

This program resets an error.



When the error reset signal (G_bErrorResetCMD: XF) turns on, Axis Error Reset (MC_Reset) and Motion Error Reset (MCv_MotionErrorReset) are executed.

(6) Monitor

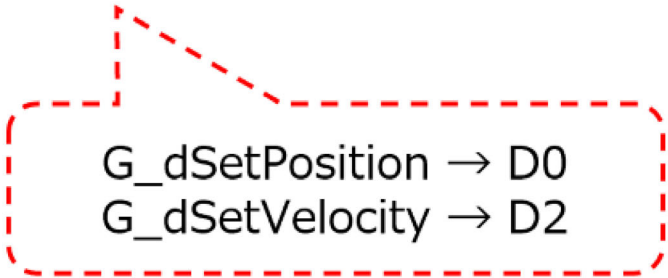
This program stores SetPosition (Set Position) and SetVelocity (Set Velocity) of the axis monitor data into global labels assigned to D0 and D2 of the MELSEC MX Controller.

Since SetPosition and SetVelocity are double-precision real numbers, convert them to the signed double word values, which are easier to handle on the MELSEC MX Controller.

Note: Although these word device areas are not used in this exercise, they are used when displaying the values in other sequence programs or on a GOT.

Only this program is written in the ST language.

```
1 G_dSetPosition := LREAL_TO_DINT(Axis0001.Md.SetPosition);  
2 G_dSetVelocity := LREAL_TO_DINT(Axis0001.Md.SetVelocity);
```

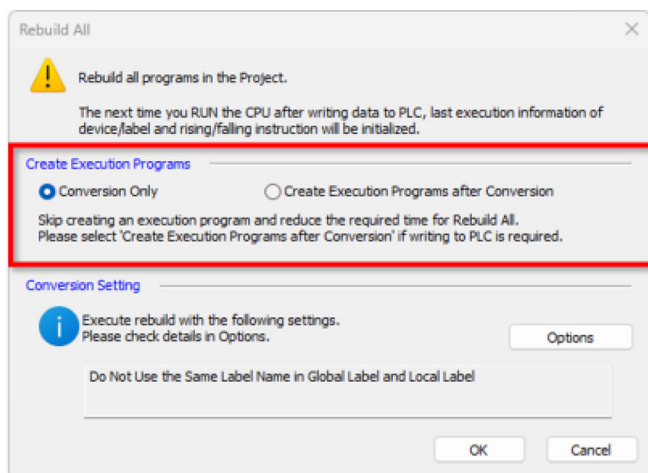


G_dSetPosition → D0
G_dSetVelocity → D2

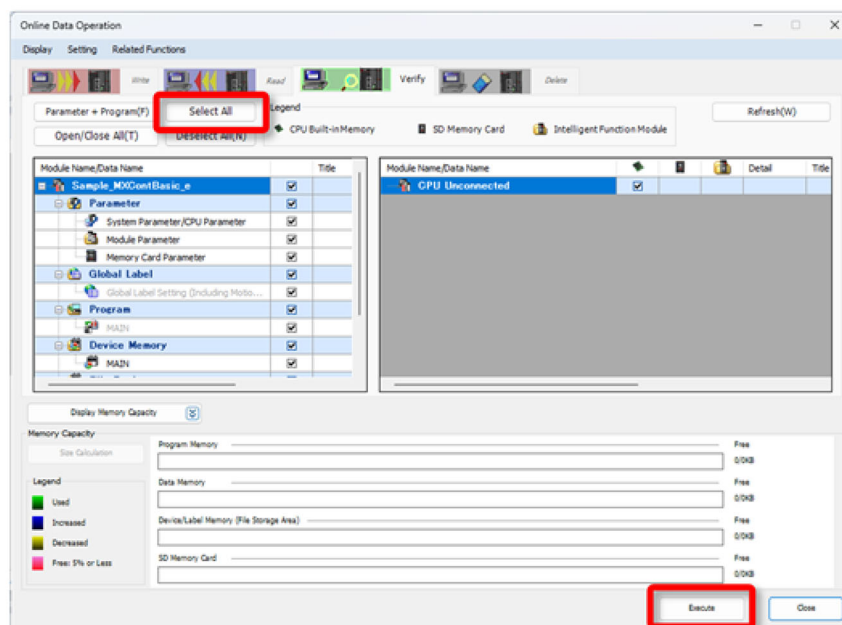
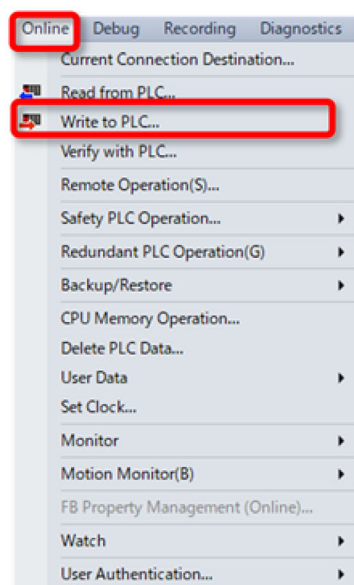
Note: When converting a double-precision real number to a signed double word value, an operation error occurs if the source value is outside the range of -2147483648 to 2147483647.

Write the programs to the MELSEC MX Controller using the same procedure as in Chapter 2.

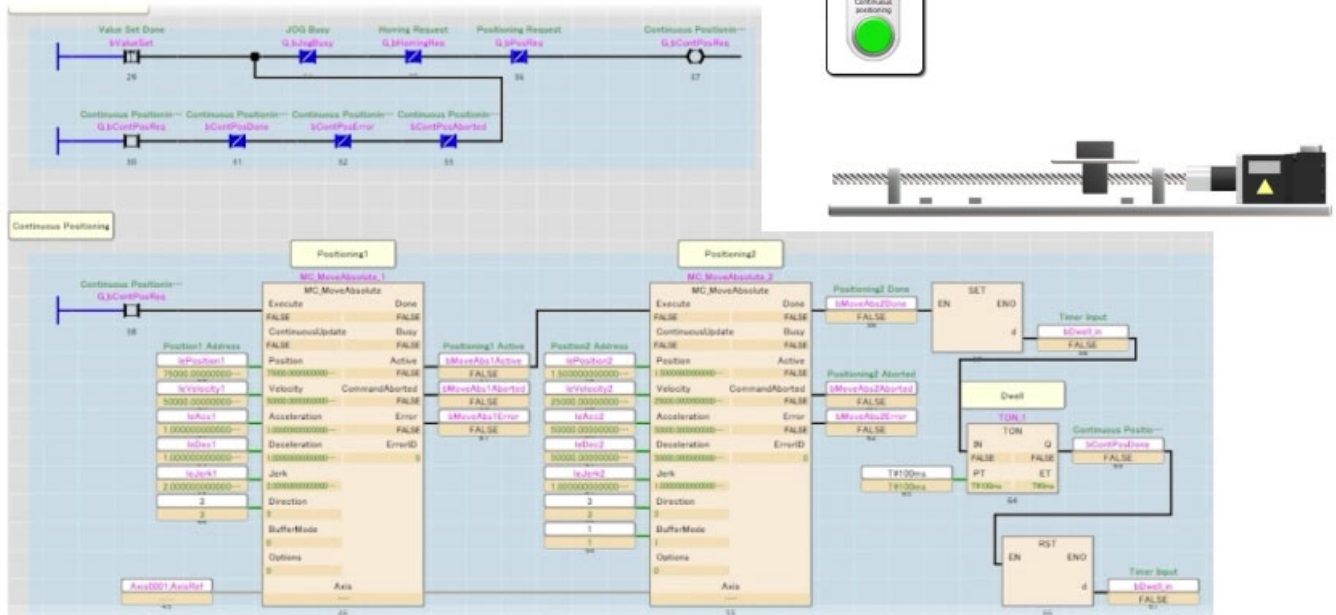
(1) Converting the programs and creating an execution program



(2) Writing programs



MELSEC MX Controller



The operation has been checked.
Go to the next page.

In this chapter, you have learned:

- How to Use Motion Control FBs
- Sample Program
- Sample Program Operation Check

Important points

How to Use Motion Control FBs	• Place a motion control FB in the program editor by dragging and dropping it from the library in the Element Selection window of GX Works3. • Connect contacts and labels to the inputs and outputs of the FB.
Sample Program	• Create a program in the FBD/LD language that performs the same operation as the sample program in Chapter 2.
Sample Program Operation Check	• Watch the video to check the sample program operation.

This chapter describes how to log various data using CPU Module Logging Configuration Tool*¹ and display the data in graphs. In this chapter, the sample program used in Chapter 2 is used again. The operation of the positioning start program is logged. MX-F_positioning_1.zip

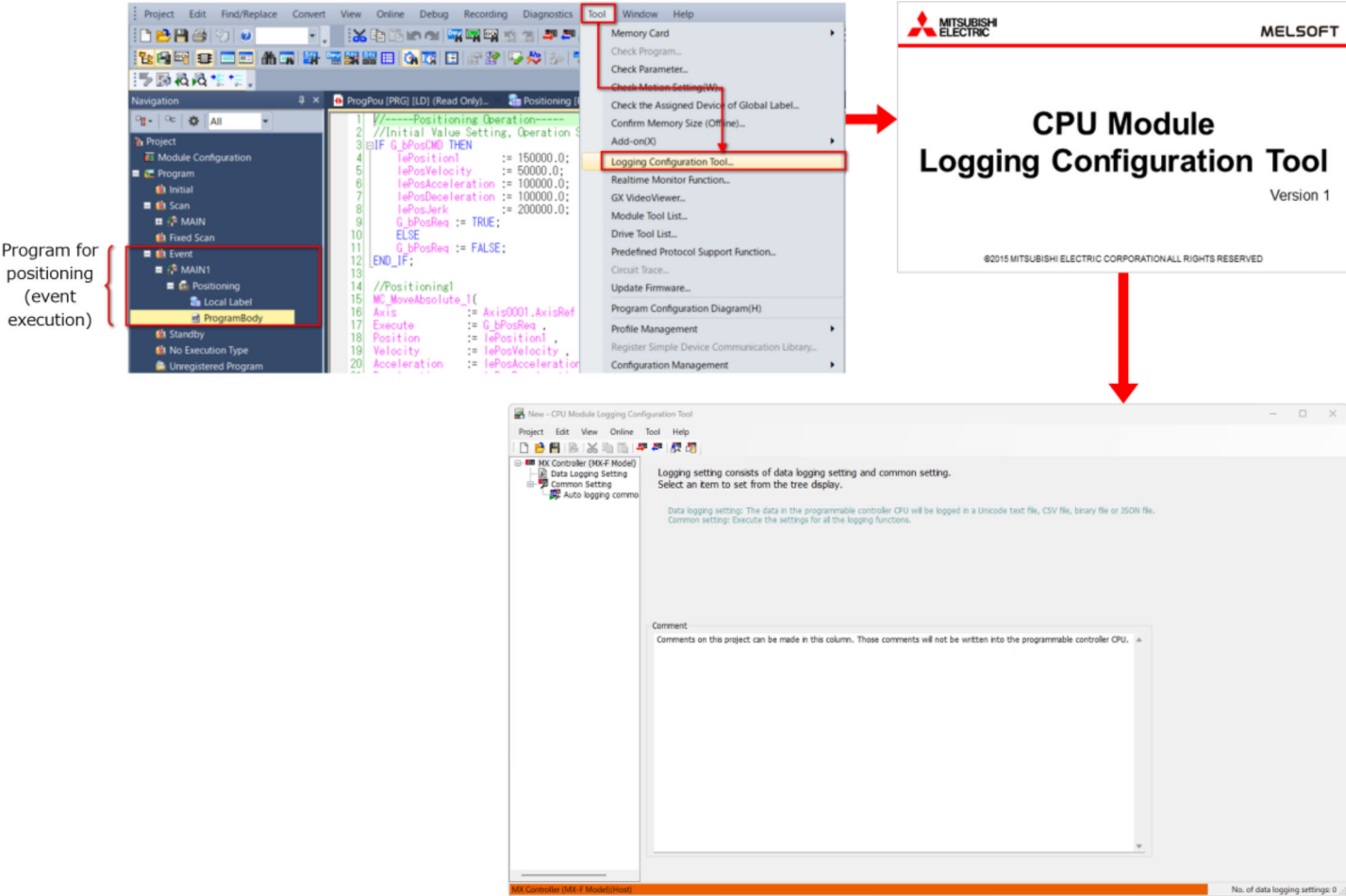
MX-F_positioning_1.zip (2.0MB)

*1: Before using the tool, download the latest version of the software from the Mitsubishi Electric FA Website.
www.mitsubishielectric.com/fa

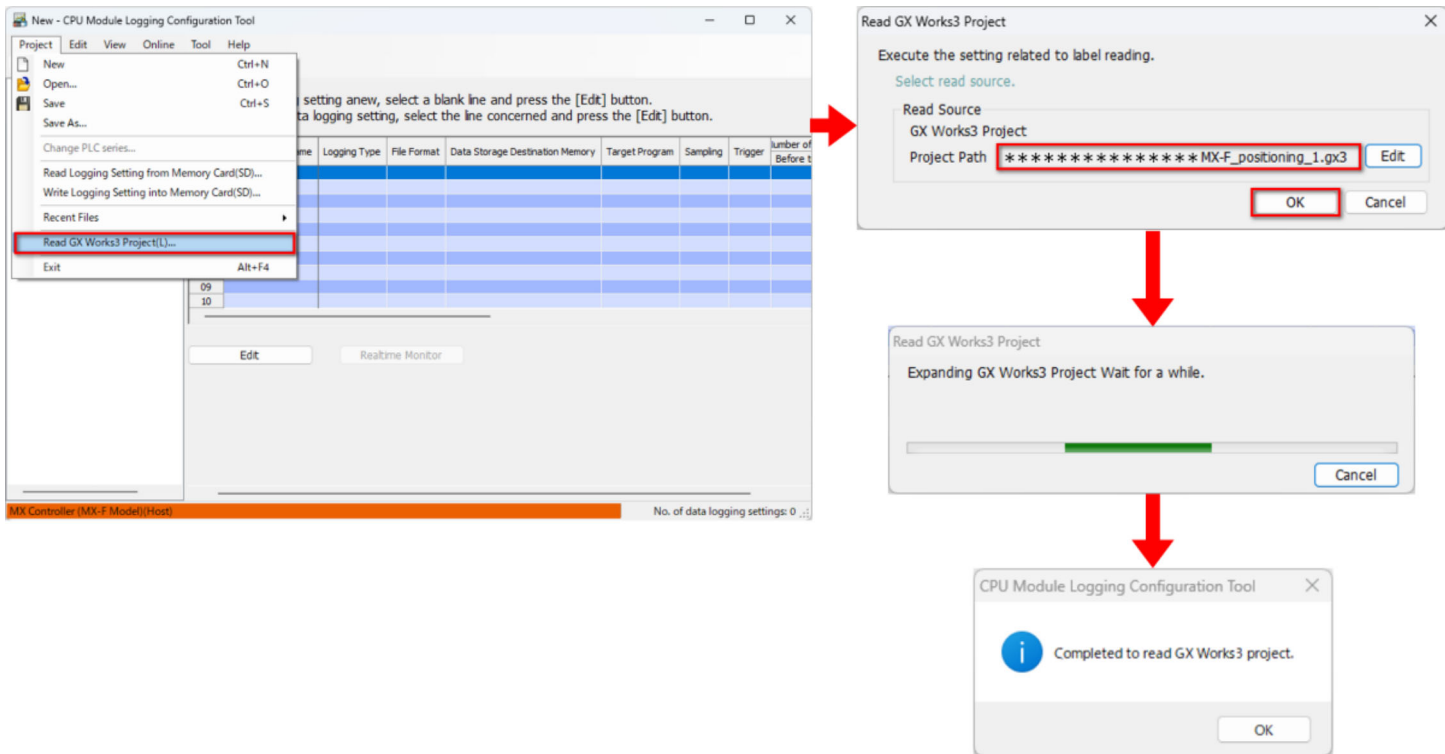
Product name	Description	Version
CPU Module Logging Configuration Tool	Software for data logging setting and maintenance	"1.178Y" or later
GX LogViewer	An easy-to-use tool for displaying and analyzing large-capacity data	"1.178Y" or later

4.1 **Launching CPU Module Logging Configuration Tool**

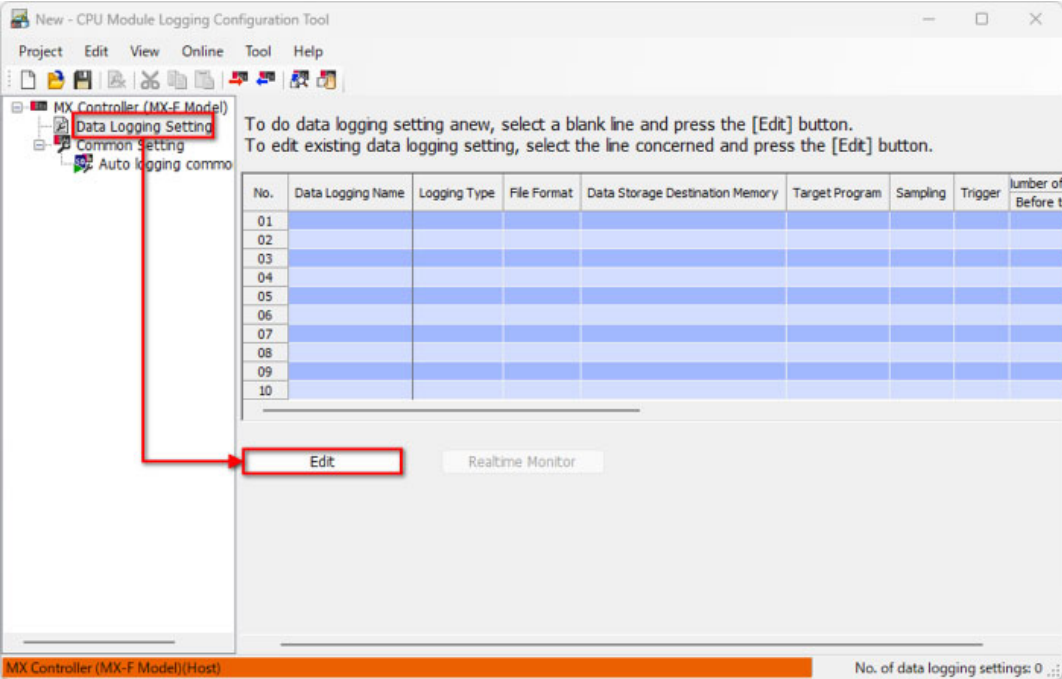
- (1) In GX Works3, select [Tool] → [Logging Configuration Tool].
CPU Module Logging Configuration Tool starts.



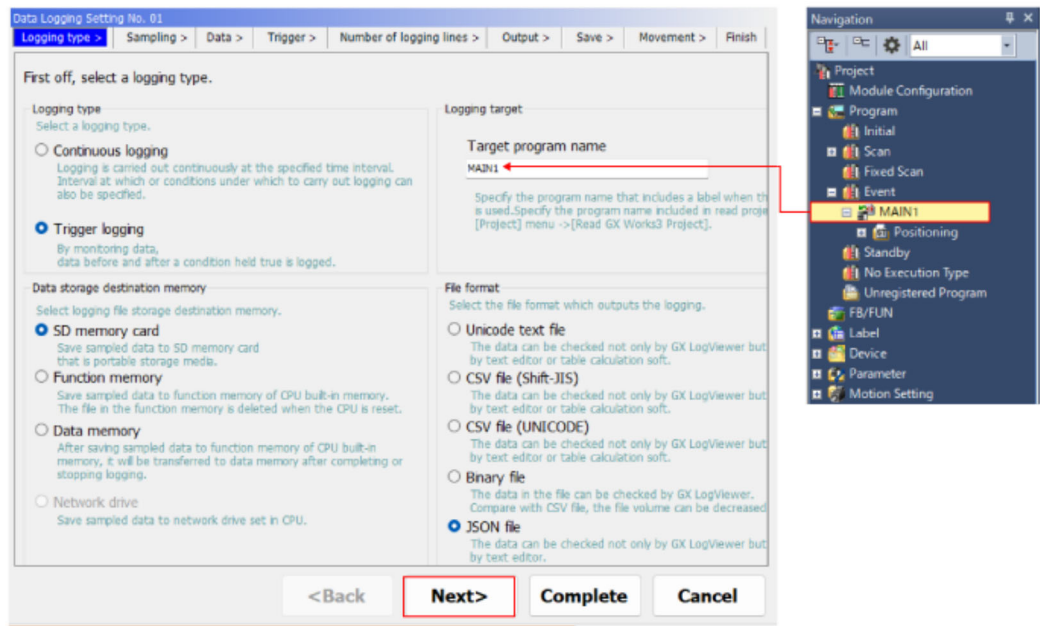
- (2) In CPU Module Logging Configuration Tool, select [Project] → [Read GX Works3 Project].
Specify the read source project path and project name, and then click the [OK] button.
Data loading starts.



(1) Click [Data Logging Setting] → [Edit].



(2) In [Logging type], set the logging type, data storage destination memory, file format, and target program name. After completing the setting, click [Next].



Set the following

Setting item	Description
Logging type	Trigger logging
Data storage destination memory	SD memory card
File format	JSON file
Target program name	MAIN1

[Point]

Data logging files can be saved in the following formats and can also be displayed in GX LogViewer.

File format	Application
Binary file format	Enables high-speed file access because the file size is smaller than that of the Unicode text file format.
Unicode text file format	A file format that can be opened with general-purpose applications
CSV file format	A file format that can be opened with general-purpose applications
JSON file format	A file format suitable for data processing on a host computer

- (3) In [Sampling], set the collection interval.
After completing the setting, click [Next].

Data Logging Setting No. 01

Logging type > **Sampling** > Data > Trigger > Number of logging lines > Output > Save > Movement > Finish

Specify the sampling interval and start conditions.

Sampling interval

☐ Each scanning cycle

Samples data at each sequence scanning cycle.

☐ Time specification

Samples data at the specified time interval.

☒ Sample data at the next END processing after the specified time has elapsed(W)

☐ Condition specification

Specifies data sampling timing by data conditions.

Device/Label

Conditional formula

Data/Constant

Data name/Constant

Data type(K)

☒ Network communication period synchronization

Synchronize with basic period

Samples data at the specified cycle interval.

☐ After executing specified program

Samples data after executing the specified program.

Program name

<Back

Next>

Complete

Cancel

Set the following.

Setting item	Description
Sampling interval	Network communication cycle synchronization: Synchronize with basic period

- (4) In [Data], set the logging target data.
After completing the setting, click [Next].

Data Logging Setting No. 01

Logging type > Sampling > **Data** > Trigger > Number of logging lines > Output > Save > Movement > Finish

Set the data for logging.

Set the data for logging.

A total of up to 1024 labels/devices can be set.

Motion Label Input Assistant... ← Target Program

MAIN1

No.	Label	Device	Data Type	Size [Byte]	Data Point	Output Format
		Head	Last			
001	G_bPosReq		Bit		1 ON:1, OFF:0	..
002	Positioning/bMoveA		Bit		1 ON:1, OFF:0	..
003	Positioning/bDwell1		Bit		1 ON:1, OFF:0	..
004	Positioning/bMoveA		Bit		1 ON:1, OFF:0	..
005	G_bPosDone		Bit		1 ON:1, OFF:0	..
006	Axis0001.Md.SetPos		Float[double preci	4		..
007	Axis0001.Md.SetVel		Float[double preci	4		..
008						..
009						..
010						..
011						..

Motion Label Input Assistant

Set the sampling data in logging.

Type: Real Drive Axis

Axis Name: Axis0001

Structured Data Type: Monitor Data

Data List

Name	Data Name (Label)
Md Control Cycle	* Md.OperationCycle
Md Operation Setting at Overrun	* Md.OverrunOperation
Md Current Position Restoration Status	* Md.PosRestoration_Status
Md Execution Profile ID No.	* Md.ProfileID
Md Set Acceleration	* Md.SetAcceleration
Md Set Position	* Md.SetPosition
Md Set Velocity	* Md.SetVelocity
Md Axis Emulating	* Md.SlaveEmulate_Enable
Md Start Permission at Homing Uncompleted	* Md.StartableAtUnhomed
Md Deceleration at Stop	* Md.StopMode_Deceleration
Md Stop Signal.Signal Detection Method	* Md.StopSignal_Detection
Md Stop Signal.Compensation Time	* Md.StopSignal.Compensation...
Md Stop Signal.Filter Time	* Md.StopSignal.FilterTime

Register

Close

Next>

Complete

Cancel

- 1) Global label
Enter the global label name in the "Label" column.
- 2) Local label
Enter the local label name in the "Label" column using the following format:
Program name/Local label name.
- 3) Structure
Click the [Motion Label Input Assistant] button to select a structure member from the list.
Select a member from the list and click the [Register] button.

Set the following.

Label name
G_bPosReq
Positioning/bMoveAbs1Done
Positioning/bDwell1_out
Positioning/bMoveAbs2Done
G_bPosDone
Axis0001.Md.SetPosition
Axis0001.Md.SetVelocity

(5) In [Trigger], click [Edit] and set the trigger condition.
After completing the setting, click [OK] and [Next].

Data Logging Setting No. 01

Logging type > | Sampling > | Data > | **Trigger >** | Number of logging lines > | Output > | Save > | Movement > | Finish

Make trigger setting.

Trigger Condition

Joint Condition(M) OR Joint Trigger occurs by meeting any of conditions which was set.

Condition List

Monitoring Data Points: 0 /32 Points Edit... Delete Delete All

No.	Type	Content
001		
002		
003		
004		
005		
006		
007		
008		
...		

Trigger Count

☒ Count Specification 1 (1 - 32767)
Repeat the specified count, trigger logging.
The operation will be "Overwrite" when it exceeds the upper bound of saved file count.

☐ Saved File Count Specification
Repeat trigger logging according to the upper bound setting of saved file count.

<Back

Next>

Complete

Cancel

[Point]

This setting is performed only when trigger logging is selected for the logging type.

OR Joint(No.1)

☒ Condition specification
Trigger occurs when the condition has been met after comparing the data and constant or the data and data.

Label name

G_bPosReq

Condition type

PLS

Data type

Bit

☐ Fixed scan [ms] (1 - 86400000)
Trigger occurs in each specified cycle.

☐ Event occurrence
Trigger occurs when the event or error of the corresponding event history occurs.
- Specify the event code or error code of trigger target.
- The multiple event codes or error codes can be specified using "" (hyphen) and "" (comma).

OK

Cancel

Set the following.

Setting item	Description
Trigger Condition	OR Joint
Condition specification	Label name: G_bPosReq
Condition type	PLS (rising edge)

Data Logging Setting No. 01

Logging type > | Sampling > | Data > | **Trigger >** | Number of logging lines > | Output > | Save > | Movement > | Finish

Make trigger setting.

Trigger Condition

Joint Condition(M) OR Joint Trigger occurs by meeting any of conditions which was set.

Condition List

Monitoring Data Points: 1 /32 Points Edit... Delete Delete All

No.	Type	Content
001	Condition Specification (Rising)	G_bPosReq
002		
003		
004		
005		
006		
007		
008		
...		

Trigger Count

☒ Count Specification 1 (1 - 32767)
Repeat the specified count, trigger logging.
The operation will be "Overwrite" when it exceeds the upper bound of saved file count.

☐ Saved File Count Specification
Repeat trigger logging according to the upper bound setting of saved file count.

<Back

Next>

Complete

Cancel

- (6) In [Number of logging lines], set values for "No. of records (before trigger)/No. of records (after trigger)".
After completing the setting, click [Next].

Data Logging Setting No. 01

Logging type > Sampling > Data > Trigger > Number of logging lines > Output > Save > Movement > Finish

Data before and after trigger condition rises will be logged.
Specify the numbers of records before and after trigger.

No. of records (before trigger)

1000

[Record] (0~999999)

No. of records (after trigger)

10000

[Record] (1~1000000)

Total No. of records

11000

[Record] (1~1000000)

<Back

Next>

Complete

Cancel

Set the following.

Setting item	Description
No. of records (before trigger)	1000
No. of records (after trigger)	10000

[Point]
This setting is performed only when trigger logging is selected for the logging type.

- (7) In [Output], set the items to be output into the file, and click [Next].

Data Logging Setting No. 01

Logging type > Sampling > Data > Trigger > Number of logging lines > Output > Save > Movement > Finish

Setting items to be output to a file.

Date

Carry out the logging with a time stamp attached to data.

☒ Output date

Set Date Line Format...

Trigger information

Data on which a trigger occurred is logged after attached with a mark.

☒ Output trigger information

Set String for Trigger...

Index

Output index numbers for checking the continuity of logging.

☒ Output index

Data sampling interval

Output data sampling interval time.

☒ Output data sampling interval(J)

Execution program name or execution program No.

Output execution program name and execution program No. which the data has been sampled.

☒ Output execution program name or program No.

Device comment output

In addition to device of data, output device comment.

☐ Output device comment(R)

Target memory

Target comment file name

Target comment number (1-16) It can be checked by "View-> Multiple Comment Display Setting" of GX Works3.

☐ Output each program device comment(Q)

Output each program device comment when it is a local device.

Target memory(W)

Comment

Outputs comments on the settings. The specified comment will be output at the top of the logging file.

☒ Output comments

Comment(K)

LOG01

<Back

Next>

Complete

Cancel

Set the following.

Setting item	Description
Date	Select the checkbox for "Output date".
Trigger information	Select the checkbox for "Output trigger information".
Index	Select the checkbox for "Output index".
Data sampling interval	Select the checkbox for "Output data sampling interval".
Execution program name or execution program No.	Select the checkbox for "Output execution program name or program No.".
Comment	Select the checkbox for "Output comments". Enter LOG01 for "Comment".

(8) In [Save], set the logging file save destination and file switching, and click [Next].

Data Logging Setting No. 01

Logging type > Sampling > Data > Trigger > Number of logging lines > Output > Save > Movement > Finish

Specify the save destination and switching timing of data logging files.

Logging file save setting

File save destination

Specify the logging file save directory (folder name). Data will be stored in the following folder:

/LOGGING/ LOG01

Folder to store file to be saved

Select the additional information to the folder name stored the file to be saved.

☐ Date(J) ☐ Time(K) Example 00000001

File name

☒ Simple setting(A) Simply set additional information...

The information which is added from folder name, date, time to file name can be set.

☐ Optional setting Optionally set additional information...

The additional information to the file name can be set freely.

Example 00000001.JSON

File switching setting

Number of files to be saved

Specify the maximum number of files to be saved.

☐ Do not specify maximum value

Create the file until the capacity of save destination is full.

Number of files to be saved 1 (1-65535)

Operation when exceeds the number of files:

☒ Overwrite

Files with lower numbers are deleted and logging continues.

☐ Stop

Logging stops.

File switching timing

Specify the file switching timing.

☐ Number of records [Record] (1-100000)

A logging file is switched to the new one when the number of records exceeds the specified value.

☐ File size [KB] (10-16384)

A logging file is switched to the new one when the file size exceeds the specified value.

☐ Condition specification Condition setting...

A logging file is switched to the new one when the device value satisfies the condition.

<Back

Next>

Complete

Cancel

Set the following

Setting item	Description
File save destination	LOG01
File name	Simple setting
Number of files to be saved	1

(9) In [Movement], set the logging start condition and the logging stop condition, and click [Next].

Data Logging Setting No. 01

Logging type > Sampling > Data > Trigger > Number of logging lines > Output > Save > Movement > Finish

Specify logging operation.

Logging start condition
Specify logging operation at data logging settings registered status when the CPU module power is ON
->RUN, reset->RUN, or the CPU module operation status is STOP->RUN.

☐ Auto start at initial completion

☐ Auto start at transition to RUN

☒ Start by user operation

Logging stop condition
Set the conditions for stopping data logging settings.

☒ Stop when transitioning to a state other than RUN

☐ Stop by user operation

<Back

Next>

Complete

Cancel

Set the following.

Setting item	Description
Logging start condition	Start by user operation
Logging stop condition	Stop when transitioning to a state other than RUN

(10) In [Finish], check the set details, set any name, and click [Complete].

Data Logging Setting No. 01

Logging type > Sampling > Data > Trigger > Number of logging lines > Output > Save > Movement > Finish

All data required for data logging have been collected.
Press the "Complete" button to complete setting.

To reflect the settings to the PLC, select [Online]->[Write Logging Setting].

Name the data logging.
Data Logging Name LOG01

Free space volume below will be necessary to execute logging of the set content.
Larger volume might be necessary depending on status of data storage destination memory.
Total Size of Output Logging Files 3 [MB]

To execute logging of the settings, the following internal buffer capacity is required.
Please set internal buffer capacity as needed.
Required Internal Buffer Capacity in Logging 49 [KB]

The internal buffer capacity can be set in
'Parameter->Control CPU->CPU Parameter->Memory/Device Setting->Internal Buffer Volume Setting' of GX Works3.
Default value: 1024[KB]

<Back

Next>

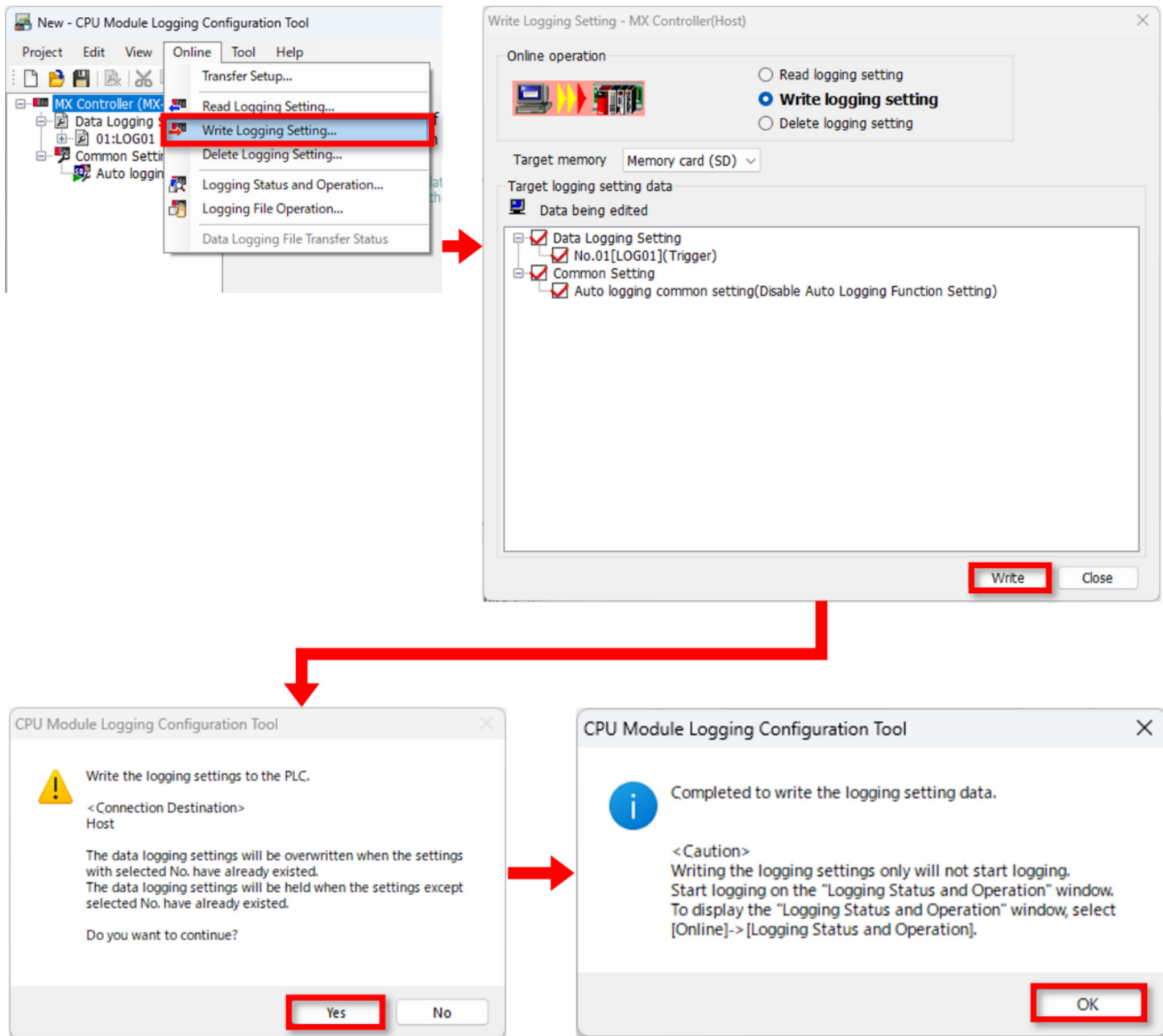
Complete

Cancel

Set the following.

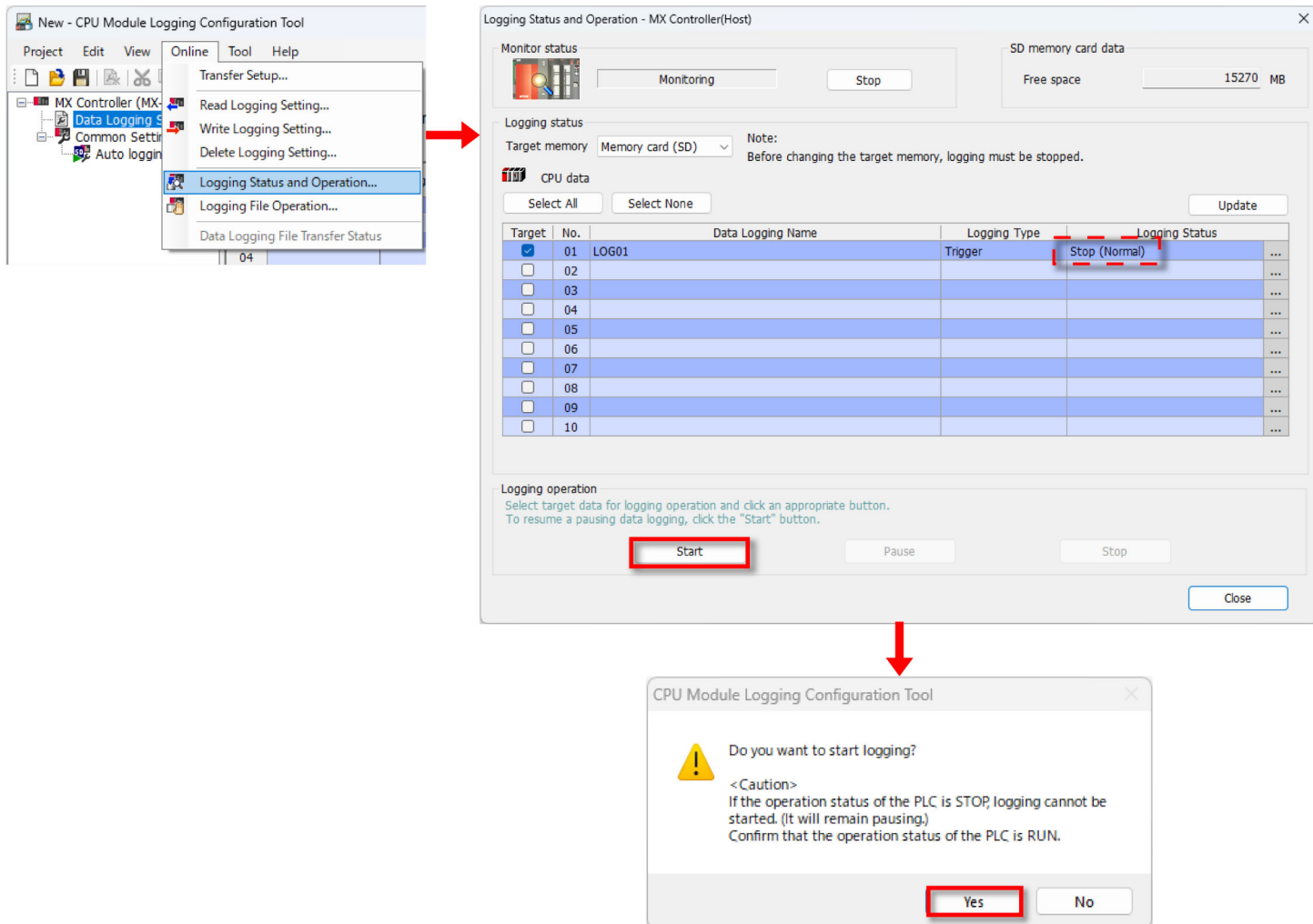
Setting item	Description
Data Logging Name	Enter "LOG01" (default value).

- (1) Insert an SD memory card into the MELSEC MX Controller, and turn on the power.
- (2) Write the data logging settings to the data memory*1 or SD memory card.
Click [Online] → [Write Logging Setting].
- (3) Click [Write].



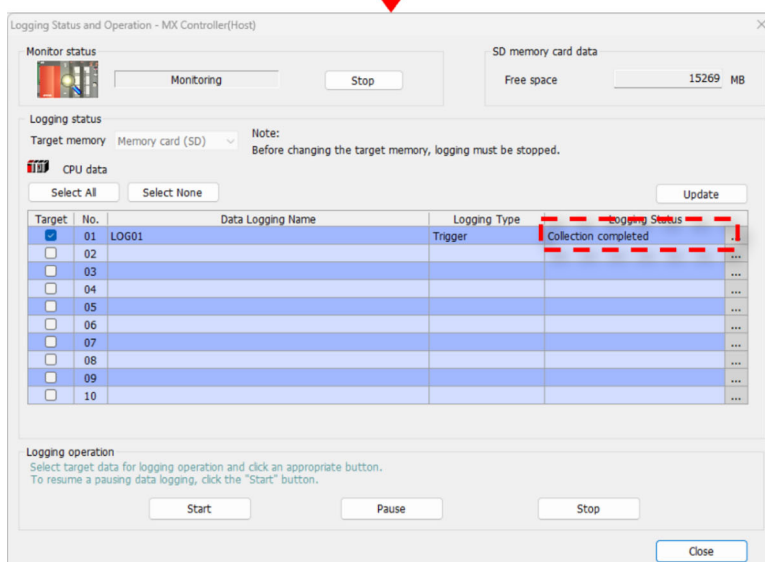
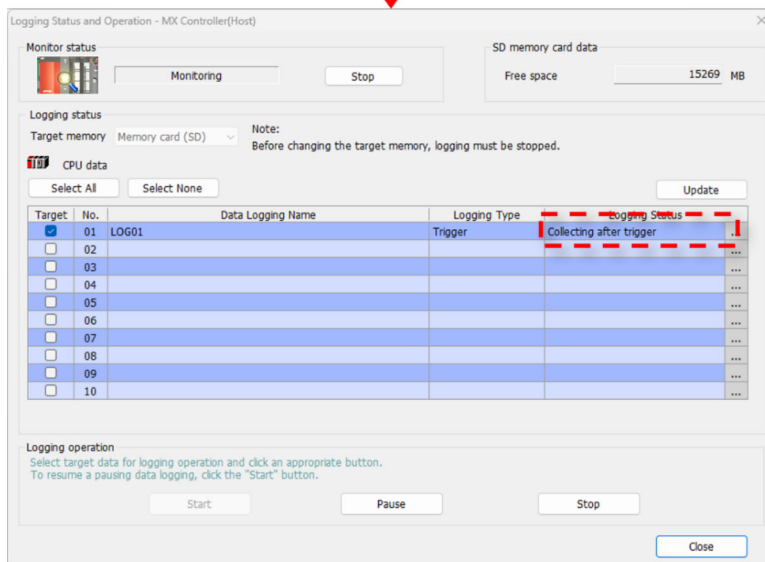
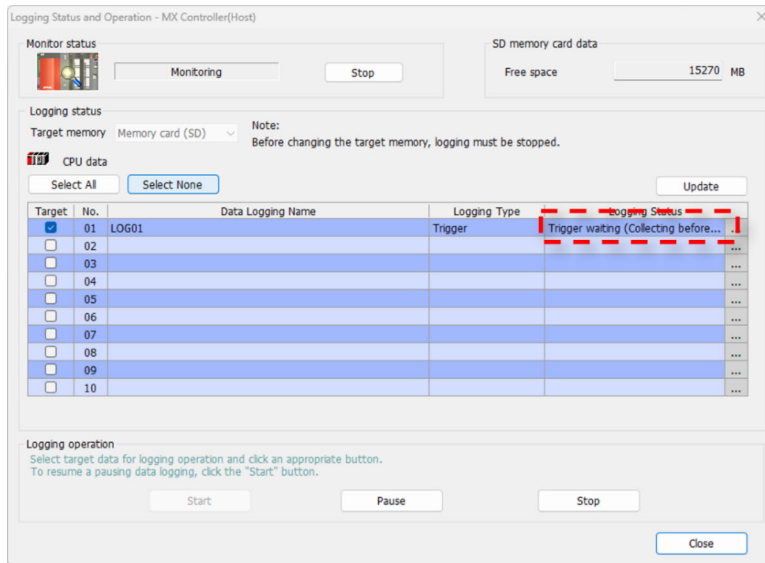
*1: When writing to the data memory, the settings can be written to the built-in memory even if an SD memory card is not inserted.

- (1) Set the MELSEC MX Controller to the RUN state.
- (2) Open the "Logging Status and Operation" window.
Click [Online] → [Logging Status and Operation].
- (3) Select the target of data logging execution (multiple targets can be selected). In this example, select "No.01".
- (4) Click [Start]. The logging status changes to "Trigger waiting".

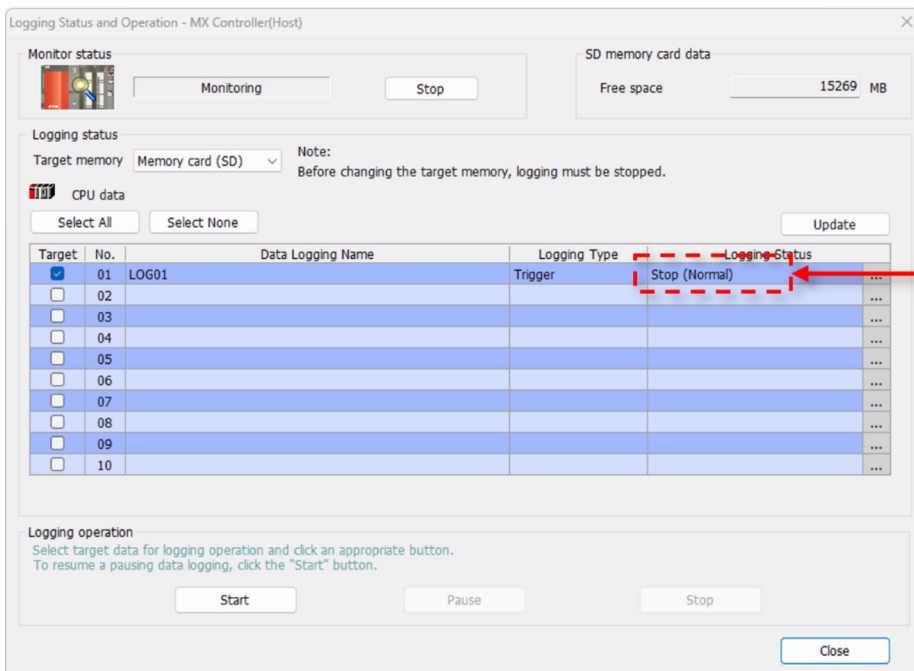
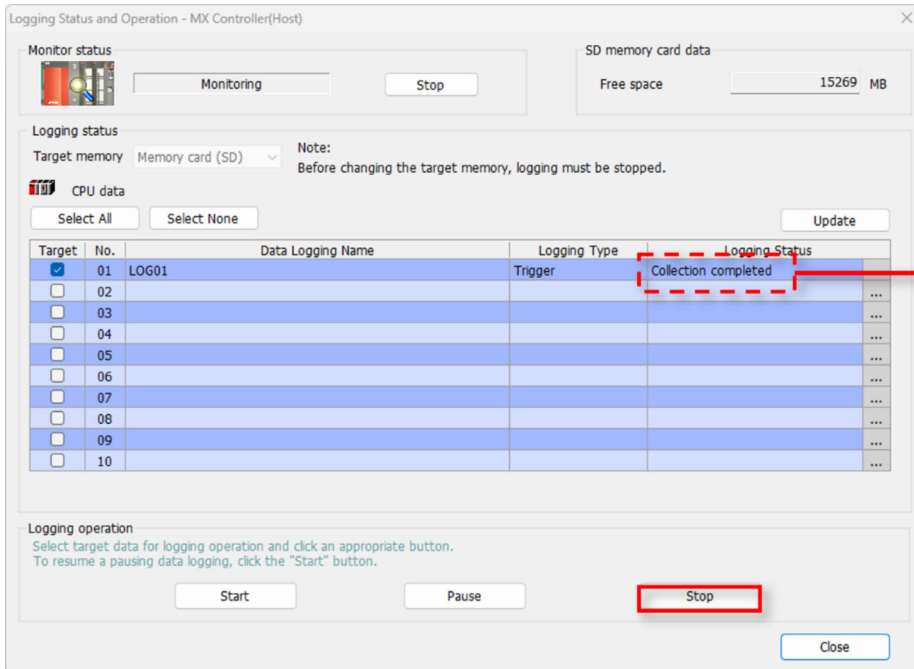


[Point]
Multiple data logging targets can be selected and executed simultaneously.

- (5) Execute the program in this status. When the trigger conditions are satisfied (when the servo ON command (X0) and the positioning control command (X4) are turned on in this course), the trigger is activated.
- (6) The logging status changes from "Stop (Normal)" to "Collecting before trigger", and then to "Collecting after trigger".
- (7) When data collection is completed, the logging status changes to "Collection completed".



- (1) Open the "Logging Status and Operation" window.
Click [Online] → [Logging Status and Operation].
- (2) Select the target of data logging stop (multiple targets can be selected).
- (3) Click [Stop] to stop the data logging.
- (4) The logging status changes to "Stop (Normal)".

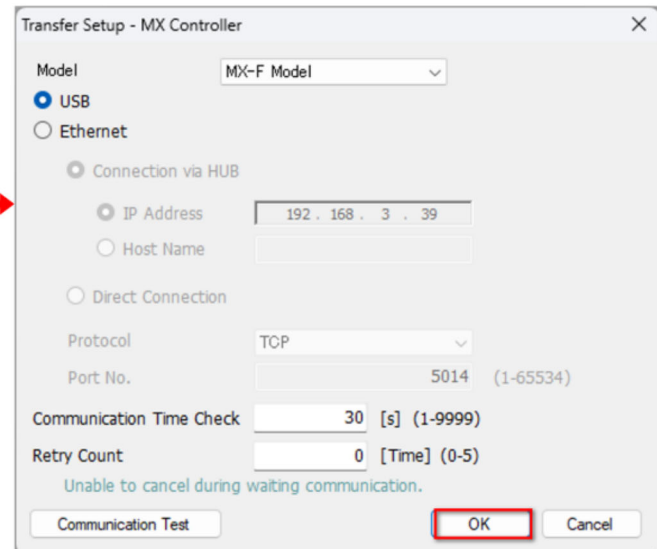
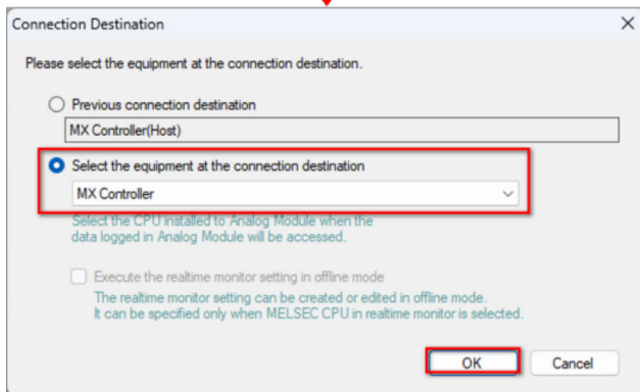
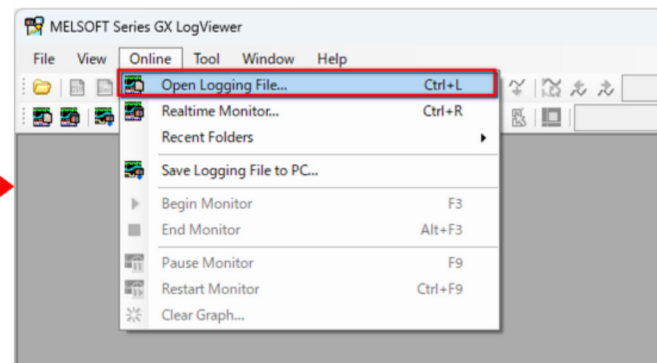
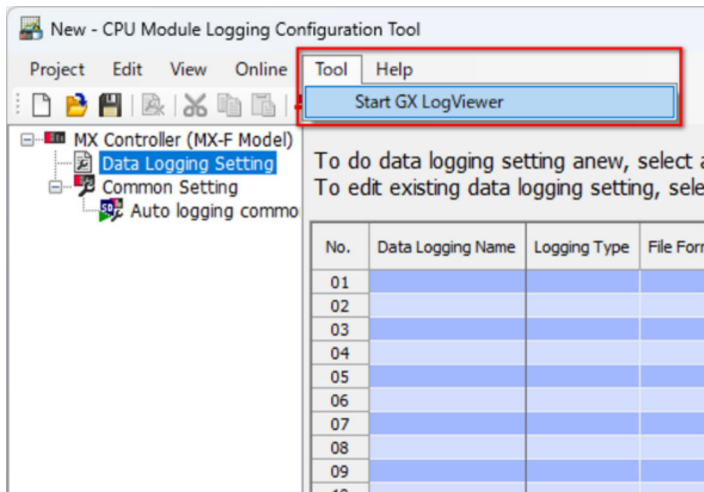


Use GX LogViewer to read logging data.

Select [Tool] → [Start GX LogViewer] in the "CPU Module Logging Configuration Tool" window.

After GX LogViewer starts, select [Online] → [Open Logging File].

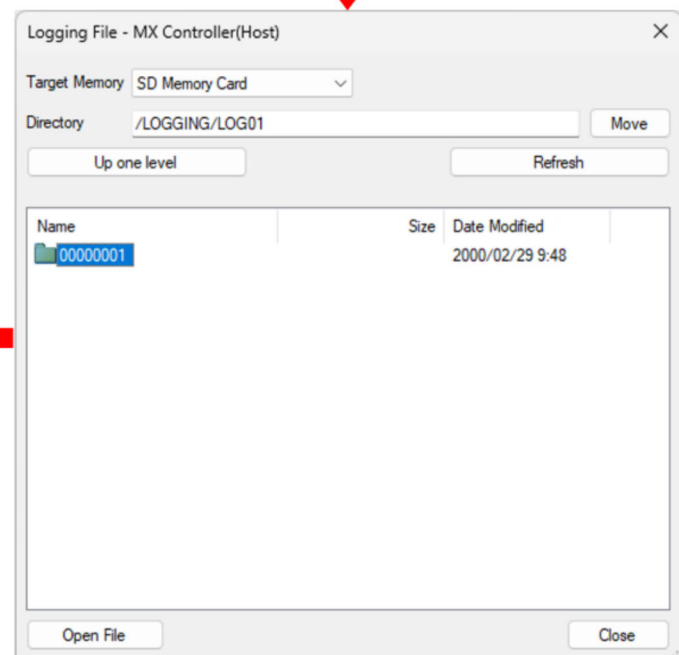
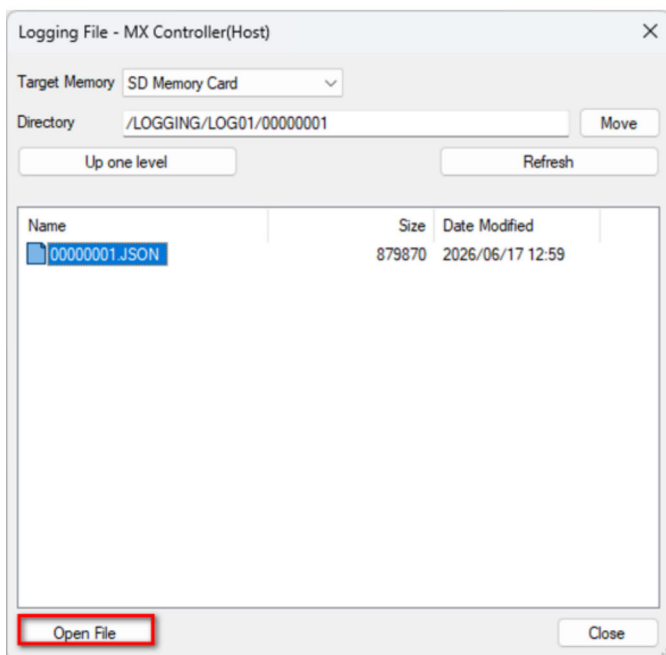
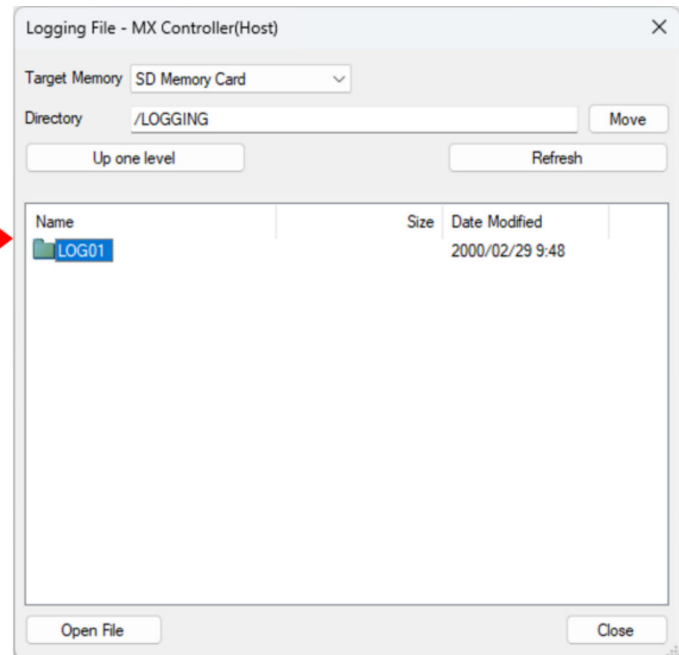
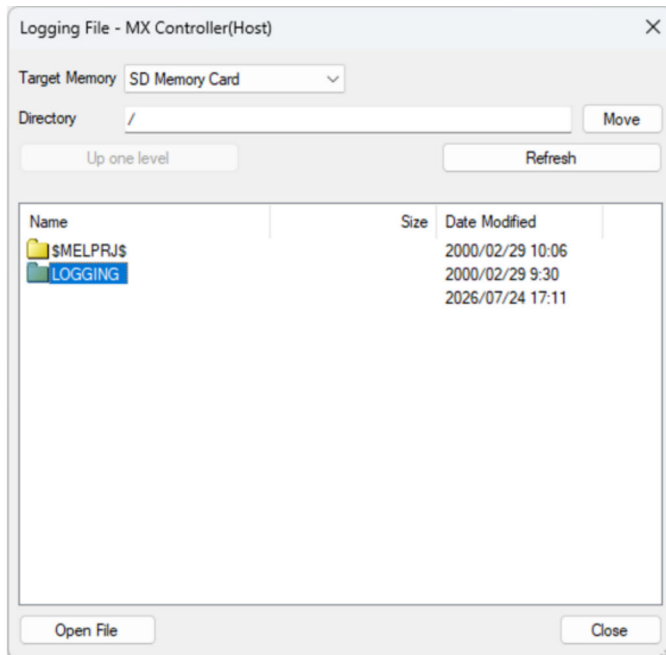
In the "Connection Destination" window, select [MX Controller], and then click [OK].



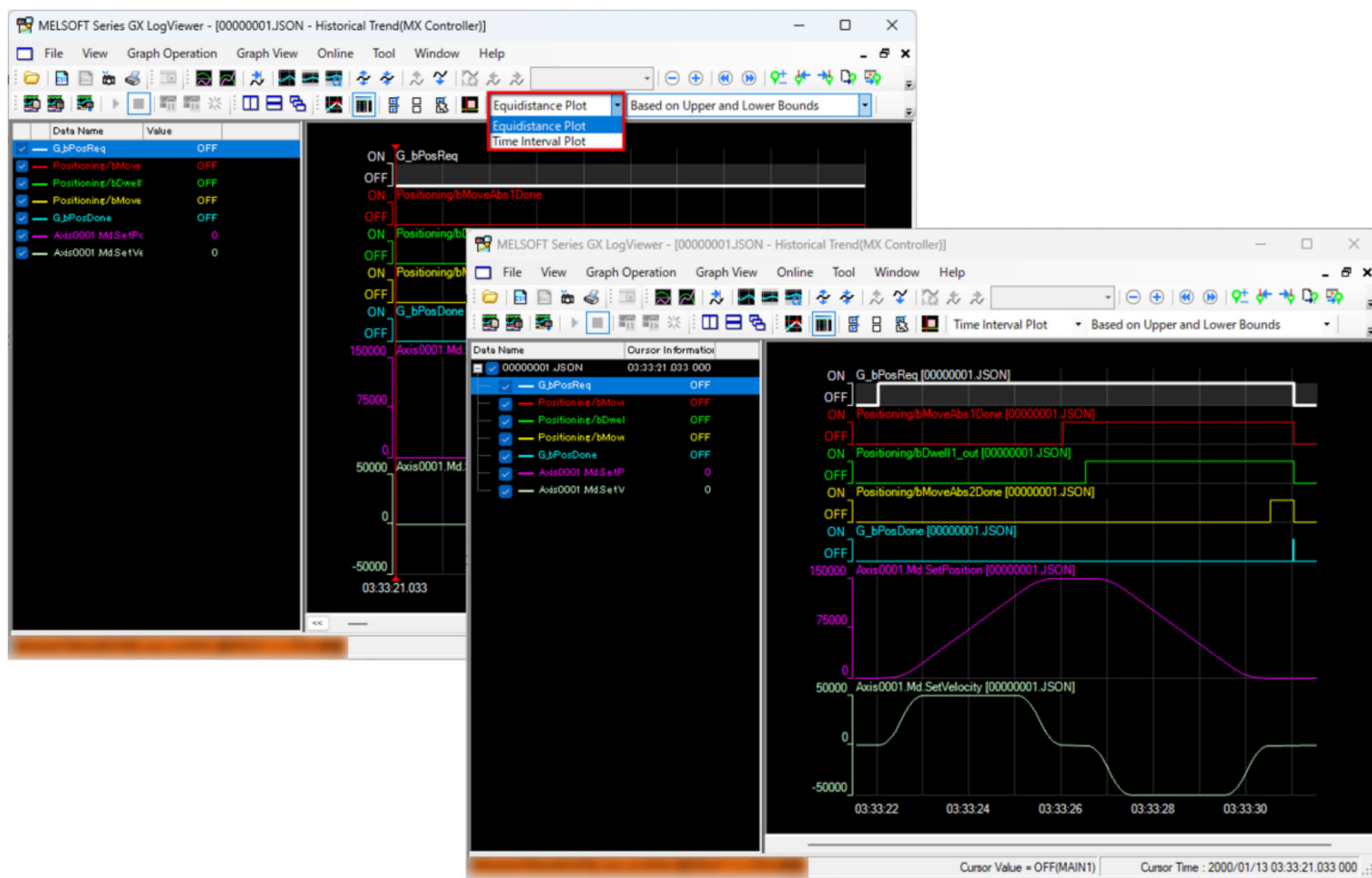
Select the logging file to be read.

In this example, the file is located in the following folder on the SD memory card: "LOGGING" → "LOG01" → "File name (serial number).JSON"

Select the file name and click the [Open File] button.

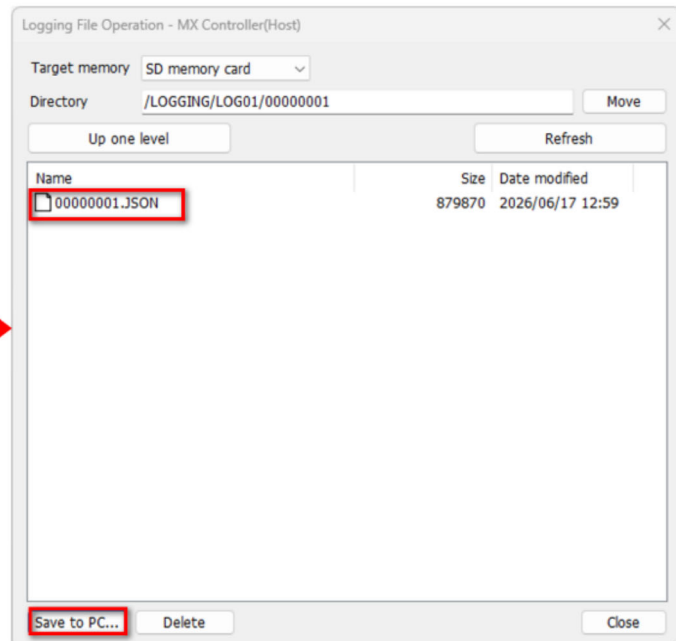
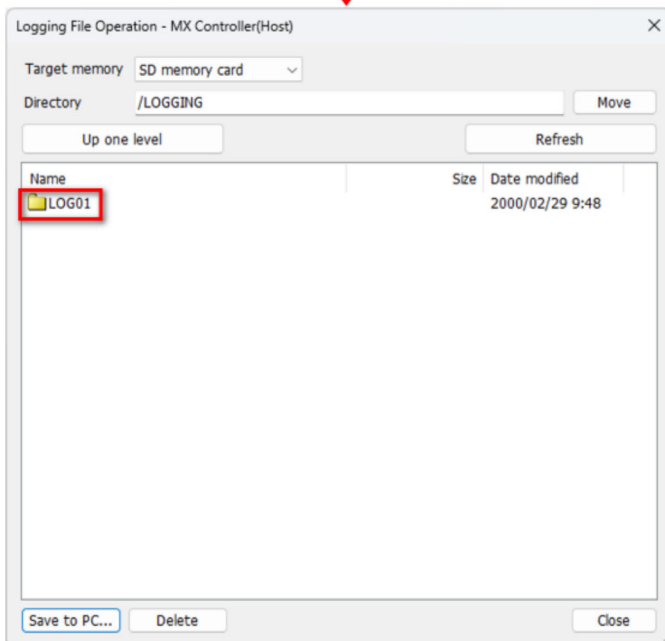
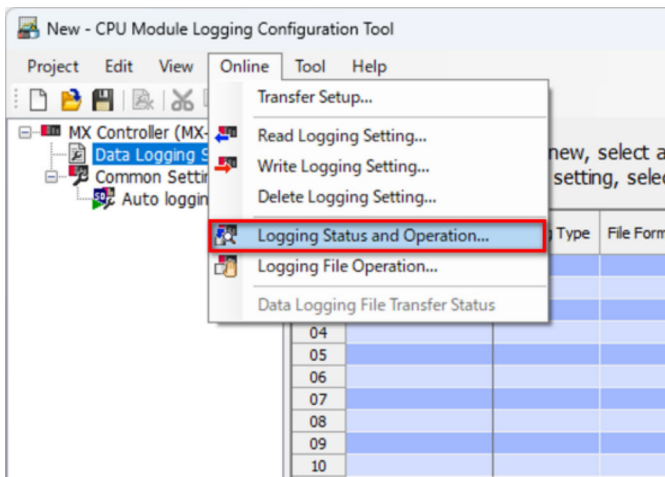


The waveform data logged in GX LogViewer is displayed. When the plot format is changed from [Equidistance Plot] to [Time Interval Plot], the entire logged waveform can be displayed.



Data logging files can be saved in the following formats: CSV file, Unicode text file, binary file, and JSON file. The files are saved to the SD memory card in the specified file format. They can be opened on a personal computer and used for such purposes as creating reports and analyzing data.

- (1) Open the "Logging File Operation" window.
Click [Online] → [Logging File Operation].
- (2) Specify the directory and select the targeted file.
- (3) Click the [Save to PC] button.
- (4) Specify the save destination and click the [Save] button.
- (5) The data logging file is saved into the specified location.



In this chapter, you have learned:

- Launching CPU Module Logging Configuration Tool
- Data Logging Settings
- Writing Data Logging Settings
- Starting Data Logging
- Reading Logging Data
- Saving Logging Data

Important points

Launching CPU Module Logging Configuration Tool	• Start CPU Module Logging Configuration Tool from GX Works3.
Data Logging Settings	• Follow the procedure displayed in CPU Module Logging Configuration Tool to configure the settings such as logging data and trigger conditions.
Writing Data Logging Settings	• Before performing logging, write the logging setting data to the MELSEC MX Controller.
Starting Data Logging	• If the logging start condition is set to "Start by user operation", click the [Start button] in the "Logging Status and Operation" window.
Reading Logging Data	• Use GX LogViewer to read logging data.
Saving Logging Data	• Logged waveform data can be saved in the following formats: CSV file, Unicode text file, binary file, and JSON file.

Now that you have completed all of the lessons of the **Programmable Automation Controller MELSEC MX Controller Basics(Positioning)** 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 4 questions (12 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.

		1	2	3	4	5	6	7	8	9	10	
Retry	Final Test 1	✓	✓	✓	✗							Total questions: 28
	Final Test 2	✓	✓	✓	✓							Correct answers: 23
	Final Test 3	✓										Percentage: 82 %
	Final Test 4	✓	✓									
	Final Test 5	✓	✓									
Retry	Final Test 6	✓	✗	✗	✗							
	Final Test 7	✓	✓	✓	✓							
	Final Test 8	✓	✓	✓	✓	✓						
	Final Test 9	✓										
Retry	Final Test 10	✗										

To pass the test, 60% of correct answers is required.

Select ○ (True) or × (False) for each statement about the characteristics of buffer mode.

Q1: The next operation can be reserved before the current operation is completed.

Q2: A stop time always occurs between operations.

Q3: There is a limit to the number of buffer registrations.

Q1

○



Q2

×



Q3

○



Q4

×

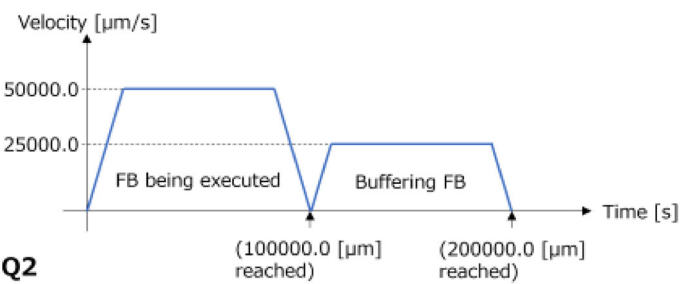


The FB to be executed first is referred to as FB1, and the FB to be executed next is referred to as FB2.

When the target positions and target velocities of FB1 and FB2 are as listed in the table below, select the buffer mode that produces the following operation pattern.

	Target position	Target velocity
FB1	100000.0 [μm]	50000.0 [μm/s]
FB2	200000.0 [μm]	25000.0 [μm/s]

Q1

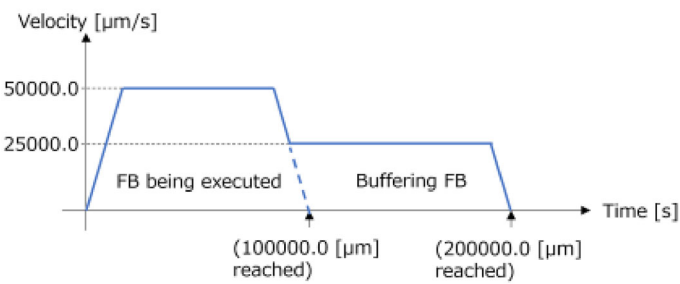


Q1

mcBuffered

▼

Q2



Q2

mcBlendingNext and mcBlendingLow

▼

Select ○ (True) or × (False) for each statement about creating a program that uses motion control FBs with the MELSEC MX Controller.

Q1: The FB library must be registered to GX Works3 to use motion control FBs.

Q2: Place motion control FBs in the program editor from the project tree of GX Works3.

Q1

×



Q2

×



Q3

○



Complete the following sentences about logging.

- Start **(Q1)** to set the data to be logged.
- Write the logging data to **(Q2)** to perform logging.
- **(Q3)** is used to read the logging data and check the waveform.

Q1

CPU Module Logging Configuration Tool

**Q2**

MELSEC MX Controller

**Q3**

GX LogViewer



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	✓	✓	✓							

Total questions: 12
Correct answers: 12
Percentage: 100 %

Clear

**You have completed the "Programmable Automation Controller
MELSEC MX Controller Basics (Positioning)" 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