OMRON

Sysmac Library

User's Manual for AI Predictive Maintenance Library

SYSMAC-ZPA□□□000W

SYSMAC always in control

W610-E1-01

NOTE

(1) All rights reserved. No part of this publication may be reproduced, stored in a retrieval system, or transmitted, in any form, or by any means, mechanical, electronic, photocopying, recording, or otherwise, without the prior written permission of OMRON.

(2) No patent liability is assumed with respect to the use of the information contained herein. Moreover, because OMRON is constantly striving to improve its high-quality products, the information contained in this manual is subject to change without notice.

(3) Every precaution has been taken in the preparation of this manual. Nevertheless, OMRON assumes no responsibility for errors or omissions.
Neither is any liability assumed for damages resulting from the use of the information contained in this publication.

Trademarks

- Sysmac and SYSMAC are trademarks or registered trademarks of OMRON Corporation in Japan and other countries for OMRON factory automation products.
- Microsoft, Windows, Windows Vista, Excel, and Visual Basic are either registered trademarks or trademarks of Microsoft Corporation in the United States and other countries.
- EtherCAT® is a patented technology and registered trademark, licensed by Beckhoff Automation GmbH, Germany.
- ODVA, CIP, CompoNet, DeviceNet, and EtherNet/IP are trademarks of ODVA.
- The SD and SDHC logos are trademarks of SD-3C, LLC.

Other company names and product names in this document are the trademarks or registered trademarks of their respective companies.

Copyrights

- Microsoft product screen shots reprinted with permission from Microsoft Corporation.

Introduction

Thank you for purchasing the NX/NY-series Artificial Intelligence Machine Automation Controller.
This manual contains information that is necessary to use the function block (hereinafter may be abbreviated as FB) of the AI Predictive Maintenance Library. Please read this manual and make sure you understand the functionality and performance of the product before you attempt to use it in a control system.
This manual provides FB (function block) specifications. It does not describe usage restrictions or combination restrictions for Controllers, each Unit, or other components.
Make sure to read the user's manual for each product before use.
Keep this manual in a safe place where it will be available for reference during operation.

Features of the Library

The AI Predictive Maintenance Library is one of the Sysmac Library software function components for using the AI functions of the NX/NY-series Artificial Intelligence Machine Automation Controller (hereinafter may be abbreviated as AI Controller).
The AI Predictive Maintenance Library contains libraries that include FB for each mechanism (devices and components such as the Cylinder, Ball Screw, and Belt Pulley) to be used.
By passing the control and status data of an operating mechanism to the FB, you can use the Feature Value/Machine Learning Function of the AI Controller to detect abnormal states of the mechanism.
With the AI Predictive Maintenance Library, you can easily achieve predictive maintenance using AI functions.
Refer to *NX/NY-series Artificial Intelligence Machine Automation Controller User's Manual (Cat. No. W594)* for details on AI functions.

Intended Audience

This manual is intended for the following personnel,
who must also have knowledge of electrical systems (an electrical engineer or the equivalent).

- Personnel in charge of introducing FA systems.
- Personnel in charge of designing FA systems.
- Personnel in charge of installing and maintaining FA systems.
- Personnel in charge of managing FA systems and facilities.

For programming, this manual is intended for personnel who understand the programming language specifications in international standard IEC 61131-3 or Japanese standard JIS B 3503.

Applicable Products

This manual covers the following products.

Item	Product name	Model	Version
Sysmac Library AI Predictive Maintenance Library	Cylinder	SYSMAC-ZPA001000W	Version 1.00 or higher
	Ball Screw	SYSMAC-ZPA002000W	Version 1.00 or higher
	Belt Pulley	SYSMAC-ZPA003000W	Version 1.00 or higher
Automation Software	Sysmac Studio	SYSMAC-SE2□□□	Version 1.25 or higher
AI Controller	NX-series CPU Unit	NX701-Z□00	Unit version 1.18 or later
	NY-series Industrial PC	NY5□2-Z□00	Unit version 1.18 or later
AI Controller Standard Software*1	AI Operator	SYSMAC-AICSTE□□□	Version 1.00 or higher
	AI Viewer		Version 1.00 or higher

*1. Before you use the AI Controller Standard Software, check the version of the Sysmac Library AI Predictive Maintenance Library that can be used.

- License

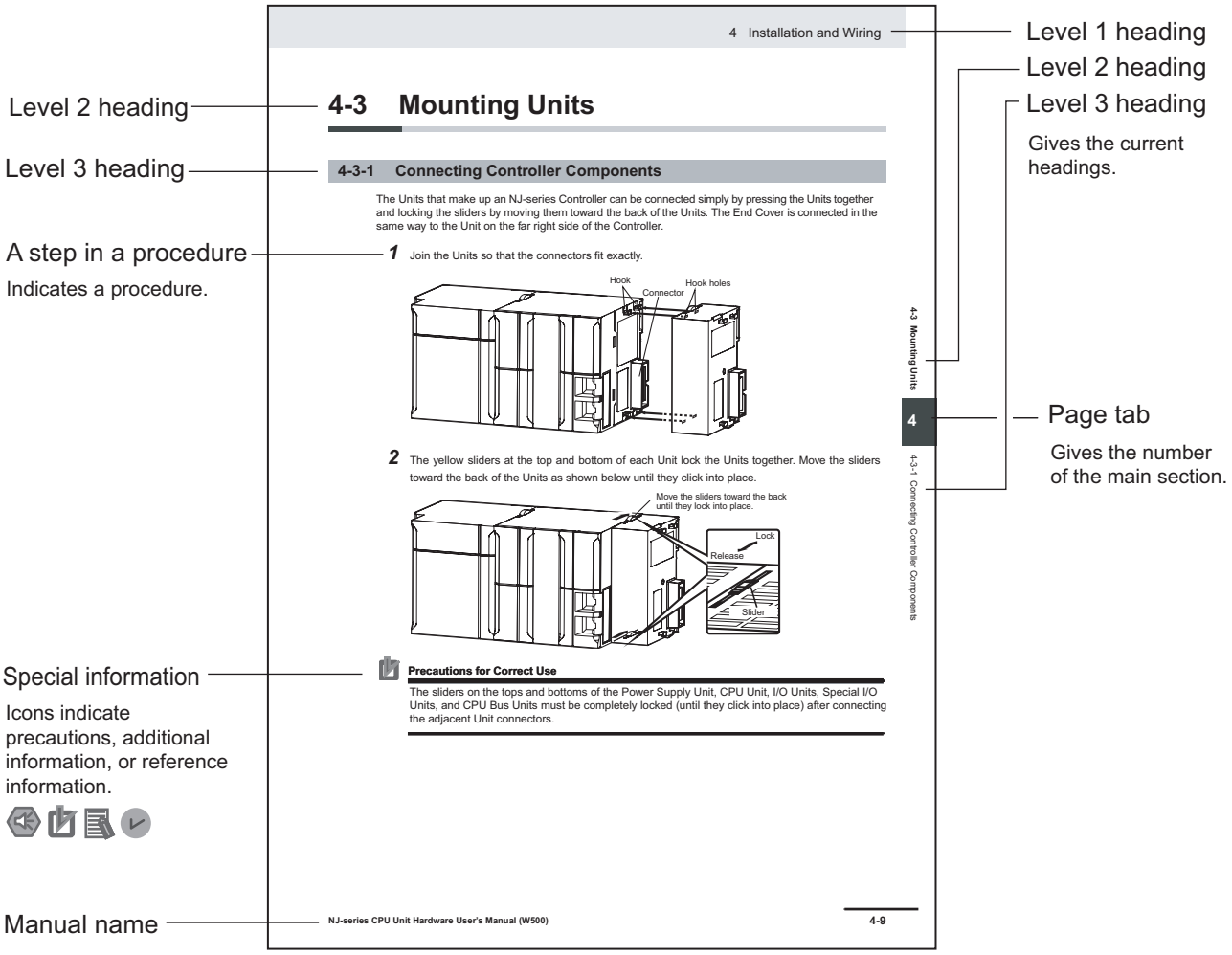
Product name	License	Model
AI Predictive Maintenance Library	5 licenses	SYSMAC-ZPA□□□005L*1
	10 licenses	SYSMAC-ZPA□□□010L*1
	50 licenses	SYSMAC-ZPA□□□050L*1

*1. □□□ is the numbers below.
001: Cylinder, 002: Ball Screw, 003: Belt Pulley

Manual Structure

Page Structure

The following page structure is used in this manual.



Note This illustration is provided only as a sample. It may not literally appear in this manual.

Special Information

Special information in this manual is classified as follows:

Precautions for Safe Use

Precautions on what to do and what not to do to ensure safe usage of the product.

Precautions for Correct Use

Precautions on what to do and what not to do to ensure proper operation and performance.

Additional Information

Additional information to read as required.
This information is provided to increase understanding and make operation easier.

Version Information

Information on differences in specifications and functionality for CPU Units with different unit versions and for different versions of the industrial-use PC, Sysmac Studio are given.

Sections in this Manual

CONTENTS

Section 7 Sample Programming

Appendix

Index

Terms and Conditions Agreement

Warranty, Limitations of Liability

Warranties

● Exclusive Warranty

Omron's exclusive warranty is that the Products will be free from defects in materials and workmanship for a period of twelve months from the date of sale by Omron (or such other period expressed in writing by Omron). Omron disclaims all other warranties, express or implied.

● Limitations

OMRON MAKES NO WARRANTY OR REPRESENTATION, EXPRESS OR IMPLIED, ABOUT NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE OF THE PRODUCTS. BUYER ACKNOWLEDGES THAT IT ALONE HAS DETERMINED THAT THE PRODUCTS WILL SUITABLY MEET THE REQUIREMENTS OF THEIR INTENDED USE.

Omron further disclaims all warranties and responsibility of any type for claims or expenses based on infringement by the Products or otherwise of any intellectual property right.

● Buyer Remedy

Omron's sole obligation hereunder shall be, at Omron's election, to (i) replace (in the form originally shipped with Buyer responsible for labor charges for removal or replacement thereof) the non-complying Product, (ii) repair the non-complying Product, or (iii) repay or credit Buyer an amount equal to the purchase price of the non-complying Product; provided that in no event shall Omron be responsible for warranty, repair, indemnity or any other claims or expenses regarding the Products unless Omron's analysis confirms that the Products were properly handled, stored, installed and maintained and not subject to contamination, abuse, misuse or inappropriate modification. Return of any Products by Buyer must be approved in writing by Omron before shipment. Omron Companies shall not be liable for the suitability or unsuitability or the results from the use of Products in combination with any electrical or electronic components, circuits, system assemblies or any other materials or substances or environments. Any advice, recommendations or information given orally or in writing, are not to be construed as an amendment or addition to the above warranty.

See http://www.omron.com/global/ or contact your Omron representative for published information.

Limitation on Liability; Etc

OMRON COMPANIES SHALL NOT BE LIABLE FOR SPECIAL, INDIRECT, INCIDENTAL, OR CONSEQUENTIAL DAMAGES, LOSS OF PROFITS OR PRODUCTION OR COMMERCIAL LOSS IN ANY

WAY CONNECTED WITH THE PRODUCTS, WHETHER SUCH CLAIM IS BASED IN CONTRACT, WARRANTY, NEGLIGENCE OR STRICT LIABILITY.

Further, in no event shall liability of Omron Companies exceed the individual price of the Product on which liability is asserted.

Application Considerations

Suitability of Use

Omron Companies shall not be responsible for conformity with any standards, codes or regulations which apply to the combination of the Product in the Buyer's application or use of the Product. At Buyer's request, Omron will provide applicable third party certification documents identifying ratings and limitations of use which apply to the Product. This information by itself is not sufficient for a complete determination of the suitability of the Product in combination with the end product, machine, system, or other application or use. Buyer shall be solely responsible for determining appropriateness of the particular Product with respect to Buyer's application, product or system. Buyer shall take application responsibility in all cases.

NEVER USE THE PRODUCT FOR AN APPLICATION INVOLVING SERIOUS RISK TO LIFE OR PROPERTY OR IN LARGE QUANTITIES WITHOUT ENSURING THAT THE SYSTEM AS A WHOLE HAS BEEN DESIGNED TO ADDRESS THE RISKS, AND THAT THE OMRON PRODUCT(S) IS PROPERLY RATED AND INSTALLED FOR THE INTENDED USE WITHIN THE OVERALL EQUIPMENT OR SYSTEM.

Programmable Products

Omron Companies shall not be responsible for the user's programming of a programmable Product, or any consequence thereof.

Disclaimers

Performance Data

Data presented in Omron Company websites, catalogs and other materials is provided as a guide for the user in determining suitability and does not constitute a warranty. It may represent the result of Omron's test conditions, and the user must correlate it to actual application requirements. Actual performance is subject to the Omron's Warranty and Limitations of Liability.

Change in Specifications

Product specifications and accessories may be changed at any time based on improvements and other reasons. It is our practice to change part numbers when published ratings or features are changed, or when significant construction changes are made. However, some specifications of the Product may

be changed without any notice. When in doubt, special part numbers may be assigned to fix or establish key specifications for your application. Please consult with your Omron's representative at any time to confirm actual specifications of purchased Product.

Errors and Omissions

Information presented by Omron Companies has been checked and is believed to be accurate; however, no responsibility is assumed for clerical, typographical or proofreading errors or omissions.

Safety Precautions

Definition of Precautionary Information

The following notation is used in this user's manual to provide precautions required to ensure safe usage of this library on the AI Controller.
The safety precautions that are provided are extremely important for safety. Always read and heed the information provided in all safety precautions.
The following notation is used.

⚠ WARNING	Indicates a potentially hazardous situation which, if not avoided, could result in death or serious injury. Additionally, there may be severe property damage.
⚠ Caution	Indicates a potentially hazardous situation which, if not avoided, may result in minor or moderate injury, or property damage.

Symbols

	The circle and slash symbol indicates operations that you must not do. The specific operation is shown in the circle and explained in text. This example indicates that disassembly is prohibited.
	The triangle symbol indicates precautions (including warnings). The specific operation is shown in the triangle and explained in text. This example indicates a precaution for electric shock.
	The triangle symbol indicates precautions (including warnings). The specific operation is shown in the triangle and explained in text. This example indicates a general precaution.
	The filled circle symbol indicates operations that you must do. The specific operation is shown in the circle and explained in text. This example shows a general precaution for something that you must do.

WARNING

⚠ WARNING

Emergency stop circuits, interlock circuits, limit circuits, and similar safety measures must be provided in external control circuits.	
Check the user program, data, and parameter settings for proper execution before you use them for actual operation.	

Caution

⚠ Caution

Read all related manuals carefully before you use this library.	
The Sysmac Library and manuals are assumed to be used by personnel that is given in Intended Audience in this manual. Otherwise, do not use them.	
Perform the test run by holding an emergency stop switch in hand or otherwise prepare for rapid motor operation in an application to control the motor. Also perform the test run by using parameters for which the motor does not rapidly accelerate or decelerate before you gradually adjust the parameters.	
In heating or cooling applications, perform the test run by using parameters for which rapid temperature changes will not occur before you gradually adjust the parameters.	
You must confirm that the user program and parameter values are appropriate to the specifications and operation methods of the devices.	
The sample programming shows only the portion of a program that uses the function or function block from the library.	
When you use actual devices, also use programs such as safety circuits, device interlocks, I/O with other devices, and other control procedures.	
Understand the contents of sample programming before you use the sample programming and create the user program.	
Create a user program that will produce the intended device operation.	

Precautions for Correct Use

Using the Library

- When you use the library, functions or function blocks that are not described in the library manual may be displayed on the Sysmac Studio. Do not use functions or function blocks that are not described in the manual.
- You cannot change the source code of the functions or function blocks that are provided in the Sysmac Library.
- You cannot perform the multi-execution (buffer mode) in the Sysmac Library.

Using Sample Programming

Check the user program for proper execution before you use it for actual operation.

Operation

- Specify the input parameter values within the valid range.
- In a function or function block with an Enabled output variable, if the value of Enabled is FALSE, do not use the processing result of the function or function block as a command value to the control target.
- In the function block with Execute, do not perform re-execution by the same instance. The output value of the function block will return to the default value.

Related Manuals

The followings are the manuals related to this manual. Use these manuals for reference.

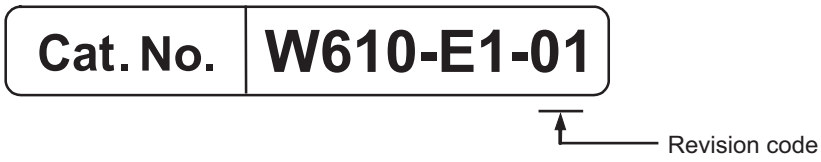
Manual name	Cat. No.	Model numbers	Application	Description
NX-series CPU Unit Hardware User's Manual	W535	NX701-□□□□	Learning the basic specifications of the NX701 CPU Units, including introductory information, designing, installation, and maintenance. Mainly hardware information is provided.	An introduction to the entire NX701 system is provided along with the following information on the CPU Unit. • Features and system configuration • Introduction • Part names and functions • General specifications • Installation and wiring • Maintenance and inspection
NJ/NX-series CPU Unit Software User's Manual	W501	NX701-□□□□ NX102-□□□□ NX1P2-□□□□ NJ501-□□□□ NJ301-□□□□ NJ101-□□□□	Learning how to program and set up an NJ/NX-series CPU Unit. Mainly software information is provided.	The following information is provided on a Controller built with an NJ/NX-series CPU Unit. • CPU Unit operation • CPU Unit features • Initial settings • Programming based on IEC 61131-3 language specifications
NJ/NX-series Instructions Reference Manual	W502	NX701-□□□□ NX102-□□□□ NX1P2-□□□□ NJ501-□□□□ NJ301-□□□□ NJ101-□□□□	Learning detailed specifications on the basic instructions of an NJ/NX-series CPU Unit.	The instructions in the instruction set (IEC 61131-3 specifications) are described.
NJ/NX-series CPU Unit Motion Control User's Manual	W507	NX701-□□□□ NX102-□□□□ NX1P2-□□□□ NJ501-□□□□ NJ301-□□□□ NJ101-□□□□	Learning about motion control settings and programming concepts.	The settings and operation of the CPU Unit and programming concepts for motion control are described.
NJ/NX-series Motion Control Instructions Reference Manual	W508	NX701-□□□□ NX102-□□□□ NX1P2-□□□□ NJ501-□□□□ NJ301-□□□□ NJ101-□□□□	Learning about the specifications of the motion control instructions.	The motion control instructions are described.
NJ/NX-series CPU Unit Built-in EtherCAT® Port User's Manual	W505	NX701-□□□□ NX102-□□□□ NX1P2-□□□□ NJ501-□□□□ NJ301-□□□□ NJ101-□□□□	Using the built-in EtherCAT port on an NJ/NX-series CPU Unit.	Information on the built-in EtherCAT port is provided. This manual provides an introduction and provides information on the configuration, features, and setup.
NJ/NX-series CPU Unit Built-in EtherNet/IP™ Port User's Manual	W506	NX701-□□□□ NX102-□□□□ NX1P2-□□□□ NJ501-□□□□ NJ301-□□□□ NJ101-□□□□	Using the built-in EtherNet/IP port on an NJ/NX-series CPU Unit.	Information on the built-in EtherNet/IP port is provided. Information is provided on the basic setup, tag data links, and other features.
NX-series CPU Unit FINS Function User's Manual	W596	NX701-□□20 NX102-□□□□	Using the FINS function of an NX-series CPU Unit.	Describes the FINS function of an NX-series CPU Unit.

Manual name	Cat. No.	Model numbers	Application	Description
NX/NY-series Artificial Intelligence Machine Automation Controller User's Manual	W594	NX701-Z□□□ NY532-Z□□□ NY512-Z□□□	Learning about the NX/NY-series Artificial Intelligence Machine Automation Controller.	Describes the NX/NY-series Artificial Intelligence Machine Automation Controller overview, AI function specifications, system start-up, maintenance, and error details.
AI Controller Standard Software Operation Manual	W611	SYSMAC-AICSTE□□L	Learning an introduction of the AI Controller standard software and how to use it.	An introduction of the AI Controller standard software (AI Operator, AI Viewer), installation procedures, basic operations, connection operations, and operating procedures for main functions are described.
Sysmac Library AI Predictive Maintenance Library User's Manual	W610	SYSMAC-ZPA□□□000W	Learning about AI predictive maintenance library and FB specifications.	Information necessary to use AI predictive maintenance library is provided.
NJ/NX-series Troubleshooting Manual	W503	NX701-□□□□ NX102-□□□□ NX1P2-□□□□ NJ501-□□□□ NJ301-□□□□ NJ101-□□□□	Learning about the errors that may be detected in an NJ/NX-series Controller.	Concepts on managing errors that may be detected in an NJ/NX-series Controller and information on individual errors are described.
Sysmac Studio Version 1 Operation Manual	W504	SYSMAC -SE2□□□	Learning about the operating procedures and functions of the Sysmac Studio.	Describes the operating procedures of the Sysmac Studio.
NY-series IPC Machine Controller Industrial Panel PC Hardware User's Manual	W557	NY532-1□□□	Learning the basic specifications of the NY-series Industrial Panel PCs, including introductory information, designing, installation, and maintenance. Mainly hardware information is provided.	An introduction to the entire NY-series system is provided along with the following information on the Industrial Panel PC. • Features and system configuration • Introduction • Part names and functions • General specifications • Installation and wiring • Maintenance and inspection
NY-series IPC Machine Controller Industrial Box PC Hardware User's Manual	W556	NY512-1□□□	Learning the basic specifications of the NY-series Industrial Box PCs, including introductory information, designing, installation, and maintenance. Mainly hardware information is provided.	An introduction to the entire NY-series system is provided along with the following information on the Industrial Box PC. • Features and system configuration • Introduction • Part names and functions • General specifications • Installation and wiring • Maintenance and inspection
NY-series IPC Machine Controller Industrial Panel PC / Industrial Box PC Setup User's Manual	W568	NY532-1□□□ NY512-1□□□	Learning about initial setting of the NY-series Industrial PCs and preparations to use Controllers.	The following information is provided on an introduction to the entire NY-series system. • Two OS systems • Initial settings • Industrial PC Support Utility • NYCompolet • Industrial PC API • Backup and recovery

Manual name	Cat. No.	Model numbers	Application	Description
NY-series IPC Machine Controller Industrial Panel PC / Industrial Box PC Software User's Manual	W558	NY532-1□□□ NY512-1□□□	Learning how to program and set up the Controller functions of an NY-series Industrial PC.	The following information is provided on the NY-series Controller functions. • Controller operation • Controller features • Controller settings • Programming based on IEC 61131-3 language specifications
NY-series Instructions Reference Manual	W560	NY532-1□□□ NY512-1□□□	Learning detailed specifications on the basic instructions of an NY-series Industrial PC.	The instructions in the instruction set (IEC 61131-3 specifications) are described.
NY-series IPC Machine Controller Industrial Panel PC / Industrial Box PC Motion Control User's Manual	W559	NY532-1□□□ NY512-1□□□	Learning about motion control settings and programming concepts of an NY-series Industrial PC.	The settings and operation of the Controller and programming concepts for motion control are described.
NY-series Motion Control Instructions Reference Manual	W561	NY532-1□□□ NY512-1□□□	Learning about the specifications of the motion control instructions of an NY-series Industrial PC.	The motion control instructions are described.
NY-series IPC Machine Controller Industrial Panel PC / Industrial Box PC Built-in EtherCAT® Port User's Manual	W562	NY532-1□□□ NY512-1□□□	Using the built-in EtherCAT port in an NY-series Industrial PC.	Information on the built-in EtherCAT port is provided. This manual provides an introduction and provides information on the configuration, features, and setup.
NY-series IPC Machine Controller Industrial Panel PC / Industrial Box PC Built-in EtherNet/IP™ Port User's Manual	W563	NY532-1□□□ NY512-1□□□	Using the built-in EtherNet/IP port in an NY-series Industrial PC.	Information on the built-in EtherNet/IP port is provided. Information is provided on the basic setup, tag data links, and other features.
NY-series Troubleshooting Manual	W564	NY532-1□□□ NY512-1□□□	Learning about the errors that may be detected in an NY-series Industrial PC.	Concepts on managing errors that may be detected in an NY-series Controller and information on individual errors are described.

Revision History

A manual revision code appears as a suffix to the catalog number on the front and back covers of the manual.



Revision code	Date	Revised content
01	October 2018	Original production

1

AI Predictive Maintenance Library

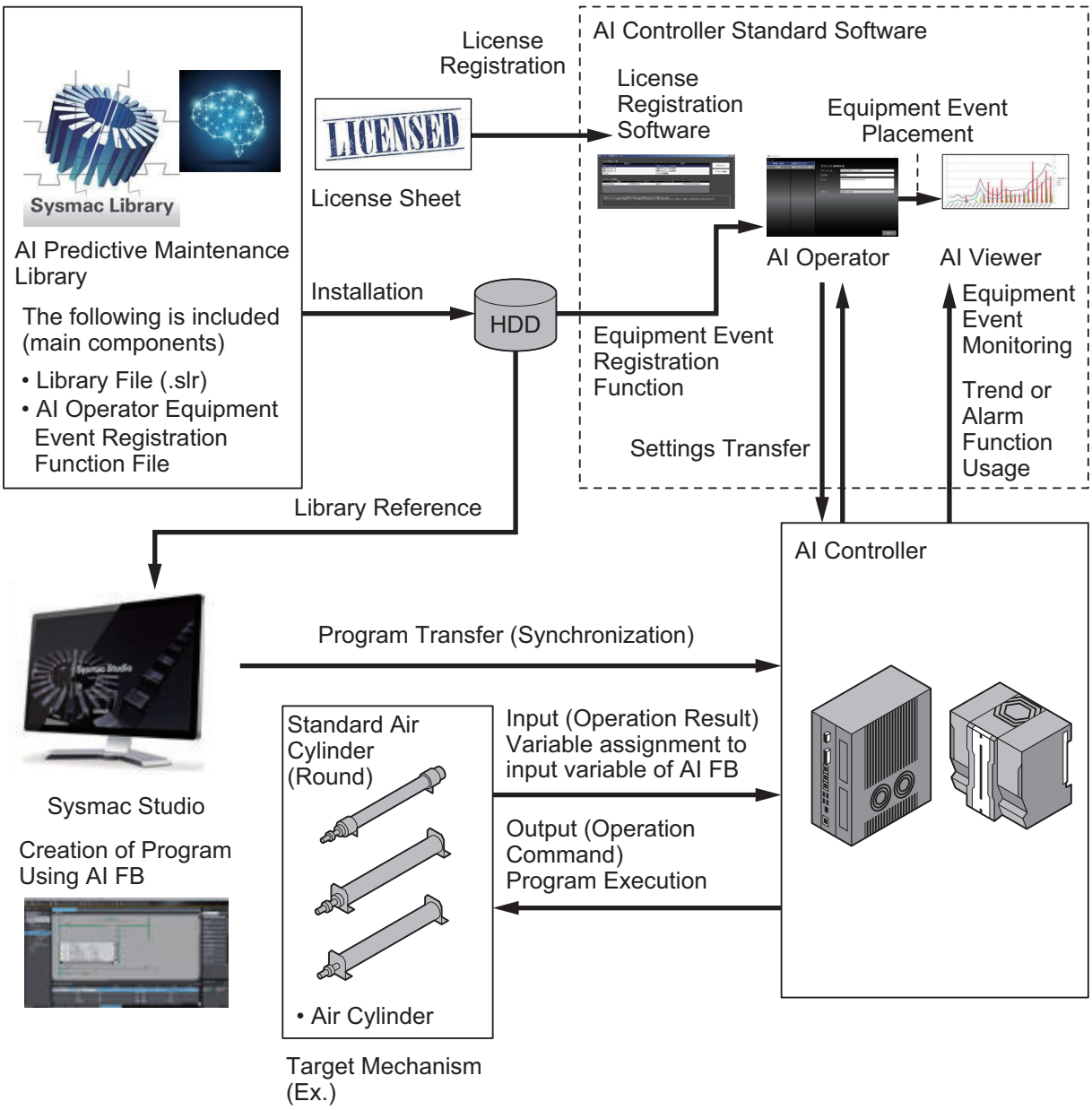
This section describes the shared specifications of each FB in the AI Predictive Maintenance Library.

1-1 Purpose of AI Predictive Maintenance Library

The purpose of the AI Predictive Maintenance Library is predictive maintenance for each mechanism (devices and components such as the Cylinder, Ball Screw, and Belt Pulley) to be used by utilizing the AI functions of the AI Controller.
The AI Predictive Maintenance Library contains FB libraries for each mechanism.
When the control and status data of the operating mechanism is input, function blocks of the AI Predictive Maintenance Library (hereinafter may be abbreviated as AI FB) generate mechanism state variables and execute the AI functions of the AI Controller. You can detect mechanism errors from the mechanism state variables generated by the AI functions.
Refer to *NX/NY-series Artificial Intelligence Machine Automation Controller User's Manual (Cat. No. W594)* for details on AI functions.

1-2 System Configuration

The system configuration related to the AI Predictive Maintenance Library is as follows.



For details on the operations in the above figure, refer to *1-3 Usage Procedure* on page 1 - 4.

1-3 Usage Procedure

This section describes the usage procedure for the AI Predictive Maintenance Library.
For details on starting up the AI Controller and the operations of the AI Controller Standard Software, refer to the *NX/NY-series Artificial Intelligence Machine Automation Controller User's Manual (Cat. No. W594)* and the *AI Controller Standard Software Operation Manual (Cat. No. W611)*.

1-3-1 Overview of Usage Procedure

The overview of the usage procedure for the AI Predictive Maintenance Library is as follows.
The usage procedure can be roughly divided into the following steps.

STEP 0: Preliminary preparation
STEP 1: Download and install the AI Predictive Maintenance Library and register the license
STEP 2: Integrate AI FB into the user program on the Sysmac Studio and transfer the program
STEP 3: Set the AI Operator and perform transfer
STEP 4: Generate and download the AI machine learning model
STEP 5: User program execution and visualization phase

1-3-2 Details of Usage Procedure

The details of the usage procedure for the AI Predictive Maintenance Library are as follows.

Procedure		Description	Reference
STEP 0: Preliminary preparation	0-1. Install the AI Operator	Install the AI Operator if it is not installed yet.	*AI Controller Standard Software Operation Manual (Cat. No. W611)*
	0-2. Apply the AI Option to the Sysmac Studio	In order to use the AI Controller with the Sysmac Studio, you will need the AI Option for the Sysmac Studio.	---
	0-3. Purchase the AI Predictive Maintenance Library license	Purchase a license for registering the AI Predictive Maintenance Library.	---
	0-4. Prepare the AI Controller	Prepare the AI Controller.	*NX/NY-series Artificial Intelligence Machine Automation Controller User's Manual (Cat. No. W594) Preliminary Preparation Phase*
STEP 1: Download and install the AI Predictive Maintenance Library and register the license	1-1. Download the AI Predictive Maintenance Library	Download the required AI Predictive Maintenance Library from the License Registration Software.	*1-5 Download and License Registration Procedure* on page 1 - 8

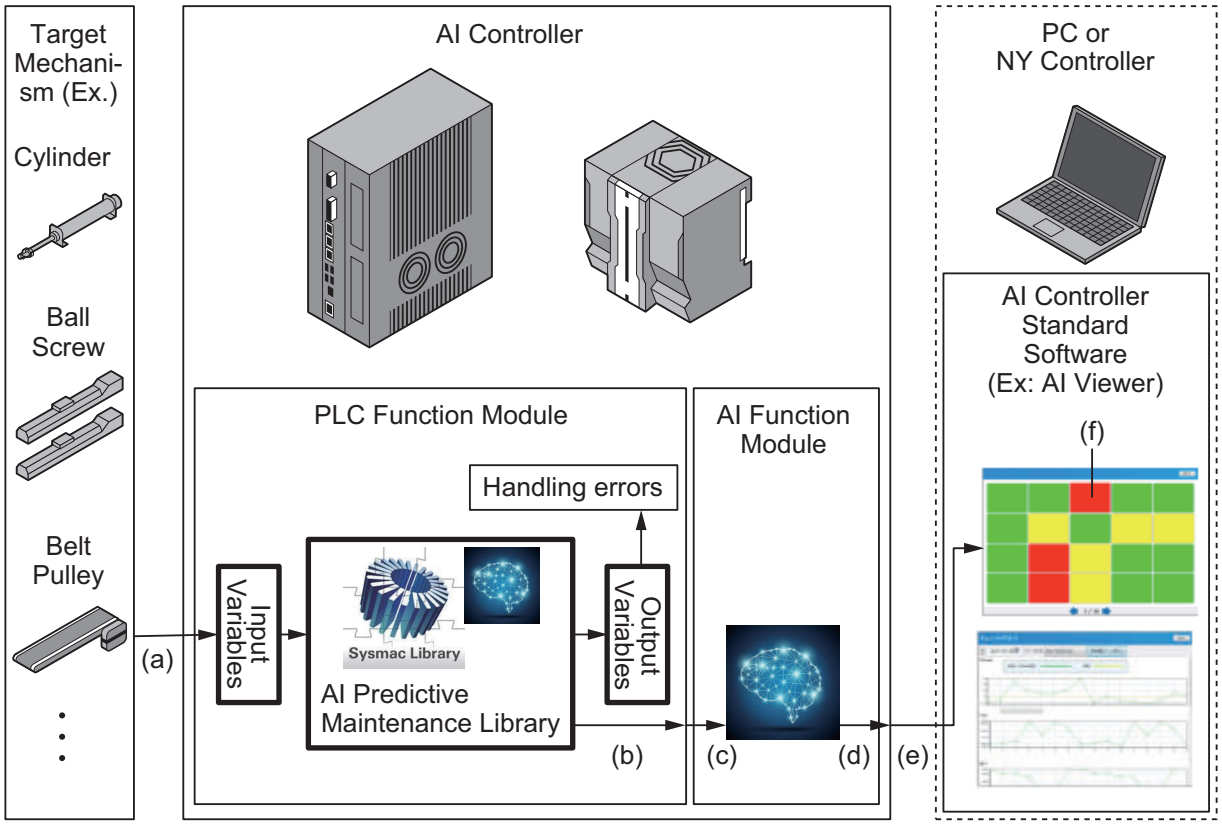
Procedure		Description	Reference
	1-2. Install the AI Predictive Maintenance Library	Exit the License Registration Software and install the AI Predictive Maintenance Library from the installer.	*2-1 Procedure to Use Sysmac Library Installed Using the Installer* on page 2 - 2
	1-3. Register the AI Predictive Maintenance Library license	Register the license of the AI Predictive Maintenance Library from the License Registration Software.	*1-5 Download and License Registration Procedure* on page 1 - 8
STEP 2: Integrate AI FB into the user program on the Sysmac Studio and transfer the program	2-1. Create user program for the Feature Value/ Machine Learning Function	Create a user program that performs the run and idle commands for the Feature Extraction Function and the Machine Learning Function.	*NX/NY-series Artificial Intelligence Machine Automation Controller User's Manual (Cat. No. W594) Data Utilization Phase* in *Start-up Procedures for the AI Controller*
	2-2. Place AI FB	Place the AI FB in the user program.	*2-2 How to use Sysmac Library in the CPU Unit or Industrial PC* on page 2 - 6
	2-3. Assign input data to input variables	Assign the information required for equipment events to the input variables of the AI FB.	Individual specifications of each FB
	2-4. Process output variables	Process the output variables of the AI FB appropriately in the user program.	Individual specifications of each FB
	2-5. Transfer the user program to the AI Controller	Perform the settings to prepare the AI Controller, and then transfer the user program and settings to the AI Controller using the synchronization function of the Sysmac Studio. *1	---
STEP 3: Set the AI Operator and perform transfer	3-1. Set and confirm the variable data and equipment events	By using the Equipment Event Registration function of the AI Operator, check whether there are any errors due to the version of the AI Predictive Maintenance Library.	*AI Controller Standard Software Operation Manual (Cat. No. W611)*
	3-2. Transfer the settings to the AI Controller	Transfer the settings of the AI Operator to the AI Controller.	
STEP 4: Generate and download the AI machine learning model		This phase is performed by the OMRON engineering department. Download the generated AI machine learning model to the AI Controller using the AI Operator.	---

Procedure		Description	Reference
STEP 5: User program execution and visualization phase	5-1. Execute the program	Execute the program transferred in STEP 2.	*NX/NY-series Artificial Intelligence Machine Automation Controller User's Manual (Cat. No. W594) Data Utilization Phase* in *Startup Procedures for the AI Controller*
	5-2. Prepare the AI Operator and check operation of the AI functions of the AI Controller and the user program	Prepare the AI Operator, make sure that the Data Collection and the Feature Value/ Machine Learning Function of the AI Controller are in operation, and then check the operation of the user program.	
	5-3. Visualization with the AI Viewer	Monitor the equipment events with the AI Viewer.	*AI Controller Standard Software Operation Manual (Cat. No. W611)*

*1. Programs related to the AI functions will not work properly unless the AI Operator settings are configured and transferred in the next step.
Be sure to check the operation after performing the next step.

1-4 Relationship of System Configuration Elements

The relationship among the AI Predictive Maintenance Library, AI Function Module as a system configuration element, and AI Controller Standard Software is as follows.



(a) Input: The user assigns the mechanism state to be monitored to the input variables of the AI FB.

(b) Output: Execution errors, etc. from the AI FB are handled.
The result of executing the AI FB is automatically input to the AI Function Module.
The user does not need to be conscious of the output to the AI functions.

(c) Input: The output of the AI Predictive Maintenance Library is automatically input to the AI Function Module.
The user does not need to be conscious of the input.
However, it is necessary for the AI Function Module to be operational beforehand.

(d) Output: The AI Function Module detects signs of an error and outputs an equipment event.

(e) Input: When you execute the AI Viewer of the AI Controller Standard Software, the equipment event output by the AI Function Module is input to the AI Viewer.

(f) Output: The equipment event is displayed on the AI Viewer of the AI Controller Standard Software.
When a sign of an abnormality is detected, a warning is output in the equipment event.

1-5 Download and License Registration Procedure

This section describes the procedure to download and perform license registration of the AI Predictive Maintenance Library.

Precautions for Correct Use

- The License Registration Software is required in order to download the AI Predictive Maintenance Library and register the license.
  Make sure that the License Registration Software is installed on the PC.
  The License Registration Software is included in the AI Controller Standard Software. If the License Registration Software is not installed, install the AI Controller Standard Software.
- You must be connected to the internet to download the AI Predictive Maintenance Library.
- A Sysmac ID and license for the AI Prediction Maintenance Library are required to download the AI Predictive Maintenance Library.

1-5-1 Download Procedure for AI Predictive Maintenance Library

1. Select **All Programs** – **OMRON** – **AI Controller Standard Software** – **AI license registration software** from the Windows Start Menu.
   The main screen of the License Registration Software is displayed.
2. Click the **Download** button.
   The browser starts and the OMRON download web page is displayed.
3. Download the required AI Predictive Maintenance Library from the web page.

Precautions for Correct Use

To install the downloaded AI Predictive Maintenance Library, close the License Registration Software.

1-5-2 License Registration Procedure for AI Predictive Maintenance Library

1. Open the AI License Registration Software.
2. Select the product for which license registration is performed from the **AI Predictive maintenance library list** on the main screen.
3. Click the **Registration** button on the main screen.
   The **Input license key** dialog box appears.
4. Enter the license number and press **OK**.

An error message will be displayed if the license number is invalid, etc. When the license registration is successful, the registered license information will be displayed in the **License information** list.

1-6 Importing Set Values for Mechanism Settings

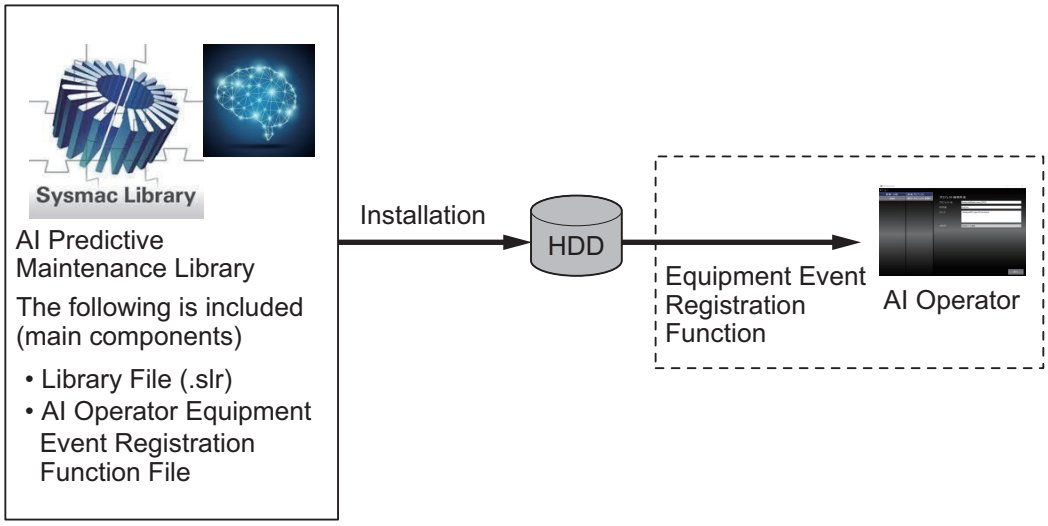
This section describes set values to be imported for the purpose of simplifying the mechanism settings on the AI Operator.

Specifically, the following items are set on the AI Operator according to the mechanism for which the AI Predictive Library is installed.

- Equipment event names
- Frame variables
- Variable data
- Subframe variables
- Feature extraction output frame variables
- Machine learning output frame variables

For details on the Equipment Event Registration function of the AI Operator, refer to the *AI Controller Standard Software Operation Manual (Cat. No. W611)*.

The relationship of the AI Operator Equipment Event Registration function is shown below with the system configuration.



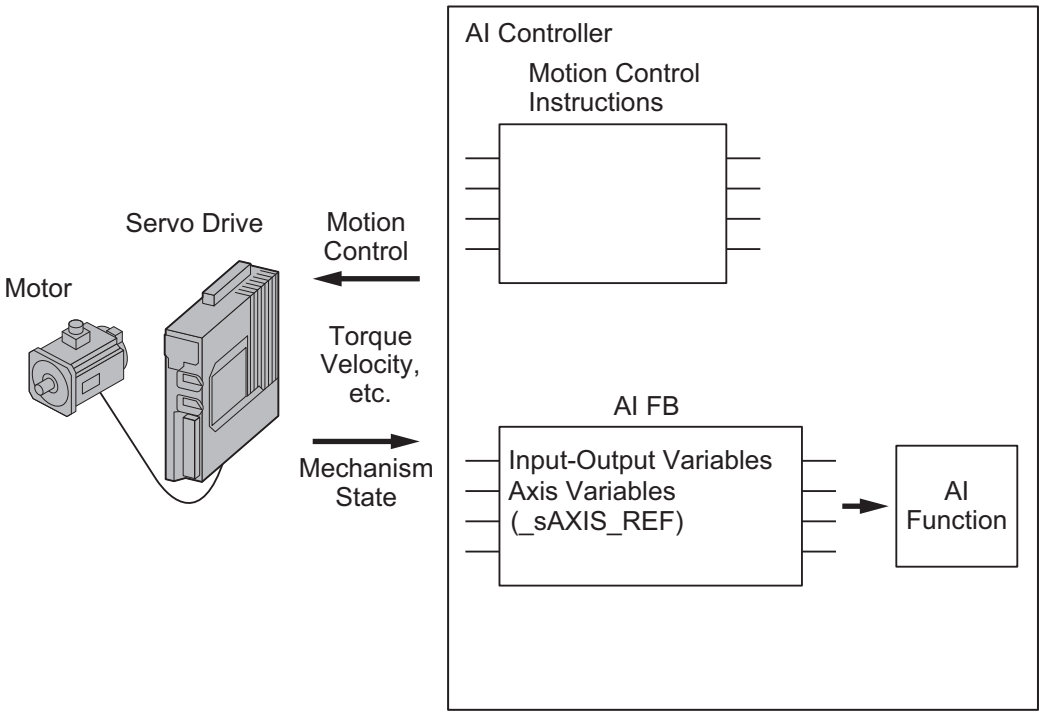
The following equipment event names are displayed on the AI Operator for respective mechanisms for which the AI Predictive Maintenance Library is installed.

Mechanism	Equipment event name
Cylinder	Cylinder velocity error
Ball Screw	Ball screw velocity error
Belt Pulley	Belt pulley torque error

1-7 Mechanisms of AI FB Linked to Motion Control

This section describes the mechanisms of the AI FB that are linked to motion control.
Examples of the mechanisms of the AI FB linked to motion control include the Ball Screw, Belt Pulley, and so on.

The relationship between motion control and the AI FB is shown in the diagram below.



The conditions required for each of the above factors are as follows.

Element	Required conditions	Remarks
Motor	It must be a motor that can be used with the following Servo Drive. Example: OMRON 1S, G5-series Servomotor	---
Servo Drive*1	• It must be a Servo Drive that can be used with the Motion Control Function Module. • It must be a servo drive that can acquire the following *mechanism states*. Example: OMRON 1S, G5-series Servo Drive	---
Mechanism states	The following states must be reflected in the *_sAXIS_REF* axis variable input to the AI FB (n: 0 to 255). *1 • _MC_AX[n].Cmd.Pos (Command Current Position) • _MC_AX[n].Cmd.Vel (Command Current Velocity) • _MC_AX[n].Cmd.Trq (Command Current Torque) *2 • _MC_AX[n].Act.Pos (Actual Current Position) • _MC_AX[n].Act.Vel (Actual Current Velocity) • _MC_AX[n].Act.Trq (Actual Current Torque)	In a system configuration in which the mechanism state is not reflected directly in the axis variable for motion control, be sure to take measures such as assigning the mechanism state to the AI FB input-output variable after reflecting the value on the left to the axis variable. *2

Element	Required conditions	Remarks
AI FB	• The *mechanism states* above must be input as input-output variables (axis variables) of the AI FB. • The AI FB must be located in the primary periodic task. • The AI FB must be executed while the AI function is in the Enabled state.	---
Motion Control Instruction*1	Refer to *Applicable Motion Control Instructions* on page 1 - 12.	---

*1. For details on the Servo Drive, axis variables, and motion control, refer to the *NJ/NX-series Motion Control Instructions Reference Manual (Cat. No. W508)* or the *NY-series Motion Control Instructions Reference Manual (Cat. No. W561)*.

*2. The actual current torque value obtained by skipping one task period is assigned to the command current torque. The following table shows an example.

Time	t0	t1	t2
Actual current torque: _MC_AX[n].Act.Trq	Trq(t0)	Trq(t1)	Trq(t2)
Command current torque: _MC_AX[n].Cmd.Trq	0	Trq(t0)	Trq(t1)

For the time in the table above, t0 is the time executed a program in a task period, t1 is the time executed a program in the next task period, and t2 is the time executed a program in the task period following that.

● Applicable Motion Control Instructions

The motion control instructions that can be used are as shown below.

Item	Instruction name	Function
Axis command	MC_Move	Positioning
	MC_MoveAbsolute	Absolute positioning
	MC_MoveRelative	Relative positioning
Axes group command	MC_MoveLinear	Linear interpolation
	MC_MoveLinearAbsolute	Absolute linear interpolation
	MC_MoveLinearRelative	Relative linear interpolation
	MC_MoveCircular2D	Circular 2D interpolation

1-8 Precautions for Correct Use

- Evaluate the system by running the equipment to check if the equipment events are correctly detected. The expected equipment events may not be detected depending on the settings of the AI Controller.
- If you define the same AI FB in a different POU, specify a different instance name.
  If you specify the same instance name, the AI Operator will not operate correctly.
  The conditions that determine the same instance name include the case that the capitalization of instance names differs.
  The following table shows an example.

	POU name	
	POU1	**POU2**
Example 1 when you can not set	AI FB: Cylinder and Instance Name: Inst1	AI FB: Cylinder and Instance Name: Inst1
Example 2 when you can not set	AI FB: Cylinder and Instance Name: INST1	AI FB: Cylinder and Instance Name: Inst1
Example 1 when you can set	AI FB: Cylinder and Instance Name: Inst1	AI FB: Ball Screw and Instance Name: Inst1
Example 2 when you can set	AI FB: Cylinder and Instance Name: Inst1	AI FB: Cylinder and Instance Name: Inst2

- Do not change the instance name of an AI FB that has been registered by the AI Operator.
- Change the state of the Feature Extraction Function and the Machine Learning Function to operating before executing the AI FB. If the AI FB is executed without these states being in operation, the AI FB will not display an error, but the equipment events may not be detected correctly.

Precautions for Safe Use

- Be sure that the AI functions are in the RUN status before executing the FB.
- Execute this function block as the primary periodic task. If you do not execute this function block as the primary periodic task, the mechanism state variables will not be generated correctly.

2

Sysmac Library Usage Procedure

The section describes the procedure to use Sysmac Library installed using the installer, and Sysmac Library in the CPU unit or Industrial PC.

2-1 Procedure to Use Sysmac Library Installed Using the Installer

This section describes the procedure to use Sysmac Library installed using the installer.
There are two ways to use libraries.

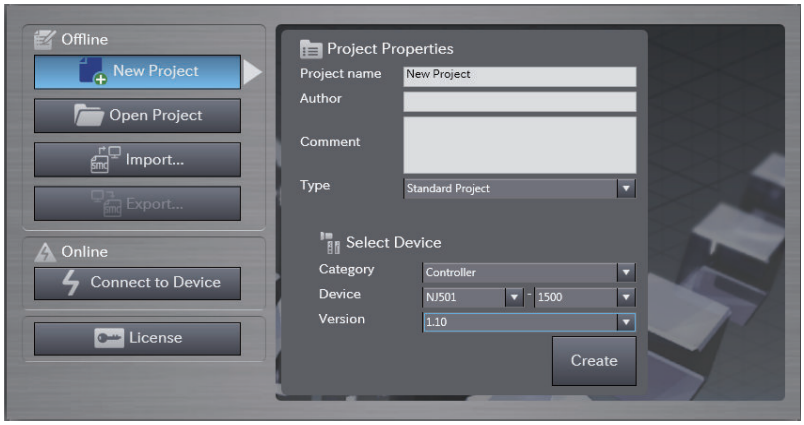
- Using a newly installed Sysmac Library
- Using an upgraded Sysmac Library

Version Information

For the Controller models and versions of the Sysmac Studio for which this library can be used, refer to *Applicable Products* on page 1.

2-1-1 Using a Newly Installed Sysmac Library

1 Start the Sysmac Studio and open a project using Sysmac Library, or create a new one.

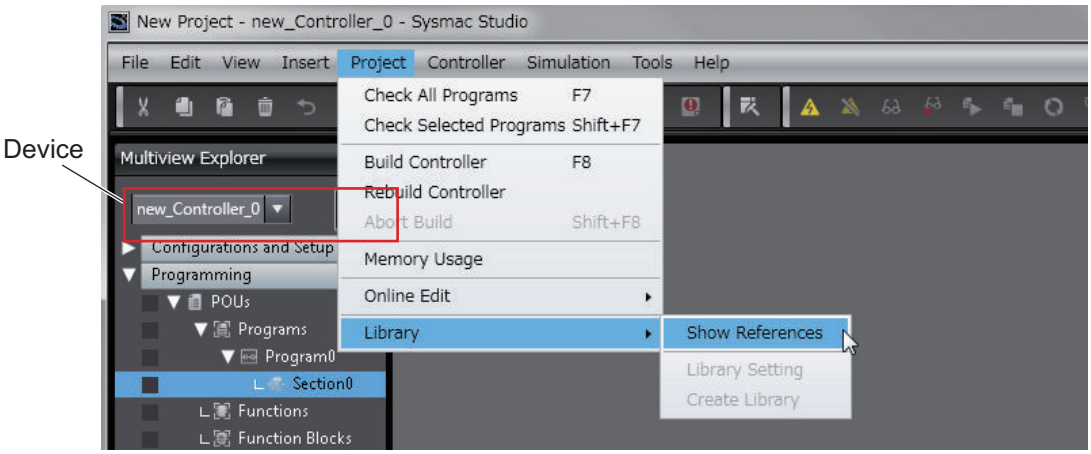


Precautions for Correct Use

If you create a new project, be sure to configure the settings as follows to enable use of the Sysmac Library. Without the settings below, you cannot proceed to Step 2 and later steps.

- Set the project type to Standard Project or Library Project.
- Set the device category to Controller.
- For the setting of Controller and Version in the Select Device section, refer to *Applicable Products* on page 1.

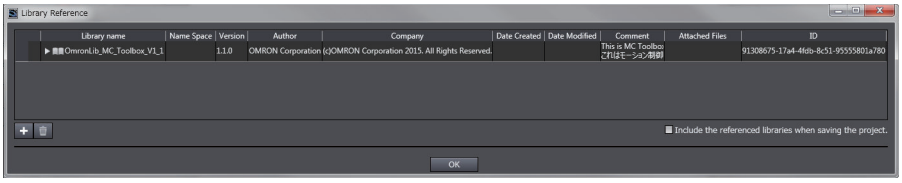
2 Select **Project** - **Library** - **Show References**.



Precautions for Correct Use

If you have multiple devices registered in the project, make sure that the currently selected device is the NJ/NX-series CPU Unit or NY-series Industrial PC. If the NJ/NX-series CPU Unit or NY-series Industrial PC is not selected, the menu for browsing the library will not appear. When the selected device is the NJ/NX-series CPU Unit or NY-series Industrial PC, the device icon displayed in Multiview Explorer changes to .

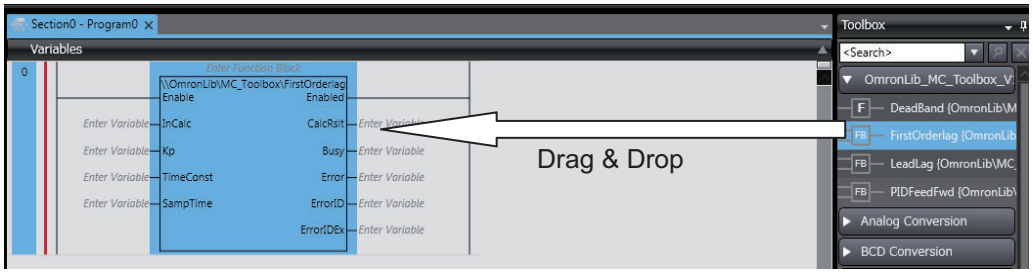
3 Add Sysmac Library to the list and click **OK**.



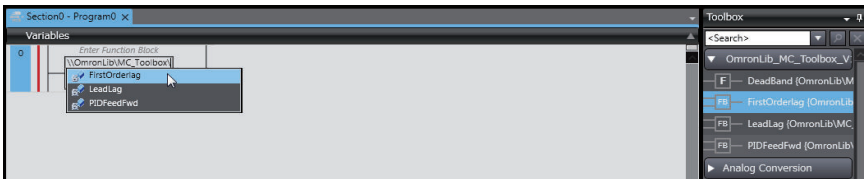
Sysmac Library is read into the project.
Now, when you select the Ladder Editor or ST Editor, the function blocks and functions included in the Sysmac Library appear in the Toolbox.
For the procedure for adding and setting libraries in the above screen, refer to *Sysmac Studio Version 1 Operation Manual (Cat. No. W504)*.

4 Insert the Sysmac Library's function blocks and functions into the circuit using one of the following two methods.

- Select the desired function block or function in the Toolbox and drag and drop it onto the Ladder Editor.



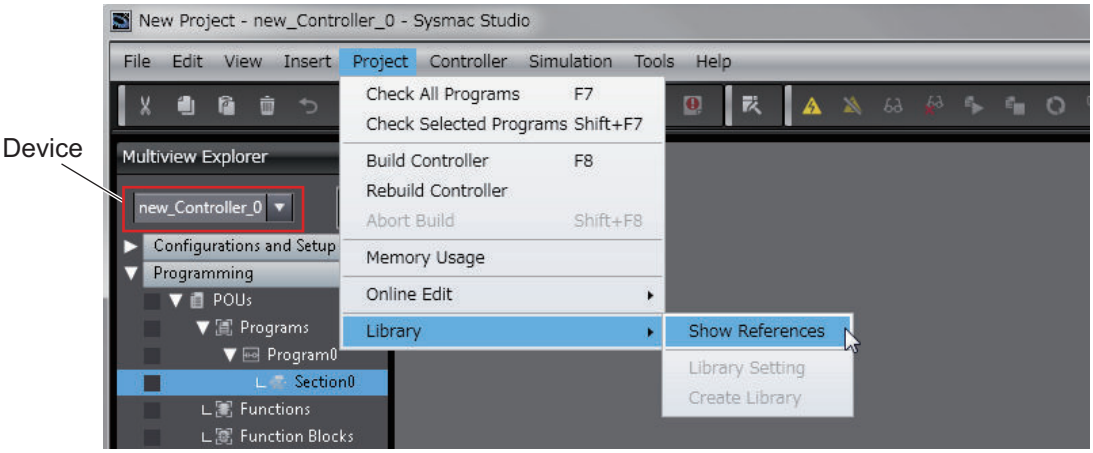
- Right-click the Ladder Editor, select **Insert Function Block** in the menu, and enter the fully qualified name (¥¥namespacename¥FBname).



2-1-2 Using an Upgraded Sysmac Library

1 Start Sysmac Studio and open a project in which any old-version Sysmac Library is included.

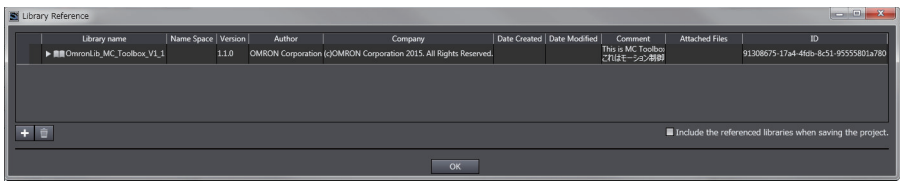
2 Select **Project** - **Library** - **Show References**.



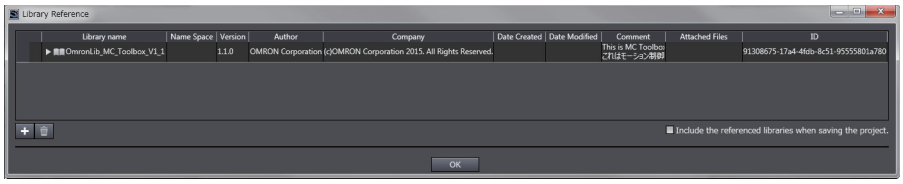
Precautions for Correct Use

If you have multiple devices registered in the project, make sure that the currently selected device is the NJ/NX-series CPU Unit or NY-series Industrial PC. If the NJ/NX-series CPU Unit or NY-series Industrial PC is not selected, the menu for browsing the library will not appear. When the selected device is the NJ/NX-series CPU Unit or NY-series Industrial PC, the device icon displayed in Multiview Explorer changes to  .

3 Select an old-version Sysmac Library and click the **Delete Reference** Button.



4 Add Sysmac Library to the list and click **OK**.



Precautions for Correct Use

Upgrade the Sysmac Library version, and then execute All Program Check, and confirm that there are no errors in the Build Window Program Check results.
From the Main Menu, select **Project** - **All Program Check**.

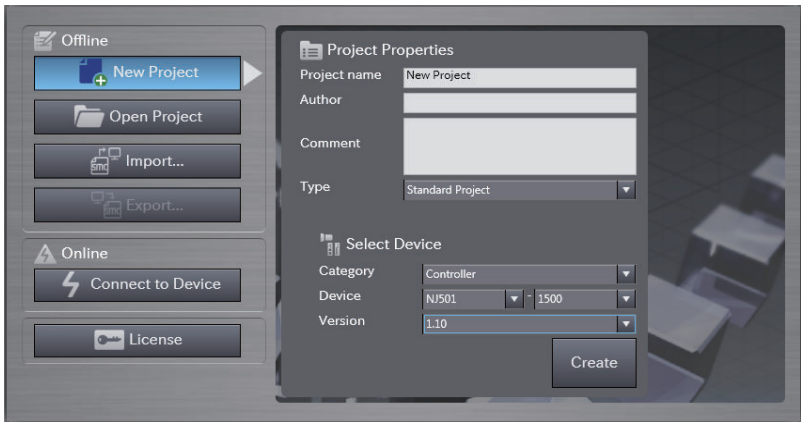
2-2 How to use Sysmac Library in the CPU Unit or Industrial PC

Even when Sysmac Library is not installed on your computer, you can use Sysmac Library by uploading it from the CPU Unit or Industrial PC to your computer.
The procedure to use Sysmac Library in the CPU Unit or Industrial PC is as follows.

Version Information

For the versions of the Sysmac Studio for which this library can be used, refer to *Applicable Products* on page 1.

1 Start the Sysmac Studio and create a new project in which you want to use Sysmac Library.

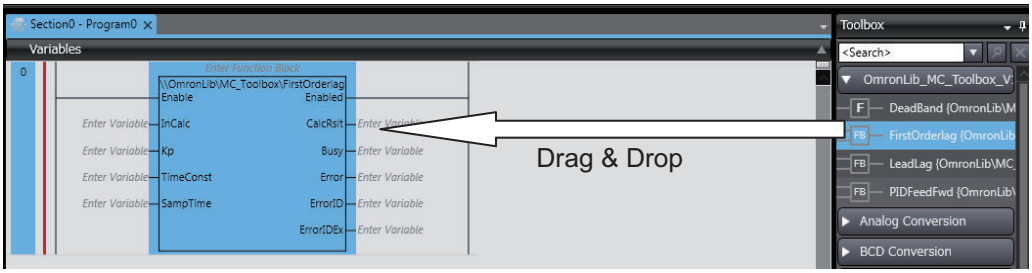


2 Connect online to the CPU Unit or Industrial PC.

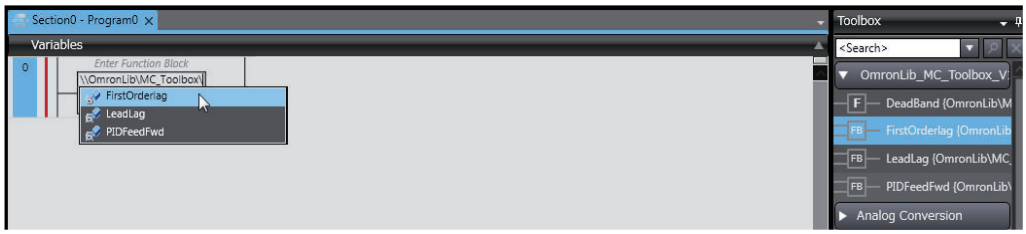
3 Upload the POUs in which Sysmac Library is used.
Now, when you select the Ladder Editor or ST Editor, the function blocks and functions included in the Sysmac Library used in the uploaded POUs appear in the Toolbox.

4 Insert the Sysmac Library's function blocks and functions into the circuit using one of the following two methods.

- Select the desired function block or function in the Toolbox and drag and drop it onto the Ladder Editor.



- Right-click the Ladder Editor, select **Insert Function Block** in the menu, and enter the fully qualified name (¥¥namespacename¥FBname).



Precautions for Correct Use

- The Sysmac Studio installs Sysmac Library library files to the specified folder on the computer if they are not present. However, the Sysmac Studio does not install libraries to the specified folder on the computer if they are present.
  The specified folder here means the folder in which library files are installed by the installer.
- Note that uploading Sysmac Library from a CPU Unit or Industrial PC does not install the manual and help files for Sysmac Library, unlike installation using the installer. Please install the manual and help files using the installer if you need them.

3

Common Specifications of FB/FUN

This section describes the specifications that are common to each FB/FUN in the Sysmac Library.

3-1 Common Variables

This section describes the specifications of variables (EN, Execute, Enable, Abort, ENO, Done, CalcRslt, Enabled, Busy, CommandAborted, Error, ErrorID, and ErrorIDEx) that are used for more than one function or function block. The specifications are described separately for functions, for execute-type function blocks, and for enable-type function blocks.

3-1-1 Definition of Input Variables and Output Variables

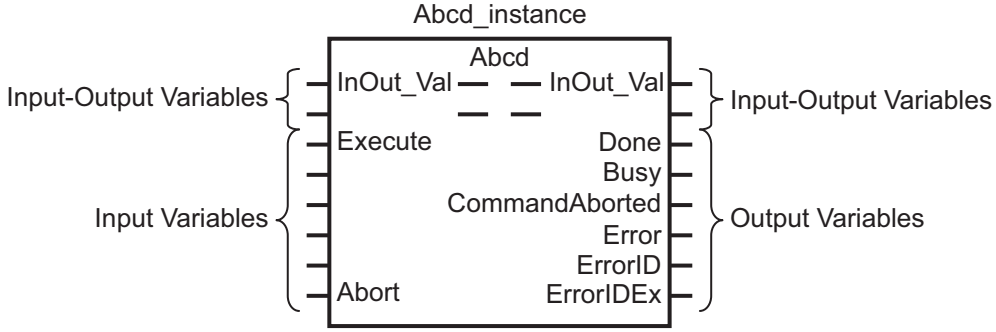
Common input variables and output variables used in functions and function blocks are as follows.

Variable	I/O	Data type	Function/function block type to use			Meaning	Definition
			Function block		Function		
			Execute-type	Enable-type			
EN	Input	BOOL			OK	Execute	The processing is executed while the variable is TRUE.
Execute		BOOL	OK			Execute	The processing is executed when the variable changes to TRUE.
Enable		BOOL		OK		Run	The processing is executed while the variable is TRUE.
Abort		BOOL	OK			Abort	The processing is aborted. You can select the aborting method.
ENO	Output	BOOL			OK	Done	The variable changes to TRUE when the processing ends normally. It is FALSE when the processing ends in an error, the processing is in progress, or the execution condition is not met.
Done		BOOL	OK			Done	The variable changes to TRUE when the processing ends normally. It is FALSE when the processing ends in an error, the processing is in progress, or the execution condition is not met.
Busy		BOOL	OK	OK		Executing	The variable is TRUE when the processing is in progress. Turns to FALSE while the process is not being executed.
CalcRslt		LREAL		OK		Calculation Result	The calculation result is output.

Variable	I/O	Data type	Function/function block type to use			Meaning	Definition
			Function block		Function		
			Execute-type	Enable-type			
Enabled		BOOL		OK		Enabled	The variable is TRUE when the output is enabled. It is used to calculate the control amount for motion control, temperature control, etc.
Command Aborted		BOOL	OK			Command Aborted	The variable changes to TRUE when the processing is aborted. It changes to FALSE when the processing is executed the next time again.
Error		BOOL	OK	OK		Error	This variable is TRUE while there is an error. It is FALSE when the processing ends normally, the processing is in progress, or the execution condition is not met.
ErrorID		WORD	OK	OK		Error Code	An error code is output.
ErrorIDEx		DWORD	OK	OK		Expansion Error Code	An expansion error code is output.

3-1-2 Execute-type Function Blocks

- Processing starts when Execute changes to TRUE.
- When Execute changes to TRUE, Busy also changes to TRUE. When processing is completed normally, Busy changes to FALSE and Done changes to TRUE.
- When continuously executing function blocks of the same instance, change the next Execute to TRUE for at least one task period after Done changes to FALSE in the previous execution.
- If the function block has a CommandAborted (Instruction Aborted) output variable and processing is aborted, CommandAborted changes to TRUE and Busy changes to FALSE.
- If an error occurs in the function block, Error changes to TRUE and Busy changes to FALSE.
- For function blocks that output the result of calculations for motion control and temperature control, you can use the BOOL input variable Abort to abort the FB process. When Abort changes to TRUE, CommandAborted changes to TRUE and the execution of the function block is aborted.

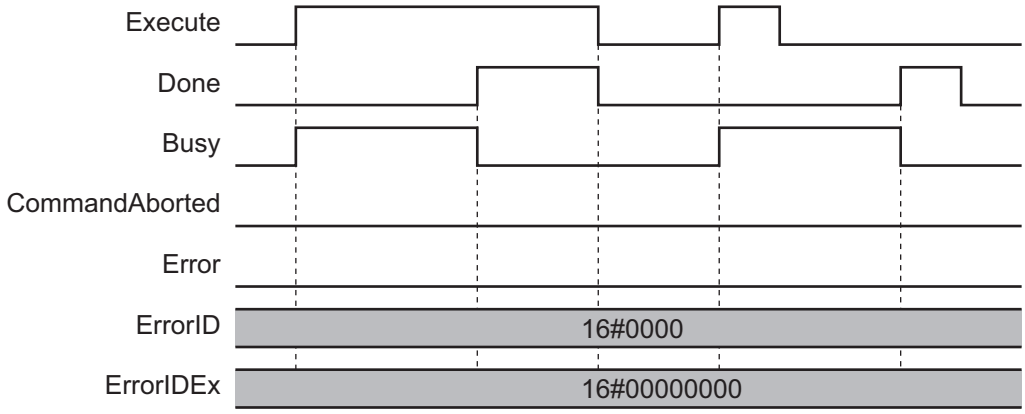


- If Execute is TRUE and Done, CommandAborted, or Error changes to TRUE, Done, CommandAborted, or Error changes to FALSE when Execute is changed to FALSE.
- If Execute is FALSE and Done, CommandAborted, or Error changes to TRUE, Done, CommandAborted, or Error changes to TRUE for only one task period.
- If an error occurs, the relevant error code and expansion error code are set in ErrorID (Error Code) and ErrorIDEx (Expansion Error Code). The error codes are retained even after Error changes to FALSE, but ErrorID is set to 16#0000 and ErrorIDEx is set to 16#0000 0000 when Execute changes to TRUE.

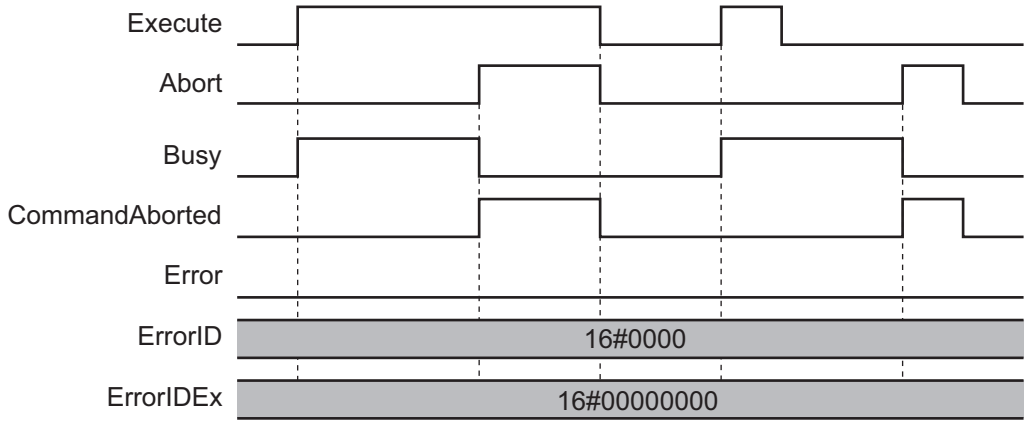
Timing Chart

This section provides timing charts for a normal end, canceled execution, aborted execution, and errors.

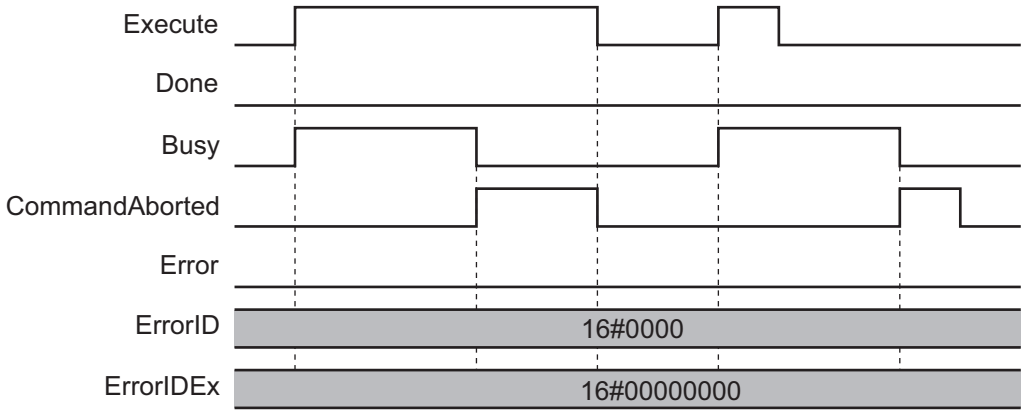
● Normal End



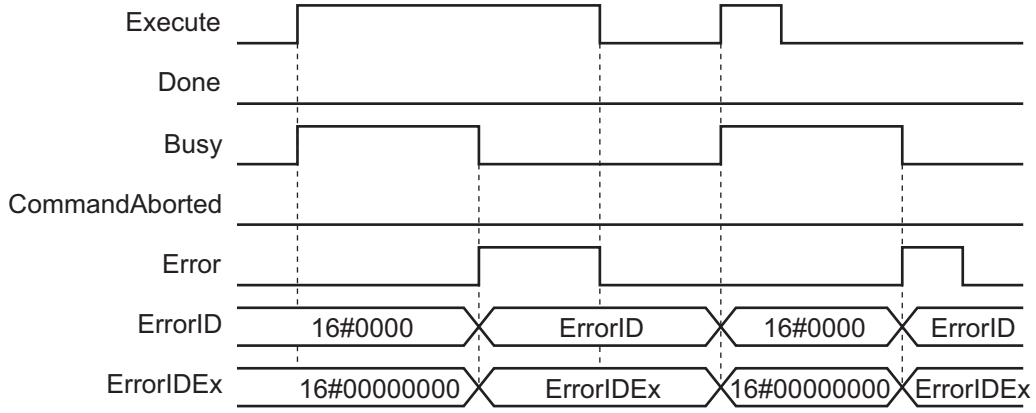
● Canceled Execution



● **Aborted Execution**

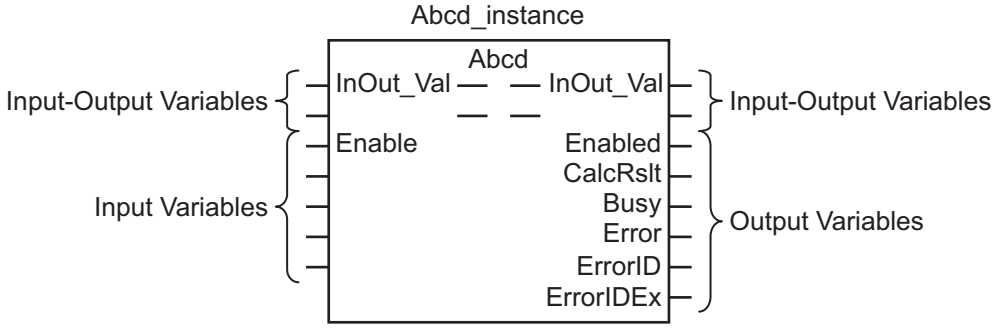


● **Errors**



3-1-3 Enable-type Function Blocks

- Processing is executed while Enable is TRUE.
- When Enable changes to TRUE, Busy also changes to TRUE. Enabled is TRUE during calculation of the output value.
- If an error occurs in the function block, Error changes to TRUE and Busy and Enabled change to FALSE. When Enable changes to FALSE, Enabled, Busy, and Error change to FALSE.



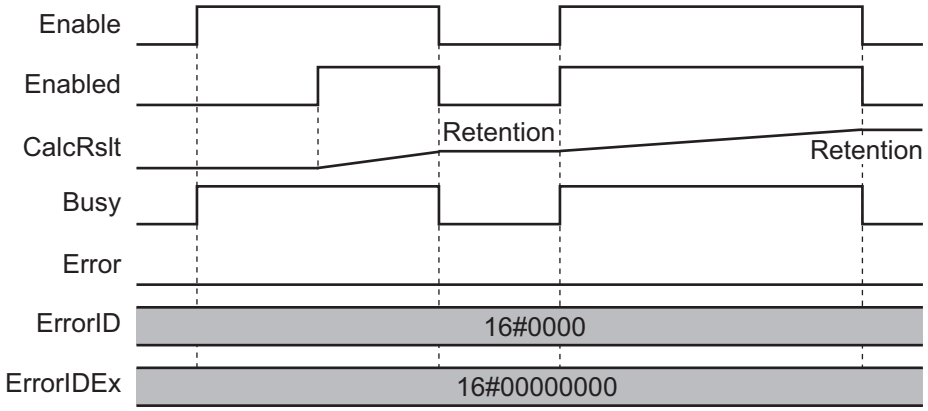
- If an error occurs, the relevant error code and expansion error code are set in ErrorID (Error Code) and ErrorIDEx (Expansion Error Code). The error codes are retained even after Error changes to FALSE, but ErrorID is set to 16#0000 and ErrorIDEx is set to 16#0000 0000 when Execute changes to TRUE.

- For function blocks that calculate the control amount for motion control, temperature control, etc., Enabled is FALSE when the value of CalcRslt (Calculation Result) is incorrect. In such a case, do not use CalcRslt. In addition, after the function block ends normally or after an error occurs, the value of CalcRslt is retained until Enable changes to TRUE. The control amount will be calculated based on the retained CalcRslt value, if it is the same instance of the function block that changed Enable to TRUE. If it is a different instance of the function block, the control amount will be calculated based on the initial value.

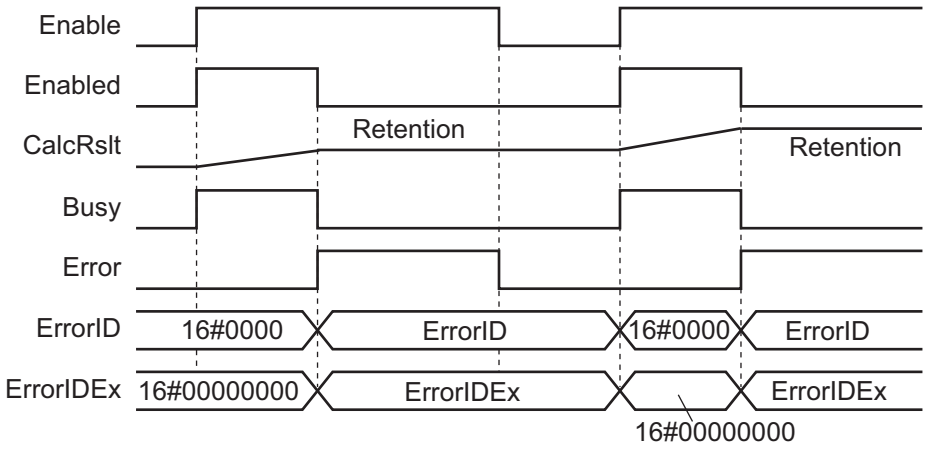
Timing Charts

This section provides timing charts for a normal end and errors.

● Normal End



● Errors



3-2 Precautions

This section provides precautions for the use of this function block.

3-2-1 Nesting

You can nest calls to this function block for up to four levels.
Refer to *NJ/NX-series CPU Unit Software User's Manual (Cat. No. W501)* or *NY-series IPC Machine Controller Industrial Panel PC / Industrial Box PC Software User's Manual (Cat. No. W558)* for details on the nesting function block.

3-2-2 Instruction Options

You cannot use the upward differentiation option for this function block.

3-2-3 Re-execution of Function Blocks

Execute-type function blocks cannot be re-executed by the same instance.
If you do so, the output value will be the initial value.
Refer to *NJ/NX-series CPU Unit Motion Control User's Manual (Cat. No. W507)* or *NY-series IPC Machine Controller Industrial Panel PC / Industrial Box PC Motion Control User's Manual (Cat. No. W559)* for details on re-execution.

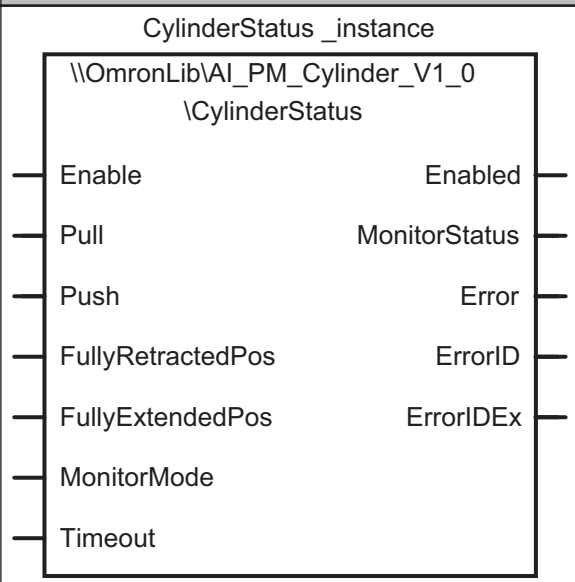
4

Individual Specifications of Function Block (Cylinder)

This section describes the individual specifications (cylinder) of the AI Predictive Maintenance Library.

CylinderStatus

CylinderStatus generates mechanism state variables that reflect the state of the cylinder, and those variables are referenced by the Feature Value/Machine Learning Function.

FB name	Name	FB/ FUN	Graphic expression	ST expression
Cylinder-Status	Cylinder Status Output	FB		CylinderStatus _in-stance(Enable:=, Pull:=, Push:=, FullyRetracted-Pos:=, FullyExtended-Pos:=, MonitorMode:=, Timeout:=, Enabled=>, MonitorStatus=>, Error=>, ErrorID=>, ErrorIDEx=>);

Function Block and Function Information

Item	Description
Library file name	OmronLib_AI_PM_Cylinder_V1_0.slr
Namespace	OmronLib\AI_PM_Cylinder_V1_0
FB/FUN number	00215
Publish/Do not publish source code	Not Published

Input Variables

Variables	Meaning	Data type	Description	Valid range	Unit	De-fault
Enable	Execute	BOOL	TRUE: Execute FALSE: Do not execute	TRUE, FALSE	---	FALSE
Pull	Pull Command Flag	BOOL	TRUE: Pull command ON FALSE: Pull command OFF	TRUE, FALSE	---	FALSE
Push	Push Command Flag	BOOL	TRUE: Push command ON FALSE: Push command OFF	TRUE, FALSE	---	FALSE
FullyRetracted-Pos	Fully Retracted Position	BOOL	TRUE: Fully retracted po-sition is reached FALSE: Fully retracted po-sition is not reached	TRUE, FALSE	---	FALSE

Variables	Meaning	Data type	Description	Valid range	Unit	Default
FullyExtended-Pos	Fully Extended Position	BOOL	TRUE: Fully extended position is reached FALSE: Fully extended position is not reached	TRUE, FALSE	---	FALSE
MonitorMode	Monitor Mode	OmronLib \AI_PM_Cylinder_V1_0\sMonitorMode	Inputs the monitor mode. *MonitorMode* is read when *Enable* changes to TRUE.	---	---	---
Timeout	Measurement Timeout Time	UINT	Inputs the measurement timeout time. When the set value is 0, the measurement timeout monitoring will not be executed.	Depends on data type	0.1 s	UINT #100 (10 s)

Output Variables

Variables	Meaning	Data type	Description	Valid range	Unit	Default
Enabled	Executing	BOOL	It is TRUE while executing.	TRUE, FALSE	---	---
MonitorStatus	Monitoring	BOOL	It is TRUE while monitoring the operation time of the cylinder, and FALSE while monitoring is suspended.	TRUE, FALSE	---	---
Error	Error	BOOL	TRUE: Error end FALSE: Normal end, execution in progress, or execution condition not met	TRUE, FALSE	---	---
ErrorID	Error Code	WORD	This is the error ID for an error end. The value is 16#0 for a normal end.	*1	---	---
ErrorIDEx	Expansion Error Code	DWORD	This is the error ID for an Expansion Error. The value is 16#0 for a normal end.	*1	---	---

*1. Refer to *Troubleshooting* on page 4 - 10 for details.

Structure

OmronLib\AI_PM_Cylinder_V1_0\sMonitorMode

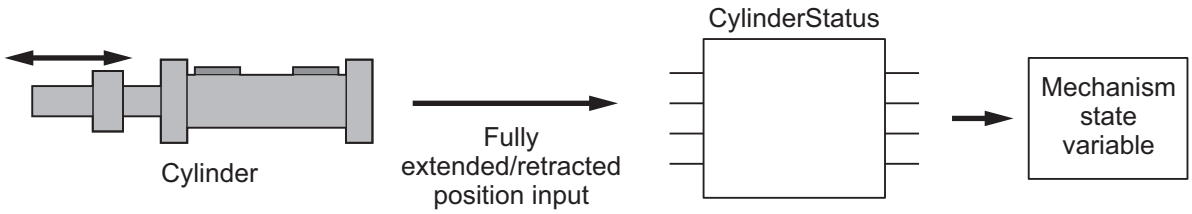
Member	Member name	Data type	Valid range	Description
CylinderType	Cylinder Type	UINT	0, 1	Inputs the cylinder type. CylinderType = 0: Two signals, the *Push Command Flag* and *Pull Command Flag*, are used for cylinder control (double mode). CylinderType = 1: Only the *Push Command Flag* is used for cylinder control (single mode).

Member	Member name	Data type	Valid range	Description
TargetType	Cylinder Operation Time to be Monitored	UINT	0, 1, 2	Inputs the cylinder operation time to be monitored. TargetType = 0: push operation time, pull operation time TargetType = 1: push operation time only TargetType = 2: pull operation time only
StartTimer	Monitoring Start Standby Time	UINT	Depends on data type	Inputs the time from the start of the first measurement until the start of monitoring. *1 When the set value is 0, monitoring is performed from the start of the first measurement. Unit: s
StopMonitorTimer	Stop Monitoring Time	UINT	Depends on data type	Inputs the cylinder operation stop monitoring time. When the set value is 0, the time during which the cylinder is stopped will not be monitored. Unit: s

*1. This is not the time since *Enable* changed to TRUE. Refer to *Specification of Monitoring Start Standby Time and Monitoring Stop* on page 4 - 5 for details.

Function

CylinderStatus generates the mechanism state variables from the cylinder's push operation time and pull operation time.
The Feature Value/Machine Learning Function of the AI function module references the generated mechanism state variables to determine and detect errors (decreasing velocity) of the cylinder.
This FB operates while *Enable* (Execute) is TRUE.



Connection with Cylinder

The connections between the cylinder and the input variables of this FB are as shown below.

Cylinder	Input variable to be connected
Push command	Push (Push Command Flag)
Pull command *1	Pull (Pull Command Flag)
Fully extended position reed switch	FullyExtendedPos (Fully Extended Position)
Fully retracted position reed switch*1	FullyRetractedPos (Fully Retracted Position)

*1. Connection is not required when the cylinder is in single mode. Refer to *Cylinder Operation Time Measurement Specifications for Each Cylinder Type* on page 4 - 5 for details on single mode.

Cylinder Operation Time Measurement Specifications for Each Cylinder Type

Set the cylinder type to be monitored with the member *CylinderType* (Cylinder Type) of the structure *sMonitorMode*. There are two cylinder types: double mode and single mode. The double mode uses *Push* and *Pull* signals, and the single mode uses only *Push*.
This FB starts measurement at a push command regardless of the cylinder type.
There are two types of cylinder operation time: push operation time and pull operation time. The value of *CylinderType* (Cylinder Type), measurement start timing of the cylinder operation time, and measurement end timing for each cylinder type are as follows.

Cylinder type	CylinderType value	Push operation time		Pull operation time	
		Measurement start timing	Measurement end timing	Measurement start timing	Measurement end timing
Double mode	0	The following AND conditions are satisfied • Push changes to TRUE • FullyRetracted-Pos is TRUE	FullyExtendedPos changes to TRUE	The following AND conditions are satisfied • Pull changes to TRUE • FullyExtended-Pos is TRUE	FullyRetractedPos changes to TRUE
Single mode	1	The following AND conditions are satisfied • Push changes to TRUE • FullyRetracted-Pos is TRUE	FullyExtendedPos changes to TRUE	The following AND conditions are satisfied • Push changes to FALSE • FullyExtended-Pos is TRUE	FullyRetractedPos changes to FALSE

Specification of Monitoring Target

Set the cylinder operation time to be monitored with the member *TargetType* (Cylinder Operation Time to be Monitored) of the structure *sMonitorMode*. The values of *TargetType* and the cylinder operation times to be monitored are as follows.

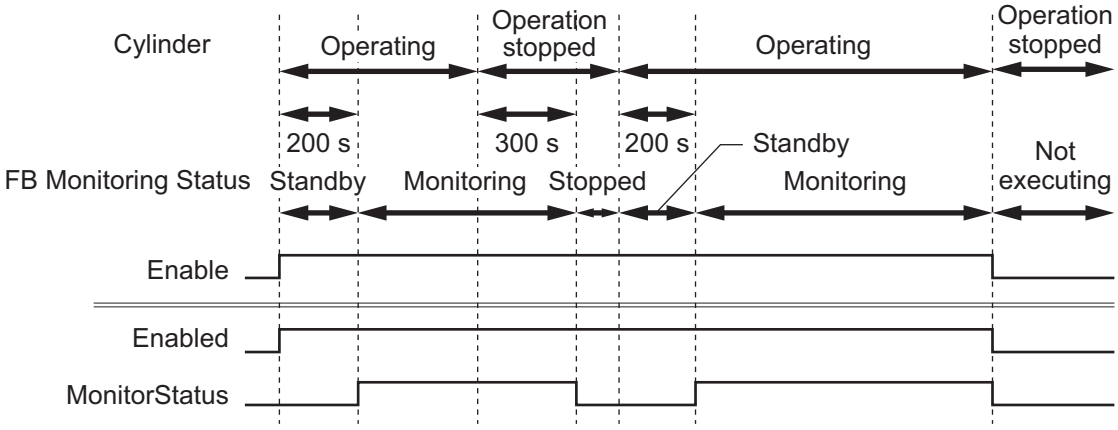
TargetType value	Cylinder operation time to be monitored
0	Push operation time and pull operation time
1	Push operation time only
2	Pull operation time only

Specification of Monitoring Start Standby Time and Monitoring Stop

Immediately after the cylinder operation starts, the push operation time and the pull operation time may be longer until the piston packing fits well in the cylinder. To exclude this period from the monitoring period, input the time from the start of operation of the cylinder to the start of monitoring to the member *StartTimer* (Monitoring Start Standby Time) of the structure *sMonitorMode*.
If the cylinder stops operating for a long time after the monitoring starts, the above phenomenon may occur again.

Therefore, after the cylinder is stopped for a long time, you must wait again for the time from the start of the operation of the cylinder to the start of monitoring (Monitoring Start Standby Time).
Input the time from the operation stop of the cylinder to the execution of monitoring standby with the monitoring start standby time in the member *StopMonitorTimer* (Stop Monitoring Time) of the structure *sMonitorMode*.

The two operation examples are shown in the following figure.
The following figure shows a timing chart of the operations.



● Example of switching from monitoring start standby to monitoring

If *StartTimer* is 200, the system waits to start the monitoring for 200 seconds after the start of operation while the push operation time is not stabilized, and the monitoring is started at the next Push operation performed after the 200 seconds. At this time, *MonitorStatus* changes to TRUE.

● Example of switching from monitoring to monitoring start standby

If *StopMonitorTimer* is 300, *MonitorStatus* changes to FALSE when the operation stops for 300 seconds or more. Monitoring start standby begins again when the next Push operation starts.

Measurement Timeout

If the cylinder operation time to be monitored exceeds the set value of *Timeout* (Measurement Timeout Time), timeout occurs and *Error* changes to TRUE.
The same value should be set for *Timeout* for both the push operation time and the pull operation time.
If you set this value to 0, timeout errors will not occur. The default value is 10 s.
Measurement timeout monitoring is always executed regardless of the monitoring state.

Setting Changes During Execution

Even if you change the value of *MonitorMode* or *Timeout* during execution, the change will not be applied.
The program executes with the values of the variables applied when *Enable* changes to TRUE.
For input variables other than the above, changed settings will be reflected.

Error Detection Conditions of Each Input Variable for Cylinder Operation States

The following tables show the cases where the FB operation results in an error due to the values of the input variables for the cylinder operation state.
When the conditions of both the measurement end timing and the measurement start timing of the next command are satisfied, there will be no error.
For details on the errors, refer to *Troubleshooting* on page 4 - 10.

● For single mode

Cylinder operating state	Error detection condition *1
Stopped at fully retracted position (*FullyRetractedPos* = TRUE)	• *Pull* = TRUE • *FullyExtendedPos* = TRUE
Performing push operation (*Push* = TRUE)	• *Push* = FALSE • *Pull* = TRUE
Stopped at fully extended position (*FullyExtendedPos* = TRUE)	• *Pull* = TRUE • *FullyRetractedPos* = TRUE
Performing pull operation (*Push* = FALSE)	• *Push* = TRUE • *Pull* = TRUE

*1. Bullet items are OR conditions.

● For double mode

Cylinder operating state	Error detection condition *1
Stopped at fully retracted position (*FullyRetractedPos* = TRUE)	• *Push* = TRUE and *Pull* = TRUE • *FullyExtendedPos* = TRUE
Performing push operation (*Push* = TRUE)	• *Pull* = TRUE
Stopped at fully extended position (*FullyExtendedPos* = TRUE)	• *Push* = TRUE and *Pull* = TRUE • *FullyRetractedPos* = TRUE
Performing pull operation (*Pull* = TRUE)	• *Push* = TRUE

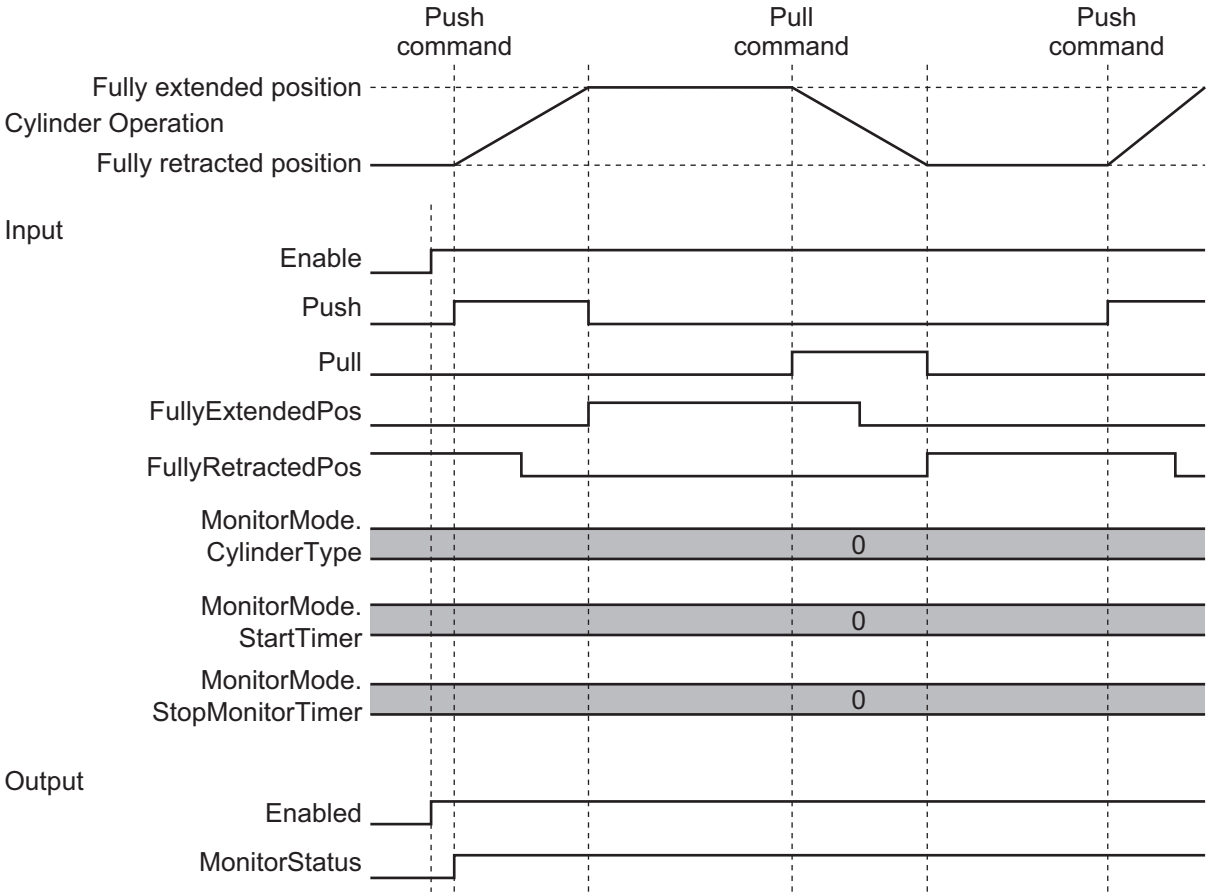
*1. Bullet items are OR conditions.

Timing Charts

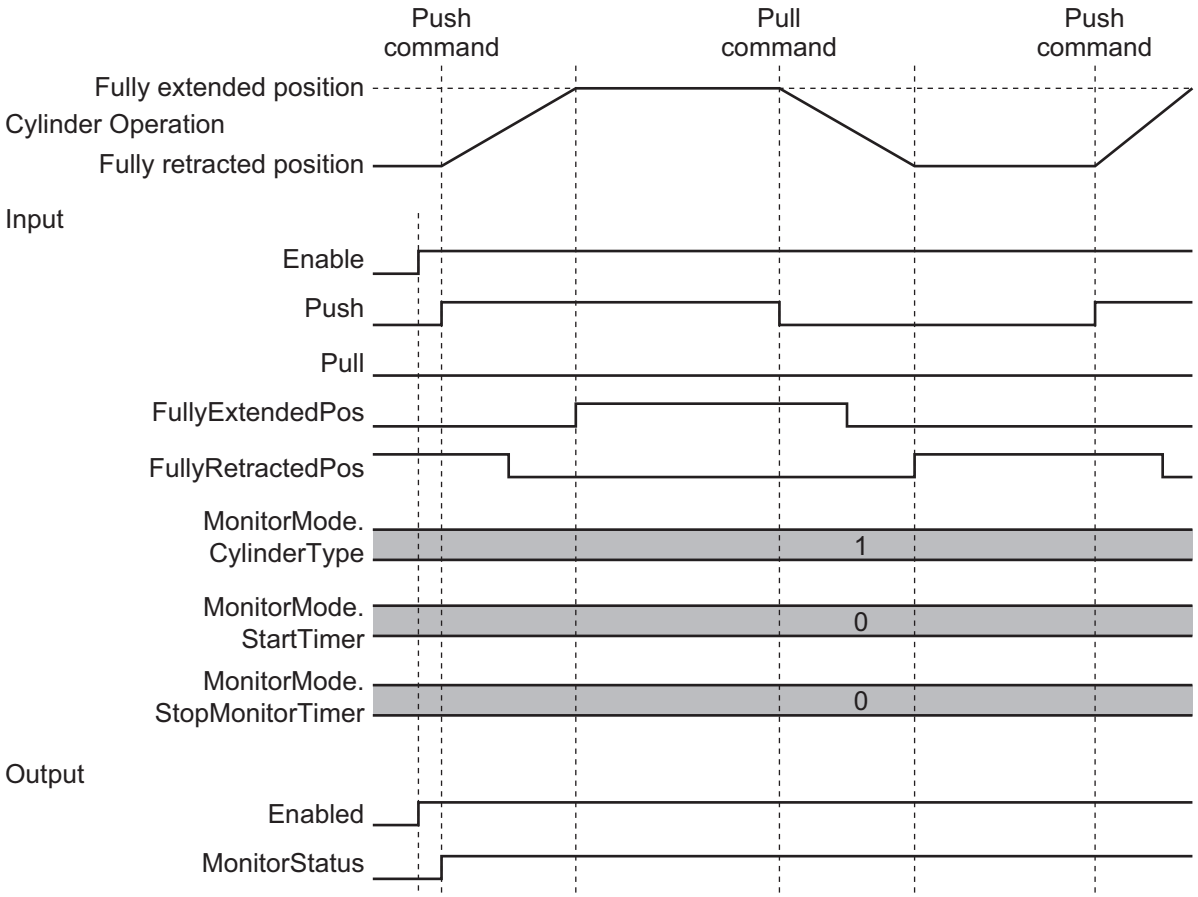
The timing chart for each state is shown below.

Timing Chart for Normal End

● **When *MonitorMode.CylinderType* = 0 (double mode)**

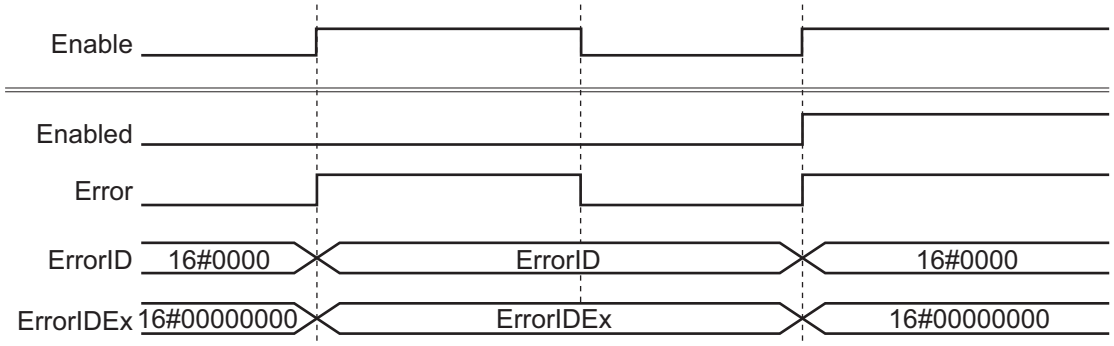


● **When *MonitorMode.CylinderType* = 1 (single mode)**

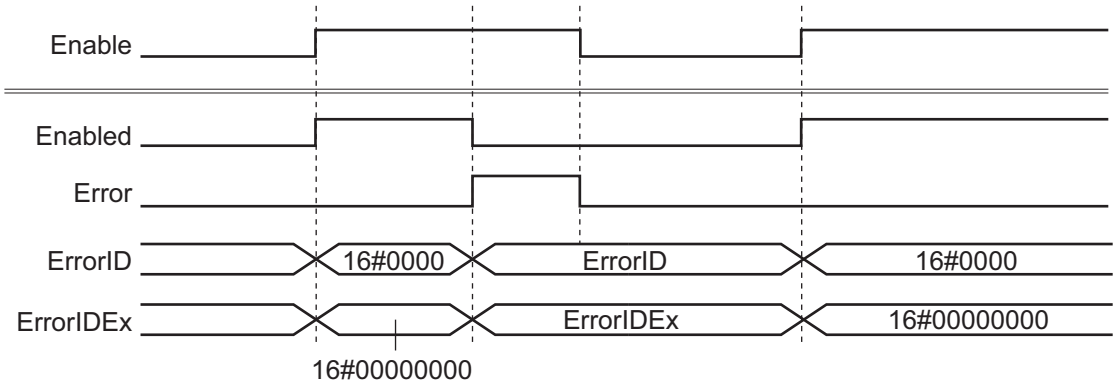


Timing Chart for Error End

● **If error occurs when *Enable* changes to TRUE**



- **If error occurs while *Enable* is TRUE**



Precautions for Correct Use

Execute the AI Predictive Maintenance Library on the program assigned to the primary periodic task. If you execute it on a program that is not assigned to the primary periodic task, you cannot generate a mechanism state variable that correctly reflects the state of the mechanism.

Troubleshooting

This section provides the statuses, descriptions, and corrective actions for the error codes and expansion error codes.

Error code	Expansion error code	Status	Description	Corrective action
16#0000	16#00000000	Normal End	---	---
16#3D15	16#00000001	CylinderType Input Value Out of Range	The value of *MonitorMode.CylinderType* is outside the valid range.	Correct the *MonitorMode.CylinderType* value so that it is within the valid range.
	16#00000002	TargetType Input Value Out of Range	The value of *MonitorMode.TargetType* is outside the valid range.	Correct the *MonitorMode.TargetType* value so that it is within the valid range.
	16#00000004	Push/Pull Simultaneous Input (double mode)	In double mode, the values of *Push* and *Pull* became TRUE at the same time.	Make sure that there are no errors in the input variable assignment and wiring.
	16#00000005	Pull Input Error (single mode)	In single mode, the value of *Pull* became TRUE.	
	16#00000006	FullyExtendedPos/FullyRetractedPos Simultaneous Input	The values of *FullyExtendedPos* and *FullyRetractedPos* became TRUE at the same time.	

Error code	Expansion error code	Status	Description	Corrective action
	16#00000007	Push Input Aborted	• Single mode While the *Push* value is TRUE, the *Push* value becomes FALSE before the *FullyExtendedPos* value becomes TRUE. • Double mode After the *Push* value changes to TRUE, the *Pull* value becomes TRUE before the *FullyExtendedPos* value becomes TRUE.	
	16#00000008	Pull Input Aborted	• Single mode While the *Push* value is FALSE, the *Push* value becomes TRUE before the *FullyRetractedPos* value becomes TRUE. • Double mode After the *Pull* value changes to TRUE, the *Push* value becomes TRUE before the *FullyRetractedPos* value becomes TRUE.	
	16#00000009	Timeout Error	The operation time of the cylinder to be monitored exceeded the *Timeout* set value.	

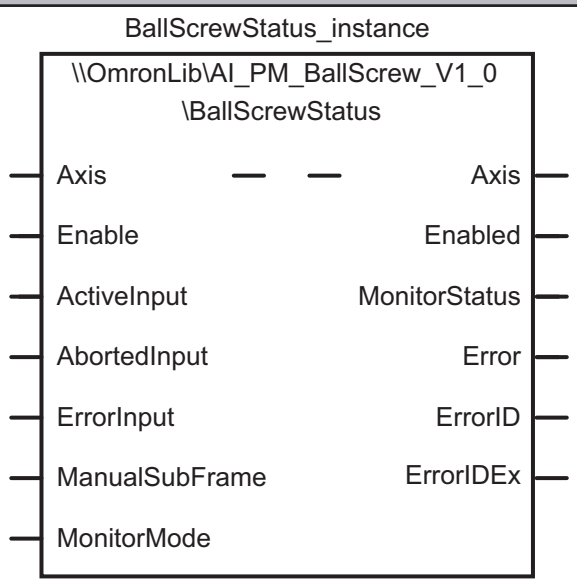
5

Individual Specifications of Function Block (Ball Screw)

This section describes the individual specifications (ball screw) of the AI Predictive Maintenance Library.

BallScrewStatus

BallScrewStatus generates mechanism state variables that reflect the state of the ball screw, and those variables are referenced by the Feature Value/Machine Learning Function.

FB name	Name	FB/ FUN	Graphic expression	ST expression
Ball-Screw-Status	Ball Screw Status Output	FB		BallScrewStatus_instance(Axis:=, Enable:=, ActiveInput:=, AbortedInput:=, ErrorInput:=, ManualSubFrame:=, MonitorMode=, Enabled=>, MonitorStatus=>, Error=>, ErrorID=>, ErrorIDEx=>);

Function Block and Function Information

Item	Description
Library file name	OmronLib_AI_PM_BallScrew_V1_0.slr
Namespace	OmronLib\AI_PM_BallScrew_V1_0
FB/FUN number	00216
Publish/Do not publish source code	Not Published

Input Variables

Variables	Meaning	Data type	Description	Valid range	Unit	Default
Enable	Execute	BOOL	TRUE: Execute FALSE: Do not execute	TRUE, FALSE	---	FALSE
ActiveInput	Active Input	BOOL	Inputs *Active*[*2] of the motion control instruction[*1] to be monitored.	TRUE, FALSE	---	FALSE
AbortedInput	Command Aborted Input	BOOL	Inputs *CommandAborted*[*2] of the motion control instruction[*1] to be monitored.	TRUE, FALSE	---	FALSE
ErrorInput	Error Input	BOOL	Inputs *Error*[*2] of the motion control instruction[*1] to be monitored.	TRUE, FALSE	---	FALSE

Variables	Meaning	Data type	Description	Valid range	Unit	Default
Manual-Sub-Frame	User-defined Subframe	BOOL	Inputs the user-defined subframe. It is valid when *MonitorMode.SubFrameMode* is TRUE. When *MonitorMode.SubFrameMode* is FALSE, it becomes invalid and the subframe variable that has been generated inside this FB is used.	TRUE, FALSE	---	FALSE
Monitor-Mode	Monitor Mode	OmronLib\AI_PM_BallScrew_V1_0\sMonitor-Mode	Inputs the monitor mode. *MonitorMode* is read when *Enable* changes to TRUE.	---	---	---

*1. Refer to *1-7 Mechanisms of AI FB Linked to Motion Control* on page 1 - 11 for details.

*2. Refer to the *NJ/NX-series Motion Control Instructions Reference Manual (Cat. No. W508)* or the *NY-series Motion Control Instructions Reference Manual (Cat. No. W561)* for details.

Output Variables

Variables	Meaning	Data type	Description	Valid range	Unit	Default
Enabled	Executing	BOOL	Set to TRUE during execution.	TRUE, FALSE	---	---
MonitorStatus	Monitoring	BOOL	Set to TRUE while monitoring the movement of the ball screw, and set to FALSE while monitoring is paused. When the value of Moving Average Count after *Enable* is TRUE is less than *MonitorMode.MVAveCount,* it changes to FALSE, and if the value is equal to or higher than *MonitorMode.MVAveCount*, it changes to TRUE.	TRUE, FALSE	---	---
Error	Error	BOOL	TRUE: Error end FALSE: Normal end, execution in progress, or execution condition not met	TRUE, FALSE	---	---
ErrorID	Error Code	WORD	This is the error ID for an error end. The value is 16#0 for a normal end.	*1	---	---
ErrorIDEx	Expansion Error Code	DWORD	This is the error ID for an Expansion Error. The value is 16#0 for a normal end.	*1	---	---

*1. Refer to *Troubleshooting* on page 5 - 10 for details.

Input-Output Variables

Variables	Meaning	Data type	Description	Valid range	Unit	Default
Axis	Axis	_sAXIS_REF*1	Specifies *Axis* of the motion control instruction*2 to be monitored.	---	---	---

*1. Refer to *NJ/NX-series Motion Control Instructions Reference Manual (Cat. No. W508)* or *NY-series Motion Control Instructions Reference Manual (Cat. No. W561)* for details.

*2. Refer to *1-7 Mechanisms of AI FB Linked to Motion Control* on page 1 - 11 for details.

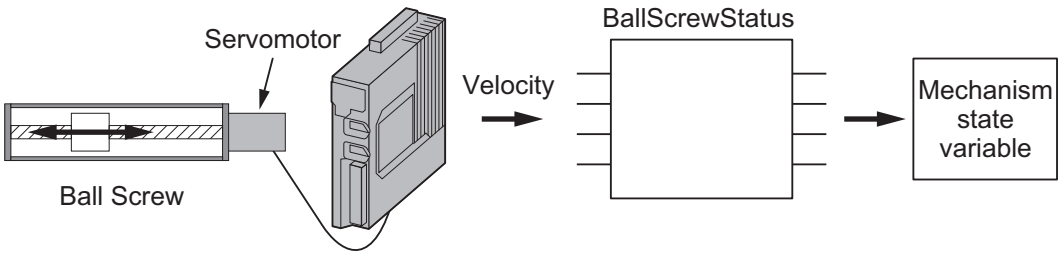
Structure

OmronLib\AI_PM_BallScrew_V1_0\sMonitorMode

Member	Member name	Data type	Valid range	Description
Mode	Mode Type	UINT	0, 1	Inputs the Mode type. *Mode* = 0: Where one FB handles one movement. *Mode* = 1: Where one FB handles multiple movements.
ActiveMax-Num	Active Count	UINT	1 to 65535	It is valid when *Mode* is 1. Inputs the number of times the change from TRUE to FALSE of *ActiveInput* is input within one frame.
TargetActi-veNo	Monitoring Target Active Number	UINT	1 to 65535	It is valid when *Mode* is 1. Inputs which *Active* is to be monitored from the start of the frame.
MVAve-Count	Moving Aver-age Count	UINT	0 to 100	Inputs the count for the moving average. When it is 0 and 1, the moving average is not performed.
SubFrame-Mode	Subframe Se-lection Type	BOOL	TRUE, FALSE	Inputs the subframe selection type. *SubFrameMode* = FALSE: Uses the subframe generated in the FB. *SubFrameMode* = TRUE: Uses the user-defined subframe varia-ble *ManualSubFrame*.

Function

BallScrewStatus generates the mechanism state variables from the ball screw velocity.
The Feature Value/Machine Learning Function of the AI function module references the generated mechanism state variables to determine and detect errors of the ball screw.
This FB operates while *Enable* (Execute) is TRUE.



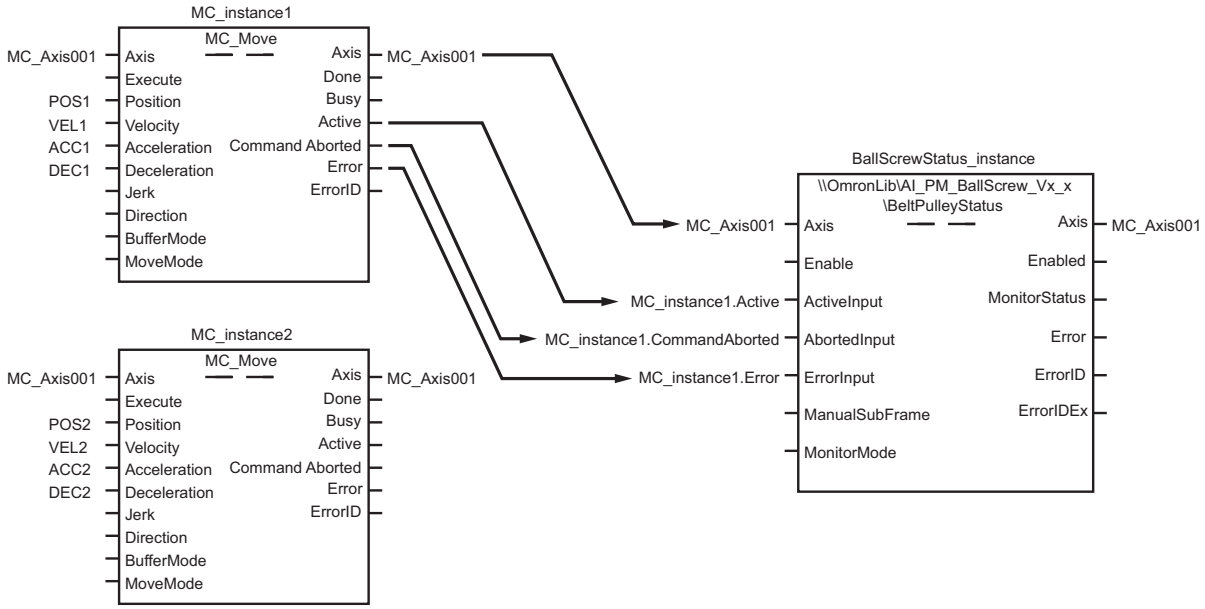
Connection with Ball Screw

Connect the *axis variable* controlling the ball screw and the output variables *Active*, *CommandAborted*, and *Error* of the target motion control instruction to the input variables of this FB. If there are multiple motion control instructions controlling the ball screw, input the output variables of one representative motion control instruction to this FB.

Additional Information

When selecting the representative motion control instruction, be sure to select one with a long operation stroke, large operation speed, large acceleration, large deceleration, and large load on the ball screw, such as the workpiece weight. The more that these conditions are satisfied, the more accurately errors can be detected.

Example of connection when there are two motion control instructions controlling the ball screw



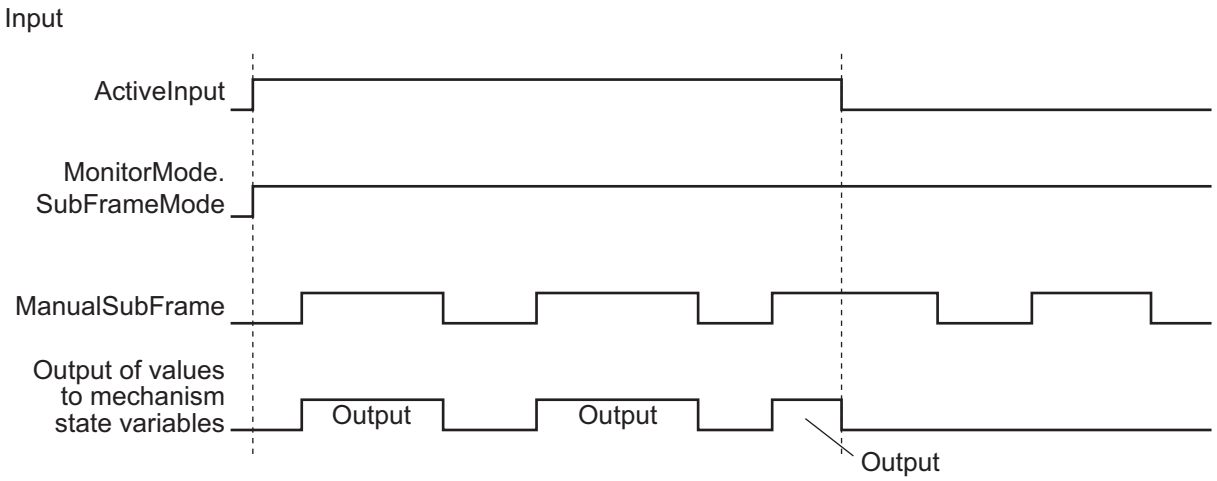
Specification of Monitoring Target

Set the movement of the ball screw to be monitored with the *Mode*, *ActiveMaxNum*, and *TargetActiveNo* members of the *sMonitorMode* structure.

- When monitoring a single movement with one FB, set *Mode* to 0, and the monitoring target is the period in which *ActiveInput* is ON.
- When monitoring multiple movements with one FB, set *Mode* to 1, and the monitoring target is specified by *ActiveMaxNum* and *TargetActiveNo*.
  In *ActiveMaxNum*, enter the number of times the change from TRUE to FALSE of *ActiveInput* is input within one frame.
  In *TargetActiveNo*, enter which change from TRUE to FALSE of *ActiveInput* is to be monitored from the start of the frame.
  For example, if there are three movements in the same FB within one frame and the first movement is targeted for monitoring, enter 3 for *ActiveMaxNum* and 1 for *TargetActiveNo*.

Specification of Subframe Selection Type

Set the subframe selection type with the member *SubFrameMode* of the structure *sMonitorMode*.
When *SubFrameMode* is FALSE, the subframe generated inside the FB is used, and while *ActiveInput* is TRUE, the value is output to the mechanism state variable.
When *SubFrameMode* is TRUE, enter the user-defined subframe. While *ManualSubFrame* is TRUE and *ActiveInput* is TRUE, the value is output to the mechanism state variable.
The following is a timing chart when the subframe selection type is set to a user-defined subframe.



Specification of Moving Average Count

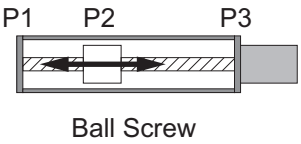
The moving average is calculated with the backward moving average with respect to the mechanism state variables for the frames specified by the Moving Average Count.
You can set the count for the moving average in the calculation of the mechanism state variable with the member *MVAveCount* of the structure *sMonitorMode*. The moving average of the maximum value of the velocity absolute for each frame is calculated for the number of data set in *MCAveCount* value.
If there is a large amount of variation in the velocity, you can use the moving average for noise elimination applications.
The moving average is not calculated for a *MVAveCount* of 0 and a *MVAveCount* of 1.
When the frame number from when *Enable* is TRUE exceeds the set value, *MonitorStatus* changes to TRUE and monitoring starts.

Setting Changes During Execution

Changes to the *MonitorMode* settings made during execution will not be reflected.
For input variables other than *MonitorMode*, changed settings will be reflected.

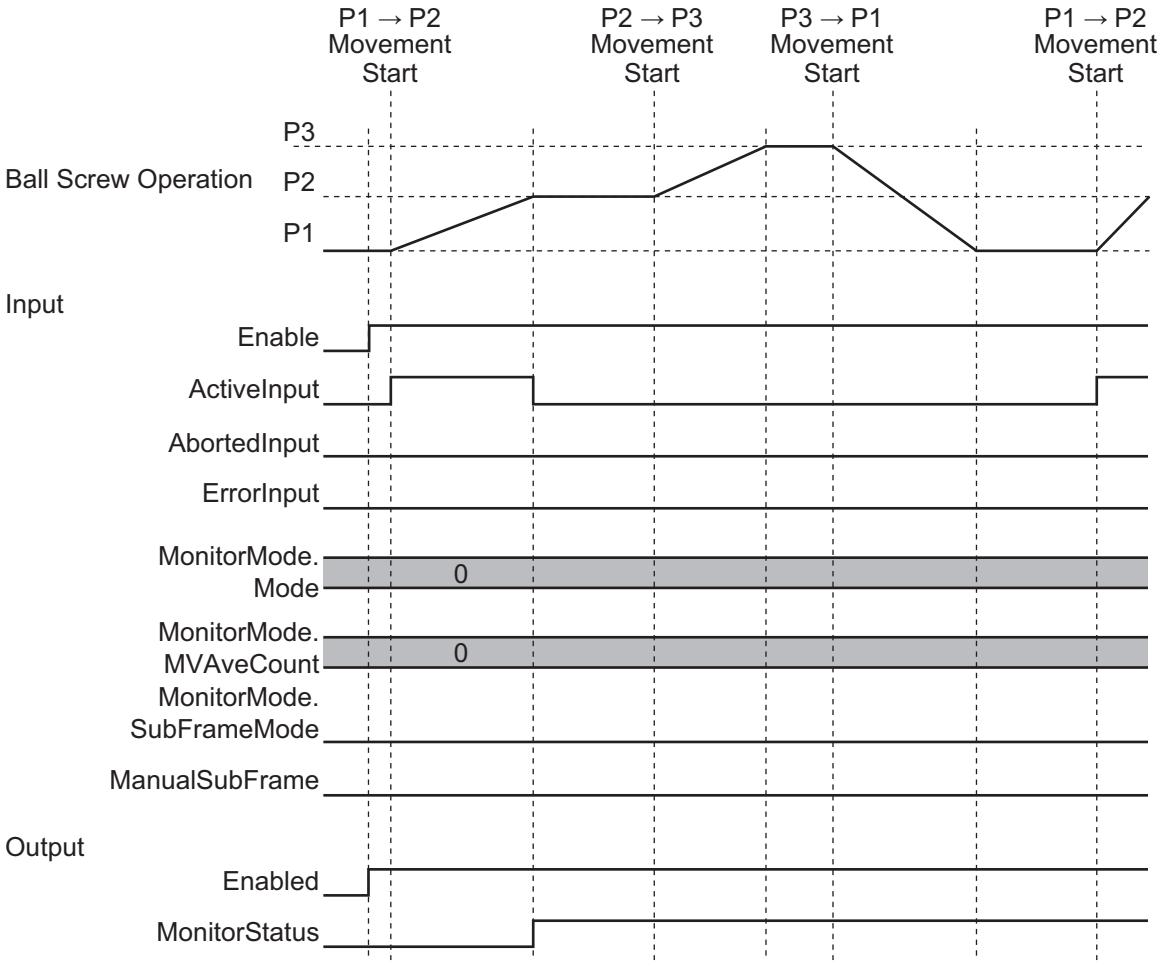
Timing Charts

The timing charts for the normal end and the error end of the execution results of FB are shown below.

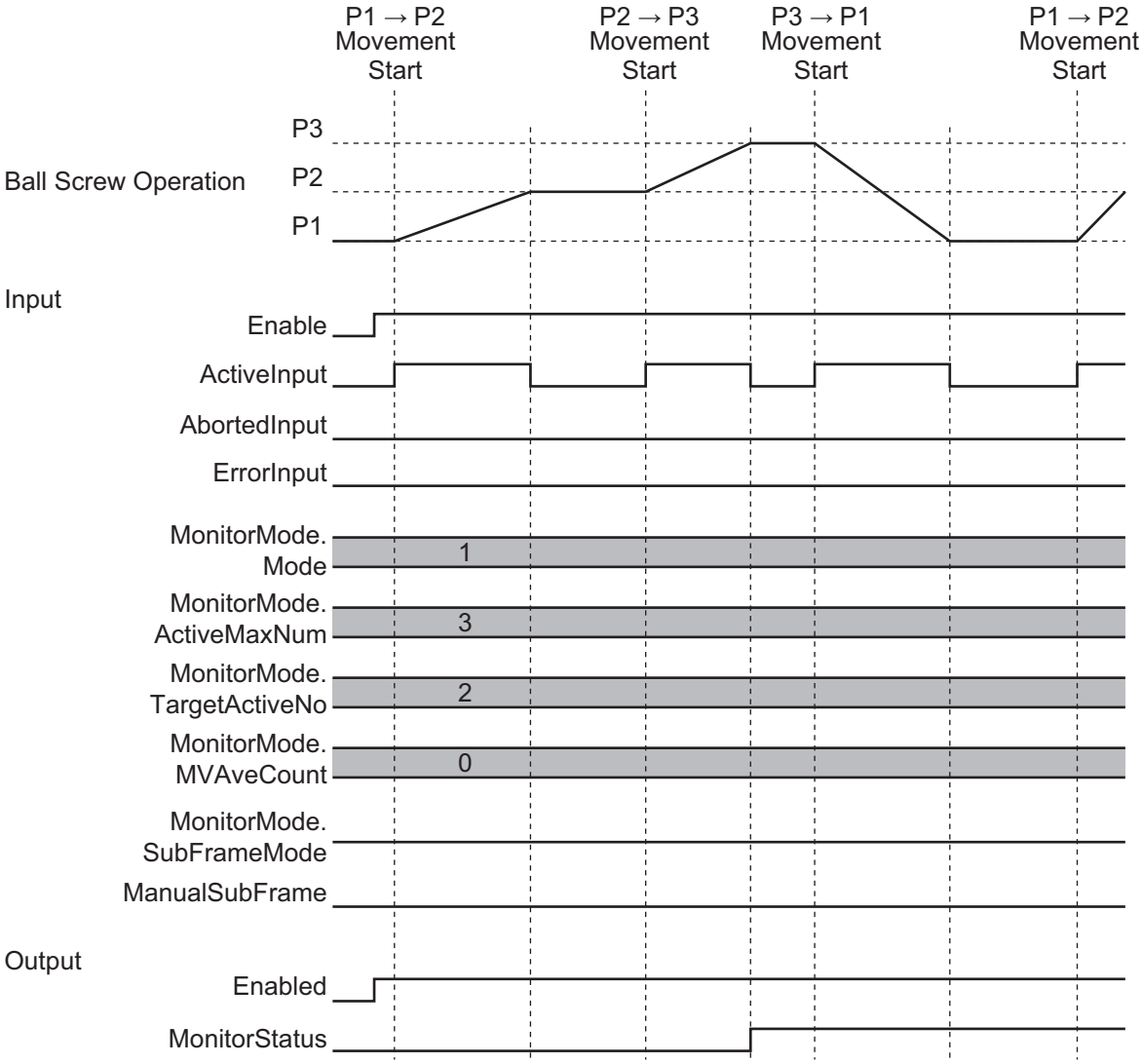


Timing Chart for Normal End

● **When *MonitorMode.Mode* = 0 and the movement from P1 to P2 is monitored**

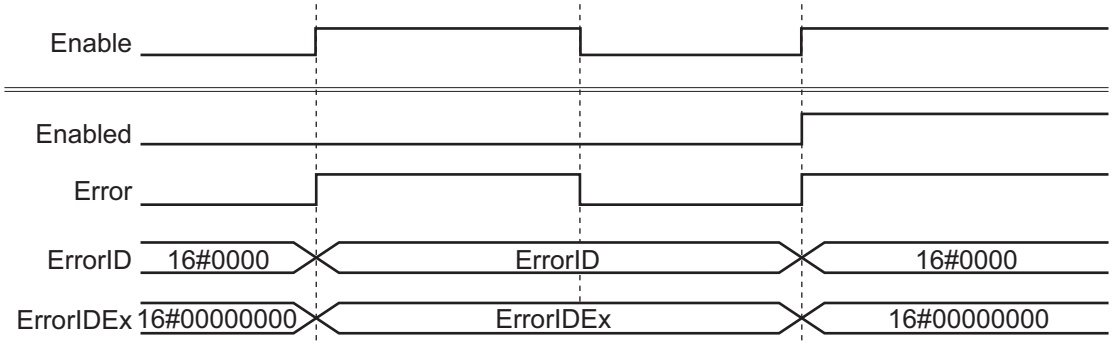


- **When *MonitorMode.Mode* = 1, *MonitorMode.ActiveMaxNum* = 3, and *MonitorMode.TargetActiveNo* = 2**

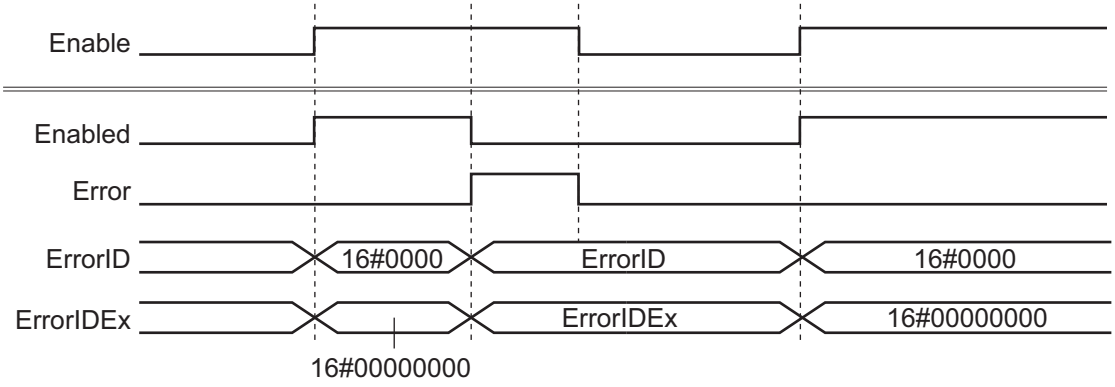


Timing Chart for Error End

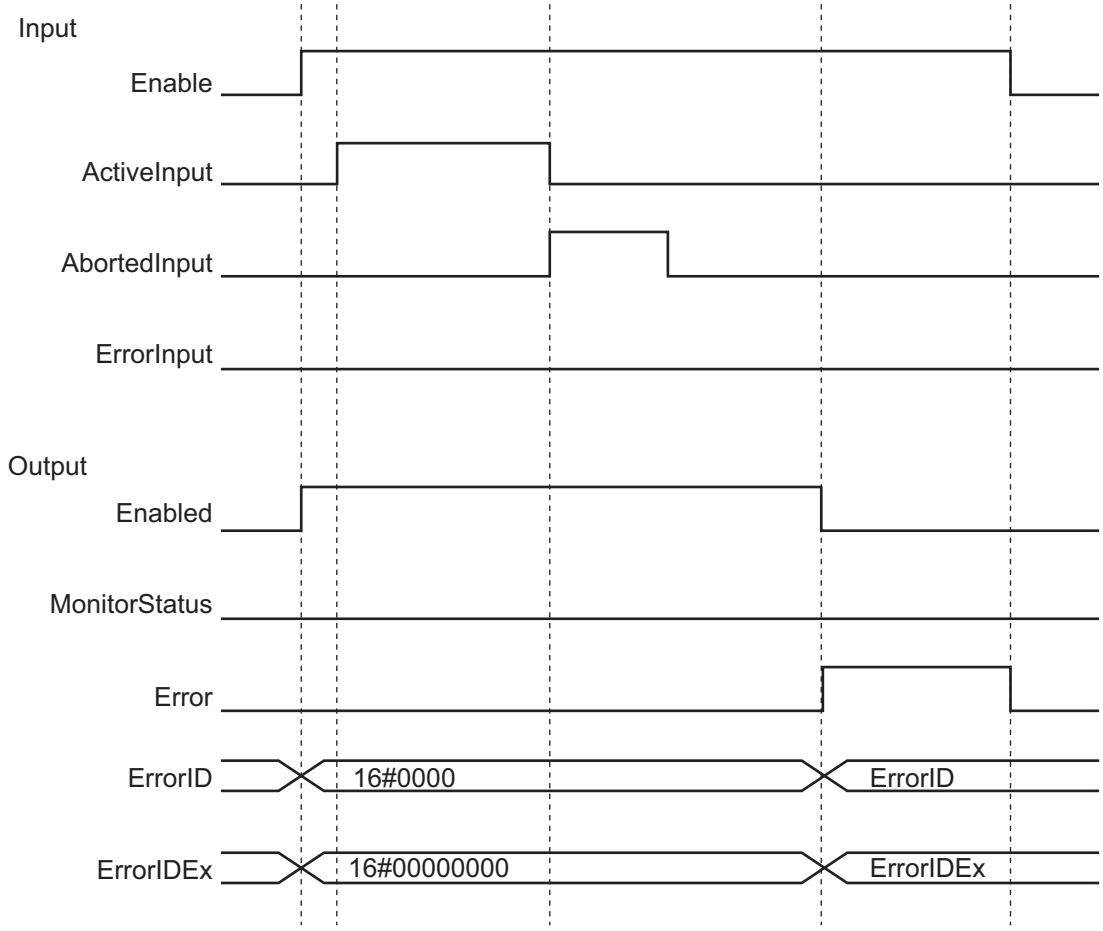
- **If error occurs when *Enable* changes to TRUE**



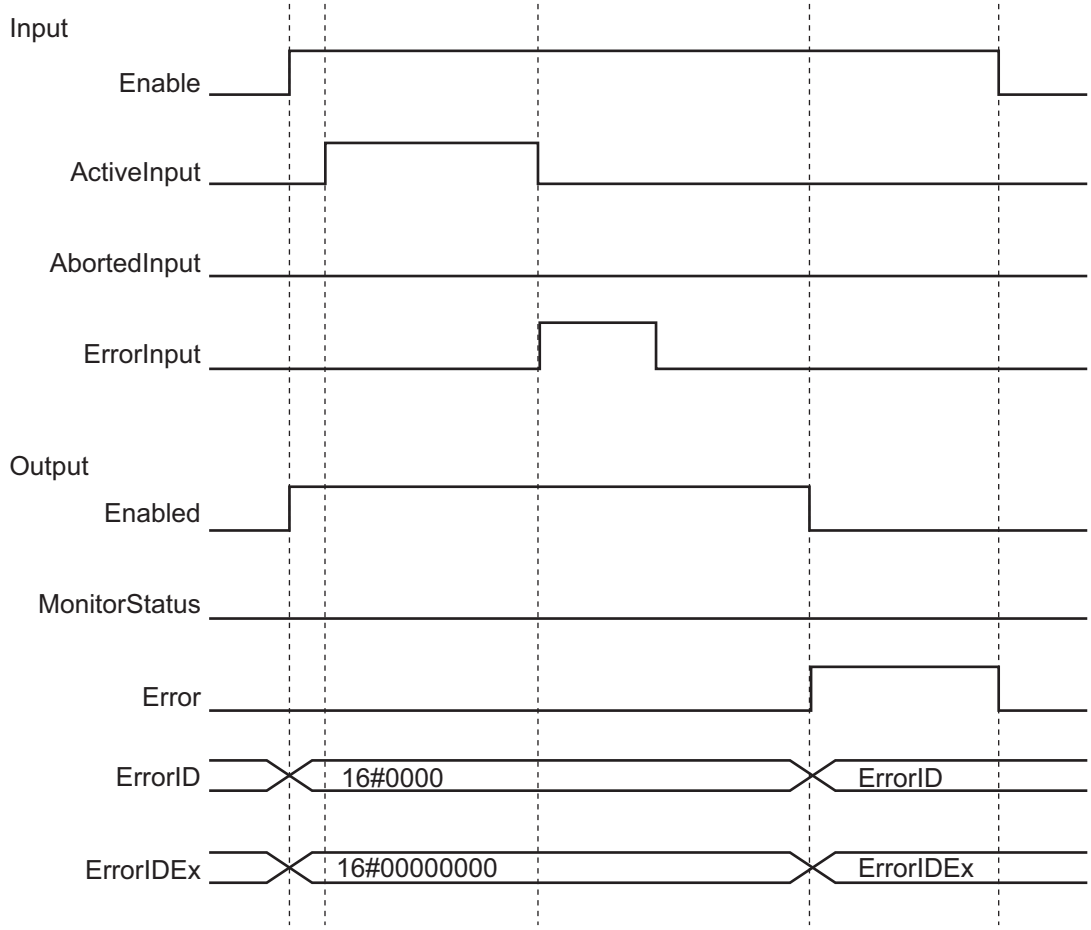
● **If error occurs while *Enable* is TRUE**



● **When *AbortedInput* changes to TRUE**



● **When *ErrorInput* changes to TRUE**



Precautions for Correct Use

Execute the AI Predictive Maintenance Library on the program assigned to the primary periodic task. If you execute it on a program that is not assigned to the primary periodic task, you cannot generate a mechanism state variable that correctly reflects the state of the mechanism.

Troubleshooting

This section provides the statuses, descriptions, and corrective actions for the error codes and expansion error codes.

Error code	Expansion error code	Status	Description	Corrective action
16#0000	16#00000000	Normal End	---	---
16#3D16	16#00000001	Mode Type Input Value Out of Range	The value of *MonitorMode.Mode* is outside the valid range.	Correct the *MonitorMode.Mode* value so that it is within the valid range.
	16#00000002	Active Count Input Value Out of Range	When *MonitorMode.Mode* is 1, the value of *MonitorMode.ActiveMaxNum* is 0.	Correct the *MonitorMode.ActiveMaxNum* value so that it is within the valid range.

Error code	Expansion error code	Status	Description	Corrective action
	16#00000003	Active Number Input Value Out of Range	When *MonitorMode.Mode* is 1, the value of *MonitorMode.TargetActiveNo* is 0.	Correct the *MonitorMode.TargetActiveNo* value so that it is within the valid range.
	16#00000004	Active Input Mismatch	When *MonitorMode.Mode* is 1, the value of *MonitorMode.ActiveMaxNum* is less than the value of *MonitorMode.TargetActiveNo.*	Set the value of *MonitorMode.ActiveMaxNum* so that it is greater than or equal to *MonitorMode.TargetActiveNo*.
	16#00000005	MVAveCount Input Value Out of Range	The value of *MonitorMode.MVAveCount* is outside the valid range.	Correct the *MonitorMode.MVAveCount* value so that it is within the valid range.
	16#00000006	Axis Command Error	The value of *ErrorInput* changed to TRUE.	Check the execution result of the motion control instruction. Refer to *NJ/NX-series Motion Control Instructions Reference Manual (Cat. No. W508)* or *NY-series Motion Control Instructions Reference Manual (Cat. No. W561)* for details.
	16#00000007	Axis Command Aborted	The value of *AbortedInput* changed to TRUE.	
	16#00000008	Mechanism State Variable Calculation Error	The operation result of the mechanism state variable has become a non-number or ± ∞.	Check the input value of *MonitorMode*.

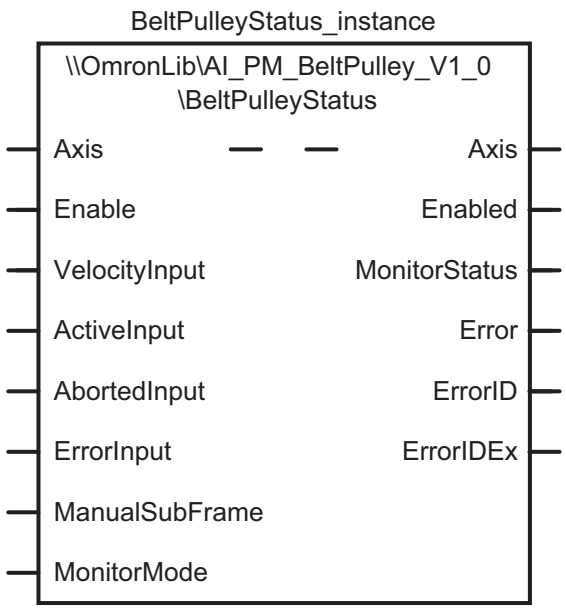
6

Individual Specifications of Function Block (Belt Pulley)

This section describes the individual specifications (belt pulley) of the AI Predictive Maintenance Library.

BeltPulleyStatus

BeltPulleyStatus generates mechanism state variables that reflect the state of the belt pulley, and those variables are referenced by the Feature Value/Machine Learning Function.

FB name	Name	FB/ FUN	Graphic expression	ST expression
BeltPulleyStatus	Belt Pulley Status Output	FB		BeltPulleyStatus_instance(Axis:=, Enable:=, VelocityInput:=, ActiveInput:=, AbortedInput:=, ErrorInput:=, ManualSubFrame:=, MonitorMode=, Enabled=>, MonitorStatus=>, Error=>, ErrorID=>, ErrorIDEx=>);

Function Block and Function Information

Item	Description
Library file name	OmronLib_AI_PM_BeltPulley_V1_0.slr
Namespace	OmronLib\AI_PM_BeltPulley_V1_0
FB/FUN number	00217
Publish/Do not publish source code	Not Published

Input Variables

Variables	Meaning	Data type	Description	Valid range	Unit	Default
Enable	Execute	BOOL	TRUE: Execute FALSE: Do not execute	TRUE, FALSE	---	FALSE
VelocityInput	Target Velocity Input	LREAL	Inputs the target velocity *Velocity*[*2] of the motion control instruction[*1] to be monitored.	Positive number	Command unit / s	0
ActiveInput	Active Input	BOOL	Inputs *Active*[*2] of the motion control instruction[*1] to be monitored.	TRUE, FALSE	---	FALSE
AbortedInput	Command Aborted Input	BOOL	Inputs *CommandAborted*[*2] of the motion control instruction[*1] to be monitored.	TRUE, FALSE	---	FALSE

Variables	Meaning	Data type	Description	Valid range	Unit	Default
ErrorInput	Error Input	BOOL	Inputs *Error*[*2] of the motion control instruction[*1] to be monitored.	TRUE, FALSE	---	FALSE
ManualSubFrame	User-defined Subframe	BOOL	Inputs the user-defined subframe. It is valid when *MonitorMode.SubFrameMode* is TRUE. When *MonitorMode.SubFrameMode* is FALSE, it becomes invalid and the subframe variable that has been generated inside this FB is used.	TRUE, FALSE	---	FALSE
MonitorMode	Monitor Mode	OmronLib\AI_PM_BeltPulley_V1_0\sMonitorMode	Inputs the monitor mode. *MonitorMode* is read when *Enable* changes to TRUE.	---	---	---

*1. Refer to *1-7 Mechanisms of AI FB Linked to Motion Control* on page 1 - 11 for details.
*2. Refer to the *NJ/NX-series Motion Control Instructions Reference Manual (Cat. No. W508)* or the *NY-series Motion Control Instructions Reference Manual (Cat. No. W561)* for details.

Output Variables

Variables	Meaning	Data type	Description	Valid range	Unit	Default
Enabled	Executing	BOOL	Set to TRUE during execution.	TRUE, FALSE	---	---
MonitorStatus	Monitoring	BOOL	Set to TRUE while monitoring the operation time of the belt pulley, and FALSE while monitoring is paused.	TRUE, FALSE	---	---
Error	Error	BOOL	TRUE: Error end FALSE: Normal end, execution in progress, or execution condition not met	TRUE, FALSE	---	---
ErrorID	Error Code	WORD	This is the error ID for an error end. The value is 16#0 for a normal end.	*1	---	---
ErrorIDEx	Expansion Error Code	DWORD	This is the error ID for an Expansion Error. The value is 16#0 for a normal end.	*1	---	---

*1. Refer to *Troubleshooting* on page 6 - 12 for details.

Input-Output Variables

Variables	Meaning	Data type	Description	Valid range	Unit	Default
Axis	Axis	_sAXIS_REF[*1]	Specifies *Axis* of the motion control instruction[*2] to be monitored.	---	---	---

*1. Refer to the *NJ/NX-series Motion Control Instructions Reference Manual (Cat. No. W508)* or the *NY-series Motion Control Instructions Reference Manual (Cat. No. W561)* for details.

*2. Refer to *1-7 Mechanisms of AI FB Linked to Motion Control* on page 1 - 11 for details.

Structure

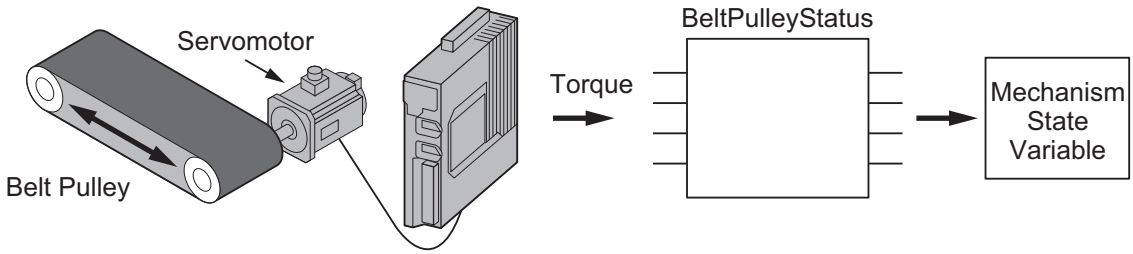
OmronLib\AI_PM_BeltPulley_V1_0\sMonitorMode

Member	Member name	Data type	Valid range	Description
Mode	Mode Type	UINT	0, 1	Inputs the Mode type. *Mode* = 0: Where one FB handles one movement. *Mode* = 1: Where one FB handles multiple movements.
ActiveMaxNum	Active Count	UINT	1 to 65535	It is valid when *Mode* is 1. Inputs the number of times the change from TRUE to FALSE of *ActiveInput* is input within one frame.
TargetActiveNo	Monitoring Target Active Number	UINT	1 to 65535	It is valid when *Mode* is 1. Inputs which *Active* is to be monitored from the start of the frame.
VelocityRatio	Velocity Ratio	UINT	0 to 100	Parameter for calculating the subframe. Specifies the percentage of the target velocity *VelocityInput*; the subframe period is defined as the time during which this value is exceeded by the velocity command value. The Unit is %. When it is 0, it is treated as 95%.
SubFrameMode	Subframe Selection Type	BOOL	TRUE, FALSE	Inputs the subframe selection type. *SubFrameMode* = FALSE: Uses the subframe generated in the FB. *SubFrameMode* = TRUE: Uses the user-defined subframe variable (ManualSubFrame).
StartTimer	Monitoring Start Standby Time	UINT	Depends on data type	Inputs the time from the start of the first measurement until the start of monitoring. *1 When the set value is 0, monitoring is performed from the start of the first measurement. Unit: s
StopMonitorTimer	Stop Monitoring Time	UINT	Depends on data type	Inputs the stop monitoring time. When the set value is 0, the time during which the belt pulley is stopped will not be monitored. Unit: s

*1. This is not the time since *Enable* changed to TRUE. Refer to *Specification of Monitoring Start Standby Time and Monitoring Stop* on page 6 - 6 for details.

Function

BeltPulleyStatus generates the mechanism state variables from the belt pulley torque.
The Feature Value/Machine Learning Function of the AI function module references the generated mechanism state variables to determine and detect errors of the belt pulley.
This FB operates while *Enable* (Execute) is TRUE.



Connection with Belt Pulley

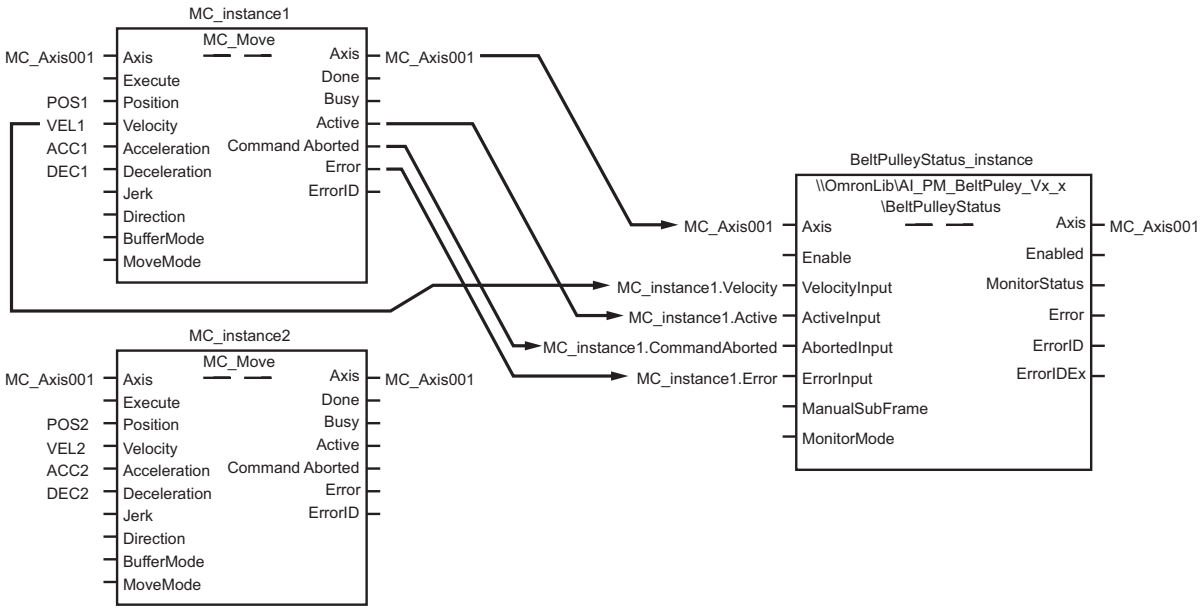
Connect the *axis variable* controlling the belt pulley and the input variable *Velocity* and output variables *Active*, *Command Aborted*, and *Error* of the target motion control instruction to the input variables of this FB.

If there are multiple motion control instructions controlling the belt pulley, input the output variables of one representative motion control instruction to this FB.

Additional Information

When selecting the representative motion control instruction, be sure to select one with a long operation stroke, large operation speed, large acceleration, large deceleration, and large load on the belt pulley. The more that these conditions are satisfied, the more accurately errors can be detected.

Example of connection when there are two motion control instructions controlling the belt pulley



Specification of Monitoring Target

Set the movement of the belt pulley to be monitored with the members *Mode*, *ActiveMaxNum*, and *TargetActiveNo* of the structure *sMonitorMode*.

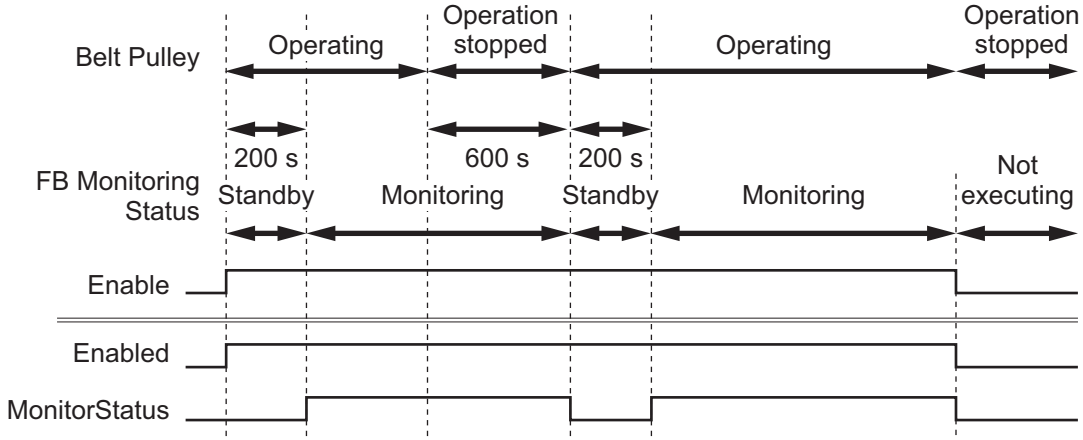
- When monitoring a single movement with one FB, set *Mode* to 0, and the monitoring target is the period in which *ActiveInput* is ON.

- When monitoring multiple movements with one FB, set *Mode* to 1, and the monitoring target is specified by *ActiveMaxNum* and *TargetActiveNo*.
  In *ActiveMaxNum*, enter the number of times the change from TRUE to FALSE of *ActiveInput* is input within one frame.
  In *TargetActiveNo*, enter which change from TRUE to FALSE of *ActiveInput* is to be monitored from the start of the frame.
  For example, if there are three movements in the same FB within one frame and the first movement is targeted for monitoring, enter 3 for *ActiveMaxNum* and 1 for *TargetActiveNo*.

Specification of Monitoring Start Standby Time and Monitoring Stop

The torque value may be unstable immediately after starting the belt pulley operation. To exclude this period from the monitoring period, input the time from the start of operation of the belt pulley to the start of monitoring to the member *StartTimer* (Monitoring Start Standby Time) of the structure *sMonitorMode*.
If the belt pulley stops operating for a long time after the monitoring starts, the above phenomenon may occur again.
Therefore, after the belt pulley is stopped for a long time, you must wait again for the time from the start of the operation of the belt pulley to the start of monitoring (Monitoring Start Standby Time).
Input the time from the operation stop of the belt pulley to the execution of monitoring standby with the monitoring start standby time in the member *StopMonitorTimer* (Stop Monitoring Time) of the structure *sMonitorMode*.

The two operation examples are shown in the following figure.
The following figure shows a timing chart of the operations.



● Example of switching from monitoring start standby to monitoring

When *StartTimer* is 200, the system waits to start the monitoring for 200 seconds after the start of operation while the torque value is not stabilized, and the monitoring is started when *ActiveInput* changes to TRUE for the next monitoring after the 200 seconds. At this time, *MonitorStatus* changes to TRUE.

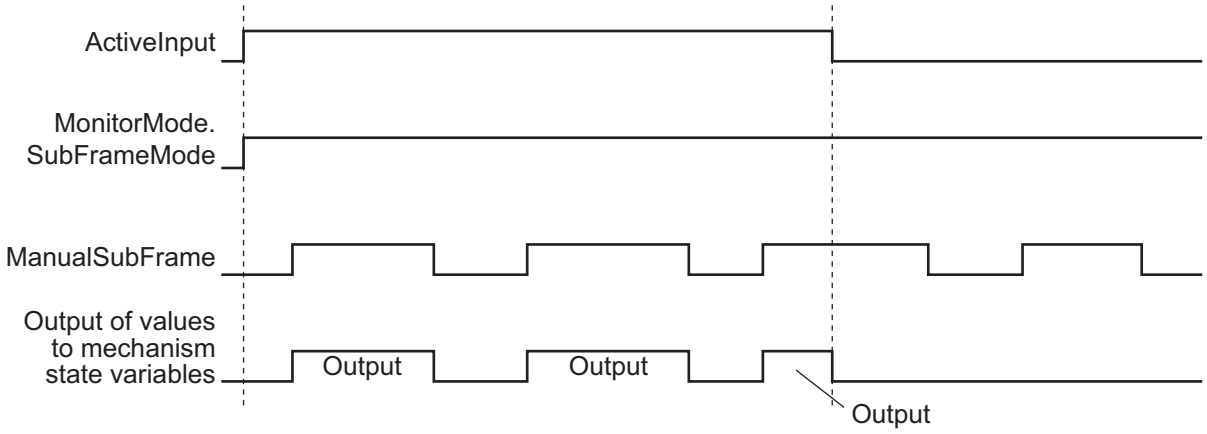
● Example of switching from monitoring to monitoring start standby

If *StopMonitorTimer* is 600, *MonitorStatus* changes to FALSE when the operation stops for 600 seconds or more. Monitoring start standby begins again when *ActiveInput* changes to TRUE for the next monitoring.

Specification of Subframe Selection Type

Set the subframe selection type with the member *SubFrameMode* of the structure *sMonitorMode*.
When *SubFrameMode* is FALSE, the subframe generated inside the FB is used, and while *ActiveInput* is TRUE, the value is output to the mechanism state variable if the absolute value of the velocity command (ABS (Axis.Cmd.Vel)) is equal to or larger than the specified percentage of the target velocity ((VelocityInput * (VelocityRatio * 0.01)).
When *SubFrameMode* is TRUE, enter the user-defined subframe. While *ManualSubFrame* is TRUE and *ActiveInput* is TRUE, the value is output to the mechanism state variable.

Input

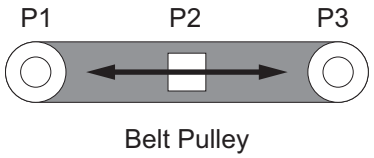


Setting Changes During Execution

Even if you change the value of *MonitorMode* during execution, the change will not be applied.
For input variables other than *MonitorMode*, changed settings will be reflected.
However, for *VelocityInput*, the value when *ActiveInput* became TRUE will be reflected.

Timing Charts

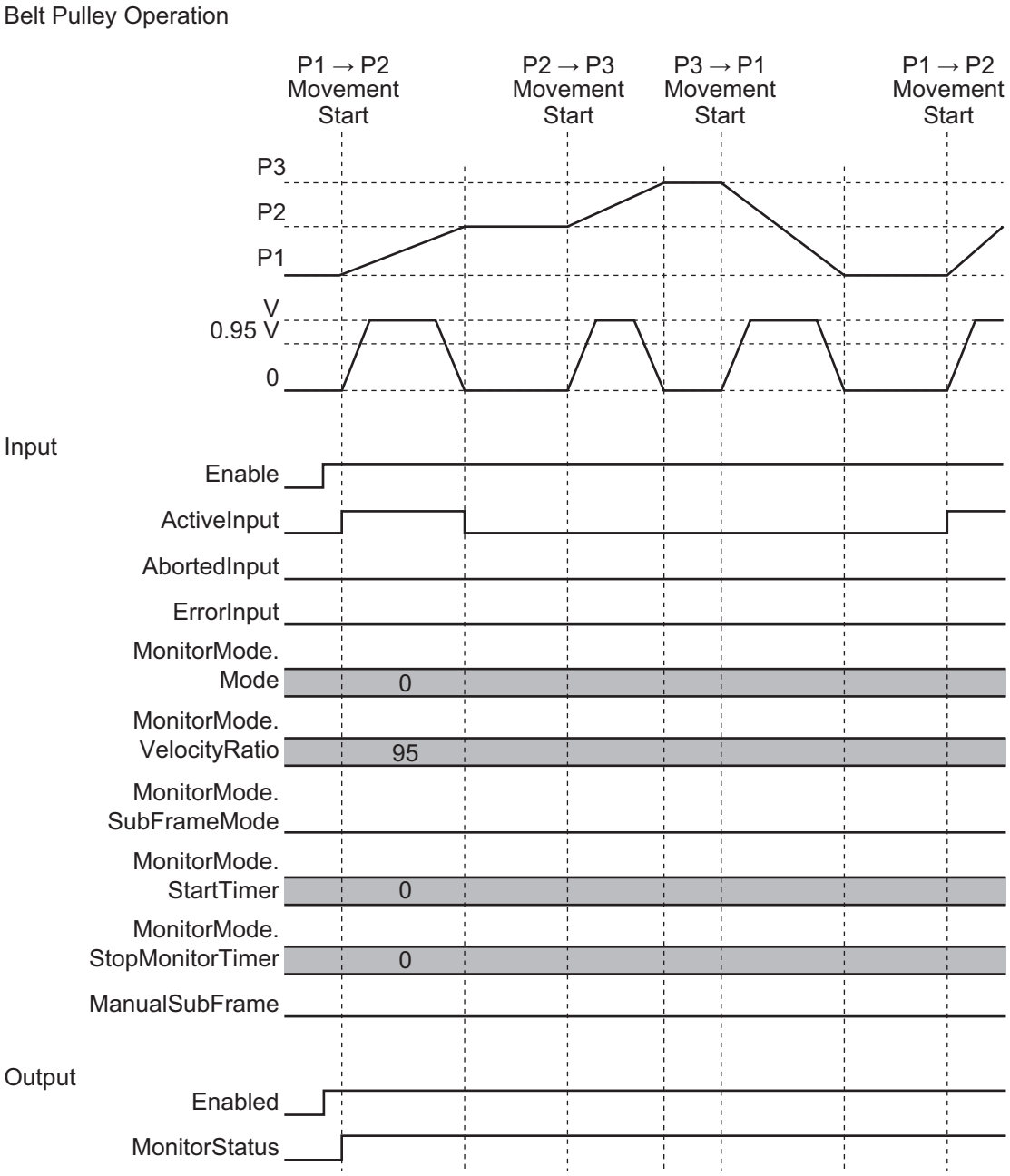
The timing chart for each state is shown below.



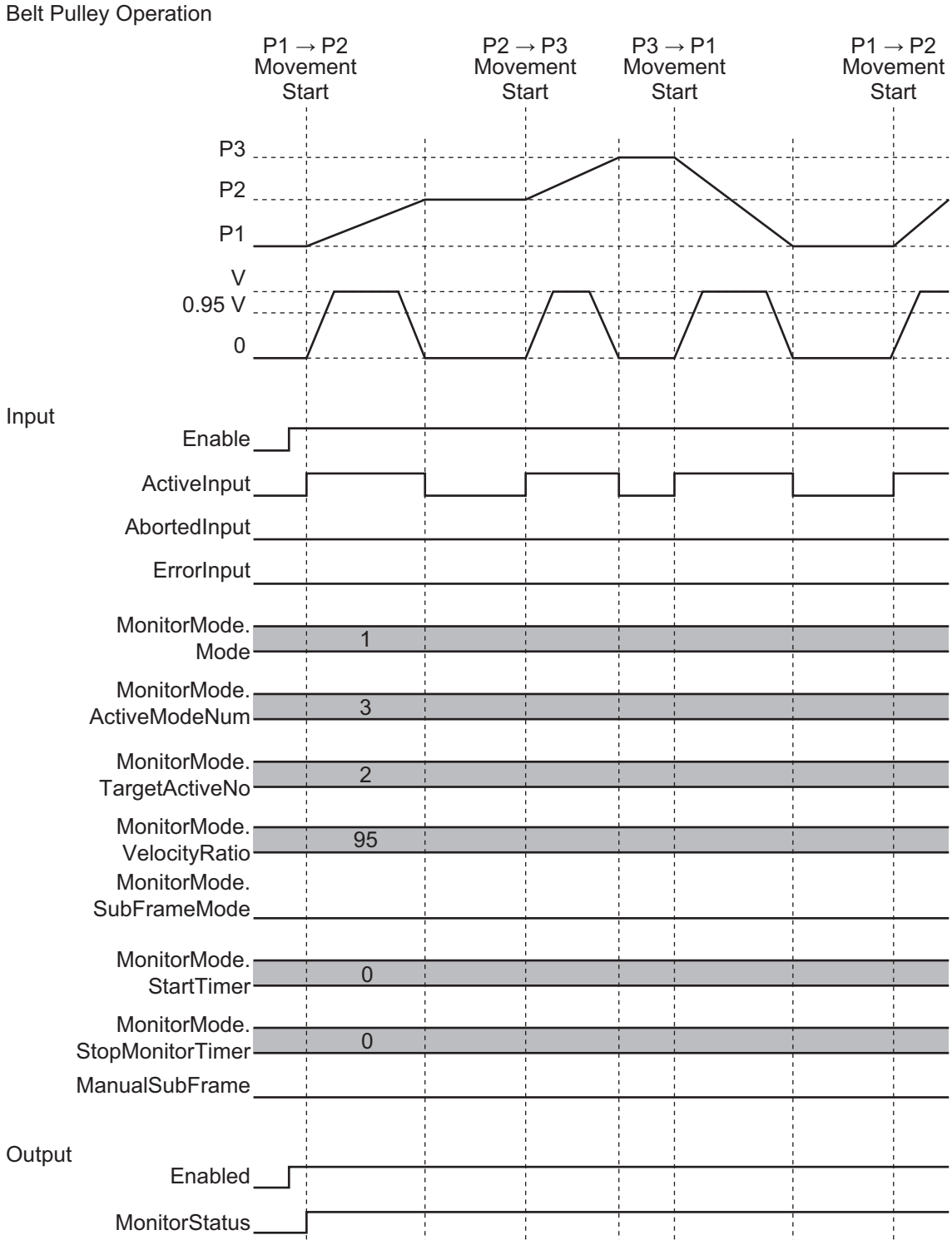
Belt Pulley

Timing Chart for Normal End

● **When *MonitorMode.Mode* = 0 and the movement from P1 to P2 is monitored**

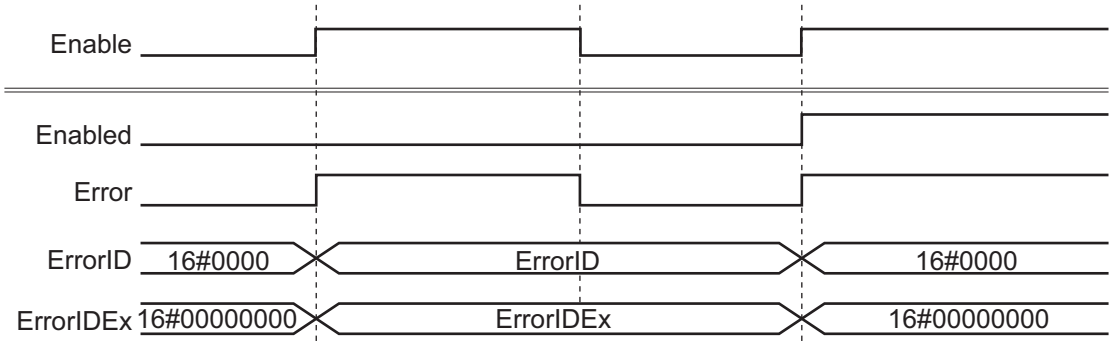


● **When *MonitorMode.Mode* = 1, *MonitorMode.ActiveMaxNum* = 3, and *MonitorMode.TargetActiveNo* = 2**

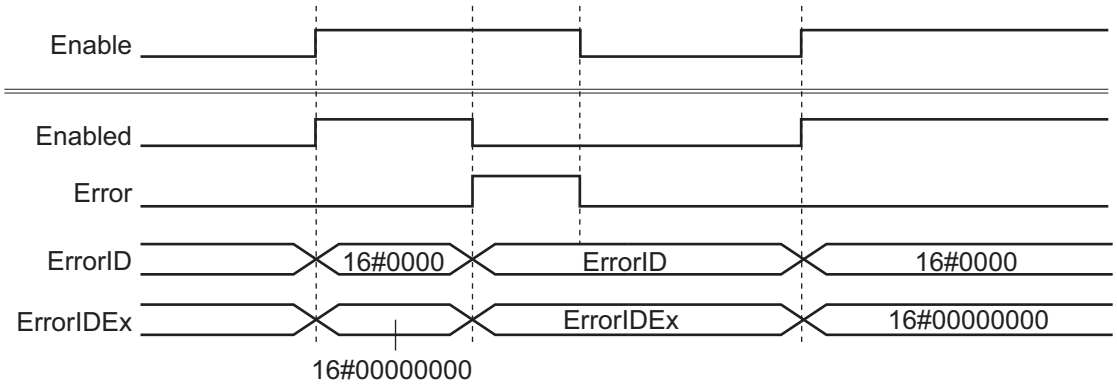


Timing Chart for Error End

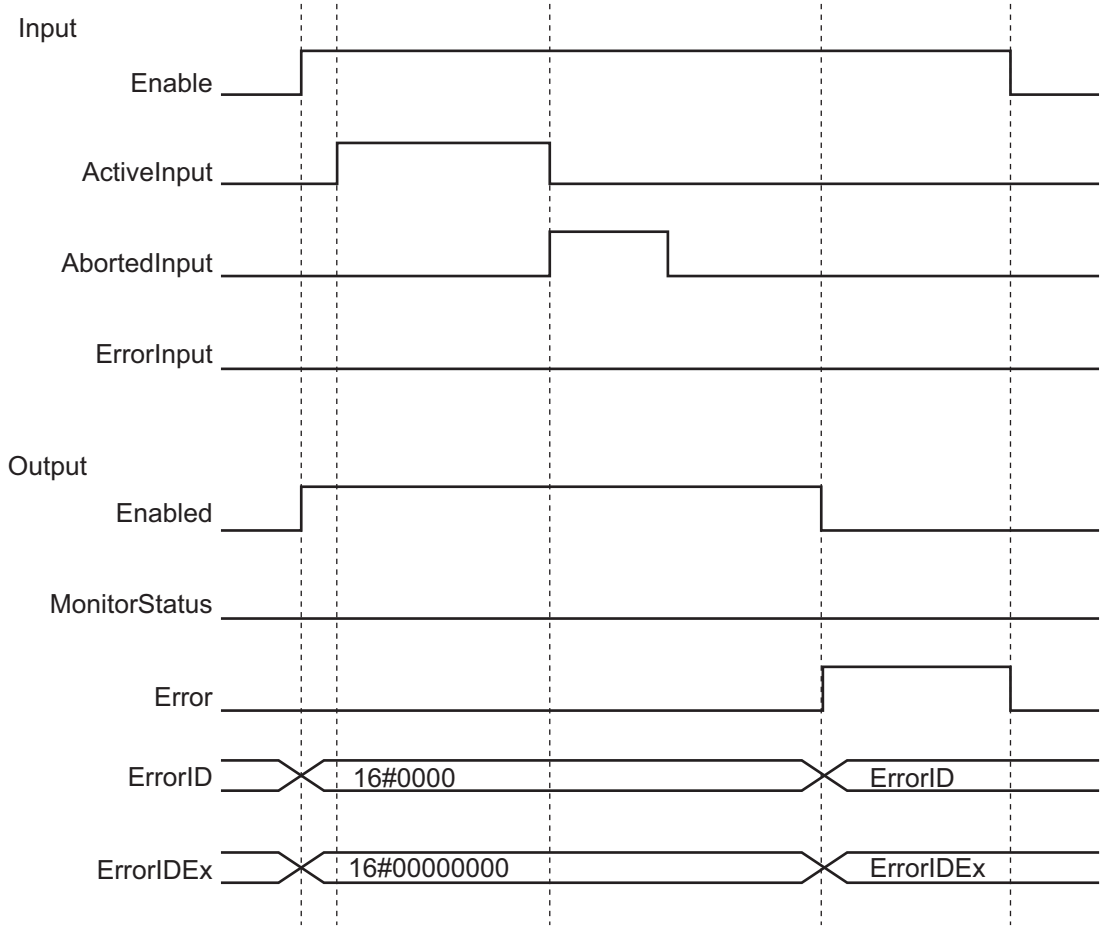
- **If error occurs when *Enable* changes to TRUE**



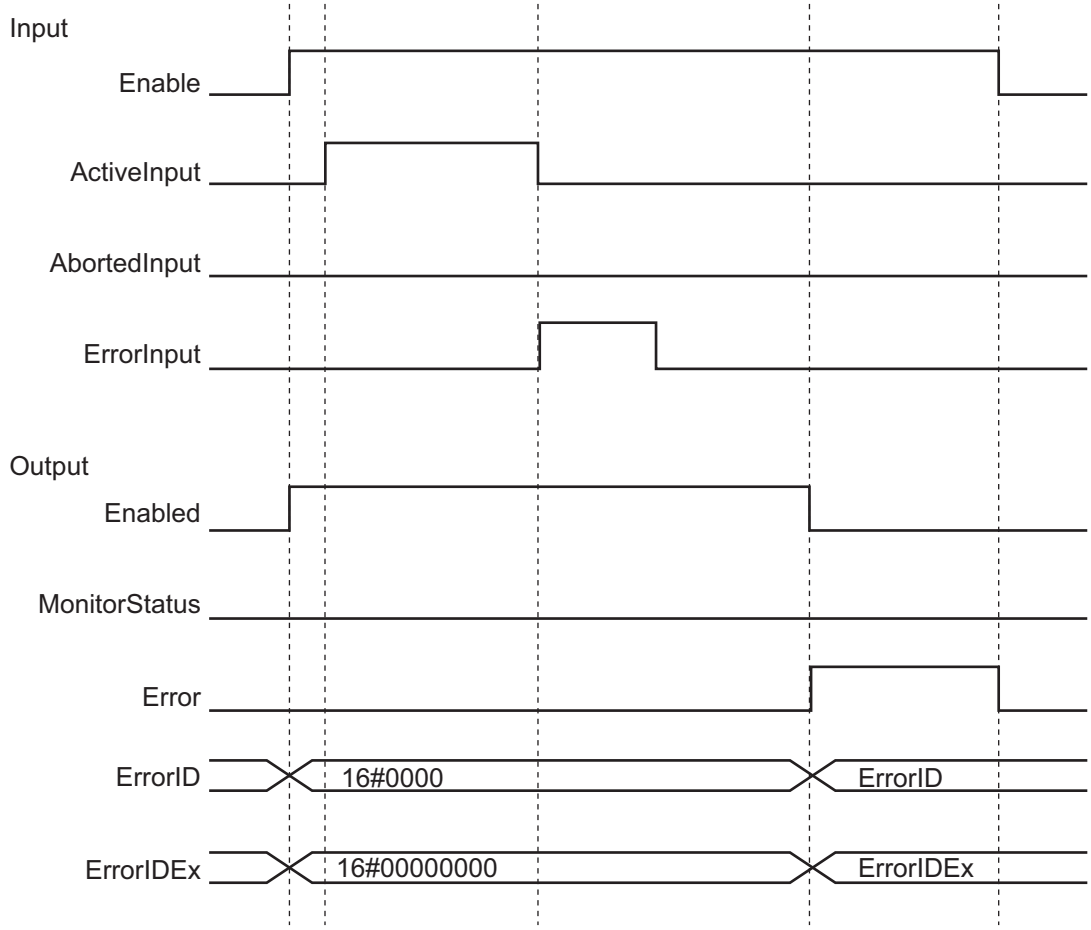
- **If error occurs while *Enable* is TRUE**



● **When *AbortedInput* changes to TRUE**



● **When *ErrorInput* changes to TRUE**



Precautions for Correct Use

Execute the AI Predictive Maintenance Library on the program assigned to the primary periodic task. If you execute it on a program that is not assigned to the primary periodic task, you cannot generate a mechanism state variable that correctly reflects the state of the mechanism.

Troubleshooting

This section provides the statuses, descriptions, and corrective actions for the error codes and expansion error codes.

Error code	Expansion error code	Status	Description	Corrective action
16#0000	16#00000000	Normal End	---	---
16#3D17	16#00000001	Mode Type Input Value Out of Range	The value of *MonitorMode.Mode* is outside the valid range.	Correct the *MonitorMode.Mode* value so that it is within the valid range.
	16#00000002	Active Count Input Value Out of Range	When *MonitorMode.Mode* is 1, the value of *MonitorMode.ActiveMaxNum* is 0.	Correct the *MonitorMode.ActiveMaxNum* value so that it is within the valid range.

Error code	Expansion error code	Status	Description	Corrective action
	16#00000003	Active Number Input Value Out of Range	When *MonitorMode.Mode* is 1, the value of *MonitorMode.TargetActiveNo* is 0.	Correct the *MonitorMode.TargetActiveNo* value so that it is within the valid range.
	16#00000004	Active Input Mismatch	When *MonitorMode.Mode* is 1, the value of *MonitorMode.ActiveMaxNum* is less than the value of *MonitorMode.TargetActiveNo.*	Set the value of *MonitorMode.ActiveMaxNum* so that it is greater than or equal to *MonitorMode.TargetActiveNo*.
	16#00000005	Invalid VelocityRatio Value	The value of *MonitorMode.VelocityRatio* is outside the valid range.	Correct the *MonitorMode.VelocityRatio* value so that it is within the valid range.
	16#00000006	Invalid Target Velocity Input	The value of *VelocityInput* is outside the valid range.	Correct the *VelocityInput* value so that it is within the valid range.
	16#00000007	Axis Command Error	The value of *ErrorInput* is TRUE.	Check the execution result of the motion control instruction. Refer to the *NJ/NX-series Motion Control Instructions Reference Manual (Cat. No. W508)* or the *NY-series Motion Control Instructions Reference Manual (Cat. No. W561)* for details.
	16#00000008	Axis Command Aborted	The value of *AbortedInput* is TRUE.	
	16#00000009	Mechanism State Variable Calculation Error	The calculation result of the mechanism status variable has become a non-number or ± ∞.	Check the input value of *MonitorMode*.

7

Sample Programming

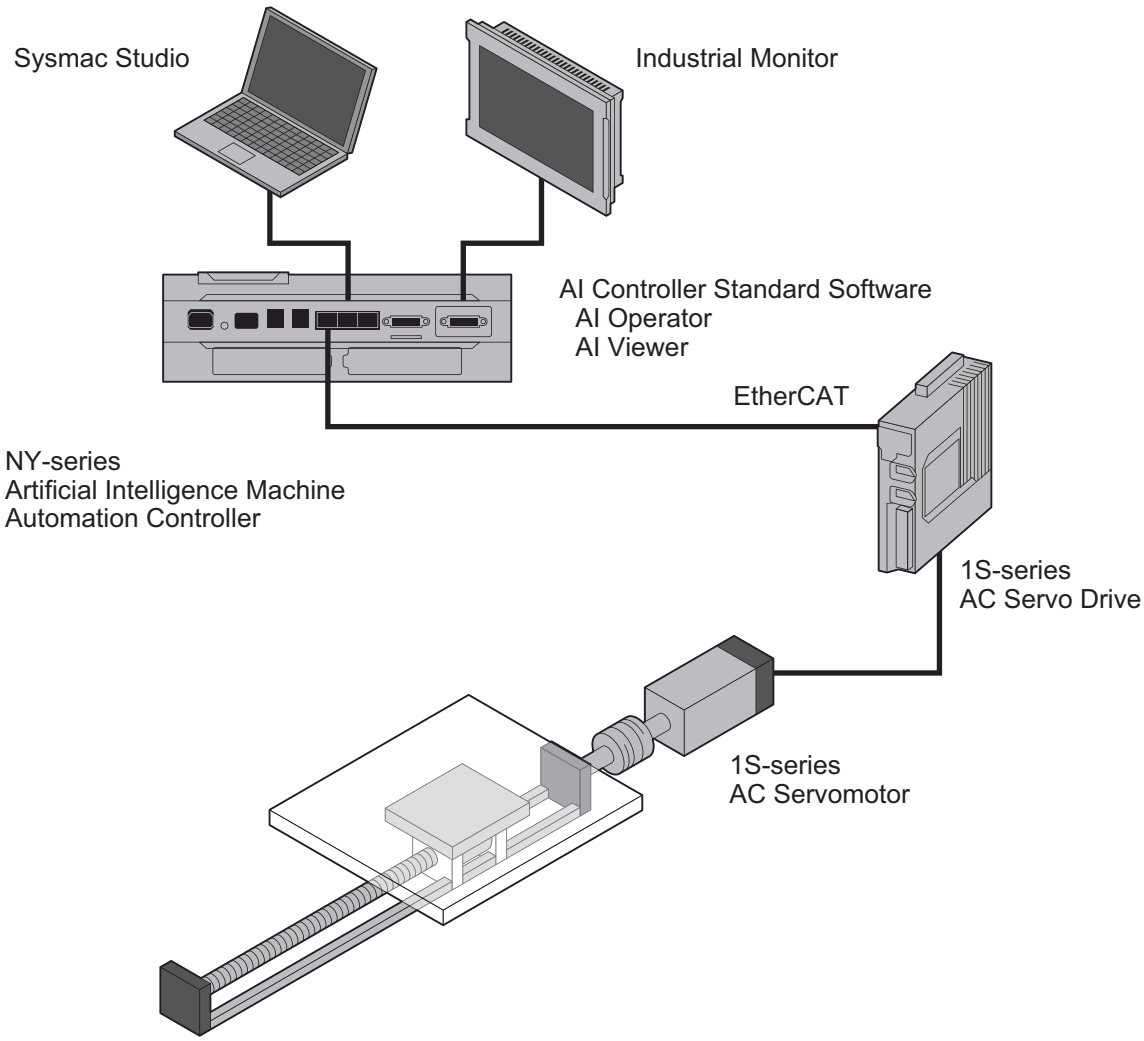
This section describes sample programming that combines multiple function blocks and functions.

7-1 Overview

This sample programming corresponds to the program part in *STEP 2: Integrate AI FB into the user program on the Sysmac Studio and transfer the program* in *1-3 Usage Procedure* on page 1 - 4.
The AI FB of the AI Predictive Maintenance Library (ball screw) generates mechanism state variables for the ball screw mechanism to be monitored.

7-2 System Configuration

The system configuration of this sample programming is shown below.



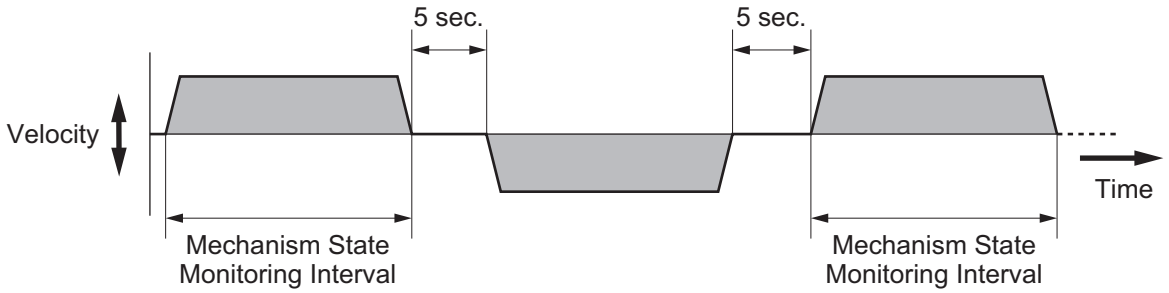
Name	Model	Notes
NY-series Artificial Intelligence Machine Automation Controller	NY512-Z500	Unit version 1.18 or later
Industrial Monitor	NYM15W-C1000	15.4-inch DVI-D Interface
AI Controller Standard Software	SYSMAC-AICSTE□□□	Version 1.00 or higher
Sysmac Studio	SYSMAC-SE2□□□	Version 1.25 or higher
Sysmac Library AI Predictive Maintenance Library (Ball Screw)	SYSMAC-ZPA002000W	Version 1.00 or higher
1S-series AC Servo Drive	R88D-1SN01L-ECT	---
1S-series AC Servomotor	R88M-1□	---
1-axis Stage	---	Ball Screw

7-3 Operation Explanation

The operation of this sample programming is shown below.

- STEP 0: When you set *Start* to TRUE, this sample programming starts operating.
  The program will continue to operate until you set *Stop* to TRUE.
- STEP 1: Enable the AI functions.
  a) Set the Feature Extraction Function and the Machine Learning Function in running status.
  b) Instruct the Time Series DB (TSDB) to start exporting. The target data is the feature values (FTR_DATA) and the equipment event monitoring results (AIS_DATA).
  Refer to the *NX/NY-series Artificial Intelligence Machine Automation Controller User's Manual (Cat. No. W594)* for details on TSDB, FTR_DATA, and AIS_DATA.
- STEP 2: Confirm that the AI functions in STEP 1 are enabled.
  When they are enabled, *Operatable* becomes TRUE.
- STEP 3: Turn the Servo ON.
- STEP 4: Shift the position by 100m in the positive direction relative to the current position, then wait 5 seconds after positioning is completed.
  After the standby time has passed, perform relative positioning by 100 mm in the negative direction, then wait 5 seconds after positioning is completed.
  After the standby time has passed, start relative positioning again in the positive direction, and repeat the positive and negative movements.
- STEP 5: While *Operatable* is TRUE, set *Enable* for the BallScrewStatus FB to TRUE and generate the mechanism state variable.
  As the monitoring interval of the mechanism state is during the relative positioning in the positive direction, connect the output of the relative positioning in the positive direction FB (MV_RLV_Fwd) to *ActiveInput*, *AbortedInput* and *ErrorInput* of the BallScrewStatus FB.

Stage Operation and Mechanism State Monitoring Interval



7-4 Parameter Settings

The settings of *MonitorMode* (Monitor Mode) for the BallScrewStatus FB are as follows.

Variable	Set value	Setting details
Mode (Mode Type)	0	*Mode* = 0: Where one FB handles one movement. The target is monitored during the period in which *ActiveInput* is ON.
ActiveMaxNum (Active Count)	1	This parameter is valid when *Mode* is 1, so it is the default value.
TargetActiveNo (Monitoring Target Active Number)	1	This parameter is valid when *Mode* is 1, so it is the default value.
MVAveCount (Moving Average Count)	50	The moving average count is set to 50 times.
SubFrameMode (Subframe Selection Type)	FALSE	The subframe generated in the FB is used.

7-5 Program

● External Variables

Name	Data type	Con-stant*1	Comment
MC_Axis000	_sAXIS_REF	○	Axis 0
_FE_Enable	BOOL	---	Machine Learning Service Enable Command
_MLE_Enable	BOOL	---	Feature Extraction Service Enable Command
_FE_Status	_eFE_STATUS	○	Feature Extraction Service Status
_MLE_Status	_eMLE_STATUS	○	Machine Learning Service Status
_TSDB_Status	_sTSDB_STATUS	○	TSDB Service Status
_TSDB_Ex-ptStartCmd	ARRAY[0..31] OF BOOL	---	TSDB Export Start Command
_TSDB_TSSta	ARRAY[0..31] OF _sTSDB_TSSta	○	TimeSeries Status

*1. ○: Constant, ---: Not constant

● Internal Variables

Name	Data type	Default	Comment
Start	BOOL	---	Program start
Stop	BOOL	---	Program stop
Operatable	BOOL	---	Data Collection start condition
Executing	RS	---	Execution status of this pro-gram
BSS	OmronLib\AI_PM_Ball-Screw_V1_0\BallScrewStatus	---	---
MonitorMode	OmronLib\AI_PM_Ball-Screw_V1_0\sMonitorMode	(Mode := 0, Active-MaxNum := 1, Tar-getActiveNo := 1, MVAveCount := 50, SubFrameMode := False)	---
ServoOn	MC_Power	---	---
MV_RLV_Fwd	MC_MoveRelative	---	---
MV_RLV_Rvs	MC_MoveRelative	---	---
Motion_Enabled	BOOL	---	Servo ON
FwdEnd	BOOL	---	Positive direction operation completion
RvsEnd	BOOL	---	Negative direction operation completion
Move_Fwd	RS	---	Positive direction movement state
Move_Rvs	RS	---	Negative direction movement state
Wait1	TON	---	Wait for 5 seconds after the movement is completed

Name	Data type	Default	Comment
Wait2	TON	---	Wait for 5 seconds after the movement is completed
TS_Command	INT	---	Start/stop of TSDB export INT#1: Start INT#2: Stop

● Task Settings

Locate in the primary periodic task.

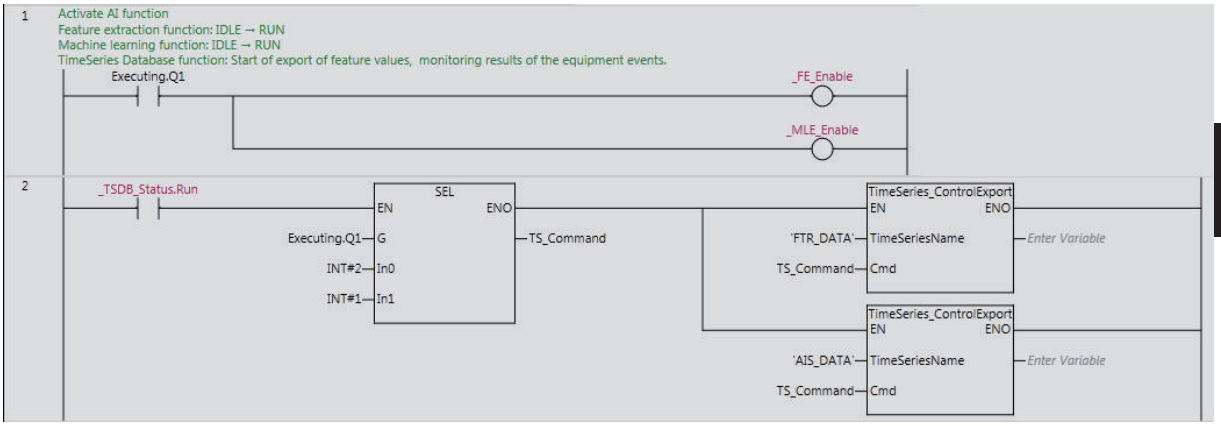
● Axis Settings

Set Axis 0 to the servo axis.

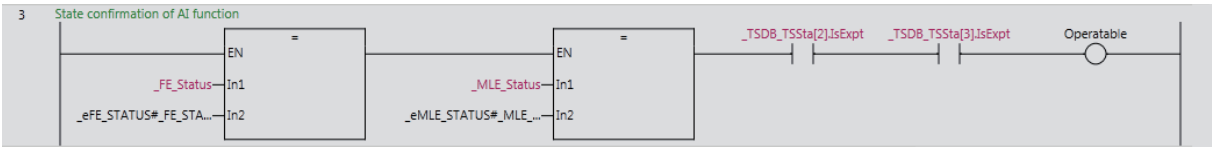
● Ladder Diagram

STEP 0

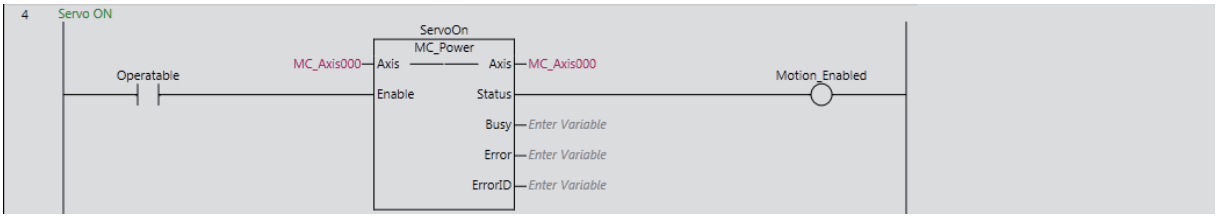
STEP 1



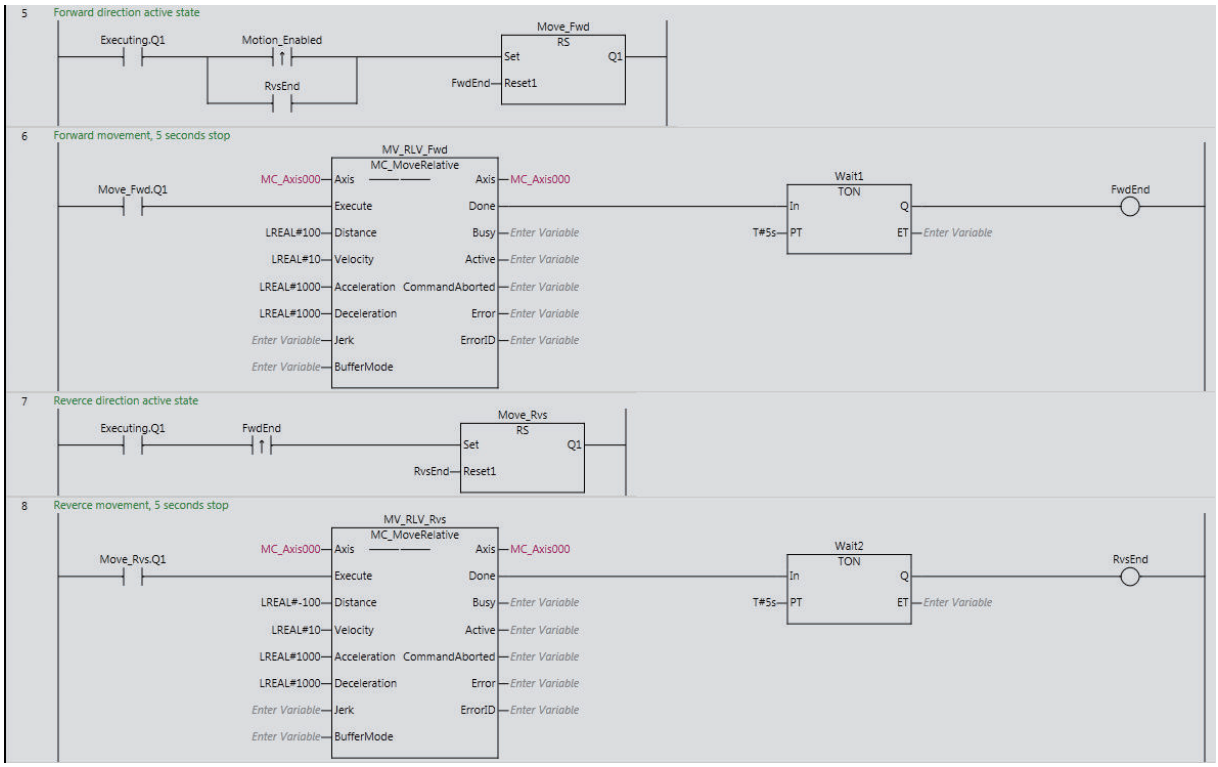
STEP 2



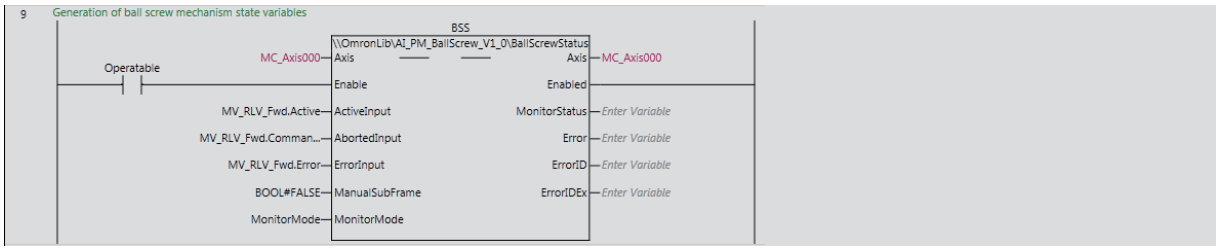
STEP 3



STEP 4


5 Forward direction active state
Executing.Q1
Motion_Enabled
RvsEnd
Move_Fwd
RS
Set
Q1
FwdEnd
Reset1
6 Forward movement, 5 seconds stop
MV_RLV_Fwd
MC_MoveRelative
Move_Fwd.Q1
MC_Axis000
Axis
Execute
Done
LREAL#100
Distance
Busy
LREAL#10
Velocity
Active
LREAL#1000
Acceleration
CommandAborted
LREAL#1000
Deceleration
Error
Enter Variable
Jerk
ErrorID
BufferMode
Wait1
TON
In
Q
T#5s
PT
ET
FwdEnd
7 Reverce direction active state
Move_Rvs
RvsEnd
8 Reverce movement, 5 seconds stop
MV_RLV_Rvs
Move_Rvs.Q1
LREAL#-100
Wait2
RvsEnd

STEP 5


9 Generation of ball screw mechanism state variables
BSS
\\OmronLib\AI_PM_BallScrew_V1_0\BallScrewStatus
Operatable
MC_Axis000
Axis
Enable
Enabled
MV_RLV_Fwd.Active
ActiveInput
MonitorStatus
MV_RLV_Fwd.Comman...
AbortedInput
Error
MV_RLV_Fwd.Error
ErrorInput
ErrorID
BOOL#FALSE
ManualSubFrame
ErrorIDEx
MonitorMode

A

Appendix

This section describes information that is convenient to know, such as library information reference methods, FB or FUN source code reference methods, etc.

A-1 Referring to Library Information

When you make an inquiry to OMRON about a library, you can refer to the library information to identify the library to ask about.
The library information is useful in identifying the target library among the libraries provided by OMRON or created by the user.
The library information consists of the attributes of the library and the attributes of function blocks and functions contained in the library.

- Attributes of libraries
  Information for identifying the library itself
- Attributes of function blocks and functions
  Information for identifying the function block and function contained in the library

Use the Sysmac Studio to access the library information.

A-1-1 Library Attributes, and FB or FUN Attributes

The following attributes of libraries, function blocks, and functions are provided as library information.

Library Attributes

No.*1	Attribute	Description
(1)	Library file name	The name of the library file
(2)	Library version	The version of the library
(3)	Author	The name of the creator of the library
(4)	Comment	The description of the library*2

*1. These numbers correspond to the numbers shown on the screen images in the next section, *A-1-2 Referring to Attributes of Libraries, Function Blocks, and Functions* on page A - 3.
*2. It is provided in English and Japanese.

Attributes of Function Blocks and Functions

No.*1	Attribute	Description
(5)	FB/FUN name	The name of the function block or function
(6)	Name space	The name of the name space for the function block or function
(7)	FB/FUN version	The version of the function block or function
(8)	Author	The name of the creator of the function block or function
(9)	FB/FUN number	The function block number or function number
(10)	Comment	The description of the function block or function *2

*1. These numbers correspond to the numbers shown on the screen images in the next section, *A-1-2 Referring to Attributes of Libraries, Function Blocks, and Functions* on page A - 3.
*2. It is provided in English and Japanese.

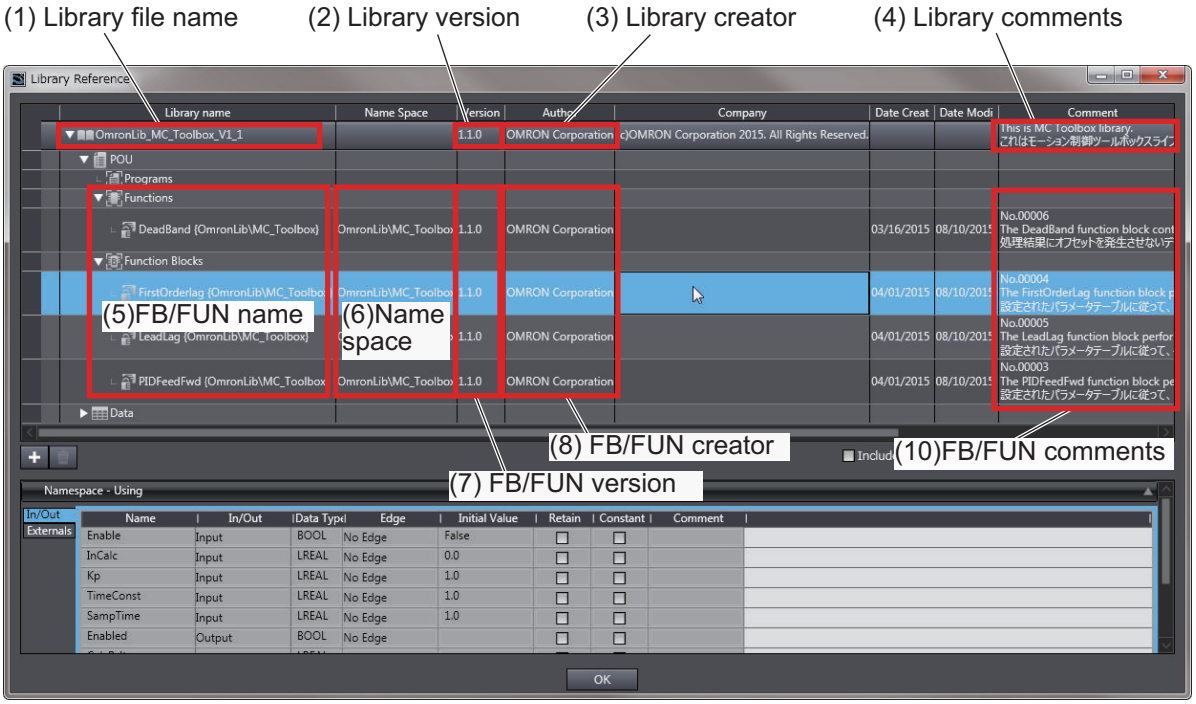
A-1-2 Referring to Attributes of Libraries, Function Blocks, and Functions

You can refer to the library attributes of library information, and FB or FUN attributes at the following Sysmac Studio locations.

- Library Reference Dialog Box
- Toolbox
- Programming screen

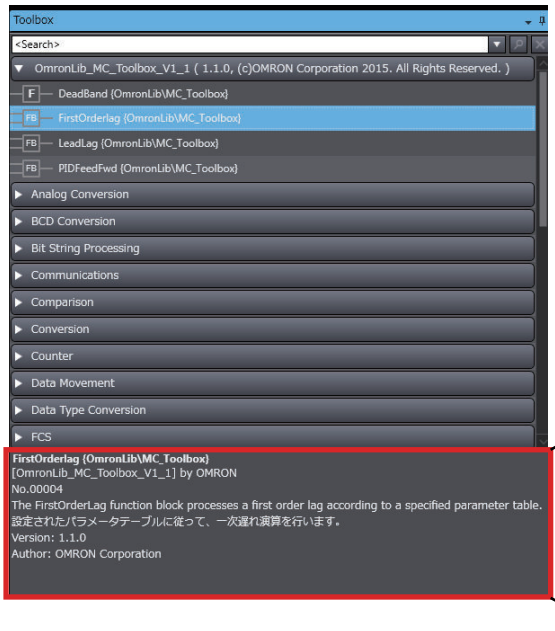
Library Reference Dialog Box

When you refer to the libraries, the library information is displayed at the locations shown below.



Toolbox

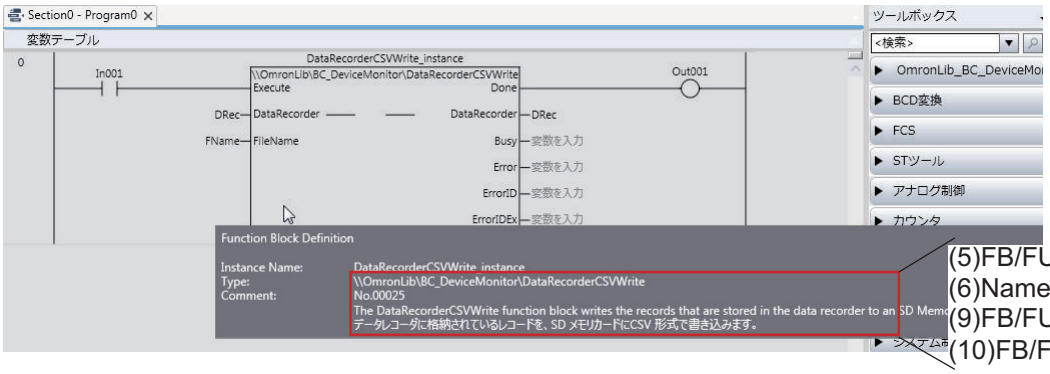
Select a function block or function to display its library information at the bottom of the Toolbox Pane. The text "**by OMRON**" which is shown on the right of the library name (1) indicates that this library was provided by OMRON.



(5)FB/FUN name (6)Name space
(1)Library file name
(9)FB/FUN number
(10)FB/FUN comment
(7)FB/FUN version
(8)FB/FUN author

Programming Screen

Place the mouse on a function block and function to display the library information in a tooltip.



(5)FB/FUN name
(6)Name space
(9)FB/FUN number
(10)FB/FUN comment

I

Index

Index

B

C

OMRON Corporation **Industrial Automation Company**
Kyoto, JAPAN

Contact: www.ia.omron.com

Regional Headquarters
OMRON EUROPE B.V.
Wegalaan 67-69, 2132 JD Hoofddorp
The Netherlands
Tel: (31)2356-81-300/Fax: (31)2356-81-388

OMRON ELECTRONICS LLC
2895 Greenspoint Parkway, Suite 200
Hoffman Estates, IL 60169 U.S.A.
Tel: (1) 847-843-7900/Fax: (1) 847-843-7787

OMRON ASIA PACIFIC PTE. LTD.
No. 438A Alexandra Road # 05-05/08 (Lobby 2),
Alexandra Technopark,
Singapore 119967
Tel: (65) 6835-3011/Fax: (65) 6835-2711

OMRON (CHINA) CO., LTD.
Room 2211, Bank of China Tower,
200 Yin Cheng Zhong Road,
PuDong New Area, Shanghai, 200120, China
Tel: (86) 21-5037-2222/Fax: (86) 21-5037-2200

Authorized Distributor:

Cat. No. W610-E1-01 1018