

# OMRON

**Sysmac Library**

**User's Manual for High-speed Analog Inspection Li-  
brary**

**SYSMAC-XR016**

## 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 an NJ/NX-series CPU Unit, PC for NY-series production.

This manual contains information that is necessary to use Function block for High-speed Analog Inspection Library (hereafter, sometimes abbreviated to FB). 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 function block specifications. It does not describe application restrictions or combination restrictions for Controllers, Units, and 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 High-speed Analog Inspection Library records analog input values of the NX-series High-speed Analog Input Unit (NX-HAD□□□) in chronological order.

It provides the functionality required for product inspection in the production process, including calculation of feature values such as the maximum, minimum, and average values, comparison test based on master data, and recording of data files.

## 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.

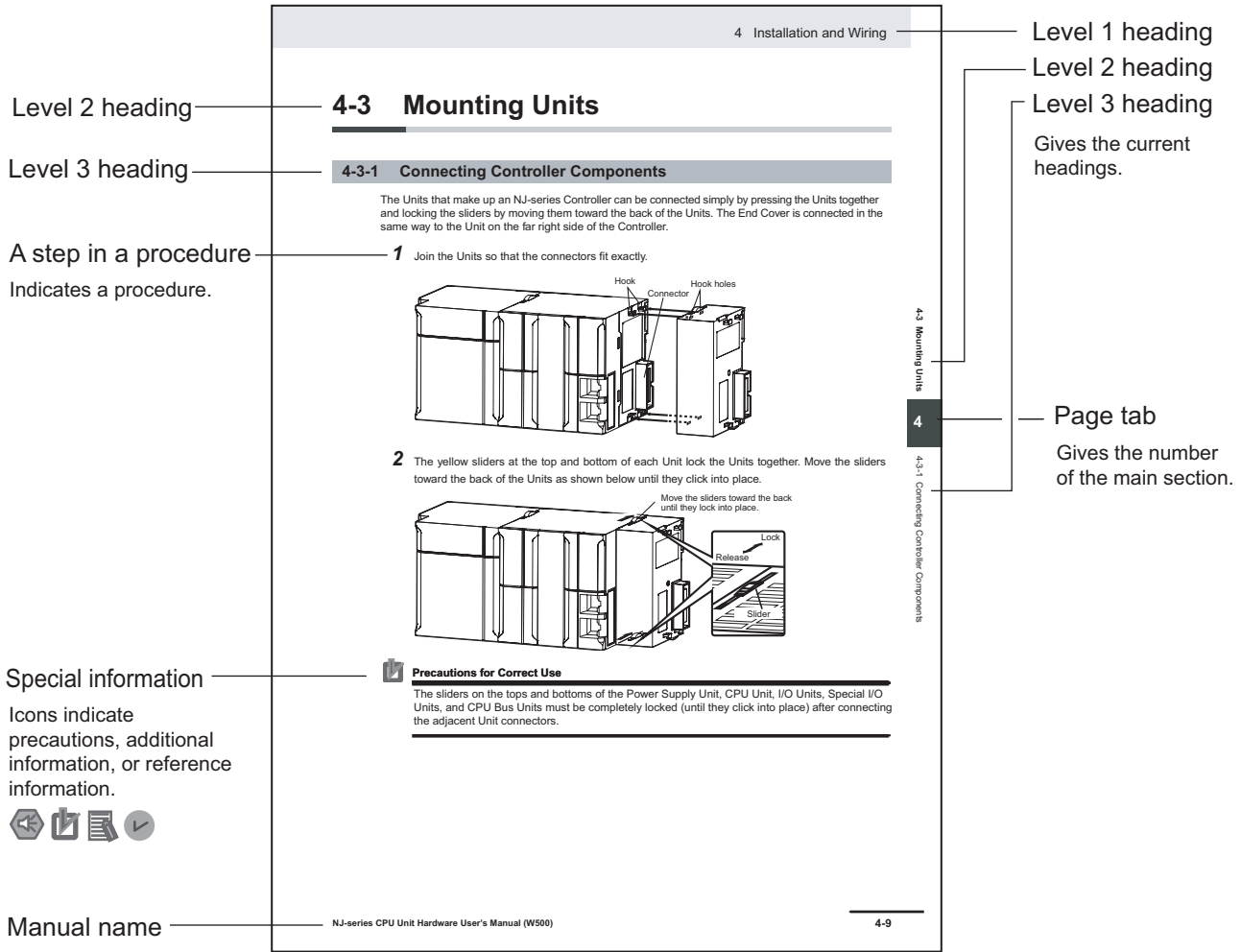
## Applicable Products

For the model numbers and versions of an NJ/NX-series CPU Unit, NY-series Industrial PC, and the Sysmac Studio that this library supports, refer to *Sysmac Library Version Information* in the *SYSMAC-XR□□□ Sysmac Library Catalog (Cat. No. P102)*. This catalog can be downloaded from the OMRON website (<http://www.ia.omron.com/products/family/3459/download/catalog.html>).

# 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.



# CONTENTS

---

<b>Introduction .....</b>	<b>1</b>
Features of the Library.....	1
Intended Audience.....	1
Applicable Products .....	1
<b>Manual Structure.....</b>	<b>2</b>
Page Structure.....	2
Special Information .....	3
<b>Terms and Conditions Agreement.....</b>	<b>8</b>
Warranty, Limitations of Liability .....	8
Application Considerations .....	9
Disclaimers .....	9
<b>Safety Precautions.....</b>	<b>11</b>
Definition of Precautionary Information.....	11
Symbols .....	11
CAUTIONS .....	11
<b>Precautions for Correct Use .....</b>	<b>13</b>
Using the Library.....	13
<b>Related Manuals.....</b>	<b>14</b>
<b>Revision History.....</b>	<b>17</b>
<b>Sections in this Manual .....</b>	<b>19</b>

## Section 1 Sysmac Library Usage Procedure

---

<b>1-1 Procedure to Use Sysmac Library Installed Using the Installer.....</b>	<b>1 - 2</b>
1-1-1 Using a Newly Installed Sysmac Library .....	1 - 2
1-1-2 Using an Upgraded Sysmac Library .....	1 - 4
<b>1-2 How to use Sysmac Library in the CPU Unit or Industrial PC .....</b>	<b>1 - 6</b>

## Section 2 High-speed Analog Inspection Library

---

<b>2-1 Overview .....</b>	<b>2 - 2</b>
2-1-1 System Configuration Example.....	2 - 2
2-1-2 Library Configuration.....	2 - 2
2-1-3 Data Flow and FB/FUN Structure .....	2 - 3

## Section 3 Common Specifications of Function Blocks

---

<b>3-1 Common Variables.....</b>	<b>3 - 2</b>
3-1-1 Definition of Input Variables and Output Variables .....	3 - 2
3-1-2 Execute-type Function Blocks.....	3 - 3
3-1-3 Enable-type Function Blocks.....	3 - 5
<b>3-2 Precautions .....</b>	<b>3 - 7</b>
3-2-1 Nesting .....	3 - 7
3-2-2 Instruction Options .....	3 - 7

3-2-3	Re-execution of Function Blocks.....	3 - 7
-------	--------------------------------------	-------

## Section 4 FB/FUN Individual Specifications (NX\_HAD)

<b>DeviceVariableToArray_***</b> .....	<b>4 - 2</b>
Library Information .....	4 - 2
Input Variables .....	4 - 3
Output Variables .....	4 - 3
Input-Output Variables .....	4 - 3
Function .....	4 - 3
Precautions for Correct Use .....	4 - 4
Sample Programming 1 .....	4 - 5
Sample Programming 2 .....	4 - 7
<b>ScaleTrans_HAD</b> .....	<b>4 - 11</b>
Library Information .....	4 - 11
Input Variables .....	4 - 11
Output Variables .....	4 - 12
Input-Output Variables .....	4 - 12
Function .....	4 - 12
Sample Programming .....	4 - 13
<b>LimitAlarm_HAD</b> .....	<b>4 - 14</b>
Library Information .....	4 - 14
Input Variables .....	4 - 14
Output Variables .....	4 - 15
Input-Output Variables .....	4 - 16
Function .....	4 - 16
Timing Chart .....	4 - 18
Troubleshooting .....	4 - 19
Sample Programming .....	4 - 19

## Section 5 FB/FUN Individual Specifications (DataRecorder)

<b>TrigControl</b> .....	<b>5 - 2</b>
Library Information .....	5 - 2
Input Variables .....	5 - 2
Output Variables .....	5 - 3
Input-Output Variables .....	5 - 3
Structure .....	5 - 4
Function .....	5 - 4
Timing Chart .....	5 - 6
Troubleshooting .....	5 - 7
Sample Programming .....	5 - 7
<b>DataRecorder</b> .....	<b>5 - 12</b>
Library Information .....	5 - 12
Input Variables .....	5 - 12
Output Variables .....	5 - 13
Input-Output Variables .....	5 - 13
Structure .....	5 - 14
Function .....	5 - 14
Timing Chart .....	5 - 16
Troubleshooting .....	5 - 17
Sample Programming .....	5 - 18
<b>LimitTest</b> .....	<b>5 - 19</b>
Library Information .....	5 - 19
Input Variables .....	5 - 19
Output Variables .....	5 - 20
Input-Output Variables .....	5 - 20
Structure .....	5 - 21
Function .....	5 - 21



Timing Chart .....	5 - 22
Precautions for Correct Use .....	5 - 23
Troubleshooting .....	5 - 23
Sample Programming .....	5 - 24
<b>CalcFeatureValues .....</b>	<b>5 - 25</b>
Library Information .....	5 - 25
Input Variables .....	5 - 25
Output Variables .....	5 - 26
Input-Output Variables .....	5 - 27
Structure .....	5 - 27
Function .....	5 - 27
Timing Chart .....	5 - 28
Precautions for Correct Use .....	5 - 29
Troubleshooting .....	5 - 29
Sample Programming .....	5 - 29
<b>LogDataToCSV .....</b>	<b>5 - 34</b>
Library Information .....	5 - 34
Input Variables .....	5 - 34
Output Variables .....	5 - 35
Input-Output Variables .....	5 - 35
Structure .....	5 - 35
Function .....	5 - 36
Additional Information .....	5 - 37
Timing Chart .....	5 - 37
Precautions for Correct Use .....	5 - 38
Troubleshooting .....	5 - 39
Sample Programming .....	5 - 40
<b>CSVToLogData .....</b>	<b>5 - 41</b>
Library Information .....	5 - 41
Input Variables .....	5 - 41
Output Variables .....	5 - 42
Input-Output Variables .....	5 - 42
Structure .....	5 - 42
Function .....	5 - 43
Timing Chart .....	5 - 43
Precautions for Correct Use .....	5 - 44
Troubleshooting .....	5 - 45
Sample Programming .....	5 - 46

## Appendix

---

<b>A-1 Referring to Library Information.....</b>	<b>A - 2</b>
A-1-1 Library Attributes, and FB or FUN Attributes .....	A - 2
A-1-2 Referring to Attributes of Libraries, Function Blocks, and Functions .....	A - 3
<b>A-2 Referring to Function Block and Function Source Codes .....</b>	<b>A - 5</b>

## 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 NJ/NX-series CPU Unit, PC for NY-series production.

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.

## CAUTIONS



### Caution

Read all related manuals carefully before you use this library.



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.



---

Do not allow anyone other than personnel corresponding to *Intended Audience* on page 1 in this manual, or personnel under the guidance of such, to use the Sysmac library and its manual.



---

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 program.
  - Create a user program that will produce the intended device operation.
  - Check the user program for proper execution before you use it for actual operation.
- 



# Precautions for Correct Use

---

## Using the Library

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

# Related Manuals

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

Manual name	Man. No.	Model	Application	Description
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. <ul style="list-style-type: none"> <li>• CPU Unit operation</li> <li>• CPU Unit features</li> <li>• Initial settings</li> <li>• Programming based on IEC 61131-3 language specifications</li> </ul>
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 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.
NJ/NY-series NC Integrated Controller User's Manual	O030	NJ501-5300 NY532-5400	Performing numerical control with NJ/NY-series Controllers.	Describes the functionality to perform the numerical control.
NJ/NY-series G code Instructions Reference Manual	O031	NJ501-5300 NY532-5400	Learning about the specifications of the G code/M code instructions.	The G code/M code instructions 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.
CNC Operator Operation Manual	O032	SYSMAC-RTNC0□□□□D	Learning an introduction of the CNC Operator and how to use it.	An introduction of the CNC Operator, installation procedures, basic operations, connection operations, and operating procedures for main functions are described.



Manual name	Man. No.	Model	Application	Description
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-series CPU Unit Hardware User's Manual	W500	NJ501-□□□□ NJ301-□□□□ NJ101-□□□□	Learning the basic specifications of the NJ-series CPU Units, including introductory information, designing, installation, and maintenance. Mainly hardware information is provided.	An introduction to the entire NJ-series system is provided along with the following information on the CPU Unit. <ul style="list-style-type: none"> <li>• Features and system configuration</li> <li>• Introduction</li> <li>• Part names and functions</li> <li>• General specifications</li> <li>• Installation and wiring</li> <li>• Maintenance and inspection</li> </ul>
NY-series IPC Machine Controller Industrial Panel PC / Industrial Box PC Software User's Manual	W558	NY532-□□□□ NY512-□□□□	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. <ul style="list-style-type: none"> <li>• Controller operation</li> <li>• Controller features</li> <li>• Controller settings</li> <li>• Programming based on IEC 61131-3 language specifications</li> </ul>
NY-series Instructions Reference Manual	W560	NY532-□□□□ NY512-□□□□	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 Troubleshooting Manual	W564	NY532-□□□□ NY512-□□□□	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.
NY-series IPC Machine Controller Industrial Panel PC / Industrial Box PC Motion Control User's Manual	W559	NY532-□□□□ NY512-□□□□	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.

Manual name	Man. No.	Model	Application	Description
NY-series Motion Control Instructions Reference Manual	W561	NY532-□□□□ NY512-□□□□	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 EtherNet/IP™ Port User's Manual	W563	NY532-□□□□ NY512-□□□□	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.
NX-series EtherCAT® Coupler Unit User's Manual	W519	NX-ECC20□	Learning how to use an NX-series EtherCAT Coupler Unit and EtherCAT Slave Terminals.	The following items are described: the overall system and configuration methods of an EtherCAT Slave Terminal (which consists of an NX-series EtherCAT Coupler Unit and NX Units), and information on hardware, setup, and functions to set up, control, and monitor NX Units through EtherCAT.
NX-series Analog I/O Units User's Manual for High-speed Analog Input Units	SBCA-461	NX-HAD□□□	Learning how to use the NX-series High-speed Analog Input Unit.	The hardware, setup methods, and functions of the NX-series High-speed Analog Input Units 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.

<b>Cat. No.</b>	<b>W607-E1-02</b>
-----------------	-------------------

↑  
Revision code

Revision code	Date	Revised content
01	May 2018	Original production
02	January 2019	Added the target model number.



# Sections in this Manual

---

<b>1</b>	<b>Sysmac Library Usage Procedure</b>	<b>1</b>
<b>2</b>	<b>High-speed Analog Inspection Library</b>	<b>2</b>
<b>3</b>	<b>Common Specifications of Function Blocks</b>	<b>3</b>
<b>4</b>	<b>FB/FUN Individual Specifications (NX_HAD)</b>	<b>4</b>
<b>5</b>	<b>FB/FUN Individual Specifications (DataRecorder)</b>	<b>A</b>
<b>A</b>	<b>Appendix</b>	<b>A</b>
<b>I</b>	<b>Index</b>	<b>I</b>



# 1

## 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.

---

<b>1-1</b>	<b>Procedure to Use Sysmac Library Installed Using the Installer.....</b>	<b>1 - 2</b>
1-1-1	Using a Newly Installed Sysmac Library.....	1 - 2
1-1-2	Using an Upgraded Sysmac Library .....	1 - 4
<b>1-2</b>	<b>How to use Sysmac Library in the CPU Unit or Industrial PC .....</b>	<b>1 - 6</b>

# 1-1 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

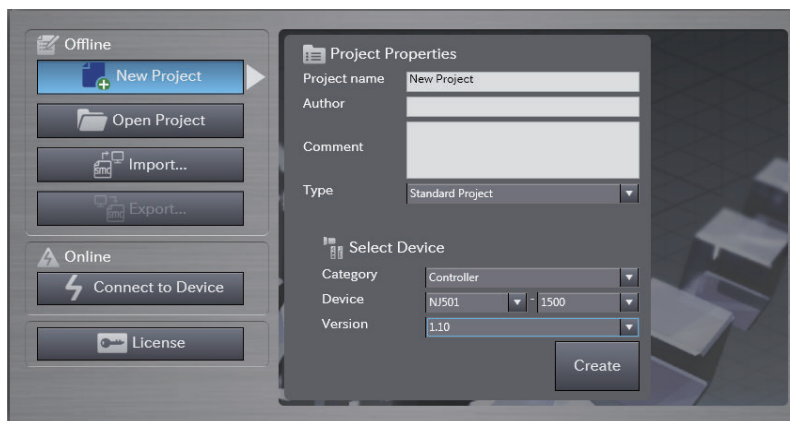


## Version Information

To use Sysmac Library, you need Sysmac Studio Ver.1.14 or higher.

### 1-1-1 Using a Newly Installed Sysmac Library

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



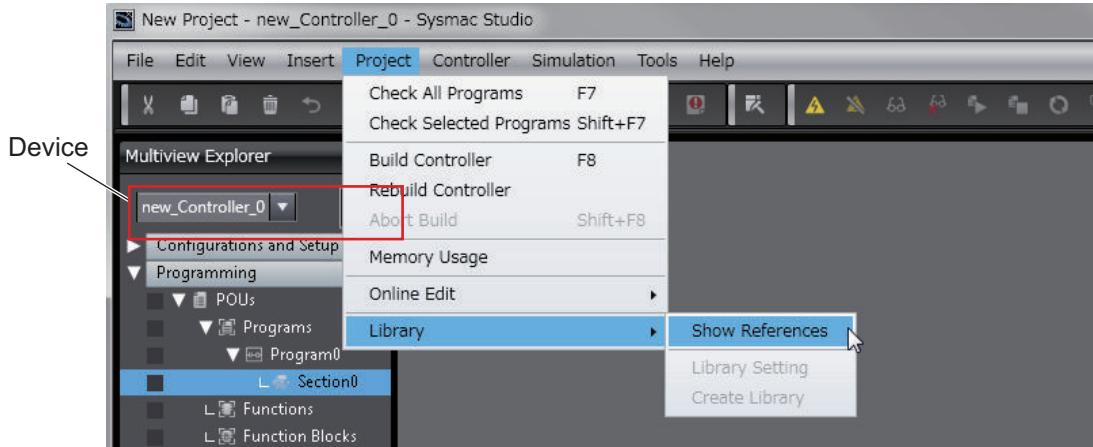
## Precautions for Correct Use

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 .

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

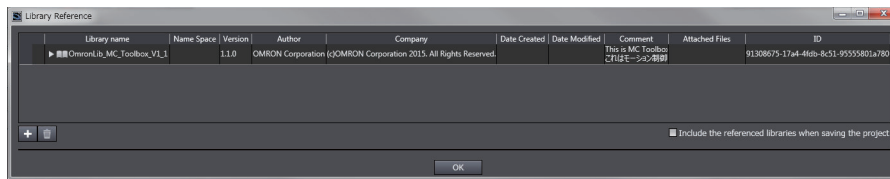




**Precautions for Correct Use**

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.**



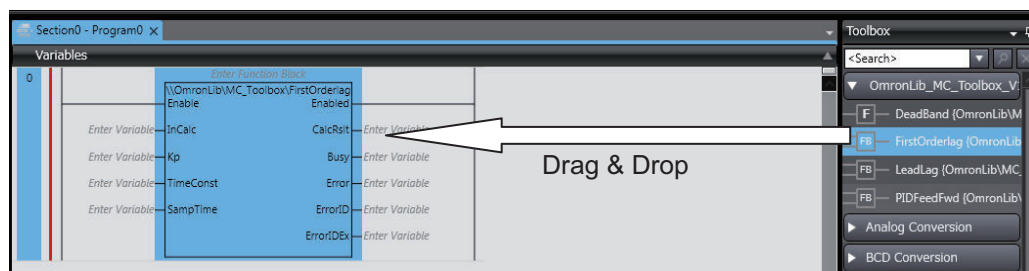
Sysmac Library is read into the project.

Now, when you select the Ladder Editor or ST Editor, the function blocks and functions included in the Sysmac Library appear in the Toolbox.

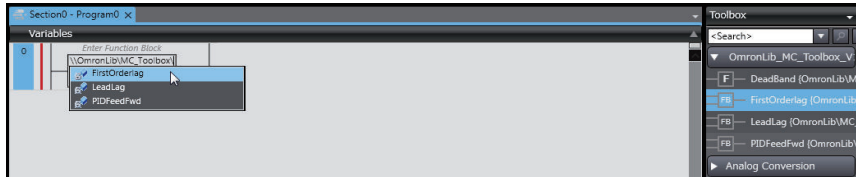
For the procedure for adding and setting libraries in the above screen, refer to *Sysmac Studio Version 1 Operation Manual (Cat. No. W504)*.

**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.

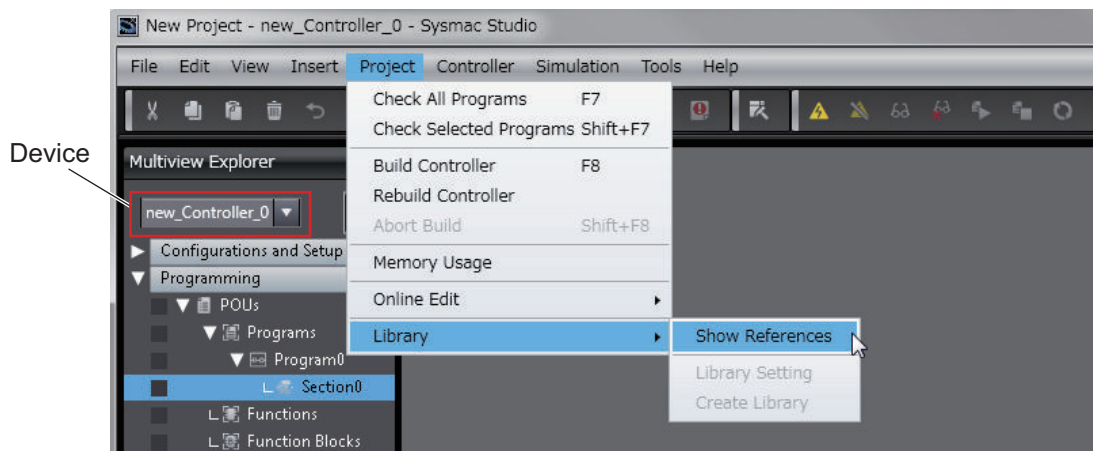


- Right-click the Ladder Editor, select **Insert Function Block** in the menu, and enter the fully qualified name (¥¥namespace¥¥FBname).




## 1-1-2 Using an Upgraded Sysmac Library

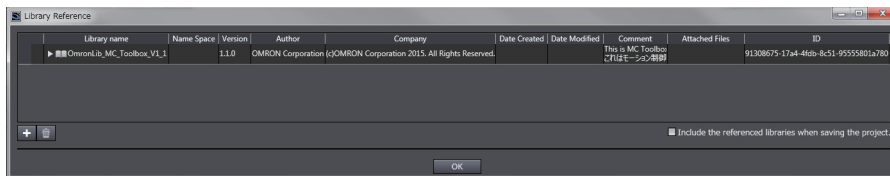
- 1 Start Sysmac Studio and open a project in which any old-version Sysmac Library is included.
- 2 Select **Project - Library - Show References**.



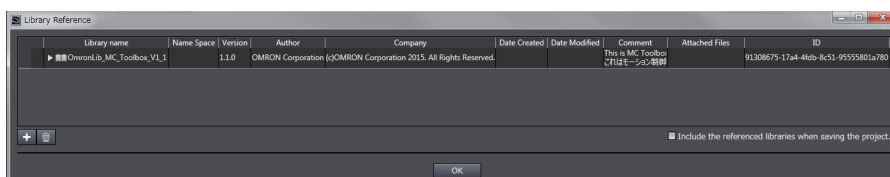
### Precautions for Correct Use

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**.

---

## 1-2 How to use Sysmac Library in the CPU Unit or Industrial PC

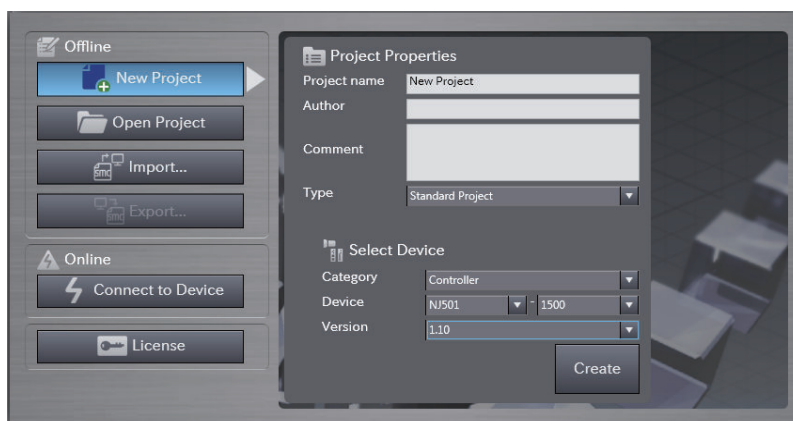
Even when Sysmac Library is not installed on your computer, you can use Sysmac Library by uploading it from the CPU Unit or Industrial PC to your computer.

The procedure to use Sysmac Library in the CPU Unit or Industrial PC is as follows.

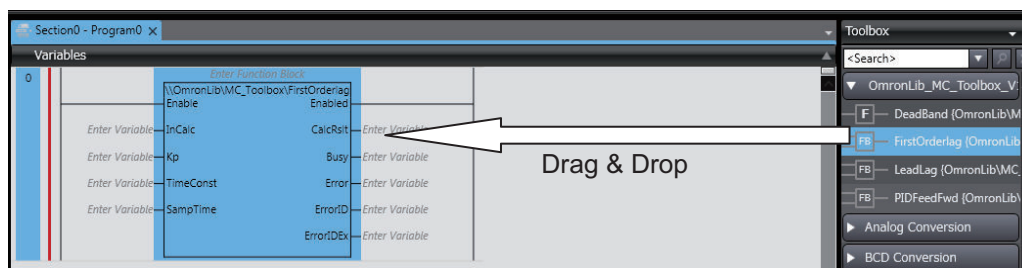
### Version Information

To use Sysmac Library, you need Sysmac Studio Ver.1.14 or higher.

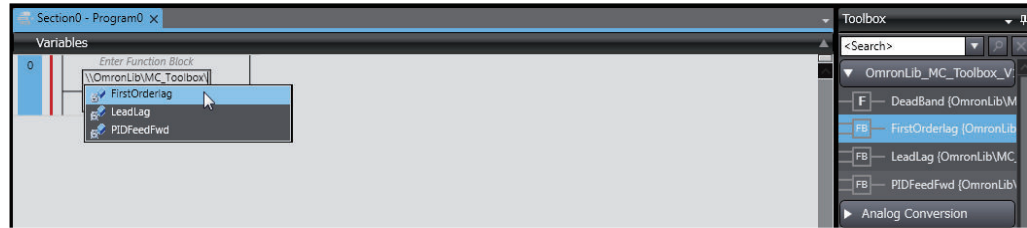
- 1 Start the Sysmac Studio and create a new project in which you want to use Sysmac Library.



- 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 (¥¥namespace¥¥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.



# 2

## High-speed Analog Inspection Library

This section describes the shared specifications of each FB in the High-speed Analog Inspection Library.

---

<b>2-1</b>	<b>Overview .....</b>	<b>2 - 2</b>
2-1-1	System Configuration Example .....	2 - 2
2-1-2	Library Configuration .....	2 - 2
2-1-3	Data Flow and FB/FUN Structure .....	2 - 3

## 2-1 Overview

The High-speed Analog Inspection Library records analog input values of the NX-series High-speed Analog Input Unit (NX-HAD□□□) in a chronological order.

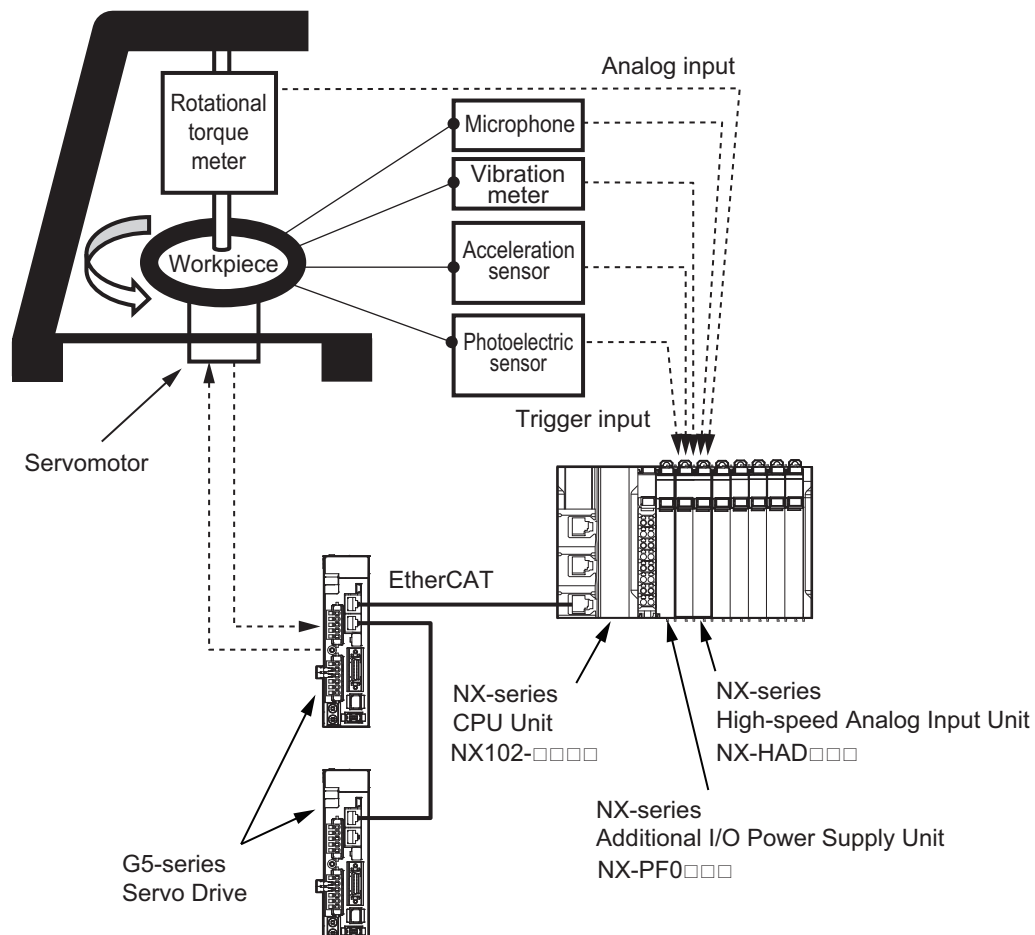
In addition, it provides various functions required for product inspection in the production process, such as feature-value calculation and pass-fail judgment based on the recorded data.

### 2-1-1 System Configuration Example

The figure below shows an application system configuration example for characteristic test equipment that uses this library.

The NX-series High-speed Analog Input Unit reads analog input signals of noises, vibrations, and torque values generated during a test, and uses the FB/FUN of this library to make pass-fail judgment of the workpiece.

You can use input from a photoelectric sensor and other devices as trigger input to efficiently obtain the analog input data necessary for the test.



### 2-1-2 Library Configuration

This library consists of two library files, NX\_HAD and DataRecorder.

These libraries can be used either in combination or separately.



The FB/FUN included in each library file are shown below.

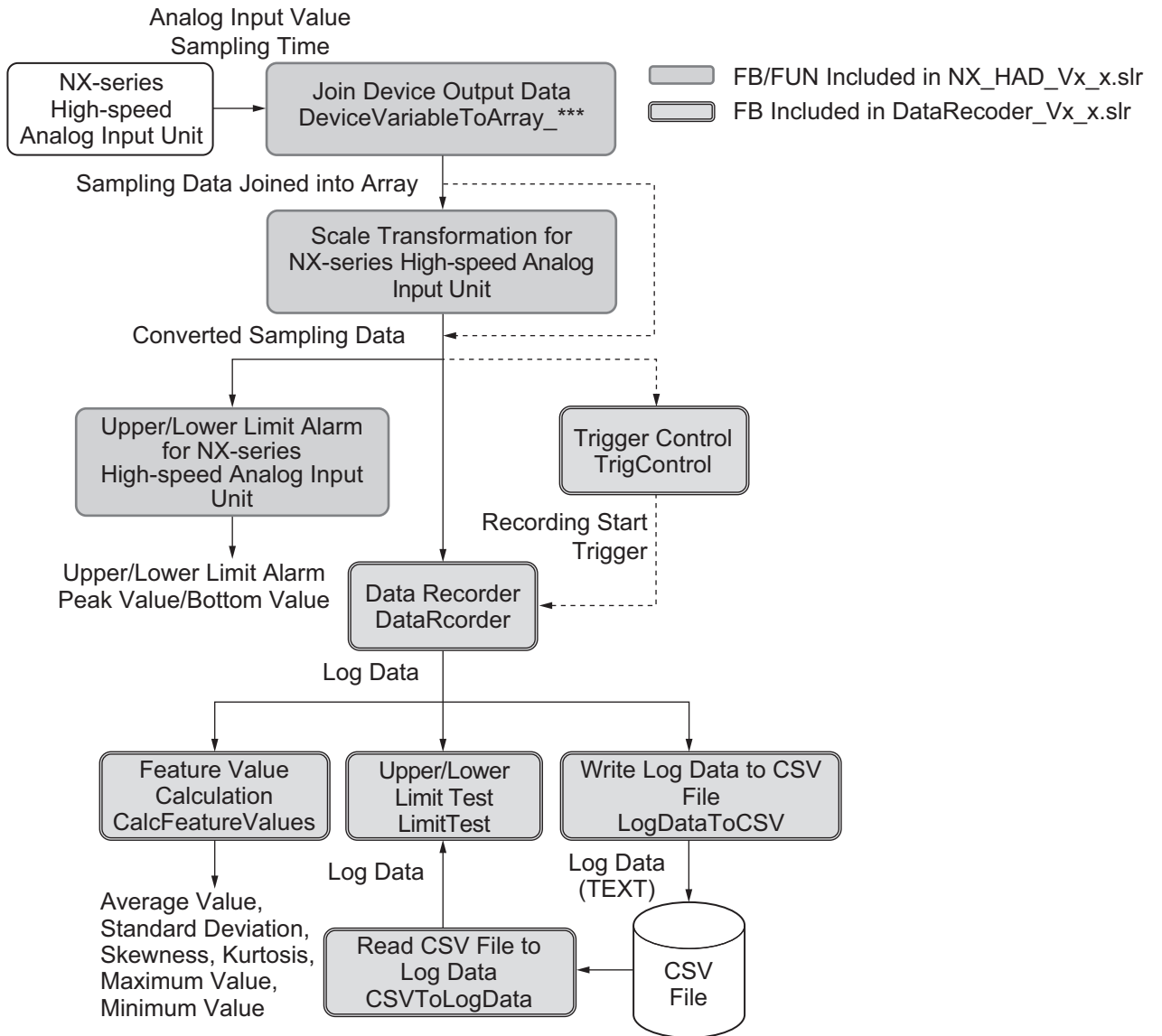
Library file name	FB/FUN	FB/FUN name
OmronLib_NX_HAD_Vx_x.slr <sup>*1</sup>	FUN	DeviceVariableToArray_***
	FUN	ScaleTrans_HAD
	FB	LimitAlarm_HAD
OmronLib_DataRecorder_Vx_x.slr <sup>*1</sup>	FB	TrigControl
	FB	DataRecorder
	FB	LimitTest
	FB	CalcFeatureValues
	FB	LogDataToCSV
	FB	CSVToLogData

\*1. Vx\_x shows the library file version.

### 2-1-3 Data Flow and FB/FUN Structure

This section provides the data flow and structure to use both library files.

- Use both NX\_HAD.slr and DataRecorder.slr files to log analog input values in chronological order, and perform calculations, tests, and file recording based on the log data.
- Use only the NX\_HAD.slr file to make alarm judgment for analog input values or detect the peak and bottom values in one task period.



# 3

## Common Specifications of Function Blocks

This section describes the shared specifications of each FB in the Sysmac Library.

---

<b>3-1</b>	<b>Common Variables .....</b>	<b>3 - 2</b>
3-1-1	Definition of Input Variables and Output Variables.....	3 - 2
3-1-2	Execute-type Function Blocks .....	3 - 3
3-1-3	Enable-type Function Blocks .....	3 - 5
<b>3-2</b>	<b>Precautions.....</b>	<b>3 - 7</b>
3-2-1	Nesting.....	3 - 7
3-2-2	Instruction Options.....	3 - 7
3-2-3	Re-execution of Function Blocks .....	3 - 7

## 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

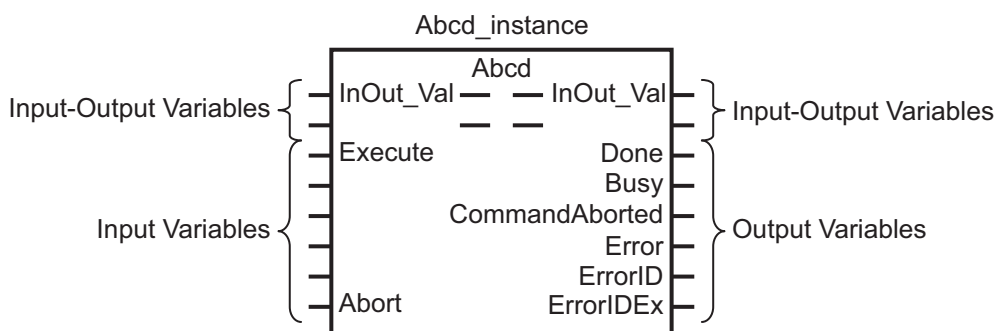
Common input variables and output variables used in functions and function blocks are as follows.

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.

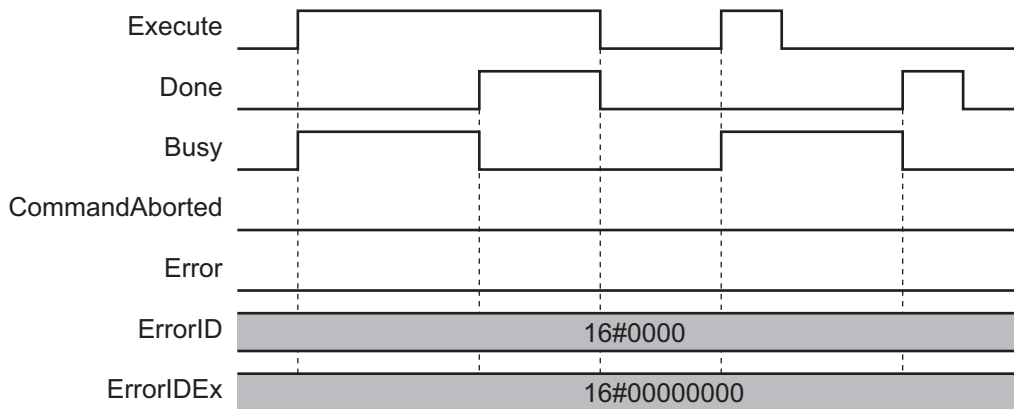


- If Execute is TRUE and Done, CommandAborted, or Error changes to TRUE, Done, CommandAborted, or Error changes to FALSE when Execute is changed to FALSE.
- If Execute is FALSE and Done, CommandAborted, or Error changes to TRUE, Done, CommandAborted, or Error changes to TRUE for only one task period.
- If an error occurs in the function block, 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.

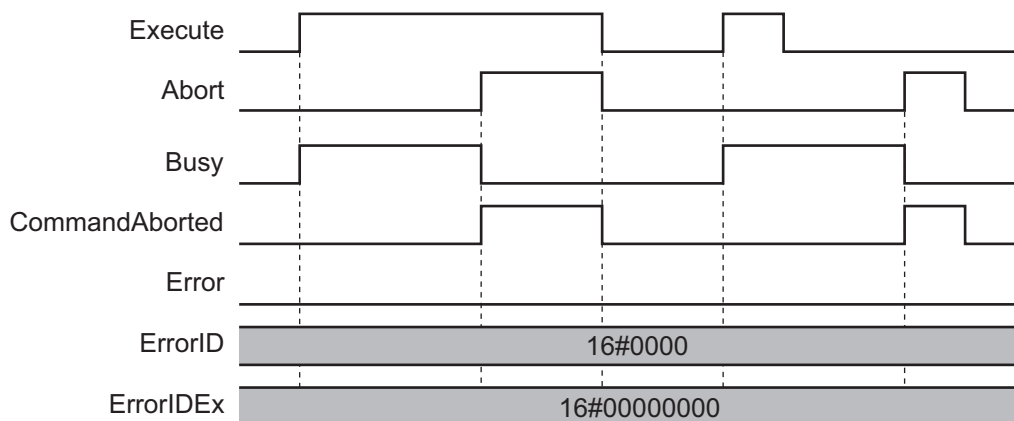
## Timing Chart

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

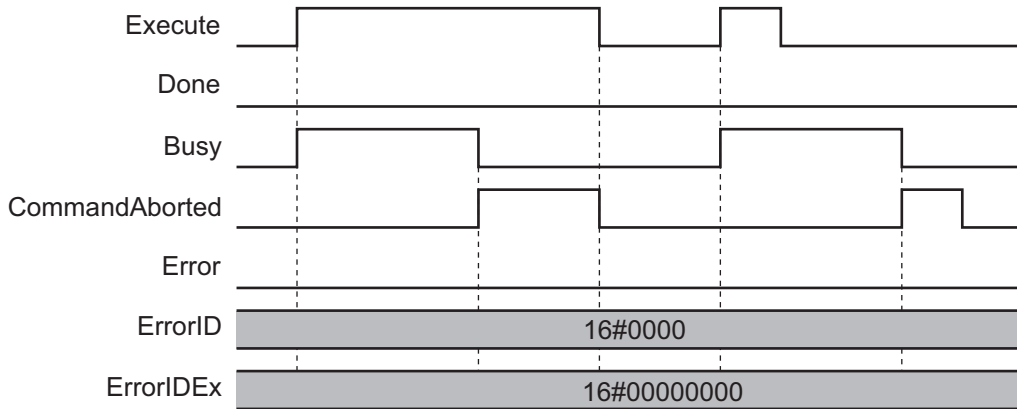
### ● Normal End



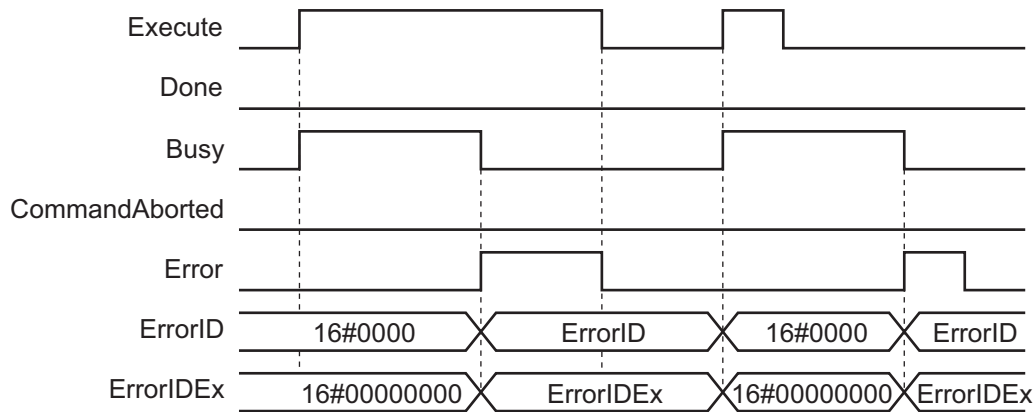
### ● Canceled Execution



● **Aborted Execution**

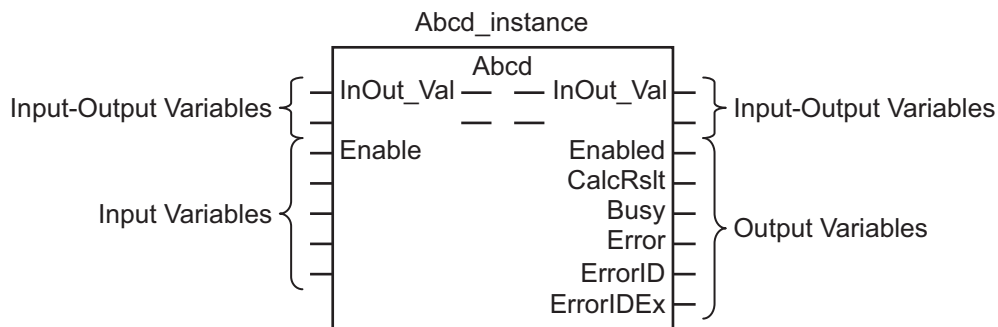


● **Errors**



**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.



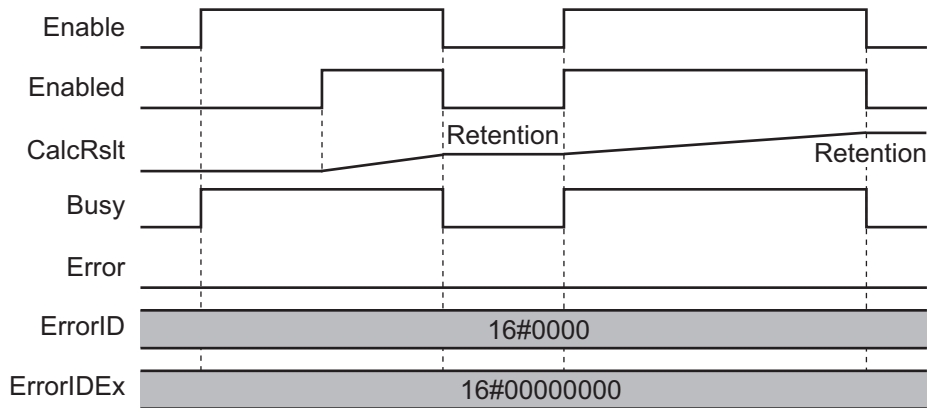
- If an error occurs in the function block, the relevant error code and expansion error code are set in ErrorID (Error Code) and ErrorIDEx (Expansion Error Code). The error codes are retained even after Error changes to FALSE, but ErrorID is set to 16#0000 and ErrorIDEx is set to 16#0000 0000 when Execute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

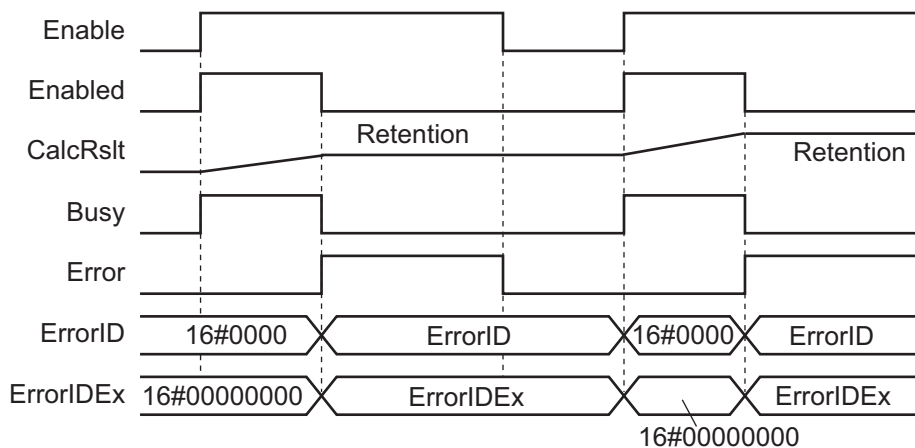
## Timing Charts

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

## FB/FUN Individual Specifications (NX\_HAD)

This section describes the FB/FUN individual specifications included in NX\_HAD.slr.

---

DeviceVariableToArray_*** .....	4 - 2
ScaleTrans_HAD.....	4 - 11
LimitAlarm_HAD .....	4 - 14

# DeviceVariableToArray\_\*\*\*

DeviceVariableToArray\_\*\*\* reads analog input values of one task period from the NX-series High-speed Analog Input Unit, and joins them into a single array variable.

The FUN name ends with "\_\*\*\*", which will be either 020 or 100 to show the maximum number of times of sampling input for this FUN. Select either of the following two FUNs based on the required number of samplings.

FB/FUN name	Name	FB/FUN	Graphic expression	ST expression
Device-Variable-ToArray_100	Device Output Data Binding	FUN		<pre>DeviceVariableToArray_100( Input01, Input02, Input03, Input04, Input05, Input06, Input07, Input08, Input09, Input10, StartPos, InputSize, NextPos, dataArray );</pre>
Device-Variable-ToArray_020				<pre>DeviceVariableToArray_020( Input01, Input02, StartPos, InputSize, NextPos, dataArray );</pre>

## Library Information

Item	Description
Library file name	OmronLib_NX_HAD_Vx_x.slr (x shows the version)
Namespace	OmronLib\NX_HAD
Function block and function number	DeviceVariableToArray_100: 00192 DeviceVariableToArray_020: 00191
Source code	Not Published

## Input Variables

	Meaning	Data type	Description	Valid range	Unit	Default
EN	Execute	BOOL	TRUE: Execute FALSE: Do not execute	TRUE, FALSE	---	FALSE
Input01[] - Input10[]	Device Output Data 01 to 10	AR- RAY[0..9] OF INT	Inputs the Analog Input Value acquired from the I/O data	---	---	---
StartPos	Start position	UINT	Specifies an array element number of <i>DataArray[]</i> to indicate where joining is started. Refer to <i>Function</i> on page 4 - 3 for details.	Depends on data type	---	0
InputSize	Number of data	UINT	Inputs the <i>Number of samplings</i> acquired from the I/O data	Depends on data type	---	1

## Output Variables

	Meaning	Data type	Description	Valid range	Unit	Default
ENO	Joining result	BOOL	TRUE: Normal end FALSE: Error end, or execution condition not met.	TRUE, FALSE	---	---
NextPos	Next position	UINT	Outputs the element number where the next joining will be started. Refer to <i>Function</i> on page 4 - 3 for details.	Depends on data type	---	---

## Input-Output Variables

	Meaning	Data type	Description	Valid range	Unit	Default
DataArray[] <sup>*1</sup>	Joined data	ARRAY[*] OF REAL	Stores joined input data	Depends on data type	---	---

\*1. The number of array elements is arbitrary. However, the number of array elements must be equal to or more than *InputSize*. In addition, subscripts of the array can start with 0 or any other number.

## Function

Since analog input values of one or more ARRAY[0..9] OF INT array types are obtained from the NX-series High-speed Analog Input Unit, DeviceVariableToArray\_\*\*\* joins the array values into one REAL type array.

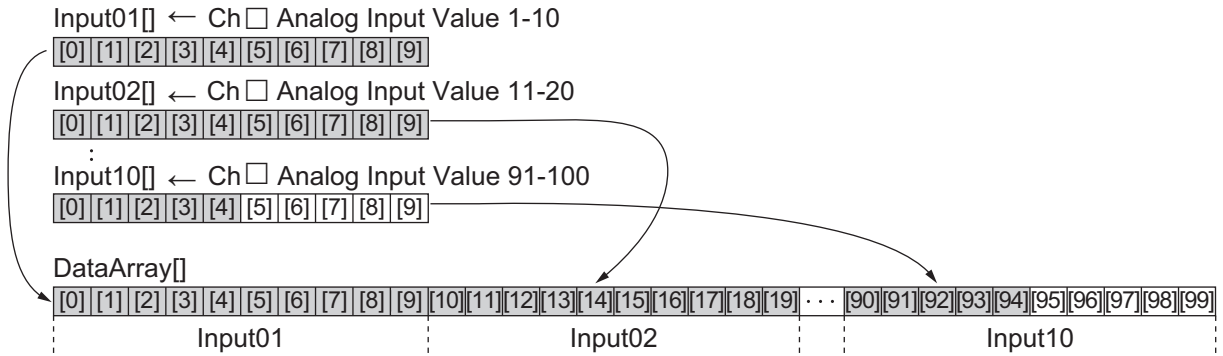
The data of the size specified by *InputSize* is joined in order from *Input01[0]*.

When the data is successfully joined, *ENO* changes to TRUE.

In the cases below, *ENO* changes to FALSE without joining array values.

- When *InputSize* is 0
- When the number of *DataArray[]* array elements is less than the *InputSize*
- When *StartPos* is outside the valid range of *DataArray[]*

Example: When the number of *DataArray*[] array elements is 100 and *InputSize* is 95, join the array data up to *Input10*[4].

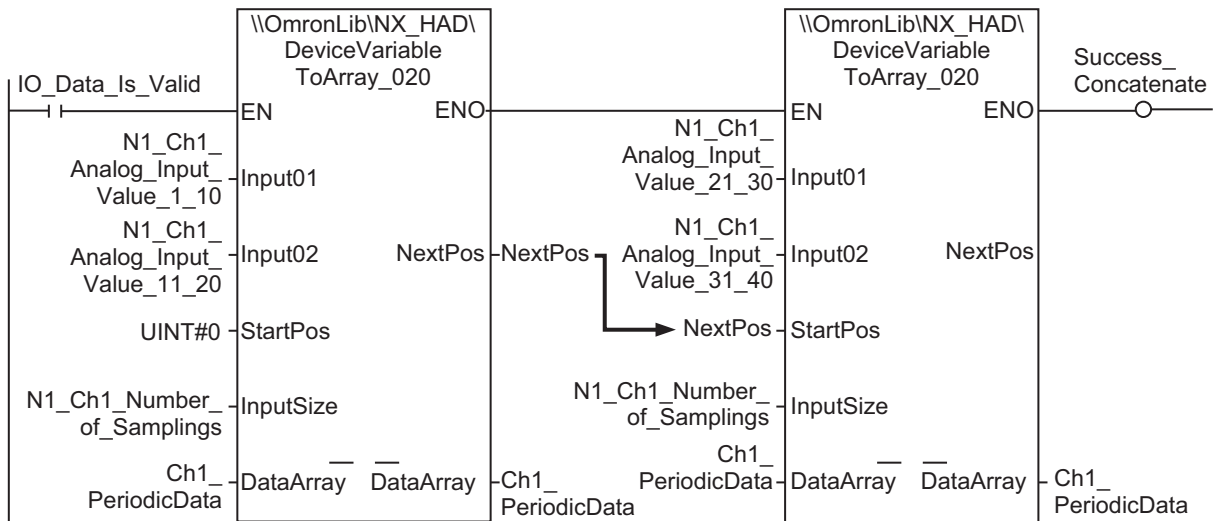


**• Execution in Series**

If this FUN is executed in series as shown below, the joined data can exceed the maximum number of input samplings for one FUN.

DeviceVariableToArray\_020 and DeviceVariableToArray\_100 can be used in combination and executed in series.

Example: The number of samplings is 40.



**Precautions for Correct Use**

For execution in series, observe the following.

- In all *InputSize*, input the number of samplings for Input Ch from the I/O data. In the above example, *N1\_Ch1\_Number\_of\_Sampling* is input.
- Specify the same variable in *DataArray*[] of the FUNs in series.
- For the number of *DataArray*[] array elements, specify a value equal to or more than *InputSize*.
- For *StartPos* of the first FUN, input 0.
- For *StartPos* of the second and subsequent FUNs, input the *NextPos* value of the preceding FUN.

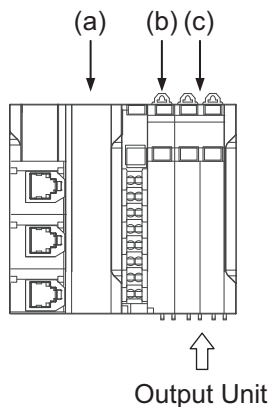
## Sample Programming 1

In this sample program, analog input values sampled 100 times per task period by the NX-series High-speed Analog Input Unit are joined into one array variable. This program also performs scale transformation of the joined data, and monitors the upper/lower limit values of the data.

This example assumes that the NX-series High-speed Analog Input Unit is connected to an NX-series CPU Unit.

## System Configuration

The system configuration is as shown below.



Let-ter	Description	Model	Description
(a)	NX-series CPU Unit	NX102-□□□□	---
(b)	I/O Power Additional Supply Unit	NX-PF0730	• NX Unit No.: 1
(c)	NX-series High-speed Analog Input Unit	NX-HAD401	• NX Unit No.: 2 • Channel Used: Ch1

## Unit Operation Settings

Set the NX-series High-speed Analog Input Unit as shown in the table below. Refer to the *NX-series Analog I/O Units User's Manual for High-speed Analog Input Units (Cat. No. W592)* for the setting method.

Setting item	Setting value	Meaning
Ch1 Enable/Disable	TRUE	Ch1 Enable
Ch1 Range Setting	0	-10 to +10 V
Ch1 Number of Samplings Setting	100	100 times

## Program

- External Variables

Name	Data type	Constant	Comment
NXBus_N2_NX_Unit_I_O_Data_Active_Status	BOOL		
NXBus_N2_NX_Unit_Error_Status	BOOL		
N2_Ch1_Number_of_Samplings	UINT		
N2_Ch1_Analog_Input_Value_1_10	ARRAY[0..9] OF INT		
N2_Ch1_Analog_Input_Value_11_20	ARRAY[0..9] OF INT		
N2_Ch1_Analog_Input_Value_21_30	ARRAY[0..9] OF INT		
N2_Ch1_Analog_Input_Value_31_40	ARRAY[0..9] OF INT		
N2_Ch1_Analog_Input_Value_41_50	ARRAY[0..9] OF INT		
N2_Ch1_Analog_Input_Value_51_60	ARRAY[0..9] OF INT		
N2_Ch1_Analog_Input_Value_61_70	ARRAY[0..9] OF INT		
N2_Ch1_Analog_Input_Value_71_80	ARRAY[0..9] OF INT		
N2_Ch1_Analog_Input_Value_81_90	ARRAY[0..9] OF INT		
N2_Ch1_Analog_Input_Value_91_100	ARRAY[0..9] OF INT		
Ch1_Input_Array100	ARRAY[0..99] OF REAL		Analog input value of one task period

- Internal Variables

Name	Data type	Default	AT	Retention	Constant	Comment
LimitAlarm_instance	OmronLib\NX_HAD\LimitAlarm_HAD					
IO_Data_Is_Valid	BOOL					Set to TRUE when input values from the NX-series High-speed Analog Input Unit are normal.
Success_Concatenate	BOOL					
Alarm	BOOL					Set to TRUE when any of <i>QHH</i> , <i>QH</i> , <i>QL</i> , or <i>QLL</i> is TRUE
QHH	BOOL					
QH	BOOL					
QL	BOOL					
QLL	BOOL					
PeakVal	REAL					
BottomBal	REAL					
Clear_PkBtm	BOOL	FALSE				
EN_P1	BOOL	FALSE				While this variable is TRUE, scale transformation and alarm judgment are executed

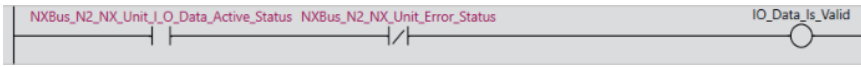
- Task Settings

Locate in the primary periodic task.

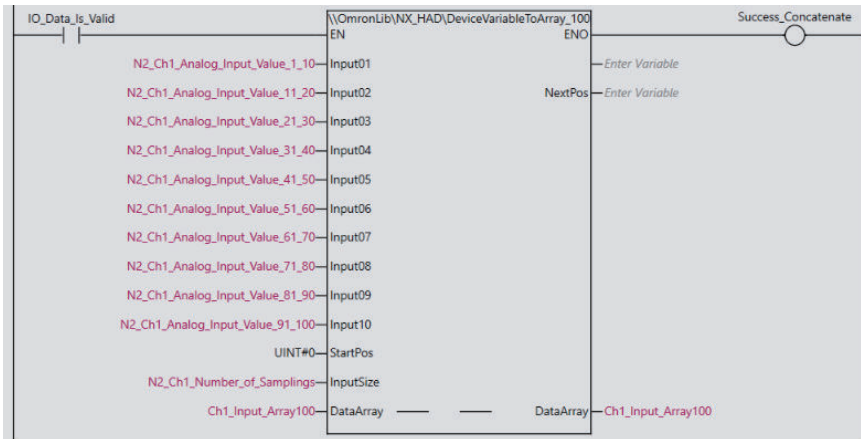


• LD Program

(1) Check that normal data is being received from the NX-series High-speed Analog Input Unit.

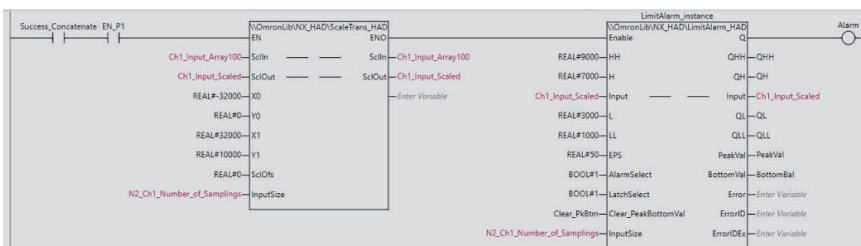


(2) Join analog input values of one task period into a single variable, *Ch1\_Input\_Array100[]*.



(3) After the scale transformation of the analog input values of the task period, change *Alarm* to TRUE if the input value is outside the valid ranges specified for *HH*, *H*, *L*, and *LL*.

Scale transformation setting	From "-32,000 to 32,000" to "0 to 10,000"	
Alarm settings	Top Upper Limit <i>HH</i> Alarm	More than 9,000
	Upper Limit <i>H</i> Alarm	More than 7,000
	Lower Limit <i>L</i> Alarm	Less than 3,000
	Bottom Lower Limit <i>LL</i> Alarm	Less than 1,000



DeviceVariableToArray\_\*\*\*

4

Sample Programming 2

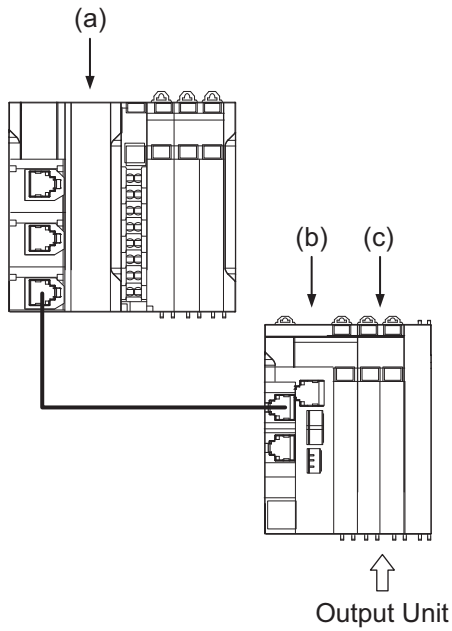
## Sample Programming 2

In this sample program, analog input values sampled 100 times per task period by the NX-series High-speed Analog Input Unit are joined into one array variable. This program also performs scale transformation of joined data, and monitors the upper/lower limit values of the data.

This example assumes that the NX-series High-speed Analog Input Unit is connected to an EtherCAT Slave Terminal.

## System Configuration

The system configuration is as shown below.



Letter	Description	Model	Description
(a)	NX-series CPU Unit	NX102-□□□□	---
(b)	EtherCAT Coupler Unit	NX-ECC203	<ul style="list-style-type: none"> <li>Node Address: 1</li> </ul>
(c)	NX-series High-speed Analog Input Unit	NX-HAD401	<ul style="list-style-type: none"> <li>NX Unit No.: 1</li> <li>Channel Used: Ch1</li> </ul>

## Unit Operation Settings

Set the NX-series High-speed Analog Input Unit as shown in the table below. Refer to the *NX-series Analog I/O Units User's Manual for High-speed Analog Input Units (Cat. No. W592)* for the setting method.

Setting item	Setting value	Meaning
Ch1 Enable/Disable	TRUE	Ch1 Enable
Ch1 Range Setting	0	-10 to +10 V
Ch1 Number of Samplings Setting	100	100 times

## Program

- External Variables

Name	Data type	Constant	Comment
_EC_PDslavTbl	ARRAY[1..192] OF BOOL	√	
_EC_CommErrTbl	ARRAY[1..192] OF BOOL	√	
_EC_SlavErrTbl	ARRAY[1..192] OF WORD	√	
_EC_InDataInvalid	BOOL	√	
N1_Ch1_Number_of_Samplings	UINT		
N1_Ch1_Analog_Input_Value_1_10	ARRAY[0..9] OF INT		
N1_Ch1_Analog_Input_Value_11_20	ARRAY[0..9] OF INT		

Name	Data type	Constant	Comment
N1_Ch1_Analog_Input_Value_21_30	ARRAY[0..9] OF INT		
N1_Ch1_Analog_Input_Value_31_40	ARRAY[0..9] OF INT		
N1_Ch1_Analog_Input_Value_41_50	ARRAY[0..9] OF INT		
N1_Ch1_Analog_Input_Value_51_60	ARRAY[0..9] OF INT		
N1_Ch1_Analog_Input_Value_61_70	ARRAY[0..9] OF INT		
N1_Ch1_Analog_Input_Value_71_80	ARRAY[0..9] OF INT		
N1_Ch1_Analog_Input_Value_81_90	ARRAY[0..9] OF INT		
N1_Ch1_Analog_Input_Value_91_100	ARRAY[0..9] OF INT		
Ch1_Input_Array100	ARRAY[0..99] OF REAL		Analog input value of one task period

• Internal Variables

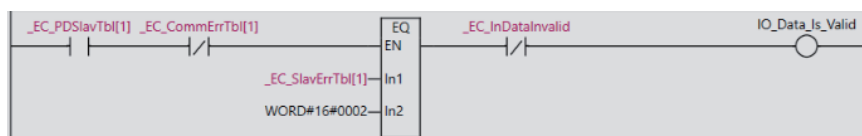
Name	Data type	Default	AT	Retention	Constant	Comment
LimitAlarm_instance	OmronLib\NX_HAD\LimitAlarm_HAD					
IO_Data_Is_Valid	BOOL					Set to TRUE when input values from the NX-series High-speed Analog Input Unit are normal.
Success_Concatenate	BOOL					
Alarm	BOOL					Set to TRUE when any of QHH, QH, QL, or QLL is TRUE
QHH	BOOL					
QH	BOOL					
QL	BOOL					
QLL	BOOL					
PeakVal	REAL					
BottomBal	REAL					
Clear_PkBtm	BOOL	FALSE				
EN_P1	BOOL	FALSE				While this variable is TRUE, scale transformation and alarm judgment are executed

• Task Settings

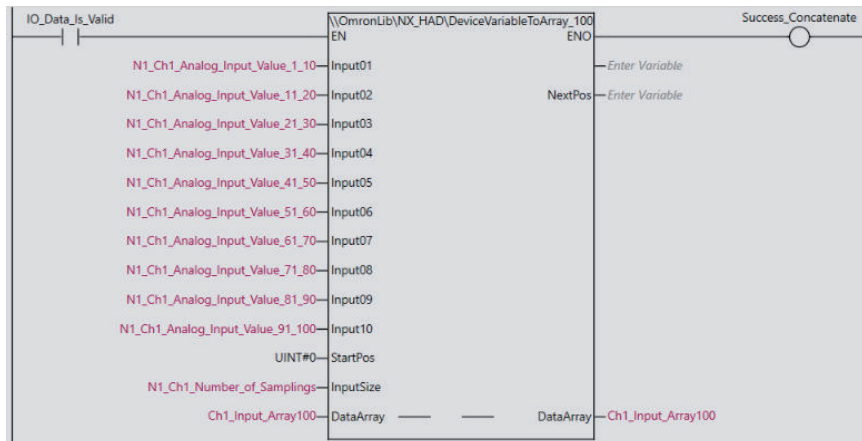
Locate in the primary periodic task.

• LD Program

(1) Check that normal data is being received from the NX-series High-speed Analog Input Unit.

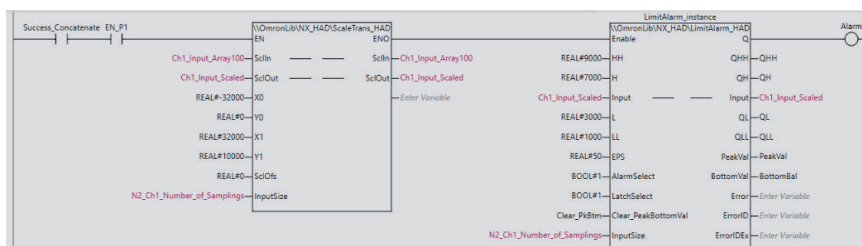


(2) Join analog input values of one task period into a single variable, *Ch1\_Input\_Array100*].



(3) After the scale transformation of the analog input values of the task period, change *Alarm* to TRUE if the input value is outside the valid ranges specified for *HH*, *H*, *L*, and *LL*.

Scale transformation setting		From "-32,000 to 32,000" to "0 to 10,000"
Alarm settings	Top Upper Limit <i>HH</i> Alarm	More than 9,000
	Upper Limit <i>H</i> Alarm	More than 7,000
	Lower Limit <i>L</i> Alarm	Less than 3,000
	Bottom Lower Limit <i>LL</i> Alarm	Less than 1,000



### Precautions for Correct Use

- The sample programming shows only the portion of a program that uses the function or function block from the library. When programming actual applications, also program safety circuits, device interlocks, I/O with other devices, and other control procedures.
- Create a user program that will produce the intended device operation.
- Check the user program for proper execution before you use it for actual operation.

# ScaleTrans\_HAD

ScaleTrans\_HAD performs scale transformation of data from the NX-series High-speed Analog Input Unit.

FB/FUN name	Name	FB/FUN	Graphic expression	ST expression
Scale-Trans_HAD	Scale transformation for NX-series High-speed Analog Input Unit	FUN	<pre> \OmronLib\NX_HAD\ScaleTrans_HAD - EN ----- ENO - ScIn ----- ScIn - ScOut ----- ScOut - X0 - Y0 - X1 - Y1 - ScOfs - InputSize </pre>	ScaleTrans_HAD( ScIn, ScOut, X0, Y0, X1, Y1, ScaleOfs, InputSize );

## Library Information

Item	Description
Library file name	OmronLib_NX_HAD_Vx_x.slr (x shows the version)
Namespace	OmronLib\NX_HAD
Function block and function number	00193
Source code	Not Published

## Input Variables

	Meaning	Data type	Description	Valid range	Unit	Default
EN	Execute	BOOL	TRUE: Execute FALSE: Do not execute	TRUE, FALSE	---	FALSE
X0	Input Range Lower Limit Value	REAL	Lower limit value of input range	Depends on data type	---	-32000
Y0	Output Range Lower Limit Value	REAL	Lower limit value of output range	Depends on data type	---	-32000
X1	Input Range Upper Limit Value	REAL	Upper limit value of input range	Depends on data type	---	32000

	Meaning	Data type	Description	Valid range	Unit	Default
Y1	Output Range Upper Limit Value	REAL	Upper limit value of output range	Depends on data type	---	32000
ScIOfs	Offset Value	REAL	Offset value exceeding the output value	Depends on data type	---	0.0
InputSize	Number of data	UINT	Inputs the <i>Number of samplings</i> acquired from the I/O data	1 to 1000	---	1

## Output Variables

	Meaning	Data type	Description	Valid range	Unit	Default
ENO	Transformation Results	BOOL	TRUE: Normal end FALSE: Error end, or execution condition not met	TRUE, FALSE	---	---

## Input-Output Variables

	Meaning	Data type	Description	Valid range	Unit	Default
ScIIn[]*1	Input Value	ARRAY[*] OF REAL	Data array subject to scale transformation	Depends on data type	---	---
ScIOut[]*2	Output Value	ARRAY[*] OF REAL	Data array after scale transformation	Depends on data type	---	---

\*1. The number of array elements is arbitrary. However, the number of array elements must be equal or more than *InputSize*. In addition, subscripts of the array can start with 0 or any other number.

\*2. The number of array elements is arbitrary. However, the number of array elements must be the same as that of *ScIIn[]*. In addition, subscripts of the array can start with 0 or any other number. Even if the first subscript in the array is different from that of *ScIIn[]*, scale transformation can be executed.

## Function

ScaleTrans\_HAD transforms each element of *ScIIn[]* from the input range specified with *X0* and *X1* to the output range specified from *Y0* to *Y1*.

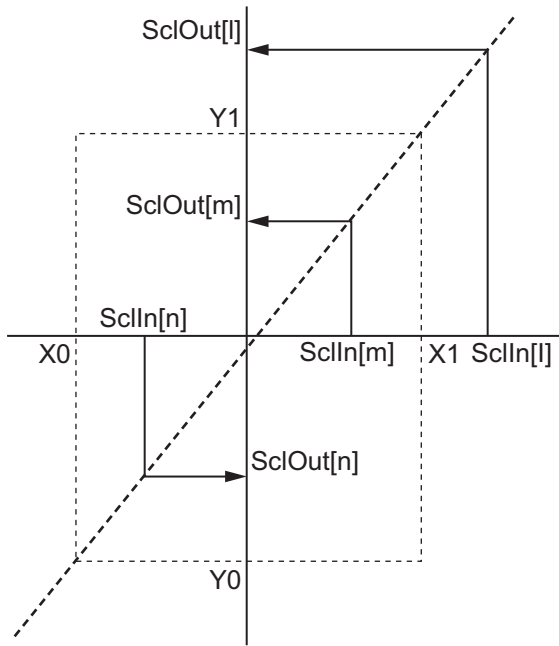
It also stores the sum of the transformed values and *ScIOfs* in *ScIOut[]*.

You can perform the transformation even if the input values are outside the input range.

When the transformation is successful, *ENO* changes to TRUE.

In the cases below, *ENO* changes to FALSE without performing scale transformation.

- When the number of *ScIIn[]* and *ScIOut[]* array elements each is less than *InputSize*
- When the number of elements in *ScIIn[]* and that of *ScIOut[]* do not match.  
However, the first subscript of each can be different.
- When *X0* and *X1* are equal, or when *Y0* and *Y1* are equal
- When *InputSize* is 0, or equal to or more than 1001



$$\text{SclOut}[n] = \frac{Y1-Y0}{X1-X0} X (\text{SclIn}[n]-X0) + Y0 + \text{SclOfs}$$

The range of  $n$  is from the first subscript of  $\text{SclIn}[]$  to the sum of the first subscript and  $\text{InputSize} - 1$ .

## Sample Programming

Refer to *Sample Programming 1* on page 4 - 5 and *Sample Programming 2* on page 4 - 7.

# LimitAlarm\_HAD

LimitAlarm\_HAD monitors input data from the NX-series High-speed Analog Input Unit and issues alarms in terms of the top upper limit, upper limit, lower limit, and bottom lower limit.

FB/FUN name	Name	FB/FUN	Graphic expression	ST expression
LimitAlarm_HAD	Upper/lower alarm for NX-series High-speed Analog Input Unit	FB		<pre>LimitAlarm_HAD_instance( Enable, HH, H, Input, L, LL, EPS, AlarmSelect, LatchSelect, Clear_PeakBottomVal, InputSize, Q, QHH, QH, QL, QLL, PeakVal, BottomVal, Error, ErrorID, ErrorIDEx );</pre>

## Library Information

Item	Description
Library file name	OmronLib_NX_HAD_Vx_x.slr (x shows the version)
Namespace	OmronLib\NX_HAD
Function block and function number	00194
Source code	Not Published

## Input Variables

	Meaning	Data type	Description	Valid range	Unit	Default
Enable	Execute	BOOL	TRUE: Execute FALSE: Do not execute	TRUE, FALSE	---	FALSE
HH	Top Upper Limit Setting Value	REAL	Top Upper Limit Setting Value of Input Value	Depends on data type	---	0.0



	Meaning	Data type	Description	Valid range	Unit	Default
H	Upper Limit Setting Value	REAL	Upper Limit Setting Value of Input Value	Depends on data type	---	0.0
L	Lower Limit Setting Value	REAL	Lower Limit Setting Value of Input Value	Depends on data type	---	0.0
LL	Bottom Lower Limit Setting Value	REAL	Bottom Lower Limit Setting Value of Input Value	Depends on data type	---	0.0
EPS	Hysteresis	REAL	Alarm Hysteresis	0 and greater	---	0.0
AlarmSelect	Alarm Selection	BOOL	TRUE: HH, H, L, LL Alarm FALSE: H, L Alarm	TRUE, FALSE	---	FALSE
LatchSelect	Latch Selection	BOOL	TRUE: Latch the alarm FALSE: Do not latch the alarm	TRUE, FALSE	---	FALSE
Clear_PeakBottomVal	Peak Value, Bottom Value Cleared	BOOL	TRUE: Clears the peak value/bottom value	TRUE, FALSE	---	FALSE
InputSize	Number of data	UINT	Inputs the <i>Number of samplings</i> acquired from the I/O data	1 to 1000	---	1

## Output Variables

	Meaning	Data type	Description	Valid range	Unit	Default
Q	Alarm Output	BOOL	TRUE: Any of QHH, QH, QL, or QLL is TRUE FALSE: All of QHH, QH, QL, and QLL are FALSE	TRUE, FALSE	---	---
QHH	Top Upper Limit Alarm	BOOL	TRUE: Top Upper Limit Alarm ON FALSE: Top Upper Limit Alarm OFF	TRUE, FALSE	---	---
QH	Upper Limit Alarm	BOOL	TRUE: Upper Limit Alarm ON FALSE: Upper Limit Alarm OFF	TRUE, FALSE	---	---
QL	Lower Limit Alarm	BOOL	TRUE: Lower Limit Alarm ON FALSE: Lower Limit Alarm OFF	TRUE, FALSE	---	---
QLL	Bottom Lower Limit Alarm	BOOL	TRUE: Bottom Lower Limit Alarm ON FALSE: Bottom Lower Limit Alarm OFF	TRUE, FALSE	---	---
PeakVal	Peak Value	REAL	Maximum Input Value	Depends on data type	---	---

	Meaning	Data type	Description	Valid range	Unit	Default
BottomVal	Bottom Value	REAL	Minimum Input Value	Depends on data type	---	---
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 - 19 for details.

## Input-Output Variables

	Meaning	Data type	Description	Valid range	Unit	Default
Input[]*1	Input Value	ARRAY[*] OF REAL	Data array to be monitored	Depends on data type	---	---

\*1. The number of array elements is arbitrary. However, the number of array elements must be equal to or more than *InputSize*. In addition, subscripts of the array can start with 0 or any other number.

## Function

While *Enable* is TRUE, an alarm is output if any of the element values in *Input[]* exceeds the setting value of *HH* or *H*, or falls below that of *L* or *LL*.

The relationship of input, setting values, and alarm output is shown below.

- When *Input[]* is larger than *HH*, set *QHH* to TRUE.  
When *QHH* is TRUE, and the input value becomes smaller than the result of  $HH - EPS$ , set *QHH* to FALSE.
- When *Input[]* is larger than *H*, set *QH* to TRUE.  
When *QH* is TRUE, and the input value becomes smaller than the result of  $H - EPS$ , set *QH* to FALSE.
- When *Input[]* is smaller than *L*, set *QL* to TRUE.  
When *QL* is TRUE, and the input value becomes larger than the result of  $L + EPS$ , set *QL* to FALSE.
- When *Input[]* is smaller than *LL*, set *QLL* to TRUE.  
When *QLL* is TRUE, and the input value becomes larger than the result of  $LL + EPS$ , set *QLL* to FALSE.
- When any of *QHH*, *QH*, *QL*, or *QLL* is TRUE, set *Q* to TRUE.

If *AlarmSelect* is TRUE, make alarm judgment in terms of all the *HH*, *H*, *L*, and *LL*.

If *AlarmSelect* is FALSE, make alarm judgment in terms of *H* and *L*. In this case, *QHH* and *QLL* are always FALSE.

Alarm judgment is made on input values for the number of elements specified by *InputSize*. As a result, more than one of *QHH*, *QH*, *QL*, and *QLL* may become TRUE at the same time. Refer to below for details.

In the cases below, *Error* changes to TRUE and terminates the monitoring.

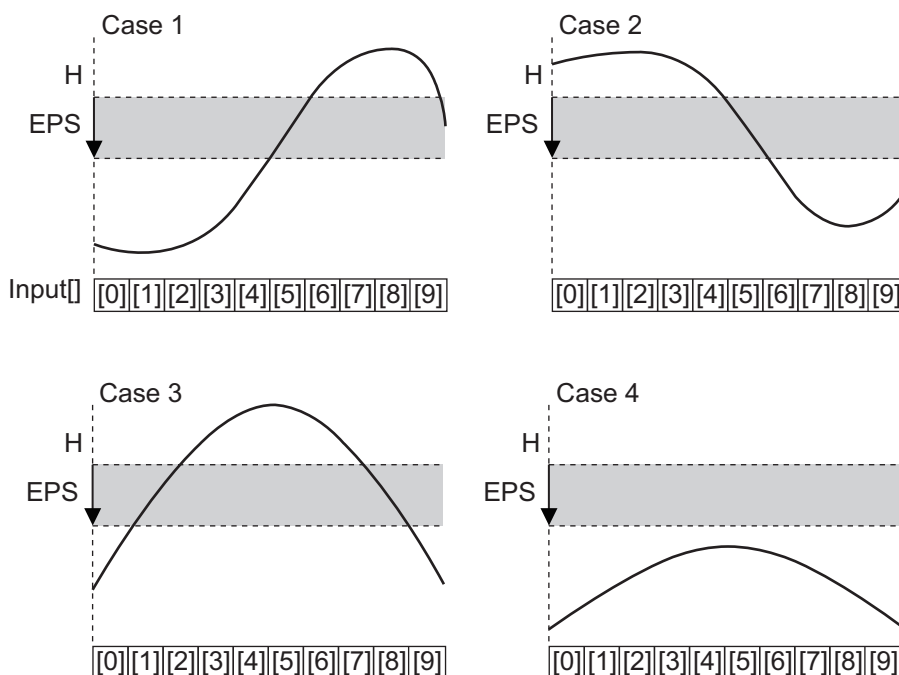
- When the number of *Input[]* array elements is less than *InputSize*
- When each alarm setting value is NOT specified as below.  
 $LL \leq L \leq H \leq HH$
- When *EPS* is non-numeric or  $\pm\infty$
- When *InputSize* is 0 or equal to or more than 1001

The maximum input value and the minimum input value during the execution in progress, which is from the start of the execution up to the present, are output to *PeakVal* and *BottomVal*, respectively. When *Clear\_PeakBottomVal* is TRUE, *PeakVal* and *BottomVal* change to 0. If *Clear\_PeakBottomVal* changes to FALSE, the maximum and minimum values will be output based on the subsequent input values.

If *Enable* changes to FALSE, this FUN execution ends.

- *Q*, *QHH*, *QH*, *QL*, and *QLL* all change to FALSE.
- The values of *PeakVal* and *BottomVal* are retained until *Enable* changes to TRUE again.
- Alarm Output/Reset  
The alarm output/reset operation is determined by *LatchSelect*.
  - a) When *LatchSelect* is FALSE  
Perform alarm judgment operation for the last element of *Input[]*, whose number of elements is specified by *InputSize*.
  - b) When *LatchSelect* is TRUE  
In *Input[]*, set alarm judgment for all elements specified in *InputSize*.

The following examples show how *QH* operates when changes in the input of the NX-series High-speed Analog Input Unit in one task period are stored in *Input[]*.



LatchSelect	QH			
	Case 1	Case 2	Case 3	Case 4
FALSE	TRUE	FALSE	FALSE	FALSE
TRUE	TRUE	TRUE	TRUE	FALSE

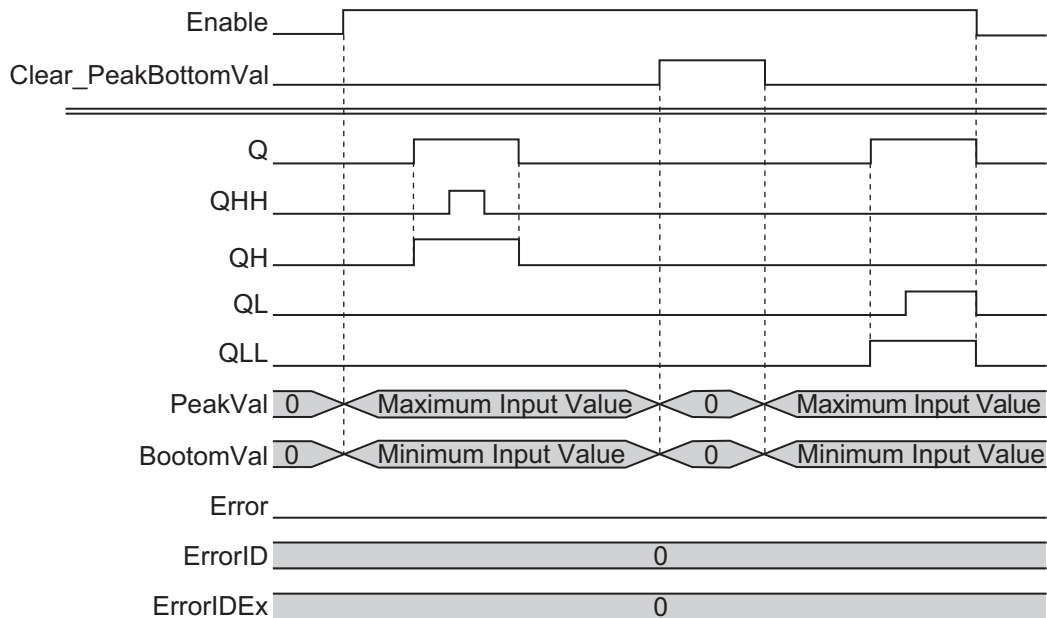
- Setting changes during execution

Changes in the values of *HH*, *H*, *L*, *LL*, and *EPS* are always reflected while execution is in progress. Any changes in the values of *AlarmSelect* and *LatchSelect* will not be reflected during the execution. If those variables are changed during the execution, the Alarm Output will change to FALSE and perform alarm judgment operation based on the setting values after the change.

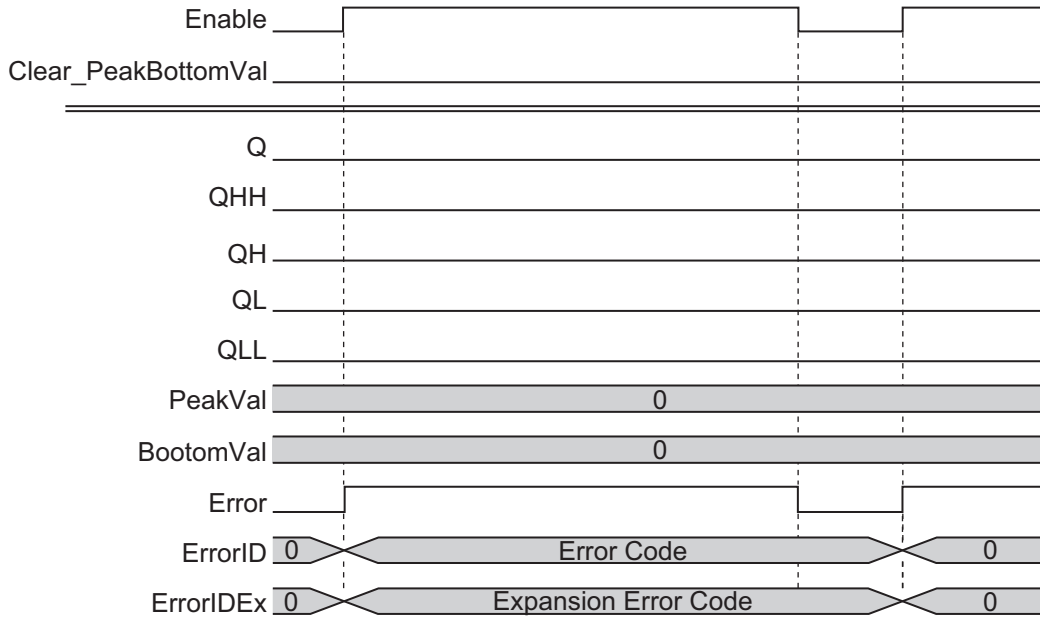
## Timing Chart

The timing charts are shown below.

- If *Enable* is set to TRUE, the alarm monitoring starts.  
*PeakVal* and *BottomVal* are set to the maximum input value and the minimum input value, respectively, in the period from the start of execution up to the present.
- If *Enable* is set to FALSE, the alarm monitoring ends.
- While *Clear\_PeakBottomVal* is TRUE, 0 is output to *PeakVal* and *BottomVal*.
- If an error occurs, *Error* changes to TRUE. At the same time, *ErrorID* and *ErrorIDEx* are output.
- When an error occurs, the value of *Error* is retained while *Enable* is TRUE.  
 After *Enable* changes to FALSE, *ErrorID* and *ErrorIDEx* are retained until *Enable* changes to TRUE again.
- Timing Chart for Normal End



- Timing Chart for Error End



### Troubleshooting

Error code	Expansion error code	Status	Description	Corrective action
16#0000	16#00000000	Normal End	---	---
16#3D00	16#00000001	Illegal Data Size	When the number of <i>Input[]</i> array elements is less than <i>InputSize</i>	Check and correct the number of <i>Input[]</i> array elements if necessary.
	16#00000002	Alarm Setting Illegal Size Relationship	The <i>HH</i> , <i>H</i> , <i>L</i> , and <i>LL</i> size relationships are incorrect	Set the alarm setting values as follows: $LL \leq L \leq H \leq HH$
	16#00000003	Illegal EPS	The <i>EPS</i> value is outside the valid range	Correct the <i>EPS</i> value so that it is within the valid range.
	16#00000004	Illegal InputSize	The <i>InputSize</i> value is outside the valid range	Correct the <i>InputSize</i> value so that it is within the valid range.

### Sample Programming

Refer to *Sample Programming 1* on page 4 - 5 and *Sample Programming 2* on page 4 - 7.



# 5

## FB/FUN Individual Specifications (DataRecorder)

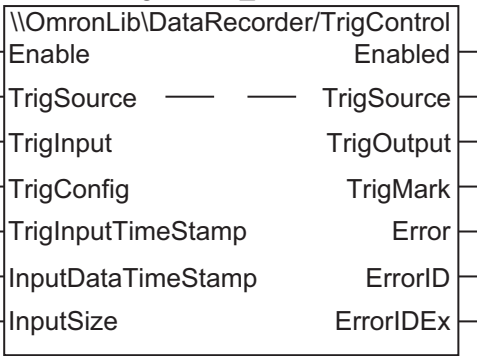
This section describes the FB/FUN individual specifications included in DataRecorder.sr.

---

TrigControl .....	5 - 2
DataRecorder .....	5 - 12
LimitTest.....	5 - 19
CalcFeatureValues .....	5 - 25
LogDataToCSV.....	5 - 34
CSVToLogData.....	5 - 41

# TrigControl

TrigControl generates trigger information, which allows DataRecorder start data logging.

FB/FUN name	Name	FB/FUN	Graphic expression	ST expression
TrigControl	Trigger Control	FB		<pre>TrigControl_instance(   Enable,   TrigSource,   TrigInput,   TrigConfig,   TrigInputTimeS-   tamp,   InputDataTimeS-   tamp,   InputSize,   Enabled,   TrigOutput,   TrigMark,   Error,   ErrorID,   ErrorIDEx );</pre>

## Library Information

Item	Description
Library file name	OmronLib_DataRecorder_Vx_x.slr (x shows the version)
Namespace	OmronLib\DataRecorder
Function block and function number	00195
Source code	Not Published

## Input Variables

	Meaning	Data type	Description	Valid range	Unit	Default
Enable	Execute	BOOL	TRUE: Execute FALSE: Do not execute	TRUE, FALSE	---	---
TrigInput	Trigger input	BOOL	Inputs the digital signal for detecting the trigger. This input is valid when the trigger mode is <i>Digital Input Trigger Mode</i> or <i>Time Stamp Trigger Mode</i> .	TRUE, FALSE	---	---
TrigConfig	Trigger setting	OmronLib \DataRe- corder \sTrigConfig	Sets the trigger conditions	---	---	---



	Meaning	Data type	Description	Valid range	Unit	Default
TrigInputTimeStamp	Trigger Input Time Stamp	ULINT	Inputs the <i>Trigger Input Time Stamp</i> acquired from the I/O data. This input is valid when the trigger mode is <i>Time Stamp Trigger Mode</i> .	Depends on data type	nsec	0
InputDataTimeStamp	Analog Input Time Stamp	ULINT	Inputs the <i>Analog Input Time Stamp</i> acquired from the I/O data	Depends on data type	nsec	0
InputSize	Number of data	UINT	Inputs the <i>Number of samplings</i> acquired from the I/O data	1 to 1000	---	1

## Output Variables

	Meaning	Data type	Description	Valid range	Unit	Default
Enabled	Executing	BOOL	Set to TRUE during execution	TRUE, FALSE	---	---
TrigOutput	Trigger Output	BOOL	When the trigger conditions are met, output TRUE	TRUE, FALSE	---	---
TrigMark	Trigger Information	OmronLib \DataRecorder \sTrigMark	Output the <i>TrigSource[]</i> element numbers and array element values when the trigger conditions are met	---	---	---
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 - 7 for details.

## Input-Output Variables

	Meaning	Data type	Description	Valid range	Unit	Default
Trig-Source[]*1	Input Value for Trigger Judgment	ARRAY[*] OF REAL	Input the data array which is used to check if trigger conditions are met.	---	---	---

\*1. The number of array elements is arbitrary. However, the number of array elements must be equal to or more than *InputSize*. In addition, subscripts of the array can start with 0 or any other number.

## Structure

- OmronLib\DataRecorder'sTrigConfig

Member Name	Data type	Description
TrigMode	USINT	Sets the trigger mode. Refer to <i>Function</i> on page 5 - 4 for details. <ul style="list-style-type: none"> <li>• Analog Level Trigger Mode Uses the <i>TrigSource[]</i> value and value changes for trigger checking.</li> <li>• Digital Input Trigger Mode Detects the trigger with a rise in trigger input.</li> <li>• Time Stamp Trigger Mode Detects the trigger with a rise in trigger input. In this mode, you can get more accurate time when the trigger conditions are met, than in Digital Input Trigger mode. The trigger input employs the input of the NX-series High-speed Analog Input Unit, and the input refreshing with input changed time of NX-series Units.</li> </ul>
TrigSlope	BOOL	Specifies when to detect a trigger, with a change in the <i>TrigSource[]</i> value. Valid for <i>Analog Level Trigger Mode</i> . FALSE: Rising TRUE: Falling
AnalogTrigLevel	REAL	Specifies the input threshold value for detecting the trigger. Valid for <i>Analog Level Trigger Mode</i> .

- OmronLib\DataRecorder'sTrigMark

Member Name	Data type	Description
Pos	UINT	Outputs the <i>TrigSource[]</i> element number when the trigger conditions are met
Data	REAL	Outputs the data stored in <i>TrigSource[]</i> when the trigger conditions are met

## Function

While *Enable* is TRUE, TrigControl determines whether the trigger conditions specified in *TrigConfig* are satisfied.

When the trigger conditions are met, *TrigOutput* changes to TRUE, and the element number and value of the *TrigSource[]* element that satisfies the trigger conditions are output to *TrigMark*.

Once the trigger conditions are met, the values of *TrigOutput* and *TrigMark* are retained while *Enable* is TRUE.

For a re-execution, *Enable* should stay FALSE for one or more task period.

When *Enable* changes to FALSE, the trigger checking ends, and *Enabled* and *TrigOutput* change to FALSE.

In addition, 0 is output to *TrigMark.Pos* and *TrigMark.Data*.

In the cases below, *Error* changes to TRUE, and the trigger checking ends.

- When the number of *TrigSource[]* elements is less than the *InputSize*
- When *TrigConfig.TrigMode* is a number other than 0, 1, or 2
- When *InputSize* is 0, or equal to or more than 1001
- When the function block was executed in an event task

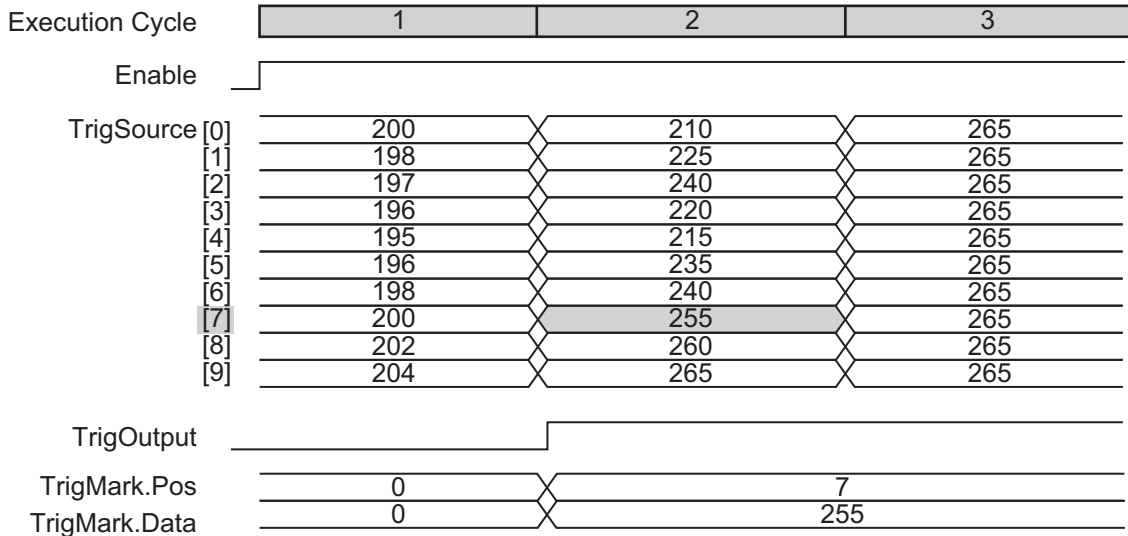
The trigger mode setting and action are as follows.

- Analog Level Trigger Mode: *TrigConfig.TrigMode* is 0

When *TrigConfig.TrigMode* is 0, *TrigConfig.TrigSlope* and *TrigConfig.AnalogTrigLevel* are used for the trigger conditions.

When each *TrigSource[]* element specified in *InputSize* crosses the *TrigConfig.AnalogTrigLevel*, the trigger conditions are met. At this time, *TrigOutput* changes to TRUE, and the numbers and values of *TrigSource[]* elements that meet the trigger conditions are output to *TrigMark*.

Example: When *TrigConfig.TrigMode* is 0, *AnalogInputLevel* is 250, *TrigConfig.Slope* is FALSE (rising), and *TrigSource[0..9]* and *InputSize* are 10

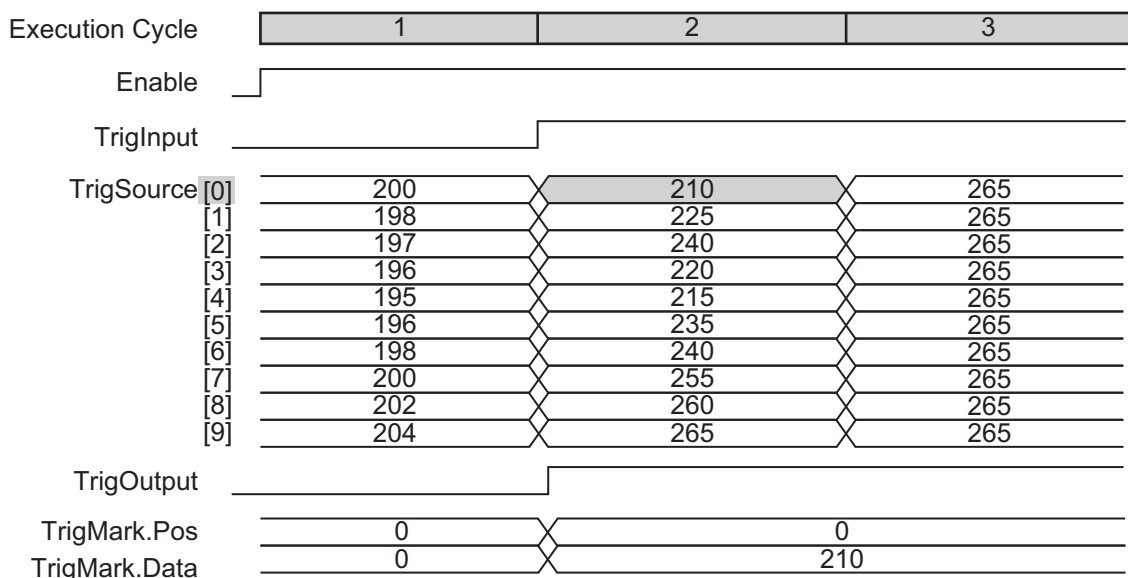


- Digital Input Trigger Mode: *TrigConfig.TrigMode* is 1

When *TrigInput* changes to TRUE, *TrigOutput* changes to TRUE, and the *TrigSource[]* first element number and its array element value are output to *TrigMark*.

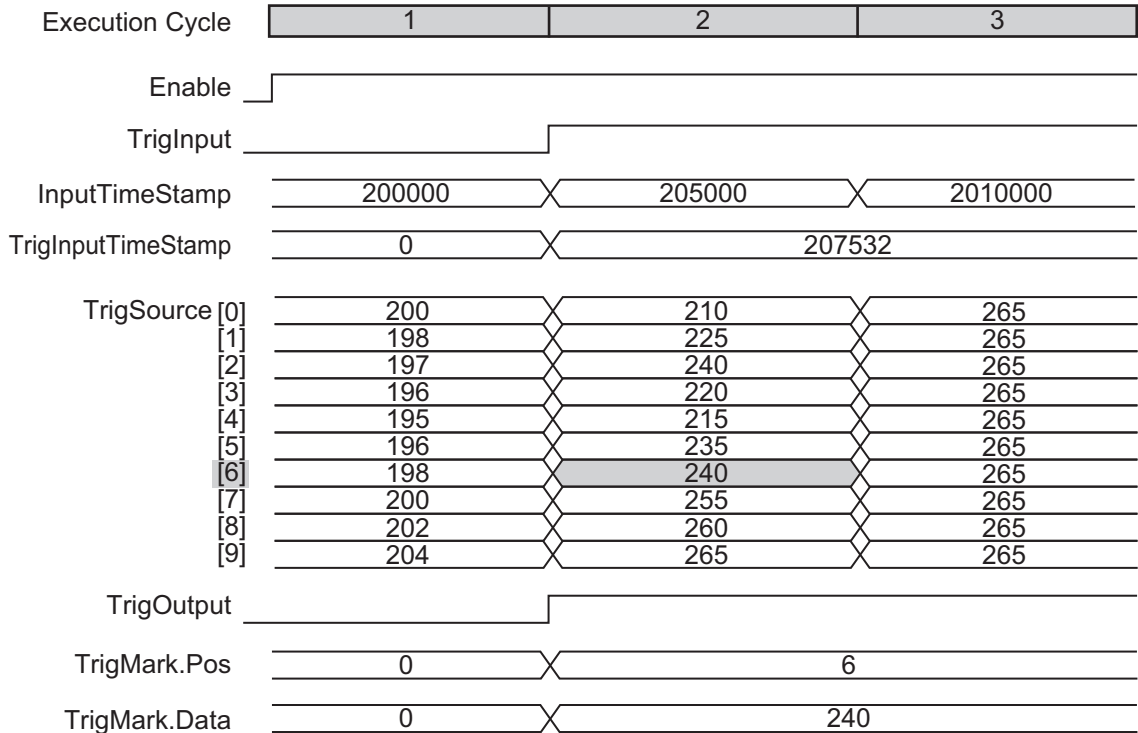
Do not use *TrigConfig.TrigSlope* and *TrigConfig.AnalogTrigLevel* for the trigger conditions.

Example: When *TrigConfig.TrigMode* is 1, and *TrigSource[0..9]* and *InputSize* are 10



- Time Stamp Trigger Mode: *TrigConfig.TrigMode* is 2  
 When *TrigInput* is set to TRUE, the trigger output changes to TRUE. At this time, *InputDataTimeStamp* and *TrigInputTimeStamp* are used to specify the element number and value of the *TrigSource[]* element immediately after the trigger conditions are met, and the element number and value are output to *TrigMark*.

Do not use *TrigConfig.TrigSlope* and *TrigConfig.AnalogTrigLevel* for the trigger conditions.

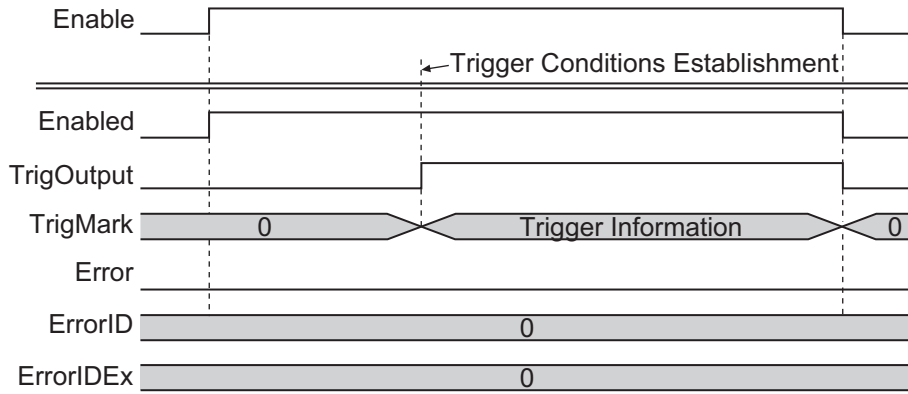


- Setting changes during execution  
 If *TrigConfig* has been changed while execution is in progress, use the changed settings to perform trigger checking.  
 However, changes after trigger conditions are met (when *TrigOutput* changes to TRUE) are not reflected.

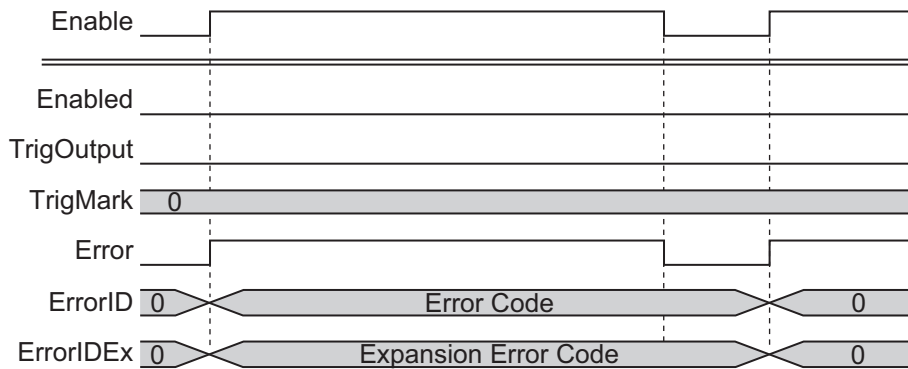
## Timing Chart

The timing charts are shown below.

- If *Enable* is set to TRUE, the trigger checking is started.
- If *Enable* is set to FALSE, the trigger checking is ended.
- When the trigger conditions are met, *TrigOutput* changes to TRUE, and the element number and value of the *TrigSource[]* element that meets the trigger conditions are output to *TrigMark* as trigger information.
- If an error occurs, *Error* changes to TRUE. At the same time, *ErrorID* and *ErrorIDEx* are output.
- When an error occurs, *Error* is retained while *Enable* is TRUE.  
 After *Enable* changes to FALSE, *ErrorID* and *ErrorIDEx* are retained until *Enable* again changes to TRUE.
- Timing Chart for Normal End



• Timing Chart for Error End



## Troubleshooting

Error code	Expansion error code	Status	Description	Corrective action
16#0000	16#00000000	Normal End	---	---
16#3D01	16#00000001	Illegal Element Number	The number of <i>TrigSource[]</i> array elements is less than the <i>InputSize</i>	Check and correct the number of <i>TrigSource[]</i> array elements.
	16#00000002	Illegal Trigger Mode	The trigger mode <i>TrigConfig.TrigMode</i> is outside the valid range	Check and correct the trigger mode.
	16#00000003	Illegal InputSize	The <i>InputSize</i> value is outside the valid range	Correct the <i>InputSize</i> value so that it is within the valid range.
	16#00000004	Event Task Execution	The FB was executed in an event task	Execute this function block in a periodic task.

## Sample Programming

In this sample program, analog input values sampled 100 times per task period by the NX-series High-speed Analog Input Unit are joined into one array variable. This program also uses DataRecorder to record the joined data, and uses LimitTest to compare the recorded data with the test standard. This example assumes that the NX-series High-speed Analog Input Unit is connected to an NX-series CPU Unit.

This sample program consists of two POU: Program 0 in the primary periodic task, and Program 1 in the periodic task 17.

## Program 0 (Primary Periodic Task)

- External Variables

Name	Data type	Constant	Comment
NXBus_N2_NX_Unit_I_O_Data_Active_Status	BOOL		
NXBus_N2_NX_Unit_Error_Status	BOOL		
N2_Ch1_Analog_Input_Value_1_10	ARRAY[0..9] OF INT		
N2_Ch1_Analog_Input_Value_11_20	ARRAY[0..9] OF INT		
N2_Ch1_Analog_Input_Value_21_30	ARRAY[0..9] OF INT		
N2_Ch1_Analog_Input_Value_31_40	ARRAY[0..9] OF INT		
N2_Ch1_Analog_Input_Value_41_50	ARRAY[0..9] OF INT		
N2_Ch1_Analog_Input_Value_51_60	ARRAY[0..9] OF INT		
N2_Ch1_Analog_Input_Value_61_70	ARRAY[0..9] OF INT		
N2_Ch1_Analog_Input_Value_71_80	ARRAY[0..9] OF INT		
N2_Ch1_Analog_Input_Value_81_90	ARRAY[0..9] OF INT		
N2_Ch1_Analog_Input_Value_91_100	ARRAY[0..9] OF INT		
N2_Ch1_Number_of_Samplings	UINT		
N2_Analog_Input_Time_Stamp	ULINT		
Ch1_Input_Array100	ARRAY[0..99] OF REAL		Analog input value of one task period
LogData	ARRAY[0..4999] OF OmronLib\DataRecorder\LogData		Data array to which data is logged
RecordEnd	BOOL		
MasterReadEnd	BOOL		

- Internal Variables

Name	Data type	Default	AT	Retention	Constant	Comment
TrigControl_instance	OmronLib\DataRecorder\TrigControl					
DataRecorder_instance	OmronLib\DataRecorder\DataRecorder					
IO_Data_Is_Valid	BOOL					
TrigConfig	OmronLib\DataRecorder\sTrigConfig					
TrigMark	OmronLib\DataRecorder\sTrigMark					
TrigOut	BOOL					
DataBuf	ARRAY[0..4999] OF OmronLib\DataRecorder\LogData					
RecordInfo	OmronLib\DataRecorder\sRecordInfo					

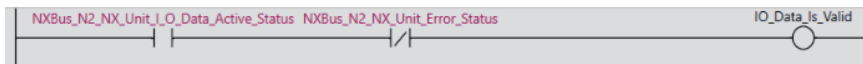
Name	Data type	Default	AT	Retention	Constant	Comment
LogEnable	BOOL					
Success_Concatenate	BOOL					

- Task Settings

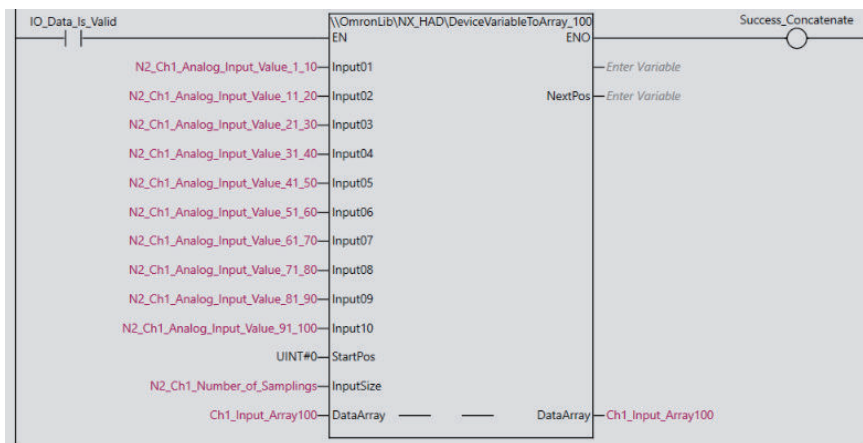
Locate in the primary periodic task.

- LD Program

- (1) Check that normal data is being received from the NX-series High-speed Analog Input Unit. If the NX-series High-speed Analog Input Unit is connected to an EtherCAT Slave Terminal, refer to *Sample Programming 2* on page 4 - 7.



- (2) Join analog input values of one task period into a single variable, `Ch1_Input_Array100[]`.



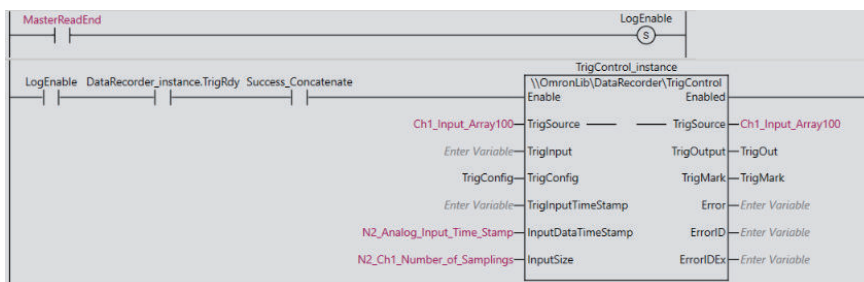
- (3) After the completion of reading the master data CSV file, which is executed in *Program 1 (Periodic Task 17)* on page 5 - 10 as described later, start to wait for a trigger to record input values from the NX-series High-speed Analog Input Unit. *TrigConfig* is used to set the trigger conditions.

- Trigger Conditions

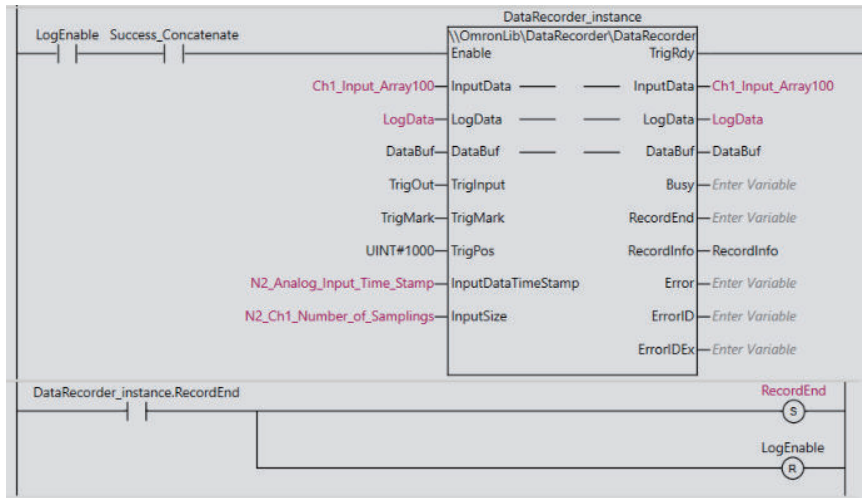
*TrigConfig.TrigMode*=0 (Analog Level Trigger Mode)

*TrigMode.TrigSlope*=FALSE (Rising)

*TrigMode.AnalogTrigLevel*=REAL#100



- (4) If the trigger conditions are met, record into *LogData* the input values for 1,000 samplings before the trigger conditions are met, and input values for 4,000 samplings after the trigger conditions are met.



### Program 1 (Periodic Task 17)

- External Variables

Name	Data type	Constant	Comment
RecordEnd	BOOL		
MasterReadEnd	BOOL		
LogData	ARRAY[0..4999] OF OmronLib\DataRecorder\LogData		

- Internal Variables

Name	Data type	Default	AT	Retention	Constant	Comment
CSVToLogData_instance	OmronLib\DataRecorder\CSVToLogData					
LimitTest_instance	OmronLib\DataRecorder\LimitTest					
MasterData	ARRAY[0..4999] OF OmronLib\DataRecorder\LogData					
FirstOutOfLimitPos	UINT					
Start_P2	BOOL	FALSE				If the value of this variable changes to TRUE, execute the program.
Pass	BOOL					
Cancel	BOOL					

- Task Settings

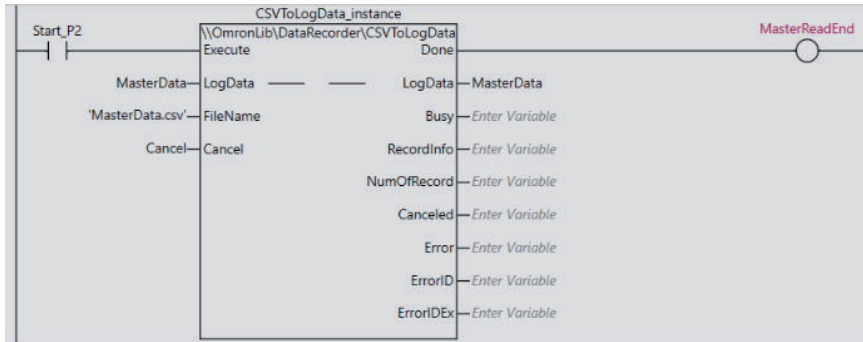
Locate in periodic task 17.



• LD Program

- (1) If *Start\_P2* is set to TRUE, read out the *MasterData.csv* in the SD Memory Card, and set as the test standard data.

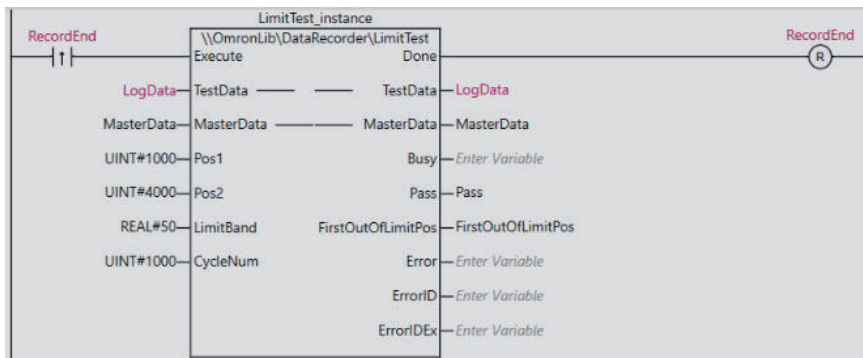
You will have to prepare the test standard data again.



- (2) Compare the *LogData[0..4999]* recorded in *Program 0 (Primary Periodic Task)* on page 5 - 8 with the test standard data, *MasterData[0..4999]*, which was read out from the csv file, for the test.

- Test range: Element numbers from 1,000 to 4,000
- Passing standard: -50 to 50 of the test standard data of the same element number.

Perform the test for every 1,000 elements per periodic task. If they are found to pass the test at the completion, *Pass* changes to TRUE.



**Precautions for Correct Use**

- The sample programming shows only the portion of a program that uses the function or function block from the library. When programming actual applications, also program safety circuits, device interlocks, I/O with other devices, and other control procedures.
- Create a user program that will produce the intended device operation.
- Check the user program for proper execution before you use it for actual operation.

# DataRecorder

DataRecorder joins specified elements of array data into a single array variable every task period, and creates log data in chronological order.

FB/FUN name	Name	FB/FUN	Graphic expression	ST expression
DataRecorder	Data Recorder	FB		<pre>DataRecorder_instance( Enable, InputData, LogData, DataBuf, TrigInput, TrigMark, TrigPos, InputDataTimeStamp, InputSize, TrigRdy, Busy, RecordEnd, RecordInfo, Error, ErrorID, ErrorIDEx );</pre>

## Library Information

Item	Description
Library file name	OmronLib_DataRecorder_Vx_x.slr (x shows the version)
Namespace	OmronLib\DataRecorder
Function block and function number	00196
Source code	Not Published

## Input Variables

	Meaning	Data type	Description	Valid range	Unit	Default
Enable	Execute	BOOL	TRUE: Execute FALSE: Do not execute	TRUE, FALSE	---	FALSE
TrigInput	Trigger input	BOOL	Recording Start Trigger. Inputs the BOOL type variables, I/O data, or Trig-Control FB <i>Trigger Output</i>	TRUE, FALSE	---	FALSE
TrigMark	Trigger Information	OmronLib \DataRecorder \sTrigMark	Inputs the <i>Trigger Information</i> output by the TrigControl FB	---	---	Pos=0 Data=0

	Meaning	Data type	Description	Valid range	Unit	Default
TrigPos	Trigger Position	UINT	For data logging, specifies an element number of <i>LogData[]</i> where the data at the time of trigger input is logged.	Depends on data type	---	0
InputData-TimeStamp	Analog Input Time Stamp	ULINT	Inputs the <i>Analog Input Time Stamp</i> acquired from the I/O data	Depends on data type	nsec	0
InputSize	Number of data	UINT	Inputs the <i>Number of samplings</i> acquired from the I/O data	1 to 1000	---	1

## Output Variables

	Meaning	Data type	Description	Valid range	Unit	Default
TrigRdy	Trigger input enabled	BOOL	Set to TRUE when trigger input can be accepted	TRUE, FALSE	---	---
Busy	Now recording	BOOL	Set to TRUE for the period from when trigger input is set to TRUE until recording is completed	TRUE, FALSE	---	---
RecordEnd	Recording completed	BOOL	Set to TRUE when recorded	TRUE, FALSE	---	---
RecordInfo	Record Information	OmronLib \DataRecorder\RecordInfo	Outputs information such as recording start time, trigger input time, etc.	---	---	---
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 - 17 for details.

## Input-Output Variables

	Meaning	Data type	Description	Valid range	Unit	Default
InputData[]*1	Input Data	ARRAY[*] OF REAL	Inputs the data array to log.	---	---	---

	Meaning	Data type	Description	Valid range	Unit	Default
LogData[]*2	Log Data	ARRAY[*] OF Omron- Lib\DataRe- corder \sLogData	Inputs the data array to store the log into. The number of array elements is the number of records that can be recorded	---	---	---
DataBuf[]*2	Input Buffer	ARRAY[*] OF Omron- Lib\DataRe- corder \sLogData	Buffer that temporarily stores the recording data. The variables need to be assigned, but since this is intermediate data during execution of this FB, do not perform write or referencing.	---	---	---

\*1. The number of array elements is arbitrary. However, the number of array elements must be equal to or more than *InputSize*. In addition, subscripts of the array can start with 0 or any other number.

\*2. The number of array elements is arbitrary. However, the number of array elements must be equal to or more than *TrigPos*. In addition, subscripts of the array can start with 0 or any other number.

## Structure

- OmronLib\DataRecorder\sTrigMark  
Refer to *TrigControl* on page 5 - 2 for the sTrigMark structure.
- OmronLib\DataRecorder\sRecordInfo

Name	Data type	Description
StartDateTime	DT	Record the date and time of the first data item.
EndDateTime	DT	Record the date and time of the last data item. If the FB execution is ended during the recording, no recording is made.
TriggerDateTime	DT	Record the date and time when <i>TrigInput</i> is input. The recording content may vary depending on the usage method. Refer to <i>Function</i> on page 5 - 14 for details.
RecordNum	UINT	Record the number of data elements recorded in <i>LogData[]</i> .
TrigPos	UINT	Record the element number of <i>LogData[]</i> when <i>TrigInput</i> is set to TRUE.

- OmronLib\DataRecorder\sLogData

Name	Data type	Description
xTime	ULINT	Record the time stamp when <i>xData</i> is sampled.
xData	REAL	Record the input data.

## Function

While *Enable* is TRUE, DataRecorder records the data in element numbers of *InputData[]* specified by *InputSize* to *DataBuf[]* for every task period.

After *Enable* changes to TRUE, *TrigRdy* changes to TRUE at the completion of recording the number of data specified by *TrigPos* into *DataBuf[]*.

When *TrigRdy* is TRUE and *TrigInput* changes to TRUE, *TrigRdy* changes to FALSE, and *TrigPos* and subsequent data will be recorded into *LogData[]*. During this recording, *Busy* is TRUE.

When the recording is completed, data in *DataBuf[]* up to *TrigPos* is joined and recorded into *LogData[]*, and *RecordEnd* changes to TRUE.

At the same time, *RecordInfo* is output.

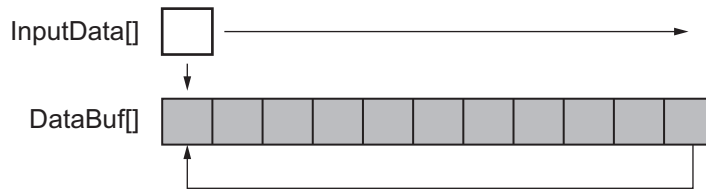
Even if *TrigInput* changes to TRUE while *TrigRdy* is FALSE, it is ignored.

If *Enable* changes to FALSE after the recording is completed, the execution of this FB is ended.

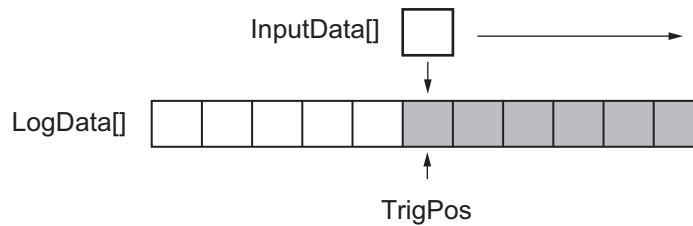
After *Enable* changes to FALSE, *RecordEnd* and *RecordInfo* are retained until *Enable* again changes to TRUE.

- Data Storage Method

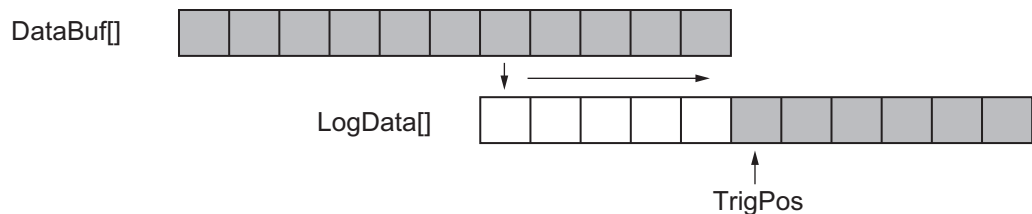
- (1) After *Enable* changes to TRUE, continue recording *InputData[]* to *DataBuf[]* until *TrigInput* changes to TRUE.



- (2) When *TrigInput* is set to TRUE, record data elements in order, starting from *TrigPos* up to the last element of *LogData[]*.



- (3) When recording up to the last element is ended, join the data in *DataBuf[]* with data elements, from the first element of *LogData[]* up to the element of *TrigPos* - 1, in reverse chronological order so as to create a joined data with the first to the last elements of *LogData[]*.



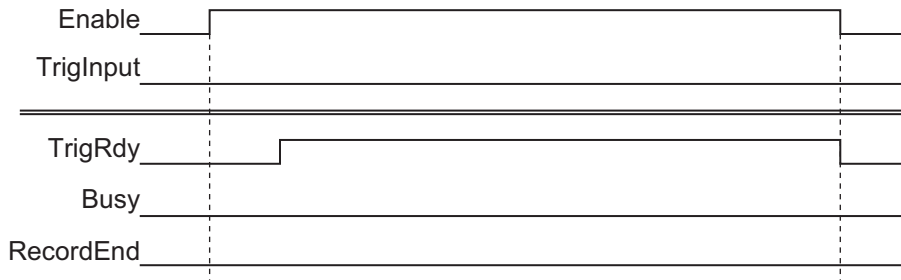
In the cases below, *Error* changes to TRUE and the recording is ended.

- a) When the number of *InputData[]* array elements is less than *InputSize*
- b) When the number of *LogData[]* or *DataBuf[]* array elements is less than the number of *InputData[]* array elements
- c) When *TrigMark.Pos* is an element number outside the range of *InputData[]* array element numbers
- d) When *TrigPos* is outside the valid range of *LogData[]* or *DataBuf[]*
- e) When *InputSize* is 0, or equal to or more than 1001
- f) When the function block was executed in an event task

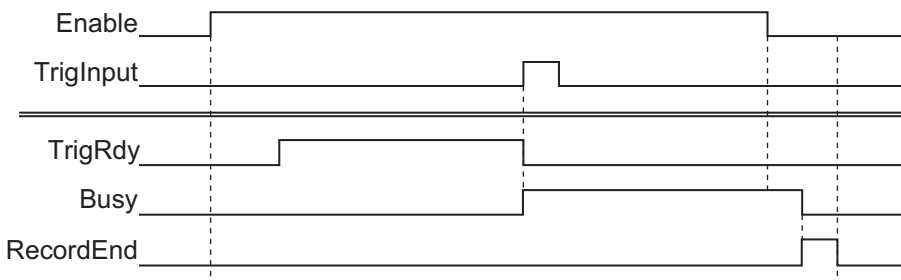
*TrigInput* and *TrigMark* should be input as shown below.

TrigInput	TrigMark
When inputting arbitrary variables	Variable is not assigned
When inputting the TrigControl FB TrigOutput	TrigControl FB TrigMark

- Setting changes during execution  
Changes in *TrigPos* during execution are not accepted. Perform changes with *Enable* set to FALSE.
- Operation when *Enable* is set to FALSE while execution of this FB is in progress
  - a) Operation when *Enable* is set to FALSE before *TrigInput* is set to TRUE  
Immediately end the recording of the *InputData[]*, and set *TrigRdy* to FALSE.  
*LogData[]* and *RecordInfo* are not defined.



- b) Operation when *Enable* is set to FALSE after *Busy* is set to TRUE, and before *RecordEnd* is set to TRUE:  
Record the data of *InputData[]* one task period before *Enable* changes to FALSE. After *Enable* changes to FALSE, the data of *InputData[]* is not recorded.  
*Busy* does not immediately change to FALSE, but stays TRUE until *LogData[]* is created (*RecordEnd* changes to TRUE).  
*RecordEnd* is TRUE only for one task period.  
*RecordInfo.RecordNum* is the number of all data, from the first element of the *LogData[]* array until one task period before *Enable* changes to FALSE.  
The *LogData[]* array element specified by *RecordInfo.RecordNum* and subsequent elements are not defined.  
*RecordInfo.EndDateTime* outputs the date and time when *Enable* changed to FALSE.



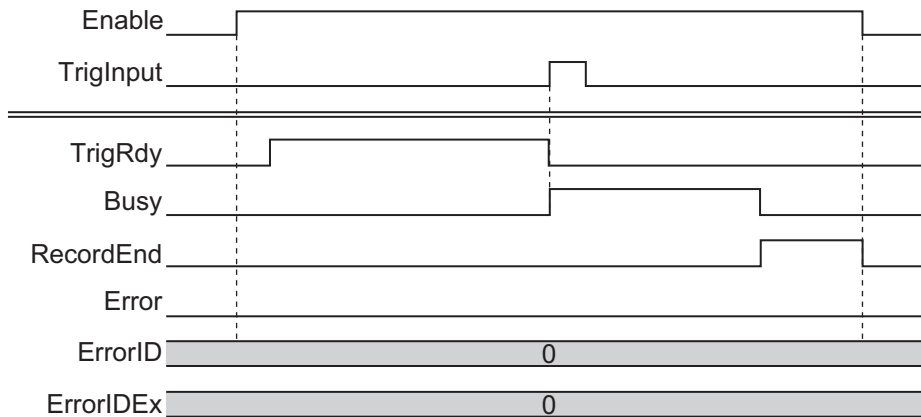
While *Busy* is TRUE, this FB cannot be re-executed.  
After *RecordEnd* is set to TRUE, set *Enable* to FALSE for one or more task period.

## Timing Chart

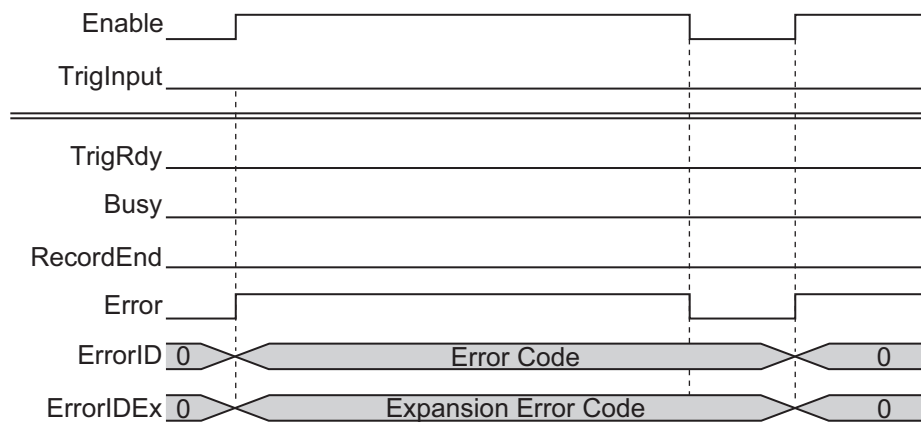
The timing charts are shown below.

- If *Enable* changes to TRUE, recording of *InputData[]* to *DataBuf[]* is started.
- When *Enable* changes to FALSE, the recording is ended.

- If *TrigInput* changes to TRUE, recording to *LogData[]* is started. During the recording, *Busy* is TRUE. When the recording is completed, *RecordEnd* changes to TRUE. If an error occurs, *Error* changes to TRUE. At the same time, *ErrorID* and *ErrorIDEx* are output. When an error occurs, *Error* is retained while *Enable* is TRUE. After *Enable* changes to FALSE, *ErrorID* and *ErrorIDEx* are retained until *Enable* again changes to TRUE.
- Timing Chart for Normal End



- Timing Chart for Error End



## Troubleshooting

Error code	Expansion error code	Status	Description	Corrective action
16#0000	16#00000000	Normal End	---	---
16#3D02	16#00000001	Illegal InputData[] Element Number	The number of <i>InputData[]</i> array elements is less than the the InputSize	Check and correct the number of <i>InputData[]</i> array elements.
	16#00000002	Illegal LogData[], DataBuf[] Element Number	The number of <i>LogData[]</i> array or <i>DataBuf[]</i> array elements is less than the number of <i>InputData[]</i> array elements	Check and correct the number of <i>LogData[]</i> array or <i>DataBuf[]</i> array elements.

Error code	Expansion error code	Status	Description	Corrective action
	16#00000003	Illegal InputSize	The <i>InputSize</i> value is outside the valid range	Correct the <i>InputSize</i> value so that it is within the valid range.
	16#00000004	Illegal Trig-Mark.Pos	<i>TrigMark.Pos</i> exceeds the <i>InputSize</i>	Join the <i>TrigMark</i> output for the TrigControl FB to the <i>TrigMark</i> for this FB.
	16#00000005	Illegal TrigPos	The <i>TrigPos</i> is outside the range of the <i>LogData[]</i> array or <i>DataBuf[]</i> array	<ul style="list-style-type: none"> <li>• Check and correct the <i>TrigPos</i> setting value.</li> <li>• Check and correct the number of <i>LogData[]</i> array or <i>DataBuf[]</i> array elements.</li> </ul>
	16#00000006	Event Task Execution	The FB was executed in an event task	Execute this function block in a periodic task.

## Sample Programming

Refer to *Sample Programming* on page 5 - 7.



# LimitTest

LimitTest checks whether each element value in the data array is within the allowable range of the test standard data.

FB/FUN name	Name	FB/FUN	Graphic expression	ST expression
LimitTest	Upper and Lower Limit Test	FB	<p>The graphic expression shows a function block named 'LimitTest_instance'. It has several inputs on the left: 'Execute', 'TestData', 'MasterData', 'Pos1', 'Pos2', 'LimitBand', and 'CycleNum'. On the right, it has several outputs: 'Done', 'Busy', 'Pass', 'FirstOutOfLimitPos', 'Error', 'ErrorID', and 'ErrorIDEx'. The 'Execute' input is connected to the top of the block, and the other inputs are connected to their respective ports. The outputs are connected to their respective ports on the right side of the block.</p>	<pre>LimitTest_instance( Execute, TestData, MasterData, Pos1, Pos2, LimitBand, CycleNum, Done, Busy, Pass, FirstOutOfLimitPos, Error, ErrorID, ErrorIDEx );</pre>

## Library Information

Item	Description
Library file name	OmronLib_DataRecorder_Vx_x.slr (x shows the version)
Namespace	OmronLib\DataRecorder
Function block and function number	00197
Source code	Not Published

## Input Variables

	Meaning	Data type	Description	Valid range	Unit	Default
Execute	Execute	BOOL	TRUE: Execute FALSE: Do not execute	TRUE, FALSE	---	FALSE
Pos1	Test Range Element 1	UINT	Specifies the element number where testing is started in the test range	Depends on data type	---	0
Pos2	Test Range Element 2	UINT	Specifies the element number where the testing is ended in the test range	Depends on data type	---	0
LimitBand	Tolerance Value	REAL	Specifies the allowable range	0 and greater	---	0.0

	Meaning	Data type	Description	Valid range	Unit	Default
CycleNum	Number of Elements to be Tested in One Period	UINT	Specifies the number of elements to be tested in one task period. If 0 is specified, process the elements to be tested in one task period.	Depends on data type	---	0

## Output Variables

	Meaning	Data type	Description	Valid range	Unit	Default
Done	Done	BOOL	TRUE: Normal end FALSE: Error end, execution in progress, or execution condition not met	TRUE, FALSE	---	---
Busy	Executing	BOOL	TRUE: Executing FALSE: Not executing	TRUE, FALSE	---	---
Pass	Test Pass	BOOL	TRUE: Pass FALSE: Some elements fail	TRUE, FALSE	---	---
FirstOutOfLimitPos	First non-passing element number	UINT	Element number of the first element found to be outside the allowable range	Depends on data type	---	---
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 - 23 for details.

## Input-Output Variables

	Meaning	Data type	Description	Valid range	Unit	Default
TestData[]*1	Test Target Data	ARRAY[*] OF Omron-Lib\DataRecorder \sLogData	Specifies the data array to be tested	---	---	---

	Meaning	Data type	Description	Valid range	Unit	Default
MasterData[]*2	Test Standard Data	ARRAY[*] OF Omron-Lib\DataRecorder\lsLogData	Specifies the data array as the test standard	---	---	---

- \*1. The number of array elements is arbitrary. In addition, subscripts of the array can start with 0 or any other number.
- \*2. The number of array elements is arbitrary. However, it must have the same number of elements as *TestData[]*. In addition, subscripts of the array can start with 0 or any other number.

## Structure

- OmronLib\DataRecorder\lsLogData  
Refer to *TrigControl* on page 5 - 2.

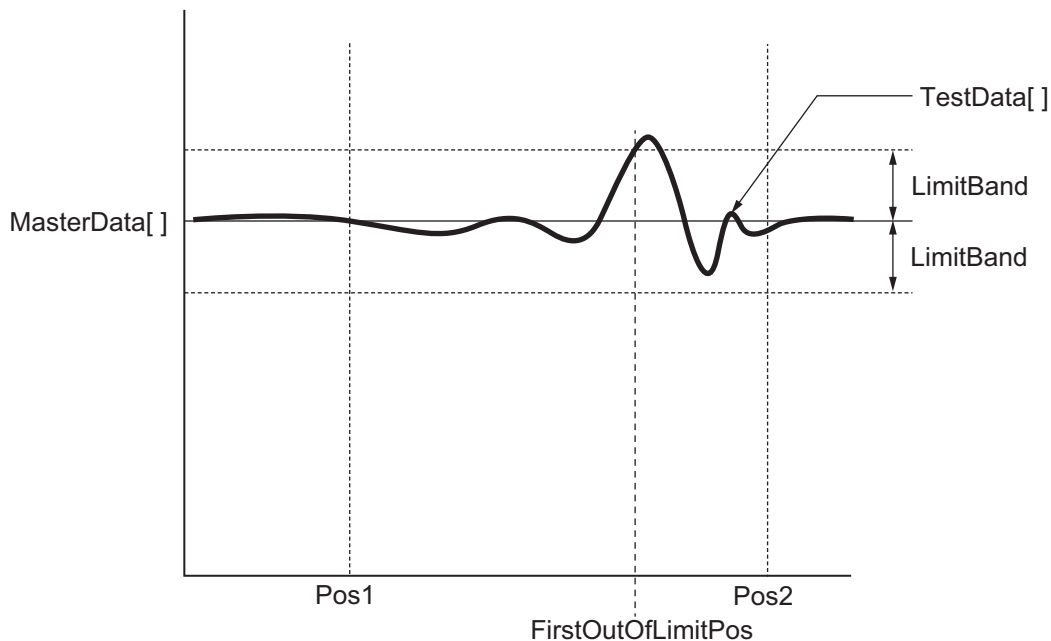
## Function

LimitTest checks data elements in *TestData[]* to verify that they are within the range of *MasterData[]*  $\pm$  *LimitBand*.

*TestData[]* elements to be tested are limited within the specified range between *Pos1* and *Pos2*. Elements outside the range will not be tested.

When *Execute* changes to TRUE, each element in *TestData[]* is tested in order starting from *Pos1*. When all the elements in *TestData[]* starting from *Pos1* to *Pos2* are within the range of *MasterData[]*  $\pm$  *LimitBand*, *Pass* changes to TRUE, and 65535 is output to *FirstOutOfLimitPos*.

If any element is found to be outside the allowable range during the testing, the testing should not be performed up to *Pos2*. When any element value outside the range is detected, *Busy* changes to FALSE and *Done* changes to TRUE. The element number of the first element found to be outside the range is output into *FirstOutOfLimitPos*, as a value relative to the first element number of the array.



The passing standard is as shown below.

$$\text{MasterData}[x] - \text{LimitBand} \leq \text{TestData}[x] \leq \text{MasterData}[x] + \text{LimitBand}$$

In the cases below, *Error* changes to TRUE, and the test is ended.

- When the number of elements in *TestData[]* and that of *MasterData[]* do not match
- When the *LimitBand* value is a negative number, non-numeric number, or  $\infty$
- When *Pos1* and *Pos2* are outside the range of *TestData[]* or *MasterData[]*

*Pos1* and *Pos2* specification method

Specify the *Pos1* and *Pos2* as values relative to the first element.

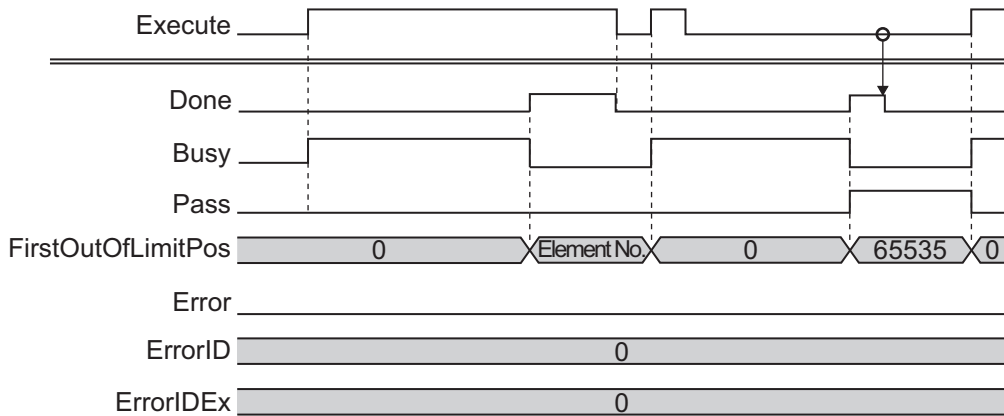
Example: For *TestData* [10..100], if *Pos1* is 0, it indicates *TestData*[10].

- Setting changes during execution  
Any changes in *Pos1*, *Pos2*, *LimitBand*, and *CycleNum* are not accepted while more than one *LimitTest* is enabled or being executed.
- *CycleNum* specification  
You can adjust the execution time of this FB in one task period, as well as the time required to complete the test by specifying *CycleNum* with the number of elements to be tested per task period. If a small value is specified, the execution time per task period is shortened, and the execution period until test completion is lengthened. If a large value is specified, the time until test completion is shortened, and the FB execution time per task period is lengthened. You should make adjustments in consideration of the number of *TestData[]* array elements, the task period of this FB, and load from other operating programs.

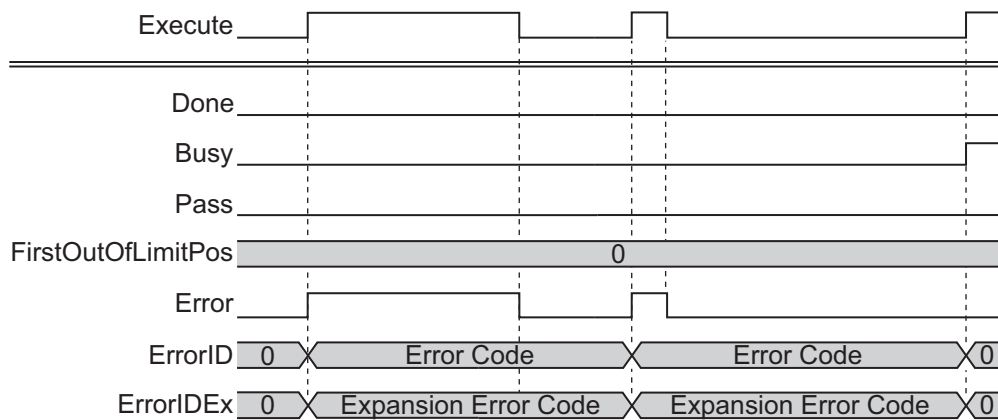
## Timing Chart

The timing charts are shown below.

- When *Execute* changes to TRUE, *Busy* changes to TRUE.
- When the test for all the elements is completed, *Done* changes to TRUE, and the test results are output to *Pass* and *FirstOutOfLimitPos*.  
*Done* is retained until *Execute* changes to FALSE.  
After *Execute* changes to FALSE, *Pass* and *FirstOutOfLimitPos* are retained until *Execute* again changes to TRUE.
- If an error occurs, *Error* changes to TRUE. At the same time, *ErrorID* and *ErrorIDEx* are output. Retain *Error* until *Execute* changes to FALSE.  
After *Execute* changes to FALSE, *ErrorID* and *ErrorIDEx* are retained until *Execute* again changes to TRUE.
- If *Execute* changes to FALSE while *Busy* is TRUE, then *Done* and *Error* are TRUE only for one period after the test end.
- Timing Chart for Normal End



• Timing Chart for Error End



### Precautions for Correct Use

Execution of this function block will be continued until processing is ended even if the value of *Execute* changes to FALSE or the execution time exceeds the task period.  
 The value of *Done* changes to TRUE when processing is ended. Use this to confirm normal ending of processing.

### Troubleshooting

Error code	Expansion error code	Status	Description	Corrective action
16#0000 0	16#00000000	Normal End	---	---
16#3D0 3	16#00000001	Array Element Mismatch	The number of elements in <i>TestData[]</i> and that of <i>MasterData[]</i> do not match	Check and correct the number of elements in <i>TestData[]</i> and <i>MasterData[]</i> respectively.
	16#00000002	Illegal LimitBand	The <i>LimitBand</i> value is outside the valid range	Correct the <i>LimitBand</i> value so that it is within the valid range.
	16#00000003	Illegal Test Range	<i>Pos1</i> and <i>Pos2</i> are outside the <i>TestData[]</i> or <i>MasterData[]</i> range	Check and correct the <i>Pos1</i> and <i>Pos2</i> values.

## Sample Programming

Refer to *Sample Programming* on page 5 - 7.

# CalcFeatureValues

CalcFeatureValues calculates the mean, standard deviation, skewness, kurtosis, maximum value, and minimum value for the test target data array.

FB/FUN name	Meaning	FB/FUN	Graphic expression	ST expression
CalcFeatureValues	Feature Values Calculation	FB		<pre>CalcFeatureValues_instance( Execute, TestData, Pos1, Pos2, Pos3, Pos4, CycleNum, Done, Busy, FeatureValues, Error, ErrorID, ErrorIDEx );</pre>

## Library Information

Item	Description
Library file name	OmronLib_DataRecorder_Vx_x.slr (x shows the version)
Namespace	OmronLib\DataRecorder
Function block and function number	00198
Source code	Not Published

## Input Variables

	Meaning	Data type	Description	Valid range	Unit	Default
Execute	Execute	BOOL	TRUE: Execute FALSE: Do not execute	TRUE, FALSE	---	FALSE
Pos1	Calculation Range Specification 1	UINT	Specifies the last element number in the range for calculating FeatureValues[0]	Depends on data type	---	65535
Pos2	Calculation Range Specification 2	UINT	Specifies the last element number in the range for calculating FeatureValues[1]	Depends on data type	---	65535
Pos3	Calculation Range Specification 3	UINT	Specifies the last element number in the range for calculating FeatureValues[2]	Depends on data type	---	65535

	Meaning	Data type	Description	Valid range	Unit	Default
Pos4	Calculation Range Specification 4	UINT	Specifies the last element number in the range for calculating FeatureValues[3]	Depends on data type	---	65535
CycleNum	Number of Elements Calculated in One Period	UINT	Specifies the number of elements calculated in one task period. If 0 is specified, process the calculation target range element for one task period.	Depends on data type	---	0

## Output Variables

	Meaning	Data type	Description	Valid range	Unit	Default
Done	Done	BOOL	TRUE: Normal end FALSE: Error end, execution in progress, or execution condition not met.	TRUE, FALSE	---	---
Busy	Executing	BOOL	TRUE: Executing FALSE: Not executing	TRUE, FALSE	---	---
FeatureValues[]	Feature Values	ARRAY[0..3] OF Omron-Lib\DataRecorder\Feature-Value	The result of calculating feature values.	---	---	---
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 - 29 for details.



## Input-Output Variables

	Meaning	Data type	Description	Valid range	Unit	Default
TestData[]*1	Calculation Target Data	ARRAY[*] OF Omron-Lib\DataRecorder\Logs\LogData	Specifies the array targeted for calculation of feature values	---	---	---

\*1. The number of array elements is arbitrary. In addition, subscripts of the array can start with 0 or any other number.

## Structure

- OmronLib\DataRecorder\FeatureValue

Name	Data type	Description
xMean	REAL	Mean Value
xSD	REAL	Standard Deviation
xSkewness	REAL	Skew
xKurtosis	REAL	Kurtosis
xMax	REAL	Maximum Value
xMin	REAL	Minimum Value

- OmronLib\DataRecorder\LogData  
Refer to *DataRecorder* on page 5 - 12.

## Function

When *Execute* changes to TRUE, CalcFeatureValues calculates the mean value, standard deviation, skewness, kurtosis, maximum value, and minimum value of the *TestData[]* elements within the range specified in *Pos1*, *Pos2*, *Pos3*, and *Pos4*, and stores the calculation results in *FeatureValues[]*.

- Calculation range specification

The relationships between the calculation range and calculation result storage location are shown below.

You can perform calculation even if the four calculation range elements are overlapping. In addition, you can perform calculation even if the first element and last element sizes are reversed.

Calculation Range		Calculation Result Storage Location
First Element	Last Element	
TestData[Array Start]	TestData[Pos1]*1	FeatureValues[0]
TestData[Pos1+1]*2	TestData[Pos2]*1	FeatureValues[1]
TestData[Pos2+1]*2	TestData[Pos3]*1	FeatureValues[2]
TestData[Pos3+1]*2	TestData[Pos4]*1	FeatureValues[3]

- \*1. If values exceeding 65535 or the number of *TestData[]* elements are set for *Pos1*, *Pos2*, *Pos3*, or *Pos4*, the calculation range and later calculation ranges become invalid, and all calculation results are stored as 0.
- \*2. If *Pos1+1*, *Pos2+1*, and *Pos3+1* are outside the *TestData[]* range, the calculation range and later calculation ranges become invalid, and all calculation results are stored as 0.

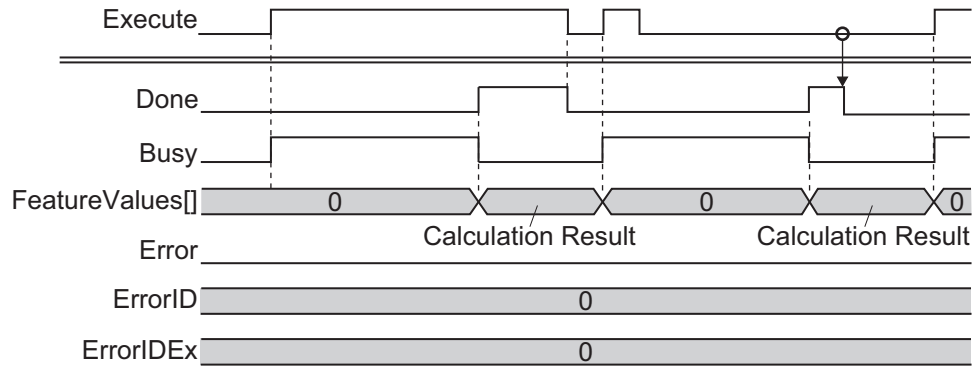
Pos1	Pos2	Pos3	Pos4	FeatureValues [0]	FeatureValues [1]	Feature-Values [2]	Feature-Values [3]
65535	Invalid	Invalid	Invalid	0	0	0	0
Valid value	65535	Invalid	Invalid	Calculation result	0	0	0
Valid value	Valid value	65535	Invalid	Calculation result	Calculation result	0	0
Valid value	Valid value	Valid value	65535	Calculation result	Calculation result	Calculation result	0
Valid value	Valid value	Valid value	Valid value	Calculation result	Calculation result	Calculation result	Calculation result

- Specification method for *Pos1*, *Pos2*, *Pos3*, and *Pos4*.  
Specify *Pos1*, *Pos2*, *Pos3*, and *Pos4* as relative values from the beginning of the *TestData[]* array.  
Example: For *TestData[10..100]*, if *Pos1* is 0, it indicates *TestData[10]*.
- Setting changes during execution  
Multi-execution of this FB, and changes while execution is in progress for *Pos1*, *Pos2*, *Pos3*, *Pos4*, and *CycleNum* are not accepted.
- *CycleNum* specification  
When you specify the number of calculation elements per task period for *CycleNum*, you can adjust the execution time per task period of the FB and the time until the calculation is completed.  
If a small value is specified, the execution time per task period is shortened, and the execution period until calculation completion is lengthened.  
If a large value is specified, the time until calculation completion is shortened, and the FB execution time per task period is lengthened.  
Perform the adjustment with consideration for the number of *TestData[]* array elements, the task period that is located in this FB, and the load from programs other than this FB that are operating.

## Timing Chart

The timing charts are shown below.

- When *Execute* changes to TRUE, *Busy* changes to TRUE.
- If the calculation of all feature values is completed, *Done* changes to TRUE, and the calculation result is output to *FeatureValues[]*.  
*Done* is retained until *Execute* changes to FALSE.  
After *Execute* changes to FALSE, *FeatureValues[]* is retained until *Execute* changes to TRUE again.
- If *Busy* changes to FALSE while *Busy* is TRUE, then *Done* is TRUE only for one period after the calculation end.



## Precautions for Correct Use

Execution of this function block will be continued until processing is ended even if the value of *Execute* changes to FALSE or the execution time exceeds the task period.

The value of *Done* changes to TRUE when processing is ended. Use this to confirm normal ending of processing.

## Troubleshooting

Error code	Expansion error code	Status	Description	Corrective action
16#0000	16#00000000	Normal End	---	---

## Sample Programming

The NX-series High-speed Analog Input Unit uses *DeviceVariableToArray\_100* to combine the analog input values sampled 100 times per task period to an array variable. Furthermore, this is a program to record the joined data with *DataRecorder*, and perform a test with *LimitTest* to compare the recording data with the test standard data.

This example assumes that the NX-series High-speed Analog Input Unit is connected to an NX-series CPU Unit.

This sample program consists of two POU: Program 0 in the primary periodic task, and Program 1 in the periodic task 17.

### Program 0 (Primary Periodic Task)

- External Variables

Name	Data type	Constant	Comment
NXBus_N2_NX_Unit_I_O_Data_Active_Status	BOOL		
NXBus_N2_NX_Unit_Error_Status	BOOL		
N2_Ch1_Analog_Input_Value_1_10	ARRAY[0..9] OF INT		
N2_Ch1_Analog_Input_Value_11_20	ARRAY[0..9] OF INT		
N2_Ch1_Analog_Input_Value_21_30	ARRAY[0..9] OF INT		

Name	Data type	Constant	Comment
N2_Ch1_Analog_Input_Value_31_40	ARRAY[0..9] OF INT		
N2_Ch1_Analog_Input_Value_41_50	ARRAY[0..9] OF INT		
N2_Ch1_Analog_Input_Value_51_60	ARRAY[0..9] OF INT		
N2_Ch1_Analog_Input_Value_61_70	ARRAY[0..9] OF INT		
N2_Ch1_Analog_Input_Value_71_80	ARRAY[0..9] OF INT		
N2_Ch1_Analog_Input_Value_81_90	ARRAY[0..9] OF INT		
N2_Ch1_Analog_Input_Value_91_100	ARRAY[0..9] OF INT		
N2_Ch1_Number_of_Samplings	UINT		
N2_Analog_Input_Time_Stamp	ULINT		
N2_Ch1_Trigger_Input_Time_Stamp	ULINT		
N2_Ch1_Trigger_Input	BOOL		
Ch1_Input_Array100	ARRAY[0..99] OF REAL		Analog input value of one task period
LogData	ARRAY[0..4999] OF OmronLib\DataRecorder\LogData		Recorder Data Array
RecordInfo	OmronLib\DataRecorder\LogRecordInfo		
RecordEnd	BOOL		

- Internal Variables

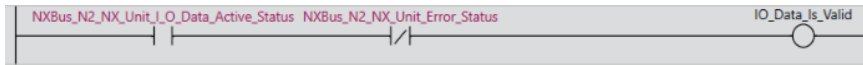
Name	Data type	Default	AT	Retention	Constant	Comment
TrigControl_instance	OmronLib\DataRecorder\TrigControl					
DataRecorder_instance	OmronLib\DataRecorder\DataRecorder					
IO_Data_Is_Valid	BOOL					
TrigConfig	OmronLib\DataRecorder\LogTrigConfig					
TrigMark	OmronLib\DataRecorder\LogTrigMark					
TrigOut	BOOL					
DataBuf	ARRAY[0..4999] OF OmronLib\DataRecorder\LogData					
Start_P3	BOOL	FALSE				If the value of this variable changes to TRUE, execute the program.
LogEnable	BOOL					
Success_Concatenate	BOOL					

- Task Settings

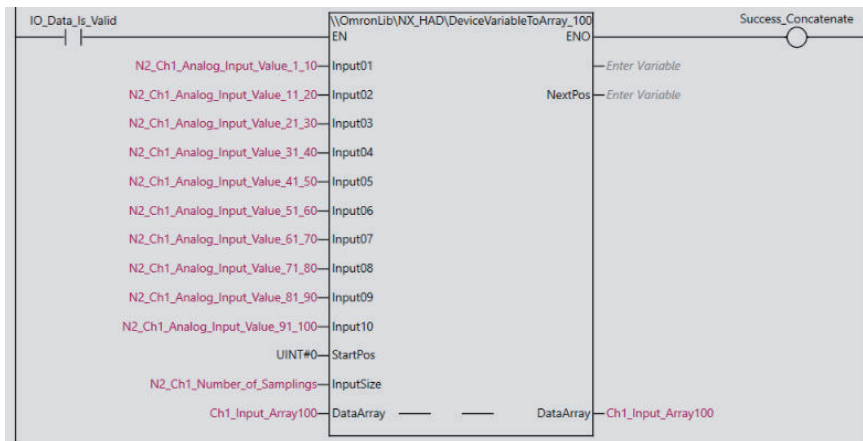
Locate in the primary periodic task.

- LD Program

- (1) Check that normal data is being received from the NX-series High-speed Analog Input Unit. If the NX-series High-speed Analog Input Unit is connected to an EtherCAT Slave Terminal, refer to *Sample Programming 2* on page 4 - 7.



- (2) Join analog input values of one task period into a single variable, *Ch1\_Input\_Array100[]*.

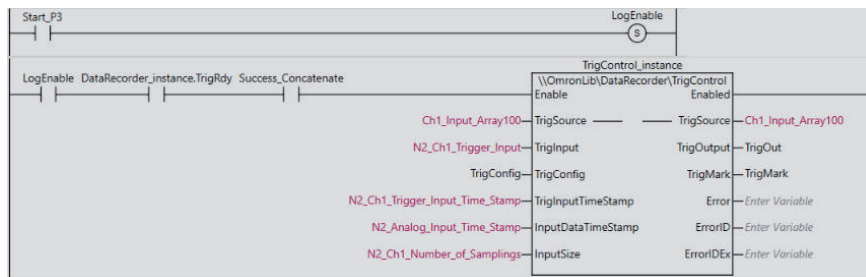


- (3) If *Start\_P3* changes to TRUE, start to wait for a trigger to record input values from the NX-series High-speed Analog Input Unit. Set the trigger conditions in *TrigConfig*.

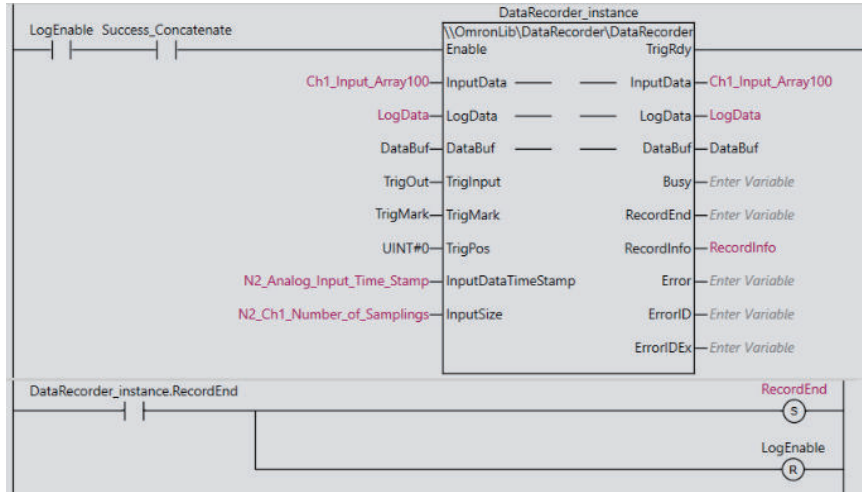
- Trigger Conditions

*TrigConfig.TrigMode=2* (Time Stamp Trigger Mode)

*TrigInput=N2\_Ch1\_Trigger\_Input* (Ch1 Trigger Input of NX-series High-speed Analog Input Unit)



- (4) If the trigger conditions are met, record into *LogData[]* input values for 5,000 samplings after the trigger conditions are met.



## Program 1 (Periodic Task 17)

- External Variables

Name	Data type	Constant	Comment
RecordEnd	BOOL		
RecordInfo	OmronLib\DataRecorder\sRecordInfo		
LogData	ARRAY[0..4999] OF OmronLib\DataRecorder\sLogData		

- Internal Variables

Name	Data type	Default	AT	Retain	Constant	Comment
CalcFeatureValues_instance	OmronLib\DataRecorder\CalcFeatureValues					
LogDataToCSV_instance	OmronLib\DataRecorder\LogDataToCSV					
FeatureValues	ARRAY[0..3] OF OmronLib\DataRecorder\sFeatureValue					
Cancel	BOOL					

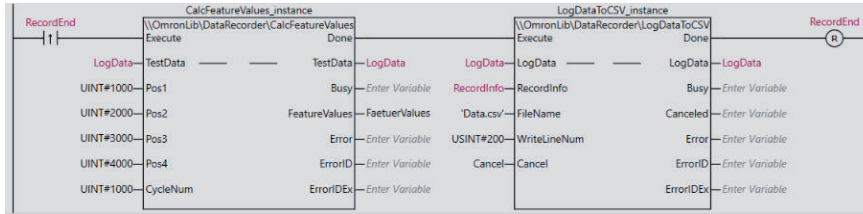
- Task Settings

Locate in periodic task 17.

- LD Program

(1) When the recording of the data executed in *Program 0 (Primary Periodic Task)* on page 5 - 29 is completed, and *RecordEnd* changes to TRUE, calculate the feature values of the recording data *LogData[]* for each data interval specified in *Pos1* to *Pos4*.

When you complete the calculation of feature values, write the *LogData[]* data to an SD Memory Card with the file name *Data.csv*.



In *FeatureValues[]*, store the following feature values.

	.xMean	.xSD	.xSkew-ness	.xKurto-sis	.xMax	.xMin	Remarks
Feature-Values [0]	Mean value	Standard deviation	Skew	Kurtosis	Maximum value	Minimum value	Feature values of LogData[0..1000]
Feature-Values [1]	Mean value	Standard deviation	Skew	Kurtosis	Maximum value	Minimum value	Feature values of LogData[1001..2000]
Feature-Values [2]	Mean value	Standard deviation	Skew	Kurtosis	Maximum value	Minimum value	Feature values of LogData[2001..3000]
Feature-Values [3]	Mean value	Standard deviation	Skew	Kurtosis	Maximum value	Minimum value	Feature values of LogData[3001..4000]



**Precautions for Correct Use**

- The sample programming shows only the portion of a program that uses the function or function block from the library. When programming actual applications, also program safety circuits, device interlocks, I/O with other devices, and other control procedures.
- Create a user program that will produce the intended device operation.
- Check the user program for proper execution before you use it for actual operation.

# LogDataToCSV

LogDataToCSV outputs the log data created in the DataRecorder FB as a CSV file (\*.csv) to an SD Memory Card.

FB/FUN name	Name	FB/FUN	Graphic expression	ST expression
LogDataToCSV	Log Data CSV File Write	FB		<pre>LogDataToCSV_instance( Execute, LogData, RecordInfo, FileName, WriteLineNum, Cancel, Done, Busy, Canceled, Error, ErrorID, ErrorIDEx );</pre>

## Library Information

Item	Description
Library file name	OmronLib_DataRecorder_Vx_x.slr (x shows the version)
Namespace	OmronLib\DataRecorder
Function block and function number	00199
Source code	Not Published

## Input Variables

	Meaning	Data type	Description	Valid range	Unit	Default
Execute	Execute	BOOL	TRUE: Execute FALSE: Do not execute	TRUE, FALSE	---	FALSE
RecordInfo	Record Information	OmronLib \DataRecorder'sRecordInfo	Inputs record information output by the DataRecorder FB	---	---	---
FileName	File Name	STRING[66 ]	File name of CSV file to write. Items including directory can be specified.	66 bytes max. (65 single-byte alphanumeric characters plus the final NULL character)	---	''



	Meaning	Data type	Description	Valid range	Unit	Default
WriteLine-Num	Number of Lines Output in One Period	USINT	Specifies the number of lines written in each execution period with this parameter	1 to 255	---	1
Cancel	Cancel	BOOL	TRUE: Cancels the file write	TRUE, FALSE	---	FALSE

## Output Variables

	Meaning	Data type	Description	Valid range	Unit	Default
Done	Done	BOOL	TRUE: Normal end FALSE: Error end, execution in progress, or execution condition not met	TRUE, FALSE	---	---
Busy	Executing	BOOL	TRUE: Executing FALSE: Not executing	TRUE, FALSE	---	---
Canceled	Cancel End	BOOL	TRUE: Canceled FALSE: Not canceled	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 - 39 for details.

## Input-Output Variables

	Meaning	Data type	Description	Valid range	Unit	Default
LogData[]*1	Log Data	ARRAY[*] OF Omron-Lib\DataRecorder\nsLogData	Specifies log data to read-out to the CSV file	---	---	---

\*1. The number of array elements is arbitrary. In addition, subscripts of the array can start with 0 or any other number.

## Structure

- OmronLib\DataRecorder\nsLogData  
Refer to *DataRecorder* on page 5 - 12.
- OmronLib\DataRecorder\nsRecordInfo  
Refer to *DataRecorder* on page 5 - 12.

## Function

When *Execute* changes to TRUE, *LogDataToCSV* outputs the log data stored in *LogData[]* and *RecordInfo* to the SD Memory Card in the CSV comma separated values file format. The output file name is specified in *FileName*.

For *FileName*, you can specify items by including the directory. If a specified directory does not exist, an error occurs.

If a directory is not specified, the output file is created in the root directory on the SD Memory Card.

If *Cancel* changes to TRUE while write execution is in progress, cancel the file to write. When the cancel processing ends, *Canceled* changes to TRUE. Even when it is canceled, the files that were already output remain in the SD Memory Card.

- File Format

The file format is as follows.

```
"StartDateTime"      , "(RecordInfo.StartDateTime) "
"EndDateTime"        , "(RecordInfo.EndDateTime) "
"TriggerDateTime"    , "(RecordInfo.TriggerDateTime) "
"RecordNum"          , "(RecordInfo.RecordNum) "
"TriggerPos"         , "(RecordInfo.TrigPos) "
"TimeStamp"          , "Data"
"(LogData[m].xTime) " , "(LogData[m].xData) "
"(LogData[m+1].xTime) , "(LogData[m+1].xData) "
:
:
:
"(LogData[n-1].xTime) , "(LogData[n-1].xData) "
"(LogData[n].xTime)   , "(LogData[n].xData) "
```

The data on the left side and right side are separated by a comma.

The parentheses ( ) output the value of the variable written in the parentheses.

"m" refers to the first element number of *LogData[]*, and "n" to the last element number.

An output example is shown below.

```
"StartDateTime" , "2017-11-22-10:20:30.123456789"
"EndDateTime"   , "2017-11-22-10:20:305.123456789"
"TriggerDateTime" , "2017-11-22-10:20:303.123456789"
"RecordNum"     , "10000"
"TriggerPos"    , "5000"
"TimeStamp"     , "Data"
"123456789"     , "0.98765"
"123456790"     , "0.88765"
"123456791"     , "0.98765"
:
:
:
"123457000"     , "0.88765"
"123457001"     , "0.89765"
```

- Setting changes during execution  
Multi-execution of this FB and changes in the *FileName* and *WriteLineNum* are not accepted.
- *WriteLineNum* specification  
When you specify the number of output lines per task period for *WriteLineNum*, you can adjust the execution time per task period of the FB, and the time until the output is completed.  
If a small value is specified, the execution time per task period is shortened, and the execution period until calculation completion is lengthened.  
If a large value is specified, the time until calculation completion is shortened, and the FB execution time per task period is lengthened.  
Perform the adjustment with consideration for the number of *LogData[]* array elements, the task period that is located in this FB, and the load from programs other than this FB that are operating.  
However, 1 KB per task period in this FB is set as the upper limit.

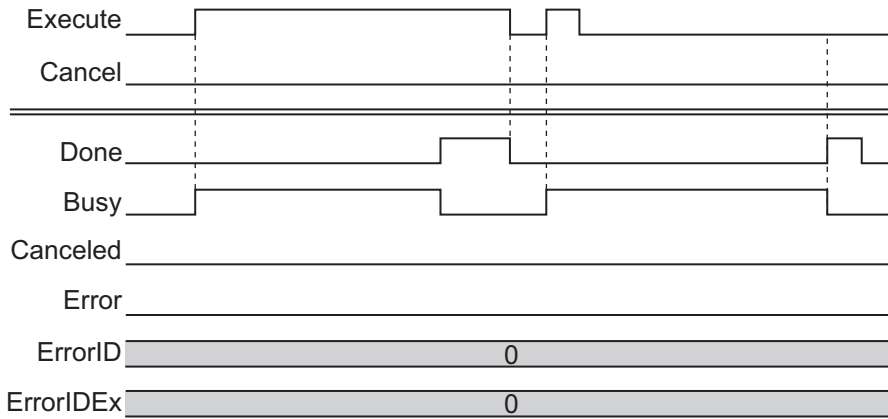
## Additional Information

- *RecordInfo.StartDateTime*, *RecordInfo.EndDateTime*, and *RecordInfo.TriggerDateTime* are used to convert to text strings and write with *DtToString* instruction. Refer to the *NJ/NX-series Instructions Reference Manual (Cat. No. W502)* for details on the *DtToString* instruction.
- *LogData[].xData* is used to convert to a text string with *RealToFormatString* instruction. For the number of digits, the overall is eight and the fractional part is six. Refer to the *NJ/NX-series Instructions Reference Manual (Cat. No. W502)* for details on the *RealToFormatString* instruction.

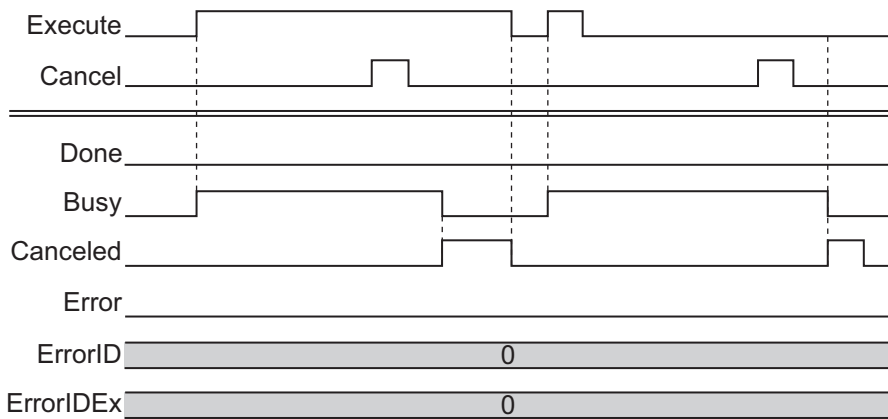
## Timing Chart

The timing charts are shown below.

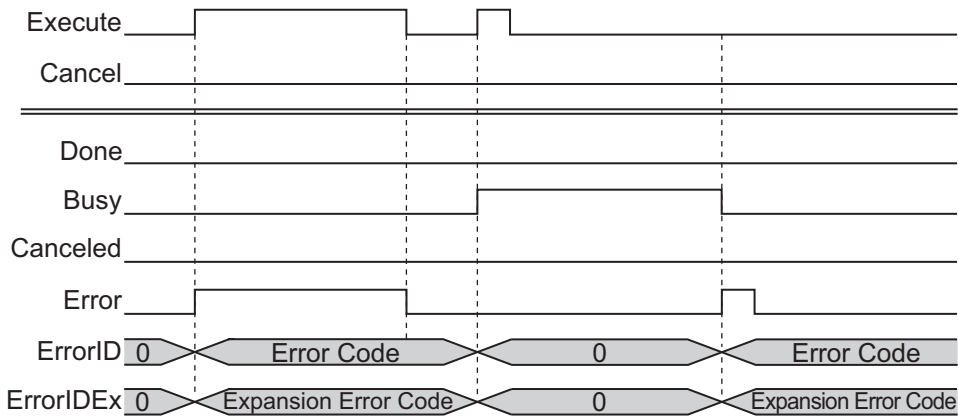
- When *Execute* changes to TRUE, *Busy* changes to TRUE.
- When the write of all the log data is completed, *Done* changes to TRUE. *Done* is retained until *Execute* changes to FALSE.
- If *Cancel* changes to TRUE while write execution is in progress, the file to write is canceled, and *Canceled* changes to TRUE.
- If an error occurs, *Error* changes to TRUE. At the same time, *ErrorID* and *ErrorIDEx* are output. *Error* and *Canceled* are retained until *Execute* changes to FALSE. After *Execute* changes to FALSE, *ErrorID* and *ErrorIDEx* are retained until *Execute* changes to TRUE again.
- If *Execute* changes to FALSE while *Busy* is TRUE, then *Done*, *Canceled*, and *Error* are TRUE for only one period after the end of write/cancel processing.
- Timing Chart for Normal End



• Timing Chart for Cancel End



• Timing Chart for Error End



### Precautions for Correct Use

- Execution of this function block will be continued until processing is ended even if the value of *Execute* changes to FALSE or the execution time exceeds the task period. The value of *Done* changes to TRUE when processing is ended. Use this to confirm normal ending of processing.
- While execution of this FB is in progress, do not access a write target file from another FB.

## Troubleshooting

Error code	Expansion error code	Status	Description	Corrective action
16#0000	16#00000000	Normal End	---	---
16#0400	16#00000000	Input Value Out of Range	The file name specified with <i>FileName</i> contains one or more characters that cannot be used.	Set <i>FileName</i> correctly.
			The directory name specified with <i>FileName</i> is too long.	Check the length of the text strings specified with <i>FileName</i> , and set the directory name so that it is within the valid range.
16#1400	16#00000000	SD Memory Card Access Failure	This occurs when this FB is used in the NJ/NX-series CPU Units. Refer to the <i>NJ/NX-series Troubleshooting Manual (Cat. No. W503)</i> .	
16#1401	16#00000000	SD Memory Card Write-protected		
16#1402	16#00000000	SD Memory Card Insufficient Capacity		
16#1404	16#00000000	Too Many Files/Directories		
16#1405	16#00000000	File Already in Use		
16#140A	16#00000000	Write Access Denied		
16#140B	16#00000000	Too Many Files Open		
16#140D	16#00000000	File or Directory Name Is Too Long		
16#140E	16#00000000	SD Memory Card Access Failure		
16#4400	16#00000000	Shared Folder Access Failure		This occurs when this FB is used in the NY-series Industrial PC. Refer to the <i>NY-series Troubleshooting Manual (Cat. No. W564)</i> .
16#4402	16#00000000	Shared Folder Insufficient Capacity		
16#4404	16#00000000	Too Many Files/Directories		
16#440D	16#00000000	File or Directory Name Is Too Long		
16#440E	16#00000000	Shared Folder Access Failure		
16#3D05	16#00000001	Illegal <i>WriteLineNum</i>	<i>WriteLineNum</i> is outside the range.	Specify a value which is within the valid range.

## Sample Programming

Refer to *Sample Programming* on page 5 - 7.

# CSVToLogData

CSVToLogData reads out the log data recorded in the SD Memory Card from a CSV file to the *LogData[]* array variables as the test standard data for LimitTest FB.

FB/FUN name	Name	FB/FUN	Graphic expression	ST expression
CSVTo-LogData	Log Data CSV File Read-Out	FB		<pre> CSVToLogData_in- stance( Execute, LogData, FileName, Cancel, Done, Busy, RecordInfo, NumOfRecord, Canceled, Error, ErrorID, ErrorIDEx ); </pre>

## Library Information

Item	Description
Library file name	OmronLib_DataRecorder_Vx_x.slr (x shows the version)
Namespace	OmronLib\DataRecorder
Function block and function number	00200
Source code	Not Published

## Input Variables

	Meaning	Data type	Description	Valid range	Unit	Default
Execute	Execute	BOOL	TRUE: Execute FALSE: Do not execute	TRUE, FALSE	---	FALSE
FileName	File Name	STRING[66 ]	File name of CSV file to read. Items including directory can be specified.	66 bytes max. (65 single-byte alphanumeric characters plus the final NULL character)	---	''
Cancel	Cancel	BOOL	TRUE: Cancels the file read-out	TRUE, FALSE	---	FALSE

## Output Variables

	Meaning	Data type	Description	Valid range	Unit	Default
Done	Done	BOOL	TRUE: Normal end FALSE: Error end, execution in progress, or execution condition not met	TRUE, FALSE	---	---
Busy	Executing	BOOL	TRUE: Executing FALSE: Not executing	TRUE, FALSE	---	---
RecordInfo	Record Information	OmronLib \DataRecorder\sRecordInfo	Records information for read-out log data.	---	---	---
NumOfRecord	Number of read-out records	UINT	Number of records read-out	Depends on data type	---	---
Canceled	Cancel End	BOOL	TRUE: Canceled. FALSE: Not canceled.	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 - 45 for details.

## Input-Output Variables

	Meaning	Data type	Description	Valid range	Unit	Default
LogData[]*1	Log Data	ARRAY[*] OF OmronLib\DataRecorder \sLogData	Specifies log data to read-out to the CSV file	---	---	---

\*1. The number of array elements is arbitrary. However, it must be more than the number of data recorded in the read-out file. In addition, subscripts of the array can start with 0 or any other number.

## Structure

- OmronLib\DataRecorder\sLogData  
Refer to *DataRecorder* on page 5 - 12.
- OmronLib\DataRecorder\sRecordInfo  
Refer to *DataRecorder* on page 5 - 12.



## Function

When *Execute* changes to TRUE, CSVToLogData reads out the log data from the file specified in *FileName*, and stores it in *LogData[]*.

In addition, store the recording information in *RecordInfo*.

Output the number of records read out from the file to *NumOfRecord*.

After *Execute* changes to FALSE, *RecordInfo* and *NumOfRecord* are retained until *Execute* changes to TRUE again.

For *FileName*, you can specify items including the directory. If a directory is not specified, the root directory on the SD Memory Card is referenced. If a specified file does not exist, an error occurs.

In addition, the file type that this FB can handle is the CSV file created in the LogDataToCSV FB. An error occurs if a different file format is specified.

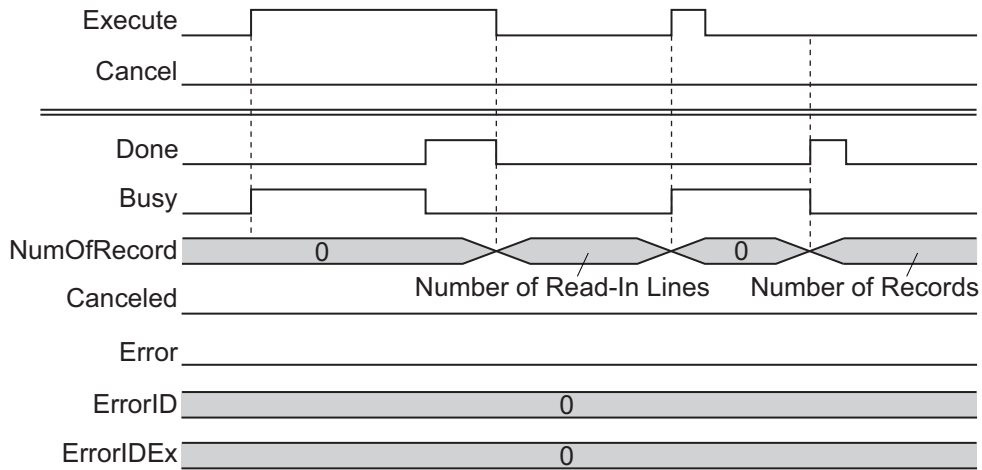
If *Cancel* changes to TRUE while read-out execution is in progress, cancel the file read-out. When the cancel processing ends, *Canceled* changes to TRUE. If the read-out is canceled, *RecordInfo* and *NumOfRecord* are not output. The content of *LogData[]* will be undefined.

- File Format  
Refer to the LogDataToCSV FB.
- Setting changes during execution  
Multi-execution of this FB and changes in the *FileName* while execution is in progress are not accepted.

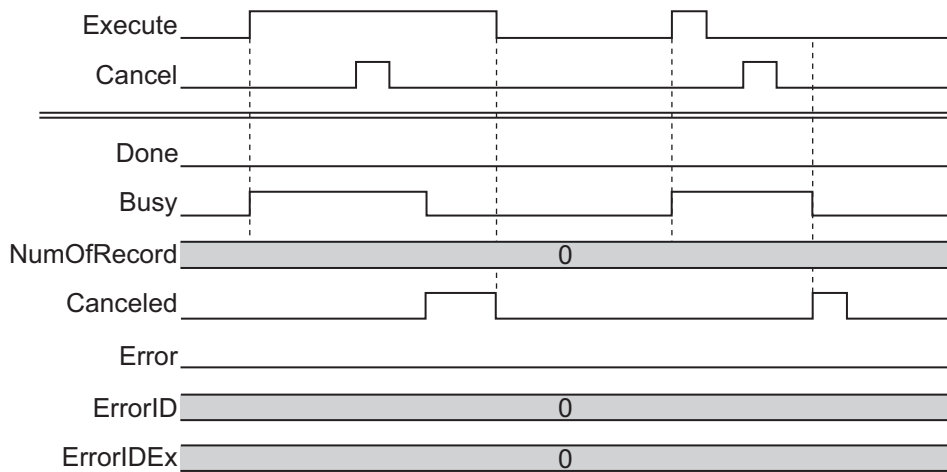
## Timing Chart

The timing charts are shown below.

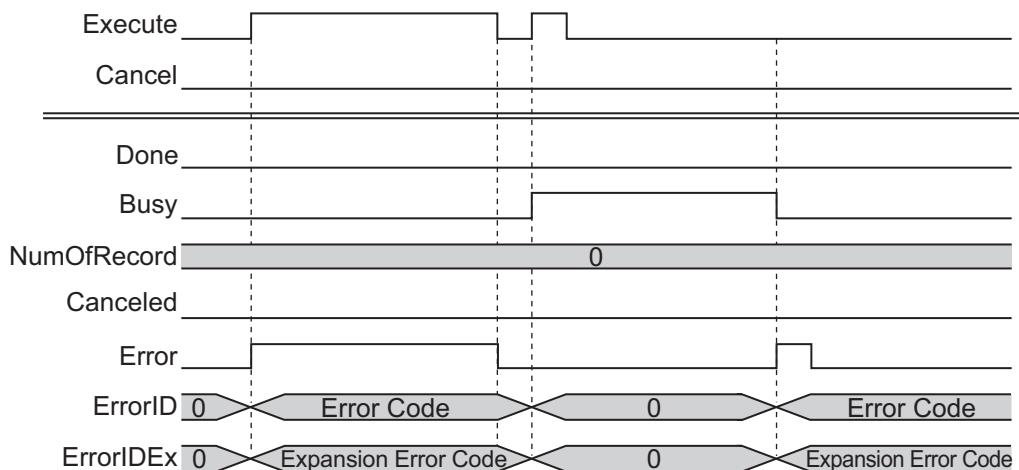
- When *Execute* changes to TRUE, *Busy* changes to TRUE.
- When the read-out of all the log data is completed, *Done* changes to TRUE.  
*Done* is retained until *Execute* changes to FALSE.  
After *Execute* changes to FALSE, *RecordInfo* and *NumOfRecord* are retained until *Execute* changes to TRUE again.
- If *Cancel* changes to TRUE while read-out execution is in progress, the file to read-out is canceled, and *Canceled* changes to TRUE.
- If an error occurs, *Error* changes to TRUE. At the same time, *ErrorID* and *ErrorIDEx* are output.  
*Error* and *Canceled* are retained until *Execute* changes to FALSE.  
After *Execute* changes to FALSE, *ErrorID* and *ErrorIDEx* are retained until *Execute* changes to TRUE again.
- If *Execute* changes to FALSE while *Busy* is TRUE, then *Done*, *Canceled*, and *Error* are TRUE for only one period after the end of the read-out/cancel processing.
- Timing Chart for Normal End



- Timing Chart for Cancel End



- Timing Chart for Error End



## Precautions for Correct Use

- Execution of this function block will be continued until processing is ended even if the value of *Execute* changes to FALSE or the execution time exceeds the task period. The value of *Done* changes to TRUE when processing is ended. Use this to confirm normal ending of processing.
- While execution of this FB is in progress, do not access a read-out target file from another FB.

## Troubleshooting

Error code	Expansion error code	Status	Description	Corrective action
16#0000	16#00000000	Normal End	---	---
16#0400	16#00000000	Input Value Out of Range	The file name specified with <i>FileName</i> contains one or more characters that cannot be used.	Set <i>FileName</i> correctly.
			The directory name specified with <i>FileName</i> is too long.	Check the length of the text strings specified with <i>FileName</i> , and set the directory name so that it is within the valid range.
16#1400	16#00000000	SD Memory Card Access Failure	This occurs when this FB is used in the NJ/NX-series CPU Units. Refer to the <i>NJ/NX-series Troubleshooting Manual (Cat. No. W503)</i> .	
16#1401	16#00000000	SD Memory Card Write-protected		
16#1402	16#00000000	SD Memory Card Insufficient Capacity		
16#1404	16#00000000	Too Many Files/Directories		
16#1405	16#00000000	File Already in Use		
16#140A	16#00000000	Write Access Denied		
16#140B	16#00000000	Too Many Files Open		
16#140D	16#00000000	File or Directory Name Is Too Long		
16#140E	16#00000000	SD Memory Card Access Failure		
16#4400	16#00000000	Shared Folder Access Failure	This occurs when this FB is used in the NY-series Industrial PC. Refer to the <i>NY-series Troubleshooting Manual (Cat. No. W564)</i> .	
16#4402	16#00000000	Shared Folder Insufficient Capacity		
16#4404	16#00000000	Too Many Files/Directories		
16#440D	16#00000000	File or Directory Name Is Too Long		
16#440E	16#00000000	Shared Folder Access Failure		

Error code	Expansion error code	Status	Description	Corrective action
16#3D06	16#00000001	Insufficient number of <i>LogData[]</i> elements	The number of <i>LogData[]</i> elements is insufficient for the number of records in the file.	Check the number of records in the file, and extend the <i>LogData[]</i> to an element number exceeding the number of records.
	16#00000002	Invalid Format	The format does not match the record content in the file.	Check whether the read-out target file is the one that was output by LogData-ToCSV.

## Sample Programming

Refer to *Sample Programming* on page 5 - 7.



# Appendix

---

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

---

<b>A-1</b>	<b>Referring to Library Information .....</b>	<b>A - 2</b>
A-1-1	Library Attributes, and FB or FUN Attributes .....	A - 2
A-1-2	Referring to Attributes of Libraries, Function Blocks, and Functions .....	A - 3
<b>A-2</b>	<b>Referring to Function Block and Function Source Codes .....</b>	<b>A - 5</b>

A

# 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

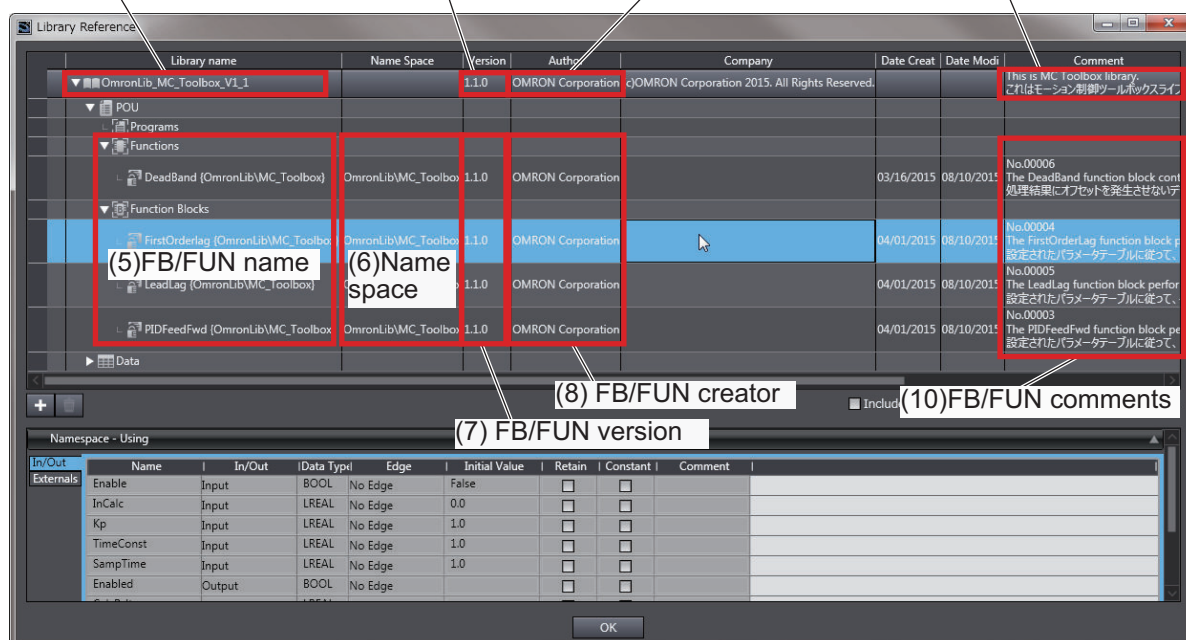
You can refer to the library attributes of library information, and FB or FUN attributes at the following Sysmac Studio locations.

- Library Reference Dialog Box
- Toolbox
- Programming screen

### Library Reference Dialog Box

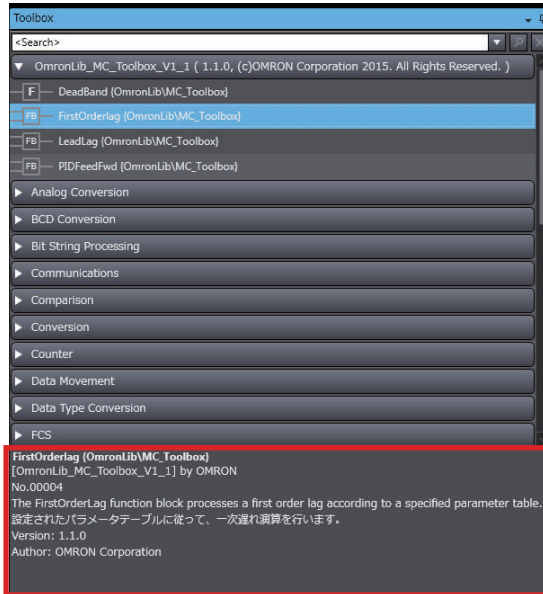
When you refer to the libraries, the library information is displayed at the locations shown below.

- (1) Library file name      (2) Library version      (3) Library creator      (4) Library comments



### Toolbox

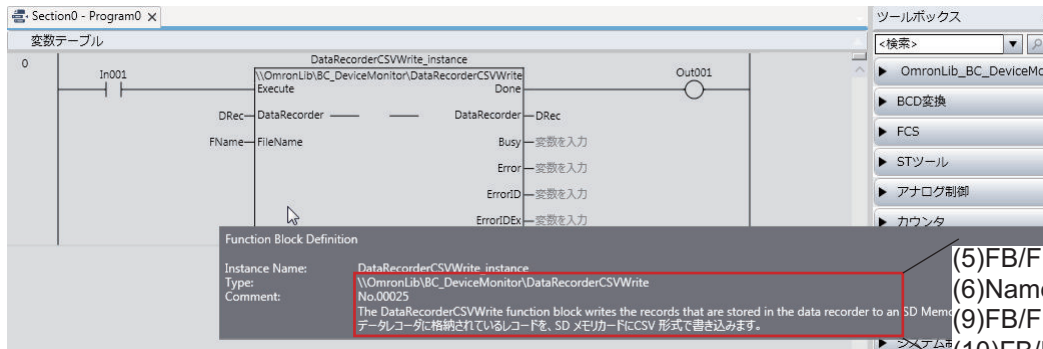
Select a function block or function to display its library information at the bottom of the Toolbox Pane. The text "**by OMRON**" which is shown on the right of the library name (1) indicates that this library was provided by OMRON.



- (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



## A-2 Referring to Function Block and Function Source Codes

You can refer to the source codes of function blocks and functions provided by OMRON to customize them to suit the user's environment.

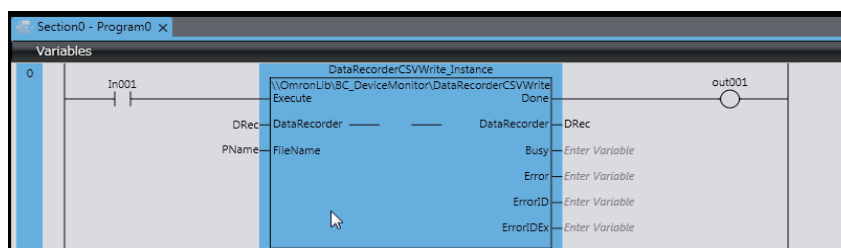
User function blocks and user functions can be created based on the copies of these source codes. The following are the examples of items that you may need to customize.

- Customizing the "Array Size" to suit the memory capacity of the user's Controller
- Customizing the "Data Type" to suit the user-defined data types

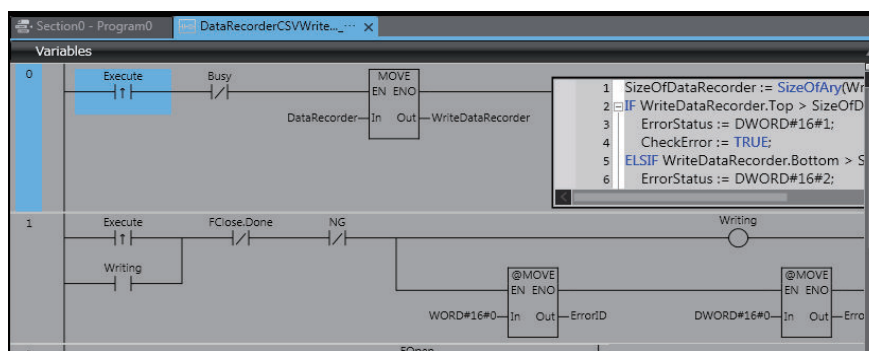
Note that you can access only function blocks and functions whose Source code published/not published is set to "Published" in the library information shown in their individual specifications.

Use the following procedure to refer to the source codes of function blocks and functions.

- 1 Select a function block or function in the program.



- 2 Double-click or right-click and select **To Lower Layer** from the menu. The source code is displayed.





### Precautions for Correct Use

---

- For function blocks and functions whose source codes are not published, the following dialog box is displayed in the above step 2. Click the **Cancel** button.





# Index



# Index

---

## C

---

CalcFeatureValues..... 5 - 25  
CSVToLogData..... 5 - 41

## D

---

DataRecorder..... 5 - 12  
DeviceVariableToArray\_\*\*\*..... 4 - 2

## L

---

LimitAlarm\_HAD..... 4 - 14  
LimitTest..... 5 - 19  
LogDataToCSV..... 5 - 34

## S

---

ScaleTrans\_HAD..... 4 - 11

## T

---

TrigControl..... 5 - 2



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

Contact: [www.ia.omron.com](http://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:

© OMRON Corporation 2018-2019 All Rights Reserved.  
In the interest of product improvement,  
specifications are subject to change without notice.

Cat. No. **W607-E1-02**

0119