



**Sysmac Library**

# **High Speed Analog Test Library**

---

**User's Manual**

**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 Test 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 Test Library uses a time series to record the NX-series High Speed Analog Input Unit NX-HAD□□□ analog input values.

It provides the necessary functions for product tests in the production process, such as calculation of the data maximum value, minimum value, average value, and other feature amounts, and comparative judgment with master data, saving data files, etc.

## Intended Audience

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

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

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

## Applicable Products

This manual covers the following products.

Item	Product name	Model	Version
Automation Software	Sysmac Studio	SYSMAC-SE□□□□	Version 1.23 or higher
Device	CPU Unit	NX701-□□□□	Version 1.18 or later
		NX1P2-□□□□□□(1)	
		NX102-□□□□	Version 1.30 or later
	NJ501-□□□□ NJ301-□□□□ NJ101-□□□□	Version 1.18 or later	
	Industrial PC	NY5□□-1□□□	Version 1.18 or later

## Introduction

Item	Product name	Model	Version
	NX-series High Speed Analog Input Unit	NX-HAD□□□	Version 1.0 or later

# Manual Structure

## Page Structure

The following page structure is used in this manual.

Level 2 heading — 4-3 Mounting Units

Level 3 heading — 4-3-1 Connecting Controller Components

A step in a procedure — 1 Join the Units so that the connectors fit exactly.  
Indicates a procedure.

Special information — **Precautions for Correct Use**  
Icons indicate precautions, additional information, or reference information.

Manual name — NJ-series CPU Unit Hardware User's Manual (W500)

Level 1 heading — 4 Installation and Wiring

Level 2 heading — 4-3 Mounting Units

Level 3 heading — 4-3-1 Connecting Controller Components

Page tab — 4

Gives the current headings.

Gives the number of the main section.

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>3</b>
Page Structure.....	3
Special Information .....	4
<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
WARNING.....	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 Test 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 - 2
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 an 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.

## WARNING



### 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.



## Safety Precautions

---

Check the user program, data, and parameter settings for proper execution before you use them for actual operation.



---

Do not allow anyone other than persons corresponding to *Intended Audience* on page 1, or persons 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 using actual devices, also program 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.
- In a function or function block with an Enabled output variable, if the value of Enabled is FALSE, do not use the processing result of the function or function block as a command value to the control target.
- In the function block with Execute, do not perform re-execution by the same instance. The output value of the function block will return to the default value.

# Related Manuals

The 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-□□□□ NX1P2-□□□□ NJ501-□□□□ NJ301-□□□□ NJ101-□□□□	Learning about the errors that may be detected in an NJ/NX-series Controller.	Concepts on managing errors that may be detected in an NJ/NX-series Controller and information on individual errors are described.
Sysmac Studio Version 1 Operation Manual	W504	SYSMAC-SE2□□□	Learning about the operating procedures and functions of the Sysmac Studio.	Describes the operating procedures of the Sysmac Studio.
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.

Manual name	Man. No.	Model	Application	Description
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	NY532-1□□□ NY512-1□□□	Learning how to program and set up the Controller functions of an NY-series Industrial PC.	The following information is provided on the NY-series Controller functions. <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	NY532-1□□□ NY512-1□□□	Learning detailed specifications on the basic instructions of an NY-series Industrial PC.	The instructions in the instruction set (IEC 61131-3 specifications) are described.
NY-series Troubleshooting Manual	W564	NY532-1□□□ NY512-1□□□	Learning about the errors that may be detected in an NY-series Industrial PC.	Concepts on managing errors that may be detected in an NY-series Controller and information on individual errors are described.
NY-series IPC Machine Controller Industrial Panel PC / Industrial Box PC Motion Control User's Manual	W559	NY532-1□□□ NY512-1□□□	Learning about motion control settings and programming concepts of an NY-series Industrial PC.	The settings and operation of the Controller and programming concepts for motion control are described.
NY-series Motion Control Instructions Reference Manual	W561	NY532-1□□□ NY512-1□□□	Learning about the specifications of the motion control instructions of an NY-series Industrial PC.	The motion control instructions are described.

## Related Manuals

Manual name	Man. No.	Model	Application	Description
NY-series IPC Machine Controller Industrial Panel PC / Industrial Box PC Built-in EtherNet/IP™ Port User's Manual	W563	NY532-1□□□ NY512-1□□□	Using the built-in EtherNet/IP port in an NY-series Industrial PC.	Information on the built-in EtherNet/IP port is provided. Information is provided on the basic setup, tag data links, and other features.
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 Unit High Speed Analog Input Unit User's Manual	SBCA-461	NX-HAD□□□	Learning how to use the NX-series High Speed Analog Input Unit.	Describes the hardware, setup methods, and functions of the NX-series High Speed Analog Input Unit.

# 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-E1-01</b>
-----------------	-------------------

↑  
Revision code

Revision code	Date	Revised content
A	May 2018	Original production

## Revision History

# Sections in this Manual

---

<b>1</b>	<b>Sysmac Library Usage Procedure</b>	<b>1</b>
<b>2</b>	<b>High Speed Analog Test 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>

## Sections in this Manual



# 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

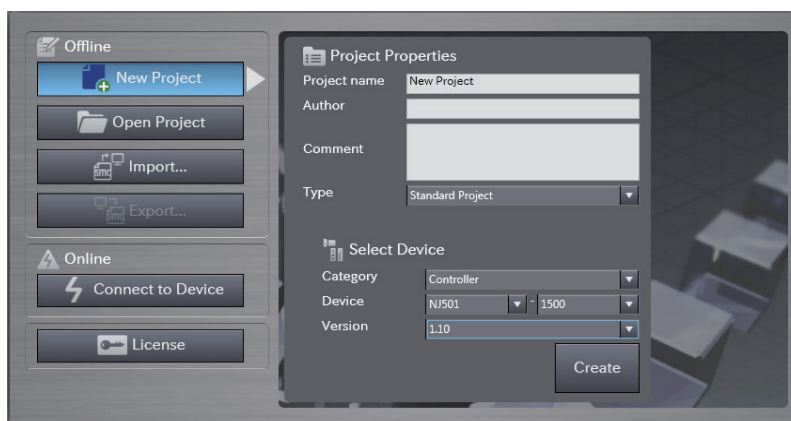


## Version Information

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.

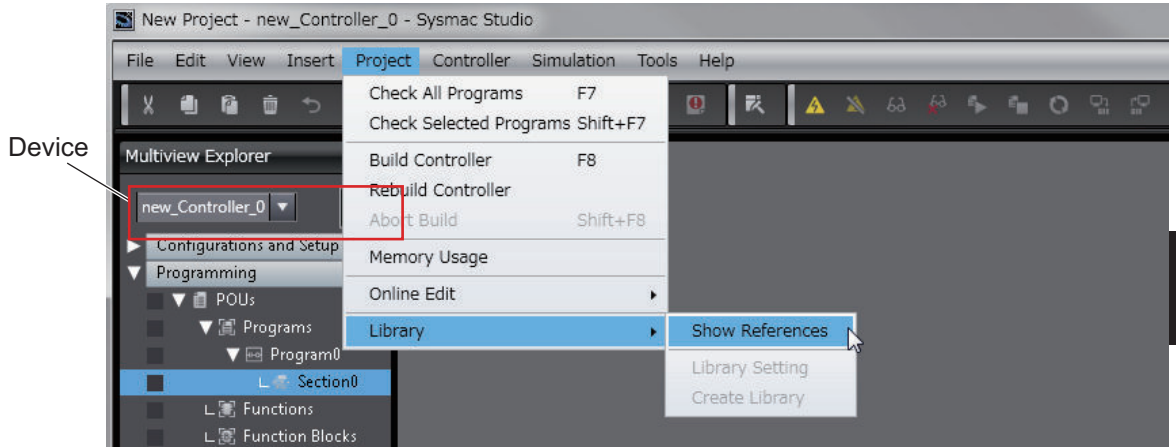


## Precautions for Correct Use

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 device selection version, refer to *Applicable Products* on page 1.

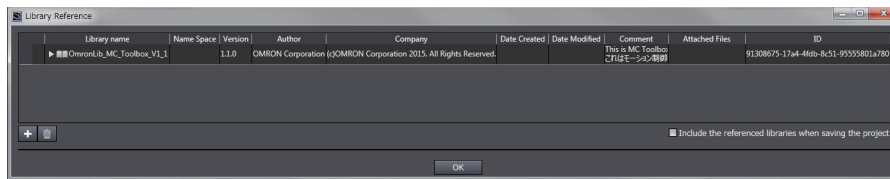
- 2 Select **Project - Library - Show References**.



**Precautions for Correct Use**

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

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



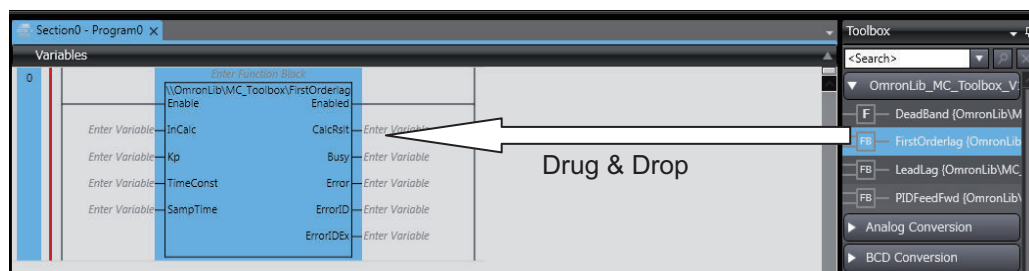
Sysmac Library is read into the project.

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

For the procedure for adding and setting libraries in the above screen, refer to the *Sysmac Studio Version 1 Operation Manual (W504)*.

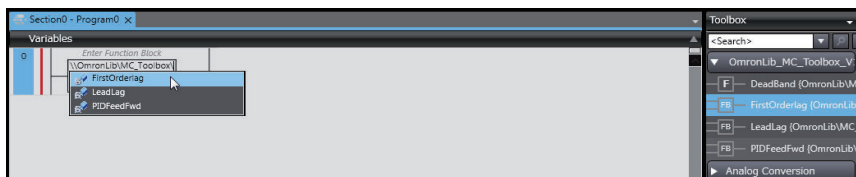
**4 Insert the Sysmac Library's function blocks and functions into the circuit using one of the following two methods.**

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



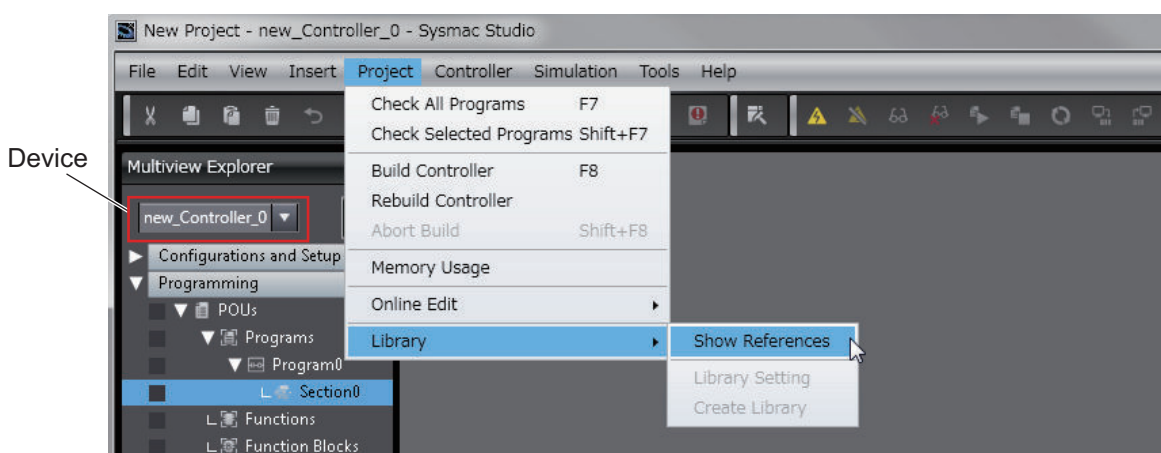
- Right-click the Ladder Editor, select **Insert Function Block** in the menu, and enter the fully qualified name (¥¥namespace¥¥FBname).

# 1 Sysmac Library Usage Procedure



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

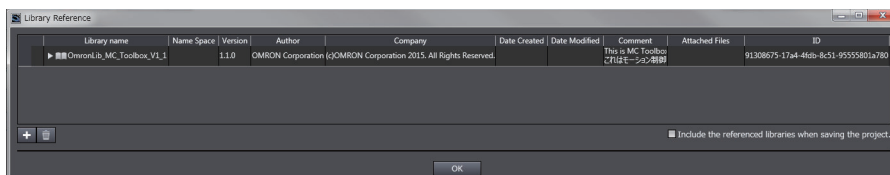
- 1 Start Sysmac Studio and open a project in which any old-version Sysmac Library is included.
- 2 Select **Project - Library - Show References**.



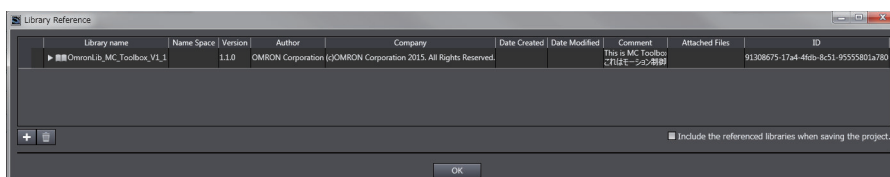
### Precautions for Correct Use

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

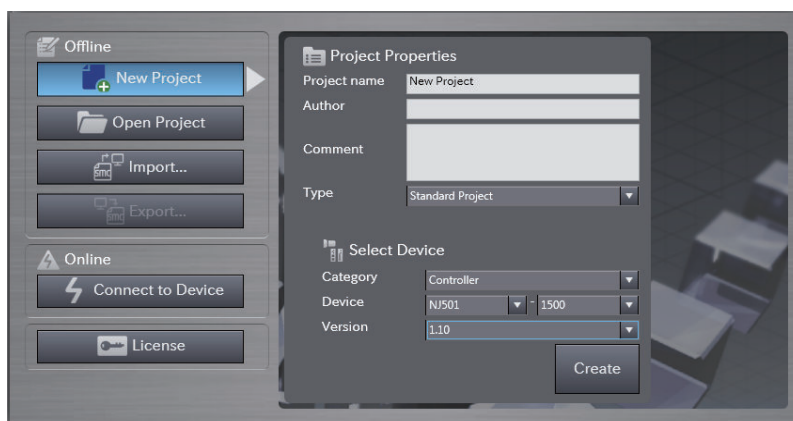
Even when Sysmac Library is not installed on your computer, you can use Sysmac Library by uploading it from the CPU Unit or Industrial PC to your computer.

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

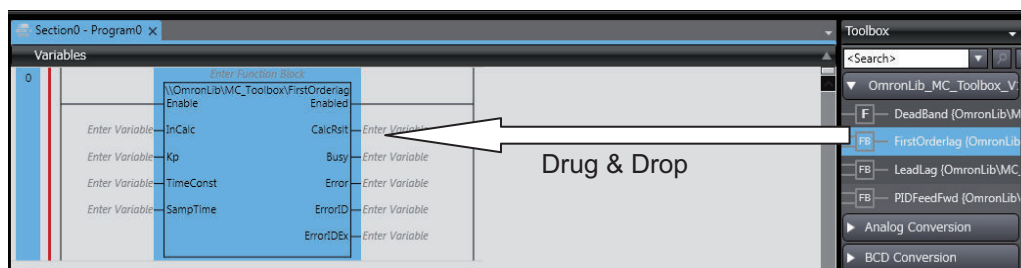
### Version Information

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

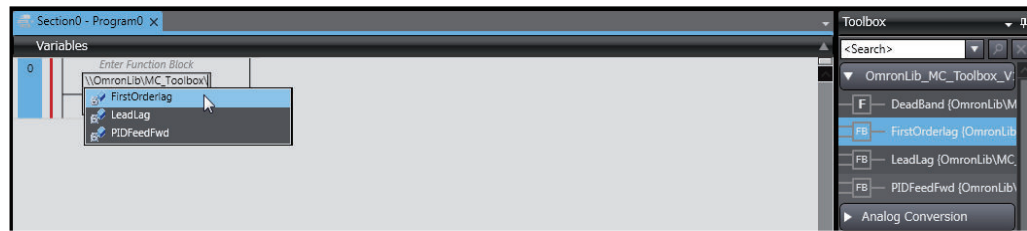
- 1 Start the Sysmac Studio and create a new project in which you want to use Sysmac Library.



- 2 Connect online to the CPU Unit or Industrial PC.
- 3 Upload the POU's 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 POU's 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.

# 1 Sysmac Library Usage Procedure



# High Speed Analog Test Library

This section describes the shared specifications of each FB in the High Speed Analog Test 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 Test Library uses a time series to record the NX-series High Speed Analog Input Unit NX-HAD□□□ analog input values.

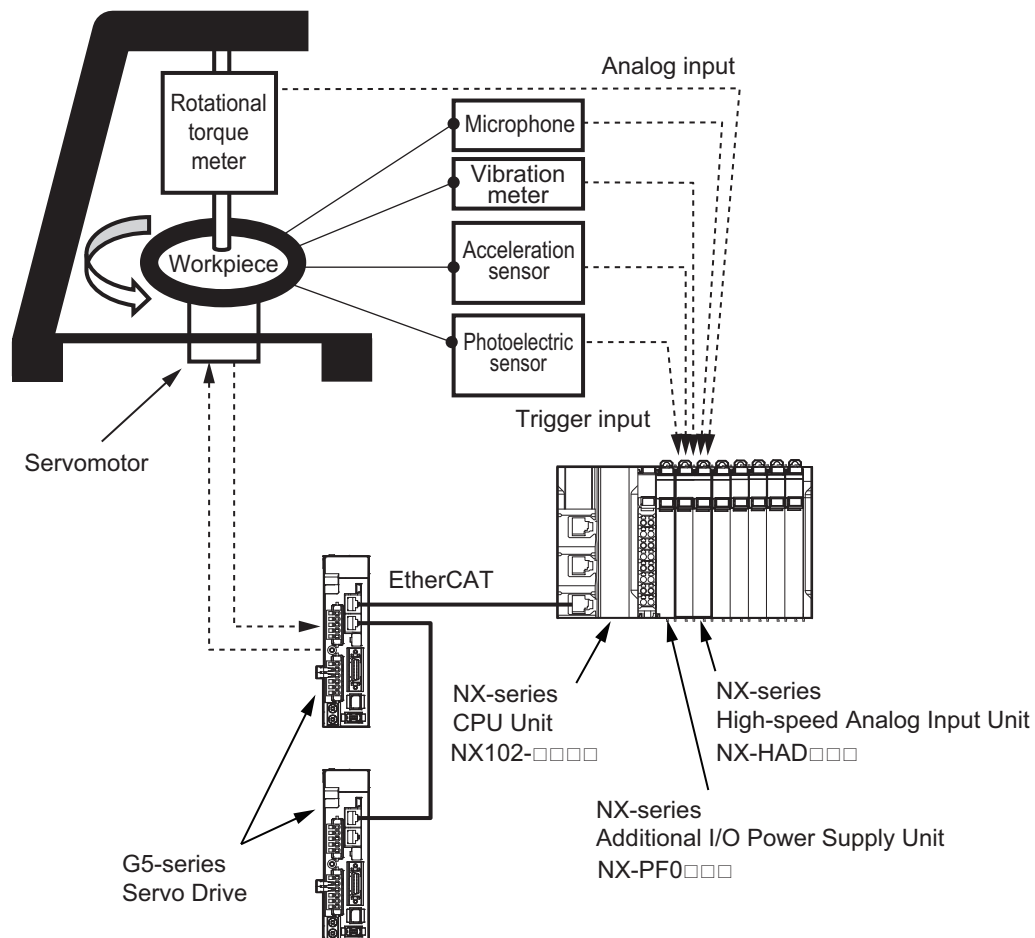
Furthermore, various functions necessary for product inspections in the production process, such as calculation of the feature amount, pass-fail judgment, etc. of the recorded data, are provided.

### 2-1-1 System Configuration Example

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

Incorporate the analog input signals of sounds, vibrations, and torque values generated during tests into the NX-series High Speed Analog Input Unit, and use the library FB/FUN to perform workpiece pass/fail judgments.

You can use the input from a photoelectric sensor, etc. as trigger input to easily obtain the analog input data necessary for tests.



### 2-1-2 Library Configuration

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

These can be used either at the same time, or as separate respective units.

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

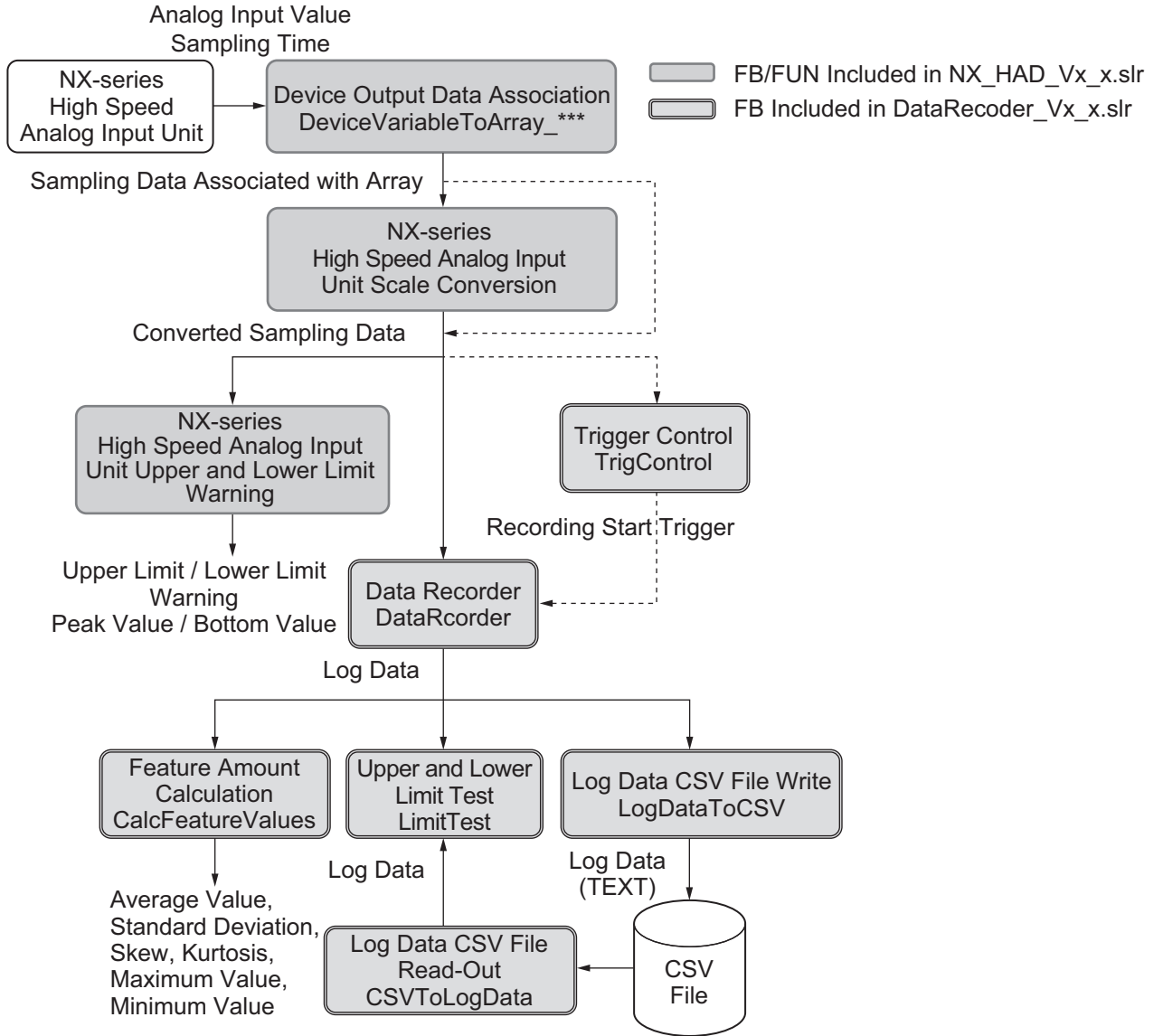
\*1. Vx\_x shows the library file version.

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

The data flow and structure when using both library files are described below.

- When logging analog input values in a time series, if you want to perform calculations, tests, file recording, etc., of the logging data, use the two NX\_HAD.slr and DataRecorder.slr files.
- When you want to perform alarm judgment of the analog input values for 1 task period or detect only the peak and bottom values, use only the NX\_HAD.slr file.

## 2 High Speed Analog Test Library





# 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

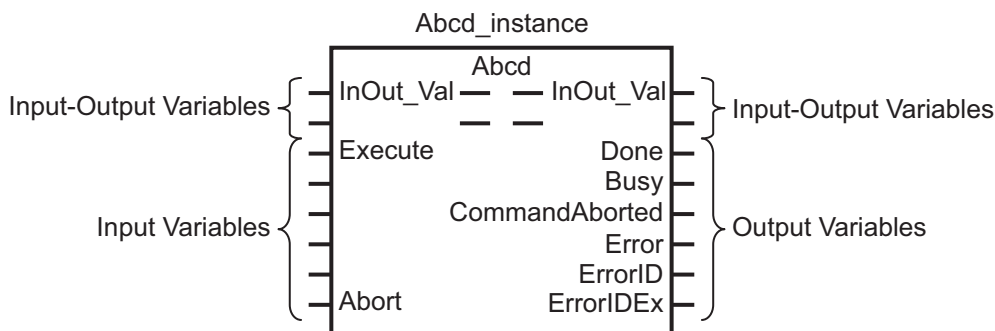
Common input variables and output variables used in functions and function blocks are as follows.

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. It is FALSE when the processing is not in progress.
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.



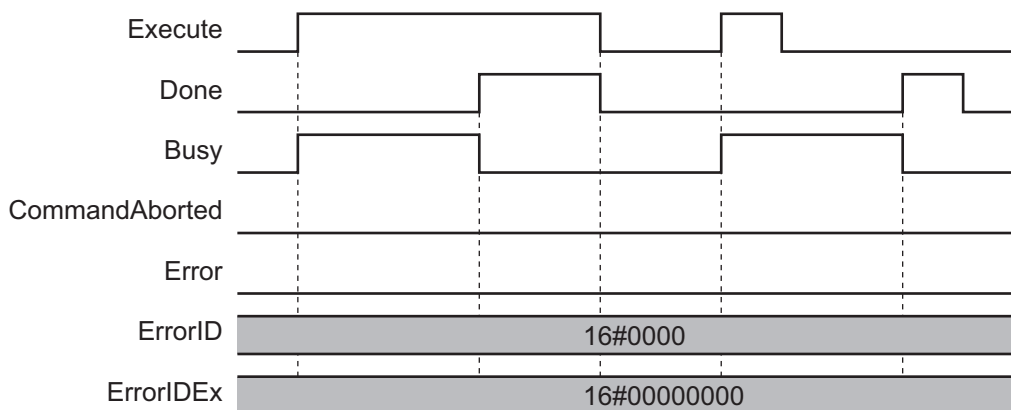
## 3 Common Specifications of Function Blocks

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

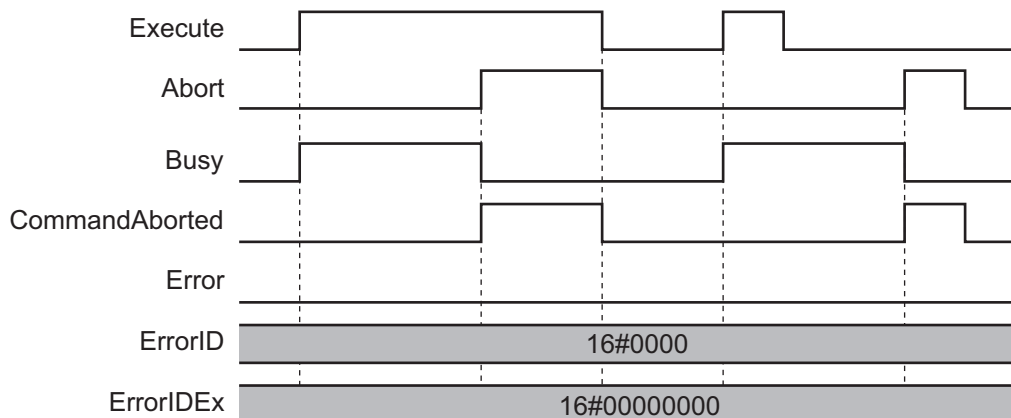
### Timing Chart

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

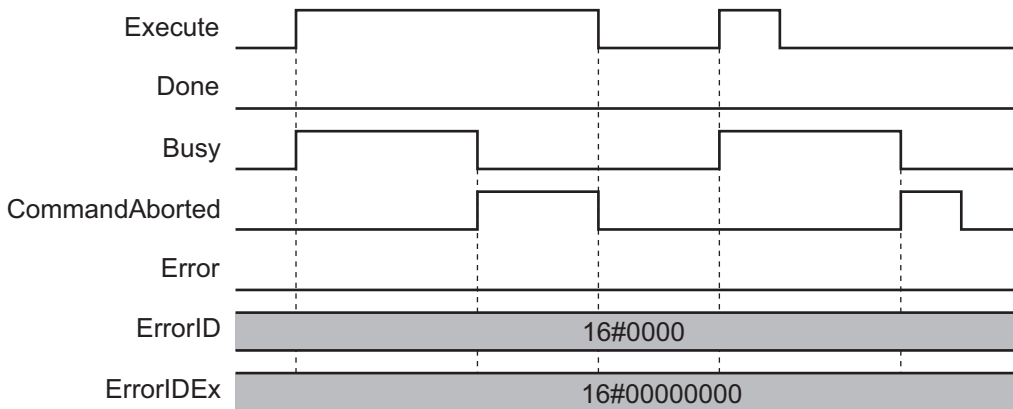
#### ● Normal End



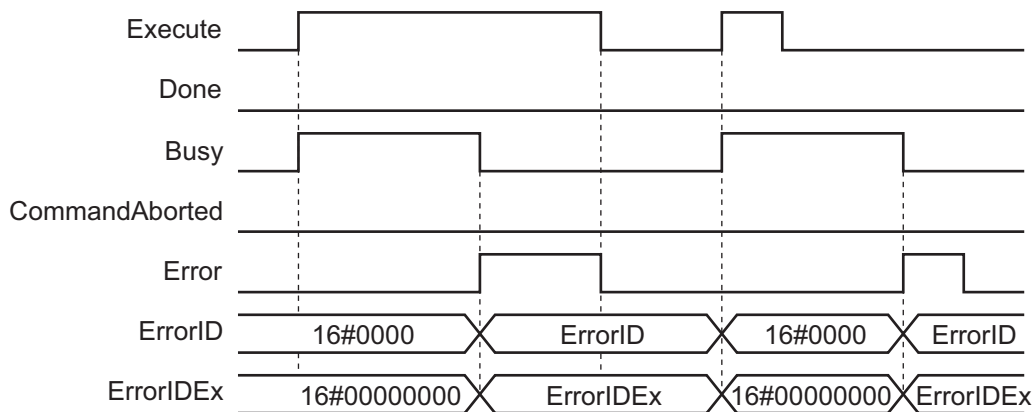
#### ● Canceled Execution



● **Aborted Execution**

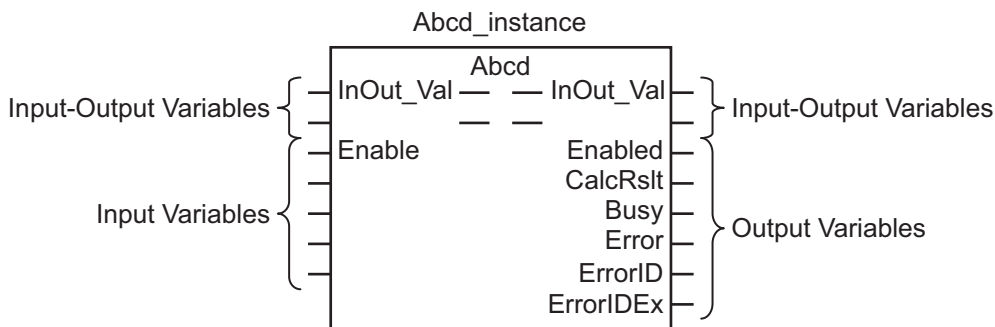


● **Errors**



**3-1-3 Enable-type Function Blocks**

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



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

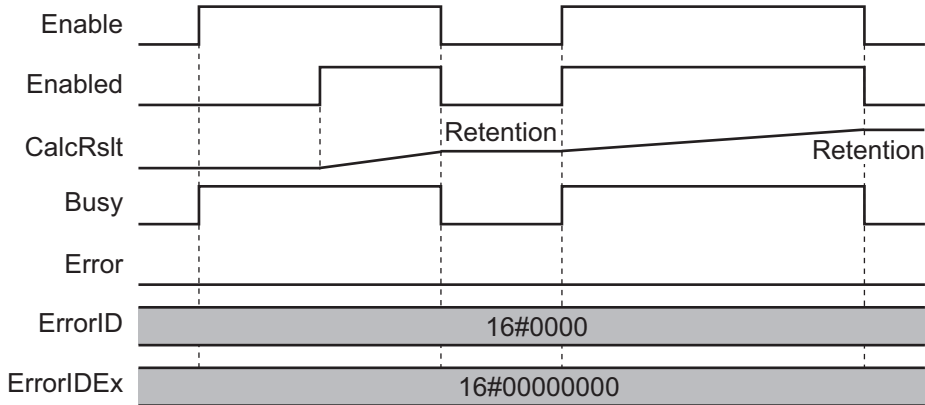
### 3 Common Specifications of Function Blocks

- 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.

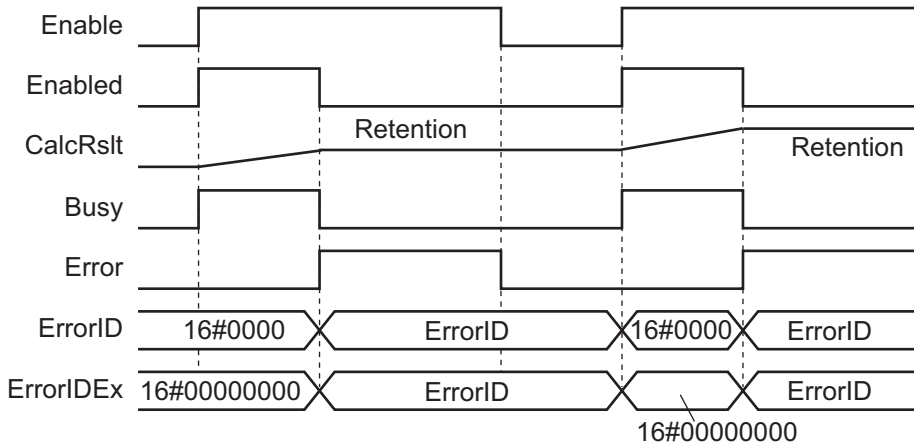
## Timing Charts

This section provides timing charts for a normal end, aborted execution, 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 (W501)* or *NY-series IPC Machine Controller Industrial Panel PC / Industrial Box PC Software User's Manual (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 (W507)* or *NY-series IPC Machine Controller Industrial Panel PC / Industrial Box PC Motion Control User's Manual (W559)* for details on re-execution.

## 3 Common Specifications of Function Blocks



# FB/FUN Individual Specifications (NX\_HAD)

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

---

<b>DeviceVariableToArray_***</b> .....	<b>4 - 2</b>
<b>ScaleTrans_HAD</b> .....	<b>4 - 11</b>
<b>LimitAlarm_HAD</b> .....	<b>4 - 14</b>

# DeviceVariableToArray\_\*\*\*

Bind the analog input values for 1 task period imported from the NX-series High Speed Analog Input Unit to a single array variable.

The FB/FUN end\_\*\*\* shows the maximum value of the FUN input sampling number with either 020 or 100. Select the FUN most suitable for the required sampling number.

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

## 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

	Meaning	Data type	Description	Valid range	Unit	Default
Input01[] - Input10[]	Device Output Data 01 to 10	ARRAY[0..9] OF INT	Inputs the <i>Analog Input Value</i> acquired from the I/O data	—	—	—
StartPos	Start position	UINT	Specifies which <i>DataArray[]</i> element No. the device output data is bound from. Refer to <i>Function</i> on page 4 - 3 for details.	Depends on data type.	—	0
InputSize	Number of data	UINT	Inputs the <i>Sampling number</i> acquired from the I/O data	Depends on data type.	—	1

## Output Variables

	Meaning	Data type	Description	Valid range	Unit	Default
ENO	Binding Results	BOOL	TRUE: Normal end FALSE: Error end, or execution condition not met.	TRUE, FALSE	—	—
NextPos	Next Position	UINT	Output the next bound start element No. 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>	Bound Data	ARRAY[*] OF REAL	Stores the bound input data	Depends on data type.	—	—

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

## Function

Since the analog values obtained from the NX-series High Speed Analog Input Unit are single or multiple ARRAY[0..9] OF INT type arrays, bind these to a single REAL type array.

Bind the data specified in *InputSize* in order from *Input01[0]*.

When the data has been successfully bound, output TRUE to *ENO*.

In the cases below, output FALSE to *ENO* without binding to an array.

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

Example: When the number of *DataArray[]* array elements is 100 and *InputSize* is 95, bind until *Input10[4]*.



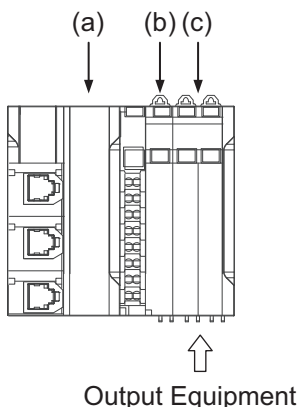
## Sample Programming 1

The NX-series High Speed Analog Input Unit binds the analog input values sampled 100 times per 1 task period to a single array variable. Furthermore, this program performs a scale conversion of the bound data, and monitors the upper and lower limits of the data.

This is an example of the connection of the NX-series High Speed Analog Input Unit to an NX-series CPU Unit.

### System Configuration

The system configuration is as shown below.



Letter	Description	Model	Description
(a)	NX-series CPU Units	NX102-□□□□	—
(b)	I/O Power Additional Supply Unit	NX-PF0730	• NX Unit No.: 1
(c)	NX-series High Speed Analog 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 *NX-series Analog I/O Unit High Speed Analog Input Unit User's Manual (SBCA-461)* for the setting method.

Setting Item	Setting Value	Setting Meaning
Ch1 Valid / Invalid	TRUE	Ch1 Valid
Ch1 Range Setting	0	-10 to +10 V
Ch1 Sampling Setting	100	100 times

### Program

- External Variables

Name	Data type	Constant	Comment
NXBus_N2_NX_Unit_I_O_Data_Active_Status	BOOL		

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

Name	Data type	Constant	Comment
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 for 1 Task Period

- Internal Variables

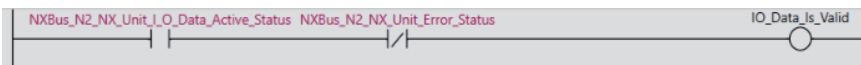
Name	Data type	Default	AT	Retention	Constant	Comment
LimitAlarm_instance	OmronLib\NX_HAD\LimitAlarm_HAD					
IO_Data_Is_Valid	BOOL					Set to TRUE when the input value from the NX-series High Speed Analog Unit is normal
Success_Concatenate	BOOL					
Alarm	BOOL					Set to TRUE when any of QHH, QH, QL, 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's value is TRUE, scale conversion and alarm judgment are executed

- Task Setting

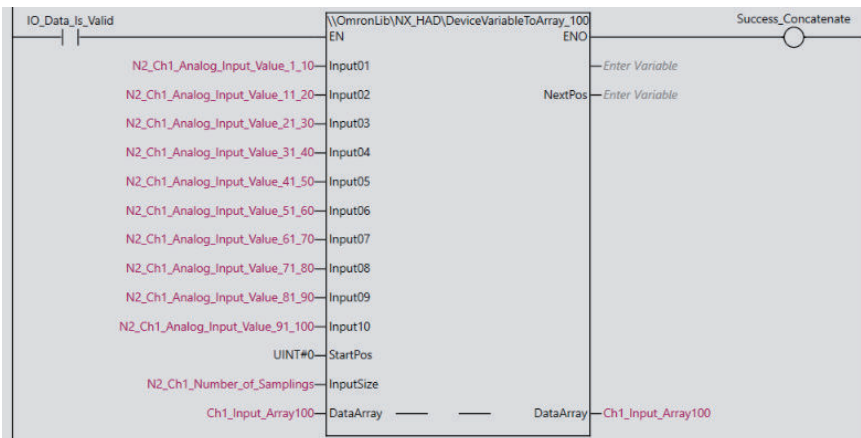
Deploy in the primary periodic task.

• LD Program

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

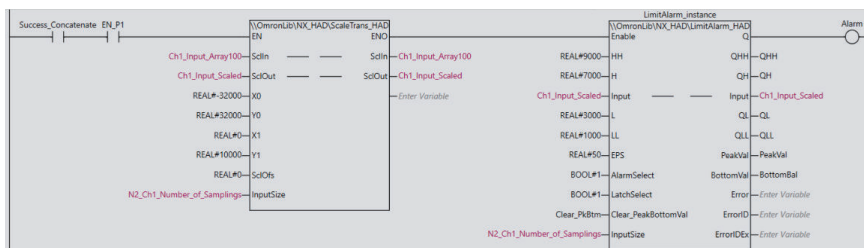


(2) Bind the analog input values for 1 task period to the single variable *Ch1\_Input\_Array100[]*.



(3) After scale conversion of the analog input values for 1 task period, and when the input values are outside the range set in *HH*, *H*, *L*, and *LL*, set *Alarm* to TRUE.

Scale Conversion Setting	-32,000 to +32,000 → 0 to 10,000	
Alarm Settings	Top Upper Limit <i>HH</i> Alarm	When 9,000 is exceeded
	Upper Limit <i>H</i> Alarm	When 7,000 is exceeded
	Lower Limit <i>L</i> Alarm	When less than 3,000
	Bottom Lower Limit <i>LL</i> Alarm	When less than 1,000



DeviceVariableToArray\_\*\*\*

Sample Programming 2

## Sample Programming 2

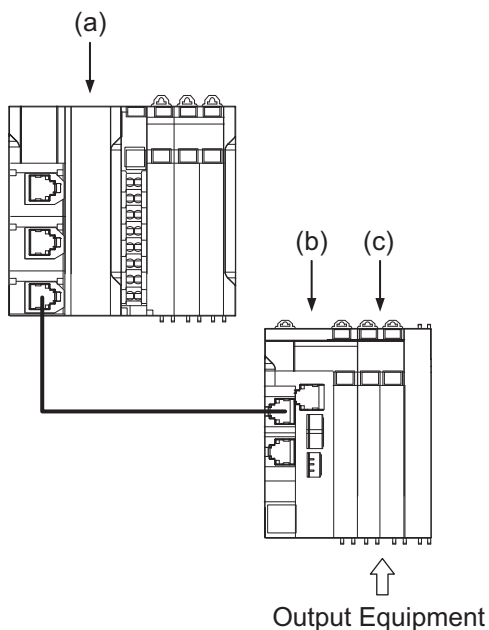
The NX-series High Speed Analog Input Unit binds the analog input values sampled 100 times per 1 task period to a single array variable. Furthermore, this program performs a scale conversion of the bound data, and monitors the upper and lower limits of the data.

This is an example of the connection of the NX-series High Speed Analog Input Unit to an EtherCat slave terminal.

## System Configuration

The system configuration is as shown below.

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



Letter	Description	Model	Description
(a)	NX-series CPU Units	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 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 *NX-series Analog I/O Unit High Speed Analog Input Unit User's Manual (SBCA-461)* for the setting method.

Setting Item	Setting Value	Setting Meaning
Ch1 Valid / Invalid	TRUE	Ch1 Valid
Ch1 Range Setting	0	-10 to +10 V
Ch1 Sampling Setting	200	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		
N1_Ch1_Analog_Input_Value_21_30	ARRAY[0..9] OF INT		

Name	Data type	Constant	Comment
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 for 1 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 the input value from the NX-series High Speed Analog Unit is 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's value is TRUE, scale conversion and alarm judgment are executed

• Task Setting

Deploy in the primary periodic task.

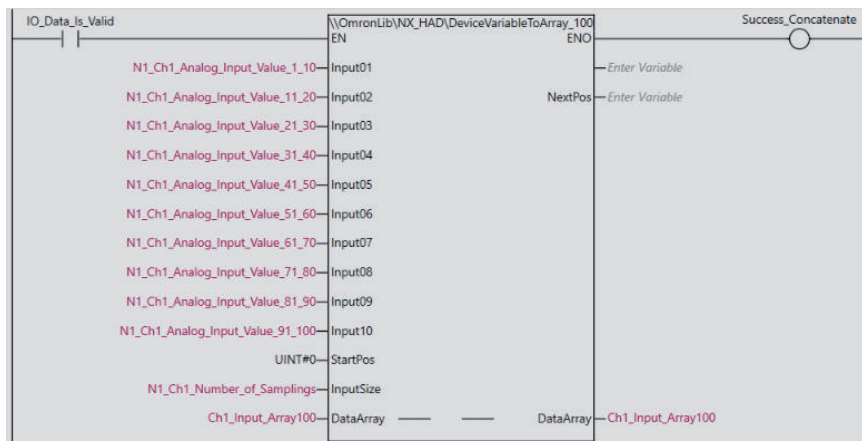
• LD Program

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



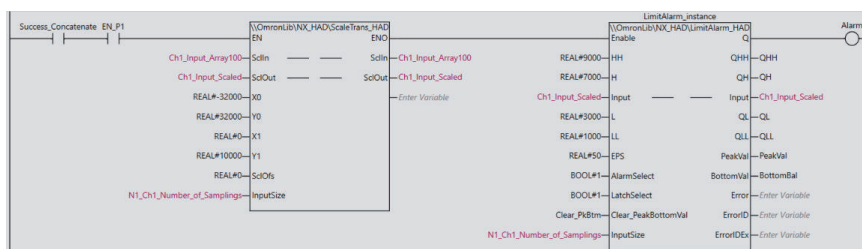
## 4 FB/FUN Individual Specifications (NX\_HAD)

(2) Bind the analog input values for 1 task period to the single variable *Ch1\_Input\_Array100*.



(3) After scale conversion of the analog input values for 1 task period, and when the input values are outside the range set in *HH*, *H*, *L*, and *LL*, set *Alarm* to TRUE.

Scale Conversion Setting	-32,000 to +32,000 → 0 to 10,000	
Alarm Settings	Top Upper Limit <i>HH</i> Alarm	When 9,000 is exceeded
	Upper Limit <i>H</i> Alarm	When 7,000 is exceeded
	Lower Limit <i>L</i> Alarm	When less than 3,000
	Bottom Lower Limit <i>LL</i> Alarm	When 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

Perform scale conversion of data imported from the NX-series High Speed Analog Input Unit.

FB/FUN name	Name	FB/FUN	Graphic expression	ST expression
Scale-Trans_HAD	For NX-series High Speed Analog Input Unit Scale Conversion	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 );

ScaleTrans\_HAD

4

Library Information

## 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
Y1	Output Range Upper Limit Value	REAL	Upper limit value of output range	Depends on data type.	—	32000

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

	Meaning	Data type	Description	Valid range	Unit	Default
ScIOfs	Offset Value	REAL	Offset value exceeding the output value	Depends on data type.	—	0.0
InputSize	Number of data	UINT	Inputs the <i>Sampling number</i> acquired from the I/O data	1 to 1000	—	1

## Output Variables

	Meaning	Data type	Description	Valid range	Unit	Default
ENO	Conversion 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
ScIn[] <sup>*1</sup>	Input Value	ARRAY[*] OF REAL	Data array targeted for conversion	Depends on data type.	—	—
ScOut[] <sup>*2</sup>	Output Value	ARRAY[*] OF REAL	Data array after scale conversion	Depends on data type.	—	—

- \*1. The number of array elements is arbitrary. However, the number of array elements must be equal to the *InputSize* or more. In addition, the array element start number can be either 0 or a number other than 0.
- \*2. The number of array elements is arbitrary. However, the number of array elements must be the same as *ScIn[]*. In addition, the array element start number can be either 0 or a number other than 0. Even if the start number for the array elements differs from *ScIn[]*, it can still be executed.

## Function

Convert each of the *ScIn[]* elements from the input range specified in *X0* and *X1* to the output range specified from *Y0* to *Y1*.

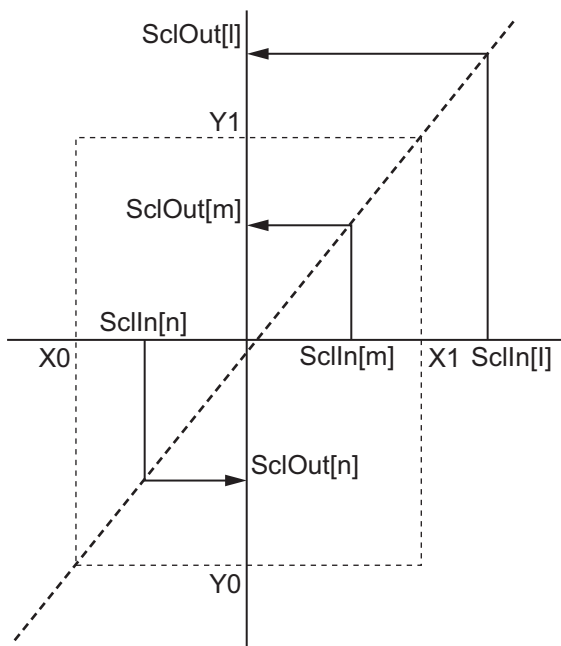
Store the sum of the value converted to the output range and *ScIOfs* in *ScOut[]*.

You can perform execution even if the input value is outside the input range.

When conversion is successful, output TRUE to *ENO*.

In the cases below, output FALSE to *ENO* without performing scale conversion.

- When the number of *ScIn[]* and *ScOut[]* array elements is less than the *InputSize*
- When *ScIn[]* and *ScOut[]* array element numbers do not match  
However, differences in the Start No. are possible
- When *X0* and *X1* are equal, or when *Y0* and *Y1* are equal
- When *InputSize* is 0 or at 1001 or more



$$ScOut[n] = \frac{Y1-Y0}{X1-X0} X (ScIn[n]-X0) + Y0 + ScOfs$$

The range of n is from the first element number of "ScIn[]" to the "InputSize"-1 for the first element number.

### Sample Programming

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

# LimitAlarm\_HAD

Perform the top upper limit, upper limit, lower limit, and bottom lower limit alarms for the data imported from the NX-series High Speed Analog Input Unit.

FB/FUN name	Name	FB/FUN	Graphic expression	ST expression
LimitAlarm_HAD	For NX-series High Speed Analog Input Unit Upper and Lower Limit Alarm	FB		<pre>LimitAlarm_HAD_instance( Enable, HH, H, Input, L, LL, EPS, AlarmSelect, LatchSelect, Clear_PeakBottomVal, BottomVal, Error, 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>Sampling number</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 <i>QHH</i> , <i>QH</i> , <i>QL</i> , or <i>QLL</i> are set to TRUE FALSE: All of <i>QHH</i> , <i>QH</i> , <i>QL</i> , and <i>QLL</i> 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	—	—

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

	Meaning	Data type	Description	Valid range	Unit	Default
PeakVal	Peak Value	REAL	Maximum Input Value	Depends on data type.	—	—
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 Unit Hardware 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 targeted for monitoring	Depends on data type.	—	—

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

## Function

While *Enable* is TRUE, when each element value in *Input[]* exceeds the *HH* or *H* setting value, or when they fall short of the *L* or *LL* setting value, output an alarm.

The relationship of the input and each setting value to the 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 *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 *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 *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 *LL+EPS*, set *QLL* to FALSE.
- When any of *QHH*, *QH*, *QL*, or *QLL* is TRUE, set *Q* to TRUE.

If *AlarmSelect* is TRUE, set the four *HH*, *H*, *L*, and *LL* to alarm judgment.

If *AlarmSelect* is FALSE, set the two *H* and *L* to alarm judgment. In this case, *QHH* and *QLL* are always FALSE.

Perform alarm judgment for the element number import values specified in *InputSize*. This may cause more than one of *QHH*, *QH*, *QL*, and *QLL* to become TRUE at the same time. Refer to below for details.

In the cases below, output TRUE to *Error*, and end monitoring.

- When the number of *Input[]* array elements is less than the *InputSize*
- When each alarm setting value has the relationship shown below  
 $LL \leq L \leq H \leq HH$
- When *EPS* is a non-number or  $\pm\infty$
- When *InputSize* is 0 or at 1001 or more

Output the maximum input value from execution in progress or execution start up to the present to *PeakVal*, and the minimum input value to *BottomVal*.

When *Clear\_PeakBottomVal* is TRUE, output 0 to *PeakVal* and *BottomVal*. When

*Clear\_PeakBottomVal* continues to be TRUE, continually output 0 to *PeakVal* and *BottomVal*.

If *Clear\_PeakBottomVal* is set to FALSE, in the later input values output the maximum value and minimum value.

If *Enable* is set to FALSE, end the FUN execution.

- *Q*, *QHH*, *QH*, *QL*, and *QLL* are all set to FALSE.
- Maintain the *PeakVal* and *BottomVal* until *Enable* is again set to TRUE.

• About Alarm Output/Reset

The alarm output/reset operation is determined by *LatchSelect*.

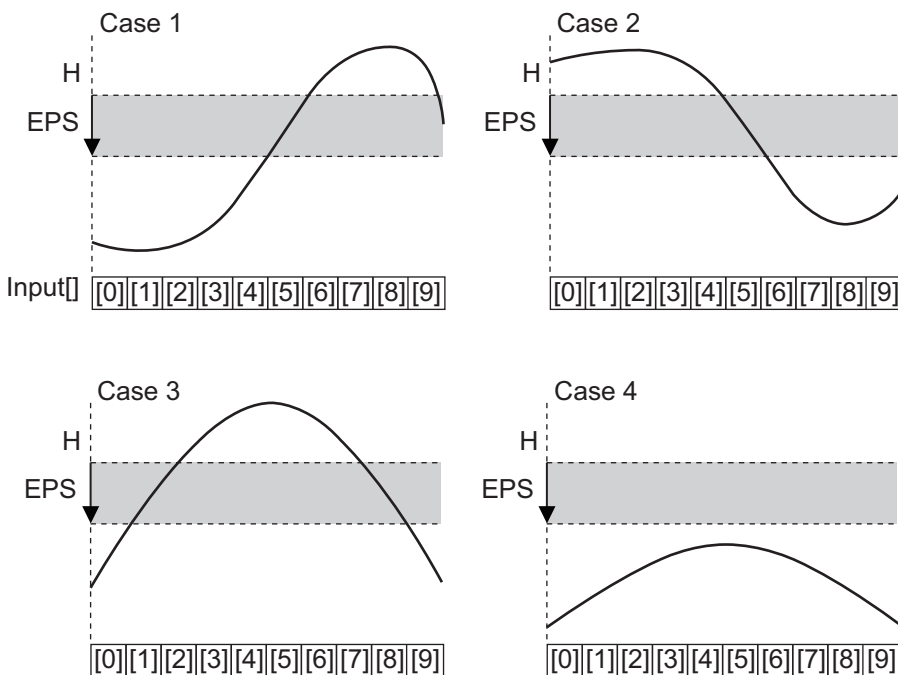
a) When *LatchSelect* is FALSE

In *Input[]*, set alarm judgment for the final element in the element number specified in *InputSize*.

b) When *LatchSelect* is TRUE

In *Input[]*, set alarm judgment for all elements specified in *InputSize*.

As an example, the *QH* operation when the input change in 1 task period of the NX-series High Speed Analog Input Unit is stored in *Input[]* is shown.



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

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

- Setting changes during execution

Constantly reflect *HH*, *H*, *L*, *LL*, and *EPS* while execution is in progress.

Even if changed during execution, *AlarmSelect* and *LatchSelect* will not be reflected.

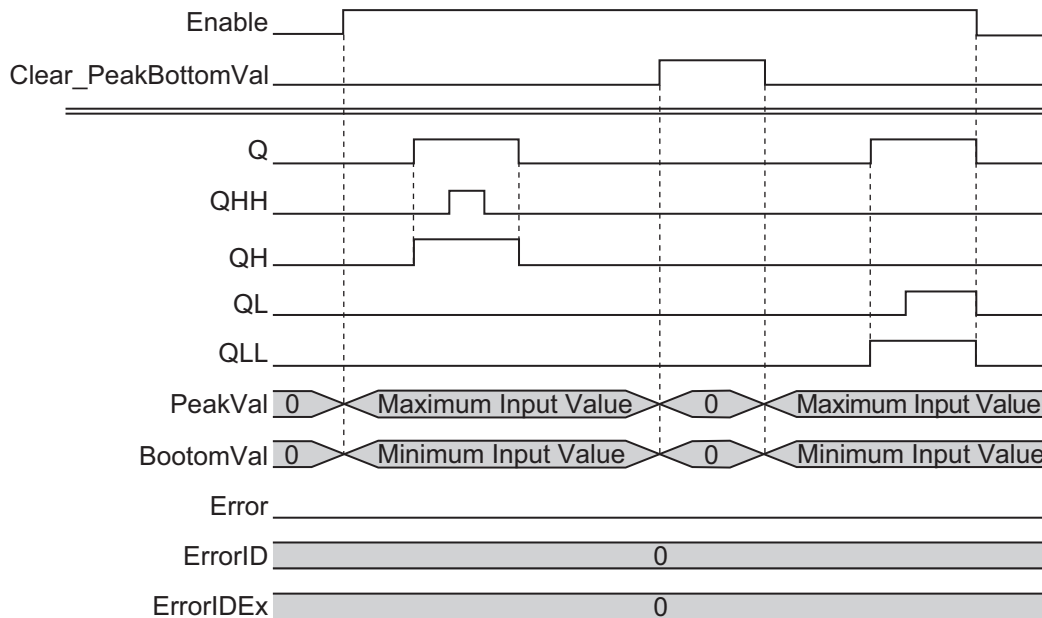
If changed during execution, set the alarm output to FALSE, and perform alarm judgment at the setting value after the change.

### Timing Chart

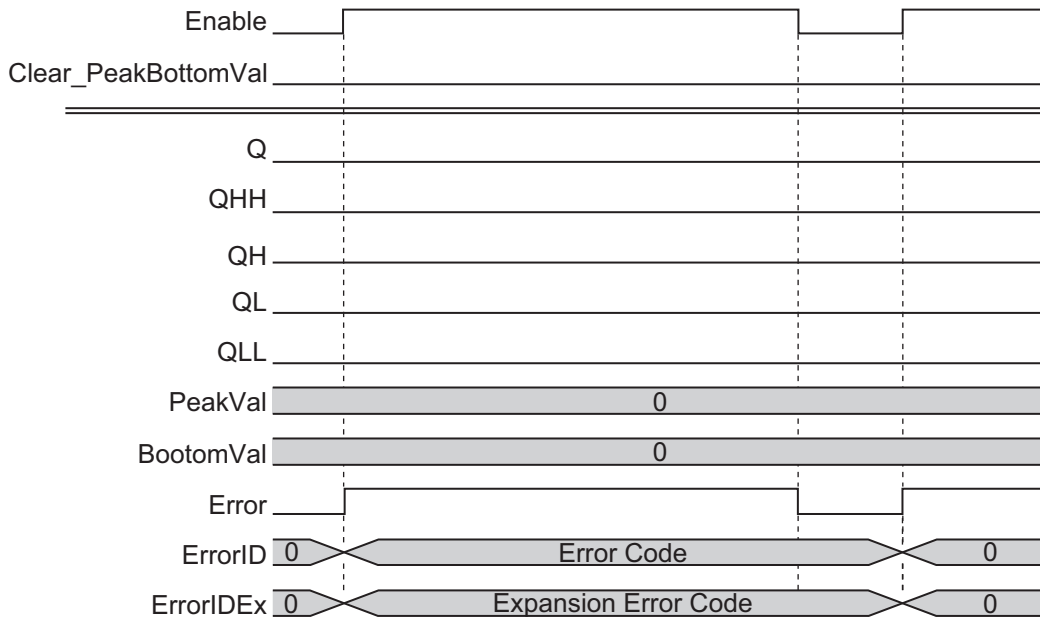
The timing charts are shown below.

- If *Enable* is set to TRUE, start the alarm monitoring.  
Output the maximum input value from execution start up to the present to *PeakVal*, and the minimum input value to *BottomVal*.
- If *Enable* is set to FALSE, end the alarm monitoring.
- While *Clear\_PeakBottomVal* is TRUE, output 0 to *PeakVal* and *BottomVal*.
- If an error occurs, set *Error* to TRUE. At the same time, output *ErrorID* and *ErrorIDEx*.
- When an error occurs, retain *Error* while *Enable* is TRUE.  
After *Enable* is set to FALSE, retain *ErrorID* and *ErrorIDEx* until *Enable* is again set 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#3D00	16#00000001	Illegal Data Number	When the number of <i>Input[]</i> array elements is less than the <i>InputSize</i>	Revise the number of <i>Input[]</i> array elements.
	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 such that $LL \leq L \leq H \leq HH$ .
	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.

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



# FB/FUN Individual Specifications (DataRecorder)

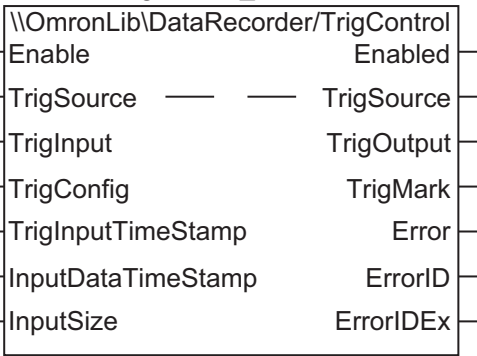
This section describes the FB/FUN individual specifications included in DataRecorder.slr.

---

<b>TrigControl .....</b>	<b>5 - 2</b>
<b>DataRecorder .....</b>	<b>5 - 12</b>
<b>LimitTest.....</b>	<b>5 - 19</b>
<b>CalcFeatureValues .....</b>	<b>5 - 25</b>
<b>LogDataToCSV.....</b>	<b>5 - 34</b>
<b>CSVToLogData.....</b>	<b>5 - 41</b>

# TrigControl

Generate the trigger information to start the log data recording by the DataRecorder FB.

FB/FUN name	Name	FB/FUN	Graphic expression	ST expression
TrigControl	Trigger Control	FB	 <p>The graphic expression shows a box labeled 'TrigControl_instance' with the following inputs and outputs:</p> <ul style="list-style-type: none"> <li>Enable: Enabled</li> <li>TrigSource: TrigSource</li> <li>TrigInput: TrigOutput</li> <li>TrigConfig: TrigMark</li> <li>TrigInputTimeStamp: Error</li> <li>InputDataTimeStamp: ErrorID</li> <li>InputSize: ErrorIDEx</li> </ul>	<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. Trigger mode is valid for <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. Trigger mode is valid for <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>Sampling number</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 Nos. 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 Unit Hardware 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
TrigSource[]*1	Input Value for Trigger Judgment	ARRAY[*] OF REAL	Input the data array targeted for trigger condition judgment	—	—	—

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

## 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 direction of change to judge the trigger.</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. You can more accurately obtain the time established by the trigger conditions than with the digital input trigger mode. The trigger input uses the trigger input function of the NX-series High Speed Analog Input Unit, or the NX-series Digital Input Unit, with a change time input method.</li> </ul>
TrigSlope	BOOL	Specifies the <i>TrigSource[]</i> direction of change to detect the trigger. 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 No. 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, judge whether the trigger conditions specified in *TrigConfig* are satisfied.

When the trigger conditions are met, set *TrigOutput* to TRUE, and output the *TrigSource[]* element Nos. and array elements that meet the trigger conditions established in *TrigMark*.

Once the trigger conditions have been met, retain the *TrigOutput* and *TrigMark* while *Enable* is TRUE. If restarted, set *Enable* to FALSE for 1 task period or more.

When *Enable* is set to FALSE, end the trigger judgment, and set *Enabled* and *TrigOutput* to FALSE.

In addition, for *TrigMark.Pos* and *TrigMark.Data*, output 0.

In the cases below, set *Error* to TRUE, and end the trigger judgment.

- 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 at 1001 or more
- When the function block was executed in an event task

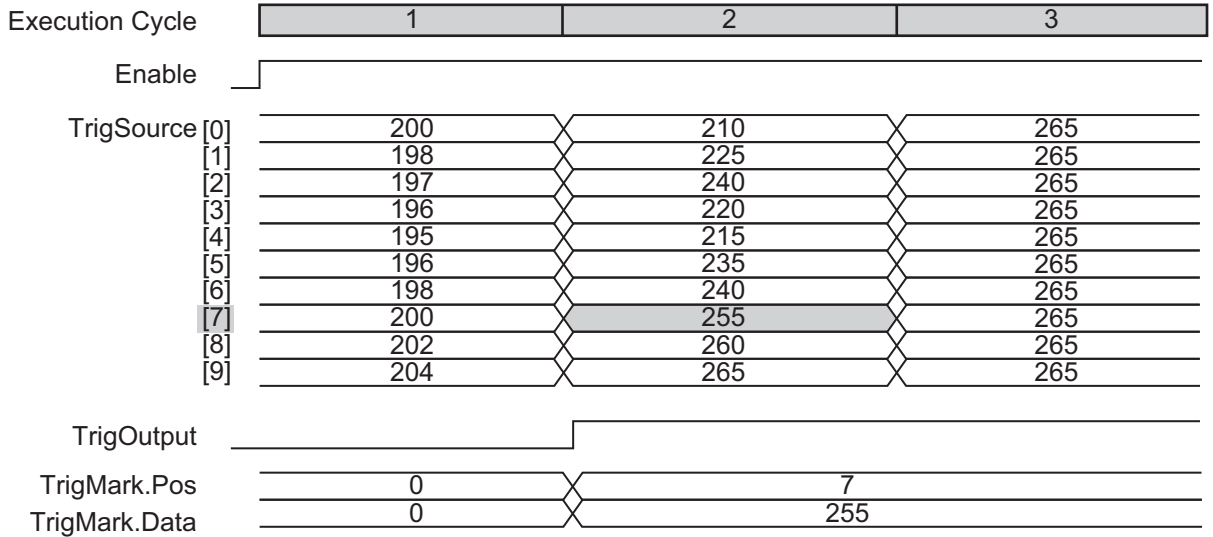
The trigger mode setting and action are as follows.

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

When *TrigConfig.TrigMode* is 0, use *TrigConfig.TrigSlope* and *TrigConfig.AnalogTrigLevel* in the trigger conditions.

When each *TrigSource[]* element specified in *InputSize* crosses the *TrigConfig.AnalogTrigLevel*, judge the trigger conditions to be met. At this time, set *TrigOutput* to TRUE, and output the *TrigSource[]* element Nos. and array element values that meet the trigger conditions to *TrigMark*.

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

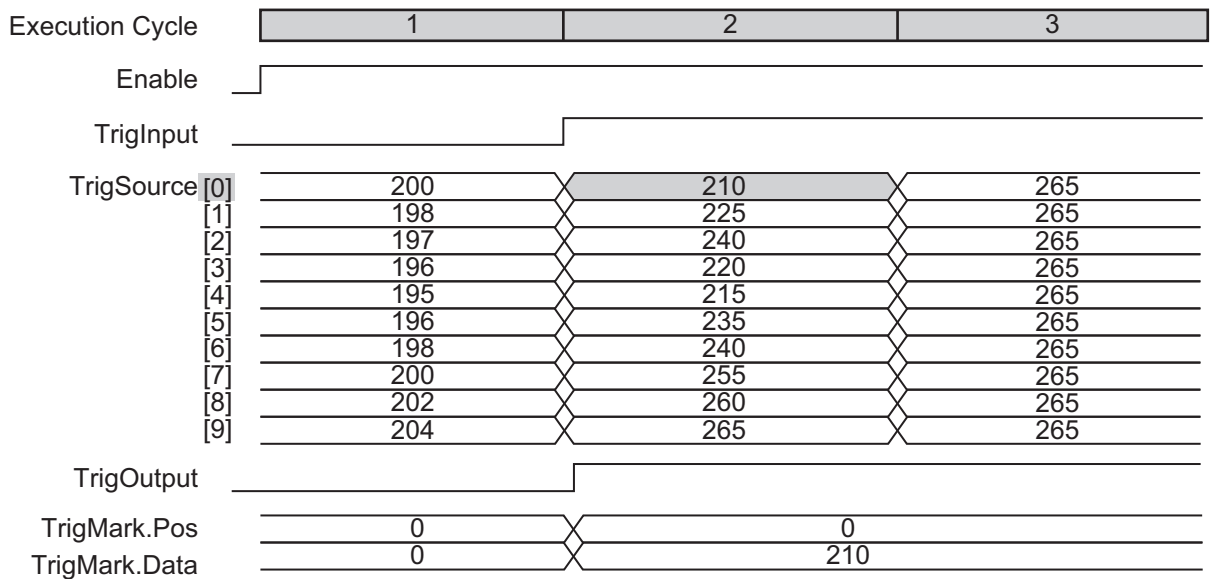


- Digital Input Trigger Mode: *TrigConfig.TrigMode* is 1

When *TrigInput* is set to TRUE, set *TrigOutput* to TRUE, and output the *TrigSource[]* first element No. and its array element value to *TrigMark*.

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

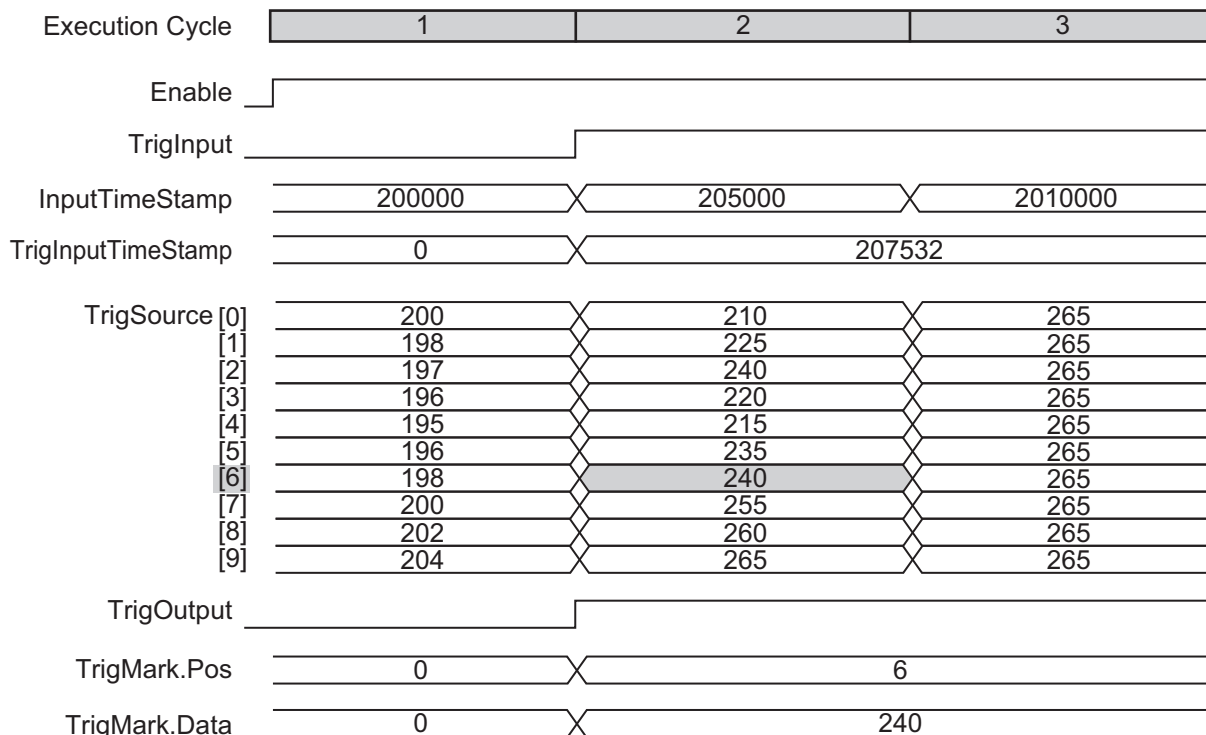
Example: When *TrigConfig.TrigMode* is 1, and *TrigSource[0..9]* and *InputSize* are 10



- Time Stamp Trigger Mode: *TrigConfig.TrigMode* is 2

## 5 FB/FUN Individual Specifications (DataRecorder)

When *TrigInput* is set to TRUE, set the trigger output to TRUE. At this time, specify the *TrigSource[]* array element No. immediately after trigger conditions are met from *InputDataTimeStamp* and *TrigInputTimeStamp*, and output the element No. and its array element value to *TrigMark*. Do not use *TrigConfig.TrigSlope* and *TrigConfig.AnalogTrigLevel* for the judgment conditions.



- Setting changes during execution

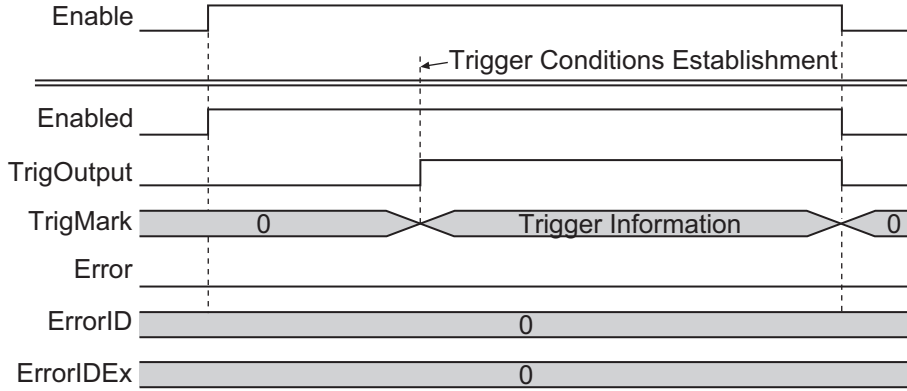
If *TrigConfig* has been changed while execution is in progress, use the changed settings to perform trigger judgment.

However, changes after trigger conditions are met (when *TrigOutput* is set to TRUE) are not reflected.

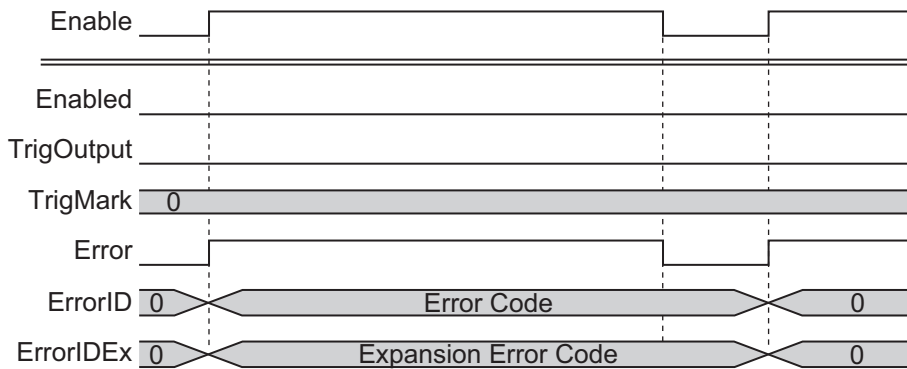
## Timing Chart

The timing charts are shown below.

- If *Enable* is set to TRUE, start the trigger judgment.
- If *Enable* is set to FALSE, end the trigger judgment.
- When the trigger conditions are met, set *TrigOutput* to TRUE, and output the *TrigSource[]* element Nos. and array element values that meet the trigger conditions established in *TrigMark* as trigger information.
- If an error occurs, set *Error* to TRUE. At the same time, output *ErrorID* and *ErrorIDEx*.
- When an error occurs, retain *Error* while *Enable* is TRUE.  
After *Enable* is set to FALSE, retain *ErrorID* and *ErrorIDEx* until *Enable* is again set to TRUE.
- Timing Chart for Normal End



• Timing Chart for Error End



## Troubleshooting

Error code	Expansion error code	Status	Description	Corrective action
16#0000 0	16#00000000	Normal End	—	—
16#3D0 1	16#00000001	Illegal Element Number	The number of <i>TrigSource[]</i> array elements is less than the <i>InputSize</i>	Revise 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	Revise 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

The NX-series High Speed Analog Input Unit uses DeviceVariableToArray\_100 to bind the analog input values sampled 100 times per 1 task period to a single array variable. Furthermore, this is a program for recording the bound data with DataRecorder, and performing a comparison test of the test standard data using LimitTest.

This is an example of the connection of the NX-series High Speed Analog Input Unit to an NX-series CPU Unit.

## 5 FB/FUN Individual Specifications (DataRecorder)

This sample programming consists of two POU, a program 0 for executing in the primary periodic task, and a program 1 for operation in 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 for 1 Task Period
LogData	ARRAY[0..4999] OF OmronLib\DataRecorder\LogData		Recorder Data Array
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					

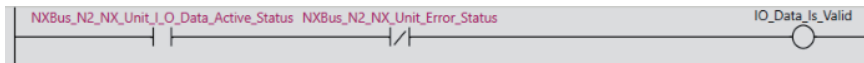
Name	Data type	Default	AT	Retention	Constant	Comment
RecordInfo	OmronLib\DataRecorder\RecordInfo					
LogEnable	BOOL					
Success_Concatenate	BOOL					

• Task Setting

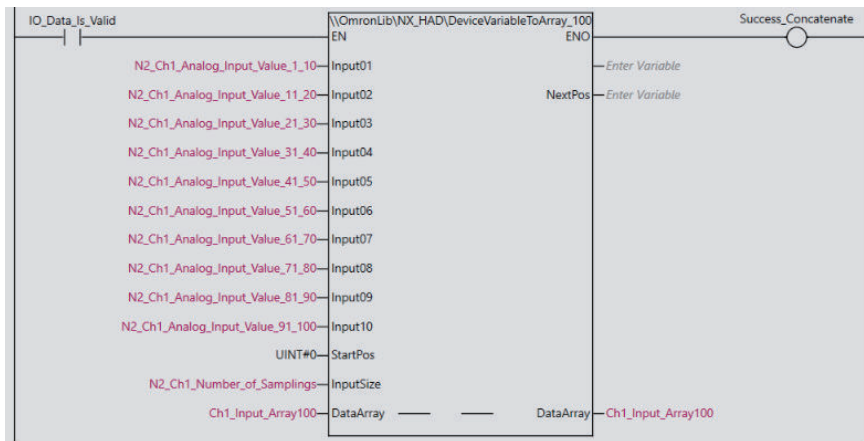
Deploy in the primary periodic task.

• LD Program

a) Check that normal data is being received from the NX-series High Speed Analog Input Unit. If connecting the NX-series High Speed Analog Input Unit to the EtherCAT slave terminal, refer to *Sample Programming 2* on page 4 - 7.



b) Bind the analog input values for 1 task period to the single variable *Ch1\_Input\_Array100[]*.



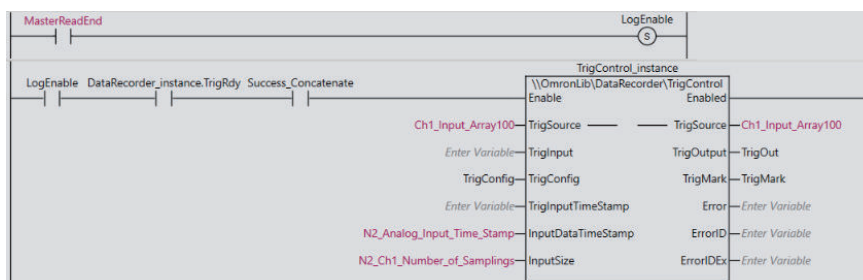
c) If executing the *Program 1 (Periodic Task 17)* on page 5 - 10 described below, and the master data CSV file readout is completed, the NX-series High Speed Analog Input Unit input value record enters a trigger wait state. Set the trigger conditions in *TrigConfig*.

• Trigger Conditions

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

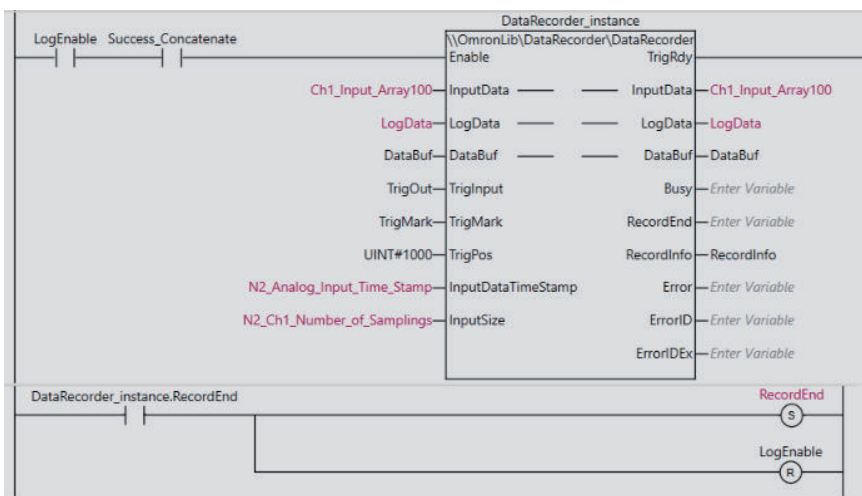
*TrigMode.TrigSlope=FALSE* (Startup)

*TrigMode.AnalogTrigLevel=REAL#100*



5 FB/FUN Individual Specifications (DataRecorder)

- d) If the trigger conditions are met, record the sampling data before trigger conditions were met with an input value of 1,000, and the sampling data after trigger conditions were met with a value of 4,000 in *LogData[]*.



### Program 1 (Periodic Task 17)

- External Variables

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

- 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\sLogData					
FirstOutOfLimitPos	UINT					
Start_P2	BOOL	FALSE				If the value of this variable is set to TRUE, execute the program.
Pass	BOOL					
Cancel	BOOL					

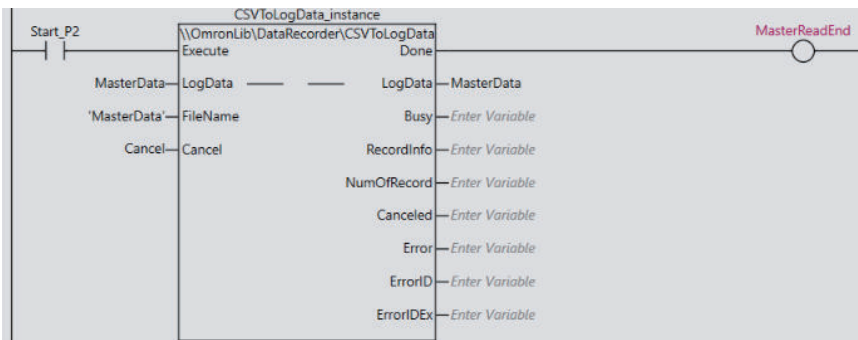
- Task Setting

Deploy 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 the test standard data.

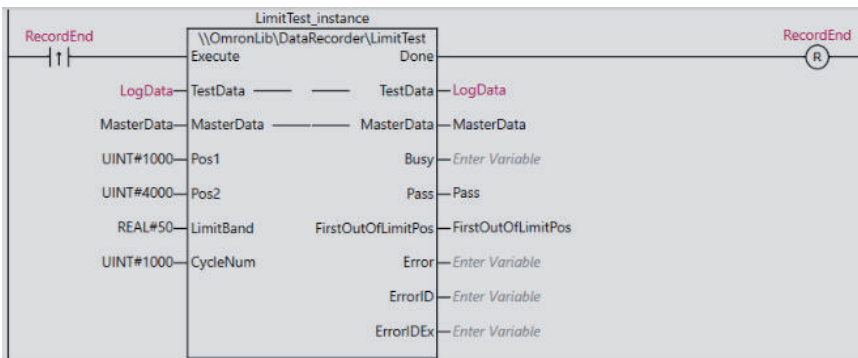
You will have to prepare the test standard data again.



- (2) Test the *LogData[0..4999]* recorded in *Program 0 (Primary Periodic Task)* on page 5 - 8 and the test standard data *MasterData[0..4999]* read out from the csv file.

- Test range: From element No. 1,000 to 4,000
- Pass standard: -50 to +50 for test standard data of the same element No.

Perform the test for 1,000 elements at a time per 1 periodic task. If the result is a pass when the test is completed, set *Pass* 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

Bind the data array for the specified number of elements to a single array variable for each task period, and create the time series log data.

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 Input the BOOL type variables, I/O data, or TrigControl FB <i>Trigger Output</i>	TRUE, FALSE	—	FALSE
TrigMark	Trigger Information	OmronLib \DataRecorder \sTrigMark	Input 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	During data recording, specifies to which element No. in the <i>LogData[]</i> the data during trigger input will be recorded	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>Sampling number</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\sRecordInfo	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 Unit Hardware 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	Input the recording source data array.	—	—	—

## 5 FB/FUN Individual Specifications (DataRecorder)

	Meaning	Data type	Description	Valid range	Unit	Default
LogData[]*2	Log Data	ARRAY[*] OF Omron- Lib\DataRe- corder \sLogData	Input the recording desti- nation data array. The number of array ele- ments is the number of records that can be re- corded	—	—	—
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 refer- encing.	—	—	—

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

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

## 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 for the first data item.
EndDateTime	DT	Record the date and time for the final data item. If the FB execution is ended during the recording, no recording is made.
TriggerDateTime	DT	Record the date and time input by <i>TrigInput</i> . 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 <i>LogData[]</i> element No. when <i>TrigInput</i> is set to TRUE.

- OmronLib\DataRecorder\sLogData

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

## Function

While *Enable* is TRUE, record the element No. data specified in *InputSize* from *InputData[]* into *DataBuf[]* for every task period.

After *Enable* is set to TRUE, and if the data specified with *TrigPos* has finished being recorded in *DataBuf[]*, set *TrigRdy* to TRUE.

When *TrigRdy* is TRUE, *TrigInput* is set to TRUE, and when *TrigRdy* is FALSE, record the *TrigPos* and later data in *LogData[]*. During this recording, set *Busy* to TRUE.

When recording is completed, link the *DataBuf[]* data up to *TrigPos* to *LogData[]* and output it, and then set *RecordEnd* to TRUE.

At the same time, output *RecordInfo*.

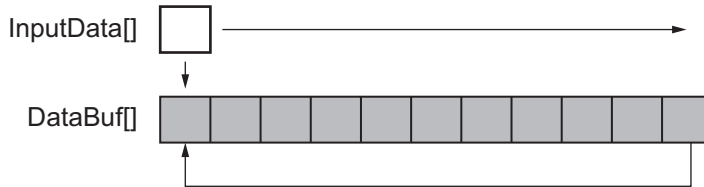
When *TrigRdy* is FALSE, ignore *TrigInput* even if set to TRUE.

If *Enable* is set to FALSE after recording is completed, end execution of this FB.

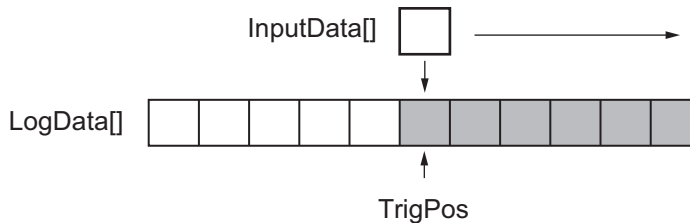
After *Enable* is set to FALSE, retain *RecordEnd* and *RecordInfo* until *Enable* is again set to TRUE.

• Data Storage Method

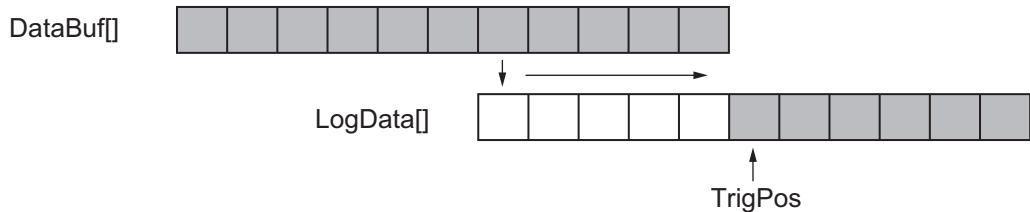
- (1) After *Enable* is set to TRUE and until *TrigInput* is set to TRUE, continue recording *InputData[]* to *DataBuf[]*.



- (2) When *TrigInput* is set to TRUE, record in order from *TrigPos* until the final element in *LogData[]*.



- (3) When recording up to the final element has ended, link the *DataBuf[]* data to the beginning element of *LogData[]* until the *TrigPos* - 1 element in the new order, and complete the log data from the beginning to end elements of *LogData[]*.



In the cases below, output TRUE to *Error*, and end the recording.

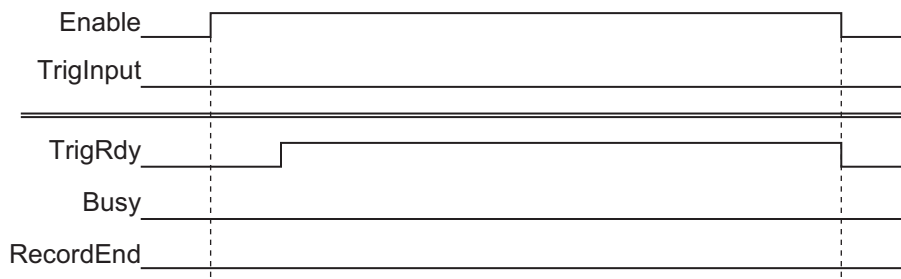
- a) When the number of *InputData[]* array elements is the *InputSize*
- b) When the number of *LogData[]* or *DataBuf[]* array elements is less than the number of *InputData[]* array elements
- c) When *TrigMark.Pos* is an element No. outside the *InputData[]* array range
- d) When *TrigPos* is outside the *LogData[]* or *DataBuf[]* range
- e) When *InputSize* is 0, or at 1001 or more
- f) When the function block was executed in an event task

Perform the *TrigInput* and *TrigMark* input as shown below.

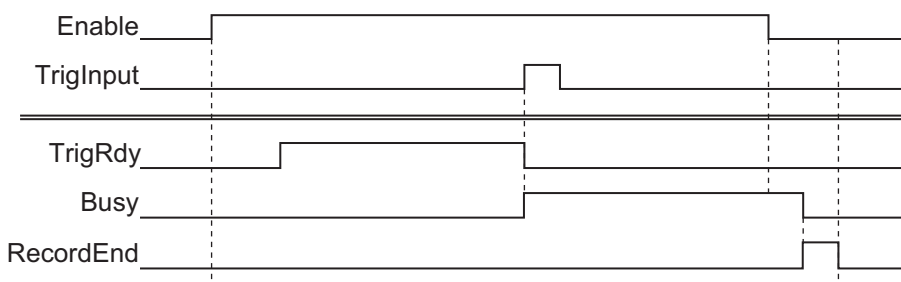
TrigInput	TrigMark
When inputting arbitrary variables	Variable is not assigned
When inputting the TrigControl FB TrigOutput	TrigControl FB TrigMark

## 5 FB/FUN Individual Specifications (DataRecorder)

- 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 *InputData[]* recording, and set *TrigRdy* to FALSE.  
*LogData[]* and *RecordInfo* are indefinite.



- b) Operation when *Enable* is set to FALSE after *Busy* is set to TRUE, and before *RecordEnd* is set to TRUE:  
Recording is done until the *InputData[]* 1 task period before *Enable* is set to FALSE. Recording is not done for the *InputData[]* after *Enable* is set to FALSE.  
When *Busy* is not immediately set to FALSE, it is set to TRUE until the *LogData[]* creation is completed (*RecordEnd* is set to TRUE).  
For *RecordEnd*, TRUE is output for 1 task period only.  
*RecordInfo.RecordNum* is the data from the beginning of the *LogData[]* array until 1 task period before *Enable* is set to FALSE.  
The *LogData[]* array element data for the array elements shown in *RecordInfo.RecordNum* and onward is indefinite.  
For the *RecordInfo.EndDate*, output the date and time when *Enable* was set to FALSE.



While *Busy* is TRUE, this FB cannot be restarted.

After *RecordEnd* is set to TRUE, set *Enable* to FALSE for 1 task period or more.

## Timing Chart

The timing charts are shown below.

- If *Enable* is set to TRUE, start the recording to the *DataBuff[]* of the *InputData[]*.
- If *Enable* is set to FALSE, end the recording.
- If *TrigInput* is set to TRUE, start recording to *LogData[]*.

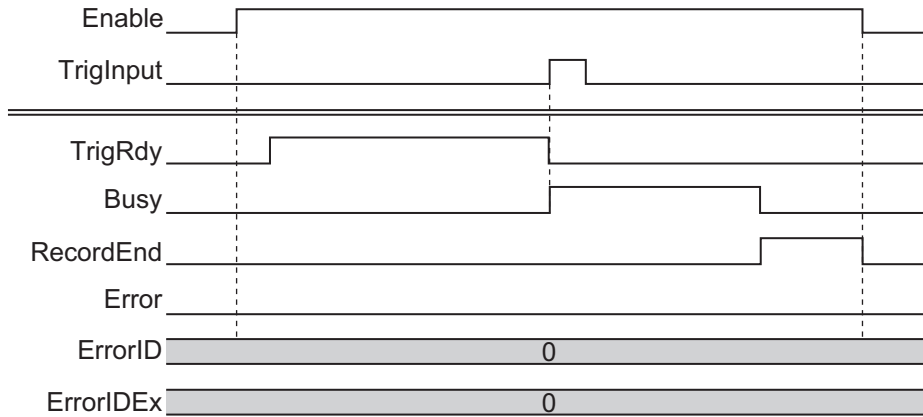
During recording, *Busy* is set to TRUE. When recording is complete, *RecordEnd* is set to TRUE.

If an error occurs, set *Error* to TRUE. At the same time, output *ErrorID* and *ErrorIDEx*.

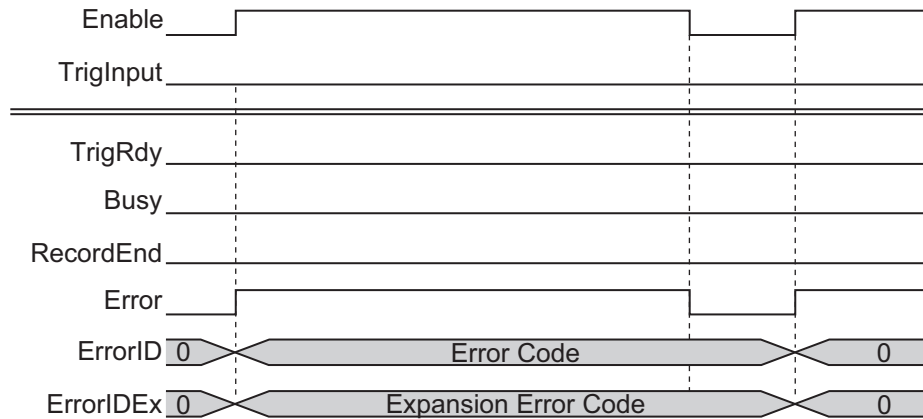
When an error occurs, retain *Error* while *Enable* is TRUE.

After *Enable* is set to FALSE, retain *ErrorID* and *ErrorIDEx* until *Enable* is again set 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	Revise 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	Revise the number of <i>LogData[]</i> array or <i>DataBuf[]</i> array elements.
	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 TrigMark.Pos	<i>TrigMark.Pos</i> exceeds the <i>InputSize</i>	Connect the <i>TrigMark</i> output for the TrigControl FB to the <i>TrigMark</i> for this FB.

## 5 FB/FUN Individual Specifications (DataRecorder)

Error code	Expansion error code	Status	Description	Corrective action
	16#00000005	Illegal TrigPos	The <i>TrigPos</i> is outside the <i>LogData[]</i> array or <i>DataBuf[]</i> array range	<ul style="list-style-type: none"> <li>• Revise the <i>TrigPos</i> setting value.</li> <li>• Revise 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

Test whether each element value in the data array targeted for tests is held to within the allowed range for test standard data.

FB/FUN name	Name	FB/FUN	Graphic expression	ST expression
LimitTest	Upper and Lower Limit Test	FB		<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 start element No. in the test target range	Depends on data type.	—	0
Pos2	Test Range Element 2	UINT	Specifies the end element No. in the test target range	Depends on data type.	—	0
LimitBand	Tolerance Value	REAL	Specifies the tolerance range	0 and greater	—	0.0

## 5 FB/FUN Individual Specifications (DataRecorder)

	Meaning	Data type	Description	Valid range	Unit	Default
CycleNum	Number of Elements Tested in 1 Period	UINT	Specifies the number of elements tested in 1 task period If 0 is specified, process the test target range element for 1 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: Non-passing elements exist	TRUE, FALSE	—	—
FirstOutOfLimitPos	First non-passing element No.	UINT	Element No. of first element discovered to be out of the allowed 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 Unit Hardware 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 targeted for testing	—	—	—

	Meaning	Data type	Description	Valid range	Unit	Default
MasterData[] <sup>2</sup>	Test Standard Data	ARRAY[*] OF Omron-Lib\DataRecorder\sLogData	Specifies the data array forming the test standard	—	—	—

- \*1. The number of array elements is arbitrary. In addition, the array element start number can be either 0 or a number other than 0.
- \*2. The number of array elements is arbitrary. However, it must have the same number of elements as *TestData[]*. In addition, the array element start number can be either 0 or a number other than 0.

## Structure

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

## Function

Test whether each *TestData[]* element is within the *MasterData[]* ± *LimitBand* range.

The *TestData[]* test range is elements specified between *Pos1* and *Pos2*. Do not test elements outside of the range.

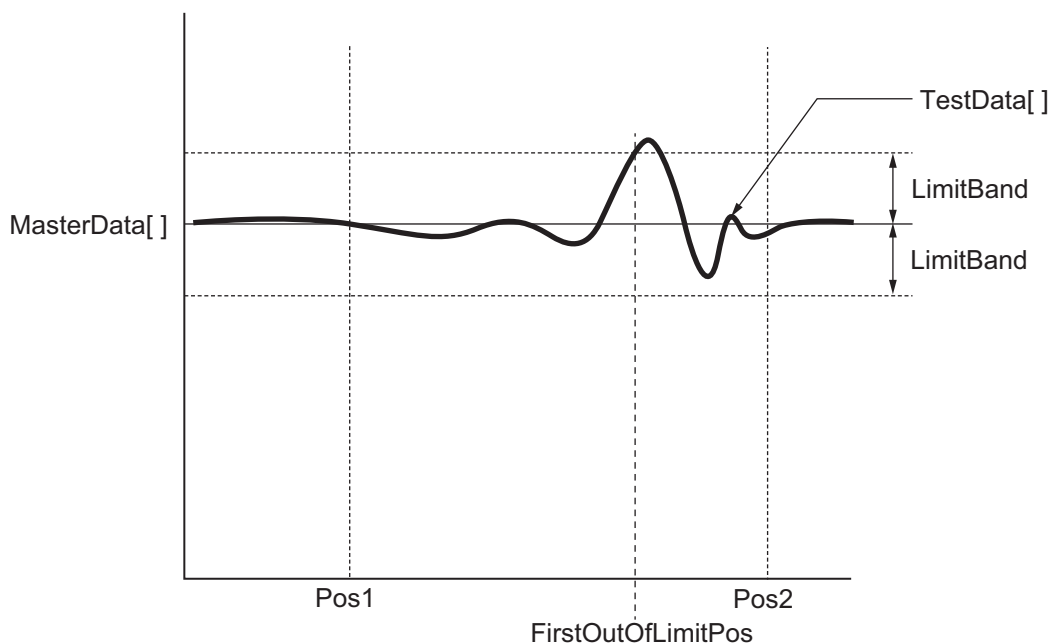
With the startup of *Execute*, test each *TestData[]* element in order from *Pos1*.

When all *TestData[]* elements from *Pos1* to *Pos2* are within the *MasterData[]* ± *LimitBand* range, set *Pass* to TRUE, and output 65535 to *FirstOutOfLimitPos*.

If testing elements outside the allowed range during testing, do not perform the tests up to *Pos2*.

When elements outside the range are detected, set *Busy* to FALSE and *Done* to TRUE. In

*FirstOutOfLimitPos*, output the element No. where the value outside the allowed range was first detected as a relative value from the beginning of the array.



The pass judgment standards are as shown below.

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

## 5 FB/FUN Individual Specifications (DataRecorder)

In the cases below, output TRUE to *Error*, and end the test.

- When *TestData[]* and *MasterData[]* array element numbers do not match
- When the *LimitBand* value is a negative number, non-number, or  $\infty$
- When *Pos1* and *Pos2* are outside the *TestData[]* or *MasterData[]* array range

*Pos1* and *Pos2* specification method

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

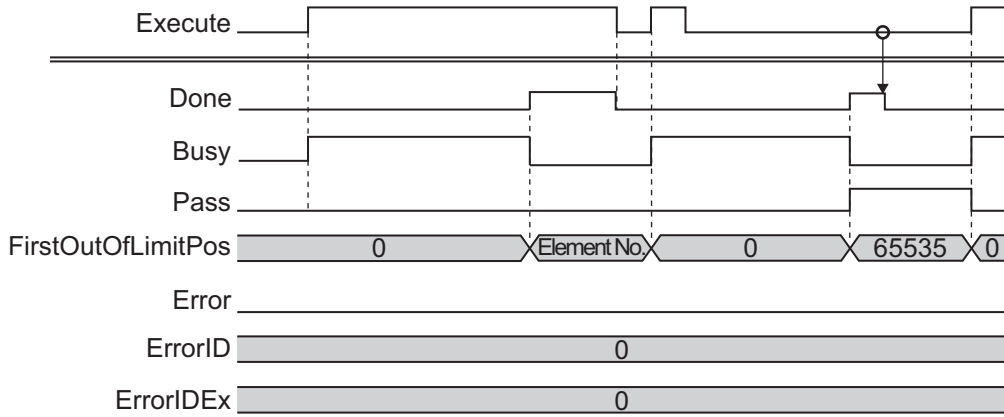
Example: For *TestData* [10..100], if *Pos1* is 0, it indicates *TestData*[10].

- Setting changes during execution  
Multiple startup of this FB, and changes while execution is in progress for *Pos1*, *Pos2*, *LimitBand*, and *CycleNum* are not accepted.
- *CycleNum* specification  
With specification of the number of test elements per 1 task period in *CycleNum*, you can adjust the execution time per 1 task period of the FB and the time until test completion.  
If a small value is specified, the execution time per 1 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 1 task period is lengthened.  
Perform the adjustment with consideration for the number of *TestData[]* array elements, the task period when deploying this FB, and the load from programs other than this FB that are operating.

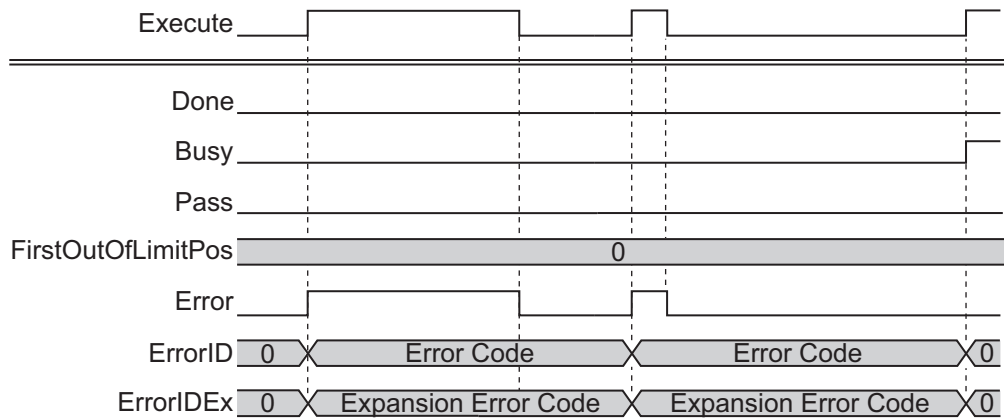
## Timing Chart

The timing charts are shown below.

- With the startup of *Execute*, set *Busy* to TRUE.
- If the judgment for all elements is completed, set *Done* to TRUE, and output the test results to *Pass* and *FirstOutOfLimitPos*.  
Retain *Done* until *Execute* is set to FALSE.  
After *Execute* is set to FALSE, retain *Pass* and *FirstOutOfLimitPos* until *Execute* is again set to TRUE.
- If an error occurs, set *Error* to TRUE. At the same time, output *ErrorID* and *ErrorIDEx*.  
Retain *Error* until *Execute* is set to FALSE.  
After *Execute* is set to FALSE, retain *ErrorID* and *ErrorIDEx* until *Execute* is again set to TRUE.
- If *Execute* is set to FALSE while *Busy* is TRUE, then *Done* and *Error* are TRUE only for 1 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 *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 <i>TestData[]</i> and <i>MasterData[]</i> array element numbers do not match	Revise the <i>TestData[]</i> and <i>MasterData[]</i> array element number.
	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	Revise the <i>Pos1</i> and <i>Pos2</i> values.

## Sample Programming

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

# CalcFeatureValues

For the test target data array, calculate the mean, standard deviation, skew, kurtosis, maximum value, and minimum value.

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

CalcFeatureValues

5

Library Information

## 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 final element No. in the range for calculating FeatureValues[0]	Depends on data type.	—	65535
Pos2	Calculation Range Specification 2	UINT	Specifies the final element No. in the range for calculating FeatureValues[1]	Depends on data type.	—	65535
Pos3	Calculation Range Specification 3	UINT	Specifies the final element No. in the range for calculating FeatureValues[2]	Depends on data type.	—	65535

## 5 FB/FUN Individual Specifications (DataRecorder)

	Meaning	Data type	Description	Valid range	Unit	Default
Pos4	Calculation Range Specification 4	UINT	Specifies the final element No. in the range for calculating FeatureValues[3]	Depends on data type.	—	65535
CycleNum	Number of Elements Calculated in 1 Period	UINT	Specifies the number of elements calculated in 1 task period. If 0 is specified, process the calculation target range element for 1 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 Amount	ARRAY[0..3] OF Omron-Lib\DataRecorder\Feature-Value	Feature Amount Calculation Result	—	—	—
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 Unit Hardware 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 amounts	—	—	—

\*1. The number of array elements is arbitrary. In addition, the array element start number can be either 0 or a number other than 0.

## 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

With the startup of *Execute*, calculate the mean value, standard deviation, skew, kurtosis, maximum value, and minimum value of the *TestData[]* elements within the range specified in *Pos1*, *Pos2*, *Pos3*, and *Pos4*, and store the calculation results in *FeatureValues[]*.

- Calculation range specification

Items related to 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 start element and end element sizes are reversed.

Calculation Range		Calculation Result Storage Location
Start Element	Final 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 to *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.

## 5 FB/FUN Individual Specifications (DataRecorder)

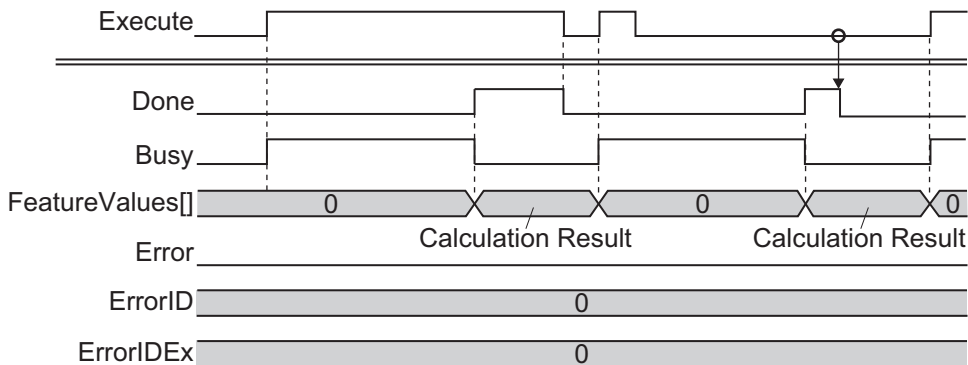
Pos1	Pos2	Pos3	Pos4	FeatureVal- ues [0]	FeatureVal- ues [1]	Feature- Values [2]	Feature- Values [3]
65535	Invalid	Invalid	Invalid	0	0	0	0
Valid Value	65535	Invalid	Invalid	Calculation Result	0	0	0
Valid Value	Valid Value	65535	Invalid	Calculation Result	Calculation Result	0	0
Valid Value	Valid Value	Valid Value	65535	Calculation Result	Calculation Result	Calculation Result	0
Valid Value	Valid Value	Valid Value	Valid Val- ue	Calculation Result	Calculation Result	Calculation Result	Calcula- tion Re- sult

- *Pos1*, *Pos2*, *Pos3*, and *Pos4* specification method  
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  
Multiple startups of this FB, and changes while execution is in progress for *Pos1*, *Pos2*, *Pos3*, *Pos4*, and *CycleNum* are not accepted.
- *CycleNum* specification  
With specification of the number of calculation elements per 1 task period in *CycleNum*, you can adjust the execution time per 1 task period of the FB and the time until calculation completion.  
If a small value is specified, the execution time per 1 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 1 task period is lengthened.  
Perform the adjustment with consideration for the number of *TestData[]* array elements, the task period when deploying this FB, and the load from programs other than this FB that are operating.

## Timing Chart

The timing charts are shown below.

- With the startup of *Execute*, set *Busy* to TRUE.
- If the calculation of all feature amounts is completed, set *Done* to TRUE, and output the calculation result to *FeatureValues[]*.  
Retain *Done* until *Execute* is set to FALSE.  
After *Execute* is set to FALSE, retain *FeatureValues[]* until *Execute* is again set to TRUE.
- If *Busy* is set to FALSE while *Busy* is TRUE, then *Done* is TRUE only for 1 period after the calculation end.



### Precautions for Correct Use

Execution of this function block will be continued until processing is ended even if *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 bind the analog input values sampled 100 times per 1 task period to a single array variable. Furthermore, this is a program for recording the bound data with DataRecorder, and performing a comparison test of the test standard data using LimitTest.

This is an example of the connection of the NX-series High Speed Analog Input Unit to an NX-series CPU Unit.

This sample programming consists of two POU, a program 0 for executing in the primary periodic task, and a program 1 for operation in 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		

## 5 FB/FUN Individual Specifications (DataRecorder)

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 for 1 Task Period
LogData	ARRAY[0..4999] OF OmronLib\DataRecorder\LogData		Recorder Data Array
RecordInfo	OmronLib\DataRecorder\LogData		
RecordEnd	BOOL		

- Internal Variables

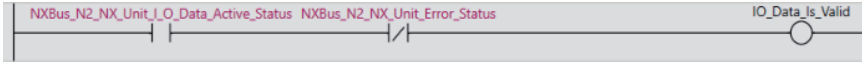
Name	Data type	Default	AT	Retention	Constant	Comment
TrigControl_instance	OmronLib\DataRecorder\TrigControl					
DataRecorder_instance	OmronLib\DataRecorder\DataRecorder					
IO_Data_Is_Valid	BOOL					
TrigConfig	OmronLib\DataRecorder\TrigConfig					
TrigMark	OmronLib\DataRecorder\TrigMark					
TrigOut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

- Task Setting

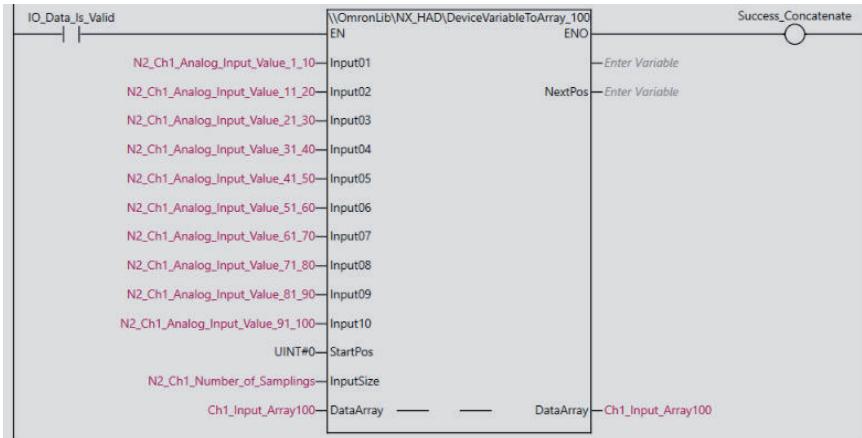
Deploy in the primary periodic task.

• LD Program

- (1) Check that normal data is being received from the NX-series High Speed Analog Input Unit. If connecting the NX-series High Speed Analog Input Unit to the EtherCAT slave terminal, refer to *Sample Programming 2* on page 4 - 7.



- (2) Bind the analog input values for 1 task period to the single variable *Ch1\_Input\_Array100[]*.

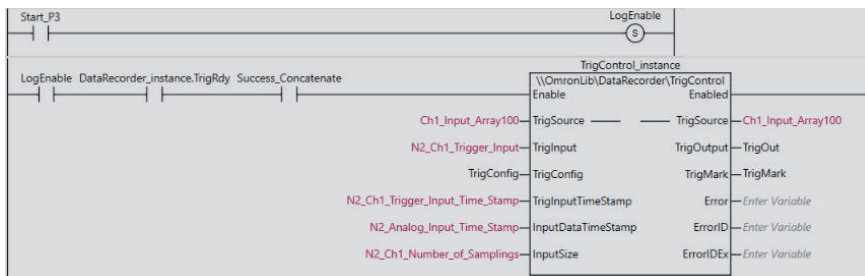


- (3) If *Start\_P3* is set to TRUE, the NX-series High Speed Analog Input Unit input value record enters a trigger wait state. Set the trigger conditions in *TrigConfig*.

• Trigger Conditions

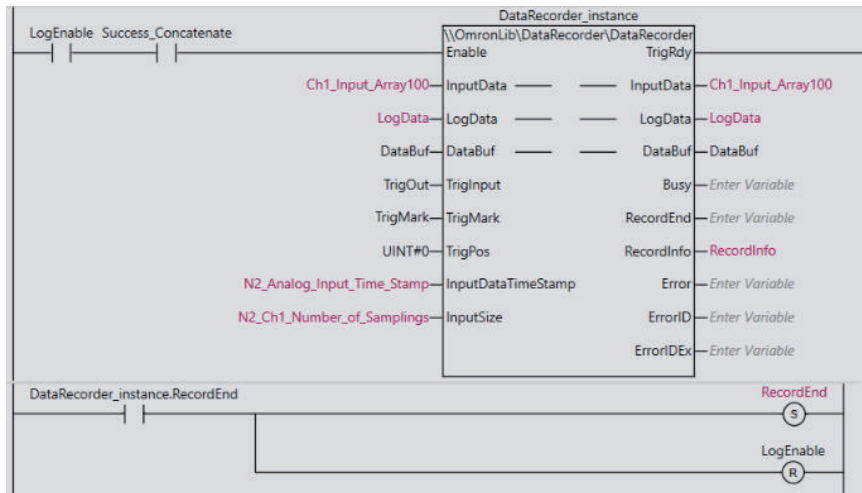
*TrigConfig.TrigMode=2* (Time Stamp Trigger Mode)

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



- (4) If the trigger conditions are met, record the 5,000 sampling data from trigger establishment in *LogData[]*.

## 5 FB/FUN Individual Specifications (DataRecorder)



### Program 1 (Periodic Task 17)

- External Variables

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

- Internal Variables

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

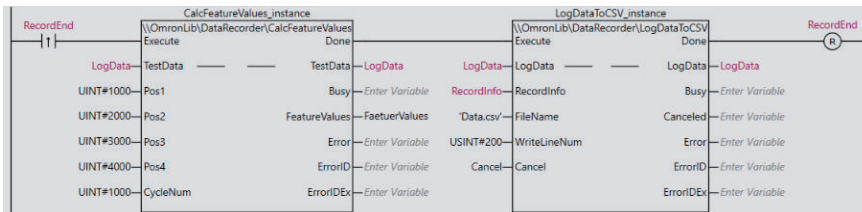
- Task Setting

Deploy in periodic task 17.

- LD Program

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

When you complete the feature amount calculation, write the *LogData[]* data to an SD memory card with the file name *Data.csv*.



In *FeatureVlues[]*, store the following feature amounts.

	.xMean	.xSD	.xSkewness	.xKurtosis	.xMax	.xMin	Remarks
FeatureVlues [0]	Mean Value	Standard Deviation	Skew	Kurtosis	Maximum Value	Minimum Value	LogData[0..1000] feature amount
FeatureVlues [1]	Mean Value	Standard Deviation	Skew	Kurtosis	Maximum Value	Minimum Value	LogData[1001..2000] feature amount
FeatureVlues [2]	Mean Value	Standard Deviation	Skew	Kurtosis	Maximum Value	Minimum Value	LogData[2001..3000] feature amount
FeatureVlues [3]	Mean Value	Standard Deviation	Skew	Kurtosis	Maximum Value	Minimum Value	LogData[3001..4000] feature amount



**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

Output the log data created in the DataRecorder FB as a CSV type 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. Can specify items including directory	Maximum 66 bytes (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 1 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 Unit Hardware 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[*] <sup>1</sup>	Log Data	ARRAY[*] OF OmronLib\DataRecorder\nsLogData	Specifies log data read-out to the CSV file	—	—	—

\*1. The number of array elements is arbitrary. In addition, the array element start number can be either 0 or a number other than 0.

## Structure

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

## Function

With the startup of *Execute*, output the log data stored in *LogData[]* and *RecordInfo* to the SD memory card in the CSV comma separated values file format. Specify the output file name 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 SD memory card route directory.

If *Cancel* is set to TRUE while write execution is in progress, cancel the file write. When the cancel process ends, set *Canceled* to TRUE. Even when 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 left side data and right side data are separated by a comma.

The parentheses ( ) output the value of the variable listed inside the parentheses.

m refers to the first element No. of *LogData[]*, and n to the final element No.

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

Multiple startups of this FB and changes in the *FileName* and *WriteLineNum* are not accepted.

- *WriteLineNum* specification

With specification of the number of output lines per 1 task period in *WriteLineNum*, you can adjust the execution time per 1 task period of the FB, and the time until output end.

If a small value is specified, the execution time per 1 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 1 task period is lengthened.

Perform the adjustment with consideration for the number of *LogData[]* array elements, the task period when deploying this FB, and the load from programs other than this FB that are operating.

However, in this FB, the limit of 1 KB per 1 task period is set as the upper limit.

## Additional Information

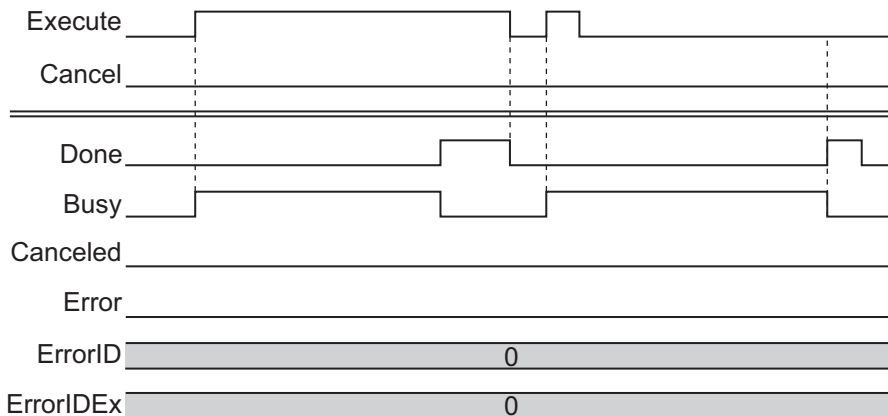
- For *RecordInfo.StartDateTime*, *RecordInfo.EndDateTime*, and *RecordInfo.TriggerDateTime*, use the *DtToString* command to convert to text strings and write. Refer to the *NJ/NX-series Instructions Reference Manual (Cat. No. W508)* for details on the *DtToString* command.
- For *LogData[].xData*, use the *RealToFormatString* command and convert to a text string. 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. W508)* for details on the *RealToFormatString* command.

## Timing Chart

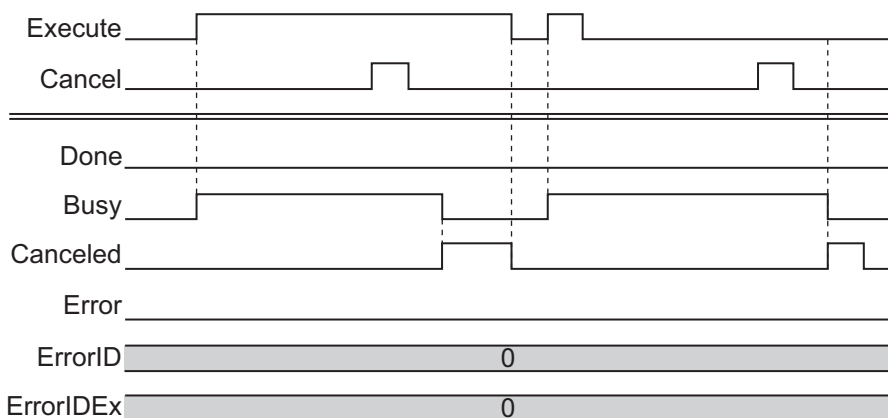
The timing charts are shown below.

- With the startup of *Execute*, set *Busy* to TRUE.
- When the write of all the log data is completed, set *Done* to TRUE. Retain *Done* until *Execute* is set to FALSE.
- If *Cancel* is set to TRUE while write execution is in progress, cancel the file write, and set *Canceled* to TRUE.
- If an error occurs, set *Error* to TRUE. At the same time, output *ErrorID* and *ErrorIDEx*. Retain *Error* and *Canceled* until *Execute* is set to FALSE. After *Execute* is set to FALSE, retain *ErrorID* and *ErrorIDEx* until *Execute* is again set to TRUE.
- If *Execute* is set to FALSE while *Busy* is TRUE, then *Done*, *Canceled*, and *Error* are TRUE for only 1 period after the end of write/cancel processing.
- Timing Chart for Normal End

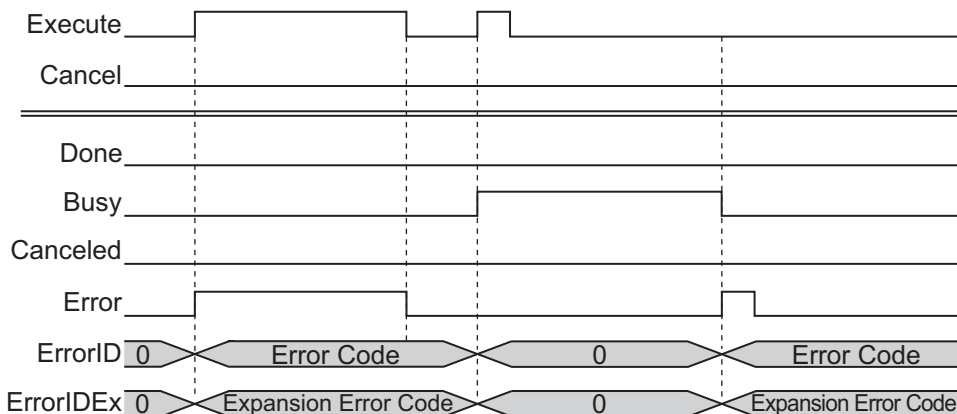
## 5 FB/FUN Individual Specifications (DataRecorder)



• 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> 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 Unit. 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 Failed		
16#4400	16#00000000	Shared Folder Cannot Be Used		This occurs when this FB is used in the NY-series Industrial PC. Refer to the <i>NY-series Troubleshooting Manual (SBCA-438)</i> .
16#4402	16#00000000	Insufficient Shared Folder 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 Failed		
16#3D05	16#00000001	Illegal <i>WriteLineNum</i>	<i>WriteLineNum</i> is outside the range	

## Sample Programming

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

# CSVToLogData

Read out the log data recorded in the SD memory card from a CSV type file to the *LogData[]* array variables as LimitTest FB test standard data.

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>

CSVToLogData

5

Library Information

## 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. Can specify items including directory	Maximum 66 bytes (65 single-byte alpha-numeric characters plus the final NULL character)	—	''
Cancel	Cancel	BOOL	TRUE: Cancels the file read-out	TRUE, FALSE	—	FALSE

## 5 FB/FUN Individual Specifications (DataRecorder)

## 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	Record 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 Unit Hardware 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[] <sup>*1</sup>	Log Data	ARRAY[*] OF OmronLib\DataRecorder \sLogData	Specifies log data read-out to the CSV file	—	—	—

\*1. The number of array elements is arbitrary. However, it must be equal to or more than the number of data items recorded in the read-out file. In addition, the array element start number can be either 0 or a number other than 0.

## Structure

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

## Function

With the startup of *Execute*, read out the log data from the file specified in *FileName*, and store 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* is set to FALSE, retain *RecordInfo* and *NumOfRecord* until *Execute* is again set to TRUE.

For *FileName*, you can specify items by including the directory. If a directory is not specified, the SD memory card route directory 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 type created in the LogDataToCSV FB.

An error occurs if a different file format is specified.

If *Cancel* is set to TRUE while read-out execution is in progress, abort the file read-out. When the cancel process ends, set *Canceled* to TRUE. If the read-out is canceled, do not output *RecordInfo* and *NumOfRecord*. The *LogData[]* content is indefinite.

- File Format

Refer to the LogDataToCSV FB.

- Setting changes during execution

Multiple startups of this FB and changes in the "FileName" while execution is in progress are not accepted.

## Timing Chart

The timing charts are shown below.

- With the startup of *Execute*, set *Busy* to TRUE.

- When the read-out of all the log data is completed, set *Done* to TRUE.

Retain *Done* until *Execute* is set to FALSE.

After *Execute* is set to FALSE, retain *RecordInfo* and *NumOfRecord* until *Execute* is again set to TRUE.

- If *Cancel* is set to TRUE while read-out execution is in progress, cancel the file read-out, and set *Canceled* to TRUE.

- If an error occurs, set *Error* to TRUE. At the same time, output *ErrorID* and *ErrorIDEx*.

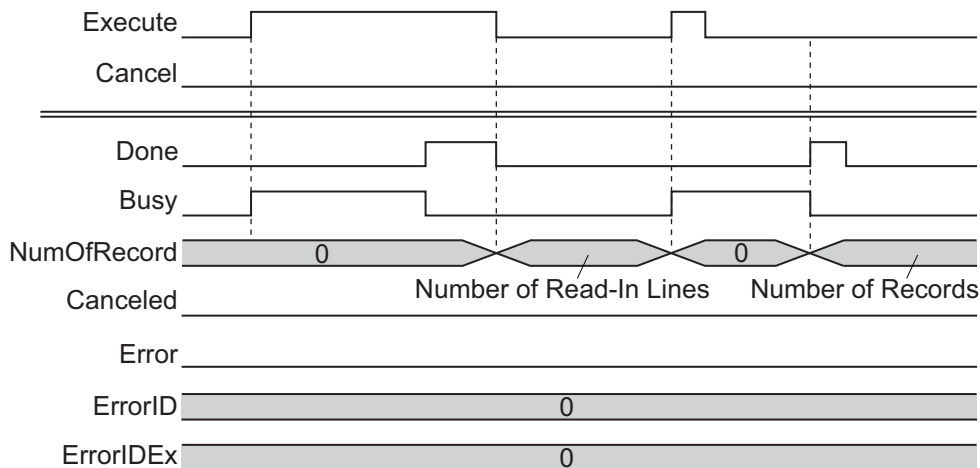
Retain *Error* and *Canceled* until *Execute* is set to FALSE.

After *Execute* is set to FALSE, retain *ErrorID* and *ErrorIDEx* until *Execute* is again set to TRUE.

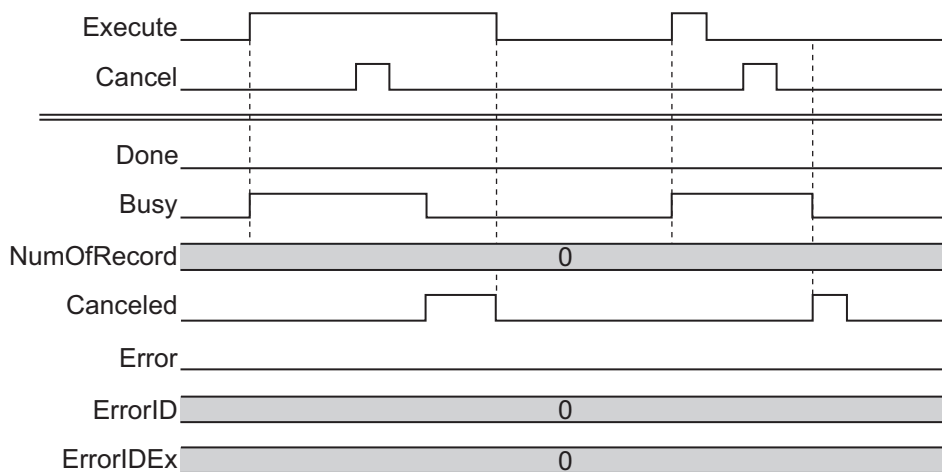
- If *Execute* is set to FALSE while *Busy* is TRUE, then *Done*, *Canceled*, and *Error* are TRUE for 1 only period after the end of the read-out/cancel processing end.

- Timing Chart for Normal End

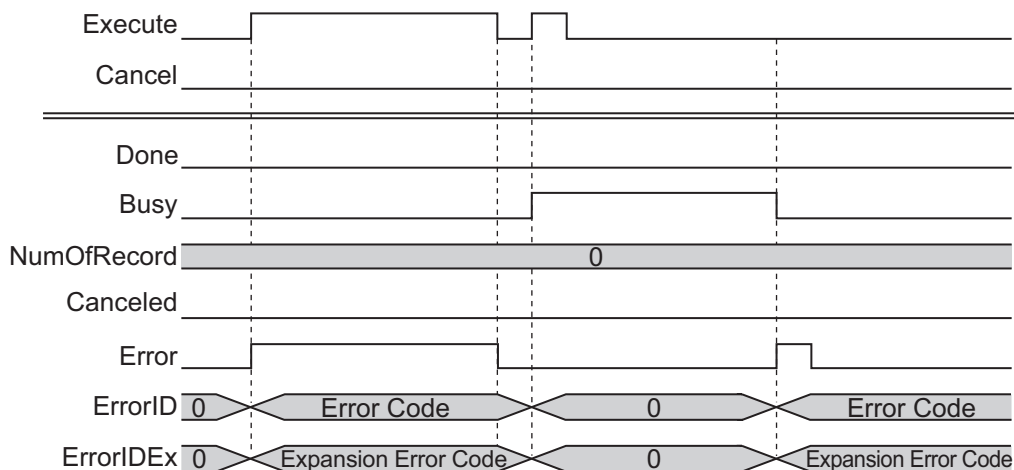
## 5 FB/FUN Individual Specifications (DataRecorder)



• 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

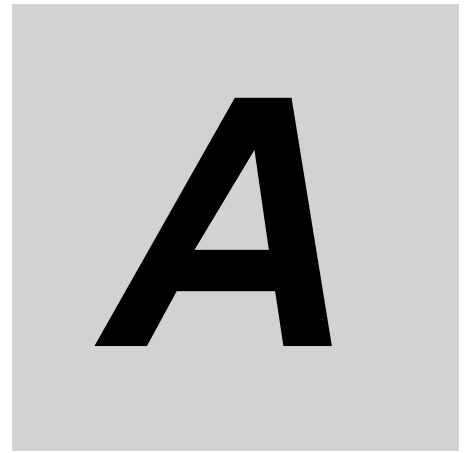
Error code	Expansion error code	Status	Description	Corrective action
16#0000	16#00000000	Normal End	—	—
16#0400	16#00000000	Input Value Out of Range	The file name specified with <i>FileName</i> contains one or more characters that cannot be used	Set <i>FileName</i> correctly.
			The directory name specified with <i>FileName</i> is too long	Check the length of the text strings specified with <i>FileName</i> 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 Unit. 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 Failed		
16#4400	16#00000000	Shared Folder Cannot Be Used		This occurs when this FB is used in the NY-series Industrial PC. Refer to the <i>NY-series Troubleshooting Manual (SBCA-438)</i> .
16#4402	16#00000000	Insufficient Shared Folder 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 Failed		

## 5 FB/FUN Individual Specifications (DataRecorder)

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	Illegal Format	The format does not match the record content inside the file	Check whether the read-out target file is the one that was output by LogDataToCSV.

## 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

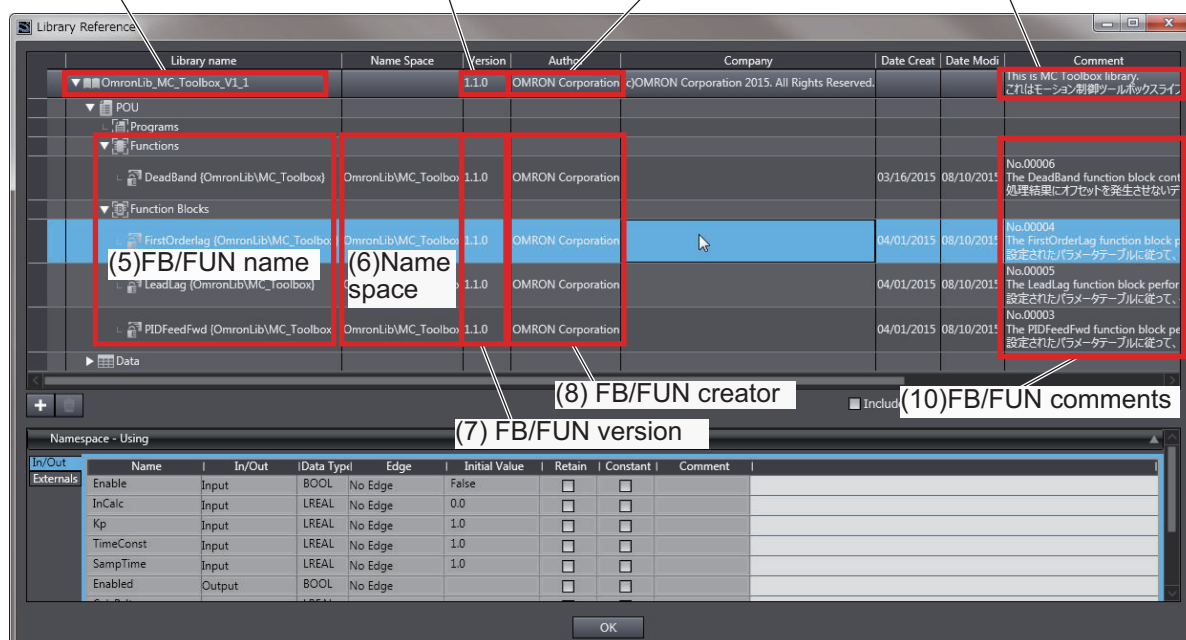
You can refer to the library attributes of library information, and FB or FUN attributes at the following Sysmac Studio locations.

- Library Reference Dialog Box
- Toolbox
- Programming screen

### Library Reference Dialog Box

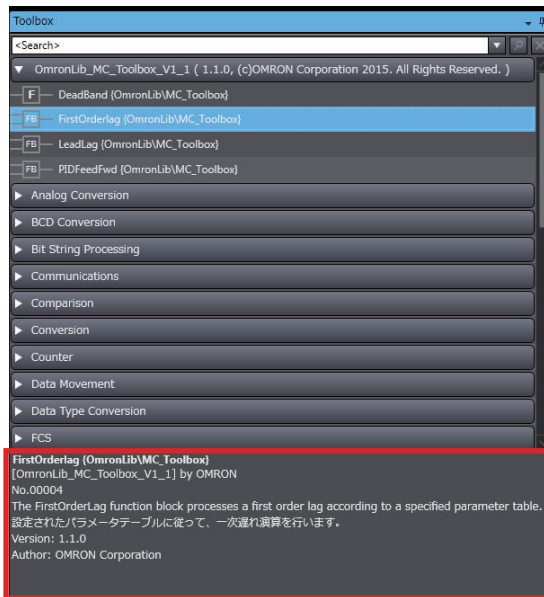
When you refer to the libraries, the library information is displayed at the locations shown below.

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



### Toolbox

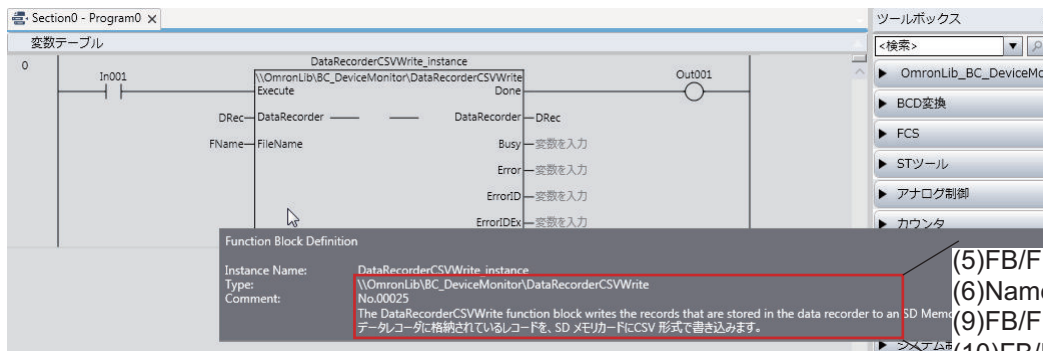
Select a function block or function to display its library information at the bottom of the Toolbox Pane. The text "**by OMRON**" which is shown on the right of the library name (1) indicates that this library was provided by OMRON.



- (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.

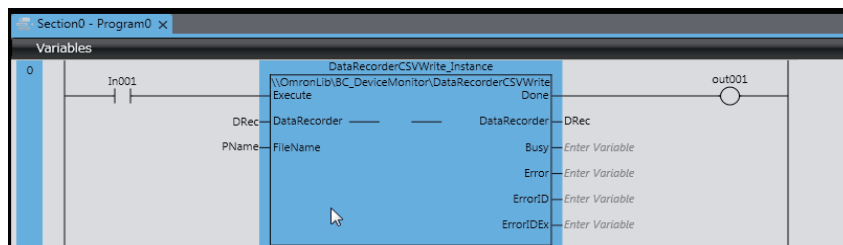
User function blocks and user functions can be created based on the copies of these source codes. The following are the examples of items that you may need to customize.

- 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

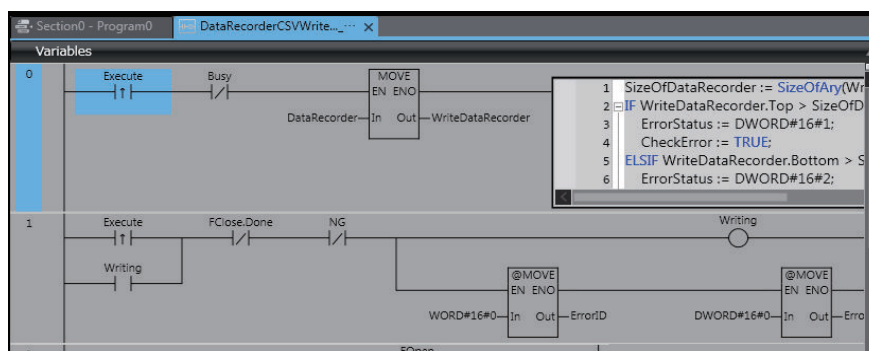
Note that you can access only function blocks and functions whose Source code published/not published is set to "Published" in the library information shown in their individual specifications.

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.



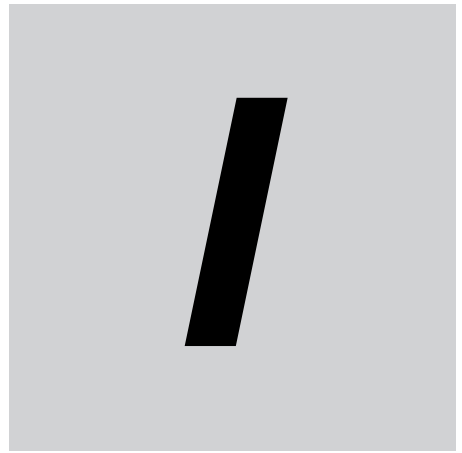


### Precautions for Correct Use

---

- 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 All Rights Reserved.  
In the interest of product improvement,  
specifications are subject to change without notice.

Cat. No. **W607-E1-01**

0518



# SCATTERGOOD & JOHNSON LTD

ELECTRICAL ENGINEERING & FLUID CONTROL DISTRIBUTORS

Est.1899

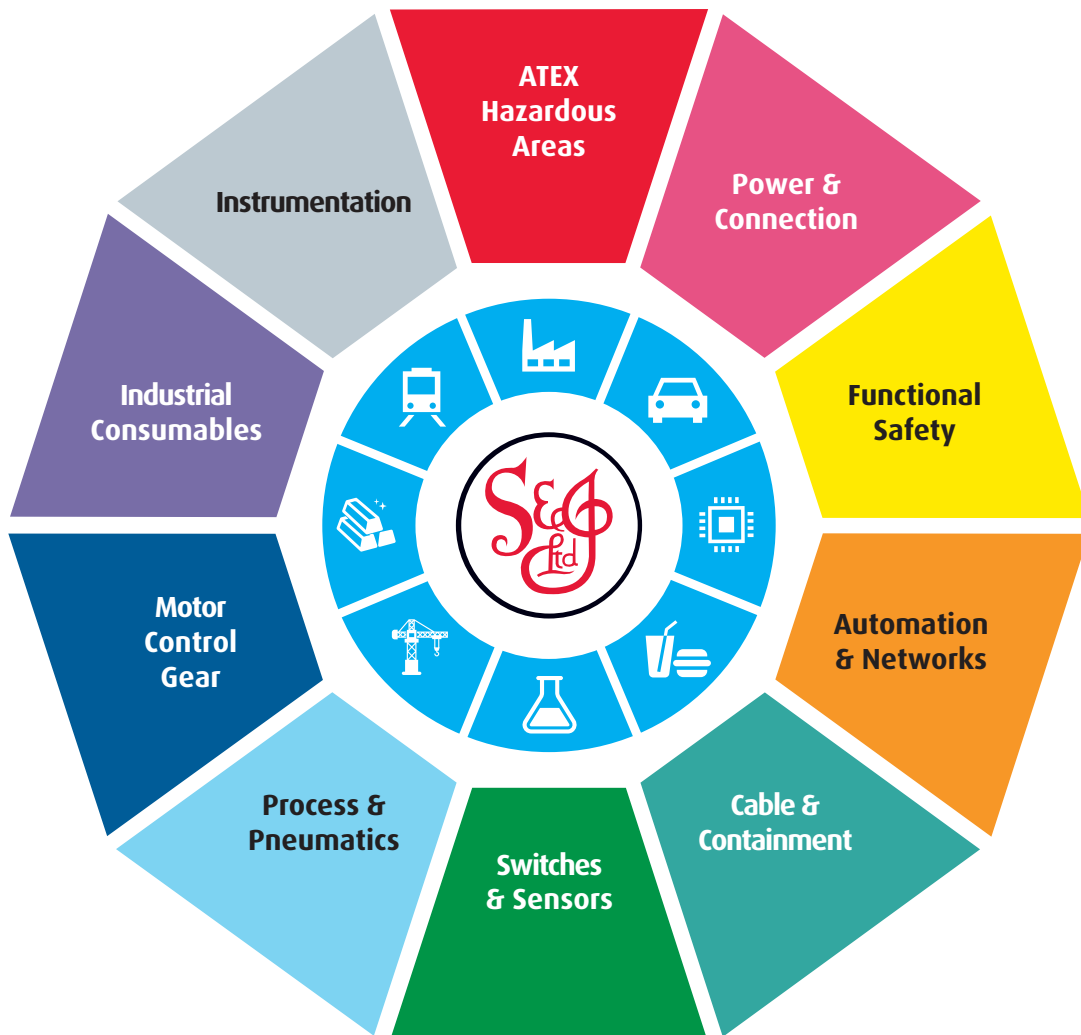
At Scattergood & Johnson Ltd, we pride ourselves on being a technical distributor to specialist industries.

Working with a range of quality product suppliers across a number of specialist markets, we are not your average 'box shifter' - we are your technical and supply chain partner.

We fully support every product we sell - for free! Our internal team and external sales engineers can answer any product or application question, no matter the complexity.

Backing up this technical ability is a range of 50,000+ products available from stock for nationwide next day delivery (same day if required!), or you can collect what you need from any of our trade counters around the UK.

Select your specialist interest below to learn more about how we can help.



Online, In Branch and On the Road - Scattergood & Johnson Ltd, there when you need us.

# [www.scatts.co.uk](http://www.scatts.co.uk)