OMRON

Sysmac Library

User's Manual for Weighing Control Library

SYSMAC-XR010

SYSMAC always in control

W569-E1-03

NOTE

Trademarks

Copyrights

Introduction

Thank you for purchasing an NJ/NX-series CPU Unit or an NY-series Industrial PC.

This manual contains information that is necessary to use the function blocks in the Weighing Control Library. (“Function block” is sometimes abbreviated as “FB”.) Please read this manual and make sure you understand the functionality and performance of the NJ/NX-series CPU Unit 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.

Refer to the user’s manuals for all of the products in the application before you use any of the products.

Keep this manual in a safe place where it will be available for reference during operation.

Features of the Library

The Weighing Control Library is used to perform weighing control of raw materials for industrial products with NX-series Load Cell Input Unit NX-RS1201. Also, functions that are used to display and correct measurement values are included. You can use this Weighing Control Library to reduce programming work when you implement processing for weighing in each device.

Intended Audience

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

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

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

Applicable Products

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

Manual Structure

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 or make operation easier.

Version Information

Information on differences in specifications and functionality for CPU Units and Industrial PCs with different unit versions and for different versions of the Sysmac Studio are given.

Note References are provided to more detailed or related information.

CONTENTS

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 Controller and an NY-series Industrial PC.

The safety precautions that are provided are extremely important to 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 prohibiting disassembly.
	The triangle symbol indicates precautions (including warnings). The specific operation is shown in the triangle and explained in text. This example indicates a precaution for electric shock.
	The triangle symbol indicates precautions (including warnings). The specific operation is shown in the triangle and explained in text. This example indicates a general precaution.
	The filled circle symbol indicates operations that you must do. The specific operation is shown in the circle and explained in text. This example shows a general precaution for something that you must do.

Cautions

⚠ Caution

Read all related manuals carefully before you use this library.	
Emergency stop circuits, interlock circuits, limit circuits, and similar safety measures must be provided in external control circuits.	
Check the user program, data, and parameter settings for proper execution before you use them for actual operation.	
The Sysmac Library and manuals are assumed to be used by personnel that is given in Intended Audience in this manual. Otherwise, do not use them.	
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 user program.	
Create a user program so that the actual device operates as intended.	
The value of the output parameter for weighing may be negative. When you design a program, take into account the case where the weighed value is negative.	
This function block uses weighed values that were read with a Load Cell Input Unit as input parameters. Use them within the rated range.	

Precautions for Correct Use

Operation

- Specify the input parameter values within the valid range.
- In the 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	Cat. No.	Model numbers	Application	Description
NX-series CPU Unit Hardware User's Manual	W535	NX701-□□□□	Learning the basic specifications of the NX-series NX701 CPU Units, including introductory information, designing, installation, and maintenance. Mainly hardware information is provided	An introduction to the entire NX701 CPU Unit system is provided along with the following information on the CPU Unit. Features and system configuration Overview Part names and functions General specifications Installation and wiring Maintenance and inspection
NX-series NX102 CPU Unit Hardware User's Manual	W593	NX102-□□□□	Learning the basic specifications of the NX102 CPU Units, including introductory information, designing, installation, and maintenance. Mainly hardware information is provided.	An introduction to the entire NX102 system is provided along with the following information on the CPU Unit. Features and system configuration Introduction Part names and functions General specifications Installation and wiring Maintenance and Inspection
NX-series NX1P2 CPU Unit Hardware User's Manual	W578	NX1P2-□□□□	Learning the basic specifications of the NX-series NX1P2 CPU Units, including introductory information, designing, installation, and maintenance. Mainly hardware information is provided	An introduction to the entire NX1P2 CPU Unit system is provided along with the following information on the CPU Unit. Features and system configuration Overview Part names and functions General specifications Installation and wiring Maintenance and Inspection
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. Features and system configuration Overview Part names and functions General specifications Installation and wiring Maintenance and inspection
NY-series IPC Machine Controller Industrial Panel PC Hardware User's Manual	W557	NY532-□□□□	Learning the basic specifications of the NY-series Industrial Panel PCs, including introductory information, designing, installation, and maintenance. Mainly hardware information is provided	An introduction to the entire NY-series system is provided along with the following information on the Industrial Panel PC. Features and system configuration Introduction Part names and functions General specifications Installation and wiring Maintenance and inspection

Manual name	Cat. No.	Model numbers	Application	Description
NY-series IPC Machine Controller Industrial Box PC Hardware User's Manual	W556	NY512-□□□□	Learning the basic specifications of the NY-series Industrial Box PCs, including introductory information, designing, installation, and maintenance. Mainly hardware information is provided	An introduction to the entire NY-series system is provided along with the following information on the Industrial Box PC. Features and system configuration Introduction Part names and functions General specifications Installation and wiring Maintenance and inspection
NJ/NX-series CPU Unit Software User's Manual	W501	NX701-□□□□ NX102-□□□□ NX1P2-□□□□ NJ501-□□□□ NJ301-□□□□ NJ101-□□□□	Learning how to program and set up an NJ/NX-series CPU Unit. Mainly software information is provided	The following information is provided on a Controller built with an NJ/NX-series CPU Unit. CPU Unit operation CPU Unit features Initial settings Programming based on IEC 61131-3 language specifications
NY-series IPC Machine Controller Industrial Panel PC / Industrial Box PC Software User's Manual	W558	NY532-□□□□ NY512-□□□□	Learning how to program and set up the Controller functions of an NY-series Industrial PC	The following information is provided on NY-series Machine Automation Control Software. Controller operation Controller features Controller settings Programming based on IEC 61131-3 language specifications
NJ/NX-series Instructions Reference Manual	W502	NX701-□□□□ NX102-□□□□ NX1P2-□□□□ NJ501-□□□□ NJ301-□□□□ NJ101-□□□□	Learning detailed specifications on the basic instructions of an NJ/NX-series CPU Unit	The instructions in the instruction set (IEC 61131-3 specifications) are described.
NY-series Instructions Reference Manual	W560	NY532-□□□□ NY512-□□□□	Learning detailed specifications on the basic instructions of an NY-series Industrial PC	The instructions in the instruction set (IEC 61131-3 specifications) are described.
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 of an NJ/NX-series CPU Unit.	The settings and operation of the CPU Unit and programming concepts for motion control are described.
NY-series IPC Machine Controller Industrial Panel PC / Industrial Box PC Motion Control User's Manual	W559	NY532-□□□□ NY512-□□□□	Learning about motion control settings and programming concepts of an NY-series Industrial PC.	The settings and operation of the Controller and programming concepts for motion control are described.
NJ/NX-series Motion Control Instructions Reference Manual	W508	NX701-□□□□ NX102-□□□□ NX1P2-□□□□ NJ501-□□□□ NJ301-□□□□ NJ101-□□□□	Learning about the specifications of the motion control instructions of an NJ/NX-series CPU Unit.	The motion control instructions are described.
NY-series Motion Control Instructions Reference Manual	W561	NY532-□□□□ NY512-□□□□	Learning about the specifications of the motion control instructions of an NY-series Industrial PC.	The motion control instructions are described.
NJ/NY-series NC Integrated Controller User's Manual	O030	NJ501-5300 NY532-5400	Performing numerical control with NJ/NY-series Controllers.	Describes the functionality to perform the numerical control. Use this manual together with the *NJ/NY-series G code Instructions Reference Manual* (Cat. No. O031) when programming.

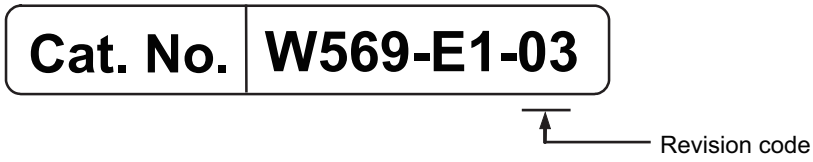
Manual name	Cat. No.	Model numbers	Application	Description
G code Instructions Reference Manual	O031	NJ501-5300 NY532-5400	Learning about the specifications of the G code/M code instructions.	The G code/M code instructions are described. Use this manual together with the *NJ/NY-series NC Integrated Controller User's Manual* (Cat. No. O030) when programming.
Sysmac Studio Version 1 Operation Manual	W504	SYSMAC -SE2□□□	Learning about the operating procedures and functions of the Sysmac Studio.	Describes the operating procedures of the Sysmac Studio.
CNC Operator Operation Manual	O032	SYSMAC -RTNC0□□□D	Learning an introduction of the CNC Operator and how to use it.	An introduction of the CNC Operator, installation procedures, basic operations, connection operations, and operating procedures for main functions are described.
NX-series Load Cell Input Unit User's Manual	W565	NX-RS□□□□	Learning how to use an NX-series Load Cell Input Unit.	The hardware, setup methods, and functions of the NX-series Load Cell Input Unit are described.

Terminology

Term	Description
discharging weighing	A weighing method that controls the discharge amount based on the weight value that decreases when the material is discharged from the scale hopper.
fall	The weight of fluid or powder that floats in the air after the valve, which is used to feed or discharge fluid or powder during quantitative discharging control, is closed.
feed weighing	A weighing method that controls the feed amount based on the weight value that increases when the material is fed into the scale hopper.
maximum capacity	The maximum weight that can correctly be measured by scale.
net weight	Weight of only the measured material that is acquired when you subtract the tare weight from the gross weight.
quantitative discharging control	A control method that is used to discharge a constant quantity of fluid or powder.
scale interval	The smallest value that is viewed on the scale.
tare	A container or a bag in which the measured material is placed when the weight is measured.
tare subtraction	A function that indicates the net weight value that is acquired when you subtract the tare weight from the gross weight beforehand.

Revision History

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



Revision code	Date	Revised content
01	April 2016	Original production
02	November 2016	Changed the manual name.
03	January 2019	Added compatible models.

Procedure to Use Sysmac Libraries

Procedure to Use Sysmac Libraries Installed Using the Installer

This section describes the procedure to use Sysmac Libraries that you installed using the installer.

There are two ways to use libraries.

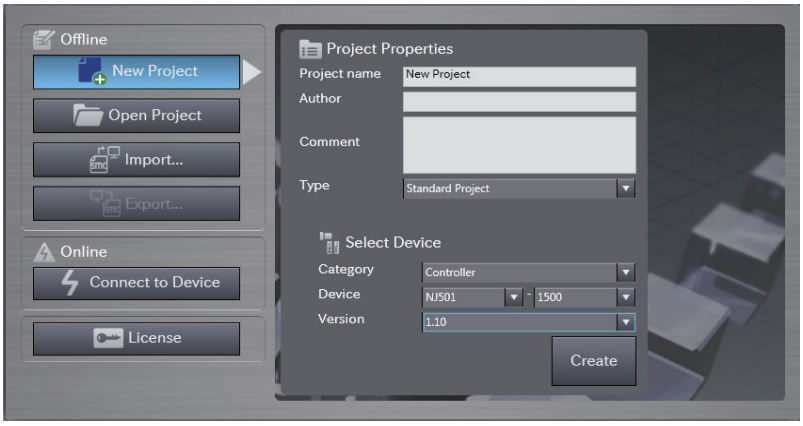
- Using newly installed Sysmac Libraries
- Using upgraded Sysmac Libraries

Version Information

To use Sysmac Libraries, you need the Sysmac Studio version 1.14 or higher.

Using Newly Installed Libraries

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

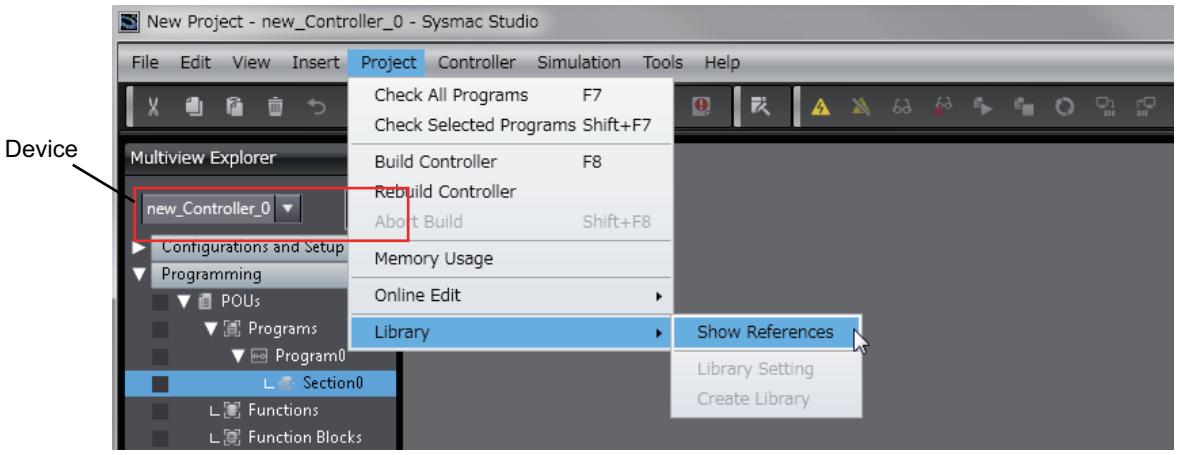


Precautions for Correct Use

If you create a new project, be sure to configure the settings as follows to enable the use of Sysmac Libraries. If you do not configure the following settings, you cannot proceed to the step 2 and later steps.

- Set the project type to Standard Project or Library Project.
- Set the device category to Controller.
- Set the device version to 1.01 or later.

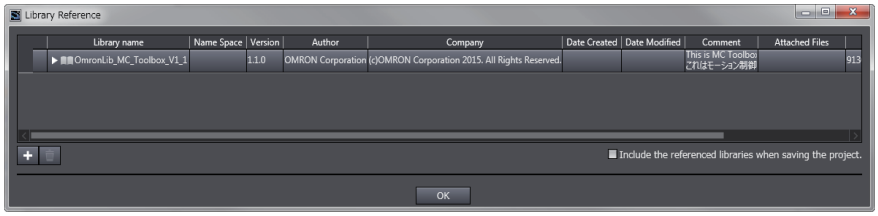
2 Select **Project** – **Library** – **Show References**.



Precautions for Correct Use

If you have more than one registered device in the project, make sure that the device selected currently is an NJ/NX-series CPU Unit or an NY-series Industrial PC. If you do not select an NJ/NX-series CPU Unit or an NY-series Industrial PC as the device, Library References does not appear in the above menu. When the device selected currently is an NJ/NX-series CPU Unit or an NY-series Industrial PC, the device icon is displayed in the Multiview Explorer.

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



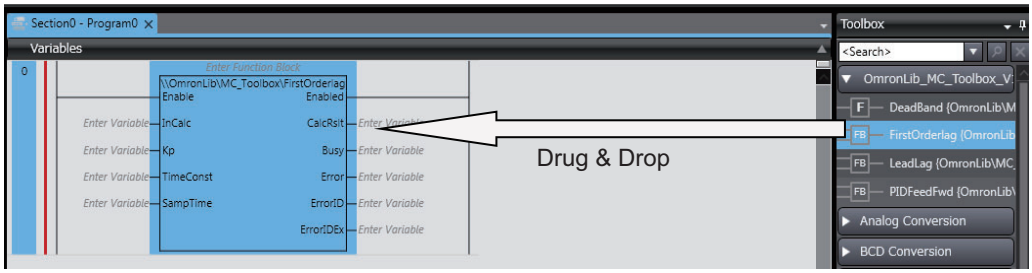
The Sysmac Library file is read into the project.

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

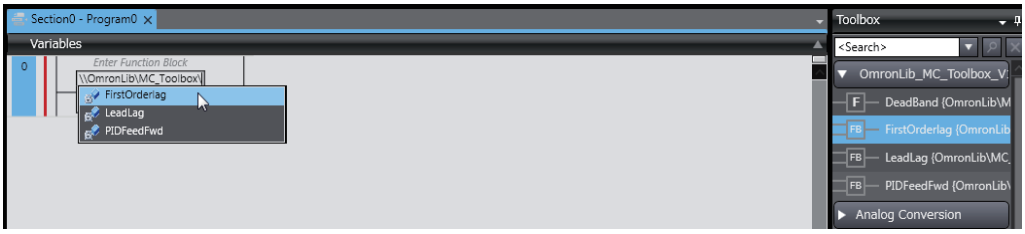
For the procedure for adding and setting libraries in the above screen, refer to the *Sysmac Studio Version 1 Operation Manual* (Cat. No. W504).

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

- Select the desired function block or function in the Toolbox and drag and drop it onto the programming editor.



- Right-click the programming editor, select **Insert Function Block** in the menu, and enter the fully qualified name (\\name of namespace\name of function block).



Precautions for Correct Use

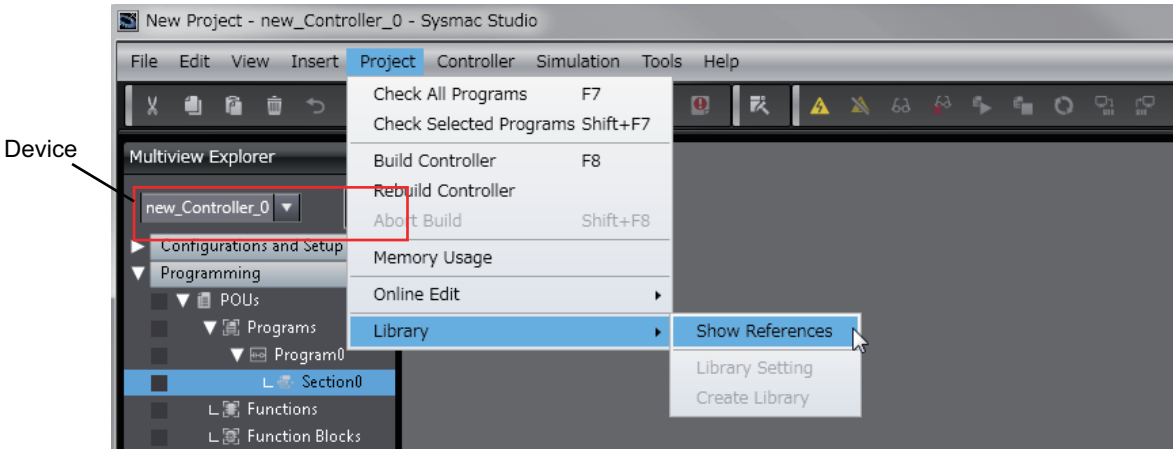
After you upgrade the Sysmac Studio, check all programs and make sure that there is no error of the program check results on the Build Tab Page.

Select **Project** – **Check All Programs** from the Main Menu.

Using Upgraded Libraries

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

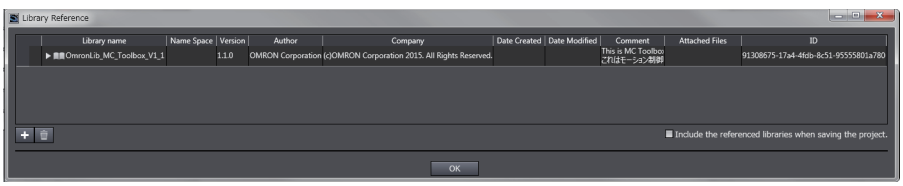
2 Select **Project** – **Library** – **Show References**.



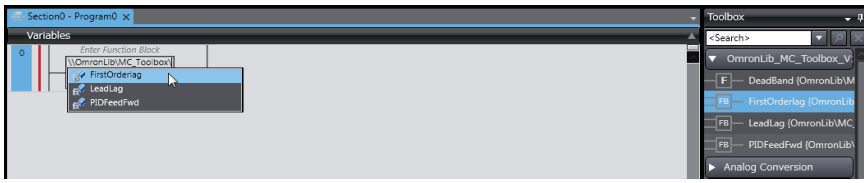
Precautions for Correct Use

If you have more than one registered device in the project, make sure that the device selected currently is an NJ/NX-series CPU Unit or an NY-series Industrial PC. Otherwise, Library References does not appear in the above menu. When the device selected currently is an NJ/NX-series CPU Unit or an NY-series Industrial PC, the device icon is displayed in the Multiview Explorer.

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



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



Procedure to Use Sysmac Libraries Uploaded from a CPU Unit or an Industrial PC

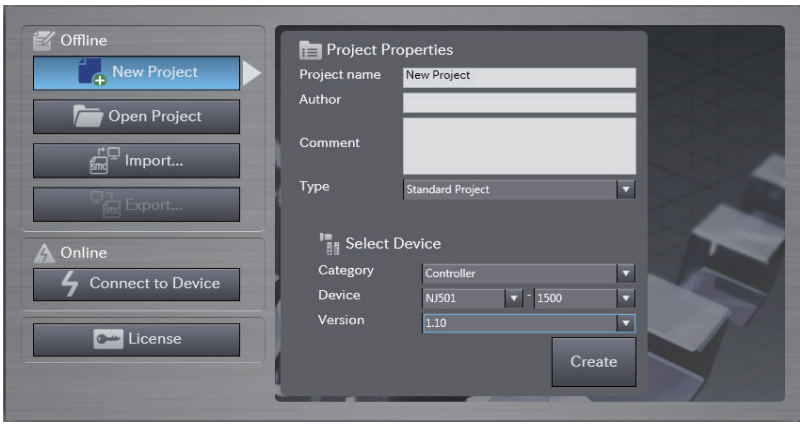
You can use Sysmac Libraries uploaded from a CPU Unit or an Industrial PC to your computer if they are not installed.

The procedure to use uploaded Sysmac Libraries from a CPU Unit or an Industrial PC is as follows.

Version Information

To use Sysmac Libraries, you need the Sysmac Studio version 1.14 or higher.

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



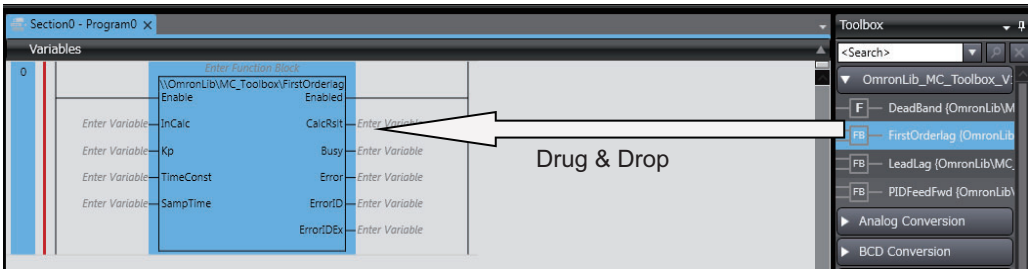
2 Connect the computer to the CPU Unit or the Industrial PC and place it online.

3 Upload POUs in which any Sysmac Library is used to the computer.

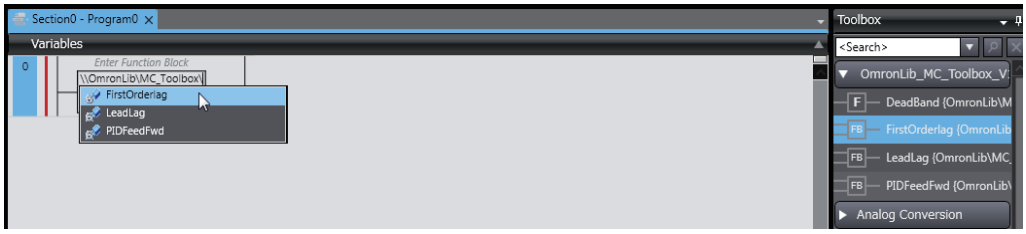
Now, when you select the Ladder Editor or ST Editor, the function blocks and functions included in the Sysmac Library used in the uploaded POUs appear in the Toolbox.

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

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



- Right-click the programming editor, select **Insert Function Block** in the menu, and enter the fully qualified name (\\name of namespace\name of function block).



Precautions for Correct Use

- The Sysmac Studio installs library files of the uploaded Sysmac Stutio to the specified folder on the computer if they are not present. However, the Sysmac Studio does not install library files 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 Libraries from a CPU Unit or an Industrial PC does not install the manual and help files for the Sysmac Libraries, unlike the case where you install then using the installer. Please install the manual and help files using the installer if you need them.

Common Specifications of Function Blocks

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.

Definition of Input Variables and Output Variables

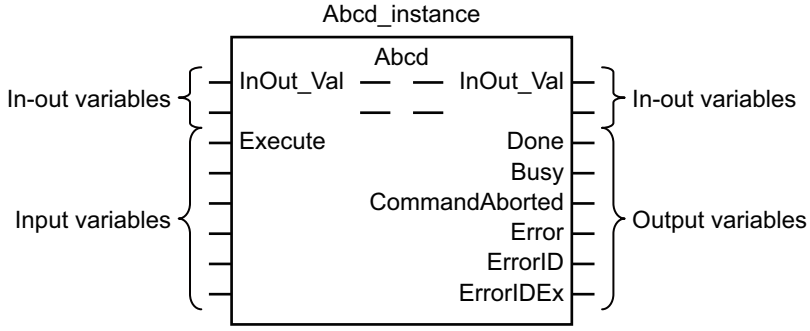
Common input variables and output variables used in functions and function blocks are as follows.

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			OK			Execute	The processing is executed when the variable changes to TRUE.
Enable				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.

Variable	I/O	Data type	Function/function block type to use			Meaning	Definition
			Function block		Function		
			Execute-type	Enable-type			
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.
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 re-executed the next time.
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.

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 continously executes the 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 calculation for motion control and temperature control, you can use the BOOL input variable *Abort* to abort the processing of a function block. When *Abort* changes to TRUE, *CommandAborted* changes to TRUE and the execution of the function block is aborted.

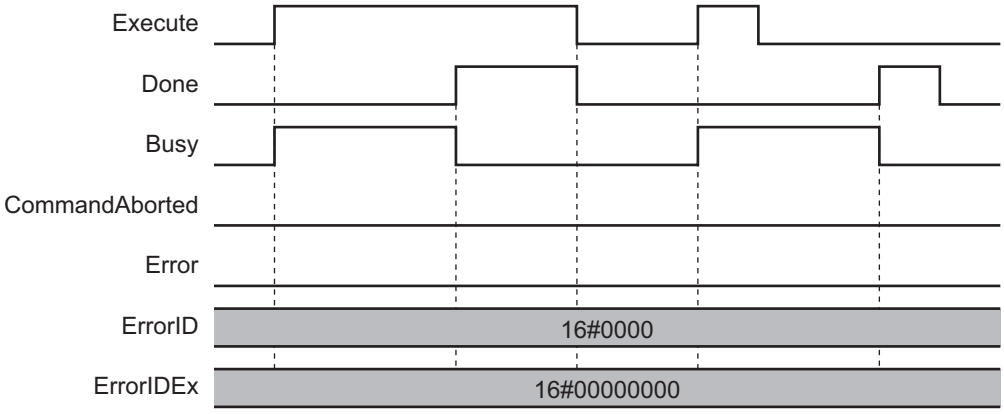


- If *Execute* is TRUE and *Done*, *CommandAborted*, or *Error* changes to TRUE, *Done*, *Command-Aborted*, and *Error* changes to FALSE when *Execute* is changed to FALSE.
- If *Execute* is FALSE and *Done*, *CommandAborted*, or *Error* changes to TRUE, *Done*,*Command-Aborted*, and *Error* changes 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.

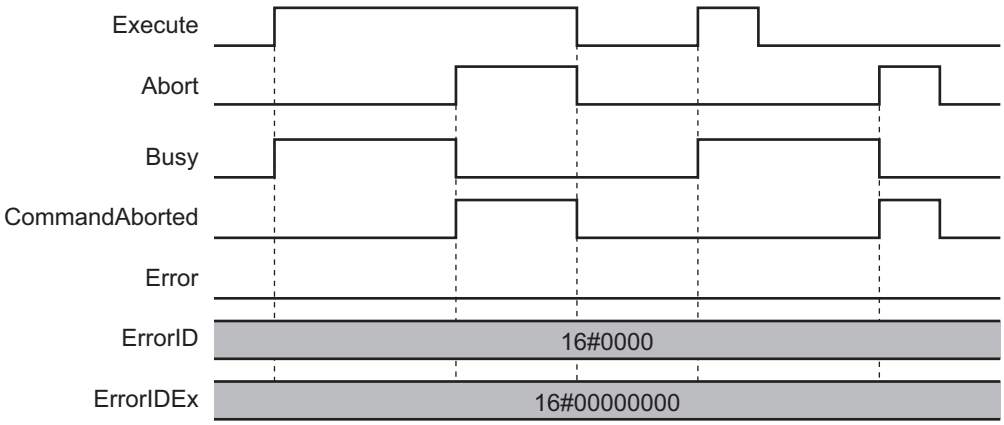
Timing Charts

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

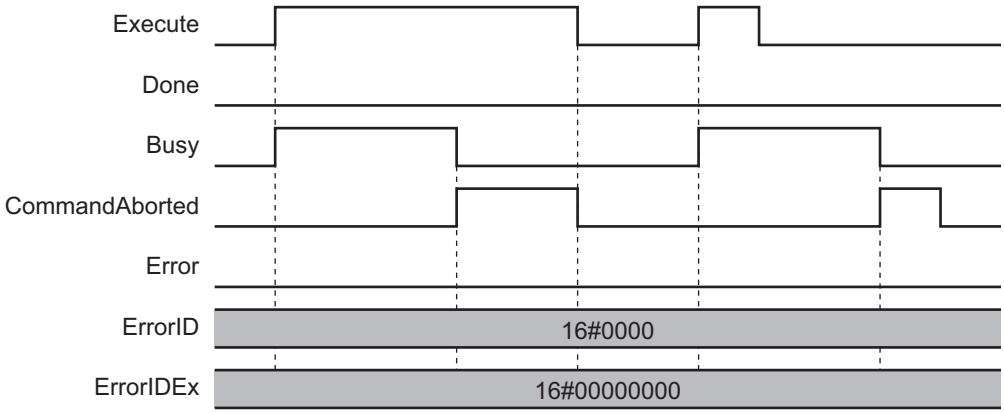
● Normal End



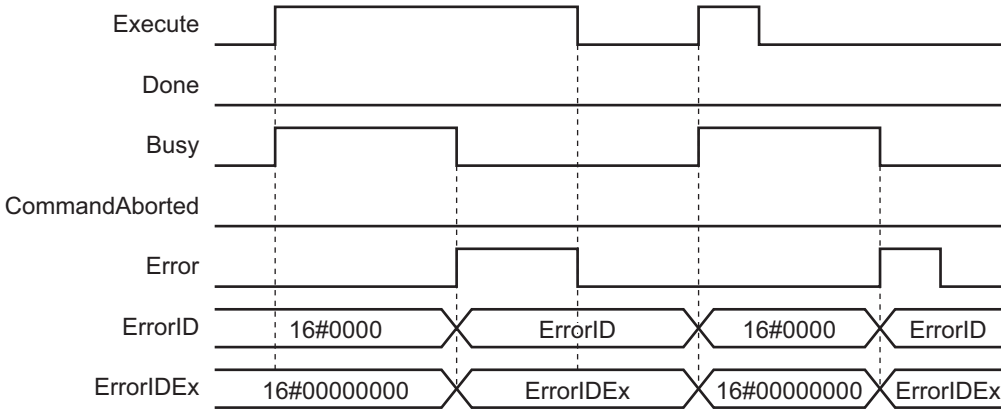
● Canceled Execution



Aborted Execution

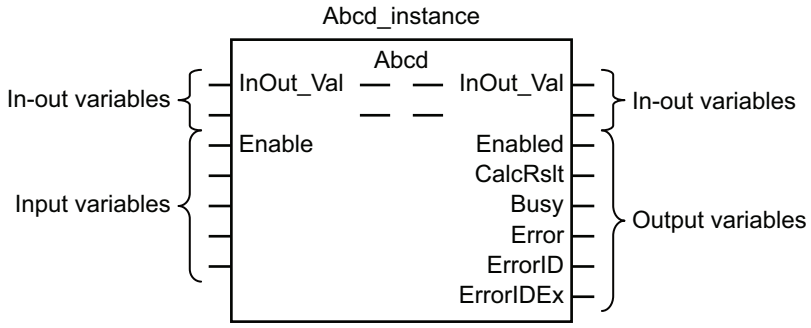


Errors



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.

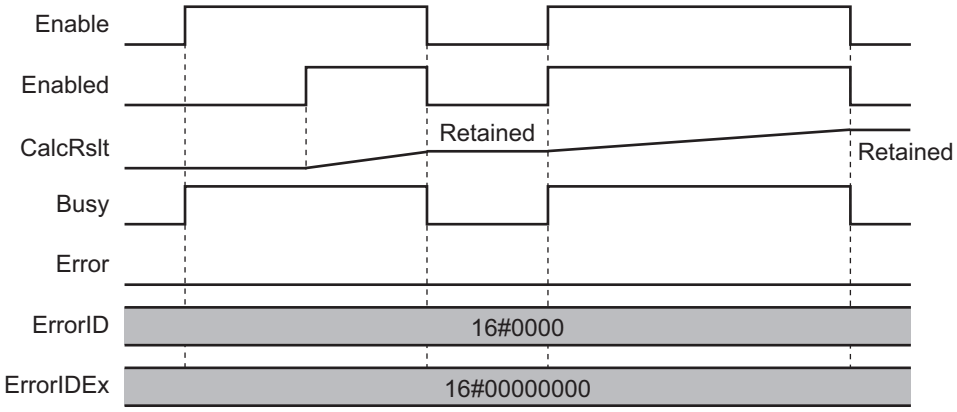


- If an error occurs, the relevant error code and expansion error code are set in *ErrorID* (Error Code) and *ErrorIDEx* (Expansion Error Code). The error codes are retained even after *Error* changes to FALSE, but *ErrorID* is set to 16#0000 and *ErrorIDEx* is set to 16#0000 0000 when *Enable* changes to TRUE.
- For function blocks that calculate the control amount for motion control, temperature control, etc., *Enabled* is FALSE when the value of *CalcRslt* (Calculation Result) is incorrect. In such a case, do not use *CalcRslt*. In addition, after the function block ends normally or after an error occurs, the value of *CalcRslt* is retained until *Enable* changes to TRUE. The control amount will be calculated based on the retained *CalcRslt* value, if it is the same instance of the function block that changed *Enable* to TRUE. If it is a different instance of the function block, the control amount will be calculated based on the initial value.

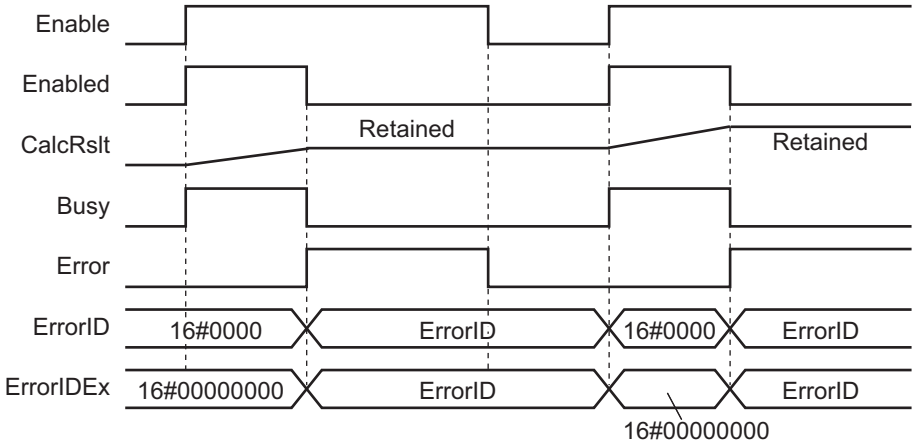
Timing Charts

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

● Normal End



Errors


Enable
Enabled
CalcRslt
Busy
Error
ErrorID
ErrorIDEx
Retained
Retained
16#0000
ErrorID
16#0000
ErrorID
16#00000000
ErrorID
ErrorID
16#00000000

Precautions

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

Nesting

You can nest calls to this function block for up to four levels.

For details on nesting, refer to the software user's manual.

Instruction Options

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

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.

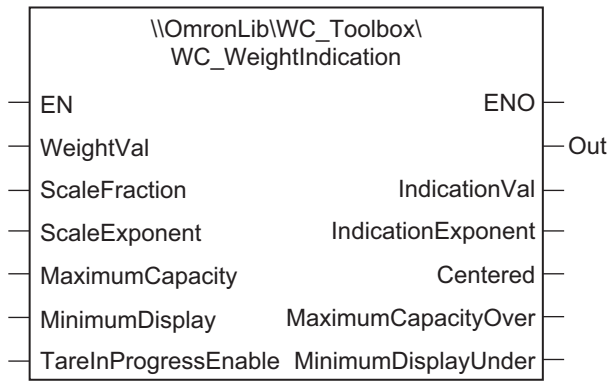
For details on re-execution, refer to the motion control user's manual.

Individual Specifications of Function Blocks

Function block name	Name	Page
WC_WeightIndication	Display Value Conversion	P.34
WC_FixedWeightCtrl	Quantitative Discharging Control	P.42
WC_FallCompensation	Fall Compensation	P.65

WC_WeightIndication

The WC_WeightIndication function block converts a measurement value to a display value to be shown on the HMI.

Function block name	Name	FB/ FUN	Graphic expression	ST expression
WC _Weight Indication	Display Value Con- version	FUN		Out:=WC_WeightIndication(WeightVal, ScaleFraction, ScaleExponent, MaximumCapacity, MinimumDisplay, TareInProgressEnable, IndicationVal, IndicationExponent, Centered, MaximumCapacityOver, MinimumDisplayUnder);

Function Block and Function Information

Item	Description
Library file name	OmronLib_WC_Toolbox_V1_0.slr
Namespace	OmronLib\WC_Toolbox
Function block and function number	00045
Publish/Do not publish source code	Do not publish
Function block and function version	1.00

Compatible Models

Item	Name	Model numbers	Version
Device	Load Cell Input Unit	NX-RS1201	Version 1.00 or later

Variables

Input Variables

Name	Meaning	Data type	Default	Valid range	Unit	Description
EN	Execute	BOOL	FALSE	Depends on data type.	---	Execute trigger for this function Executes the function when it changes to TRUE.
WeightVal	Weight Value	REAL	0	-3.402823e + 38 < WeightVal < 3.402823e + 38	User defined*1	Measurement value
Scale Fraction	Scale Interval Mantissa	INT	1	1,2,5	User defined*1	Mantissa of scale interval*2
Scale Exponent	Scale Interval Exponent	UDINT	0	-3 to 2	User defined*1	Exponent of scale interval*2
Maximum Capacity	Maximum Capacity Coefficient	UDINT	100	1 to 100000	---	This value indicates how many times the scale interval is multiplied to arrive at the maximum capacity.*3
Minimum Display	Minimum Display Coefficient	UDINT	20	If *TareInProgress Enable* = TRUE: 1 to 100000 If *TareInProgress Enable* = FALSE: 1 to 20	---	This value indicates how many times the scale interval × (-1) is multiplied to arrive at the minimum displayable quantity.*4
TareIn Progress Enable	Tare In Progress Enable	BOOL	FALSE	Depends on data type.	---	This flag indicates if the tare subtraction function is currently performed by the Load Cell Input Unit. TRUE: The tare subtraction function is currently performed. FALSE: The tare subtraction function is not currently performed.

*1. The same unit is required for *WeightVal*, *MaximumCapacity*, *MinimumDisplay* and the scale interval.

*2. For example, when the scale interval is 0.001 (1×10^{-3}), *ScaleFraction* = INT#1, and *ScaleExponent* = INT#-3.

*3. For example, when the scale interval is 0.1 and *MaximumCapacity* = UDINT#100, the maximum capacity is 10.0.

*4. For example, when the scale interval is 0.1 and *MinimumDisplay* = UDINT#20, the minimum displayable quantity is -2.0.

Output Variables

Name	Meaning	Data type	Valid range	Unit	Description
ENO	Done	BOOL	Depends on data type.	---	Always TRUE
Out	Return Value	BOOL	Depends on data type.	---	Return Value TRUE: Normal end FALSE: Error end
IndicationVal	Display Value Mantissa	STRING	0 to 9, -, ., NULL A string consists of at least 10 characters.	User defined	Mantissa of display value
IndicationExponent	Display Value Exponent	INT	Depends on data type.	User defined	Exponent of display value
Centered	Center Value Judgment Result	BOOL	Depends on data type.	---	This flag indicates if the rounding error is one-fourth the display value resolution or less. TRUE: The rounding error is one-fourth the display value resolution or less. FALSE: The rounding error is greater than one-fourth the display value resolution.
Maximum CapacityOver	Maximum Capacity Over Judgment Result	BOOL	Depends on data type.	---	This flag indicates if the maximum capacity + scale interval × 9 is exceeded. TRUE: The display value is greater than the maximum capacity + scale interval × 9. FALSE: The display value is equal to or smaller than the maximum capacity + scale interval × 9.
Minimum DisplayUnder	Minimum Display Under Judgment Result	BOOL	Depends on data type.	---	This flag indicates if the display value became smaller than the minimum displayable quantity. TRUE: The display value is smaller than the minimum displayable quantity. FALSE: The display value is equal to or greater than the minimum displayable quantity.

Function

This function block converts the *WeightVal* (Weight Value) to a display value to be shown on the HMI.

When converting a value, this function block also makes center value judgment, maximum capacity over judgment and minimum display under judgment.

Conversion from WeightVal (Weight Value) to Display Value

A *WeightVal* (Weight Value) of REAL type is converted to a display value of STRING type. The display resolution conforms to the setting for scale interval. If the scale interval is 0.1, for example, the display resolution becomes 0.1 and the value is indicated in units of 0.1.

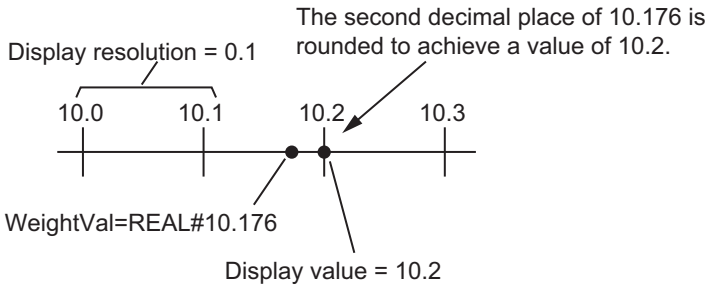
If the *WeightVal* resolution is different from the display resolution, value conversion takes place as follows.

● WeightVal Resolution Higher Than Display Resolution

If the *WeightVal* resolution is higher than the display resolution, the *WeightVal* value is rounded according to the display resolution.

If the *WeightVal* resolution is 0.001 and the display resolution is 0.1, for example, the second decimal point of the *WeightVal* value is rounded as it is converted to a display value.
If *WeightVal* = REAL#10.176, then display value becomes 10.2.

The relationship of *WeightVal* and the display value is shown in the following figure.



● WeightVal Resolution Lower Than Display Resolution

If the *WeightVal* resolution is lower than the display resolution, the *WeightVal* value becomes a display value as is.

For example, assume that the *WeightVal* resolution is 0.1 and display resolution is 0.01. If *WeightVal* = REAL#12.6, then the display value bocomes 12.6.

IndicationVal (Display Value Mantissa) and IndicationExponent (Display Value Exponent)

Display value is expressed with *IndicationVal* (Display Value Mantissa) in which the mantissa is expressed by STRING data and *IndicationExponent* (Display Value Exponent) in which the exponent is expressed by INT data.

In those, *IndicationExponent* is determined so that the rightest character cannot be '0' which is not valid. Specifically, the value for *IndicationExponent* is determined from the value for *ScaleExponent* (Scale Interval Exponent) as follows.

ScaleExponent value	IndicationExponent value
-3,-2,-1,0	0
1,2	3

The following example shows how the display value changes according to scale interval setting when the converted value is 1800.0. The value for *IndicationExponent* is always 0 or 3 as shown in the upper table.

Scale interval value	ScaleFraction value	ScaleExponent value	Display value	IndicationVal value	IndicationExponent value
1×10^{-3}	1	-3	1800.000×10^{0}	'1800.000'	0
1×10^{-2}	1	-2	1800.00×10^{0}	'1800.00'	0
1×10^{-1}	1	-1	1800.0×10^{0}	'1800.0'	0
1×10^{0}	1	0	1800×10^{0}	'1800'	0
1×10^{1}	1	1	1.80×10^{3}	'1.80'	3
1×10^{2}	1	2	1.8×10^{3}	'1.8'	3

Center Value Judgment

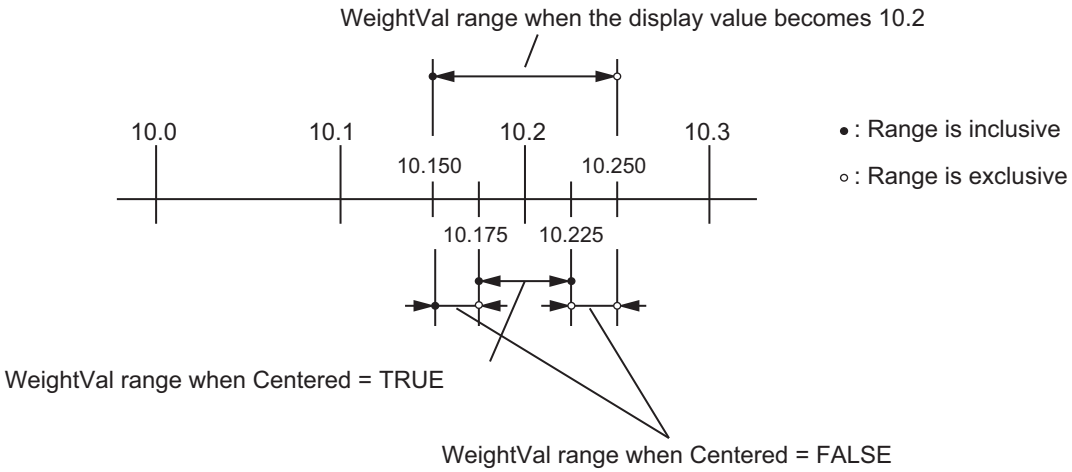
Center value judgment refers to determining, when the *WeightVal* (Weight Value) is rounded, whether the difference between the *WeightVal* and display value, or specifically the rounding error, is sufficiently small.

Whether the rounding error is sufficiently small is determined based on whether the rounding error is one-fourth the display resolution or less. *Centered* (Center Value Judgment Result) is the output variable indicating the judgment result of center value.

The relationship of the difference between *WeightVal* and the display resolution, and the *Centered* value, is as follows.

Difference between WeightVal and display resolution	Centered value
One-fourth the display resolution or less	TRUE
Over one-fourth the display resolution	FALSE

As an example, the relationship of *WeightVal* and *Centered* when the display resolution is 0.1 and the display value is 10.2 is shown below. The *Centered* value is TRUE when the *WeightVal* value is 10.2 + 0.1/4 = 10.225 or less and 10.2 − 0.1/4 = 10.175 or more.



Maximum Capacity Over Judgment

Maximum capacity over judgment refers to determining whether the converted value exceeds the preset threshold of maximum capacity.

The threshold of maximum capacity is indicated by *MaximumCapacity* (Maximum Capacity Coefficient) × scale interval + scale interval × 9.

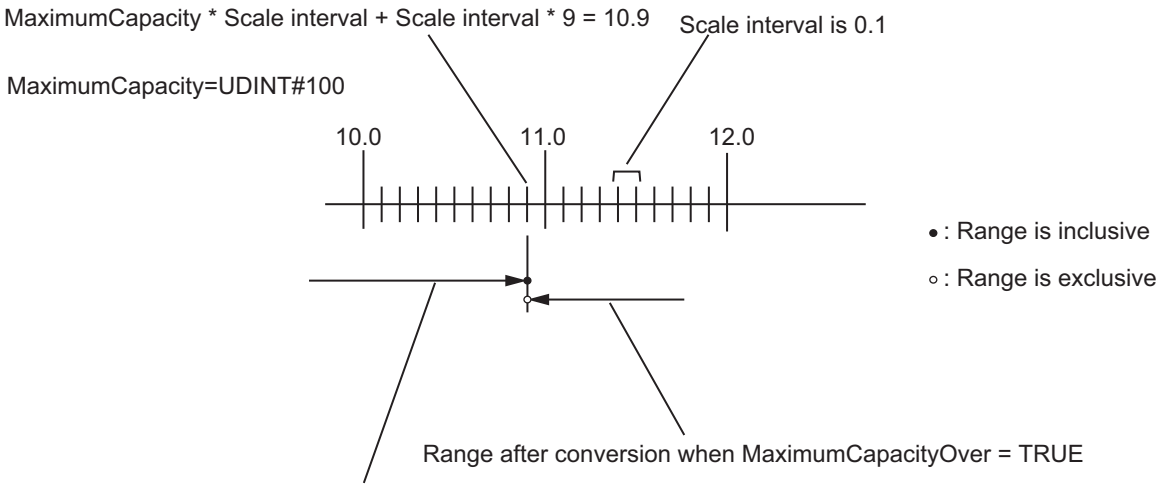
MaximumCapacityOver (Maximum Capacity Over Judgment Result) is the output variable indicating the judgment result of maximum capacity over.

If the converted value exceeds the threshold maximum capacity, the *IndicationValue* (Display Value Mantissa) becomes " (string consisting of 0 character).

The relationship between the converted value and *MaximumCapacityOver* value is as follows.

Converted value	MaximumCapacityOver value
Exceeds MaximumCapacity × scale Interval + scale interval × 9.	TRUE
Equal to or less than MaximumCapacity × scale Interval + scale interval × 9.	FALSE

As an example, the relationship of *WeightVal* and *MaximumCapacityOver* when *MaximumCapacity* = UDINT#100 and the scale interval is 0.1 is shown below. The *MaximumCapacityOver* value becomes TRUE if the converted value exceeds 10.9.



Minimum Display Under Judgment

Minimum display under judgment refers to determining whether the converted value is less than the preset minimum displayable quantity.

The minimum displayable quantity is indicated by *MinimumDisplay* (Minimum Display Coefficient) × Scale interval × (-1).

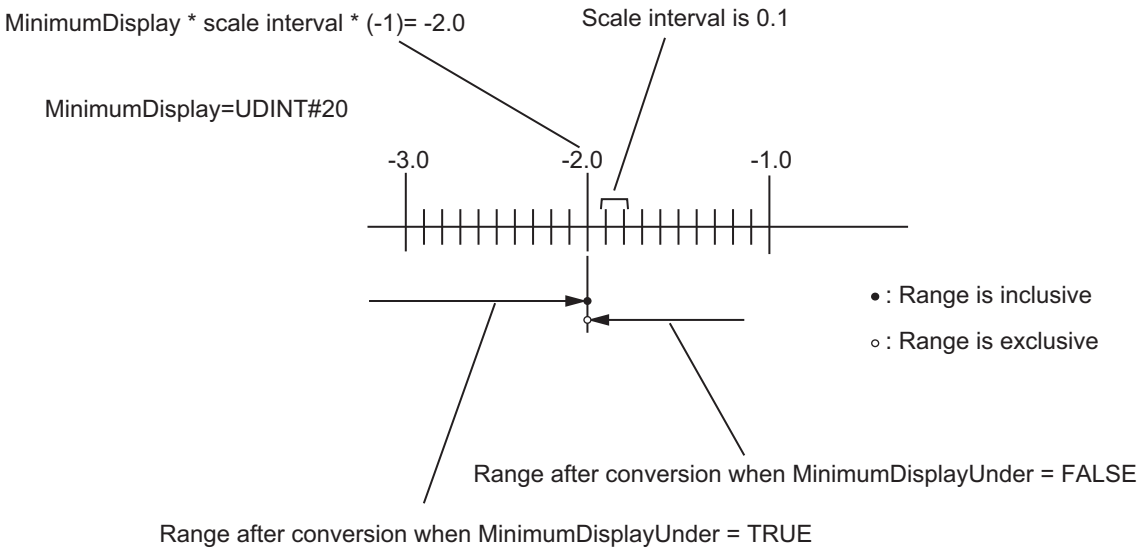
MinimumDisplayUnder (Minimum Display Under Judgment Result) is the output variable indicating the judgment result of minimum display under.

If the converted value is less than the minimum displayable quantity, the *IndicationValue* (Display Value Mantissa) value becomes " (string consisting of 0 character).

The relationship between the converted value and *MinimumDisplayUnder* value is as follows.

Converted value	MinimumDisplayUnder value
Less than *MinimumDisplay* × Scale interval × (-1).	TRUE
Equal to or greater than *MinimumDisplay* × Scale interval × (-1).	FALSE

As an example, the relationship of *WeightVal* and *MinimumDisplayUnder* when *MinimumDisplay* = UDINT#20 and the scale interval is 0.1 is shown below. The *MinimumDisplayUnder* value becomes TRUE if the converted value is less than -2.0.



Upper Limit of MinimumDisplay (Minimum Display Coefficient) when the Tare Subtraction is Disabled

TareInProgressEnable (Tare In Progress Enable) is a flag that indicates if the tare subtraction function is currently performed by the Load Cell Input Unit. If the tare subtraction function is currently performed, the *TareInProgressEnable* value should be TRUE.

If the tare subtraction function is not executed, the value for *WeightVal* (Weight Value) does not normally become significantly below REAL#0.0. This means that, if the *TareInProgressEnable* value is FALSE, the upper limit of *MinimumDisplay* (Minimum Display Coefficient) becomes UDINT#20. If the *TareInProgressEnable* value is FALSE and a value greater than UDINT#20 is set for *MinimumDisplay*, *MinimumDisplay* is set to UDINT#20. In this case, an error will not occur.

Additional Information

How to decide the values for *IndicationVal* (Display Value Mantissa) and *IndicationExponent* (Display Value Exponent) conforms to OIML R76-1.

Precautions for Correct Use

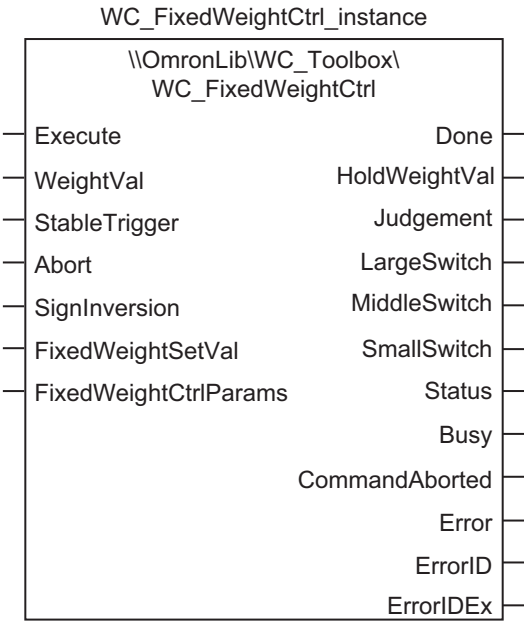
- The unit of *WeightVal* (Weight Value) and that of *ScaleInterval* (Scale Interval) should be the same.
- An error will occur in the following cases. *Out* becomes FALSE and the value for *IndicationValue* (Display Value Mantissa) becomes " (string consisting of 0 character).
  a) The value for *WeightVal* (Weight Value) is out of the valid range.
  b) The value for *ScaleFraction* (Scale Interval Mantissa) is out of the valid range.
  c) The value for *ScaleExponent* (Scale Interval Exponent) is out of the valid range.
  d) The value for *MaximumCapacity* (Maximum Capacity Coefficient) is out of the valid range.
  e) The value for *MinimumDisplay* (Minimum Display Coefficient) is out of the valid range.

Sample Programming

Refer to the sample programming for *WC_FixedWeightCtrl* on page 42.

WC_FixedWeightCtrl

The WC_FixedWeightCtrl function block performs quantitative discharging control for feed weighing and discharge weighing.

Function block name	Name	FB/ FUN	Graphic expression	ST expression
WC_Fixed WeightCtrl	Quantita-tive Dis-charging Control	FB		WC_FixedWeightCtrl_instance(Execute, WeightVal, StableTrigger, Abort, SignInversion, FixedWeightSetVal, FixedWeightCtrlParams, Done, HoldWeightVal, Judgement, LargeSwitch, MiddleSwitch, SmallSwitch, Status, Busy, CommandAborted, Error, ErrorID, ErrorIDEx);

Function Block and Function Information

Item	Description
Library file name	OmronLib_WC_Toolbox_V1_0.slr
Namespace	OmronLib\WC_Toolbox
Function block and function number	00046
Publish/Do not publish source code	Do not publish
Function block and function version	1.00

Compatible Models

Item	Name	Model numbers	Version
Device	Load Cell Input Unit	NX-RS1201	Version 1.00 or later

Variables

Input Variables

Name	Meaning	Data type	Default	Valid range	Unit	Description
Execute	Execute	BOOL	FALSE	Depends on data type.	---	Execute trigger for this function block Executes the function block when it changes to TRUE.
WeightVal	Weight Value	REAL	0.0	-3.402823e + 38 < WeightVal < 3.402823e + 38	User defined*1	Measurement value
Stable Trigger	Stable Trigger	BOOL	FALSE	Depends on data type.	---	Ch1 Stable Status that is input from the Load Cell Input Unit.
Abort	Abort	BOOL	FALSE	Depends on data type.	---	Abort trigger for this function block Aborts the function block when it changes to TRUE.
Sign Inversion	Discharging Method	BOOL	FALSE	Depends on data type.	---	Discharging method TRUE: Discharge weighing FALSE: Feed weighing
FixedWeight SetVal	Fixed Weight Set Value	REAL	1.0	Depends on data type.	User defined*1	Fixed weight set value
FixedWeight CtrlParams	Quantitative Discharging Control Settings	OmronLib\WC_Toolbox\sFIXED_WEIGHT_CTRL_PARAMS	---	---	---	Set values for quantitative discharging control

*1. The same unit is required for *WeightVal* and *FixedWeightSetVal*.

Structures

The data type of the *FixedWeightCtrlParams* input variable is structure OmronLib\WC_Toolbox\sFIXED_WEIGHT_CTRL_PARAMS. The specifications are as follows:

Name	Meaning	Description	Data type	Valid range	Unit	Default
Fixed WeightCtrl Paramas	Quantitative Discharging Control Settings	Set values for quantitative discharging control	OmronLib\WC_Toolbox\sFIXED_WEIGHT_CTRL_PARAMS	---	---	---
Large Input Weight	Large Feed Set Value	A value that is acquired when you subtract the weight of raw materials that are fed by the end of large feed from *FixedWeightSetVal*.	REAL	0, 1.175495e - 38 to 3.402823e + 38	User defined[*1]	0
PreFixed Weight	Pre-fixed Weight Set Value	A value that is acquired when you subtract the weight of the raw material that is fed by the end of medium feed from *FixedWeightSetVal*.	REAL	0, 1.175495e - 38 to 3.402823e + 38	User defined[*1]	0
FallComp Weight	Fall Set Value	A value that is acquired when you subtract the weight of the raw material that is fed by the end of small feed from *FixedWeightSetVal*.	REAL	1.175495e - 38 to 3.402823e + 38	User defined[*1]	0
OverSet Weight	Weight Over Set Value	Threshold to judge whether or not the fixed weight is exceeded.	REAL	0, 1.175495e - 38 to 3.402823e + 38[*2]	User defined[*1]	0
ShortSet Weight	Weight Short Set Value	Threshold to judge whether or not there is a shortage.	REAL	0, 1.175495e - 38 to 3.402823e + 38[*2]	User defined[*1]	0
CompBan Time1	Comparison Banned Time 1	The time when comparison between *FixedWeightSetVal* and *WeightVal* is not performed immediately after the valve for large feed is changed to the valve for medium feed and for medium feed to small feed	TIME	0 to 9223372036854.775807	ms	0

Name	Meaning	Description	Data type	Valid range	Unit	Default
CompBanTime2	Comparison Banned Time 2	The time when comparison between *FixedWeightSetVal* and *WeightVal* is not performed immediately after the valve is changed at the end of small feed	TIME	0 to 9223372036854.775807	ms	0
JudgeTime	Judgment Timeout Time	Timeout time from the end of small feed until the judgment is performed	TIME	0 to 9223372036854.775807	ms	0
MonitoringTime	Execution Timeout Time	The execution timeout time of this function block	TIME	0 to 9223372036854.775807*3	ms	0

*1. The unit that is used for *LargeInputWeight*, *PreFixedWeight*, *FallCompWeight*, *OverSetWeight*, and *ShortSetWeight* must be the same as for *FixedWeightSetVal* and *WeightVal*.

*2. When *OverSetWeight* = REAL#0, whether or not the fixed weight is exceeded is not judged. In the same manner, when *ShortSetWeight* = REAL#0, whether or not there is a shortage is not judged.

*3. When TIME#0s is set, the execution timeout time is set to 60 s and processed.

Output Variables

Name	Meaning	Data type	Valid range	Unit	Description
Done	Done	BOOL	Depends on data type.	---	Execution done TRUE: Normal end FALSE: Error end, execution in progress, or execution condition not met
HoldWeightVal	Held Weight Value	REAL	Depends on data type.	*1	Measurement value that is held after judgment
Judgement	Judgment Result	UINT	0 to 3	---	Judgment result 0: Not yet judged 1: The judgment result shows that the fixed weight value is exceeded. 2: The judgment result shows that the weight is short. 3: The judgment result shows that the weight is normal.
LargeSwitch	Large Feed Switch	BOOL	Depends on data type.	---	A flag that indicates that large feed is performed. TRUE: Large feed is performed. FALSE: Large feed is not performed.
MiddleSwitch	Medium Feed Switch	BOOL	Depends on data type.	---	A flag that indicates that medium feed is performed. TRUE: Medium feed is performed. FALSE: Medium feed is not performed.
SmallSwitch	Small Feed Switch	BOOL	Depends on data type.	---	A flag that indicates that small feed is performed. TRUE: Small feed is performed. FALSE: Small feed is not performed.
Status	Status	UINT	0 to 4	---	Processing status 0: Before execution 1: Large feed 2: Medium feed 3: Small feed 4: Waiting for judgment
Busy	Executing	BOOL	Depends on data type.	---	Executing TRUE: Execution processing is in progress. FALSE: Execution processing is not in progress.
CommandAborted	Command Aborted	BOOL	Depends on data type.	---	Command aborted Changes to TRUE when the command is aborted.

Name	Meaning	Data type	Valid range	Unit	Description
Error	Error	BOOL	Depends on data type.	---	Error end TRUE: Error end FALSE: Normal end, execution in progress, or execution condition not met
ErrorID	Error Code	WORD	*2	---	Error code when an error occurs WORD#16#0 when normal
ErrorIDEx	Expansion Error Code	DWORD	*2	---	Expansion error code when an error occurs DWORD#16#0 when normal

*1. Same unit as *FixedWeightSetVal* and *WeightVal*.

*2. Refer to *Troubleshooting* on page 56 for details.

Function

This function block performs quantitative discharging control that discharges a constant quantity of the raw material.

Also, after the raw material is discharged, it compares the net weight of the raw material and the fixed weight set value to judge whether the weight is normal, exceeded, or short.

Parameter Input from Load Cell Input Unit

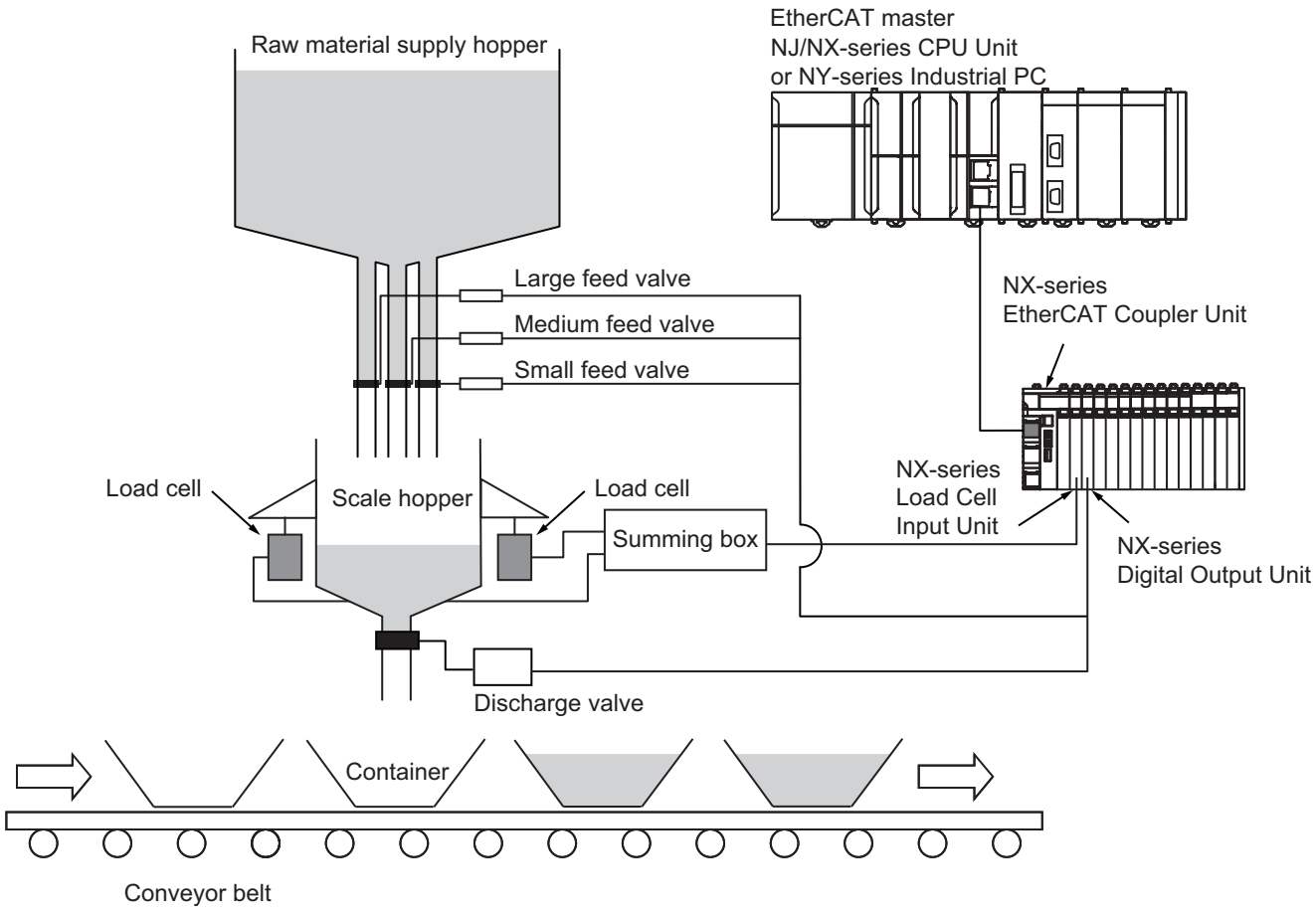
Enter the Ch1 Net Weight Value REAL of the Load Cell Input Unit for *WeightVal* (Weight Value). Also, enter the Ch1 Stable Status data of the Load Cell Input Unit in *StableTrigger* (Stable Trigger).

It is necessary to perform I/O allocation of these two types of data through the settings of the Load Cell Input Unit in order to enter the value. Refer to the *NX-series Load Cell Input Unit User's Manual* (Cat. No. W565) for details on the Load Cell Input Unit specifications.

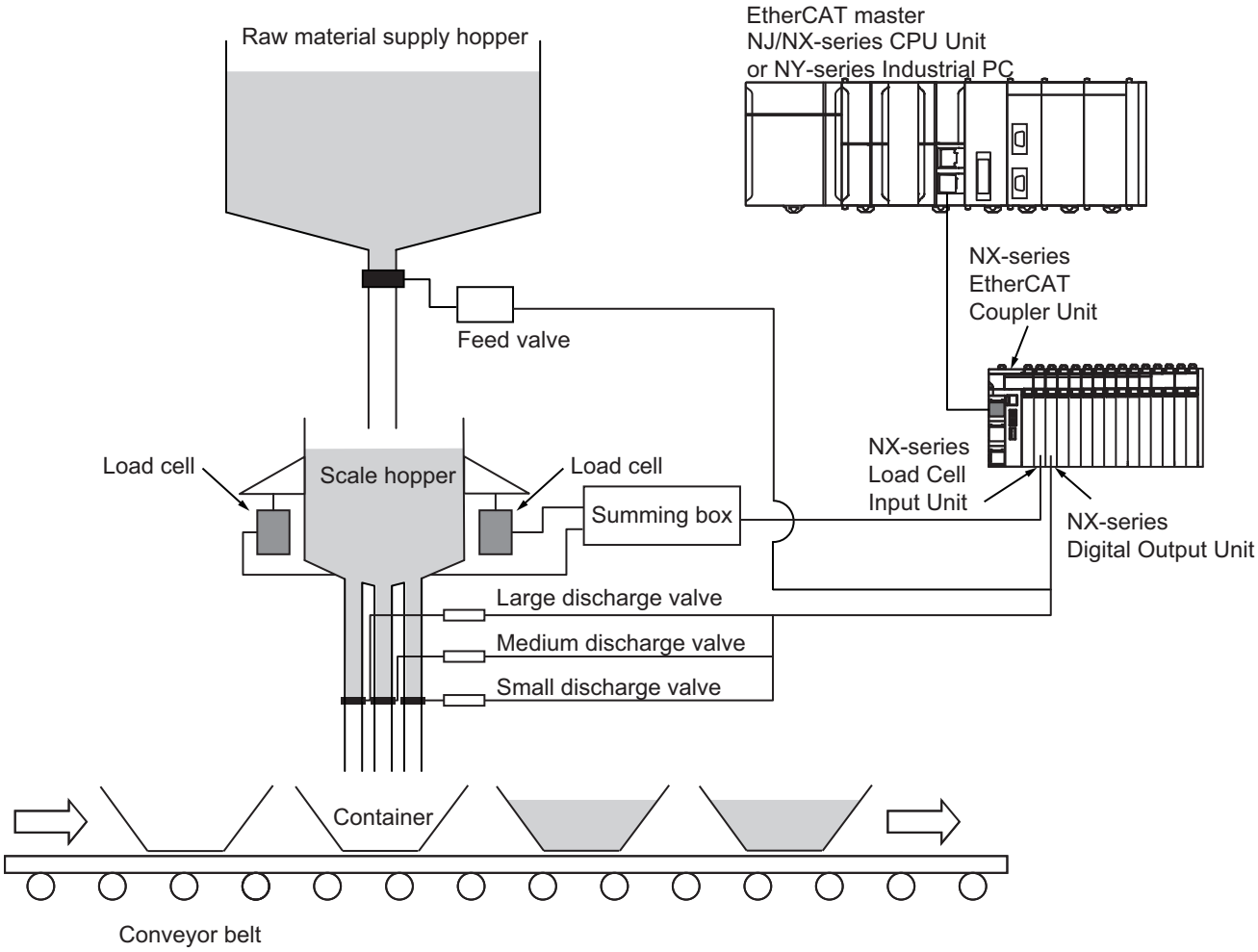
Feed Weighing and Discharge Weighing

There are two ways of discharging raw materials: Feed weighing and discharge weighing

With feed weighing, the raw material is fed to the scale hopper and once the net weight in the scale hopper reaches the fixed weight, feeding stops and the raw material is discharged into a container. As the raw material is fed to the scale hopper, the net weight of the raw material in the scale hopper increases.



With discharge weighing, the raw material that was loaded into the scale hopper beforehand is discharged into a container and once the net weight of the discharged raw material reaches the fixed weight, discharging stops. As the raw material is discharged from the scale hopper, the net weight of the raw material in the scale hopper decreases.



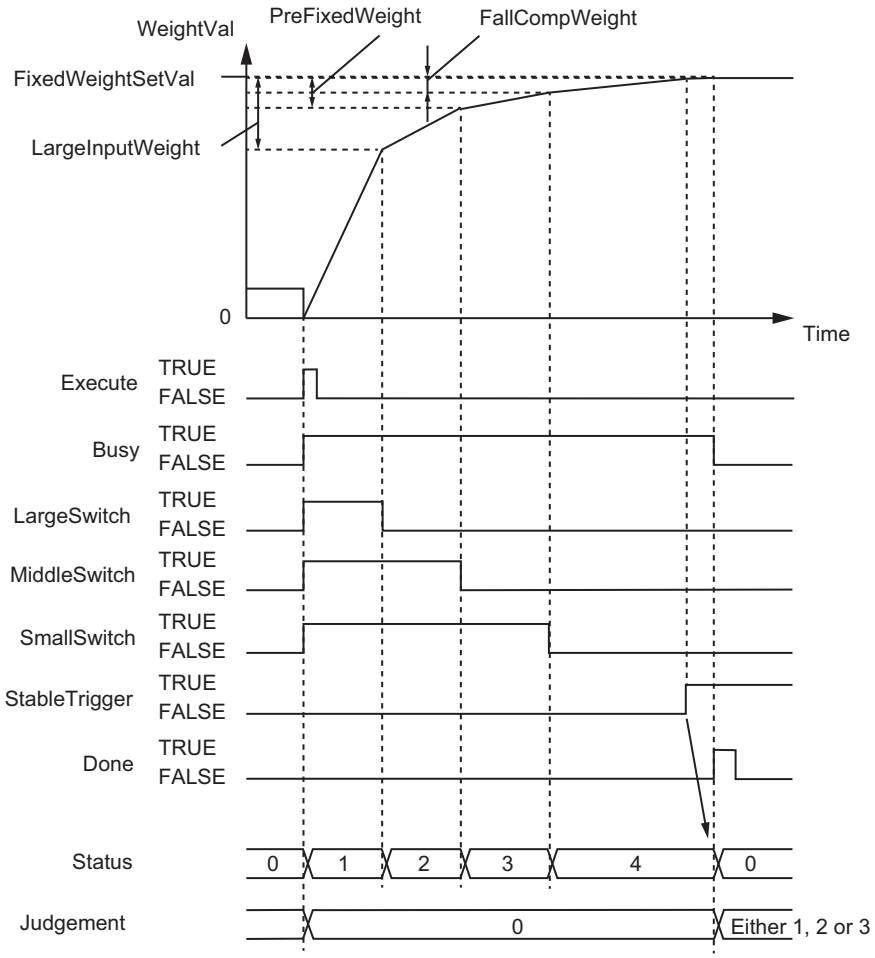
Quantitative Discharging Control of Feed Weighing

This function block is designed on the assumption that the feed valve for raw materials that is located upstream of the scale hopper is opened in three levels: large, medium, and small. The opening of the feed valve is set to large for large feed, medium for medium feed, and small for small feed. Gradually decrease the feed amount from large feed → medium feed → small feed to bring the net weight in the scale hopper close to the fixed weight.

There are five processing statuses: Before execution, Large feed, Medium feed, Small feed, and Waiting for judgment The status of the feed device, related parameter values, and conditions for moving to the next status in each processing status are shown in the following table.

Status	Device status	Status value	LargeSwitch, MiddleSwitch, and SmallSwitch values	Condition for moving to the next status
Before execution	• The feed valve is closed. • Weighing is not performed.	0	L: FALSE M: FALSE S: FALSE	The user changes the *Execute* value from FALSE to TRUE.
Large feed	• *WeightVal* value is initialized to 0. • Large feed is performed. • Weighing is performed for each task period.	1	L: TRUE M: TRUE S: TRUE	*WeightVal* ≥ *FixedWeightSetVal* − *LargeInputWeight*
Medium feed	• Medium feed is performed. • Weighing is performed for each task period.	2	L: FALSE M: TRUE S: TRUE	*WeightVal* ≥ *FixedWeightSetVal* − *PreFixedWeight*
Small feed	• Small feed is performed. • Weighing is performed for each task period.	3	L: FALSE M: FALSE S: TRUE	*WeightVal* ≥ *FixedWeightSetVal* − *FallCompWeight*
Waiting for judgment	• The feed valve is closed. • Weighing is performed for each task period. • Waiting for the *WeightVal* value to be stabilized.	4	L: FALSE M: FALSE S: FALSE	The value for *StableTrigger* changes from FALSE to TRUE.
Before execution	• The feed valve is closed. • Weighing is not performed. • *WeightVal* judgment is performed and *Judgement* value is finalized. • After the lapse of time that was set for *JudgeTime* since Waiting for judgment status started, *Done* changes to TRUE. • The discharge valve opens and the raw material in the scale hopper is discharged.	0	L: FALSE M: FALSE S: FALSE	The user changes the *Execute* value from FALSE to TRUE.

How the parameter values change is shown in the following figure.

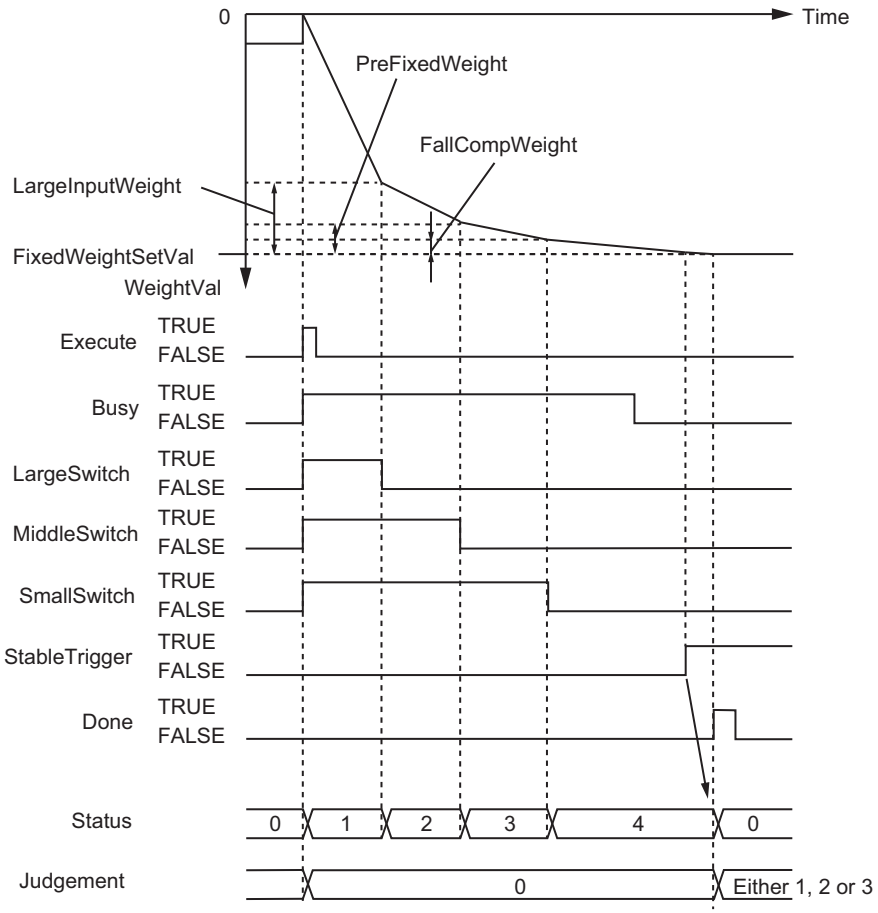


Quantitative Discharging Control of Discharge Weighing

This function block is designed on the assumption that the discharge valve for raw materials that is located downstream of the scale hopper is opened in three levels: large, medium, and small.

For discharge weighing, change the value for *SignInversion* (Discharge Method) to TRUE before you change the value for *Execute* (Execute) to TRUE. When the value for *SignInversion* is changed to TRUE, the processing is performed with the code for the value for *WeightVal* (Weight Value) inverted, therefore meanings of parameters become the same as for feed weighing. Therefore, the processing procedures are the same as for feed weighing.

How the parameter values change is shown in the following figure.



Other Parameters

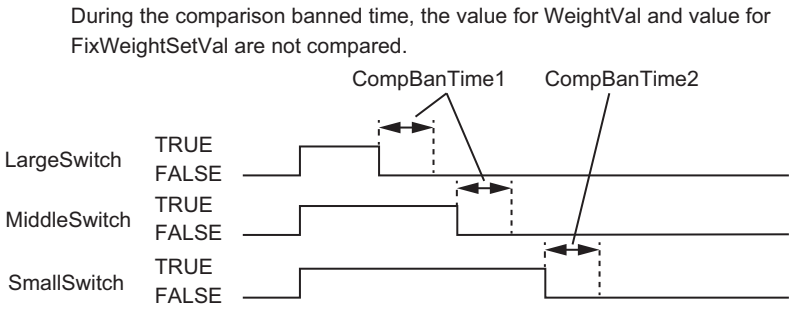
The meanings of other parameters are as follows.

● CompBanTime1 (Comparison Banned Time 1), and CompBanTime2 (Comparison Banned Time 2)

The value for *WeightVal* (Weight Value) may not be stable immediately after the valve is opened or closed due to mechanical vibrations or other factors. *CompBanTime1* (Comparison Banned Time 1) and *CompBanTime2* (Comparison Banned Time 2) are input parameters that are used to set the time when the values for *WeightVal* and *FixWeightSetVal* (Fixed Weight Set Value) are not compared after the valve is opened or closed.

After either one of the value for *LargeSwitch* (Large Feed Switch) and value for *MiddleSwitch* (Medium Feed Switch) changes to FALSE, the value for *WeightVal* and *FixWeightSetVal* value are not compared during the time that was set for *CompBanTime1*. In the same manner, after the value for *SmallSwitch* (Small Feed Switch) changes to FALSE, the value for *WeightVal* and value for *FixWeightSetVal* are not compared during the time that was set for *CompBanTime2*.

The figure below shows how *LargeSwitch*, *MiddleSwitch*, and *SmallSwitch* are related to *CompBanTime1* and *CompBanTime2*.



When *CompBanTime1* = T#0s and *CompBanTime2* = T#0s, the *WeightVal* value and *FixWeightSetVal* value are compared immediately after the valve is opened or closed.

● OverSetWeight (Weight Over Set Value), ShortSetWeight (Weight Short Set Value), and Judgement (Judgment Result)

Thresholds for judging the net weight of the raw material. The value for *Judgement* (Judgment Result) is determined as follows according to the values for *OverSetWeight* (Weight Over Set Value), *ShortSetWeight* (Weight Short Set Value), and *WeightVal* (Weight Value) at the time of judgment execution.

Relations between parameters	Judgment result	Judgement value
FixWeightSetVal + OverSetWeight < WeightVal	Exceeded	UINT#1
WeightVal < FixWeightSetVal − ShortSetWeight	Short	UINT#2
FixWeightSetVal − ShortSetWeight ≤ WeightVal ≤ FixWeightSetVal + OverSetWeight	Normal	UINT#3

The value for *Judgement* is UINT#0 until judgment is performed after this function block is executed.

When value for *OverSetWeight* is REAL#0.0, whether or not the fixed weight is exceeded is not judged. In the same manner, when the value for *ShortSetWeight* is REAL#0.0, whether or not there is a shortage is not judged. Therefore, when the values for *OverSetWeight* and *ShortSetWeight* are both REAL#0.0, the value for *Judgement* (Judgment Result) is always UINT#3.

● JudgeTime (Judgment Timeout Time)

Timeout time from the end of small feed until the judgment is performed. If the value for *StableTrigger* (Stable Trigger) is not changed to TRUE within the time period that was set for *JudgeTime* after the value for *SmallSwitch* (Small Feed Switch) changes to FALSE, an error occurs.

When *JudgeTime* is T#0s, the waiting status for the judgment continues until the value for *StableTrigger* changes to TRUE.

● MonitoringTime (Execution Timeout Time)

The execution timeout time of this function block. If the value for *Done* (Done) does not change to TRUE within the time period that was set for *MonitoringTime* after the value for *Execute* (Execute) changes to TRUE, an error occurs.

This is used to detect an error that quantitative discharging control does not end within the time limit due to an invalid parameter or valve failure.

When *MonitoringTime* = T#0s, the execution timeout time is set to 60 s are processed.

● HoldWeightVal (Held Weight Value)

An output parameter that holds the value for *WeightVal* (Weight Value) at the point when the judgment is finished. The value for *HoldWeightVal* (Held Weight Value) is held until the next judgment is performed.

For discharge weighing, the value is output with the code inverted.

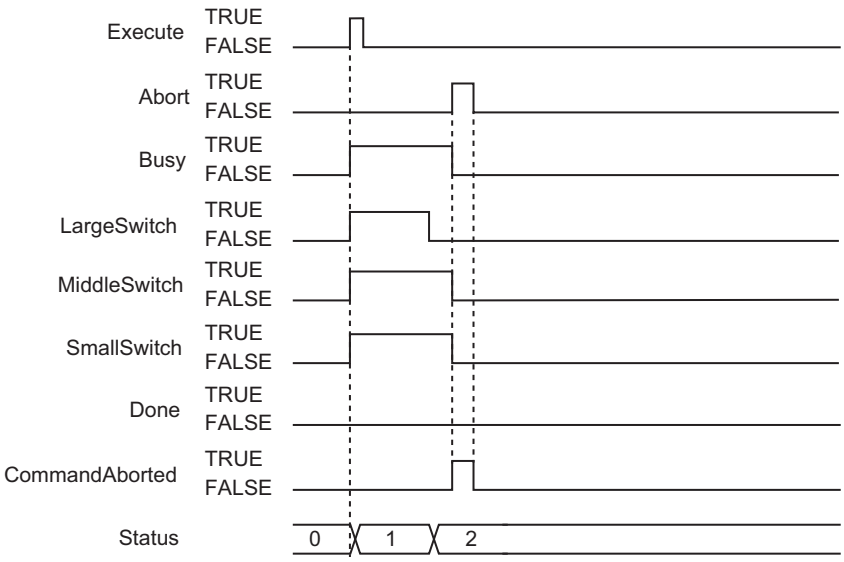
HoldWeightVal is valid when you combine this function block with WC_FallCompensation function block to perform quantitative discharging control with fall compensation. When it is combined with WC_FallCompensation function block, connect *HoldWeightVal* of this function block to the *WeightVal* input parameter of WC_FallCompensation function block.

Timing Chart

Timing charts for when processing is aborted and when an error occurs are shown below. Refer to *Quantitative Discharging Control of Feed Weighing* on page 50 for the timing chart for normal end.

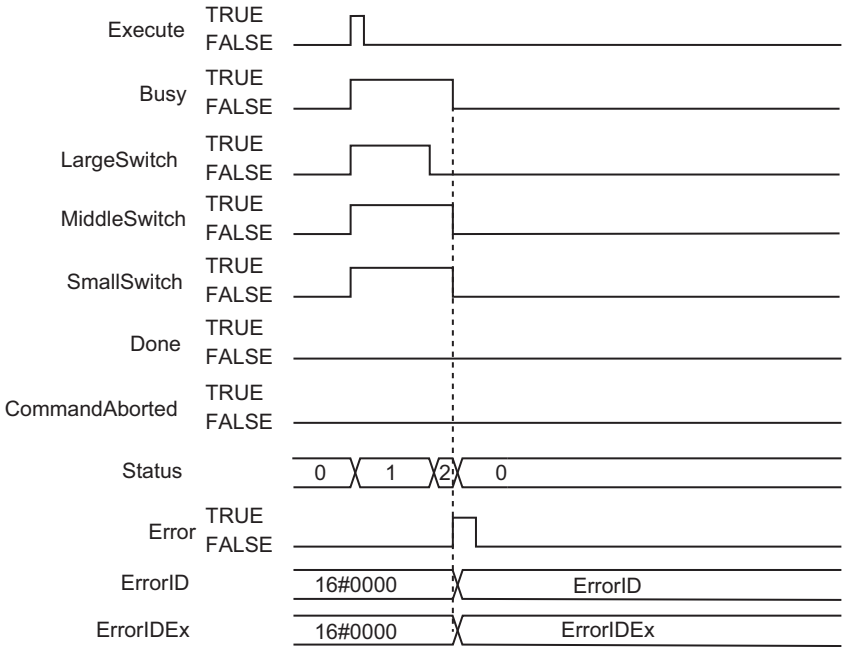
● When Processing is Aborted

- When the value for *Abort* (Abort) is changed to TRUE, the processing is aborted.
- The values for *LargeSwitch* (Large Feed Switch), *MiddleSwitch* (Medium Feed Switch), and *SmallSwitch* (Small Feed Switch) all change to FALSE.
- The value at the point when the processing was aborted is held as the value for *Status* (Status).



When an Error Occurs

- the value for *Error* changes to TRUE when an error occurs. A value based on the error is stored for *ErrorID* (Error Code) and *ErrorIDEx* (Expansion Error Code).
- The values for *LargeSwitch* (Large Feed Switch), *MiddleSwitch* (Medium Feed Switch), and *SmallSwitch* (Small Feed Switch) all change to FALSE.
- The value for *Status* (Status) changes to UINT#0.



Precautions for Correct Use

- The values at the point when *Execute* (Execute) changes to TRUE are valid for *FixedWeightSetVal* (Fixed Weight Set Value) and *FixedWeightCtrlParams* (Quantitative Discharging Control Settings). The value is not refreshed even if it is changed during processing of the function block.
- If the value for *WeightVal* (Weight Value) or *StableTrigger* (Stable Trigger) is changed while this function block is in process, the values are refreshed for the processing within the same task period.
- When the value for *LargeInputWeight* (Large Feed Set Value) is REAL#0.0. large feed is not performed. In the same manner, when the value for *PreFixedWeight* (Pre-fixed Weight Set Value) is REAL#0.0, medium feed is not performed.
- The data type for *LargeInputWeight* (Large Feed Value), *PreFixedWeight* (Pre-fixed Weight Set Value), *FallCompWeight* (Fall Set Value), *OverSetWeight* (Weight Over Set Value), and *ShortSetWeight* (Weight Short Set Value) is REAL. Therefore, the value is possibly inaccurate over 7 digits.

Troubleshooting

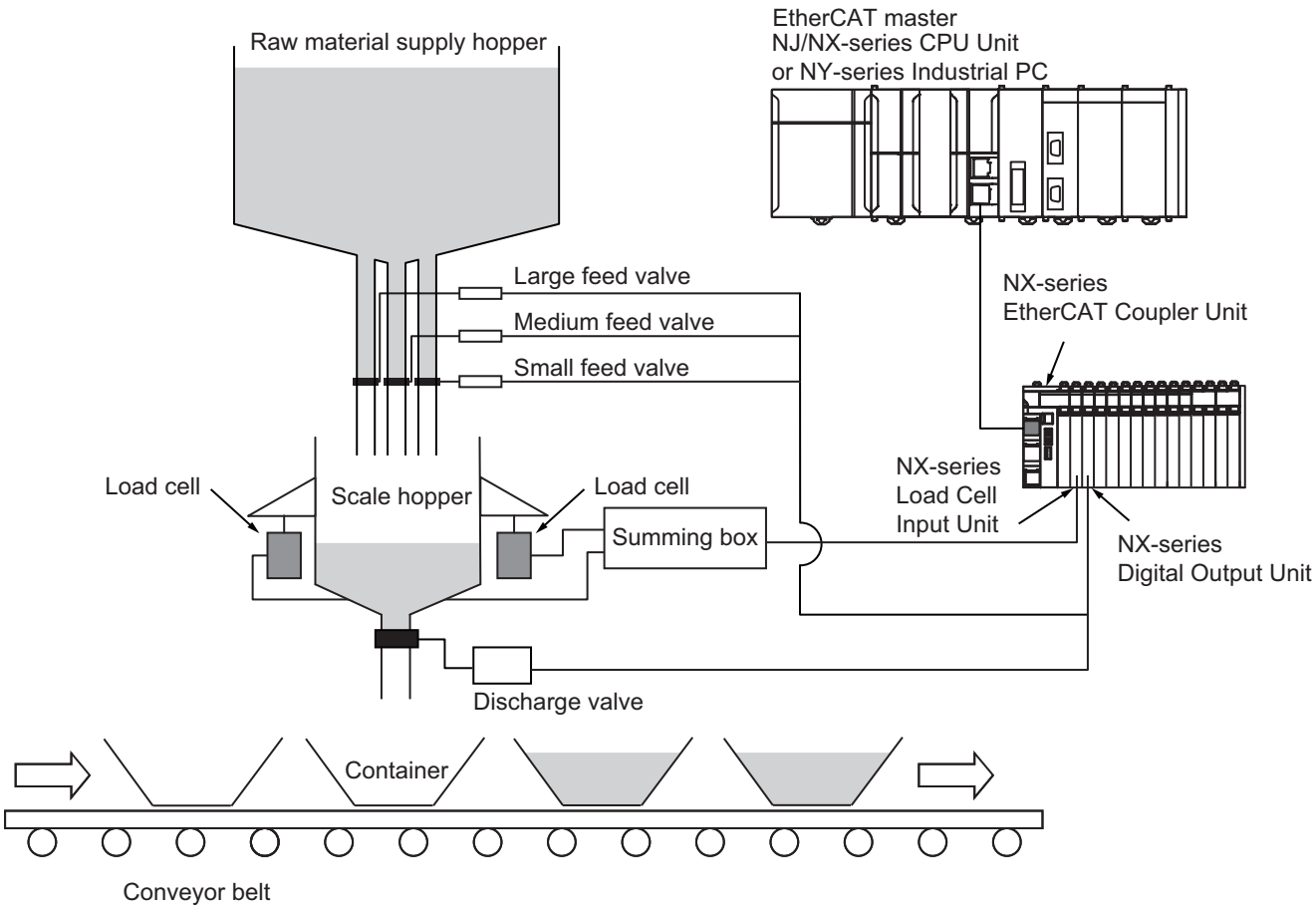
The error codes, expansion error codes, statuses, descriptions and corrections are as follows.

Error code	Expansion error code	Status	Description	Correction
16#0000	16#00000000	Normal End		
16#3C30	16#00000001	*FixedWeightSetVal* Outside Input Range	The value for *FixedWeightSetVal* (Fixed Weight Set Value) is outside the valid range.	Check the valid range of the value for *FixedWeightSetVal* (Fixed Weight Set Value) and set the value within the valid range.
	16#00000002	*LargeInputWeight* Outside Input Range	The value for *LargeInputWeight* (Large Feed Set Value) is outside the valid range.	Check the valid range of the value for *LargeInputWeight* (Large Feed Set Value) and set the value within the valid range.
	16#00000003	*PreFixedWeight* Outside Input Range	The value for *PreFixedWeight* (Pre-fixed Weight Set Value) is outside the valid range.	Check the valid range of the value for *PreFixedWeight* (Pre-fixed Weight Set Value) and set the value within the valid range.
	16#00000004	*FallCompWeight* Outside Input Range	The value for *FallCompWeight* (Fall Set Value) is outside the valid range.	Check the valid range of the value for *FallCompWeight* (Fall Set Value) and set the value within the valid range.
	16#00000005	*OverSetWeight* Outside Input Range	The value for *OverSetWeight* (Weight Over Set Value) is outside the valid range.	Check the valid range of the value for *OverSetWeight* (Weight Over Set Value) and set the value within the valid range.
	16#00000006	*ShortSetWeight* Outside Input Range	The value for *ShortSetWeight* (Weight Short Set Value) is outside the valid range.	Check the valid range of the value for *ShortSetWeight* (Weight Short Set Value) and set the value within the valid range.
	16#00000007	*CompBanTime1* Outside Input Range	The value for *CompBanTime1* (Comparison Banned Time 1) is outside the valid range.	Check the valid range of the value for *CompBanTime1* (Comparison Banned Time 1) and set the value within the valid range.
	16#00000008	*CompBanTime2* Outside Input Range	The value for *CompBanTime2* (Comparison Banned Time 2) is outside the valid range.	Check the valid range of the value for *CompBanTime2* (Comparison Banned Time 2) and set the value within the valid range.
	16#00000009	*JudgeTime* Outside Input Range	The value for *JudgeTime* (Judgment Timeout Time) is outside the valid range.	Check the valid range of the value for *JudgeTime* (Judgment Timeout Time) and set the value within the valid range.
	16#0000000A	*MonitoringTime* Outside Input Range	The value for *MonitoringTime* (Execution Timeout Time) is outside the valid range.	Check the valid range of the value for *MonitoringTime* (Execution Timeout Time) and set the value within the valid range.
	16#0000000B	Set Values for *LargeInputWeight*, *PreFixedWeight*, and *FallCompWeight* Invalid	The large and small relations among the values for *LargeInputWeight* (Large Feed Set Value), *PreFixedWeight* (Pre-fixed Set Value), and *FallCompWeight* (Fall Set Value) are not as follows. *LargeInputWeight* ≥ *PreFixedWeight* ≥ *FallCompWeight*	Set the values of three variables so that their relations are as follows. *LargeInputWeight* ≥ *PreFixedWeight* ≥ *FallCompWeight*

Error code	Expansion error code	Status	Description	Correction
16#3C30	16#0000000C	*WeightVal* Outside Input Range	The value for *WeightVal* (Weight Value) is out of the valid range.	Adjust the value so that Ch1 Net Weight Value REAL of the Load Cell Input Unit is input normally.
	16#0000000D	Judgment Timeout	During the period of *JudgeTime* (Judgment Timeout) after the end of small feed, the value for *StableTrigger* (Stable Trigger) did not change to TRUE.	• Check the value for *JudgeTime* (Judgment Timeout Time). • Adjust the value so that Ch1 Stable Status of the Load Cell Input Unit is input normally.
	16#0000000E	Execution Timeout	The value for *Done* (Done) did not change to TRUE during the time period of *MonitoringTime* (Execution Timeout Time) after the value for *Execute* (Execute) changed to TRUE.	• Check the value for *MonitoringTime* (Execution Timeout Time). • Check the value for output variable, and remove the cause that this function block is not complete such as feed valve clogging.

Sample Programming

The sample programming to perform quantitative discharging control for feed weighing is shown.



Unit Configuration

The following table shows the Unit configuration of the Controller.

Type of Unit	Model	Settings
EtherCAT master CPU Unit	NJ501-1500	---
EtherCAT Coupler Unit	NX-ECC203	Node address 1
NX Unit Load Cell Input Unit	NX-RS1201	NX unit No.:1
NX Unit Digital Output Unit	NX-OD3121	NX unit No.:2

I/O Map

The I/O maps for the Units are set as shown in the following tables.

● NX-RS1201

Port	Description	R/W	Data Type	Variable	Variable Comment	Variable Type
Ch1 Under Range	Under range error of Ch1	R	BOOL	WC1_UIO_Under_Range	Load Cell Input Unit Ch1 Under Range	Global variables
Ch1 Over Range	Over range error of Ch1	R	BOOL	WC1_UIO_Over_Range	Load Cell Input Unit Ch1 Over Range	Global variables
Ch1 Stable Status	Stable status of Ch1	R	BOOL	WC1_UIO__Stable_Status	Load Cell Input Unit Ch1 Stable Status	Global variables
Ch1 Net Weight Value REAL	Net weight value REAL of Ch1	R	REAL	WC1_UIO_Net_Weight_Value_REAL	Load Cell Input Unit Ch1 Net Weight Value	Global variables
Ch1 One-touch Tare Subtraction Execution	One-touch tare subtraction execution of Ch1	W	BOOL	WC1_UIO_OnetouchTare SubtractionExecution	Load Cell Input Unit Ch1 One-touch Tare Subtraction Execution	Global variables

- **NX-OD3121**

Port	Description	R/W	Data Type	Variable	Variable Comment	Variable Type
Output Bit 00	Output Bit 00	W	BOOL	WC1_UIO_LargeSwith	Large feed valve output signal	Global variables
Output Bit 01	Output Bit 01	W	BOOL	WC1_UIO_MiddleSwith	Medium feed valve output signal	Global variables
Output Bit 02	Output Bit 02	W	BOOL	WC1_UIO_SmallSwith	Small feed valve output signal	Global variables
Output Bit 03	Output Bit 03	W	BOOL	WC1_UIO_DischargeSwith	Discharge valve output signal	Global variables

Program Operation

Large feed, medium feed and small feed valves are controlled by the WC_FixedWeightCtrl function block which performs discharging control for feed weighing.
This function block also performs fall compensation in combination with the WC_FallCompensation function block.
After raw material is discharged, and the judgment result for the net weight is normal. In this case, use the WC_WeightIndication function to convert the value to a display value so that the net weight value is displayed on the HMI.

The operating procedure for the sample programming is as follows.

Step No.	Operation	Description
1	Checking net weight value near zero	Judge whether or not the net weight value which is input from a Load Cell Input Unit is near zero. If the value is near zero, execute feed weighing.
2	Setting fixed weight and executing fall compensation	• Set the fixed weight set value. • Execute fall compensation.
3	Executing feed weighing	• Set the set values for the WC_FixedWeightCtrl function block. • Control valves and execute feed weighing. • After raw material is discharged, execute judgment for the new weight.
4	Executing the judgment result processing	Execute processing as follows depending on the judgment result. The weight is normal: Convert the value to a display value and display the net weight value on the HMI. Open the discharging valve. The weight is short: Execute additional feed weighing with small feed only. The fixed weight value is exceeded: Turn ON the weight over flag.

Set Value

This sample programming sets the set values for function blocks or functions as follows.

FB or FUN	Parameter	Value
WC_FixedWeightCtrl	Fixed weight set value	100g
	Large feed set value	50g
	Pre-fixed weight set value	20g
	Fall set value	It is not set because fall compensation is performed.
	Weight over set value	1g
	Weight short set value	1g
	Comparison banned time 1	100ms
	Comparison banned time 2	100ms
	Judgment timeout time	500ms
	Execution timeout time	1s
WC_FallCompensation	Fixed weight set value	100g
	Fall Regulation value	1g
	Differential memory size	9
WC_WeightIndication	Scale Interval Mantissa	1
	Scale Interval Exponent	0
	Maximum capacity coefficient	1000
	Minimum display coefficient	1000

Variables

● Internal Variables

Name	Data Type	Initial Value	AT	Comment
FixedWeightVal	REAL	0		Fixed weight set value
HoldingMeasuringVal	REAL	0		Hold weight value
Start_WeightCtrl	BOOL	FALSE		Start flag for feed weighing
Start_CalcTrigger	BOOL	FALSE		Execution flag for fall compensation
FallCompVal	REAL	0		Fall compensation value
CalcCountNum	UINT	0		Differential memory size
Int_FallComp	OmronLib\WC_Toolbox\WC_FallCompensation			Instance of the WC_FallCompensation function block
SignInversion	BOOL	FALSE		Discharging method
Shortage Weight	BOOL	FALSE		Execution flag for processing additional feed weighing
AbortWeightCtrl	BOOL	FALSE		Abort for quantitative discharging control
Int_FixedWeightCtrl	OmronLib\WC_Toolbox\WC_FixedWeightCtrl			Instance of the WC_FixedWeightCtrl function block

Name	Data Type	Initial Value	AT	Comment
FixedWeightCtrlParams	OmronLib\WC_Toolbox\s_FIXED_WEIGHT_CTRL_PARAMS			Quantitative discharging control settings
GWC_Done	BOOL	FALSE		Completion for quantitative discharging control
CorrectedWeight	BOOL	FALSE		Normal
MeasuringVal	REAL	0		Weight value
OverWeight	BOOL	FALSE		Exceeded
Center	BOOL	FALSE		Center value judgment result
MaxCapacityOver	BOOL	FALSE		Maximum capacity over judgment result
MinimumDisplayUnder	BOOL	FALSE		Minimum display under judgment result
For_In_NA	BOOL	FALSE		Output signal to display on the HMI
ValueForDisplay	STRING[10]	''		Display value mantissa
ExponentForDisplay	INT	0		Display value exponent
NearZero	REAL	0		Judgment value near zero
WC1_FixedWeightCtrlParams	OmronLib\WC_Toolbox\s_FIXED_WEIGHT_CTRL_PARAMS			
WC1_ReguVal	REAL	0		
WC1_FWC_Status	UINT	0		
WC1_FWC_Judgement	UINT	0		
WC1_EXWC	BOOL	FALSE		
WC1_Start	BOOL	FALSE		

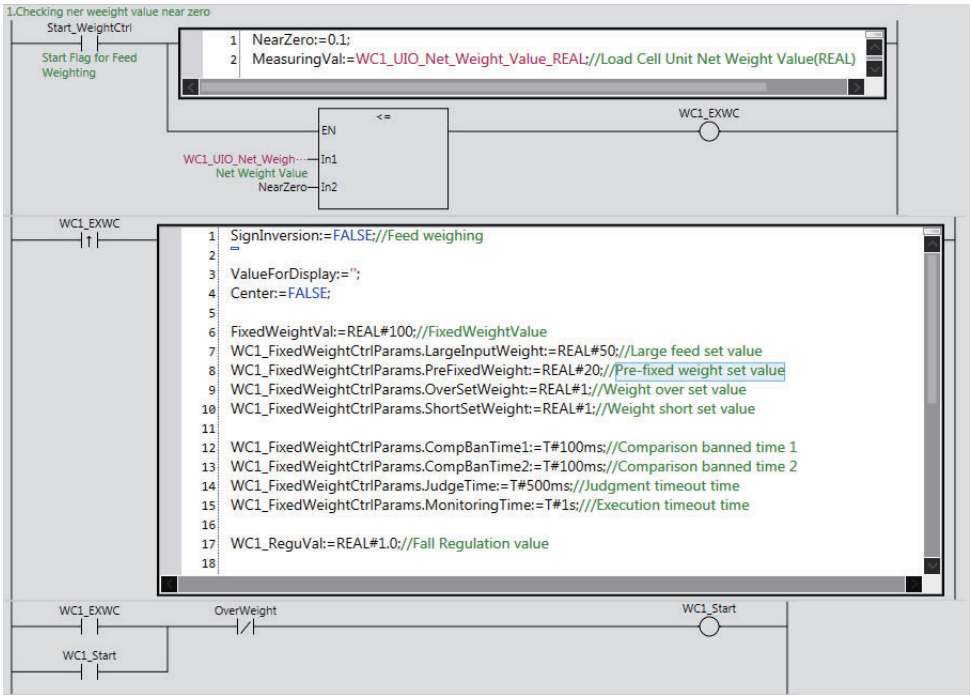
External Variables

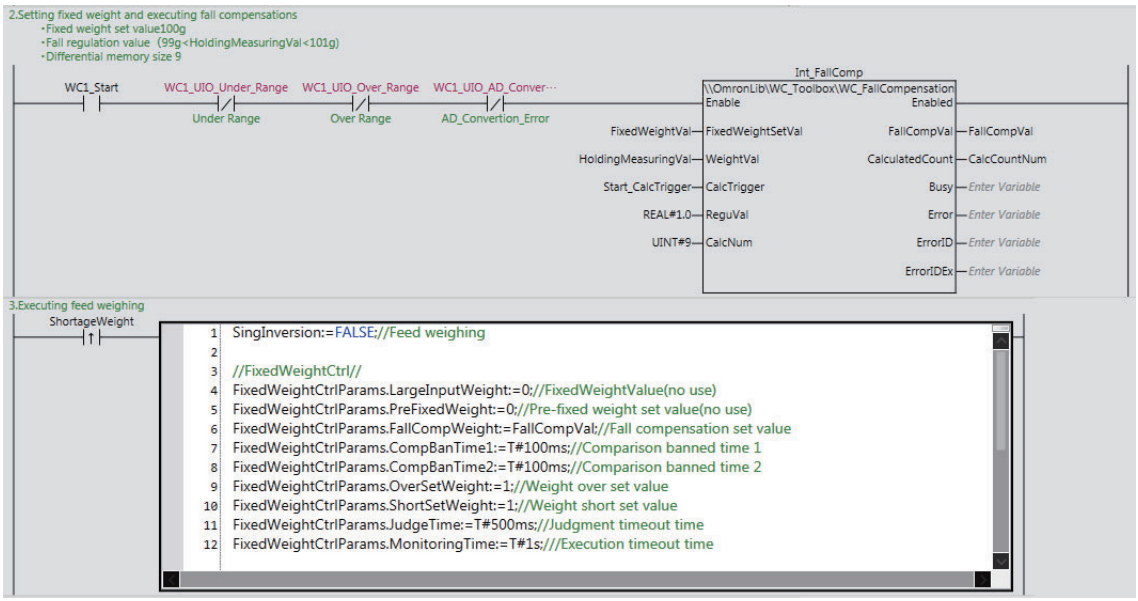
Name	Data Type	Comment
WC1_UIO_Net_Weight_Value_REAL	REAL	Net weight value UIO
WC1_UIO_Stable_Status	BOOL	Stable status UIO
WC1_UIO_AD_Convertion_Error	BOOL	Load Cell Input Unit Ch1 AD Convertion Error
WC1_UIO_OnetouchTareSubtractionExecution	BOOL	Load Cell Input Unit Ch1 One-touch Tare Subtraction Execution
WC1_UIO_Under_Range	BOOL	Load Cell Input Unit Ch1 Under Range
WC1_UIO_Over_Range	BOOL	Load Cell Input Unit Ch1 Over Range
WC1_UIO_LargeSwitch	BOOL	Large feed valve output signal
WC1_UIO_MiddleSwitch	BOOL	Medium feed valve output signal
WC1_UIO_SmallSwitch	BOOL	Small feed valve output signal
WC1_UIO_DischargeSwitch	BOOL	Discharge valve output signal

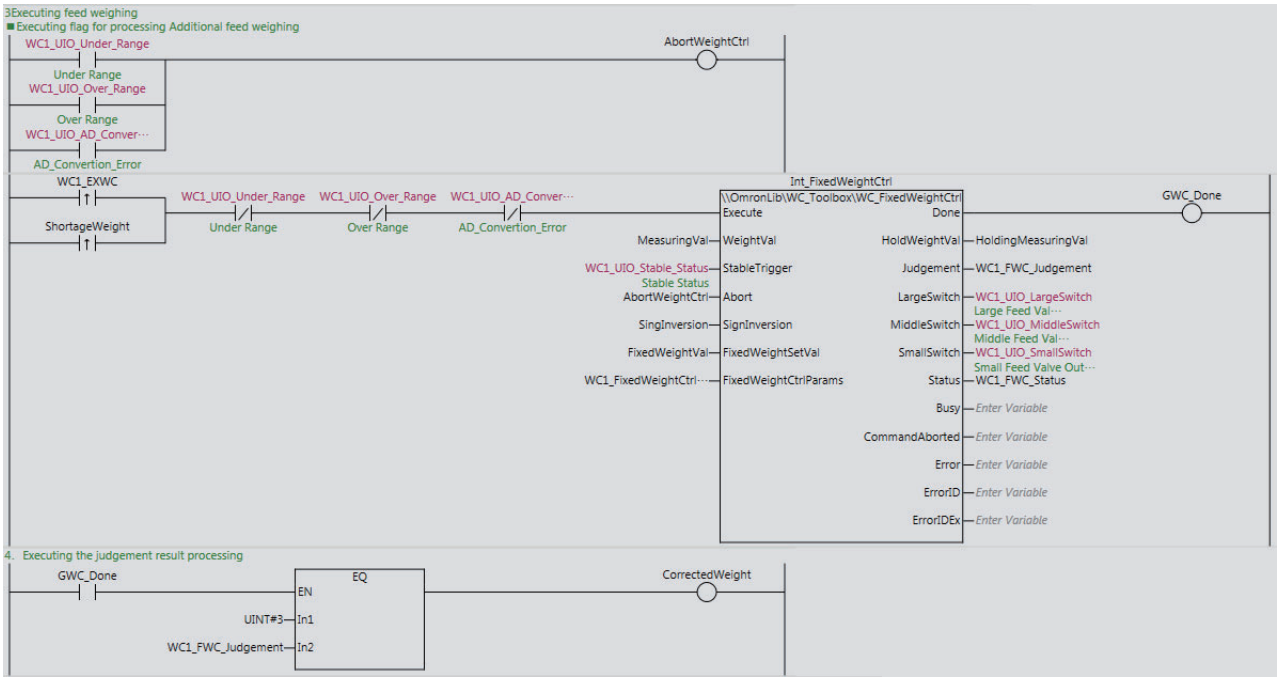
Global Variables

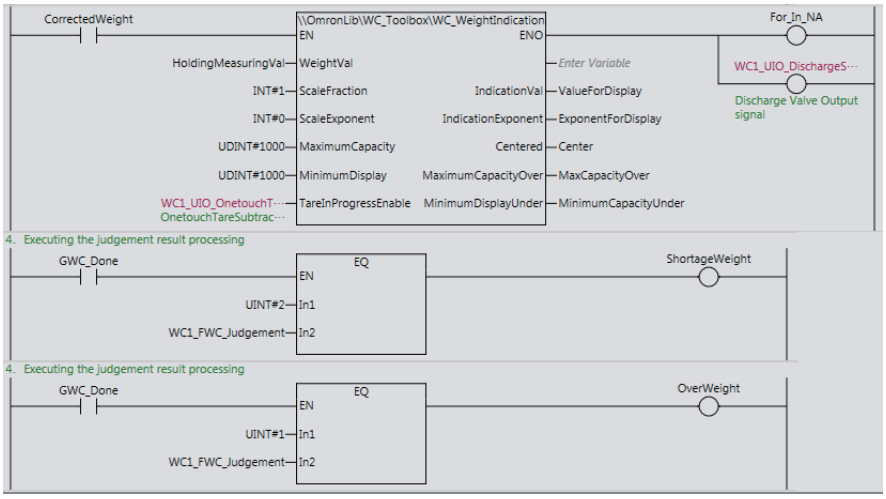
Name	Data Type	Initial Value	AT	Comment
WC1_UIO_Stable_Status	BOOL		ECAT://node#[1,1]/Ch1 Detection Status/Ch1 Stable Status	Load Cell Input Unit Ch1 Stable Status
WC1_UIO_Net_Weight_Value_REAL	REAL		ECAT://node#[1.1]/Ch1 Net Weight Value REAL	Load Cell Input Unit Ch1 Net Weight Value (REAL)
WC1_UIO_Under_Range	BOOL		ECAT://node#[1,1]/Ch1 Detection Status/Ch1 Under Range	Load Cell Input Unit Ch1 Under Range
WC1_UIO_Over_Range	BOOL		ECAT://node#[1,1]/Ch1 Detection Status/Ch1 Over Range	Load Cell Input Unit Ch1 Over Range
WC1_UIO_OnetouchTareSubtractionExecution	BOOL		ECAT://node#[1,1]/Ch1 Operation Command/Ch1 One-touch Tare Subtraction Execution	Load Cell Input Unit Ch1 One-touch Tare Subtraction Execution
WC1_UIO_AD_Convertion_Error	BOOL			Load Cell Input Unit Ch1 AD Convertion Error
WC1_UIO_LargeSwitch	BOOL		ECAT://node#[1,2]/Output Bit 00	Large feed valve output signal
WC1_UIO_MiddleSwitch	BOOL		ECAT://node#[1,2]/Output Bit 01	Medium feed valve output signal
WC1_UIO_SmallSwitch	BOOL		ECAT://node#[1,2]/Output Bit 02	Small feed valve output signal
WC1_UIO_DischargeSwitch	BOOL		ECAT://node#[1,2]/Output Bit 03	Discharge valve output signal

Ladder Diagram



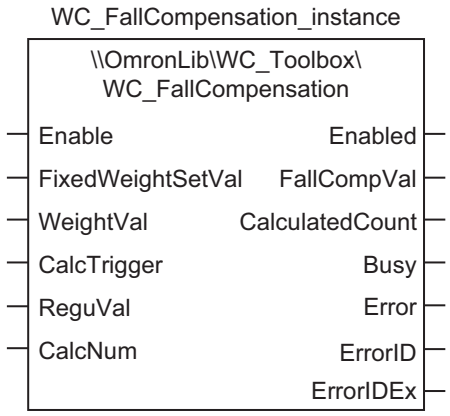



3Executing feed weighing
Executing flag for processing Additional feed weighing
WC1_UIO_Under_Range
Under Range
WC1_UIO_Over_Range
Over Range
WC1_UIO_AD_Conver···
AD_Convertion_Error
AbortWeightCtrl
WC1_EXWC
ShortageWeight
Int_FixedWeightCtrl
\\OmronLib\WC_Toolbox\WC_FixedWeightCtrl
Execute
Done
GWC_Done
MeasuringVal
WeightVal
HoldWeightVal
HoldingMeasuringVal
WC1_UIO_Stable_Status
Stable Status
StableTrigger
Judgement
WC1_FWC_Judgement
AbortWeightCtrl
Abort
LargeSwitch
WC1_UIO_LargeSwitch
Large Feed Val···
SingInversion
SignInversion
MiddleSwitch
WC1_UIO_MiddleSwitch
Middle Feed Val···
FixedWeightVal
FixedWeightSetVal
SmallSwitch
WC1_UIO_SmallSwitch
Small Feed Valve Out···
WC1_FixedWeightCtrl···
FixedWeightCtrlParams
Status
WC1_FWC_Status
Busy
Enter Variable
CommandAborted
Error
ErrorID
ErrorIDEx
4. Executing the judgement result processing
GWC_Done
EQ
EN
UINT#3
In1
WC1_FWC_Judgement
In2
CorrectedWeight


CorrectedWeight
\\OmronLib\WC_Toolbox\WC_WeightIndication
EN
ENO
For_In_NA
HoldingMeasuringVal
WeightVal
Enter Variable
WC1_UIO_DischargeS···
INT#1
ScaleFraction
IndicationVal
ValueForDisplay
Discharge Valve Output signal
INT#0
ScaleExponent
IndicationExponent
ExponentForDisplay
UDINT#1000
MaximumCapacity
Centered
Center
UDINT#1000
MinimumDisplay
MaximumCapacityOver
MaxCapacityOver
WC1_UIO_OnetouchT···
OnetouchTareSubtrac···
TareInProgressEnable
MinimumDisplayUnder
MinimumCapacityUnder
4. Executing the judgement result processing
GWC_Done
EQ
EN
UINT#2
In1
WC1_FWC_Judgement
In2
ShortageWeight
4. Executing the judgement result processing
GWC_Done
EQ
EN
UINT#1
In1
WC1_FWC_Judgement
In2
OverWeight

WC_FallCompensation

The WC_FallCompensation function block performs fall compensation for quantitative discharging control.

Function block name	Name	FB/ FUN	Graphic expression	ST expression
WC_Fall Compensation	Fall Compensation	FB		WC_FallCompensation_instance(Enable, FixedWeightSetVal, WeightVal, CalcTrigger, ReguVal, CalcNum, Enabled, FallCompVal, CalculatedCount, Busy, Error, ErrorID, ErrorIDEx);

Function Block and Function Information

Item	Description
Library file name	OmronLib_WC_Toolbox_V1_0.slr
Namespace	OmronLib\WC_Toolbox
Function block and function number	00047
Publish/Do not publish source code	Do not publish
Function block and function version	1.00

Compatible Models

Item	Name	Model numbers	Version
Device	Load Cell Input Unit	NX-RS1201	Version 1.00 or later

Variables

Input Variables

Name	Meaning	Data type	Default	Valid range	Unit	Description
Enable	Enable	BOOL	FALSE	Depends on data type.	---	Enabled trigger for this function block Enables the function block when it changes to TRUE.
FixedWeight SetVal	Fixed Weight Set Value	REAL	0.0	Depends on data type.	User defined*1	Fixed weight set value
WeightVal	Weight Value	REAL	0.0	-3.402823e + 38 < WeightVal < 3.402823e + 38	User defined*1	Measurement value
CalcTrigger	Calculation trigger	BOOL	FALSE	Depends on data type.	---	Calculation trigger for this function block Starts calculation when it changes to TRUE.
ReguVal	Fall Regulation Value	REAL	0.0	Depends on data type.	User defined*1	Threshold to select *WeightVal* used for fall compensation calculation.
CalcNum	Differential Memory Size	UINT	1	1 to 9	---	Maximum number of differences between *WeightVal* and *FixedWeightSetVal* stored in differential memory

*1. The same unit is required for *FixedWeightSetVal*, *WeightVal* and *ReguVal*.

Output Variables

Name	Meaning	Data type	Valid range	Unit	Description
Enabled	Enabled	BOOL	Depends on data type.	---	Enabled Changes to TRUE while processing.
FallCompVal	Fall Compensation Value	REAL	Depends on data type.	User defined *1	Fall Compensation Value
CalculatedCount	Differential Memory Counter	UINT	0 to 9	---	Number of differences between *WeightVal* and *FixedWeightSetVal* in differential memory
Busy	Executing	BOOL	Depends on data type.	---	Executing TRUE: Execution processing is in progress. FALSE: Execution processing is not in progress.
Error	Error	BOOL	Depends on data type.	---	Error end TRUE: Error end FALSE: Normal end, execution in progress, or execution condition not met
ErrorID	Error Code	WORD	*2	---	Error code when an error occurs WORD#16#0 when normal
ErrorIDEx	Expansion Error Code	DWORD	*2	---	Expansion error code when an error occurs DWORD#16#0 when normal

*1. Same unit as *FixedWeightSetVal* and *WeightVal*.

*2. Refer to *Troubleshooting* on page 74 for details.

Function

This function block performs fall compensation calculation for quantitative discharging control.

With quantitative discharging control, some of the raw material floats in the air even after the valve is completely shut. Therefore, set the fall set value in consideration of the weight of the raw material that floats in the air. However, the weight of the raw material that floats in the air changes depending on changes in the environment. Fall compensation refers to the use of the actual difference between the past measurement value and the fixed weight set value to compensate the fall set value. It is possible to reduce the error difference between the fixed weight set value and final measurement value through fall compensation.

Parameter Input from Load Cell Input Unit

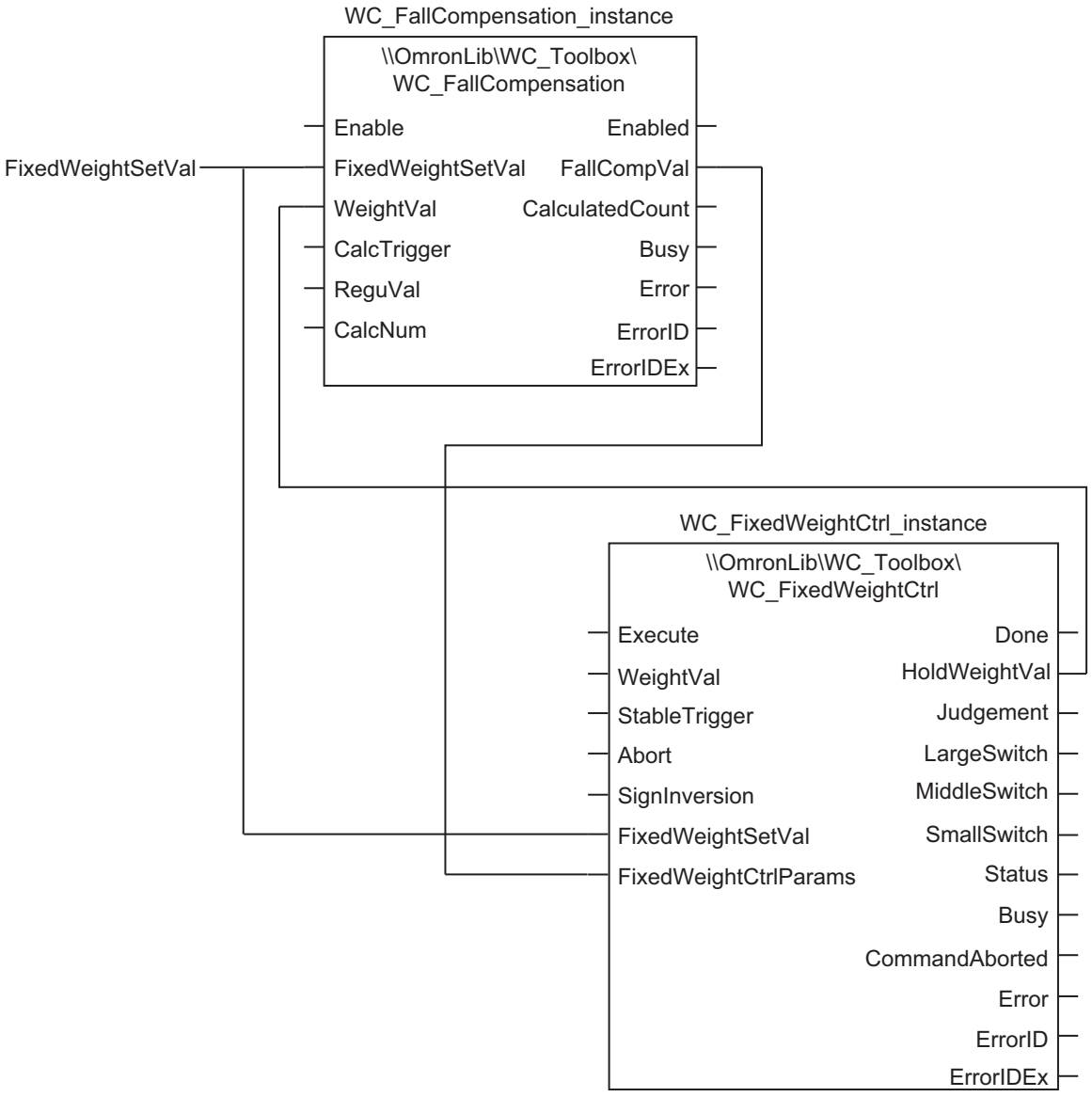
Enter the Ch1 Net Weight Value REAL of the Load Cell Input Unit for *WeightVal* (Weight Value). It is necessary to perform I/O allocation of the Ch1 Net Weight Value REAL through the settings of the Load Cell Input Unit in order to enter the value. Refer to the *NX-series Load Cell Input Unit User's Manual* (Cat. No. W565) for details on the Load Cell Input Unit specifications.

Connection with the WC_FixedWeightCtrl Function Block

This function block is used together with the WC_FixedWeightCtrl function block. The connections for both are as follows.

- Use the same value for the input parameter named *FixedWeightSetVal* (Fixed Weight Set Value) of both function blocks.
- Connect the WC_FixedWeightCtrl function block output parameter named *HoldWeightVal* (Held Weight Value) to the WC_FallCompensation function block input parameter named *WeightVal* (Weight Value).
- Connect the WC_FallCompensation function block output parameter named *FallCompVal* (Fall Compensation Value) to *FallCompWeight* (Fall Set Value) of the WC_FixedWeightCtrl function block.

The connection diagram between this function block and the WC_FixedWeightCtrl function block is as follows.



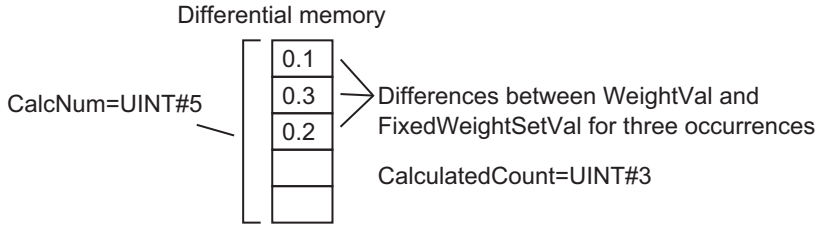
Fall Compensation Calculation

There is a differential memory that stores the differences between the values for *WeightVal* (Weight Value) and *FixedWeightSetVal* (Fixed Weight Set Value) every time the *CalcTrigger* (Calculation Trigger) value changes from FALSE to TRUE in this function block. The maximum number of values that are stored in the differential memory is set with *CalcNum* (Differential Memory Size). Also, the current number of stored values in the differential memory is shown with *CalculatedCount* (Differential Memory Counter).

As an example, when *CalcNum* = UINT#5, *FixedWeightSetVal* = REAL#20.0 and there are three *CalcTrigger* inputs, the values stored in differential memory are shown in the following figure. The value s for *WeightVal* for three occurrences are REAL#20.1, REAL#20.3 and REAL#20.2.

Values of WeightVal for three occurrences FixedWeightSetVal=REAL#20.0

REAL#20.1
REAL#20.3
REAL#20.2



The *FallCompVal* (Fall Compensation Value) is calculated with the following formula based on these parameters.

$$\text{FallCompVal} = \frac{\Sigma\ (\text{Wn-F})}{\text{n}}$$

n : CalculatedCount

Wn : WeightVal for n th time

F : FixedWeightSetVal

Therefore, the value for *FallCompVal* is REAL#0.2 based on the above example.

$$\text{FallCompVal} = \frac{(0.1+0.3+0.2)}{3}$$

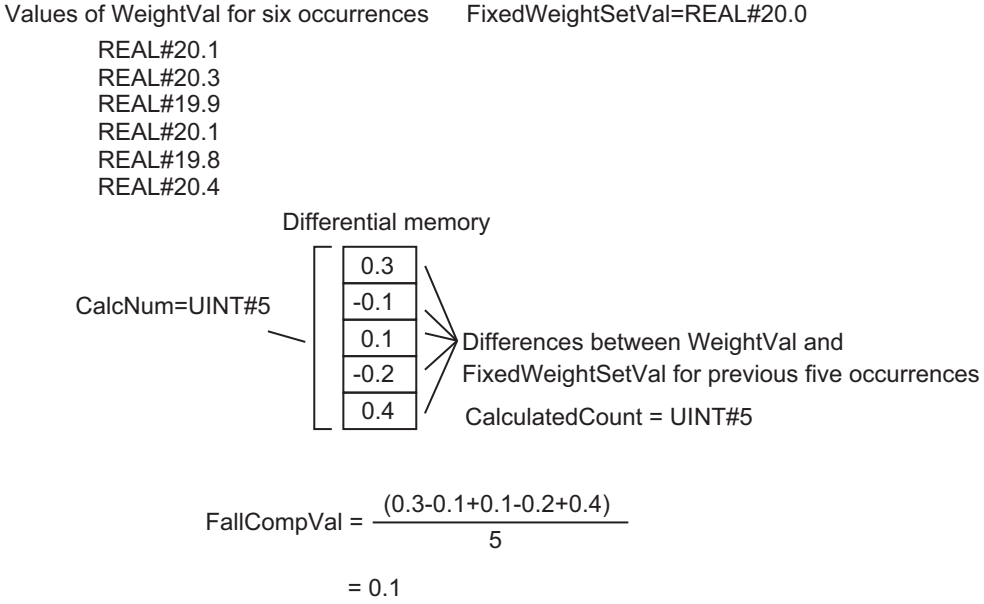
$$= 0.2$$

● Fall Compensation Calculation when Number of CalcTrigger Inputs Greater than CalcNum

Storage for the number of values for *CalcNum* (Differential Memory Size) is solely possible. When *CalcTrigger* (Calculation Trigger) that exceeds *CalcNum* is input, the differential memory is overwritten from the oldest value to the latest. Therefore, the value for *CalculatedCount* (Differential Memory Counter) becomes the same value as for *CalcNum*.

When *CalcNum* = UINT#5, *FixedWeightSetVal* = REAL#20.0 and there are six *CalcTrigger* inputs, processing is performed as follows. The values for *WeightVal* for six occurrences are REAL#20.1, REAL#20.3, REAL#19.9, REAL#20.1, REAL#19.8 and REAL#20.4.

The value for *FallCompVal* is REAL#0.1.



$$\text{FallCompVal} = \frac{(0.3-0.1+0.1-0.2+0.4)}{5}$$

$$= 0.1$$

Fall Regulation Value

When the difference between the values for *FixedWeightSetVal* (Fixed Weight Set Value) and *WeightVal* (Weight Value) is too large, you can exclude the value from the fall compensation calculation. The threshold to determine whether to use a value or not is *ReguVal* (Fall Regulation Value).

If the absolute difference between the values for *FixedWeightSetVal* and *WeightVal* exceeds the value of *ReguVal*, the value is not stored in differential memory. Also, the values for *CalculatedCount* (Differential Memory Counter) and *FallCompVal* (Fall Compensation Value) do not change.

Example of Fall Compensation Calculation

An example of a fall compensation calculation is as follows. The value of each input parameter is as follows.

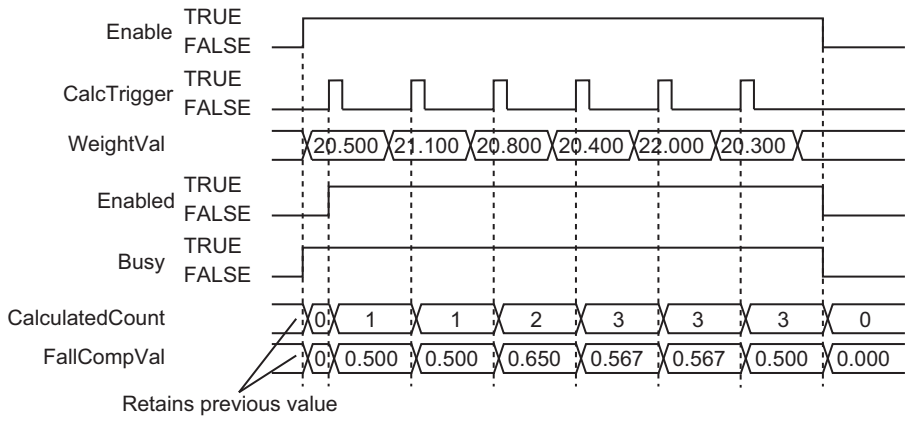
Input parameter	Value
FixedWeightSetVal (Fixed Weight Set Value)	20.000
ReguVal (Fall Regulation Value)	1.000
CalcNum (Differential Memory Size)	3

The following chart shows an example of differential memory contents, *CalculatedCount* (Differential Memory Counter), and FallCompVal (Fall Compensation Value) when *CalcTrigger* (Calculation Trigger) is input six times.

Number of CalcTrigger inputs	WeightVal value	Differential memory	Calculated-Count value	FallCompVal value	Description
0		0	0	0.000	---
1	20.500	0.500	1	0.500	---
2	21.100	0.500	1	0.500	The value is not stored in differential memory, as the difference between *FixedWeightSetVal* and *WeightVal* is larger than *ReguVal*.
3	20.800	0.500 0.800	2	0.650	---
4	20.400	0.500 0.800 0.400	3	0.567	---
5	22.000	0.500 0.800 0.400	3	0.567	The value is not stored in differential memory, as the difference between *FixedWeightSetVal* and *WeightVal* is larger than *ReguVal*.
6	20.300	0.800 0.400 0.300	3	0.500	The oldest value, 0.500, is overwritten in the differential memory.

The changes in the parameters in the above chart are shown in the following figure.

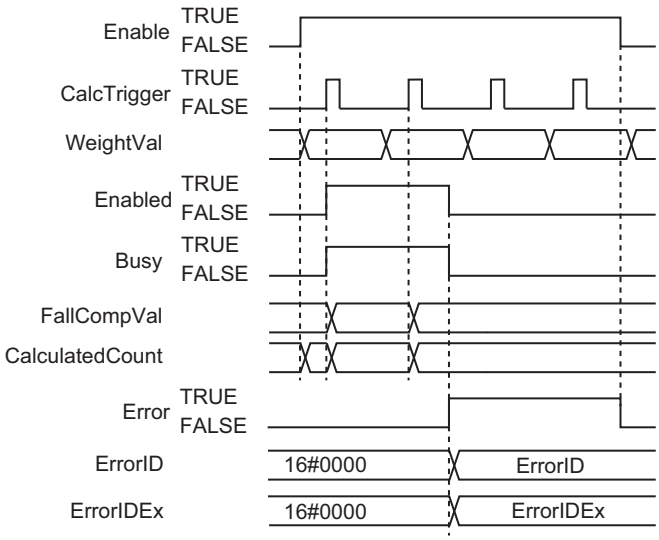
- The value for *FallCompVal* (Fall Compensation Value) is refreshed every time the *CalcTrigger* (Calculation Trigger) is input.
- The value for *Enabled* (Enabled) is stored in differential memory and changes to TRUE when the value for *FallCompVal* is enabled.
- When the value for *Enable* (Enable) is changed to FALSE, the value for *CalculatedCount* (Differential Memory Counter) is initialized to UINT#0 and the value for *FallCompVal* is initialized to REAL#0.0. Also, the value for *Enabled* changes to FALSE.



Timing Chart

The timing chart when an error occurs is as follows. Refer to *Example of Fall Compensation Calculation* on page 72 for the timing chart for normal end.

- The value for *Error* changes to TRUE when an error occurs. A value based on the error is stored for *ErrorID* (Error Code) and *ErrorIDEx* (Expansion Error Code).
- The value for *Error* (Error) is TRUE while the value for *Enable* (Enable) is TRUE.
- The value for *Enabled* (Enabled) changes to FALSE when an error occurs.
- After the error occurs, the values for *FallCompVal* (Fall Compensation Value) and *CalculatedCount* (Differential Memory Counter) are retained.



Precautions for Correct Use

- The value for *CalcNum* (Differential Memory Size) when *Enable* (Enable) changes to TRUE is valid. The value is not refreshed even if it is changed during processing of this function block.
- If the value for *FixedWeightSetVal* (Fixed Weight Set Value), *WeightVal* (Weight Value) or *ReguVal* (Fall Regulation Value) is changed while this function block is in process, the value is refreshed for the processing within the same task period.
- The data type for *FixedWeightVal* (Fixed Weight Set Value), *WeightVal* (Weight Value) and *ReguVal* (Fall Regulation Value) is REAL. Therefore, the value is possibly inaccurate over 7 digits.

Troubleshooting

The error codes, expansion error codes, statuses, descriptions and corrections are as follows.

Error code	Expansion error code	Status	Description	Correction
16#0000	16#00000000	Normal End		
16#3C31	16#00000001	*FixedWeightSetVal* Outside Input Range	The value for *FixedWeightSetVal* (Measurement Set Value) is outside the input range	Check the valid range of the value for *FixedWeightSetVal* (Fixed Weight Set Value) and set the value within the valid range.
	16#00000002	*WeightVal* Outside Input Range	The value for *WeightVal* (Weight Value) is outside the input range	Check the valid range of the value for *WeightVal* (Weight Value) and set the value within the valid range.
	16#00000003	*ReguVal* Outside Input Range	The value for *ReguVal* (Fall Regulation Value) is outside the input range	Check the valid range of the value for *ReguVal* (Fall Regulation Value) and set the value within the valid range.
	16#00000004	*CalcNum* Outside Input Range	The value for *CalcNum* (Differential Memory Size) is outside the input range	Check the valid range of the value for *CalcNum* (Differential Memory Size) and set the value within the valid range.

Sample Programming

Refer to the sample programming for *WC_FixedWeightCtrl* on page 42.

Appendix

Referring to Library Information

When you make an inquiry to OMRON about the 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.

Attributes of Libraries, Function Blocks and Functions

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

● Attributes of Libraries

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 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, *Referring to Attributes of Libraries, Function Blocks and Functions* on page 77.

*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 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 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, *Referring to Attributes of Libraries, Function Blocks and Functions* on page 77.

*2. It is provided in English and Japanese.

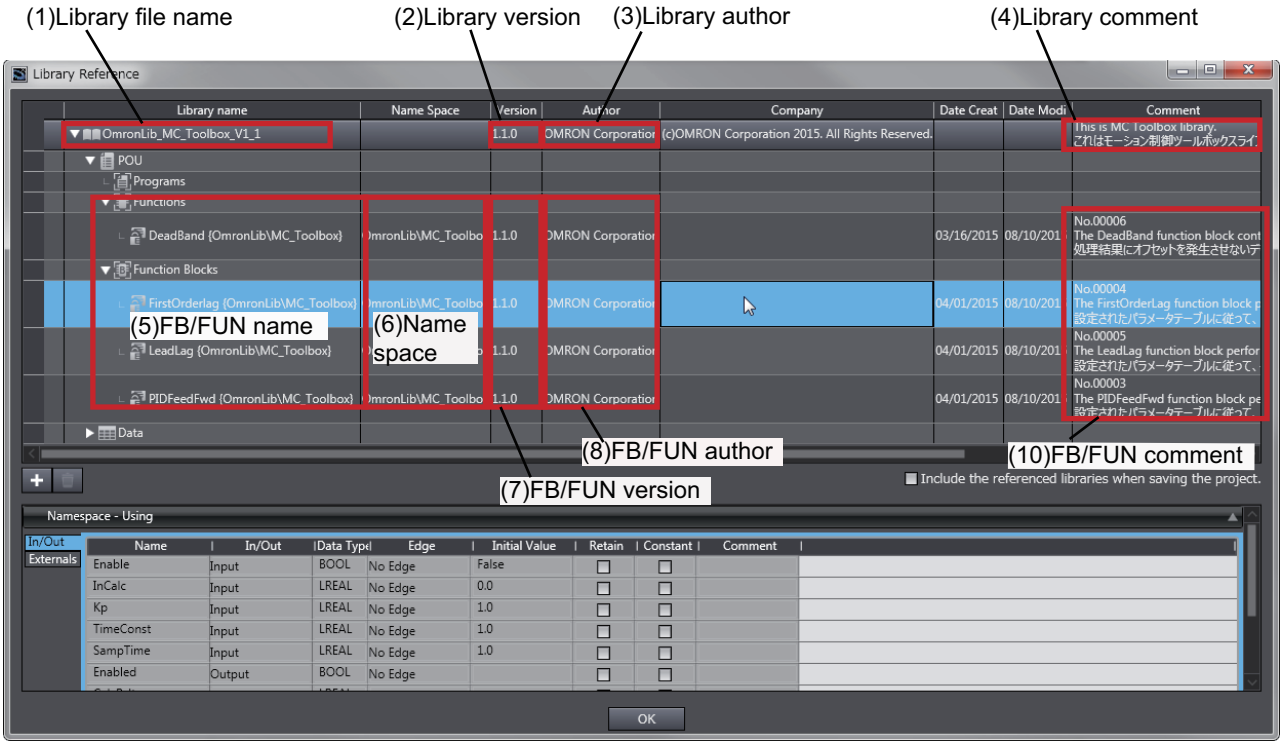
Referring to Attributes of Libraries, Function Blocks and Functions

You can refer to the attributes of libraries, function blocks and functions of the library information at the following locations on the Sysmac Studio.

- Library Reference Dialog Box
- Toolbox Pane
- Ladder Editor

(a) Library Reference Dialog Box

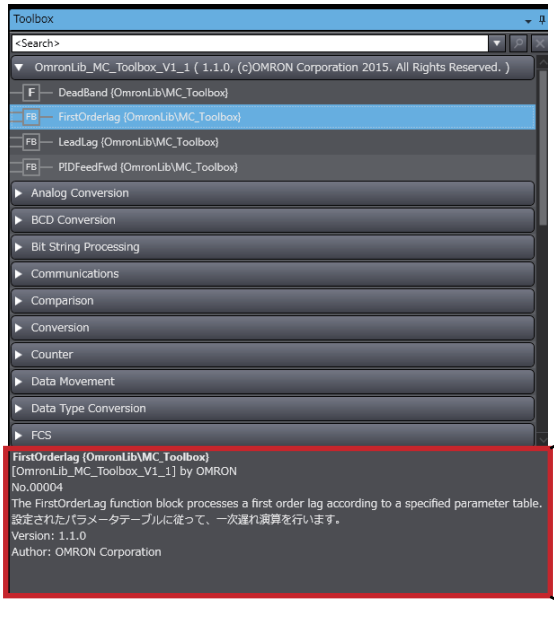
When you refer to the libraries, the library information is displayed at the locations shown below.



(b) Toolbox Pane

Select a function block and function to display its library information at the bottom of the Toolbox Pane.

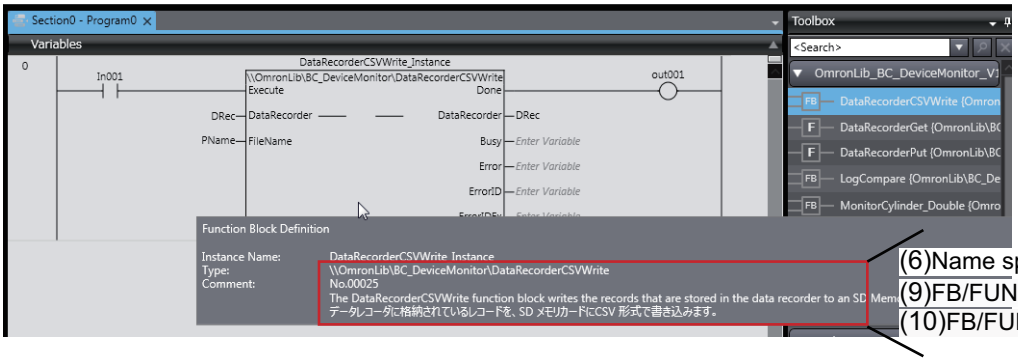
The text “by OMRON” which is shown on the right of the library name (1) indicates that this library was provided by OMRON.



(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

(c) Ladder Editor

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



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

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.

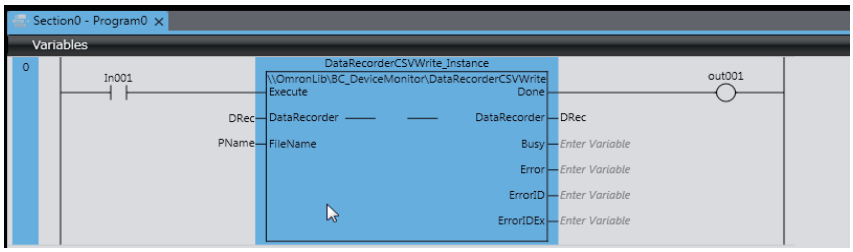
The following are the examples of items that you may need to customize.

- Customizing the size of arrays to suit the memory capacity of the user's Controller
- Customizing the data types 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.

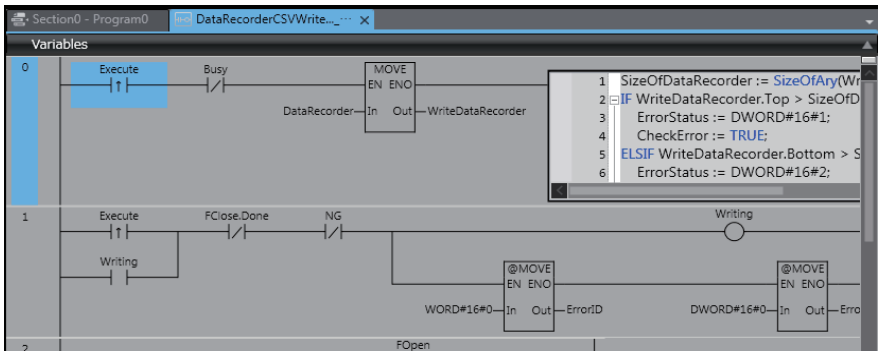
Use the following procedure to refer to the source codes of function blocks and functions.

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.

OMRON Corporation **Industrial Automation Company**
Kyoto, JAPAN
Contact: www.ia.omron.com

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

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

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

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

Authorized Distributor:

Cat. No. W569-E1-03 0119